



PIC18F2331/2431/4331/4431

数据手册

采用纳瓦技术、高性能 PWM 和 A/D 的
28/40/44 引脚增强型闪存单片机

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最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

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

提供本文档的中文版本仅为了便于理解。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适用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。未经 Microchip 书面批准，不得将 Microchip 的产品用作生命维持系统中的关键组件。在 Microchip 知识产权保护下，不得暗或以其他方式转让任何许可证。

商标

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

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

Analog-for-the-Digital Age、Application Maestro、dsPICDEM、dsPICDEM.net、dsPICworks、ECAN、ECONOMONITOR、FanSense、FlexROM、fuzzyLAB、In-Circuit Serial Programming、ICSP、ICEPIC、Linear Active Thermistor、MPASM、MPLIB、MPLINK、MPSIM、PICKit、PICDEM、PICDEM.net、PICLAB、PICtail、PowerCal、PowerInfo、PowerMate、PowerTool、rfLAB、rfPICDEM、Select Mode、Smart Serial、SmartTel、Total Endurance 和 WiperLock 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。

© 2005, Microchip Technology Inc. 版权所有。

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

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 及位于加利福尼亚州 Mountain View 的全球总部、设计中心和晶圆生产厂均于 2003 年 10 月通过了 ISO/TS-16949:2002 质量体系认证。公司在 PICmicro® 8 位单片机、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品方面的质量体系流程均符合 ISO/TS-16949:2002。此外，Microchip 在开发系统的设计和和生产方面的质量体系也已通过了 ISO 9001:2000 认证。



MICROCHIP

PIC18F2331/2431/4331/4431

采用纳瓦技术、高性能 PWM 和 A/D 的 28/40/44 引脚增强型闪存单片机

14 位功率控制 PWM 模块:

- 多达 4 个具有互补输出的通道
- 边沿或中心对齐的操作
- 灵活的死区发生器
- 硬件故障保护输入
- 占空比和周期同步更新
 - 灵活的特殊事件触发器输出

运动反馈模块:

- 三个独立的输入捕捉通道:
 - 灵活的周期和脉冲宽度测量操作模式
 - 特殊霍尔传感器接口模块
 - 输出到其他模块的特殊事件触发器
- 正交编码器接口:
 - 来自编码器的 2 相输入和一个索引输入
 - 带方向状态和方向变化中断的高低位置追踪
 - 速率测量

高速的 200 Ksps 10 位 A/D 转换器:

- 多达 9 个通道
- 两个通道同步采样
- 连续采样: 1、2 或 4 个选定通道
- 自动转换能力
- 带可选中断频率的 4 字 FIFO
- 可选的外部转换触发器
- 可编程的采集时间

灵活的振荡器结构:

- 4 种晶振模式, 频率高达 40 MHz
- 两个外部时钟模式, 频率高达 40 MHz
- 内部振荡电路:
 - 8 个用户可选频率: 从 31 kHz 到 8 MHz
 - OSCTUNE 可以对频率漂移进行补偿
- 辅助振荡器使用 Timer1 (32 kHz 时)
- 故障保护时钟监视器:
 - 如果时钟发生故障, 允许器件安全关断

功耗管理模式:

- 运行 CPU 打开, 外设打开
- 空闲 CPU 关闭, 外设打开
- 休眠 CPU 关闭, 外设关闭
- 通常空闲模式电流降至 5.8 μ A
- 通常休眠模式电流降至 0.1 μ A
- Timer1 振荡器典型参数为 1.8 μ A、32 kHz、2V
- 看门狗定时器 (WDT), 典型值为 2.1 μ A
- 双速振荡器起振

外设特点:

- 灌电流 / 拉电流峰值 25 mA/25 mA
- 三个外部中断
- 两个捕捉 / 比较 / PWM (CCP) 模块:
 - 捕捉为 16 位, 最大分辨率为 6.25 ns (Tcy/16)
 - 比较为 16 位, 最大分辨率为 100 ns (Tcy)
 - PWM 输出: PWM 分辨率为 1 到 10 位
- 增强型 USART 模块:
 - 支持 RS-485、RS-232 和 LIN 1.2
 - 启动位自动唤醒
 - 自动波特率检测
- 使用内部振荡电路的 RS-232 工作模式 (无需外部晶振)

特殊单片机特性:

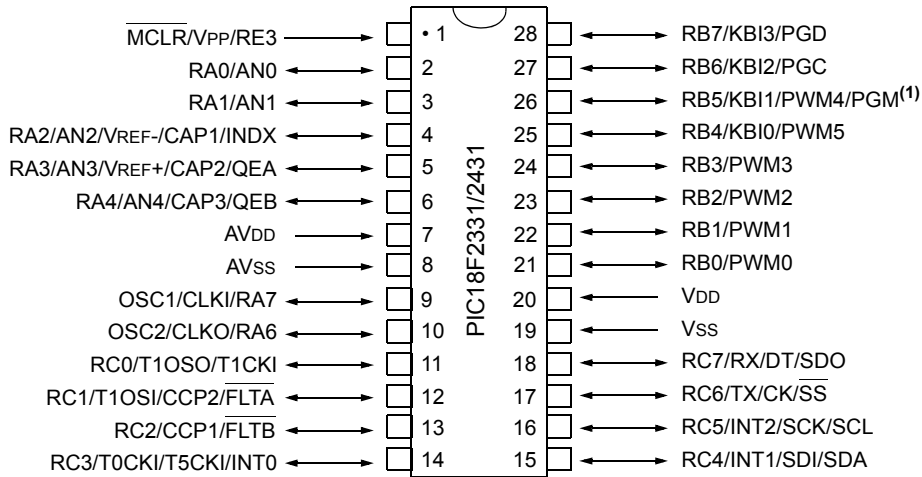
- 可进行 100,000 次擦写操作的增强型闪存程序存储器 (典型值)
- 可进行 1,000,000 次擦写操作的数据 EEPROM 存储器 (典型值)
- 闪存 / 数据 EEPROM 保存时间: 100 年
- 可在软件控制下自行编程
- 中断有优先级
- 8 X 8 单周期硬件乘法器
- 扩展的看门狗定时器 (WDT):
 - 具有 41 ms 到 131s 的可编程周期
- 通过两个引脚进行单电源供电在线串行编程 (In-Circuit Serial Programming™, ICSP™)
- 通过两个引脚在线调试 (In-Circuit Debug, ICD)
 - 调试时安全驱动 PWM 输出

器件	程序存储器		数据存储器		I/O	10 位 A/D (通道)	CCP	SSP		EUSART	正交编码器	14 位 PWM (通道)	定时器 8/16 位
	闪存 (字节)	单字指令数	SRAM (字节)	EEPROM (字节)				SPI	从动 I ² C™				
PIC18F2331	8192	4096	768	256	24	5	2	Y	Y	Y	Y	6	1/3
PIC18F2431	16384	8192	768	256	24	5	2	Y	Y	Y	Y	6	1/3
PIC18F4331	8192	4096	768	256	36	9	2	Y	Y	Y	Y	8	1/3
PIC18F4431	16384	8192	768	256	36	9	2	Y	Y	Y	Y	8	1/3

PIC18F2331/2431/4331/4431

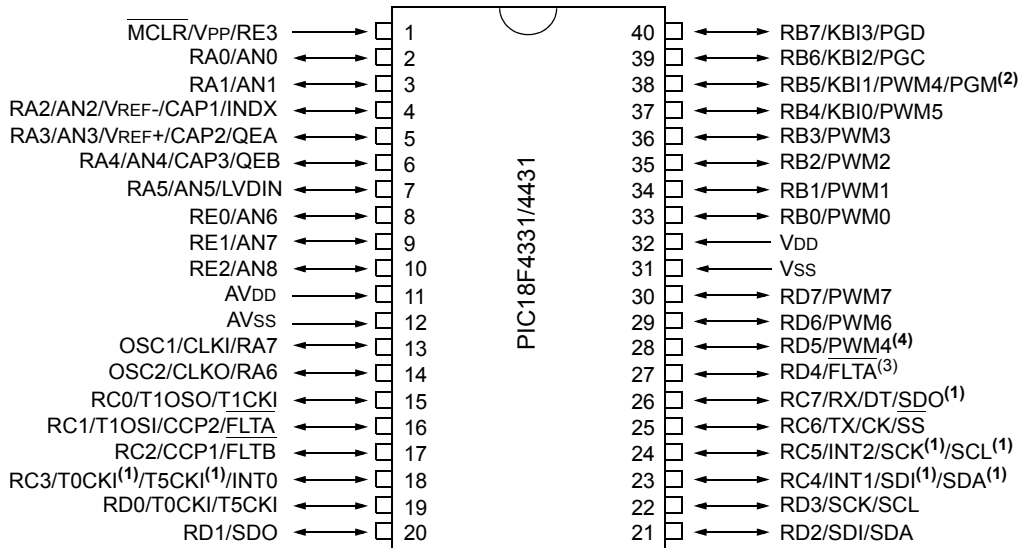
引脚图

28 引脚 SDIP 和 SOIC



注 1: 必须使能低电压编程。

40 引脚 PDIP



注 1: RC3 与 T0CKI/T5CKI 复用一个引脚; RC4 与 SDI/SDA 复用一个引脚; RC5 与 SCK/SCL 复用一个引脚。

2: 必须使能低电压编程。

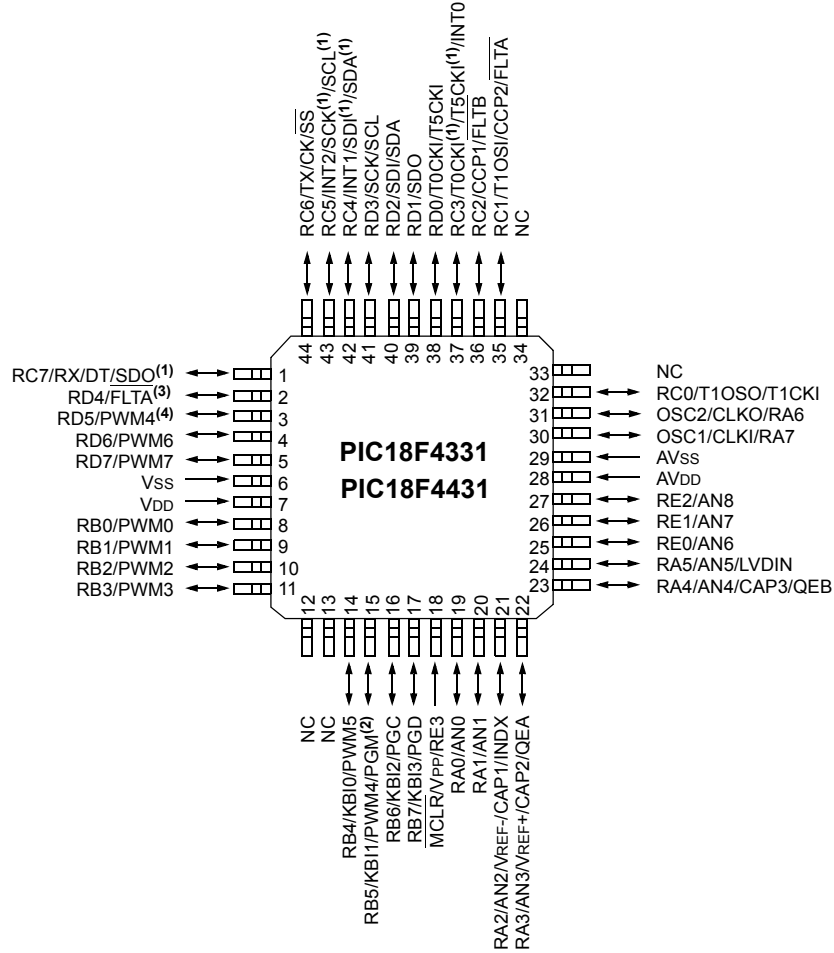
3: RD4 与 FLTA 复用一个引脚。

4: RD5 与 PWM4 复用一个引脚。

PIC18F2331/2431/4331/4431

引脚图 (续)

44 引脚 TQFP

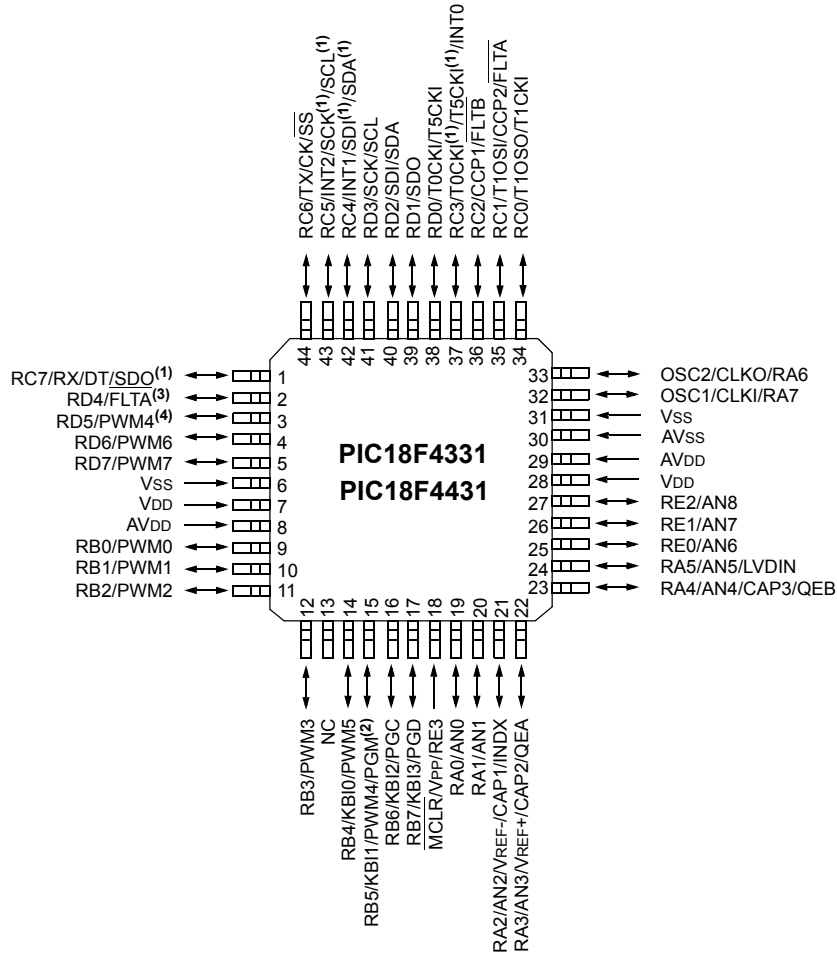


- 注
- 1: RC3 与 T0CKI/T5CKI 复用一个引脚; RC4 与 SDI/SDA 复用一个引脚; RC5 与 SCK/SCL 复用一个引脚。
 - 2: 必须使能低电压编程。
 - 3: RD4 与 FLTA 复用一个引脚。
 - 4: RD5 与 PWM4 复用一个引脚。

PIC18F2331/2431/4331/4431

引脚图 (续)

44 引脚 QFN



- 注 1: RC3 与 T0CKI/T5CKI 复用一个引脚; RC4 与 SDI/SDA 复用一个引脚; RC5 与 SCK/SCL 复用一个引脚。
 2: 必须使能低电压编程。
 3: RD4 与 FLTA 复用一个引脚。
 4: RD5 与 PWM4 复用一个引脚。

PIC18F2331/2431/4331/4431

目录

1.0	器件综述	7
2.0	振荡器配置	21
3.0	功耗管理模式	31
4.0	复位	45
5.0	存储器构成	57
6.0	闪存程序存储器	75
7.0	数据 EEPROM 存储器	85
8.0	8 × 8 硬件乘法器	89
9.0	中断	91
10.0	I/O 端口	107
11.0	Timer0 模块	133
12.0	Timer1 模块	137
13.0	Timer2 模块	143
14.0	Timer5 模块	145
15.0	捕捉 / 比较 / PWM (CCP) 模块	151
16.0	运动反馈模块	159
17.0	功率控制 PWM 模块	181
18.0	同步串行端口 (SSP) 模块	211
19.0	增强型通用同步 / 异步收发器	221
20.0	10 位高速模数转换器 (A/D) 模块	243
21.0	低压检测	261
22.0	CPU 的特殊功能	267
23.0	指令集综述	287
24.0	开发支持	331
25.0	电气特性	337
26.0	直流和交流特性图表初稿	371
27.0	封装信息	373
附录 A:	版本历史	379
附录 B:	器件比较	379
附录 C:	转换注意事项	380
附录 D:	从低档器件移植到增强型器件	380
附录 E:	从中档器件移植到增强型器件	381
附录 F:	从高档器件移植到增强型器件	381
索引		383
Microchip 网站		391
变更通知客户服务		391
客户支持		391
读者反馈表		392
产品标识体系		393

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。

1.0 器件综述

本文档涉及以下器件的具体信息：

- PIC18F2331
- PIC18F4331
- PIC18F2431
- PIC18F4431

该系列具备所有 PIC18 单片机固有的优点，即以实惠的价格提供出色计算性能，以及高耐久性的增强型闪存程序存储器和高速 10 位 A/D 转换器。除了这些优点之外，PIC18F2331/2431/4331/4431 系列还增强了器件设计，使得该系列单片机成为许多高性能、功率控制和电机控制应用的明智选择。这些特别的外设包括：

- 插入可编程死区时间的 14 位分辨率功率控制 PWM 模块（Power Control PWM Module, PCPWM）
- 运动反馈模块（Motion Feedback Module, MFM）包含一个 3 通道输入捕捉（Input Capture, IC）模块和正交编码器接口（Quadrature Encoder Interface, QEI）
- 高速 10 位 A/D 转换器（HSADC）

PCPWM 能够产生至多 8 个插入死区时间的互补 PWM 输出。过驱动电流由片外模拟比较器或数字故障输入引脚（FLTA 和 FLTB）检测。

MFM 正交编码器接口提供精确的转子位置反馈和 / 或速度测量。可以使用 MFM 的 3 个输入捕捉或外部中断检测使用霍尔传感器反馈的电子换相电机应用（例如，BLDC 电机驱动）的转子状态。

PIC18F2331/2431/4331/4431 器件还有闪存程序存储器和一个内建了 LP 振荡模式的内部 RC 振荡器。

1.1 新的内核功能

1.1.1 纳瓦（nanoWatt）技术

PIC18F2331/2431/4331/4431 系列的所有器件具有一系列能在运作时显著降低功耗的功能。关键的几项包括：

- **备用运行模式：**通过将 Timer1 或内部振荡电路作为单片机时钟源，可使代码执行时的功耗降低大约 90%。
- **多种空闲模式：**单片机还可在其 CPU 内核禁止的情况下运行，而外设仍能继续工作。处于这种状态时，功耗能降得更低，甚至降低到只有正常工作需求的 4%。

- **动态模式切换：**运用代码可在器件工作时调用功耗管理模式，以使用户将节约能耗的理念融入到他们的应用软件设计中。
- **关键模块的功耗更低：**Timer1 和看门狗定时器模块的功耗需求降低多达 80%，两者的典型值分别为 1.1 μ A 和 2.1 μ A。

1.1.2 多个振荡器选项和功能

PIC18F2331/2431/4331/4431 系列的所有器件均可提供 9 个不同的振荡器选项，这样用户在开发应用硬件时就有很大的选择范围。包括：

- 四个晶振模式，使用晶振或陶瓷谐振器。
- 两个外部时钟模式，可选择使用两个引脚（振荡器输入引脚和四分频时钟输出引脚）或一个引脚（振荡器输入引脚，四分频时钟输出引脚重新分配为通用 I/O 引脚）。
- 两个外部 RC 振荡器模式，与外部时钟模式选用相同的引脚。
- 一个内部振荡电路，提供一个 8 MHz 的时钟源和一个 INTRC 时钟源（大约 31 kHz，温度和 VDD 变化时频率保持稳定），并有 6 个时钟频率可供用户选择（从 125 kHz 到 4 MHz），总共 8 种时钟频率。

除了可将内部振荡电路用作时钟源，它还提供了一个稳定的参考源，为使该系列器件稳健地工作，增加了以下功能：

- **故障保护时钟监视：**该选项持续监测主时钟源，将其与内部振荡电路提供的参考信号作比较。如果发生了时钟故障，控制器切换到内部振荡电路，允许继续低速工作或安全地关闭应用。
- **双速起振：**该选项允许内部振荡器在上电复位时或从休眠模式唤醒时用作时钟源，直到主时钟源可用时为止。这样使得代码可在本来的时钟起振间隔期间执行，甚至允许一个应用在不返回全功率工作模式的情况下，执行常规的后台工作并返回休眠模式。

PIC18F2331/2431/4331/4431

1.2 其他特别功能

- **存储器耐久性:** 程序存储器和数据 EEPROM 的增强型闪存单元经评测, 可以耐受数万次擦写, 程序存储器高达 100,000 次, EEPROM 高达 1,000,000 次。不刷新的情况下, 保守地估计数据保存时间超过 100 年。
- **自编程能力:** 这些器件能在内部软件控制下写入各自的程序存储空间。通过使用受保护的引导块 (位于程序存储器的顶端) 中的引导加载程序, 可创建能在现场进行自我更新的应用。
- **功率控制 PWM 模块:** 在 PWM 模式下, 该模式提供 1、2 或 4 个调制输出端来控制半桥和全桥驱动器。其他功能包括检测到故障时自动关闭, 一旦条件清除后, 自动重新启动以重新激活输出端。
- **增强型 USART:** 该串行通讯模块可以通过使用内部振荡电路完成标准 RS-232 操作, 在与外部世界通讯中不需要外部晶振 (也就消除了随之所需的功耗)。该模块还包括自动波特率检测和 LIN 功能。
- **高速 10 位 A/D 转换器:** 该模块加入了可编程采集时间, 允许在选择一个通道、启动一个转换前不需要等待一个采样周期, 从而减少了代码开销。
- **运动反馈模块 (MFM):** 该模块的特点是拥有正交编码器接口 (QE1) 和输入捕捉 (IC) 模块。QE1 接收来自增量编码器的两个相位输入 (QEA, QEB) 和一个索引输入 (INDX)。QE1 支持高低精确位置记录, 方向状态和方向改变中断以及速度测量。该输入捕捉模块的特点是拥有 3 个独立的以 Timer5 为时基的输入捕捉通道、一个连接其他模块的特殊事件触发器和每一个 IC 输入通道都有可调节噪声滤波器。
- **扩展型看门狗定时器 (WDT):** 该增强型版本加入了一个 16 位预分频器, 允许超时范围从 4 ms 到 2 分钟以上, 在工作电压和温度变化时保持稳定。

PIC18F2331/2431/4331/4431

1.3 系列中各产品的具体情况

PIC18F2331/2431/4331/4431 系列中的器件有 28 引脚 (PIC18F2X31) 封装和 40 / 44 引脚 (PIC18F4X31) 封装。这两种封装的框图如图 1-1 所示。

各器件在三个方面存在差异：

1. 闪存程序存储器 (PIC18F2X31 器件为 8 KB, PIC18F4X31 器件为 16 KB)。
2. A/D 通道数 (PIC18F2X31 器件有 5 个, PIC18F4X31 器件有 9 个)。
3. I/O 端口 (PIC18F2X31 器件有 3 个双向端口, PIC18F4X31 器件有 5 个双向端口)。

该系列器件的其他特性都是相同的。这些特性汇总在表 1-1 中。

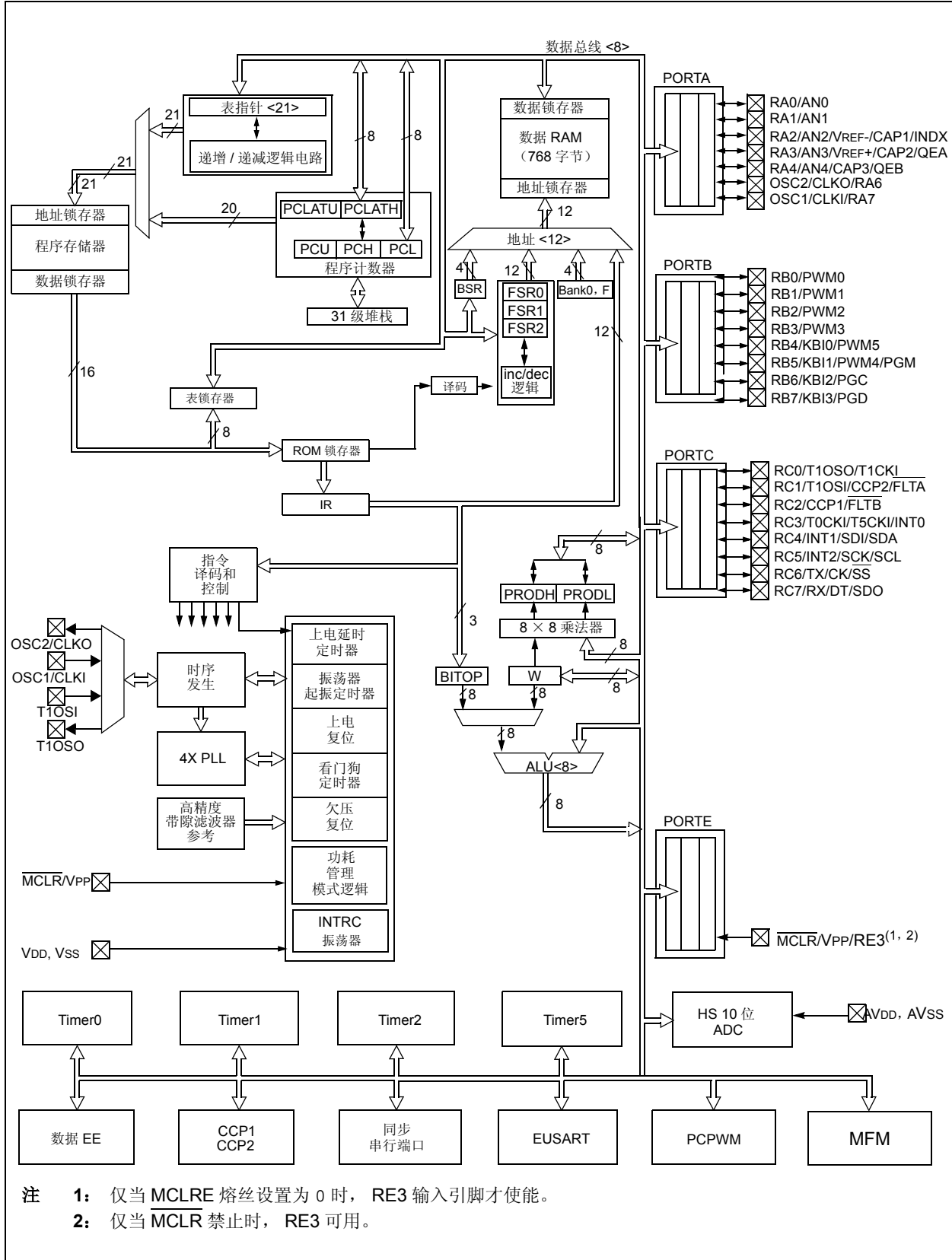
表 1-2 和表 1-3 列出了所有器件的引脚功能。

表 1-1: 器件特性

特性	PIC18F2331	PIC18F2431	PIC18F4331	PIC18F4431
工作频率	DC – 40 MHz	DC – 40 MHz	DC – 40 MHz	DC – 40 MHz
程序存储器 (字节)	8192	16384	8192	16384
程序存储器 (指令)	4096	8192	4096	8192
数据存储器 (字节)	768	768	768	768
数据 EEPROM 存储器 (字节)	256	256	256	256
中断源	22	22	34	34
I/O 端口	端口 A, B, C	端口 A, B, C	端口 A, B, C, D, E	端口 A, B, C, D, E
定时器	4	4	4	4
捕捉 / 比较 / PWM 模块	2	2	2	2
14 位功率控制 PWM	(6 通道)	(6 通道)	(8 通道)	(8 通道)
运动反馈模块 (输入捕捉 / 正交编码器接口)	1 个 QEI 或 3 个 IC	1 个 QEI 或 3 个 IC	1 个 QEI 或 3 个 IC	1 个 QEI 或 3 个 IC
串行通讯	SSP, 增强型 USART	SSP, 增强型 USART	SSP, 增强型 USART	SSP, 增强型 USART
10 位高速模数转换器模块	5 个输入通道	5 个输入通道	9 个输入通道	9 个输入通道
复位 (和延迟)	POR、BOR, RESET 指令, 堆栈满、 堆栈下溢 (PWRT, OST), MCLR (任选), WDT	POR、BOR, RESET 指令, 堆栈满、 堆栈下溢 (PWRT, OST), MCLR (任选), WDT	POR、BOR, RESET 指令, 堆栈满、 堆栈下溢 (PWRT, OST), MCLR (任选), WDT	POR、BOR, RESET 指令, 堆栈满、 堆栈下溢 (PWRT, OST), MCLR (任选), WDT
可编程低压检测	有	有	有	有
可编程欠压复位	有	有	有	有
指令集	75 条指令	75 条指令	75 条指令	75 条指令
封装	28 引脚 SDIP 28 引脚 SOIC	28 引脚 SDIP 28 引脚 SOIC	40 引脚 DIP 44 引脚 TQFP 44 引脚 QFN	40 引脚 DIP 44 引脚 TQFP 44 引脚 QFN

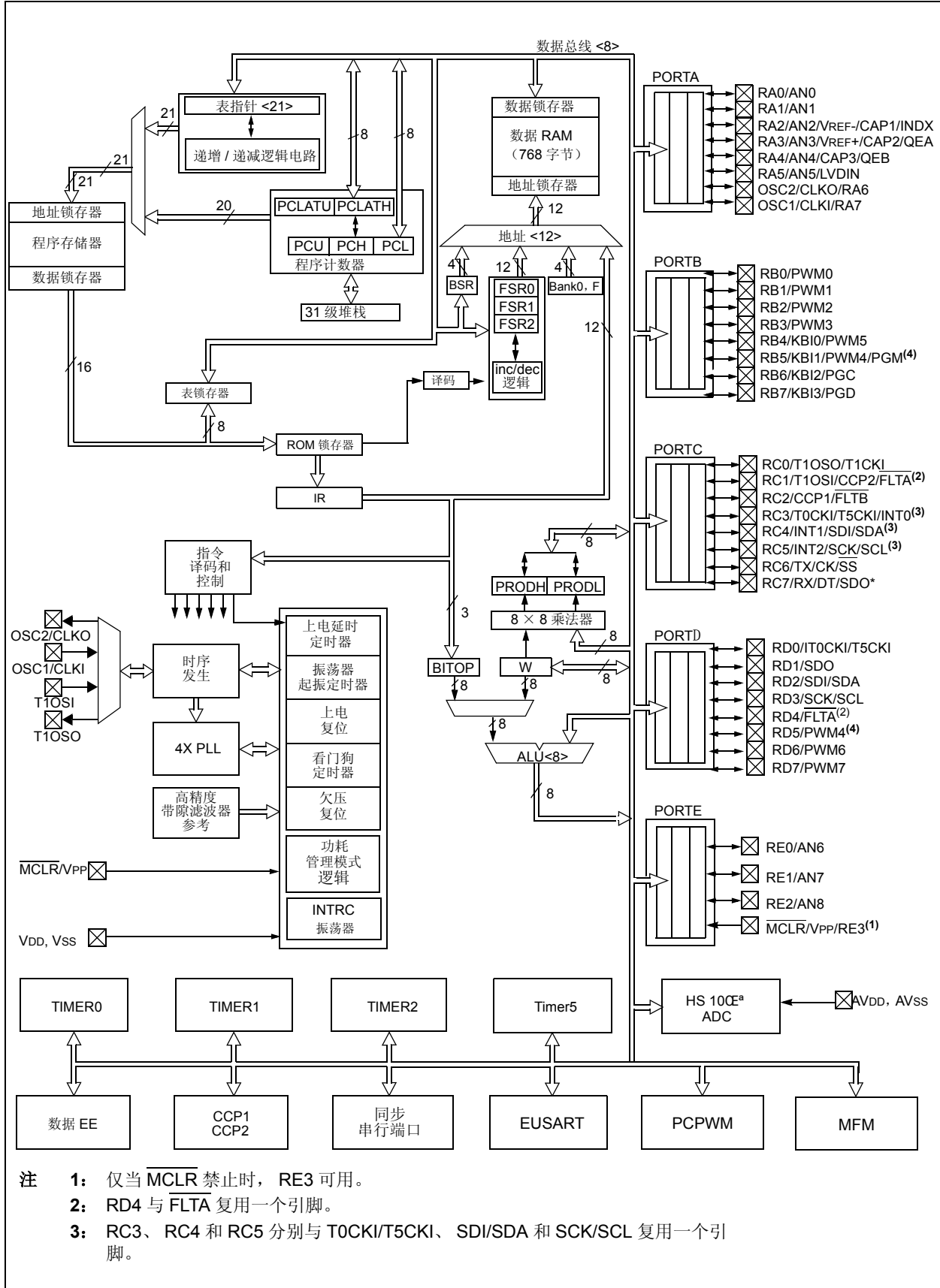
PIC18F2331/2431/4331/4431

图 1-1: PIC18F2331/2431 框图



PIC18F2331/2431/4331/4431

图 1-2: PIC18F4331/4431 框图



PIC18F2331/2431/4331/4431

表 1-2: PIC18F2331/2431 引脚 I/O 说明

引脚名称	引脚编号		引脚类型	缓冲器类型	描述
	DIP	SOIC			
MCLR/VPP/RE3 MCLR VPP RE3	1	1	I P I	ST ST	主清零（输入）或编程电压（输入）。 主清零（复位）输入。此引脚为低电平有效，使器件复位。 高压 ICSP 编程使能引脚。 数字输入。仅当 MCLR 禁止时可用。
OSC1/CLKI/RA7 OSC1 CLKI RA7	9	9	I I I/O	ST CMOS TTL	振荡器晶振或外部时钟输入。 振荡器晶振输入或外部时钟源输入。 配置为 RC 模式时，为 ST 缓冲器，否则为 CMOS 缓冲器。 外部时钟源输入。总是与 OSC1 引脚功能复用。（参见相关 OSC1/CLKI 和 OSC2/CLKO 引脚。） 通用 I/O 引脚。
OSC2/CLKO/RA6 OSC2 CLKO RA6	10	10	O O I/O	— — TTL	振荡器晶振或时钟输出。 振荡器晶振输出。在晶振模式下，连接晶振和谐振器。 在 RC 模式下，OSC2 引脚输出 CLKO 信号，该信号频率是 OSC1 的四分频，并且当作指令周期信号。 通用 I/O 引脚。
RA0/AN0 RA0 AN0 RA1/AN1 RA1 AN1 RA2/AN2/VREF-/CAP1/INDX RA2 AN2 VREF- CAP1 INDX RA3/AN3/VREF+/CAP2/QEA RA3 AN3 VREF+ CAP2 QEA RA4/AN4/CAP3/QEB RA4 AN4 CAP3 QEB	2 3 4 5 6	2 3 4 5 6	I/O I I/O I I/O I I I I/O I I I I/O I I I	TTL 模拟 TTL 模拟 TTL 模拟 模拟 ST ST TTL 模拟 模拟 ST ST TTL 模拟 ST ST	PORTA 是双向 I/O 端口。 数字 I/O。 模拟输入 0。 数字 I/O。 模拟输入 1。 数字 I/O。 模拟输入 2。 模拟输入 2。 A/D 参考电压（低）输入。 输入捕捉引脚 1。 正交编码器接口索引输入引脚。 数字 I/O。 模拟输入 3。 模拟输入 3。 A/D 参考电压（高）输入。 输入捕捉引脚 2。 正交编码器接口通道 A 输入引脚。 数字 I/O。 模拟输入 4。 模拟输入 4。 输入捕捉引脚 3。 正交编码器接口通道 B 输入引脚。

图注：

TTL = TTL 兼容输入	CMOS = CMOS 兼容输入或输出
ST = CMOS 电平的施密特触发器输入	I = 输入
O = 输出	P = 电源
OD = 漏极开路（无二极管连接到 VDD）	

PIC18F2331/2431/4331/4431

表 1-2: PIC18F2331/2431 引脚 I/O 说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	描述
	DIP	SOIC			
RB0/PWM0 RB0 PWM0	21	21	I/O O	TTL TTL	PORTB 是双向 I/O 端口。PORTB 所有输入端都有可编程的内部弱上拉。 数字 I/O。 PWM 输出 0。
RB1/PWM1 RB1 PWM1	22	22	I/O O	TTL TTL	数字 I/O。 PWM 输出 1。
RB2/PWM2 RB2 PWM2	23	23	I/O O	TTL TTL	数字 I/O。 PWM 输出 2。
RB3/PWM3 RB3 PWM3	24	24	I/O O	TTL TTL	数字 I/O。 PWM 输出 3。
RB4/KBI0/PWM5 RB4 KBI0 PWM5	25	25	I/O I O	TTL TTL TTL	数字 I/O。 电平变化中断引脚。 PWM 输出 5。
RB5/KBI1/PWM4/PGM RB5 KBI1 PWM4 PGM	26	26	I/O I O I/O	TTL TTL TTL ST	数字 I/O。 电平变化中断引脚。 PWM 输出 4。 低压 ICSP 编程入口引脚。
RB6/KBI2/PGC RB6 KBI2 PGC	27	27	I/O I I/O	TTL TTL ST	数字 I/O。 电平变化中断引脚。 在线调试器和 ICSP 编程时钟引脚。
RB7/KBI3/PGD RB7 KBI3 PGD	28	28	I/O I I/O	TTL TTL ST	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。

图注: TTL = TTL 兼容输入
 ST = CMOS 电平的施密特触发器输入
 O = 输出
 OD = 漏极开路 (无二极管连接到 VDD)

CMOS = CMOS 兼容输入或输出
 I = 输入
 P = 电源

PIC18F2331/2431/4331/4431

表 1-2: PIC18F2331/2431 引脚 I/O 说明 (续)

引脚名称	引脚编号		引脚类型	缓冲器类型	描述
	DIP	SOIC			
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	11	11	I/O O I	ST — ST	PORTC 是双向 I/O 端口。 数字 I/O。 Timer1 振荡器输出。 Timer1 外部时钟输入。
RC1/T1OSI/CCP2/FLTA RC1 T1OSI CCP2 FLTA	12	12	I/O I I/O I	ST CMOS ST ST	数字 I/O。 Timer1 振荡器输入。 Capture2 输入、Compare2 输出、PWM2 输出。 故障中断输入引脚。
RC2/CCP1/FLTB RC2 CCP1 FLTB	13	13	I/O I/O I	ST ST ST	数字 I/O。 Capture1 输入 / Compare1 输出 / PWM1 输出。 故障中断输入引脚，
RC3/T0CKI/T5CKI/INT0 RC3 T0CKI T5CKI INT0	14	14	I/O I I I	ST ST ST ST	数字 I/O。 Timer0 备用时钟输入。 Timer5 备用时钟输入。 外部中断 0。
RC4/INT1/SDI/SDA RC4 INT1 SDI SDA	15	15	I/O I I I/O	ST ST ST ST	数字 I/O。 外部中断 1。 SPI™ 数据输入。 I²C™ 数据 I/O。
RC5/INT2/SCK/SCL RC5 INT2 SCK SCL	16	16	I/O I I/O I/O	ST ST ST ST	数字 I/O。 外部中断 2。 SPI 模式的同步串行时钟输入 / 输出。 I²C 模式的同步串行时钟输入 / 输出。
RC6/TX/CK/SS RC6 TX CK SS	17	17	I/O O I/O I	ST - ST TTL	数字 I/O。 USART 异步发送。 USART 同步时钟 (参见相关 RX/DT)。 SPI 从动选择输入。
RC7/RX/DT/SDO RC7 RX DT SDO	18	18	I/O I I/O O	ST ST ST -	数字 I/O。 USART 异步接收。 USART 同步数据 (参见相关 TX/CK)。 SPI 数据输出。
VSS	8, 19	8, 19	P	-	逻辑和 I/O 引脚的接地参考点。
VDD	7, 20	7, 20	P	-	逻辑和 I/O 引脚的正供电电源。

图注: TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
O = 输出
OD = 漏极开路 (无二极管连接到 VDD)

CMOS = CMOS 兼容输入或输出
I = 输入
P = 电源

PIC18F2331/2431/4331/4431

表 1-3: PIC18F4331/4431 引脚 I/O 说明

引脚名称	引脚编号			引脚类型	缓冲类型	描述
	DIP	TQFP	QFN			
MCLR/VPP/RE3 MCLR VPP RE3	1	18	18	I P I	ST ST	主清零（输入）或编程电压（输入）。 主清零（复位）输入。此引脚为低电平有效，使器件复位。 编程电压输入。 数字输入。仅当 MCLR 禁止时可用。
OSC1/CLKI/RA7 OSC1 CLKI RA7	13	30	32	I I I/O	ST CMOS TTL	振荡器晶振或外部时钟输入。 振荡器晶振输入或外部时钟源输入。 在 RC 模式下为 ST 缓冲器，否则为 CMOS 缓冲器。 外部时钟源输入。总是与 OSC1 引脚功能复用。（参见相关 OSC1/CLKI 和 OSC2/CLKO 引脚。） 通用 I/O 引脚。
OSC2/CLKO/RA6 OSC2 CLKO RA6	14	31	33	O O I/O	— - TTL	振荡器晶振或时钟输出。 振荡器晶振输出。在晶振模式下，连接晶振或谐振器。 在 RC 模式下，OSC2 引脚输出 CLKO 信号，该信号频率是 OSC1 的四分频，并且当作指令周期信号。 通用 I/O 引脚。
RA0/AN0 RA0 AN0 RA1/AN1 RA1 AN1 RA2/AN2/VREF-/CAP1/ INDX RA2 AN2 VREF- CAP1 INDX RA3/AN3/VREF+/ CAP2/QEA RA3 AN3 VREF+ CAP2 QEA RA4/AN4/CAP3/QEB RA4 AN4 CAP3 QEB RA5/AN5/LVDIN RA5 AN5 LVDIN	2 3 4 5 6 7	19 20 21 22 23 24	19 20 21 22 23 24	I/O I I/O I I/O I I I I I/O I I I I I/O I I I I/O I I	TTL 模拟 TTL 模拟 TTL 模拟 模拟 ST ST TTL 模拟 模拟 ST ST TTL 模拟 ST ST TTL 模拟 ST ST TTL 模拟 模拟	PORTA 是双向 I/O 端口。 数字 I/O。 模拟输入 0。 数字 I/O。 模拟输入 1。 数字 I/O。 模拟输入 2。 A/D 参考电压（低）输入。 输入捕捉引脚 1。 正交编码器接口索引输入引脚。 数字 I/O。 模拟输入 3。 A/D 参考电压（高）输入。 输入捕捉引脚 2。 正交编码器接口通道 A 输入引脚。 数字 I/O。 模拟输入 4。 输入捕捉引脚 3。 正交编码器接口通道 B 输入引脚。 数字 I/O。 模拟输入 5。 低压检测输入。

图注: TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
O = 输出
OD = 漏极开路（无二极管连接到 VDD）

CMOS = CMOS 兼容输入或输出
I = 输入
P = 电源

PIC18F2331/2431/4331/4431

表 1-3: PIC18F4331/4431 引脚 I/O 说明 (续)

引脚名称	引脚编号			引脚类型	缓冲类型	描述
	DIP	TQFP	QFN			
RB0/PWM0 RB0 PWM0	33	8	9	I/O O	TTL TTL	PORTB 是双向 I/O 端口。PORTB 所有输入端都有可编程的内部弱上拉。 数字 I/O。 PWM 输出 0。
RB1/PWM1 RB1 PWM1	34	9	10	I/O O	TTL TTL	数字 I/O。 PWM 输出 1。
RB2/PWM2 RB2 PWM2	35	10	11	I/O O	TTL TTL	数字 I/O。 PWM 输出 2。
RB3/PWM3 RB3 PWM3	36	11	12	I/O O	TTL TTL	数字 I/O。 PWM 输出 3。
RB4/KBI0/PWM5 RB4 KBI0 PWM5	37	14	14	I/O I O	TTL TTL TTL	数字 I/O。 电平变化中断引脚。 PWM 输出 5。
RB5/KBI1/PWM4/ PGM RB5 KBI1 PWM4 PGM	38	15	15	I/O I O I/O	TTL TTL TTL ST	数字 I/O。 电平变化中断引脚。 PWM 输出 4。 低压 ICSP 编程入口引脚。
RB6/KBI2/PGC RB6 KBI2 PGC	39	16	16	I/O I I/O	TTL TTL ST	数字 I/O。 电平变化中断引脚。 在线调试器和 ICSP 编程时钟引脚。
RB7/KBI3/PGD RB7 KBI3 PGD	40	17	17	I/O I I/O	TTL TTL ST	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。

图注: TTL = TTL 兼容输入
 ST = CMOS 电平的施密特触发器输入
 O = 输出
 OD = 漏极开路 (无二极管连接到 VDD)

CMOS = CMOS 兼容输入或输出
 I = 输入
 P = 电源

PIC18F2331/2431/4331/4431

表 1-3: PIC18F4331/4431 引脚 I/O 说明 (续)

引脚名称	引脚编号			引脚类型	缓冲类型	描述
	DIP	TQFP	QFN			
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	15	32	34	I/O O I	ST - ST	PORTC 是双向 I/O 端口。 数字 I/O。 Timer1 振荡器输出。 Timer1 外部时钟输入。
RC1/T1OSI/CCP2/ FLTA RC1 T1OSI CCP2 FLTA	16	35	35	I/O I I/O I	ST CMOS ST ST	数字 I/O。 Timer1 振荡器输入。 Capture2 输入、Compare2 输出、PWM2 输出。 故障中断输入引脚。
RC2/CCP1/FLTB RC2 CCP1 FLTB	17	36	36	I/O I/O I	ST ST ST	数字 I/O。 Capture1 输入 / Compare1 输出 / PWM1 输出。 故障中断输入引脚。
RC3/T0CKI/T5CKI/ INT0 RC3 T0CKI T5CKI INT0	18	37	37	I/O I I I	ST ST ST ST	数字 I/O。 Timer0 备用时钟输入。 Timer5 备用时钟输入。 外部中断 0。
RC4/INT1/SDI/SDA RC4 INT1 SDI SDA	23	42	42	I/O I I I/O	ST ST ST ST	数字 I/O。 外部中断 1。 SPI 数据输入。 I ² C 数据 I/O。
RC5/INT2/SCK/SCL RC5 INT2 SCK SCL	24	43	43	I/O I I/O I/O	ST ST ST ST	数字 I/O。 外部中断 2。 SPI 模式的同步串行时钟输入 / 输出。 I ² C 模式的同步串行时钟输入 / 输出。
RC6/TX/CK/SS RC6 TX CK SS	25	44	44	I/O O I/O I	ST - ST ST	数字 I/O。 USART 异步发送。 USART 同步时钟 (参见相关 RX/DT)。 SPI 从动选择输入。
RC7/RX/DT/SDO RC7 RX DT SDO	26	1	1	I/O I I/O O	ST ST ST -	数字 I/O。 USART 异步接收。 USART 同步数据 (参见相关 TX/CK)。 SPI 数据输出。

图注: TTL = TTL 兼容输入
ST = CMOS 电平的施密特触发器输入
O = 输出
OD = 漏极开路 (无二极管连接到 VDD)

CMOS = CMOS 兼容输入或输出
I = 输入
P = 电源

PIC18F2331/2431/4331/4431

表 1-3: PIC18F4331/4431 引脚 I/O 说明 (续)

引脚名称	引脚编号			引脚类型	缓冲类型	描述
	DIP	TQFP	QFN			
RD0/T0CKI/T5CKI RD0 T0CKI T5CKI	19	38	38	I/O I I	ST ST ST	PORTD 是双向 I/O 端口，或与微处理器端口连接的并行从动端口 (PSP)。当使能 PSP 模块时，这些引脚具有 TTL 输入缓冲器。 数字 I/O。 Timer0 外部时钟源输入。 Timer5 输入时钟。
RD1/SDO RD1 SDO	20	39	39	I/O O	ST -	数字 I/O。 SPI 数据输出。
RD2/SDI/SDA RD2 SDI SDA	21	40	40	I/O I I/O	ST ST ST	数字 I/O。 SPI 数据输入。 I ² C 数据 I/O。
RD3/SCK/SCL RD3 SCK SCL	22	41	41	I/O I/O I/O	ST ST ST	数字 I/O。 SPI 模式的同步串行时钟输入 / 输出。 I ² C 模式的同步串行时钟输入 / 输出。
RD4/FLTA RD4 FLTA	27	2	2	I/O I	ST ST	数字 I/O。 错误中断输入引脚。
RD5/PWM4 RD5 PWM4	28	3	3	I/O O	ST TTL	数字 I/O。 PWM 输出 4。
RD6/PWM6 RD6 PWM6	29	4	4	I/O O	ST TTL	数字 I/O。 PWM 输出 6。
RD7/PWM7 RD7 PWM7	30	5	5	I/O O	ST TTL	数字 I/O。 PWM 输出 7。

图注: TTL = TTL 兼容输入
 ST = CMOS 电平的施密特触发器输入
 O = 输出
 OD = 漏极开路 (无二极管连接到 VDD)

CMOS = CMOS 兼容输入或输出
 I = 输入
 P = 电源

PIC18F2331/2431/4331/4431

注:

PIC18F2331/2431/4331/4431

2.0 振荡器配置

2.1 振荡器类型

PIC18F2331/2431/4331/4431 器件可以在 10 种不同的振荡模式下工作。通过编程配置寄存器 1H 中的配置位 Fosc3:Fosc0，用户可以对 10 种模式进行选择。

1. LP 低功耗晶振模式
2. XT 晶振 / 谐振器模式
3. HS 高速晶振 / 谐振器模式
4. HSPLL 使能 PLL 的高速晶振 / 谐振器模式
5. RC 在 RA6 上输出 Fosc/4 的外部电阻 / 电容模式
6. RCIO RA6 为 I/O 的外部电阻 / 电容模式
7. INTIO1 内部振荡模式，RA6 为 Fosc/4 输出，RA7 为 I/O
8. INTIO2 内部振荡模式，I/O 为 RA6 和 RA7
9. EC 带 Fosc/4 输出的外部时钟模式
10. ECIO RA6 为 I/O 的外部时钟模式

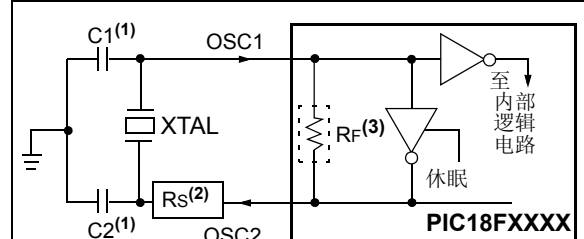
2.2 晶体振荡器 / 陶瓷谐振器模式

在 XT、LP、HS 或 HSPLL 振荡器模式下，晶振或陶瓷谐振器与 OSC1 和 OSC2 引脚连接，以产生振荡。图 2-1 说明了引脚连接的情况。

振荡器的设计要求使用平行切割的晶体。

注： 使用顺序切割的晶体，可能会使振荡器产生的频率超出晶振制造厂商的规范。

图 2-1: 晶振 / 陶瓷谐振器工作原理 (XT、LP、HS 或 HSPLL 配置)



- 注**
- 1: 如需了解 C1 和 C2 的初始值，请参见表 2-1 和表 2-2。
 - 2: AT 条形切割的晶体可能会需要一个串联电阻 (Rs)。
 - 3: Rf 的值根据振荡模式而定。

表 2-1: 陶瓷谐振器的电容选择

使用的典型电容值			
模式	频率	OSC1	OSC2
XT	455 kHz	56 pF	56 pF
	2.0 MHz	47 pF	47 pF
	4.0 MHz	33 pF	33 pF
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。
 这些电容搭配下列谐振器时的基本起振和工作情况，已通过测试。**这些值并非最佳值。**
 要获得合适的振荡器工作状态，可能需要不同的电容值。用户应当测试振荡器在应用的预期 VDD 和温度下的性能。
 更多信息，请参见 22 的注释。

使用的谐振器	
455 kHz	4.0 MHz
2.0 MHz	8.0 MHz
16.0 MHz	

PIC18F2331/2431/4331/4431

表 2-2: 晶体振荡器的电容选择

振荡器类型	晶振频率	已测试的典型电容值:	
		C1	C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	1 MHz	33 pF	33 pF
	4 MHz	27 pF	27 pF
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

上述电容值仅供设计参考。

这些电容搭配下列谐振器时的基本起振和工作情况，已通过测试。**这些值并非最佳值。**

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

更多信息，请参见本表后的注释。

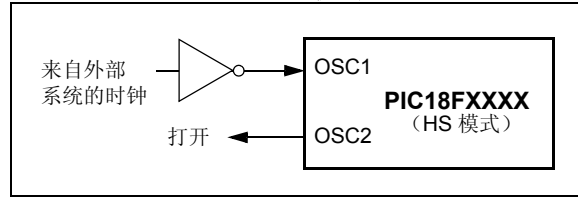
使用的谐振器

32 kHz	4 MHz
200 kHz	8 MHz
1 MHz	20 MHz

- 注**
- 1: 选择较高的电容值可以提高振荡器的稳定性，但同时也延长了起振时间。
 - 2: 当工作于 3V 以下 VDD，或在任何电压下使用某些陶瓷谐振器时，可能需要使用 HS 模式或切换到晶体振荡器。
 - 3: 因为每种谐振器 / 晶振都有其自身特性，用户应当向谐振器 / 晶振制造厂商咨询外部元件的正确值。
 - 4: 为避免对低驱动规格的晶振造成过驱动，可能会需要使用电阻 Rs。
 - 5: 请一定验证在应用中的预期 VDD 和温度范围内的振荡器性能。

如图 2-2 所示，在 HS 模式下，OSC1 引脚也可以连接外部时钟源。

图 2-2: 外部时钟输入工作原理 (HS 振荡器配置)



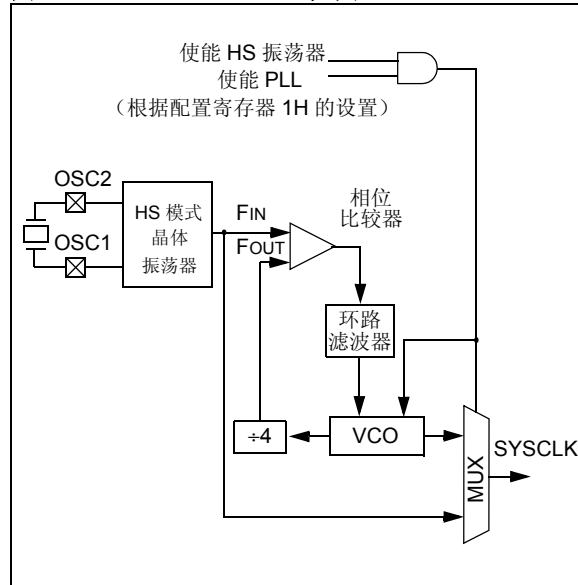
2.3 HSPLL

如果用户希望使用低频晶振电路或通过晶体振荡器将器件频率调至其最高额定频率，可以选择使用锁相环 (Phase Locked Loop, PLL) 电路。对于关心高频晶振产生 EMI 的用户，这样做非常有用。

在 10 MHz 以下频率时，HSPLL 模式利用了 HS 模式振荡器。然后 PLL 将振荡器输出频率乘以 4，从而可以产生最高 40 MHz 的内部时钟频率。

仅当振荡器配置位编程为 HSPLL 模式时，PLL 模式才会使能。如果编程为其他模式，则 PLL 模式禁止。

图 2-3: PLL 框图

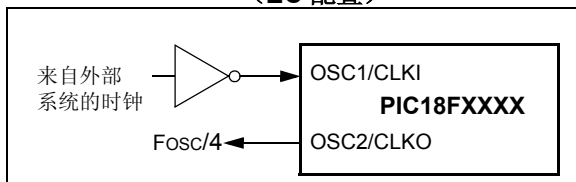


2.4 外部时钟输入

EC 和 ECIO 振荡器模式需要 OSC1 引脚连接一个外部时钟源。在上电复位后或从休眠模式退出后，不需要振荡器起振时间。

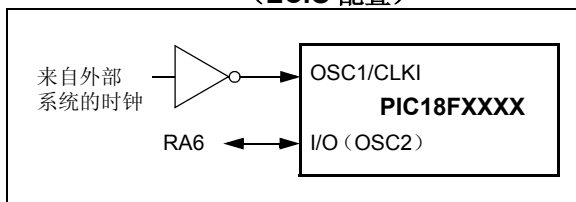
在 EC 振荡器模式下，振荡器频率的 4 分频信号可由 OSC2 引脚输出。此信号可用于测试或同步其他逻辑单元。图 2-4 给出了 EC 振荡器模式的引脚连接。

图 2-4: 外部时钟输入工作原理 (EC 配置)



ECIO 振荡器模式的功能类似于 EC 模式，不同之处在于 OSC2 引脚变成了一个附加的通用 I/O 引脚。I/O 引脚变成 PORTA 的第 6 位 (RA6)。图 2-5 给出了 ECIO 振荡器模式下的引脚连接。

图 2-5: 外部时钟输入工作原理 (ECIO 配置)

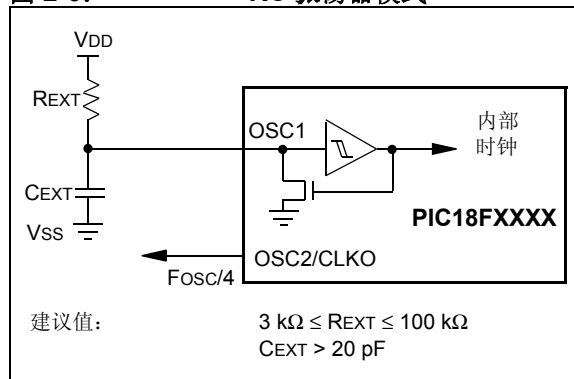


2.5 RC 振荡器

对于定时要求不高的应用，RC 和 RCIO 器件选项能更好地节约成本。RC 振荡器频率与电源电压、电阻 (REXT) 和电容 (CEXT) 值及工作温度密切相关。另外，由于正常制造工艺的变动因素，每个器件的振荡频率也会有所不同。而不同封装类型的引线电容间的差异也会影响振荡频率，特别是 CEXT 值较小时。用户还需要考虑由于所使用外部 R 和 C 元件的公差造成的不同。图 2-6 给出了 R/C 组合的连接方式。

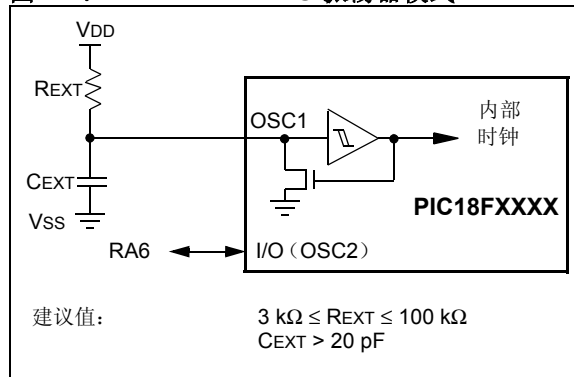
在 RC 振荡器模式下，振荡器频率的 4 分频信号可由 OSC2 引脚输出。此信号可以用于测试目的或同步其他逻辑单元。

图 2-6: RC 振荡器模式



RCIO 振荡器模式 (图 2-7) 的功能类似于 RC 模式，不同之处在于 OSC2 引脚变成了一个附加的通用 I/O 引脚。I/O 引脚变成 PORTA 的第 6 位 (RA6)。

图 2-7: RCIO 振荡器模式



PIC18F2331/2431/4331/4431

2.6 内部振荡电路

PIC18F2331/2431/4331/4431 器件包含一个内部振荡电路，可以产生两个不同的时钟信号，两者都可以充当系统的时钟源。可以避免使用 OSC1 和 / 或 OSC2 引脚构成外部振荡电路。

主输出 (INTOSC) 是一个 8 MHz 的时钟源，可以用于直接驱动系统时钟。还可以驱动后分频器，提供从 125 kHz 到 4 MHz 间的时钟频率。当选择了 125 kHz 到 8 MHz 的系统时钟频率时，将使能 INTOSC 输出。

另一个时钟源是内部 RC 振荡器 (INTRC)，提供 31 kHz 的输出。通过选择内部振荡电路作为系统时钟源，或使能以下任何组件，都可以使能 INTRC 振荡器：

- 上电延时定时器
- 故障保护时钟监视器
- 看门狗定时器
- 双速起振

在第 22.0 节“CPU 的特殊功能”中详细讨论了这些功能。

通过配置 OSCCON 寄存器（见寄存器 2-2）的 IRCF 位，可以选择时钟源频率（INTOSC 直接输出、INTRC 直接输出或 INTOSC 后分频输出）。

2.6.1 INTIO 模式

使用内部振荡器作为时钟源，最多可以避免使用两个外部振荡器引脚，留出的引脚可用于数字 I/O。目前有两种不同的配置。

- 在 INTIO1 模式，OSC2 引脚输出 $F_{osc}/4$ ，而 OSC1 引脚则充当 RA7，用于数字输入 / 输出。
- 在 INTIO2 模式，OSC1 充当 RA7，OSC2 充当 RA6，两者都用于数字输入 / 输出。

2.6.2 INTRC 输出频率

为了产生 8.0 MHz 的 INTOSC 输出频率，在工厂中对内部振荡电路校准。这就改变了 INTRC 源的标称频率 31.25 kHz。依靠 INTRC 源的外设和功能会受到此频率变化的影响。

2.6.3 OSCTUNE 寄存器

内部振荡器的输出虽然已在工厂经过校准，但用户可以在使用时进行调整。可通过写 OSCTUNE 寄存器（寄存器 2-1）实现。调节灵敏度在整个调节范围内保持不变。

当修改 OSCTUNE 寄存器后，INTOSC 和 INTRC 的频率将开始调节至新的频率。INTRC 时钟将在 8 个时钟周期（大约 $8 * 32 \mu s = 256 \mu s$ ）内达到新的频率。INTOSC 时钟会在 1 ms 内稳定下来。在频率变化期间，代码会继续执行。不会有任何迹象表明发生了频率变化。但依靠 INTRC 时钟源频率的功能（例如 WDT、故障保护时钟监视器和外设等），其运行会受到频率变化的影响。

PIC18F2331/2431/4331/4431

寄存器 2-1:

OSCTUNE: 振荡器调节寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	
bit 7								bit 0

bit 7, 6

未实现位: 读作 0

bit 5-0

TUN<5:0>: 频率调节位

011111 =最高频率

·

·

000001

000000 =中心频率。振荡器模块运行在校准后的频率上。

111111

·

·

100000 =最低频率

图注:

R= 可读位

W= 可写位

U= 未实现位, 读作 0

-n = 上电复位时的值

1 = 置位

0= 清零

x= 未知

PIC18F2331/2431/4331/4431

2.7 时钟源与振荡器切换

与以前的 PIC18 器件类似，PIC18F2331/2431/4331/4431 器件包括了一项功能，可以将时钟源从主振荡器切换为备用低频时钟源。PIC18F2331/2431/4331/4431 器件提供了两个备用时钟源。使能这些备用时钟源后，可切换到各种功耗管理工作模式提供更多的选择。

基本上，这些器件都有 3 种时钟源：

- 主振荡器
- 辅助振荡器
- 内部振荡电路

主振荡器包括外部晶振和谐振器模式、外部 RC 模式、外部时钟模式和内部振荡电路。具体模式在上电复位时由配置寄存器 1H 的内容决定。这些模式的具体情况已在本章的前半部分作了介绍。

辅助振荡器是不与 OSC1 或 OSC2 引脚连接的外部时钟源。这些时钟源在控制器处于功耗管理模式时仍然可以继续工作。

PIC18F2331/2431/4331/4431 器件只支持使用 Timer1 振荡器作为辅助振荡器。Timer1 振荡器（在所有的功耗管理模式）通常是实时时钟等功能的时基。

大部分情况下，在 RC0/T1OSO 和 RC1/T1OSI 之间接有一个 32.768 kHz 的时钟晶振。与 LP 模式振荡电路类似，每个引脚都通过负载电容接地。

在第 12.2 节“Timer1 振荡器”中对 Timer1 振荡器进行了详细讨论。

除了作为主时钟源，**内部振荡电路**还可作为功耗管理模式的时钟源。INTRC 源也可作为几种特别功能（如 WDT 和故障保护时钟监视器）的时钟源。

在图 2-8 中给出了 PIC18F2331/2431/4331/4431 器件的时钟源。如需了解 Timer1 振荡器的更多信息，请参阅第 12.0 节“Timer1 模块”。如需了解配置寄存器的详细信息，请参阅第 22.1 节“配置位”。

2.7.1 振荡器控制寄存器

OSCCON 寄存器（寄存器 2-2）控制全功率模式和功耗管理模式下系统时钟多方面的工作方式。

系统时钟选择位（SCS1:SCS0），选择器件在功耗管理模式下运行时所使用的时钟源。可用的时钟源包括主时钟（在配置寄存器 1H 中定义）、辅助时钟（Timer1 振荡器）和内部振荡电路。时钟选择在执行 SLEEP 指令，器件进入功耗管理工作模式后才会生效。任何形式的复位都会使 SCS 位清零。

内部振荡器选择位（IRCF2:IRCF0）用来指定驱动系统时钟的内部振荡电路的输出频率。包括 INTRC 频率、INTOSC（8 MHz）频率或 INTOSC 后分频器产生的 6 个频率之一（125 kHz 到 4 MHz）。如果系统时钟源是内部振荡电路，改变这些位的状态时，内部振荡器的输出频率会立即发生改变。

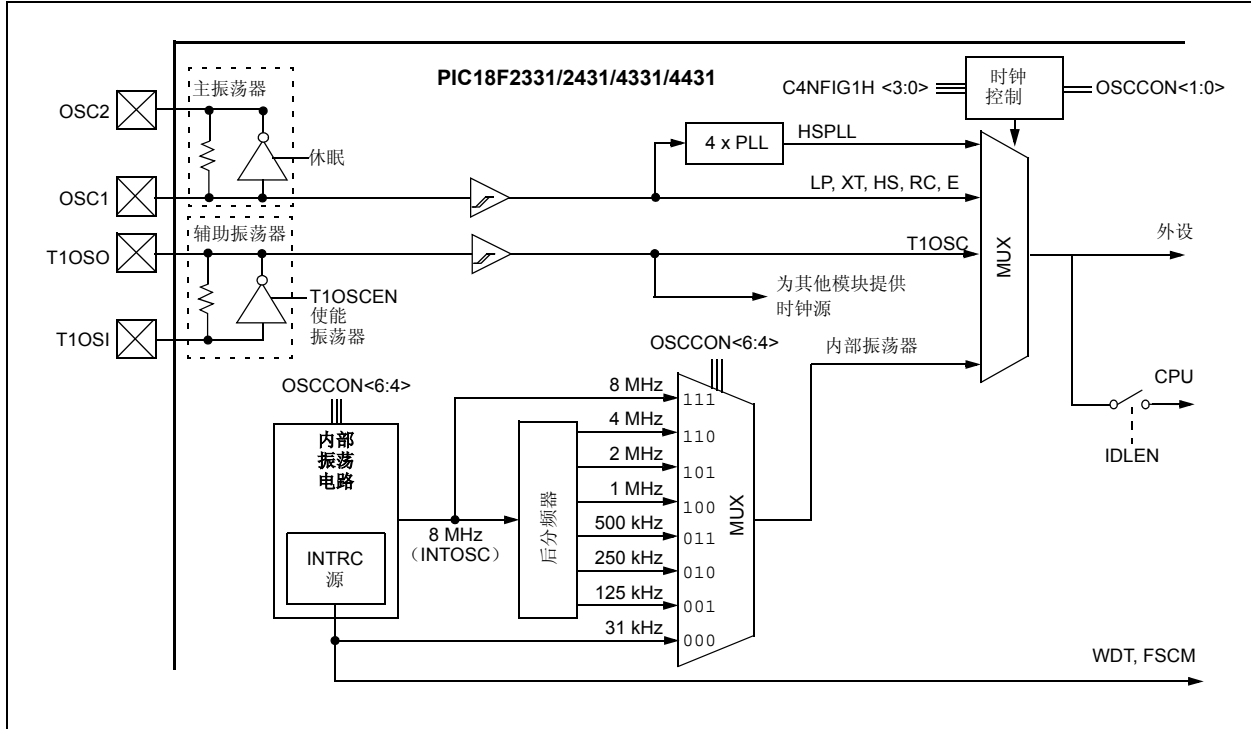
OSTS、IOFS 和 T1RUN 位表明哪个时钟源是当前的系统时钟源。OSTS 表明振荡器起振定时器已超时，主时钟是主时钟模式下的系统时钟源。IOFS 位表明内部振荡电路已稳定，并在 RC 时钟模式下提供系统时钟。T1RUN 位（T1CON<6>）表明何时 Timer1 振荡器在辅助时钟模式下，提供系统时钟。在功耗管理模式下，任何时候这 3 个位都只有一个被置位。如果这些位都没有置位，则当前系统时钟源是 INTRC，或内部振荡电路刚刚起振尚未稳定。

IDLEN 位控制功耗管理模式下控制器 CPU 的选择性关断。在第 3.0 节“功耗管理模式”中详细讨论了这些位的用法。

- 注 1:** 要选择辅助时钟源，必须使能 Timer1 振荡器。通过将 Timer1 控制寄存器中的 T1OSCEN 位（T1CON<3>）置位，可以使能 Timer1 振荡器。如果未使能 Timer1 振荡器，则在执行 SLEEP 指令期间任何选择辅助时钟源的尝试都会被忽略。
- 2:** 建议在 Timer1 振荡器进入稳定工作状态后，执行 SLEEP 指令。否则当 Timer1 振荡器起振时，可能会有很长的延迟。

PIC18F2331/2431/4331/4431

图 2-8: PIC18F2331/2431/4331/4431 框图



PIC18F2331/2431/4331/4431

寄存器 2-2:

OSCCON 寄存器

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

- bit 7 **IDLEN:** 空闲使能位
1 = 使能空闲模式, CPU 内核未处于功耗管理模式
0 = 使能运行模式, CPU 内核处于功耗管理模式
- bit 6-4 **IRCF2:IRCF0:** 内部振荡器频率选择位
111 = 8 MHz (8 MHz 源直接驱动时钟)
110 = 4 MHz
101 = 2 MHz
100 = 1 MHz
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31 kHz (INTRC 源直接驱动时钟)
- bit 3 **OSTS:** 振荡器起振超时状态位
1 = 振荡器起振超时定时器已到期, 主振荡器正在运行
0 = 振荡器起振超时定时器正在运行, 主振荡器尚未就绪
- bit 2 **IOFS:** INTOSC 频率稳定位
1 = INTOSC 频率已稳定
0 = INTOSC 频率尚未稳定
- bit 1-0 **SCS1:SCS0:** 系统时钟选择位
1x = 内部振荡电路 (RC 模式)
01 = Timer1 振荡器 (间接模式)
00 = 主振荡器 (休眠与 PRI_IDLE 模式)
- 注 1: 由配置寄存器 1H 中的 IESO 位的状态决定。

图注:

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

2.7.2 振荡器转换

PIC18F2331/2431/4331/4431 器件包含有防止切换时钟源时产生脉冲干扰的电路。在时钟切换过程中, 系统时钟会有短暂停顿。停顿时间为新时钟源的 8 到 9 个时钟周期。这就保证了新时钟源的稳定性, 且脉冲宽度不小于这两个时钟源最窄的脉冲宽度。

在第 3.1.2 节“进入功耗管理模式”中详细讨论了时钟转换。

2.8 功耗管理模式对各种时钟源的影响

当器件执行 SLEEP 指令时，系统会切换到某个功耗管理模式，这由 OSCCON 寄存器的 IDLEN 位和 SCS1:SCS0 位的状态决定。如需详细信息，参见第 3.0 节“功耗管理模式”。

当选择了 PRI_IDLE 模式时，指定的主振荡器继续运行，而不中断。对于所有其他功耗管理模式，使用 OSC1 的振荡器会被禁止。OSC1 引脚（若振荡器使用 OSC2 引脚，则还有 OSC2 引脚）将会停止振荡。

在辅助时钟模式（SEC_RUN 和 SEC_IDLE）下，Timer1 振荡器作为系统时钟源。如需要使用 Timer1 的时钟，Timer1 振荡器也可在所有功耗管理模式下运行。

在内部振荡模式（RC_RUN 和 RC_IDLE）下，由内部振荡电路提供系统时钟源。无论是否为功耗管理模式，INTRC 输出可以直接用作系统时钟，都可以被使能支持几种特殊功能（见第 22.2 节到第 22.4 节）。8 MHz 的 INTOSC 输出可以直接为系统提供时钟源，或先分频。如果直接由 INTRC 输出作为系统时钟源，则会禁止 INTOSC 输出。

如果选择了休眠模式，所有的时钟源都会被停止。因为关闭了所有的晶体管切换电流，休眠模式能实现最小的器件电流消耗（仅泄漏电流）。

如果使能任何在休眠期间仍可运行的片上功能，将会增加休眠时的电流消耗。要支持 WDT 工作，需要使能 INTRC。要支持实时时钟，可能需要 Timer1 振荡器工作。不需要系统时钟源的其他功能也可以工作（即 SSP 从动器件、PSP、INTn 引脚和 A/D 转换等）。

2.9 上电延迟

有两个定时器控制上电延迟，这样大部分应用都无需外接复位电路实现上电延迟，就可以确保在正常环境下，器件的供电电源达到稳定且主时钟也稳定运行前，器件保持复位状态。如需了解更多上电延迟的信息，请参阅第 4.1 节至第 4.5 节。

第一个定时器是上电延时定时器（PWRT），如果它在配置寄存器 2L 使能，则在上电时提供一段固定延迟（表 25-8 中的参数 33）。第二个定时器是振荡器起振定时器（OST），用于在晶体振荡器稳定前（LP、XT 和 HS 模式），保持单片机在复位状态。在振荡器为器件提供时钟源之前，OST 通过等候 1024 个振荡周期来实现此延迟。

当选择 HSPLL 振荡器模式时，器件还会多保持复位状态 2 ms，之后是 HS 模式的 OST 延迟，这样 PLL 可以锁定到输入时钟频率。

当控制器准备好执行指令时，在上电复位之后有 5 到 10 μs 的延迟。此延迟与其他延迟同时进行。任何 EC、RC 或 INTIO 模式用作主时钟源时，这可能是唯一的延迟。

表 2-3: 休眠模式下 OSC1 和 OSC2 引脚的状态

振荡器模式	OSC1 引脚	OSC2 引脚
RC, INTIO1	悬空，经外部电阻上拉为高电平	处于逻辑低电平（频率的 1/4）
RCIO, INTIO2	悬空，经外部电阻上拉为高电平	配置为 PORTA 的 bit 6
ECIO	悬空，经外部时钟上拉为高电平	配置为 PORTA 的 bit 6
EC	悬空，经外部时钟上拉为高电平	处于逻辑低电平（频率的 1/4）
LP、XT 和 HS	反馈反相器被禁止，处于静态电平	反馈反相器被禁止，处于静态电平

注：有关休眠和 MCLR 复位造成的超时的信息，请参见第 4.0 节“复位”的表 4-1。

PIC18F2331/2431/4331/4431

注:

3.0 功耗管理模式

为更有效的功耗管理，PIC18F2331/2431/4331/4431 器件提供总共六种运行模式（见表 3-1）。这些运行模式为资源受限的应用（即，电池供电的器件）提供了多种可选择的节能模式。

有三类功耗管理模式：

- 休眠模式
- 空闲模式
- 运行模式

这些类别定义了器件的哪些部分作为时钟源，有时还可以定义时钟频率。运行和空闲模式可以使用三种有效时钟源中（主、间接或 INTOSC 复用器）的任意一种；而休眠模式则不能使用时钟源。

其他 PIC18 器件具有的时钟源切换功能（即，在主振荡器中使用 Timer1 振荡器）和所有 PICmicro® 器件均有的休眠模式（此时所有系统时钟都停止），在 PIC18F2331/2431/4331/4431 器件中都有（可分别运行 SEC_RUN 和休眠模式）。但是，新增的功耗管理模式可以更灵活地让用户决定器件的哪些部分在运行。功耗管理模式是事件驱动的，也就是说，器件必须发生某些特殊事件来进入或（更加特别）退出这些操作模式。

对于 PIC18F2331/2431/4331/4431 器件来说，功耗管理模式是通过现成的 SLEEP 指令来调用。当发生中断、复位或 WDT 超时的时候都可以退出至 PRI_RUN 模式（PRI_RUN 模式是正常的全功率执行模式；CPU 和外设都由主振荡器源作为时钟源）。此外，功耗管理运行模式也可以退出至休眠模式或相应的空闲模式。

3.1 选择功耗管理模式

选择功耗管理模式要确定有提供给 CPU 的时钟源还是没有，以及选择了哪个时钟源。IDLEN 位控制 CPU 提供时钟源，SCS1:SCS0 位则选择时钟源。表 3-1 总结了各个模式、置位、时钟源和受影响的模块。

3.1.1 时钟源

可以通过置位 OSCCON 寄存器中的 SCS 位来选择时钟源。在功耗管理模式的空闲模式中有三个时钟源可供使用：主时钟（和配置寄存器 1H 中配置的一样）、辅助时钟（Timer1 振荡器）和内部振荡电路。间接和内部振荡电路时钟源可用于功耗管理模式（PRI_RUN 模式是全功率执行模式；CPU 和外设由主振荡器时钟源提供时钟源）。

表 3-1: 功耗管理模式

模式	OSCCON 位		模块时钟		有效时钟和振荡器源
	IDLEN <7>	SCS1:SCS0 <1:0>	CPU	外设	
休眠	0	00	关闭	关闭	无——所有的时钟都被禁止
PRI_RUN	0	00	被提供时钟源	被提供时钟源	主时钟——LP、XT、HS、HSPLL、RC、EC 和 INTRC ⁽¹⁾ 这是正常的全功率运行模式。
SEC_RUN	0	01	被提供时钟源	被提供时钟源	辅助时钟——Timer1 振荡器
RC_RUN	0	1x	被提供时钟源	被提供时钟源	内部振荡电路 ⁽¹⁾
PRI_IDLE	1	00	关闭	被提供时钟源	主时钟——LP、XT、HS、HSPLL、RC 和 EC
SEC_IDLE	1	01	关闭	被提供时钟源	辅助时钟——Timer1 振荡器
RC_IDLE	1	1x	关闭	被提供时钟源	内部振荡电路 ⁽¹⁾

注 1: 包含 INTOSC 和 INTOSC 后分频器以及 INTRC 源。

PIC18F2331/2431/4331/4431

3.1.2 进入功耗管理模式

一般情况下，进入、退出和在功耗管理时钟源之间切换都需要进行时钟源切换。在这几种情况下，事件的顺序都相同。

功耗管理模式的所有变化都由载入 OSCCON 寄存器并执行 SLEEP 指令开始。SCS1:SCS0 位选择三个功耗管理时钟源中的一个：主时钟（和配置寄存器 1H 中配置的一样）、辅助时钟（Timer1 振荡器）和内部振荡电路（在 RC 模式中使用）。在 SLEEP 指令执行之前修改 SCS 位不会有任何影响。执行 SLEEP 指令就可以进入功耗管理模式。

图 3-5 给出了从主时钟切换到 Timer1 振荡器时，系统如何计时。当执行 SLEEP 指令时，器件时钟在下一个指令周期开始时被停止。来自新时钟源的八个时钟周期与新的时钟源同步计时。来自新时钟源的八个时钟脉冲之后，来自新时钟源的时钟会恢复系统时钟计数。停顿的实际时间为新时钟源的八到九个时钟周期之间。这就保证了新时钟源是稳定的，且脉冲宽度不小于这两个时钟源最窄的脉冲宽度。

以下三位表明当前的时钟源：OSCCON 寄存器中的 OSTS 和 IOFS 以及 T1CON 寄存器中的 T1RUN。在功耗管理模式中，除了 PRI_RUN，这些位只有一位被置位。当 OSTS 位被置位时，主时钟提供系统时钟。当 IOFS 位被置位时，INTOSC 输出提供稳定的 8 MHz 时钟源并提供系统时钟。当 T1RUN 位被置位时，Timer1 振荡器提供系统时钟。如果这些位都不置位，则 INTRC 时钟源为系统时钟源，而 INTOSC 源并不稳定。

如果在配置寄存器 1H 中将内部振荡电路配置为主时钟源，则在 PRI_RUN 或 PRI_IDLE 模式中，OSTS 和 IOFS 位可能都会被置位。这表示主时钟（INTOSC 输出）正在产生稳定的 8 MHz 输出。进入 RC 功耗管理模式（频率相同）将清零 OSTS 位。

- 注 1:** 在修改单独的 IRCF 位时应该特别小心。如果 VDD 电压小于 3V，可以选择比低 VDD 电压支持的更高的时钟速度。违反 VDD/Fosc 规范会导致不正确的器件操作。
- 2:** 执行 SLEEP 指令并不需要将器件置于休眠模式；执行 SLEEP 指令只是一个简单地触发，目的是将控制器进入功耗管理模式，该模式通过 OSCCON 寄存器选择，休眠模式是这些功耗管理模式中的一种。

3.1.3 多种休眠命令

在 SLEEP 指令执行时，根据 IDLEN 和 SCS 位的位置，进入功耗管理模式。如果执行了另一条 SLEEP 指令，器件将进入此时这些相同位所指定的功耗管理模式。如果这些位已改变，器件将进入由新的位置位所指定的新功耗管理模式。

3.1.4 运行和空闲模式之间的比较

运行模式的时钟源选择与相应的空闲模式相同。当执行 SLEEP 指令时，OSCCON 寄存器的 SCS 位用于在不同时钟源之间切换。因此，如果执行 SLEEP 指令时时钟源发生改变，时钟将切换。

在空闲模式，CPU 没有时钟源，同时也不运行。在运行模式，CPU 有时钟源并执行代码。这一不同将造成 WDT 在超时溢出时的操作不同。在空闲模式，WDT 超时会将器件从功耗管理模式中唤醒。在运行模式，WDT 超时将引发 WDT 复位（见表 3-2）。

在从空闲模式唤醒的过程中，主时钟做好准备之前，CPU 进入相应的运行模式，并开始执行代码。主时钟做好准备后，时钟源自动切换到主时钟。在唤醒的过程中及唤醒后，IDLEN 和 SCS 位都不变。

图 3-2 所示为时钟源切换过程中系统的时序。这个示例假设器件处于 SEC_IDLE 或 SEC_RUN 模式时被唤醒（主时钟被配置为 HSPLL 模式）。

表 3-2: 功耗管理模式之间的比较

功耗管理模式	CPU 的时钟源	WDT 超时结果	外设的时钟源	唤醒过程中的时钟源 (当主时钟就绪时)
休眠	没有时钟源 (未运行)	唤醒	没有时钟源	如果双速起振或故障保护时钟监视器被使能, 则没有时钟源或者是 INTOSC 复用。
任何空闲模式	没有时钟源 (未运行)	唤醒	主时钟源、辅助时钟源或 INTOSC 复用	仍为空闲模式 (CPU 的运行与相应的运行模式相同)。
任何运行模式	辅助时钟源或 INTOSC 复用	复位	辅助时钟源或 INTOSC 复用器	仍为运行模式。

3.2 休眠模式

PIC18F2331/2431/4331/4431 器件的功耗管理休眠模式与所有其他 PICmicro[®] 控制器中的相同。通过清零 IDLEN 和 SCS1:SCS0 位 (在复位状态时), 并执行 SLEEP 指令就可以进入功耗管理休眠模式。这样做将关闭主振荡器并清零 OSTS 位 (见图 3-1)。

当在休眠模式中 (因中断、复位或 WDT 超时) 发生唤醒事件时, 在主时钟源就绪前, 系统没有时钟源 (见图 3-2), 或者, 当双速起振或故障保护时钟监视器被使能时, 系统将内部振荡电路作为时钟源 (见第 22.0 节“CPU 的特殊功能”)。在这两种情况下, 当主时钟提供系统时钟时, OSTS 位被置位。在这期间, 唤醒不会影响 IDLEN 和 SCS 位。

3.3 空闲模式

当外设继续运行时, IDLEN 位允许控制器的 CPU 选择性地关闭。清零 IDLEN 位允许 CPU 有时钟源。置位 IDLEN 位, 禁止 CPU 的时钟源, 并有效停止程序执行 (见寄存器 2-2)。不管 IDLEN 位是否置位, 外设都继续有时钟源。

IDLEN 位的工作方式有一个例外。当所有低功耗 OSCCON 位被清零时 (IDLEN:SCS1:SCS0 = 000), 执行 SLEEP 指令, 器件就进入休眠模式。这既是 OSCCON 寄存器的复位状态, 也是选择休眠模式的设置。这将保持与其他不提供功耗管理模式的 PICmicro 器件的兼容性。

如果在 SLEEP 指令执行时, 空闲使能位 IDLEN (OSCCON<7>) 置位, 外设将通过 SCS1:SCS0 位选定的时钟源计时, 但 CPU 没有时钟源。由于 CPU 没有执行指令, 只能通过中断、WDT 超时或复位从空闲模式退出。

当唤醒事件发生时, CPU 在准备好执行代码前, 要延迟大约 10 μ s。当 CPU 开始执行代码时, 它的时钟源与功耗管理模式所选定的时钟源相同 (也就是说, 当从 RC_IDLE 模式唤醒时, 内部振荡电路将为 CPU 和外设计时, 直到主时钟源就绪。这是基本的 RC_RUN 模式)。一直持续到主时钟源就绪。届时, OSTS 位被置位, 系统时钟源切换到主时钟 (见图 3-4)。在此期间, 唤醒不会影响 IDLEN 和 SCS 位。

在任何空闲模式或休眠模式中, WDT 超时会导致 WDT 唤醒, 器件进入全功率运行状态。

PIC18F2331/2431/4331/4431

图 3-1: 进入休眠模式的转换时序

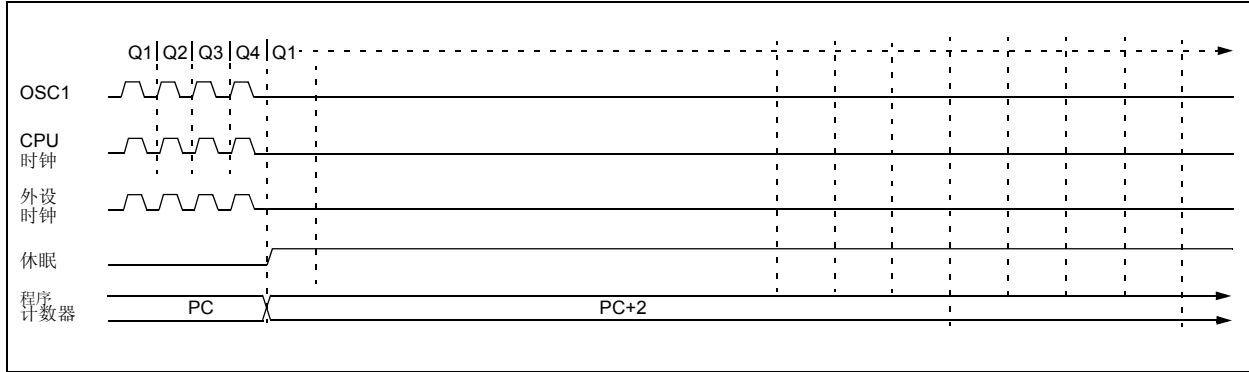
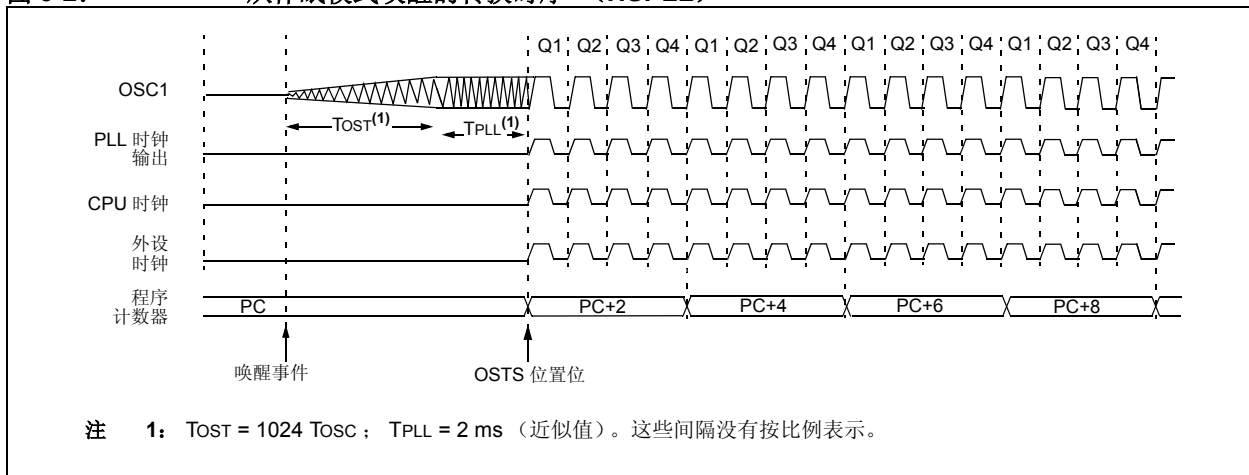


图 3-2: 从休眠模式唤醒的转换时序 (HSPLL)



3.3.1 PRI_IDLE 模式

此模式是三种低功耗空闲模式中唯一不禁止系统主时钟的。对于时间敏感的应用来说，由于时钟源不需要“热身”或从其他振荡器转换，选用此模式可以有更加精确的主时钟源，并以最快的速度恢复器件运行。

置位 IDLEN 位、清零 SCS 位并执行 SLEEP 指令就可以进入 PRI_IDLE 模式。虽然 CPU 被禁止，但外设将继续将配置寄存器 1H 指定的主时钟源作为时钟源。在 PRI_IDLE 模式中 OSTS 位保持置位（见图 3-3）。

当唤醒事件发生时，CPU 将主时钟源作为时钟源。在唤醒事件到开始执行代码大约有 10 μs 的延迟。需要这段时间让 CPU 做好准备执行指令。在唤醒之后，OSTS 位保持置位状态。在这期间，唤醒不会影响 IDLEN 和 SCS 位（见图 3-4）。

图 3-3: 转换到 PRI_IDLE 模式的时序

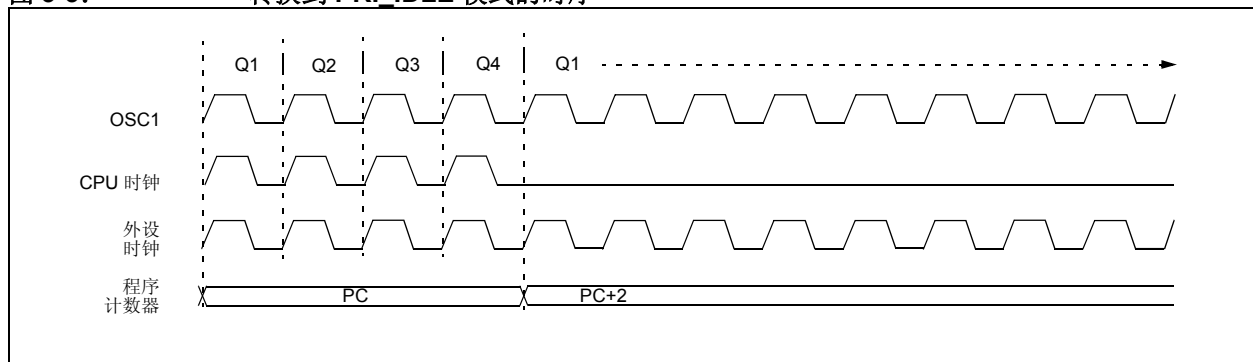
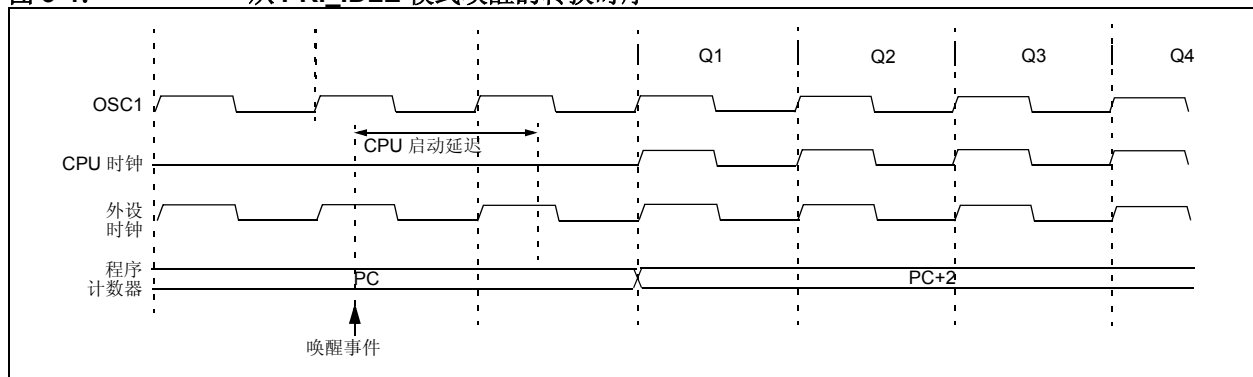


图 3-4: 从 PRI_IDLE 模式唤醒的转换时序



PIC18F2331/2431/4331/4431

3.3.2 SEC_IDLE 模式

在 SEC_IDLE 模式中，CPU 被禁止，但外设继续将 Timer1 振荡器作为时钟源。通过置位空闲位、指定 SCS1:SCS0 = 01，并执行 SLEEP 指令进入此模式。当时钟源切换到（见图 3-5）Timer1 振荡器时，主振荡器关闭，而 OSTS 位被清零且 T1RUN 位被置位。

注： Timer1 振荡器应该在进入 SEC_IDLE 模式前运行。如果执行 SLEEP 指令时 T1OSCEN 位没有被置位，实际会将强制执行 NOP 指令并不进入 SEC_IDLE 模式。如果 Timer1 振荡器被使能，但是还没有运行，外设时钟将被延迟，直到振荡器启动，在这样的情况下，初始振荡器运行很不稳定，它的运行结果无法预料。

当唤醒事件发生时，外设继续将 Timer1 振荡器作为时钟源。在唤醒事件后将延迟 10 μs，CPU 才开始执行代码，并将 Timer1 振荡器作为时钟源。单片机在 SEC_RUN 模式下运行，直到主时钟就绪。当主时钟就绪以后，时钟开始切换回主时钟（见图 3-6）。时钟切换完成后，T1RUN 位被清零，OSTS 位被置位，且主时钟提供系统时钟。在这期间，唤醒不影响 IDLEN 和 SCS 位，Timer1 振荡器继续运行。

图 3-5: 进入 SEC_IDLE 模式的转换时序

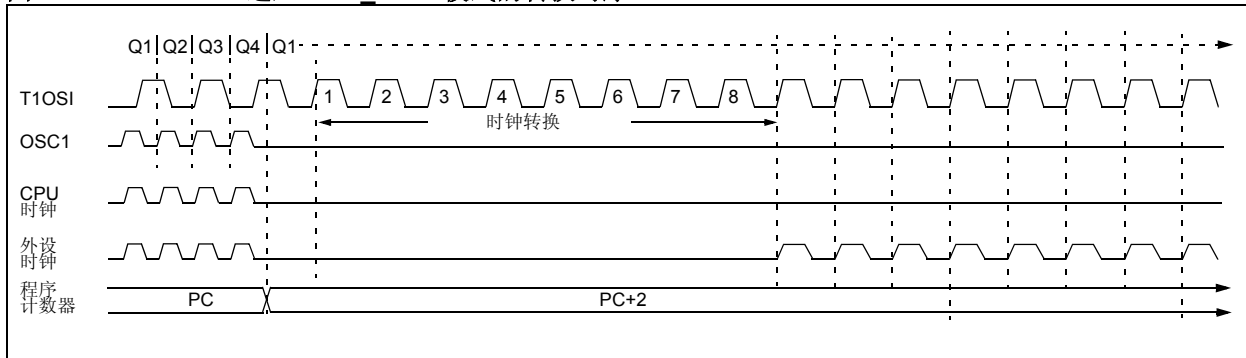
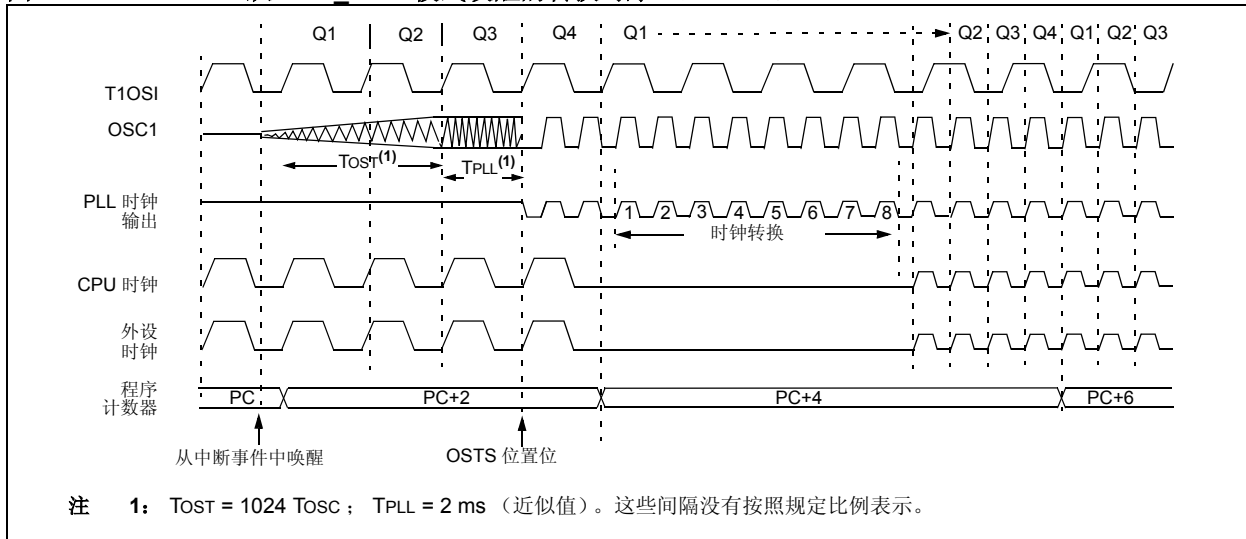


图 3-6: 从 SEC_RUN 模式唤醒的转换时序 (HSPLL)



3.3.3 RC_IDLE 模式

在 RC_IDLE 模式下，CPU 被禁止，但外设仍继续使用 INTOSC 复用器的内部振荡电路作为时钟源。该模式允许在空闲周期可控制地节省功耗。

通过置位 IDLEN 位、置位 SCS1 位（SCS0 被忽略），并执行 SLEEP 指令进入此模式。在执行 SLEEP 指令之前，通过指定 IRCF 位，INTOSC 复用器可以用于选择更高的时钟频率。当时钟源切换到 INTOSC 复用器时（见图 3-7），主振荡器关闭，OSTS 位清零。

如果 IRCF 位设置为一个非零的值（从而使能 INTOSC 输出），在 INTOSC 输出趋于稳定后，IOFS 位大约 1 ms 内被置位。当 INTOSC 源稳定时，外设的时钟源将继

续。如果 IRCF 位在 SLEEP 指令执行之前就设置为一个非零的值，且 INTOSC 源已经稳定，则 IOFS 位将保持置位状态。如果 IRCF 位全部清零，INTOSC 输出没有被使能，IOFS 位将保持清零状态，就不会表明当前时钟源。

当唤醒事件发生时，外设继续将 INTOSC 复用器作为时钟源。唤醒事件后将延迟 10 μs，CPU 才开始执行代码，将 INTOSC 复用器作为时钟源。单片机在 RC_RUN 模式下运行，直到主时钟就绪。当主时钟就绪后，时钟开始切换回主时钟（见图 3-8）。当时钟切换完成后，IOFS 位清零，OSTS 位置位，且主时钟提供系统时钟。在这期间，唤醒不影响 IDLEN 和 SCS 位。如果 WDT 或故障保护时钟监视器被使能，INTRC 源将继续运行。

图 3-7: 进入 RC_IDLE 模式的转换时序

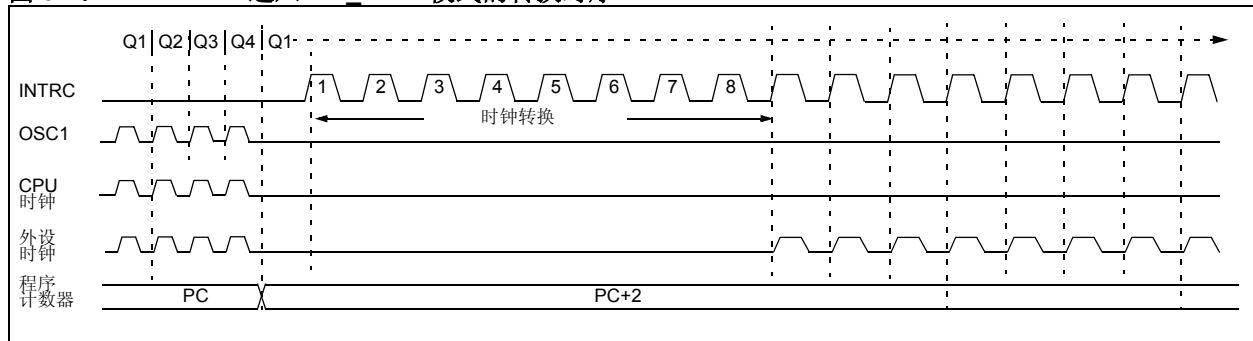
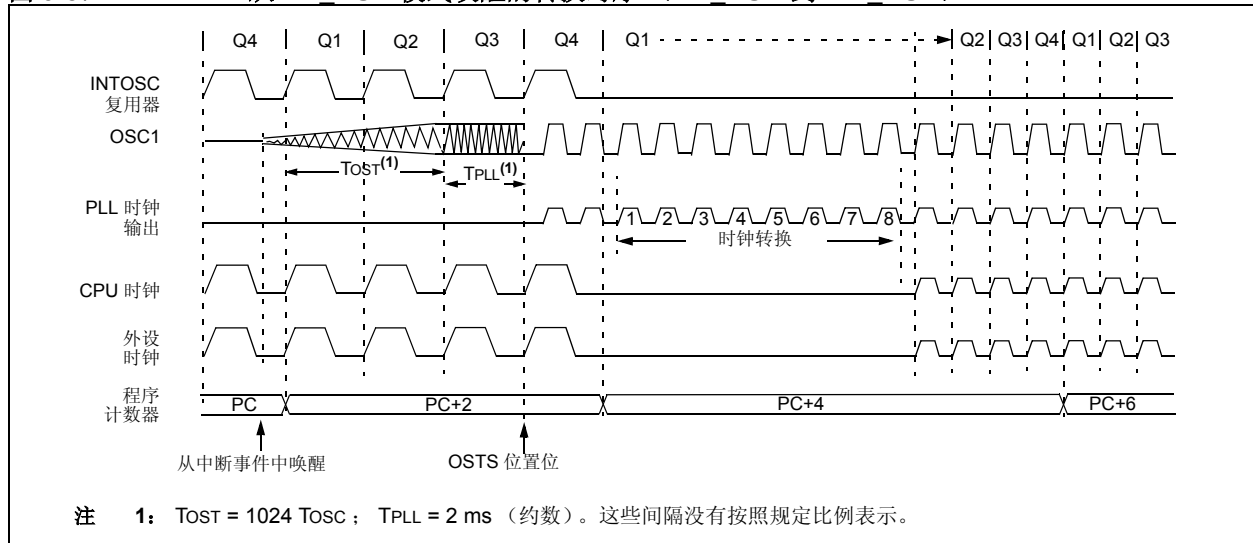


图 3-8: 从 RC_RUN 模式唤醒的转换时序 (RC_RUN 到 PRI_RUN)



注 1: $T_{OST} = 1024 T_{OSC}$; $T_{PLL} = 2 \text{ ms}$ (约数)。这些间隔没有按照规定比例表示。

PIC18F2331/2431/4331/4431

3.4 运行模式

如果 IDLEN 位在 SLEEP 指令执行时被清零，CPU 和外设都以 SCS1:SCS0 位所选时钟作为时钟源。而如果这些操作模式不能提供节省功耗的空闲或休眠模式，它们允许器件使用更低频率的时钟源来继续执行指令。RC_RUN 模式可在高于主时钟的频率下执行代码。

通过中断或任何复位可从功耗管理运行模式中唤醒，返回到全功率运行。当 CPU 在运行模式执行代码时，有几种可以退出运行模式的方法。包括退出到休眠模式、退出到相应的空闲模式以及通过执行 RESET 指令退出。当器件处于任一种功耗管理运行模式时，WDT 超时将导致 WDT 复位。

3.4.1 PRI_RUN 模式

PRI_RUN 模式是正常的全功率执行模式。如果不执行 SLEEP 指令，单片机就在此模式下运行（执行 SLEEP 指令可进入其他功耗管理模式）。当发生中断或 WDT 超时时，所有其他功耗管理模式都退出至 PRI_RUN 模式。

没有其他方法可以进入 PRI_RUN 模式。OSTS 位被置位。如果内部振荡电路为主时钟源，IOFS 位可能被置位（见第 2.7.1 节“振荡器控制寄存器”）。

3.4.2 SEC_RUN 模式

SEC_RUN 模式与其他 PIC18 器件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。这让用户在使用高精度时钟源的情况下可以选择更低的功耗。

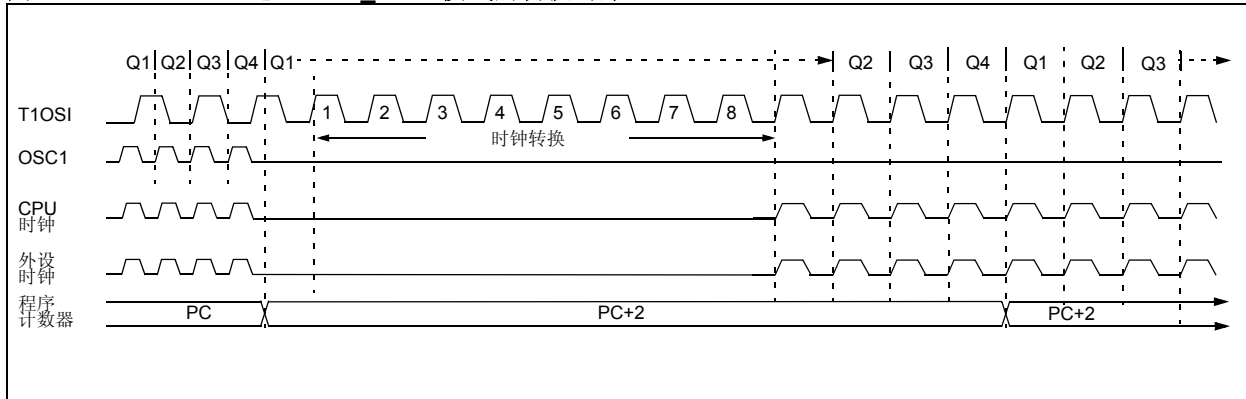
通过清零 IDLEN 位、设置 SCS1:SCS0 = 01 并执行 SLEEP 指令进入 SEC_RUN 模式。此时，系统时钟源切换到 Timer1 振荡器（见图 3-9），主振荡器关闭，T1RUN 位（T1CON<6>）置位且 OSTS 位清零。

注： Timer1 振荡器应该在进入 SEC_RUN 模式之前开始运行。如果执行 SLEEP 指令时 T1OSCN 位没有被置位，实际会将强制执行 NOP 指令并不进入 SEC_IDLE 模式。如果 Timer1 振荡器已经被使能，但仍然没有开始运行，系统时钟将会延迟，直到振荡器起振。在这种情况下，初始振荡器运行很不稳定，它的运行结果无法预料。

当发生唤醒事件时，外设和 CPU 继续将 Timer1 振荡器作为时钟源，同时主时钟启动。当主时钟就绪后，时钟切换回主时钟（见图 3-6）。当时钟切换完成后，T1RUN 位被清零，OSTS 位被置位而且主时钟提供系统时钟。在这期间，唤醒不影响 IDLEN 和 SCS 位，Timer1 振荡器继续运行。

固件可以强制从 SEC_RUN 模式退出。通过清零 T1OSCN 位（T1CON<3>）可以从 SEC_RUN 退出，返回到正常的全功率运行模式。尽管 T1OSCN 位被清零，Timer1 振荡器仍将继续运行并提供系统时钟。主时钟启动。当主时钟就绪后，时钟切换回主时钟（见图 3-6）。当时钟切换完成后，Timer1 振荡器被禁止，T1RUN 位被清零，OSTS 位被置位，而且主时钟做为系统的时钟源。在这期间，唤醒不影响 IDLEN 和 SCS 位。

图 3-9: 进入 SEC_RUN 模式的转换时序



3.4.3 RC_RUN 模式

RC_RUN 模式中，内部振荡电路通过 INTOSC 复用器作为 CPU 和外设的时钟源，此时主时钟关闭。在使用 INTRC 源时，此模式是所有运行模式中最节省功耗的运行模式，并仍可执行代码。对于对时间不太敏感或不是总是需要高速时钟的用户应用，选用此模式运行非常合适。

如果主时钟源为内部振荡电路（INTIO1 或 INTIO2 振荡器），在 PRI_RUN 和 RC_RUN 模式下运行就没有区别。但是在进入和退出 RC_RUN 模式时会发生时钟切换延迟。因此，如果主时钟源为内部振荡电路，不建议使用 RC_RUN 模式。

可以通过清零 IDLEN 位、置位 SCS1 位（SCS0 被忽略）并执行 SLEEP 指令来进入此模式。IRCF 位可能会在执行 SLEEP 指令前选择时钟频率。当时钟源切换到 INTOSC 复用器时（见图 3-10），主振荡器关闭，OSTS 位被清零。

IRCF 位在任何时候都可能被更改，以便在瞬间更改系统时钟的速度。从 INTOSC 复用器选择一个新的时钟频率不需要执行 SLEEP 指令。

注： 在更改单独的 IRCF 位时应该特别小心。如果 VDD 电压低于 3V，可以选择比低 VDD 电压所支持的更高的时钟速度。违反 VDD/Fosc 规范会导致不正确的器件操作。

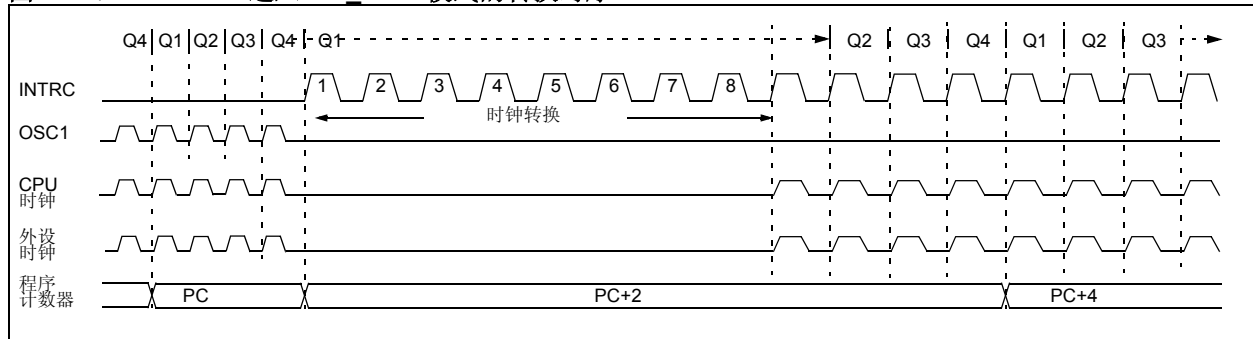
如果 IRCF 位全部被清零，则 INTOSC 输出没有被使能，且 IOFS 将保持清零状态，这样就没有当前时钟源的指示。INTRC 源提供系统时钟。

如果 IRCF 位从全清零状态改变（从而使能 INTOSC 输出），IOFS 位在 INTOSC 输出开始稳定后置位。系统时钟继续运行，而 INTOSC 源约在 1 ms 之内趋于稳定。

如果 IRCF 位在 SLEEP 指令执行之前就被设置为一个非零值，而且 INTOSC 源已经稳定，IOFS 位将保持置位状态。

当发生唤醒事件时，INTOSC 复用器继续作为系统的时钟源，同时主时钟启动。当主时钟就绪后，时钟切换回主时钟（见图 3-8）。当时钟切换完成后，IOFS 位被清零，OSTS 位被置位，且主时钟提供系统时钟。在这期间，唤醒不影响 IDLEN 和 SCS 位。如果 WDT 或故障保护时钟监视器被使能，INTRC 源将继续运行。

图 3-10: 进入 RC_RUN 模式的转换时序



PIC18F2331/2431/4331/4431

3.4.4 退出到空闲模式

通过置位 **IDLEN** 位和执行 **SLEEP** 指令可以从功耗管理运行模式退出到相应的空闲模式。CPU 在 **SLEEP** 指令后的下一个指令开始时停止运行。任何时钟源状态位 (**OSTS**、**IOFS** 或 **T1RUN**) 都不会发生改变。当 CPU 停止时，外设继续使用前面选定的时钟源。

3.4.5 退出到休眠模式

通过清零 **IDLEN** 位和 **SCS1:SCS0** 位并执行 **SLEEP** 指令，从功耗管理运行模式退出到休眠模式。此代码与从正常运行（全功率）模式调用休眠模式的方法没有区别。

主时钟和内部振荡电路被禁止。如果 **WDT** 使能，**INTRC** 将继续运行。如果在 **T1CON** 寄存器中使能了 **Timer1** 振荡器，它将继续运行。所有时钟源状态位 (**OSTS**、**IOFS** 和 **T1RUN**) 被清零。

3.5 从功耗管理模式唤醒

通过中断、复位或 **WDT** 超时可以从任何功耗管理模式退出。本节将讨论会导致退出功耗管理模式的触发。在每个功耗管理模式中我们还讨论了时钟源子系统的作用（见第 3.2 到 3.4 节）。

注：	如果应用代码对时间敏感，它应在继续执行之前等待 OSTS 位置位。利用低功耗退出序列期间的间隔时间 (OSTS 位置位前) 执行对时间不敏感的“日常管理”任务。
-----------	---

从低功耗模式退出期间，器件的行为见表 3-3 中的总结。

3.5.1 通过中断退出

任何可用的中断源都可以导致器件退出功耗管理模式并恢复全功率运行模式。要使能此功能，必须通过置位 **INTCON** 或 **PIE** 寄存器中对应的使能位来使能中断源。当相应的中断标志位被置位时，启动退出序列。当使用中断从低功耗模式退出时，如果 **GIE/GIEH** 位 (**INTCON<7>**) 被置位，代码执行就会跳转到中断向量。否则代码执行就会继续或重新开始，不会跳转（见第 9.0 节“中断”）。

PIC18F2331/2431/4331/4431

表 3-3: 从休眠模式或任何空闲模式唤醒时的行为和退出延迟 (通过时钟源)

功耗管理模式的时钟源	系统主时钟	功耗管理模式退出延迟	时钟就绪状态位 (OSCCON)	从功耗管理模式唤醒过程中的行为		
				通过中断退出	通过复位退出	
系统主时钟 (PRI_IDLE 模式)	LP、XT、HS	5-10 μ s ⁽⁵⁾	OSTS	主时钟提供 CPU 和外设时钟源并执行指令。	没有时钟源或双速起振 (如果被使能) ⁽³⁾ 。	
	HSPLL		—			
	EC、RC、INTRC ⁽¹⁾		—			
	INTOSC ⁽²⁾		IOFS			
T1OSC 或 INTRC ⁽¹⁾	LP, XT, HS	OST	OSTS	CPU 和外设时钟源由选定的功耗管理模式时钟提供并执行指令直到主时钟源就绪。		
	HSPLL	OST + 2 ms				
	EC, RC, INTRC ⁽¹⁾	5-10 μ s ⁽⁵⁾				—
	INTOSC ⁽²⁾	1 ms ⁽⁴⁾				IOFS
INTOSC ⁽²⁾	LP, XT, HS	OST	OSTS			
	HSPLL	OST + 2 ms				
	EC, RC, INTRC ⁽¹⁾	5-10 μ s ⁽⁵⁾				—
	INTOSC ⁽²⁾	无				IOFS
休眠模式	LP, XT, HS	OST	OSTS	无时钟源或双速起振 (如果被使能) 直到主时钟源就绪 ⁽³⁾ 。		
	HSPLL	OST + 2 ms				
	EC, RC, INTRC ⁽¹⁾	5-10 μ s ⁽⁵⁾				—
	INTOSC ⁽²⁾	1 ms ⁽⁴⁾				IOFS

- 注 1: 这种情况是特别针对 INTRC 时钟源的。
 2: 包括 INTOSC 8 MHz 时钟源和后分频器的输出频率。
 3: 第 22.3 节“双速起振”包含有关双速起振的详细内容。
 4: 在 INTOSC 稳定期间, 代码将继续执行。
 5: 从休眠模式和所有空闲模式唤醒时都需要延迟。此延迟与其他所需延迟同时进行 (见第 3.3 节“空闲模式”)。

PIC18F2331/2431/4331/4431

3.5.2 通过复位退出

通常，器件通过振荡器起振定时器（OST）保持在复位状态，直到主时钟（由配置寄存器 1H 定义）就绪。在这个时候，OSTS 位被置位，器件开始执行代码。

代码可以在主时钟就绪之前开始执行。如果双速起振（见第 22.3 节“双速起振”）或故障保护时钟监视器（见第 22.4 节“故障保护时钟监视器”）在配置寄存器 1H 中被使能，一旦复位源清除就开始执行代码了。由内部振荡电路驱动的 INTOSC 复用器作为代码执行的时钟源。由于 OSCCON 寄存器在所有复位之后被清零，所以选择 INTRC 时钟源。通过更改 OSCCON 寄存器中的 IRCF 位可以选择速度更高的时钟。执行代码时，由内部振荡电路提供时钟源直到主时钟就绪，或者在主时钟就绪前进入功耗管理模式，然后关闭主时钟。

3.5.3 通过 WDT 超时退出

根据超时发生时器件所处的不同功耗管理模式，WDT 超时会引发不同的操作。

如果器件没有在执行代码（所有空闲模式和休眠模式），超时将导致器件从功耗管理模式唤醒（见第 3.2 节“休眠模式”到第 3.4 节“运行模式”）。

如果器件在执行代码（所有运行模式），超时将导致 WDT 复位（见第 22.2 节“看门狗定时器（WDT）”）。

通过执行 SLEEP 或 CLRWDT 指令使 WDT 定时器和后分频器清零，当前选择的时钟源失效（如果故障保护时钟监视器使能时），并且内部振荡电路作为系统时钟源时，修改 OSCCON 寄存器中的 IRCF 位。

3.5.4 在没有振荡器起振延迟的情况下退出

从功耗管理模式的某些退出根本不会调用 OST。它们是：

- 主时钟源没有停止的 PRI_IDLE 模式；和
- 主时钟源不是 LP、XT、HS 或 HSPLL 中的任意一种。

在这些情况下，主时钟源不需要振荡器起振延迟，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迟（RC、EC 和 INTIO 振荡器模式）。

但是，当离开休眠和空闲模式时，在唤醒事件之后需要一个固定的延迟（大约为 10 μ s）。CPU 需要此延迟来准备执行代码。指令执行在延迟后的第一个时钟周期重新开始。

3.6 INTOSC 频率漂移

厂家将内部振荡电路输出（INTOSC）校准为 8 MHz。但是，此频率可能会随着 VDD 电压或温度的改变而发生漂移，这可能会以各种方式影响控制器运行。

通过修改 OSCTUNE 寄存器内的值能够调整 INTOSC 频率。这样做的副作用就是 INTRC 时钟源频率也会受到影响。但是，使用 INTRC 时钟源的功能部件往往不要求精确的频率。这些部件包括选择 INTRC 时钟源时，故障保护时钟监视器、看门狗定时器和 RC_RUN/RC_IDLE 模式。

调整 INTOSC 需要了解何时需要调整、应该在哪个方向进行调整，以及在某些情况下需要做多大的改变。以下是三个示例，但是可能会用到其他技术。

3.6.1 USART 示例

当 USART 在异步模式下开始产生帧错误或者接收数据有误时，表示可能需要调整。帧错误表示系统时钟频率太高，尝试减小 OSCTUNE 寄存器中的值来降低系统时钟频率。数据中的错误可能暗示着系统时钟速度太慢，尝试增加 OSCTUNE 寄存器中的值。

3.6.2 定时器示例

此技术可以将系统时钟的速度与其他参考时钟进行比较。可能要用两个定时器，一个为外设时钟提供时钟源，而另一个由一个固定的参考时钟提供时钟源，比如 Timer1 振荡器。

两个定时器都被清零，但由参考时钟为时钟源的定时器产生中断。当发生中断时，时钟源在内部的定时器被读取，且两个定时器都被清零。如果时钟源在内部的定时器值大于期望值，则表示内部振荡电路运行过快，尝试减小 OSCTUNE 寄存器中的值。

3.6.3 捕捉模式的 CCP 示例

CCP 模块可以使用自由运行的 Timer1，由内部振荡电路和有已知周期的外部事件（即，AC 电源频率）提供时钟源。在 CCPRxH:CCPRxL 寄存器中记录捕获的第一个事件时间，以供稍后使用。当第二个事件引起捕捉时，第二个事件的时间减去第一个事件的时间。由于外部事件周期是已知的，可以计算事件之间的时间差。

如果测量得到的时间比计算得到的时间大很多，则表示内部振荡电路运行过快，尝试减小 OSCTUNE 寄存器中的值。如果测量得到的时间比计算得到的时间小，则表示内部振荡电路运行过慢，尝试增加 OSCTUNE 寄存器中的值。

PIC18F2331/2431/4331/4431

注:

4.0 复位

PIC18F2331/2431/4331/4431 器件有多种复位方式：

- 上电复位 (POR)
- 正常工作下的 MCLR 复位
- 休眠状态下的 MCLR 复位
- 看门狗定时器 (WDT) 复位 (执行程序期间)
- 可编程欠压复位 (Brown-out Reset, BOR)
- RESET 指令
- 堆栈满复位
- 堆栈下溢复位

大多数寄存器不受复位的影响。在 POR 时寄存器状态未知，而在其他复位时寄存器状态不会改变。根据所发生复位的类型，另一些寄存器会被强行置为“复位状态”。

大多数寄存器不受 WDT 唤醒的影响，这是因为 WDT 唤醒被视为对正常运行的恢复。如表 4-2 所示，RCON 寄存器中的状态位：RI、TO、PD、POR 和 BOR，在不同的复位情况下会分别被置位或清零。这些状态位在软件中用于判断复位的类型。关于所有寄存器复位状态的全面描述，参见表 4-3。

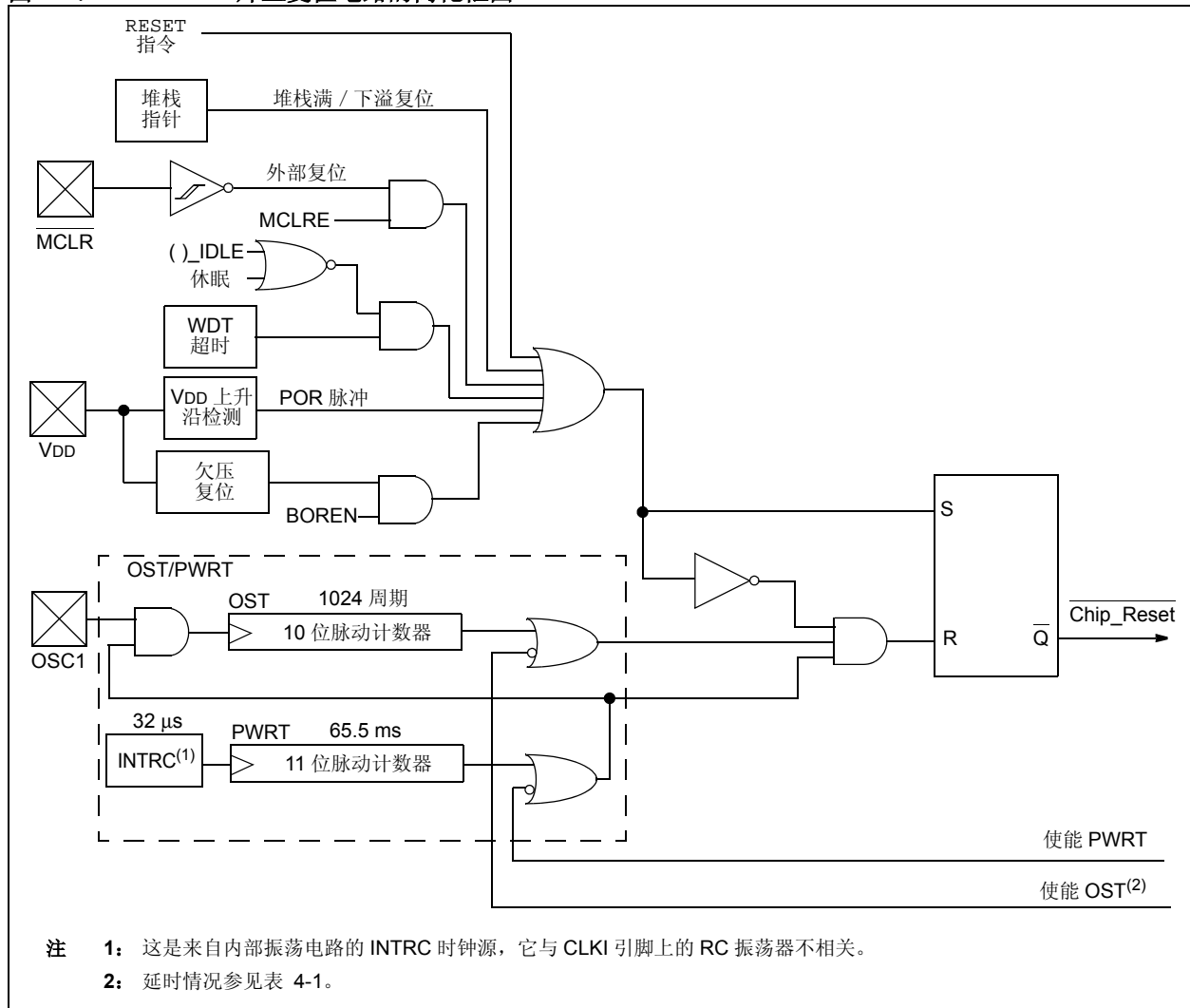
图 4-1 显示了一个片上复位电路的简化框图。

增强型 MCU 器件在 MCLR 复位路径上有一个 MCLR 噪声滤波器。该滤波器可以检测并滤出小的尖脉冲信号。

任何内部复位（包括 WDT 复位）都不能将 MCLR 引脚驱动为低电平。

由 MCLR 引脚提供的 MCLR 输入可用配置寄存器 3H 中的 MCLRE 位 (CONFIG3H<7>) 禁止。如需更多信息，参见第 22.1 节“配置位”。

图 4-1： 片上复位电路的简化框图



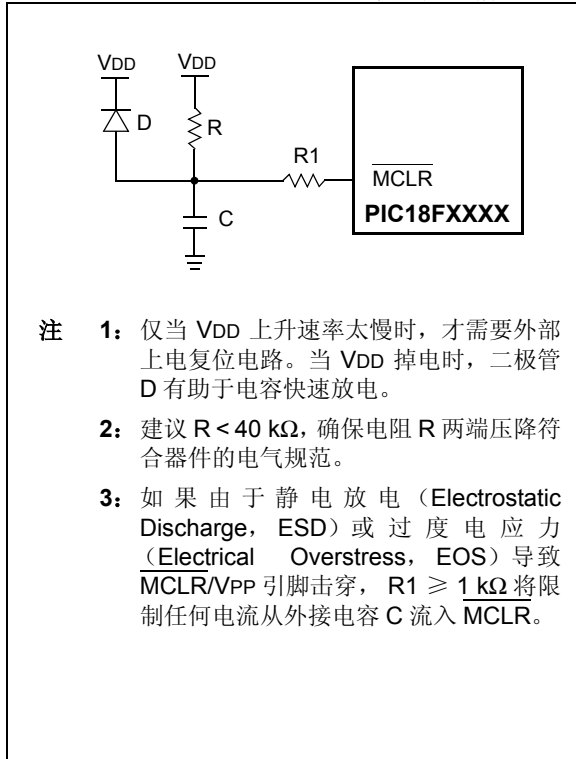
PIC18F2331/2431/4331/4431

4.1 上电复位 (POR)

检测到 VDD 上升时, 会在片上产生一个上电复位脉冲。要利用 POR 电路, 只需将 MCLR 引脚通过一个电阻 (1k 至 10 kΩ) 与 VDD 相连。这样可以省去产生上电复位延时通常所需的外接 RC 元件。参数 D004 标明了 VDD 的最小上升速率。对于上升速率缓慢的情况, 参见图 4-2。

当器件开始正常工作 (即退出复位状态) 时, 其工作参数 (电压、频率和温度等) 必须在相应的工作范围内, 以确保正常工作。如果不满足这些条件, 器件必须保持在复位状态直到工作条件满足为止。

图 4-2: 外部上电复位电路 (针对 VDD 上升速率缓慢的情况)



4.2 上电延时定时器 (PWRT)

PIC18F2331/2431/4331/4431 器件的上电延时定时器 (PWRT) 是一个 11 位计数器, 它使用 INTRC 时钟源作为时钟输入。这就产生共计 $2048 \times 32\text{ }\mu\text{s} = 65.6\text{ ms}$ 的延迟时长。当 PWRT 计数时, 器件保持在复位状态。

上电延时时间取决于 INTRC 时钟, 而且由于温度和制造工艺的影响, 不同芯片的延迟时间也各不相同。如需了解详细信息, 参见 DC 参数 #33。

通过清零配置位 PWRTE 可使能 PWRT。

4.3 振荡器起振定时器 (OST)

在 PWRT 延时结束以后 (见参数 #33), 振荡器起振定时器 (OST) 提供了 1024 个振荡器周期的延时 (从 OSC1 输入)。这可以确保晶振或谐振器起振和并建立稳定的振荡。

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

4.4 PLL 锁定延时

在 PLL 模式下使能 PLL, 上电复位后的延时序列与其他振荡器模式稍微有所不同。上电延时定时器会部分用于提供足够的固定延时, 让 PLL 锁定在主振荡器频率上。此 PLL 锁定延时 (TPLL) 通常为 2 ms, 从振荡器起振延时后开始。

4.5 欠压复位 (BOR)

配置位 BOREN 可以禁止 (如果清零 / 编程清零) 或使能 (如果置位) 欠压复位电路。如果 VDD 电压下降到 VBOR (参数 D005) 以下, 且持续时间超过 TBOR (参数 #35), 这种欠压状况将使芯片复位。如果 VDD 电压下降到 VBOR 以下, 持续时间少于 TBOR, 不一定会发生复位。芯片将保持在欠压复位状态, 直到 VDD 电压上升到 VBOR 以上。如果使能上电延时定时器, 则它将在 VDD 电压上升到 VBOR 以上之后启动工作; 并使芯片在一个额外的延时 TPWRT (参数 #33) 期间保持复位。如果上电延时定时器运行时, VDD 电压降到 VBOR 以下, 芯片将重新回到欠压复位状态, 并将初始化上电延时定时器。一旦 VDD 电压上升到 VBOR 以上, 上电延时定时器将执行一个额外的延时。使能 BOR 复位并不会自动使能 PWRT。

4.6 延时时序

上电时的延时时序如下: 首先, 在 POR 脉冲清零后, 启动 PWRT 延时 (若使能)。然后, OST 被激活。总延时时间将取决于振荡器的配置和 PWRT 的状态。例如, RC 模式下禁止 PWRT 时, 根本不会出现延时。图 4-3 到 4-7 描述了上电时的延时时序。

因为延时的发生由 POR 脉冲引起, 如果 MCLR 保持足够长时间的低电平, 所有延时将结束。将 MCLR 电平拉高后程序将立即执行 (图 4-5)。这对于测试或同步并行工作的多个 PIC18FXXXX 器件是非常有用的。

表 4-2 给出了一些特殊功能寄存器的复位条件, 而表 4-3 给出了所有寄存器的复位状态。

PIC18F2331/2431/4331/4431

表 4-1: 不同情况下的延时

振荡器模式	上电复位 ⁽²⁾ 和欠压复位		从功耗管理模式退出
	PWRTE \overline{N} =0	PWRTE \overline{N} =1	
HSPLL	66 ms ⁽¹⁾ +1024Tosc+2 ms ⁽²⁾	1024 TOSC+2 ms ⁽²⁾	1024 TOSC+2 ms ⁽²⁾
HS, XT, LP	66 ms ⁽¹⁾ +1024 TOSC	1024 TOSC	1024 TOSC
EC, ECIO	66 ms ⁽¹⁾	-	-
RC, RCIO	66 ms ⁽¹⁾	-	-
INTIO1, INTIO2	66 ms ⁽¹⁾	-	-

注 1: 66 ms (65.5 ms) 是上电延时定时器 (PWRT) 标称的延时长。

注 2: 2 ms 是 4x PLL 锁定所需的标称时间。

寄存器 4-1: RCON 寄存器位和位置

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

注: 关于位的定义, 请参阅第 5.14 节 “RCON 寄存器”。

表 4-2: 状态位及其含义和 RCON 寄存器的初始化状态

条件	程序计数器	RCON 寄存器	RI	TO	PD	POR	BOR	STKFUL	STKUNF
上电复位	0000h	0--1 1100	1	1	1	0	0	0	0
RESET 指令	0000h	0--0 uuuu	0	u	u	u	u	u	u
欠压	0000h	0--1 11u-	1	1	1	u	0	u	u
处于功耗管理运行模式下的 MCLR	0000h	0--u 1uuu	u	1	u	u	u	u	u
处于功耗管理空闲和休眠模式下的 MCLR	0000h	0--u 10uu	u	1	0	u	u	u	u
全速或功耗管理运行模式期间的 WDT 超时	0000h	0--u 0uuu	u	0	u	u	u	u	u
全速运行期间的 MCLR	0000h	0--u uuuu	u	u	u	u	u	u	u
堆栈满复位 (STVREN=1)								1	u
堆栈下溢复位 (STVREN=1)								u	1
堆栈下溢错误 (不是真正的复位, STVREN=0)	0000h	u--u uuuu	u	u	u	u	u	u	1
功耗管理空闲或休眠模式期间的 WDT 超时	PC+2	u--u 00uu	u	0	0	u	u	u	u
发生中断, 退出功耗管理模式	PC+2 ⁽¹⁾	u--u u0uu	u	u	0	u	u	u	u

图注: u= 不变, x= 未知, -= 未实现位, 读作 0。

注 1: 当芯片被中断唤醒且 GIEH 或 GIEL 位被置位时, PC 装入中断向量 (0x000008h 或 0x000018h)。

PIC18F2331/2431/4331/4431

表 4-3: 所有寄存器的初始化状态

寄存器	适用器件				上电复位和 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位	通过 WDT 或中断唤醒器件
TOSU	2331	2431	4331	4431	---0 0000	---0 0000	---0 uuuu ⁽³⁾
TOSH	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu ⁽³⁾
TOSL	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu ⁽³⁾
STKPTR	2331	2431	4331	4431	00-0 0000	uu-0 0000	uu-u uuuu ⁽³⁾
PCLATU	2331	2431	4331	4431	---0 0000	---0 0000	---u uuuu
PCLATH	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PCL	2331	2431	4331	4431	0000 0000	0000 0000	PC+2 ⁽²⁾
TBLPTRU	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
TBLPTRH	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
TBLPTL	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
TABLAT	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PRODH	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	2331	2431	4331	4431	0000 000x	0000 000u	uuuu uuuu ⁽¹⁾
INTCON2	2331	2431	4331	4431	1111 -1-1	1111 -1-1	uuuu -u-u ⁽¹⁾
INTCON3	2331	2431	4331	4431	11-0 0-00	11-0 0-00	uu-u u-uu ⁽¹⁾
INDF0	2331	2431	4331	4431	N/A	N/A	N/A
POSTINC0	2331	2431	4331	4431	N/A	N/A	N/A
POSTDEC0	2331	2431	4331	4431	N/A	N/A	N/A
PREINC0	2331	2431	4331	4431	N/A	N/A	N/A
PLUSW0	2331	2431	4331	4431	N/A	N/A	N/A
FSR0H	2331	2431	4331	4431	---- xxxx	---- uuuu	---- uuuu
FSR0L	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	2331	2431	4331	4431	N/A	N/A	N/A
POSTINC1	2331	2431	4331	4431	N/A	N/A	N/A
POSTDEC1	2331	2431	4331	4431	N/A	N/A	N/A
PREINC1	2331	2431	4331	4431	N/A	N/A	N/A
PLUSW1	2331	2431	4331	4431	N/A	N/A	N/A

图注: u= 不变, x= 未知, -= 未实现位, 读作 0, q= 值取决于状态。
阴影单元格表示状态不适用于指定器件。

- 注
- 1: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响（而引起唤醒）。
 - 2: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置位时, PC 装入中断向量（0008h 或 0018h）。
 - 3: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置位时, TOSU、TOSH 和 TOSL 被 PC 的当前值更新。STKPTR 被修改为指向硬件堆栈中的下一个单元。
 - 4: 关于特定状态下的复位值, 参见表 4-2。
 - 5: 根据选择的振荡器模式, PORTA、LATA 和 TRISA 中的 bit 6 和 bit 7 被使能。如果不作为 PORTA 引脚使能, 它们将被禁止且读作 0。
 - 6: 如果 MCLR 功能被禁止, PORTE 和 LATE 中的 bit 3 将被使能。如果不作为 PORTE 引脚使能, 它们将被禁止且读作 0。当禁止 MCLR 时, 28 引脚器件的 PORTE 上只有 RE3 可用。

PIC18F2331/2431/4331/4431

表 4-3: 所有寄存器的初始化状态 (续)

寄存器	适用器件				上电复位和 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位	通过 WDT 或中断唤醒器件
FSR1H	2331	2431	4331	4431	---- xxxx	---- uuuu	---- uuuu
FSR1L	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	2331	2431	4331	4431	---- 0000	---- 0000	---- uuuu
INDF2	2331	2431	4331	4431	N/A	N/A	N/A
POSTINC2	2331	2431	4331	4431	N/A	N/A	N/A
POSTDEC2	2331	2431	4331	4431	N/A	N/A	N/A
PREINC2	2331	2431	4331	4431	N/A	N/A	N/A
PLUSW2	2331	2431	4331	4431	N/A	N/A	N/A
FSR2H	2331	2431	4331	4431	---- xxxx	---- uuuu	---- uuuu
FSR2L	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	2331	2431	4331	4431	---x xxxx	---u uuuu	---u uuuu
TMR0H	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
TMR0L	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	2331	2431	4331	4431	11-- 1111	11-- 1111	uu-- uuuu
OSCCON	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
LVDCON	2331	2431	4331	4431	--00 0101	--00 0101	--uu uuuu
WDTCON	2331	2431	4331	4431	---- ---0	---- ---0	---- ---u
RCON ⁽⁴⁾	2331	2431	4331	4431	0--1 11q0	0--q qquu	u--u qquu
TMR1H	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	2331	2431	4331	4431	0000 0000	u0uu uuuu	uuuu uuuu
TMR2	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PR2	2331	2431	4331	4431	1111 1111	1111 1111	1111 1111
T2CON	2331	2431	4331	4431	-000 0000	-000 0000	-uuu uuuu
SSPBUF	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPADD	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
SSPCON	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu

图注: u= 不变, x= 未知, -= 未实现位, 读作 0, q= 值取决于状态。
阴影单元格表示状态不适用于指定器件。

- 注
- 1: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (而引起唤醒)。
 - 2: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置位时, PC 装入中断向量 (0008h 或 0018h)。
 - 3: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置位时, TOSU、TOSH 和 TOSL 被 PC 的当前值更新。STKPTR 被修改为指向硬件堆栈中的下一个单元。
 - 4: 关于特定状态下的复位值, 参见表 4-2。
 - 5: 根据选择的振荡器模式, PORTA、LATA 和 TRISA 中的 bit 6 和 bit 7 被使能。如果不作为 PORTA 引脚使能, 它们将被禁止且读作 0。
 - 6: 如果 MCLR 功能被禁止, PORTE 和 LATE 中的 bit 3 将被使能。如果不作为 PORTE 引脚使能, 它们将被禁止且读作 0。当禁止 MCLR 时, 28 引脚器件的 PORTE 上只有 RE3 可用。

PIC18F2331/2431/4331/4431

表 4-3: 所有寄存器的初始化状态 (续)

