

PIC24FJ256GB110 系列 数据手册

**带 USB On-The-Go (OTG)
的 64/80/100 引脚
16 位闪存单片机**

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

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

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

商标

Microchip 的名称和徽标组合、Microchip 徽标、dsPIC、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PIC³² 徽标、rfPIC 和 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、Application Maestro、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified 徽标、MPLIB、MPILINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rfLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

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

在此提及的所有其他商标均为各持有公司所有。

© 2011, Microchip Technology Inc. 版权所有。

ISBN: 978-1-60932-995-2

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

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚洲和印度的设计中心均通过了 ISO/TS-16949:2002 认证。公司在 PIC[®] MCU 与 dsPIC[®] DSC、KEELOQ[®] 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品方面的质量体系流程均符合 ISO/TS-16949:2002。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。



MICROCHIP

PIC24FJ256GB110 系列

带 USB On-The-Go (OTG) 的 64/80/100 引脚 16 位闪存单片机

功耗管理:

- 片上 2.5V 稳压器
- 时钟源之间的实时切换
- 带快速唤醒的空闲、休眠和打盹模式以及双速启动
- 运行模式: 1 mA/MIPS, 2.0V 典型值
- 休眠模式的典型电流值低至 100 nA
- 采用 32 kHz 振荡器时, 2.0V 电压下的典型待机电流为 2.5 μA

通用串行总线特性:

- 符合 USB v2.0 On-The-Go (OTG) 规范
- 双角色能力——可以充当主机或外设
- 主机模式下的低速 (1.5 Mb/s) 和全速 (12 Mb/s) USB 操作
- 器件模式下的全速 USB 操作
- USB 的高精度 PLL
- 用于产生 USB 总线电压的内部升压辅助电路
- 用于产生 USB 总线电压的片外电荷泵的接口
- 支持最多 32 个端点 (两个方向各 16 个):
 - USB 模块可以将器件上的任意 RAM 存储单元用作 USB 端点缓冲区
- 带有片上稳压器的片上 USB 收发器
- 与外部 USB 收发器相连的接口
- 支持控制、中断、同步和批量传输
- 片上上拉和下拉电阻

高性能 CPU:

- 改进型哈佛架构
- 32 MHz 时最高 16 MIPS 工作速度
- 8 MHz 内部振荡器
- 17 位 x 17 位单周期硬件乘法器
- 32 位 /16 位硬件除法器
- 16 x 16 位工作寄存器阵列
- 带灵活寻址模式的 C 编译器优化指令集架构
- 可寻址最大 12 MB 的线性程序存储空间
- 可寻址最大 64 KB 的线性数据存储空间
- 2 个地址发生单元, 分别用于数据存储器的读和写寻址

模拟特性:

- 最多 16 路通道的 10 位模数 (Analog-to-Digital, A/D) 转换器, 转换速率为 500 ksps:
 - 可在休眠模式下进行转换
- 3 个带可编程输入 / 输出配置的模拟比较器
- 充电时间测量单元 (Charge Time Measurement Unit, CTMU)

器件	引脚数	程序存储器 (字节数)	SRAM (字节数)	可重映射的外设							I ² C TM	10 位 A/D (通道数)	比较器	PMP/PSP	JTAG	CTMU	USB OTG
				可重映射的 引脚数	16 位定时器	输入捕捉	比较 / PWM 输出	带 IrDA [®] 的 UART	SPI								
PIC24FJ64GB106	64	64K	16K	29	5	9	9	4	3	3	16	3	有	有	有	有	
PIC24FJ128GB106	64	128K	16K	29	5	9	9	4	3	3	16	3	有	有	有	有	
PIC24FJ192GB106	64	192K	16K	29	5	9	9	4	3	3	16	3	有	有	有	有	
PIC24FJ256GB106	64	256K	16K	29	5	9	9	4	3	3	16	3	有	有	有	有	
PIC24FJ64GB108	80	64K	16K	40	5	9	9	4	3	3	16	3	有	有	有	有	
PIC24FJ128GB108	80	128K	16K	40	5	9	9	4	3	3	16	3	有	有	有	有	
PIC24FJ192GB108	80	192K	16K	40	5	9	9	4	3	3	16	3	有	有	有	有	
PIC24FJ256GB108	80	256K	16K	40	5	9	9	4	3	3	16	3	有	有	有	有	
PIC24FJ64GB110	100	64K	16K	44	5	9	9	4	3	3	16	3	有	有	有	有	
PIC24FJ128GB110	100	128K	16K	44	5	9	9	4	3	3	16	3	有	有	有	有	
PIC24FJ192GB110	100	192K	16K	44	5	9	9	4	3	3	16	3	有	有	有	有	
PIC24FJ256GB110	100	256K	16K	44	5	9	9	4	3	3	16	3	有	有	有	有	

PIC24FJ256GB110 系列

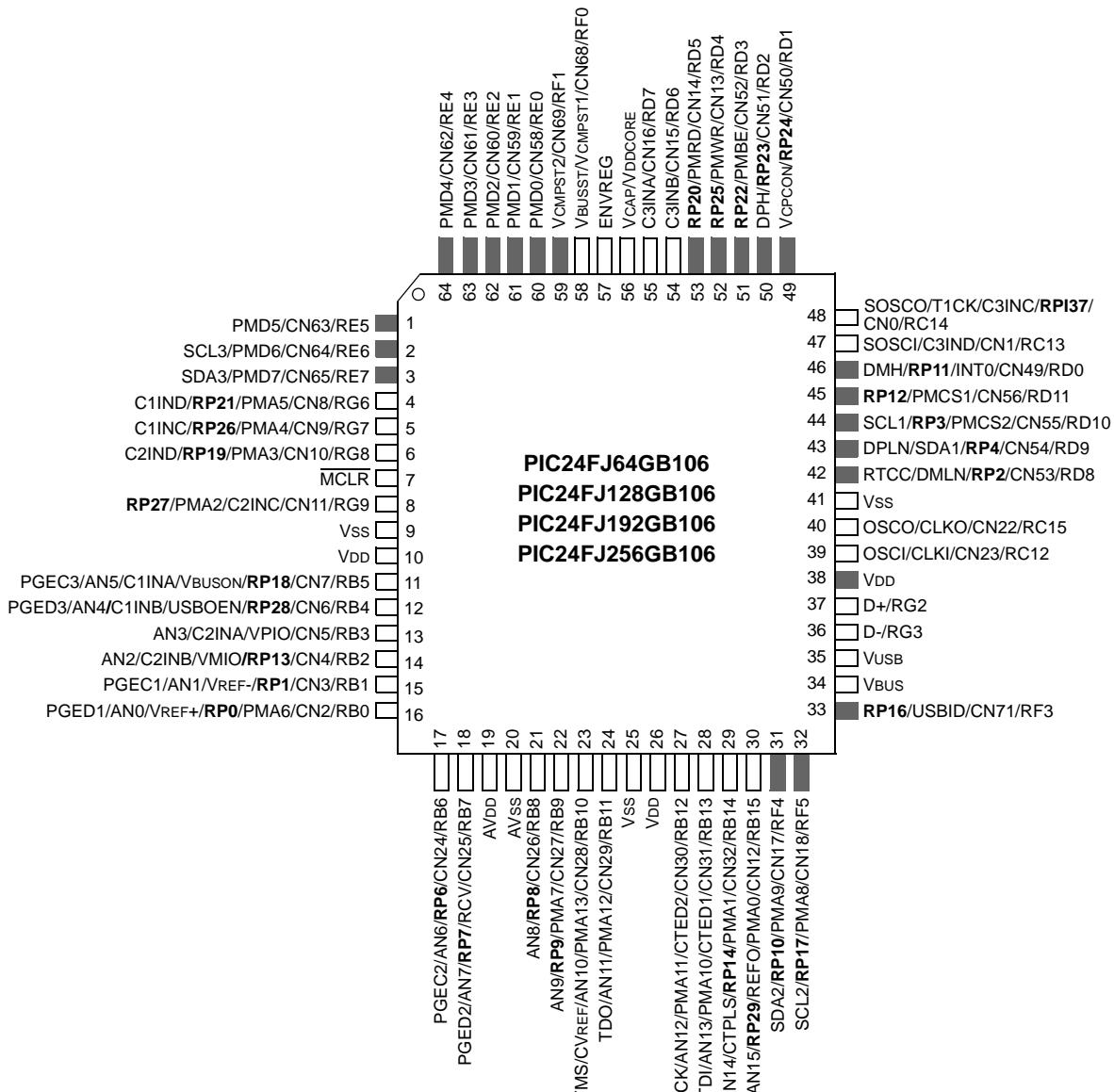
外设特性:

- 外设引脚选择 (Peripheral Pin Select, PPS):
 - 允许在运行时对许多外设进行独立 I/O 映射
 - 持续的硬件完整性检查和安全互锁以防意外的配置更改
 - 最多 44 个可用引脚 (100 引脚器件)
- 3 个带有 8 级深 FIFO 缓冲区的 3 线/4 线 SPI 模块 (支持 4 帧模式)
- 3 个 I²CTM 模块支持多主器件 / 从模式和 7 位/10 位寻址
- 4 个 UART 模块:
 - 支持 RS-485、RS-232、LIN/J2602 协议和 IrDA[®]
 - 片上 IrDA 硬件编码器 / 解码器
 - 自动唤醒和自动波特率检测 (Auto-Baud Detect, ABD)
 - 4 级深 FIFO 缓冲区
- 5 个带可编程预分频器的 16 位定时器 / 计数器
- 9 个 16 位捕捉输入, 每个均带有专用时基
- 9 个 16 位比较 /PWM 输出, 每个均带有专用时基
- 8 位并行主端口 (PMP/PSP):
 - 最多 16 个地址引脚
 - 控制线上的可编程极性
- 硬件实时时钟 / 日历 (Real-Time Clock/Calendar, RTCC):
 - 提供时钟、日历和闹钟功能
 - 可编程循环冗余校验 (Cyclic Redundancy Check, CRC) 发生器
 - 最多 5 个外部中断源

单片机特性:

- 工作电压范围为 2.0V 到 3.6V
- 可在软件控制下自编程
- 最大容许输入电压为 5.5V (仅限数字引脚)
- 数字 I/O 上的可配置漏极开路输出
- 所有 I/O 上的高灌 / 拉电流 (18 mA/18 mA)
- 可选择的功耗管理模式:
 - 带快速唤醒的休眠、空闲和打盹模式
- 故障保护时钟监视器操作:
 - 检测时钟故障并切换到片上低功耗 RC 振荡器
- 片上 LDO 稳压器
- 上电复位 (Power-on Reset, POR)、上电延时定时器 (Power-up Timer, PWRT)、低压检测 (Low-Voltage Detect, LVD) 和振荡器起振定时器 (Oscillator Start-up Timer, OST)
- 灵活的看门狗定时器 (Watchdog Timer, WDT), 带片上低功耗 RC 振荡器以便可靠地工作
- 通过 2 个引脚进行在线串行编程 (In-Circuit Serial ProgrammingTM, ICSPTM) 和在线调试 (In-Circuit Debug, ICD)
- 支持 JTAG 边界扫描和编程
- 欠压复位 (Brown-out Reset, BOR)
- 闪存程序存储器:
 - 至少可耐受 10,000 次擦写
 - 最少 20 年数据保存时间
 - 可选择的写保护边界
 - 用于闪存配置字的写保护选项

引脚图 (64 引脚 TQFP 和 QFN)

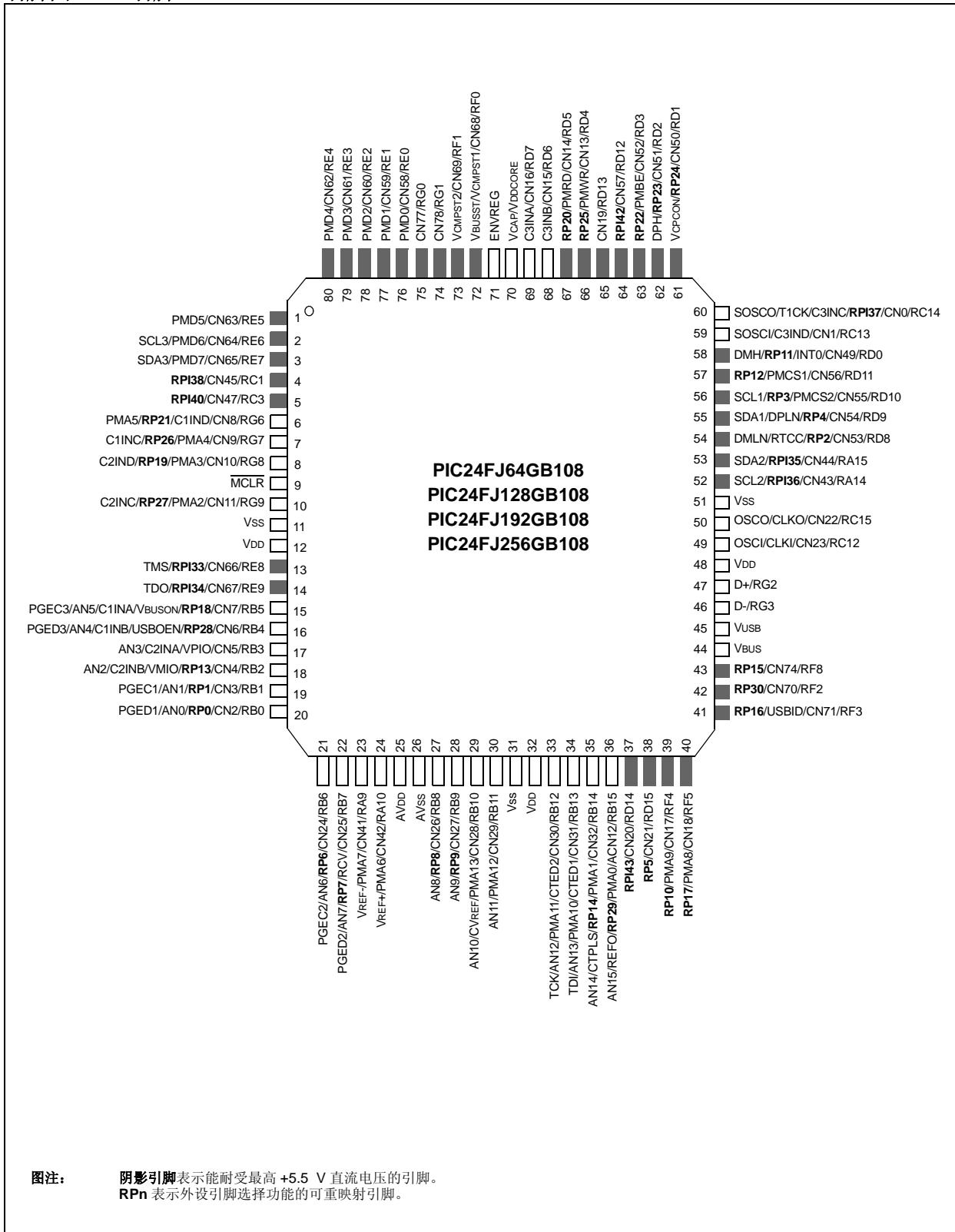


图注： 阴影引脚表示能耐受最高 +5.5 V 直流电压的引脚。

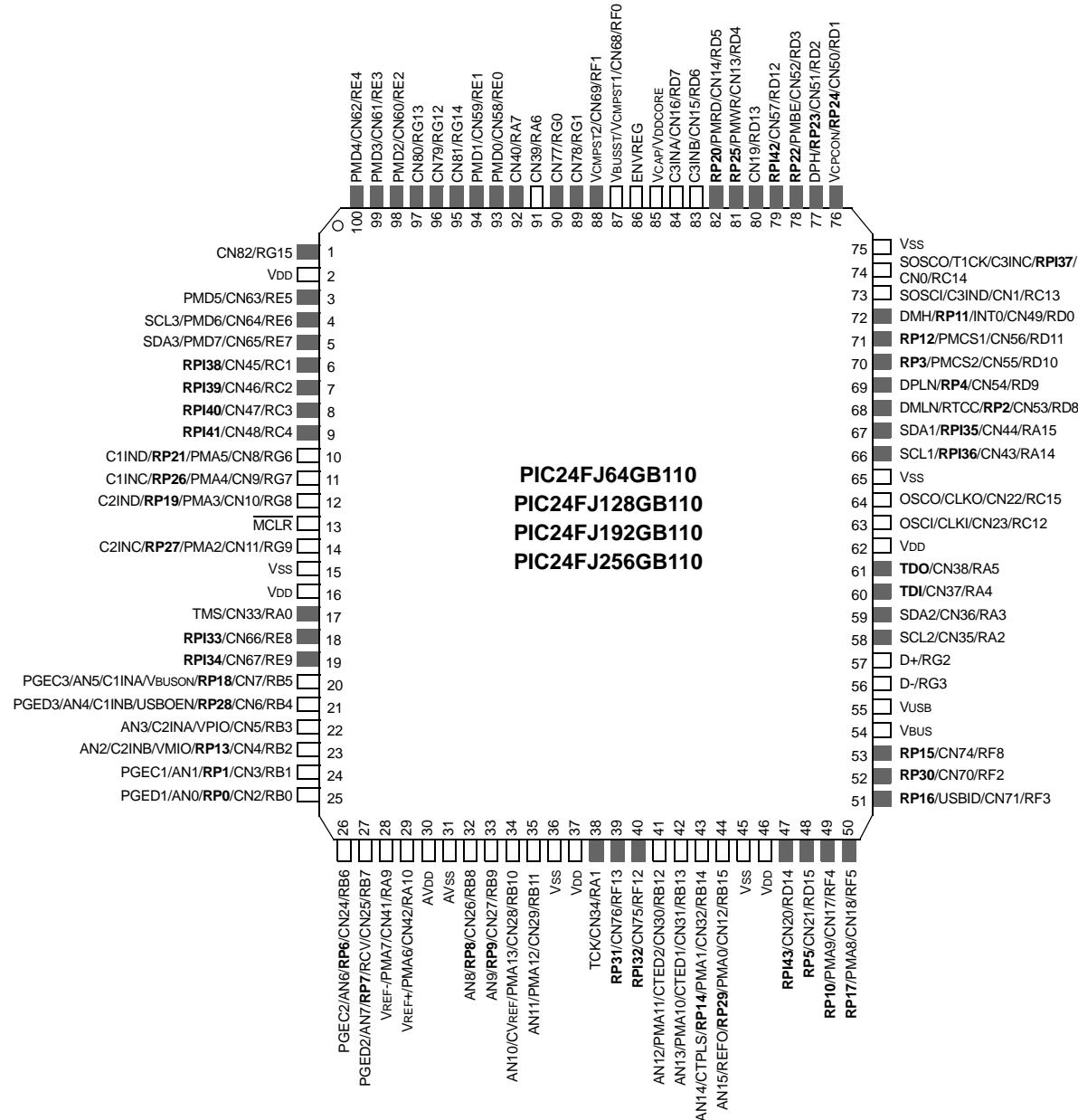
注 1： RP_n 表示外设引脚选择功能的可重映射引脚。
对于 QFN 器件，器件反面的底板必须连接至 V_{SS}。

PIC24FJ256GB110 系列

引脚图（80 引脚 TQFP）



引脚图 (100 引脚 TQFP)



图注：

阴影引脚表示能耐受最高 +5.5 V 直流电压的引脚。
R_{Pn} 和 R_{PIn} 表示外设引脚选择功能的可重映射引脚。

PIC24FJ256GB110 系列

目录

1.0 器件概述	11
2.0 16 位单片机入门指南	27
3.0 CPU.....	33
4.0 存储器构成	39
5.0 闪存程序存储器	63
6.0 复位	71
7.0 中断控制器	77
8.0 振荡器配置	121
9.0 节能特性.....	131
10.0 I/O 端口	133
11.0 Timer1	161
12.0 Timer2/3 和 Timer4/5	163
13.0 带专用定时器的输入捕捉.....	169
14.0 带专用定时器的输出比较.....	173
15.0 串行外设接口（SPI）	181
16.0 I ² C TM	191
17.0 通用异步收发器（UART）	199
18.0 带 On-The-Go 支持的通用串行总线（USB OTG）	207
19.0 并行主端口（PMP）	241
20.0 实时时钟和日历（RTCC）	251
21.0 可编程循环冗余校验（CRC）发生器	263
22.0 10 位高速 A/D 转换器	267
23.0 三比较器模块.....	277
24.0 比较器参考电压	281
25.0 充电时间测量单元（CTMU）	283
26.0 特殊功能.....	287
27.0 开发支持	299
28.0 指令集汇总	303
29.0 电气特性	311
30.0 封装信息	327
附录 A: 版本历史	341
索引	343
Microchip 网站	349
变更通知客户服务	349
客户支持	349
读者反馈表	350
产品标识体系	351

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。

PIC24FJ256GB110 系列

注:

1.0 器件概述

本文档涉及以下器件的具体信息：

- PIC24FJ64GB106 • PIC24FJ192GB108
- PIC24FJ128GB106 • PIC24FJ256GB108
- PIC24FJ192GB106 • PIC24FJ64GB110
- PIC24FJ256GB106 • PIC24FJ128GB110
- PIC24FJ64GB108 • PIC24FJ192GB110
- PIC24FJ128GB108 • PIC24FJ256GB110

它扩展了 Microchip 已有的 16 位单片机产品线，结合了扩展的外设功能部件和增强的计算性能，并提供了新的连接选项：USB On-The-Go。PIC24FJ256GB110 系列为高性能 USB 应用提供了新的平台，这些应用可能需要超过 8 位的平台，但无需数字信号处理器的功能。

1.1 内核特性

1.1.1 16 位架构

所有 PIC24F 器件的核心都是 16 位改进型哈佛架构，第一次引入该架构的就是 Microchip 的 dsPIC® 数字信号控制器。PIC24F CPU 内核提供众多增强功能，例如：

- 16 位数据和 24 位地址路径，能够在数据和存储空间之间传送信息
- 最大 12 MB（程序空间）和 64 KB（数据）的线性寻址
- 16 个元素组成的工作寄存器阵列，支持内置软件堆栈
- 17 x 17 位的硬件乘法器，支持整数数学运算
- 硬件支持 32 位除以 16 位的除法运算
- 指令集支持多寻址模式，为高级语言（如 C 语言）做了优化
- 工作性能最高可达 16 MIPS

1.1.2 节能技术

PIC24FJ256GB110 系列的所有器件具有一系列能显著降低工作时功耗的功能。主要包含以下几项：

- **动态时钟切换：**工作期间，器件时钟可由软件控制更改为 Timer1 时钟源或内部低功耗 RC 振荡器，使用户能在他们的软件设计中融入节能理念。

- **打盹模式工作原理：**当对时序敏感的应用（如串行通信）需要不间断地操作外设时，可有选择地降低 CPU 时钟速度，在不丢失信号节拍的前提下节约功耗。

- **基于指令的节能模式：**单片机可用一条软件指令，在保持外设活动的同时暂停所有操作，或有选择地关闭内核。

1.1.3 振荡器选项和特性

PIC24FJ256GB110 系列的所有器件可提供 5 个不同的振荡器选项，使用户在开发应用硬件时有很大的选择范围。这些选项包括：

- 两种晶振模式，使用晶振或陶瓷谐振器。
- 两种外部时钟模式，提供 2 分频时钟输出选项。
- 标称 8 MHz 输出的快速内部振荡器（Fast Internal Oscillator, FRC），它也可以由软件控制分频，提供低至 31 kHz 的时钟速度。
- 1 个锁相环（Phase Lock Loop, PLL）倍频器，可在外部振荡器模式和 FRC 振荡器模式下使用，可使时钟速度最高达到 32 MHz。
- 带有固定 31 kHz 输出的独立内部 RC 振荡器（LPRC），为对时序不敏感的应用提供低功耗选择。

内部振荡器模块也为故障保护时钟监视器提供稳定的参考源。该选件持续对照内部振荡器提供的参考信号监视主时钟源，使控制器能够切换到内部振荡器，从而允许持续低速工作或安全关闭应用。

1.1.4 移植方便

无论存储器容量如何，所有器件都共享相同的丰富外设，使应用的扩展和升级工作变得轻松而容易。整个系列的引脚排列设计一致也有助于从一个器件向下一代更大的器件移植，甚至从 64 引脚移植到 100 引脚器件。

PIC24F 系列的引脚兼容 dsPIC33 系列的器件，并部分兼容 PIC18 和 dsPIC30 的引脚排列设计。这就方便了相对简单的应用扩展为功能强大而复杂的应用，但仍选择 Microchip 器件。

PIC24FJ256GB110 系列

1.2 USB On-The-Go

Microchip 通过 PIC24FJ256GB110 系列器件在单个芯片上引入了 USB On-The-Go 功能，丰富了它的产品线。这一新模块提供了与符合 USB 2.0 标准的目标器件一样的片上功能，以及与 USB 嵌入式主机一样的有限独立功能。通过实现 USB 主机协商协议（Host Negotiation Protocol, HNP），该模块也可以动态地在器件和主机操作间切换，从而可以在单片机平台上实现范围更广的 USB 灵活应用。

除了 USB 主机功能外，PIC24FJ256GB110 系列器件还提供了真正的单芯片 USB 解决方案，包括片上收发器和稳压器，以及升压发生器，在主机操作期间提供总线电源。

1.3 其他特殊功能

- 外设引脚选择：**外设引脚选择（PPS）功能允许许多数字外设通过固定的一组数字 I/O 引脚映射。用户可以将许多数字外设的输入和 / 或输出独立地映射到 I/O 引脚中的任何一个。
- 通信：**PIC24FJ256GB110 系列包含了一系列串行通信外设，可处理各种应用需求。有 3 个独立的 I²C 模块，支持主 / 从工作模式。通过外设引脚选择功能，器件还具有 4 个独立 UART，带有内置 IrDA 编码器 / 解码器和 3 个 SPI 模块。
- 模拟特性：**PIC24FJ256GB110 系列的所有成员都包含一个 10 位 A/D 转换器模块和一个三比较器模块。A/D 模块具备可编程采集时间，从而不必在选择通道和启动转换之间等待一个采样周期，并且加快了采样速度。比较器模块包含 3 个模拟比较器，它们可针对多种操作进行配置。
- CTMU 接口：**除了它们的其他模拟功能外，PIC24FJ256GB110 系列的成员包含全新的 CTMU 接口模块。它提供了精确测量时间和生成脉冲的方便途径，并可用作容性传感器的接口。

- 并行主 / 增强型并行从端口：**通用 I/O 端口之一可为增强型并行数据通信重新进行配置。在该模式下，该端口可配置为主 / 从操作，并且支持主模式中最多 16 根外部地址线的 8 位和 16 位数据传输。
- 实时时钟 / 日历：**该模块在硬件中实现了带闹钟功能的全功能时钟和日历，释放了定时器资源和程序存储空间供内核应用使用。

1.4 系列中各产品的详细说明

PIC24FJ256GB110 系列器件具有 64 引脚、80 引脚和 100 引脚封装形式。图 1-1 给出了所有器件的一般框图。这些器件在以下四个方面存在差异：

- 闪存程序存储器（PIC24FJ64GB1 器件为 64 KB，PIC24FJ128GB1 器件为 128 KB，PIC24FJ192GB1 器件为 192 KB，PIC24FJ256GB1 器件为 256 KB）。
- 可用的 I/O 引脚数和端口数（64 引脚器件有 6 个端口 51 个引脚，80 引脚器件有 7 个端口 65 个引脚，100 引脚器件有 7 个端口 83 个引脚）。
- 可用的电平变化中断通知（Interrupt-on-Change Notification, ICN）输入（64 引脚器件上有 49 个，80 引脚器件上有 63 个，100 引脚器件上有 81 个）。
- 可用的可重映射引脚数（64 引脚器件上有 29 个引脚，80 引脚器件上有 40 个引脚，100 引脚器件上有 44 个引脚）。

本系列器件的其他功能都是相同的。表 1-1 汇总了这些功能。

按功能排序的 PIC24FJ256GB110 系列器件可用引脚功能列表显示在表 1-4 中。请注意，该表显示的是各外设功能的引脚位置，而非它们如何在同一引脚上复用。这一信息在数据手册开头处的引脚图中提供。复用的功能按功能的优先级排序，优先级最高的外设列在最前面。

PIC24FJ256GB110 系列

表 1-1： PIC24FJ256GB110 系列器件特性：64 引脚器件

特性	64GB106	128GB106	192GB106	256GB106
工作频率	DC – 32 MHz			
程序存储器（字节数）	64K	128K	192K	256K
程序存储器（指令数）	22,016	44,032	67,072	87,552
数据存储器（字节数）	16,384			
中断源（软向量 /NMI 陷阱）	66 (62/4)			
I/O 端口	端口 B, C, D, E, F, G			
总 I/O 引脚数	51			
可重映射的引脚数	29 (28 个 I/O, 1 个仅输入)			
定时器：				
总数（16 位）	5 ⁽¹⁾			
32 位（来自成对的 16 位定时器）	2			
输入捕捉通道数	9 ⁽¹⁾			
输出比较 /PWM 通道数	9 ⁽¹⁾			
输入电平变化通知中断	49			
串行通信：				
UART	4 ⁽¹⁾			
SPI (3 线 /4 线)	3 ⁽¹⁾			
I ² C TM	3			
并行通信 (PMP/PSP)	有			
JTAG 边界扫描 / 编程	有			
10 位模数转换模块（输入通道数）	16			
模拟比较器	3			
CTMU 接口	有			
复位（和延时）	POR、BOR、RESET 指令、MCLR、WDT；非法操作码、 REPEAT 指令、硬件陷阱、配置字不匹配 (PWRT、OST 和 PLL 锁定)			
指令集	76 条基本指令，多种寻址模式			
封装	64 引脚 TQFP			

注 1：外设可通过可重映射的引脚进行访问。

PIC24FJ256GB110 系列

表 1-2： PIC24FJ256GB110 系列器件特性： 80 引脚器件

特性	64GB108	128GB108	192GB108	256GB108
工作频率	DC – 32 MHz			
程序存储器（字节数）	64K	128K	192K	256K
程序存储器（指令数）	22,016	44,032	67,072	87,552
数据存储器（字节数）	16,384			
中断源（软向量 /NMI 陷阱）	66 (62/4)			
I/O 端口	端口 A, B, C, D, E, F, G			
总 I/O 引脚数	65			
可重映射的引脚数	40 (31 个 I/O, 9 个仅输入)			
定时器：				
总数（16 位）	5 ⁽¹⁾			
32 位（来自成对的 16 位定时器）	2			
输入捕捉通道数	9 ⁽¹⁾			
输出比较 /PWM 通道数	9 ⁽¹⁾			
输入电平变化通知中断	63			
串行通信：				
UART	4 ⁽¹⁾			
SPI (3 线 /4 线)	3 ⁽¹⁾			
I ² C TM	3			
并行通信 (PMP/PSP)	有			
JTAG 边界扫描 / 编程	有			
10 位模数转换模块（输入通道数）	16			
模拟比较器	3			
CTMU 接口	有			
复位（和延时）	POR、BOR、RESET 指令、MCLR、WDT；非法操作码、 REPEAT 指令、硬件陷阱、配置字不匹配 (PWRT、OST 和 PLL 锁定)			
指令集	76 条基本指令，多种寻址模式			
封装	80 引脚 TQFP			

注 1： 外设可通过可重映射的引脚进行访问。

PIC24FJ256GB110 系列

表 1-3： PIC24FJ256GB110 系列器件特性：100 引脚器件

特性	64GB110	128GB110	192GB110	256GB110
工作频率	DC – 32 MHz			
程序存储器（字节数）	64K	128K	192K	256K
程序存储器（指令数）	22,016	44,032	67,072	87,552
数据存储器（字节数）	16,384			
中断源（软向量 /NMI 陷阱）	66 (62/4)			
I/O 端口	端口 A, B, C, D, E, F, G			
总 I/O 引脚数	83			
可重映射的引脚数	44 (32 个 I/O, 12 个仅输入)			
定时器：				
总数（16 位）	5 ⁽¹⁾			
32 位（来自成对的 16 位定时器）	2			
输入捕捉通道数	9 ⁽¹⁾			
输出比较 /PWM 通道数	9 ⁽¹⁾			
输入电平变化通知中断	81			
串行通信：				
UART	4 ⁽¹⁾			
SPI (3 线 /4 线)	3 ⁽¹⁾			
I ² C TM	3			
并行通信 (PMP/PSP)	有			
JTAG 边界扫描 / 编程	有			
10 位模数转换模块（输入通道数）	16			
模拟比较器	3			
CTMU 接口	有			
复位（和延时）	POR、BOR、RESET 指令、MCLR、WDT；非法操作码、 REPEAT 指令、硬件陷阱、配置字不匹配 (PWRT、OST 和 PLL 锁定)			
指令集	76 条基本指令，多种寻址模式			
封装	100 引脚 TQFP			

注 1：外设可通过可重映射的引脚进行访问。

PIC24FJ256GB110 系列

图 1-1: PIC24FJ256GB110 系列一般框图

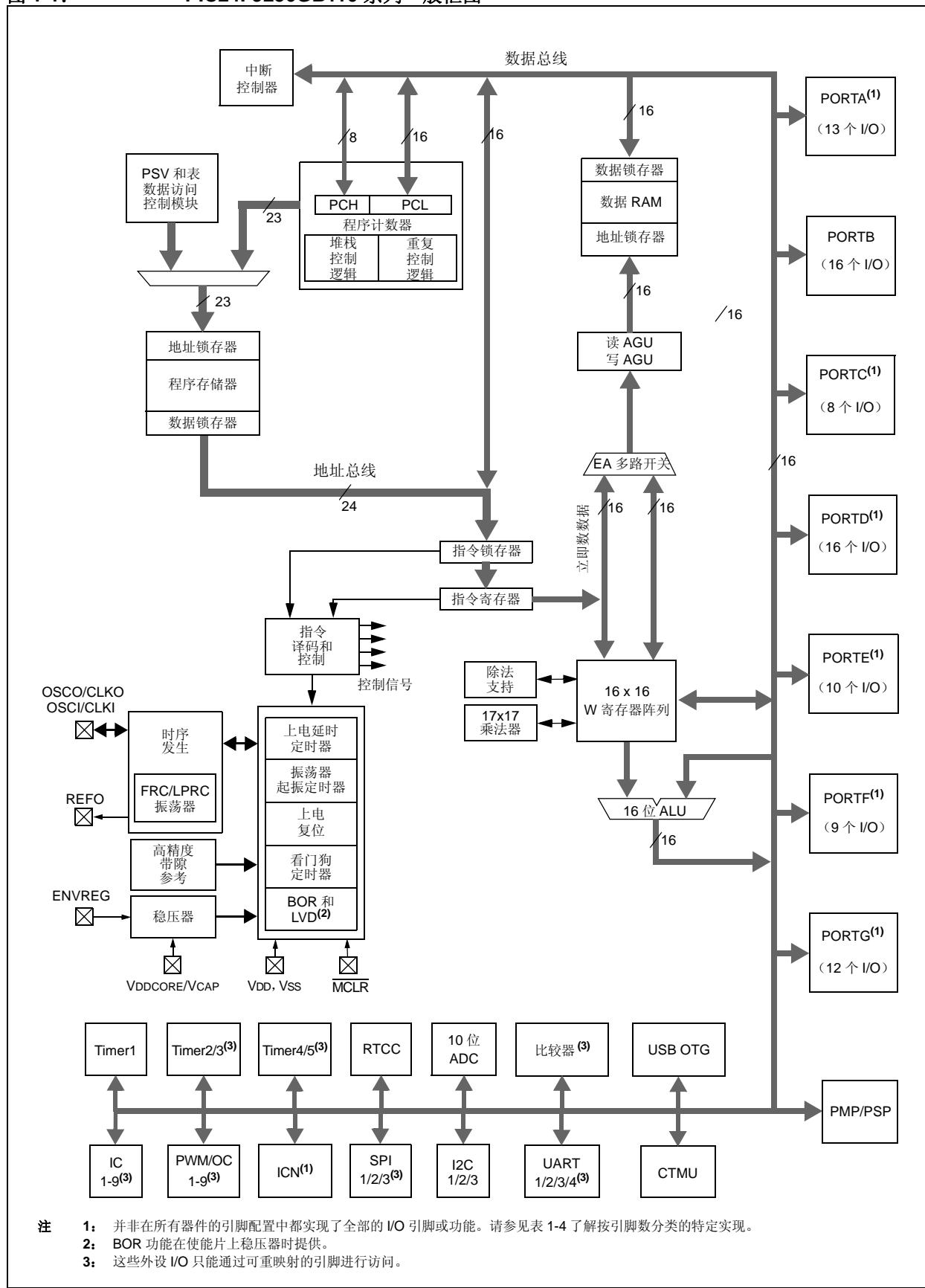


表 1-4: PIC24FJ256GB110 系列引脚说明

功能	引脚编号			I/O	输入 缓冲器	说明
	64 引脚 TQFP 和 QFN	80 引脚 TQFP	100 引脚 TQFP			
AN0	16	20	25	I	ANA	A/D 模拟输入。
AN1	15	19	24	I	ANA	
AN2	14	18	23	I	ANA	
AN3	13	17	22	I	ANA	
AN4	12	16	21	I	ANA	
AN5	11	15	20	I	ANA	
AN6	17	21	26	I	ANA	
AN7	18	22	27	I	ANA	
AN8	21	27	32	I	ANA	
AN9	22	28	33	I	ANA	
AN10	23	29	34	I	ANA	
AN11	24	30	35	I	ANA	
AN12	27	33	41	I	ANA	
AN13	28	34	42	I	ANA	
AN14	29	35	43	I	ANA	
AN15	30	36	44	I	ANA	
AVDD	19	25	30	P	—	模拟模块的正电源。
AVSS	20	26	31	P	—	模拟模块的参考地。
C1INA	11	15	20	I	ANA	比较器 1 的输入 A。
C1INB	12	16	21	I	ANA	比较器 1 的输入 B。
C1INC	5	7	11	I	ANA	比较器 1 的输入 C。
C1IND	4	6	10	I	ANA	比较器 1 的输入 D。
C2INA	13	17	22	I	ANA	比较器 2 的输入 A。
C2INB	14	18	23	I	ANA	比较器 2 的输入 B。
C2INC	8	10	14	I	ANA	比较器 2 的输入 C。
C2IND	6	8	12	I	ANA	比较器 2 的输入 D。
C3INA	55	69	84	I	ANA	比较器 3 的输入 A。
C3INB	54	68	83	I	ANA	比较器 3 的输入 B。
C3INC	48	60	74	I	ANA	比较器 3 的输入 C。
C3IND	47	59	73	I	ANA	比较器 3 的输入 D。
CLKI	39	49	63	I	ANA	主时钟输入连接。
CLKO	40	50	64	O	—	系统时钟输出。

图注: TTL = TTL 输入缓冲器
ANA = 模拟电平输入 / 输出

ST = 施密特触发器输入缓冲器
 $\text{I}^2\text{C}^{\text{TM}}$ = $\text{I}^2\text{C}/\text{SMBus}$ 输入缓冲器

PIC24FJ256GB110 系列

表 1-4: PIC24FJ256GB110 系列引脚说明 (续)

功能	引脚编号			I/O	输入 缓冲器	说明
	64 引脚 TQFP 和 QFN	80 引脚 TQFP	100 引脚 TQFP			
CN0	48	60	74	I	ST	电平变化中断输入。
CN1	47	59	73	I	ST	
CN2	16	20	25	I	ST	
CN3	15	19	24	I	ST	
CN4	14	18	23	I	ST	
CN5	13	17	22	I	ST	
CN6	12	16	21	I	ST	
CN7	11	15	20	I	ST	
CN8	4	6	10	I	ST	
CN9	5	7	11	I	ST	
CN10	6	8	12	I	ST	
CN11	8	10	14	I	ST	
CN12	30	36	44	I	ST	
CN13	52	66	81	I	ST	
CN14	53	67	82	I	ST	
CN15	54	68	83	I	ST	
CN16	55	69	84	I	ST	
CN17	31	39	49	I	ST	
CN18	32	40	50	I	ST	
CN19	—	65	80	I	ST	
CN20	—	37	47	I	ST	
CN21	—	38	48	I	ST	
CN22	40	50	64	I	ST	
CN23	39	49	63	I	ST	
CN24	17	21	26	I	ST	
CN25	18	22	27	I	ST	
CN26	21	27	32	I	ST	
CN27	22	28	33	I	ST	
CN28	23	29	34	I	ST	
CN29	24	30	35	I	ST	
CN30	27	33	41	I	ST	
CN31	28	34	42	I	ST	
CN32	29	35	43	I	ST	
CN33	—	—	17	I	ST	
CN34	—	—	38	I	ST	
CN35	—	—	58	I	ST	
CN36	—	—	59	I	ST	
CN37	—	—	60	I	ST	
CN38	—	—	61	I	ST	
CN39	—	—	91	I	ST	
CN40	—	—	92	I	ST	
CN41	—	23	28	I	ST	
CN42	—	24	29	I	ST	

图注: TTL = TTL 输入缓冲器
ANA = 模拟电平输入 / 输出

ST = 施密特触发器输入缓冲器
 I^2C^{TM} = $I^2C/SMBus$ 输入缓冲器

PIC24FJ256GB110 系列

表 1-4: PIC24FJ256GB110 系列引脚说明 (续)

功能	引脚编号			I/O	输入 缓冲器	说明
	64 引脚 TQFP 和 QFN	80 引脚 TQFP	100 引脚 TQFP			
CN43	—	52	66	I	ST	电平变化中断输入。
CN44	—	53	67	I	ST	
CN45	—	4	6	I	ST	
CN46	—	—	7	I	ST	
CN47	—	5	8	I	ST	
CN48	—	—	9	I	ST	
CN49	46	58	72	I	ST	
CN50	49	61	76	I	ST	
CN51	50	62	77	I	ST	
CN52	51	63	78	I	ST	
CN53	42	54	68	I	ST	
CN54	43	55	69	I	ST	
CN55	44	56	70	I	ST	
CN56	45	57	71	I	ST	
CN57	—	64	79	I	ST	
CN58	60	76	93	I	ST	
CN59	61	77	94	I	ST	
CN60	62	78	98	I	ST	
CN61	63	79	99	I	ST	
CN62	64	80	100	I	ST	
CN63	1	1	3	I	ST	
CN64	2	2	4	I	ST	
CN65	3	3	5	I	ST	
CN66	—	13	18	I	ST	
CN67	—	14	19	I	ST	
CN68	58	72	87	I	ST	
CN69	59	73	88	I	ST	
CN70	—	42	52	I	ST	
CN71	33	41	51	I	ST	
CN74	—	43	53	I	ST	
CN75	—	—	40	I	ST	
CN76	—	—	39	I	ST	
CN77	—	75	90	I	ST	
CN78	—	74	89	I	ST	
CN79	—	—	96	I	ST	
CN80	—	—	97	I	ST	
CN81	—	—	95	I	ST	
CN82	—	—	1	I	ST	
CTED1	28	34	42	I	ANA	CTMU 外部边沿输入 1。
CTED2	27	33	41	I	ANA	CTMU 外部边沿输入 2。
CTPLS	29	35	43	O	—	CTMU 脉冲输出。
CVREF	23	29	34	O	—	比较器参考电压输出。

图注: TTL = TTL 输入缓冲器
ANA = 模拟电平输入 / 输出

ST = 施密特触发器输入缓冲器
 I^2C^{TM} = I^2C/SMBus 输入缓冲器

PIC24FJ256GB110 系列

表 1-4: PIC24FJ256GB110 系列引脚说明 (续)

功能	引脚编号			I/O	输入 缓冲器	说明
	64 引脚 TQFP 和 QFN	80 引脚 TQFP	100 引脚 TQFP			
D+	37	47	57	I/O	—	USB 差分正信号线 (内部收发器)。
D-	36	46	56	I/O	—	USB 差分负信号线 (内部收发器)。
DMH	46	58	72	O	—	D- 外部上拉控制输出。
DMLN	42	54	68	O	—	D- 外部下拉控制输出。
DPH	50	62	77	O	—	D+ 外部上拉控制输出。
DPLN	43	55	69	O	—	D+ 外部下拉控制输出。
ENVREG	57	71	86	I	ST	稳压器使能。
INT0	46	58	72	I	ST	外部中断输入。
MCLR	7	9	13	I	ST	主复位 (器件复位) 输入。此线变为低电平, 引起复位。
OSCI	39	49	63	I	ANA	主振荡器输入连接。
OSCO	40	50	64	O	ANA	主振荡器输出连接。
PGEC1	15	19	24	I/O	ST	在线调试器 / 仿真器 /ICSP™ 编程时钟。
PGED1	16	20	25	I/O	ST	在线调试器 / 仿真器 /ICSP™ 编程数据。
PGEC2	17	21	26	I/O	ST	在线调试器 / 仿真器 /ICSP™ 编程时钟。
PGED2	18	22	27	I/O	ST	在线调试器 / 仿真器 /ICSP™ 编程数据。
PGEC3	11	15	20	I/O	ST	在线调试器 / 仿真器 /ICSP™ 编程时钟。
PGED3	12	16	21	I/O	ST	在线调试器 / 仿真器 /ICSP™ 编程数据。
PMA0	30	36	44	I/O	ST	并行主端口地址 bit 0 的输入 (带缓冲的从模式) 和输出 (主模式)。
PMA1	29	35	43	I/O	ST	并行主端口地址 bit 1 的输入 (带缓冲的从模式) 和输出 (主模式)。
PMA2	8	10	14	O	—	并行主端口地址 (非复用的主模式)。
PMA3	6	8	12	O	—	
PMA4	5	7	11	O	—	
PMA5	4	6	10	O	—	
PMA6	16	24	29	O	—	
PMA7	22	23	28	O	—	
PMA8	32	40	50	O	—	
PMA9	31	39	49	O	—	
PMA10	28	34	42	O	—	
PMA11	27	33	41	O	—	
PMA12	24	30	35	O	—	
PMA13	23	29	34	O	—	
PMCS1	45	57	71	I/O	ST/TTL	并行主端口片选 1 选通 / 地址 bit 15。
PMCS2	44	56	70	O	ST	并行主端口片选 2 选通 / 地址 bit 14。
PMBE	51	63	78	O	—	并行主端口字节使能选通。

图注: TTL = TTL 输入缓冲器
ANA = 模拟电平输入 / 输出

ST = 施密特触发器输入缓冲器
 $\text{I}^2\text{C}^\text{TM}$ = $\text{I}^2\text{C}/\text{SMBus}$ 输入缓冲器

表 1-4: PIC24FJ256GB110 系列引脚说明 (续)

功能	引脚编号			I/O	输入 缓冲器	说明
	64 引脚 TQFP 和 QFN	80 引脚 TQFP	100 引脚 TQFP			
PMD0	60	76	93	I/O	ST/TTL	并行主端口数据（非复用的主模式）或地址 / 数据（复用的主模式）。
PMD1	61	77	94	I/O	ST/TTL	
PMD2	62	78	98	I/O	ST/TTL	
PMD3	63	79	99	I/O	ST/TTL	
PMD4	64	80	100	I/O	ST/TTL	
PMD5	1	1	3	I/O	ST/TTL	
PMD6	2	2	4	I/O	ST/TTL	
PMD7	3	3	5	I/O	ST/TTL	
PMRD	53	67	82	O	—	并行主端口读选通。
PMWR	52	66	81	O	—	并行主端口写选通。
RA0	—	—	17	I/O	ST	PORTA 数字 I/O。
RA1	—	—	38	I/O	ST	
RA2	—	—	58	I/O	ST	
RA3	—	—	59	I/O	ST	
RA4	—	—	60	I/O	ST	
RA5	—	—	61	I/O	ST	
RA6	—	—	91	I/O	ST	
RA7	—	—	92	I/O	ST	
RA9	—	23	28	I/O	ST	
RA10	—	24	29	I/O	ST	
RA14	—	52	66	I/O	ST	
RA15	—	53	67	I/O	ST	
RB0	16	20	25	I/O	ST	PORTB 数字 I/O。
RB1	15	19	24	I/O	ST	
RB2	14	18	23	I/O	ST	
RB3	13	17	22	I/O	ST	
RB4	12	16	21	I/O	ST	
RB5	11	15	20	I/O	ST	
RB6	17	21	26	I/O	ST	
RB7	18	22	27	I/O	ST	
RB8	21	27	32	I/O	ST	
RB9	22	28	33	I/O	ST	
RB10	23	29	34	I/O	ST	
RB11	24	30	35	I/O	ST	
RB12	27	33	41	I/O	ST	
RB13	28	34	42	I/O	ST	
RB14	29	35	43	I/O	ST	
RB15	30	36	44	I/O	ST	

图注: TTL = TTL 输入缓冲器
ANA = 模拟电平输入 / 输出

ST = 施密特触发器输入缓冲器
 I^2C^{TM} = I^2C /SMBus 输入缓冲器

PIC24FJ256GB110 系列

表 1-4: PIC24FJ256GB110 系列引脚说明 (续)

功能	引脚编号			I/O	输入 缓冲器	说明
	64 引脚 TQFP 和 QFN	80 引脚 TQFP	100 引脚 TQFP			
RC1	—	4	6	I/O	ST	PORTC 数字 I/O。
RC2	—	—	7	I/O	ST	
RC3	—	5	8	I/O	ST	
RC4	—	—	9	I/O	ST	
RC12	39	49	63	I/O	ST	
RC13	47	59	73	I/O	ST	
RC14	48	60	74	I/O	ST	
RC15	40	50	64	I/O	ST	
RCV	18	22	27	I	ST	USB 接收输入 (来自外部收发器)。
RD0	46	58	72	I/O	ST	PORTD 数字 I/O。
RD1	49	61	76	I/O	ST	
RD2	50	62	77	I/O	ST	
RD3	51	63	78	I/O	ST	
RD4	52	66	81	I/O	ST	
RD5	53	67	82	I/O	ST	
RD6	54	68	83	I/O	ST	
RD7	55	69	84	I/O	ST	
RD8	42	54	68	I/O	ST	
RD9	43	55	69	I/O	ST	
RD10	44	56	70	I/O	ST	
RD11	45	57	71	I/O	ST	
RD12	—	64	79	I/O	ST	
RD13	—	65	80	I/O	ST	
RD14	—	37	47	I/O	ST	
RD15	—	38	48	I/O	ST	
RE0	60	76	93	I/O	ST	PORTE 数字 I/O。
RE1	61	77	94	I/O	ST	
RE2	62	78	98	I/O	ST	
RE3	63	79	99	I/O	ST	
RE4	64	80	100	I/O	ST	
RE5	1	1	3	I/O	ST	
RE6	2	2	4	I/O	ST	
RE7	3	3	5	I/O	ST	
RE8	—	13	18	I/O	ST	
RE9	—	14	19	I/O	ST	
REF0	30	36	44	O	—	参考时钟输出。

图注: TTL = TTL 输入缓冲器
ANA = 模拟电平输入 / 输出

ST = 施密特触发器输入缓冲器
 I^2C^{TM} = $I^2C/SMBus$ 输入缓冲器

表 1-4: PIC24FJ256GB110 系列引脚说明 (续)

功能	引脚编号			I/O	输入 缓冲器	说明
	64 引脚 TQFP 和 QFN	80 引脚 TQFP	100 引脚 TQFP			
RF0	58	72	87	I/O	ST	PORTF 数字 I/O。
RF1	59	73	88	I/O	ST	
RF2	—	42	52	I/O	ST	
RF3	33	41	51	I/O	ST	
RF4	31	39	49	I/O	ST	
RF5	32	40	50	I/O	ST	
RF8	—	43	53	I/O	ST	
RF12	—	—	40	I/O	ST	
RF13	—	—	39	I/O	ST	
RG0	—	75	90	I/O	ST	
RG1	—	74	89	I/O	ST	
RG2	37	47	57	I	ST	
RG3	36	46	56	I	ST	
RG6	4	6	10	I/O	ST	
RG7	5	7	11	I/O	ST	
RG8	6	8	12	I/O	ST	
RG9	8	10	14	I/O	ST	
RG12	—	—	96	I/O	ST	
RG13	—	—	97	I/O	ST	
RG14	—	—	95	I/O	ST	
RG15	—	—	1	I/O	ST	
RP0	16	20	25	I/O	ST	可重映射的外设 (输入或输出)。
RP1	15	19	24	I/O	ST	
RP2	42	54	68	I/O	ST	
RP3	44	56	70	I/O	ST	
RP4	43	55	69	I/O	ST	
RP5	—	38	48	I/O	ST	
RP6	17	21	26	I/O	ST	
RP7	18	22	27	I/O	ST	
RP8	21	27	32	I/O	ST	
RP9	22	28	33	I/O	ST	
RP10	31	39	49	I/O	ST	
RP11	46	58	72	I/O	ST	
RP12	45	57	71	I/O	ST	
RP13	14	18	23	I/O	ST	
RP14	29	35	43	I/O	ST	
RP15	—	43	53	I/O	ST	
RP16	33	41	51	I/O	ST	
RP17	32	40	50	I/O	ST	
RP18	11	15	20	I/O	ST	
RP19	6	8	12	I/O	ST	

图注: TTL = TTL 输入缓冲器
ANA = 模拟电平输入 / 输出

ST = 施密特触发器输入缓冲器
 $\text{I}^2\text{C}^{\text{TM}}$ = I^2C /SMBus 输入缓冲器

PIC24FJ256GB110 系列

表 1-4: PIC24FJ256GB110 系列引脚说明 (续)

功能	引脚编号			I/O	输入 缓冲器	说明
	64 引脚 TQFP 和 QFN	80 引脚 TQFP	100 引脚 TQFP			
RP20	53	67	82	I/O	ST	可重映射的外设 (输入或输出)。
RP21	4	6	10	I/O	ST	
RP22	51	63	78	I/O	ST	
RP23	50	62	77	I/O	ST	
RP24	49	61	76	I/O	ST	
RP25	52	66	81	I/O	ST	
RP26	5	7	11	I/O	ST	
RP27	8	10	14	I/O	ST	
RP28	12	16	21	I/O	ST	
RP29	30	36	44	I/O	ST	
RP30	—	42	52	I/O	ST	
RP31	—	—	39	I/O	ST	
RPI32	—	—	40	I	ST	可重映射的外设 (仅输入)。
RPI33	—	13	18	I	ST	
RPI34	—	14	19	I	ST	
RPI35	—	53	67	I	ST	
RPI36	—	52	66	I	ST	
RPI37	48	60	74	I	ST	
RPI38	—	4	6	I	ST	
RPI39	—	—	7	I	ST	
RPI40	—	5	8	I	ST	
RPI41	—	—	9	I	ST	
RPI42	—	64	79	I	ST	
RPI43	—	37	47	I	ST	
RTCC	42	54	68	O	—	实时时钟闹钟 / 秒脉冲输出。
SCL1	44	56	66	I/O	I ² C	I ² C1 同步串行时钟输入 / 输出。
SCL2	32	52	58	I/O	I ² C	I ² C2 同步串行时钟输入 / 输出。
SCL3	2	2	4	I/O	I ² C	I ² C3 同步串行时钟输入 / 输出。
SDA1	43	55	67	I/O	I ² C	I ² C1 数据输入 / 输出。
SDA2	31	53	59	I/O	I ² C	I ² C2 数据输入 / 输出。
SDA3	3	3	5	I/O	I ² C	I ² C3 数据输入 / 输出。
SOSCI	47	59	73	I	ANA	辅助振荡器 /Timer1 时钟输入。
SOSCO	48	60	74	O	ANA	辅助振荡器 /Timer1 时钟输出。
T1CK	48	60	74	I	ST	Timer1 时钟。
TCK	27	33	38	I	ST	JTAG 测试时钟 / 编程时钟输入。
TDI	28	34	60	I	ST	JTAG 测试数据 / 编程数据输入。
TDO	24	14	61	O	—	JTAG 测试数据输出。
TMS	23	13	17	I	ST	JTAG 测试模式选择输入。
USBID	33	41	51	I	ST	USB OTG ID (仅限 OTG 模式)。
USBOEN	12	16	21	O	—	USB 输出使能控制 (适用于外部收发器)。

图注: TTL = TTL 输入缓冲器
ANA = 模拟电平输入 / 输出

ST = 施密特触发器输入缓冲器
I²CTM = I²C/SMBus 输入缓冲器

表 1-4: PIC24FJ256GB110 系列引脚说明 (续)

功能	引脚编号			I/O	输入缓冲器	说明
	64 引脚 TQFP 和 QFN	80 引脚 TQFP	100 引脚 TQFP			
V _{BUS}	34	44	54	P	—	USB 电压, 主机模式 (5V)。
V _{BUSON}	11	15	20	O	—	USB OTG 外部电荷泵控制。
V _{BUSST}	58	72	87	I	ANA	USB OTG 内部电荷泵反馈控制。
V _{CAP}	56	70	85	P	—	外部滤波电容连接 (稳压器使能)。
V _{CMPST1}	58	72	87	I	ST	USB V _{BUS} 升压发生器, 比较器输入 1。
V _{CMPST2}	59	73	88	I	ST	USB V _{BUS} 升压发生器, 比较器输入 2。
V _{CPCON}	49	61	76	O	—	USB OTG V _{BUS} PWM/ 电荷输出。
V _{DD}	10, 26, 38	12, 32, 48	2, 16, 37, 46, 62	P	—	外设数字逻辑和 I/O 引脚的正电源。
V _{DDCORE}	56	70	85	P	—	单片机内核逻辑的正电源 (稳压器禁止)。
V _{MIO}	14	18	23	I/O	ST	USB 差分负输入 / 输出 (外部收发器)。
V _{PIO}	13	17	22	I/O	ST	USB 差分正输入 / 输出 (外部收发器)。
V _{REF-}	15	23	28	I	ANA	A/D 和比较器低参考电压 (低电压) 输入。
V _{REF+}	16	24	29	I	ANA	A/D 和比较器参考电压 (高电压) 输入。
V _{SS}	9, 25, 41	11, 31, 51	15, 36, 45, 65, 75	P	—	逻辑和 I/O 引脚的参考地。
V _{USB}	35	45	55	P	—	USB 电压 (3.3V)。

图注: TTL = TTL 输入缓冲器
ANA = 模拟电平输入 / 输出

ST = 施密特触发器输入缓冲器
 I^2C^{TM} = I^2C /SMBus 输入缓冲器

PIC24FJ256GB110 系列

注:

2.0 16 位单片机入门指南

2.1 基本连接要求

在着手使用 16 位单片机 PIC24FJ256GB110 系列进行开发之前，要求注意器件引脚的最少连接。

以下引脚必须始终处于连接状态：

- 所有 VDD 和 VSS 引脚（见第 2.2 节“电源引脚”）
- 所有 AVDD 和 AVSS 引脚，而与是否使用模拟器件功能无关（见第 2.2 节“电源引脚”）
- MCLR 引脚（见第 2.3 节“主复位（MCLR）引脚”）
- ENVREG/DISVREG 和 VCAP/VDDCORE 引脚（仅 PIC24FJ 器件）（见第 2.4 节“稳压器引脚（ENVREG/DISVREG 和 VCAP/VDDCORE）”）。

如果在最终应用中使用了以下引脚，那么也必须连接这些引脚：

- 用于在线串行编程（ICSP™）和调试的 PGECx/PGEDx 引脚（见第 2.5 节“ICSP 引脚”）
- 使用外部振荡器源时用到的 OSC1 和 OSCO 引脚（见第 2.6 节“外部振荡器引脚”）

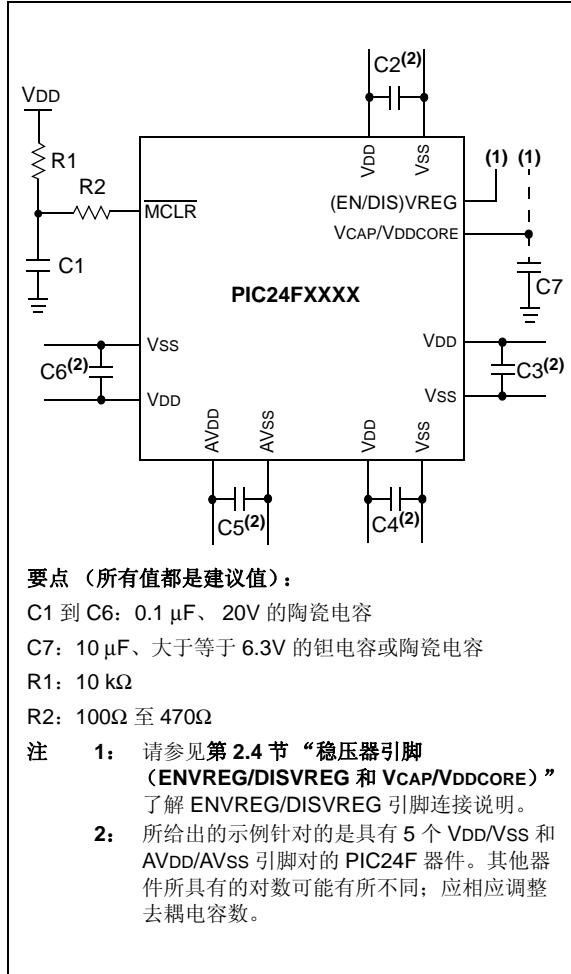
此外，可能需要以下引脚：

- 实现模拟模块的外部参考电压时使用的 VREF+/VREF- 引脚

注：无论是否使用任何模拟模块，AVDD 和 AVSS 引脚都必须始终处于已连接状态。

图 2-1 中给出了必需的最少连接。

图 2-1：建议的最少连接



2.2 电源引脚

2.2.1 去耦电容

需要在每对电源引脚（如 VDD 与 VSS，以及 AVDD 与 AVSS）上使用去耦电容。

使用去耦电容时，需考虑以下条件：

- 电容的值和类型：**建议使用 $0.1 \mu\text{F}$ (100 nF)、10-20V 的电容。应使用谐振频率在 200 MHz 及更高范围内的低 ESR 电容。建议使用陶瓷电容。
- 印刷电路板上的放置：**去耦电容应尽可能靠近相应的引脚放置。建议将电容放在电路板上与器件相同的一侧。如果空间有限，可使用过孔将电容放到 PCB 的另一层上；但是，需要确保从引脚到电容的走线长度不超过 0.25 英寸 (6 mm)。
- 处理高频噪声：**如果电路板产生高达几十兆赫兹的高频噪声，请在上述去耦电容旁并联一个陶瓷类型的辅助电容。辅助电容值的范围为 $0.01 \mu\text{F}$ 至 $0.001 \mu\text{F}$ 。请将这个辅助电容靠近每个主去耦电容放置。在高速电路设计中，请考虑在尽可能靠近电源和接地引脚的地方布置十对电容（例如，一个 $0.1 \mu\text{F}$ 的电容与一个 $0.001 \mu\text{F}$ 的电容并联构成一对）。
- 最大程度提高性能：**在从电源电路布置电路板时，请首先连接电源并将走线返回到去耦电容，然后再连接到器件引脚。这可以确保去耦电容在电源链中处于第一位。保持电容和电源引脚之间走线长度尽可能短也同样重要，因为这可以减少 PCB 的走线感抗。

2.2.2 槽路电容

在电源走线长度大于 6 英寸的电路板上，建议对包括单片机在内的集成电路使用槽路电容，以提供本地电源。槽路电容的值应根据以下因素确定：连接电源和器件的走线的电阻值以及应用中器件消耗的最大电流。也就是说，选择槽路电容使之满足器件的可接受的电压骤降要求。典型值范围为 $4.7 \mu\text{F}$ 到 $47 \mu\text{F}$ 。

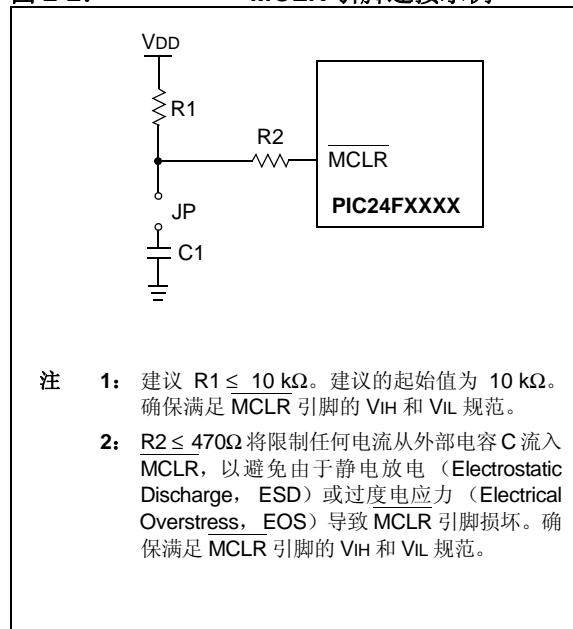
2.3 主复位 (MCLR) 引脚

MCLR 引脚提供两个特定的器件功能：器件复位及器件编程与调试。如果在最终应用中不需要编程和调试，则只需直接将该引脚连接到 VDD。为帮助增加应用的电阻以避免因电压骤降而意外复位，可能需要添加其他元件。图 2-1 给出了典型的配置。可根据应用的要求实现其他电路设计。

在编程和调试期间，必须考虑可以添加到引脚的电阻和电容。器件编程器和调试器驱动 MCLR 引脚。因此，不允许对特殊电压电平 (VIH 和 Vil) 以及快速信号变化造成不良影响。这就需要根据应用和 PCB 的要求调整 R1 和 C1 的具体值。例如，建议在编程和调试操作期间，通过使用跳线将电容 C1 和 MCLR 引脚隔离（图 2-2）。正常的运行时操作不需要这样的跳线。

与 MCLR 引脚相关的任何元件应放置在距离该引脚 0.25 英寸 (6 mm) 的范围内。

图 2-2: MCLR 引脚连接示例



- 注
- 1: 建议 $R1 \leq 10 \text{ k}\Omega$ 。建议的起始值为 $10 \text{ k}\Omega$ 。确保满足 MCLR 引脚的 VIH 和 Vil 规范。
 - 2: $R2 \leq 470\Omega$ 将限制任何电流从外部电容 C 流入 MCLR，以避免由于静电放电 (Electrostatic Discharge, ESD) 或过度应力 (Electrical Overstress, EOS) 导致 MCLR 引脚损坏。确保满足 MCLR 引脚的 VIH 和 Vil 规范。

2.4 稳压器引脚 (ENVREG/DISVREG 和 VCAP/VDDCORE)

注: 本节仅适用于具有片上稳压器的 PIC24FJ 器件。

片上稳压器使能 / 禁止引脚 (ENVREG 或 DISVREG, 具体取决于器件系列) 必须始终直接连接到电源电压或接地。具体连接取决于是否使用稳压器：

- 对于 ENVREG, 连接到 VDD 使能稳压器, 接地禁止稳压器
- 对于 DISVREG, 接地使能稳压器, 连接到 VDD 禁止稳压器

连接和使用片上稳压器的详情, 请参见第 26.2 节 “片上稳压器”。

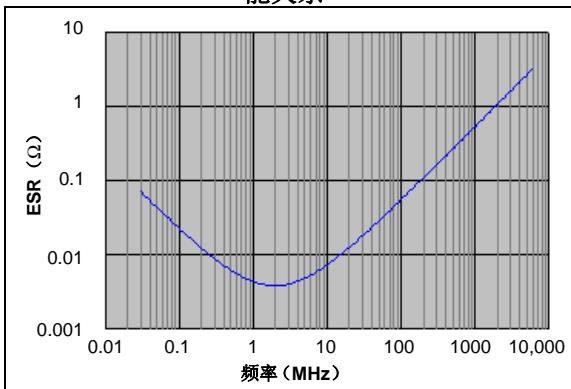
使能稳压器后, VCAP/VDDCORE 引脚上需要连接一个低 ESR (<5 Ω) 的电容来稳定稳压器输出电压。VCAP/VDDCORE 引脚不能连接到 VDD, 并且必须使用一个 10 μF 的电容接地。电容类型可以是陶瓷电容或者钽电容。Murata GRM21BF50J106ZE01 (10 μF, 6.3V) 或同等规格的电容就很适合。设计人员可以使用图 2-3 来计算备选电容的 ESR。

该电容应靠近 VCAP/VDDCORE 放置。建议走线长度不超过 0.25 英寸 (6 mm)。更多信息请参见第 29.0 节 “电气特性”。

当稳压器被禁止时, VCAP/VDDCORE 引脚必须连接 VDDCORE 电平的电源。有关 VDD 和 VDDCORE 的信息, 请参见第 29.0 节 “电气特性”。

图 2-3:

建议 VCAP 的频率与 ESR 性能关系



注: 给出的是 Murata GRM21BF50J106ZE01 的数据。
25°C、0 V 直流偏置电压下测量。

2.5 ICSP 引脚

PGECx 和 PGEDx 引脚用于在线串行编程 (ICSP™) 和调试。建议保持器件上的 ICSP 连接器和 ICSP 引脚之间的走线长度尽可能短。如果预期 ICSP 连接器上会发生 ESD 事件, 建议使用一个串联电阻, 且电阻值在几十欧姆范围内, 不要超过 100 Ω。

建议不要在 PGECx 和 PGEDx 引脚上使用上拉电阻、串联二极管和电容, 因为它们会干扰编程器 / 调试器与器件之间的通信。如果应用需要这类分立元件, 应在编程和调试期间将它们从电路中除去。或者, 参阅相关器件闪存编程规范中的交流 / 直流特性和时序要求信息, 以了解关于容性负载限制以及引脚高输入电压 (VIH) 和低输入电压 (VIL) 要求的信息。

对于器件仿真, 请确保编程到器件中的“通信通道选择”(即, PGECx/PGEDx 引脚) 与 ICSP 到 Microchip 调试器 / 仿真器工具的物理连接一致。

关于可用 Microchip 开发工具连接要求的更多信息, 请参见第 27.0 节 “开发支持”。

PIC24FJ256GB110 系列

2.6 外部振荡器引脚

许多单片机都至少有两个振荡器选项：高频主振荡器和低频辅助振荡器（详情请参见第 8.0 节“振荡器配置”）。

振荡器电路应放到电路板上与器件相同的一侧。将振荡器电路靠近相关振荡器引脚放置，且电路元件与引脚之间的距离不要超过 0.5 英寸（12 mm）。负载电容应在电路板的同一侧挨着振荡器本身放置。

在振荡器电路周围设置接地覆铜区，将其与周围电路隔离。接地覆铜区应直接连接到 MCU 地。不要在接地覆铜区内使用任何信号线或电源线。而且，如果使用双面的电路板，请避免在电路板上放置晶振的另一面布线。

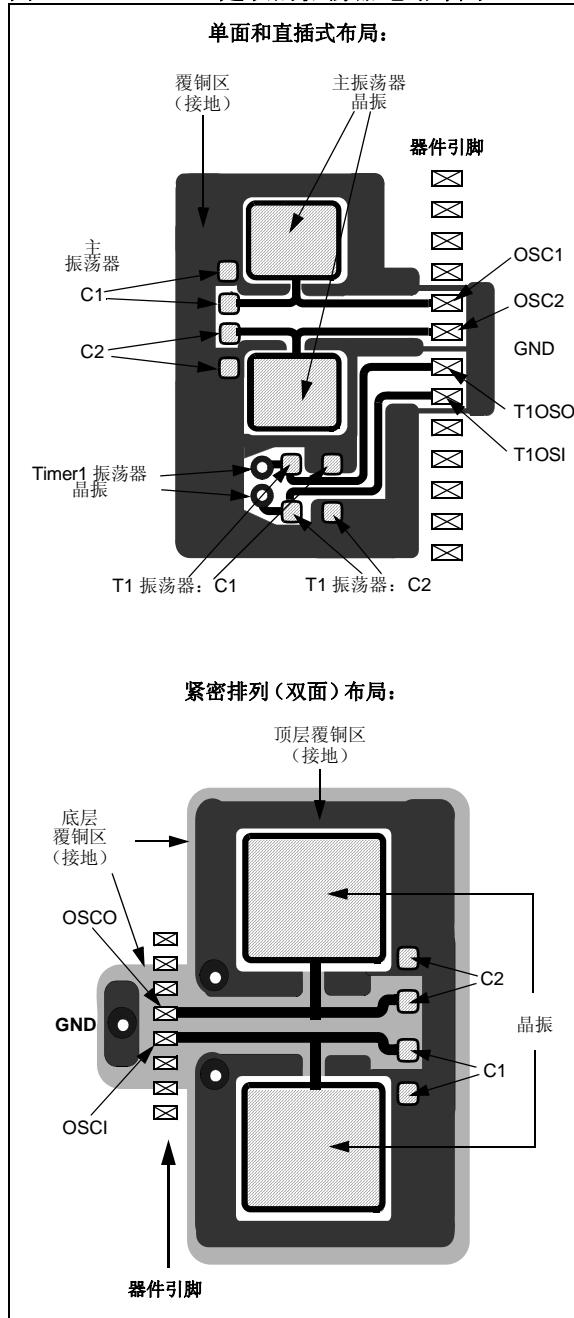
图 2-4 给出了建议的电路板布局。可以使用包含完整振荡器引脚的单面布局来处理直插式封装器件。对于引脚排列紧密的器件，单面布局则可能无法始终完全包含所有的引脚和元件。适当的解决方案是将含有保护走线的部分连接到反面的接地层。在所有情况下，保护走线必须返回到地。

在规划应用的走线和 I/O 分配过程中，确保临近的端口引脚和离振荡器非常近的其他信号状态良好（即，没有很高的频率、陡升和陡降以及其他类似的噪声）。

关于振荡器电路的其他信息和设计指导，请参考我公司网站（www.microchip.com）上提供的以下应用笔记：

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

图 2-4: 建议的振荡器电路布局



2.7 ICSP 工作期间的模拟和数字引脚配置

如果将兼容 ICSP 的仿真器选择为调试器，它会自动初始化所有 A/D 输入引脚 (ANx) 为“数字”引脚。根据具体器件，可通过将 ADnPCFG 寄存器中的所有位置 1 或清零 ANSx 寄存器的所有位来实现。

所有 PIC24F 器件将具有一个或多个 ADnPCFG 寄存器或几个 ANSx 寄存器（每个端口一个），但没有器件同时具有这两者。更多详细信息，请参见第 22.0 节“**10 位高速 A/D 转换器**”。

用户应用固件不得更改此寄存器中与初始化仿真器的 A/D 引脚相对应的位；否则将导致调试器和器件之间发生通信错误。

如果在调试会话期间应用需要使用某些 A/D 引脚作为模拟输入引脚，那么用户应用必须在 ADC 模块初始化期间修改相应位，具体如下：

- 对于具有 ADnPCFG 寄存器的器件，清零引脚相对应的位以将其配置为模拟引脚。在任何时候都不要更改任何其他位，特别是与 PGECx/PGEDx 对应的位。
- 对于具有 ANSx 寄存器的器件，置 1 引脚相对应的位以将其配置为模拟引脚。在任何时候都不要更改任何其他位，特别是与 PGECx/PGEDx 对应的位。

当 Microchip 调试器 / 仿真器用作编程器时，用户应用固件必须正确配置 ADnPCFG 或 ANSx 寄存器。仅在调试器操作期间自动初始化此寄存器。如果未能正确配置该寄存器，将导致所有 A/D 引脚被识别为模拟输入引脚，以致端口值被读为逻辑 0，从而可能影响用户应用的功能。

2.8 未使用的 I/O

未使用的 I/O 引脚应配置为输出，并驱动为逻辑低电平状态。或者，使用一个 $1\text{ k}\Omega$ 至 $10\text{ k}\Omega$ 电阻将未使用的引脚连接至 V_{SS}，并驱动输出为逻辑低电平。

PIC24FJ256GB110 系列

注:

3.0 CPU

注：本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》的第 2 章“CPU”(DS39703A_CN)。

PIC24F CPU 采用 16 位（数据）的改进型哈佛架构，具有增强指令集以及带有长度可变操作码字段的 24 位指令字。程序计数器（Program Counter, PC）为 23 位宽，可以寻址最大 4M 指令字的用户程序存储空间。单周期指令预取机制可帮助维持吞吐量并使指令的执行具有预测性。除了改变程序流的指令、双字传送（MOV.D）指令和表指令以外，所有指令都在单个周期内执行。使用 REPEAT 指令支持无开销的程序循环结构，该指令在任何时间都可以被中断。

PIC24F 器件在编程模型中有 16 个 16 位工作寄存器。每个工作寄存器都可以充当数据、地址或地址偏移量寄存器。第 16 个工作寄存器（W15）作为软件堆栈指针，用于中断和调用。

可以选择将数据存储空间的高 32 KB 映射到由 8 位程序空间可视性页地址（Program Space Visibility Page Address, PSVPAG）寄存器定义的任何 16K 字边界内的程序空间内。程序空间到数据空间的映射功能让任何指令都能象访问数据空间一样访问程序空间。

指令集架构（Instruction Set Architecture, ISA）在 PIC18 的指令集架构基础上显著增强了，但仍保持了相当程度的向后兼容性。直接支持或通过简单的宏支持所有 PIC18 指令和寻址模式。许多 ISA 增强功能都是为了提高编译器效率。

内核支持固有（无操作数）、相对、立即数、存储器直接和三组寻址模式。所有模式都支持寄存器直接寻址和多种寄存器间接寻址模式。每组都提供最多 7 种寻址模式。指令根据其功能要求，与预定义的寻址模式相关联。

对于大多数指令，内核能够在每个指令周期内执行一次数据（或程序数据）存储器读取、一次工作寄存器（数据）读取、一次数据存储器写入以及一次程序（指令）存储器读取操作。因此，支持 3 操作数指令，允许在单个周期内执行三操作数运算（即， $A + B = C$ ）。

包含了一个高速 17 位 \times 17 位乘法器，显著提高了内核算术能力和吞吐量。乘法器支持有符号、无符号和混合模式的 16 位 \times 16 位或 8 位 \times 8 位整数乘法。所有乘法指令都在一个周期内执行。

16 位 ALU 用支持迭代非还原除法算法的整数除法支持硬件进行了增强。它和 REPEAT 指令循环机制一起（还可以加上迭代除法指令）使用，支持用 16 位整数除 32 位（或 16 位）的有符号和无符号除法。所有除法运算都需要 19 个周期完成，但可以在任何周期边界上中断。

PIC24F 有一个向量异常机制，具有最多 8 个不可屏蔽陷阱源和最多 118 个中断源。每个中断源都可分配为 7 个优先级之一。

图 3-1 给出了 CPU 的框图。

3.1 编程模型

图 3-2 给出了 PIC24F 的编程模型。编程模型中的所有寄存器都是存储器映射的，可以直接用指令控制。表 3-1 中提供了每个寄存器的说明。所有与编程模型相关的寄存器都是存储器映射的。

PIC24FJ256GB110 系列

图 3-1: PIC24F CPU 内核框图

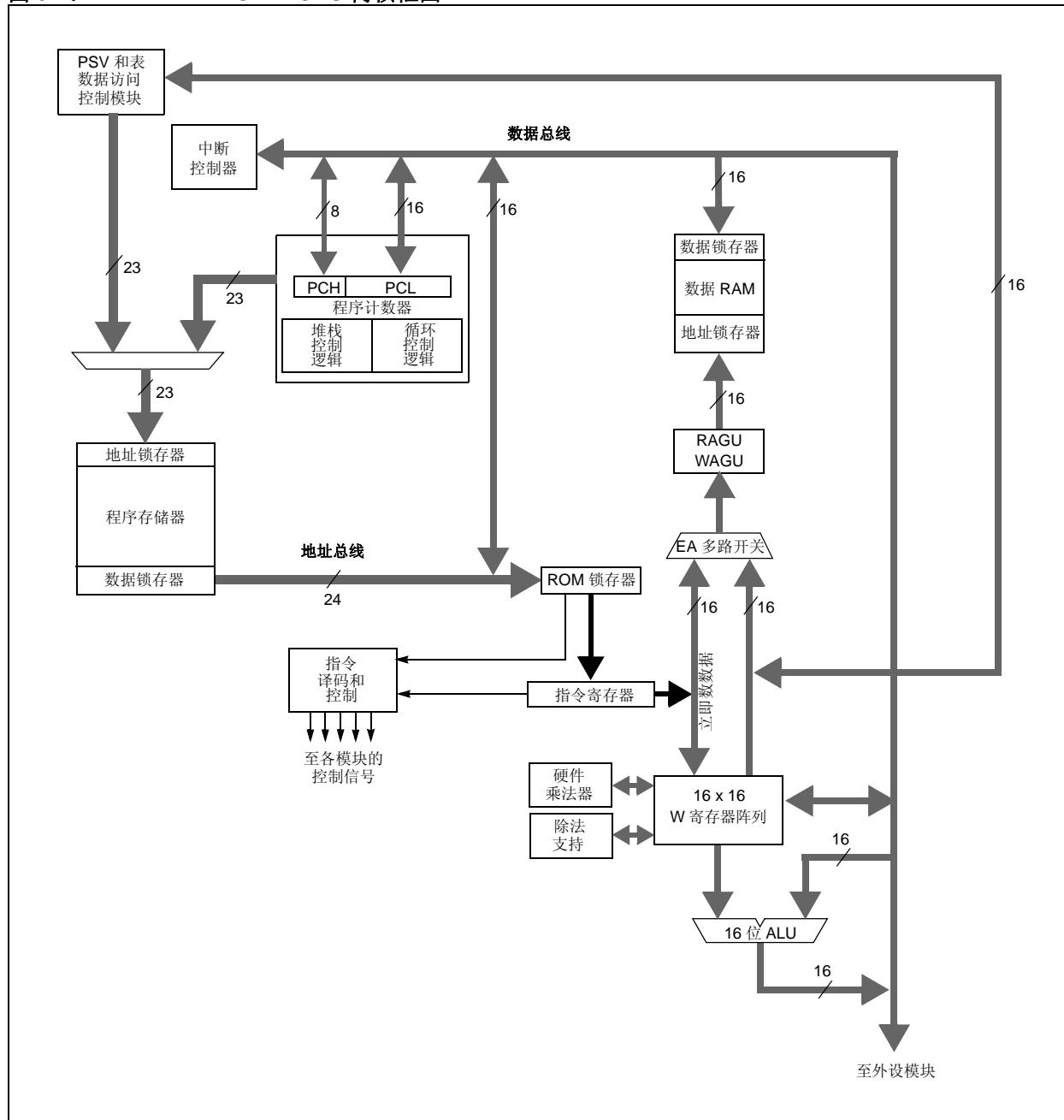
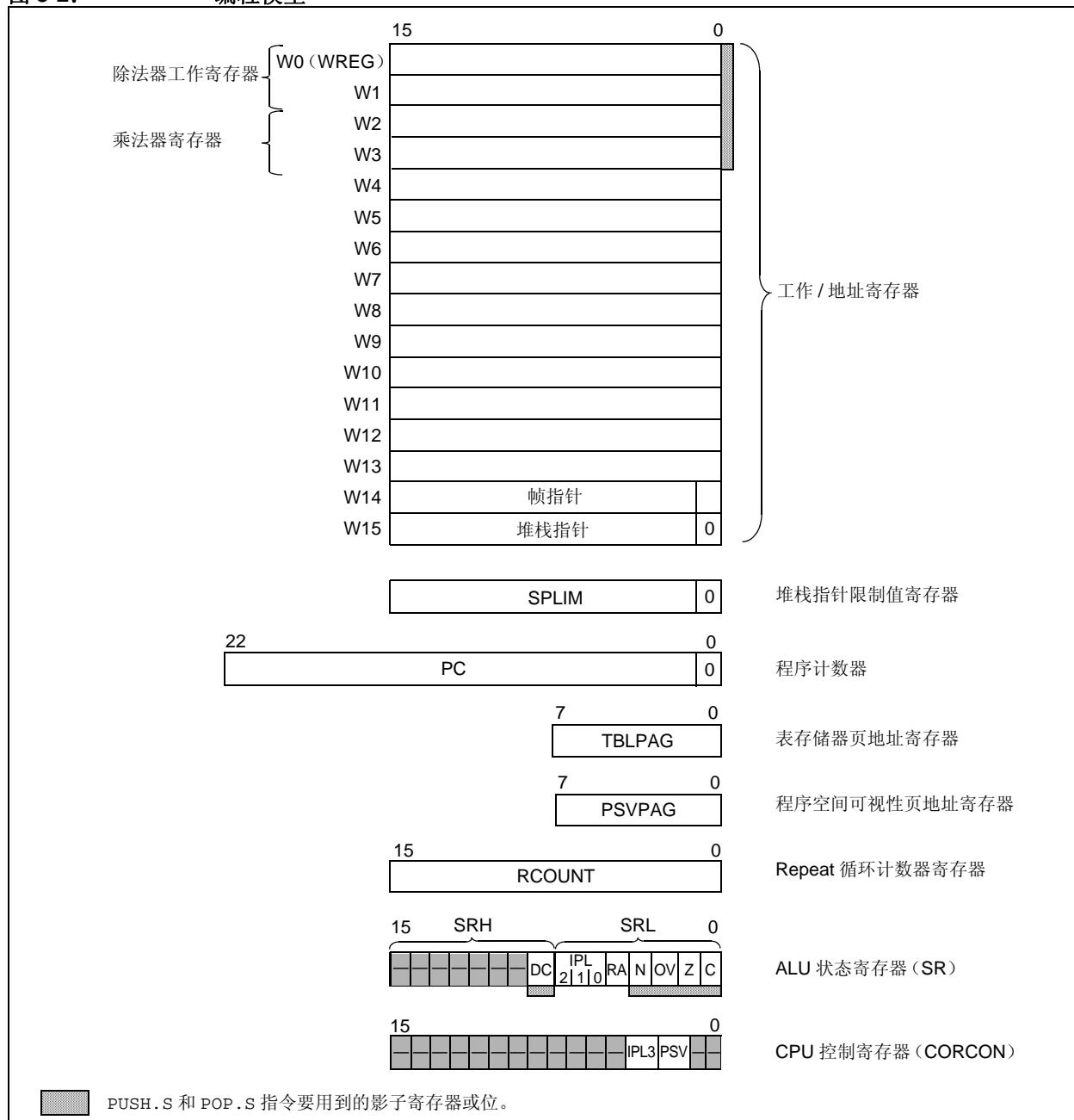


表 3-1: CPU 内核寄存器

寄存器名称	说明
W0 到 W15	工作寄存器阵列
PC	23 位程序计数器
SR	ALU 状态寄存器
SPLIM	堆栈指针限制值寄存器
TBLPAG	表存储器页地址寄存器
PSVPAG	程序空间可视性页地址寄存器
RCOUNT	Repeat 循环计数器寄存器
CORCON	CPU 控制寄存器

图 3-2: 编程模型



PIC24FJ256GB110 系列

3.2 CPU 控制寄存器

寄存器 3-1: SR: ALU 状态寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	DC
bit 15							bit 8

R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2 ⁽²⁾	IPL1 ⁽²⁾	IPL0 ⁽²⁾	RA	N	OV	Z	C
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-9 未实现: 读为 0

bit 8 DC: ALU 半进位 / 借位标志位

1 = 结果的第 4 个低位 (对于字节大小的数据) 或第 8 个低位 (对于字大小的数据) 发生了进位
0 = 结果的第 4 个或第 8 个低位未发生进位

bit 7-5 IPL<2:0>: CPU 中断优先级状态位 ^(1,2)

111 = CPU 中断优先级为 7 (15); 禁止用户中断
110 = CPU 中断优先级为 6 (14)
101 = CPU 中断优先级为 5 (13)
100 = CPU 中断优先级为 4 (12)
011 = CPU 中断优先级为 3 (11)
010 = CPU 中断优先级为 2 (10)
001 = CPU 中断优先级为 1 (9)
000 = CPU 中断优先级为 0 (8)

bit 4 RA: REPEAT 循环活动位

1 = 正在进行 REPEAT 循环
0 = 不在进行 REPEAT 循环

bit 3 N: ALU 负标志位

1 = 结果为负
0 = 结果为非负 (零或正值)

bit 2 OV: ALU 溢出标志位

1 = 有符号 (二进制补码) 算术运算中发生溢出 (本次运算)
0 = 未发生溢出

bit 1 Z: ALU 全零标志位

1 = 影响 Z 位的任何运算在过去某时已将该位置 1
0 = 影响 Z 位的最近一次运算已将该位清零 (即运算结果非零)

bit 0 C: ALU 进位 / 借位标志位

1 = 结果的最高位发生了进位
0 = 结果的最高位未发生进位

注 1: 当 NSTDIS (INTCON1<15>) = 1 时, IPL 状态位是只读的。

2: IPL 状态位与 IPL3 位 (CORCON<3>) 组合形成 CPU 中断优先级 (Interrupt Priority Level, IPL)。当 IPL3 = 1 时, 括号中的值表示 IPL。

寄存器 3-2: CORCON: CPU 控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0	R/W-0	U-0	U-0
—	—	—	—	IPL3 ⁽¹⁾	PSV	—	—
bit 7							bit 0

图注:

C = 可清零位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-4 未实现: 读为 0

bit 3 **IPL3:** CPU 中断优先级状态位⁽¹⁾

1 = CPU 中断优先级大于 7

0 = CPU 中断优先级等于或小于 7

bit 2 **PSV:** 数据空间中程序空间可视性使能位

1 = 程序空间在数据空间中可视

0 = 程序空间在数据空间中不可视

bit 1-0 未实现: 读为 0

注 1: 当 IPL3 = 1 时, 禁止用户中断。

3.3 算术逻辑单元 (ALU)

PIC24F ALU 为 16 位宽, 并能进行加法、减法、移位和逻辑运算。除非特别指明, 算术运算一般采用二进制补码。根据不同的运算, ALU 可能会影响 SR 寄存器中的进位标志位 (C)、全零标志位 (Z)、负标志位 (N)、溢出标志位 (OV) 和半进位标志位 (DC) 的值。在减法运算中, C 和 DC 状态位分别作为借位位和半借位位。

根据所使用的指令模式, ALU 可执行 8 位或 16 位运算。根据指令的寻址模式, ALU 运算的数据可以来自 W 寄存器阵列或数据存储器。同样, ALU 的输出数据可被写入 W 寄存器阵列或数据存储单元。

PIC24F CPU 融入了对乘法和除法的硬件支持。它带有专门的硬件乘法器以及支持 16 位除数除法的硬件。

3.3.1 乘法器

ALU 包含一个高速 17 位 x 17 位乘法器。它支持几种乘法模式下的无符号、有符号或混合符号运算:

1. 16 位 x 16 位有符号
2. 16 位 x 16 位无符号
3. 16 位有符号 x 5 位 (立即数) 无符号
4. 16 位无符号 x 16 位无符号
5. 16 位无符号 x 5 位 (立即数) 无符号
6. 16 位无符号 x 16 位有符号
7. 8 位无符号 x 8 位无符号

3.3.2 除法器

除法模块支持具有以下数据长度的有符号和无符号整数除法运算：

1. 32 位有符号 /16 位有符号除法
2. 32 位无符号 /16 位无符号除法
3. 16 位有符号 /16 位有符号除法
4. 16 位无符号 /16 位无符号除法

所有除法指令的商都被放在 W0 中，余数放在 W1 中。
16 位有符号和无符号 DIV 指令可为 16 位除数指定任一 W 寄存器 (Wn)，为 32 位被除数指定任意两个连续的 W 寄存器 (W(m + 1):Wm)。除法运算中处理除数的每一位需要一个周期，因此 32 位 /16 位和 16 位 /16 位指令的执行周期数相同。

表 3-2： 使用单位和多位移位操作的指令

指令	说明
ASR	将源寄存器算术右移 1 位或多位。
SL	将源寄存器左移 1 位或多位。
LSR	将源寄存器逻辑右移 1 位或多位。

3.3.3 多位移位支持

PIC24F ALU 支持单位和单周期、多位算术和逻辑移位。多位移位使用移位器模块实现，能够在单个周期内执行最多 15 位的算术右移或最多 15 位的左移。所有多位移位指令都只支持操作数源和结果目标的寄存器直接寻址模式。

下面的表 3-2 提供了使用移位操作的指令的完整汇总。

4.0 存储器构成

作为哈佛架构器件，PIC24F 单片机具有独立的程序和数据存储空间以及总线。这一架构同时还允许在代码执行过程中从数据空间直接访问程序存储器。

4.1 程序地址空间

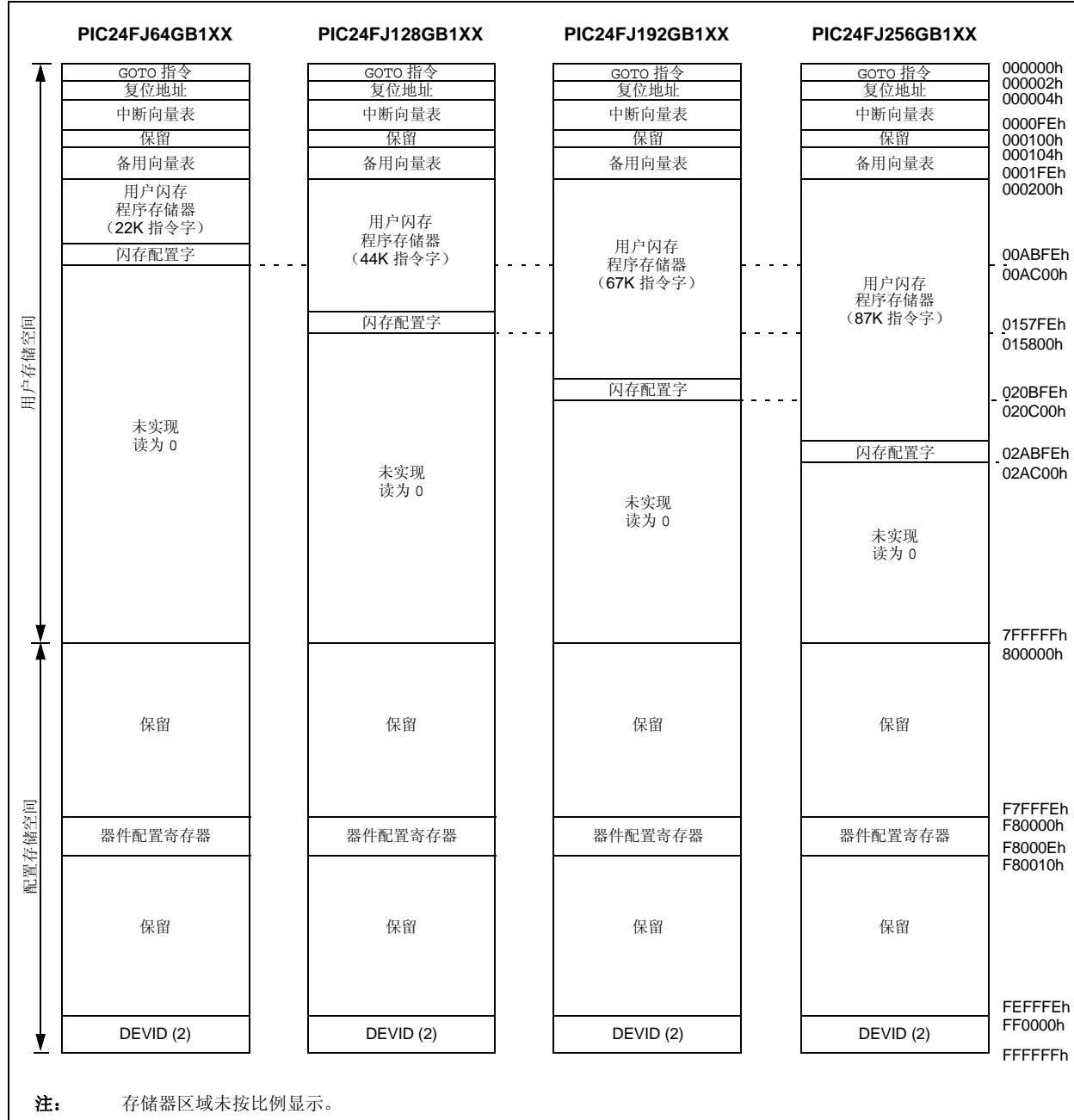
PIC24FJ256GB110 系列器件的程序地址存储空间可存储 4M 个指令字。可通过由程序执行过程中 23 位程序计

数器（PC）和第 4.3 节“程序存储空间与数据存储空间的接口”中所述的表操作或数据空间重映射得到的 24 位值寻址这一空间。

用户只能访问程序存储空间的低半地址部分（地址范围为 000000h 至 7FFFFFh）。使用 TBLRD/TBLWT 指令时，情况有所不同，这两条指令采用 TBLPAG<7> 以允许访问配置存储空间中的配置位和器件 ID。

图 4-1 给出了 PIC24FJ256GB110 系列器件的存储器映射。

图 4-1：PIC24FJ256GB110 系列器件的程序存储空间映射



PIC24FJ256GB110 系列

4.1.1 程序存储器构成

程序存储空间由可字寻址的块构成。虽然它被视为 24 位宽，但将程序存储器的每个地址视作一个低位字和一个高位字的组合更加合理，其中高位字的高字节部分没有实现。低位字的地址始终为偶数，而高位字的地址为奇数（图 4-2）。

程序存储器地址始终在低位字处按字对齐，并且在代码执行过程中地址将递增或递减 2。这种寻址模式与数据存储空间寻址兼容，且为访问程序存储空间中的数据提供了可能。

4.1.2 存储器硬编码向量

所有 PIC24F 器件中从 00000h 到 000200h 之间的地址空间都是保留的，用来存储硬编码的程序执行向量。提供了一个硬件复位向量将代码执行从器件复位时 PC 的默认值重新定位到代码实际起始处。用户可在地址 000000h 处编写一条 GOTO 指令以将代码的实际起始地址设置为 000002h。

PIC24F 器件也具有两个中断向量表，地址分别为从 000004h 到 0000FFh 和 000100h 到 0001FFh。这两个向量表允许使用不同的 ISR 处理每个器件中断源。关于中断向量表更详细的讨论，请参见第 7.1 节“中断向量表”。

4.1.3 闪存配置字

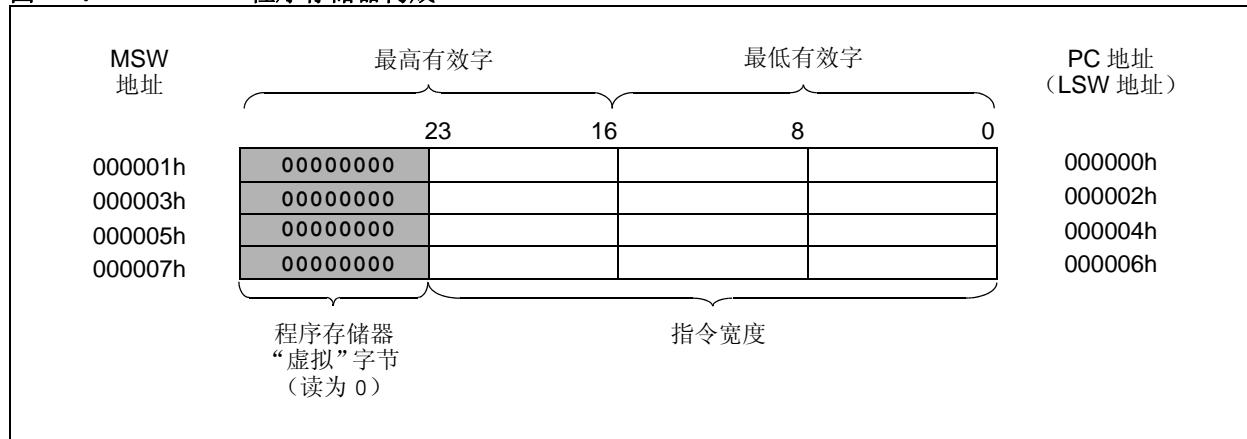
在 PIC24FJ256GB110 系列器件中，保留片上程序存储器顶部的 3 个字来保存配置信息。器件复位时，该配置信息被复制到相应的配置寄存器中。PIC24FJ256GB110 系列器件的闪存配置字的地址如表 4-1 所示。图 4-1 给出了闪存配置字以及其他存储器向量在存储器映射中的位置。

