

## 支持ISO 7816、具有加密引擎和USB On-The-Go且 采用 XLP 技术的 28/44 引脚 16 位通用闪存单片机

### 加密引擎

- 支持 128、192 或 256 位密钥的 AES 引擎
- 支持 ECB、CBC、OFB、CTR 和 CFB128 模式
- DES/ 三重 DES (Triple DES, TDES) 引擎: 支持 2 密钥和 3 密钥 EDE 或 DED TDES
- 最多支持 3 个 TDES 唯一密钥
- 编程安全
- 真随机数发生器
- 伪随机数发生器
- 不可读的片内 OTP 密钥储存器

### 通用串行总线特性

- 符合 USB v2.0 On-The-Go (OTG) 规范
- 双角色能力: 可充当主机或外设
- 主机模式下低速 (1.5 Mb/s) 和全速 (12 Mb/s) USB 操作
- 设备模式下的全速 USB 操作
- 用于 USB 的高精度 PLL
- 使用 FRC 振荡器的 USB 设备模式操作:
  - 不需要晶振
- 最多支持 32 个端点 (16 个双向端点):
  - USB 模块可将器件上的任何 RAM 地址单元用作 USB 端点缓冲区
- 片上 USB 收发器
- 支持控制、中断、同步和批量传输
- 片内上拉和下拉电阻

### 超低功耗特性

- 多种功耗管理选项可将功耗降至超低水平:
  - VBAT 允许器件切换到备用电池, 将使用 RTCC 时的功耗降至最低
  - 深度休眠能够实现接近于完全掉电的运行状态, 同时能够在内部或外部触发时唤醒
  - 休眠和空闲模式有选择地关闭外设和 / 或内核, 从而大幅降低功耗和实现快速唤醒
  - 打盹模式允许 CPU 以低于外设的时钟速度运行
- 备用时钟模式允许即时切换到低时钟速度以选择性降低功耗
- 深度休眠模式下的超低功耗电流消耗:
  - WDT: 3.3V 时典型值为 270 nA
  - RTCC: 32 kHz、3.3V 时典型值为 400 nA
  - 深度休眠电流: 3.3V 时典型值为 40 nA

器件	存储器		引脚数	模拟外设			数字外设							USB OTG	深度休眠 w/VBAT	AES/DES 加密
	程序闪存 (字节)	数据 RAM (字节)		10/12 位 A/D (通道数)	比较器	CTMU (通道数)	输入捕捉	输出比较 /PWM	I <sup>2</sup> C™	SPI	UART w/IrDA® 7816	EPMP/ISP	16 位定时器			
PIC24FJ128GB204	128K	8K	44	12	3	12	6	6	2	3	4	有	5	有	有	有
PIC24FJ128GB202	128K	8K	28	9	3	9	6	6	2	3	4	无	5	有	有	有
PIC24FJ64GB204	64K	8K	44	12	3	12	6	6	2	3	4	有	5	有	有	有
PIC24FJ64GB202	64K	8K	28	9	3	9	6	6	2	3	4	无	5	有	有	有

# PIC24FJ128GB204 系列

## 模拟特性

- 12 通道的 10/12 位模数 (Analog-to-Digital, A/D) 转换器:
  - 500 ksps (10 位) 和 200 ksps (12 位) 的转换速率
  - 可在休眠和空闲模式下进行转换
- 3 个带可编程输入 / 输出配置的轨到轨增强型模拟比较器
- 3 个片上可编程参考电压
- 充电时间测量单元 (Charge Time Measurement Unit, CTMU):
  - 用于电容触摸传感, 最多 12 路通道
  - 时间测量分辨率达 100 ps
  - 休眠模式下工作

## 外设特性

- 最多 5 个外部中断源
- 外设引脚选择 (Peripheral Pin Select, PPS): 允许对许多外设进行独立的 I/O 映射
- 5 个带预分频器的 16 位定时器 / 计数器:
  - 可以配对成 32 位定时器 / 计数器
- 6 通道 DMA 支持所有外设模块:
  - 最大程度减小 CPU 开销并提高数据吞吐量
- 6 个输入捕捉模块, 每个带有一个专用 16 位定时器
- 6 个输出比较 / PWM 模块, 每个带有一个专用 16 位定时器
- 增强型并行主 / 从端口 (Enhanced Parallel Master/Slave Port, EPMP/EPSP)
- 硬件实时时钟 / 日历 (Real-Time Clock/Calendar, RTCC):
  - 在休眠、深度休眠和 VBAT 模式下运行
- 3 个 3 线 / 4 线 SPI 模块:
  - 支持 4 帧模式
  - 可变的 FIFO 缓冲区
  - I<sup>2</sup>S 模式
  - 可变宽度从 2 位至 32 位
- 2 个 I<sup>2</sup>C™ 模块, 支持多主 / 从模式和 7 位 / 10 位寻址
- 4 个 UART 模块:
  - 支持 RS-485、RS-232 和 LIN/J2602
  - 用于 IrDA® 的片上硬件编码器 / 解码器
  - 智能卡 ISO 7816 仅在 UART1 和 UART2 上受支持:
    - T = 0 协议, 具有自动错误处理功能
    - T = 1 协议
    - 专用保护时间计数器 (Guard Time Counter, GTC)
    - 专用等待时间计数器 (Waiting Time Counter, WTC)
    - 自动波特率检测 (Auto-Baud Detect, ABD) 时自动唤醒
    - 4 级深 FIFO 缓冲区
- 可编程 32 位循环冗余校验 (Cyclic Redundancy Check, CRC) 发生器
- 数字信号调制器为数字信号流提供片上 FSK 和 PSK 调制
- 所有 I/O 引脚上的高拉 / 灌电流 (18 mA/18 mA)
- 数字 I/O 引脚上的可配置漏极开路输出
- 多数引脚上最大容许输入电压为 5.5V

## 高性能 CPU

- 改进型哈佛架构
- 最高运行速度可达 16 MIPS (32 MHz 时)
- 8 MHz 内部振荡器:
  - 96 MHz PLL 选项
  - 多个时钟分频选项
  - 运行时自校准功能, 能保持精度好于 ±0.20%
  - 快速启动
- 17 位 x 17 位单周期硬件小数 / 整数乘法器
- 32 位 / 16 位硬件除法器
- 16 x 16 位工作寄存器阵列
- 优化的 C 编译器指令集架构 (Instruction Set Architecture, ISA)
- 2 个地址发生单元 (Address Generation Unit, AGU), 分别用于数据存储器的读 / 写寻址

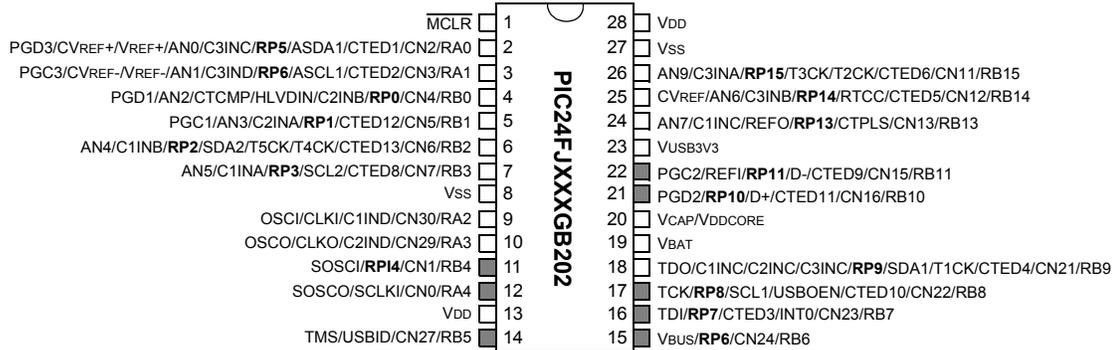
## 单片机特性

- 供电电压范围为 2.0V 至 3.6V
- 2 个片内稳压器 (1.8V 和 1.2V) 用于普通和超低功耗工作
- 可承受 20,000 次擦 / 写 (典型值) 的闪存程序存储器
- 闪存数据的保持时间: 最少 20 年
- 可在软件控制下自编程
- 可编程参考时钟输出
- 通过 2 个引脚进行在线串行编程 (In-Circuit Serial Programming™, ICSP™) 和在线仿真 (In-Circuit Emulation, ICE)
- 支持 JTAG 编程和边界扫描
- 故障保护时钟监视器 (Fail-Safe Clock Monitor, FSCM) 操作:
  - 检测时钟故障并切换至片内低功耗 RC 振荡器
- 上电复位 (Power-on Reset, POR)、上电延时定时器 (Power-up Timer, PWRT) 和振荡器起振定时器 (Oscillator Start-up Timer, OST)
- 独立的欠压复位 (Brown-out Reset, BOR) 和深度休眠欠压复位 (Deep Sleep Brown-out Reset, DSBOR) 电路
- 可编程高 / 低电压检测 (High/Low-Voltage Detect, HLVD)
- 灵活的看门狗定时器 (Watchdog Timer, WDT), 自带 RC 振荡器以实现可靠运行
- 标准和超低功耗看门狗定时器 (Ultra Low-Power Watchdog Timer, ULPW), 用于在标准和深度休眠模式下可靠运行

# PIC24FJ128GB204 系列

## 引脚图

### 28 引脚 SPDIP、 SOIC 和 SSOP<sup>(1)</sup>



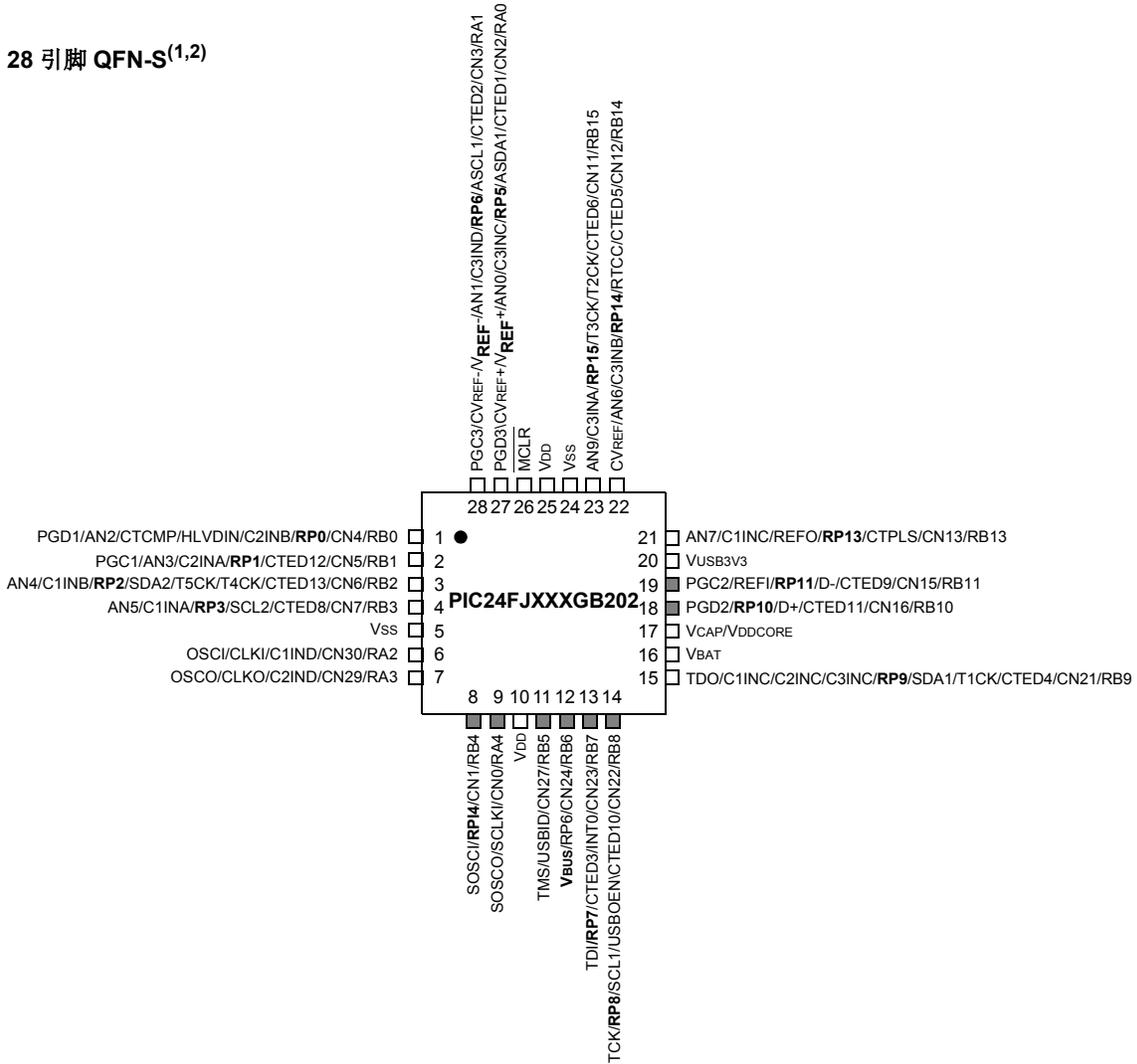
图注: RPn 表示可重新映射的外设引脚。

注 1: 灰色阴影框表示可承受 5.5V 电压的输入引脚。

# PIC24FJ128GB204 系列

## 引脚图 (续)

28 引脚 QFN-S<sup>(1,2)</sup>

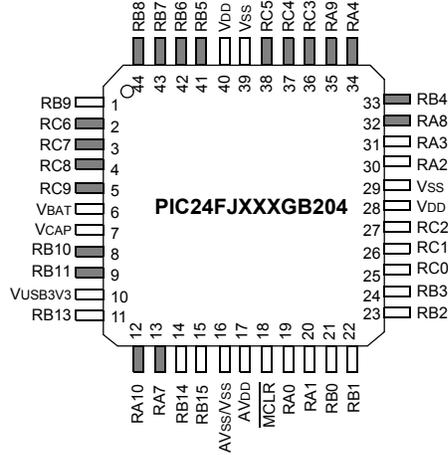


- 图注: RPn 表示可重新映射的外设引脚。
- 注 1: 灰色阴影框表示可承受 5.5V 电压的输入引脚。
- 2: QFN 器件背面的焊盘应连接到 Vss。

# PIC24FJ128GB204 系列

## 引脚图 (续)

44 引脚 TQFP 和  
44 引脚 QFN<sup>(1,2,3)</sup>



- 注
- 1: 灰色阴影框表示可承受 5.5V 电压的输入引脚。
  - 2: QFN 器件背面的焊盘应连接到 Vss。
  - 3: 有关引脚的详细说明, 请参见表 1。

表 1: PIC24FJXXXGB204 引脚功能描述

引脚	功能	引脚	功能
1	C1INC/C2INC/C3INC/ <b>RP9</b> /SDA1/T1CK/CTED4/PMD3/CN21/RB9	23	AN4/C1INB/ <b>RP2</b> /SDA2/T5CK/T4CK/CTED13/CN6/PMD2/RB2
2	<b>RP22</b> /PMA1/PMALH/CN18/RC6	24	AN5/C1INA/ <b>RP3</b> /SCL2/CTED8/CN7/PMWR/RB3
3	<b>RP23</b> /PMA0/PMALL/CN17/RC7	25	AN10/ <b>RP16</b> /PMBE1/CN8/RC0
4	<b>RP24</b> /PMA5/CN20/RC8	26	AN11/ <b>RP17</b> /CN9/RC1
5	<b>RP25</b> /CTED7/PMA6/CN19/RC9	27	AN12/ <b>RP18</b> /PMACK1/CN10/RC2
6	VBAT	28	VDD
7	VCAP	29	VSS
8	<b>RP10</b> /CTED11/CN16/PGD2/D+/RB10	30	OSCI/C1IND/CLKI/PMCS1/CN30/RA2
9	REFI/ <b>RP11</b> /CTED9/CN15/PGC2/D-/RB11	31	OSCO/C2IND/CLKO/CN29/RA3
10	VUSB3V3	32	TDO/PMA8/CN34/RA8
11	AN7/C1INC/REFO/ <b>RP13</b> /CTPLS/PMRD/CN13/RB13	33	SOSCI/CN1/ <b>RP14</b> /RB4
12	TMS/PMA2/PMALU/CN36/RA10	34	SOSCO/SCLKI/CN0/RA4
13	TCK/PMA7/CN33/RA7	35	TDI/PMA9/CN35/RA9
14	CVREF/AN6/C3INB/ <b>RP14</b> /RTCC/CTED5/CN12/RB14	36	<b>RP19</b> /PMBE0/CN28/RC3
15	AN9/C3INA/ <b>RP15</b> /T3CK/T2CK/CTED6/PMA14/CS1/CN11/PMCS/ PMCS1/RB15	37	<b>RP20</b> /PMA4/CN25/RC4
16	AVSS/VSS	38	<b>RP21</b> /PMA3/CN26/RC5
17	AVDD	39	VSS
18	MCLR	40	VDD
19	CVREF+/VREF+/AN0/C3INC/ <b>RP5</b> /ASDA1 <sup>(1)</sup> /CTED1/CN2/PM7/PGD3/RA0	41	CN27/USBID/RB5
20	CVREF-/VREF-/AN1/C3IND/ <b>RP6</b> /ASCL1 <sup>(1)</sup> /CTED2/CN3/PGC3/RA1	42	PMD6/CN24/VBus/RB6
21	AN2/CTCMP/C2INB/ <b>RP0</b> /CN4/PGD1/HLVDIN/PMD0/RB0	43	<b>RP7</b> /CTED3/INT0/CN23/PMD5/RB7
22	AN3/C2INA/ <b>RP1</b> /CTED12/CN5/PMD1/PGC1/RB1	44	<b>RP8</b> /SCL1/CTED10/PMD4/CN22/USBOEN/RB8

图注: **RPn** 表示可重新映射的外设引脚。

注 1: 当 I2C1SEL 配置位置 1 时, 可与 SDA1 和 SCL1 复用。

# PIC24FJ128GB204 系列

---

## 目录

1.0	器件概述	9
2.0	16 位单片机入门指南	23
3.0	CPU	29
4.0	存储器构成	35
5.0	直接存储器访问控制器 (DMA)	71
6.0	闪存程序存储器	79
7.0	复位	85
8.0	中断控制器	91
9.0	振荡器配置	147
10.0	节能特性	161
11.0	I/O 端口	173
12.0	Timer1	201
13.0	Timer2/3 和 Timer4/5	205
14.0	带专用定时器的输入捕捉	211
15.0	带专用定时器的输出比较	217
16.0	串行外设接口 (SPI)	227
17.0	I <sup>2</sup> C™	245
18.0	通用异步收发器 (UART)	253
19.0	带 On-The-Go 支持的通用串行总线 (USB OTG)	265
20.0	数据信号调制器 (DSM)	299
21.0	增强型并行主端口 (EPMP)	303
22.0	实时时钟和日历 (RTCC)	315
23.0	加密引擎	329
24.0	32 位可编程循环冗余校验 (CRC) 发生器	345
25.0	带阈值检测功能的 12 位 A/D 转换器	351
26.0	三比较器模块	371
27.0	比较器参考电压	377
28.0	充电时间测量单元 (CTMU)	379
29.0	高/低电压检测 (HLVD)	387
30.0	特殊功能	389
31.0	开发支持	403
32.0	指令集汇总	407
33.0	电气特性	415
34.0	封装信息	447
附录 A:	版本历史	465
索引		467
Microchip 网站		475
变更通知客户服务		475
客户支持		475
产品标识体系		477

## 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需求。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](mailto:CTRC@microchip.com)。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，请访问我公司网站：

<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。文献编号中紧跟数字串后的字母是版本号，例如：DS300000000A\_CN 是文档的 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 缙戛珑 <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](http://www.microchip.com) 上注册。

# PIC24FJ128GB204 系列

---

注:

## 1.0 器件概述

本文档包含以下器件的具体信息：

- PIC24FJ64GB202
- PIC24FJ128GB202
- PIC24FJ64GB204
- PIC24FJ128GB204

PIC24FJ128GB204 系列器件扩展了 PIC24F 系列器件的功能，不仅包含该系列器件的现有特性，还增加了一组完整的加密引擎、ISO 7816 支持和 I2S 支持。此组合与其超低功耗功能（用于外设的直接存储器访问（DMA）和 USB On-The-Go）一起，使此系列成为混合信号 PIC<sup>®</sup> 单片机的一种经济节能的新标准。

## 1.1 内核特性

### 1.1.1 16 位架构

所有 PIC24F 器件的内核都采用了 16 位改进型哈佛架构，该架构最初是在 Microchip 的 dsPIC<sup>®</sup> 数字信号控制器（Digital Signal Controller, DSC）中引入的。PIC24F CPU 内核提供了大量增强功能，例如：

- 16 位数据路径和 24 位地址路径，可在数据空间和程序空间之间传送信息
- 线性寻址能力最高可达 12 MB（程序空间）和 32 KB（数据空间）
- 利用内建软件堆栈支持 16 元工作寄存器阵列
- 支持整数算术运算的 17 位 x 17 位硬件乘法器
- 支持 32 位 / 16 位除法运算的硬件
- 指令集支持多种寻址模式并针对高级语言（如 C 语言）进行了优化
- 工作性能最高可达 16 MIPS

### 1.1.2 XLP 节能技术

PIC24FJ128GB204 系列器件具有许多能显著降低功耗的节能工作模式。新模式包括：

- 保持休眠，带由独立的低压稳压器供电的基本电路
- 不带 RTCC 的深度休眠，通过软件控制实现最低可能功耗
- VBAT 模式（带或不带 RTCC），当 VDD 移除时继续使用备用电池执行受限操作

许多新的低功耗模式还支持低功耗片上实时时钟和日历（RTCC）继续工作，使应用在器件进入休眠时仍能计时。

除了这些新特性之外，PIC24FJ128GB204 系列器件还具有先前 PIC24F 单片机的所有传统节能特性，例如：

- 动态时钟切换，允许在运行期间选择低功耗时钟
- 打盹模式操作，用于在降低 CPU 时钟速度时维持外设时钟速度
- 基于指令的节能模式，用于快速进入空闲和许多休眠模式

### 1.1.3 振荡器选项和特性

PIC24FJ128GB204 系列中的所有器件提供 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。这些功能包括：

- 两种晶振模式
- 两种外部时钟（External Clock, EC）模式
- 锁相环（Phase-Locked Loop, PLL）倍频器，可使时钟速度最高达到 32 MHz
- 标称输出值为 8 MHz 的快速内部振荡器（Fast Internal Oscillator, FRC），带多个分频选项并能在运行时自动进行频率自校准
- 具有 31 kHz 标称值的独立低功耗内部 RC 振荡器（LPRC），用于时序要求不高的低功耗应用。

内部振荡器模块还为故障保护时钟监视器（FSCM）提供了一个稳定的参考源。故障保护时钟监视器不断地监视主时钟源，将之与内部振荡器提供的参考信号作比较。一旦发生时钟故障，允许控制器将时钟源切换到内部振荡器，继续保持低速工作或安全地关闭应用。

### 1.1.4 易于移植

无论存储器容量如何，所有器件均共享同一组丰富外设，从而可在应用扩展和升级时方便地进行移植。这样仍然可以使用 Microchip 器件而将相对简单的应用顺利移植为强大而复杂的应用。

# PIC24FJ128GB204 系列

---

## 1.2 DMA 控制器

PIC24FJ128GB204 系列器件还在现有的 PIC24F 架构上增加了直接存储器访问 (DMA) 控制器。DMA 与 CPU 配合, 允许在数据存储器 and 外设之间传送数据, 而无需 CPU 干预, 提高了数据吞吐量并降低了执行时间开销。6 个可独立编程的通道使同时服务多个外设成可能, 每个通道外设执行不同的操作。支持多种类型数据传送操作。

## 1.3 USB On-The-Go (OTG)

USB On-The-Go 模块提供了与 USB 2.0 标准兼容的目标器件一样的片上功能, 以及与 USB 嵌入式主机那样的有限独立功能。通过实现 USB 主机协商协议 (Host Negotiation Protocol, HNP), 模块还可以在设备和主机操作之间动态切换, 从而可以在单片机平台上实现更多支持 USB 的应用。

PIC24FJ128GB204 系列器件也实现了集成 USB 收发器和高精度振荡器, 降低实现完整 USB 设备、嵌入式主机、双角色或 On-The-Go 应用所需的复杂性。

## 1.4 加密引擎

加密引擎提供了一组新的数据安全选项。使用其自己的独立状态机, 引擎可独立于 CPU 单独执行符合 NIST 标准的数据加密和解密。

支持真随机数生成 (True Random Number Generation, TRNG) 和伪随机数生成 (Pseudo Random Number Generation, PRNG); 符合 NIST SP800-90 标准。

## 1.5 其他特性

- **外设引脚选择 (PPS):** 外设引脚选择功能允许大部分的数字外设被映射到固定的数字 I/O 引脚集。用户可将许多数字外设之一的输入和 / 或输出独立地映射到其中的任一 I/O 引脚。
- **通信:** PIC24FJ128GB204 系列器件集成了许多串行通信外设以处理各种应用需求。提供了两个同时支持主从工作模式的独立 I<sup>2</sup>C™ 模块。通过 PPS 功能, 器件还具有 4 个带内置 IrDA® 编码器 / 解码器并支持 ISO 7816 智能卡 (仅 UART1 和 UART2) 的独立 UART, 以及 3 个带 I<sup>2</sup>S 并支持各种数据宽度的 SPI 模块。
- **模拟特性:** PIC24FJ128GB204 系列的所有器件都包含一个 12 位 A/D 转换器模块和一个三比较器模块。A/D 模块具有大量新特性, 允许转换器评估传入数据并据此作出决定, 从而减少日常 A/D 转换的 CPU 开销。比较器模块包含三个模拟比较器, 它们可针对多种操作进行配置。
- **CTMU 接口:** 除其他模拟功能外, PIC24FJ128GB204 系列器件还包含 CTMU 接口模块。此模块为精确时间测量和脉冲生成提供了一种便利的方法, 并可用作容性传感器的接口。
- **增强型并行主 / 从端口:** 此模块允许快速透明地访问单片机数据总线, 并支持 CPU 直接寻址外部数据存储器。并行端口可用于主或从模式, 支持 4、8 或 16 位数据宽度, 在主模式下最多寻址 23 位。
- **实时时钟和日历 (RTCC):** 此模块通过硬件实现带有闹钟功能的全功能时钟和日历, 从而释放了定时器资源和程序存储空间供核心应用使用。
- **数据信号调制器 (DSM):** 数据信号调制器 (Data Signal Modulator, DSM) 允许用户将数字数据流 (调制器信号) 与载波信号混合以产生调制输出。

## 1.6 系列中各器件的详细说明

PIC24FJ128GB204 系列中的器件具有 28 引脚和 44 引脚两种封装形式。图 1-1 给出了所有器件的一般框图。

这些器件在以下 6 个方面存在差异：

1. 闪存程序存储器（对于 PIC24FJ64GB2XX 器件为 64 KB，对于 PIC24FJ128GB2XX 器件为 128 KB）。
2. 可用 I/O 引脚和端口数（对于 28 引脚器件为 2 个端口 21 个引脚，对于 44 引脚器件为 3 个端口 35 个引脚）。
3. 可用输入电平变化通知（ICN）输入数（对于 28 引脚器件为 20 个，对于 44 引脚器件为 34 个）。
4. 可用的可重映射引脚数（对于 28 引脚器件为 14 个引脚，对于 44 引脚器件为 24 个引脚）。
5. A/D 转换器的模拟输入通道数（对于 44 引脚器件为 12 路通道，对于 28 引脚器件为 9 路通道）。

本系列器件的所有其他功能都是相同的。表 1-1 和表 1-2 对此进行了总结。

表 1-3 给出了 PIC24FJ128GB204 系列器件上可用的引脚功能列表，按功能排序。注意，此表只显示了各个外设功能所使用的引脚单元，而没有显示同一引脚上的多种功能的复用方式。在本数据手册开始部分的引脚框图中提供了有关引脚复用的信息。复用的功能按功能的优先级排列，最前面的是优先级最高的外设功能。

# PIC24FJ128GB204 系列

表 1-1: PIC24FJ128GB204 系列器件的特性: 44 引脚器件

特性	PIC24FJ64GB204	PIC24FJ128GB204
工作频率	DC – 32 MHz	
程序存储器 (字节数)	64K	128K
程序存储器 (指令数)	22,016	44,032
数据存储器 (字节数)	8K	
中断源 (软向量 / NMI 陷阱)	72 (68/4)	
I/O 端口	端口 A、B 和 C	
I/O 引脚总数	34	
可重映射的引脚数	24 (23 个 I/O, 仅 1 个输入)	
定时器:		
总数 (16 位)	5 <sup>(1)</sup>	
32 位 (由一对 16 位定时器组成)	2	
带定时器的输入捕捉通道数	6 <sup>(1)</sup>	
输出比较 /PWM 通道数	6 <sup>(1)</sup>	
输入电平变化通知中断	34	
串行通信:		
UART	4 <sup>(1)</sup>	
SPI (3 线 /4 线)	3 <sup>(1)</sup>	
I <sup>2</sup> C™	2	
数字信号调制器 (DSM)	有	
并行通信 (EPMP/PSP)	有	
JTAG 边界扫描	有	
12 位 SAR 模数 (A/D) 转换器 (输入通道数)	12	
模拟比较器	3	
CTMU 接口	12 路通道	
复位 (和延时)	内核 POR、VDD POR、VBAT POR、BOR、RESET 指令、MCLR、WDT；非法操作码、REPEAT 指令、硬件陷阱和配置字失配 (OST 和 PLL 锁定)	
指令集	76 条基本指令和多种寻址模式	
封装	44 引脚 TQFP 和 QFN	
加密引擎	支持带 128、192 和 256 位密钥的 AES、DES 和 TDES、真随机数发生器、伪随机数发生器、片内 OTP 存储器	
USB	兼有 USB 全速和低速模式的 On-The-Go (OTG) USB	
RTCC	有	

注 1: 外设可通过可重映射的引脚访问。

# PIC24FJ128GB204 系列

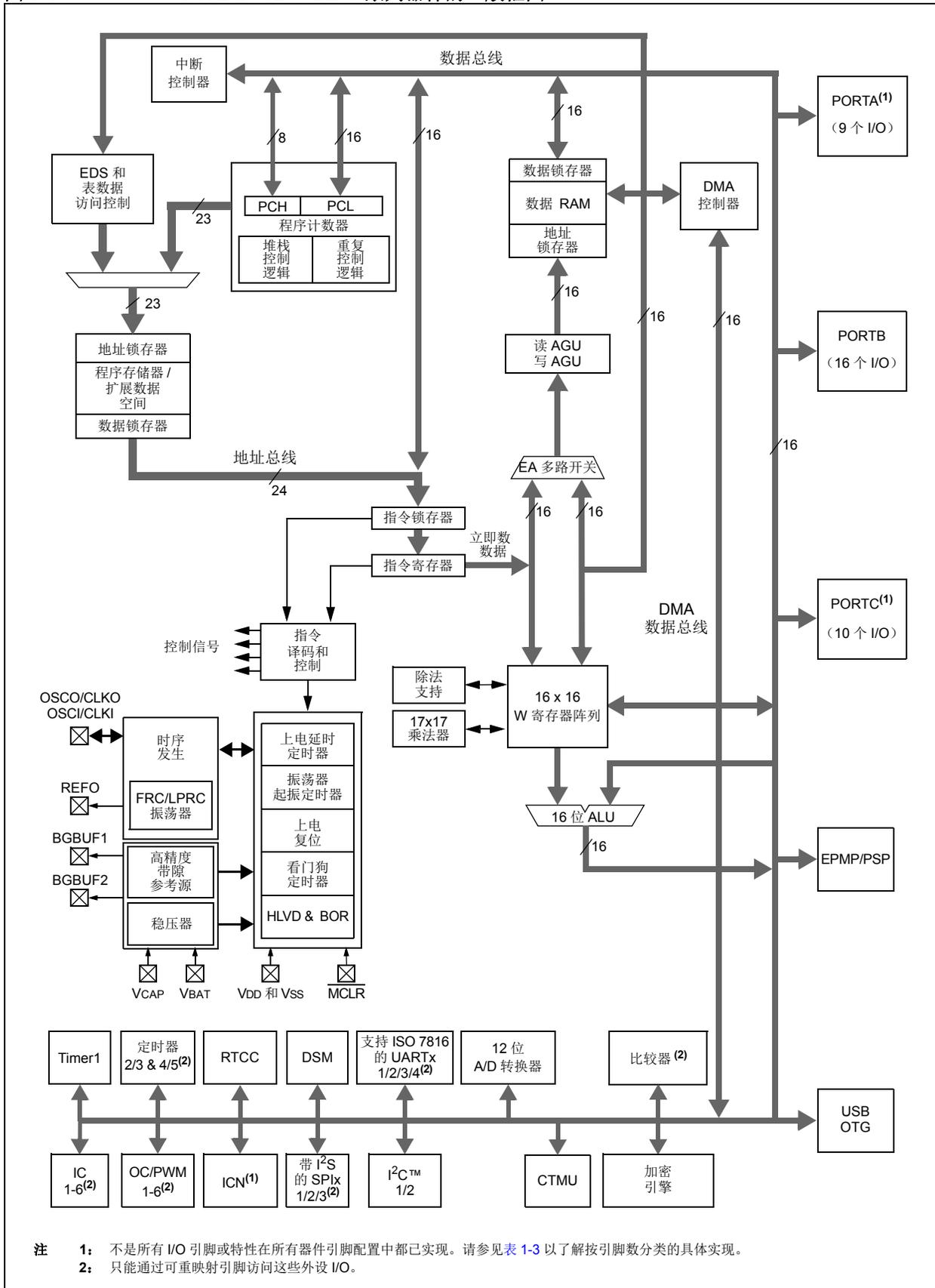
表 1-2: PIC24FJ128GB204 系列器件的特性: 28 引脚器件

特性	PIC24FJ64GB202	PIC24FJ128GB202
工作频率	DC – 32 MHz	
程序存储器 (字节数)	64K	128K
程序存储器 (指令数)	22,016	44,032
数据存储器 (字节数)	8K	
中断源 (软向量 / NMI 陷阱)	72 (68/4)	
I/O 端口	端口 A 和 B	
I/O 引脚总数	20	
可重映射的引脚数	15 (14 个 I/O, 仅 1 个输入)	
定时器:		
总数 (16 位)	5 <sup>(1)</sup>	
32 位 (由一对 16 位定时器组成)	2	
带定时器的输入捕捉通道数	6 <sup>(1)</sup>	
输出比较 /PWM 通道数	6 <sup>(1)</sup>	
输入电平变化通知中断	20	
串行通信:		
UART	4 <sup>(1)</sup>	
SPI (3 线 /4 线)	3 <sup>(1)</sup>	
I <sup>2</sup> C™	2	
数字信号调制器 (DSM)	有	
JTAG 边界扫描	有	
12 位 SAR 模数 (A/D) 转换器 (输入通道数)	9	
模拟比较器	3	
CTMU 接口	9 路通道	
复位 (和延时)	内核 POR、V <sub>DD</sub> POR、V <sub>BAT</sub> POR、BOR、RESET 指令、MCLR、WDT; 非法操作码、REPEAT 指令、硬件陷阱和配置字失配 (OST 和 PLL 锁定)	
指令集	76 条基本指令和多种寻址模式	
封装	28 引脚 SPDIP、SSOP、SOIC 和 QFN-S	
加密引擎	支持带 128、192 和 256 位密钥的 AES、DES 和 TDES、真随机数发生器、伪随机数发生器、片内 OTP 存储器	
USB	兼有 USB 全速和低速模式的 On-The-Go (OTG) USB	
RTCC	有	

注 1: 外设可通过可重映射的引脚访问。

# PIC24FJ128GB204 系列

图 1-1: PIC24FJ128GB204 系列器件的一般框图



注 1: 不是所有 I/O 引脚或特性在所有器件引脚配置中都已实现。请参见表 1-3 以了解按引脚数分类的具体实现。  
 2: 只能通过可重映射引脚访问这些外设 I/O。

# PIC24FJ128GB204 系列

表 1-3: PIC24FJ128GB204 系列引脚配置说明

引脚功能	引脚编号 / 栅格位置			I/O	输入 缓冲器	说明
	28 引脚 SPDIP/SOIC/ SSOP	28 引脚 QFN-S	44 引脚 TQFP/QFN			
AN0	2	27	19	I	ANA	12 位 SAR A/D 转换器输入。
AN1	3	28	20	I	ANA	
AN2	4	1	21	I	ANA	
AN3	5	2	22	I	ANA	
AN4	6	3	23	I	ANA	
AN5	7	4	24	I	ANA	
AN6	25	22	14	I	ANA	
AN7	24	21	11	I	ANA	
AN9	26	23	15	I	ANA	
AN10	—	—	25	I	ANA	
AN11	—	—	26	I	ANA	
AN12	—	—	27	I	ANA	
ASCL1	3	28	20	—	—	
ASDA1	2	27	19	—	—	
AVDD	—	—	17	P	ANA	模拟模块的正电源。
AVSS	—	24	16	P	ANA	模拟模块的参考地。
C1INA	7	4	24	I	ANA	比较器 1 的输入 A。
C1INB	6	3	23	I	ANA	比较器 1 的输入 B。
C1INC	24	15	1	I	ANA	比较器 1 的输入 C。
C1IND	9	6	30	I	ANA	比较器 1 的输入 D。
C2INA	5	2	22	I	ANA	比较器 2 的输入 A。
C2INB	4	1	21	I	ANA	比较器 2 的输入 B。
C2INC	18	15	1	I	ANA	比较器 2 的输入 C。
C2IND	10	7	31	I	ANA	比较器 2 的输入 D。
C3INA	26	23	15	I	ANA	比较器 3 的输入 A。
C3INB	25	22	14	I	ANA	比较器 3 的输入 B。
C3INC	2	15	1	I	ANA	比较器 3 的输入 C。
C3IND	3	28	20	I	ANA	比较器 3 的输入 D。
CLKI	9	6	30	I	ANA	主时钟输入连接。
CLKO	10	7	31	O	—	系统时钟输出。

图注: ST = 施密特触发器输入  
ANA = 模拟输入  
I<sup>2</sup>C = 带 I<sup>2</sup>C™ 或 SMBus 电平的 ST

TTL = TTL 兼容输入  
O = 输出

I = 输入  
P = 功耗

# PIC24FJ128GB204 系列

表 1-3: PIC24FJ128GB204 系列引脚配置说明 (续)

引脚功能	引脚编号 / 栅格位置			I/O	输入 缓冲器	说明	
	28 引脚 SPDIP/SOIC/ SSOP	28 引脚 QFN-S	44 引脚 TQFP/QFN				
CN0	12	9	34	—	—	电平变化中断输入。	
CN1	11	8	33	—	—		
CN2	2	27	19	—	—		
CN3	3	28	20	—	—		
CN4	4	1	21	—	—		
CN5	5	2	22	—	—		
CN6	6	3	23	—	—		
CN7	7	4	24	—	—		
CN8	—	—	25	—	—		
CN9	—	—	26	—	—		
CN10	—	—	27	—	—		
CN11	26	23	15	—	—		
CN12	25	22	14	—	—		
CN13	24	21	11	—	—		
CN15	22	19	9	—	—		
CN16	21	18	8	—	—		
CN17	—	—	3	—	—		
CN18	—	—	2	—	—		
CN19	—	—	5	—	—		
CN20	—	—	4	—	—		
CN21	18	15	1	—	—		
CN22	17	14	44	—	—		
CN23	16	13	43	—	—		
CN24	15	12	42	—	—		
CN25	—	—	37	—	—		
CN26	—	—	38	—	—		
CN27	14	11	41	—	—		
CN28	—	—	36	—	—		
CN29	10	7	31	—	—		
CN30	9	6	30	—	—		
CN33	—	—	13	—	—		
CN34	—	—	32	—	—		
CN35	—	—	35	—	—		
CN36	—	—	12	—	—		
CTCMP	4	1	21	I	ANA		CTMU 比较器 2 输入 (脉冲模式)。

图注: ST = 施密特触发器输入

ANA = 模拟输入

I<sup>2</sup>C = 带 I<sup>2</sup>C™ 或 SMBus 电平的 ST

TTL = TTL 兼容输入

O = 输出

I = 输入

P = 功耗

# PIC24FJ128GB204 系列

表 1-3: PIC24FJ128GB204 系列引脚配置说明 (续)

引脚功能	引脚编号 / 栅格位置			I/O	输入 缓冲器	说明
	28 引脚 SPDIP/SOIC/ SSOP	28 引脚 QFN-S	44 引脚 TQFP/QFN			
CTED1	2	27	19	I	ANA	CTMU 外部边沿输入。
CTED2	3	28	20	I	ANA	
CTED3	16	13	43	I	ANA	
CTED4	18	15	1	I	ANA	
CTED5	25	22	14	I	ANA	
CTED6	26	23	15	I	ANA	
CTED7	—	—	5	I	ANA	
CTED8	7	4	24	I	ANA	
CTED9	22	19	9	I	ANA	
CTED10	17	14	44	I	ANA	
CTED11	21	18	8	I	ANA	
CTED12	5	2	22	I	ANA	
CTED13	6	3	23	I	ANA	
CTPLS	24	21	11	O	—	CTMU 脉冲输出。
CVREF	25	22	14	O	ANA	比较器参考电压输出。
CVREF+	2	27	19	I	ANA	比较器参考电压 (高) 输入。
CVREF-	3	28	20	I	ANA	比较器参考电压 (低) 输入。
D+	21	18	8	I/O	—	USB 差分正信号线 (内部收发器)。
D-	22	19	9	I/O	—	USB 差分负信号线 (内部收发器)。
INT0	16	13	43	I	ST	外部中断输入 0。
HLVDIN	4	1	21	I	ANA	高 / 低压检测输入。
MCLR	1	26	18	I	ST	主复位 (器件复位) 输入。将此引脚拉为低电平可导致器件复位。
OSCI	9	6	30	I	ANA	主振荡器输入连接。
OSCO	10	7	31	O	—	主振荡器输出连接。
PGC1	5	2	22	I/O	ST	在线调试器 / 仿真器 / ICSP™ 编程时钟。
PGC2	22	19	9	I/O	ST	
PGC3	3	28	20	I/O	ST	
PGD1	4	1	21	I/O	ST	
PGD2	21	18	8	I/O	ST	
PGD3	2	27	19	I/O	ST	

图注: ST = 施密特触发器输入

ANA = 模拟输入

I<sup>2</sup>C = 带 I<sup>2</sup>C™ 或 SMBus 电平的 ST

TTL = TTL 兼容输入

O = 输出

I = 输入

P = 功耗

# PIC24FJ128GB204 系列

表 1-3: PIC24FJ128GB204 系列引脚配置说明 (续)

引脚功能	引脚编号 / 栅格位置			I/O	输入 缓冲器	说明
	28 引脚 SPDIP/SOIC/ SSOP	28 引脚 QFN-S	44 引脚 TQFP/QFN			
PMA0/PMALL	—	—	3	O	—	并行主端口地址。
PMA1/PMALH	—	—	2	O	—	
PMA14/PMCS/ PMCS1	—	—	15	O	—	
PMA2/PMALU	—	—	12	O	—	
PMA3	—	—	38	O	—	
PMA4	—	—	37	O	—	
PMA5	—	—	4	O	—	
PMA6	—	—	5	O	—	
PMA7	—	—	13	O	—	
PMA8	—	—	32	O	—	
PMA9	—	—	35	O	—	
PMACK1	—	—	27	I	ST/TTL	并行主端口应答输入 1。
PMBE0	—	—	36	O	—	并行主端口字节使能 0 选通。
PMBE1	—	—	25	O	—	并行主端口字节使能 1 选通。
PMCS1	—	—	30	I/O	ST/TTL	并行主端口片选 1 选通。
PMD0	—	—	21	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	—	—	22	I/O	ST/TTL	
PMD2	—	—	23	I/O	ST/TTL	
PMD3	—	—	1	I/O	ST/TTL	
PMD4	—	—	44	I/O	ST/TTL	
PMD5	—	—	43	I/O	ST/TTL	
PMD6	—	—	20	I/O	ST/TTL	
PMD7	—	—	19	I/O	ST/TTL	
PMRD	—	—	11	O	—	并行主端口读选通。
PMWR	—	—	24	O	—	并行主端口写选通。
RA0	2	27	19	I/O	ST	PORTA 数字 I/O。
RA1	3	28	20	I/O	ST	
RA2	9	6	30	I/O	ST	
RA3	10	7	31	I/O	ST	
RA4	12	9	34	I	ST	
RA7	—	—	13	I/O	ST	
RA8	—	—	32	I/O	ST	
RA9	—	—	35	I/O	ST	
RA10	—	—	12	I/O	ST	

图注: ST = 施密特触发器输入      TTL = TTL 兼容输入      I = 输入  
ANA = 模拟输入                      O = 输出                                  P = 功耗  
I<sup>2</sup>C = 带 I<sup>2</sup>C™ 或 SMBus 电平的 ST

# PIC24FJ128GB204 系列

表 1-3: PIC24FJ128GB204 系列引脚配置说明 (续)

引脚功能	引脚编号 / 栅格位置			I/O	输入 缓冲器	说明
	28 引脚 SPDIP/SOIC/ SSOP	28 引脚 QFN-S	44 引脚 TQFP/QFN			
RB0	4	1	21	I/O	ST	PORTB 数字 I/O。
RB1	5	2	22	I/O	ST	
RB2	6	3	23	I/O	ST	
RB3	7	4	24	I/O	ST	
RB4	11	8	33	I	ST	
RB5	14	11	41	I/O	ST	
RB6	15	12	42	I/O	ST	
RB7	16	13	43	I/O	ST	
RB8	17	14	44	I/O	ST	
RB9	18	15	1	I/O	ST	
RB10	21	18	8	I/O	ST	
RB11	22	19	9	I/O	ST	
RB13	24	21	11	I/O	ST	
RB14	25	22	14	I/O	ST	
RB15	26	23	15	I/O	ST	
RC0	—	—	25	I/O	ST	PORTC 数字 I/O。
RC1	—	—	26	I/O	ST	
RC2	—	—	27	I/O	ST	
RC3	—	—	36	I/O	ST	
RC4	—	—	37	I/O	ST	
RC5	—	—	38	I/O	ST	
RC6	—	—	2	I/O	ST	
RC7	—	—	3	I/O	ST	
RC8	—	—	4	I/O	ST	
RC9	—	—	5	I/O	ST	
REFI	22	19	9	—	—	参考时钟输入。
REFO	24	21	11	—	—	参考时钟输出。

图注: ST = 施密特触发器输入      TTL = TTL 兼容输入      I = 输入  
ANA = 模拟输入                    O = 输出                                P = 功耗  
I<sup>2</sup>C = 带 I<sup>2</sup>C™ 或 SMBus 电平的 ST

# PIC24FJ128GB204 系列

表 1-3: PIC24FJ128GB204 系列引脚配置说明 (续)