寄存器	适用器件				上电复位和 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位	通过 WDT 或中断唤醒器件
ADRESH	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
ADCON1	2331	2431	4331	4431	00-0 1000	00-- 1000	uu-u uuuu
ADCON2	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
CCPR1H	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
CCPR2H	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2L	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
ANSEL0	2331	2431	4331	4431	1111 1111	1111 1111	uuuu uuuu
ANSEL1	2331	2431	4331	4431	---- ---0	---- ---0	---- ---u
T5CON	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
QEICON	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
SPBRGH	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
SPBRG	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
RCREG	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
TXREG	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
TXSTA	2331	2431	4331	4431	0000 -010	0000 -010	uuuu -uuu
RCSTA	2331	2431	4331	4431	0000 000x	0000 000x	uuuu uuuu
BAUDCTL	2331	2431	4331	4431	-1-1 0-00	-1-1 0-00	-u-u u-uu
EEADR	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
EEDATA	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
EECON1	2331	2431	4331	4431	xx-0 x000	uu-0 u000	uu-0 u000
EECON2	2331	2431	4331	4431	0000 0000	0000 0000	0000 0000
IPR3	2331	2431	4331	4431	---1 1111	---1 1111	---u uuuu
PIE3	2331	2431	4331	4431	---0 0000	---0 0000	---u uuuu
PIR3	2331	2431	4331	4431	---0 0000	---0 0000	---u uuuu

图注: u= 不变, x= 未知, -= 未实现位, 读作 0, q= 值取决于状态。
阴影单元格表示状态不适用于指定器件。

- 注 1: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (而引起唤醒)。
 2: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置位时, PC 装入中断向量 (0008h 或 0018h)。
 3: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置位时, TOSU、TOSH 和 TOSL 被 PC 的当前值更新。STKPTR 被修改为指向硬件堆栈中的下一个单元。
 4: 关于特定状态下的复位值, 参见表 4-2。
 5: 根据选择的振荡器模式, PORTA、LATA 和 TRISA 中的 bit 6 和 bit 7 被使能。如果不作为 PORTA 引脚使能, 它们将被禁止且读作 0。
 6: 如果 MCLR 功能被禁止, PORTE 和 LATE 中的 bit 3 将被使能。如果不作为 PORTE 引脚使能, 它们将被禁止且读作 0。当禁止 MCLR 时, 28 引脚器件的 PORTE 上只有 RE3 可用。

PIC18F2331/2431/4331/4431

表 4-3: 所有寄存器的初始化状态 (续)

寄存器	适用器件				上电复位和 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位	通过 WDT 或中断唤醒器件
IPR2	2331	2431	4331	4431	1--1 -1-1	1--1 -1-1	u--u -u-u
PIR2	2331	2431	4331	4431	0--0 -0-0	0--0 -0-0	u--u -u-u
PIE2	2331	2431	4331	4431	0--0 -0-0	0--0 -0-0	u--u -u-u
IPR1	2331	2431	4331	4431	1111 1111	1111 1111	uuuu uuuu
	2331	2431	4331	4431	-111 1111	-111 1111	-uuu uuuu
PIR1	2331	2431	4331	4431	-000 0000	-000 0000	-uuu uuuu ⁽¹⁾
	2331	2431	4331	4431	-000 0000	-000 0000	-uuu uuuu ⁽¹⁾
PIE1	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
	2331	2431	4331	4431	-000 0000	-000 0000	-uuu uuuu
OSCTUNE	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
ADCON3	2331	2431	4331	4431	00-0 0000	00-0 0000	uu-u uuuu
ADCHS	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
TRISE ⁽⁶⁾	2331	2431	4331	4431	---- -111	---- -111	---- -uuu
TRISD	2331	2431	4331	4431	1111 1111	1111 1111	uuuu uuuu
TRISC	2331	2431	4331	4431	1111 1111	1111 1111	uuuu uuuu
TRISB	2331	2431	4331	4431	1111 1111	1111 1111	uuuu uuuu
TRISA ⁽⁵⁾	2331	2431	4331	4431	1111 1111 ⁽⁵⁾	1111 1111 ⁽⁵⁾	uuuu uuuu ⁽⁵⁾
PR5H	2331	2431	4331	4431	1111 1111	1111 1111	uuuu uuuu
PR5L	2331	2431	4331	4431	1111 1111	1111 1111	uuuu uuuu
LATE ⁽⁶⁾	2331	2431	4331	4431	---- -xxx	---- -uuu	---- -uuu
LATD	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATC	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATB	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
LATA ⁽⁵⁾	2331	2431	4331	4431	xxxx xxxx ⁽⁵⁾	uuuu uuuu ⁽⁵⁾	uuuu uuuu ⁽⁵⁾
TMR5H	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR5L	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE ⁽⁶⁾	2331	2431	4331	4431	---- xxxxx	---- xxxxx	---- uuuu
PORTD	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA ⁽⁵⁾	2331	2431	4331	4431	xx0x 0000 ⁽⁵⁾	uu0u 0000 ⁽⁵⁾	uuuu uuuu ⁽⁵⁾

图注: u= 不变, x= 未知, -= 未实现位, 读作 0, q= 值取决于状态。
阴影单元格表示状态不适用于指定器件。

- 注
- 1: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (而引起唤醒)。
 - 2: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置位时, PC 装入中断向量 (0008h 或 0018h)。
 - 3: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置位时, TOSU、TOSH 和 TOSL 被 PC 的当前值更新。STKPTR 被修改为指向硬件堆栈中的下一个单元。
 - 4: 关于特定状态下的复位值, 参见表 4-2。
 - 5: 根据选择的振荡器模式, PORTA、LATA 和 TRISA 中的 bit 6 和 bit 7 被使能。如果不作为 PORTA 引脚使能, 它们将被禁止且读作 0。
 - 6: 如果 MCLR 功能被禁止, PORTE 和 LATE 中的 bit 3 将被使能。如果不作为 PORTE 引脚使能, 它们将被禁止且读作 0。当禁止 MCLR 时, 28 引脚器件的 PORTE 上只有 RE3 可用。

PIC18F2331/2431/4331/4431

表 4-3: 所有寄存器的初始化状态 (续)

寄存器	适用器件				上电复位和 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位	通过 WDT 或中断唤醒器件
PTCON0	2331	2431	4331	4431	0000 0000	uuuu uuuu	uuuu uuuu
PTCON1	2331	2431	4331	4431	00-- ----	00-- ----	uu-- ----
PTMRL	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PTMRH	2331	2431	4331	4431	---- 0000	---- 0000	---- uuuu
PTPERL	2331	2431	4331	4431	1111 1111	1111 1111	uuuu uuuu
PTPERH	2331	2431	4331	4431	---- 1111	---- 1111	---- uuuu
PDC0L	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
PDC0H	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PDC1L	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PDC1H	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
PDC2L	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PDC2H	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
PDC3L	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
PDC3H	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
SEVTCMPL	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
SEVTCMPH	2331	2431	4331	4431	---- 0000	---- 0000	---- uuuu
PWMCON0	2331	2431	4331	4431	-101 0000	-101 0000	-uuu uuuu
PWMCON1	2331	2431	4331	4431	0000 0-00	0000 0-00	uuuu u-uu
DTCON	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
FLTCONFIG	2331	2431	4331	4431	-000 0000	-000 0000	-uuu uuuu
OVDCOND	2331	2431	4331	4431	1111 1111	1111 1111	uuuu uuuu
OVDCONS	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
CAP1BUFH/ VELRH	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CAP1BUFL/ VELRL	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CAP2BUFH/ POSCNTH	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CAP2BUFL/ POSCNTL	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu

图注: u= 不变, x= 未知, -= 未实现位, 读作 0, q= 值取决于状态。
阴影单元格表示状态不适用于指定器件。

- 注
- 1: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (而引起唤醒)。
 - 2: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置位时, PC 装入中断向量 (0008h 或 0018h)。
 - 3: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置位时, TOSU、TOSH 和 TOSL 被 PC 的当前值更新。STKPTR 被修改为指向硬件堆栈中的下一个单元。
 - 4: 关于特定状态下的复位值, 参见表 4-2。
 - 5: 根据选择的振荡器模式, PORTA、LATA 和 TRISA 中的 bit 6 和 bit 7 被使能。如果不作为 PORTA 引脚使能, 它们将被禁止且读作 0。
 - 6: 如果 MCLR 功能被禁止, PORTE 和 LATE 中的 bit 3 将被使能。如果不作为 PORTE 引脚使能, 它们将被禁止且读作 0。当禁止 MCLR 时, 28 引脚器件的 PORTE 上只有 RE3 可用。

PIC18F2331/2431/4331/4431

表 4-3: 所有寄存器的初始化状态 (续)

寄存器	适用器件				上电复位和 欠压复位	MCLR 复位 WDT 复位 RESET 指令 堆栈复位	通过 WDT 或中断唤醒器件
	2331	2431	4331	4431			
CAP3BUFH/ MAXCNTH	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CAP3BUFL/ MAXCNL	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CAP1CON	2331	2431	4331	4431	-0-- 0000	-0-- 0000	-u-- uuuu
CAP2CON	2331	2431	4331	4431	-0-- 0000	-0-- 0000	-u-- uuuu
CAP3CON	2331	2431	4331	4431	-0-- 0000	-0-- 0000	-u-- uuuu
DFLTCON	2331	2431	4331	4431	-000 0000	-000 0000	-uuu uuuu

图注: u= 不变, x= 未知, -= 未实现位, 读作 0, q= 值取决于状态。
阴影单元格表示状态不适用于指定器件。

- 注
- 1: INTCONx 或 PIRx 寄存器中的一位或多位会受到影响 (而引起唤醒)。
 - 2: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置位时, PC 装入中断向量 (0008h 或 0018h)。
 - 3: 当芯片被中断唤醒且 GIEL 或 GIEH 位被置位时, TOSU、TOSH 和 TOSL 被 PC 的当前值更新。STKPTR 被修改为指向硬件堆栈中的下一个单元。
 - 4: 关于特定状态下的复位值, 参见表 4-2。
 - 5: 根据选择的振荡器模式, PORTA、LATA 和 TRISA 中的 bit 6 和 bit 7 被使能。如果不作为 PORTA 引脚使能, 它们将被禁止且读作 0。
 - 6: 如果 MCLR 功能被禁止, PORTE 和 LATE 中的 bit 3 将被使能。如果不作为 PORTE 引脚使能, 它们将被禁止且读作 0。当禁止 MCLR 时, 28 引脚器件的 PORTE 上只有 RE3 可用。

PIC18F2331/2431/4331/4431

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

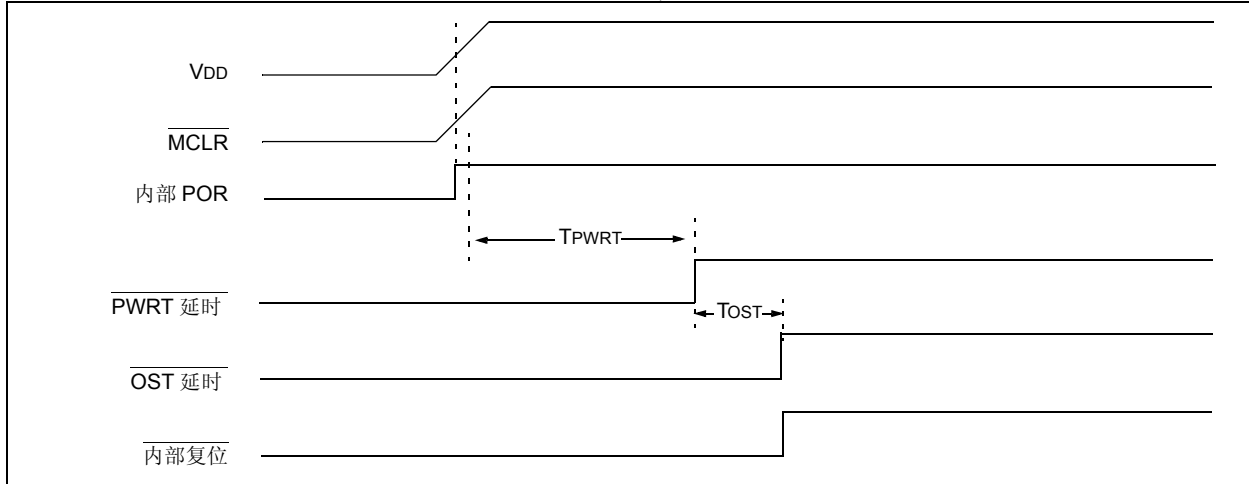


图 4-4: 上电延时时序 ($\overline{\text{MCLR}}$ 未连接到 VDD): 情形 1

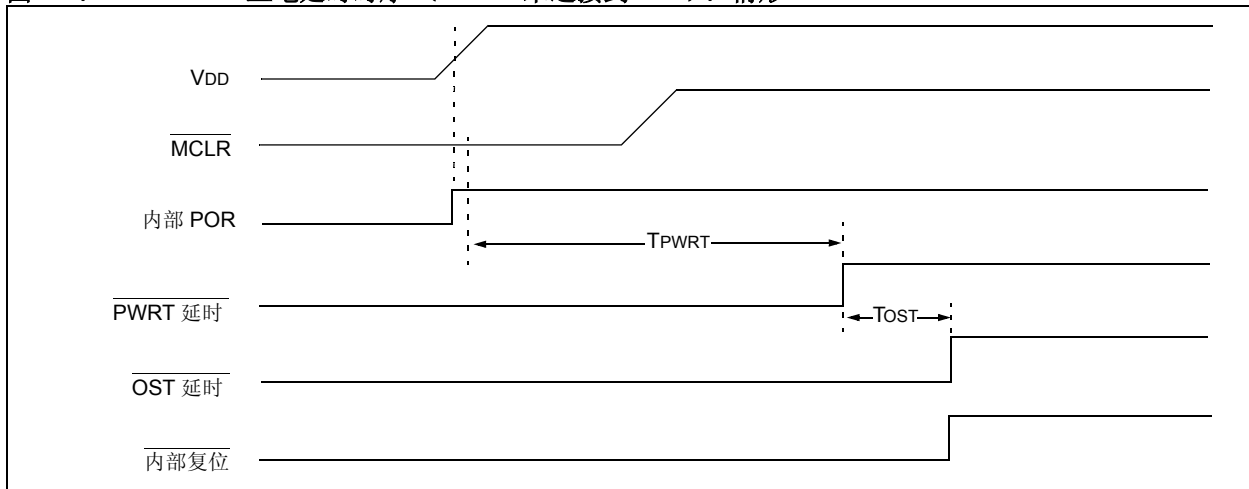
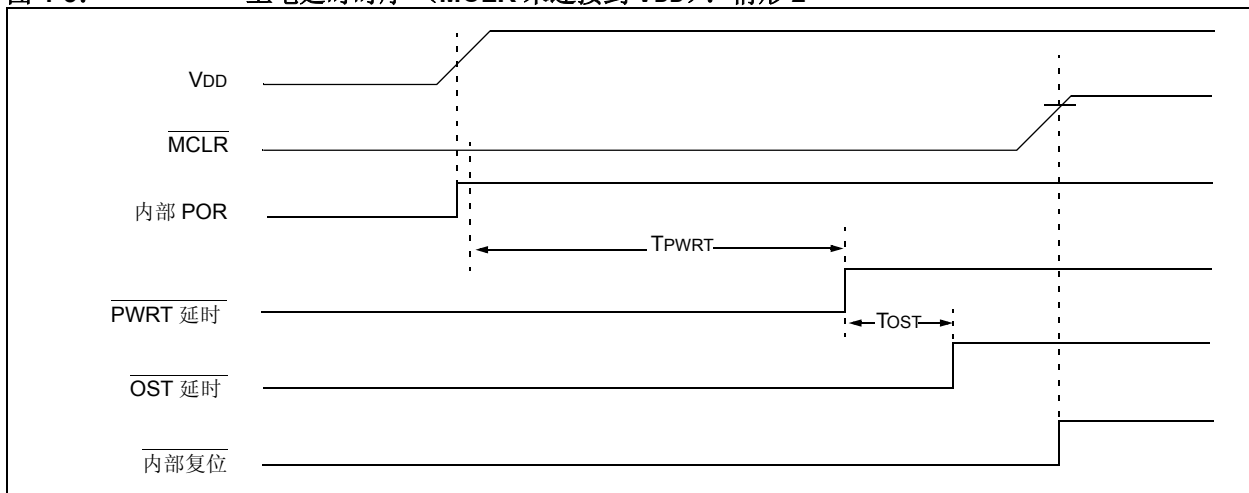


图 4-5: 上电延时时序 ($\overline{\text{MCLR}}$ 未连接到 VDD): 情形 2



PIC18F2331/2431/4331/4431

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

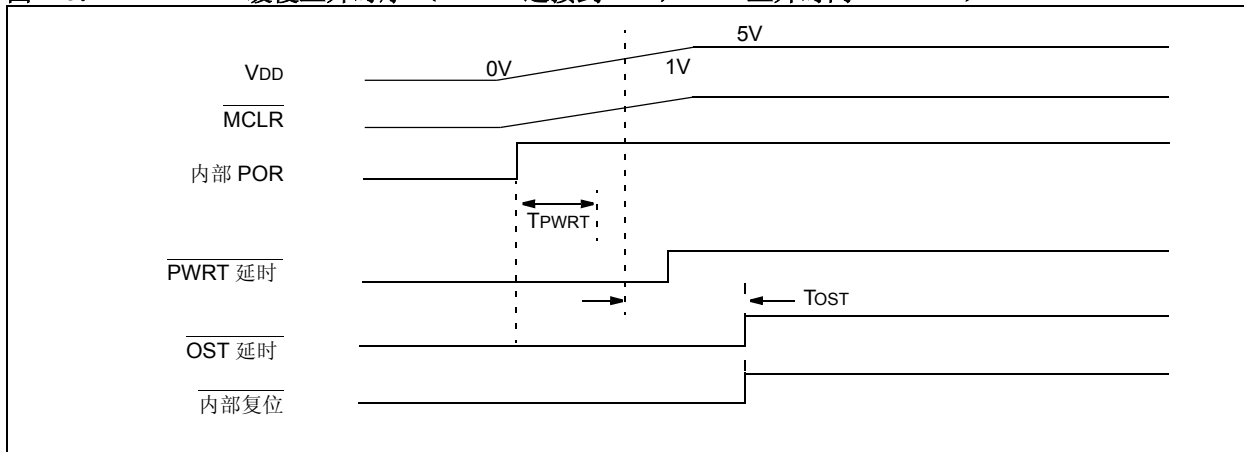
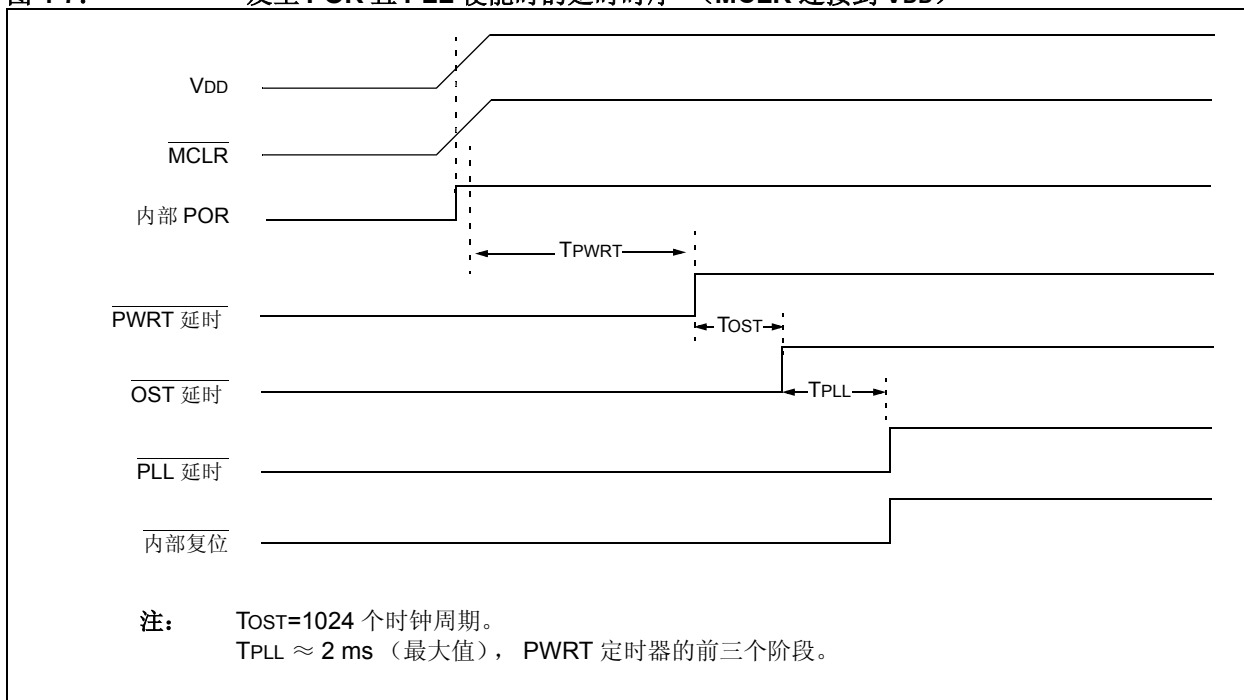


图 4-7: 发生 POR 且 PLL 使能时的延时时序 ($\overline{\text{MCLR}}$ 连接到 VDD)



PIC18F2331/2431/4331/4431

注:

PIC18F2331/2431/4331/4431

5.0 存储器构成

在增强型 MCU 器件中有三种存储器类型。它们分别是

- 程序存储器
- 数据 RAM
- 数据 EEPROM

数据和程序存储器使用不同的总线，因此允许同时访问这两种存储器。

在第 6.0 节“闪存程序存储器”和第 7.0 节“数据 EEPROM 存储器”中分别提供了更多关于闪存程序存储器和数据 EEPROM 的详细信息。

5.1 程序存储器构成

21 位的程序计数器可以寻址 2MB 的程序存储器空间。访问物理存储器和这个 2MB 地址之间的存储单元时，读取都为“0”（一条 NOP 指令）。

PIC18F2331 和 PIC18F4331 都有 8 KB 的闪存存储器，能够存储多达 4,096 条单字指令。

PIC18F2431 和 PIC18F4431 都有 16 KB 的闪存存储器，能够存储多达 8,192 条单字指令。

复位向量地址为 000000h，中断向量地址为 000008h 和 000018h。

图 5-1 和图 5-2 所示分别为 PIC18F2X31 和 PIC18F4X31 器件的程序存储器映射图和堆栈。

图 5-1: PIC18F2331/4331 的程序存储器映射图和堆栈

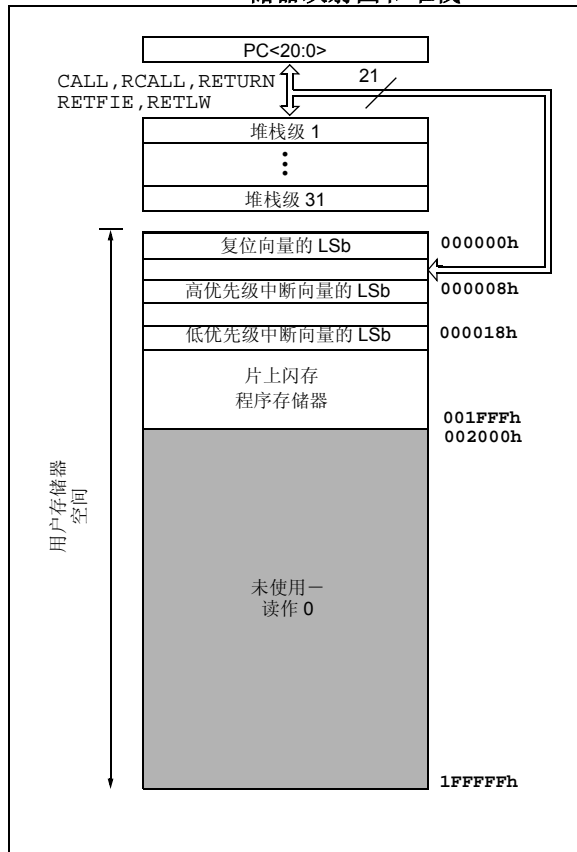
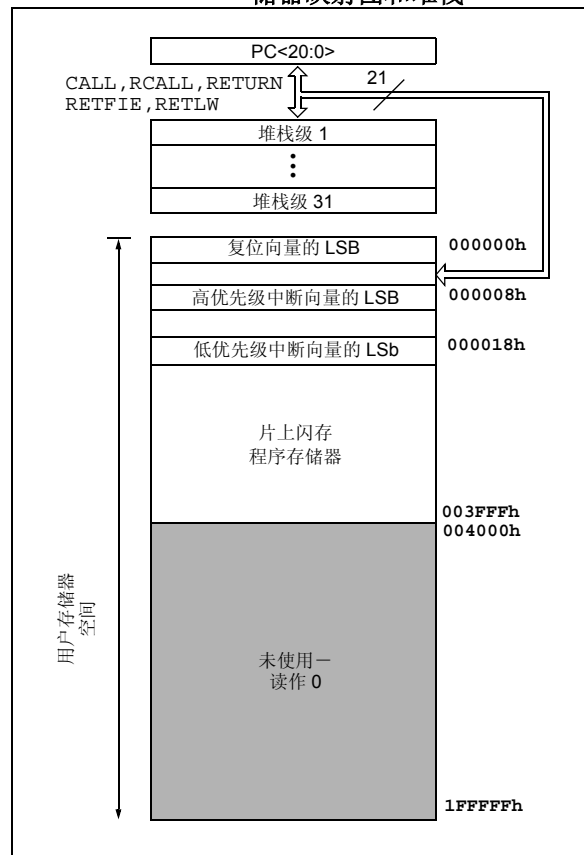


图 5-2: PIC18F2431/4431 的程序存储器映射图和堆栈



PIC18F2331/2431/4331/4431

5.2 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。当执行 CALL 或 RCALL 指令或响应中断发生跳转时，程序计数器（Program Counter, PC）的值会被压入堆栈。而执行 RETURN、RETLW 或 RETFIE 指令时，PC 值从堆栈弹出。PCLATU 和 PCLATH 不受任何 RETURN 或 CALL 指令的影响。

通过 21 位的 RAM 和一个 5 位的堆栈指针来实现 31 字的堆栈操作，在全部复位后，堆栈指针初始化为 00000b。堆栈指针 00000b 与 RAM 无关。这仅是复位值。执行 CALL 类型指令时，产生进栈操作，堆栈指针首先加 1，并且将 PC 的内容写入堆栈指针指向的 RAM 单元（PC 已指向调用后的指令）。执行 RETURN 类型指令时，产生出栈操作，STKPTR 所指向的 RAM 单元的内容会传递给 PC，堆栈指针减 1。

堆栈既不占用程序存储器空间，也不占用数据存储器空间。堆栈指针可以读写，并且通过栈顶的特殊功能寄存器可以读写栈顶地址。使用栈顶 SFR 可以将数据压入或弹出堆栈。状态位表明堆栈是满了、溢出还是下溢。

5.2.1 栈顶访问

栈顶可以读写。三个寄存器单元 TOSU、TOSH 和 TOSL 保存 STKPTR 寄存器所指向的堆栈单元的内容（图 5-3）。这可以让用户在必要时实现软件堆栈。在 CALL、RCALL 或中断后，软件可以读取 TOSU、TOSH 和 TOSL 寄存器来获取进栈值。这些值可以被置入用户定义的软件堆栈。返回时，软件可以替换 TOSU、TOSH 和 TOSL 的内容并执行返回。

为防止意外的堆栈破坏，访问堆栈时，用户必须禁止全局中断使能位。

5.2.2 返回堆栈指针（STKPTR）

STKPTR 寄存器（寄存器 5-1）包含堆栈指针值、STKFUL（堆栈满）状态位和 STKUNF（堆栈下溢）状态位。堆栈指针值是 0 到 31。堆栈压入值前，堆栈指针加 1，堆栈弹出值后，堆栈指针减 1。复位时，堆栈指针值是 0。用户可以读写堆栈指针的值。在实时操作系统可以利用此特性来维护返回堆栈。

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

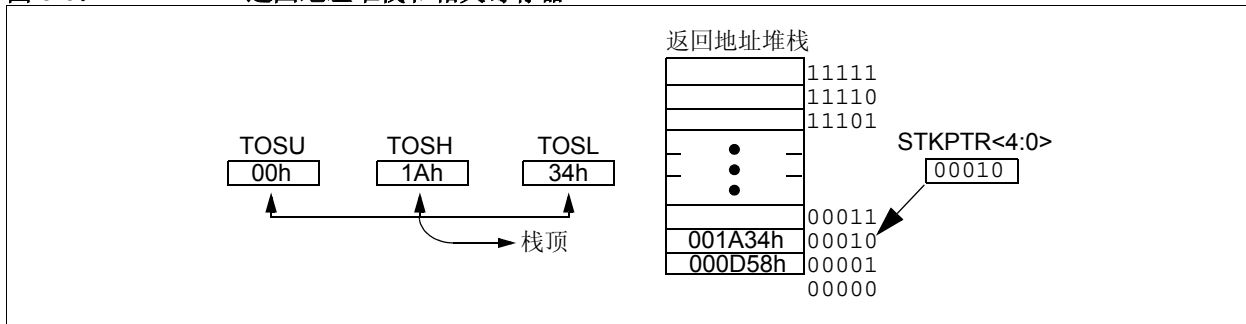
堆栈满时执行的操作由 STVREN（堆栈溢出复位使能，Stack Overflow Reset Enable）配置位的状态决定。（有关器件配置位的介绍，请参阅第 22.1 节“配置位”。）如果 STVREN 位已置位（缺省），第 31 次进栈将把（PC+2）值压入堆栈，将 STKFUL 位置位，并复位器件。STKFUL 位将保持置位，而堆栈指针将被置 0。

如果 STVREN 位被清零，第 31 次进栈时 STKFUL 位会被置位，堆栈指针则递增至 31。任何其他进栈都不会覆盖第 31 次的进栈值，并且 STKPTR 将保持为 31。

当堆栈弹出次数足够清空堆栈时，下一次出栈会向 PC 返回一个零值，并将 STKUNF 位置位，而堆栈指针则保持为 0。STKUNF 位将保持置位状态，直到软件清零或发生上电复位。

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

图 5-3: 返回地址堆栈和相关寄存器



PIC18F2331/2431/4331/4431

寄存器 5-1:

STKPTR 寄存器

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

- bit 7⁽¹⁾ **STKFUL:** 堆栈满标志位
1= 堆栈满或溢出
0= 堆栈未滿或未溢出
- bit 6⁽¹⁾ **STKUNF:** 堆栈下溢标志位
1= 发生堆栈下溢
0= 未发生堆栈下溢
- bit 5 **未实现位:** 读作 0
- bit 4-0 **SP4:SP0:** 堆栈指针位置位

注 1: 通过用户软件或上电复位清零 bit 7 和 bit 6。

图注:			
R= 可读位	W= 可写位	U= 未实现位	C= 只可清零位
-n= 上电复位时的值	1= 置位	0= 清零	x= 未知

5.2.3 PUSH 和 POP 指令

因为栈顶 (Top-of-Stack, TOS) 可以读写, 所以将值压入堆栈或从堆栈弹出而不影响程序的正常执行的功能是一个很好的选择。要将当前 PC 值压入堆栈, 可以执行 PUSH 指令。这将使堆栈指针加 1, 并将当前 PC 值装入堆栈。然后就可以修改 TOSU、TOSH 和 TOSL, 将数据或返回地址放到堆栈中。

使用 POP 指令可以从堆栈中弹出 TOS 的值, 并用前一个入栈值来替换 TOS 值, 而不会影响程序的正常执行。POP 指令通过将堆栈指针减 1 来丢弃当前的 TOS。然后前一个入栈值就成为 TOS 值。

5.2.4 堆栈满 / 下溢复位

通过对配置寄存器 4L 中的 STVREN 位编程可以使能这些复位。当 STVREN 位清零时, 堆栈满或堆栈下溢的情况会将相应的 STKFUL 或 STKUNF 位置位, 但不会使器件复位。当 STVREN 位置位时, 堆栈满或堆栈下溢的状态会将相应的 STKFUL 或 STKUNF 位置位, 然后使器件复位。通过用户软件或上电复位清零 STKFUL 或 STKUNF 位。

PIC18F2331/2431/4331/4431

5.3 快速寄存器堆栈

中断可以使用“快速返回”选项。为 Status、WREG 和 BSR 寄存器提供了快速寄存器堆栈，其深度仅为 1。此堆栈不可读写，当处理器用于中断时，此堆栈装入对应寄存器的当前值。如果使用 RETFIE，FAST 指令从中断返回，这些寄存器中的值会装回至工作寄存器。

所有中断源都会将值压入堆栈寄存器。如果同时使能了低优先级中断和高优先级中断，从低优先级中断返回时，无法可靠地使用堆栈寄存器。如果在为低优先级中断提供服务时，发生了高优先级中断，则低优先级中断存储的堆栈寄存器值将被覆盖。在低优先级中断过程中，用户必须用软件保护关键寄存器。

如果未使用中断优先级，所有中断都可以使用快速寄存器堆栈从中断返回。

如果没有使用中断，快速寄存器堆栈可以用于在子程序调用结束时恢复 Status、WREG 和 BSR 寄存器。要将快速寄存器堆栈用于子程序调用，必须执行 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 指令时，从快速寄存器堆栈恢复这些寄存器的值。

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

例 5-1: 快速寄存器堆栈代码示例

```
CALL SUB1, FAST      ;STATUS, WREG, BSR
                    ;SAVED IN FAST REGISTER
                    ;STACK
    .
    .
SUB1  .
    .
    RETURN FAST      ;RESTORE VALUES SAVED
                    ;IN FAST REGISTER STACK
```

5.4 PCL、PCLATH 和 PCLATU

程序计数器 (PC) 用于指定要取出执行指令的地址。PC 的长度为 21 位。其中的低字节称为 PCL 寄存器，该寄存器可读写。高字节，即 PCH 寄存器，包含 PC<15:8> 位，不可直接读写。但可以通过 PCLATH 寄存器更新 PCH 寄存器。更高字节称为 PCU。该寄存器包含 PC<20:16> 位，不可直接读写。但可以通过 PCLATU 寄存器更新 PCU 寄存器。

任何写 PCL 的操作都将使 PCLATH 和 PCLATU 的内容传送到程序计数器。类似地，读 PCL，程序计数器的高两位字节将被传送到 PCLATH 和 PCLATU。这对于计算 PC 偏移量很有用（见第 5.8.1 节“计算 goto”）。

PC 在程序存储器中按字节寻址。为防止 PC 偏离字指令，PCL 的 LSB 值固定为 0。PC 在程序存储器中以 2 为增量对连续指令寻址。

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。

5.5 时序图 / 指令周期

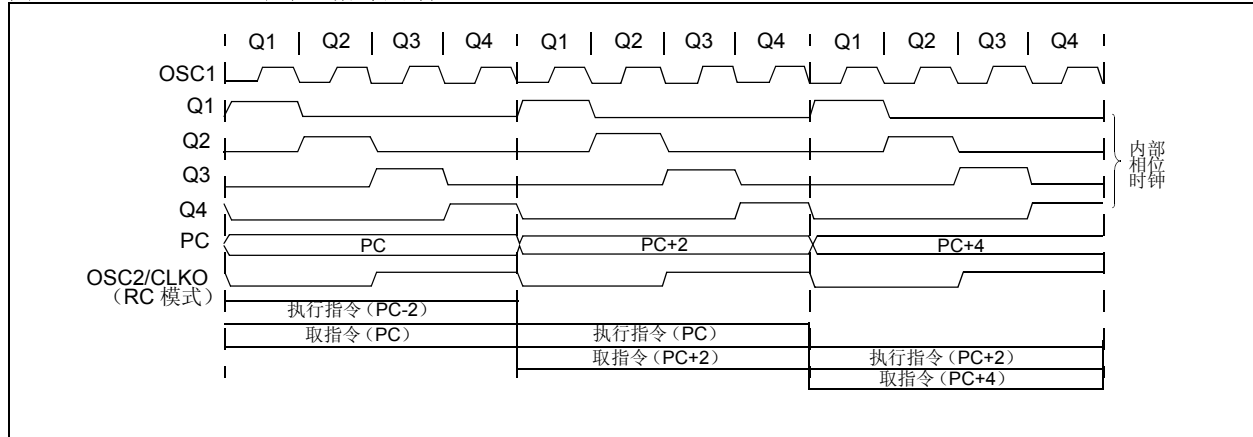
OSC1 引脚输入的时钟信号在器件内部经过 4 分频后产生 4 个非重叠的正交时钟信号，即 Q1、Q2、Q3 和 Q4。在此过程中，程序计数器（PC）在每个 Q1 时递增，在 Q4 时，从程序存储器读取指令并将指令锁存到指令寄存器中。指令的译码和执行是在下一个 Q1 到 Q4 中完成。图 5-4 所示为时钟和指令执行流程图。

5.6 指令流 / 流水线

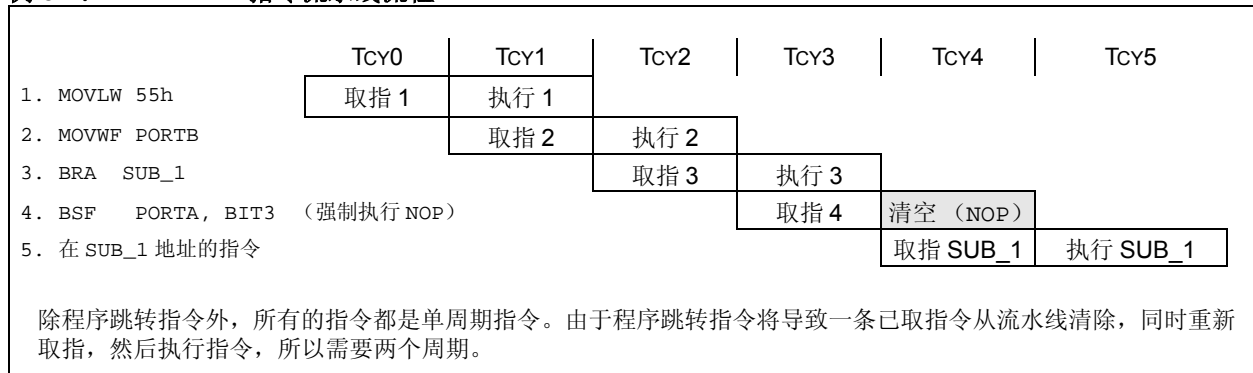
一个“指令周期”由 4 个 Q 周期组成（即 Q1、Q2、Q3 和 Q4）。取指和指令执行是流水执行的，取指需要一个指令周期，而译码和执行需要另一个指令周期。但由于是流水线操作，每条指令的等效执行时间为一个指令周期。如果某条指令改变了程序计数器（如 GOTO 指令），则需要两个指令周期才能完成该指令（见例 5-2）。

在 Q1 周期，开始取指周期，程序计数器（PC）递增。在执行周期，所取指令在 Q1 周期中被锁存到指令寄存器（Instruction Register, IR）。在随后的 Q2、Q3 和 Q4 周期中译码并执行该指令。其中读数据存储器（读操作数）发生在 Q2 周期，写操作发生在 Q4 周期（写目标）。

图 5-4: 时钟 / 指令周期



例 5-2: 指令流水线流程



PIC18F2331/2431/4331/4431

5.7 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。指令字的低有效字节始终存储在偶数地址的程序存储单元中 (LSB=0)。图 5-5 给出了指令字存储在程序存储器中的示例。要保持与指令边界对齐, PC 以 2 为单位递增, 并且 LSB 总是读作 0 (见第 5.4 节“PCL、PCLATH 和 PCLATU”)。

CALL和GOTO指令在指令中嵌入了程序存储器的绝对地址。指令总是存储在字边界, 因而指令所包含的数据为字地址。字地址会写入 PC<20:1>, 它可以在程序存储器中访问所需的字节地址。图 5-5 中的指令 2 说明了指令“GOTO 000006h”在程序存储器中是如何进行编码的。程序跳转指令也采取同样的方式操作, 对相对地址偏移量进行编码。在跳转指令中的偏移值代表单字指令数, PC 将以此作为偏移量。第 23.0 节“指令集综述”提供了指令集的更多详情。

图 5-5: 程序存储器中的指令

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

5.7.1 双字指令

PIC18F2331/2431/4331/4431 器件有 4 个双字指令: MOVFF、CALL、GOTO 和 LFSR。这些指令第二个字的 4 个 MSB 均置为“1”, 并译码为 NOP 指令。第二个字的低 12 位则包含指令要使用的数据。如果执行指令的第一个字, 则访问指令第二个字中的数据。如果自行执

行指令的第二个字 (跳过了第一个字), 则其效果相当于 NOP 指令。如果双字指令跟在导致跳过操作的条件指令后, 就有必要执行此操作。例 5-3 中的程序示例说明此概念。如需了解指令集的更多详情, 请参阅第 23.0 节“指令集综述”。

例 5-3: 双字指令

情形 1:	
目标代码	源代码
0110 0110 0000 0000	TSTFSZ REG1 ; is RAM location 0?
1100 0001 0010 0011	MOVFF REG1, REG2 ; No, skip this word
1111 0100 0101 0110	; Execute this word as a NOP
0010 0100 0000 0000	ADDWF REG3 ; continue code
情形 2:	
目标代码	源代码
0110 0110 0000 0000	TSTFSZ REG1 ; is RAM location 0?
1100 0001 0010 0011	MOVFF REG1, REG2 ; Yes, execute this word
1111 0100 0101 0110	; 2nd word of instruction
0010 0100 0000 0000	ADDWF REG3 ; continue code

5.8 查找表

实现查找表有两种方法：

- 计算 GOTO
- 读表

5.8.1 计算 GOTO

通过向程序计数器加一个偏移量来计算 GOTO。例 5-4 中给出了这样的示例。

使用 ADDWF PCL 指令和一组 RETLW 0xnn 指令可以组成一个查找表。在调用该表前，会先将偏移量装入 WREG 中。被调用子程序的第一条指令是 ADDWF PCL。接下来执行的是 RETLW 0xnn 指令，它将向调用函数返回值 0xnn。

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

在这种方法中，每个指令单元只能存储一个数据字节，并且要求返回地址堆栈未滿。

例 5-4: 计算 GOTO 使用偏移量

```
MOVFWOFFSET
CALLTABLE
ORG 0xnn00
TABLEADDWFPCL
RETLW0xnn
RETLW0xnn
RETLW0xnn
.
.
.
```

5.8.2 读表 / 写表

将数据存储的程序存储器的更好方法允许在每个指令单元存储 2 个字节的数据。

通过读 / 写表，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）则包含从程序存储器读取或写入程序存储器的数据。进出程序存储器的数据每次为一个字节。

第 6.1 节“读表与写表”中有关于读表 / 写表操作的进一步讨论。

5.9 数据存储器的构成

数据存储以静态 RAM 实现。在数据存储器中，每个寄存器有 12 位地址，其空间可达 4096 个字节。图 5-6 所示为 PIC18F2331/2431/4331/4431 器件的数据存储器构成。

数据存储器分为 16 个存储区，每个存储区包含 256 字节。存储区选择寄存器（Bank Select Register, BSR）的低 4 位（BSR<3:0>）选择将要访问的存储区。BSR 的高 4 位是未实现位。

数据存储器由特殊功能寄存器（Special Function Register, SFR）和通用寄存器（General Purpose Register, GPR）组成。SFR 用于单片机和外设功能的控制和状态显示，GPR 则用于在用户应用程序中存储数据和高速暂存操作。SFR 从存储区 15 的最末单元（FFFh）开始并扩展到 F60h。存储区中 SFR 以外的所有剩余空间都可以用做 GPR。GPR 从存储区 0 的首单元开始，并向上分布。读取任何未实现单元时，将读作“0”。

整个数据存储器可以采用直接寻址或间接寻址来访问。直接寻址可能需要使用 BSR 寄存器。间接寻址需要使用文件选择寄存器（File Select Register, FSRn）和相应的间接文件操作数（Indirect File Operand, INDFn）。每个 FSR 保存一个 12 位的地址值，使用该值无需选择存储区即可访问数据存储器映射的任何单元。有关间接寻址的详细信息，请参阅第 5.12 节“间接寻址、INDF 和 FSR 寄存器”。

此指令集和架构支持对所有存储区的操作。可以通过间接寻址或使用 MOVFF 指令实现。MOVFF 指令是一条双字 / 双周期指令，它将值从一个寄存器移到另一个寄存器。

无论当前 BSR 值如何，要确保能在一个周期访问常用寄存器（SFR 和所选的 GPR），需要实现一个快速访问存储区（Access Bank）。存储区 0 的一段和存储区 15 的一段构成了快速存取 RAM（Access RAM）。第 5.10 节“快速访问存储区”给出了快速存取 RAM 的详细说明。

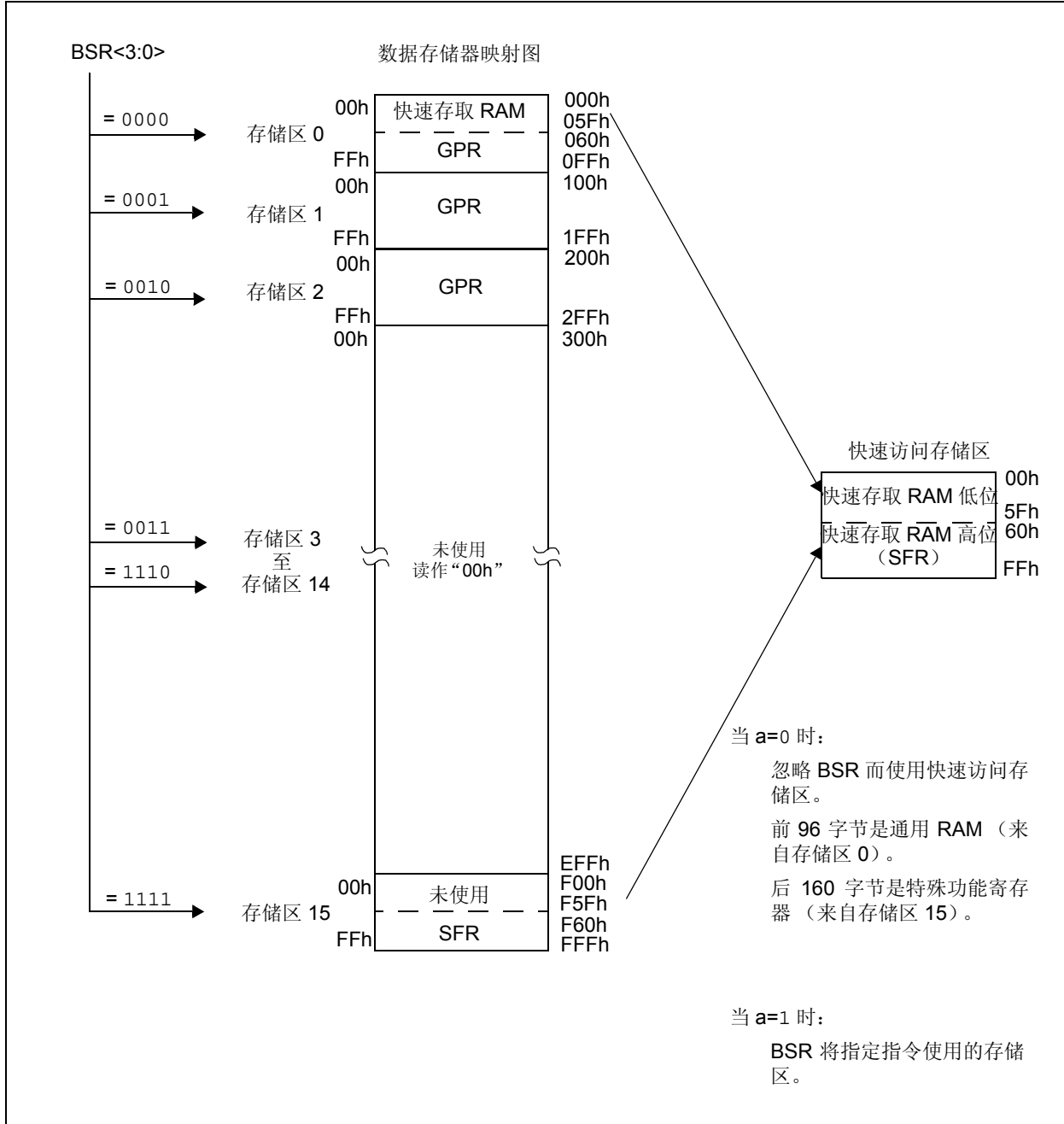
5.9.1 通用寄存器文件

增强型 MCU 器件可能在 GRP 区有连续的存储区。上电复位不会将 GPR 初始化，且其他复位也不会改变其内容。

所有指令都可以使用数据 RAM 作为 GPR 寄存器。SFR 都包含在存储区 15 的后半部分（F60h 至 FFFh）。数据存储器的所有其他存储区，从存储区 0 开始包含 GPR 寄存器。

PIC18F2331/2431/4331/4431

图 5-6: PIC18F2331/2431/4331/4431 器件的数据存储器映射图



PIC18F2331/2431/4331/4431

5.9.2 特殊功能寄存器

特殊功能寄存器（SFR）是 CPU 和外设模块用来控制器件操作的寄存器。这些寄存器以静态RAM实现。表 5-1 和表 5-2 列出了这些寄存器。

SFR 可分为两类，一类与内核功能有关，另一类与外设功能有关。本节将讲述与内核功能有关的特殊功能寄存器，而与外设功能操作有关的特殊功能寄存器将在相应的外设功能模块章节中讲述。

SFR 常分布在外设中，用来控制外设功能模块。

未使用的 SFR 单元将不会实现，并读作 0。

表 5-1: PIC18F2331/2431/4331/4431 器件的特殊功能寄存器映射图

地址	名称	地址	名称	地址	名称	地址	名称	地址	名称
FFFh	TOSU	FDFh	INDF2	FBFh	CCPR1H	F9Fh	IPR1	F7Fh	PTCON0
FFEh	TOSH	FDEh	POSTINC2	FBEh	CCPR1L	F9Eh	PIR1	F7Eh	PTCON1
FFDh	TOSL	FDDh	POSTDEC2	FBDh	CCP1CON	F9Dh	PIE1	F7Dh	PTMRL
FFCh	STKPTR	FDCCh	PREINC2	FBCh	CCPR2H	F9Ch	—	F7Ch	PTMRH
FFBh	PCLATU	FDBh	PLUSW2	FBHh	CCPR2L	F9Bh	OSCTUNE	F7Bh	PTPERL
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	ADCON3	F7Ah	PTPERH
FF9h	PCL	FD9h	FSR2L	FB9h	ANSEL1	F99h	ADCHS	F79h	PDC0L
FF8h	TBLPTRU	FD8h	STATUS	FB8h	ANSEL0	F98h	—	F78h	PDC0H
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	T5CON	F97h	—	F77h	PDC1L
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	QEICON	F96h	TRISE	F76h	PDC1H
FF5h	TABLAT	FD5h	T0CON	FB5h	—	F95h	TRISD	F75h	PDC2L
FF4h	PRODH	FD4h	—	FB4h	—	F94h	TRISC	F74h	PDC2H
FF3h	PRODL	FD3h	OSCCON	FB3h	—	F93h	TRISB	F73h	PDC3L
FF2h	INTCON	FD2h	LVDCON	FB2h	—	F92h	TRISA	F72h	PDC3H
FF1h	INTCON2	FD1h	WDTCON	FB1h	—	F91h	PR5H	F71h	SEVTCMPL
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRGH	F90h	PR5L	F70h	SEVTCMPH
FEFh	INDF0	FCFh	TMR1H	FAFh	SPBRG	F8Fh	—	F6Fh	PWMCON0
FEEh	POSTINC0	FCEh	TMR1L	FAEh	RCREG	F8Eh	—	F6Eh	PWMCON1
FEDh	POSTDEC0	FCDh	T1CON	FADh	TXREG	F8Dh	LATE	F6Dh	DTCON
FECh	PREINC0	FCCh	TMR2	FACH	TXSTA	F8Ch	LATD	F6Ch	FLTCONFIG
FEBh	PLUSW0	FCBh	PR2	FABh	RCSTA	F8Bh	LATC	F6Bh	OVDCOND
FEAh	FSR0H	FCAh	T2CON	FAAh	BAUDCTL	F8Ah	LATB	F6Ah	OVDCONS
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA	F69h	CAP1BUFH
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	TMR5H	F68h	CAP1BUFL
FE7h	INDF1	FC7h	SSPSTAT	FA7h	EECON2	F87h	TMR5L	F67h	CAP2BUFH
FE6h	POSTINC1	FC6h	SSPCON	FA6h	EECON1	F86h	—	F66h	CAP2BUFL
FE5h	POSTDEC1	FC5h	—	FA5h	IPR3	F85h	—	F65h	CAP3BUFH
FE4h	PREINC1	FC4h	ADRESH	FA4h	PIR3	F84h	PORTE	F64h	CAP3BUFL
FE3h	PLUSW1	FC3h	ADRESL	FA3h	PIE3	F83h	PORTD	F63h	CAP1CON
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	CAP2CON
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	CAP3CON
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA	F60h	DFLTCON

PIC18F2331/2431/4331/4431

表 5-2: 寄存器文件综述 (PIC18F2331/2431/4331/4431)

文件名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 与 BOR 时的值	详情请见: (页)
TOSU	—	—	—	栈顶更高字节 (TOS<20:16>)					---0 0000	48, 58
TOSH	栈顶高字节 (TOS<15:8>)								0000 0000	48, 58
TOSL	栈顶低字节 (TOS<7:0>)								0000 0000	48, 58
STKPTR	STKFUL	STKUNF	—	返回堆栈指针					00-0 0000	48, 59
PCLATU	—	—	bit 21 ⁽³⁾	PC<20:16> 的保持寄存器					---0 0000	48, 60
PCLATH	PC<15:8> 的保持寄存器								0000 0000	48, 60
PCL	PC 的低字节 (PC<7:0>)								0000 0000	48, 60
TBLPTRU	—	—	bit 21 ⁽³⁾	程序存储器表指针更高字节 (TBLPTR<20:16>)					--00 0000	48, 78
TBLPTRH	程序存储器表指针高字节 (TBLPTR<15:8>)								0000 0000	48, 78
TBLPTRL	程序存储器表指针低字节 (TBLPTR<7:0>)								0000 0000	48, 78
TABLAT	程序存储器表锁存器								0000 0000	48, 78
PRODH	乘积寄存器高字节								xxxx xxxx	48, 89
PRODL	乘积寄存器低字节								xxxx xxxx	48, 89
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0F	RBIF	0000 000x	48, 93
INTCON2	RBPV	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	1111 -1-1	48, 94
INTCON3	INT2P	INT1P	—	INT2IE	INT1IE	—	INT2IF	INT1IF	11-0 0-00	48, 95
INDF0	使用 FSR0 的内容寻址数据存储器, FSR0 的值不改变 (不是物理寄存器)								N/A	48, 71
POSTINC0	使用 FSR0 的内容寻址数据存储器, FSR0 的值后递增 (不是物理寄存器)								N/A	48, 71
POSTDEC0	使用 FSR0 的内容寻址数据存储器, FSR0 的值后递减 (不是物理寄存器)								N/A	48, 71
PREINC0	使用 FSR0 的内容寻址数据存储器, FSR0 的值预递增 (不是物理寄存器)								N/A	48, 71
PLUSW0	使用 FSR0 的内容寻址数据存储器, FSR0 的值按 W 中的值偏移 (不是物理寄存器)								N/A	48, 71
FSR0H	—	—	—	—	间接数据存储器地址指针 0 的高位字节				---- 0000	48, 71
FSR0L	间接数据存储器地址指针 0 的低位字节								xxxx xxxx	48, 71
WREG	工作寄存器								xxxx xxxx	48
INDF1	使用 FSR1 的内容寻址数据存储器, FSR1 的值不改变 (不是物理寄存器)								N/A	48, 71
POSTINC1	使用 FSR1 的内容寻址数据存储器, FSR1 的值后递增 (不是物理寄存器)								N/A	48, 71
POSTDEC1	使用 FSR1 的内容寻址数据存储器, FSR1 的值后递减 (不是物理寄存器)								N/A	48, 71
PREINC1	使用 FSR1 的内容寻址数据存储器, FSR1 的值预递增 (不是物理寄存器)								N/A	48, 71
PLUSW1	使用 FSR1 的内容寻址数据存储器, FSR1 的值按 W 中的值偏移 (不是物理寄存器)								N/A	48, 71
FSR1H	—	—	—	—	间接数据存储器地址指针 1 的高位字节				---- 0000	49, 71
FSR1L	间接数据存储器地址指针 1 的低位字节								xxxx xxxx	49, 71
BSR	—	—	—	—	存储区选择寄存器				---- 0000	49, 70
INDF2	使用 FSR2 的内容寻址数据存储器, FSR2 的值不改变 (不是物理寄存器)								N/A	49, 71
POSTINC2	使用 FSR2 的内容寻址数据存储器, FSR2 的值后递增 (不是物理寄存器)								N/A	49, 71
POSTDEC2	使用 FSR2 的内容寻址数据存储器, FSR2 的值后递减 (不是物理寄存器)								N/A	49, 71
PREINC2	使用 FSR2 的内容寻址数据存储器, FSR2 的值预递增 (不是物理寄存器)								N/A	49, 71
PLUSW2	使用 FSR2 的内容寻址数据存储器, FSR2 的值按 W 中的值偏移 (不是物理寄存器)								N/A	49, 71
FSR2H	—	—	—	—	间接数据存储器地址指针 2 的高位字节				---- 0000	49, 71
FSR2L	间接数据存储器地址指针 2 的低位字节								xxxx xxxx	49, 71
STATUS	—	—	—	N	OV	Z	DC	C	---x xxxxx	49, 73
TMR0H	Timer0 寄存器高位字节								0000 0000	49, 135
TMR0L	Timer0 寄存器低位字节								xxxx xxxx	49, 135
T0CON	TMR0ON	T016BIT	—	—	T0PS3	T0PS2	T0PS1	T0PS0	11-- 1111	49, 133

- 图注: x= 未知, u= 不变, -= 未实现, q= 值取决于条件
- 注 1: 仅在 RCIO、ECIO 和 INTIO2 (RA6 使能 I/O 端口引脚功能) 振荡模式下, RA6 和相关位被配置为端口引脚, 在所有其他振荡模式下均读作 0。
- 2: 仅在 INTIO2 振荡模式下, RA7 和相关位被配置为端口引脚, 在所有其他振荡模式下均读作 0。
- 3: PC 的 bit 21 仅在测试模式和串行编程模式下可用。
- 4: 如果 PBADEN=0, PORTB<4:0> 被配置为数字输入且读作未知。而如果 PBADEN=1, 则 PORTB<4:0> 被配置为模拟输入且在复位后读作 0。
- 5: 这些寄存器和 / 或在 PIC18F2X31 器件上未实现, 且读作 0。
- 6: 仅当 MCLR 熔丝 (CONFIG3H<7>) 被编程为 0 时, RE3 端口位可用。否则, RE3 读作 0。该位只读。

PIC18F2331/2431/4331/4431

表 5-2: 寄存器文件综述 (PIC18F2331/2431/4331/4431) (续)

文件名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 与 BOR 时的值	详情请见: (页)
OSCCON	IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0	0000 q000	28, 49
LVDCON	—	—	IVRST	LVDEN	LVDL3	LVDL2	LVDL1	LVDL0	--00 0101	49, 263
WDTCON	WDTW	—	—	—	—	—	—	SWDTEN	0000 0000	49, 279
RCON	IPEN	—	—	RI	TO	PD	POR	BOR	0--1 11qq	47, 74, 105
TMR1H	Timer1 寄存器高字节								xxxx xxxx	49, 141
TMR1L	Timer1 寄存器低字节								xxxx xxxx	49, 141
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	49, 137
TMR2	Timer2 寄存器								0000 0000	49, 143
PR2	Timer2 周期寄存器								1111 1111	49, 143
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	49, 143
SSPBUF	SSP 接收缓冲器 / 发送寄存器								xxxx xxxx	49, 220
SSPADD	I ² C 从动模式下的 SSP 地址寄存器。I ² C 主动模式下的 SSP 波特率重载寄存器。								0000 0000	49, 220
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	49, 212
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	49, 213
ADRESH	A/D 结果寄存器高字节								xxxx xxxx	50, 259
ADRESL	A/D 结果寄存器低字节								xxxx xxxx	50, 259
ADCON0	—	—	ACONV	ACSCH	ACMOD1	ACMOD0	GO/DONE	ADON	--00 0000	50, 244
ADCON1	VCFG1	VCFG0	—	FIFOEN	BFEMT	FFOVFL	ADPNT1	ADPNT0	00-0 1000	50, 245
ADCON2	ADFM	ACQT3	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0000 0000	50, 246
ADCON3	ADRS1	ADRS0	—	SSRC4	SSRC3	SSRC2	SSRC1	SSRC0	00-0 0000	51, 247
ADCSH	GDSEL1	GDSEL0	GBSEL1	GBSEL0	GCSEL1	GCSEL0	GASEL1	GASEL0	0000 0000	51, 248
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 高字节								xxxx xxxx	50, 152
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 低字节								xxxx xxxx	50, 152
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	50, 155, 149
CCPR2H	捕捉 / 比较 / PWM 寄存器 2 高字节								xxxx xxxx	50, 152
CCPR2L	捕捉 / 比较 / PWM 寄存器 2 低字节								xxxx xxxx	50, 152
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	50, 155
ANSEL1	—	—	—	—	—	—	—	ANS8	---- --1	50, 249
ANSEL0	ANS7 ⁽⁶⁾	ANS6 ⁽⁶⁾	ANS5 ⁽⁶⁾	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	50, 249
T5CON	T5SEN	RESEN ⁽⁵⁾	T5MOD	T5PS1	T5PS0	T5SYNC	TMR5CS	TMR5ON	0100 0000	50, 145
QEICON	VELM	ERROR	UP/DOWN	QEIM2	QEIM1	QEIM0	PDEC1	PDEC0	0000 0000	50, 171
SPBRGH	波特率发生器寄存器的高字节								0000 0000	50, 225
SPBRG	USART 波特率发生器								0000 0000	50, 225
RCREG	USART 接收寄存器								0000 0000	50, 233, 232
TXREG	USART 发送寄存器								0000 0000	50, 230, 232
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	50, 222
RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	50, 223
BAUDCTL	—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-1 0-00	50, 224

图注: x= 未知, u= 不变, -= 未实现, q= 值取决于条件

注 1: 仅在 RCIO、ECIO 和 INTIO2 (RA6 使能 I/O 端口引脚功能) 振荡模式下, RA6 和相关位被配置为端口引脚, 在所有其他振荡模式下均读作 0。

2: 仅在 INTIO2 振荡模式下, RA7 和相关位被配置为端口引脚, 在所有其他振荡模式下均读作 0。

3: PC 的 bit 21 仅在测试模式和串行编程模式下可用。

4: 如果 PBaden=0, PORTB<4:0> 被配置为数字输入且读作未知。而如果 PBaden=1, 则 PORTB<4:0> 被配置为模拟输入且在复位后读作 0。

5: 这些寄存器和 / 或在 PIC18F2X31 器件上未实现, 且读作 0。

6: 仅当 MCLRE 熔丝 (CONFIG3H<7>) 被编程为 0 时, RE3 端口位可用。否则, RE3 读作 0。该位只读。

PIC18F2331/2431/4331/4431

表 5-2: 寄存器文件综述 (PIC18F2331/2431/4331/4431) (续)

文件名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 与 BOR 时的值	详情请见: (页)
EEADR	EEPROM 地址寄存器								0000 0000	50, 85
EEDATA	EEPROM 数据寄存器								0000 0000	50, 88
EECON2	EEPROM 控制寄存器 2 (不是物理寄存器)								0000 0000	50, 76, 85
EECON1	EEPGD	CFGFS	—	FREE	WRERR	WREN	WR	RD	xx-0 x000	50, 77, 86
IPR3	—	—	—	PTIP	IC3DRIP	IC2QEIP	IC1IP	TMR5IP	---1 1111	50
PIR3	—	—	—	PTIF	IC3DRIF	IC2QEIF	IC1IF	TMR5IF	---0 0000	50
PIE3	—	—	—	PTIE	IC3DRIE	IC2QEIE	IC1IE	TMR5IE	---0 0000	50
IPR2	OSFIP	—	—	EEIP	—	LVDIP	—	CCP2IP	1--1 -1-1	51, 103
PIR2	OSFIF	—	—	EEIF	—	LVDIF	—	CCP2IF	0--0 -0-0	51, 97
PIE2	OSFIE	—	—	EEIE	—	LVDIE	—	CCP2IE	0--0 -0-0	51, 100
IPR1	—	ADIP	RCIP	TXIP	SSIP	CCP1IP	TMR2IP	TMR1IP	-111 1111	51, 102
PIR1	—	ADIF	RCIF	TXIF	SSIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	51, 96
PIE1	—	ADIE	RCIE	TXIE	SSIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	51, 99
OSCTUNE	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	--00 0000	25, 51
ADCON3	ADRS1	ADRS0	—	SSRC4	SSRC3	SSRC2	SSRC1	SSRC0	00-0 0000	50
ADCHS	GDSEL1	GDSEL0	GBSEL1	GBSEL0	GCSEL1	GCSEL0	GASEL1	GASEL0	0000 0000	50
TRISE ⁽⁵⁾	—	—	—	—	—	PORTE 的数据方向位 ⁽⁵⁾			---- -111	51, 131
TRISD ⁽⁵⁾	PORTD 的数据方向控制寄存器								1111 1111	51, 128
TRISC	PORTC 的数据方向控制寄存器								1111 1111	51, 123
TRISB	PORTB 的数据方向控制寄存器								1111 1111	51, 117
TRISA	TRISA7 ⁽²⁾	TRISA6 ⁽¹⁾	PORTA 的数据方向控制寄存器						1111 1111	51, 111
PR5H	Timer5 周期寄存器高字节								1111 1111	50
PR5L	Timer5 周期寄存器低字节								1111 1111	50
LATE ⁽⁵⁾	—	—	—	—	—	读 / 写 PORTE 数据锁存器			---- -xxx	51, 132
LATD ⁽⁵⁾	读 / 写 PORTD 数据锁存器								xxxx xxxx	51, 128
LATC	读 / 写 PORTC 数据锁存器								xxxx xxxx	51, 123
LATB	读 / 写 PORTB 数据锁存器								xxxx xxxx	51, 117
LATA	LATA<7> ⁽²⁾	LATA<6> ⁽¹⁾	读 / 写 PORTA 数据锁存器						xxxx xxxx	51, 111
TMR5H	Timer5 定时器寄存器高字节								xxxx xxxx	146
TMR5L	Timer5 定时器寄存器低字节								xxxx xxxx	146
PORTE	—	—	—	—	RE3 ⁽⁶⁾	读 PORTE 引脚, 写 PORTE 数据锁存器 ⁽⁵⁾			---- xxxx	51, 132
PORTD	读 PORTD 引脚, 写 PORTD 数据锁存器								xxxx xxxx	51, 128
PORTC	读 PORTC 引脚, 写 PORTC 数据锁存器								xxxx xxxx	51, 123
PORTB	读 PORTB 引脚, 写 PORTB 数据锁存器 ⁽⁴⁾								xxxx xxxx	51, 117
PORTA	RA7 ⁽²⁾	RA6 ⁽¹⁾	读 PORTA 引脚, 写 PORTA 数据锁存器						xx0x 0000	51, 111
PTCON0	PTOPS3	PTOPS2	PTOPS1	PTOPS0	PTCKPS1	PTCKPS0	PTMOD1	PTMOD0	0000 0000	52, 186
PTCON1	PTEN	PTDIR	—	—	—	—	—	—	00-- ----	52, 186
PTMRL	PWM 时基寄存器 (低 8 位)								0000 0000	184
PTMRH	未使用				PWM 时基寄存器 (高 4 位)				---- 0000	184

- 图注:** x= 未知, u= 不变, -- 未实现, q= 值取决于条件
- 注**
- 1: 仅在 RCIO、ECIO 和 INTIO2 (RA6 使能 I/O 端口引脚功能) 振荡模式下, RA6 和相关位被配置为端口引脚, 在所有其他振荡模式下均读作 0。
 - 2: 仅在 INTIO2 振荡模式下, RA7 和相关位被配置为端口引脚, 在所有其他振荡模式下均读作 0。
 - 3: PC 的 bit 21 仅在测试模式和串行编程模式下可用。
 - 4: 如果 PBaden=0, PORTB<4:0> 被配置为数字输入且读作未知。而如果 PBaden=1, 则 PORTB<4:0> 被配置为模拟输入且在复位后读作 0。
 - 5: 这些寄存器和 / 或位在 PIC18F2X31 器件上未实现, 且读作 0。
 - 6: 仅当 MCLRE 熔丝 (CONFIG3H<7>) 被编程为 0 时, RE3 端口位可用。否则, RE3 读作 0。该位只读。

PIC18F2331/2431/4331/4431

表 5-2: 寄存器文件综述 (PIC18F2331/2431/4331/4431) (续)

文件名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 与 BOR 时的值	详情请见: (页)
PTPERL	PWM 时基周期寄存器 (低 8 位)								1111 1111	184
PTPERH	未使用				PWM 时基周期寄存器 (高 4 位)				---- 1111	184
PDC0L	PWM 占空比 0L 寄存器 (低 8 位)								--00 0000	184
PDC0H	未使用			PWM 占空比 0H 寄存器 (高 6 位)					0000 0000	184
PDC1L	PWM 占空比 1L 寄存器 (低 8 位)								0000 0000	184
PDC1H	未使用			PWM 占空比 1H 寄存器 (高 6 位)					--00 0000	184
PDC2L	PWM 占空比 2L 寄存器 (低 8 位)								0000 0000	184
PDC2H	未使用			PWM 占空比 2H 寄存器 (高 6 位)					--00 0000	184
PDC3L	PWM 占空比 3L 寄存器 (低 8 位)								0000 0000	184
PDC3H	未使用			PWM 占空比 3H 寄存器 (高 6 位)					--00 0000	184
SEVTCMPL	PWM 特殊事件比较寄存器 (低 8 位)								0000 0000	N/A
SEVTCMPH	未使用				PWM 特殊事件比较寄存器 (高 4 位)				---- 0000	N/A
PWMCON0	—	PWMEN2	PWMEN1	PWMEN0	PMOD3	PMOD2	PMOD1	PMOD0	-101 0000	52, 187
PWMCON1	SEVOPS3	SEVOPS2	SEVOPS1	SEVOPS0	SEVTDIR	—	UDIS	OSYNC	0000 0-00	52, 188
DTCON	DTPS1	DTPS0	DT5	DT4	DT3	DT2	DT1	DT0	0000 0000	52, 200
FLTCONFIG	—	FLTBS	FLTBMOD	FLTBEN	FLTCON	FLTAS	FLTAMOD	FLTAEN	-000 0000	52, 208
OVDCOND	POVD7	POVD6	POVD5	POVD4	POVD3	POVD2	POVD1	POVD0	1111 1111	52, 203
OVDCONS	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0	0000 0000	52, 204
CAP1BUFH/ VELRH	捕捉 1 寄存器的高字节 / 速率寄存器的高字节								xxxx xxxx	52
CAP1BUFL/ VELRL	捕捉 1 寄存器的低字节 / 速率寄存器的低字节								xxxx xxxx	52
CAP2BUFH/ POSCNTH	捕捉 2 寄存器的高字节 / QEI 位置计数器寄存器的高字节								xxxx xxxx	52
CAP2BUFL/ POSCNTL	捕捉 2 寄存器的低字节 / QEI 位置计数器寄存器的低字节								xxxx xxxx	52
CAP3BUFH/ MAXCNTH	捕捉 3 寄存器的高字节 / QEI 最大计数限度寄存器的高字节								xxxx xxxx	53
CAP3BUFL/ MAXCNTL	捕捉 3 寄存器的低字节 / QEI 最大计数限度寄存器的低字节								xxxx xxxx	53
CAP1CON	—	CAP1REN	—	—	CAP1M3	CAP1M2	CAP1M1	CAP1M0	-0-0 0000	53, 163
CAP2CON	—	CAP2REN	—	—	CAP2M3	CAP2M2	CAP2M1	CAP2M0	-0-0 0000	53, 163
CAP3CON	—	CAP3REN	—	—	CAP3M3	CAP3M2	CAP3M1	CAP3M0	-0-0 0000	53, 163
DFLTCON	—	FLT4EN	FLT3EN	FLT2EN	FLT1EN	FLTCK2	FLTCK1	FLTCK0	-000 0000	53, 178

图注: x= 未知, u= 不变, -= 未实现, q= 值取决于条件

- 注
- 1: 仅在 RCIO、ECIO 和 INTIO2 (RA6 使能 I/O 端口引脚功能) 振荡模式下, RA6 和相关位被配置为端口引脚, 在所有其他振荡模式下均读作 0。
 - 2: 仅在 INTIO2 振荡模式下, RA7 和相关位被配置为端口引脚, 在所有其他振荡模式下均读作 0。
 - 3: PC 的 bit 21 仅在测试模式和串行编程模式下可用。
 - 4: 如果 PBadEN=0, PORTB<4:0> 被配置为数字输入且读作未知。而如果 PBadEN=1, 则 PORTB<4:0> 被配置为模拟输入且在复位后读作 0。
 - 5: 这些寄存器和 / 或在 PIC18F2X31 器件上未实现, 且读作 0。
 - 6: 仅当 MCLRE 熔丝 (CONFIG3H<7>) 被编程为 0 时, RE3 端口位可用。否则, RE3 读作 0。该位只读。

PIC18F2331/2431/4331/4431

5.10 快速访问存储区

快速访问存储区是一个增强架构，对于 C 编译器的代码优化非常有用。C 编译器采用的技术对于汇编语言编写的程序也可能会有用。

此数据存储器可用于：

- 中间计算值
- 子程序的局部变量
- 变量的快速现场保存 / 切换
- 常用变量
- SFR 的快速求值 / 控制（无存储区选择）

快速访问存储区由存储区 15（SFR）的最末 160 字节和存储区 0 的首 96 字节组成。这两部分分别为快速存取 RAM 的高位和低位。图 5-6 所示为快速存取 RAM 的区域。

指令字中的一位用来指定操作应在 BSR 寄存器指定的存储区还是在快速访问存储区进行。此位表示为 ‘a’ 位（用于存取位）。

当强制操作在快速访问存储区进行时（a=0），快速存取 RAM 低位末位地址的后面是快速存取 RAM 高位的首地址。快速存取 RAM 高位会映射特殊功能寄存器，这样无需任何软件开销即可对这些寄存器存取。这样做对于测试状态标志和修改控制位很有用。

5.11 存储区选择寄存器（BSR）

因需要大的通用存储器空间，所以要指定 RAM 存储方案。数据存储器被分为 16 个存储区。当使用直接寻址时，应通过配置 BSR 选择目标存储区。

BSR<3:0> 保存 12 位 RAM 地址的高 4 位。BSR<7:4> 位总是读作 0，写入值没有任何影响（见图 5-7）。

在指令集中提供了 MOVLB 指令，用来帮助选择存储区。

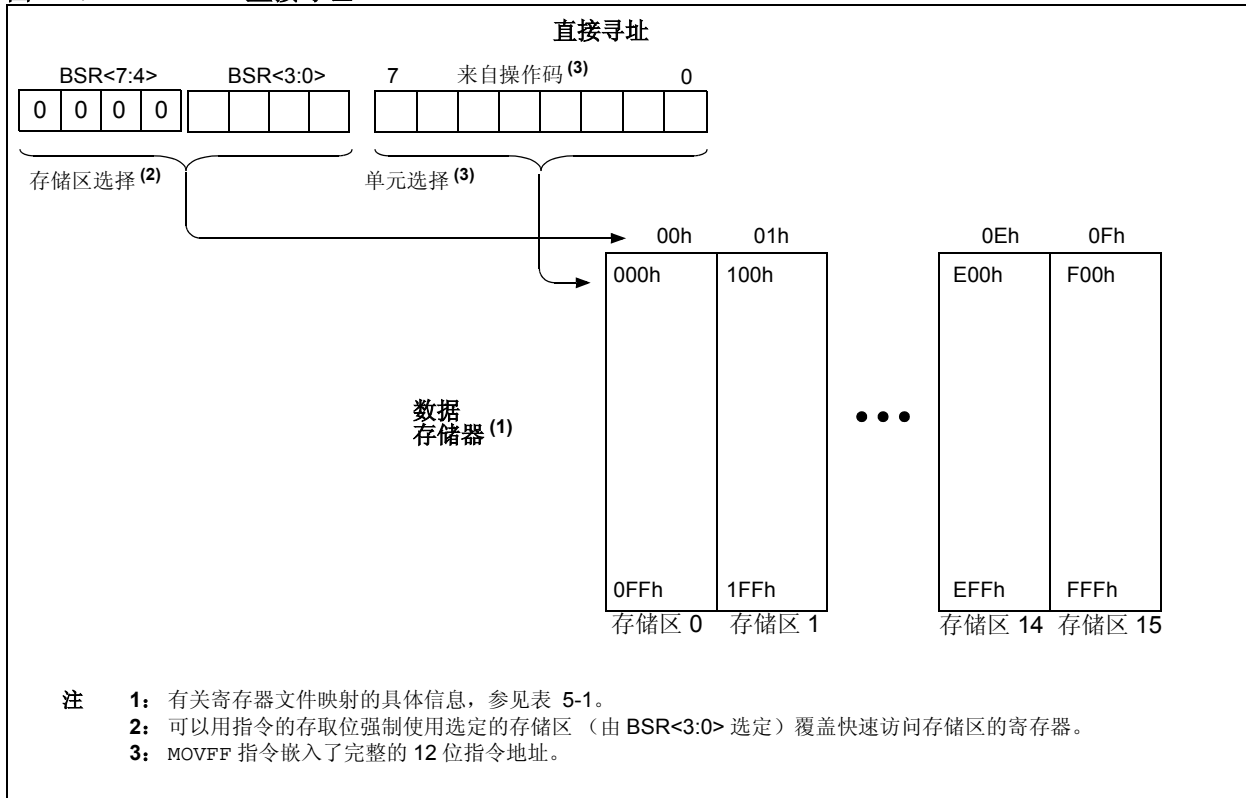
如果当前选择的存储区不存在，任何读操作都会返回 0，写操作则被忽略。指令执行时，状态寄存器中的位会相应被置位或清零。

每个存储区空间最大为 FFh（256 字节）。所有数据存储器都以静态 RAM 实现。

MOVFF 指令会忽略 BSR，因为该指令字嵌入了 12 位地址。

第 5.12 节“间接寻址、INDF 和 FSR 寄存器”说明了间接寻址，间接寻址允许对整个 RAM 空间进行线性寻址。

图 5-7: 直接寻址



5.12 间接寻址、INDF 和 FSR 寄存器

间接寻址是对数据存储器寻址的一种模式，间接寻址时指令中的数据存储器地址不固定。FSR 寄存器充当指向要读或写的数据存储器单元的指针。由于该指针位于 RAM 中，其内容可以通过程序修改。这对于操作数据存储器中的数据表和实现软件堆栈很有用。图 5-8 给出了在执行前，如何修改取出的指令。

使用 INDF 寄存器之一即可以实现间接寻址。任何使用 INDF 寄存器的指令实际上访问的是由文件选择寄存器 (FSR) 指向的寄存器。如使用间接寻址方式 (FSR=0) 对 INDF 寄存器本身进行读操作，读出值将为 00h。而使用间接寻址对 INDF 寄存器进行写操作，实际是空操作。如图 5-9 所示，FSR 寄存器包含一个 12 位地址。

INDFn 寄存器不是物理寄存器。对 INDFn 的寻址实际上是对 FSRn 寄存器中地址对应的寄存器寻址 (FSRn 是一个指针)。这就是间接寻址。

例 5-5 给出用最少的指令，采用间接寻址的方法实现对存储区 1 (单元 100h-1FFh) 中 RAM 单元的清零。

例 5-5: 如何使用间接寻址将 RAM (存储区 1) 清零的方法

```

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

有三种间接寻址寄存器。为了寻址整个数据存储器空间 (4096 字节)，这些寄存器的宽度都是 12 位。要存储 12 位的寻址信息，需要使用两个 8 位寄存器。

1. FSR0: 由 FSR0H:FSR0L 组成
2. FSR1: 由 FSR1H:FSR1L 组成
3. FSR2: 由 FSR2H:FSR2L 组成

此外还有寄存器 INDF0、INDF1 和 INDF2，它们都不是物理实现的。读 / 写这些寄存器将引起间接寻址，相应的 FSR 寄存器中的值是数据地址。如果指令向 INDF0 写入一个值，该值会被写入到 FSR0H:FSR0L 指向的地址。从 INDF1 读取数据，读取的是 FSR1H:FSR1L 指向的地址中的数据。只要代码中使用了操作数，就可以使用 INDFn。

如果通过 FSR 间接读取 INDF0、INDF1 或 INDF2，所有的读取将为 0 (Z 位被置位)。与此类似，如果间接地写 INDF0、INDF1 或 INDF2，实际等同于 NOP 指令，状态位不受影响。

5.12.1 间接寻址操作

除了 4 个寄存器地址外，每个 FSR 寄存器都有与其相关的 INDF 寄存器。对这 5 个寄存器之一进行操作将决定间接寻址时修改 FSR 的方式。

当对 5 个 INDFn 单元中的任何一个进行数据存取时，选择的地址将 FSRn 寄存器配置为：

- 在间接访问后对 FSRn 不做任何操作 (无变化) – INDFn
- 在间接访问后自动递减 FSRn (后减) – OSTDECn
- 在间接访问后自动递增 FSRn (后加) – POSTINCn
- 在间接访问前自动递增 FSRn (预加) – PREINCn
- 将 WREG 寄存器中的值作为 FSRn 的偏移量。在间接访问后不修改 WREG 或 FSRn 寄存器的值 (无变化) – PLUSWn

当使用自动递增递减功能时，状态寄存器中不会反映对 FSR 的影响。例如，如果间接寻址导致 FSR 等于“0”，Z 位将不会置位。

自动递增递减 FSR 会影响所有的 12 位。也就是说，如果 FSRnL 因为递增而溢出，则 FSRnH 将会自动递增。

增加的这些功能可以让 FSRn 用于数据存储器中的表操作外，还可用于堆栈指针。

每个 FSR 都有一个与之相关的地址，该地址用于变址间接访问。当对此 INDFn 单元 (PLUSWn) 进行数据存取时，FSRn 将在进行间接访问前，把 WREG 寄存器中的带符号值和 FSR 中的值相加做为地址。FSR 的值不会改变。WREG 偏移量范围为 -128 到 +127。

如果某个 FSR 寄存器包含的值指向 INDFn 之一，间接读取的值为 00h (Z 位置位)，而间接写入则相当于 NOP (状态位不受影响)。

如果执行间接寻址写操作的目标地址处于 FSRnH 或 FSRnL 寄存器中，数据将被写入 FSR 寄存器，但不执行预加 / 后加 / 后减功能。

PIC18F2331/2431/4331/4431

图 5-8: 间接寻址操作

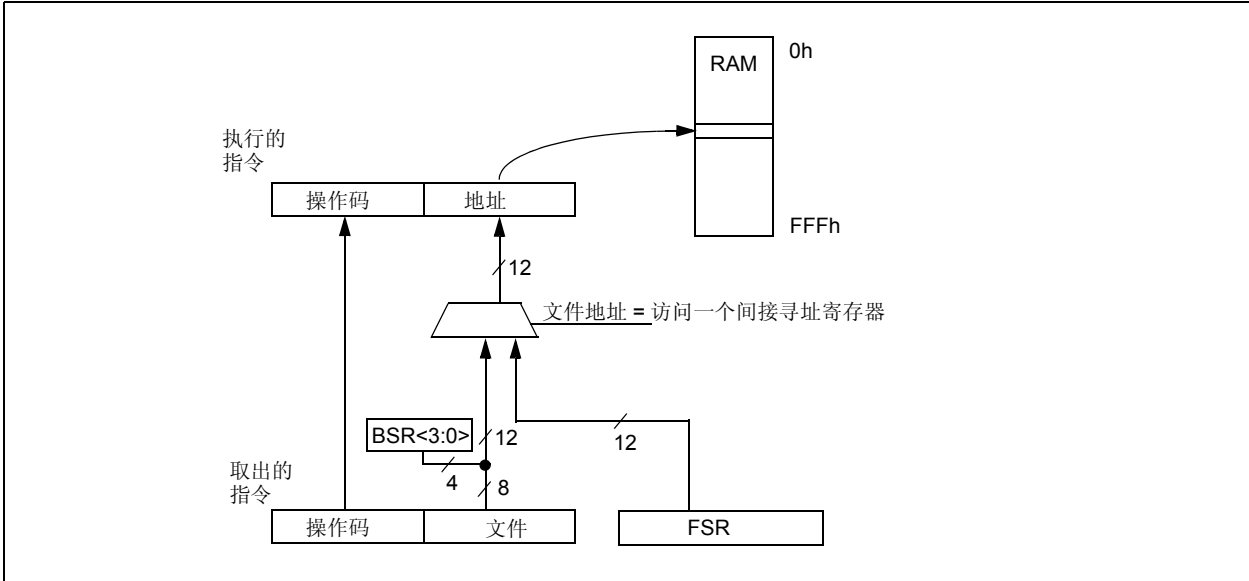
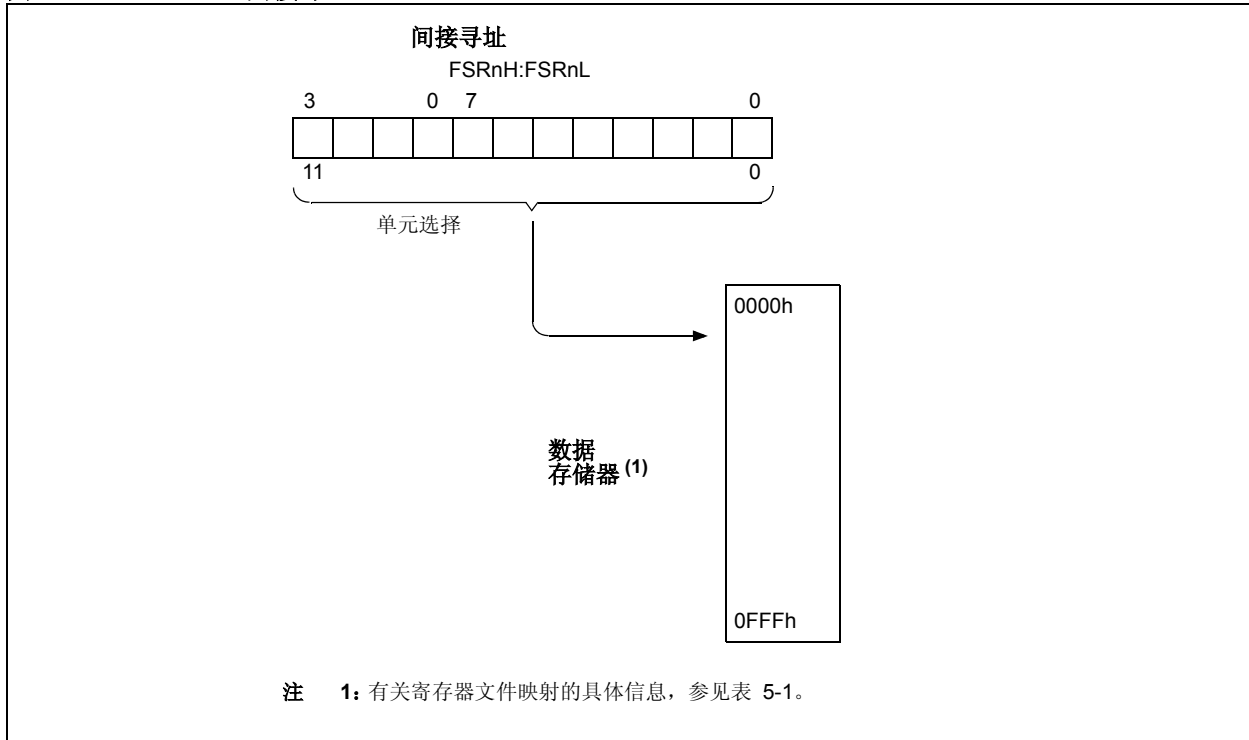


图 5-9: 间接寻址



PIC18F2331/2431/4331/4431

5.13 状态寄存器

如寄存器 5-2 所示，状态寄存器包含 ALU 的算术运算状态。和其他寄存器一样，它可以作为任何指令的操作数。如果一条影响 Z、DC、C、OV、或 N 位的指令的目标寄存器是状态寄存器，则会禁止对这 5 位进行直接写操作。根据器件逻辑对这些位置位或清零。所以，当执行一条把状态寄存器作为目标寄存器的指令后，状态寄存器的结果可能和预想的不一樣。

例如，指令 CLRF STATUS 将会把状态寄存器的高 3 位清零，并将 Z 位置位。这使状态寄存器的结果为 000u u1uu（其中 u 表示不变）。

因此，建议仅使用 BCF、BSF、SWAPF、MOVFF 和 MOVWF 指令来改变状态寄存器，因为这些指令不影响状态寄存器中的 Z、C、DC、OV 或 N 位。关于其他不影响任何状态位的指令，请参见表 23-2。

注： 在减法运算中，C 和 DC 位分别作为借位和辅助借位标志位。

寄存器 5-2:

状态寄存器

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	
—	—	—	N	OV	Z	DC	C	
bit 7								bit 0

bit 7-5 **未实现位：** 读作 0

bit 4 **N：** 负数位
此位用于有符号的算术运算（二进制补码）。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：** 溢出位
此位用于有符号的算术运算（二进制补码）。表明 7 位数值的溢出，这样导致了符号位（bit7）改变状态。
1= 有符号算术运算中发生溢出（本次运算）
0= 没有发生溢出

bit 2 **Z：** 零位
1= 算术运算或逻辑运算结果为零
0= 算术运算或逻辑运算结果不是零

bit 1 **DC：** 辅助进位 / 借位标志位
用于 ADDWF、ADDLW、SUBLW 和 SUBWF 指令
1= 结果的第 4 低位发生了进位
0= 结果中没有发生第 4 低位的进位

注： 由于辅助借位极性是相反的，减法指令通过加上第二个操作数的二进制补码来实现。对于移位指令（RRF、RLF），此位的值来自源寄存器的 bit4 或 bit3。

bit 0 **C：** 进位 / 借位标志位
用于 ADDWF、ADDLW、SUBLW 和 SUBWF 指令
1= 结果中发生了最高有效位进位
0= 结果中未发生最高有效位进位

注： 由于借位极性是相反的，减法指令通过加上第二个操作数的二进制补码来实现。对于移位指令（RRF、RLF），此位的值来自源寄存器的高位或低位。

图注：

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

PIC18F2331/2431/4331/4431

5.14 RCON 寄存器

复位控制 (RCON) 寄存器包含标志位, 用于区分器件复位的来源。这些标志位包括 TO、PD、POR、BOR 和 RI 位。该寄存器可读写。

注 1: 如果 BOREN 配置位置位 (欠压复位使能), BOR 位在上电复位时为“1”。发生了欠压复位后, BOR 位将被清零并必须由固件置位, 用于显示下一次欠压复位的发生。

注 2: 建议在检测到上电复位时, 将 $\overline{\text{POR}}$ 位置位, 以便检测后续的上电复位。

寄存器 5-3:

RCON 寄存器

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

- bit 7 **IPEN:** 中断优先级使能位
 1= 中断优先级使能
 0= 中断优先级禁止 (PIC16CXXX 兼容模式)
- bit 6-5 **未实现位:** 读作 0
- bit 4 **$\overline{\text{RI}}$:** RESET 指令标志位
 1= 未执行 RESET 指令 (只能用固件置位)
 0= 执行了 RESET 指令, 导致器件复位 (欠压复位发生后, 必须用固件置位)
- bit 3 **$\overline{\text{TO}}$:** 看门狗定时器超时标志位
 1= 通过上电、CLRWDT 指令或 SLEEP 指令置位
 0= 发生 WDT 超时
- bit 2 **$\overline{\text{PD}}$:** 掉电检测标志位
 1= 通过上电或 CLRWDT 指令置位
 0= 通过执行 SLEEP 指令清零
- bit 1 **$\overline{\text{POR}}$:** 上电复位状态位
 1= 未发生上电复位 (只能用固件置位)
 0= 发生上电复位 (上电复位发生后, 必须用固件置位)
- bit 0 **$\overline{\text{BOR}}$:** 欠压复位状态位
 1= 未发生欠压复位 (只能用固件置位)
 0= 发生欠压复位 (欠压复位发生后, 必须用固件置位)

图注:

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

6.0 闪存程序存储器

正常工作状态下，闪存程序存储器在整个 VDD 范围内都是可读写可擦除的。

读程序存储器时，每次为 1 个字节。写程序存储器时，每次为一个 8 字节的块。擦除程序存储器时，每次为一个 64 字节的块。用户代码不能执行批量擦除操作。

在擦写程序存储器时，系统会停止取指直到操作完成。擦写期间不能访问该程序存储器，因此也就无法执行代码。可用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。执行构成无效指令的程序存储器单元会导致执行 NOP。

6.1 读表与写表

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

- 读表 (TBLRD)
- 写表 (TBLWT)

程序存储器空间为 16 位宽，而数据 RAM 空间为 8 位宽。读表和写表操作通过一个 8 位寄存器 (TABLAT) 在这两个存储器空间之间传送数据。

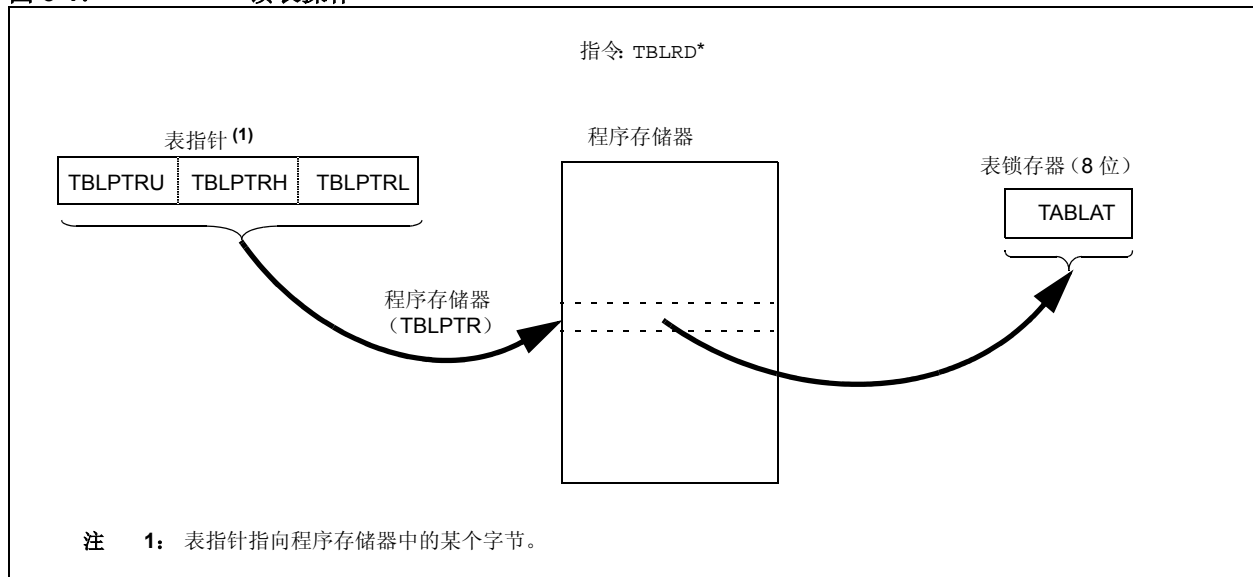
读表操作从程序存储器获取数据并将其放入数据 RAM 空间的 TABLAT。图 6-1 显示了程序存储器和数据 RAM 之间的一次读表操作。

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

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。如果使用写表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 (TBLPTRL<0>=0)。

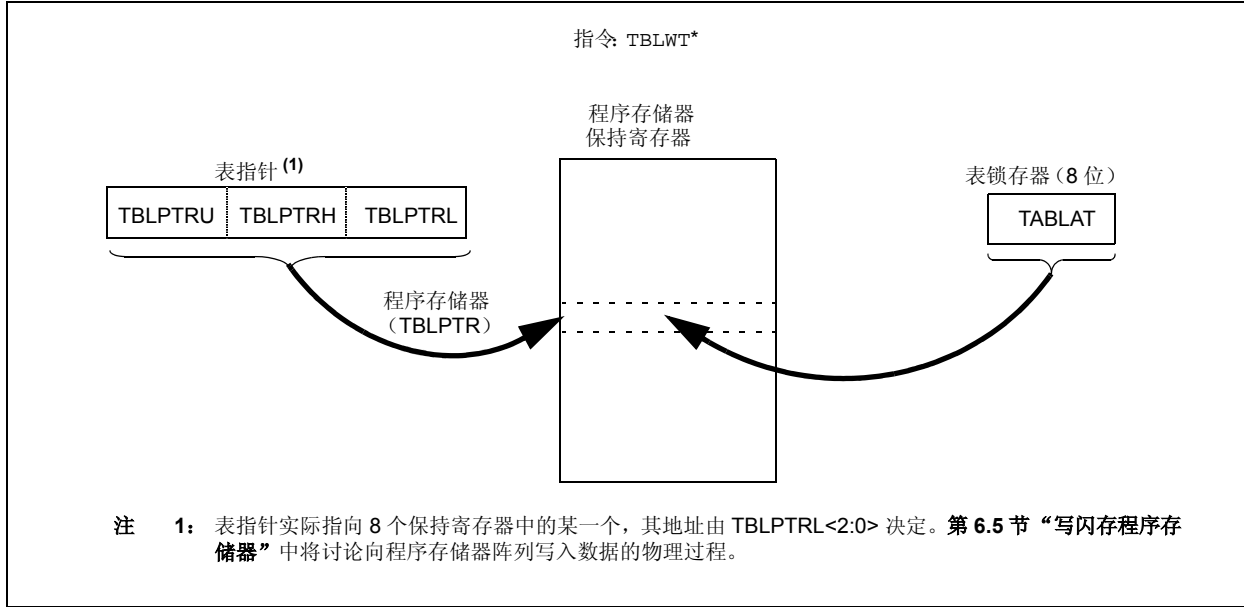
EEPROM 片上定时器控制擦写时间。擦写电压由可在器件电压范围内运行的片上电荷泵产生，执行字节或字操作。

图 6-1: 读表操作



PIC18F2331/2431/4331/4431

图 6-2: 写表操作



6.2 控制寄存器

TBLRD 和 TBLWT 指令要用到几个控制寄存器。包括:

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

6.2.1 EECON1 和 EECON2 寄存器

EECON1 是存储器访问的控制寄存器。

EECON2 不是物理寄存器。读 EECON2 得到的将全是 0。EECON2 寄存器专用于存储器的擦写时序。

控制位 EEPGD 决定访问的是程序存储器还是数据 EEPROM 存储器。清零该位, 操作将访问数据 EEPROM 存储器; 置位该位, 则访问程序存储器。

控制位 CFGS 决定访问的是配置寄存器还是程序存储器 / 数据 EEPROM 存储器。置位该位, 后续操作将访问配置寄存器; 清零该位时, 则由 EEPGD 位来选择具体访问程序存储器还是数据 EEPROM 存储器。

FREE 位控制程序存储器擦除操作。当该位置位时, 擦除操作在下一个 WR 命令时开始。当该位清零时, 则仅使能写操作。

擦写操作由 WREN 位使能和禁止。置位时, 允许擦写操作; 清零时, 禁止擦写操作, 当 WREN 位清零时不能置位 WR 位。这样处理可防止由于意外执行错误 (不期望的) 代码而造成对 EEPROM 存储器的误写入。

除了开始执行擦写操作以外, 固件应该始终保持 WREN 位清零。一旦固件已经将 WR 位置位, 就可以将 WREN 位清零。将 WREN 位清零不会影响进行中的操作。

当写操作由于复位被中断时, WRERR 位会置位。在这些情况下, 用户可以检查 WRERR 位, 并重新写入该单元。数据寄存器和地址寄存器 (EEDATA 和 EEADR) 需要被重新载入, 因为复位已使这些寄存器清零。

控制位 RD 和 WR 分别用于启动读操作和擦写操作。这些位由固件置位, 并在操作完成时由硬件清零。

在访问程序存储器 (EEPGD=1) 时, RD 位无法置位。通过读表指令来读程序存储器。有关读表的信息, 请参见第 6.3 节“读取闪存程序存储器”。

注: PIR2 寄存器中的中断标志位 EEIF 在写操作结束时置位。此标志位必须用软件清零。

PIC18F2331/2431/4331/4431

寄存器 6-1:

EECON1 寄存器

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD
bit 7						bit 0	

- bit 7 **EEPGD:** 闪存程序存储器或数据 EEPROM 存储器选择位
 1 = 访问闪存程序存储器
 0 = 访问数据 EEPROM 存储器
- bit 6 **CFGS:** 闪存程序存储器 / 数据 EEPROM 存储器或配置寄存器选择位
 1 = 访问配置寄存器
 0 = 访问闪存程序存储器或数据 EEPROM 存储器
- bit 5 **未实现位:** 读作 0
- bit 4 **FREE:** 闪存行擦除使能位
 1 = 在下一个 WR 命令时擦除 TBLPTR 寻址的程序存储器行
 (擦除操作完成时清零, 忽略 TBLPTR<5:0>)
 0 = 仅执行写操作
- bit 3 **WRERR:** EEPROM 错误标志位
 1 = 提前终止写操作 (由于自定时编程期间的任何复位)
 0 = 写操作正常完成
注: 当发生 WRERR 时, EEGD 和 CFGS 位不会清零。这样可以跟踪错误状况。
- bit 2 **WREN:** 写使能位
 1 = 允许擦写周期
 0 = 禁止擦写周期
- bit 1 **WR:** 写控制位
 1 = 启动数据 EEPROM 擦写周期或程序存储器的擦写周期
 (这是一个自定时操作, 一旦完成写操作, 该位即由硬件清零。用软件只能将 WR 位置位, 但不能清零。)
 0 = 写周期完成
- bit 0 **RD:** 读控制位
 1 = 启动读存储器
 (读操作需要一个周期。该位由硬件清零。用软件只能将 RD 位置位, 但不能清零。
 EEGD=1 时该位无法置位。)
 0 = 读操作完成

图注:

R= 可读位	S= 仅可置位	U= 未实现位, 读作 0
W= 可写位	-n= 上电复位时的值	1= 置位
x= 未知位		0= 清零

PIC18F2331/2431/4331/4431

6.2.2 TABLAT——表锁存寄存器

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

6.2.3 TBLPTR——表指针寄存器

表指针 (TBLPTR) 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：表指针高位字节、表指针高位字节和表指针低位字节 (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 位允许器件的寻址空间达到 2MB 程序存储器空间。置位第 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 TBLPTR。这些指令可以基于表操作以 4 种方法更新 TBLPTR。表 6-1 列出了这些操作。这些操作只会影响 TBLPTR 的低 21 位。

6.2.4 表指针范围

TBLPTR 用于读、写和擦除闪存程序存储器。

当执行 TBLRD 时，表指针的所有 22 位决定将程序存储器或配置寄存器中的哪个字节读入 TABLAT。

当执行 TBLWT 时，表指针的 3 个最低有效位 (TBLPTR<2:0>) 决定要写入程序存储器的哪个保持寄存器 (共有 8 个)。当程序存储器的定时写入 (长写) 开始时，表指针 TBLPTR 的 19 个最高有效位 (TBLPTR<21:3>) 将决定要写入哪个程序存储器块 (每块 8 字节) (忽略 TBLPTR<2:0>)。如需更多信息，请参见第 6.5 节 “写闪存程序存储器”。

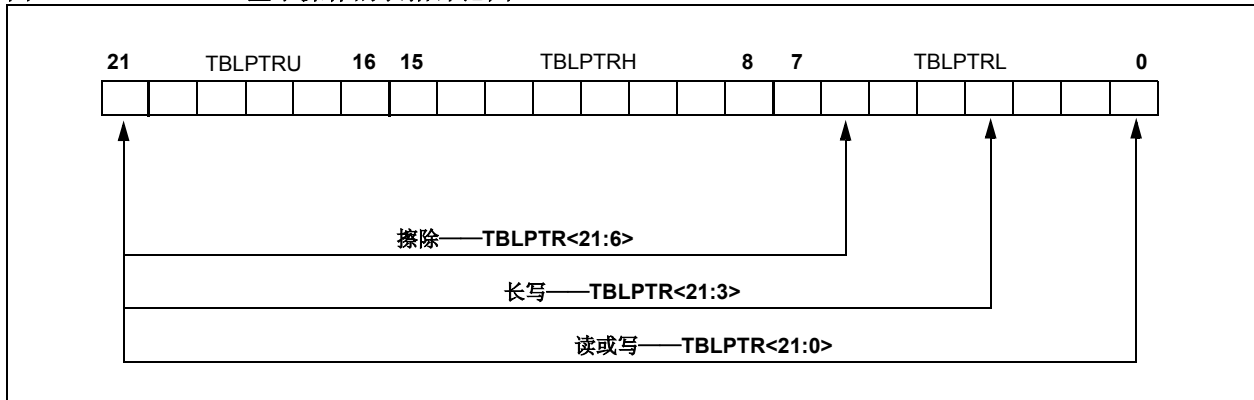
当执行擦除程序存储器时，表指针的 16 个最高有效位 (TBLPTR<21:6>) 指向将要擦除的 64 字节块。最低有效位 (TBLPTR<5:0>) 被忽略。

图 6-3 说明了 TBLPTR 基于闪存程序存储器操作的相关范围。

表 6-1: 执行 TBLRD 和 TBLWT 指令的表指针操作

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

图 6-3: 基于操作的表指针范围



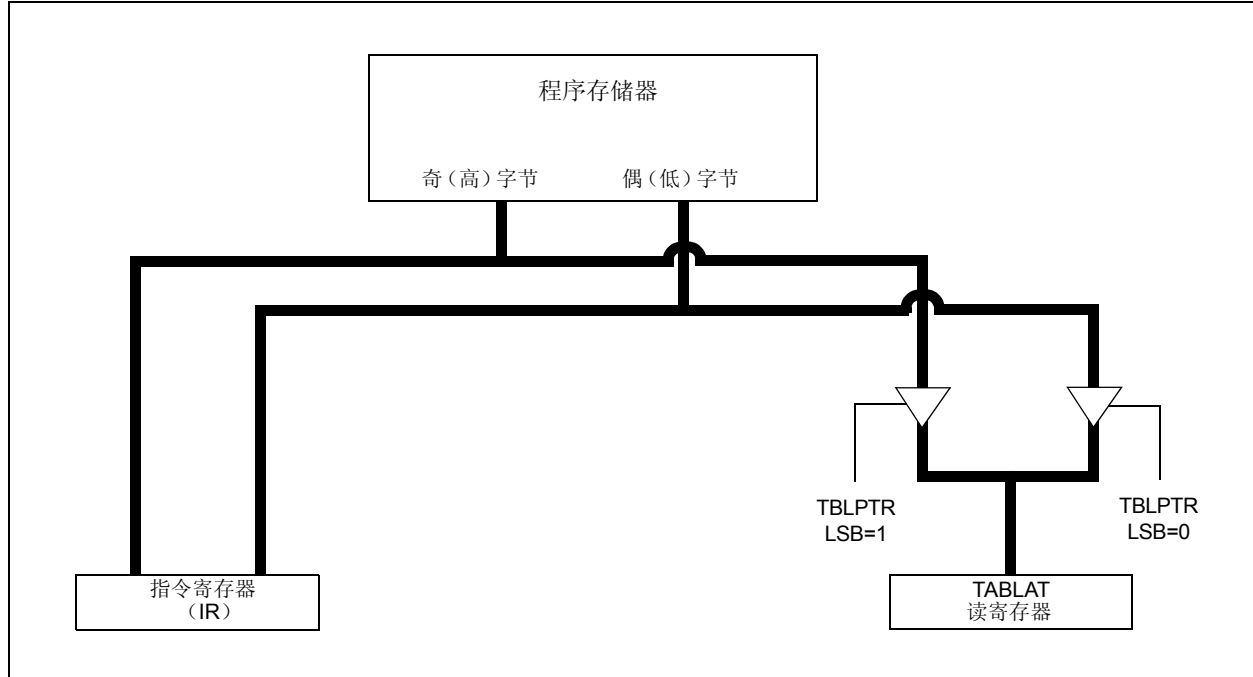
6.3 读取闪存程序存储器

TBLRD 指令用于从程序存储器获取数据并放入数据 RAM。读表操作每次从程序存储器读取 1 个字节。

TBLPTR 指向程序存储器空间的某个字节地址。执行 TBLRD 指令将把指向的字节装入 TABLAT。此外还可以自动修改 TBLPTR 以进行下一个读表操作。

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

图 6-4: 读取闪存程序存储器



例 6-1: 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 TBLPTR
MOVWF TABLAT               ; get data
MOVWF WORD_EVEN
TBLRD*+                    ; read into TABLAT and increment TBLPTR
MOVWF TABLAT               ; get data
MOVWF WORD_ODD
    
```

PIC18F2331/2431/4331/4431

6.4 擦除闪存程序存储器

在固件控制下最小擦除块大小为 32 字，即 64 字节。只有通过使用外部编程器，或通过 ICSP 控制，才能够批量擦除更大的程序存储器块。闪存存储器不支持字擦除。

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

擦除操作由 EECON1 寄存器控制。必须将 EEPGD 位置位以指向闪存程序存储器。必须将 CFGS 位清零以访问闪存程序存储器和数据 EEPROM 存储器。必须将 WREN 位置位以启用写操作。置位 FREE 位以选择擦除操作。作为必需的指令序列的一部分，要将 WR 位置位（如例 6-2 所示），然后开始真正的擦除操作。因为 TABLAT 寄存器被忽略，所以不必向其中装入任何数据。

作为一种保护机制，必须采用使用 EECON2 的写开始序列。

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

6.4.1 闪存程序存储器擦除顺序

擦除内部程序存储器地址单元块的事件顺序如下：

1. 将要擦除的行地址装入表指针。
2. 设置 EECON1 寄存器的相关位来执行擦除操作
 - 将 EEPGD 位置位以指向程序存储器；
 - 将 CFGS 位清零以访问程序存储器；
 - 将 WREN 位置位以启用写操作；
 - 将 FREE 位置位以启用擦除操作。
3. 禁止中断。
4. 向 EECON2 写入 55h。
5. 向 EECON2 写入 AAh。
6. 将 WR 位置位，这将开始行擦除周期。
7. CPU 在擦除期间将会停止（利用内部定时器约为 2 ms）。
8. 执行一个 NOP。
9. 重新允许中断。

例 6-2: 擦除闪存程序存储器的一行

	MOVLW	CODE_ADDR_UPPER		; load TBLPTR with the base
	MOVWF	TBLPTRU		; address of the memory block
	MOVLW	CODE_ADDR_HIGH		
	MOVWF	TBLPTRH		
	MOVLW	CODE_ADDR_LOW		
	MOVWF	TBLPTRL		
ERASE_ROW				
	BSF	EECON1,EEPGD		; point to Flash program memory
	BSF	EECON1,WREN		; enable write to memory
	BSF	EECON1,FREE		; enable Row Erase operation
	BCF	INTCON,GIE		; disable interrupts
	MOVLW	55h		
	MOVWF	EECON2		; write 55H
必须按照 这一顺序	MOVLW	AAh		
	MOVWF	EECON2		; write AAH
	BSF	EECON2,WR		; start erase (CPU stall)
	NOP			
	BSF	INTCON,GIE		; re-enable interrupts

6.5 写闪存程序存储器

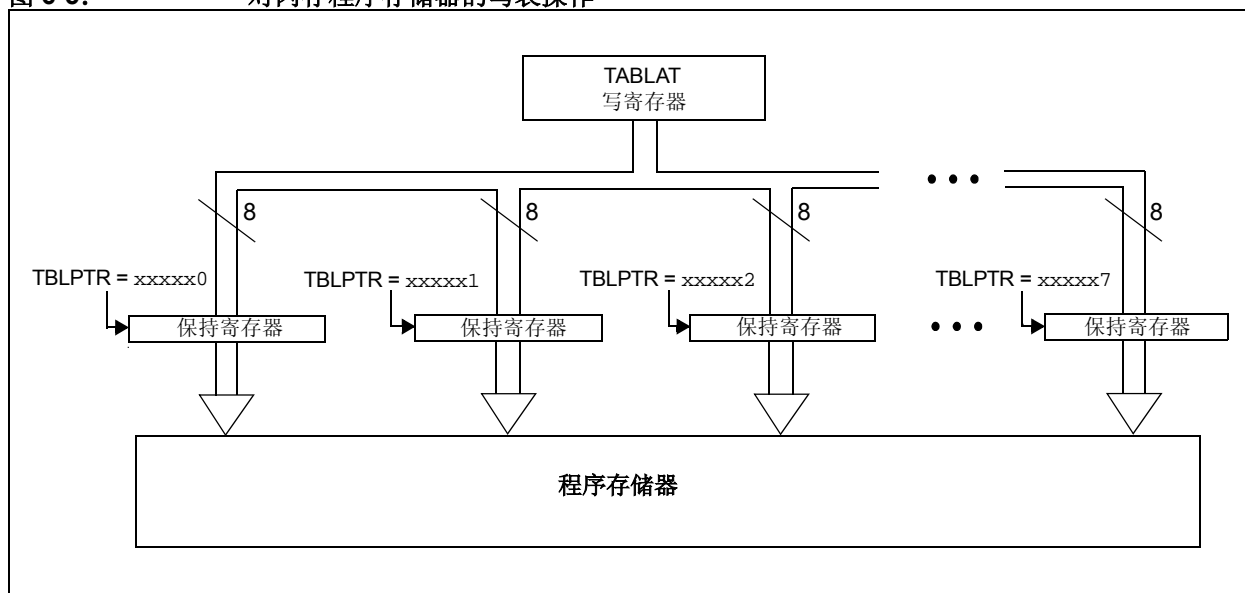
编程块大小为 4 字，即 8 字节。不支持字或字节编程。

在内部先使用写表操作将需要写入闪存存储器的内容装入保持寄存器中。写表操作使用 8 个保持寄存器进行编程。

因为表锁寄存器 (TABLAT) 只是单字节寄存器，所以对于每次编程操作，TBLWT 指令都必须执行 8 次。因为只写保持寄存器，所以所有的写表操作实际上都是短写。更新 8 个寄存器后，必须写 EECON1 寄存器，以使用长写开始编程操作。

对内部闪存编程要求长写操作。在长写周期中，指令停止执行。长写会被内部编程定时器终止。

图 6-5: 对闪存程序存储器的写表操作



6.5.1 闪存程序存储器写顺序

内部程序存储单元的编程操作顺序应该为：

1. 将 64 字节读入 RAM。
2. 必要时更新 RAM 中的数据值。
3. 把要擦除的目标地址装入表指针。
4. 执行行擦除（参见第 6.4.1 节“闪存程序存储器擦除顺序”）。
5. 将要写入第一个字节的地址装入表指针。
6. 将前 8 个字节写入保持寄存器（自动递增）。
7. 设置 EECON1 寄存器的相关位来执行写操作：
 - 将 EEPGD 位置位以指向程序存储器；
 - 将 CFGS 位清零以访问程序存储器；
 - 将 WREN 位置位以使能字节写操作。
8. 禁止中断。
9. 向 EECON2 写入 55h。
10. 向 EECON2 写入 AAh。
11. 将 WR 位置位，这将开始写周期。
12. CPU 在写入期间将会停止（利用内部定时器约为 2 ms）。
13. 执行一个 NOP。
14. 重新允许中断。
15. 重复 7 次 6-14 步骤，写入 64 字节。
16. 验证存储器（读表）。

此过程需要大约 18 ms 以更新存储器的一行 64 字节。例 6-3 举例给出了必要代码。

PIC18F2331/2431/4331/4431

例 6-3: 写入闪存程序存储器

```
        MOVLW    D'64                ; number of bytes in erase block
        MOVWF    COUNTER
        MOVLW    BUFFER_ADDR_HIGH    ; point to buffer
        MOVWF    FSR0H
        MOVLW    BUFFER_ADDR_LOW
        MOVWF    FSR0L
        MOVLW    CODE_ADDR_UPPER     ; Load TBLPTR with the base
        MOVWF    TBLPTRU             ; address of the memory block
        MOVLW    CODE_ADDR_HIGH
        MOVWF    TBLPTRH
        MOVLW    CODE_ADDR_LOW
        MOVWF    TBLPTRL             ; 6 LSB = 0
        MOVWF    TBLPTRL

READ_BLOCK
        TBLRD*+                      ; read into TABLAT, and inc
        MOVFW    TABLAT              ; get data
        MOVWF    POSTINC0           ; store data and increment FSR0
        DECFSZ   COUNTER            ; done?
        GOTO     READ_BLOCK         ; repeat

MODIFY_WORD
        MOVLW    DATA_ADDR_HIGH     ; point to buffer
        MOVWF    FSR0H
        MOVLW    DATA_ADDR_LOW
        MOVWF    FSR0L
        MOVLW    NEW_DATA_LOW        ; update buffer word and increment FSR0
        MOVWF    POSTINC0
        MOVLW    NEW_DATA_HIGH
        MOVWF    INDF0

ERASE_BLOCK
        MOVLW    CODE_ADDR_UPPER     ; load TBLPTR with the base
        MOVWF    TBLPTRU             ; address of the memory block
        MOVLW    CODE_ADDR_HIGH
        MOVWF    TBLPTRH
        MOVLW    CODE_ADDR_LOW
        MOVWF    TBLPTRL             ; 6 LSB = 0
        MOVWF    TBLPTRL
        BCF     EECON1,CFG5          ; point to PROG/EEPROM memory
        BSF     EECON1,EEPGD         ; point to Flash program memory
        BSF     EECON1,WREN          ; enable write to memory
        BSF     EECON1,FREE          ; enable Row Erase operation
        BCF     INTCON,GIE           ; disable interrupts
        MOVLW    55h                 ; Required sequence
        MOVWF    EECON2              ; write 55H
        MOVLW    AAh
        MOVWF    EECON2              ; write AAH
        BSF     EECON1,WR            ; start erase (CPU stall)
        NOP
        BSF     INTCON,GIE           ; re-enable interrupts

WRITE_BUFFER_BACK
        MOVLW    8                   ; number of write buffer groups of 8 bytes
        MOVWF    COUNTER_HI
        MOVLW    BUFFER_ADDR_HIGH    ; point to buffer
        MOVWF    FSR0H
        MOVLW    BUFFER_ADDR_LOW
        MOVWF    FSR0L

PROGRAM_LOOP
        MOVLW    8                   ; number of bytes in holding register
        MOVWF    COUNTER

WRITE_WORD_TO_HREGS
        MOVFW    POSTINC0           ; get low byte of buffer data and increment FSR0
        MOVWF    TABLAT            ; present data to table latch
        TBLWT*+                     ; short write
        ; to internal TBLWT holding register, increment
        ; TBLPTR
        DECFSZ   COUNTER            ; loop until buffers are full
        GOTO     WRITE_WORD_TO_HREGS
```

PIC18F2331/2431/4331/4431

例 6-3: 写入闪存程序存储器 (续)

```

PROGRAM_MEMORY
    BCF    INTCON,GIE           ; disable interrupts
    MOVLW 55h                   ; required sequence
    MOVWF EECON2                ; write 55H
    MOVLW AAh
    MOVWF EECON2                ; write AAH
    BSF    EECON1,WR           ; start program (CPU stall)
    NOP
    BSF    INTCON, GIE         ; re-enable interrupts
    DECFSZ COUNTER_HI          ; loop until done
    GOTO  PROGRAM_LOOP
    BCF    EECON1, WREN        ; disable write to memory
    
```

6.5.2 写校验

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

6.5.3 意外终止写操作

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

6.6 代码保护时的闪存程序存储器操作

有关闪存程序存储器代码保护的详细资料，请参见第 22.5 节“程序校验和代码保护”。

表 6-2: 与闪存程序存储器有关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	其他复位时的值
TBLPTRU	—	—	bit21	程序存储器表指针最高位字节 (TBLPTR<20:16>)					--00 0000	--00 0000
TBPLTRH	程序存储器表指针高位字节 (TBLPTR<15:8>)								0000 0000	0000 0000
TBLPTRL	程序存储器表指针低位字节 (TBLPTR<7:0>)								0000 0000	0000 0000
TABLAT	程序存储器表锁存器								0000 0000	0000 0000
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
EECON2	EEPROM 控制寄存器 2 (不是物理寄存器)								—	—
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	xx-0 x000	uu-0 u000
IPR2	OSFIP	—	—	EEIP	—	LVDIP	—	CCP2IP	1--1 -1-1	1--1 -1-1
PIR2	OSFIF	—	—	EEIF	—	LVDIF	—	CCP2IF	0--0 -0-0	0--0 -0-0
PIE2	OSFIE	—	—	EEIE	—	LVDIE	—	CCP2IE	0--0 -0-0	0--0 -0-0

图注: x= 未知, u= 不变, r= 保留, -= 未实现, 读作 0。
阴影单元格表示在访问闪存 / EEPROM 时未使用。

PIC18F2331/2431/4331/4431

注:

7.0 数据 EEPROM 存储器

在整个 V_{DD} 范围内的正常运行期间，数据 EEPROM 是可读写的。该数据存储器并不直接映射到寄存器文件空间。而是通过特殊功能寄存器（SFR）来间接寻址。

有四个 SFR 用于读写程序和数据 EEPROM 存储器。它们是：

- EECON1
- EECON2
- EEDATA
- EEADR

EEPROM 数据存储器允许字节读写。在与数据存储器块进行数据读写时，EEDATA 内存放 8 位读写数据，而 EEADR 存放要访问的 EEPROM 地址。这些器件都有 256 字节的数据 EEPROM，地址范围从 00h 到 FFh。

EEPROM 数据存储器具有高耐擦写周期。字节写操作将自动擦除该地址单元并写入新值（即先擦后写）。写入的时间由片上定时器控制。其值根据电压、温度和不同的芯片而不同。具体的限制值，请见参数 D122（第 25.0 节“电气特性”中的表 25-1）。

7.1 EEADR

地址寄存器可寻址 256 字节的数据 EEPROM。

7.2 EECON1 和 EECON2 寄存器

EECON1 是控制存储器访问的控制寄存器。

EECON2 并不是物理存在的寄存器。读 EECON2 将得到全 0。EECON2 寄存器专用于存储器写和擦除操作的时序。

控制位 EEPGD 决定访问的是程序存储器还是数据 EEPROM 存储器。清零时，操作将访问数据 EEPROM 存储器；置位时，则访问程序存储器。

控制位 CFGS 决定访问的是配置寄存器还是程序存储器 / 数据 EEPROM 存储器。置位时，后续操作会访问配置寄存器。而当 CFGS 清零时，则由 EEPGD 位来选择具体访问程序闪存存储器还是数据 EEPROM 存储器。

擦写操作由 WREN 位使能和禁止。置位时，允许擦写操作。清零时，禁止擦写操作。在 WREN 位清零期间 WR 位无法置位。这种机制可防止因意外执行错误（不期望的）代码而造成对存储器的误写入。

除了开始执行擦写操作时以外，固件必须始终保持 WREN 位清零。一旦固件已经将 WR 位置位，就可以将 WREN 位清零。将 WREN 位清零不会影响进行中的操作。

当写操作因复位而中断时，WRERR 位被置位。在这种情况下，用户可以检查 WRERR 位，并重新对该单位写入。数据寄存器和地址寄存器（EEDATA 和 EEADR）需要重新载入，因为复位已使这些寄存器清零。

控制位 RD 和 WR 分别开始读和擦写操作。这些位由固件置位，并在操作完成时由硬件清零。

在访问程序存储器（EEPGD=1）时，RD 位无法置位。程序存储器是通过读表指令读取的。有关读表的信息，请参见第 6.1 节“读表与写表”。

注： PIR2 寄存器中的中断标志位 EEIF 在写操作完成时置位。此标志位必须用软件清零。

PIC18F2331/2431/4331/4431

寄存器 7-1:

EECON1 寄存器

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGFS	—	FREE	WRERR	WREN	WR	RD
bit 7						bit 0	

- bit 7 **EEPGD:** 闪存程序存储器或数据 EEPROM 存储器选择位
 1= 访问闪存程序存储器
 0= 访问数据 EEPROM 存储器
- bit 6 **CFGFS:** 闪存程序存储器 / 数据 EEPROM 存储器或配置寄存器选择位
 1= 访问配置寄存器或校准寄存器
 0= 访问闪存程序存储器或数据 EEPROM 存储器
- bit 5 **未实现位:** 读作 0
- bit 4 **FREE:** 闪存行擦除使能位
 1= 在下一个 WR 命令时擦除 TBLPTR 寻址的程序存储器行
 (擦除操作结束后清零)
 0= 仅执行写操作
- bit 3 **WRERR:** EEPROM 错误标志位
 1= 提前终止写操作
 (自定时擦除或编程操作期间 MCLR 或 WDT 复位引起)
 0= 写操作正常完成
注: 当发生 WRERR 时, EEPROM 和 FREE 位不会清零。这样可以跟踪错误状况。
- bit 2 **WREN:** 擦写使能位
 1= 允许擦写周期
 0= 禁止擦写周期
- bit 1 **WR:** 写控制位
 1= 启动数据 EEPROM 擦写周期或程序存储器擦写周期。
 (操作是自定时的, 一旦写操作结束该位即由硬件清零。用软件只能将 WR 位置位, 但不能清零。)
 0= 写周期完成
- bit 0 **RD:** 读控制位
 1= 启动读存储器
 (读操作需要一个周期。RD 位由硬件清零。用软件只能将 RD 位置位, 但不能清零。
 EEPROM=1 时 RD 位无法置位。)
 0= 读操作完成

图注:

R= 可读位	S= 仅可置位	U= 未实现位, 读作 0
W= 可写位	-n= 上电复位时的值	1= 置位 0= 清零
x= 未知位		

7.3 读数据 EEPROM 存储器

要读取数据存储器某一地址单元内的值，用户必须将该地址写入 EEADR 寄存器，并把 EEPGD 控制位 (EECON1<7>) 清零，将控制位 RD (EECON1<0>) 置位。可在下一个指令周期访问该数据；因此，EEDATA 寄存器可由下一条指令读取。EEDATA 将保存这个值，直到另一个读操作开始或用户对其执行写入 (在写操作过程中)。

7.4 写数据 EEPROM 存储器

要写 EEPROM 数据地址单元，必须首先将该地址写入 EEADR 寄存器并将数据写入 EEDATA 寄存器。必须按照例 7-2 中的操作顺序才能启动写周期。

如果未完全按照这个顺序 (将 55h 写入 EECON2, 将 AAh 写入 EECON2, 然后将 WR 位置位) 逐字节写入, 写操作将不会开始。强烈建议在这个代码段执行期间禁止中断。

此外, EECON1 中的 WREN 位也必须被置位以启用写操作。这种机制可防止由于意外执行代码 (即程序跑飞) 造成对数据 EEPROM 的错误写入。除了更新 EEPROM 时以外, WREN 位应始终保持清零状态。硬件不能清零 WREN 位。

写序列启动以后, EECON1、EEADR 和 EDATA 不会被修改。除非将 WREN 位置位, 否则将禁止 WR 位被置位。WREN 位必须在前一个指令时被置位。WR 和 WREN 位不能由同一指令置位。

写周期完成后, WR 位将被硬件清零, 同时 EEPROM 中断标志位 (EEIF) 被置位。用户可以使能此中断或对该位进行查询。EEIF 必须用软件清零。

7.5 写校验

根据应用的实际需要, 将已写入存储器的值与原始值进行对比校验是很好的编程习惯。在应用中, 如果 EEPROM 位的写次数接近额定极限值时, 就应该采用写校验。

7.6 误写保护

有些情况下, 器件不宜向数据 EEPROM 存储器中写入数据。为了防止 EEPROM 误写操作, 器件内建了各种保护机制。上电时, WREN 位被清零。而且, 上电延时定时器 (持续时间 72 ms) 也可防止误写 EEPROM。

在欠压、电源毛刺或软件故障期间, 写操作启动序列和 WREN 位可共同防止意外写操作的发生。

例 7-1: 读数据 EEPROM

```

MOVLW DATA_EE_ADDR ;
MOVWF EEADR ; Data Memory Address to read
BCF EECON1, EEPGD ; Point to DATA memory
BSF EECON1, RD ; EEPROM Read
MOVF EEDATA, W ; W = EEDATA
    
```

例 7-2: 写数据 EEPROM

```

MOVLW DATA_EE_ADDR ;
MOVWF EEADR ; Data Memory Address to write
MOVLW DATA_EE_DATA ;
MOVWF EEDATA ; Data Memory Value to write
BCF EECON1, EEPGD ; Point to DATA memory
BSF EECON1, WREN ; Enable writes
BCF INTCON, GIE ; Disable Interrupts
必须按照
这一顺序
MOVLW 55h ;
MOVWF EECON2 ; Write 55h
MOVLW AAh ;
MOVWF EECON2 ; Write AAh
BSF EECON1, WR ; Set WR bit to begin write
BSF INTCON, GIE ; Enable Interrupts

SLEEP ; Wait for interrupt to signal write complete
BCF EECON1, WREN ; Disable writes
    
```

PIC18F2331/2431/4331/4431

7.7 代码保护时的操作

数据 EEPROM 存储器在配置字中有它自己的代码保护位。如果使能任一种代码保护机制，外部读写操作就被禁止。

单片机本身可以读写内部数据 EEPROM，与代码保护配置位的状态无关。其他信息请参阅第 22.0 节“CPU 的特殊功能”。

7.8 使用数据 EEPROM

数据 EEPROM 是高耐久性，可字节寻址的阵列，已对其进行优化，允许存储频繁更改的信息（例如，程序变量或其他经常更新的数据）。经常更改的值的更新频率通常高于规范 D124 或 D124A 中的规定。如果情况并非如此，必须执行阵列刷新。因此，只是偶尔更改的变量（例如常量、ID 及校准等）应该存储在闪存程序存储器中。

简单的数据 EEPROM 更新程序如例 7-3 所示。

注： 如果数据 EEPROM 仅用于存储常量和 / 或很少改变的数据，没有必要执行阵列刷新。参见规范 D124 或 D124A。

例 7-3: 数据 EEPROM 刷新程序

```

CLRF  EEADR          ; Start at address 0
BCF   EECON1, CFGS   ; Set for memory
BCF   EECON1, EEPGD  ; Set for Data EEPROM
BCF   INTCON, GIE    ; Disable interrupts
BSF   EECON1, WREN   ; Enable writes

LOOP
BSF   EECON1, RD     ; Read current address
MOVLW 55h           ;
MOVWF EECON2        ; Write 55h
MOVLW AAh           ;
MOVWF EECON2        ; Write AAh
BSF   EECON1, WR     ; Set WR bit to begin write
BTFSC EECON1, WR    ; Wait for write to complete
BRA   $-2
INCF  EEADR, F      ; Increment address
BRA   Loop          ; Not zero, do it again

BCF   EECON1, WREN   ; Disable writes
BSF   INTCON, GIE    ; Enable interrupts
    
```

表 7-1: 与数据 EEPROM 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	POR 和 BOR 复位时的值	所有其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMROIE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
EEADR	EEPROM 地址寄存器								0000 0000	0000 0000
EEDATA	EEPROM 数据寄存器								0000 0000	0000 0000
EECON2	EEPROM 控制寄存器 2（不是物理寄存器）								—	—
EECON1	EEPGD	CFGFS	—	FREE	WRERR	WREN	WR	RD	xx-0 x000	uu-0 u000
IPR2	OSFIP	—	—	EEIP	—	LVDIP	—	CCP2IP	1--1 -1-1	1--1 -1-1
PIR2	OSFIF	—	—	EEIF	—	LVDIF	—	CCP2IF	0--0 -0-0	0--0 -0-0
PIE2	OSFIE	—	—	EEIE	—	LVDIE	—	CCP2IE	0--0 -0-0	0--0 -0-0

图注： x= 未知，u= 不变，r= 保留，--= 未实现，读作 0。
阴影单元格表示在访问闪存 / EEPROM 时未使用。

PIC18F2331/2431/4331/4431

8.0 8 × 8 硬件乘法器

8.1 简介

在 PIC18F2331/2431/4331/4431 器件的 ALU 中包含了一个 8 × 8 硬件乘法器。通过利用硬件操作执行乘法，从而在一个指令周期内完成。这是一个无符号乘法运算，得到一个 16 位结果。结果存储在 16 位乘积寄存器对 (PRODH:PRODL) 中。该乘法器不会影响状态寄存器中的任何标志。

使用 8 × 8 乘法器在一个周期中执行任务会有如下优点：

- 更高的计算吞吐量
- 减少乘法算法所需的代码量

性能的增强使器件可以应用于以前仅能使用数字信号处理器的场合中。

表 8-1 显示了使用增加了单周期硬件乘法运算功能的器件与不支持硬件乘法运算的器件在执行相同功能时的性能对比。

表 8-1: 性能对比

程序	乘法方法	程序存储器 (字)	周期 (最大)	时间		
				40 MHz	10 MHz	4 MHz
8 × 8 无符号	无硬件乘法器	13	69	6.9 μs	27.6 μs	69 μs
	有硬件乘法器	1	1	100 ns	400 ns	1 μs
8 × 8 有符号	无硬件乘法器	33	91	9.1 μs	36.4 μs	91 μs
	有硬件乘法器	6	6	600 ns	2.4 μs	6 μs
16 × 16 无符号	无硬件乘法器	21	242	24.2 μs	96.8 μs	242 μs
	有硬件乘法器	24	24	2.4 μs	9.6 μs	24 μs
16 × 16 有符号	无硬件乘法器	52	254	25.4 μs	102.6 μs	254 μs
	有硬件乘法器	36	36	3.6 μs	14.4 μs	36 μs

8.2 工作原理

例 8-1 显示了执行 8 × 8 无符号乘法的运算过程。当 WREG 寄存器中已装入一个乘数时仅需一条指令。

例 8-2 显示了执行 8 × 8 有符号乘法的运算过程。考虑到乘数的符号位，必须测试每个乘数的最高有效位 (MSb)，并进行适当的减法。

例 8-1: 8 × 8 无符号乘法程序

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

例 8-2: 8 × 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
```

PIC18F2331/2431/4331/4431

例 8-3 显示了执行 16×16 无符号乘法的运算过程。公式 8-1 显示了使用的算法。32 位结果存储在 4 个寄存器 (RES3:RES0) 中。

公式 8-1: 16×16 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: 16×16 无符号乘法程序

```

MOV FARG1L, W
MULWF ARG2L      ; ARG1L * ARG2L ->
                  ; PRODH:PRODL

MOV FPRODH, RES1
MOV FPRODL, RES0
;

MOV FARG1H, W
MULWF ARG2H      ; ARG1H * ARG2H ->
                  ; PRODH:PRODL

MOV FPRODH, RES3
MOV FPRODL, RES2
;

MOV FARG1L, W
MULWF ARG2H      ; ARG1L * ARG2H ->
                  ; PRODH:PRODL

MOV FPRODL, W
ADDWF RES1, F    ; Add cross
MOV FPRODH, W   ; products
ADDWFC RES2, F  ;
CLRF WREG       ;
ADDWFC RES3, F  ;
;

MOV FARG1H, W
MULWF ARG2L      ; ARG1H * ARG2L ->
                  ; PRODH:PRODL

MOV FPRODL, W
ADDWF RES1, F    ; Add cross
MOV FPRODH, W   ; products
ADDWFC RES2, F  ;
CLRF WREG       ;
ADDWFC RES3, F  ;

```

例 8-4 显示了执行 16×16 有符号乘法的运算过程。。公式 8-2 是使用的算法。32 位结果存储在 4 个寄存器 (RES3:RES0) 中。考虑到乘数的符号位，必须测试每个乘数对的最高位 (MSb)，并进行适当的减法。

公式 8-2: 16×16 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: 16×16 有符号乘法程序

```

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

MOV F PRODH, RES1
MOV F PRODL, RES0
;

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

MOV F PRODH, RES3
MOV F PRODL, RES2
;