程序存储器中的配置字为紧凑格式。实际的配置位被映射到配置存储空间中的几个不同寄存器中。它们在闪存配置字中的顺序并不反映在配置空间中的对应排列。第 26.1 节“配置位”中提供了有关器件配置字的更多详细信息。

表 4-1： PIC24FJ256GB110 系列器件的闪存配置字

器件	程序存储器 (字数)	配置字地址
PIC24FJ64GB	22,016	00ABFAh: 00ABFEh
PIC24FJ128GB	44,032	0157FAh: 0157FEh
PIC24FJ192GB	67,072	020BFAh: 020BFEh
PIC24FJ256GB	87,552	02ABFAh: 02ABFEh

图 4-2： 程序存储器构成



4.2 数据地址空间

PIC24F 内核具有独立的 16 位宽数据存储空间，可将其作为一个线性空间寻址。使用两个地址发生单元（Address Generation Unit, AGU）对数据空间进行寻址，分别用于读操作和写操作。数据存储空间映射如图 4-3 所示。

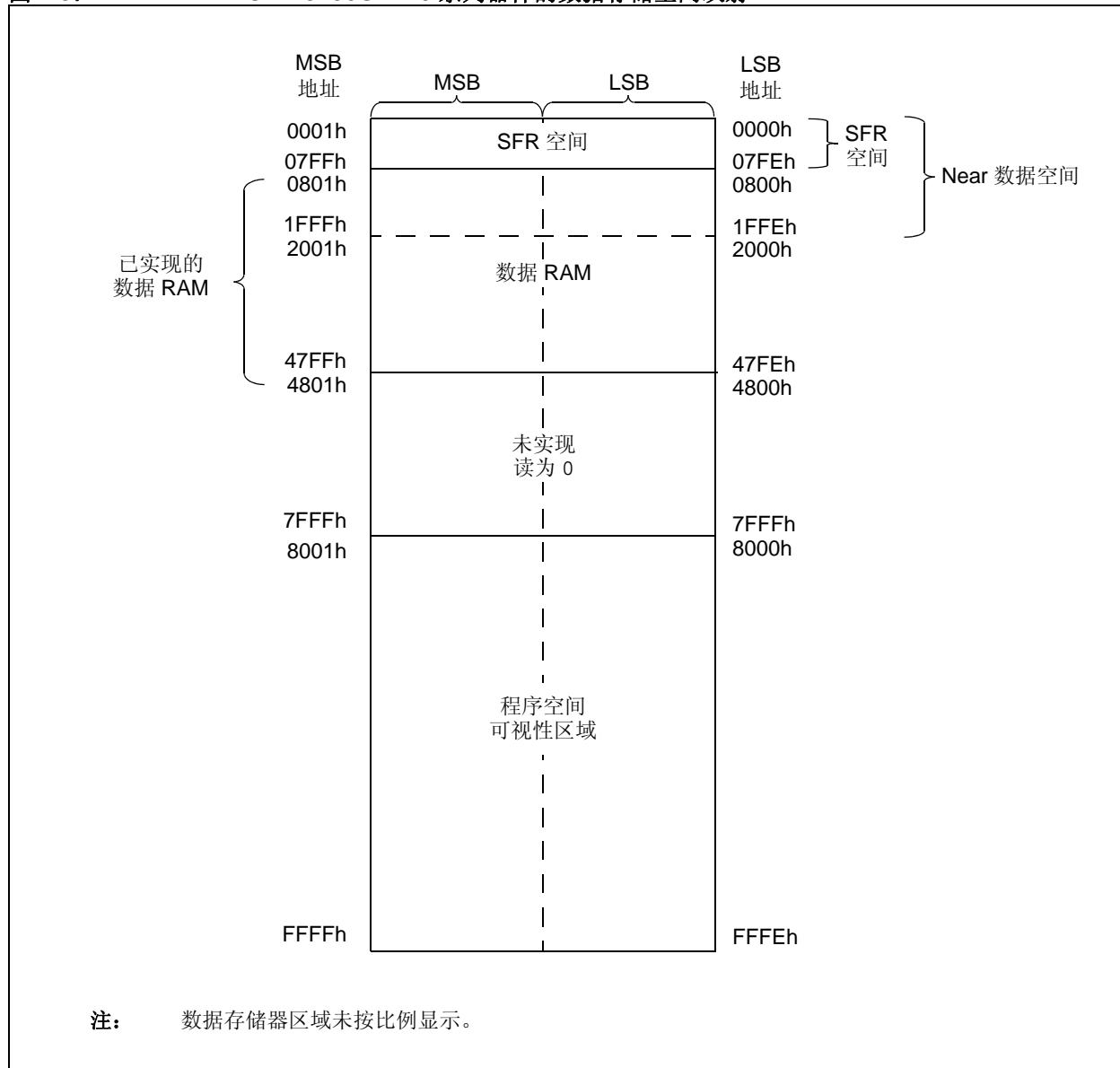
数据存储空间中的所有有效地址（Effective Address, EA）均为 16 位宽，并且指向数据空间内的字节。这使数据空间的地址范围为 64 KB 即 32K 字。数据存储空间的低半地址部分（即当 EA_{15:0} = 0 时）用作实现的存储单元，而高半地址部分（EA_{15:0} = 1）则保留为程序空间可视性区域（见第 4.3.3 节“使用程序空间可视性读程序存储器中的数据”）。

PIC24FJ256GB110 系列器件共实现了 16 KB 的数据存储空间。如果 EA 指向了该区域以外的存储单元，则将返回一个全零的字或字节。

4.2.1 数据空间宽度

数据存储空间组织为可字节寻址的 16 位宽的块。在数据存储器和寄存器中的数据是以 16 位字为单位对齐的，但所有数据空间 EA 都将解析为字节。每个字的低字节（Least Significant Byte, LSB）部分具有偶地址，而高字节（Most Significant Byte, MSB）部分则具有奇地址。

图 4-3：PIC24FJ256GB110 系列器件的数据存储空间映射



4.2.2 数据存储器构成和对齐方式

为维持与 PIC® 器件的后向兼容性和提高数据存储空间的使用效率, PIC24F 指令集同时支持字和字节操作。字节访问会在内部按字对齐的存储空间的所有有效地址进行计算调整。例如, 对于执行后修改寄存器间接寻址模式 [Ws++ 的结果, 字节操作时, 内核将其识别为值 Ws + 1, 而字操作时, 内核将其识别为值 Ws + 2。

使用任何 EA 的 LSb 来确定要选取的字节, 数据字节读取将读取包含字节的整个字。选定的字节被放在数据总线的 LSB 处。这就是说, 数据存储器和寄存器被组织为两个并行的字节宽的实体, 它们共享 (字) 地址译码, 但写入线独立。数据字节写操作只写阵列或寄存器中与字节地址匹配的那一侧。

所有字访问必须按偶地址对齐。不支持不对齐的字数据取操作, 所以在混合字节和字操作时, 或者从 8 位 MCU 代码移植时, 必须要小心。如果试图进行不对齐的读或写操作, 将产生地址错误陷阱。如果在读操作时产生错误, 正在执行的指令将完成; 而如果在写操作时产生错误, 指令仍将执行, 但不会进行写入。无论是哪种情况, 都会产生陷阱, 从而系统和 / 或用户能够检查地址错误发生之前的机器状态。

所有装入 W 寄存器的字节都将装入 W 寄存器的低字节 (LSB), W 寄存器的高字节 (MSB) 不变。

提供了一条符号扩展 (SE) 指令, 允许用户把 8 位有符号数据转换为 16 位有符号值。或者, 对于 16 位无符号数据, 用户可以通过在适当地址处执行一条零扩展 (ZE) 指令清零任何 W 寄存器的 MSB。

尽管大多数指令能够对字或字节大小的数据进行操作, 但要注意的是, 部分指令只对字大小的数据进行操作。

4.2.3 NEAR 数据空间

在 0000h 和 1FFFh 之间的 8 KB 的区域被称为 near 数据空间。可以使用所有存储器直接寻址指令中的 13 位绝对地址字段直接寻址这一空间中的存储单元。可以间接寻址数据空间的其余部分。此外, 还可以使用 MOV 指令寻址整个数据空间, 支持使用 16 位地址字段的存储器直接寻址模式。

4.2.4 SFR 空间

Near 数据空间的前 2 KB 存储单元 (从 0000h 到 07FFh) 主要被特殊功能寄存器 (Special Function Register, SFR) 占用。PIC24F 内核和外设模块使用这些寄存器来控制器件的工作。

SFR 分布在受其控制的模块中, 通常一个模块会使用一组 SFR。大部分 SFR 空间包含未用的地址单元, 它们读为 0。显示实际实现的 SFR 位置的 SFR 空间图表如表 4-2 所示。每个已实现区域指示至少有一个地址被实现为 SFR 的 32 字节区域。表 4-3 到表 4-30 给出了已实现 SFR 的完整列表 (包括它们的地址)。

表 4-2: SFR 数据空间的已实现区域

SFR 空间地址								
	xx00	xx20	xx40	xx60	xx80	xxA0	xxC0	xxE0
000h	内核			ICN	中断			—
100h	定时器		捕捉		比较			
200h	I ² C TM	UART	SPI/UART	SPI/I ² C	SPI	UART	I/O	
300h	A/D	A/D/CTMU	—	—	—	—	—	—
400h	—	—	—	—	USB			—
500h	—	—	—	—	—	—	—	—
600h	PMP	RTC/ 比较	CRC	—	PPS			—
700h	—	—	系统	NVM/PMD	—	—	—	—

图注: — = 此存储块中未实现的 SFR

表 4-3: CPU 内核寄存器映射

图注: — = 未实现, 读为 0。复位值以十六进制显示。

表 4-4: ICN 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
CNPD1	0054	CN15PDE	CN14PDE	CN13PDE	CN12PDE	CN11PDE	CN10PDE	CN9PDE	CN8PDE	CN7PDE	CN6PDE	CN5PDE	CN4PDE	CN3PDE	CN2PDE	CN1PDE	CN0PDE	0000
CNPD2	0056	CN31PDE	CN30PDE	CN29PDE	CN28PDE	CN27PDE	CN26PDE	CN25PDE	CN24PDE	CN23PDE	CN22PDE	CN21PDE ⁽¹⁾	CN20PDE ⁽¹⁾	CN19PDE ⁽¹⁾	CN18PDE	CN17PDE	CN16PDE	0000
CNPD3	0058	CN47PDE ⁽¹⁾	CN46PDE ⁽²⁾	CN45PDE ⁽¹⁾	CN44PDE ⁽¹⁾	CN43PDE ⁽¹⁾	CN42PDE ⁽¹⁾	CN41PDE ⁽¹⁾	CN40PDE ⁽²⁾	CN39PDE ⁽²⁾	CN38PDE ⁽²⁾	CN37PDE ⁽²⁾	CN36PDE ⁽²⁾	CN35PDE ⁽²⁾	CN34PDE ⁽²⁾	CN33PDE ⁽²⁾	CN32PDE	0000
CNPD4	005A	CN63PDE	CN62PDE	CN61PDE	CN60PDE	CN59PDE	CN58PDE	CN57PDE ⁽¹⁾	CN56PDE	CN55PDE	CN54PDE	CN53PDE	CN52PDE	CN51PDE	CN50PDE	CN49PDE	CN48PDE ⁽²⁾	0000
CNPD5	005C	CN79PDE ⁽²⁾	CN78PDE ⁽¹⁾	CN77PDE ⁽¹⁾	CN76PDE ⁽²⁾	CN75PDE ⁽²⁾	CN74PDE ⁽¹⁾	—	—	CN71PDE	CN70PDE ⁽¹⁾	CN69PDE	CN68PDE	CN67PDE ⁽¹⁾	CN66PDE ⁽¹⁾	CN65PDE	CN64PDE	0000
CNPD6 ⁽²⁾	005E	—	—	—	—	—	—	—	—	—	—	—	—	—	CN82PDE ⁽²⁾	CN81PDE ⁽²⁾	CN80PDE ⁽²⁾	0000
CNEN1	0060	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0062	CN31IE	CN30IE	CN29IE	CN28IE	CN27IE	CN26IE	CN25IE	CN24IE	CN23IE	CN22IE	CN21IE ⁽¹⁾	CN20IE ⁽¹⁾	CN19IE ⁽¹⁾	CN18IE	CN17IE	CN16IE	0000
CNEN3	0064	CN47IE ⁽¹⁾	CN46IE ⁽²⁾	CN45IE ⁽¹⁾	CN44IE ⁽¹⁾	CN43IE ⁽¹⁾	CN42IE ⁽¹⁾	CN41IE ⁽¹⁾	CN40IE ⁽²⁾	CN39IE ⁽²⁾	CN38IE ⁽²⁾	CN37IE ⁽²⁾	CN36IE ⁽²⁾	CN35IE ⁽²⁾	CN34IE ⁽²⁾	CN33IE ⁽²⁾	CN32IE	0000
CNEN4	0066	CN63IE	CN62IE	CN61IE	CN60IE	CN59IE	CN58IE	CN57IE ⁽¹⁾	CN56IE	CN55IE	CN54IE	CN53IE	CN52IE	CN51IE	CN50IE	CN49IE	CN48IE ⁽²⁾	0000
CNEN5	0068	CN79IE ⁽²⁾	CN78IE ⁽¹⁾	CN77IE ⁽¹⁾	CN76IE ⁽²⁾	CN75IE ⁽²⁾	CN74IE ⁽¹⁾	—	—	CN71IE	CN70IE ⁽¹⁾	CN69IE	CN68IE	CN67IE ⁽¹⁾	CN66IE ⁽¹⁾	CN65IE	CN64IE	0000
CNEN6 ⁽²⁾	006A	—	—	—	—	—	—	—	—	—	—	—	—	—	CN82IE ⁽²⁾	CN81IE ⁽²⁾	CN80IE ⁽²⁾	0000
CNPU1	006C	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	006E	CN31PUE	CN30PUE	CN29PUE	CN28PUE	CN27PUE	CN26PUE	CN25PUE	CN24PUE	CN23PUE	CN22PUE	CN21PUE ⁽¹⁾	CN20PUE ⁽¹⁾	CN19PUE ⁽¹⁾	CN18PUE	CN17PUE	CN16PUE	0000
CNPU3	0070	CN47PUE ⁽¹⁾	CN46PUE ⁽²⁾	CN45PUE ⁽¹⁾	CN44PUE ⁽¹⁾	CN43PUE ⁽¹⁾	CN42PUE ⁽¹⁾	CN41PUE ⁽¹⁾	CN40PUE ⁽²⁾	CN39PUE ⁽²⁾	CN38PUE ⁽²⁾	CN37PUE ⁽²⁾	CN36PUE ⁽²⁾	CN35PUE ⁽²⁾	CN34PUE ⁽²⁾	CN33PUE ⁽²⁾	CN32PUE	0000
CNPU4	0072	CN63PUE	CN62PUE	CN61PUE	CN60PUE	CN59PUE	CN58PUE	CN57PUE ⁽¹⁾	CN56PUE	CN55PUE	CN54PUE	CN53PUE	CN52PUE	CN51PUE	CN50PUE	CN49PUE	CN48PUE ⁽²⁾	0000
CNPU5	0074	CN79PUE ⁽²⁾	CN78PUE ⁽¹⁾	CN77PUE ⁽¹⁾	CN76PUE ⁽²⁾	CN75PUE ⁽²⁾	CN74PUE ⁽¹⁾	—	—	CN71PUE	CN70PUE ⁽¹⁾	CN69PUE	CN68PUE	CN67PUE ⁽¹⁾	CN66PUE ⁽¹⁾	CN65PUE	CN64PUE	0000
CNPU6 ⁽²⁾	0076	—	—	—	—	—	—	—	—	—	—	—	—	—	CN82PUE ⁽²⁾	CN81PUE ⁽²⁾	CN80PUE ⁽²⁾	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。

注 1: 在 64 引脚器件上未实现; 读为 0。

2: 在 64 引脚和 80 引脚器件上未实现; 读为 0。

表 4-5：中断控制器寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
INTCON1	0080	NSTDIS	—	—	—	—	—	—	—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000	
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP	0000	
IFS0	0084	—	—	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF	T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	—	IC8IF	IC7IF	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF	0000
IFS2	0088	—	—	PMPIF	OC8IF	OC7IF	OC6IF	OC5IF	IC6IF	IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF	0000
IFS3	008A	—	—	RTCIF	—	—	—	—	—	—	INT4IF	INT3IF	—	—	MI2C2IF	SI2C2IF	—	0000
IFS4	008C	—	—	CTMUIF	—	—	—	—	LVDIF	—	—	—	—	CRCIF	U2ERIF	U1ERIF	—	0000
IFS5	008E	—	—	IC9IF	OC9IF	SPI3IF	SPF3IF	U4TXIF	U4RXIF	U4ERIF	USB1IF	MI2C3IF	SI2C3IF	U3TXIF	U3RXIF	U3ERIF	—	0000
IEC0	0094	—	—	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE	T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	—	IC8IE	IC7IE	—	INT1IE	CNIE	CMIE	MI2C1IE	SI2C1IE	0000
IEC2	0098	—	—	PMPIE	OC8IE	OC7IE	OC6IE	OC5IE	IC6IE	IC5IE	IC4IE	IC3IE	—	—	—	SPI2IE	SPF2IE	0000
IEC3	009A	—	—	RTCIE	—	—	—	—	—	—	INT4IE	INT3IE	—	—	MI2C2IE	SI2C2IE	—	0000
IEC4	009C	—	—	CTMUIE	—	—	—	—	LVDIE	—	—	—	—	CRCIE	U2ERIE	U1ERIE	—	0000
IEC5	009E	—	—	IC9IE	OC9IE	SPI3IE	SPF3IE	U4TXIE	U4RXIE	U4ERIE	USB1IE	MI2C3IE	SI2C3IE	U3TXIE	U3RXIE	U3ERIE	—	0000
IPC0	00A4	—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0	—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0	4444
IPC1	00A6	—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0	—	IC2IP2	IC2IP1	IC2IP0	—	—	—	—	4440
IPC2	00A8	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0	—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0	4444
IPC3	00AA	—	—	—	—	—	—	—	—	—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0	0044
IPC4	00AC	—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0	—	MI2C1P2	MI2C1P1	MI2C1P0	—	SI2C1P2	SI2C1P1	SI2C1P0	4444
IPC5	00AE	—	IC8IP2	IC8IP1	IC8IP0	—	IC7IP2	IC7IP1	IC7IP0	—	—	—	—	INT1IP2	INT1IP1	INT1IP0	4404	
IPC6	00B0	—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0	—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—	4440
IPC7	00B2	—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0	4444
IPC8	00B4	—	—	—	—	—	—	—	—	—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0	0044
IPC9	00B6	—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0	—	IC3IP2	IC3IP1	IC3IP0	—	—	—	—	4440
IPC10	00B8	—	OC7IP2	OC7IP1	OC7IP0	—	OC6IP2	OC6IP1	OC6IP0	—	OC5IP2	OC5IP1	OC5IP0	—	IC6IP2	IC6IP1	IC6IP0	4444
IPC11	00BA	—	—	—	—	—	—	—	—	—	PMPIP2	PMPIP1	PMPIP0	—	OC8IP2	OC8IP1	OC8IP0	0044
IPC12	00BC	—	—	—	—	—	MI2C2P2	MI2C2P1	MI2C2P0	—	SI2C2P2	SI2C2P1	SI2C2P0	—	—	—	—	0440
IPC13	00BE	—	—	—	—	—	INT4IP2	INT4IP1	INT4IP0	—	INT3IP2	INT3IP1	INT3IP0	—	—	—	—	0440
IPC15	00C2	—	—	—	—	—	RTCP2	RTCP1	RTCP0	—	—	—	—	—	—	—	0400	
IPC16	00C4	—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—	4440
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0	0004
IPC19	00CA	—	—	—	—	—	—	—	—	—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—	0040
IPC20	00CC	—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0	—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—	4440
IPC21	00CE	—	U4ERIP2	U4ERIP1	U4ERIP0	—	USB1IP2	USB1IP1	USB1IP0	—	MI2C3P2	MI2C3P1	MI2C3P0	—	SI2C3P2	SI2C3P1	SI2C3P0	4444
IPC22	00D0	—	SPI3IP2	SPI3IP1	SPI3IP0	—	SPF3IP2	SPF3IP1	SPF3IP0	—	U4TXIP2	U4TXIP1	U4TXIP0	—	U4RXIP2	U4RXIP1	U4RXIP0	4444
IPC23	00D2	—	—	—	—	—	—	—	—	—	IC9IP2	IC9IP1	IC9IP0	—	OC9IP2	OC9IP1	OC9IP0	0044
INTTREG	00E0	CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0	—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0	0000

图注：— = 未实现，读为 0。复位值以十六进制显示。

表 4-6: 定时器寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
TMR1	0100																0000	
PR1	0102																FFFF	
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000	
TMR2	0106																0000	
TMR3HLD	0108																0000	
TMR3	010A																0000	
PR2	010C																FFFF	
PR3	010E																FFFF	
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000	
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000	
TMR4	0114																0000	
TMR5HLD	0116																0000	
TMR5	0118																0000	
PR4	011A																FFFF	
PR5	011C																FFFF	
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000	
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000	

图注: — = 未实现, 读为 0。复位值以十六进制显示。

表 4-7: 输入捕捉寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
IC1CON1	0140	—	—	ICSDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC1CON2	0142	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC1BUF	0144	输入捕捉 1 缓冲寄存器															0000	
IC1TMR	0146	定时器值 1 寄存器															xxxx	
IC2CON1	0148	—	—	ICSDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC2CON2	014A	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC2BUF	014C	输入捕捉 2 缓冲寄存器															0000	
IC2TMR	014E	定时器值 2 寄存器															xxxx	
IC3CON1	0150	—	—	ICSDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC3CON2	0152	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC3BUF	0154	输入捕捉 3 缓冲寄存器															0000	
IC3TMR	0156	定时器值 3 寄存器															xxxx	
IC4CON1	0158	—	—	ICSDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC4CON2	015A	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC4BUF	015C	输入捕捉 4 缓冲寄存器															0000	
IC4TMR	015E	定时器值 4 寄存器															xxxx	
IC5CON1	0160	—	—	ICSDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC5CON2	0162	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC5BUF	0164	输入捕捉 5 缓冲寄存器															0000	
IC5TMR	0166	定时器值 5 寄存器															xxxx	
IC6CON1	0168	—	—	ICSDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC6CON2	016A	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC6BUF	016C	输入捕捉 6 缓冲寄存器															0000	
IC6TMR	016E	定时器值 6 寄存器															xxxx	
IC7CON1	0170	—	—	ICSDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC7CON2	0172	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC7BUF	0174	输入捕捉 7 缓冲寄存器															0000	
IC7TMR	0176	定时器值 7 寄存器															xxxx	
IC8CON1	0178	—	—	ICSDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC8CON2	017A	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC8BUF	017C	输入捕捉 8 缓冲寄存器															0000	
IC8TMR	017E	定时器值 8 寄存器															xxxx	
IC9CON1	0180	—	—	ICSDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC9CON2	0182	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC9BUF	0184	输入捕捉 9 缓冲寄存器															0000	
IC9TMR	0186	定时器值 9 寄存器															xxxx	

图注: — = 未实现, 读为 0。复位值以十六进制显示。

表 4-8: 输出比较寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态	
OC1CON1	0190	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLTO	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000	
OC1CON2	0192	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC1RS	0194	输出比较 1 辅助寄存器																0000	
OC1R	0196	输出比较 1 寄存器																0000	
OC1TMR	0198	定时器值 1 寄存器																xxxx	
OC2CON1	019A	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLTO	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000	
OC2CON2	019C	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC2RS	019E	输出比较 2 辅助寄存器																0000	
OC2R	01A0	输出比较 2 寄存器																0000	
OC2TMR	01A2	定时器值 2 寄存器																xxxx	
OC3CON1	01A4	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLTO	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000	
OC3CON2	01A6	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC3RS	01A8	输出比较 3 辅助寄存器																0000	
OC3R	01AA	输出比较 3 寄存器																0000	
OC3TMR	01AC	定时器值 3 寄存器																xxxx	
OC4CON1	01AE	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLTO	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000	
OC4CON2	01B0	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC4RS	01B2	输出比较 4 辅助寄存器																0000	
OC4R	01B4	输出比较 4 寄存器																0000	
OC4TMR	01B6	定时器值 4 寄存器																xxxx	
OC5CON1	01B8	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLTO	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000	
OC5CON2	01BA	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC5RS	01BC	输出比较 5 辅助寄存器																0000	
OC5R	01BE	输出比较 5 寄存器																0000	
OC5TMR	01C0	定时器值 5 寄存器																xxxx	
OC6CON1	01C2	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLTO	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000	
OC6CON2	01C4	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC6RS	01C6	输出比较 6 辅助寄存器																0000	
OC6R	01C8	输出比较 6 寄存器																0000	
OC6TMR	01CA	定时器值 6 寄存器																xxxx	
OC7CON1	01CC	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLTO	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000	
OC7CON2	01CE	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC7RS	01D0	输出比较 7 辅助寄存器																0000	
OC7R	01D2	输出比较 7 寄存器																0000	
OC7TMR	01D4	定时器值 7 寄存器																xxxx	

图注: — = 未实现, 读为 0。复位值以十六进制显示。

表 4-8：输出比较寄存器映射（续）

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
OC8CON1	01D6	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCMO	0000
OC8CON2	01D8	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC8RS	01DA	输出比较 8 辅助寄存器															0000	
OC8R	01DC	输出比较 8 寄存器															0000	
OC8TMR	01DE	定时器值 8 寄存器															xxxx	
OC9CON1	01E0	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—	ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2	OCM1	OCMO	0000
OC9CON2	01E2	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	—	—	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC9RS	01E4	输出比较 9 辅助寄存器															0000	
OC9R	01E6	输出比较 9 寄存器															0000	
OC9TMR	01E8	定时器值 9 寄存器															xxxx	

图注：— = 未实现，读为 0。复位值以十六进制显示。

表 4-9：I²CTM 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
I2C1RCV	0200	—	—	—	—	—	—	—	—	接收寄存器								0000
I2C1TRN	0202	—	—	—	—	—	—	—	—	发送寄存器								00FF
I2C1BRG	0204	—	—	—	—	—	—	—	波特率发生器寄存器								0000	
I2C1CON	0206	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C1STAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/A	P	S	R/W	RBF	TBF	0000
I2C1ADD	020A	—	—	—	—	—	—	—	地址寄存器								0000	
I2C1MSK	020C	—	—	—	—	—	—	—	地址掩码寄存器								0000	
I2C2RCV	0210	—	—	—	—	—	—	—	—	接收寄存器								0000
I2C2TRN	0212	—	—	—	—	—	—	—	—	发送寄存器								00FF
I2C2BRG	0214	—	—	—	—	—	—	—	波特率发生器寄存器								0000	
I2C2CON	0216	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C2STAT	0218	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/A	P	S	R/W	RBF	TBF	0000
I2C2ADD	021A	—	—	—	—	—	—	—	地址寄存器								0000	
I2C2MSK	021C	—	—	—	—	—	—	—	地址掩码寄存器								0000	
I2C3RCV	0270	—	—	—	—	—	—	—	—	接收寄存器								0000
I2C3TRN	0272	—	—	—	—	—	—	—	—	发送寄存器								00FF
I2C3BRG	0274	—	—	—	—	—	—	—	波特率发生器寄存器								0000	
I2C3CON	0276	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C3STAT	0278	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/A	P	S	R/W	RBF	TBF	0000
I2C3ADD	027A	—	—	—	—	—	—	—	地址寄存器								0000	
I2C3MSK	027C	—	—	—	—	—	—	—	地址掩码寄存器								0000	

图注：— = 未实现，读为 0。复位值以十六进制显示。

表 4-10: UART 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
U1MODE	0220	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UENO	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U1TXREG	0224	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	发送寄存器	xxxxx
U1RXREG	0226	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	接收寄存器	0000
U1BRG	0228	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	波特率发生器预分频器寄存器	0000
U2MODE	0230	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UENO	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U2STA	0232	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U2TXREG	0234	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	发送寄存器	xxxxx
U2RXREG	0236	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	接收寄存器	0000
U2BRG	0238	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	波特率发生器预分频器寄存器	0000
U3MODE	0250	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UENO	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U3STA	0252	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U3TXREG	0254	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	发送寄存器	xxxxx
U3RXREG	0256	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	接收寄存器	0000
U3BRG	0258	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	波特率发生器预分频器寄存器	0000
U4MODE	02B0	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UENO	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U4STA	02B2	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U4TXREG	02B4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	发送寄存器	xxxxx
U4RXREG	02B6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	接收寄存器	0000
U4BRG	02B8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	波特率发生器预分频器寄存器	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。

表 4-11: SPI 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态	
SPI1STAT	0240	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000	
SPI1CON1	0242	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000	
SPI1CON2	0244	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI1BUF	0248	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	发送和接收缓冲器	0000	
SPI2STAT	0260	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000	
SPI2CON1	0262	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000	
SPI2CON2	0264	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI2BUF	0268	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	发送和接收缓冲器	0000	
SPI3STAT	0280	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000	
SPI3CON1	0282	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000	
SPI3CON2	0284	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI3BUF	0288	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	发送和接收缓冲器	0000	

图注: — = 未实现, 读为 0。复位值以十六进制显示。

表 4-12: PORTA 寄存器映射(1)

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7 ⁽²⁾	Bit 6 ⁽²⁾	Bit 5 ⁽²⁾	Bit 4 ⁽²⁾	Bit 3 ⁽²⁾	Bit 2 ⁽²⁾	Bit 1 ⁽²⁾	Bit 0 ⁽²⁾	所有复位时的状态
TRISA	02C0	TRISA15	TRISA14	—	—	—	TRISA10	TRISA9	—	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	36FF
PORTA	02C2	RA15	RA14	—	—	—	RA10	RA9	—	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx
LATA	02C4	LATA15	LATA14	—	—	—	LATA10	LATA9	—	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx
ODCA	02C6	ODA15	ODA14	—	—	—	ODA10	ODA9	—	ODA7	ODA6	ODA5	ODA4	ODA3	ODA2	ODA1	ODA0	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。所示复位值适用于 100 引脚器件。

注 1: PORTA 及其所有相关位在 64 引脚器件上未实现, 读为 0。除非另外说明, 否则这些位仅在 80 引脚和 100 引脚器件上可用。

2: 这些位仅在 100 引脚器件上实现; 否则读为 0。

表 4-13: PORTB 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
TRISB	02C8	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	FFFF
PORTB	02CA	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx
LATB	02CC	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx
ODCB	02CE	ODB15	ODB14	ODB13	ODB12	ODB11	ODB10	ODB9	ODB8	ODB7	ODB6	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0	0000

图注: 复位值以十六进制显示。

表 4-14: PORTC 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4 ⁽¹⁾	Bit 3 ⁽²⁾	Bit 2 ⁽¹⁾	Bit 1 ⁽²⁾	Bit 0	所有复位时的状态
TRISC	02D0	TRISC15	TRISC14	TRISC13	TRISC12	—	—	—	—	—	—	—	TRISC4	TRISC3	TRISC2	TRISC1	—	F01E
PORTC	02D2	RC15 ^(3,4)	RC14	RC13	RC12 ⁽³⁾	—	—	—	—	—	—	—	RC4	RC3	RC2	RC1	—	xxxx
LATC	02D4	LATC15	LATC14	LATC13	LATC12	—	—	—	—	—	—	—	LATC4	LATC3	LATC2	LATC1	—	xxxx
ODCC	02D6	ODC15	ODC14	ODC13	ODC12	—	—	—	—	—	—	—	ODC4	ODC3	ODC2	ODC1	—	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。所示复位值适用于 100 引脚器件。

注 1: 这些位在 64 引脚和 80 引脚器件上未实现; 读为 0。

2: 这些位在 64 引脚器件上未实现; 读为 0。

3: 当禁止主振荡器或选取 EC 模式时 (POSCMD<1:0> 配置位 = 11 或 00), 才能使用 RC12 和 RC15; 否则读为 0。

4: 当 POSCMD<1:0> 配置位 = 11 或 00 且 OSCIOFN 配置位 = 1 时, 才能使用 RC15。

表 4-15: PORTD 寄存器映射

寄存器名称	地址	Bit 15 ⁽¹⁾	Bit 14 ⁽¹⁾	Bit 13 ⁽¹⁾	Bit 12 ⁽¹⁾	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
TRISD	02D8	TRISD15	TRISD14	TRISD13	TRISD12	TRISD11	TRISD10	TRISD9	TRISD8	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	FFFF
PORTD	02DA	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx
LATD	02DC	LATD15	LATD14	LATD13	LATD12	LATD11	LATD10	LATD9	LATD8	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx
ODCD	02DE	ODD15	ODD14	ODD13	ODD12	ODD11	ODD10	ODD9	ODD8	ODD7	ODD6	ODD5	ODD4	ODD3	ODD2	ODD1	ODD0	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。所示复位值适用于 100 引脚器件。

注 1: 这些位在 64 引脚器件上未实现; 读为 0。

表 4-16: PORTE 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9 ⁽¹⁾	Bit 8 ⁽¹⁾	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
TRISE	02E0	—	—	—	—	—	—	TRISE9	TRISE8	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	03FF
PORTE	02E2	—	—	—	—	—	—	RE9	RE8	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	xxxx
LATE	02E4	—	—	—	—	—	—	LATE9	LATE8	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	xxxx
ODCE	02E6	—	—	—	—	—	—	ODE9	ODE8	ODE7	ODE6	ODE5	ODE4	ODE3	ODE2	ODE1	ODE0	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。所示复位值适用于 100 引脚器件。

注 1: 这些位在 64 引脚器件上未实现; 读为 0。

表 4-17: PORTF 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13 ⁽¹⁾	Bit 12 ⁽¹⁾	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2 ⁽²⁾	Bit 1	Bit 0	所有复位时的状态
TRISF	02E8	—	—	TRISF13	TRISF12	—	—	—	—	—	—	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	31FF
PORTF	02EA	—	—	RF13	RF12	—	—	—	—	—	—	RF5	RF4	RF3	RF2	RF1	RF0	xxxx
LATF	02EC	—	—	LATF13	LATF12	—	—	—	—	—	—	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	xxxx
ODCF	02EE	—	—	ODF13	ODF12	—	—	—	—	—	—	ODF5	ODF4	ODF3	ODF2	ODF1	ODF0	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。所示复位值适用于 100 引脚器件。

注 1: 这些位在 64 引脚和 80 引脚器件上未实现; 读为 0。

2: 这些位在 64 引脚器件上未实现; 读为 0。

表 4-18: PORTG 寄存器映射

寄存器名称	地址	Bit 15 ⁽¹⁾	Bit 14 ⁽¹⁾	Bit 13 ⁽¹⁾	Bit 12 ⁽¹⁾	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1 ⁽²⁾	Bit 0 ⁽²⁾	所有复位时的状态
TRISG	02F0	TRISG15	TRISG14	TRISG13	TRISG12	—	—	TRISG9	TRISG8	TRISG7	TRISG6	—	—	TRISG3	TRISG2	TRISG1	TRISG0	F3CF
PORTG	02F2	RG15	RG14	RG13	RG12	—	—	RG9	RG8	RG7	RG6	—	—	RG3	RG2	RG1	RG0	xxxx
LATG	02F4	LATG15	LATG14	LATG13	LATG12	—	—	LATG9	LATG8	LATG7	LATG6	—	—	LATG3	LATG2	LATG1	LATG0	xxxx
ODCG	02F6	ODG15	ODG14	ODG13	ODG12	—	—	ODG9	ODG8	ODG7	ODG6	—	—	ODG3	ODG2	ODG1	ODG0	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。所示复位值适用于 100 引脚器件。

注 1: 这些位在 64 引脚和 80 引脚器件上未实现; 读为 0。

2: 这些位在 64 引脚器件上未实现; 读为 0。

表 4-19: 焊垫配置寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
PADCFG1	02FC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTSECSEL	PMPTTL	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。

表 4-20: ADC 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
ADC1BUF0	0300																xxxx	
ADC1BUF1	0302																xxxx	
ADC1BUF2	0304																xxxx	
ADC1BUF3	0306																xxxx	
ADC1BUF4	0308																xxxx	
ADC1BUF5	030A																xxxx	
ADC1BUF6	030C																xxxx	
ADC1BUF7	030E																xxxx	
ADC1BUF8	0310																xxxx	
ADC1BUF9	0312																xxxx	
ADC1BUFA	0314																xxxx	
ADC1BUFB	0316																xxxx	
ADC1BUFC	0318																xxxx	
ADC1BUFD	031A																xxxx	
ADC1BUFE	031C																xxxx	
ADC1BUFF	031E																xxxx	
AD1CON1	0320	ADON	—	ADSIDL	—	—	—	FORM1	FORM0	SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE	0000
AD1CON2	0322	VCFG2	VCFG1	VCFG0	r	—	CSCNA	—	—	BUFS	—	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000
AD1CON3	0324	ADRC	r	r	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS	0328	CH0NB	—	—	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA	—	—	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1PCFGH	032A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCFG17	PCFG16	0000
AD1PCFGL	032C	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSL	0330	CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000

图注: — = 未实现 (读为 0), r = 保留 (保持为 0)。复位值以十六进制显示。

表 4-21: CTMU 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
CTMUCON	033C	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	0000
CTMUICON	033E	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	—	—	—	—	—	—	—	—	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。

表 4-22: USB OTG 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
U1OTGIR	0480	—	—	—	—	—	—	—	IDIF	T1MSECIF	LSTATEIF	ACTVIF	SESVDF	SESENDIF	—	VBUVDIF	0000	
U1OTGIE	0482	—	—	—	—	—	—	—	IDIE	T1MSECIE	LSTATEIE	ACTVIE	SESVDE	SESENDIE	—	VBUVDIE	0000	
U1OTGSTAT	0484	—	—	—	—	—	—	—	ID	—	LSTATE	—	SESVD	SESEND	—	VBUVD	0000	
U1OTGCON	0486	—	—	—	—	—	—	—	DPPULUP	DMPULUP	DPPULDWN	DMPULDWN	VBUSON	OTGEN	VBUCHG	VBUUSDIS	0000	
U1PWRC	0488	—	—	—	—	—	—	—	UACTPND	—	—	USLPGRD	—	—	USUSPND	USBPWR	0000	
U1IR 048A ⁽¹⁾	—	—	—	—	—	—	—	—	STALLIF	—	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	URSTIF	0000	
	—	—	—	—	—	—	—	—	STALLIF	ATTACHIF ⁽¹⁾	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	DETACHIF ⁽¹⁾	0000	
U1IE 048C ⁽¹⁾	—	—	—	—	—	—	—	—	STALLIE	—	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	URSTIE	0000	
	—	—	—	—	—	—	—	—	STALLIE	ATTACHIE ⁽¹⁾	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	DETACHIE ⁽¹⁾	0000	
U1EIR 048E ⁽¹⁾	—	—	—	—	—	—	—	—	BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	0000	
	—	—	—	—	—	—	—	—	BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	EOFEF ⁽¹⁾	PIDEF	0000	
U1EIE 0490 ⁽¹⁾	—	—	—	—	—	—	—	—	BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	0000	
	—	—	—	—	—	—	—	—	BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	EOFEE ⁽¹⁾	PIDEE	0000	
U1STAT	0492	—	—	—	—	—	—	—	ENDPT3	ENDPT2	ENDPT1	ENDPT0	DIR	PPBI	—	—	0000	
U1CON 0494 ⁽¹⁾	—	—	—	—	—	—	—	—	—	SE0	PKTDIS	—	HOSTEN	RESUME	PPBRST	USBEN	0000	
	—	—	—	—	—	—	—	—	JSTATE ⁽¹⁾	SE0	TOKBUSY	USBRST	HOSTEN	RESUME	PPBRST	SOFEN ⁽¹⁾	0000	
U1ADDR	0496	—	—	—	—	—	—	—	LSPDEN ⁽¹⁾	USB 器件地址 (DEVADDR) 寄存器								0000
U1BDTP1	0498	—	—	—	—	—	—	—	缓冲器描述符表基址寄存器								—	0000
U1FRML	049A	—	—	—	—	—	—	—	帧计数寄存器低字节								—	0000
U1FRMH	049C	—	—	—	—	—	—	—	帧计数寄存器高字节								—	0000
U1TOK ⁽²⁾	049E	—	—	—	—	—	—	—	—	PID3	PID2	PID1	PIDO	EP3	EP2	EP1	EP0	0000
U1SOF ⁽²⁾	04A0	—	—	—	—	—	—	—	—	帧起始计数寄存器								0000
U1CNFG1	04A6	—	—	—	—	—	—	—	UTEYE	UOEMON	—	USBSIDL	—	—	PPB1	PPB0	0000	
U1CNFG2	04A8	—	—	—	—	—	—	—	—	—	—	PUVBU	EXTI2CEN	UVBUSDIS	UVCMPDIS	UTRDIS	0000	

图注: — = 未实现, 读为 0。复位值以十六进制显示。

注 1: 当模块在主机模式下工作时的备用寄存器或位定义。

2: 此寄存器仅在主机模式下可用。

表 4-22: USB OTG 寄存器映射 (续)

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
U1EP0	04AA	—	—	—	—	—	—	—	—	LSPD ⁽¹⁾	RETRYDIS ⁽¹⁾	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP1	04AC	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP2	04AE	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP3	04B0	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP4	04B2	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP5	04B4	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP6	04B6	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP7	04B8	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP8	04BA	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP9	04BC	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP10	04BE	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP11	04CO	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP12	04C2	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP13	04C4	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP14	04C6	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP15	04C8	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1PWMMRRS	04CC	USB 电源 PWM 占空比寄存器								USB 电源 PWM 周期寄存器								0000
U1PWMCN	04CE	PWMEN	—	—	—	—	—	PWMPOL	CNTEN	—	—	—	—	—	—	—	—	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。

注 1: 当模块在主机模式下工作时的备用寄存器或位定义。

2: 此寄存器仅在主机模式下可用。

表 4-23: 并行主 / 从端口寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
PMCON	0600	PMPPEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEEN	PTWREN	PTRDEN	CSF1	CSF0	ALP	CS2P	CS1P	BEP	WRSP	RDSP	0000
PMMODE	0602	BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0	WAITB1	WAITB0	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1	WAITE0	0000
PMADDR	0604	CS2	CS1	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	0000
并行端口数据输出寄存器 1 (缓冲区 0 和 1)																		0000
PMDOUT1	0606	并行端口数据输出寄存器 2 (缓冲区 2 和 3)																0000
PMDOUT2	0608	并行端口数据输入寄存器 1 (缓冲区 0 和 1)																0000
PMDIN1	060A	并行端口数据输入寄存器 2 (缓冲区 2 和 3)																0000
PMAEN	060C	PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8	PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0	0000
PMSTAT	060E	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。

表 4-24: 实时时钟和日历寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
ALRMLVAL	0620																	xxxx
ALCFGGRPT	0622	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
RTCVVAL	0624																	xxxx
RCFGCAL	0626	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	xxxx

图注: — = 未实现, 读为 0。复位值以十六进制显示。

表 4-25: 比较器寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
CMSTAT	0630	CMIDL	—	—	—		C3EVT	C2EVT	C1EVT	—	—	—	—	—	C3OUT	C2OUT	C1OUT	0000
CVRCON	0632	—	—	—	—	—	—	—	—	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000
CM1CON	0634	CEN	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOLO	—	CREF	—	—	CCH1	CCH0	0000
CM2CON	0636	CEN	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOLO	—	CREF	—	—	CCH1	CCH0	0000
CM3CON	0638	CEN	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOLO	—	CREF	—	—	CCH1	CCH0	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。

表 4-26: CRC 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
CRCCON	0640	—	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0	CRCFUL	CRCMPT	—	CRCGO	PLEN3	PLEN2	PLEN1	PLEN0	0040
CRCXOR	0642	X15	X14	X13	X12	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	—	0000
CRCDAT	0644																	0000
CRCWDAT	0646																	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。

表 4-27: 外设引脚选择寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
RPINR0	0680	—	—	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0	—	—	—	—	—	—	—	—	3F00
RPINR1	0682	—	—	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0	—	—	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0	3F3F
RPINR2	0684	—	—	—	—	—	—	—	—	—	—	INT4R5	INT4R4	INT4R3	INT4R2	INT4R1	INT4R0	003F
RPINR3	0686	—	—	T3CKR5	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0	—	—	T2CKR5	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0	3F3F
RPINR4	0688	—	—	T5CKR5	T5CKR4	T5CKR3	T5CKR2	T5CKR1	T5CKR0	—	—	T4CKR5	T4CKR4	T4CKR3	T4CKR2	T4CKR1	T4CKR0	3F3F
RPINR7	068E	—	—	IC2R5	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0	—	—	IC1R5	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0	3F3F
RPINR8	0690	—	—	IC4R5	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0	—	—	IC3R5	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0	3F3F
RPINR9	0692	—	—	IC6R5	IC6R4	IC6R3	IC6R2	IC6R1	IC6R0	—	—	IC5R5	IC5R4	IC5R3	IC5R2	IC5R1	IC5R0	3F3F
RPINR10	0694	—	—	IC8R5	IC8R4	IC8R3	IC8R2	IC8R1	IC8R0	—	—	IC7R5	IC7R4	IC7R3	IC7R2	IC7R1	IC7R0	3F3F
RPINR11	0696	—	—	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0	—	—	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0	3F3F
RPINR15	069E	—	—	IC9R5	IC9R4	IC9R3	IC9R2	IC9R1	IC9R0	—	—	—	—	—	—	—	—	3F00
RPINR17	06A2	—	—	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0	—	—	—	—	—	—	—	—	3F00
RPINR18	06A4	—	—	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0	—	—	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0	3F3F
RPINR19	06A6	—	—	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0	—	—	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0	3F3F
RPINR20	06A8	—	—	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0	—	—	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0	3F3F
RPINR21	06AA	—	—	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0	—	—	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0	3F3F
RPINR22	06AC	—	—	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0	—	—	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0	3F3F
RPINR23	06AE	—	—	—	—	—	—	—	—	—	—	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0	003F
RPINR27	06B6	—	—	U4CTSR5	U4CTSR4	U4CTSR3	U4CTSR2	U4CTSR1	U4CTSR0	—	—	U4RXR5	U4RXR4	U4RXR3	U4RXR2	U4RXR1	U4RXR0	3F3F
RPINR28	06B8	—	—	SCK3R5	SCK3R4	SCK3R3	SCK3R2	SCK3R1	SCK3R0	—	—	SDI3R5	SDI3R4	SDI3R3	SDI3R2	SDI3R1	SDI3R0	3F3F
RPINR29	06BA	—	—	—	—	—	—	—	—	—	—	SS3R5	SS3R4	SS3R3	SS3R2	SS3R1	SS3R0	003F
RPOR0	06C0	—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0	—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0	0000
RPOR1	06C2	—	—	RP3R5	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0	—	—	RP2R5	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0	0000
RPOR2	06C4	—	—	RP5R5 ⁽¹⁾	RP5R4 ⁽¹⁾	RP5R3 ⁽¹⁾	RP5R2 ⁽¹⁾	RP5R1 ⁽¹⁾	RP5R0 ⁽¹⁾	—	—	RP4R5	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0	0000
RPOR3	06C6	—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0	—	—	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0	0000
RPOR4	06C8	—	—	RP9R5	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0	—	—	RP8R5	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0	0000
RPOR5	06CA	—	—	RP11R5	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0	—	—	RP10R5	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0	0000
RPOR6	06CC	—	—	RP13R5	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0	—	—	RP12R5	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0	0000
RPOR7	06CE	—	—	RP15R5 ⁽¹⁾	RP15R4 ⁽¹⁾	RP15R3 ⁽¹⁾	RP15R2 ⁽¹⁾	RP15R1 ⁽¹⁾	RP15R0 ⁽¹⁾	—	—	RP14R5	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0	0000
RPOR8	06D0	—	—	RP17R5	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0	—	—	RP16R5	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0	0000
RPOR9	06D2	—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0	—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0	0000
RPOR10	06D4	—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0	—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0	0000
RPOR11	06D6	—	—	RP23R5	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0	—	—	RP22R5	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0	0000
RPOR12	06D8	—	—	RP25R5	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0	—	—	RP24R5	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0	0000
RPOR13	06DA	—	—	RP27R5	RP27R4	RP27R3	RP27R2	RP27R1	RP27R0	—	—	RP26R5	RP26R4	RP26R3	RP26R2	RP26R1	RP26R0	0000
RPOR14	06DC	—	—	RP29R5	RP29R4	RP29R3	RP29R2	RP29R1	RP29R0	—	—	RP28R5	RP28R4	RP28R3	RP28R2	RP28R1	RP28R0	0000
RPOR15	06DE	—	—	RP31R5 ⁽²⁾	RP31R4 ⁽²⁾	RP31R3 ⁽²⁾	RP31R2 ⁽²⁾	RP31R1 ⁽²⁾	RP31R0 ⁽²⁾	—	—	RP30R5	RP30R4	RP30R3	RP30R2	RP30R1	RP30R0	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。

注 1: 这些位在 64 引脚器件上未实现; 读为 0。

2: 这些位在 64 引脚和 80 引脚器件上未实现; 读为 0。

表 4-28: 系统寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
RCON	0740	TRAPR	IOPUWR	—	—	—	—	CM	PMSLP	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	注 1
OSCCON	0742	—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0	CLKLOCK	IOLOCK	LOCK	—	CF	POSCEN	SOSCEN	OSWEN	注 2
CLKDIV	0744	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	CPDIV1	CPDIV0	—	—	—	—	—	—	0100
OSCTUN	0748	—	—	—	—	—	—	—	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000	
REFOCON	074E	ROEN	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	—	—	—	—	—	—	—	0000	

图注: — = 未实现, 读为 0。复位值以十六进制显示。

注 1: RCON 寄存器的复位值取决于复位事件的类型。更多信息, 请参见第 6.0 节 “复位”。

2: OSCCON 寄存器的复位值取决于复位事件的类型和器件配置。更多信息, 请参见第 8.0 节 “振荡器配置”。

表 4-29: NVM 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
NVMCON	0760	WR	WREN	WRERR	—	—	—	—	—	—	ERASE	—	—	NVMOP3	NVMOP2	NVMOP1	NVMOP0	0000 ⁽¹⁾
NVMKEY	0766	—	—	—	—	—	—	—	—	—	—	—	—	NVMKEY 寄存器 <7:0>				0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。

注 1: 所示复位值仅适用于 POR。其他复位状态下的值取决于复位时存储器写操作或擦除操作的状态。

表 4-30: PMD 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
PMD1	0770	T5MD	T4MD	T3MD	T2MD	T1MD	—	—	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADC1MD	0000
PMD2	0772	IC8MD	IC7MD	IC6MD	IC5MD	IC4MD	IC3MD	IC2MD	IC1MD	OC8MD	OC7MD	OC6MD	OC5MD	OC4MD	OC3MD	OC2MD	OC1MD	0000
PMD3	0774	—	—	—	—	—	CMPMD	RTCCMD	PMPMD	CRCMD	—	—	—	U3MD	I2C3MD	I2C2MD	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	—	UPWMMD	U4MD	—	REFOMD	CTMUMD	LVDMD	USB1MD	0000
PMD5	0778	—	—	—	—	—	—	—	IC9MD	—	—	—	—	—	—	—	OC9MD	0000
PMD6	077A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPI3MD	0000

图注: — = 未实现, 读为 0。复位值以十六进制显示。

4.2.5 软件堆栈

除了用作工作寄存器外，PIC24F 器件中的 W15 寄存器也可用作软件堆栈指针。指针总是指向第一个可用的空字，并且从低地址向高地址方向增长。它在弹出堆栈之前递减，而在压入堆栈之后递增，如图 4-4 所示。注意，对于任何 CALL 指令时的 PC 压栈，在压入堆栈之前，PC 的 MSB 要进行零扩展，从而确保 MSB 始终是清零的。

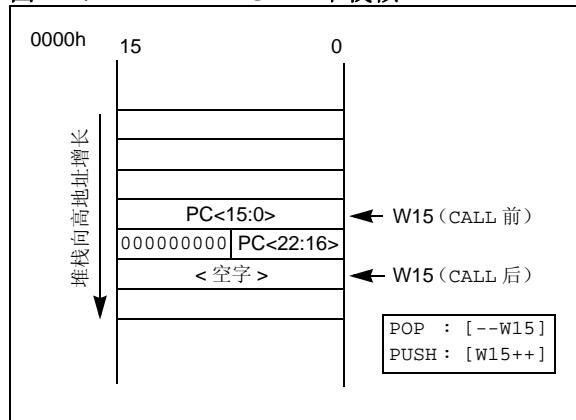
注： 在异常处理期间，在将 PC 压入堆栈之前，要先将 PC 的 MSB 与 SRL 寄存器组合在一起。

堆栈指针限制寄存器（SPLIM）与堆栈指针相关联，它设置堆栈上边界的值。复位时 SPLIM 不被初始化。与堆栈指针的情况一样，SPLIM<0> 被强制为 0，因为所有的堆栈操作必须是字对齐的。每当使用 W15 作为源指针或目标指针产生 EA 时，有效地址会与 SPLIM 中的值进行比较。如果堆栈指针（W15）的内容与 SPLIM 寄存器的内容相等，则会执行压栈操作而不产生堆栈错误陷阱，但在随后的压栈操作时将会产生堆栈错误陷阱。这样的话，当堆栈增长超过 RAM 中地址 2000h 时，如果要想产生堆栈错误陷阱，用值 1FFEh 来初始化 SPLIM 即可。

类似地，当堆栈指针地址小于 0800h 时，就会产生堆栈指针下溢（堆栈错误）陷阱。这可防止堆栈进入特殊功能寄存器（SFR）空间。

在对 SPLIM 寄存器进行写操作之后，不应紧跟着使用 W15 进行间接读操作的指令。

图 4-4：CALL 堆栈帧



4.3 程序存储空间与数据存储空间的接口

PIC24F 架构采用 24 位宽的程序空间和 16 位宽的数据空间。该架构也是一种改进型哈佛结构，这意味着数据也能存放在程序空间内。要成功使用该数据，在访问数据时必须确保这两种存储空间中的信息是对齐的。

除了正常执行外，PIC24F 架构还提供了两种可在操作过程中访问程序空间的方法：

- 使用表指令访问程序空间中任意位置的各个字节或字
- 将程序空间的一部分重新映射到数据空间（程序空间可视性）

表指令允许应用程序读写程序存储器的一小块区域。这对于访问需要随时更新的数据表来说非常理想。也可通过表操作访问一个程序字的所有字节。重映射方法允许应用程序访问一大块数据，但只限于读操作，它非常适合于在一个大的静态数据表中进行查找。这一方法只能访问程序字的低位字。

4.3.1 对程序空间进行寻址

由于数据和程序空间的地址范围分别为 16 位和 24 位，因此需要一个从 16 位数据寄存器创建一个 23 位或 24 位程序地址的方法。方法取决于所采用的接口方式。

对于表操作，使用 8 位的表存储器页地址寄存器（TBLPAG）定义程序空间内一个 32K 字的区域。这与 16 位 EA 组合形成了一个完整的 24 位程序空间地址。在这种地址形式下，TBLPAG 的最高位用来决定操作是发生在用户存储区中（TBLPAG<7> = 0）还是配置存储区中（TBLPAG<7> = 1）。

对于重映射操作，使用 8 位的程序空间可视性页地址寄存器（PSVPAG）定义程序空间中的 16K 字页。当 EA 的最高位为 1 时，PSVPAG 与 EA 的低 15 位组合形成一个 23 位的程序空间地址。与表操作不同，重映射操作被严格限制在用户存储区中。

表 4-31 和图 4-5 显示了如何从数据 EA 创建程序 EA 以进行表操作和重映射访问。这里，P<23:0> 指的是程序空间字；而 D<15:0> 指的是数据空间字。

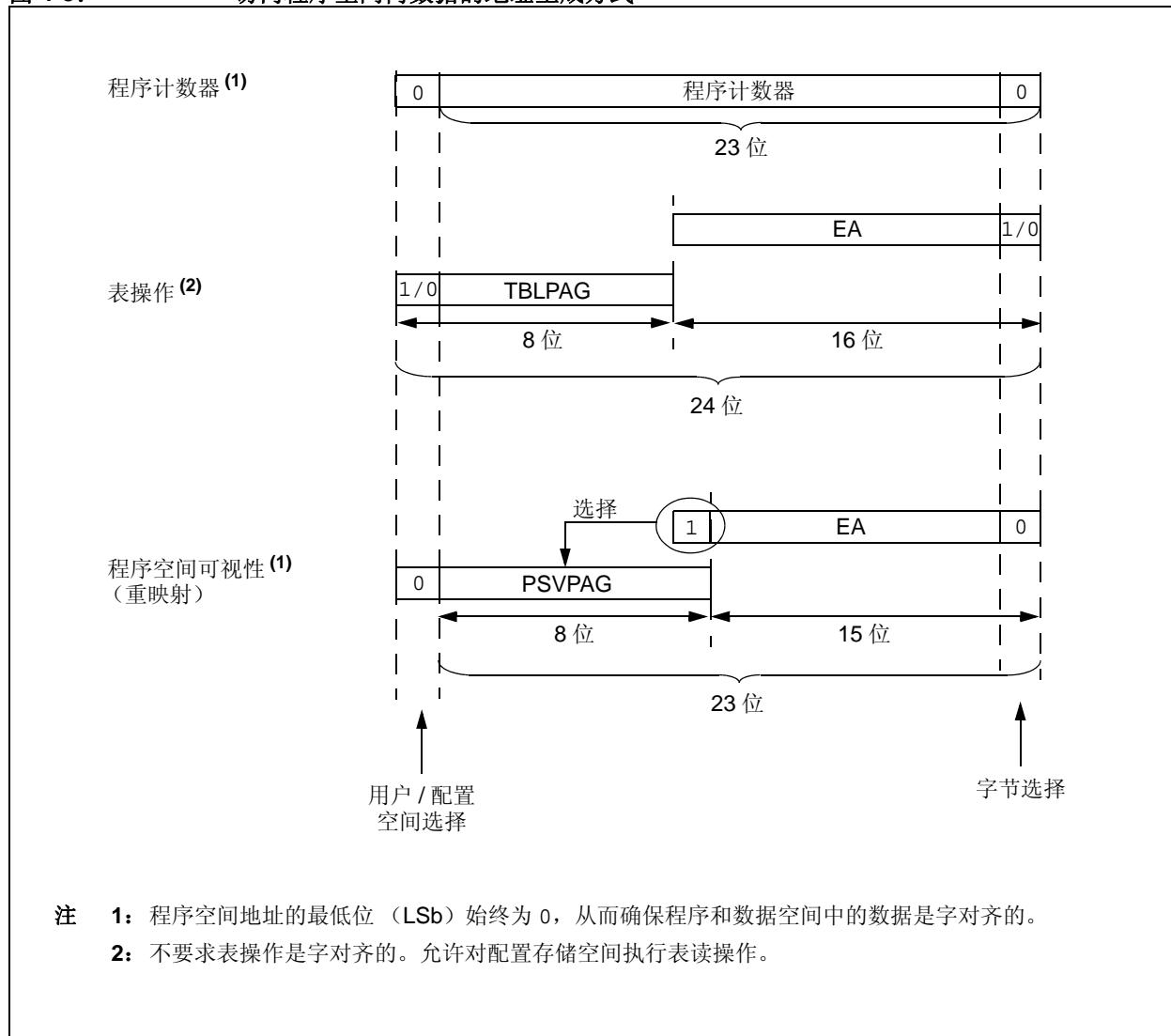
PIC24FJ256GB110 系列

表 4-31: 程序空间地址构成

访问类型	访问空间	程序空间地址				
		<23>	<22:16>	<15>	<14:1>	<0>
指令访问 (代码执行)	用户	0	PC<22:1>		0	
		0xx xxxx xxxx xxxx xxxx xxxx xxx0				
TBLRD/TBLWT (读 / 写字节或字)	用户	TBLPAG<7:0>		数据 EA<15:0>		
		0xxx xxxx	xxxx xxxx xxxx xxxx			
程序空间可视性 (块重映射 / 读)	配置	TBLPAG<7:0>		数据 EA<15:0>		
		1xxx xxxx	xxxx xxxx xxxx xxxx			
程序空间可视性 (块重映射 / 读)	用户	0	PSVPAG<7:0>		数据 EA<14:0> ⁽¹⁾	
		0	xxxx xxxx		xxx xxxx xxxx xxxx	

注 1: 在这种情况下, 数据 EA<15> 始终为 1, 但并不用它来计算程序空间地址。地址的 bit 15 为 PSVPAG<0>。

图 4-5: 访问程序空间内数据的地址生成方式



注 1: 程序空间地址的最低位 (LSb) 始终为 0, 从而确保程序和数据空间中的数据是字对齐的。

2: 不要求表操作是字对齐的。允许对配置存储空间执行表读操作。

4.3.2 使用表指令访问程序存储器中的数据

TBLRDL 和 TBLWTL 指令提供了读或写程序空间内任何地址的低位字的直接方法，无需通过数据空间。TBLRDH 和 TBLWTH 指令是可以把一个程序空间字的高 8 位作为数据读写的唯一方法。

对于每个连续的 24 位程序字，PC 的递增量为 2。这使得程序存储器地址能够直接映射到数据空间地址。于是，程序存储器可以看作是两个 16 位字宽的地址空间，它们并排放置，具有相同的地址范围。TBLRDL 和 TBLWTL 访问存有数据低位字的空间，而 TBLRDH 和 TBLWTH 则访问存有最高数据字节的空间。

提供了两条表指令来对程序空间执行字节或字（16 位）大小的数据读写。读和写都可以采用字节或字操作的形式。

1. TBLRDL（表读低位字）：在字模式下，该指令将程序空间地址的低位字（P<15:0>）映射到数据地址（D<15:0>）中。

在字节模式下，低位程序字的高字节或低字节被映射到数据地址的低字节中。当字节选择位为 1 时映射高字节；当字节选择位为 0 时映射低字节。

2. TBLRDH（表读高位字）：在字模式下，该指令将程序地址的整个高位字（P<23:16>）映射到数据地址中。注意，D<15:8> 为“虚拟”字节，它始终为 0。

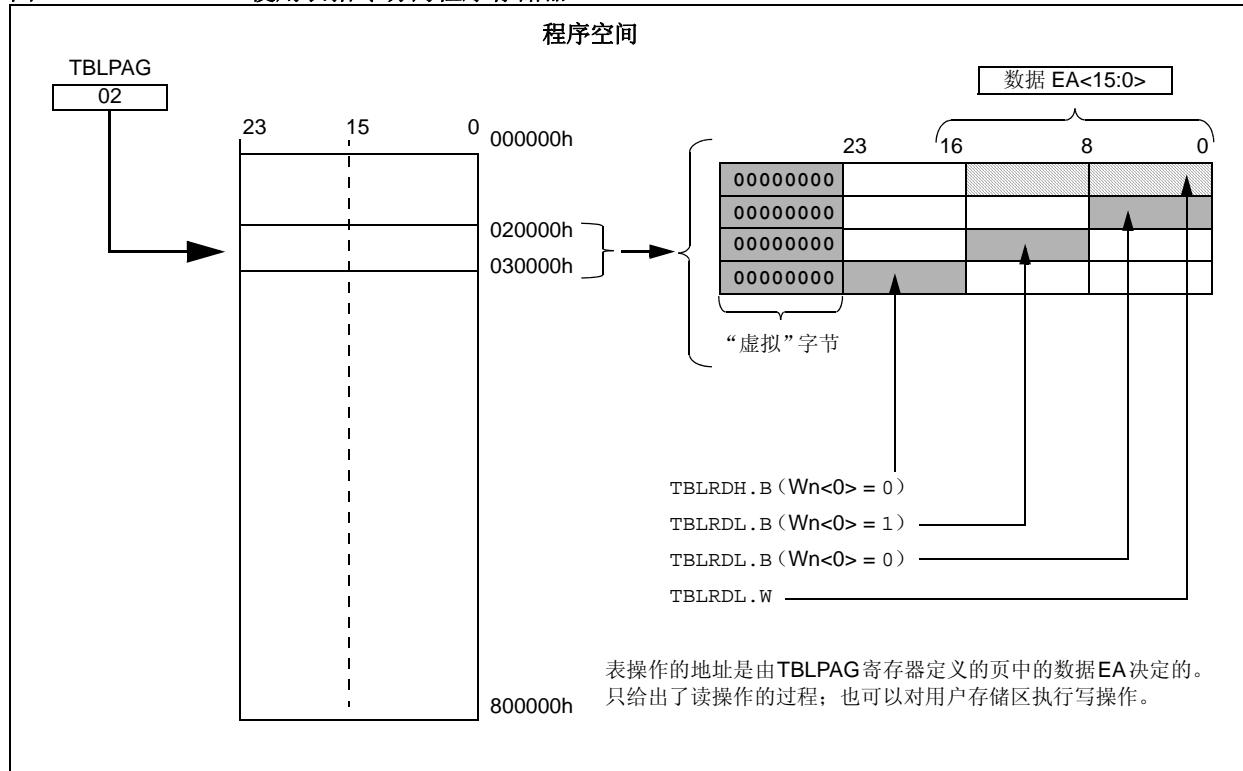
在字节模式下，该指令将程序字的高字节或低字节映射到数据地址的 D<7:0> 中，就如同 TBLRDL 指令。注意，当选择高位“虚拟”字节（字节选择位 = 1）时，数据将始终为 0。

表指令 TBLWTH 和 TBLWTL 以类似的方式向程序空间地址写入各字节或字。第 5.0 节“闪存程序存储器”对这两条指令的详细操作给出了说明。

对于所有的表操作，要访问程序存储空间的哪个区域是由表存储器页地址寄存器（TBLPAG）决定的。TBLPAG 可寻址器件的整个程序存储空间，包括用户和配置空间。当 TBLPAG<7> = 0 时，表页位于用户存储空间中。当 TBLPAG<7> = 1 时，表页位于配置存储空间中。

注：只有表读操作将在配置存储空间中执行，且只能在已实现区域中执行，例如器件 ID。不允许表写操作。

图 4-6：使用表指令访问程序存储器



4.3.3 使用程序空间可视性读程序存储器中的数据

可选择将数据空间的高 32 KB 映射到程序空间中的任何 16K 字页中。这提供了通过数据空间对存储的常量数据的透明访问，而无需使用特殊指令（即 TBLRDL/H）。

如果数据空间 EA 的最高有效位为 1，并且程序空间可视性使能（方法是将 CPU 控制寄存器中的 PSV 位（CORCON<2>）置 1）时，就能通过数据空间访问程序空间。由程序空间可视性页地址寄存器（PSVPAG）决定要被映射到数据空间中的程序存储空间的位置。这一 8 位的寄存器定义程序空间中 256 个可能的 16K 字页中的一个。事实上，PSVPAG 作为程序存储地址的高 8 位，而 EA 的 15 位则作为地址的低位。请注意，对于每个程序存储字，PC 都将递增 2，数据空间地址的低 15 位将直接映射到相应程序空间地址的低 15 位。

正在执行的指令，如果要对这个区域进行数据读取的话，就需要增加一个额外的指令周期，因为需要进行两次程序存储空间读取操作。

尽管大于或等于 8000h 的每个数据空间地址直接映射到对应的程序存储器地址（见图 4-7），但只使用 24 位程序字的低 16 位来存放数据。所有用来存放数据的程序

存储单元的高 8 位都应被设置为 1111 1111 或 0000 0000，强制为一条 NOP 指令，从而避免了可能出现意外执行这一区域内代码的情况。

注： 在表读 / 写期间，暂时禁止 PSV 访问。

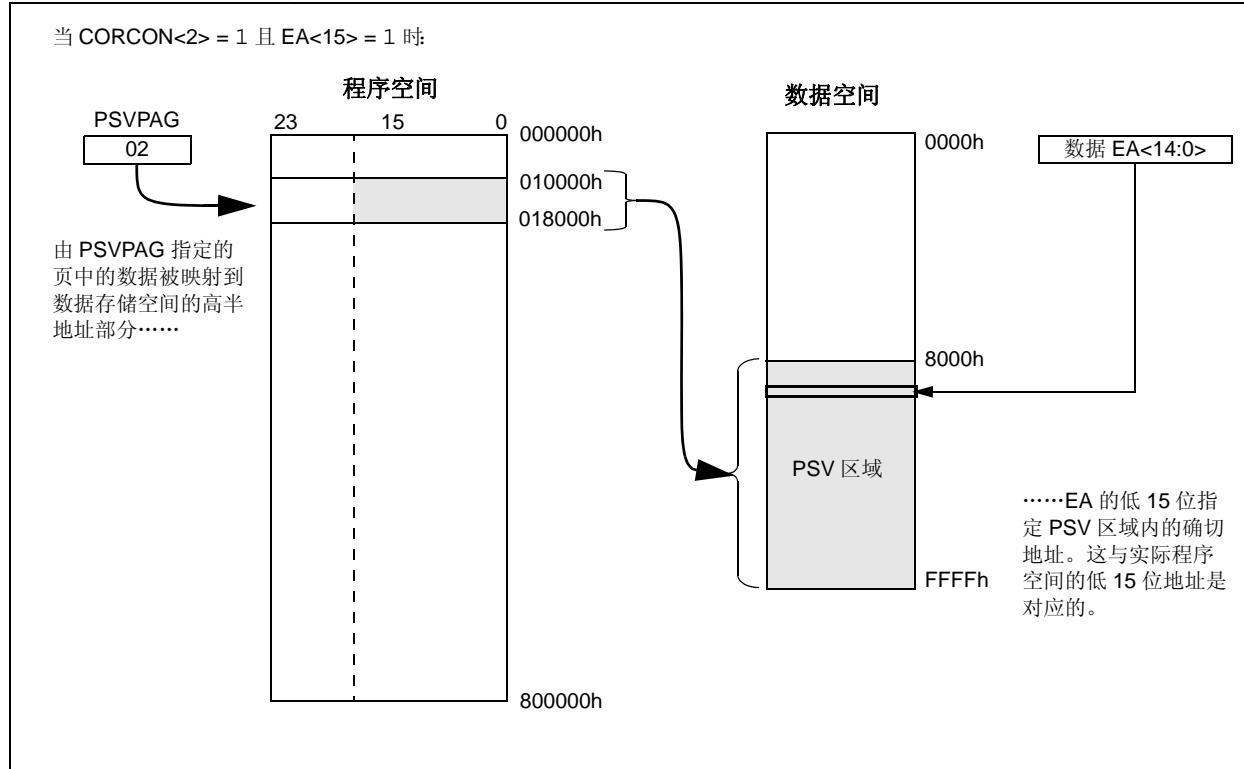
对于使用 PSV 而又在 REPEAT 循环外执行的操作，MOV 和 MOV.D 指令除了规定的执行时间之外，还需要一个额外的指令周期。其他所有指令都需要在规定的执行时间之外额外增加两个指令周期。

对于使用 PSV 而又在 REPEAT 循环内执行的操作，下列情况，除了规定的指令执行时间之外，还需要两个额外的指令周期：

- 在第一次迭代中执行的指令
- 在最后一次迭代中执行的指令
- 由于中断而退出循环之前执行的指令
- 中断得到处理后再次进入循环时执行的指令

REPEAT 循环的所有其他各次迭代，都允许使用 PSV 访问数据的指令在一个周期内执行。

图 4-7：程序空间可视性操作



5.0 闪存程序存储器

注: 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见《PIC24F 系列参考手册》的第 4 章“程序存储器”(DS39715A_CN)。

PIC24FJ256GB110 系列器件包含用于存储和执行应用代码的内部闪存程序存储器。可采用 4 种方式对它进行编程:

- 在线串行编程 (ICSP™)
- 运行时自编程 (Run-Time Self-Programming, RTSP)
- JTAG
- 增强型在线串行编程 (增强型 ICSP)

ICSP 允许在最终的应用电路中对 PIC24FJ256GB110 系列器件进行串行编程。只需要 5 根线即可完成这一操作, 其中编程时钟线和编程数据线 (名称分别为 PGECx 和 PGEDx) 各一根, 其余 3 根分别是电源线 (VDD)、接地线 (VSS) 和主复位线 (MCLR)。这允许用户在生产电路板时使用未编程器件, 仅在产品交付之前才对单片机进行编程, 从而可以使用最新版本的固件或者定制固件进行编程。

使用 TBLRD (表读) 和 TBLWT (表写) 指令来实现 RTSP。使用 RTSP, 用户可以一次写入 64 条指令 (192 字节) 块的程序存储数据, 一次擦除 512 条指令 (1536 字节) 块的程序存储单元。

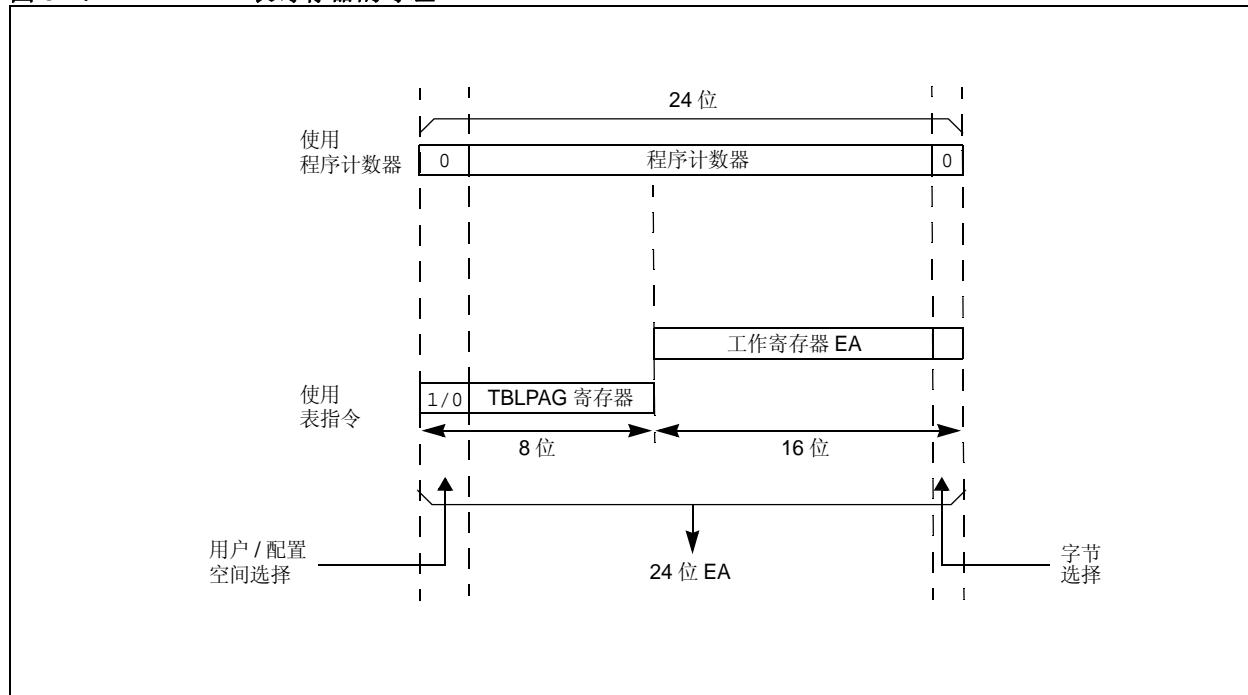
5.1 表指令和闪存编程

闪存存储器的编程都是用表读和表写指令实现的, 与使用的编程方法无关。这些指令允许器件在正常工作模式下通过数据存储器直接读写程序存储空间。程序存储器中 24 位目标地址由 TBLPAG<7:0> 位和表指令中指定 W 寄存器中的有效地址 (EA) 组成, 如图 5-1 所示。

TBLRDL 和 TBLWTL 指令用来读或写程序存储器的 bit<15:0>。TBLRDL 和 TBLWTL 能以字或字节模式访问程序存储器。

TBLRDH 和 TBLWTH 指令用来读或写程序存储器的 bit<23:16>。TBLRDH 和 TBLWTH 同样能以字或字节模式访问程序存储器。

图 5-1: 表寄存器的寻址



5.2 RTSP 工作原理

PIC24F 闪存程序存储器阵列是由 64 条指令或 192 字节的行组成的。RTSP 允许用户一次擦除 8 行（512 条指令）的块以及一次编程一行。它也可以编程单个字。

8 行擦除块和单行写块都是边沿对齐的，从程序存储器起始地址开始，分别到 1536 字节边界和 192 字节边界。

当使用 TBLWT 指令将数据写入程序存储器时，数据并不直接被写入存储器。相反，使用表写指令写入的数据被存储在保持锁存器中，直到执行编程序列。

可以执行任意数量的 TBLWT 指令，随后将成功执行一次写操作。但是，需要 64 条 TBLWT 指令写入存储器的整行。

为确保写操作期间没有数据损坏，任何未用的地址都应编程为 FFFFFFFh。这是因为保持锁存器复位为未知状态，因此如果地址保持在复位状态，它们可能会改写未重新写入的行上的存储单元。

RTSP 编程的基本步骤是先建立一个表指针，然后执行一系列 TBLWT 指令将数据装入缓冲器。编程是通过设置 NVMCON 寄存器中的控制位执行的。

可以任何顺序装入数据，而且在执行写操作前可以多次写入保持寄存器。但是，后续的写操作将擦除任何先前的写操作。

注： 不建议对同一个存储单元进行多次写操作而不进行擦除操作。

所有的表写操作都是单字写操作（2 个指令周期），因为只写缓冲器。编程每一行需要一个编程周期。

5.3 JTAG 工作原理

PIC24F 系列器件支持 JTAG 边界扫描。边界扫描通过校验引脚到 PCB 的连接性以改进制造工艺。

5.4 增强型在线串行编程

增强型在线串行编程使用板上自举程序（称为编程执行程序）来管理编程过程。使用 SPI 数据帧格式，编程执行程序能够擦除、编程和校验程序存储器。关于增强型 ICSP 的更多信息，请参见器件编程规范。

5.5 控制寄存器

有两个 SFR 用于读写闪存程序存储器：NVMCON 和 NVMKEY。

NVMCON 寄存器（寄存器 5-1）控制要擦除的块、要编程的存储器类型，以及编程周期的开始。

NVMKEY 是一个只写寄存器，用于写保护。要启动编程或擦除序列，用户必须连续地将 55h 和 AAh 写入 NVMKEY 寄存器。更多详细信息，请参见第 5.6 节“编程操作”。

5.6 编程操作

在 RTSP 模式下，对内部闪存进行编程或擦除需要执行完整的编程过程。编程或擦除操作期间，处理器暂停（等待）直到操作完成。将 WR 位（NVMCON<15>）置 1 启动操作，当操作完成时 WR 位会自动清零。

寄存器 5-1: NVMCON: 闪存控制寄存器

R/SO-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	U-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	—	—	—	—	—
bit 15	bit 8						

U-0	R/W-0 ⁽¹⁾	U-0	U-0	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾
—	ERASE	—	—	NVMOP3 ⁽²⁾	NVMOP2 ⁽²⁾	NVMOP1 ⁽²⁾	NVMOP0 ⁽²⁾
bit 7	bit 0						

图注: SO = 只可置 1 的位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **WR:** 写控制位⁽¹⁾

1 = 启动闪存存储器编程或擦除操作。该操作是自定时的, 一旦操作完成该位即由硬件清零。

0 = 编程或擦除操作完成, 并处于停止状态

bit 14 **WREN:** 写使能位⁽¹⁾

1 = 使能闪存编程 / 擦除操作

0 = 禁止闪存编程 / 擦除操作

bit 13 **WRERR:** 写序列错误标志位⁽¹⁾

1 = 试图执行不合法的编程或擦除序列, 或者发生终止 (试图将 WR 位置 1 时自动置 1 该位)

0 = 编程或擦除操作正常完成

bit 12-7 未实现: 读为 0

bit 6 **ERASE:** 擦除 / 编程使能位⁽¹⁾

1 = 在下一条 WR 命令时执行 NVMOP<3:0> 指定的擦除操作

0 = 在下一条 WR 命令时执行 NVMOP<3:0> 指定的编程操作

bit 5-4 未实现: 读为 0

bit 3-0 **NVMOP<3:0>:** NVM 操作选择位^(1,2)

1111 = 存储器批量擦除操作 (ERASE = 1) 或无操作 (ERASE = 0)⁽³⁾

0011 = 存储器字编程操作 (ERASE = 0) 或无操作 (ERASE = 1)

0010 = 存储器页擦除操作 (ERASE = 1) 或无操作 (ERASE = 0)

0001 = 存储器行编程操作 (ERASE = 0) 或无操作 (ERASE = 1)

注 1: 这些位只能在 POR 时复位。

2: NVMOP<3:0> 的所有其他组合均未实现。

3: 仅在 ICSPTM 模式下可用。请参见器件编程规范。

PIC24FJ256GB110 系列

5.6.1 闪存程序存储器的编程算法

用户可以一次对闪存存储器的一行进行编程。要实现该操作，必须擦除包含该行在内的 8 行擦除块。一般过程如下：

1. 读取程序存储器的 8 行（512 条指令），并存储在数据 RAM 中。
2. 用所需的新数据更新 RAM 中的程序数据。
3. 擦除程序块（见例 5-1 了解采用汇编器的实现方法）：
 - a) 将 NVMOP 位（NVMCON<3:0>）设置为 0010，配置为块擦除操作。将 ERASE（NVMCON<6>）和 WREN（NVMCON<14>）位置 1。
 - b) 将要被擦除的块的起始地址写入 TBLPAG 和 W 寄存器。
 - c) 将 55h 写入 NVMKEY。
 - d) 将 AAh 写入 NVMKEY。
 - e) 将 WR 位（NVMCON<15>）置 1。擦除周期开始，在擦除周期中 CPU 会暂停。当擦除完成时，WR 位会自动清零。
4. 将数据 RAM 中的前 64 条指令写入程序存储器缓冲器（见例 5-3 了解采用汇编器的实现方法）。

5. 将程序块写入闪存存储器：

- a) 将 NVMOP 位设置为 0001，配置为行编程操作。将 ERASE 位清零，将 WREN 位置 1。
- b) 将 55h 写入 NVMKEY。
- c) 将 AAh 写入 NVMKEY。
- d) 将 WR 位置 1。编程周期开始，在写周期中 CPU 会暂停。当闪存存储器写操作完成时，WR 位会自动清零。

6. 将 TBLPAG 中的值递增 1，使用数据 RAM 中下一个 64 条指令块重复步骤 4 和 5，直到所有 512 条指令被写回闪存存储器。

为防止意外操作，必须向 NVMKEY 写入启动序列从而允许执行擦除或编程操作。在执行编程命令后，用户必须等待一段编程时间，直至编程完成。紧跟编程启动序列后面的两条指令应为 NOP，如例 5-5 所示。

注： 例 5-2、5-4 和 5-6 给出了针对这些步骤的等效 C 代码，这些代码采用内置硬件函数的特定库，并使用 Microchip MPLAB C30 C 编译器编译。

例 5-1： 擦除程序存储器块（汇编语言代码）

```
; Set up NVMCON for block erase operation
    MOV      #0x4042, W0                      ;
    MOV      W0, NVMCON                         ; Initialize NVMCON
; Init pointer to row to be ERASED
    MOV      #tblpage(PROG_ADDR), W0           ;
    MOV      W0, TBLPAG                          ; Initialize PM Page Boundary SFR
    MOV      #tbloffset(PROG_ADDR), W0          ;
    MOV      W0, [W0]                            ; Initialize in-page EA[15:0] pointer
    TBLWTL  W0, [W0]                            ; Set base address of erase block
    DISI    #5                                ; Block all interrupts with priority <7
                                                ; for next 5 instructions
    MOV      #0x55, W0                          ; Write the 55 key
    MOV      W0, NVMKEY                         ;
    MOV      #0xAA, W1                          ; Write the AA key
    MOV      W1, NVMKEY                         ; Start the erase sequence
    BSET   NVMCON, #WR                         ; Insert two NOPs after the erase
                                                ; command is asserted
    NOP
    NOP
```

例 5-2: 擦除程序存储器块 (C 语言代码)

```
// C example using MPLAB C30
unsigned long progAddr = 0xFFFFFFFF;           // Address of row to write
unsigned int offset;

//Set up pointer to the first memory location to be written
TBLPAG = progAddr>>16;                         // Initialize PM Page Boundary SFR
offset = progAddr & 0xFFFF;                        // Initialize lower word of address

__builtin_tblwtl(offset, 0x0000);                 // Set base address of erase block
                                                // with dummy latch write

NVMCON = 0x4042;                                 // Initialize NVMCON

asm("DISI #5");                                // Block all interrupts with priority <7
                                                // for next 5 instructions
__builtin_write_NVM();                           // C30 function to perform unlock
                                                // sequence and set WR
```

例 5-3: 装载写缓冲器 (汇编语言代码)

```
; Set up NVMCON for row programming operations
    MOV      #0x4001, W0          ;
    MOV      W0, NVMCON          ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
    MOV      #0x0000, W0          ;
    MOV      W0, TBLPAG          ; Initialize PM Page Boundary SFR
    MOV      #0x6000, W0          ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
    MOV      #LOW_WORD_0, W2        ;
    MOV      #HIGH_BYTE_0, W3        ;
    TBLWTL  W2, [W0]              ; Write PM low word into program latch
    TBLWTH  W3, [W0++]            ; Write PM high byte into program latch
; 1st_program_word
    MOV      #LOW_WORD_1, W2        ;
    MOV      #HIGH_BYTE_1, W3        ;
    TBLWTL  W2, [W0]              ; Write PM low word into program latch
    TBLWTH  W3, [W0++]            ; Write PM high byte into program latch
; 2nd_program_word
    MOV      #LOW_WORD_2, W2        ;
    MOV      #HIGH_BYTE_2, W3        ;
    TBLWTL  W2, [W0]              ; Write PM low word into program latch
    TBLWTH  W3, [W0++]            ; Write PM high byte into program latch
    .
    .
    .
; 63rd_program_word
    MOV      #LOW_WORD_31, W2       ;
    MOV      #HIGH_BYTE_31, W3       ;
    TBLWTL  W2, [W0]              ; Write PM low word into program latch
    TBLWTH  W3, [W0]              ; Write PM high byte into program latch
```

PIC24FJ256GB110 系列

例 5-4: 装载写缓冲器 (C 语言代码)

```
// C example using MPLAB C30

#define NUM_INSTRUCTION_PER_ROW 64
unsigned int offset;
unsigned int i;
unsigned long progAddr = 0xFFFFFFFF; // Address of row to write
unsigned int progData[2*NUM_INSTRUCTION_PER_ROW]; // Buffer of data to write

//Set up NVMCON for row programming
NVMCON = 0x4001; // Initialize NVMCON

//Set up pointer to the first memory location to be written
TBLPAG = progAddr>>16; // Initialize PM Page Boundary SFR
offset = progAddr & 0xFFFF; // Initialize lower word of address

//Perform TBLWT instructions to write necessary number of latches
for(i=0; i < 2*NUM_INSTRUCTION_PER_ROW; i++)
{
    __builtin_tblwtl(offset, progData[i++]); // Write to address low word
    __builtin_tblwth(offset, progData[i]); // Write to upper byte
    offset = offset + 2; // Increment address
}
```

例 5-5: 启动编程序列 (汇编语言代码)

```
DISI    #5          ; Block all interrupts with priority <7
           ; for next 5 instructions
MOV     #0x55, W0
MOV     W0, NVMKEY      ; Write the 55 key
MOV     #0xAA, W1
MOV     W1, NVMKEY      ; Write the AA key
BSET   NVMCON, #WR      ; Start the erase sequence
NOP
NOP
BTSC   NVMCON, #15      ; and wait for it to be
BRA    $-2          ; completed
```

例 5-6: 启动编程序列 (C 语言代码)

```
// C example using MPLAB C30

asm("DISI #5"); // Block all interrupts with priority < 7
                  // for next 5 instructions

__builtin_write_NVM(); // Perform unlock sequence and set WR
```

5.6.2 编程闪存程序存储器的单字

如果已擦除某个闪存单元，则可以使用表写指令将其编程为将一个指令字（24位）写入写锁存器。TBLPAG寄存器被装入闪存地址的8个最高有效字节。TBLWTL和TBLWTH指令将所需数据写入写锁存器，并指定要写

入的程序存储器地址的低16位。要将NVMCON寄存器配置为字写操作，请将NVMOP位（NVMCON<3:0>）设置为0011。通过执行解锁序列并将WR位置1来执行写操作，如例5-7所示。采用MPLAB C30 C编译器和内置硬件函数的等效C代码如例5-8所示。

例5-7：编程闪存程序存储器的单字（汇编语言代码）

```

; Setup a pointer to data Program Memory
MOV    #tblpage(PROG_ADDR), W0      ;
MOV    W0, TBLPAG                  ;Initialize PM Page Boundary SFR
MOV    #tbloffset(PROG_ADDR), W0    ;Initialize a register with program memory address

MOV    #LOW_WORD, W2      ;
MOV    #HIGH_BYTE, W3      ;
TBLWTL W2, [W0]           ; Write PM low word into program latch
TBLWTH W3, [W0++]         ; Write PM high byte into program latch

; Setup NVMCON for programming one word to data Program Memory
MOV    #0x4003, W0      ;
MOV    W0, NVMCON            ; Set NVMOP bits to 0011

DISI   #5                  ; Disable interrupts while the KEY sequence is written
MOV    #0x55, W0            ; Write the key sequence
MOV    W0, NVMKEY
MOV    #0xAA, W0
MOV    W0, NVMKEY
BSET   NVMCON, #WR          ; Start the write cycle

```

例5-8：编程闪存程序存储器的单字（C语言代码）

```

// C example using MPLAB C30

unsigned int offset;
unsigned long progAddr = 0xFFFFFFFF;           // Address of word to program
unsigned int progDataL = 0xFFFF;                 // Data to program lower word
unsigned char progDataH = 0xFF;                  // Data to program upper byte

//Set up NVMCON for word programming
NVMCON = 0x4003;                                // Initialize NVMCON

//Set up pointer to the first memory location to be written
TBLPAG = progAddr>>16;                          // Initialize PM Page Boundary SFR
offset = progAddr & 0xFFFF;                        // Initialize lower word of address

//Perform TBLWT instructions to write latches
__builtin_tblwtl(offset, progDataL);              // Write to address low word
__builtin_tblwth(offset, progDataH);              // Write to upper byte
asm("DISI #5");                                  // Block interrupts with priority < 7
                                                // for next 5 instructions
__builtin_write_NVM();                            // C30 function to perform unlock
                                                // sequence and set WR

```

PIC24FJ256GB110 系列

注:

6.0 复位

注: 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见《PIC24F 系列参考手册》的第 7 章“复位”(DS39712A_CN)。

复位模块结合了所有复位源并控制器件的主复位信号 **SYSRST**。下面列出了器件的复位源:

- **POR:** 上电复位
- **MCLR:** 引脚复位
- **SWR:** RESET 指令
- **WDT:** 看门狗定时器复位
- **BOR:** 欠压复位
- **CM:** 配置不匹配复位
- **TRAPR:** 陷阱冲突复位
- **IOPUWR:** 非法操作码复位
- **UWR:** 未初始化的 W 寄存器复位

图 6-1 给出了复位模块的简化框图。

任何有效的复位源都会使 **SYSRST** 信号有效。很多与 CPU 和外设相关的寄存器均会被强制为已知的复位状态。大多数寄存器都不受复位影响; 它们的状态在 POR 时未知, 而在所有其他复位时不变。

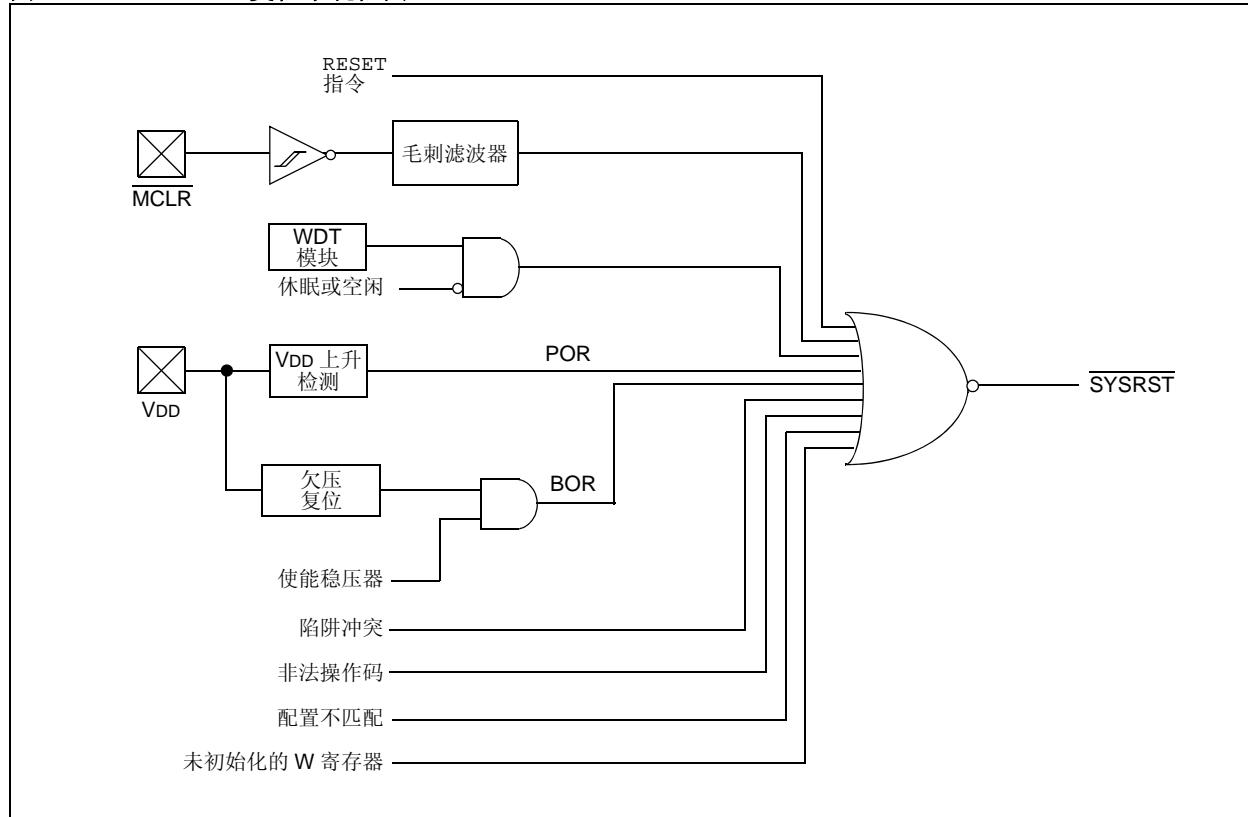
注: 如需了解寄存器复位状态的信息, 请参见本手册中特定的外设或 CPU 章节。

任何类型的器件复位都会将 RCON 寄存器中相应的位置 1, 以表明复位类型(见寄存器 6-1)。上电复位会清零除 BOR 和 POR 位(RCON<1:0>)以外的所有位, BOR 和 POR 位被置 1。用户可以在代码执行过程中的任何时间置 1 或清零任何位。RCON 寄存器中的位仅用作状态位。在软件中将特定的复位状态位置 1 不会导致器件复位发生。

RCON 寄存器还有与看门狗定时器和器件节能状态相关的其他位。在本手册的其他章节中对这些位的功能进行了讨论。

注: RCON 寄存器中的状态位应该在被读取后清零, 这样在器件复位后 RCON 寄存器的下一个值才有意义。

图 6-1: 复位系统框图



PIC24FJ256GB110 系列

寄存器 6-1: RCON: 复位控制寄存器⁽¹⁾

R/W-0, HS	R/W-0, HS	U-0	U-0	U-0	U-0	R/W-0, HS	R/W-0
TRAPR	IOPUWR	—	—	—	—	CM	PMSLP
bit 15	bit 8						

R/W-0, HS	R/W-0, HS	R/W-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-1, HS	R/W-1, HS
EXTR	SWR	SWDTEN ⁽²⁾	WDTO	SLEEP	IDLE	BOR	POR
bit 7	bit 0						

图注:	HS = 可由硬件置 1 的位
R = 可读位	W = 可写位
-n = POR 时的值	1 = 置 1 0 = 清零 x = 未知

- bit 15 **TRAPR:** 陷阱复位标志位
1 = 发生了陷阱冲突复位
0 = 未发生陷阱冲突复位
- bit 14 **IOPUWR:** 非法操作码或未初始化的 W 寄存器访问复位标志位
1 = 检测到非法操作码、非法地址模式或未初始化的 W 寄存器用作地址指针而导致复位
0 = 未发生非法操作码或未初始化的 W 寄存器复位
- bit 13-10 未实现: 读为 0
- bit 9 **CM:** 配置字不匹配复位标志位
1 = 发生了配置字不匹配复位
0 = 未发生配置字不匹配复位
- bit 8 **PMSLP:** 休眠模式下程序存储器掉电位。
1 = 在休眠模式下保持提供程序存储器偏置电压。
0 = 在休眠模式下不提供程序存储器偏置电压。稳压器进入待机模式。
- bit 7 **EXTR:** 外部复位 (MCLR) 引脚位
1 = 发生了主复位 (引脚) 复位
0 = 未发生主复位 (引脚) 复位
- bit 6 **SWR:** 软件复位 (指令) 标志位
1 = 执行了 RESET 指令
0 = 未执行 RESET 指令
- bit 5 **SWDTEN:** 软件使能 / 禁止 WDT 位⁽²⁾
1 = 使能 WDT
0 = 禁止 WDT
- bit 4 **WDTO:** 看门狗定时器超掩码标志位
1 = 发生了 WDT 超时
0 = 未发生 WDT 超时
- bit 3 **SLEEP:** 从休眠状态唤醒标志位
1 = 器件处于休眠模式
0 = 器件不处于休眠模式
- bit 2 **IDLE:** 从空闲状态唤醒标志位
1 = 器件处于空闲模式
0 = 器件不处于空闲模式
- bit 1 **BOR:** 欠压复位标志位
1 = 发生了欠压复位。注意 BOR 在上电复位后也将置 1。
0 = 未发生欠压复位
- bit 0 **POR:** 上电复位标志位
1 = 发生了上电复位
0 = 未发生上电复位