引脚功能	引脚编号 / 栅格位置			I/O	输入 缓冲器	说明
	28 引脚 SPDIP/SOIC/ SSOP	28 引脚 QFN-S	44 引脚 TQFP/QFN			
RP0	4	1	21	I/O	ST	可重映射外设 (输入或输出)。
RP1	5	2	22	I/O	ST	
RP2	6	3	23	I/O	ST	
RP3	7	4	24	I/O	ST	
RP5	2	27	19	I/O	ST	
RP6	3,15	28	20	I/O	ST	
RP7	16	13	43	I/O	ST	
RP8	17	14	44	I/O	ST	
RP9	18	15	1	I/O	ST	
RP10	21	18	8	I/O	ST	
RP11	22	19	9	I/O	ST	
RP13	24	21	11	I/O	ST	
RP14	25	22	14	I/O	ST	
RP15	26	23	15	I/O	ST	
RP16	—	—	25	I/O	ST	
RP17	—	—	26	I/O	ST	
RP18	—	—	27	I/O	ST	
RP19	—	—	36	I/O	ST	
RP20	—	—	37	I/O	ST	
RP21	—	—	38	I/O	ST	
RP22	—	—	2	I/O	ST	
RP23	—	—	3	I/O	ST	
RP24	—	—	4	I/O	ST	
RP25	—	—	5	I/O	ST	
RPI4	11	8	33	I	ST	
RTCC	25	22	14	O	—	实时时钟闹钟 / 秒脉冲输出。
SCL1	17	14	44	I/O	I <sup>2</sup> C	I2C1 同步串行时钟输入 / 输出。
SCL2	7	4	24	I/O	I <sup>2</sup> C	I2C2 同步串行时钟输入 / 输出。
SCLKI	12	9	34	I	—	辅助振荡器数字时钟输入。
SDA1	18	15	1	I/O	I <sup>2</sup> C	I2C1 数据输入 / 输出。
SDA2	6	3	23	I/O	I <sup>2</sup> C	I2C2 数据输入 / 输出。
SOSCI	11	8	33	I	ANA	辅助振荡器 / Timer1 时钟输入。
SOSCO	12	9	34	O	ANA	辅助振荡器 / Timer1 时钟输出。

图注: ST = 施密特触发器输入      TTL = TTL 兼容输入      I = 输入  
ANA = 模拟输入      O = 输出      P = 功耗  
I<sup>2</sup>C = 带 I<sup>2</sup>C™ 或 SMBus 电平的 ST

# PIC24FJ128GB204 系列

表 1-3: PIC24FJ128GB204 系列引脚配置说明 (续)

引脚功能	引脚编号 / 栅格位置			I/O	输入 缓冲器	说明
	28 引脚 SPDIP/SOIC/ SSOP	28 引脚 QFN-S	44 引脚 TQFP/QFN			
T1CK	18	15	1	I	ST	Timer1 时钟。
T2CK	26	23	15	I	ST	Timer2 时钟。
T3CK	26	23	15	I	ST	Timer3 时钟。
T4CK	6	3	23	I	ST	Timer4 时钟。
T5CK	6	3	23	I	ST	Timer5 时钟。
TCK	17	14	13	I	ST	JTAG 测试时钟 / 编程时钟输入。
TDI	16	13	35	I	ST	JTAG 测试数据 / 编程数据输入。
TDO	18	15	32	O	—	JTAG 测试数据输出。
TMS	14	11	12	I	—	JTAG 测试模式选择输入。
USBID	14	11	41	I	ST	USB OTG ID (仅 OTG 模式)。
USBOEN	17	14	44	O	—	USB 输出使能控制 (用于外部收发器)。
VBAT	19	16	6	P	—	备用电池 (B+) 输入 (标称值为 1.2V)。
VBUS	15	12	42	P	—	USB 电压, 主机模式 (5V)。
VCAP	20	17	7	P	—	外部滤波电容连接。
VDD	13,28	25	28,40	P	—	外设数字逻辑和 I/O 引脚的正电源。
VDDCORE	20	17	7	—	—	单片机内核电源。
VREF+	2	27	19	I	ANA	A/D 参考电压输入 (+)。
VREF-	3	28	20	I	ANA	A/D 参考电压输入 (-)。
VSS	8,27	5,24	29,39	P	—	逻辑和 I/O 引脚的参考地。
VUSB3V3	23	20	10	P	—	USB 收发器的电源输入电压 (标称值为 3.3V)。

图注: ST = 施密特触发器输入  
 ANA = 模拟输入  
 I<sup>2</sup>C = 带 I<sup>2</sup>C™ 或 SMBus 电平的 ST

TTL = TTL 兼容输入  
 O = 输出

I = 输入  
 P = 功耗

# PIC24FJ128GB204 系列

---

注:

## 2.0 16 位单片机入门指南

### 2.1 基本连接要求

在开始使用 PIC24FJ128GB204 系列 16 位单片机进行开发之前，必须至少连接一些器件引脚。

必须始终连接以下引脚：

- 所有 VDD 和 VSS 引脚（见第 2.2 节“电源引脚”）
- 所有 AVDD 和 AVSS 引脚（不论是否使用模拟器件功能）（见第 2.2 节“电源引脚”）
- MCLR 引脚（见第 2.3 节“主复位（MCLR）引脚”）
- ENVREG/DISVREG 和 VCAP/VDDCORE 引脚（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，则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 OSCI 和 OSCO 引脚（见第 2.6 节“外部振荡器引脚”）

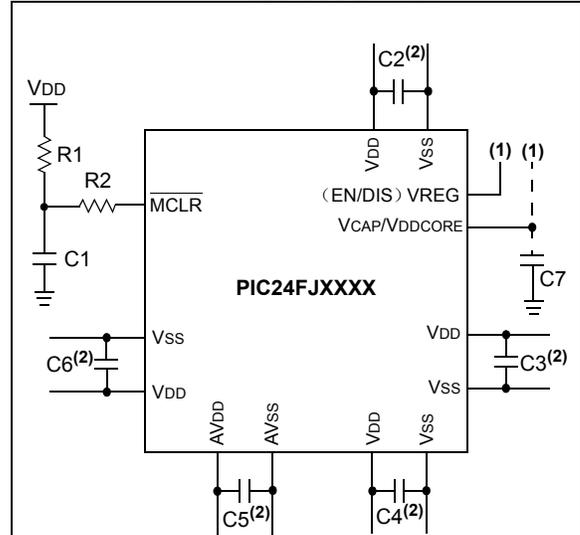
此外，可能还需要连接以下引脚：

- 实现模拟模块的外部参考电压时使用的 VREF+/VREF- 引脚

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

图 2-1 中显示了必需的最少连接。

图 2-1: 建议的最少连接



**要点（所有值都是建议值）：**

C1 至 C6: 0.1  $\mu$ F、20V 陶瓷电容

C7: 10  $\mu$ F、6.3V 或更大的钽电容或陶瓷电容

R1: 10 k $\Omega$

R2: 100 $\Omega$  至 470 $\Omega$

- 注**
- 1: 请参见第 2.4 节“稳压器引脚（ENVREG/DISVREG 和 VCAP/VDDCORE）”了解 ENVREG/DISVREG 引脚连接的说明。
  - 2: 此处所示示例针对的是具有 5 个 VDD/VSS 和 AVDD/AVSS 引脚对的 PIC24F 器件。其他器件所具有的引脚对可能有所不同，请相应地调整去耦电容的数量。

# PIC24FJ128GB204 系列

## 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，需要考虑以下条件：

- **电容的值和类型：**建议值为 0.1  $\mu\text{F}$ （100 nF）、10-20V 的电容。此电容应为低 ESR 电容且谐振频率在 200 MHz 及更高范围内。建议使用陶瓷电容。
- **印刷电路板上的位置：**去耦电容应尽可能靠近引脚放置。建议将电容放在电路板上器件所在的一侧。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：**如果电路板上存在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。该辅助电容值的范围为 0.001  $\mu\text{F}$  至 0.01  $\mu\text{F}$ 。请将这个辅助电容挨着主去耦电容放置。在高速电路设计中，应考虑尽可能在靠近电源和地引脚的位置放置一对容值相差数十倍的电容（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）。
- **性能最大化：**从电源电路开始布置电路板的走线时，请首先布置电源线并把线返回到去耦电容，然后再走线到器件引脚。这可确保去耦电容在电源链中处于第一位置。保持电容和电源引脚之间的走线长度尽可能短也同样重要，因为这可以减少 PCB 走线间的互感。

### 2.2.2 槽路电容

对于电源走线长度大于 6 英寸的电路板，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  $\mu\text{F}$  至 47  $\mu\text{F}$ 。

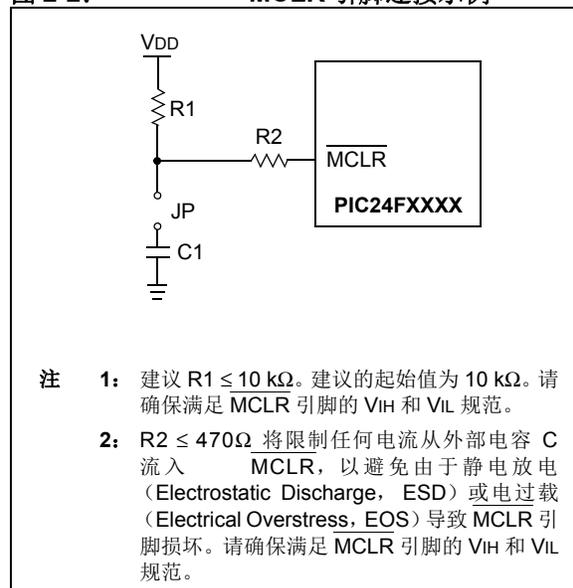
## 2.3 主复位 ( $\overline{\text{MCLR}}$ ) 引脚

$\overline{\text{MCLR}}$  引脚提供了两个特殊的器件功能：器件复位及器件编程与调试。如果最终应用中不需要进行编程和调试，则只需直接连接 VDD 即可。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 给出了一种典型配置。根据应用的需求，还可以实现其他电路设计。

在编程和调试过程中，必须考虑可添加到该引脚的电阻和电容。器件编程器和调试器会驱动  $\overline{\text{MCLR}}$  引脚。因此，不能对特定的电压电平 ( $V_{IH}$  和  $V_{IL}$ ) 和快速信号跳变造成不利影响。为此需要根据应用和 PCB 要求调整 R1 和 C1 的具体值。例如，在编程和调试操作期间，建议使用跳线将电容 C1 与  $\overline{\text{MCLR}}$  引脚隔离（图 2-2）。正常的运行时操作不需要这样的跳线。

与  $\overline{\text{MCLR}}$  引脚关联的所有元件都应放置在距离该引脚 0.25 英寸（6 mm）的范围内。

图 2-2:  $\overline{\text{MCLR}}$  引脚连接示例



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

**注：** 本节仅适用于具有片内稳压器的 PIC24FJ 器件。

片内稳压器使能/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，将其连接到 VDD 会使能稳压器，接地会禁止稳压器
- 对于 DISVREG，将其接地会使能稳压器，连接到 VDD 会禁止稳压器

关于连接和使用片内稳压器的详细信息，请参见第 30.2 节“片内稳压器”。

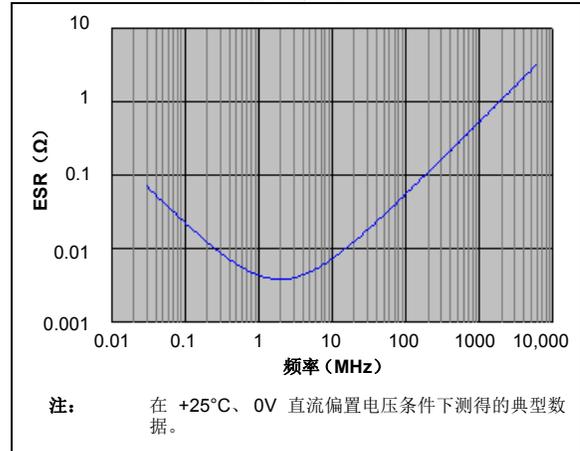
使能稳压器后，VCAP/VDDCORE 引脚上需要连接一个低 ESR ( $< 5\Omega$ ) 的电容，来稳定稳压器输出电压。VCAP/VDDCORE 引脚不得连接到 VDD，而是必须通过使用一个  $10\ \mu\text{F}$  的电容接地。电容类型可以是陶瓷电容或者钽电容。表 2-1 列举了一些可采用的电容。也可以使用具有同等规格的其他电容。

设计人员可以使用图 2-3 来计算备选电容的 ESR。

此电容应靠近 VCAP/VDDCORE 放置。建议走线长度不要超出 0.25 英寸（6 mm）。更多信息，请参见第 33.0 节“电气特性”。

禁止稳压器时，VCAP/VDDCORE 引脚必须与 VDDCORE 级别的电源相连。关于 VDD 和 VDDCORE 的信息，请参见第 33.0 节“电气特性”。

**图 2-3：** 建议 VCAP 下频率与 ESR 性能的关系



**表 2-1：** 适用的同等规格电容

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

# PIC24FJ128GB204 系列

## 2.4.1 采用陶瓷电容时的注意事项

近年来，大容量、低电压的表面贴装式陶瓷电容极具成本效益，其容值最高可达数十微法。低 ESR、较小的物理尺寸和其他特性使得陶瓷电容在许多类型的应用中非常抢眼。

陶瓷电容适用于本系列单片机的内部稳压器。但是，选择电容时需谨慎，应确保在应用的目标工作范围内维持足够的电容。

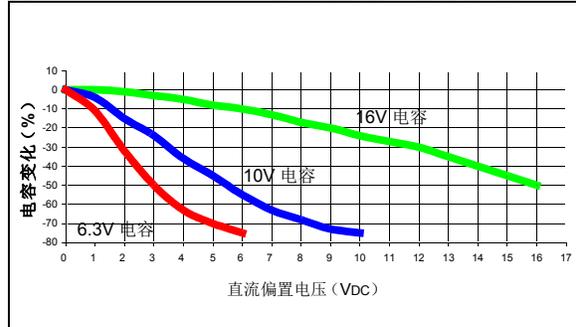
典型值为  $10\ \mu\text{F}$  的低成本陶瓷电容具有 X5R、X7R 和 Y5V 这三种介电特性（也可使用其他类型，但是不太常见）。这些类型电容的初始公差规范通常指定为  $\pm 10\%$  到  $\pm 20\%$ （X5R 和 X7R），或者  $-20\%/+80\%$ （Y5V）。然而，这些电容在应用电路中提供的有效电容还将根据其其他因素（如施加的直流偏置电压和温度）而变化。因此，电容在电路中的总公差要比初始公差规范宽的多。

X5R 和 X7R 电容通常显示出令人满意的温度稳定性（例如：在宽温度范围内为  $\pm 15\%$ ，具体规范请查询制造商的数据手册）。但是，Y5V 电容在温度变化时的电容变化范围通常很糟糕，为  $+22\%/-82\%$ 。因此，标称值为  $10\ \mu\text{F}$  的 Y5V 类型电容可能无法提供足够的总电容，无法满足内部稳压器稳定性和瞬态响应要求。因此，如果应用必须在宽温度范围内工作，不建议将 Y5V 电容用于内部稳压器。

除了温度的影响外，电容值较大的陶瓷电容的有效电容还可能根据施加到电容的直流电压量不同而有较大的变化。这一影响可能非常大，但却往往被人们忽视，或常常在文档中找不到相关说明。

典型的直流偏置电压与 X7R 类型电容的电容值关系如图 2-4 所示。

图 2-4: 直流偏置电压与电容特性



选择要用于内部稳压器的陶瓷电容时，建议选择额定电压较高的电容，以使工作电压占电容最大电压的一小部分。例如，对于 2.5V 或 1.8V 的内核电压应选择额定值为 16V 的陶瓷电容。建议的电容在表 2-1 中给出。

## 2.5 ICSP 引脚

PGECx 和 PGEDx 引脚用于在线串行编程（ICSP）和调试。建议尽可能减小 ICSP 连接器与器件 ICSP 引脚之间的走线长度。如果预期 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  $100\ \Omega$ 。

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。如果应用需要此类分立元件，则在编程和调试期间应将它们从电路中去除。或者，参阅相关器件闪存编程规范中的直流 / 交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输入高电压（VIH）和输入低电压（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。

关于可用的 Microchip 开发工具连接要求的更多信息，请参见第 31.0 节“开发支持”。

## 2.6 外部振荡器引脚

许多单片机都有至少两个振荡器可供选择：高频主振荡器和低频辅助振荡器（详情请参见第 9.0 节“振荡器配置”）。

振荡器电路应放在电路板上器件所在的一侧。而且，振荡器电路应靠近相应振荡器引脚放置，且电路元件与引脚之间的距离不要超出 0.5 英寸（12 mm）。负载电容应在电路板的同一侧挨着振荡器放置。

应在振荡器电路周围布置接地的灌铜区，将其与周围电路隔离。接地的灌铜区应直接连接到 MCU 地。不要在接地的灌铜区内使用任何信号线或电源线。而且，如果使用双面电路板，请避免在放置晶振的电路板背面走线。

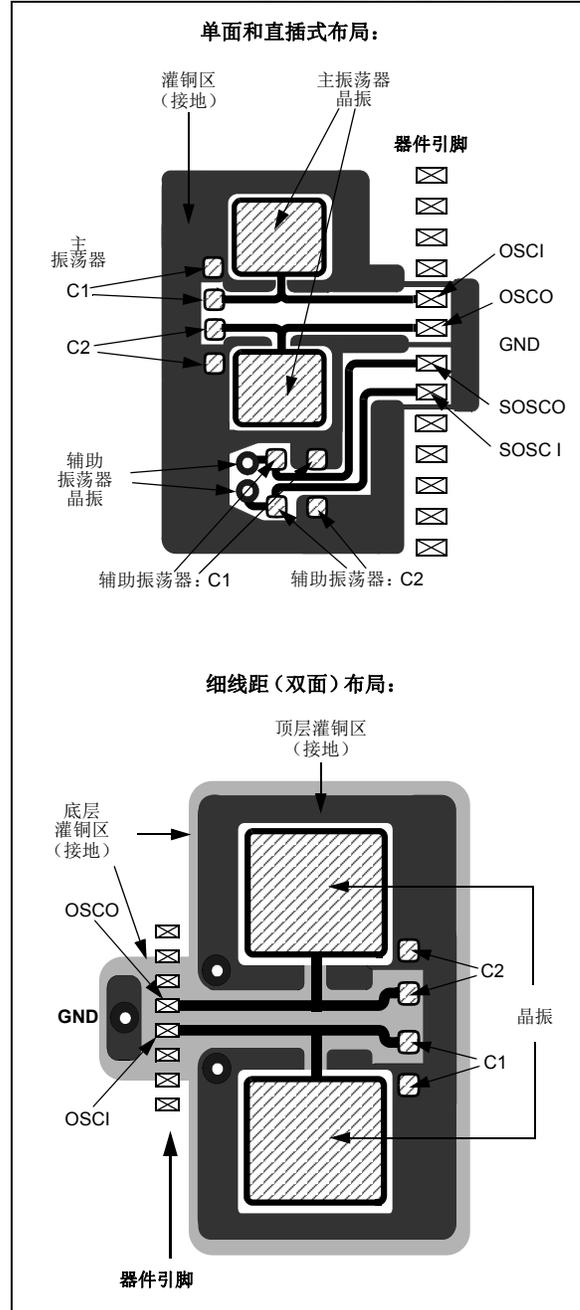
图 2-5 给出了建议的电路板布局。可以使用包含完整振荡器引脚的单面布局来处理直插式封装器件。对于引脚排列紧密的器件，单面布局则可能无法始终完全包含所有的引脚和元件。一种适合的解决方案是将含有保护走线的部分连接到反面的接地层。在所有情形中，保护走线都必须返回到地。

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，请参见 Microchip 公司网站 ([www.microchip.com](http://www.microchip.com)) 上提供的以下应用笔记：

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

图 2-5: 建议的振荡器电路布局



# PIC24FJ128GB204 系列

---

## 2.7 ICSP 操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 ADxPCFG 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 ADxPCFG 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 11.2 节“配置模拟端口引脚（ANSx）”。

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 A/D 模块初始化期间修改相应位，具体如下：

- 对于具有 ADxPCFG 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 相对应的位。
- 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 相对应的位。

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 ADxPCFG 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 引脚被确认为模拟输入引脚，以致端口值被读为逻辑 0，从而可能影响用户应用的功能性。

## 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，通过一个 1 kΩ 至 10 kΩ 的电阻将未使用的引脚连接到 Vss，并驱动输出为逻辑低电平。

## 3.0 CPU

**注：** 本数据手册总结了该 PIC24F 系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“具有扩展数据空间（EDS）的 CPU”（DS39732）。本数据手册中的信息取代了 FRM 中的信息。

PIC24F CPU 采用 16 位（数据）的改进型哈佛架构，具有增强的指令集和带有长度可变的操作码字段的 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 指令可以支持无开销的程序循环结构，该指令在任何时候都可被中断。

PIC24F 器件的编程模型中有 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（Software Stack Pointer, SSP）工作，用于中断和调用。

可对数据空间（Data Space, DS）的低 32 KB 进行线性访问。数据空间的高 32 KB 是指扩展数据 RAM、EPMP 存储空间或程序存储器可映射到的扩展数据空间。

指令集架构（Instruction Set Architecture, ISA）与 PIC18 的相比有了显著的增强，但仍保持了一定程度的向后兼容性。该架构直接或通过简单的宏支持所有的 PIC18 指令和寻址模式。对编译器执行效率的需求促使了对 ISA 的许多改进。

内核支持固有（无操作数）寻址、相对寻址、立即数寻址、存储器直接寻址及其他三组寻址模式。所有模式都支持寄存器直接寻址和各种寄存器间接寻址模式。每组都提供了最多 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。

对于大多数指令，在每个指令周期内，内核能执行一次数据（或程序数据）存储器读操作、一次工作寄存器（数据）读操作、一次数据存储器写操作和一次程序（指令）存储器读操作。因此，可以支持 3 个操作数的指令，使 3 操作数的运算（即， $A + B = C$ ）能在单周期内执行。

包含了一个高速 17 位 x 17 位乘法器，显著提高了内核的运算能力和吞吐量。此乘法器支持有符号、无符号和混合模式的 16 位 x 16 位或 8 位 x 8 位整数乘法。所有的乘法指令都在单周期内执行。

已对 16 位 ALU 进行了改进使其具备一个支持整数除法的硬件，该硬件支持迭代的不可撤销的除法算法。它可以和 REPEAT 指令循环机制和迭代除法指令（可选）一起工作，支持 32 位（或 16 位）除以 16 位有符号和无符号整数的除法运算。所有除法运算都需要 19 个周期来完成，但是可在任何周期边界被中断。

PIC24F 具有向量异常机制，具有最多 8 个不可屏蔽的陷阱源和 118 个中断源。可以为每个中断源分配 7 个优先级之一。

CPU 的框图如图 3-1 所示。

### 3.1 编程模型

图 3-2 中所示为 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，并且可以由指令直接控制。

表 3-1 对各寄存器进行了描述。所有与编程模型相关联的寄存器都是存储器映射的。

# PIC24FJ128GB204 系列

图 3-1: PIC24F CPU 内核框图

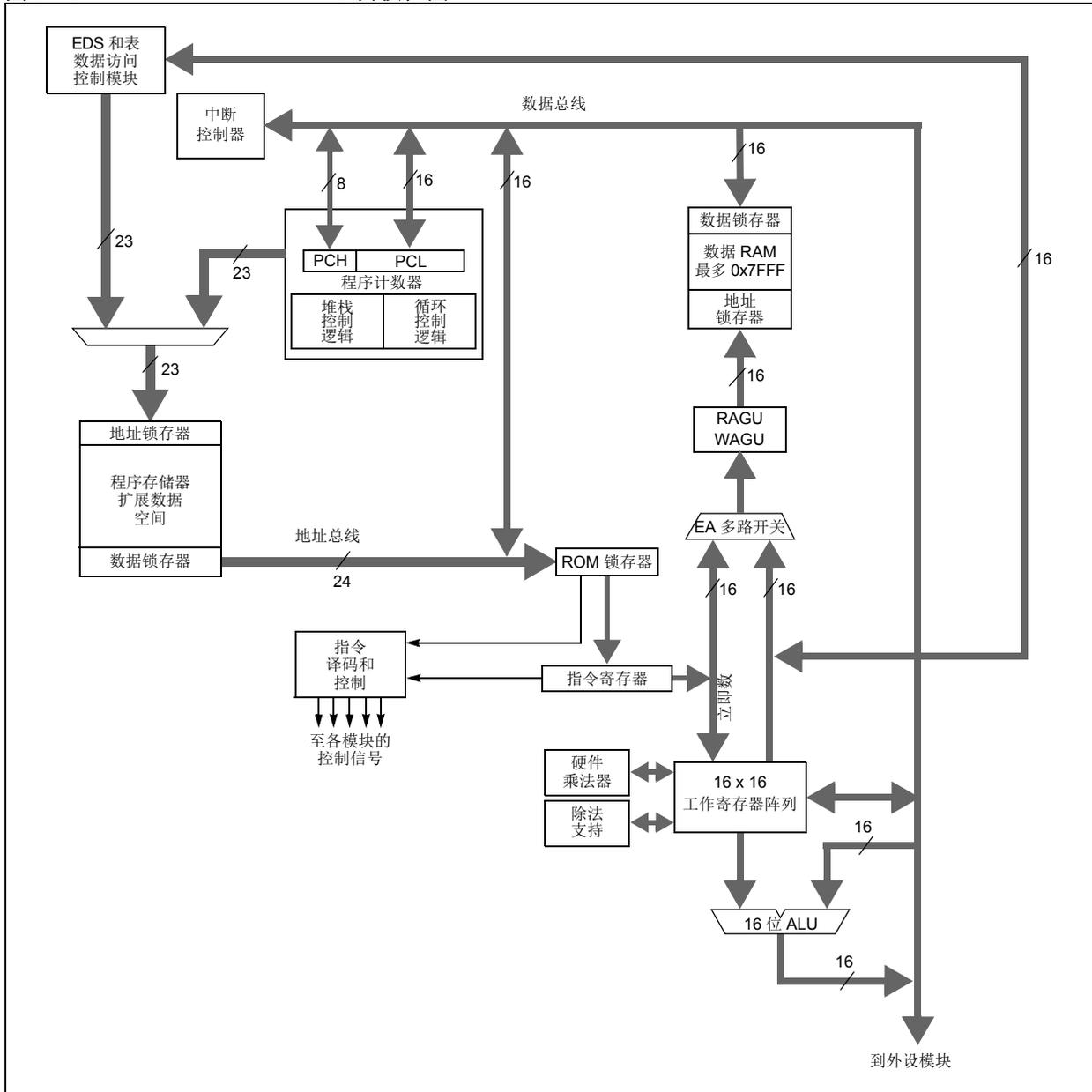
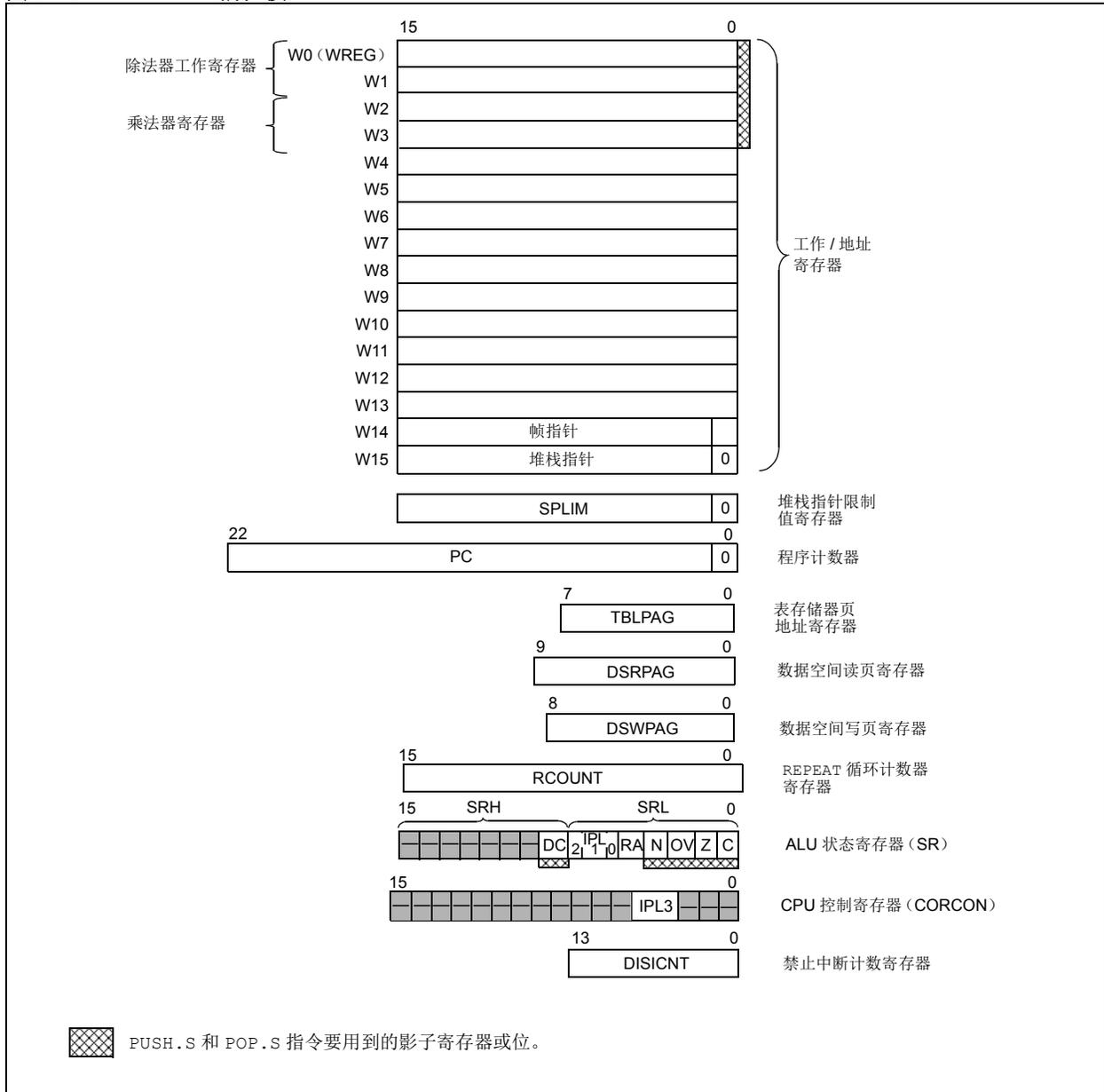


表 3-1: CPU 内核寄存器

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
RCOUNT	REPEAT 循环计数器寄存器
CORCON	CPU 控制寄存器
DISICNT	禁止中断计数寄存器
DSRPAG	数据空间读页寄存器
DSWPAG	数据空间写页寄存器

图 3-2: 编程模型



# PIC24FJ128GB204 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 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2 <sup>(2)</sup>	IPL1 <sup>(2)</sup>	IPL0 <sup>(2)</sup>	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) 发生了向高位的进位  
 0 = 结果的第 4 个低位或第 8 个低位未发生向高位的进位
- bit 7-5      **IPL<2:0>:** CPU 中断优先级状态位 <sup>(1,2)</sup>  
 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 = 未进行 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 = 在本次算术运算中有符号 (二进制补码) 运算发生了溢出  
 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经将该位清零 (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 (Most Significant bit, MSb) 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, IPLx 状态位是只读位。  
 2: IPLx 状态位与 IPL3 状态位 (CORCON<3>) 共同决定 CPU 的中断优先级 (Interrupt Priority Level, IPL)。如果 IPL3 = 1, 则括号中的值表示 IPL。

# PIC24FJ128GB204 系列

寄存器 3-2: CORCON: CPU 内核控制寄存器

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-1	U-0	U-0
—	—	—	—	IPL3 <sup>(1)</sup>	—	—	—
bit 7						bit 0	

<b>图注:</b>	C = 可清零位	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 15-4      **未实现:** 读为 0
- bit 3        **IPL3:** CPU 中断优先级状态位 <sup>(1)</sup>
  - 1 = CPU 中断优先级大于 7
  - 0 = CPU 中断优先级为 7 或更小
- bit 2        **保留:** 读为 1
- bit 1-0     **未实现:** 读为 0

**注 1:** IPL3 位与 IPL<2:0> 位 (SR<7:5>) 共同决定 CPU 中断优先级; 请参见寄存器 3-1 了解位说明。

# PIC24FJ128GB204 系列

## 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。除非另外声明，否则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和半借位。

根据所使用的指令模式，ALU 可以执行 8 位或 16 位运算。依据指令的寻址模式，ALU 运算的数据可以来自 W 寄存器阵列或数据存储单元。同样，ALU 的输出数据可以被写入 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。它带有专用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 包含一个高速的 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：

- 16 位 x 16 位有符号数
- 16 位 x 16 位无符号数
- 16 位有符号数 x 5 位 (立即数) 无符号数
- 16 位无符号数 x 16 位无符号数
- 16 位无符号数 x 5 位 (立即数) 无符号数
- 16 位无符号数 x 16 位有符号数
- 8 位无符号数 x 8 位无符号数

### 3.3.2 除法器

除法模块支持以下数据长度的 32 位 /16 位和 16 位 /16 位有符号和无符号整数除法运算：

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

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 W 寄存器 (对齐的) 对 (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。

### 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操作。由一个移位寄存器模块实现多位移位，在单个周期内最多可将数据算术右移或左移 15 位。所有的多位移位指令仅支持源操作数和目标结果的寄存器直接寻址模式。

在下面的表 3-2 中汇总了所有使用移位操作的指令。

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一位或多位。
SL	将源寄存器左移一位或多位。
LSR	将源寄存器逻辑右移一位或多位。



# PIC24FJ128GB204 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为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。低位字的地址总是偶数，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。这种对齐方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 000000h 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开始处。用户可在 000000h 地址编写一条 GOTO 指令以将代码的实际起始地址定义为 000002h。

PIC24F 器件也具有 2 个中断向量表（Interrupt Vector Table, IVT），地址范围分别为 000004h 至 0000FFh 和 000100h 至 0001FFh。这两个中断向量表允许使用不同的 ISR 处理每个器件中断源。第 8.1 节“中断向量表”提供了有关中断向量表的更多详细信息。

## 4.1.3 闪存配置字

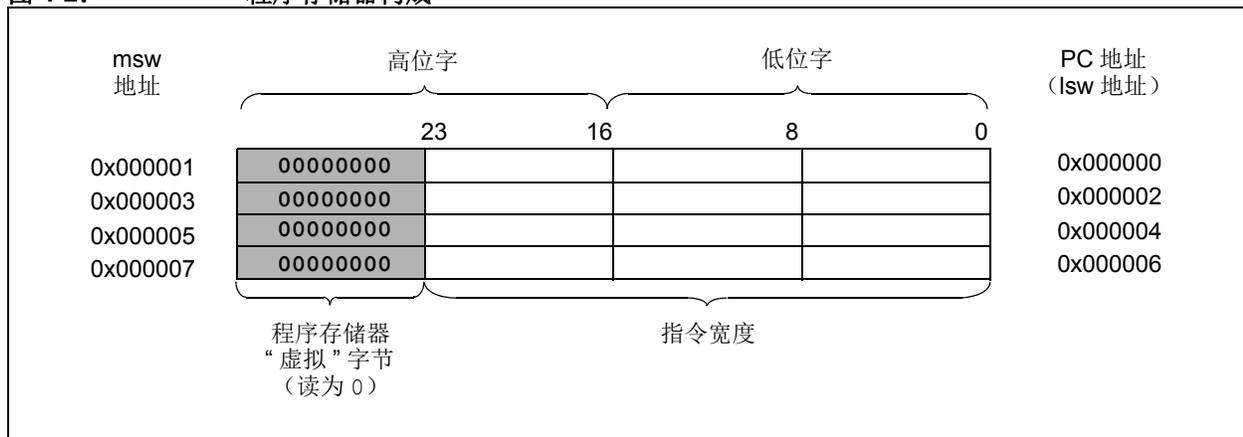
在 PIC24FJ128GB204 系列器件中，片上程序存储器的高 4 个字是保留的，用于存储配置信息。器件复位时，将配置信息复制到相应的配置寄存器中。PIC24FJ128GB204 系列器件的闪存配置字的地址如表 4-1 所示。图 4-1 给出了闪存配置字以及其他存储器向量在存储器映射中的位置。

程序存储器中的配置字为紧凑的格式。实际配置位被映射到配置存储空间的几个不同的寄存器中。它们在闪存配置字中的顺序并不反映它们在配置空间中的相应顺序。第 30.0 节“特殊功能”给出了器件配置字的更多详细信息。

表 4-1: PIC24FJ128GB204 系列器件的闪存配置字

器件	程序存储器 (字)	配置字地址
PIC24FJ64GB2XX	22,016	00ABF8h:00ABFEh
PIC24FJ128GB2XX	44,032	0157F8h:0157FEh

图 4-2: 程序存储器构成



## 4.2 数据存储空间

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“具有扩展数据空间（EDS）的数据存储器”（DS39733）。本数据手册中的信息取代了FRM中的信息。

PIC24F 内核具有一个 16 位宽的数据存储空间，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（DS）执行读写操作。图 4-3 给出了数据存储空间映射。

数据存储空间中的 16 位宽数据地址指向数据空间中的字节。这种构成方式使得数据空间的地址范围为 64 KB 或 32 K 字。低半部分（0000h 至 7FFFh）用于实现的（片上）存储器地址。

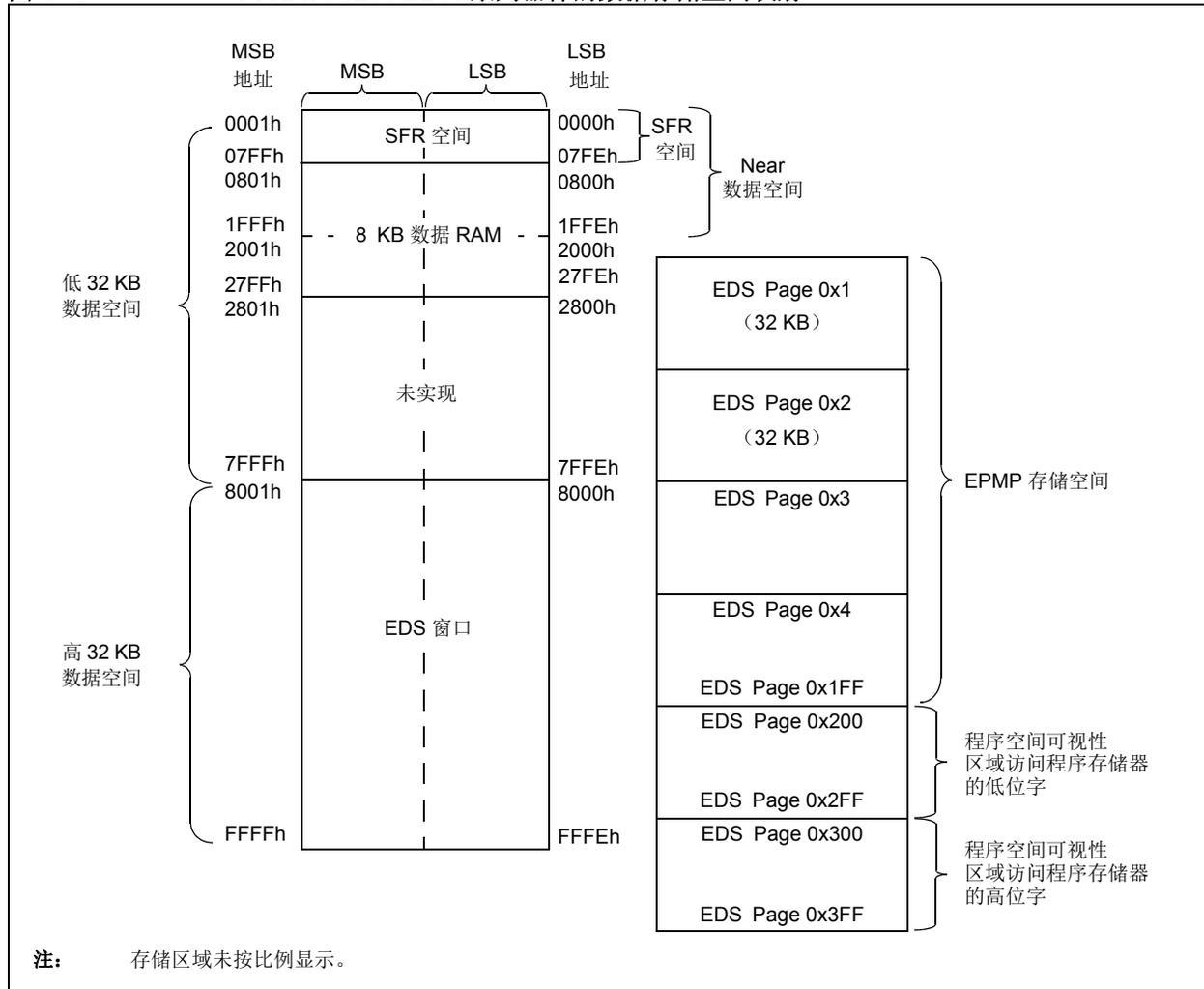
数据存储地址空间的高半部分（8000h 至 FFFFh）用作一个扩展数据空间（Extended Data Space, EDS）窗口。这允许单片机直接访问标准 16 位地址范围之外的较大范围的数据。在第 4.2.5 节“扩展数据空间（EDS）”中对 EDS 进行了详细的说明。

DS 的低半部分与不具有 EDS 的先前 PIC24F 单片机兼容。所有 PIC24FJ128GB204 系列器件在 DS 的低半部分（0800h 至 27FFh）实现 8 KB 的数据 RAM。

### 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 and 寄存器中的数据是以 16 位字为单位对齐的，但所有数据空间的有效地址（Effective Address, EA）都被解析为字节。每个字的低字节（Least Significant Byte, LSB）部分为偶地址，而高字节（Most Significant Byte, MSB）部分则为奇地址。

图 4-3: PIC24FJ128GB204 系列器件的数据存储空间映射



# PIC24FJ128GB204 系列

## 4.2.2 数据存储器构成和对齐方式

为了保持与 PIC<sup>®</sup>MCU 向后兼容和提高数据存储空间的效率，PIC24F 指令集既支持字操作，也支持字节操作。字节访问会在内部对按字对齐的存储空间的所有有效地址（EA）计算进行调整。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。

使用任何有效地址的低字节（LSB）决定要选择的字节，数据字节读操作将读取包含此字节的整个字。选中的字节将被放在数据总线的低字节处。也就是说，数据存储器与寄存器被组织为两个字节宽的并行实体，它们共享（字）地址译码，而写入线相互独立。数据字节写操作仅写入存储阵列和寄存器中与字节地址匹配的相应部分。

所有字访问都必须按偶地址对齐。不支持不对齐的数据取操作。因此当混合字节和字操作或从 8 位 MCU 代码移植时，必须要小心。若试图进行这种不对齐的读或写操作，则会产生地址错误陷阱。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。无论是哪种情况都将发生陷阱，从而允许系统和 / 或用户检查地址错误发生之前的机器状态。

所有装入任意 W 寄存器的字节都将装入 W 寄存器的低字节。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 位的有符号数据转换为 16 位的有符号值。或者，对于 16 位无符号数据，用户可以清零任何 W 寄存器的 MSB，方法是在相应的地址处执行一条零扩展（ZE）指令。

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

## 4.2.3 NEAR 数据空间

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地址单元。剩余的数据空间通过间接寻址访问。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 特殊功能寄存器（SFR）空间

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

SFR 分布在受其控制的大量模块中，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。表 4-2 给出了 SFR 空间的布局表，显示了实际实现 SFR 的位置。每个已实现的区域表示一个 32 字节区域，其中至少有一个地址实现为 SFR。表 4-3 至表 4-33 给出了已实现的 SFR 及其地址的完整列表。

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	系统	NVM/RTCC	PMP	CRC	PMD	I/O	加密	
200h	A/D/CTMU		CMP	TMR	OC		IC	I <sup>2</sup> C™/DSM
300h	SPI			PPS				
400h	USB			DMA				
500h	UART		—					
600h	—							
700h	—							

图注: — = 此存储块中未实现的 SFR。

表 4-3: CPU 内核寄存器映射

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	所有复位时的值	
WREG0	0000	工作寄存器 0																0000	
WREG1	0002	工作寄存器 1																0000	
WREG2	0004	工作寄存器 2																0000	
WREG3	0006	工作寄存器 3																0000	
WREG4	0008	工作寄存器 4																0000	
WREG5	000A	工作寄存器 5																0000	
WREG6	000C	工作寄存器 6																0000	
WREG7	000E	工作寄存器 7																0000	
WREG8	0010	工作寄存器 8																0000	
WREG9	0012	工作寄存器 9																0000	
WREG10	0014	工作寄存器 10																0000	
WREG11	0016	工作寄存器 11																0000	
WREG12	0018	工作寄存器 12																0000	
WREG13	001A	工作寄存器 13																0000	
WREG14	001C	工作寄存器 14																0000	
WREG15	001E	工作寄存器 15																0800	
SPLIM	0020	堆栈指针限制值寄存器																xxxx	
PCL	002E	程序计数器寄存器低位字																0000	
PCH	0030	—	—	—	—	—	—	—	—	程序计数器高字寄存器									0000
DSRPAG	0032	—	—	—	—	—	—	扩展数据空间读页地址寄存器										0001	
DSWPAG	0034	—	—	—	—	—	—	扩展数据空间写页地址寄存器										0001	
RCOUNT	0036	REPEAT 循环计数器寄存器																xxxx	
SR	0042	—	—	—	—	—	—	—	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000	
CORCON	0044	—	—	—	—	—	—	—	—	—	—	—	—	IPL3	r	—	—	0004	
DISICNT	0052	—	—	禁止中断计数器寄存器														xxxx	
TBLPAG	0054	—	—	—	—	—	—	—	—	表存储器页地址寄存器									0000