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

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

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

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

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

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

CONT_CODE
:

```

9.0 中断

PIC18F2331/2431/4331/4431 器件提供多个中断源和中断优先权，为每个中断源分配一个或高或低的优先级。高优先级中断向量地址为 000008h，低优先级中断向量地址为 000018h。高优先级中断事件将中断所有可能正在进行处理的低优先级中断。

有 13 个寄存器用于控制中断的操作。它们是：

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

建议这些寄存器中的符号位名称使用由 MPLAB® IDE 提供的 Microchip 头文件。这使得汇编器 / 编译器能够自动处理指定寄存器内这些位的位置。

通常，每个中断源有 3 位用来控制中断的操作。这些位的功能是：

- 标志位表明发生了中断事件
- 当标志位置位时，使能位便将程序执行转移到中断向量地址。
- 优先级位用于选择是高优先级还是低优先级（大多数中断源有优先级位）

将 IPEN 位（RCON<7>）置位，可使能中断优先级功能。当中断优先级使能时，有 2 位允许全局中断。将 GIEH 位（INTCON<7>）置位，可允许所有优先级位已置位（高优先级）的中断。将 GIEL 位（INTCON<6>）置位，可允许所有优先级位清零（低优先级）的中断。当中断标志位、使能位以及相应的全局中断使能位均被置位时，中断将立即转到地址 000008h 或 000018h，转到哪个地址取决于优先级位的设置。通过对应的使能位也能禁止相应的中断。

当 IPEN 位被清零（缺省状态），便会禁止中断优先级，并且中断与 PICmicro® 中档系列器件相兼容。在兼容模式下，各个中断源的中断优先级位均不起作用。INTCON<6> 是 PEIE 位，它可使能 / 禁止所有的外设中断源。INTCON<7> 是 GIE 位，它可使能 / 禁止所有的中断源。在兼容模式下，所有中断均跳转到地址 000008h。

当响应中断时，全局中断使能位被清零以禁止其他中断。IPEN 位清零后，全局中断使能位就是 GIE 位。如果使用中断优先级，这个位就是 GIEH 位或者 GIEL 位。高优先级中断源会中断低优先级的中断。处理高优先级中断时，不会处理低优先级中断。

返回地址被压入堆栈，PC 中装入中断向量地址（000008h 或 000018h）。在中断服务程序中，通过查询中断标志位可判断中断源。在重新允许中断前，必须用软件将中断标志位清零，以避免重复响应该中断。

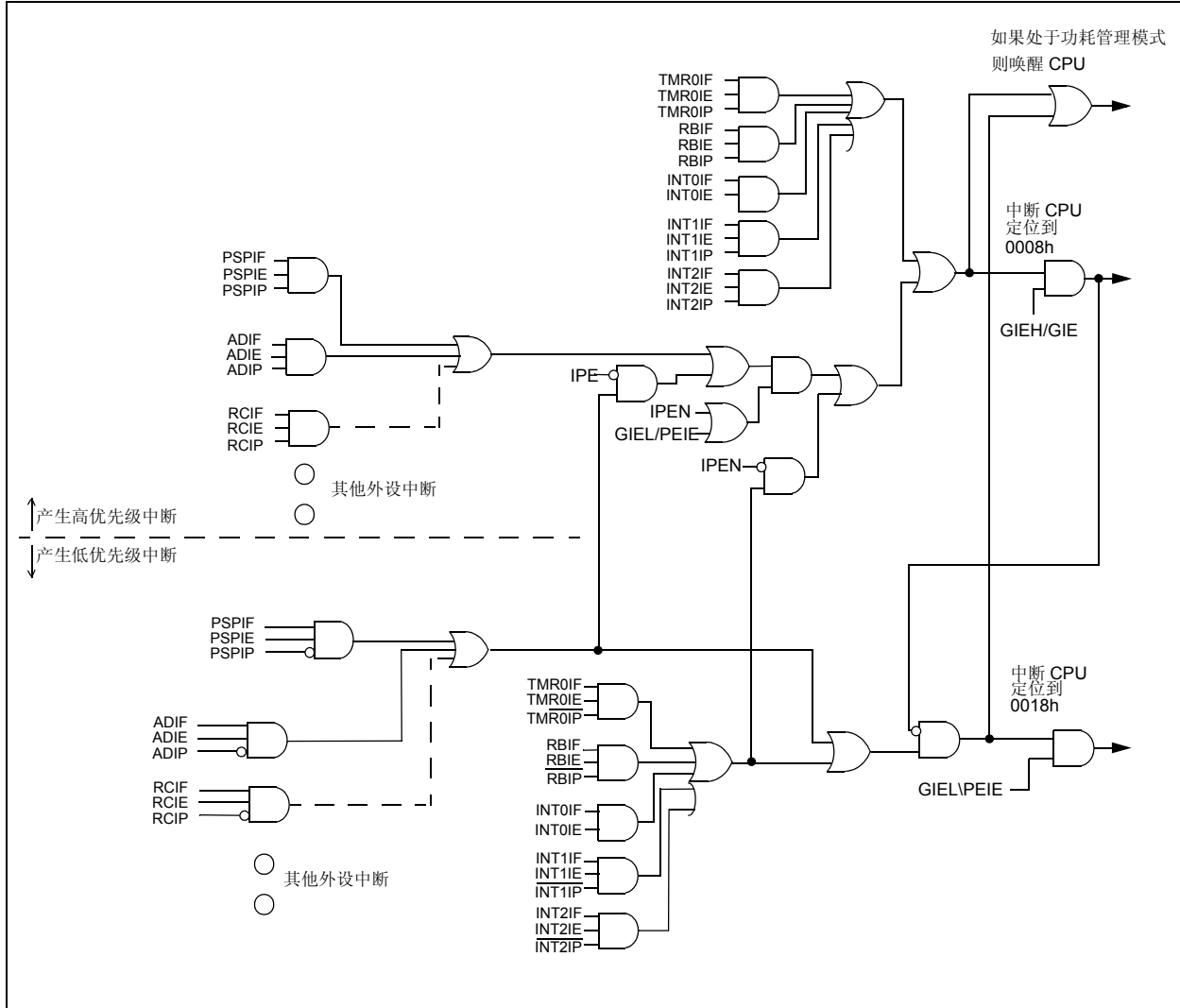
“中断返回”指令 RETFIE 退出中断程序并置位 GIE 位（若使用中断优先级，则为 GIEH 或 GIEL 位），GIE 位会重新允许中断。

对于外部中断事件，例如 INT 引脚中断或 PORTB 输入电平变化中断，中断响应延时会是 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。各中断标志位的置位不受对应的中断使能位和 GIE 位状态的影响。

注： 当任何中断被允许时，不要使用 MOVFF 指令修改中断控制寄存器。否则可能引起单片机状态出错。

PIC18F2331/2431/4331/4431

图 9-1: 中断逻辑电路



PIC18F2331/2431/4331/4431

9.1 INTCON 寄存器

INTCON 寄存器是可读写的寄存器，包含各种使能位、优先级位和标志位。

注： 当一个中断条件发生时，不管相应的中断使能位或全局使能位的状态如何，中断标志位都将置位。用户软件应在允许一个中断之前，确保先将该中断标志位清零。中断标志位可以用于软件查询。

寄存器 9-1:

INTCON 寄存器

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

- bit 7 **GIE/GIEH:** 全局中断使能位
当 IPEN=0 时:
1 = 允许所有未屏蔽的中断
0 = 禁止所有中断
当 IPEN=1 时:
1 = 允许所有高优先级中断
0 = 禁止所有高优先级中断
- bit 6 **PEIE/GIEL:** 外设中断使能位
当 IPEN=0 时:
1 = 允许所有未屏蔽的外设中断
0 = 禁止所有外设中断
当 IPEN=1 时:
1 = 允许所有低优先级的外设中断
0 = 禁止所有低优先级的外设中断
- bit 5 **TMR0IE:** TMR0 溢出中断使能位
1 = 允许 TMR0 溢出中断
0 = 禁止 TMR0 溢出中断
- bit 4 **INT0IE:** INT0 外部中断使能位
1 = 允许 INT0 外部中断
0 = 禁止 INT0 外部中断
- bit 3 **RBIE:** RB 端口电平变化中断使能位
1 = 允许 RB 端口 (RB7:RB4 引脚) 电平变化中断
0 = 禁止 RB 端口 (RB7:RB4 引脚) 电平变化中断
- bit 2 **TMR0IF:** TMR0 溢出中断标志位
1 = TMR0 寄存器已经溢出 (必须用软件清零)
0 = TMR0 寄存器未发生溢出
- bit 1 **INT0IF:** INT0 外部中断标志位
1 = 发生 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 = RB7:RB4 引脚中至少有一个的状态发生了变化 (必须用软件清零)
0 = RB7:RB4 引脚状态没有变化
- 注：** 不匹配的状况会不断地将 RBIF 位置位。读 PORTB，将结束不匹配状况，并将该位清零。

图注:

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

PIC18F2331/2431/4331/4431

寄存器 9-2:

INTCON2 寄存器

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
RBPU	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP
bit 7							bit 0

- bit 7 **RBPU**: PORTB 上拉使能位
1 = 禁止所有 PORTB 上拉
0 = 按各个端口锁存器值使能 PORTB 上拉
- bit 6 **INTEDG0**: 外部 Interrupt0 边沿选择位
1 = 上升沿时中断
0 = 下降沿时中断
- bit 5 **INTEDG1**: 外部 Interrupt1 边沿选择位
1 = 上升沿时中断
0 = 下降沿时中断
- bit 4 **INTEDG2**: 外部 Interrupt2 边沿选择位
1 = 上升沿时中断
0 = 下降沿时中断
- bit 3 **未实现位**: 读作 0
- bit 2 **TMR0IP**: TMR0 溢出中断优先级位
1 = 高优先级
0 = 低优先级
- bit 1 **未实现位**: 读作 0
- bit 0 **RBIP**: RB 端口电平变化中断优先级位
1 = 高优先级
0 = 低优先级

图注:

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

注: 当一个中断条件发生时, 不管相应的中断使能位或全局使能位的状态如何, 中断标志位都将置位。用户软件应在允许一个中断之前, 确保先将该中断标志位清零。中断标志位可以用于软件查询。

PIC18F2331/2431/4331/4431

寄存器 9-3:

INTCON3 寄存器

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

- bit 7 **INT2IP:** INT2 外部中断优先级位
1 = 高优先级
0 = 低优先级
- bit 6 **INT1IP:** INT1 外部中断优先级位
1 = 高优先级
0 = 低优先级
- bit 5 **未实现位:** 读作 0
- bit 4 **INT2IE:** INT2 外部中断使能位
1 = 允许 INT2 外部中断
0 = 禁止 INT2 外部中断
- bit 3 **INT1IE:** INT1 外部中断使能位
1 = 允许 INT1 外部中断
0 = 禁止 INT1 外部中断
- bit 2 **未实现位:** 读作 0
- bit 1 **INT2IF:** INT2 外部中断标志位
1 = 发生 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 INT1 外部中断 (必须用软件清零)
0 = 未发生 INT1 外部中断

图注:

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

注: 当一个中断条件发生时, 不管相应的中断使能位或全局使能位的状态如何, 中断标志位都将置位。用户软件应在允许一个中断之前, 确保先将该中断标志位清零。中断标志位可以用于软件查询。

PIC18F2331/2431/4331/4431

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，有两个外设中断标志寄存器（PIR1 和 PIR2）。

注 1: 当一个中断条件发生时，不管相应的中断使能位或全局使能位 GIE（INTCON<7>）的状态如何，中断标志位都将置位。

2: 用户软件应在允许一个中断之前，确保先将该中断标志位清零；同时在响应该中断后，也应将该中断标志位清零。

寄存器 9-4: PIR1: 外设中断请求（标志）寄存器 1

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

- bit 7 **未实现位:** 读作 0。
- bit 6 **ADIF:** A/D 转换器中断标志位
1 = A/D 转换已完成（必须用软件清零）
0 = A/D 转换未完成
- bit 5 **RCIF:** USART 接收中断标志位
1 = USART 接收缓冲器 RCREG 已满（读取 RCREG 时清零）
0 = USART 接收缓冲器为空
- bit 4 **TXIF:** USART 发送中断标志位
1 = USART 发送缓冲器 TXREG 为空（写入 TXREG 时清零）
0 = USART 发送缓冲器已满
- bit 3 **SSPIF:** 同步串行端口中断标志位
1 = 发送 / 接收完成（必须用软件清零）
0 = 等待发送 / 接收
- bit 2 **CCP1IF:** CCP1 中断标志位
捕捉模式:
1 = 发生了 TMR1 寄存器捕捉（必须用软件清零）
0 = 未发生 TMR1 寄存器捕捉
比较模式:
1 = 发生了 TMR1 寄存器的比较匹配（必须用软件清零）
0 = 未发生 TMR1 寄存器的比较匹配
PWM 模式:
在此模式下未使用
- bit 1 **TMR2IF:** TMR2 与 PR2 匹配中断标志位
1 = TMR2 与 PR2 匹配（必须用软件清零）
0 = TMR2 与 PR2 不匹配
- bit 0 **TMR1IF:** TMR1 溢出中断标志位
1 = TMR1 寄存器已经溢出（必须用软件清零）
0 = TMR1 寄存器未发生溢出

注 1: 在 PIC18F2X31 器件中，该位是保留位；总是保持清零状态。

图注:

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

PIC18F2331/2431/4331/4431

寄存器 9-5:

PIR2: 外设中断请求 (标志) 寄存器 2

R/W-0	U-0	U-0	R/W-0	U-0	R/W-0	U-0	R/W-0	
OSFIF	—	—	EEIF	—	LVDIF	—	CCP2IF	
bit 7								bit 0

- bit 7 **OSFIF:** 振荡器故障中断标志位
 1 = 系统振荡器故障, 改由 INTOSC 作为时钟输入 (必须用软件清零)
 0 = 系统时钟正在工作
- bit 6-5 **未实现位:** 读作 0
- bit 4 **EEIF:** EEPROM 或闪存写操作中中断标志位
 1 = 写操作完成 (必须用软件清零)
 0 = 写操作未完成或未开始
- bit 3 **未实现位:** 读作 0
- bit 2 **LVDIF:** 低压检测中断标志位
 1 = 供电电压已经降到规定的 LVD 电压以下 (必须用软件清零)
 0 = 供电电压高于规定的 LVD 电压
- bit 1 **未实现位:** 读作 0
- bit 0 **CCP2IF:** CCP2 中断标志位
捕捉模式:
 1 = 发生了 TMR1 寄存器捕捉 (必须用软件清零)
 0 = 未发生 TMR1 寄存器捕捉
比较模式:
 1 = 发生了 TMR1 寄存器的比较匹配 (必须用软件清零)
 0 = 未发生 TMR1 寄存器的比较匹配
PWM 模式:
 在 PWM 模式下未使用

图注:

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

PIC18F2331/2431/4331/4431

寄存器 9-6: PIR3: 外设中断标志寄存器 3

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	—	PTIF	IC3DRIF	IC2QEIF	IC1IF	TMR5IF	
bit 7								bit 0

bit 7-5 未实现位: 读作 0

bit 4 **PTIF**: PWM 时基中断位

1 = PWM 时基与 PTPER 寄存器中的值匹配。根据后分频器的设置发出中断。PTIF 必须用软件清零。

0 = PWM 时基与 PTPER 寄存器中的值不匹配。

bit 3 **IC3DRIF**: IC3 中断标志 / 方向变化中断标志位

IC3 使能 (CAP3CON<3:0>)

1 = 当 CAP3 输入出现有效沿时, TMR5 的值被其捕捉 (必须用软件清零)。

0 = 未发生 TMR5 捕捉。

QE1 使能 (QEIM<2:0>)

1 = 转动方向发生变化 (必须用软件清零)。

0 = 转动方向未变化。

bit 2 **IC2QEIF**: IC2 中断标志 / QE1 中断标志位

IC2 使能 (CAP2CON<3:0>)

1 = 当 CAP2 输入出现有效沿时, TMR5 的值被其捕捉 (必须用软件清零)。

0 = 未发生 TMR5 捕捉。

QE1 使能 (QEIM<2:0>)

1 = QE1 位置计数器已经达到 MAXCNT 值或已经检测到索引脉冲 INDX。取决于使能的 QE1 工作模式。必须用软件清零。

0 = QE1 位置计数器还未达到 MAXCNT 值或还未检测到索引脉冲。

bit 1 **IC1IF**: IC1 中断标志位

IC1 使能 (CAP1CON<3:0>)

1 = 当 CAP1 输入出现有效沿时, TMR5 的值被其捕捉 (必须用软件清零)。

0 = 未发生 TMR5 捕捉。

QE1 使能 (QEIM<2:0>) 且速率测量模式使能

(QEICON 寄存器中 VELM = 0)

1 = 当 PHA 或 PHB 输入出现有效速率沿时, TMR5 的值被其捕捉。必须将 CAP1CON 寄存器中的 CAP1REN 位置位。IC1IF 必须用软件清零。

0 = Timer5 的值未被有效速率沿捕捉。

bit 0 **TMR5IF**: Timer5 中断标志位

1 = Timer5 时基与 PR5 的值匹配 (必须用软件清零)。

0 = Timer5 时基与 PR5 的值不匹配。

图注:

R= 可读位

W= 可写位

U= 未实现位, 读作 0

-n= 上电复位时的值

1= 置位

0= 清零

x= 未知

9.3 PIE 寄存器

PIE寄存器包含各外设中断的使能位。根据外设中断源的数量，有两个外设中断使能寄存器（PIE1 和 PIE2）。当 IPEN = 0 时，要允许任何外设中断就必须将 PEIE 位置位。

寄存器 9-7:

PIE1: 外设中断使能寄存器 1

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

- bit 7 **未实现位:** 读作 0
- bit 6 **ADIE:** A/D 转换器中断使能位
1 = 允许 A/D 中断
0 = 禁止 A/D 中断
- bit 5 **RCIE:** USART 接收中断使能位
1 = 允许 USART 接收中断
0 = 禁止 USART 接收中断
- bit 4 **TXIE:** USART 发送中断使能位
1 = 允许 USART 发送中断
0 = 禁止 USART 发送中断
- bit 3 **SSPIE:** 同步串行端口中断使能位
1 = 允许 SSP 中断
0 = 禁止 SSP 中断
- bit 2 **CCP1IE:** CCP1 中断使能位
1 = 允许 CCP1 中断
0 = 禁止 CCP1 中断
- bit 1 **TMR2IE:** TMR2 与 PR2 匹配中断使能位
1 = 允许 TMR2 与 PR2 匹配中断
0 = 禁止 TMR2 与 PR2 匹配中断
- bit 0 **TMR1IE:** TMR1 溢出中断使能位
1 = 允许 TMR1 溢出中断
0 = 禁止 TMR1 溢出中断

图注:

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

PIC18F2331/2431/4331/4431

寄存器 9-8:

PIE2: 外设中断使能寄存器 2

R/W-0	U-0	U-0	R/W-0	U-0	R/W-0	U-0	R/W-0
OSFIE	—	—	EEIE	—	LVDIE	—	CCP2IE

bit 7 bit 0

bit 7 **OSFIE:** 振荡器故障中断使能位

1= 允许
0= 禁止

bit 6-5 **未实现位:** 读作 0

bit 4 **EEIE:** EEPROM 或闪存写操作中中断使能位

1= 允许
0= 禁止

bit 3 **未实现位:** 读作 0

bit 2 **LVDIE:** 低压检测中断使能位

1= 允许
0= 禁止

bit 1 **未实现位:** 读作 0

bit 0 **CCP2IE:** CCP2 中断使能位

1= 允许
0= 禁止

图注:

R= 可读位

W= 可写位

U= 未实现位, 读作 0

-n= 上电复位时的值

1= 置位

0= 清零

x= 未知

PIC18F2331/2431/4331/4431

寄存器 9-9: PIE3: 外设中断使能寄存器 3

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	—	PTIE	IC3DRIE	IC2QEIE	IC1IE	TMR5IE	
bit 7								bit 0

- bit 7-5 **未实现位:** 读作 0
- bit 4 **PTIE:** PWM 时基中断使能位
1 = 使能 PTIF
0 = 禁止 PTIF
- bit 3 **IC3DRIE:** IC3 中断使能 / 方向变化中断使能位
IC3 使能 (CAP3CON<3:0>)
1 = 允许 IC3 中断
0 = 禁止 IC3 中断
QE1 使能 (QEIM<2:0>)
1 = 允许方向变化中断
0 = 禁止方向变化中断
- bit 2 **IC2QEIE:** IC2 中断标志 / QE1 中断标志使能位
IC2 使能 (CAP2CON<3:0>)
1 = 允许 IC2 中断
0 = 禁止 IC2 中断
QE1 使能 (QEIM<2:0>)
1 = 允许 QE1 中断
0 = 禁止 QE1 中断
- bit 1 **IC1IE:** IC1 中断使能位
1 = 允许 IC1 中断
0 = 禁止 IC1 中断
- bit 0 **TMR5IE:** Timer5 中断使能位
1 = 允许 Timer5 中断
0 = 禁止 Timer5 中断

图注:

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

PIC18F2331/2431/4331/4431

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）。使用优先级位要求将中断优先级使能（IPEN）位置位。

寄存器 9-10: IPR1: 外设中断优先级寄存器 1

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

- bit 7 未实现位：读作 0
- bit 6 **ADIP**: A/D 转换器中断优先级位
1 = 高优先级
0 = 低优先级
- bit 5 **RCIP**: USART 接收中断优先级位
1 = 高优先级
0 = 低优先级
- bit 4 **TXIP**: USART 发送中断优先级位
1 = 高优先级
0 = 低优先级
- bit 3 **SSPIP**: 同步串行端口中断优先级位
1 = 高优先级
0 = 低优先级
- bit 2 **CCP1IP**: CCP1 中断优先级位
1 = 高优先级
0 = 低优先级
- bit 1 **TMR2IP**: TMR2 与 PR2 匹配中断优先级位
1 = 高优先级
0 = 低优先级
- bit 0 **TMR1IP**: TMR1 溢出中断优先级位
1 = 高优先级
0 = 低优先级

图注:

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

PIC18F2331/2431/4331/4431

寄存器 9-11:

IPR2: 外设中断优先级寄存器 2

R/W-1	U-0	U-0	R/W-1	U-0	R/W-1	U-0	R/W-1	
OSFIP	—	—	EEIP	—	LVDIP	—	CCP2IP	
bit 7								bit 0

- bit 7 **OSFIP:** 振荡器故障中断优先级位
1= 高优先级
0= 低优先级
- bit 6-5 **未实现位:** 读作 0
- bit 4 **EEIP:** EEPROM 或闪存写操作中中断优先级位
1= 高优先级
0= 低优先级
- bit 3 **未实现位:** 读作 0
- bit 2 **LVDIP:** 低压检测中断优先级位
1= 高优先级
0= 低优先级
- bit 1 **未实现位:** 读作 0
- bit 0 **CCP2IP:** CCP2 中断优先级位
1= 高优先级
0= 低优先级

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

PIC18F2331/2431/4331/4431

寄存器 9-12: **IPR3: 外设中断优先级寄存器 3**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	
—	—	—	PTIP	IC3DRIP	IC2QEIP	IC1IP	TMR5IP	
bit 7								bit 0

- bit 7-5 **未实现位:** 读作 0
- bit 4 **PTIP:** PWM 时基中断优先级位
1 = 高优先级
0 = 低优先级
- bit 3 **IC3DRIP:** IC3 中断优先级 / 方向变化中断优先级位
IC3 使能 (CAP3CON<3:0>)
1 = IC3 中断高优先级
0 = IC3 中断低优先级
QEI 使能 (QEIM<2:0>)
1 = 方向变化中断高优先级
0 = 方向变化中断低优先级
- bit 2 **IC2QEIP:** IC2 中断优先级 / QEI 中断优先级位
IC2 使能 (CAP2CON<3:0>)
1 = IC2 中断高优先级
0 = IC2 中断低优先级
QEI 使能 (QEIM<2:0>)
1 = QEI 中断高优先级
0 = QEI 中断低优先级
- bit 1 **IC1IP:** IC1 中断优先级位
1 = 高优先级
0 = 低优先级
- bit 0 **TMR5IP:** Timer5 中断优先级位
1 = 高优先级
0 = 低优先级

图注:

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

9.5 RCON 寄存器

RCON 寄存器包含了这样的位，可以用来确定器件上次从功耗管理模式被复位或唤醒的原因。RCON 中还包含一位，该位可以使能中断优先级（IPEN）。

寄存器 9-13: RCON 寄存器

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

- 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 6-5 **未实现位:** 读作 0
- bit 4 **\overline{RI} :** RESET 指令标志位
关于位操作的具体细节，参见寄存器 5-3
- bit 3 **\overline{TO} :** 看门狗定时器超时溢出标志位
关于位操作的具体细节，参见寄存器 5-3
- bit 2 **\overline{PD} :** 掉电检测标志位
关于位操作的具体细节，参见寄存器 5-3
- bit 1 **\overline{POR} :** 上电复位状态位
关于位操作的具体细节，参见寄存器 5-3
- bit 0 **\overline{BOR} :** 欠压复位状态位
关于位操作的具体细节，参见寄存器 5-3

图注:

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

PIC18F2331/2431/4331/4431

9.6 INTn 引脚中断

RC3/INT0、RC4/INT1 和 RC5/INT2 引脚的外部中断是边沿触发的。如果 INTCON2 寄存器中相应的 INTEDGx 位被置位，则为上升沿触发；如果 INTEDGx 位被清零，则为下降沿触发。当 RC3/INT0 引脚上出现一个有效边沿时，相应的标志位 INTxIF 被置位。通过对相应的使能位 INTxE 清零，该中断可被禁止。在重新允许该中断前，必须在中断服务程序中用软件将标志位 INTxIF 清零。如果 INTxE 位在进入功耗管理模式前被置位，则所有的外部中断（INT0、INT1 及 INT2）可以将处理器从功耗管理模式中唤醒。如果全局中断使能位 GIE 被置位，则处理器在唤醒之后将转至中断向量。

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

9.7 TMR0 中断

在 8 位模式下（缺省设置），TMR0 寄存器的溢出（FFh → 00h）将会使标志位 TMR0IF 置位。在 16 位模式下，TMR0H:TMR0L 寄存器的溢出（FFFFh → 0000h）将会使标志位 TMR0IF 置位。通过置位 / 清零使能位 TMR0IE（INTCON<5>），该中断可以被允许 / 禁止。Timer0 的中断优先级由中断优先级位 TMR0IP（INTCON2<2>）包含的值决定。更多详情，请参阅第 11.0 节“Timer0 模块”。

9.8 PORTB 电平变化中断

PORTB<7:4> 的输入电平变化，会将标志位 RBIF（INTCON<0>）置位。通过置位 / 清零使能位 RBIE（INTCON<3>），该中断可以被允许 / 禁止。PORTB 电平变化中断的优先级由中断优先级位 RBIP（INTCON2<0>）包含的值决定。

9.9 中断的现场保护

在中断期间，将返回的 PC 地址压入堆栈。另外，将 WREG、Status 和 BSR 寄存器的值压入快速返回堆栈。如果未使用中断快速返回（见第 5.3 节“快速寄存器堆栈”），用户可能需要在进入中断服务程序时，保存 WREG、Status 和 BSR 寄存器的值。根据用户的实际应用，还可能需要在保存其他寄存器的值。例 9-1 给出了如何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: 在 RAM 内保存 STATUS、WREG 和 BSR 寄存器

```
MOVWF  W_TEMP                ; W_TEMP is in virtual 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，最多有五个端口可用。I/O 端口的一些引脚与器件上外设功能部件的某个功能复用。一般来说，当使能某个外设时，对应的引脚就不能作为通用 I/O 引脚。

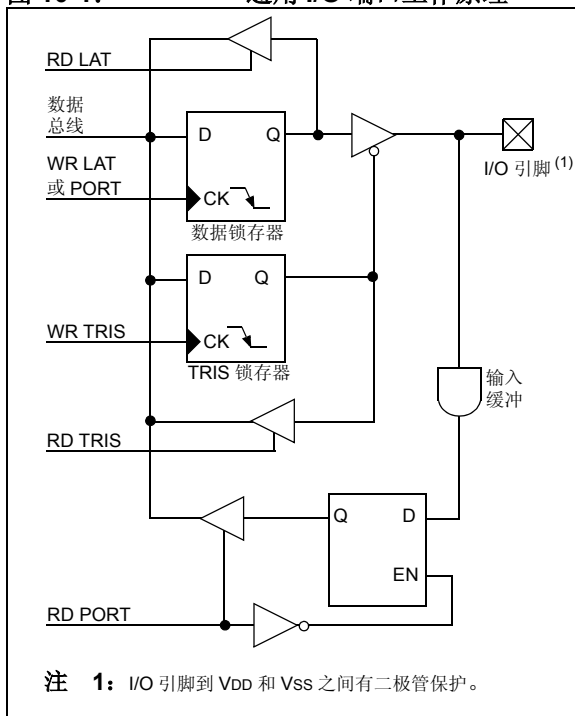
每个端口都有三个寄存器用于操作。它们是：

- TRIS 寄存器（数据方向寄存器）
- PORT 寄存器（读取器件的引脚电平）
- LAT 寄存器（输出锁存器）

在对 I/O 引脚电平驱动的值执行读—修改—写操作时，数据锁存器（LAT 寄存器）非常有用。

图 10-1 所示为通用 I/O 端口的简化模型，其中没有与其他外设进行交互。

图 10-1: 通用 I/O 端口工作原理



10.1 PORTA、TRISA 和 LATA 寄存器

PORTA 是一个 8 位宽的双向端口。相应的数据方向寄存器为 TRISA。置位 TRISA 位 (= 1) 可以让相应 PORTA 引脚作为输入引脚（即将相应的输出驱动器置于高阻态）。清零 TRISA 位 (= 0) 将使相应的 PORTA 引脚作为输出引脚（即将输出锁存器的内容置于所选的引脚上）。

读 PORTA 寄存器就是读引脚状态，而写该寄存器就是写入端口锁存器。

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—写操作就是读写 PORTA 的锁存输出值。

RA<2:4> 引脚与三个输入捕捉引脚以及正交编码器接口引脚复用。RA6 和 RA7 引脚与主振荡器引脚复用，通过在配置寄存器 1H 中选择主振荡器可使能这些引脚为振荡器还是 I/O 引脚（如需了解详细信息，请参阅第 22.1 节“配置位”）。当它们没有被用作端口引脚时，RA6 和 RA7 及其相关的 TRIS 和 LAT 位读作 0。

其他 PORTA 引脚与模拟输入、模拟 VREF+ 和 VREF- 输入以及比较器参考电压输出复用。通过清零 / 置位 ANSEL0 和 ANSEL1 寄存器内的控制位可以选择 RA3:RA0 和 RA5 引脚作为 A/D 转换器输入。

- 注
- 1: 在上电复位时，RA5:RA0 被配置为模拟输入并读作 0。
 - 2: RA5 I/F 只在 40 引脚的器件（PIC18F4X31）上可用。

即使将 RA 引脚用作模拟输入，TRISA 寄存器仍控制 RA 引脚的方向。在将它们用作模拟输入时，用户必须确保 TRISA 寄存器中的相应位保持置位状态。

例 10-1: 初始化 PORTA

```

CLRf   PORTA   ; Initialize PORTA by
             ; clearing output
             ; data latches
CLRf   LATA    ; Alternate method
             ; to clear output
             ; data latches
MOVLW  0x3F   ; Configure A/D
MOVWF  ANSEL0 ; for digital inputs
MOVLW  0xCF   ; Value used to
             ; initialize data
             ; direction
MOVWF  TRISA  ; Set RA<3:0> as inputs
             ; RA<5:4> as outputs
    
```

PIC18F2331/2431/4331/4431

图 10-2: RA0 框图

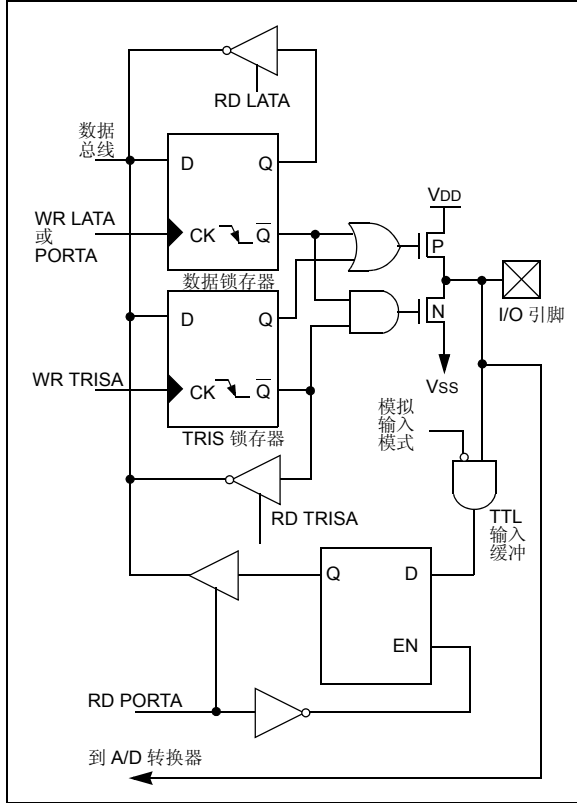


图 10-3: RA1 框图

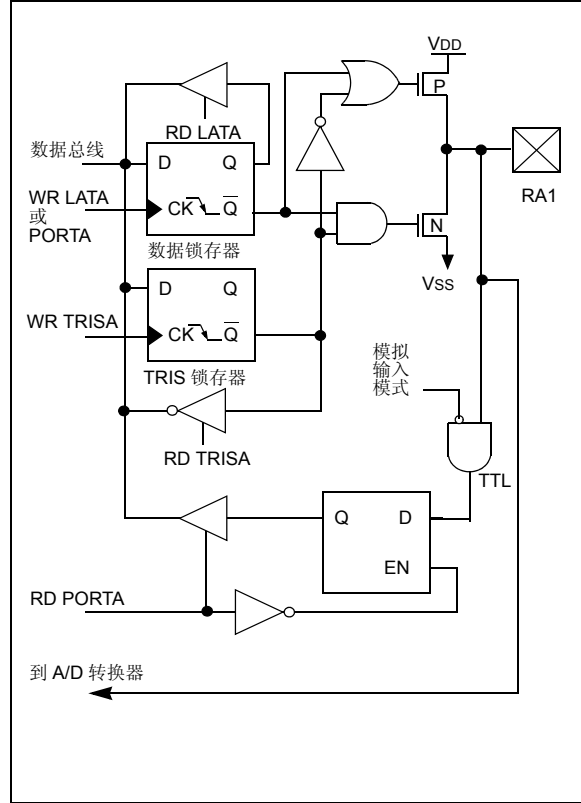


图 10-4: RA3:RA2 引脚框图

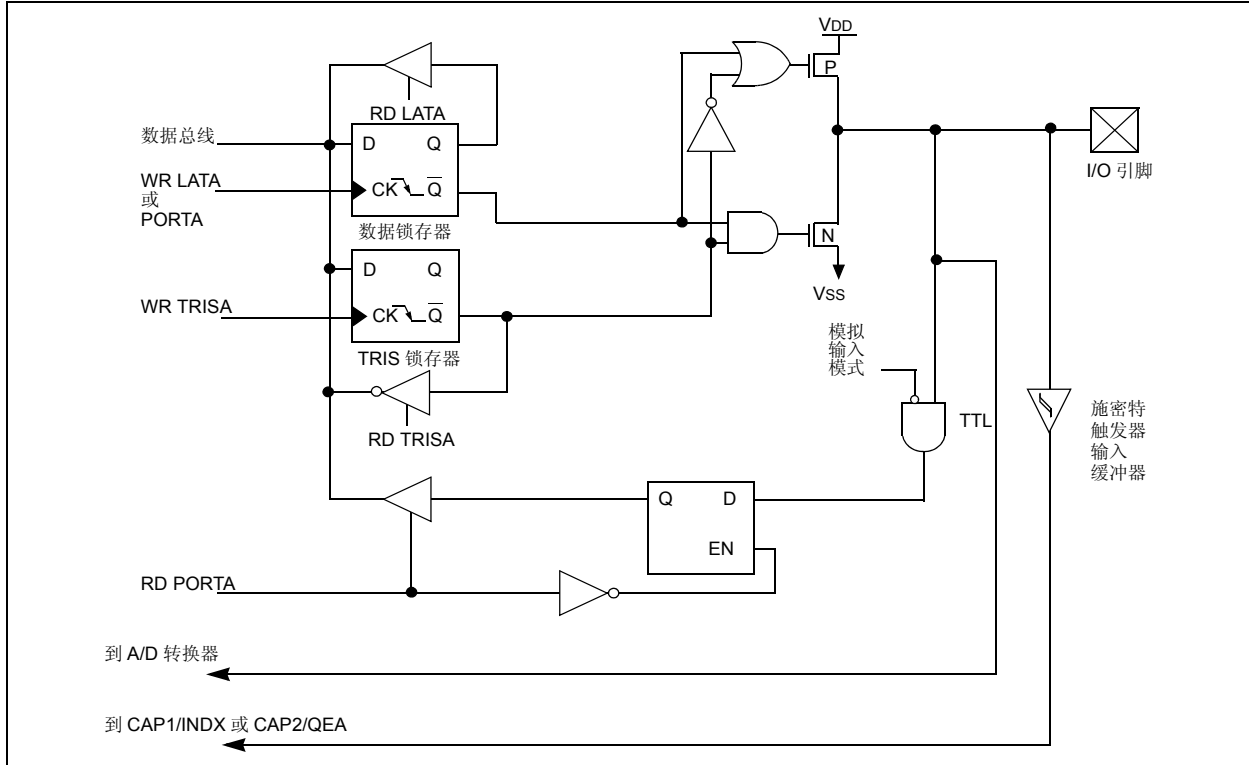
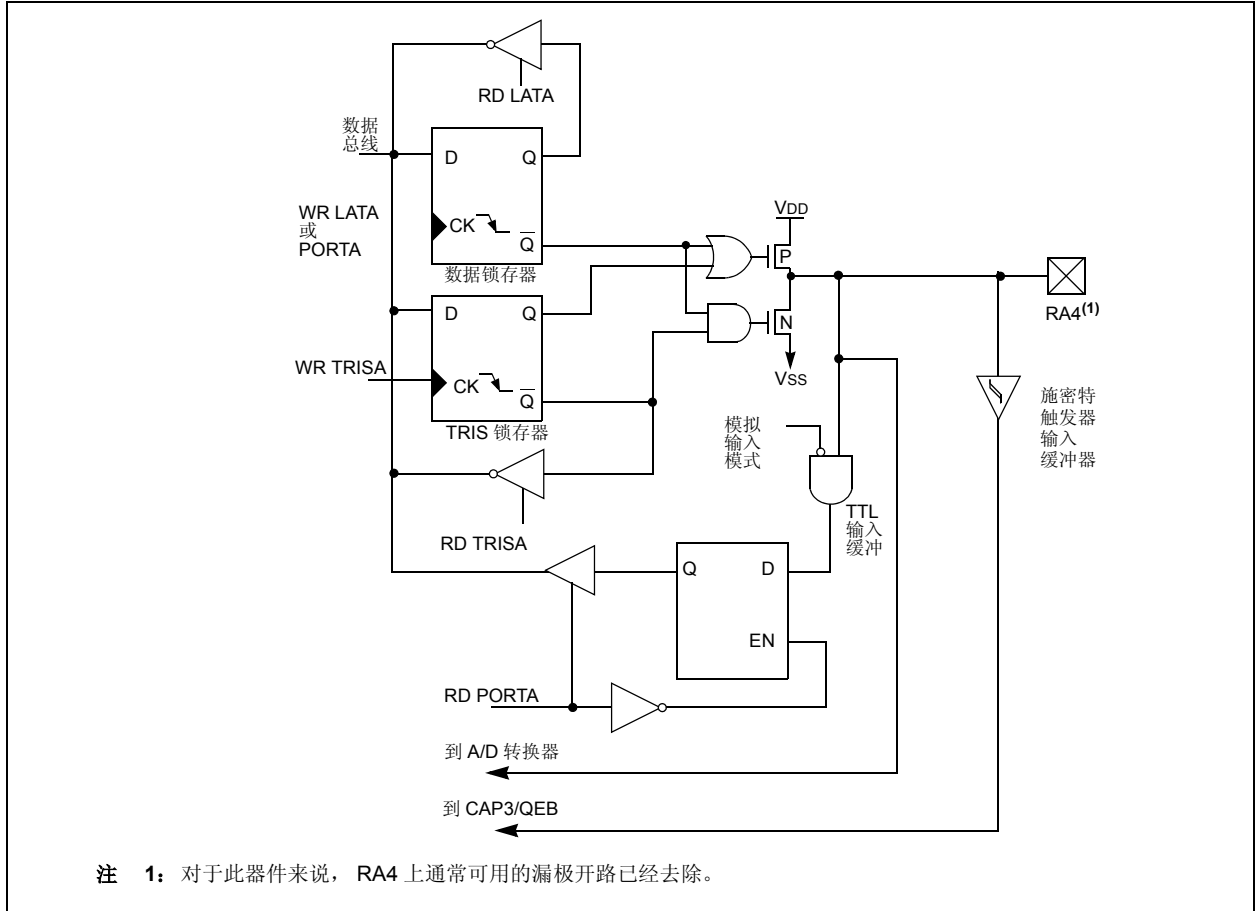


图 10-5: RA4 框图



PIC18F2331/2431/4331/4431

图 10-6: RA5 框图

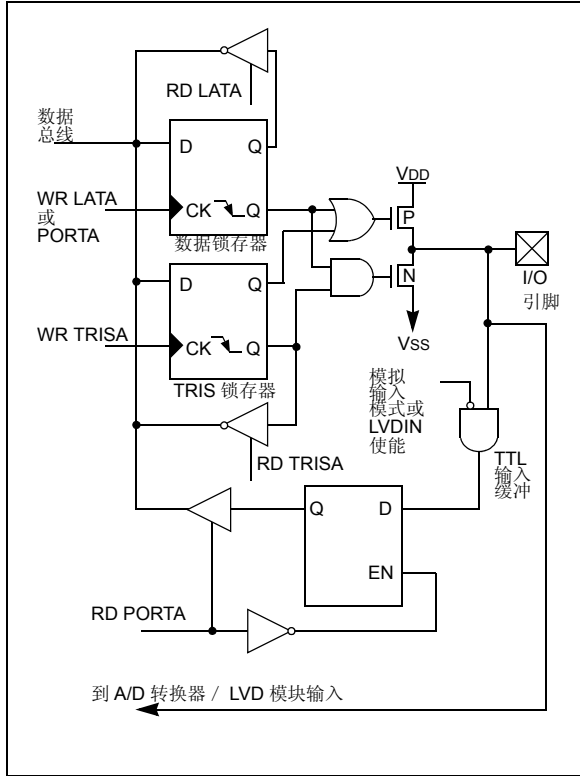


图 10-8: RA7 框图

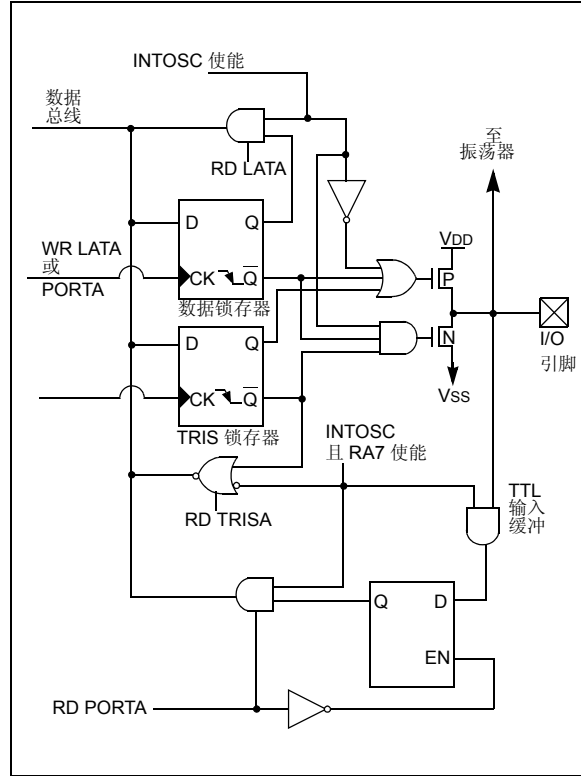
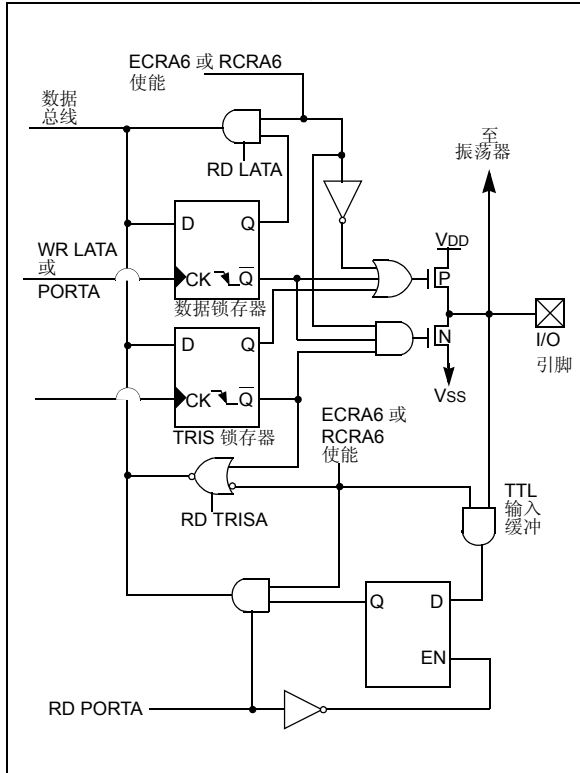


图 10-7: RA6 框图



PIC18F2331/2431/4331/4431

表 10-1: PORTA 功能

名称	Bit #	缓冲器	功能
RA0/AN0	bit 0	TTL	输入 / 输出或模拟输入。
RA1/AN1	bit 1	TTL	输入 / 输出或模拟输入。
RA2/AN2/VREF-/CAP1/INDX	bit 2	TTL/ST	输入 / 输出、模拟输入、VREF-、捕捉输入或 QE1 索引输入。
RA3/AN3/VREF+/CAP2/QEA	bit 3	TTL/ST	输入 / 输出、模拟输入、VREF+、捕捉输入或正交通道 A 输入。
RA4/AN4/CAP3/QEB	bit 4	TTL/ST	输入 / 输出、模拟输入、捕捉输入或正交通道 B 输入。
RA5/AN5/LVDIN	bit 5	TTL	输入 / 输出、模拟输入或低压检测输入。
OSC2/CLKO/RA6	bit 6	TTL	OSC2、时钟输出或 I/O 引脚。
OSC1/CLKI/RA7	bit 7	TTL	OSC1、时钟输入或 I/O 引脚。

图注: TTL = TTL 输入, ST = 施密特触发器输入

表 10-2: 与 PORTA 相关的寄存器综述

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
PORTA	RA7 ⁽¹⁾	RA6 ⁽¹⁾	RA5	RA4	RA3	RA2	RA1	RA0	xx0x 0000	uu0u 0000
LATA	LATA7 ⁽¹⁾	LATA6 ⁽¹⁾	LATA 数据输出寄存器						xxxx xxxx	uuuu uuuu
TRISA	TRISA7 ⁽¹⁾	TRISA6 ⁽¹⁾	PORTA 数据方向寄存器						1111 1111	1111 1111
ADCON1	VCFG1	VCFG0	—	FIFOEN	BFEMT	BFOVFL	ADPNT1	ADPNT0	00-1 0000	00-1 0000
ANSEL0	ANS7 ⁽²⁾	ANS6 ⁽²⁾	ANS5 ⁽²⁾	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
ANSEL1	—	—	—	—	—	—	—	ANS8 ⁽²⁾	---- -1	---- -1

图注: x= 未知, u= 不变, -= 未实现位, 读作 0。阴影单元格表示 PORTA 未使用。

- 注 1: RA7:RA6 以及与之相关的锁存器和数据方向位根据振荡器配置使能为 I/O 引脚; 否则, 它们将读为 0。
 注 2: 只有在 PIC18F4X31 器件中 ANS5 至 ANS8 才可用。

PIC18F2331/2431/4331/4431

10.2 PORTB、TRISB 和 LATB 寄存器

PORTB 是一个 8 位宽的双向端口。相应的数据方向寄存器为 TRISB。置位 TRISB 位 (= 1) 可以让相应 PORTB 引脚作为输入引脚（即将相应的输出驱动器置于高阻态）。清零 TRISB 位 (= 0) 将使相应的 PORTB 引脚作为输出引脚（即将输出锁存器的内容置于所选的引脚上）。

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

例 10-2: 初始化 PORTB

CLRF	PORTB	; Initialize PORTB by ; clearing output ; data latches
CLRF	LATB	; Alternate method ; to clear output ; data latches
MOVLW	0xCF	; Value used to ; initialize data ; direction
MOVWF	TRISB	; Set RB<3:0> as inputs ; RB<5:4> as outputs ; RB<7:6> as inputs

PORTB 的每个引脚都有内部弱上拉电路。只要对控制位 RBPU (INTCON2<7>) 清零, 就可以接通所有引脚的弱上拉功能。当端口引脚设置为输出时, 其弱上拉电路会自动切断。这些弱上拉电路在上电复位时被禁止。

四个 PORTB 引脚 (RB7:RB4) 具有电平变化中断功能。只有配置为输入的引脚会导致这种中断发生 (即当 RB7:RB4 的任何一个引脚被配置为输出时, 该引脚不再具有电平变化的中断功能)。将 RB7:RB4 引脚上的输入电平与前次读 PORTBe 而锁存的旧值进行比较。对 RB7:RB4 的不匹配输出进行“或”运算, 通过将标志位 RBIF (INTCON<0>) 置 1 产生 RB 端口电平变化中断。

此中断可以将器件从休眠状态唤醒。用户可以用以下方法在中断服务程序中清除中断请求。

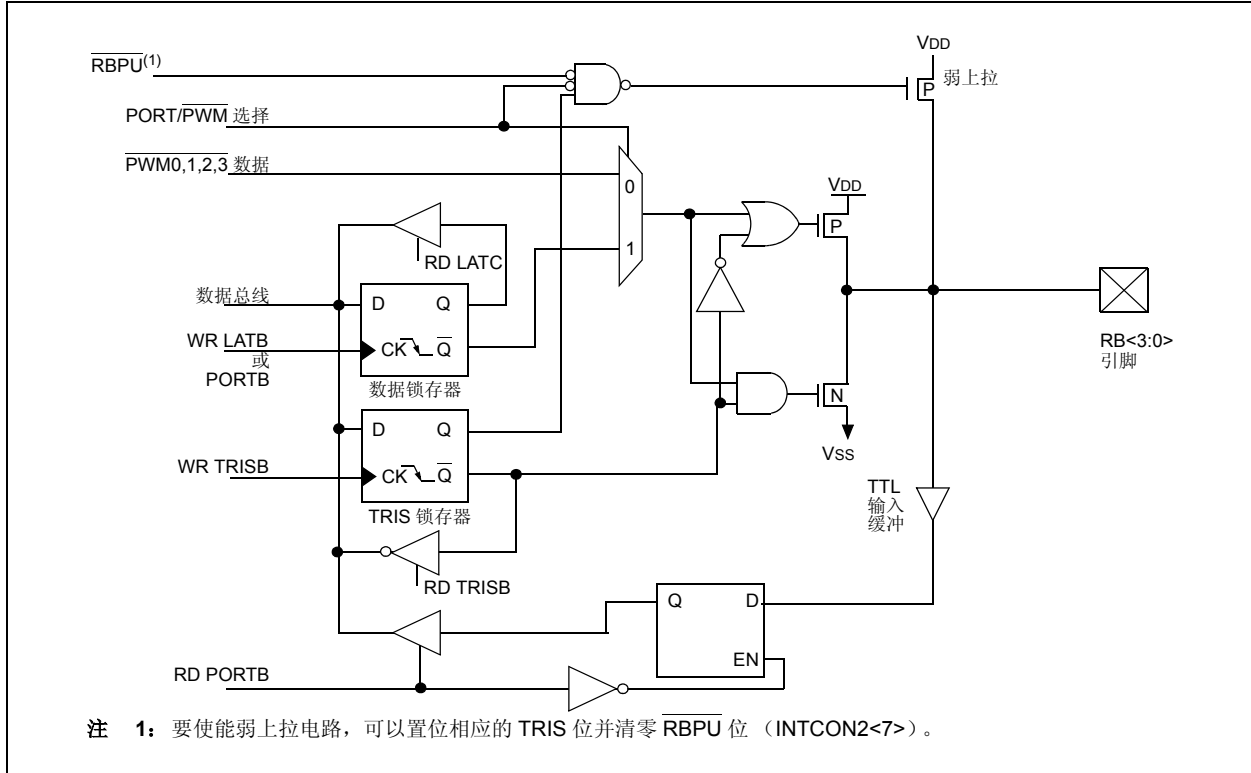
- 读或写 PORTB (除了使用 MOVFF (ANY), PORTB 指令)。这将结束电平变化的情况。
- 清零标志位 RBIF。

电平变化的情况会一直不断地将标志位 RBIF 置位。而读 PORTB 将结束这种电平变化的情况并允许清零 RBIF 标志位。

对于按键唤醒操作以及其他仅将 PORTB 用于电平变化中断功能的操作, 建议用该电平变化中断来实现。在使用电平变化中断功能时, 建议不要查询 PORTB 的状态。

RB<0:3> 和 RB4 引脚分别与 14 位 PWM 模块的 PWM<0:3> 和 PWM5 输出复用。可以通过配置位 PWM4MX 将 RB5 引脚配置为 PWM4 输出的复用引脚。

图 10-9: RB3:RB0 引脚框图



PIC18F2331/2431/4331/4431

图 10-10: RB4 框图

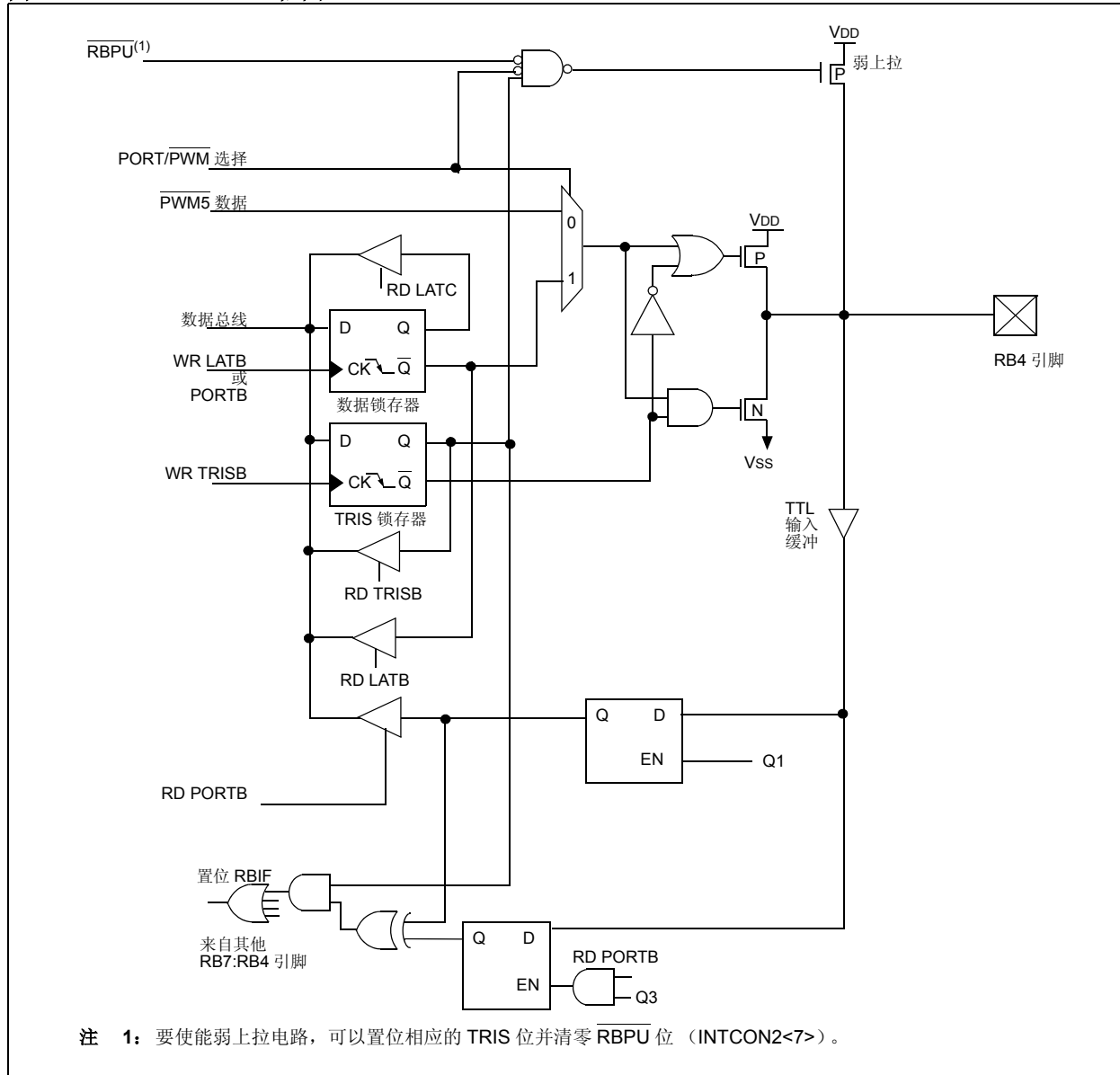
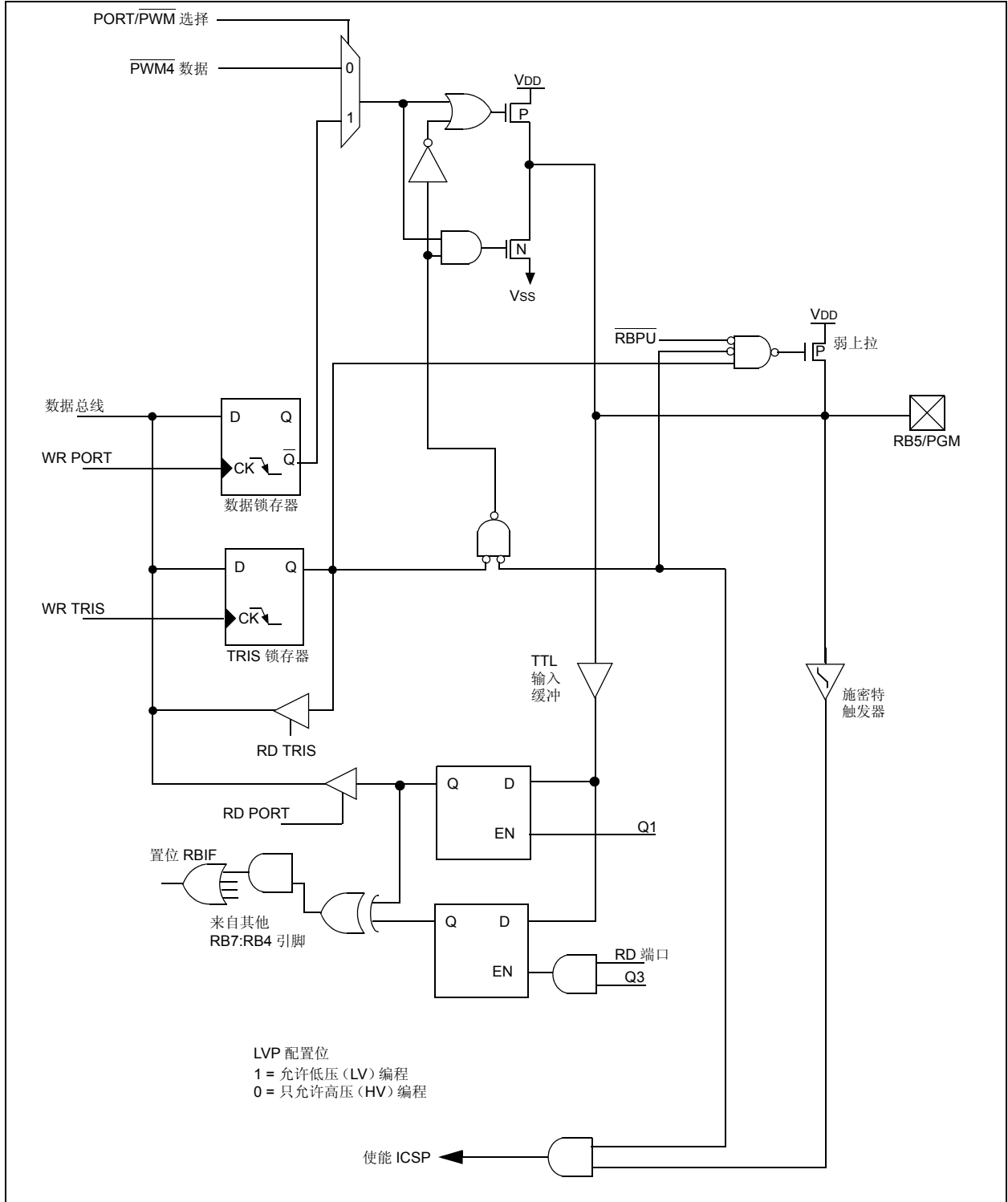
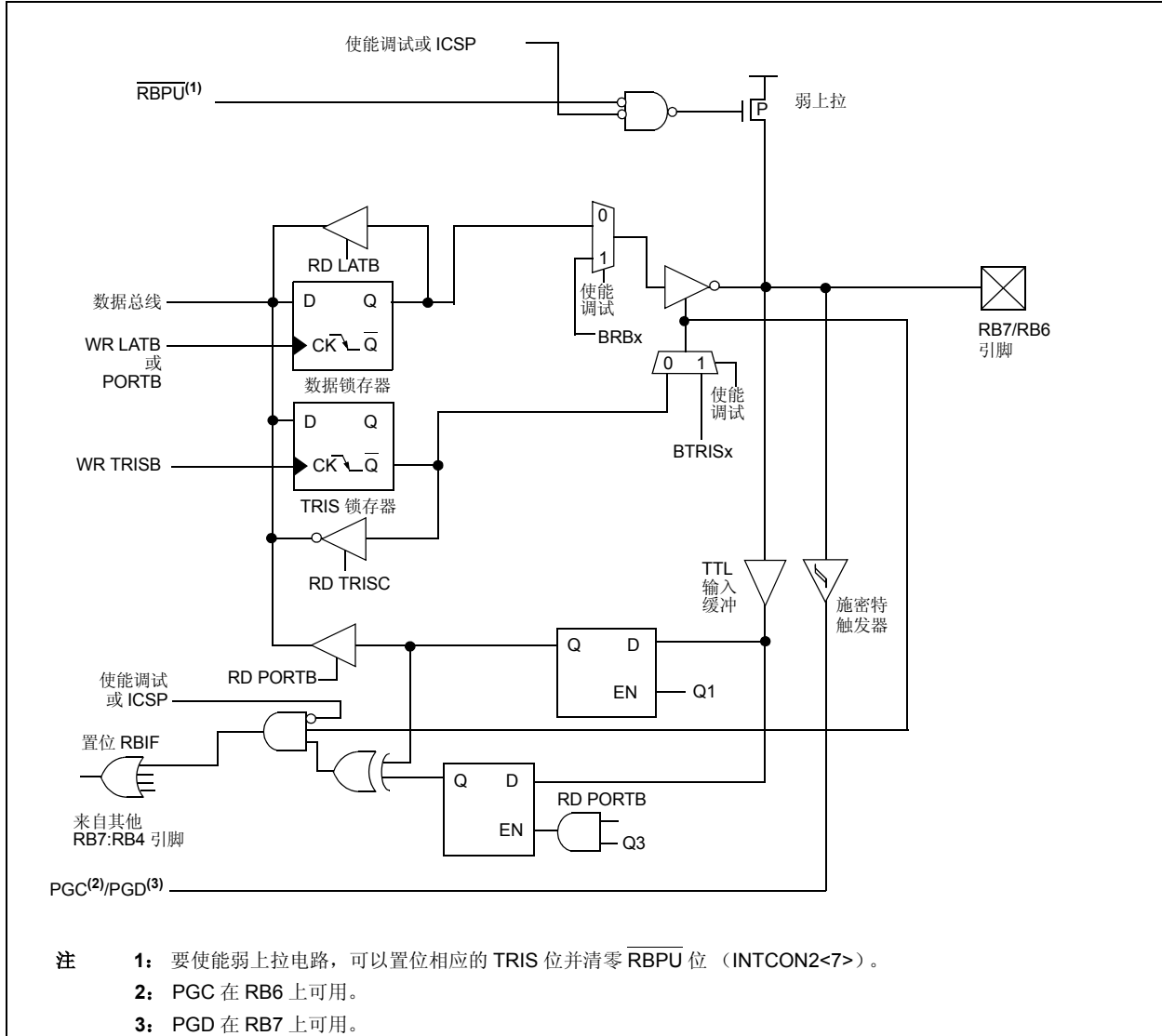


图 10-11: RB5 框图



PIC18F2331/2431/4331/4431

图 10-12: RB7:RB6 引脚框图



PIC18F2331/2431/4331/4431

表 10-3: PORTB 功能

名称	Bit #	缓冲器	功能
RB0/PWM0	bit 0	TTL ⁽¹⁾	输入 / 输出引脚或 PCPWM 输出 PWM0。 内部软件可编程的弱上拉电路。
RB1/PWM1	bit 1	TTL ⁽¹⁾	输入 / 输出引脚或 PCPWM 输出 PWM1。 内部软件可编程的弱上拉电路。
RB2/PWM2	bit 2	TTL ⁽¹⁾	输入 / 输出引脚或 PCPWM 输出 PWM2。 内部软件可编程的弱上拉电路。
RB3/PWM3	bit 3	TTL ⁽¹⁾	输入 / 输出引脚或 PCPWM 输出 PWM3。 内部软件可编程的弱上拉电路。
RB4/KBI0/PWM5	bit 4	TTL	输入 / 输出引脚（带电平变化中断）或 PCPWM 输出 PWM5。 内部软件可编程的弱上拉电路。
RB5/KBI1/PWM4/ PGM	bit 5	TTL/ST ⁽²⁾	输入 / 输出引脚（带电平变化中断）或 PCPWM 输出 PWM4。 内部软件可编程的弱上拉电路。 低压 ICSP 使能引脚。
RB6/KBI2/PGC	bit 6	TTL/ST ⁽²⁾	输入 / 输出引脚（带电平变化中断）。 内部软件可编程的弱上拉电路。 串行编程时钟。
RB7/KBI3/PGD	bit 7	TTL/ST ⁽²⁾	输入 / 输出引脚（带电平变化中断）。 内部软件可编程的弱上拉电路。 串行编程数据。

图注: TTL = TTL 输入, ST = 施密特触发器输入

注 1: 配置为数字 I/O 时缓冲器为 TTL 输入。

注 2: 在串行编程模式下使用时, 该缓冲器是施密特触发器输入。

表 10-4: 与 PORTB 相关的寄存器综述

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx qqqq	uuuu uuuu
LATB	LATB 数据输出寄存器								xxxx xxxx	uuuu uuuu
TRISB	PORTB 数据方向寄存器								1111 1111	1111 1111
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	1111 -1-1	1111 -1-1
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	11-0 0-00	11-0 0-00

图注: x= 未知, u= 不变, q= 根据情况而变。阴影单元格表示 PORTB 未使用。

PIC18F2331/2431/4331/4431

10.3 PORTC、TRISC 和 LATC 寄存器

PORTC 是一个 8 位宽的双向端口。相应的数据方向寄存器为 TRISC。置位 TRISC 位 (= 1) 可以让相应 PORTC 引脚作为输入引脚（即将相应的输出驱动器置于高阻态）。清零 TRISC 位 (= 0) 将使相应的 PORTC 引脚作为输出引脚（即将输出锁存器的内容置于所选的引脚上）。

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

PORTC 与几种外设功能复用（表 10-5）。这些引脚具有施密特触发器输入缓冲器。

当外设功能使能时，在定义每个 PORTC 引脚的 TRIS 位时应小心。有些外设会覆盖相应引脚的 TRIS 位设置而将引脚直接定义为输出，而另一些外设也会覆盖相应引脚的 TRIS 位设置，但却将引脚直接定义为输入。用户应该参考相应的外设章节来正确设置 TRIS 位。

注： 在上电复位时，这些引脚被配置为数字输入。

外设覆盖会影响 TRISC 寄存器的内容。虽然外设器件可能会覆盖一个或多个引脚，但是读 TRISC 结果总是会返回当前内容。

外部中断 IN0、INT1 和 INT2 分别位于 RC3、RC4 和 RC5。

SSP 复用接口引脚 SDI/SDA、SCK/SCL 和 SDO 分别位于 RC4、RC5 和 RC7 引脚。

通过使用 CONFIG3L 寄存器的 SSPMX 位可以将这些引脚与 PORTC 和 PORTD 复用。

USART 引脚 RX/DT 和 TX/CK 分别位于 RC7 和 RC6 引脚。

Timer5 复用外部时钟输入 T5CKI 和 TMR0 复用外部时钟输入 T0CKI 位于 RC3 引脚，利用 CONFIG3L 寄存器中的 EXCLKMX 配置位可以将它们与 PORTD（RD0）引脚复用。14 位 PWM 模块的故障输入 FLTA 和 FLTB 分别位于 RC1 和 RC2。RC1 上的 FLTA 输入通过利用 FLTAMX 位与 RD4 复用。

例 10-3: 初始化 PORTC

```

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

图 10-13: RC0 框图

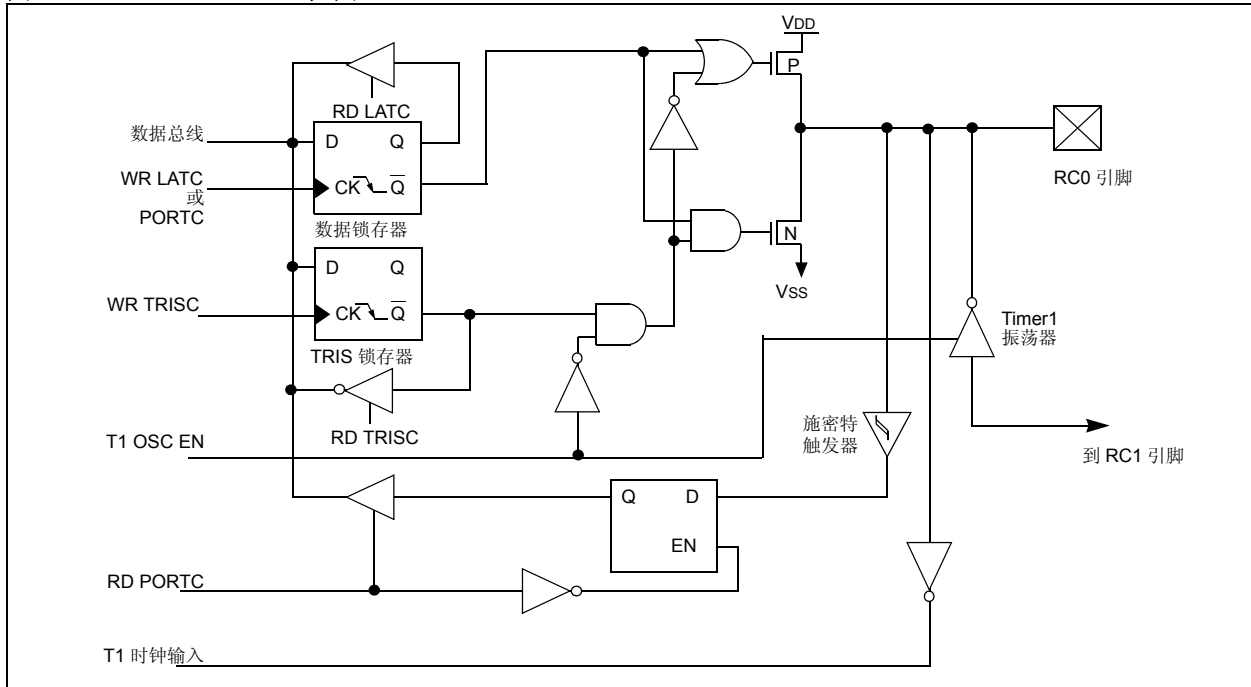


图 10-14: RC1 框图

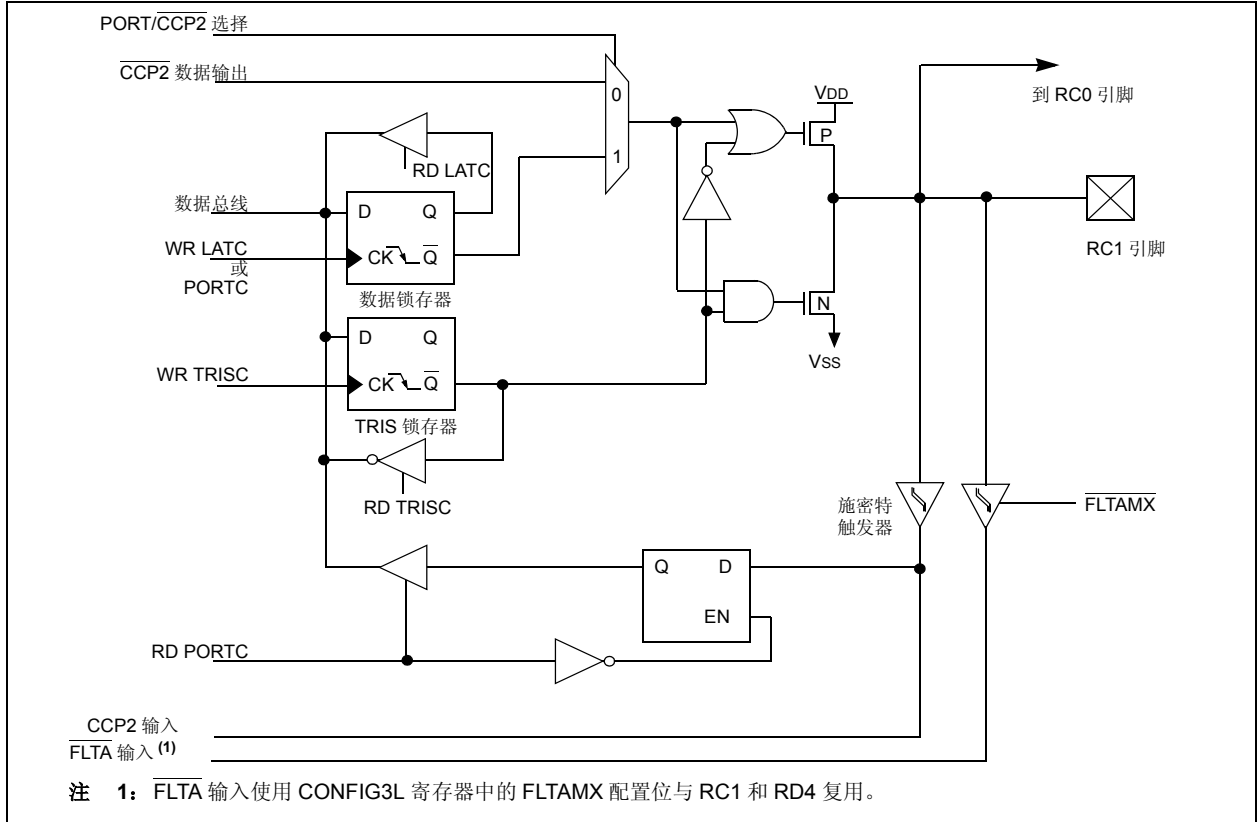
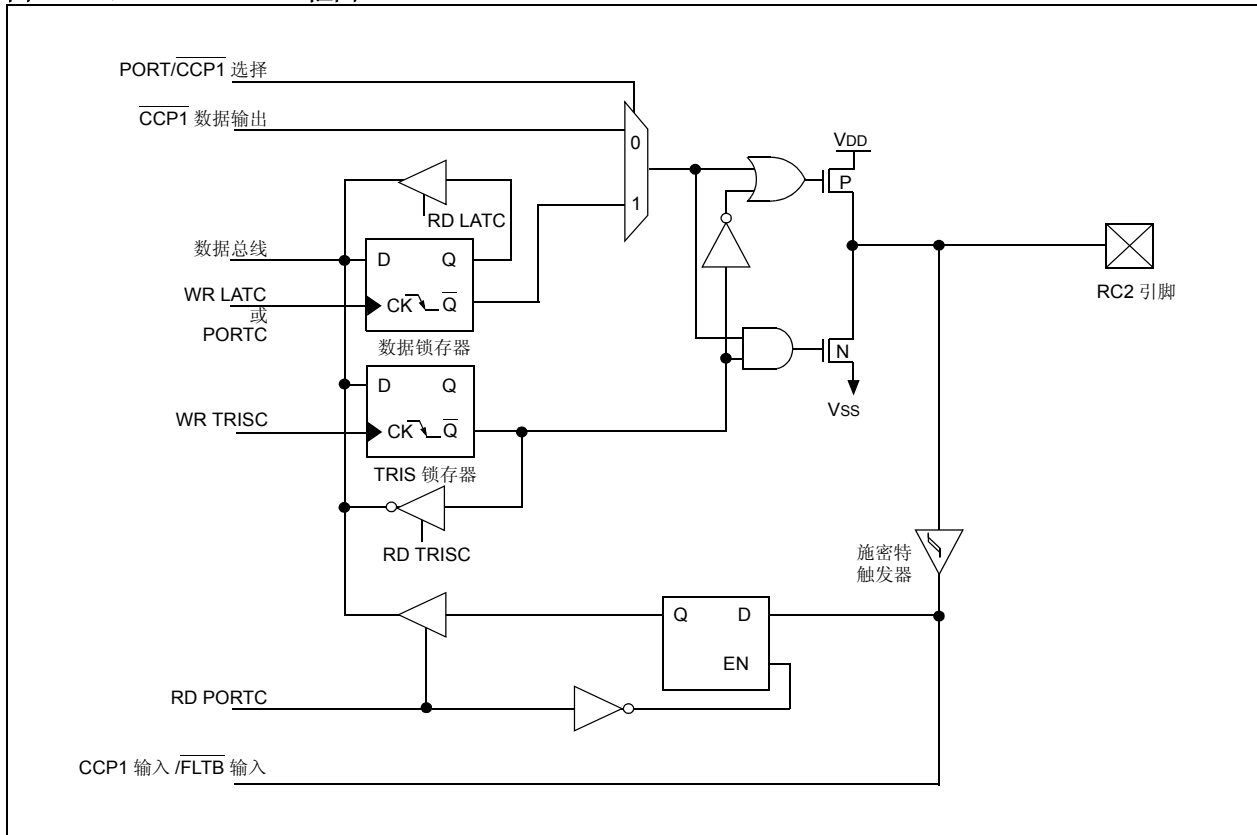


图 10-15: RC2 框图



PIC18F2331/2431/4331/4431

图 10-16: RC3 框图

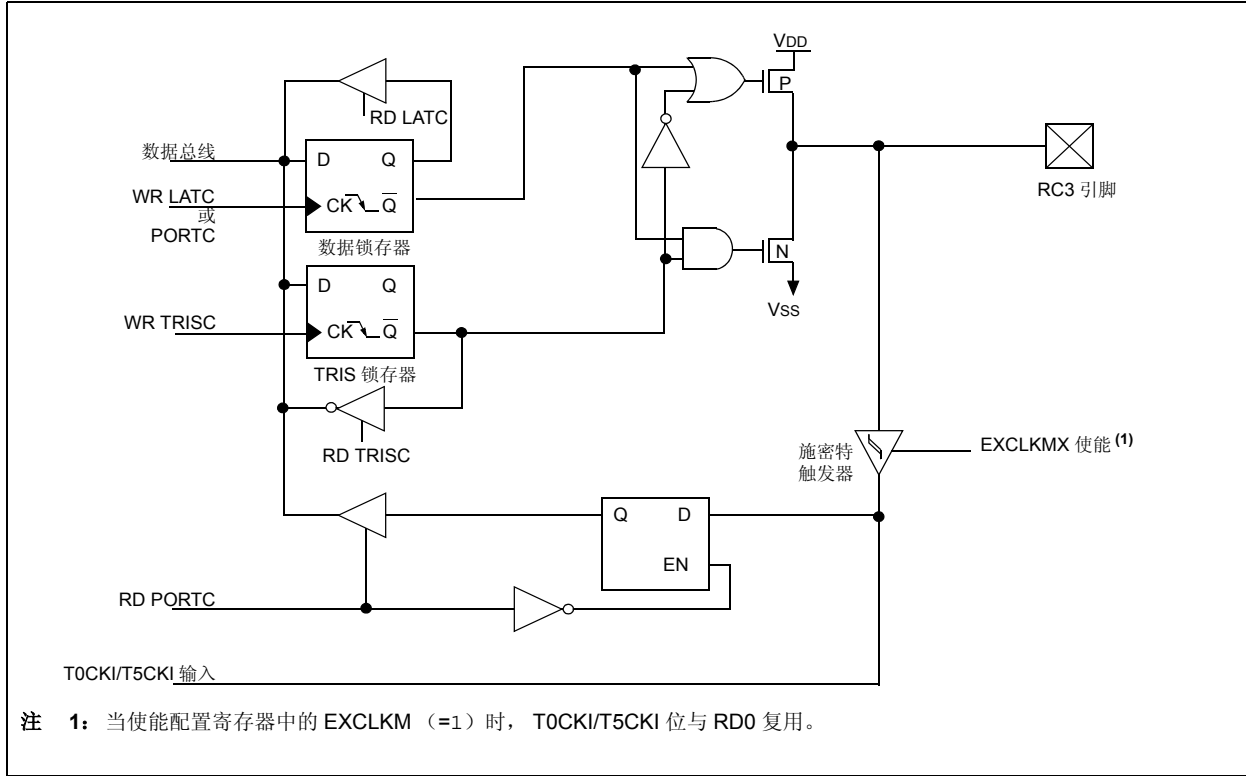


图 10-17: RC4 框图

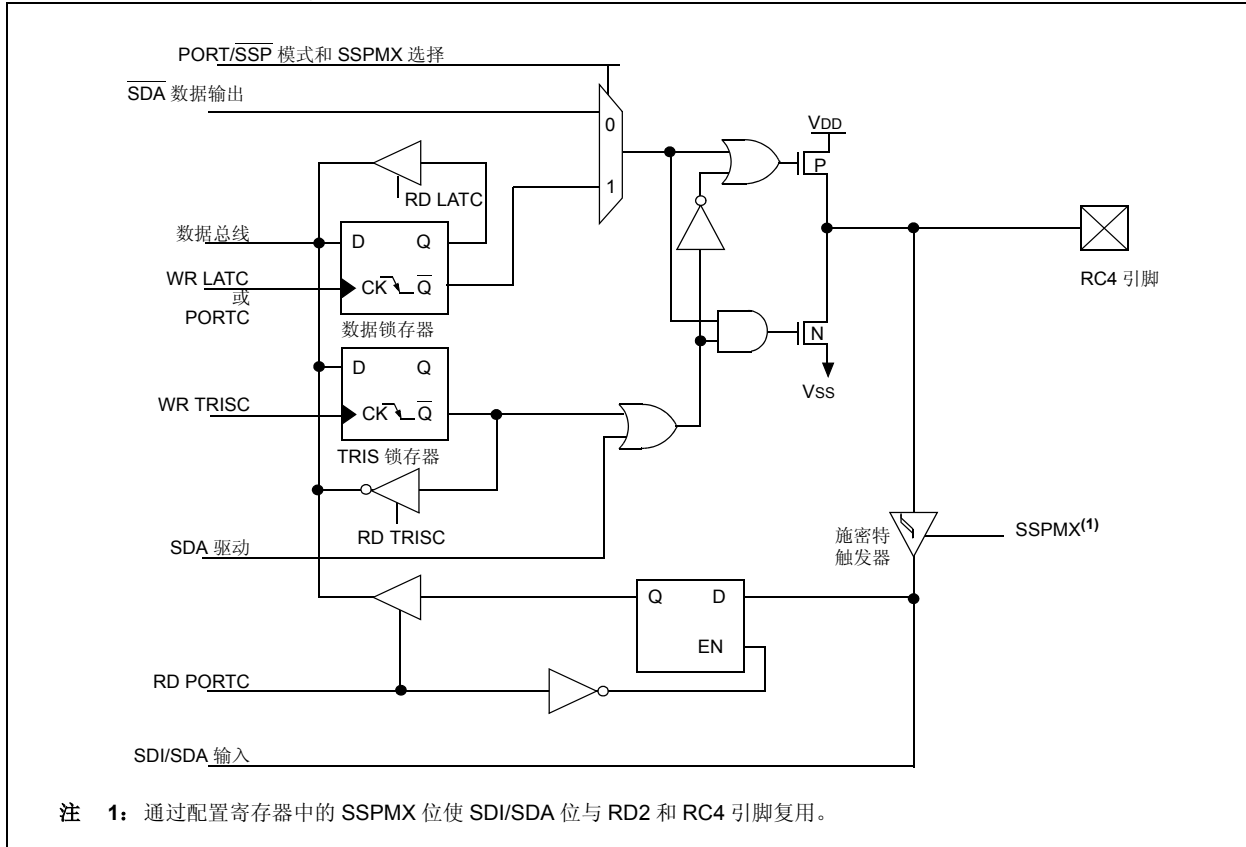


图 10-18: RC5 框图

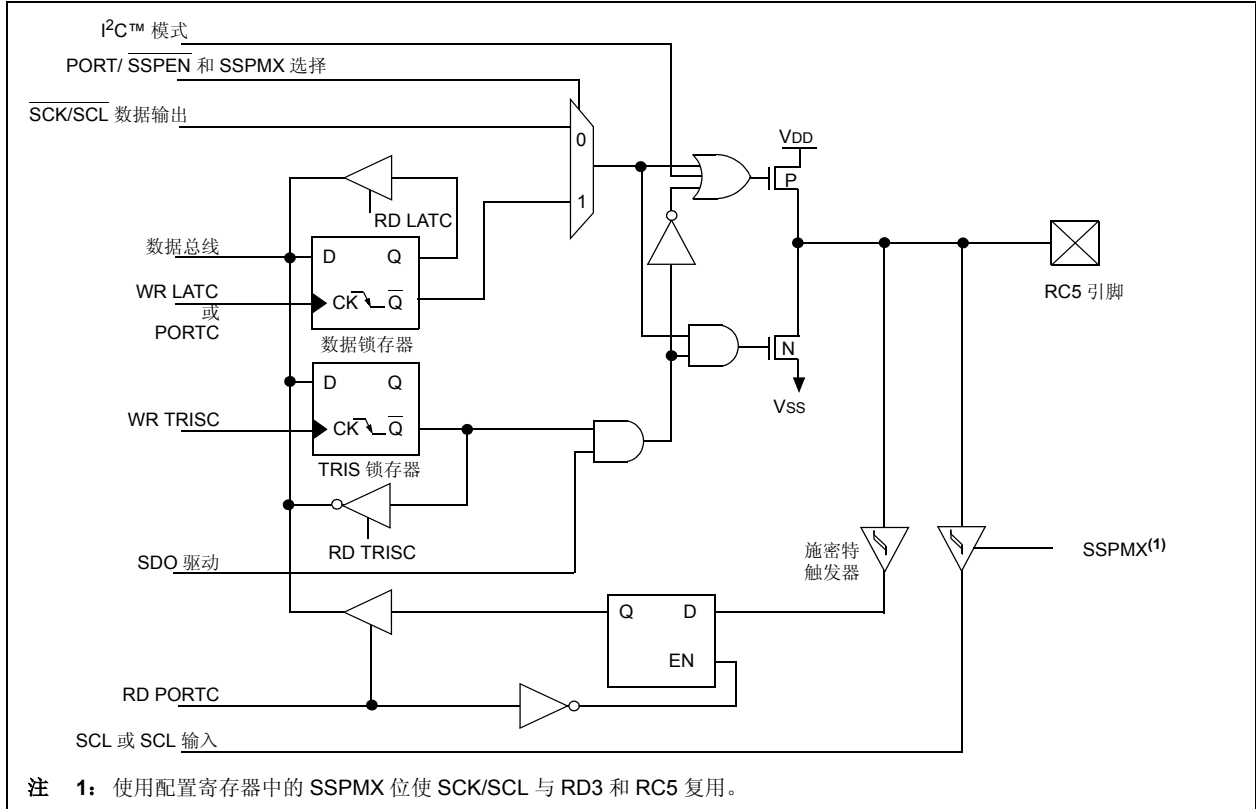
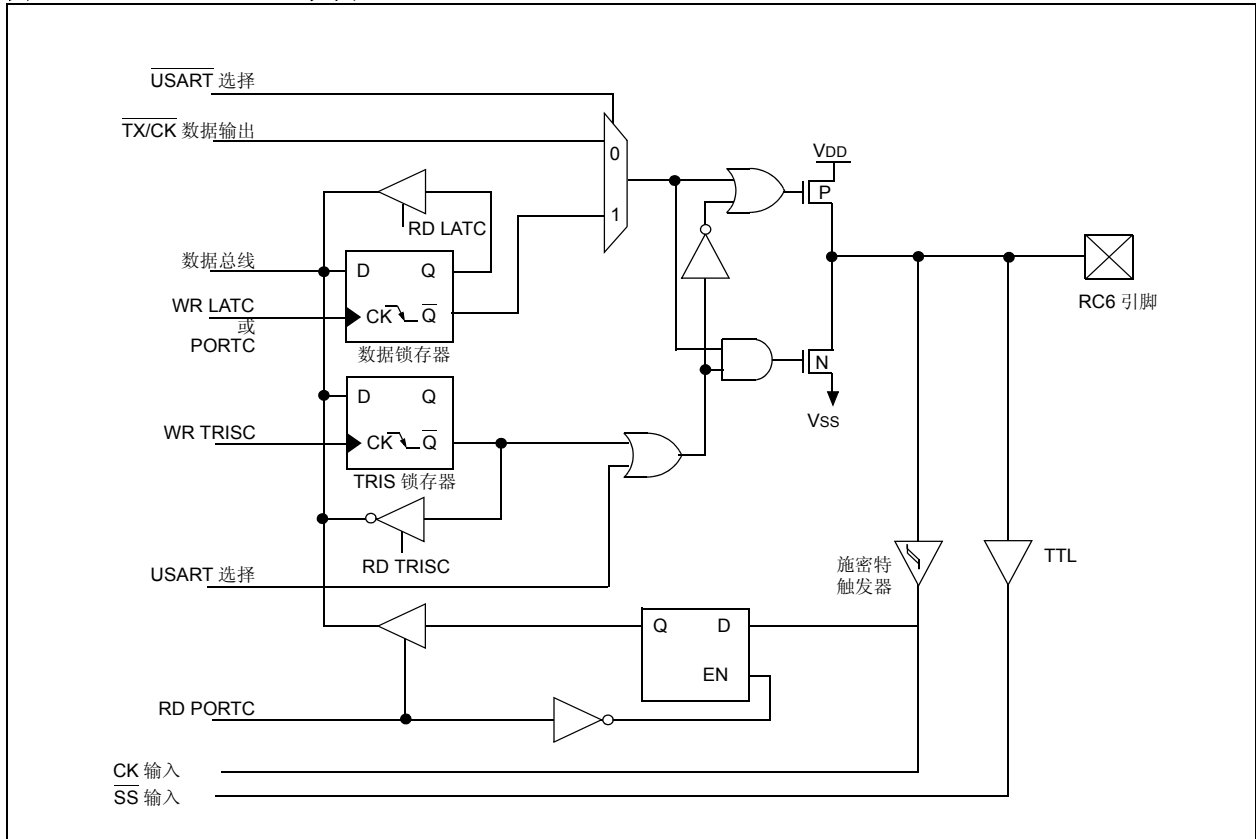
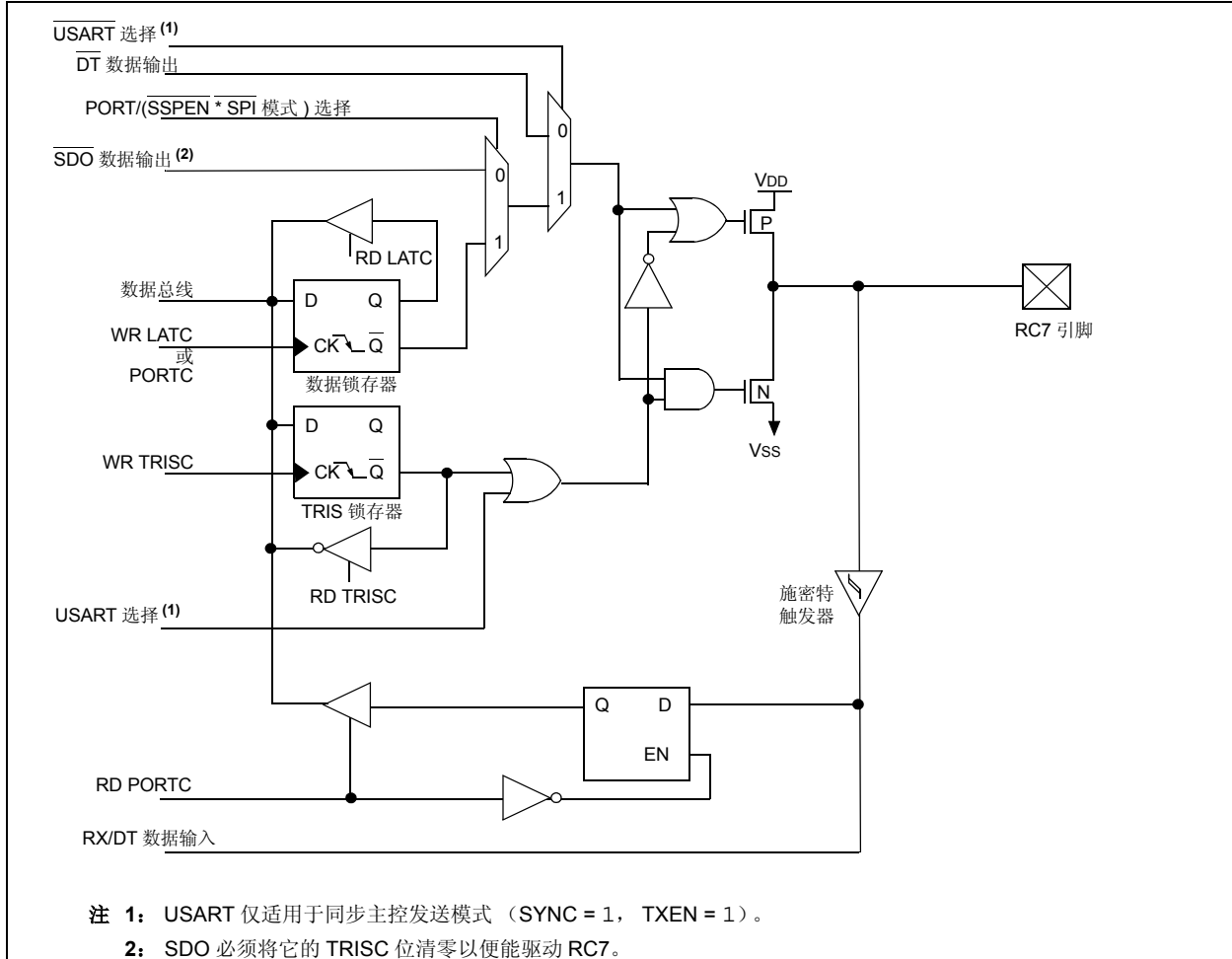


图 10-19: RC6 框图



PIC18F2331/2431/4331/4431

图 10-20: RC6 框图



PIC18F2331/2431/4331/4431

表 10-5: PORTC 功能

名称	Bit #	缓冲器类型	功能
RC0/T1OSO/T1CKI	bit 0	ST	输入 / 输出端口引脚或 Timer1 振荡器输出 / Timer1 时钟输入。
RC1/T1OSI/CCP2/ FLTA	bit 1	ST/CMOS	输入 / 输出端口引脚、Timer1 振荡器输入、当 CCP2MX 配置位被禁止时为 Capture2 输入 / Compare2 输出 / PWM 输出、或 FLTA 输入。
RC2/CCP1/FLTB	bit 2	ST	输入 / 输出端口引脚、Capture1 输入 / Compare1 输出 / PWM1 输出或 FLTB 输入。
RC3/T0CKI/T5CKI/ INT0	bit 3	ST	输入 / 输出端口引脚、Timer0 和 Timer5 复用时钟输入或外部中断 0。
RC4/INT1/SDI/SDA	bit 4	ST	输入 / 输出端口引脚、SPI 数据输入、I ² C 数据 I/O 或外部中断 1。
RC5/INT2/SCK/SCL	bit 5	ST	输入 / 输出端口引脚或同步串行端口时钟 I/O 或外部中断 2。
RC6/TX/CK/SS	bit 6	ST	输入 / 输出端口引脚、EUSART 异步发送、EUSART 同步时钟或 SPI 从动选择输入。
RC7/RX/DT/SDO	bit 7	ST	输入 / 输出端口引脚、EUSART 异步接收、EUSART 同步数据或 SPI 数据输出。

图注: ST= 施密特触发器输入

表 10-6: 与 PORTC 相关的寄存器综述

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
LATC	LATC 数据输出寄存器								xxxx xxxx	uuuu uuuu
TRISC	PORTC 数据方向寄存器								1111 1111	1111 1111
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	1111 -1-1	1111 -1-1
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	11-0 0-00	11-0 0-00

图注: x= 未知, u= 不变

PIC18F2331/2431/4331/4431

10.4 PORTD、TRISD 和 LATD 寄存器

注： PORTD 只在 PIC18F4X31 器件上可用。

PORTD 是一个 8 位宽的双向端口。相应的数据方向寄存器是 TRISD。置位 TRISD 位 (= 1) 可以让相应 PORTD 引脚作为输入引脚 (即将相应的输出驱动器置于高阻态)。清零 TRISD 位 (= 0) 将使相应的 PORTD 引脚作为输出引脚 (即将输出锁存器的内容置于所选的引脚上)。

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

PORTD 上的所有引脚都通过施密特触发器输入缓冲器实现。每个引脚被单独设置为输入或输出。

注： 在上电复位时，这些引脚被配置为数字输入。

PORTD 包含 PWM<7:6> 的互补第四通道 PWM 输出。PMW4 是 PMW5 (第三通道) 的互补输出，PMW4 与 RB5 引脚复用。在 RB5 上使用低压编程引脚 (PGM) 时使用 CONFIG3L 中的 PWM4MX 配置位可以将该输出用作 PWM4 的复用输出。

使用 CONFIG3L 中的 SSPMX 配置位可以将 RD1、RD2 和 RD3 用作 SDO、SDI/SDA 和 SCK/SCL 的复用输出。

使用 CONFIG3L 中的 FLTAMX 配置位可以将 RD4 用作 FLTA 的复用输出。

例 10-4: 初始化 PORTD

```

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

图 10-21: RD7:RD6 引脚框图

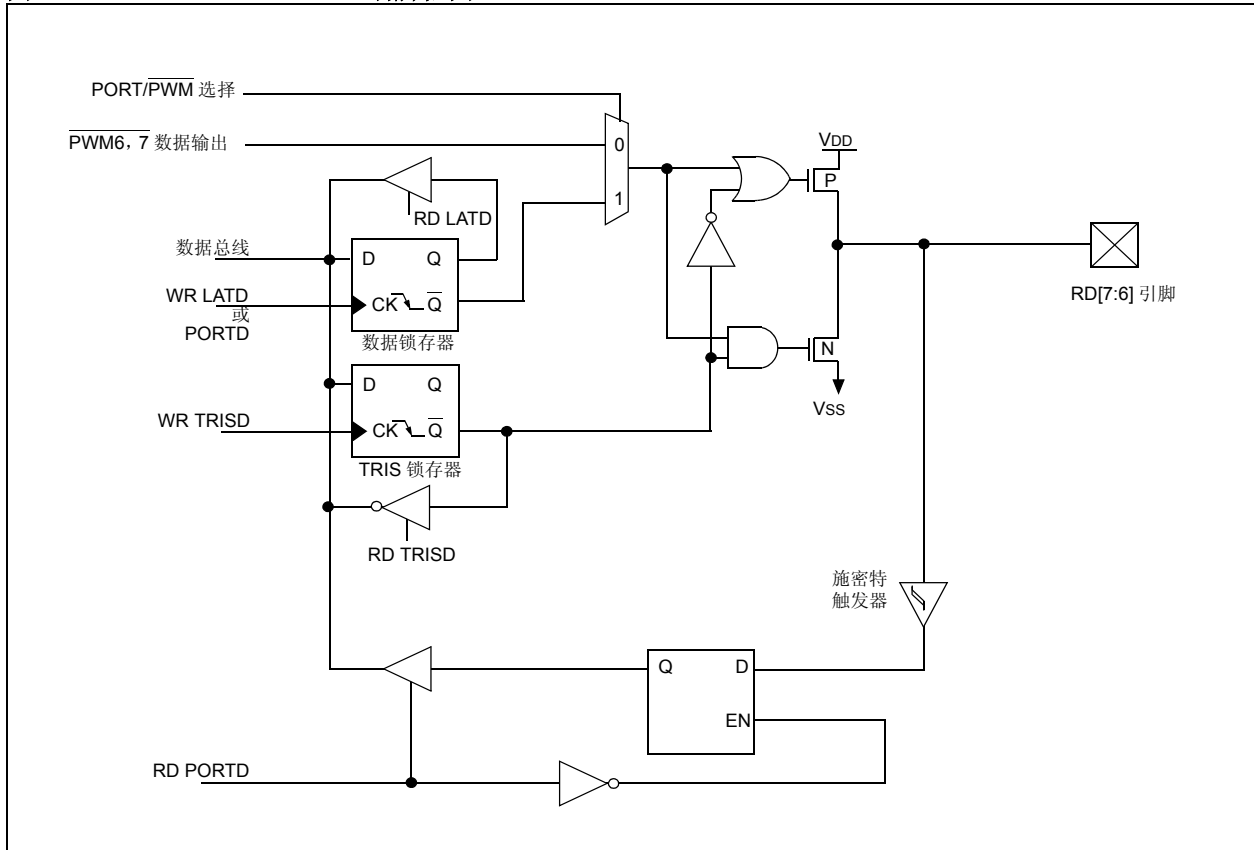


图 10-22: RD5 框图

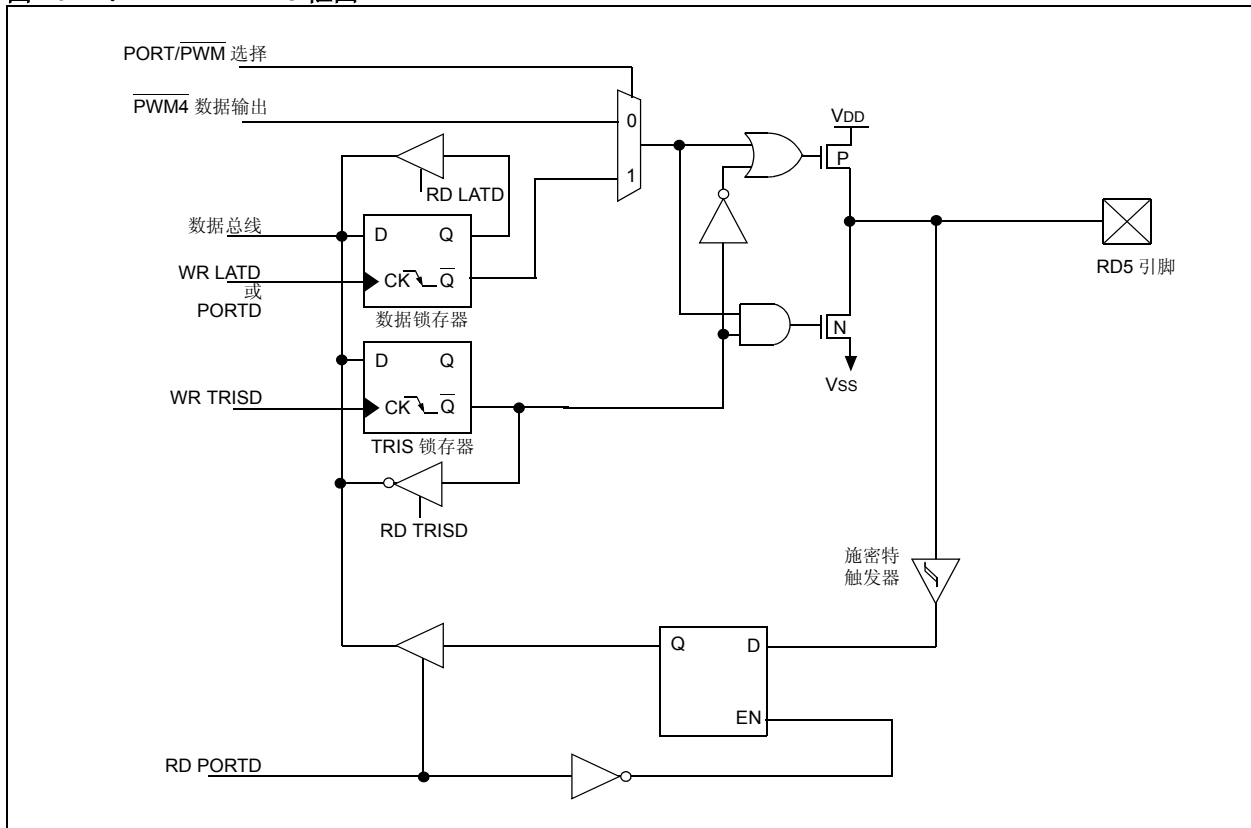
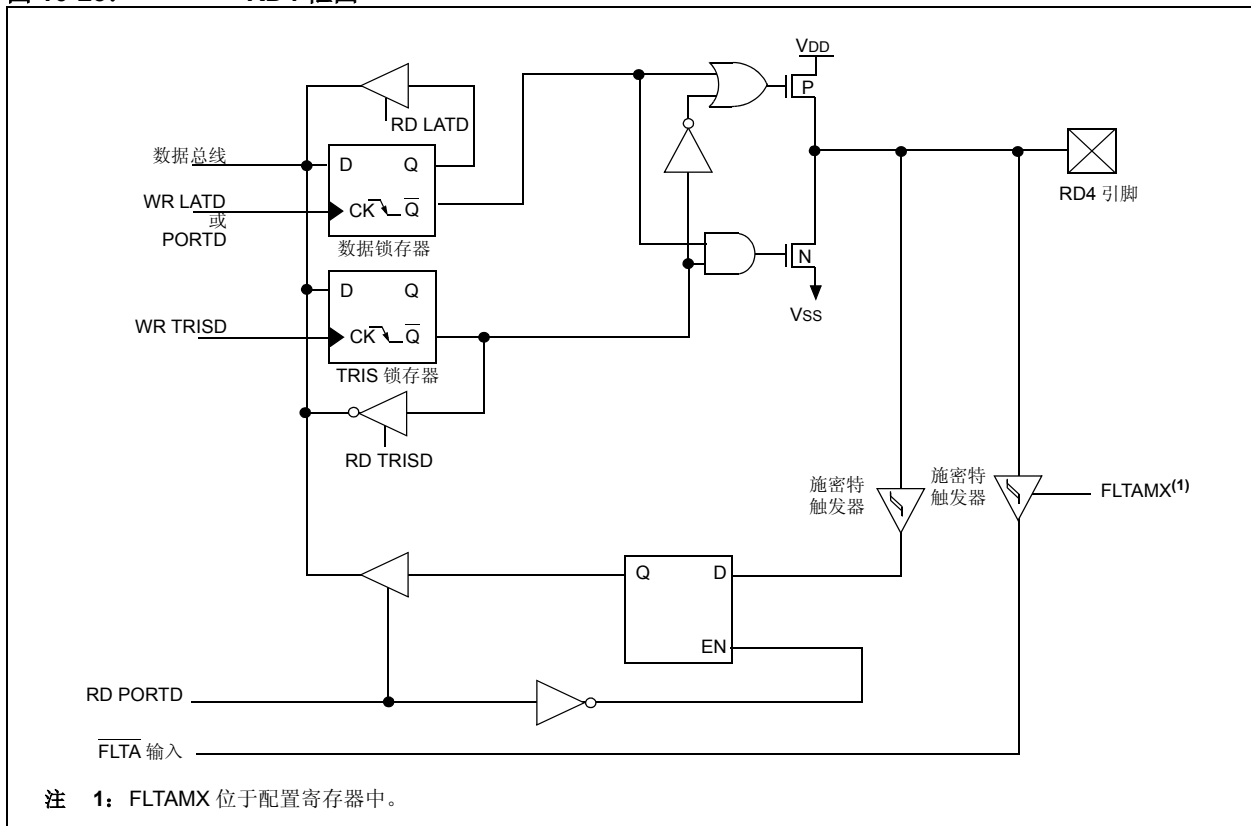


图 10-23: RD4 框图



注 1: FLTAMX 位于配置寄存器中。

PIC18F2331/2431/4331/4431

图 10-24: RD3 框图

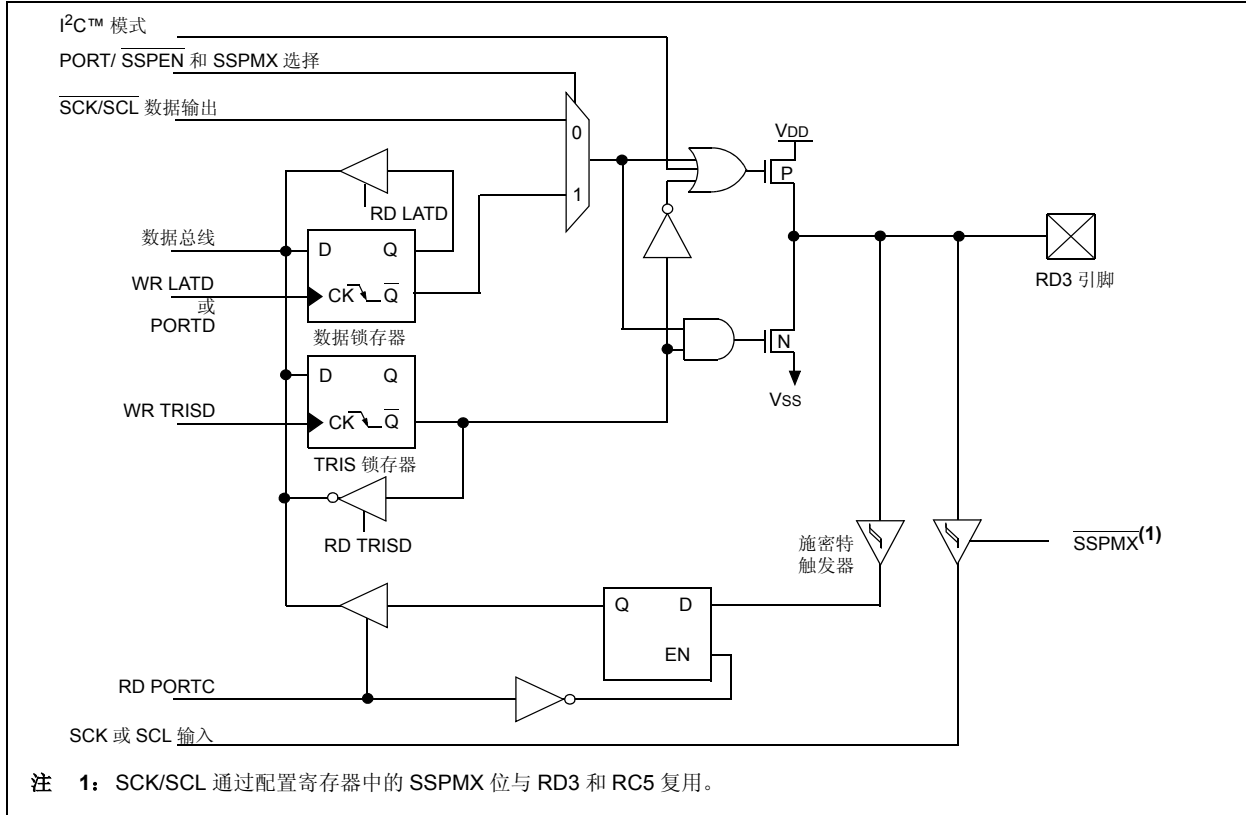


图 10-25: RD2 框图

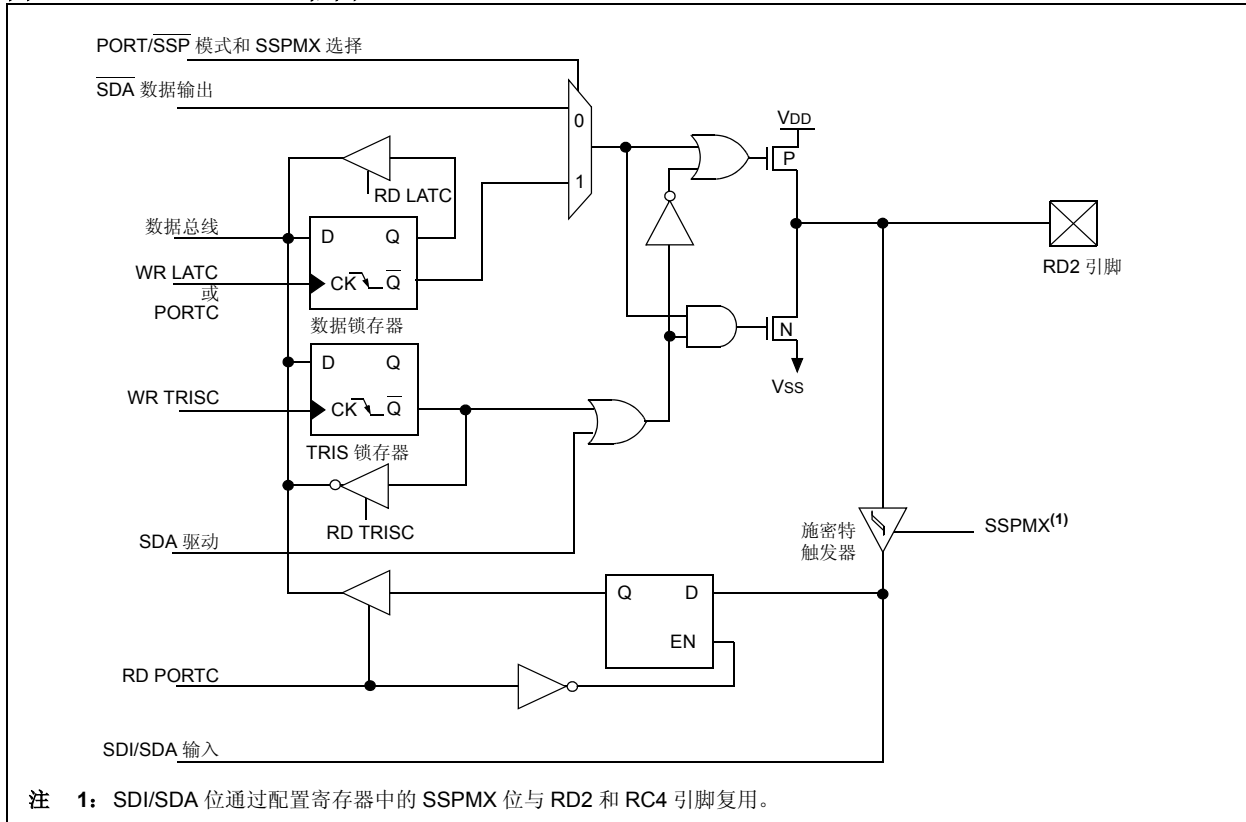


图 10-26: RD1 框图

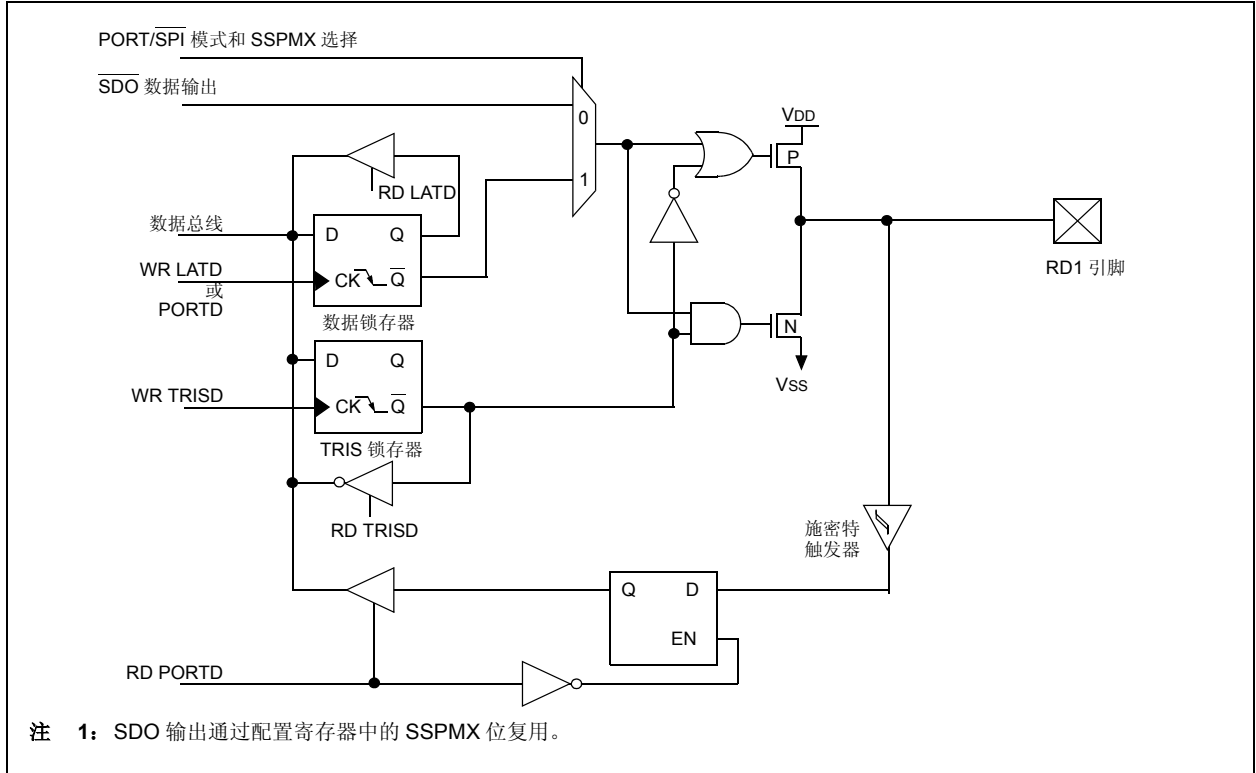
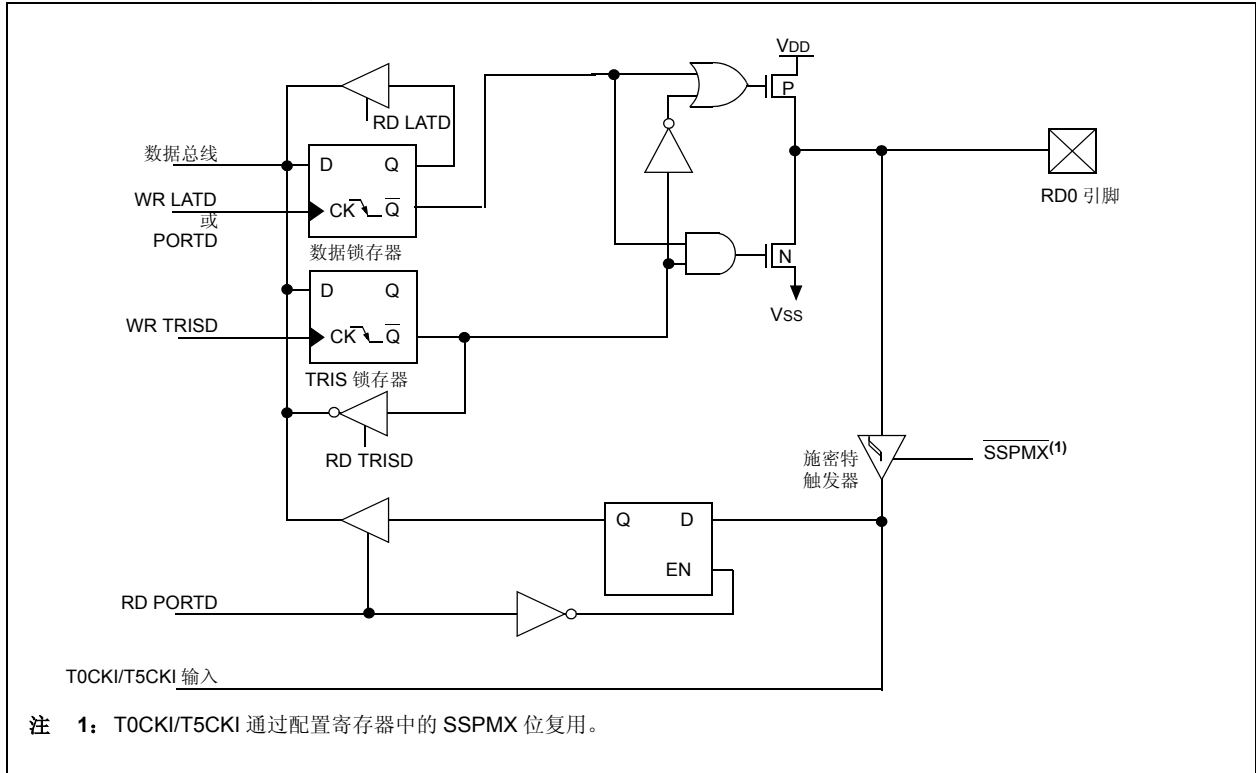


图 10-27: RD0 框图



PIC18F2331/2431/4331/4431

表 10-7: PORTD 功能

名称	Bit #	缓冲器类型	功能
RD0/T0CKI/T5CKI	bit 0	ST	输入 / 输出端口引脚。
RD1/SDO	bit 1	ST	输入 / 输出端口引脚。
RD2/SDI/SDA	bit 2	ST	输入 / 输出端口引脚。
RD3/SCK/SCL	bit 3	ST	输入 / 输出端口引脚。
RD4/FLTA	bit 4	ST	输入 / 输出端口引脚。
RD5/PWM4	bit 5	ST	输入 / 输出端口引脚或 PCPWM 输出 PWM4。
RD6/PWM6	bit 6	ST	输入 / 输出端口引脚或 PCPWM 输出 PWM6。
RD7/PWM7	bit 7	ST	输入 / 输出端口引脚或 PCPWM 输出 PWM7。

图注: ST= 施密特触发器输入, TTL=TTL 输入

表 10-8: 与 PORTD 相关的寄存器综述

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
LATD	LATD 数据输出寄存器								xxxx xxxx	uuuu uuuu
TRISD	PORTD 数据方向寄存器								1111 1111	1111 1111

图注: x= 未知, u= 不变, -= 未实现位, 读作 0。阴影单元格表示 PORTD 未使用。

PIC18F2331/2431/4331/4431

10.5 PORTE、TRISE 和 LATE 寄存器

注： PORTE 只在 PIC18F4X31 器件上可用。

PORTE 是一个 4 位宽的双向端口。三个引脚（RE0/AN6、RE1/AN7 和 RE2/AN8）分别被配置为输入或输出。这些引脚具有施密特触发器输入缓冲器。当选择为模拟输入时，这些引脚将读作 0。

相应的数据方向寄存器是 TRISE。置位 TRISE 位（= 1）可以让相应 PORTE 引脚作为输入引脚（即将相应的输出驱动器置于高阻态）。清零 TRISE 位（= 0）将使相应的 PORTE 引脚作为输出引脚（即将输出锁存器的内容置于所选的引脚上）。

即使 TRISE 用作模拟输入的时候，TRISE 寄存器仍会控制 RE 引脚的方向。当它们用作模拟输入的时候，用户必须确保把引脚配置为输入。

注： 在上电复位时，RE2:RE0 被配置为模拟输入。

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

PORTE 的第四个引脚（ $\overline{\text{MCLR}}/\text{VPP}/\text{RE3}$ ）是一个只允许输入的引脚。其操作受配置寄存器 3H 中的 MCLRE 配置位（CONFIG3H<7>）控制。如果它被选为一个端口引脚（MCLRE = 0），它就只有数字输入引脚的功能。这样，TRIS 或 LAT 位就与它的操作不相关了。否则，它将作为器件的主清零输入。无论是哪种配置，RE3 在编程过程中还作为编程电压输入。

注： 在上电复位时，仅当主清零功能被禁止时，RE3 才能被使能为数字输入。

例 10-5: 初始化 PORTE

```
CLRF    PORTE        ; Initialize PORTE by
                    ; clearing output
                    ; data latches
CLRF    LATE         ; Alternate method
                    ; to clear output
                    ; data latches
MOVLW  0x3F         ; Configure A/D
MOVWF  ANSEL0       ; for digital inputs
bcf    ANSEL1, 0    ;
MOVLW  0x03         ; Value used to
                    ; initialize data
                    ; direction
MOVWF  TRISE        ; Set RE<0> as input
                    ; RE<1> as output
                    ; RE<2> as input
```

10.5.1 28 引脚器件中的 PORTE

对于 PIC18F2X31 器件来说，PORTE 只有在主清零功能被禁止（CONFIG3H<7> = 0）时才可用。在这种情况下，PORTE 是一个只允许输入的端口，仅包含 RE3 一个引脚。此引脚的工作状态如前所述。

PIC18F2331/2431/4331/4431

图 10-28: RE2:RE0 框图

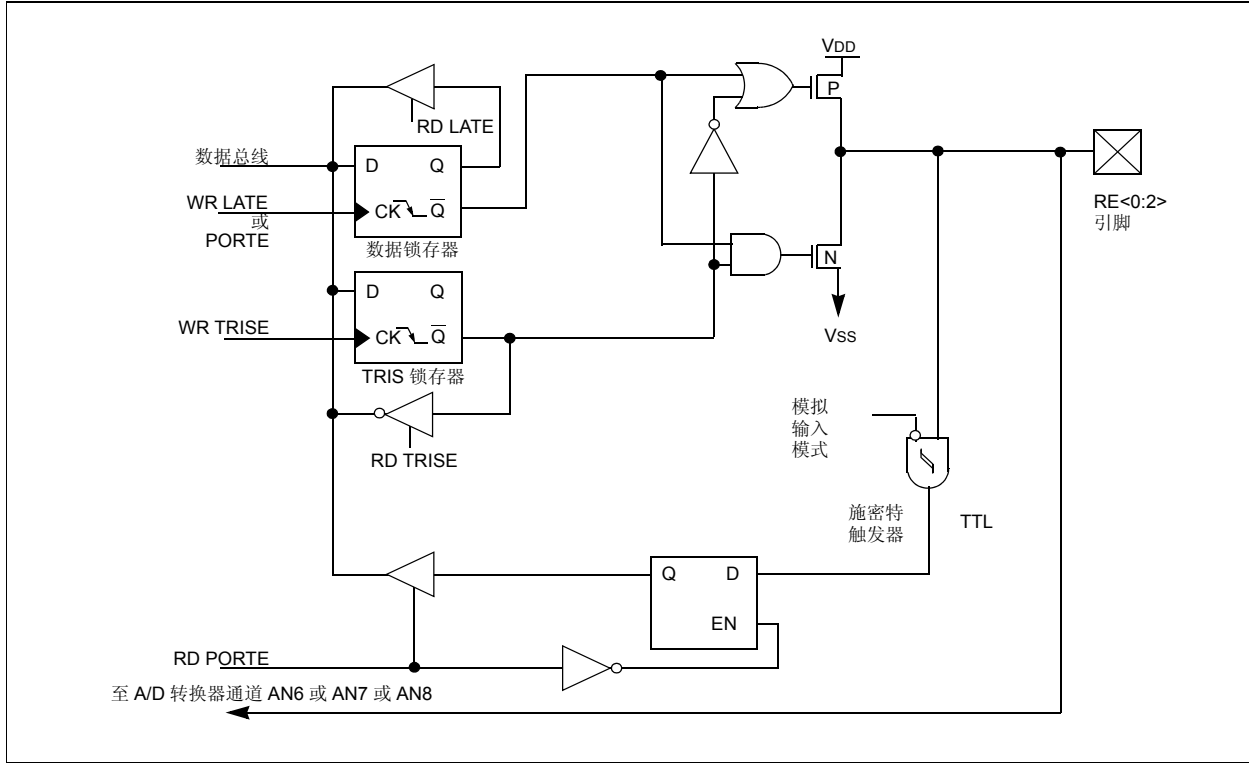
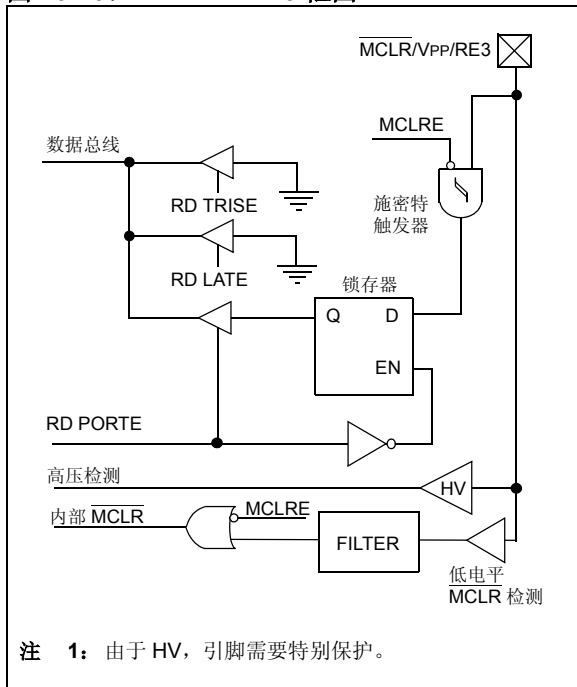


图 10-29: RE3 框图



注 1: 由于 HV, 引脚需要特别保护。

PIC18F2331/2431/4331/4431

寄存器 10-1:

TRISE 寄存器

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1
—	—	—	—	—	TRISE2	TRISE1	TRISE0
bit 7					bit 0		

- bit 7 未实现位: 读作 0
- bit 6 未实现位: 读作 0
- bit 5 未实现位: 读作 0
- bit 4 未实现位: 读作 0
- bit 3 未实现位: 读作 0
- bit 2 **TRISE2:** RE2 方向控制位
1= 输入
0= 输出
- bit 1 **TRISE1:** RE1 方向控制位
1= 输入
0= 输出
- bit 0 **TRISE0:** RE0 方向控制位
1= 输入
0= 输出

图注:

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

PIC18F2331/2431/4331/4431

表 10-9: PORTE 功能

名称	Bit #	缓冲器类型	功能
RE0/AN6	bit 0	ST	输入 / 输出端口引脚，模拟输入。
RE1/AN7	bit 1	ST	输入 / 输出端口引脚，模拟输入。
RE2/AN8	bit 2	ST	输入 / 输出端口引脚，模拟输入。
MCLR/VPP/RE3	bit 3	ST	只允许作为输入的端口引脚或编程电压输入（如果 MCLR 被禁止）；主清零输入或编程电压输入（如果 MCLR 被使能）。

图注： ST= 施密特触发器输入， TTL=TTL 输入

表 10-10: 与 PORTE 相关的寄存器综述

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
PORTE	—	—	—	—	RE3 ⁽¹⁾	RE2	RE1	RE0	---- q000	---- q000
LATE	—	—	—	—	—	LATE 数据输出寄存器			---- -xxx	---- -uuu
TRISE	—	—	—	—	—	PORTE 数据方向位			---- -111	---- -111
ANSEL0	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
ANSEL1	ANS15	ANS14	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	---- ---0	---- ---0

图注： x= 未知， u= 不变， -= 未实现位，读作 0， q= 根据情况而变。
阴影单元格表示 PORTE 未使用。

注 1： 只有在主清零功能被禁止（CONFIG3H<7> = 0）时才会实现。

PIC18F2331/2431/4331/4431

11.0 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
- 可软件编程的专用 8 位预分频器
- 可选内部或外部时钟源
- 8 位模式下 FFh 到 00h 的溢出中断, 16 位模式下 FFFFh 到 0000h 的溢出中断
- 外部时钟的边沿选择

图 11-1 给出 8 位模式下 Timer0 模块的简化框图, 图 11-2 给出 16 位模式下 Timer0 模块的简化框图。

T0CON 寄存器 (寄存器 11-1) 是个可读写的寄存器, 控制 Timer0 的各个方面, 包括预分频的选择。

寄存器 11-1: T0CON: TIMER0 控制寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T016BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

- bit 7 **TMR0ON:** Timer0 开 / 关控制位
1= 使能 Timer0
0= 停止 Timer0
- bit 6 **T016BIT:** Timer0 16 位控制位
1= Timer0 配置为 8 位定时器 / 计数器
0= Timer0 配置为 16 位定时器 / 计数器
- bit 5 **T0CS:** Timer0 时钟源选择位
1= T0CKI 引脚上的脉冲信号触发
0= 内部指令周期时钟源 (CLKO)
- bit 4 **T0SE:** Timer0 时钟源边沿选择位
1= T0CKI 引脚下降沿触发递增
0= T0CKI 引脚上升沿触发递增
- bit 3 **PSA:** Timer0 预分频器分配位
1= 未分配 Timer0 预分频器。Timer0 时钟输入不经过预分频器处理。
0= 已分配 Timer0 预分频器。Timer0 时钟输入来自预分频器输出。
- bit 2-0 **T0PS2:T0PS0:** Timer0 预分频选择位
111=1:256 预分频值
110=1:128 预分频值
101=1:64 预分频值
100=1:32 预分频值
011=1:16 预分频值
010=1:8 预分频值
001=1:4 预分频值
000=1:2 预分频值

图注:

R= 可读位	W= 可写位	U= 未实现位, 读作 0
-n= 上电复位时的值	1= 置位	0= 清零
		x= 未知位

PIC18F2331/2431/4331/4431

图 11-1: 8 位模式下的 TIMER0 框图

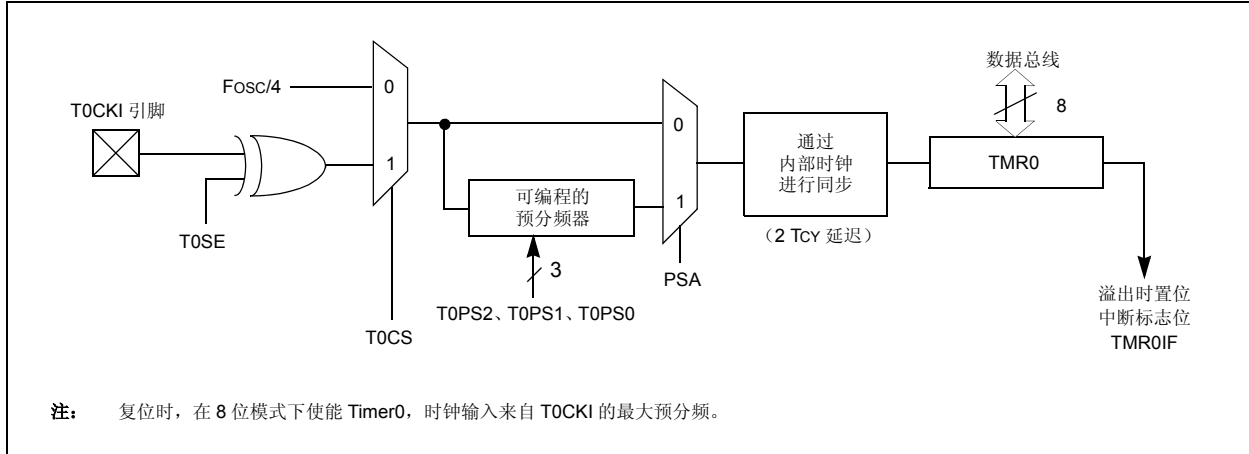
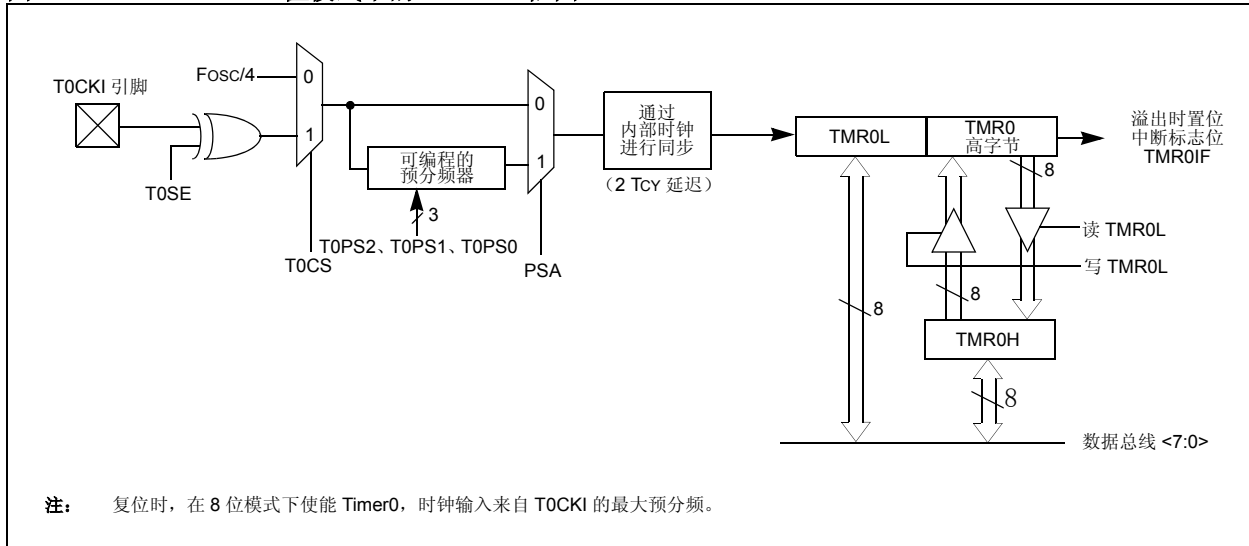


图 11-2: 16 位模式下的 TIMER0 框图



11.1 Timer0 工作原理

Timer0 既可用作定时器，也可用作计数器。

通过将 T0CS 位清零可以选择定时器模式。在定时器模式下，Timer0 模块在每个指令周期递增（不使用预分频器）。如果对 TMR0 寄存器执行写操作，则在随后两个指令周期中禁止递增。用户可通过将校正值写入 TMR0 寄存器避开这一问题。

通过将 T0CS 位置位可以选择计数器模式。在计数器模式下，Timer0 会在 RA4/T0CKI 引脚的每个上升沿或下降沿递增计数。递增边沿由 Timer0 时钟源边沿选择位（T0SE）决定。将 T0SE 位清零选择上升沿。

要将外部时钟输入信号用于 Timer0，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。同步之后，真正递增 Timer0 时还会有一段延时。

11.2 预分频器

8 位计数器可以作为 Timer0 模块的预分频器。预分频器是不可读写的。

PSA 和 T0PS2:T0PS0 位决定预分频器的分配以及预分频比。

清零 PSA 位可将预分频器分配给 Timer0 模块。当预分频器分配给 Timer0 时，可选择的预分频值有 1:2、1:4、……和 1:256。

当预分频器分配给 Timer0 模块时，所有对 TMR0 寄存器执行写操作的指令（如 CLRF TMR0、MOVWF TMR0 和 BSF TMR0 x 等）都将使预分频器计数清零。

注： 当将预分频器分配给 Timer0 时，对 TMR0 执行写操作会将预分频器计数清零，但不会改变预分频器的分配。

11.2.1 预分频器分配的切换

预分频器的分配完全由软件控制（也就是说，在程序执行期间可以随时更改分配）。

11.3 Timer0 中断

8 位模式下的 TMR0 寄存器从 FFh 到 00h 发生溢出，或 16 位模式下的 TMR0 从 FFFFh 到 0000h 发生溢出时，将产生 TMR0 中断。该溢出会置位 TMR0IF 位。清零 TMR0IE 位可以屏蔽该中断。在重新允许该中断之前，必须通过 Timer0 模块的中断服务程序用软件将 TMR0IF 位清零。因为 TMR0 需要时钟周期，所以即使已置位 T0CS，TMR0 中断仍无法将处理器从休眠模式中唤醒。

11.4 16 位模式下的定时器读写

TMR0H 并非 16 位模式下定时器 / 计数器的高字节，它实际上是被缓存的 Timer0 高字节（参见图 11-2）。Timer0 计数器 / 定时器的高字节不可直接读写。在读 TMR0L 时用 Timer0 高字节的内容更新 TMR0H。由于在连续读取高低字节之间会发生滚存，需要验证读取的高低字节是否有效，而现在可以直接读取 Timer0 的全部 16 位，就不需要验证读取到的高低字节的有效性了。

写 Timer0 的高字节也必须通过 TMR0H 缓冲寄存器进行。当写 TMR0L 时，会用 TMR0H 的内容更新 Timer0 高字节。这样就可以一次性完成 Timer0 的全部 16 位更新。

表 11-1: 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	POR 和 BOR 时的值	其他复位时的值	
TMR0L	Timer0 模块的低字节寄存器								xxxx xxxx	uuuu uuuu	
TMR0H	Timer0 模块的高字节寄存器								0000 0000	0000 0000	
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u	
T0CON	TMR0ON	T016BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	1111 1111	
TRISA	RA7 ⁽¹⁾	RA6 ⁽¹⁾	PORTA 数据方向寄存器							1111 1111	1111 1111

图注： x= 未知，u= 不变，-= 未实现，读作 0。阴影单元格表示 Timer0 未使用。

注 1： 根据配置字 1H 中选择的振荡器模式，RA6 和 RA7 可作为 I/O 引脚使能。

PIC18F2331/2431/4331/4431

注:

12.0 TIMER1 模块

Timer1 模块定时器 / 计数器具有以下特性:

- 16 位计数器 / 定时器
(两个 8 位寄存器: TMR1H 和 TMR1L)
- 可读写 (TMR1H 和 TMR1L 寄存器)
- 内部或外部时钟选择
- FFFFh 到 0000h 的溢出中断
- CCP 模块特殊事件触发器引起的复位
- 系统时钟的工作状态

图 12-1 是 Timer1 模块的简化框图。

寄存器 12-1 详细说明了 Timer1 控制寄存器。此寄存器控制 Timer1 模块的工作模式, 并且包含 Timer1 振荡器使能位 (T1OSCEN)。通过将控制位 TMR1ON (T1CON<0>) 置位或清零可以使能或禁止 Timer1。

Timer1 振荡器可以用作功耗管理模式的辅助时钟源。当 T1RUN 位被置位时, 由 Timer1 振荡器提供系统时钟。当使能故障保护时钟监视器时, 如果作为系统时钟源的 Timer1 振荡器发生故障, 可以查询 T1RUN 位, 了解时钟源是 Timer1 振荡器还是其他时钟源。

只需要增加极少的外部元件和代码开销, Timer1 就可以为应用提供实时时钟 (Real-Time Clock, RTC) 功能。

寄存器 12-1: T1CON: TIMER1 控制寄存器

	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
	bit 7							bit 0
bit 7	RD16: 16 位读 / 写模式使能位 1= 使能 16 位模式的 Timer1 寄存器读 / 写操作 0= 使能两个 8 位模式的 Timer1 寄存器读 / 写操作							
bit 6	T1RUN: Timer1 系统时钟状态位 1= 系统时钟由 Timer1 振荡器产生 0= 系统时钟由其他时钟源产生							
bit 5-4	T1CKPS1:T1CKPS0: Timer1 输入时钟预分频选择位 11= 1:8 预分频值 10= 1:4 预分频值 01= 1:2 预分频值 00= 1:1 预分频值							
bit 3	T1OSCEN: Timer1 振荡器使能位 1= 使能 Timer1 振荡器 0= 关闭 Timer1 振荡器 为了避免功率消耗, 关断振荡器的反相器和反馈电阻。							
bit 2	T1SYNC: Timer1 外部时钟输入同步选择位 <u>当 TMR1CS=1 (外部时钟) 时:</u> 1= 不同步外部时钟输入 0= 同步外部时钟输入 <u>当 TMR1CS=0 (内部时钟) 时:</u> 忽略此位。当 TMR1CS=0 时, Timer1 使用内部时钟。							
bit 1	TMR1CS: Timer1 时钟源选择位 1= RC0/T1OSO/T1CKI 引脚的外部时钟 (上升沿计数) 0= 内部时钟 (Fosc/4)							
bit 0	TMR1ON: Timer1 使能位 1= 使能 Timer1 0= 停止 Timer1							

图注:

R= 可读位	W= 可写位	U= 未实现位, 读作 0
-n= 上电复位时的值	1= 置位	0= 清零
		x= 未知位

PIC18F2331/2431/4331/4431

12.1 Timer1 工作原理

Timer1 可工作在以下模式：

- 作为定时器
- 作为同步计数器
- 作为异步计数器

工作模式由时钟选择位 TMR1CS (T1CON<1>) 决定。

当 TMR1CS=0 时, Timer1 在每个指令周期递增。当 TMR1CS=1 时, Timer1 在外部时钟输入或 Timer1 振荡器 (如使能了) 的每个上升沿递增。

当使能 Timer1 振荡器 (T1OSCEN 已置位) 时, RC1/T1OSI/CCP2/FLTA 和 RC0/T1OSO/T1CKI 引脚成为输入引脚。也就是说, 忽略 TRISC1:TRISC0 值, 这些引脚读作 0。