注 1: 所有复位状态位都可以用软件置 1 或清零。用软件将这些位中的一位置 1 不会导致器件复位。

2: 如果 FWDTEN 配置位为 1 (未编程)，则 WDT 始终使能，而与 SWDTEN 位的设置无关。

表 6-1：复位标志位操作

标志位	置 1 所表示的事件	清零所表示的事件
TRAPR (RCON<15>)	陷阱冲突事件	POR
IOPUWR (RCON<14>)	非法操作码或访问了未初始化的 W 寄存器	POR
CM (RCON<9>)	配置不匹配复位	POR
EXTR (RCON<7>)	MCLR 复位	POR
SWR (RCON<6>)	RESET 指令	POR
WDTO (RCON<4>)	WDT 超时	PWRSAV 指令和 POR
SLEEP (RCON<3>)	PWRSAV #SLEEP 指令	POR
IDLE (RCON<2>)	PWRSAV #IDLE 指令	POR
BOR (RCON<1>)	POR 和 BOR	—
POR (RCON<0>)	POR	—

注：所有复位标志位均可由用户软件置 1 或清零。

6.1 复位时的时钟源选择

如果使能了时钟切换，器件复位时的系统时钟源选择如表 6-2 中所示。如果禁止了时钟切换功能，则总是根据振荡器配置位选择系统时钟源。更多详细信息，请参见第 8.0 节“振荡器配置”。

表 6-2：不同复位类型的振荡器选择（使能时钟切换功能）

复位类型	确定时钟源的方式
POR	FNOSC 配置位 (CW2<10:8>)
BOR	
MCLR	COSC 控制位 (OSCCON<14:12>)
WDTO	
SWR	

6.2 器件复位时间

表 6-3 总结了各类器件复位的复位时间。请注意，系统复位信号 SYSRST 在 POR 和 PWRT 延时后释放。

器件实际开始执行代码的时间还取决于系统振荡器延时，它包括振荡器起振定时器（OST）延时和 PLL 锁定时间。OST 和 PLL 锁定时间与相应的 SYSRST 延时同时发生。

FSCM 延时决定在 SYSRST 信号发出到 FSCM 开始监视系统时钟源的时间。

PIC24FJ256GB110 系列

表 6-3：各种器件复位的复位延时

复位类型	时钟源	SYSRST 延时	系统时钟延时	注
POR ⁽⁶⁾	EC	TPOR + TPWRT	—	1, 2
	FRC, FRCDIV	TPOR + TPWRT	TFRC	1, 2, 3, 6
	LPRC	TPOR + TPWRT	TLPRC	1, 2, 3
	ECPLL	TPOR + TPWRT	TLOCK	1, 2, 4
	FRCPLL	TPOR + TPWRT	TFRC + TLOCK	1, 2, 3, 4
	XT, HS, SOSC	TPOR+ TPWRT	TOST	1, 2, 5
	XTPLL, HSPLL	TPOR + TPWRT	TOST + TLOCK	1, 2, 4, 5
BOR	EC	TPWRT	—	2
	FRC, FRCDIV	TPWRT	TFRC	2, 3, 6
	LPRC	TPWRT	TLPRC	2, 3
	ECPLL	TPWRT	TLOCK	2, 4
	FRCPLL	TPWRT	TFRC + TLOCK	2, 3, 4
	XT, HS, SOSC	TPWRT	TOST	2, 5
	XTPLL, HSPLL	TPWRT	TFRC + TLOCK	2, 3, 4
所有其他复位	任意时钟	—	—	—

注 1: TPOR = 上电复位延时。

2: 若禁止稳压器 (ENVREG 连接至 Vss)，则 TPWRT = 64 ms (标称值)。

3: TFRC 和 TLPRC = RC 振荡器起振时间。

4: TLOCK = PLL 锁定时间。

5: TOST = 振荡器起振定时器 (OST) 延时。10 位计数器计数 1024 个振荡器周期后，才将振荡器时钟释放给系统使用。

6: 若使能双速启动，无论选择什么主振荡器，器件启动时都将使用 FRC 振荡器。在这种情形下，FRC 起振时间必须考虑在内。

注：欲知详细的工作频率和时序规范，请参见第 29.0 节 “电气特性”。

6.2.1 POR 和长振荡器起振时间

振荡器起振电路及其相关的延时定时器与上电时发生的器件复位延时无关。一些晶振电路（尤其是低频晶振）的起振时间会相对较长。因此，在 SYSRST 信号发出后，可能会发生以下一种或多种情况：

- 振荡电路尚未起振。
- 振荡器起振定时器尚未超时（如果使用了晶振）。
- PLL 未实现锁定（如果使用了 PLL）。

在有效时钟源供系统使用前，器件不会开始执行代码。因此，如果必须确定复位延时，必须考虑到振荡器和 PLL 起振延时。

6.2.2 故障保护时钟监视器（FSCM）和器件复位

如果使能了 FSCM，它将在 SYSRST 释放后开始监视系统时钟源。如果有效时钟源在此时不可用，器件将自动切换至 FRC 振荡器，用户可以切换至陷阱服务程序中要求的晶振。

6.3 特殊功能寄存器的复位状态

大多数与 PIC24F CPU 和外设相关的特殊功能寄存器（SFR）会在器件复位时复位为某个特定值。SFR 是按其外设或 CPU 功能分组的，其复位值在本手册的相应部分说明。

除了 4 个寄存器外，所有其他 SFR 的复位值都与复位类型无关。复位控制寄存器 RCON 的复位值取决于器件复位的类型。振荡器控制寄存器 OSCCON 的复位值取决于复位类型和在闪存配置字 2（CW2）中的 FNOSC 位的编程值（见表 6-2）。RCFGCAL 和 NVMCON 寄存器仅受 POR 影响。

PIC24FJ256GB110 系列

注:

7.0 中断控制器

注: 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见《PIC24F 系列参考手册》的第 8 章“中断”(DS39707A_CN)。

PIC24F 中断控制器将诸多外设中断请求信号缩减到一个到 PIC24F CPU 的中断请求。该控制器具有以下特性

- 多达 8 个处理器异常和软件陷阱
- 7 个可由用户选择的优先级
- 具有多达 118 个向量的中断向量表 (Interrupt Vector Table, IVT)
- 每个中断或异常源对应一个唯一的向量
- 在所指定的用户优先级内具有固定的优先级
- 用于支持调试的备用中断向量表 (Alternate Interrupt Vector Table, AIVT)
- 固定的中断进入和返回延时

7.1 中断向量表

中断向量表 (IVT) 如图 7-1 所示。IVT 位于程序存储器中, 起始存储单元地址是 0000004h。IVT 包含 126 个向量, 由 8 个不可屏蔽的陷阱向量和多达 118 个中断源组成。一般来说, 每个中断源都有自己的中断向量。每个中断向量都包含一个 24 位宽的地址。每个中断向量存储单元中设置的值是其相关的中断服务程序 (Interrupt Service Routine, ISR) 的起始地址。

中断向量有一个自然优先级; 也就是说每个中断向量的优先级与其在向量表中的位置有关。如果其他方面都相同, 较低地址的中断向量具有较高的自然优先级。例如, 与向量 0 相关的中断比任何其他向量地址的中断具有更高的自然优先级。

PIC24FJ256GB110 系列器件实现了不可屏蔽的陷阱和唯一中断。表 7-1 和表 7-2 对此做了总结。

7.1.1 备用中断向量表

备用中断向量表 (AIVT) 位于 IVT 之后, 如图 7-1 所示。ALTIVT 控制位 (INTCON2<15>) 控制对 AIVT 的访问。如果 ALTIVT 位置 1, 则所有的中断和异常处理都将使用备用向量, 而非默认向量。备用向量与默认向量的组织方式相同。

AIVT 支持仿真和调试功能, 它提供了一种不需要将中断向量再编程就可以在应用程序和支持环境之间切换的方法。此特性也支持运行时在不同应用程序之间切换以便评估各种软件算法。如果不需要 AIVT, 则应该用 IVT 中使用的地址设置 AIVT。

7.2 复位过程

器件复位不是真正的异常, 因为复位过程中并不涉及到中断控制器。作为对复位的响应, PIC24F 器件清零其寄存器, 同时强制 PC 为零。然后单片机从地址 000000h 处开始执行程序。用户在复位地址处设定一条 GOTO 指令, 该指令会将程序重定位到相应的启动程序。

注: 应该使用包含 RESET 指令的默认中断处理程序的入口地址编程 IVT 和 AIVT 中所有未实现或未使用的向量单元。

PIC24FJ256GB110 系列

图 7-1: PIC24F 中断向量表

自然优先级降序排列 	复位——GOTO 指令	000000h
	复位——GOTO 地址	000002h
	保留	000004h
	振荡器故障陷阱向量	
	地址错误陷阱向量	
	堆栈错误陷阱向量	
	数学错误陷阱向量	
	保留	
	保留	
	保留	
	中断向量 0	000014h
	中断向量 1	
	—	
	—	
	—	
中断向量表 (IVT) ⁽¹⁾	中断向量 52	00007Ch
	中断向量 53	00007Eh
	中断向量 54	000080h
	—	
	—	
	—	
	中断向量 116	0000FCCh
	中断向量 117	0000FEh
	保留	000100h
	保留	000102h
	保留	
	振荡器故障陷阱向量	
	地址错误陷阱向量	
	堆栈错误陷阱向量	
	数学错误陷阱向量	
备用中断向量表 (AIVT) ⁽¹⁾	保留	
	保留	
	保留	
	中断向量 0	000114h
	中断向量 1	
	—	
	—	
	—	
	中断向量 52	00017Ch
	中断向量 53	00017Eh
	中断向量 54	000180h
	—	
	—	
	—	
	中断向量 116	0001FEh
	中断向量 117	000200h
代码起始单元		

注 1: 请参见表 7-2 了解中断向量列表。

表 7-1: 陷阱向量详细信息

向量编号	IVT 地址	AIVT 地址	陷阱源
0	000004h	000104h	保留
1	000006h	000106h	振荡器故障
2	000008h	000108h	地址错误
3	00000Ah	00010Ah	堆栈错误
4	00000Ch	00010Ch	数学错误
5	00000Eh	00010Eh	保留
6	000010h	000110h	保留
7	000012h	000112h	保留

表 7-2: 已实现的中断向量

中断源	向量编号	IVT 地址	AIVT 地址	中断位位置		
				标志	允许	优先级
ADC1 转换完成	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
比较器事件	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CRC 发生器	67	00009Ah	00019Ah	IFS4<3>	IEC4<3>	IPC16<14:12>
CTMU 事件	77	0000AEh	0001AEh	IFS4<13>	IEC4<13>	IPC19<6:4>
外部中断 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
外部中断 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
外部中断 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
外部中断 3	53	00007Eh	00017Eh	IFS3<5>	IEC3<5>	IPC13<6:4>
外部中断 4	54	000080h	000180h	IFS3<6>	IEC3<6>	IPC13<10:8>
I2C1 主事件	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
I2C1 从事件	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
I2C2 主事件	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
I2C2 从事件	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
I2C3 主事件	85	0000BEh	0001BEh	IFS5<5>	IEC5<5>	IPC21<6:4>
I2C3 从事件	84	0000BCh	0001BCh	IFS5<4>	IEC5<4>	IPC21<2:0>
输入捕捉 1	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
输入捕捉 2	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
输入捕捉 3	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
输入捕捉 4	38	000060h	000160h	IFS2<6>	IEC2<6>	IPC9<10:8>
输入捕捉 5	39	000062h	000162h	IFS2<7>	IEC2<7>	IPC9<14:12>
输入捕捉 6	40	000064h	000164h	IFS2<8>	IEC2<8>	IPC10<2:0>
输入捕捉 7	22	000040h	000140h	IFS1<6>	IEC1<6>	IPC5<10:8>
输入捕捉 8	23	000042h	000142h	IFS1<7>	IEC1<7>	IPC5<14:12>
输入捕捉 9	93	0000CEh	0001CEh	IFS5<13>	IEC5<13>	IPC23<6:4>
输入状态变化通知	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
LVD 低压检测	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC18<2:0>
输出比较 1	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
输出比较 2	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
输出比较 3	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
输出比较 4	26	000048h	000148h	IFS1<10>	IEC1<10>	IPC6<10:8>
输出比较 5	41	000066h	000166h	IFS2<9>	IEC2<9>	IPC10<6:4>
输出比较 6	42	000068h	000168h	IFS2<10>	IEC2<10>	IPC10<10:8>
输出比较 7	43	00006Ah	00016Ah	IFS2<11>	IEC2<11>	IPC10<14:12>
输出比较 8	44	00006Ch	00016Ch	IFS2<12>	IEC2<12>	IPC11<2:0>
输出比较 9	92	0000CCh	0001CCh	IFS5<12>	IEC5<12>	IPC23<2:0>
并行主端口	45	00006Eh	00016Eh	IFS2<13>	IEC2<13>	IPC11<6:4>
实时时钟 / 日历	62	000090h	000190h	IFS3<14>	IEC3<14>	IPC15<10:8>
SPI1 错误	9	000026h	000126h	IFS0<9>	IEC0<9>	IPC2<6:4>
SPI1 事件	10	000028h	000128h	IFS0<10>	IEC0<10>	IPC2<10:8>
SPI2 错误	32	000054h	000154h	IFS2<0>	IEC2<0>	IPC8<2:0>
SPI2 事件	33	000056h	000156h	IFS2<1>	IEC2<1>	IPC8<6:4>
SPI3 错误	90	0000C8h	0001C8h	IFS5<10>	IEC5<10>	IPC22<10:8>
SPI3 事件	91	0000CAh	0001CAh	IFS5<11>	IEC5<11>	IPC22<14:12>

PIC24FJ256GB110 系列

表 7-2：已实现的中断向量（续）

中断源	向量编号	IVT 地址	AIVT 地址	中断位位置		
				标志	允许	优先级
Timer1	3	00001Ah	00011Ah	IFS0<3>	IEC0<3>	IPC0<14:12>
Timer2	7	000022h	000122h	IFS0<7>	IEC0<7>	IPC1<14:12>
Timer3	8	000024h	000124h	IFS0<8>	IEC0<8>	IPC2<2:0>
Timer4	27	00004Ah	00014Ah	IFS1<11>	IEC1<11>	IPC6<14:12>
Timer5	28	00004Ch	00014Ch	IFS1<12>	IEC1<12>	IPC7<2:0>
UART1 错误	65	000096h	000196h	IFS4<1>	IEC4<1>	IPC16<6:4>
UART1 接收器	11	00002Ah	00012Ah	IFS0<11>	IEC0<11>	IPC2<14:12>
UART1 发送器	12	00002Ch	00012Ch	IFS0<12>	IEC0<12>	IPC3<2:0>
UART2 错误	66	000098h	000198h	IFS4<2>	IEC4<2>	IPC16<10:8>
UART2 接收器	30	000050h	000150h	IFS1<14>	IEC1<14>	IPC7<10:8>
UART2 发送器	31	000052h	000152h	IFS1<15>	IEC1<15>	IPC7<14:12>
UART3 错误	81	0000B6h	0001B6h	IFS5<1>	IEC5<1>	IPC20<6:4>
UART3 接收器	82	0000B8h	0001B8h	IFS5<2>	IEC5<2>	IPC20<10:8>
UART3 发送器	83	0000BAh	0001BAh	IFS5<3>	IEC5<3>	IPC20<14:12>
UART4 错误	87	0000C2h	0001C2h	IFS5<7>	IEC5<7>	IPC21<14:12>
UART4 接收器	88	0000C4h	0001C4h	IFS5<8>	IEC5<8>	IPC22<2:0>
UART4 发送器	89	0000C6h	0001C6h	IFS5<9>	IEC5<9>	IPC22<6:4>
USB 中断	86	0000C0h	0001C0h	IFS5<6>	IEC5<6>	IPC21<10:8>

7.3 中断控制和状态寄存器

PIC24FJ256GB110 系列器件共有 37 个用于中断控制器的寄存器：

- INTCON1
- INTCON2
- IFS0 至 IFS5
- IEC0 至 IEC5
- IPC0 至 IPC23 (IPC14 和 IPC17 除外)
- INTTREG

INTCON1 和 INTCON2 控制全局中断。INTCON1 中含有中断嵌套禁止 (NSTDIS) 位，以及处理器陷阱源的控制和状态标志。INTCON2 寄存器控制外部中断请求信号行为以及备用中断向量表的使用。

IFSx 寄存器包含所有中断请求标志。每个中断源都具有一个状态位，该状态位由相应的外设或外部信号置 1，通过软件进行清零。

IECx 寄存器包含所有中断允许位。这些控制位用于单独允许外设或外部信号中断。

IPCx 寄存器用于设置每个中断源的中断优先级。可以给每个用户中断源分配 8 个优先级之一。

INTTREG 寄存器包含相关的中断向量编号和新的 CPU 中断优先级，分别体现在向量编号 (VECNUM<6:0>) 和中断优先级 (ILR<3:0>) 位域中。新的中断优先级是待处理中断的优先级。

中断源按表 7-2 中向量编号的顺序分配给 IFSx、IECx 和 IPCx 寄存器。例如，INT0 (外部中断 0) 表示向量编号为 0，自然优先级为 0 的外部中断。所以 INT0IF 状态位在 IFS0<0> 中，INT0IE 允许位在 IEC0<0> 中，INT0IP<2:0> 优先级位在 IPC0 最前面的位置 (IPC0<2:0>) 中。

尽管两个 CPU 控制寄存器不是中断控制硬件的特定组成部分，但它们仍包含控制中断功能的位。ALU 状态寄存器 (SR) 包含 IPL<2:0> 位 (SR<7:5>)。这些位表示当前 CPU 中断优先级。用户可以通过写 IPL 位来更改当前 CPU 优先级。

CORCON 寄存器包含 IPL3 位，这个位与 IPL<2:0> 位一起表示当前 CPU 优先级。IPL3 是只读位，所以用户软件无法屏蔽陷阱事件。

在下面各页中的寄存器 7-1 到寄存器 7-39 说明了所有的中断寄存器。

寄存器 7-1: SR: ALU 状态寄存器 (在 CPU 中)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0
—	—	—	—	—	—	—	DC ⁽¹⁾
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2 ^(2,3)	IPL1 ^(2,3)	IPL0 ^(2,3)	RA ⁽¹⁾	N ⁽¹⁾	OV ⁽¹⁾	Z ⁽¹⁾	C ⁽¹⁾
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-5 **IPL<2:0>**: CPU 中断优先级状态位 ^(2,3)

111 = CPU 中断优先级为 7 (15)。禁止用户中断。

110 = CPU 中断优先级为 6 (14)

101 = CPU 中断优先级为 5 (13)

100 = CPU 中断优先级为 4 (12)

011 = CPU 中断优先级为 3 (11)

010 = CPU 中断优先级为 2 (10)

001 = CPU 中断优先级为 1 (9)

000 = CPU 中断优先级为 0 (8)

注 1: 关于那些不是专用于中断控制功能的保留位的说明, 请参见寄存器 3-1。

2: IPL 位与 IPL3 位 (CORCON<3>) 组合形成 CPU 中断优先级。如果 IPL3 = 1, 那么括号中的值表示中断优先级。

3: 当 NSTDIS (INTCON1<15>) = 1 时, IPL 状态位是只读的。

寄存器 7-2: CORCON: CPU 控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0	R/W-0	U-0	U-0
—	—	—	—	IPL3 ⁽²⁾	PSV ⁽¹⁾	—	—
bit 7							bit 0

图注:

C = 可清零位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 3 **IPL3**: CPU 中断优先级状态位 ⁽²⁾

1 = CPU 中断优先级大于 7

0 = CPU 中断优先级等于或小于 7

注 1: 关于那些不是专用于中断控制功能的保留位的说明, 请参见寄存器 3-2。

2: IPL3 位与 IPL<2:0> 位 (SR<7:5>) 组合形成 CPU 中断优先级。

PIC24FJ256GB110 系列

寄存器 7-3: INTCON1: 中断控制寄存器 1

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
NSTDIS	—	—	—	—	—	—	—
bit 15	bit 8						

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **NSTDIS:** 中断嵌套禁止位

1 = 禁止中断嵌套

0 = 使能中断嵌套

bit 14-5 未实现: 读为 0

bit 4 **MATHERR:** 算术错误陷阱状态位

1 = 发生了溢出陷阱

0 = 未发生溢出陷阱

bit 3 **ADDRERR:** 地址错误陷阱状态位

1 = 发生了地址错误陷阱

0 = 未发生地址错误陷阱

bit 2 **STKERR:** 堆栈错误陷阱状态位

1 = 发生了堆栈错误陷阱

0 = 未发生堆栈错误陷阱

bit 1 **OSCFAIL:** 振荡器故障陷阱状态位

1 = 发生了振荡器故障陷阱

0 = 未发生振荡器故障陷阱

bit 0 未实现: 读为 0

寄存器 7-4: INTCON2: 中断控制寄存器 2

R/W-0	R-0	U-0	U-0	U-0	U-0	U-0	U-0
ALTIVT	DISI	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **ALTIVT:** 备用中断向量表使能位

1 = 使用备用中断向量表

0 = 使用标准 (默认) 向量表

bit 14 **DISI:** DISI 指令状态位

1 = 执行了 DISI 指令

0 = 没有执行 DISI 指令

bit 13-5 未实现: 读为 0

bit 4 **INT4EP:** 外部中断 4 边沿检测极性选择位

1 = 下降沿中断

0 = 上升沿中断

bit 3 **INT3EP:** 外部中断 3 边沿检测极性选择位

1 = 下降沿中断

0 = 上升沿中断

bit 2 **INT2EP:** 外部中断 2 边沿检测极性选择位

1 = 下降沿中断

0 = 上升沿中断

bit 1 **INT1EP:** 外部中断 1 边沿检测极性选择位

1 = 下降沿中断

0 = 上升沿中断

bit 0 **INT0EP:** 外部中断 0 边沿检测极性选择位

1 = 下降沿中断

0 = 上升沿中断

PIC24FJ256GB110 系列

寄存器 7-5： IFS0： 中断标志状态寄存器 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-14 未实现：读为 0
- bit 13 **AD1IF:** A/D 转换完成中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 12 **U1TXIF:** UART1 发送器中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 11 **U1RXIF:** UART1 接收器中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 10 **SPI1IF:** SPI1 事件中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 9 **SPF1IF:** SPI1 故障中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 8 **T3IF:** Timer3 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 7 **T2IF:** Timer2 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 6 **OC2IF:** 输出比较通道 2 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 5 **IC2IF:** 输入捕捉通道 2 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 4 未实现：读为 0
- bit 3 **T1IF:** Timer1 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 2 **OC1IF:** 输出比较通道 1 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 1 **IC1IF:** 输入捕捉通道 1 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 0 **INT0IF:** 外部中断 0 标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

寄存器 7-6: IFS1: 中断标志状态寄存器 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	—
bit 15	bit 8						

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IC8IF	IC7IF	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- | | |
|--------|--|
| bit 15 | U2TXIF: UART2 发送器中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 14 | U2RXIF: UART2 接收器中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 13 | INT2IF: 外部中断 2 标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 12 | T5IF: Timer5 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 11 | T4IF: Timer4 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 10 | OC4IF: 输出比较通道 4 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 9 | OC3IF: 输出比较通道 3 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 8 | 未实现: 读为 0 |
| bit 7 | IC8IF: 输入捕捉通道 8 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 6 | IC7IF: 输入捕捉通道 7 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 5 | 未实现: 读为 0 |
| bit 4 | INT1IF: 外部中断 1 标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 3 | CNIF: 输入电平变化通知中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 2 | CMIF: 比较器中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 1 | MI2C1IF: 主 I2C1 事件中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 0 | SI2C1IF: 从 I2C1 事件中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |

PIC24FJ256GB110 系列

寄存器 7-7: IFS2: 中断标志状态寄存器 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	PMPIF	OC8IF	OC7IF	OC6IF	OC5IF	IC6IF
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-14 未实现: 读为 0
- bit 13 **PMPIF:** 并行主端口中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 12 **OC8IF:** 输出比较通道 8 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 11 **OC7IF:** 输出比较通道 7 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 10 **OC6IF:** 输出比较通道 6 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 9 **OC5IF:** 输出比较通道 5 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 8 **IC6IF:** 输入捕捉通道 6 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 7 **IC5IF:** 输入捕捉通道 5 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 6 **IC4IF:** 输入捕捉通道 4 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 5 **IC3IF:** 输入捕捉通道 3 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 4-2 未实现: 读为 0
- bit 1 **SPI2IF:** SPI2 事件中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 0 **SPF2IF:** SPI2 故障中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

寄存器 7-8: IFS3: 中断标志状态寄存器 3

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	RTCIF	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
—	INT4IF	INT3IF	—	—	MI2C2IF	SI2C2IF	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- | | |
|----------|--|
| bit 15 | 未实现: 读为 0 |
| bit 14 | RTCIF: 实时时钟 / 日历中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 13-7 | 未实现: 读为 0 |
| bit 6 | INT4IF: 外部中断 4 标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 5 | INT3IF: 外部中断 3 标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 4-3 | 未实现: 读为 0 |
| bit 2 | MI2C2IF: 主 I2C2 事件中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 1 | SI2C2IF: 从 I2C2 事件中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求 |
| bit 0 | 未实现: 读为 0 |

PIC24FJ256GB110 系列

寄存器 7-9: IFS4: 中断标志状态寄存器 4

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	CTMUIF	—	—	—	—	LVDIF
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	CRCIF	U2ERIF	U1ERIF	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13 **CTMUIF:** CTMU 中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 12-9 未实现: 读为 0

bit 8 **LVDIF:** 低压检测中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 7-4 未实现: 读为 0

bit 3 **CRCIF:** CRC 发生器中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 2 **U2ERIF:** UART2 错误中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 1 **U1ERIF:** UART1 错误中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 0 未实现: 读为 0

寄存器 7-10: IFS5: 中断标志状态寄存器 5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	IC9IF	OC9IF	SPI3IF	SPF3IF	U4TXIF	U4RXIF
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U4ERIF	USB1IF	MI2C3IF	SI2C3IF	U3TXIF	U3RXIF	U3ERIF	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-14 未实现: 读为 0
- bit 13 **IC9IF:** 输入捕捉通道 9 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 12 **OC9IF:** 输出比较通道 9 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 11 **SPI3IF:** SPI3 事件中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 10 **SPF3IF:** SPI3 故障中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 9 **U4TXIF:** UART4 发送器中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 8 **U4RXIF:** UART4 接收器中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 7 **U4ERIF:** UART4 错误中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 6 **USB1IF:** USB1 (USB OTG) 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 5 **MI2C3IF:** 主 I2C3 事件中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 4 **SI2C3IF:** 从 I2C3 事件中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 3 **U3TXIF:** UART3 发送器中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 2 **U3RXIF:** UART3 接收器中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 1 **U3ERIF:** UART3 错误中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 0 未实现: 读为 0

PIC24FJ256GB110 系列

寄存器 7-11： IEC0： 中断允许控制寄存器 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-14 未实现：读为 0
- bit 13 **AD1IE:** A/D 转换完成中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 12 **U1TXIE:** UART1 发送器中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 11 **U1RXIE:** UART1 接收器中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 10 **SPI1IE:** SPI1 传输完成中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 9 **SPF1IE:** SPI1 故障中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 8 **T3IE:** Timer3 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 7 **T2IE:** Timer2 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 6 **OC2IE:** 输出比较通道 2 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 5 **IC2IE:** 输入捕捉通道 2 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 4 未实现：读为 0
- bit 3 **T1IE:** Timer1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 2 **OC1IE:** 输出比较通道 1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 1 **IC1IE:** 输入捕捉通道 1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 0 **INT0IE:** 外部中断 0 允许位
1 = 允许中断请求
0 = 禁止中断请求

寄存器 7-12: IEC1: 中断允许控制寄存器 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U2TXIE	U2RXIE	INT2IE ⁽¹⁾	T5IE	T4IE	OC4IE	OC3IE	—
bit 15	bit 8						

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IC8IE	IC7IE	—	INT1IE ⁽¹⁾	CNIE	CMIE	MI2C1IE	SI2C1IE
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15	U2TXIE: UART2 发送器中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 14	U2RXIE: UART2 接收器中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 13	INT2IE: 外部中断 2 允许位 ⁽¹⁾ 1 = 允许中断请求 0 = 禁止中断请求
bit 12	T5IE: Timer5 中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 11	T4IE: Timer4 中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 10	OC4IE: 输出比较通道 4 中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 9	OC3IE: 输出比较通道 3 中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 8	未实现: 读为 0
bit 7	IC8IE: 输入捕捉通道 8 中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 6	IC7IE: 输入捕捉通道 7 中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 5	未实现: 读为 0
bit 4	INT1IE: 外部中断 1 允许位 ⁽¹⁾ 1 = 允许中断请求 0 = 禁止中断请求
bit 3	CNIE: 输入电平变化通知中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 2	CMIE: 比较器中断允许位 1 = 允许中断请求 0 = 禁止中断请求

注 1: 如果允许了外部中断, 则中断输入必须也配置为可用的 RPn 或 PRIn 引脚。更多信息, 请参见第 10.4 节“外设引脚选择”。

PIC24FJ256GB110 系列

寄存器 7-12: IEC1: 中断允许控制寄存器 1 (续)

bit 1 **MI2C1IE:** 主 I2C1 事件中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 0 **SI2C1IE:** 从 I2C1 事件中断允许位

1 = 允许中断请求

0 = 禁止中断请求

注 1: 如果允许了外部中断，则中断输入必须也配置为可用的 RPn 或 PRIn 引脚。更多信息，请参见第 10.4 节“外设引脚选择”。

寄存器 7-13: IEC2: 中断允许控制寄存器 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	PMPIE	OC8IE	OC7IE	OC6IE	OC5IE	IC6IE
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
IC5IE	IC4IE	IC3IE	—	—	—	SPI2IE	SPF2IE
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-14 未实现: 读为 0
- bit 13 **PMPIE:** 并行主端口中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 12 **OC8IE:** 输出比较通道 8 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 11 **OC7IE:** 输出比较通道 7 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 10 **OC6IE:** 输出比较通道 6 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 9 **OC5IE:** 输出比较通道 5 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 8 **IC6IE:** 输入捕捉通道 6 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 7 **IC5IE:** 输入捕捉通道 5 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 6 **IC4IE:** 输入捕捉通道 4 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 5 **IC3IE:** 输入捕捉通道 3 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 4-2 未实现: 读为 0
- bit 1 **SPI2IE:** SPI2 事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 0 **SPF2IE:** SPI2 故障中断允许位
1 = 允许中断请求
0 = 禁止中断请求

PIC24FJ256GB110 系列

寄存器 7-14: IEC3: 中断允许控制寄存器 3

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	RTCIE	—	—	—	—	—	—
bit 15	bit 8						

U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
—	INT4IE ⁽¹⁾	INT3IE ⁽¹⁾	—	—	MI2C2IE	SI2C2IE	—
bit 7	bit 0						

图注:

R = 可读位

-n = POR 时的值

W = 可写位

1 = 置 1

U = 未实现位, 读为 0

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14 **RTCIE:** 实时时钟 / 日历中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 13-7 未实现: 读为 0

bit 6 **INT4IE:** 外部中断 4 允许位⁽¹⁾

1 = 允许中断请求

0 = 禁止中断请求

bit 5 **INT3IE:** 外部中断 3 允许位⁽¹⁾

1 = 允许中断请求

0 = 禁止中断请求

bit 4-3 未实现: 读为 0

bit 2 **MI2C2IE:** 主 I2C2 事件中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 1 **SI2C2IE:** 从 I2C2 事件中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 0 未实现: 读为 0

注 1: 如果允许了某个外部中断, 则中断输入必须也配置给可用的 RPn 或 PRIn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

寄存器 7-15: IEC4: 中断允许控制寄存器 4

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	CTMUIE	—	—	—	—	LVDIE
bit 15							
bit 8							

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	CRCIE	U2ERIE	U1ERIE	—
bit 7							
bit 0							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13 **CTMUIE:** CTMU 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 12-9 未实现: 读为 0

bit 8 **LVDIE:** 低压检测中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 7-4 未实现: 读为 0

bit 3 **CRCIE:** CRC 发生器中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 2 **U2ERIE:** UART2 错误中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 1 **U1ERIE:** UART1 错误中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 0 未实现: 读为 0

PIC24FJ256GB110 系列

寄存器 7-16: IEC5: 中断允许控制寄存器 5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	IC9IE	OC9IE	SPI3IE	SPF3IE	U4TXIE	U4RXIE
bit 15							

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U4ERIE	USB1IE	MI2C3IE	SI2C3IE	U3TXIE	U3RXIE	U3ERIE	—
bit 7							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-14 未实现: 读为 0
- bit 13 **IC9IE:** 输入捕捉通道 9 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 12 **OC9IE:** 输出比较通道 9 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 11 **SPI3IE:** SPI3 事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 10 **SPF3IE:** SPI3 故障中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 9 **U4TXIE:** UART4 发送器中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 8 **U4RXIE:** UART4 接收器中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 7 **U4ERIE:** UART4 错误中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 6 **USB1IE:** USB1 (USB OTG) 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 5 **MI2C3IE:** 主 I2C3 事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 4 **SI2C3IE:** 从 I2C3 事件中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 3 **U3TXIE:** UART3 发送器中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 2 **U3RXIE:** UART3 接收器中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 1 **U3ERIE:** UART3 错误中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 0 未实现: 读为 0

寄存器 7-17： IPC0： 中断优先级控制寄存器 0

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现：读为 0

bit 14-12 **T1IP<2:0>**: Timer1 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现：读为 0

bit 10-8 **OC1IP<2:0>**: 输出比较通道 1 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现：读为 0

bit 6-4 **IC1IP<2:0>**: 输入捕捉通道 1 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现：读为 0

bit 2-0 **INT0IP<2:0>**: 外部中断 0 优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

PIC24FJ256GB110 系列

寄存器 7-18: IPC1: 中断优先级控制寄存器 1

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	IC2IP2	IC2IP1	IC2IP0	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **T2IP<2:0>**: Timer2 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现: 读为 0

bit 10-8 **OC2IP<2:0>**: 输出比较通道 2 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **IC2IP<2:0>**: 输入捕捉通道 2 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

寄存器 7-19: IPC2: 中断优先级控制寄存器 2

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- | | |
|-----------|---|
| bit 15 | 未实现: 读为 0 |
| bit 14-12 | U1RXIP<2:0>: UART1 接收器中断优先级位
111 = 中断优先级为 7 (最高优先级中断)
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源 |
| bit 11 | 未实现: 读为 0 |
| bit 10-8 | SPI1IP<2:0>: SPI1 事件中断优先级位
111 = 中断优先级为 7 (最高优先级中断)
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源 |
| bit 7 | 未实现: 读为 0 |
| bit 6-4 | SPF1IP<2:0>: SPI1 故障中断优先级位
111 = 中断优先级为 7 (最高优先级中断)
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源 |
| bit 3 | 未实现: 读为 0 |
| bit 2-0 | T3IP<2:0>: Timer3 中断优先级位
111 = 中断优先级为 7 (最高优先级中断)
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源 |

PIC24FJ256GB110 系列

寄存器 7-20: IPC3: 中断优先级控制寄存器 3

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-7 未实现: 读为 0

bit 6-4 **AD1IP<2:0>**: A/D 转换完成中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 **U1TXIP<2:0>**: UART1 发送器中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

寄存器 7-21：IPC4：中断优先级控制寄存器 4

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0
bit 15							

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	MI2C1P2	MI2C1P1	MI2C1P0	—	SI2C1P2	SI2C1P1	SI2C1P0
bit 7							

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- | | |
|-----------|--|
| bit 15 | 未实现：读为 0 |
| bit 14-12 | CNIP<2:0> ：输入电平变化通知中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源 |
| bit 11 | 未实现：读为 0 |
| bit 10-8 | CMIP<2:0> ：比较器中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源 |
| bit 7 | 未实现：读为 0 |
| bit 6-4 | MI2C1P<2:0> ：主 I2C1 事件中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源 |
| bit 3 | 未实现：读为 0 |
| bit 2-0 | SI2C1P<2:0> ：从 I2C1 事件中断优先级位
111 = 中断优先级为 7（最高优先级中断）
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源 |

PIC24FJ256GB110 系列

寄存器 7-22: IPC5: 中断优先级控制寄存器 5

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC8IP2	IC8IP1	IC8IP0	—	IC7IP2	IC7IP1	IC7IP0
bit 15	bit 8						

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **IC8IP<2:0>**: 输入捕捉通道 8 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现: 读为 0

bit 10-8 **IC7IP<2:0>**: 输入捕捉通道 7 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7-3 未实现: 读为 0

bit 2-0 **INT1IP<2:0>**: 外部中断 1 优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

寄存器 7-23: IPC6: 中断优先级控制寄存器 6

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **T4IP<2:0>**: Timer4 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现: 读为 0

bit 10-8 **OC4IP<2:0>**: 输出比较通道 4 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **OC3IP<2:0>**: 输出比较通道 3 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

PIC24FJ256GB110 系列

寄存器 7-24: IPC7: 中断优先级控制寄存器 7

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **U2TXIP<2:0>**: UART2 发送器中断优先级位
111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现: 读为 0

bit 10-8 **U2RXIP<2:0>**: UART2 接收器中断优先级位
111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **INT2IP<2:0>**: 外部中断 2 优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 **T5IP<2:0>**: Timer5 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

寄存器 7-25: IPC8: 中断优先级控制寄存器 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-7 未实现: 读为 0

bit 6-4 **SPI2IP<2:0>: SPI2 事件中断优先级位**

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 **SPF2IP<2:0>: SPI2 故障中断优先级位**

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

PIC24FJ256GB110 系列

寄存器 7-26: IPC9: 中断优先级控制寄存器 9

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	IC3IP2	IC3IP1	IC3IP0	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **IC5IP<2:0>**: 输入捕捉通道 5 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现: 读为 0

bit 10-8 **IC4IP<2:0>**: 输入捕捉通道 4 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **IC3IP<2:0>**: 输入捕捉通道 3 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

寄存器 7-27: IPC10: 中断优先级控制寄存器 10

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC7IP2	OC7IP1	OC7IP0	—	OC6IP2	OC6IP1	OC6IP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC5IP2	OC5IP1	OC5IP0	—	IC6IP2	IC6IP1	IC6IP0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- | | |
|-----------|---|
| bit 15 | 未实现: 读为 0 |
| bit 14-12 | OC7IP<2:0> : 输出比较通道 7 中断优先级位
111 = 中断优先级为 7 (最高优先级中断)
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源 |
| bit 11 | 未实现: 读为 0 |
| bit 10-8 | OC6IP<2:0> : 输出比较通道 6 中断优先级位
111 = 中断优先级为 7 (最高优先级中断)
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源 |
| bit 7 | 未实现: 读为 0 |
| bit 6-4 | OC5IP<2:0> : 输出比较通道 5 中断优先级位
111 = 中断优先级为 7 (最高优先级中断)
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源 |
| bit 3 | 未实现: 读为 0 |
| bit 2-0 | IC6IP<2:0> : 输入捕捉通道 6 中断优先级位
111 = 中断优先级为 7 (最高优先级中断)
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源 |

PIC24FJ256GB110 系列

寄存器 7-28: IPC11: 中断优先级控制寄存器 11

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	PMPIP2	PMPIP1	PMPIP0	—	OC8IP2	OC8IP1	OC8IP0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-7 未实现: 读为 0

bit 6-4 **PMPIP<2:0>**: 并行主端口中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 **OC8IP<2:0>**: 输出比较通道 8 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

寄存器 7-29: IPC12: 中断优先级控制寄存器 12

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	MI2C2P2	MI2C2P1	MI2C2P0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	SI2C2P2	SI2C2P1	SI2C2P0	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10-8 **MI2C2P<2:0>**: 主 I2C2 事件中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **SI2C2P<2:0>**: 从 I2C2 事件中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

PIC24FJ256GB110 系列

寄存器 7-30: IPC13: 中断优先级控制寄存器 13

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT4IP2	INT4IP1	INT4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	INT3IP2	INT3IP1	INT3IP0	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10-8 INT4IP<2:0>: 外部中断 4 优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 INT3IP<2:0>: 外部中断 3 优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

寄存器 7-31：IPC15：中断优先级控制寄存器 15

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现：读为 0

bit 10-8 **RTCIP<2:0>** 实时时钟 / 日历中断优先级位

111 = 中断优先级为 7（最高优先级中断）

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7-0 未实现：读为 0

PIC24FJ256GB110 系列

寄存器 7-32: IPC16: 中断优先级控制寄存器 16

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **CRCIP<2:0>**: CRC 发生器错误中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现: 读为 0

bit 10-8 **U2ERIP<2:0>**: UART2 错误中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **U1ERIP<2:0>**: UART1 错误中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

寄存器 7-33: IPC18: 中断优先级控制寄存器 18

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15	bit 8						

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-3 未实现: 读为 0

bit 2-0 **LVDIP<2:0>**: 低压检测中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

寄存器 7-34: IPC19: 中断优先级控制寄存器 19

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-7 未实现: 读为 0

bit 6-4 **CTMUIP<2:0>**: CTMU 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

PIC24FJ256GB110 系列

寄存器 7-35: IPC20: 中断优先级控制寄存器 20

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **U3TXIP<2:0>: UART3 发送器中断优先级位**

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现: 读为 0

bit 10-8 **U3RXIP<2:0>: UART3 接收器中断优先级位**

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **U3ERIP<2:0>: UART3 错误中断优先级位**

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

寄存器 7-36: IPC21: 中断优先级控制寄存器 21

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U4ERIP2	U4ERIP1	U4ERIP0	—	USB1IP2	USB1IP1	USB1IP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	MI2C3P2	MI2C3P1	MI2C3P0	—	SI2C3P2	SI2C3P1	SI2C3P0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 未实现: 读为 0
- bit 14-12 **U4ERIP<2:0>**: UART4 错误中断优先级位
111 = 中断优先级为 7 (最高优先级中断)
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源
- bit 11 未实现: 读为 0
- bit 10-8 **USB1IP<2:0>**: USB1 (USB OTG) 中断优先级位
111 = 中断优先级为 7 (最高优先级中断)
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源
- bit 7 未实现: 读为 0
- bit 6-4 **MI2C3P<2:0>**: 主 I2C3 事件中断优先级位
111 = 中断优先级为 7 (最高优先级中断)
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源
- bit 3 未实现: 读为 0
- bit 2-0 **SI2C3P<2:0>**: 从 I2C3 事件中断优先级位
111 = 中断优先级为 7 (最高优先级中断)
•
•
•
001 = 中断优先级为 1
000 = 禁止中断源

PIC24FJ256GB110 系列

寄存器 7-37: IPC22: 中断优先级控制寄存器 22

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPI3IP2	SPI3IP1	SPI3IP0	—	SPF3IP2	SPF3IP1	SPF3IP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U4TXIP2	U4TXIP1	U4TXIP0	—	U4RXIP2	U4RXIP1	U4RXIP0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **SPI3IP<2:0>**: SPI3 事件中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现: 读为 0

bit 10-8 **SPF3IP<2:0>**: SPI3 故障中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **U4TXIP<2:0>**: UART4 发送器中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 **U4RXIP<2:0>**: UART4 接收器中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

寄存器 7-38: IPC23: 中断优先级控制寄存器 23

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC9IP2	IC9IP1	IC9IP0	—	OC9IP2	OC9IP1	OC9IP0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-7 未实现: 读为 0

bit 6-4 **IC9IP<2:0>**: 输入捕捉通道 9 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 **OC9IP<2:0>**: 输出比较通道 9 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

PIC24FJ256GB110 系列

寄存器 7-39: INTTREG: 中断控制和状态寄存器

R-0	U-0	R/W-0	U-0	R-0	R-0	R-0	R-0
CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0
bit 15	bit 8						

U-0	R-0						
—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **CPUIRQ:** 来自中断控制器的 CPU 中断请求位
1 = 产生了中断请求, 但尚未被 CPU 响应; 此情况发生在 CPU 优先级高于中断优先级的情况下
0 = 没有尚未响应的中断请求
- bit 14 未实现: 读为 0
- bit 13 **VHOLD:** 向量编号捕捉配置位
1 = VECNUM 包含优先级最高的待处理中断的向量值
0 = VECNUM 包含上次响应的中断的向量值 (指的是上次发生的优先级高于 CPU 的中断, 即使有其他中断等待处理也是如此)
- bit 12 未实现: 读为 0
- bit 11-8 **ILR<3:0>:** 新的 CPU 中断优先级位
1111 = CPU 中断优先级为 15
•
•
•
0001 = CPU 中断优先级为 1
0000 = CPU 中断优先级为 0
- bit 7 未实现: 读为 0
- bit 6-0 **VECNUM<6:0>:** 待处理中断向量 ID 位 (待处理中断的向量编号为 VECNUM + 8)
0111111 = 待处理中断向量的编号为 135
•
•
•
0000001 = 待处理中断向量的编号为 9
0000000 = 待处理中断向量的编号为 8

7.4 中断设置过程

7.4.1 初始化

要配置中断源：

1. 如果不需要嵌套中断，则将 **NSTDIS** 控制位 (**INTCON1<15>**) 置 1。
2. 通过写相应 **IPCx** 寄存器中的控制位为中断源选择由用户分配的优先级。优先级将取决于具体的应用和中断源类型。如果不需要多重优先级，则可以将所有允许的中断源的 **IPCx** 寄存器控制位设定为同一个非零值。
3. 将相应 **IFSx** 寄存器中与外设相关的中断标志状态位清零。
4. 通过将相应 **IECx** 寄存器中与中断源相关的中断允许控制位置 1 来允许中断源。

注： 在器件复位时，**IPCx** 寄存器被初始化，为所有用户中断源分配优先级 4。

7.4.2 中断服务程序

用于声明 **ISR** 和使用正确向量地址初始化 **IVT** 的方法取决于编程语言（即，C 语言或汇编语言）和用于开发应用程序的语言开发工具套件。通常，用户必须清零 **IFSx** 寄存器中与 **ISR** 处理的中断源相对应的中断标志。否则，在从 **ISR** 退出后将立即重新进入 **ISR**。如果 **ISR** 用汇编语言编码，则必须使用 **RETFIE** 指令结束 **ISR**，以便将保存的 **PC** 值、**SRL** 值和原先的 **CPU** 优先级弹出堆栈。

7.4.3 陷阱服务程序

除了必须清零 **INTCON1** 寄存器中相应的陷阱状态标志来避免重新进入陷阱服务程序 (Trap Service Routine, **TSR**) 之外，**TSR** 使用与 **ISR** 类似的方式编写。

7.4.4 中断禁止

所有的用户中断均可以使用以下步骤禁止：

1. 使用 **PUSH** 指令将当前的 **SR** 值压入软件堆栈。
2. 通过将值 **E0h** 与 **SRL** 进行逻辑或运算来强制将 **CPU** 的优先级设置为 7。

如果要允许用户中断，则可以使用 **POP** 指令来恢复先前的 **SR** 值。

请注意，只能禁止优先级小于等于 7 的用户中断。不能禁止陷阱源（优先级为 8-15）。

使用 **DISI** 指令可以方便地将优先级为 1-6 的中断禁止一段固定的时间。**DISI** 指令不能禁止优先级为 7 的中断源。

PIC24FJ256GB110 系列

注:

8.0 振荡器配置

注：本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》的第 6 章“振荡器”(DS39700A_CN)。

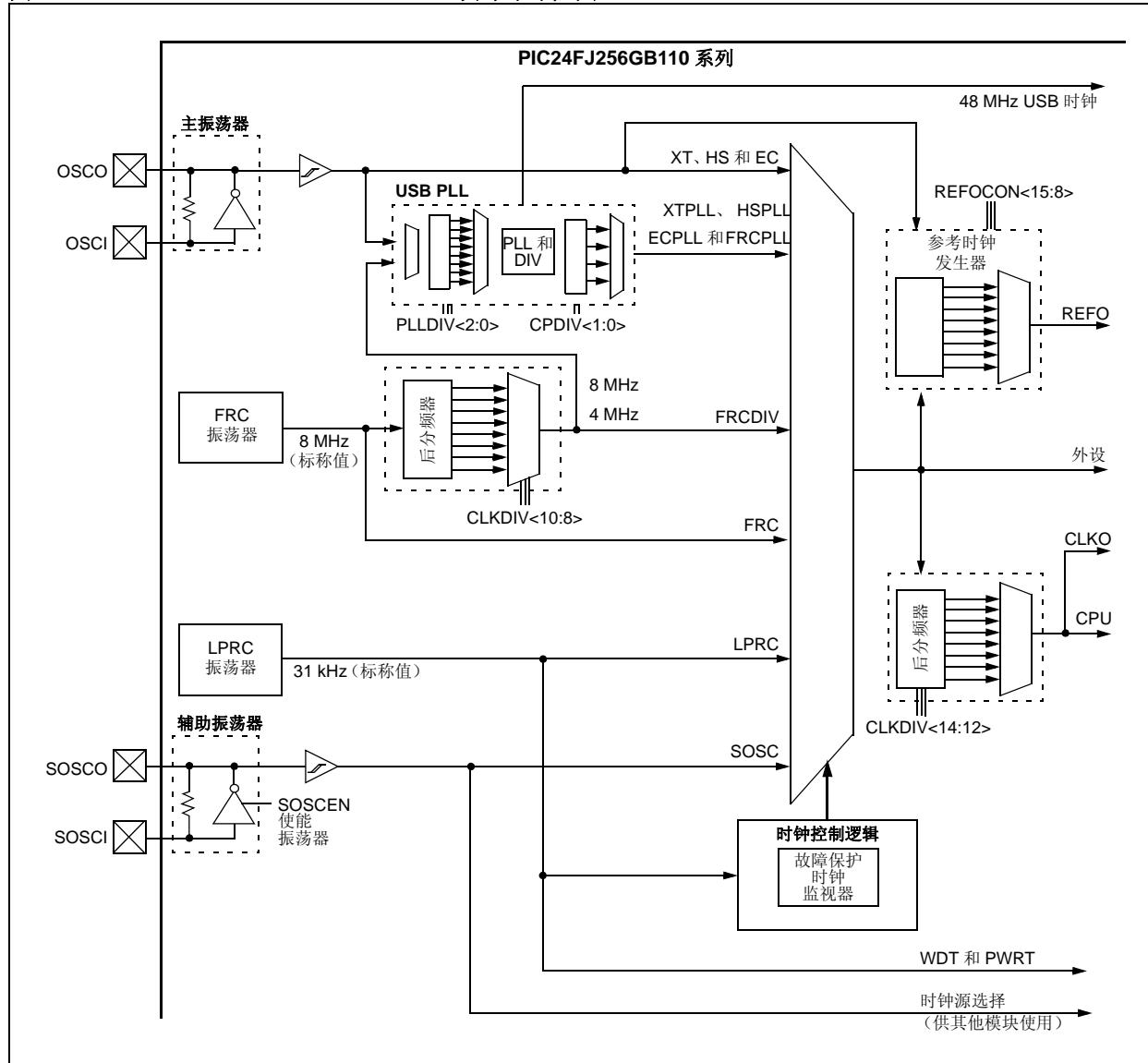
PIC24FJ256GB110 系列器件的振荡器系统具有以下特性：

- 可选择 4 个外部和内部振荡器作为时钟源，提供 11 种不同的时钟模式

- 一个片上 USB PLL 模块，为 USB 模块提供稳定的 48 MHz 时钟，并为系统时钟提供较大范围的频率选择
- 可通过软件控制在各种时钟源之间切换
- 可通过软件控制预分频器有选择地为 CPU 提供时钟，以节省系统功耗
- 具有故障保护时钟监视器 (Fail-Safe Clock Monitor, FSCM)，可检测时钟故障，以便应用可安全地恢复或关闭
- 用于同步外部硬件的可单独使用且独立配置的系统时钟输出

图 8-1 给出了振荡器系统的简化框图。

图 8-1：PIC24FJ256GB110 系列时钟框图



8.1 CPU 时钟机制

系统时钟源可以由以下 4 种之一提供：

- OSCI 和 OSCO 引脚上的主振荡器（POSC）
- SOSCI 和 SOSCO 引脚上的辅助振荡器（SOSC）
- 内部快速 RC（FRC）振荡器
- 内部低功耗 RC（LPRC）振荡器

主振荡器和 FRC 源可以选择使用内部 USB PLL 模块，用来产生 USB 模块时钟和来自 96 MHZ PLL 的独立系统时钟。更多信息，请参见第 8.5 节“振荡器模式和 USB 操作”。

快速内部 FRC 提供 8 MHz 时钟源。可选择用可编程时钟分频降低它，从而提供一系列系统时钟频率。

选定的时钟源将产生处理器和外设时钟。处理器时钟源需进行二分频，以产生内部指令周期时钟 FCY。在本文档中，指令周期时钟也表示为 Fosc/2。内部指令周期时钟 Fosc/2 可以在 OSCO I/O 引脚提供，用于主振荡器的一些工作模式。

表 8-1：时钟选择的配置位值

振荡器模式	振荡器源	POSCMD<1:0>	FNOSC<2:0>	注
带后分频器的快速 RC 振荡器（FRCDIV）	内部	11	111	1, 2
（保留）	内部	xx	110	1
低功耗 RC 振荡器（LPRC）	内部	11	101	1
辅助（Timer1）振荡器（SOSC）	辅助	11	100	1
带 PLL 模块的主振荡器（XT）（XTPLL）	主	01	011	
带 PLL 模块的主振荡器（EC）（ECPLL）	主	00	011	
主振荡器（HS）	主	10	010	
主振荡器（XT）	主	01	010	
主振荡器（EC）	主	00	010	
带 PLL 模块的快速 RC 振荡器（FRCPLL）	内部	11	001	1
快速 RC 振荡器（FRC）	内部	11	000	1

注 1：OSCO 引脚功能由 OSCIOFCN 配置位决定。

2：这是未编程（已擦除）器件的默认振荡器模式。

8.2 POR 时的初始配置

在发生器件上电复位事件时使用的振荡器源（以及工作模式）使用配置位设置进行选择。振荡器配置位在程序存储器中的配置寄存器中进行设置（更多详细信息，请参见第 26.1 节“配置位”）。主振荡器配置位 POSCMD<1:0>（配置字 2<1:0>）和初始振荡器选择配置位 FNOSC<2:0>（配置字 2<10:8>）用于选择在上电复位时使用的振荡器源。默认情况下（未编程时）将选择带后分频器的 FRC 主振荡器（FRCDIV）。通过编程这些位，可以选择辅助振荡器或一个内部振荡器。这些配置位使用户可以选择多种时钟模式，如表 8-1 所示。

8.2.1 时钟切换模式配置位

FCKSM 配置位（配置字 2<7:6>）用于联合配置器件时钟切换和故障保护时钟监视器（FSCM）。只有将 FCKSM1 编程（0）时，才会使能时钟切换。只有同时将 FCKSM<1:0> 编程（00）时，才会使能 FSCM。

8.3 控制寄存器

振荡器的操作由 3 个特殊功能寄存器控制：

- OSCCON
- CLKDIV
- OSCTUN

OSCCON 寄存器（寄存器 8-1）是振荡器的主要控制寄存器。它控制时钟源切换和时钟源监控。

CLKDIV 寄存器（寄存器 8-2）控制与打盹模式相关功能，以及 FRC 振荡器的后分频器。OSCTUN 寄存器（寄存器 8-3）使用户可以对 FRC 振荡器在大约 $\pm 12\%$ 的范围内进行微调。

寄存器 8-1： OSCCON：振荡器控制寄存器

U-0	R-0	R-0	R-0	U-0	R/W-x ⁽¹⁾	R/W-x ⁽¹⁾	R/W-x ⁽¹⁾
—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
bit 15	bit 8						

R/SO-0	R/W-0	R-0 ⁽³⁾	U-0	R/CO-0	R/W-0	R/W-0	R/W-0
CLKLOCK	IOLOCK ⁽²⁾	LOCK	—	CF	POSCEN	SOSCEN	OSWEN
bit 7	bit 0						

图注：

CO = 只可清零位

SO = 只可置 1 位

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现：读为 0

bit 14-12 **COSC<2:0>**：当前振荡器选择位

111 = 带后分频器的快速 RC 振荡器（FRCDIV）

110 = 保留

101 = 低功耗 RC 振荡器（LPRC）

100 = 辅助振荡器（SOSC）

011 = 带 PLL 模块的主振荡器（XTPLL、HSPLL 和 ECPLL）

010 = 主振荡器（XT、HS 和 EC）

001 = 带后分频器和 PLL 模块的快速 RC 振荡器（FRCPLL）

000 = 快速 RC 振荡器（FRC）

bit 11 未实现：读为 0

bit 10-8 **NOSC<2:0>**：新振荡器选择位⁽¹⁾

111 = 带后分频器的快速 RC 振荡器（FRCDIV）

110 = 保留

101 = 低功耗 RC 振荡器（LPRC）

100 = 辅助振荡器（SOSC）

011 = 带 PLL 模块的主振荡器（XTPLL、HSPLL 和 ECPLL）

010 = 主振荡器（XT、HS 和 EC）

001 = 带后分频器和 PLL 模块的快速 RC 振荡器（FRCPLL）

000 = 快速 RC 振荡器（FRC）

注 1：这些位的复位值由 FNOSC 配置位决定。

2： IOLOCK 位的状态只能在执行解锁序列后更改。此外，如果 IOL1WAY 配置位为 1，则 IOLOCK 位置 1 后它不能清零。

3： 在任意有效时钟切换期间，或者每当选择了非 PLL 时钟模式时，也复位为 0。

PIC24FJ256GB110 系列

寄存器 8-1: OSCCON: 振荡器控制寄存器 (续)

bit 7	CLKLOCK: 时钟选择锁定使能位 <u>如果使能 FSCM (FCKSM1 = 1):</u> 1 = 时钟和 PLL 选择被锁定 0 = 时钟和 PLL 选择未锁定, 可以通过将 OSWEN 位置 1 进行修改 <u>如果禁止 FSCM (FCKSM1 = 0):</u> 时钟和 PLL 选择始终未锁定, 可以通过将 OSWEN 位置 1 进行修改。
bit 6	IOLOCK: I/O 锁定使能位 ⁽²⁾ 1 = I/O 锁定工作 0 = I/O 锁定不工作
bit 5	LOCK: PLL 锁定状态位 ⁽³⁾ 1 = PLL 模块处于锁定状态, 或者满足 PLL 模块起振定时器 0 = PLL 模块脱离锁定状态, PLL 起振定时器正在运行或 PLL 被禁止
bit 4	未实现: 读为 0
bit 3	CF: 时钟故障检测位 1 = FSCM 检测到时钟故障 0 = 未检测到时钟故障
bit 2	POSCEN: 主振荡器休眠使能位 1 = 主振荡器在休眠模式下继续工作 0 = 主振荡器在休眠模式下禁止
bit 1	SOSCEN: 32 kHz 辅助振荡器 (SOSC) 使能位 1 = 使能辅助振荡器 0 = 禁止辅助振荡器
bit 0	OSWEN: 振荡器切换使能位 1 = 启动振荡器切换, 切换为 NOSC<2:0> 位指定的时钟源 0 = 振荡器切换完成

- 注 1: 这些位的复位值由 FNOSC 配置位决定。
2: IOLOCK 位的状态只能在执行解锁序列后更改。此外, 如果 IOL1WAY 配置位为 1, 则 IOLOCK 位置 1 后它不能清零。
3: 在任意有效时钟切换期间, 或者每当选择了非 PLL 时钟模式时, 也复位为 0。

寄存器 8-2: CLKDIV: 时钟分频器寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
ROI	DOZE2	DOZE1	DOZE0	DOZEN ⁽¹⁾	RCDIV2	RCDIV1	RCDIV0
bit 15	bit 8						

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
CPDIV1	CPDIV0	—	—	—	—	—	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **ROI:** 中断恢复位
 1 = 发生中断时清零 DOZEN 位, 并将 CPU 外设时钟比复位为 1:1
 0 = 中断对 DOZEN 位无影响

bit 14-12 **DOZE<2:0>:** CPU 外设时钟比选择位
 111 = 1:128
 110 = 1:64
 101 = 1:32
 100 = 1:16
 011 = 1:8
 010 = 1:4
 001 = 1:2
 000 = 1:1

bit 11 **DOZEN:** 打盹使能位⁽¹⁾
 1 = DOZE<2:0> 位指定 CPU 外设时钟比
 0 = CPU 外设时钟比设置为 1:1

bit 10-8 **RCDIV<2:0>:** FRC 后分频比选择位
 111 = 31.25 kHz (256 分频)
 110 = 125 kHz (64 分频)
 101 = 250 kHz (32 分频)
 100 = 500 kHz (16 分频)
 011 = 1 MHz (8 分频)
 010 = 2 MHz (4 分频)
 001 = 4 MHz (2 分频)
 000 = 8 MHz (1 分频)

bit 7-6 **CPDIV<1:0>:** USB 系统时钟选择位 (通过将 32 MHz 时钟进行后分频)
 11 = 4 MHz (8 分频) ⁽²⁾
 10 = 8 MHz (4 分频) ⁽²⁾
 01 = 16 MHz (2 分频)
 00 = 32 MHz (1 分频)

bit 5-0 未实现: 读为 0

注 1: 该位在 ROI 位置 1 和发生中断时自动清零。

2: 当 USB 模块使能时, 不允许该设置。

PIC24FJ256GB110 系列

寄存器 8-3: OSCTUN: FRC 振荡器调节寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN5 ⁽¹⁾	TUN4 ⁽¹⁾	TUN3 ⁽¹⁾	TUN2 ⁽¹⁾	TUN1 ⁽¹⁾	TUN0 ⁽¹⁾
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-6 未实现: 读为 0

bit 5-0 TUN<5:0>: FRC 振荡器调节位⁽¹⁾

011111 = 最大频率偏移

011110 =

•

•

•

000001 =

000000 = 中心频率, 振荡器以出厂校准频率工作

111111 =

•

•

•

100001 =

100000 = 最小频率偏移

注 1: TUN<5:0> 的递增和递减不能在 FRC 调节范围内同步更改 FRC 频率, 并且频率的变化也可能不是单调的。

8.4 时钟切换操作

在软件控制下, 应用程序可以随时在 4 种时钟源 (POSC、SOSC、FRC 和 LPRC) 之间自由切换, 几乎没有什么限制。为了限制这种灵活性可能带来的副作用, PIC24F 器件在切换过程中采用了保护锁定。

注: 主振荡器模式具有三种不同的子模式 (XT、HS 和 EC), 它们由 POSCMDx 配置位决定。虽然在软件中, 应用程序可以从其他模式切换为主振荡器模式并从主振荡器模式切换为其他模式, 但它不能在不重新编程器件的情况下在不同的主振荡器子模式之间切换。

8.4.1 使能时钟切换

要使能时钟切换, CW2 中的 FCKSM1 配置位必须编程为 0。(更多详细信息, 请参见第 26.1 节“配置位”。)如果 FCKSM1 配置位未编程 (1), 则时钟切换功能和故障保护时钟监视器功能被禁止。这是默认设置。

当时钟切换被禁止时, NOSCx 控制位 (OSCCON<10:8>) 不控制时钟选择。但是, COSCx 位 (OSCCON<14:12>) 将反映由 FNOSCx 配置位选择的时钟源。

在时钟切换被禁止时, OSWEN 控制位 (OSCCON<0>) 不起作用。它始终保持为 0。

8.4.2 振荡器切换过程

执行时钟切换需要以下基本过程：

1. 根据需要读 COSCx 位 (OSCCON<14:12>) 以确定当前的振荡器源。
2. 执行解锁序列以允许写入 OSCCON 寄存器的高字节。
3. 向 NOSCx 位 (OSCCON<10:8>) 写入新振荡器源的对应值。
4. 执行解锁序列以允许写入 OSCCON 寄存器的低字节。
5. 将 OSWEN 位置 1 以启动振荡器切换。

一旦基本过程完成，系统时钟硬件将自动进行如下响应：

1. 时钟切换硬件将 NOSCx 位的新值与 COSCx 位进行比较。如果它们相同，则时钟切换是冗余操作。在这种情况下，OSWEN 位自动清零，时钟切换中止。
2. 如果启动了有效的时钟切换，则 LOCK (OSCCON<5>) 和 CF (OSCCON<3>) 位清零。
3. 如果新的振荡器当前不在运行，则硬件会将其启动。如果必须要启动晶振，则硬件会等待到振荡器起振定时器 (OST) 计时期满。如果新的振荡器源需要使用 PLL，则硬件将等待到检测到 PLL 锁定 (LOCK = 1)。
4. 硬件将等待新的时钟源 10 个时钟周期，然后执行时钟切换。
5. 硬件清零 OSWEN 位表示时钟转换成功。此外，NOSCx 位的值被传送到 COSCx 位。
6. 此时，原来的时钟源被关闭，LPRC (如果 WDT 或 FSCM 使能) 或 SOSC (如果 SOSCEN 保持置 1) 除外。

注 1：在整个时钟切换过程中，处理器将继续执行代码。在此期间，不应执行对时序要求高的代码。

2: 不允许在任何带 PLL 的主振荡器模式和 FRCPLL 模式之间进行直接时钟切换。这适用于任何方向下的时钟切换。在这些情况下，应用必须首先切换到 FRC 模式将其作为两个 PLL 模式之间的过渡时钟源。

时钟切换的建议代码序列包括以下：

1. 禁止在 OSCCON 寄存器解锁和写序列期间的中断。
2. 在两个紧接的指令中将 78h 和 9Ah 写入 OSCCON<15:8>，以执行 OSCCON 高字节的解锁序列。
3. 在紧接解锁序列之后的指令中将新的振荡器源写入 NOSCx 位。
4. 在两个紧接的指令中将 46h 和 57h 写入 OSCCON<7:0>，以执行 OSCCON 低字节的解锁序列。
5. 在紧接解锁序列之后的指令中将 OSWEN 位置 1。
6. 继续执行对时钟时序要求不高的代码（可选）。
7. 调用相应数量的软件延时（周期计数），以使选中的振荡器和 / 或 PLL 启动并稳定。
8. 检查 OSWEN 是否为 0。如果为 0，则说明切换成功。如果 OSWEN 仍然置 1，则检查 LOCK 位以确定故障的原因。

例 8-1 中显示了解锁 OSCCON 寄存器和启动时钟切换的核心序列。

例 8-1：时钟切换的基本代码序列

```
;Place the new oscillator selection in w0
;OSCCONH (high byte) Unlock Sequence
MOV      #OSCCONH, w1
MOV      #0x78, w2
MOV      #0x9A, w3
MOV.b   w2, [w1]
MOV.b   w3, [w1]
;Set new oscillator selection
MOV.b   WREG, OSCCONH
;OSCCONL (low byte) unlock sequence
MOV      #OSCCONL, w1
MOV      #0x46, w2
MOV      #0x57, w3
MOV.b   w2, [w1]
MOV.b   w3, [w1]
;Start oscillator switch operation
BSET    OSCCON, #0
```

PIC24FJ256GB110 系列

8.5 振荡器模式和 USB 操作

由于 USB 特定的时序要求，使能 USB 模块时，必须始终使用 48 MHz 的内部时钟。由于它远远超出了 CPU 最高时钟速度，我们提供了一种方法从单一振荡器源内部生成 USB 和系统时钟。PIC24FJ256GB110 系列器件使用和其他 PIC24FJ 器件相同的时钟结构，但包含一个有两个分支的 PLL 系统，用于生成两个时钟信号。

图 8-2 给出了 USB PLL 框图。在本系统中，来自主振荡器的输入由 PLL 预分频器分频，生成 4 MHz 输出。它用于驱动片上 96 MHz PLL 倍频器，从而驱动两路时钟。一路使用固定的 2 分频分频器生成 48 MHz USB 时钟。另一路使用固定的 3 分频分频器和可配置的 PLL 预分频器 / 分频器生成一系列系统时钟频率。CPDIV 位选择系统时钟速度；可用的时钟选项列于表 8-2 中。

USB PLL 预分频器不会自动检测输入的振荡器频率。用户必须手动配置 PLL 分频器，用 PLLDIV<2:0> 配置位生成所需的 4 MHz 输出。这样就把主振荡器频率的选择限制为 8 种，如表 8-3 所示。

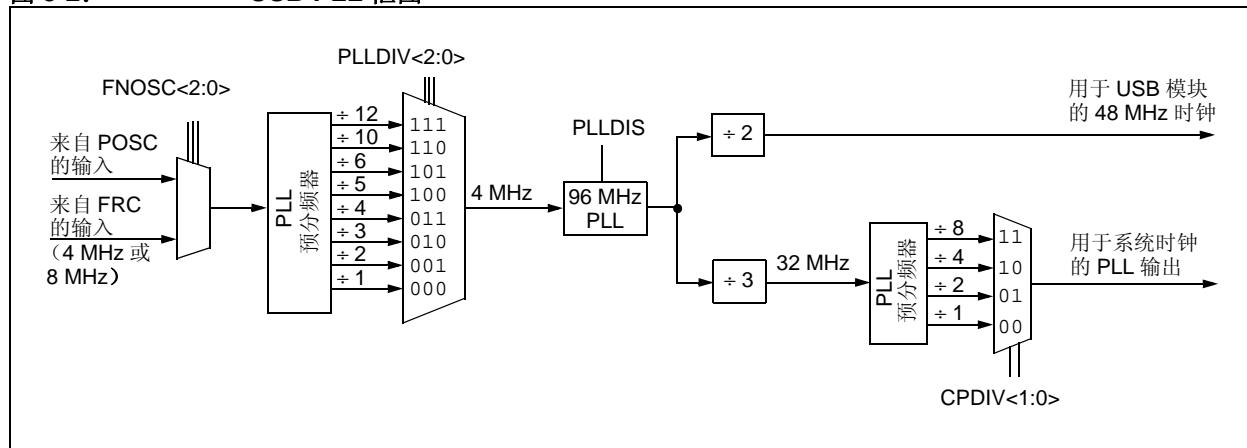
表 8-2: USB 操作期间的系统时钟选项

MCU 时钟分频 (CPDIV<1:0>)	单片机 时钟频率
无 (00)	32 MHz
+2 (01)	16 MHz
+4 (10)	8 MHz
+8 (11)	4 MHz

表 8-3: USB 操作的有效主振荡器配置

输入振荡器 频率	时钟模式	PLL 分频 (PLLDIV<2:0>)
48 MHz	ECPLL	+12 (111)
40 MHz	ECPLL	+10 (110)
24 MHz	HSPLL 和 ECPLL	+6 (101)
20 MHz	HSPLL 和 ECPLL	+5 (100)
16 MHz	HSPLL 和 ECPLL	+4 (011)
12 MHz	HSPLL 和 ECPLL	+3 (010)
8 MHz	ECPLL 和 XTPLL	+2 (001)
4 MHz	ECPLL 和 XTPLL	+1 (000)

图 8-2: USB PLL 框图



8.5.1 USB 操作的注意事项

使用PIC24FJ256GB110系列器件中的USB On-The-Go模块块时，用户在配置系统时钟时必须始终遵守以下规则：

- 对于 USB 操作，所选时钟源（EC、HS 或 XT）必须满足 USB 时钟容差规范。
- 主振荡器 /PLL 模式是允许 USB 操作的唯一振荡器配置。不提供 USB 模块的独立外部时钟源。
- 这些器件中 FRCPLL 振荡器模式可用时，绝不要用于 USB 应用。应用程序未使用 USB 模块时，FRCPLL 模式仍可用。但是，用户必须始终确保 FRC 源配置为提供 4 MHz 或 8 MHz 的时钟（ $RCDIV<2:0> = 001$ 或 000 ），并且已恰当地配置了 USB PLL 预分频器。
- 所有其他振荡器模式也是可用的；但是 USB 操作在选择那些模式时不可用。需要其他工作功耗级别并且不需要 USB 模块时，它们仍可能有用（例如，应用处于休眠状态和等待总线连接）。

8.6 参考时钟输出

除了某些振荡器模式中可用的 CLKO 输出（ $Fosc/2$ ）外，PIC24FJ256GB110 系列器件中的器件时钟也可以配置为向端口引脚提供参考时钟输出信号。该功能在所有振荡器配置中都可用，允许用户选择更大范围的时钟分频因数来驱动应用中的外部器件。

该参考时钟输出受 REFOCON 寄存器（寄存器 8-4）控制。将 ROEN 位（REFOCON<15>）置 1 将使时钟信号在 REFO 引脚上可用。RODIV 位（REFOCON<11:8>）允许选择 16 个不同的时钟分频比选项。

ROSSLP 和 ROSEL 位（REFOCON<13:12>）控制休眠模式下参考输出的可用性。ROSEL 位决定用 OSC1 和 OSC2 上的振荡器还是当前系统时钟源作为参考时钟输出。ROSSLP 位决定器件处于休眠模式时 REFO 上的参考源是否可用。

要在休眠模式下使用参考时钟输出，ROSSLP 和 ROSEL 位都必须置 1。器件时钟也必须配置为主模式之一（EC、HS 或 XT）；否则，如果 POSCEN 位没有同时置 1，OSC1 和 OSC2 上的振荡器将在器件进入休眠模式后掉电。清零 ROSEL 位允许参考输出频率在任何时钟切换期间随着系统时钟的改变而改变。

PIC24FJ256GB110 系列

寄存器 8-4: REFOCON: 参考振荡器控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROEN	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0
bit 15	bit 8						

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **ROEN:** 参考振荡器输出使能位
1 = 在 REFO 引脚上使能参考振荡器
0 = 禁止参考振荡器
- bit 14 未实现: 读为 0
- bit 13 **ROSSLP:** 参考振荡器输出在休眠模式下停止的位
1 = 参考振荡器在休眠模式下继续运行
0 = 参考振荡器在休眠模式下禁止
- bit 12 **ROSEL:** 参考振荡器源选择位
1 = 主振荡器用作基本时钟。请注意, 晶振必须用 FOSC<2:0> 位使能; 晶振在休眠模式下仍保持工作。
0 = 用作基本时钟的系统时钟; 基本时钟反映器件的任何时钟切换
- bit 11-8 **RODIV<3:0>:** 参考振荡器分频比选择位
1111 = 基本时钟值 32,768 分频
1110 = 基本时钟值 16,384 分频
1101 = 基本时钟值 8,192 分频
1100 = 基本时钟值 4,096 分频
1011 = 基本时钟值 2,048 分频
1010 = 基本时钟值 1,024 分频
1001 = 基本时钟值 512 分频
1000 = 基本时钟值 256 分频
0111 = 基本时钟值 128 分频
0110 = 基本时钟值 64 分频
0101 = 基本时钟值 32 分频
0100 = 基本时钟值 16 分频
0011 = 基本时钟值 8 分频
0010 = 基本时钟值 4 分频
0001 = 基本时钟值 2 分频
0000 = 基本时钟值
- bit 7-0 未实现: 读为 0

9.0 节能特性

注: 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见《PIC24F 系列参考手册》的第 10 章“节能特性”(DS39698A_CN)。

PIC24FJ256GB110 系列器件提供了管理功耗的功能, 该功能是通过有选择地管理 CPU 和外设的时钟源来实现的。一般而言, 较低的时钟频率和减少时钟源驱动电路的数目会使功耗降低。所有 PIC24F 器件用以下四种方法管理功耗:

- 时钟频率
- 基于指令的休眠模式和空闲模式
- 软件控制的打盹模式
- 用软件有选择地进行外设控制

可以组合使用这些方法从而在保证关键应用特性(如对于时序敏感的通信)的情况下有选择地调节应用的功耗。

9.1 时钟频率和时钟切换

PIC24F 器件提供的时钟频率范围较大, 用户可根据应用需要进行选择。如果未锁定系统时钟配置, 用户只需更改 NOSC 位即可选择低功耗或高精度振荡器。在工作期间更改系统时钟的过程以及相应的限制, 将在第 8.0 节“振荡器配置”中进行更详细的讨论。

9.2 基于指令的节能模式

PIC24F 器件有两种特殊的节能模式, 通过执行特殊的 PWRSAV 指令可以进入这两种模式。休眠模式下时钟停止工作并停止所有代码执行; 空闲模式下 CPU 停止工作并停止代码执行, 但是允许外设模块继续工作。例 9-1 中所示为 PWRSAV 指令的汇编语法。

例 9-1: PWRSAV 指令语法

```
PWRSAV #SLEEP_MODE      ; Put the device into SLEEP mode  
PWRSAV #IDLE_MODE       ; Put the device into IDLE mode
```

在被允许的中断产生、WDT 超时或器件复位时, 器件会退出休眠和空闲模式。器件退出这两种模式的过程称为“唤醒”。

注: SLEEP_MODE 和 IDLE_MODE 是在所选器件的汇编头文件中定义的常量。

9.2.1 休眠模式

休眠模式具有下列特征:

- 系统时钟源关闭。如果使用了片上振荡器, 也要关闭它。
- 如果 I/O 引脚上不消耗电流, 则器件电流消耗将降至最低。
- 由于系统时钟源被禁止, 所以故障保护时钟监视器在休眠模式下不工作。
- 如果 WDT 被使能, 则 LPRC 时钟将继续在休眠模式下运行。
- 如果 WDT 被使能, 则在进入休眠模式之前自动清零。
- 有些器件功能或外设可能在休眠模式下继续工作, 包括 I/O 端口上的输入电平变化通知功能或使用外部时钟输入的外设等。任何需要使用系统时钟源来工作的外设在休眠模式下将被禁止。

当发生以下任何事件时, 器件将从休眠模式唤醒:

- 产生任何被单独允许的中断源
- 任何形式的器件复位
- WDT 超时

从休眠模式唤醒时, 处理器将使用在进入休眠模式时处于工作状态的时钟源重新开始工作。

9.2.2 空闲模式

空闲模式具有下列特征：

- CPU 将停止执行指令。
- WDT 被自动清零。
- 系统时钟源保持工作状态。默认情况下，所有外设模块将继续使用系统时钟源正常工作，也可以有选择地禁止它们（见第 9.4 节“选择性外设模块控制”）。
- 如果 WDT 或 FSCM 被使能，则 LPRC 也将保持工作状态。

当发生以下任何事件时，器件将从空闲模式唤醒：

- 产生任何被单独允许的中断。
- 任何器件复位。
- WDT 超时。

从空闲模式唤醒时，重新为 CPU 提供时钟，且立即从 PWRSAV 指令之后的下一条指令或 ISR 中的第一条指令开始执行指令。

9.2.3 在节能指令执行期间的中断

在 PWRSAV 指令执行期间发生的中断都将延迟到进入休眠或空闲模式后才产生，并导致器件从休眠或空闲模式中唤醒。

9.3 打盹模式

通常，更改时钟速度和进入某种节能模式是降低功耗的首选策略。然而，有些情况下不可行。例如，某些应用可能必须保持不间断的同步通信，即便在它不执行任何其他操作时也不例外。降低系统时钟速度可能会带来通信错误，而使用节能模式可能会完全终止通信。

打盹模式是另一种简单有效的节能方法，它可以在器件仍然执行代码的情况下降低功耗。在此模式下，系统时钟以相同的时钟源和相同的速度继续工作。外设模块时钟速度保持不变，但 CPU 时钟的速度降低了。保持这两个时钟域同步，可以保持外设访问 SFR 的能力，同时 CPU 以较慢的速度执行代码。

通过将 DOZEN 位 (CLKDIV<11>) 置 1 使能打盹模式。外设与内核的时钟速度之比是由 DOZE<2:0> 位 (CLKDIV<14:12>) 决定的。有八种可能的配置，从 1:1 到 1:256，其中 1:1 是默认设置。

在事件驱动的应用中，使用打盹模式有选择地降低功耗是可行的。这样就可以实现不间断地运行对时序要求高的功能（如同步通信），而 CPU 保持空闲等待事件调用中断服务程序。通过将 ROI 位 (CLKDIV<15>) 置 1，可以使器件在产生中断时自动返回到全速 CPU 工作模式。默认情况下，中断事件对打盹模式工作没有影响。

9.4 选择性外设模块控制

空闲和打盹模式使用户可通过降低或停止 CPU 时钟显著降低功耗。即便如此，外设模块仍然保持有时钟供应，因此会有功耗。有些应用可能有这些模式无法提供的需要：即将用电资源分配给 CPU 处理，而使外设功耗最低。

PIC24F 器件通过允许有选择地禁止外设模块以降低或消除其功耗来满足这种需求。这可以通过两个控制位来实现：

- 外设使能位，一般命名为“XXXEN”，位于模块的主控制 SFR 中。
- 外设模块禁止 (PMD) 位，一般命名为“XXXMD”，位于相应的 PMD 控制寄存器中。

这两个位在使能或禁止其相关模块方面具有相似的功能。将模块的 PMD 位置 1 会禁止该模块的所有时钟源，从而将其功耗降至绝对最低。在该状态下，与外设相关的控制寄存器和状态寄存器也被禁止，因此写入那些寄存器不会有影响，且读取值无效。许多外设模块都有一个对应的 PMD 位。

相反，通过清零模块的 XXXEN 位以禁止其功能来禁止该模块，会保留其寄存器的读写功能。这也会降低功耗，但不如将 PMD 位置 1 降低得多。多数外设模块都有一个使能位；输入捕捉、输出比较和 RTCC 则例外。

要实现更多可选择的节能，可在器件进入空闲模式后有选择地禁止外设模块。这可通过通用名称格式为“XXXIDL”的控制位实现。默认情况下，所有能在空闲模式下工作的模块都适用。使用“空闲模式禁止”功能，使功耗在空闲模式下得到进一步降低，从而增强了对功耗要求异常严格的应用的节能功能。

10.0 I/O 端口

注：本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》的第 12 章“带外设引脚选择 (PPS) 的 I/O 端口”(DS39711A_CN)。

所有器件引脚 (VDD、Vss、MCLR 和 OSC1/CLK1 除外) 均由外设和并行 I/O 端口所共用。所有 I/O 输入端口都为施密特触发器输入，提高了抗噪声能力。

10.1 并行 I/O (PIO) 端口

通常，与某个外设共用一个引脚的并行 I/O 端口总是服务于该外设。外设的输出缓冲器数据和控制信号提供给一对多路开关。这对多路开关用于选择 I/O 引脚的输出数据和控制信号是用于外设还是相应的端口。该逻辑电路同时会阻止“环回进入”(loop through)，即一个端口的数字输出可以驱动共用同一个引脚的外设输入。

图 10-1 显示了端口是如何与其他外设复用的以及相关的 I/O 引脚。

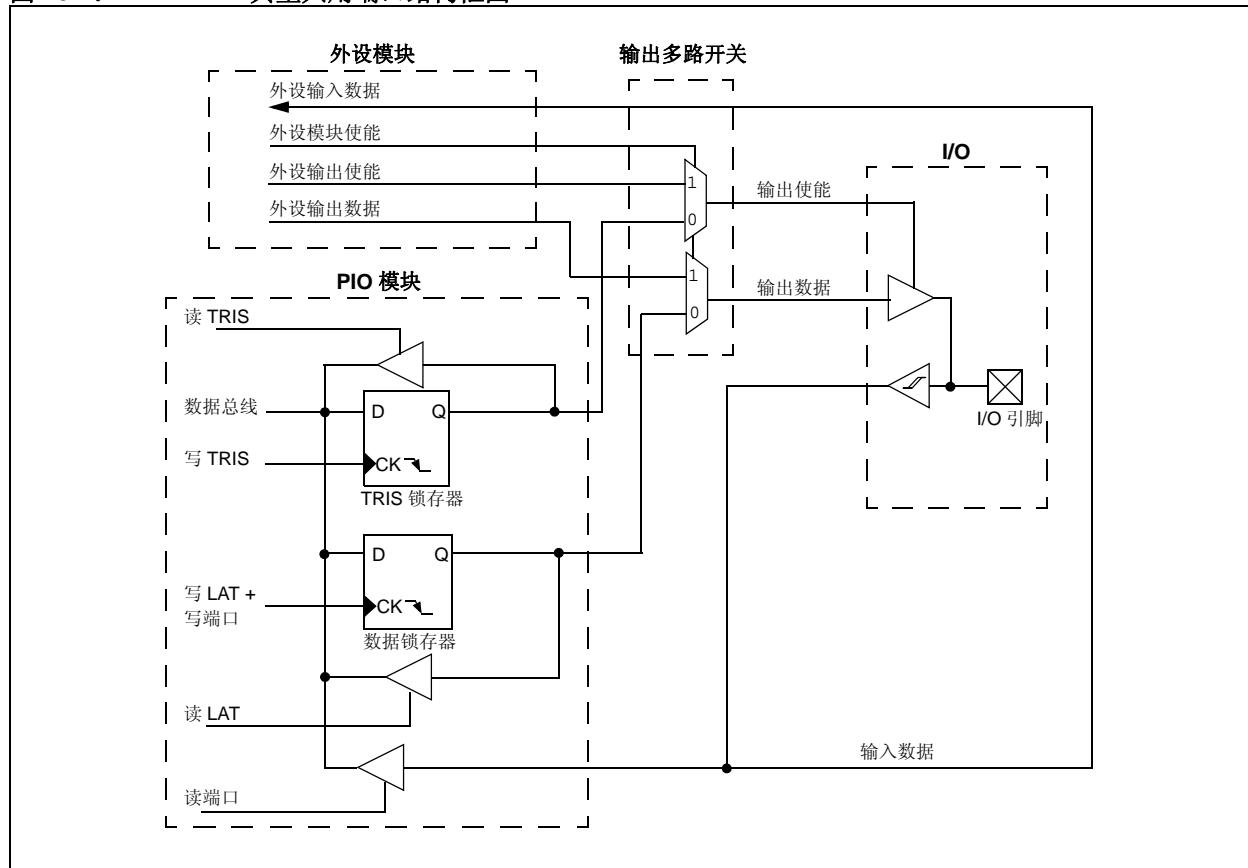
如果外设使能，并且正在驱动相关引脚时，该引脚将不再作为通用输出引脚。可以读该 I/O 引脚，但并行端口位的输出驱动器将被禁止。如果使能某外设但没有驱动引脚时，则该引脚可由端口驱动。

所有端口引脚都有 3 个寄存器与端口引脚作为数字 I/O 时的工作直接相关。数据方向寄存器 (TRISx) 决定引脚是输入还是输出。如果数据方向位为 1，则引脚是输入。复位后，所有端口引脚均定义为输入。读输出锁存寄存器 (LATx) 时，读到的是锁存器中的值；写锁存器时，写入的是锁存器。读端口 (PORTx) 时，读到的是端口引脚的值；而写端口引脚时，写入的是锁存器。

对于特定器件无效的任何位及其相关的数据和控制寄存器都将被禁止。这意味着对应的 LATx 和 TRISx 寄存器以及端口引脚都将读为零。

当与其他外设或功能共享的引脚被定义为仅能输入的引脚时，该引脚被认为是专用端口，因为没有与之竞争的输出源。

图 10-1：典型共用端口结构框图



10.1.1 漏极开路配置

除 PORT、LAT 和 TRIS 寄存器用于数据控制外，每个端口引脚也可被单独地配置为数字输出或漏极开路输出。这是由与每个端口相关的漏极开路控制寄存器 ODCx 控制的，将其中的任何位置 1 即可将相应的引脚配置为漏极开路输出。

这种漏极开路特性允许通过使用外部上拉电阻，在所需的任意仅用作数字功能的引脚上产生高于 VDD (如 5V) 的输出电平。允许的最大开漏电压与最大 VH 规范相同。

10.2 配置模拟端口引脚

AD1PCFGL 和 TRIS 寄存器用于控制 A/D 端口引脚的操作。如果要将端口引脚设置为模拟输入，则对应的 TRIS 位也必须置 1。如果将 TRIS 位清零（输出），则数字输出电平（VOH 或 VOL）将被转换。

当读取端口寄存器时，所有配置为模拟输入通道的引脚均读为零（低电平）。

配置为数字输入的引脚将不会对模拟输入进行转换。对于任何定义为数字输入的引脚（包括 ANx 引脚），加在引脚上的模拟电压可能导致输入缓冲器消耗的电流超出器件规范。

10.2.1 I/O 端口写 / 读时序

端口方向改变或端口写操作与同一端口的读操作之间需要一个指令周期。通常该指令是 NOP。

10.2.2 模拟输入引脚和电压注意事项

用作器件输入的引脚可承受的电压与该引脚的输入功能有关。只用作数字输入的引脚能够承受最高 5.5V 的直流电压，这是数字逻辑电路的典型电平值。而同时具有任何类型模拟输入功能的引脚只能承受最高 VDD 的电压。应避免使这些引脚上的电压超出 VDD。表 10-1 总结了输入电压能力。更多详细信息，请参见第 29.1 节“直流量特性”。

注： 为了便于识别，在数据手册开头处的引脚图中以黑灰色的阴影标识了能耐受 5.5V 的引脚。

表 10-1： 输入电压⁽¹⁾

端口或引脚	能耐受的输入电压	说明
PORTE<10:9>	VDD	只能耐受 VDD 输入电压的引脚
PORTE<15:0>		
PORTE<15:12>		
PORTE<7:6>		
PORTE<0>		
PORTE<9:6>, PORTE<3:2>		
PORTE<15:14>, PORTE<7:0>	5.5V	能耐受高于 VDD 的输入电压，对于大多数标准逻辑有用。
PORTE<4:1>		
PORTE<15:8>, PORTE<5:0>		
PORTE<9:0>		
PORTE<13:12>, PORTE<8>, PORTE<5:1>		
PORTE<15:12>, PORTE<1:0>		

注 1： 并非所有在表中列出的端口引脚都可以在 64 引脚和 80 引脚器件上实现。请参见第 1.0 节“器件概述”确认特定器件具有哪些端口。

例 10-1： 端口写 / 读示例

```
MOV 0xFF00, W0          ; Configure PORTB<15:8> as inputs
MOV W0, TRISBB          ; and PORTB<7:0> as outputs
NOP                   ; Delay 1 cycle
BTSS  PORTB, #13; Next Instruction
```

10.3 输入状态变化通知

I/O 端口的输入状态变化通知功能允许 PIC24FJ256GB110 系列器件在选定输入引脚的状态变化 (Change-of-State, COS) 时，向处理器发出中断请求。当禁止时钟时，该特性还可在休眠模式下检测到输入状态变化。根据器件的引脚数，最多可以选择（允许）81 个外部输入在状态发生变化时产生中断请求。

寄存器 CNEN1 到 CNEN6 包含每个 CN 输入引脚的中断允许控制位。将其中任一位置 1 将允许相应引脚的 CN 中断。

每个 CN 引脚都有与之相连的一个弱上拉电路和一个弱下拉电路。上拉电路充当连接到该引脚的电流源，而下拉电路则充当连接到该引脚的灌电流。当连接了按钮或键盘设备时，不再需要使用外部电阻。使用 CNPU1 到 CNPU6 寄存器（用于上拉）以及 CNPD1 到 CNPD6 寄存器（用于下拉）可分别使能各个上拉电路和下拉电路。每个 CN 引脚都有独立的上拉和下拉控制位。将控制位置 1 可使能相应引脚的弱上拉或下拉功能。

当选择内部上拉时，引脚上拉至 VDD - 0.7V（典型值）。请确保在使能内部上拉时没有任何外部上拉源，因为这样会导致电压差异，从而形成电流通路。

注： 当端口引脚被配置为数字输出时，状态变化通知引脚的上拉电路将始终被禁止。

10.4 外设引脚选择

通用器件的主要挑战是提供尽可能多的外设功能部件，同时将其与 I/O 引脚功能的冲突降到最小。在必须使用一个引脚上复用的多个外设的应用中，大量修改应用程序代码或完全地重新设计可能是唯一的选择。

外设引脚选择 (PPS) 功能提供了这些选择的替代方法，即在较宽的 I/O 引脚范围内使能用户的外设功能部件选择及其配置。通过增加特定器件上可用的引脚排列选项，用户可以让单片机更好地适合他们的整个应用，而不是通过修改应用来适应器件。

外设引脚选择功能在数字 I/O 引脚的固定子集下工作。用户可以将许多数字外设的输入和 / 或输出独立地映射到这些 I/O 引脚中的任何一个。外设引脚选择通过软件来执行，通常不需要对器件进行再编程。一旦建立外设引脚选择，就同时包含了硬件保护，以防止对外设映射的意外或错误更改。

10.4.1 可用的引脚

外设引脚选择功能可在最多 44 个引脚的范围内使用，取决于特定器件及其引脚数。支持外设引脚选择功能的引脚在它们的引脚全称中包含名称 “R_n” 或 “R_{Pln}”，其中 “n” 是可重映射的引脚编号。“RP” 用于表示支持可重映射输入和输出功能的引脚，而 “RPI” 表示仅支持可重映射输入功能的引脚。

PIC24FJ256GB110 系列器件支持的可重映射仅输入引脚的数目远大于可重映射输入 / 输出引脚的数目。在本器件系列中，最多有 32 个可重映射输入 / 输出引脚，取决于所选特定器件的引脚数；编号从 RP0 到 RP31。可重映射仅输入引脚的编号在该范围之上，从 RPI32 到 RPI43（即该特定器件可重映射引脚的上限）。

请参见表 1-4 了解每种封装中引脚排列选项的汇总。

10.4.2 可用的外设

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

外设引脚选择不适用于 I^2C ™电平变化通知输入、RTCC 闹钟输出或带模拟输入的外设。

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

10.4.2.1 外设引脚选择功能优先级

可选引脚的外设的输出（例如 OC 和 UART 发送）的优先级高于与该引脚连接的所有通用数字功能（例如 PMP 和端口 I/O）。特殊的数字输出（例如 USB 功能）的优先级高于同一引脚上的 PPS 输出。在本数据手册开头部分的引脚框图按照优先级顺序列出了外设输出。请参见这些引脚框图以了解关于某个特定引脚的优先级信息。

与具有固定外设的 PIC24F 器件不同，可选引脚的外设输入从不具有该引脚的所有权。引脚的输出缓冲器由该引脚的 TRISx 位设置控制，或由该引脚上的固定外设控制。如果引脚配置为数字模式，则 PPS 输入正常工作。如果使能了同一引脚上的模拟功能，则禁止 PPS 输入。

10.4.3 控制外设引脚选择

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

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

10.4.3.1 输入映射

外设引脚选择选项的输入根据外设进行映射；即，与外设相关的控制寄存器指示要被映射的引脚。RPINRx 寄存器用来配置外设输入映射（见寄存器 10-1 到寄存器 10-21）。每个寄存器包含两组 6 位位域，每组都与引脚可选择外设之一相关。用适当的 6 位值编程给定外设的位域，会将具有该值的 RPn 引脚映射到该外设。对于任何给定的器件，任何位域的值的有效范围与器件所支持的外设引脚选择的最大数目相对应。

10.4.3.2 输出映射

与输入相比，外设引脚选择选项的输出根据引脚进行映射。在这种情况下，与特定引脚相关的控制寄存器指示要被映射的外设输出。RPORx 寄存器用来控制输出映射。每个寄存器包含两组 6 位位域，每组都与一个 RPn 引脚相关（见寄存器 10-22 到寄存器 10-37）。位域的值与外设之一相对应，并且该外设的输出被映射到引脚（见表 10-3）。

由于映射技术，输出映射的外设列表也包含 000000 的空值。这允许任何给定的引脚保持与任何引脚可选择外设的输出之间的未连接状态。

表 10-2: 可选择的输入源 (将输入映射到功能) (1)

输入名称	功能名称	寄存器	功能映射位
外部中断 1	INT1	RPINR0	INT1R<5:0>
外部中断 2	INT2	RPINR1	INT2R<5:0>
外部中断 3	INT3	RPINR1	INT3R<5:0>
外部中断 4	INT4	RPINR2	INT4R<5:0>
输入捕捉 1	IC1	RPINR7	IC1R<5:0>
输入捕捉 2	IC2	RPINR7	IC2R<5:0>
输入捕捉 3	IC3	RPINR8	IC3R<5:0>
输入捕捉 4	IC4	RPINR8	IC4R<5:0>
输入捕捉 5	IC5	RPINR9	IC5R<5:0>
输入捕捉 6	IC6	RPINR9	IC6R<5:0>
输入捕捉 7	IC7	RPINR10	IC7R<5:0>
输入捕捉 8	IC8	RPINR10	IC8R<5:0>
输入捕捉 9	IC9	RPINR15	IC9R<5:0>
输出比较故障 A	OCFA	RPINR11	OCFAR<5:0>
输出比较故障 B	OCFB	RPINR11	OCFBR<5:0>
SPI1 时钟输入	SCK1IN	RPINR20	SCK1R<5:0>
SPI1 数据输入	SDI1	RPINR20	SDI1R<5:0>
SPI1 从选择输入	SS1IN	RPINR21	SS1R<5:0>
SPI2 时钟输入	SCK2IN	RPINR22	SCK2R<5:0>
SPI2 数据输入	SDI2	RPINR22	SDI2R<5:0>
SPI2 从选择输入	SS2IN	RPINR23	SS2R<5:0>
SPI3 时钟输入	SCK3IN	RPINR23	SCK3R<5:0>
SPI3 数据输入	SDI3	RPINR28	SDI3R<5:0>
SPI3 从选择输入	SS3IN	RPINR29	SS3R<5:0>
Timer2 外部时钟	T2CK	RPINR3	T2CKR<5:0>
Timer3 外部时钟	T3CK	RPINR3	T3CKR<5:0>
Timer4 外部时钟	T4CK	RPINR4	T4CKR<5:0>
Timer5 外部时钟	T5CK	RPINR4	T5CKR<5:0>
UART1 允许发送	<u>U1CTS</u>	RPINR18	U1CTSR<5:0>
UART1 接收	U1RX	RPINR18	U1RXR<5:0>
UART2 允许发送	<u>U2CTS</u>	RPINR19	U2CTSR<5:0>
UART2 接收	U2RX	RPINR19	U2RXR<5:0>
UART3 允许发送	<u>U3CTS</u>	RPINR21	U3CTSR<5:0>
UART3 接收	U3RX	RPINR17	U3RXR<5:0>
UART4 允许发送	<u>U4CTS</u>	RPINR27	U4CTSR<5:0>
UART4 接收	U4RX	RPINR27	U4RXR<5:0>

注 1: 除非另外声明, 否则所有输入都使用施密特触发器输入缓冲器。

PIC24FJ256GB110 系列

表 10-3: 可选择的输出源 (将功能映射到输出)

输出功能编号 ⁽¹⁾	功能	输出名称
0	NULL ⁽²⁾	空
1	C1OUT	比较器 1 的输出
2	C2OUT	比较器 2 的输出
3	U1TX	UART1 发送
4	U1RTS ⁽³⁾	UART1 请求发送
5	U2TX	UART2 发送
6	U2RTS ⁽³⁾	UART2 请求发送
7	SDO1	SPI1 数据输出
8	SCK1OUT	SPI1 时钟输出
9	SS1OUT	SPI1 从选择输出
10	SDO2	SPI2 数据输出
11	SCK2OUT	SPI2 时钟输出
12	SS2OUT	SPI2 从选择输出
18	OC1	输出比较 1
19	OC2	输出比较 2
20	OC3	输出比较 3
21	OC4	输出比较 4
22	OC5	输出比较 5
23	OC6	输出比较 6
24	OC7	输出比较 7
25	OC8	输出比较 8
28	U3TX	UART3 发送
29	U3RTS ⁽³⁾	UART3 请求发送
30	U4TX	UART4 发送
31	U4RTS ⁽³⁾	UART4 请求发送
32	SDO3	SPI3 数据输出
33	SCK3OUT	SPI3 时钟输出
34	SS3OUT	SPI3 从选择输出
35	OC9	输出比较 9
36	C3OUT	比较器 3 输出
37-63	(未用)	NC