图注: — = 未实现, 读为 0; r = 保留位, 不要修改; x = 复位时的值未知。复位值以十六进制格式显示。

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	所有复位时的值
CNPD1	0056	CN15PDE	—	CN13PDE	CN12PDE	CN11PDE	CN10PDE <sup>(1)</sup>	CN9PDE <sup>(1)</sup>	CN8PDE <sup>(1)</sup>	CN7PDE	CN6PDE	CN5PDE	CN4PDE	CN3PDE	CN2PDE	CN1PDE	CN0PDE	0000
CNPD2	0058	—	CN30PDE	CN29PDE	CN28PDE <sup>(1)</sup>	CN27PDE	CN26PDE <sup>(1)</sup>	CN25PDE <sup>(1)</sup>	CN24PDE	CN23PDE	CN22PDE	CN21PDE	CN20PDE <sup>(1)</sup>	CN19PDE <sup>(1)</sup>	CN18PDE <sup>(1)</sup>	CN17PDE <sup>(1)</sup>	CN16PDE	0000
CNPD3	005A	—	—	—	—	—	—	—	—	—	—	—	CN36PDE <sup>(1)</sup>	CN35PDE <sup>(1)</sup>	CN34PDE <sup>(1)</sup>	CN33PDE <sup>(1)</sup>	—	0000
CNEN1	0062	CN15IE	—	CN13IE	CN12IE	CN11IE	CN10IE <sup>(1)</sup>	CN9IE <sup>(1)</sup>	CN8IE <sup>(1)</sup>	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0064	—	CN30IE	CN29IE	CN28IE <sup>(1)</sup>	CN27IE	CN26IE <sup>(1)</sup>	CN25IE <sup>(1)</sup>	CN24IE	CN23IE	CN22IE	CN21IE	CN20IE <sup>(1)</sup>	CN19IE <sup>(1)</sup>	CN18IE <sup>(1)</sup>	CN17IE <sup>(1)</sup>	CN16IE	0000
CNEN3	0066	—	—	—	—	—	—	—	—	—	—	—	CN36IE <sup>(1)</sup>	CN35IE <sup>(1)</sup>	CN34IE <sup>(1)</sup>	CN33IE <sup>(1)</sup>	—	0000
CNPU1	006E	CN15PUE	—	CN13PUE	CN12PUE	CN11PUE	CN10PUE <sup>(1)</sup>	CN9PUE <sup>(1)</sup>	CN8PUE <sup>(1)</sup>	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	0070	—	CN30PUE	CN29PUE	CN28PUE <sup>(1)</sup>	CN27PUE	CN26PUE <sup>(1)</sup>	CN25PUE <sup>(1)</sup>	CN24PUE	CN23PUE	CN22PUE	CN21PUE	CN20PUE <sup>(1)</sup>	CN19PUE <sup>(1)</sup>	CN18PUE <sup>(1)</sup>	CN17PUE <sup>(1)</sup>	CN16PUE	0000
CNPU3	0072	—	—	—	—	—	—	—	—	—	—	—	CN36PUE <sup>(1)</sup>	CN35PUE <sup>(1)</sup>	CN34PUE <sup>(1)</sup>	CN33PUE <sup>(1)</sup>	—	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 这些位在 28 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	所有复位时的值
INTCON1	0080	NSTDIS	—	—	—	—	—	—	—	—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTVT	DISI	—	—	—	—	—	—	—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	DMA1IF	AD1IF	U1TXIF	U1RXIF	SPI1TXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	DMA0IF	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	DMA2IF	—	—	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF	0000
IFS2	0088	—	DMA4IF	PMPIF	—	—	OC6IF	OC5IF	IC6IF	IC5IF	IC4IF	IC3IF	DMA3IF	CRYROLLIF	CRYFREEIF	SPI2TXIF	SPI2IF	0000
IFS3	008A	—	RTCIF	DMA5IF	SPI3RXIF	SPI2RXIF	SPI1RXIF	—	KEYSTRIF	CRYDNIF	INT4IF	INT3IF	—	—	MI2C2IF	SI2C2IF	—	0000
IFS4	008C	—	—	CTMUIF	—	—	—	—	HLVDIF	—	—	—	—	CRCIF	U2ERIF	U1ERIF	—	0000
IFS5	008E	—	—	—	—	SPI3TXIF	SPI3IF	U4TXIF	U4RXIF	U4ERIF	USB1IF	I2C2BCIF	I2C1BCIF	U3TXIF	U3RXIF	U3ERIF	—	0000
IFS6	0090	—	—	—	—	—	FSTIF	—	—	—	—	—	—	—	—	—	—	0000
IFS7	0092	—	—	—	—	—	—	—	—	—	—	JTAGIF	—	—	—	—	—	0000
IEC0	0094	—	DMA1IE	AD1IE	U1TXIE	U1RXIE	SPI1TXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	DMA0IE	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	DMA2IE	—	—	—	INT1IE	CNIE	CMIE	MI2C1IE	SI2C1IE	0000
IEC2	0098	—	DMA4IE	PMPIE	—	—	OC6IE	OC5IE	IC6IE	IC5IE	IC4IE	IC3IE	DMA3IE	CRYROLLIE	CRYFREEIE	SPI2TXIE	SPI2IE	0000
IEC3	009A	—	RTCIE	DMA5IE	SPI3RXIE	SPI2RXIE	SPI1RXIE	—	KEYSTRIE	CRYDNIE	INT4IE	INT3IE	—	—	MI2C2IE	SI2C2IE	—	0000
IEC4	009C	—	—	CTMUIE	—	—	—	—	HLVDIE	—	—	—	—	CRCIE	U2ERIE	U1ERIE	—	0000
IEC5	009E	—	—	—	—	SPI3TXIE	SPI3IE	U4TXIE	U4RXIE	U4ERIE	USB1IE	I2C2BCIE	I2C1BCIE	U3TXIE	U3RXIE	U3ERIE	—	0000
IEC6	00A0	—	—	—	—	—	FSTIE	—	—	—	—	—	—	—	—	—	—	0000
IEC7	00A2	—	—	—	—	—	—	—	—	—	—	JTAGIE	—	—	—	—	—	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	—	DMA0IP2	DMA0IP1	DMA0IP0	4444
IPC2	00A8	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1TXIP2	SPI1TXIP1	SPI1TXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0	—	T3IP2	T3IP1	T3IP0	4444
IPC3	00AA	—	—	—	—	—	DMA1IP2	DMA1IP1	DMA1IP0	—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0	0444
IPC4	00AC	—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0	—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0	4444
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP<2:0>			0004
IPC6	00B0	—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0	—	OC3IP2	OC3IP1	OC3IP0	—	DMA2IP2	DMA2IP1	DMA2IP0	4444
IPC7	00B2	—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0	4444
IPC8	00B4	—	CRYROLLIP2	CRYROLLIP1	CRYROLLIP0	—	CRYFREEIP2	CRYFREEIP1	CRYFREEIP0	—	SPI2TXIP2	SPI2TXIP1	SPI2TXIP0	—	SPI2IP2	SPI2IP1	SPI2IP0	4444
IPC9	00B6	—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0	—	IC3IP2	IC3IP1	IC3IP0	—	DMA3IP2	DMA3IP1	DMA3IP0	4444
IPC10	00B8	—	—	—	—	—	OC6IP2	OC6IP1	OC6IP0	—	OC5IP2	OC5IP1	OC5IP0	—	IC6IP2	IC6IP1	IC6IP0	0444
IPC11	00BA	—	—	—	—	—	DMA4IP2	DMA4IP1	DMA4IP0	—	PMPIP2	PMPIP1	PMPIP0	—	—	—	—	0440
IPC12	00BC	—	—	—	—	—	MI2C2IP2	MI2C2IP1	MI2C2IP0	—	SI2C2IP2	SI2C2IP1	SI2C2IP0	—	—	—	—	0440
IPC13	00BE	—	CRYDNIP2	CRYDNIP1	CRYDNIP0	—	INT4IP2	INT4IP1	INT4IP0	—	INT3IP2	INT3IP1	INT3IP0	—	—	—	—	4440
IPC14	00C0	—	SPI2RXIP2	SPI2RXIP1	SPI2RXIP0	—	SPI1RXIP2	SPI1RXIP1	SPI1RXIP0	—	—	—	—	—	KEYSTRIP2	KEYSTRIP1	KEYSTRIP0	4404
IPC15	00C2	—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0	—	DMA5IP2	DMA5IP1	DMA5IP0	—	SPI3RXIP2	SPI3RXIP1	SPI3RXIP0	0444

图注: — = 未实现, 读为 0, r = 保留位, 保持为 0。复位值以十六进制格式显示。

表 4-5: 中断控制器寄存器映射 (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	所有复位时的值
IPC16	00C4	—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—	4440
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	HLVDIP<2:0>			0004
IPC19	00CA	—	—	—	—	—	—	—	—	—	CTMUIP<2:0>			—	—	—	—	0040
IPC20	00CC	—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0	—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—	4440
IPC21	00CE	—	U4ERIP2	U4ERIP1	U4ERIP0	—	USB1IP2	USB1IP1	USB1IP0	—	I2C2BCIP2	I2C2BCIP1	I2C2BCIP0	—	I2C1BCIP2	I2C1BCIP1	I2C1BCIP0	4444
IPC22	00D0	—	SPI3TXIP2	SPI3TXIP1	SPI3TXIP0	—	SPI3IP2	SPI3IP1	SPI3IP0	—	U4TXIP2	U4TXIP1	U4TXIP0	—	U4RXIP2	U4RXIP1	U4RXIP0	4444
IPC26	00D8	—	—	—	—	—	FSTIP<2:0>			—	—	—	—	—	—	—	—	0400
IPC29	00DE	—	—	—	—	—	—	—	—	—	JTAGIP<2:0>			—	—	—	—	0040
INTTREG	00E0	CPUIRQ	r	VHOLD	—	ILR3	ILR2	ILR1	ILR0	VECNUM7	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0	0000

图注: — = 未实现, 读为 0, r = 保留位, 保持为 0。复位值以十六进制格式显示。

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	所有复位时的值
TMR1	024C	Timer1 寄存器																0000
PR1	024E	Timer1 周期寄存器																FFFF
T1CON	0250	TON	—	TSIDL	—	—	—	TECS1	TECS0	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	0252	Timer2 寄存器																0000
TMR3HLD	0254	Timer3 保持寄存器 (仅适用于 32 位定时器操作)																0000
TMR3	0256	Timer3 寄存器																0000
PR2	0258	Timer2 周期寄存器																FFFF
PR3	025A	Timer3 周期寄存器																FFFF
T2CON	025C	TON	—	TSIDL	—	—	—	TECS1	TECS0	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	025E	TON	—	TSIDL	—	—	—	TECS1	TECS0	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
TMR4	0260	Timer4 寄存器																0000
TMR5HLD	0262	Timer5 保持寄存器 (仅适用于 32 位定时器操作)																0000
TMR5	0264	Timer5 寄存器																0000
PR4	0266	Timer4 周期寄存器																FFFF
PR5	0268	Timer5 周期寄存器																FFFF
T4CON	026A	TON	—	TSIDL	—	—	—	TECS1	TECS0	—	TGATE	TCKPS1	TCKPS0	T45	—	TCS	—	0000
T5CON	026C	TON	—	TSIDL	—	—	—	TECS1	TECS0	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

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	所有复位时的值	
IC1CON1	02AA	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	IC1	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC1CON2	02AC	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D	
IC1BUF	02AE	输入捕捉 1 缓冲寄存器																	0000
IC1TMR	02B0	输入捕捉定时器值 1 寄存器																	xxxx
IC2CON1	02B2	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	IC1	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC2CON2	02B4	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D	
IC2BUF	02B6	输入捕捉 2 缓冲寄存器																	0000
IC2TMR	02B8	输入捕捉定时器值 2 寄存器																	xxxx
IC3CON1	02BA	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	IC1	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC3CON2	02BC	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D	
IC3BUF	02BE	输入捕捉 3 缓冲寄存器																	0000
IC3TMR	02C0	输入捕捉定时器值 3 寄存器																	xxxx
IC4CON1	02C2	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	IC1	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC4CON2	02C4	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D	
IC4BUF	02C6	输入捕捉 4 缓冲寄存器																	0000
IC4TMR	02C8	输入捕捉定时器值 4 寄存器																	xxxx
IC5CON1	02CA	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	IC1	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC5CON2	02CC	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D	
IC5BUF	02CE	输入捕捉 5 缓冲寄存器																	0000
IC5TMR	02D0	输入捕捉定时器值 5 寄存器																	xxxx
IC6CON1	02D2	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	IC1	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC6CON2	02D4	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D	
IC6BUF	02D6	输入捕捉 6 缓冲寄存器																	0000
IC6TMR	02D8	输入捕捉定时器值 6 寄存器																	xxxx

图注: — = 未实现, 读为 0。x = 复位时的值未知。复位值以十六进制格式显示。

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	所有复位时的值
OC1CON1	026E	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC1CON2	0270	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC1RS	0272	输出比较 1 辅助寄存器																0000
OC1R	0274	输出比较 1 寄存器																0000
OC1TMR	0276	输出比较定时器值 1 寄存器																xxxx
OC2CON1	0278	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC2CON2	027A	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC2RS	027C	输出比较 2 辅助寄存器																0000
OC2R	027E	输出比较 2 寄存器																0000
OC2TMR	0280	输出比较定时器值 2 寄存器																xxxx
OC3CON1	0282	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC3CON2	0284	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC3RS	0286	输出比较 3 辅助寄存器																0000
OC3R	0288	输出比较 3 寄存器																0000
OC3TMR	028A	输出比较定时器值 3 寄存器																xxxx
OC4CON1	028C	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC4CON2	028E	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC4RS	0290	输出比较 4 辅助寄存器																0000
OC4R	0292	输出比较 4 寄存器																0000
OC4TMR	0294	输出比较定时器值 4 寄存器																xxxx
OC5CON1	0296	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT1	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC5CON2	0298	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC5RS	029A	输出比较 5 辅助寄存器																0000
OC5R	029C	输出比较 5 寄存器																0000
OC5TMR	029E	输出比较定时器值 5 寄存器																xxxx
OC6CON1	02A0	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC6CON2	02A2	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC6RS	02A4	输出比较 6 辅助寄存器																0000
OC6R	02A6	输出比较 6 寄存器																0000
OC6TMR	02A8	输出比较定时器值 6 寄存器																xxxx

图注: — = 未实现, 读为 0。x = 复位时的值未知。复位值以十六进制格式显示。

表 4-9: I<sup>2</sup>C™ 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	所有复位时的值
I2C1RCV	02DA	—	—	—	—	—	—	—	—	I2C1 接收寄存器								0000
I2C1TRN	02DC	—	—	—	—	—	—	—	—	I2C1 发送寄存器								00FF
I2C1BRG	02DE	—	—	—	—	I2C1 波特率发生器寄存器												0000
I2C1CONL	02E0	I2CEN	—	I2CSIDL	SCLREL	STRICT	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C1CONH	02E2	—	—	—	—	—	—	—	—	—	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000
I2C1STAT	02E4	ACKSTAT	TRSTAT	ACKTIM	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/A	P	S	R/W	RBF	TBF	0000
I2C1ADD	02E6	—	—	—	—	—	—	I2C1 地址寄存器										0000
I2C1MSK	02E8	—	—	—	—	—	—	I2C1 地址掩码寄存器										0000
I2C2RCV	02EA	—	—	—	—	—	—	—	—	I2C2 接收寄存器								0000
I2C2TRN	02EC	—	—	—	—	—	—	—	—	I2C2 发送寄存器								00FF
I2C2BRG	02EE	—	—	—	—	I2C2 波特率发生器寄存器												0000
I2C2CONL	02F0	I2CEN	—	I2CSIDL	SCLREL	STRICT	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C2CONH	02F2	—	—	—	—	—	—	—	—	—	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000
I2C2STAT	02F4	ACKSTAT	TRSTAT	ACKTIM	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/A	P	S	R/W	RBF	TBF	0000
I2C2ADD	02F6	—	—	—	—	—	—	I2C2 地址寄存器										0000
I2C2MSK	02F8	—	—	—	—	—	—	I2C2 地址掩码寄存器										0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

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	所有复位时的值	
U1MODE	0500	UARTEN	—	USIDL	IREN	RTSMO	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	URXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000	
U1STA	0502	UTXISEL1	UTXINV	UTXISEL0	URXEN	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDL	PERR	FERR	OERR	URXDA	0110	
U1TXREG	0504	LAST	—	—	—	—	—	—	U1TXREG<8:0>									xxxx	
U1RXREG	0506	—	—	—	—	—	—	—	U1RXREG<8:0>									0000	
U1BRG	0508	U1BRG<15:0>																0000	
U1ADMD	050A	ADMASK<7:0>								ADMADDR<7:0>								0000	
U1SCCON	050C	—	—	—	—	—	—	—	—	—	—	TXRPT1	TXRPT0	CONV	T0PD	PTRCL	SCEN	0000	
U1SCINT	050E	—	—	RXRPTIF	TXRPTIF	—	—	WTCIF	GTCIF	—	PARIE	RXRPTIE	TXRPTIE	—	—	WTCIE	GTCIE	0000	
U1GTC	0510	—	—	—	—	—	—	—	GTC<8:0>									0000	
U1WTCL	0512	WTC<15:0>																0000	
U1WTCH	0514	—	—	—	—	—	—	—	—	WTC<23:16>									0000
U2MODE	0516	UARTEN	—	USIDL	IREN	RTSMO	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	URXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000	
U2STA	0518	UTXISEL1	UTXINV	UTXISEL0	URXEN	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDL	PERR	FERR	OERR	URXDA	0110	
U2TXREG	051A	LAST	—	—	—	—	—	—	U2TXREG<8:0>									xxxx	
U2RXREG	051C	—	—	—	—	—	—	—	U2RXREG<8:0>									0000	
U2BRG	051E	U2BRG<15:0>																0000	
U2ADMD	0520	ADMASK<7:0>								ADMADDR<7:0>								0000	
U2SCCON	0522	—	—	—	—	—	—	—	—	—	—	TXRPT1	TXRPT0	CONV	T0PD	PTRCL	SCEN	0000	
U2SCINT	0524	—	—	RXRPTIF	TXRPTIF	—	—	WTCIF	GTCIF	—	PARIE	RXRPTIE	TXRPTIE	—	—	WTCIE	GTCIE	0000	
U2GTC	0526	—	—	—	—	—	—	—	GTC<8:0>									0000	
U2WTCL	0528	WTC<15:0>																0000	
U2WTCH	052A	—	—	—	—	—	—	—	—	WTC<23:16>									0000
U3MODE	052C	UARTEN	—	USIDL	IREN	RTSMO	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	URXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000	
U3STA	052E	UTXISEL1	UTXINV	UTXISEL0	URXEN	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDL	PERR	FERR	OERR	URXDA	0110	
U3TXREG	0530	LAST	—	—	—	—	—	—	U3TXREG<8:0>									xxxx	
U3RXREG	0532	—	—	—	—	—	—	—	U3RXREG<8:0>									0000	
U3BRG	0534	U3BRG<15:0>																0000	
U3ADMD	0536	ADMASK<7:0>								ADMADDR<7:0>								0000	
U4MODE	0538	UARTEN	—	USIDL	IREN	RTSMO	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	URXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000	
U4STA	053A	UTXISEL1	UTXINV	UTXISEL0	URXEN	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDL	PERR	FERR	OERR	URXDA	0110	
U4TXREG	053C	LAST	—	—	—	—	—	—	U4TXREG<8:0>									xxxx	
U4RXREG	053E	—	—	—	—	—	—	—	U4RXREG<8:0>									0000	
U4BRG	0540	U4BRG<15:0>																0000	
U4ADMD	0542	ADMASK<7:0>								ADMADDR<7:0>								0000	

图注: — = 未实现, 读为 0。x = 复位时的值未知。复位值以十六进制格式显示。

表 4-11: SPI1 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	所有复位时的值
SPI1CON1L	0300	SPIEN	—	SPIIDL	DISSDO	MODE32	MODE16	SMP	CKE	SSEN	CKP	MSTEN	DISSDI	DISSCK	MCLKEN	SPIFE	ENHBUF	0000
SPI1CON1H	0302	AUDEN	SPIGNEXT	IGNROV	IGNTUR	AUDMONO	URDTEN	AUDMOD1	AUDMOD0	FRMEN	FRMSYNC	FRMPOL	MSEN	FRMSYPW	FRMCNT2	FRMCNT1	FRMCNT0	0000
SPI1CON2L	0304	—	—	—	—	—	—	—	—	—	—	—	WLENGTH<4:0>					0000
SPI1STATL	0308	—	—	—	FRMERR	SPIBUSY	—	—	SPITUR	SRMT	SPIROV	SPIRBE	—	SPITBE	—	SPITBF	SPIRBF	0028
SPI1STATH	030A	—	—	RXELM5	RXELM4	RXELM3	RXELM2	RXELM1	RXELM0	—	—	TXELM5	TXELM4	TXELM3	TXELM2	TXELM1	TXELM0	0000
SPI1BUFL	030C	SPI1BUFL<15:0>																0000
SPI1BUFH	030E	SPI1BUFH<31:16>																0000
SPI1BRGL	0310	—	—	—	SPI1BRG<12:0>												0000	
SPI1IMSKL	0314	—	—	—	FRMERREN	BUSYEN	—	—	SPITUREN	SRMTEN	SPIROVEN	SPIRBEN	—	SPITBEN	—	SPITBFEN	SPIRBFEN	0000
SPI1IMSKH	0316	RXWIEN	—	RXMSK5	RXMSK4	RXMSK3	RXMSK2	RXMSK1	RXMSK0	TXWIEN	—	TXMSK5	TXMSK4	TXMSK3	TXMSK2	TXMSK1	TXMSK0	0000
SPI1URDTL	0318	SPI1URDTL<15:0>																0000
SPI1URDTH	031A	SPI1URDTH<31:16>																0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-12: SPI2 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	所有复位时的值
SPI2CON1L	031C	SPIEN	—	SPIIDL	DISSDO	MODE32	MODE16	SMP	CKE	SSEN	CKP	MSTEN	DISSDI	DISSCK	MCLKEN	SPIFE	ENHBUF	0000
SPI2CON1H	031E	AUDEN	SPIGNEXT	IGNROV	IGNTUR	AUDMONO	URDTEN	AUDMOD1	AUDMOD0	FRMEN	FRMSYNC	FRMPOL	MSEN	FRMSYPW	FRMCNT2	FRMCNT1	FRMCNT0	0000
SPI2CON2L	0320	—	—	—	—	—	—	—	—	—	—	—	WLENGTH<4:0>					0000
SPI2STATL	0324	—	—	—	FRMERR	SPIBUSY	—	—	SPITUR	SRMT	SPIROV	SPIRBE	—	SPITBE	—	SPITBF	SPIRBF	0028
SPI2STATH	0326	—	—	RXELM5	RXELM4	RXELM3	RXELM2	RXELM1	RXELM0	—	—	TXELM5	TXELM4	TXELM3	TXELM2	TXELM1	TXELM0	0000
SPI2BUFL	0328	SPI2BUFL<15:0>																0000
SPI2BUFH	032A	SPI2BUFH<31:16>																0000
SPI2BRGL	032C	—	—	—	SPI2BRG<12:0>												0000	
SPI2IMSKL	0330	—	—	—	FRMERREN	BUSYEN	—	—	SPITUREN	SRMTEN	SPIROVEN	SPIRBEN	—	SPITBEN	—	SPITBFEN	SPIRBFEN	0000
SPI2IMSKH	0332	RXWIEN	—	RXMSK5	RXMSK4	RXMSK3	RXMSK2	RXMSK1	RXMSK0	TXWIEN	—	TXMSK5	TXMSK4	TXMSK3	TXMSK2	TXMSK1	TXMSK0	0000
SPI2URDTL	0334	SPI2URDTL<15:0>																0000
SPI2URDTH	0336	SPI2URDTH<31:16>																0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-13: SPI3 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	所有复位时的值
SPI3CON1L	0338	SPIEN	—	SPISIDL	DISSDO	MODE32	MODE16	SMP	CKE	SSEN	CKP	MSTEN	DISSDI	DISSCK	MCLKEN	SPIFE	ENHBUF	0000
SPI3CON1H	033A	AUDEN	SPISGNEXT	IGNROV	IGNTUR	AUDMONO	URDTEN	AUDMOD1	AUDMOD0	FRMEN	FRMSYNC	FRMPOL	MSEN	FRMSYPW	FRMCNT2	FRMCNT1	FRMCNT0	0000
SPI3CON2L	033C	—	—	—	—	—	—	—	—	—	—	—	WLENGTH<4:0>					0000
SPI3STATL	0340	—	—	—	FRMERR	SPIBUSY	—	—	SPITUR	SRMT	SPIROV	SPIRBE	—	SPITBE	—	SPITBF	SPIRBF	0028
SPI3STATH	0342	—	—	RXELM5	RXELM4	RXELM3	RXELM2	RXELM1	RXELM0	—	—	TXELM5	TXELM4	TXELM3	TXELM2	TXELM1	TXELM0	0000
SPI3BUFL	0344	SPI3BUFL<15:0>																0000
SPI3BUFH	0346	SPI3BUFH<31:16>																0000
SPI3BRGL	0348	—	—	—	SPI3BRG<12:0>													0000
SPI3IMSKL	034C	—	—	—	FRMERREN	BUSYEN	—	—	SPITUREN	SRMTEN	SPIROVEN	SPIRBEN	—	SPITBEN	—	SPITBFEN	SPIRBFEN	0000
SPI3IMSKH	034E	RXWIEN	—	RXMSK5	RXMSK4	RXMSK3	RXMSK2	RXMSK1	RXMSK0	TXWIEN	—	TXMSK5	TXMSK4	TXMSK3	TXMSK2	TXMSK1	TXMSK0	0000
SPI3URDTL	0350	SPI3URDTL<15:0>																0000
SPI3URDTH	0352	SPI3URDTH<31:16>																0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-14: PORTA 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	所有复位时的值
TRISA	0180	—	—	—	—	—	TRISA<10:7>					—	—	—	TRISA<3:0>			078F
PORTA	0182	—	—	—	—	—	RA<10:7>					—	—	RA<4:0>			xxxx	
LATA	0184	—	—	—	—	—	LATA<10:7>					—	—	—	LATA<3:0>			xxxx
ODCA	0186	—	—	—	—	—	ODA<10:7>					—	—	—	ODA<3:0>			0000

图注: — = 未实现, 读为 0。x = 复位时的值未知。复位值以十六进制格式显示。

表 4-15: 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	所有复位时的值
TRISB	018A	TRISB<15:13>			—	TRISB<11:5>					—	TRISB<3:0>			EFEF			
PORTB	018C	RB<15:13>			—	RB<11:0>								xxxx				
LATB	018E	LATB<15:13>			—	LATB<11:5>					—	LATB<3:0>			xxxx			
ODCB	0190	ODB<15:13>			—	ODB<11:5>					—	ODB<3:0>			0000			

图注: — = 未实现, 读为 0。x = 复位时的值未知。复位值以十六进制格式显示。

表 4-16: 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 <sup>(1)</sup>	Bit 8 <sup>(1)</sup>	Bit 7 <sup>(1)</sup>	Bit 6 <sup>(1)</sup>	Bit 5 <sup>(1)</sup>	Bit 4 <sup>(1)</sup>	Bit 3 <sup>(1)</sup>	Bit 2 <sup>(1)</sup>	Bit 1 <sup>(1)</sup>	Bit 0 <sup>(1)</sup>	所有复位时的值
TRISC	0194	—	—	—	—	—	—	TRISC<9:0>										03FF <sup>(2)</sup>
PORTC	0196	—	—	—	—	—	—	RC<9:0>										xxxx <sup>(2)</sup>
LATC	0198	—	—	—	—	—	—	LATC<9:0>										xxxx <sup>(2)</sup>
ODCC	019A	—	—	—	—	—	—	ODC<9:0>										0000 <sup>(2)</sup>

图注: — = 未实现, 读为 0。x = 复位时的值未知。复位值以十六进制格式显示。

注 1: 这些位在 28 引脚器件上未实现; 读为 0。

注 2: 显示的复位值适用于 44 引脚器件。

表 4-17: 引脚配置寄存器映射 (PADCFG1)

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	所有复位时的值
PADCFG1	01A0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PMPCTL	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-18: A/D 转换器寄存器映射

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	所有复位时的值	
ADC1BUF0	0200	A/D 数据缓冲区 0/ 通道 0 的阈值																xxxx	
ADC1BUF1	0202	A/D 数据缓冲区 1/ 通道 1 的阈值																xxxx	
ADC1BUF2	0204	A/D 数据缓冲区 2/ 通道 2 的阈值																xxxx	
ADC1BUF3	0206	A/D 数据缓冲区 3/ 通道 3 的阈值																xxxx	
ADC1BUF4	0208	A/D 数据缓冲区 4/ 通道 4 的阈值																xxxx	
ADC1BUF5	020A	A/D 数据缓冲区 5/ 通道 5 的阈值																xxxx	
ADC1BUF6	020C	A/D 数据缓冲区 6/ 通道 6 的阈值																xxxx	
ADC1BUF7	020E	A/D 数据缓冲区 7/ 通道 7 的阈值																xxxx	
ADC1BUF8	0210	A/D 数据缓冲区 8/ 通道 8 的阈值 / 窗口比较模式下通道 0 的阈值																xxxx	
ADC1BUF9	0212	A/D 数据缓冲区 9/ 通道 9 的阈值 / 窗口比较模式下通道 1 的阈值																xxxx	
ADC1BUF10	0214	A/D 数据缓冲区 10/ 通道 10 的阈值 / 窗口比较模式下通道 2 的阈值 <sup>(1)</sup>																xxxx	
ADC1BUF11	0216	A/D 数据缓冲区 11/ 通道 11 的阈值 / 窗口比较模式下通道 3 的阈值 <sup>(1)</sup>																xxxx	
ADC1BUF12	0218	A/D 数据缓冲区 12/ 通道 12 的阈值 / 窗口比较模式下通道 4 的阈值 <sup>(1)</sup>																xxxx	
ADC1BUF13	021A	A/D 数据缓冲区 13																xxxx	
ADC1BUF14	021C	A/D 数据缓冲区 14																xxxx	
ADC1BUF15	021E	A/D 数据缓冲区 15																xxxx	
AD1CON1	0220	ADON	—	ADSIDL	DMABM	DMAEN	MODE12	FORM1	FORM0	SSRC3	SSRC2	SSRC1	SSRC0	—	ASAM	SAMP	DONE	0000	
AD1CON2	0222	PVCFG1	PVCFG0	NVCFG0	OFFCAL	BUFREGEN	CSCNA	—	—	BUFS	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000	
AD1CON3	0224	ADRC	EXTSAM	PUMPEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000	
AD1CHS	0228	CH0NB2	CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA2	CH0NA1	CH0NA0	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000	
AD1CSSH	022A	CSS<31:27>						—	—	—	—	—	—	—	—	—	—	—	0000
AD1CSSL	022C	—	CSS<14:9> <sup>(2)</sup>						—	CSS<7:0>								0000	
AD1CON4	022E	—	—	—	—	—	—	—	—	—	—	—	—	—	DMABL<2:0>			0000	
AD1CON5	0230	ASEN	LPEN	CTMREQ	BGREQ	—	—	ASINT1	ASINT0	—	—	—	—	WM1	WM0	CM1	CM0	0000	
AD1CHITL	0234	—	—	—	CHH<12:9> <sup>(2)</sup>				—	CHH<7:0>								0000	
AD1CTMENL	0238	—	—	—	CTMEN<12:9> <sup>(2)</sup>				—	CTMEN<7:0>								0000	
AD1RESDMA	023A	A/D 转换数据缓冲区 (扩展缓冲模式)																xxxx	

图注: — = 未实现, 读为 0。x = 复位时的值未知。复位值以十六进制格式显示。

注 1: 这些位在 28 引脚器件上未实现; 读为 0。

注 2: CSS<12:10>、CHH<12:10> 和 CTMEN<12:10> 位在 28 引脚器件中未实现, 读为 0。

表 4-19: 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	所有复位时的值
CTMUCON1	023C	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	—	—	—	—	—	—	—	—	0000
CTMUCON2	023E	EDG1MOD	EDG1POL	EDG1SEL3	EDG1SEL2	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	EDG2MOD	EDG2POL	EDG2SEL3	EDG2SEL2	EDG2SEL1	EDG2SEL0	—	—	0000
CTMUICON	0240	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	—	—	—	—	—	—	—	—	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-20: 模拟配置寄存器映射

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	所有复位时的值
ANCFG	019E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VBG2EN	VBGEN	0000
ANSA	0188	—	—	—	—	—	—	—	—	—	—	—	—	ANSA<3:0>			000F	
ANSB	0192	ANSB<15:13>			—	—	—	ANSB9	—	—	ANSB6	—	—	ANSB<3:0>			E24F	
ANSC	019C	—	—	—	—	—	—	—	—	—	—	—	—	—	ANSC<2:0> <sup>(1)</sup>		0007	

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 这些位在 28 引脚器件上未实现; 读为 0。

表 4-21: DMA 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	所有复位时的值
DMACON	0450	DMAEN	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRSSEL	0000
DMABUF	0452	DMA 传输数据缓冲区																0000
DMAL	0454	DMA 高地址限制寄存器																0000
DMAH	0456	DMA 低地址限制寄存器																0000
DMACH0	0458	—	—	—	r	—	NULLW	RELOAD	CHREQ	SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	SIZE	CHEN	0000
DMAINT0	045A	DBUFWF	—	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	HIGHIF	LOWIF	DONEIF	HALFIF	OVRUNIF	—	—	HALFEN	0000
DMASRC0	045C	DMA 通道 0 源地址寄存器																0000
DMADST0	045E	DMA 通道 0 目标地址寄存器																0000
DMACNT0	0460	DMA 通道 0 事务计数寄存器																0001
DMACH1	0462	—	—	—	r	—	NULLW	RELOAD	CHREQ	SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	SIZE	CHEN	0000
DMAINT1	0464	DBUFWF	—	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	HIGHIF	LOWIF	DONEIF	HALFIF	OVRUNIF	—	—	HALFEN	0000
DMASRC1	0466	DMA 通道 1 源地址寄存器																0000
DMADST1	0468	DMA 通道 1 目标地址寄存器																0000
DMACNT1	046A	DMA 通道 1 事务计数寄存器																0001
DMACH2	046C	—	—	—	r	—	NULLW	RELOAD	CHREQ	SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	SIZE	CHEN	0000
DMAINT2	046E	DBUFWF	—	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	HIGHIF	LOWIF	DONEIF	HALFIF	OVRUNIF	—	—	HALFEN	0000
DMASRC2	0470	DMA 通道 2 源地址寄存器																0000
DMADST2	0472	DMA 通道 2 目标地址寄存器																0000
DMACNT2	0474	DMA 通道 2 事务计数寄存器																0001
DMACH3	0476	—	—	—	r	—	NULLW	RELOAD	CHREQ	SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	SIZE	CHEN	0000
DMAINT3	0478	DBUFWF	—	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	HIGHIF	LOWIF	DONEIF	HALFIF	OVRUNIF	—	—	HALFEN	0000
DMASRC3	047A	DMA 通道 3 源地址寄存器																0000
DMADST3	047C	DMA 通道 3 目标地址寄存器																0000
DMACNT3	047E	DMA 通道 3 事务计数寄存器																0001
DMACH4	0480	—	—	—	r	—	NULLW	RELOAD	CHREQ	SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	SIZE	CHEN	0000
DMAINT4	0482	DBUFWF	—	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	HIGHIF	LOWIF	DONEIF	HALFIF	OVRUNIF	—	—	HALFEN	0000
DMASRC4	0484	DMA 通道 4 源地址寄存器																0000
DMADST4	0486	DMA 通道 4 目标地址寄存器																0000
DMACNT4	0488	DMA 通道 4 事务计数寄存器																0001
DMACH5	048A	—	—	—	r	—	NULLW	RELOAD	CHREQ	SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	SIZE	CHEN	0000
DMAINT5	048C	DBUFWF	—	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	HIGHIF	LOWIF	DONEIF	HALFIF	OVRUNIF	—	—	HALFEN	0000
DMASRC5	048E	DMA 通道 5 源地址寄存器																0000
DMADST5	0490	DMA 通道 5 目标地址寄存器																0000
DMACNT5	0492	DMA 通道 5 事务计数寄存器																0001

图注: — = 未实现, 读为 0。r = 保留位。复位值以十六进制格式显示。

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	所有复位时的值
U1OTGIR	0400	—	—	—	—	—	—	—	—	IDIF	T1MSECIF	LSTATEIF	ACTVIF	SESVDFIF	SESENDIF	—	VBUSVDIF	0000
U1OTGIE	0402	—	—	—	—	—	—	—	—	IDIE	T1MSECIE	LSTATEIE	ACTVIE	SESVDFIE	SESENDIE	—	VBUSVDIE	0000
U1OTGSTAT	0404	—	—	—	—	—	—	—	—	ID	—	LSTATE	—	SESVDF	SESEND	—	VBUSVD	0000
U1OTGCON	0406	—	—	—	—	—	—	—	—	DPPULUP	DMPULUP	DPPULDWN	DMPULDWN	VBUSON	OTGEN	VBUSCHG	VBUSDIS	0000
U1PWRC	0408	—	—	—	—	—	—	—	—	UACTPND	—	—	USLPGRD	—	—	USUSPND	USBPWR	00x0
U1IR	040A <sup>(1)</sup>	—	—	—	—	—	—	—	—	STALLIF	—	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	URSTIF	0000
		—	—	—	—	—	—	—	—	STALLIF	ATTACHIF <sup>(1)</sup>	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	DETACHIF <sup>(1)</sup>	0000
U1IE	040C <sup>(1)</sup>	—	—	—	—	—	—	—	—	STALLIE	—	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	URSTIE	0000
		—	—	—	—	—	—	—	—	STALLIE	ATTACHIE <sup>(1)</sup>	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	DETACHIE <sup>(1)</sup>	0000
U1EIR	040E <sup>(1)</sup>	—	—	—	—	—	—	—	—	BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	0000
		—	—	—	—	—	—	—	—	BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	EOFEF <sup>(1)</sup>	PIDEF	0000
U1EIE	0410 <sup>(1)</sup>	—	—	—	—	—	—	—	—	BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	0000
		—	—	—	—	—	—	—	—	BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	EOFEE <sup>(1)</sup>	PIDEE	0000
U1STAT	0412	—	—	—	—	—	—	—	—	ENDPT3	ENDPT2	ENDPT1	ENDPT0	DIR	PPBI	—	—	0000
U1CON	0414 <sup>(1)</sup>	—	—	—	—	—	—	—	—	—	SE0	PKTDIS	—	HOSTEN	RESUME	PPBRST	USBEN	0000
		—	—	—	—	—	—	—	—	—	JSTATE <sup>(1)</sup>	SE0	TOKBUSY	USBRST	HOSTEN	RESUME	PPBRST	SOFEN <sup>(1)</sup>
U1ADDR	0416	—	—	—	—	—	—	—	—	LSPDEN <sup>(1)</sup>	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	0000
U1BDTP1	0418	—	—	—	—	—	—	—	—	USB 缓冲区描述符表基址寄存器							—	0000
U1FRML	041A	—	—	—	—	—	—	—	—	USB 帧计数器寄存器低字节							—	0000
U1FRMH	041C	—	—	—	—	—	—	—	—	USB 帧计数器寄存器高字节							—	0000
U1TOK <sup>(2)</sup>	041E	—	—	—	—	—	—	—	—	PID3	PID2	PID1	PID0	EP3	EP2	EP1	EP0	0000
U1SOF <sup>(2)</sup>	0420	—	—	—	—	—	—	—	—	USB 帧起始 (SOF) 计数寄存器							—	0000
U1BDTP2	0422	—	—	—	—	—	—	—	—	USB 缓冲区描述符表基址寄存器							—	0000
U1BDTP3	0424	—	—	—	—	—	—	—	—	USB 缓冲区描述符表基址寄存器							—	0000
U1CNFG1	0426	—	—	—	—	—	—	—	—	UTEYE	UOEMON	—	USBSIDL	—	—	PPB1	PPB0	0000
U1CNFG2	0428	—	—	—	—	—	—	—	—	—	—	—	PUVBUS	EXTI2CEN	UVBUSDIS	—	UTRDIS	0000
U1EP0	042A	—	—	—	—	—	—	—	—	LSPD <sup>(1)</sup>	RETRYDIS <sup>(1)</sup>	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP1	042C	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP2	042E	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP3	0430	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 模块工作在主机模式下的备用寄存器或位定义。

注 2: 这些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	所有复位时的值
U1EP4	0432	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP5	0434	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP6	0436	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP7	0438	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP8	043A	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP9	043C	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP10	043E	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP11	0440	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP12	0442	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP13	0444	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP14	0446	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP15	0448	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 模块工作在主机模式下的备用寄存器或位定义。

2: 这些寄存器仅在主机模式下可用。

表 4-23: 增强型并行主 / 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	所有复位时的值	
PMCON1	0128	PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	—	MODE1	MODE0	CSF1	CSF0	ALP	ALMODE	—	BUSKEEP	IRQM1	IRQM0	0000	
PMCON2	012A	PMPBUSY	—	ERROR	TIMEOUT	—	—	—	—	RADDR23	RADDR22	RADDR21	RADDR20	RADDR19	RADDR18	RADDR17	RADDR16	0000	
PMCON3	012C	PTWREN	PTRDEN	PTBE1EN	PTBE0EN	—	AWAITM1	AWAITM0	AWAITE	—	—	—	—	—	—	—	—	0000	
PMCON4	012E	—	PTEN14	—	—	—	—	PTEN<9:0>										0000	
PMCS1CF	0130	CSDIS	CSP	CSPTEN	BEP	—	WRSP	RDSP	SM	ACKP	PTSZ1	PTSZ0	—	—	—	—	—	0000	
PMCS1BS	0132	BASE<23:15>										—	—	—	BASE11	—	—	—	0200
PMCS1MD	0134	ACKM1	ACKM0	AMWAIT2	AMWAIT1	AMWAIT0	—	—	—	DWAITB1	DWAITB0	DWAITM3	DWAITM2	DWAITM1	DWAITM0	DWAITE1	DWAITE0	0000	
PMCS2CF	0136	CSDIS	CSP	CSPTEN	BEP	—	WRSP	RDSP	SM	ACKP	PTSZ1	PTSZ0	—	—	—	—	—	0000	
PMCS2BS	0138	BASE<23:15>										—	—	—	BASE11	—	—	—	0600
PMCS2MD	013A	ACKM1	ACKM0	AMWAIT2	AMWAIT1	AMWAIT0	—	—	—	DWAITB1	DWAITB0	DWAITM3	DWAITM2	DWAITM1	DWAITM0	DWAITE1	DWAITE0	0000	
PMDOUT1	013C	EPMP 数据输出寄存器 1<15:8>										EPMP 数据输出寄存器 1<7:0>						xxxx	
PMDOUT2	013E	EPMP 数据输出寄存器 2<15:8>										EPMP 数据输出寄存器 2<7:0>						xxxx	
PMDIN1	0140	EPMP 数据输入寄存器 1<15:8>										EPMP 数据输入寄存器 1<7:0>						xxxx	
PMDIN2	0142	EPMP 数据输入寄存器 2<15:8>										EPMP 数据输入寄存器 2<7:0>						xxxx	
PMSTAT	0144	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	008F	

图注: — = 未实现, 读为 0。x = 复位时的值未知。复位值以十六进制格式显示。

表 4-24: 实时时钟和日历 (RTCC) 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	所有复位时的值
ALRMVAL	011E	基于 ALRMPTR<1:0> 的闹钟值寄存器窗口																xxxx
ALCFGPRPT	0120	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
RTCVAL	0122	基于 RTCPTR<1:0> 的 RTCC 值寄存器窗口																xxxx
RCFGCAL	0124	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	注 1
RTCPWC	0126	PWCEN	PWCPOL	PWCPRE	PWSPRE	RTCLK1	RTCLK0	RTCOUT1	RTCOUT0	—	—	—	—	—	—	—	—	注 1

图注: — = 未实现, 读为 0。x = 复位时的值未知。复位值以十六进制格式显示。

注 1: POR 时 RCFGAL 和 RTCPWC 寄存器的状态为 0000 且在其他复位时保持不变。

表 4-25: 数据信号调制器 (DSM) 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	所有复位时的值
MDCON	02FA	MDEN	—	MDSIDL	—	—	—	—	—	—	MDOE	MDSLRL	MDOPOL	—	—	—	MDBIT	0020
MDSRC	02FC	—	—	—	—	—	—	—	—	SODIS	—	—	—	MS3	MS2	MS1	MS0	0000
MDCAR	02FE	CHODIS	CHPOL	CHSYNC	—	CH3	CH2	CH1	CH0	CLODIS	CLPOL	CLSYNC	—	CL3	CL2	CL1	CL0	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-26: 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	所有复位时的值
CMSTAT	0242	CMIDL	—	—	—	—	C3EVT	C2EVT	C1EVT	—	—	—	—	—	C3OUT	C2OUT	C1OUT	0000
CVRCON	0244	—	—	—	—	—	CVREFP	CVREFM1	CVREFM0	CVREN	CVROE	CVRSS	CVR4	CVR3	CVR2	CVR1	CVR0	0000
CM1CON	0246	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOLO	—	CREF	—	—	CCH1	CCH0	0000
CM2CON	0248	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOLO	—	CREF	—	—	CCH1	CCH0	0000
CM3CON	024A	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOLO	—	CREF	—	—	CCH1	CCH0	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-27: 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	所有复位时的值
CRCCON1	0158	CRCEN	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0	CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	—	—	—	0040
CRCCON2	015A	—	—	—	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0	—	—	—	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0	0000
CRCXORL	015C	X<15:1>															—	0000
CRCXORH	015E	X<31:16>															0000	
CRCDATL	0160	CRC 数据输入寄存器低位字															xxxx	
CRCDATL	0160	CRC 数据输入寄存器低位字															xxxx	
CRCDATH	0162	CRC 数据输入寄存器高位字															xxxx	
CRCWDATL	0164	CRC 结果寄存器低位字															xxxx	
CRCWDATH	0166	CRC 结果寄存器高位字															xxxx	