Timer1 还有一个内部“复位输入”。此复位信号可以由 CCP 模块产生 (参见第 15.4.4 节“特殊事件触发器”)。

图 12-1: TIMER1 框图

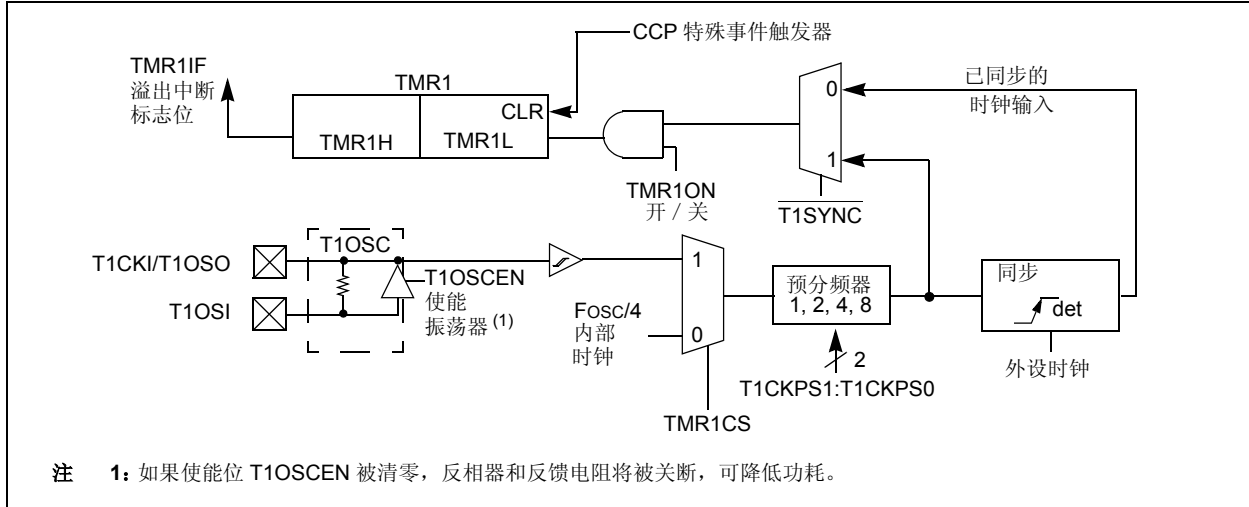
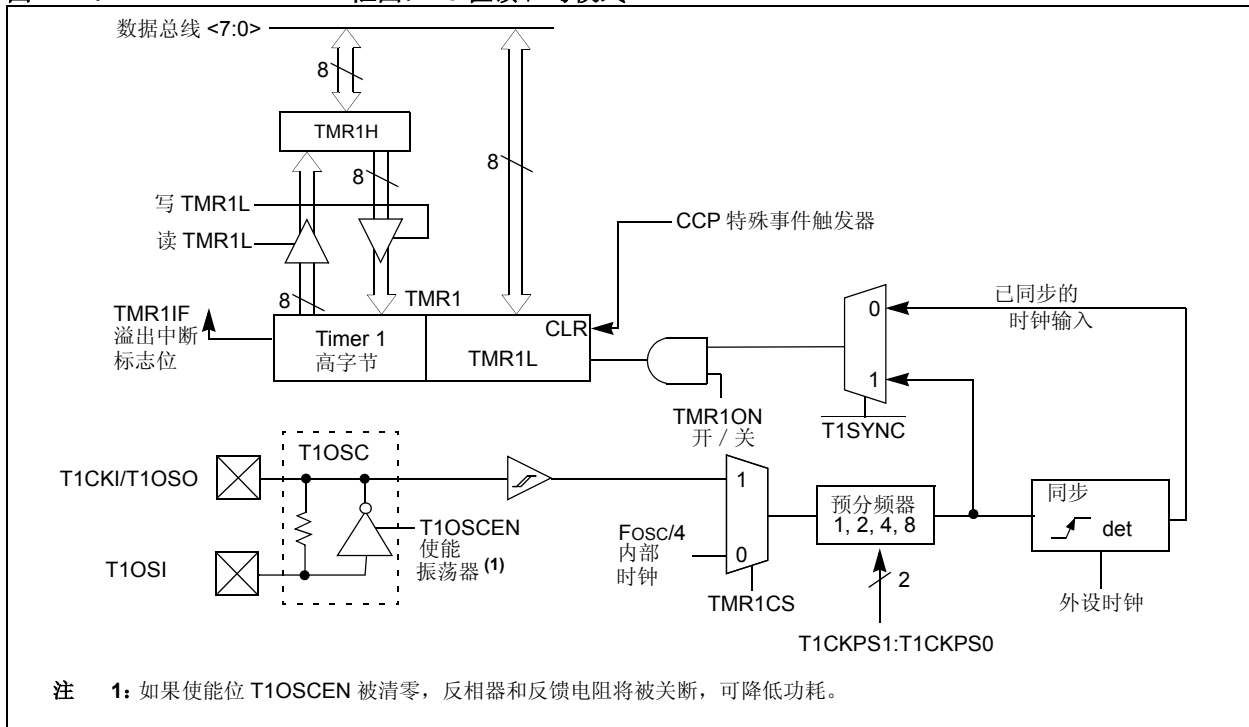


图 12-2: TIMER1 框图: 16 位读 / 写模式



12.2 Timer1 振荡器

晶体振荡电路内建在 T1OSI 引脚（输入）和 T1OSO 引脚（放大器输出）之间。通过置位控制位 T1OSCEN（T1CON<3>）可以使能该振荡器。这是个低功耗振荡器，可作为 32 kHz 晶振。在所有的功耗管理模式下都可以持续运行。典型 LP 振荡器的电路如图 12-3 所示。表 12-1 给出了 Timer1 振荡器的电容选择。

用户必须提供软件延时来确保能够正常起振 Timer1 振荡器。

图 12-3: TIMER1 LP 振荡器的外接元件

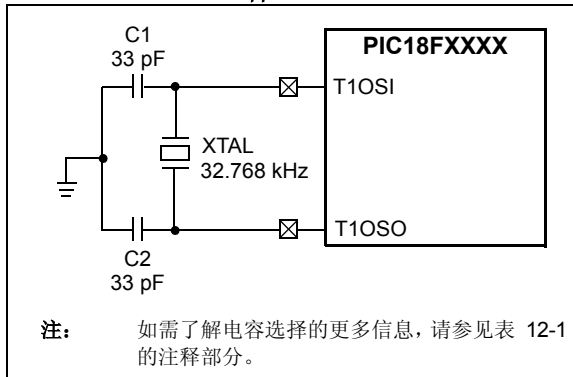


表 12-1: TIMER1 振荡器的电容选择

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

- 注**
- 1: Microchip 建议将该值作为验证振荡电路的起始点。
 - 2: 电容越大，振荡器越稳定，但起振时间越长。
 - 3: 因为每种谐振器 / 晶振都有其自身特性，用户应当向谐振器 / 晶振制造厂商咨询外部元件的适当值。
 - 4: 上述电容值仅供设计参考。

12.3 Timer1 振荡器设计注意事项

PIC18F2331/2431/4331/4431 器件的 Timer1 振荡器还包括了一项低功耗性能。当选择了此选项时，可以在单片机处于休眠模式时让振荡器自动降低功耗。在器件正常工作期间，振荡器会产生最大拉电流。因为高噪声环境可能会增加振荡器在休眠模式下的不稳定性，所以该选项尤其适合低噪声的应用场合，其中一个重要的设计注意事项就是节能。

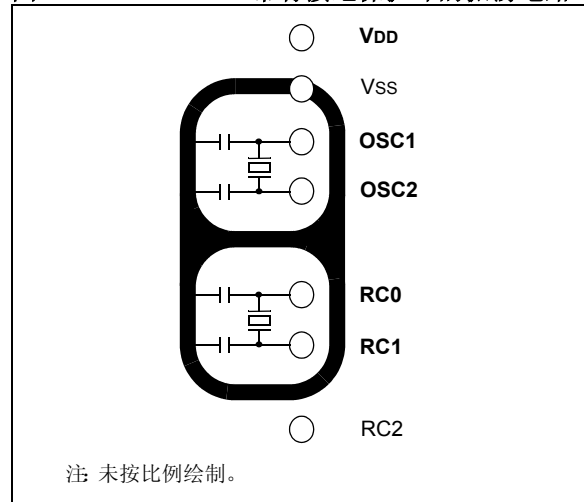
通过将 T1OSCMX 位（CONFIG3L<5>）清零，可以使能此低功耗选项。缺省情况下，此选项是禁用的，这会导致 Timer1 振荡器产生或多或少的恒定拉电流。

由于此振荡器的低功耗性能，因此可能对短时间内快速改变信号比较敏感。

如图 12-3 所示，振荡电路应该尽可能靠近单片机。除了 Vss 或 VDD 外，振荡电路边界范围内不应该有其他的电路通过。

当使用单面 PCB 时，如果在振荡器附近必须要有一个高速电路（如 CCP1 引脚处于输出比较模式或 PWM 模式，或主振荡器使用 OSC2 引脚），则在振荡器电路周围设置一个接地保护环可能会很有帮助（如图 12-4 所示），或者增加一个地平面。

图 12-4: 带有接地保护环的振荡电路



PIC18F2331/2431/4331/4431

12.4 Timer1 中断

TMR1 寄存器对 (TMR1H:TMR1L) 从 0000h 开始, 增加到 FFFFh, 然后回零到 0000h 重新开始。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>)。可以通过置位 / 清零 Timer1 中断使能位 TMR1IE (PIE1<0>) 来允许 / 禁止该中断。

12.5 使用 CCP 触发器输出复位 Timer1

如果 CCP 模块已配置为比较模式以产生特殊事件触发器 (CCP1M3:CCP1M0=1011), 此信号会复位 Timer1, (如已使能 A/D 模块) 还会启动 A/D 转换 (如需更多信息, 请参见第 15.4.4 节“特殊事件触发器”)。

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

要利用这一点, Timer1 必须设置为定时器或同步计数器模式。如果 Timer1 在异步计数器模式下运行, 该复位可能无效。

如果写 Timer1 操作和 CCP1 模块的特殊事件触发复位同时发生, 则写操作优先。

在这种工作模式下, CCPR1H:CCPR1L 寄存器对实际上变成了 Timer1 的周期寄存器。

12.6 Timer1 16 位读 / 写模式

Timer1 可以配置为 16 位读写模式 (见图 12-2)。当 RD16 控制位 (T1CON<7>) 置位时, TMR1H 的地址映射到 Timer1 高位字节缓冲寄存器。对 TMR1L 执行读操作会将 Timer1 的高位字节内容装入 Timer1 高字节缓冲器。这样使得用户可以准确地读取 Timer1 的全部 16 位, 而不会因两次读取之间发生滚存而需要判断读高字节再读低字节是否有效。

写 Timer1 的高字节也必须通过 TMR1H 缓冲寄存器进行。当写 TMR1L 时, 会用 TMR1H 的内容更新 Timer1 高字节。这允许用户可以一次性写入 Timer1 的高位字节和低位字节的全部 16 位。

Timer1 的高字节在此模式下不可直接读写。所有的读写必须通过 Timer1 高字节缓冲寄存器进行。写入 TMR1H 不会清零 Timer1 预分频器。预分频器只有在写 TMR1L 时才会清零。

12.7 使用 Timer1 作为实时时钟

为 Timer1 外接一个 LP 振荡器 (如第 12.2 节“Timer1 振荡器”中所述), 可以允许用户在他们的应用中包括 RTC 功能。这一功能的实现是由一个便宜的晶体提供精确时基, 并用几行应用代码计算时间。在休眠工作模式下采用电池或超级电容作为电源时, 这样可以完全避免另外增加分立的 RTC 器件和备用电池。

应用代码程序 RTCisr (如例 12-1 所示) 演示了一种简单的方法, 它通过使用中断服务程序实现每秒递增计数。TMR1 寄存器对递增至溢出状态将触发中断, 并调用程序, 该程序会使秒计数器递增。当秒计数器溢出时, 分计数器和小时计数器会相应递增。

因此此寄存器对为 16 位宽, 因此直接按 32.768 kHz 时钟计数溢出需要 2 秒。要使溢出按所需的 1 秒间隔进行, 必须预先将其加载; 最简单的方法是使用 BSF 指令将 TMR1H 的最高有效位置位。请注意 TMR1L 寄存器从不会预先加载或变更, 因此这样做可能会导致很多个周期后出现累积错误。

要使此方法变精确, Timer1 必须工作在异步模式, 且必须允许 Timer1 溢出中断 (PIE1<0>=1), 如程序 RTCinit 所示。同时 Timer1 振荡器也必须使能并始终保持运行。

PIC18F2331/2431/4331/4431

例 12-1: 使用 TIMER1 中断服务程序实现实时时钟

```

RTCinit
    MOVLW    0x80          ; Preload TMR1 register pair
    MOVWF   TMR1H        ; for 1 second overflow
    CLRF    TMR1L
    MOVLW   b'00001111'  ; Configure for external clock,
    MOVWF   T1OSC        ; Asynchronous operation, external oscillator
    CLRF    secs         ; Initialize timekeeping registers
    CLRF    mins         ;
    MOVLW   .12
    MOVWF   hours
    BSF     PIE1, TMR1IE ; Enable Timer1 interrupt
    RETURN

RTCisr
    BSF     TMR1H, 7     ; Preload for 1 sec overflow
    BCF     PIR1, TMR1IF ; Clear interrupt flag
    INCF    secs, F      ; Increment seconds
    MOVLW   .59         ; 60 seconds elapsed?
    CPFSGT  secs
    RETURN            ; No, done
    CLRF    secs        ; Clear seconds
    INCF    mins, F     ; Increment minutes
    MOVLW   .59         ; 60 minutes elapsed?
    CPFSGT  mins
    RETURN            ; No, done
    CLRF    mins       ; clear minutes
    INCF    hours, F   ; Increment hours
    MOVLW   .23        ; 24 hours elapsed?
    CPFSGT  hours
    RETURN            ; No, done
    MOVLW   .01        ; Reset hours to 1
    MOVWF   hours
    RETURN            ; Done
    
```

表 12-2: 与作为定时器 / 计数器的 TIMER1 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	-000 000x	0000 000u
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
IPR1	—	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	-111 1111
TMR1L	16 位 TMR1 寄存器的最低有效字节保持寄存器								xxxx xxxx	uuuu uuuu
TMR1H	16 位 TMR1 寄存器的最高有效字节保持寄存器								xxxx xxxx	uuuu uuuu
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	u0uu uuuu

图注: x= 未知, u= 不变, -= 未实现, 读作 0。阴影单元格表示 Timer1 未使用。

PIC18F2331/2431/4331/4431

注:

13.0 TIMER2 模块

Timer2 模块具有以下特性:

- 8 位定时器 (TMR2 寄存器)
- 8 位周期寄存器 (PR2)
- 可读写 (2 个寄存器)
- 可软件编程的预分频器 (1:1、1:4 和 1:16)
- 可软件编程的后分频器 (1:1 至 1:16)
- TMR2 与 PR2 匹配时中断
- SSP 模块可选使用 TMR2 输出产生时钟移位

如寄存器 13-1 所示, Timer2 有一个控制寄存器。可以通过将控制位 TMR2ON (T2CON<2>) 清零关断 TMR2, 以便尽量减少功耗。图 13-1 是 Timer2 模块的简化框图。寄存器 13-1 显示了 Timer2 的控制寄存器。该寄存器控制 Timer2 预分频器和后分频器的选择。

13.1 Timer2 工作原理

Timer2 可以与 CCP 模块配合使用, 在 PWM 模式下作为 PWM 时基。TMR2 寄存器是可读写的, 任何方式的器件复位都会使之清零。可以通过控制位 T2CKPS1:T2CKPS0 (T2CON<1:0>) 选择输入时钟 (Fosc/4) 的预分频值 1:1、1:4 或 1:16。TMR2 的匹配输出通过 4 位后分频器 (分频比为 1:1 到 1:16, 包含 1:1 和 1:16) 产生 TMR2 中断 (锁存在 TMR2IF 标志位 (PIR1<1>))。

如果发生下列任何一种情况, 预分频器和后分频器计数器都会被清零:

- 写 TMR2 寄存器
- 写 T2CON 寄存器
- 任何器件复位 (上电复位、MCLR 复位、看门狗定时器复位或欠压复位)

写 T2CON 时 TMR2 不会清零。

寄存器 13-1: T2CON: TIMER2 控制寄存器

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

bit 7 未实现: 读作 0

bit 6-3 TOUTPS3:TOUTPS0: Timer2 输出后分频值选择位

0000=1:1 后分频值
 0001=1:2 后分频值
 .
 .
 .
 1111=1:16 后分频值

bit 2 TMR2ON: Timer2 使能位

1= 使能 Timer2
 0= 禁止 Timer2

bit 1-0 T2CKPS1:T2CKPS0: Timer2 时钟预分频值选择位

00= 预分频值为 1
 01= 预分频值为 4
 1x= 预分频值为 16

图注:

R= 可读位	W= 可写位	U= 未实现位, 读作 0
-n= 上电复位时的值	1= 置位	0= 清零 x= 未知位

PIC18F2331/2431/4331/4431

13.2 Timer2 中断

Timer2 模块有一个 8 位周期寄存器 PR2。Timer2 从 00h 开始递增，直到与 PR2 匹配，然后在下一递增计数周期开始时复位为 00h。PR2 为可读写寄存器。PR2 寄存器会在复位时初始化为 FFh。

13.3 TMR2 的输出

将 TMR2 的输出信号（在后分频器之前）会馈送至同步串行端口模块，以便生成移位时钟。

图 13-1: TIMER2 框图

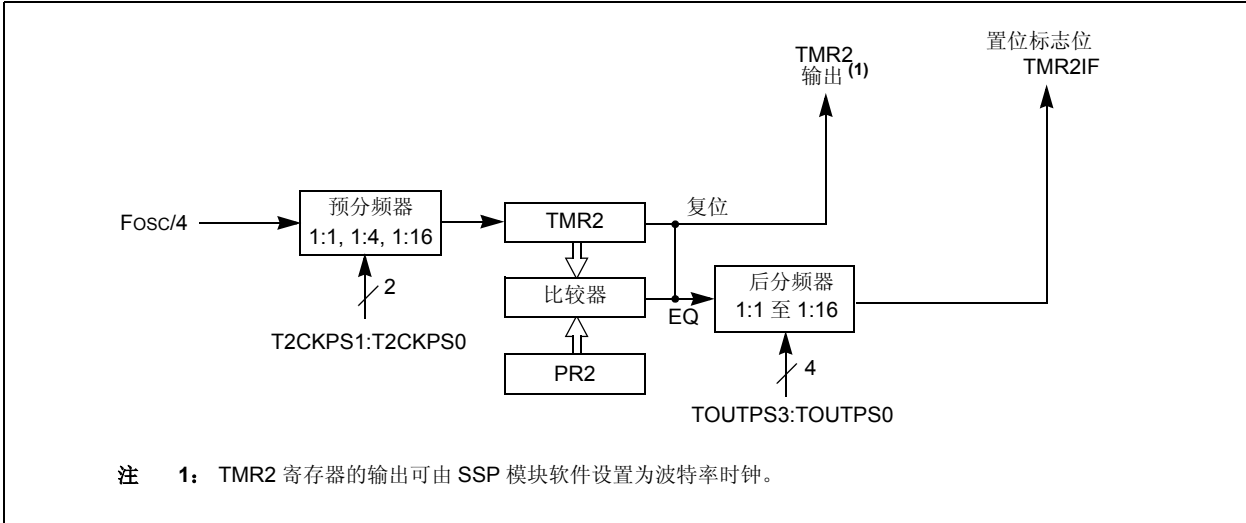


表 13-1: 与作为定时器 / 计数器的 TIMER2 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
IPR1	—	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	-111 1111	-111 1111
TMR2	Timer2 模块寄存器								0000 0000	0000 0000
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
PR2	Timer2 周期寄存器								1111 1111	1111 1111

图注: x= 未知, u= 不变, -= 未实现, 读作 0。阴影单元格表示 Timer2 未使用。

PIC18F2331/2431/4331/4431

14.0 TIMER5 模块

Timer5 模块实现以下特性:

- 16 位定时器 / 计数器
- 同步和异步计数器模式
- 连续计数和单脉冲运行模式
- 四个可编程的预分频值 (1:1 到 1:8)
- 周期匹配时产生中断
- 特殊事件触发复位功能
- 双缓冲寄存器
- 休眠模式下运行
- 将 CPU 从休眠模式唤醒
- 带唤醒功能的可选择硬件复位输入

Timer5 是一个通用的定时器 / 计数器, 它包含了更多的功能, 可以与运动反馈模块 (Motion Feedback module) 一起使用 (见第 16.0 节 “运动反馈模块”)。它也可以用作通用定时器或特殊事件触发延迟定时器。当作为通用定时器时, 它可以被配置为使用预编程周期延迟来产生一个延迟的特殊事件触发 (例如, ADC 特殊事件触发)。

通过 Timer5 控制寄存器 (T5CON) 来控制 Timer5, 如寄存器 14-1 所示。通过将控制位 TMR5ON (T5CON<0>) 置位或清零可以使能或禁止 Timer5。

Timer5 的框图如图 14-1 所示。

寄存器 14-1:

T5CON: TIMER5 控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
T5SEN	RESEN	T5MOD	T5PS1	T5PS0	T5SYNC	TMR5CS	TMR5ON	
bit 7								bit 0

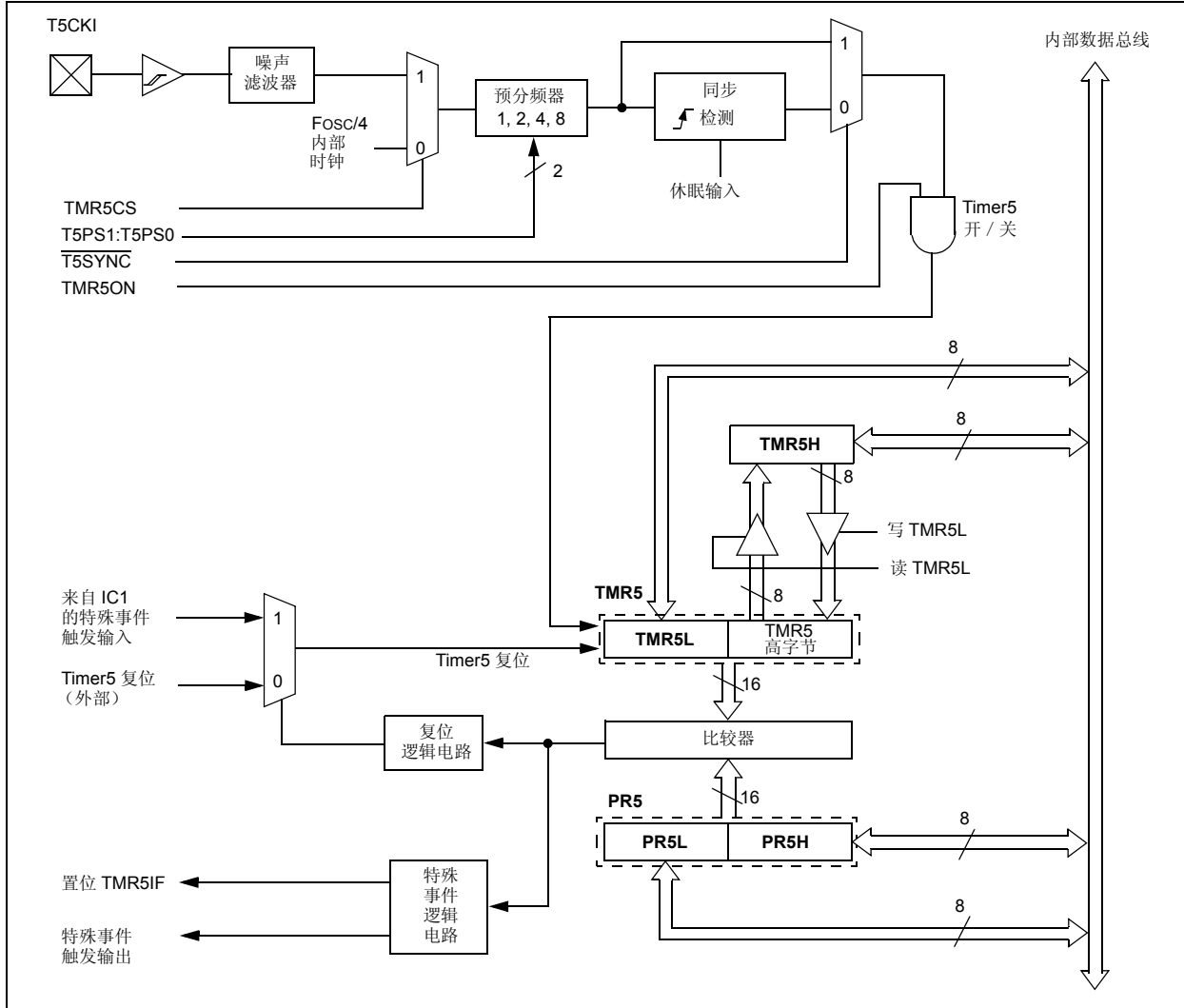
- bit 7 **T5SEN:** Timer5 休眠使能位 ⁽¹⁾
 1 = 在休眠状态使能 Timer5
 0 = 在休眠状态禁止 Timer5
- bit 6 **RESEN:** 特殊事件复位使能位
 1 = 特殊事件复位禁止
 0 = 特殊事件复位使能
- bit 5 **T5MOD:** Timer5 模式位
 1 = 使能单脉冲模式
 0 = 使能连续计数模式
- bit 4:3 **T5PS1:T5PS0:** Timer5 输入时钟预分频选择位
 11 = 1:8
 10 = 1:4
 01 = 1:2
 00 = 1:1
- bit 2 **T5SYNC:** Timer5 外部时钟输入同步选择位
当 TMR5CS=1:
 1 = 不同步外部时钟输入
 0 = 同步外部时钟输入
当 TMR5CS=0:
 此位被忽略。当 TMR5CS=0 时, Timer5 使用内部时钟
- bit 1 **TMR5CS:** Timer5 时钟源选择位
 1 = 来自引脚 T5CKI 的外部时钟
 0 = 内部时钟 (Tcy)
- bit 0 **TMR5ON:** Timer5 使能位
 1 = 使能 Timer5
 0 = 禁止 Timer5

注 1: 由于 Timer5 在休眠模式运行, T5SYNC 必须被置位。

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

PIC18F2331/2431/4331/4431

图 14-1: TIMER5 框图 (16 位读 / 写模式)



14.1 Timer5 工作原理

Timer5 包含了两个 8 位寄存器，组成一个 16 位定时器。TMR5L 寄存器是真正的定时器低字节，可以直接读写。高字节包含在未映射的寄存器中，通过作为缓冲器的 TMR5H 进行读写。每个寄存器都从 00h 增加到 FFh。

另一个寄存器对 PR5H 和 PR5L 作为周期寄存器使用，用来设置 TMR5 寄存器对的最大计数。当 TMR5 达到 PR5 的值时，定时器计满回零为 0000h 并置位 TMR5IF 中断标志位。Timer5 模块的简化框图如图 2-1 所示。

注： TIMER5 可以用作通用定时器，也可用作运动反馈模块（输入捕捉或正交编码器接口）的时基源。

Timer5 支持三种配置：

- 16 位同步定时器
- 16 位同步计数器
- 16 位异步计数器

在同步定时器配置中，定时器由内部器件时钟提供时钟信号。如果选择 Timer5 作为预分频器，可以选择将时钟输入信号进行 2、4、8 分频或不进行分频（1:1）。TMR5 寄存器对在 Q1 时进行递增计数。清零 TMR5CS 则选择内部器件时钟作为定时器的采样时钟。

在同步计数器配置中，定时器由外部时钟（T5CKI）和可选的预分频器共同提供时钟信号。置位 TMR5CS 位（TMR5CS = 1）选择外部 T5CKI；清零 TMR5CS 则选择内部时钟源。通过将 T5SYNC 位清零，可使外部时钟与内部时钟保持同步。在内部时钟的每个 Q2 和 Q4 时对 T5CKI 的输入信号进行采样。如果连续三次采样结果都是上升沿变化，则 Timer5 在下一个 Q1 时进行递增计数。T5CKI 最短的脉宽高电平时间和低电平时间必须大于 $T_{cy}/2$ 。

在异步计数器配置中，Timer5 由外部时钟（T5CKI）和可选的预分频器共同提供时钟信号。在此模式中，T5CKI 与内部时钟不同步。通过置位 TMR5CS，外部输入时钟（T5CKI）可以用作计数器采样时钟。当 T5SYNC 被置位时，外部时钟与内部器件时钟不同步。

当该模块被禁止时，定时器计数不会自动复位。用户可以通过对计数寄存器执行写操作来启动计数器。

注： 当 Timer5 定时器被使能时，该模块不会阻止写 PR5 寄存器（PR5H:PR5L）。定时器使能时写 PR5 可能导致出现意外的周期匹配事件。

14.1.1 连续计数和单脉冲运行

Timer5 有两种工作模式：连续计数和单脉冲。

通过清零 T5MOD 控制位（= 0）选择连续计数模式。在此模式中，Timer5 时基根据预分频器设置开始递增，直到发生 TMR5/PR5 匹配，或 TMR5 计满回零（FFFFh 回到 0000h）。置位 TMR5IF 中断标志位，并在随后的输入时钟沿时复位 TMR5 寄存器，且只要 TMR5ON 位保持在置位状态，定时器就继续计数。

置位 T5MOD（= 1）选择单脉冲模式。在此模式中 Timer5 时基根据预分频器设置开始递增，直到发生 TMR5/PR5 匹配。这将引起置位 TMR5IF 中断标志位并在随后的输入时钟沿时复位 TMR5 寄存器对，而且 TMR5ON 位将被硬件清零以停止定时器。

Timer5 时基只能在两种情况下的单脉冲模式开始递增计数：

1. Timer5 使能（TMR5ON 置位），或者
2. Timer5 禁止，且在 Timer5 复位输入上出现特殊事件复位触发。（更多信息参见第 14.7 节“Timer5 特殊事件复位输入”）。

14.2 16 位读 / 写和写模式

前面已经提到，Timer5 寄存器对的真正高字节映射到 TMR5H，TMR5H 作为缓冲器。读 TMR5L 将使寄存器对的高字节内容加载到 TMR5H 寄存器。这样使得用户可以准确地读取寄存器对的全部 16 位，而不会因两次读取之间发生滚存而需要判断读高字节再读低字节是否有效。

由于 Timer5 寄存器对的实际高字节不可以直接读写，它必须通过 Timer5 高字节缓冲寄存器（TMR5H）来读写。当发生写 TMR5L 的操作时，会用 TMR5H 的内容更新 T5 高字节。这允许用户一次写入 Timer5 高位字节和低位字节的全部 16 位。写入 TMR5H 不会清零 Timer5 预分频器。预分频器只有在写 TMR5L 时清零。

14.2.1 16 位读—修改—写操作

读—修改—写指令（如 BSF 和 BCF）先读寄存器的内容，进行适当修改，然后将结果写回到寄存器。对 TMR5H 执行读—修改—写指令的写部分直到发生写 TMR5L 时才会更新 TMR5 的高字节内容。之后才会将 TMR5H 的内容放入 TMR5 高字节。

14.3 Timer5 预分频器

可以使用 Timer5 的可编程预分频器对 Timer5 时钟输入（Tcy 或外部时钟）进行分频。预分频器控制位 T5PS1:T5PS0（T5CON<4:3>）选择预分频因子 2、4、8 或不预分频。

Timer5 预分频器可以用下面任一种方法清零：

- 写 Timer5 寄存器
- 禁止 Timer5（TMR5ON = 0）
- 器件复位，比如主清零、POR 或 BOR

注： 写 T5CON 寄存器不会清零 Timer5。

14.4 噪声滤波器

Timer5 模块包含一个可选的输入噪声滤波器，专用于降低噪声工作环境下的杂散信号。该滤波器确保只有在连续三个采样时钟周期都得到一个稳定值的时候才会改变输入。

噪声滤波器是与运动反馈模块相关的输入滤波网络的一部分（见第 16.0 节“运动反馈模块”）。所有这些滤波器都由数据滤波器控制（DFLTCON）寄存器（寄存器 16-3）控制。Timer5 滤波器可以通过置位或清零 FLT4EN 位（DFLTCON<7>）被单独使能或禁止。它在所有 POR 和 BOR 复位时都会被禁止。

如需了解更多信息，请参阅运动反馈模块中的第 16.3 节“噪声滤波器”。

PIC18F2331/2431/4331/4431

14.5 Timer5 中断

Timer5 能够在周期匹配时产生中断。当 PR5 寄存器装入新的周期值 (00FFh) 时, Timer5 时基进行递增计数直到它的值与 PR5 的值相等。当发生匹配时, 在 Q4 的上升沿产生 Timer5 中断; 并在下一个 Tcy 置位 TMR5IF。

中断等待时间 (也就是从 Timer5 计满回零到 TMR5IF 置位所经过的时间) 不会超过 1 Tcy。当 Timer5 时钟输入被预分频且发生 TMR5/PR5 匹配时, 在 TMR5 复位后的第一个 Q4 上升沿会产生中断。

14.6 Timer5 特殊事件触发输出

在 TMR5/PR5 匹配时产生 Timer5 特殊事件触发。此特殊事件触发在 Q3 的下降沿产生。

Timer5 必须配置为同步模式 (计数器或定时器) 才能利用特殊事件触发功能。如果 Timer5 在异步计数器模式下运行, 特殊事件触发器可能不会工作, 也因此无法使用。

14.7 Timer5 特殊事件复位输入

除了特殊事件输出, Timer5 还有特殊事件复位输入, 该功能可以与运动反馈模块的输入捕捉通道 1 (IC1) 配合使用。要使用特殊事件复位, 捕捉 1 控制寄存器 CAP1CON 必须配置为某一种特殊事件触发模式 (CAP1M3:CAP1M0=1110 或 1111)。置位 RESEN 控制位 (T5CON<6>) 可以禁止特殊事件复位触发。

特殊事件复位将复位 Timer5 时基。在连续计数或单脉冲模式下都会发生此复位。

14.7.1 IC1 边沿唤醒

Timer5 特殊事件复位输入可用作 Timer5 唤醒和启动脉冲。Timer5 必须在单脉冲模式且被禁止 (TMR5ON = 0)。CAP1 输入引脚上的活动边沿将置位 TMR5ON; 然后定时器在下一个时钟根据预分频器和 Timer5 时钟设置进行递增计数。接下来的硬件超时 (如 TMR5/PR5 匹配) 将清零 TMR5ON 位并停止该定时器。

14.7.2 延迟操作事件触发

CAP1 上的活动边沿还可用于启动某个后续操作, 并由 Timer5 时基对该后续操作进行延迟。这种情况下, Timer5 在被触发后会像之前一样递增计数。当发生硬件超时, 将产生特殊事件触发输出用于触发另一个操作, 如 A/D 转换。这就允许从 CAP1 的捕获边沿引用某个给定的硬件操作, 且该硬件操作由定时器进行延迟。

我们将在第 16.1 节 “输入捕捉” 中更深入地讨论延迟操作事件触发的事件时序。

14.7.3 当 TIMER5 递增时的特殊事件复位

如果 Timer5 的总线写操作与特殊事件复位触发同时发生, 该总线写操作的优先级始终高于特殊事件复位触发。

14.8 休眠模式下运行

当 Timer5 配置为异步操作模式时, 它将在每个定时器时钟信号下 (或时钟的预分频倍乘) 继续递增。执行 SLEEP 指令可能会停止定时器也可能让定时器继续工作, 这取决于是否置位 Timer5 的休眠使能位 (T5SE)。如果 T5SE 被置位 (= 1) 且选定外部时钟 (TMR5CS = 1), 执行 SLEEP 指令时定时器将继续工作。如果 T5SE 被清零, 则无论 GTPCS 位的状态如何, 当执行 SLEEP 指令时定时器都将停止。

总之, 只有以下所有位被置位时, Timer5 在执行 SLEEP 指令时才会保持继续递增:

- TMR5ON
- T5SE
- TMR5CS
- T5SYNC

14.8.1 休眠模式下的中断检测

如果按上述进行配置, 休眠模式下 Timer5 将在 T5CKI 的每个上升沿继续递增。当发生 TMR5/PR5 匹配时, 会产生中断, 可唤醒该器件。

PIC18F2331/2431/4331/4431

表 14-1: 与 TIMER5 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
IPR3	—	—	—	PTIP	IC3DRIP	IC2QEIP	IC1IP	TMR5IP	---1 1111	---1 1111
PIE3	—	—	—	PTIE	IC3DRIE	IC2QEIE	IC1IE	TMR5IE	---0 0000	---0 0000
PIR3	—	—	—	PTIF	IC3DRIF	IC2QEIF	IC1IF	TMR5IF	---0 0000	---0 0000
TMR5H	Timer5 寄存器高字节								xxxx xxxx	uuuu uuuu
TMR5L	Timer5 寄存器低字节								xxxx xxxx	uuuu uuuu
PR5H	Timer5 周期寄存器高字节								1111 1111	1111 1111
PR5L	Timer5 周期寄存器低字节								1111 1111	1111 1111
T5CON	T5SEN	RESEN	T5MOD	T5PS1	T5PS0	T5SYNC	TMR5CS	TMR5ON	0000 0000	0000 0000
CAP1CON	—	CAP1REN	—	—	CAP1M3	CAP1M2	CAP1M1	CAP1M0	-1-- 0000	-1-0 0000
DFLTCON	—	FLT4EN	FLT3EN	FLT2EN	FLT1EN	FLTCK2	FLTCK1	FLTCK0	-000 0000	-000 0000

图注: x= 未知, u= 不变, -= 未实现。

PIC18F2331/2431/4331/4431

注:

15.0 捕捉 / 比较 / PWM (CCP) 模块

CCP (捕捉 / 比较 / PWM) 模块包含一个 16 位的寄存器, 它可作为 16 位捕捉寄存器、比较寄存器或 PWM 主控 / 从动占空比寄存器。表 15-1 所示为 CCP 模块的每种工作模式所需要的定时器资源。

除了特殊事件触发器之外, CCP1 的操作和 CCP2 相同。因此, 除了特别注明之处为, 以下描述的 CCP 模块操作皆是针对 CCP1 而言的。

寄存器 15-1: CCPxCON: CCP 模块控制寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	
bit 7								bit 0

bit 7-6 **未实现位:** 读作 0

bit 5-4 **DCxB1:DCxB0:** PWM 占空比 bit1 和 bit0

捕捉模式:

未使用

比较模式:

未使用

PWM 模式:

这些位是 10 位 PWM 占空比的两个 LSb (bit 1 和 bit 0)。占空比的高八位 (DCx9:DCx2) 在 CCPxL 中。

bit 3-0 **CCPxM3:CCPxM0:** CCPx 模式选择位

0000 = 禁止捕捉 / 比较 / PWM (复位 CCPx 模块)

0001 = 保留

0010 = 比较模式, 匹配时翻转输出 (置位 CCPxIF 位)

0011 = 保留

0100 = 捕捉模式, 每个下降沿

0101 = 捕捉模式, 每个上升沿

0110 = 捕捉模式, 每 4 个上升沿

1000 = 捕捉模式, 每 16 个上升沿

1000 = 比较模式, 初始化 CCP 引脚为低电平, 比较匹配时强制 CCP 引脚为高电平 (置位 CCPxIF 位)

1001 = 比较模式, 初始化 CCP 引脚为高电平, 比较匹配时强制 CCP 引脚为低电平 (置位 CCPxIF 位)

1010 = 比较模式, 比较匹配时产生软件中断 (置位 CCPxIF 位, CCP 引脚不受影响)

1010 = 比较模式, 产生特殊触发事件 (置位 CCP2IF 位)

11xx = PWM 模式

图注:

R= 可读位

W= 可写位

U= 未实现位, 读作 0

-n = 上电复位时的值

1= 置位

0= 清零

x= 未知

PIC18F2331/2431/4331/4431

15.1 CCP1 模块

捕捉 / 比较 / PWM 寄存器 1 (CCPR1) 由两个 8 位寄存器组成: CCPR1L (低字节) 和 CCPR1H (高字节)。CCP1CON 寄存器控制 CCP1 的操作。所有位均可读写。

15.2 CCP2 模块

捕捉 / 比较 / PWM 寄存器 2 (CCPR2) 由两个 8 位寄存器组成: CCPR2L (低字节) 和 CCPR2H (高字节)。CCP2CON 寄存器控制 CCP2 的操作。所有位均可读写。

表 15-1: CCP 模块一定时器资源

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

15.3 捕捉模式

在捕捉模式下，当引脚 RC2/CCP1 上发生事件时，CCPR1H:CCPR1L 将捕捉 TMR1 寄存器的 16 位值。事件定义如下：

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

事件由控制位 CCP1M3:CCP1M0 (CCP1CON<3:0>) 选择。当一个捕捉发生时，中断请求标志位 CCP1IF (PIR1<2>) 置位，它必须用软件清零。如果在寄存器 CCPR1 中的值被读出之前，发生了另一个捕捉，那么之前捕捉的值将会被新值覆盖。

15.3.1 CCP 引脚配置

在捕捉模式下，应通过将 TRISC<2> 位置位使 RC2/CCP1 引脚配置为输入。

注： 如果 RC2/CCP1 被配置为输出，对端口的写操作可以产生捕捉条件。

15.3.2 TIMER1 模式选择

在使用捕捉功能时，Timer1 必须为定时器模式或同步计数器模式。在异步计数器模式下，捕捉可能无法进行。

15.3.3 软件中断

捕捉模式的改变可能会产生错误的捕捉中断。为避免产生错误中断，用户应该保持 CCP1IE 位 (PIE1<2>) 为零，且应在操作模式改变后将标志位 CCP1IF 清零。

15.3.4 CCP 预分频器

通过指定 CCP1M3:CCP1M0 位，可设置四种预分频比。只要 CCP 模块关闭或未处于捕捉模式，预分频器的计数器即清零。这意味着任何复位都会将预分频器计数器清零。

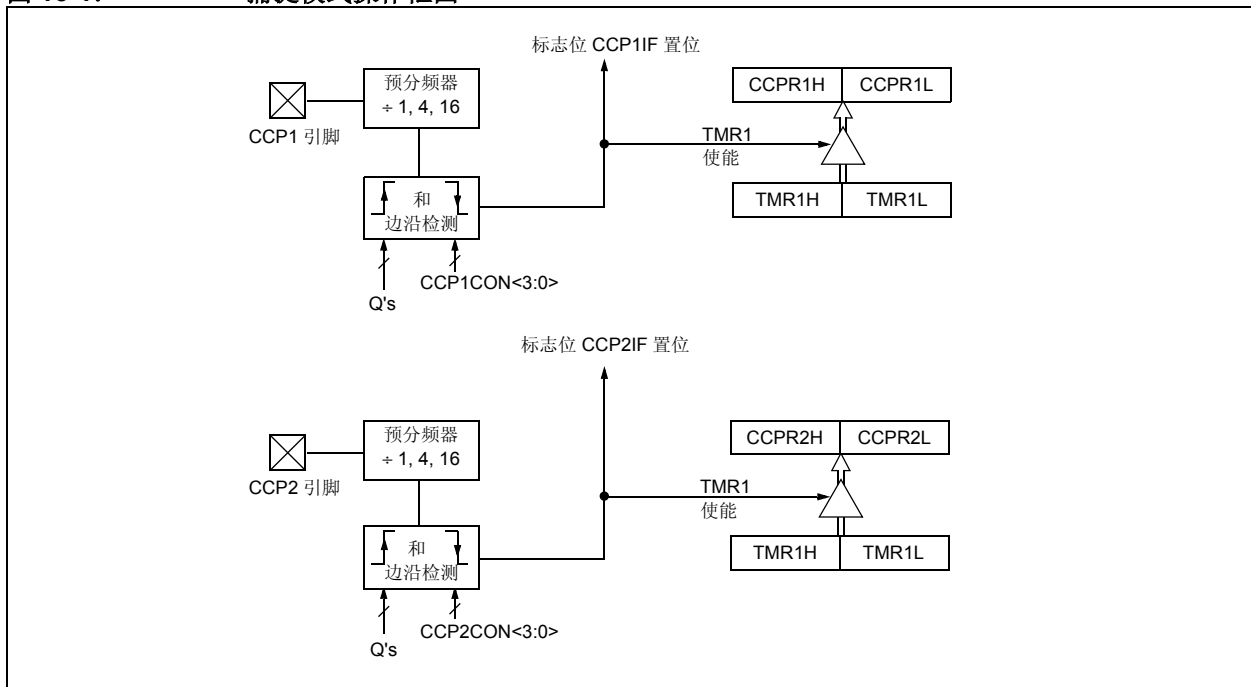
从一个捕捉预分频值切换到另一个捕捉预分频值可能产生中断。而且，预分频器计数器将不会清零，因此，第一个捕捉可能来自一个非零的预分频值。例 15-1 给出了在捕捉预分频值之间切换的建议方法。这个例子也使预分频器计数器清零且不会产生“错误”中断。