注 1: 用所列的值设置 RPORx 寄存器会将输出功能分配给相关的 RPn 引脚。

2: NULL 功能在器件复位时被分配给所有 RPn 输出，并禁止 RPn 输出功能。

3: IrDA® BCLK 功能使用该输出。

10.4.3.3 映射限制

外设引脚选择的控制机制相当灵活。除了防止两个物理引脚配置为同一功能输入，或两个功能输出配置为同一物理引脚导致的信号争用的系统模块，没有硬件强制的锁定。这种灵活性达到以下程度：允许一个输入驱动多个外设，或一个功能输出驱动多个输出引脚。

10.4.3.4 PIC24FJ256GB110 系列器件的映射例外

尽管 PPS 寄存器理论上允许最多 64 个可重映射的 I/O 引脚，但并非在所有器件上都全部实现。对于 PIC24FJ256GB110 系列器件，可用的最大可重映射引脚数是 44，包括 12 个仅输入引脚。此外，RP 和 RPI 序列中的部分引脚在低引脚数器件中未实现。可用的可重映射引脚数的差别归纳在表 10-4 中。

开发使用可重映射引脚的应用时，用户应时刻牢记：

- 对于 RPINRx 寄存器，对应于特定器件的未实现引脚的位组合被视为无效；对应的模块无输入映射到它。对于所有 PIC24FJ256GB110 系列器件，这包括所有大于 43 (101011) 的值。
- 对于 RPORx 寄存器，对应于未实现引脚的位域也未实现。写入这些位域不会有影响。

10.4.4 控制配置更改

由于可以在运行时更改外设的重映射，因此必须对外设重映射设置一些限制条件以防止意外更改配置。PIC24F 器件包含 3 个功能以防对外设映射的更改：

- 控制寄存器锁定序列
- 连续状态监视
- 配置位重映射锁定

10.4.4.1 控制寄存器锁定

在正常操作下，不允许写入 RPINRx 和 RPORx 寄存器。尝试写入操作看似正常执行，但实际上寄存器的内容保持不变。要更改这些寄存器，必须用硬件进行解锁。寄存器锁定由 IOLOCK 位 (OSCCON<6>) 控制。将 IOLOCK 置 1 可防止对控制寄存器的写操作；将 IOLOCK 清零则允许写操作。

要置 1 或清零 IOLOCK，必须执行特定的命令序列：

1. 将 46h 写入 OSCCON<7:0>。
2. 将 57h 写入 OSCCON<7:0>。
3. 通过一次操作清零（或置 1）IOLOCK。

与振荡器的 LOCK 位的序列不同，IOLOCK 会保持一种状态直到被更改。这允许所有的外设引脚选择均可被配置为：执行一条解锁序列，紧接着对所有控制寄存器进行更新，然后用第二个锁定序列锁定。

10.4.4.2 连续状态监视

除了防止直接写操作，RPINRx 和 RPORx 寄存器的内容一直由影子寄存器通过硬件进行监视。如果任何寄存器发生了意外更改（例如 ESD 或其他外部事件引起的存储单元干扰），将会触发配置不匹配复位。

10.4.4.3 配置位引脚选择锁定

为了进一步确保安全，可以将器件配置为防止对 RPINRx 和 RPORx 寄存器进行多于一个写会话。IOL1WAY (CW2<4>) 配置位会阻止 IOLOCK 位在置 1 后被清零。如果 IOLOCK 保持置 1，则不会执行寄存器解锁过程，且不能写入外设引脚选择控制寄存器。清零该位并重新使能外设映射的唯一方法是执行器件复位。

在默认（未编程）状态下，IOL1WAY 被置 1，限制用户只能进行一次写会话。对 IOL1WAY 编程可允许用户对外设引脚选择寄存器不受限制的访问（通过正确地执行解锁序列）。

表 10-4： PIC24FJ256GB110 系列器件的可重映射引脚例外

器件引脚数	RP 引脚数 (I/O)		RPI 引脚数	
	总共	未实现	总共	未实现
64 引脚	28	RP5, RP15, RP30, RP31	1	RPI32-36, RPI38-43
80 引脚	31	RP31	9	RPI32, RPI39, RPI41
100 引脚	32	—	12	—

10.4.5 外设引脚选择的注意事项

在应用设计中使用控制外设引脚选择功能有一些可能被忽略的注意事项。对于几个只能作为可重映射外设的常见外设尤其如此。

主要的注意事项是在器件的默认（复位）状态下，外设引脚选择在默认引脚上不可用。由于所有 RPINRx 寄存器复位为 111111，所有 RPORx 寄存器复位为 000000，因此所有外设引脚选择输入连接到 Vss，而所有外设引脚选择输出处于未连接状态。

注： 在将外设引脚选择输入连接到 RP63 时，RP63 不必存在于器件之上，因为寄存器复位总是默认为这样的设置。

这种情况要求在执行任何其他应用程序代码前，用户必须用适当的外设配置初始化器件。由于 IOLOCK 位复位到解锁状态，因此在器件复位结束后不必执行解锁序列。然而，基于应用安全考虑，在写入控制寄存器后最好将 IOLOCK 置 1 并锁定配置。

由于解锁序列对时序要求很高，它必须作为汇编语言程序以与更改振荡器配置相同的方式执行。如果应用程序是用 C 语言或其他高级语言编写的，则解锁序列应通过写行内汇编代码来执行。

选择配置需要查看所有外设引脚选择及其引脚分配，尤其是那些不会在应用中使用的外设。在所有情况下，必须完全禁止未用的引脚可选择外设。未用的外设应将它们的输入分配给未用的 RPin 引脚功能。带有未用 RPin 功能的 I/O 引脚应被配置为空外设输出。

外设到特定引脚的分配不会自动执行引脚的 I/O 电路的任何其他配置。理论上，这意味着将引脚可选择输出加到引脚，当驱动输出时，引脚可能会意外驱动现有的外设输入。用户必须熟悉共用同一个可重映射引脚的其他固定外设的行为，了解何时使能或禁止它们。为安全起见，共用同一个引脚的固定数字外设在不使用时应被禁止。

根据这些概念，配置特定外设的可重映射引脚不会自动开启该功能。必须将外设特别配置为工作并使能，好像是连接到固定引脚一样。这部分在应用程序代码中的位置（紧跟器件复位和外设配置，或在主应用程序内）取决于外设及其在应用中的使用。

最后的注意事项是外设引脚选择功能既不会改写模拟输入，也不会将带模拟功能的引脚重新配置为数字 I/O。如果器件复位时引脚被配置为模拟输入，则使用外设引脚选择时，必须明确将其重新配置为数字 I/O。

例 10-2 给出了使用 UART1 为具有流控制的双向通信所作的配置。使用了以下输入和输出功能：

- 输入功能：U1RX 和 U1CTS
- 输出功能：U1TX 和 U1RTS

例 10-2：配置 UART1 输入和输出功能

```
// Unlock Registers
__builtin_write_OSCCONL(OSCCON & 0xBF);

// Configure Input Functions (Table 9-1)
// Assign U1RX To Pin RP0
RPINR18bits.U1RXR = 0;

// Assign U1CTS To Pin RP1
RPINR18bits.U1CTSR = 1;

// Configure Output Functions (Table 9-2)
// Assign U1TX To Pin RP2
RPOR1bits.RP2R = 3;

// Assign U1RTS To Pin RP3
RPOR1bits.RP3R = 4;

// Lock Registers
__builtin_write_OSCCONL(OSCCON | 0x40);
```

10.4.6 外设引脚选择寄存器

PIC24FJ256GB110 系列器件共实现了 37 个寄存器用于可重映射的外设配置：

- 输入可重映射的外设寄存器 (21)
- 输出可重映射的外设寄存器 (16)

注：仅在 IOLOCK (OSCCON<6>) = 0 时可以改变输入和输出寄存器的值。请参见第 10.4.4.1 节“控制寄存器锁定”了解特定命令序列。

寄存器 10-1： RPINR0：外设引脚选择输入寄存器 0

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0
bit 15							
bit 8							

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							
bit 0							

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现：读为 0

bit 13-8 **INT1R<5:0>**：将外部中断 1 (INT1) 分配给对应 RPin 或 RPIn 引脚的位

bit 7-0 未实现：读为 0

寄存器 10-2： RPINR1：外设引脚选择输入寄存器 1

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0
bit 15							
bit 8							

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
bit 7							
bit 0							

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现：读为 0

bit 13-8 **INT3R<5:0>**：将外部中断 3 (INT3) 分配给对应 RPin 或 RPIn 引脚的位

bit 7-6 未实现：读为 0

bit 5-0 **INT2R<5:0>**：将外部中断 2 (INT2) 分配给对应 RPin 或 RPIn 引脚的位

PIC24FJ256GB110 系列

寄存器 10-3: RPINR2: 外设引脚选择输入寄存器 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT4R5	INT4R4	INT4R3	INT4R2	INT4R1	INT4R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-6 未实现: 读为 0

bit 5-0 INT4R<5:0>: 将外部中断 4 (INT4) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-4: RPINR3: 外设引脚选择输入寄存器 3

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T3CKR5	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T2CKR5	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 T3CKR<5:0>: 将 Timer3 外部时钟 (T3CK) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 T2CKR<5:0>: 将 Timer2 外部时钟 (T2CK) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-5: RPINR4: 外设引脚选择输入寄存器 4

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T5CKR5	T5CKR4	T5CKR3	T5CKR2	T5CKR1	T5CKR0
bit 15				bit 8			

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T4CKR5	T4CKR4	T4CKR3	T4CKR2	T4CKR1	T4CKR0
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **T5CKR<5:0>**: 将 Timer5 外部时钟 (T5CK) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **T4CKR<5:0>**: 将 Timer4 外部时钟 (T4CK) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-6: RPINR7: 外设引脚选择输入寄存器 7

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC2R5	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0
bit 15				bit 8			

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC1R5	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **IC2R<5:0>**: 将输入捕捉 2 (IC2) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **IC1R<5:0>**: 将输入捕捉 1 (IC1) 分配给对应 RPn 或 RPIn 引脚的位

PIC24FJ256GB110 系列

寄存器 10-7: RPINR8: 外设引脚选择输入寄存器 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC4R5	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0
bit 15	bit 8						

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC3R5	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **IC4R<5:0>**: 将输入捕捉 4 (IC4) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **IC3R<5:0>**: 将输入捕捉 3 (IC3) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-8: RPINR9: 外设引脚选择输入寄存器 9

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC6R5	IC6R4	IC6R3	IC6R2	IC6R1	IC6R0
bit 15	bit 8						

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC5R5	IC5R4	IC5R3	IC5R2	IC5R1	IC5R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **IC6R<5:0>**: 将输入捕捉 6 (IC6) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **IC5R<5:0>**: 将输入捕捉 5 (IC5) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-9: RPINR10: 外设引脚选择输入寄存器 10

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC8R5	IC8R4	IC8R3	IC8R2	IC8R1	IC8R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC7R5	IC7R4	IC7R3	IC7R2	IC7R1	IC7R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **IC8R<5:0>**: 将输入捕捉 8 (IC8) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **IC7R<5:0>**: 将输入捕捉 7 (IC7) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-10: RPINR11: 外设引脚选择输入寄存器 11

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **OCFBR<5:0>**: 将输出比较故障 B (OCFB) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **OCFAR<5:0>**: 将输出比较故障 A (OCFA) 分配给对应 RPn 或 RPIn 引脚的位

PIC24FJ256GB110 系列

寄存器 10-11: RPINR15: 外设引脚选择输入寄存器 15

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC9R5	IC9R4	IC9R3	IC9R2	IC9R1	IC9R0
bit 15	bit 8						

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 IC9R<5:0>: 将输入捕捉 9 (IC9) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-0 未实现: 读为 0

寄存器 10-12: RPINR17: 外设引脚选择输入寄存器 17

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0
bit 15	bit 8						

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 U3RXR<5:0>: 将 UART3 接收 (U3RX) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-0 未实现: 读为 0

寄存器 10-13: RPINR18: 外设引脚选择输入寄存器 18

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **U1CTSR<5:0>**: 将 UART1 允许发送 (U1CTS) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **U1RXR<5:0>**: 将 UART1 接收 (U1RX) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-14: RPINR19: 外设引脚选择输入寄存器 19

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **U2CTSR<5:0>**: 将 UART2 允许发送 (U2CTS) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **U2RXR<5:0>**: 将 UART2 接收 (U2RX) 分配给对应 RPn 或 RPIn 引脚的位

PIC24FJ256GB110 系列

寄存器 10-15: RPINR20: 外设引脚选择输入寄存器 20

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0
bit 15							

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0
bit 7							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **SCK1R<5:0>**: 将 SPI1 时钟输入 (SCK1IN) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **SDI1R<5:0>**: 将 SPI1 数据输入 (SDI1) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-16: RPINR21: 外设引脚选择输入寄存器 21

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0
bit 15							

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0
bit 7							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

bit 15-14 未实现: 读为 0

bit 13-8 **U3CTSR<5:0>**: 将 UART3 允许发送 (U3CTS) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **SS1R<5:0>**: 将 SPI1 从选择输入 (SS1IN) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-17: RPINR22: 外设引脚选择输入寄存器 22

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **SCK2R<5:0>**: 将 SPI2 时钟输入 (SCK2IN) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **SDI2R<5:0>**: 将 SPI2 数据输入 (SDI2) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-18: RPINR23: 外设引脚选择输入寄存器 23

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-6 未实现: 读为 0

bit 5-0 **SS2R<5:0>**: 将 SPI2 从选择输入 (SS2IN) 分配给对应 RPn 或 RPIn 引脚的位

PIC24FJ256GB110 系列

寄存器 10-19: RPINR27: 外设引脚选择输入寄存器 27

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U4CTSR5	U4CTSR4	U4CTSR3	U4CTSR2	U4CTSR1	U4CTSR0
bit 15	bit 8						

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U4RXR5	U4RXR4	U4RXR3	U4RXR2	U4RXR1	U4RXR0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **U4CTSR<5:0>**: 将 UART4 允许发送 (U4CTS) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **U4RXR<5:0>**: 将 UART4 接收 (U4RX) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-20: RPINR28: 外设引脚选择输入寄存器 28

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK3R5	SCK3R4	SCK3R3	SCK3R2	SCK3R1	SCK3R0
bit 15	bit 8						

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI3R5	SDI3R4	SDI3R3	SDI3R2	SDI3R1	SDI3R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **SCK3R<5:0>**: 将 SPI3 时钟输入 (SCK3IN) 分配给对应 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **SDI3R<5:0>**: 将 SPI3 数据输入 (SDI3) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-21: RPINR29: 外设引脚选择输入寄存器 29

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS3R5	SS3R4	SS3R3	SS3R2	SS3R1	SS3R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-6 未实现: 读为 0

bit 5-0 **SS3R<5:0>**: 将 SPI3 从选择输入 (SS3IN) 分配给对应 RPn 或 RPIn 引脚的位

寄存器 10-22: RPOR0: 外设引脚选择输出寄存器 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP1R<5:0>**: RP1 输出引脚映射位

将外设输出编号 n 分配给引脚 RP1 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP0R<5:0>**: RP0 输出引脚映射位

将外设输出编号 n 分配给引脚 RP0 (请参见表 10-3 了解外设功能编号)

PIC24FJ256GB110 系列

寄存器 10-23: RPOR1: 外设引脚选择输出寄存器 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP3R5	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP2R5	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 RP3R<5:0>: RP3 输出引脚映射位

将外设输出编号 n 分配给引脚 RP3 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 RP2R<5:0>: RP2 输出引脚映射位

将外设输出编号 n 分配给引脚 RP2 (请参见表 10-3 了解外设功能编号)

寄存器 10-24: RPOR2: 外设引脚选择输出寄存器 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP5R5 ⁽¹⁾	RP5R4 ⁽¹⁾	RP5R3 ⁽¹⁾	RP5R2 ⁽¹⁾	RP5R1 ⁽¹⁾	RP5R0 ⁽¹⁾
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP4R5	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

bit 15-14 未实现: 读为 0

bit 13-8 RP5R<5:0>: RP5 输出引脚映射位⁽¹⁾

将外设输出编号 n 分配给引脚 RP5 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 RP4R<5:0>: RP4 输出引脚映射位

将外设输出编号 n 分配给引脚 RP4 (请参见表 10-3 了解外设功能编号)

注 1: 在 64 引脚器件上未实现; 读为 0。

寄存器 10-25: RPOR3: 外设引脚选择输出寄存器 3

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP7R<5:0>: RP7 输出引脚映射位**

将外设输出编号 n 分配给引脚 RP7 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

RP6R<5:0>: RP6 输出引脚映射位

将外设输出编号 n 分配给引脚 RP6 (请参见表 10-3 了解外设功能编号)

寄存器 10-26: RPOR4: 外设引脚选择输出寄存器 4

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP9R5	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP8R5	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP9R<5:0>: RP9 输出引脚映射位**

将外设输出编号 n 分配给引脚 RP9 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

RP8R<5:0>: RP8 输出引脚映射位

将外设输出编号 n 分配给引脚 RP8 (请参见表 10-3 了解外设功能编号)

PIC24FJ256GB110 系列

寄存器 10-27: RPOR5: 外设引脚选择输出寄存器 5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP11R5	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0
bit 15	bit 8						

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP10R5	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP11R<5:0>**: RP11 输出引脚映射位

将外设输出编号 n 分配给引脚 RP11 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP10R<5:0>**: RP10 输出引脚映射位

将外设输出编号 n 分配给引脚 RP10 (请参见表 10-3 了解外设功能编号)

寄存器 10-28: RPOR6: 外设引脚选择输出寄存器 6

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP13R5	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0
bit 15	bit 8						

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP12R5	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP13R<5:0>**: RP13 输出引脚映射位

将外设输出编号 n 分配给引脚 RP13 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP12R<5:0>**: RP12 输出引脚映射位

将外设输出编号 n 分配给引脚 RP12 (请参见表 10-3 了解外设功能编号)

寄存器 10-29: RPOR7: 外设引脚选择输出寄存器 7

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP15R5 ⁽¹⁾	RP15R4 ⁽¹⁾	RP15R3 ⁽¹⁾	RP15R2 ⁽¹⁾	RP15R1 ⁽¹⁾	RP15R0 ⁽¹⁾
bit 15	bit 8						

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP14R5	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP15R<5:0>: RP15** 输出引脚映射位⁽¹⁾

将外设输出编号 n 分配给引脚 RP15 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP14R<5:0>: RP14** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP14 (请参见表 10-3 了解外设功能编号)

注 1: 在 64 引脚器件上未实现; 读为 0。

寄存器 10-30: RPOR8: 外设引脚选择输出寄存器 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP17R5	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0
bit 15	bit 8						

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP16R5	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP17R<5:0>: RP17** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP17 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP16R<5:0>: RP16** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP16 (请参见表 10-3 了解外设功能编号)

PIC24FJ256GB110 系列

寄存器 10-31: RPOR9: 外设引脚选择输出寄存器 9

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0
bit 15	bit 8						

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP19R<5:0>**: RP19 输出引脚映射位

将外设输出编号 n 分配给引脚 RP19 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP18R<5:0>**: RP18 输出引脚映射位

将外设输出编号 n 分配给引脚 RP18 (请参见表 10-3 了解外设功能编号)

寄存器 10-32: RPOR10: 外设引脚选择输出寄存器 10

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0
bit 15	bit 8						

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP21R<5:0>**: RP21 输出引脚映射位

将外设输出编号 n 分配给引脚 RP21 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP20R<5:0>**: RP20 输出引脚映射位

将外设输出编号 n 分配给引脚 RP20 (请参见表 10-3 了解外设功能编号)

寄存器 10-33: RPOR11: 外设引脚选择输出寄存器 11

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP23R5	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0
bit 15	bit 8						

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP22R5	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP23R<5:0>**: RP23 输出引脚映射位

将外设输出编号 n 分配给引脚 RP23 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP22R<5:0>**: RP22 输出引脚映射位

将外设输出编号 n 分配给引脚 RP22 (请参见表 10-3 了解外设功能编号)

寄存器 10-34: RPOR12: 外设引脚选择输出寄存器 12

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP25R5	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0
bit 15	bit 8						

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP24R5	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP25R<5:0>**: RP25 输出引脚映射位

将外设输出编号 n 分配给引脚 RP25 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP24R<5:0>**: RP24 输出引脚映射位

将外设输出编号 n 分配给引脚 RP24 (请参见表 10-3 了解外设功能编号)

PIC24FJ256GB110 系列

寄存器 10-35: RPOR13: 外设引脚选择输出寄存器 13

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP27R5	RP27R4	RP27R3	RP27R2	RP27R1	RP27R0
bit 15	bit 8						

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP26R5	RP26R4	RP26R3	RP26R2	RP26R1	RP26R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 RP27R<5:0>: RP27 输出引脚映射位

将外设输出编号 n 分配给引脚 RP27 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 RP26R<5:0>: RP26 输出引脚映射位

将外设输出编号 n 分配给引脚 RP26 (请参见表 10-3 了解外设功能编号)

寄存器 10-36: RPOR14: 外设引脚选择输出寄存器 14

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP29R5	RP29R4	RP29R3	RP29R2	RP29R1	RP29R0
bit 15	bit 8						

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP28R5	RP28R4	RP28R3	RP28R2	RP28R1	RP28R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

bit 15-14 未实现: 读为 0

bit 13-8 RP29R<5:0>: RP29 输出引脚映射位

将外设输出编号 n 分配给引脚 RP29 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 RP28R<5:0>: RP28 输出引脚映射位

将外设输出编号 n 分配给引脚 RP28 (请参见表 10-3 了解外设功能编号)

寄存器 10-37: RPOR15: 外设引脚选择输出寄存器 15

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP31R5 ⁽¹⁾	RP31R4 ⁽¹⁾	RP31R3 ⁽¹⁾	RP31R2 ⁽¹⁾	RP31R1 ⁽¹⁾	RP31R0 ⁽¹⁾
bit 15	bit 8						

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP30R5	RP30R4	RP30R3	RP30R2	RP30R1	RP30R0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP31R<5:0>: RP31** 输出引脚映射位 ⁽¹⁾

将外设输出编号 n 分配给引脚 RP31 (请参见表 10-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP30R<5:0>: RP30** 输出引脚映射位 ⁽²⁾

将外设输出编号 n 分配给引脚 RP30 (请参见表 10-3 了解外设功能编号)

注 1: 在 64 引脚和 80 引脚器件上未实现; 读为 0。

2: 在 64 引脚器件上未实现; 读为 0。

PIC24FJ256GB110 系列

注:

11.0 TIMER1

注: 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见《PIC24F 系列参考手册》的第 14 章“定时器”(DS39704A_CN)。

Timer1 模块是一个 16 位定时器, 可作为实时时钟(RTC)的时间计数器, 或作为自由运行的间隔定时器/计数器。Timer1 可在以下三种模式下工作:

- 16 位定时器
- 16 位同步计数器
- 16 位异步计数器

Timer1 还支持以下功能:

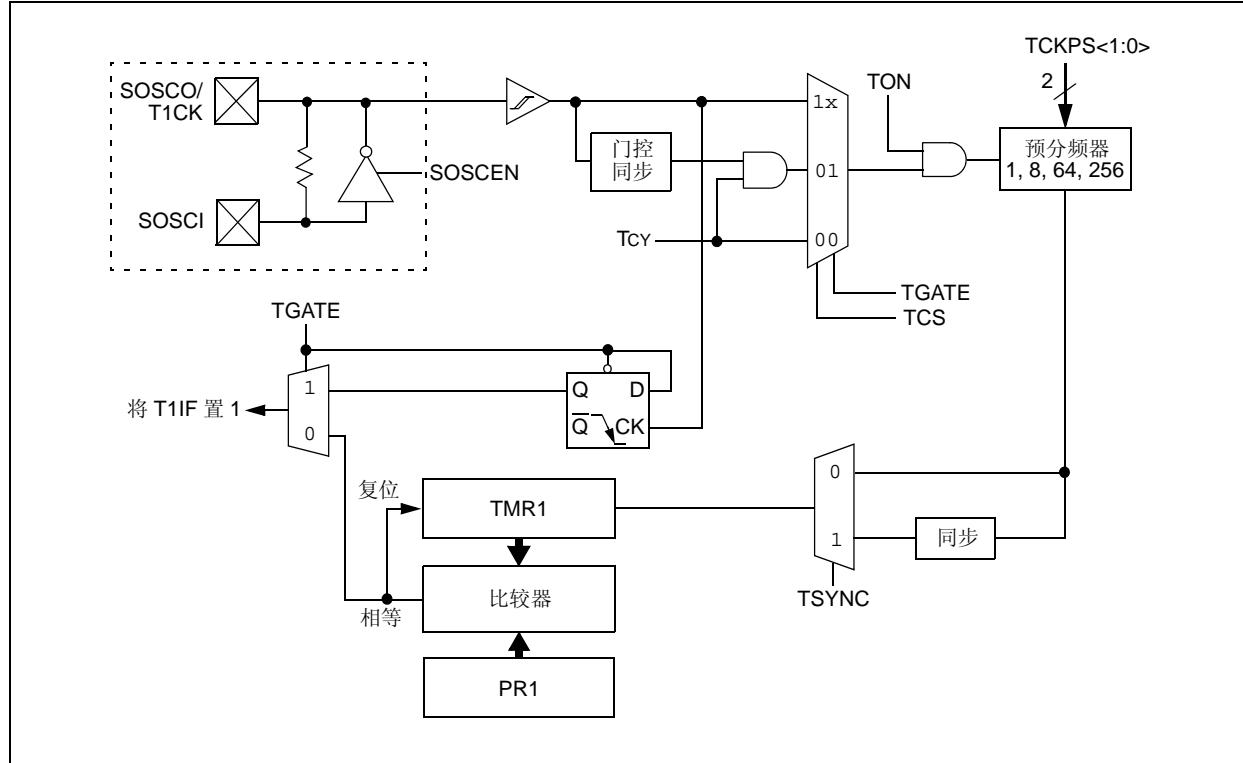
- 定时器门控操作
- 可选择的预分频比设置
- 在 CPU 空闲和休眠模式期间的定时器操作
- 在 16 位周期寄存器匹配时或外部门控信号的下降沿产生中断

图 11-1 给出了 16 位定时器模块的框图。

配置 Timer1 的操作:

1. 将 TON 位置 1 (= 1)。
2. 使用 TCKPS<1:0> 位选择定时器预分频比。
3. 使用 TCS 和 TGATE 位设置时钟和门控模式。
4. 将 TSYNC 位置 1 或清零来配置同步或异步操作。
5. 将定时器的周期值装入 PR1 寄存器。
6. 如果需要中断, 将中断允许位 T1IE 置 1。使用优先级位 T1IP<2:0> 来设置中断优先级。

图 11-1: 16 位 TIMER1 模块框图



PIC24FJ256GB110 系列

寄存器 11-1: T1CON: TIMER1 控制寄存器⁽¹⁾

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **TON:** Timer1 使能位
1 = 启动 16 位 Timer1
0 = 停止 16 位 Timer1
- bit 14 未实现: 读为 0
- bit 13 **TSIDL:** 空闲模式停止位
1 = 当器件进入空闲模式时, 模块停止工作
0 = 在空闲模式下模块继续工作
- bit 12-7 未实现: 读为 0
- bit 6 **TGATE:** Timer1 门控时间累加使能位
当 TCS = 1 时:
该位为无关位。
当 TCS = 0 时:
1 = 使能门控时间累加
0 = 禁止门控时间累加
- bit 5-4 **TCKPS<1:0>:** Timer1 输入时钟预分频比选择位
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1
- bit 3 未实现: 读为 0
- bit 2 **TSYNC:** Timer1 外部时钟输入同步选择位
当 TCS = 1 时:
1 = 同步外部时钟输入
0 = 不同步外部时钟输入
当 TCS = 0 时:
该位为无关位。
- bit 1 **TCS:** Timer1 时钟源选择位
1 = 来自 T1CK 引脚的外部时钟 (上升沿触发计数)
0 = 内部时钟 (Fosc/2)
- bit 0 未实现: 读为 0

注 1: 在定时器运行过程中 (TON = 1) 修改 TxCON 的值, 会导致定时器预分频计数器复位, 因而建议不要这样做。

12.0 TIMER2/3 和 TIMER4/5

注: 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见《PIC24F 系列参考手册》的第 14 章“定时器”(DS39704A_CN)。

Timer2/3 和 Timer4/5 模块为 32 位定时器, 也可被配置为 4 个具有可选工作模式的独立 16 位定时器。

作为 32 位定时器, Timer2/3 和 Timer4/5 具有三种工作模式:

- 具有所有 16 位工作模式 (异步计数器模式除外) 的两个独立的 16 位定时器
- 单个 32 位定时器
- 单个 32 位同步计数器

还支持以下功能:

- 定时器门控操作
- 可选择的预分频比设置
- 空闲和休眠模式下的定时器工作
- 在 32 位周期寄存器匹配时产生中断
- ADC 事件触发器 (仅限 Timer4/5)

所有 4 个 16 位定时器都能单独作为同步定时器或计数器。它们也提供上面所列的功能, 但 ADC 事件触发功能除外; 这仅由 Timer3 实现。通过设置 T2CON、T3CON、T4CON 和 T5CON 寄存器中的相应位来确定工作模式和使能特性。T2CON 和 T4CON 在寄存器 12-1 中作了一般介绍; T3CON 和 T5CON 如寄存器 12-2 所示。

对于 32 位定时器 / 计数器工作, Timer2 和 Timer4 是 32 位定时器的低位字, 而 Timer3 和 Timer5 是高位字。

注: 对于 32 位工作, T3CON 和 T5CON 控制位将被忽略。设置和控制只使用 T2CON 和 T4CON 控制位。32 位定时器模块采用 Timer2 和 Timer4 时钟和门控输入, 但中断由 Timer3 或 Timer5 中断标志位产生。

要将 Timer2/3 或 Timer4/5 配置为 32 位工作:

1. 将 T32 位置 1 (T2CON<3> 或 T4CON<3> = 1)。
2. 使用 TCKPS<1:0> 位为 Timer2 或 Timer4 选择预分频比。
3. 使用 TCS 和 TGATE 位设置时钟和门控模式。如果 TCS 被设置为外部时钟, 则 RPINRx (TxCK) 必须配置为可用的 RPn 引脚。更多信息, 请参见第 10.4 节“外设引脚选择”。
4. 装入定时器的周期值。PR3 (或 PR5) 将包含值的高位字, 而 PR2 (或 PR4) 包含低位字。
5. 如果需要中断, 将中断允许位 T3IE 或 T5IE 置 1; 使用优先级位 T3IP<2:0> 或 T5IP<2:0> 来设置中断优先级。请注意, Timer2 或 Timer4 控制定时器, 而中断由 Timer3 或 Timer5 产生。
6. 将 TON 位置 1 (= 1)。

任意时刻定时器的值被存储在寄存器对 TMR3:TMR2 (或 TMR5:TMR4) 中。TMR3 (TMR5) 总是包含计数值的高位字, 而 TMR2 (TMR4) 包含低位字。

要将任一定时器配置为独立的 16 位工作:

1. 清零与该定时器对应的 T32 位 (Timer2 和 Timer3 为 T2CON<3>; Timer4 和 Timer5 为 T4CON<3>)。
2. 使用 TCKPS<1:0> 位选择定时器预分频比。
3. 使用 TCS 和 TGATE 位设置时钟和门控模式。更多信息, 请参见第 10.4 节“外设引脚选择”。
4. 将定时器的周期值装入 PRx 寄存器。
5. 如果需要中断, 将中断允许位 TxIE 置 1; 使用中断优先级位 TxIP<2:0> 来设置中断优先级。
6. 将 TON 位置 1 (TxCON<15> = 1)。

PIC24FJ256GB110 系列

图 12-1: TIMER2/3 和 TIMER4/5 (32 位) 框图

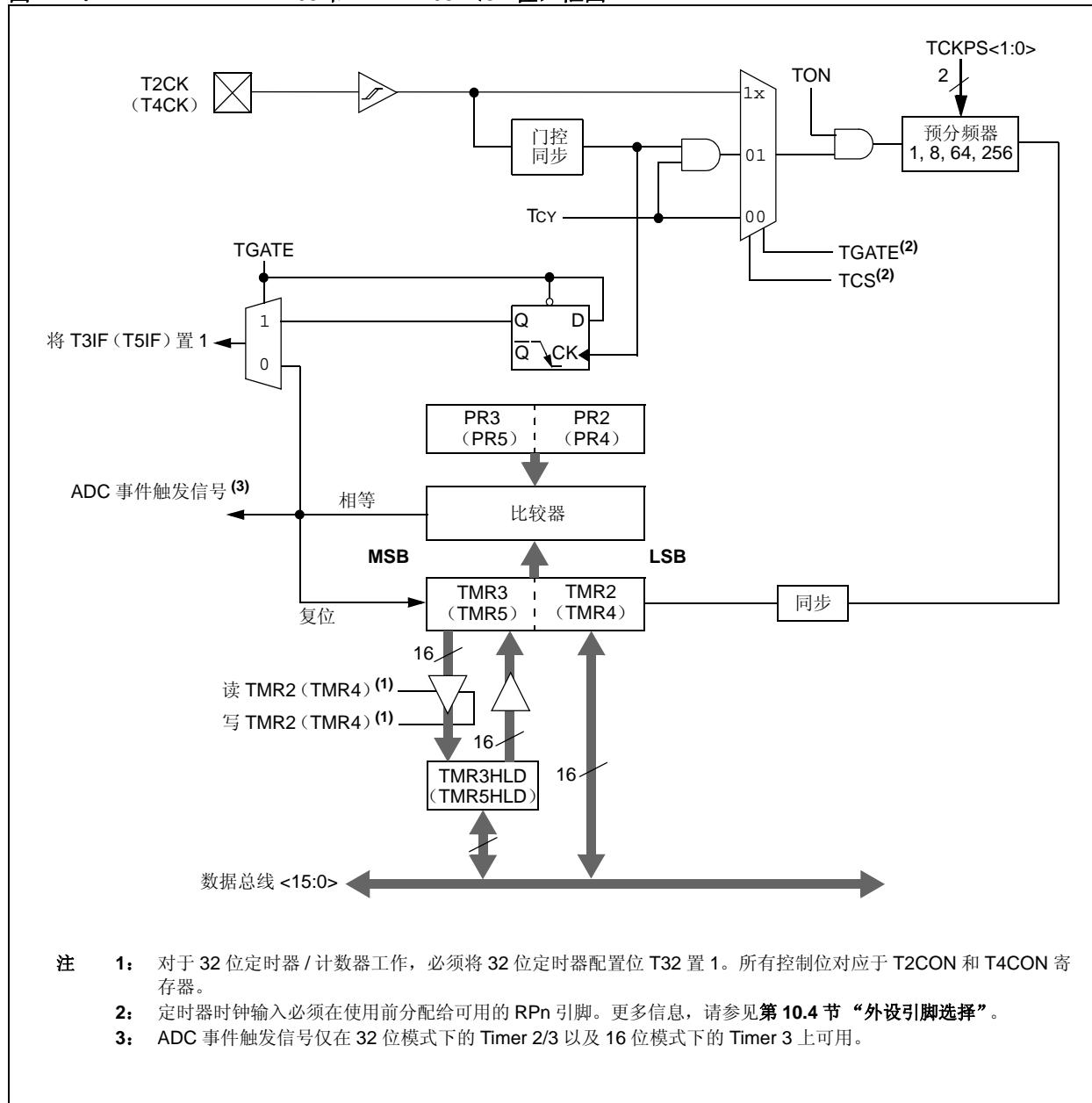


图 12-2: TIMER2 和 TIMER4 (16 位同步) 框图

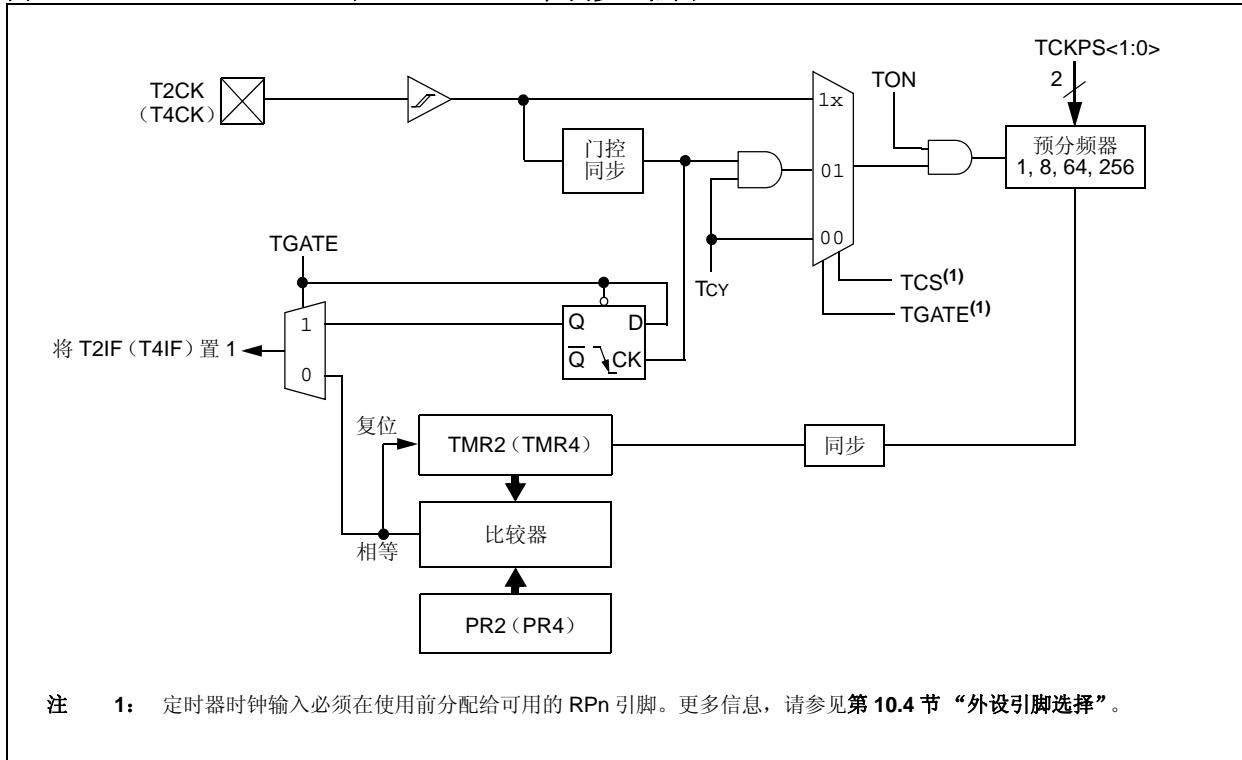
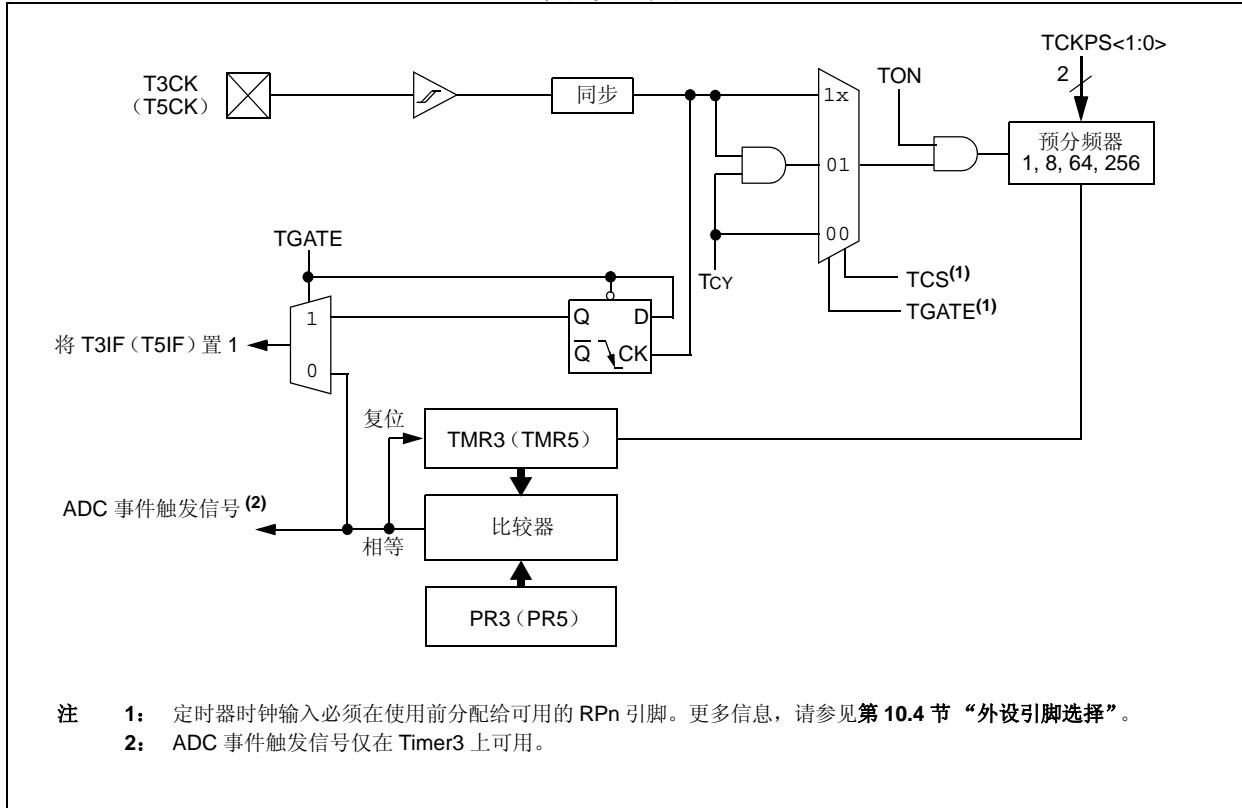


图 12-3: TIMER3 和 TIMER5 (16 位异步) 框图



PIC24FJ256GB110 系列

寄存器 12-1: TxCON: TIMER2 和 TIMER4 控制寄存器 (3)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15	bit 8						

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	T32 ⁽¹⁾	—	TCS ⁽²⁾	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **TON:** Timerx 使能位

当 TxCON<3> = 1 时:

1 = 启动 32 位 Timerx/y

0 = 停止 32 位 Timerx/y

当 TxCON<3> = 0 时:

1 = 启动 16 位 Timerx

0 = 停止 16 位 Timerx

bit 14 未实现: 读为 0

bit 13 **TSIDL:** 空闲模式停止位

1 = 当器件进入空闲模式时, 模块停止工作

0 = 在空闲模式下模块继续工作

bit 12-7 未实现: 读为 0

bit 6 **TGATE:** Timerx 门控时间累加使能位

当 TCS = 1 时:

该位为无关位。

当 TCS = 0 时:

1 = 使能门控时间累加

0 = 禁止门控时间累加

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

11 = 1:256

10 = 1:64

01 = 1:8

00 = 1:1

bit 3 **T32:** 32 位定时器模式选择位 (1)

1 = Timerx 和 Timery 形成一个 32 位定时器

0 = Timerx 和 Timery 充当两个 16 位定时器

在 32 位模式下, T3CON 控制位不影响 32 位定时器的工作。

bit 2 未实现: 读为 0

bit 1 **TCS:** Timerx 时钟源选择位 (2)

1 = 来自 TxCK 引脚的外部时钟 (上升沿触发计数)

0 = 内部时钟 (Fosc/2)

bit 0 未实现: 读为 0

注 1: 在 32 位模式下, T3CON 或 T5CON 控制位不影响 32 位定时器的工作。

2: 如果 TCS = 1, 则 RPINRx (TxCK) 必须配置为可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

3: 在定时器运行过程中 (TON = 1) 修改 TxCON 的值, 会导致定时器预分频计数器复位, 因而建议不要这样做。

寄存器 12-2: TyCON: TIMER3 和 TIMER5 控制寄存器⁽³⁾

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON ⁽¹⁾	—	TSIDL ⁽¹⁾	—	—	—	—	—
bit 15	bit 8						

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE ⁽¹⁾	TCKPS1 ⁽¹⁾	TCKPS0 ⁽¹⁾	—	—	TCS ^(1,2)	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **TON:** Timery 使能位⁽¹⁾

1 = 启动 16 位 Timery

0 = 停止 16 位 Timery

bit 14 未实现: 读为 0

bit 13 **TSIDL:** 空闲模式停止位⁽¹⁾

1 = 当器件进入空闲模式时, 模块停止工作

0 = 在空闲模式下模块继续工作

bit 12-7 未实现: 读为 0

bit 6 **TGATE:** Timery 门控时间累加使能位⁽¹⁾

当 TCS = 1 时:

该位为无关位。

当 TCS = 0 时:

1 = 使能门控时间累加

0 = 禁止门控时间累加

bit 5-4 **TCKPS<1:0>:** Timery 输入时钟预分频比选择位⁽¹⁾

11 = 1:256

10 = 1:64

01 = 1:8

00 = 1:1

bit 3-2 未实现: 读为 0

bit 1 **TCS:** Timery 时钟源选择位^(1,2)

1 = 来自 TyCK 引脚的外部时钟 (上升沿触发计数)

0 = 内部时钟 (Fosc/2)

bit 0 未实现: 读为 0

注 1: 当使能 32 位工作 (T2CON<3> 或 T4CON<3> = 1) 时, 这些位对 Timery 的工作没有影响; 所有定时器功能都通过 T2CON 和 T4CON 进行设置。

2: 如果 TCS = 1, 则 RPINRx (TxCK) 必须配置为可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

3: 在定时器运行过程中 (TON = 1) 修改 TyCON 的值, 会导致定时器预分频计数器复位, 因而建议不要这样做。

PIC24FJ256GB110 系列

注:

13.0 带专用定时器的输入捕捉

注：本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》的第 34 章“带专用定时器的输入捕捉”(DS39722A_CN)。

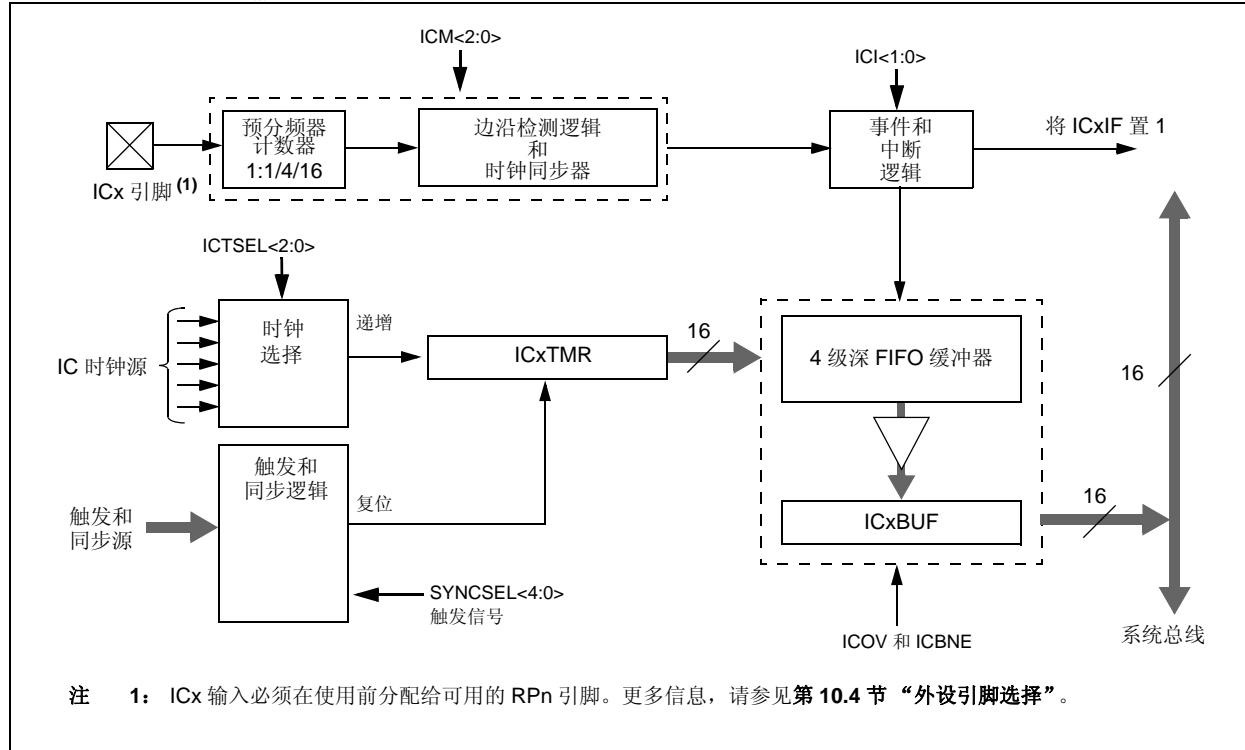
PIC24FJ256GB110 系列中的器件都具有 9 个独立的输入捕捉模块。每个模块都提供用于捕捉外部脉冲事件和产生中断的一系列配置和工作选项。

输入捕捉模块的主要特性包括：

- 通过级联两个相邻模块，可通过硬件配置为所有的 32 位工作模式
- 输出比较操作的同步和触发模式，最多可有 30 个用户可选择的触发 / 同步源
- 用于捕捉和保存几个事件的定时器值的 4 级深 FIFO 缓冲器
- 可配置中断产生
- 每个模块最多有 6 个时钟源可用，驱动独立的内部 16 位计数器

该模块通过两个寄存器 ICxCON1 (寄存器 13-1) 和 ICxCON2 (寄存器 13-2) 进行控制。图 13-1 给出了该模块的一般框图。

图 13-1： 输入捕捉框图



13.1 一般工作模式

13.1.1 同步和触发模式

默认情况下，输入捕捉模块工作于自由运行模式。内部 16 位计数器 ICxTMR 连续向上计数，每当溢出时从 FFFFh 回到 0000h，其周期与所选的外部时钟源同步。发生捕捉事件时，内部计数器的当前 16 位值写入 FIFO 缓冲器。

在同步模式下，一旦模块所选时钟源使能，它就开始捕捉 ICx 引脚上的事件。只要所选同步源发生事件，内部计数器就复位。在触发模式下，模块等待来自其他内部模块的同步事件发生，然后允许运行内部计数器。

标准的自由运行操作可通过将 SYNCSEL 位设置为 00000 并清零 ICTRIG 位 (ICxCON2<7>) 选择。只要 SYNCSEL 位设置为除 00000 外的任何值，就选择同步和触发模式。 ICTRIG 位选择同步或触发模式；将该位置 1 会选择触发模式操作。在两种模式下， SYNCSEL 位都确定同步 / 触发源。

当 SYNCSEL 位设置为 00000 并且 ICTRIG 置 1 时，该模块工作于软件触发模式。在这种情况下，捕捉操作可通过将 TRIGSTAT 位 (ICxCON2<6>) 手动置 1 启动。

13.1.2 级联（32 位）模式

默认情况下，每个模块都以其自身的 16 位定时器独立运行。要提高分辨率，可将相邻的奇偶模块配置为单个 32 位模块。（例如，模块 1 和 2 配对，3 和 4 配对等。）编号为奇数的模块（ICx）提供 32 位寄存器对的低 16 位，偶数模块（ICy）提供高 16 位。ICx 寄存器的归零会导致它们对应的 ICy 寄存器递增。

级联操作是通过将两个模块的 IC32 位（ICxCON2<8>）置 1 在硬件中配置的。

13.2 捕捉操作

输入捕捉模块可配置为在 ICx 的上升沿或所有跳变处捕捉定时器值并产生中断。捕捉可配置为在所有上升沿发生，或者只在部分上升沿（每到第 4 个或第 16 个）发生。可单独配置中断，使之在每个事件发生时或在一组事件发生时产生。

要为捕捉操作设置模块：

1. 将 ICx 输入引脚配置为可用外设引脚选择引脚之一。
2. 如果要使用同步模式，在执行前禁止同步源。
3. 通过读 ICxBUF 直到 ICBNE 位（ICxCON1<3>）清零，确保任何先前数据都已从 FIFO 删除。
4. 将 SYNCSEL 位（ICxCON2<4:0>）设置为所需的同步 / 触发源。
5. 将所需的时钟源的 ICTSEL 位（ICxCON1<12:10>）置 1。
6. 将 ICI 位（ICxCON1<6:5>）设置为所需的中断频率。
7. 选择同步或触发模式操作：
 - a) 确认 SYNCSEL 位未设置为 00000。
 - b) 对于同步模式，清零 ICTRIG 位（ICxCON2<7>）。
 - c) 对于触发模式，将 ICTRIG 置 1，清零 TRIGSTAT 位（ICxCON2<6>）。
8. 将 ICM 位（ICxCON1<2:0>）设置为所需的工作模式。
9. 使能所选的触发 / 同步源。

对于 32 位级联操作，设置步骤略有不同：

1. 将两个模块的 IC32 位（ICyCON2<8> 和 ICxCON2<8>）置 1，先使能偶数编号的模块。这样可以确保这两个模块同时启动。
2. 将两个模块的 ICTSEL 和 SYNCSEL 置 1，选择同一个同步/触发和时基源。先设置偶数模块，再设置奇数模块。两个模块都必须使用相同的 ICTSEL 和 SYNCSEL 设置。
3. 清零偶数模块的 ICTRIG 位（ICyCON2<7>）；这会强制该模块运行于与奇数模块同步的模式，与其触发设置无关。
4. 使用奇数模块的 ICI 位（ICxCON1<6:5>）设置为所需的中断频率。
5. 使用奇数模块的 ICTRIG 位（ICxCON2<7>）配置触发或同步模式操作。

注： 对于同步模式操作，使能同步源作为最后步骤。两个输入捕捉模块都将保持复位状态，直到使能同步源。

6. 使用奇数模块的 ICM 位（ICxCON1<2:0>）设置所需的捕捉模式。

时基和触发/同步源使能时，该模块已准备好捕捉事件。ICBNE 位（ICxCON1<3>）变为置 1 时，FIFO 中至少有一个捕捉值可用。从 FIFO 读取输入捕捉值，直到 ICBNE 清零。

对于 32 位操作，要同时读 ICxBUF 和 ICyBUF 获取完整的 32 位定时器值（ICxBUF 用作低位字，ICyBUF 用作高位字）。奇数模块的 ICBNE 位（ICxCON1<3>）置 1 时，FIFO 缓冲器中至少有一个捕捉值可用。继续读缓冲寄存器，直到 ICBNE 清零（由硬件自动执行）。

寄存器 13-1: ICxCON1: 输入捕捉 x 控制寄存器 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R-0, HC	R-0, HC	R/W-0	R/W-0	R/W-0
—	ICI1	ICI0	ICOV	ICBNE	ICM2 ⁽¹⁾	ICM1 ⁽¹⁾	ICM0 ⁽¹⁾
bit 7							bit 0

图注:	HC = 硬件清零位
R = 可读位	W = 可写位
-n = POR 时的值	1 = 置 1 0 = 清零 x = 未知

bit 15-14	未实现: 读为 0
bit 13	ICSIDL: 输入捕捉 x 模块在空闲模式下停止的控制位 1 = 在 CPU 空闲模式下输入捕捉模块停止工作 0 = 在 CPU 空闲模式下输入捕捉模块继续工作
bit 12-10	ICTSEL<2:0>: 输入捕捉定时器选择位 111 = 系统时钟 (Fosc/2) 110 = 保留 101 = 保留 100 = Timer1 011 = Timer5 010 = Timer4 001 = Timer2 000 = Timer3
bit 9-7	未实现: 读为 0
bit 6-5	ICI<1:0>: 选择每次发生中断捕捉的次数的位 11 = 每 4 次捕捉事件中断一次 10 = 每 3 次捕捉事件中断一次 01 = 每 2 次捕捉事件中断一次 00 = 每次捕捉事件中断一次
bit 4	ICOV: 输入捕捉 x 溢出状态标志位 (只读) 1 = 发生了输入捕捉溢出 0 = 未发生输入捕捉溢出
bit 3	ICBNE: 输入捕捉 x 缓冲器空状态位 (只读) 1 = 输入捕捉缓冲器非空, 至少可以再读一次捕捉值 0 = 输入捕捉缓冲器为空
bit 2-0	ICM<2:0>: 输入捕捉模式选择位 ⁽¹⁾ 111 = 中断模式: 当器件处于休眠或空闲模式时, 输入捕捉仅用作中断引脚 (只检测上升沿, 所有其他控制位都不适用) 110 = 未使用 (模块被禁止) 101 = 预分频器捕捉模式: 每 16 个上升沿捕捉一次 100 = 预分频器捕捉模式: 每 4 个上升沿捕捉一次 011 = 简单捕捉模式: 每个上升沿捕捉一次 010 = 简单捕捉模式: 每个下降沿捕捉一次 001 = 边沿检测捕捉模式: 每个边沿 (上升沿和下降沿) 捕捉一次, ICI<1:0> 位不控制该模式下的中断产生 000 = 输入捕捉模块关闭

注 1: ICx 输入必须也配置为可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

PIC24FJ256GB110 系列

寄存器 13-2: ICxCON2: 输入捕捉 x 控制寄存器 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	IC32
bit 15							bit 8

R/W-0	R/W-0 HS	U-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-1
ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

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

bit 15-9	未实现: 读为 0
bit 8	IC32: 级联两个 IC 模块使能位 (32 位操作) 1 = ICx 和 ICy 级联起来用作 32 位模块 (该位必须在两个模块中置 1) 0 = ICx 单独用作 16 位模块
bit 7	ICTRIG: ICx 触发 / 同步选择位 1 = 用 SYNCSELx 位指定的源触发 ICx 0 = 用 SYNCSELx 位指定的源同步 ICx
bit 6	TRIGSTAT: 定时器触发状态位 1 = 定时器源已触发, 正在运行 (硬件置 1, 可用软件置 0) 0 = 定时器源未触发, 保持清零
bit 5	未实现: 读为 0
bit 4-0	SYNCSEL<4:0>: 触发 / 同步源选择位 11111 = 保留 11110 = 输入捕捉 9 11101 = 输入捕捉 6 11100 = CTMU ⁽¹⁾ 11011 = A/D ⁽¹⁾ 11010 = 比较器 3 ⁽¹⁾ 11001 = 比较器 2 ⁽¹⁾ 11000 = 比较器 1 ⁽¹⁾ 10111 = 输入捕捉 4 10110 = 输入捕捉 3 10101 = 输入捕捉 2 10100 = 输入捕捉 1 10011 = 输入捕捉 8 10010 = 输入捕捉 7 1000x = 保留 01111 = Timer 5 01110 = Timer 4 01101 = Timer 3 01100 = Timer 2 01011 = Timer 1 01010 = 输入捕捉 5 01001 = 输出比较 9 01000 = 输出比较 8 00111 = 输出比较 7 00110 = 输出比较 6 00101 = 输出比较 5 00100 = 输出比较 4 00011 = 输出比较 3 00010 = 输出比较 2 00001 = 输出比较 1 00000 = 不与任何其他模块同步

注 1: 只用这些输入作为触发源, 从不用作同步源。

14.0 带专用定时器的输出比较

注：本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》的第 35 章“带专用定时器的输出比较”(DS39723A_CN)。

PIC24FJ256GB110 系列中的器件都具有 9 个独立的输出比较模块。这些模块中的每一个都能提供范围较大的配置和工作选项，用于在发生器件内部事件时产生连续脉冲，并为驱动功率应用生成脉宽调制的波形。

输出比较模块的主要特性包括：

- 通过级联两个相邻模块，可通过硬件配置为所有的 32 位工作模式
- 输出比较操作的同步和触发模式，最多可有 30 个用户可选择的触发 / 同步源
- 两个独立的周期寄存器（主寄存器 OCxR 和辅助寄存器 OCxRS）在生成不同宽度脉冲时提供了更大的灵活性
- 可配置为输出事件时生成单脉冲或连续脉冲，也可以配置为生成连续 PWM 波形
- 每个模块最多有 6 个时钟源可用，驱动独立的内部 16 位计数器

14.1 一般工作模式

14.1.1 同步和触发模式

默认情况下，输出比较模块工作于自由运行模式。内部 16 位计数器 OCxTMR 连续向上计数，每当溢出时从 FFFFh 回到 0000h，其周期与所选的外部时钟源同步。每当内部计数器和周期寄存器之一发生匹配时，就会发生比较或 PWM 事件。

在同步模式下，一旦模块所选时钟源使能，它就开始执行其比较或 PWM 操作。只要所选同步源发生事件，该模块的内部计数器就复位。在触发模式下，模块等待来自其他内部模块的同步事件发生，然后允许运行计数器。

默认情况下，或 SYNCSEL 位 (OCxCON2<4:0>) 设置为 00000 时，选择自由运行模式。SYNCSEL 位设置为除 00000 外的任何值时选择同步或触发模式。OCTRIG 位 (OCxCON2<7>) 选择同步或触发模式；将该位置 1 将选择触发模式工作。在两种模式下，SYNCSEL 位都确定同步 / 触发源。

14.1.2 级联 (32 位) 模式

默认情况下，每个模块都以其自身的一组 16 位定时器和占空比寄存器独立运行。要提高分辨率，可将相邻的奇偶模块配置为单个 32 位模块。（例如，模块 1 和 2 配对，3 和 4 配对等。）编码为奇数的模块 (OCx) 提供 32 位寄存器对的低 16 位，偶数模块 (OCy) 提供高 16 位。OCx 寄存器的归零会导致它们对应的 OCy 寄存器递增。

级联操作是通过将两个模块的 OC32 位 (OCxCON2<8>) 置 1 在硬件中配置的。

14.2 比较操作

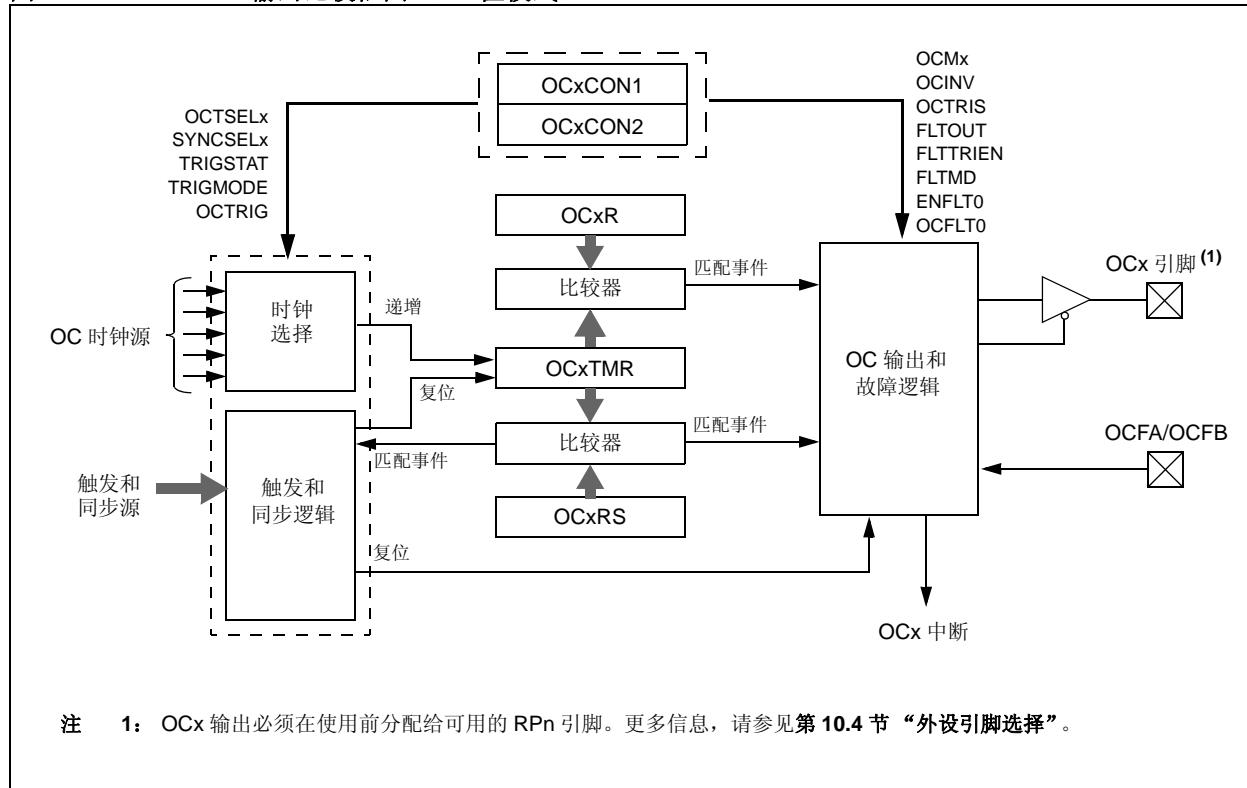
在比较模式下（图 14-1），输出比较模块可配置为产生单次脉冲或连续脉冲；也可以在每个定时器事件时重复翻转输出引脚。

要为比较操作设置模块：

1. 将 OCx 输出配置为可用外设引脚选择引脚之一。
2. 计算 OCxR 和（对于双比较模式）OCxRS 占空比寄存器所需的值：
 - a) 确定指令时钟周期时间。考虑定时器源的外部时钟频率（如果使用）和定时器预分频比的设置。
 - b) 计算从定时器起始值（0000h）到输出脉冲的上升沿所需的时间。
 - c) 根据所需的脉冲宽度和到脉冲上升沿的时间计算出现脉冲下降沿的时间。

3. 将上升边沿值写入 OCxR，将下降边沿值写入 OCxRS。
4. 将定时器周期寄存器 PRy 的值设置为等于或大于 OCxRS 中的值。
5. 为相应的比较操作设置 OCM<2:0> 位（= 0xx）。
6. 对于触发模式操作，将 OCTRIG 置 1 可使能触发模式。置 1 或清零 TRIGMODE 配置触发操作，置 1 或清零 TRIGSTAT 选择硬件或软件触发。对于同步模式，清零 OCTRIG。
7. 设置 SYNCSEL<4:0> 位以配置触发或同步源。如果需要自由运行定时器操作，将 SYNCSEL 位设置为 00000（无同步 / 触发源）。
8. 用 OCTSEL<2:0> 位选择时基源。如果需要，将使能比较时基进行计数的选定定时器的 TON 位置 1。同步模式操作在时基使能时启动；触发模式操作在触发源事件发生时启动。

图 14-1：输出比较框图（16 位模式）



对于 32 位级联操作，以下步骤也是必需的：

1. 为两个寄存器 (OCyCON2<8>和OCxCON2<8>) 设置 OC32 位。先使能偶数编码的模块，以确保模块同时启动。
2. 清零偶数模块的 OCTRIG 位 (OCyCON2) 使模块运行于同步模式。
3. 为 OCy 配置所需的输出和故障设置。
4. 通过将 OCTRIS 位清零将 OCx 的输出引脚强制为输出状态。
5. 如果需要触发模式操作，用 OCTRIG (OCxCON2<7>)、TRIGSTAT (OCxCON2<6>) 和 SYNCSEL (OCxCON2<4:0>) 位配置 OCx 中的触发选项。
6. 先为 OCy，再为 OCx 配置所需的比较或 PWM 工作模式 (OCM<2:0>)。

根据选择的输出模式，该模块将把 OCx 引脚保持在它的默认状态，并在 OCxR 和定时器匹配时强制转换到相反状态。在双比较模式下，OCx 在 OCxRS 与定时器值发生匹配时强制回到其默认状态。OCxIF 中断标志位在单比较模式下发生 OCxR 匹配后，以及双比较模式下的每次 OCxRS 匹配后，都会置 1。

单次触发脉冲事件只发生一次，但只需重写 OCxCON1 寄存器的值即可重复。连续脉冲事件会无限继续，直到被终止。

14.3 脉宽调制 (PWM) 模式

在 PWM 模式下，输出比较模块可配置为产生边沿对齐或中心对齐的脉冲波形。所有 PWM 操作都是双重缓冲的（缓冲寄存器在模块内部，未映射到 SFR 空间）。

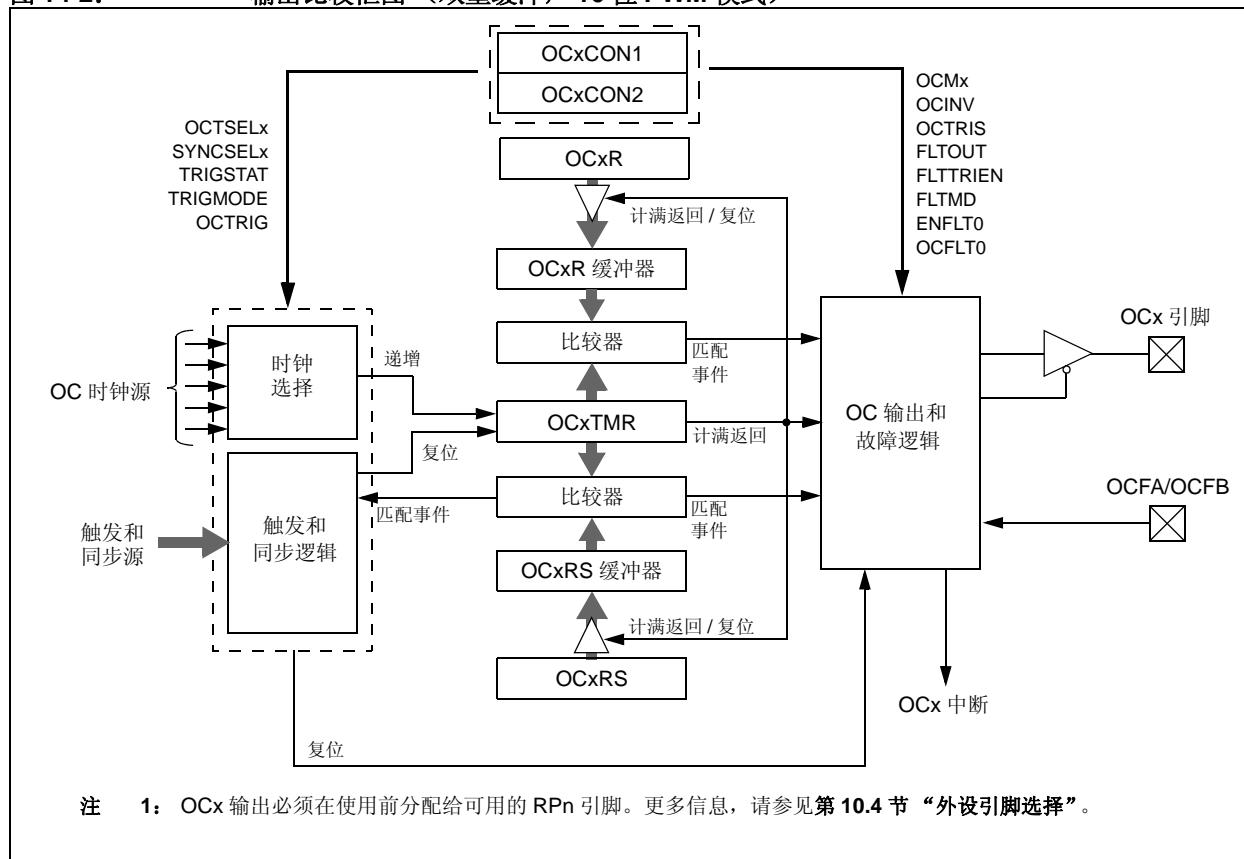
要将输出比较模块配置为 PWM 工作模式：

1. 为某个可用的外设引脚选择引脚配置 OCx 输出。
2. 计算所需的占空比，并将其装入 OCxR 寄存器。
3. 计算所需的周期，并将其装入 OCxRS 寄存器。
4. 通过将 0x1F 写入 SYNCSEL<4:0> (OCxCON2<4:0>) 并清零 OCTRIG (OCxCON2<7>) 选择当前 OCx 作为同步源。
5. 通过写入 OCTSEL<2:0> (OCxCON<12:10>) 位来选择时钟源。
6. 如果需要的话，允许定时器和输出比较模块的中断。如果要使用 PWM 故障引脚，则必须设置输出比较中断。
7. 在 OCM<2:0> (OCxCON1<2:0>) 位中选择所需的 PWM 模式。
8. 如果选择定时器作为时钟源，则应设置 TMRY 预分频值，并通过将 TON (TxCON<15>) 位置 1 使能时基。

注： 该外设包含可能需要由外设引脚选择配置的输入和输出功能。更多信息，请参见第 10.4 节“外设引脚选择”。

PIC24FJ256GB110 系列

图 14-2: 输出比较框图 (双重缓冲, 16 位 PWM 模式)



注 1: OCx 输出必须在使用前分配给可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

14.3.1 PWM 周期

PWM 周期可通过写入 PRy (定时器周期寄存器) 来指定。可使用公式 14-1 计算 PWM 周期。

公式 14-1: 计算 PWM 周期 (1)

$$\text{PWM 周期} = [(PRy) + 1] \cdot TCY \cdot (\text{定时器预分频值})$$

其中: PWM 频率 = $1/\text{[PWM 周期]}$

注 1: 基于 $TCY = TOSC * 2$, 打盹模式和 PLL 被禁止。

注: 如果 PRy 的值为 N, 则会使 PWM 周期为 $N + 1$ 个时基计数周期。例如, 如果写入 PRy 寄存器的值为 7, 则将产生包含 8 个时基周期的周期。

14.3.2 PWM 占空比

PWM 占空比是通过写 OCxRS 和 OCxR 寄存器指定的。可以在任何时候写 OCxRS 和 OCxR 寄存器, 但是在 PRy 和 TMry 发生匹配 (即周期结束) 前占空比值不会被锁存。这可以为 PWM 占空比提供双重缓冲, 对于 PWM 的无毛刺操作是极其重要的。

PWM 占空比有一些重要的边界参数, 包括:

- 如果 OCxR、OCxRS 和 PRy 中都装入 0000h, 则 OCx 引脚将保持低电平 (占空比为 0%)。
- 如果 OCxRS 大于 PRy, 则引脚将保持高电平 (占空比为 100%)。

请参见例 14-1 了解 PWM 模式时序的详细信息。表 14-1 和表 14-2 给出了器件分别以 4 MIPS 和 10 MIPS 工作时, 所对应的 PWM 频率和分辨率的示例。

公式 14-2: 计算最大 PWM 分辨率⁽¹⁾

$$\text{最大 PWM 分辨率(位)} = \frac{\log_{10}\left(\frac{FCY}{FPWM \cdot (\text{定时器预分频值})}\right)}{\log_{10}(2)} \text{ 位}$$

注 1: 基于 $FCY = Fosc/2$, 打盹模式和 PLL 被禁止。

例 14-1: PWM 周期和占空比计算⁽¹⁾

- 确定对应于所需的 PWM 频率 52.08 kHz 的定时器周期寄存器值, 其中 $FOSC = 8 \text{ MHz}$, PLL (32 MHz 器件时钟速率) 和 Timer2 预分频比设置为 1:1。

$$TCY = 2 * TOSC = 62.5 \text{ ns}$$

$$\text{PWM 周期} = 1/\text{PWM 频率} = 1/52.08 \text{ kHz} = 19.2 \mu\text{s}$$

$$\text{PWM 周期} = (PR2 + 1) * TCY * (\text{Timer 2 预分频值})$$

$$19.2 \mu\text{s} = (PR2 + 1) * 62.5 \text{ ns} * 1$$

$$PR2 = 306$$

- 在 PWM 频率为 52.08 kHz 且器件的时钟速率为 32 MHz 时, 计算占空比的最大分辨率:

$$\text{PWM 分辨率} = \log_{10}(FCY/FPWM)/\log_{10}2 \text{ 位}$$

$$= (\log_{10}(16 \text{ MHz}/52.08 \text{ kHz})/\log_{10}2) \text{ 位}$$

$$= 8.3 \text{ 位}$$

注 1: 基于 $TCY = 2 * TOSC$; 打盹模式和 PLL 被禁止。

表 14-1: 4 MIPS 时的 PWM 频率和分辨率示例 ($FCY = 4 \text{ MHz}$)⁽¹⁾

PWM 频率	7.6 Hz	61 Hz	122 Hz	977 Hz	3.9 kHz	31.3 kHz	125 kHz
定时器预分频比	8	1	1	1	1	1	1
周期寄存器值	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
分辨率(位)	16	16	15	12	10	7	5

注 1: 基于 $FCY = Fosc/2$, 打盹模式和 PLL 被禁止。

表 14-2: 16 MIPS 时的 PWM 频率和分辨率示例 ($FCY = 16 \text{ MHz}$)⁽¹⁾

PWM 频率	30.5 Hz	244 Hz	488 Hz	3.9 kHz	15.6 kHz	125 kHz	500 kHz
定时器预分频比	8	1	1	1	1	1	1
周期寄存器值	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
分辨率(位)	16	16	15	12	10	7	5

注 1: 基于 $FCY = Fosc/2$, 打盹模式和 PLL 被禁止。

PIC24FJ256GB110 系列

寄存器 14-1: OCxCON1: 输出比较 x 控制寄存器 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	—	—
bit 15							bit 8

R/W-0	U-0	U-0	R/W-0, HCS	R/W-0	R/W-0	R/W-0	R/W-0
ENFLT0	—	—	OCFLT0	TRIGMODE	OCM2 ⁽¹⁾	OCM1 ⁽¹⁾	OCM0 ⁽¹⁾
bit 7							bit 0

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

bit 15-14	未实现: 读为 0
bit 13	OCSIDL: 在空闲模式下停止输出比较 x 的控制位 1 = 输出比较 x 在 CPU 空闲模式下停止工作 0 = 输出比较 x 在 CPU 空闲模式下继续工作
bit 12-10	OCTSEL<2:0>: 输出比较 x 定时器选择位 111 = 系统时钟 110 = 保留 101 = 保留 100 = Timer1 011 = Timer5 010 = Timer4 001 = Timer3 000 = Timer2
bit 9-8	未实现: 读为 0
bit 7	ENFLT0: 故障 0 输入使能位 1 = 使能故障 0 输入 0 = 禁止故障 0 输入
bit 6-5	未实现: 读为 0
bit 4	OCFLT0: PWM 故障条件状态位 1 = 发生了 PWM 故障条件 (仅可用硬件清零) 0 = 未发生 PWM 故障条件 (仅当 OCM<2:0> = 111 时, 才使用该位)
bit 3	TRIGMODE: 触发状态模式选择位 1 = 当 OCxRS = OCxTMR 时, TRIGSTAT (OCxCON2<6>) 被清零, 或用软件清零 0 = TRIGSTAT 只能用软件清零
bit 2-0	OCM<2:0>: 输出比较 x 模式选择位 ⁽¹⁾ 111 = OCx 输出中心对齐的 PWM 信号 ⁽²⁾ 110 = OCx 输出边沿对齐的 PWM 信号 ⁽²⁾ 101 = 双比较连续脉冲模式: 初始化 OCx 引脚为低电平, 在连续周期内 OCxR 和 OCxRS 匹配时翻转 OCx 状态 100 = 双比较单次触发模式: 初始化 OCx 引脚为低电平, 在一个周期内 OCxR 和 OCxRS 匹配时翻转 OCx 状态 011 = 单比较连续脉冲模式: 比较事件使 OCx 引脚的电平连续翻转 010 = 单比较单次触发模式: 初始化 OCx 引脚为高电平, 比较事件强制 OCx 引脚为低电平 001 = 单比较单次触发模式: 初始化 OCx 引脚为低电平, 比较事件强制 OCx 引脚为高电平 000 = 输出比较通道禁止

- 注 1: OCx 输出必须也配置给可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。
2: OCFA 引脚控制 OC1-OC4 通道; OCFB 引脚控制 OC5-OC9 通道。OCxR 和 OCxRS 仅在 PWM 模式下是双重缓冲的。

寄存器 14-2: OCxCON2: 输出比较 x 控制寄存器 2

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
FLTMD	FLTO	FLTTRIEN	OCINV	—	—	—	OC32
bit 15	bit 8						

R/W-0	R/W-0 HS	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7	bit 0						

图注: HS = 硬件置 1 位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- | | |
|----------|---|
| bit 15 | FLTMD: 故障模式选择位
1 = 故障模式将保持到故障源消除, 并且对应的 OCFLT0 位用软件清零
0 = 故障模式将保持到故障源消除, 并且新的 PWM 周期开始 |
| bit 14 | FLTO: 故障结果位
1 = 发生故障时 PWM 输出驱动为高电平
0 = 发生故障时 PWM 输出驱动为低电平 |
| bit 13 | FLTTRIEN: 故障输出状态选择位
1 = 发生故障时引脚强制为输出
0 = 引脚 I/O 状况不受故障影响 |
| bit 12 | OCINV: OCMP 反相位
1 = OCx 输出反相
0 = OCx 输出不反相 |
| bit 11-9 | 未实现: 读为 0 |
| bit 8 | OC32: 级联两个 OC 模块使能位 (32 位操作)
1 = 使能级联模块工作
0 = 禁止级联模块工作 |
| bit 7 | OCTRIG: OCx 触发 / 同步选择位
1 = 用 SYNCSELx 位指定的源触发 OCx
0 = 用 SYNCSELx 位指定的源同步 OCx |
| bit 6 | TRIGSTAT: 定时器触发状态位
1 = 定时器源已触发并正在运行
0 = 定时器源未触发, 保持清零 |
| bit 5 | OCTRIS: OCx 输出引脚方向选择位
1 = OCx 引脚呈现三态
0 = 输出比较外设 x 与 OCx 引脚相连 |

注 1: 绝不要通过选择该模式或其他相当的 SYNCSEL 设置将 OC 模块用作它自己的触发源。

2: 只用这些输入作为触发源, 从不用作同步源。

PIC24FJ256GB110 系列

寄存器 14-2: OCxCON2: 输出比较 x 控制寄存器 2 (续)

bit 4-0 **SYNCSEL<4:0>**: 触发 / 同步源选择位

11111 = 该 OC 模块⁽¹⁾

11110 = 输入捕捉 9⁽²⁾

11101 = 输入捕捉 6⁽²⁾

11100 = CTMU⁽²⁾

11011 = A/D⁽²⁾

11010 = 比较器 3⁽²⁾

11001 = 比较器 2⁽²⁾

11000 = 比较器 1⁽²⁾

10111 = 输入捕捉 4⁽²⁾

10110 = 输入捕捉 3⁽²⁾

10101 = 输入捕捉 2⁽²⁾

10100 = 输入捕捉 1⁽²⁾

10011 = 输入捕捉 8⁽²⁾

10010 = 输入捕捉 7⁽²⁾

1000x = 保留

01111 = Timer 5

01110 = Timer 4

01101 = Timer 3

01100 = Timer 2

01011 = Timer 1

01010 = 输入捕捉 5⁽²⁾

01001 = 输出比较 9⁽¹⁾

01000 = 输出比较 8⁽¹⁾

00111 = 输出比较 7⁽¹⁾

00110 = 输出比较 6⁽¹⁾

00101 = 输出比较 5⁽¹⁾

00100 = 输出比较 4⁽¹⁾

00011 = 输出比较 3⁽¹⁾

00010 = 输出比较 2⁽¹⁾

00001 = 输出比较 1⁽¹⁾

00000 = 不与任何其他模块同步

注 1: 绝不要通过选择该模式或其他相当的 SYNCSEL 设置将 OC 模块用作它自己的触发源。

2: 只用这些输入作为触发源, 从不用作同步源。

15.0 串行外设接口（SPI）

注：本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》的第 23 章“串行外设接口（SPI）”（DS39699A_CN）。

串行外设接口（Serial Peripheral Interface，SPI）模块是用于同其他外设或单片机器件进行通信的同步串行接口。这些外设器件可以是串行 EEPROM、移位寄存器、显示驱动器和 A/D 转换器等。SPI 模块与 Motorola 的 SPI 和 SIOP 接口兼容。PIC24FJ256GB110 系列的所有器件都包含 3 个 SPI 模块。

该模块支持在两种缓冲模式下工作。在标准模式下，数据通过单个串行缓冲器移动。在增强型缓冲模式下，数据通过一个 8 级深 FIFO 缓冲器移动。

注：在标准或增强型缓冲模式下，都不要在 SPIxBUF 寄存器上执行读 - 修改 - 写操作（如针对位的指令）。

工作于主或从模式时，模块还支持基本的帧 SPI 协议。共支持 4 种帧 SPI 配置。

SPI 串行接口由 4 个引脚组成：

- **SDIx：** 串行数据输入
- **SDOx：** 串行数据输出
- **SCKx：** 移位时钟输入或输出
- **SSx：** 低电平有效从选择或帧同步 I/O 脉冲

SPI 模块可以配置为使用 2、3 或 4 个引脚工作。在 3 引脚模式下，SSx 不使用。在 2 引脚模式下，SDOx 和 SSx 均不使用。

标准和增强型模式下的模块框图如图 15-1 和图 15-2 所示。

注：在本章中，SPI 模块统称为 SPIx，或分别称为 SPI1、SPI2 或 SPI3。特殊功能寄存器也使用类似的符号表示。例如，SPIxCON1 和 SPIxCON2 指 3 个 SPI 模块中任何一个的控制寄存器。

PIC24FJ256GB110 系列

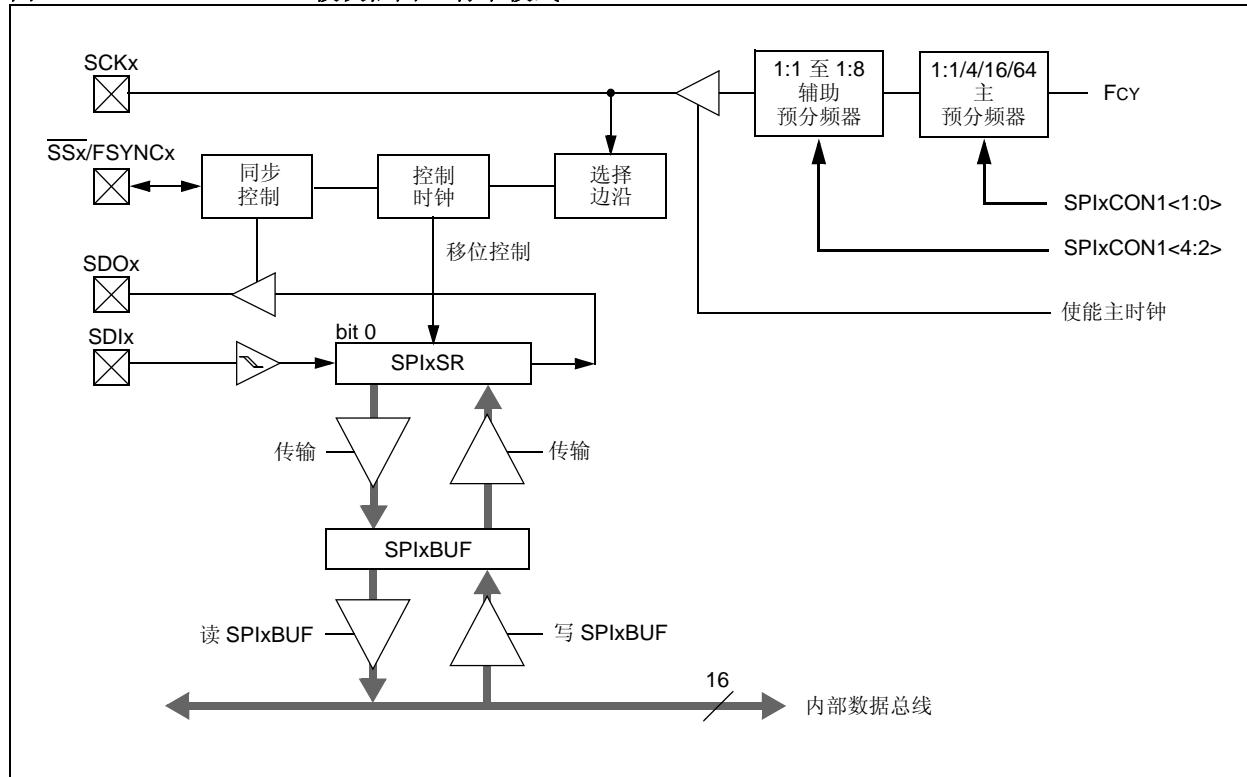
要将 SPI 模块设置为工作于标准主模式：

1. 如果使用中断：
 - a) 将相应 IFS 寄存器中的 SPIxIF 位清零。
 - b) 将相应 IEC 寄存器中的 SPIxIE 位置 1。
 - c) 通过写相应 IPC 寄存器中的 SPIxIP 位来设置中断优先级。
2. 将所需的设置写入 SPIxCON1 和 SPIxCON2 寄存器，且 MSTEN (SPIxCON1<5>) = 1。
3. 将 SPIROV 位 (SPIxSTAT<6>) 清零。
4. 通过将 SPIEN 位 (SPIxSTAT<15>) 置 1 使能 SPI 工作。
5. 将待发送数据写入 SPIxBUF 寄存器。发送 (和接收) 在数据写入 SPIxBUF 寄存器时立即开始。

要将 SPI 模块设置为工作于标准从模式：

1. 将 SPIxBUF 寄存器清零。
2. 如果使用中断：
 - a) 将相应 IFS 寄存器中的 SPIxIF 位清零。
 - b) 将相应 IEC 寄存器中的 SPIxIE 位置 1。
 - c) 通过写相应 IPC 寄存器中的 SPIxIP 位来设置中断优先级。
3. 将所需的设置写入 SPIxCON1 和 SPIxCON2 寄存器，且 MSTEN (SPIxCON1<5>) = 0。
4. 将 SMP 位清零。
5. 如果 CKE 位 (SPIxCON1<8>) 置 1，则必须将 SSEN 位 (SPIxCON1<7>) 置 1 来使能 SS_x 引脚。
6. 将 SPIROV 位 (SPIxSTAT<6>) 清零。
7. 通过将 SPIEN 位 (SPIxSTAT<15>) 置 1 使能 SPI 工作。

图 15-1： SPIx 模块框图（标准模式）



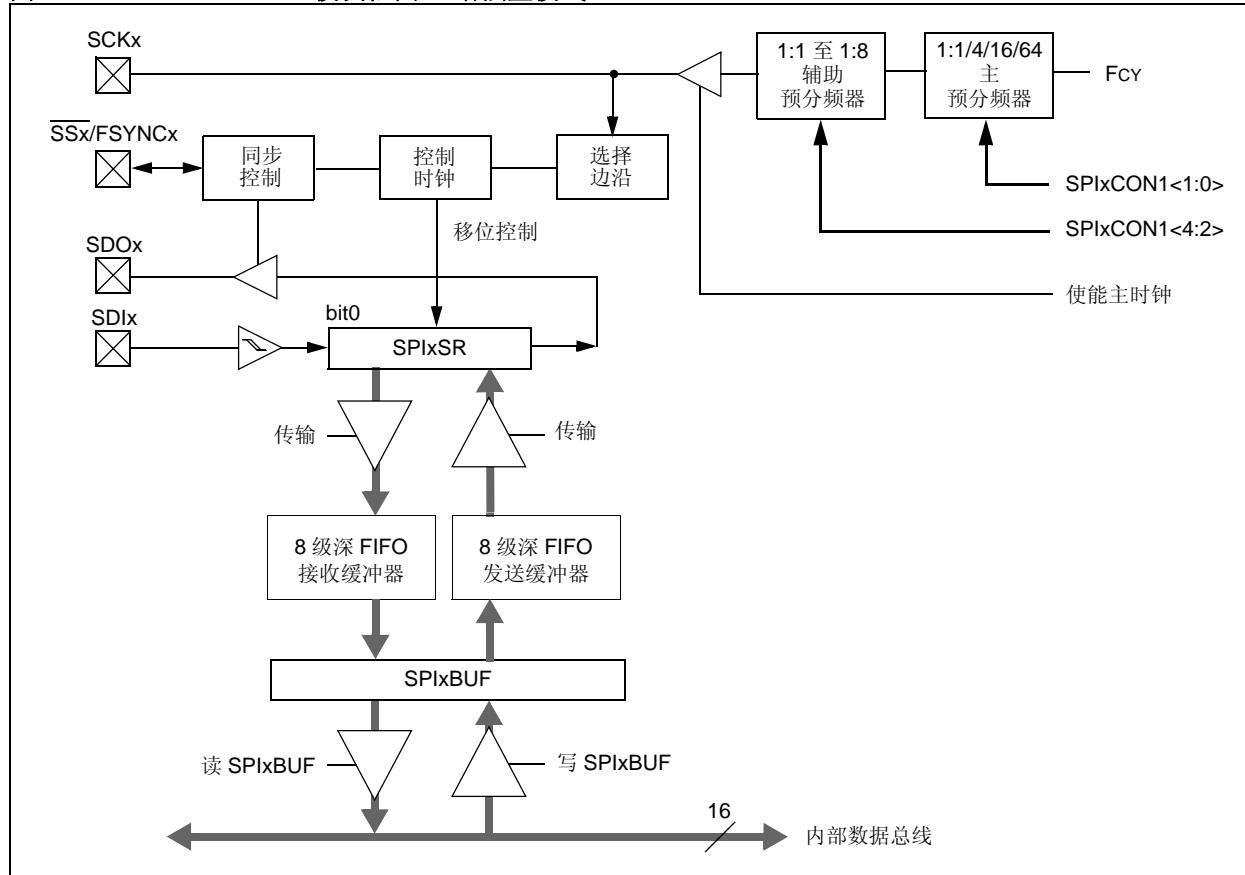
要将 SPI 模块设置为工作于增强型缓冲主模式:

1. 如果使用中断:
 - a) 将相应 IFS 寄存器中的 SPIxIF 位清零。
 - b) 将相应 IEC 寄存器中的 SPIxIE 位置 1。
 - c) 写入相应 IPC 寄存器中的 SPIxIP 位。
2. 将所需的设置写入 SPIxCON1 和 SPIxCON2 寄存器, 且 MSTEN (SPIxCON1<5>) = 1。
3. 将 SPIROV 位 (SPIxSTAT<6>) 清零。
4. 通过将 SPIBEN 位 (SPIxCON2<0>) 置 1 选择增强型缓冲模式。
5. 通过将 SPIEN 位 (SPIxSTAT<15>) 置 1 使能 SPI 工作。
6. 将待发送数据写入 SPIxBUF 寄存器。发送 (和接收) 在数据写入 SPIxBUF 寄存器时立即开始。

要将 SPI 模块设置为工作于增强型缓冲从模式:

1. 将 SPIxBUF 寄存器清零。
2. 如果使用中断:
 - a) 将相应 IFS 寄存器中的 SPIxIF 位清零。
 - b) 将相应 IEC 寄存器中的 SPIxIE 位置 1。
 - c) 通过写相应 IPC 寄存器中的 SPIxIP 位来设置中断优先级。
3. 将所需的设置写入 SPIxCON1 和 SPIxCON2 寄存器, 且 MSTEN (SPIxCON1<5>) = 0。
4. 将 SMP 位清零。
5. 如果 CKE 位置 1, 则必须将 SSEN 位置 1 来使能 SS_x 引脚。
6. 将 SPIROV 位 (SPIxSTAT<6>) 清零。
7. 通过将 SPIBEN 位 (SPIxCON2<0>) 置 1 选择增强型缓冲模式。
8. 通过将 SPIEN 位 (SPIxSTAT<15>) 置 1 使能 SPI 工作。

图 15-2: SPIx 模块框图 (增强型模式)



PIC24FJ256GB110 系列

寄存器 15-1: SPIxSTAT: SPIx 状态和控制寄存器

R/W-0	U-0	R/W-0	U-0	U-0	R-0	R-0	R-0
SPIEN ⁽¹⁾	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0
bit 15	bit 8						

R-0	R/C-0 HS	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0
SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF
bit 7	bit 0						

图注:	C = 可清零位	HS = 只可硬件置 1 的位
R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = POR 时的值	1 = 置 1	0 = 清零

- bit 15 **SPIEN:** SPIx 使能位⁽¹⁾
1 = 使能模块并将 SCKx、SDOx、SDIx 和 SSx 配置为串口引脚
0 = 禁止模块
- bit 14 未实现: 读为 0
- bit 13 **SPISIDL:** 空闲模式停止位
1 = 当器件进入空闲模式时, 模块停止工作
0 = 在空闲模式下模块继续工作
- bit 12-11 未实现: 读为 0
- bit 10-8 **SPIBEC<2:0>:** SPIx 缓冲器计数位 (在增强型缓冲模式下有效)
主模式:
SPI 传输等待的数目。
从模式:
SPI 传输未读的数目。
- bit 7 **SRMPT:** 移位寄存器 (SPIxSR) 空位 (在增强型缓冲模式下有效)
1 = SPIx 移位寄存器为空, 等待发送或接收
0 = SPIx 移位寄存器非空
- bit 6 **SPIROV:** 接收溢出标志位
1 = 一个新字节 / 字已完全接收并丢弃。用户软件尚未读取 SPIxBUF 寄存器中的先前数据。
0 = 未发生溢出
- bit 5 **SRXMPT:** 接收 FIFO 空位 (在增强型缓冲模式下有效)
1 = 接收 FIFO 为空
0 = 接收 FIFO 非空
- bit 4-2 **SISEL<2:0>:** SPIx 缓冲器中断模式位 (在增强型缓冲模式下有效)
111 = 当 SPIx 发送缓冲器已满时产生中断 (SPITBF 位置 1)
110 = 当最后一位移入 SPIxSR, 从而导致发送 FIFO 为空时产生中断
101 = 当最后一位移出 SPIxSR 时产生中断, 此时发送完成
100 = 当一个数据移入 SPIxSR, 从而发送 FIFO 至少有一个空位时产生中断
011 = 当 SPIx 接收缓冲器已满时产生中断 (SPIRBF 位置 1)
010 = 当 SPIx 接收缓冲器为 3/4 满或更满时产生中断
001 = 当接收缓冲器中有数据时产生中断 (SRMPT 位置 1)
000 = 当接收缓冲器中的最后数据被读取, 从而导致缓冲器为空时产生中断 (SRXMPT 位置 1)

注 1: 如果 SPIEN = 1, 这些功能必须在使用前分配给可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

寄存器 15-1: SPIxSTAT: SPIx 状态和控制寄存器 (续)

bit 1

SPITB_F: SPI_x 发送缓冲器满状态位

1 = 发送尚未开始, SPI_xTXB 已满
0 = 发送开始, SPI_xTXB 为空

在标准缓冲模式下:

当 CPU 通过写 SPI_xBUF 存储单元装入 SPI_xTXB 时, 该位由硬件自动置 1。
当 SPI_x 模块将数据从 SPI_xTXB 传送到 SPI_xSR 时, 该位由硬件自动清零。

在增强型缓冲模式下:

当 CPU 通过写 SPI_xBUF 存储单元装入最后的可用缓冲单元时, 该位由硬件自动置 1。
当有缓冲单元可用于 CPU 写操作时, 该位由硬件自动清零。

bit 0

SPIRB_F: SPI_x 接收缓冲器满状态位

1 = 接收完成, SPI_xRXB 已满
0 = 接收未完成, SPI_xRXB 为空

在标准缓冲模式下:

当 SPI_x 将数据从 SPI_xSR 传送到 SPI_xRXB 时, 该位由硬件自动置 1。
当内核通过读 SPI_xBUF 存储单元读 SPI_xRXB 时, 该位由硬件自动清零。

在增强型缓冲模式下:

当 SPI_x 通过将数据从 SPI_xSR 传送到缓冲器填充最后未读的缓冲单元时, 该位由硬件自动置 1。
当有缓冲单元可用于从 SPI_xSR 进行传送时, 该位由硬件自动清零。

注 1: 如果 SPIEN = 1, 这些功能必须在使用前分配给可用的 RP_n 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

PIC24FJ256GB110 系列

寄存器 15-2: SPIxCON1: SPIx 控制寄存器 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DISSCK ⁽¹⁾	DISSDO ⁽²⁾	MODE16	SMP	CKE ⁽³⁾
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSEN ⁽⁴⁾	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-13 未实现: 读为 0

bit 12 **DISSCK:** 禁止 SCKx 引脚位 (仅限 SPI 主模式) ⁽¹⁾

1 = 禁止内部 SPI 时钟; 引脚用作 I/O

0 = 使能内部 SPI 时钟

bit 11 **DISSDO:** 禁止 SDOx 引脚位 ⁽²⁾

1 = 模块不使用 SDOx 引脚; 引脚用作 I/O

0 = SDOx 引脚由模块控制

bit 10 **MODE16:** 字 / 字节通信选择位

1 = 采用字宽 (16 位) 通信

0 = 采用字节宽 (8 位) 通信

bit 9 **SMP:** SPIx 数据输入采样阶段位

主模式:

1 = 在数据输出时间的末端采样输入数据

0 = 在数据输出时间的中间采样输入数据

从模式:

当在从模式下使用 SPIx 时, 必须将 SMP 清零。

bit 8 **CKE:** SPIx 时钟边沿选择位 ⁽³⁾

1 = 串行输出数据在时钟从工作状态转变为空闲状态时变化 (见 bit 6)

0 = 串行输出数据在时钟从空闲状态转变为工作状态时变化 (见 bit 6)

bit 7 **SSEN:** 从选择使能 (从模式) 位 ⁽⁴⁾

1 = SSx 引脚用于从模式

0 = 模块不使用 SSx 引脚; 引脚由端口功能控制

bit 6 **CKP:** 时钟极性选择位

1 = 空闲状态时时钟信号为高电平; 工作状态时为低电平

0 = 空闲状态时时钟信号为低电平; 工作状态时为高电平

bit 5 **MSTEN:** 主模式使能位

1 = 主模式

0 = 从模式

注 1: 如果 DISSCK = 0, SCKx 必须配置为可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

2: 如果 DISSDO = 0, SDOx 必须配置为可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

3: 在帧 SPI 模式下不使用 CKE 位。在帧 SPI 模式 (FRMEN = 1) 下, 用户应将该位编程为 0。

4: 如果 SSEN = 1, SSx 必须配置为可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

寄存器 15-2: SPIxCON1: SPIx 控制寄存器 1 (续)

bit 4-2 **SPRE<2:0>**: 辅助预分频比位 (主模式)

111 = 辅助预分频比 1:1

110 = 辅助预分频比 2:1

...