图注: — = 未实现, 读为 0。x = 复位时的值未知。复位值以十六进制格式显示。

表 4-28: 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	所有复位时的值
RPINR0	038C	—	—	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0	—	—	OCTRIG1R5	OCTRIG1R4	OCTRIG1R3	OCTRIG1R2	OCTRIG1R1	OCTRIG1R0	3F3F
RPINR1	038E	—	—	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0	—	—	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0	3F3F
RPINR2	0390	—	—	OCTRIG2R5	OCTRIG2R4	OCTRIG2R3	OCTRIG2R2	OCTRIG2R1	OCTRIG2R0	—	—	INT4R5	INT4R4	INT4R3	INT4R2	INT4R1	INT4R0	3F3F
RPINR7	039A	—	—	IC2R5	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0	—	—	IC1R5	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0	3F3F
RPINR8	039C	—	—	IC4R5	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0	—	—	IC3R5	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0	3F3F
RPINR9	039E	—	—	IC6R5	IC6R4	IC6R3	IC6R2	IC6R1	IC6R0	—	—	IC5R5	IC5R4	IC5R3	IC5R2	IC5R1	IC5R0	3F3F
RPINR11	03A2	—	—	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0	—	—	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0	3F3F
RPINR17	03AE	—	—	U3RXR<5:0>							—	—	—	—	—	—	—	3F00
RPINR18	03B0	—	—	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0	—	—	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0	3F3F
RPINR19	03B2	—	—	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0	—	—	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0	3F3F
RPINR20	03B4	—	—	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0	—	—	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0	3F3F
RPINR21	03B6	—	—	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0	—	—	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0	3F3F
RPINR22	03B8	—	—	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0	—	—	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0	3F3F
RPINR23	03BA	—	—	TMRCKR5	TMRCKR4	TMRCKR3	TMRCKR2	TMRCKR1	TMRCKR0	—	—	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0	3F3F
RPINR27	03C2	—	—	U4CTSR5	U4CTSR4	U4CTSR3	U4CTSR2	U4CTSR1	U4CTSR0	—	—	U4RXR5	U4RXR4	U4RXR3	U4RXR2	U4RXR1	U4RXR0	3F3F
RPINR28	03C4	—	—	SCK3R5	SCK3R4	SCK3R3	SCK3R2	SCK3R1	SCK3R0	—	—	SDI3R5	SDI3R4	SDI3R3	SDI3R2	SDI3R1	SDI3R0	3F3F
RPINR29	03C6	—	—	—	—	—	—	—	—	—	—	SS3R<5:0>					003F	
RPINR30	03C8	—	—	—	—	—	—	—	—	—	—	MDMIR<5:0>					003F	
RPINR31	03CA	—	—	MDC2R5	MDC2R4	MDC2R3	MDC2R2	MDC2R1	MDC2R0	—	—	MDC1R5	MDC1R4	MDC1R3	MDC1R2	MDC1R1	MDC1R0	3F3F

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-28: 外设引脚选择寄存器映射 (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	所有复位时的值	
RPOR0	03D6	—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0	—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0	0000	
RPOR1	03D8	—	—	RP3R5	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0	—	—	RP2R5	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0	0000	
RPOR2	03DA	—	—	RP5R<5:0>							—	—	—	—	—	—	—	—	0000
RPOR3	03DC	—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0	—	—	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0	0000	
RPOR4	03DE	—	—	RP9R5	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0	—	—	RP8R5	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0	0000	
RPOR5	03E0	—	—	RP11R5	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0	—	—	RP10R5	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0	0000	
RPOR6	03E2	—	—	RP13R<5:0>							—	—	—	—	—	—	—	—	0000
RPOR7	03E4	—	—	RP15R5	RP15R4	RP15R3	RP15R2	RP15R1	RP15R0	—	—	RP14R5	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0	0000	
RPOR8	03E6	—	—	RP17R5	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0	—	—	RP16R5	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0	0000	
RPOR9	03E8	—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0	—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0	0000	
RPOR10	03EA	—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0	—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0	0000	
RPOR11	03EC	—	—	RP23R5	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0	—	—	RP22R5	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0	0000	
RPOR12	03EE	—	—	RP25R5	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0	—	—	RP24R5	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0	0000	

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-29: 系统控制 (时钟和复位) 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	所有复位时的值	
RCON	0108	TRAPR	IOPUWR	—	RETEN	—	DPSLP	CM	VREGS	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	注 1	
OSCCON	0100	—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0	CLKLOCK	IOLOCK	LOCK	—	CF	POSCEN	SOSCEN	OSWEN	注 2	
CLKDIV	0102	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	CPDIV1	CPDIV0	PLEN	—	—	—	—	—	0100	
OSCTUN	0106	STEN	—	STSIDL	STSRC	STLOCK	STLPOL	STOR	STORPOL	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000	
REFOCONL	0168	ROEN	—	ROSIDL	ROOUT	ROSLP	—	ROSWEN	ROACTIVE	—	—	—	—	ROSEL3	ROSEL2	ROSEL1	ROSEL0	0000	
REFOCONH	016A	—	RODIV<14:0>															0000	
REFOTRIML	016C	ROTRIM<15:7>											—	—	—	—	—	—	0000
HLVDCON	010C	HLVDEN	—	LSIDL	—	—	—	—	—	VDIR	BGVST	IRVST	—	HLVDL3	HLVDL2	HLVDL1	HLVDL0	0000	
RCON2	010A	—	—	—	—	—	—	—	—	—	—	—	r	VDDBOR	VDDPOR	VBPOR	VBAT	注 1	

图注: — = 未实现, 读为 0; r = 保留位。复位值以十六进制格式表示。

注 1: RCON 寄存器的复位值与复位事件的类型有关。更多信息, 请参见第 7.0 节“复位”。

注 2: OSCCON 寄存器的复位值与复位事件的类型以及器件配置有关。更多信息, 请参见第 9.0 节“振荡器配置”。

表 4-30: 深度休眠寄存器映射

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	所有复位时的值
DSCON	010E	DSEN	—	—	—	—	—	—	—	—	—	—	—	—	r	DSBOR	RELEASE	0000 <sup>(1)</sup>
DSWAKE	0110	—	—	—	—	—	—	—	DSINT0	DSFLT	—	—	DSWDT	DSRTCC	DSMCLR	—	—	0000 <sup>(1)</sup>
DSGPR0	0112	深度休眠信号数据 0 寄存器																0000 <sup>(1)</sup>
DSGPR1	0114	深度休眠信号数据 1 寄存器																0000 <sup>(1)</sup>

图注: — = 未实现, 读为 0; r = 保留位。复位值以十六进制格式表示。

注 1: 这些位仅在发生 VDD POR 事件时复位。

表 4-31: 加密引擎寄存器映射

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	所有复位时的值
CRYCONL	01A4	CRYON	—	CRYSIDL	ROLLIE	DONEIE	FREEIE	—	CRYGO	OPMOD3	OPMOD2	OPMOD1	OPMOD0	CPHRSEL	CPHRMOD2	CPHRMOD1	CPHRMOD0	0000
CRYCONH	01A6	—	CTRSIZE6	CTRSIZE5	CTRSIZE4	CTRSIZE3	CTRSIZE2	CTRSIZE1	CTRSIZE0	SKEYSEL	KEYMOD1	KEYMOD0	—	KEYSRC3	KEYSRC2	KEYSRC1	KEYSRC0	0000
CRYSTAT	01A8	—	—	—	—	—	—	—	—	CRYBSY	TXTABSY	CRYABRT	ROLLOVR	—	MODFAIL	KEYFAIL	PGMFAIL	0000
CRYOTP	01AC	—	—	—	—	—	—	—	—	PGMTST	OTPIE	CRYREAD	KEYPG3	KEYPG2	KEYPG1	KEYPG0	CRYWR	0020
CRYTXTA	01B0	加密文本寄存器 A (128 位宽)																xxxx
CRYKEY	01C0	加密密钥寄存器 (256 位宽, 只写)																xxxx
CRYXTB	01E0	加密文本寄存器 B (128 位宽)																xxxx
CRYXTC	01F0	加密文本寄存器 C (128 位宽)																xxxx

图注: — = 未实现, 读为 0。x = 复位时的值未知。复位值以十六进制格式显示。

表 4-32: 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	所有复位时的值
NVMCON	0760	WR	WREN	WRERR	—	—	—	—	—	—	ERASE	—	—	NVMOP3	NVMOP2	NVMOP1	NVMOP0	0000 <sup>(1)</sup>
NVMKEY	0766	—	—	—	—	—	—	—	—	NVMKEY 寄存器 <7:0>							0000	

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 显示的复位值仅针对上电复位。其他复位状态的值取决于复位时存储器写操作或擦除操作的状态。

表 4-33: 外设模块禁止 (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	所有复位时的值
PMD1	0170	T5MD	T4MD	T3MD	T2MD	T1MD	—	—	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADC1MD	0000
PMD2	0172	—	—	IC6MD	IC5MD	IC4MD	IC3MD	IC2MD	IC1MD	—	—	OC6MD	OC5MD	OC4MD	OC3MD	OC2MD	OC1MD	0000
PMD3	0174	—	—	—	—	DSMMD	CMPMD	RTCCMD	PMPMD	CRCMD	—	—	—	U3MD	—	I2C2MD	—	0000
PMD4	0176	—	—	—	—	—	—	—	—	—	UPWMMD	U4MD	—	REFOMD	CTMUMD	HLVDMD	USB1MD	0000
PMD6	017A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPI3MD	0000
PMD7	017C	—	—	—	—	—	—	—	—	—	—	DMA1MD	DMA0MD	—	—	—	—	0000
PMD8	017E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CRYMD	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

## 4.2.5 扩展数据空间 (EDS)

扩展数据空间 (EDS) 允许 PIC24F 器件寻址相对于可能的 16 位地址范围更大范围的数据。EDS 包含低 32 KB 数据地址空间不可直接访问的任何其他内部数据存储空间和任何通过增强型并行主端口 (Enhanced Parallel Master Port, EPMP) 访问的外部存储空间。

此外, EDS 还允许对程序存储空间进行读访问。该特性称为程序空间可视性 (Program Space Visibility, PSV), 第 4.3.3 节“使用 EDS 从程序存储器读数据”将进行详细说明。

图 4-4 显示了整个 EDS 空间。EDS 以页形式组织, 其中名为 EDS 页的单个页等同于 EDS 窗口 (32 KB) 的大小。通过数据空间读寄存器 (DSRPAG) 或数据空间写寄存器 (DSWPAG) 选择特定 EDS 页。对于 PSV, 仅使用 DSRPAG 寄存器。DSRPAG 寄存器的值和 16 位宽数据地址的组合构成了 24 位有效地址 (EA)。

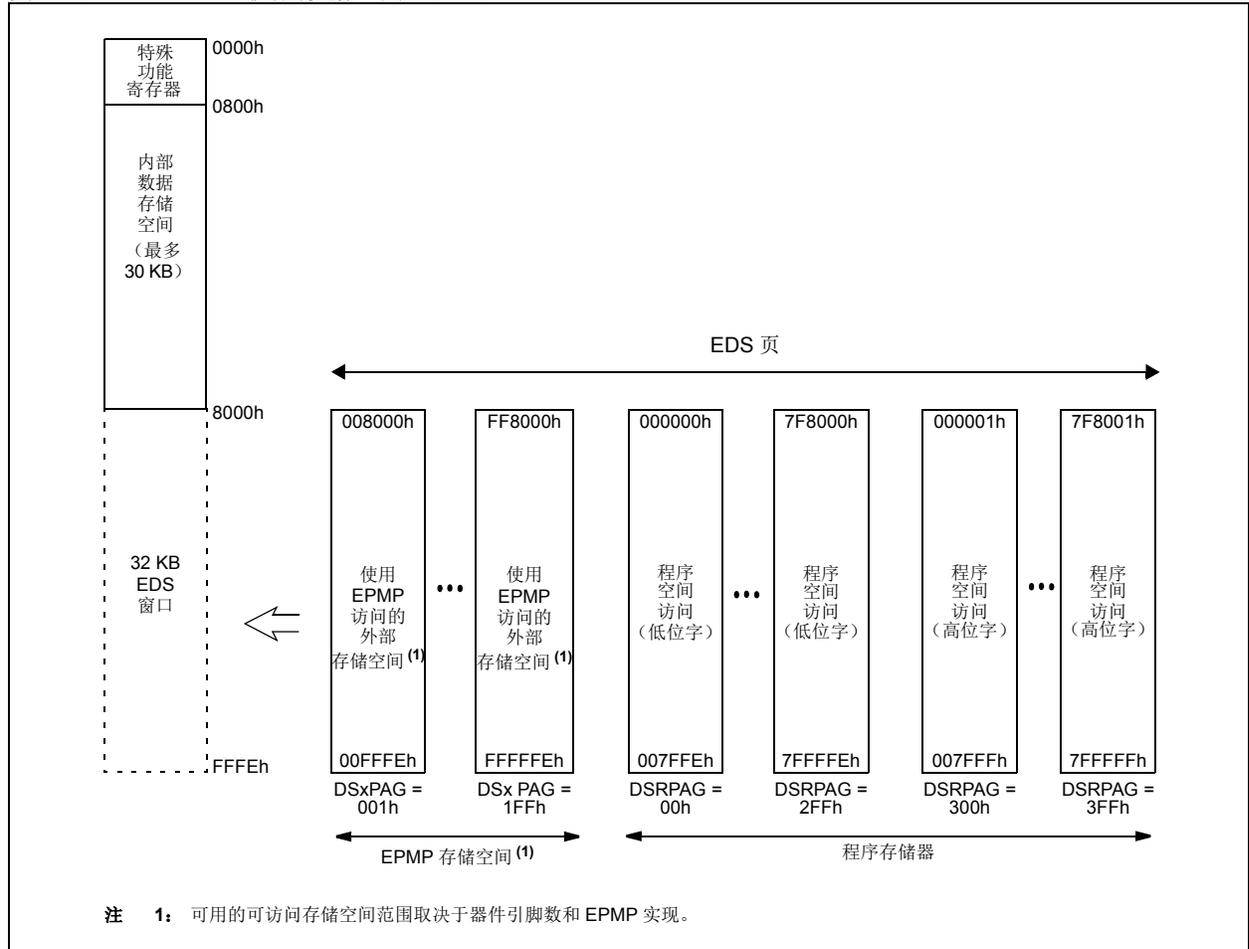
PIC24FJ128GB204 系列器件的数据寻址范围取决于具体器件上实现的增强型并行主端口的版本; 即与器件引脚数有关。表 4-34 列出了该系列中每个器件可访问的总存储空间。有关使用 EPMP 访问外部存储空间的更多详细信息, 请参见《dsPIC33/PIC24 系列参考手册》的“增强型并行主端口” (DS39730)。

表 4-34: 可访问的总数据存储空间

系列	内部 RAM	使用 EPMP 访问的外部 RAM
PIC24FJXXXGB204	8K	最多 16 MB
PIC24FJXXXGB202	8K	最多 64K

注: 访问 EDS 窗口的 Page 0 将生成地址错误陷阱, 因为 Page 0 是基本数据存储空间 (低数据空间中的数据存储单元 0800h 至 7FFFh)。

图 4-4: 扩展数据空间 (EDS)



# PIC24FJ128GB204 系列

## 4.2.5.1 从 EDS 读取数据

为了从 EDS 空间读取数据，首先，通过将所需的 EDS 页码装入到 DSRPAG 寄存器并将偏移地址分配给其中一个 W 寄存器来设置地址指针。一旦完成上述分配任务，通过将分配了偏移地址的工作寄存器的 bit 15 置 1 使能 EDS 窗口；然后，将读取所指 EDS 存储单元的内容。

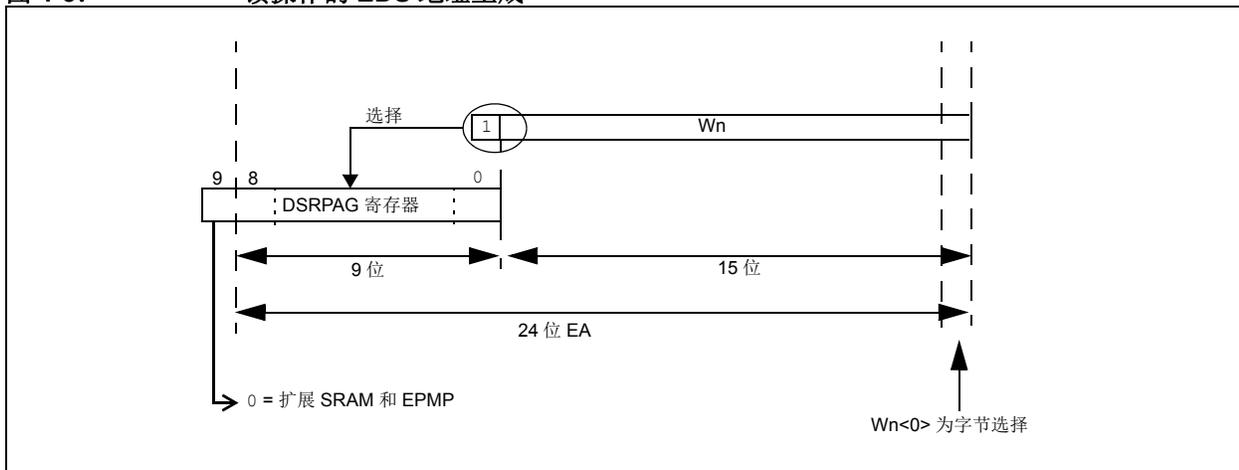
图 4-5 描述了如何为读操作生成 EDS 空间地址。

当 EA 的最高有效位为 1 且 DSRPAG<9> = 0 时，DSRPAG 的低 9 位与 EA 的低 15 位共同组成一个 24 位 EDS 空间地址以进行读操作。

例 4-1 显示如何从 EDS 读取一个字节、字和双字。

**注：** 对 EDS 空间的所有读操作均有一个指令周期的开销。因此，完成一次 EDS 读操作最少需要两个指令周期。在指令 REPEAT 下执行 EDS 读操作时，最初两次访问分别需要三个周期，而随后的访问仅需一个周期。

图 4-5: 读操作的 EDS 地址生成



例 4-1: EDS 读代码 (汇编语言)

```
; Set the EDS page from where the data to be read
mov    #0x0002, w0
mov    w0, DSRPAG    ;page 2 is selected for read
mov    #0x0800, w1    ;select the location (0x800) to be read
bset   w1, #15       ;set the MSB of the base address, enable EDS mode

;Read a byte from the selected location
mov.b  [w1++], w2    ;read Low byte
mov.b  [w1++], w3    ;read High byte

;Read a word from the selected location
mov    [w1], w2      ;

;Read Double - word from the selected location
mov.d  [w1], w2      ;two word read, stored in w2 and w3
```

## 4.2.5.2 数据写入 EDS

要将数据写入 EDS，与 EDS 读操作类似，通过将所需的 EDS 页码装入到 DSWPAG 寄存器并将偏移地址分配给其中一个 W 寄存器来设置地址指针。一旦完成上述分配任务，通过将分配了偏移地址的工作寄存器的 bit 15 置 1 使能 EDS 窗口，从而写入所访问的地址单元。

图 4-2 描述了如何为写操作生成 EDS 空间地址。

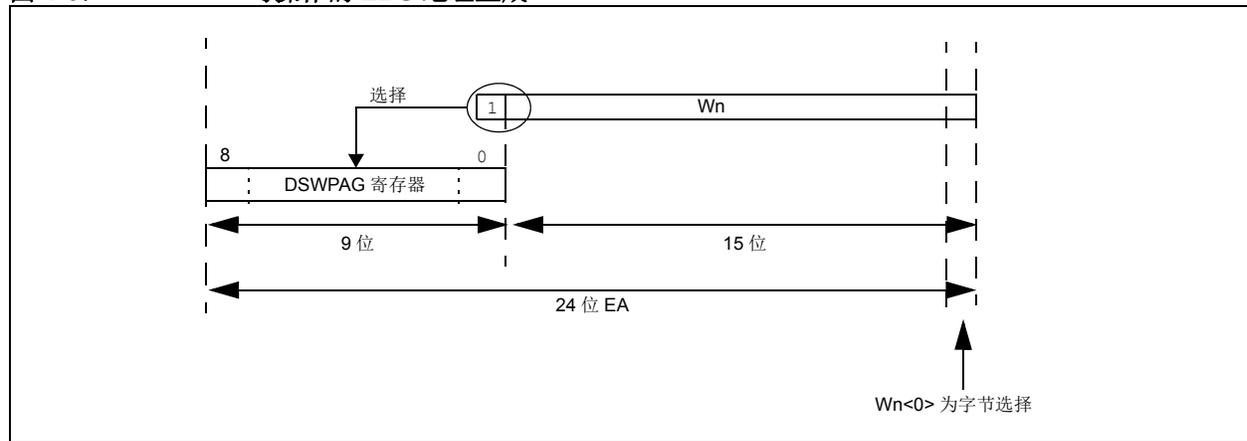
当 EA 的 MSB 为 1 时，DSWPAG 的低 9 位与 EA 的低 15 位共同组成一个 24 位 EDS 地址以进行写操作。例 4-2 显示了如何将一个字节、字或双字写入 EDS。

数据空间页寄存器（DSRPAG/DSWPAG）在跨页边界（发生从 0xFFFF 至 0x8000 的计满返回）时不会自动更新。用汇编语言开发代码时，在地址指针跨页边界时

更新数据空间页寄存器必须注意。C 编译器跟踪寻址，并在访问连续数据存储单元时相应增加或减少页寄存器值。

- 注 1:** 所有写 EDS 操作均在一个单周期内执行。
- 注 2:** 不支持在 REPEAT 指令下对任何 EDS 存储单元进行读 - 修改 - 写操作。例如，BCLR、BSW、BTG、RLC f、RLNC f、RRC f、RRNC f、ADD f、SUB f、SUBR f、AND f、IOR f、XOR f、ASR f 和 ASL f。
- 注 3:** 在执行读 - 修改 - 写操作时，使用 DSRPAG 寄存器。

图 4-6: 写操作的 EDS 地址生成



例 4-2: EDS 写代码（汇编语言）

```

; Set the EDS page where the data to be written
mov    #0x0002, w0
mov    w0, DSWPAG    ;page 2 is selected for write
mov    #0x0800, w1    ;select the location (0x800) to be written
bset   w1, #15       ;set the MSB of the base address, enable EDS mode

;Write a byte to the selected location
mov    #0x00A5, w2
mov    #0x003C, w3
mov.b  w2, [w1++]    ;write Low byte
mov.b  w3, [w1++]    ;write High byte

;Write a word to the selected location
mov    #0x1234, w2    ;
mov    w2, [w1]      ;

;Write a Double - word to the selected location
mov    #0x1122, w2
mov    #0x4455, w3
mov.d  w2, [w1]      ;2 EDS writes
    
```

# PIC24FJ128GB204 系列

表 4-35: 具有不同页和地址的 EDS 存储地址

DSRPAG (数据空间读 寄存器)	DSWPAG (数据空间写 寄存器)	间接寻址时的 源 / 目标地址	指向 EDS 的 24 位 EA	备注
x <sup>(1)</sup>	x <sup>(1)</sup>	0000h 至 1FFFh	000000h 至 001FFFh	Near 数据空间 <sup>(2)</sup>
		2000h 至 7FFFh	002000h 至 007FFFh	
001h	001h	8000h 至 FFFFh	008000h 至 00FFFEh	EPMP 存储空间
002h	002h		010000h 至 017FFEh	
003h	003h		018000h 至 0187FEh	
·	·		·	
1FFh	1FFh		FF8000h 至 FFFFFEh	
000h	000h		无效地址	地址错误陷阱 <sup>(3)</sup>

- 注 1: 如果源 / 目标地址低于 8000h, 将忽略 DSRPAG 和 DSWPAG 寄存器。  
 注 2: 该数据空间也可通过直接寻址访问。  
 注 3: 当源 / 目标地址高于 8000h 且 DSRPAG/DSWPAG 为 0 时, 将发生地址错误陷阱。

## 4.2.6 软件堆栈

除了用作工作寄存器外, PIC24F 器件中的 W15 寄存器还可用作软件堆栈指针 (SSP)。此指针总是指向第一个可用的空字, 从低地址向高地址方向增长。堆栈指针在弹出堆栈之前递减, 而在压入堆栈后递增, 如图 4-7 所示。注意, 对于执行任何 CALL 指令时的 PC 压栈操作, 在压入堆栈之前, PC 的 MSB 要进行零扩展, 从而确保 MSB 始终清零。

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

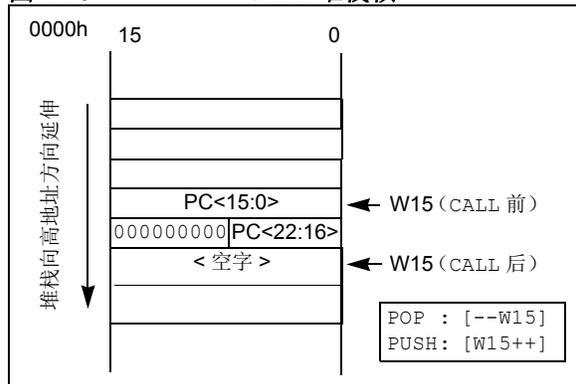
堆栈指针限制值寄存器 (Stack Pointer Limit Value, SPLIM) 与堆栈指针相关联, 它设置堆栈上边界的地址。SPLIM 在复位时不会被初始化。与堆栈指针一样, SPLIM<0> 被强制为 0, 因为所有的堆栈操作必须是字对齐的。每当使用 W15 作为源指针或目标指针产生有效地址时, 有效地址会与 SPLIM 中的值进行比较。如果堆栈指针 (W15) 的内容与 SPLIM 寄存器的内容相等, 则会执行压栈操作而不产生堆栈错误陷阱。但在随后的压栈操作时将会产生堆栈错误陷阱。因此, 例如如果想

要在堆栈增长到超过 RAM 中的地址 2000h 时产生堆栈错误陷阱, 则需用值 1FFEh 来初始化 SPLIM。

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

在对 SPLIM 寄存器进行写操作后, 不应紧跟使用 W15 进行间接读操作的指令。

图 4-7: 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中的一小块区域。这一功能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。重映射方式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。这一方式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，因此需要一种方法来从 16 位数据寄存器创建一个 23 位或 24 位程序地址。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）定义程序空间中一个 32K 字的区域。TBLPAG 寄存器的 8 位与 16 位 EA 组合形成了一个完整的 24 位程序空间地址。在这种地址格式下，TBLPAG 的 MSb 用来决定操作是发生在用户存储区（TBLPAG<7> = 0）中还是配置存储区（TBLPAG<7> = 1）中。

对于重映射操作，使用 10 位的扩展数据空间读寄存器（DSRPAG）定义程序空间中的 16K 字页。当 EA 的最高位（MSb）为 1 且 DSRPAG 的 MSb（bit 9）为 1 时，DSRPAG 的低 8 位与 EA 的低 15 位组合形成一个 23 位的程序空间地址。DSRPAG<8> 位决定是映射程序存储空间的低字位（当该位为 0 时）还是高字位（当该位为 1 时）。与表操作不同，重映射操作被严格限制在用户存储区中。

表 4-36 和图 4-8 显示了如何从数据 EA 创建程序 EA 来进行表操作和重映射访问。本文中，P<23:0> 指的是程序空间字，而 D<15:0> 指的是数据空间字。

表 4-36: 程序空间地址构成

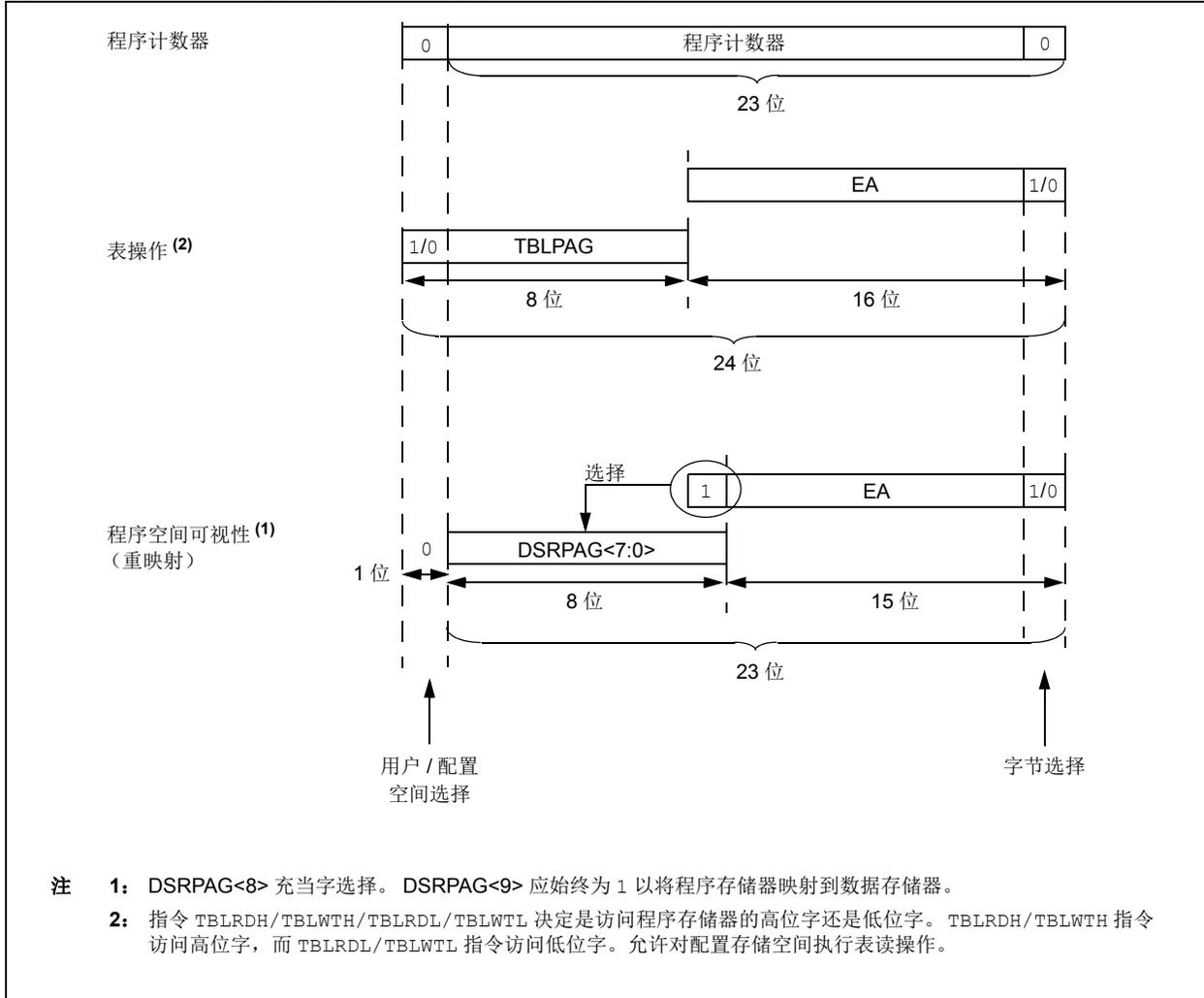
访问类型	访问空间	程序空间地址				
		<23>	<22:16>	<15>	<14:1>	<0>
通过指令访问 (代码执行)	用户	0	PC<22:1>			0
		0xx xxxx xxxx xxxx xxxx xxx0				
TBLRD/TBLWT (读 / 写字节或字)	用户	TBLPAG<7:0>		数据 EA<15:0>		
		0xxx xxxx		xxxx xxxx xxxx xxxx		
	配置	TBLPAG<7:0>		数据 EA<15:0>		
		1xxx xxxx		xxxx xxxx xxxx xxxx		
程序空间可视性 (块重映射 / 读)	用户	0	DSRPAG<7:0> <sup>(2)</sup>		数据 EA<14:0> <sup>(1)</sup>	
		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 DSRPAG<0>。

注 2: 在这种情况下，DSRPAG<9> 始终为 1。DSRPAG<8> 决定读取的是程序存储空间的低字位还是高字位。当 DSRPAG<8> 为 0，读低字位，而为 1 时，读高字位。

# PIC24FJ128GB204 系列

图 4-8: 访问程序空间内数据的地址生成方式



## 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 指令是可以把程序空间字的高 8 位作为数据读写的惟一方法。

对于每个连续的 24 位程序字，PC 的递增量通常为 2。这使得程序存储器地址能够被直接映射到数据空间地址中。于是，程序存储器可以被看作是两个 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 访问包含数据低位字的空间，TBLRDH 和 TBLWTH 访问包含高数据字的空间。

提供了两条表指令来对程序空间执行字节或字（16 位）大小的数据读写操作。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，始终为 0。

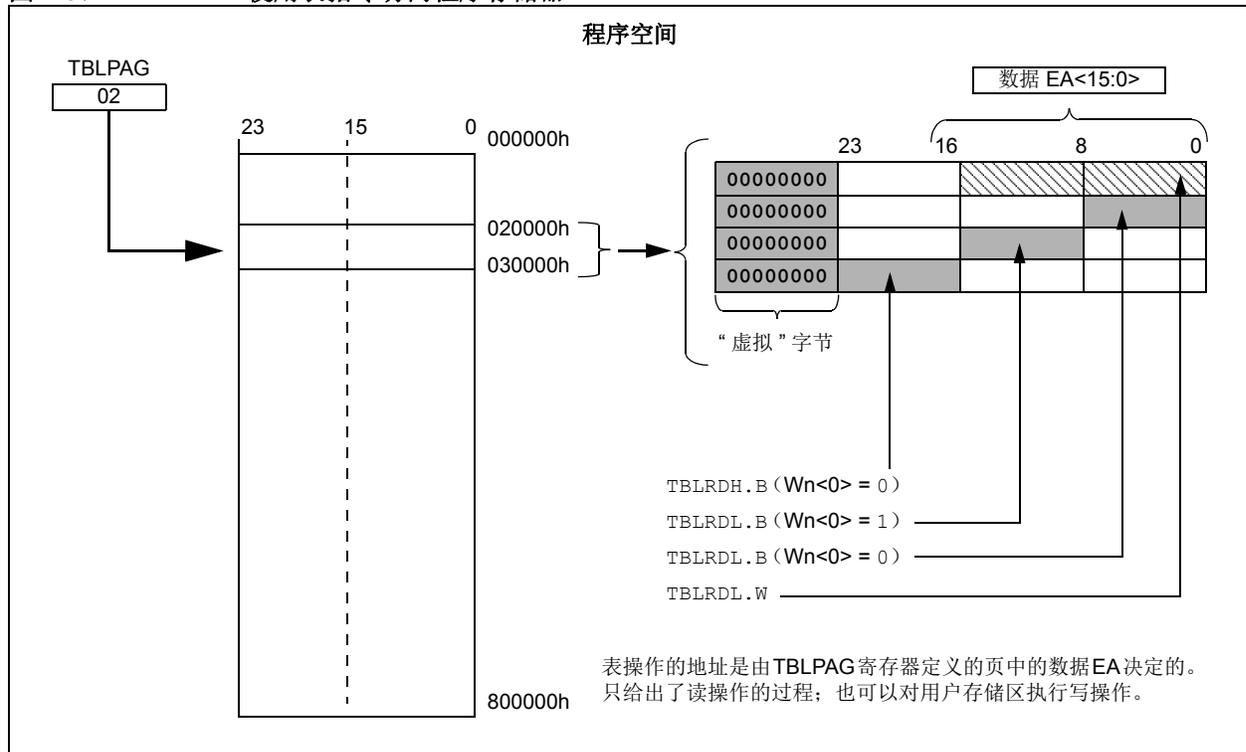
在字节模式下，程序字的高字节或低字节被映射到数据地址的 D<7:0> 中，这和 TBLRDL 指令相同。注意当选择高位“虚拟”字节（字节选择位 = 1）时，数据将始终为 0。

表指令 TBLWTH 和 TBLWTL 以类似的方式向程序地址空间写入各字节或字。第 6.0 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，包括用户存储空间和配置空间。当 TBLPAG<7> = 0 时，表页位于用户存储空间中。当 TBLPAG<7> = 1 时，表页位于配置空间中。

**注：** 仅表读操作可在配置存储空间（器件 ID 所在的空间）中执行；不允许表写操作。

图 4-9: 使用表指令访问程序存储器



# PIC24FJ128GB204 系列

## 4.3.3 使用 EDS 从程序存储器读数据

可选择将数据空间的高 32 KB 映射到程序空间中的任一 16K 字节中。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 指令）。

当 EA 的 MSb 为 1 且 DSRPAG<9> 位也为 1 时，就可以通过数据空间访问程序空间。DSRPAG 的低 8 位与 Wn<14:0> 组合形成了一个 23 位 EA 来访问程序存储器。DSRPAG<8> 位决定应访问哪个字；当该位为 0 时，访问低字位，而当该位为 1 时，访问程序存储器的高字位。

整个程序存储器分为 512 个 EDS 页，从 200h 至 3FFh，每个页由 16K 字数据组成。页 200h 至 2FFh 对应程序存储器的低位字，而 300h 至 3FFh 对应程序存储器的高位字。

使用 EDS 技术，可访问整个程序存储器。此前，不支持对程序存储器高位字的访问。

表 4-37 提供了与 EDS 页和源地址对应的程序存储器的 23 位 EDS 地址。

对于使用 PSV 而又在 REPEAT 循环之外执行的指令，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。其他所有的指令，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。

对于使用 PSV 而又在 REPEAT 循环内执行的指令，下列情况下，除了规定的指令执行时间之外，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。

表 4-37: 具有不同页和地址的 EDS 程序存储器地址

DSRPAG (数据空间读寄存器)	间接寻址时的源地址	指向 EDS 的 23 位 EA	备注
200h · · · 2FFh	8000h 至 FFFFh	000000h 至 007FFEh · · · 7F8000h 至 7FFFFEh	4M 程序指令的低位字； (8 MB) 仅用于读操作
300h · · · 3FFh		000001h 至 007FFFh · · · 7F8001h 至 7FFFFFh	
000h		无效地址	地址错误陷阱 <sup>(1)</sup>

注 1: 当源 / 目标地址高于 8000h 且 DSRPAG/DSWPAG 为 0 时，将发生地址错误陷阱。

例 4-3: 从程序存储器进行 EDS 读操作的代码（汇编语言）

```

; Set the EDS page from where the data to be read
mov    #0x0202, w0
mov    w0, DSRPAG                ;page 0x202, consisting lower words, is selected for read
mov    #0x000A, w1                ;select the location (0x0A) to be read
bset   w1, #15                    ;set the MSB of the base address, enable EDS mode
;Read a byte from the selected location
mov.b  [w1++], w2                 ;read Low byte
mov.b  [w1++], w3                 ;read High byte
;Read a word from the selected location
mov    [w1], w2                   ;
;Read Double - word from the selected location
mov.d  [w1], w2                   ;two word read, stored in w2 and w3
    
```

图 4-10: 访问低位字的程序空间可视性操作

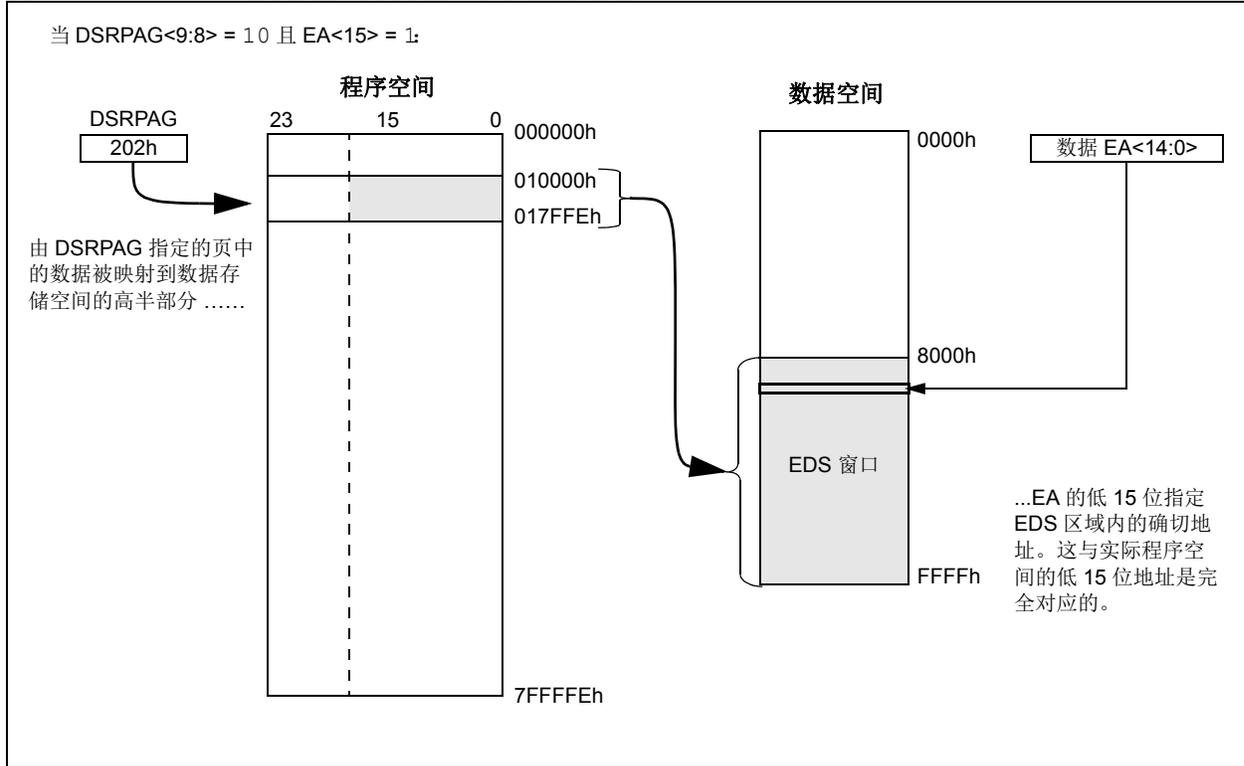
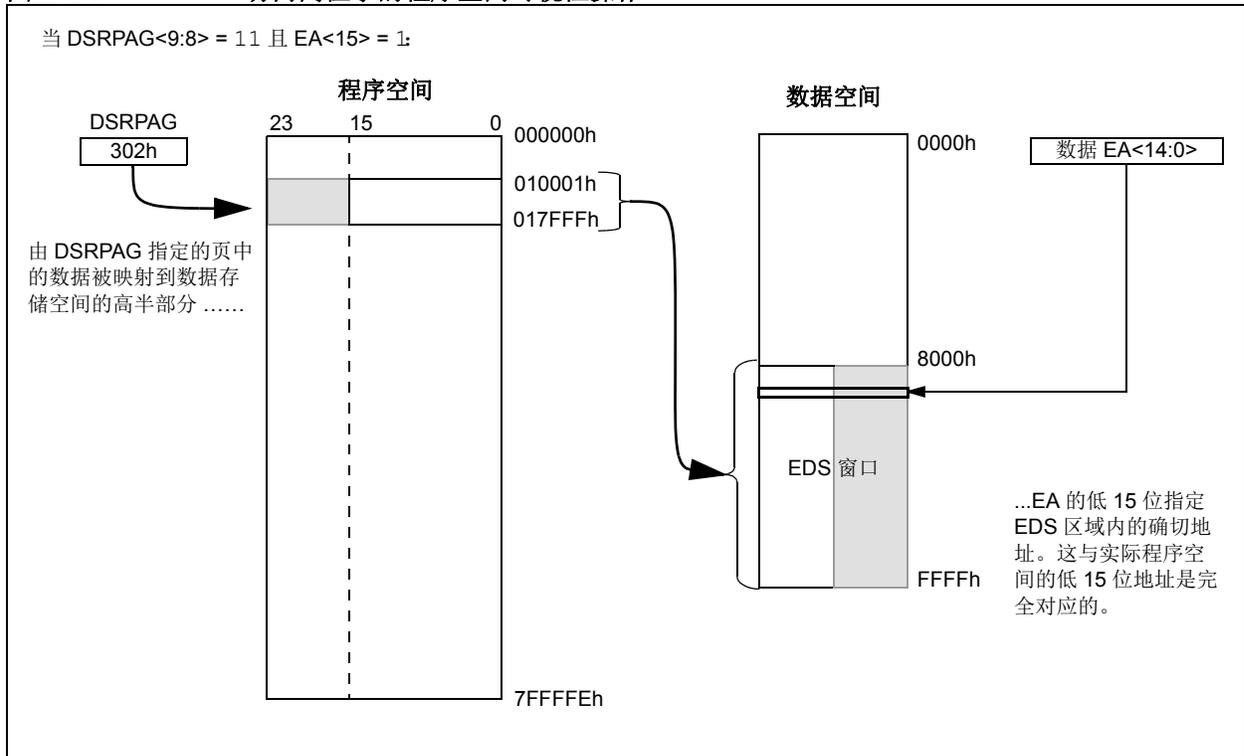


图 4-11: 访问高位字的程序空间可视性操作



# PIC24FJ128GB204 系列

---

注:

## 5.0 直接存储器访问控制器 (DMA)

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“直接存储器访问控制器 (DMA)” (DS39742)。本数据手册中的信息取代了FRM中的信息。

直接存储器访问控制器 (DMA) 旨在为工作在 SFR 总线上的高数据吞吐量外设提供服务，使这些外设可直接访问数据存储，而无需 CPU 密集型管理。通过使这些数据密集型外设共享各自的数据路径，也可减轻主数据总线负荷，从而进一步降低功耗。

DMA 控制器充当外设及 CPU 的直接扩展。DMA 控制器位于 CPU 和使能 DMA 的外设之间的单片机数据总线上，可直接访问 SRAM。这将 SFR 总线分为两个总线，使 DMA 控制器可访问位于新的 DMA SFR 总线上具有 DMA 功能的外设。控制器充当 DMA SFR 总线上的主器件，控制来自具有 DMA 功能的外设的数据流。

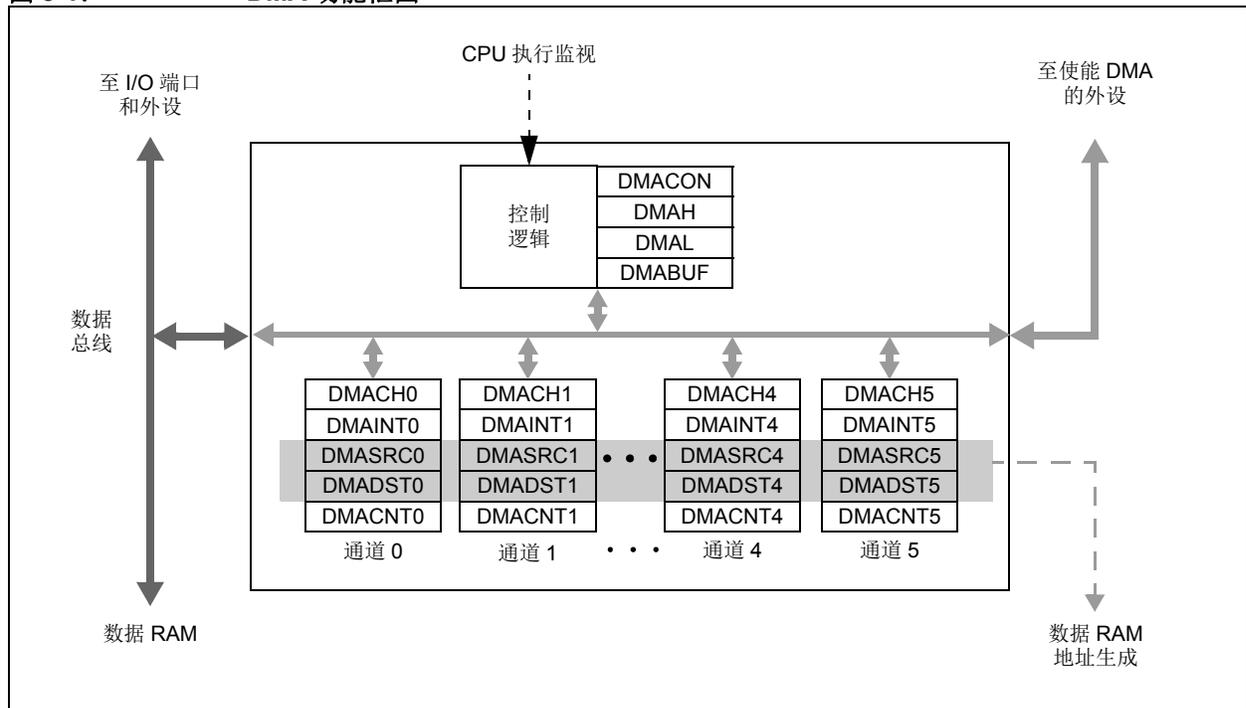
控制器还直接监视 CPU 指令处理，使其知道何时 CPU 需要访问 DMA 总线上的外设，并根据需要自动放弃对 CPU 的控制。这样，增大了用于处理数据的有效带宽，同时 DMA 操作不会造成处理器暂停。这使控制器基本上对用户是透明的。