例 15-1: 在捕捉预分频值之间进行切换

```

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

图 15-1: 捕捉模式操作框图



PIC18F2331/2431/4331/4431

15.4 比较模式

在比较模式下，16 位 CCP1（CCP2）寄存器的值不断与 TMR1 寄存器对的值进行比较。当两者相符时，RC2/CCP1（RC1/CCP2）引脚将：

- 被拉高
- 被拉低
- 翻转输出（高电平变为低电平或低电平变为高电平）
- 保持不变（仅中断）

引脚的行为取决于控制位 CCP1M3:CCP1M0（CCP2M3:CCP2M0）的值。同时，中断标志位 CCP1IF（CCP2IF）被置位。

15.4.1 CCP 引脚配置

用户必须通过将相应的 TRISC 位清零，将 CCPx 引脚配置为输出引脚。

注： 清零 CCP1CON 寄存器将迫使 RC2/CCP1 比较输出锁存器为缺省低电平。这不是 PORTC I/O 数据锁存器。

15.4.2 TIMER1 模式选择

当 CCP 模块使用比较功能时，Timer1 必须为定时器模式或同步计数器模式。在异步计数器模式下，可能无法进行比较操作。

15.4.3 软件中断模式

当选择产生软件中断时，CCP1 引脚上的电平不受影响。只会产生一个 CCP 中断（若中断使能）。

15.4.4 特殊事件触发器

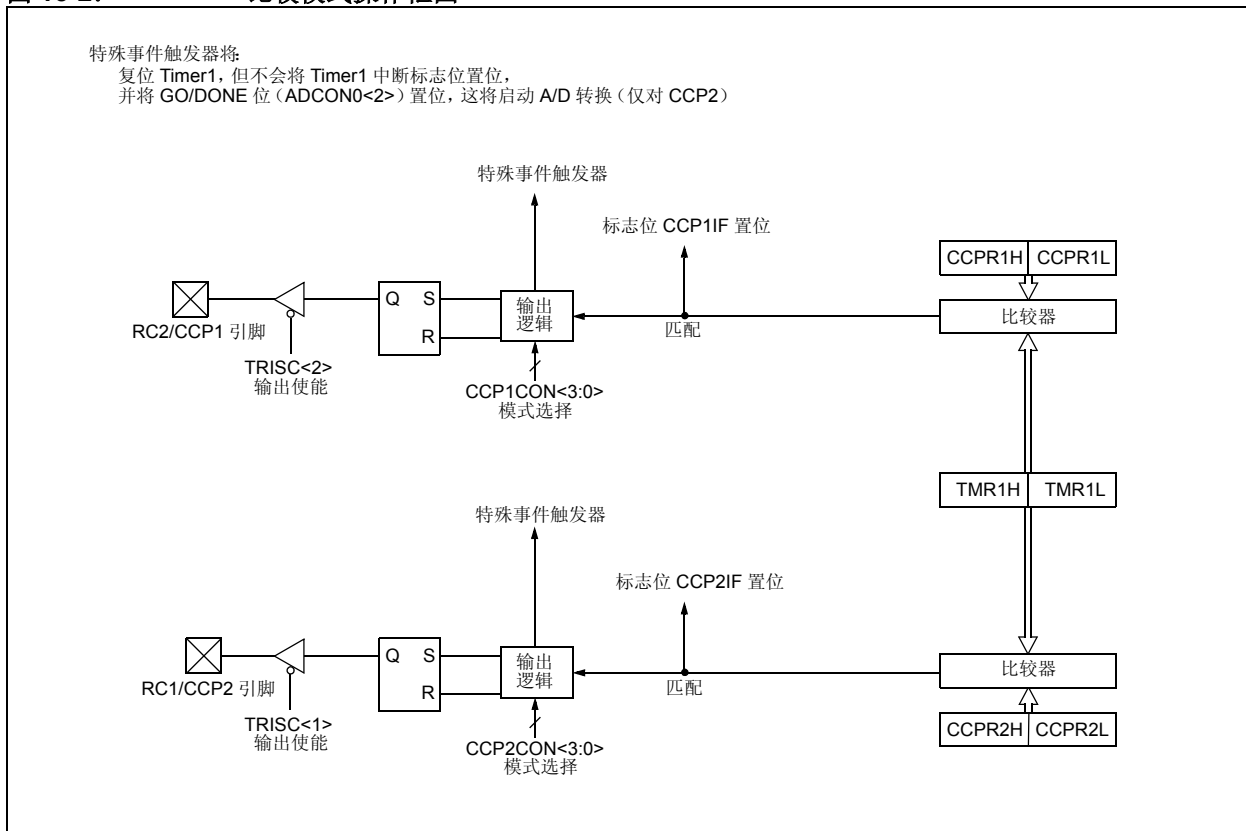
在这一模式下，将产生一个内部硬件触发信号，该信号可用于启动一个操作。

CCP1 的特殊事件触发器输出使 TMR1 寄存器对复位。这将使 CCP1 寄存器有效地成为 Timer1 的 16 位可编程周期寄存器。

CCP2 的特殊事件触发器输出使 TMR1 寄存器对复位。另外，如果使能了 A/D 模块，则 CCP2 特殊事件触发器将启动 A/D 转换。

注： CCP2 模块的特殊事件触发器不会将 Timer1 的中断标志位置位。

图 15-2: 比较模式操作框图



PIC18F2331/2431/4331/4431

表 15-2: 与捕捉、比较和 TIMER1 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
IPR1	—	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	-111 1111	-111 1111
TRISC	PORTC 数据方向寄存器								1111 1111	1111 1111
TMR1L	保持寄存器的值为 16 位 TMR1 寄存器的低有效字节								xxxx xxxx	uuuu uuuu
TMR1H	保持寄存器的值为 16 位 TMR1 寄存器的高有效字节								xxxx xxxx	uuuu uuuu
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 (LSB)								xxxx xxxx	uuuu uuuu
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 (MSB)								xxxx xxxx	uuuu uuuu
CCP1CON	-	-	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
CCPR2L	捕捉 / 比较 / PWM 寄存器 2 (LSB)								xxxx xxxx	uuuu uuuu
CCPR2H	捕捉 / 比较 / PWM 寄存器 2 (MSB)								xxxx xxxx	uuuu uuuu
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000
PIR2	OSCFIF	CMIF	—	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF	00-0 0000	00-0 0000
PIE2	OSCFIE	CMIE	—	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE	00-0 0000	00-0 0000
IPR2	OSCFIP	CMIP	—	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP	11-1 1111	11-1 1111

图注: x= 未知, u= 不变, - = 未实现, 读作 0。阴影单元格表示捕捉或 Timer1 未使用。

PIC18F2331/2431/4331/4431

15.5 PWM 模式

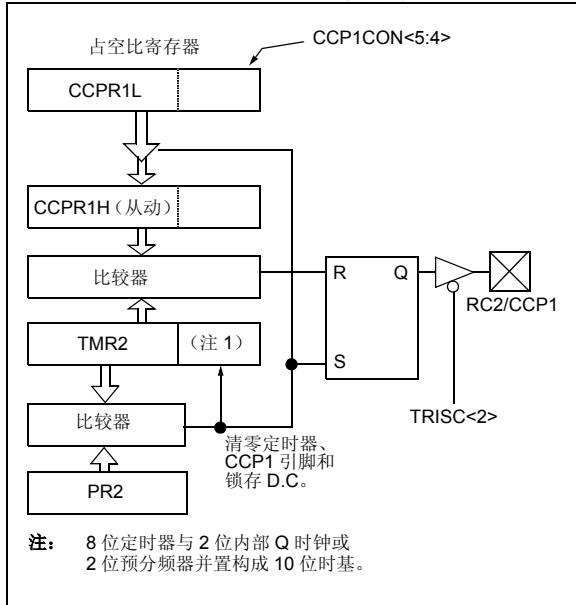
在脉宽调制 (PWM) 模式下, CCP1 引脚会产生高达 10 位分辨率的 PWM 输出。由于 CCP1 引脚与 PORTC 数据锁存器复用, 所以必须清零 TRISC<2> 位以使 CCP1 引脚为输出引脚。

注: 清零 CCP1CON 寄存器将迫使 CCP1 PWM 输出锁存器为缺省低电平。这不是 PORTC I/O 数据锁存器。

图 15-3 所示为 PWM 模式下 CCP 模块的简化框图。

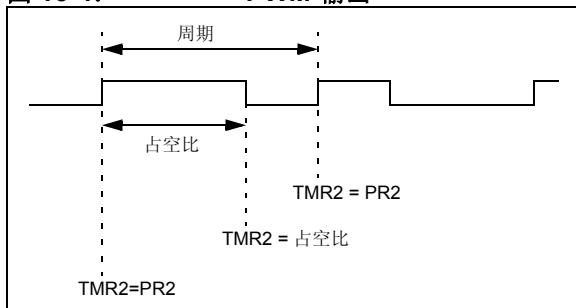
如需了解将 CCP 模块设置为 PWM 操作的详细步骤, 请参阅第 15.5.3 节“设置 PWM 操作”。

图 15-3: PWM 的简化框图



PWM 输出 (图 15-4) 有一个时基 (周期) 和输出为高电平的时间 (占空比)。PWM 的频率是周期的倒数 (1/周期)。

图 15-4: PWM 输出



15.5.1 PWM 周期

通过对 PR2 寄存器进行写操作, 指定 PWM 周期。可使用以下公式来计算 PWM 周期。

公式 15-1:

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

PWM 频率定义为 $1/[\text{PWM 周期}]$ 。当 TMR2 等于 PR2 时, 在下一个递增周期会发生以下三个事件:

- TMR2 被清零
- CCP1 引脚被置位 (如果 PWM 占空比 = 0%, CCP1 引脚不会被置位)
- PWM 占空比从 CCPR1L 复制到 CCPR1H

注: Timer2 后分频器 (见第 13.0 节“Timer2 模块”)与 PWM 频率无关。使用后分频器, 可以以不同于 PWM 频率的速率进行数据更新。

15.5.2 PWM 占空比

通过对 CCPR1L 寄存器和 CCP1CON<5:4> 位进行写操作, 指定 PWM 的占空比。最高分辨率可达 10 位。CCPR1L 包含八个 MSb, CCP1CON<5:4> 包含两个 LSb。CCPR1L:CCP1CON<5:4> 就代表这 10 位值。PWM 占空比根据以下公式计算。

公式 15-2:

$$\text{PWM 占空比} = \frac{(\text{CCPR1L}:\text{CCP1CON}\langle 5:4 \rangle) \cdot \text{Tosc}}{\text{TMR2 预分频值}}$$

任何时候都可以写 CCPR1L 和 CCP1CON<5:4>, 但直到 PR2 与 TMR2 中的值相符 (即周期结束时), 占空比的值才被复制到 CCPR1H。在 PWM 模式下, CCPR1H 是只读寄存器。

PIC18F2331/2431/4331/4431

CCPR1H 寄存器和一个 2 位内部锁存器为 PWM 占空比提供双重缓冲。双重缓冲对 PWM 的无毛刺操作是极其重要的。当 CCPR1H 和 2 位锁存值与 TMR2 加上内部 2 位 Q 时钟或 TMR2 预分频器的两位相符时，CCP1 引脚被清零。对于给定的 PWM 频率，最大 PWM 分辨率（位）可以根据以下公式计算。

公式 15-3:

$$\text{PWM 分辨率(最大)} = \frac{\log\left(\frac{F_{\text{OSC}}}{F_{\text{PWM}}}\right)}{\log(2)} \text{ 位}$$

注：如果 PWM 占空比的值大于 PWM 周期，CCP1 引脚将不会被清零。

15.5.3 设置 PWM 操作

将 CCP 模块配置为 PWM 操作，应执行以下步骤：

1. 写 PR2 寄存器来设置 PWM 周期
2. 写 CCPR1L 寄存器和 CCP1CON<5:4> 位来设置 PWM 的占空比。
3. 清零 TRISC<2> 位，使 CCP1 引脚为输出。
4. 写 T2CON 来设置 TMR2 预分频值并使能 Timer2。
5. 将 CCP1 模块配置为 PWM 操作。

表 15-3: 40 MHz 时 PWM 频率和分辨率示例

PWM 频率	2.44 kHz	9.77 kHz	39.06 kHz	156.25 kHz	312.50 kHz	416.67 kHz
定时器预分频器 (1,4,16)	16	4	1	1	1	1
PR2 值	FFh	FFh	FFh	3Fh	1Fh	17h
最大分辨率 (位)	10	10	10	8	7	6.58

表 15-4: 与 PWM 和 TIMER2 相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
IPR1	—	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	-111 1111	-111 1111
TRISC	PORTC 数据方向寄存器								1111 1111	1111 1111
TMR2	Timer2 模块寄存器								0000 0000	0000 0000
PR2	Timer2 模块周期寄存器								1111 1111	1111 1111
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
CCPR1L	捕捉 / 比较 / PWM 寄存器 1 (LSB)								xxxx xxxx	uuuu uuuu
CCPR1H	捕捉 / 比较 / PWM 寄存器 1 (MSB)								xxxx xxxx	uuuu uuuu
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
CCPR2L	捕捉 / 比较 / PWM 寄存器 2 (LSB)								xxxx xxxx	uuuu uuuu
CCPR2H	捕捉 / 比较 / PWM 寄存器 2 (MSB)								xxxx xxxx	uuuu uuuu
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

图注：x= 未知，u= 不变，- = 未实现，读作 0。阴影单元格表示 PWM 或 Timer2 未使用。

PIC18F2331/2431/4331/4431

注:

16.0 运动反馈模块

运动反馈模块是为运动反馈应用设计的特殊功能外设。配合功率控制 PWM 模块（参见第 17.0 节“功率控制 PWM 模块”），它可以为多种电动机提供各种不同的控制解决方案。

此模块实际上由两个硬件子模块组成：

- 输入捕捉模块（Input Capture, IC）
- 正交编码器接口（Quadrature Encoder Interface, QEI）

这些模块和 Timer5（参见第 14.0 节“Timer5 模块”）一起为运动和控制应用提供了许多选择。

IC 和 QEI 子模块的很多功能都是完全可编程的，因此为外设结构提供了很好的灵活性，可以满足一系列系统内应用。表 16-1 概括了这些可用功能。图 16-1 显示了整个运动反馈模块的简化框图。

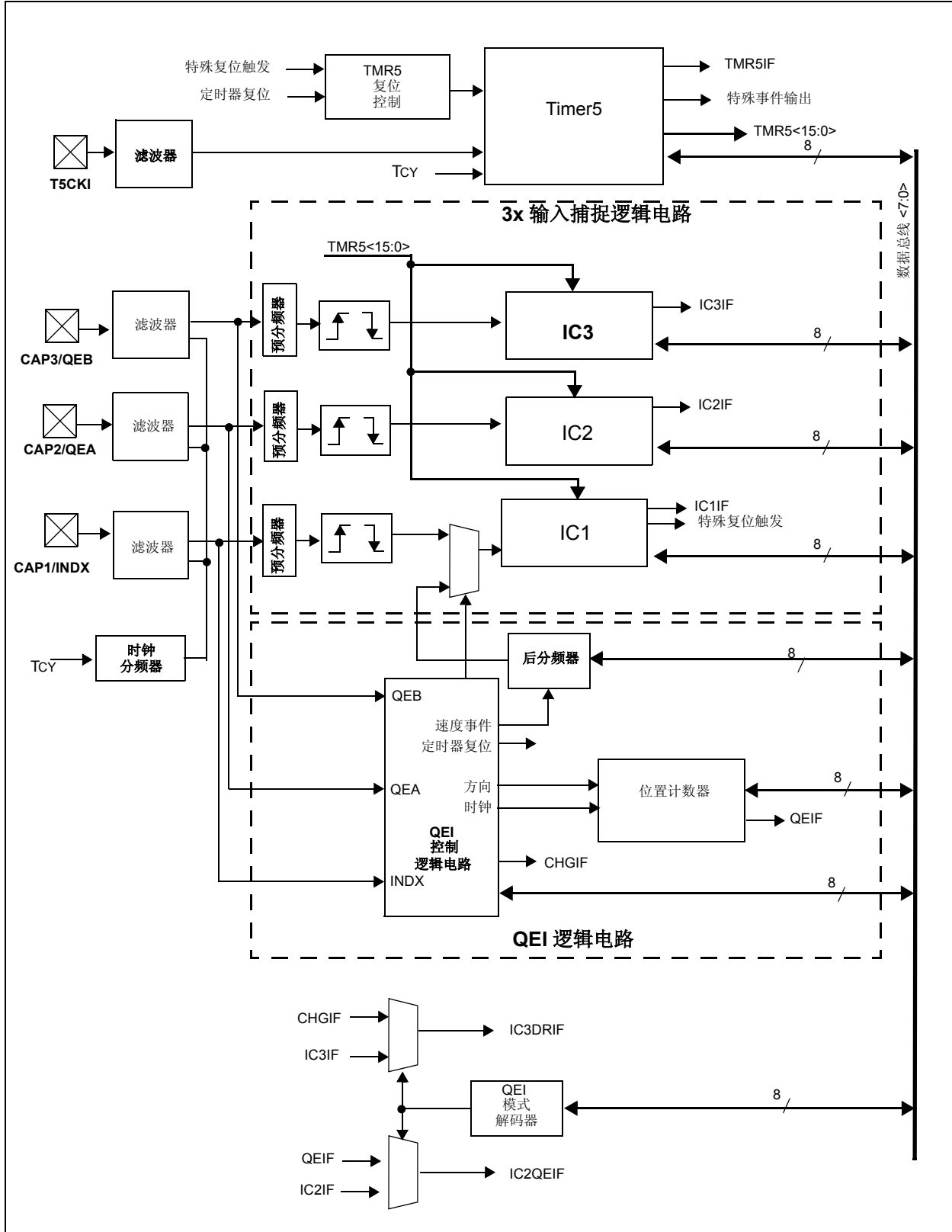
注： 因为 IC 和 QEI 子模块的输入引脚是共用的，所以任何时候，这两个子模块中只能有一个可以使用。如果两个模块都被使用，QEI 子模块有更高的优先级。

表 16-1: 运动反馈模块功能概要

子模块	模式	功能	定时器	功能
IC (3x)	<ul style="list-style-type: none"> • 同步 • 输入捕捉 	<ul style="list-style-type: none"> • 灵活的输入捕捉模式 • 可用预分频器 • 可选时基复位 • ADC 采样 / 转换特殊事件触发或可选 TMR5 复位功能（仅 CAP1） • 休眠唤醒功能 • 可选中断频率 • 可选噪声滤波器 	TMR5	<ul style="list-style-type: none"> • 3x 输入捕捉（边沿捕捉、脉宽、周期测量、变化捕捉） • 在 CAP1 输入引脚上的特殊事件触发 A/D 转换
QEI	QEI	<ul style="list-style-type: none"> • 检测位置 • 检测转动方向 • 大带宽 (Fcy/16) • 可选噪声滤波器 	16 位位置计数器	<ul style="list-style-type: none"> • 位置测量 • 转动方向状态
	速度测量	<ul style="list-style-type: none"> • 2x 和 4x 更新模式 • 速度事件后分频器 • 低转速计数器溢出标志 • 利用输入捕捉 (IC1) 逻辑电路 • 支持高速和低速 	TMR5	<ul style="list-style-type: none"> • 精确的速度测量 • 转动方向状态

PIC18F2331/2431/4331/4431

图 16-1: 运动反馈模块框图



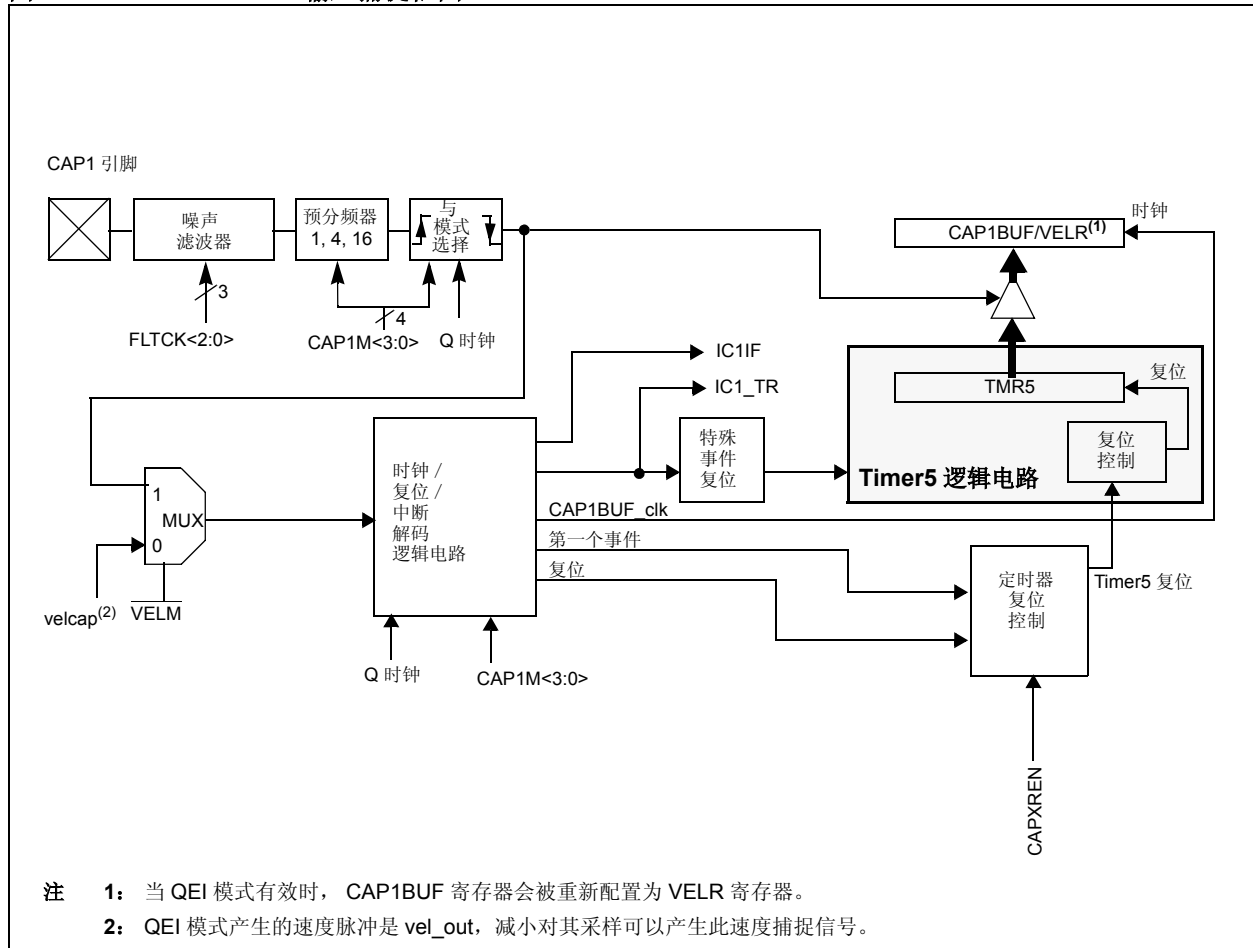
16.1 输入捕捉

输入捕捉 (IC) 子模块实现了以下功能:

- CAP1、CAP2 和 CAP3 引脚上的 3 个独立输入捕捉通道 (16 位 / 通道)
- 每个通道的边沿触发、周期或脉宽测量工作模式
- 每个输入捕捉通道的可编程预分频器
- 特殊事件触发输出 (仅 IC1)
- 每个捕捉输入的可选噪声滤波器

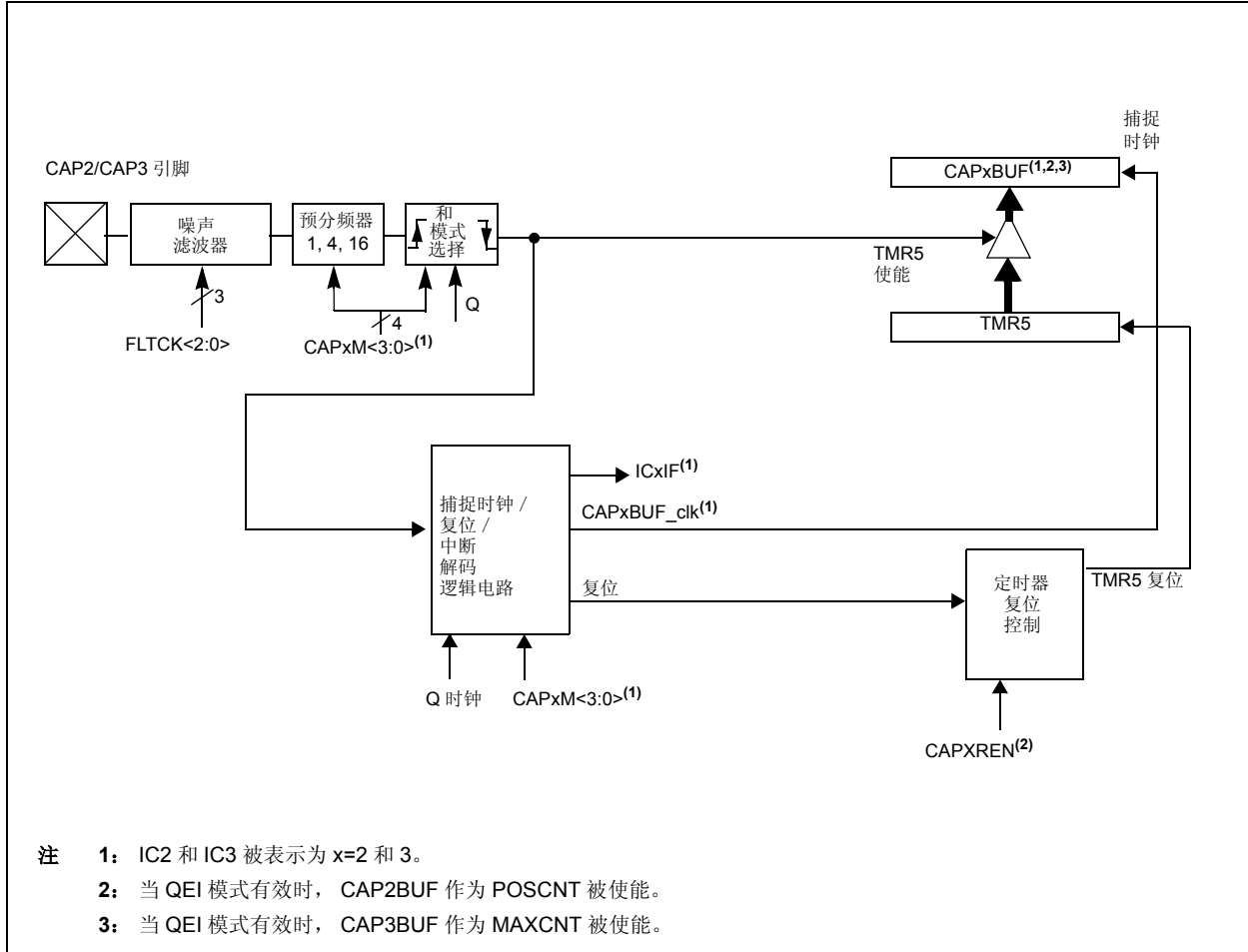
输入通道 (IC1) 包括一个特殊事件触发器, 可以配置用于速度测量模式。图 16-2 显示了其框图。IC2 和 IC3 与此相似, 但缺少特殊事件触发器功能或附加的速度测量逻辑电路。图 16-3 中显示了有代表性的框图。请注意时基是 Timer5。

图 16-2: IC1 输入捕捉框图



PIC18F2331/2431/4331/4431

图 16-3: IC2 和 IC3 的输入捕捉框图



PIC18F2331/2431/4331/4431

3 条输入捕捉通道由输入捕捉控制寄存器 CAP1CON、CAP2CON 和 CAP3CON 控制。每条通道都由其专用寄存器独立配置。除了特殊事件触发器（参见第 16.1.8 节“特殊事件触发（仅适用于 CAP1）”）外，其他寄存器的实现都是相同的。寄存器 16-1 所示为典型的捕捉控制寄存器。

寄存器 16-1: CAPxCON: 输入捕捉控制寄存器

U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	CAPxREN	—	—	CAPxM3	CAPxM2	CAPxM1	CAPxM0	
bit 7								bit 0

bit 7 **未实现位:** 读作 0

bit 6 **CAPxREN:** 时基复位使能位
1 = 使能
0 = 在捕捉时禁止所选时基复位

bit 5 **未实现位:** 读作 0

bit 4 **未实现位:** 读作 0

bit 3-0 **CAPxM3:CAPxM0:** 输入捕捉 1 (ICx) 模式选择位
1111 = 特殊事件触发模式。触发发生在 CAP1 输入引脚⁽¹⁾ 的每个上升沿
1110 = 特殊事件触发模式。触发发生在 CAP1 输入引脚⁽¹⁾ 的每个下降沿
1101 = 未使用
1100 = 未使用
1011 = 未使用
1010 = 未使用
1001 = 未使用
1000 = 在每次 CAPx 输入状态改变时捕捉
0111 = 脉宽测量模式，每个上升沿到下降沿
0110 = 脉宽测量模式，每个下降沿到上升沿
0101 = 频率测量模式，每个上升沿
0100 = 捕捉模式，每 16 个上升沿
0011 = 捕捉模式，每 4 个上升沿
0010 = 捕捉模式，每个上升沿
0001 = 捕捉模式，每个下降沿
0000 = 输入捕捉 1 (ICx) 关闭

注 1: 特殊事件触发器仅在 CAP1 引脚上有效。CAP2 和 CAP3 不使用此配置。

图注:

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

注: 在本章中，在提到可能与具体捕捉模式有关的寄存器和位名称，会使用“x”代替通道编号表示一般的寄存器和位。例如，“CAPxREN”指 CAP1CON、CAP2CON 或 CAP3CON 中的捕捉复位使能位。

处于计数器模式时，计数器只能配置为同步计数器 (TMR5SYNC = 0)。当配置为异步模式时，IC 模块将不会正常工作。

PIC18F2331/2431/4331/4431

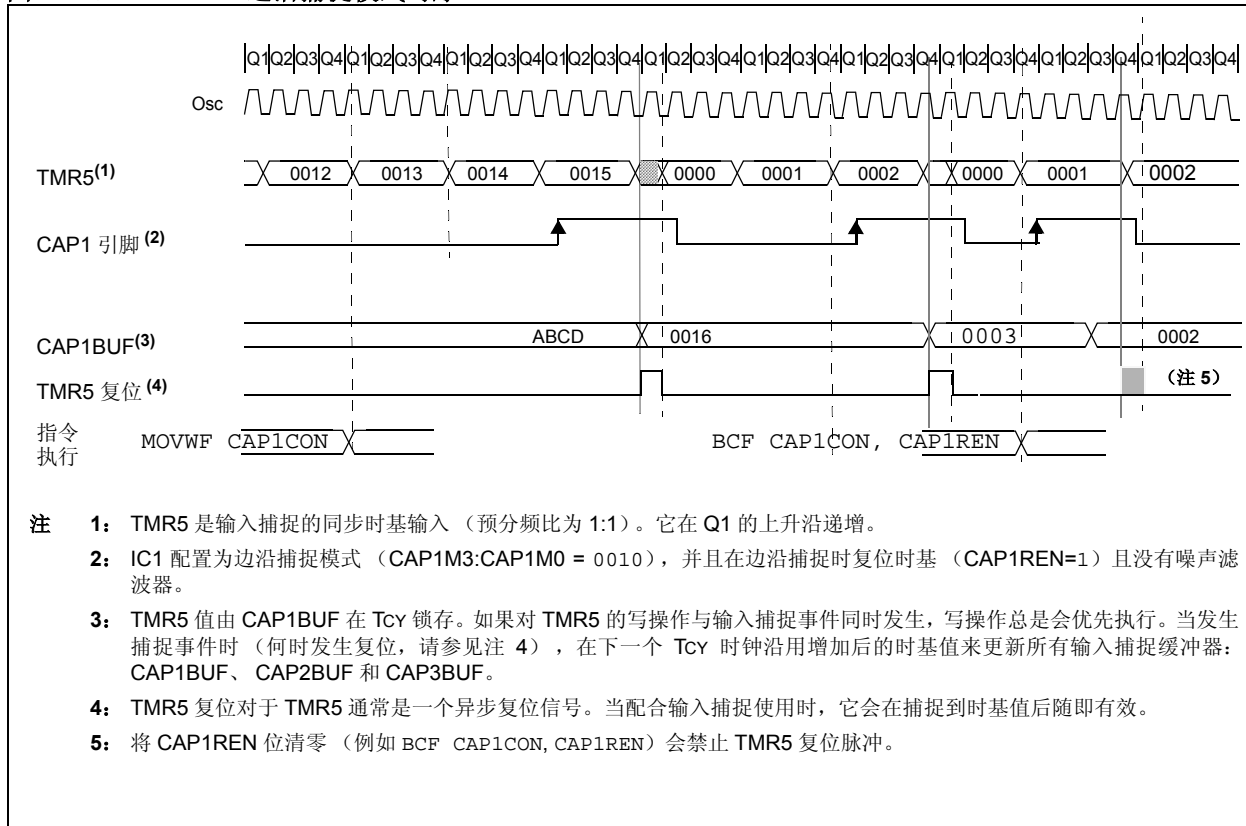
- 注 1:** 当禁止输入捕捉模块 (CAPxM=0000) 时, 输入捕捉预分频器会复位 (清零)。
- 2:** 如果没有首先禁止此模块, 输入捕捉模式便发生改变进入新输入捕捉模式, 可能会产生错误的中断 (或 IC1 上出现特殊事件触发)。用户应该 (1) 在进入另一个模式前禁止输入捕捉或 (2) 禁止 IC 中断以免在 IC 模式改变期间产生错误中断。
- 3:** 在 IC 模式改变期间, 预分频计数器不会清零, 因此在新 IC 模式下的第一个捕捉结果可能来自非零预分频器。

16.1.1 边沿捕捉模式

在此模式中, 会在每个上升沿、每个下降沿、每 4 个上升沿或每 16 个上升沿捕捉时基的值。通过同步锁存器对输入捕捉引脚 (CAP1、CAP2 和 CAP3) 上的信号沿进行采样。此信号用于在随后的 Q1 时钟周期载入输入捕捉缓冲器 (ICxBUF 寄存器) (参见图 16-4)。结果, Timer5 可能会复位为 0 (Q1 紧接着捕捉事件) 或保持自由运行 (由 CAPxCON 寄存器中的捕捉复位使能位 CAPxREN 决定)。

注: 在设置输入捕捉模式 (即 MOVWF CAP1CON) 后的第一个捕捉边沿, Timer5 的内容总是会被捕捉到相应的输入捕捉缓冲器 (即 CAPxBUF)。Timer5 也可以复位, 但是这取决于捕捉复位使能位 (CAPxREN) 的设置状态, 参见图 16-4。

图 16-4: 边沿捕捉模式时序



16.1.2 周期测量模式

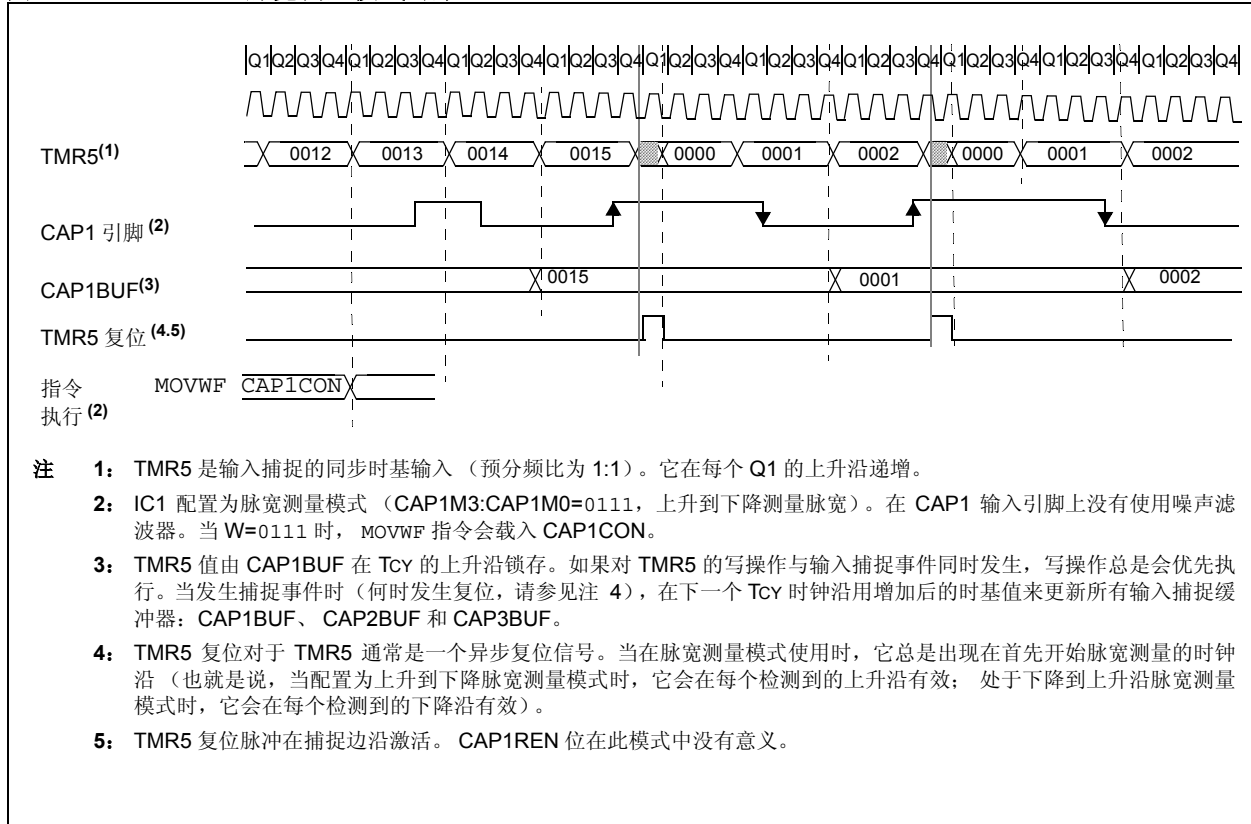
将 CAPxM3:CAPxM0 设置为 0101 可以选择周期测量模式。在此模式下，Timer5 的值在输入捕捉触发信号的上升沿锁存到 CAPxBUF 寄存器，然后 Timer5 会在下一个 Tcy 复位为 0000h（或者也可通过将 CAPxREN 置位为 1）（参见图 16-4 中的捕捉和复位关系）。

16.1.3 脉宽测量模式

可将脉宽测量模式配置为两种不同的边沿序列，这样脉宽可以基于 CAPx 输入引脚的下降沿到上升沿（CAPxM3:CAPxM0=0110），也可以基于上升到下降沿（CAPxM3:CAPxM0=0111）。

当开始第一次测量时，Timer5 总是会在边沿复位。例如，当测量基于下降到上升沿时，Timer5 首先会在下降沿复位，然后在上升沿捕捉到定时器值。当进入脉宽测量模式时，总是能捕捉到 CAPx 引脚上的第一个边信号沿。TMR5 值在第一个有效信号沿复位（参见图 16-5）。

图 16-5: 脉宽测量模式时序



PIC18F2331/2431/4331/4431

16.1.3.1 脉宽测量时序

只有 CAPx 输入引脚上出现的高低脉宽超过一个 Tcy 时钟周期时，才可能保证脉宽测量的精度。此局限性因所选的模式而异。

- 当 CAPxM3:CAPxM0=0110 时（上升到下降沿延迟），CAPx 输入的高脉宽（TccH）必须大于 Tcy+10 ns。
- 当 CAPxM3:CAPxM0=0111 时（下降到上升沿延迟），CAPx 输入的低脉宽（TccL）必须大于 Tcy+10 ns。

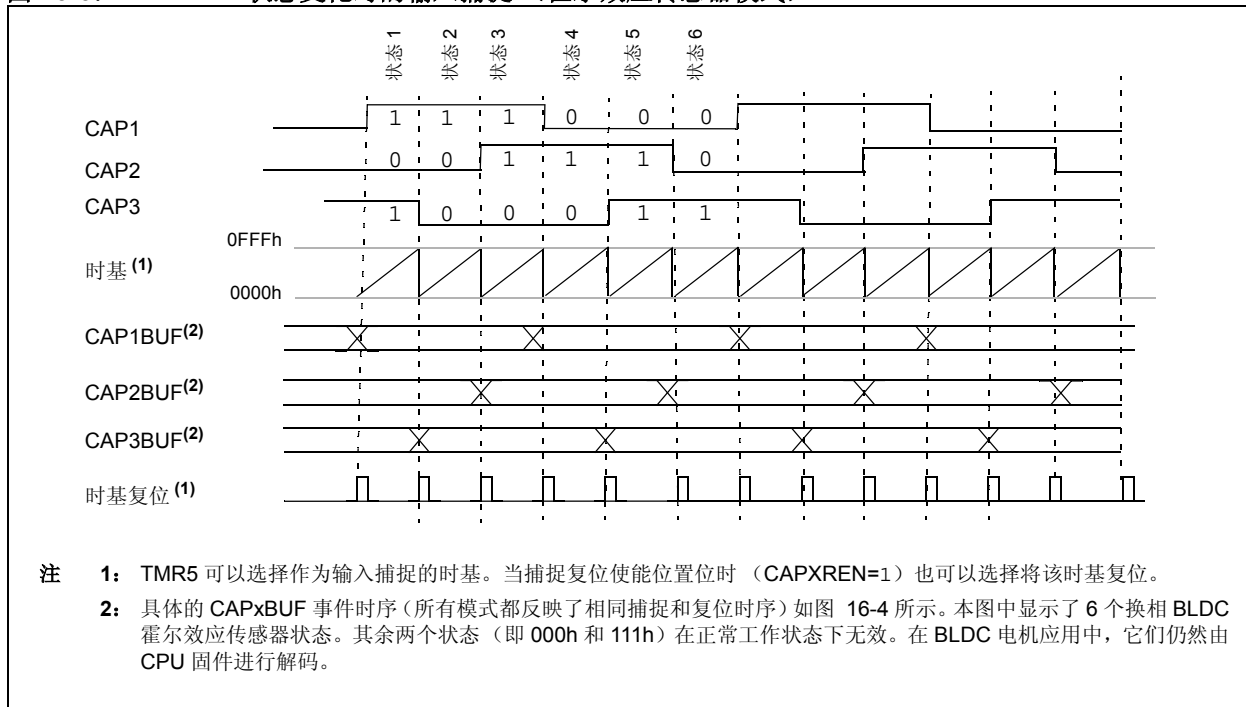
- 注 1:** 周期测量模式会在采样输入捕捉的第 2 个上升沿时产生有效结果。在初始化后的第一个有效沿期间锁存的 CAPxBUF 值无效。
- 2:** 脉宽测量模式将会在输入捕捉采样第一个输入信号沿时锁存定时器值。

16.1.4 状态变化时的输入捕捉

当 CAPxM3:CAPxM0=1000 时，会在 CAPx 输入的每次信号改变时捕捉此值。如果所有 3 个捕捉通道都配置为此模式，这种 3 输入捕捉可作为霍尔效应传感器状态变换检测器使用。可以捕获 Timer5 的值，Timer5 复位并产生中断。CAP1、CAP2 或 CAP3 的任何变化都会被检测，同时会捕捉相关的时基数。

对于该模式下的位置和速度测量，可以选择复位定时器（有关复位选项，请参见第 16.1.6 节“Timer5 复位”）。

图 16-6: 状态变化时的输入捕捉（霍尔效应传感器模式）



16.1.5 进入输入捕捉模式和捕捉时序

以下为进入任何输入捕捉模式时的有效操作概述：

1. 通过设置模式选择位（CAPxM3:CAPxM0）将模块配置为某一个捕捉模式后，首先检测到的信号沿会捕捉 Timer5 的值并将其保存在 CAPxBUF 寄存器。然后定时器复位（视 CAPxREN 位的设置而定）并根据设置情况开始递增，参见图 16-4、图 16-5 和图 16-6。
2. 在所有的边沿，捕捉逻辑电路会执行以下操作：
 - a) 解码输入捕捉模式并确定有效沿。
 - b) 检查 CAPxREN 位以确定 Timer5 是否复位。
 - c) 在每个有效沿上，将 Timer5 的值记录在输入捕捉缓冲器（CAPxBUF）中。
 - d) 当使能了 CAPxREN 位时，在捕捉到 Timer5 的值后复位该定时器。这种情况下，Timer5 在每个有效捕捉沿复位。
 - e) 所有后续的捕捉沿事件将重复步骤 a 到 d，直到由用户固件、POR 或 BOR 终止此工作模式。
 - f) 当在各种输入捕捉模式之间切换时，定时器值不受影响。

16.1.6 TIMER5 复位

每个输入捕捉触发器能够可选地复位 TMR5。捕捉复位使能位 CAPxREN 使用此使能复位信号使能或禁止自动复位捕捉事件的时基。当 CAPxREN 置位时，所有捕捉事件都会使所选定时器复位。如果 CAPxREN 清零，则禁止复位（参见图 16-4、图 16-5 和图 16-6）。

注： CAPxREN 位在脉宽测量模式中不起作用。

16.1.7 IC 中断

IC 模块可以在四种工作模式下产生中断并将其中一个中断捕捉标志位（IC1IF、IC2QEIF 或 IC3DRIF）置位。中断标志位是否置位取决于事件发生的通道。这四种模式为：

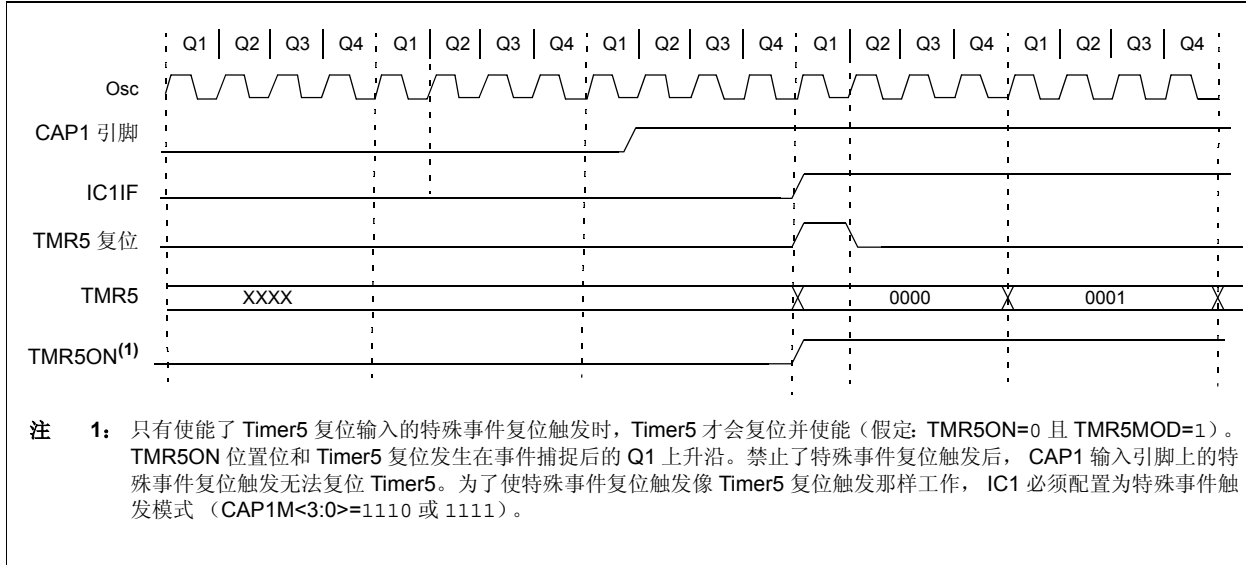
- 边沿捕捉（CAPxM3:CAPxM0=0001、0010、0011 或 0100）
- 周期测量事件（CAPxM3:CAPxM0=0101）
- 脉宽测量事件（CAPxM3:CAPxM0=0110 或 0111）
- 状态改变事件（CAPxM3:CAPxM0=1000）

注： 只有在特殊事件触发模式下才会在 CAP1 输入（CAP1M2:CAP1M0>=1110 和 1111）上发生特殊事件触发。在此模式中不会置位 IC1IF 中断标志位。

图 16-7 显示了中断和特殊触发事件的时序。在 Q2 周期的上升沿检测任意有效沿，并在 Q4 周期的上升沿进行传播。如果有效沿正好发生在此时间以后（例如在 Q2 的下降沿），则它要在下一个 Q2 上升沿才能被识别。

PIC18F2331/2431/4331/4431

图 16-7: CAPxIF 中断和 IC1 特殊事件触发器



16.1.8 特殊事件触发 (仅适用于 CAP1)

IC1 的特殊事件触发模式 (CAP1M3:CAP1M0=1110 或 1111) 会使能特殊事件触发信号。触发信号可以作为 TMR5 的特殊事件复位输入信号, 当 IC1 上发生特定事件时将 TMR5 定时器复位。表 16-2 概括了这些事件。

表 16-2: 特殊事件触发器

CAP1M3: CAP1M0	描述
1110	在 CAP1 输入引脚的每个下降沿触发
1111	在 CAP1 输入引脚的每个下降沿触发

16.1.9 工作模式概要

表 16-3 给出与 TMR5 定时器资源配合使用时的输入捕捉配置概要。

16.1.10 其他工作模式

虽然 IC 和 QEI 子模块是互斥的, 但仍然可重新配置 IC, 使其同 QEI 模块一起工作以执行特定功能。实际上, QEI 会“借用” IC 的硬件资源来执行这些操作。

QEI 使用 IC1 通道中的专用硬件进行速度测量。CAP1BUF 寄存器被重新映射为 VREG 寄存器。其工作原理和用法请参见第 16.2.6 节“速度测量”。

在 QEI 模式下, IC2 和 IC3 通道的 CAP2BUF 和 CAP3BUF 寄存器用于位置测定。它们也分别被重新映射为 POSCNT 和 MAXCNT 缓冲寄存器。

PIC18F2331/2431/4331/4431

表 16-3: 输入捕捉时基复位概要

引脚	CAPxM	模式	定时器	捕捉复位定时器	描述
CAP1	0001-0100	边沿捕捉	TMR5	可选 ⁽¹⁾	简单边沿捕捉模式（包括一个可选的预分频器）
	0101	周期测量	TMR5	可选 ⁽¹⁾	在周期边沿捕捉 Timer5
	0110-0111	脉宽测量	TMR5	总是	在脉冲边沿捕捉 Timer5
	1000	状态改变时的输入捕捉	TMR5	可选 ⁽¹⁾	状态改变时捕捉 Timer5
	1110-1111	特殊事件触发（上升沿或下降沿）	TMR5	可选 ⁽²⁾	作为特殊事件触发，用于与 Timer5 或其他外设模块一起使用
CAP2	0001-0100	边沿捕捉	TMR5	可选 ⁽¹⁾	简单边沿捕捉模式（包括一个可选的预分频器）
	0101	周期测量	TMR5	可选 ⁽¹⁾	在周期边沿捕捉 Timer5
	0110-0111	脉宽测量	TMR5	总是	在脉冲边沿捕捉 Timer5
	1000	状态改变时的输入捕捉	TMR5	可选 ⁽¹⁾	状态改变时捕捉 Timer5
CAP3	0001-0100	边沿捕捉	TMR5	可选 ⁽¹⁾	简单边沿捕捉模式（包括一个可选的预分频器）
	0101	周期测量	TMR5	可选 ⁽¹⁾	在周期边沿捕捉 Timer5
	0110-0111	脉宽测量	TMR5	总是	在脉冲边沿捕捉 Timer5
	1000	状态改变时的输入捕捉	TMR5	可选 ⁽¹⁾	状态改变时捕捉 Timer5

- 注 1: 仅当 CAPxRE=1 时，在发生捕捉事件时 Timer5 才能被复位。
 注 2: 除非在 T5CON 寄存器中 RESEN=0，否则触发模式不会将 Timer5 复位。

PIC18F2331/2431/4331/4431

16.2 正交编码器接口

正交编码器接口（QEI）解码速度和运动传感器信息。它可以在任何将正交编码器用于反馈的应用中使用。此接口实现了以下功能：

- 三种 QEI 输入信号：两个相位信号（QEA 和 QEB）和一个索引信号（INDX）
- 移动方向检测，带方向变化中断（IC3DRIF）
- 16 位递增 / 递减位置计数器
- 标准高精度位置记录模式
- 两个位置刷新模式（x2 和 x4）
- 速度测量，带有用于高速速度测量的可编程后分频器
- 位置计数器中断（PIR3 寄存器的 IC2QEIF）
- 速度控制中断（PIR3 寄存器的 IC1IF）

QEI 子模块有三个主要部分：QEI 控制逻辑电路、位置计数器和速度后分频器。

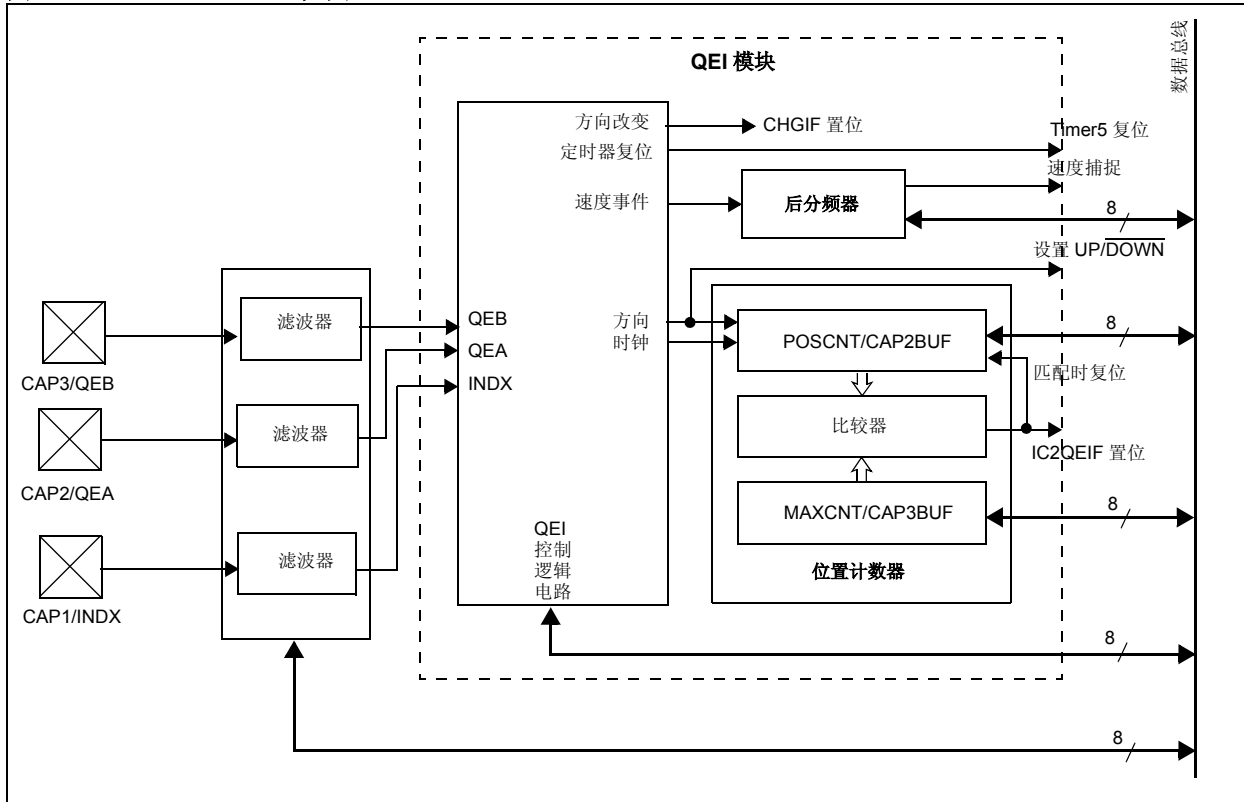
QEI 控制逻辑电路检测 QEA 或 QEB 相输入引脚的前沿，产生计数脉冲，然后发送到位置计数器逻辑电路。它还对索引输入信号（INDX）进行采样，并产生转动方向信号（正向 / 反向）和速度事件信号。

位置计数器则作为积分器，用来记录所经过的距离。QEA 和 QEB 输入边沿作为激励源产生输入时钟，此时钟信号使位置计数器寄存器（POSCNT）递增。此寄存器是在 QEA 输入沿还是 QEA 和 QEB 输入沿递增，这取决于工作模式。如果与周期寄存器 MAXCNT 匹配或因为外部索引脉冲输入信号（INDX）产生计满回零，该寄存器复位。如果允许位置计数器中断，当 POSCNT 复位时会产生一个中断。

速度后分频器减少对速度脉冲（此脉冲信号按指定比例递增速度计数器）的采样。它实际上是将速度脉冲的数量分成一定数量的输入为一个输出，以便在过程中保留脉宽。

图 16-8 所示为 QEI 模块的简化框图。

图 16-8: QEI 框图



PIC18F2331/2431/4331/4431

16.2.1 QEI 配置

QEI 模块与输入捕捉模块共用其输入引脚。输入信号是互斥的，每次只能使能 IC 模块或 QEI 模块中的一个。同样，因为 IC 和 QEI 复用到同样的输入引脚，所有可编程噪声滤波器只能由一个模块专用。

QEI 的工作模式由 QEICON 配置寄存器控制。参见寄存器 16-2。

注： 如果同时使能 QEI 和 IC，则 QEI 具有优先权，IC 将保持禁止。

寄存器 16-2: QEICON: 正交编码器接口控制寄存器

	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	VELM	ERROR	UP/DOWN	QEIM2	QEIM1	QEIM0	PDEC1	PDEC0
bit 7								bit 0

- bit 7 **VELM:** 速度模式位
 1 = 禁止速度模式
 0 = 使能速度模式
- bit 6 **ERROR:** QEI 错误位 ⁽¹⁾
 1 = 位置计数器 ⁽⁴⁾ 上溢或下溢
 0 = 没有上溢或下溢
- bit 5 **UP/DOWN:** 转动方向状态位 ⁽¹⁾
 1 = 正向
 0 = 逆向
- bit 4-2 **QEIM2:QEIM0:** QEI 模式位 ^(2,3)
 111 = 未使用
 110 = 使能 QEI 4x 刷新模式；在周期匹配 (POSCNT=MAXCNT) 时将位置计数器复位
 101 = 使能 QEI 4x 刷新模式；INDX 将位置计数器复位
 100 = 未使用
 010 = 使能 QEI 2x 刷新模式；在周期匹配 (POSCNT=MAXCNT) 时将位置计数器复位
 001 = 使能 QEI 2x 刷新模式；INDX 将位置计数器复位
 000 = 禁止 QEI
- bit 1-0 **PDEC1:PDEC0:** 速度脉冲降低比例位
 11 = 1:64
 10 = 1:16
 01 = 1:4
 00 = 1:1

- 注 1:** QEI 必须使能并处于索引模式。
- 2:** 要使能 CAP1、CAP2 或 CAP3 输入，必须将 QEI 模式选择位清零 (=000)。如果 QEI 和 IC 模块同时使能，QEI 具有优先权。
- 3:** 使能任何一个 QEI 模式都会将 IC 缓冲寄存器 CAP1BUFH、CAP1BUFL、CAP2BUFH、CAP2BUFL、CAP3BUFH 和 CAP3BUFL 分别重新映射为 QEI 的 VREGH、VREGL、POSCNTH、POSCNTL、MAXCNTH 和 MAXCNTL 寄存器。
- 4:** ERROR 位必须用软件清零。

图注:

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

PIC18F2331/2431/4331/4431

16.2.2 QEI 模式

位置测量分辨率取决于位置计数器寄存器 POSCNT 递增的频率。有两种 QEI 刷新模式可测量转子位置：QEI x2 和 QEI x4。

表 16-4: QEI 模式

QEIM2 QEIM0	模式 / 复位	描述
000	—	禁止 QEI ⁽¹⁾
001	x2 刷新 / 索引脉冲	每个 QEA 脉冲两个时钟周期。 INDX 将 POSCNT 复位。
010	x2 刷新 / 周期匹配	每个 QEA 脉冲两个时钟周期。 周期匹配 (MAXCNT) 将 POSCNT 复位。
011	—	未使用
100	—	未使用
101	x4 刷新 / 索引	每个 QEA 和 QEB 脉冲对四个 时钟周期。 INDX 将 POSCNT 复位。
110	x4 刷新 / 周期匹配	每个 QEA 和 QEB 脉冲对四个 时钟周期。 周期匹配 (MAXCNT) 将 POSCNT 复位。
111	—	未使用

注 1: 禁止 QEI 模块。在此模式下会完全禁止位置计数器和速度测量功能。

16.2.2.1 QEI x2 刷新模式

通过将 QEI 模式选择位 (QEIM2:QEIM0) 置为 001 或 010 可选择 QEI x2 刷新模式。在此模式下, QEI 逻辑电路只会检测 QEA 输入的每个边沿。QEA 信号的每个上升和下降沿为位置计数器提供时钟源。

即使当 POSCNT 寄存器对与 MAXCNT 相等 (QEIM2:QEIM0=010) 时, 位置计数器也可以通过 INDX 引脚的输入或周期匹配 (QEIM2:QEIM0=001) 复位。

16.2.2.2 QEI 4x 刷新模式

QEI x4 刷新模式为转子位置提供了更高的分辨率, 因为比起 QEI x2 模式而言, 计数器对每个 QEA/QEB 输入脉冲对的递增或递减更加频繁。此模式是通过将 QEI 模式选择位设置为 101 或 110 选择的。在 QEI x4 模式中, 相位测量是在 QEA 和 QEB 输入的上升沿和下降沿进行的。每个 QEA 和 QEB 边沿就为位置计数器提供一个时钟信号。

和 QEI x2 模式类似, 可以通过引脚的输入信号 (QEIM2:QEIM0=101) 或周期匹配事件 (QEIM2:QEIM0=010) 将位置计数器复位。

16.2.3 QEI 工作原理

位置计数器寄存器对 (POSCNTH: POSCNTL) 用作积分器, 其值与传感器转子位置 (对应于所检测到的有效沿数量) 成正比。POSCNT 可以递增也可以递减, 这取决于很多由 QEI 逻辑电路解码的可选因素。其中包括所选的计数模式、QEA 与 QEB 的相位关系 (超前/滞后)、转动方向以及是否发生复位事件等。此逻辑电路将在后续章节详细讲述。

16.2.3.1 边沿与相位检测

在第一步中, 检测到了 QEA 和 QEB 的有效边沿, 并确定它们之间的相位关系。位置计数器根据所选 QEI 模式改变。

在 QEI x2 刷新模式, 位置计数器根据 QEA 和 QEB 信号的相位关系在每个 QEA 边沿递增或递减。

在 QEI x4 刷新模式, 位置计数器根据 QEA 和 QEB 信号的相位关系在每个 QEA 和 QEB 边沿递增或递减。例如, 如果 QEA 超前于 QEB, 位置计数器将加 1; 如果 QEB 滞后于 QEA, 则位置计数器减 1。

16.2.3.2 计数方向

QEI 控制逻辑电路会产生一个信号来置位 UP/DOWN 位 (QEICON<5>); 因此也就决定了计数的方向。当 QEA 超前于 QEB 时, UP/DOWN 位置位 (=1), 而位置计数器会在每个有效边沿递增。当 QEA 滞后于 QEB 时, UP/DOWN 位清零, 而位置计数器会在每个有效边沿递减。

表 16-5: 转动方向

当前检测到的 信号	上一个检测到的信号				位置 控制 ⁽¹⁾
	上升		下降		
	QEA	QEB	QEA	QEB	
QEA 上升沿				x	递增
		x			递减
QEA 下降沿				x	递减
		x			递增
QEB 上升沿	x				递增
			x		递减
QEB 下降沿			x		递增
	x				递减

注 1: 当 UP/DOWN=1 时, 位置计数器递增;
当 UP/DOWN=0 时, 位置计数器递减。

16.2.3.3 复位和刷新事件

位置计数器会一直递增或递减，直到发生以下事件之一。事件类型和事件发生时的转动方向决定是发生寄存器复位还是刷新。

1. 在 **INDX** 输入引脚检测到了索引脉冲 (QEIM2:QEIM0=001)。

如果编码器**正向**移动，POSCNT会在检测到索引标记 **INDX** 后的下一个时钟沿复位 (00h)。检测到 **INDX** 上升沿后，位置计数器会在 **QEA** 或 **QEB** 边沿复位。

如果编码器**反向**移动，会在检测到 **INDX** 下降沿后的下一个正交脉冲边沿 (**QEA** 或 **QEB**) 将 **MAXCNT** 寄存器中的值装入 **POSCNT**。

2. 发生了 **POSTCNT/MAXCNT** 周期匹配 (QEIM2:QEIM0=010)。

如果编码器**正向**移动，**POSCNT** 会在 **POSCNT=MAXCNT** 后的下一个时钟边沿复位 (00h)。复位后，在下一个 **Tcy** 会触发中断事件 (参见图 16-10)。

如果编码器**反向**移动且 **POSCNT** 的值成为 00h，**MAXCNT** 寄存器的内容会在下一个时钟边沿装入 **POSCNT**。装入操作后，在下一个 **Tcy** 会触发中断事件 (参见图 16-10)。

在 **QEI** 模式改变或完全禁止 **QEI** 时位置计数器的值不受影响。

16.2.4 QEI 中断

基于下列事件发生位置计数器中断，将中断标志位 (**IC2QEIF**) 置位：

- **POSCNT/MAXCNT** 周期匹配事件 (QEIM2:QEIM0=010 或 110)
- **POSCNT** 计满回零 (FFFFh 到 0000h)，仅适用于周期模式 (QEIM2:QEIM0=010 或 110)
- 在 **INDX** 检测到索引脉冲。

图 16-10 和图 16-11 显示了 **IC2QEIF** 的中断时序图。

当方向改变时，方向改变中断标志位 (**IC3DRIF**) 会在下一个 **Tcy** 时钟周期置位 (参见图 16-10)。

如果位置计数器在索引模式计满回零，**ERROR** 位会置位。

16.2.5 QEI 采样时序

正交输入信号 **QEA** 和 **QEB** 的正交频率可能不同。最小正交输入周期 **Tqei** 为 **16Tcy**。

位置计数频率 **Fpos** 与转子的 **RPM**、行计数 **D** 和 **QEI** 刷新模式 (**x2** 和 **x4**) 直接成正比，即：

$$F_{POS} = \frac{4D \cdot RPM}{60}$$

注： 位置编码器中递增行的数量通常设置为 **D=1024**，并且 **QEI** 刷新模式 =**x4**。

Fcy=10 MIPS 时，最大位置计数频率 (即 **4x QEI** 刷新模式，**D=1024**) 等同于 **2.5 MHz**，对应于 **625 kHz** 的 **Fqei**。

图 16-9 显示了当通过噪声滤波器采样时，**QEA** 和 **QEB** 正交输入时序。

PIC18F2331/2431/4331/4431

图 16-9: 当滤波器采样时 (分频比 = 1:1) 的 QEI 输入

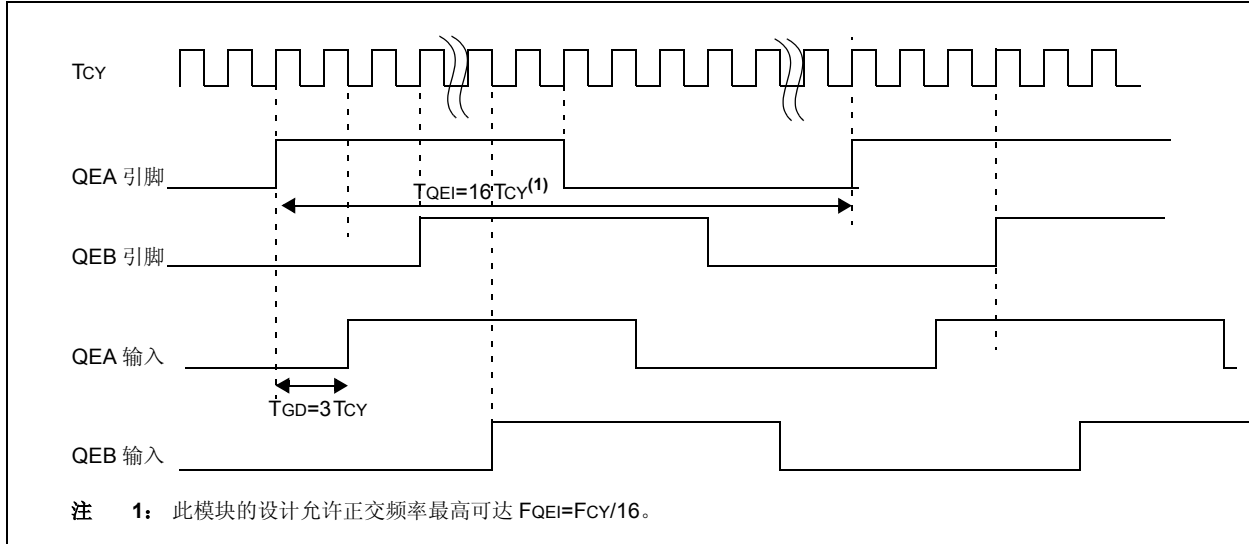


图 16-10: 周期匹配时的 QEI 模块复位时序

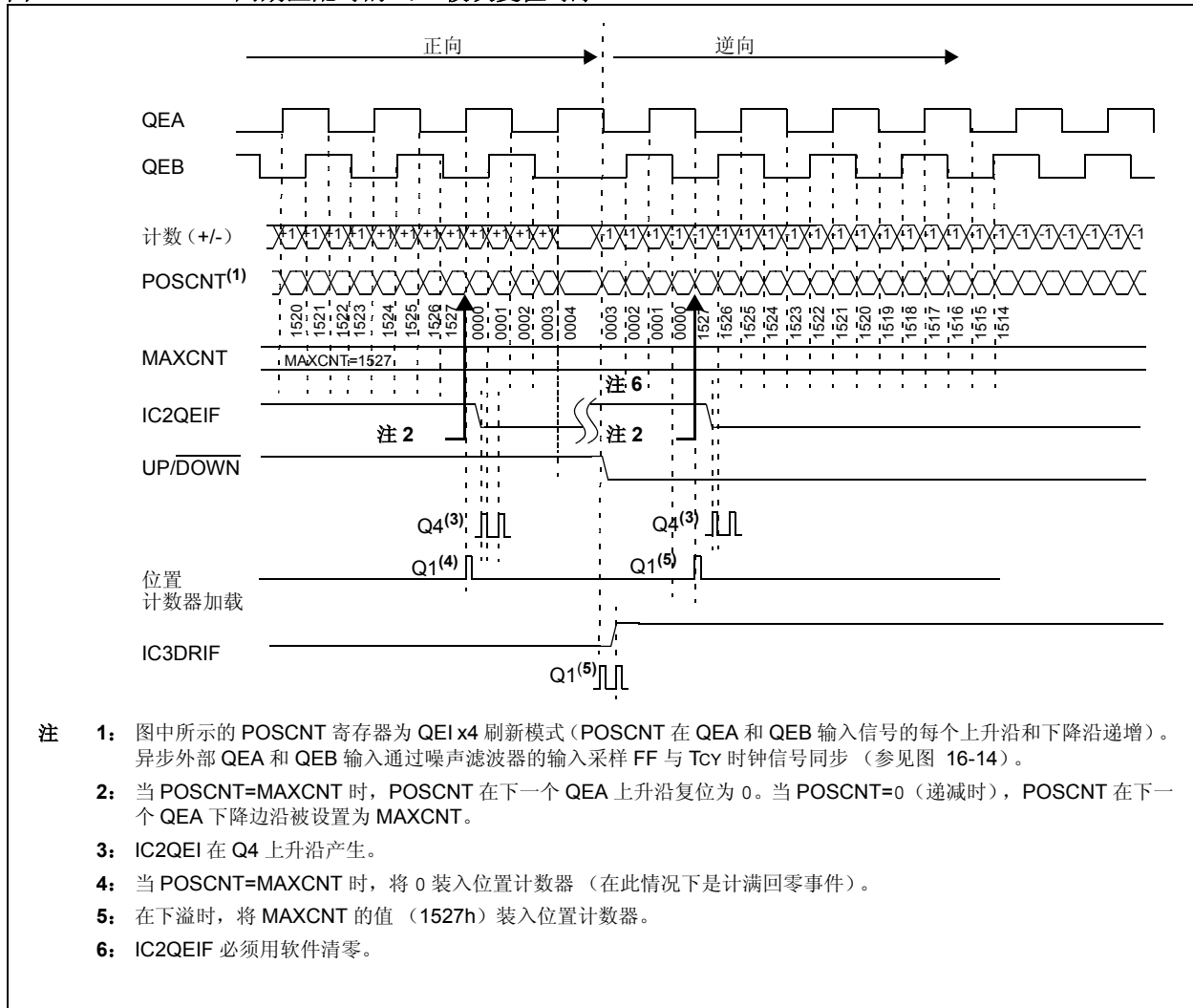
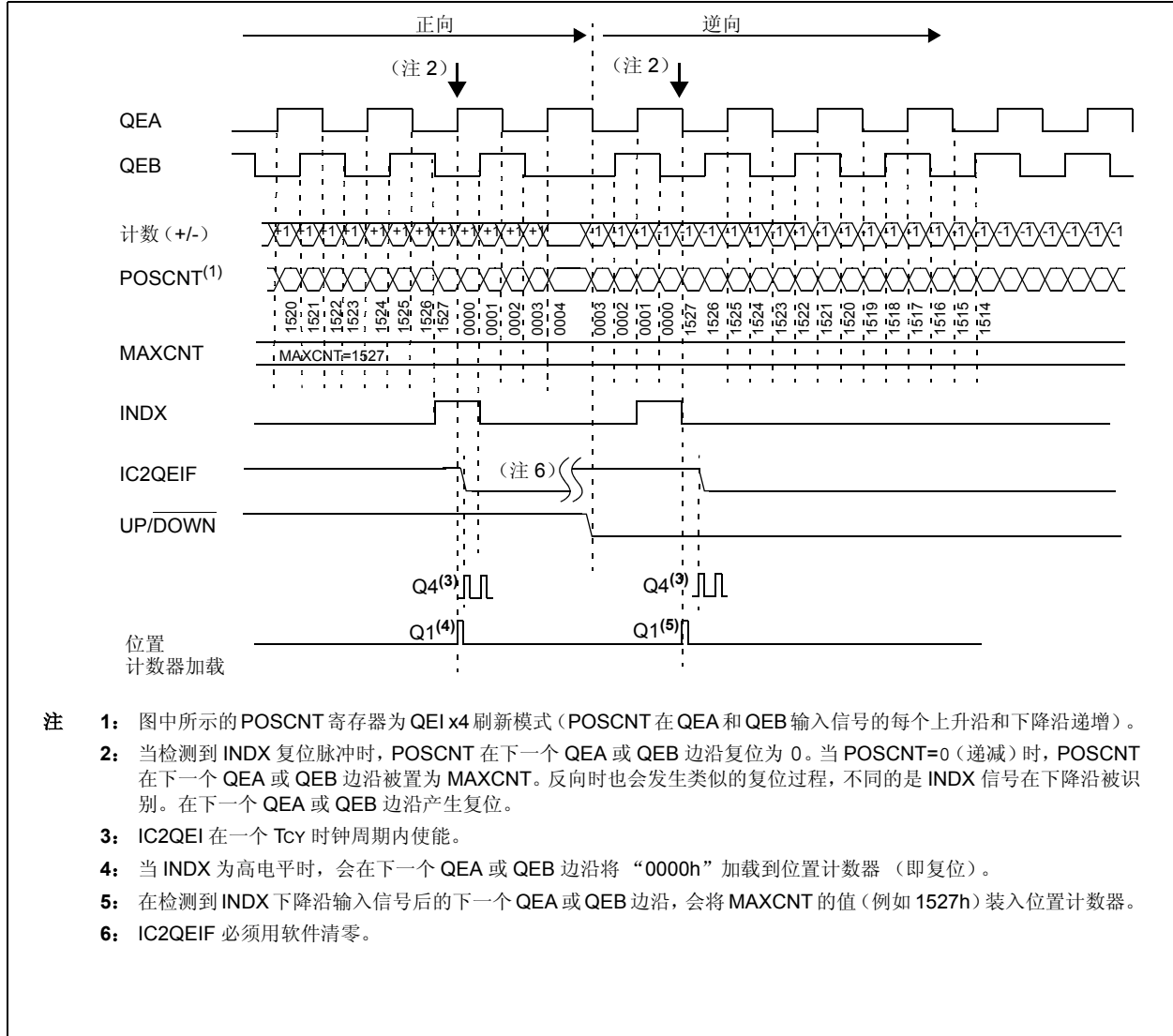


图 16-11: 带索引输入的 QEI 模块复位时序



16.2.6 速度测量

速度脉冲发生器配合 IC1 和同步 TMR5 (同步工作模式下) 共同为高速和低速运行的机械电机提供了一种高精度的速度测量方法。当 VELM 位为零 (=0) 且 QEI 置于某个工作模式 (参见表 16-6) 时, 使能速度模式。

要优化寄存器空间, 输入捕捉通道 1 (IC1) 用于捕捉 TMR5 计数器的值。输入捕捉缓冲寄存器 CAP1BUF 被重新定义为在速度测量模式 (VELM = 0) 下作为速度寄存器缓冲器 (VREGH, VREGL)。

表 16-6: 速度脉冲

QEIM<2:0>	速度事件模式
001	x2 速度事件模式。速度脉冲在每个 QEA 边沿产生。
010	
101	x4 速度事件模式。速度脉冲在每个 QEA 和 QEB 有效边沿产生。
110	

PIC18F2331/2431/4331/4431

16.2.6.1 速度事件时序

速度脉冲分频器按固定比例减少事件脉冲。分频器在速度事件频繁发生的高速测量应用中非常有用。通过为给定数量的输入事件脉冲产生单个输出脉冲，计数器在给定的时间间隔内可以记录更多的脉冲计数（即，经过的距离）。时间的测量是利用 TMR5 时基完成的。

每个速度脉冲都作为捕捉脉冲。当 TMR5 处于同步定时器模式时，在后分频器的每个输出脉冲捕捉 TMR5 的值。计数器随后复位为 0。TMR5 在发生捕捉事件时复位。

图 16-13 为速度测量时序图。

图 16-12: 速度测量框图

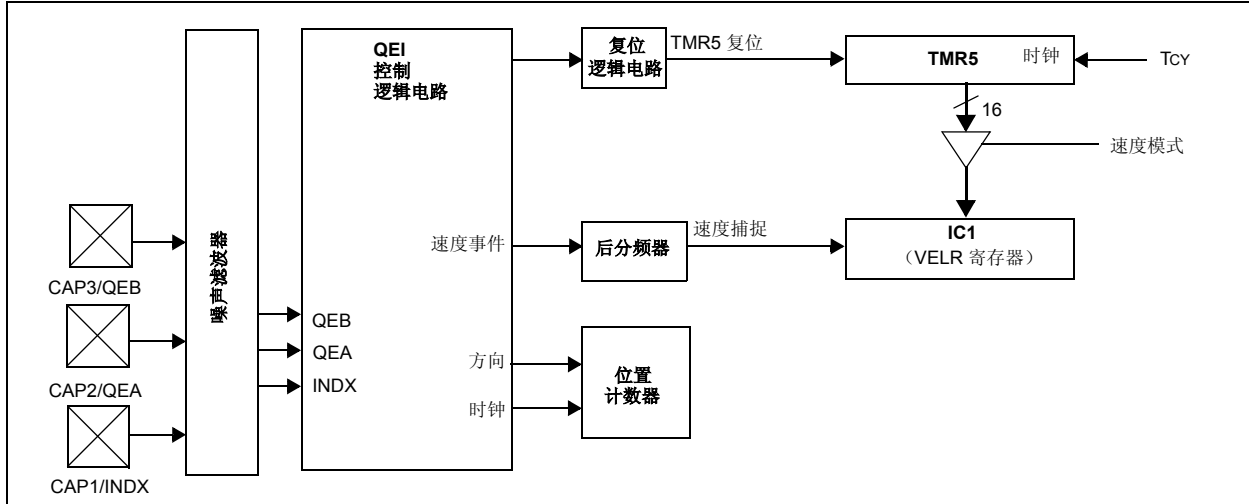
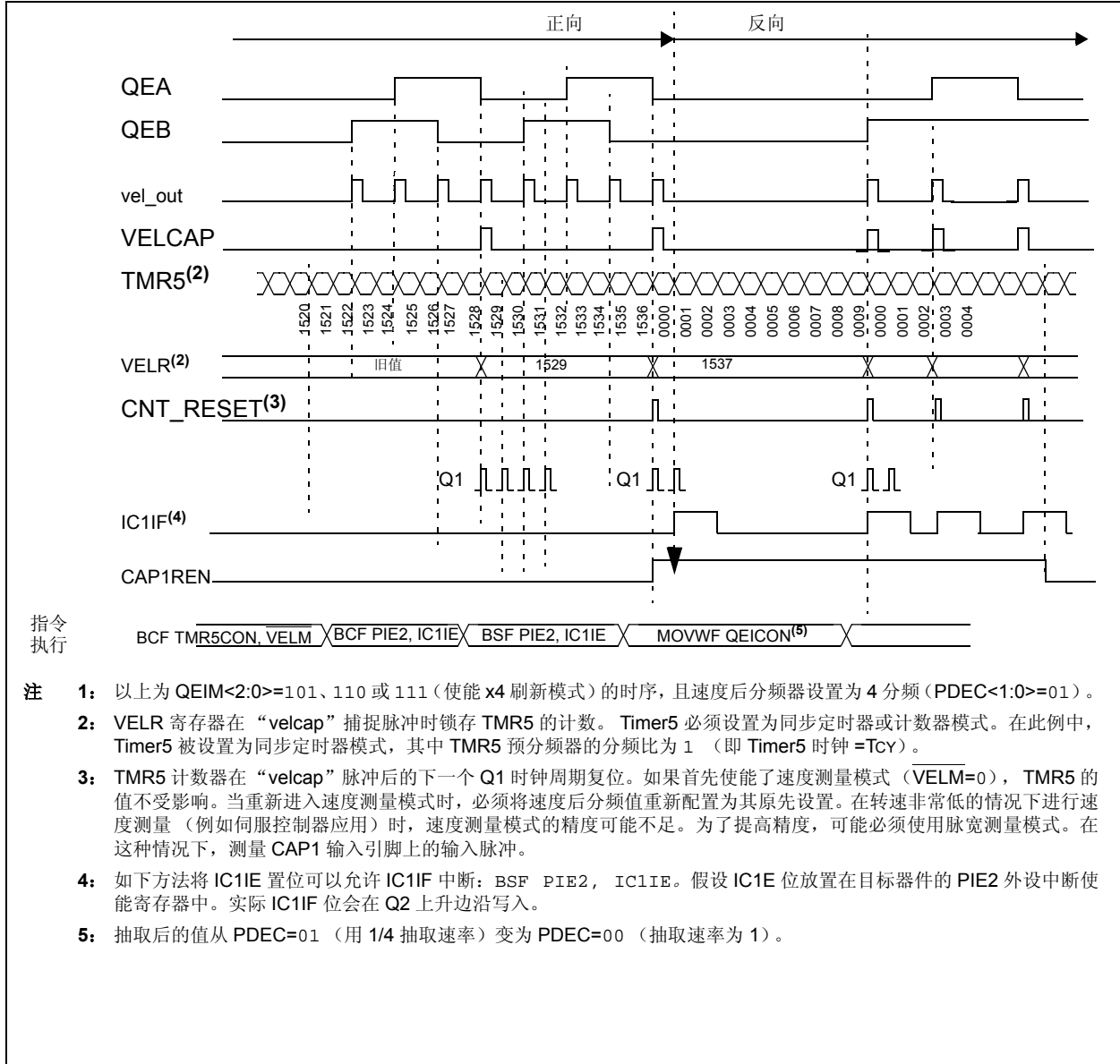


图 16-13: 速度测量时序 (1)



16.2.6.2 速度后分频器

速度事件脉冲 (velcap, 参见图 16-12) 在速度模式中充当 IC1 的 TMR5 捕捉触发器。速度后分频器先减少速度事件的数量, 然后将它们作为输入捕捉时钟。速度事件减少的比例可以通过 PDEC1:PDEC0 控制位 (QEICON<1:0>) 设置为 1:4、1:16、1:64 或不减少 (1:1) 实现。

当重新使能速度模式时, 速度后分频器设置自动重新加载为其原先的值。

16.2.6.3 速度模式下的 CAP1REN

通过将 CAP1REN 位 (CAP1CON<6>) 置位, 可以在发生速度事件捕捉时将 TMR5 值复位 (TMR5 寄存器对为 0000h)。当 CAP1REN 清零时, 任何事件捕捉脉冲都不会使 TMR5 时基复位。但是 VELR 寄存器对会继续使用当前的 TMR5 值刷新。

PIC18F2331/2431/4331/4431

16.3 噪声滤波器

运动反馈模块在 CAP1/INDX、CAP2/QEA 和 CAP3/QEB 上有 3 个噪声抑制滤波器。滤波电路还包括 T5CKI 引脚的第 4 个滤波器。这些滤波器的作用是帮助减少寄生噪声尖峰信号，这种信号可能会破坏输入引脚上的输入信号。滤波器确保在三个连续的采样时钟周期都获得同一个稳定值之后才允许改变输入信号。

所有这些滤波器都由数字滤波器控制 (DFLTCON) 寄存器控制 (参见寄存器 16-3)。通过将 DFLTCON 寄存器中相应的 FLTxEN 位置位或清零，可以单独使能或禁

止这些滤波器。可以通过 FLTCK2:FLTCK0 配置位对采样频率进行编程，确保全部 3 个噪声滤波器都具有相同的采样率。Tcy 作为时钟分频器电路的参考时钟源。

通过分别将相应的 FLTxEN 位置位或清零，可以在输入捕捉通道或 QE1 信道上添加或除掉噪声滤波器。每个捕捉通道都提供了单独的滤波器输出使能控制。FLT4EN 位可以使能或禁止 Timer5 模块中 TMR5CKI 输入引脚上的噪声滤波器。

在 POR 和 BOR 复位时 DFLTCON 寄存器清零，因此禁止所有通道上的滤波电路。滤波器的工作原理请见图 16-14 中的时序图。

寄存器 16-3: DFLTCON: 数字滤波器控制寄存器

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	FLT4EN	FLT3EN	FLT2EN	FLT1EN	FLTCK2	FLTCK1	FLTCK0
bit 7							bit 0

- bit 7 **未实现位:** 读作 0
- bit 6 **FLT4EN:** 噪声滤波器输出使能位, T5CKI 输入
1 = 使能
0 = 禁止
- bit 5 **FLT3EN:** 噪声滤波器输出使能位, CAP3/QEB 输入 ⁽¹⁾
1 = 使能
0 = 禁止
- bit 4 **FLT2EN:** 噪声滤波器输出使能位, CAP2/QEA 输入 ⁽¹⁾
1 = 使能
0 = 禁止
- bit 3 **FLT1EN:** 噪声滤波器输出使能位, CAP1/INDX 输入 ⁽¹⁾
1 = 使能
0 = 禁止
- bit 2-0 **FLTCK<2:0>:** 噪声滤波器时钟分频比位
111 = 未使用
110 = 1:128
101 = 1:64
100 = 1:32
011 = 1:16
010 = 1:4
001 = 1:2
000 = 1:1

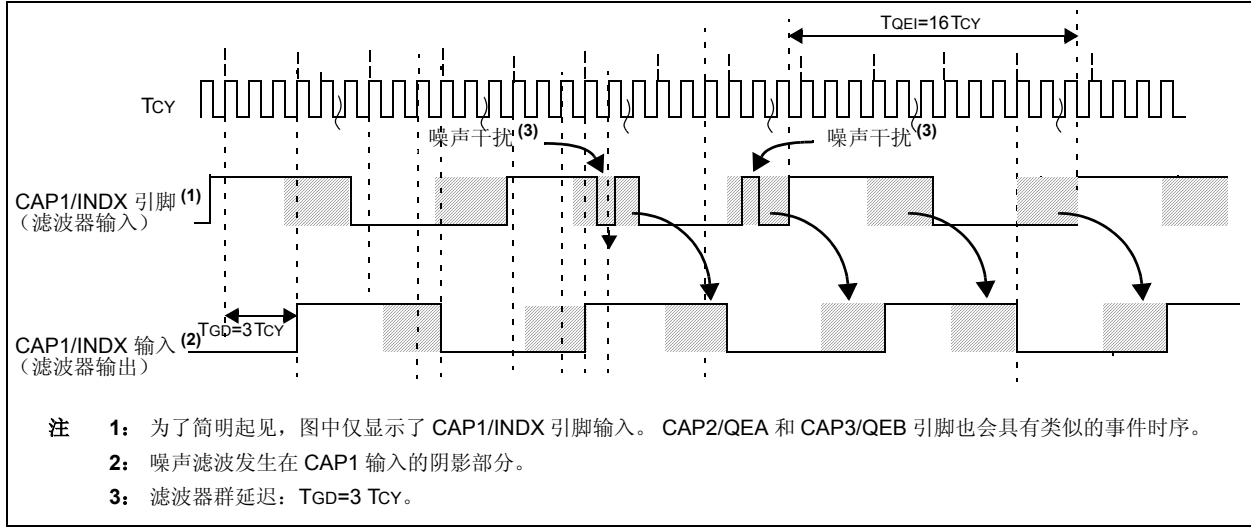
注 1: 在 QE1 和 IC 工作模式下都可用噪声滤波器输出使能。

图注:

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

注: 噪声滤波器主要用于随机高频滤波, 而非连续高频滤波。

图 16-14: 滤波器时序图 (时钟分频器 =1:1)



16.4 IC 和 QEI 共享中断

IC 和 QEI 子模块各自可以产生 3 个不同的中断信号; 但是它们共同使用 PIR3 寄存器中的 3 个中断标志位。任意给定时间特定中断标志位的含义取决于中断置位时哪个模块有效。表 16-7 概括了具体情况下的中断标志位的含义。

当 IC 子模块有效时, 3 个标志位 (IC1IF、IC2QEIF 和 IC3DRIF) 作为相应输入捕捉通道的捕捉中断事件标志。通道必须针对某个产生中断的事件进行配置 (如需更多信息, 请参见第 16.1.7 节 “IC 中断”)。

当使能 QEI 时, IC1IF 中断标志位表示速度测量事件 (通常是 VELR 寄存器刷新) 造成的中断。IC2QEIF 中断标志位表示发生了位置测量事件。IC3DRIF 表示检测到了方向改变。

表 16-7: IC 和 QEI 中断标志位的含义

中断标志	含义	
	IC 模式	QEI 模式
IC1IF	IC1 捕捉事件	速度寄存器刷新
IC2QEIF	IC2 捕捉事件	位置测量刷新
IC3DRIF	IC3 捕捉事件	方向改变

16.5 休眠模式下的操作

16.5.1 休眠模式下的 3X 输入捕捉

因为输入捕捉只能在其时基被配置为同步模式时才能进行, 所以休眠模式下输入捕捉不会捕捉任何事件。这是因为器件内部时钟已经停止, 并且同步模式的任何内部定时器都不会递增。预分频器将继续对事件计数 (非同步)。

当发生了指定捕捉事件时, CAPxIF 中断置位。捕捉缓冲寄存器会在器件从休眠中唤醒时刷新为当前 TMR5 的值。如果使能了 CAPxIF 中断, 器件将从休眠中被唤醒。这实际上使所有的输入捕捉通道都可以作为外部中断使用。

16.5.2 休眠模式下的 QEI

所有的 QEI 功能在休眠模式下都被暂停。

PIC18F2331/2431/4331/4431

表 16-8: 与运动反馈模块相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
IPR3	—	—	—	PTIP	IC3DRIP	IC2QEIP	IC1IP	TMR5IP	---1 1111	---1 1111
PIE3	—	—	—	PTIE	IC3DRIE	IC2QEIE	IC1IE	TMR5IE	---0 0000	---0 0000
PIR3	—	—	—	PTIF	IC3DRIF	IC2QEIF	IC1IF	TMR5IF	---0 0000	---0 0000
TMR5H	Timer5 寄存器高字节 (缓冲器)								xxxx xxxx	uuuu uuuu
TMR5L	Timer5 寄存器低字节								xxxx xxxx	uuuu uuuu
PR5H	Timer5 周期寄存器高字节								1111 1111	1111 1111
PR5L	Timer5 周期寄存器低字节								1111 1111	1111 1111
T5CON	T5SEN	RESEN	T5MOD	T5PS1	T5PS0	T5SYNC	TMR5CS	TMR5ON	0000 0000	0000 0000
CAP1BUFH/ VELRH	捕捉 1 寄存器, 高字节 / 速度寄存器, 高字节 (1)								xxxx xxxx	uuuu uuuu
CAP1BUFL/ VELRL	捕捉 1 寄存器, 低字节 / 速度寄存器, 低字节 (1)								xxxx xxxx	uuuu uuuu
CAP2BUFH/ POSCNTH	捕捉 2 寄存器, 高字节 / QEI 位置计数器寄存器, 高字节 (1)								xxxx xxxx	uuuu uuuu
CAP2BUFL/ POSCNTL	捕捉 2 寄存器, 低字节 / QEI 位置计数器寄存器, 低字节 (1)								xxxx xxxx	uuuu uuuu
CAP3BUFH/ MAXCNTH	捕捉 3 寄存器, 高字节 / QEI 最大计数限制寄存器, 高字节 (1)								xxxx xxxx	uuuu uuuu
CAP3BUFL/ MAXCNTL	捕捉 3 寄存器, 低字节 / QEI 最大计数限制寄存器, 低字节 (1)								xxxx xxxx	uuuu uuuu
CAP1CON	—	CAP1REN	—	—	CAP1M3	CAP1M2	CAP1M1	CAP1M0	-0-- 0000	-0-- 0000
CAP2CON	—	CAP2REN	—	—	CAP2M3	CAP2M2	CAP2M1	CAP2M0	-0-- 0000	-0-- 0000
CAP3CON	—	CAP3REN	—	—	CAP3M3	CAP3M2	CAP3M1	CAP3M0	-0-- 0000	-0-- 0000
DFLTCON	—	FLT4EN	FLT3EN	FLT2EN	FLT1EN	FLTCK2	FLTCK1	FLTCK0	-000 0000	-000 0000
QEICON	VELM	ERROR	UP/DOWN	QEIM2	QEIM1	QEIM0	PDEC1	PDEC0	0000 0000	0000 0000

图注: x= 未知, u= 不变, -= 未实现, q= 根据条件变化。
阴影单元格表示运动反馈模块未使用。

注 1: 寄存器名称和功能取决于所选择的子模块 (分别是 IC/QEI)。如需更多信息, 请参见第 16.1.10 节 “其他工作模式”。

17.0 功率控制 PWM 模块

在电机控制器和功率转换应用中，功率控制 PWM 模块简化了产生多种同步脉冲宽度调制（PWM）输出的过程。特别是，PWM 模块支持以下功率和动作控制应用：

- 三相和单相 AC 感应电机
- 开关磁阻电机
- 无刷直流（Brushless DC, BLDC）电机
- 不间断电源（UPS）
- 多种 DC 有刷电机

PWM 模块有如下特点：

- 多达八个的 PWM I/O 引脚，有四个占空比发生器。可以将引脚配对成完整的半桥控制。
- 分辨率高达 14 位，取决于 PWM 的周期。
- 动态 PWM 频率变化。
- 边沿或中心对齐输出模式。
- 单脉冲发生模式。
- 可在成对的 PWM 之间进行可编程死区控制。
- 中心对齐模式中不对称更新的中断支持。
- 电子换相电机（Electrically Commutated Motor, ECM）的输出改写，如 BLDC。
- 预定其他外设事件的特殊事件比较器。
- 在调试模式中，PWM 输出禁止功能会将 PWM 输出设置为无效状态。

在 PIC18F2X31 器件上，功率控制 PWM 模块支持三个 PWM 发生器和六个输出通道；在 PIC18F4X31 器件上，则支持四个发生器和八个输出通道。图 17-1 给出了该模块的简化框图。图 17-2 和图 17-3 给出了模块硬件如何将每个 PWM 输出对配置为互补和独立输出模式。

以下各节将会探讨 PWM 模块的每个功能单元。

PIC18F2331/2431/4331/4431

图 17-1: 功率控制 PWM 模块框图

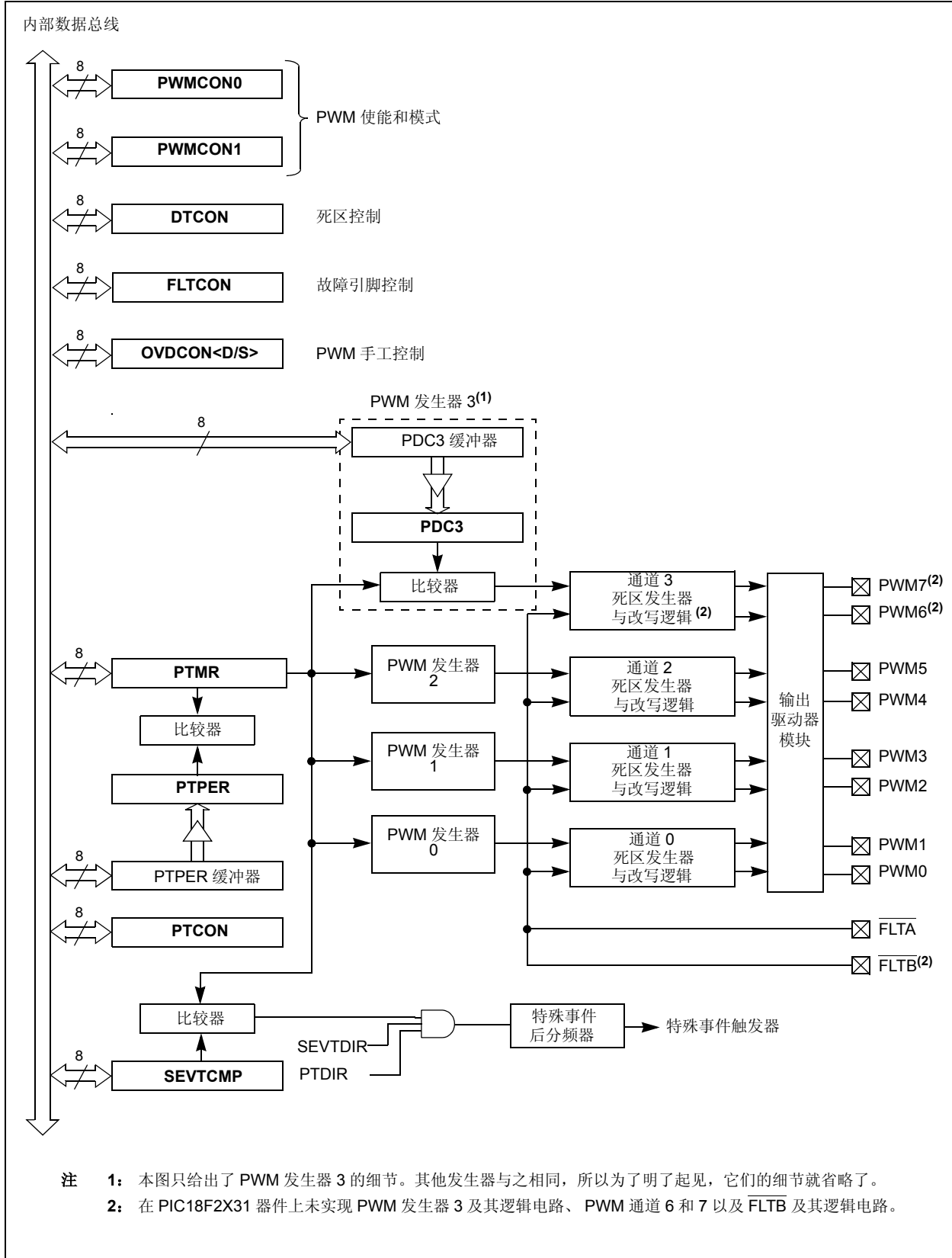


图 17-2: PWM 模块框图, 一个输出对, 互补模式

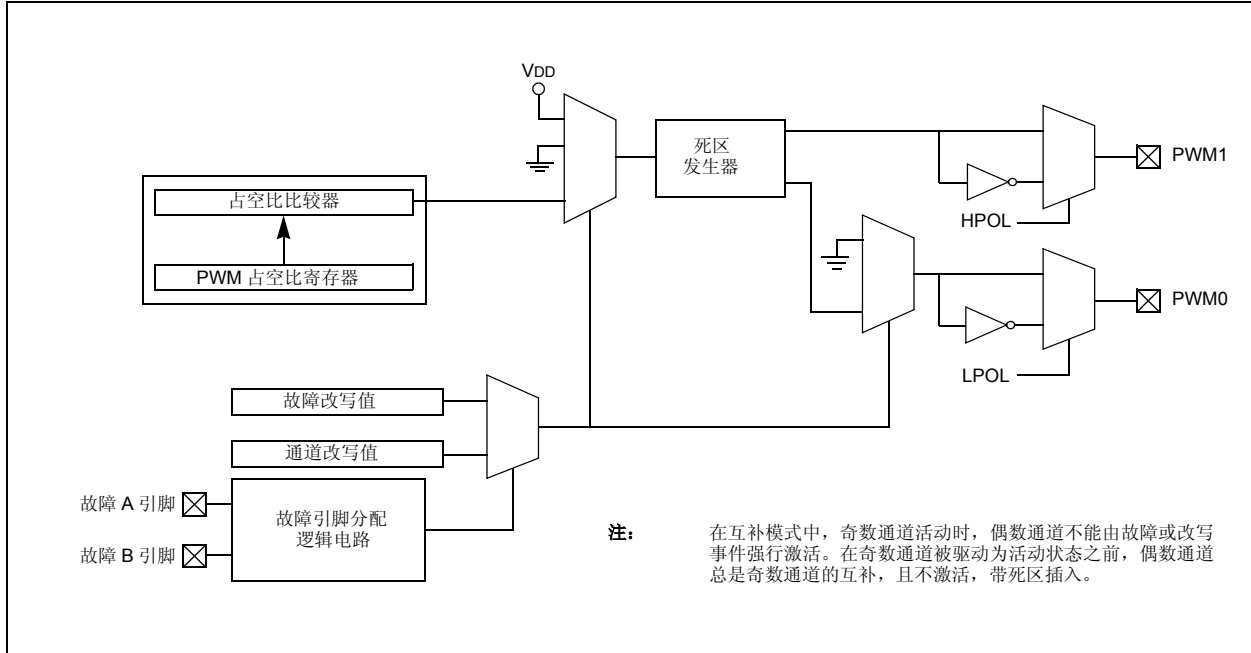
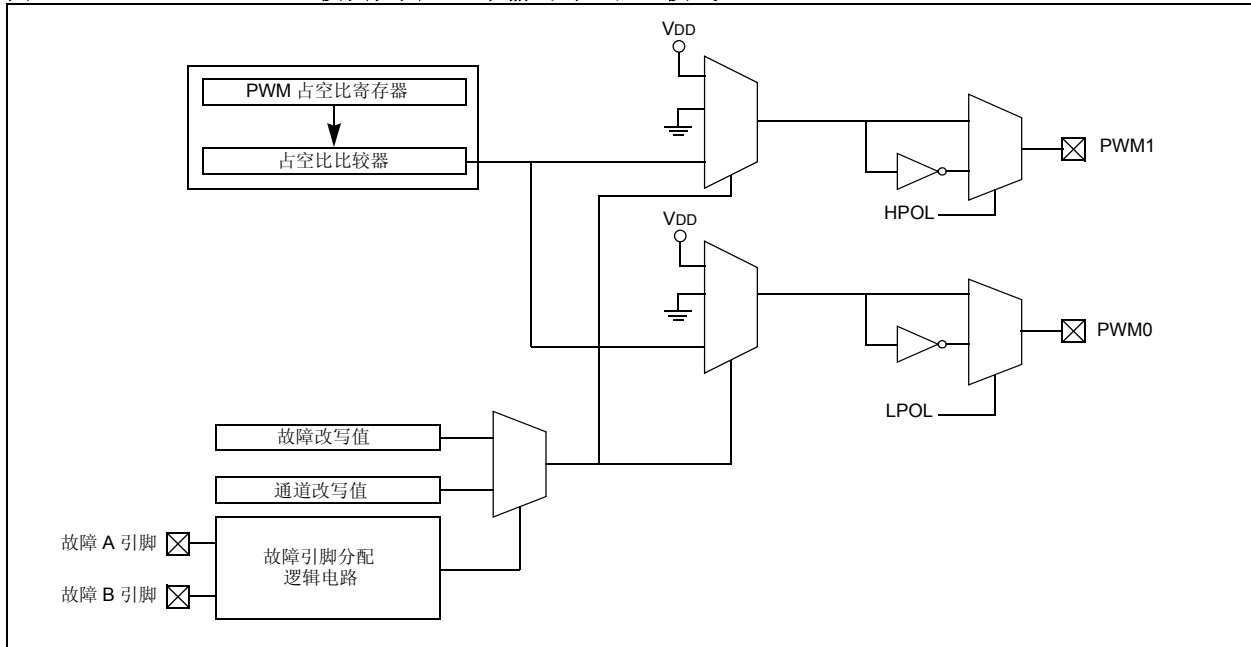


图 17-3: PWM 模块框图, 一个输出对, 独立模式



此模块包含四个占空比发生器, 编号为 0 到 3。八个 PWM 输出引脚, 编号为 0 到 7。这八个 PWM 输出引脚被分为由偶数和奇数输出引脚组成的输出引脚对。在互补模式中, 偶数 PWM 引脚必须总是作为相应奇数 PWM 引脚的互补。例如, PWM0 作为 PWM1 的互补, 而 PWM2 作为 PWM3 的互补, 等等。死区发生器在成对引脚中从一个引脚关闭到互补引脚开启之间插入一个“关闭时间”, 称作“死区”。这是为了防止对连接到 PWM 输出引脚的功率开关器件的破坏。

PWM 模块的时基由本身的 12 位定时器提供, 也可由可选预分频器和后分频器选项组成。

PIC18F2331/2431/4331/4431

17.1 控制寄存器

PWM 模块的操作总共由 22 个寄存器控制。其中有八个寄存器用于配置模块功能：

- PWM 定时器控制寄存器 0 (PTCON0)
- PWM 定时器控制寄存器 1 (PTCON1)
- PWM 控制寄存器 0 (PWMCON0)
- PWM 控制寄存器 1 (PWMCON1)
- 死区控制寄存器 (DTCON)
- 输出改写控制寄存器 (OVDCOND)
- 输出状态寄存器 (OVDCONS)
- 故障配置寄存器 (FLTCONFIG)

其他的 14 个寄存器被配置为七个 16 位的寄存器对。它们用于配置特殊功能值。它们分别是：

- PWM 时基寄存器 (PTMRH 和 PTMRL)
- PWM 周期寄存器 (PTPERH 和 PTPERL)
- PWM 特殊事件比较寄存器 (SEVTCMPH 和 SEVTCMPL)
- PWM 占空比 0 寄存器 (PDC0H 和 PDC0L)
- PWM 占空比 1 寄存器 (PDC1H 和 PDC1L)
- PWM 占空比 2 寄存器 (PDC2H 和 PDC2L)
- PWM 占空比 3 寄存器 (PDC3H 和 PDC3L)

所有这些寄存器对都是双缓冲的。

17.2 模块功能

PWM 模块支持几种工作模式，这些模式对特定功率和电机控制应用很有好处。在后续各节中将对每个工作模式进行描述。

PWM 模块由几个功能模块组成。每个模块的运行将分别在相关的工作模式中进行说明：

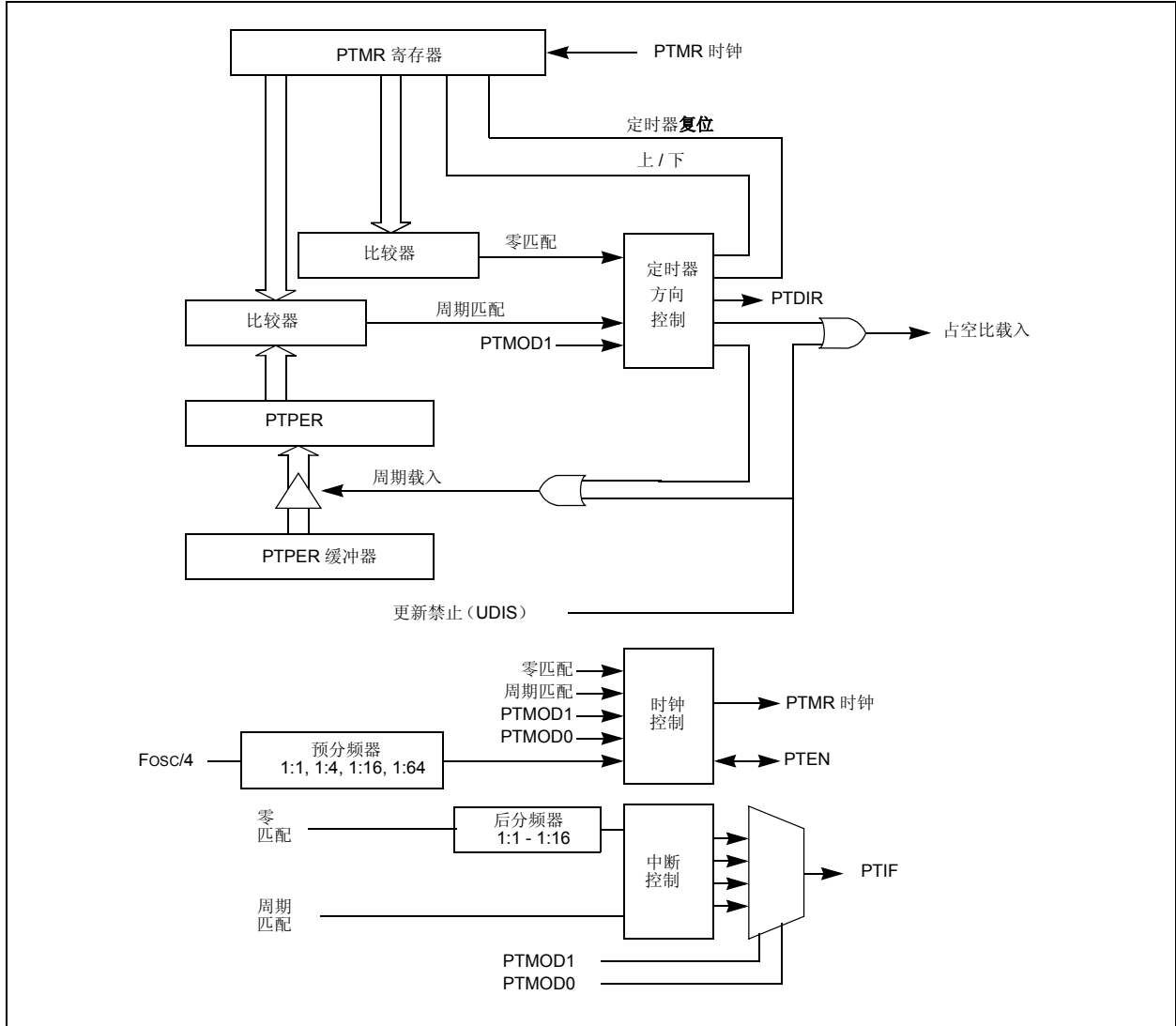
- PWM 时基
- PWM 时基中断
- PWM 周期
- PWM 占空比
- 死区发生器
- PWM 输出改写
- PWM 故障输入
- PWM 特殊事件触发器

17.3 PWM 时基

PWM 时基由带预分频器和后分频器功能的 12 位定时器提供。图 17-4 给出了 PWM 时基的简化框图。PWM 时基由 PTCON0 和 PTCON1 寄存器配置。通过置位或清零 PTCON1 寄存器中的 PTEN 位来实现时基使能或禁止。

注： 当 PTEN 位由软件清零时，PTMR 寄存器对 (PTMRL:PTMRH) 没有被清零。

图 17-4: PWM 时基框图



可以为以下四种不同的工作模式配置 PWM 时基：

- 自由运行模式
- 单步模式
- 连续上 / 下计数模式
- 带双更新中断的连续上 / 下计数模式

这四个模式由PTCON0寄存器中的PTMOD1:PTMOD0位选定。自由运行模式产生边沿对齐的PWM生成。上/下计数模式产生中心对齐的PWM生成。单步模式允许PWM模块支持某些电子换相电机（ECM）的脉冲控制并产生边沿对齐操作。

PIC18F2331/2431/4331/4431

寄存器 17-1: PTCON0: PWM 定时器控制寄存器 0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTOPS3	PTOPS2	PTOPS1	PTOPS0	PTCKPS1	PTCKPS0	PTMOD1	PTMOD0
bit 7						bit 0	

bit 7-4 **PTOPS3:PTOPS0:** PWM 时基输出后分频值选择位

0000 =1:1 后分频值
 0001 =1:2 后分频值
 .
 .
 1111 =1:16 后分频值

bit 3-2 **PTCKPS1:PTCKPS0:** PWM 时基输入时钟预分频值选择位

00 =PWM 时基输入时钟为 $F_{osc}/4$ (1:1 预分频值)
 01 =PWM 时基输入时钟为 $F_{osc}/16$ (1:4 预分频值)
 10 =PWM 时基输入时钟为 $F_{osc}/64$ (1:16 预分频值)
 11 =PWM 时基输入时钟为 $F_{osc}/256$ (1:64 预分频值)

bit 1-0 **PTMOD1:PTMOD0:** PWM 时基模式选择位

11 =PWM 时基工作在带双 PWM 更新中断的连续上/下模式下。
 10 =PWM 时基工作在连续上/下计数模式下。
 01 =PWM 时基配置为单步模式。
 00 =PWM 时基工作在自由运行模式下。

图注:

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

寄存器 17-2: PTCON1: PWM 定时器控制寄存器 1

R/W-0	R-0	U-0	U-0	U-0	U-0	U-0	U-0
PTEN	PTDIR	—	—	—	—	—	—
bit 7						bit 0	

bit 7 **PTEN:** PWM 时基定时器使能位

1 = PWM 时基启动
 0 = PWM 时基关闭

bit 6 **PTDIR:** PWM 时基计数方向状态位

1 = PWM 时基向下计数
 0 = PWM 时基向上计数

bit 5-0 **未实现位:** 读作 0。

图注:

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

PIC18F2331/2431/4331/4431

寄存器 17-3: PWMCON0: PWM 控制寄存器 0

U-0	R/W-1 ⁽¹⁾	R/W-1 ⁽¹⁾	R/W-1 ⁽¹⁾	R/W-0	R/W-0	R/W-0	R/W-0
—	PWMEN2	PWMEN1	PWMEN0	PMOD3 ⁽³⁾	PMOD2	PMOD1	PMOD0

bit 7 bit 0

bit 7 **未实现位:** 读作 0。

bit 6-4 **PWMEN2:PWMEN0: PWM 模块使能位⁽¹⁾**

111 =所有奇数 PWM I/O 引脚使能为 PWM 输出⁽²⁾。
 110 =PWM1 和 PWM3 引脚使能为 PWM 输出。
 101 =所有 PWM I/O 引脚使能为 PWM 输出⁽²⁾。
 100 =PWM0、PWM1、PWM2、PWM3、PWM4 和 PWM5 引脚使能为 PWM 输出。
 011 =PWM0、PWM1、PWM2 和 PWM3 I/O 引脚使能为 PWM 输出。
 010 =PWM0 和 PWM1 引脚使能为 PWM 输出。
 001 =PWM1 引脚使能为 PWM 输出。
 000 =PWM 模块禁止。所有 PWM I/O 引脚均为通用 I/O 引脚。

bit 3-0 **PMOD3:PMOD0: PWM 输出对模式位**

对于 PMOD0:

1 = PWM I/O 引脚对 (PWM0, PWM1) 在独立模式中。

0 = PWM I/O 引脚对 (PWM0, PWM1) 在互补模式中。

对于 PMOD1:

1 = PWM I/O 引脚对 (PWM2, PWM3) 在独立模式中。

0 = PWM I/O 引脚对 (PWM2, PWM3) 在互补模式中。

对于 PMOD2:

1 = PWM I/O 引脚对 (PWM4, PWM5) 在独立模式中。

0 = PWM I/O 引脚对 (PWM4, PWM5) 在互补模式中。

对于 PMOD3⁽³⁾:

1 = PWM I/O 引脚对 (PWM6, PWM7) 在独立模式中。

0 = PWM I/O 引脚对 (PWM6, PWM7) 在互补模式中。

注 1: PWMEN 位的复位条件取决于 PWMPIN 的器件配置位。

2: 当 PWMEN2:PWMEN0=101 时, 对于 PIC18F2X31 器件, PWM[5:0] 输出使能;
 对于 PIC18F4X31 器件, PWM[7:0] 输出使能。
 当 PWMEN2:PWMEN0=111 时, 对于 PIC18F2X31 器件, PWM 输出 1、3 和 5
 使能; 对于 PIC18F4X31 器件, PWM 输出 1、3、5 和 7 使能。

3: PIC18F2X31 器件中未实现, 保持这些位清零。

图注:

R= 可读位

W= 可写位

U= 未实现位, 读作 0

-n= 上电复位时的值

1= 置位

0= 清零

x= 未知

PIC18F2331/2431/4331/4431

寄存器 17-4: PWMCON1: PWM 控制寄存器 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
SEVOPS3	SEVOPS2	SEVOPS1	SEVOPS0	SEVTDIR	—	UDIS	OSYNC
bit 7					bit 0		

- bit 7-4 **SEVOPS3:SEVOPS0**: PWM 特殊事件触发器输出后分频值选择位
 0000 =1:1 后分频值
 0001 =1:2 后分频值
 .
 .
 .
 1111 =1:16 后分频值
- bit 3 **SEVTDIR**: 特殊事件触发器时基方向位
 1 = PWM 时基向下计数时会触发特殊事件。
 0 = PWM 时基向上计数时会触发特殊事件。
- bit 2 **未实现位**: 读作 0。
- bit 1 **UDIS**: PWM 更新禁止位
 1 = 禁止从占空比和周期缓冲寄存器更新。
 0 = 使能从占空比和周期缓冲寄存器更新。
- bit 0 **OSYNC**: PWM 输出改写同步位
 1 = 通过 OVDCON 寄存器的输出改写与 PWM 时基同步。
 0 = 通过 OVDCON 寄存器的输出改写与 PWM 时基异步。

图注:

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

17.3.1 自由运行模式

在自由运行模式中, PWM 时基 (PTMRL 和 PTMRH) 将开始向上计数, 直到与时基周期寄存器 PTPER (PTPERL 和 PTPERH) 中的值匹配。PTMR 寄存器将在接下来的输入时钟沿复位, 且只要 PTEN 位保持置位, 时基将继续向上计数。

17.3.2 单步模式

在单步模式下, 当 PTEN 位置位时, PWM 时基开始向上计数。当 PTMR 寄存器的值与 PTPER 寄存器匹配时, PTMR 寄存器将在接下来的输入时钟沿复位, 且 PTEN 位将由硬件清零, 停止时基。

17.3.3 连续上 / 下计数模式

在连续上 / 下计数模式下, PWM 时基向上计数直到 PTPER 寄存器中的值与 PTMR 匹配。在其后的输入时钟边沿, 定时器向下计数。PTCON1 寄存器中的 PTDIR 位是只读位, 用来表示计数方向。当定时器向下计数时, PTDIR 位置位。

注: 当 PWM 定时器在上 / 下计数模式使能时, 在上 / 下计数模式的第一个周期的前半周期中, PWM 输出保持停止。这样的话, PWM 引脚会因 PTMR 寄存器中的未知值而输出无意义的占空比。

17.3.4 PWM 时基预分频器

PTMR (Fosc/4) 的输入时钟的预分频值选项为 1:1、1:4、1:16 或 1:64。它们由 PTCON0 寄存器中的控制位 PTCKPS<1:0> 来选择。当发生以下任何一种情况时, 预分频器计数器清零:

- 写 PTMR 寄存器
- 写 PTCON (PTCON0 或 PTCON1) 寄存器
- 任何器件复位

注: 写 PTCON 时, PTMR 寄存器不会清零。

表 17-1 给出了 PWM 时基和预分频器可以产生的最小 PWM 频率。表中假设工作频率为 40 MHz (FcyC = 10 MHz) 且 PTPER = 0xFFFF。PWM 模块必须有能力为特定的功率控制应用产生行频 (50 Hz 或 60 Hz) 的 PWM 信号。

表 17-1: 最小 PWM 频率

F _{cyc} = 10 MIPS 且 PTPER = 0FFFh 时, 最小 PWM 频率与预分频值的关系		
预分频	边沿对齐 PWM 频率	中心对齐 PWM 频率
1:1	2441 Hz	1221 Hz
1:4	610 Hz	305 Hz
1:16	153 Hz	76 Hz
1:64	38 Hz	19 Hz

17.3.5 PWM 时基后分频器

PTMR 的匹配输出可以通过一个 4 位后分频值（包括 1:1 到 1:16 的分频）任意进行后分频产生中断。当发生以下任何一种情况时，后分频器计数器清零：

- 写 PTMR 寄存器
- 写 PTCON 寄存器
- 任何器件复位

写 PTCON 寄存器时，PTMR 寄存器不会清零。

17.4 PWM 时基中断

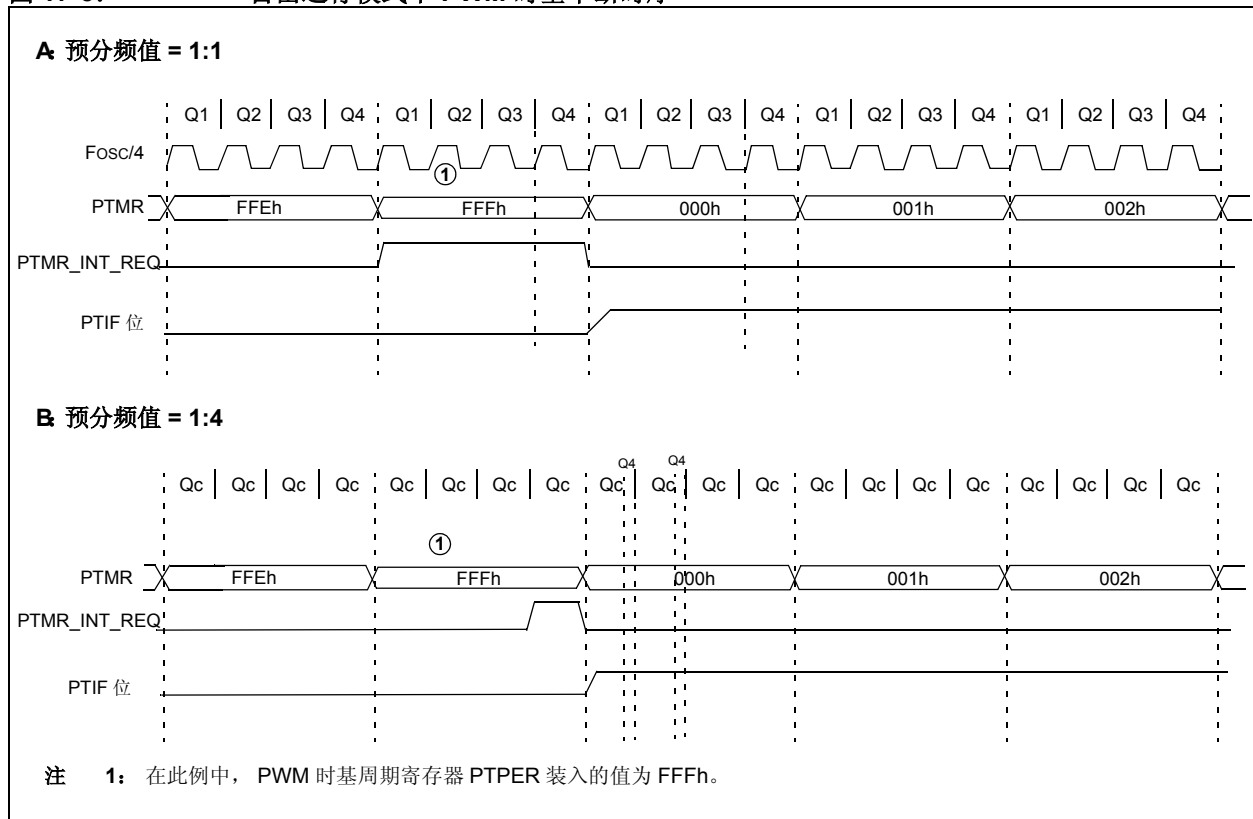
PWM 定时器可以根据 PTMOD<1:0> 位和后分频值位 (PTOPS<3:0>) 选择的工作模式产生中断。

17.4.1 自由运行模式下的中断

当 PWM 时基在自由运行模式 (PTMOD<1:0> = 00) 时，每当发生与 PTPER 寄存器的匹配时，即产生中断事件。PTMR 寄存器在接下来的时钟边沿复位到 0。

使用非 1:1 的后分频值选项，将会降低中断事件的发生频率。

图 17-5: 自由运行模式下 PWM 时基中断时序



PIC18F2331/2431/4331/4431

17.4.2 单步模式下的中断

当 PWM 时基在单步模式下 (PTMOD<1:0> = 01) 时, 与 PTPER 寄存器匹配发生时, 即产生中断事件。在接下来的输入时钟沿, PWM 定时器寄存器 (PTMR) 复位到零且 PTEN 位清零。此定时器模式不会影响后分频值选择位。

17.4.3 连续上 / 下计数模式下的中断

在上 / 下计数模式 (PTMOD<1:0> = 10) 中, 当 PTMR 寄存器的值变为零, 且 PWM 时基开始向上计数时产生中断事件。在此模式中使用后分频值选择位, 可以降低中断事件的发生频率。图 17-7 给出了连续上 / 下计数模式下的中断。

图 17-6: 单步模式下 PWM 时基中断时序

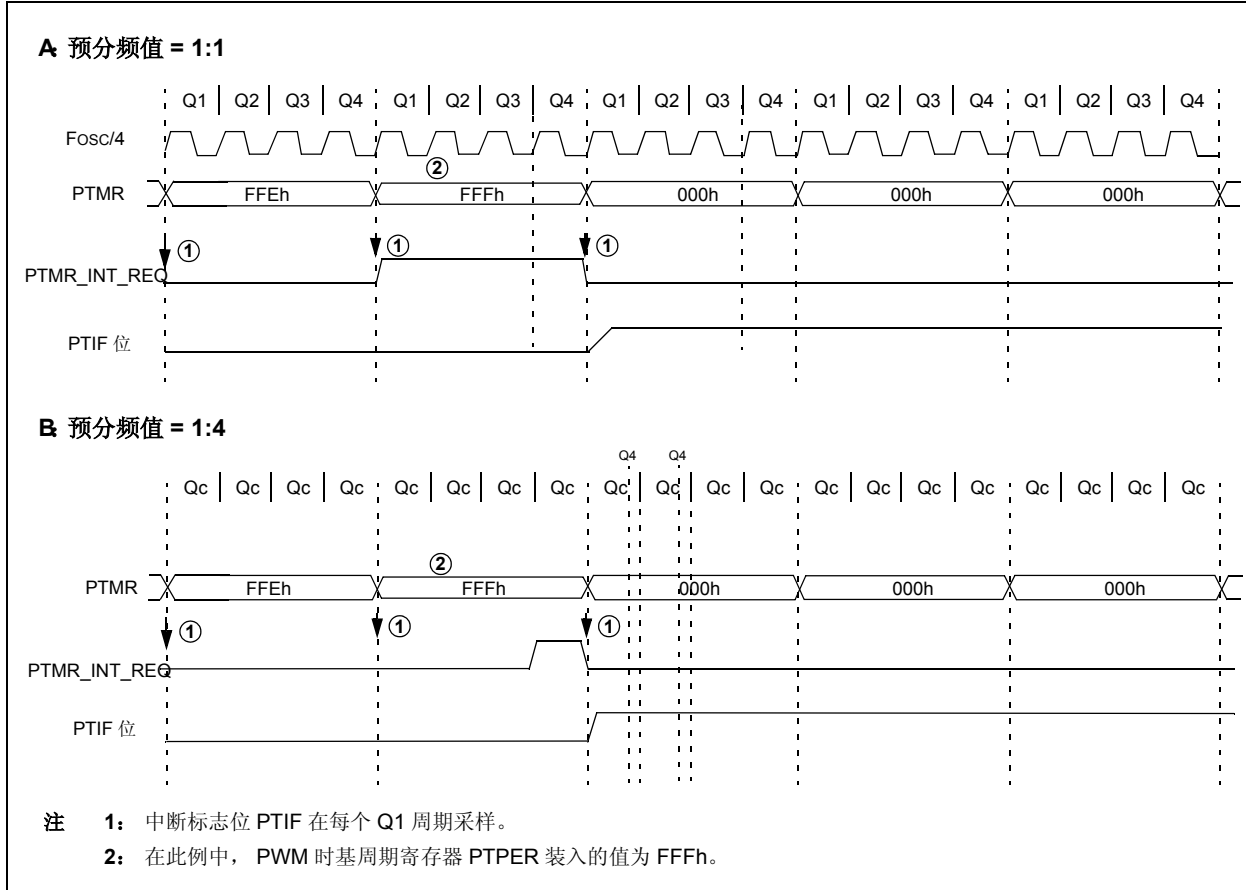
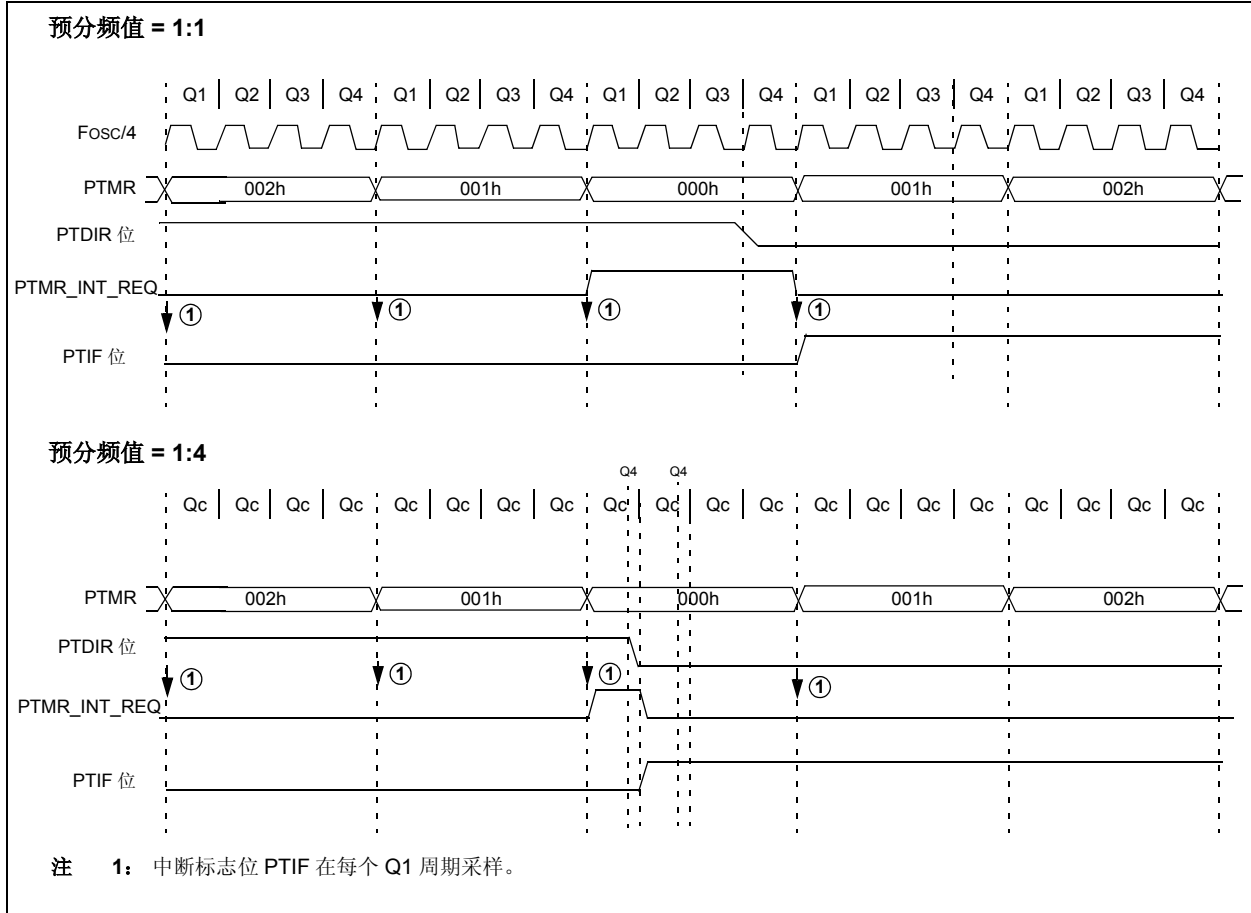


图 17-7: 上/下计数模式下 PWM 时基中断



PIC18F2331/2431/4331/4431

17.4.4 双更新模式下的中断

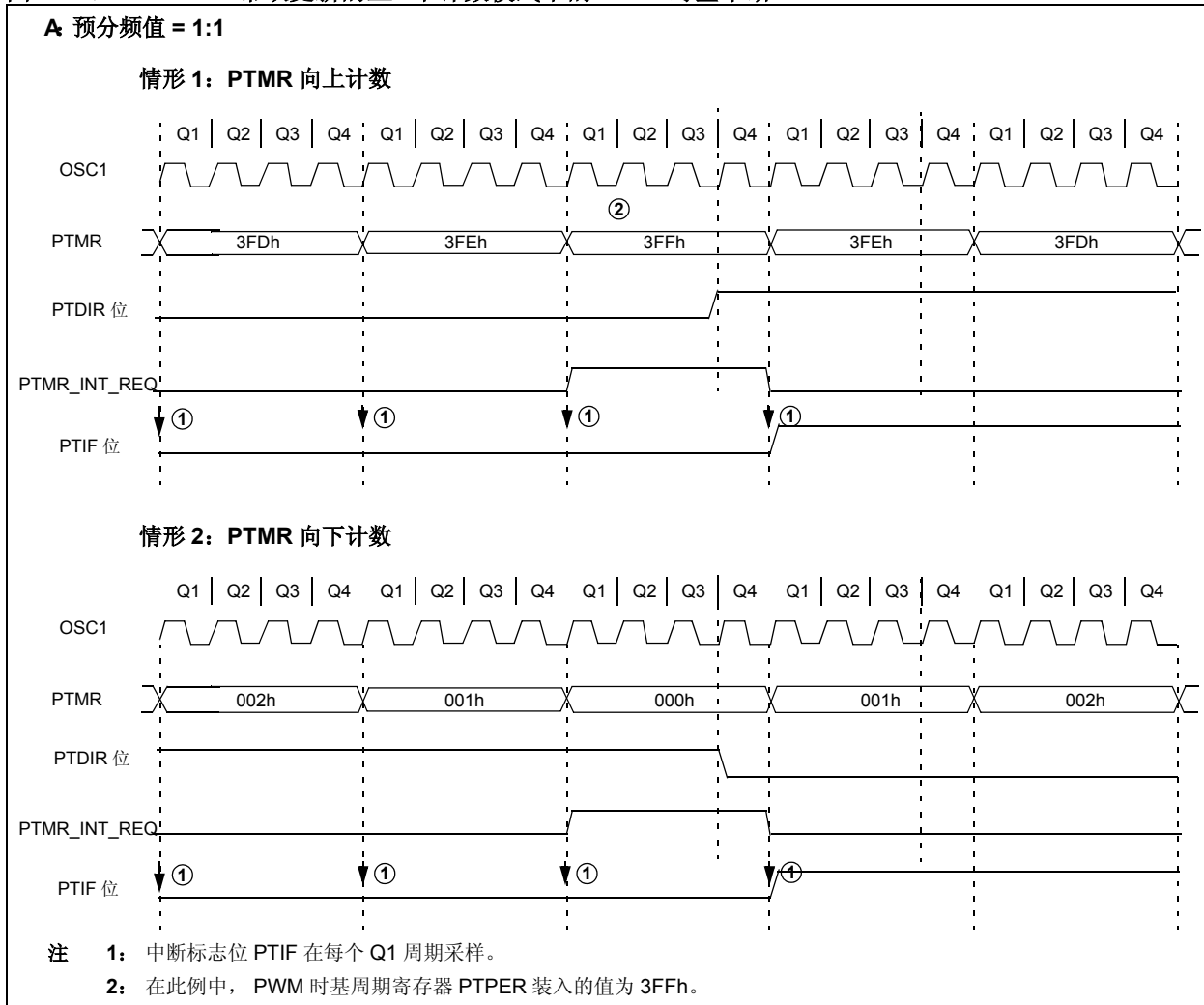
此模式在上 / 下计数模式中有效。在双更新模式 (PTMOD<1:0> = 11) 中, 当 PTMR 寄存器等于零及 PTMR 与 PTPER 寄存器相匹配时, 即产生中断事件。图 17-8 给出了带双更新的上 / 下计数模式下的中断。

双更新模式为中心对齐模式的用户提供两种附加功能。

1. 由于 PWM 占空比在每个周期更新两次, 所以控制环带宽加倍。
2. 可以产生不对称的中心对齐 PWM 波形, 它对于在特定电机控制应用中尽可能地减小输出波形失真很有用处。

注: 当 PTEN 有效时不要改变 PTMOD。它会产生意料之外的值。要改变 PWM 定时器的的工作模式, 首先清零 PTEN 位, 将必需的数据载入 PTMOD, 然后再将 PTEN 位置位。

图 17-8: 带双更新的上 / 下计数模式下的 PWM 时基中断



17.5 PWM 周期

PWM 周期由 PTPER 寄存器对 (PTPERL 和 PTPERH) 定义。PWM 周期有 12 位的分辨率，由 PTPERH 的 4 个 LSB 和 PTPERL 的 8 位构成。PTPER 是双缓冲寄存器，用于设置 PWM 时基计数周期。

在以下情况下，PTPER 缓冲器的内容会装入 PTPER 寄存器：

- 自由运行和单步模式：与 PTPER 寄存器匹配之后，PTMR 寄存器复位到零。
- 上/下计数模式：当 PTMR 寄存器为零。当 PWM 时基禁止 (PTEN = 0) 时，PTPER 缓冲器中的值会自动装入 PTPER 寄存器。图 17-9 和图 17-10 给出了 PTPER 缓冲器中的内容装入真正的 PTPER 寄存器的时刻。

以下公式用来计算 PWM 周期。

公式 17-1: 自由运行模式的 PWM 周期

$$TPWM = \frac{(PTPER + 1)}{Fosc / (PTMRPS / 4)}$$

或

$$TPWM = \frac{(PTPER + 1) \times PTMRPS}{Fosc / 4}$$

公式 17-2: 上/下计数模式的 PWM 周期

$$TPWM = \frac{(2 \times PTPER)}{Fosc / (PTMRPS / 4)}$$

PWM 频率是周期的倒数，即

$$PWM \text{ 频率} = \frac{1}{TPWM}$$

给定器件振荡器和 PWM 频率时，最大分辨率（以位为单位）可以由以下公式确定：

公式 17-3: PWM 分辨率

$$\text{分辨率} = \frac{\log\left(\frac{Fosc/4}{Fpwm}\right)}{\log(2)}$$

表 17-2 给出了选定执行速度和 PTPER 值时，PWM 分辨率和频率。表 17-2 中的 PWM 频率是针对边沿对齐的 PWM 模式而计算的。对于中心对齐模式，PWM 频率大约为表中所示值的一半。

表 17-2: PWM 频率和分辨率示例

PWM 频率 = 1/TPWM				
Fosc	MIPS	PTPER 值	PWM 分辨率	PWM 频率
40 MHz	10	0FFFh	14 位	2.4 kHz
40 MHz	10	07FFh	13 位	4.9 kHz
40 MHz	10	03FFh	12 位	9.8 kHz
40 MHz	10	01FFh	11 位	19.5 kHz
40 MHz	10	FFh	10 位	39.0 kHz
40 MHz	10	7Fh	9 位	78.1 kHz
40 MHz	10	3Fh	8 位	156.2 kHz
40 MHz	10	1Fh	7 位	312.5 kHz
40 MHz	10	0Fh	6 位	625 kHz
25 MHz	6.25	0FFFh	14 位	1.5 kHz
25 MHz	6.25	03FFh	12 位	6.1 kHz
25 MHz	6.25	FFh	10 位	24.4 kHz
10 MHz	2.5	0FFFh	14 位	610 Hz
10 MHz	2.5	03FFh	12 位	2.4 kHz
10 MHz	2.5	FFh	10 位	9.8 kHz
5 MHz	1.25	0FFFh	14 位	305 Hz
5 MHz	1.25	03FFh	12 位	1.2 kHz
5 MHz	1.25	FFh	10 位	4.9 kHz
4 MHz	1	0FFFh	14 位	244 Hz
4 MHz	1	03FFh	12 位	976 Hz
4 MHz	1	FFh	10 位	3.9 kHz

注： 对于中心对齐操作，PWM 频率大约为表中所示值的 1/2。

PIC18F2331/2431/4331/4431

图 17-9: 自由运行计数模式下的 PWM 周期缓冲器更新

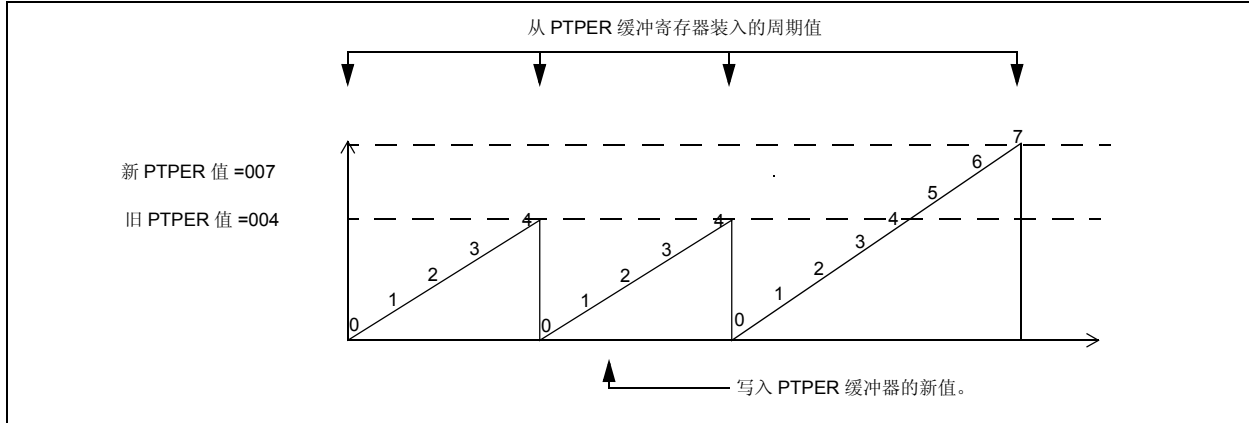
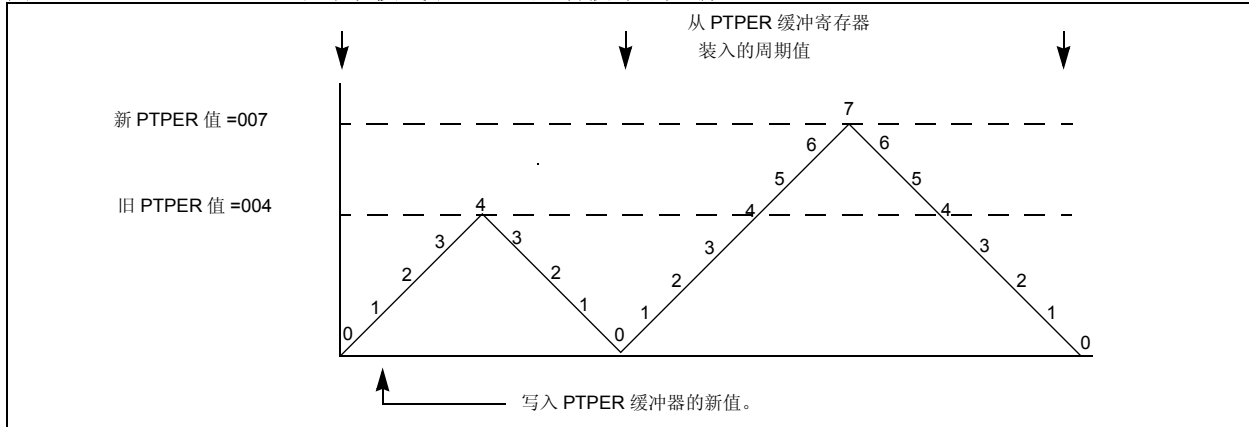


图 17-10: 上/下计数模式下 PWM 周期缓冲器更新



17.6 PWM 占空比

PWM 占空比由 PDCx (PDCxL 和 PDCxH) 寄存器定义。共有 4 个 PWM 占空比寄存器，对应 4 对 PWM 通道。PDCxH 的 6 个 LSB 和 PDCxL 的 8 位构成了 14 位分辨率的占空比寄存器。PDCx 是双缓冲寄存器，用于设置 PWM 时基计数周期。

17.6.1 PWM 占空比寄存器

四个 14 位特殊功能寄存器用于指定 PWM 模块的占空比值：

- PDC0 (PDC0L 和 PDC0H)
- PDC1 (PDC1L 和 PDC1H)
- PDC2 (PDC2L 和 PDC2H)
-CLOCKPS = 00 时) 考虑到占空比匹配。

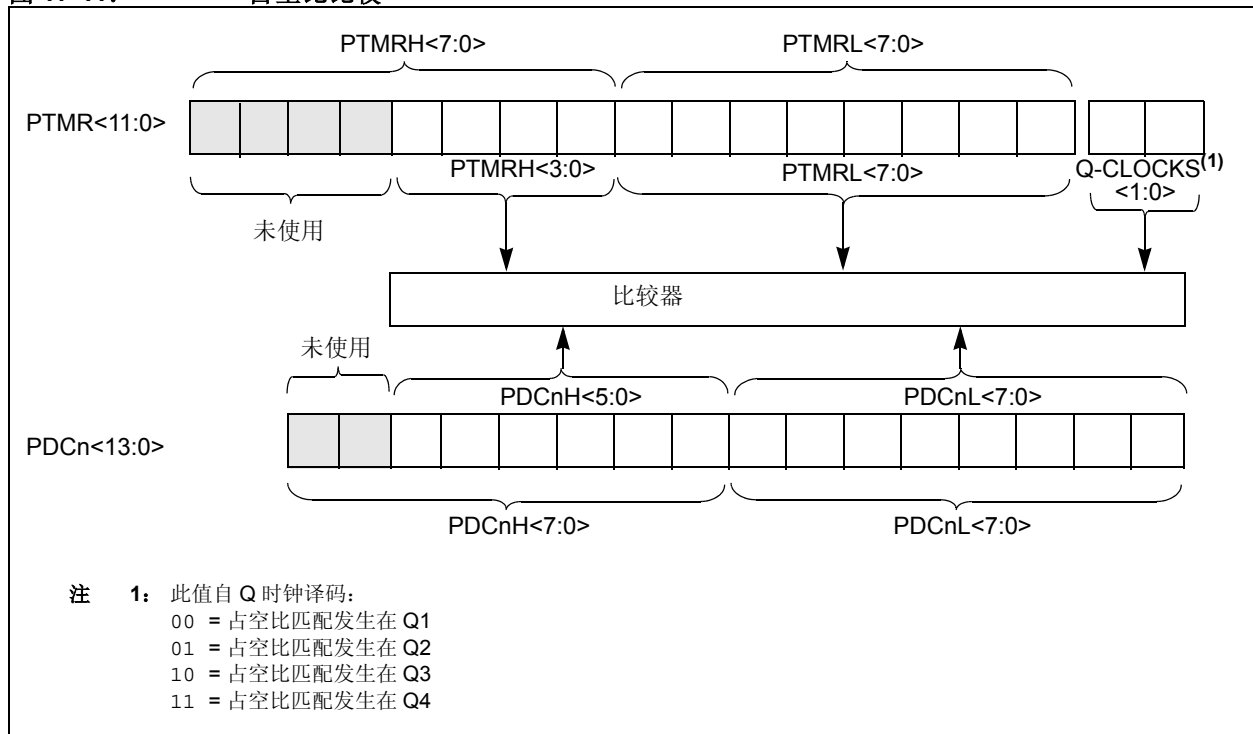
每个占空比寄存器中的值确定 PWM 输出处于有效状态的时间。PDCn 的高 12 位保持了 PTMRH/L<11:0> 中真正的占空比值，而低 2 位控制占空比匹配在哪个内部 Q 时钟上发生。如图 17-11 所示，这 2 位值自 Q 时钟译码 (当预分频值为 1:1 时 (PTCKPS = 00))。

在边沿对齐模式中，PWM 周期从 Q1 开始，之后当占空比寄存器与 PTMR 寄存器匹配时结束。当 PDC 的高 12 位等于 PTMR，且低 2 位等于 Q1、Q2、Q3 或 Q4 时，需要根据 PDC 的低两位 (当预分频值为 1:1 或

注： 当预分频值不是 1:1 时 (PTCKPS ≠ ~00)，PTMR 与 PDC 匹配时，在指令周期的 Q1 时钟发生占空比匹配。

每个比较单元都有允许改写 PWM 信号的逻辑。此逻辑也确保 PWM 信号在互补模式中互相补充 (带死区插入) (参阅第 17.7 节 “死区发生器”)。

图 17-11: 占空比比较



PIC18F2331/2431/4331/4431

17.6.2 占空比寄存器缓冲器

四个 PWM 占空比寄存器是双缓冲，允许 PWM 输出无故障更新。对于每个占空比电路来说，用户可以访问一个的占空比缓冲寄存器和另一个占空比寄存器，它用来保持当前 PWM 周期中使用的真实比较值。

在边沿对齐的 PWM 输出模式中，只要发生 PTMR 和 PTPER 寄存器匹配且 PTMR 复位（如图 17-12 中所示），新的占空比值就会被更新。另外，当 PWM 时基被禁止（PTEN = 0）时，占空比缓冲器的内容会自动装入占空比寄存器。

如果 PWM 时基为上/下计数模式，当 PTMR 寄存器值为零且 PWM 时基开始向上计数时，新占空比值将被更新。当 PWM 时基被禁止（PTEN = 0）时，占空比缓冲器的内容被自动装入占空比寄存器。图 17-13 给出了上/下计数模式下发生占空比更新的时序。在此模式中，修改生效之前，最多整个 PWM 周期都可用于计算和装入新的 PWM 占空比。

如果 PWM 时基为双更新上/下计数模式，当 PTMR 寄存器的值为零和 PTMR 寄存器的值与 PTPER 寄存器中的值相匹配时，将更新新的占空比值。在上述两个条件发生时，占空比缓冲器的内容被自动装入占空比寄存器。图 17-14 给出了双更新的上/下计数模式下的占空比更新。在此模式中，修改生效之前，最多只有半个 PWM 周期可用于计算和装入新的 PWM 占空比。

17.6.3 边沿对齐的 PWM

当 PWM 时基为自由运行模式或单步模式时，由此模块产生边沿对齐的 PWM 信号。对于边沿对齐的 PWM 输出，给定 PWM 通道输出的周期由 PTPER 中的值指定，其占空比则由相应占空比寄存器指定（参见图 17-12）。在周期开始时（PTMR = 0），PWM 输出驱动为有效，而当占空比寄存器中的值与 PTMR 匹配时，驱动为无效。如 PWM 周期部分所述，当 PTMR 与 PTPER 匹配时，将开始新的周期。

如果特定占空比寄存器中的值为零，相应 PWM 引脚上的输出对整个 PWM 周期无效。此外，如果占空比寄存器的值大于 PTPER 寄存器中的值，PWM 引脚上的输出对整个 PWM 周期有效。

图 17-12: 边沿对齐的 PWM

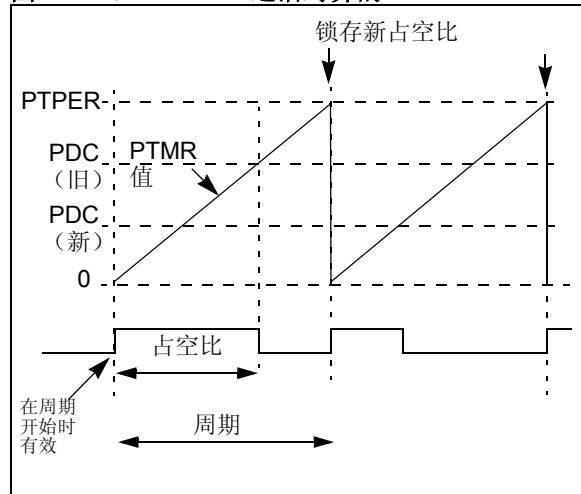


图 17-13: 上/下计数模式中占空比的更新时序

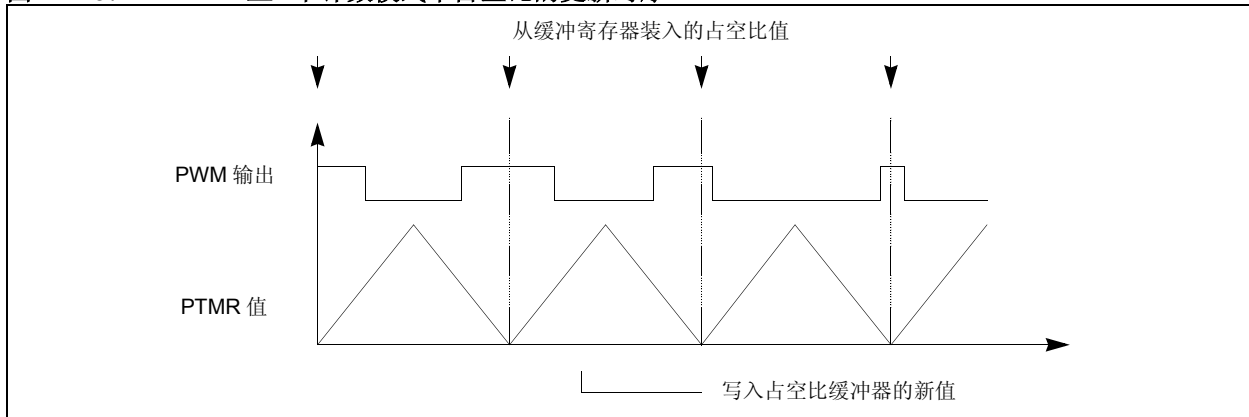
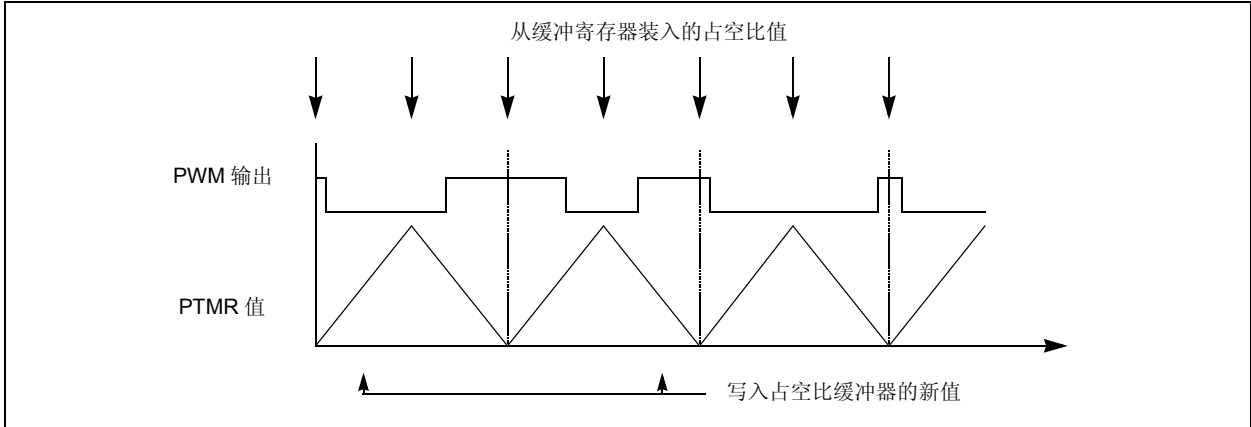


图 17-14: 双更新的上/下计数模式中占空比的更新时序



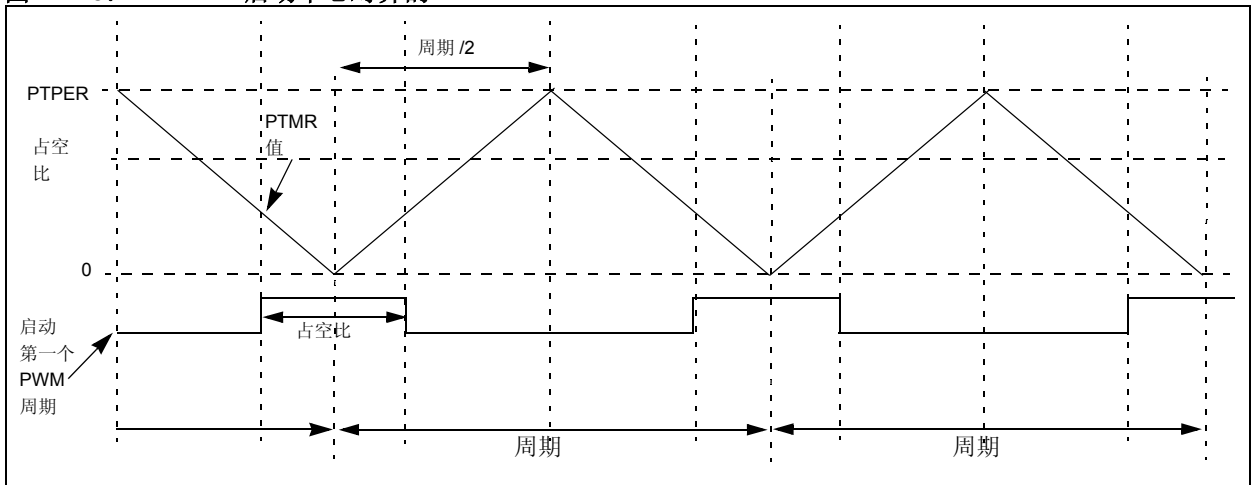
17.6.4 中心对齐 PWM

当 PWM 时基在上/下计数模式中进行配置时，此模块产生中心对齐的 PWM 信号（参见图 17-15）。当占空比寄存器的值与 PTMR 的值匹配且 PWM 时基向下计数时（PTDIR = 1），PWM 比较输出被驱动为有效状态。当 PWM 时基向上计数（PTDIR = 0），且 PTMR 寄存器中的值与占空比值匹配时，PWM 比较输出被驱动为无效状态。如果特定占空比寄存器中的值为零，相应

PWM 引脚上的输出将对整个 PWM 周期无效。此外，如果占空比寄存器的值大于或等于 PTPER 寄存器的值，PWM 引脚上的输出将对整个 PWM 周期有效。

注： 当 PWM 在中心对齐模式启动时，周期寄存器（PTPER）的值被装入 PWM 定时器寄存器（PTMR）且 PTMR 被自动配置以开始向下计数。这样可以确保所有 PWM 信号不会同时启动。

图 17-15: 启动中心对齐的 PWM



PIC18F2331/2431/4331/4431

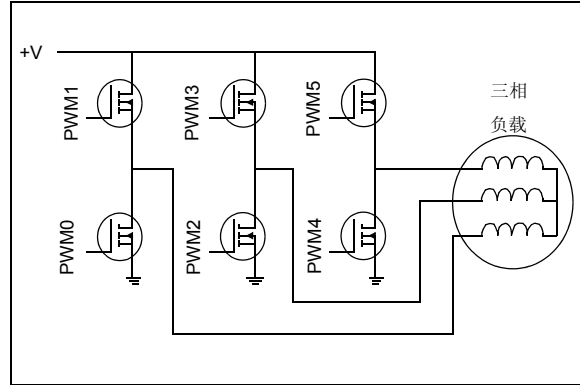
17.6.5 互补 PWM 运行

如图 17-16 所示，PWM 的互补运行模式对于在半桥配置中驱动一个或多个功率开关很有用处。此逆变器拓扑常见于三相感应电机、无刷 DC 电机或三相不间断电源（UPS）的控制应用中。每个上端 / 下端功率开关对都由一个互补的 PWM 信号源提供信号。在器件开关时，也可以选择插入死区（参见第 17.7 节“死区发生器”），此时两个输出均暂时失效。在互补模式中，PWM 输出的占空比比较单元分配如下：

- PDC0 寄存器控制 PWM1/PWM0 输出
- PDC1 寄存器控制 PWM3/PWM2 输出
- PDC2 寄存器控制 PWM5/PWM4 输出
- PDC3 寄存器控制 PWM7/PWM6 输出

PWM1/3/5/7 是主 PWM，由 PDC 寄存器控制，PWM0/2/4/6 则是互补输出。当使用 PWM 控制半桥时，奇数编号的 PWM 可用于控制上部功率开关，偶数编号的 PWM 则用于控制下部开关。

图 17-16: 互补 PWM 输出的典型负载



每个 PWM I/O 引脚对都可以通过将 PWMCON0 寄存器中相应的 PMODx 位清零来选择互补模式。缺省情况下，PWM I/O 引脚在所有的器件复位时被置为互补模式。

17.7 死区发生器

在功率逆变器应用中，PWM 以互补模式控制半桥电路的上下部开关，强烈建议使用死区插入。死区插入可以使两个输出暂时无效。在因 TON 和 TOFF 特性引起的功率器件状态改变时，避免开关过程中发生交叠。

因为功率输出器件不可能瞬时切换，所以必须在互补对中的 PWM 输出关闭事件和另一个晶体管开关开启事件之间留出一定的时间。PWM 模块允许对死区编程。后续部分详细说明了死区电路。

17.7.1 死区插入

PWM 模块的每个互补输出对都有一个 6 位的向下计数器，用于产生死区插入。如图 17-17 所示，每个死区单元都有与占空比比较输出连接的一个上升 / 下降沿探测器。当发生 PWM 边沿事件时，死区就会装入定时器。根据上升沿和下降沿的不同，在定时器向下计数至零后，互补输出中的一个转换才会延迟到那时进行。图 17-18 给出了将死区插入一对 PWM 输出的时序图。

图 17-17: PWM 输出对的死区控制单元框图

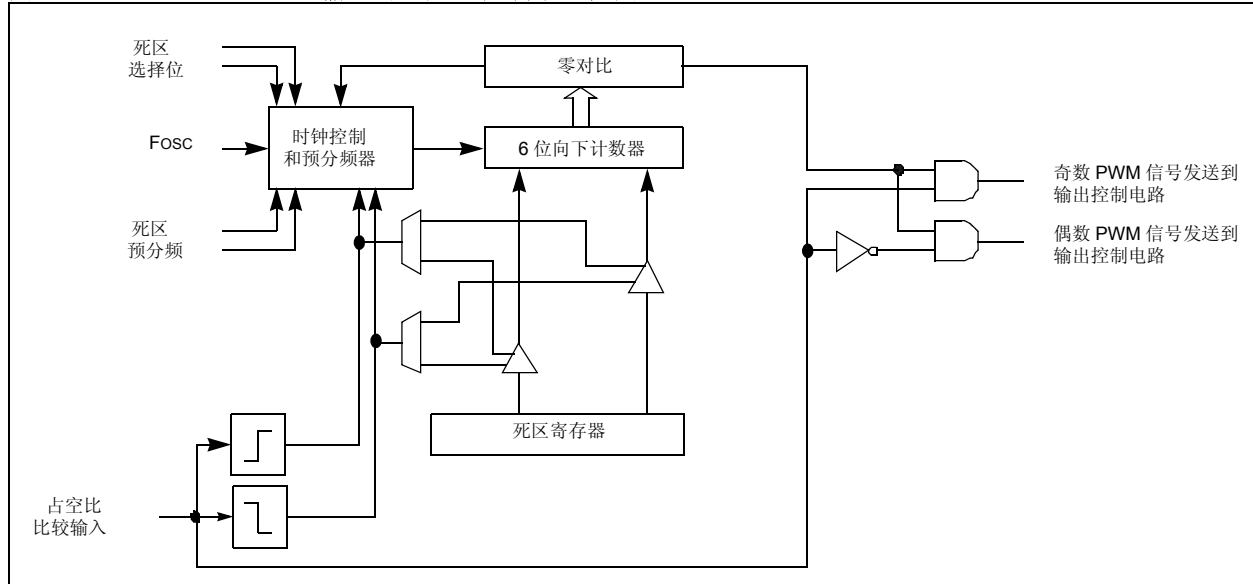
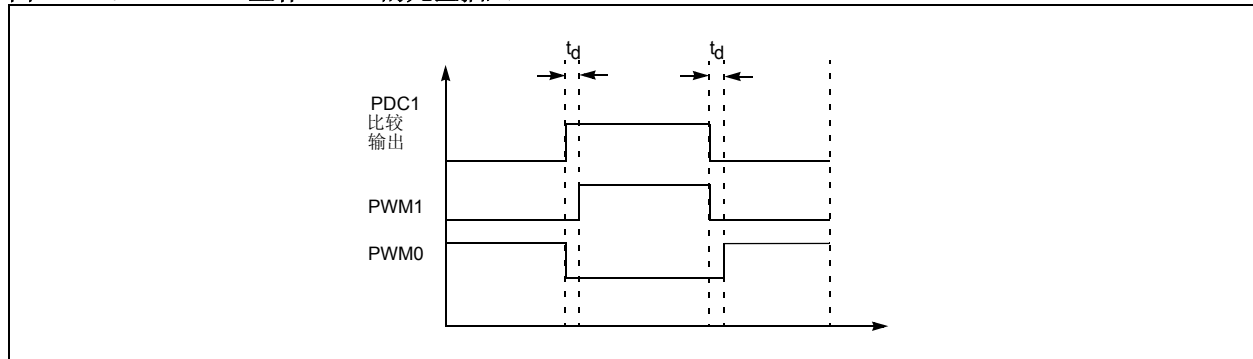


图 17-18: 互补 PWM 的死区插入



PIC18F2331/2431/4331/4431

寄存器 17-5: DTCON: 死区控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DTPS1	DTPS0	DT5	DT4	DT3	DT2	DT1	DT0

bit 7 bit 0

bit 7-6 **DTPS1:DTPS0**: 死区单元 A 预分频值选择位

- 11 = 死区单元时钟源为 $F_{osc}/16$ 。
- 10 = 死区单元时钟源为 $F_{osc}/8$ 。
- 01 = 死区单元时钟源为 $F_{osc}/4$ 。
- 00 = 死区单元时钟源为 $F_{osc}/2$ 。

bit 5-0 **DT5:DT0**: 死区单元的无符号 6 位死区值位。

图注:

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

17.7.2 死区范围

通过在 DTCON 寄存器中指定输入时钟预分频值的 6 位无符号值, 选择死区单元提供的死区量。为了根据器件工作频率提供适当的死区范围, 有四个输入时钟预分频值可供选择。Fosc/2、Fosc/4、Fosc/8 和 Fosc/16 为时钟预分频值选项, 可以用 DTCON 寄存器中的 DTPS1:DTPS0 控制位来选择。

在选择了合适的预分频值后, 将 6 位无符号值装入 DTCON<5:0> 来调整死区。发生如下任何事件时, 死区单元预分频值会被清零:

- 由于占空比比较边沿事件, 装入向下定时器;
- 写 DTCON 寄存器; 或
- 任何器件复位。

17.7.3 递减死区计数器

基于以下条件, 死区计数器以任意 Q 时钟作为时钟源。

1. 当如下情况发生时, 死区计数器以 Q1 作为时钟源:
 - DTPS 位设置为以下任一死区预分频值:
Fosc/4、Fosc/8 或 Fosc/16;
 - PWM 时基预分频位 (PTCKPS) 设置为以下任一预分频值: Fosc/16、Fosc/64 和 Fosc/256。
2. 当 PWM 时基预分频位设置为 1:1 (PTCKPS1:PTCKPS0 = 00, Fosc/4), 且死区计数器时钟源频率为 Fosc/2 (DTPS1:DTPS0 = 00), 死区计数器由一对 Q 时钟提供时钟源。

3. 根据占空比寄存器的两个 LSb 的置位情况, 死区计数器可使用所有其他 Q 时钟作为时钟源:

- 如果在 Q1 或 Q3 发生 PWM 占空比匹配, 则死区计数器使用 Q1 和 Q3 作为时钟源;
- 如果在 Q2 或 Q4 发生 PWM 占空比匹配, 则死区计数器使用 Q2 和 Q4 作为时钟源。

4. 当 DTPS1:DTPS0 位设置为其他死区预分频值设置 (即 Fosc/4、Fosc/8 或 Fosc/16), 且 PWM 时基预分频值置为 1:1 时, 死区计数器的时钟源是发生 PWM 占空比匹配时相应的 Q 时钟。

根据 DTCON 寄存器的值, 用如下公式计算实际死区:

$$\text{死区} = \text{死区值} / (\text{Fosc} / \text{预分频值})$$

表 17-3 给出的示例为死区范围与所选输入时钟预分频值和器件工作频率的函数关系。

表 17-3: 死区范围示例

Fosc (MHz)	MIPS	预分频值选择	最小死区	最大死区
40	10	Fosc/2	50 ns	3.2 μs
40	10	Fosc/4	100 ns	6.4 μs
40	10	Fosc/8	200 ns	12.8 μs
40	10	Fosc/16	400 ns	25.6 μs
32	8	Fosc/2	62.5 ns	4 μs
32	8	Fosc/4	125 ns	8 μs
32	8	Fosc/8	250 ns	16 μs
32	8	Fosc/16	500 ns	32 μs
25	6.25	Fosc/2	80 ns	5.12 μs
25	6.25	Fosc/4	160 ns	10.2 μs
25	6.25	Fosc/8	320 ns	20.5 μs
25	6.25	Fosc/16	640 ns	41 μs
20	5	Fosc/2	100 ns	6.4 μs
20	5	Fosc/4	200 ns	12.8 μs
20	5	Fosc/8	400	25.6 μs
20	5	Fosc/16	800	51.2 μs
10	2.5	Fosc/2	200 ns	12.8 μs
10	2.5	Fosc/4	400 ns	25.6 μs
10	2.5	Fosc/8	800 ns	51.2 μs
10	2.5	Fosc/16	1.6 μs	102.4 μs
5	1.25	Fosc/2	400 ns	25.6 μs
5	1.25	Fosc/4	800 ns	51.2 μs
5	1.25	Fosc/8	1.6 μs	102.4 μs
5	1.25	Fosc/16	3.2 μs	204.8 μs
4	1	Fosc/2	.5 μs	32 μs
4	1	Fosc/4	1 μs	64 μs
4	1	Fosc/8	2 μs	128 μs
4	1	Fosc/16	4 μs	256 μs

17.7.4 死区失真

- 注 1:** 对于较小的 PWM 占空比, 死区与有效 PWM 时间的比可能会很大。在这种情况下, 插入的死区可能会导致 PWM 模块产生的波形发生失真。至少保持 PWM 占空比为死区的 4 倍, 用户就可以确保将死区失真的影响降到最低。在占空比为 100% 或接近 100% 时, 会产生类似影响。在应用中, 应使用最大占空比, 使最小无效时间至少是死区的 4 倍。如果死区大于或等于其中一个 PWM 输出对的占空比, 则该 PWM 对在整个周期内都会保持无效。
- 2:** 当 PWM 使能时, 修改 DTCON 中的死区值可能会导致意外结果。在修改死区值前应先禁止 PWM (PTEN = 0)。

17.8 独立 PWM 输出

如图 17-19 所示, 该电路用于驱动开关磁阻电机的一个绕组, 也就是说, 独立 PWM 模式可用于驱动负载。当 PWMCON0 寄存器中相应的 PMOD 位置位时, 会将一对特定 PWM 输出配置为独立输出模式。当模块工作在独立模式下, 且允许两个 I/O 引脚同时有效时, 在 PWM I/O 引脚之间不实现死区控制。此模式还用于驱动步进电机。

17.8.1 独立模式下的占空比分配

在独立模式下, 每个占空比发生器都与给定的 PWM 输出对的两个 PWM 输出引脚连接。奇数和偶数 PWM 输出引脚都由单个 PWM 占空比发生器驱动。PWM1 和 PWM0 由 PDC0 寄存器设置占空比的 PWM 通道驱动。相应地, PWM3 和 PWM2 对应的是 PDC1, PWM5 和 PWM4 对应的是 PDC2, PWM7 和 PWM6 对应的是 PDC3, 参见图 17-3 和寄存器 17-3。

PIC18F2331/2431/4331/4431

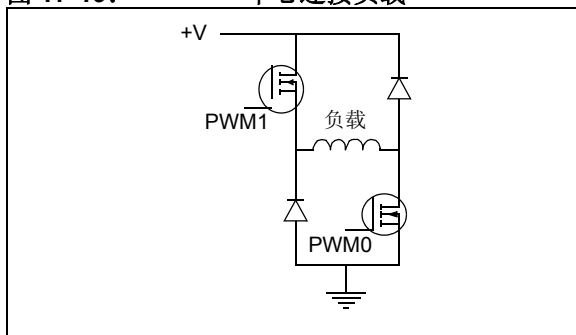
17.8.2 PWM 通道改写

使用 OVDCOND 和 OVDCONS 寄存器中适当的位可以手工改写每个 PWM 通道的 PWM 输出。对于每个工作在独立模式下的 PWM 输出引脚，用户可以选择下列信号输出选项：

- I/O 引脚输出 PWM 信号
- I/O 引脚无效
- I/O 引脚有效

如需所有改写功能的详细信息，请参阅第 17.10 节“PWM 输出改写”。

图 17-19: 中心连接负载



17.9 单脉冲 PWM 工作模式

仅在边沿对齐模式下可以使用单脉冲 PWM 工作模式。在此模式下，PWM 模块将产生单脉冲输出。当 PTCON0 寄存器中的 PTMOD1:PTMOD0 位设置为 01 时，模块将被配置为单脉冲工作模式。此工作模式对于驱动某些类型的 ECM 很有用。

在单脉冲模式下，当 PTEN 位置位时，PWM I/O 引脚被驱动为有效状态。当 PWM 定时器与占空比寄存器匹配时，PWM I/O 引脚被驱动为无效状态。当 PWM 定时器与 PTPER 寄存器匹配时，PTMR 寄存器清零，所有的有效 PWM I/O 引脚被驱动为无效状态，PTEN 位清零，如果相应的中断位置位，还会产生一个中断。

注： 在单脉冲输出后，PTPER 和 PDC 保持原值。要产生另一个单脉冲周期，只需要将 PTEN 位置位使能。

17.10 PWM 输出改写

无论占空比较单元的状态如何，PWM 输出改写位允许用户手工将 PWM I/O 引脚驱动为指定的逻辑状态。在控制各种 ECM（如 BLDC 电机）时，PWM 改写位是很有用的。

使用 OVDCOND 和 OVDCONS 寄存器定义 PWM 改写选项。OVDCOND 寄存器包含 8 位 (POVD7:POVD0)，用于决定要改写哪个 PWM I/O 引脚。OVDCONS 寄存器也包含 8 位 (POUT7:POUT0)，当通过 POVD 位改写特定输出时，这 8 位决定 PWM I/O 引脚的状态。

POVD 位是低电平有效控制位。当 POVD 位置位时，相应的 POUT 位并不影响 PWM 输出。换句话说，与置位的 POVD 位对应的引脚的 PWM 占空比由 PDC 寄存器决定。当一个 POVD 位清零后，相应 PWM I/O 引脚的输出将由 POUT 位的状态决定。当 POUT 位置位时，PWM 引脚将被驱动为有效状态。当 POUT 位清零时，PWM 引脚将被驱动为无效状态。

17.10.1 互补输出模式

当一对 PWM I/O 引脚工作在互补模式 (PMODx = 0) 时，对偶数 PWM I/O 引脚的改写有限制。在互补模式下，如果通过将相应的 POVD 位清零且 OVDCOND 和 OVDCONS 寄存器中的 POUT 位置位使偶数编号的引脚驱动为有效，则输出信号将被强制作为引脚对中奇数 I/O 引脚的互补 (如需详细信息，请参见图 17-2)。

17.10.2 改写同步

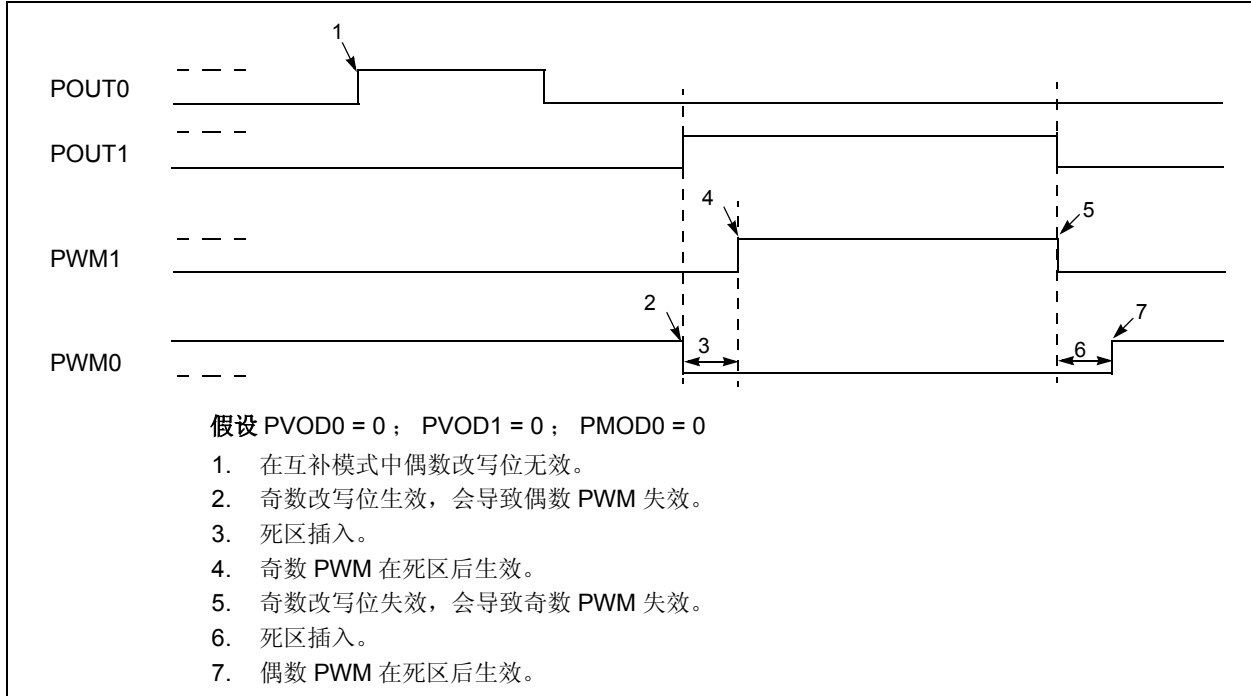
如果 PWMCON1 寄存器中的 OSYNC 位置位，所有通过 OVDCOND 和 OVDCONS 寄存器改写的输出都会与 PWM 时基同步。在下列情况中会发生同步输出改写：

- PWM 处于边沿对齐模式，当 PTMR 为 0 时发生同步。
- PWM 处于中心对齐模式，当 PTMR 为 0 且 PTMR 与 PTPER 的值匹配时发生同步。

注 1： 在互补模式中，当奇数通道被驱动为有效时，不会因错误或改写事件强制偶数通道驱动为有效。死区插入的情况下，奇数通道被驱动为有效状态前，偶数通道总是奇数通道的互补 (参见图 17-20)。

注 2： 即使 PWM 通道处于改写模式，死区也会插入 PWM 通道。

图 17-20: 互补模式下改写位



17.10.3 输出改写示例

图 17-21 给出了使用 PWM 输出改写功能时，可能会产生的波形示例。该图给出了 BLDC 电机的 6 步换相序列。如图 17-16 所示，电机由三相逆变器驱动。当检测到相应的转子位置时，PWM 输出会切换到序列中的下一个换相状态。在此例中，PWM 输出驱动为特定的逻辑状态。表 17-4 给出了用来生成图 17-21 中信号的 OVDCOND 和 OVDCONS 寄存器的值。

PWM 占空比寄存器与 OVDCOND 和 OVDCONS 寄存器配合使用。负载上的电压由占空比寄存器控制，换相序列则由 OVDCOND 和 OVDCONS 寄存器控制。图 17-22 给出了波形图，表 17-4 和表 17-5 给出了用于生成所需信号的 OVDCOND 和 OVDCONS 寄存器的值。

寄存器 17-6: OVDCOND: 输出改写控制寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
POVD7 ⁽¹⁾	POVD6 ⁽¹⁾	POVD5	POVD4	POVD3	POVD2	POVD1	POVD0
bit 7					bit 0		

bit 7-0 **POVD7:POVD0: PWM 输出改写位⁽¹⁾**

- 1 = PWM I/O 引脚输出由占空比寄存器的值和 PWM 时基控制。
- 0 = PWM I/O 引脚输出由相应 POUT 位的值控制。

注 1: PIC18F2X31 器件中未实现；保持这些位清零。

图注:

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

PIC18F2331/2431/4331/4431

寄存器 17-7: **OVDCONS: 输出状态寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
POUT7 ⁽¹⁾	POUT6 ⁽¹⁾	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
bit 7							bit 0

bit7-0 **POUT7:POUT0: PWM 手工输出位⁽¹⁾**

- 1 = 当相应 PWM 输出改写位清零时, PWM I/O 引脚输出有效。
- 0 = 当相应 PWM 输出改写位清零时, PWM I/O 引脚输出无效。

注 1: PIC18F2X31 器件中未实现; 保持这些位清零。

图注:

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

图 17-21: PWM 输出改写示例 1

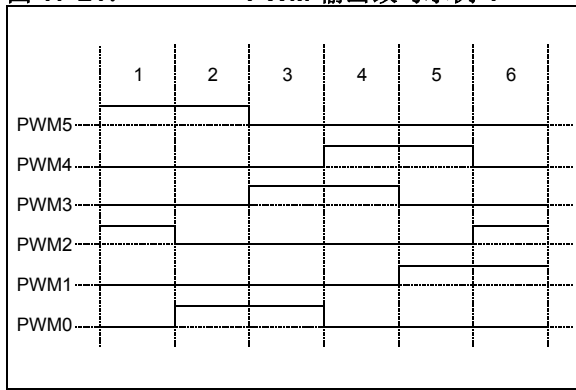


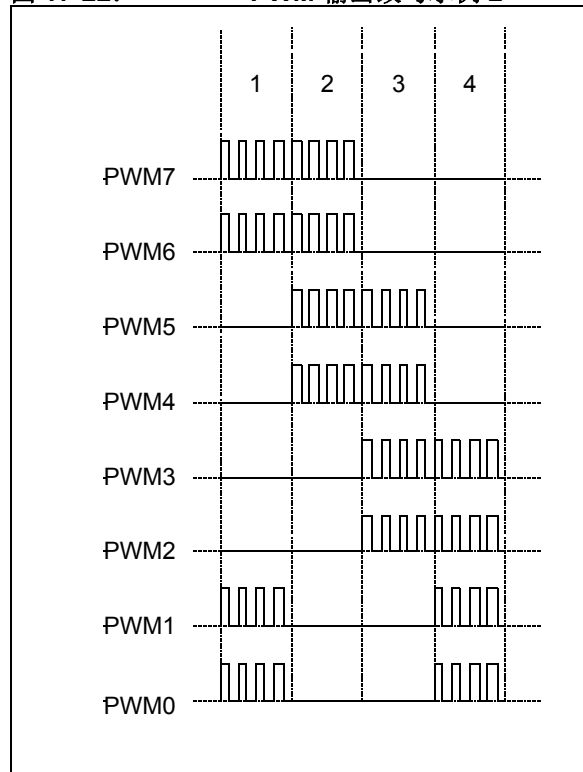
表 17-4: PWM 输出改写示例 1

状态	OVDCOND(POVD)	OVDCONS(POUT)
1	00000000b	00100100b
2	00000000b	00100001b
3	00000000b	00001001b
4	00000000b	00011000b
5	00000000b	00010010b
6	00000000b	00000110b

表 17-5: PWM 输出改写示例 2

状态	OVDCOND(POVD)	OVDCONS(POUT)
1	11000011b	00000000b
2	11110000b	00000000b
3	00111100b	00000000b
4	00001111b	00000000b

图 17-22: PWM 输出改写示例 2



17.11 PWM 输出与极性控制

有 3 个与 PWM 模块相关的器件配置位，它们提供 CONFIG3L 配置寄存器中定义的 PWM 输出引脚控制。

- HPOL 配置位
- LPOL 配置位
- PWMPIN 配置位

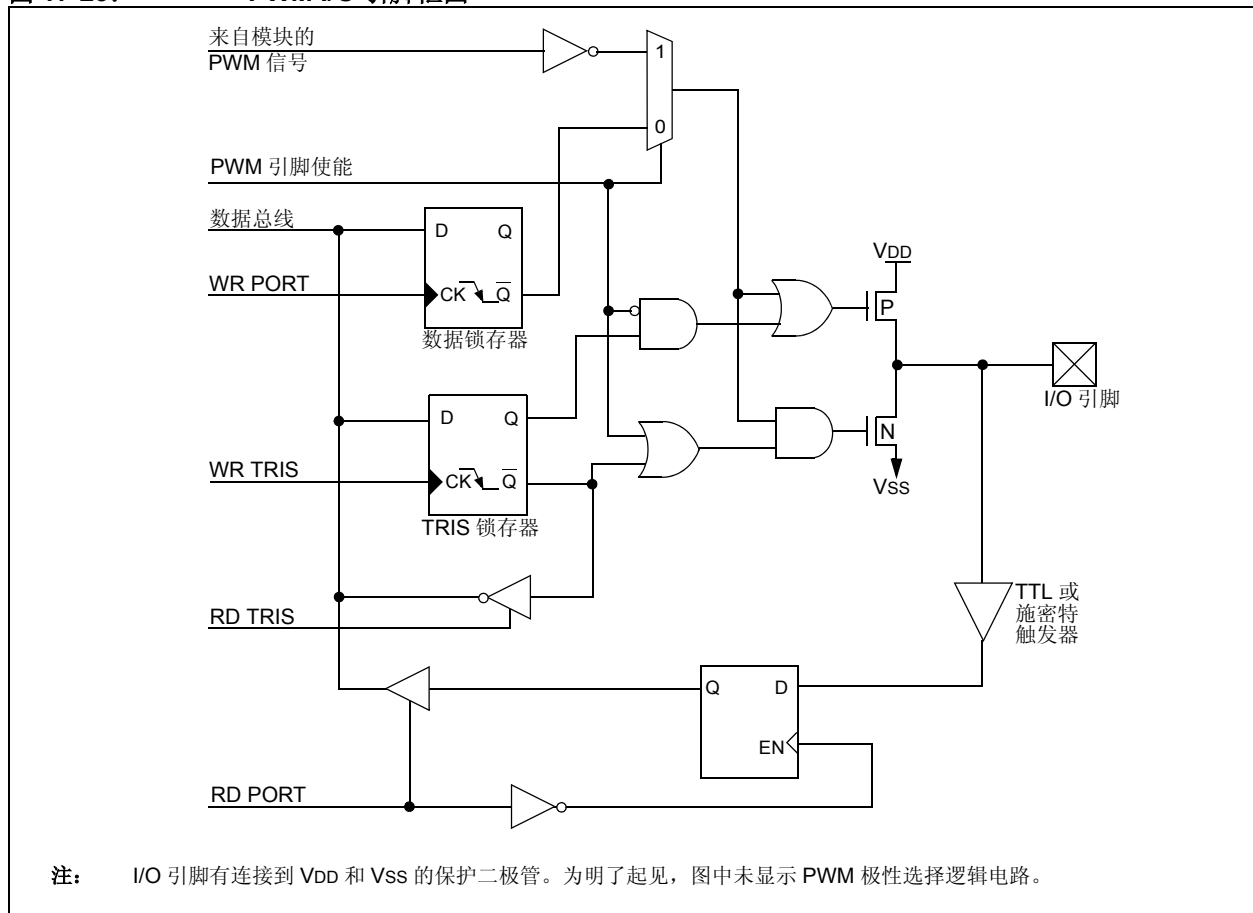
这 3 个配置位与 PWMCON0 寄存器中的 3 个 PWM 使能位 (PWMEN2:PWMEN0) 联合工作。配置位和 PWM 使能位确保发生器件复位后，PWM 引脚处于正确的状态。

17.11.1 输出引脚控制

按应用所需，PWMEN2:PWMEN0 控制位使能每个 PWM 输出引脚。

所有的 PWM I/O 引脚都是通用 I/O。当一对引脚使能为 PWM 输出时，禁止控制引脚的 PORT 和 TRIS 寄存器。如需详细信息，请参见图 17-23。

图 17-23: PWM I/O 引脚框图



17.11.2 输出极性控制

在器件编程时，通过 CONFIG3L 器件配置寄存器中的 HPOL 和 LPOL 配置位设置 PWM I/O 引脚的极性。HPOL 配置位设置的是上半 PWM 输出 (PWM1、PWM3、PWM5 和 PWM7) 的极性。当 HPOL 清零 (= 0) 时，极性为高电平有效；当其置位 (= 1) 时，极性为低电平有效。

LPOL 配置位设置的是下半 PWM 输出 (PWM0、PWM2、PWM4 和 PWM6) 的极性。与 HPOL 一样，当 LPOL 清零时，这些引脚为高电平有效。当 LPOL 置位时，极性为低电平有效。

PWM 模块产生的所有的输出信号 (包括由于故障输入或手工改写产生的信号) 都会引用极性控制位 (参见第 17.10 节 “PWM 输出改写”)。

缺省极性配置位是使 PWM I/O 引脚输出极性为高电平有效。

PIC18F2331/2431/4331/4431

17.11.3 PWM 输出引脚复位状态

器件脱离复位状态后，PWMPIN 配置位决定 PWM 输出引脚是 PWM 输出引脚还是数字 I/O 引脚。如果 PWMPIN 配置位未编程（缺省），PWMEN2:PWMEN0 控制位会在器件复位时清零。因此，所有的 PWM 输出都将为三态，并由相应的 PORT 和 TRIS 寄存器控制。如果 PWMPIN 配置位已编程为低电平，PWMEN2:PWMEN0 控制位会在器件复位时置位：

- 如果器件有 8 个 PWM 引脚（PIC18F4X31 器件），则 PWMEN2:PWMEN0 = 101
- 如果器件有 6 个 PWM 引脚（PIC18F2X31 器件），则 PWMEN2:PWMEN0 = 100

所有的 PWM 引脚将使能 PWM 输出引脚，且其输出极性由 HPOL 和 LPOL 配置位定义。

17.12 PWM 故障输入

有两种与 PWM 模块相关的故障输入。输入故障引脚的主要作用是禁止 PWM 输出信号，并将其驱动为无效状态。对故障输入的操作直接在硬件中执行，这样的话，当发生故障时，可以快速做出反应，且 PWM 输出被置为无效状态，以便保护连接到 PWM 的电源设备。

PWM 故障输入为 \overline{FLTA} 和 \overline{FLTB} ，可以来自 I/O 引脚、CPU 或另一个模块。 \overline{FLTA} 和 \overline{FLTB} 引脚为低电平有效输入，因此很容易将很多信号源进行“或”运算，并进入到同一个输入。

FLTCONFIG 寄存器（寄存器 17-8）定义 \overline{FLTA} 和 \overline{FLTB} 输入设置。

注：	PWM 引脚的无效状态与 HPOL 和 LPOL 配置位的设置情况有关，这两个配置位定义了 PWM 输出的有效 / 无效状态。
-----------	---

17.12.1 故障引脚使能位

将 FLTCONFIG 寄存器中的 FLTAEN 和 FLTBEN 位置位，可使能相应的故障输入。如果这两个位都被清零，则故障输入对 PWM 模块没有影响。

17.12.2 故障输入模式

当 PWM I/O 引脚因故障输入而改写为无效时，FLTCONFIG 寄存器中的 FLTAMOD 和 FLTBMOD 位决定这些引脚的模式。

FLTCONFIG 寄存器中的 FLTAS 和 FLTBS 位表明 FaultA 和 FaultB 输入的状态。

每个故障输入都有两个工作模式：

- **无效模式（FLT_xMOD = 0）**

这是灾难性故障管理模式。当发生此模式的故障时，PWM 输出变为无效。在故障清除（故障输入驱动为高电平），并且相应的故障状态位用软件清零前，PWM 引脚保持无效状态。故障状态位（FLT_xS）清零后，PWM 输出在下一个 PWM 周期开始时立即使能。

- **逐周期模式（FLT_xMOD = 1）**

当发生此模式的故障时，PWM 输出无效。只要故障引脚保持低电平，PWM 输出将保持定义的故障状态（所有的 PWM 输出无效）。故障引脚驱动为高电平后，PWM 输出将在下一个 PWM 周期开始时返回正常工作状态，且 FLTS 位自动清零。

17.12.3 故障状态下的 PWM 输出

当处于故障状态（即 \overline{FLTA} 和 / 或 $\overline{FLT B}$ 输出有效）时，PWM 输出信号被驱动为无效。如下所示，选择使哪个 PWM 输出失效（在故障状态下）取决于 FLTCONFIG 寄存器的 FLTCON 位：

- FLTCON = 1。当 \overline{FLTA} 或 $\overline{FLT B}$ 拉高时，PWM 输入（即 PWM[7:0]）被驱动为无效状态
- FLTCON = 0。当 \overline{FLTA} 或 $\overline{FLT B}$ 拉高时，只有 PWM[5:0] 输出被驱动为无效，PWM[7:6] 保持有效。

注： 在故障状态下，只禁止 3 个 PWM 通道，1 个 PWM 通道保持使能；这可以使器件具有至少使能一个 PWM 通道的灵活性。当 FLTCON = 1，且出现故障状态时，所有的 PWM 输出都被禁止（由 PWM 占空比寄存器驱动）。

注： 在开发固件并使用了高功率电路时，如果使用调试工具，强烈建议使能在断点产生故障条件。在调试完固件可以对器件进行编程时，就可以禁止 BRFEN 位。

17.12.4 调试模式下的 PWM 输出

使用在线仿真器（In-Circuit Emulator, ICE）或在线调试器（In-Circuit Debugger, ICD）调试应用时，FLTCONFIG 寄存器中的 BRFEN 位控制遇到断点时故障条件的模拟。BRFEN 置为高，允许在断点产生故障状态，这将使 PWM 输出驱动为无效状态。这样做的目的是为了以防连续保持 PWM 引脚的状态，在这种情况下，有可能会连接到 PWM 输出上的电源设备遭到损坏。

如果 BRFEN = 0，则禁止在断点产生故障状态。

PIC18F2331/2431/4331/4431

寄存器 17-8: FLTCONFIG: 故障配置寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRFEN	FLTBS ⁽¹⁾	FLTBMOD ⁽¹⁾	FLT BEN ⁽¹⁾	FLTCON	FLTAS	FLTA-MOD	FLTAEN
bit 7						bit 0	

- bit 7 **BRFEN:** 断点故障使能位
 1 = 在断点（即，仅当 HDMIN = 1 时）时使能故障状态
 0 = 禁止故障状态
- bit 6 **FLTBS:** 故障 B 状态位 ⁽¹⁾
 1 = FLT B 拉高;
 如果 FLTBMOD = 0, 则由用户清零
 如果 FLTBMOD = 1, 当 FLT B 拉低时, 在新周期开始时自动清零
 0 = 无故障
- bit 5 **FLTBMOD:** 故障 B 模式位 ⁽¹⁾
 1 = 逐周期模式: 引脚在当前 PWM 周期剩余时间或 FLT B 拉低前无效。当 FLT B 无效（没有出现故障）时, FLTBS 自动清零。
 0 = 无效模式: 引脚失效（灾难性失败）, 直到 FLT B 被拉低且 FLTBS 仅由用户清零。
- bit 4 **FLT BEN:** 故障 B 使能位 ⁽¹⁾
 1 = 故障 B 使能
 0 = 故障 B 禁止
- bit 3 **FLTCON:** 故障配置位
 1 = FLTA 和 / 或 FLT B 使所有的 PWM 输出失效
 0 = FLTA 或 FLT B 使 PWM[5:0] 失效
- bit 2 **FLTAS:** 故障 A 状态位
 1 = FLTA 拉高;
 如果 FLTAMOD = 0, 则由用户清零
 如果 FLTAMOD = 1, 当 FLTA 拉低时, 在新周期开始时自动清零
 0 = 无故障
- bit 1 **FLTAMOD:** 故障 A 模式位
 1 = 逐周期模式: 引脚在当前 PWM 周期剩余时间或 FLTA 拉低前无效。FLTAS 会自动清零。
 0 = 无效模式: 引脚失效（灾难性失败）, 直到 FLTA 被拉低且 FLTAS 仅由用户清零。
- bit 0 **FLTAEN:** 故障 A 使能位
 1 = 故障 A 使能
 0 = 故障 A 禁止

注 1: PIC18F2X31 器件中未实现; 保持这些位清零。

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

17.13 PWM 更新锁定

对于复杂的 PWM 应用，在给定时间内，用户最多可能需要写 4 个占空比寄存器和时基周期寄存器 PTPER。在某些应用中，在装入新占空比和周期值供模块使用前写入所有缓冲寄存器是很重要的。

可选择使能 PWM 更新锁定功能，因此用户可以指定何时新占空比缓冲器值有效。将 PWMCON1 寄存器中的 UDIS 控制位置位可以使能 PWM 更新锁定功能。此位对所有占空比缓冲寄存器和 PWM 时基周期缓冲器 PTPER 都有影响。

执行 PWM 更新锁定：

1. 将 UDIS 位置位。
2. 如果适用，写所有占空比寄存器和 PTPER。
3. 将 UDIS 位清零，重新使能更新。
4. 这样，UDIS 位清零后，缓冲器值将装入真正的寄存器。这会同步装入寄存器。

17.14 PWM 特殊事件触发器

PWM 模块有一个特殊事件触发器功能，可以让 A/D 转换与 PWM 时基同步。A/D 采样和转换时间可以通过编程在 PWM 周期内的任意时间点发生。特殊事件触发器可以让用户在采集 A/D 转换结果和更新占空比值时的延迟最小化。

PWM 16 位特殊事件触发器寄存器 SEVTCMP（高或低）和 PWMCON1 寄存器中的 5 个控制位用于控制其操作。

应当发生的特殊事件触发器的 PTMR 值装入 SEVTCMP 寄存器对。PWMCON1 寄存器中的 SEVTDIR 位指定当 PWM 时基处于向上或向下计数模式时的计数相位。

如果 SEVTDIR 位清零，特殊事件触发器将在 PWM 时基的向上计数周期发生。如果 SEVTDIR 位置位，特殊事件触发器将在 PWM 时基的向下计数周期发生。SEVTDIR 位仅在 PWM 定时器处于向上或向下计数模式时才有效。

17.14.1 特殊事件触发器使能

PWM 模块总是会产生特殊事件触发脉冲。此信号也可以由 A/D 模块选用。如需详细信息，请参阅第 20.0 节“10 位高速模数转换器（A/D）模块”。

17.14.2 特殊事件触发器后分频器

PWM 特殊事件触发器有一个后分频器，其后分频值为 1:1 到 1:16。通过写 PWMCON1 寄存器中的 SEVOPS3:SEVOPS0 控制位可以配置后分频值。

在任何对 SEVTCMP 寄存器对的写操作或器件复位时，特殊事件输出后分频器清零。

PIC18F2331/2431/4331/4431

表 17-6: 与功率控制 PWM 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	POR 和 BOR 时的值	所有其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBFIF	0000 000x	0000 000u
IPR3	—	—	—	PTIP	IC3DRIP	IC2QEIP	IC1IP	TMR5IP	---1 1111	---1 1111
PIE3	—	—	—	PTIE	IC3DRIE	IC2QEIE	IC1IE	TMR5IE	---0 0000	---0 0000
PIR3	—	—	—	PTIF	IC3DRIF	IC2QEIF	IC1IF	TMR5IF	---0 0000	---0 0000
PTCON0	PTOPS3	PTOPS2	PTOPS1	PTOPS0	PTCKPS1	PTCKPS0	PTMOD1	PTMOD0	0000 0000	0000 0000
PTCON1	PTEN	PTDIR	—	—	—	—	—	—	00-- ----	00-- ----
PTMRL ⁽¹⁾	PWM 时基 (低 8 位)								0000 0000	0000 0000
PTMRH ⁽¹⁾	—	—	—	—	PWM 时基 (高 4 位)				---- 0000	---- 0000
PTPERL ⁽¹⁾	PWM 时基周期 (低 8 位)								1111 1111	1111 1111
PTPERH ⁽¹⁾	—	—	—	—	PWM 时基周期 (高 4 位)				---- 1111	---- 1111
SEVTCMPL ⁽¹⁾	PWM 特殊事件比较 (低 8 位)								0000 0000	0000 0000
SEVTCMPH ⁽¹⁾	—	—	—	—	PWM 特殊事件比较 (高 4 位)				---- 0000	---- 0000
PWMCON0	—	PWMEN2	PWMEN1	PWMEN0	PMOD3 ⁽²⁾	PMOD2	PMOD1	PMOD0	-101 0000	-101 0000
PWMCON1	SEVOPS3	SEVOPS2	SEVOPS1	SEVOPS0	SEVTDIR	—	UDIS	OSYNC	0000 0-00	0000 0-00
DTCON	DTPS1	DTPS0	死区 A 值寄存器						0000 0000	0000 0000
FLTCONFIG	BRFEN	FLTBS ⁽²⁾	FLTBMOD ⁽²⁾	FLTBEN ⁽²⁾	FLTCON	FLTAS	FLTAMOD	FLTAEN	0000 0000	0000 0000
OVDCOND	POVD7 ⁽²⁾	POVD6 ⁽²⁾	POVD5	POVD4	POVD3	POVD2	POVD1	POVD0	1111 1111	1111 1111
OVDCONS	POUT7 ⁽²⁾	POUT6 ⁽²⁾	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0	0000 0000	0000 0000
PDC0L ⁽¹⁾	PWM 占空比 0L 寄存器 (低 8 位)								--00 0000	--00 0000
PDC0H ⁽¹⁾	—	—	PWM 占空比 0H 寄存器 (高 6 位)						0000 0000	0000 0000
PDC1L ⁽¹⁾	PWM 占空比 1L 寄存器 (低 8 位)								0000 0000	0000 0000
PDC1H ⁽¹⁾	—	—	PWM 占空比 1H 寄存器 (高 6 位)						--00 0000	--00 0000
PDC2L ⁽¹⁾	PWM 占空比 2L 寄存器 (低 8 位)								0000 0000	0000 0000
PDC2H ⁽¹⁾	—	—	PWM 占空比 2H 寄存器 (高 6 位)						--00 0000	--00 0000
PDC3L ^(1,2)	PWM 占空比 3L 寄存器 (低 8 位)								0000 0000	0000 0000
PDC3H ^(1,2)	—	—	PWM 占空比 3H 寄存器 (高 6 位)						--00 0000	--00 0000

图注: -- 未实现, u= 不变。阴影单元格表示功率控制 PWM 未使用。

注 1: 双缓冲寄存器对。有关寄存器读写的内容请参见正文。

注 2: PIC18F2X31 器件中未实现; 保持这些位清零。给出的是 PIC18F4X31 器件的复位值。

18.0 同步串行端口（SSP）模块

18.1 SSP 模块概述

同步串行端口（Synchronous Serial Port, SSP）模块是用于与其他外设或单片机器件通讯的非常有用的串行接口。这些外设器件包括串行 EEPROM、移位寄存器、显示驱动器和 A/D 转换器等。SSP 模块可以工作在以下两种模式之一：

- 串行外设接口（Serial Peripheral Interface, SPI™）
- 内部互联总线（Inter-Integrated Circuit, I²C™）

有关 SSP 模块的 I²C 工作原理概述和其他信息可以参考《PICmicro® 中档单片机系列参考手册》（DS33023A_CN）。

参见应用笔记 AN578, “Use of the SSP module in the I²C™ Multi-Master Environment”（DS00578）。

18.2 SPI 模式

本节包括寄存器定义和 SPI 模式的工作特点。有关 SPI 模块的其他信息，可以参考《PICmicro® 中档单片机系列参考手册》（DS33023A_CN）。

SPI 模式允许同时进行同步收发 8 位数据。要完成通讯，通常会使用 3 个引脚：

- 串行数据输出（SDO）– RC7/RX/DT/SDO
- 串行数据输入（SDI）– RC4/INT1/SDI/SDA
- 串行时钟（SCK）– RC5/INT2/SCK/SCL

此外，处于从动工作模式时，可能会用到第 4 个引脚。

- 从动选择（ \overline{SS} ）– RC6/TX/CK/ \overline{SS}

当初初始化 SPI 时，需要指定几个选项。需通过对 SSPCON 寄存器中相应的控制位（SSPCON<5:0>）和 SSPSTAT<7:6> 进行编程完成。这些控制位允许指定以下选项：

- 主控模式（SCK 作为时钟输出）
- 从动模式（SCK 作为时钟输入）
- 时钟极性（SCK 的空闲状态）
- 时钟边沿（在 SCK 上升沿 / 下降沿输出数据）
- 时钟速率（仅主控模式）
- 从动选择模式（仅从动模式）

PIC18F2331/2431/4331/4431

寄存器 18-1: **SSPSTAT: 同步串行端口状态寄存器 (地址 94h)**

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit 7						bit 0	

- bit 7 **SMP:** SPI 数据输入采样相位位
SPI 主控模式:
 1= 在数据输出结束时采样输入数据
 0= 在数据输出中间时采样输入数据 (Microwire®)
SPI 从动模式:
 当 SPI 用于从动模式时, 必须将 SMP 清零
I²C 模式:
 此位必须保持清零
- bit 6 **CKE:** SPI 时钟边沿选择位 (图 18-2、图 18-3 和图 18-4)
SPI 模式, CKP=0:
 1= 在 SCK 的上升沿发送数据 (Microwire® 备选方案)
 0= 在 SCK 的下降沿发送数据
SPI 模式, CKP=1:
 1= 在 SCK 的下降沿发送数据 (Microwire® 缺省方案)
 0= 在 SCK 的上升沿发送数据
I²C 模式:
 此位必须保持清零
- bit 5 **D/A:** 数据 / 地址位 (仅 I²C 模式)
 1= 表示接收或发送的最后字节是数据
 0= 表示接收或发送的最后字节是地址
- bit 4 **P:** 停止位 (仅 I²C 模式)
 当 SSP 模块禁止或最后检测到了停止位时, 此位会清零。
 SSPEN 清零。
 1= 表示最后检测到了停止位 (此位在复位时为 0)
 0= 最后没有检测到停止位
- bit 3 **S:** 起始位 (仅 I²C 模式)
 当 SSP 模块禁止或最后检测到了起始位时, 此位会清零。
 SSPEN 清零。
 1= 表示最后检测到了起始位 (此位在复位时为 0)
 0= 最后没有检测到起始位
- bit 2 **R/W:** 读 / 写位信息 (仅 I²C 模式)
 此位保存最后一个地址匹配后的 R/W 位信息。此位仅在地址匹配到下一个起始位、停止位或 ACK 位之间有效。
 1= 读
 0= 写
- bit 1 **UA:** 更新地址位 (仅 10 位 I²C 模式)
 1= 表示用户需要更新 SSPADD 寄存器中的地址
 0= 不需要更新地址
- bit 0 **BF:** 缓冲器满状态位
接收 (SPI 和 I²C 模式):
 1= 接收完成, SSPBUF 满
 0= 接收未完成, SSPBUF 空
发送 (仅 I²C 模式):
 1= 正在发送, SSPBUF 满
 0= 发送完成, SSPBUF 空

图注:

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

PIC18F2331/2431/4331/4431

寄存器 18-2: SSPCON: 同步串行端口控制寄存器 (地址 14h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7						bit 0	

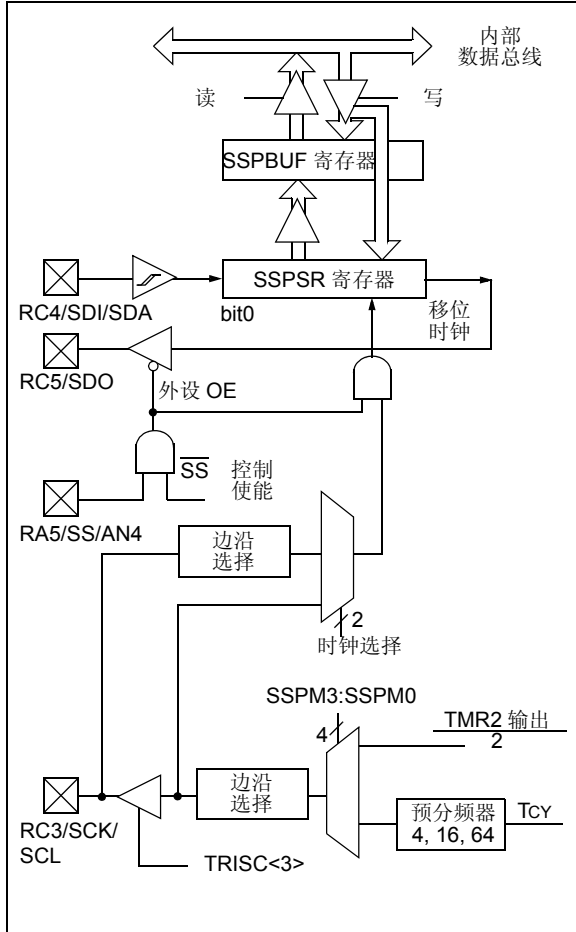
- bit 7 WCOL:** 写冲突检测位
 1= 正在发送前一个字时对 SSPBUF 寄存器进行写操作 (必须用软件清零)
 0= 无冲突
- bit 6 SSPOV:** 接收溢出指示位
 在 SPI 模式下:
 1= SSPBUF 寄存器中保存有前一数据时, 接收到新的字节。如果发生溢出, SSPSR 中的数据会丢失。仅在从动模式可能会发生溢出。即使只是发送数据, 用户也必须读 SSPBUF, 以避免置位溢出位。在主机模式下, 溢出位未置位, 因为每个新接收 (和发送) 的字节都是通过写 SSPBUF 寄存器开始的。
 0= 无溢出
 在 I²C 模式中:
 1= 当 SSPBUF 寄存器中仍保存原先的数据时, 接收到新字节。在发送模式下可以忽略 SSPOV。两种模式下 SSPOV 都必须用软件清零。
 0= 无溢出
- bit 5 SSPEN:** 同步串行端口使能位
 在 SPI 模式中:
 1= 使能串行端口, 并将 SCK、SDO 和 SDI 配置为串行端口引脚
 0= 禁止串行端口, 并将这些引脚配置为 I/O 端口引脚
 在 I²C 模式中:
 1= 使能串行端口, 并将 SDA 和 SCL 引脚配置为串行端口引脚
 0= 禁止串行端口, 并将这些引脚配置为 I/O 端口引脚
 在两种模式下, 当使能时, 这些引脚必须被正确配置为输入或输出引脚。
- bit 4 CKP:** 时钟极性选择位
 在 SPI 模式中:
 1= 高电平为时钟的空闲状态 (Microwire[®] 缺省)
 0= 低电平为时钟的空闲状态 (Microwire[®] 备选)
 在 I²C 模式中:
 SCK 释放控制
 1= 使能时钟
 0= 保持时钟低电平 (时钟延长)。(用于确保数据建立时间。)
- bit 3-0 SSPM3:SSPM0:** 同步串行端口模式选择位
 0000 = SPI 主控模式, 时钟 =Fosc/4
 0001 = SPI 主控模式, 时钟 =Fosc/16
 0010 = SPI 主控模式, 时钟 =Fosc/64
 0011 = SPI 主控模式, 时钟 =TMR2 输出 /2
 0100 = SPI 从动模式, 时钟 =SCK 引脚。使能 SS 引脚控制。
 0101 = SPI 从动模式, 时钟 =SCK 引脚。禁止 SS 引脚控制。SS 可作为 I/O 引脚使用。
 0110 = I²C 从动模式, 7 位地址
 0111 = I²C 从动模式, 10 位地址
 1011 = I²C 固件控制主控模式 (从单元空闲)
 1110 = I²C 从动模式, 7 位地址, 且起始位和停止位中断使能
 1111 = I²C 从动模式, 10 位地址, 且起始位和停止位中断使能

图注:

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

PIC18F2331/2431/4331/4431

图 18-1: SSP 框图 (SPI 模式)



要启用串行端口，必须将 SSP 使能位 SSPEN (SSPCON<5>) 置位。要复位或重新配置 SPI 模式，需将 SSPEN 位清零，重新初始化 SSPCON 寄存器，然后将 SSPEN 位置位。这样做会将 SDI、SDO、SCK 和 SS 引脚配置为串行端口引脚。要使这些引脚作为串行端口，必须对它们的数据方向位 (在 TRISC 寄存器中) 正确编程。即：

- 必须将 SDI 的 TRISC<4> 置位
- 必须将 SDO 的 TRISC<5> 清零
- 必须将 SCK (主控模式) 的 TRISC<3> 清零
- 必须将 SCK (从动模式) 的 TRISC<3> 置位
- 必须将 SS 的 TRISA<5> 置位，而且必须配置 ADCON 使 RA5 作为数字 I/O

- 注 1:** 当 SPI 处于从动模式且 SS 引脚控制使能 (SSPCON<3:0>=0100) 时，如果 SS 引脚设置为 VDD，SPI 模块将会复位。
- 2:** 如果使用 SPI 从动模式，且 CKE = 1，则必须使能 SS 引脚控制。
- 3:** 当 SPI 处于从动模式且使能 SS 引脚控制 (SSPCON<3:0>=0100) 时，SS 引脚的状态可能会影响从 TRISC<5> 位读取的状态。从 SSP 模块向 PORTC 发送的外设 OE 信号控制从 TRISC<5> 位读取的状态 (如需了解 PORTC 的更多信息，请参阅第 10.3 节“PORTC、TRISC 和 LATC 寄存器”)。如果对 TRISC 寄存器执行一修改一写指令 (例如 BSF)，而 SS 引脚处于高电平，这将使 TRISC<5> 位置位，从而禁止 SDO 输出。

图 18-2: SPI 模式时序, 主控模式

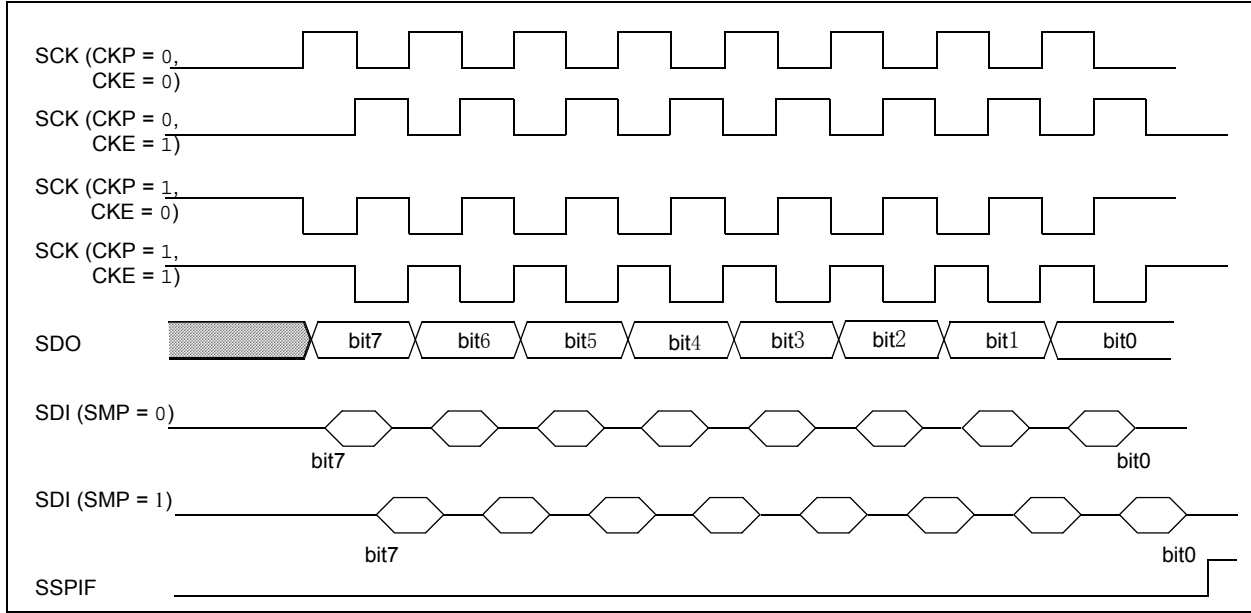
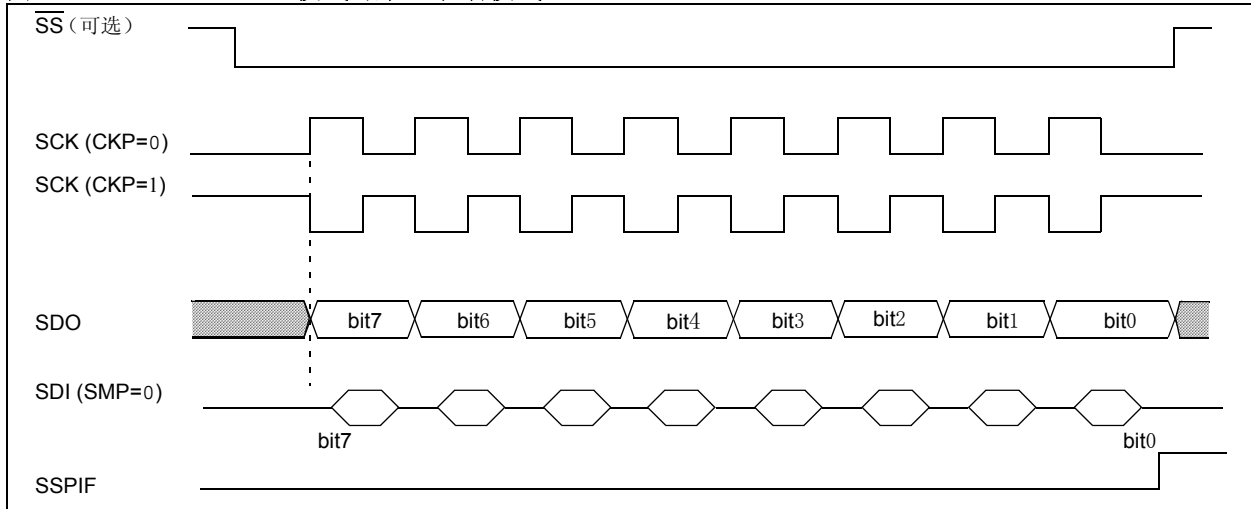


图 18-3: SPI 模式时序 (从动模式, CKE = 0)



PIC18F2331/2431/4331/4431

图 18-4: SPI 模式时序 (从动模式, CKE = 1)

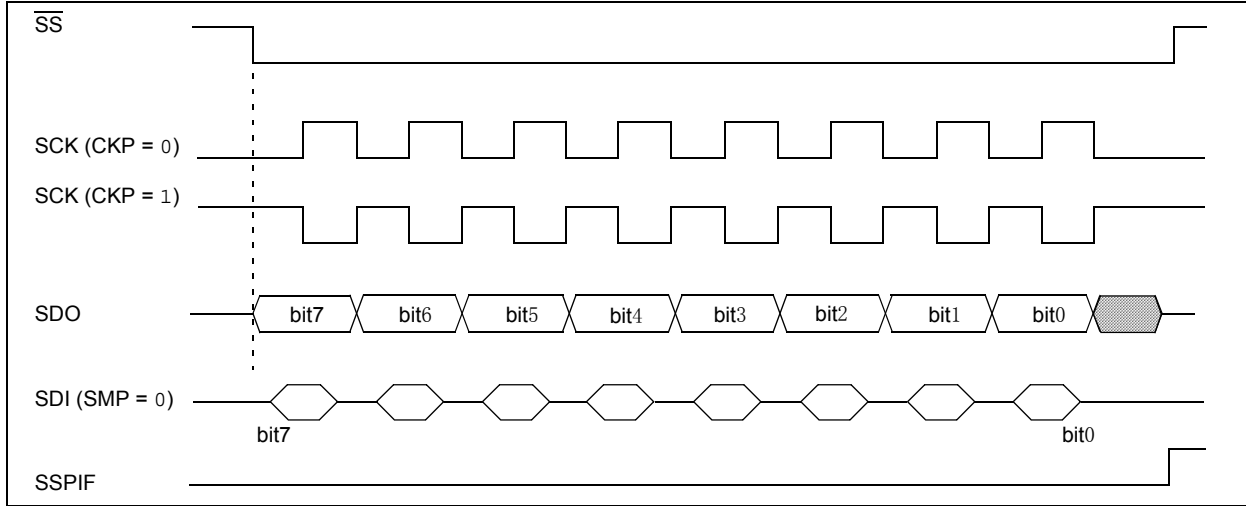


表 18-1: 与 SPI 工作模式相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
TRISC	PORTC 数据方向寄存器								1111 1111	1111 1111
SSPBUF	同步串行端口接收缓冲器 / 发送寄存器								xxxxx xxxxx	uuuu uuuu
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
TRISA	—	—	PORTA 数据方向寄存器						--11 1111	--11 1111
SSP-STAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000

图注: x= 未知, u= 不变, -= 未实现, 读作 0。阴影单元格表示 SSP 未在 SPI 模式中使用。

注 1: 在 PIC16F73/76 中保留了 PSPIE 和 PSPIF 位; 应该将这些位保持为零。

18.3 SSP I²C 工作原理

除了全局呼叫支持外，I²C 模式的 SSP 模块几乎完全实现了所有的从动功能，并提供了起始位和停止位的硬件中断，帮助固件实现主控功能。SSP 模块实现标准模式规范及 7 位和 10 位寻址。

数据传输使用了两个引脚。它们是 SCK/SCL 引脚和 SDI/SDA 引脚，即时钟引脚（SCL）和数据引脚（SDA）。用户必须通过 TRISC<5:4> 或 TRISD<3:2> 位将这两个引脚配置为输入或输出。

通过将 SSP 使能位 SSPEN（SSPCON<5>）置位，可以使能 SSP 模式的功能。

通过 SSPCON 寄存器可以控制 I²C 的工作状况。通过 4 个模式选择位（SSPCON<3:0>）可以选择以下 I²C 模式之一：

- I²C 从动模式（7 位地址）
- I²C 从动模式（10 位地址）
- I²C 从动模式（7 位地址），使能起始和停止位中断，支持固件主控模式
- I²C 从动模式（10 位地址），使能起始和停止位中断，支持固件主控模式
- 使能 I²C 的开始和停止位中断，支持固件主控模式，从单元空闲

倘若将 TRISC 或 TRISD 位适当置位，使这些引脚编程为输入引脚，在 SSPEN 位置位时选择任何 I²C 模式都将强制 SCL 和 SDA 引脚漏极开路。SCL 和 SDA 引脚上必须有外部上拉电阻，才能使 I²C 模块正常工作。

有关 SSP 模块 I²C 工作原理的其他信息，可以参考《PICmicro[®] 中档单片机系列参考手册》（DS33023A_CN）。

18.3.1 从动模式

在从动模式下，必须将 SCL 和 SDA 引脚配置为输入引脚（TRISC<5:4> 或 TRISD<3:2> 置位）。需要时（从发送器）SSP 模块将用输出数据覆盖输入状态。

当地址匹配时，或接收到了匹配地址后进行数据传输时，硬件将自动产生应答（ACK）脉冲，然后将 SSPSR 寄存器当前接收到的值装入 SSPBUF 寄存器。

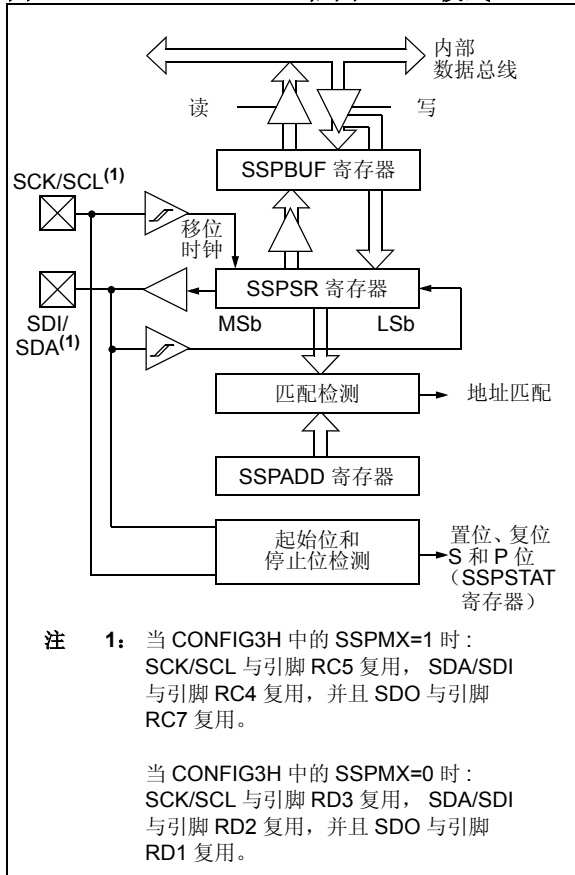
在某些情况下，SSP 模块不会发出此 ACK 脉冲信号。这些情况包括（之一或两者）：

- 缓冲器满位 BF（SSPSTAT<0>）在接收到传输数据前置位。
- 溢出位 SSPOV（SSPCON<6>）在接收到传输数据前置位。

在这种情况下，SSPSR 寄存器的值没有装入 SSPBUF，但 SSPIF 位（PIR1<3>）已置位。表 18-2 给出了根据 BF 和 SSPOV 位的状态，接收到数据传输字节时发生的情况。阴影单元格表示用户软件没有正确将溢出状态清零时的情况。当 SSPOV 位软件清零时，通过读 SSPBUF 寄存器可以将标志位 BF 清零。

为确保正常工作，SCL 时钟输入必须满足最小高电平时间和最小低电平时间。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 SSP 模块的要求。

图 18-5: SSP 框图 (I²C 模式)



SSP 模块有 5 个寄存器用于 I²C 工作。它们是：

- SSP 控制寄存器（SSPCON）
- SSP 状态寄存器（SSPSTAT）
- 串行接收 / 发送缓冲器（SSPBUF）
- SSP 移位寄存器（SSPSR）：不能直接访问
- SSP 地址寄存器（SSPAD）

PIC18F2331/2431/4331/4431

18.3.1.1 寻址

SSP 模块使能后，它会等待启动条件发生。当启动条件发生后，就会向 SSPSR 寄存器移入 8 位。所有移入的位都是在时钟线（SCL）的上升沿进行采样的。寄存器 SSPSR<7:1> 的值会和 SSPADD 寄存器的值进行比较。地址会在第 8 个时钟（SCL）脉冲的下降沿进行比较。如果地址匹配，并且 BF 和 SSPOV 位清零，将发生下列事件：

- SSPSR 寄存器的值装入 SSPBUF 寄存器。
- 缓冲器满位 BF 置位。
- 产生 $\overline{\text{ACK}}$ 脉冲。
- 在第 9 个 SCL 脉冲的下降沿，SSP 中断标志位 SSPIF（PIR1<3>）置位（如果允许中断则发生中断）。

在 10 位地址模式下，从单元需要接收两个地址字节（图 18-7）。第一个地址字节的五个最高有效数位（MSb）用以说明它是否为 10 位地址。R/W 位（SSPSTAT<2>）必须指定写操作，这样从器件才能收到第二个地址字节。对于 10 位地址，第一个字节等于 ‘1111 0 A9 A8 0’，其中 A9 和 A8 是地址的两个 MSb。

10 位地址的事件顺序如下，其中第 7 步到第 9 步所述为从发送器：

- 接收地址的第一个（高）字节（SSPIF 位、BF 位和 UA 位（SSPSTAT<1>）置位）。
- 用地址的第二个（低）字节更新 SSPADD 寄存器（UA 位清零并释放 SCL 线）。
- 读 SSPBUF 寄存器（BF 位清零），并将标志位 SSPIF 清零。
- 接收地址的第二个（低）字节（SSPIF 位、BF 位和 UA 位置位）。
- 使用地址的第一个（高）字节更新 SSPADD 寄存器。如果匹配释放 SCL 线，则 UA 位清零。
- 读 SSPBUF 寄存器（BF 位清零），并将标志位 SSPIF 清零。
- 接收重复启动条件。
- 接收地址的第一个（高）字节（SSPIF 位和 BF 位置位）。
- 读 SSPBUF 寄存器（BF 位清零），并将标志位 SSPIF 清零。

表 18-2: 数据传输已接收字节操作

接收到传输数据时的状态位		SSPSR → SSPBUF	产生 $\overline{\text{ACK}}$ 脉冲	SSPIF 位置位 (如 SSP 中断使能, 则发生该中断)
BF	SSPOV			
0	0	是	是	是
1	0	否	否	是
1	1	否	否	是
0	1	否	否	是

注： 阴影单元格表示用户软件没有正确将溢出状态清零时的情况。

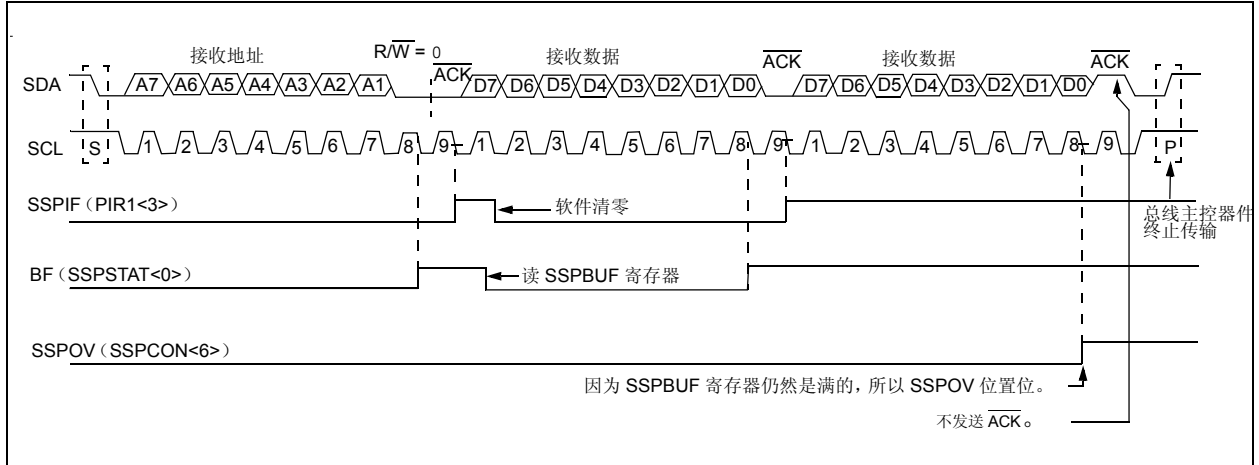
18.3.1.2 接收

当地址字节的 $\overline{\text{R/W}}$ 位清零并发生地址匹配时，SSPSTAT 寄存器的 R/W 位清零。接收到的地址装入 SSPBUF 寄存器。

当地址字节溢出时，不会产生应答（ $\overline{\text{ACK}}$ ）脉冲。溢出状态定义为 BF 位（SSPSTAT<0>）置位或 SSPOV 位（SSPCON<6>）置位的状态。这是因用户固件造成的一个错误状况。

每个数据传输字节都会产生 SSP 中断。标志位 SSPIF（PIR1<3>）必须用软件清零。SSPSTAT 寄存器用于确定该字节的状态。

图 18-6: I²C 接收波形 (7 位地址)



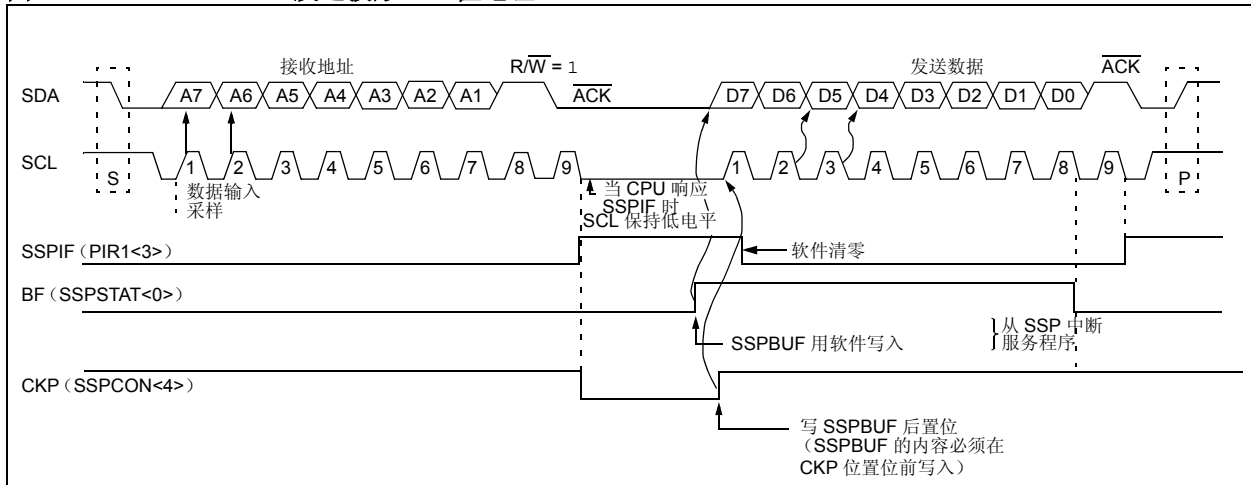
18.3.1.3 发送

当进入的地址字节之 $\overline{R/W}$ 位置位并发生地址匹配时, SSPSTAT 寄存器的 $\overline{R/W}$ 位置位。接收到的地址装入 SSPBUF 寄存器。ACK 脉冲将在第 9 位发送, 且 SCK/SCL 引脚保持低电平。发送数据必须装入 SSPBUF 寄存器, 同时也装入 SSPSR 寄存器。然后, 应通过将 CKP (SSPCON<4>) 位置位使能 SCK/SCL 引脚。主单元必须在判定另一个时钟脉冲前监视 SCL 引脚。从器件可以通过时钟延长, 从而不与主器件同步。8 个数据位在 SCL 输入的下降沿移出。这样做, 可以保证 SDA 信号在 SCL 高电平时有效 (图 18-7)。

每个数据传输字节都会产生 SSP 中断。标志位 SSPIF 必须用软件清零, SSPSTAT 寄存器用于判断该字节的状态。标志位 SSPIF 在第 9 个时钟脉冲的下降沿置位。

作为从发送器, 主接收器的 ACK 脉冲在第 9 个 SCL 输入脉冲的上升沿锁存。如果 SDA 线是高电平 (非 ACK), 则完成数据传输。当 ACK 由从器件锁存时, 从器件逻辑复位 (SSPSTAT 寄存器复位), 然后从器件监控另一个起始位出现。如果 SDA 线为低电平 (ACK), 发送数据必须装入 SSPBUF 寄存器, 同时也装入 SSPSR 寄存器。然后, 应通过将 CKP 位置位使能 SCK/SCL 引脚。

图 18-7: I²C 发送波形 (7 位地址)



PIC18F2331/2431/4331/4431

18.3.2 主控模式

利用在检测到启动和停止条件时产生的中断，固件可以支持主控模式。停止（P）位和起始（S）位在复位时或禁止 SSP 模块时清零。停止（P）位和开始（S）位会根据启动和停止条件切换。当 P 位置位时，或总线处于空闲状态且 S 和 P 位均清零时，就可以取得 I²C 总线的控制权。

在 主控模式下，SCL 和 SDA 线是通过将相应 TRISC<5:4> 或 TRISD<3:2> 位清零控制的。不管 PORTC<5:4> 或 PORTD<3:2> 的值如何，输出电平始终为低。因此当发送数据时，为 1 的数据位必须将 TRISC<4> 位置位（输入），而为 0 的数据位则将 TRISC<4> 位清零（输出）。对于 SCL 线以及 TRISC<4> 和 TRISD<2> 位也应做类似处理。SCL 和 SDA 引脚上必须外接上拉电阻，I²C 模块才能正常工作。

以下事件将导致 SSP 中断标志位 SSPIF 置位（如果使能 SSP 中断，则发生此中断）：

- 启动条件
- 停止条件
- 数据传输字节已发送 / 已接收

可以在从动模式空闲（SSPM3:SSPM0=1011）或从动模式有效状态完成主控模式操作。当主控模式和从动模式同时使能时，软件需要区分中断源。

18.3.3 多主模式

在多主模式下，在检测到启动和停止条件时产生中断，这可以用于判断总线是否空闲。停止（P）位和起始（S）位在复位时或禁止 SSP 模块时清零。停止（P）位和起始（S）位会根据启动和停止条件切换。当 P 位（SSPSTAT<4>）置位时，或总线处于空闲状态且 S 位和 P 位均清零时，就可以取得 I²C 总线的控制权。当总线忙时，如果出现停止条件，则使能 SSP 中断会产生中断。

在多主操作中，必须监视 SDA 线，以确定信号电平是否为预期的输出电平。此检测仅需在输出为高电平时进行。如果预期为高电平输出，但检测到了低电平，器件必须释放 SDA 和 SCL 线（TRISC<5:4> 或 TRISD<3:2> 置位）。此仲裁在以下两个阶段会失败：

- 地址传输
- 数据传输

当从动逻辑使能时，从动器件将继续接收。如果在地址传输阶段仲裁失败，与该器件的通讯将继续进行。如果被寻址，将产生 ACK 脉冲。如果在数据传输阶段仲裁失败，器件需要稍后重新传输数据。

表 18-3: 与 I²C 工作模式相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
SSPBUF	同步串行端口接收缓冲器 / 发送寄存器								xxxx xxxx	uuuu uuuu
SSPADD	同步串行端口（I ² C 模式）地址寄存器								0000 0000	0000 0000
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
SSPSTAT	SMP ⁽²⁾	CKE ⁽²⁾	D/A	P	S	R/W	UA	BF	0000 0000	0000 0000
TRISC ⁽³⁾	PORTC 数据方向寄存器								1111 1111	1111 1111
TRISD ⁽³⁾	PORTD 数据方向寄存器								1111 1111	1111 1111

图注： x= 未知， u= 不变， -= 未实现，读作 0。阴影单元格表示 SSP 模块在 I²C 模式下未使用。

- 注 1: 在 PIC16F73/76 中保留了 PSPIE 和 PSPIF 位；应始终保持这些位清零。
- 注 2: 在 I²C 模式下保持这些位清零。
- 注 3: 根据 CONFIG3H 中 SSPMX 的设置，这些引脚与 PORTC 或 PORTD 复用。

19.0 增强型通用同步 / 异步收发器

增强型通用同步 / 异步收发器（Enhanced Universal Synchronous Asynchronous Receiver Transmitter, EUSART）模块是 PIC18F2331/2431/4331/4431 系列单片机具有的两个串行 I/O 模块之一。EUSART 也称为“串行通讯接口”（即 SCI）。

可以将 EUSART 配置为能与 CRT 终端和个人计算机等外设通讯的全双工异步系统。也可以将它配置成能够与 A/D 或 D/A 集成电路、串行 EEPROM 等外设通讯的半双工同步系统。

EUSART 模块还实现了其他功能，包括自动波特率检测和校准、接收到同步间隔字符时的自动唤醒和 12 位间隔字符发送。因为具有这些功能，所以局域互连网络（Local Interconnect Network, LIN）总线系统使用 EUSART 模块非常理想。

USART 可配置为以下几种工作模式：

- 异步模式（全双工）：
 - 接收到字符时自动唤醒
 - 自动波特率校准
 - 12 位间隔字符发送
- 同步——主控（半双工）模式，带可选时钟极性
- 同步——从动（半双工）模式，带可选时钟极性

为了将 RC6/TX/CK/SS 引脚和 RC7/RX/DT/SDO 引脚配置为通用同步 / 异步收发器：

- SPEN（RCSTA<7>）位必须置位（=1），
- TRISC<6> 位必须置位（=1），并且
- TRISC<1> 位必须置位（=1）。

注： USART 控制电路会根据需要自动将引脚从输入重新配置为输出。

增强型 USART 模块的工作状况是通过 3 个寄存器控制的：

- 发送状态和控制（TXSTA）
- 接收状态和控制（RCSTA）
- 波特率控制（BAUDCTL）

后续几页的寄存器 19-1、寄存器 19-2 和寄存器 19-3 分别对它们进行了详细的介绍。

19.1 功耗管理模式下的异步工作原理

当外设时钟由内部振荡电路提供时，USART 可以在异步模式下工作。因此可以去掉 OSC1 和 OSC2 引脚上通常作为主时钟连接的晶振或谐振器。

工厂会将内部振荡电路输出（INTOSC）校准为 8 MHz（参见表 25-6）。但是，在 VDD 或温度变化时，此频率可能会发生漂移，这将直接影响异步波特率。两种方法可用于调整波特率时钟，但这两种方法都需要某种类型的参考时钟源。

第一种（建议）方法使用 OSCTUNE 寄存器将 INTOSC 输出调整回 8 MHz。调整 OSCTUNE 寄存器的值可以微调系统时钟源（如需更多信息，请参见第 3.6 节“INTOSC 频率漂移”）。

另一个方法是调整波特率发生器的值。当调整波特率发生器时，其微调分辨率可能不足以补偿外设时钟频率的渐进变化。

PIC18F2331/2431/4331/4431

寄存器 19-1: TXSTA: 发送状态和控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
						bit 7	bit 0

bit 7 **CSRC:** 时钟源选择位

异步模式:

忽略

同步模式:

1 = 主控模式 (时钟来自内部 BRG)

0 = 从动模式 (时钟来自外部时钟源)

bit 6 **TX9:** 9 位发送使能位

1 = 选择 9 位发送

0 = 选择 8 位发送

bit 5 **TXEN:** 发送使能位

1 = 使能发送

0 = 禁止发送

注: 同步模式下 SREN/CREN 会覆盖 TXEN 的值。

bit 4 **SYNC:** USART 模式选择位

1 = 同步模式

0 = 异步模式

bit 3 **SENDB:** 发送间隔字符位

异步模式:

1 = 在下次发送时发送同步间隔字符 (完成后由硬件清零)

0 = 同步间隔发送完成

同步模式:

忽略

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。

图注:

R= 可读位

W= 可写位

U= 未实现位, 读作 0

-n= 上电复位时的值

1= 置位

0= 清零

x= 未知位

PIC18F2331/2431/4331/4431

寄存器 19-2: RCSTA: 接收状态和控制寄存器

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

- bit 7 **SPEN:** 串行端口使能位
 1 = 使能串行端口 (配置 RX/DT 和 TX/CK 引脚为串行端口引脚)
 0 = 禁止串行端口 (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:** 单字接收使能位
异步模式:
 忽略
同步主控模式:
 1 = 使能单字接收
 0 = 禁止单字接收
 此位在接收完成后清零。
同步从动模式:
 忽略
- bit 4 **CREN:** 连续接收使能位
异步模式:
 1 = 使能接收器
 0 = 禁止接收器
同步模式:
 1 = 使能连续接收, 直到使能位 CREN 清零 (CREN 覆盖 SREN)
 0 = 禁止连续接收
- bit 3 **ADDEN:** 地址检测使能位
9 位异步模式 (RX9=1):
 1 = 使能地址检测, 允许中断, 当 RSR<8> 置位时装入接收缓冲器
 0 = 禁止地址检测, 接收所有字节并且第 9 位可作为奇偶校验位
8 位异步模式 (RX9=0):
 忽略
- bit 2 **FERR:** 帧错误位
 1 = 帧错误 (可通过读 RCREG 寄存器刷新该位, 并接收下一个有效字节)
 0 = 没有帧错误
- bit 1 **OERR:** 溢出错误位
 1 = 溢出错误 (可以通过清零 CREN 位来清零该位)
 0 = 没有溢出错误
- bit 0 **RX9D:** 接收数据的第 9 位
 这可以是地址 / 数据位或奇偶校验位, 必须由用户固件计算。

图注:

R= 可读位	W= 可写位	U= 未实现位, 读作 0
-n= 上电复位时的值	1= 置位	0= 清零
		x= 未知位

PIC18F2331/2431/4331/4431

寄存器 19-3: BAUDCTL: 波特率控制寄存器

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

- bit 7 **未实现位:** 读作 0
- bit 6 **RCIDL:** 接收操作空闲状态位
1= 接收器空闲
0= 正在接收
- bit 5 **未实现位:** 读作 0
- bit 4 **SCKP:** 同步时钟极性选择位
异步模式:
在此模式下未使用
同步模式:
1= 时钟 (CK) 为高电平时处于空闲状态
0= 时钟 (CK) 为低电平时处于空闲状态
- bit 3 **BRG16:** 16 位波特率寄存器使能位
1=16 位波特率发生器——SPBRGH 和 SPBRG
0= 8 位波特率发生器——仅 SPBRG (兼容模式), 忽略 SPBRGH 值
- bit 2 **未实现位:** 读作 0
- bit 1 **WUE:** 唤醒使能位
异步模式:
1= USART 将继续采样该 RX 引脚——中断在下降沿产生; 在下一个上升沿用硬件清零该位
0= 未检测 RX 引脚或检测到了上升沿
同步模式:
在此模式下未使用
- bit 0 **ABDEN:** 自动波特率检测使能位
异步模式:
1= 在下一个字符使能波特率测量——需要接收同步字段 (55h); 完成后用硬件清零
0= 波特率测量被禁止或已完成
同步模式:
在此模式下未使用

图注:

R= 可读位	W= 可写位	U= 未实现位, 读作 0
-n= 上电复位时的值	1= 置位	0= 清零
		x= 未知位

19.2 USART 波特率发生器 (BRG)

BRG 是一个专用的 8 位或 16 位发生器，支持 USART 的异步和同步模式。缺省情况下，BRG 工作在 8 位模式下，通过置位 BRG16 位 (BAUDCTL<3>) 可以选择 16 位模式。

SPBRGH:SPBRG 寄存器对控制自由运行的定时器周期。在异步模式，BRGH (TXSTA<2>) 和 BRG16 位也控制波特率。在同步模式，会忽略 BRGH 位。表 19-1 所示为不同 USART 模式的波特率计算公式，但仅适用于主控模式 (内部生成时钟信号)。

给出期望的波特率和 Fosc 值，就可以使用表 19-1 中的公式计算出 SPBRGH:SPBRG 寄存器的最近似整数值。这样就可以确定波特率的误差。例 19-1 给出了一个计算示例。表 19-2 中给出了不同异步模式下的典型波特率和误差值。这对于使用高波特率 (BRGH=1) 或 16 位 BRG 可以减小波特率误差，或实现快速振荡频率下的低波特率可能是很有用的。

向 SPBRGH:SPBRG 寄存器写入新值会使 BRG 定时器复位 (或清零)。这可以确保 BRG 不必在定时器溢出之后才输出新波特率。

19.2.1 功耗管理模式工作原理

系统时钟用于产生所需的波特率，但是当进入功耗管理模式时，时钟源的工作频率可能不同于 PRI_RUN 模式下的工作频率。休眠状态不使用任何时钟，而在 PRI_IDLE 状态下，主时钟源将继续为波特率发生器提供时钟信号，但是，在其他功耗管理模式下，时钟频率可能会发生变化。可能需要调整 SPBRG 中的值。

如果系统时钟在主动接收操作过程中发生变化，可能会导致接收错误或数据丢失。为了避免此问题，应该检查 RCIDL 位的状态并确保在改变系统时钟前接收处于空闲状态。

19.2.2 采样

主要检测电路会对 RC7/RX/DT/SDO 引脚上的数据采样 3 次，以确定在 RX 引脚上是出现高电平还是低电平。

表 19-1: 波特率计算公式

配置位			BRG / USART 模式	波特率计算公式
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 = 忽略, n=SPBRGH:SPBRG 寄存器对的值

PIC18F2331/2431/4331/4431

例 19-1: 计算波特率误差

如果某个器件 F_{osc} 为 16MHz、波特率期望值为 9600、异步工作模式、8 位 BRG:
 波特率期望值 = $F_{osc} / (64 ([SPBRGH:SPBRG] + 1))$
 求解 SPBRGH:SPBRG:

$$X = ((F_{osc} / \text{Desired Baud Rate}) / 64) - 1$$

$$= ((16000000 / 9600) / 64) - 1$$

$$= [25.042] = 25$$
 波特率计算值 = $16000000 / (64 (25 + 1))$

$$= 9615$$
 误差 = $(\text{波特率计算值} - \text{波特率期望值}) / \text{波特率期望值}$

$$= (9615 - 9600) / 9600 = 0.16\%$$

表 19-2: 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	POR 和 BOR 时的值	所有其他复位时的值
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 -010	0000 -010
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
BAUDCTL	—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-1 0-00	-1-1 0-00
SPBRGH	波特率发生器寄存器的高位字节								0000 0000	0000 0000
SPBRG	波特率发生器寄存器的低位字节								0000 0000	0000 0000

图注: x= 未知位, -= 未实现位, 读作 0。阴影单元格表示不适用于 BRG。

表 19-3: 异步模式波特率

目标波特率 (Kbps)	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8,000 MHz		
	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

目标波特率 (Kbps)	SYNC = 0, BRGH = 0, BRG16 = 0								
	Fosc = 4,000 MHz			Fosc = 2,000 MHz			Fosc = 1,000 MHz		
	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)
0.3	0.300	0.16	207	300	-0.16	103	300	-0.16	51
1.2	1.202	0.16	51	1201	-0.16	25	1201	-0.16	12
2.4	2.404	0.16	25	2403	-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	-	0	—	—	—	—	—	—

PIC18F2331/2431/4331/4431

表 19-3: 异步模式波特率 (续)

目标波特率 (Kbps)	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20,000 MHz			Fosc = 10.000 MHz			Fosc = 8,000 MHz		
	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)
2.4	—	—	—	—	—	—	2.441	1.73	255	2403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

目标波特率 (Kbps)	SYNC = 0, BRGH = 1, BRG16 = 0								
	Fosc = 4,000 MHz			Fosc = 2,000 MHz			Fosc = 1,000 MHz		
	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)
0.3	—	—	—	—	—	—	300	-0.16	207
1.2	1.202	0.16	207	1201	-0.16	103	1201	-0.16	51
2.4	2.404	0.16	103	2403	-0.16	51	2403	-0.16	25
9.6	9.615	0.16	25	9615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

目标波特率 (Kbps)	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20,000 MHz			Fosc = 10.000 MHz			Fosc = 8,000 MHz		
	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)
0.3	0.300	0.00	8332	0.300	0.02	4165	0.300	0.02	2082	300	-0.04	1665
1.2	1.200	0.02	2082	1.200	-0.03	1041	1.200	-0.03	520	1201	-0.16	415
2.4	2.402	0.06	1040	2.399	-0.03	520	2.404	0.16	259	2403	-0.16	207
9.6	9.615	0.16	259	9.615	0.16	129	9.615	0.16	64	9615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

目标波特率 (Kbps)	SYNC = 0, BRGH = 0, BRG16 = 1								
	Fosc = 4,000 MHz			Fosc = 2,000 MHz			Fosc = 1,000 MHz		
	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)
0.3	0.300	0.04	832	300	-0.16	415	300	-0.16	207
1.2	1.202	0.16	207	1201	-0.16	103	1201	-0.16	51
2.4	2.404	0.16	103	2403	-0.16	51	2403	-0.16	25
9.6	9.615	0.16	25	9615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

PIC18F2331/2431/4331/4431

表 19-3: 异步模式波特率 (续)

目标 波特率 (Kbps)	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20,000 MHz			Fosc = 10.000 MHz			Fosc = 8,000 MHz		
	实际 波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际 波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际 波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际 波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)
0.3	0.300	0.00	33332	0.300	0.00	16665	0.300	0.00	8332	300	-0.01	6665
1.2	1.200	0.00	8332	1.200	0.02	4165	1.200	0.02	2082	1200	-0.04	1665
2.4	2.400	0.02	4165	2.400	0.02	2082	2.402	0.06	1040	2400	-0.04	832
9.6	9.606	0.06	1040	9.596	-0.03	520	9.615	0.16	259	9615	-0.16	207
19.2	19.193	-0.03	520	19.231	0.16	259	19.231	0.16	129	19230	-0.16	103
57.6	57.803	0.35	172	57.471	-0.22	86	58.140	0.94	42	57142	0.79	34
115.2	114.943	-0.22	86	116.279	0.94	42	113.636	-1.36	21	117647	-2.12	16

目标 波特率 (Kbps)	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1								
	Fosc = 4,000 MHz			Fosc = 2,000 MHz			Fosc = 1,000 MHz		
	实际 波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际 波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)	实际 波特率 (Kbps)	误差 (%)	SPBRG 值 (十进制)
0.3	0.300	0.01	3332	300	-0.04	1665	300	-0.04	832
1.2	1.200	0.04	832	1201	-0.16	415	1201	-0.16	207
2.4	2.404	0.16	415	2403	-0.16	207	2403	-0.16	103
9.6	9.615	0.16	103	9615	-0.16	51	9615	-0.16	25
19.2	19.231	0.16	51	19230	-0.16	25	19230	-0.16	12
57.6	58.824	2.12	16	55555	3.55	8	—	—	—
115.2	111.111	-3.55	8	—	—	—	—	—	—

19.2.3 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。该功能仅在异步模式下有效，此时 WUE 位清零。

只要接收到起始位并且 AEDEN 位已置位就会开始自动波特率测量序列（图 19-1）。此计算是自平均的。

在自动波特率检测（Auto-Baud Rate Detect, ABD）模式下，BRG 的时钟信号反向。不是由 BRG 为进入的 RX 信号提供时钟信号，而是由 RX 信号为 BRG 定时。在 ABD 模式下，内部波特率发生器作为计数器计算进入的串行字节流的位周期。