000 = 辅助预分频比 8:1

bit 1-0 **PPRE<1:0>**: 主预分频比位 (主模式)

11 = 主预分频比 1:1

10 = 主预分频比 4:1

01 = 主预分频比 16:1

00 = 主预分频比 64:1

注 1: 如果 DISSCK = 0, SCKx 必须配置为可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

2: 如果 DISSDO = 0, SDOx 必须配置为可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

3: 在帧 SPI 模式下不使用 CKE 位。在帧 SPI 模式 (FRMEN = 1) 下, 用户应将该位编程为 0。

4: 如果 SSEN = 1, SSx 必须配置为可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

寄存器 15-3: SPIxCON2: SPIx 控制寄存器 2

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—
bit 15	bit 8						

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SPIFE	SPIBEN
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **FRMEN**: 帧 SPIx 支持位

1 = 使能帧 SPIx 支持

0 = 禁止帧 SPIx 支持

bit 14 **SPIFSD**: SSx 引脚帧同步脉冲方向控制位

1 = 帧同步脉冲输入 (从模式)

0 = 帧同步脉冲输出 (主模式)

bit 13 **SPIFPOL**: 帧同步脉冲极性位 (仅限帧模式)

1 = 帧同步脉冲高电平有效

0 = 帧同步脉冲低电平有效

bit 12-2 未实现: 读为 0

bit 1 **SPIFE**: 帧同步脉冲边沿选择位

1 = 帧同步脉冲与第一个位时钟同时发生

0 = 帧同步脉冲先于第一个位时钟

bit 0 **SPIBEN**: 增强型缓冲器使能位

1 = 使能增强型缓冲器

0 = 禁止增强型缓冲器 (传统模式)

PIC24FJ256GB110 系列

图 15-3: SPI 主 / 从连接 (标准模式)

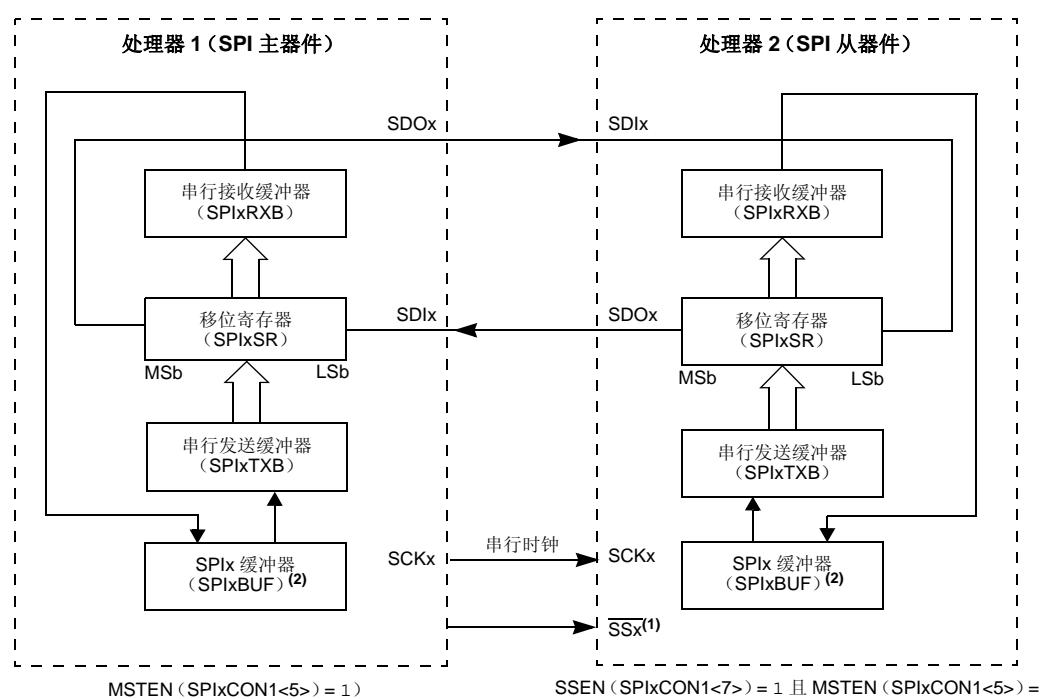


图 15-4: SPI 主 / 从连接 (增强型缓冲模式)

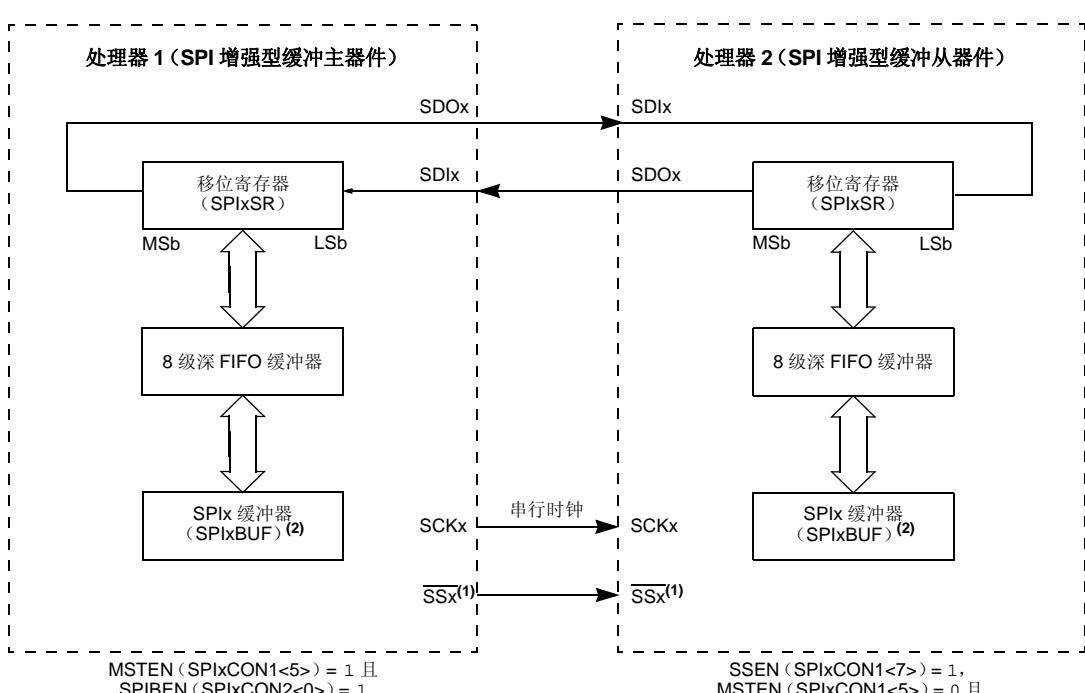


图 15-5: SPI 主 - 帧主模式连接图

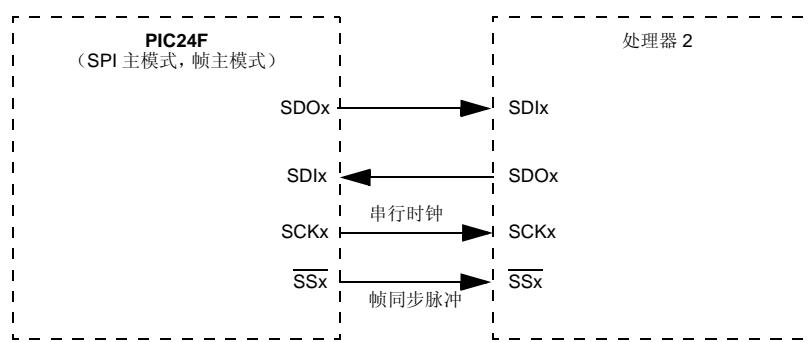


图 15-6: SPI 主 - 帧从模式连接图

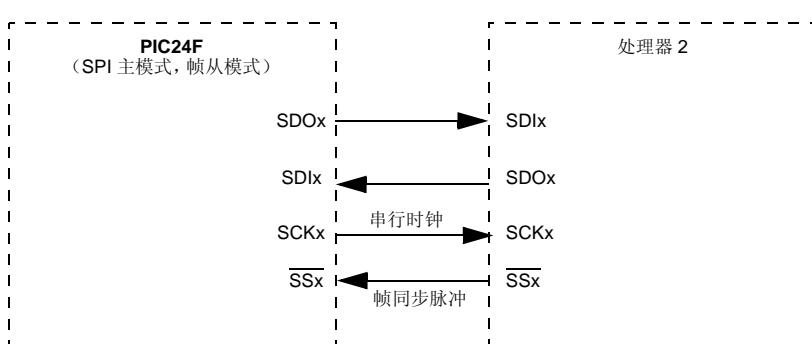


图 15-7: SPI 从 - 帧主模式连接图

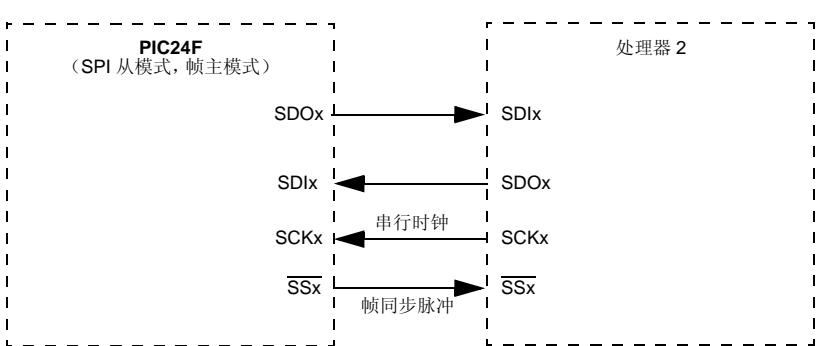
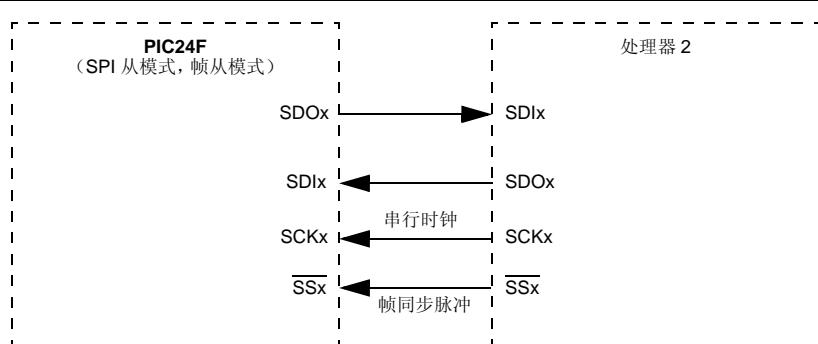


图 15-8: SPI 从 - 帧从模式连接图



PIC24FJ256GB110 系列

公式 15-1： 器件工作频率和 SPI 时钟速度之间的关系⁽¹⁾

$$F_{SCK} = \frac{F_{CY}}{\text{主预分频比} * \text{辅助预分频比}}$$

注 1： 基于 $F_{CY} = F_{OSC}/2$, 打盹模式和 PLL 被禁止。

表 15-1： 示例 SCK 频率^(1,2)

FCY = 16 MHz		辅助预分频比设置				
		1:1	2:1	4:1	6:1	8:1
主预分频比设置	1:1	无效	8000	4000	2667	2000
	4:1	4000	2000	1000	667	500
	16:1	1000	500	250	167	125
	64:1	250	125	63	42	31
FCY = 5 MHz						
主预分频比设置	1:1	5000	2500	1250	833	625
	4:1	1250	625	313	208	156
	16:1	313	156	78	52	39
	64:1	78	39	20	13	10

注 1： 基于 $F_{CY} = F_{OSC}/2$, 打盹模式和 PLL 被禁止。

2： 表中 SCKx 频率的单位为 kHz。

16.0 I²CTM

注: 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见《PIC24F 系列参考手册》的第 24 章 “I²C” (DS39702A_CN)。

I²C 模块是用于同其他外设或单片机器件进行通信的串行接口。这些外设可以是串行 EEPROM、显示驱动器、A/D 转换器等。

I²C 模块支持以下特性:

- 独立的主 / 从逻辑
- 7 位和 10 位器件地址
- I²C 协议中所定义的广播呼叫地址
- 时钟延长功能, 为处理器响应从器件数据请求提供延时
- 100 kHz 和 400 kHz 总线规范
- 可配置的地址掩码
- 多主机模式以防仲裁时报文丢失
- 总线转发器模式, 允许作为从器件接收所有消息, 与地址无关
- 自动 SCL

图 16-1 给出了此模块的框图。

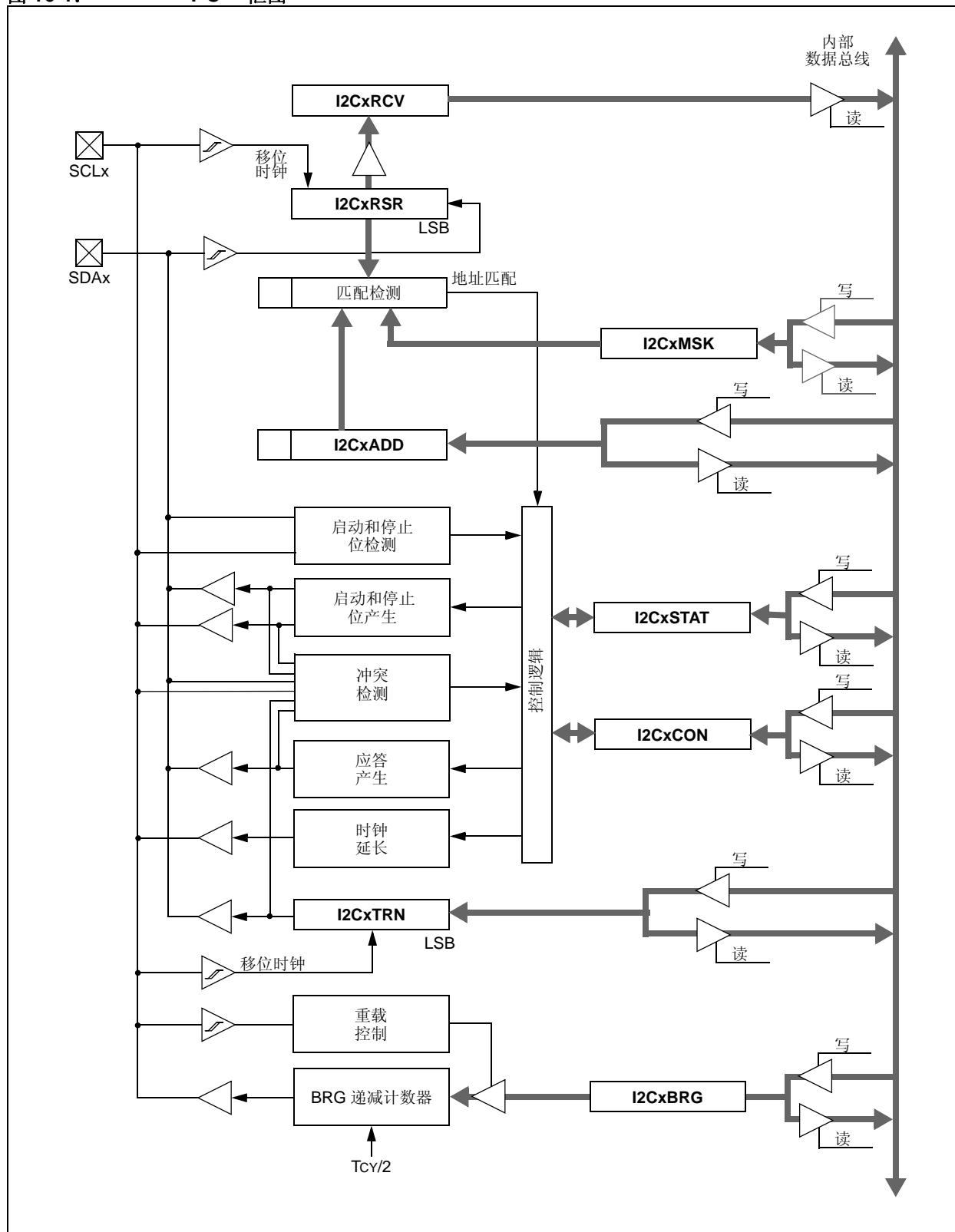
16.1 在单主机环境下作为主器件进行通信

在主模式下发送报文的详细信息取决于用于与器件通信的通信协议。通常, 发送报文的步骤如下:

1. 在 SDAx 和 SCLx 上发出一个启动条件。
2. 向从器件发送 I²C 器件地址字节和写操作指示。
3. 等待并验证来自从器件的应答。
4. 向从器件发送第一个数据字节 (有时称为命令)。
5. 等待并验证来自从器件的应答。
6. 向从器件发送串行存储器地址低字节。
7. 重复步骤 4 和 5, 直到发送完所有数据字节。
8. 在 SDAx 和 SCLx 上发出一个重复启动条件。
9. 向从器件发送器件地址字节和读操作指示。
10. 等待并验证来自从器件的应答。
11. 使能主器件接收, 以接收串行存储器数据。
12. 在数据字节接收结束时产生 ACK 或 NACK 条件。
13. 在 SDAx 和 SCLx 上产生一个停止条件。

PIC24FJ256GB110 系列

图 16-1: I²CTM 框图



16.2 设置作为总线主器件工作时的波特率

要计算波特率发生器的重载值，可使用公式 16-1。

公式 16-1: 计算波特率重载值 (1,2)

$$Fscl = \frac{FcY}{I2CxBRG + 1 + \frac{FcY}{10,000,000}}$$

或

$$I2CxBRG = \left(\frac{FcY}{Fscl} - \frac{FcY}{10,000,000} \right) - 1$$

注 1: 基于 $FcY = Fosc/2$ ；打盹模式和 PLL 被禁止。

2: 这些时钟速率值仅供参考。实际的时钟速率会受到各种系统级参数影响。应在目标应用中测量实际时钟速率。

16.3 从地址掩码

I2CxMSK 寄存器（寄存器 16-3）将 7 位和 10 位寻址模式下地址中的某些位指定为“无关位”。将 I2CxMSK 寄存器中某个特定位置 1 (=1)，不论相应的地址位的值是 0 还是 1，工作在从模式下的模块都会作出响应。例如，当将 I2CxMSK 设置为 00100000 时，工作在从模式下的模块将检测两个地址 0000000 和 0100000。

要使能地址掩码，必须通过将 **IPMIEN** 位 (**I2CxCON<11>**) 清零来禁止智能外设管理接口 (Intelligent Peripheral Management Interface, IPMI)。

注: 新修改的 I²CTM 协议使得表 16-2 中的地址保留，而且在从模式下不会应答。这包括包含任何这些地址的任何地址掩码设置。

表 16-1: I²CTM 时钟速率 (1,2)

必需的系统 Fscl	FcY	I2CxBRG 值		实际 Fscl
		(十进制)	(十六进制)	
100 kHz	16 MHz	157	9D	100 kHz
100 kHz	8 MHz	78	4E	100 kHz
100 kHz	4 MHz	39	27	99 kHz
400 kHz	16 MHz	37	25	404 kHz
400 kHz	8 MHz	18	12	404 kHz
400 kHz	4 MHz	9	9	385 kHz
400 kHz	2 MHz	4	4	385 kHz
1 MHz	16 MHz	13	D	1.026 MHz
1 MHz	8 MHz	6	6	1.026 MHz
1 MHz	4 MHz	3	3	0.909 MHz

注 1: 基于 $FcY = Fosc/2$ ，打盹模式和 PLL 被禁止。

2: 这些时钟速率值仅供参考。实际的时钟速率会受到各种系统级参数影响。应在目标应用中测量实际时钟速率。

表 16-2: I²CTM 保留的地址 (1)

从器件地址	R/W 位	说明
0000 000	0	广播呼叫地址 (2)
0000 000	1	启动字节
0000 001	x	Cbus 地址
0000 010	x	保留
0000 011	x	保留
0000 1xx	x	HS 模式主机码
1111 1xx	x	保留
1111 0xx	x	10 位从地址高字节 (3)

注 1: 这里所列的地址位将永远不会导致地址匹配，与地址掩码设置无关。

2: 只有 $GCEN = 1$ 时才会应答地址。

3: 与这些地址的匹配只会发生在 10 位寻址模式下的高字节中。

PIC24FJ256GB110 系列

寄存器 16-1: I2CxCON: I2Cx 控制寄存器

R/W-0	U-0	R/W-0	R/W-1, HC	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	R/W-0, HC				
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7	bit 0						

图注:

HC = 硬件清零位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15

I2CEN: I2Cx 使能位

1 = 使能 I2Cx 模块, 并将 SDAx 和 SCLx 引脚配置为串口引脚
0 = 禁止 I2Cx 模块。所有 I²C 引脚由端口功能控制。

bit 14

未实现: 读为 0

bit 13

I2CSIDL: 空闲模式停止位

1 = 当器件进入空闲模式时, 模块停止工作
0 = 在空闲模式下模块继续工作

bit 12

SCLREL: SCLx 释放控制位 (作为 I²C 从器件工作时)

1 = 释放 SCLx 时钟
0 = 保持 SCLx 时钟为低电平 (时钟延长)

如果 STREN = 1:

该位可读可写 (即软件可以写入 0 来启动时钟延长或写入 1 来释放时钟)。
在从器件发送开始时由硬件清零。
在从器件接收结束时由硬件清零。

如果 STREN = 0:

该位可读且可被置 1 (即软件只能写入 1 来释放时钟)。
在从器件发送开始时由硬件清零。

bit 11

IPMIEN: 智能外设管理接口 (IPMI) 使能位

1 = 使能 IPMI 支持模式; 应答所有地址
0 = 禁止 IPMI 模式

bit 10

A10M: 10 位从器件寻址位

1 = I2CxADD 为 10 位从器件地址
0 = I2CxADD 为 7 位从器件地址

bit 9

DISSLW: 禁止斜率控制位

1 = 禁止斜率控制
0 = 使能斜率控制

bit 8

SMEN: SMBus 输入电平位

1 = 使能符合 SMBus 规范的 I/O 引脚限值
0 = 禁止 SMBus 输入限值

bit 7

GCEN: 广播呼叫使能位 (作为 I²C 从器件工作时)

1 = 允许在 I2CxRSR 接收到广播呼叫地址时产生中断 (已使能模块接收)
0 = 禁止广播呼叫地址

bit 6

STREN: SCLx 时钟延长使能位 (作为 I²C 从器件工作时)

与 SCLREL 位配合使用。
1 = 使能软件或接收时钟延长
0 = 禁止软件或接收时钟延长

寄存器 16-1: I2CxCON: I2Cx 控制寄存器 (续)

bit 5	ACKDT: 应答数据位 (作为 I ² C 主器件工作时。主器件接收期间适用。) 当软件启动应答序列时将发送的值。 1 = 在应答时发送 NACK 0 = 在应答时发送 ACK
bit 4	ACKEN: 应答序列使能位 (作为 I ² C 主器件工作时。主器件接收期间适用。) 1 = 在 SDAx 和 SCLx 引脚上发出应答序列，并发送 ACKDT 数据位。在主器件应答序列结束时由硬件清零。 0 = 应答序列不在进行中
bit 3	RCEN: 接收使能位 (作为 I ² C 主器件工作时) 1 = 使能 I ² C 接收模式。在主器件接收完数据字节的第 8 位时由硬件清零。 0 = 接收序列不在进行中
bit 2	PEN: 停止条件使能位 (作为 I ² C 主器件工作时) 1 = 在 SDAx 和 SCLx 引脚上发出停止条件。在主器件停止序列结束时由硬件清零。 0 = 停止条件不在进行中
bit 1	RSEN: 重复启动条件使能位 (作为 I ² C 主器件工作时) 1 = 在 SDAx 和 SCLx 引脚上发出重复启动条件。在主器件重复启动序列结束时由硬件清零。 0 = 重复启动条件不在进行中
bit 0	SEN: 启动条件使能位 (作为 I ² C 主器件工作时) 1 = 在 SDAx 和 SCLx 引脚上发出启动条件。在主器件启动序列结束时由硬件清零。 0 = 启动条件不在进行中

PIC24FJ256GB110 系列

寄存器 16-2: I2CxSTAT: I2Cx 状态寄存器

R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HS	R-0, HSC	R-0, HSC
ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
bit 15							bit 8

R/C-0, HS	R/C-0, HS	R-0, HSC	R/C-0, HSC	R/C-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IWCOL	I2COV	D/A	P	S	R/W	RBF	TBF
bit 7							bit 0

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

- bit 15 **ACKSTAT:** 应答状态位
1 = 最后检测到 NACK
0 = 最后检测到 ACK
在应答结束时由硬件置 1 或清零。
- bit 14 **TRSTAT:** 发送状态位 (作为 I²C 主器件工作时。适用于主器件发送操作。)
1 = 主器件正在进行发送 (8 位 + ACK)
0 = 主器件不在进行发送
在主器件发送开始时由硬件置 1。在从器件应答结束时由硬件清零。
- bit 13-11 **未实现:** 读为 0
- bit 10 **BCL:** 主器件总线冲突检测位
1 = 主器件工作期间检测到了总线冲突
0 = 未发生冲突
检测到总线冲突时由硬件置 1。
- bit 9 **GCSTAT:** 广播呼叫状态位
1 = 接收到广播呼叫地址
0 = 未接收到广播呼叫地址
当地址与广播呼叫地址匹配时由硬件置 1。当检测到停止条件时由硬件清零。
- bit 8 **ADD10:** 10 位地址状态位
1 = 10 位地址匹配
0 = 10 位地址不匹配
当与匹配的 10 位地址的第 2 个字节匹配时由硬件置 1。当检测到停止条件时由硬件清零。
- bit 7 **IWCOL:** 写冲突检测位
1 = 因为 I²C 模块忙, 尝试写 I2CxTRN 寄存器失败
0 = 未发生冲突
当总线忙时写 I2CxTRN 会使硬件将该位置 1 (由软件清零)。
- bit 6 **I2COV:** 接收溢出标志位
1 = 当 I2CxRCV 寄存器仍然保存原先的字节时接收到了新字节
0 = 未溢出
尝试将数据从 I2CxRSR 传输到 I2CxRCV 时由硬件置 1 (由软件清零)。
- bit 5 **D/A:** 数据 / 地址位 (作为 I²C 从器件工作时)
1 = 表示上次接收的字节为数据
0 = 表示上次接收的字节为器件地址
器件地址匹配时由硬件清零。发送完成或接收到从器件字节时由硬件置 1。

寄存器 16-2: I2CxSTAT: I2Cx 状态寄存器 (续)

bit 4	P: 停止位 1 = 表示上一次检测到停止位 0 = 上一次未检测到停止位 当检测到启动、重复启动或停止条件时由硬件置 1 或清零。
bit 3	S: 启动位 1 = 表示上一次检测到启动（或重复启动）位 0 = 上一次未检测到启动位 当检测到启动、重复启动或停止条件时由硬件置 1 或清零。
bit 2	R/W: 读 / 写信息位（作为 I ² C 从器件工作时） 1 = 读——表示数据自从器件输出 0 = 写——表示数据输入到从器件 接收到 I ² C 器件地址字节后由硬件置 1 或清零。
bit 1	RBF: 接收缓冲器满状态位 1 = 接收完成, I2CxRCV 为满 0 = 接收未完成, I2CxRCV 为空 用接收到的字节写 I2CxRCV 时由硬件置 1。用软件读 I2CxRCV 时由硬件清零。
bit 0	TBF: 发送缓冲器满状态位 1 = 发送正在进行中, I2CxTRN 为满 0 = 发送完成, I2CxTRN 为空 用软件写 I2CxTRN 时由硬件置 1。数据发送完成时由硬件清零。

PIC24FJ256GB110 系列

寄存器 16-3: I2CxMSK: I2Cx 从模式地址掩码寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	AMSK9	AMSK8
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| AMSK7 | AMSK6 | AMSK5 | AMSK4 | AMSK3 | AMSK2 | AMSK1 | AMSK0 |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-10 未实现: 读为 0

bit 9-0 AMSK<9:0>: 地址中 bit x 的掩码选择位

1 = 使能输入报文的地址中 bit x 的掩码; 在此位置上不需要位匹配

0 = 禁止地址中 bit x 的掩码; 在此位置上需要位匹配

17.0 通用异步收发器 (UART)

注: 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见《PIC24F 系列参考手册》的第 21 章 “UART” (DS39708A_CN)。

通用异步收发器 (Universal Asynchronous Receiver Transmitter, UART) 模块是 PIC24F 器件系列提供的串行 I/O 模块之一。UART 是可以与外设 (例如个人电脑、LIN、RS-232 和 RS-485 接口) 通信的全双工异步系统。模块还通过 UxCTS 和 UxRTS 引脚支持硬件流控制选项, 其中还包括 IrDA® 编码器和解码器。

UART 模块的主要特性有:

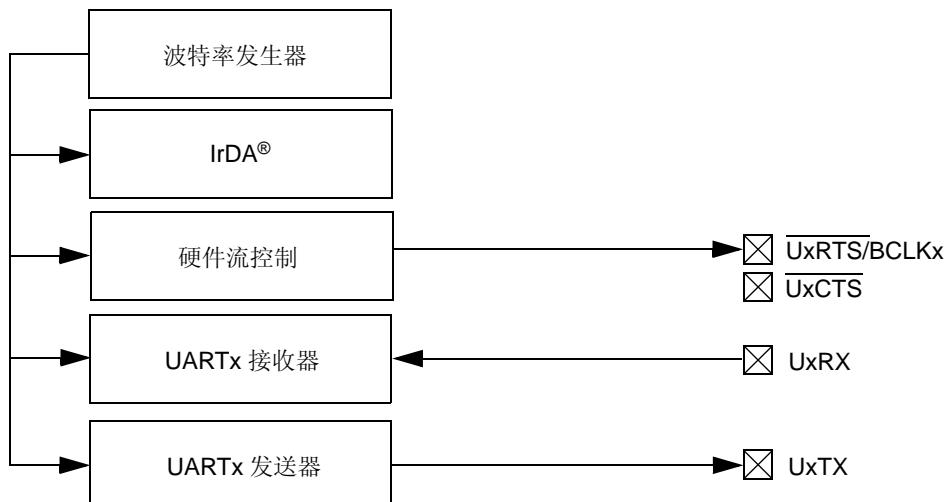
- 通过 UxTX 和 UxRX 引脚进行全双工 8 位或 9 位数据传输
- 偶校验、奇校验或无奇偶校验选项 (对于 8 位数据)
- 一个或两个停止位
- 通过 UxCTS 和 UxRTS 引脚支持硬件流控制选项

- 完全集成的波特率发生器, 具有 16 位预分频器
- 当器件工作在 16 MIPS 时, 波特率范围从 1 Mbps 到 15 Mbps
- 4 级深度先进先出 (First-In-First-Out, FIFO) 发送数据缓冲器
- 4 级深度 FIFO 接收数据缓冲器
- 奇偶校验、帧和缓冲器溢出错误检测
- 支持带地址检测的 9 位模式 (第 9 位 = 1)
- 发送和接收中断
- 用于诊断支持的环回模式
- 支持同步和间隔字符
- 支持自动波特率检测
- IrDA® 编码器和解码器逻辑
- 用于 IrDA 支持的 16 倍频波特率时钟输出

图 17-1 给出了 UART 的简化框图。UART 模块由以下至关重要的硬件组件组成:

- 波特率发生器
- 异步发送器
- 异步接收器

图 17-1: UART 简化框图



注: UART 输入和输出都必须在使用前分配给可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

17.1 UART 波特率发生器 (BRG)

UART 模块包含一个专用的 16 位波特率发生器。UxBRG 寄存器控制一个自由运行的 16 位定时器的周期。公式 17-1 给出了 BRGH = 0 时计算波特率的公式。

公式 17-1: UART 波特率 (BRGH = 0) (1,2)

$$\text{波特率} = \frac{\text{FCY}}{16 \cdot (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{\text{FCY}}{16 \cdot \text{波特率}} - 1$$

- 注 1: FCY 表示指令周期时钟频率 (Fosc/2)。
2: 基于 FCY = Fosc/2, 打盹模式和 PLL 被禁止。

最大可能波特率 (BRGH = 0) 是 FCY/16 (当 UxBRG = 0 时), 最小可能波特率是 FCY/(16 * 65536)。

公式 17-2 给出了 BRGH = 1 时计算波特率的公式。

公式 17-2: UART 波特率 (BRGH = 1) (1,2)

$$\text{波特率} = \frac{\text{FCY}}{4 \cdot (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{\text{FCY}}{4 \cdot \text{波特率}} - 1$$

- 注 1: FCY 表示指令周期时钟频率。
2: 基于 FCY = Fosc/2, 打盹模式和 PLL 被禁止。

最大可能波特率 (BRGH = 1) 是 FCY/4 (当 UxBRG = 0 时), 最小可能波特率是 FCY/(4 * 65536)。

向 UxBRG 寄存器写入新值会使 BRG 定时器复位 (清零)。这可以确保 BRG 无需等待定时器溢出就可以产生新的波特率。

例 17-1 给出了如下条件下的波特率误差计算:

- FCY = 4 MHz
- 目标波特率 = 9600

例 17-1: 波特率误差计算 (BRGH = 0) (1)

$$\text{目标波特率} = \text{FCY}/(16(\text{UxBRG} + 1))$$

求解 UxBRG 值:

$$\text{UxBRG} = ((\text{FCY}/\text{目标波特率})/16) - 1$$

$$\text{UxBRG} = ((4000000/9600)/16) - 1$$

$$\text{UxBRG} = 25$$

$$\text{计算波特率} = 4000000/(16(25 + 1))$$

$$= 9615$$

$$\text{误差} = (\text{计算波特率} - \text{目标波特率})/\text{目标波特率}$$

$$= (9615 - 9600)/9600$$

$$= 0.16\%$$

- 注 1: 基于 FCY = Fosc/2, 打盹模式和 PLL 被禁止。

17.2 8位数据模式下的发送

1. 设置 UART:
 - a) 将适当的值写入数据位、奇偶校验位和停止位。
 - b) 将适当的波特率值写入 UxBRG 寄存器。
 - c) 设置发送和接收中断允许位和优先级位。
2. 使能 UART。
3. 将 UTXEN 位置 1（该位置 1 后发送操作中断两个周期）。
4. 将数据字节写入 UxTXREG 字的低字节。该值将被立即传输到发送移位寄存器（Transmit Shift Register, TSR），且在波特率时钟的下一个上升沿开始移出串行比特流。
5. 或者，当 UTXEN = 0 时，数据字节也可被发送，且随后用户可将 UTXEN 置 1。由于波特率时钟将从清零状态启动，这将立即开始发送串行比特流。
6. 中断控制位 UTXISELx 的设置决定何时产生发送中断。

17.3 9位数据模式下的发送

1. 设置 UART（如第 17.2 节“8位数据模式下的发送”中所述）。
2. 使能 UART。
3. 将 UTXEN 位置 1（产生发送中断）。
4. 仅向 UxTXREG 写入一个 16 位的值。
5. 向 UxTXREG 写入一个字可触发 9 位数据向 TSR 的传输。串行比特流将会在波特率时钟的第一个上升沿开始移出。
6. 中断控制位 UTXISELx 的设置决定何时产生发送中断。

17.4 间隔和同步发送序列

下述序列会发送一个报文帧头，包括一个间隔字符和其后的一个自动波特率同步字符。

1. 将 UART 配置为所需的模式。
2. 将 UTXEN 和 UTXBRK 置 1 以设置间隔字符。
3. 将一个“虚拟”字符装入 UxTXREG 中以启动发送（值被忽略）。
4. 向 UxTXREG 写入 “55h”；将同步字符装入发送 FIFO 中。
5. 间隔字符发送后，硬件会将 UTXBRK 位复位。然后开始发送同步字符。

17.5 8位或9位数据模式下的接收

1. 设置 UART（如第 17.2 节“8位数据模式下的发送”中所述）。
2. 使能 UART。
3. 当接收到一个或多个数据字符时，将会根据中断控制位 URXISELx 的设置产生接收中断。
4. 读 OERR 位以确定是否发生了溢出错误。OERR 位必须用软件复位。
5. 读 UxRXREG。

读取 UxRXREG 字符的行为会将下一个字符传送到接收 FIFO 的顶部，其中包含一组新的 PERR 和 FERR 值。

17.6 UxCTS 和 UxRTS 控制引脚的操作

UARTx 允许发送（UxCTS）和请求发送（UxRTS）是两个与 UART 模块相关、由硬件控制的引脚。这两个引脚允许 UART 运行在单工模式和流控制模式下。它们用于控制 UART 与数据终端设备（Data Terminal Equipment, DTE）之间的发送和接收。UxMODE 寄存器中的 UEN<1:0> 位用来配置这两个引脚。

17.7 红外支持

UART 模块提供两种类型的红外 UART 支持：一种是 IrDA 时钟输出，用于支持外部 IrDA 编码器和解码器（传统模块支持）；另一种是完全实现的 IrDA 编码器和解码器。请注意，由于 IrDA 模式需要 16 倍频波特率时钟，它们仅在 BRGH 位（UxMODE<3>）为 0 时才能工作。

17.7.1 用于外部 IrDA 支持的 IrDA 时钟输出

为了支持外部 IrDA 编码器和解码器，可将 BCLKx 引脚（和 UxRTS 引脚相同）配置为产生 16 倍频波特率时钟。当使能了 UART 模块且 UEN<1:0> = 11 时，BCLKx 引脚将输出 16 倍频波特率时钟，用于支持 IrDA 编解码器芯片。

17.7.2 内置 IrDA 编码器和解码器

UART 模块在其内部完全实现了 IrDA 编码器和解码器。内置 IrDA 编码器和解码器的功能可通过 IREN 位（UxMODE<12>）来使能。当使能（IREN = 1）时，接收引脚（UxRX）可作为红外接收器的输入引脚。发送引脚（UxTX）可作为红外发送器的输出引脚。

PIC24FJ256GB110 系列

寄存器 17-1: UxMODE: UARTx 模式寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
UARTEN ⁽¹⁾	—	USIDL	IREN ⁽²⁾	RTSMD	—	UEN1	UEN0
bit 15	bit 8						

R/C-0, HC	R/W-0	R/W-0, HC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL
bit 7	bit 0						

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

bit 15	UARTEN: UARTx 使能位 ⁽¹⁾
	1 = 使能 UARTx; UARTx 根据 UEN<1:0> 的定义控制所有 UARTx 引脚 0 = 禁止 UARTx; 由端口锁存器控制所有 UARTx 引脚; UARTx 的功耗最小
bit 14	未实现: 读为 0
bit 13	USIDL: 空闲模式停止位 1 = 当器件进入空闲模式时, 模块停止工作 0 = 在空闲模式下模块继续工作
bit 12	IREN: IrDA® 编码器和解码器使能位 ⁽²⁾ 1 = 使能 IrDA 编码器和解码器 0 = 禁止 IrDA 编码器和解码器
bit 11	RTSMD: <u>UxRTS</u> 引脚模式选择位 1 = <u>UxRTS</u> 引脚处于单工模式 0 = <u>UxRTS</u> 引脚处于流控制模式
bit 10	未实现: 读为 0
bit 9-8	UEN<1:0>: UARTx 使能位 11 = 使能并使用 <u>UxTX</u> 、 <u>UxRX</u> 和 <u>BCLKx</u> 引脚; <u>UxCTS</u> 引脚由端口锁存器控制 10 = 使能并使用 <u>UxTX</u> 、 <u>UxRX</u> 、 <u>UxCTS</u> 和 <u>UxRTS</u> 引脚 01 = 使能并使用 <u>UxTX</u> 、 <u>UxRX</u> 和 <u>UxRTS</u> 引脚; <u>UxCTS</u> 引脚由端口锁存器控制 00 = 使能并使用 <u>UxTX</u> 和 <u>UxRX</u> 引脚; <u>UxCTS</u> 和 <u>UxRTS/BCLKx</u> 引脚由端口锁存器控制
bit 7	WAKE: 在休眠模式下检测到启动位唤醒使能位 1 = UARTx 将继续采样 <u>UxRX</u> 引脚; 在出现下降沿时产生中断, 在出现下一个上升沿时由硬件清零该位 0 = 禁止唤醒
bit 6	LPBACK: UARTx 环回模式选择位 1 = 使能环回模式 0 = 禁止环回模式
bit 5	ABAUD: 自动波特率使能位 1 = 使能对下一个字符的波特率测量——需要在接收其他数据前接收同步字段 (55h); 完成时由硬件清零 0 = 禁止波特率测量或测量已完成

- 注 1: 如果 UARTEN = 1, 则外设输入和输出必须配置给可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。
- 2: 此功能只能在 16 倍频 BRG 模式 (BRGH = 0) 下使用。

寄存器 17-1: UxMODE: UARTx 模式寄存器 (续)

bit 4	RXINV: 接收极性翻转位 1 = UxRX 的空闲状态为 0 0 = UxRX 的空闲状态为 1
bit 3	BRGH: 高波特率使能位 1 = 高速模式 (波特率来自 FCY/4) 0 = 标准模式 (波特率来自 FCY/16)
bit 2-1	PDSEL<1:0>: 奇偶校验和数据选择位 11 = 9 位数据, 无奇偶校验 10 = 8 位数据, 奇校验 01 = 8 位数据, 偶校验 00 = 8 位数据, 无奇偶校验
bit 0	STSEL: 停止位选择位 1 = 2 个停止位 0 = 1 个停止位

- 注 1: 如果 **UARTEN** = 1, 则外设输入和输出必须配置给可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。
- 2: 此功能只能在 16 倍频 BRG 模式 (**BRGH** = 0) 下使用。

PIC24FJ256GB110 系列

寄存器 17-2: UxSTA: UARTx 状态和控制寄存器

R/W-0	R/W-0	R/W-0	U-0	R/W-0 HC	R/W-0	R-0	R-1
UTXISEL1	UTXINV ⁽¹⁾	UTXISEL0	—	UTXBRK	UTXEN ⁽²⁾	UTXBF	TRMT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

图注:

R = 可读位

-n = POR 时的值

C = 可清零位

W = 可写位

1 = 置 1

HC = 硬件清零位

U = 未实现位, 读为 0

0 = 清零

x = 未知

bit 15,13	UTXISEL<1:0> : 发送中断模式选择位	
	11 = 保留; 不要使用	
	10 = 当一个字符被传输到发送移位寄存器 (TSR) 导致发送缓冲器为空时, 产生中断	
	01 = 当最后一个字符被移出发送移位寄存器; 所有发送操作完毕时产生中断	
	00 = 当一个字符被传输到发送移位寄存器 (这意味着发送缓冲器中至少还有一个字符) 时产生中断	
bit 14	UTXINV : IrDA® 编码器发送极性翻转位 ⁽¹⁾	
	<u>IREN = 0</u> :	
	1 = UxTX 的空闲状态为 0	
	0 = UxTX 的空闲状态为 1	
	<u>IREN = 1</u> :	
	1 = UxTX 的空闲状态为 1	
	0 = UxTX 的空闲状态为 0	
bit 12	未实现: 读为 0	
bit 11	UTXBRK : 发送间隔位	
	1 = 在下次发送时发出同步间隔字符——启动位, 后跟 12 个 0 位, 然后是停止位; 完成时由硬件清零	
	0 = 禁止或已完成同步间隔字符的发送	
bit 10	UTXEN : 发送使能位 ⁽²⁾	
	1 = 使能发送, UARTx 控制 UxTX 引脚	
	0 = 禁止发送, 中止所有等待的发送, 缓冲器复位。由端口控制 UxTX 引脚。	
bit 9	UTXBF : 发送缓冲器满状态位 (只读)	
	1 = 发送缓冲器满	
	0 = 发送缓冲器未满, 至少还可写入一个或多个字符	
bit 8	TRMT : 发送移位寄存器空位 (只读)	
	1 = 发送移位寄存器为空, 同时发送缓冲器为空 (上一次发送已完成)	
	0 = 发送移位寄存器非空, 发送在进行中或排队	
bit 7-6	URXISEL<1:0> : 接收中断模式选择位	
	11 = 当 RSR 传输使接收缓冲器为满时 (即, 有 4 个数据字符), 中断标志位置 1	
	10 = 当 RSR 传输使接收缓冲器 3/4 满时 (即, 有 3 个数据字符), 中断标志位置 1	
	0x = 当接收到任一字符且 RSR 的内容被传输给接收缓冲器时, 中断标志位置 1。接收缓冲器有一个或多个字符。	

注 1: 仅当使能了 IrDA 编码器 (IREN = 1) 时, 该位的值才影响模块的发送属性。

2: 如果 UARLEN = 1, 则外设输入和输出必须配置给可用的 RPn 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”。

寄存器 17-2: UxSTA: UARTx 状态和控制寄存器 (续)

bit 5	ADDEN: 地址字符检测位 (接收数据的第 8 位 = 1) 1 = 使能地址检测模式。如果没有选择 9 位模式，这个控制位将无效。 0 = 禁止地址检测模式
bit 4	RIDLE: 接收器空闲位 (只读) 1 = 接收器空闲 0 = 接收器工作
bit 3	PERR: 奇偶校验错误状态位 (只读) 1 = 检测到当前字符 (在接收 FIFO 顶部的字符) 的奇偶校验错误 0 = 未检测到奇偶校验错误
bit 2	FERR: 帧错误状态位 (只读) 1 = 检测到当前字符 (在接收 FIFO 顶部的字符) 的帧错误 0 = 未检测到帧错误
bit 1	OERR: 接收缓冲器溢出错误状态位 (清零 / 只读) 1 = 接收缓冲器已溢出 0 = 接收缓冲器未溢出。清零原来置1的OERR位 ($1 \rightarrow 0$ 的跳变) 将使接收缓冲器和RSR复位到空状态
bit 0	URXDA: 接收缓冲器中是否有数据位 (只读) 1 = 接收缓冲器中有数据，有至少一个或多个字符可被读取 0 = 接收缓冲器为空

- 注 1: 仅当使能了 IrDA 编码器 (IREN = 1) 时，该位的值才影响模块的发送属性。
2: 如果 UARTEN = 1，则外设输入和输出必须配置给可用的 RPn 引脚。更多信息，请参见第 10.4 节 “外设引脚选择”。

PIC24FJ256GB110 系列

注:

18.0 带 ON-THE-GO 支持的通用串行总线（USB OTG）

注：本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》的第 27 章“USB On-The-Go (OTG)”。

PIC24FJ256GB110 系列器件包含全速和低速兼容的 USB 串行接口引擎（Serial Interface Engine, SIE），并带有 On-The-Go (OTG) 功能。OTG 功能使器件可用作 USB 外设器件，也可以用作具有有限主机功能的 USB 嵌入式主机。OTG 功能允许器件用 OTG 的主机协商协议（HNP）在器件与主机操作间动态切换。

关于 OTG 操作的详细信息，请参见 USB-IF 公布的“USB 2.0 规范 On-The-Go 补充信息”。关于 USB 操作的更多详细信息，请参见“通用串行总线规范” v2.0。

USB OTG 模块提供以下特性：

- 器件和主机模式下的 USB 功能与应用程序控制模式下的 OTG 功能切换
- 可用软件选择的模块速度：全速（12 Mbps）或低速（1.5 Mbps，仅在主机模式下可用）
- 支持所有 4 种 USB 传输类型：控制、中断、批量和同步
- 总共 32 个单向端点可构成 16 个双向端点
- 用于数据 RAM 快速操作的 DMA 接口
- 最多 16 个单向端点传输队列，无需服务
- 集成的片上 USB 收发器，通过数字接口支持片外收发器
- 使用片上比较器的集成 VBUS 生成和升压生成，通过数字接口支持外部 VBUS 比较器和稳压器
- 片上总线上拉和下拉电阻配置

图 18-1 给出了 USB OTG 模块的简化框图。

USB OTG 模块可用作 USB 外设器件或 USB 主机，并且可通过软件控制在器件和主机模式间动态切换。两种模式使用相同的数据路径和缓冲器描述符进行数据的发送和接收。

讨论 USB 操作时，本章使用以控制器为中心的命名法说明单片机与 USB 间数据传输的方向。Rx（接收）用于说明将数据从 USB 移动到单片机的传输，Tx（发送）用于说明将数据从单片机移动到 USB 的传输。表 18-1 给出了这种命名法的数据方向与 USB 令牌交换间的关系。

表 18-1：USB 主机或目标器件以控制器为中心的数据方向

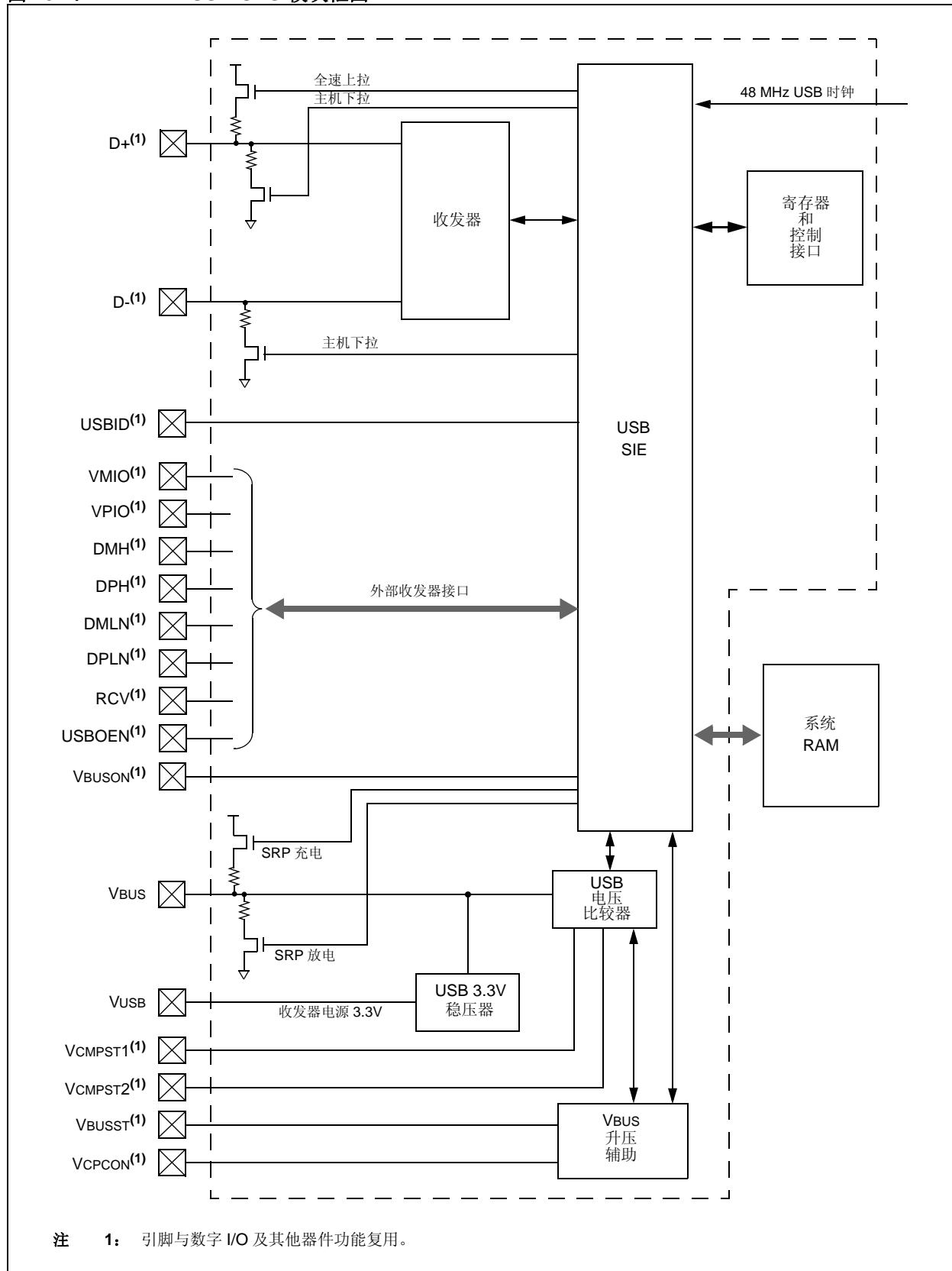
USB 模式	方向	
	接收	发送
器件	OUT 或 SETUP	IN
主机	IN	OUT 或 SETUP

本章将介绍在应用中实现 USB OTG 功能需要的基本操作。对 USB 协议及其 OTG 补充信息的完整详细的讨论超出了本数据手册的范围。我们假定用户对 USB 架构和最新版该协议已有基本了解。

这里并未囊括正确的 USB 操作的所有步骤（如器件枚举）。建议应用程序开发人员用正确的器件驱动程序实现所有必需的功能。Microchip 提供许多特定于应用的资源，例如 USB 固件和驱动程序支持。请访问 www.microchip.com 获得最新的固件和驱动程序支持。

PIC24FJ256GB110 系列

图 18-1: USB OTG 模块框图



18.1 硬件配置

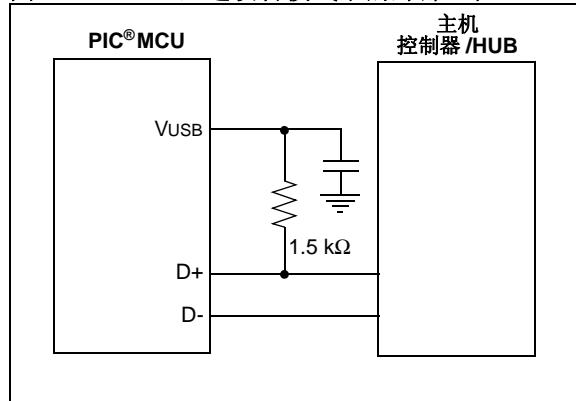
18.1.1 器件模式

18.1.1.1 D+ 上拉电阻

PIC24FJ256GB110 系列器件在 D+ 线上有一个内置的 $1.5\text{ k}\Omega$ 电阻，当单片机工作在设备模式下时可用。这用于以信号形式告知外部主机：器件工作在全速设备模式。可通过 DPPULUP (U1OTGCON<7>) 位使能 D+ 上拉。

或者，也可以在 D+ 上使用外部电阻，如图 18-2 所示。

图 18-2：全速设备模式下的外部上拉



18.1.1.2 电源模式

许多 USB 应用可能有多组不同的电源要求和配置。最常遇见的电源模式有：

- 仅总线电源
- 仅自供电
- 双电源且自供电为主

仅总线电源模式（图 18-3）是最简单的方法。应用的所有电源都从 USB 汲取。

要满足 USB 2.0 规范的涌流要求，VBUS 和地之间的总有效电容不得大于 $10\text{ }\mu\text{F}$ 。

在 USB 暂挂模式下，器件从 USB 电缆的 5V VBUS 线消耗的电流不得大于 2.5 mA 。在 USB 暂挂模式期间，D+ 或 D- 上拉电阻必须保持有效，其消耗的电流为允许的暂挂电流的一部分。

在仅自供电模式（图 18-4）下，USB 应用有自己的电源，从 USB 消耗的电源极小。注意，将添加一个连接指示，表示何时连接 USB 以及主机何时有效地为 VBUS 供电。

要满足规范要求，在主机将 VBUS 有效地驱动为高电平之前不应使能 USB 模块以及 D+ 或 D- 上拉电阻。可使用一个 5.5V 容差的 I/O 引脚来实现此目的。

应用不应向 USB 电缆的 5V VBUS 引脚汲取任何电流。

双电源且自供电为主的选项（图 18-5）允许应用主要使用内部电源，但是在内部电源不可用的情况下可切换到从 USB 供电。双电源器件还必须满足前面描述的涌流和暂挂模式电流的特殊要求，并且在 VBUS 驱动为高电平之前不得使能 USB 模块。

图 18-3：仅总线电源

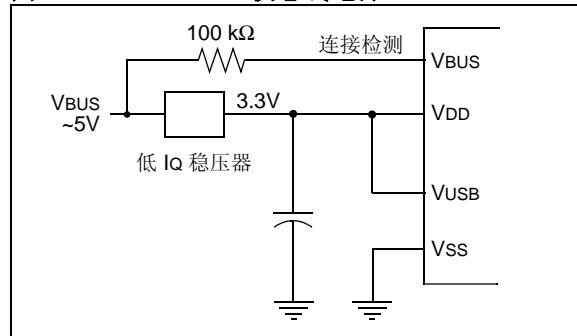


图 18-4：仅自供电

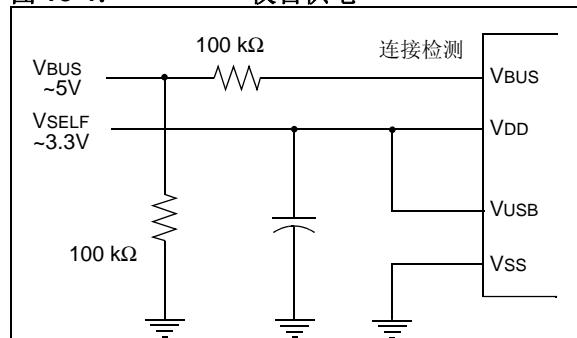
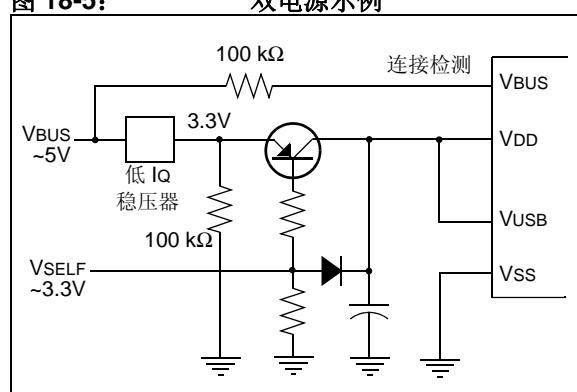


图 18-5：双电源示例



PIC24FJ256GB110 系列

18.1.2 主机和 OTG 模式

18.1.2.1 D+ 和 D- 下拉电阻

PIC24FJ256GB110 系列器件在 D+ 和 D- 线上有内置的 $15\text{ k}\Omega$ 下拉电阻。这两个电阻串联起来，以信号形式告知总线：单片机工作在主机模式下。将 DPPULDWN 和 DMPULDWN (U1OTGCON<5:4>) 位置 1 可使能这两个下拉电阻。

18.1.2.2 电源配置

在主机模式下以及主机模式下的 On-the-Go 操作，USB 2.0 规范要求主机应用在 VBUS 上提供电源。由于单片机

以低于 VBUS 的电压运行，而且无法提供足够的电流，因此必须提供单独的电源。

当应用始终在主机模式下工作时，可使用一个简单的电路在总线上为 VBUS 供电并调节总线上的电流（图 18-6）。对于 OTG 操作，必须能够根据需要开启或关闭 VBUS，因为单片机会在设备模式和主机模式之间切换。图 18-7 给出了使用外部电荷泵的典型示例。

图 18-6： 主机接口示例

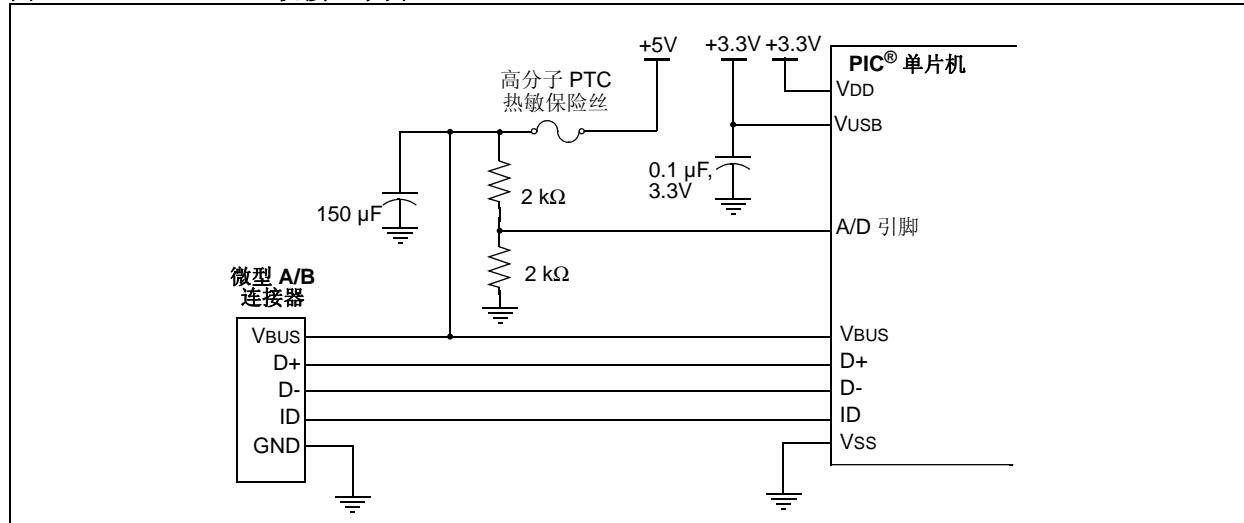
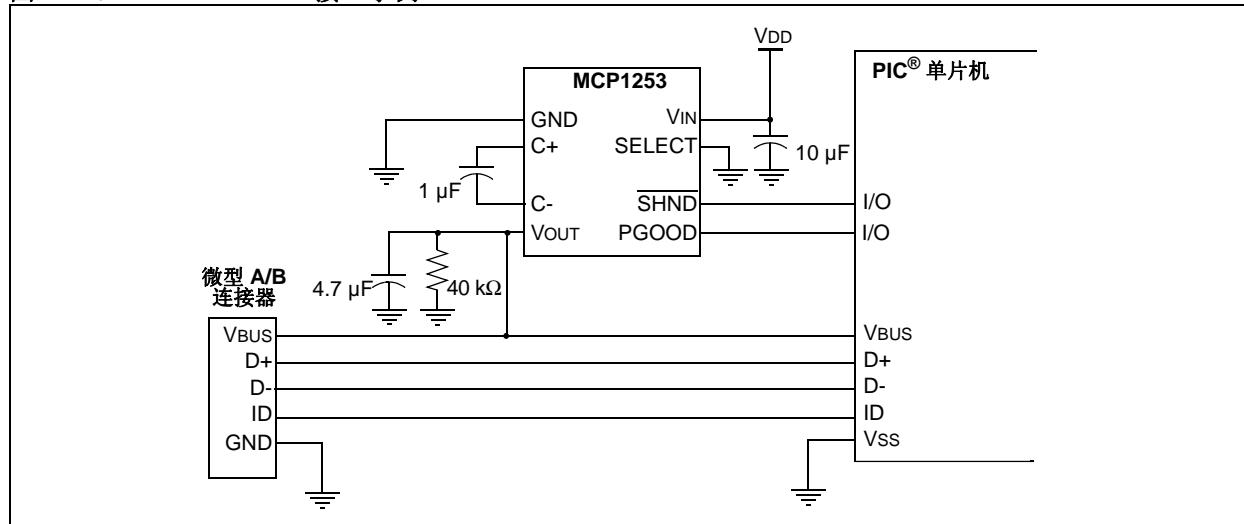


图 18-7： OTG 接口示例



18.1.2.3 使用外部器件生成 VBUS 电压

作为 USB 主机（即作为 OTG 配置中的 A 设备或嵌入式主机）工作时，必须为连接的器件提供 VBUS。PIC24FJ256GB110 系列器件具有一个内部 VBUS 升压辅助工具，帮助从电路板上的可用电压产生所需的 5V VBUS。该辅助工具由一个简单的 PWM 输出（用于控制开关电源）和内置比较器（用于监视输出电压和限制电流）组成。

要使能电压生成：

1. 确认 USB 模块已供电 ($\text{U1PWRC}\langle 0 \rangle = 1$)，并且已禁止 VBUS 放电 ($\text{U1OTGCON}\langle 0 \rangle = 0$)。
2. 如果需要，设置 PWM 周期 ($\text{U1PWMRRS}\langle 7:0 \rangle$) 和占空比 ($\text{U1PWMRRS}\langle 15:8 \rangle$)。
3. 根据外部电路的配置，用 PWMPOL 位 ($\text{U1PWMCON}\langle 9 \rangle$) 选择所需的输出信号的极性。
4. 用 VBUSCHG 位 ($\text{U1OTGCON}\langle 1 \rangle$) 选择所需的目标电压。
5. 通过将 CNTEN 位 ($\text{U1PWMCON}\langle 8 \rangle$) 设置为 1 使能 PWM 计数器。
6. 通过将 PWMEN 位 ($\text{U1PWMCON}\langle 15 \rangle$) 设置为 1 使能 PWM 模块。
7. 使能 VBUS 生成电路 ($\text{U1OTGCON}\langle 3 \rangle = 1$)。

注：本节将说明 VBUS 电压生成和控制的一般过程。更多示例请参见《PIC24F 系列参考手册》。

公式 18-1：估算 USB 收发器的电流消耗

$$\text{IXCVR} = \frac{(40 \text{ mA} \cdot \text{VUSB} \cdot \text{PZERO} \cdot \text{PIN} \cdot \text{LCABLE})}{(3.3\text{V} \cdot 5\text{m})} + \text{IPULLUP}$$

图注： VUSB——施加到 VUSB 引脚上的电压（单位为 V，3.0V 至 3.6V）。

PZERO——PIC® 单片机发送的 IN（输入）通信中值为 0 的位所占的百分比（以 10 进制数表示）。

PIN——用于 IN 通信的总线带宽占总线带宽的百分比（以 10 进制数表示）。

LCABLE——USB 电缆的长度（单位为 m）。USB 2.0 规范要求全速应用不得使用长度超过 5m 的电缆。

IPULLUP——1.5 kΩ 的标称上拉电阻（被使能时）必须为 USB 电缆提供的电流。

18.1.3 使用外部接口

某些应用可能需要将 USB 接口与系统的其余部分隔离。PIC24FJ256GB110 系列器件包含一个可与外部 USB 收发器通信的完整接口，并通过该接口控制外部 USB 收发器，包括数据线上拉和下拉的控制。也可针对不同的 VBUS 发生拓扑来配置 VBUS 电压发生控制电路。

有关使用外部接口的信息，请参见《PIC24F 系列参考手册》的第 27 章“USB On-The-Go (OTG)”。

18.1.4 计算收发器功耗要求

根据 USB 电缆的特性阻抗、电缆长度、VUSB 供电电压以及 USB 电缆上的实际数据传输模式，USB 收发器消耗的电流量也有所不同。电缆越长，电容就越大，切换到输出状态时消耗的总能耗也就越多。对于不同的应用，收发器的总电流消耗也各不相同。公式 18-1 可帮助估算在全速应用中可能需要的实际电流量。

关于收发器功耗的全面讨论，请参见《PIC24F 系列参考手册》的第 27 章“USB On-The-Go (OTG)”。

18.2 USB 缓冲器描述符和 BDT

端点缓冲器控制是通过叫做缓冲器描述符表（Buffer Descriptor Table, BDT）的结构处理的。它为用户提供了灵活的方法，用于构造和控制不同长度和配置的端点缓冲器。

BDT 可位于任何可用的 512 字节对齐的数据 RAM 块中。BDT 指针（U1BDTP1）包含 BDT 的高位地址字节，设置 BDT 在 RAM 中的位置。用户必须设置该指针，以指示该表的位置。

BDT 由缓冲器描述符（Buffer Descriptor, BD）组成，后者用于定义和控制 USB RAM 空间中的实际缓冲器。每个 BD 都包含两个 16 位“软”（非固定地址）寄存器、BDnSTAT 和 BDnADR，其中 n 代表 64 个可能的 BD（范围从 0 到 63）之一。BDnSTAT 是 BDn 的状态寄存器，而 BDnADR 指定与 BDn 相关的缓冲器的起始地址。

根据所用的端点缓冲配置，对于总共 256 字节，最多有 64 组缓冲器描述符。BDT 至少要有 8 个字节长。这是因为 USB 规范要求每个器件都必须有带输入和输出的端点 0，用于初始设置。

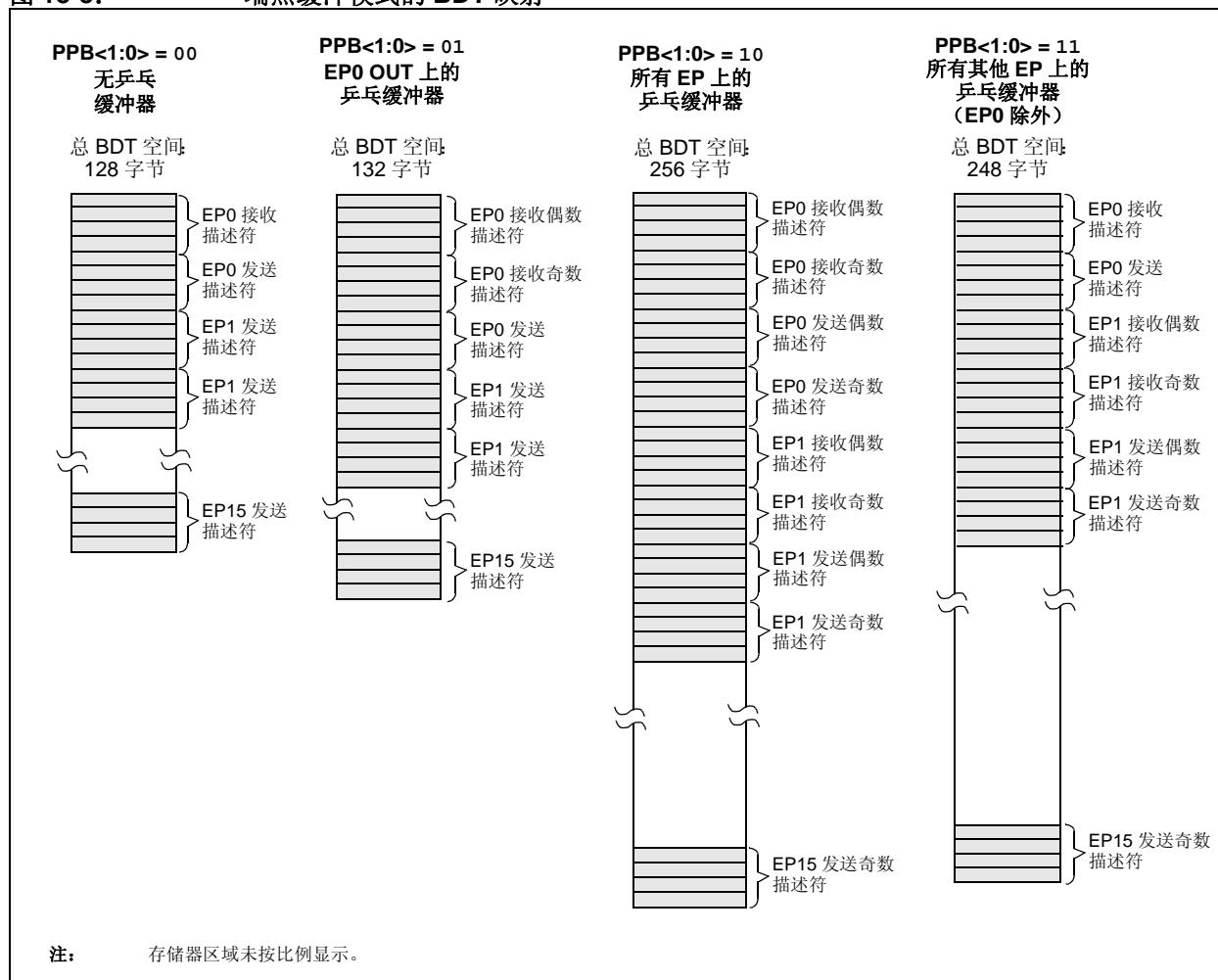
BDT 中的端点映射取决于三个变量：

- 端点号（0 到 15）
- 端点方向（接收或发送）
- 乒乓设置（U1CNFG1<1:0>）

图 18-8 例举了如何用这些变量在 BDT 中映射端点。

在主机模式下，只使用端点 0 缓冲器描述符。所有传输都使用端点 0 缓冲器描述符和端点控制寄存器（U1EP0）。对于接收到的数据包，所连接器件的源端点用 USB 状态寄存器中的 ENDPT<3:0>（U1STAT<7:4>）的值表示。对于发送的数据包，所连接器件的目标端点用写入令牌寄存器（U1TOK）的值表示。

图 18-8：端点缓冲模式的 BDT 映射



根据缓冲配置，BD 与特定端点有固定关系。表 18-2 提供了 BD 到端点的映射。这种关系还意味着如果没有连续使能端点，BDT 中可能会存在间隙。从理论上而言，这表示用于已禁止端点的 BD 能够用作缓冲空间。但实际上，用户应避免在 BDT 中使用这类空间，除非执行验证 BD 地址操作。

18.2.1 缓冲器所有权

由于缓冲器及其 BD 是在 CPU 和 USB 模块之间共享的，因此采用了简单的信号机制来区分允许哪个模块更新 BD 及其在存储器中的相关缓冲区。这是通过使用 UOWN 位作为信号来实现的。UOWN 是在 BDnSTAT 的两个配置之间共享的唯一一个位。

UOWN 清零时，BD 归单片机内核“所有”。UOWN 位置 1 时，BD 和缓冲存储器归 USB 外设“所有”。此时内核不应修改 BD 或其对应的数据缓冲器。注意，SIE 拥有缓冲器时，单片机内核仍可以读取 BDnSTAT，反之亦然。

表 18-2：不同缓冲模式下缓冲区描述符的分配

端点	分配给端点的 BD							
	模式 0 (无乒乓缓冲区)		模式 1 (EP0 OUT 上的乒乓缓冲区)		模式 2 (所有 EP 上的乒乓缓冲区)		模式 3 (所有其他 EP 上的乒乓缓冲区，EP0 除外)	
	输出	输入	输出	输入	输出	输入	输出	输入
0	0	1	0 (E), 1 (O)	2	0 (E), 1 (O)	2 (E), 3 (O)	0	1
1	2	3	3	4	4 (E), 5 (O)	6 (E), 7 (O)	2 (E), 3 (O)	4 (E), 5 (O)
2	4	5	5	6	8 (E), 9 (O)	10 (E), 11 (O)	6 (E), 7 (O)	8 (E), 9 (O)
3	6	7	7	8	12 (E), 13 (O)	14 (E), 15 (O)	10 (E), 11 (O)	12 (E), 13 (O)
4	8	9	9	10	16 (E), 17 (O)	18 (E), 19 (O)	14 (E), 15 (O)	16 (E), 17 (O)
5	10	11	11	12	20 (E), 21 (O)	22 (E), 23 (O)	18 (E), 19 (O)	20 (E), 21 (O)
6	12	13	13	14	24 (E), 25 (O)	26 (E), 27 (O)	22 (E), 23 (O)	24 (E), 25 (O)
7	14	15	15	16	28 (E), 29 (O)	30 (E), 31 (O)	26 (E), 27 (O)	28 (E), 29 (O)
8	16	17	17	18	32 (E), 33 (O)	34 (E), 35 (O)	30 (E), 31 (O)	32 (E), 33 (O)
9	18	19	19	20	36 (E), 37 (O)	38 (E), 39 (O)	34 (E), 35 (O)	36 (E), 37 (O)
10	20	21	21	22	40 (E), 41 (O)	42 (E), 43 (O)	38 (E), 39 (O)	40 (E), 41 (O)
11	22	23	23	24	44 (E), 45 (O)	46 (E), 47 (O)	42 (E), 43 (O)	44 (E), 45 (O)
12	24	25	25	26	48 (E), 49 (O)	50 (E), 51 (O)	46 (E), 47 (O)	48 (E), 49 (O)
13	26	27	27	28	52 (E), 53 (O)	54 (E), 55 (O)	50 (E), 51 (O)	52 (E), 53 (O)
14	28	29	29	30	56 (E), 57 (O)	58 (E), 59 (O)	54 (E), 55 (O)	56 (E), 57 (O)
15	30	31	31	32	60 (E), 61 (O)	62 (E), 63 (O)	58 (E), 59 (O)	60 (E), 61 (O)

图注：(E) = 偶数事务缓冲区，(O) = 奇数事务缓冲区

缓冲器描述符随寄存器更新来源不同而有不同含义。寄存器 18-1 和寄存器 18-2 显示了 BDnSTAT 当前“所有权”不同时的差别。

UOWN 置 1 时，用户不能再信赖写入 BD 的值。从此时起，USB 模块根据需要更新 BD，改写原始 BD 值。BDnSTAT 寄存器由 SIE 用令牌 PID 更新，传输计数也被更新。

18.2.2 DMA 接口

USB OTG 模块使用专用 DMA 访问 BDT 和端点数据缓冲器。由于 DMA 的部分地址空间专用于缓冲器描述符，连接到 DMA 的存储器的一部分必须包含正确映射的连续地址空间，供该模块访问。

PIC24FJ256GB110 系列

寄存器 18-1: **BDnSTAT**: 缓冲器描述符 n 状态寄存器原型, USB 模式 (BD0STAT 到 BD63STAT)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
UOWN	DTS	PID3	PID2	PID1	PID0	BC9	BC8
bit 15							bit 8

| R/W-x |
|-------|-------|-------|-------|-------|-------|-------|-------|
| BC7 | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **UOWN**: USB 所有权位
1 = USB 模块拥有 BD 及其对应缓冲器; CPU 不得修改 BD 或缓冲器
- bit 14 **DTS**: 数据翻转数据包位
1 = 数据 1 数据包
0 = 数据 0 数据包
- bit 13-10 **PID<3:0>**: 数据包标识符位 (由 USB 模块写入)
在器件模式下: 表示上次传输中收到的令牌的 PID。
在主机模式下: 表示上次返回的 PID 或传输状态指示器。
- bit 9-0 **BC<9:0>**: 字节计数位
它表示一次传输中发送的字节数或接收的最大字节数。完成后, 由 USB 模块用实际发送或接收的字节数更新字节计数。

寄存器 18-2: BDnSTAT: 缓冲器描述符 n 状态寄存器原型, CPU 模式 (BD0STAT 到 BD63STAT)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
UOWN	DTS ⁽¹⁾	0	0	DTSEN	BSTALL	BC9	BC8
bit 15							bit 8

| R/W-x |
|-------|-------|-------|-------|-------|-------|-------|-------|
| BC7 | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **UOWN:** USB 所有权位
0 = 单片机内核拥有 BD 及其对应缓冲器。USB 模块忽略 BD 中的所有其他字段。
- bit 14 **DTS:** 数据翻转数据包位⁽¹⁾
1 = 数据 1 数据包
0 = 数据 0 数据包
- bit 13-12 保留的功能: 保持为 0
- bit 11 **DTSEN:** 数据翻转同步使能位
1 = 使能数据翻转同步; 同步值不正确的数据包将被忽略
0 = 不会执行数据翻转同步
- bit 10 **BSTALL:** 缓冲器停止使能位
1 = 使能缓冲器停止; 如果收到使用指定地址 BD 的令牌, 将发出停止握手 (UOWN 位保持置 1, BD 值不变); 在任何停止握手发生时, 对应的 EPSTALL 位将置 1
0 = 禁止缓冲器停止
- bit 9-0 **BC<9:0>:** 字节计数位
它表示一次传输中发送的字节数或接收的最大字节数。完成后, 由 USB 模块用实际发送或接收的字节数更新字节计数。

注 1: 除非 DTSEN = 1, 否则该位将被忽略。

PIC24FJ256GB110 系列

18.3 USB 中断

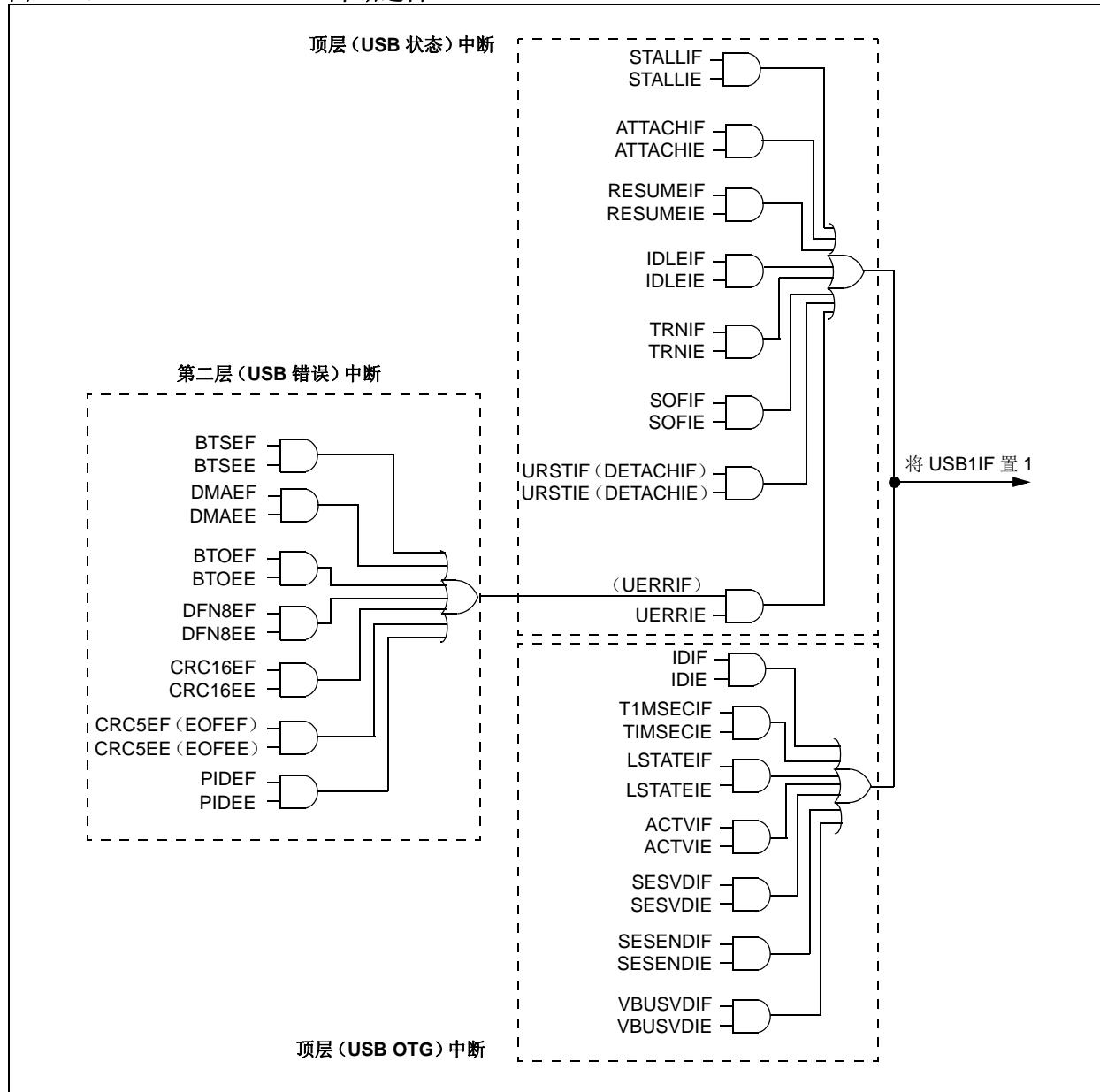
USB OTG 模块在许多条件下可配置为引起中断。所有中断源使用相同的中断向量。

图 18-9 给出了 USB 模块的中断逻辑。USB 模块的中断寄存器分为两层。顶层全部由 USB 状态中断组成；这些中断分别是在 U1IE 和 U1IR 寄存器中允许和标记的。

第二层由 USB 错误条件中断组成，分别是在 U1EIR 和 U1IEIE 寄存器中允许和标记的。任意一个中断条件都将触发顶层的 USB 错误中断标志（UERRIF）。

中断可用于在 USB 事务中产生陷阱。图 18-10 提供了一些 USB 帧中常见的事件及其相应的中断。

图 18-9: USB OTG 中断逻辑



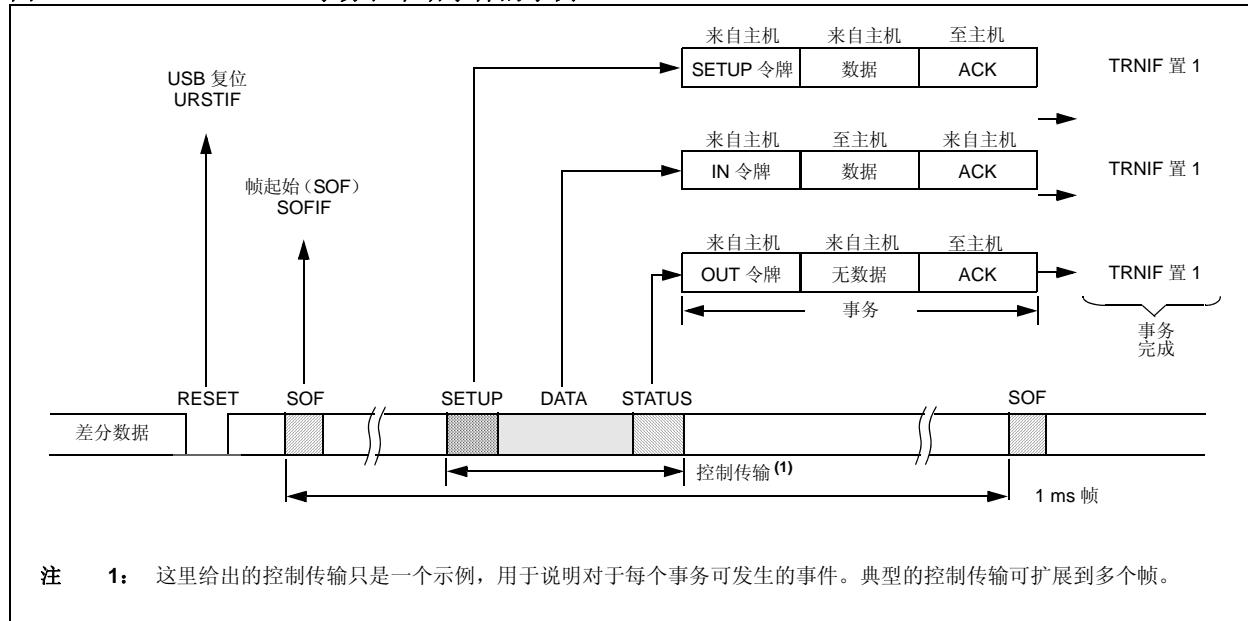
18.3.1 清除 USB OTG 中断

和器件级中断不同，USB OTG 中断的状态标志位不能用软件自由写入。所有的 USB OTG 标志位都实现为只能由硬件置 1。此外，这些位只能在软件中通过写入 1 清零（即执行 MOV 类指令）。将 0 写入标志位（即 BCLR 指令）不起作用。

能由硬件置 1。此外，这些位只能在软件中通过写入 1 清零（即执行 MOV 类指令）。将 0 写入标志位（即 BCLR 指令）不起作用。

注： 在本数据手册中，只能通过写入 1 清零的位称为“写 1 清零”。在寄存器说明中，该功能用描述符“K”表示。

图 18-10：USB 事务和中断事件的示例



18.4 设备模式的操作

下一节将说明如何执行常见的器件模式任务。在器件模式下，USB 传输在传输层执行。USB 模块会自动执行传输的状态阶段。

18.4.1 使能器件模式

1. 通过置 1 然后清零乒乓缓冲器复位位 PPBRST (U1CON<1>) 将乒乓缓冲器指针复位。
2. 禁止所有中断 (U1IE 和 U1EIE = 00h)。
3. 通过将 FFh 写入 U1IR 和 U1EIR 清零任何现有中断标志位。
4. 确认 VBUS 存在（仅限非 OTG 器件）。

5. 通过将 USBEN 位 (U1CON<0>) 置 1，使能 USB 模块。
6. 将 OTGEN 位 (U1OTGCON<2>) 置 1 可使能 OTG 操作。
7. 通过将端点 0 的 EPRXEN 和 EPHSHK 位置 1 (U1EP0<3,0> = 1) 使端点 0 的缓冲器能接收第一个设置数据包。
8. 通过将 USBPWR 位 (U1PWRC<0>) 置 1 对 USB 模块上电。
9. 通过将 DPPULUP (U1OTGCON<7>) 置 1 使能 D+ 上拉电阻发出连接的信号。

18.4.2 在设备模式下接收 IN 令牌

1. 按 USB 2.0 规范第 9 章中的说明连接到 USB 主机并执行枚举。
2. 创建数据缓冲器，装入要发送到主机的数据。
3. 在所需端点的适当（奇数或偶数）发送 BD 中：
 - a) 用正确的数据翻转（DATA0/1）值和数据缓冲器的字节计数设置状态寄存器（BDnSTAT）。
 - b) 用数据缓冲器的起始地址设置地址寄存器（BDnADR）。
 - c) 将状态寄存器的 UOWN 位设置为 1。
4. USB 模块接收到 IN 令牌时，会自动发送该缓冲器中的数据。完成后，该模块会更新状态寄存器（BDnSTAT）并将传输完成中断标志位 TRNIF（U1IR<3>）置 1。

18.4.3 在设备模式下接收 OUT 令牌

1. 按 USB 2.0 规范第 9 章中的说明连接到 USB 主机并枚举。
2. 用您期望从主机收到的数据量创建数据缓冲器。
3. 在所需端点的适当（偶数或奇数）发送 BD 中：
 - a) 用正确的数据翻转（DATA0/1）值和数据缓冲器的字节计数设置状态寄存器（BDnSTAT）。
 - b) 用数据缓冲器的起始地址设置地址寄存器（BDnADR）。
 - c) 将状态寄存器的 UOWN 位设置为 1。
4. USB 模块接收到 OUT 令牌时，会自动接收主机发送到缓冲器的数据。完成后，该模块会更新状态寄存器（BDnSTAT）并将传输完成中断标志位 TRNIF（U1IR<3>）置 1。

18.5 主机模式的操作

下一节将说明如何执行常见的主机模式任务。在主机模式下，USB 传输由主机软件显式调用。主机软件负责传输的应答部分。同时，所有传输都是用端点 0 控制寄存器（U1EP0）和缓冲器描述符执行的。

18.5.1 使能主机模式并发现连接的设备

1. 通过将 U1CON<3>（HOSTEN）置 1 使能主机模式。这会使其他 USB OTG 寄存器中的主机模式控制位变为可用。
2. 通过将 DPPULDWN 和 DMPULDWN（U1OTGCON<5:4>）置 1 使能 D+ 和 D- 下拉电阻。通过将 DPPULUP 和 DMPULUP（U1OTGCON<7:6>）清零禁止 D+ 和 D- 上拉电阻。
3. 此时，SOF 计数器装入 12,000 开始 SOF 生成。可通过将 SOFEN 位（U1CON<0>）清零禁止 Start-Of-Frame 数据包生成来消除噪声。
4. 通过将 ATTACHIE（U1IE<6>）置 1 使能所连接器件的中断。
5. 等待所连接器件的中断（U1IR<6> = 1）。这是通过 USB 器件将 D+ 或 D- 的状态从 0 变为 1（SE0 到 J 状态）发出信号的。发生后，等待 100 ms 让器件电源稳定下来。
6. 检查 U1CON 中 JSTATE 和 SEO 位的状态。如果 JSTATE 位（U1CON<7>）为 0，则连接器件为低速。如果连接器件为低速，将 LSPDEN 和 LSPD 低位（U1ADDR<7> 和 U1EP0<7>）置 1 使能低速操作。
7. 通过将 USBRST 位（U1CON<4>）置 1 至少 50 ms 在总线上发出复位信号，将 USB 器件复位。50 ms 后，清零 USBRST 终止复位。
8. 要保持连接的器件不进入暂停状态，可通过将 SOFEN 置 1 使能 SOF 数据包生成。
9. 等待 10 ms 让器件从复位中恢复。
10. 按 USB 2.0 规范第 9 章所述执行枚举。

18.5.2 完成对所连接设备的控制事务

1. 按第 18.5.1 节“使能主机模式并发现连接的设备”中所述步骤发现设备。
2. 通过将 0Dh 写入 U1EP0 (这样会将 EPCONDIS、EPTXEN 和 EPHSHK 位置 1) 将端点控制寄存器设置为双向控制传输。
3. 将器件帧设置命令的副本放在存储器缓冲器中。关于器件帧命令集的信息，请参见 USB 2.0 规范第 9 章。
4. 为当前（偶数或奇数）发送 EP0 初始化缓冲器描述符（BD），传输某器件帧命令的 8 字节命令数据（即 GET DEVICE DESCRIPTOR）：
 - a) 将 BD 数据缓冲器地址（BD0ADR）设置为包含该命令的 8 字节存储器缓冲器的起始地址。
 - b) 将 8008h 写入 BD0STAT（将 UOWN 位置 1，设置字节计数为 8）。
5. 在地址寄存器（U1ADDR<6:0>）中设置目标器件的 USB 器件地址。USB 总线复位后，器件 USB 地址将为 0。枚举后，它将设置为 1 到 127 之间的另一个值。
6. 将 D0h 写入 U1TOK；这是目标器件默认控制管道端点 0 的 SETUP 令牌。该操作将在总线上启动 SETUP 令牌，后面跟着数据包。数据包完成后，在 BD0STAT 的 PID 字段中返回器件握手。USB 模块更新 BD0STAT 时，将产生一个传输完成中断（TRNIF 标志位置 1）。该操作将完成设置事务的设置阶段，如 USB 规范第 9 章中所述。
7. 为启动设置事务的数据阶段（即获取 GET DEVICE 描述符命令所需数据），要在存储器中设置缓冲器存储接收到的数据。
8. 初始化当前（偶数或奇数）接收或发送（对于 IN 是接收，对于 OUT 是发送）EP0 BD，以传输数据。
 - a) 将 C040h 写入 BD0STAT。该操作将把 UOWN 置 1，将数据翻转（DTS）配置为 DATA1，并将字节计数设置为数据缓冲器的长度（在本例中是 64 或 40h）。
 - b) 将 BD0ADR 设置为数据缓冲器的起始地址。
9. 用对于目标器件的默认控制管道端点 0 适当的 IN 或 OUT 令牌写入令牌寄存器（例如，对于 GET DEVICE DESCRIPTOR 命令的 IN 令牌，将 90h 写入 U1TOK）。该操作将在总线上启动 IN 令牌，后面跟着从器件到主机的数据包。数据包完成时，将写入 BD0STAT 并产生一个传输完成中断（TRNIF 标志位置 1）。对于单数据包数据阶段的控制传输，该操作将完成设置事务的数据阶段，如 USB 规范第 9 章中所述。如果需要传输更多数据，请返回步骤 8。
10. 要启动设置事务的状态阶段，要在存储器中设置缓冲器，以接收或发送零长度状态阶段数据包。
11. 初始化当前（偶数或奇数）发送 EP0 BD 以传输状态数据：
 - a) 将 BDT 缓冲器地址字段设置为数据缓冲器的起始地址。
 - b) 将 8000h 写入 BD0STAT（将 UOWN 位置 1，将 DTS 配置为 DATA0，并将字节计数设置为 0）。
12. 用对于目标器件的默认控制管道端点 0 适当的 IN 或 OUT 令牌写入令牌寄存器（例如，对于 GET DEVICE DESCRIPTOR 命令的 OUT 令牌，将 01h 写入 U1TOK）。该操作将在总线上启动 OUT 令牌，后面跟着从主机到器件的零长度数据包。数据包完成时，将用来自器件的握手更新 BD，并产生一个传输完成中断（TRNIF 标志位置 1）。该操作将完成设置事务的状态阶段，如 USB 规范第 9 章中所述。

注： 对于每个帧，只能执行一个控制事务。

18.5.3 向目标器件发送全速批量数据传输

- 按第 18.5.1 节“使能主机模式并发现连接的设备”和第 18.5.2 节“完成对所连接设备的控制事务”中所述步骤发现并配置器件。
- 要在使能握手时使能发送和接收传输，将 1Dh 写入 U1EP0。如果目标器件是低速器件，还要将 LSPD 位 (U1EP0<7>) 置 1。如果希望硬件在目标器件在传输中产生 NAK 时无限期自动重试，可清零重试禁止位 RETRYDIS (U1EP0<6>)。
- 设置当前（偶数或奇数）发送 EP0 的 BD，以传输最多 64 字节。
- 在地址寄存器 (U1ADDR<6:0>) 中设置目标器件的 USB 器件地址。
- 将所需端点的 OUT 令牌写入 U1TOK。该操作会触发该模块的发送状态机，开始发送令牌和数据。
- 等待传输完成中断标志位 TRNIF 置 1。这表示 BD 所有权已释放给微处理器，传输已完成。如果重试禁止位置 1，将在 BD PID 字段中返回握手 (ACK、NAK、STALL 或 ERROR (0Fh))。如果发生 STALL 中断，等待中的数据包必须离队，并清除目标器件中的错误状态。如果发生断开连接中断 (SE0 超过 2.5 μs)，则目标已断开连接 (U1IR<0> 置 1)。
- 发生传输完成中断 (TRNIF 置 1) 时，可返回步骤 2 检查 BD 并将下一个数据包入队。

注： USB 速度、收发器和上拉只应在模块设置阶段配置。模块使能时，建议不要改变这些设置。

18.6 OTG 工作原理

18.6.1 会话请求协议 (SRP)

OTG A 器件可在未使用 USB 连接时，通过会话请求协议 (Session Request Protocol, SRP) 关闭 VBUS 供电。软件可通过清零 VBUSON (U1OTGCON<3>) 完成该操作。VBUS 供电关闭时，说明 A 器件已结束 USB 会话。

任何时候 OTG A 器件或嵌入式主机都可以重新对 VBUS 供电（启动新的会话）。OTG B 器件也可以请求 OTG A 器件对 VBUS 重新供电（启动新会话）。这是通过会话请求协议 (SRP) 实现的。

请求新会话前，B 器件必须先检查确定上个会话已结束。为此，B 器件必须检查两个条件：

- VBUS 供电低于会话有效电压，以及
- D+ 和 D- 都已为低电平至少 2 ms。

条件 1 会通过 SESENDIF (U1OTGIR<2>) 中断通知 B 器件。软件必须手动检查条件 2。

注： A 器件关闭 VBUS 电源时，B 器件必须断开上拉电阻与电源的连接。如果器件是自身供电的，可通过清零 DPPULUP (U1OTGCON<7>) 和 DMPULUP (U1OTGCON<6>) 完成该操作。

B 器件可通过电阻将 VBUS 电源放电，来帮助达到条件 1。软件可通过将 VBUSDIS (U1OTGCON<0>) 置 1 完成该操作。

满足这些初始条件后，B 器件可开始请求新会话。B 器件通过向 D+ 数据线发出脉冲开始请求。软件应通过将 DPPULUP (U1OTGCON<7>) 置 1 完成该操作。数据线应保持高电平 5 到 10 ms。

随后 B 器件对 VBUS 电源发出脉冲。软件应通过将 PUVBUS (U1CNFG2<4>) 置 1 完成该操作。A 器件检测到 SRP 信号（通过 ATTACHIF (U1IR<6>) 中断或 SESVDIF (U1OTGIR<3>) 中断）时，A 器件必须通过将 VBUSON (U1OTGCON<3>) 置 1 或将控制外部电源的 I/O 端口置为全 1 来恢复 VBUS 供电。

B 器件执行 VBUS 电源脉冲时不会监视 VBUS 电源的状态。B 器件确实检测到 VBUS 电源已恢复（通过 SESVDIF (U1OTGIR<3>) 中断）时，B 器件必须通过上拉 D+ 或 D-（通过 DPPULUP 或 DMPULUP）重新连接到 USB 链路。

A 器件必须通过驱动 USB 复位信号完成 SRP。

18.6.2 主机协商协议 (HNP)

在 USB OTG 应用中，双角色器件 (Dual Role Device, DRD) 是既能作为主机又能作为外设的器件。OTG DRD 必须支持主机协商协议 (HNP)。

HNP 允许 OTG B 器件临时作为 USB 主机。A 器件必须首先使能 B 器件，以遵循 HNP。关于 HNP 的更多信息，请参见“USB 2.0 规范 On-The-Go 补充信息”。HNP 只能在全速状态下启动。

由 A 器件使能 HNP 后，B 器件只需表示断开连接即可请求在 USB 连接处于暂停状态的任何时刻成为主机。这可以通过清零 DPPULUP 和 DMPULUP 在软件中完成。A 器件检测到断开连接条件（通过 URSTIF (U1IR<0>) 中断）时，A 器件可以允许 B 器件作为主机接管。A 器件是通过发出全速连接信号完成该操作的。软件可通过将 DPPULUP 置 1 实现该操作。

如果 A 器件以恢复信号响应，则 A 器件保持为主机。B 器件检测到连接条件（通过 ATTACHIF (U1IR<6>)）时，B 器件变为主机。B 器件在使用总线前驱动复位信号。

B 器件完成主机角色后，停止所有总线活动，将 DPPULUP 置 1 使能其 D+ 上拉电阻。A 器件检测到暂停条件（空闲 3 ms）时，A 器件将关闭其 D+ 上拉电阻。A 器件也可以关闭 VBUS 供电来结束该会话。A 器件检测到连接条件（通过 ATTACHIF）时，A 器件会恢复主机操作，并驱动复位信号。

18.7 USB OTG 模块寄存器

共有 37 个存储器映射寄存器与 USB OTG 模块关联。它们可分为四大类：

- USB OTG 模块控制 (12)
- USB 中断 (7)
- USB 端点管理 (16)
- USB VBUS 电源控制 (2)

该总数不包括 BDT 中的（最多）128 个 BD 寄存器。它们的原型如寄存器 18-1 和寄存器 18-2 中所述，分别显示在第 18.2 节“USB 缓冲器描述符和 BDT”中。

对于例外的 U1PWMCON 和 U1PWMRSS，所有 USB OTG 寄存器都实现在寄存器的低字节。高位字节未实现，无功能。请注意，部分寄存器只在主机模式下有意义，而其他寄存器在器件和主机模式下有不同的位含义。

以下章节中所述寄存器包含具有特定控制和配置功能的位。以下寄存器只用于数据或地址值：

- **U1BDTP1**: 在数据 RAM 中指定 256 字的页用于 BDT；8 位值 (bit 0 固定为 0) 用于边界对齐
- **U1FRML** 和 **U1FRMH**: 包含用于当前数据帧的 11 位字节计数器
- **U1PWMRSS** 包含用于 VBUS 升压辅助 PWM 模块的表示 PWM 占空比 (bit<15:8>) 和 PWM 周期 (bit<7:0>) 的 8 位值。

PIC24FJ256GB110 系列

18.7.1 USB OTG 模块控制寄存器

寄存器 18-3: U1OTGSTAT: USB OTG 状态寄存器 (仅限主机模式)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R-0, HSC	U-0	R-0, HSC	U-0	R-0, HSC	R-0, HSC	U-0	R-0, HSC
ID	—	LSTATE	—	SESVD	SESEND	—	VBUSVD
bit 7							bit 0

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

- bit 15-8 **未实现:** 读为 0
- bit 7 **ID:** ID 引脚状态指示位
1 = 未连接插头, 或者将 B 类电缆插入了 USB 插座
0 = 一个 A 类插头已插入 USB 插座
- bit 6 **未实现:** 读为 0
- bit 5 **LSTATE:** 线状态稳定指示位
1 = 上 1 ms USB 线状态 (由 SEO 和 JSTATE 定义) 已稳定
0 = 上 1 ms USB 线状态未稳定
- bit 4 **未实现:** 读为 0
- bit 3 **SESVD:** 会话有效指示位
1 = VBUS 电压高于 A 或 B 器件上的 VA_SESS_VLD (如 USB OTG 规范中所定义)
0 = VBUS 电压低于 A 或 B 器件上的 VA_SESS_VLD
- bit 2 **SESEND:** B 会话结束指示位
1 = VBUS 电压低于 B 器件上的 VB_SESS_END (如 USB OTG 规范中所定义)
0 = VBUS 电压高于 B 器件上的 VB_SESS_END
- bit 1 **未实现:** 读为 0
- bit 0 **VBUSVD:** A-VBUS 有效指示位
1 = VBUS 电压高于 A 器件上的 VA_VBUS_VLD (如 USB OTG 规范中所定义)
0 = VBUS 电压低于 A 器件上的 VA_VBUS_VLD

寄存器 18-4: U1OTGCON: USB On-the-Go 控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPPULUP	DMPULUP	DPPULDWN ⁽¹⁾	DMPULDWN ⁽¹⁾	VBUSON ⁽¹⁾	OTGEN ⁽¹⁾	VBUSCHG ⁽¹⁾	VBUSDIS ⁽¹⁾
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7 **DPPULUP:** D+ 上拉使能位

1 = 使能 D+ 数据线上拉电阻
0 = 禁止 D+ 数据线上拉电阻

bit 6 **DMPULUP:** D- 上拉使能位

1 = 使能 D- 数据线上拉电阻
0 = 禁止 D- 数据线上拉电阻

bit 5 **DPPULDWN:** D+ 下拉使能位⁽¹⁾

1 = 使能 D+ 数据线下拉电阻
0 = 禁止 D+ 数据线下拉电阻

bit 4 **DMPULDWN:** D- 下拉使能位⁽¹⁾

1 = 使能 D- 数据线下拉电阻
0 = 禁止 D- 数据线下拉电阻

bit 3 **VBUSON:** VBUS 上电位⁽¹⁾

1 = VBUS 线上电
0 = VBUS 线未上电

bit 2 **OTGEN:** OTG 功能使能位⁽¹⁾

1 = 使能 USB OTG ; 使能所有 D+/D- 上拉和下拉位
0 = 禁止 USB OTG ; D+/D- 上拉和下拉电路通过设置 HOSTEN 和 USBEN 位 (U1CON<3,0>) 在硬件中控制

bit 1 **VBUSCHG:** VBUS 充电选择位⁽¹⁾

1 = VBUS 线设置为充电到 3.3V
0 = VBUS 线设置为充电到 5V

bit 0 **VBUSDIS:** VBUS 放电使能位⁽¹⁾

1 = VBUS 线通过电阻放电
0 = VBUS 线不放电

注 1: 这些位仅在主机模式下使用; 不要在器件模式下使用。

PIC24FJ256GB110 系列

寄存器 18-5: U1PWRC: USB 电源控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0, HS	U-0	U-0	R/W-0	U-0	U-0	R/W-0, HC	R/W-0
UACTPND	—	—	USLPGRD	—	—	USUSPND	USBPWR
bit 7							bit 0

图注:

HS = 硬件置 1 位

HC = 硬件清零位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7 **UACTPND: USB 活动暂停位**

1 = 此刻不应暂停模块 (要求 USLPGRD 位置 1)

0 = 模块可能暂停或掉电

bit 6-5 未实现: 读为 0

bit 4 **USLPGRD: 休眠 / 暂停保护位**

1 = 通知 USB 模块即将暂停或掉电

0 = 不暂停

bit 3-2 未实现: 读为 0

bit 1 **USUSPND: USB 暂停模式使能位**

1 = USB OTG 模块处于暂停模式; USB 时钟是门控的, 并且收发器置于低功耗状态

0 = 常规 USB OTG 操作

bit 0 **USBPWR: USB 操作使能位**

1 = 使能 USB OTG 模块

0 = 禁止 USB OTG 模块⁽¹⁾

注 1: 不要清零该位, 除非 HOSTEN、USBEN 和 OTGEN 位 (U1CON<3,0> 和 U1OTGCON<2>) 都已清零。

寄存器 18-6: U1STAT: USB 状态寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R-0, HSC	U-0	U-0					
ENDPT3	ENDPT2	ENDPT1	ENDPT0	DIR	PPBI ⁽¹⁾	—	—
bit 7							bit 0

图注:

U = 未实现位, 读为 0

R = 可读位

W = 可写位

HSC = 硬件置 1/ 清零位

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7-4 **ENDPT<3:0>**: 上个端点活动编号位 (代表上个 USB 传输更新的 BDT 号)

1111 = 端点 15

1110 = 端点 14

....