DMA 控制器具有以下特性：

- 6 个独立的可独立编程通道
- 与 CPU 并行操作 (无 DMA 导致的等待状态)
- DMA 总线仲裁
- 5 个可编程地址模式
- 4 个可编程传输模式
- 4 个灵活的内部数据传输模式
- 支持字节或字数据传输
- 每个通道具有 16 位源和目标地址寄存器，可动态更新和重载
- 16 位事务计数寄存器，可动态更新和重载
- 高 / 低地址限制寄存器
- 计数器半满中断
- 软件触发传输
- 用于对称缓冲操作的空写模式

图 5-1 显示了 DMA 控制器的简化框图。

图 5-1: DMA 功能框图



# PIC24FJ128GB204 系列

## 5.1 DMA 操作汇总

DMA 控制器能够根据许多不同的参数在地址间移动数据。对于任何事务，其中每个参数都可以单独配置。另外，任何或所有 DMA 通道可以同时独立执行不同的事务。事务通过这些参数来分类：

- 源和目标（SFR 和数据 RAM）
- 数据大小（字节或字）
- 触发源
- 传输模式（单数据块、重复数据块或连续数据块）
- 寻址模式（固定地址或带或不带地址递增 / 递减的地址块）

另外，DMA 控制器为所有通道提供了优先级通道仲裁。

### 5.1.1 源和目标

使用 DMA 控制器，可在数据空间的任何两个地址之间传送数据。SFR 空间（0000h 至 07FFh）或数据 RAM 空间（0800h 至 FFFFh）可以用作源和目标。可以在这些区域或任何区域的地址之间进行任何方向的数据传送。图 5-2 给出了 4 个不同组合。

如果需要保护数据 RAM 区域，DMA 控制器允许用户为超出 SFR 空间的数据空间操作设置高 / 低地址边界。边界由 DMAH 和 DMAL 高 / 低地址限制寄存器设置。如果 DMA 通道尝试执行地址边界外的操作，事务会终止且产生中断。

### 5.1.2 数据大小

DMA 控制器可处理 8 位和 16 位事务。大小由用户选择，通过使用 SIZE 位（DMACHn<1>）。默认情况下，每个通道配置为字大小事务。当选择字节大小事务时，源和 / 或目标地址的 LSb 决定数据是表示数据 RAM 存储单元的高字节还是低字节。

### 5.1.3 触发源

DMA 控制器可使用器件中断源中的 63 个来启动事务。DMA 触发源的发生顺序与其自然中断优先级相反，如表 5-1 所示。

以下源不能用作 DMA 触发信号：

- 输入捕捉 8 和 9
- 输出比较 7、8 和 9
- USB

由于所有事务的源地址和目标地址可以独立于触发源编程，DMA 控制器可使用任何触发信号对任何外设执行操作。这也允许级联 DMA 通道以执行更复杂的传输操作。

### 5.1.4 传输模式

DMA 控制器支持 4 种类型的数据传输，基于每次触发要传送的数据量：

- 单数据块：每次触发发生单个事务。
- 连续数据块：每次触发发生一系列背靠背事务；事务的数量由 DMACNTn 事务计数器决定。
- 重复单数据块：每次发生触发时，重复执行单个事务，直到禁止 DMA 通道。
- 重复连续数据块：每次触发时，一个周期重复执行一系列事务，直到禁止 DMA 通道。

所有传输模式均支持事务完成后自动重载源地址和目标地址以及计数器值的选项；重复模式传输自动完成重载。

### 5.1.5 寻址模式

DMA 控制器也支持单个地址或地址范围内的传输。4 个基本选项为：

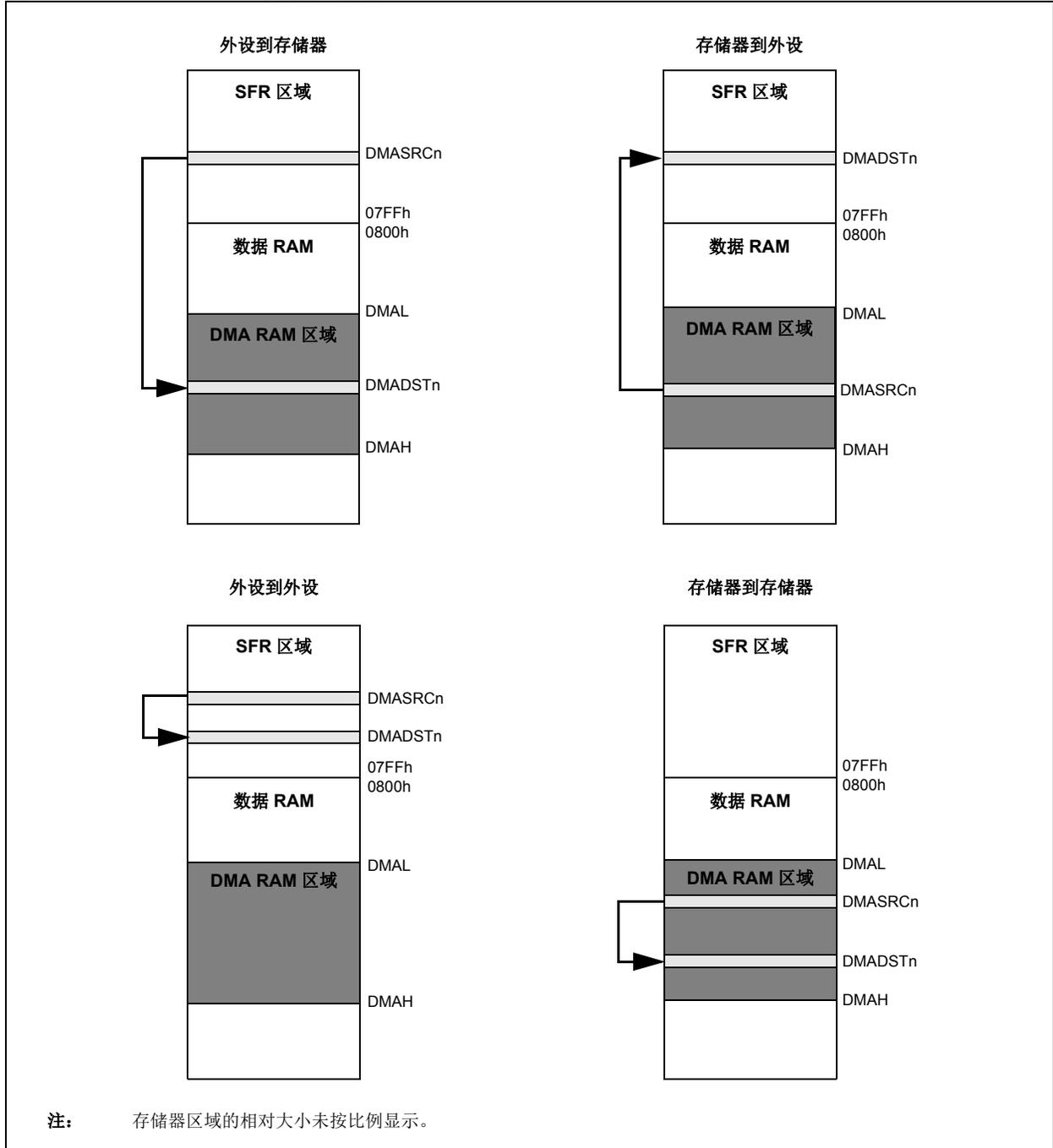
- 固定地址到固定地址：2 个固定地址间
- 固定地址到地址块：从固定源地址到一系列目标地址
- 地址块到固定地址：从一系列源地址到单个、固定目标地址
- 地址块到地址块：从一系列源地址到一系列目标地址

选择自动递增或递减源地址和 / 或目标地址的选项在地址块寻址模式下可用。

除了 4 种基本模式之外，DMA 控制器还支持外设间接寻址（Peripheral Indirect Addressing, PIA）模式，其中源地址或目标地址由 DMA 控制器和支持 PIA 的外设共同生成。当使能时，DMA 通道提供基本源地址和 / 或目标地址，而外设提供固定范围的偏移地址。

对于 PIC24FJ128GB204 系列器件，12 位 A/D 转换器模块是唯一支持 PIA 的外设。第 25.0 节“带阈值检测功能的 12 位 A/D 转换器”给出了其在 PIA 模式下使用的详细信息。

图 5-2: DMA 数据传输类型



# PIC24FJ128GB204 系列

## 5.1.6 通道优先级

每个 DMA 通道独立于其他通道工作，但是还与其他通道竞争访问数据和 DMA 总线。发生访问冲突时，DMA 控制器使用用户选择的优先级机制对通道仲裁。2 个可用的机制：

- 轮转：当 2 个或更多通道发生冲突时，编号较低的通道在发生第一次冲突时获得优先权。在后续冲突中，编号较高的通道基于其通道编号依次获得优先权。
- 固定优先级：当两个或更多通道发生冲突时，编号最低的通道始终获得优先权，与过去的历史无关。

## 5.2 典型的设置

要设置 DMA 通道用于基本数据传输：

1. 使能 DMA 控制器 (DMAEN = 1) 并通过置 1 或清零 PRSSEL 选择相应的通道优先级机制。
2. 使用用于数据 RAM 操作的适当高/低地址边界设置 DMAH 和 DMAL。
3. 选择要使用的 DMA 通道并禁止其操作 (CHEN = 0)。
4. 将事务的相应源和目标地址编程到通道的 DMASRCn 和 DMADSTn 寄存器。对于 PIA 寻址模式，使用基地址值。
5. 设置 DMACNTn 寄存器以获得每次传输的触发数量 (单数据块或连续数据块模式) 或要传输的字节 (字节) 数 (重复模式)。
6. 置 1 或清零 SIZE 位以选择数据大小。
7. 设置 TRMODEx 位以选择数据传输模式。
8. 设置 SAMODEx 和 DAMODEx 位以选择寻址模式。
9. 通过置 1 CHEN 使能 DMA 通道。
10. 允许触发源中断。

## 5.3 外设模块禁止

与其他外设模块不一样，DMA 控制器的通道不能使用外设模块禁止 x (PMDx) 寄存器单独掉电。改为将通道分两组控制。DMA0MD 位 (PMD7<4>) 用于选择控制 DMACH0 至 DMACH3。DMA1MD 位 (PMD7<5>) 用于控制 DMACH4 和 DMACH5。置 1 这两个位将有效禁止 DMA 控制器。

## 5.4 寄存器

DMA 控制器使用一些寄存器来控制其操作。寄存器的数量取决于特定器件所实现的通道数。

始终有 4 个模块级寄存器 (1 个控制寄存器和 3 个缓冲/地址寄存器)：

- DMACON：DMA 控制寄存器 (寄存器 5-1)
- DMAH 和 DMAL：DMA 高/低地址限制寄存器
- DMABUF：DMA 传输数据缓冲区

每个 DMA 通道实现 5 个寄存器 (2 个控制寄存器和 3 个缓冲/地址寄存器)：

- DMACHn：DMA 通道 n 控制寄存器 (寄存器 5-2)
- DMAINTn：DMA 通道 n 中断控制寄存器 (寄存器 5-3)
- DMASRCn：通道 n 的 DMA 通道源地址指针寄存器
- DMADSTn：通道 n 的 DMA 目标地址寄存器
- DMACNTn：通道 n 的 DMA 事务计数寄存器

对于 PIC24FJ128GB204 系列器件，总共有 34 个寄存器。

# PIC24FJ128GB204 系列

寄存器 5-1: **DMACON: DMA 引擎控制寄存器**

R/W-0	U-0						
DMAEN	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PRSEL
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

**DMAEN:** DMA 模块使能位

1 = 使能模块

0 = 禁止模块并终止所有正在进行的 DMA 操作

bit 14-1

**未实现:** 读为 0

bit 0

**PRSEL:** 通道优先级机制选择位

1 = 轮转机制

0 = 固定优先级机制

# PIC24FJ128GB204 系列

寄存器 5-2: **DMACHn: DMA 通道 n 控制寄存器**

U-0	U-0	U-0	r-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	NULLW	RELOAD <sup>(1)</sup>	CHREQ <sup>(3)</sup>
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
SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	SIZE	CHEN
bit 7							bit 0

<b>图注:</b>	r = 保留位
R = 可读位	W = 可写位
-n = POR 时的值	1 = 置 1
	U = 未实现位, 读为 0
	0 = 清零
	x = 未知

- bit 15-13 **未实现:** 读为 0
- bit 12 **保留:** 保持为 0
- bit 11 **未实现:** 读为 0
- bit 10 **NULLW:** 空写模式位  
1 = 每次写 DMADSTn 时, 启动对 DMASRCn 的假写操作  
0 = 不启动任何假写操作
- bit 9 **RELOAD:** 地址和计数重载位<sup>(1)</sup>  
1 = 在启动下一个操作时, DMASRCn、DMADSTn 和 DMACNTn 寄存器重新装载它们先前的值  
0 = 在启动下一个操作时, DMASRCn、DMADSTn 和 DMACNTn 不会重载<sup>(2)</sup>
- bit 8 **CHREQ:** DMA 通道软件请求位<sup>(3)</sup>  
1 = DMA 请求由软件启动; 在 DMA 传输结束时自动清零  
0 = 无任何 DMA 请求在等待
- bit 7-6 **SAMODE<1:0>:** 源地址模式选择位  
11 = DMASRCn 用于外设间接寻址且保持不变  
10 = 传输完成后, DMASRCn 根据 SIZE 位递减  
01 = 传输完成后, DMASRCn 根据 SIZE 位递增  
00 = 传输完成后, DMASRCn 保持不变
- bit 5-4 **DAMODE<1:0>:** 目标地址模式选择位  
11 = DMADSTn 用于外设间接寻址且保持不变  
10 = 传输完成后, DMADSTn 根据 SIZE 位递减  
01 = 传输完成后, DMADSTn 根据 SIZE 位递增  
00 = 传输完成后, DMADSTn 保持不变
- bit 3-2 **TRMODE<1:0>:** 传输模式选择位  
11 = 重复连续数据块模式  
10 = 连续数据块模式  
01 = 重复单数据块模式  
00 = 单数据块模式
- bit 1 **SIZE:** 数据大小选择位  
1 = 字节 (8 位)  
0 = 字 (16 位)
- bit 0 **CHEN:** DMA 通道使能位  
1 = 使能相应的通道  
0 = 禁止相应的通道

注 1: 只需存储 DMACNTn 的原始值来恢复 DMASRCn 和 DMADSTn 的原始值。  
 2: DMACNTn 始终在重复传输模式时重载, 而与 RELOAD 位的状态无关。  
 3: CHREQ 置 1 时执行的传输数量取决于 TRMODE<1:0> 的配置。

**寄存器 5-3: DMAINTn: DMA 通道 n 中断寄存器**

R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DBUFWF <sup>(1)</sup>	—	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
HIGHIF <sup>(1,2)</sup>	LOWIF <sup>(1,2)</sup>	DONEIF <sup>(1)</sup>	HALFIF <sup>(1)</sup>	OVRUNIF <sup>(1)</sup>	—	—	HALFEN
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        **DBUFWF:** DMA 缓冲数据写标志位 <sup>(1)</sup>  
 1 = 在空写模式下, DMA 缓冲区的内容未被写入 DMADSTn 或 DMASRCn 中指定的地址单元  
 0 = 在空写模式下, DMA 缓冲区的内容被写入 DMADSTn 或 DMASRCn 中指定的地址单元
- bit 14        **未实现:** 读为 0
- bit 13-8     **CHSEL<5:0>:** DMA 通道触发源选择位  
 完整列表请参见表 5-1。
- bit 7        **HIGHIF:** DMA 高地址限制中断标志位 <sup>(1,2)</sup>  
 1 = DMA 通道尝试访问高于 DMAH 的地址或数据 RAM 空间的上限  
 0 = DMA 通道未调用高地址限制中断
- bit 6        **LOWIF:** DMA 低地址限制中断标志位 <sup>(1,2)</sup>  
 1 = DMA 通道尝试访问低于 DMAL 但超出 SFR 范围 (07FFh) 的 DMA SFR 地址  
 0 = DMA 通道未调用低地址限制中断。
- bit 5        **DONEIF:** DMA 完成操作中中断标志位 <sup>(1)</sup>  
 如果 CHEN = 1:  
 1 = 前一 DMA 会话完成  
 0 = 当前 DMA 会话尚未完成  
 如果 CHEN = 0:  
 1 = 前一 DMA 会话完成  
 0 = 前一 DMA 会话尚未完成
- bit 4        **HALFIF:** DMA 50% 流水线级中断标志位 <sup>(1)</sup>  
 1 = DMACNTn 已达到与 0000h 相距的中点  
 0 = DMACNTn 未达到中点
- bit 3        **OVRUNIF:** DMA 通道溢出标志位 <sup>(1)</sup>  
 1 = DMA 通道仍在完成基于前一触发信号的操作时被触发  
 0 = 未发生溢出条件
- bit 2-1     **未实现:** 读为 0
- bit 0        **HALFEN:** 完成一半流水线位  
 1 = 当 DMACNTn 达到其中点和完成时调用中断  
 0 = 仅在传输完成时调用中断

**注 1:** 用软件将这些标志位置 1 不会产生中断。  
**2:** 实际访问之前未完成地址限制违反测试 (DMASRCn 或 DMADSTn 大于 DMAH 或小于 DMAL)。

# PIC24FJ128GB204 系列

表 5-1: DMA 通道触发源

CHSEL<5:0>	触发信号 (中断)	CHSEL<5:0>	触发信号 (中断)
000000	(未实现)	100000	UART2 发送
000001	SPI3 通用事件	100001	UART2 接收
000010	I2C1 从事件	100010	外部中断 2
000011	UART4 发送	100011	Timer5
000100	UART4 接收	100100	Timer4
000101	UART4 错误	100101	输出比较 4
000110	UART3 发送	100110	输出比较 3
000111	UART3 接收	100111	DMA 通道 2
001000	UART3 错误	101000	I2C2 从事件
001001	CTMU 事件	101001	外部中断 1
001010	HLVD	101010	电平变化中断
001011	CRC 完成	101011	比较器事件
001100	UART2 错误	101100	SPI3 接收事件
001101	UART1 错误	101101	I2C1 主事件
001110	RTCC	101110	DMA 通道 1
001111	DMA 通道 5	101111	A/D 转换器
010000	外部中断 4	110000	UART1 发送
010001	外部中断 3	110001	UART1 接收
010010	SPI2 接收事件	110010	SPI1 发送事件
010011	I2C2 主事件	110011	SPI1 通用事件
010100	DMA 通道 4	110100	Timer3
010101	EPMP	110101	Timer2
010110	SPI1 接收事件	110110	输出比较 2
010111	输出比较 6	110111	输入捕捉 2
011000	输出比较 5	111000	DMA 通道 0
011001	输入捕捉 6	111001	Timer1
011010	输入捕捉 5	111010	输出比较 1
011011	输入捕捉 4	111011	输入捕捉 1
011100	输入捕捉 3	111100	外部中断 0
011101	DMA 通道 3	111101	USB
011110	SPI2 发送事件	111110	SPI3 发送事件
011111	SPI2 通用事件	111111	加密完成

## 6.0 闪存程序存储器

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》中的“程序存储器”（DS39715）。本数据手册中的信息取代了FRM中的信息。

PIC24FJ128GB204 系列器件包含内部闪存程序存储器，用于存储和执行应用程序代码。程序存储器是可读写并可擦除的。闪存有以下四种编程方式：

- 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 允许在最终应用电路中对 PIC24FJ128GB204 系列器件进行串行编程。只需五根线就可以完成编程，它们分别是编程时钟线（PGECx）、编程数据线（PGEDx）、电源线（VDD）、接地线（VSS）和主复

位线（MCLR）。这允许用户使用未编程器件制造电路板，仅在产品交付前才对单片机进行编程。从而可以将最新版本的固件或定制固件烧写到单片机中。

RTSP 是通过 TBLRD（表读）和 TBLWT（表写）指令完成的。使用 RTSP，用户可以一次将 64 条指令（192 字节）的数据块写入程序存储器，也可以一次擦除 512 条指令（1536 字节）的数据块。

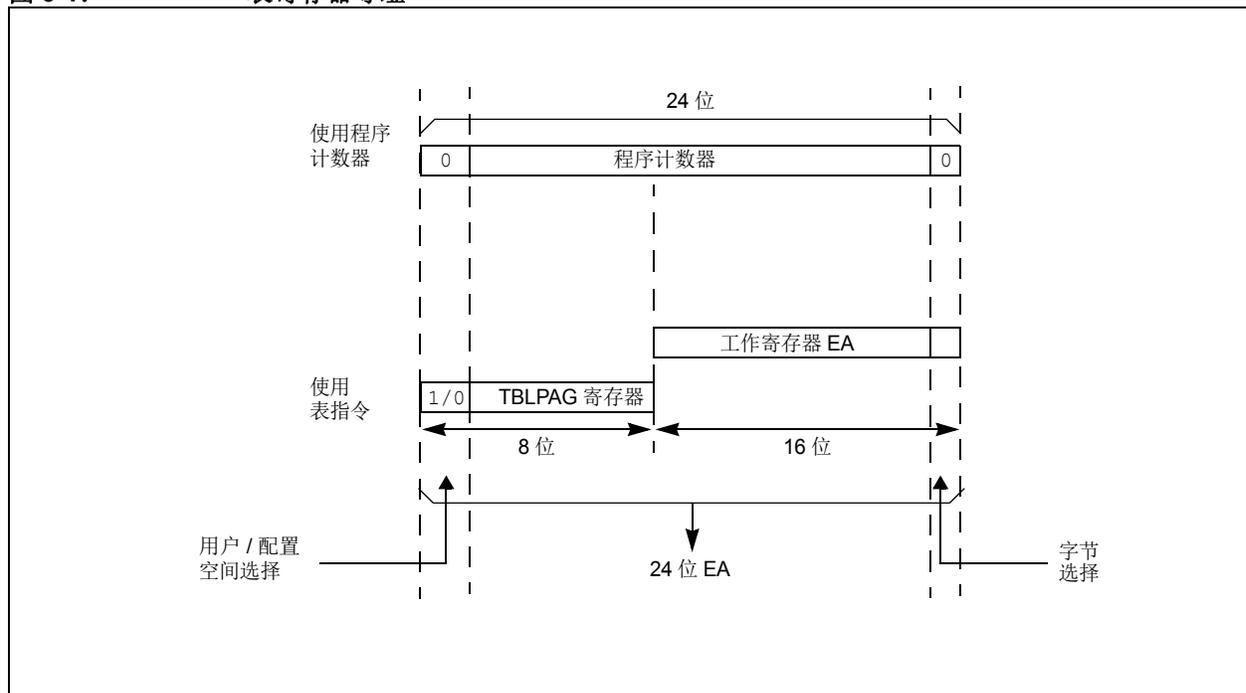
### 6.1 表指令和闪存编程

闪存的所有编程都是通过表读和表写指令完成的，与使用的编程方法无关。这些指令允许在器件正常工作模式下通过数据存储器直接读写程序存储空间。24 位程序存储器目标地址由 TBLPAG<7:0> 位和表指令中指定的 W 寄存器中的有效地址（EA）组成，如图 6-1 所示。

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

TBLRDH 和 TBLWTH 指令用于读或写程序存储器的 bit<23:16>。TBLRDH 和 TBLWTH 指令也能以字或字节模式访问程序存储器。

图 6-1: 表寄存器寻址



# PIC24FJ128GB204 系列

## 6.2 RTSP 操作

PIC24F 闪存程序存储器阵列以 64 条指令或 192 字节的行为单位构成。RTSP 允许用户一次擦除 8 行（512 条指令）的块并一次编程一行，它还可以编程单字。

8 行擦除块和单行写入块都是边界对齐的，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。

用 TBLWT 指令将数据写入程序存储器时，数据并未直接写入存储器，而是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，且写操作都将成功执行。但是，需要 64 条 TBLWT 指令来写存储器的整行。

要确保在写操作期间没有数据被改动，应将所有未使用的地址编程为 FFFFFFFh。这是因为保持锁存器复位为未知状态，因此当地址处于复位状态时，就可能改写未被重写的行上的存储单元。

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 指令以将数据装入缓冲器。通过将 NVMCON 寄存器的控制位置 1 来执行编程。

可按任何顺序装入数据，且在执行写操作之前可以多次写保持寄存器。但后续写操作将覆盖先前的所有写操作。

**注：** 不推荐多次写入一个存储单元而不擦除其内容。

因为只写缓冲器，因此所有表写操作都是单字写操作（2 个指令周期）。编程每行都需要一个编程周期。

## 6.3 JTAG 操作

PIC24F 系列器件支持 JTAG 边界扫描。边界扫描可以通过验证引脚到 PCB 的连通性改进制造工艺。

## 6.4 增强型在线串行编程

增强型在线串行编程使用片上自举程序（称为编程执行程序（Program Executive, PE）管理编程过程。通过使用 SPI 数据帧格式，编程执行程序可以擦除、编程和校验程序存储器。如需了解更多有关增强型 ICSP 的信息，请参见器件编程规范。

## 6.5 控制寄存器

有两个用于读写闪存程序存储器的 SFR：NVMCON 和 NVMKEY。

NVMCON 寄存器（寄存器 6-1）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时间。

NVMKEY 是只写寄存器，用于写保护。要启动编程或擦除过程，用户必须把 55h 和 AAh 连续写入 NVMKEY 寄存器。更多信息，请参见第 6.6 节“编程操作”。

## 6.6 编程操作

在 RTSP 模式下，对内部闪存进行编程或擦除需要完整的编程过程。在编程或擦除操作期间，处理器将停止（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。

# PIC24FJ128GB204 系列

寄存器 6-1: NVMCON: 闪存控制寄存器

R/S-0, HC <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R-0, HSC <sup>(1)</sup>	U-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	—	—	—	—	—
bit 15							bit 8

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

<b>图注:</b>	HC = 可由硬件清零的位	HSC = 可由硬件置 1/ 清零的位
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
		S = 可置 1 的位

- bit 15      **WR:** 写控制位 <sup>(1)</sup>  
 1 = 启动闪存编程或擦除操作。操作是自定时的, 一旦操作完成此位即由硬件清零。  
 0 = 编程或擦除操作完成, 并处于停止状态
- bit 14      **WREN:** 写使能位 <sup>(1)</sup>  
 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 <sup>(1)</sup>  
 1 = 尝试执行错误的编程或擦除序列或发生终止 (任何试图将 WR 位置 1 的操作都会自动置 1 此位)  
 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 <sup>(1)</sup>  
 1 = 在下一个写命令执行由 NVMOP<3:0> 位指定的擦除操作  
 0 = 在下一个写命令执行由 NVMOP<3:0> 位指定的编程操作
- bit 5-4     **未实现:** 读为 0
- bit 3-0     **NVMOP<3:0>:** NVM 操作选择位 <sup>(1,2)</sup>  
 1111 = 存储器块擦除操作 (ERASE = 1) 或无操作 (ERASE = 0) <sup>(3)</sup>  
 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:** 只能在上电复位时复位这些位。  
**2:** NVMOP<3:0> 的所有其他组合均未实现。  
**3:** 仅在 ICSP™ 模式下可用, 请参见器件编程规范。

# PIC24FJ128GB204 系列

## 6.6.1 闪存程序存储器编程算法

用户一次可编程闪存程序存储器的一行。要实现此操作，必须擦除包含该行在内的一个 8 行擦除块。一般步骤如下：

1. 读程序存储器中的 8 行（512 条指令），并将其存储在数据 RAM 中。
2. 用需要的新数据更新 RAM 中对应的程序数据。
3. 擦除块（见例 6-1）：
  - a) 将 NVMOPx 位（NVMCON<3:0>）设置为 0010，以配置为块擦除操作。将 ERASE（NVMCON<6>）和 WREN（NVMCON<14>）位置 1。
  - b) 将要擦除块的起始地址写入 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。启动擦除周期，CPU 停止工作等待擦除周期结束。擦除操作完成时，WR 位自动清零。

4. 将数据 RAM 的前 64 条指令写入程序存储器缓冲器（见例 6-3）。
5. 将程序块写入闪存：
  - a) 将 NVMOPx 位设置为 0001，以配置为行编程操作。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。启动编程周期，CPU 停止工作等待写周期结束。当写闪存的操作完成时，WR 位自动清零。
6. 将 TBLPAG 的值递增 1，使用数据 RAM 中下一个 64 条指令的块重复步骤 4 和 5，直到将所有 512 条指令写回闪存。

为了防止意外操作，必须向 NVMKEY 写入启动序列从而允许执行任何擦除或编程操作。执行了编程命令以后，用户必须等待一定的编程时间，直到编程操作完成。编程序列开始后紧跟的两条指令必须为 NOP，如例 6-4 所示。

### 例 6-1: 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 Program Memory (PM) Page Boundary SFR
MOV    #tbloffset(PROG_ADDR), 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.B  #0x55, W0
MOV    W0, NVMKEY ; Write the 0x55 key
MOV.B  #0xAA, W1 ;
MOV    W1, NVMKEY ; Write the 0xAA key
BSET   NVMCON, #WR ; Start the erase sequence
NOP ; Insert two NOPs after the erase
NOP ; command is asserted
```

## 例 6-2: 擦除程序存储器块 (C 语言代码)

```
// C example using MPLAB C30
unsigned long progAddr = 0XXXXXXX; // 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(); // check function to perform unlock
// sequence and set WR
```

## 例 6-3: 装载写缓冲器

```
; 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_63, W2 ;
MOV #HIGH_BYTE_63, W3 ;
TBLWTL W2, [W0] ; Write PM low word into program latch
TBLWTH W3, [W0] ; Write PM high byte into program latch
```

## 例 6-4: 启动编程序列

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

# PIC24FJ128GB204 系列

## 6.6.2 编程闪存程序存储器的一个单字

若已擦除了一个闪存单元，则可用表写指令对此单元进行编程以将一个指令字（24 位）写入写锁寄存器。将闪存地址的 8 个高字节（MSB）装入 TBLPAG 寄存器。TBLWTL 和 TBLWTH 指令将所需数据写入写锁寄存器，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，应将 NVMOPx 位（NVMCON<3:0>）设置为 0011。通过执行解锁序列并将 WR 位置 1 来执行此写操作（见例 6-5）。例 6-6 为使用 MPLAB® C30 编译器和内置硬件功能通过 C 编译器语言实现的等同过程。

### 例 6-5: 编程闪存程序存储器的一个单字

```
; 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_N, W2              ;
MOV    #HIGH_BYTE_N, 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.B  #0x55, W0                     ; Write the key sequence
MOV    W0, NVMKEY
MOV.B  #0xAA, W0
MOV    W0, NVMKEY
BSET   NVMCON, #WR                  ; Start the write cycle
NOP                                         ; Required delays
NOP
```

### 例 6-6: 编程闪存程序存储器的一个单字（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(&#xD2;DISI #5&#xD3;); // Block interrupts with priority <7
// for next 5 instructions
__builtin_write_NVM(); // C30 function to perform unlock
// sequence and set WR
```

## 7.0 复位

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“复位”（DS39712）。本数据手册中的信息取代了FRM中的信息。

复位模块将所有的复位源组合在一起并控制器件主复位信号 **SYSRST**。以下所列是器件的复位源：

- **POR**：上电复位
- **MCLR**：主复位引脚
- **SWR**：RESET 指令
- **WDT**：看门狗定时器复位
- **BOR**：欠压复位
- **CM**：配置失配复位
- **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

复位模块的简化框图如图 7-1 所示。

任何激活的复位源都会激活 **SYSRST** 信号。许多与 CPU 和外设有关的寄存器被强制为一个已知的复位状态。大多数寄存器不受复位的影响；在 **POR** 时它们的状态未知，而在所有其他复位时它们的状态不变。

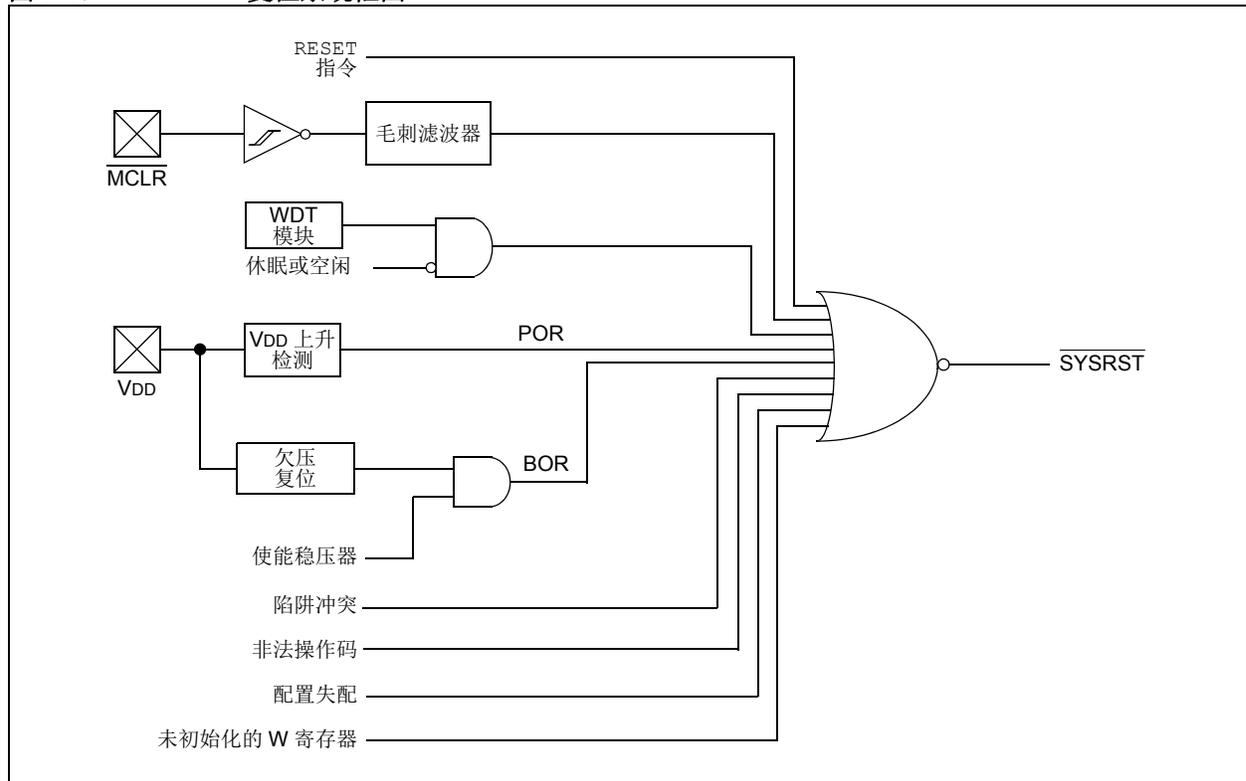
**注：** 有关寄存器复位状态的信息，请参见本手册中的特定外设或 CPU 章节。

任何类型的器件复位都会将 **RCON** 寄存器中相应的状态位置 1 以表明复位类型（见寄存器 7-1）。另外，使用超低功耗功能时（例如 **VBAT**）发生复位事件，将 **RCON2** 寄存器（寄存器 7-2）中的一个或多个状态位置 1。上电复位会清零 **BOR** 和 **POR** 位（**RCON**<1:0>）外的所有位，而 **BOR** 和 **POR** 位会被置 1。用户可在代码执行过程中的任何时间置 1 或清零任何位。**RCON** 位仅作为状态位。用软件将特定的复位状态位置 1 不会导致器件发生复位。

**RCON** 寄存器也有一些与看门狗定时器和器件节能状态相关的位。本数据手册的其他章节中将讨论这些位的功能。

**注：** **RCON** 寄存器中的状态位应该在被读取后清零，这样下一次器件复位后的 **RCON** 寄存器值才会有意义。

图 7-1: 复位系统框图



# PIC24FJ128GB204 系列

## 寄存器 7-1: RCON: 复位控制寄存器

R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
TRAPR <sup>(1)</sup>	IOPUWR <sup>(1)</sup>	—	RETEN <sup>(2)</sup>	—	DPSLP <sup>(1)</sup>	CM <sup>(1)</sup>	VREGS <sup>(3)</sup>
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1
EXTR <sup>(1)</sup>	SWR <sup>(1)</sup>	SWDTEN <sup>(4)</sup>	WDTO <sup>(1)</sup>	SLEEP <sup>(1)</sup>	IDLE <sup>(1)</sup>	BOR <sup>(1)</sup>	POR <sup>(1)</sup>
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      **TRAPR:** 陷阱复位标志位 <sup>(1)</sup>  
 1 = 发生了陷阱冲突复位  
 0 = 未发生陷阱冲突复位
- bit 14      **IOPUWR:** 非法操作码或访问未初始化的 W 寄存器的复位标志位 <sup>(1)</sup>  
 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的复位  
 0 = 未发生由非法操作码或未初始化的 W 寄存器而导致的复位
- bit 13      **未实现:** 读为 0
- bit 12      **RETEN:** 保持模式使能位 <sup>(2)</sup>  
 1 = 当器件处于休眠模式时, 使能保持模式 (1.2V 稳压器向内核供电)  
 0 = 禁止保持模式; 提供标称电平
- bit 11      **未实现:** 读为 0
- bit 10      **DPSLP:** 深度休眠标志位 <sup>(1)</sup>  
 1 = 器件处于深度休眠模式  
 0 = 器件不处于深度休眠模式
- bit 9        **CM:** 配置字失配复位标志位 <sup>(1)</sup>  
 1 = 发生了配置字失配复位  
 0 = 未发生配置字失配复位
- bit 8        **VREGS:** 休眠期间程序存储器上电位 <sup>(3)</sup>  
 1 = 休眠期间, 程序存储器偏置电压保持上电状态  
 0 = 休眠期间, 程序存储器偏置电压掉电
- bit 7        **EXTR:** 外部复位 (MCLR) 引脚位 <sup>(1)</sup>  
 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 <sup>(1)</sup>  
 1 = 执行了 RESET 指令  
 0 = 未执行 RESET 指令

- 注    1: 所有复位状态位都可以用软件置 1 或清零。用软件将任何一位置 1 不会导致器件复位。
- 2: 如果 LPCFG 配置位为 1 (未编程), 则禁止保持稳压器并且 RETEN 位不起作用。
- 3: 在稳压器进入待机模式后将其重新使能, 将在从休眠中唤醒时增加一个延时 TVREG。不使用稳压器的应用应该将此位置 1, 以防止发生此延时。
- 4: 如果 FWDTEN 配置位为 1 (未编程), 则 WDT 总是使能, 而与 SWDTEN 位的设置无关。

## 寄存器 7-1: RCON: 复位控制寄存器 (续)

bit 5	<b>SWDTEN:</b> 软件使能 / 禁止 WDT 位 <sup>(4)</sup> 1 = 使能 WDT 0 = 禁止 WDT
bit 4	<b>WDTO:</b> 看门狗定时器超时标志位 <sup>(1)</sup> 1 = 发生了 WDT 超时 0 = 未发生 WDT 超时
bit 3	<b>SLEEP:</b> 从休眠状态唤醒标志位 <sup>(1)</sup> 1 = 器件处于休眠模式 0 = 器件未处于休眠模式
bit 2	<b>IDLE:</b> 从空闲状态唤醒标志位 <sup>(1)</sup> 1 = 器件处于空闲模式 0 = 器件未处于空闲模式
bit 1	<b>BOR:</b> 欠压复位标志位 <sup>(1)</sup> 1 = 发生了欠压复位 (在上电复位后也被置 1) 0 = 未发生欠压复位
bit 0	<b>POR:</b> 上电复位标志位 <sup>(1)</sup> 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。用软件将任何一位置 1 不会导致器件复位。
  - 2: 如果 **LPCFG** 配置位为 1 (未编程), 则禁止保持稳压器并且 **RETEN** 位不起作用。
  - 3: 在稳压器进入待机模式后将其重新使能, 将在从休眠中唤醒时增加一个延时 **TVREG**。不使用稳压器的应用应该将此位置 1, 以防止发生此延时。
  - 4: 如果 **FWDTEN** 配置位为 1 (未编程), 则 WDT 总是使能, 而与 **SWDTEN** 位的设置无关。

# PIC24FJ128GB204 系列

寄存器 7-2: RCON2: 复位和系统控制寄存器 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-0	R/CO-1	R/CO-1	R/CO-1	R/CO-0
—	—	—	—	VDDBOR <sup>(1)</sup>	VDDPOR <sup>(1,2)</sup>	VBPOR <sup>(1,3)</sup>	VBAT <sup>(1)</sup>
bit 7							bit 0

<b>图注:</b>	CO = 只可清零位	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 15-5 **未实现:** 读为 0
- bit 4 **保留:** 保持为 0
- bit 3 **VDDBOR:** VDD 欠压复位标志位 <sup>(1)</sup>  
1 = 发生了 VDD 欠压复位 (由硬件置 1)  
0 = 未发生 VDD 欠压复位
- bit 2 **VDDPOR:** VDD 上电复位标志位 <sup>(1,2)</sup>  
1 = 发生了 VDD 上电复位 (由硬件置 1)  
0 = 未发生 VDD 上电复位
- bit 1 **VBPOR:** VBPOR 标志位 <sup>(1,3)</sup>  
1 = 发生了 VBAT POR (没有电池连接至 VBAT 引脚或 VBAT 电源低于深度休眠信号保持电平时, 由硬件置 1)  
0 = 未发生 VBAT POR
- bit 0 **VBAT:** VBAT 标志位 <sup>(1)</sup>  
1 = 当电源施加到 VBAT 引脚时, 发生 POR 退出 (由硬件置 1)  
0 = 未发生来自 VBAT 的 POR 退出

- 注**
- 1: 此位只能由硬件置 1, 由软件清零。
  - 2: 此位指示 VDD 上电复位。将 POR 位 (RCON<0>) 置 1 指示 V<sub>core</sub> 上电复位。
  - 3: 器件原先已上电时, 该位置 1, 即使 VBAT 上存在电源也是如此。

表 7-1: 复位标志位操作

标志位	置 1 事件	清零事件
TRAPR (RCON<15>)	陷阱冲突事件	POR
IOPUWR (RCON<14>)	非法操作码或访问未初始化的 W 寄存器	POR
CM (RCON<9>)	配置失配复位	POR
EXTR (RCON<7>)	MCLR 复位	POR
SWR (RCON<6>)	RESET 指令	POR
WDTO (RCON<4>)	看门狗定时器超时	CLRWDT、PWRSAV 指令和 POR
SLEEP (RCON<3>)	PWRSAV #0 指令	POR
DPSLP (RCON<10>)	DSEN 位置 1 时的 PWRSAV #0 指令	POR
IDLE (RCON<2>)	PWRSAV #1 指令	POR
BOR (RCON<1>)	POR 和 BOR	—
POR (RCON<0>)	POR	—

**注:** 所有复位标志位可由用户软件置 1 或清零。

## 7.1 特殊功能寄存器的复位状态

大部分与 PIC24F CPU 和外设有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各章节中有所说明。

除了四个寄存器之外, 其他 SFR 的复位值都不受复位类型的影响。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的类型和闪存配置字 2 (CW2) 中 FNOSC<sub>x</sub> 位的编程值 (见表 7-2)。RCFGCAL 和 NVMCON 寄存器只受 POR 影响。

## 7.2 器件复位时间

表 7-3 总结了各种类型的器件复位所需的时间。注意, 在 POR 延时结束后会发出主复位信号  $\overline{\text{SYSRST}}$ 。

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  $\overline{\text{SYSRST}}$  延时同时发生。

故障保护时钟监视器 (FSCM) 延时决定在  $\overline{\text{SYSRST}}$  信号发出后 FSCM 开始监视系统时钟源的时间。

## 7.3 欠压复位 (BOR)

PIC24FJ128GB204 系列器件实现了一个 BOR 电路, 它为用户提供几个配置和节能选项。BOR 通过 BOREN (CW3<12>) 配置位控制。

当使能 BOR 时, 只要  $V_{DD}$  下降至低于 BOR 阈值就会产生器件 BOR。第 33.1 节“直流特性”(参数 DC17A) 说明了阈值。

## 7.4 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可按照表 7-2 进行选择。如果禁止时钟切换, 则总是根据振荡器配置位选择系统时钟源。更多信息, 请参见《dsPIC33/PIC24 系列参考手册》中的“振荡器”(DS39700)。

表 7-2: 振荡器选择与复位类型的关系 (使能时钟切换)

复位类型	时钟源的确定
POR	FNOSC<2:0> 配置位 (CW2<10:8>)
BOR	
MCLR	COSC<2:0> 控制位 (OSCCON<14:12>)
WDTO	
SWR	

# PIC24FJ128GB204 系列

表 7-3: 各种器件复位的复位延时

复位类型	时钟源	$\overline{\text{SYSRST}}$ 延时	系统时钟延时	备注
POR	EC	$\text{TPOR} + \text{TSTARTUP} + \text{TRST}$	—	1, 2, 3
	ECPLL	$\text{TPOR} + \text{TSTARTUP} + \text{TRST}$	TLOCK	1, 2, 3, 5
	XT、HS 和 SOSC	$\text{TPOR} + \text{TSTARTUP} + \text{TRST}$	TOST	1, 2, 3, 4
	XTPLL 和 HSPLL	$\text{TPOR} + \text{TSTARTUP} + \text{TRST}$	$\text{TOST} + \text{TLOCK}$	1, 2, 3, 4, 5
	FRC 和 FRCDIV	$\text{TPOR} + \text{TSTARTUP} + \text{TRST}$	TFRC	1, 2, 3, 6, 7
	FRCPLL	$\text{TPOR} + \text{TSTARTUP} + \text{TRST}$	$\text{TFRC} + \text{TLOCK}$	1, 2, 3, 5, 6
	LPRC	$\text{TPOR} + \text{TSTARTUP} + \text{TRST}$	TLPRC	1, 2, 3, 6
BOR	EC	$\text{TSTARTUP} + \text{TRST}$	—	2, 3
	ECPLL	$\text{TSTARTUP} + \text{TRST}$	TLOCK	2, 3, 5
	XT、HS 和 SOSC	$\text{TSTARTUP} + \text{TRST}$	TOST	2, 3, 4
	XTPLL 和 HSPLL	$\text{TSTARTUP} + \text{TRST}$	$\text{TOST} + \text{TLOCK}$	2, 3, 4, 5
	FRC 和 FRCDIV	$\text{TSTARTUP} + \text{TRST}$	TFRC	2, 3, 6, 7
	FRCPLL	$\text{TSTARTUP} + \text{TRST}$	$\text{TFRC} + \text{TLOCK}$	2, 3, 5, 6
	LPRC	$\text{TSTARTUP} + \text{TRST}$	TLPRC	2, 3, 6
MCLR	任何时钟	TRST	—	3
WDT	任何时钟	TRST	—	3
软件	任何时钟	TRST	—	3
非法操作码	任何时钟	TRST	—	3
未初始化的 W 寄存器	任何时钟	TRST	—	3
陷阱冲突	任何时钟	TRST	—	3