ABDEN 位置位后，状态机将 BRG 清零并寻找起始位。要计算正确的比特率，自动波特率检测必须接收到值为 55h 的字节（ASCII 码为 U，也是 LIN 总线的同步字符）。为了尽量减小进入信号不对称造成的影响，测量需要一个低位时间和一个高位时间。在起始位后，SPBRG 使用预先选定的时钟源在 RX 的第一个上升沿开始计数。在 RX 引脚传输了 8 位以后，即在第 5 个上升沿时，将相应 BRG 周期累加，并将结果保存在 SPBRGH:SPBRG 寄存器中。一旦出现第 5 个上升沿（应该对应于停止位），ABDEN 位会自动清零。

当校准波特率周期时，BRG 寄存器时钟源为预配置时钟频率的 1/8。请注意 BRG 时钟将由 BRG16 和 BRGH 位配置。不管 BRG16 如何设置，SPBRG 和 SPBRGH 都是 16 位计数器。通过检查 SPBRGH 寄存器值是否为 00h，用户可以验证 8 位模式下是否发生进位。参见表 19-4 了解 BRG 计数器时钟频率。

当发生 ABD 序列时，USART 状态机保持在空闲状态。一旦在 RX 上检测到第 5 个上升沿 RCIF 中断就会置位。要清除 RCIF 中断，需要读取 RCREG 中的值。RCREG 的内容应该被丢弃。

注 1: 如果 WUE 位与 ABDEN 位一起置位，自动波特率检测会在间隔字符之后的字节开始。（参见第 19.3.4 节“接收到同步间隔字符时自动唤醒”）。

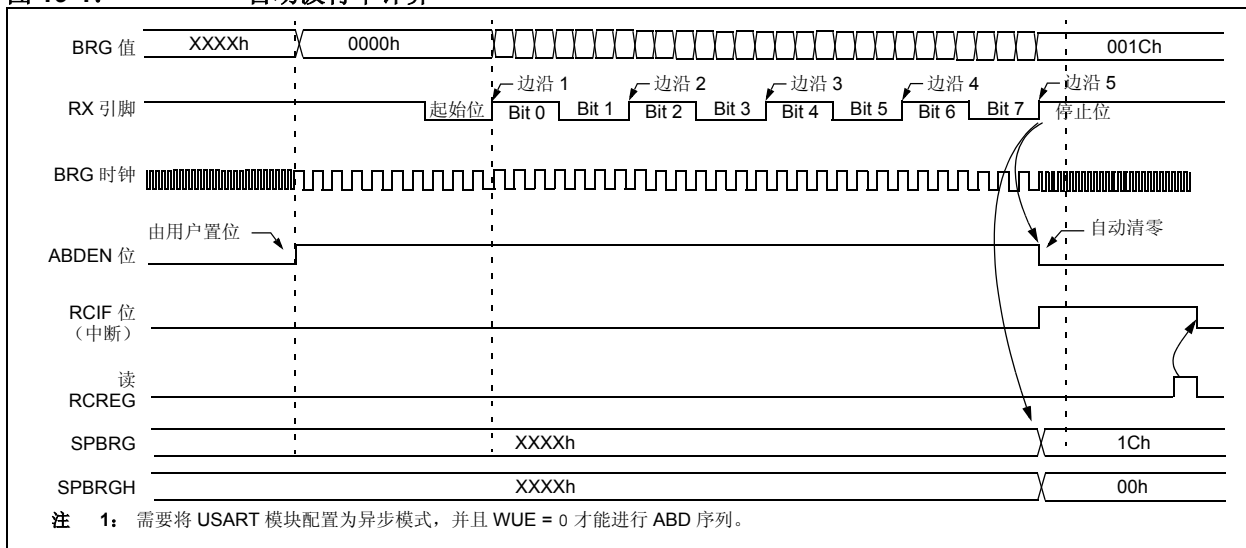
注 2: 判断进入字符波特率是否处于所选 BRG 时钟源范围内是由用户决定的。由于位误差率的原因，无法实现某些振荡频率和 USART 波特率的组合。在使用自动波特率检测功能时，必须考虑系统总体时序和通讯波特率。

表 19-4: BRG 计数器时钟频率

BRG16	BRGH	BRG 计数器时钟
0	0	Fosc/512
0	1	Fosc/256
1	0	Fosc/128
1	1	Fosc/32

注: 在 ABD 序列中，不管 BRG16 如何设置，SPBRG 和 SPBRGH 都是 16 位计数器。

图 19-1: 自动波特率计算



PIC18F2331/2431/4331/4431

19.3 USART 异步模式

通过将 SYNC 位 (TXSTA<4>) 清零可选择异步工作模式。在此模式下, USART 使用标准的“不归零”(non-return-to-zero, NRZ) 格式 (1 个起始位, 8 个或 9 个数据位, 1 个停止位)。最常用的数据格式为 8 位。片上专用 8 位 / 16 位波特率发生器可用于从振荡器产生标准波特率频率。

USART 首先发送和接收 LSB。USART 的发送器和接收器独立工作, 但使用相同的数据格式和波特率。根据 BRGH 和 BRG16 位 (TXSTA<2> 和 BAUDCTL<3>) 的设置情况, 波特率发生器会产生一个 16 倍频或 64 倍频移位速率的时钟信号。硬件不支持奇偶校验, 但可以用软件实现, 校验位保存在第 9 个数据位。

在所有的低功耗模式中都可以使用异步模式, 但在休眠模式下, 只有使能了“同步间隔自动唤醒”时才能使用异步模式。在 PRI_IDLE 模式下, 不要求改变波特率发生器值, 但是其他低功耗模式时钟的工作频率可能与主时钟不同。因此, 可能还是需要调整波特率发生器的值。

当工作在异步模式时, USART 模块包括以下主要部分:

- 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

19.3.1 USART 异步发送器

图 19-2 所示为 USART 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 / 写发送缓冲器 TXREG 获取数据。TXREG 寄存器要用软件装入数据。在前一次装入数据的停止位发送前, 不会向 TSR 寄存器装入数据。一旦发送了停止位, 就会将 TXREG 寄存器的新数据 (如果有的话) 装入 TSR。

TXREG 寄存器向 TSR 寄存器传输数据后 (在 1 个 Tcy 内发生), TXREG 寄存器为空, 同时标志位 TXIF (PIR1<4>) 置位。通过置位 / 清零使能位 TXIE (PIE1<4>), 该中断可以被使能 / 禁止。不管使能位 TXIE 的状态如何, 标志位 TXIF 都将置位并且无法用软件清零。在加载发送缓冲寄存器 TXREG 时, 标志位 TXIF 不会立即清零。TXIF 在装载指令后的第 2 个指令周期生效。在装入 TXREG 后立即查询 TXIF 会得到无效结果。

标志位 TXIF 表示的是 TXREG 寄存器的状态, 而另一个位 TRMT (TXSTA<1>) 则表示 TSR 寄存器的状态。状态位 TRMT 是只读位, 当 TSR 寄存器为空时置位。因为没有与此位关联的中断逻辑, 所以用户必须查询此位以判断 TSR 寄存器是否为空。

- | |
|--|
| <p>注 1: TSR 寄存器不映射到数据存储器, 因此用户无法使用它。</p> <p>2: 当使能位 TXEN 置位时, 标志位 TXIF 置位。</p> |
|--|

要建立异步发送:

1. 用适当的波特率初始化 SPBRGH:SPBRG 寄存器。按需要将 BRGH 和 BRG16 位置位或清零, 以获得所需的波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置位使能异步串行端口。
3. 如果需要中断, 将使能位 TXIE 置位。
4. 如果需要 9 位发送, 将发送位 TX9 置位。可以作为地址 / 数据位使用。
5. 通过将 TXEN 位置位使能发送, 此操作同时也会将 TXIF 位置位。
6. 如果选择了 9 位发送, 应该将第 9 位装入 TX9D 位。
7. 将数据装入 TXREG 寄存器 (开始发送)。

如果使用中断, 应确保 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 已置位。

图 19-2: USART 发送框图

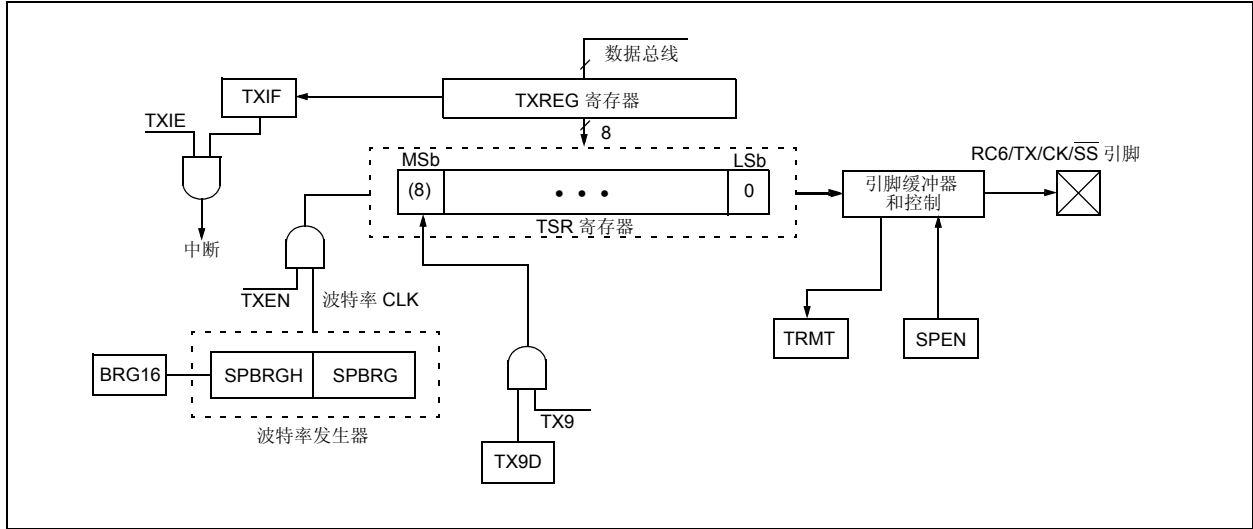


图 19-3: 异步发送

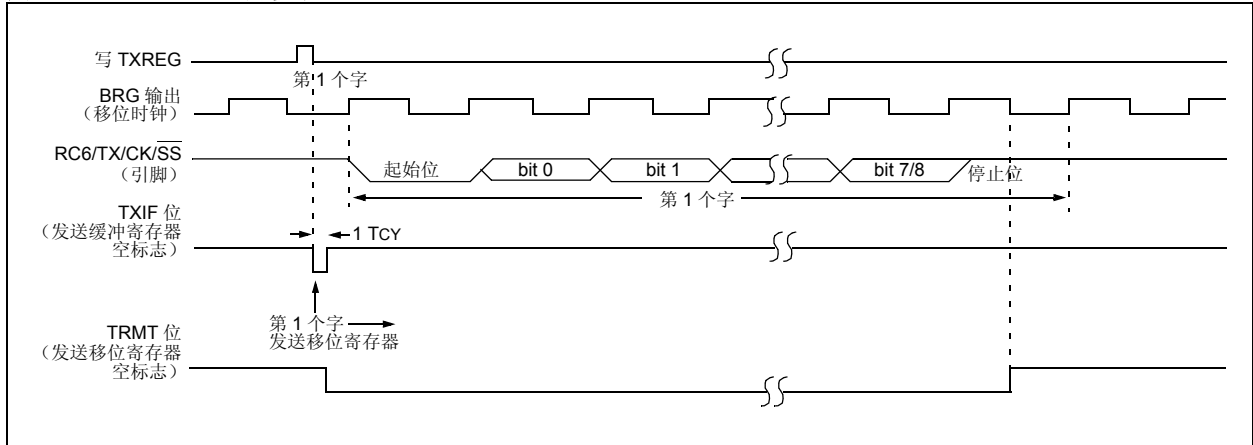
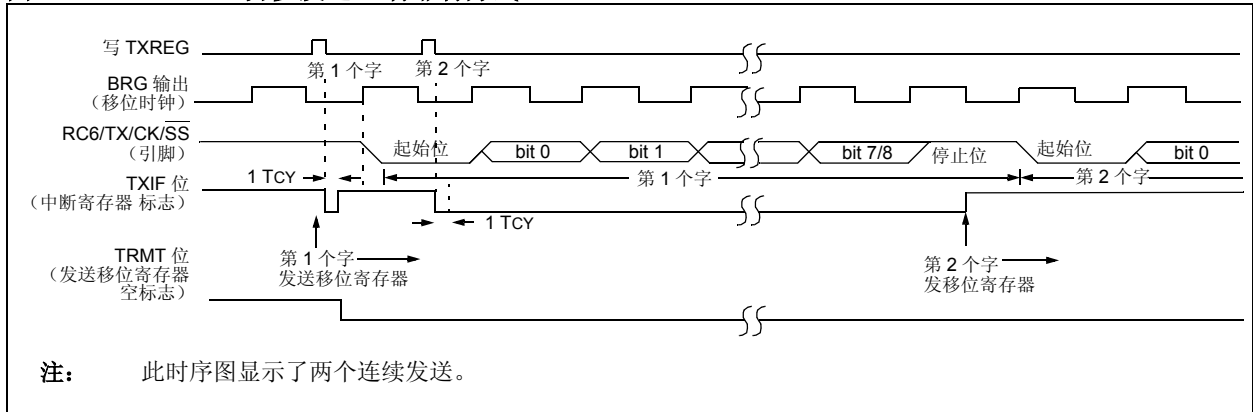


图 19-4: 异步发送（背靠背方式）



PIC18F2331/2431/4331/4431

表 19-5: 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	POR 和 BOR 时的值	所有其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	-000 -000	-000 -000
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 -000	-000 -000
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	-111 -111	-111 -111
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
TXREG	USART 发送寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
BAUDCTL	—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-1 0-00	-1-1 0-00
SPBRGH	波特率发生器寄存器的高位字节								0000 0000	0000 0000
SPBRG	波特率发生器寄存器的低位字节								0000 0000	0000 0000

图注: x= 未知, -= 未实现的地址单元, 读作 0。阴影单元格表示在异步发送模式下未使用。

19.3.2 USART 异步接收器

图 19-5中显示了接收器框图。数据在RC7/RX/DT/SDO引脚接收，并驱动数据恢复电路。数据恢复电路实际上是一个高速移位器，工作在波特率的 16 倍频下，而主接收串行移位器工作在比特率或 Fosc 下。此模式通常用于 RS-232 系统。

要建立异步接收：

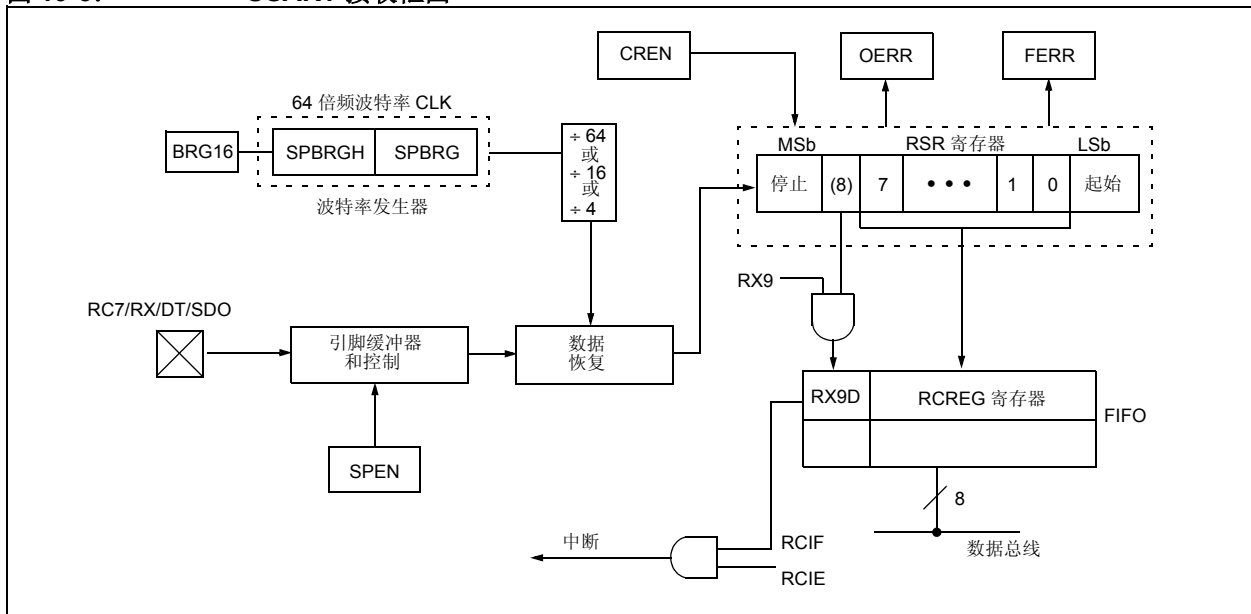
1. 用适当的波特率初始化 SPBRGH:SPBRG 寄存器。按需要将 BRGH 和 BRG16 位置位或清零，以获得所需的波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置位使能异步串行端口。
3. 如果需要中断，将使能位 RXIE 置位。
4. 如果需要 9 位接收，将 RX9 位置位。
5. 通过将 CREN 位置位使能接收。
6. 当接收完成时 RCIF 位将置位，并且如果使能位 RCIE 置位，会产生一个中断。
7. 读 RCSTA 寄存器以获取第 9 位（如果使能）并判断是否在接收过程中发生了错误。
8. 读 RCREG 寄存器来读取 8 位接收数据。
9. 如果发生错误，通过将使能位 CREN 清零清除错误。
10. 如果使用中断，应确保 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）已置位。

19.3.3 建立带地址检测的 9 位模式

此模式通常用于 RS-485 系统。要建立使能地址检测的异步接收：

1. 用适当的波特率初始化 SPBRGH:SPBRG 寄存器。按需要将 BRGH 和 BRG16 位置位或清零，以获得所需的波特率。
2. 通过将 SYNC 位清零并将 SPEN 位置位使能异步串行端口。
3. 如果需要中断，将 RCEN 位置位并用 RCIP 位选择所需的优先级别。
4. 将 RX9 位置位以使能 9 位接收。
5. 将 ADDEN 位置位使能地址检测。
6. 将 CREN 位置位使能接收。
7. 接收完成时 RCIF 位会置位。如果 RCIE 和 GIE 位置位，则将应答中断。
8. 读 RCSTA 寄存器和数据的第 9 位（如适用）以确定是否在接收时发生了错误。
9. 读 RCREG 以判断是否正在寻址该器件。
10. 如果发生了错误，将 CREN 位清零。
11. 如果已寻址到器件，将 ADDEN 位清零以允许所有的接收数据进入接收缓冲器并中断 CPU。

图 19-5: USART 接收框图



PIC18F2331/2431/4331/4431

要建立异步发送：

1. 用适当的波特率初始化 SPBRG 寄存器。如果需要高速波特率，将 BRGH 位置位（参见第 19.2 节“USART 波特率发生器（BRG）”）。
2. 通过将 SYNC 位清零并将 SPEN 位置位使能异步串行端口。
3. 如果需要中断，将使能位 TXIE 置位。
4. 如果需要 9 位发送，将发送位 TX9 置位。可以作为地址 / 数据位使用。
5. 通过将 TXEN 位置位使能发送，此操作同时也会将 TXIF 位置位。
6. 如果选择了 9 位发送，应该将第 9 位装入 TX9D 位。
7. 将数据装入 TXREG 寄存器（开始发送）。

如果使用中断，应确保 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）已置位。

图 19-6: 异步接收

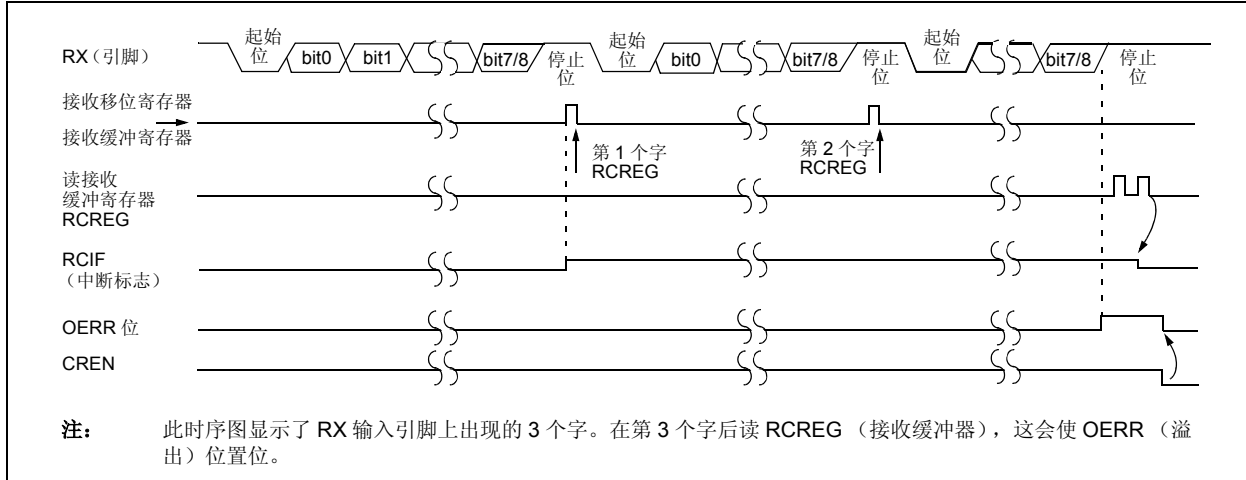


表 19-6: 与异步接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR和BOR时的值	所有其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBF	0000 000x	0000 000u
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	-000 -000	-000 -000
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 -000	-000 -000
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	-111 -111	-111 -111
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
RCREG	USART 接收寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
BAUDCTL	—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-1 0-00	-1-1 0-00
SPBRGH	波特率发生器寄存器的高位字节								0000 0000	0000 0000
SPBRG	波特率发生器寄存器的低位字节								0000 0000	0000 0000

图注：x= 未知，-= 未实现的地址单元，读作 0。阴影单元格表示在异步接收模式下未使用。

19.3.4 接收到同步间隔字符时自动唤醒

在休眠模式下，所有的 USART 时钟都会暂停。因此，波特率发生器处于不活动状态，并且无法进行正确的字节接收。当 USART 工作在异步模式下时，自动唤醒功能允许 RX/DT 线上的活动唤醒控制器。

通过将 WUE 位 (BAUDCTL<1>) 置位可以启用自动唤醒功能。置位后，将禁止 RX/DT 上的典型接收序列，并且 USART 保持在空闲状态，监视唤醒事件（不管 CPU 运行模式如何）。唤醒事件包括 RX/DT 线上一个下降沿信号。（与 LIN 协议的同步间隔或唤醒信号字符中的起始位保持一致。）

唤醒事件后，模块会产生一个 RCIF 中断。在正常工作模式下，产生的中断与 Q 时钟同步（图 19-7）；如果器件处于休眠模式，则异步（图 19-8）。通过读 RCREG 寄存器可清除中断状态。

唤醒事件后一旦 RX 线出现上升沿信号，WUE 位就会自动清零。此时，USART 模块处于空闲状态并返回正常工作模式。这就通知用户同步间隔事件已经结束。

19.3.4.1 使用自动唤醒的特别注意事项

因为自动唤醒功能是通过在 RX/DT 检测到上升沿信号执行的，任何在停止位前的状态改变都可能会产生错误的结束信号并导致数据错误或帧错误。因此，要正常工作，

发送的初始字符必须全为 0。对于标准的 RS-232 器件可以是 00h（8 位），对于 LIN 总线则是 000h（12 位）。

另外还必须考虑振荡器起振时间，尤其是在所采用振荡器的起振间隔时间（比如 LP、XT 或 HS/PLL 模式）较长的应用中。同步间隔（或唤醒信号）字符必须足够长并且之后的间隔时间也足够长，以便所选振荡器有足够的时间起振并让 USART 正确初始化。

19.3.4.2 使用 WUE 位的特别注意事项

当判断接收数据的有效性时，WUE 和 RCIF 时间的时序可能会造成某些困扰。如前所述，将 WUE 位置位会使 USART 进入空闲模式。唤醒事件通过置位 RCIF 位来产生一个接收中断。此后当 RX/DT 出现上升沿时 WUE 位清零。然后通过读 RCREG 寄存器清除中断状态。一般情况下，RCREG 中的数据是无效数据，应该丢弃。

不应该将 WUE 位清零（或仍然置位）以及 RCIF 标志位置位作为 RCREG 中数据完整性的指示。用户应该考虑用固件实现一种并行方法以验证接收数据的完整性。

要确保没有丢失有效数据，应检查 RCIDL 位以验证是否在进行接收。如果没有发生接收操作，则随后可以在进入休眠模式前将 WUE 位置位。

图 19-7: 正常工作模式下的自动唤醒位 (WUE) 时序

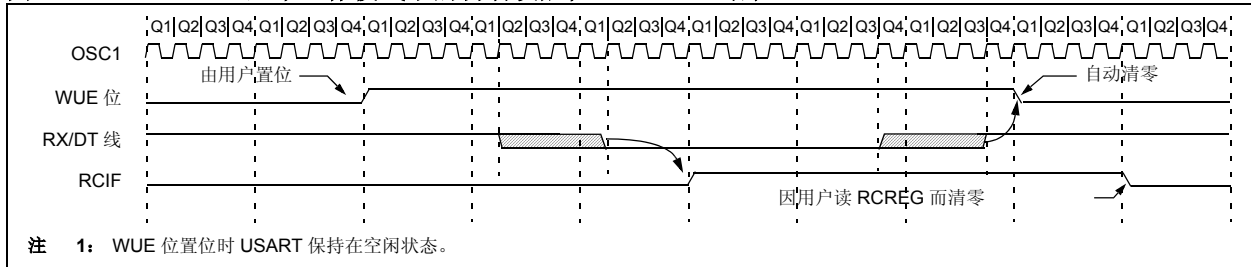
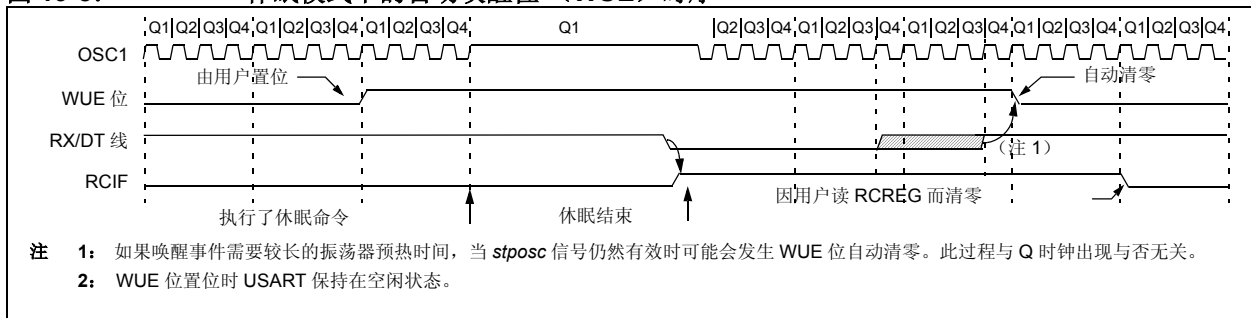


图 19-8: 休眠模式下的自动唤醒位 (WUE) 时序



PIC18F2331/2431/4331/4431

19.3.5 间隔字符序列

增强型 USART 模块能够发送 LIN 总线标准所需的特殊间隔字符序列。发送的间隔字符包括 1 个起始位、后面跟着 12 位 0 位，还有 1 个停止位。当发送移位寄存器装入数据时，只要 SENDB 和 TXEN 位 (TXSTA<3> 和 TXSTA<5>) 置位，就会发送帧间隔字符。请注意写入 TXREG 的数据值会被忽略，并作为 0 发送。

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

请注意间隔字符中写入 TXREG 的数据值会被忽略。这个写操作仅仅是为了开始正确的序列。

TRMT 位表明发送处于活动还是空闲状态，这与正常发送模式相同。关于间隔字符序列，请参见图 19-9。

19.3.5.1 间隔和同步发送序列

下列序列会发送一个报文帧头，包括一个间隔字符，后面跟着一个自动波特率同步字节。此序列适用于典型的主 LIN 总线。

1. 将 USART 配置为所需的模式。
2. 将 TXEN 和 SENDB 位置位以建立间隔字符。
3. 将无效字符装入 TXREG，开始发送 (该值会被忽略)。
4. 将 55h 写入 TXREG，以便把同步字符装入发送 FIFO 缓冲器。
5. 间隔字符发送后，硬件会将 SENDB 位复位。然后同步字符在预配置模式下开始发送。

当 TXIF 标识 TXREG 变空以后，下一个数据字节会写入 TXREG。

19.3.6 接收间隔字符

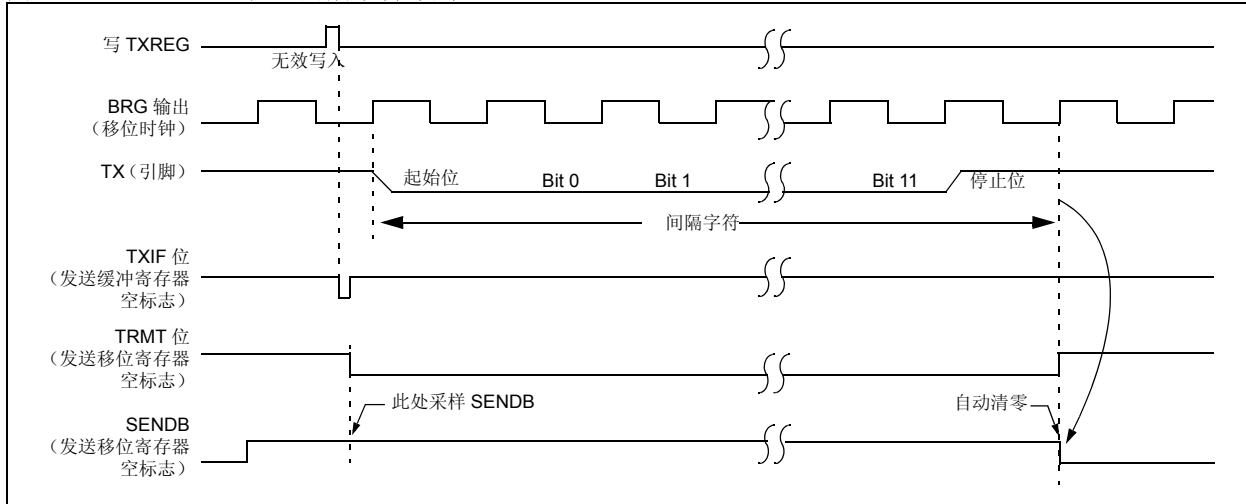
增强型 USART 模块有两种方法接收间隔字符。

第一种方法是强制将波特率配置为标准速度的 9/13。这可以使停止位在正确的采样点 (对于间隔字符和起始位是 13 位，典型数据则是 8 个数据位) 被发送。

第二种方法使用第 19.3.4 节“接收到同步间隔字符时自动唤醒”中描述的自动唤醒功能。通过使能此功能，USART 将采样 RX/DT 线的下两个发送信号，产生一个 RCIF 中断，并且接收下一个数据字节，之后再产生另一个中断。

请注意在间隔字符后，用户通常希望使能自动波特率检测功能。在这两种方法中，用户都可以在 USART 进入休眠模式前将 ABD 位置位。

图 19-9: 发送间隔字符序列



19.4 USART 同步主控模式

将 CSRC 位 (TXSTA<7>) 置位可以进入同步主控模式。在此模式中, 数据以半双工方式发送 (即发送和接收不能同时进行)。发送数据时禁止接收, 反之亦然。将 SYNC 位 (TXSTA<4>) 置位可进入同步模式。此外, 置位使能位 SPEN (RCSTA<7>), 会使 RC6/TX/CK/SS 和 RC7/RX/DT/SDO I/O 引脚分别配置为 CK (时钟) 和 DT (数据) 线。

主控模式表示处理器在 CK 线发送主时钟。时钟极性是通过 SCKP 位 (BAUDCTL<5>) 进行选择的; 将 SCKP 置位使 CK 为高电平时进入空闲状态, 将该位清零则将空闲状态置为低电平。提供此选项是为了支持 Microwire® 器件使用本模块。

19.4.1 USART 同步主控发送模式

图 19-2 所示为 USART 发送器框图。发送器的核心是发送 (串行) 移位寄存器 (TSR)。移位寄存器中的数据通过读/写发送缓冲寄存器 TXREG 获取。TXREG 寄存器要用软件装入数据。直到前一次装入的数据被发送完成才会再次向 TSR 寄存器装入数据。一旦发送了最后一位, 就会将 TXREG 寄存器的新数据 (如果有的话) 装入 TSR。

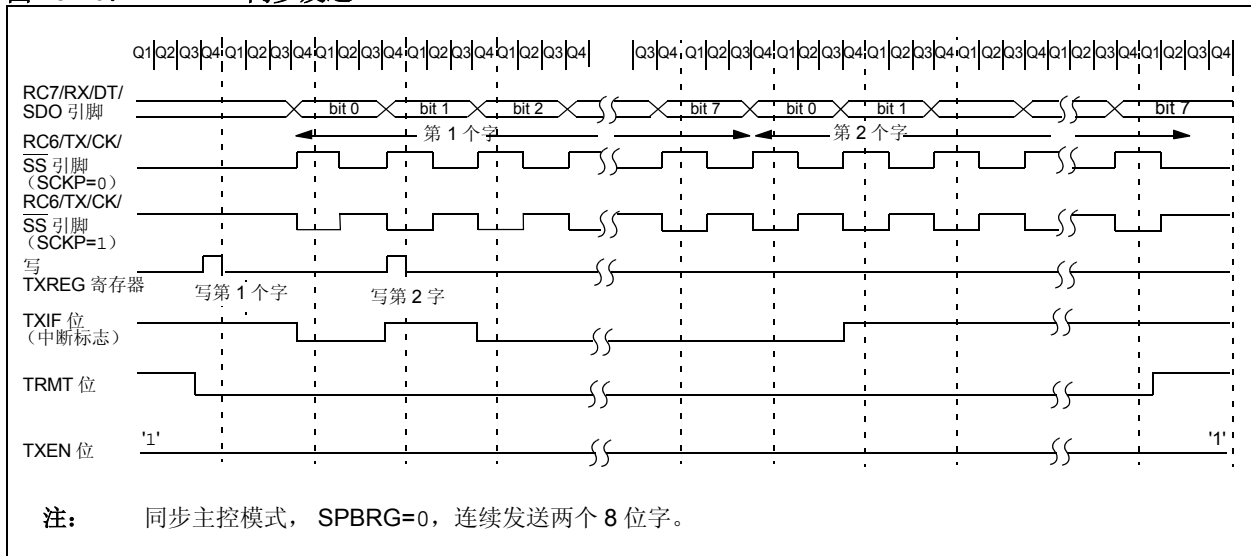
一旦 TXREG 寄存器将数据传递到 TSR 寄存器 (在 1 个 Tcy 周期内发生) 之后, TXREG 寄存器为空, 同时中断位 TXIF (PIR1<4>) 置位。通过置位 / 清零中断使能位 TXIE (PIE1<4>), 该中断可以被使能 / 禁止。不管使能位 TXIE 的状态如何, 标志位 TXIF 都将置位并且无法用软件清零。只有在 TXREG 寄存器装入了新数据后此位才会复位。

标志位 TXIF 表示 TXREG 寄存器的状态, 而另一位 TRMT (TXSTA<1>) 则表示 TSR 寄存器的状态。TRMT 是只读位, 当 TSR 寄存器为空时置位。因为没有与此位关联的中断逻辑, 所以用户必须查询此位以判断 TSR 寄存器是否为空。TSR 寄存器不映射到数据存储器, 因此用户无法使用它。

要建立同步主控发送模式:

1. 用适当的波特率初始化 SPBRGH:SPBRG 寄存器。按需要将 BRGH 和 BRG16 位置位或清零, 以获得所需的波特率。
2. 通过将 SYNC、SPEN 和 CSRC 位置位可以使能同步主串行端口。
3. 如果需要中断, 将使能位 TXIE 置位。
4. 如果需要 9 位发送, 将 TX9 置位。
5. 将 TXEN 位置位以使能发送。
6. 如果选择了 9 位发送, 应该将第 9 位装入 TX9D 位。
7. 将数据装入 TXREG 寄存器以开始发送。
8. 如果使用中断, 应确保 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 已置位。

图 19-10: 同步发送



PIC18F2331/2431/4331/4431

图 19-11: 同步发送 (通过 TXEN)

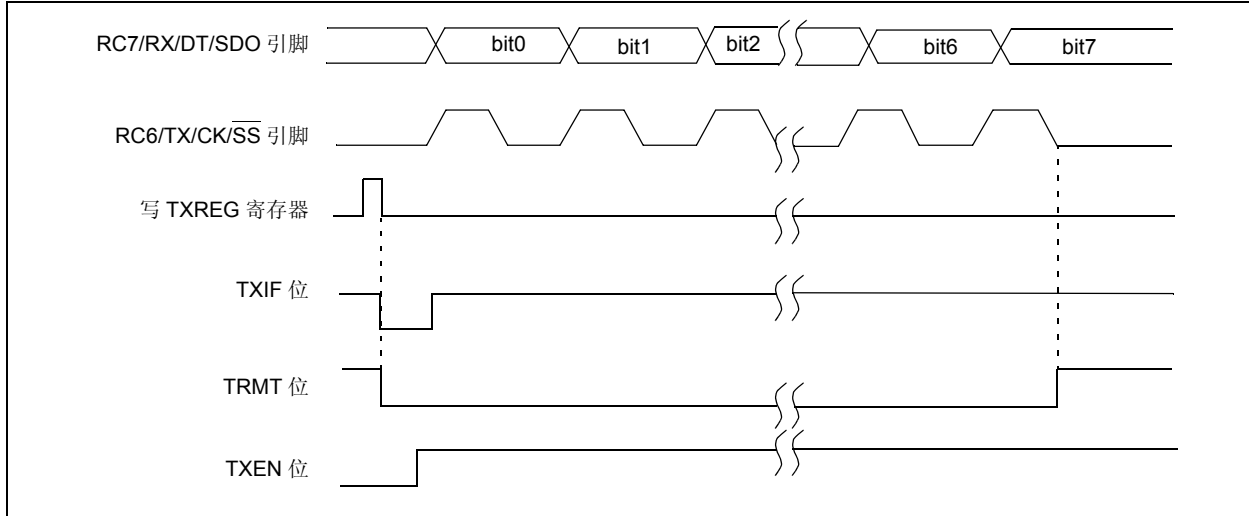


表 19-7: 与同步主控发送模式相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	-000 -000	-000 -000
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 -000	-000 -000
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	-000 -000	-000 -000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
TXREG	USART 发送寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
BAUDCTL	—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-1 0-00	-1-1 0-00
SPBRGH	波特率发生器寄存器的高位字节								0000 0000	0000 0000
SPBRG	波特率发生器寄存器的低位字节								0000 0000	0000 0000

图注: x= 未知, -= 未实现, 读作 0。阴影单元格表示在同步主控发送模式下未使用。

19.4.2 USART 同步主控接收模式

选择了同步模式后，通过将单字接收使能位 **SREN** (**RCSTA<5>**) 或连续接收使能位 **CREN** (**RCSTA<4>**) 置位可使能接收。在时钟下降沿会对 **RC7/RX/DT/SDO** 引脚上的数据采样。

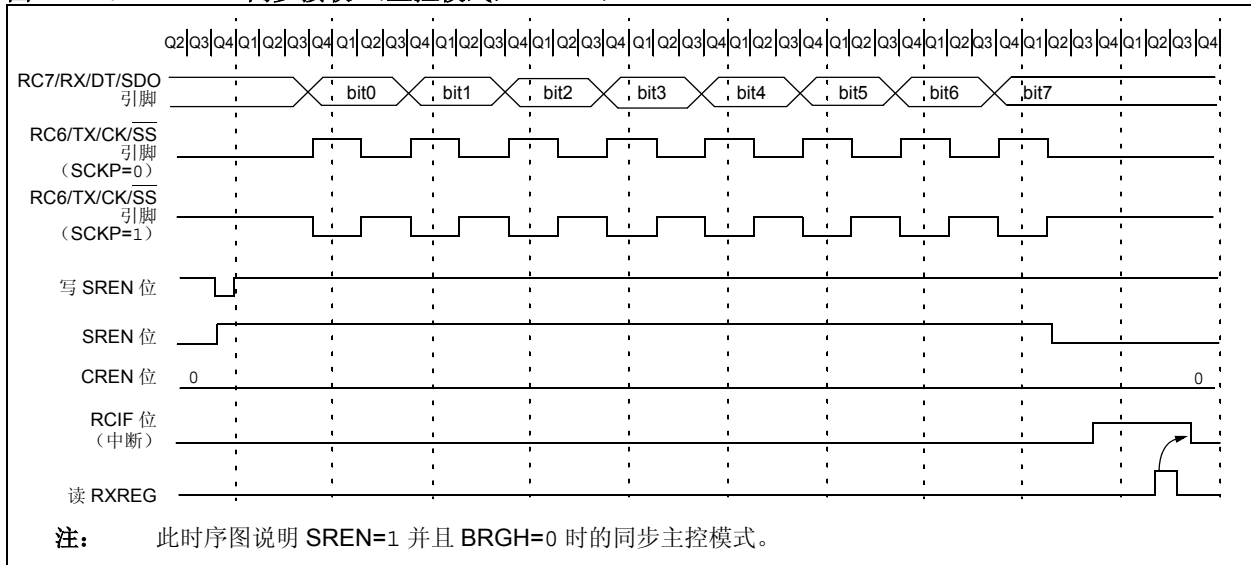
如果使能位 **SREN** 置位，则只接收一个字。如果使能位 **CREN** 置位，在 **CREN** 清零前会连续接收数据。如果两个位同时置位，则 **CREN** 优先。

要建立同步主控接收模式：

1. 用适当的波特率初始化 **SPBRGH:SPBRG** 寄存器。按需要将 **BRGH** 和 **BRG16** 位置位或清零，以获得所需的波特率。
2. 通过将 **SYNC**、**SPEN** 和 **CSRC** 位置位可以使能同步主串行端口。

3. 确保将 **CREN** 和 **SREN** 位清零。
4. 如果需要中断，将中断使能位 **RXIE** 置位。
5. 如果需要 9 位接收，将 **RX9** 位置位。
6. 如果需要单字节接收，将 **SREN** 位置位。如果需要连续接收，将 **CREN** 位置位。
7. 当接收完成时中断标志 **RCIF** 位将置位，并且如果使能位 **RCIE** 置位，会产生一个中断。
8. 读 **RCSTA** 寄存器以获取第 9 位（如果使能）并判断是否在接收过程中发生了错误。
9. 读 **RCREG** 寄存器来读取 8 位接收数据。
10. 如果发生错误，将 **CREN** 位清零以清除错误。
11. 如果使用中断，应确保 **INTCON** 寄存器中的 **GIE** 和 **PEIE** 位 (**INTCON<7:6>**) 已置位。

图 19-12: 同步接收（主控模式，SREN）



PIC18F2331/2431/4331/4431

表 19-8: 与同步主控接收模式相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	-000 -000	-000 -000
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 -000	-000 -000
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	-111 -111	-111 -111
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
RCREG	USART 接收寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
BAUDCTL	—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-1 0-00	-1-1 0-00
SPBRGH	波特率发生器寄存器的高位字节								0000 0000	0000 0000
SPBRG	波特率发生器寄存器的低位字节								0000 0000	0000 0000

图注: x= 未知, -= 未实现, 读作 0。阴影单元格表示同步主控接收模式下未使用。

19.5 USART 同步从动模式

将 CSRC 位 (TXSTA<7>) 清零可进入同步从动模式。此模式与同步主控模式的区别在于移位时钟是由 RC6/TX/CK/SS 引脚外部提供的 (在同步主控模式中是内部提供的)。这使器件可以在任何低功耗模式下发送或接收数据。

19.5.1 USART 同步从动发送模式

除了休眠模式以外, 同步主控模式和从动模式的工作原理是相同的。

如果向 TXREG 写两个字, 然后执行 SLEEP 指令, 则会发生以下事件:

- a) 第一个字立即传输到 TSR 寄存器并发送。
- b) 第二个字留在 TXREG 寄存器。
- c) 标志位 TXIF 不会置位。
- d) 当第一个字移出 TSR 时, TXREG 寄存器将第二个字发送到 TSR, 然后标志位 TXIF 置位。
- e) 如果中断使能位 TXIE 置位, 中断将芯片从休眠状态唤醒。如果使能全局中断, 程序会转移到中断向量。

要建立同步从动发送:

1. 通过将 SYNC 和 SPEN 位置位并清零 CSRC 位可以使能同步从串行端口。
2. 将 CREN 和 SREN 位清零。
3. 如果需要中断, 将中断使能位 TXIE 置位。
4. 如果需要 9 位发送, 将 TX9 位置位。
5. 将使能位 TXEN 置位以启用发送。
6. 如果选择了 9 位发送, 应该将第 9 位装入 TX9D 位。
7. 将数据装入 TXREG 寄存器以开始发送。
8. 如果使用中断, 应确保 INTCON 寄存器中的 GIE 和 PEIE 位 (INTCON<7:6>) 已置位。

表 19-9: 与同步从动发送模式相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	-000 -000	-000 -000
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 -000	-000 -000
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	-000 -000	-000 -000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
TXREG	USART 发送寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
BAUDCTL	—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-1 0-00	-1-1 0-00
SPBRGH	波特率发生器寄存器的高位字节								0000 0000	0000 0000
SPBRG	波特率发生器寄存器的低位字节								0000 0000	0000 0000

图注: x= 未知, -= 未实现, 读作 0。阴影单元格表示在同步从动发送模式下未使用。

PIC18F2331/2431/4331/4431

19.5.2 USART 同步从动接收模式

除了休眠模式、任何空闲模式和 SREN 位（在从动模式被忽略）外，同步主控模式和从动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 位置位以使能接收，则在此低功耗模式下可以接收一个字。接收到该字后，RSR 寄存器将把数据发送到 RCREG 寄存器，如果 RCIE 使能位置位，产生的中断将把芯片从低功耗模式唤醒。如果使能全局中断，程序会转移到中断向量。

要建立同步从动接收模式：

1. 通过将 SYNC 和 SPEN 位置位并清零 CSRC 位可以使能同步主串行端口。
2. 如果需要中断，将中断使能位 RCIE 置位。
3. 如果需要 9 位接收，将 RX9 位置位。
4. 要使能接收，将使能位 CREN 置位。
5. 接收完成时标志位 RCIF 会置位。如果中断使能位 RCIE 置位，会产生一个中断。
6. 读 RCSTA 寄存器以获取第 9 位（如果使能）并判断是否在接收过程中发生了错误。
7. 读 RCREG 寄存器来读取 8 位接收数据。
8. 如果发生错误，将 CREN 位清零以清除错误。
9. 如果使用中断，应确保 INTCON 寄存器中的 GIE 和 PEIE 位（INTCON<7:6>）已置位。

表 19-10: 与同步从动接收模式相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	-111 -111	-111 -111
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
RCREG	USART 接收寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
BAUDCTL	—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	-1-1 0-00	-1-1 0-00
SPBRGH	波特率发生器寄存器的高位字节								0000 0000	0000 0000
SPBRG	波特率发生器寄存器的低位字节								0000 0000	0000 0000

图注： x= 未知， -= 未实现，读作 0。阴影单元格表示同步从动接收模式下未使用。

20.0 10 位高速模数转换器 (A/D) 模块

高速模数 (A/D) 转换器模块允许将模拟信号转换为对应的 10 位数字。

A/D 模块在 PIC18F2X31 器件上支持多达 5 个输入通道，在 PIC18F4X31 器件上支持多达 9 个输入通道。

高速 10 位 A/D 模块提供以下性能：

- 每秒多达 200K 次的采样
- 两采样保持输入端用于双通道同步采样
- 可选的同步或连续采样模式
- 用于 A/D 转换结果的 4 字数据缓冲器
- 可选的数据采集时序
- 可选的 A/D 事件触发
- 使用内部振荡器在休眠模式运行

这些性能使它们可以用于很多应用场合，包括电机控制、传感器连接、数据采集和过程控制。在很多情况下，这些功能可以减少与标准 A/D 模块相关的软件开销。

此模块有 9 个寄存器：

- A/D 结果高位寄存器 (ADRESH)
- A/D 结果低位寄存器 (ADRESL)
- A/D 控制寄存器 0 (ADCON0)
- A/D 控制寄存器 1 (ADCON1)
- A/D 控制寄存器 2 (ADCON2)
- A/D 控制寄存器 3 (ADCON3)
- A/D 通道选择寄存器 (ADCHS)
- 模拟 I/O 选择寄存器 0 (ANSEL0)
- 模拟 I/O 选择寄存器 1 (ANSEL1)

PIC18F2331/2431/4331/4431

寄存器 20-1: ADCON0: A/D 控制寄存器 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	ACONV	ACSCH	ACMOD1	ACMOD0	GO/DONE	ADON
bit 7							bit 0

bit 7-6 **未实现位:** 读作 0

bit 5 **ACONV:** 自动转换连续循环模式或单步模式选择位

1= 使能连续循环模式

0= 使能单步模式

bit 4 **ACSCH:** 自动转换单通道或多通道模式位

1= 使能多通道模式, 禁止单通道模式

0= 使能单通道模式, 禁止多通道模式

bit 3-2 **ACMOD:** 自动转换模式顺序选择位

如果 ACSCH=1:

00= 连续模式 1 (SEQM1)。两次采样按顺序进行:

第一次采样: A 组

第二次采样: B 组

01= 连续模式 2 (SEQM2)。四次采样按顺序进行:

第一次采样: A 组

第二次采样: B 组

第三次采样: C 组

第四次采样: D 组

10= 同步模式 1 (STNM1)。两次采样同时进行:

第一次采样: A 组和 B 组

11= 同步模式 2 (STNM2)。两次采样同时进行:

第一次采样: A 组和 B 组

第二次采样: C 组和 D 组

如果 ACSCH=0, 自动转换单通道顺序模式使能:

00= 单通道模式 1 (SCM1)。对 A 组采样并转换

01= 单通道模式 2 (SCM2)。对 B 组采样并转换

10= 单通道模式 3 (SCM3)。对 C 组采样并转换

11= 单通道模式 4 (SCM4)。对 D 组采样并转换

注: 有关 A 组、B 组、C 组和 D 组的信息, 请参阅 ADCHS 寄存器。

bit 1 **GO/DONE:** A/D 转换状态位

1= A/D 转换周期正在进行。将该位置位来启动 A/D 转换周期。如果自动转换单步模式被使能 (ACONV = 0), 当 A/D 转换 (ACMOD 的置位决定是单通道还是多通道) 完成时该位自动由硬件清零。如果使能自动转换连续循环模式 (ACONV=1), 该位在被用户 / 触发器置位 (连续转换) 后将保持置位状态。用户可以手工清零该位来停止转换。

0= A/D 转换或多次转换完成 / 没有处于进行中

bit 0 **ADON:** A/D 启动位

1= 使能 A/D 转换器模块 (在快速上电延迟之后, 开始连续采样)

0= 禁止 A/D 转换器模块

图注:

R= 可读位

W= 可写位

U = 未实现位, 读作 0

-n= 复位时的值

1= 置位

0= 清零

x= 未知位

PIC18F2331/2431/4331/4431

寄存器 20-2: ADCON1: A/D 控制寄存器 1

R/W-0	R/W-0	U-0	R/W-0	R-0	R-0	R-0	R-0
VCFG1	VCFG0	—	FIFOEN	BFEMT	BFOVFL	ADPNT1	ADPNT0
bit 7						bit 0	

bit 7-6 **VCFG<1:0>**: A/D VREF+ 和 A/D VREF- 源选择位

00= VREF+=AVDD, VREF-=AVSS (AN2 和 AN3 为模拟输入或数字 I/O)

01= VREF+= 外部 VREF+, VREF-=AVSS (AN2 为模拟输入或数字 I/O)

10= VREF+=AVDD, VREF-= 外部 VREF- (AN3 为模拟输入或数字 I/O)

11= VREF+= 外部 VREF-, VREF-= 外部 VREF-

bit 5 **未实现位**: 读作 0

bit 4 **FIFOEN**: FIFO 缓冲器使能位

1= 使能 FIFO

0= 禁止 FIFO

bit 3 **BFEMT**: 缓冲器变空位

1= FIFO 已空

0= FIFO 未空 (四个缓冲器位置中至少有一个包含未读取 A/D 结果数据)

bit 2 **BFOVFL**: 缓冲器溢出位

1= A/D 结果覆盖了包含未读取数据的缓冲器位置

0= A/D 结果未溢出

bit 1-0 **ADPNT<1:0>**: 缓冲器读指针位置位

指定下一个读位置。

00= 缓冲器地址 0

01= 缓冲器地址 1

10= 缓冲器地址 2

11= 缓冲器地址 3

图注:

R= 可读位

W= 可写位

U = 未实现位, 读作 0

-n= 复位时的值

1= 置位

0= 清零

x= 未知位

PIC18F2331/2431/4331/4431

寄存器 20-3: **ADCON2: A/D 控制寄存器 2**

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	ACQT3	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7						bit 0	

bit 7 **ADFN:** A/D 结果格式选择位

- 1= 右对齐
- 0= 左对齐

bit 6-3 **ACQT<3:0>:** A/D 采集时间选择位

- 0000= 无延迟⁽¹⁾ (当 GO/DONE 位置位时转换立即开始)
- 0001= 2 TAD
- 0010= 4 TAD
- 0011= 6 TAD
- 0100= 8 TAD
- 0101= 10 TAD
- 0110= 12 TAD
- 0111= 16 TAD
- 1000= 20 TAD
- 1001= 24 TAD
- 1010= 28 TAD
- 1011= 32 TAD
- 1100= 36 TAD
- 1101= 40 TAD
- 1110= 48 TAD
- 1111= 64 TAD

bit 2-0 **ADCS<2:0>:** A/D 转换时钟选择位

- 000 = FOSC/2
- 001 = FOSC/8
- 010 = FOSC/32
- 011 = FRC/4⁽²⁾
- 100 = FOSC/4
- 101 = FOSC/16
- 110 = FOSC/64
- 111 = FRC (内部 A/D RC 振荡器)

- 注**
- 1: 如果选择 RC 作为 A/D 时钟源, 在采样 / 转换开始之前增加一个 T_{CY} 时间。
 - 2: 由于内部 A/D RC 振荡器的频率增加, FRC/4 提供与以前的 A/D 模块相兼容的时钟频率。
 - 3: TACQ 在连续模式中应该为 12 TAD 以上。

图注:

R= 可读位	W= 可写位	U = 未实现位, 读作 0
-n= 复位时的值	1= 置位	0= 清零 x= 未知位

PIC18F2331/2431/4331/4431

寄存器 20-4: ADCON3: A/D 控制寄存器 3

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
ADRS1	ADRS0	—	SSRC4	SSRC3	SSRC2	SSRC1	SSRC0	
bit 7								bit 0

bit 7-6 **ADRS<1:0>**: 连续循环模式下 A/D 结果缓冲器深度中断选择控制位

在单步模式中 ADRS 位被忽略。

00= 每向缓冲器写入一个字就产生中断

01= 每向缓冲器写入第二个和第四个字就产生中断

10= 每向缓冲器写入第四个字就产生中断

11= 未实现

bit 5 **未实现位**: 读作 0

bit 4:0 **SSRCx<4:0>**: A/D 触发源选择位

00000=禁止所有触发

xxxx1=外部中断 RC3/INT0 启动 A/D 序列

xxx1x=Timer5 启动 A/D 序列

xx1xx=输入捕捉 1 (IC1) 启动 A/D 序列

x1xxx=CCP2 比较匹配启动 A/D 序列

1xxxx=功率控制 PWM 模块上升沿启动 A/D 序列

注 1: SSRCx<4:0> 位可以被置位, 这样任何触发都将启动转换
(例如, SSRCx<4:0>=00101, 将在 RC3/INT0 或输入捕捉 1 事件发生时触发 A/D 转换)。

图注:

R= 可读位

W= 可写位

U = 未实现位, 读作 0

-n= 复位时的值

1= 置位

0= 清零

x= 未知位

PIC18F2331/2431/4331/4431

寄存器 20-5: **ADCHS: A/D 通道选择寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GDSEL1	GDSEL0	GBSEL1	GBSEL0	GCSEL1	GCSEL0	GASEL1	GASEL0
bit 7						bit 0	

bit 7-6 **GDSEL1:GDSEL0**: D 组选择位

S/H-2 正输入端

00= AN3

01= AN7⁽¹⁾

1x= 保留

bit 5-4 **GBSEL1:GBSEL0**: B 组选择位

S/H-2 正输入端

00= AN1

01= AN5⁽¹⁾

1x= 保留

bit 3-2 **GCSEL1:GCSEL0**: C 组选择位

S/H-1 正输入端

00= AN2

01= AN6⁽¹⁾

1x= 保留

bit 1-0 **GASEL1:GASEL0**: A 组选择位

S/H-1 正输入端

00= AN0

01= AN4

01= AN8⁽¹⁾

11= 保留

注 1: 只有在 PIC18F4X31 器件中 AN5 至 AN8 才可用。

图注:

R= 可读位

W= 可写位

U = 未实现, 读作 0

-n= 复位时的值

1= 置位

0= 清零

x= 未知

PIC18F2331/2431/4331/4431

寄存器 20-6: ANSEL0: 模拟选择寄存器 0⁽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
ANS7 ⁽²⁾	ANS6 ⁽²⁾	ANS5 ⁽²⁾	ANS4	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

bit 7-0 **ANS<7:0>**: 模拟输入功能选择位

对应引脚 AN<7:0>

1= 模拟输入

0= 数字 I/O

注 1: 将引脚设置为模拟输入将禁止数字输入缓冲器。作为输入引脚时应该置位相应的 TRIS 位，作为输出（模拟或数字）引脚时应该清零相应的 TRIS 位。ANSx 位直接对应于 ANx 引脚（例如，ANS0=AN0，ANS1=AN1 等等）。未使用的 ANSx 位读为 0。

2: 只有在 PIC18F4X31 器件中 ANS7 至 ANS5 才可用。

图注:

R= 可读位

W= 可写位

U = 未实现位，读作 0

-n= 复位时的值

1= 置位

0= 清零

x= 未知位

寄存器 20-7: ANSEL1: 模拟选择寄存器 1⁽¹⁾

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-1
—	—	—	—	—	—	—	ANS8 ⁽²⁾
bit 15							bit 8

bit 15-9 **未实现位:** 读作 0

bit 8 **ANS8:** 模拟输入功能选择位

1= 模拟输入

0= 数字 I/O

注 1: 将引脚设置为模拟输入将禁止数字输入缓冲器。作为输入引脚时应该置位相应的 TRIS 位，作为输出引脚（模拟或数字）时应该清零相应的 TRIS 位。ANSx 位直接对应于 ANx 引脚（例如，ANS0=AN0，ANS1=AN1 等等）。未使用的 ANSx 位读为 0。

2: ANS8 只在 PIC18F4X31 器件上才可用。

图注:

R= 可读位

W= 可写位

U = 未实现位，读作 0

-n= 复位时的值

1= 置位

0= 清零

x= 未知位

PIC18F2331/2431/4331/4431

A/D 通道分为四组，每组分别有 2 或 3 个通道。对于 PIC18F2X31 器件来说，AN0 和 AN4 在 A 组，AN1 在 B 组，AN2 在 C 组而 AN3 在 D 组。对于 PIC18F4X31 器件来说 AN0、AN4 和 AN8 在 A 组，AN1 和 AN5 在 B 组，AN2 和 AN6 在 C 组而 AN3 和 AN7 在 D 组。配置 A/D 通道选择寄存器 ADCHS 可以选择使用每组中的通道。

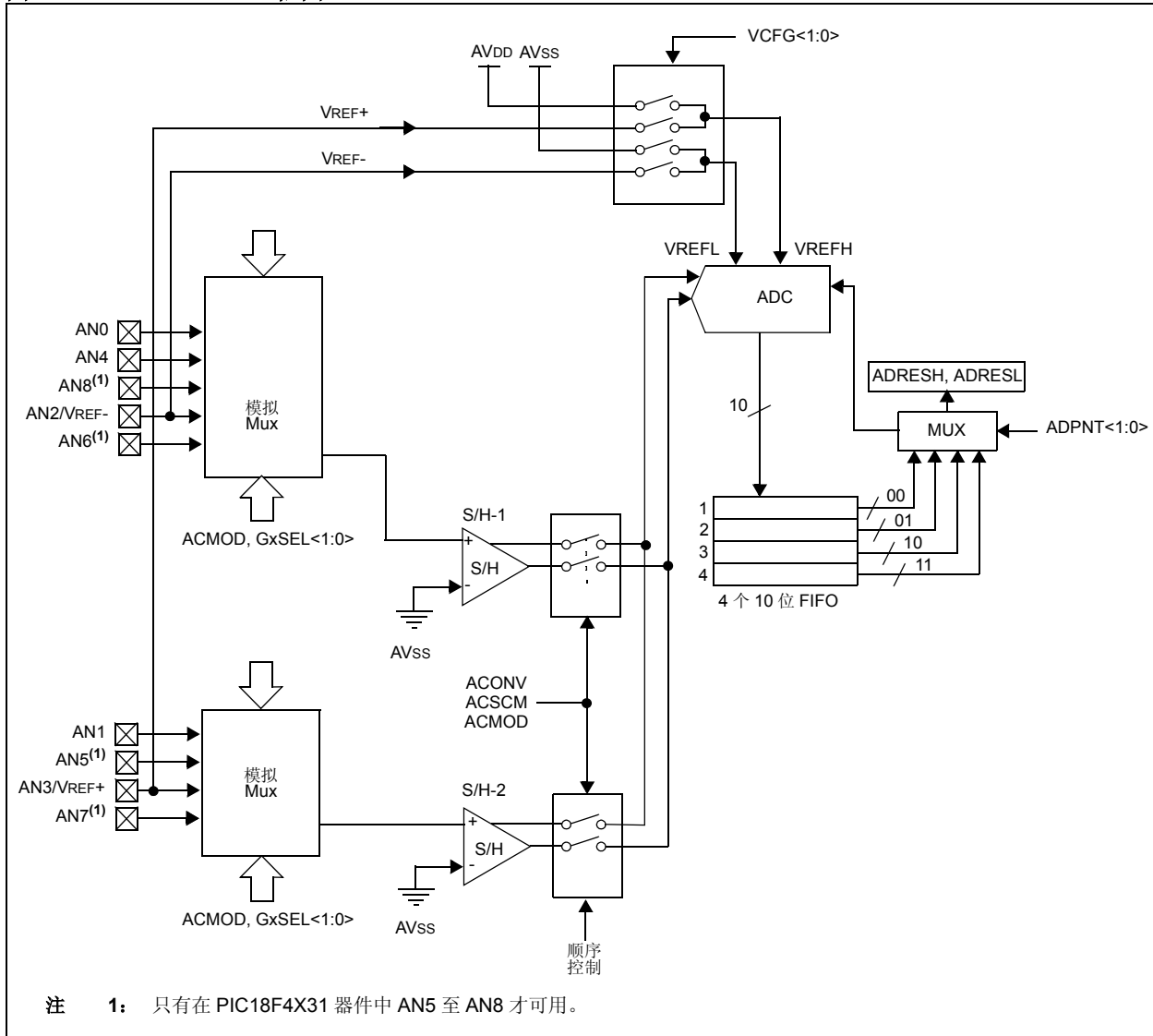
可用软件选择模拟参考电压作为器件的正模拟供电电压和负模拟供电电压 (AVDD 和 AVSS)，或 RA3/AN3/VREF+/CAP2/QEA 和 RA2/AN2/VREF-/CAP1/INDX 电平，又或是供电电源和外部电源的组合。寄存器 ADCON1 控制参考电压的设置。

A/D 转换器具备可在休眠状态下工作的独特特性。要使 A/D 转换器在休眠状态下运行，A/D 转换时钟必须来自于 A/D 模块内部的 RC 振荡器。

器件复位强制所有寄存器进入复位状态。同时强制关闭 A/D 模块并中止任何正在进行的转换。

可以用 ANSEL0 和 ANSEL1 寄存器将每个与 A/D 转换器相关的端口引脚分别配置成模拟输入或数字 I/O。ADRESH 和 ADRESL 寄存器包含 ADPNT<1:0> (ADCON1<1:0>) 指向的结果缓冲器中的值。结果缓冲器是一个深度为 4 层的循环缓冲器，它有一个空状态位 BEMT (ADCON1<3>) 和一个溢出状态位 BOVFL (ADCON1<2>)。

图 20-1: A/D 框图



20.1 配置 A/D 转换器

A/D 转换器有两种转换类型，两种操作模式以及八种不同的顺序模式。这些特性由 ACONV 位 (ADCON0<5>)、ACSH 位 (ADCON0<4>) 和

ACMOD<1:0> 位 (ADCON0<3:2>) 控制。此外，A/D 通道根据 ADCHS 寄存器中的定义分为四组。表 20-1 所示为 ACSCH 和 ACMOD<1:0> 位所控制的顺序配置。

表 20-1: 自动转换顺序配置

模式	ACSCH	ACMOD	描述
多通道连续模式 1 (SEQM1)	1	00	连续采样并转换 A 组和 B 组
多通道连续模式 2 (SEQM2)	1	01	连续采样并转换 A、B、C 和 D 组
多通道同步模式 1 (STNM1)	1	10	同步采样 A 组和 B 组，然后连续进行转换
多通道同步模式 2 (STNM2)	1	11	同步采样 A 组和 B 组，然后连续进行转换。同步采样 C 组和 D 组，然后连续进行转换。
单通道模式 1 (SCM1)	0	00	采样并转换 A 组
单通道模式 2 (SCM2)	0	01	采样并转换 B 组
单通道模式 3 (SCM3)	0	10	采样并转换 C 组
单通道模式 4 (SCM4)	0	11	采样并转换 D 组

20.1.1 转换类型

高速 10 位 A/D 转换器模块有两种转换类型，可使用 ACONV 位选择。当 ACONV=0 时，单步模式允许单次转换或顺序转换。在顺序转换结束的时候，GO/DONE 位将被自动清零且置位中断标志位 ADIF。当采用单步模式并配置为同步模式 STNM2 时，必须使用采集时间来确保模拟输入信号的正确转换。

当 ACONV=1 时，连续循环模式允许以连续循环的方式来执行已定义顺序转换。在这个模式中，用户可以通过置位 GO/DONE 位来启动转换，或通过一个 A/D 触发来启动转换。中断标志位 ADIF 根据 ADRS<1:0> 位 (ADCON3<7:6>) 的配置而置位。在同步模式 STNM1 和 STNM2 中，必须配置采集时间以确保模拟输入信号的正确转换。

20.1.2 转换模式

ACSCH 位 (ADCON0<4>) 控制在配置的顺序中将使用多少个通道。当清零时，A/D 配置为单通道转换，它将转换 ACMOD<1:0> 所选定的组和 GxSEL<1:0> (ADCHS) 所选定的通道。当 ACSCH=1 时，A/D 配置为多通道转换，顺序由 ACMOD<1:0> 定义。

PIC18F2331/2431/4331/4431

20.1.3 转换顺序

ACMOD<1:0> 位控制 A/D 转换顺序。当 ACSCH = 0 时，A/D 被配置为采样并转换一个单通道。ACMOD 位选择用哪个组来执行转换，而 GxSEL<1:0> 位选择要转换组中的哪个通道。如果使能单步模式，A/D 中断标志位在通道被转换后置位。如果使能连续循环模式，A/D 中断标志位将根据 ADRS<1:0> 位置位。

当 ACSCH = 1 时，多通道顺序转换被使能而且有两个子模式可供选择。第一个模式是带两个设置的连续模式。第一种设置称为 SEQM1 且首先对选定的 A 组通道进行采样和转换，然后对选定的 B 组通道进行采样和转换。第二种模式叫做 SEQM2，它对 A 组通道、B 组通道、C 组通道以及 D 组通道依次进行采样和转换。

第二个多通道顺序转换子模式是同步采样模式。这个模式中也有两种设置。第一种设置叫做 STNM1，使用 A/D 模块上的两个采样保持电路。所选择的 A 组和 B 组通道被同步采样，然后先转换 A 组通道再转换 B 组通道。第二种设置叫做 STNM2，它的启动与 STNM1 相同，但在它之后有 C 组和 D 通道的同步采样。然后 A/D 模块将转换 C 组通道，再转换 D 组通道。

20.1.4 触发 A/D 转换

PIC18F2331/2431/4331/4431 器件能够由很多不同的源触发转换。其他的单片机也使用置位 GO/DONE 位的方法触发转换。其他触发源包括：

- RC3/INT0 引脚
- Timer5 溢出
- 输入捕捉 1 (IC1)
- CCP2 比较匹配
- 功率控制 PWM 上升沿

用 SSR<4:0> 位 (ADCON3<4:0>) 使能这些触发。只要置位 ADCON3 中的相应位，这五种触发源的任意组合都可以触发转换。当发生触发时，GO/DONE 位自动由硬件置位，一旦转换完成，该位就被清零。

20.1.5 A/D 模块初始化步骤

按照以下步骤进行 A/D 模块的初始化：

1. 配置 A/D 模块：
 - a) 配置模拟引脚、参考电压和数字 I/O
 - b) 选择 A/D 输入通道
 - c) 选择 A/D 自动转换模式 (单步或连续循环)
 - d) 选择 A/D 转换时钟
 - e) 选择 A/D 转换触发
2. 配置 A/D 中断 (如果需要的话)：
 - a) GIE 位置位
 - b) PEIE 位置位
 - c) ADIE 位置位
 - d) ADIF 位清零
 - e) 选择 A/D 触发器设置
 - f) 选择 A/D 中断优先级
3. 启动 ADC：
 - a) 置位 ADCON0 寄存器中的 ADON 位
 - b) 等待必需的上电启动时间，约为 5-10 μ s
4. 开始采样 / 转换顺序：
 - a) 采样时间最少为 $2T_{AD}$ ，置位 $\overline{GO/DONE}$ 位开始转换。 $\overline{GO/DONE}$ 位由用户用软件进行置位，如果由触发引起转换的话，则由模块进行置位。
 - b) 如果分配给 TACQ 一个值 (TAD 的倍数)，则置位 $\overline{GO/DONE}$ 位启动采样周期，这时采样周期等于 TACQ，然后开始转换。
5. 使用以下其中一个方法等待 A/D 转换完成：
 - a) 如果是单步模式则查询 $\overline{GO/DONE}$ 是否被清零。
 - b) 等待 A/D 中断标志位 (ADIF) 置位。
 - c) 查询是否已清零 BFEMT 位，表示至少第一个转换已完成。
6. 读 A/D 结果，清零 ADIF 标志位，重新配置触发。

20.2 A/D 结果缓冲器

A/D 模块有一个深度为 4 层的结果缓冲器，地址范围从 0 到 3，通过置位 ADCON1 寄存器中的 FIFOEN 位使能。缓冲器以循环方式实现，其中将 A/D 结果存储在一个单元，然后对地址进行递增。如果地址大于 3，指针将回到 0。结果缓冲器有一个缓冲器空标志位 BEMT，表示缓冲器内是否有数据。它也有一个溢出标志位 BOVFL，表示新的采样数据是否已经覆盖之前未读取的单元。

指向下一个读操作地址的指针与该缓冲器相关。ADPNT<1:0> 位配置下次读操作的地址。这些位是只读的。

结果缓冲器也有可配置中断触发级别，由 ADRS<1:0> 位配置。用户有三个选择：每次写缓冲器时置位中断标志位、每隔一次写缓冲器时触发中断或者每四次写缓冲器时触发中断。每次开始转换时（GO/DONE 位置位或触发时）将 ADPNT<1:0> 复位为“00”。

注： 当右对齐时，读 ADRESL，并递增 ADPNT。当左对齐时，读 ADRESH，并递减 ADPNT。

20.3 A/D 采集时间要求

为了使 A/D 转换器达到规定精度，必须让充电保持电容（CHOLD）充满至输入通道的电平。模拟输入模块如图 20-2 所示。源阻抗（Rs）和内部采样开关（Rss）阻抗直接影响给电容 CHOLD 充电所需要的时间。采样开关（Rss）阻抗随器件电压（VDD）而改变。源阻抗将影响模拟输入的偏置电压（由于引脚泄漏电流）。**模拟源的最大建议阻抗为 2.5 kΩ。**在选择（改变）了模拟输入通道之后，对通道采样的时间必须大于或等于最小采集时间。在过了这段时间后才开始转换。

注： 当开始转换时，保持电容从输入引脚断开。

可以使用公式 20-1 计算最小采集时间。该公式假设误差为 1/2 LSB（即 A/D 的 1024 步）。1/2 LSB 误差是 A/D 模块达到规定精度所能允许的最大误差。

例 20-1 所示为如何计算所需的最小采集时间 TACQ。在这种情况下，转换器模块在初始阶段被完全上电，因此放大器稳定时间 TAMP 可以忽略。这个计算是在以下应用系统假设的基础上进行的：

CHOLD	=	9 pF
Rs	=	100 Ω
转换误差	≤	1/2 LSB
VDD	=	5V → Rss = 6 kΩ
温度	=	50°C（系统最大值）
VHOLD	=	0V（time = 0 时）

公式 20-1: 采集时间

$$\begin{aligned} TACQ &= \text{放大器稳定时间} + \text{保持电容充电时间} + \text{温度系数} \\ &= TAMP + TC + TCOFF \end{aligned}$$

公式 20-2: 最小 A/D 保持电容充电时间

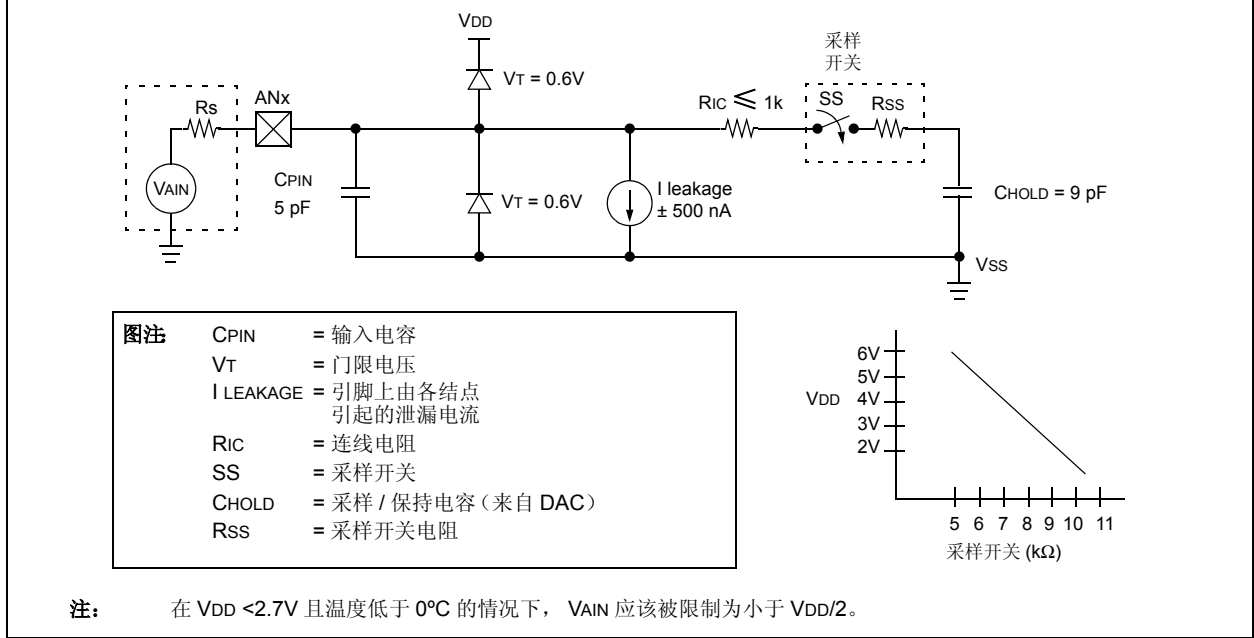
$$\begin{aligned} VHOLD &= (VREF - (VREF/2048)) \cdot (1 - e^{-(Tc/CHOLD)(RIC + RSS + RS)}) \\ \text{或} \\ TC &= -(CHOLD)(RIC + RSS + RS) \ln(1/2048) \end{aligned}$$

PIC18F2331/2431/4331/4431

例 20-1: 计算必需的最小采集时间

TACQ	=	TAMP + TC + TCOFF
TAMP	=	可忽略
TCOFF	=	(Temp-25°C)(0.005 μs/°C) (50°C -25°C)(0.005 μs/°C) = .13 μs
温度系数只有在温度 > 25°C 时需要。当温度低于 25°C 时，TCOFF = 0 μs。		
TC	=	-(CHOLD) (RIC + RSS + RS) ln(1/2047) μs -(9 pF) (1 kΩ + 6 kΩ + 100 Ω) ln(0.0004883) μs = .49 μs + .13 μs = .62 μs
TACQ	=	0 + .62 μs + .13 μs = .75 μs
注: 如果转换器模块已经在休眠模式中，从休眠模式退出起算 TAMP 为 2.0 μs。		

图 20-2: 模拟输入模型



20.4 A/D 参考电压

如果外部参考电压用于取代内部 AVDD 和 AVSS 源，就必须考虑到 VREF+ 和 VREF- 电压源的源阻抗。在采集过程中，这些源提供的电流可以被忽略。但是在转换过程中，A/D 模块将通过这些参考源产生灌电流和拉电流。

为了保持 A/D 精度，参考电压源阻抗应保持为低以减少电压的变化。当参考电流流经这些参考源阻抗时电压将发生变化。

注： 在使用外部参考源时，外部参考电压的电源阻抗必须小于 75Ω 以达到指定 ADC 精度。参考源阻抗更高会增加 ADC 偏置和增益误差。电阻分压器不能提供足够低的源阻抗。为了确保 ADC 可能达到的最佳性能，应该使用运算放大器或其他低阻抗电路对外部 VREF 输入进行缓冲。

20.5 选择和配置自动采集时间

ADCON2 寄存器允许用户选择每次触发 A/D 转换时的采集时间。

当 GO/DONE 位置位时，采样停止，转换开始。用户负责确保在选择了期望的输入通道后，经过必需的采集时间才开始转换。这在 ACQT3:ACQT0 位 (ADCON2<6:3>) 保持在复位状态 (0000) 时完成。

如果需要，可以置位 ACQT 位来选择 A/D 模块的可编程采集时间。当发生触发时，A/D 模块在选定的采集时间内继续采样输入信号，然后自动开始转换。因为采集时间已被编程，就不需要在选择通道以后等待一个采集时间才触发 A/D 转换。如果采集时间被编程，采集时间结束和转换开始都不会有任何指示。

20.6 选择 A/D 转换时钟

每个位的 A/D 转换时间被定义为 TAD。每次 10 位 A/D 转换需要 12 TAD。A/D 转换时钟源可用软件选择。TAD 有以下八种选择：

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- 内部 RC 振荡器
- 内部 RC 振荡器 /4

要进行正确的 A/D 转换，A/D 转换时钟 (TAD) 必须尽可能小，但它必须大于最小 TAD (大约为 416 μs，参见参数 130 了解更多的信息)。

表 20-2 所示为根据器件工作频率和选定的 A/D 时钟源得到的 TAD 结果。

PIC18F2331/2431/4331/4431

表 20-2: TAD 与器件工作频率

AD 时钟源 (TAD)		最大器件频率	
工作时间	ADCS2:ADCS0	PIC18FXX31	PIC18LFXX31 ⁽⁴⁾
2 TOSC	000	4.8 MHz	666 kHz
4 TOSC	100	9.6 MHz	1.33 MHz
8 TOSC	001	19.2 MHz	2.66 MHz
16 TOSC	101	38.4 MHz	5.33 MHz
32 TOSC	010	40.0 MHz	10.65 MHz
64 TOSC	110	40.0 MHz	21.33 MHz
RC/4 ⁽³⁾	011	1.00 MHz ⁽¹⁾	1.00 MHz ⁽²⁾
RC ⁽³⁾	111	4.0 MHz ⁽²⁾	4.0 MHz ⁽²⁾

- 注 1: RC 源的典型 TAD 时间为 2-6 μ s。
注 2: RC 源的典型 TAD 时间为 .5-1.5 μ s。
注 3: 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, 单步模式除外。
注 4: 只适用于低功率器件。

20.7 功耗管理模式下的工作原理

功耗管理模式下, 时钟源和频率部分决定了自动采集时间和 A/D 转换时钟的选择。

如果希望器件在功耗管理模式下进行 A/D 转换, 就应该根据将要使用的功耗管理模式时钟来更新 ADCON2 中的 ACQT3:ACQT0 和 ADCS2:ADCS0 位。在进入功耗管理模式之后 (或者正在运行功耗管理模式时), 可以开始 A/D 采集或转换。一旦采集或转换开始, 就应该由同一个功耗管理模式时钟源继续为器件提供时钟信号, 直到这次转换结束。希望的话可以在转换过程中让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。

在休眠模式下工作要选定 A/D RC 时钟。如果 ACQT3:ACQT0 位置为 “0000” 且转换开始, 则转换将延迟一个指令周期以允许执行 SLEEP 指令并进入休眠模式。OSCCON 寄存器中的 IDLEN 和 SCS 位必须在转换开始之前被清零。

注: A/D 转换只有配置为单步操作时才可以在休眠模式下运行。如果该器件在休眠模式下, 可能会是一个非 A/D 模块的来源唤醒该器件, 用户必须在读结果之前查询 ADCON<GO/DONE> 位来确保它已清零。

20.8 配置模拟端口引脚

ANSEL0、ANSEL1、TRISA 和 TRISE 寄存器都配置 A/D 端口引脚。需要配置为模拟输入的端口引脚必须将它们相应的 TRIS 位置位 (输入)。如果 TRIS 位被清零 (输出), 将转换数字输出电平 (VOH 或 VOL)。

A/D 操作与 ANSEL0、ANSEL1 和 TRIS 位的状态无关。

- 注 1: 读取端口寄存器时, 所有配置为模拟输入通道的引脚均读为 0 (低电平)。配置为数字输入的引脚将转换模拟输入信号。将正确转换配置为数字输入引脚上的模拟电平。
- 注 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所消耗的电流超出器件规范的限制。

20.9 A/D 转换

图 20-3 所示为 GO 位被置位且 ACQT2:ACQT0 位被清零后 A/D 转换器的操作。转换在下一个指令执行之后开始，以允许器件在转换开始之前进入休眠模式。必须选择内部 A/D RC 振荡器在休眠模式中执行转换。

图 20-4 所示为 GO 位被置位且 ACQT3:ACQT0 位被置为“010”后 A/D 转换器的操作，并选择了转换开始前的采集时间为 4 TAD。

在转换过程中清零 $\overline{GO/DONE}$ 位将终止正在进行的转换。结果寄存器的单元将包含部分完成的 A/D 转换采样。这不会置位 ADIF 标志位，因此，用户必须在缓冲器单元被转换序列覆盖之前读缓冲器单元。

在 A/D 转换完成或停止以后，在下次采集开始之前需要等待 2 TAD。这个等待时间之后将自动对所选通道进行采样。

注： 用相同的指令开始 A/D 转换时 $\overline{GO/DONE}$ 位不应该被置位。

图 20-3: A/D 转换 TAD 周期 (ACQT<2:0> = 000, TACQ = 0)

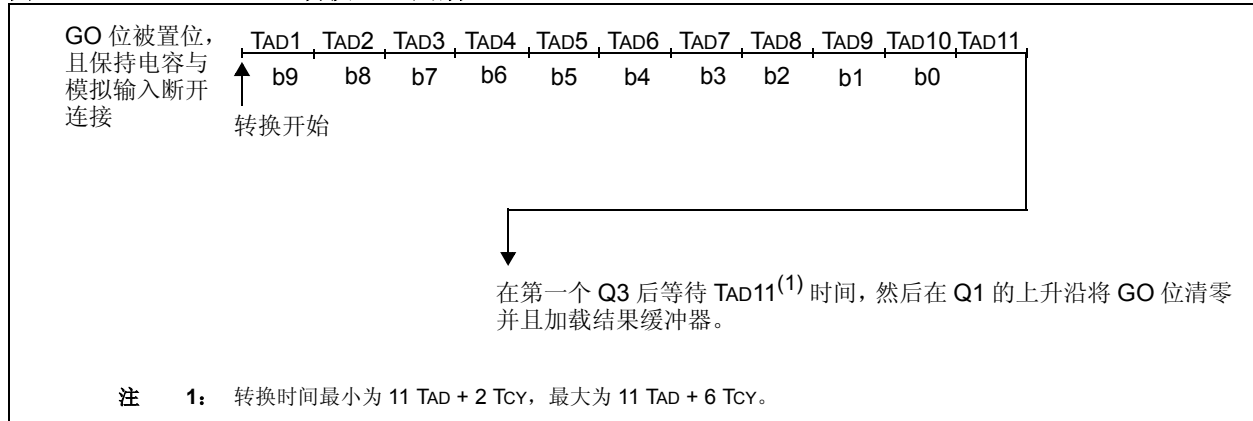
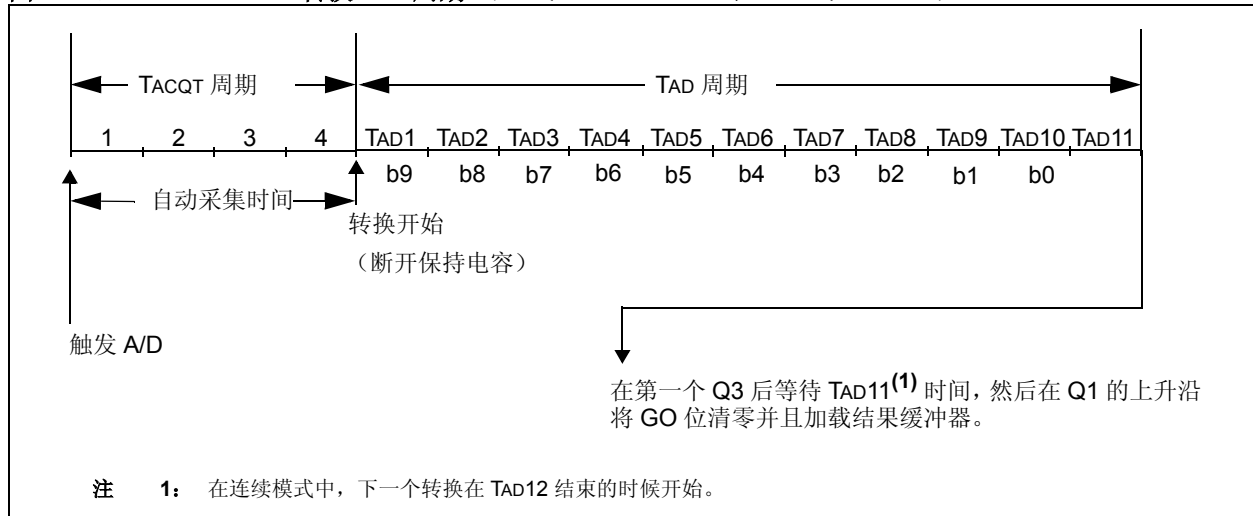


图 20-4: A/D 转换 TAD 周期 (ACQT<3:0> = 0010, TACQ = 4 TAD)



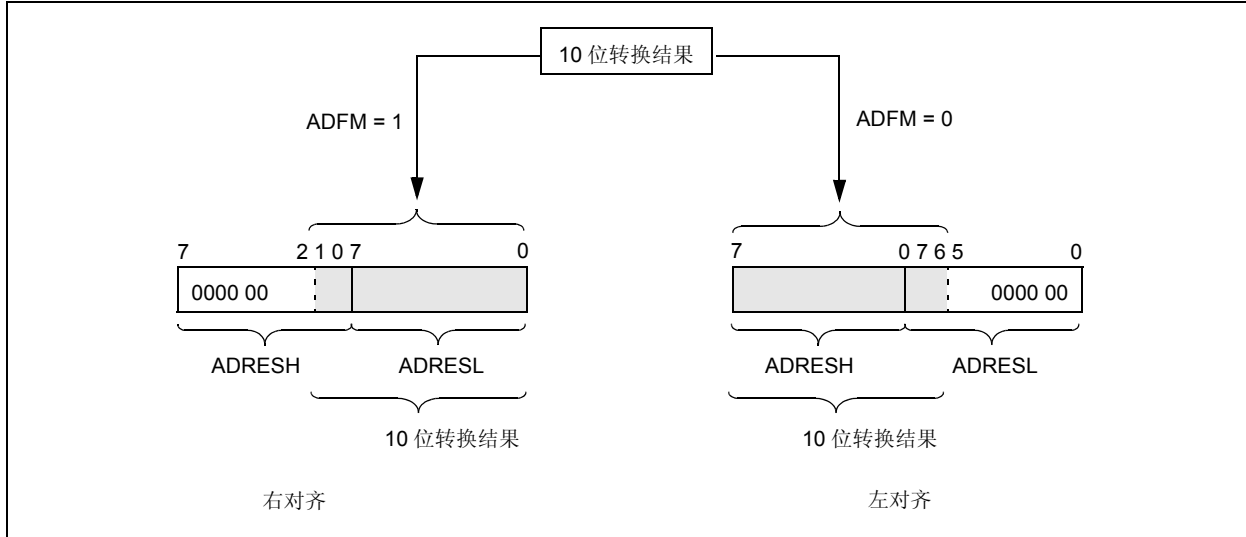
PIC18F2331/2431/4331/4431

20.9.1 A/D 结果寄存器

在 A/D 转换完成时，10 位 A/D 结果被装入 ADRESH:ADRESL 寄存器对。该寄存器对的宽度为 16 位。可以灵活选择该 A/D 模块的 10 位结果是以左对齐还是右对齐的格式存放在 16 位结果寄存器中。A/D 格式

选择位 (ADFM) 控制对齐方式。图 20-5 所示为 A/D 转换结果对齐的操作。多余位以 0 填充。当这些位置没有被 A/D 转换结果覆盖时 (A/D 被禁止)，这两个寄存器可以作为两个通用 8 位寄存器。

图 20-5: A/D 转换结果对齐



公式 20-3: 多通道模式的转换时间

连续模式

$$T = (TACQ)_A + (TCON)_A + [(TACQ)_B - 12TAD] + (TCON)_B + [(TACQ)_C - 12TAD] + (TCON)_C + [(TACQ)_D - 12TAD] + (TCON)_D$$

同步模式

$$T = TACQ + (TCON)_A + (TCON)_B + TACQ + (TCON)_C + (TCON)_D$$

PIC18F2331/2431/4331/4431

表 20-3: A/D 寄存器综述

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 与 BOR 时的值	所有其他复位时的值	
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 0000	0000 0000	
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000	
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000	
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111	
PIR2	OSCFIF	CMIF	—	EEIF	BCLIF	LVDIF	TMR3IF	CCP2IF	00-0 0000	00-0 0000	
PIE2	OSCFIE	CMIE	—	EEIE	BCLIE	LVDIE	TMR3IE	CCP2IE	00-0 0000	00-0 0000	
IPR2	OSCFIP	CMIP	—	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP	11-1 1111	11-1 1111	
ADRESH	A/D 结果寄存器高字节								xxxx xxxx	uuuu uuuu	
ADRESL	A/D 结果寄存器低字节								xxxx xxxx	uuuu uuuu	
ADCON0	—	—	ACONV	ACMOD1	ACMOD0	CHS0	GO/DONE	ADON	00-1 0000	00-1 0000	
ADCON1	VCFG1	VCFG0	—	FIFOEN	BFEMT	BFOVFL	ADPNT1	ADPNT0	--00 qqqq	--00 qqqq	
ADCON2	ADFM	ACQT3	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	0-00 0000	0-00 0000	
ADCON3	ADRS1	ADRS0	—	SSRC4	SSRC3	SSRC2	SSRC1	SSRC0	00-0 0000	00-0 0000	
ADCHS	GDSEL1	GDSEL0	GBSEL1	GBSEL0	GCSEL1	GCSEL0	GASEL1	GASEL0	0000 0000	0000 0000	
ANSEL0	ANS7 ⁽⁶⁾	ANS6 ⁽⁶⁾	ANS5 ⁽⁶⁾	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111	
ANSEL1	—	—	—	—	—	—	—	ANS8 ⁽⁵⁾	---- -1	---- -1	
PORTA	RA7 ⁽⁴⁾	RA6 ⁽⁴⁾	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000	
TRISA	TRISA7 ⁽⁴⁾	TRISA6 ⁽⁴⁾	PORTA 的数据方向控制寄存器							--11 1111	--11 1111
PORTE ⁽²⁾	—	—	—	—	RE3 ⁽¹⁾	读 PORTE 引脚, 然后写 ⁽⁴⁾			---- xxxx	---- uuuu	
TRISE ⁽³⁾	IBF	OBE	IBOV	PSPMODE	—	PORTE 数据方向			0000 -111	0000 -111	
LATE ⁽³⁾	—	—	—	—	PORTE 输出数据锁存器				---- -xxx	---- -uuu	

图注: x = 未知, u = 不变 - = 未实现, 读作 0, q = 根据条件而变化。
阴影单元格不适用于 A/D 转换。

- 注 1: 当配置寄存器中的 MCLRE 位为 0 时, RE3 端口位只可作为输入引脚。
2: 该寄存器在 PIC18F2X31 器件上未实现。
3: 这些位在 PIC18F2X31 器件上未实现。
4: 这些引脚可以根据选定的振荡模式配置为端口引脚。
5: 只有在 PIC18F4X31 器件中 ANS5 至 ANS8 才可用。
6: 在 28 引脚的器件上不可用。

PIC18F2331/2431/4331/4431

注:

21.0 低压检测

很多应用中都期待能有一个判断器件电压 (VDD) 是否低于指定电压的功能。可以为应用创建一个工作窗口，在器件电压偏离有效工作范围前，应用软件可在此窗口执行“保护性工作”。这可以使用低压检测 (Low-Voltage Detect module, LVD) 模块来实现。

该模块是一个软件可编程电路，其中可以指定器件的电压跳变点。当器件电压低于此指定跳变点时，中断标志位置位。如果允许中断，程序执行会转到中断向量地址，然后软件会响应中断源。

低压检测电路完全由软件控制。这使得电路可由软件关断，以便器件的电流消耗最低。

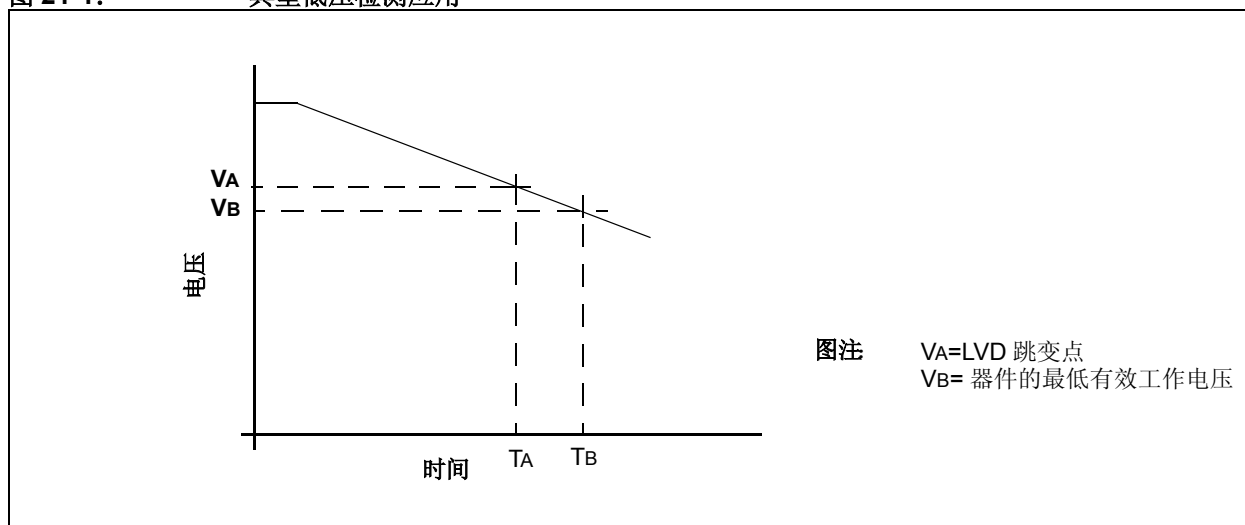
图 21-1 是一种可能的应用电压曲线 (常用于电池)。器件电压会随时间逐渐下降。当器件电压等于电压 V_A 时，LVD 逻辑电路会在 T_A 时刻产生中断。然后应用软件有

时间在器件电压偏离有效工作范围之前关闭系统。电压点 V_B 是最低有效工作电压的指定值，对应时间 T_B 。 T_B 减 T_A 的差就是总的关闭时间。

图 21-2 所示为 LVD 模块的框图。比较器使用内部产生的参考电压作为设定点。当选定器件分压输出经过 (低于) 设定点时，LVDIF 位就会被置位。

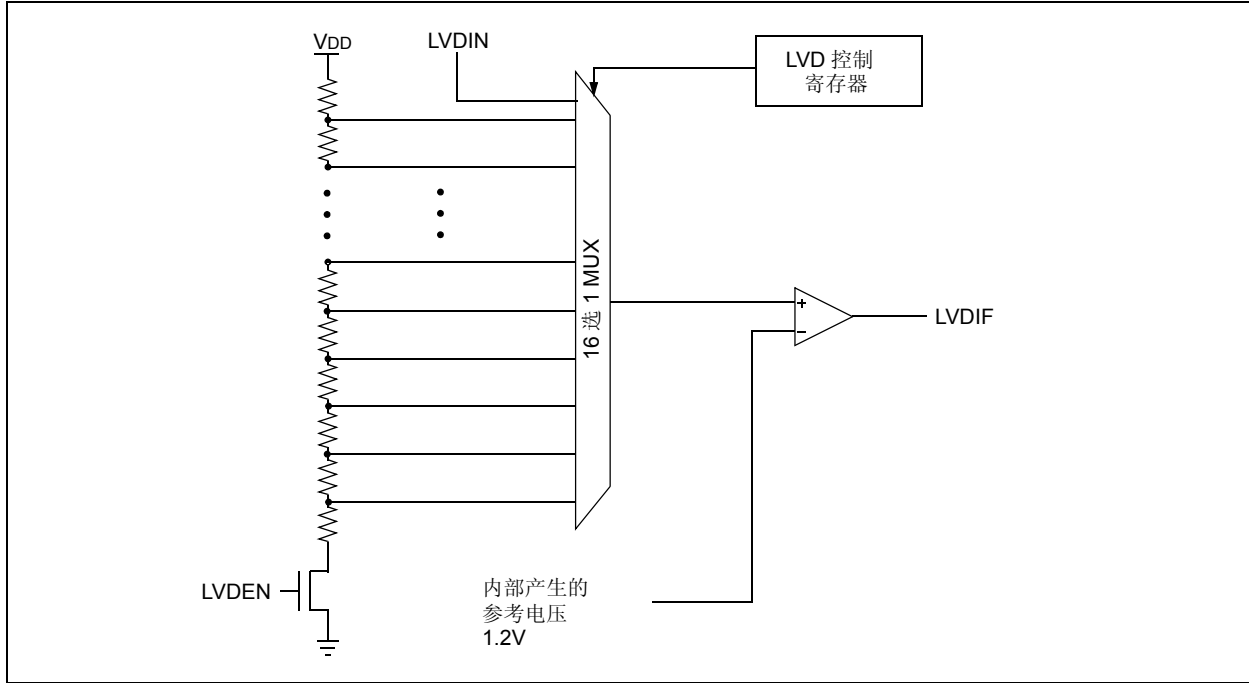
电阻分压器的每个节点代表一个“跳变点”电压。“跳变点”电压指在 LVD 模块发出中断前器件工作所需的最低电压。当电源电压等于跳变点时，从电阻阵列得到的分压等于电压参考模块产生的 1.2 V 内部参考电压。比较器随后会发出一个中断信号将 LVDIF 位置位。此电压可以用软件编程为 16 个值中的任何一个 (见图 21-2)。对 LVDL3:LVDL0 位 (LVDCON<3:0>) 编程可选择跳变点。

图 21-1: 典型低压检测应用



PIC18F2331/2431/4331/4431

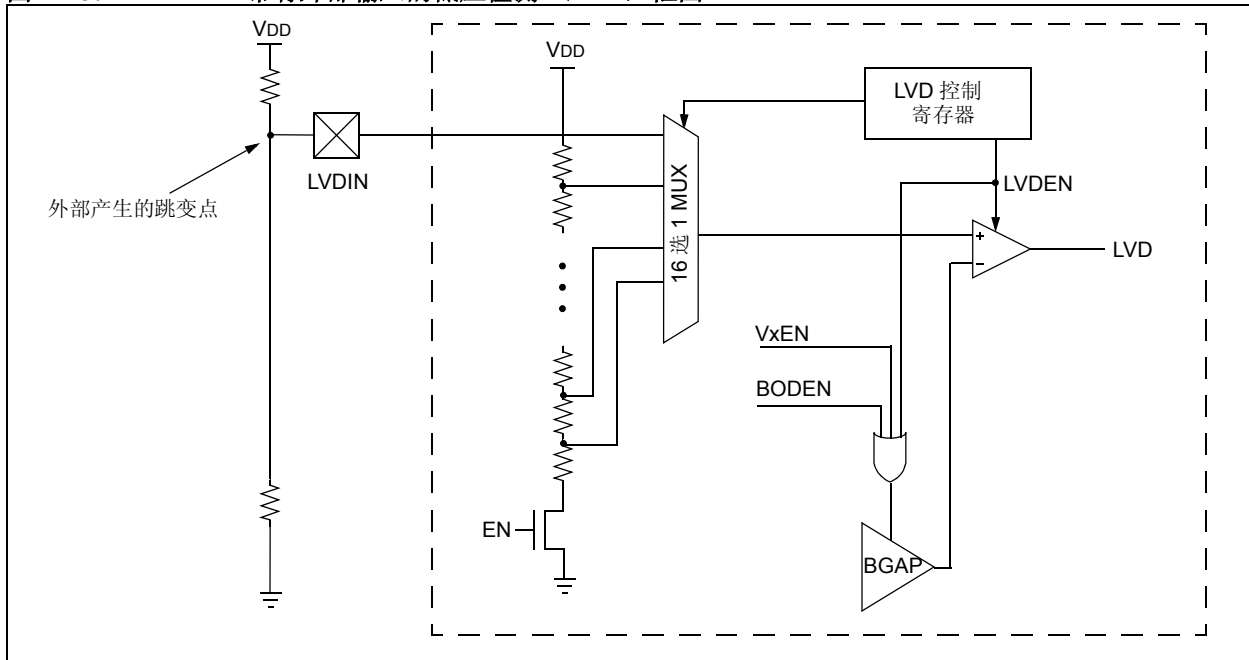
图 21-2: 低压检测 (LVD) 框图



LVD 模块还有一项功能可以让用户使用外部电源为模块提供检测电压。当 LVLDL3:LVLDL0 位设置为 1111 时, 此模式被使能。在此状态下, 比较器输入复用外部输入

引脚 LVDIN (图 21-3)。这可以让用户配置在有效工作电压范围内的任何电压发生低压检测中断, 从而更具灵活性。

图 21-3: 带有外部输入的低压检测 (LVD) 框图



PIC18F2331/2431/4331/4431

21.1 控制寄存器

低压检测控制寄存器控制低压检测电路的工作。

寄存器 21-1:

LVDCON 寄存器

U-0	U-0	R-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1	
—	—	IRVST	LVDEN	LVDL3	LVDL2	LVDL1	LVDL0	
bit 7								bit 0

bit 7-6 **未实现位:** 读作 0

bit 5 **IRVST:** 内部参考电压稳定标志位

1 = 低压检测逻辑将在指定电压范围产生中断标志

0 = 低压检测逻辑将不会在指定电压范围产生中断标志, 并且不应该允许 LVD 中断

bit 4 **LVDEN:** 低压检测电源使能位

1= 使能 LVD, LVD 电路通电

0= 禁止 LVD, LVD 电路断电

bit 3-0 **LVDL3:LVDL0:** 低压检测限制位

1111= 使用外部模拟输入 (输入来自 LVDIN 引脚)

1110= 4.23V - 4.96V

1101= 3.93V - 4.62V

1100= 3.75V - 4.40V

1011= 3.56V - 4.18V

1010= 3.38V - 3.96V

1001= 3.29V - 3.86V

1000= 3.09V - 3.63V

0111= 2.82V - 3.31V

0110= 2.64V - 3.10V

0101= 2.55V - 2.99V

0100= 2.35V - 2.76V

0011= 2.26V - 2.65V

0010= 2.08V - 2.44V

0001= 保留

0000= 保留

注: 没有测试某些 LVDL3:LVDL0 模式下会出现的低于器件有效工作电压的跳变点。

图注:

R= 可读位

W= 可写位

U= 未实现位, 读作 0

-n= 上电复位时的值

1= 置位

0= 清零

x= 未知位

PIC18F2331/2431/4331/4431

21.2 工作原理

通常电压下降相对较慢，这取决于器件电压的电源。这意味着 LVD 模块不需要一直工作。为了降低电流需求，只需要在检测电压时，短时间使能 LVD 电路。检测完成之后，可以禁止 LVD 模块。

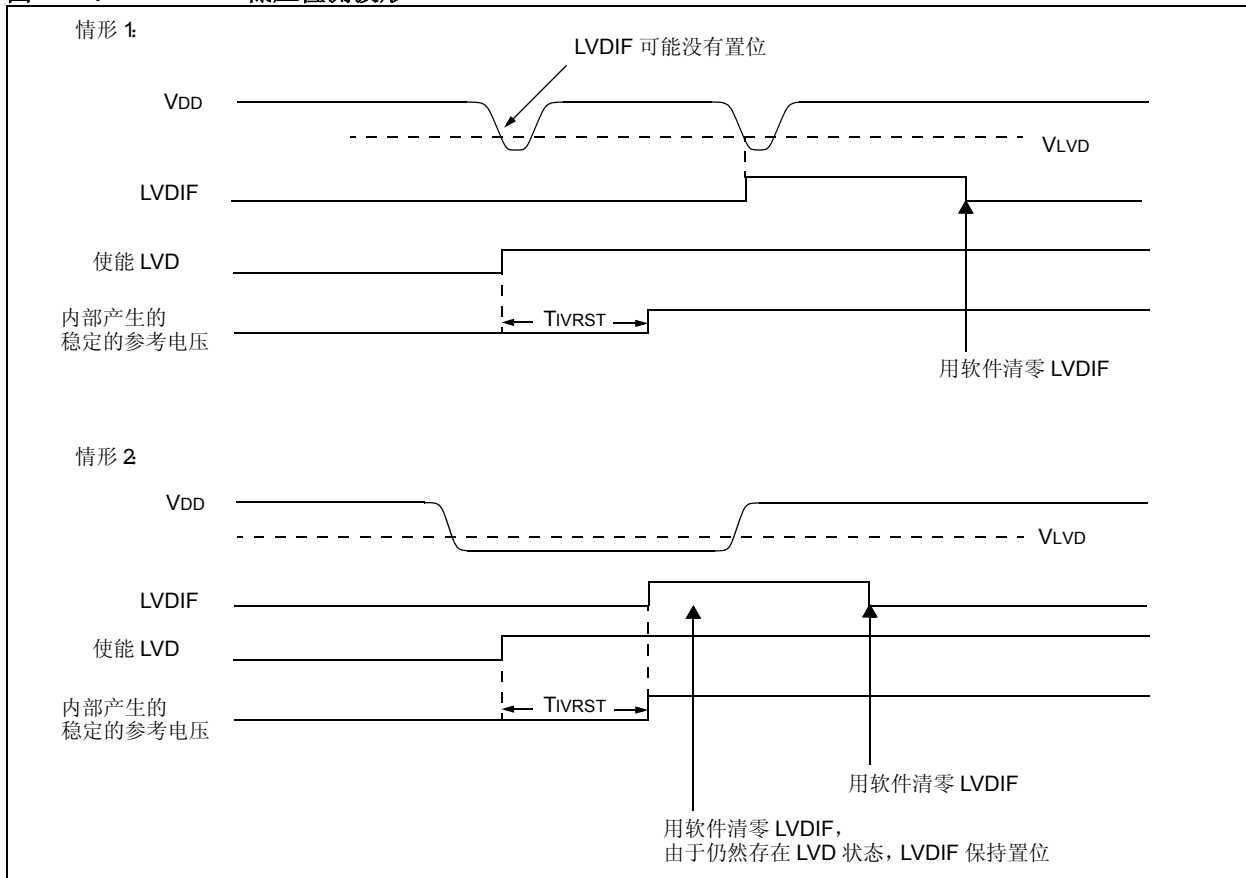
每次使能 LVD 模块时，电路需要过些时间才能稳定下来。当电路稳定之后，可以将所有状态标志都清零。然后该模块会指示正确的系统状态。

需要执行以下步骤来设置 LVD 模块：

1. 将值写入 LVDL3:LVDL0 位 (LVDCON 寄存器) 来选择所需的 LVD 跳变点。
2. 确保 LVD 中断被禁止 (LVDIE 位或 GIE 位已清零)。
3. 使能 LVD 模块 (将 LVDCON 寄存器中的 LVDEN 位置位)。
4. 等待 LVD 模块稳定下来 (等待 IRVST 置位)。
5. 在 LVD 模块稳定前 LVD 中断标志可能被错误置位，应将该标志位清零 (将 LVDIF 位清零)。
6. 允许 LVD 中断 (LVDIE 位和 GIE 位置位)。

图 21-4 所示为可用于低压检测的 LVD 模块的典型波形。

图 21-4: 低压检测波形



21.2.1 参考电压设置点

LVD 模块的内部参考电压可以被其他内部电路（可编程的欠压复位）使用。如果禁止这些电路（降低电流消耗），参考电压电路需要一段时间才能稳定下来，之后才能可靠检测低压状态。这一时间不随系统时钟频率变化。在电气规范参数 36 中指定了稳定所需时间。在达到稳定参考电压后，低压中断标志才会被使能。参见图 21-4 中的波形。

21.2.2 电流消耗

如果使能 LVD 模块，就使能了 LVD 比较器和分压器，并会消耗静态电流。分压器可以从电阻阵列中的多个位置接出。电气规范参数 D022B 中指出了使能时的总电流消耗。

21.3 在休眠模式下工作

使能后，LVD 电路在休眠期间将继续工作。如果器件电压经过跳变点，LVDIF 位将会被置位，并且从休眠状态中唤醒器件。如果已允许全局中断，器件会从中断向量地址处继续执行。

21.4 复位的影响

器件复位强制所有寄存器进入复位状态。这会导致 LVD 模块被关闭。

PIC18F2331/2431/4331/4431

注:

22.0 CPU 的特殊功能

PIC18F2331/2431/4331/4431 器件包含了一些功能，旨在最大限度地提高系统可靠性，并通过去除外部元器件将成本降至最低。这些功能包括：

- 振荡器选择
- 复位：
 - 上电复位 (POR)
 - 上电延时定时器 (PWRT)
 - 振荡器起振定时器 (OST)
 - 欠压复位 (BOR)
- 中断
- 看门狗定时器 (WDT)
- 故障保护时钟监视器
- 双速起振
- 代码保护
- ID 单元
- 在线串行编程 (In-Circuit Serial Programming™, ICSP™)

可以针对应用配置振荡器，如何配置则取决于频率、功耗、精度和成本。第 2.0 节“振荡器配置”对所有选项进行了详细讨论。

关于器件复位和中断的完整讨论可在本数据手册前几章中找到。

除了为复位提供上电延时定时器和振荡器起振定时器之外，PIC18F2331/2431/4331/4431 器件还提供一个看门狗定时器，该定时器可以通过配置位永久使能，也可以用软件控制（如果配置为禁止）。

包含在器件内部的 RC 振荡器对故障保护时钟监视器 (FSCM) 和双速起振有额外的好处。FSCM 对外设时钟提供后台监视，并在外设时钟发生故障时自动切换时钟源。双速起振使得在起振发生后，几乎立即可执行代码，与此同时，主时钟源完成起振延时。

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。

22.1 配置位

可以通过对配置位编程（读作 0）或不编程（读作 1）来选择不同的器件配置。这些配置位从程序存储器单元 300000h 开始映射。

用户会注意到地址 300000h 超出了用户程序存储器空间范围。事实上，它属于配置存储器空间（300000h – 3FFFFFFh），仅能通过读/写表来访问。

在某种意义上，对配置寄存器编程类似于对闪存存储器编程。EECON1 寄存器中的 WR 位可启动对配置寄存器的自定时写操作。在正常工作模式下，执行 TBLWT 指令时，如果 TBLPTR 指向配置寄存器地址，将按照此指令指定的地址和数据写配置寄存器。将 WR 位置位可启动对配置寄存器的长写操作。每次只能往配置寄存器写入一个字节。要写入或擦除一个配置单元，可用 TBLWT 指令分别对该单元写入 1 或 0。关于闪存编程的更多详情，请参阅第 6.5 节“写闪存程序存储器”。

PIC18F2331/2431/4331/4431

表 22-1: 配置位和器件 ID

文件名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	缺省值 / 未编程值	
300000h	CONFIG1L	—	—	—	—	—	—	—	---- ----	
300001h	CONFIG1H	IESO	FCMEN	—	—	FOSC3	FOSC2	FOSC1	FOSC0	11-- 1111
300002h	CONFIG2L	—	—	—	—	BORV1	BORV0	BOREN	PWRTEN	---- 1111
300003h	CONFIG2H	—	—	$\overline{\text{WINEN}}$	WDPS3	WDPS2	WDPS1	WDPS0	WDTEN	---1 1111
300004h	CONFIG3L	—	—	$\overline{\text{T1OSCMX}}$	HPOL	LPOL	PWMPIN	—	—	--11 11--
300005h	CONFIG3H	MCLRE	—	—	EXCLKMX	PWM4MX	SSPMX	—	FLTAMX	1--1 1-11
300006h	CONFIG4L	$\overline{\text{DEBUG}}$	—	—	—	—	LVP	—	STVREN	1--- -1-1
300007h	CONFIG4H	—	—	—	—	—	—	—	—	---- ----
300008h	CONFIG5L	—	—	—	—	CP3	CP2	CP1	CP0	---- 1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah	CONFIG6L	—	—	—	—	WRT3	WRT2	WRT1	WRT0	---- 1111
30000Bh	CONFIG6H	WRD	WRTB	WRTC	—	—	—	—	—	111- ----
30000Ch	CONFIG7L	—	—	—	—	EBTR3	EBTR2	EBTR1	EBTR0	---- 1111
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFEh	DEVID1 ⁽¹⁾	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx ⁽¹⁾
3FFFFFh	DEVID2 ⁽¹⁾	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0000 0101

图注: x= 未知, u= 不变, -= 未实现, q= 其值根据条件变化。
阴影单元格表示未实现位, 读作 0。

注 1: DEVID1 的值参见寄存器 22-13。DEVID 寄存器是只读的, 用户不能对其编程。

寄存器 22-1: CONFIG1H: 配置寄存器 1 高字节 (字节地址 300001h)

R/P-1	R/P-1	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
IESO	FCMEN	—	—	Fosc3	Fosc2	Fosc1	Fosc0
bit 7				bit 0			

bit 7 **IESO:** 内 / 外部切换位

1= 使能内 / 外部切换模式
0= 禁止内 / 外部切换模式

bit 6 **FCMEN:** 故障保护时钟监视器使能位

1= 使能故障保护时钟监视器
0= 禁止故障保护时钟监视器

bit 5-4 **未实现位:** 读作 0

bit 3-0 **Fosc<3:0>:** 振荡器选择位

11xx= 外部 RC 振荡器, RA6 复用为 CLK0
1001= 内部振荡电路, RA6 复用为 CLK0, RA7 复用为端口引脚
1000= 内部振荡电路, RA6 复用为 CLK0, RA7 复用为端口引脚
0111= 外部 RC 振荡器, RA6 复用为端口引脚
0110= HS 振荡器, PLL 使能 (时钟频率 = 4 × Fosc1)
0101= EC 振荡器, RA6 复用为端口引脚
0100= EC 振荡器, RA6 复用为 CLK0
0010= HS 振荡器
0001= XT 振荡器
0000= LP 振荡器

图注:

R= 可读位

P= 可编程位

U= 未实现位, 读作 0

-n= 器件未编程时的值

u= 编程后状态不变

PIC18F2331/2431/4331/4431

寄存器 22-2: **CONFIG2L: 配置寄存器 2 低字节 (字节地址 300002h)**

U-0	U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	—	BORV1	BORV0	BOREN	$\overline{\text{PWRTEN}}$
bit 7				bit 0			

- bit 7-4 **未实现位:** 读作 0
- bit 3-2 **BORV1:BORV0:** 欠压复位电压位
11= 保留
10= VBOR 设置为 2.7V
01= VBOR 设置为 4.2V
00= VBOR 设置为 4.5V
- bit 1 **BOREN:** 欠压复位使能位 ⁽¹⁾
1= 使能欠压复位
0= 禁止欠压复位
- bit 0 **$\overline{\text{PWRTEN}}$:** 上电延时定时器使能位 ⁽¹⁾
1= 禁止 PWRT
0= 使能 PWRT

注 1: BOREN = 1 时, 不会自动将 $\overline{\text{PWRTEN}}$ 改写为 0, 也不会自动使能上电延时定时器。

图注:

R= 可读位	P= 可编程位	U= 未实现位, 读作 0
-n= 器件未编程时的值		u= 编程后状态不变

PIC18F2331/2431/4331/4431

寄存器 22-3: **CONFIG2H: 配置寄存器 2 高字节 (字节地址 300003h)**

U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	—	WINEN	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN
bit 7							bit 0

bit 7-6 **未实现位:** 读作 0

bit 5 **WINEN:** 看门狗定时器窗口使能位

1= 禁止 WDT 窗口
0= 使能 WDT 窗口

bit 4-1 **WDPS<3:0>:** 看门狗定时器后分频选择位

1111 = 1:32,768
1110 = 1:16,384
1101 = 1:8,192
1100 = 1:4,096
1011 = 1:2,048
1010 = 1:1,024
1001 = 1:512
1000 = 1:256
0111 = 1:128
0110 = 1:64
0101 = 1:32
0100 = 1:16
0011 = 1:8
0010 = 1:4
0001 = 1:2
0000 = 1:1

bit 0 **WDTEN:** 看门狗定时器使能位

1= 使能 WDT
0= 禁止 WDT (控制位为 SWDTEN 位)

图注:

R= 可读位

P= 可编程位

U= 未实现位, 读作 0

-n= 器件未编程时的值

u= 编程后状态不变

PIC18F2331/2431/4331/4431

寄存器 22-4: **CONFIG3L: 配置寄存器 3 低字节 (字节地址 300004h)**

U-0	U	R/P-1	R/P-1	R/P-1	R/P-1	U	U
—	—	T1OSCMX	HPOL	LPOL	PWMPIN	—	—
bit 7						bit 0	

bit 7-6 **未实现位:** 读作 0

bit 5 **T1OSCMX:** Timer1 振荡模式位
 1= 低功耗 Timer1 运行模式 (单片机处于休眠模式)。
 0= 标准 (传统) Timer1 振荡器运行模式。

bit 4 **HPOL⁽¹⁾:** 高端晶体管极性位 (即奇数编号 PWM 输出的极性控制位)
 1= PWM 1、3、5 和 7 高电平有效 (缺省)
 0= PWM 1、3、5 和 7 低电平有效

bit 3 **LPOL⁽¹⁾:** 低端晶体管极性位 (即偶数编号 PWM 输出极性控制位)
 1= PWM 0、2、4 和 6 高电平有效 (缺省)
 0= PWM 0、2、4 和 6 低电平有效

bit 2 **PWMPIN⁽²⁾:** PWM 输出引脚复位状态控制位
 1= 复位时, PWM 输出禁止 (缺省)
 0= 复位时, PWM 输出驱动为有效状态 ⁽³⁾

bit 1-0 **未实现位:** 读作 0

注 1: 极性控制位 HPOL 和 LPOL 定义了 PWM 输出信号的有效和无效状态; 错误输入或人工改写 PWM 都可以改变 PWM 的状态。

2: PWM6 和 PWM7 输出通道仅在 PIC18F4X21 器件上可用。

3: 当 PWMPIN=0 时, 如果器件有 8 个 PWM 输出引脚 (40 和 44 引脚器件), 则 PWMEN<2:0>=101; 如果器件有 6 个 PWM 输出引脚 (28 引脚器件), 则 PWMEN<2:0>=100。PWM 的输出极性由 HPOL 和 LPOL 定义。

图注:

R= 可读位

P= 可编程位

U= 未实现位, 读作 0

-n= 器件未编程时的值

u= 编程后状态不变

PIC18F2331/2431/4331/4431

寄存器 22-5: **CONFIG3H: 配置寄存器 3 高字节 (字节地址 30005h)**

R/P-1	U	U	R/P-1	R/P-1	R/P-1	U	R/P-1	
MCLRE	—	—	EXCLKMX ⁽¹⁾	PWM4MX ⁽¹⁾	SSPMX ⁽¹⁾	—	FLTAMX ⁽¹⁾	
bit 7								bit 0

- bit 7 **MCLRE:** $\overline{\text{MCLR}}$ 引脚使能位
 1= 使能 RE3 引脚; 禁止 MCLR。
 0= 使能 MCLR 引脚; 禁止 RE3 输入引脚。
- bit 6-5 **未实现位:** 读作 0
- bit 4 **EXCLKMX:** TMR0/T5CKI 外部时钟复用选择位
 1= TMR0/T5CKI 外部时钟输入与 RC3 复用
 0= TMR0/T5CKI 外部时钟输入与 RD0 复用
- bit 3 **PWM4MX:** PWM4 复用选择位
 1= PWM4 输出与 RB5 复用
 0= PWM4 输出与 RD5 复用
- bit 2 **SSPMX:** SSP I/O 复用选择位
 1= SCK/SCL 时钟引脚和 SDA/SDI 数据引脚分别与 RC5 和 RC4 复用。
 SDO 输出与 RC7 复用。
 0= SCK/SCL 时钟引脚和 SDA/SDI 数据引脚分别与 RD3 和 RD2 复用。
 SDO 输出与 RD1 复用。
- bit 1 **未实现位:** 读作 0
- bit 0 **FLTAMX:** $\overline{\text{FLTA}}$ 复用选择位
 1= $\overline{\text{FLTA}}$ 输入引脚与 RC1 复用
 0= $\overline{\text{FLTA}}$ 输入引脚与 RD4 复用

注 1: PIC18F2X31 器件中未实现; 保持该位置位。

图注:

R= 可读位	P= 可编程位	U= 未实现位, 读作 0
-n= 器件未编程时的值		u= 编程后状态不变

PIC18F2331/2431/4331/4431

寄存器 22-6:

CONFIG4L: 配置寄存器 4 低字节 (字节地址 300006h)

R/P-1	U-0	U-0	U-0	U-0	R/P-1	U-0	R/P-1
DEBUG	—	—	—	—	LVP	—	STVREN

bit 7 bit 0

- 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 6-3 **未实现位**: 读作 0
- bit 2 **LVP**: 低压 ICSP 使能位
1= 使能低压 ICSP
0= 禁止低压 ICSP
- bit 1 **未实现位**: 读作 0
- bit 0 **STVREN**: 堆栈满 / 下溢复位使能位
1= 堆栈满 / 下溢会导致复位
0= 堆栈满 / 下溢不会导致复位

图注:

R= 可读位 C= 可清零位 U= 未实现位, 读作 0
-n= 器件未编程时的值 u= 编程后状态不变

PIC18F2331/2431/4331/4431

寄存器 22-7: CONFIG5L: 配置寄存器 5 低字节 (字节地址 300008h)

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
—	—	—	—	CP3 ⁽¹⁾	CP2 ⁽¹⁾	CP1	CP0
bit 7				bit 0			

- bit 7-4 **未实现位:** 读作 0
- bit 3 **CP3:** 代码保护位
1=Block3 (001800 到 001FFFh) 无代码保护
0=Block3 (001800 到 001FFFh) 有代码保护
- bit 2 **CP2:** 代码保护位
1=Block2 (001000 到 0017FFh) 无代码保护
0=Block2 (001000 到 0017FFh) 有代码保护
- bit 1 **CP1:** 代码保护位
1=Block1 (000800 到 000FFFh) 无代码保护
0=Block1 (000800 到 000FFFh) 有代码保护
- bit 0 **CP0:** 代码保护位
1=Block0 (000200 到 0007FFh) 无代码保护
0=Block0 (000200 到 0007FFh) 有代码保护

注 1: PIC18F2X31 器件中未实现; 保持该位置位。

图注:

R= 可读位 C= 可清零位 U= 未实现位, 读作 0
-n= 器件未编程时的值 u= 编程后状态不变

寄存器 22-8: CONFIG5H: 配置寄存器 5 高字节 (字节地址 300009h)

R/C-1	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
CPD	CPB	—	—	—	—	—	—
bit 7				bit 0			

- bit 7 **CPD:** 数据 EEPROM 代码保护位
1= 数据 EEPROM 无代码保护
0= 数据 EEPROM 有代码保护
- bit 6 **CPB:** 引导块代码保护位
1= 引导块 (000000 到 0001FFh) 无代码保护
0= 引导块 (000000 到 0001FFh) 有代码保护
- bit 5-0 **未实现位:** 读作 0

图注:

R= 可读位 C= 可清零位 U= 未实现位, 读作 0
-n= 器件未编程时的值 u= 编程后状态不变

PIC18F2331/2431/4331/4431

寄存器 22-9: CONFIG6L: 配置寄存器 6 低字节 (字节地址 3000Ah)

U-0	U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	—	WRT3 ⁽¹⁾	WRT2 ⁽¹⁾	WRT1	WRT0
bit 7				bit 0			

- bit 7-4 **未实现位:** 读作 0
- bit 3 **WRT3:** 写保护位⁽¹⁾
 1=Block3 (001800 到 001FFFh) 无写保护
 0=Block3 (001800 到 001FFFh) 有写保护
- bit 2 **WRT2:** 写保护位⁽¹⁾
 1=Block2 (001000 到 0017FFh) 无写保护
 0=Block2 (001000 到 0017FFh) 有写保护
- bit 1 **WRT1:** 写保护位
 1=Block1 (000800 到 000FFFh) 无写保护
 0=Block1 (000800 到 000FFFh) 有写保护
- bit 0 **WRT0:** 写保护位
 1=Block0 (000200 到 0007FFh) 无写保护
 0=Block0 (000200 到 0007FFh) 有写保护

注 1: PIC18F2X31 器件中未实现; 保持该位置位。

图注:

R= 可读位 P= 可编程位 U= 未实现位, 读作 0
 -n= 器件未编程时的值 u= 编程后状态不变

寄存器 22-10: CONFIG6H: 配置寄存器 6 高字节 (字节地址 3000Bh)

R/P-1	R/P-1	R-1	U-0	U-0	U-0	U-0	U-0
WRTD	WRTB	WRTC	—	—	—	—	—
bit 7			bit 0				

- bit 7 **WRTD:** 数据 EEPROM 写保护位
 1= 数据 EEPROM 无写保护
 0= 数据 EEPROM 有写保护
- bit 6 **WRTB:** 引导块写保护位
 1= 引导块 (000000 到 0001FFh) 无写保护
 0= 引导块 (000000 到 0001FFh) 有写保护
- bit 5 **WRTC:** 配置寄存器写保护位
 1= 配置寄存器 (300000 到 3000FFh) 无写保护
 0= 配置寄存器 (300000 到 3000FFh) 有写保护
- 注:** 在正常执行模式下, 该位只读; 仅在编程模式下可写该位。
- bit4-0 **未实现位:** 读作 0

图注:

R= 可读位 P= 可编程位 U= 未实现位, 读作 0
 -n= 器件未编程时的值 u= 编程后状态不变

PIC18F2331/2431/4331/4431

寄存器 22-11: CONFIG7L: 配置寄存器 7 低字节 (字节地址 3000Ch)

U-0	U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	—	EBTR3 ⁽¹⁾	EBTR2 ⁽¹⁾	EBTR1	EBTR0
bit 7				bit 0			

bit 7-4 **未实现位:** 读作 0

bit 3 **EBTR3:** 读表保护位 ⁽¹⁾

1=Block3 (001800 到 001FFFh) 未读保护, 可从其他块执行读表操作
0=Block3 (001800 到 001FFFh) 读保护, 不能从其他块执行读表操作

bit 2 **EBTR2:** 读表保护位 ⁽¹⁾

1=Block2 (001000 到 0017FFh) 未读保护, 可从其他块执行读表操作
0=Block2 (001000 到 0017FFh) 读保护, 不能从其他块执行读表操作

bit 1 **EBTR1:** 读表保护位

1=Block1 (000800 到 000FFFh) 未读保护, 可从其他块执行读表操作
0=Block1 (000800 到 000FFFh) 读保护, 不能从其他块执行读表操作

bit 0 **EBTR0:** 读表保护位

1=Block0 (000200 到 0007FFh) 未读保护, 可从其他块执行读表操作
0=Block0 (000200 到 0007FFh) 读保护, 不能从其他块执行读表操作

注 1: PIC18F2X31 器件中未实现; 保持该位置位。

图注:

R= 可读位

P= 可编程位

U= 未实现位, 读作 0

-n= 器件未编程时的值

u= 编程后状态不变

寄存器 22-12: CONFIG7H: 配置寄存器 7 高字节 (字节地址 3000Dh)

U-0	R/P-1	U-0	U-0	U-0	U-0	U-0	U-0
—	EBTRB	—	—	—	—	—	—
bit 7				bit 0			

bit 7 **未实现位:** 读作 0

bit 6 **EBTRB:** 引导块读表保护位

1= 引导块 (000000 到 0001FFFh) 未读保护, 可从其他块执行读表操作
0= 引导块 (000000 到 0001FFFh) 读保护, 不能从其他块执行读表操作

bit 5-0 **未实现位:** 读作 0

图注:

R= 可读位

P= 可编程位

U= 未实现位, 读作 0

-n= 器件未编程时的值

u= 编程后状态不变

PIC18F2331/2431/4331/4431

寄存器 22-13: PIC18F2331/2431/4331/4431 器件的器件 ID 寄存器 1

R	R	R	R	R	R	R	R	
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	
bit 7								bit 0

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> 结合使用, 以确定器件编号。

000= PIC18F4331

001= PIC18F4431

100= PIC18F2331

101= PIC18F2431

bit 4-0 **REV<4:0>**: 版本 ID 位

这些位用于表明器件版本。

图注:

R= 只读位

P= 可编程位

U= 未实现位, 读作 0

-n= 器件未编程时的值

u= 编程后状态不变

寄存器 22-14: PIC18F2331/2431/4331/4431 器件的器件 ID 寄存器 2

R	R	R	R	R	R	R	R	
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	
bit 7								bit 0

bit 7-0 **DEV10:DEV3**: 器件 ID 位

这些位与器件 ID 寄存器 1 中的 DEV2:DEV0 结合使用, 以确定器件编号。

0000 0101= PIC18F2331/2431/4331/4431 器件

注 1: DEV10:DEV3 的这些值可能与其他器件共用。完整的 DEV10:DEV0 位序列总是标示了某个特定的器件。

图注:

R= 只读位

P= 可编程位

U= 未实现位, 读作 0

-n= 器件未编程时的值

u= 编程后状态不变

PIC18F2331/2431/4331/4431

22.2 看门狗定时器 (WDT)

对于 PIC18F2331/2431/4331/4431 器件, WDT 由 INTRC 源驱动。当 WDT 使能时, 时钟源也被使能。WDT 的标称值为 4 ms, 与 INTRC 振荡器的稳定性相同。

4 ms 周期的 WDT 可用 16 位后分频器倍乘。WDT 后分频器的任何输出通过多路转换器选定, 由配置寄存器 2H 中的位控制 (见寄存器 22-3)。可用的周期范围从 4 ms 到 131.072 秒 (2.18 分钟)。当以下任一事件发生时, WDT 和后分频器将被清零, 包括执行 SLEEP 或 CLRWDT 指令、改变 IRCF 位 (OSCCON<6:4>) 或发生时钟故障 (见第 22.4.1 节 “FSCM 和看门狗定时器”)。

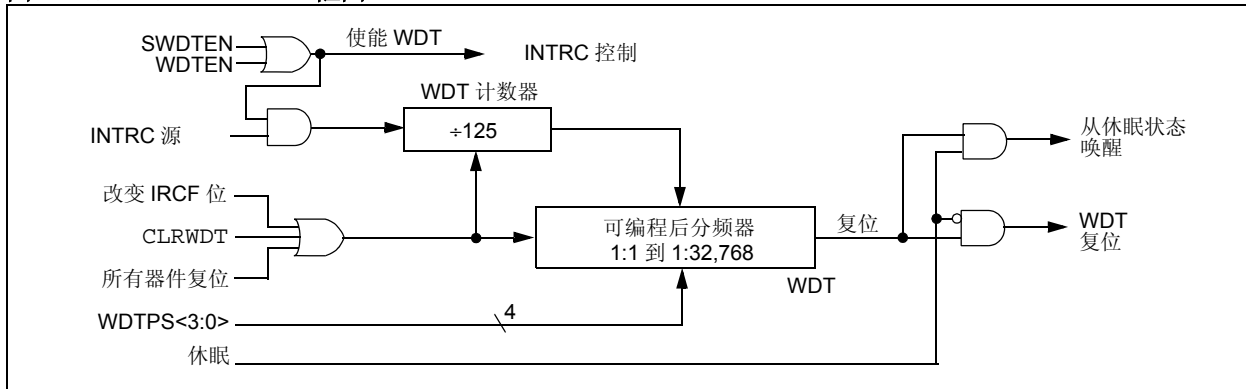
使用 OSCTUNE 寄存器调节内部振荡器时钟周期也会以同样的因子影响 WDT 的周期。例如, 如果 INTRC 的周期延长 3%, 则 WDT 的周期随之延长 3%。

- 注 1:** 当执行 CLRWDT 和 SLEEP 指令时, WDT 和后分频器计数将被清零。
- 注 2:** 通过更改 IRCF 位 (OSCCON<6:4>) 的设置可以清零 WDT 和后分频器计数。
- 注 3:** 当执行 CLRWDT 指令时, 后分频器计数将被清零。
- 注 4:** 如果 WINEN=0, 则仅当 WDTW=1 时, CLRWDT 指令才可执行; 否则, 将导致器件复位。

22.2.1 控制寄存器

寄存器 22-15 给出了 WDTCON 寄存器。该寄存器可读写。仅当配置位禁止 WDT 时, SWDTEN 位才允许软件使能或禁止 WDT。WDTW 位是只读位, 该位表示 WDT 计数是否处于第四段 (也就是说, 是否 8 位的 WDT 值达到二进制数 11000000 或更大)。

图 22-1: WDT 框图



寄存器 22-15: WDTCON 寄存器

R-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
WDTW	-	-	-	-	-	-	SWDTEN
bit 7							bit 0

bit 7 **WDTW:** 看门狗定时器窗口位
1 = WDT 计数处于第四段
0 = WDT 计数不处于第四段

bit 6 **未实现位**

bit 0 **SWDTEN:** 软件使能 / 禁止看门狗定时器位 (1)
1 = WDT 打开
0 = WDT 关闭

注 1: 如果 WDTEN 配置位为 1, 则 WDT 总是使能, 与 SWDTEN 控制位的状态无关。如果 WDTEN 配置位为 0, 则可以使用 SWDTEN 控制位打开 / 关闭 WDT。

图注:

R= 可读位

W= 可写位

U= 未实现位, 读作 0

-n= 上电复位时的值

表 22-2: 看门狗定时器的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG2H	—	—	WINEN	WDTPS3	WDTPS2	WDTPS2	WDTPS0	WDTEN
RCON	IPEN	—	—	RI	TO	PD	POR	BOR
WDTCON	WDTW	—	—	—	—	—	—	SWDTEN

图注: 阴影单元格表示看门狗定时器未使用。

22.3 双速起振

双速起振功能允许单片机在主时钟源可用之前, 使用 INTRC 振荡器作为时钟源, 从而帮助器件最大限度地缩短振荡器起振到代码执行间的延时。通过将配置寄存器 1H 中的 IESO 位 (CONFIG1H<7>) 置位可以使能该功能。

仅当主振荡器模式为 LP、XT、HS 或 HSPLL (基于晶振的模式) 时双速起振才可用。其他源不需要 OST 起振延时; 对于这些时钟源, 双速起振禁止。

当双速起振使能时, 器件复位或从休眠状态唤醒时, 器件将配置为使用内部振荡电路作为时钟源, 接着, 当上电复位使能后, 将发生上电延时定时器延时。这样允许代码执行几乎与主振荡器起振、OST 运行同时发生。一旦 OST 延时结束, 器件自动换到 PRI_RUN 模式。

因为 OSCCON 寄存器在发生复位事件时会被清零, INTOSC (或后分频器) 时钟源在复位发生后不是立即可用, INTRC 时钟源可以直接以基频使用。为了在唤醒器件时使用更高的时钟速度, 通过在复位后立即将 IFRC2:IFRC0 位置位, 选择 INTOSC 或其后分频器时钟源提供更高的时钟速度。对于从休眠模式唤醒的情况, 可以通过在进入休眠模式之前置位 IFRC2:IFRC0 位来选择 INTOSC 或其后分频时钟源。

对于所有其他的功耗管理模式, 不使用双速起振。器件将使用当前选择的时钟源直到主时钟源可用为止。IESO 位的设置被忽略。

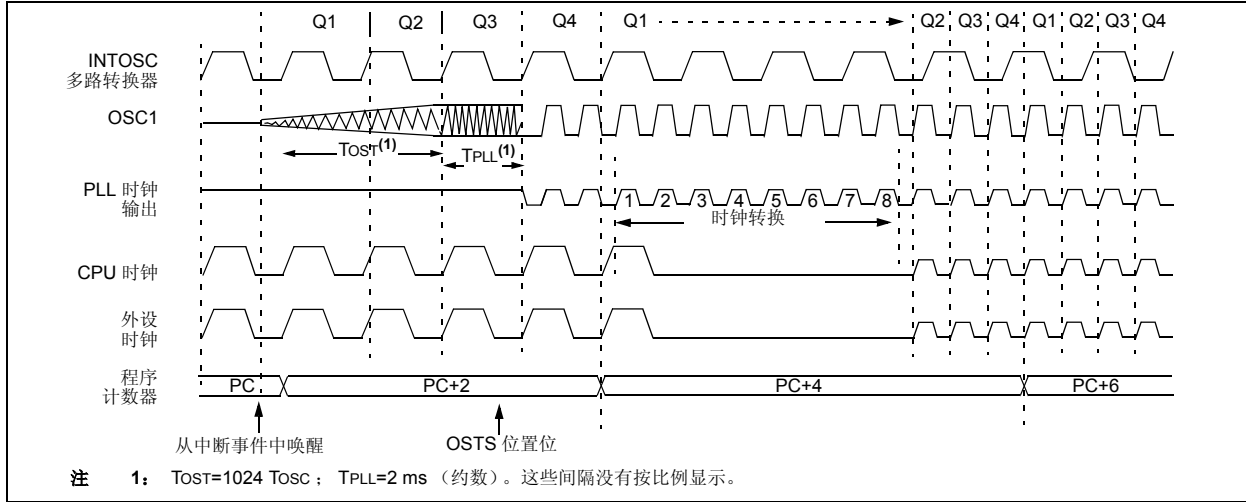
22.3.1 使用双速起振的特别注意事项

当在双速起振中使用 INTRC 振荡器时, 器件仍遵守进入功耗管理模式的正常命令序列, 包括连续的 SLEEP 指令 (见第 3.1.3 节 “多种休眠命令”)。实际上, 这意味着在 OST 延时结束之前, 用户代码可以改变 SCS1:SCS0 位的设置, 并且发出 SLEEP 指令。这就允许应用程序短暂唤醒器件, 执行 “现场保护” 任务, 并在器件开始使用主振荡器运行之前返回休眠状态。

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为当前系统时钟。如果该位置位, 则主振荡器为当前系统时钟。否则, 当器件从复位或休眠模式唤醒期间, 由内部振荡电路为系统提供时钟。

PIC18F2331/2431/4331/4431

图 22-2: 双速起振的转换时序 (从 INTOSC 到 HSPLL)

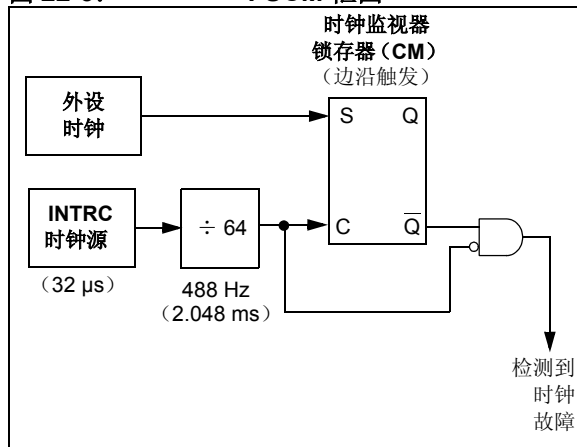


22.4 故障保护时钟监视器

故障保护时钟监视器（FSCM）允许单片机在发生外部振荡器故障时，自动地将系统时钟切换为内部振荡电路，以保持器件继续运行。通过将故障保护时钟监视器使能位 FCMEN（CONFIG1H<6>）置位，可以使能 FSCM 功能。

当 FSCM 使能时，INTRC 振荡器将一直保持运行以监视外设时钟，并且在发生外设时钟故障时立即用作备用时钟。时钟监视（如图 22-3 所示）通过创建一个采样时钟信号实现，该信号为 INTRC 输出的 64 分频。这样就使得 FSCM 采样时钟沿与外设时钟沿有充足的时间间隔。外设系统时钟和采样时钟作为时钟监视器锁存器（Clock Monitor latch, CM）的输入。CM 在系统时钟源的下降沿被置位，在采样时钟的上升沿被清零。

图 22-3: FSCM 框图



时钟故障在采样时钟的下降沿被检测。如果采样时钟出现下降沿时，CM 仍然置位，则检测到外部时钟故障（图 22-4）。这将引发以下事件：

- 通过置位 OSCFIF 位（PIR2<7>），FSCM 产生振荡器故障中断；
- 系统时钟源切换为内部振荡电路（OSCCON 不会被更新，所以无法显示当前时钟源，这就是故障保护状态）；并发生
- WDT 复位。

由于来自内部振荡电路的后分频器频率可能不够稳定，所以最好能选择另一个时钟配置并进入一个备用功耗管理模式（更多细节请参阅第 22.3.1 节“使用双速起振的特别注意事项”和第 3.1.3 节“多种休眠命令”）。如果要尝试部分恢复或执行受控关闭，可以这样做。

为了在唤醒器件时使用更高的时钟速度，可以通过在复位发生后立即设置 IFRC2:IFRC0 位，选择 INTOSC 或后分频器时钟源提供更高的时钟速度。对于从休眠模式唤醒的情况，可以通过在进入休眠模式之前设置 IFRC2:IFRC0 位来选择 INTOSC 或后分频器时钟源。

使用 OSCTUNE 寄存器调节内部振荡电路也会以同样的因子影响 FSCM 的周期，但这种影响通常被忽略，因为被监视的时钟频率通常比采样时钟频率高得多。

FSCM 只能检测出主时钟源或辅助时钟源的故障。如果内部振荡电路发生故障时，无法检测到其他时钟故障，当然也不可能采取任何措施。

22.4.1 FSCM 和看门狗定时器

FSCM 和 WDT 均以 INTRC 振荡器作为时钟源。由于 WDT 使用独立的分频器和计数器，当 FSCM 使能时，禁止 WDT 对 INTRC 振荡器的运行没有影响。

如上所述，当检测到时钟故障时，时钟源将切换到 INTOSC 时钟。根据 IRCF2:IRCF0 位选择的频率的不同，代码执行速度也会相应发生很大的变化。如果用小预分频值的时钟源使能 WDT，时钟速度的下降将允许 WDT 发生超时，以及随后的器件复位。由于这个原因，故障保护时钟事件也会复位 WDT 和后分频器，使 WDT 从执行速度发生变化的时刻开始重新计数，从而减少发生错误超时的可能性。

22.4.2 退出故障安全运行

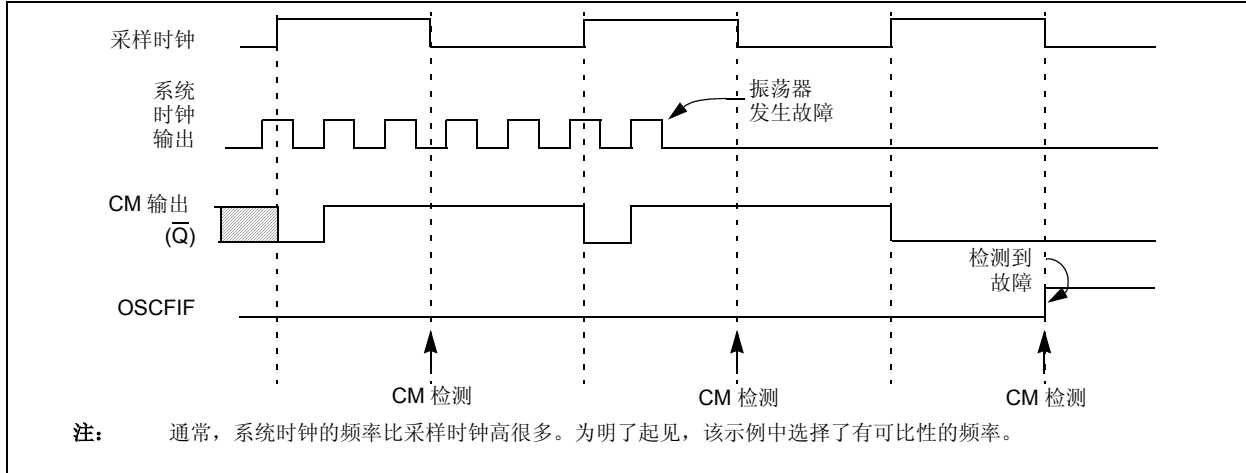
故障保护状态可以通过器件复位或进入功耗管理模式来终止。发生复位时，控制器启动配置寄存器 1H 中指定的主时钟源（将发生所有该振荡模式所需的起振延时，诸如 OST 或 PLL 定时器延时）。INTOSC 多路转换器将提供系统时钟直到主时钟源就绪为止（类似于双速起振）。此时系统时钟源切换到主时钟（用 OSCCON 寄存器中的 OSTS 位置位表示）。然后，故障保护时钟监视器恢复对外设时钟的监视。

在起振期间，主时钟源可能永远不就绪。如果是这样，器件运行将以 INTOSC 多路转换器作为时钟源。OSCCON 寄存器将保持复位状态直至进入功耗管理模式。

通过装载 OSCCON 寄存器，并执行 SLEEP 指令进入功耗管理模式将会清零故障保护状态。当故障保护状态清零后，时钟监视器将重新开始监视外设时钟。

PIC18F2331/2431/4331/4431

图 22-4: FSCM 时序图



22.4.3 功耗管理模式下的 FSCM 中断

正如前面提到的，进入功耗管理模式将会清零故障保护状态。进入功耗管理模式时，时钟多路转换器选择由 OSCCON 寄存器选定的时钟源。在功耗管理模式下将恢复对功耗管理时钟源的故障保护监视。

如果在功耗管理运行期间发生振荡器故障，随后将会发生的事件取决于振荡器故障中断是否使能。如果使能 (OSCFIF = 1)，代码执行将以 INTOSC 多路转换器作为时钟源。不会自动转回到发生故障的时钟源。

如果该中断禁止，当发生振荡器故障时，器件将不会退出功耗管理模式。相反，器件将继续像以前一样运行，但是以 INTOSC 多路转换器作为时钟源。当处于空闲模式时，随后发生的中断将导致 CPU 开始执行指令，此时 CPU 以 INTOSC 多路转换器作为时钟。器件将在故障保护状态清零时才会切换到另一个不同的时钟源。

22.4.4 上电复位或从休眠中唤醒

设计 FSCM 的目的是在器件退出上电复位 (POR) 或低功耗休眠模式后的任意时刻，检测振荡器故障。当系统主时钟为 EC、RC 或 INTRC 模式时，监视会在这些事件发生后立即开始。

对于涉及到晶振或谐振器的振荡模式 (如 HS、HSPLL、LP 或 XT)，情况稍有不同。由于这类振荡器需要的起振时间可能比 FSCM 采样时钟的时间长很多，所以有可能检测到假的时钟故障。为了防止这一问题，内部振荡电路会被自动配置为系统时钟，并工作到主时钟稳定为止 (OST 和 PLL 定时器已完成延时)。这与双速起振模式相同。一旦主时钟稳定下来，INTRC 重新作为 FSCM 时钟源。

注：防止在发生 POR 或从休眠状态唤醒时检测到假振荡器故障中断的逻辑，同样也可以防止在发生这些事件后启动对振荡器故障的检测。避免这种情况的方法是：监视 OST 位，并使用定时程序来确定振荡器起振所需时间是否过长。虽然如此，所有振荡器故障中断都不会留下标志。

正如第 22.3.1 节“使用双速起振的特别注意事项”中所述，在等待系统主时钟稳定的过程中，选择另一个时钟配置并进入一个备用的功耗管理模式也是可能的。当选择了新的功耗管理模式时，主时钟将被禁止。

PIC18F2331/2431/4331/4431

22.5 程序校验和代码保护

PIC18 闪存器件的整个代码保护结构与其他 PICmicro® 系列器件显著不同。

用户程序存储器被分为五块。其中的引导块有 512 字节。存储器的剩余部分根据二进制边界被分为四块。

这五块中的每一个都有三位代码保护位与之相关联。它们是：

- 代码保护位 (CPn)
- 写保护位 (WRTn)
- 外部存储块读表位 (EBTRn)

图 22-5 给出了 8KB 和 16KB 器件的程序存储器构成，以及与每个块相关的指定代码保护位。这些位的真实位置汇总在表 22-3 中。

图 22-5: PIC18F2331/2431/4331/4431 的受代码保护的程序存储器

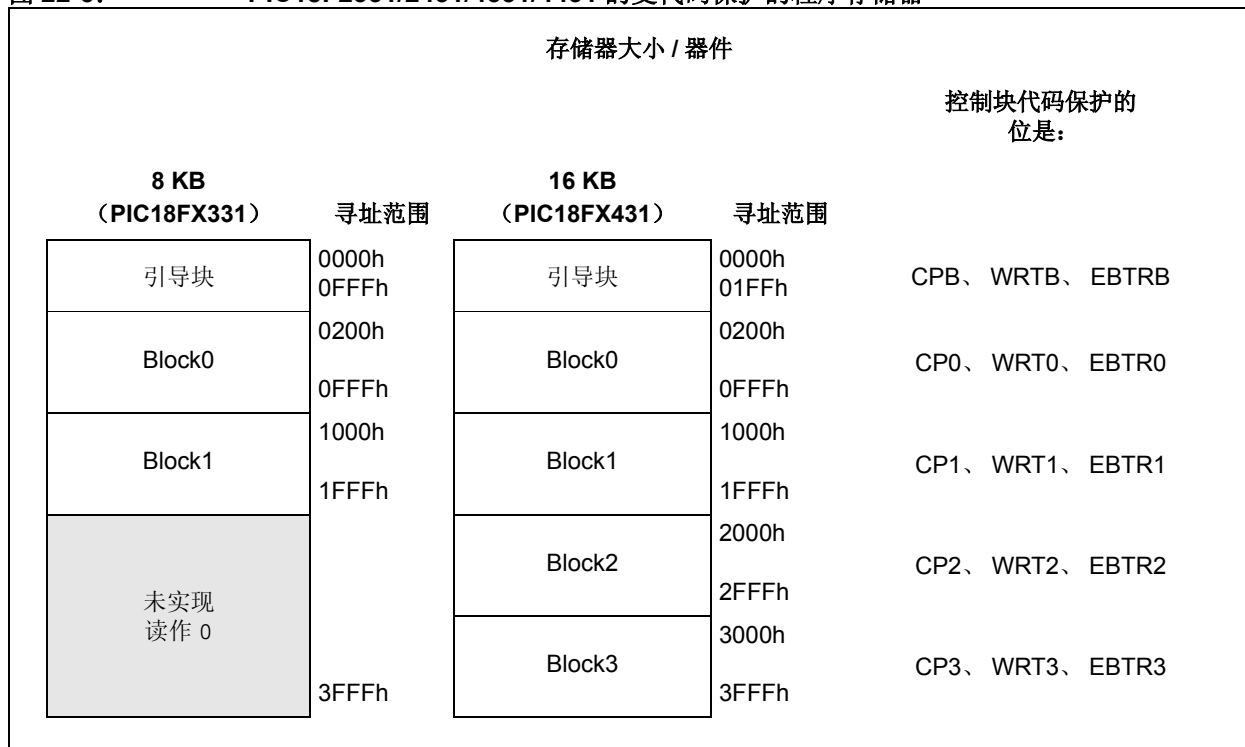


表 22-3: 代码保护寄存器汇总

文件名称		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
300008h	CONFIG5L	—	—	—	—	CP3	CP2	CP1	CP0
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—
30000Ah	CONFIG6L	—	—	—	—	WRT3	WRT2	WRT1	WRT0
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—
30000Ch	CONFIG7L	—	—	—	—	EBTR3	EBTR2	EBTR1	EBTR0
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—

图注: 阴影单元格未实现。

PIC18F2331/2431/4331/4431

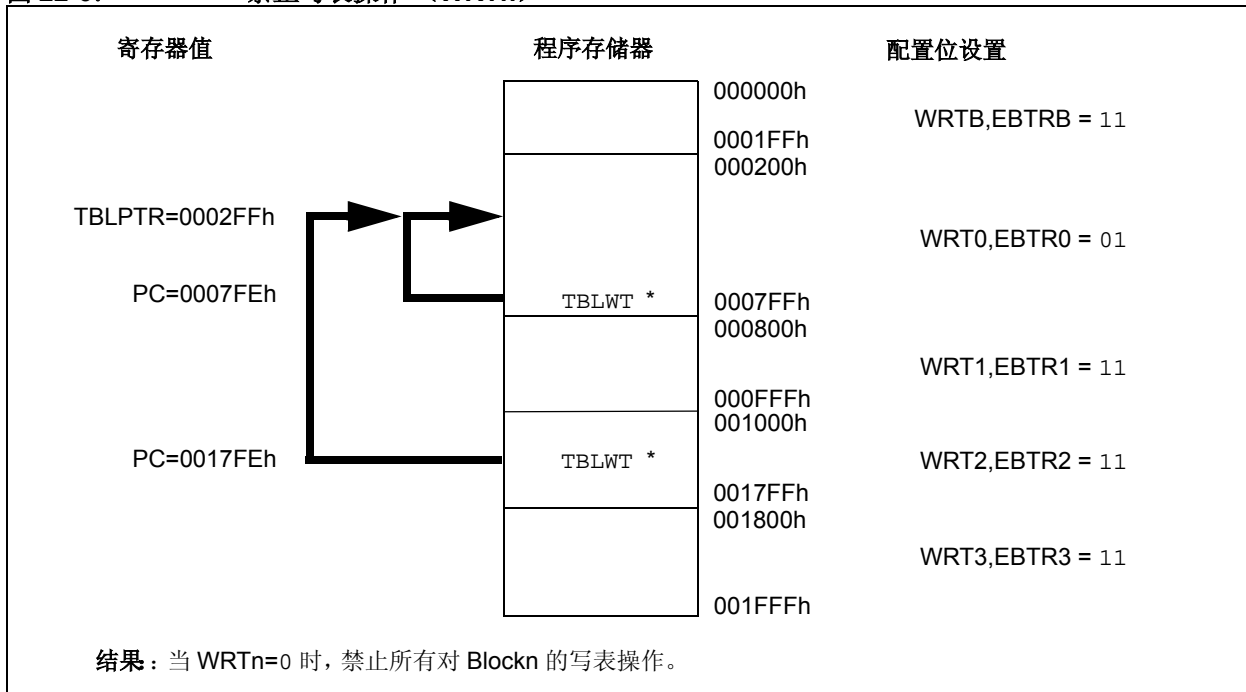
22.5.1 程序存储器代码保护

使用读 / 写表指令可以对程序存储器的任何单元进行读写操作。读表操作会读取器件 ID。使用读 / 写表指令可以对配置寄存器进行读写操作。

在正常执行模式下，CPn 位不产生直接的作用。CPn 位限制了外部的读写操作。如果 WRTn 配置位是“0”，可以保护用户存储器的某个块不受写表指令的影响。EBTRn 位控制读表操作。如果某一用户存储块的 EBTRn 位置为“0”，该块内的读表指令就允许执行读取操作。而来自该块以外的读表指令则不被允许，读出的结果为“0”。图 22-6 到 22-8 举例说明了读 / 写表保护。

注： 代码保护位仅能从“1”状态改写为“0”状态。而不可能从“0”状态改写到“1”状态。只有使用整个芯片擦除或块擦除功能才能将代码保护位置“1”。而整个芯片或块擦除功能仅能通过 ICSP 或外部编程器启动。

图 22-6: 禁止写表操作 (WRTn)



PIC18F2331/2431/4331/4431

图 22-7: 禁止外部存储块读表操作 (EBTRn)

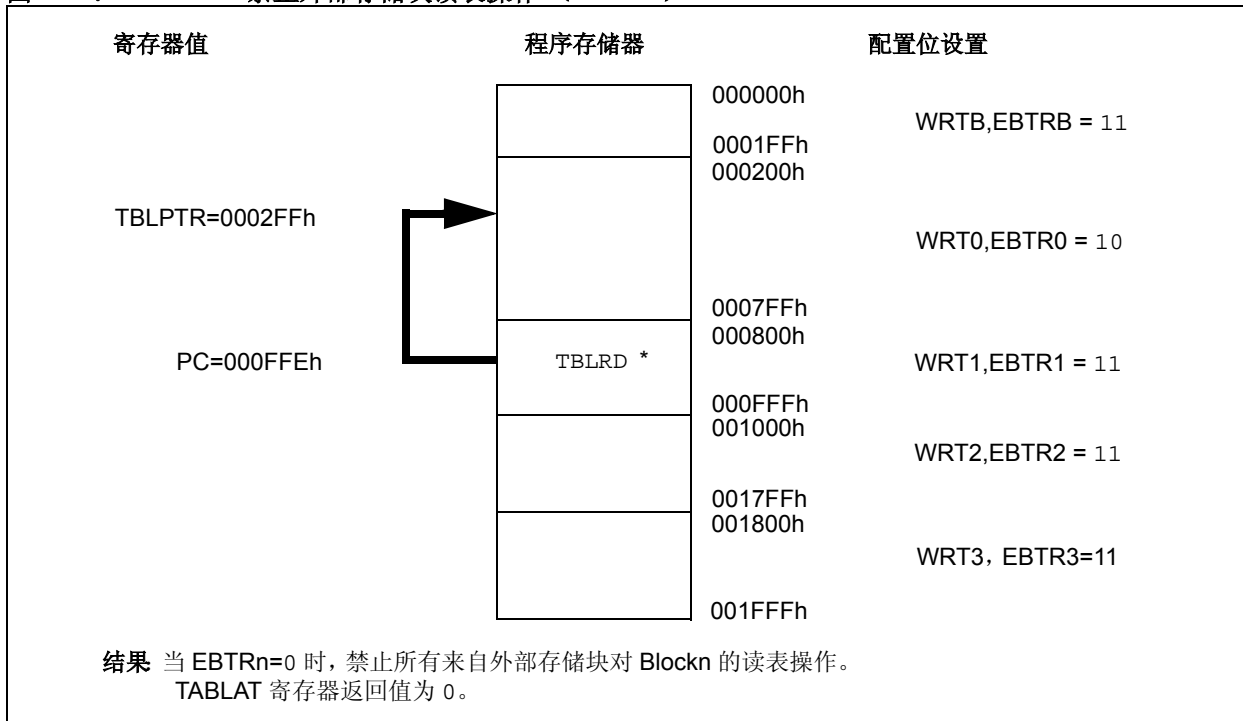
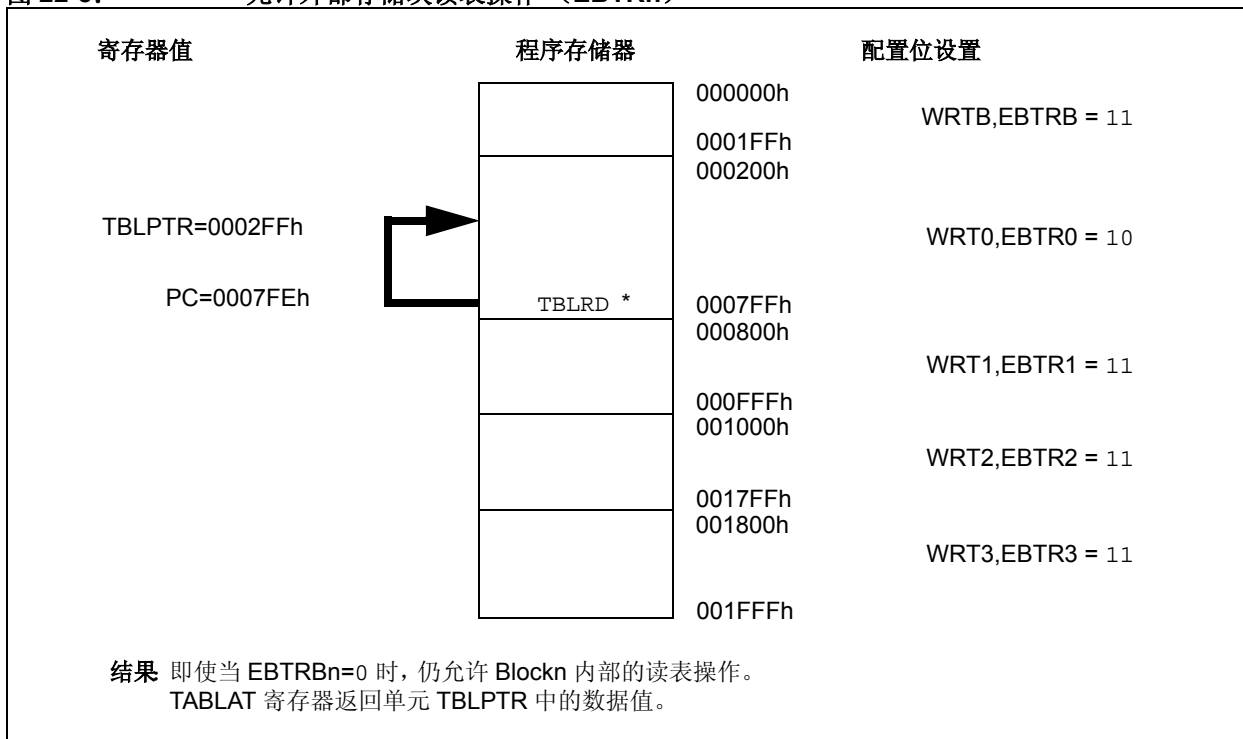


图 22-8: 允许外部存储块读表操作 (EBTRn)



PIC18F2331/2431/4331/4431

22.5.2 数据 EEPROM 代码保护

整个数据 EEPROM 的外部读写由两个配置位 CPD 和 WRD 保护。CPD 禁止对数据 EEPROM 的外部读写操作。WRD 禁止对数据 EEPROM 的外部写操作。CPU 可以不受这些保护位的设置影响，继续读写数据 EEPROM。

22.5.3 配置寄存器保护

配置寄存器可以写保护。WRTC 位控制配置寄存器的保护。在正常执行模式下，WRTC 位只读。WRTC 仅能通过 ICSP 或外部编程器修改。

22.6 ID 单元

八个存储单元 (20000h-20007h) 指定为 ID 单元，供用户存储校验和或其他标识代码编号。在正常运行或编程 / 校验期间，可以用 TBLRD 和 TBLWT 指令读写这些单元。当器件有代码保护时，可以读取 ID 单元。

22.7 在线串行编程

PIC18F2331/2431/4331/4431 单片机可以在最终应用电路中进行串行编程。ICSP 只需使用五根线，其中时钟和数据线各 1 根，另外三根线作为电源、接地和编程电压线。这就允许用户可生产带有未编程器件的电路板，仅在产品交付前对单片机进行编程。同样允许对最新的或定制的固件进行编程。

22.8 在线调试器

当配置寄存器 CONFIG4L 中的 DEBUG 位被编程为“0”时，在线调试功能使能。这一功能在同 MPLAB® IDE 一起使用的时候，就能进行一些简单的调试。当单片机的这项功能使能时，某些资源就不再是通用的了。表 22-4 给出了后台调试器所需的资源。

表 22-4: 调试器资源

I/O 引脚:	RB6, RB7
堆栈:	2 级
程序存储器:	512 字节
数据存储器:	10 字节

要使用单片机的在线调试器功能，就必须实现 MCLR/VPP、VDD、VSS、RB7 和 RB6 在线串行编程连接的设计。这可以与 Microchip 或第三方开发工具公司提供的在线调试器模块相连。

22.9 低压 ICSP 编程

配置寄存器 4L 中的 LVP 位 (CONFIG4L<2>) 使能低压 ICSP 编程 (LVP)。当 LVP 使能时，单片机无需在 MCLR/VPP 引脚加上高电压就能够编程，但此时 RB5/PGM 引脚用于控制进入编程模式，不能再作为通用 I/O 引脚。

器件被擦除后，LVP 被使能。

当使用 LVP 编程时，与正常执行模式一样，将 VDD 电压加到 MCLR/VPP 引脚。要进入编程模式，将 VDD 电压加到 PGM 引脚上。

- | | |
|----------|--|
| 注 | 1: 不管 LVP 位和 PGM 引脚的状态如何，只要把 VIH 电压加在 MCLR 引脚上，就可以实现高压编程。 |
| | 2: 当低压编程使能时，RB5 引脚不能再用作通用 I/O 引脚。 |
| | 3: 当 LVP 使能时，从外部将 PGM 引脚上拉到 VSS，以允许正常程序执行。 |

如果不使用低压 ICSP 编程模式，可以清零 LVP 位，且 RB5/PGM 可用作数字 I/O 引脚 RB5。仅当使用标准高压编程 (VIH 电压加在 MCLR/VPP 引脚上) 时，才可以清零或置位 LVP 位。一旦 LVP 禁止，只有标准高压编程可用，且只能用于器件编程。

没有代码保护的存储器可用块擦除或逐行擦除的方法来擦除，然后可在任意规定的 VDD 电压下对其写入。如果要擦除有代码保护的存储器，需要使用块擦除。当使用低压编程时，若要执行块擦除，必须提供 4.5V 到 5.5V 的 VDD 电压。

23.0 指令集综述

PIC18 指令集与以前的 PICmicro 指令集相比，添加了很多增强功能，并保持了易于从其他 PICmicro 指令集移植的特点。

大部分指令只占一个程序存储字的空间（16 位），但有 3 条指令需要两个程序存储字单元。

每条单字指令都是一个 16 位字，由指明指令类型的操作码和指令操作的一个或多个操作数组成。

整个指令集具有高度正交性，分为四种基本类型：

- 针对字节的操作类指令
- 针对位的操作类指令
- 立即数操作类指令
- 控制操作类指令

表 23-2 中的 PIC18 指令集综述列出了针对字节的操作类指令、针对位的操作类指令、立即数操作类指令和控制操作类指令。表 23-1 中是对操作码字段的说明。

大多数针对字节的操作类指令都有三种操作数：

1. 数据寄存器（由“f”指定）
2. 保存结果的目标寄存器（由“d”指定）
3. 访问的存储器（由“a”指定）

数据寄存器指示符“f”指定指令会使用哪个数据寄存器。

目标寄存器指示符“d”指定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，操作结果则存入指令指定的数据寄存器中。

所有针对位的操作类指令都有三种操作数：

1. 数据寄存器（由“f”指定）
2. 数据寄存器中的位（由“b”指定）
3. 访问的存储器（由“a”指定）

位字段指示符“b”选择受操作影响的位，而数据寄存器指示符“f”则表示该位所在的数据寄存器地址。

立即数操作类指令可以使用以下操作数：

- 要装入数据寄存器的立即数值（由“k”指定）
- 希望装入立即数值的 FSR 寄存器（由“f”指定）
- 不需要操作数（用“—”表示）
-

控制操作类指令可以使用以下一些操作数：

- 程序存储器地址（由“n”指定）
- 调用或返回指令模式（由“s”指定）
- 读表和写表指令模式（由“m”指定）
- 不需要操作数（用“—”表示）

所有的指令都是单字指令，除了三条双字指令。这三条指令被作为双字指令，这样所有需要的信息就可以保存在 32 位中。在第二个字中，4 个 MSb 都是 1。如果第二个字作为指令执行（仅第二个字），将执行一个 NOP 指令。

除非条件测试结果为 true 或者指令执行改变了程序计数器值，否则所有单字指令的执行都只需要一个指令周期。对于条件测试结果为 true 或者指令执行改变了程序计数器值的这些情况，需要两个指令周期，其中新增加的指令周期是执行 NOP 指令。

执行双字指令需要两个指令周期。

每条指令周期包括 4 个振荡周期。因此，对于频率为 4 MHz 的振荡器，其正常指令执行时间为 1 μs。如果条件测试为 true 或指令执行改变了程序计数器值，则该指令的执行时间为 2 μs。双字转移指令（如果为 true）的执行需要 3 μs。

图 23-1 给出了指令的几种通用格式。

所有示例都使用“nnh”格式来表示十六进制数，其中“h”表示一个十六进制数字。

表 23-2 “指令集综述”列出了 Microchip 汇编器（MPASM™ 汇编器）能识别的指令。第 23.2 节“指令集”将一一介绍每条指令。

23.1 读—修改—写操作指令

指令中如果指定了数据寄存器，则这些指令都会执行读—修改—写（Read-Modify-Write, R-M-W）操作。先读寄存器，然后修改数据，再将结果存放在由指令或目标指示符“d”指定的位置。即使指令将写某个寄存器，之前也仍然会对此寄存器执行读操作。

例如，“BCF PORTB,1”指令先读取 PORTB，然后将数据的 bit 1 清零，最后把结果写回到 PORTB。读操作可能会产生意外的结果：任何将 RBIF 标志位置位的状态都被清零。R-M-W 操作还会把输入引脚的电平锁存到相应的输出锁存器。

PIC18F2331/2431/4331/4431

表 23-1: 操作码字段描述

字段	描述
a	RAM 存取位: a=0: RAM 位于快速访问存储区 (忽略 BSR 寄存器) a=1: RAM 存储区由 BSR 寄存器指定
bbb	8 位数据寄存器内的位地址 (0 到 7)。
BSR	存储区选择寄存器。用于选择当前的 RAM 存储区。
d	目标寄存器选择位: d=0: 结果保存至 WREG d=1: 结果保存至数据寄存器 f
dest	目标寄存器, 可以是 WREG 寄存器或指定的寄存器文件地址。
f	8 位寄存器文件地址 (0x00 到 0xFF)。
fs	12 位寄存器文件地址 (0x000 到 0xFFF)。这是源地址。
fd	12 位寄存器文件地址 (0x000 到 0xFFF)。这是目标地址。
k	立即数、常数或者标号 (可能是 8 位、12 位或 20 位值)。
label	标号名。
mm	读表和写表指令的 TBLPTR 寄存器模式。 只与读表和写表指令一起使用:
*	不改变寄存器 (比如读表和写表的 TBLPTR)。
*+	后增寄存器 (比如读表和写表的 TBLPTR)。
*-	后减寄存器 (比如读表和写表的 TBLPTR)。
++	先增寄存器 (比如读表和写表的 TBLPTR)。
n	相对于转移指令的相对地址 (二进制补码), 或调用 / 转移和返回指令的直接地址。
PRODH	乘积高位字节。
PRODL	乘积低位字节。
s	快速调用 / 返回模式选择位: s=0: 不更新影子寄存器, 也不用影子寄存器的内容更新其他寄存器 s=1: 将某些寄存器值存入影子寄存器, 或把影子寄存器的内容载入某些寄存器 (快速模式)
u	未使用或未改变。
WREG	工作寄存器 (累加器)。
x	与取值无关的位 (0 或 1)。 汇编器将产生 x=0 的代码。为了与所有的 Microchip 软件工具兼容, 建议使用这种格式。
TBLPTR	21 位表指针 (指向程序存储单元)。
TABLAT	8 位表锁存器。
TOS	栈顶。
PC	程序计数器。
PCL	程序计数器低位字节。
PCH	程序计数器高位字节。
PCLATH	程序计数器次高位字节锁存器。
PCLATU	程序计数器最高位字节锁存器。
GIE	全局中断允许位。
WDT	看门狗定时器。
TO	超时溢出位。
PD	掉电位。
C, DC, Z, OV, N	ALU 状态位: 进位标志位、辅助进位标志位、全零标志位、溢出标志位及负标志位。
[]	可选的。
()	内容。
→	赋值给。
< >	寄存器位域。
∈	表示属于某个集合。
italics	用户定义项 (字体为 courier)。

图 23-1: 指令的通用格式

针对字节的数据寄存器操作		示例指令
15	10 9 8 7 0	
操作码	d a f (数据寄存器地址)	ADDWF MYREG, W, B
d=0 表示结果存入 WREG 寄存器 d=1 表示结果存入数据寄存器 (f) a=0 访问快速存取 RAM a=1 由 BSR 选择存储区 f=8 位数据寄存器地址		
从字节到字节传送操作 (双字)		
15	12 11 0	
操作码	f (源数据寄存器地址)	MOVFF MYREG1, MYREG2
15	12 11 0	
1111	f (目标数据寄存器地址)	
f=12 位数据寄存器地址		
针对位的数据寄存器操作		
15	12 11 9 8 7 0	
操作码	b (BIT #) a f (数据寄存器地址)	BSF MYREG, bit, B
b= 占 3 位, 表示数据寄存器 (f) 中某位的位置 a= 0 访问快速存取 RAM a= 1 由 BSR 选择存储区 f= 8 位数据寄存器地址		
立即数操作		
15	8 7 0	
操作码	k (立即数)	MOVLW 0x7F
k=8 位立即数		
控制操作		
CALL、GOTO 和转移操作		
15	8 7 0	
操作码	n<7:0> (立即数)	GOTO Label
15	12 11 0	
1111	n<19:8> (立即数)	
n=20 位立即数		
15	8 7 0	
操作码	S n<7:0> (立即数)	CALL MYFUNC
15	12 11 0	
	n<19:8> (立即数)	
S= 快速位		
15	11 10 0	
操作码	n<10:0> (立即数)	BRA MYFUNC
15	8 7 0	
操作码	n<7:0> (立即数)	BC MYFUNC

PIC18F2331/2431/4331/4431

表 23-2: PIC18FXXX 指令集

助记符, 操作数	描述	周期	16 位指令字			影响的状态位	备注		
			MSb		LSb				
针对字节的数据寄存器操作									
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、f 和进位位相加	1	0010	00da	ffff	ffff	C, DC, Z, OV, N	1, 2
ANDWF	f, d, a	WREG 与 f 进行“与”运算	1	0001	01da	ffff	ffff	Z, N	1, 2
CLRF	f, a	f 清零	1	0110	101a	ffff	ffff	Z	2
COMF	f, d, a	f 取反	1	0001	11da	ffff	ffff	Z, N	1, 2
CPFSEQ	f, a	比较 f 和 WREG, = 则跳过	1 (2 或 3)	0110	001a	ffff	ffff	无	4
CPFSGT	f, a	比较 f 和 WREG, > 则跳过	1 (2 或 3)	0110	010a	ffff	ffff	无	4
CPFSLT	f, a	比较 f 和 WREG, < 则跳过	1 (2 或 3)	0110	000a	ffff	ffff	无	1, 2
DECF	f, d, a	f 减 1	1	0000	01da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
DECFSZ	f, d, a	f 减 1, 为 0 则跳过	1 (2 或 3)	0010	11da	ffff	ffff	无	1, 2, 3, 4
DCFSNZ	f, d, a	f 减 1, 非 0 则跳过	1 (2 或 3)	0100	11da	ffff	ffff	无	1, 2
INCF	f, d, a	f 增 1	1	0010	10da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4
INCFSZ	f, d, a	f 增 1, 为 0 则跳过	1 (2 或 3)	0011	11da	ffff	ffff	无	4
INFSNZ	f, d, a	f 增 1, 非 0 则跳过	1 (2 或 3)	0100	10da	ffff	ffff	无	1, 2
IORWF	f, d, a	WREG 与 f 进行“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	将第一个字 f _s (源) 传送到第二个字 f _d (目标)	2	1100	ffff	ffff	ffff	无	
MOVWF	f, a	WREG 传送到 f	1	0110	111a	ffff	ffff	无	
MULWF	f, a	WREG 与 f 相乘	1	0000	001a	ffff	ffff	无	
NEGF	f, a	f 取补	1	0110	110a	ffff	ffff	C, DC, Z, OV, N	1, 2
RLCF	f, d, a	带进位的循环左移 f	1	0011	01da	ffff	ffff	C, Z, N	
RLNCF	f, d, a	循环左移 f (无进位)	1	0100	01da	ffff	ffff	Z, N	1, 2
RRCF	f, d, a	带进位循环右移 f	1	0011	00da	ffff	ffff	C, Z, N	
RRNCF	f, d, a	循环右移 f (无进位)	1	0100	00da	ffff	ffff	Z, N	
SETF	f, a	f 置位	1	0110	100a	ffff	ffff	无	
SUBFWB	f, d, a	从 WREG 减去 f 和借位位	1	0101	01da	ffff	ffff	C, DC, Z, OV, N	1, 2
SUBWF	f, d, a	从 f 减去 WREG	1	0101	11da	ffff	ffff	C, DC, Z, OV, N	
SUBWFB	f, d, a	从 f 减去 WREG 和借位位	1	0101	10da	ffff	ffff	C, DC, Z, OV, N	1, 2
SWAPF	f, d, a	对 f 进行半字节交换	1	0011	10da	ffff	ffff	无	
TSTFSZ	f, a	测试 f, 为 0 则跳过	1 (2 或 3)	0110	011a	ffff	ffff	无	4
XORWF	f, d, a	WREG 与 f 进行“异或”运算	1	0001	10da	ffff	ffff	Z, N	1, 2
针对位的数据寄存器操作									
BCF	f, b, a	对 f 进行位清零	1	1001	bbba	ffff	ffff	无	1, 2
BSF	f, b, a	对 f 进行位置位	1	1000	bbba	ffff	ffff	无	1, 2
BTFSC	f, b, a	检测 f 的位, 为 0 则跳过	1 (2 或 3)	1011	bbba	ffff	ffff	无	3, 4
BTFSS	f, b, a	检测 f 的位, 为 1 则跳过	1 (2 或 3)	1010	bbba	ffff	ffff	无	3, 4
BTG	f, d, a	对 f 中的某位取反	1	0111	bbba	ffff	ffff	无	1, 2

注 1: 当修改端口寄存器使其随本身内容改变时 (例如, MOVF PORTB, 1, 0), 使用的值是引脚上的当前值。例如, 如果一引脚配置为输入, 其数据锁存器中的值为“1”, 但此时外部器件将该引脚拉为低电平, 则写回数据锁存器的数据值将是“0”。

2: 当对 TMR0 寄存器执行指令 (并且, 可能的话 d=1) 时, 如果对预分频器赋值, 则将清零该预分频器。

3: 如果程序计数器 (PC) 被修改或者执行条件测试为 true, 则执行指令需要两个指令周期。第二个周期执行 NOP 指令。

4: 某些指令是双字指令。如果指令的第一个字无法从该 16 位获得内嵌的信息, 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都具有合法指令。

5: 如果写表操作开始写内部存储器, 则写操作将持续到终止为止。

PIC18F2331/2431/4331/4431

表 23-2: PIC18FXXX 指令集 (续)

助记符, 操作数	描述	周期	16 位指令字				影响的状态位	备注	
			MSb			LSb			
控制操作									
BC	n	进位则转移	1 (2)	1110	0010	nnnn	nnnn	无	
BN	n	为负则转移	1 (2)	1110	0110	nnnn	nnnn	无	
BNC	n	无进位则转移	1 (2)	1110	0011	nnnn	nnnn	无	
BNN	n	不为负则转移	1 (2)	1110	0111	nnnn	nnnn	无	
BNOV	n	不溢出则转移	1 (2)	1110	0101	nnnn	nnnn	无	
BNZ	n	不为 0 则转移	2	1110	0001	nnnn	nnnn	无	
BOV	n	溢出则转移	1 (2)	1110	0100	nnnn	nnnn	无	
BRA	n	无条件转移	1 (2)	1101	0nnn	nnnn	nnnn	无	
BZ	n	为 0 则转移	1 (2)	1110	0000	nnnn	nnnn	无	
CALL	n, s	调用子程序的第一个字	2	1110	110s	kkkk	kkkk	无	
		第二个字		1111	kkkk	kkkk	kkkk		
CLRWDT	-	清零看门狗定时器	1	0000	0000	0000	0100	TO,PD	
DAW	-	十进制调整 WREG	1	0000	0000	0000	0111	C,DC	
GOTO	n	跳转到地址的第一个字	2	1110	1111	kkkk	kkkk	无	
		第二个字		1111	kkkk	kkkk	kkkk		
NOP	-	空操作	1	0000	0000	0000	0000	无	
NOP	-	空操作	1	1111	XXXX	XXXX	XXXX	无	4
POP	-	弹出返回堆栈栈顶 (TOS) 内容	1	0000	0000	0000	0110	无	
PUSH	-	将内容压入返回堆栈栈顶	1	0000	0000	0000	0101	无	
RCALL	n	相对调用	2	1101	1nnn	nnnn	nnnn	无	
RESET		用软件使器件复位	1	0000	0000	1111	1111	所有	
RETFIE	s	中断返回	2	0000	0000	0001	000s	GIE/GIEH, PEIE/GIEL	
RETLW	k	返回时将立即数写入 WREG	2	0000	1100	kkkk	kkkk	无	
RETURN	s	从子程序返回	2	0000	0000	0001	001s	无	
SLEEP	-	进入休眠模式	1	0000	0000	0000	0011	TO,PD	

- 注 1: 当修改端口寄存器使其随本身内容改变时 (例如, MOVF PORTB, 1, 0), 使用的值是引脚上的当前值。例如, 如果一引脚配置为输入, 其数据锁存器中的值为 “1”, 但此时外部器件将该引脚拉为低电平, 则写回数据锁存器的数据值将是 “0”。
- 2: 当对 TMRO 寄存器执行指令 (并且, 可能的话 d=1) 时, 如果对预分频器赋值, 则将清零该预分频器。
- 3: 如果程序计数器 (PC) 被修改或者执行条件测试为 true, 则执行指令需要两个指令周期。第二个周期执行 NOP 指令。
- 4: 某些指令是双字指令。如果指令的第一个字无法从该 16 位获得内嵌的信息, 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都具有合法指令。
- 5: 如果写表操作开始写内部存储器, 则写操作将持续到终止为止。

PIC18F2331/2431/4331/4431

表 23-2: PIC18FXXX 指令集 (续)

助记符, 操作数	描述	周期	16 位指令字				影响的状态位	备注
			MSb			LSb		
立即数操作								
ADDLW k	立即数与 WREG 相加	1	0000	1111	kkkk	kkkk	C, DC, Z, OV, N	
ANDLW k	立即数与 WREG 进行“与”运算	1	0000	1011	kkkk	kkkk	Z, N	
IORLW k	立即数与 WREG 进行“或”运算	1	0000	1001	kkkk	kkkk	Z, N	
LFSR f, k	将立即数 (12 位) 的第二个字传送到 FSRx 的第一个字	2	1110	1110	00ff	kkkk	无	
MOVLB k	将立即数传送到 BSR<3:0>	1	0000	0001	0000	kkkk	无	
MOVLW k	将立即数传送到 WREG	1	0000	1110	kkkk	kkkk	无	
MULLW k	立即数与 WREG 相乘	1	0000	1101	kkkk	kkkk	无	
RETLW k	返回时将立即数写入 WREG	2	0000	1100	kkkk	kkkk	无	
SUBLW k	从立即数减去 WREG	1	0000	1000	kkkk	kkkk	C, DC, Z, OV, N	
XORLW k	立即数与 WREG 进行“异或”运算	1	0000	1010	kkkk	kkkk	Z, N	
数据存储器 ↔ 程序存储器操作								
TBLRD*	读表	2	0000	0000	0000	1000	无	
TBLRD*+	后增读表		0000	0000	0000	1001	无	
TBLRD*-	后减读表		0000	0000	0000	1010	无	
TBLRD*+	先增读表		0000	0000	0000	1011	无	
TBLWT*	写表	2 (5)	0000	0000	0000	1100	无	
TBLWT*+	后增写表		0000	0000	0000	1101	无	
TBLWT*-	后减写表		0000	0000	0000	1110	无	
TBLWT*+	先增写表		0000	0000	0000	1111	无	

- 注 1: 当修改端口寄存器使其随本身内容改变时 (例如, MOVF PORTB, 1, 0), 使用的值是引脚上的当前值。例如, 如果一引脚配置为输入, 其数据锁存器中的值为“1”, 但此时外部器件将该引脚拉为低电平, 则写回数据锁存器的数据值将是“0”。
- 2: 当对 TMR0 寄存器执行指令 (并且, 可能的话 d=1) 时, 如果对预分频器赋值, 则将清零该预分频器。
- 3: 如果程序计数器 (PC) 被修改或者执行条件测试为 true, 则执行指令需要两个指令周期。第二个周期执行 NOP 指令。
- 4: 某些指令是双字指令。如果指令的第一个字无法从该 16 位获得内嵌的信息, 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都具有合法指令。
- 5: 如果写表操作开始写内部存储器, 则写操作将持续到终止为止。

23.2 指令集

ADDLW 立即数与 W 内容相加

语法: [label] 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、Z

机器码:

0000	1111	kkkk	kkkk
------	------	------	------

描述: W 的内容与 8 位立即数 k 相加, 结果存入 W。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写 W

示例: ADDLW 0x15

指令执行前

W = 0x10

指令执行后

W = 0x25

ADDWF W 内容与 f 内容相加

语法: [label] ADDWF f [,d [,a]]

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(W)+(f) \rightarrow \text{dest}$

受影响的状态位: N、OV、C、DC、Z

机器码:

0010	01da	ffff	ffff
------	------	------	------

描述: W 内容与寄存器 f 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 (缺省)。如果 a 为 0, 选择快速存取 RAM。如果 a 为 1, 使用 BSR。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写目标寄存器

示例: ADDWF REG, W

指令执行前

W = 0x17

REG = 0xC2

指令执行后

W = 0xD9

REG = 0xC2

PIC18F2331/2431/4331/4431

ADDWFC W 内容、f 内容和进位位相加

语法: [*label*] ADDWFC f [,d [,a]]

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: (W)+(f)+(C) → dest

受影响的状态位: N、OV、C、DC、Z

机器码:

0010	00da	ffff	ffff
------	------	------	------

描述: W 内容、进位标志位和数据存储单元 f 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，选择快速存取 RAM。如果 a 为 1，则使用 BSR。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写目标 寄存器

示例: ADDWFC REG, W

指令执行前
 进位标志位 = 1
 REG = 0x02
 W = 0x4D

指令执行后
 进位标志位 = 0
 REG = 0x02
 W = 0x50

ANDLW 立即数与 W 内容进行“与”运算

语法: [*label*] ANDLW k

操作数: $0 \leq k \leq 255$

操作: (W) .AND. k → W

受影响的状态位: N、Z

机器码:

0000	1011	kkkk	kkkk
------	------	------	------

描述: W 的内容与 8 位立即数 k 进行“与”运算。结果存储在 W 中。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写 W

示例: ANDLW 0x5F

指令执行前
 W = 0xA3

指令执行后
 W = 0x03

PIC18F2331/2431/4331/4431

ANDWF W 内容与 f 内容进行“与”运算

语法: [label] ANDWF f [,d [,a]]

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: (W) .AND. (f) → dest

受影响的状态位: N、Z

机器码:

0001	01da	ffff	ffff
------	------	------	------

描述: W 的内容与寄存器 f 的内容进行“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（缺省）。如果 a 为 0，选择快速存取 RAM。如果 a 为 1，则使用 BSR（缺省）。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写目标寄存器

示例: ANDWF REG, W

指令执行前
W = 0x17
REG = 0xC2
指令执行后
W = 0x02
REG = 0xC2

BC 进位则转移

语法: [label] BC n

操作数: $-128 \leq n \leq 127$

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
------	------	------	------

描述: 如果进位标志位为 1，程序将转移。二进制补码“2n”与 PC 相加。因为 PC 要先递增才能取下一条指令，因此新地址将为 PC+2+2n。然后该指令成为一条双周期指令。

指令字数: 1

指令周期: 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 PC
无操作	无操作	无操作	无操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	无操作

示例: HERE BC JUMP

指令执行前
PC = 地址 (HERE)
指令执行后
如果进位标志位 = 1;
PC = 地址 (JUMP)
如果进位标志位 = 0;
PC = 地址 (HERE+2)

PIC18F2331/2431/4331/4431

BCF 对 f 进行位清零

语法: [label] BCF f,b[a]
 操作数: $0 \leq f \leq 255$
 $0 \leq b \leq 7$
 $a \in [0,1]$
 操作: $0 \rightarrow f < b >$
 受影响的状态位: 无
 机器码:

1001	bbba	ffff	ffff
------	------	------	------

 描述: 对寄存器 f 中的 bit 'b' 清零。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。
 指令字数: 1
 指令周期: 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

指令执行前
 FLAG_REG=0xC7
 指令执行后
 FLAG_REG=0x47

BN 为负则转移

语法: [label] BN n
 操作数: $-128 \leq n \leq 127$
 操作: 如果负标志位为 “1”
 $(PC) + 2 + 2n \rightarrow PC$
 受影响的状态位: 无
 机器码:

1110	0110	nnnn	nnnn
------	------	------	------

 描述: 如果负标志位为 “1”, 程序将转移。
 二进制补码 “2n” 与 PC 相加。因为 PC 要先递增才能取下一条指令, 因此新地址将为 $PC + 2 + 2n$ 。然后该指令成为一条双周期指令。
 指令字数: 1
 指令周期: 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 PC
无操作	无操作	无操作	无操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	无操作

示例: HERE BN Jump

指令执行前
 PC = 地址 (HERE)
 指令执行后
 如果负标志位 = 1;
 PC = 地址 (Jump)
 如果负标志位 = 0;
 PC = 地址 (HERE+2)

PIC18F2331/2431/4331/4431

BNC 无进位则转移

语法: [label] BNC n

操作数: $-128 \leq n \leq 127$

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
------	------	------	------

描述: 如果进位标志位为“0”，程序将转移。
二进制补码“2n”与PC相加。因为PC要先递增才能取下一条指令，因此新地址将为PC+2+2n。然后该指令成为一条双周期指令。

指令字数: 1

指令周期: 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 PC
无操作	无操作	无操作	无操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	无操作

示例: HERE BNC Jump

指令执行前
PC = 地址 (HERE)

指令执行后
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 不为负则转移

语法: [label] BNN n

操作数: $-128 \leq n \leq 127$

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
------	------	------	------

描述: 如果负标志位为“0”，程序将转移。
二进制补码“2n”与PC相加。因为PC要先递增才能取下一条指令，因此新地址将为PC+2+2n。然后该指令成为一条双周期指令。

指令字数: 1

指令周期: 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 PC
无操作	无操作	无操作	无操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	无操作

示例: HERE BNN Jump

指令执行前
PC = 地址 (HERE)

指令执行后
如果负标志位 = 0
PC = 地址 (Jump)
如果负标志位 = 1
PC = 地址 (HERE+2)

PIC18F2331/2431/4331/4431

BNOV 不溢出则转移

语法: [label] BNOV n
 操作数: $-128 \leq n \leq 127$
 操作: 如果溢出标志位为“0”，
 (PC) + 2 + 2n → PC

受影响的状态位: 无

机器码:	1110	0101	nnnn	nnnn
------	------	------	------	------

描述: 如果溢出标志位为“0”，程序将转移。
 二进制补码“2n”与PC相加。因为PC要先递增才能取下一条指令，因此新地址将为PC+2+2n。然后该指令成为一条双周期指令。

指令字数: 1

指令周期: 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 PC
无操作	无操作	无操作	无操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	无操作

示例: HERE BNOV Jump

指令执行前
 PC = 地址 (HERE)

指令执行后
 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 0 则转移

语法: [label] BNZ n
 操作数: $-128 \leq n \leq 127$
 操作: 如果全零标志位为“0”，
 (PC) + 2 + 2n → PC

受影响的状态位: 无

机器码:	1110	0001	nnnn	nnnn
------	------	------	------	------

描述: 如果全零标志位为“0”，程序将转移。
 二进制补码“2n”与PC相加。因为PC要先递增才能取下一条指令，因此新地址将为PC+2+2n。然后该指令成为一条双周期指令。

指令字数: 1

指令周期: 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 PC
无操作	无操作	无操作	无操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	无操作

示例: HERE BNZ Jump

指令执行前
 PC = 地址 (HERE)

指令执行后
 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)

PIC18F2331/2431/4331/4431

BRA 无条件转移

语法: [label] BRA n
 操作数: $-1024 \leq n \leq 1023$
 操作: $(PC) + 2 + 2n \rightarrow PC$
 受影响的状态位: 无
 机器码:

1101	0nnn	nnnn	nnnn
------	------	------	------

 描述: 二进制补码“2n”与PC相加。因为PC要先递增才能取下一条指令，因此新地址将为PC+2+2n。然后该指令成为一条双周期指令。
 指令字数: 1
 指令周期: 2
 Q周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	写 PC
无操作	无操作	无操作	无操作

示例: HERE BRA Jump

指令执行前
 PC = 地址 (HERE)
 指令执行后
 PC = 地址 (Jump)

BSF 对 f 进行位置位

语法: [label] BSF f,b[a]
 操作数: $0 \leq f \leq 255$
 $0 \leq b \leq 7$
 $a \in [0,1]$
 操作: $1 \rightarrow f$
 受影响的状态位: 无
 机器码:

1000	bbba	ffff	ffff
------	------	------	------

 描述: 将寄存器 f 中的 bit 'b' 置 1。如果 a 为 0，选择快速存取 RAM，忽略 BSR 的值。如果 a 为 1，则会根据 BSR 的值选择存储区。
 指令字数: 1
 指令周期: 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

指令执行前
 FLAG_REG = 0x0A
 指令执行后
 FLAG_REG = 0x8A

PIC18F2331/2431/4331/4431

BTFSC 对数据寄存器进行位测试，为 0 则跳过

语法: [label] 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
------	------	------	------

 描述: 如果寄存器 f 的 bit 'b' 为 0, 则跳过下一条指令。
 如果 bit 'b' 为 0, 则丢弃下一条指令 (已在当前指令执行期间取指) 而执行一条 NOP 指令, 使该指令变成双周期指令。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1
 指令周期: 1(2)
 注: 如果跳过并且后面为一条双字指令, 则需要 3 个指令周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	无操作

如果跳过:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作

如果跳过并且后面为一条双字指令:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作
无操作	无操作	无操作	无操作

示例:

```
HERE    BTFSC    FLAG, 1
FALSE   :
TRUE    :
```

指令执行前
 PC = 地址 (HERE)
 指令执行后
 如果 FLAG<1> = 0
 PC = 地址 (TRUE)
 如果 FLAG<1> = 1
 PC = 地址 (FALSE)

BTFSS 对数据寄存器进行位测试，为 1 则跳过

语法: [label] 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
------	------	------	------

 描述: 如果寄存器 f 的 bit 'b' 为 1, 则跳过下一条指令。
 如果 bit 'b' 为 1, 则丢弃下一条指令 (已在当前指令执行期间取指), 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, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1
 指令周期: 1(2)
 注: 如果跳过并且后面为一条双字指令, 则需要 3 个指令周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	无操作

如果跳过:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作

如果跳过并且后面为一条双字指令:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作
无操作	无操作	无操作	无操作

示例:

```
HERE    BTFSS    FLAG, 1
FALSE   :
TRUE    :
```

指令执行前
 PC = 地址 (HERE)
 指令执行后
 如果 FLAG<1> = 0
 PC = 地址 (FALSE)
 如果 FLAG<1> = 1
 PC = 地址 (TRUE)

PIC18F2331/2431/4331/4431

BTG

对 f 中的某位取反

语法: [label] BTG f,b[,a]

操作数: $0 \leq f \leq 255$
 $0 \leq b < 7$
 $a \in [0,1]$

操作: $(f < b) \rightarrow f < \overline{b}$

受影响的状态位: 无

机器码:

0111	bbba	ffff	ffff
------	------	------	------

描述: 对数据存储单元 f 的 bit 'b' 取反。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器 f

示例: BTG PORTC, 4

指令执行前:
PORTC = 0111 0101 [0x75]

指令执行后:
PORTC = 0110 0101 [0x65]

BOV

溢出则转移

语法: [label] 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
------	------	------	------

描述: 如果溢出标志位为 “1”, 程序将转移。二进制补码 “2n” 与 PC 相加。因为 PC 要先递增才能取下一条指令, 因此新地址将为 $PC+2+2n$ 。然后该指令成为一条双周期指令。

指令字数: 1

指令周期: 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 PC
无操作	无操作	无操作	无操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	无操作

示例: HERE BOV JUMP

指令执行前
PC = 地址 (HERE)

指令执行后
 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)

PIC18F2331/2431/4331/4431

BZ 为 0 则转移

语法: [label] BZ n
 操作数: $-128 \leq n \leq 127$
 操作: 如果全零标志位为“1”
 (PC) + 2 + 2n → PC
 受影响的状态位: 无
 机器码:

1110	0000	nnnn	nnnn
------	------	------	------

 描述: 如果全零标志位为“1”，程序将转移。
 二进制补码“2n”与 PC 相加。因为 PC 要先递增才能取下一条指令，因此新地址将为 PC+2+2n。然后该指令成为一条双周期指令。
 指令字数: 1
 指令周期: 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 PC
无操作	无操作	无操作	无操作

如果不跳转:

Q1	Q2	Q3	Q4
译码	读立即数 n	处理数据	无操作

示例: HERE BZ Jump

指令执行前
 PC = 地址 (HERE)
 指令执行后
 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: [label] CALL k [s]
 操作数: $0 \leq k \leq 1048575$
 $s \in [0,1]$
 操作: (PC) + 4 → TOS,
 k → PC<20:1>,
 如果 s=1
 (W) → WS,
 (STATUS) → STATUSS,
 (BSR) → BSRS
 受影响的状态位: 无
 机器码:
 第一个字 (k<7:0>)

1110	110s	k ₇ kkk	kkkk ₀
------	------	--------------------	-------------------

 第二个字 (k<19:8>)

1111	k ₁₉ kkk	kkkk	kkkk ₈
------	---------------------	------	-------------------

 描述: 可在 2MB 存储空间内进行子程序调用。首先，将返回地址 (PC+4) 压入返回堆栈。如果 s=1，则 W、STATUS 和 BSR 寄存器也会被压入对应的影子寄存器 WS、STATUSS 和 BSRS。如果 s=0，则不会进行更新 (缺省)。然后，将 20 位的 k 值装入 PC<20:1>。CALL 是一条双周期指令。

指令字数: 2
 指令周期: 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
无操作	无操作	无操作	无操作

示例: HERE CALL THERE,FAST

指令执行前
 PC = 地址 (HERE)
 指令执行后
 PC = 地址 (THERE)
 TOS = 地址 (HERE+4)
 WS = W
 BSRS = BSR
 STATUSS = STATUS

PIC18F2331/2431/4331/4431

CLRF **f** 清零

语法: [label] CLRF f [,a]
 操作数: $0 \leq f \leq 255$
 $a \in [0,1]$
 操作: $000h \rightarrow f$
 $1 \rightarrow Z$
 受影响的状态位: Z
 机器码:

0110	101a	ffff	ffff
------	------	------	------

 描述: 清零指定寄存器的内容。如果 **a** 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 **a** 为 1, 则会根据 BSR 的值选择存储区 (缺省)。
 指令字数: 1
 指令周期: 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

指令执行前
 FLAG_REG = 0x5A
 指令执行后
 FLAG_REG = 0x00

CLRWDT 清零看门狗定时器

语法: [label] CLRWDT
 操作数: 无
 操作: $000h \rightarrow WDT$,
 $000h \rightarrow WDT$ 后分频器,
 $1 \rightarrow \overline{TO}$,
 $1 \rightarrow \overline{PD}$
 受影响的状态位: \overline{TO} 、 \overline{PD}
 机器码:

0000	0000	0000	0100
------	------	------	------

 描述: CLRWDT 指令复位看门狗定时器。而且还复位 WDT 的后分频器。状态位 \overline{TO} 和 \overline{PD} 被置位。
 指令字数: 1
 指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	无操作	处理数据	无操作

示例: CLRWDT

指令执行前
 WDT 计数器 = ?
 指令执行后
 WDT 计数器 = 0x00
 WDT 后分频器 = 0
 \overline{TO} = 1
 \overline{PD} = 1

PIC18F2331/2431/4331/4431

COMF f 取反

语法: [label] COMF f [,d [,a]]

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(\bar{f}) \rightarrow \text{dest}$

受影响的状态位: N、Z

机器码:

0001	11da	ffff	ffff
------	------	------	------

描述: 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 (缺省)。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写目标寄存器

示例: COMF REG, W

指令执行前
REG = 0x13

指令执行后
REG = 0x13
W = 0xEC

CPFSEQ 比较 f 和 W, 如果 f = W 则跳过

语法: [label] CPFSEQ f [,a]

操作数: $0 \leq f \leq 255$
 $a \in [0,1]$

操作: $(f) - (W)$,
 如果 $(f) = (W)$ 则跳过
 (无符号比较)

受影响的状态位: 无

机器码:

0110	001a	ffff	ffff
------	------	------	------

描述: 通过执行无符号减法比较数据存储器单元 f 和 W 的内容。如果 f 的内容等于 W 的内容, 则所取的指令会被丢弃, 转而执行一个 NOP, 从而使该指令变成双周期指令。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1

指令周期: 1(2)

注: 如果跳过并且后面为一条双字指令, 则需要 3 个指令周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	无操作

如果跳过:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作

如果跳过并且后面为一条双字指令:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作
无操作	无操作	无操作	无操作

示例: HERE CPFSEQ REG
NEQUAL :
EQUAL :

指令执行前
PC 地址 = HERE
W = ?
REG = ?

指令执行后
如果 REG = W;
PC = 地址 (EQUAL)
如果 REG \neq W;
PC = 地址 (NEQUAL)

PIC18F2331/2431/4331/4431

CPFSGT 比较 f 和 W，如果 f > W 则跳过

语法: [label] CPFSGT f [,a]

操作数: 0 ≤ f ≤ 255
 a ∈ [0,1]

操作: (f) - (W),
 如果 (f) > (W) 则跳过
 (无符号比较)

受影响的状态位: 无

机器码:

0110	010a	ffff	ffff
------	------	------	------

描述: 通过执行无符号减法比较数据存储单元 f 和 W 的内容。
 如果 f 的内容大于 W 的内容，则所取的指令会被丢弃，转而执行一个 NOP，从而使该指令变成双周期指令。如果 a 为 0，选择快速存取 RAM，忽略 BSR 的值。如果 a 为 1，则会根据 BSR 的值选择存储区（缺省）。

指令字数: 1

指令周期: 1(2)
注: 如果跳过并且后面为一条双字指令，则需要 3 个指令周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	无操作

如果跳过:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作

如果跳过并且后面为一条双字指令:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作
无操作	无操作	无操作	无操作

示例: HERE CPFSGT REG
 NGREATER :
 GREATER :

指令执行前
PC = 地址 (HERE)
W = ?

指令执行后
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: [label] 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
------	------	------	------

描述: 通过执行无符号减法比较数据存储单元 f 和 W 的内容。
 如果 f 的内容小于 W 的内容，则所取的指令会被丢弃，转而执行一个 NOP，从而使该指令变成双周期指令。如果 a 为 0，选择快速存取 RAM。如果 a 为 1，则使用 BSR（缺省）。

指令字数: 1

指令周期: 1(2)
注: 如果跳过并且后面为一条双字指令，则需要 3 个指令周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	无操作

如果跳过:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作

如果跳过并且后面为一条双字指令:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作
无操作	无操作	无操作	无操作

示例: HERE CPFSLT REG
 NLESS :
 LESS :

指令执行前
PC = 地址 (HERE)
W = ?

指令执行后
如果 REG < W;
PC = 地址 (LESS)
如果 REG ≥ W;
PC = 地址 (NLESS)

PIC18F2331/2431/4331/4431

DAW 十进制调整 W 寄存器

语法: [*label*] DAW

操作数: 无

操作: 如果 [W<3:0> >9] 或 [DC = 1], 则
(W<3:0>) + 6 → W<3:0>;
否则
(W<3:0>) → W<3:0>;

 如果 [W<7:4> >9] 或 [C = 1], 则
(W<7:4>) + 6 → W<7:4>;
否则
(W<7:4>) → W<7:4>;

受影响的状态位: C、DC

机器码:

0000	0000	0000	0111
------	------	------	------

描述: DAW 调整 W 内的 8 位值, 这 8 位值为前面两个变量 (格式均为紧凑型 BCD 格式) 的和, 并产生正确的紧凑型 BCD 结果。进位标志位可用 DAW 指令设置, 与执行 DAW 指令前的该位设置无关。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 W	处理数据	写 W

示例 1: DAW

指令执行前

W = 0xA5

C = 0

DC = 0

指令执行后

W = 0x05

C = 1

DC = 0

示例 2:

指令执行前

W = 0xCE

C = 0

DC = 0

指令执行后

W = 0x34

C = 1

DC = 0

DECF f 减 1

语法: [*label*] 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、Z

机器码:

0000	01da	ffff	ffff
------	------	------	------

描述: 寄存器 f 内容减 1。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存回寄存器 f (缺省)。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写目标寄存器

示例: DECF CNT,

指令执行前

CNT = 0x01

Z = 0

指令执行后

CNT = 0x00

Z = 1

PIC18F2331/2431/4331/4431

GOTO 无条件转移

语法: [label] 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:

第一个字 (k<7:0>)	1110	1111	k ₇ kkk	kkkk ₀
第二个字 (k<19:8>)	1111	k ₁₉ kkk	kkkk	kkkk ₈

描述: GOTO 允许无条件转移到整个 2MB 存储空间中的任何地方。将 20 位的 k 值装入 PC<20:1>。GOTO 永远是双周期指令。

指令字数: 2

指令周期: 2

Q 周期操作:

	Q1	Q2	Q3	Q4
译码	读立即数 k<7:0>	无操作	无操作	读立即数 k<19:8>, 写入 PC
	无操作	无操作	无操作	无操作

示例: GOTO THERE

指令执行后

PC = 地址 (THERE)

INCF f 增 1

语法: [label] INCF f [,d [,a]]

操作数: $0 \leq f \leq 255$

$d \in [0,1]$

$a \in [0,1]$

操作: $(f) + 1 \rightarrow \text{dest}$

受影响的状态位: C、DC、N、OV、Z

机器码:

0010	10da	ffff	ffff
------	------	------	------

描述: 寄存器 f 的内容增 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 (缺省)。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1

指令周期: 1

Q 周期操作:

	Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写目标寄存器	

示例: INCF CNT,

指令执行前

CNT = 0xFF
Z = 0
C = ?
DC = ?

指令执行后

CNT = 0x00
Z = 1
C = 1
DC = 1

PIC18F2331/2431/4331/4431

INCFSZ **f 增 1, 为 0 则跳过**

语法: [label] INCFSZ f [,d [,a]]

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(f) + 1 \rightarrow \text{dest}$,
 如果结果为 0 则跳过

受影响的状态位: 无

机器码:

0011	1lda	ffff	ffff
------	------	------	------

描述: 寄存器 f 的内容增 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 (缺省)。如果结果为 0, 则已经取指的下一条指令被丢弃, 转而执行一条 NOP, 使该指令变成双周期指令。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1

指令周期: 1(2)
注: 如果跳过并且后面为一条双字指令, 则需要 3 个指令周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写目标 寄存器

如果跳过:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作

如果跳过, 而且后面为一条双字指令:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作
无操作	无操作	无操作	无操作

示例: HERE INCFSZ CNT
 NZERO :
 ZERO :

指令执行前

PC = 地址 (HERE)

指令执行后

CNT = CNT + 1
 如果 CNT = 0;
 PC = 地址 (ZERO)
 如果 CNT \neq 0;
 PC = 地址 (NZERO)

INFSNZ **f 增 1, 非 0 则跳过**

语法: [label] INFSNZ f [,d [,a]]

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(f) + 1 \rightarrow \text{dest}$,
 如果结果不为 0 则跳过

受影响的状态位: 无

机器码:

0100	10da	ffff	ffff
------	------	------	------

描述: 寄存器 f 的内容增 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 (缺省)。如果结果不为 0, 则已经取指的下一条指令被丢弃, 转而执行一条 NOP, 使该指令变成双周期指令。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1

指令周期: 1(2)
注: 如果跳过并且后面为一条双字指令, 则需要 3 个指令周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写目标 寄存器

如果跳过:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作

如果跳过, 而且后面为一条双字指令:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作
无操作	无操作	无操作	无操作

示例: HERE INFSNZ REG
 ZERO :
 NZERO :

指令执行前

PC = 地址 (HERE)

指令执行后

REG = REG + 1
 如果 REG \neq 0;
 PC = 地址 (NZERO)
 如果 REG = 0;
 PC = 地址 (ZERO)

PIC18F2331/2431/4331/4431

IORLW 立即数与 W 内容进行“或”运算

语法: [label] 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、Z
 机器码:

0000	1001	kkkk	kkkk
------	------	------	------

 描述: W 的内容与 8 位立即数 k 进行“或”运算。结果存储在 W 中。
 指令字数: 1
 指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写 W

示例: IORLW 0x35

指令执行前

W = 0x9A

指令执行后

W = 0xBF

IORWF W 内容与 f 内容进行“或”运算

语法: [label] 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、Z
 机器码:

0001	00da	ffff	ffff
------	------	------	------

 描述: W 的内容与寄存器 f 的内容进行“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（缺省）。如果 a 为 0，选择快速存取 RAM，忽略 BSR 的值。如果 a 为 1，则会根据 BSR 的值选择存储区（缺省）。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写目标寄存器

示例: IORWF RESULT, W

指令执行前

RESULT = 0x13

W = 0x91

指令执行后

RESULT = 0x13

W = 0x93

PIC18F2331/2431/4331/4431

LFSR **载入 FSR**

语法: [*label*] LFSR f,k

操作数: $0 \leq f \leq 2$
 $0 \leq k \leq 4095$

操作: $k \rightarrow \text{FSRf}$

受影响的状态位: 无

机器码:

1110	1110	00ff	$k_{11}kkk$
1111	0000	k_7kkk	kkkk

描述: 12 位立即数 *k* 装入 *f* 指向的文件选择寄存器。

指令字数: 2

指令周期: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 <i>k</i> 的 MSB	处理数据	将立即数 <i>k</i> 的 MSB 写入 FSRfH
译码	读立即数 <i>k</i> 的 LSB	处理数据	将立即数 <i>k</i> 的 LSB 入 SRfL

示例: LFSR 2, 0x3AB

指令执行后
 FSR2H = 0x03
 FSR2L = 0xAB

MOVF **传送 f**

语法: [*label*] MOVF f[d,a]

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $f \rightarrow \text{dest}$

受影响的状态位: N、Z

机器码:

0101	00da	ffff	ffff
------	------	------	------

描述: 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* (缺省)。单元 *f* 可以位于 256 字节存储区中的任何地方。如果 *a* 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 *a* 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 <i>f</i>	处理数据	写 <i>W</i>

示例: MOVF REG, W

指令执行前
 REG = 0x22
 W = 0xFF

指令执行后
 REG = 0x22
 W = 0x22

PIC18F2331/2431/4331/4431

MOVFF 将 f_s 的内容送入 f_d

语法: [label] 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:

第一个字 (源)	1100	ffff	ffff	ffff f_s
第二个字 (目标)	1111	ffff	ffff	ffff f_d

描述: 源寄存器 f_s 的内容被送入目标寄存器 f_d 。源 f_s 可以是 4096 字节数据空间 (000h 到 FFFh) 中的任何存储单元, 目标 f_d 也可以是 000h 到 FFFh 中的任何存储单元。源或目标都可以是 W (这是个有用的特殊情况)。

MOVFF 可将数据存储单元传递到外设寄存器 (例如发送缓冲器或 I/O 端口), 这一点特别有用。

MOVFF 指令中的目标寄存器不能是 PCL、TOSU、TOSH 或 TOSL。

当允许任何中断时, 不应当使用 MOVFF 指令来修改中断设置 (参见第 91 页上的注)。

指令字数: 2
 指令周期: 2 (3)

Q 周期操作:

	Q1	Q2	Q3	Q4
译码	译码	读寄存器 “f” (源)	处理数据	无操作
译码	译码	无操作 无哑读取	无操作	写寄存器 “f” (目标)

示例: MOVFF REG1, REG2

指令执行前
 REG1 = 0x33
 REG2 = 0x11

指令执行后
 REG1 = 0x33
 REG2 = 0x33

MOVLB 将立即数传送到 BSR 的低 4 位

语法: [label] MOVLB k

操作数: $0 \leq k \leq 255$

操作: $k \rightarrow$ BSR

受影响的状态位: 无

机器码:

0000	0001	kkkk	kkkk
------	------	------	------

描述: 8 位立即数 k 装入存储区选择寄存器 (BSR)。

指令字数: 1
 指令周期: 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

指令执行前
 BSR 寄存器 = 0x02

指令执行后
 BSR 寄存器 = 0x05

PIC18F2331/2431/4331/4431

MOVLW 将立即数传送到 W

语法: [label] MOVLW k
操作数: $0 \leq k \leq 255$
操作: $k \rightarrow W$
受影响的状态位: 无
机器码:

0000	1110	kkkk	kkkk
------	------	------	------

描述: 8 位立即数 k 装入 W。
指令字数: 1
指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写 W

示例: MOVLW 0x5A

指令执行后

W = 0x5A

MOVWF 将 W 的内容传送到 f

语法: [label] MOVWF f[,a]
操作数: $0 \leq f \leq 255$
 $a \in [0,1]$
操作: $(W) \rightarrow f$
受影响的状态位: 无
机器码:

0110	111a	ffff	ffff
------	------	------	------

描述: 将数据从 W 传送到寄存器 f。单元 f 可以位于 256 字节存储区中的任何地方。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1

指令周期: 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

指令执行前

W = 0x4F
REG = 0xFF

指令执行后

W = 0x4F
REG = 0x4F

PIC18F2331/2431/4331/4431

MULLW 立即数与 W 相乘

语法: [label] 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
------	------	------	------

描述: W 的内容与 8 位立即数 k 执行无符号乘法运算。16 位结果存储在 PRODH:PRODL 寄存器对中。PRODH 包含高字节。W 不改变。不影响任何状态标志位。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。

指令字数: 1
 指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写寄存器对 PRODH:PRODL

示例: MULLW 0xC4

指令执行前
 W = 0xE2
 PRODH = ?
 PRODL = ?
 指令执行后
 W = 0xE2
 PRODH = 0xAD
 PRODL = 0x08

MULWF W 与 f 相乘

语法: [label] 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
------	------	------	------

描述: W 的内容与数据寄存器单元 f 执行无符号乘法运算。16 位结果存储在 PRODH:PRODL 寄存器对中。PRODH 包含高字节。W 和 f 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,选择快速存取 RAM,忽略 BSR 的值。如果 a 为 1,则会根据 BSR 的值选择存储区(缺省)。

指令字数: 1
 指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写寄存器对 PRODH:PRODL

示例: MULWF REG

指令执行前
 W = 0xC4
 REG = 0xB5
 PRODH = ?
 PRODL = ?
 指令执行后
 W = 0xC4
 REG = 0xB5
 PRODH = 0x8A
 PRODL = 0x94

PIC18F2331/2431/4331/4431

NEGF f 取补

语法: [label] 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、Z
 机器码:

0110	110a	ffff	ffff
------	------	------	------

 描述: 对单元 **f** 取二进制补码。结果存储在数据存储器单元 **f** 中。如果 **a** 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 **a** 为 1, 则会根据 BSR 的值选择存储区。
 指令字数: 1
 指令周期: 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

指令执行前
 REG = 0011 1010 [0x3A]
 指令执行后
 REG = 1100 0110 [0xC6]

NOP 空操作

语法: [label] NOP
 操作数: 无
 操作: 无操作
 受影响的状态位: 无
 机器码:

0000	0000	0000	0000
1111	xxxx	xxxx	xxxx

 描述: 无操作。
 指令字数: 1
 指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	无操作	无操作	无操作

示例:

无。

PIC18F2331/2431/4331/4431

POP 弹出返回堆栈栈顶内容

语法: [label] POP
 操作数: 无
 操作: (TOS) → 位存储桶
 受影响的状态位: 无
 机器码:

0000	0000	0000	0110
------	------	------	------

 描述: 从返回堆栈取出 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以组成软件堆栈。
 指令字数: 1
 指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	无操作	弹出 TOS 值	无操作

示例:

	POP		NEW
	GOTO		
指令执行前			
TOS	=	0x0031A2	
堆栈 (下一级)	=	0x014332	
指令执行后			
TOS	=	0x014332	
PC	=	NEW	

PUSH 将内容压入返回堆栈栈顶

语法: [label] PUSH
 操作数: 无
 操作: (PC+2) → TOS
 受影响的状态位: 无
 机器码:

0000	0000	0000	0101
------	------	------	------

 描述: PC+2 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: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	将 PC+2 压入返回堆栈	无操作	无操作

示例:

		PUSH	
指令执行前			
TOS	=	0x00345A	
PC	=	0x000124	
指令执行后			
PC	=	0x000126	
TOS	=	0x000126	
堆栈 (下一级)	=	0x00345A	

PIC18F2331/2431/4331/4431

RCALL 相对调用

语法: [label] RCALL n
 操作数: $-1024 \leq n \leq 1023$
 操作: $(PC) + 2 \rightarrow TOS$,
 $(PC) + 2 + 2n \rightarrow PC$

受影响的状态位: 无

机器码:

1101	1nnn	nnnn	nnnn
------	------	------	------

描述: 从当前单元跳转 (最多 1KB) 来调用子程序。首先, 将返回地址 (PC+2) 压入堆栈。然后, 将二进制补码 “2n” 与 PC 相加。因为 PC 要先递增才能取下一条指令, 因此新地址将为 PC+2+2n。此指令是一条双周期指令。

指令字数: 1

指令周期: 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 PC
无操作	无操作	无操作	无操作

示例: HERE RCALL Jump

指令执行前

PC = 地址 (HERE)

指令执行后

PC = 地址 (Jump)

TOS = 地址 (HERE+2)

RESET 复位

语法: [label] RESET
 操作数: 无
 操作: 将所有受 MCLR 复位影响的寄存器和标志位复位。

受影响的状态位: 所有

机器码:

0000	0000	1111	1111
------	------	------	------

描述: 软件可使用此指令执行 MCLR 复位。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	开始复位	无操作	无操作

示例: RESET

指令执行后

寄存器 = 复位值

标志位 * = 复位值

PIC18F2331/2431/4331/4431

RETFIE 中断返回

语法: [label] 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 都不改变。

受影响的状态位: GIE/GIEH 和 PEIE/GIEL。

0000	0000	0001	000s
------	------	------	------

机器码:

描述: 从中断返回。执行出栈操作, 将栈顶 (Top-of-Stack, TOS) 单元内容装入 PC。通过将高 / 低优先级全局中断使能位置位, 可以允许中断。如果 s=1, 则将影子寄存器 WS、STATUS 和 BSRS 的内容装入对应的寄存器 W、STATUS 和 BSR。如果 s=0, 则不更新这些寄存器 (缺省)。

指令字数: 1

指令周期: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	无操作	无操作	从堆栈弹出 PC 值, 将 GIEH 或 GIEL 置位
无操作	无操作	无操作	无操作

示例: RETFIE 1

中断后

PC	=	TOS
W	=	WS
BSR	=	BSRS
STATUS	=	STATUS
GIE/GIEH、PEIE/GIEL	=	1

RETLW 返回时将立即数写入 W

语法: [label] RETLW k

操作数: 0 ≤ k ≤ 255

操作: k → W,
(TOS) → PC,
PCLATU 和 PCLATH 都不改变

受影响的状态位: 无

0000	1100	kkkk	kkkk
------	------	------	------

机器码:

描述: 将 8 位立即数 k 装入 W。将栈顶单元内容 (返回地址) 装入程序计数器。高位地址锁存器 (PCLATH) 保持不变。

指令字数: 1

指令周期: 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 W
无操作	无操作	无操作	无操作

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
```

指令执行前

W	=	0x07
---	---	------

指令执行后

W	=	kn 值
---	---	------

PIC18F2331/2431/4331/4431

RETURN 从子程序返回

语法: [label] 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 都不改变

受影响的状态位: 无

机器码:

0000	0000	0001	001s
------	------	------	------

描述: 从子程序返回。执行出栈操作, 将栈顶 (TOS) 单元内容装入程序计数器。如果 s=1, 则将影子寄存器 WS、STATUS 和 BSRS 的内容装入对应的寄存器: W、STATUS 和 BSR。如果 s=0, 则不更新这些寄存器 (缺省)。

指令字数: 1

指令周期: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	无操作	处理数据	从堆栈弹出 PC 值
无操作	无操作	无操作	无操作

示例: RETURN

中断后
PC = TOS

RLCF 带进位循环左移 f

语法: [label] RLCF f[,d[,a]]

操作数: 0 ≤ f ≤ 255

d ∈ [0,1]

a ∈ [0,1]

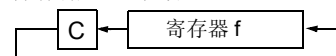
操作: (f<n>) → dest<n+1>,
(f<7>) → C,
(C) → dest<0>

受影响的状态位: C、N、Z

机器码:

0011	01da	ffff	ffff
------	------	------	------

描述: 寄存器 f 的内容带进位标志位循环左移 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 (缺省)。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。



指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写目标寄存器

示例: RLCF REG, W

指令执行前
REG = 1110 0110
C = 0

指令执行后
REG = 1110 0110
W = 1100 1100
C = 1

PIC18F2331/2431/4331/4431

RLNCF 循环左移 f (无进位)

语法: [label] RLNCF f [,d [,a]]

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

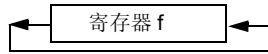
操作: $(f<n>) \rightarrow dest<n+1>$,
 $(f<7>) \rightarrow dest<0>$

受影响的状态位: N、Z

机器码:

0100	01da	ffff	ffff
------	------	------	------

描述: 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 (缺省)。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。



指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写目标 寄存器

示例: RLNCF REG

指令执行前
REG = 1010 1011
指令执行后
REG = 0101 0111

RRCF 带进位循环右移 f

语法: [label] RRCF f [,d [,a]]

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

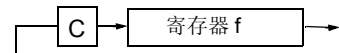
操作: $(f<n>) \rightarrow dest<n-1>$,
 $(f<0>) \rightarrow C$,
 $(C) \rightarrow dest<7>$

受影响的状态位: C、N、Z

机器码:

0011	00da	ffff	ffff
------	------	------	------

描述: 寄存器 f 的内容带进位标志位循环右移 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 (缺省)。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。



指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写目标 寄存器

示例: RRCF REG, W

指令执行前
REG = 1110 0110
C = 0
指令执行后
REG = 1110 0110
W = 0111 0011
C = 0

PIC18F2331/2431/4331/4431

RRNCF 循环右移 f (无进位)

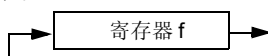
语法: [label] RRNCF f [,d [,a]]
 操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$
 操作: (f<n>) → dest<n-1>,
 (f<0>) → dest<7>

受影响的状态位: N、Z

机器码:

0100	00da	ffff	ffff
------	------	------	------

描述: 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 (缺省)。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。



指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写目标寄存器

示例 1: RRNCF REG, 1, 0

指令执行前
REG = 1101 0111

指令执行后
REG = 1110 1011

示例 2: RRNCF REG, W

指令执行前
W = ?
REG = 1101 0111

指令执行后
W = 1110 1011
REG = 1101 0111

SETF f 置位

语法: [label] SETF f [,a]
 操作数: $0 \leq f \leq 255$
 $a \in [0,1]$

操作: FFh → f

受影响的状态位: 无

机器码:

0110	100a	ffff	ffff
------	------	------	------

描述: 将指定寄存器的内容置为 FFh。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1

指令周期: 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

指令执行前
REG = 0x5A

指令执行后
REG = 0xFF

PIC18F2331/2431/4331/4431

SLEEP 进入休眠模式

语法: [label] 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} 、 \overline{PD}

机器码:

0000	0000	0000	0011
------	------	------	------

描述: 掉电状态位 (\overline{PD}) 清零。超时状态位 (\overline{TO}) 置位。看门狗定时器及其后分频器清零。振荡器停振, 单片机进入休眠模式。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	无操作	处理数据	进入休眠模式

示例: SLEEP

指令执行前
 \overline{TO} = ?
 \overline{PD} = ?

指令执行后
 \overline{TO} = 1
 \overline{PD} = 0

† 如果 WDT 唤醒单片机, 则此位将被清零。

SUBFWB 带借位从 W 减去 f

语法: [label] SUBFWB f[,d[,a]]

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(W) - (f) - (\overline{C}) \rightarrow \text{dest}$

受影响的状态位: N、OV、C、DC、Z

机器码:

0101	01da	ffff	ffff
------	------	------	------

描述: 从 W 减去 f 寄存器的内容和进位标志位 (借位) (通过二进制补码方法进行运算)。如果 d 为 0, 结果存储在 W 中。如果 d 为 1, 结果存入寄存器 f (缺省)。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写目标寄存器

示例 1: SUBFWB REG

指令执行前
REG = 0x03
W = 0x02
C = 0x01

指令执行后
REG = 0xFF
W = 0x02
C = 0x00
Z = 0x00
N = 0x01 ; result is negative

示例 2: SUBFWB REG, 0, 0

指令执行前
REG = 2
W = 5
C = 1

指令执行后
REG = 2
W = 3
C = 1
Z = 0
N = 0 ; result is positive

示例 3: SUBFWB REG, 1, 0

指令执行前
REG = 1
W = 2
C = 0

指令执行后
REG = 0
W = 2
C = 1
Z = 1 ; result is zero
N = 0

PIC18F2331/2431/4331/4431

SUBLW 从立即数减去 W

语法: [label] 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、Z
 机器码:

0000	1000	kkkk	kkkk
------	------	------	------

 描述: 从 8 位立即数 k 减去 W。结果存储在 W 中。
 指令字数: 1
 指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写 W

示例 1: SUBLW 0x02

指令执行前

W = 1
C = ?

指令执行后

W = 1
C = 1 ; result is positive
Z = 0
N = 0

示例 2: SUBLW 0x02

指令执行前

W = 2
C = ?

指令执行后

W = 0
C = 1 ; result is zero
Z = 1
N = 0

示例 3: SUBLW 0x02

指令执行前

W = 3
C = ?

指令执行后

W = FF ;(2's complement)
C = 0 ;result is negative
Z = 0
N = 1

SUBWF 从 f 减去 W

语法: [label] 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、Z
 机器码:

0101	11da	ffff	ffff
------	------	------	------

 描述: 从寄存器 f 减去 W (通过二进制补码方法进行运算)。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 (缺省)。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。
 指令字数: 1
 指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写目标寄存器

示例 1: SUBWF REG

指令执行前

REG = 3
W = 2
C = ?

指令执行后

REG = 1
W = 2
C = 1 ; result is positive
Z = 0
N = 0

示例 2: SUBWF REG, W

指令执行前

REG = 2
W = 2
C = ?

指令执行后

REG = 2
W = 0
C = 1 ;result is zero
Z = 1
N = 0

示例 3: SUBWF REG

指令执行前

REG = 0x01
W = 0x02
C = ?

指令执行后

REG = 0xFFh ;(2's complement)
W = 0x02
C = 0x00 ; result is negative
Z = 0x00
N = 0x01

PIC18F2331/2431/4331/4431

SUBWFB 从 f 减去 W 和借位

语法: [label] SUBWFB f[,d[,a]]

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: $(f) - (W) - (\bar{C}) \rightarrow \text{dest}$

受影响的状态位: N、OV、C、DC、Z

机器码:

0101	10da	ffff	ffff
------	------	------	------

描述: 从寄存器 f 减去 W 的内容和进位标志位 (借位) (通过二进制补码方法进行运算)。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 (缺省)。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写目标 寄存器

示例 1: SUBWFB REG, 1, 0

指令执行前

REG = 0x19 (0001 1001)
W = 0x0D (0000 1101)
C = 0x01

指令执行后

REG = 0x0C (0000 1011)
W = 0x0D (0000 1101)
C = 0x01
Z = 0x00
N = 0x00 ; result is positive

示例 2: SUBWFB REG, 0, 0

指令执行前

REG = 0x1B (0001 1011)
W = 0x1A (0001 1010)
C = 0x00

指令执行后

REG = 0x1B (0001 1011)
W = 0x00
C = 0x01
Z = 0x01 ; result is zero
N = 0x00

示例 3: SUBWFB REG, 1, 0

指令执行前

REG = 0x03 (0000 0011)
W = 0x0E (0000 1101)
C = 0x01

指令执行后

REG = 0xF5 (1111 0100)
; [2's comp]
W = 0x0E (0000 1101)
C = 0x00
Z = 0x00
N = 0x01 ; result is negative

PIC18F2331/2431/4331/4431

SWAPF 对 **f** 的内容进行半字节交换

语法: [*label*] SWAPF f[,d [,a]]

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: ($f<3:0>$) → $dest<7:4>$,
 ($f<7:4>$) → $dest<3:0>$

受影响的状态位: 无

机器码:

0011	10da	ffff	ffff
------	------	------	------

描述: 交换寄存器 **f** 的高 4 位和低 4 位。
 如果 **d** 为 0, 结果存储在 **W** 中。如
 果 **d** 为 1, 结果存入寄存器 **f** (缺
 省)。如果 **a** 为 0, 选择快速存取
 RAM, 忽略 **BSR** 的值。如果 **a** 为
 1, 则会根据 **BSR** 的值选择存储区
 (缺省)。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读 寄存器 f	处理数据	写目标 寄存器

示例: SWAPF REG

指令执行前
REG = 0x53

指令执行后
REG = 0x35

PIC18F2331/2431/4331/4431

TBLRD 读表

语法: [label] TBLRD (*, *+, *-, +*)

操作数: 无

操作: 对于 TBLRD *,
(程序存储器 (TBLPTR)) → TABLAT;
TBLPTR 不变;
对于 TBLRD *+,
(程序存储器 (TBLPTR)) → TABLAT;
(TBLPTR) +1 → TBLPTR;
对于 TBLRD *-,
(程序存储器 (TBLPTR)) → TABLAT;
(TBLPTR) -1 → TBLPTR;
对于 TBLRD +*,
(TBLPTR) +1 → TBLPTR;
(程序存储器 (TBLPTR)) → TABLAT;

受影响的状态位: 无

机器码:

0000	0000	0000	10nn nn=0 * =1 *+ =2 *- =3 +*
------	------	------	---

描述: 该指令用于读取程序存储器 (PM) 的内容。使用表指针 (TBLPTR) 对程序存储器进行寻址。
TBLPTR (21 位指针) 可以指向程序存储器中的所有字节。TBLPTR 的寻址空间是 2MB。

- TBLPTR[0]=0: 程序存储器字的最低有效字节
- TBLPTR[0]=1: 程序存储器字的最高有效字节

TBLRD 指令可以如下修改 TBLPTR 的值:

- 不变
- 后加
- 后减
- 先加

指令字数: 1

指令周期: 2

Q 周期操作:

	Q1	Q2	Q3	Q4
译码	无操作	无操作	无操作	无操作
无操作	无操作	无操作 (读程序存储器)	无操作	无操作 (写 TABLAT)

TBLRD 读表 (续)

示例 1: TBLRD *+ ;

指令执行前

TABLAT	=	0x55
TBLPTR	=	0x00A356
存储器 (0x00A356)	=	0x34

指令执行后

TABLAT	=	0x34
TBLPTR	=	0x00A357

示例 2: TBLRD +* ;

指令执行前

TABLAT	=	0xAA
TBLPTR	=	0x01A357
存储器 (0x01A357)	=	0x12
存储器 (0x01A358)	=	0x34

指令执行后

TABLAT	=	0x34
TBLPTR	=	0x01A358

PIC18F2331/2431/4331/4431

TBLWT 写表

语法: [label] TBLWT (*, *+, *-, **)

操作数: 无

操作: 对于 TBLWT*,
(TABLAT) → 保持寄存器;
TBLPTR 不变;
对于 TBLWT*+,
(TABLAT) → 保持寄存器;
(TBLPTR) +1 → TBLPTR ;
对于 TBLWT*-,
(TABLAT) → 保持寄存器;
(TBLPTR) -1 → TBLPTR ;
对于 TBLWT**+,
(TBLPTR) +1 → TBLPTR ;
(TABLAT) → 保持寄存器;

受影响的状态位: 无

0000	0000	0000	11nn nn=0 * =1 ** =2 *- =3 **
------	------	------	---

描述: 该指令使用 TBLPTR 的 3 个 LSb 来决定要将 TABLAT 的数据写入 8 个保持寄存器中的哪一个。该保持寄存器用于对程序存储器 (PM) 的内容编程。(关于闪存存储器编程的更多详情, 请参阅第 6.0 节“闪存程序存储器”。)
TBLPTR (21 位指针) 可以指向程序存储器中的所有字节。TBLPTR 的寻址空间是 2MB。TBLPTR 的 LSb 决定要访问程序存储单元的哪个字节。

TBLPTR[0]=0: 程序存储器字的最低有效字节

TBLPTR[0]=1: 程序存储器字的最高有效字节

TBLWT 指令可以如下修改 TBLPTR 的值:

- 不变
- 后加
- 后减
- 先加

TBLWT 写表 (续)

指令字数: 1

指令周期: 2

Q 周期操作:

Q1	Q2	Q3	Q4
译码	无操作	无操作	无操作
无操作	无操作 (读 TABLAT)	无操作	无操作 (写保持 寄存器)

示例 1: TBLWT **+;

指令执行前
TABLAT = 0x55
TBLPTR = 0x00A356
保持寄存器
(0x00A356) = 0xFF

指令执行后 (写表完成)
TABLAT = 0x55
TBLPTR = 0x00A357
保持寄存器
(0x00A356) = 0x55

示例 2: TBLWT **+;

指令执行前
TABLAT = 0x34
TBLPTR = 0x01389A
保持寄存器
(0x01389A) = 0xFF
保持寄存器
(0x01389B) = 0xFF

指令执行后 (写表完成)
TABLAT = 0x34
TBLPTR = 0x01389B
保持寄存器
(0x01389A) = 0xFF
保持寄存器
(0x01389B) = 0x34

PIC18F2331/2431/4331/4431

TSTFSZ 测试 f, 为 0 则跳过

语法: [label] TSTFSZ f[,a]
 操作数: $0 \leq f \leq 255$
 $a \in [0,1]$
 操作: 如果 $f=0$ 则跳过
 受影响的状态位: 无
 机器码:

0110	011a	ffff	ffff
------	------	------	------

 描述: 如果 $f=0$, 则丢弃下一条指令 (已在当前指令执行期间取指), 且执行一条 NOP 指令, 使该指令变成双周期指令。如果 a 为 0, 选择快速存取 RAM, 忽略 BSR 的值。如果 a 为 1, 则会根据 BSR 的值选择存储区 (缺省)。

指令字数: 1
 指令周期: 1(2)
注: 如果跳过并且后面为一条双字指令, 则需要 3 个指令周期。

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	无操作

如果跳过:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作

如果跳过并且后面为一条双字指令:

Q1	Q2	Q3	Q4
无操作	无操作	无操作	无操作
无操作	无操作	无操作	无操作

示例: HERE TSTFSZ CNT
 NZERO :
 ZERO :

指令执行前
 PC= 地址 (HERE)
 指令执行后
 如果 CNT = 0x00,
 PC = 地址 (ZERO)
 如果 CNT \neq 0x00,
 PC = 地址 (NZERO)

XORLW 立即数与 W 进行“异或”运算

语法: [label] XORLW k
 操作数: $0 \leq k \leq 255$
 操作: (W) .XOR. k \rightarrow W
 受影响的状态位: N、Z
 机器码:

0000	1010	kkkk	kkkk
------	------	------	------

 描述: W 的内容与 8 位立即数 k 进行“异或”运算。结果存储在 W 中。
 指令字数: 1
 指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读立即数 k	处理数据	写 W

示例: XORLW 0xAF
 指令执行前
 W = 0xB5
 指令执行后
 W = 0x1A

PIC18F2331/2431/4331/4431

XORWF **W** 的内容与 **f** 的内容进行“异或”运算

语法: `[label] XORWF f[,d[,a]]`

操作数: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

操作: `(W).XOR. (f) → dest`

受影响的状态位: **N**、**Z**

机器码:

0001	10da	ffff	ffff
------	------	------	------

描述: **W** 的内容与寄存器 **f** 进行“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**（缺省）。如果 **a** 为 0，选择快速存取 **RAM**，忽略 **BSR** 的值。如果 **a** 为 1，则会根据 **BSR** 的值选择存储区（缺省）。

指令字数: 1

指令周期: 1

Q 周期操作:

Q1	Q2	Q3	Q4
译码	读寄存器 f	处理数据	写目标寄存器

示例: XORWF REG

指令执行前

REG = 0xAF

W = 0xB5

指令执行后

REG = 0x1A

W = 0xB5

PIC18F2331/2431/4331/4431

注:

24.0 开发支持

一系列硬件及软件开发工具对 PICmicro® 单片机提供支持：

- 集成开发环境
 - MPLAB® IDE 软件
- 汇编器 / 编译器 / 链接器
 - MPASM™ 汇编器
 - MPLAB C18 和 MPLAB C30 C 编译器
 - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
 - MPLAB ASM30 汇编器 / 链接器 / 库
- 模拟器
 - MPLAB SIM 软件模拟器
- 仿真器
 - MPLAB ICE 2000 在线仿真器
 - MPLAB ICE 4000 在线仿真器
- 在线调试器
 - MPLAB ICD 2
- 器件编程器
 - PICSTART® Plus 开发编程器
 - MPLAB PM3 器件编程器
- 低成本演示和开发板及评估工具包

24.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16 位单片机市场提供了前所未有的易于使用的软件开发平台。MPLAB IDE 是基于 Windows® 操作系统的应用软件，包括：

- 一个包含所有调试工具的图形界面
 - 模拟器
 - 编程器（单独销售）
 - 仿真器（单独销售）
 - 在线调试器（单独销售）
- 具有彩色上下文代码显示的全功能编辑器
- 多项目管理器
- 内容可直接编辑的可定制式数据窗口
- 高级源代码调试
- 可视化器件初始化程序，便于进行寄存器的初始化
- 鼠标停留在变量上进行查看的功能
- 通过拖放把变量从源代码窗口拉到观察窗口
- 丰富的在线帮助
- 集成了可选的第三方工具，如 HI-TECH 软件 C 编译器和 IAR C 编译器

MPLAB IDE 可以让您：

- 编辑源文件（汇编语言或 C 语言）
- 点击一次即可完成汇编（或编译）并将代码下载到 PICmicro MCU 仿真器和模拟器工具中（自动更新所有项目信息）
- 可使用如下各项进行调试：
 - 源文件（汇编语言或 C 语言）
 - 混合汇编语言和 C 语言
 - 机器码

MPLAB IDE 在单个开发范例中支持使用多种调试工具，包括从成本效益高的模拟器到低成本的在线调试器，再到全功能的仿真器。这样缩短了用户升级到更加灵活而功能更强大的工具时的学习时间。

PIC18F2331/2431/4331/4431

24.2 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于所有的 PICmicro MCU。

MPASM 汇编器可生成用于 MPLINK 目标链接器的可重定位目标文件、Intel® 标准 HEX 文件、详细描述存储器使用状况和符号参考的 MAP 文件、包含源代码行及生成机器码的绝对 LST 文件以及用于调试的 COFF 文件。

MPASM 汇编器具有如下特征：

- 集成在 MPLAB IDE 项目中
- 用户定义的宏可简化汇编代码
- 对多用途源文件进行条件汇编
- 允许完全控制汇编过程的指令

24.3 MPLAB C18 和 MPLAB C30 C 编译器

MPLAB C18 和 MPLAB C30 代码开发系统是完整的 ANSI C 编译器，分别适用于 Microchip 的 PIC18 系列单片机和 dsPIC30F 系列数据信号控制器。这些编译器可提供其他编译器并不具备的强大的集成功能和出众的代码优化能力，且使用方便。

为便于源代码调试，编译器提供了针对 MPLAB IDE 调试器的优化符号信息。

24.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器包含了由 MPASM 汇编器、MPLAB C18 C 编译器产生的可重定位目标。通过使用链接器脚本中的指令，它还可链接预编译库中的可重定位目标。

MPLIB 目标库管理器管理预编译代码库文件的创建和修改。当从源文件调用库中的一段子程序时，只有包含此子程序的模块被链接到应用中。这样可使大型库在许多不同应用中被高效地利用。

目标链接器 / 库管理器具有如下特征：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起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

24.5 MPLAB ASM30 汇编器、链接器和库管理器

MPLAB ASM30 汇编器为 dsPIC30F 器件提供转换自符号汇编语言的可重定位机器码。MPLAB C30 C 编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特征：

- 支持整个 dsPIC30F 指令集
- 支持定点数据和浮点数据
- 命令行界面
- 丰富的指令集
- 灵活的宏语言
- MPLAB IDE 兼容性

24.6 MPLAB SIM 软件模拟器

MPLAB SIM 软件模拟器在指令级对 PICmicro MCU 和 dsPIC® DSC 进行模拟，使得用户可以在 PC 主机的环境下进行代码开发。对于任何给定的指令，用户均可对数据区进行检查或修改，并通过各种触发机制来产生激励。可以将各寄存器的情况记录在文件中，以便进行进一步地运行时分析。跟踪缓冲器和逻辑分析器的显示使模拟器还能记录和跟踪程序的执行、I/O 的动作以及内部寄存器的状况。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C18 和 MPLAB C30 C 编译器以及 MPASM 和 MPLAB ASM30 汇编器的符号调试。该软件模拟器可用于在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。

24.7 MPLAB ICE 2000 高性能在线仿真器

MPLAB ICE 2000 在线仿真器旨在为产品开发工程师提供一整套用于 PICmicro 单片机的设计工具。MPLAB ICE 2000 在线仿真器的软件控制由 MPLAB 集成开发环境平台提供，它允许在单一环境下进行编辑、编译、下载以及源代码调试。

MPLAB ICE 2000 是全功能仿真器系统，它具有增强的跟踪、触发和数据监控功能。处理器模块可插拔，使系统可轻松进行重新配置以适应各种不同处理器的仿真需要。MPLAB ICE 2000 在线仿真器的架构允许对其进行扩展以支持新的 PICmicro 单片机。

MPLAB ICE 2000 在线仿真器系统设计为一款实时仿真系统，该仿真系统具备通常只有昂贵的开发工具中才有的高级功能。选择 PC 平台和 Microsoft® Windows® 32 位操作系统可使这些功能在一个简单而统一的应用中得到很好的利用。

24.8 MPLAB ICE 4000 高性能在线仿真器

MPLAB ICE 4000 在线仿真器旨在为产品开发工程师提供一整套用于高端 PICmicro MCU 和 dsPIC DSC 的设计工具。MPLAB ICE 4000 在线仿真器的软件控制由 MPLAB 集成开发环境平台提供，它允许在单一环境下进行编辑、编译、下载以及源代码调试。

MPLAB ICE 4000 是高级的仿真系统，除具备 MPLAB ICE 2000 的所有功能外，它还增加了适用于 dsPIC30F 和 PIC18XXXX 器件的仿真存储容量以及高速性能。该仿真器的先进特性包括复杂触发和定时功能及高达 2 Mb 的仿真存储容量。

MPLAB ICE 4000 在线仿真器系统设计为一款实时仿真系统，该仿真系统具备通常只有在更加昂贵的开发工具中才有的高级功能。选择 PC 平台和 Microsoft Windows 32 位操作系统可使这些功能在一个简单而统一的应用程序中得以很好的利用。

24.9 MPLAB ICD 2 在线调试器

Microchip 的在线调试器 MPLAB ICD 2 是一款功能强大而成本低廉的运行时开发工具，通过 RS-232 或高速 USB 接口与 PC 主机相连。该工具基于闪存 PICmicro MCU，可用于开发本系列及其他 PICmicro MCU 和 dsPIC DSC。MPLAB ICD 2 使用了闪存器件中内建的在线调试功能。该功能结合 Microchip 的在线串行编程 (In-Circuit Serial Programming™, ICSP™) 协议，可在 MPLAB 集成开发环境的图形用户界面上提供成本效益很高的在线闪存调试。这使设计人员可通过设置断点、单步运行以及对变量、CPU 状态以及外设寄存器进行监视的方法实现源代码的开发和调试。其全速运行特性可对硬件和应用进行实时测试。MPLAB ICD 2 还可用作某些 PICmicro 器件的开发编程器。

24.10 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款通用的、符合 CE 规范的器件编程器，其可编程电压设置在 VDDMIN 和 VDDMAX 之间时可靠性最高。它有一个用来显示菜单和错误信息的大 LCD 显示器 (128 x 64)，以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP™ 电缆。在单机模式下，MPLAB PM3 器件编程器不必与 PC 相连即可对 PICmicro 器件进行读取、验证和编程。在该模式下它还可设置代码保护。MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。MPLAB PM3 具备高速通信能力以及优化算法，可对存储器很大的器件进行快速编程，它还采用 SD/MMC 卡用作文件存储及数据安全应用。

PIC18F2331/2431/4331/4431

24.11 PICSTART Plus 开发编程器

PICSTART Plus 开发编程器是一款易于使用而成本低廉的原型编程器。它通过 COM (RS-232) 端口与 PC 相连。MPLAB 集成开发环境软件使得该编程器的使用简便、高效。PICSTART Plus 开发编程器支持采用 DIP 封装的大部分 PICmicro 器件，其引脚数最多可达 40 个。引脚数更多的器件，如 PIC16C92X 和 PIC17C76X，可通过连接一个转接插槽来获得支持。PICSTART Plus 开发编程器符合 CE 规范。

24.12 演示、开发和评估板

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

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

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

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

有关演示、开发和评估工具包的完整列表，请查阅 Microchip 公司网页 (www.microchip.com) 以及最新的 “*Product Selector Guide (产品选型指南)*” (DS00148)。

注:

PIC18F2331/2431/4331/4431

注:

25.0 电气特性

绝对最大额定值 (†)

环境温度.....	-55°C 至 +125°C
储存温度.....	-65°C 至 +150°C
任一引脚（除了 VDD、 $\overline{\text{MCLR}}$ ，和 RA4 之外）相对于 Vss 的电压.....	-0.3V 至 (VDD+0.3V)
VDD 相对于 Vss 的电压.....	-0.3V 至 +7.5V
$\overline{\text{MCLR}}$ 引脚上相对于 Vss 的电压（注 2）.....	0V 至 +13.25V
RA4 引脚上相对于 Vss 的电压.....	0V 至 +8.5V
总功耗（注 1）.....	1.0W
Vss 引脚的最大电流.....	300 mA
VDD 引脚的最大电流.....	250 mA
输入箝位电流 I _{IK} （V _I < 0 或 V _I > VDD）.....	±20 mA
输出箝位电流 I _{OK} （V _O < 0 或 V _O > VDD）.....	±20 mA
任一 I/O 引脚的最大输出灌电流.....	25 mA
任一 I/O 引脚的最大输出拉电流.....	25 mA
所有引脚的最大灌电流.....	200 mA
所有引脚的最大拉电流.....	200 mA

注 1: 功耗按如下公式计算:

$$P_{dis} = VDD \times \{I_{DD} - \sum I_{OH}\} + \sum \{(VDD - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$$

2: 如果 $\overline{\text{MCLR}}/V_{PP}$ 引脚上的尖峰电压低于 Vss，感应电流大于 80 mA，可能引起锁死。因此当对 $\overline{\text{MCLR}}/V_{PP}$ 引脚施加低电平时，应该串联一个 50-100Ω 的电阻，而不是直接将该引脚拉到 Vss。

† 注意: 如果工作条件超过上述“绝对最大额定值”可能会对器件造成永久性损坏。上述值仅为工作条件极大值，我们不建议器件在该极大值条件下或超出该规范范围的条件运行。长时间运行在最大额定值条件下会影响器件的可靠性。

PIC18F2331/2431/4331/4431

图 25-1: PIC18F2331/2431/4331/4431 电压-频率关系图 (工业级)

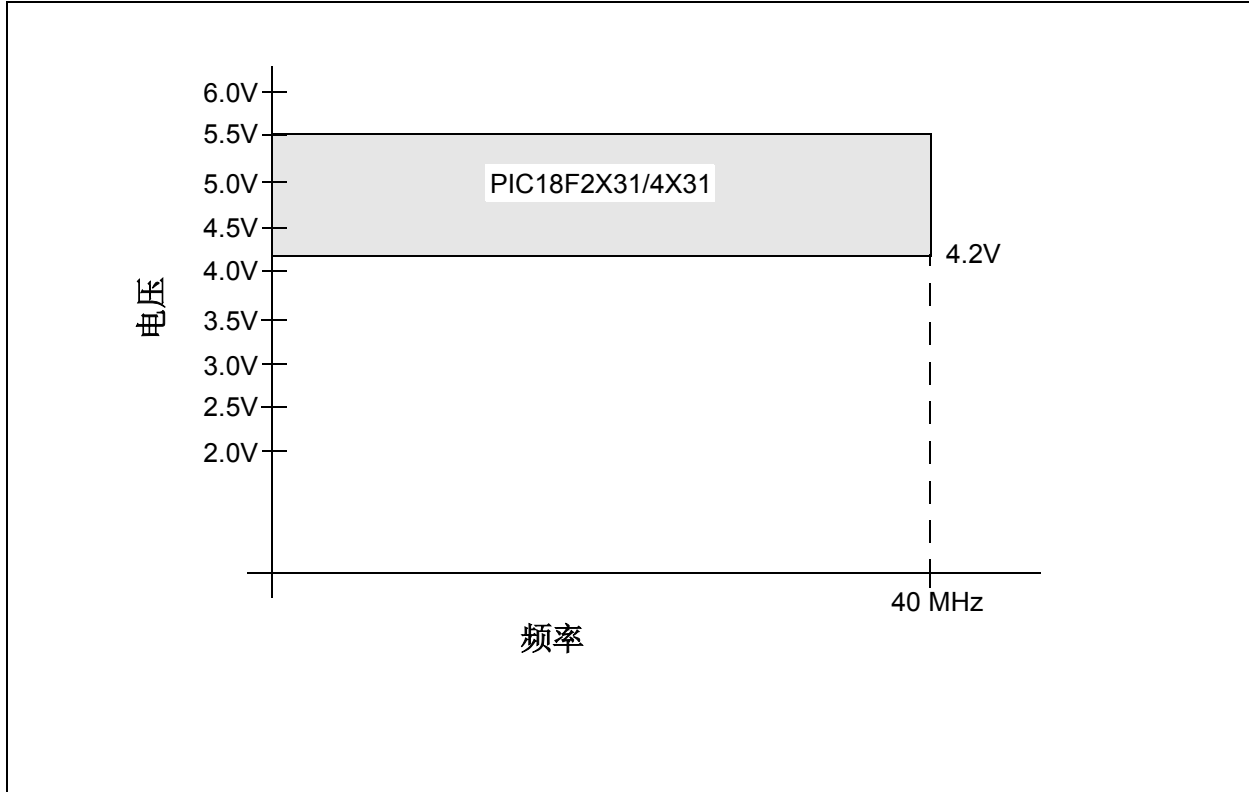
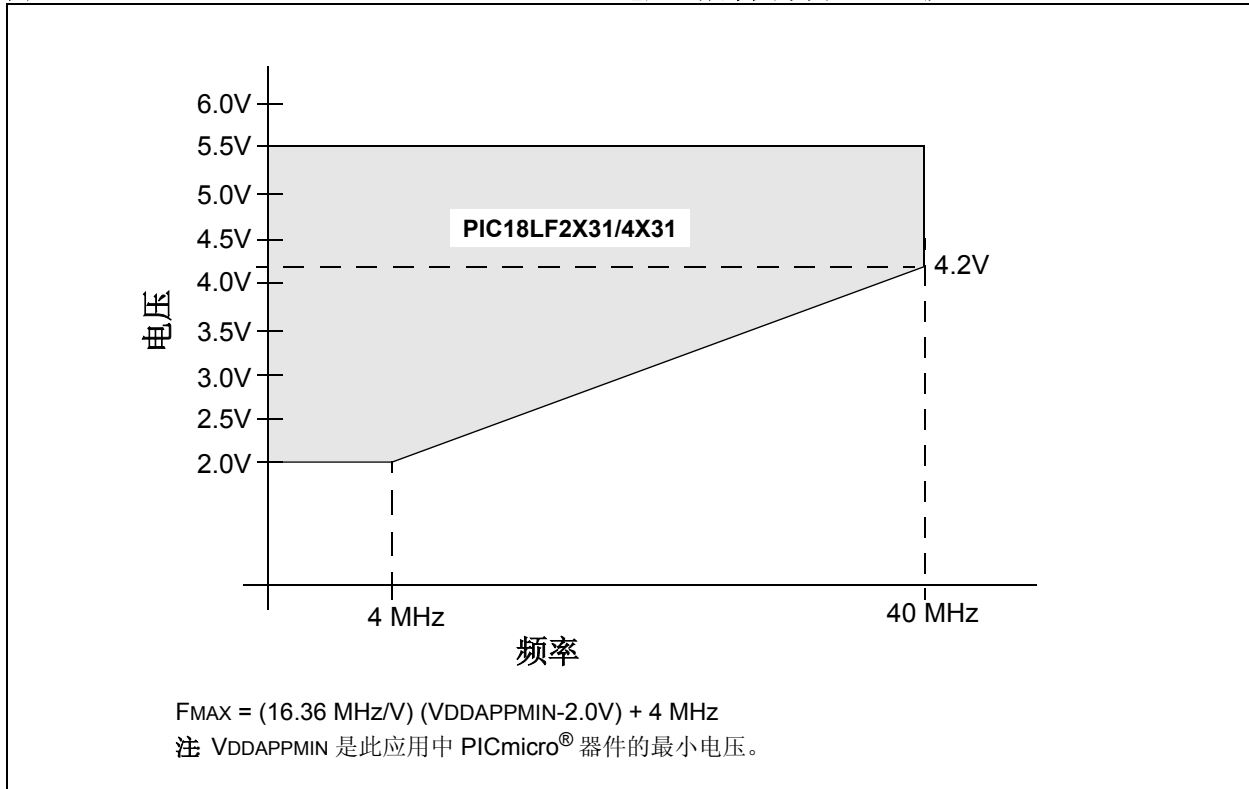


图 25-2: PIC18LF2331/2431/4331/4431 电压-频率关系图 (工业级)



PIC18F2331/2431/4331/4431

25.1 DC 特性:

电源电压

PIC18F2331/2431/4331/4431 (工业级, 扩展级)

PIC18LF2331/2431/4331/4431 (工业级)

PIC18F2331/2431/4331/4431 (工业级, 扩展级)		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
PIC18LF2331/2431/4331/4431 (工业级)		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
参数号	符号	特性	最小值	典型值	最大值	单位	条件
D001	VDD	电源电压					
		PIC18LF2X31/4X31	2.0	—	5.5	V	HS、XT、RC 和 LP 振荡模式
		PIC18F2X31/4X31	4.2	—	5.5	V	
D002	VDR	RAM 数据保持电压 ⁽¹⁾	1.5	—	—	V	
D003	VPOR	VDD 起始电压 确保能够产生内部上电复位信号	—	—	0.7	V	详情请参阅有关上电复位的章节
D004	SVDD	VDD 上升率 确保能够产生内部上电复位信号	0.05	—	—	V/ms	详情请参阅有关上电复位的章节
D005	VBOR	欠压复位电压					
		BORV1: BORV0 = 10	2.45	—	2.99	V	
		BORV1: BORV0 = 01	3.80	—	4.64	V	
		BORV1: BORV0 = 00	4.09	—	4.99	V	

图注: 阴影行是为了增强表的可读性。

注 1: 这是在休眠模式下或器件复位时保证不会丢失 RAM 数据的 VDD 下限值。

PIC18F2331/2431/4331/4431

25.2 DC 特性: 掉电电流和供电电流

PIC18F2331/2431/4331/4431 (工业级, 扩展级)
PIC18LF2331/2431/4331/4431 (工业级)

PIC18F2331/2431/4331/4431 (工业级, 扩展级)			标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)		
PIC18LF2331/2431/4331/4431 (工业级)			标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)		
参数号	器件	典型值	最大值	单位	条件
掉电电流 (I_{PD}) (1)					
	PIC18LF2X31/4X31	0.1	0.5	μA	-40°C
		0.1	0.5	μA	25°C
		0.2	1.9	μA	85°C
	PIC18LF2X31/4X31	0.1	0.5	μA	-40°C
		0.1	0.5	μA	25°C
		0.3	1.9	μA	85°C
	所有器件	0.1	2.0	μA	-40°C
		0.1	2.0	μA	25°C
		0.4	6.5	μA	85°C

图注: 阴影行是为了增强表的可读性。

- 注 1:** 休眠模式下的掉电电流与振荡器类型无关。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并连接到 V_{DD} 或者 V_{SS}, 且禁止了所有会增加电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 2:** 供电电流主要随工作电压、频率和模式而变化。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
 在有源工作模式下, 所有 I_{DD} 测量的测试条件为:
 OSC1= 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 V_{DD} ;
 MCLR=V_{DD} ; WDT 按规定使能 / 禁止。
- 3:** 器件配置为 RC 振荡器模式时, 该电流不包括流经 R_{EXT} 的电流。流经此电阻的电流可以通过公式 $I_r = V_{DD} / 2R_{EXT}$ (mA) 进行估算, 其中 R_{EXT} 的单位是 kΩ。
- 4:** 标准低成本 32 kHz 晶体振荡器的工作温度范围是从 -10°C 到 +70°C。扩展级温度的晶体振荡器成本就更高一些。

PIC18F2331/2431/4331/4431

25.2 DC 特性:

掉电电流和供电电流

PIC18F2331/2431/4331/4431 (工业级, 扩展级)

PIC18LF2331/2431/4331/4431 (工业级) (续)

PIC18F2331/2431/4331/4431 (工业级, 扩展级)		标准工作条件 (除非另有声明)						
		工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)						
PIC18LF2331/2431/4331/4431 (工业级)		标准工作条件 (除非另有声明)						
		工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)						
参数号	器件	典型值	最大值	单位	条件			
供电电流 (IDD) (2,3)								
PIC18LF2X31/4X31		8	40	μA	-40°C	VDD = 2.0V	FOSC = 31 kHz (RC_RUN 模式, 内部振荡源)	
		9	40	μA	25°C			
		11	40	μA	85°C			
PIC18LF2X31/4X31		25	68	μA	-40°C	VDD = 3.0V		
		25	68	μA	25°C			
		20	68	μA	85°C			
所有器件		55	180	μA	-40°C	VDD = 5.0V		
		55	180	μA	25°C			
		50	180	μA	85°C			
PIC18LF2X31/4X31		140	220	μA	-40°C	VDD = 2.0V		FOSC = 1 MHz (RC_RUN 模式, 内部振荡源)
		145	220	μA	25°C			
		155	220	μA	85°C			
PIC18LF2X31/4X31		215	330	μA	-40°C	VDD = 3.0V		
		225	330	μA	25°C			
		235	330	μA	85°C			
所有器件		385	550	μA	-40°C	VDD = 5.0V		
		390	550	μA	25°C			
		405	550	μA	85°C			

图注: 阴影行是为了增强表的可读性。

- 注 1: 休眠模式下的掉电电流与振荡器类型无关。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并连接到 VDD 或者 VSS, 且禁止了所有会增加电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 2: 供电电流主要随工作电压、频率和模式而变化。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在有源工作模式下, 所有 IDD 测量的测试条件为:
OSC1= 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD;
MCLR=VDD; WDT 按规定使能 / 禁止。
- 3: 器件配置为 RC 振荡器模式时, 该电流不包括流经 REXT 的电流。流经此电阻的电流可以通过公式 $I_r = V_{DD} / 2R_{EXT}$ (mA) 进行估算, 其中 REXT 的单位是 k Ω 。
- 4: 标准低成本 32 kHz 晶体振荡器的工作温度范围是从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶体振荡器成本就更高一些。

PIC18F2331/2431/4331/4431

25.2 DC 特性: 掉电电流和供电电流 PIC18F2331/2431/4331/4431 (工业级, 扩展级) PIC18LF2331/2431/4331/4431 (工业级) (续)

PIC18F2331/2431/4331/4431 (工业级, 扩展级)			标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)				
PIC18LF2331/2431/4331/4431 (工业级)			标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)				
参数号	器件	典型值	最大值	单位	条件		
	PIC18LF2X31/4X31	410	600	μA	-40°C	VDD = 2.0V	FOSC = 4 MHz (RC_RUN 模式, 内部振荡源)
		425	600	μA	25°C		
		435	600	μA	85°C		
	PIC18LF2X31/4X31	650	900	μA	-40°C	VDD = 3.0V	
		670	900	μA	25°C		
		680	900	μA	85°C		
	所有器件	1.2	1.8	mA	-40°C	VDD = 5.0V	
		1.2	1.8	mA	25°C		
		1.2	1.8	mA	85°C		

图注: 阴影行是为了增强表的可读性。

- 注 1: 休眠模式下的掉电电流与振荡器类型无关。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并连接到 VDD 或者 VSS, 且禁止了所有会增加电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 2: 供电电流主要随工作电压、频率和模式而变化。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在有源工作模式下, 所有 IDD 测量的测试条件为:
OSC1= 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD ;
MCLR=VDD ; WDT 按规定使能 / 禁止。
- 3: 器件配置为 RC 振荡器模式时, 该电流不包括流经 REXT 的电流。流经此电阻的电流可以通过公式 $I_r = V_{DD} / 2R_{EXT}$ (mA) 进行估算, 其中 REXT 的单位是 k Ω 。
- 4: 标准低成本 32 kHz 晶体振荡器的工作温度范围是从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶体振荡器成本就更高一些。

PIC18F2331/2431/4331/4431

25.2 DC 特性:

掉电电流和供电电流

PIC18F2331/2431/4331/4431 (工业级, 扩展级)

PIC18LF2331/2431/4331/4431 (工业级) (续)

PIC18F2331/2431/4331/4431 (工业级, 扩展级)		标准工作条件 (除非另有声明)					
		工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
PIC18LF2331/2431/4331/4431 (工业级)		标准工作条件 (除非另有声明)					
		工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
参数号	器件	典型值	最大值	单位	条件		
供电电流 (IDD) (2,3)							
PIC18LF2X31/4X31		4.7	8	μA	-40°C	VDD = 2.0V	FOSC = 31 kHz (RC_IDLE 模式, 内部振荡源)
		5.0	8	μA	25°C		
		5.8	11	μA	85°C		
PIC18LF2X31/4X31		7.0	11	μA	-40°C	VDD = 3.0V	
		7.8	11	μA	25°C		
		8.7	15	μA	85°C		
所有器件		12	16	μA	-40°C	VDD = 5.0V	
		14	16	μA	25°C		
		14	22	μA	85°C		
PIC18LF2X31/4X31		75	150	μA	-40°C	VDD = 2.0V	FOSC = 1 MHz (RC_IDLE 模式, 内部振荡源)
		85	150	μA	25°C		
		95	150	μA	85°C		
PIC18LF2X31/4X31		110	180	μA	-40°C	VDD = 3.0V	
		125	180	μA	25°C		
		135	180	μA	85°C		
所有器件		180	300	μA	-40°C	VDD = 5.0V	
		195	300	μA	25°C		
		200	300	μA	85°C		

图注:

阴影行是为了增强表的可读性。

注

- 休眠模式下的掉电电流与振荡器类型无关。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并连接到 VDD 或者 VSS, 且禁止了所有会增加电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 供电电流主要随工作电压、频率和模式而变化。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在有源工作模式下, 所有 IDD 测量的测试条件为:
OSC1= 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD;
MCLR=VDD; WDT 按规定使能 / 禁止。
- 器件配置为 RC 振荡器模式时, 该电流不包括流经 REXT 的电流。流经此电阻的电流可以通过公式 $I_r = V_{DD} / 2R_{EXT}$ (mA) 进行估算, 其中 REXT 的单位是 k Ω 。
- 标准低成本 32 kHz 晶体振荡器的工作温度范围是从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶体振荡器成本就更高一些。

PIC18F2331/2431/4331/4431

25.2 DC 特性: 掉电电流和供电电流 PIC18F2331/2431/4331/4431 (工业级, 扩展级) PIC18LF2331/2431/4331/4431 (工业级) (续)

PIC18F2331/2431/4331/4431 (工业级, 扩展级)			标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)				
PIC18LF2331/2431/4331/4431 (工业级)			标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)				
参数号	器件	典型值	最大值	单位	条件		
	PIC18LF2X31/4X31	175	275	μA	-40°C	VDD = 2.0V	FOSC = 4 MHz (RC_IDLE 模式, 内部振荡源)
		185	275	μA	25°C		
		195	275	μA	85°C		
	PIC18LF2X31/4X31	265	375	μA	-40°C	VDD = 3.0V	
		280	375	μA	25°C		
		300	375	μA	85°C		
	所有器件	475	800	μA	-40°C	VDD = 5.0V	
		500	800	μA	25°C		
		505	800	μA	85°C		