0001 = 端点 1

0000 = 端点 0

bit 3 **DIR**: 上次 BD 方向指示位

1 = 上次事务是发送传输 (Tx)

0 = 上次事务是接收传输 (Rx)

bit 2 **PPBI**: 乒乓 BD 指针指示位⁽¹⁾

1 = 上次事务针对奇数 BD 存储区

0 = 上次事务针对偶数 BD 存储区

bit 1-0 未实现: 读为 0

注 1: 该位仅对具有可用的偶数和奇数 BD 寄存器的端点有效。

PIC24FJ256GB110 系列

寄存器 18-7: U1CON: USB 控制寄存器 (器件模式)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							

U-0	R-x, HSC	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SE0	PKTDIS	—	HOSTEN	RESUME	PPBRST	USBEN
bit 7							

图注:

U = 未实现位, 读为 0

R = 可读位

W = 可写位

HSC = 硬件置 1/ 清零位

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-7 未实现: 读为 0

bit 6 **SE0:** 有效单端 0 标志位

1 = 在 USB 总线上存在有效的单端 0

0 = 没有检测到单端 0

bit 5 **PKTDIS:** 数据包传输禁止位

1 = 禁止处理 SIE 令牌和数据包; 接收到 SETUP 令牌时自动置 1

0 = 使能 SIE 令牌和数据包处理

bit 4 未实现: 读为 0

bit 3 **HOSTEN:** 主机模式使能位

1 = 使能 USB 主机功能; D+ 和 D- 上的下拉电路由硬件激活

0 = 禁止 USB 主机功能

bit 2 **RESUME:** 恢复信号使能位

1 = 激活恢复信号

0 = 禁止恢复信号

bit 1 **PPBRST:** 乒乓缓冲器复位位

1 = 将所有乒乓缓冲器指针复位到偶数 BD 存储区

0 = 乒乓缓冲器指针不复位

bit 0 **USBEN:** USB 模块使能位

1 = 使能 USB 模块和支持电路 (器件已连接); D+ 上拉由硬件激活

0 = 禁止 USB 模块和支持电路 (器件已断开连接)

寄存器 18-8: U1CON: USB 控制寄存器 (仅限主机模式)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R-x, HSC	R-x, HSC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
JSTATE	SE0	TOKBUSY	USBRST	HOSTEN	RESUME	PPBRST	SOFEN
bit 7							bit 0

图注:

U = 未实现位, 读为 0

R = 可读位

W = 可写位

HSC = 硬件置 1/ 清零位

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7 **JSTATE:** 有效差分接收机 J 状态标志位

1 = USB 上检测到 J 状态 (低速模式下差分 0, 全速模式下差分 1)

0 = 未检测到 J 状态

bit 6 **SE0:** 有效单端 0 标志位

1 = 在 USB 总线上存在有效的单端 0

0 = 没有检测到单端 0

bit 5 **TOKBUSY:** 令牌忙状态位

1 = USB 模块在 On-The-Go 状态中执行令牌

0 = 未执行令牌

bit 4 **USBRST:** 模块复位位

1 = 产生了 USB 复位; 对于软件复位, 应用程序必须置 1 该位 50 ms 然后清零

0 = USB 复位已终止

bit 3 **HOSTEN:** 主机模式使能位

1 = 使能 USB 主机功能; D+ 和 D- 上的下拉电路由硬件激活

0 = 禁止 USB 主机功能

bit 2 **RESUME:** 恢复信号使能位

1 = 恢复信号已激活; 软件必须置 1 该位 10 ms 然后清零, 来使能远程唤醒

0 = 禁止恢复信号

bit 1 **PPBRST:** 乒乓缓冲器复位位

1 = 将所有乒乓缓冲器指针复位到偶数 BD 存储区

0 = 乒乓缓冲器指针不复位

bit 0 **SOFEN:** Start-Of-Frame 使能位

1 = 每隔 1 毫秒发送 Start-Of-Frame 令牌

0 = 禁止 Start-Of-Frame 令牌

PIC24FJ256GB110 系列

寄存器 18-9: U1ADDR: USB 地址寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LSPDEN ⁽¹⁾	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
bit 7							bit 0

图注:

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

- bit 15-8 未实现: 读为 0
bit 7 **LSPDEN:** 低速使能指示位 ⁽¹⁾
 1 = USB 模块工作于低速模式
 0 = USB 模块工作于全速模式
bit 6-0 **ADDR<6:0>:** USB 器件地址位

注 1: 仅限主机模式。在器件模式下, 该位未实现且读为 0。

寄存器 18-10: U1TOK: USB 令牌寄存器 (仅限主机模式)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| PID3 | PID2 | PID1 | PID0 | EP3 | EP2 | EP1 | EP0 |
| bit 7 | | | | | | | bit 0 |

图注:

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

- bit 15-8 未实现: 读为 0
bit 7-4 **PID<3:0>:** 令牌类型标识符
 1101 = SETUP (TX) 令牌类型事务 ⁽¹⁾
 1001 = IN (RX) 令牌类型事务 ⁽¹⁾
 0001 = OUT (TX) 令牌类型事务 ⁽¹⁾
bit 3-0 **EP<3:0>:** 令牌命令端点地址位
 该值必须指定所连接器件上的有效端点。

注 1: 所有其他组合都被保留不用。

寄存器 18-11: U1SOF: USB OTG Start-Of-Frame 门限值寄存器 (仅限主机模式)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CNT7 | CNT6 | CNT5 | CNT4 | CNT3 | CNT2 | CNT1 | CNT0 |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7-0 **CNT<7:0>**: Start-Of-Frame 大小位: 值表示 $10 + (n$ 字节的数据包大小)。例如:

0100 1010 = 64 字节数据包

0010 1010 = 32 字节数据包

0001 0010 = 8 字节数据包

寄存器 18-12: U1CNFG1: USB 配置寄存器 1

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
UTEYE	UOEMON ⁽¹⁾	—	USBSIDL	—	—	PPB1	PPB0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7 **UTEYE**: USB 眼图测试使能位

1 = 使能眼图测试

0 = 禁止眼图测试

bit 6 **UOEMON**: USB OE 监视器使能位⁽¹⁾

1 = \overline{OE} 信号有效; 它指示驱动 D+/D- 线的间隔

0 = OE 信号无效

bit 5 未实现: 读为 0

bit 4 **USBSIDL**: USB OTG 空闲模式停止位

1 = 当器件进入空闲模式时, 模块停止工作

0 = 在空闲模式下模块继续工作

bit 3-2 未实现: 读为 0

bit 1-0 **PPB<1:0>**: 乒乓缓冲器配置位

11 = 使能端点 1 到 15 的奇 / 偶乒乓缓冲器

10 = 使能所有端点的奇 / 偶乒乓缓冲器

01 = 使能 OUT 端点 0 的奇 / 偶乒乓缓冲器

00 = 禁止奇 / 偶乒乓缓冲器

注 1: 该位仅在 UTRDIS 位 (U1CNFG2<0>) 置 1 时有效。

PIC24FJ256GB110 系列

寄存器 18-13: U1CNFG2: USB 配置寄存器 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	PUVBUS	EXTI2CEN	UVBUSDIS ⁽¹⁾	UVCMPDIS ⁽¹⁾	UTRDIS ⁽¹⁾
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-5 未实现: 读为 0

bit 4 **PUVBUS:** VBUS 上拉使能位

1 = 使能 VBUS 引脚上的上拉
0 = 禁止 VBUS 引脚上的上拉

bit 3 **EXTI2CEN:** 外部模块的 I²CTM 接口控制使能位

1 = 通过 I²C 接口控制外部模块
0 = 通过专用引脚控制外部模块

bit 2 **UVBUSDIS:** 片上 5V 升压稳压器构建器禁止位⁽¹⁾

1 = 禁止片上升压稳压器构建器; 使能数字输出控制接口
0 = 片上升压稳压器构建器有效

bit 1 **UVCMPDIS:** 片上 VBUS 比较器禁止位⁽¹⁾

1 = 禁止片上充电 VBUS 比较器; 使能数字输入状态接口
0 = 片上充电 VBUS 比较器有效

bit 0 **UTRDIS:** 片上收发器禁止位⁽¹⁾

1 = 禁止片上收发器; 使能数字收发器接口
0 = 片上收发器有效

注 1: 当 USBPWR 位置 1 (U1PWRC<0> = 1) 时, 绝不要更改这些位。

18.7.2 USB 中断寄存器

寄存器 18-14: U1OTGIR: USB OTG 中断状态寄存器（仅限主机模式）

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/K-0, HS	U-0	R/K-0, HS					
IDIF	T1MSECIF	LSTATEIF	ACTVIF	SESVdif	SESENDIF	—	VBUSVDIF
bit 7							bit 0

图注:

U = 未实现位, 读为 0

R = 可读位

K = 写入 1 清零该位

HS = 硬件置 1 位

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7 **IDIF:** ID 状态变化指示位

1 = 检测到 ID 状态变化

0 = ID 状态不变

bit 6 **T1MSECIF:** 1 毫秒定时器位

1 = 1 毫秒定时器已到期

0 = 1 毫秒定时器未到期

bit 5 **LSTATEIF:** 线状态稳定指示位

1 = USB 线状态（如 SEO 和 JSTATE 位所定义）已稳定 1 ms，但和上次不同

0 = USB 线状态未稳定达 1 ms

bit 4 **ACTVIF:** 总线活动指示位

1 = D+/D- 线或 VBUS 上检测到活动

0 = D+/D- 线或 VBUS 上没有检测到活动

bit 3 **SESVdif:** 会话有效改变指示位

1 = VBUS 已超过 VA_SESS_END（如 USB OTG 规范中所定义）⁽¹⁾

0 = VBUS 未超过 VA_SESS_END

bit 2 **SESENDIF:** B 器件 VBUS 改变指示位

1 = 检测到 B 器件上 VBUS 改变; VBUS 已超过 VB_SESS_END（如 USB OTG 规范中所定义）⁽¹⁾

0 = VBUS 未超过 VA_SESS_END

bit 1 未实现: 读为 0

bit 0 **VBUSVDIF:** A 器件 VBUS 改变指示位

1 = 检测到 A 器件上 VBUS 改变; VBUS 已超过 VA_VBUS_VLD（如 USB OTG 规范中所定义）⁽¹⁾

0 = 未检测到 A 器件上 VBUS 改变

注 1: 穿越 VBUS 门限值的事件可以发生在上升沿也可以发生在下降沿。

注: 单个位只能通过在对整个寄存器执行字写操作时, 作为该操作的一部分向该位写入 1 来清零。用布尔指令或位操作写入单个位, 会将写入时所有置 1 的位变为清零。

PIC24FJ256GB110 系列

寄存器 18-15: U1OTGIE: USB OTG 中断允许寄存器 (仅限主机模式)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
IDIE	T1MSECIE	LSTATEIE	ACTVIE	SESVDIE	SESENDIE	—	VBUSVDIE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7 **IDIE:** ID 中断允许位

1 = 允许中断

0 = 禁止中断

bit 6 **T1MSECIE:** 1 毫秒定时器中断允许位

1 = 允许中断

0 = 禁止中断

bit 5 **LSTATEIE:** 线状态稳定中断允许位

1 = 允许中断

0 = 禁止中断

bit 4 **ACTVIE:** 总线活动中断允许位

1 = 允许中断

0 = 禁止中断

bit 3 **SESVDIE:** 会话有效中断允许位

1 = 允许中断

0 = 禁止中断

bit 2 **SESENDIE:** B 器件会话结束中断允许位

1 = 允许中断

0 = 禁止中断

bit 1 未实现: 读为 0

bit 0 **VBUSVDIE:** A 器件 VBUS 有效中断允许位

1 = 允许中断

0 = 禁止中断

寄存器 18-16: U1IR: USB 中断状态寄存器 (仅限器件模式)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/K-0, HS	U-0	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R-0	R/K-0, HS
STALLIF	—	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	URSTIF
bit 7							bit 0

图注:

U = 未实现位, 读为 0

R = 可读位

K = 写入 1 清零该位

HS = 硬件置 1 位

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8	未实现: 读为 0
bit 7	STALLIF: STALL 握手中断位 1 = 器件模式下事务的握手阶段中, 外设发送 STALL 握手 0 = 未发送 STALL 握手
bit 6	未实现: 读为 0
bit 5	RESUMEIF: 恢复中断位 1 = D+ 或 D- 引脚上观察到 K 状态 $2.5 \mu\text{s}$ (低速模式下差分 1, 全速模式下差分 0) 0 = 未观察到 K 状态
bit 4	IDLEIF: 空闲检测中断位 1 = 检测到空闲状态 (3 ms 或更长的连续空闲状态) 0 = 未检测到空闲状态
bit 3	TRNIF: 令牌处理完成中断位 1 = 处理完当前令牌; 从 U1STAT 寄存器读取端点信息 0 = 未处理完当前令牌; 清零 U1STAT 寄存器或从 STAT 装入下个令牌 (清零该位会使 STAT FIFO 递增)
bit 2	SOFIF: Start-Of-Frame 令牌中断位 1 = 外设接收到 Start-Of-Frame 令牌, 或主机达到 Start-Of-Frame 门限值 0 = 未接收到 Start-Of-Frame 令牌或未达到门限值
bit 1	UERRIF: USB 错误条件中断位 (只读) 1 = 发生了未屏蔽的错误条件; 只有 U1EIE 寄存器中使能的错误状态才能将该位置 1 0 = 未发生未屏蔽的错误条件
bit 0	URSTIF: USB 复位中断位 1 = 有效的 USB 复位已发生至少 $2.5 \mu\text{s}$; 必须清除复位状态才能重新拉低该位 0 = 没有发生 USB 复位。单个位只能通过在对整个寄存器执行字写操作时, 作为该操作的一部分向该位写入 1 来清零。用布尔指令或位操作写入单个位, 会将写入时所有置 1 的位变为清零。

注: 单个位只能通过在对整个寄存器执行字写操作时, 作为该操作的一部分向该位写入 1 来清零。用布尔指令或位操作写入单个位, 会将写入时所有置 1 的位变为清零。

PIC24FJ256GB110 系列

寄存器 18-17: U1IR: USB 中断状态寄存器 (仅限主机模式)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/K-0, HS	R-0	R/K-0, HS					
STALLIF	ATTACHIF	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	DETACHIF
bit 7							bit 0

图注:	U = 未实现位, 读为 0
R = 可读位	K = 写入 1 清零该位
-n = POR 时的值	HS = 硬件置 1 位
1 = 置 1	0 = 清零
	x = 未知

bit 15-8	未实现: 读为 0
bit 7	STALLIF: STALL 握手中断位 1 = 器件模式下事务的握手阶段中, 外设发送 STALL 握手 0 = 未发送 STALL 握手
bit 6	ATTACHIF: 外设连接中断位 1 = 模块已检测到外设连接; 如果总线状态不是 SE0, 并且 2.5 μs 内无总线活动, 则置 1 0 = 未检测到外设连接
bit 5	RESUMEIF: 恢复中断位 1 = D+ 或 D- 引脚上观察到 K 状态 2.5 μs (低速模式下差分 1, 全速模式下差分 0) 0 = 未观察到 K 状态
bit 4	IDLEIF: 空闲检测中断位 1 = 检测到空闲状态 (3 ms 或更长的连续空闲状态) 0 = 未检测到空闲状态
bit 3	TRNIF: 令牌处理完成中断位 1 = 处理完当前令牌; 从 U1STAT 寄存器读取端点信息 0 = 未处理完当前令牌; 清零 U1STAT 寄存器或从 U1STAT 装入下个令牌
bit 2	SOFIF: Start-Of-Frame 令牌中断位 1 = 外设接收到 Start-Of-Frame 令牌, 或主机达到 Start-Of-Frame 门限值 0 = 未接收到 Start-Of-Frame 令牌或未达到门限值
bit 1	UERRIF: USB 错误条件中断位 1 = 发生了未屏蔽的错误条件; 只有 U1EIE 寄存器中使能的错误状态才能将该位置 1 0 = 未发生未屏蔽的错误条件
bit 0	DETACHIF: 断开连接中断位 1 = 模块检测到外设断开连接; 必须清除复位状态才能重新拉低该位 0 = 未检测到外设断开连接。单个位只能通过在对整个寄存器执行字写操作时, 作为该操作的一部分向该位写入 1 来清零。用布尔指令或位操作写入单个位, 会将写入时所有置 1 的位变为清零。

注:	单个位只能通过在对整个寄存器执行字写操作时, 作为该操作的一部分向该位写入 1 来清零。用布尔指令或位操作写入单个位, 会将写入时所有置 1 的位变为清零。
----	--

寄存器 18-18: U1IE: USB 中断允许寄存器 (所有 USB 模式)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STALLIE	ATTACHIE ⁽¹⁾	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	URSTIE DETACHIE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- | | |
|----------|---|
| bit 15-8 | 未实现: 读为 0 |
| bit 7 | STALLIE: STALL 握手中断允许位 |
| | 1 = 允许中断 |
| | 0 = 禁止中断 |
| bit 6 | ATTACHIE: 外设连接中断位 (仅限主机模式) ⁽¹⁾ |
| | 1 = 允许中断 |
| | 0 = 禁止中断 |
| bit 5 | RESUMEIE: 恢复中断位 |
| | 1 = 允许中断 |
| | 0 = 禁止中断 |
| bit 4 | IDLEIE: 空闲检测中断位 |
| | 1 = 允许中断 |
| | 0 = 禁止中断 |
| bit 3 | TRNIE: 令牌处理完成中断位 |
| | 1 = 允许中断 |
| | 0 = 禁止中断 |
| bit 2 | SOFIE: Start-of-Frame 令牌中断位 |
| | 1 = 允许中断 |
| | 0 = 禁止中断 |
| bit 1 | UERRIE: USB 错误条件中断位 |
| | 1 = 允许中断 |
| | 0 = 禁止中断 |
| bit 0 | URSTIE 或 DETACHIE: USB 复位中断 (器件模式) 或 USB 断开连接中断 (主机模式) 使能位 |
| | 1 = 允许中断 |
| | 0 = 禁止中断 |

注 1: 在器件模式下未实现, 读为 0。

PIC24FJ256GB110 系列

寄存器 18-19: U1EIR: USB 错误中断状态寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15				bit 8			

R/K-0, HS	U-0	R/K-0, HS	R/K-0, HS				
BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	CRC5EF EOFEF	PIDEF
bit 7				bit 0			

图注:

U = 未实现位, 读为 0

R = 可读位

K = 写入 1 清零该位

HS = 硬件置 1 位

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7 **BTSEF:** 位填充错误标志位

1 = 检测到位填充错误

0 = 无位填充错误

bit 6 未实现: 读为 0

bit 5 **DMAEF:** DMA 错误标志位

1 = 检测到USB DMA 错误条件; BD字节计数字段指示的数据大小小于接收到的字节数。接收到的数据被截短。

0 = 无 DMA 错误

bit 4 **BTOEF:** 总线周转 (Turnaround) 超时错误标志位

1 = 发生总线周转超时

0 = 未发生总线周转超时

bit 3 **DFN8EF:** 数据字段大小错误标志位

1 = 数据字段的字节数不是整数

0 = 数据字段的字节数是整数

bit 2 **CRC16EF:** CRC16 失败标志位

1 = CRC16 失败

0 = CRC16 通过

bit 1 对于器件模式:

CRC5EF: CRC5 主机错误标志位

1 = 令牌数据包由于 CRC5 错误而被拒绝

0 = 令牌数据包被接受 (无 CRC5 错误)

对于主机模式:

EOFEF: End-Of-Frame 错误标志位

1 = 发生 End-Of-Frame 错误

0 = 禁止 End-Of-Frame 中断

bit 0 **PIDEF:** PID 检查失败标志位

1 = PID 检查失败

0 = PID 检查通过。单个位只能通过在对整个寄存器执行字写操作时, 作为该操作的一部分向该位写入 1 来清零。用布尔指令或位操作写入单个位, 会将写入时所有置 1 的位变为清零。

注: 单个位只能通过在对整个寄存器执行字写操作时, 作为该操作的一部分向该位写入 1 来清零。用布尔指令或位操作写入单个位, 会将写入时所有置 1 的位变为清零。

寄存器 18-20: U1EIE: USB 错误中断允许寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	CRC5EE EOFEE	PIDEE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-8 未实现: 读为 0
- bit 7 **BTSEE:** 位填充错误中断允许位
 1 = 允许中断
 0 = 禁止中断
- bit 6 未实现: 读为 0
- bit 5 **DMAEE:** DMA 错误中断允许位
 1 = 允许中断
 0 = 禁止中断
- bit 4 **BTOEE:** 总线周转超时错误中断允许位
 1 = 允许中断
 0 = 禁止中断
- bit 3 **DFN8EE:** 数据字段大小错误中断允许位
 1 = 允许中断
 0 = 禁止中断
- bit 2 **CRC16EE:** CRC16 失败中断允许位
 1 = 允许中断
 0 = 禁止中断
- bit 1 对于器件模式:
CRC5EE: CRC5 主机错误中断允许位
 1 = 允许中断
 0 = 禁止中断
对于主机模式:
EOFEE: End-of-Frame 错误中断允许位
 1 = 允许中断
 0 = 禁止中断
- bit 0 **PIDEE:** PID 检查失败中断允许位
 1 = 允许中断
 0 = 禁止中断

PIC24FJ256GB110 系列

18.7.3 USB 端点管理寄存器

寄存器 18-21: U1EPn: USB 端点控制寄存器 (n = 0 到 15)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LSPD ⁽¹⁾	RETRYDIS ⁽¹⁾	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7 **LSPD:** 低速直接连接使能位 (仅限 U1EP0) ⁽¹⁾

1 = 使能直接连接到低速器件

0 = 禁止直接连接到低速器件

bit 6 **RETRYDIS:** 重试禁止位 (仅限 U1EP0) ⁽¹⁾

1 = 禁止重试 NAK 事务

0 = 使能重试 NAK 事务; 由硬件完成重试

bit 5 未实现: 读为 0

bit 4 **EPCONDIS:** 双向端点控制位

如果 EPTXEN 和 EPRXEN = 1:

1 = 禁止端点 n 的控制传输; 只允许发送和接收传输

0 = 使能端点 n 的控制 (SETUP) 传输; 同时允许发送和接收传输

对于 EPTXEN 和 EPRXEN 的所有其他组合:

该位为无关位。

bit 3 **EPRXEN:** 端点接收使能位

1 = 使能端点 n 接收

0 = 禁止端点 n 接收

bit 2 **EPTXEN:** 端点发送使能位

1 = 使能端点 n 发送

0 = 禁止端点 n 发送

bit 1 **EPSTALL:** 端点停止状态位

1 = 端点 n 已停止

0 = 端点 n 未停止

bit 0 **EPHSHK:** 端点握手使能位

1 = 使能端点握手

0 = 禁止端点握手 (通常用于同步端点)

注 1: 这些位仅在主机模式下对 U1EP0 可用。对于所有其他 U1EPn 寄存器, 这些位始终未实现且读为 0。

18.7.4 USB VBUS 电源控制寄存器

寄存器 18-22: U1PWMCON: USB VBUS PWM 发生器控制寄存器

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
PWMEN	—	—	—	—	—	PWMPOL	CNTEN
bit 15							

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **PWMEN:** PWM 使能位

1 = 使能 PWM 发生器

0 = 禁止 PWM 发生器; 输出保存在由 PWMPOL 指定的复位状态

bit 14-10 未实现: 读为 0

bit 9 **PWMPOL:** PWM 极性位

1 = PWM 输出低电平有效, 其复位状态为高电平

0 = PWM 输出高电平有效, 其复位状态为低电平

bit 8 **CNTEN:** PWM 计数器使能位

1 = 使能计数器

0 = 禁止计数器

bit 7-0 未实现: 读为 0

PIC24FJ256GB110 系列

注:

19.0 并行主端口 (PMP)

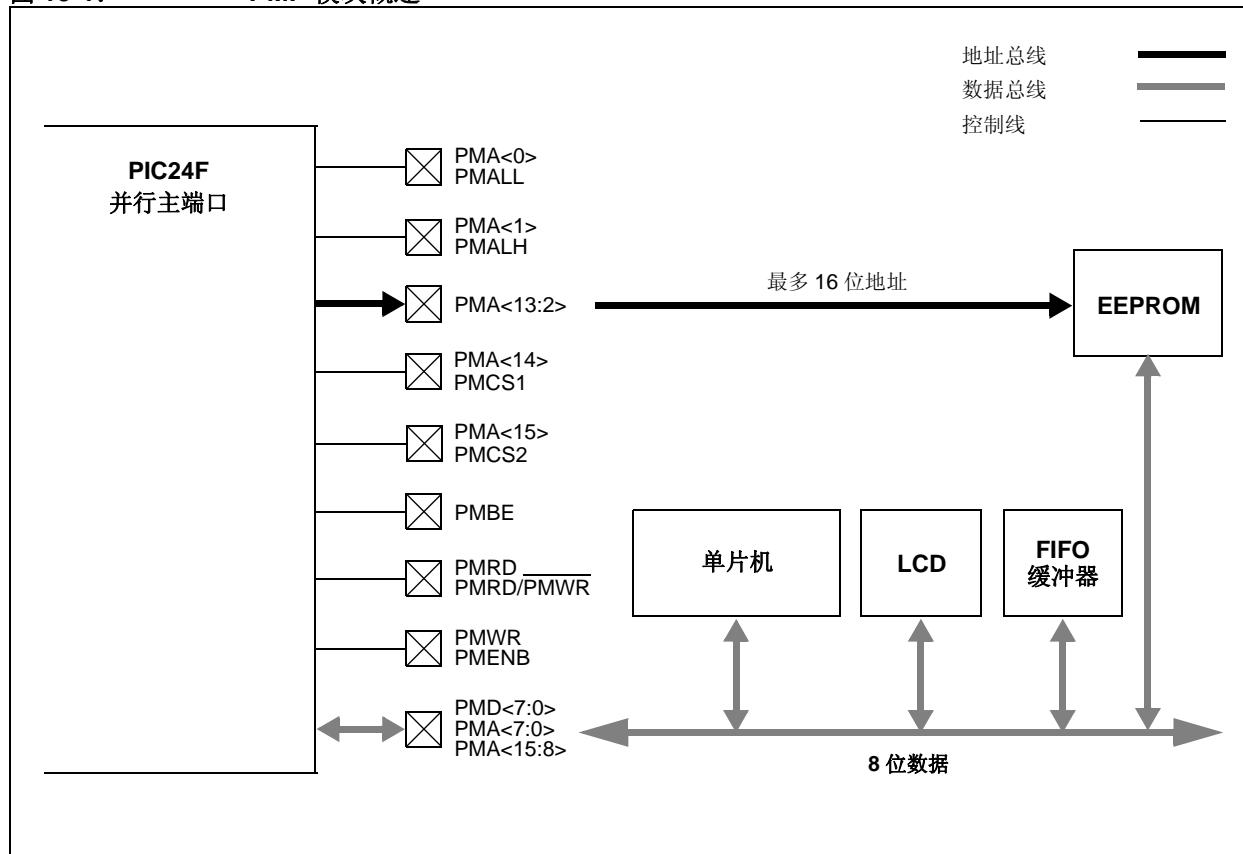
注: 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见《PIC24F 系列参考手册》的第 13 章“并行主端口 (PMP)”(DS39713A_CN)。

并行主端口 (Parallel Master Port, PMP) 模块是一个并行 8 位 I/O 模块, 专用于与例如通信外设、LCD、外部存储器件及单片机的较宽范围的并行器件进行通信。由于并行外设的接口变化很大, PMP 具有高度可配置性。

PMP 模块的主要特性包括:

- 最多 16 根可编程地址线
- 最多 2 根片选线
- 可编程选通选项:
 - 单独读和写选通, 或;
 - 带使能选通的读 / 写选通
- 地址自动递增 / 自动递减
- 可编程地址 / 数据复用
- 控制信号的可编程极性
- 支持传统并行从端口
- 增强型并行从动支持:
 - 地址支持
 - 4 字节深的自动递增缓冲器
- 可编程等待状态
- 可选择的输入电平

图 19-1: PMP 模块概述



PIC24FJ256GB110 系列

寄存器 19-1: PMCON: 并行端口控制寄存器

R/W-0	U-0	R/W-0	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0	R/W-0	R/W-0
PMSEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEN	PTWREN	PTRDEN
bit 15	bit 8						

R/W-0	R/W-0	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0 ⁽¹⁾	R/W-0	R/W-0	R/W-0
CSF1	CSF0	ALP	CS2P	CS1P	BEP	WRSP	RDSP
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **PMSEN:** 并行主端口使能位
1 = 使能 PMP
0 = 禁止 PMP, 不执行片外访问
- bit 14 未实现: 读为 0
- bit 13 **PSIDL:** 空闲模式停止位
1 = 当器件进入空闲模式时, 模块停止工作
0 = 在空闲模式下模块继续工作
- bit 12-11 **ADRMUX<1:0>:** 地址 / 数据复用选择位⁽¹⁾
11 = 保留
10 = 地址的所有 16 位在 PMD<7:0> 引脚上复用
01 = 地址的低 8 位在 PMD<7:0> 引脚上复用, 高 3 位在 PMA<10:8> 上复用
00 = 地址和数据出现在独立的引脚上
- bit 10 **PTBEN:** 字节使能端口使能位 (16 位主模式)
1 = 使能 PMBE 端口
0 = 禁止 PMBE 端口
- bit 9 **PTWREN:** 写使能选通端口使能位
1 = 使能 PMWR/PMENB 端口
0 = 禁止 PMWR/PMENB 端口
- bit 8 **PTRDEN:** 读 / 写选通端口使能位
1 = 使能 PMRD/PMWR 端口
0 = 禁止 PMRD/PMWR 端口
- bit 7-6 **CSF<1:0>:** 片选功能位
11 = 保留
10 = PMCS1 和 PMCS2 用作片选功能
01 = PMCS2 用作片选功能, PMCS1 用作地址的 bit 14
00 = PMCS1 和 PMCS2 分别用作地址的 bit 15 和 bit 14
- bit 5 **ALP:** 地址锁存器极性位⁽¹⁾
1 = 高电平有效 (PMALL 和 PMALH)
0 = 低电平有效 (PMALL 和 PMALH)
- bit 4 **CS2P:** 片选 2 极性位⁽¹⁾
1 = 高电平有效 (PMCS2/PMCS2)
0 = 低电平有效 (PMCS2/PMCS2)
- bit 3 **CS1P:** 片选 1 极性位⁽¹⁾
1 = 高电平有效 (PMCS1/PMCS1)
0 = 低电平有效 (PMCS1/PMCS1)

注 1: 这些位在相应引脚用作地址线时无效。

寄存器 19-1: PMCON: 并行端口控制寄存器 (续)

bit 2	BEP: 字节使能极性位 1 = 字节使能高电平有效 (PMBE) 0 = 字节使能低电平有效 (PMBE)
bit 1	WRSP: 写选通极性位 <u>对于从模式和主模式 2 (PMMODE<9:8> = 00、01 和 10):</u> 1 = 写选通高电平有效 (PMWR) 0 = 写选通低电平有效 (PMWR) <u>对于主模式 1 (PMMODE<9:8> = 11):</u> 1 = 使能选通高电平有效 (PMENB) 0 = 使能选通低电平有效 (PMENB)
bit 0	RDSP: 读选通极性位 <u>对于从模式和主模式 2 (PMMODE<9:8> = 00、01 和 10):</u> 1 = 读选通高电平有效 (PMRD) 0 = 读选通低电平有效 (PMRD) <u>对于主模式 1 (PMMODE<9:8> = 11):</u> 1 = 读 / 写选通高电平有效 (PMRD/PMWR) 0 = 读 / 写选通低电平有效 (PMRD/PMWR)

注 1: 这些位在相应引脚用作地址线时无效。

PIC24FJ256GB110 系列

寄存器 19-2: PMMODE: 并行端口模式寄存器

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAITB1 ⁽¹⁾	WAITB0 ⁽¹⁾	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1 ⁽¹⁾	WAITE0 ⁽¹⁾
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **BUSY:** 忙位 (仅限主模式)
1 = 端口处于忙状态 (当处理器停止工作时无用)
0 = 端口未处于忙状态
- bit 14-13 **IRQM<1:0>:** 中断请求模式位
11 = 当对读缓冲器 3 执行读操作或对写缓冲器 3 执行写操作时产生中断 (缓冲 PSP 模式)
或者当 PMA<1:0> = 11 (仅限可寻址 PSP 模式) 时的读 / 写操作时产生中断
10 = 不产生中断, 处理器停止工作
01 = 读 / 写周期末尾产生中断
00 = 不产生中断
- bit 12-11 **INCM<1:0>:** 递增模式位
11 = PSP 读和写缓冲器自动递增 (仅限传统 PSP 模式)
10 = 每个读 / 写周期 ADDR<10:0> 减 1
01 = 每个读 / 写周期 ADDR<10:0> 加 1
00 = 无地址递增或递减
- bit 10 **MODE16:** 8/16 位模式位
1 = 16 位模式: 数据寄存器为 16 位, 对数据寄存器执行读或写操作调用两次 8 位传输
0 = 8 位模式: 数据寄存器为 8 位, 对数据寄存器执行读或写操作调用一次 8 位传输
- bit 9-8 **MODE<1:0>:** 并行端口模式选择位
11 = 主模式 1 (PMCS1、PMRD/PMWR、PMENB、PMBE、PMA<x:0> 和 PMD<7:0>)
10 = 主模式 2 (PMCS1、PMRD、PMWR、PMBE、PMA<x:0> 和 PMD<7:0>)
01 = 增强型 PSP, 控制信号 (PMRD、PMWR、PMCS1、PMD<7:0> 和 PMA<1:0>)
00 = 传统并行从端口, 控制信号 (PMRD、PMWR、PMCS1 和 PMD<7:0>)
- bit 7-6 **WAITB<1:0>:** 数据建立到执行读 / 写的等待状态配置位⁽¹⁾
11 = 数据等待 4 个 TCY; 复用地址阶段等待 4 个 TCY
10 = 数据等待 3 个 TCY; 复用地址阶段等待 3 个 TCY
01 = 数据等待 2 个 TCY; 复用地址阶段等待 2 个 TCY
00 = 数据等待 1 个 TCY; 复用地址阶段等待 1 个 TCY
- bit 5-2 **WAITM<3:0>:** 读操作到字节使能选通的等待状态配置位
1111 = 等待额外 15 个 TCY
...
0001 = 等待额外 1 个 TCY
0000 = 无额外等待周期 (强制操作在 1 个 TCY 内执行) ⁽²⁾
- bit 1-0 **WAITE<1:0>:** 选通后数据保持等待状态配置位⁽¹⁾
11 = 等待 4 个 TCY
10 = 等待 3 个 TCY
01 = 等待 2 个 TCY
00 = 等待 1 个 TCY

注 1: 只要 WAITM<3:0> = 0000, WAITB 和 WAITE 位就被忽略。

2: 在连续的读和 / 或写操作之间需要一个周期的延时。

寄存器 19-3: PMADDR: 并行端口地址寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CS2	CS1			ADDR<13:8>			
bit 15						bit 8	

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **CS2:** 片选 2 位

1 = 片选 2 有效

0 = 片选 2 无效

bit 14 **CS1:** 片选 1 位

1 = 片选 1 有效

0 = 片选 1 无效

bit 13-0 **ADDR<13:0>:** 并行端口目标地址位

寄存器 19-4: PMAEN: 并行端口使能寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8
bit 15						bit 8	

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| PTEN7 | PTEN6 | PTEN5 | PTEN4 | PTEN3 | PTEN2 | PTEN1 | PTEN0 |
| bit 7 | | | | | | bit 0 | |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 **PTEN<15:14>:** PMCSx 选通使能位

1 = PMA15 和 PMA14 作为 PMA<15:14> 或 PMCS2 和 PMCS1

0 = PMA15 和 PMA14 作为端口 I/O

bit 13-2 **PTEN<13:2>:** PMP 地址端口使能位

1 = PMA<13:2> 作为 PMP 地址线

0 = PMA<13:2> 作为端口 I/O

bit 1-0 **PTEN<1:0>:** PMALH/PMALL 选通使能位

1 = PMA1 和 PMA0 作为 PMA<1:0> 或 PMALH 和 PMALL

0 = PMA1 和 PMA0 引脚作为端口 I/O

PIC24FJ256GB110 系列

寄存器 19-5: PMSTAT: 并行端口状态寄存器

R-0	R/W-0, HS	U-0	U-0	R-0	R-0	R-0	R-0
IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F
bit 15	bit 8						

R-1	R/W-0, HS	U-0	U-0	R-1	R-1	R-1	R-1
OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E
bit 7	bit 0						

图注: HS = 可由硬件置 1 位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **IBF:** 输入缓冲器满状态位
1 = 所有可写输入缓冲寄存器均已满
0 = 部分或所有可写输入缓冲寄存器为空
- bit 14 **IBOV:** 输入缓冲器溢出状态位
1 = 尝试对已满的输入字节寄存器进行写操作 (必须用软件清零)
0 = 未发生溢出
- bit 13-12 未实现: 读为 0
- bit 11-8 **IB3F:IB0F:** 输入缓冲器 x 状态满位
1 = 输入缓冲器包含尚未读取的数据 (读缓冲器将清零该位)
0 = 输入缓冲器不包含任何未读数据
- bit 7 **OBE:** 输出缓冲器空状态位
1 = 所有可读输出缓冲寄存器均为空
0 = 部分或所有可读输出缓冲寄存器已满
- bit 6 **OBUF:** 输出缓冲器下溢状态位
1 = 对空输出字节寄存器执行读操作 (必须用软件清零)
0 = 未发生下溢
- bit 5-4 未实现: 读为 0
- bit 3-0 **OB3E:OB0E:** 输出缓冲器 x 状态空位
1 = 输出缓冲器为空 (向缓冲器写数据将清零该位)
0 = 输出缓冲器包含尚未发送的数据

寄存器 19-6: PADCFG1: 焊垫配置控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-2 未实现: 读为 0

bit 1 **RTSECSEL:** RTCC 秒时钟输出选择位⁽¹⁾

1 = 选择 RTCC 引脚输出 RTCC 秒时钟

0 = 选择 RTCC 引脚输出 RTCC 闹钟脉冲

bit 0 **PMPTTL:** PMP 模块 TTL 输入缓冲器选择位

1 = PMP 模块输入 (PMDx 和 PMCS1) 使用 TTL 输入缓冲器

0 = PMP 模块输入使用施密特触发器输入缓冲器

注 1: 要使能实际 RTCC 输出, RTCOE (RCFGCAL<10>) 位必须也置 1。

PIC24FJ256GB110 系列

图 19-2：传统并行从端口示例

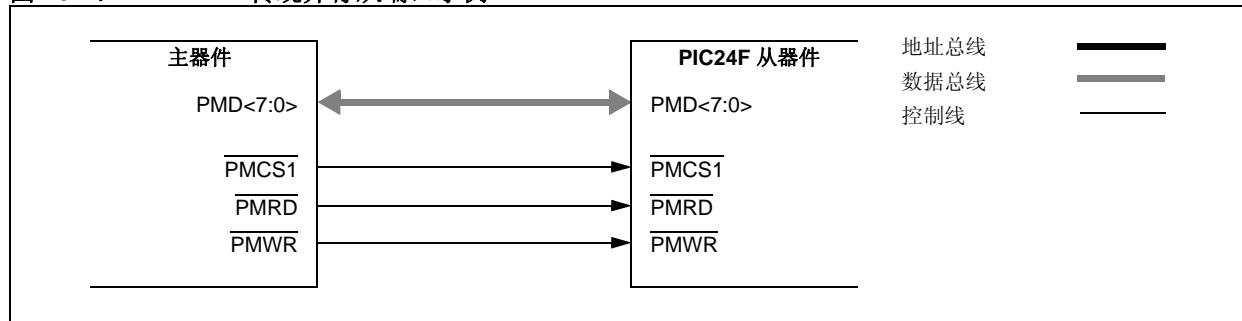


图 19-3：可寻址的并行从端口示例

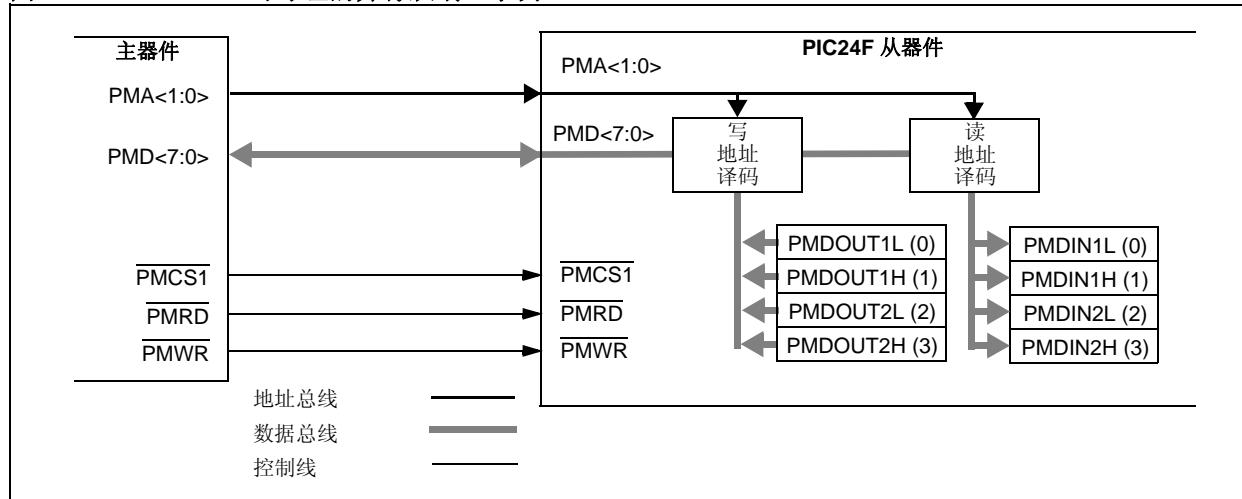


表 19-1：从模式地址解析

PMA<1:0>	输出寄存器（缓冲器）	输入寄存器（缓冲器）
00	PMDOUT1<7:0> (0)	PMDIN1<7:0> (0)
01	PMDOUT1<15:8> (1)	PMDIN1<15:8> (1)
10	PMDOUT2<7:0> (2)	PMDIN2<7:0> (2)
11	PMDOUT2<15:8> (3)	PMDIN2<15:8> (3)

图 19-4：主模式，未复用的寻址（独立读和写选通，两个片选）

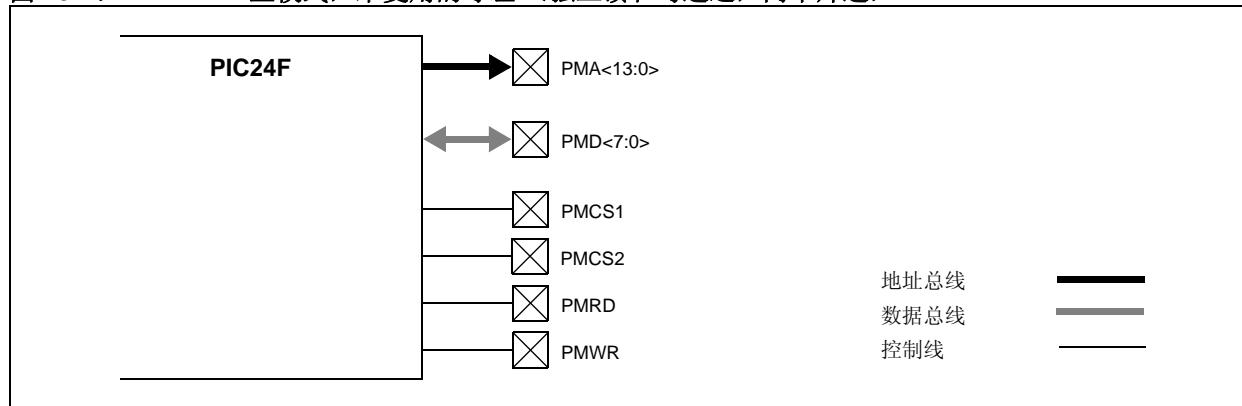


图 19-5: 主模式, 部分复用的寻址 (独立读和写选通, 两个片选)

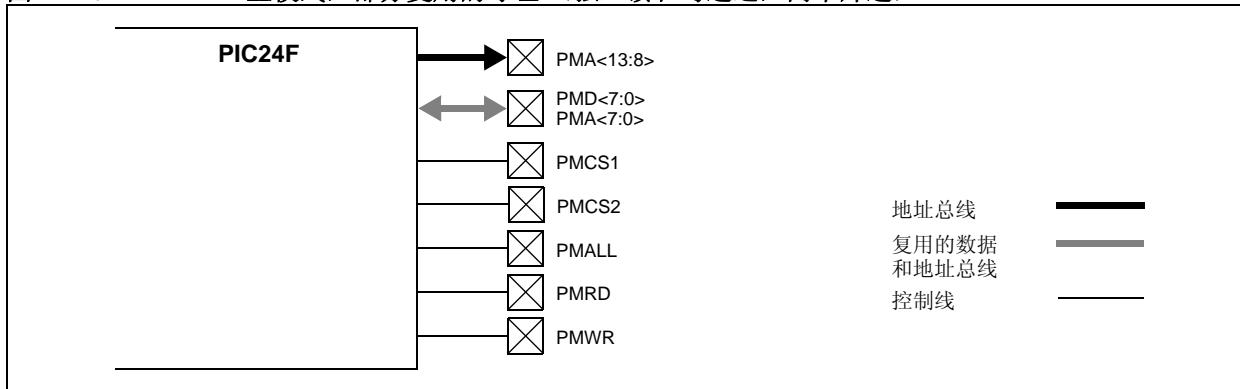


图 19-6: 主模式, 完全复用的寻址 (独立读和写选通, 两个片选)

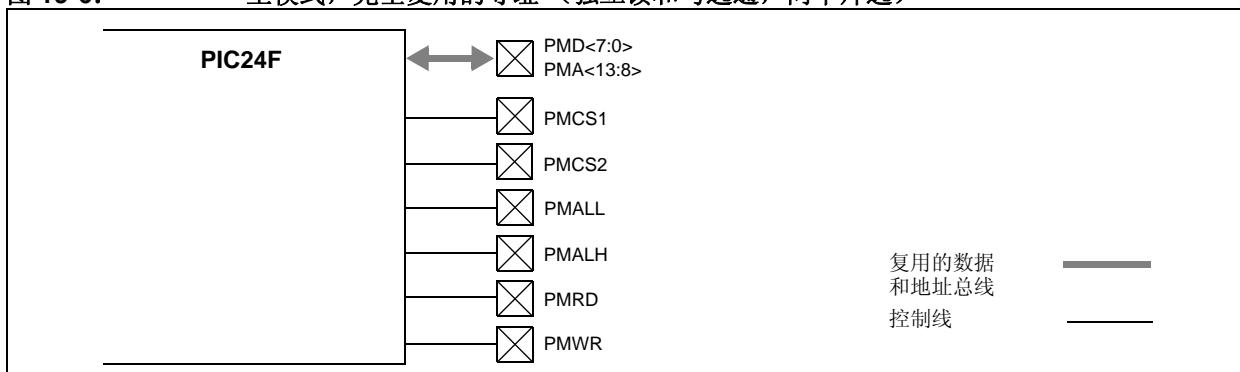


图 19-7: 复用寻址应用示例

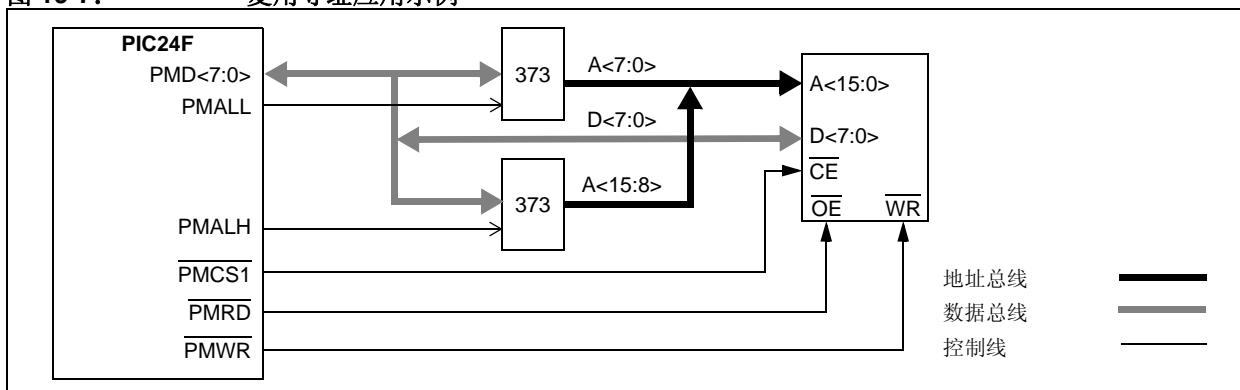
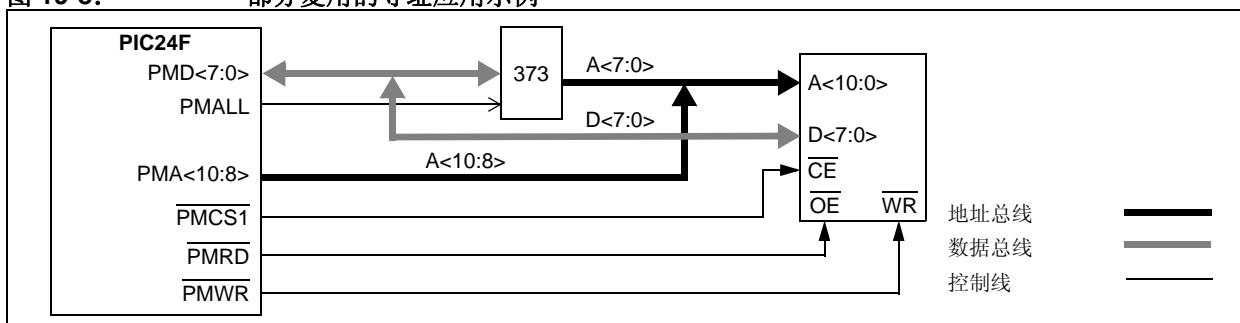


图 19-8: 部分复用的寻址应用示例



PIC24FJ256GB110 系列

图 19-9: 8 位复用地址和数据应用示例



图 19-10: 并行 EEPROM 示例 (最多 15 位地址, 8 位数据)

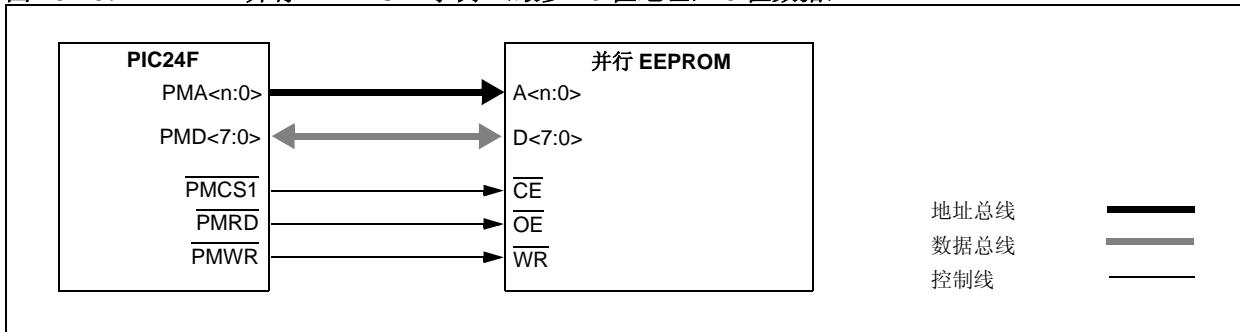


图 19-11: 并行 EEPROM 示例 (最多 15 位地址, 16 位数据)

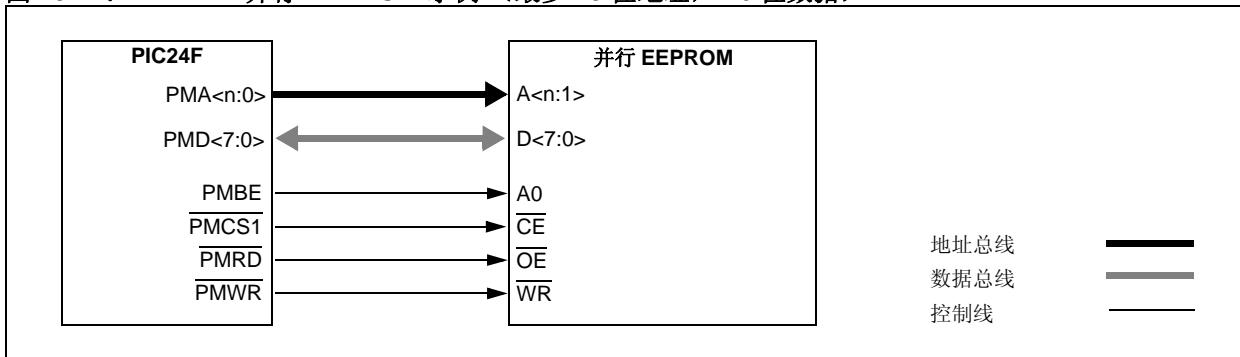
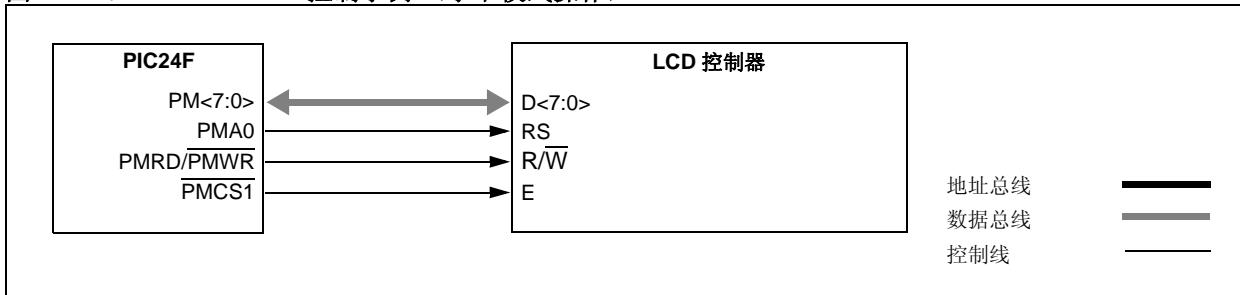


图 19-12: LCD 控制示例 (字节模式操作)



20.0 实时时钟和日历 (RTCC)

注：本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》的第 29 章“实时时钟和日历 (RTCC)”(DS39696A_CN)。

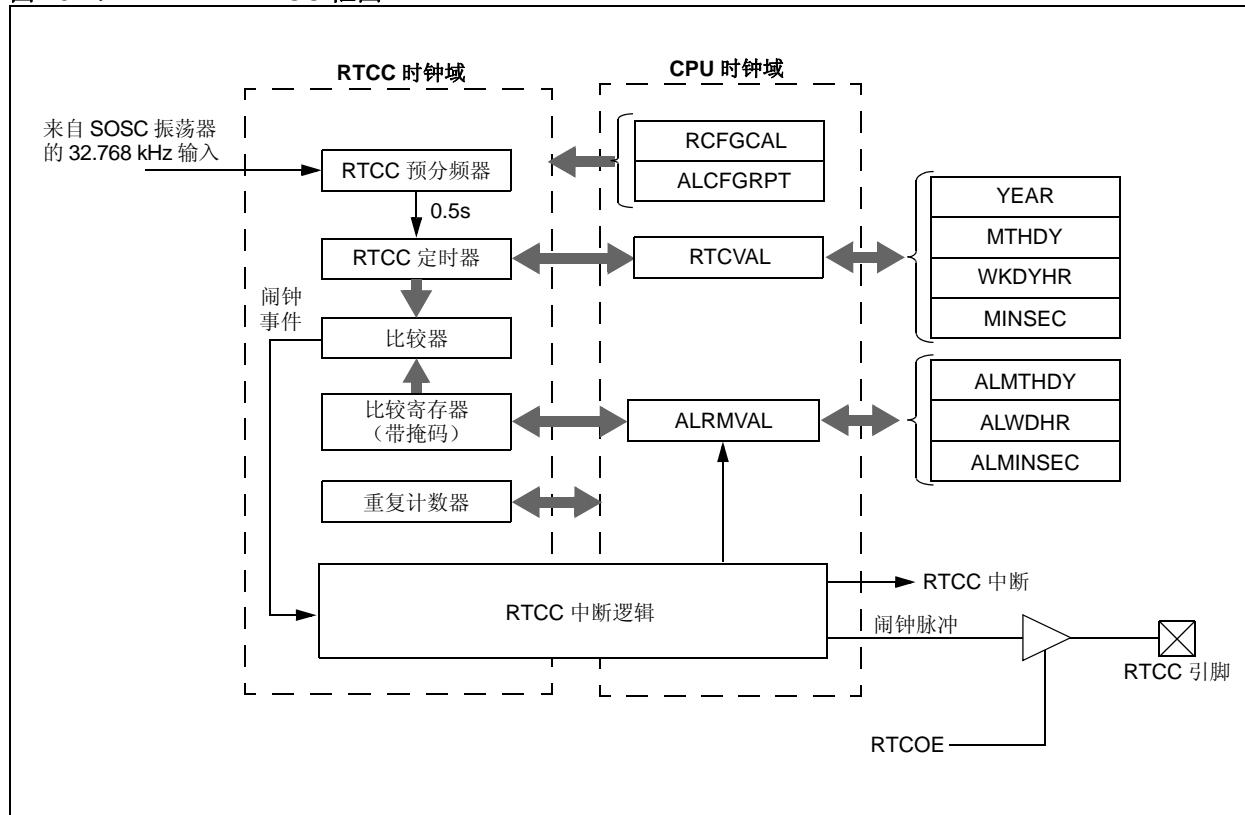
实时时钟和日历 (RTCC) 提供了片上硬件时钟和日历功能，CPU 开销极少或为零。该功能适用于必须长时间保持时间精准，但 CPU 活动极少且电源受限的应用（如电池供电应用）。

包括以下主要功能：

- 以时、分、秒表示的时间数据，精度为 1/2 秒
- 24 小时格式（军用时间）显示选项
- 以日、月、年表示的日历数据
- 由硬件自动计算星期和闰年（年限从 2000 到 2099）
- 压缩固件的时间和日历数据为 BCD 格式
- 高度可配置的闹钟功能
- 外部输出引脚，可选择输出闹钟信号或秒“滴答”信号
- 可自动调节的用户校准功能

图 20-1 给出了该模块的简单框图。当器件由于有效的 MCLR 信号处于复位状态时，SOSC 和 RTCC 均保持工作，并且当器件从 MCLR 复位恢复工作后，仍继续运行。

图 20-1： RTCC 框图



20.1 RTCC 模块寄存器

RTCC 模块寄存器可分为以下三类：

- RTCC 控制寄存器
- RTCC 值寄存器
- 闹钟值寄存器

20.1.1 寄存器映射

为限制寄存器接口，RTCC 定时器和闹钟时间寄存器通过相应的寄存器指针访问。RTCC 值寄存器窗口（RTCVALH 和 RTCVALL）使用 RTCPTR 位（RCFGCAL<9:8>）选择所需定时器寄存器对（见表 20-1）。

通过写 RTCVALH 字节，RTCC 指针值 RTCPTR<1:0> 位减 1，直到达到 00。一旦达到 00，分钟和秒值可通过 RTCVALH 和 RTCVALL 访问，直到手动更改指针值。

表 20-1： RTCVAL 寄存器映射

RTCPTR <1:0>	RTCC 值寄存器窗口	
	RTCVAL<15:8>	RTCVAL<7:0>
00	分钟	秒
01	星期	小时
10	月	日
11	—	年

闹钟值寄存器窗口（ALRMVALH 和 ALRMMVAL）使用 ALRMPTR 位（ALCFGPR<9:8>）选择所需闹钟寄存器对（见表 20-2）。

通过写 ALRMVALH 字节，闹钟指针值 ALRMPTR<1:0> 位减 1，直到达到 00。一旦达到 00，ALRMMIN 和 ALRMSEC 值可通过 ALRMVALH 和 ALRMMVAL 访问，直到手动更改指针值。

例 20-1： 将 RTCWREN 位置 1

```
_builtin_write_RTCWEN(); //set the RTCWREN bit
```

表 20-2： ALRMVAL 寄存器映射

ALRMPTR <1:0>	闹钟值寄存器窗口	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

考虑到 16 位内核并不区分 8 位和 16 位读操作，用户要注意，读 ALRMVALH 或 ALRMMVAL 字节时都会使 ALRMPTR<1:0> 值减 1。同样的规律也适用于 RTCVALH 或 RTCVALL 字节，读取它们时会使 RTCPTR<1:0> 减 1。

注： 这只适用于读操作，不适用于写操作。

20.1.2 写锁定

要对任何 RTCC 定时器寄存器执行写操作，必须先将 RTCWREN 位（RCFGCAL<13>）置 1（见例 20-1）。

注： 为避免意外写入定时器，建议其他任何时候 RTCWREN 位（RCFGCAL<13>）都保持清零。要将 RTCWREN 置 1，在解锁序列和 RTCWREN 置 1 之间只允许 1 个指令周期的时间段；因此，建议遵循例 20-1 中的代码示例。对于用 C 语言编写的应用程序，解锁序列应使用行内汇编实现。

20.1.3 RTCC 控制寄存器

寄存器 20-1: RCFGCAL: RTCC 校准和配置寄存器⁽¹⁾

R/W-0	U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0
RTCEN ⁽²⁾	—	RTCWREN	RTCSYNC	HALFSEC ⁽³⁾	RTCOE	RTC PTR1	RTC PTR0
bit 15	bit 8						

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CAL7 | CAL6 | CAL5 | CAL4 | CAL3 | CAL2 | CAL1 | CAL0 |
| bit 7 | bit 0 | | | | | | |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15	RTCEN: RTCC 使能位 ⁽²⁾
	1 = 使能 RTCC 模块
	0 = 禁止 RTCC 模块
bit 14	未实现: 读为 0
bit 13	RTCWREN: RTCC 值寄存器写使能位
	1 = RTCVALH 和 RTCVALL 寄存器可由用户写入
	0 = RTCVALH 和 RTCVALL 寄存器已锁定, 不可由用户写入
bit 12	RTCSYNC: RTCC 值寄存器读同步位
	1 = 由于计满返回, RTCVALH、RTCVALL 和 ALCFGRPT 寄存器在读操作过程中可能改变, 从而导致读取的数据无效。如果两次读取寄存器得到的数据相同, 可认为数据是有效的。
	0 = RTCVALH、RTCVALL 或 ALCFGRPT 寄存器在读取时无需考虑计满返回
bit 11	HALFSEC: 半秒状态位 ⁽³⁾
	1 = 一秒的后一半
	0 = 一秒的前一半
bit 10	RTCOE: RTCC 输出使能位
	1 = 使能 RTCC 输出
	0 = 禁止 RTCC 输出
bit 9-8	RTC PTR<1:0>: RTCC 值寄存器窗口指针位
	读取 RTCVALH 和 RTCVALL 寄存器时, 指向相应的 RTCC 值寄存器; 每当读或写 RTCVALH 时 RTC PTR<1:0> 的值就减 1, 直到达到 00。
	RTCVAL<15:8>:
	00 = 分钟数
	01 = 星期
	10 = 月
	11 = 保留
	RTCVAL<7:0>:
	00 = 秒数
	01 = 小时数
	10 = 日
	11 = 年

注 1: RCFGCAL 寄存器只受 POR 的影响。

2: 仅当 RTCWREN = 1 时允许写入 RTCEN 位。

3: 该位是只读的。写入 MINSEC 寄存器的低半部分时, 它被清零。

PIC24FJ256GB110 系列

寄存器 20-1: RCFGCAL: RTCC 校准和配置寄存器⁽¹⁾ (续)

bit 7-0	CAL<7:0> : RTC 漂移校准位
	01111111 = 最大正向调整; 每分钟增加 508 个 RTC 时钟脉冲
	... 00000001 = 最小正向调整; 每分钟增加 4 个 RTC 时钟脉冲
	00000000 = 无调整
	11111111 = 最小负向调整; 每分钟减少 4 个 RTC 时钟脉冲
	... 10000000 = 最大负向调整; 每分钟减少 512 个 RTC 时钟脉冲

- 注 1: RCFGCAL 寄存器只受 POR 的影响。
2: 仅当 RTCWREN = 1 时允许写入 RTCEN 位。
3: 该位是只读的。写入 MINSEC 寄存器的低半部分时, 它被清零。

寄存器 20-2: PADCFG1: 焊垫配置控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							

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

图注:

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

bit 15-2	未实现: 读为 0
bit 1	RTSECSEL : RTCC 秒时钟输出选择位 ⁽¹⁾ 1 = 选择 RTCC 引脚输出 RTCC 秒时钟 0 = 选择 RTCC 引脚输出 RTCC 频率脉冲
bit 0	PMPTTL : PMP 模块 TTL 输入缓冲器选择位 1 = PMP 模块输入 (PMDx 和 PMCS1) 使用 TTL 输入缓冲器 0 = PMP 模块输入使用施密特触发器输入缓冲器

- 注 1: 要使能实际 RTCC 输出, RTCOE (RCFGCAL<10>) 位也必须置 1。

寄存器 20-3: ALCFGRPT: 闹钟配置寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| ARPT7 | ARPT6 | ARPT5 | ARPT4 | ARPT3 | ARPT2 | ARPT1 | ARPT0 |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **ALRMEN:** 闹钟使能位

1 = 闹钟已使能 (每当 ARPT<7:0> = 00h 且 CHIME = 0 时, 发生闹钟事件后都自动清零)
0 = 闹钟已禁止

bit 14 **CHIME:** 响铃 (Chime) 使能位

1 = 响铃已使能; ARPT<7:0> 位允许从 00h 进位到 FFh
0 = 响铃已禁止; ARPT<7:0> 到达 00h 就停止

bit 13-10 **AMASK<3:0>:** 闹钟掩码配置位

0000 = 每半秒
0001 = 每秒
0010 = 每 10 秒
0011 = 每分钟
0100 = 每 10 分钟
0101 = 每小时
0110 = 一天一次
0111 = 一周一次
1000 = 一月一次
1001 = 一年一次 (除非配置在 2 月 29 日, 亦即每 4 年一次的情况外)
101x = 保留——不要使用
11xx = 保留——不要使用

bit 9-8 **ALRMPTR<1:0>:** 闹钟值寄存器窗口指针位

读取 ALRMVALH 和 ALRMVALL 寄存器时, 指向相应的闹钟值寄存器; 每当读或写 ALRMVALH 时 ALRMPTR<1:0> 的值就减 1, 直到达到 00。

ALRMVAL<15:8>:

00 = ALRMMIN
01 = ALRMWD
10 = ALRMMNTH
11 = 未实现

ALRMVAL<7:0>:

00 = ALRMSEC
01 = ALRMHR
10 = ALRMDAY
11 = 未实现

bit 7-0 **ARPT<7:0>:** 闹钟重复计数器值位

11111111 = 闹钟将再重复 255 次

...

00000000 = 闹钟将不再重复

每当发生闹钟事件时计数器就减 1。除非 CHIME = 1, 否则计数器不能从 00h 进位到 FFh。

PIC24FJ256GB110 系列

20.1.4 RTCVAL 寄存器映射

寄存器 20-4: YEAR: 年值寄存器⁽¹⁾

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

| R/W-x |
|--------|--------|--------|--------|--------|--------|--------|--------|
| YRTEN3 | YRTEN2 | YRTEN1 | YRTEN0 | YRONE3 | YRONE2 | YRONE1 | YRONE0 |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7-4 YRTEN<3:0>: 年份的十位数的二 - 十进制码 (Binary Coded Decimal, BCD) 值位
包含的值为 0 到 9。

bit 3-0 YRONE<3:0>: 年份的个位数的 BCD 值位
包含的值为 0 到 9。

注 1: 仅当 RTCWREN = 1 时允许写入 YEAR 寄存器。

寄存器 20-5: MTHDY: 月和日值寄存器⁽¹⁾

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

bit 15-13 未实现: 读为 0

bit 12 MTHTEN0: 月份的十位数的 BCD 值位
包含的值为 0 或 1。

bit 11-8 MTHONE<3:0>: 月份的个位数的 BCD 值位
包含的值为 0 到 9。

bit 7-6 未实现: 读为 0

bit 5-4 DAYTEN<1:0>: 日的十位数的 BCD 值位
包含的值为 0 到 3。

bit 3-0 DAYONE<3:0>: 日的个位数的 BCD 值位
包含的值为 0 到 9。

注 1: 仅当 RTCWREN = 1 时允许写入该寄存器。

寄存器 20-6: WKDYHR: 星期和小时值寄存器⁽¹⁾

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15						bit 8	

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10-8 **WDAY<2:0>**: 星期的 BCD 值位

包含值为 0 到 6。

bit 7-6 未实现: 读为 0

bit 5-4 **HRTEN<1:0>**: 小时的十位数的 BCD 值位

包含的值为 0 到 2。

bit 3-0 **HRONE<3:0>**: 小时的个位数的 BCD 值位

包含的值为 0 到 9。

注 1: 仅当 RTCWREN = 1 时允许写入该寄存器。

寄存器 20-7: MINSEC: 分钟和秒值寄存器

U-0	R/W-x						
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15						bit 8	

U-0	R/W-x						
—	SECTEN2	SECTEN1	SECTEN0	SECOME3	SECOME2	SECOME1	SECOME0
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **MINTEN<2:0>**: 分钟的十位数的 BCD 值位

包含的值为 0 到 5。

bit 11-8 **MINONE<3:0>**: 分钟的个位数的 BCD 值位

包含的值为 0 到 9。

bit 7 未实现: 读为 0

bit 6-4 **SECTEN<2:0>**: 秒的十位数的 BCD 值位

包含的值为 0 到 5。

bit 3-0 **SECOME<3:0>**: 秒的个位数的 BCD 值位。

包含的值为 0 到 9。

PIC24FJ256GB110 系列

20.1.5 ALRMVAL 寄存器映射

寄存器 20-8: ALMTHDY: 阵钟月和日值寄存器⁽¹⁾

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-13 未实现: 读为 0

bit 12 MTHTEN0: 月份的十位数的 BCD 值位
包含的值为 0 或 1。

bit 11-8 MTHONE<3:0>: 月份的个位数的 BCD 值位
包含的值为 0 到 9。

bit 7-6 未实现: 读为 0

bit 5-4 DAYTEN<1:0>: 日的十位数的 BCD 值位
包含的值为 0 到 3。

bit 3-0 DAYONE<3:0>: 日的个位数的 BCD 值位
包含的值为 0 到 9。

注 1: 仅当 RTCWREN = 1 时允许写入该寄存器。

寄存器 20-9: ALWDHR: 闹钟星期和小时值寄存器⁽¹⁾

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10-8 **WDAY<2:0>**: 星期的 BCD 值位
包含的值为 0 到 6。

bit 7-6 未实现: 读为 0

bit 5-4 **HRTEN<1:0>**: 小时的十位数的 BCD 值位
包含的值为 0 到 2。

bit 3-0 **HRONE<3:0>**: 小时的个位数的 BCD 值位
包含的值为 0 到 9。

注 1: 仅当 RTCWREN = 1 时允许写入该寄存器。

寄存器 20-10: ALMINSEC: 闹钟分钟和秒值寄存器

U-0	R/W-x						
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0	R/W-x						
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **MINTEN<2:0>**: 分钟的十位数的 BCD 值位
包含的值为 0 到 5。

bit 11-8 **MINONE<3:0>**: 分钟的个位数的 BCD 值位
包含的值为 0 到 9。

bit 7 未实现: 读为 0

bit 6-4 **SECTEN<2:0>**: 秒的十位数的 BCD 值位
包含的值为 0 到 5。

bit 3-0 **SECONE<3:0>**: 秒的个位数的 BCD 值位
包含的值为 0 到 9。

20.2 校准

实时晶振输入可用周期性自动调节功能校准。正确校准后，RTCC 可提供小于每月 3 秒的误差。这是通过找到一分钟内的误差时钟脉冲数，将该值存储到 RCFGCAL 寄存器的低位部分实现的。装入 RCFGCAL 低位部分的 8 位有符号值乘以 4，每分钟一次从 RTCC 定时器中加上或减去。关于 RTCC 校准，请参见以下步骤：

1. 用器件上的其他定时器资源，用户必须找出 32.768 kHz 晶振的误差。
2. 知道误差后，必须将它转换为每分钟误差时钟脉冲数，并将其装载到 RCFGCAL 寄存器。

公式 20-1： RTCC 校准

$$\text{误差 (每分钟时钟数)} = (\text{理想频率} - \text{测得频率}) * 60$$

† 理想频率 = 32,768 Hz

3. a) 如果振荡器快于理想频率（从步骤 2 得出的负的结果），RCFGCAL 寄存器值必须为负。这会导致每分钟从定时器计数器上减去指定的时钟脉冲数。
b) 如果振荡器慢于理想频率（从步骤 2 得出的正的结果），RCFGCAL 寄存器值必须为正。这会导致每分钟在定时器计数器上加上指定的时钟脉冲数。
4. 将每分钟误差时钟数除以 4 得到正确的 CAL 值，并将正确的值装入 RCFGCAL 寄存器。

（CAL 中每 1 位递增会加上或减去 4 个脉冲）。

只有当定时器关闭或紧接秒脉冲的上升沿时，才会发生对 RCFGCAL 寄存器低位部分的写入。

注：是否在误差值中包含晶振初始误差、温度造成的漂移和晶振老化造成的漂移，由用户自行决定。

20.3 闹钟

- 可在半秒到一年的范围内配置
- 使用 ALRMEN 位（寄存器 20-3 的 ALCFGRPT<15>）使能
- 有一次性闹钟和重复闹钟选项可用

20.3.1 配置闹钟

闹钟功能用 ALRMEN 位使能。发出闹钟后该位清零。当 ALRMEN = 0 时，只应发生对 ALRMVAL 的写操作。闹钟的间隔选择通过 AMASK 位（ALCFGRPT<13:10>）配置，如图 20-2 所示。这些位决定了要触发闹钟，闹钟的哪些位、多少位必须和时钟值匹配。

也可以配置闹钟使之根据预先配置的间隔重复。闹钟使能后发生的总次数存储在 ARPT 位 ARPT<7:0>（ALCFGRPT<7:0>）中。当 ARPT 位的值等于 00h 且 CHIME 位（ALCFGRPT<14>）清零时，重复功能被禁止，只发生单次闹钟。通过将 FFh 装入 ARPT<7:0>，闹钟可重复最多 255 次。

每个闹钟发出后，ARPT 位的值都减 1。值达到 00h 后，将最后一次发出闹钟，此后 ALRMEN 位将自动清零，闹钟将关闭。

如果 CHIME 位 = 1，闹钟可能不断重复。当 CHIME 置 1 时，ARPT 位的值达到 00h 时不会禁止闹钟，而是进位到 FFh，继续无限计数。

20.3.2 闹钟中断

每个闹钟事件发生时，都会产生中断。此外会提供闹钟脉冲输出，其频率是闹钟频率的一半。该输出完全和 RTCC 时钟同步，可用作其他外设的触发时钟。

注：闹钟使能时（ALRMEN = 1），更改除 RCFGCAL 和 ALCFGRPT 寄存器以外的任何寄存器以及 CHIME 位，都会导致误闹钟事件，进而导致错误的闹钟中断。为避免误闹钟事件，只应在闹钟禁止时（ALRMEN = 0）更改定时器和闹钟值。建议在 RTCSYNC = 0 时更改 ALCFGRPT 寄存器和 CHIME 位。

图 20-2: 闹钟掩码设置

闹钟掩码设置 (AMASK<3:0>)	星期	月	日	小时	分钟	秒
0000 ——每半秒	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/>
0001 ——每秒	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/>
0010 ——每 10 秒	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/> s
0011 ——每分钟	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> s <input type="checkbox"/> s
0100 ——每 10 分钟	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> m <input type="checkbox"/>	: <input type="checkbox"/> s <input type="checkbox"/> s
0101 ——每小时	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> m <input type="checkbox"/> m <input type="checkbox"/>	: <input type="checkbox"/> s <input type="checkbox"/> s
0110 ——每天	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> h <input type="checkbox"/> h <input type="checkbox"/>	: <input type="checkbox"/> m <input type="checkbox"/> m <input type="checkbox"/>	: <input type="checkbox"/> s <input type="checkbox"/> s
0111 ——每周	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> h <input type="checkbox"/> h <input type="checkbox"/>	: <input type="checkbox"/> m <input type="checkbox"/> m <input type="checkbox"/>	: <input type="checkbox"/> s <input type="checkbox"/> s
1000 ——每月	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> d <input type="checkbox"/> d <input type="checkbox"/>	<input type="checkbox"/> h <input type="checkbox"/> h <input type="checkbox"/>	: <input type="checkbox"/> m <input type="checkbox"/> m <input type="checkbox"/>	: <input type="checkbox"/> s <input type="checkbox"/> s
1001 ——每年 ⁽¹⁾	<input type="checkbox"/>	<input type="checkbox"/> m <input type="checkbox"/> m <input type="checkbox"/>	/ <input type="checkbox"/> d <input type="checkbox"/> d <input type="checkbox"/>	<input type="checkbox"/> h <input type="checkbox"/> h <input type="checkbox"/>	: <input type="checkbox"/> m <input type="checkbox"/> m <input type="checkbox"/>	: <input type="checkbox"/> s <input type="checkbox"/> s

注 1: 每年, 除非配置为 2 月 29 日。

PIC24FJ256GB110 系列

注:

21.0 可编程循环冗余校验（CRC）发生器

注： 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》的第 30 章“可编程循环冗余校验（CRC）”（DS39714A_CN）。

可编程 CRC 发生器提供以下特性：

- 用户可编程多项式 CRC 方程
- 中断输出
- 数据 FIFO

该模块实现了可用软件配置的 CRC 发生器。多项式的项及其长度可分别用 X<15:1> 位（CRCXOR<15:1>）和 PLEN<3:0> 位（CRCCON<3:0>）设定。

考虑以下 CRC 方程：

$$x^{16} + x^{12} + x^5 + 1$$

要将该多项式编程到 CRC 发生器中，CRC 寄存器位应置 1，如表 21-1 所示。

表 21-1：CRC 设置示例

位名称	位值
PLEN<3:0>	1111
X<15:1>	00010000010000

请注意，对于 X<15:1> 的值，第 12 位和第 5 位按照方程要求置为 1。方程要求第 0 位总是经过逻辑异或运算的。对于 16 次多项式，第 16 位也总是经过逻辑异或运算的；因此，X<15:1> 位没有第 0 位，也没有第 16 位。

图 21-1 给出了该模块的简单框图。图 21-2 给出了移位引擎的一般拓扑。

图 21-1：CRC 框图

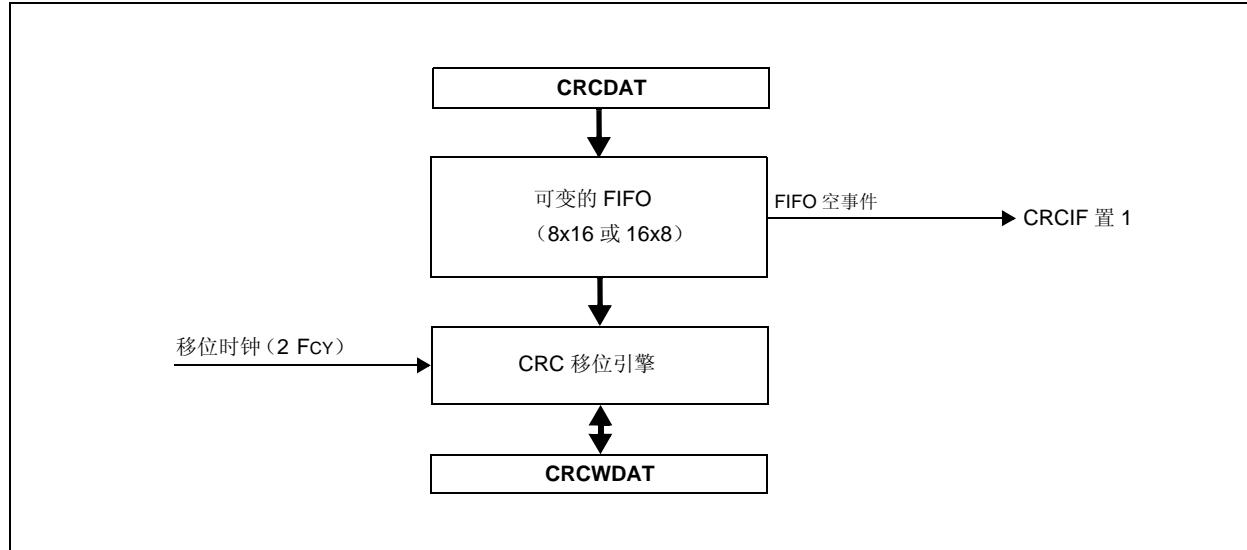
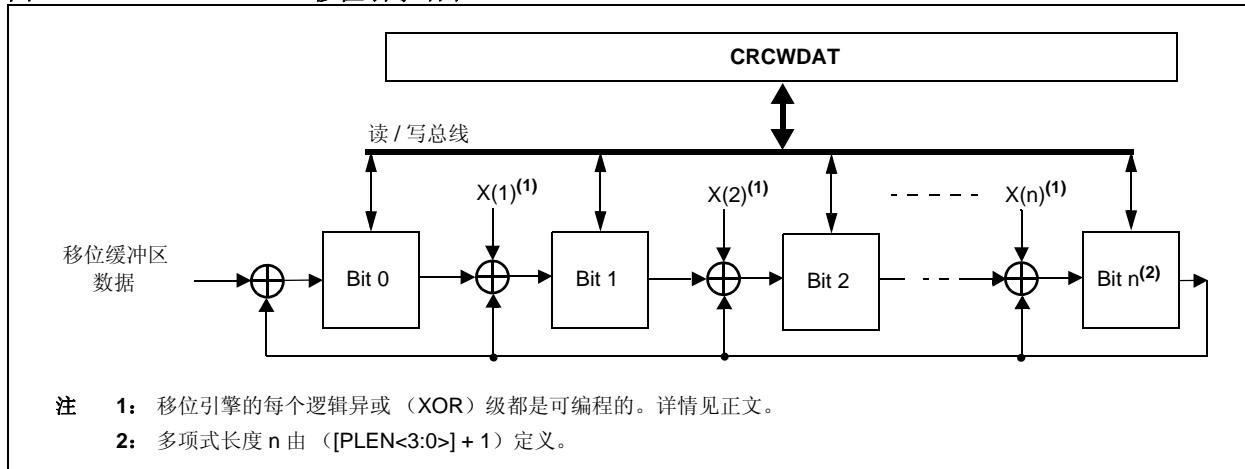


图 21-2: CRC 移位引擎细节



21.1 用户接口

21.1.1 数据接口

要开始串行移位，必须将 1 写入 CRCGO 位。

该模块包含的 FIFO 在 PLEN 位 (CRCCON<3:0>) > 7 时为 8 级深，其他情况下为 16 级深。CRC 计算所需数据必须首先写入 FIFO。可写入 FIFO 的最小数据元素是 1 个字节。例如，如果 $PLEN = 5$ ，则数据大小是 $PLEN + 1 = 6$ 。装载数据时，数据字节的两个最高位 (MSb) 被忽略。

数据一旦被写入 CRCWDAT 的最高位 (由 PLEN 定义)，VWORD (CRCCON<12:8>) 的值将递增一。当 $CRCGO = 1$ 且 $VWORD > 0$ 时，要移位的数据字就会从 FIFO 传递到移位引擎。数据字一旦从 FIFO 传递到移位引擎，VWORD 位就会递减 1。串行移位器会持续从 FIFO 接收数据进行移位操作，直到 VWORD 位为 0。VWORD 达到 0 后，经过 $(PLEN + 1)/2$ 个时钟周期后，最后一个数据位将经由 CRC 模块移位。此时，模块完成 CRC 计算。

因此，对给定的 PLEN 值，需要 $(PLEN + 1)/2 * VWORD$ 个时钟周期才能完成 CRC 计算。

VWORD 达到 8 (或 16) 时，CRCFUL 位将置 1。VWORD 达到 0 时，CRCMPT 位将置 1。

要连续将数据输送到 CRC 引擎中，建议的工作模式是预先为 FIFO 准备好足够的字数，这样下个字可写入之前不会产生中断。完成后，通过将 CRCGO 位置为 1 来启动 CRC。从那点开始将查询 VWORD 位。如果它们读到的数小于 8 或 16，将把另一个字写入 FIFO。

要清空已写入 FIFO 的字，CRCGO 位必须置为 1 并允许 CRC 移位器运行，直到 CRCMPT 位置 1。

同时，为了正确地读 CRC，必须等待 CRCMPT 位转为高电平才能读 CRCWDAT 寄存器。

如果 CRCFUL 位置 1 时写入一个字，VWORD 指针将计满返回到 0。硬件的行为就像 FIFO 为空时一样。但是，产生中断的条件并不满足；因此，不会产生中断（见第 21.1.2 节“中断操作”）。

写入 CRCWDAT 后必须经过至少一个指令周期才能读 VWORD 位。

21.1.2 中断操作

当 VWORD<4:0> 位的值从 1 变为 0 时，产生中断。注意，此时 CRC 计算并未完成，在读取输出之前，还需要 $(PLEN + 1)/2$ 个时钟周期。

21.2 省电模式下的操作

21.2.1 休眠模式

如果器件在模块工作时进入休眠模式，模块将暂停于当前状态，直到时钟恢复执行。

21.2.2 空闲模式

要在空闲模式下继续完整的模块工作，CSIDL 位必须在进入该模式前清零。

如果 CSDL = 1，该模块的行为与在休眠模式下相同；暂停的中断事件将传递下去，即使模块时钟不可用。

21.3 寄存器

有 4 个寄存器用于控制可编程 CRC 工作：

- CRCCON
- CRCXOR
- CRCDAT
- CRCWDAT

寄存器 21-1：CRCCON：CRC 控制寄存器

U-0	U-0	R/W-0	R-0	R-0	R-0	R-0	R-0
—	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0
bit 15							

R-0	R-1	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CRCFUL	CRCMPT	—	CRCGO	PLEN3	PLEN2	PLEN1	PLEN0
bit 7							

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现：读为 0

bit 13 **CSIDL：CRC 空闲模式停止位**

1 = 当器件进入空闲模式时，模块停止工作
0 = 在空闲模式下模块继续工作

bit 12-8 **VWORD<4:0>：指针值位**

表示 FIFO 中有效字的个数。PLEN<3:0> > 7 时最大值是 8，或者 PLEN<3:0> ≤ 7 时最大值是 16。

bit 7 **CRCFUL：FIFO 满位**

1 = FIFO 已满
0 = FIFO 未满

bit 6 **CRCMPT：FIFO 空位**

1 = FIFO 为空
0 = FIFO 非空

bit 5 未实现：读为 0

bit 4 **CRCGO：启动 CRC 位**

1 = 启动 CRC 串行移位器
0 = CRC 串行移位器关闭

bit 3-0 **PLEN<3:0>：多项式长度位**

表示生成多项式的长度减去 1。

PIC24FJ256GB110 系列

寄存器 21-2: CRCXOR: CRC 异或多项式寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X15	X14	X13	X12	X11	X10	X9	X8
bit 15							bit 8

R/W-0	U-0						
X7	X6	X5	X4	X3	X2	X1	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-1 **X<15:1>**: 多项式的项 X^n 的异或使能位

bit 0 未实现: 读为 0

22.0 10 位高速 A/D 转换器

注: 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见《PIC24F 系列参考手册》的第 17 章“10 位 A/D 转换器”(DS39705A_CN)。

10 位 A/D 转换器具有以下主要特性:

- 逐次逼近 (Successive Approximation, SAR) 转换
- 转换速度最高可达 500 ksps
- 16 个模拟输入引脚
- 外部参考电压输入引脚
- 内部带隙参考输入
- 自动通道扫描模式
- 可选择的转换触发源
- 16 字的转换结果缓冲器
- 可选择的缓冲器填充模式
- 4 个结果对齐选项
- 可在 CPU 休眠和空闲模式下工作

在所有 PIC24FJ256GB110 系列器件上, 10 位 A/D 转换器有 16 个模拟输入引脚, 指定为 AN0 到 AN15。此外, 还有两个模拟输入引脚用于连接外部参考电压 (V_{REF+} 和 V_{REF-})。这两个参考电压输入可以与其他模拟输入引脚共用。

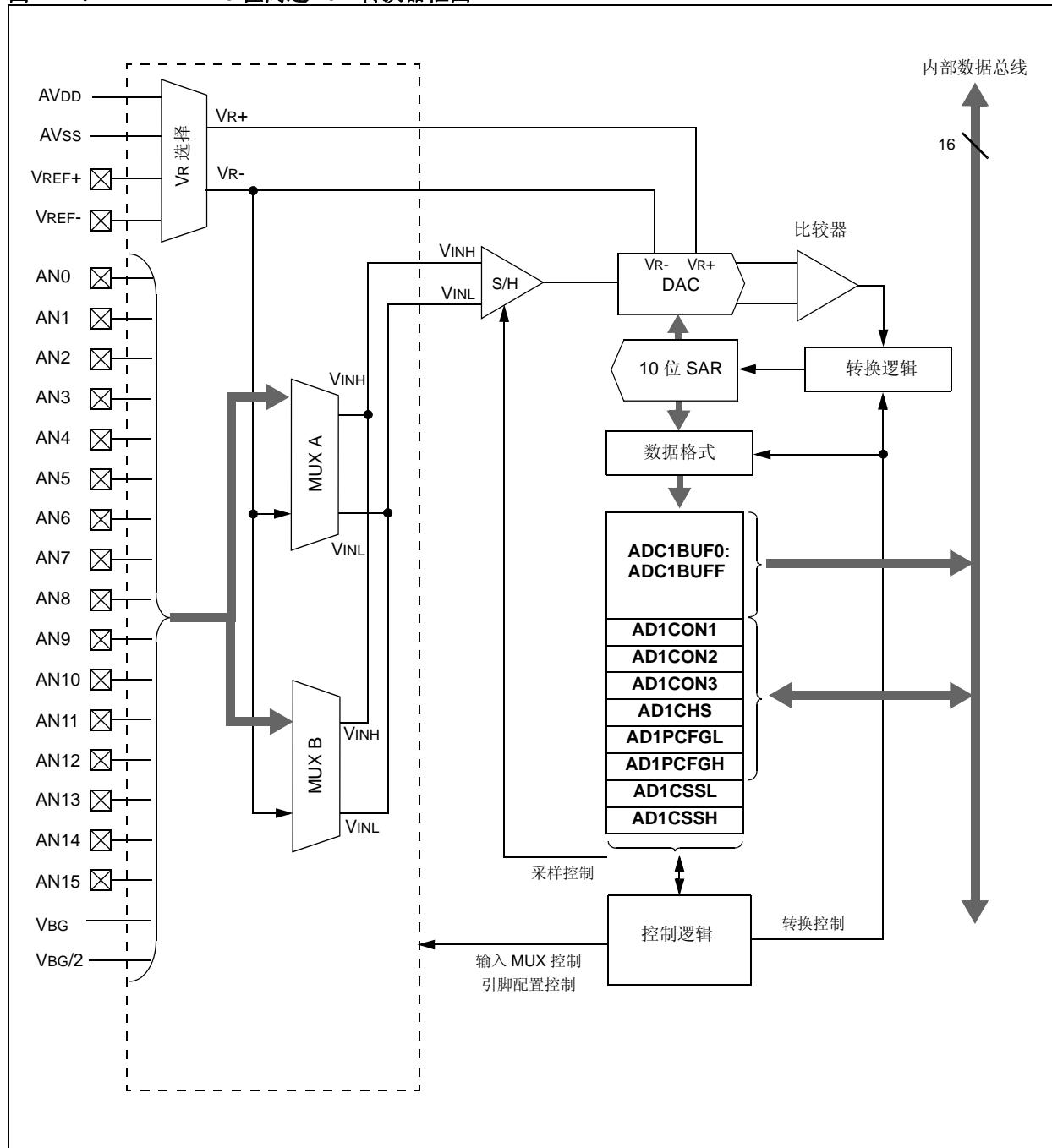
图 22-1 给出了 A/D 转换器的框图。

要执行 A/D 转换:

1. 配置 A/D 模块:
 - a) 配置端口引脚作为模拟输入, 还可选择带隙参考输入 (AD1PCFG<15:0> 和 AD1PCFGH<1:0>)。
 - b) 选择参考电压源以匹配模拟输入的预期范围 (AD1CON2<15:13>)。
 - c) 选择模拟转换时钟以便使期望的数据速率与处理器时钟匹配 (AD1CON3<7:0>)。
 - d) 选择适当的采样 / 转换序列 (AD1CON1<7:5> 和 AD1CON3<12:8>)。
 - e) 选择转换结果在缓冲器中的放置形式 (AD1CON1<9:8>)。
 - f) 选择中断速率 (AD1CON2<5:2>)。
 - g) 开启 A/D 模块 (AD1CON1<15>)。
2. 配置 A/D 中断 (如需要):
 - a) 清零 AD1IF 位。
 - b) 选择 A/D 中断优先级。

PIC24FJ256GB110 系列

图 22-1：10 位高速 A/D 转换器框图



寄存器 22-1: AD1CON1: A/D 控制寄存器 1

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ADON ⁽¹⁾	—	ADSLD	—	—	—	FORM1	FORM0
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0, HCS	R/W-0, HCS
SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE
bit 7	bit 0						

图注: HCS = 硬件清零 / 置 1 位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- | | |
|-----------|---------------------------------------|
| bit 15 | ADON: A/D 工作模式位 ⁽¹⁾ |
| | 1 = A/D 转换器模块正在工作 |
| | 0 = A/D 转换器关闭 |
| bit 14 | 未实现: 读为 0 |
| bit 13 | ADSLD: 空闲模式停止位 |
| | 1 = 当器件进入空闲模式时, 模块停止工作 |
| | 0 = 在空闲模式下模块继续工作 |
| bit 12-10 | 未实现: 读为 0 |
| bit 9-8 | FORM<1:0>: 数据输出格式位 |
| | 11 = 有符号小数 (sddd dddd dd00 0000) |
| | 10 = 小数 (ddd dddd dd00 0000) |
| | 01 = 有符号整数 (ssss sssd dddd dddd) |
| | 00 = 整数 (0000 00dd dddd dddd) |
| bit 7-5 | SSRC<2:0>: 转换触发源选择位 |
| | 111 = 由内部计数器结束采样并启动转换 (自动转换) |
| | 110 = 由 CTMU 事件结束采样并启动转换 |
| | 101 = 保留 |
| | 100 = 由 Timer5 比较结束采样并启动转换 |
| | 011 = 保留 |
| | 010 = 由 Timer3 比较结束采样并启动转换 |
| | 001 = 由 INT0 引脚的有效跳变结束采样并启动转换 |
| | 000 = 由清零 SAMP 位结束采样并启动转换 |
| bit 4-3 | 未实现: 读为 0 |
| bit 2 | ASAM: A/D 采样自动启动位 |
| | 1 = 最后一次转换结束后立即开始采样。 SAMP 位自动置 1。 |
| | 0 = SAMP 位置 1 时开始采样 |
| bit 1 | SAMP: A/D 采样使能位 |
| | 1 = A/D 采样 / 保持放大器正在采样输入 |
| | 0 = A/D 采样 / 保持放大器保持采样结果 |
| bit 0 | DONE: A/D 转换状态位 |
| | 1 = A/D 转换已完成 |
| | 0 = A/D 转换未完成 |

注 1: 一旦 ADON 位清零, ADC1BUFx 寄存器将不再保留其值。应在禁止模块前从该缓冲器中读出转换结果。

PIC24FJ256GB110 系列

寄存器 22-2: AD1CON2: A/D 控制寄存器 2

R/W-0	R/W-0	R/W-0	r-0	U-0	R/W-0	U-0	U-0
VCFG2	VCFG1	VCFG0	r	—	CSCNA	—	—
bit 15							bit 8

R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS	—	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS
bit 7							bit 0

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

bit 15-13 **VCFG<2:0>**: 参考电压配置位

VCFG<2:0>	VR+	VR-
000	AVDD	AVSS
001	外部 VREF+ 引脚	AVSS
010	AVDD	外部 VREF- 引脚
011	外部 VREF+ 引脚	外部 VREF- 引脚
1xx	AVDD	AVSS

bit 12 **保留**: 保持为 0

bit 11 **未实现**: 读为 0

bit 10 **CSCNA**: 选择是否在使用采样多路开关 A 时扫描 CH0+ S/H 输入的位

1 = 扫描输入

0 = 不扫描输入

bit 9-8 **未实现**: 读为 0

bit 7 **BUFS**: 缓冲器填充状态位 (仅当 BUFM = 1 时有效)

1 = A/D 当前正在填充缓冲器 08-0F, 用户应访问 00-07 中的数据

0 = A/D 当前正在填充缓冲器 00-07, 用户应访问 08-0F 中的数据

bit 6 **未实现**: 读为 0

bit 5-2 **SMPI<3:0>**: 每个中断对应采样 / 转换序列数量选择位

1111 = 每完成 16 个采样 / 转换序列时产生中断

1110 = 每完成 15 个采样 / 转换序列时产生中断

.....