- 注
- 1:  $\text{TPOR}$  = 上电复位延时 (标称值为  $10\ \mu\text{s}$ )。
  - 2:  $\text{TSTARTUP}$  =  $\text{TVREG}$ 。
  - 3:  $\text{TRST}$  = 内部状态复位时间 (标称值为  $2\ \mu\text{s}$ )。
  - 4:  $\text{TOST}$  = 振荡器起振定时器 (OST) 延时。10 位计数器计满 1024 个振荡器周期后, 才将振荡器时钟释放给系统使用。
  - 5:  $\text{TLOCK}$  = PLL 锁定时间。
  - 6:  $\text{TFRC}$  和  $\text{TLPRC}$  = RC 振荡器起振时间。
  - 7: 如果使能了双速启动, 无论选择了哪种主振荡器, 器件都将使用 FRC 启动, 因此, 系统时钟延时只是  $\text{TFRC}$ , 在这些情况下, FRC 起振时间有效。将在各自的时钟延时后切换到主振荡器。

## 7.4.1 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没有关系。某些晶振电路 (尤其是低频晶振) 的起振时间会相对较长。因此, 在发出  $\overline{\text{SYSRST}}$  之后, 可能会发生以下一种或多种情况:

- 振荡器电路未起振。
- 振荡器起振定时器尚未超时 (如果使用了晶振)。
- PLL 还未锁定 (如果使用了 PLL)。

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

## 7.4.2 故障保护时钟监视器 (FSCM) 和器件复位

如果使能了 FSCM, 它将在发出  $\overline{\text{SYSRST}}$  时开始监视系统时钟源。如果此时没有可用的有效时钟源, 器件将会自动切换到 FRC 振荡器, 且用户可在陷阱服务程序 (Trap Service Routine, TSR) 中将系统时钟源切换到所需的晶振。

## 8.0 中断控制器

**注：** 本数据手册总结了该 PIC24F 系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“中断”（DS70000600）。本数据手册中的信息取代了 FRM 中的信息。

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 的中断请求信号。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 个用户可选择的优先级
- 具有多达 118 个向量的中断向量表（IVT）
- 每个中断或异常源对应一个惟一的向量
- 指定的用户优先级内具有固定优先级
- 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

### 8.1 中断向量表

中断向量表（IVT）如图 8-1 所示。IVT 位于程序存储器中，起始单元地址为 000004h。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对应中断服务程序（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。

PIC24FJ128GB204 系列器件实现了不可屏蔽的陷阱和惟一中断。表 8-1 和表 8-2 对此做了总结。

#### 8.1.1 备用中断向量表

如图 8-1 所示，备用中断向量表（AIVT）位于 IVT 之后。ALTIVT（INTCON2<15>）控制位控制对 AIVT 的访问。如果 ALTIVT 位置 1，所有中断和异常处理将使用备用向量而不是默认的向量。备用向量与默认向量的构成方式相同。

AIVT 提供了一种不需要重新设定中断向量就可以在应用程序和支持环境之间切换的方法，以此支持仿真和调试功能。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，应该用 IVT 中使用的地址设置 AIVT。

### 8.2 复位过程

由于复位过程中不涉及到中断控制器，所以器件复位并不是真的异常。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，单片机从单元 000000h 开始执行程序。用户在复位地址中设置一条 GOTO 指令，该指令会使程序执行重新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地址设定 IVT 和 AIVT 中所有未实现或未使用的向量存储单元。

# PIC24FJ128GB204 系列

图 8-1: PIC24F 中断向量表

自然优先级降序排列 ↓	复位 - GOTO 指令	000000h	中断向量表 (IVT) <sup>(1)</sup>	
	复位 - GOTO 指令	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			
	算术错误陷阱向量			
	保留			
	保留			
	保留			
	中断向量 0	000014h		
	中断向量 1			
	—			
	—			
	—			
	中断向量 52	00007Ch		
	中断向量 53	00007Eh		
	中断向量 54	000080h		
	—			
	—			
	—			
	中断向量 116	0000FCh		
	中断向量 117	0000FEh		
	保留	000100h		备用中断向量表 (AIVT) <sup>(1)</sup>
	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				
算术错误陷阱向量				
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	0001FEh			
代码起始单元	000200h			

注 1: 有关中断向量列表的信息, 请参见表 8-2。

表 8-1: 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	算术错误
5	00000Eh	00010Eh	保留
6	000010h	000110h	保留
7	000012h	000112h	保留

# PIC24FJ128GB204 系列

表 8-2: 实现的中断向量

中断源	向量 编号	IRQ 编号	IVT 地址	AIVT 地址	中断位位置		
					标志	允许	优先级
ADC1 中断	21	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
比较器事件	26	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CRC 发生器	75	67	00009Ah	00019Ah	IFS4<3>	IEC4<3>	IPC16<14:12>
CTMU 事件	85	77	0000AEh	0001AEh	IFS4<13>	IEC4<13>	IPC19<6:4>
加密操作完成	63	55	000082h	000182h	IFS3<7>	IEC3<7>	IPC13<14:12>
加密密钥存储编程完成	64	56	000084h	000184h	IFS3<8>	IEC3<8>	IPC14<2:0>
加密缓冲区就绪	42	34	000058h	000158h	IFS2<2>	IEC2<2>	IPC8<10:8>
加密计满返回	43	35	00005Ah	00015Ah	IFS2<3>	IEC2<3>	IPC8<14:12>
DMA 通道 0	12	4	00001Ch	00011Ch	IFS0<4>	IEC0<4>	IPC1<2:0>
DMA 通道 1	22	14	000030h	000130h	IFS0<14>	IEC0<14>	IPC3<10:8>
DMA 通道 2	32	24	000044h	000144h	IFS1<8>	IEC1<8>	IPC6<2:0>
DMA 通道 3	44	36	00005Ch	00015Ch	IFS2<4>	IEC2<4>	IPC9<2:0>
DMA 通道 4	54	46	000070h	000170h	IFS2<14>	IEC2<14>	IPC11<10:8>
DMA 通道 5	69	61	00008Eh	00018Eh	IFS3<13>	IEC3<13>	IPC15<6:4>
外部中断 0	8	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
外部中断 1	28	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
外部中断 2	37	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
外部中断 3	61	53	00007Eh	00017Eh	IFS3<5>	IEC3<5>	IPC13<6:4>
外部中断 4	62	54	000080h	000180h	IFS3<6>	IEC3<6>	IPC13<10:8>
FRC 自调整	114	106	0000E8h	0001E8h	IFS6<10>	IEC6<10>	IPC26<10:8>
I2C1 主事件	25	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
I2C1 从事件	24	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
I2C1 总线冲突	92	84	0000BC	0001BC	IFS5<4>	IEC5<4>	IPC21<2:0>
I2C2 主事件	58	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
I2C2 从事件	57	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
I2C2 总线冲突	93	85	0000BE	0001BE	IFS5<5>	IEC5<5>	IPC21<6:4>
输入捕捉 1	9	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
输入捕捉 2	13	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
输入捕捉 3	45	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
输入捕捉 4	46	38	000060h	000160h	IFS2<6>	IEC2<6>	IPC9<10:8>
输入捕捉 5	47	39	000062h	000162h	IFS2<7>	IEC2<7>	IPC9<14:12>
输入捕捉 6	48	40	000064h	000164h	IFS2<8>	IEC2<8>	IPC10<2:0>
JTAG	125	117	0000FEh	0001FEh	IFS7<5>	IEC7<5>	IPC29<6:4>
输入电平变化通知 (ICN)	27	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
高/低电压检测 (HLVD)	80	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC18<2:0>
输出比较 1	10	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
输出比较 2	14	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
输出比较 3	33	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
输出比较 4	34	26	000048h	000148h	IFS1<10>	IEC1<10>	IPC6<10:8>
输出比较 5	49	41	000066h	000166h	IFS2<9>	IEC2<9>	IPC10<6:4>
输出比较 6	50	42	000068h	000168h	IFS2<10>	IEC2<10>	IPC10<10:8>
增强型并行主端口 (EPMP)	53	45	00006Eh	00016Eh	IFS2<13>	IEC2<13>	IPC11<6:4>
实时时钟和日历 (RTCC)	70	62	000090h	000190h	IFS3<14>	IEC3<14>	IPC15<10:8>

# PIC24FJ128GB204 系列

表 8-2: 实现的中断向量 (续)

中断源	向量 编号	IRQ 编号	IVT 地址	AIVT 地址	中断位位置		
					标志	允许	优先级
SPI1 通用	17	9	000026h	000126h	IFS0<9>	IEC0<9>	IPC2<6:4>
SPI1 发送	18	10	000028h	000128h	IFS0<10>	IEC0<10>	IPC2<10:8>
SPI1 接收	66	58	000088h	000188h	IFS3<10>	IEC3<10>	IPC14<10:8>
SPI2 通用	40	32	000054h	000154h	IFS2<0>	IEC2<0>	IPC8<2:0>
SPI2 发送	41	33	000056h	000156h	IFS2<1>	IEC2<1>	IPC8<6:4>
SPI2 接收	67	59	00008Ah	00018Ah	IFS3<11>	IEC3<11>	IPC14<14:12>
SPI3 通用	98	90	0000C8h	0001C8h	IFS5<10>	IEC5<10>	IPC22<10:8>
SPI3 发送	99	91	0000CAh	0001CAh	IFS5<11>	IEC5<11>	IPC22<14:12>
SPI3 接收	68	60	000054h	000154h	IFS3<12>	IEC3<12>	IPC15<2:0>
Timer1	11	3	00001Ah	00011Ah	IFS0<3>	IEC0<3>	IPC0<14:12>
Timer2	15	7	000022h	000122h	IFS0<7>	IEC0<7>	IPC1<14:12>
Timer3	16	8	000024h	000124h	IFS0<8>	IEC0<8>	IPC2<2:0>
Timer4	35	27	00004Ah	00014Ah	IFS1<11>	IEC1<11>	IPC6<14:12>
Timer5	36	28	00004Ch	00014Ch	IFS1<12>	IEC1<12>	IPC7<2:0>
UART1 错误	73	65	000096h	000196h	IFS4<1>	IEC4<1>	IPC16<6:4>
UART1 接收器	19	11	00002Ah	00012Ah	IFS0<11>	IEC0<11>	IPC2<14:12>
UART1 发送器	20	12	00002Ch	00012Ch	IFS0<12>	IEC0<12>	IPC3<2:0>
UART2 错误	74	66	000098h	000198h	IFS4<2>	IEC4<2>	IPC16<10:8>
UART2 接收器	38	30	000050h	000150h	IFS1<14>	IEC1<14>	IPC7<10:8>
UART2 发送器	39	31	000052h	000152h	IFS1<15>	IEC1<15>	IPC7<14:12>
UART3 错误	89	81	0000B6h	0001B6h	IFS5<1>	IEC5<1>	IPC20<6:4>
UART3 接收器	90	82	0000B8h	0001B8h	IFS5<2>	IEC5<2>	IPC20<10:8>
UART3 发送器	91	83	0000BAh	0001BAh	IFS5<3>	IEC5<3>	IPC20<14:12>
UART4 错误	95	87	0000C2h	0001C2h	IFS5<7>	IEC5<7>	IPC21<14:12>
UART4 接收器	96	88	0000C4h	0001C4h	IFS5<8>	IEC5<8>	IPC22<2:0>
UART4 发送器	97	89	0000C6h	0001C6h	IFS5<9>	IEC5<9>	IPC22<6:4>
USB	94	86	0000C0h	0001C0h	IFS5<6>	IEC5<6>	IPC21<10:8>

## 8.3 中断控制和状态寄存器

PIC24FJ128GB204 系列器件共实现了用于中断控制器的 43 个寄存器：

- INTCON1
- INTCON2
- IFS0 至 IFS7
- IEC0 至 IEC7
- IPC0 至 IPC16、IPC18 至 IPC22 以及 IPC26 和 IPC29
- INTTREG

INTCON1 和 INTCON2 具有全局中断控制功能。INTCON1 包含中断嵌套禁止 (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和备用中断向量表 (AIVT) 的使用。

IFSx 寄存器包含所有中断请求标志。每个中断源都有一个状态位，由各自的外设或外部信号置 1，且由软件清零。

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 (IPL)。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，它们被锁存到 INTTREG 寄存器中的向量编号 (VECNUM<7:0>) 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。

中断源按其在表 8-2 中的向量编号顺序分配给 IFSx、IECx 和 IPCx 寄存器。例如，INT0 (外部中断 0) 表示向量编号为 0，且自然优先级为 0。所以 INTOIF 状态位在 IFS0<0> 中，INT0IE 允许位在 IEC0<0> 中，INT0IP<2:0> 优先级位在 IPC0 的第一个位域 (IPC0<2:0>) 中。

尽管两个 CPU 控制寄存器都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>)。这些位指示当前 CPU 中断优先级。用户可以通过写 IPLx 位更改 CPU 的当前优先级。

CORCON 寄存器包含 IPL3 位，此位与 IPL<2:0> 位一起指示当前 CPU 优先级。IPL3 是一个只读位，所以用户软件不能屏蔽陷阱事件。

中断控制器具有显示其状态的中断控制器测试寄存器 INTTREG。发生中断请求时，其关联的向量编号和新的中断优先级被锁存到 INTTREG 中。当一个通用 ISR 用于多个向量 (如在自举程序中使用 ISR 重映射) 时，该信息可用于确定特定的中断源，或用于检查在 ISR 中是否有其他中断在等待处理。

下面各页中的寄存器 8-1 到寄存器 8-45 说明了所有的中断寄存器。

# PIC24FJ128GB204 系列

寄存器 8-1: **SR: ALU 状态寄存器 (CPU 中)**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	DC <sup>(1)</sup>
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 <sup>(2,3)</sup>	IPL1 <sup>(2,3)</sup>	IPL0 <sup>(2,3)</sup>	RA <sup>(1)</sup>	N <sup>(1)</sup>	OV <sup>(1)</sup>	Z <sup>(1)</sup>	C <sup>(1)</sup>
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 7-5            **IPL<2:0>:** CPU 中断优先级状态位 <sup>(2,3)</sup>

- 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其他位 (bit 0、1、2、3、4 和 8) 的描述, 请参见 [寄存器 3-1](#)。
- 注 2:** IPLx 位与 IPL3 位 (CORCON<3>) 共同决定 CPU 的中断优先级。如果 IPL3 = 1, 则括号中的值表示中断优先级。
- 注 3:** 当 NSTDIS (INTCON1<15>) = 1 时, IPLx 位是只读的。

# PIC24FJ128GB204 系列

**寄存器 8-2: CORCON: CPU 内核控制寄存器**

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-1	U-0	U-0
—	—	—	—	IPL3 <sup>(1)</sup>	—	—	—
bit 7				bit 0			

<b>图注:</b>	r = 保留位	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 <sup>(1)</sup>  
                  1 = CPU 中断优先级大于 7  
                  0 = CPU 中断优先级为 7 或更小
- bit 2        **保留:** 读为 PSV 位
- bit 1-0      **未实现:** 读为 0

**注 1:** IPL3 位与 IPL<2:0> 位 (SR<7:5>) 共同决定 CPU 中断优先级; 关于此位的说明, 请参见[寄存器 3-2](#)。

# PIC24FJ128GB204 系列

寄存器 8-3: INTCON1: 中断控制寄存器 1

R/W-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 = 允许中断嵌套
- 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

# PIC24FJ128GB204 系列

## 寄存器 8-4: INTCON2: 中断控制寄存器 2

R/W-0	R-0, HSC	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

<b>图注:</b>	HSC = 可硬件置 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     **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) 中断向量表
- bit 14     **DISI:** DISI 指令状态位  
           1 = DISI 指令有效  
           0 = DISI 指令无效
- 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

# PIC24FJ128GB204 系列

## 寄存器 8-5: IFS0: 中断标志状态寄存器 0

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	DMA1IF	AD1IF	U1TXIF	U1RXIF	SPI1TXIF	SPI1IF	T3IF
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
T2IF	OC2IF	IC2IF	DMA0IF	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            **未实现:** 读为 0
- bit 14            **DMA1IF:** DMA 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 13            **AD1IF:** ADC1 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 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            **SPI1TXIF:** SPI1 发送中断标志状态位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 9             **SPI1IF:** SPI1 通用中断标志状态位  
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             **DMA0IF:** DMA 通道 0 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             **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

**寄存器 8-5: IFS0: 中断标志状态寄存器 0 (续)**

- 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

# PIC24FJ128GB204 系列

## 寄存器 8-6: 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	R/W-0
U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	DMA2IF
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
—	—	—	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         **DMA2IF:** DMA 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 7-5       **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

**寄存器 8-6: IFS1: 中断标志状态寄存器 1 (续)**

- 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:** I2C1 主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:** I2C1 从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

# PIC24FJ128GB204 系列

## 寄存器 8-7: IFS2: 中断标志状态寄存器 2

U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	DMA4IF	PMPIF	—	—	OC6IF	OC5IF	IC6IF
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
IC5IF	IC4IF	IC3IF	DMA3IF	CRYROLLIF	CRYFREEIF	SPI2TXIF	SPI2IF
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        **DMA4IF:** DMA 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 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 12-11    **未实现:** 读为 0
- 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         **DMA3IF:** DMA 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 3         **CRYROLLIF:** 加密计满返回状态位  
               1 = 产生了中断请求  
               0 = 未产生中断请求
- bit 2         **CRYFREEIF:** 加密缓冲区空闲状态位  
               1 = 产生了中断请求  
               0 = 未产生中断请求

**寄存器 8-7: IFS2: 中断标志状态寄存器 2 (续)**

bit 1 **SPI2TXIF:** SPI2 发送中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 0 **SPI2IF:** SPI2 通用中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

# PIC24FJ128GB204 系列

## 寄存器 8-8: IFS3: 中断标志状态寄存器 3

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
—	RTCIF	DMA5IF	SPI3RXIF	SPI2RXIF	SPI1RXIF	—	KEYSTRIF
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
CRYDNIF	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:** 实时时钟和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      **DMA5IF:** DMA 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 12      **SPI3RXIF:** SPI3 接收中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 11      **SPI2RXIF:** SPI2 接收中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 10      **SPI1RXIF:** SPI1 接收中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 9        **未实现:** 读为 0
- bit 8        **KEYSTRIF:** 加密密钥存储编程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 7        **CRYDNIF:** 加密操作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 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

**寄存器 8-8: IFS3: 中断标志状态寄存器 3 (续)**

bit 5	<b>INT3IF:</b> 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	<b>未实现:</b> 读为 0
bit 2	<b>MI2C2IF:</b> I2C2 主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	<b>SI2C2IF:</b> I2C2 从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	<b>未实现:</b> 读为 0

# PIC24FJ128GB204 系列

## 寄存器 8-9: 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	—	—	—	—	HLVDIF
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        **HLVDIF:** 高 / 低电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

# PIC24FJ128GB204 系列

## 寄存器 8-10: IFS5: 中断标志状态寄存器 5

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	SPI3TXIF	SPI3IF	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	I2C2BCIF	I2C1BCIF	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 15-12   **未实现:** 读为 0
- bit 11     **SPI3TXIF:** SPI3 发送中断标志状态位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 10     **SPI3IF:** SPI3 通用中断标志状态位  
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      **I2C2BCIF:** I2C2 总线冲突中断标志状态位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 4      **I2C1BCIF:** I2C1 总线冲突中断标志状态位  
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

# PIC24FJ128GB204 系列

## 寄存器 8-11: IFS6: 中断标志状态寄存器 6

U-0	U-0	U-0	U-0	U-0	R/W-0	U-0	U-0
—	—	—	—	—	FSTIF	—	—
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        **FSTIF:** FRC 自调整中断标志状态位  
                   1 = 产生了中断请求  
                   0 = 未产生中断请求  
 bit 9-0      **未实现:** 读为 0

## 寄存器 8-12: IFS7: 中断标志状态寄存器 7

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	U-0	U-0	U-0	U-0	U-0
—	—	JTAGIF	—	—	—	—	—
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        **JTAGIF:** JTAG 控制器中断标志状态位  
                   1 = 产生了中断请求  
                   0 = 未产生中断请求  
 bit 4-0      **未实现:** 读为 0

# PIC24FJ128GB204 系列

## 寄存器 8-13: IEC0: 中断允许控制寄存器 0

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	DMA1IE	AD1IE	U1TXIE	U1RXIE	SPI1TXIE	SPI1IE	T3IE
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
T2IE	OC2IE	IC2IE	DMA0IE	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     **未实现:** 读为 0
- bit 14     **DMA1IE:** DMA 通道 1 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 13     **AD1IE:** ADC1 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     **SPI1TXIE:** SPI1 发送完成中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 9       **SPI1IE:** SPI1 通用中断允许位  
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       **DMA0IE:** DMA 通道 0 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 3       **T1IE:** Timer1 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求

# PIC24FJ128GB204 系列

---

## 寄存器 8-13: IEC0: 中断允许控制寄存器 0 (续)

- 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

# PIC24FJ128GB204 系列

寄存器 8-14: 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	R/W-0
U2TXIE	U2RXIE	INT2IE <sup>(1)</sup>	T5IE	T4IE	OC4IE	OC3IE	DMA2IE
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
—	—	—	INT1IE <sup>(1)</sup>	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 <sup>(1)</sup>  
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        **DMA2IE:** DMA 通道 2 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 7-5      **未实现:** 读为 0
- bit 4        **INT1IE:** 外部中断 1 允许位 <sup>(1)</sup>  
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, 则此中断输入还必须配置给可用的 RPN 或 RPIIn 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。

# PIC24FJ128GB204 系列

---

## 寄存器 8-14: IEC1: 中断允许控制寄存器 1 (续)

bit 1 **MI2C1IE:** I2C1 主事件中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 0 **SI2C1IE:** I2C1 从事件中断允许位

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，则此中断输入还必须配置给可用的 RPN 或 RPln 引脚。更多信息，请参见第 11.4 节“外设引脚选择 (PPS)”。

# PIC24FJ128GB204 系列

## 寄存器 8-15: IEC2: 中断允许控制寄存器 2

U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	DMA4IE	PMPIE	—	—	OC6IE	OC5IE	IC6IE
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
IC5IE	IC4IE	IC3IE	DMA3IE	CRYROLLIE	CRYFREEIE	SPI2TXIE	SPI2IE
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      **DMA4IE:** DMA 通道 4 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- 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 12-11   **未实现:** 读为 0
- 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        **DMA3IE:** DMA 通道 3 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 3        **CRYROLLIE:** 加密计满返回中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 2        **CRYFREEIE:** 加密缓冲区空闲中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求

# PIC24FJ128GB204 系列

---

## 寄存器 8-15: IEC2: 中断允许控制寄存器 2 (续)

- bit 1      **SPI2TXIE:** SPI2 发送中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 0      **SPI2IE:** SPI2 通用中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求

# PIC24FJ128GB204 系列

寄存器 8-16: IEC3: 中断允许控制寄存器 3

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
—	RTCIE	DMA5IE	SPI3RXIE	SPI2RXIE	SPI1RXIE	—	KEYSTRIE
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
CRYDNIE	INT4IE <sup>(1)</sup>	INT3IE <sup>(1)</sup>	—	—	MI2C2IE	SI2C2IE	—
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 **RTCIE:** 实时时钟和日历中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 13 **DMA5IE:** DMA 通道 5 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 12 **SPI3RXIE:** SPI3 接收中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 11 **SPI2RXIE:** SPI2 接收中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 10 **SPI1RXIE:** SPI1 接收中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 9 **未实现:** 读为 0
- bit 8 **KEYSTRIE:** 加密密钥存储编程完成中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 7 **CRYDNIE:** 加密操作完成中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 6 **INT4IE:** 外部中断 4 允许位 <sup>(1)</sup>  
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 <sup>(1)</sup>  
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:** I2C2 主事件中断允许位  
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, 则此中断输入还必须配置给可用的 RPN 或 RPIIn 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。

# PIC24FJ128GB204 系列

---

## 寄存器 8-16: IEC3: 中断允许控制寄存器 3 (续)

bit 1      **SI2C2IE:** I2C2 从事件中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 0      **未实现:** 读为 0

**注 1:** 如果允许了外部中断，则此中断输入还必须配置给可用的 RPN 或 RPln 引脚。更多信息，请参见第 11.4 节“**外设引脚选择 (PPS)**”。

# PIC24FJ128GB204 系列

## 寄存器 8-17: 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	—	—	—	—	HLVDIE
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        **HLVDIE:** 高 / 低电压检测中断允许位  
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

# PIC24FJ128GB204 系列

## 寄存器 8-18: IEC5: 中断允许控制寄存器 5

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	SPI3TXIE	SPI3IE	U4TXIE	U4RXIE
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
U4ERIE	USB1IE	I2C2BCIE	I2C1BCIE	U3TXIE	U3RXIE	U3ERIE	—
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 15-12      **未实现:** 读为 0
- bit 11        **SPI3TXIE:** SPI3 发送中断允许位  
               1 = 允许中断请求  
               0 = 禁止中断请求
- bit 10        **SPI3IE:** SPI3 通用中断允许位  
               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         **I2C2BCIE:** I2C2 总线冲突中断允许位  
               1 = 允许中断请求  
               0 = 禁止中断请求
- bit 4         **I2C1BCIE:** I2C1 总线冲突中断允许位  
               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

# PIC24FJ128GB204 系列

## 寄存器 8-19: IEC6: 中断允许控制寄存器 6

U-0	U-0	U-0	U-0	U-0	R/W-0	U-0	U-0
—	—	—	—	—	FSTIE	—	—
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 **FSTIE**: FRC 自调整中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 9-0 未实现: 读为 0

## 寄存器 8-20: IEC7: 中断允许控制寄存器 7

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	U-0	U-0	U-0	U-0	U-0
—	—	JTAGIE	—	—	—	—	—
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 **JTAGIE**: JTAG 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 4-0 未实现: 读为 0

# PIC24FJ128GB204 系列

## 寄存器 8-21: 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

# PIC24FJ128GB204 系列

## 寄存器 8-22: 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	R/W-1	R/W-0	R/W-0
—	IC2IP2	IC2IP1	IC2IP0	—	DMA0IP2	DMA0IP1	DMA0IP0
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 未实现: 读为 0

bit 2-0 **DMA0IP<2:0>**: DMA 通道 0 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

# PIC24FJ128GB204 系列

## 寄存器 8-23: 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	—	SPI1TXIP2	SPI1TXIP1	SPI1TXIP0
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
—	SPI1IP2	SPI1IP1	SPI1IP0	—	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        **SPI1TXIP<2:0>:** SPI1 发送中断优先级位
  - 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         **SPI1IP<2:0>:** SPI1 通用中断优先级位
  - 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

# PIC24FJ128GB204 系列

## 寄存器 8-24: IPC3: 中断优先级控制寄存器 3

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	DMA1IP2	DMA1IP1	DMA1IP0
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 15-11 **未实现:** 读为 0

bit 10-8 **DMA1IP<2:0>:** DMA 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 **AD1IP<2:0>:** ADC1 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

# PIC24FJ128GB204 系列

寄存器 8-25:      **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						bit 8	

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0
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    **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    **MI2C1IP<2:0>:** I2C1 主事件中断优先级位
  - 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    **SI2C1IP<2:0>:** I2C1 从事件中断优先级位
  - 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

# PIC24FJ128GB204 系列

寄存器 8-26: IPC5: 中断优先级控制寄存器 5

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
—	—	—	—	—	INT1IP<2:0>		
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 15-3     **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

# PIC24FJ128GB204 系列

## 寄存器 8-27: 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	R/W-1	R/W-0	R/W-0
—	OC3IP2	OC3IP1	OC3IP0	—	DMA2IP2	DMA2IP1	DMA2IP0
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       **未实现:** 读为 0

bit 2-0    **DMA2IP<2:0>:** DMA 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

# PIC24FJ128GB204 系列

## 寄存器 8-28: 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

# PIC24FJ128GB204 系列

## 寄存器 8-29: IPC8: 中断优先级控制寄存器 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CRYROLLIP2	CRYROLLIP1	CRYROLLIP0	—	CRYFREEIP2	CRYFREEIP1	CRYFREEIP0
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
—	SPI2TXIP2	SPI2TXIP1	SPI2TXIP0	—	SPI2IP2	SPI2IP1	SPI2IP0
bit 7							bit 0

<b>图注:</b>							
R = 可读位	W = 可写位	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    **CRYROLLIP<2:0>:** 加密计满返回中断优先级位
  - 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    **CRYFREEIP<2:0>:** 加密缓冲区空闲中断优先级位
  - 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    **SPI2TXIP<2:0>:** SPI2 发送中断优先级位
  - 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    **SPI2IP<2:0>:** SPI2 通用中断优先级位
  - 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

# PIC24FJ128GB204 系列

## 寄存器 8-30: 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	R/W-1	R/W-0	R/W-0
—	IC3IP2	IC3IP1	IC3IP0	—	DMA3IP2	DMA3IP1	DMA3IP0
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       **未实现:** 读为 0
- bit 2-0     **DMA3IP<2:0>:** DMA 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



# PIC24FJ128GB204 系列

## 寄存器 8-32: IPC11: 中断优先级控制寄存器 11

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	DMA4IP2	DMA4IP1	DMA4IP0
bit 15					bit 8		

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	PMPIP2	PMPIP1	PMPIP0	—	—	—	—
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     **DMA4IP<2:0>:** DMA 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     **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 3-0     **未实现:** 读为 0

# PIC24FJ128GB204 系列

寄存器 8-33: 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
—	—	—	—	—	MI2C2IP2	MI2C2IP1	MI2C2IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	SI2C2IP2	SI2C2IP1	SI2C2IP0	—	—	—	—
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      **MI2C2IP<2:0>:** I2C2 主事件中断优先级位
  - 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      **SI2C2IP<2:0>:** I2C2 从事件中断优先级位
  - 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

# PIC24FJ128GB204 系列

## 寄存器 8-34: IPC13: 中断优先级控制寄存器 13

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CRYDNIP2	CRYDNIP1	CRYDNIP0	—	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      **未实现:** 读为 0
- bit 14-12   **CRYDNIP<2:0>:** 加密操作完成中断优先级位  
             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 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

# PIC24FJ128GB204 系列

寄存器 8-35: IPC14: 中断优先级控制寄存器 14

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPI2RXIP2	SPI2RXIP1	SPI2RXIPO	—	SPI1RXIP2	SPI1RXIP1	SPI1RXIPO
bit 15				bit 8			

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	KEYSTRIP2	KEYSTRIP1	KEYSTRIP0
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       **SPI2RXIP<2:0>:** SPI2 接收中断优先级位
  - 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        **SPI1RXIP<2:0>:** SPI1 接收中断优先级位
  - 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         **KEYSTRIP<2:0>:** 加密密钥存储编程完成中断优先级位
  - 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

# PIC24FJ128GB204 系列

## 寄存器 8-36: 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	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	DMA5IP2	DMA5IP1	DMA5IP0	—	SPI3RXIP2	SPI3RXIP1	SPI3RXIP0
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 **未实现:** 读为 0

bit 6-4 **DMA5IP<2:0>:** DMA 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 **SPI3RXIP<2:0>:** SPI3 接收中断优先级位

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

# PIC24FJ128GB204 系列

## 寄存器 8-37: 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

# PIC24FJ128GB204 系列

**寄存器 8-38: 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
—	—	—	—	—	HLVDIP<2:0>		
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 15-3      **未实现:** 读为 0  
 bit 2-0      **HLVDIP<2:0>:** 高 / 低电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 8-39: 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
—	CTMUIP<2:0>			—	—	—	—
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 15-7      **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

# PIC24FJ128GB204 系列

## 寄存器 8-40: 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

# PIC24FJ128GB204 系列

## 寄存器 8-41: 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
—	I2C2BCIP2	I2C2BCIP1	I2C2BCIP0	—	I2C1BCIP2	I2C1BCIP1	I2C1BCIP0
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     **I2C2BCIP<2:0>:** I2C2 总线冲突中断优先级位  
             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     **I2C1BCIP<2:0>:** I2C1 总线冲突中断优先级位  
             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

# PIC24FJ128GB204 系列

寄存器 8-42: **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
—	SPI3TXIP2	SPI3TXIP1	SPI3TXIP0	—	SPI3IP2	SPI3IP1	SPI3IP0
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   **SPI3TXIP<2:0>:** SPI3 发送中断优先级位
  - 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    **SPI3IP<2:0>:** SPI3 通用中断优先级位
  - 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

# PIC24FJ128GB204 系列

## 寄存器 8-43: IPC26: 中断优先级控制寄存器 26

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	FSTIP<2:0>		
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    **FSTIP<2:0>**: FRC 自调整中断优先级位  
               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

## 寄存器 8-44: IPC29: 中断优先级控制寄存器 29

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15					bit 8		
U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	JTAGIP<2:0>			—	—	—	—
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 15-7    未实现: 读为 0  
 bit 6-4    **JTAGIP<2:0>**: JTAG 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

# PIC24FJ128GB204 系列

## 寄存器 8-45: INTTREG: 中断控制器测试寄存器

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

R/W-0							
VECNUM7	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0
bit 7							bit 0

<b>图注:</b>	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      **CPUIRQ:** 来自中断控制器的 CPU 中断请求位  
 1 = 产生了中断请求, 但 CPU 尚未应答  
 0 = 未应答任何中断请求
- bit 14      **保留:** 保持为 0
- bit 13      **VHOLD:** 向量编号捕捉配置位  
 1 = VECNUM<7:0> 位保存优先级最高的待处理中断的值  
 0 = VECNUM<7:0> 位保存上一次应答的中断值 (即已发生且优先级高于 CPU 的上一个中断, 即使有其他待处理中断时也是如此)
- 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  
 0001 = CPU 中断优先级为 0
- bit 7-0     **VECNUM<7:0>:** 待处理中断或上一次应答的中断的向量编号位  
当 VHOLD = 1 时:  
 指示上一次发生的的中断的向量编号 (0-118)  
当 VHOLD = 0 时:  
 指示当前处理的中断请求的向量编号 (0-118)。

## 8.4 中断设置过程

### 8.4.1 初始化

按以下步骤配置中断源：

1. 如果不需要嵌套中断，则将 `NSTDIS` 控制位 (`INTCON1<15>`) 置 1。
2. 通过写相应的 `IPCx` 寄存器中的控制位来为中断源选择用户分配的优先级。优先级将取决于具体的应用和中断源的类型。如果不需要多个优先级，可以将所有允许的中断源的 `IPCx` 寄存器控制位编程为相同的非零值。

**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。

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。

### 8.4.2 中断服务程序 (ISR)

用于声明 `ISR` 以及使用正确的向量地址初始化 `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。一般情况下，用户必须将相应 `IFSx` 寄存器中与 `ISR` 处理的中断源相对应的中断标志清零。否则，在退出程序后会立即再次进入 `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。

### 8.4.3 陷阱服务程序 (TSR)

陷阱服务程序 (`Trap Service Routine, TSR`) 的编码方式类似于 `ISR`，只是必须将 `INTCON1` 寄存器中相应的陷阱状态标志清零，以避免重新进入 `TSR`。

### 8.4.4 禁止中断

可以通过以下步骤禁止所有用户中断：

1. 使用 `PUSH` 指令将当前 `SR` 值压入软件堆栈。
2. 通过将值 `0Eh` 与 `SRL` 进行逻辑或运算来强制把 `CPU` 的优先级设置为 7。

要允许用户中断，可以使用 `POP` 指令恢复先前的 `SR` 值。

注意只能禁止优先级小于或等于 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。

# PIC24FJ128GB204 系列

---

注:

## 9.0 振荡器配置

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的第6章“振荡器”（DS39700）。

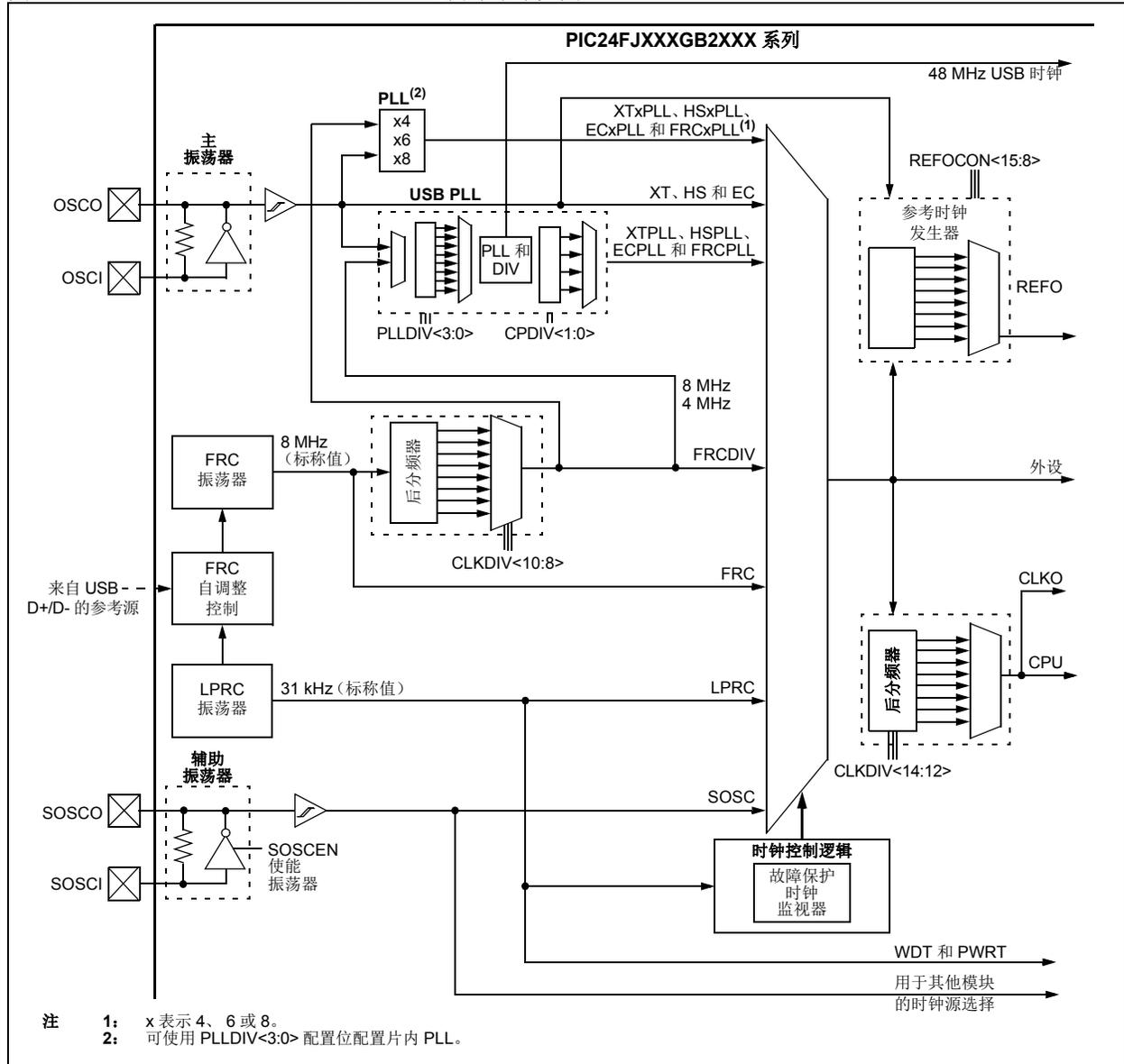
PIC24FJ128GB204 系列器件的振荡器系统具有以下特性：

- 共有四个外部和内部振荡器可选作为时钟源，提供 15 种不同的时钟模式
- 一个片内 USB PLL 模块，用于为 USB 模块提供稳定的 48 MHz 时钟以及为系统时钟提供一系列频率选择

- 一个片内 PLL（x4、x6 和 x8）模块，可用于主振荡器（Primary Oscillator，POSC）源或 FRCDIV（见第 9.8 节“片内 PLL”）。
- 可通过软件控制在多个时钟源之间切换
- 可通过软件控制后分频器有选择地为 CPU 选择时钟，以节省系统功耗
- 具有故障保护时钟监视器（FSCM），可检测时钟故障，以使应用可安全地恢复或关闭
- 可单独使用且独立配置的系统时钟输出，以同步外部硬件

图 9-1 所示为振荡器系统的简化框图。

图 9-1: PIC24FJ128GB204 系列时钟框图



# PIC24FJ128GB204 系列

## 9.1 CPU 时钟机制

系统时钟源可由以下四种时钟源之一提供：

- OSCI 和 OSCO 引脚上的主振荡器（Primary Oscillator, POSC）
- SOSCI 和 SOSCO 引脚上的辅助振荡器（Secondary Oscillator, 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 源可以使用内部 USB PLL 模块，用它从 96 MHz PLL 提供 USB 模块的时钟和单独的系统时钟。更多信息，请参见第 9.6 节“振荡器模式和 USB 工作”。

内部 FRC 提供 8 MHz 的时钟源。可选择用可编程时钟分频器降低其频率，从而提供一系列系统时钟频率。

选择的时钟源提供处理器和外设的时钟。将处理器时钟源二分频可以产生内部指令周期时钟 Fcy。在本文档中，指令周期时钟也由 Fosc/2 表示。在 OSCO I/O 引脚上可以提供内部指令周期时钟 Fosc/2，用于主振荡器的某些工作模式。

## 9.2 上电复位时的初始配置

通过使用配置位设置选择器件发生上电复位事件时使用的振荡器源（和工作模式）。振荡器配置位设置位于程序存储器中的配置寄存器中（更多详细信息，请参见第 30.1 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>）用于选择上电复位时使用的振荡器源。带后分频器的 FRC 主振荡器（FRCDIV）是默认（未编程）的选择。对这些位所在的存储单元进行编程可以选择辅助振荡器或其中一个内部振荡器。

配置位允许用户在多个时钟模式中进行选择，如表 9-1 所示。

### 9.2.1 时钟切换模式配置位

FCKSM<1:0> 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 编程为 0 时才可以使能时钟切换功能。只有将 FCKSM<1:0> 编程为 00 时才可以使能 FSCM。

表 9-1: 用于时钟选择的配置位值

振荡器模式	振荡源	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。

注 2: 对于未编程（已擦除）器件，这是默认的振荡器模式。

## 9.3 控制寄存器

振荡器的操作由三个特殊功能寄存器控制：

- OSCCON
- CLKDIV
- OSCTUN

OSCCON 寄存器（寄存器 9-1）是振荡器的主要控制寄存器。它控制时钟源的切换并允许监视时钟源。CLKDIV 寄存器（寄存器 9-2）控制打盹模式的相关特

性，以及 FRC 振荡器的后分频器。

OSCTUN 寄存器（寄存器 9-3）允许用户在约  $\pm 1.5\%$  的范围内微调 FRC 振荡器。该寄存器还控制 FRC 自调整功能，如第 9.5 节“FRC 自调节”所述。

**寄存器 9-1: OSCCON: 振荡器控制寄存器**

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

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

<b>图注:</b>	CO = 只可清零位	SO = 只能置 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 **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<sup>(1)</sup>

- 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<sub>x</sub> 配置位决定。
  - 2: 只能在执行解锁序列后更改 IOLOCK 位的状态。另外，如果 IOL1WAY 配置位为 1，一旦 IOLOCK 位置 1，它就不能清零。
  - 3: 在进行有效的时钟切换或选择非 PLL 时钟模式时，该位也复位为 0。

# PIC24FJ128GB204 系列

## 寄存器 9-1: OSCCON: 振荡器控制寄存器 (续)

bit 7	<b>CLKLOCK:</b> 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, 可通过将 OSWEN 位置 1 来进行修改。 <u>如果禁止 FSCM (FCKSM1 = 0):</u> 时钟和 PLL 选择始终未锁定, 可通过将 OSWEN 位置 1 来进行修改。
bit 6	<b>IOLOCK:</b> I/O 锁定使能位 <sup>(2)</sup> 1 = I/O 锁定有效 0 = I/O 锁定无效
bit 5	<b>LOCK:</b> PLL 锁定状态位 <sup>(3)</sup> 1 = PLL 模块处于锁定状态或 PLL 模块的起振定时器延时结束 0 = PLL 模块不处于锁定状态, PLL 起振定时器正在运行或 PLL 被禁止
bit 4	<b>未实现:</b> 读为 0
bit 3	<b>CF:</b> 时钟故障检测位 1 = FSCM 检测到一个时钟故障 0 = 未检测到时钟故障
bit 2	<b>POSCEN:</b> 主振荡器 (POSC) 休眠使能位 1 = 主振荡器在休眠模式期间继续工作 0 = 主振荡器在休眠模式期间被禁止
bit 1	<b>SOSCEN:</b> 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	<b>OSWEN:</b> 振荡器切换使能位 1 = 启动振荡器切换, 切换到由 NOSC<2:0> 位指定的时钟源 0 = 完成振荡器切换

- 注
- 1: 这些位的复位值由 FNOSC<sub>x</sub> 配置位决定。
  - 2: 只能在执行解锁序列后更改 IOLOCK 位的状态。另外, 如果 IOL1WAY 配置位为 1, 一旦 IOLOCK 位置 1, 它就不能清零。
  - 3: 在进行有效的时钟切换或选择非 PLL 时钟模式时, 该位也复位为 0。

# PIC24FJ128GB204 系列

寄存器 9-2: 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 <sup>(1)</sup>	RCDIV2	RCDIV1	RCDIV0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
CPDIV1	CPDIV0	PLLEN	—	—	—	—	—
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:** 中断时恢复位  
 1 = 中断时清零 DOZEN 位并将 CPU/ 外设的时钟比复位为 1:1  
 0 = 中断不影响 DOZEN 位
- bit 14-12    **DOZE<2:0>:** CPU/ 外设的时钟比选择位  
 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<sup>(1)</sup>  
 1 = DOZE<2:0> 位指定 CPU/ 外设时钟比  
 0 = 将 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 (不分频)
- 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)<sup>(2)</sup>  
 10 = 8 MHz (4 分频)<sup>(2)</sup>  
 01 = 16 MHz (2 分频)  
 00 = 32 MHz (不分频)
- bit 5        **PLLEN:** PLL 使能位  
 1 = 使能 PLL  
 0 = 禁止 PLL
- bit 4-0     **未实现:** 读为 0