图注: 阴影行是为了增强表的可读性。

- 注 1: 休眠模式下的掉电电流与振荡器类型无关。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并连接到 VDD 或者 VSS, 且禁止了所有会增加电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 2: 供电电流主要随工作电压、频率和模式而变化。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在有源工作模式下, 所有 IDD 测量的测试条件为:
OSC1= 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD ;
MCLR=VDD ; WDT 按规定使能 / 禁止。
- 3: 器件配置为 RC 振荡器模式时, 该电流不包括流经 REXT 的电流。流经此电阻的电流可以通过公式 $I_r = V_{DD} / 2R_{EXT}$ (mA) 进行估算, 其中 REXT 的单位是 k Ω 。
- 4: 标准低成本 32 kHz 晶体振荡器的工作温度范围是从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶体振荡器成本就更高一些。

PIC18F2331/2431/4331/4431

25.2 DC 特性:

掉电电流和供电电流

PIC18F2331/2431/4331/4431 (工业级, 扩展级)

PIC18LF2331/2431/4331/4431 (工业级) (续)

PIC18F2331/2431/4331/4431 (工业级, 扩展级)		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
PIC18LF2331/2431/4331/4431 (工业级)		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
参数号	器件	典型值	最大值	单位	条件		
供电电流 (IDD) (2,3)							
PIC18LF2X31/4X31		150	250	μA	-40°C	VDD = 2.0V	Fosc = 1 MHz (PRI_RUN, EC 振荡器)
		150	250	μA	25°C		
		160	250	μA	85°C		
PIC18LF2X31/4X31		340	350	μA	-40°C	VDD = 3.0V	
		300	350	μA	25°C		
		280	350	μA	85°C		
所有器件		0.72	1.0	mA	-40°C	VDD = 5.0V	
		0.63	1.0	mA	25°C		
		0.57	1.0	mA	85°C		
PIC18LF2X31/4X31		440	600	μA	-40°C	VDD = 2.0V	Fosc = 4 MHz (PRI_RUN, EC 振荡器)
		450	600	μA	25°C		
		460	600	μA	85°C		
PIC18LF2X31/4X31		0.80	1.0	mA	-40°C	VDD = 3.0V	
		0.78	1.0	mA	25°C		
		0.77	1.0	mA	85°C		
所有器件		1.6	2.0	mA	-40°C	VDD = 5.0V	
		1.5	2.0	mA	25°C		
		1.5	2.0	mA	85°C		
所有器件		9.5	12	mA	-40°C	VDD = 4.2V	Fosc = 40 MHz (PRI_RUN, EC 振荡器)
		9.7	12	mA	25°C		
		9.9	12	mA	85°C		
所有器件		11.9	15	mA	-40°C	VDD = 5.0V	
		12.1	15	mA	25°C		
		12.3	15	mA	85°C		

图注: 阴影行是为了增强表的可读性。

- 注 1: 休眠模式下的掉电电流与振荡器类型无关。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并连接到 VDD 或者 VSS, 且禁止了所有会增加电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 2: 供电电流主要随工作电压、频率和模式而变化。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在有源工作模式下, 所有 IDD 测量的测试条件为:
OSC1= 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD;
MCLR=VDD; WDT 按规定使能 / 禁止。
- 3: 器件配置为 RC 振荡器模式时, 该电流不包括流经 REXT 的电流。流经此电阻的电流可以通过公式 $I_r = V_{DD} / 2R_{EXT}$ (mA) 进行估算, 其中 REXT 的单位是 k Ω 。
- 4: 标准低成本 32 kHz 晶体振荡器的工作温度范围是从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶体振荡器成本就更高一些。

PIC18F2331/2431/4331/4431

25.2 DC 特性:

掉电电流和供电电流

PIC18F2331/2431/4331/4431 (工业级, 扩展级)

PIC18LF2331/2431/4331/4431 (工业级) (续)

PIC18F2331/2431/4331/4431 (工业级, 扩展级)			标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)				
PIC18LF2331/2431/4331/4431 (工业级)			标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)				
参数号	器件	典型值	最大值	单位	条件		
供电电流 (I_{DD}) (2,3)							
PIC18LF2X31/4X31		35	50	μA	-40°C	$V_{DD} = 2.0\text{V}$	Fosc = 1 MHz (PRI_IDLE 模式, EC 振荡器)
		35	50	μA	25°C		
		35	60	μA	85°C		
PIC18LF2X31/4X31		55	80	μA	-40°C	$V_{DD} = 3.0\text{V}$	
		50	80	μA	25°C		
		60	100	μA	85°C		
所有器件		105	150	μA	-40°C	$V_{DD} = 5.0\text{V}$	
		110	150	μA	25°C		
		115	150	μA	85°C		
PIC18LF2X31/4X31		135	180	μA	-40°C	$V_{DD} = 2.0\text{V}$	Fosc = 4 MHz (PRI_IDLE 模式, EC 振荡器)
		140	180	μA	25°C		
		140	180	μA	85°C		
PIC18LF2X31/4X31		215	280	μA	-40°C	$V_{DD} = 3.0\text{V}$	
		225	280	μA	25°C		
		230	280	μA	85°C		
所有器件		410	525	μA	-40°C	$V_{DD} = 5.0\text{V}$	
		420	525	μA	25°C		
		430	525	μA	85°C		
所有器件		3.2	4.1	mA	-40°C	$V_{DD} = 4.2\text{V}$	Fosc = 40 MHz (PRI_IDLE 模式, EC 振荡器)
		3.2	4.1	mA	25°C		
		3.3	4.1	mA	85°C		
所有器件		4.0	5.1	mA	-40°C	$V_{DD} = 5.0\text{V}$	
		4.1	5.1	mA	25°C		
		4.1	5.1	mA	85°C		

图注: 阴影行是为了增强表的可读性。

- 注 1: 休眠模式下的掉电电流与振荡器类型无关。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并连接到 V_{DD} 或者 V_{SS} , 且禁止了所有会增加电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 2: 供电电流主要随工作电压、频率和模式而变化。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在有源工作模式下, 所有 I_{DD} 测量的测试条件为:
 $OSC1 =$ 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 V_{DD} ;
 $MCLR = V_{DD}$; WDT 按规定使能 / 禁止。
- 3: 器件配置为 RC 振荡器模式时, 该电流不包括流经 R_{EXT} 的电流。流经此电阻的电流可以通过公式 $I_r = V_{DD} / 2R_{EXT}$ (mA) 进行估算, 其中 R_{EXT} 的单位是 $k\Omega$ 。
- 4: 标准低成本 32 kHz 晶体振荡器的工作温度范围是从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶体振荡器成本就更高一些。

PIC18F2331/2431/4331/4431

25.2 DC 特性:

掉电电流和供电电流

PIC18F2331/2431/4331/4431 (工业级, 扩展级)

PIC18LF2331/2431/4331/4431 (工业级) (续)

PIC18F2331/2431/4331/4431 (工业级, 扩展级)		标准工作条件 (除非另有声明)				
		工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)				
PIC18LF2331/2431/4331/4431 (工业级)		标准工作条件 (除非另有声明)				
		工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)				
参数号	器件	典型值	最大值	单位	条件	
供电电流 (IDD) (2,3)						
	PIC18LF2X31/4X31	5.1	9	μA	-10°C	$V_{DD} = 2.0\text{V}$ $V_{DD} = 3.0\text{V}$ $V_{DD} = 5.0\text{V}$
		5.8	9	μA	25°C	
		7.9	11	μA	70°C	
	PIC18LF2X31/4X31	7.9	12	μA	-10°C	
		8.9	12	μA	25°C	
		10.5	14	μA	70°C	
	所有器件	12.5	20	μA	-10°C	
		16.3	20	μA	25°C	
		18.9	25	μA	70°C	
供电电流 (IDD) (2,3)						
	PIC18LF2X31/4X31	9.2	15	μA	-10°C	$V_{DD} = 2.0\text{V}$ $V_{DD} = 3.0\text{V}$ $V_{DD} = 5.0\text{V}$
		9.6	15	μA	25°C	
		12.7	18	μA	70°C	
	PIC18LF2X31/4X31	22.0	30	μA	-10°C	
		21.0	30	μA	25°C	
		20.0	35	μA	70°C	
	所有器件	30	80	μA	-10°C	
		45	80	μA	25°C	
		45	85	μA	70°C	

图注:

阴影行是为了增强表的可读性。

注

- 1: 休眠模式下的掉电电流与振荡器类型无关。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并连接到 VDD 或者 VSS, 且禁止了所有会增加电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 2: 供电电流主要随工作电压、频率和模式而变化。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在有源工作模式下, 所有 IDD 测量的测试条件为:
OSC1= 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 VDD ;
MCLR=VDD; WDT 按规定使能 / 禁止。
- 3: 器件配置为 RC 振荡器模式时, 该电流不包括流经 REXT 的电流。流经此电阻的电流可以通过公式 $I_r = V_{DD}/2R_{EXT}$ (mA) 进行估算, 其中 REXT 的单位是 k Ω 。
- 4: 标准低成本 32 kHz 晶体振荡器的工作温度范围是从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶体振荡器成本就更高一些。

PIC18F2331/2431/4331/4431

25.2 DC 特性:

掉电电流和供电电流

PIC18F2331/2431/4331/4431 (工业级, 扩展级)

PIC18LF2331/2431/4331/4431 (工业级) (续)

PIC18F2331/2431/4331/4431 (工业级, 扩展级)		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
PIC18LF2331/2431/4331/4431 (工业级)		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
参数号	器件	典型值	最大值	单位	条件		
模块电流差 (ΔI_{WDT} 、 ΔI_{BOR} 、 ΔI_{LVD} 、 ΔI_{OSCB} 、 ΔI_{AD})							
D022 (ΔI_{WDT})	看门狗定时器	1.5	4.0	μA	-40°C	$V_{\text{DD}} = 2.0\text{V}$	
		2.2	4.0	μA	25°C		
		3.1	5.0	μA	85°C		
		$V_{\text{DD}} = 3.0\text{V}$	2.5	6.0	μA	-40°C	
			3.3	6.0	μA	25°C	
			4.7	7.0	μA	85°C	
			$V_{\text{DD}} = 5.0\text{V}$	3.7	10.0	μA	-40°C
				4.5	10.0	μA	25°C
				6.1	13.0	μA	85°C
D022A (ΔI_{BOR})	欠压复位	19	35.0	μA	-40°C 至 $+85^{\circ}\text{C}$	$V_{\text{DD}} = 3.0\text{V}$	
		24	45.0	μA	-40°C 至 $+85^{\circ}\text{C}$	$V_{\text{DD}} = 5.0\text{V}$	
D022B (ΔI_{LVD})	低压检测	8.5	25.0	μA	-40°C 至 $+85^{\circ}\text{C}$	$V_{\text{DD}} = 2.0\text{V}$	
		16	35.0	μA	-40°C 至 $+85^{\circ}\text{C}$	$V_{\text{DD}} = 3.0\text{V}$	
		20	45.0	μA	-40°C 至 $+85^{\circ}\text{C}$	$V_{\text{DD}} = 5.0\text{V}$	
D025 (ΔI_{OSCB})	Timer1 振荡器	1.7	3.5	μA	-40°C	$V_{\text{DD}} = 2.0\text{V}$	Timer1 ⁽⁴⁾ 振荡器为 32 kHz
		1.8	3.5	μA	25°C		
		2.1	4.5	μA	85°C		
		$V_{\text{DD}} = 3.0\text{V}$	2.2	4.5	μA	-40°C	
			2.6	4.5	μA	25°C	
			2.8	5.5	μA	85°C	
		$V_{\text{DD}} = 5.0\text{V}$	3.0	6.0	μA	-40°C	
			3.3	6.0	μA	25°C	
3.6	7.0		μA	85°C			
D026 (ΔI_{AD})	A/D 转换器	1.0	3.0	μA	-40°C 到 85°C	$V_{\text{DD}} = 2.0\text{V}$	A/D 开启, 但不在进行转换
		1.0	4.0	μA	-40°C 到 85°C	$V_{\text{DD}} = 3.0\text{V}$	
		2.0	10.0	μA	-40°C 到 85°C	$V_{\text{DD}} = 5.0\text{V}$	

图注: 阴影行是为了增强表的可读性。

- 注 1: 休眠模式下的掉电电流与振荡器类型无关。掉电电流是在器件休眠时, 所有 I/O 引脚处于高阻态并连接到 V_{DD} 或者 V_{SS} , 且禁止了所有会增加电流的功能部件 (比如 WDT, Timer1 振荡器, BOR 等) 时测得的。
- 2: 供电电流主要随工作电压、频率和模式而变化。其他因素, 如 I/O 引脚负载和开关频率、振荡器类型和电路、内部代码执行模式和温度也会影响电流消耗。
在有源工作模式下, 所有 I_{DD} 测量的测试条件为:
OSC1= 外部方波, 满幅; 所有 I/O 引脚均为三态, 拉至 V_{DD} ;
MCLR= V_{DD} ; WDT 按规定使能 / 禁止。
- 3: 器件配置为 RC 振荡器模式时, 该电流不包括流经 R_{EXT} 的电流。流经此电阻的电流可以通过公式 $I_r = V_{\text{DD}}/2R_{\text{EXT}}$ (mA) 进行估算, 其中 R_{EXT} 的单位是 $\text{k}\Omega$ 。
- 4: 标准低成本 32 kHz 晶体振荡器的工作温度范围是从 -10°C 到 $+70^{\circ}\text{C}$ 。扩展级温度的晶体振荡器成本就更高一些。

PIC18F2331/2431/4331/4431

25.3 DC 特性

PIC18F2331/2431/4331/4431 (工业级)
PIC18LF2331/2431/4331/4431 (工业级)

DC 特性			标准工作条件 (除非另有声明) 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	V _{IL}	输入低电压 I/O 端口: 带 TTL 缓冲器	V _{SS}	0.15 V _{DD}	V	V _{DD} < 4.5V	
D030A			—	0.8	V	4.5V ≤ V _{DD} ≤ 5.5V	
D031		带施密特触发缓冲器 RC3 和 RC4	V _{SS}	0.2 V _{DD}	V	LP、XT、HS 和 HSPLL 模式 ⁽¹⁾ EC 模式 ⁽¹⁾	
D032			V _{SS}	0.3 V _{DD}	V		
D032A		MCLR	V _{SS}	0.2 V _{DD}	V		
D032A		OSC1 和 T1OSI	V _{SS}	0.3 V _{DD}	V		
D033		OSC1	V _{SS}	0.2 V _{DD}	V		
D040	V _{IH}	输入高电压 I/O 端口: 带 TTL 缓冲器	0.25 V _{DD} + 0.8V	V _{DD}	V		V _{DD} < 4.5V
D040A			2.0	V _{DD}	V		4.5V ≤ V _{DD} ≤ 5.5V
D041		带施密特触发缓冲器 RC3 和 RC4	0.8 V _{DD}	V _{DD}	V	LP、XT、HS 和 HSPLL 模式 ⁽¹⁾ EC 模式 ⁽¹⁾	
D042			0.7 V _{DD}	V _{DD}	V		
D042A		MCLR	0.8 V _{DD}	V _{DD}	V		
D042A		OSC1 和 T1OSI	0.7 V _{DD}	V _{DD}	V		
D043		OSC1	0.8 V _{DD}	V _{DD}	V		
D060	I _{IL}	输入泄漏电流 ^(2,3) I/O 端口	—	±1	μA		V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态
D061			MCLR	—	±1		μA
D063			OSC1	—	±1	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD}
D070	I _{PU} I _{PURB}	弱上拉电流 PORTB 弱上拉电流	50	400	μA	V _{DD} = 5V, V _{PIN} = V _{SS}	

- 注 1: 在 RC 振荡器配置中, OSC1/CLKI 引脚是施密特触发输入。在 RC 模式中建议不要使用外部时钟驱动 PICmicro 器件。
- 2: MCLR 引脚上的泄漏电流主要取决于所施加的电平。规定的电平表示正常工作条件。不同的输入电压可能导致测得更高的泄漏电流。
- 3: 负电流定义为引脚的拉电流。
- 4: 参数仅为特性数据, 未经测试。

PIC18F2331/2431/4331/4431

25.3 DC 特性

PIC18F2331/2431/4331/4431 (工业级)
PIC18LF2331/2431/4331/4431 (工业级) (续)

DC 特性			标准工作条件 (除非另有声明) 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
D080	VOL	输出低电压 I/O 端口	—	0.6	V	$I_{OL}=8.5\text{ mA}$, $V_{DD}=4.5\text{V}$, -40°C 至 $+85^{\circ}\text{C}$
D083		OSC2/CLKO (RC、RCIO、EC 和 ECIO 模式)	—	0.6	V	$I_{OL}=1.6\text{ mA}$, $V_{DD}=4.5\text{V}$, -40°C 至 $+85^{\circ}\text{C}$
D090	VOH	输出高电压 ⁽³⁾ I/O 端口	$V_{DD}-0.7$	—	V	$I_{OH}=-3.0\text{ mA}$, $V_{DD}=4.5\text{V}$, -40°C 至 $+85^{\circ}\text{C}$
D092		OSC2/CLKO (RC、RCIO、EC 和 ECIO 模式)	$V_{DD}-0.7$	—	V	$I_{OH}=-1.3\text{ mA}$, $V_{DD}=4.5\text{V}$, -40°C 至 $+85^{\circ}\text{C}$
D150	VOD	开漏高电压	—	8.5	V	RA4 引脚
D100 ⁽⁴⁾	Cosc2	输出引脚上的容性负载规范 OSC2 引脚	—	15	pF	当外部时钟用于驱动 OSC1 时, 处于 XT、HS 和 LP 模式
D101	Cio	所有 I/O 引脚和 OSC2 (RC 模式)	—	50	pF	满足 AC 时序规范
D102	CB	SCL, SDA	—	400	pF	I ² C™ 规范

- 注 1: 在 RC 振荡器配置中, OSC1/CLKI 引脚是施密特触发输入。在 RC 模式中建议不要使用外部时钟驱动 PICmicro 器件。
- 2: MCLR 引脚上的泄漏电流主要取决于所施加的电平。规定的电平表示正常工作条件。不同的输入电压可能导致测得更高的泄漏电流。
- 3: 负电流定义为引脚的拉电流。
- 4: 参数仅为特性数据, 未经测试。

PIC18F2331/2431/4331/4431

表 25-1: 存储器编程要求

DC 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	典型值 †	最大值	单位	条件
内部程序存储器编程规范 (1)							
D110	VPP	MCLR/VPP 引脚上的电压	9.00	—	13.25	V	(注 3)
D112	IPP	MCLR/VPP 引脚的电流	—	—	300	μA	
D113	IDDP	编程期间的供电电流	—	—	1	mA	
数据 EEPROM 存储器							
D120	Ed	耐擦写能力	100K	1M	—	E/W	-40°C 至 +85°C 用 EECON 进行读 / 写操作 V _{MIN} = 最小工作电压
D121	VDRW	读 / 写操作时的 VDD	V _{MIN}	—	5.5	V	
D122	TDEW	擦 / 写周期时间	—	4	—	ms	假设没有违反其他规范
D123	TRETD	特性保持时间	40	—	—	年	
D124	TREF	在刷新 (2) 之前的总擦写次数	1M	10M	—	E/W	
闪存程序存储器							
D130	EP	电池耐久性	10K	100K	—	E/W	-40°C 至 +85°C V _{MIN} = 最小工作电压
D131	VPR	读操作时的 VDD	V _{MIN}	—	5.5	V	
D132	VIE	块擦除时的 VDD	4.5	—	5.5	V	使用 ICSP 端口
D132A	VIW	外部定时擦写时的 VDD	4.5	—	5.5	V	使用 ICSP 端口
D132B	VPEW	自定时写时的 VDD	V _{MIN}	—	5.5	V	V _{MIN} = 最小工作电压
D133	TIE	ICSP 块擦除周期时间	—	4	—	ms	VDD > 4.5V
D133A	TIW	ICSP 擦或写周期时间 (外部定时)	1	—	—	ms	VDD > 4.5V
D133A	TIW	自定时写周期时间	—	2	—	ms	
D134	TRETD	特性保持时间	40	100	—	年	假设没有违反其他规范

† 除非另外声明, 否则“典型值”栏中的数据均在 5.0V、25°C 下测得。这些参数仅供设计参考, 未经测试。

- 注 1: 这些编程规范适用于通过写表指令对片上程序存储器编程。
 2: 有关数据 EEPROM 耐擦写能力的更详细讨论, 请参阅第 7.8 节“使用数据 EEPROM”。
 3: 只有在低电压编程被禁止时才需要。

PIC18F2331/2431/4331/4431

图 25-3: 低压检测特性

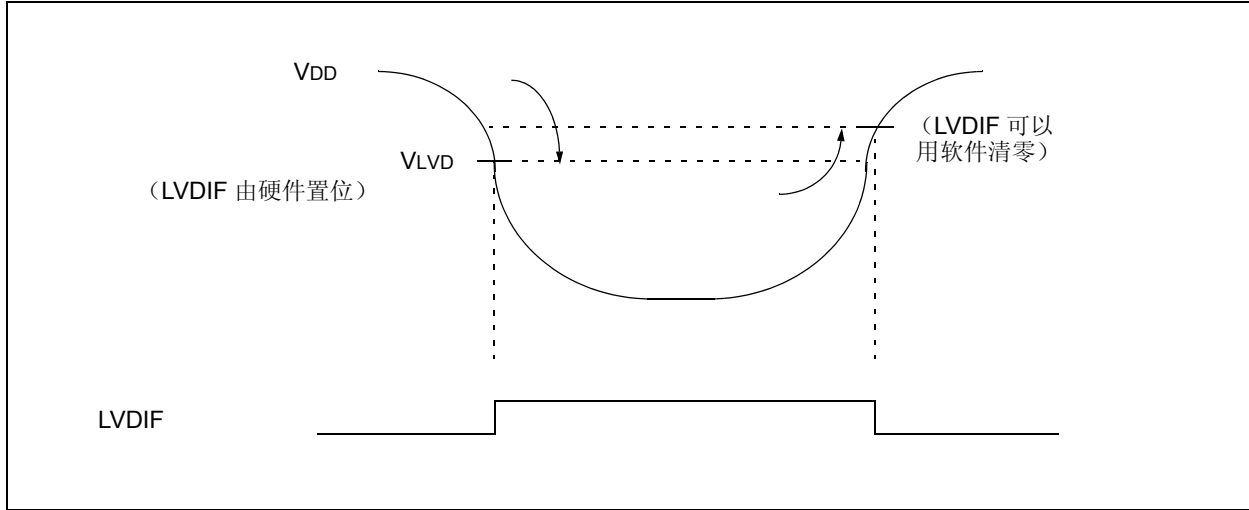


表 25-2: 低压检测特性

		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)						
参数号	符号	特性	最小值	典型值 †	最大值	单位	条件	
D420		VDD 由高电平变为低电平时的 LVD 电压	LVV = 0010	2.08	2.26	2.44	V	
			LVV = 0011	2.26	2.45	2.65	V	
			LVV = 0100	2.35	2.55	2.76	V	
			LVV = 0101	2.55	2.77	2.99	V	
			LVV = 0110	2.64	2.87	3.10	V	
			LVV = 0111	2.82	3.07	3.31	V	
			LVV = 1000	3.09	3.36	3.63	V	
			LVV = 1001	3.29	3.57	3.86	V	
			LVV = 1010	3.38	3.67	3.96	V	
			LVV = 1011	3.56	3.87	4.18	V	
			LVV = 1100	3.75	4.07	4.40	V	
			LVV = 1101	3.93	4.28	4.62	V	
		LVV = 1110	4.23	4.60	4.96	V		

† 生产测试温度 $T_{AMB} = 25^{\circ}\text{C}$ 。超过温度限制的规范由器件特性确保。

25.4 AC（时序）特性

25.4.1 时序参数符号

可根据以下格式中的一种来创建时序参数符号：

- | | | |
|-------------|-----------|-------------------------|
| 1. TppS2ppS | 3. Tcc:ST | （仅 I ² C 规范） |
| 2. TppS | 4. Ts | （仅 I ² C 规范） |

T		T	
F	频率	T	时间

小写字母（pp）及其含意：

pp		osc	OSC1
cc	CCP1	rd	\overline{RD}
ck	CLKO	rw	\overline{RD} or \overline{WR}
cs	\overline{CS}	sc	SCK
di	SDI	ss	\overline{SS}
do	SDO	t0	T0CKI
dt	数据输入	t1	T1CKI
io	I/O 端口	wr	\overline{WR}
mc	MCLR		

大写字母及其含意：

S		P	周期
F	下降	R	上升
H	高电平	V	有效
I	无效（高阻态）	Z	高阻态
L	低电平		
仅用于 I ² C 模式			
AA	输出访问	HIGH	高电平
BUF	总线空闲	LOW	低电平

Tcc: ST（仅 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		

PIC18F2331/2431/4331/4431

25.4.2 时序条件

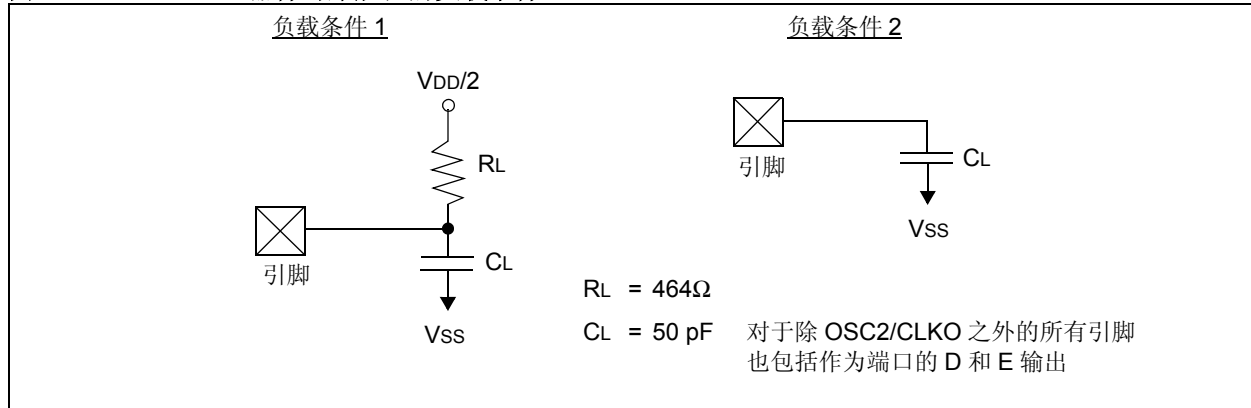
除非另外指明，否则表 25-3 中指定的温度和电压适用于所有时序规范。图 25-4 规定了时序规范的负载条件。

注： 由于篇幅所限，本节中通称的“PIC18FXX31”和“PIC18LFXX31”分别指代（而且仅指代）PIC18F2331/2431/4331/4431 和 PIC18LF2331/2431/4331/4431 系列器件。

表 25-3: 温度和电压规范 — 交流

AC 特性	标准工作条件（除非另有声明）
	工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ （工业级） 工作电压 V_{DD} 的范围如 DC 规范第 25.1 节和第 25.3 节所述。LF 器件仅在工业级温度下工作。

图 25-4: 器件时序规范的负载条件



PIC18F2331/2431/4331/4431

25.4.3 时序图和规范

图 25-5: 外部时钟时序 (除 PLL 之外的所有模式)

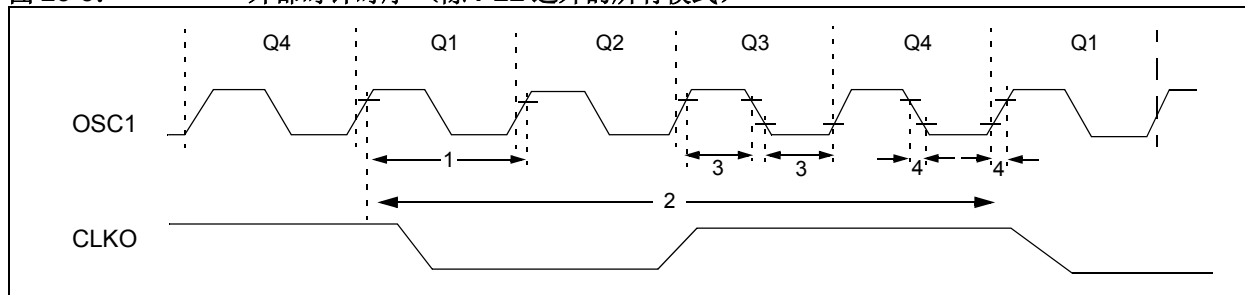


表 25-4: 外部时钟时序要求

参数编号	符号	特性	最小值	最大值	单位	条件
1A	FOSC	外部 CLKI 频率 ⁽¹⁾ 振荡器频率 ⁽¹⁾	DC	40	MHz	EC, ECIO
			DC	4	MHz	RC 振荡模式
			0.1	4	MHz	XT 振荡模式
			4	25	MHz	HS 振荡模式
			4	10	MHz	HS + PLL 振荡模式
			5	200	kHz	LP 振荡模式
1	Tosc	外部 CLKI 周期 ⁽¹⁾ 振荡器周期 ⁽¹⁾	25	—	ns	EC, ECIO
			250	—	ns	RC 振荡模式
			250	10,000	ns	XT 振荡模式
			25	250	ns	HS 振荡模式
			100	250	ns	HS + PLL 振荡模式
			25	—	μs	LP 振荡模式
2	Tcy	指令周期时间 ⁽¹⁾	100	—	ns	Tcy = 4/FOSC
3	TosL, TosH	外部时钟输入 (OSC1) 高电平或低电平时间	30	—	ns	XT 振荡模式
			2.5	—	μs	LP 振荡模式
			10	—	ns	HS 振荡模式
4	TosR, TosF	外部时钟输入 (OSC1) 上升或下降时间	—	20	ns	XT 振荡模式
			—	50	ns	LP 振荡模式
			—	7.5	ns	HS 振荡模式

注 1: 对于除 PLL 的所有配置, 指令周期 (Tcy) 等于输入振荡器时钟周期的四倍。所有指定的值均基于在标准工作条件下器件执行代码时对应特定振荡器类型的特征数据。超过这些指定的极限值可能导致振荡器运行不稳定和 / 或电流消耗比预期的要高。所有器件在“最小值”下测试时, 外部时钟接到 OSC1/CLKI 引脚。当使用了外部时钟输入时, 所有器件的“最大”周期时间极限为“DC”(没有时钟)。

PIC18F2331/2431/4331/4431

表 25-5: PLL 时钟时序规范 (VDD = 4.2V 到 5.5V)

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
F10	FOSC	振荡器频率范围	4	—	10	MHz	仅 HS 模式
F11	FSYS	片上 VCO 系统频率	16	—	40	MHz	仅 HS 模式
F12	TPLL	PLL 启动时间 (锁定时间)	—	—	2	ms	
F13	DCLK	CLKO 稳定性 (抖动)	-2	—	+2	%	

† 除非另外声明, 否则“典型值”栏中的数据均在 5.0V、25°C 下测得。这些参数仅作为设计时的参考, 未经测试。

表 25-6: 内部 RC 精度
PIC18F2331/2431/4331/4431 (工业级)
PIC18LF2331/2431/4331/4431 (工业级)

PIC18F1220/1320 (工业级)		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
PIC18LF1220/1320 (工业级)		标准工作条件 (除非另有声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)					
参数号	器件	最小值	典型值	最大值	单位	条件	
Freq = 8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz⁽¹⁾ 时的 INTOSC 精度							
F2	PIC18LF2331/2431/4331/4431	-15	+/-5	+15	%	25°C	VDD = 3.0V
F3	所有器件	-15	+/-5	+15	%	25°C	VDD = 5.0V
Freq = 31 kHz⁽²⁾ 时的 INTRC 精度							
F5	PIC18LF2331/2431/4331/4431	26.562	—	35.938	kHz	25°C	VDD = 3.0V
F6	所有器件	26.562	—	35.938	kHz	25°C	VDD = 5.0V
INTRC 稳定性⁽³⁾							
F8	PIC18LF2331/2431/4331/4431	TBD	1	TBD	%	25°C	VDD = 3.0V
F9	所有器件	TBD	1	TBD	%	25°C	VDD = 5.0V

图注: 使用阴影行是为了增强表的可读性。

注 1: 频率校准温度为 25°C。OSCTUNE 寄存器可以用于对温度漂移进行补充。

注 2: 校准后的 INTRC 频率。

注 3: INTRC 频率随 VDD 的改变而改变。

PIC18F2331/2431/4331/4431

图 25-6: CLKO 和 I/O 时序

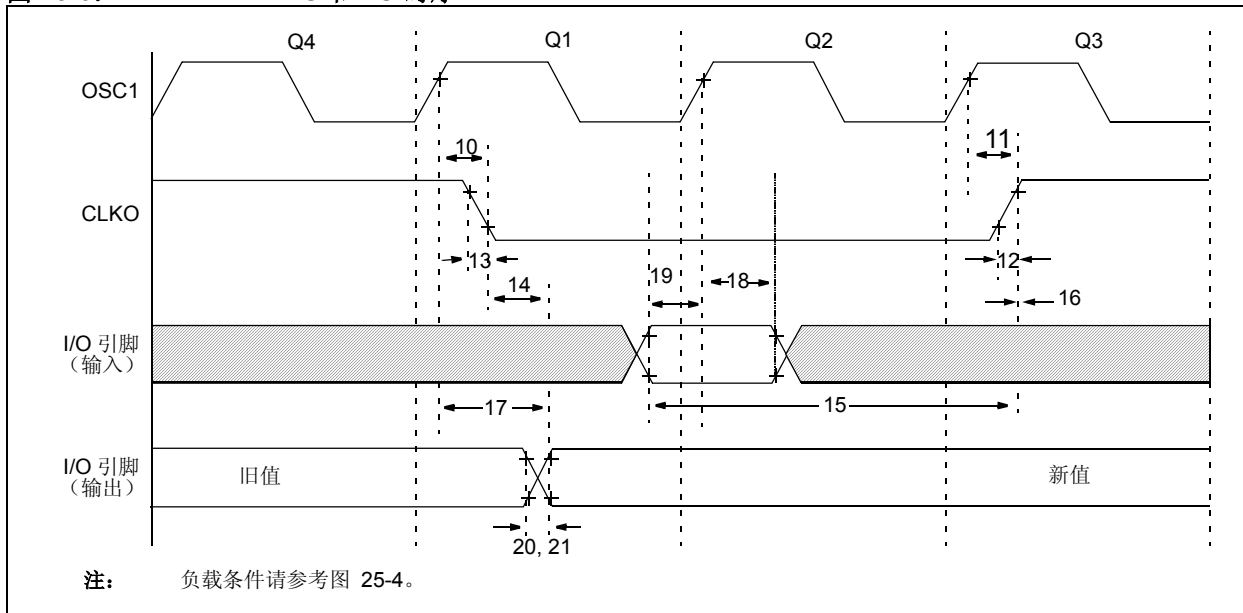


表 25-7: CLKO 和 I/O 时序要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
10	TosH2ckL	OSC1 ↑ 到 CLKO ↓	—	75	200	ns	(1)
11	TosH2ckH	OSC1 ↑ 到 CLKO ↑	—	75	200	ns	(1)
12	TckR	CLKO 上升时间	—	35	100	ns	(1)
13	TckF	CLKO 下降时间	—	35	100	ns	(1)
14	TckL2ioV	CLKO ↓ 到端口输出有效	—	—	0.5 T _{cy} + 20	ns	(1)
15	TioV2ckH	在 CLKO ↑ 前端口输入有效	0.25 T _{cy} + 25	—	—	ns	(1)
16	TckH2ioI	在 CLKO ↑ 后端口输入保持	0	—	—	ns	(1)
17	TosH2ioV	OSC1 ↑ (Q1 周期) 到端口输出有效	—	50	150	ns	
18	TosH2ioI	OSC1 ↑ (Q2 周期) 到端口输入无效 (I/O 输入保持时间)	PIC18FXX31	100	—	—	ns
18A			PIC18LFXX31	200	—	—	ns
19	TioV2osH	端口输入有效到 OSC1 ↑ (I/O 输入建立时间)	0	—	—	ns	
20	TioR	端口输出上升时间	PIC18FXX31	—	10	25	ns
20A			PIC18LFXX31	—	—	60	ns
21	TioF	端口输出下降时间	PIC18FXX31	—	10	25	ns
21A			PIC18LFXX31	—	—	60	ns
22††	TiNP	INT 引脚高电平或低电平时间	T _{cy}	—	—	ns	
23††	TRBP	RB7:RB4 变化中断高电平或低电平时间	T _{cy}	—	—	ns	
24††	TRCP	RB7:RB4 变化中断高电平或低电平时间	20	—	—	ns	

†† 这些参数为异步事件，与所有内部时钟沿无关。

注 1: 测量在 RC 模式进行，其中 CLKO 输出为 4 x T_{osc}。

PIC18F2331/2431/4331/4431

图 25-7: 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

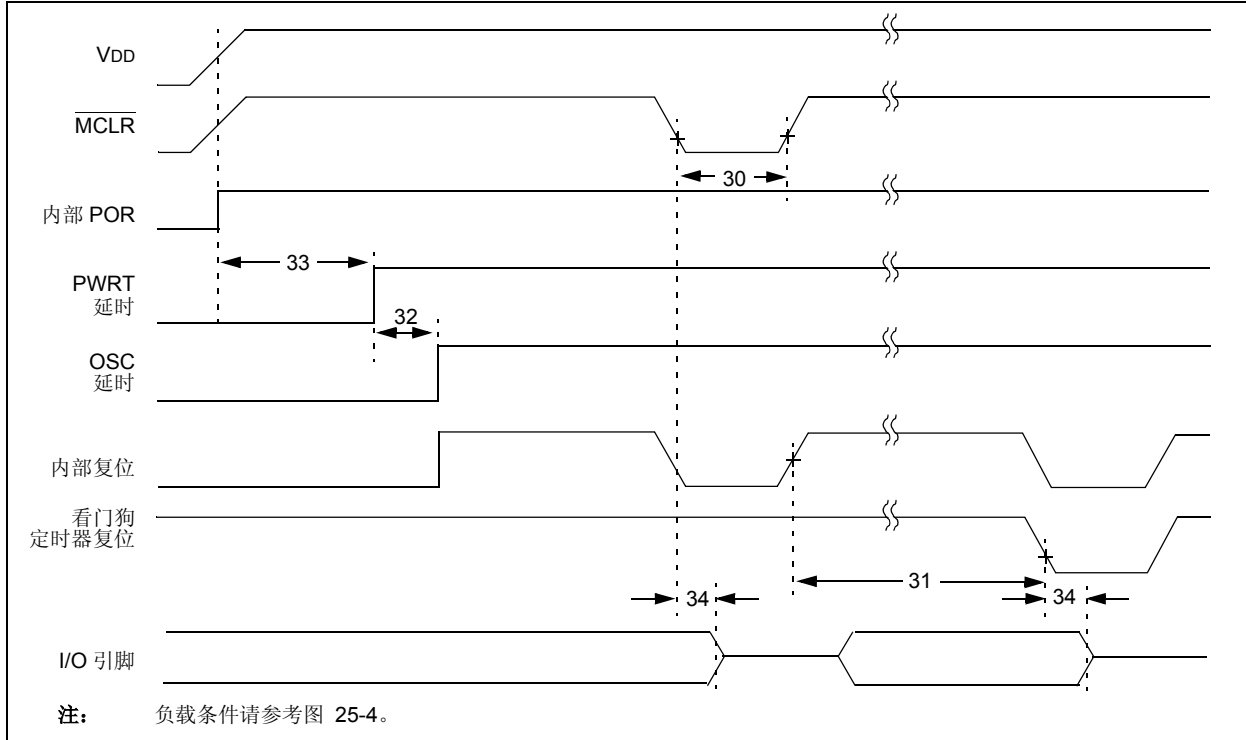


图 25-8: 欠压复位时序

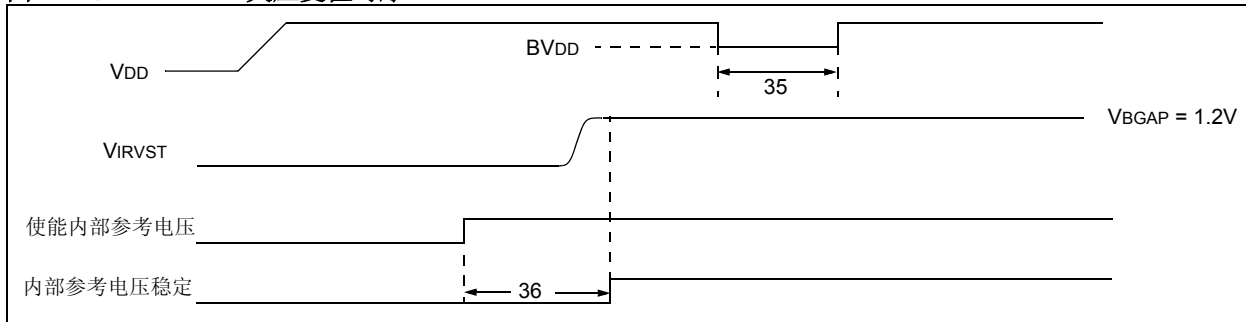


表 25-8: 复位、看门狗定时器、振荡器起振定时器、上电延时定时器和欠压复位要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
30	T _{mcl}	MCLR 脉冲宽度 (低电平)	2	—	—	μs	
31	T _{WDT}	看门狗定时器超时溢出周期 (无后分频器)	—	4.00	TBD	ms	
32	T _{OST}	振荡器起振定时器周期	1024 T _{OSC}	—	1024 T _{OSC}	—	T _{OSC} = OSC1 周期
33	T _{PWRT}	上电延时定时器周期	—	65.5	TBD	ms	
34	T _{IOZ}	MCLR 低电平或看门狗定时器复位引起 I/O 高阻态	—	2	—	μs	
35	T _{BOR}	欠压复位脉冲宽度	200	—	—	μs	V _{DD} ≤ B _{VDD} (参见 D005)
36	T _{IVRST}	内部参考电压变稳定的时间	—	20	50	μs	
37	T _{LVDT}	低压检测脉冲宽度	200	—	—	μs	V _{DD} ≤ V _{LVDT}

PIC18F2331/2431/4331/4431

图 25-9: **TIMER0 和 TIMER1 外部时钟时序**

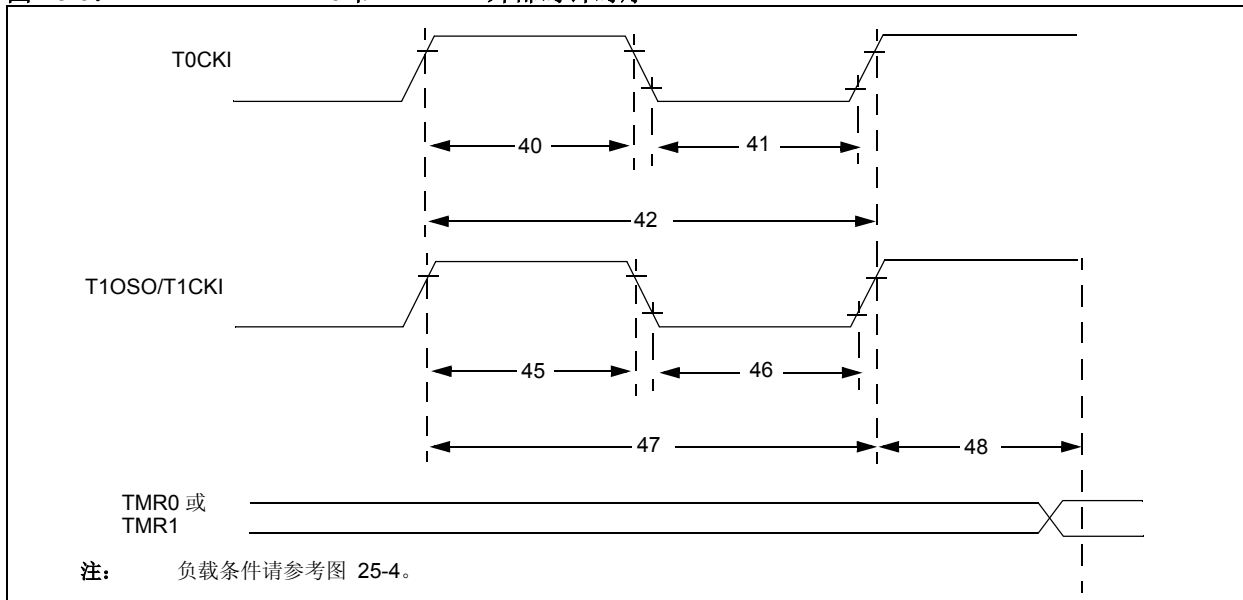


表 25-9: **TIMER0 和 TIMER1 外部时钟要求**

参数号	符号	特性		最小值	最大值	单位	条件	
40	Tt0H	T0CKI 高电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	ns		
			有预分频器	10	—	ns		
41	Tt0L	T0CKI 低电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	ns		
			有预分频器	10	—	ns		
42	Tt0P	T0CKI 周期	无预分频器	$T_{CY} + 10$	—	ns	N = 预分频比 (1, 2, 4, ..., 256)	
			有预分频器	取两者中较大值: 20 ns 或 $\frac{T_{CY} + 40}{N}$	—	ns		
45	Tt1H	T1CKI 高电平时间	同步, 无预分频器	$0.5 T_{CY} + 20$	—	ns		
			同步	PIC18FXX31	10	—		ns
			有预分频器	PIC18LFXX31	25	—		ns
			异步	PIC18FXX31	30	—		ns
46	Tt1L	T1CKI 低电平时间	同步, 无预分频器	$0.5 T_{CY} + 5$	—	ns		
			同步	PIC18FXX31	10	—		ns
			有预分频器	PIC18LFXX31	25	—		ns
			异步	PIC18FXX31	30	—		ns
47	Tt1P	T1CKI 输入周期	同步	取较大值: 20 ns 或 $\frac{T_{CY} + 40}{N}$	—	ns	N = 预分频比 (1, 2, 4, 8)	
			异步	60	—	ns		
	Ft1	T1CKI 振荡器输入频率范围		DC	50	kHz		
48	Tcke2tmr1	外部 T1CKI 时钟边沿到定时器递增的延时		$2 T_{OSC}$	$7 T_{OSC}$	—		

PIC18F2331/2431/4331/4431

图 25-10: 捕捉 / 比较 / PWM 时序 (所有 CCP 模块)

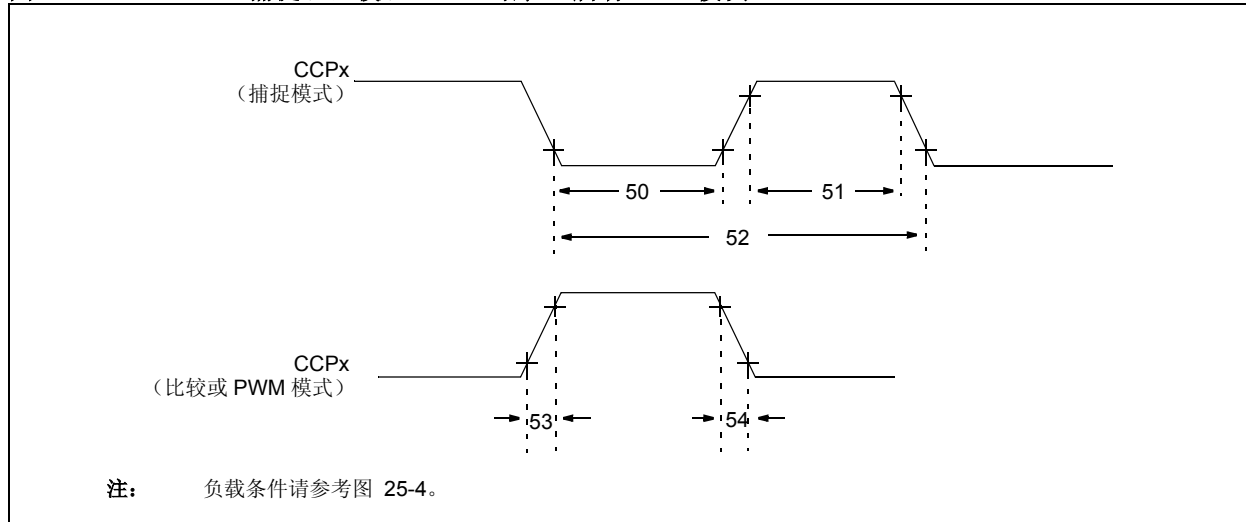


表 25-10: 捕捉 / 比较 / PWM 时序 (所有 CCP 模块)

参数号	符号	特性		最小值	最大值	单位	条件	
50	TccL	CCPx 输入低电平时间	无预分频器	$0.5 T_{CY} + 20$	—	ns		
			带预分频器	PIC18FXX31	10	—		ns
			PIC18LFXX31	20	—	ns		
51	TccH	CCPx 输入高电平时间	无预分频器	$0.5 T_{CY} + 20$	—	ns		
			带预分频器	PIC18FXX31	10	—		ns
			PIC18LFXX31	20	—	ns		
52	TccP	CCPx 输入周期		$\frac{3 T_{CY} + 40}{N}$	—	ns	N= 预分频比 (1、4 或 16)	
53	TccR	CCPx 输出上升时间	PIC18FXX31	—	25	ns		
			PIC18LFXX31	—	45	ns		
54	TccF	CCPx 输出下降时间	PIC18FXX31	—	25	ns		
			PIC18LFXX31	—	45	ns		

PIC18F2331/2431/4331/4431

图 25-11: SPI 主控模式时序示例 (CKE = 0)

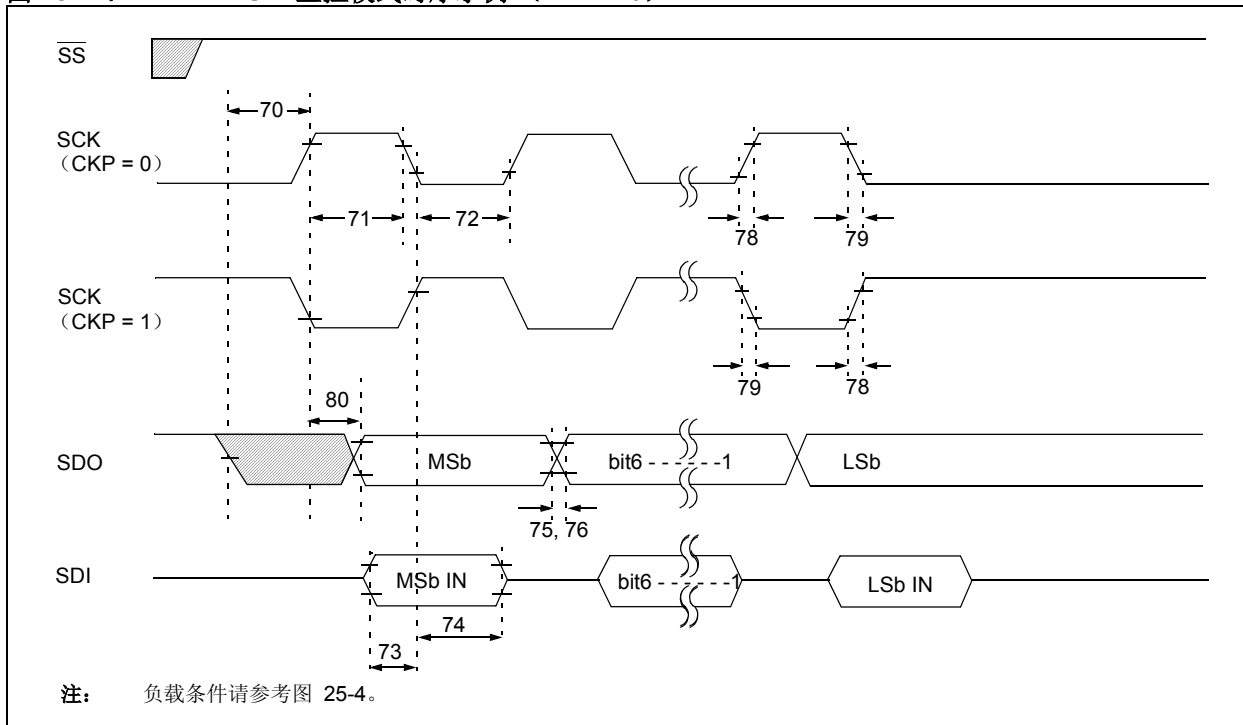


表 25-11: SPI 模式要求示例 (主控模式, CKE = 0)

参数编号	符号	特性	最小值	最大值	单位	条件
70	TssL2scH, TssL2scL	SS ↓ 到 SCK ↓ 或 SCK ↑ 输入	T _{CY}	—	ns	
71	TscH	SCK 输入高电平时间	1.25 T _{CY} + 30	—	ns	
71A		(从动模式)	连续	—	ns	
		单字节	40	—	ns	(注 1)
72	TscL	SCK 输入低电平时间	1.25 T _{CY} + 30	—	ns	
72A		(从动模式)	连续	—	ns	
		单字节	40	—	ns	(注 1)
73	TdiV2scH, TdiV2scL	SDI 数据输入到 SCK 边沿的建立时间	100	—	ns	
73A	Tb2B	Byte1 的最后一个时钟边沿到 Byte2 的第一个时钟边沿	1.5 T _{CY} + 40	—	ns	(注 2)
74	Tsch2diL, TscL2diL	SDI 数据输入到 SCK 边沿的保持时间	100	—	ns	
75	TdoR	SDO 数据输出上升时间	PIC18FXX31 —	25	ns	
			PIC18LFXX31 —	45	ns	
76	TdoF	SDO 数据输出下降时间	—	25	ns	
78	TscR	SCK 输出上升时间	PIC18FXX31 —	25	ns	
		(主控模式)	PIC18LFXX31 —	45	ns	
79	TscF	SCK 输出下降时间 (主控模式)	—	25	ns	
80	Tsch2doV, TscL2doV	SCK 边沿后 SDO 数据输出有效	PIC18FXX31 —	50	ns	
			PIC18LFXX31 —	100	ns	

注 1: 要求使用参数 73A。

注 2: 仅当使用参数 71A 和 72A 时。

PIC18F2331/2431/4331/4431

图 25-12: SPI 主控模式时序示例 (CKE = 1)

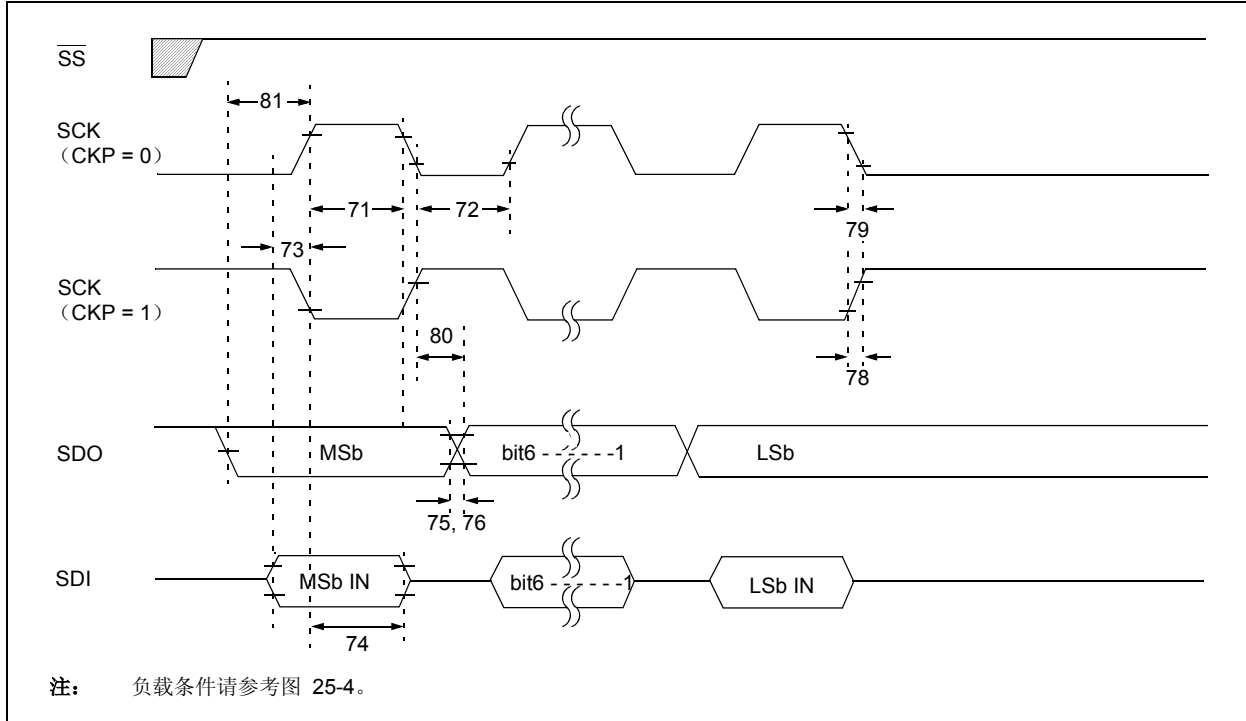


表 25-12: SPI 模式要求示例 (主控模式, CKE = 1)

参数编号	符号	特性	最小值	最大值	单位	条件
71	Tsch	SCK 输入高电平时间	1.25 Tcy + 30	—	ns	
71A		(从动模式)	连续	—	ns	(注 1)
		单字节	40	—	ns	
72	Tscl	SCK 输入低电平时间	1.25 Tcy + 30	—	ns	
72A		(从动模式)	连续	—	ns	(注 1)
		单字节	40	—	ns	
73	TdiV2sch, TdiV2scl	SDI 数据输入到 SCK 边沿的建立时间	100	—	ns	
73A	Tb2b	Byte1 最后一个时钟沿到 Byte2 第一个时钟沿	1.5 Tcy + 40	—	ns	(注 2)
74	Tsch2dil, Tscl2dil	SDI 数据输入到 SCK 边沿的保持时间	100	—	ns	
75	TdoR	SDO 数据输出上升时间	PIC18FXX31 —	25	ns	
			PIC18LFXX31	45	ns	
76	TdoF	SDO 数据输出下降时间	—	25	ns	
78	TscR	SCK 输出上升时间 (主控模式)	PIC18FXX31 —	25	ns	
			PIC18LFXX31	45	ns	
79	TscF	SCK 输出下降时间 (主控模式)	—	25	ns	
80	Tsch2doV, Tscl2doV	SCK 边沿后 SDO 数据输出有效	PIC18FXX31 —	50	ns	
			PIC18LFXX31	100	ns	
81	TdoV2sch, TdoV2scl	SDO 数据输出建立到 SCK 边沿	Tcy	—	ns	

注 1: 要求使用参数 73A。

注 2: 仅当使用参数 71A 和 72A 时。

PIC18F2331/2431/4331/4431

图 25-13: SPI 从动模式时序示例 (CKE = 0)

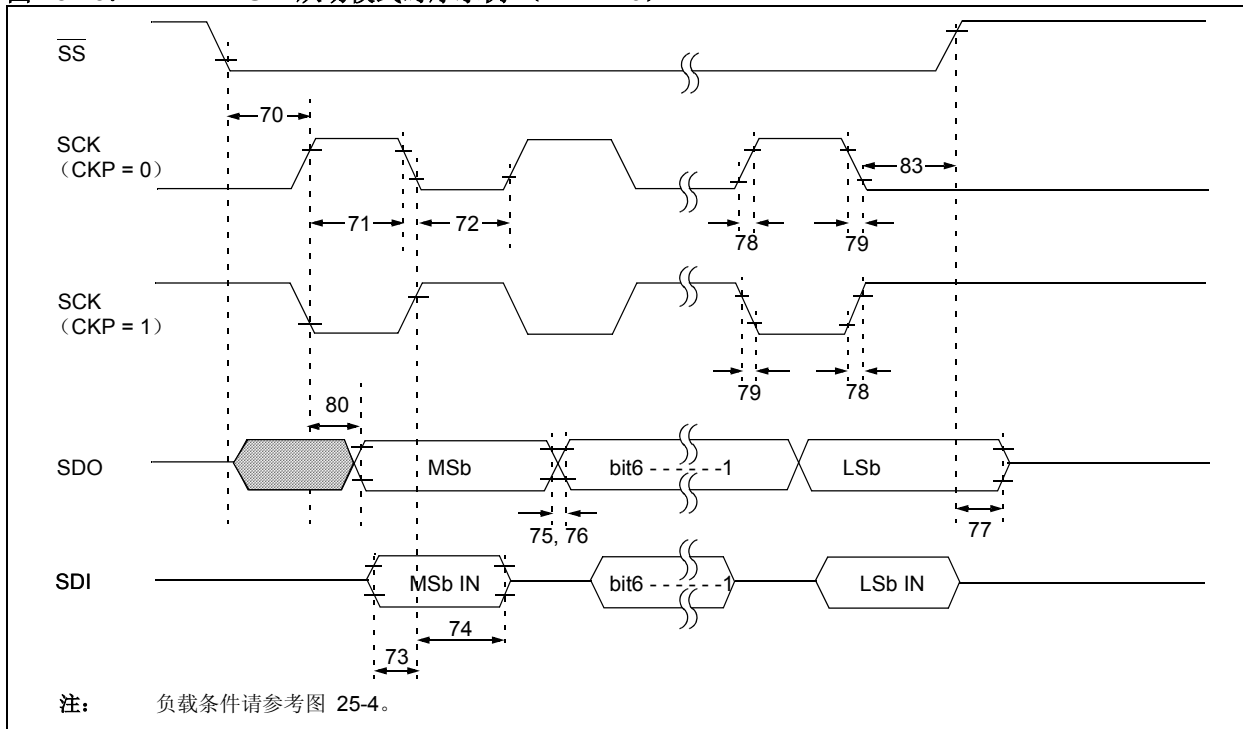


表 25-13: SPI 模式要求示例 (从动模式, CKE = 0)

参数号	符号	特性	最小值	最大值	单位	条件
70	Tssl2sch, Tssl2scl	\overline{SS} ↓ 到 SCK ↓ 或 SCK ↑ 输入	T _{CY}	—	ns	
71	TschH	SCK 输入高电平时间 (从动模式)	1.25 T _{CY} + 30	—	ns	
71A		连续 单字节	40	—	ns	(注 1)
72	TsclL	SCK 输入低电平时间 (从动模式)	1.25 T _{CY} + 30	—	ns	
72A		连续 单字节	40	—	ns	(注 1)
73	Tdiv2sch, Tdiv2scl	SDI 数据输入到 SCK 边沿的建立时间	100	—	ns	
73A	T _{B2B}	Byte1 最后一个时钟沿到 Byte2 第一个时钟沿	1.5 T _{CY} + 40	—	ns	(注 2)
74	Tsch2diL, Tscl2diL	SDI 数据输入到 SCK 边沿的保持时间	100	—	ns	
75	TdoR	SDO 数据输出上升时间	PIC18FXX31 — PIC18LFXX31	25 45	ns	
76	TdoF	SDO 数据输出下降时间	—	25	ns	
77	TssH2doZ	\overline{SS} ↑ 到 SDO 输出高阻态	10	50	ns	
78	TscR	SCK 输出上升时间 (主控模式)	PIC18FXX31 — PIC18LFXX31	25 45	ns	
79	TscF	SCK 输出下降时间 (主控模式)	—	25	ns	
80	Tsch2doV, Tscl2doV	SCK 边沿后 SDO 数据输出有效	PIC18FXX31 — PIC18LFXX31	50 100	ns	
83	Tsch2ssH, Tscl2ssH	SCK 边沿到 \overline{SS} ↑	1.5 T _{CY} + 40	—	ns	

注 1: 要求使用参数 73A。

注 2: 仅当使用参数 71A 和 72A 时。

PIC18F2331/2431/4331/4431

图 25-14: SPI 从动模式时序示例 (CKE = 1)

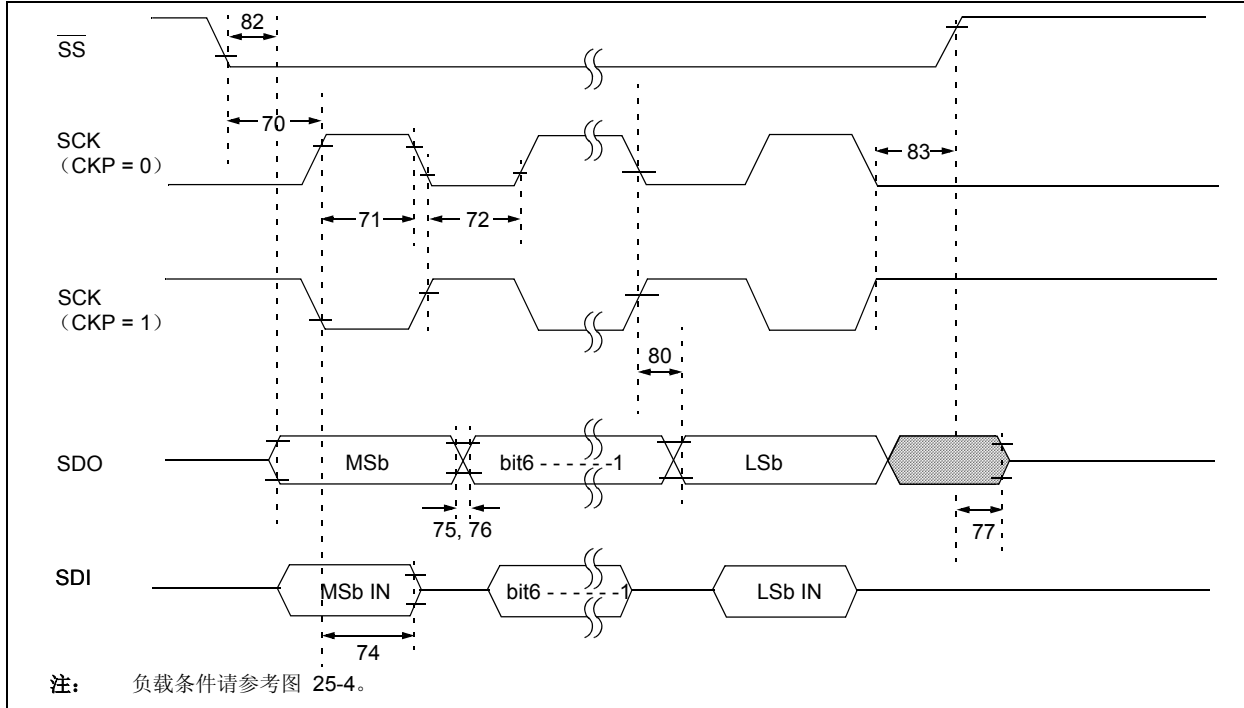


表 25-14: SPI 从动模式要求示例 (CKE = 1)

参数号	符号	特性	最小值	最大值	单位	条件
70	TssL2scH, TssL2scL	\overline{SS} ↓ 到 SCK ↓ 或 SCK ↑ 输入	Tcy	—	ns	
71	TscH	SCK 输入高电平时间	连续	1.25 Tcy + 30	—	ns
71A		(从动模式)	单字节	40	—	ns (注 1)
72	TscL	SCK 输入低电平时间	连续	1.25 Tcy + 30	—	ns
72A		(从动模式)	单字节	40	—	ns (注 1)
73A	Tb2B	Byte1 最后一个时钟沿到 Byte2 第一个时钟沿	1.5 Tcy + 40	—	ns	(注 2)
74	Tsch2diL, TscL2diL	SDI 数据输入到 SCK 边沿的保持时间	100	—	ns	
75	TdoR	SDO 数据输出上升时间	PIC18FXX31 —	25	ns	
			PIC18LFXX31	45	ns	
76	TdoF	SDO 数据输出下降时间	—	25	ns	
77	TssH2doZ	\overline{SS} ↑ 到 SDO 输出高阻态	10	50	ns	
78	TscR	SCK 输出上升时间	PIC18FXX31 —	25	ns	
		(主控模式)	PIC18LFXX31	45	ns	
79	TscF	SCK 输出下降时间 (主控模式)	—	25	ns	
80	Tsch2doV, TscL2doV	SCK 边沿后 SDO 数据输出有效	PIC18FXX31 —	50	ns	
			PIC18LFXX31	100	ns	
82	TssL2doV	在 \overline{SS} ↓ 边沿之后 SDO 数据输出有效	PIC18FXX31 —	50	ns	
			PIC18LFXX31	100	ns	
83	Tsch2ssH, TscL2ssH	SCK 边沿到 \overline{SS} ↑	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 25-15: I²C 总线启动 / 停止位时序

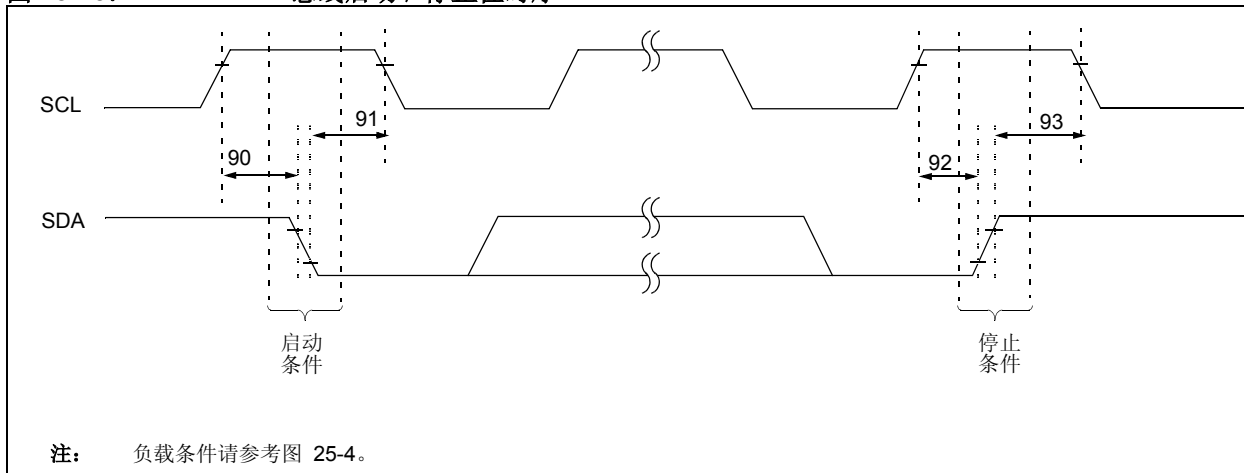
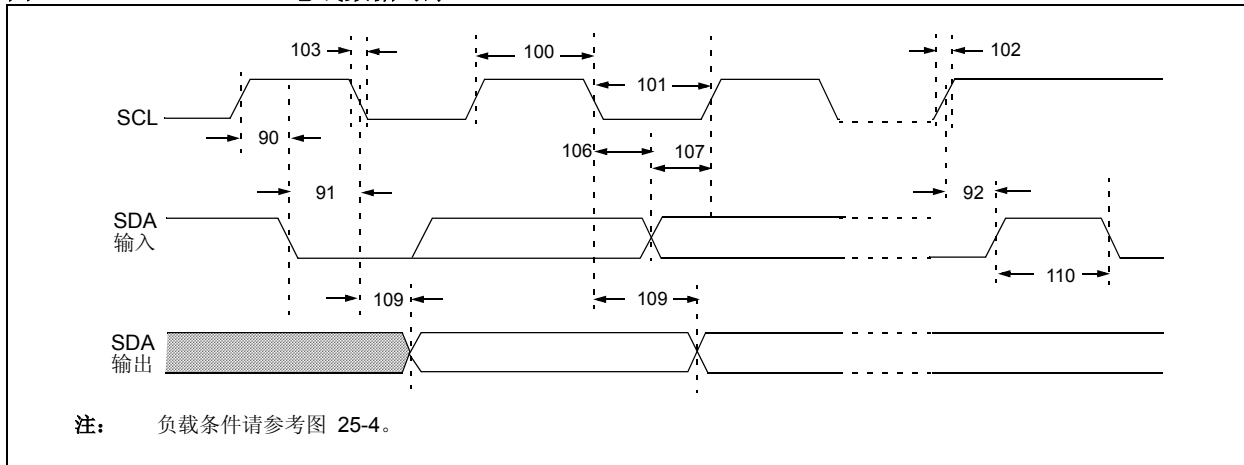


表 25-15: I²C 总线启动 / 停止位要求 (从动模式)

参数编号	符号	特性	最小值	最大值	单位	条件	
90	TSU:STA	启动条件建立时间	100 kHz 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 25-16: I²C 总线数据时序



PIC18F2331/2431/4331/4431

表 25-16: I²C 总线数据要求 (从动模式)

参数编号	符号	特性	最小值	最大值	单位	条件	
100	THIGH	时钟高电平时间	100 kHz 模式	4.0	—	μs	PIC18FXX31 的工作频率至少为 1.5 MHz
			400 kHz 模式	0.6	—	μs	PIC18FXX31 的工作频率至少为 10 MHz
			SSP 模块	1.5 T _{CY}	—		
101	TLOW	时钟低电平时间	100 kHz 模式	4.7	—	μs	PIC18FXX31 的工作频率至少为 1.5 MHz
			400 kHz 模式	1.3	—	μs	PIC18FXX31 的工作频率至少为 10 MHz
			SSP 模块	1.5 T _{CY}	—		
102	TR	SDA 和 SCL 上升时间	100 kHz 模式	—	1000	ns	
			400 kHz 模式	20 + 0.1 C _B	300	ns	规定 C _B 值为 10 -400 pF
103	TF	SDA 和 SCL 下降时间	100 kHz 模式	—	300	ns	
			400 kHz 模式	20 + 0.1 C _B	300	ns	规定 C _B 值为 10 -400 pF
90	TSU:STA	启动条件建立时间	100 kHz 模式	4.7	—	μs	仅与重复启动条件相关
			400 kHz 模式	0.6	—	μs	
91	THD:STA	启动条件保持时间	100 kHz 模式	4.0	—	μs	这个周期后, 产生第一个时钟脉冲
			400 kHz 模式	0.6	—	μs	
106	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns	
			400 kHz 模式	0	0.9	μs	
107	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns	(注 2)
			400 kHz 模式	100	—	ns	
92	TSU:STO	停止条件建立时间	100 kHz 模式	4.7	—	μs	
			400 kHz 模式	0.6	—	μs	
109	TAA	时钟输出有效时间	100 kHz 模式	—	3500	ns	(注 1)
			400 kHz 模式	—	—	ns	
110	TBUF	总线空闲时间	100 kHz 模式	4.7	—	μs	在新的传输开始之前, 总线必须保持空闲的时间
			400 kHz 模式	1.3	—	μs	
D102	C _B	总线的容性负载	—	400	pF		

- 注 1: 作为发送器, 为避免意外产生启动或停止条件, 器件必须提供此内部最小延迟时间, 以补偿 SCL 下降沿的不确定区域 (最小值 300ns)。
- 注 2: 在标准模式的 I²C 总线系统中可以使用快速模式的 I²C 总线器件, 但必须满足 TSU:DAT ≥ 250 ns 的要求。如果器件没有延长 SCL 信号的低电平周期, 将自动满足此条件。如果器件延长了 SCL 信号的低电平周期, 则下一个数据位必须输出到 SDA 线。
SCL 线释放前, TR max. + TSU:DAT = 1000 + 250 = 1250 ns (根据标准模式 I²C 总线规范)。

PIC18F2331/2431/4331/4431

图 25-17: SSP I²C 总线启动 / 停止位时序波形

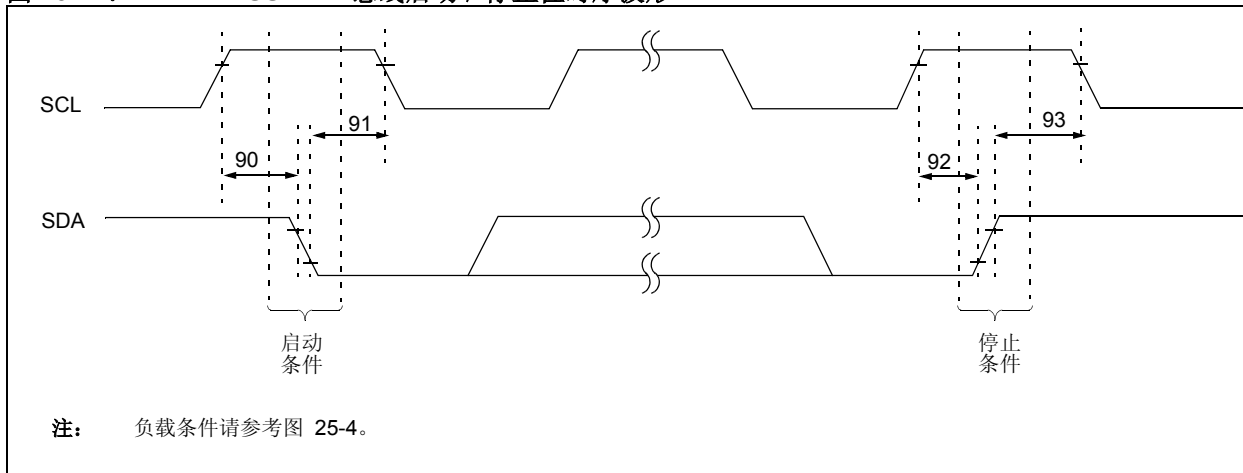
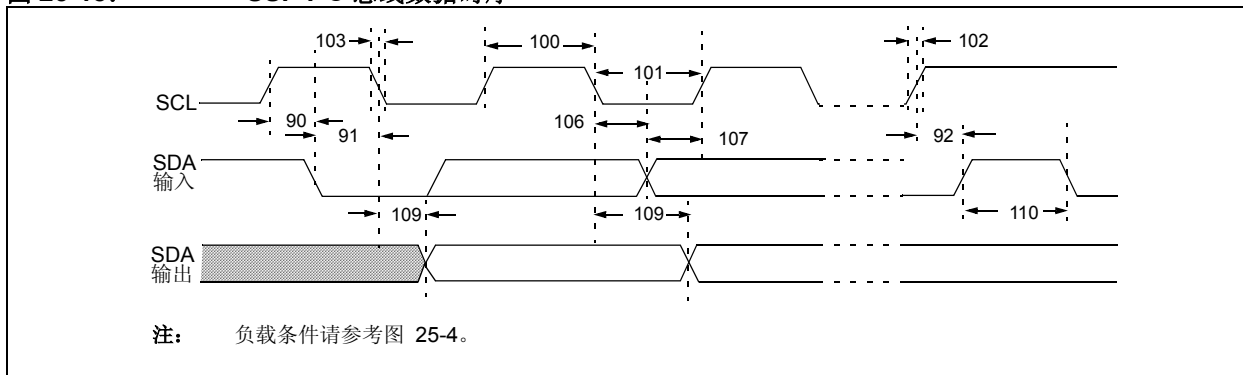


表 25-17: SSP I²C 总线启动 / 停止位要求

参数编号	符号	特性	最小值	最大值	单位	条件	
90	TSU:STA	启动条件建立时间	100 kHz 模式	$2(T_{osc})(BRG + 1)$	—	ns	仅与重复启动条件相关
			400 kHz 模式	$2(T_{osc})(BRG + 1)$	—		
			1 MHz 模式 (1)	$2(T_{osc})(BRG + 1)$	—		
91	THD:STA	启动条件保持时间	100 kHz 模式	$2(T_{osc})(BRG + 1)$	—	ns	这个周期后，产生第一个时钟脉冲
			400 kHz 模式	$2(T_{osc})(BRG + 1)$	—		
			1 MHz 模式 (1)	$2(T_{osc})(BRG + 1)$	—		
92	TSU:STO	停止条件建立时间	100 kHz 模式	$2(T_{osc})(BRG + 1)$	—	ns	
			400 kHz 模式	$2(T_{osc})(BRG + 1)$	—		
			1 MHz 模式 (1)	$2(T_{osc})(BRG + 1)$	—		
93	THD:STO	停止条件保持时间	100 kHz 模式	$2(T_{osc})(BRG + 1)$	—	ns	
			400 kHz 模式	$2(T_{osc})(BRG + 1)$	—		
			1 MHz 模式 (1)	$2(T_{osc})(BRG + 1)$	—		

注 1: 对于所有 I²C 引脚，最小引脚电容为 10 pF。

图 25-18: SSP I²C 总线数据时序



PIC18F2331/2431/4331/4431

表 25-18: SSP I²C 总线数据要求

参数号	符号	特性	最小值	最大值	单位	条件
100	THIGH	时钟高电平时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			400 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			1 MHz 模式 (1)	2(Tosc)(BRG + 1)	—	ms
101	TLOW	时钟低电平时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			400 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			1 MHz 模式 (1)	2(Tosc)(BRG + 1)	—	ms
102	TR	SDA 和 SCL 上升时间	100 kHz 模式	—	1000	ns
			400 kHz 模式	20 + 0.1 C _B	300	ns
			1 MHz 模式 (1)	—	300	ns
103	TF	SDA 和 SCL 下降时间	100 kHz 模式	—	300	ns
			400 kHz 模式	20 + 0.1 C _B	300	ns
			1 MHz 模式 (1)	—	100	ns
90	TSU:STA	启动条件建立时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			400 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			1 MHz 模式 (1)	2(Tosc)(BRG + 1)	—	ms
91	THD:STA	启动条件保持时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			400 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			1 MHz 模式 (1)	2(Tosc)(BRG + 1)	—	ms
106	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns
			400 kHz 模式	0	0.9	ms
			1 MHz 模式 (1)	TBD	—	ns
107	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns
			400 kHz 模式	100	—	ns
			1 MHz 模式 (1)	TBD	—	ns
92	TSU:STO	停止条件建立时间	100 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			400 kHz 模式	2(Tosc)(BRG + 1)	—	ms
			1 MHz 模式 (1)	2(Tosc)(BRG + 1)	—	ms
109	TAA	时钟输出有效时间	100 kHz 模式	—	3500	ns
			400 kHz 模式	—	1000	ns
			1 MHz 模式 (1)	—	—	ns
110	TBUF	总线空闲时间	100 kHz 模式	4.7	—	ms
			400 kHz 模式	1.3	—	ms
			1 MHz 模式 (1)	TBD	—	ms
D102	CB	总线的容性负载	—	400	pF	

注 1: 对于所有 I²C 引脚, 最小引脚电容为 10 pF。

2: 在标准模式的 I²C 总线系统中可以使用快速模式的 I²C 总线器件, 但必须满足参数 107 ≥ 250 ns 的要求。如果器件没有延长 SCL 信号的低电平周期, 将自动满足此条件。如果该器件延长 SCL 信号的低电平周期, 它必须将下一个数据位输出到 SDA 线; SCL 线释放前, 参数 102 + 参数 107 = 1000 + 250 = 1250 ns (100 kHz 模式)。

PIC18F2331/2431/4331/4431

图 25-19: USART 同步发送 (主控 / 从动) 时序

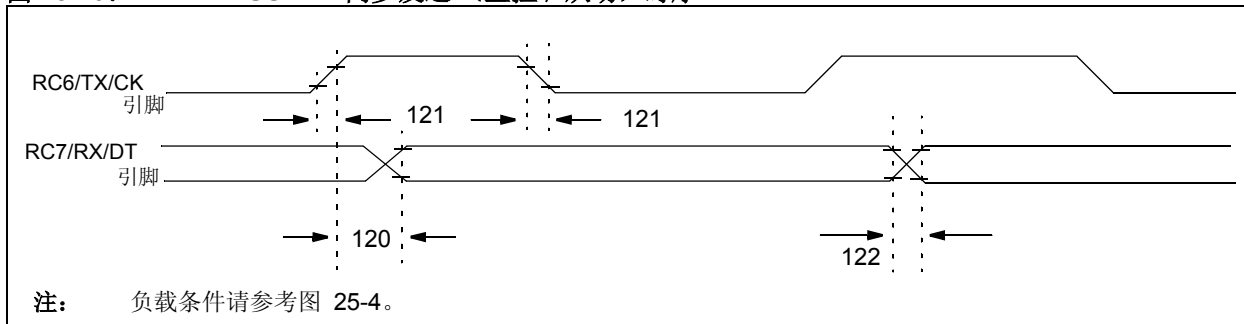


表 25-19: USART 同步发送要求

参数号	符号	特性	最小值	最大值	单位	条件
120	TckH2dtV	同步发送 (主控和从动) 时钟高电平到数据输出有效	PIC18FXX31	—	40	ns
			PIC18LFX31	—	100	ns
121	Tckrf	时钟输出上升时间和下降时间 (主控模式)	PIC18FXX31	—	20	ns
			PIC18LFX31	—	50	ns
122	TdtV	时钟输出上升时间和下降时间	PIC18FXX31	—	20	ns
			PIC18LFX31	—	50	ns

图 25-20: USART 同步接收 (主控 / 从动) 时序

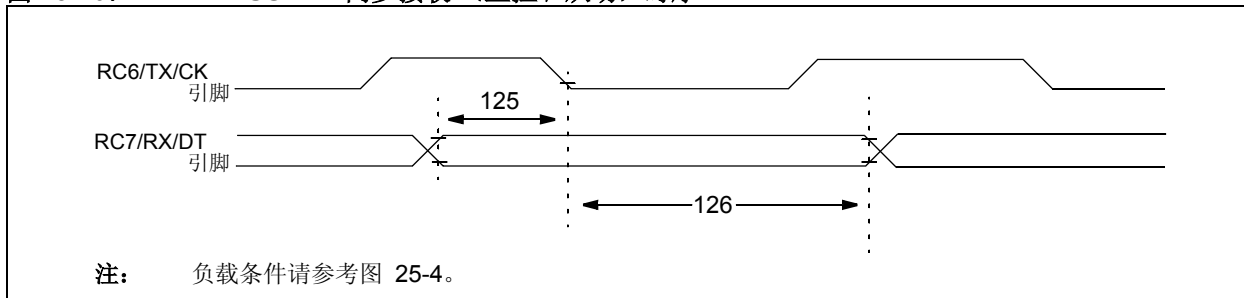


表 25-20: USART 同步接收要求

参数号	符号	特性	最小值	最大值	单位	条件
125	TdtV2ckl	同步接收 (主控和从动) 在 CK ↓ 之前数据保持 (DT 保持时间)	10	—	ns	
126	TckL2dtl	在 CK ↓ 之后数据保持 (DT 保持时间)	15	—	ns	

PIC18F2331/2431/4331/4431

表 25-21: A/D 转换器特性: PIC18F2331/2431/4331/4431 (工业级)
PIC18LF2331/2431/4331/4431 (工业级)

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
器件供电							
	AVDD	模拟 VDD	VDD-0.3	—	VDD+0.3	V	
	AVSS	模拟 Vss	Vss-0.3		Vss+0.3	V	
	IAD	模块电流 (转换过程中)		500 250		μA μA	VDD = 5V VDD = 2.5V
	IADO	模块关断电流			1.0	μA	
AC 时序参数							
A10	FTHR	吞吐率		— —	200 75	ksps ksps	VDD=5V, 单通道 VDD < 3V, 单通道
A11	TAD	A/D 时钟周期	385 1000		20,000 20,000	ns	VDD = 5V VDD = 3V
A12	TRC	A/D 内部 RC 振荡器周期		500 750 10000	1500 2250 20000	ns ns ns	PIC18F 器件 PIC18LF 器件 AVDD < 3.0V
A13	TCNV	转换时间 ⁽¹⁾	12	12	12	TAD	
A14	TACQ	采集时间 ⁽²⁾	2 ⁽²⁾			TAD	
A16	TTC	转换从外部开始	1/4 Tcy		1Tcy		
参考电压输入							
A20	VREF	10 位分辨率的参考电压 (VREF+ - VREF-)	1.5 1.8	— —	AVDD-AVSS AVDD-AVSS	V V	VDD ≥ 3V VDD < 3V
A21	VREFH	参考电压高电平 (AVDD 或 VREF+)	1.5V	—	AVDD	V	VDD ≥ 3V
A22	VREFL	参考电压低电平 (AVSS 或 VREF-)	AVSS	—	VREFH-1.5V	V	
A23	IREF	参考电流		150μA 75μA			VDD = 5V VDD = 2.5V
模拟输入特性							
A26	VAIN	输入电压 ⁽³⁾	AVSS-0.3	—	AVDD+0.3	V	
A30	ZAIN	模拟电压源的建议阻抗	—	—	2.5	kΩ	
A31	ZCHIN	模拟通道输入阻抗	—		10.0	kΩ	VDD = 3.0 V
DC 性能							
A41	NR	分辨率	10 位			—	
A42	EIL	积分非线性	—	—	< ±1	LSb	VDD ≥ 3.0V VREFH ≥ 3.0V
A43	EIL	微分非线性	—	—	< ±1	LSb	VDD ≥ 3.0V VREFH ≥ 3.0V
A45	E0FF	偏移误差	—	±0.5	< ±1.5	LSb	VDD ≥ 3.0V VREFH ≥ 3.0V
A46	E0A	增益误差	—	±0.5	< ±1.5	LSb	VDD ≥ 3.0V VREFH ≥ 3.0V
A47	—	单调性 ⁽⁴⁾	保证			—	VDD ≥ 3.0V VREFH ≥ 3.0V

- 注 1: 转换时间不包括采集时间。有关采集时间要求的全面讨论, 请参阅第 20.0 节“10 位高速模数转换器 (A/D) 模块”。
- 2: Tacq 在连续模式中应该为 12Tad 或更大。
- 3: 由于 VDD 小于 2.7V 且温度低于 0°C, 应该把 VAIN 限制在小于 VDD/2 的范围内。
- 4: A/D 转换结果不会随着输入电压的增加而减小, 也不会丢失数字码。

26.0 直流和交流特性图表初稿

目前没有图表。

PIC18F2331/2431/4331/4431

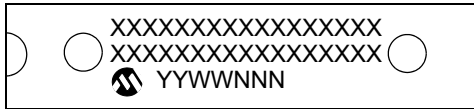
注:

PIC18F2331/2431/4331/4431

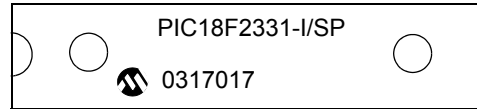
27.0 封装信息

27.1 封装标识信息

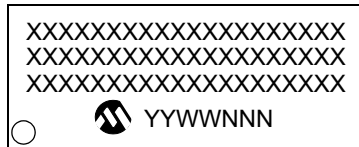
28 引脚 PDIP (小型 DIP)



示例



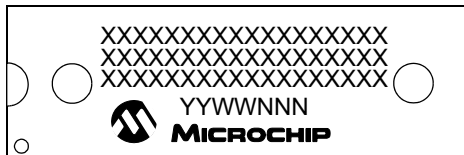
28 引脚 SOIC



示例



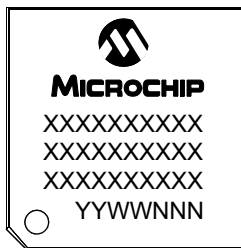
40 引脚 PDIP



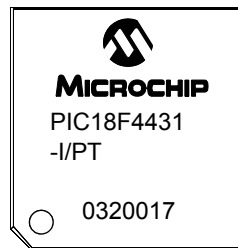
示例



44 引脚 TQFP



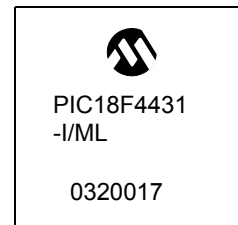
示例



44 引脚 QFN



示例



图注: XX...X 客户特定信息 *
 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

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特定信息的字符数。

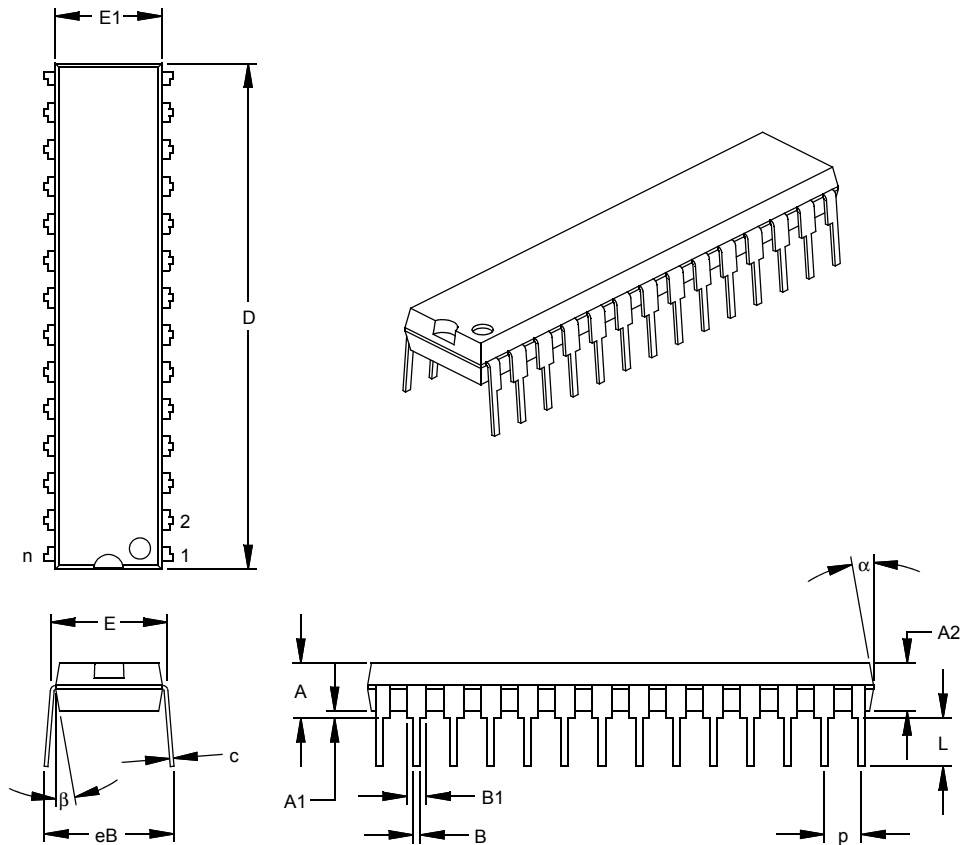
* 标准 PICmicro 器件标识由 Microchip 元器件编号、年份代码、星期代码和追踪代码组成。若 PICmicro 器件标识超出上述内容, 需支付一定的附加费用。请向当地的 Microchip 销售办事处了解确认。对于 QTP 器件, 任何特殊标记的费用都已包含在 QTP 价格中。

PIC18F2331/2431/4331/4431

27.2 封装详细信息

以下部分将介绍封装的技术细节。

28 引脚塑料窄条双列直插式封装 (SP) —— 300 mil (PDIP)



尺寸范围	单位	英寸*			毫米		
		最小	正常	最大	最小	正常	最大
引脚数	n		28			28	
引脚间距	p		.100			2.54	
顶端到固定面高度	A	.140	.150	.160	3.56	3.81	4.06
塑模封装厚度	A2	.125	.130	.135	3.18	3.30	3.43
塑封底端到固定面高度	A1	.015			0.38		
肩到肩宽度	E	.300	.310	.325	7.62	7.87	8.26
塑模封装宽度	E1	.275	.285	.295	6.99	7.24	7.49
总长度	D	1.345	1.365	1.385	34.16	34.67	35.18
引脚尖到固定面高度	L	.125	.130	.135	3.18	3.30	3.43
引脚厚度	c	.008	.012	.015	0.20	0.29	0.38
引脚上部宽度	B1	.040	.053	.065	1.02	1.33	1.65
引脚下部宽度	B	.016	.019	.022	0.41	0.48	0.56
总排列间距	§ eB	.320	.350	.430	8.13	8.89	10.92
塑模顶部锥度	α	5	10	15	5	10	15
塑模底部锥度	β	5	10	15	5	10	15

* 控制参数

§ 重要特性

注

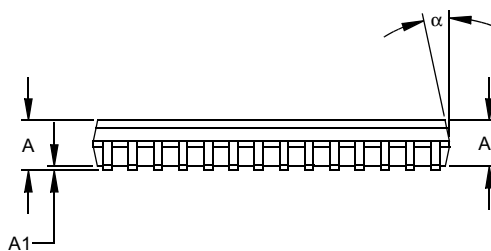
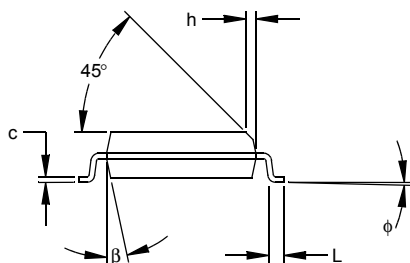
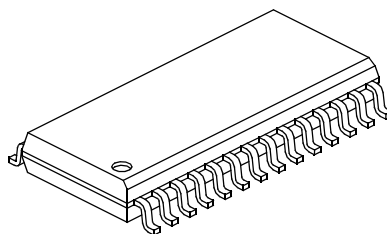
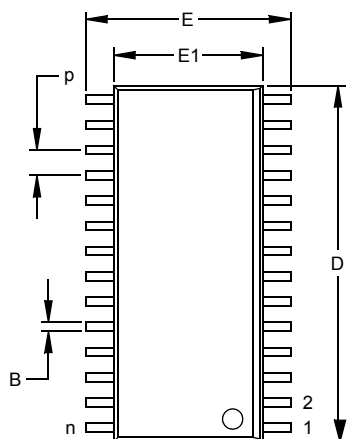
尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 .010" (0.254mm)。

等同于 JEDEC 号 MO-095

图号 C04-070

PIC18F2331/2431/4331/4431

28 引脚塑料小型封装 (SO) —— 宽条, 300 mil (SOIC)



尺寸范围	单位	英寸*			毫米		
		最小	正常	最大	最小	正常	最大
引脚数	n		28			28	
引脚间距	p		.050			1.27	
总高度	A	.093	.099	.104	2.36	2.50	2.64
塑模封装厚度	A2	.088	.091	.094	2.24	2.31	2.39
悬空间隙 §	A1	.004	.008	.012	0.10	0.20	0.30
总宽度	E	.394	.407	.420	10.01	10.34	10.67
塑模封装宽度	E1	.288	.295	.299	7.32	7.49	7.59
总长度	D	.695	.704	.712	17.65	17.87	18.08
斜面投影距离	h	.010	.020	.029	0.25	0.50	0.74
底脚长度	L	.016	.033	.050	0.41	0.84	1.27
底脚向上倾斜角	φ	0	4	8	0	4	8
引脚厚度	c	.009	.011	.013	0.23	0.28	0.33
引脚宽度	B	.014	.017	.020	0.36	0.42	0.51
塑模顶部锥度	α	0	12	15	0	12	15
塑模底部锥度	β	0	12	15	0	12	15

* 控制参数
§ 重要特性

注

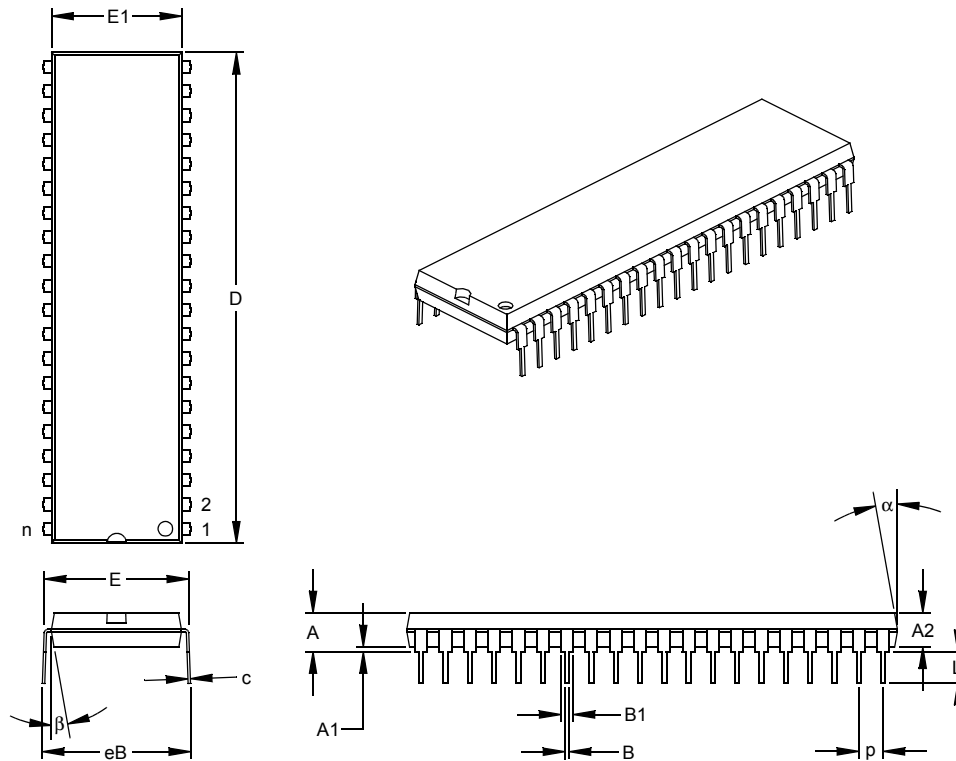
尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 .010" (0.254mm)。

等同于 JEDEC 号: MS-013

图号 C04-052

PIC18F2331/2431/4331/4431

40 引脚塑料双列直插式封装 (P) —— 600 mil (PDIP)



尺寸范围	单位	英寸*			毫米		
		最小	正常	最大	最小	正常	最大
引脚数	n		40			40	
引脚间距	p		.100			2.54	
顶端到固定面高度	A	.160	.175	.190	4.06	4.45	4.83
塑模封装厚度	A2	.140	.150	.160	3.56	3.81	4.06
塑封底端到固定面高度	A1	.015			0.38		
肩到肩宽度	E	.595	.600	.625	15.11	15.24	15.88
塑模封装宽度	E1	.530	.545	.560	13.46	13.84	14.22
总长度	D	2.045	2.058	2.065	51.94	52.26	52.45
引脚尖到固定面高度	L	.120	.130	.135	3.05	3.30	3.43
引脚厚度	c	.008	.012	.015	0.20	0.29	0.38
引脚上部宽度	B1	.030	.050	.070	0.76	1.27	1.78
引脚下部宽度	B	.014	.018	.022	0.36	0.46	0.56
总排列间距	§ eB	.620	.650	.680	15.75	16.51	17.27
塑模顶部锥度	α	5	10	15	5	10	15
塑模底部锥度	β	5	10	15	5	10	15

* 控制参数

§ 重要特性

注

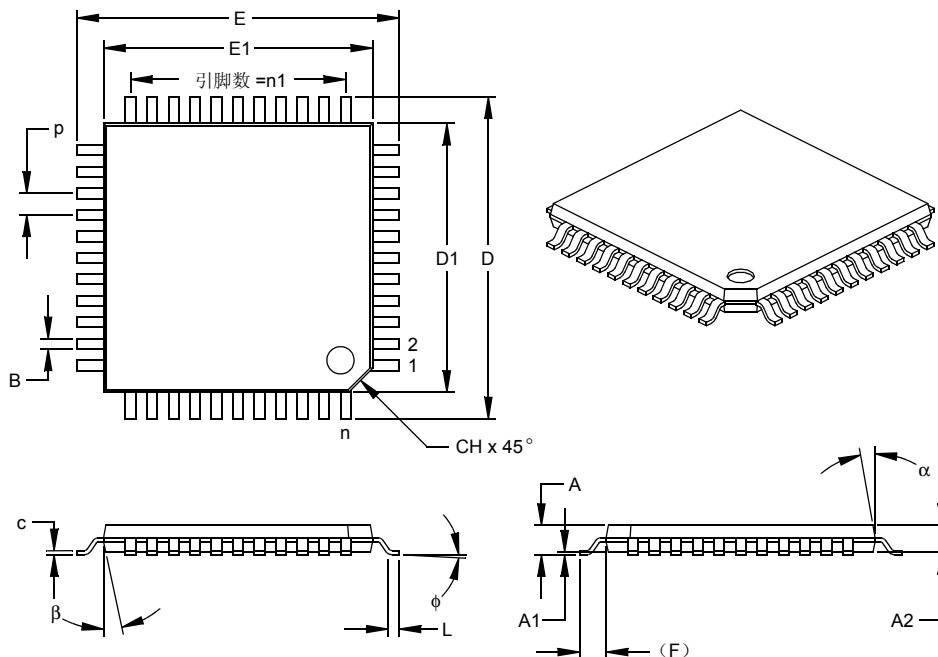
尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 .010" (0.254mm)。

等同于 JEDEC 号 MO-011

图号 C04-016

PIC18F2331/2431/4331/4431

44 引脚塑料薄型四方扁平封装 (PT) —— 主体 10x10x1 mm, 引脚形式 1.0/0.10 mm (TQFP)



	单位	英寸			毫米*		
		尺寸范围	最小	正常	最大	最小	正常
引脚数	n		44			44	
引脚间距	p		.031			0.80	
每侧引脚数	n1		11			11	
总高度	A	.039	.043	.047	1.00	1.10	1.20
塑模封装厚度	A2	.037	.039	.041	0.95	1.00	1.05
悬空间隙 §	A1	.002	.004	.006	0.05	0.10	0.15
底脚长度	L	.018	.024	.030	0.45	0.60	0.75
底脚占位 (参考)	(F)		.039		1.00		
底脚倾斜角	φ	0	3.5	7	0	3.5	7
总宽度	E	.463	.472	.482	11.75	12.00	12.25
总长度	D	.463	.472	.482	11.75	12.00	12.25
塑模封装宽度	E1	.390	.394	.398	9.90	10.00	10.10
塑模封装长度	D1	.390	.394	.398	9.90	10.00	10.10
引脚厚度	c	.004	.006	.008	0.09	0.15	0.20
引脚宽度	B	.012	.015	.017	0.30	0.38	0.44
引脚 1 处角斜面	CH	.025	.035	.045	0.64	0.89	1.14
塑模顶部锥度	α	5	10	15	5	10	15
塑封底部锥度	β	5	10	15	5	10	15

* 控制参数
§ 重要特性

注

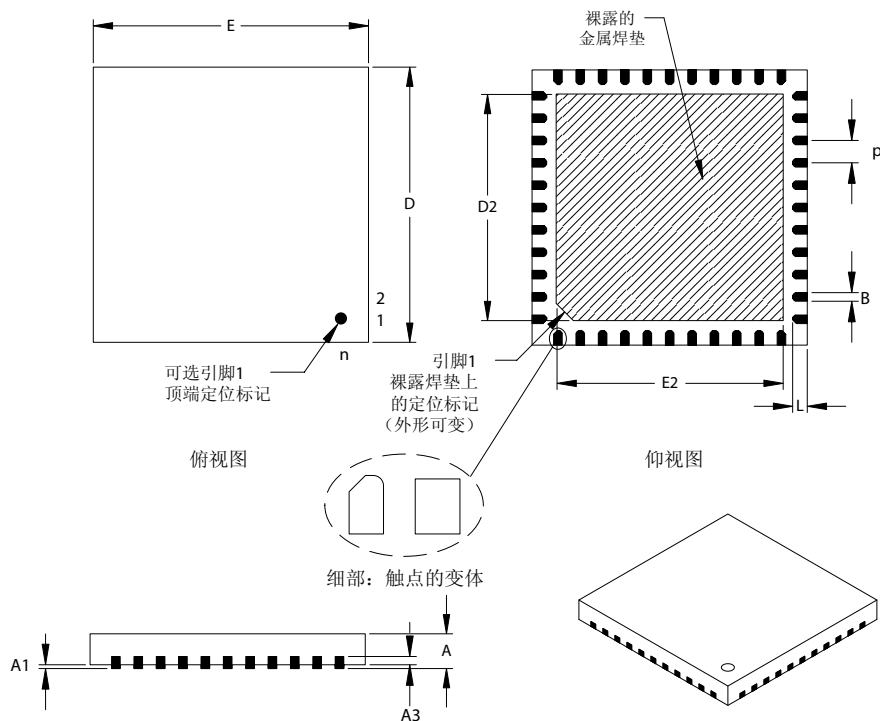
尺寸 D1 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 .010" (0.254mm)。

等同于 JEDEC 号: MS-026

图号 C04-076

PIC18F2331/2431/4331/4431

44 引脚塑料四方扁平无引线封装 (ML) —— 主体 8x8 mm (QFN)



尺寸范围	单位	英寸			毫米*		
		最小	正常	最大	最小	正常	最大
触点数	n		44			44	
触点间距	P	.026 BSC ¹			0.65 BSC ¹		
总高度	A	.031	.035	.039	0.80	0.90	1.00
悬空间隙	A1	.000	.001	.002	0	0.02	0.05
基座厚度	A3	.010 REF ²			0.25 REF ²		
总宽度	E	.309	.315	.321	7.85	8.00	8.15
裸露的焊垫宽度	E2	.236	.258	.260	5.99	6.55	6.60
总长度	D	.309	.315	.321	7.85	8.00	8.15
裸露的焊垫长度	D2	.236	.258	.260	5.99	6.55	6.60
触点宽度	B	.008	.013	.013	0.20	0.33	0.35
触点长度	L	.014	.016	.019	0.35	0.40	0.48

*控制参数

注:

1. BSC: 基本尺寸。理论上显示的是没有公差精确值。

参见ASME Y14.5M

2. REF: 参考尺寸, 通常无公差, 仅供参考。

参见ASME Y14.5M

裸露的焊垫尺寸随管芯叶片大小而变化。

等同于JEDEC号: MO-220

图号C04-103, 修订于05-05-05

PIC18F2331/2431/4331/4431

附录 A： 版本历史

版本 A（2003 年 7 月）

PIC18F2331/2431/4331/4431 器件的原始数据手册。

版本 B（2003 年 12 月）

更新了第 25.0 节“电气特性”中的电气规范，并且对数据手册文本稍作修正。

附录 B： 器件比较

表 B-1 所示为本数据手册所列器件的比较。

表 B-1： 器件比较

功能部件	PIC18F2331	PIC18F2431	PIC18F4331	PIC18F4431
程序存储器（字节）	4096	8192	4096	8192
程序存储器（指令）	2048	4096	2048	4096
中断源	22	22	34	34
I/O 端口	端口 A, B, C, D, E	端口 A, B, C, D, E	端口 A, B, C, D, E	端口 A, B, C, D, E
捕捉 / 比较 / PWM 模块	2	2	2	2
增强型捕捉 / 比较 / PWM 模块	1	1	1	1
并行通讯（PSP）	无	无	有	有
10 位模数转换模块	5 个输入通道	5 个输入通道	9 个输入通道	9 个输入通道
封装	28 引脚 SDIP 28 引脚 SOIC	28 引脚 SDIP 28 引脚 SOIC	40 引脚 DIP 44 引脚 TQFP 44 引脚 QFN	40 引脚 DIP 44 引脚 TQFP 44 引脚 QFN

PIC18F2331/2431/4331/4431

附录 C: 转换注意事项

本附录讲述了从器件的老版本升级至此数据手册中所列版本时需要注意的事项。这些变化通常是由于所采用的工艺技术不同而引起的。比方说从 PIC16C74A 转换为 PIC16C74B。

不适用

附录 D: 从低档器件移植到增强型器件

本节论述如何从低档器件（如 PIC16C5X）移植到增强型单片机（如 PIC18FXXX）。

下表列出了本系列器件与 PIC16C5X 单片机系列相比，所做的修改：

目前未提供

附录 E: 从中档器件移植到增强型器件

在 AN716 (“Migrating Designs from PIC16C74A/74B to PIC18F442”) 中详细讲述了中档单片机 (例如 PIC16CXXX) 与增强型器件 (例如 PIC18FXXX) 之间的区别。虽然所讨论的变化都是针对特定器件的, 但是通常适用于所有从中档器件到增强型器件的移植。

此应用笔记也可以从 Microchip 的网站获取, 网址是: www.Microchip.com。

附录 F: 从高档器件移植到增强型器件

在 AN726 (“PIC17CXXX to PIC18FXXX Migration”) 中详细讲述了高档单片机 (例如 PIC17CXXX) 如何移植到增强型器件 (例如 PIC18FXXX), 以及它们之间的区别。

此应用笔记也可以从 Microchip 的网站获取, 网址是: www.Microchip.com。

PIC18F2331/2431/4331/4431

注:

PIC18F2331/2431/4331/4431

索引

A

A/D	243
计算所需要的最小采集时间	254
特殊事件触发器 (CCP)	154
相关寄存器	259
ACK 脉冲	217, 218
AC (时序) 特性	353
参数符号	353
器件时序规范的负载条件	354
时序条件	354
温度和电压规范	354
ADDLW	293
ADDWF	293
ADDWFC	294
ANDLW	294
ANDWF	295

B

BC	295
BCF	296
BF 位	212
BN	296
BNC	297
BNN	297
BNOV	298
BNZ	298
BOR. 参见欠压复位。	
BOV	301
BRA	299
BSF	299
BTFSC	300
BTFSS	300
BTG	301
BZ	302
版本历史	379

比较 (CCP 模块)

CCPR1 寄存器	154
CCP 引脚配置	154
软件中断	154
Timer1 模式选择	154
特殊事件触发器	154
相关寄存器	155

编程, 器件指令

表指针操作 (表)

不同情况下的延时 (表)

捕捉 / 比较 / PWM (CCP)

 比较模式. 参见比较。

 捕捉模式. 参见捕捉。

 CCP1

 CCPR1H 寄存器

 CCPR1L 寄存器

 CCP2

 CCPR2H 寄存器

 CCPR2L 寄存器

 定时器资源

 PWM 模式. 参见 PWM。

捕捉 (CCP 模块)

 CCPR1H:CCPR1L 寄存器

 CCP 引脚配置

 软件中断

 Timer1 模式选择

 相关寄存器

C

CALL

C 编译器

 MPLAB C17

 MPLAB C18

 MPLAB C30

CKE 位

CKP 位

CLRF

CLRWDT

COMF

CPFSEQ

CPFSGT

CPFSLT

CPU 的特殊功能

 操作码字段描述

 查找表

 程序存储器

 复位向量

 PIC18F2331/4331 的映射图和堆栈

 PIC18F2431/4431 的映射图和堆栈

 中断向量

 程序存储器代码保护

 程序存储器中的指令

 双字指令

 程序计数器

 PCLATH 寄存器

 PCLATU 寄存器

 PCL 寄存器

 程序校验

 程序校验和代码保护相关寄存器

 串行时钟 (SCK) 引脚

 串行数据输出 (SDO) 引脚

 串行数据输入 (SDI) 引脚

 从低档器件迁移到增强型器件

 从动选择 (SS) 引脚

 从高档器件迁移到增强型器件

 从中档器件迁移到增强型器件

 存储器编程要求

 存储器构成

 程序存储器

 数据存储器

 存储区选择寄存器 (BSR)

 D/A 位

 DAW

 DCFSNZ

 DC 和 AC 特性

 图表 (初稿)

 DC 特性

 DECf

 DECfSZ

 代码保护

 代码示例

 16 × 16 无符号乘法程序

 16 × 16 有符号乘法程序

 8 × 8 无符号乘法程序

 8 × 8 有符号乘法程序

 擦除闪存程序存储器的一行

 初始化 PORTA

 初始化 PORTB

 初始化 PORTC

 初始化 PORTD

 初始化 PORTE

 初始化 PORTF

 初始化 PORTG

 初始化 PORTH

 初始化 PORTI

 初始化 PORTJ

 初始化 PORTK

 初始化 PORTL

 初始化 PORTM

 初始化 PORTN

 初始化 PORTO

 初始化 PORTP

 初始化 PORTQ

 初始化 PORTR

 初始化 PORTS

 初始化 PORTT

 初始化 PORTU

 初始化 PORTV

 初始化 PORTW

 初始化 PORTX

 初始化 PORTY

 初始化 PORTZ

PIC18F2331/2431/4331/4431

初始化 PORTC	118
初始化 PORTD	124
初始化 PORTE	129
读闪存程序存储器的一个字	79
计算 GOTO 使用偏移值	63
快速寄存器堆栈	60
使用间接寻址将 RAM (存储区 1) 清零的方法	71
使用 Timer1 中断服务程序实现实时时钟	141
数据 EEPROM 读取	87
数据 EEPROM 刷新程序	88
数据 EEPROM 写入	87
写入闪存程序存储器	82-83
在捕捉预分频器之间进行切换	153
在 RAM 内保存 Status、WREG 和 BSR 寄存器	106
低压 ICSP 编程	286
低压检测	261
复位的影响	265
工作原理	264
参考电压设置点	265
电流消耗	265
特性	352
在休眠模式下工作	265
电气特性	337
读表 / 写表	63
堆栈满 / 下溢复位	59

F

FSCM. 参见故障保护时钟监视器。	
返回地址堆栈	58
返回堆栈指针 (STKPTR)	58
封装信息	373
标识	373
复位	45, 267

G

GOTO	308
更新地址位, UA	212
功耗管理模式	31
唤醒	40
进入	32
空闲模式	33
休眠模式	33
选择	31
运行模式	38
综述 (表)	31

公式

16 × 16 无符号乘法算法	90
16 × 16 有符号乘法算法	90
A/D 采集时间	253
A/D 最小充电时间	253
固件指令	287
故障保护时钟监视器	267, 281
功耗管理模式下的中断	282
POR 或从休眠中唤醒	282
振荡器故障期间的 WDT	281

H

HSPLL	22
后分频器, WDT	
比率选择 (T0PS2:T0PS0 位)	135
分配 (PSA 位)	135
汇编器	
MPASM 汇编器	331

I

I/O 端口	107
I ² C 模式	
从动模式	
SCL 和 SDA 引脚	217
多主模式	220
发送	219
工作原理	217
接收	218
模式选择	217
相关寄存器	220
寻址	218
主控模式	220
ID 单元	267, 286
INCF	308
INCFSZ	309
INFSNZ	309
INTCON 寄存器	93
RBIF 位	112
INTOSC 频率漂移	42
INTOSC, INTRC. 参见内部振荡电路。	
IORLW	310
IORWF	310
IPR 寄存器	102

J

寄存器	
CONFIG1H (配置寄存器 1 高位)	268
CONFIG2H (配置寄存器 2 高位)	270, 269
CONFIG2L (配置寄存器 2 低位)	269
CONFIG3H (配置寄存器 3 高位)	272
CONFIG4L (配置寄存器 4 低位)	273
CONFIG5H (配置寄存器 5 高位)	274
CONFIG6H (配置寄存器 6 高位)	275
CONFIG6L (配置寄存器 6 低位)	275
CONFIG7H (配置寄存器 7 高位)	276
CONFIG7L (配置寄存器 7 低位)	275
EECON1 (数据 EEPROM 控制 1)	77, 86
INTCON (中断控制)	93
INTCON2 (中断控制 2)	94
INTCON3 (中断控制 3)	95
IPR1 (外设中断优先级 1)	102
IPR2 (外设中断优先级 2)	103
LVDCON (LVD 控制)	263
OSCCON (振荡器控制)	28
器件 ID 寄存器 1	277
器件 ID 寄存器 2	277
PIE1 (外设中断使能 1)	99
PIE2 (外设中断使能 2)	100
PIR1 (外设中断请求 (标志) 1)	96
PIR2 (外设中断请求 (标志) 2)	97
RCON (复位控制)	74, 105
RCSTA (接收状态和控制)	223
SSPCON (同步串行口控制) 寄存器	213
SSPSTAT (同步串行口状态) 寄存器	212
STKPTR (堆栈指针)	59
TRISE	131
TXSTA (发送状态和控制)	222
WDTCON (看门狗定时器控制)	278
状态	73
综述	66-68
寄存器文件	63
计算 GOTO	63

PIC18F2331/2431/4331/4431

间隔字符（12位）发送和接收	236
间接文件操作数	63
间接寻址	72
工作原理	71
INDF 和 FSR 寄存器	71
间接寻址操作	72
接收溢出表示位（SSPOV）	213
晶体振荡器 / 陶瓷谐振器	21
绝对最大额定值	337
K	
开发支持	331
看门狗定时器（WDT）	267, 278
编程注意事项	278
控制寄存器	278
相关寄存器	279
振荡器故障期间	281
勘误表	6
控制寄存器	
EECON1 和 EECON2	76
快速访问存储区	70
快速寄存器堆栈	60
框图	
16 位模式 Timer0	134
8 位模式 Timer0	134
比较模式操作	154
捕捉模式操作	153
带有外部输入的低压检测（LVD）	262
低压检测（LVD）	262
读表操作	75
读取闪存程序存储器	79
对闪存程序存储器的写表操作	81
故障保护时钟监视器	281
看门狗定时器	278
模拟输入模型	254
PIC18F2331/2431	10
PIC18F4331/4431	11
PLL	22
PWM（标准）	156
片上复位电路	45
RA0 引脚	108
RA1 引脚	108
RA3:RA2 引脚	108
RA4 引脚	109
RA5 引脚	110
RA6 引脚	110
RB3:RB0 引脚	113
RB4 引脚	114
RB5 引脚	115, 121
RB7:RB6 引脚	116
RC0 引脚	118
RC1 引脚	119
RC2 引脚	119
RC3 引脚	120
RC4 引脚	120
RC6 引脚	121
RC7 引脚	122
RD0 引脚	127
RD1 引脚	127
RD2 引脚	126
RD3 引脚	126

RD4 引脚	125
RD5 引脚	125
RD7:RD6 引脚	124
RE2:RE0 引脚	130
RE3 引脚	130
SSP（I ² C 模式）	217
SSP（SPI 模式）	214
Timer2	144
Timer5	146
Timer1	138
Timer1（16 位读 / 写模式）	138
通用 I/O 端口	107
USART 发送	231
USART 接收	233
外部上电复位电路（缓慢的 VDD 上电）	46
系统时钟	27
写表操作	76
中断逻辑	92

L

LFSR	311
LVD. 参见低压检测。	261

M

MOVF	311
MOVFF	312
MOVLB	312
MOVLW	313
MOVWF	313
MPLAB ASM30 汇编器、链接器、库管理器	332
MPLAB ICD 2 在线调试器	333
MPLAB ICE 2000 高性能通用在线仿真器	333
MPLAB ICE 4000 高性能通用在线仿真器	333
MPLAB IDE 软件	331
MPLINK 目标链接器 / MPLIB 目标库管理器	332
MULLW	314
MULWF	314
脉冲宽度调制. 参见 PWM（CCP 模块）和 PWM（ECCP 模块）。	
模数转换器. 参见 A/D。	

N

NEGF	315
NOP	315
内部互联总线（I ² C）. 参见 I ² C 模式。	
内部 RC 振荡器	
与 WDT 一起使用	278
内部振荡电路	
INTIO 模式	24
INTRC 输出频率	24
OSCTUNE 寄存器	24
调整	24

O

OPTION_REG 寄存器	
PSA 位	135
T0CS 位	135
T0PS2:T0PS0 位	135
T0SE 位	135

PIC18F2331/2431/4331/4431

P

PIC18F2X31/4X31 的数据存储器映射图	64
PICkit 1 闪存启动工具包	335
PICSTART Plus 开发编程器	334
PIE 寄存器	99
PIR 寄存器	96
PLL 锁定延时	46
POP	316
POR. 参见上电复位。	
PORTA	
LATA 寄存器	107
PORTA 寄存器	107
TRISA 寄存器	107
相关寄存器	111
PORTB	
LATB 寄存器	112
PORTB 寄存器	112
RB7:RB4 电平变化中断标志 (RBIF 位)	112
TRISB 寄存器	112
相关寄存器	117
PORTC	
LATC 寄存器	118
PORTC 寄存器	118
TRISC 寄存器	118
相关寄存器	123
PORTD	
LATD 寄存器	124
PORTD 寄存器	124
TRISD 寄存器	124
相关寄存器	128
PORTE	
LATE 寄存器	129
PORTE 寄存器	129
TRISE 寄存器	129
相关寄存器	132
PRO MATE II 通用器件编程器	333
PWM (CCP 模块)	156
CCPR1H:CCPR1L 寄存器	156
PWM 操作的建立	157
示例频率 / 分辨率	157
TMR2 与 PR2 匹配	143, 156
相关寄存器	157
占空比	156
周期	156
PUSH	316
PUSH 和 POP 指令	59
P (停止) 位	212
配置寄存器保护	286
配置位	267
评估和编程工具	335

Q

QEI 采样模式	172
器件比较	379
器件综述	7
其他特别功能	8
特性 (表)	9
新的内核功能	7
Q 时钟	157
欠压复位 (BOR)	46, 267

R

R/W 位	212, 218, 219
RAM. 参见数据存储器。	
RCALL	317
RCON 寄存器	
初始化期间的位状态	47
位和位置	47
RCSTA 寄存器	
SPEN 位	221
RC 振荡器	23
RCIO 振荡模式	23
RESET	317
RETFIE	318
RETLW	318
RETURN	319
RLCF	319
RLNCF	320
RRCF	320
RRNCF	321
软件模拟器 (MPLAB SIM30)	332
软件模拟器 (MPLAB SIM)	332

S

SCK	211
SCL	217
SDI	211
SDO	211
SETF	321
Sleep	322
SMP 位	212
SPI 模式	211
串行时钟	211
串行数据输出	211
串行数据输入	211
从动选择	211
相关寄存器	216
SS	211
SSP	
概述	
用于时钟移位的 TMR2 输出	143, 144
SSP I ² C 工作原理	217
从动模式	217
SSPEN 位	213
SSPM<3:0> 位	213
SSPOV 位	213
SUBFWB	322
SUBLW	323
SUBWF	323
SUBWFB	324
SWAPF	325
S (起始) 位	212
闪存程序存储器	75
表指针	78
范围 (基于操作)	78
表指针范围	78
擦除	80
擦除顺序	80
代码保护时的操作	83
读表与写表	75
读取	79

PIC18F2331/2431/4331/4431

控制寄存器	76	异步接收	234
TABLAT 寄存器	78	正常工作模式下的自动唤醒位 (WUE)	235
相关寄存器	83	主控 SSP I ² C 总线起始 / 停止位	367
写入	81	主控 SSP I ² C 总线数据	367
写入操作的意外终止	83	转换到 PRI_IDLE 模式的时序	35
写入校验	83	时序图和规范	355
上电复位 (POR)	46, 267	捕捉 / 比较 / PWM 要求	360
上电延时定时器 (PWRT)	46, 267	CLKO 和 I/O 要求	357
延时序列	46	DC 特性——内部 RC 精度	356
振荡器起振定时器 (OST)	46, 267	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	358
上电延迟	29	I ² C 总线数据要求 (从动模式)	366
上电延时定时器 (PWRT)	29, 46	PLL 时钟	356
时序图		SPI 从动模式要求示例 (CKE=1)	364
捕捉 / 比较 / PWM (CCP)	360	SPI 模式要求示例 (从动模式, CKE=0)	363
CLKO 和 I/O	357	SPI 模式要求示例 (主控模式, CKE=0)	361
从 PRI_IDLE 模式唤醒的转换时序	35	SPI 模式要求示例 (主控模式, CKE=1)	362
从 RC_RUN 模式唤醒的转换 (RC_RUN 到 NFP) ..	37	Timer0 和 Timer1 外部时钟要求	359
从 SEC_RUN 模式唤醒的转换 (辅助时钟到 HSPLL)	36	USART 同步发送要求	369
从休眠模式唤醒 (HSPLL) 的转换	34	USART 同步接收要求	369
低压检测特性	352	外部时钟时序要求	355
低压检测	264	主控 SSP I ² C 总线起始 / 停止位要求	367
发生 POR 且 PLL 使能时的延时序列 (MCLR 连接到 V _{DD})	55	主控 SSP I ² C 总线数据要求	368
发送间隔字符序列	236	时序图 / 指令周期	61
复位、看门狗定时器 (WDT)、振荡器起振 定时器 (OST) 和上电延时定时器 (PWRT)	358	时钟源	26
故障保护时钟监视器	282	使用 OSCCON 寄存器进行选择	26
缓慢上升时间 (MCLR 连接到 V _{DD} , V _{DD} 电压上升时间 > TPWRT)	55	数据 / 地址位 (D/A)	212
I ² C 发送 (7 位地址)	219	数据存储	63
I ² C 接收 (7 位地址)	219	PIC18F2X31/4X31 的映射图	64
I ² C 总线起始 / 停止位	365	特殊功能寄存器	65
I ² C 总线数据	365	通用寄存器	63
进入 SEC_IDLE 模式的转换	36	数据 EEPROM 存储器	85
进入 SEC_RUN 模式的转换	38	代码保护时的操作	88
进入 RC_IDLE 模式的转换	37	读取	87
进入 RC_RUN 模式的转换	39	EEADR 寄存器	85
进入休眠模式的转换	34	EECON1 和 EECON2 寄存器	85
PWM 输出	156	使用	88
欠压复位 (BOR)	358	误写操作保护	87
SPI 从动模式时序示例 (CKE=0)	363	相关寄存器	88
SPI 从动模式时序示例 (CKE=1)	364	写入	87
SPI 模式 (从动模式, CKE=0)	215	写入校验	87
SPI 模式 (从动模式, CKE=1)	216	数据 EEPROM 代码保护	286
SPI 模式 (主控模式)	215	双速起振	267, 279
SPI 主控模式时序示例 (CKE=0)	361	双字指令	
SPI 主控模式时序示例 (CKE=1)	362	示例情形	62
上电时的延时序列 (MCLR 不连接到 V _{DD})		所有寄存器的初始化条件	48-51
情形 1	54	T	
上电时的延时序列 (MCLR 不连接到 V _{DD})		TABLAT 寄存器	78
情形 2	54	TBLPTR 寄存器	78
上电时的延时序列 (MCLR 连接到 V _{DD} , V _{DD} 电压上升时间 < TPWRT)	54	TBLRD	326
时钟, 指令周期	61	TBLWT	327
双速起振时钟转换 (从 INTOSC 到 HSPLL)	280	Timer0	133
Timer0 和 Timer1 外部时钟	359	16 位模式定时器读写	135
同步发送	237	改变预分频器分配	135
同步发送 (通过 TXEN)	238	工作原理	135
同步接收 (主控模式, SREN)	239	时钟源边沿选择 (T0SE 位)	135
USART 同步发送 (主控 / 从动)	369	时钟源选择 (T0CS 位)	135
USART 同步接收 (主控 / 从动)	369	相关寄存器	135
外部时钟 (除 PLL 之外的所有模式)	355	预分频器 . 参见预分频器, Timer0. 中断	135
休眠模式下的自动唤醒位 (WUE)	235	Timer2	143
异步发送	231	工作原理	143
异步发送 (背靠背)	231	后分频器 . 参见后分频器, Timer2. PR2 寄存器	143, 156

PIC18F2331/2431/4331/4431

SSP 时钟移位	143, 144
TMR2 寄存器	143
TMR2 与 PR2 匹配中断	143, 144, 156
相关寄存器	144
预分频器. 参见预分频器, Timer2.	
Timer5	
框图	146
Timer1	137
16 位读 / 写模式	140
复位, 使用特殊事件触发器 (CCP)	140
工作原理	138
TMR1H 寄存器	137
TMR1L 寄存器	137
特殊事件触发器 (CCP)	154
相关寄存器	141
溢出中断	137
振荡器	137, 139
振荡器布局注意事项	139
中断	140
作为实时时钟使用	140
TSTFSZ	328
TXSTA 寄存器	
BRGH 位	225
特殊功能寄存器	65
映射图	65
特殊事件触发器. 参见比较 (CCP 模块)。	
同步串行端口模式选择位 (SSPM<3:0>)	213
同步串行端口使能位 (SSPEN)	213
同步串行端口. 参见 SSP。	
同步间隔字符自动唤醒	235
U	
UA	212
USART	
波特率发生器 (BRG)	225
波特率误差, 计算	226
波特率, 异步模式	226
采样	225
高波特率选择位 (BRGH 位)	225
功耗管理模式工作原理	225
相关寄存器	226
自动波特率检测	229
串行端口使能 (SPEN 位)	221
同步从动模式	241
发送	241
接收	242
相关寄存器, 发送	241
相关寄存器, 接收	242
同步主模式	237
发送	237
接收	239
相关寄存器, 发送	238
相关寄存器, 接收	240
异步模式	230
12 位间隔字符发送和接收	236
发送器	230
建立带地址检测的 9 位模式	233
接收器	233
同步间隔字符自动唤醒	235
相关寄存器, 发送	232
相关寄存器, 接收	234

W

WCOL 位	213
WWW 在线技术支持	6
外部时钟输入	23

X

XORLW	328
XORWF	329
写冲突检测位 (WCOL)	213
休眠	
OSC1 和 OSC2 引脚状态	29

Y

演示板

PICDEM 1	334
PICDEM 17	334
PICDEM 18R PIC18C601/801	335
PICDEM 2 Plus	334
PICDEM 3 PIC16C92X	334
PICDEM 4	334
PICDEM LIN PIC16C43X	335
PICDEM USB PIC16C7X5	335
PICDEM.net 因特网 / 以太网	334

引脚功能

MCLR/Vpp/RE3	12, 15
OSC1/CLKI/RA7	12, 15
OSC2/CLKO/RA6	12, 15
RA0/AN0	12, 15
RA1/AN1	12, 15
RA2/AN2/Vref-/CAP1/INDX	12, 15
RA3/AN3/Vref+/CAP2/QEA	12, 15
RA4/AN4/CAP3/QEB	15
RA4/CAP3/QEB	12
RA5/AN5/LVDIN	15
RB0/PWM0	13, 16
RB1/PWM1	13, 16
RB2/PWM2	13, 16
RB3/PWM3	13, 16
RB4/KBI0/PWM5	16
RB4/PWM5	13
RB5/KBI1/PWM4/PGM	13, 16
RB6/KBI2/PGC	13, 16
RB7/KBI3/PGD	13, 16
RC0/T1OSO/T1CKI	14, 17
RC1/T1OSI/CCP2/FLTA	14, 17
RC2/CCP1/FLT B	14, 17
RC3/T0CKI/T5CKI/INT0	14, 17
RC4/INT1/SDI/SDA	14, 17
RC5/INT2/SCK/SCL	14, 17
RC6/TX/CK/SS	14, 17
RC7/RX/DT/SDO	14, 17
RD0/T0CKI/T5CKI	18
RD1/SDO	18
RD2/SDI/SDA	18
RD3/SCK/SCL	18
RD4/FLTA	18
RD5/PWM4	18
RD6/PWM6	18
RD7/PWM7	18
RE0/AN6	19
RE1/AN7	19

PIC18F2331/2431/4331/4431

RE2/AN8	19	CPFSEQ	304
VDD	14, 19	CPFSGT	305
VSS	14, 19	CPFSLT	305
引脚 I/O 介绍		DAW	306
PIC18F2331/2431	12	DCFSNZ	307
PIC18F4331/4431	15	DECf	306
硬件乘法器	89	DECFSZ	307
工作原理	89	GOTO	308
简介	89	INCF	308
性能对比	89	INCFSZ	309
应用笔记		INFSNZ	309
AN578 (Use of the SSP Module in the I ² C Multi-Master Environment)	211	IORLW	310
预分频器, 捕捉	153	IORWF	310
预分频器, Timer0	135	LFSR	311
比率选择 (T0PS2:T0PS0 位)	135	MOVf	311
分配 (PSA 位)	135	MOVFF	312
预分频器, Timer2	157	MOVLB	312
Z		MOVLW	313
在线串行编程 (ICSP)	267, 286	MOVWF	313
在线调试器	286	MULLW	314
增强型通用同步 / 异步收发器 (EUSART)	221	MULWF	314
栈顶访问	58	NEGF	315
振荡器配置	21	NOP	315
EC	21	POP	316
ECIO	21	PUSH	316
HS	21	RCALL	317
HSPLL	21	RESET	317
INTIO1	21	RETFIE	318
INTIO2	21	RETLW	318
LP	21	RETURN	319
内部振荡电路	24	RLCF	319
RC	21	RLNCF	320
RCIO	21	RRCF	320
XT	21	RRNCF	321
振荡器起振定时器 (OST)	29, 46	SETF	321
振荡器切换	26	SLEEP	322
振荡器选择	267	SWAPF	325
振荡器转换	28	SUBFWB	322
振荡器, Timer1	137	SUBLW	323
指令格式	289	SUBWF	323
指令集	287	SUBWFB	324
ADDLW	293	TBLRD	326
ADDWF	293	TBLWT	327
ADDWFC	294	TSTFSZ	328
ANDLW	294	XORLW	328
ANDWF	295	XORWF	329
BC	295	总表	290
BCF	296	指令流 / 流水线	61
BN	296	指令周期	61
BNC	297	指针, FSRn	71
BNN	297	中断	91
BNOV	298	中断的现场保护	106
BNZ	298	中断源	267
BOV	301	比较完成 (CCP)	154
BRA	299	捕捉完成 (CCP)	153
BSF	299	电平变化中断 (RB7:RB4)	112
BTFSC	300	INTn 引脚	106
BTFSS	300	PORTB 电平变化中断	106
BTG	301	TMR0	106
BZ	302	TMR1 溢出	137
CALL	302	TMR2 与 PR2 匹配	144
CLRF	303	TMR2 与 PR2 匹配 (PWM)	143, 156
CLRWDT	303	中断, 标志位	
COMF	304	CCP1 标志 (CCP1IF 位)	153
		CCP1IF 标志 (CCP1IF 位)	154

PIC18F2331/2431/4331/4431

电平变化中断（RB7:RB4）标志（RBIF 位）	112
中断，使能位	
CCP1 使能（CCP1IE 位）	153
转换注意事项	380

MICROCHIP 网站

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

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

变更通知客户服务

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

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

客户支持

Microchip 产品的用户可通过以下渠道获得帮助:

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持
- 开发系统信息热线

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

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

PIC18F2331/2431/4331/4431

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: PIC18F2331/2431/4331/4431 文献编号: DS39616B_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?

PIC18F2331/2431/4331/4431

产品标识体系

欲订货，或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

器件编号	X	XX	XXX
器件	温度范围	封装	模式
器件	PIC18F2331/2431/4331/4431 ⁽¹⁾ , PIC18F2331/2431/4331/4431T ^(1,2) ; VDD 范围: 4.2V 至 5.5V		
	PIC18LF2331/2431/4331/4431 ⁽¹⁾ , PIC18LF2331/2431/4331/44310T ^(1,2) ; VDD 范围: 2.0V 至 5.5V		
温度范围	I	= -40°C 至 +85°C (工业级)	
封装	PT	= TQFP (薄型四方扁平封装)	
	SO	= SOIC	
	SP	= 窄型塑料 DIP	
	P	= PDIP	
	ML	= QFN	
模式	QTP、SQTP、编码或特殊要求 (空白为其他情况)		

示例:

- PIC18LF4431-I/P 301 = 工业级温度, PDIP 封装, 扩展级 VDD 范围, QTP 模式 #301。
- PIC18LF2331-I/SO = 工业级温度, SOIC 封装, 扩展级 VDD 范围。
- PIC18F4331-I/P = 工业级温度, PDIP 封装, 一般 VDD 范围。

注 1: F = 标准电压范围
LF = 宽电压范围

2: T = 卷带式——仅 SOIC 和 TQFP 封装。

销售与技术支持

数据手册

初始数据手册中所述的产品可能会有一份勘误表，其中描述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。要了解是否存在某一器件的勘误表，可通过以下方式联系我们：

1. Microchip 在当地的销售办事处
2. Microchip 网站 (www.microchip.com)

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)。

最新信息客户通知系统

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。



全球销售及及服务网点

美洲

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

技术支持:
<http://support.microchip.com>
网址: www.microchip.com

亚特兰大 **Atlanta**

Alpharetta, GA
Tel: 1-770-640-0034
Fax: 1-770-640-0307

波士顿 **Boston**

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

芝加哥 **Chicago**

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

达拉斯 **Dallas**

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

底特律 **Detroit**

Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

科科莫 **Kokomo**

Kokomo, IN
Tel: 1-765-864-8360
Fax: 1-765-864-8387

洛杉矶 **Los Angeles**

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

圣何塞 **San Jose**

Mountain View, CA
Tel: 1-650-215-1444
Fax: 1-650-961-0286

加拿大多伦多 **Toronto**

Mississauga, Ontario,
Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

中国 - 北京
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

中国 - 福州
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

中国 - 香港特别行政区
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

中国 - 顺德
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

中国 - 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

台湾地区 - 高雄
Tel: 886-7-536-4818
Fax: 886-7-536-4803

台湾地区 - 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-572-9526
Fax: 886-3-572-6459

亚太地区

澳大利亚 **Australia - Sydney**
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 **India - Bangalore**
Tel: 91-80-2229-0061
Fax: 91-80-2229-0062

印度 **India - New Delhi**
Tel: 91-11-5160-8631
Fax: 91-11-5160-8632

日本 **Japan - Kanagawa**
Tel: 81-45-471-6166
Fax: 81-45-471-6122

韩国 **Korea - Seoul**
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 **Malaysia - Penang**
Tel: 011-604-646-8870
Fax: 011-604-646-5086

菲律宾 **Philippines - Manila**
Tel: 011-632-634-9065
Fax: 011-632-634-9069

新加坡 **Singapore**
Tel: 65-6334-8870
Fax: 65-6334-8850

欧洲

奥地利 **Austria - Weis**
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

丹麦 **Denmark - Ballerup**
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 **France - Massy**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 **Germany - Ismaning**
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

英国 **England - Berkshire**
Tel: 44-118-921-5869
Fax: 44-118-921-5820

06/28/05