0001 = 每完成 2 个采样 / 转换序列时产生中断

0000 = 每完成 1 个采样 / 转换序列时产生中断

bit 1 **BUFM**: 缓冲器模式选择位

1 = 缓冲器配置为两个 8 字缓冲器 (ADC1BUFn<15:8> 和 ADC1BUFn<7:0>)

0 = 缓冲器配置为一个 16 字缓冲器 (ADC1BUFn<15:0>)

bit 0 **ALTS**: 交替输入采样模式选择位

1 = 对于第一次采样, 使用 MUX A 输入多路开关设置, 然后对于所有后续采样, 在 MUX B 和 MUX A 输入多路开关设置之间交替

0 = 始终使用 MUX A 输入多路开关设置

寄存器 22-3: AD1CON3: A/D 控制寄存器 3

R/W-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	r	r	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
bit 15							
							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| ADCS7 | ADCS6 | ADCS5 | ADCS4 | ADCS3 | ADCS2 | ADCS1 | ADCS0 |
| bit 7 | | | | | | | |
| | | | | | | | bit 0 |

图注:

r = 保留位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **ADRC:** A/D 转换时钟源位
 1 = A/D 内部 RC 时钟
 0 = 时钟由系统时钟产生

bit 14-13 保留: 保持为 0

bit 12-8 **SAMC<4:0>:** 自动采样时间位
 11111 = 31 TAD

 00001 = 1 TAD
 00000 = 0 TAD (不推荐)

bit 7-0 **ADCS<7:0>:** A/D 转换时钟选择位
 11111111
 = 保留, 不要使用
 01000000
 00111111 = 64 TCY
 00111110 = 63 TCY

 00000001 = 2*TCY
 00000000 = TCY

PIC24FJ256GB110 系列

寄存器 22-4: AD1CHS: A/D 输入选择寄存器

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB	—	—	CH0SB4 ⁽¹⁾	CH0SB3 ⁽¹⁾	CH0SB2 ⁽¹⁾	CH0SB1 ⁽¹⁾	CH0SB0 ⁽¹⁾
bit 15	bit 8						

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA	—	—	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **CH0NB:** MUX B 多路开关设置的通道 0 反相输入选择位

1 = 通道 0 的反相输入为 AN1

0 = 通道 0 的反相输入为 VR-

bit 14-13 未实现: 读为 0

bit 12-8 **CH0SB<4:0>:** MUX B 多路开关设置的通道 0 同相输入选择位⁽¹⁾

10001 = 通道 0 的同相输入为 内部带隙参考 (V_{BG})⁽²⁾

10000 = 通道 0 的同相输入为 V_{BG}/2⁽²⁾

01111 = 通道 0 的同相输入为 AN15

01110 = 通道 0 的同相输入为 AN14

01101 = 通道 0 的同相输入为 AN13

01100 = 通道 0 的同相输入为 AN12

01011 = 通道 0 的同相输入为 AN11

01010 = 通道 0 的同相输入为 AN10

01001 = 通道 0 的同相输入为 AN9

01000 = 通道 0 的同相输入为 AN8

00111 = 通道 0 的同相输入为 AN7

00110 = 通道 0 的同相输入为 AN6

00101 = 通道 0 的同相输入为 AN5

00100 = 通道 0 的同相输入为 AN4

00011 = 通道 0 的同相输入为 AN3

00010 = 通道 0 的同相输入为 AN2

00001 = 通道 0 的同相输入为 AN1

00000 = 通道 0 的同相输入为 AN0

bit 7 **CH0NA:** MUX A 多路开关设置的通道 0 反相输入选择位

1 = 通道 0 的反相输入为 AN1

0 = 通道 0 的反相输入为 VR-

bit 6-5 未实现: 读为 0

bit 4-0 **CH0SA<4:0>:** MUX A 多路开关设置的通道 0 同相输入选择位

已实现的组合与 CH0SB<4:0> (上面) 的相同。

注 1: 10010 至 11111 的组合未实现, 不要使用。

2: 在对这些通道进行转换之前, 带隙参考源必须稳定 (稳定时间以参数 TBG 表示)。更多信息, 请参见第 29.1 节 “直流动特性”。

寄存器 22-5: AD1PCFGL: A/D 端口配置寄存器 (低字节)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| PCFG7 | PCFG6 | PCFG5 | PCFG4 | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-0 **PCFG<15:0>**: 模拟输入引脚配置控制位

1 = 对应模拟通道的引脚配置为数字模式; 使能 I/O 端口读操作

0 = 引脚配置为模拟模式; 禁止 I/O 端口读操作, A/D 采样引脚电压

寄存器 22-6: AD1PCFGH: A/D 端口配置寄存器 (高字节)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	PCFG17	PCFG16
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-2 未实现: 读为 0

bit 1 **PCFG17**: A/D 输入配置控制位

1 = 禁止扫描模拟通道的输入

0 = 使能内部带隙 (VBG) 通道, 用于输入扫描

bit 0 **PCFG16**: A/D 输入半带隙扫描使能位

1 = 禁止扫描模拟通道的输入

0 = 使能内部 VBG/2 通道, 用于输入扫描

PIC24FJ256GB110 系列

寄存器 22-7: AD1CSSL: A/D 输入扫描选择寄存器 (低字节)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CSSL7 | CSSL6 | CSSL5 | CSSL4 | CSSL3 | CSSL2 | CSSL1 | CSSL0 |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-0 **CSSL<15:0>**: A/D 输入引脚扫描选择位

1 = 对选定的模拟通道进行输入扫描

0 = 输入扫描时忽略模拟通道

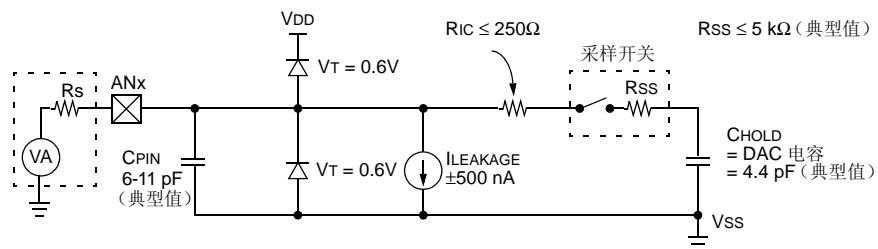
公式 22-1: A/D 转换时钟周期⁽¹⁾

$$\text{ADCS} = \frac{\text{TAD}}{\text{TCY}} - 1$$

$$\text{TAD} = \text{TCY} \cdot (\text{ADCS} + 1)$$

注 1: 基于 $\text{TCY} = 2 * \text{Tosc}$; 打盹模式和 PLL 被禁止。

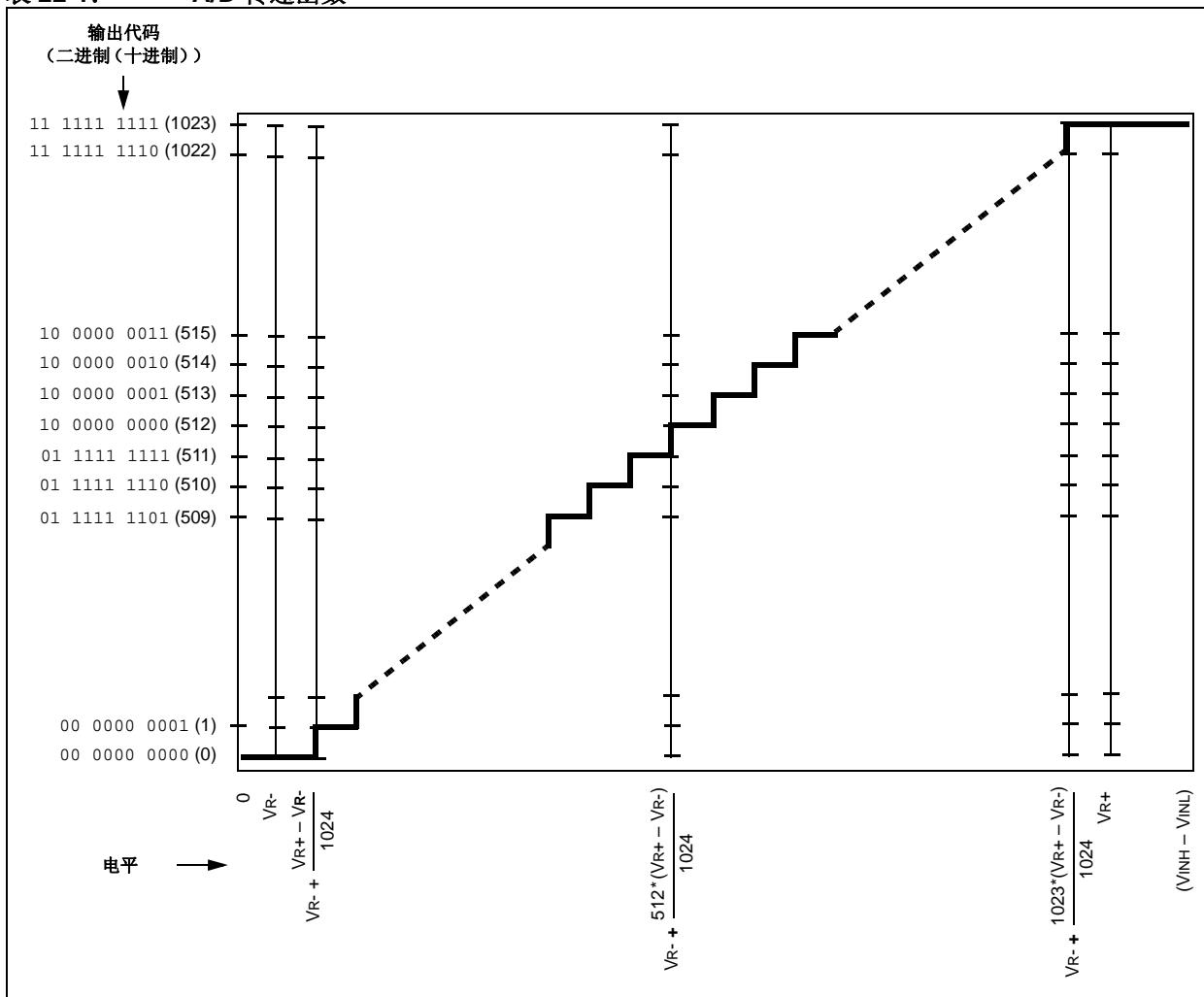
图 22-2: 10 位 A/D 转换器模拟输入模型



图注	C_{PIN} = 输入电容 V_T = 门限电压 $I_{LEAKAGE}$ = 由各连接点在引脚产生的泄漏电流 R_{IC} = 片内走线等效电阻 R_{SS} = 采样开关电阻 C_{HOLD} = 采样 / 保持电容 (来自 DAC)
----	--

注: C_{PIN} 值取决于器件封装, 未经测试。如果 $R_s \leq 5\text{ k}\Omega$, 可忽略 C_{PIN} 的影响。

表 22-1: A/D 传递函数



PIC24FJ256GB110 系列

注:

23.0 三比较器模块

注: 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见相关的《PIC24F 系列参考手册》章节。

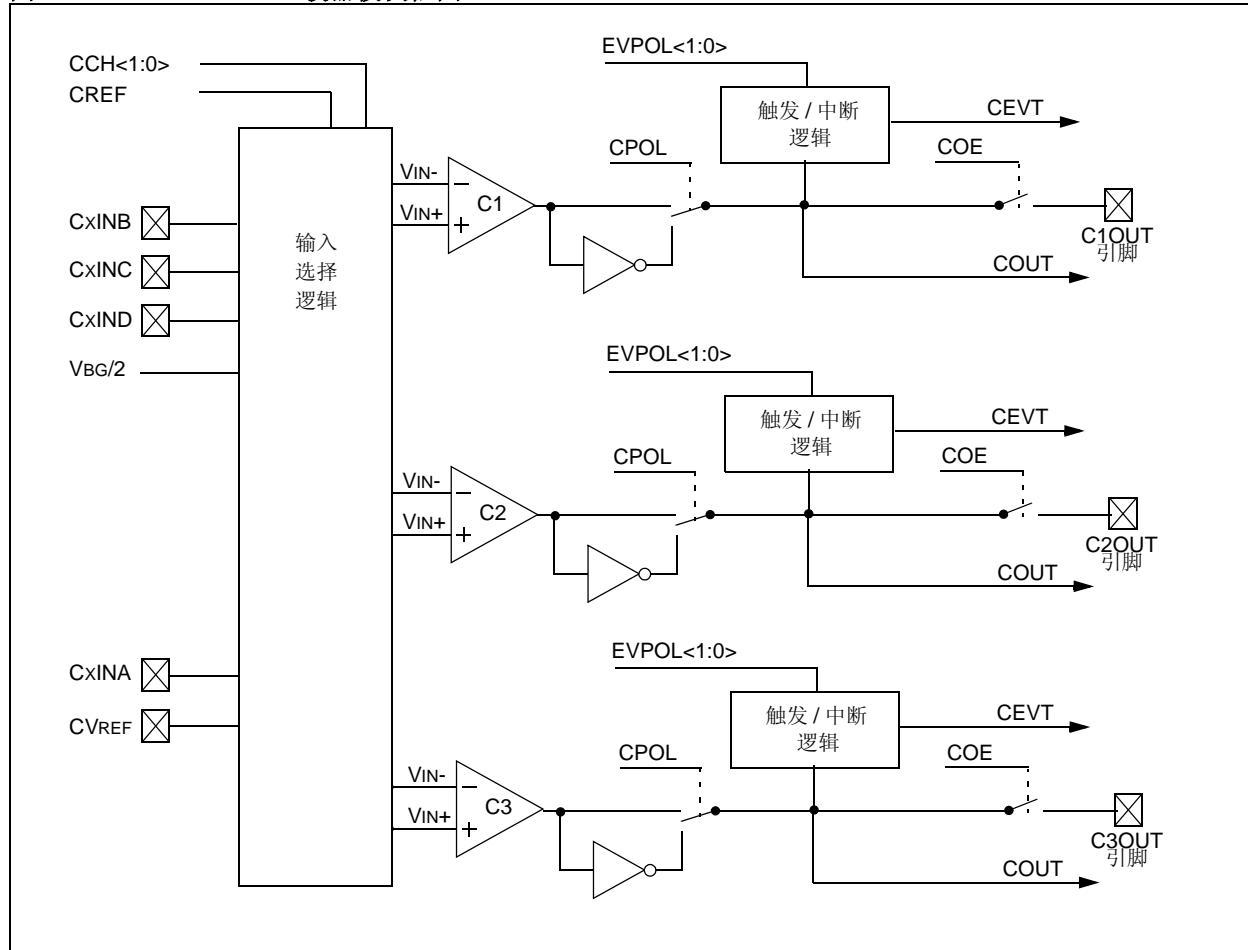
三比较器模块提供三个双输入比较器。至比较器的输入可以配置为使用 4 个外部模拟输入之一, 而且参考电压输入可以来自内部带隙参考的 2 分频 ($V_{BG}/2$) 信号或比较器参考电压发生器。

比较器输出可以直接连接到 $CxOUT$ 引脚。当相应的 COE 等于 1 时, I/O 引脚逻辑使比较器的未同步输出在引脚上可用。

图 23-1 给出了此模块的简化框图。图 23-2 给出了各种可能的比较器配置的框图。

每个比较器都有自己的控制寄存器 CM x CON (寄存器 23-1), 用于使能和配置其操作。所有 3 个比较器的输出和事件状态在 CMSTAT 寄存器 (寄存器 23-2) 中提供。

图 23-1: 三比较器模块框图



PIC24FJ256GB110 系列

图 23-2：各种比较器配置

比较器关闭 CEN = 0, CREF = x, CCH<1:0> = xx	
比较器 CxINB > CxINA 比较 CEN = 1, CREF = 0, CCH<1:0> = 00	比较器 CxINC > CxINA 比较 CEN = 1, CREF = 0, CCH<1:0> = 01
比较器 CxIND > CxINA 比较 CEN = 1, CREF = 0, CCH<1:0> = 10	比较器 VBG > CxINA 比较 CEN = 1, CREF = 0, CCH<1:0> = 11
比较器 CxINB > CVREF 比较 CEN = 1, CREF = 1, CCH<1:0> = 00	比较器 CxINC > CVREF 比较 CEN = 1, CREF = 1, CCH<1:0> = 01
比较器 CxIND > CVREF 比较 CEN = 1, CREF = 1, CCH<1:0> = 10	比较器 VBG > CVREF 比较 CEN = 1, CREF = 1, CCH<1:0> = 11

寄存器 23-1: CMxCON: 比较器 x 控制寄存器 (比较器 1 至 3)

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R-0
CEN	COE	CPOL	—	—	—	CEVT	COUT
bit 15	bit 8						

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **CEN:** 比较器使能位
 1 = 使能比较器
 0 = 禁止比较器
- bit 14 **COE:** 比较器输出使能位
 1 = 比较器输出出现在 CxOUT 引脚上
 0 = 比较器输出仅在内部出现
- bit 13 **CPOL:** 比较器输出极性选择位
 1 = 比较器输出反相
 0 = 比较器输出不反相
- bit 12-10 未实现: 读为 0
- bit 9 **CEVT:** 比较器事件位
 1 = 已发生由 EVPOL<1:0> 定义的比较器事件; 后续的触发和中断被禁止, 直到该位被清零
 0 = 未发生比较器事件
- bit 8 **COUT:** 比较器输出位
当 CPOL = 0 时:
 1 = VIN+ > VIN-
 0 = VIN+ < VIN-
当 CPOL = 1 时:
 1 = VIN+ < VIN-
 0 = VIN+ > VIN-
- bit 7-6 **EVPOL<1:0>:** 触发 / 事件 / 中断极性选择位
 11 = 在比较器输出发生任何变化时产生触发 / 事件 / 中断 (当 CEVT = 0 时)
 10 = 在比较器输出发生跳变时产生触发 / 事件 / 中断:
 如果 CPOL = 0 (同相极性):
 仅限从高至低跳变。
 如果 CPOL = 1 (反相极性):
 仅限从低至高跳变。
 01 = 在比较器输出发生跳变时产生触发 / 事件 / 中断:
 如果 CPOL = 0 (同相极性):
 仅限从低至高跳变。
 如果 CPOL = 1 (反相极性):
 仅限从高至低跳变。
 00 = 禁止触发 / 事件 / 中断发生
- bit 5 未实现: 读为 0

PIC24FJ256GB110 系列

寄存器 23-1: CMxCON: 比较器 x 控制寄存器 (比较器 1 至 3) (续)

bit 4	CREF: 比较器参考电压选择位 (同相输入) 1 = 同相输入连接到内部 CVREF 电压 0 = 同相输入连接到 CxINA 引脚
bit 3-2	未实现: 读为 0
bit 1-0	CCH<1:0>: 比较器通道选择位 11 = 比较器的反相输入连接到 VBG/2 10 = 比较器的反相输入连接到 CxIND 引脚 01 = 比较器的反相输入连接到 CxINC 引脚 00 = 比较器的反相输入连接到 CxINB 引脚

寄存器 23-2: CMSTAT: 比较器模块状态寄存器

R/W-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
CMIDL	—	—	—	—	C3EVT	C2EVT	C1EVT
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	—	—	—	C3OUT	C2OUT	C1OUT
bit 7							bit 0

图注:

R = 可读位

-n = POR 时的值

W = 可写位

1 = 置 1

U = 未实现位, 读为 0

0 = 清零

x = 未知

bit 15	CMIDL: 比较器空闲模式停止位 1 = 模块在空闲模式下不会产生中断, 而是继续工作 0 = 模块在空闲模式下继续正常工作
bit 14-11	未实现: 读为 0
bit 10	C3EVT: 比较器 3 事件状态位 (只读) 显示比较器 3 当前的事件状态 (CM3CON<9>)。
bit 9	C2EVT: 比较器 2 事件状态位 (只读) 显示比较器 2 当前的事件状态 (CM2CON<9>)。
bit 8	C1EVT: 比较器 1 事件状态位 (只读) 显示比较器 1 当前的事件状态 (CM1CON<9>)。
bit 7-3	未实现: 读为 0
bit 2	C3OUT: 比较器 3 输出状态位 (只读) 显示比较器 3 当前的输出状态 (CM3CON<8>)。
bit 1	C2OUT: 比较器 2 输出状态位 (只读) 显示比较器 2 当前的输出状态 (CM2CON<8>)。
bit 0	C1OUT: 比较器 1 输出状态位 (只读) 显示比较器 1 当前的输出状态 (CM1CON<8>)。

24.0 比较器参考电压

注：本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息，请参见《PIC24F 系列参考手册》的“第 20 章 比较器参考电压模块”(DS39709A_CN)。

24.1 配置比较器参考电压

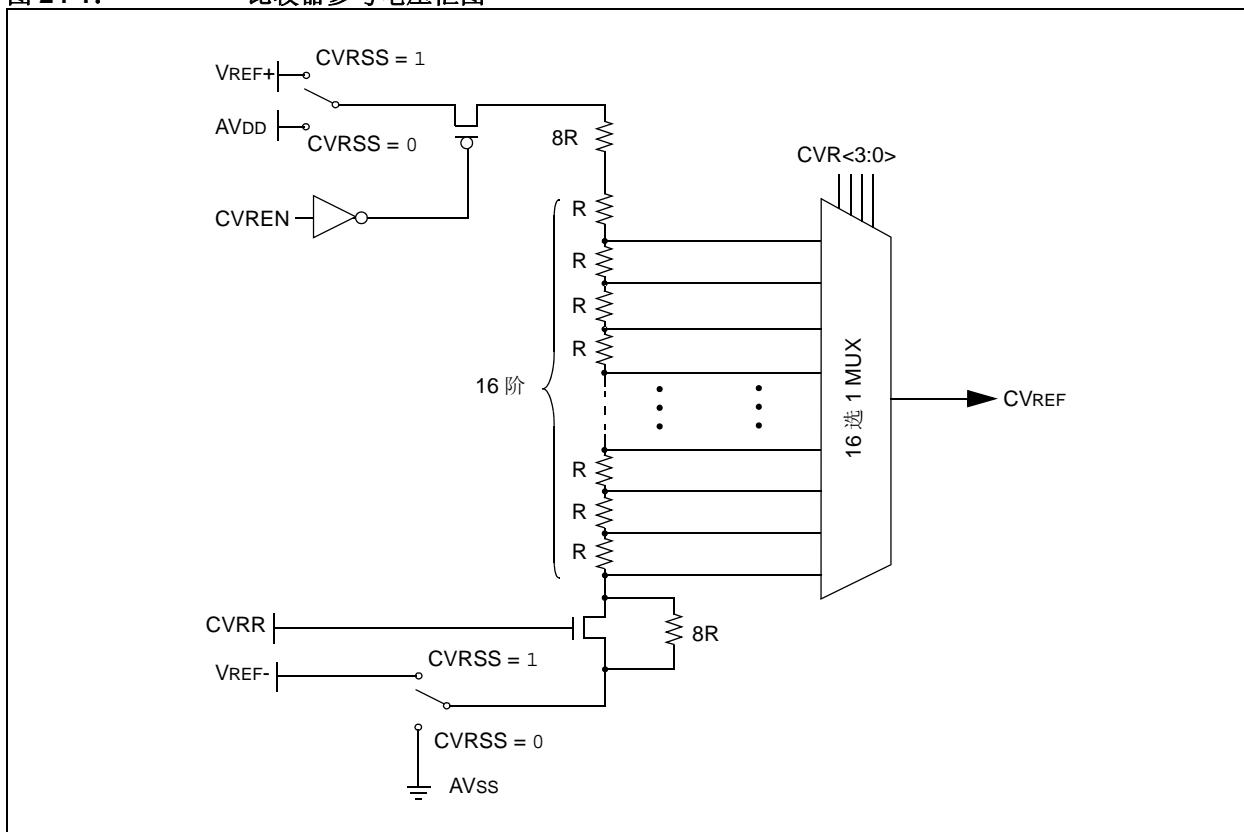
参考电压模块是通过 CVRCON 寄存器（寄存器 24-1）来控制的。比较器参考电压模块提供两种范围的输出电压，每种范围都具有 16 个不同的电平。通过 CVRR 位

(CVRCON<5>) 选择输出电压的范围。这两种范围的主要区别在于其电压值之间的步长不同（其中一种范围可提供较高的分辨率），该步长由 CVREF 选择位 (CVR<3:0>) 进行选择。

比较器参考电压模块的电压源可以来自 VDD 和 VSS，也可以来自外部 VREF+ 和 VREF-。电压源由 CVRSS 位 (CVRCON<4>) 选择。

在更改 CVREF 输出值时，必须考虑比较器参考电压的稳定时间。

图 24-1：比较器参考电压框图



PIC24FJ256GB110 系列

寄存器 24-1: CVRCON: 比较器参考电压控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0							
CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7 **CVREN:** 比较器参考电压使能位

1 = CVREF 电路上电

0 = CVREF 电路断电

bit 6 **CVROE:** 比较器 VREF 输出使能位

1 = CVREF 电平从 CVREF 引脚输出

0 = CVREF 电平与 CVREF 引脚断开

bit 5 **CVRR:** 比较器 VREF 范围选择位

1 = CVRSRC 范围应从 0 到 0.625 CVRSRC, 步长为 CVRSRC/24

0 = CVRSRC 范围应从 0.25 到 0.719 CVRSRC, 步长为 CVRSRC/32

bit 4 **CVRSS:** 比较器 VREF 源选择位

1 = 比较器参考电压源, CVRSRC = VREF+ - VREF-

0 = 比较器参考电压源, CVRSRC = AVDD - AVSS

bit 3-0 **CVR<3:0>:** 比较器 VREF 值选择位 (0 ≤ CVR3:CVR0 ≤ 15)

当 CVRR = 1 时:

CVREF = (CVR<3:0>/ 24) • (CVRSRC)

当 CVRR = 0 时:

CVREF = 1/4 • (CVRSRC) + (CVR<3:0>/32) • (CVRSRC)

25.0 充电时间测量单元 (CTMU)

注: 本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见相关的《PIC24F 系列参考手册》章节。

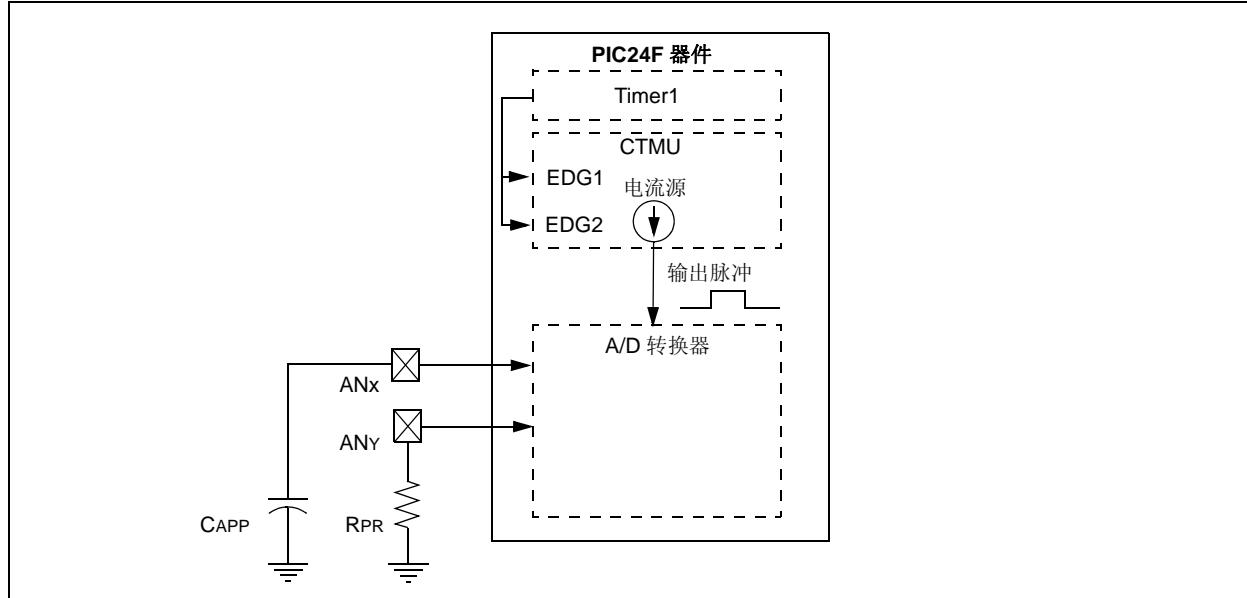
充电时间测量单元是一个灵活的模拟模块, 它提供对脉冲源输出脉冲时间差的精确测量, 以及异步脉冲生成。它的主要特性包括:

- 4 个边沿输入触发源
- 每个边沿源的极性控制
- 边沿序列控制
- 控制对边沿的响应
- 1 纳秒的时间测量分辨率
- 适合测量电容的精确电流源

CTMU 可与其他片上模拟模块一起, 用于精确测量时间、电容、电容的相对变化, 或生成独立于系统时钟的输出脉冲。CTMU 模块是与基于容性传感器接口的理想选择。

CTMU 通过两个寄存器 CTMUCON 和 CTMUICON 进行控制。CTMUCON 使能该模块, 控制边沿源选择、边沿源极性选择和边沿序列。CTMUICON 寄存器控制电流源的选择和微调。

图 25-1: 电容测量的典型连接和内部配置



25.1 测量电容

CTMU 模块通过生成脉宽和两个独立输入通道上的边沿事件间的时间宽度相等的输出脉冲, 来测量电容。两个输入通道的脉冲边沿事件都可以从 4 个来源选择: 两个内部外设模块 (OC1 和 Timer1) 和两个外部引脚 (CTEDG1 和 CTEDG2)。该脉冲和该模块的精确电流源一起使用, 可根据以下关系计算电容:

$$I = C \cdot \frac{dV}{dT}$$

测量电容时, A/D 转换器在 CTMU 输出脉冲后对其输入通道之一上的外部电容 (CAPP) 采样。由第二个通道上的高精度电阻 (RPR) 对电流源进行校准。脉冲结束后, 转换器将决定电容上的电压。电容的实际计算在软件中由应用程序执行。

图 25-1 显示了用于电容测量的外部连接, 以及该应用中 CTMU 和 A/D 模块的连接方式。该示例还显示了来自 Timer1 的边沿事件, 但使用外部边沿源的其他配置也是可能的。用 CTMU 模块测量电容和时间的详细讨论在《PIC24F 系列参考手册》中提供。

PIC24FJ256GB110 系列

25.2 测量时间

对脉冲宽度的时间测量也可类似地执行，用 A/D 模块的内部电容（CAD）和高精度电阻校准电流。图 25-2 显示了用于时间测量的外部连接，以及该应用中 CTMU 和 A/D 模块的连接方式。该示例还显示了来自外部CTEDG 引脚的两个边沿事件，但使用内部边沿源的其他配置也是可能的。用 CTMU 模块测量电容和时间的详细讨论在《PIC24F 系列参考手册》中提供。

25.3 脉冲生成和延时

CTMU 模块也可生成边沿和器件的系统时钟异步的输出脉冲。更明确地说，它可以生成距离边沿事件输入到该模块的延时可编程的脉冲。

该模块通过将 TGEN 位（CTMUCON<12>）置 1 配置为脉冲生成延时时，内部电流源连接到比较器 2 的 B 输入。电容（CDELAY）连接到比较器 2 引脚 C2INB，比较器参考电压 CVREF 连接到 C2INA。CVREF 随后配置为特定跳变点。检测到边沿事件时，该模块开始对 CDELAY 充电。CDELAY 充电达到 CVREF 跳变点时，在 CTPLS 上输出脉冲。脉冲延时的长度由 CDELAY 和 CVREF 跳变点的值决定。

图 25-3 显示了脉冲生成的外部连接，以及所需的不同模拟模块间的关系。CTEDG1 显示为输入脉冲源时，其他选项可用。用 CTMU 模块生成脉冲的详细讨论在《PIC24F 系列参考手册》中提供。

图 25-2： 测量时间的典型连接和内部配置

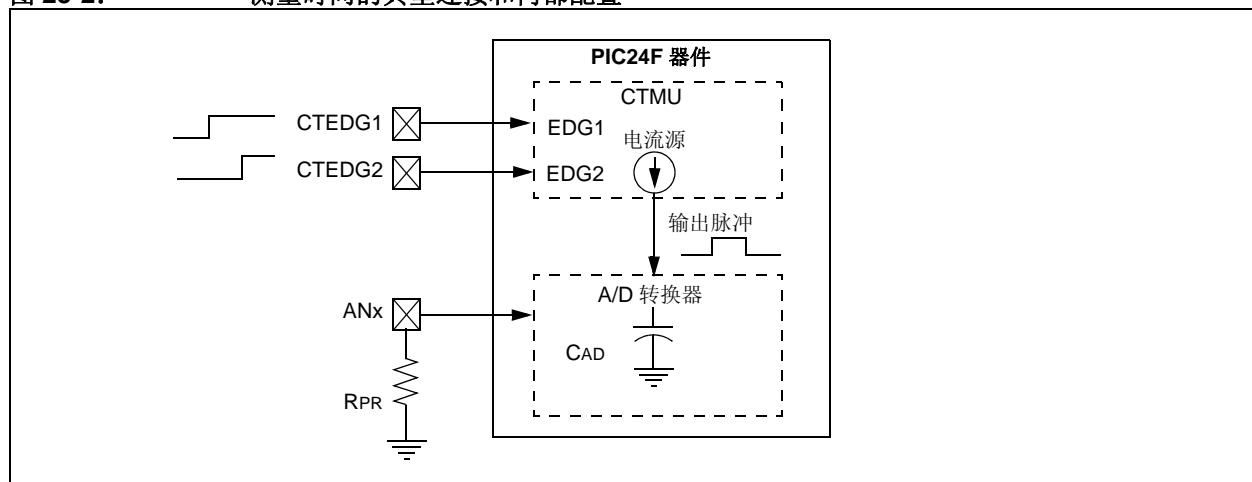
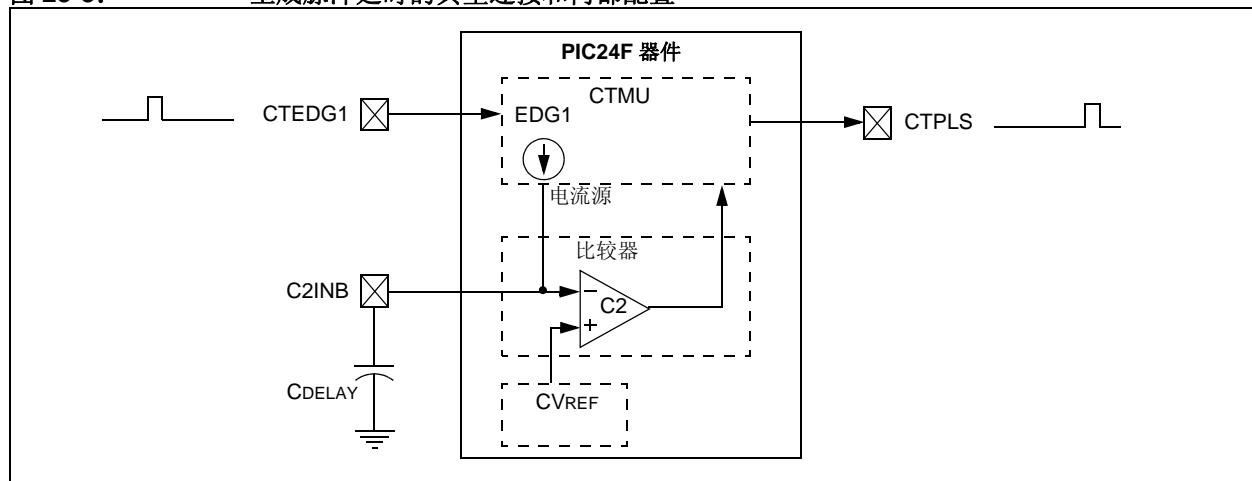


图 25-3： 生成脉冲延时的典型连接和内部配置



寄存器 25-1: CTMUCON: CTMU 控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15	CTMUEEN: CTMU 使能位 1 = 使能模块 0 = 禁止模块
bit 14	未实现: 读为 0
bit 13	CTMUSIDL: 空闲模式停止位 1 = 当器件进入空闲模式时, 模块停止工作 0 = 在空闲模式下模块继续工作
bit 12	TGEN: 时间生成使能位 ⁽¹⁾ 1 = 使能边沿时生成 0 = 禁止边沿时生成
bit 10	EDGEN: 边沿使能位 1 = 未阻止边沿 0 = 阻止边沿
bit 10	EDGSEQEN: 边沿序列使能位 1 = 边沿 1 事件必须在边沿 2 事件发生前发生 0 = 无需边沿序列
bit 9	IDISSEN: 模拟电流源控制位 1 = 模拟电流源输出接地 0 = 模拟电流源输出未接地
bit 8	CTTRIG: 触发器控制位 1 = 使能触发器输出 0 = 禁止触发器输出
bit 7	EDG2POL: 边沿 2 极性选择位 1 = 边沿 2 设定为正边沿响应 0 = 边沿 2 设定为负边沿响应
bit 6-5	EDG2SEL<1:0>: 边沿 2 源选择位 11 = CTED1 引脚 10 = CTED2 引脚 01 = OC1 模块 00 = Timer1 模块
bit 4	EDG1POL: 边沿 1 极性选择位 1 = 边沿 1 设定为正边沿响应 0 = 边沿 1 设定为负边沿响应

注 1: 如果 TGEN = 1, CTEDGx 输入和 CTPLS 输出必须在使用前分配给可用的 RPn 引脚。欲知更多信息, 请参见第 10.4 节 “外设引脚选择”。

PIC24FJ256GB110 系列

寄存器 25-1: CTMUCON: CTMU 控制寄存器 (续)

bit 3-2 **EDG1SEL<1:0>**: 边沿 1 源选择位

- 11 = CTED1 引脚
- 10 = CTED2 引脚
- 01 = OC1 模块
- 00 = Timer1 模块

bit 1 **EDG2STAT**: 边沿 2 状态位

- 1 = 已发生边沿 2 事件
- 0 = 未发生边沿 2 事件

bit 0 **EDG1STAT**: 边沿 1 状态位

- 1 = 已发生边沿 1 事件
- 0 = 未发生边沿 1 事件

注 1: 如果 TGEN = 1, CTEDGx 输入和 CTPLS 输出必须在使用前分配给可用的 RPn 引脚。欲知更多信息, 请参见第 10.4 节 “外设引脚选择”。

寄存器 25-2: CTMUICON: CTMU 电流控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-10 **ITRIM<5:0>**: 电流源微调位

- 011111 = 对称电流的最大正向调整
- 011110
-

- 000001 = 对称电流的最小正向调整
- 000000 = IRNG<1:0> 指定的标称电流输出
- 111111 = 对称电流的最小负向调整

-
- 100010
- 100001 = 对称电流的最大负向调整

bit 9-8 **IRNG<1:0>**: 电流源范围选择位

- 11 = 100 × 基本电流
- 10 = 10 × 基本电流
- 01 = 基本电流水平 (标称值为 0.55 μA)
- 00 = 禁止电流源

bit 7-0 未实现: 读为 0

26.0 特殊功能

注:	本数据手册总结了 PIC24F 器件的功能。但是不应把本手册当作无所不包的参考手册来使用。更多信息, 请参见《PIC24F 系列参考手册》的以下章节:
• 第 9 章 “看门狗定时器 (WDT)”	(DS39697A_CN)
• 第 32 章 “高级器件集成”	(DS39719A_CN)
• 第 33 章 “编程和诊断”	(DS39716A_CN)

PIC24FJ256GB110 系列器件具有几项特殊的功能旨在最大限度地提高应用的灵活性和可靠性, 并通过减少外部组件的使用将成本降至最低。提供的特殊功能包括:

- 灵活的配置
- 看门狗定时器 (WDT)
- 代码保护
- JTAG 边界扫描接口
- 在线串行编程
- 在线仿真

26.1 配置位

通过对配置位编程 (读为 0) 或不编程 (读为 1) 来选择不同的器件配置。这些配置位被映射到程序存储器以 F80000h 开始的单元中。从寄存器 26-1 到寄存器 26-5 详细解释了各配置位的不同功能。

注意, 地址 F80000h 超出了用户程序存储空间。事实上, 它属于只能使用表读和表写访问的配置存储空间 (800000h-FFFFFh)。

表 26-1: PIC24FJ256GB110 系列器件的闪存配置字位置

器件	配置字地址		
	1	2	3
PIC24FJ64GB1	ABFEh	ABFCh	ABFAh
PIC24FJ128GB1	157FEh	157FC	157FA
PIC24FJ192GB1	20BFEh	20BFC	20BFA
PIC24FJ256GB1	2ABFEh	2ABFC	2ABFA

26.1.1 配置 PIC24FJ256GB110 系列器件的注意事项

在 PIC24FJ256GB110 系列器件中, 配置字节实现为易失性存储单元。这意味着必须在每次器件上电时都设定配置数据。配置数据存储在片上程序存储空间顶部的 3 个字节中, 这些字被称为闪存配置字。它们的具体位置如表 26-1 所示。这些是实际器件配置位的打包表示, 它们的实际位置分布在配置空间的几个存储单元中。器件复位期间, 自动将配置数据从闪存配置字装入适当的配置寄存器中。

注:	发生所有类型的器件复位时, 重新装入配置数据。
-----------	-------------------------

当为这些器件创建应用程序时, 用户应该为配置数据特别分配闪存配置字单元, 以确保当编译代码时程序代码不会存储在该地址上。

程序存储器中的所有闪存配置字的高字节应该总为 1111 1111。这使得当远程事件意外执行这些存储单元时将其作为 NOP 指令来执行。由于没有在相应的存储单元中实现这些配置位, 因此向这些存储单元写入 1 不会影响器件工作。

注:	在程序存储器的最后页上执行页擦除操作会清除闪存配置字, 并使能代码保护。因此, 用户应避免在程序存储器的最后页上执行页擦除操作。
-----------	--

PIC24FJ256GB110 系列

寄存器 26-1: CW1: 闪存配置字 1

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

r-x	R/PO-1	R/PO-1	R/PO-1	R/PO-1	r-1	R/PO-1	R/PO-1
r	JTAGEN ⁽¹⁾	GCP	GWRP	DEBUG	r	ICS1	ICS0
bit 15							bit 8

R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
FWDTEN	WINDIS	—	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

图注:

r = 保留位

R = 可读位

PO = 一次编程位

U = 未实现位, 读为 0

-n = 未对器件编程时的值

1 = 置 1

0 = 清零

- bit 23-16 未实现: 读为 1
- bit 15 保留: 值未知; 编程为 0
- bit 14 **JTAGEN:** JTAG 端口使能位⁽¹⁾
1 = 使能 JTAG 端口
0 = 禁止 JTAG 端口
- bit 13 **GCP:** 通用段程序存储器代码保护位
1 = 禁止代码保护
0 = 使能整个程序存储空间的代码保护
- bit 12 **GWRP:** 通用段代码闪存写保护位
1 = 允许对程序存储器的写操作
0 = 禁止对程序存储器的写操作
- bit 11 **DEBUG:** 后台调试器使能位
1 = 器件复位至工作模式
0 = 器件复位至调试模式
- bit 10 保留: 始终保持为 1
- bit 9-8 **ICS1:ICS0:** 仿真器引脚位置选择位
11 = 仿真器功能与 PGEC1/PGED1 共用
10 = 仿真器功能与 PGEC2/PGED2 共用
01 = 仿真器功能与 PGEC3/PGED3 共用
00 = 保留; 不要使用
- bit 7 **FWDTEN:** 看门狗定时器使能位
1 = 使能看门狗定时器
0 = 禁止看门狗定时器
- bit 6 **WINDIS:** 窗口看门狗定时器禁止位
1 = 使能标准看门狗定时器
0 = 使能窗口看门狗定时器; FWDTEN 必须为 1
- bit 5 未实现: 读为 1
- bit 4 **FWPSA:** WDT 预分频比选择位
1 = 预分频比 1:128
0 = 预分频比 1:32

注 1: JTAGEN 位只能使用在线串行编程 (ICSP™) 进行修改。当通过 JTAG 接口编程器件时不能对其进行修改。

寄存器 26-1: CW1: 闪存配置字 1 (续)

bit 3-0	WDTPS<3:0> : 看门狗定时器后分频比选择位
	1111 = 1:32,768
	1110 = 1:16,384
	1101 = 1:8,192
	1100 = 1:4,096
	1011 = 1:2,048
	1010 = 1:1,024
	1001 = 1:512
	1000 = 1:256
	0111 = 1:128
	0110 = 1:64
	0101 = 1:32
	0100 = 1:16
	0011 = 1:8
	0010 = 1:4
	0001 = 1:2
	0000 = 1:1

注 1: JTAGEN 位只能使用在线串行编程 (ICSP™) 进行修改。当通过 JTAG 接口编程器件时不能对其进行修改。

PIC24FJ256GB110 系列

寄存器 26-2: CW2: 闪存配置字 2

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
IESO	PLLDIV2	PLLDIV1	PLLDIV0	PLLDIS	FNOSC2	FNOSC1	FNOSC0
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	r-1	R/PO-1	R/PO-1
FCKSM1	FCKSM0	OSCIOPFCN	IOL1WAY	DISUVREG	r	POSCMD1	POSCMD0
bit 7							bit 0

图注:

r = 保留位

R = 可读位

PO = 一次编程位

U = 未实现位, 读为 0

-n = 未对器件编程时的值

1 = 置 1

0 = 清零

bit 23-16 未实现: 读为 1

bit 15 **IESO:** 内 / 外部切换位

1 = 使能 IESO 模式 (双速启动)

0 = 禁止 IESO 模式 (双速启动)

bit 14-12 **PLLDIV<2:0>:** USB 96 MHz PLL 预分频比选择位

111 = 振荡器输入 12 分频 (48 MHz 输入)

110 = 振荡器输入 10 分频 (40 MHz 输入)

101 = 振荡器输入 6 分频 (24 MHz 输入)

100 = 振荡器输入 5 分频 (20 MHz 输入)

011 = 振荡器输入 4 分频 (16 MHz 输入)

010 = 振荡器输入 3 分频 (12 MHz 输入)

001 = 振荡器输入 2 分频 (8 MHz 输入)

000 = 直接使用振荡器输入 (4 MHz 输入)

bit 11 **PLLDIS:** USB 96 MHz PLL 禁止位

1 = 禁止 PLL

0 = 使能 PLL (所有 USB 操作均要求使能 PLL)

bit 10-8 **FNOSC<2:0>:** 初始振荡器选择位

111 = 带后分频器的快速 RC 振荡器 (FRCDIV)

110 = 保留

101 = 低功耗 RC 振荡器 (LPRC)

100 = 辅助振荡器 (SOSC)

011 = 带 PLL 模块的主振荡器 (XTPLL、HSPLL 和 ECPLL)

010 = 主振荡器 (XT、HS 和 EC)

001 = 带后分频器和 PLL 模块的快速 RC 振荡器 (FRCPLL)

000 = 快速 RC 振荡器 (FRC)

bit 7-6 **FCKSM<1:0>:** 时钟切换和故障保护时钟监视器配置位

1x = 禁止时钟切换和故障保护时钟监视器

01 = 使能时钟切换, 禁止故障保护时钟监视器

00 = 使能时钟切换和故障保护时钟监视器

bit 5 **OSCIOPFCN:** OSCO 引脚配置位

如果 POSCMD<1:0> = 11 或 00:

1 = OSCO/CLKO/RC15 用作 CLK0 (Fosc/2)

0 = OSCO/CLKO/RC15 用作端口 I/O (RC15)

如果 POSCMD<1:0> = 10 或 01:

OSCIOPFCN 对 OSCO/CLKO/RC15 没有影响。

寄存器 26-2: CW2: 闪存配置字 2 (续)

bit 4	IOL1WAY: IOLOCK 一次置 1 使能位 1 = IOLOCK 位 (OSCCON<6>) 可以置 1 一次，前提是已经完成解锁序列。置 1 后，不能再次写入外设引脚选择寄存器。 0 = IOLOCK 位可以根据需要置 1 和清零，前提是已经完成解锁序列
bit 3	DISUVREG: 内部 USB 3.3V 稳压器禁止位 1 = 禁止稳压器 0 = 使能稳压器
bit 2	保留：始终保持为 1
bit 1-0	POSCMD<1:0>: 主振荡器配置位 11 = 禁止主振荡器 10 = 选择 HS 振荡器模式 01 = 选择 XT 振荡器模式 00 = 选择 EC 振荡器模式

寄存器 26-3: CW3: 闪存配置字 3

| U-1 |
|--------|--------|--------|--------|--------|--------|--------|--------|
| — | — | — | — | — | — | — | — |
| bit 23 | | | | | | | bit 16 |
| R/PO-1 | R/PO-1 | R/PO-1 | U-1 | U-1 | U-1 | U-1 | U-1 |
| WPEND | WPCFG | WPDIS | — | — | — | — | — |
| bit 15 | | | | | | | bit 8 |
| R/PO-1 |
| WPFP7 | WPFP6 | WPFP5 | WPFP4 | WPFP3 | WPFP2 | WPFP1 | WPFP0 |
| bit 7 | | | | | | | bit 0 |

图注:

R = 可读位

PO = 一次编程位

U = 未实现位，读为 0

-n = 未对器件编程时的值

1 = 置 1

0 = 清零

bit 23-16	未实现：读为 1
bit 15	WPEND: 段写保护结束页选择位 1 = 受保护代码段的下边界位于程序存储器的底部 (000000h)；上边界是由 WPFP<7:0> 指定的代码页 0 = 受保护代码段的上边界是程序存储器的最后页；下边界是由 WPFP<7:0> 指定的代码页
bit 14	WPCFG: 配置字代码页保护选择位 1 = 最后页（位于程序存储器的顶部）和闪存配置字未受保护 0 = 最后页和闪存配置字受代码保护
bit 13	WPDIS: 段写保护禁止位 1 = 禁止段代码保护 0 = 使能段代码保护；受保护的段由 WPEND、WPCFG 和 WPFPx 配置位定义
bit 12-8	未实现：读为 1
bit 7-0	WPFP<7:0>: 受保护的代码段边界页位 指示受保护代码段的边界的 512 字程序代码页，起始于程序存储器底部的页 0。 如果 WPEND = 1: 指定代码页的末尾地址是段的上边界。 如果 WPEND = 0: 指定代码页的起始地址是段的下边界。

PIC24FJ256GB110 系列

寄存器 26-4: DEVID: 器件 ID 寄存器

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23	bit 16						

U	U	R	R	R	R	R	R
—	—	FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2
bit 15	bit 8						

R	R	R	R	R	R	R	R
FAMID1	FAMID0	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
bit 7	bit 0						

图注: R = 只读位 U = 未实现位

bit 23-14 未实现: 读为 1

bit 13-6 **FAMID<7:0>**: 器件系列标识符位

01000000 = PIC24FJ256GB110 系列

bit 5-0 **DEV<5:0>**: 各个器件的标识符位

000001 = PIC24FJ64GB106

000011 = PIC24FJ64GB108

000111 = PIC24FJ64GB110

001001 = PIC24FJ128GB106

001011 = PIC24FJ128GB108

001111 = PIC24FJ128GB110

010001 = PIC24FJ192GB106

010011 = PIC24FJ192GB108

010111 = PIC24FJ192GB110

011001 = PIC24FJ256GB106

011011 = PIC24FJ256GB108

011111 = PIC24FJ256GB110

寄存器 26-5: DEVREV: 器件版本寄存器

U	U	U	U	U	U	U	U
—	—	—	—	—	—	—	—
bit 23	bit 16						

U	U	U	U	U	U	U	R
—	—	—	—	—	—	—	MAJRV2
bit 15	bit 8						

R	R	U	U	U	R	R	R
MAJRV1	MAJRV0	—	—	—	DOT2	DOT1	DOT0
bit 7	bit 0						

图注: R = 只读位 U = 未实现位

bit 23-9 未实现: 读为 0

bit 8-6 **MAJRV<2:0>**: 主要版本标识符位

bit 5-3 未实现: 读为 0

bit 2-0 **DOT<2:0>**: 次要版本标识符位

26.2 片上稳压器

所有 PIC24FJ256GB110 系列器件使用标称值为 2.5V 的电压为其内核数字逻辑供电。对于需要工作在一个更高的典型电压值如 3.3V 的设计中，这可能会产生冲突。PIC24FJ256GB110 系列的所有器件均包含一个片上稳压器，可使器件内核逻辑运行在 VDD 下。

ENVREG 引脚控制该稳压器。把 VDD 连接到该引脚将使能稳压器，然后稳压后的电压通过其他 VDD 引脚向内核供电。当使能稳压器时，必须将一个低 ESR 电容（如陶瓷电容）连接到 VDDCORE/VCAP 引脚（图 26-1），这可帮助维持稳压器的稳定性。第 29.1 节“直流特性”中提供了该滤波电容（CEFC）的推荐值。

如果 ENVREG 与 Vss 相连，则禁止稳压器。在这种情况下，必须使用独立的 2.5V 标称值的电源通过 VDDCORE/VCAP 引脚为器件的内核逻辑供电，从而 I/O 引脚可以工作在一个较高的电平，通常为 3.3V。另外，VDDCORE/VCAP 和 VDD 引脚可以连接在一起，使器件工作在较低的标称电压下。请参见图 26-1 了解可能的配置。

26.2.1 稳压器跟踪模式和低压检测

当片上稳压器被使能时，它可以向数字内核逻辑提供标称值为 2.5V 的恒压。

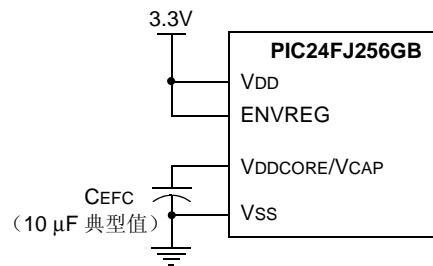
稳压器可提供从约为 2.5V 直至器件所能承受的最高电压 V_{DDMAX} 的 VDD 范围。VDD 电平降到 2.5V 以下时稳压器无法工作。为防止用于稳压器的电压降得过低时产生“欠压”条件，稳压器进入跟踪模式。在跟踪模式下，稳压器输出跟随 VDD，通常低 100 mV。

器件进入跟踪模式后，就不再能在全速下工作。为提供有关器件何时进入跟踪模式的信息，片上稳压器包含一个简单的低压检测电路。当 VDD 降到全速工作电压以下时，电路会将低压检测中断标志 LVDIF (IFS4<8>) 置 1。这可用于产生中断并将应用置于低功耗工作模式，或者触发器件受控关闭。

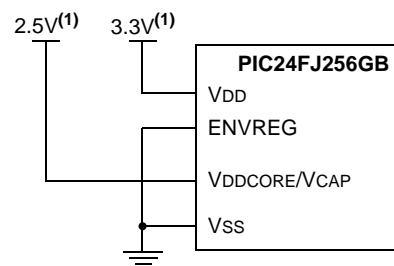
低压检测 (LVD) 仅在使能稳压器时才可用。

图 26-1：片上稳压器连接

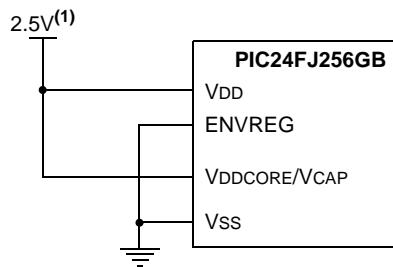
使能稳压器 (ENVREG 连接到 VDD):



禁止稳压器 (ENVREG 接地):



禁止稳压器 (VDD 连接到 VDDCORE):



注 1：这些为典型的工作电压。请参见第 29.1 节“直流特性”了解 VDD 和 VDDCORE 的全部工作电压范围。

26.2.2 片上稳压器和 POR

当使能稳压器时，它大约需要 $10\ \mu\text{s}$ 来产生输出。这段时间被称为 TVREG，在这期间禁止代码执行。每次器件在掉电（包括休眠模式）之后继续工作时，都会产生 TVREG 延时。TVREG 的长度由 PMSLP 位 (RCON<8>) 确定，如第 26.2.5 节“稳压器待机模式”所述。

如果禁止稳压器，将自动使能独立的上电延时定时器 (PWRT)。在器件启动时，PWRT 会增加一段固定的 $64\ \text{ms}$ (标称值) 的延时 (仅 POR 或 BOR)。在禁止稳压器的情况下，从休眠模式唤醒时，PMSLP 位决定唤醒时间。在禁止稳压器的情况下工作时，将 PMSLP 位置 1 会缩短器件唤醒时间。

26.2.3 片上稳压器和 BOR

当使能片上稳压器时，PIC24FJ256GB110 系列器件也会有一个简单的欠压保护功能。如果向稳压器提供的电压不足以维持跟踪电平，那么稳压器复位电路将产生欠压复位。BOR 标志位 (RCON<1>) 会捕捉该事件。欠压电压规范在《PIC24F 系列参考手册》的第 7 章“复位”(DS39712A_CN) 中提供。

26.2.4 上电要求

片上稳压器是为了满足器件的上电要求而设计的。如果应用不使用稳压器，那就必须严格遵守上电条件。在上电时，VDDCORE 不得比 VDD 高出 $0.3\ \text{伏}$ 以上。

注：更多信息，请参见第 29.0 节“电气特性”。

26.2.5 稳压器待机模式

当使能片上稳压器时，它总是消耗比 IDD/IPD 高一点的电流，包括器件处于休眠模式时，即使此时内核数字逻辑不需要能耗。要在功耗资源极其重要的应用中进一步节省功耗，可以将稳压器配置为每当器件进入休眠模式时就自动禁止。该功能由 PMSLP 位 (RCON<8>) 控制。默认情况下，该位清零使闪存程序存储器失电，从而使能待机模式。在从待机模式唤醒前，稳压器必须等待 TVREG 超时。需要这一额外时间来确保稳压器有足够的拉电流为闪存存储器供电。

对于需要较短唤醒时间的应用，可以禁止稳压器待机模式。PMSLP 位可被置 1 以关闭待机模式，从而使闪存在休眠模式下一直处于上电状态，且器件可立即被唤醒，无需等待 TVREG 超时。当 PMSLP 置 1 时，休眠模式的功耗约比允许稳压器进入待机模式时高 $40\ \mu\text{A}$ 。

26.3 看门狗定时器 (WDT)

对于 PIC24FJ256GB110 系列器件，WDT 由 LPRC 振荡器驱动。当使能 WDT 时，时钟源也将使能。

由 LPRC 提供的 WDT 时钟源的频率标称值为 $31\ \text{kHz}$ 。它提供给预分频器，后者可配置为 5 位 (32 分频) 或 7 位 (128 分频) 工作。具体工作模式由 FWPSA 配置位设置。使用 $31\ \text{kHz}$ 输入，预分频器在 5 位模式下将产生一个 $1\ \text{ms}$ 的标称 WDT 超时周期 (TWDT)，或在 7 位模式下超时周期为 $4\ \text{ms}$ 。

分频比可变的后分频器对 WDT 预分频器的输出进行分频，并扩展超时周期范围。后分频器由 WDTPS<3:0> 配置位 (CW1<3:0>) 控制，后者可选择总共 16 种设置，从 1:1 到 1:32,768。使用预分频器和后分频器可实现范围从 $1\ \text{ms}$ 到 $131\ \text{秒}$ 的超时周期。

WDT、预分频器和后分频器在以下条件下复位：

- 任何器件复位时
- 在完成时钟切换后，无论时钟切换是由软件（即，在更改 NOSC 位后将 OSWEN 位置 1）引起还是由硬件（即，故障保护时钟监视器）引起的
- 当执行 PWRSAV 指令时（即，进入休眠或空闲模式）
- 当器件退出休眠或空闲模式恢复正常工作时
- 当在正常执行过程中使用 CLRWDT 指令时

如果使能 WDT，它将在休眠或空闲模式下继续运行。当发生 WDT 超时时，将唤醒器件并且代码将继续从 PWRSAV 指令处开始执行。相应的 SLEEP 或 IDLE 位 (RCON<3:2>) 在器件唤醒后需要用软件清零。

WDT 标志位 WDTO (RCON<4>) 不会在 WDT 超时后自动清零。要检测后面的 WDT 事件，必须用软件将该标志位清零。

注：当执行 CLRWDT 和 PWRSAV 指令时，预分频器和后分频器的计数值将被清零。

26.3.1 窗口操作

看门狗定时器有一种可选的固定窗口工作模式。在该窗口模式下，CLRWDI 指令只能在 WDT 编程周期的后 1/4 复位 WDT。在该窗口前执行的 CLRWDI 指令会导致 WDT 复位，类似于 WDT 超时。

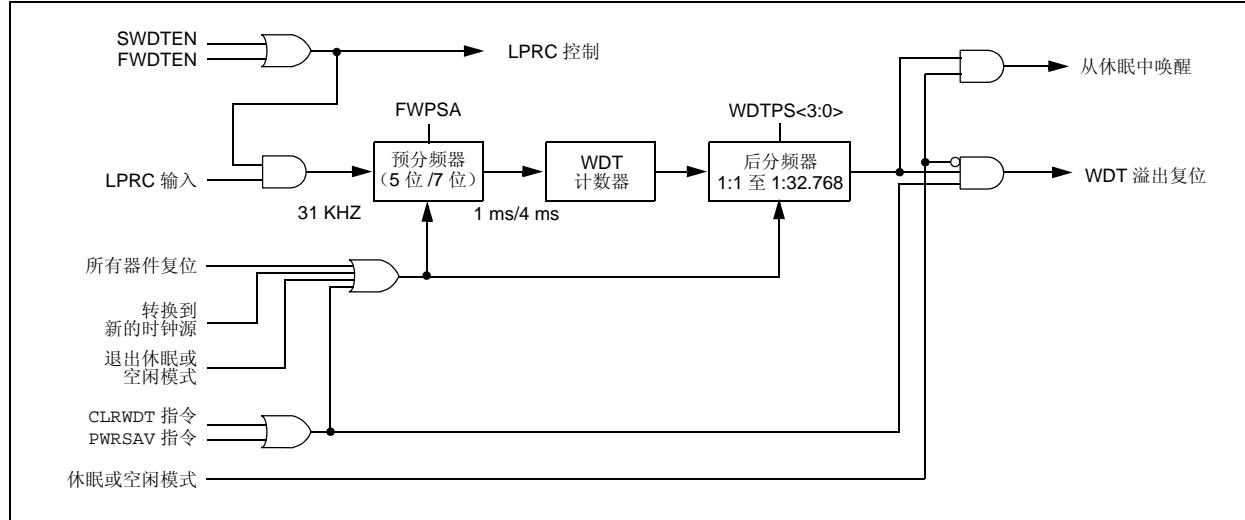
窗口 WDT 模式通过将 WINDIS 配置位（CW1<6>）编程为 0 来使能。

26.3.2 控制寄存器

WDT 通过 FWDTEN 配置位使能或禁止。当 FWDTEN 配置位置 1 时，WDT 始终是使能的。

当 FWDTEN 配置位被编程为 0 时，可以用软件有选择地控制 WDT。通过将 SWDTEN 控制位（RCON<5>）置 1 用软件使能 WDT。SWDTEN 控制位在任何器件复位时清零。软件 WDT 选项允许用户对重要代码段使能 WDT，而对不重要的代码段禁止 WDT，从而最大限度地省电。

图 26-2： WDT 框图



26.4 程序校验和代码保护

PIC24FJ256GB110 系列器件提供了两种补充方法可在改写和擦除操作时保护应用程序代码。这也有助于在运行时防止器件配置被意外更改。

26.4.1 通用段保护

对于 PIC24FJ256GB110 系列中的所有器件，片上程序存储空间被视为一个存储区，称为通用段（General Segment, GS）。配置位 GCP 控制该存储区的代码保护。该位阻止外部对程序存储空间的读和写操作。但在正常执行模式下没有直接影响。

写保护由配置字中的 GWRP 位控制。当 GWRP 被编程为 0 时，阻止内部对程序存储器的写和擦除操作。

26.4.2 代码段保护

除了全局通用段保护外，程序存储空间的独立子范围可以单独地被保护以防写和擦除操作。该区域可在需要写和擦除受保护代码的单独存储块中有多种用途，例如自举程序。与常见引导区实现不同，用户可将 PIC24FJ256GB110 系列器件中特定的受保护段放在程序空间中的任何位置，并且其大小可以在较宽范围内进行配置。

代码段保护还增加了对程序存储器的指定区域的新保护级别，这是通过在写或擦除地址发生在指定范围内时禁止 NVM 安全互锁实现的。它们不会改写由 GCP 或 GWRP 位控制的通用段保护。例如，如果使能了 GCP 和 GWRP，则使能程序存储器下半部分的段代码保护不会撤消对上半部分的通用段保护。

PIC24FJ256GB110 系列

段代码保护的大小和类型由配置字 3 中的 WPFPx、WPEND、WPCFG 和 WPDIS 位配置。代码段保护通过设定 WPDIS 位 (= 0) 使能。WPFP 位指定要保护的段的大小，这是通过指定受保护段的起始或结束地址的 512 字的代码页实现的。指定的区域包括在内，因此该页也将受保护。

WPEND 位决定受保护段是否使用程序空间的顶部或底部作为边界。设定 WPEND (= 0) 会将程序存储器的底部 (000000h) 设置为受保护段的下边界。将 WPEND 保留为未编程 (= 1) 会保护指定页到已实现程序存储器的最后页，包括配置字单元。

独立位 WPCFG 用于独立地保护程序空间的最后页，包括闪存配置字。设定 WPCFG (= 0) 会保护最后页，与其他位设置无关。在存储器底部的代码段和闪存配置字都需要写保护的情况下，这可能有用。

段代码保护的各选项如表 26-2 所示。

表 26-2：段代码保护配置选项

段配置位			代码段的写 / 擦除保护
WPDIS	WPEND	WPCFG	
1	X	X	未使能其他保护；所有程序存储器保护由 GCP 和 GWRP 配置
0	1	X	地址从由 WPFP<7:0> 定义的代码页的起始地址到已实现程序存储器的结束地址（包括在内）受擦 / 写保护，包括闪存配置字
0	0	1	地址 000000h 到由 WPFP<7:0> 定义的代码页的最后地址（包括在内）受擦 / 写保护
0	0	0	地址 000000h 到由 WPFP<7:0> 定义的代码页的最后地址（包括在内）受擦 / 写保护，最后一页也受擦 / 写保护

26.4.3 配置寄存器保护

有两种方法保护配置寄存器使其免遭意外的或不想要的更改或读取。主要保护方法与 RP 寄存器的相同——影子寄存器包含一个补充值不断地与实际值进行比较。

为防止不可预见的事件造成损害，由于单独的存储单元故障（如 ESD 事件）产生的配置位更改将导致奇偶校验错误并触发器件复位。

配置寄存器的数据来自于程序存储器中的闪存配置字。当 GCP 位置 1 时，也将保护器件配置的源数据。即使未使能通用段保护，器件配置也能通过使用相应的代码联合保护（code cement protection）设置进行保护。

26.5 JTAG 接口

PIC24FJ256GB110 系列器件实现了一个 JTAG 接口，支持边界扫描器件测试。

26.6 在线串行编程

可以在最终的应用电路中对 PIC24FJ256GB110 系列单片机进行串行编程。只需要 5 根线即可实现这一操作，其中时钟线（PGECx）、数据线（PGEDx）各一根，其余 3 根分别是电源线、接地线和编程电压线。这允许用户在生产电路板时使用未编程器件，仅在产品交付之前才对单片机进行编程，从而可以使用最新版本的固件或者定制固件进行编程。

26.7 在线调试器

当选择 MPLAB® ICD 2 作为调试器时，使能在线调试功能。该功能允许与 MPLAB IDE 配合使用进行简单的调试。通过 PGECx（仿真 / 调试时钟）和 PGEDx（仿真 / 调试数据）引脚控制调制功能。

要使用器件的在线调试功能，应用设计必须实现至 MCLR、VDD、VSS 以及由 ICS 配置位指定的 PGECx/PGEDx 引脚对的 ICSP 连接。此外，当使能该功能时，某些资源就不能用于一般用途了。这些资源包括数据 RAM 的前 80 字节和两个 I/O 引脚。

PIC24FJ256GB110 系列

注:

27.0 开发支持

一系列软件及硬件开发工具对 PIC® 单片机和 dsPIC® 数字信号控制器提供支持：

- 集成开发环境
 - MPLAB® IDE 软件
- 编译器 / 汇编器 / 链接器
 - 适用于各种器件系列的 MPLAB C 编译器
 - 适用于各种器件系列的 HI-TECH C 编译器
 - MPASM™ 汇编器
 - MPLINK™ 目标链接器 /
MPLIB™ 目标库管理器
 - 适用于各种器件系列的 MPLAB 汇编器 / 链接器 / 库管理器
- 模拟器
 - MPLAB SIM 软件模拟器
- 仿真器
 - MPLAB REAL ICE™ 在线仿真器
- 在线调试器
 - MPLAB ICD 3
 - PICkit™ 3 Debug Express
- 器件编程器
 - PICkit™ 2 编程器
 - MPLAB PM3 器件编程器
- 低成本演示 / 开发板、评估工具包及入门工具包

27.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16/32 位单片机市场提供了前所未有的易于使用的软件开发平台。MPLAB IDE 是基于 Windows® 操作系统的应用软件，包括：

- 一个包含所有调试工具的图形界面
 - 模拟器
 - 编程器（单独销售）
 - 在线仿真器（单独销售）
 - 在线调试器（单独销售）
 - 具有彩色上下文代码显示的全功能编辑器
 - 多项目管理器
 - 内容可直接编辑的可定制式数据窗口
 - 高级源代码调试
 - 鼠标停留在变量上进行查看的功能
 - 将变量从源代码窗口拖放到 Watch（观察）窗口
 - 丰富的在线帮助
 - 集成了可选的第三方工具，如 IAR C 编译器
- MPLAB IDE 可以让您：
- 编辑源文件（C 语言或汇编语言）
 - 点击一次即可完成编译或汇编，并将代码下载到仿真器和模拟器工具中（自动更新所有项目信息）
 - 可使用如下各项进行调试：
 - 源文件（C 语言或汇编语言）
 - 混合 C 语言和汇编语言
 - 机器码

MPLAB IDE 在单个开发范例中支持使用多种调试工具，包括从成本效益高的模拟器到低成本的在线调试器，再到全功能的仿真器。这样缩短了用户升级到更加灵活而功能强大的工具时的学习时间。

27.2 适用于各种器件系列的 MPLAB C 编译器

MPLAB C 编译器代码开发系统是完全的 ANSI C 编译器，适用于 Microchip 的 PIC18、PIC24 和 PIC32 系列单片机及 dsPIC30 和 dsPIC33 系列数字信号控制器。这些编译器提供强大的集成功能和出众的代码优化能力，且使用方便。

为便于源代码调试，编译器提供针对 MPLAB IDE 调试器优化的符号信息。

27.3 适用于各种器件系列的 HI-TECH C 编译器

HI-TECH C 编译器代码开发系统是完全的 ANSI C 编译器，适用于 Microchip 的 PIC 系列单片机及 dsPIC 系列数字信号控制器。这些编译器提供强大的集成功能和全知代码生成能力，且使用方便。

为便于源代码调试，编译器提供针对 MPLAB IDE 调试器优化的符号信息。

编译器包括一个宏汇编器、链接器、预处理程序和单步驱动程序，可以在多种平台上运行。

27.4 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于 PIC10/12/16/18 MCU。

MPASM 汇编器可生成用于 MPLINK 目标链接器的可重定位目标文件、Intel® 标准 HEX 文件、详细描述存储器使用状况和符号参考的 MAP 文件、包含源代码行及生成机器码的绝对 LST 文件以及用于调试的 COFF 文件。

MPASM 汇编器具有如下特性：

- 集成在 MPLAB IDE 项目中
- 用户定义的宏可简化汇编代码
- 对多用途源文件进行条件汇编
- 允许完全控制汇编过程的指令

27.5 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器包含了由 MPASM 汇编器、MPLAB C18 C 编译器产生的可重定位目标。通过使用链接器脚本中的指令，它还可链接预编译库中的可重定位目标。

MPLIB 目标库管理器管理预编译代码库文件的创建和修改。当从源文件调用库中的一段子程序时，只有包含此子程序的模块被链接到应用程序。这样可使大型库在许多不同应用中被高效地利用。

目标链接器 / 库管理器具有如下特性：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

27.6 适用于各种器件系列的 MPLAB 汇编器、链接器和库管理器

MPLAB 汇编器为 PIC24、PIC32 和 dsPIC 器件从符号汇编语言生成可重定位机器码。MPLAB C 编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特性：

- 支持整个器件指令集
- 支持定点数据和浮点数据
- 命令行界面
- 丰富的指令集
- 灵活的宏语言
- MPLAB IDE 兼容性

27.7 MPLAB SIM 软件模拟器

MPLAB SIM 软件模拟器通过在指令级对 PIC MCU 和 dsPIC® DSC 进行模拟，可在 PC 主机环境下进行代码开发。对于任何给定的指令，都可以对数据区进行检查或修改，并通过一个全面的激励控制器来施加激励。可以将各寄存器记录在文件中，以便进行进一步的运行时分析。跟踪缓冲区和逻辑分析器的显示使软件模拟器还能记录和跟踪程序的执行、I/O 的动作、大部分的外设及内部寄存器。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C 编译器以及 MPASM 和 MPLAB 汇编器的符号调试。该软件模拟器可用于在硬件实验室环境外灵活地开发和调试代码，是一款完美且经济的软件开发工具。

27.8 MPLAB REAL ICE 在线仿真器系统

MPLAB REAL ICE 在线仿真器系统是 Microchip 针对其闪存 DSC 和 MCU 器件而推出的新一代高速仿真器。结合 MPLAB 集成开发环境（IDE）所具有的易于使用且功能强大的图形用户界面，该仿真器可对 PIC® 闪存 MCU 和 dsPIC® 闪存 DSC 进行调试和编程。IDE 是随每个工具包一起提供的。

该仿真器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与在线调试器系统兼容的连接器（RJ11）或新型抗噪声、高速低压差分信号（LVDS）互连电缆（CAT5）与目标板相连。

可通过 MPLAB IDE 下载将来版本的固件，对该仿真器进行现场升级。在即将推出的 MPLAB IDE 版本中，会支持许多新器件，还将增加一些新特性。在同类仿真器中，MPLAB REAL ICE 的优势十分明显：低成本、全速仿真、运行时变量查看、跟踪分析、复杂断点、耐用的探针接口及较长（长达 3 米）的互连电缆。

27.9 MPLAB ICD 3 在线调试器系统

MPLAB ICD 3 在线调试器系统是 Microchip 成本效益最高的高速硬件调试器 / 编程器，适用于 Microchip 闪存数字信号控制器（DSC）和单片机（MCU）器件。结合 MPLAB 集成开发环境（IDE）所具有的功能强大但易于使用的图形用户界面，该调试器可对 PIC® 闪存单片机和 dsPIC® DSC 进行调试和编程。

MPLAB ICD 3 在线调试器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与 MPLAB ICD 2 或 MPLAB REAL ICE 系统兼容的连接器（RJ-11）与目标板相连。MPLAB ICD 3 支持所有 MPLAB ICD 2 转接器。

27.10 PICkit 3 在线调试器 / 编程器及 PICkit 3 Debug Express

结合 MPLAB 集成开发环境（IDE）所具有的功能强大的图形用户界面，MPLAB PICkit 3 可对 PIC® 闪存单片机和 dsPIC® 数字信号控制器进行调试和编程，且价位较低。MPLAB PICkit 3 通过全速 USB 接口与设计工程师的 PC 相连，并利用 Microchip 调试（RJ-11）连接器（与 MPLAB ICD 3 和 MPLAB REAL ICE 兼容）与目标板相连。连接器使用两个器件 I/O 引脚和复位线来实现在线调试和在线串行编程。

PICkit 3 Debug Express 包括 PICkit 3、演示板和单片机、连接电缆和光盘（内含用户指南、课程、教程、编译器和 MPLAB IDE 软件）。

PIC24FJ256GB110 系列

27.11 PICkit 2 开发编程器 / 调试器及 PICkit 2 Debug Express

PICkit™ 2 开发编程器 / 调试器是一款低成本开发工具，具有易于使用的界面，适用于对 Microchip 的闪存系列单片机进行编程和调试。这一全功能的 Windows® 编程界面支持低档 (PIC10F、PIC12F5xx 和 PIC16F5xx)、中档 (PIC12F6xx 和 PIC16F)、PIC18F、PIC24、dsPIC30、dsPIC33 和 PIC32 系列的 8 位、16 位及 32 位单片机，以及许多 Microchip 串行 EEPROM 产品。结合 Microchip 功能强大的 MPLAB 集成开发环境 (IDE)，PICkit 2 可对大多数 PIC® 单片机进行在线调试。即使 PIC 单片机已嵌入应用，在线调试功能仍可以运行、暂停和单步执行程序。在断点处暂停时，可以检查和修改文件寄存器。

PICkit 2 Debug Express 包括 PICkit 2、演示板和单片机、连接电缆和光盘（内含用户指南、课程、教程、编译器和 MPLAB IDE 软件）。

27.12 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款符合 CE 规范的通用器件编程器，在 VDDMIN 和 VDDMAX 点对其可编程电压进行校验以确保可靠性最高。它有一个用来显示菜单和错误消息的大 LCD 显示器 (128 x 64)，以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSPTM 电缆。在单机模式下，MPLAB PM3 器件编程器不必与 PC 相连即可对 PIC 器件进行读取、校验和编程。在该模式下它还可设置代码保护。MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。MPLAB PM3 具备高速通信能力以及优化算法，可对具有大存储器的器件进行快速编程。它还包含了 MMC 卡，用于文件存储及数据应用。

27.13 演示 / 开发板、评估工具包及入门工具包

有许多演示、开发和评估板可用于各种 PIC MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于检查和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、RS-232 接口、LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 和 dsPICDEM™ 演示 / 开发板系列电路外，Microchip 还有一系列评估工具包和演示软件，适用于模拟滤波器设计、KEELOQ® 数据安全产品 IC、CAN、IrDA®、PowerSmart 电池管理、SEEVVAL® 评估系统、Σ-Δ ADC、流速传感器，等等。

同时还提供入门工具包，其中包含体验指定器件功能所需的所有软硬件。通常提供单个应用以及调试功能，都包含在一块电路板上。

有关演示、开发和评估工具包的完整列表，请访问 Microchip 网站 (www.microchip.com)。

28.0 指令集汇总

注: 本章简要概述了 PIC24F 指令集架构,但是不应将其当作无所不包的参考手册来使用。

与以前的 PIC® MCU 指令集相比, PIC24F 指令集添加了许多增强功能,同时保持了易于从以前 PIC MCU 指令集移植的特点。大部分指令的长度为一个程序存储字。只有三条指令需要两个程序存储单元。

每一条单字指令长 24 位,分为一个指定指令类型的 8 位操作码和进一步指定指令操作的一个或多个操作数。指令集是高度正交的,分为 4 个基本类别:

- 针对字或字节的操作
- 针对位的操作
- 立即数操作
- 控制操作

表 28-1 给出了在说明指令时使用的通用符号。表 28-2 是 PIC24F 指令集汇总,它列出了所有指令,以及每条指令影响的状态标志位。

大多数针对字或字节的 W 寄存器指令(包括桶形移位指令)有三个操作数:

- 第一个源操作数通常是寄存器 Wb, 不带任何地址修改量
- 第二个源操作数通常是寄存器 Ws, 带或不带地址修改量
- 运算结果的目标寄存器, 通常是寄存器 Wd, 带或不带地址修改量

不过,针对字或字节的文件寄存器指令只有两个操作数:

- 文件寄存器由 f 值指定
- 目标寄存器可以是文件寄存器 f 或 W0 寄存器(用 WREG 表示)

大多数位操作类指令(包括简单的循环 / 移位指令)有两个操作数:

- W 寄存器(带或不带地址修改量)或文件寄存器(由 Ws 或 f 的值指定)
- W 寄存器或文件寄存器中的位(由一个立即数指定,或者由寄存器 Wb 的内容间接指定)

涉及数据传送的立即数指令,可以使用下列操作数:

- 要被装入到 W 寄存器或文件寄存器中的立即数(由 k 值指定)
- 要装入立即数的 W 寄存器或文件寄存器(由 Wb 或 f 指定)

然而,涉及算术或逻辑运算的立即数指令,使用如下的操作数:

- 第一个源操作数是寄存器 Wb, 不带任何地址修改量
- 第二个源操作数是立即数
- 操作结果的目标寄存器(仅在与第一个源操作数不同时)通常是寄存器 Wd(带或不带地址修改量)

控制指令可以使用下列操作数:

- 程序存储地址
- 表读和表写指令的模式

除了某些双字指令外,所有指令都是单字指令;双字指令之所以是双字长的(48 位),是因为要用 48 位来提供所需信息。在第二个字中,8 个 MSb 全为 0。如果指令自身把第二个字当作一条指令来执行的话,它将作为一条 NOP 指令来执行。

大多数单字长指令都在一个指令周期内执行,除非条件测试为真或者指令执行结果改变了程序计数器。对于上述两种特殊情况,指令执行需要两个指令周期,在第二个指令周期中执行一条 NOP 指令。值得注意的例外是 BRA(无条件/计算转移)、间接 CALL/GOTO、所有的表读和表写以及 RETURN/RETFIE 指令,它们是单字长指令,但执行需要两个或三个周期。

某些与跳过后续指令有关的指令,如果要执行跳过的话,可能需要两个或三个周期,这取决于被跳过的指令是单字还是双字指令。此外,双字传送需要两个周期。双字指令执行需要两个指令周期。

PIC24FJ256GB110 系列

表 28-1：操作码说明中使用的符号

字段	说明
#text	表示由 text 定义的立即数
(text)	表示 text 的内容
[text]	表示由 text 寻址的存储单元
{ }	可选字段或操作
<n:m>	寄存器位域
.b	字节模式选择
.d	双字模式选择
.S	影子寄存器选择
.w	字模式选择（默认）
bit4	4 位位选择字段（用于字寻址指令） $\in \{0...15\}$
C, DC, N, OV, Z	MCU 状态位：进位、半进位、负、溢出和全零标志位
Expr	绝对地址、标号或表达式（由链接器解析）
f	文件寄存器地址 $\in \{0000h...1FFFh\}$
lit1	1 位无符号立即数 $\in \{0,1\}$
lit4	4 位无符号立即数 $\in \{0...15\}$
lit5	5 位无符号立即数 $\in \{0...31\}$
lit8	8 位无符号立即数 $\in \{0...255\}$
lit10	10 位无符号立即数，对于字节模式， $\in \{0...255\}$ ；对于字模式， $\in \{0:1023\}$
lit14	14 位无符号立即数 $\in \{0...16383\}$
lit16	16 位无符号立即数 $\in \{0...65535\}$
lit23	23 位无符号立即数 $\in \{0...8388607\}$ ； LSB 必须为 0
None	字段无需内容，可为空
PC	程序计数器
Slit10	10 位有符号立即数 $\in \{-512...511\}$
Slit16	16 位有符号立即数 $\in \{-32768...32767\}$
Slit6	6 位有符号立即数 $\in \{-16...16\}$
Wb	基本 W 寄存器 $\in \{W0..W15\}$
Wd	目标 W 寄存器 $\in \{Wd, [Wd], [Wd++], [Wd-], [++Wd], [-Wd]\}$
Wdo	目标 W 寄存器 $\in \{Wnd, [Wnd], [Wnd++], [Wnd-], [++Wnd], [--Wnd], [Wnd+Wb]\}$
Wm,Wn	被除数和除数工作寄存器对（直接寻址）
Wn	16 个工作寄存器之一 $\in \{W0..W15\}$
Wnd	16 个目标工作寄存器之一 $\in \{W0..W15\}$
Wns	16 个源工作寄存器之一 $\in \{W0..W15\}$
WREG	W0（文件寄存器指令中使用的工作寄存器）
Ws	源 W 寄存器 $\in \{Ws, [Ws], [Ws++], [Ws-], [++Ws], [-Ws]\}$
Wso	源 W 寄存器 $\in \{ Wns, [Wns], [Wns++], [Wns-], [++Wns], [-Wns], [Wns+Wb] \}$

表 28-2: 指令集概述

汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
ADD	ADD f	f = f + WREG	1	1	C, DC, N, OV, Z
	ADD f, WREG	WREG = f + WREG	1	1	C, DC, N, OV, Z
	ADD #lit10, Wn	Wd = lit10 + Wd	1	1	C, DC, N, OV, Z
	ADD Wb, Ws, Wd	Wd = Wb + Ws	1	1	C, DC, N, OV, Z
	ADD Wb, #lit5, Wd	Wd = Wb + lit5	1	1	C, DC, N, OV, Z
ADDC	ADDC f	f = f + WREG + (C)	1	1	C, DC, N, OV, Z
	ADDC f, WREG	WREG = f + WREG + (C)	1	1	C, DC, N, OV, Z
	ADDC #lit10, Wn	Wd = lit10 + Wd + (C)	1	1	C, DC, N, OV, Z
	ADDC Wb, Ws, Wd	Wd = Wb + Ws + (C)	1	1	C, DC, N, OV, Z
	ADDC Wb, #lit5, Wd	Wd = Wb + lit5 + (C)	1	1	C, DC, N, OV, Z
AND	AND f	f = f .AND.WREG	1	1	N, Z
	AND f, WREG	WREG = f .AND.WREG	1	1	N, Z
	AND #lit10, Wn	Wd = lit10 .AND.Wd	1	1	N, Z
	AND Wb, Ws, Wd	Wd = Wb .AND.Ws	1	1	N, Z
	AND Wb, #lit5, Wd	Wd = Wb .AND. lit5	1	1	N, Z
ASR	ASR f	f = 算术右移 f	1	1	C, N, OV, Z
	ASR f, WREG	WREG = 算术右移 f	1	1	C, N, OV, Z
	ASR Ws, Wd	Wd = 算术右移 Ws	1	1	C, N, OV, Z
	ASR Wb, Wns, Wnd	Wnd = 将 Wb 算术右移 Wns 位	1	1	N, Z
	ASR Wb, #lit5, Wnd	Wnd = 将 Wb 算术右移 lit5 位	1	1	N, Z
BCLR	BCLR f, #bit4	将 f 中的指定位清零	1	1	无
	BCLR Ws, #bit4	将 Ws 中的指定位清零	1	1	无
BRA	BRA C, Expr	如果有进位则转移	1	1 (2)	无
	BRA GE, Expr	如果大于或等于则转移	1	1 (2)	无
	BRA GEU, Expr	如果无符号大于或等于则转移	1	1 (2)	无
	BRA GT, Expr	如果大于则转移	1	1 (2)	无
	BRA GTU, Expr	如果无符号大于则转移	1	1 (2)	无
	BRA LE, Expr	如果小于或等于则转移	1	1 (2)	无
	BRA LEU, Expr	如果无符号小于或等于则转移	1	1 (2)	无
	BRA LT, Expr	如果小于则转移	1	1 (2)	无
	BRA LTU, Expr	如果无符号小于则转移	1	1 (2)	无
	BRA N, Expr	如果为负则转移	1	1 (2)	无
	BRA NC, Expr	如果没有进位则转移	1	1 (2)	无
	BRA NN, Expr	如果不为负则转移	1	1 (2)	无
	BRA NOV, Expr	如果未溢出则转移	1	1 (2)	无
	BRA NZ, Expr	如果不为零则转移	1	1 (2)	无
	BRA OV, Expr	如果溢出则转移	1	1 (2)	无
	BRA Expr	无条件转移	1	2	无
	BRA Z, Expr	如果为零则转移	1	1 (2)	无
	BRA Wn	计算转移	1	2	无
BSET	BSET f, #bit4	将 f 中的指定位置 1	1	1	无
	BSET Ws, #bit4	将 Ws 中的指定位置 1	1	1	无
BSW	BSW.C Ws, Wb	将 C 位内容写入 Ws<Wb>	1	1	无
	BSW.Z Ws, Wb	将 Z 位内容写入 Ws<Wb>	1	1	无
BTG	BTG f, #bit4	将 f 中的指定位翻转	1	1	无
	BTG Ws, #bit4	将 Ws 中的指定位翻转	1	1	无
BTSC	BTSC f, #bit4	对 f 中的指定位进行测试, 如果为零则跳过	1	1 (2 或 3)	无
	BTSC Ws, #bit4	对 Ws 中的指定位进行测试, 如果为零则跳过	1	1 (2 或 3)	无

PIC24FJ256GB110 系列

表 28-2: 指令集概述 (续)

汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
BTSS	BTSS f,#bit4	对 f 中的指定位进行测试, 如果为 1 则跳过	1	1 (2 或 3)	无
	BTSS Ws,#bit4	对 Ws 中的指定位进行测试, 如果为 1 则跳过	1	1 (2 或 3)	无
BTST	BTST f,#bit4	对 f 中的指定位进行测试	1	1	Z
	BTST.C Ws,#bit4	对 Ws 中的指定位进行测试, 并将其值存储到 C	1	1	C
	BTST.Z Ws,#bit4	对 Ws 中的指定位进行测试, 并将其反码存储到 Z	1	1	Z
	BTST.C Ws,Wb	对 Ws<Wb> 位进行测试, 并将其值存储到 C	1	1	C
	BTST.Z Ws,Wb	对 Ws<Wb> 位进行测试, 并将其反码存储到 Z	1	1	Z
BTSTS	BTSTS f,#bit4	对 f 中的指定位进行测试, 并将 f 中的该位置 1	1	1	Z
	BTSTS.C Ws,#bit4	对 Ws 中的指定位进行测试, 并将其值存储到 C, 然后将 Ws 中的该位置 1	1	1	C
	BTSTS.Z Ws,#bit4	对 Ws 中的指定位进行测试, 并将其反码存储到 Z, 然后将 Ws 中的该位置 1	1	1	Z
CALL	CALL lit23	调用子程序	2	2	无
	CALL Wn	间接调用子程序	1	2	无
CLR	CLR f	f = 0x0000	1	1	无
	CLR WREG	WREG = 0x0000	1	1	无
	CLR Ws	Ws = 0x0000	1	1	无
CLRWDT	CLRWDT	将看门狗定时器清零	1	1	WDTO, Sleep
COM	COM f	f = \bar{f}	1	1	N, Z
	COM f,WREG	WREG = \bar{f}	1	1	N, Z
	COM Ws,Wd	Wd = \bar{Ws}	1	1	N, Z
CP	CP f	比较 f 和 WREG	1	1	C, DC, N, OV, Z
	CP Wb,#lit5	比较 Wb 和 lit5	1	1	C, DC, N, OV, Z
	CP Wb,Ws	比较 Wb 和 Ws (Wb - Ws)	1	1	C, DC, N, OV, Z
CP0	CP0 f	比较 f 和 0x0000	1	1	C, DC, N, OV, Z
	CP0 Ws	比较 Ws 和 0x0000	1	1	C, DC, N, OV, Z
CPB	CPB f	带借位比较 f 和 WREG	1	1	C, DC, N, OV, Z
	CPB Wb,#lit5	带借位比较 Wb 和 lit5	1	1	C, DC, N, OV, Z
	CPB Wb,Ws	带借位比较 Wb 和 Ws (Wb - Ws - C)	1	1	C, DC, N, OV, Z
CPSEQ	CPSEQ Wb,Wn	比较 Wb 和 Wn, 如果相等则跳过	1	1 (2 或 3)	无
CPSGT	CPSGT Wb,Wn	比较 Wb 和 Wn, 如果大于则跳过	1	1 (2 或 3)	无
CPSLT	CPSLT Wb,Wn	比较 Wb 和 Wn, 如果小于则跳过	1	1 (2 或 3)	无
CPSNE	CPSNE Wb,Wn	比较 Wb 和 Wn, 如果不相等则跳过	1	1 (2 或 3)	无
DAW	DAW.b Wn	Wn = 十进制调整 Wn	1	1	C
DEC	DEC f	f = f - 1	1	1	C, DC, N, OV, Z
	DEC f,WREG	WREG = f - 1	1	1	C, DC, N, OV, Z
	DEC Ws,Wd	Wd = Ws - 1	1	1	C, DC, N, OV, Z
DEC2	DEC2 f	f = f - 2	1	1	C, DC, N, OV, Z
	DEC2 f,WREG	WREG = f - 2	1	1	C, DC, N, OV, Z
	DEC2 Ws,Wd	Wd = Ws - 2	1	1	C, DC, N, OV, Z
DISI	DISI #lit14	在 k 个指令周期内禁止中断	1	1	无
DIV	DIV.SW Wm,Wn	有符号 16/16 位整数除法	1	18	N, Z, C, OV
	DIV.SD Wm,Wn	有符号 32/16 位整数除法	1	18	N, Z, C, OV
	DIV.UW Wm,Wn	无符号 16/16 位整数除法	1	18	N, Z, C, OV
	DIV.UD Wm,Wn	无符号 32/16 位整数除法	1	18	N, Z, C, OV
EXCH	EXCH Wns,Wnd	交换 Wns 和 Wnd 的内容	1	1	无
FF1L	FF1L Ws,Wnd	从左边 (MSb) 查找第一个 1	1	1	C
FF1R	FF1R Ws,Wnd	从右边 (LSb) 查找第一个 1	1	1	C

表 28-2: 指令集概述 (续)

汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
GOTO	GOTO Expr	转移到地址	2	2	无
	GOTO Wn	间接转移到地址	1	2	无
INC	INC f	f = f + 1	1	1	C, DC, N, OV, Z
	INC f, WREG	WREG = f + 1	1	1	C, DC, N, OV, Z
	INC Ws, Wd	Wd = Ws + 1	1	1	C, DC, N, OV, Z
INC2	INC2 f	f = f + 2	1	1	C, DC, N, OV, Z
	INC2 f, WREG	WREG = f + 2	1	1	C, DC, N, OV, Z
	INC2 Ws, Wd	Wd = Ws + 2	1	1	C, DC, N, OV, Z
IOR	IOR f	f = f .IOR.WREG	1	1	N, Z
	IOR f, WREG	WREG = f .IOR.WREG	1	1	N, Z
	IOR #lit10, Wn	Wd = lit10 .IOR.Wd	1	1	N, Z
	IOR Wb, Ws, Wd	Wd = Wb .IOR.Ws	1	1	N, Z
	IOR Wb, #lit5, Wd	Wd = Wb .IOR.lit5	1	1	N, Z
LNK	LINK #lit14	分配堆栈帧	1	1	无
LSR	LSR f	f = 逻辑右移 f	1	1	C, N, OV, Z
	LSR f, WREG	WREG = 逻辑右移 f	1	1	C, N, OV, Z
	LSR Ws, Wd	Wd = 逻辑右移 Ws	1	1	C, N, OV, Z
	LSR Wb, Wns, Wnd	Wnd = 将 Wb 逻辑右移 Wns 位	1	1	N, Z
	LSR Wb, #lit5, Wnd	Wnd = 将 Wb 逻辑右移 lit5 位	1	1	N, Z
MOV	MOV f, Wn	将 f 中的内容送入 Wn	1	1	无
	MOV [Wns+Slit10], Wnd	将 [Wns+Slit10] 中的内容送入 Wnd	1	1	无
	MOV f	将 f 中的内容送入目标寄存器	1	1	N, Z
	MOV f, WREG	将 f 中的内容送入 WREG	1	1	N, Z
	MOV #lit16, Wn	将 16 位立即数送入 Wn	1	1	无
	MOV.b #lit8, Wn	将 8 位立即数送入 Wn	1	1	无
	MOV Wn, f	将 Wn 中的内容送入 f	1	1	无
	MOV Wns, [Wns+Slit10]	将 Wns 中的内容送入 [Wns+Slit10]	1	1	
	MOV Wso, Wdo	将 Ws 中的内容送入 Wd	1	1	无
	MOV WREG, f	将 WREG 中的内容送入 f	1	1	N, Z
MOV.D	MOV.D Wns, Wd	将 W(ns):W(ns+1) 中的双字内容送入 Wd	1	2	无
	MOV.D Ws, Wnd	将 Ws 中的双字内容送入 W(nd+1):W(nd)	1	2	无
MUL	MUL.SS Wb, Ws, Wnd	{Wnd+1, Wnd} = Signed(Wb) * Signed(Ws)	1	1	无
	MUL.SU Wb, Ws, Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(Ws)	1	1	无
	MUL.US Wb, Ws, Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Signed(Ws)	1	1	无
	MUL.UU Wb, Ws, Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(Ws)	1	1	无
	MUL.SU Wb, #lit5, Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(lit5)	1	1	无
	MUL.UU Wb, #lit5, Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(lit5)	1	1	无
	MUL f	W3:W2 = f * WREG	1	1	无
NEG	NEG f	f = -f + 1	1	1	C, DC, N, OV, Z
	NEG f, WREG	WREG = -f + 1	1	1	C, DC, N, OV, Z
	NEG Ws, Wd	Wd = -Ws + 1	1	1	C, DC, N, OV, Z
NOP	NOP	空操作	1	1	无
	NOPR	空操作	1	1	无
POP	POP f	将栈顶 (TOS) 的内容弹出到 f	1	1	无
	POP Wdo	将栈顶 (TOS) 的内容弹出到 Wdo	1	1	无
	POP.D Wnd	将栈顶 (TOS) 的内容弹出到 W(nd):W(nd+1)	1	2	无
	POP.S	将影子寄存器的内容弹出到主寄存器	1	1	全部
PUSH	PUSH f	将 f 的内容压入栈顶 (TOS)	1	1	无
	PUSH Wso	将 Wso 的内容压入栈顶 (TOS)	1	1	无
	PUSH.D Wns	将 W(ns):W(ns+1) 的双字内容压入栈顶 (TOS)	1	2	无
	PUSH.S	将主寄存器中的双字内容压入影子寄存器	1	1	无

PIC24FJ256GB110 系列

表 28-2: 指令集概述 (续)

汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
PWRSAV	PWRSAV #lit1	进入休眠或空闲模式	1	1	WDTO, Sleep
RCALL	RCALL Expr	相对调用	1	2	无
	RCALL Wn	计算调用	1	2	无
REPEAT	REPEAT #lit14	将下一条指令重复执行 lit14 + 1 次	1	1	无
	REPEAT Wn	将下一条指令重复执行 (Wn) + 1 次	1	1	无
RESET	RESET	软件器件复位	1	1	无
RETFIE	RETFIE	从中断返回	1	3 (2)	无
RETLW	RETLW #lit10,Wn	返回并将立即数存入 Wn	1	3 (2)	无
RETURN	RETURN	从子程序返回	1	3 (2)	无
RLC	RLC f	f = 对 f 执行带进位的循环左移	1	1	C, N, Z
	RLC f, WREG	WREG = 对 f 执行带进位的循环左移	1	1	C, N, Z
	RLC Ws, Wd	Wd = 对 Ws 执行带进位的循环左移	1	1	C, N, Z
RLNC	RLNC f	f = 循环左移 f (不带进位)	1	1	N, Z
	RLNC f, WREG	WREG = 循环左移 f (不带进位)	1	1	N, Z
	RLNC Ws, Wd	Wd = 循环左移 Ws (不带进位)	1	1	N, Z
RRC	RRC f	f = 对 f 执行带进位的循环右移	1	1	C, N, Z
	RRC f, WREG	WREG = 对 f 执行带进位的循环右移	1	1	C, N, Z
	RRC Ws, Wd	Wd = 对 Ws 执行带进位的循环右移	1	1	C, N, Z
RRNC	RRNC f	f = 循环右移 f (不带进位)	1	1	N, Z
	RRNC f, WREG	WREG = 循环右移 f (不带进位)	1	1	N, Z
	RRNC Ws, Wd	Wd = 循环右移 Ws (不带进位)	1	1	N, Z
SE	SE Ws, Wnd	Wnd = 符号扩展后的 Ws	1	1	C, N, Z
SETM	SETM f	f = FFFFh	1	1	无
	SETM WREG	WREG = FFFFh	1	1	无
	SETM Ws	Ws = FFFFh	1	1	无
SL	SL f	f = 左移 f	1	1	C, N, OV, Z
	SL f, WREG	WREG = 左移 f	1	1	C, N, OV, Z
	SL Ws, Wd	Wd = 左移 Ws	1	1	C, N, OV, Z
	SL Wb, Wns, Wnd	Wnd = 将 Wb 左移 Wns 位	1	1	N, Z
	SL Wb, #lit5, Wnd	Wnd = 将 Wb 左移 lit5 位	1	1	N, Z
SUB	SUB f	f = f - WREG	1	1	C, DC, N, OV, Z
	SUB f, WREG	WREG = f - WREG	1	1	C, DC, N, OV, Z
	SUB #lit10, Wn	Wn = Wn - lit10	1	1	C, DC, N, OV, Z
	SUB Wb, Ws, Wd	Wd = Wb - Ws	1	1	C, DC, N, OV, Z
	SUB Wb, #lit5, Wd	Wd = Wb - lit5	1	1	C, DC, N, OV, Z
SUBB	SUBB f	f = f - WREG - (\bar{C})	1	1	C, DC, N, OV, Z
	SUBB f, WREG	WREG = f - WREG - (\bar{C})	1	1	C, DC, N, OV, Z
	SUBB #lit10, Wn	Wn = Wn - lit10 - (\bar{C})	1	1	C, DC, N, OV, Z
	SUBB Wb, Ws, Wd	Wd = Wb - Ws - (\bar{C})	1	1	C, DC, N, OV, Z
	SUBB Wb, #lit5, Wd	Wd = Wb - lit5 - (\bar{C})	1	1	C, DC, N, OV, Z
SUBR	SUBR f	f = WREG - f	1	1	C, DC, N, OV, Z
	SUBR f, WREG	WREG = WREG - f	1	1	C, DC, N, OV, Z
	SUBR Wb, Ws, Wd	Wd = Ws - Wb	1	1	C, DC, N, OV, Z
	SUBR Wb, #lit5, Wd	Wd = lit5 - Wb	1	1	C, DC, N, OV, Z
SUBBR	SUBBR f	f = WREG - f - (\bar{C})	1	1	C, DC, N, OV, Z
	SUBBR f, WREG	WREG = WREG - f - (\bar{C})	1	1	C, DC, N, OV, Z
	SUBBR Wb, Ws, Wd	Wd = Ws - Wb - (\bar{C})	1	1	C, DC, N, OV, Z
	SUBBR Wb, #lit5, Wd	Wd = lit5 - Wb - (\bar{C})	1	1	C, DC, N, OV, Z
SWAP	SWAP.b Wn	Wn = 半字节交换 Wn 内容	1	1	无
	SWAP Wn	Wn = 将 Wn 的两个字节相交换	1	1	无

表 28-2: 指令集概述 (续)

汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
TBLRDH	TBLRDH Ws,Wd	将程序存储单元的 <23:16> 读入 Wd<7:0>	1	2	无
TBLRDL	TBLRDL Ws,Wd	将程序存储单元的 <15:0> 读入 Wd	1	2	无
TBLWTH	TBLWTH Ws,Wd	将 Ws<7:0> 写入程序存储单元的 <23:16>	1	2	无
TBLWTL	TBLWTL Ws,Wd	将 Ws 写入程序存储单元的 <15:0>	1	2	无
ULNK	ULNK	释放堆栈帧	1	1	无
XOR	XOR f	f = f .XOR.WREG	1	1	N, Z
	XOR f,WREG	WREG = f .XOR.WREG	1	1	N, Z
	XOR #lit10,Wn	Wd = lit10 .XOR.Wd	1	1	N, Z
	XOR Wb,Ws,Wd	Wd = Wb .XOR.Ws	1	1	N, Z
	XOR Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N, Z
ZE	ZE Ws,Wnd	Wnd = 零扩展后的 Ws	1	1	C, Z, N

PIC24FJ256GB110 系列

注:

29.0 电气特性

本章将对 PIC24FJ256GB110 系列的电气特性进行概括介绍。其余信息将在该文档的后续版本中给出。

下面列出了 PIC24FJ256GB110 系列器件的绝对最大额定值。器件长时间工作在最大额定值条件下可能会影响其可靠性。我们不建议使器件在或超过本规范指定的最大额定值条件下运行。

绝对最大额定值 (t)

环境温度.....	-40°C 至 +100°C
储存温度.....	-65°C 至 +150°C
V _{DD} 引脚相对于 V _{SS} 的电压.....	-0.3V 至 +4.0V
任一模拟 / 数字引脚和 MCLR 引脚相对于 V _{SS} 的电压.....	-0.3V 至 (V _{DD} + 0.3V)
任一只能用作数字的引脚相对于 V _{SS} 的电压.....	-0.3V 至 +6.0V
V _{DDCORE} 引脚相对于 V _{SS} 的电压	-0.3V 至 +3.0V
流出 V _{SS} 引脚的最大电流.....	300 mA
流入 V _{DD} 引脚的最大电流 (注 1)	250 mA
任一 I/O 引脚的最大输出灌电流	25 mA
任一 I/O 引脚的最大输出拉电流	25 mA
所有端口的最大灌电流	200 mA
所有端口的最大拉电流 (注 1)	200 mA

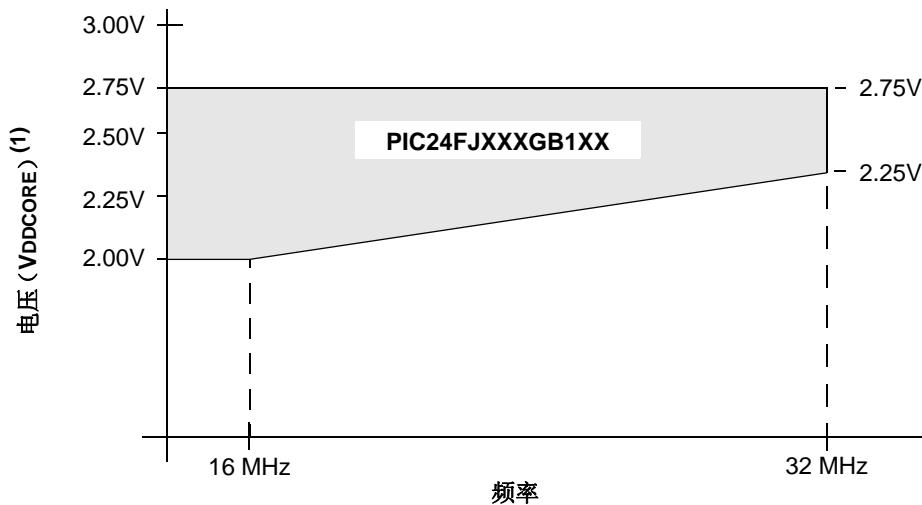
注 1: 允许的最大电流由器件最大功耗决定 (见表 29-1)。

†注: 如果器件工作条件超过上述“绝对最大额定值”，可能引起器件永久性损坏。这仅是极限参数，我们不建议器件工作在极限值甚至超过上述极限值。器件长时间工作在极限值条件下可能会影响其可靠性。

PIC24FJ256GB110 系列

29.1 直流特性

图 29-1: PIC24FJ256GB110 系列电压-频率关系图 (工业级)



对于 16 MHz 到 32 MHz 之间的频率, $F_{MAX} = (64 \text{ MHz/V}) * (V_{DDCORE} - 2\text{V}) + 16 \text{ MHz}$ 。

注 1: 当禁止稳压器时, VDD 和 VDDCORE 必须维持在以下条件: $V_{DDCORE} \leq VDD \leq 3.6\text{V}$ 。

表 29-1: 热工作条件

参数	符号	最小值	典型值	最大值	单位
PIC24FJ256GB110 系列:					
工作结温范围	T _J	-40	—	+125	°C
工作环境温度范围	T _A	-40	—	+85	°C
功耗: 芯片内部功耗: $P_{INT} = V_{DD} \times (I_{DD} - \sum I_{OH})$	P _D		$P_{INT} + P_{I/O}$		W
I/O 引脚功耗: $P_{I/O} = \sum (V_{DD} - V_{OH}) \times I_{OH} + \sum (V_{OL} \times I_{OL})$					
最大允许功耗	P _{DMAX}		$(T_J - T_A)/\theta_{JA}$		W

表 29-2: 热封装特性

特性	符号	典型值	最大值	单位	注
封装热阻, 14x14x1 mm TQFP	θ _{JA}	50.0	—	°C/W	(注 1)
封装热阻, 12x12x1 mm TQFP	θ _{JA}	69.4	—	°C/W	(注 1)
封装热阻, 10x10x1 mm TQFP	θ _{JA}	76.6	—	°C/W	(注 1)
封装热阻, 9x9x0.9 mm QFN	θ _{JA}	28.0	—	°C/W	(注 1)

注 1: 通过封装模拟获得结点与环境的热阻值 θ_{JA}。

表 29-3: 直流特性: 温度和电压规范

直流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数 编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
工作电压							
DC10	供电电压						
	VDD		2.2	—	3.6	V	稳压器使能
	VDD		VDDCORE	—	3.6	V	稳压器禁止
	VDDCORE		2.0	—	2.75	V	稳压器禁止
DC12	VDR	RAM 数据保持电压⁽²⁾	1.5	—	—	V	
DC16	VPOR	VDD 启动电压 (确保内部上电复位信号)	Vss	—	—	V	
DC17	SVDD	VDD 上升速率 (确保内部上电复位信号)	0.05	—	—	V/ms	0-3.3V/0.1s 0-2.5V/60 ms
DC18	VBOR	VDD 变化时的 BOR 电压 变化方向从高到低	—	2.05	—	V	使能稳压器

注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

2: 这是在不丢失 RAM 数据的前提下, VDD 的下限值。

PIC24FJ256GB110 系列

表 29-4: 直流特性: 工作电流 (I_{DD})

直流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)			
参数编号	典型值 ⁽¹⁾	最大值	单位	条件		
工作电流 (I_{DD})⁽²⁾						
DC20	0.83	1.2	mA	-40°C	2.0V ⁽³⁾	1 MIPS
DC20a	0.83	1.2	mA	+25°C		
DC20b	0.83	1.2	mA	+85°C		
DC20d	1.1	1.7	mA	-40°C		
DC20e	1.1	1.7	mA	+25°C		
DC20f	1.1	1.7	mA	+85°C		
DC23	3.3	4.5	mA	-40°C	2.0V ⁽³⁾	4 MIPS
DC23a	3.3	4.5	mA	+25°C		
DC23b	3.3	4.5	mA	+85°C		
DC23d	4.3	6	mA	-40°C		
DC23e	4.3	6	mA	+25°C		
DC23f	4.3	6	mA	+85°C		
DC24	18.2	24	mA	-40°C	2.5V ⁽³⁾	16 MIPS
DC24a	18.2	24	mA	+25°C		
DC24b	18.2	24	mA	+85°C		
DC24d	18.2	24	mA	-40°C		
DC24e	18.2	24	mA	+25°C		
DC24f	18.2	24	mA	+85°C		
DC31	15.0	54	μA	-40°C	2.0V ⁽³⁾	LPRC (31 kHz)
DC31a	15.0	54	μA	+25°C		
DC31b	20.0	69	μA	+85°C		
DC31d	57.0	96	μA	-40°C		
DC31e	57.0	96	μA	+25°C		
DC31f	95.0	145	μA	+85°C		

注 1: 除非另外声明, 否则 “典型值” 栏中的数据均为 3.3V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

2: 供电电流主要受工作电压和频率的影响。其他因素如 I/O 引脚负载和开关速率、振荡器类型、内部代码执行模式以及温度也对电流消耗有影响。所有 I_{DD} 测量的测试条件为: OSCI 使用满幅的外部方波进行驱动。所有 I/O 引脚配置为输入且被拉到 V_{DD}。MCLR = V_{DD}; WDT 和 FSCM 被禁止。CPU、SRAM、程序存储器和数据存储器处于工作状态。外设模块均不工作, 所有外设模块禁止 (PMD) 位被置 1。

3: 禁止片上稳压器 (ENVREG 连接到 V_{SS})。

4: 使能片上稳压器 (ENVREG 连接到 V_{DD})。使能低压检测 (LVD) 和欠压检测 (BOD)。

表 29-5: 直流特性: 空闲电流 (I_{IDLE})

直流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)			
参数编号	典型值 ⁽¹⁾	最大值	单位	条件		
空闲电流 (I_{IDLE})⁽²⁾						
DC40	220	310	μA	-40°C	2.0V ⁽³⁾	1 MIPS
DC40a	220	310	μA	+25°C		
DC40b	220	310	μA	+85°C		
DC40d	300	390	μA	-40°C		
DC40e	300	390	μA	+25°C		
DC40f	300	420	μA	+85°C		
DC43	0.85	1.1	mA	-40°C	2.0V ⁽³⁾	4 MIPS
DC43a	0.85	1.1	mA	+25°C		
DC43b	0.87	1.2	mA	+85°C		
DC43d	1.1	1.4	mA	-40°C		
DC43e	1.1	1.4	mA	+25°C	3.3V ⁽⁴⁾	16 MIPS
DC43f	1.1	1.4	mA	+85°C		
DC47	4.4	5.6	mA	-40°C		
DC47a	4.4	5.6	mA	+25°C		
DC47b	4.4	5.6	mA	+85°C	2.5V ⁽³⁾	FRC (4 MIPS)
DC47c	4.4	5.6	mA	-40°C		
DC47d	4.4	5.6	mA	+25°C		
DC47e	4.4	5.6	mA	+85°C		
DC50	1.1	1.4	mA	-40°C	2.0V ⁽³⁾	FRC (4 MIPS)
DC50a	1.1	1.4	mA	+25°C		
DC50b	1.1	1.4	mA	+85°C		
DC50d	1.4	1.8	mA	-40°C		
DC50e	1.4	1.8	mA	+25°C	3.3V ⁽⁴⁾	LPRC (31 kHz)
DC50f	1.4	1.8	mA	+85°C		
DC51	4.3	13	μA	-40°C		
DC51a	4.5	13	μA	+25°C		
DC51b	10	32	μA	+85°C	2.0V ⁽³⁾	LPRC (31 kHz)
DC51d	44	77	μA	-40°C		
DC51e	44	77	μA	+25°C		
DC51f	70	132	μA	+85°C		

注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

2: 基本 I_{IDLE} 电流的测量是在内核不工作、OSCI 受外部轨至轨方波驱动、所有 I/O 引脚配置为输入引脚并上拉至 V_{DD}、MCLR = V_{DD}、WDT 和 FSCM 被禁止, 且所有外设模块均不工作以及所有外设模块禁止 (PMD) 位被置 1 的条件下进行的。

3: 禁止片上稳压器 (ENVREG 连接到 V_{SS})。

4: 使能片上稳压器 (ENVREG 连接到 V_{DD})。使能低压检测 (LVD) 和欠压检测 (BOD)。

PIC24FJ256GB110 系列

表 29-6： 直流特性：掉电电流（IPD）

直流特性			标准工作条件：2.0V 至 3.6V（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）		
参数编号	典型值 ⁽¹⁾	最大值	单位	条件	
掉电电流（IPD）⁽²⁾					
DC60	0.1	1	μA	-40°C	2.0V ⁽³⁾
DC60a	0.15	1	μA	+25°C	
DC60m	2.25	11	μA	+60°C	
DC60b	3.7	18	μA	+85°C	
DC60c	0.2	1.4	μA	-40°C	2.5V ⁽³⁾
DC60d	0.25	1.4	μA	+25°C	
DC60n	2.6	16.5	μA	+60°C	
DC60e	4.2	27	μA	+85°C	
DC60f	3.6	10	μA	-40°C	3.3V ⁽⁴⁾
DC60g	4.0	10	μA	+25°C	
DC60p	8.1	25.2	μA	+60°C	
DC60h	11.0	36	μA	+85°C	
DC61	1.75	3	μA	-40°C	2.0V ⁽³⁾
DC61a	1.75	3	μA	+25°C	
DC61m	1.75	3	μA	+60°C	
DC61b	1.75	3	μA	+85°C	
DC61c	2.4	4	μA	-40°C	2.5V ⁽³⁾
DC61d	2.4	4	μA	+25°C	
DC61n	2.4	4	μA	+60°C	
DC61e	2.4	4	μA	+85°C	
DC61f	2.8	5	μA	-40°C	3.3V ⁽⁴⁾
DC61g	2.8	5	μA	+25°C	
DC61p	2.8	5	μA	+60°C	
DC61b	2.8	5	μA	+85°C	

注 1：除非另外声明，否则“典型值”栏中的数据均为 3.3V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

2：基本 IPD 是在所有外设和时钟都关闭的条件下进行测量的。所有 I/O 都配置为输入且被拉至高电平。WDT 等外设也都被关闭，PMSLP 位被清零，所有未用外设的外设模块禁止（PMD）位被置 1。

3：禁止片上稳压器（ENVREG 连接到 VSS）。

4：使能片上稳压器（ENVREG 连接到 VDD）。使能低压检测（LVD）和欠压检测（BOD）。

5：Δ 电流为当模块使能时额外消耗的电流。此电流应被加到基本 IPD 电流。

表 29-6: 直流特性: 掉电电流 (IPD) (续)

直流特性		标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)			
参数编号	典型值 ⁽¹⁾	最大值	单位	条件	
掉电电流 (IPD) ⁽²⁾					
DC62	2.5	7	μA	-40°C	2.0V ⁽³⁾
DC62a	2.5	7	μA	+25°C	
DC62m	3.0	7	μA	+60°C	
DC62b	3.0	7	μA	+85°C	
DC62c	2.8	7	μA	-40°C	
DC62d	3.0	7	μA	+25°C	
DC62n	3.0	7	μA	+60°C	
DC62e	3.0	7	μA	+85°C	
DC62f	3.5	10	μA	-40°C	2.5V ⁽³⁾
DC62g	3.5	10	μA	+25°C	
DC62p	4.0	10	μA	+60°C	
DC62h	4.0	10	μA	+85°C	
RTCC + Timer1, 带 32 kHz 晶振: ΔRTCC Δ T 32⁽⁵⁾					

注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

2: 基本 IPD 是在所有外设和时钟都关闭的条件下进行测量的。所有 I/O 都配置为输入且被拉至高电平。WDT 等外设也都被关闭, PMSLP 位被清零, 所有未用外设的外设模块禁止 (PMD) 位被置 1。

3: 禁止片上稳压器 (ENVREG 连接到 VSS)。

4: 使能片上稳压器 (ENVREG 连接到 VDD)。使能低压检测 (LVD) 和欠压检测 (BOD)。

5: Δ 电流为当模块使能时额外消耗的电流。此电流应被加到基本 IPD 电流。

PIC24FJ256GB110 系列

表 29-7： 直流特性：I/O 引脚输入规范

直流特性			标准工作条件：2.0V 至 3.6V（除非另外声明） 工作温度 -40°C ≤ TA ≤ +85°C（工业级）				
参数 编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
DI10 DI11 DI15 DI16 DI17 DI18 DI19	VIL	输入低电压 ⁽⁴⁾ 带 ST 缓冲器的 I/O 引脚 带 TTL 缓冲器的 I/O 引脚 <u>MCLR</u>	Vss	—	0.2 VDD	V	
		OSC1 (XT 模式)	Vss	—	0.2 VDD	V	
		OSC1 (HS 模式)	Vss	—	0.2 VDD	V	
		带 I ² C TM 缓冲器的 I/O 引脚	Vss	—	0.3 VDD	V	
		带 SMBus 缓冲器的 I/O 引脚	Vss	—	0.8	V	使能 SMBus
		输入高电压 ⁽⁴⁾ 带 ST 缓冲器的 I/O 引脚： 带模拟功能， 仅数字功能	0.8 VDD 0.8 VDD	— —	VDD 5.5	V V	
		带 TTL 缓冲器的 I/O 引脚： 带模拟功能， 仅数字功能	0.25 VDD + 0.8 0.25 VDD + 0.8	— —	VDD 5.5	V V	
DI20 DI21 DI25 DI26 DI27 DI28 DI29	VIH	<u>MCLR</u>	0.8 VDD	—	VDD	V	
		OSC1 (XT 模式)	0.7 VDD	—	VDD	V	
		OSC1 (HS 模式)	0.7 VDD	—	VDD	V	
		带 I ² C 缓冲器的 I/O 引脚： 带模拟功能， 仅数字功能	0.7 VDD 0.7 VDD	— —	VDD 5.5	V V	
		带 SMBus 缓冲器的 I/O 引脚： 带模拟功能， 仅数字功能	2.1 2.1		VDD 5.5	V V	2.5V ≤ VPIN ≤ VDD
		CNxx 上拉电流	50	250	400	μA	VDD = 3.3V, VPIN = VSS
		CNxx 下拉电流	—	80	—	μA	VDD = 3.3V, VPIN = VDD
DI50 DI51 DI52 DI55 DI56	IIL	输入泄漏电流 ^(2,3) I/O 端口	—	—	±1	μA	VSS ≤ VPIN ≤ VDD, 引脚处于高阻态
		模拟输入引脚	—	—	±1	μA	VSS ≤ VPIN ≤ VDD, 引脚处于高阻态
		USB 差分引脚 (D+, D-)	—	—	±1	μA	V _{USB} ≥ VDD
		<u>MCLR</u>	—	—	±1	μA	VSS ≤ VPIN ≤ VDD
		OSC1	—	—	±1	μA	VSS ≤ VPIN ≤ VDD, XT 和 HS 模式

注 1：除非另外声明，否则“典型值”栏中的数据均为 3.3V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

2：MCLR 引脚上的泄漏电流主要取决于所施加电压。规定电压为正常工作条件下的电压。在不同的输入电压下可能测得更高的泄漏电流。

3：负电流定义为引脚的拉电流。

4：请参见表 1-4 了解 I/O 引脚缓冲器类型。

表 29-8: 直流特性: I/O 引脚输出规范

直流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
DO10 DO16	VOL	输出低电压 I/O 端口	—	—	0.4	V	IOL = 8.5 mA, VDD = 3.6V
			—	—	0.4	V	IOL = 6.0 mA, VDD = 2.0V
		OSC2/CLKO	—	—	0.4	V	IOL = 8.5 mA, VDD = 3.6V
			—	—	0.4	V	IOL = 6.0 mA, VDD = 2.0V
DO20 DO26	VOH	输出高电压 I/O 端口	3.0	—	—	V	IOH = -3.0 mA, VDD = 3.6V
			2.4	—	—	V	IOH = -6.0 mA, VDD = 3.6V
			1.65	—	—	V	IOH = -1.0 mA, VDD = 2.0V
			1.4	—	—	V	IOH = -3.0 mA, VDD = 2.0V
		OSC2/CLKO	2.4	—	—	V	IOH = -6.0 mA, VDD = 3.6V
			1.4	—	—	V	IOH = -3.0 mA, VDD = 2.0V

注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

表 29-9: 直流特性: 程序存储器

直流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
D130	EP	单元耐擦写能力	10000	—	—	E/W	-40°C 至 +85°C
D131	VPR	读操作时的 VDD	VMIN	—	3.6	V	VMIN = 最小工作电压
D132A D132B	VPEW	用于自定时写的供电电压 VDDCORE VDD	2.25	—	3.6	V	
			2.35	—	3.6	V	
			—	3	—	ms	
D133B	TiE	自定时页擦写时间	40	—	—	ms	
D134	TRETD	特性保持时间	20	—	—	年	假设没有违反其他规范
D135	IDDP	编程时的供电电流	—	7	—	mA	

注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V 和 25°C 条件下的值。

PIC24FJ256GB110 系列

表 29-10： 内部稳压器规范

工作条件: $-40^{\circ}\text{C} < \text{T}_A < +125^{\circ}\text{C}$ (除非另外声明)

参数 编号	符号	特性	最小值	典型值	最大值	单位	备注
	VRGOUT	稳压器输出电压	—	2.5	—	V	
	VBG	内部带隙参考源	—	1.2	—	V	
	C _{EFC}	外部滤波器电容值	4.7	10	—	μF	建议等效串联阻抗 $< 3\Omega$; 必须 $< 5\Omega$ 。
	TVREG	稳压器启动时间	—	10	—	μs	PMSLP = 1, 或者任何 POR 或 BOR
			—	190	—	μs	当 PMSLP = 0 时, 从休眠 唤醒
	T _{BG}	带隙参考源启动时间	—	—	1	ms	

29.2 交流特性和时序参数

本节包含的信息说明了 PIC24FJ256GB110 系列器件的交流特性和时序参数。

表 29-11: 温度和电压规范——交流

交流特性	标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 $-40^{\circ}\text{C} \leq \text{TA} \leq +85^{\circ}\text{C}$ (工业级) 工作电压 VDD 范围如第 29.1 节 “直流特性” 中所述。
------	--

图 29-2: 器件时序规范的负载条件

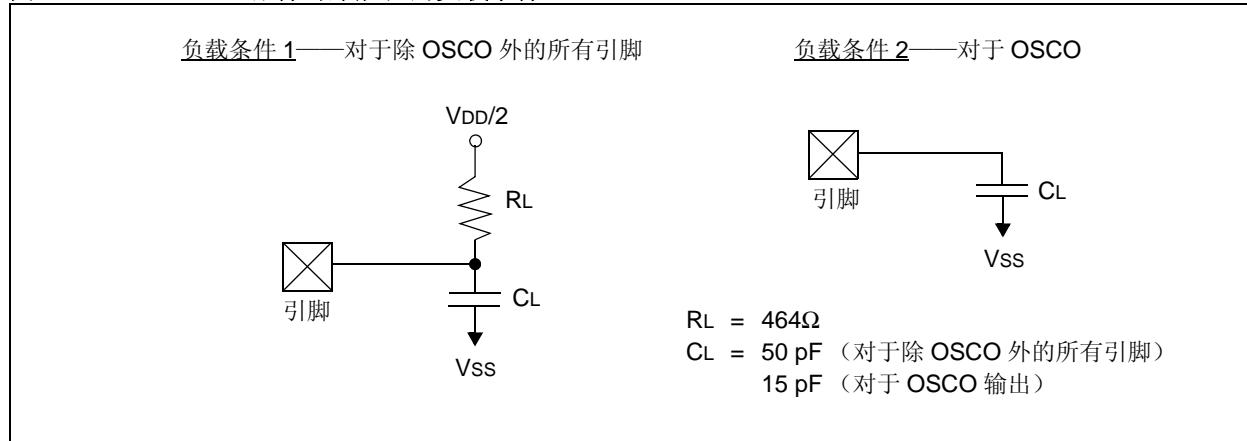


表 29-12: 输出引脚上的容性负载要求

参数编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
DO50	Cosc2	OSCO/CLKO 引脚	—	—	15	pF	当外部时钟用于驱动 OSCI 时处于 XT 和 HS 模式下。
DO56	Cio	所有 I/O 引脚和 OSCO	—	—	50	pF	EC 模式。
DO58	Cb	SCLx 和 SDAX	—	—	400	pF	在 I ² C TM 模式下。

注 1: 除非另外声明, 否则 “典型值” 栏中的数据均为 3.3V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

PIC24FJ256GB110 系列

图 29-3: 外部时钟时序

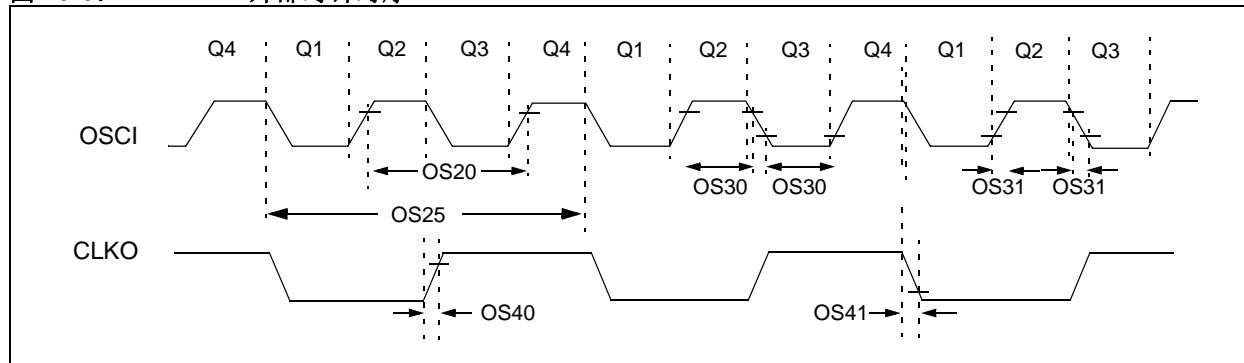


表 29-13: 外部时钟时序要求

交流特性			标准工作条件: 2.50 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
OS10	Fosc	外部 CLKI 频率 (外部时钟仅允许运行于 EC 模式)	DC 4	—	32 48	MHz MHz	EC ECPLL
		振荡器频率	3 4 10 12 31	—	10 8 32 32 33	MHz MHz MHz MHz kHz	XT XTPLL HS HSPLL SOSC
OS20	Tosc	$T_{osc} = 1/F_{osc}$	—	—	—	—	Fosc 值见参数 OS10
OS25	Tcy	指令周期 ⁽²⁾	62.5	—	DC	ns	
OS30	TosL, TosH	外部时钟输入 (OSCI) 的高电平或低电平时间	0.45 x Tosc	—	—	ns	EC
OS31	TosR, TosF	外部时钟输入 (OSCI) 的上升或下降时间	—	—	20	ns	EC
OS40	TckR	CLKO 上升时间 ⁽³⁾	—	6	10	ns	
OS41	TckF	CLKO 下降时间 ⁽³⁾	—	6	10	ns	

注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

2: 指令周期 (Tcy) 等于输入振荡器时基周期的两倍。所有规定值均为基于针对特定振荡器类型, 器件在标准工作条件下执行代码时的特性数据。超出这些规定的限定值, 可能导致振荡器运行不稳定和 / 或导致电流消耗超出预期值。所有器件在测试“最小”值时, 都在 OSCI/CLKI 引脚连接了外部时钟。当使用了外部时钟输入时, 所有器件的“最大”周期时间限制为“DC”(无时钟)。

3: 测量在 EC 模式下进行。在 OSCO 引脚上测量 CLKO 信号。CLKO 在 Q1-Q2 周期 (1/2 Tcy) 中为低电平, 在 Q3-Q4 周期 (1/2 Tcy) 中为高电平。

表 29-14: PLL 时钟时序规范 ($VDD = 2.0V$ 至 $3.6V$)

交流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	符号	特性 ⁽¹⁾	最小值	典型值 ⁽²⁾	最大值	单位	条件
OS50	FPLL1	PLL 输入频率范围 ⁽²⁾	4	—	32	MHz	ECPLL、HSPLL 和 XTPLL 模式
OS51	FSYS	PLL 输出频率范围	95.76	—	96.24	MHz	
OS52	TLOCK	PLL 起振时间 (锁定时间)	—	—	200	μs	
OS53	DCLK	CLKO 稳定性 (抗抖动性)	-0.25	—	0.25	%	

注 1: 这些参数为特性值, 但生产时未经测试。

2: 除非另外声明, 否则 “典型值” 栏中的数据均为 3.3V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

表 29-15: 内部 RC 振荡器规范

交流特性		标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)					
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
	TFRC	FRC 起振时间	—	15	—	μs	
	TLPRC	LPRC 起振时间	—	40	—	μs	

表 29-16: 内部 RC 振荡器精度

交流特性		标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	特性	最小值	典型值	最大值	单位	条件
F20	FRC 精度 @ 8 MHz ⁽¹⁾	-2	—	2	%	+25°C, 3.0V ≤ VDD ≤ 3.6V
		-5	—	5	%	-40°C ≤ TA ≤ +85°C, 3.0V ≤ VDD ≤ 3.6V
F21	LPRC 精度 @ 31 kHz ⁽²⁾	-20	—	20	%	-40°C ≤ TA ≤ +85°C, 3.0V ≤ VDD ≤ 3.6V

注 1: 频率在 25°C 和 3.3V 条件下校准。OSCTUN 位可用来补偿温度漂移。

2: LPRC 频率将随 VDD 的变化而变化。

PIC24FJ256GB110 系列

图 29-4: CLKO 和 I/O 时序特性

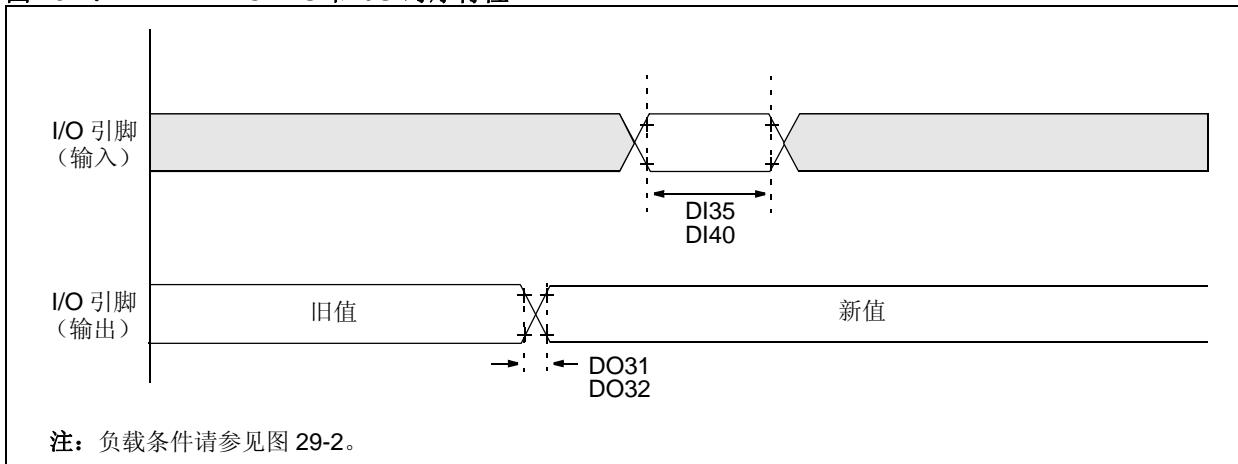


表 29-17: CLKO 和 I/O 时序要求

交流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
DO31	T _{IoR}	端口输出上升时间	—	10	25	ns	
DO32	T _{IoF}	端口输出下降时间	—	10	25	ns	
DI35	T _{InP}	INTx 引脚高电平或低电平时间 (输出)	20	—	—	ns	
DI40	T _{RPB}	CNx 高电平或低电平时间 (输入)	2	—	—	TCY	

注 1: 除非另外声明, 否则 “典型值” 栏中的数据均为 3.3V 和 25°C 条件下的值。

表 29-18: ADC 模块规范

交流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C				
参数 编号	符号	特性	最小值	典型值	最大值	单位	条件
器件电源							
AD01	AVDD	模块电源 VDD	VDD - 0.3 和 2.0 中的较大值	—	VDD + 0.3 和 3.6 中的较小值	V	
AD02	AVSS	模块电源 VSS	VSS - 0.3	—	VSS + 0.3	V	
参考输入							
AD05	VREFH	参考电压高电平	AVSS + 1.7	—	AVDD	V	
AD06	VREFL	参考电压低电平	AVSS	—	AVDD - 1.7	V	
AD07	VREF	绝对参考电压	AVSS - 0.3	—	AVDD + 0.3	V	
模拟输入							
AD10	VINH-VINL	满量程输入范围	VREFL	—	VREFH	V	(注 2)
AD11	VIN	绝对输入电压	AVSS - 0.3	—	AVDD + 0.3	V	
AD12	VINL	绝对 VINL 输入电压	AVSS - 0.3	—	AVDD/2	V	
AD13	—	泄漏电流	—	±0.001	±0.610	μA	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V, 信号源阻抗 = 2.5 kΩ
AD17	RIN	模拟信号源的推荐阻抗	—	—	2.5K	Ω	10 位
ADC 精度							
AD20b	Nr	分辨率	—	10	—	位	
AD21b	INL	积分非线性误差	—	±1	<±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22b	DNL	微分非线性误差	—	±0.5	<±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23b	GERR	增益误差	—	±1	±3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD24b	Eoff	失调误差	—	±1	±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD25b	—	单调性 ⁽¹⁾	—	—	—	—	保证

注 1: ADC 转换结果不会因输入电压的增加而减小，并且不会丢失编码。

2: 测量采用外部 VREF+ 和 VREF- 用作 ADC 参考电压。

PIC24FJ256GB110 系列

表 29-19: ADC 转换时序要求⁽¹⁾

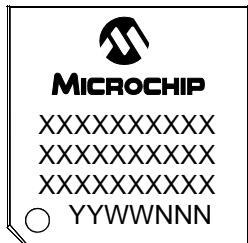
交流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C				
参数 编号	符号	特性	最小值	典型值	最大值	单位	条件
时钟参数							
AD50	TAD	ADC 时钟周期	75	—	—	ns	TCY = 75 ns, AD1CON3 处于默认状态
AD51	tRC	ADC 内部 RC 振荡器周期	—	250	—	ns	
转换速率							
AD55	tCONV	转换时间	—	12	—	TAD	
AD56	FCONV	吞吐率	—	—	500	ksp/s	AVDD > 2.7V
AD57	tsAMP	采样时间	—	1	—	TAD	
时钟参数							
AD61	tPSS	从采样位 (SAMP) 置 1 到采样启动的延时	2	—	3	TAD	

注 1: 因为采样电容最终将释放电荷, 因此低于 10 kHz 的时钟频率可能影响线性性能, 尤其是在温度较高时。

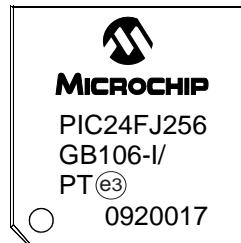
30.0 封装信息

30.1 封装标识信息

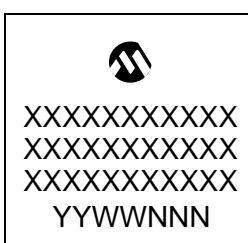
64 引脚 TQFP (10x10x1 mm)



示例



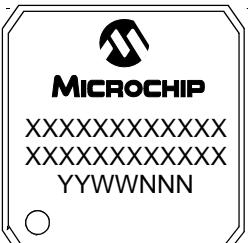
64 引脚 QFN (9x9x0.9 mm)



示例



80 引脚 TQFP (12x12x1 mm)



示例

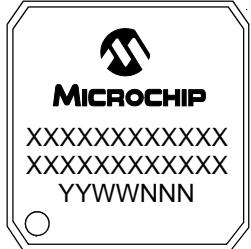


图注:	XX...X	客户指定信息
	Y	年份代码（日历年的最后一位数字）
	YY	年份代码（日历年的最后两位数字）
	WW	星期代码（一月一日的星期代码为“01”）
	NNN	以字母数字排序的追踪代码
	(e3)	雾锡（Matte Tin, Sn）的 JEDEC 无铅标志
*		本封装为无铅封装。JEDEC 无铅标志 (e3) 标示于此种封装的外包装上。

注: Microchip 元器件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户信息的字符数。

PIC24FJ256GB110 系列

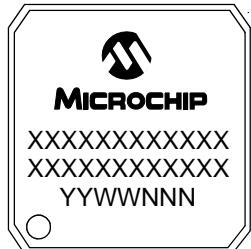
100 引脚 TQFP (12x12x1 mm)



示例



100 引脚 TQFP (14x14x1 mm)



示例



图注:	XX...X	客户指定信息
	Y	年份代码（日历年的最后一位数字）
	YY	年份代码（日历年的最后两位数字）
	WW	星期代码（一月一日的星期代码为“01”）
	NNN	以字母数字排序的追踪代码
		雾锡（Matte Tin, Sn）的 JEDEC 无铅标志
*		本封装为无铅封装。JEDEC 无铅标志（e3）标示于此种封装的外包装上。

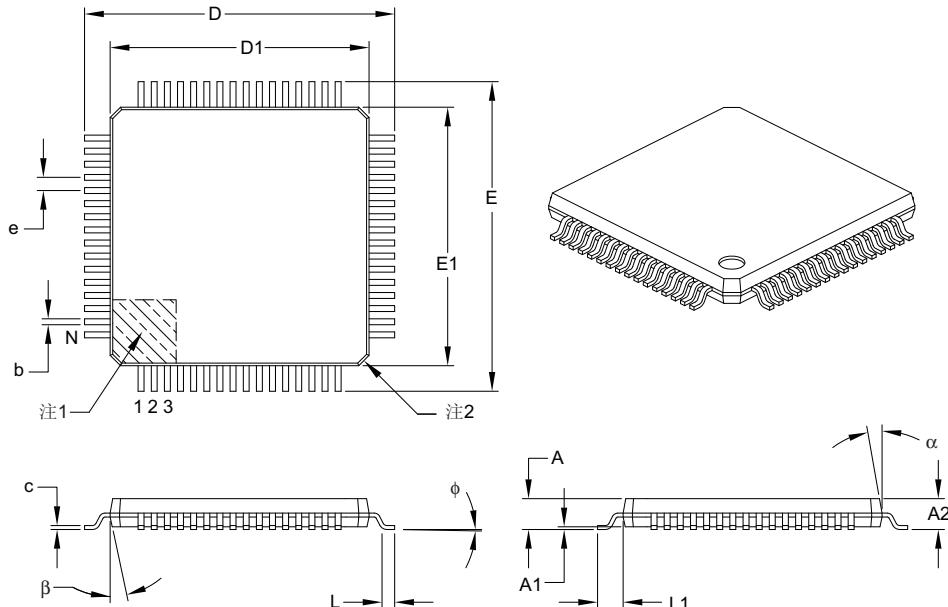
注: Microchip 元器件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户信息的字符数。

30.2 封装详细信息

以下部分将介绍各种封装的技术细节。

64引脚塑封薄型正方扁平封装 (PT) —— 主体10x10x1 mm, 2.00 mm [TQFP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



	单位	毫米		
	尺寸范围	最小	正常	最大
引脚数	N	64		
引脚间距	e	0.50 BSC		
总高度	A	—	—	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	—	0.15
底脚长度	L	0.45	0.60	0.75
引脚投影长度	L1	1.00 REF		
底脚倾斜角	phi	0°	3.5°	7°
总宽度	E	12.00 BSC		
总长度	D	12.00 BSC		
塑模封装宽度	E1	10.00 BSC		
塑模封装长度	D1	10.00 BSC		
引脚厚度	c	0.09	—	0.20
引脚宽度	b	0.17	0.22	0.27
塑模顶部锥度	alpha	11°	12°	13°
塑模底部锥度	beta	11°	12°	13°

注：

1. 引脚1的可见定位标记可能不同，但必须在阴影区域内。
2. 切角处的斜面是可选的；大小可能有所不同。
3. 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0.25毫米。
4. 尺寸和公差遵循ASME Y14.5M。

BSC：基本尺寸。显示的是没有公差的理论精确值。

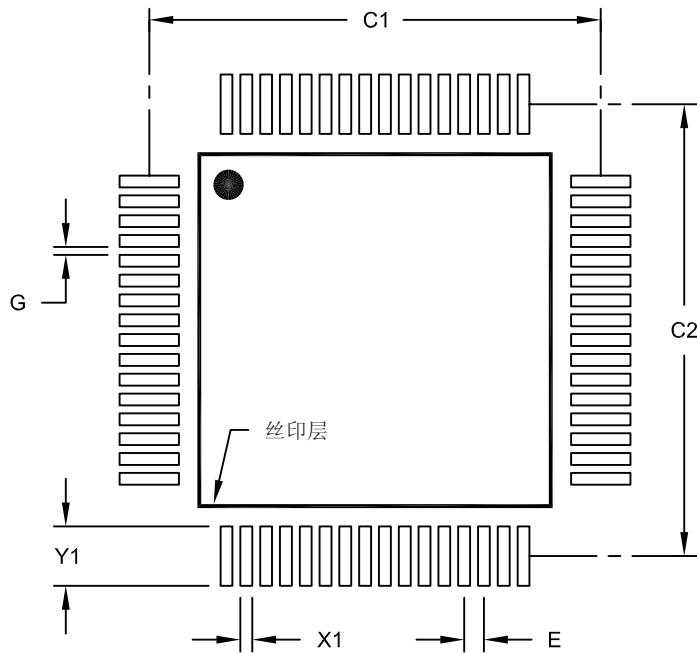
REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology图号C04-085B

PIC24FJ256GB110 系列

64引脚塑封薄型正方扁平封装 (PT) —— 主体10x10x1 mm, 2.00 mm [TQFP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



建议的焊盘布局

单位		毫米		
尺寸范围		最小	正常	最大
触点间距	E	0.50 BSC		
接触焊垫间距	C1		11.40	
接触焊垫间距	C2		11.40	
接触焊垫宽度 (X64)	X1			0.30
接触焊垫长度 (X64)	Y1			1.50
焊垫之间的距离	G	0.20		

注：

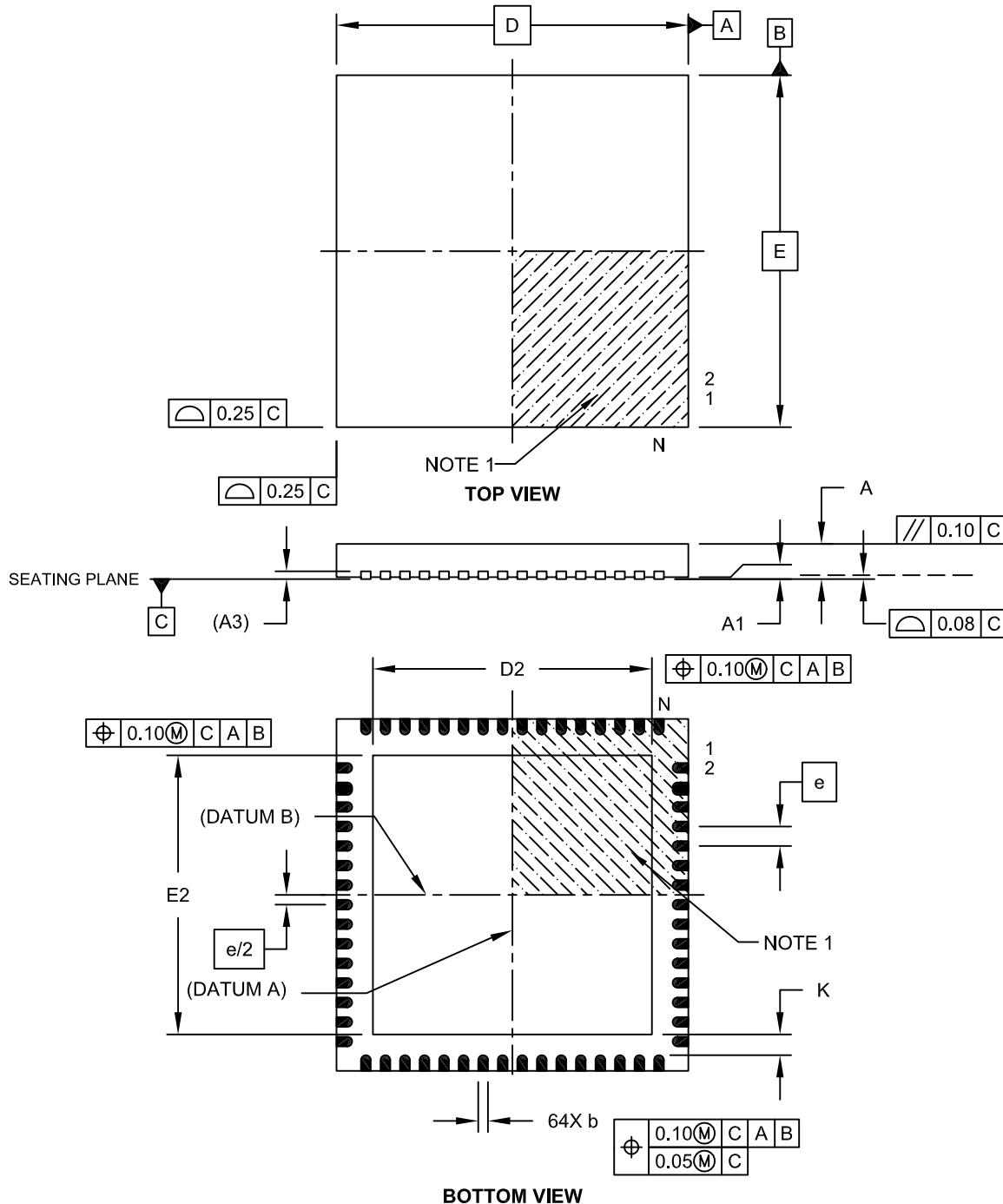
1. 尺寸和公差遵循ASME Y14.5M

BSC：基本尺寸。显示的是没有公差的理论精确值。

Microchip Technology图号C04-2085A

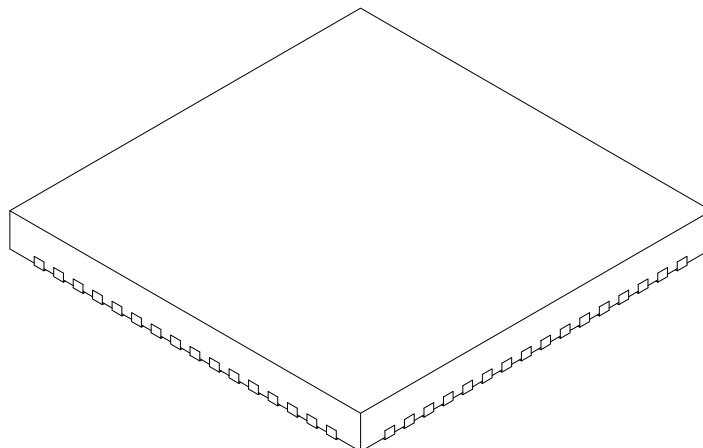
64 引脚塑封正方扁平无脚封装 (MR) —— 主体 9x9x0.9 mm [QFN]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



64 引脚塑封正方扁平无脚封装 (MR) —— 主体 9x9x0.9 mm [QFN]

注 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	64		
Pitch	e	0.50	BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20	REF	
Overall Width	E	9.00	BSC	
Exposed Pad Width	E2	7.05	7.15	7.50
Overall Length	D	9.00	BSC	
Exposed Pad Length	D2	7.05	7.15	7.50
Contact Width	b	0.18	0.25	0.30
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. Package is saw singulated.

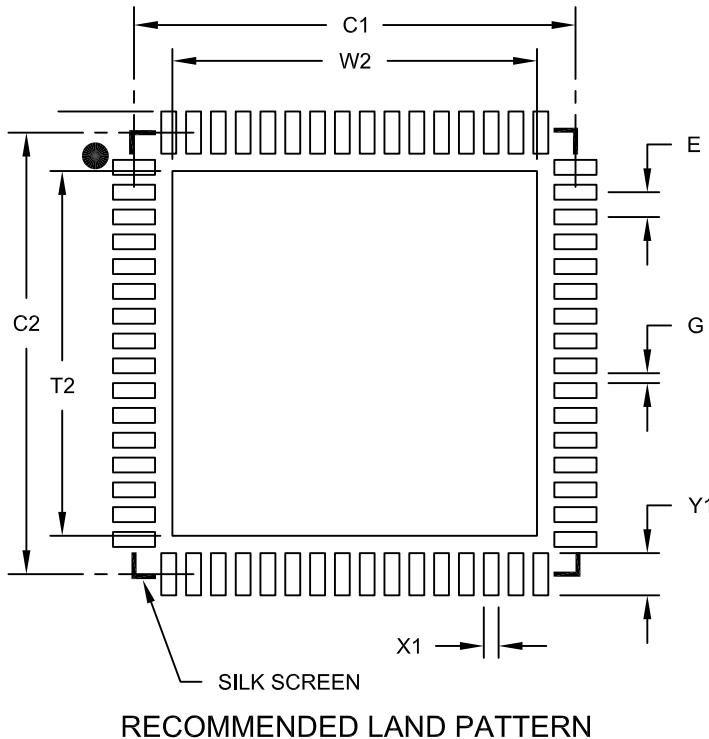
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

64 引脚塑封正方扁平无脚封装 (MR) —— 主体 9x9x0.9 mm [QFN]

注 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50	BSC
Optional Center Pad Width	W2			7.35
Optional Center Pad Length	T2			7.35
Contact Pad Spacing	C1		8.90	
Contact Pad Spacing	C2		8.90	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			0.85
Distance Between Pads	G	0.20		

Notes:

- Dimensioning and tolerancing per ASME Y14.5M

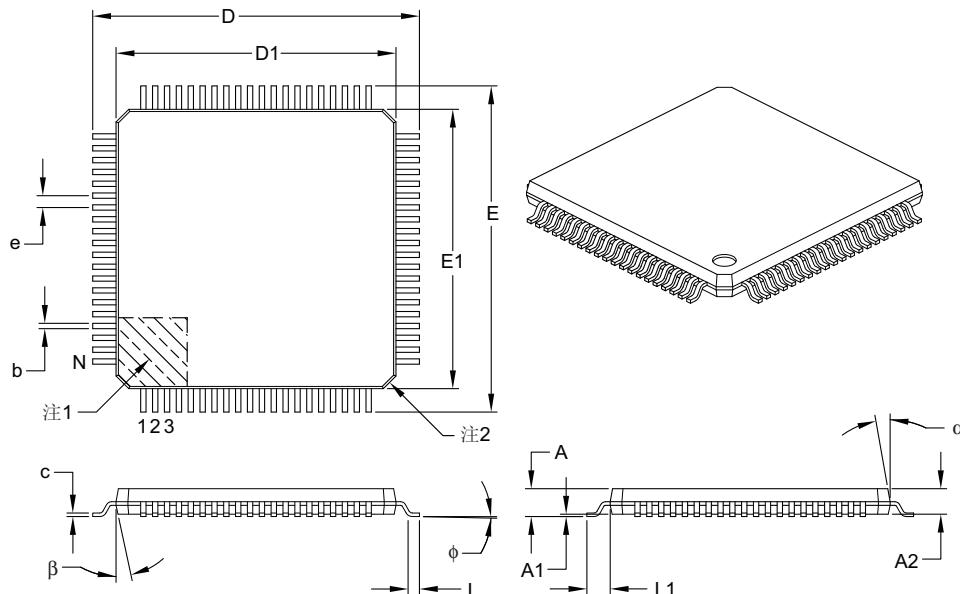
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2149A

PIC24FJ256GB110 系列

80引脚塑封薄型正方扁平封装 (PT) —— 主体12x12x1 mm, 2.00 mm [TQFP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
引脚数		N		
引脚间距		e		
总高度		A		
塑模封装厚度		A2		
悬空间隙		A1		
底脚长度		L		
引脚投影长度		L1		
底脚倾斜角		phi		
总宽度		E		
总长度		D		
塑模封装宽度		E1		
塑模封装长度		D1		
引脚厚度		c		
引脚宽度		b		
塑模顶部锥度		alpha		
塑模底部锥度		beta		

注：

1. 引脚1的可见定位标记可能不同，但必须在阴影区域内。
2. 切角处的斜面是可选的；大小可能有所不同。
3. 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0.25毫米。
4. 尺寸和公差遵循ASME Y14.5M。

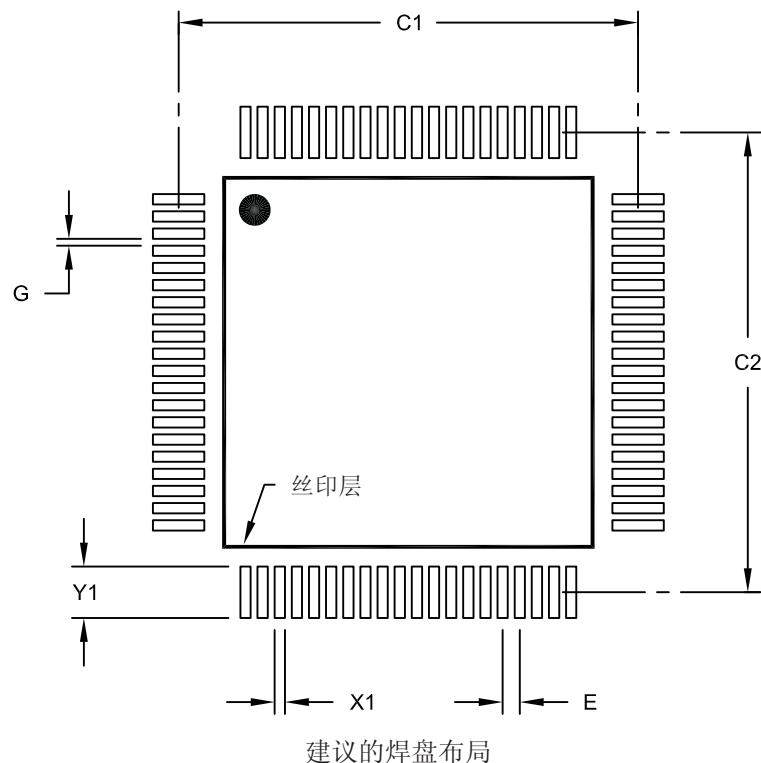
BSC：基本尺寸。显示的是没有公差的理论精确值。

REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology图号C04-092B

80引脚塑封薄型正方扁平封装 (PT) —— 主体12x12x1 mm, 2.00 mm [TQFP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
触点间距	E		0.50	BSC
接触焊垫间距	C1		13.40	
接触焊垫间距	C2		13.40	
接触焊垫宽度 (X80)	X1			0.30
接触焊垫长度 (X80)	Y1			1.50
焊垫之间的距离	G	0.20		

注：

1. 尺寸和公差遵循ASME Y14.5M

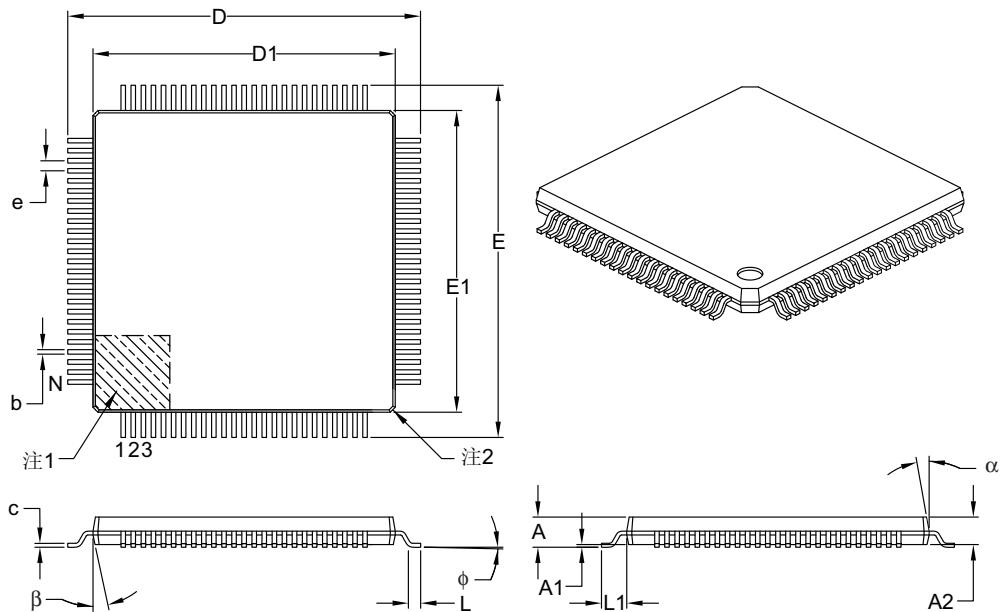
BSC： 基本尺寸。显示的是没有公差的理论精确值。

Microchip Technology图号C04-2092A

PIC24FJ256GB110 系列

100引脚塑封薄型正方扁平封装 (PT) —— 主体12x12x1 mm, 2.00 mm [TQFP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N	100		
引脚间距	e	0.40 BSC		
总高度	A	—	—	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	—	0.15
底脚长度	L	0.45	0.60	0.75
引脚投影长度	L1	1.00 REF		
底脚倾斜角	φ	0°	3.5°	7°
总宽度	E	14.00 BSC		
总长度	D	14.00 BSC		
塑模封装宽度	E1	12.00 BSC		
塑模封装长度	D1	12.00 BSC		
引脚厚度	c	0.09	—	0.20
引脚宽度	b	0.13	0.18	0.23
塑模顶部锥度	α	11°	12°	13°
塑模底部锥度	β	11°	12°	13°

注：

1. 引脚1的可见定位标记可能不同，但必须在阴影区域内。
2. 切角处的斜面是可选的；大小可能有所不同。
3. 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0.25毫米。
4. 尺寸和公差遵循ASME Y14.5M。

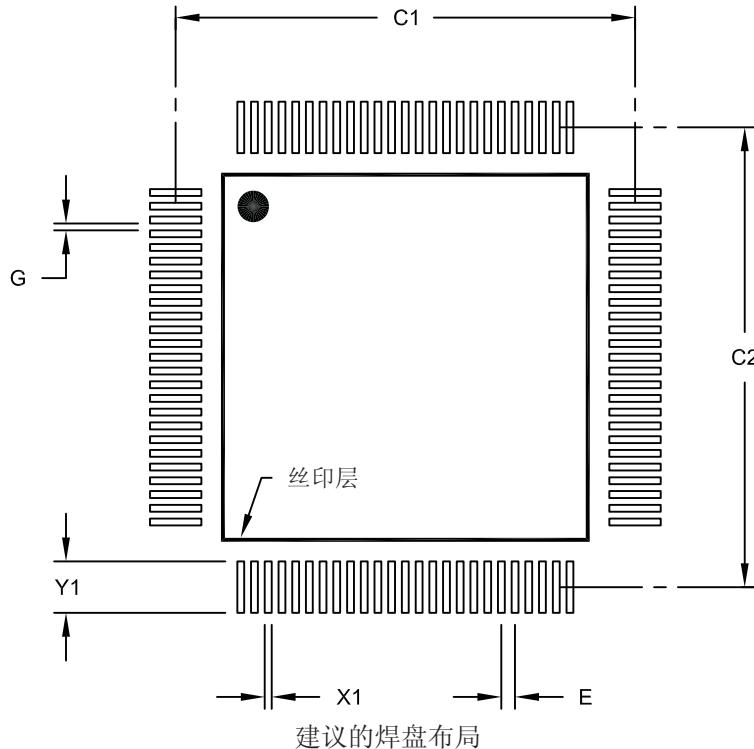
BSC：基本尺寸。显示的是没有公差的理论精确值。

REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology图号C04-100B

100引脚塑封薄型正方扁平封装 (PT) —— 主体12x12x1 mm, 2.00 mm [TQFP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
触点间距	E		0.40	BSC
接触焊垫间距	C1		13.40	
接触焊垫间距	C2		13.40	
接触焊垫宽度 (X100)	X1			0.20
接触焊垫长度 (Y1)	Y1			1.50
焊垫之间的距离	G	0.20		

注：

1. 尺寸和公差遵循ASME Y14.5M

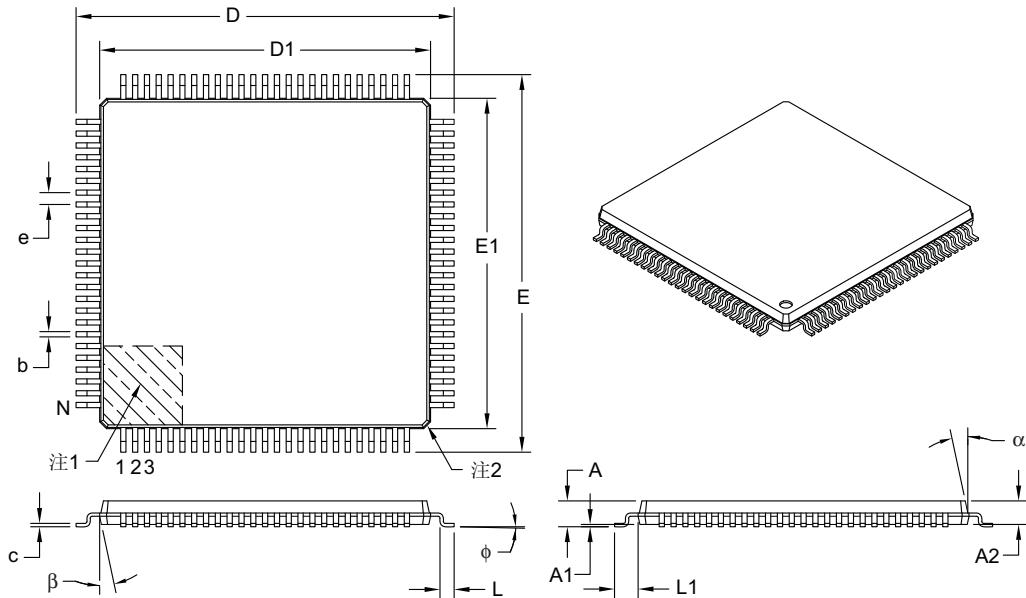
BSC：基本尺寸。显示的是没有公差的理论精确值。

Microchip Technology图号C04-2100A

PIC24FJ256GB110 系列

100引脚塑封薄型正方扁平封装 (PF) —— 主体14x14x1 mm, 2.00 mm [TQFP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



尺寸范围	单位	毫米		
		最小	正常	最大
引脚数	N	100		
引脚间距	e	0.50 BSC		
总高度	A	—	—	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	—	0.15
底脚长度	L	0.45	0.60	0.75
引脚投影长度	L1	1.00 REF		
底脚倾斜角	phi	0°	3.5°	7°
总宽度	E	16.00 BSC		
总长度	D	16.00 BSC		
塑模封装宽度	E1	14.00 BSC		
塑模封装长度	D1	14.00 BSC		
引脚厚度	c	0.09	—	0.20
引脚宽度	b	0.17	0.22	0.27
塑模顶部锥度	alpha	11°	12°	13°
塑模底部锥度	beta	11°	12°	13°

注：

1. 引脚1的可见定位标记可能不同，但必须在阴影区域内。
2. 切角处的斜面是可选的；大小可能有所不同。
3. 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0.25毫米。
4. 尺寸和公差遵循ASME Y14.5M。

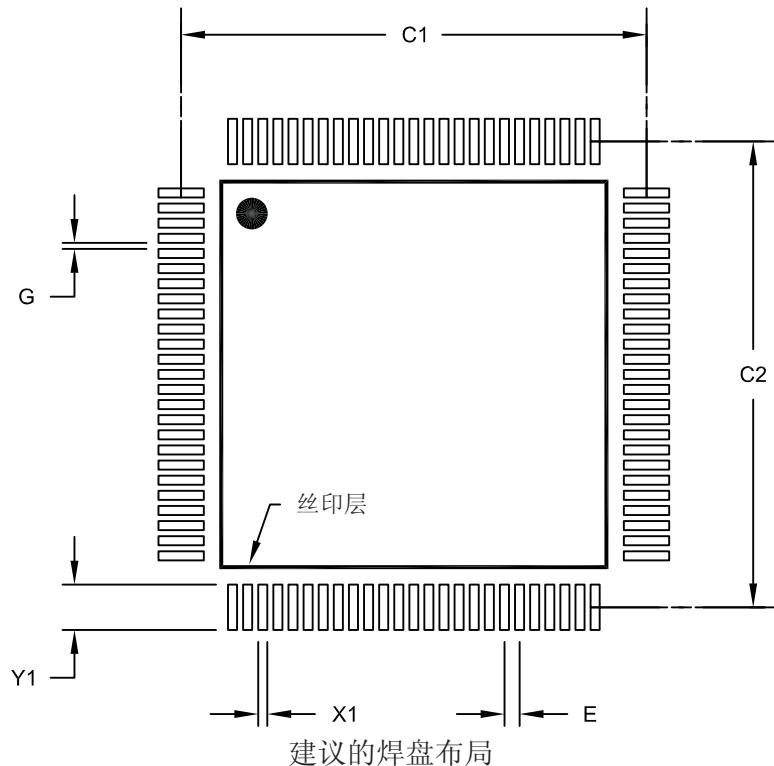
BSC：基本尺寸。显示的是没有公差的理论精确值。

REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology图号C04-110B

100引脚塑封薄型正方扁平封装 (PF) —— 主体14x14x1 mm, 2.00 mm [TQFP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
触点间距	E		0.50	BSC
接触焊垫间距	C1		15.40	
接触焊垫间距	C2		15.40	
接触焊垫宽度 (X100)	X1			0.30
接触焊垫长度 (Y100)	Y1			1.50
焊垫之间的距离	G	0.20		

注：

1. 尺寸和公差遵循ASME Y14.5M

BSC: 基本尺寸。显示的是没有公差的理论精确值。

Microchip Technology图号C04-2110A

PIC24FJ256GB110 系列

注:

附录 A: 版本历史

版本 A (2007 年 10 月)

PIC24FJ256GB110 系列器件的原始数据手册。

版本 B (2008 年 3 月)

更改了第 29.0 节“电气特性”，并对全文做了少量编辑。

版本 C (2009 年 12 月)

更新了所有的引脚图，以反映复用外设的正确优先级顺序。

在第 30.0 节“封装信息”和产品标识体系中添加了新型 64 脚 QFN 封装的封装信息。

更新了第 5.0 节“闪存程序存储器”，修改了汇编语言代码示例，添加了 C 语言代码示例。

对第 6.2 节“器件复位时间”进行了修改，尤其是表 6-3。

在第 4.0 节“存储器构成”和第 7.0 节“中断控制器”中添加了 INTTREG 寄存器。

在第 10.0 节“I/O 端口”中添加了一些内容并做了一些修改，包括：

- 修改了第 10.4.2.1 节“外设引脚选择功能优先级”
- 修改了表 10-3“可选择的输出源”

在第 18.0 节“带 On-The-Go 支持的通用串行总线 (USB OTG)”中添加了一些内容并做了一些修改，包括：

- 将 U1CON<x>位的名称从 RESET 改为 USBRST
- 将原先的第 18.3 节替换为第 18.1 节“硬件配置”，包括展开讨论了如何在不同的 USB 模式下，使单片机与应用接口

使用新的图示和更新的第 21.1 节“用户接口”修改了第 21.0 节“可编程循环冗余校验 (CRC) 发生器”。

更新了第 22.0 节“10 位高速 A/D 转换器”，将所有 AD1CHS0 改为 AD1CHS（整篇文档都做了这样的更改）。还修改对寄存器位域的说明，AD1CON3 (bit<7:0>) 和 AD1CHS (bit<2:8>)。

对第 23.0 节“三比较器模块”(寄存器 23-1) 和第 25.0 节“充电时间测量单元 (CTMU)”(寄存器 25-1) 中位的说明做了少量更改。

更新了第 26.0 节“特殊功能”，对 POR 和待机模式下稳压器的操作进行了修改。

更新了第 26.5 节“JTAG 接口”删除了通过接口编程的内容。

对第 29.0 节“电气特性”做了多处添加和修改，包括：

- 添加了在 60°C 条件下工作时的 IPD 规范
- VBOR、VBG、TBG 和 ICNPD 的新直流特性
- 添加了 VDDCORE 的新 VPEW 规范
- 内部振荡器起振时间 (TLPRC) 的新交流特性
- 将内部 RC 精度的所有信息都组合到单个表中

通篇做了少量排版修正。

PIC24FJ256GB110 系列

注:

索引

A

A/D 转换器	
模拟输入模型	275
传递函数	275

B

版本历史	341
备用中断向量表 (AIVT)	77
变更通知客户服务	349
并行主端口。参见 PMP。	241
不同缓冲模式下缓冲区描述符的分配	213

C

C 编译器	
MPLAB C18	300
CPU	
编程模型	33
控制寄存器	36
内核寄存器	35
算术逻辑单元 (ALU)	37
CRC	
设置示例	263
用户接口	264
CTMU	
测量电容	283
测量时间	284
脉冲延时和生成	284
参考时钟输出	129
产品标识体系	351
程序存储器	
程序空间可视性	62
存储器映射	39
地址构成	59
地址空间	39
构成	40
闪存配置字	40
使用表指令访问	61
程序空间可视性 (PSV)	62
充电时间测量单元。参见 CTMU。	
串行外设接口。参见 SPI。	

D

打盹模式	132
代码保护	295
代码段保护	295
配置选项	296
配置保护	296

代码示例

擦除程序存储器块	66, 67
单字闪存编程	69
I/O 端口读 / 写	135
基本时钟切换示例	127
配置 UART1 输入和输出功能 (PPS)	140
启动编程序列	68
装载写缓冲器	67
带专用定时器的输出比较	173
带专用定时器的输入捕捉	169
电气特性	
A/D 规范	325
电流规范	314 - 316
电压规范	313
规范的负载条件和要求	321
绝对最大额定值	311
内部 RC 精度	323
PLL 时钟规范	323
热条件	312
V/F 关系图	312
外部时钟	322
稳压器规范	320
读者反馈表	350

E

ENVREG 引脚	293
-----------	-----

F

封装	327
标识	327
详细信息	329
复位	
BOR (欠压复位)	71
CM (配置不匹配复位)	71
IOPUWR (非法操作码复位)	71
MCLR (引脚复位)	71
POR (上电复位)	71
器件时间	73
RCON 标志操作	73
SFR 状态	75
SWR (RESET 指令)	71
时钟源选择	73
TRAPR (陷阱冲突复位)	71
WDT (看门狗定时器复位)	71
UWR (未初始化的 W 寄存器复位)	71
延时	74

PIC24FJ256GB110 系列

G

公式

A/D 转换时钟周期.....	274
波特率重载计算.....	193
计算 PWM 周期.....	176
计算最大 PWM 分辨率.....	177
器件工作频率和 SPI 时钟速度之间的关系.....	190
RTCC 校准.....	260
UART 波特率 (BRGH = 0).....	200
UART 波特率 (BRGH = 1).....	200

H

汇编器

MPASM 汇编器	300
-----------------	-----

I

I/O 端口

并行 (PIO)	133
漏极开路配置	133
模拟端口引脚配置	134
上拉和下拉	135
输入状态变化通知	135
外设引脚选择	135

I²C

保留的地址	193
从地址掩码	193
设置作为总线主器件的波特率	193
时钟速率	193

IrDA 支持	201
---------------	-----

J

寄存器

AD1CSSL (A/D 输入扫描选择, 低字节)	274
AD1CHS0 (A/D 输入选择)	272
AD1CON1 (A/D 控制 1)	269
AD1CON2 (A/D 控制 2)	270
AD1CON3 (A/D 控制 3)	271
AD1PCFG (A/D 端口配置)	273
ALCFCRPT (闹钟配置)	255
ALMINSEC (闹钟分钟和秒值)	259
ALMTHDY (闹钟月和日值)	258
ALWDHR (闹钟星期和小时值)	259
BDnSTAT 原型 (缓冲器描述符 n 状态, CPU 模式)	215
BDnSTAT 原型 (缓冲器描述符 n 状态, USB 模式)	214
CLKDIV (时钟分频)	125
CMSTAT (比较器状态)	280
CMxCON (比较器 x 控制)	279
CORCON (CPU 控制)	37
CORCON (CPU 内核控制)	81
CRCCON (CRC 控制)	265
CRCXOR (CRC XOR 多项式)	266
CTMUCON (CTMU 控制)	285
CTMUICON (CTMU 电流控制)	286

CW1 (闪存配置字 1)	288
CW2 (闪存配置字 2)	290
CW3 (闪存配置字 3)	291
CVRCON (比较器参考电压控制)	282
DEVID (器件 ID)	292
DEVREV (器件版本)	292
I2CxCON (I2Cx 控制)	194
I2CxMSK (I2C 从模式地址掩码)	198
I2CxSTAT (I2Cx 状态)	196
ICxCON1 (输入捕捉 x 控制 1)	171
ICxCON2 (输入捕捉 x 控制 2)	172
IECn (中断允许控制 0-5)	90 - 96
IFS _n (中断标志状态 0-5)	84 - 89
INTCON1 (中断控制 1)	82
INTCON2 (中断控制 2)	83
IPCr (中断优先级控制 0-23)	97 - 117
MINSEC (RTCC 分钟和秒值)	257
MTHDY (RTCC 月和日值)	256
NVMCON (闪存存储器控制)	65
OCxCON1 (输出比较 x 控制 1)	178
OCxCON2 (输出比较 x 控制 2)	179
OSCCON (振荡器控制)	123
OSCTUN (FRC 振荡器调节)	126
PADCFG1 (焊垫配置)	254
PADCFG1 (焊垫配置控制)	247
PMADDR (PMP 地址)	245
PMAEN (PMP 使能)	245
PMMODE (并行端口模式)	244
PMPCON (PMP 控制)	242
PMSTAT (PMP 状态)	246
RCFGCAL (RTCC 校准和配置)	253
RCON (复位控制)	72
REFOCON (参考振荡器控制)	130
RPINR _n (PPS 输入映射 0-29)	141 - 151
RPOR _n (PPS 输出映射 0-15)	151 - 159
SPIxCON1 (SPIx 控制 1)	186
SPIxCON2 (SPIx 控制 2)	187
SPIxSTAT (SPIx 状态)	184
SR (ALU STATUS)	36, 81
T1CON (Timer1 控制)	162
TxCON (Timer2 和 Timer4 控制)	166
TyCON (Timer3 和 Timer5 控制)	167
U1ADDR (USB 地址)	228
U1CNFG1 (USB 配置 1)	229
U1CNFG2 (USB 配置 2)	230
U1CON (USB 控制, 器件模式)	226
U1CON (USB 控制, 主机模式)	227
U1EIE (USB 错误中断允许)	237
U1EIR (USB 错误中断状态)	236
U1EPn (USB 端点 n 控制)	238
U1IE (USB 中断允许)	235
U1IR (USB 中断状态, 器件模式)	233
U1IR (USB 中断状态, 主机模式)	234

U1OTGCON (USB OTG 控制)	223
U1OTGIE (USB OTG 中断允许)	232
U1OTGIR (USB OTG 中断状态)	231
U1OTGSTAT (USB OTG 状态)	222
U1PWMCON USB (VBUS PWM 发生器控制)	239
U1PWRC (USB 电源控制)	224
U1SOF (USB OTG Start-Of-Token 门限值)	229
U1STAT (USB 状态)	225
U1TOK (USB 令牌)	228
WKDYHHR (RTCC 星期和小时值)	257
UxMODE (UARTx 模式)	202
UxSTA (UARTx 状态和控制)	204
YEAR (RTCC 年值)	256
寄存器映射	
A/D 转换器	53
比较器	56
并行主 / 从端口	55
CPU 内核	43
CRC	56
CTMU	53
定时器	46
焊垫配置	52
I ² C	49
ICN	44
NVM	58
PMD	58
PORTA	51
PORTB	51
PORTC	51
PORTD	51
PORTE	52
PORTF	52
PORTG	52
RTCC	56
SPI	50
输出比较	48
输入捕捉	47
UART	50
USB OTG	54
外设引脚选择	57
系统	58
中断控制器	45
JTAG 接口	297
基于指令的节能模式	131, 132
交流特性	
ADC 转换时序	326
CLKO 和 I/O 时序	324
内部 RC 精度	323
节能	
时钟频率和时钟切换	131
节能特性	131
仅自供电	209
仅总线电源	209

K

开发支持	299
看门狗定时器 (WDT)	294
窗口操作	295
控制寄存器	295
勘误表	9
客户通知服务	349
客户支持	349
空闲模式	132
框图	
10 位高速 A/D 转换器	268
16 位 Timer1 模块	161
16 位同步 Timer2 和 Timer4	165
16 位异步 Timer3 和 Timer5	165
32 位 Timer2/3 和 Timer4/5	164
比较器参考电压	281
表寄存器的寻址	63
CALL 堆栈帧	59
CPU 编程模型	35
测量时间的 CTMU 典型连接和内部配置	284
单独的比较器配置	278
电容测量的 CTMU 连接和内部配置	283
端点缓冲模式的 BDT 映射	212
访问程序空间内数据的地址生成方式	60
复位系统	71
共用 I/O 端口结构	133
I ² C 模块	192
看门狗定时器 (WDT)	295
可寻址的 PMP 示例	248
LCD 控制	250
PCI24FJ256GB110 系列 (一般)	16
PIC24F CPU 内核	34
PMP 8 位复用地址和数据应用	250
PMP EEPROM (8 位数据)	250
PMP 并行 EEPROM (16 位数据)	250
PMP 部分复用的寻址	249
PMP 复用寻址	249
PMP 模块概述	241
PMP 主模式, 部分复用的寻址	249
PMP 主模式, 完全复用的寻址	249
PMP 主模式, 未复用的寻址	248
PSV 操作	62
片上稳压器连接	293
RTCC	251
SPI 从 - 帧从模式连接	189
SPI 从 - 帧主模式连接	189
SPIx 模块 (标准模式)	182
SPIx 模块 (增强型模式)	183
SPI 主 / 从连接 (标准模式)	188
SPI 主 / 从连接 (增强型缓冲模式)	188
SPI 主 - 帧从模式连接	189
SPI 主 - 帧主模式连接	189
三比较器模块	277

PIC24FJ256GB110 系列

生成脉冲延时的 CTMU 典型连接和内部配置	284
使用表操作访问程序存储器.....	61
输出比较（16 位模式）.....	174
输出比较（双重缓冲的 16 位 PWM 模式）.....	176
输入捕捉.....	169
UART（简化）.....	199
USB OTG 模块.....	208
仅总线供电.....	209
USB OTG 中断逻辑.....	216
USB PLL.....	128
系统时钟图.....	121
传统 PMP 示例.....	248
M	
MPLAB ASM30 汇编器、链接器和库管理器	300
MPLAB PM3 器件编程器.....	302
MPLAB REAL ICE 在线仿真器系统	301
MPLAB 集成开发环境软件	299
MPLINK 目标链接器 /MPLIB 目标库管理器.....	300
脉宽调制。参见 PWM。	
脉宽调制（PWM）模式.....	175
N	
Near 数据空间	42
内部集成电路。参见 I ² C。.....	191
内核特性.....	11
P	
POR	
和片上稳压器	294
PWM	
占空比和周期	176
配置位	287
Q	
器件特性（汇总）	
100 引脚	15
64 引脚	13
80 引脚	14
R	
RTCC	
寄存器映射	252
闹钟配置.....	260
校准.....	260
软件堆栈.....	59
软件模拟器（MPLAB SIM）.....	301
S	
SFR 空间	42
SPI	
闪存程序存储器.....	63
编程算法	66
单字编程	69
和表指令	63
JTAG 工作原理	64
RTSP 工作原理.....	64
增强型 ICSP 工作原理	64
闪存配置字	40, 287 - 291
上电要求	294
时序图	
CLKO 和 I/O 时序	324
外部时钟	322
输出比较	
32 位模式	173
同步和触发模式	173
数据存储器	
存储器映射	41
地址空间	41
空间构成	42
Near 数据空间.....	42
软件堆栈	59
SFR 空间	42
输入捕捉	
32 位模式	170
同步和触发模式	169
双电源	209
T	
Timer2/3 和 Timer4/5	163
Timer1	161
特殊功能	12
通用串行总线。参见 USB OTG。	
通用异步收发器。参见 UART。	
U	
UART	199
波特率发生器（BRG）	200
发送	
8 位数据模式	201
9 位数据模式	201
间隔和同步序列	201
接收	201
UxCTS 和 UxRTS 引脚的操作	201

USB On-The-Go (OTG)	12
USB OTG	
DMA 接口	213
硬件配置	
器件模式	209
外部接口	211
收发器功耗需求	211
缓冲器描述符和 BDT	212
寄存器	221 - 239
OTG 工作原理	220
器件模式的操作	217
VBUS 电压产生	216
中断	216
主机模式的操作	218
V	
VDDCORE/VCAP 引脚	293
W	
WWW, 在线支持	9
外设模块禁止位	132
外设使能位	132
外设引脚选择 (PPS)	135
寄存器	141 - 159
可用的外设和引脚	136
配置控制	139
使用注意事项	140
输出映射	136
输入映射	136
外设优先级	136
映射例外	139
稳压器 (片上)	293
待机模式	294
跟踪模式	293
和 BOR	294
X	
休眠模式	131
选择性外设功耗控制	132
Y	
引脚说明	17 - 25
Z	
振荡器配置	
POR 时的初始配置	122
时钟切换	126
过程	127
时钟选择	122
USB 操作	128
特别注意事项	129
指令集	
概述	305
汇总	303
直流特性	
程序存储器	319
I/O 引脚输出规范	319
I/O 引脚输入规范	318
中断	
和复位过程	77
控制和状态寄存器	80
设置和服务过程	119
陷阱向量	78
向量表	78
已实现的向量	79
中断向量表 (IVT)	77

PIC24FJ256GB110 系列

注:

MICROCHIP 网站

Microchip 网站 (www.microchip.com) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的因特网浏览器即可访问。网站提供以下信息：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及存档软件
- **一般技术支持**——常见问题 (FAQ)、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 www.microchip.com。在“支持”(Support) 下，点击“变更通知客户”(Customer Change Notification) 服务后按照注册说明完成注册。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

客户应联系其代理商、代表或应用工程师 (FAE) 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过<http://support.microchip.com>获得网上技术支持。

PIC24FJ256GB110 系列

读者反馈表

我们努力为您提供最佳文档，以确保您能够成功使用 Microchip 产品。如果您对文档的组织、条理性、主题及其他有助于提高文档质量的方面有任何意见或建议，请填写本反馈表并传真给我公司 TRC 经理，传真号码为 86-21-5407-5066。请填写以下信息，并从下面各方面提出您对本文档的意见。

致 TRC 经理

总页数 _____

关于： 读者反馈

发自： 姓名 _____

公司 _____

地址 _____

国家 / 省份 / 城市 / 邮编 _____

电话 (_____) _____ 传真 (_____) _____

应用 (选填)：

您希望收到回复吗？是_____ 否_____

器件： PIC24FJ256GB110 系列

文献编号：

DS39897C_CN

问题：

1. 本文档中哪些部分最有特色？

2. 本文档是否满足了您的软硬件开发要求？如何满足的？

3. 您认为本文档的组织结构便于理解吗？如果不便于理解，那么问题何在？

4. 您认为本文档应该添加哪些内容以改善其结构和主题？

5. 您认为本文档中可以删减哪些内容，而又不会影响整体使用效果？

6. 本文档中是否存在错误或误导信息？如果存在，请指出是什么信息及其具体页数。

7. 您认为本文档还有哪些方面有待改进？

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

<p style="text-align: center;">PIC 24 FJ 256 GB1 10 T - I / PT - XXX</p> <p>Microchip 的商标 <input type="text"/></p> <p>架构 <input type="text"/></p> <p>闪存系列 <input type="text"/></p> <p>程序存储器容量 (KB) <input type="text"/></p> <p>产品组 <input type="text"/></p> <p>引脚数 <input type="text"/></p> <p>卷带标志 (如果适用) <input type="text"/></p> <p>温度范围 <input type="text"/></p> <p>封装 <input type="text"/></p> <p>模式 <input type="text"/></p>		<p>示例：</p> <p>a) PIC24FJ64GB106-I/PT： 带 USB On-The-Go 的 PIC24F 器件，64 KB 程序存储器，64 引脚，工业级温度，TQFP 封装。</p> <p>b) PIC24FJ256GB110-I/PT： 带 USB On-The-Go 的 PIC24F 器件， 256 KB 程序存储器，100 引脚，工业级温度， TQFP 封装。</p>											
<p>架构 24 = 16 位改进型哈佛架构，不带 DSP</p> <p>闪存系列 FJ = 闪存程序存储器</p> <p>产品组 GB1 = 带 USB On-The-Go 的通用单片机</p> <p>引脚数</p> <table><tr><td>06</td><td>= 64 引脚</td></tr><tr><td>08</td><td>= 80 引脚</td></tr><tr><td>10</td><td>= 100 引脚</td></tr></table> <p>温度范围 I = -40°C 至 +85°C (工业级)</p> <p>封装</p> <table><tr><td>PF</td><td>= 100 引脚 (14x14x1 mm) TQFP (薄型正方扁平封装)</td></tr><tr><td>PT</td><td>= 64 引脚、80 引脚和 100 引脚 (12x12x1 mm) TQFP (薄型正方扁平封装)</td></tr><tr><td>MR</td><td>= 64 引脚 (9x9x0.9 mm) QFN (正方扁平无引脚封装)</td></tr></table> <p>模式</p> <p>三位 QTP、SQTP、代码或特殊要求 (其他情况空白)</p> <p>ES = 工程样片</p>		06	= 64 引脚	08	= 80 引脚	10	= 100 引脚	PF	= 100 引脚 (14x14x1 mm) TQFP (薄型正方扁平封装)	PT	= 64 引脚、80 引脚和 100 引脚 (12x12x1 mm) TQFP (薄型正方扁平封装)	MR	= 64 引脚 (9x9x0.9 mm) QFN (正方扁平无引脚封装)
06	= 64 引脚												
08	= 80 引脚												
10	= 100 引脚												
PF	= 100 引脚 (14x14x1 mm) TQFP (薄型正方扁平封装)												
PT	= 64 引脚、80 引脚和 100 引脚 (12x12x1 mm) TQFP (薄型正方扁平封装)												
MR	= 64 引脚 (9x9x0.9 mm) QFN (正方扁平无引脚封装)												



MICROCHIP

全球销售及服务网点

美洲

公司总部 Corporate Office

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持：
<http://www.microchip.com/>
support

网址: www.microchip.com

亚特兰大 Atlanta

Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

波士顿 Boston

Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago

Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

克里夫兰 Cleveland

Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

达拉斯 Dallas

Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit

Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

印第安纳波利斯

Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453

洛杉矶 Los Angeles

Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

圣克拉拉 Santa Clara

Santa Clara, CA
Tel: 1-408-961-6444
Fax: 1-408-961-6445

加拿大多伦多 Toronto

Mississauga, Ontario,
Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 Asia Pacific Office

Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong

Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 北京
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 香港特别行政区
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

中国 - 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门
Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

台湾地区 - 高雄
Tel: 886-7-213-7830
Fax: 886-7-330-9305

台湾地区 - 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

亚太地区

台湾地区 - 新竹

Tel: 886-3-6578-300
Fax: 886-3-6578-370

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

日本 Japan - Yokohama
Tel: 81-45-471-6166
Fax: 81-45-471-6122

韩国 Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 Austria - Wels

Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark-Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

英国 UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820