- 注    1: 当 ROI 位置 1 并发生中断时, 该位会自动清零。  
 2: 使能 USB 模块时禁止此设置。

# PIC24FJ128GB204 系列

寄存器 9-3: OSCTUN: FRC 振荡器调节寄存器

R/W-0	U-0	R/W-0	R/W-0	R-0	R/W-0	R-0	R/W-0
STEN	—	STSIDL	STSRC <sup>(1)</sup>	STLOCK	STLPOL	STOR	STORPOL
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	TUN4	TUN3	TUN2	TUN1	TUN0
bit 7							bit 0

**图注:**

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

- bit 15        **STEN:** FRC 自调节使能位  
 1 = 使能 FRC 自调节; TUNx 位由硬件控制  
 0 = 禁止 FRC 自调节; 应用程序可选择控制 TUNx 位
- bit 14        **未实现:** 读为 0
- bit 13        **STSIDL:** FRC 空闲模式自调节停止位  
 1 = 空闲模式下自调节停止  
 0 = 空闲模式下自调节继续
- bit 12        **STSRC:** FRC 自调节参考时钟源位 <sup>(1)</sup>  
 1 = FRC 调节为约匹配 USB 主时钟容差  
 0 = FRC 调节为约匹配 32.768 kHz SOSC 容差
- bit 11        **STLOCK:** FRC 自调节锁定状态位  
 1 = FRC 精度当前位于 STSRC 参考精度的  $\pm 0.2\%$  内  
 0 = FRC 精度可能不位于 STSRC 参考精度的  $\pm 0.2\%$  内
- bit 10        **STLPOL:** FRC 自调节锁定中断优先级位  
 1 = STLOCK = 0 时产生自调节锁定中断  
 0 = STLOCK = 1 时产生自调节锁定中断
- bit 9         **STOR:** FRC 自调节超出范围状态位  
 1 = STSRC 参考时钟错误超出 TUN<5:0> 的范围; 不执行任何调节  
 0 = STSRC 参考时钟在调节范围内; 执行调节
- bit 8         **STORPOL:** FRC 自调节超出范围中断优先级位  
 1 = STOR = 0 时产生自调节超出范围中断  
 0 = STOR = 1 时产生自调节超出范围中断
- bit 7-6       **未实现:** 读为 0
- bit 5-0       **TUN<5:0>:** FRC 振荡器调节位  
 011111 = 最大频率偏移  
 011110 =  
 ...  
 000001 =  
 000000 = 中心频率, 振荡器以出厂校准频率运行  
 111111 =  
 ...  
 100001 =  
 100000 = 最小频率偏移

注 1: 任一时钟恢复源的使用具有特定的应用要求。更多信息, 请参见第 9.5 节“FRC 自调节”。

## 9.4 时钟切换工作原理

应用可在软件控制下随时在四个时钟源（POSC、SOSC、FRC 和 LPRC）间自由切换，几乎没有什么限制。为限制该灵活性可能带来的负面影响，PIC24F 器件在时钟切换过程中采用了安全锁定。

**注：** 主振荡器模式有三种不同的子模式（XT、HS 和 EC），这三个子模式由 POSCMDx 配置位决定。在应用中可以用软件实现从主振荡器模式切换到其他模式，或从其他模式切换到主振荡器模式，但不能在不对器件进行再编程的情况下在主振荡器模式的不同子模式之间进行切换。

### 9.4.1 使能时钟切换

要使能时钟切换，必须将 CW2 的 FCKSM1 配置位编程为 0。（更多详细信息，请参见第 30.1 节“配置位”。）如果没有对 FCKSM1 配置位编程（即 FCKSM1 配置位为 1），则时钟切换功能和故障保护时钟监视器功能都将被禁止；这是默认设置。

在时钟切换被禁止的情况下，NOSC<sub>x</sub> 控制位（OSCCON<10:8>）不控制时钟选择。但是，COSC<2:0>位（OSCCON<14:12>）将会反映FNOSC<sub>x</sub> 配置位选择的时钟源。

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

### 9.4.2 振荡器切换序列

执行时钟切换至少需要下列基本序列：

1. 如果需要，读 COSC<2:0> 位（OSCCON<14:12>）确定当前振荡器源。
2. 执行解锁序列以允许写入 OSCCON 寄存器的高字节。
3. 将新振荡器源的相关值写入 NOSC<sub>x</sub> 位（OSCCON<10:8>）。
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 来启动振荡器切换。

一旦基本序列完成，系统时钟硬件将自动进行如下响应：

1. 时钟切换硬件将NOSC<sub>x</sub>控制位的新值和COSC<sub>x</sub>位做比较。如果相等，时钟切换为冗余操作。在这种情况下，OSWEN 位被自动清零且时钟切换被中止。
2. 如果启动了有效时钟切换，则LOCK（OSCCON<5>）和CF（OSCCON<3>）位被清零。
3. 如果新振荡器现在不运行，硬件会将其开启。如果必须打开晶振，硬件将等待直至OST 超时。如果新振荡源正在使用PLL，硬件将等待直到检测到PLL 锁定（LOCK = 1）。
4. 硬件会等待新时钟源的 10 个时钟周期，然后执行时钟切换。
5. 硬件清零OSWEN位以表示时钟切换成功。此外，NOSC<sub>x</sub> 位的值被传送到COSC<sub>x</sub> 位中。
6. 此时旧时钟源被关闭，但LPRC（如果使能了WDT 或FSCM）或SOSC（如果SOSCEN 位保持置1 状态）除外。

**注** 1: 在整个时钟切换过程中，处理器将继续执行代码。对时序敏感的代码不应在此时执行。

2: 不允许直接在使能PLL 和FRCPLL 模式的任何主振荡器模式之间进行时钟切换。这适用于任一方向的时钟切换。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。

# PIC24FJ128GB204 系列

时钟切换的建议代码序列通常包括：

1. 在 OSCCON 寄存器解锁和写序列过程中禁止中断。
2. 通过两条连续的指令将 78h 和 9Ah 写入 OSCCON<15:8> 执行 OSCCON 高字节的解锁序列。
3. 执行解锁序列之后，立即使用指令将新振荡器源写入 NOSC<sub>x</sub> 位。
4. 通过两条连续的指令将 46h 和 57h 写入 OSCCON<7:0> 执行 OSCCON 低字节的解锁序列。
5. 执行解锁序列之后，立即使用指令将 OSWEN 位置 1。
6. 继续执行对时钟要求不高的代码（可选）。
7. 调用适当时间的软件延时（周期计数），允许选定的振荡器和 / 或 PLL 启动并稳定下来。
8. 检查 OSWEN 位是否为 0。如果是，则切换成功。如果 OSWEN 位仍为置 1 状态，则检查 LOCK 位以确定故障原因。

解锁 OSCCON 寄存器并启动时钟切换的核心序列如例 9-1 所示。

## 例 9-1: 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
```

## 9.5 FRC 自调节

PIC24FJ128GB204 系列器件包含一个自动机制，可在运行时校准 FRC。该系统根据精度已知的时钟源来恢复时钟，将 FRC 保持在其标称值 8 MHz 频率的较小区间内。这允许频率精度超出 0.25%，刚好在“USB 2.0 规范”的要求内。

**注：** 自调节功能保持足够的精度，可工作在 USB 设备模式下。对于用作 USB 主机的应用，仍要求高精度时钟源 ( $\pm 0.05\%$ )。

自调节系统由 OSCTUN 寄存器的上半部分中的位控制。将 STEN 位 (OSCTUN<15>) 置 1 使能系统，并使其根据 STSRC 位 (OSCTUN<12>) 所选择的时钟源恢复校准时钟。STSRC = 1 时，系统使用来自外部 USB 主机的帧起始 (Start-of-Frame, SOF) 数据包作为其时钟源。STSRC = 0 时，系统使用晶振控制的 SOSC 作为其校准源。无论校准源为何，系统都使用 TUN<5:0> 位 (OSCTUN<5:0>) 来更改 FRC 的频率。频率监视和调整是动态的，可在运行时期间连续进行。系统活动时，无法用软件写 TUN<sub>x</sub> 位。

**注：** 要将 USB 用作时钟恢复源 (STSRC = 1)，单片机必须配置为 USB 操作且连接到外部 USB 设备。  
如果要使用 SOSC 作为时钟恢复源 (STSRC = 0)，必须始终使能 SOSC。

自调节系统可产生硬件中断 FSTIF。可在 FRC 偏移参考时钟大于 0.2%（任一方向）时或每当频率偏移超出 TUN<sub>x</sub> 可纠正的能力（即大于 1.5%）时，产生中断。STLOCK 和 STOR 状态位 (OSCTUN<11,9>) 用于指示这些条件。

STLPOL 和 STORPOL 位 (OSCTUN<10,8>) 配置 FSTIF 中断在条件存在或不存在时发生。用户应监视 STLOCK 和 STOR 位以确定中断的确切原因。

**注：** 禁止自调节系统 (STEN = 0) 时应忽略 STLPOL 和 STORPOL 位。

## 9.6 振荡器模式和 USB 工作

由于 USB 有时序要求，所以使能了 USB 模块且未处于暂停工作状态时需要一个 48 MHz 的内部时钟。当 CPU 以最大时钟速度运行时，这完全可以实现，因此提供了一个方法从单个振荡器源内部生成 USB 时钟和系统时钟。PIC24FJ128GB204 系列器件使用了与大部分其他 PIC24FJ 器件相同的时钟结构，但是包含了一个可产生两种时钟信号的双分支 PLL 系统。

USB PLL 模块如图 9-2 所示。在这个系统中，主振荡器的输入被 PLL 预分频器分频，产生 4 MHz 的输出。这可以用来驱动片内 96 MHz PLL 倍频器以驱动双时钟分支。一个分支使用固定的 2 分频分频器来产生 48 MHz 的 USB 时钟。另一个分支使用固定的 3 分频分频器和可配置的 PLL 预分频器/分频器产生一组系统时钟频率。CPDIVx 位选择系统时钟速度；表 9-2 列出了可用时钟选项。

USB PLL 预分频器不会自动检测传入的振荡器频率。用户必须使用 PLLDIV<3:0> 配置位手动配置 PLL 分频器，以产生所需的 4 MHz 输出。这将主振荡器频率选择限制为 8 种，如表 9-3 所示。

表 9-2: USB 工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) <sup>(1)</sup>	8 MHz
+8 (11) <sup>(1)</sup>	4 MHz

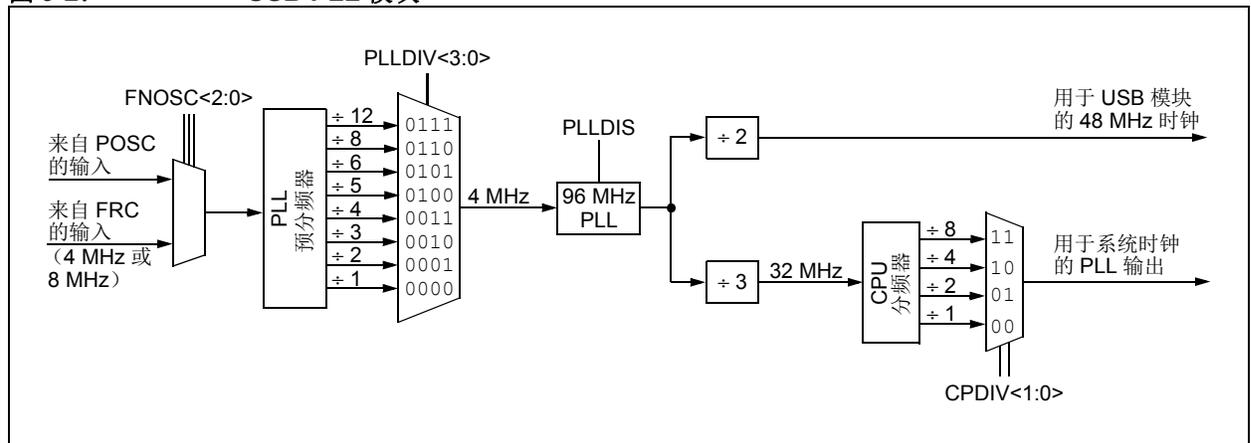
注 1: 与 USB 工作不兼容；必须禁止 USB 模块以使用系统时钟选项。

表 9-3: 确保 USB 正常工作的有效主振荡器配置

振荡器输入频率	时钟模式	PLL 分频 (PLLDIV<3:0>)
48 MHz	ECPLL	+12 (0111)
32 MHz	ECPLL	+8 (0110)
24 MHz	HSPLL 和 ECPLL	+6 (0101)
20 MHz	HSPLL 和 ECPLL	+5 (0100)
16 MHz	HSPLL 和 ECPLL	+4 (0011)
12 MHz	HSPLL 和 ECPLL	+3 (0010)
8 MHz	ECPLL、XTPLL 和 FRCPLL <sup>(1)</sup>	+2 (0001)
4 MHz	ECPLL、XTPLL 和 FRCPLL <sup>(1)</sup>	+1 (0000)

注 1: 需要使用 FRC 自调节功能来保持所需的时钟精度。

图 9-2: USB PLL 模块



# PIC24FJ128GB204 系列

## 9.6.1 USB 工作的注意事项

使用 PIC24FJ128GB204 系列器件的 USB On-The-Go 模块时，用户在配置系统时钟时必须始终遵守以下这些规则：

- 表 9-3 列出的振荡器模式是允许 USB 工作的唯一振荡器配置。没有为 USB 模块提供独立的外部时钟源。
- 要确保 USB 正常工作，所选的时钟源（EC、HS 或 XT）必须满足 USB 时钟容差的要求。
- 当 FRCPLL 振荡器模式用于 USB 应用时，也应使用 FRC 自调节系统。FRC 精确时，在整个应用的工作范围内确保“USB 2.0 规范”所需的精度级别的唯一两个方法是：自调节系统或手动更改 TUNx 位。
- 用户必须始终确保 FRC 源已被配置为提供 4 MHz 或 8 MHz 频率（RCDIV<2:0> = 001 或 000）且 USB PLL 预分频器配置正确。
- 所有其他振荡器模式都是可用的；但是，选择这些模式时 USB 无法工作。然而这些模式仍然可以在需要其他功耗级别且无需 USB 模块的应用中（例如，应用于处于休眠模式下并等待总线连接时）。

## 9.7 参考时钟输出

除了某些振荡器模式中使用的 CLKO 输出（Fosc/2）外，还可对 PIC24FJ128GB204 系列器件中的器件时钟进行配置，以为端口引脚提供参考时钟输出信号。此特性可用于所有的振荡器配置，用户可通过它选择更大范围的时钟分频比以驱动应用中的外部器件。

该参考时钟输出由 REFOCONL、REFOCONH 和 REFOTRIML 寄存器（寄存器 9-4、寄存器 9-5 和寄存器 9-6）控制。将 ROEN 位（REFOCONL<15>）置 1 使能该模块。置 1 ROOUT 位（REFOCONL<12>）可使 REFO 引脚上的时钟信号可用。

RODIVx 位（REFOCON<14:0>）允许选择 32768 种不同的时钟分频比选项。

### 9.7.1 时钟源请求

ROSELx 位用于确定模块的不同基本时钟源。

如果所需的时钟源具有全局器件使能选项（通过器件配置熔丝设置），用户必须先使能时钟源，然后再将其选作基本时钟源。

ROACTIVE 位（REFOCONL<8>）用于在开启和关闭模块时同步 REFO 模块。

**注：** ROEN 位在置 1 后，不应清零，直至 ROACTIVE 位读为 1。

### 9.7.2 时钟切换

可切换模块的基本时钟。首先，通过清零 ROEN 位（REFOCONL<15> = 0）关闭模块，并等待 ROACTIVE 位（REFOCONL<8>）由硬件清零。

这避免 REFO 输出中存在毛刺。

可实时更改 ROTRIMx 和 RODIVx 位。更改 ROTRIMx 和 RODIVx 位之前，请遵循下面所述的步骤。

- REFO 未有效执行分频器切换（ROSWEN = 0）。
- 使用最新的值更新 ROTRIMx 和 RODIVx 位。
- 将 ROSWEN 位置 1。
- 等待 ROSWEN 位由硬件清零。

ROTRIMx 位允许将小数分频器添加到整数分频器上，通过 RODIVx 寄存器位指定。

#### 公式 9-1: ROTRIMx 位的小数分频

对于  $RODIV<14:0> = 0$ ，不分频：

$RODIV<14:0> > 0$ ，周期 =  $2 * (RODIVx + ROTRIMx)$

### 9.7.3 休眠模式下的操作

ROSLP 和 ROSELx 位（REFOCONL<11,3:0>）控制休眠模式下参考时钟输出的可用性。

ROSLP 位决定当器件处于休眠模式时 REFO 引脚上的参考时钟源是否可用。

要在休眠模式下使用参考时钟输出，必须将 ROSLP 置 1 且参考基本时钟不应为系统时钟或外设时钟（ROSELx 位不应为 0b0000 或 0b0001）。

器件时钟还必须进行以下某种配置：

- 其中一种主模式（EC、HS 或 XT）；应置 1 POSCEN 位
- 应置 1 辅助振荡器位（SOSCEN）
- LPRC 振荡器

如果未满足以上条件之一，那么当器件进入休眠模式时，OSC1、OSC2 和 SOSC1 上的振荡器将掉电。

# PIC24FJ128GB204 系列

## 寄存器 9-4: REFOCONL: 参考振荡器控制寄存器低位字

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ROEN	—	ROSIDL	ROOUT	ROSLP	—	ROSWEN	ROACTIVE
bit 15						bit 8	

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	ROSEL3	ROSEL2	ROSEL1	ROSEL0
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 = 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     **ROSIDL:** 空闲模式下参考振荡器输出停止位  
           1 = 空闲模式下禁止参考振荡器  
           0 = 空闲模式下参考振荡器继续运行
- bit 12     **ROOUT:** 参考时钟输出使能位  
           1 = 在 REFO 引脚上驱动 REFO 时钟输出  
           0 = 禁止 REFO 时钟输出
- bit 11     **ROSLP:** 休眠模式下参考振荡器输出位  
           1 = 休眠模式下参考振荡器输出继续运行  
           0 = 休眠模式下禁止参考振荡器输出
- bit 10     **未实现:** 读为 0
- bit 9      **ROSWEN:** 参考振荡器时钟源切换使能位  
           1 = 参考时钟源切换当前正在进行中  
           0 = 参考时钟源切换已完成
- bit 8      **ROACTIVE:** 参考时钟请求状态位  
           1 = 参考时钟请求有效 (用户不应更新 REFOCONL 寄存器)  
           0 = 参考时钟请求无效 (用户可更新 REFOCONL 寄存器)
- bit 7-4    **未实现:** 读为 0  
           (保留用于其他 ROSELx 位。)
- bit 3-0    **ROSEL<3:0>:** 参考时钟源选择位  
           选择各种时钟源之一用作参考时钟:  
           1001-1111 = 保留  
           1000 = REFI (参考时钟输入)  
           0111 = 保留  
           0110 = 8x PLL 或 USB-PLL  
           0101 = 辅助振荡器 (SOSC)  
           0100 = 低功耗 RC 振荡器 (LPRC)  
           0011 = 快速 RC 振荡器 (FRC)  
           0010 = 主振荡器 (XT、HS 和 EC)  
           0001 = 外设时钟 (PBCLK) —— 内部指令周期时钟 Fcy  
           0000 = 系统时钟 (Fosc)

# PIC24FJ128GB204 系列

寄存器 9-5:            **REFOCONH: 参考振荡器控制寄存器高位字**

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	RODIV<14:8>							
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
RODIV<7:0>							
bit 7							bit 0

<b>图注:</b>							
R = 可读位	W = 可写位	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 14-0        **RODIV<14:0>:** 参考振荡器分频比选择位  
在源时钟中指定 1/2 周期的参考时钟。

例如: `ref_clk_output` 的周期  $\leq$  [参考源 \* 2] \* RODIV<14:0>:

1111111111111111 = REFO 时钟为基本时钟频率的 65,534 (32,767 \* 2) 分频

1111111111111110 = REFO 时钟为基本时钟频率的 65,532 (32,766 \* 2) 分频

•

•

•

000000000000011 = REFO 时钟为基本时钟频率的 6 (3 \* 2) 分频

000000000000010 = REFO 时钟为基本时钟频率的 4 (2 \* 2) 分频

000000000000001 = REFO 时钟为基本时钟频率的 2 (1 \* 2) 分频

000000000000000 = REFO 时钟与基本时钟相同 (无分频) <sup>(1)</sup>

注 1: ROTRIMx 值被忽略。

# PIC24FJ128GB204 系列

寄存器 9-6: REFOTRIML: 参考振荡器调节寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROTRIM<15:8>							
bit 15							bit 8

R/W-0	U-0						
ROTRIM7	—	—	—	—	—	—	—
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      **ROTRIM<15:7>**: 参考振荡器调节位  
 提供小数值与 RODIVx 值的和作为 1/2 周期的 REFO 时钟  
 111111111 = 511/512 (0.998046875) 分频值与 RODIVx 值的和  
 111111110 = 510/512 (0.99609375) 分频值与 RODIVx 值的和  
 .  
 .  
 .  
 100000000 = 256/512 (0.5000) 分频值与 RODIVx 值的和  
 .  
 .  
 .  
 000000010 = 2/512 (0.00390625) 分频值与 RODIVx 值的和  
 000000001 = 1/512 (0.001953125) 分频值与 RODIVx 值的和  
 000000000 = 0/512 (0.0) 分频值与 RODIVx 值的和

bit 6-0      未实现: 读为 0

## 9.8 片内 PLL

PLL 模块使用内部 FRC 源可生成以下频率, 如表 9-4 所示。

可通过配置位 PLLDIV<3:0> 选择片内 PLL (x4、x6 和 x8)。主振荡器和 FRC 源 (FRCDIV) 可选择使用该 PLL。

表 9-4: 片内 PLL 的有效 FRC 配置<sup>(1)</sup>

FRC	RCDIV<2:0> (FRCDIV)	x4 PLL	x6 PLL	x8 PLL
8 MHz	000 (1 分频)	32 MHz	—	—
8 MHz	001 (2 分频)	16 MHz	24 MHz	32 MHz
8 MHz	010 (4 分频)	8 MHz	12 MHz	16 MHz

注 1: 片内 PLL 的最低频率输入为 2 MHz。

# PIC24FJ128GB204 系列

---

注:

## 10.0 节能特性

**注：** 本数据手册总结了该 PIC24FJ 系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“带深度休眠的节能特性” (DS39727)。

PIC24FJ128GB204 系列器件提供了管理功耗的功能，该功能是通过选择性地管理 CPU 和外设的时钟来实现的。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。

PIC24FJ128GB204 系列器件通过以下 5 种方法管理功耗：

- 基于指令的低功耗模式
- 基于硬件的低功耗特性
- 时钟频率控制
- 软件控制的打盹模式
- 通过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应用关键特性（如对于时序要求高的通信）的情况下有选择地调节应用的功耗。

## 10.1 节能模式概述

除了全功耗操作（也称为运行模式），PIC24FJ128GB204 系列器件提供了三种基于指令的节能模式和一种基于硬件的模式：

- 空闲
- 休眠（休眠和低压休眠）
- 深度休眠
- VBAT（带或不带 RTCC）

通过依次关断单片机不同功能区可激活这四种模式，以逐渐降低工作和空闲功耗。此外，可以调节其中的三种模式以进一步降低功耗，但要以某些工作性能为代价。表 10-1 以节能程度递增的顺序列出了所有工作模式。表 10-2 汇总了单片机退出不同模式的方式。以下几节中提供了具体信息。

**表 10-1: PIC24FJ128GB204 系列器件的工作模式**

模式	进入	有效系统				
		内核	外设	数据 RAM 保持	RTCC <sup>(1)</sup>	DSGPR0/DSGPR1 保持
运行（默认情况）	N/A	是	是	是	是	是
空闲	指令	否	是	是	是	是
休眠：						
休眠	指令	否	可选 <sup>(2)</sup>	是	是	是
低电压休眠	指令 + RETEN 位	否	可选 <sup>(2)</sup>	是	是	是
深度休眠：						
深度休眠	指令 + DSEN 位	否	否	否	是	是
VBAT：						
带 RTCC	硬件	否	否	否	是	是

**注 1：** 如果 RTCC 另外由固件使能。

**注 2：** 通过使用 LPRC 或外部时钟，选定外设可在该模式下继续工作。

# PIC24FJ128GB204 系列

表 10-2: 退出节能模式

模式	退出条件								恢复代码执行
	中断		复位			RTCC 闹钟	WDT	VDD 恢复 <sup>(2)</sup>	
	全部	INT0	全部	POR	MCLR				
空闲	是	是	是	是	是	是	是	N/A	下一条指令
休眠 (所有模式)	是	是	是	是	是	是	是	N/A	
深度休眠	否	是	否	是	是	是	是 <sup>(1)</sup>	N/A	复位向量
VBAT	否	否	否	否	否	否	否	是	复位向量

注 1: 深度休眠 WDT。

注 2: 一旦 VDD 被移除并且在任何除保持深度休眠外的模式下恢复时, 就会发生 POR 或类似于 POR 的复位。

## 10.1.1 基于指令的节能模式

三种节能模式均通过执行 PWRSAV 指令进入。休眠模式下时钟停止工作并暂停所有代码执行。空闲模式下 CPU 暂停工作并暂停代码执行, 但是允许外设模块继续工作。深度休眠模式下时钟停止工作, 代码执行停止, 以及 RTCC 和 DSWDT 之外的所有外设都停止。它还冻结 I/O 状态和除去对闪存和 SRAM 的供电。

PWRSAV 指令的汇编语法如例 10-1 所示。休眠和空闲模式可通过单个汇编命令直接进入。深度休眠需要额外序列来解锁和使能进入, 如第 10.4.1 节“进入深度休眠模式”中所述。

**注:** SLEEP\_MODE 和 IDLE\_MODE 是在所选器件的汇编器头文件中定义的常量。

允许的中断、WDT 超时或器件复位会导致器件从休眠和空闲模式退出。器件退出这两种模式的过程称为“唤醒”。

通过低压 / 保持稳压器使能此特性会改变休眠和深度休眠模式下的行为。更多详细信息, 请参见第 10.3 节“休眠模式”和第 10.4 节“深度休眠模式”。

### 10.1.1.1 与节能指令并发的中断

任何在执行 PWRSAV 指令时产生的中断都将被推迟到进入休眠或空闲模式后发生。然后将器件从休眠或空闲模式唤醒。

关于深度休眠模式, 在执行 PWRSAV 指令时产生的中断可能会丢失。如果未使能低压 / 保持稳压器, 单片机将在退出深度休眠时复位, 中断将丢失。

在深度休眠解锁序列期间产生的中断, 将中断必需的 5 指令周期序列时序, 导致进入深度休眠失败。为此, 建议在深度休眠解锁序列期间禁止所有中断。

例 10-1: PWRSAV 指令语法

```
// Syntax to enter Sleep mode:
PWRSAV    #SLEEP_MODE        ; Put the device into SLEEP mode
//
//Syntax to enter Idle mode:
PWRSAV    #IDLE_MODE         ; Put the device into IDLE mode
//
// Syntax to enter Deep Sleep mode:
// First use the unlock sequence to set the DSEN bit (see Example 10-2)
BSET      DSCON, #DSEN        ;Enable Deep Sleep
BSET      DSCON, #DSEN        ; Enable Deep Sleep(repeat the command)
PWRSAV    #SLEEP_MODE        ; Put the device into Deep SLEEP mode
```

## 10.1.2 基于硬件的节能模式

基于硬件的VBAT模式在代码开发期间，无需用户执行任何操作。此模式是一种硬件设计的功能，允许单片机在从应用移除VDD时保留关键数据（使用DSGPRx寄存器）并且维持RTCC运行。这通过提供备用电源给指定的电源引脚来完成。在[第10.5节“VBAT模式”](#)中详细介绍了VBAT模式。

## 10.1.3 低压 / 保持稳压器

PIC24FJ128GB204系列器件结合了第二个片上稳压器，旨在为选择的单片机功能提供1.2V的标称电源。此稳压器允许如数据RAM和WDT之类的功能，在节能模式下继续工作（其他情况为非活动状态）或以低于其他情况下的功耗模式工作。

仅在休眠模式下，低压 / 保持稳压器才可用。通过LPCFG配置位（CW1<10>）及在固件中通过RETE位（RCON<12>）控制。LPCFG必须编程（=0）并且RETE位必须置1（=1）以使能稳压器。

## 10.2 空闲模式

空闲模式包含下列这些特性：

- CPU将停止执行指令。
- WDT自动清零。
- 系统时钟源保持活动状态。默认情况下，所有外设模块将使用系统时钟源继续正常工作，但也可被有选择地禁止（见[第10.8节“选择性外设模块控制”](#)）。
- 若使能了WDT或FSCM，LPRC也将保持活动状态。

当发生以下任何事件时，器件将从空闲模式唤醒：

- 产生被单独允许的中断
- 任何器件复位
- WDT超时

从空闲模式唤醒时，为CPU重新提供时钟，且立即从PWRSAV指令之后的下一条指令或中断服务程序（ISR）中的第一条指令开始执行。

## 10.3 休眠模式

休眠模式包含以下这些特性：

- 系统时钟源关闭。如果使用了片上振荡器，也要关闭它。
- 如果没有I/O引脚拉电流，则器件电流消耗将降至最低。
- 冻结I/O引脚方向和状态。
- 由于禁止了系统时钟源，所以故障保护时钟监视器在休眠模式下不工作。
- 如果使能了以LPRC作为时钟源的WDT或RTCC，LPRC时钟将继续在休眠模式下运行。
- 如果使能了WDT，它将在进入休眠模式之前自动清零。
- 某些器件功能或外设可在休眠模式下继续工作。这包括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。任何需要系统时钟源工作的外设都会在休眠模式下被禁止。

当发生以下任何事件时，器件将从休眠模式唤醒：

- 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的有效时钟源重新启动。

### 10.3.1 低压 / 保持休眠模式

低压 / 保持休眠模式与普通休眠模式具有相同的特性和唤醒触发源。差异在于低压 / 保持稳压器允许内核数字逻辑电压（V<sub>CORE</sub>）降至1.2V标称值。这使得进一步降低保持V<sub>CORE</sub>在1.8V（最小）电平所需的功耗。

低电压休眠模式相比普通休眠模式需要较长的唤醒时间，由于需要额外的时间使V<sub>CORE</sub>回到1.8V（称为T<sub>REG</sub>）。另外，低压 / 保持稳压器的使用限制了可提供给任何活动外设（如RTCC等）的电流。

# PIC24FJ128GB204 系列

## 10.4 深度休眠模式

深度休眠模式提供了基于指令的模式中可用的最低功耗级别。

深度休眠模式有 3 种特性：

- 系统时钟源关闭。如果使用了片上振荡器，也要关闭它。
- 器件电流消耗将降为最小。
- 冻结 I/O 引脚方向和状态。
- 由于禁止了系统时钟源，所以故障保护时钟监视器在休眠模式下不工作。
- 如果使能了以 LPRC 作为时钟源的 WDT 或 RTCC，LPRC 时钟将继续在深度休眠模式下运行。
- 如果使能，则使用专用深度休眠 WDT 和 BOR 系统。
- 如果使能，RTCC 和其时钟源继续运行。禁止所有其他外设。

深度休眠模式的进入是完全受软件控制的。发生以下任何事件时可使器件退出深度休眠模式：

- POR 事件
- MCLR 事件
- RTCC 闹钟（如果使用了 RTCC）
- 外部中断 0
- 深度休眠看门狗定时器（Deep Sleep Watchdog Timer, DSWDT）超时

### 10.4.1 进入深度休眠模式

进入深度休眠模式的方法是：将 DSCON 寄存器中的 DSEN 位置 1，然后在一个指令周期内执行休眠命令（PWRSAV #SLEEP\_MODE）以减少伪进入深度休眠模式的可能性。

如果 PWRSAV 命令不是在一个指令周期内给出的，DSEN 位将由硬件清零，且必须由软件再次置 1，然后才能进入深度休眠模式。退出深度休眠模式时，DSEN 位也将自动清零。

**注：** 要在从深度休眠唤醒之后再次进入深度休眠模式，允许在清零 RELEASE 位之后至少有 3 个 Tcy 的延时。

进入深度休眠模式的步骤是：

1. 如果应用需要深度休眠 WDT，可使其能并配置其时钟源。关于深度休眠 WDT 的更多信息，请参见第 10.4.5 节“深度休眠 WDT”。
2. 如果应用需要深度休眠 BOR，可通过编程 DSBOREN 配置位（CW4<6>）使其能。
3. 如果应用需要在发生 RTCC 闹钟事件时从深度休眠模式唤醒，可使其能并配置 RTCC 模块。关于 RTCC 的更多信息，请参见第 22.0 节“实时时钟和日历（RTCC）”。
4. 如有需要，可通过把应用的任何重要的现场数据写入 DSGPR0 和 DSGPR1 寄存器（可选）来保存这些数据。
5. 通过置 1 DSEN 位（DSCON<15>）使其能深度休眠模式。

**注：** 重复序列需要置 1 DSEN 位。重复序列（重复指令两次）需要写入任何深度休眠寄存器（DSCON、DSWAKE、DSGPR0 和 DSGPR1）。此要求是为了防止用户误进入深度休眠。任何对这些寄存器的写操作必须完成两次，以实际完成写操作（见例 10-2）。

6. 通过发出 3 条 NOP 命令，然后发出 PWRSAV #0 指令进入深度休眠模式。

置 1 DSEN 位时，DSWAKE 寄存器中的所有位都将自动清零。

### 例 10-2: 重复序列

```
例 1:
mov #8000, w2      ; enable DS
mov w2, DSCON
mov w2, DSCON      ; second write required to
                   ; actually write to DSCON

例 2:
bset   DSCON, #15
nop
nop
nop
bset   DSCON, #15 ; enable DS (two writes
                   ; required)
```

## 10.4.2 退出深度休眠模式

发生以下任一事件时退出深度休眠模式：

- VDD 电源上的 POR 事件。如果没有提供 DSBOR 电路重新激活 VDD 电源 POR 电路，那么必须把外部 VDD 电源降低到 POR 电路的自然激活电压。
- DSWDT 超时。DSWDT 超时，器件退出深度休眠模式。
- RTCC 闹钟（如果 RTCEN = 1）。
- MCLR 引脚有效（0）。
- INT0 引脚有效（如果在进入深度休眠模式之前允许此中断的话）。极性配置用于确定导致器件退出深度休眠模式的引脚的有效电平（0 或 1）。深度休眠模式期间，INT0 引脚上的电平变化会导致器件退出深度休眠模式。

**注：** 进入深度休眠模式时待处理的任何中断都将被清除。

退出深度休眠通常不保持器件的状态，它等同于器件的上电复位（POR）。除此以外，还包括在唤醒过程中始终保持工作状态的 RTCC（如果有）、DSGPRx 寄存器和 DSWDT。

不要忽略从退出深度休眠模式到 POR 序列完成过程中发生的唤醒事件。DSWAKE 寄存器将捕捉从置 1 DSEN 位至清零 RELEASE 位期间发生的所有唤醒事件。

退出深度休眠模式的步骤是：

1. 发生唤醒事件后，器件退出深度休眠模式并执行 POR。DSEN 位自动清零。从复位向量处继续执行代码。
2. 要确定器件是否已退出深度休眠模式，请读取深度休眠位 DPSLP（RCON<10>）。如果确定已退出深度休眠模式，该位将置 1。如果该位置 1，请将其清零。
3. 通过读 DSWAKE 寄存器确定唤醒源。
4. 通过读 DSBOR 位（DSCON<1>）确定深度休眠模式期间是否发生 DSBOR 事件。
5. 如果已保存应用的现场数据，请将其从 DSGPR0 和 DSGPR1 寄存器读回。
6. 清零 RELEASE 位（DSCON<0>）。

## 10.4.3 使用 DSGPRx 寄存器保存现场数据

由于退出深度休眠模式会导致 POR，大部分特殊功能寄存器都将复位为其默认的 POR 值。而且，由于深度休眠模式下不提供 V<sub>CORE</sub> 电源，所以退出此模式时有可能丢失数据 RAM 中的信息。

需要在进入深度休眠模式之前保存重要数据的应用可使用深度休眠通用寄存器 DSGPR0 和 DSGPR1 或数据 EEPROM（如果可用的话）。这些寄存器与其他 SFR 不同，当器件处于深度休眠模式时将保留其内容。退出深度休眠模式后，可用软件通过读这些寄存器并清零 RELEASE 位（DSCON<0>）来恢复数据。

**注：** 用户软件应使能 DSSWEN（CW4<8>）配置熔丝位，以保存 DSGPRx 寄存器中的关键数据。

## 10.4.4 深度休眠模式下的 I/O 引脚

深度休眠模式期间，通用 I/O 引脚保留其之前的状态且辅助振荡器（SOSC）将继续运行（如果使能的话）。在进入深度休眠模式之前配置为输入（TRISx 位置 1）的引脚在深度休眠模式期间保持高阻态。在进入深度休眠模式之前配置为输出（TRISx 位清零）的引脚在深度休眠模式期间保持为输出引脚。在此模式期间，它们继续驱动在进入深度休眠模式时其对应的 LATx 位所确定的输出电平。

一旦器件被唤醒，所有 I/O 引脚将继续保持其之前的状态，即使在器件完成了 POR 序列并再次执行应用程序代码后也是如此。配置为输入的引脚在深度休眠模式期间保持高阻态，配置为输出的引脚继续驱动其之前的电平值。唤醒后，TRISx 和 LATx 寄存器以及 SOSCEN 位（OSCCON<1>）都将复位。如果固件修改了这些位或寄存器中的任何一个，I/O 将不会立即进入新配置的状态。在固件清零 RELEASE 位（DSCON<0>）时，I/O 引脚将“被释放”。从而导致 I/O 引脚处于其对应的 TRISx 和 LATx 位的值所配置的状态。

这意味着唤醒后保持 SOSC 运行将需要在清零 RELEASE 之前把 SOSCEN 位置 1。

如果使能了深度休眠 BOR（DSBOR），且深度休眠期间发生了 DSBOR 或真正的 POR 事件，那么 I/O 引脚将立即被释放，这与清零 RELEASE 位有些类似。所有之前的状态信息都将丢失，包括通用 DSGPR0 和 DSGPR1 的内容。

# PIC24FJ128GB204 系列

如果深度休眠期间发生了  $\overline{\text{MCLR}}$  复位事件，那么 DSGPRx、DSCON 和 DSWAKE 寄存器将保持有效且 RELEASE 位将保持置 1 状态。SOSC 的状态也将保持。但是，I/O 引脚将复位为其 MCLR 复位状态。由于 RELEASE 仍然置 1，所以对 SOSCEN 位 (OSCCON<1>) 的更改直到 RELEASE 位清零之后才会生效。

在所有其他深度休眠唤醒情况下，应用固件必须清零 RELEASE 位才能重新配置 I/O 引脚。

## 10.4.5 深度休眠 WDT

要启用深度休眠模式下的 DSWDT，请编程配置位 DSWDTEN (CW4<7>)。无需为了 DSWDT 正常工作而使能器件 WDT。进入深度休眠模式将自动复位 DSWDT。

通过 DSWDTOSC 配置位 (CW4<5>) 选择 DSWDT 的时钟源。通过 DSWDTPS<4:0> 配置位 (CW4<4:0>) 编程后分频比选项。可实现的最小超时时间为 1 ms，最大为 25.7 天。更多关于 CW4 配置寄存器和 DSWDT 配置选项的信息，请参见第 30.0 节“特殊功能”。

### 10.4.5.1 深度休眠模式下切换时钟

RTCC 和 DSWDT 可以从 SOSC 或 LPRC 时钟源提供的时钟运行。这使得 RTCC 和 DSWDT 可以在不需要同时使能 LPRC 和 SOSC 的情况下运行，从而降低功耗。

从 LPRC 提供的时钟运行 RTCC 将导致 RTCC 的精度损失大约 5% 至 10%。如果需要确保 RTCC 更加精确，必须从 SOSC 时钟源提供的时钟运行 RTCC。通过 RTCLK<1:0> 位 (RTCPWC<11:10>) 选择 RTCC 的时钟源。

在某些情况下，可以在进入深度休眠模式时关闭 DSWDT 的时钟源。在这种情况下，无需使用软件，就可自动开启时钟源（如果使能了 DSWDT）。但是，这会导致 DSWDT 计数器启动时有一段延时。为了避免 SOSC 用作时钟源时的这段延时，应用可在进入深度休眠模式之前激活 SOSC。

## 10.4.6 检查并清零深度休眠的状态位

进入深度休眠模式时，状态位 DPSLP (RCON<10>) 置 1，且必须由软件清零。

上电时，软件应读取该状态位以确定复位是否由退出深度休眠模式引起，如果该位置 1 的话，则将其清零。在 DPSLP 和 POR 位状态的四种可能组合中，须注意以下三种情况：

- DPSLP 和 POR 位都清零。在这种情况下，复位由退出深度休眠模式外的其他事件引起。
- 清零 DPSLP 位，但是置 1 POR 位；这是正常的上电复位。
- DPSLP 和 POR 位都置 1。这表示进入深度休眠模式，器件掉电，然后退出了深度休眠模式。

## 10.4.7 上电复位 (POR)

监视 VDD 电压以产生 POR。由于退出深度休眠模式在功能上看起来与 POR 类似，但是应使用第 10.4.6 节“检查并清零深度休眠的状态位”中描述的技巧来区别退出深度休眠和真正的 POR 事件。发生真正的 POR 时，包括所有深度休眠逻辑（深度休眠寄存器、RTCC 和 DSWDT 等）在内的整个器件都将复位。

## 10.5 VBAT 模式

此模式是单片机可实现且仍能恢复工作的最低功耗状态。当 VDD 上的单片机主电源发生故障时，会自动触发 VBAT 模式。发生这种情况时，单片机的片上电源开关连接到为 VBAT 引脚供电的备用电源（如电池）。此时以超低功耗维持一些关键系统，直到恢复 VDD。

VBAT 上提供的电源仅为两个系统运行供电：RTCC 和深度休眠信号量寄存器 (DSGPR0 和 DSGPR1)。要在突然丢失 VDD 期间维持这些系统工作，需将 VDD 或 AVDD 以外的电源连接至 VBAT 引脚。

RTCC 在使能时，使用在进入 VBAT 模式前所选择的同一时钟源 (SOSC 或 LPRC) 继续工作。在切换模式后不能切换为较低功耗时钟源。

由于丢失 VDD 通常是不可预计的事件，建议深度休眠信号量寄存器的内容中装入在代码执行早期保持的数据。

### 10.5.1 不带 RTCC 的 VBAT 模式

通过在 VBAT 模式期间禁止 RTCC 操作，功耗降为所有节能模式中的最低。在此模式下，仅保持深度休眠信号量寄存器。

## 10.5.2 从 VBAT 模式唤醒

当器件的 VDD 恢复时，将自动从 VBAT 模式唤醒。发生 POR 唤醒后，器件从复位向量处开始代码执行。所有的 SFR，除了深度休眠信号量之外，复位至其 POR 值。如果在 VBAT 模式期间，RTCC 未配置为运行，将保持禁止，并且 RTCC 将不运行。唤醒时序与正常 POR 时类似。

为了区分从 VBAT 模式和从其他 POR 状态的唤醒，检查 VBAT 状态位 (RCON2<0>)。如果此位在器件开始执行复位向量处的代码时置 1，指示从 VBAT 模式退出。应用必须清零 VBAT 位以确保可以捕捉将来的 VBAT 唤醒事件。

如果 POR 发生时电源没有连接至 VBAT 引脚，VBPOR 位 (RCON2<1>) 置 1。如果此位在上电复位置 1，指示需要将电池连接至 VBAT 引脚。

另外，在 VBAT 模式下，如果 VBAT 电源降到深度休眠信号量操作所需的电平以下（例如，电池已经耗尽），VBPOR 位将置 1。VBPOR 在单片机首次上电时也会置 1，即使已为 VBAT 供电。

## 10.5.3 VBAT 模式期间的 I/O 引脚

VBAT 模式期间所有 I/O 引脚都切换至输入模式。唯一例外是 SOSCI 和 SOSCO 引脚，在将辅助振荡器用作 RTCC 时钟源时这两个引脚保持其状态。一旦恢复 VDD，用户应使用 TRISx 和 LATx 位将 I/O 引脚恢复至其正确状态。

## 10.5.4 使用 DSGPRx 寄存器保存现场数据

和深度休眠模式一样（即，不带低压 / 保持稳压器），所有的 SFR 在恢复 VDD 之后复位至其 POR 值。仅保留深度休眠信号量寄存器。应用将需要保存的关键数据保存在 DSGPR0 和 DSGPR1 中。

<b>注：</b>	如果未使用 VBAT 模式，建议将 VBAT 引脚连接至 VDD。
-----------	-----------------------------------

应使能 POR 以获取可靠的 VBAT 操作。

# PIC24FJ128GB204 系列

寄存器 10-1: **DSCON: 深度休眠控制寄存器 (1)**

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

U-0	U-0	U-0	U-0	U-0	r-0	R/W-0	R/C-0, HS
—	—	—	—	—	—	DSBOR <sup>(2)</sup>	RELEASE
bit 7							bit 0

<b>图注:</b>	C = 可清零位	U = 未实现位, 读为 0		
R = 可读位	W = 可写位	HS = 可由硬件置 1 的位	r = 保留位	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知	

bit 15 **DSEN:** 深度休眠使能位

1 = 执行 PWRSAV #0 之后进入深度休眠模式  
0 = 执行 PWRSAV #0 之后进入正常休眠模式

bit 14-3 **未实现:** 读为 0

bit 2 **保留:** 保持为 0

bit 1 **DSBOR:** 深度休眠 BOR 事件位 (2)

1 = 深度休眠模式期间, DSBOR 处于活动状态且检测到 BOR 事件  
0 = 深度休眠模式期间, DSBOR 可处于活动或非活动状态但未检测到 BOR 事件

bit 0 **RELEASE:** I/O 引脚状态释放位

1 = 从深度休眠模式唤醒时, I/O 引脚保持进入深度休眠模式之前的状态  
0 = 将 I/O 引脚从进入深度休眠模式之前的状态释放, 允许相应的 TRISx 和 LATx 位控制它们的状态

**注 1:** 仅在 POR 事件不在深度休眠模式期间发生时, 复位所有寄存器位。

**注 2:** 不同于所有其他事件, 深度休眠 BOR 事件不会导致从深度休眠模式唤醒; 它会重新激活 POR。

# PIC24FJ128GB204 系列

## 寄存器 10-2: DSWAKE: 深度休眠唤醒源寄存器 (1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HS
—	—	—	—	—	—	—	DSINT0
bit 15							bit 8

R/W-0, HS	U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0	U-0
DSFLT	—	—	DSWDT	DSRTCC	DSMCLR	—	—
bit 7							bit 0

<b>图注:</b>	HS = 可由硬件置 1 的位
R = 可读位	W = 可写位
-n = POR 时的值	U = 未实现位, 读为 0
	1 = 置 1
	0 = 清零
	x = 未知

bit 15-9 未实现: 读为 0

bit 8 **DSINT0:** 深度休眠电平变化中断位  
 1 = 深度休眠期间, 发生了电平变化中断  
 0 = 深度休眠期间, 未发生电平变化中断

bit 7 **DSFLT:** 深度休眠故障检测位  
 1 = 深度休眠期间发生故障且一些深度休眠配置设置可能遭到破坏  
 0 = 深度休眠期间未检测到任何故障

bit 6-5 未实现: 读为 0

bit 4 **DSWDT:** 深度休眠看门狗定时器超时位  
 1 = 深度休眠期间, 深度休眠看门狗定时器已超时  
 0 = 深度休眠期间, 深度休眠看门狗定时器未超时

bit 3 **DSRTCC:** 深度休眠实时时钟和日历闹钟位  
 1 = 深度休眠期间实时时钟和日历已触发闹钟  
 0 = 深度休眠期间实时时钟和日历未触发闹钟

bit 2 **DSMCLR:** 深度休眠 MCLR 事件位  
 1 = MCLR 引脚处于工作状态, 并且在深度休眠期间被拉低  
 0 = MCLR 引脚可能工作也可能不工作, 但在深度休眠期间没有被拉低

bit 1-0 未实现: 读为 0

**注 1:** 所有寄存器位在 DSEN (DSCON<15>) 位置 1 时清零。

# PIC24FJ128GB204 系列

寄存器 10-3: RCON2: 复位和系统控制寄存器 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-0	R/CO-1	R/CO-1	R/CO-1	R/CO-0
—	—	—	—	VDDBOR <sup>(1)</sup>	VDDPOR <sup>(1,2)</sup>	VBPOR <sup>(1,3)</sup>	VBAT <sup>(1)</sup>
bit 7							bit 0

<b>图注:</b>	CO = 只可清零位	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 15-5     **未实现:** 读为 0
- bit 4       **保留:** 保持为 0
- bit 3       **VDDBOR:** VDD 欠压复位标志位 <sup>(1)</sup>  
 1 = 发生了 VDD 欠压复位 (由硬件置 1)  
 0 = 未发生 VDD 欠压复位
- bit 2       **VDDPOR:** VDD 上电复位标志位 <sup>(1,2)</sup>  
 1 = 发生了 VDD 上电复位 (由硬件置 1)  
 0 = 未发生 VDD 上电复位
- bit 1       **VBPOR:** VBAT 上电复位标志位 <sup>(1,3)</sup>  
 1 = 发生了 VBAT POR (没有电池连接至 VBAT 引脚或 VBAT 电源低于深度休眠信号保持电平; 通过硬件置 1)  
 0 = 未发生 VBAT POR
- bit 0       **VBAT:** VBAT 标志位 <sup>(1)</sup>  
 1 = 在向 VBAT 引脚施加电压时退出 POR (由硬件置 1)  
 0 = 未从 VBAT 退出 POR

注 1: 此位只能用硬件置 1; 只能用软件清零。  
 2: 此位指示 VDD 上电复位。将 POR 位 (RCON<0>) 置 1 指示 V<sub>CORE</sub> 上电复位。  
 3: 该位在器件初始上电时置 1, 即使已向 VBAT 施加电压。

## 10.6 时钟频率和时钟切换

在运行和空闲模式下，所有的 PIC24FJ 器件允许在应用控制下，从较宽的时钟频率范围内进行选择。如果未锁定系统时钟配置，用户只需更改 NOSC<sub>x</sub> 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，在第 9.0 节“振荡器配置”中进行了更详细的讨论。

## 10.7 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，即使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导致通信错误，而使用节能模式就可能完全停止通信。

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，继续以相同的时钟源和相同的速度驱动系统时钟。外设模块时钟速度保持不变，但 CPU 时钟的速度降低了。保持两个时钟域同步，以允许外设 CPU 以较慢的速率执行代码时访问 SFR。

通过置 1 DOZEN 位 (CLKDIV<11>) 使能打盹模式。外设和内核时钟速度之比由 DOZE<2:0> 位 (CLKDIV<14:12>) 决定。有 8 种可能的配置，从 1:1 到 1:128，其中 1:1 是默认设置。

还可使用打盹模式在事件驱动应用中有选择地降低功耗。这样就可以实现不间断地运行对时钟要求高的功能（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操作没有影响。

## 10.8 选择性外设模块控制

空闲和打盹模式允许用户通过降低 CPU 时钟速度或停止 CPU 时钟大幅降低功耗。然而，外设模块的时钟仍然保持运行，因此会有功耗产生。可能在有些情况下应用需要这些模式无法提供的功能，比如将绝大部分电资源分配给 CPU 处理工作，而使外设功耗最低。

PIC24F 器件允许有选择地禁止外设模块，从而降低或消除它们的功耗，以此满足上述需求。这可以通过两个控制位来实现：

- 外设使能位，通常称为“XXXEN”，位于模块主控制 SFR 中。
- 外设模块禁止 (PMD) 位，通常称为“XXXMD”，位于某个 PMD<sub>x</sub> 控制寄存器 (XXXMD 位在 PMD1、PMD2、PMD3、PMD4、PMD6、PMD7、PMD8 寄存器中)。

这两位在使用或禁止相关模块时具有相似的功能。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。在此状态下，与此外设相关的控制和状态寄存器也会被禁止，所以无法写这些寄存器且读取无效。许多外设模块都有一个对应的 PMD 位。

而通过清零某个模块的 XXXEN 位将会禁止其功能，但是仍然允许对其寄存器进行读写操作。这也会降低功耗，但不如将 PMD 位置 1 降低得多。大多数外设模块都有一个使能位，但捕捉、比较和 RTCC 模块除外。

要节省更多的功耗，也可在器件进入空闲模式时有选择地禁止外设模块。这可通过通用名称格式为“XXXSIDL”的控制位实现。默认情况下，可以在空闲模式下工作的所有模块都可以执行此操作。在空闲模式下使用空闲模式禁止功能禁止模块，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。

# PIC24FJ128GB204 系列

---

注:

## 11.0 I/O 端口

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“带外设引脚选择（PPS）的I/O端口”（DS39711）。本数据手册中的信息取代了FRM中的信息。

所有器件引脚（除了V<sub>DD</sub>、V<sub>SS</sub>、MCLR和OSCI/CLKI以外）均由外设和并行I/O端口共用。所有I/O输入端口都为施密特触发器（ST）输入，以便增强抗干扰性。

### 11.1 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端口总是服从于该外设。外设的输出缓冲器数据和控制信号提供一对多路开关。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。该逻辑同时还会阻止“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11-1中显示端口是如何与其他外设复用的，及其连接的关联I/O引脚。

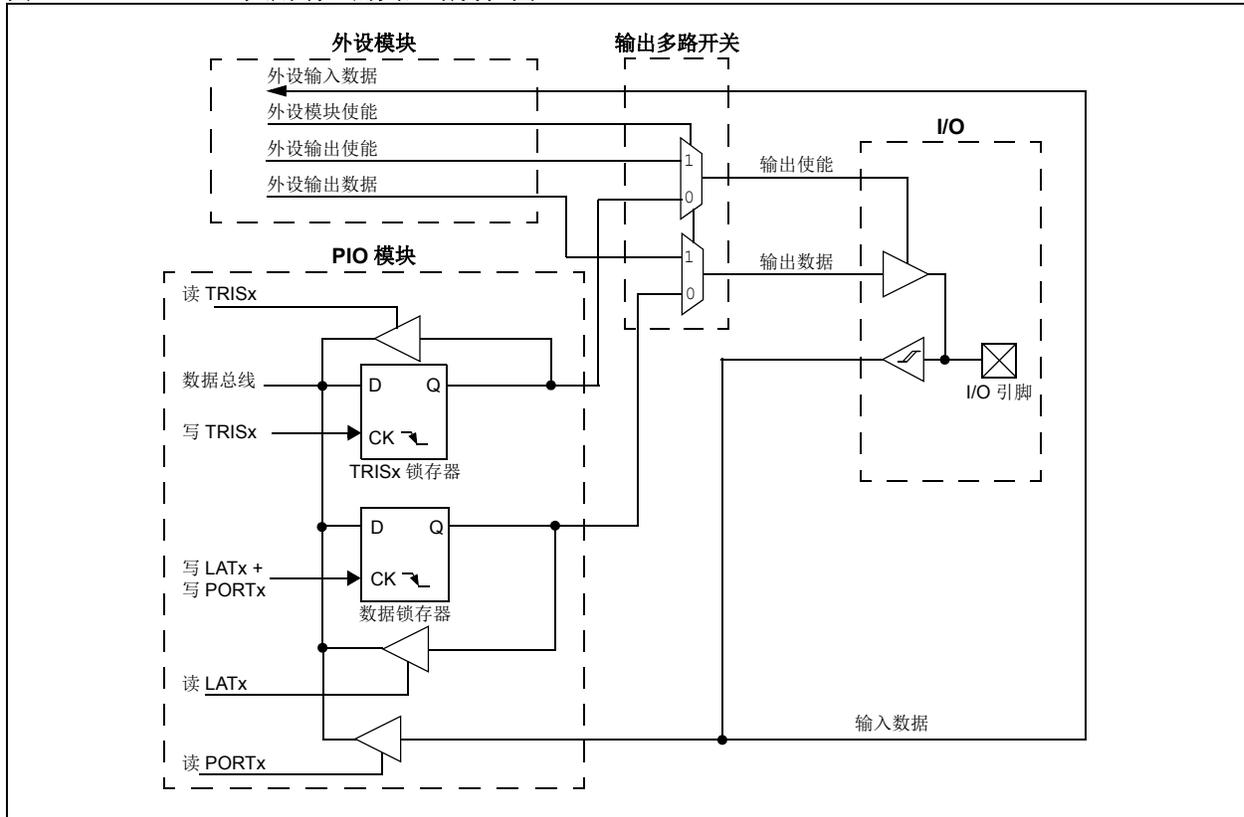
当使能某外设并驱动与其相对应的引脚时，将禁止此引脚的通用输出功能。可以读该I/O引脚，但并行端口位的输出驱动器将被禁止。若使能某外设但没有驱动引脚时，则该引脚可由一个端口驱动。

所有端口引脚都有三个与端口引脚作为数字I/O时的工作直接相关的寄存器，和一个与端口引脚作为模拟输入时的工作相关的寄存器。数据方向寄存器（TRIS）决定引脚是输入引脚还是输出引脚。如果数据方向位为1，则为输入引脚。复位以后，所有端口引脚被定义为输入引脚。读输出锁存寄存器（LAT）时，读到的是锁存器中的值；写锁存器时，写入的是锁存器。但读取端口（PORT）时，读到的是端口引脚的值；而写入端口引脚时，写入的是相应的锁存器。

对于某个特定器件，无效的位及其相关的数据和控制寄存器都将被禁止。这意味着相应的LAT<sub>x</sub>和TRIS<sub>x</sub>寄存器以及该端口引脚将读为0。

当一个定义为只用作输入的引脚与另一个外设或功能复用，由于没有其他竞争的输入源，它将被视为专用端口。

图 11-1: 共用端口的典型结构框图



# PIC24FJ128GB204 系列

## 11.1.1 I/O 端口写 / 读时序

在改变端口方向或对端口执行写操作，与对同一端口执行读操作之间需要间隔一个指令周期。通常在两者之间插一条 NOP 指令。

## 11.1.2 漏极开路配置

除 PORTx、LATx 和 TRISx 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对应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。

这种开漏特性允许通过使用外部上拉电阻在任何只能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 VIH 规范相同。

## 11.2 配置模拟端口引脚（ANSx）

ANSx 和 TRISx 寄存器用于控制具有模拟功能的引脚的操作。每个具有模拟功能的端口引脚都与某个 ANSx 位相关（见寄存器 11-1 至寄存器 11-3），该位决定了引脚功能是模拟的还是数字的。有关不同 ANSx 和 TRISx 位设置时引脚的详细操作，请参见表 11-1。

当读取 PORTx 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）。

### 11.2.1 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。多数输入引脚能够承受最高 5.5V 的直流电压，这是数字逻辑电路的典型电平值。但是，一些引脚只能承受最高 VDD 的电压。应避免使这些引脚上的电压超出 VDD。

表 11-2 总结了不同的电压承受能力。更多详细信息，请参见第 33.0 节“电气特性”。

表 11-1: 配置 I/O 引脚的模拟 / 数字功能

引脚功能	ANSx 设置	TRISx 设置	备注
模拟输入	1	1	建议保持 ANSx = 1。
模拟输出	1	1	建议保持 ANSx = 1。
数字输入	0	1	固件必须在将引脚配置为数字输入后等待至少一个指令周期，然后才能读取有效输入值。
数字输出	0	0	请确保禁止该引脚上的模拟输出功能（如果具备的话）。

表 11-2: 端口或引脚可承受的输入电压说明

端口或引脚	可承受的输入电压	说明
PORTA<10:7,4> <sup>(1)</sup>	5.5V	可承受高于 VDD 的输入电平，这对于大部分标准逻辑有用。
PORTB<11:10,8:4>		
PORTC<9:3> <sup>(1)</sup>		
PORTA<3:0>	VDD	只能承受 VDD 输入电平。
PORTB<15:13,9,3:0>		
PORTC<2:0> <sup>(1)</sup>		

注 1: 所有引脚在 28 引脚器件上未全部实现。关于端口引脚实现的完整描述，请参见第 1.0 节“器件概述”。

# PIC24FJ128GB204 系列

## 寄存器 11-1: ANSA: PORTA 模拟功能选择寄存器

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/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	ANSA<3:0>			
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 15-4 未实现: 读为 0

bit 3-0 **ANSA<3:0>**: PORTA 模拟功能选择位

1 = 引脚配置为模拟模式; 禁止读 I/O 端口

0 = 引脚配置为数字模式; 使能读 I/O 端口

## 寄存器 11-2: ANSB: PORTB 模拟功能选择寄存器

R/W-1	R/W-1	R/W-1	U-0	U-0	U-0	R/W-1	U-0
ANSB<15:13>			—	—	—	ANSB9	—
bit 15							bit 8

U-0	R/W-1	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
—	ANSB6	—	—	ANSB<3:0>			
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 15-13 **ANSB<15:13>**: PORTB 模拟功能选择位

1 = 引脚配置为模拟模式; 禁止读 I/O 端口

0 = 引脚配置为数字模式; 使能读 I/O 端口

bit 12-10 未实现: 读为 0

bit 9 **ANSB9**: PORTB 模拟功能选择位

1 = 引脚配置为模拟模式; 禁止读 I/O 端口

0 = 引脚配置为数字模式; 使能读 I/O 端口

bit 8-7 未实现: 读为 0

bit 6 **ANSB6**: PORTB 模拟功能选择位

1 = 引脚配置为模拟模式; 禁止读 I/O 端口

0 = 引脚配置为数字模式; 使能读 I/O 端口

bit 5-4 未实现: 读为 0

bit 3-0 **ANSB<3:0>**: PORTB 模拟功能选择位

1 = 引脚配置为模拟模式; 禁止读 I/O 端口

0 = 引脚配置为数字模式; 使能读 I/O 端口

# PIC24FJ128GB204 系列

寄存器 11-3: ANSC: PORTC 模拟功能选择寄存器 <sup>(1)</sup>

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-1	R/W-1
—	—	—	—	—	ANSC<2:0>		
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 15-3      **未实现:** 读为 0  
 bit 2-0      **ANSC<2:0>:** 模拟功能选择位  
                     1 = 引脚配置为模拟模式; 禁止读 I/O 端口  
                     0 = 引脚配置为数字模式; 使能读 I/O 端口

**注 1:** 这些引脚在 28 引脚器件上不可用。

## 11.3 输入电平变化通知 (ICN)

I/O 端口的输入电平变化通知功能允许 PIC24FJ128GB204 系列器件在选定输入引脚的电平状态变化 (Change-Of-State, COS) 时向处理器发出中断请求。即便是在时钟被禁止的休眠模式下, 该特性也可检测到输入电平状态变化。根据器件的引脚数, 最多可以选择 (允许) 82 个外部输入在电平状态发生变化时产生中断请求。

CNEN1 至 CNEN3 寄存器包含每个 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 引脚都有一个与之相连的弱上拉和弱下拉电路。弱上拉电路提供连接到该引脚的拉电流, 而弱下拉电路提供连接到该引脚的灌电流。这样的话, 当连接按钮或键盘设备时, 不再需要使用外部电阻。使用 CNPU1 至 CNPU3 寄存器可单独使能上拉电路, 使用 CNPD1 至 CNPD3 寄存器可单独使能下拉电路。每个 CN 引脚都为其上拉电路和下拉电路提供了独立的控制位。将某个控制位置 1 可使其对应引脚的弱上拉或弱下拉电路。

选择内部上拉电路时, 引脚电平将拉至  $V_{DD} - 1.1V$  (典型值)。当选择内部下拉电路时, 引脚电平下拉至  $V_{SS}$ 。

**注:** 只要端口引脚被配置为数字输出引脚, 输入电平变化通知引脚上的弱上拉电路将始终被禁止。

### 例 11-1: 端口读 / 写 (汇编语言)

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

### 例 11-2: 端口读 / 写 (C 语言)

```
TRISB = 0xFF00;           // Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
Nop();                    // Delay 1 cycle
If (PORTBbits.RB13){ };   // Next Instruction
```

# PIC24FJ128GB204 系列

## 11.4 外设引脚选择 (PPS)

通用器件的最大挑战是在最小化 I/O 引脚上功能冲突的同时，提供最大可能的外设功能集。在需要使用复用同一个引脚的多个外设的应用中，对应用代码进行繁琐的更改或彻底重新设计可能是惟一的选择。

外设引脚选择 (PPS) 功能通过使能用户外设集选择并将外设功能放置到大量的 I/O 引脚中提供了一个替代这两种选择的方案。通过增加特定器件上的引脚配置选项，用户可以更好地调节单片机以满足整个应用的需要，而不是调整应用来满足器件。

外设引脚选择功能通过固定数量的数字 I/O 引脚进行操作。用户可将任一数字外设的输入和 / 或输出独立映射到这些 I/O 引脚之一。外设引脚选择通过软件执行，通常不需要器件重新编程。器件带有保护硬件，可防止在建立了外设映射后意外或误更改此映射。

### 11.4.1 可用引脚

外设引脚选择功能可在最多 44 个引脚的范围内使用；可用的引脚数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其完整引脚名称中有“RPn”或“RPI n”标识，其中“n”指的是可重映射的引脚的编号。“RP”用于指定支持可重映射输入和输出功能的引脚，而“RPI”指定仅支持可重映射输入功能的引脚。

PIC24FJ128GB204 系列器件支持的可重映射仅输入引脚数远大于可重映射输入 / 输出引脚数。在此器件系列中，最多有 25 个可重映射输入 / 输出引脚，具体取决于所选特定器件的引脚数；这些引脚编号为 RP0 至 RP25。

请参见表 1-3，了解各封装提供的引脚排列选项汇总。

### 11.4.2 可用外设

外设引脚选择管理的外设都是仅数字外设，包括通用串行通信接口 (UART 和 SPI)、通用定时器时钟输入、定时器相关外设 (输入捕捉和输出比较)，以及外部中断输入。由于比较器模块的输出为离散数字信号，因此也包括在内。

PPS 不适用于以下外设：

- I<sup>2</sup>C™ (输入和输出)
- USB (所有模块输入和输出)
- 电平变化通知输入
- RTCC 闹钟输出
- EPMP 信号 (输入和输出)
- 模拟 (输入和输出)
- INTO

有引脚选择和无引脚选择的外设的最大区别是有引脚选择的外设与默认 I/O 引脚之间无关联。必须在使用外设前始终将其分配给指定的 I/O 引脚。相反，假设无引脚选择的外设处于活动状态且未与其他外设发生冲突，则此外设可始终通过默认引脚使用。

### 11.4.2.1 外设引脚选择功能优先级

可选引脚的外设输出 (例如 OC 和 UART 发送) 的优先级高于该引脚上的所有通用数字功能 (例如 EPMP 和端口 I/O)。特殊的数字输出 (例如，USB 使能器件上的 USB) 的优先级高于同一引脚上的 PPS 输出。本数据手册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 位设置或固定外设控制。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。

### 11.4.3 控制外设引脚选择

PPS 功能是通过以下两组特殊功能寄存器 (SFR) 控制的：一组用于映射外设输入，另一组用于映射输出。由于输入和输出是单独控制的，因此特定外设的输入和输出 (若该外设都有) 均可施加到任何可选的功能引脚上，而没有限制。

根据映射的是输入还是输出，有两种方法可处理外设与外设可选引脚之间的关联。

## 11.4.3.1 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映射到的引脚。RPINRx 寄存器用于配置外设输入映射（见寄存器 11-4 至寄存器 11-22）。

每个寄存器均包含两组 6 位位域，每组都与一个可选引脚的外设相关。给指定外设的位域赋上正确的 6 位值，会将具有此值的 RPn/RPIn 引脚映射到该外设。对于任何给定器件，任何位域值的有效范围对应于此器件所支持的外设引脚选择的最大值。

**表 11-3: 可选输入源（将输入映射到功能）<sup>(1)</sup>**

输入名称	功能名称	寄存器	功能映射位
DSM 调制输入	MDMIN	RPINR30	MDMIR<5:0>
DSM 载波 1 输入	MDCIN1	RPINR31	MDC1R<5:0>
DSM 载波 2 输入	MDCIN2	RPINR31	MDC2R<5:0>
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>
输出比较故障 A	OCFA	RPINR11	OCFAR<5:0>
输出比较故障 B	OCFB	RPINR11	OCFBR<5:0>
输出比较触发 1	OCTRIG1	RPINR0	OCTRIG1R<5:0>
输出比较触发 2	OCTRIG2	RPINR2	OCTRIG2R<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	RPINR28	SCK3R<5:0>
SPI3 数据输入	SDI3	RPINR28	SDI3R<5:0>
SPI3 从选择输入	SS3IN	RPINR29	SS3R<5:0>
通用定时器外部时钟	TMRCK	RPINR23	TMRCKR<5:0>
UART1 允许发送	U1CTS	RPINR18	U1CTSR<5:0>
UART1 接收	U1RX	RPINR18	U1RXR<5:0>
UART2 允许发送	U2CTS	RPINR19	U2CTSR<5:0>
UART2 接收	U2RX	RPINR19	U2RXR<5:0>
UART3 允许发送	U3CTS	RPINR21	U3CTSR<5:0>
UART3 接收	U3RX	RPINR17	U3RXR<5:0>
UART4 允许发送	U4CTS	RPINR27	U4CTSR<5:0>
UART4 接收	U4RX	RPINR27	U4RXR<5:0>

**注 1:** 除非另外声明，否则所有输入均使用施密特触发器（ST）输入缓冲器。

# PIC24FJ128GB204 系列

## 11.4.3.2 输出映射

与输入相反，外设引脚选择选项的输出根据引脚进行映射。这种情况下，与特定引脚相关的控制寄存器指示要映射的外设输出。RPORx 寄存器用于控制输出映射。每个寄存器均包含两组 6 位位域；各位域都与一个 RPn 引脚相关（见寄存器 11-23 至寄存器 11-35）。位域值对

应一个外设，该外设的输出映射到引脚（见表 11-4）。

由于采用的映射技术，输出映射的外设列表中还包含一个空值 000000。此值允许任何给定引脚保持与所有可选引脚的外设的输出断开。

表 11-4: 可选输出源（将功能映射到输出）

输出功能编号 <sup>(1)</sup>	功能	输出名称
0	NULL <sup>(2)</sup>	空
1	C1OUT	比较器 1 输出
2	C2OUT	比较器 2 输出
3	U1TX	UART1 发送
4	$\overline{U1RTS}$ <sup>(3)</sup>	UART1 请求发送
5	U2TX	UART2 发送
6	$\overline{U2RTS}$ <sup>(3)</sup>	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
13	OC1	输出比较 1
14	OC2	输出比较 2
15	OC3	输出比较 3
16	OC4	输出比较 4
17	OC5	输出比较 5
18	OC6	输出比较 6
19	U3TX	UART3 发送
20	U3RTS	UART3 请求发送
21	U4TX	UART4 发送
22	$\overline{U4RTS}$ <sup>(3)</sup>	UART4 请求发送
23	SDO3	SPI3 数据输出
24	SCK3OUT	SPI3 时钟输出
25	SS3OUT	SPI3 从选择输出
26	C3OUT	比较器 3 输出
27	MDOUT	DSM 调制器输出

- 注
- 1: 使用列出的值设置 RPORx 寄存器会将输出功能指定到相应的 RPn 引脚。
  - 2: 在器件复位时将 NULL 功能分配给所有的 RPn 输出，并禁止 RPn 输出功能。
  - 3: IrDA<sup>®</sup> BCLK 功能使用此输出。

### 11.4.3.3 映射限制

外设引脚选择的控制模式相当灵活。除阻止由两个配置为相同输入功能的物理引脚或配置给同一引脚的两个输出功能引起的信号冲突的系统电路外，无其他依靠硬件的输出锁定电路。此灵活性可扩展到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。

### 11.4.3.4 PIC24FJ128GB204 系列器件的映射例外

尽管 PPS 寄存器理论上可具有最多 24 个可重映射 I/O 引脚，但并非在所有器件中都全部实现。对于 PIC24FJ128GB204 系列器件，可用的可重映射引脚的最大数量为 24，其中包含 1 个仅输入引脚。表 11-5 总结了可用的可重映射引脚的差异。

当开发使用可重映射引脚的应用时，用户还应记住以下事项：

- 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；对应的模块将不具有映射到它的输入。
- 对于 RPORx 寄存器，对应未实现引脚的位域也未实现；写入这些位域不会有任何影响。

### 11.4.4 控制配置更改

由于可在运行时更改外设重映射，因此需要对外设重映射设置一些限制条件以防止意外更改配置。PIC24F 器件有以下三种用于阻止更改外设映射的功能：

- 控制寄存器锁定序列
- 连续状态监视
- 配置位重映射锁定

### 11.4.4.1 控制寄存器锁定

正常工作状态下，不允许写 RPINRx 和 RPORx 寄存器。尝试的写操作看似正常执行，但寄存器的内容并没有发生变化。要更改这些寄存器的内容，寄存器必须用硬件解锁。寄存器锁定由 IOLOCK 位（OSCCON<6>）控制。将 IOLOCK 置 1 将阻止写入控制寄存器；而将 IOLOCK 清零将允许写入。

要置 1 或清零 IOLOCK，必须执行以下指定命令序列：

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 LOCK 位的类似序列不同，IOLOCK 在更改前一直保持一种状态。这允许使用一个解锁序列对所有外设引脚选择进行配置，然后对所有控制寄存器进行更新，最后用第二个锁定序列锁定。

### 11.4.4.2 连续状态监视

除了阻止直接写入外，RPINRx 和 RPORx 寄存器的内容还由影子寄存器通过硬件不停地进行监视。如果任何寄存器发生了不希望的更改（例如，由 ESD 或其他外部事件引起的电池干扰），则将触发配置失配复位。

### 11.4.4.3 配置位引脚选择锁定

作为又一层保护，可配置器件以阻止对 RPINRx 和 RPORx 寄存器执行多次写会话。IOL1WAY（CW4<15>）配置位会阻止 IOLOCK 位在置 1 后再被清零。若 IOLOCK 保持置 1 状态，寄存器解锁过程将不会执行，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重映射的惟一方法是执行器件复位。

默认（未编程）状态下，IOL1WAY 置 1，限制用户只能进行一次写会话。编程 IOL1WAY 允许用户对外设引脚选择寄存器进行不受限制的访问（通过正确地使用解锁序列）。

表 11-5: PIC24FJ128GB204 系列器件的可重映射引脚的例外情况

器件	RPn 引脚 (I/O)		RPI n 引脚	
	总共	未实现	总共	未实现
PIC24FJXXXGB202	14	RP4 和 RP12	1	—
PIC24FJXXXGB204	24	RP4 和 RP12	1	—

# PIC24FJ128GB204 系列

## 11.4.5 外设引脚选择注意事项

控制外设引脚选择的功能需要注意应用设计中几个容易被忽视的事项。对于仅可用作可重映射外设的几个常用外设尤其如此。

主要问题是在器件默认（复位）状态下外设引脚选择功能在默认引脚上不可用。由于所有 **RPINRx** 寄存器复位为 111111 且所有 **RPORx** 寄存器复位为 000000，因此所有外设引脚选择输入与 **Vss** 相连，而所有外设引脚选择输出断开连接。

此情景需要用户在执行任何其他应用代码前使用正确的外设配置初始化器件。由于 **IOLOCK** 位复位为解锁状态，所以不必在器件退出复位后执行解锁序列。但是，考虑到应用的安全性，最好在写入控制寄存器之后将 **IOLOCK** 位置 1 并锁定配置。

由于解锁序列对时序有严格要求，因此必须用汇编语言编程，执行方式与更改振荡器配置的方式相同。若应用程序是用 **C** 或其他高级语言编写的，则仍应通过编写行内汇编代码执行解锁序列。

选择配置需要查看所有外设引脚选择及其引脚分配，尤其是那些未在应用中使用的引脚。在所有情况下，未用的可选引脚的外设应被完全禁止。未用外设的输入应分配给未用的 **RPn/RPIn** 引脚功能。具有未用的 **RPn** 功能的 **I/O** 引脚应配置为空外设输出。

将外设分配给特定引脚的操作不能自动执行引脚 **I/O** 电路的任何其他配置。理论上，也就是说将可选引脚的输出添加到某个引脚意味着在驱动输出时可能会无意中驱动现有的外设输入。用户必须熟悉其他共用一个可重映射引脚的固定外设的行为，了解何时使能或禁止这些外设。为了安全起见，共用一个引脚的固定数字外设在不应用时应禁止。

遵照这些方针，对特定外设的可重映射引脚进行配置并不会自动使能该功能。必须特别配置外设实现特定的操作并使能外设，如同将外设连接到固定引脚时一样。此操作在应用代码中的位置（紧跟在器件复位和外设配置之后或在主应用程序内）取决于外设及其在应用中的使用。

最后一个注意事项是外设引脚选择功能既不改写模拟输入也不会将具有模拟功能的引脚重新配置为数字 **I/O**。若某个引脚在器件复位时配置为模拟输入，则在使用该引脚的外设引脚选择功能时，必须明确将此引脚重新配置为数字 **I/O**。

例 11-3 所示为使用 **UART1** 实现具有流控制的双向通信的配置。使用了以下输入和输出功能：

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

### 例 11-3: 配置 UART1 输入和输出功能

```
// Unlock Registers
asm volatile ("MOV #OSCCON, w1 \n"
             "MOV #0x46, w2 \n"
             "MOV #0x57, w3 \n"
             "MOV.b w2, [w1] \n"
             "MOV.b w3, [w1] \n"
             "BCLR OSCCON, #6") ;

// or use C30 built-in macro:
// __builtin_write_OSCCONL(OSCCON & 0xbf);

// Configure Input Functions (Table 11-3)
// Assign U1RX To Pin RP0
RPINR18bits.U1RXR = 0;

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

// Configure Output Functions (Table 11-4)
// Assign U1TX To Pin RP2
RPOR1bits.RP2R = 3;

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

// Lock Registers
asm volatile ("MOV #OSCCON, w1 \n"
             "MOV #0x46, w2 \n"
             "MOV #0x57, w3 \n"
             "MOV.b w2, [w1] \n"
             "MOV.b w3, [w1] \n"
             "BSET OSCCON, #6") ;

// or use C30 built-in macro:
// __builtin_write_OSCCONL(OSCCON | 0x40);
```

# PIC24FJ128GB204 系列

## 11.4.6 外设引脚选择寄存器

PIC24FJ128GB204 系列器件共实现了 32 个寄存器用于配置可重映射的外设：

- 输入可重映射外设寄存器 (19)
- 输出可重映射外设寄存器 (13)

**注：** 仅在 IOLOCK (OSCCON<6>) = 0 时才能更改输入和输出寄存器的值。请参见第 11.4.4.1 节“控制寄存器锁定”了解特定的命令序列。

### 寄存器 11-4: 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	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCTRIG1R5	OCTRIG1R4	OCTRIG1R3	OCTRIG1R2	OCTRIG1R1	OCTRIG1R0
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) 分配给相应的 RPn 或 RPIn 引脚的位
- bit 7-6      **未实现：** 读为 0
- bit 5-0      **OCTRIG1R<5:0>：** 将输出比较触发 1 分配给相应的 RPn 或 RPIn 引脚的位

### 寄存器 11-5: 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) 分配给相应的 RPn 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) 分配给相应的 RPn 或 RPIn 引脚的位

# PIC24FJ128GB204 系列

**寄存器 11-6: RPINR2: 外设引脚选择输入寄存器 2**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	
—	—	OCTRIG2R5	OCTRIG2R4	OCTRIG2R3	OCTRIG2R2	OCTRIG2R1	OCTRIG2R0	
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 15-14      **未实现:** 读为 0
- bit 13-8      **OCTRIG2R<5:0>:** 将输出比较触发 2 分配给相应的 RPn 或 RPI n 引脚的位
- bit 7-6        **未实现:** 读为 0
- bit 5-0        **INT4R<5:0>:** 将外部中断 4 (INT4) 分配给相应的 RPn 或 RPI n 引脚的位

**寄存器 11-7: 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) 分配给相应的 RPn 或 RPI n 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) 分配给相应的 RPn 或 RPI n 引脚的位

# PIC24FJ128GB204 系列

**寄存器 11-8: 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) 分配给相应的 RPn 或 RPI n 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) 分配给相应的 RPn 或 RPI n 引脚的位

**寄存器 11-9: 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) 分配给相应的 RPn 或 RPI n 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) 分配给相应的 RPn 或 RPI n 引脚的位

# PIC24FJ128GB204 系列

**寄存器 11-10: 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) 分配给相应的 RPn 或 RPIIn 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) 分配给相应的 RPn 或 RPIIn 引脚的位

**寄存器 11-11: 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
—	—	U3RXR<5:0>					—
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) 分配给相应的 RPn 或 RPIIn 引脚的位
- bit 7-0     **未实现:** 读为 0

# PIC24FJ128GB204 系列

## 寄存器 11-12: 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 ( $\overline{\text{U1CTS}}$ ) 分配给相应的 RPn 或 RPI n 引脚的位
- bit 7-6     未实现: 读为 0
- bit 5-0     **U1RXR<5:0>**: 将 UART1 接收 (U1RX) 分配给相应的 RPn 或 RPI n 引脚的位

## 寄存器 11-13: 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 ( $\overline{\text{U2CTS}}$ ) 分配给相应的 RPn 或 RPI n 引脚的位
- bit 7-6     未实现: 读为 0
- bit 5-0     **U2RXR<5:0>**: 将 UART2 接收 (U2RX) 分配给相应的 RPn 或 RPI n 引脚的位

# PIC24FJ128GB204 系列

## 寄存器 11-14: 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							bit 8

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							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      **SCK1R<5:0>:** 将 SPI1 时钟输入 (SCK1IN) 分配给相应的 RPn 或 RPIIn 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) 分配给相应的 RPn 或 RPIIn 引脚的位

## 寄存器 11-15: 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							bit 8

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							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      **U3CTSR<5:0>:** 将 UART3 允许发送 (U3CTS) 分配给相应的 RPn 或 RPIIn 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) 分配给相应的 RPn 或 RPIIn 引脚的位

# PIC24FJ128GB204 系列

## 寄存器 11-16: 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) 分配给相应的 RPn 或 RPI n 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) 分配给相应的 RPn 或 RPI n 引脚的位

## 寄存器 11-17: RPINR23: 外设引脚选择输入寄存器 23

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	TMRCKR5	TMRCKR4	TMRCKR3	TMRCKR2	TMRCKR1	TMRCKR0
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 15-14    未实现: 读为 0
- bit 13-8    **TMRCKR<5:0>**: 将通用定时器外部输入 (TMRCK) 分配给相应的 RPn 或 RPI n 引脚的位
- bit 7-6     未实现: 读为 0
- bit 5-0     **SS2R<5:0>**: 将 SPI2 从选择输入 (SS2IN) 分配给相应的 RPn 或 RPI n 引脚的位

# PIC24FJ128GB204 系列

## 寄存器 11-18: 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 允许发送输入 (U4CTS) 分配给相应的 RPn 或 RPI n 引脚的位
- bit 7-6        **未实现:** 读为 0
- bit 5-0        **U4RXR<5:0>:** 将 UART4 接收输入 (U4RX) 分配给相应的 RPn 或 RPI n 引脚的位

## 寄存器 11-19: 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) 分配给相应的 RPn 或 RPI n 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) 分配给相应的 RPn 或 RPI n 引脚的位

# PIC24FJ128GB204 系列

## 寄存器 11-20: 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
—	—	SS3R<5:0>					
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 15-6      未实现: 读为 0  
 bit 5-0      **SS3R<5:0>**: 将 SPI3 从选择输入 (SS3IN) 分配给相应的 RPn 或 RPI n 引脚的位

## 寄存器 11-21: RPINR30: 外设引脚选择输入寄存器 30

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
—	—	MDMIR<5:0>					
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 15-6      未实现: 读为 0  
 bit 5-0      **MDMIR<5:0>**: 将发送调制输入 (MDMI) 分配给相应的 RPn 或 RPI n 引脚的位

# PIC24FJ128GB204 系列

寄存器 11-22: **RPINR31: 外设引脚选择输入寄存器 31**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	MDC2R5	MDC2R4	MDC2R3	MDC2R2	MDC2R1	MDC2R0
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
—	—	MDC1R5	MDC1R4	MDC1R3	MDC1R2	MDC1R1	MDC1R0
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 **MDC2R<5:0>:** 将发送载波 2 输入 (MDCIN2) 分配给相应的 RPn 或 RPI n 引脚的位

bit 7-6 **未实现:** 读为 0

bit 5-0 **MDC1R<5:0>:** 将发送载波 1 输入 (MDCIN1) 分配给相应的 RPn 或 RPI n 引脚的位

## 寄存器 11-23: 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 11-4 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 11-4 了解外设功能编号)。

## 寄存器 11-24: 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 11-4 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 11-4 了解外设功能编号)。

# PIC24FJ128GB204 系列

## 寄存器 11-25: 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
—	—	RP5R<5:0>					
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      **RP5R<5:0>:** RP5 输出引脚映射位  
 将外设输出编号 n 分配给引脚 RP5 (请参见表 11-4 了解外设功能编号)。
- bit 7-0        **未实现:** 读为 0

## 寄存器 11-26: 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 11-4 了解外设功能编号)。
- bit 7-6        **未实现:** 读为 0
- bit 5-0        **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 11-4 了解外设功能编号)。

# PIC24FJ128GB204 系列

## 寄存器 11-27: 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 11-4 了解外设功能编号)。
- bit 7-6     **未实现:** 读为 0
- bit 5-0     **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 11-4 了解外设功能编号)。

## 寄存器 11-28: 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 11-4 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 11-4 了解外设功能编号)。

# PIC24FJ128GB204 系列

## 寄存器 11-29: 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
—	—	RP13R<5:0>					
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     **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 11-4 了解外设功能编号)。
- bit 7-0      **未实现:** 读为 0

## 寄存器 11-30: 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	RP15R4	RP15R3	RP15R2	RP15R1	RP15R0
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  
 将外设输出编号 n 分配给引脚 RP15 (请参见表 11-4 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 11-4 了解外设功能编号)。

# PIC24FJ128GB204 系列

## 寄存器 11-31: RPOR8: 外设引脚选择输出寄存器 8<sup>(1)</sup>

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 11-4 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 11-4 了解外设功能编号)。

注 1: 这些引脚在 28 引脚器件上不可用。

## 寄存器 11-32: RPOR9: 外设引脚选择输出寄存器 9<sup>(1)</sup>

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 11-4 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 11-4 了解外设功能编号)。

注 1: 这些引脚在 28 引脚器件上不可用。

# PIC24FJ128GB204 系列

## 寄存器 11-33: RPOR10: 外设引脚选择输出寄存器 10<sup>(1)</sup>

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 11-4 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 11-4 了解外设功能编号)。

注 1: 这些引脚在 28 引脚器件上不可用。

## 寄存器 11-34: RPOR11: 外设引脚选择输出寄存器 11<sup>(1)</sup>

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 11-4 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 11-4 了解外设功能编号)。

注 1: 这些引脚在 28 引脚器件上不可用。

# PIC24FJ128GB204 系列

寄存器 11-35: RPOR12: 外设引脚选择输出寄存器 12<sup>(1)</sup>

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 11-4 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 11-4 了解外设功能编号)。

注 1: 这些引脚在 28 引脚器件上不可用。

# PIC24FJ128GB204 系列

---

注:

## 12.0 TIMER1

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“定时器”（DS39704）。本数据手册中的信息取代了FRM中的信息。

Timer1 模块是一个 16 位定时器，可作为实时时钟（Real-Time Clock, RTC）的时间计数器，或作为独立运行的时隙定时器 / 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

而且，Timer1 还支持下列特性：

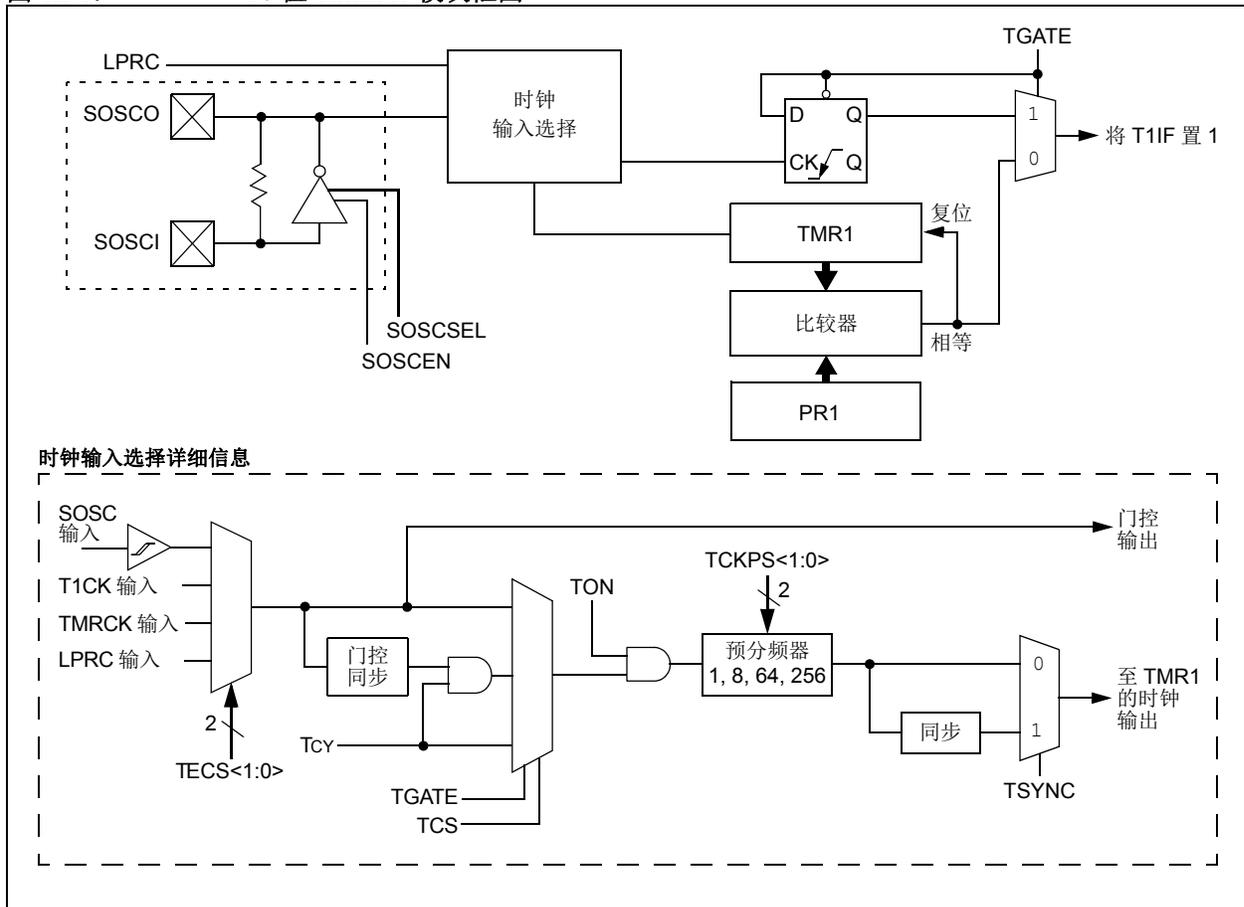
- 定时器门控操作
- 可选的预分频比设置
- CPU 空闲和休眠模式下的定时器操作
- 在 16 位周期寄存器匹配或外部门控信号的下降沿产生中断

图 12-1 给出了 16 位定时器模块的框图。

配置 Timer1 的步骤：

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、TECS<1:0> 和 TGATE 位设置时钟和门控模式。
4. 将 TSYNC 位置 1 或清零以配置同步或异步操作。
5. 将定时器周期值装入 PR1 寄存器。
6. 如果需要中断，则将 Timer1 中断允许位 T1IE 置 1。使用 Timer1 中断优先级位 T1IP<2:0> 来设置中断的优先级。

图 12-1: 16 位 TIMER1 模块框图



# PIC24FJ128GB204 系列

寄存器 12-1: T1CON: TIMER1 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
TON	—	TSIDL	—	—	—	TECS1	TECS0
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:** Timer1 空闲模式停止位  
               1 = 当器件进入空闲模式后, 模块停止工作  
               0 = 模块在空闲模式下继续工作
- bit 12-10    **未实现:** 读为 0
- bit 9-8       **TECS<1:0>:** Timer1 扩展时钟源选择位 (当 TCS = 1 时选择)  
               当 TCS = 1 时:  
               11 = 通用定时器 (TMRCK) 外部输入  
               10 = LPRC 振荡器  
               01 = T1CK 外部时钟输入  
               00 = SOSC  
               当 TCS = 0 时:  
               这些位被忽略; 定时器将内部系统时钟 (Fosc/2) 作为时钟源。
- bit 7         **未实现:** 读为 0
- bit 6         **TGATE:** Timer1 门控时间累加使能位  
               当 TCS = 1 时:  
               该位被忽略。  
               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

**注 1:** 定时器运行 (TON = 1) 时改变 T1CON 的值会导致定时器预分频计数器复位, 不建议这么做。

**寄存器 12-1: T1CON: TIMER1 控制寄存器<sup>(1)</sup> (续)**

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:

该位被忽略。

bit 1      **TCS:** Timer1 时钟源选择位

1 = 通过 TECS<1:0> 位选择扩展时钟

0 = 内部时钟 (Fosc/2)

bit 0      **未实现:** 读为 0

**注 1:** 定时器运行 (TON = 1) 时改变 T1CON 的值会导致定时器预分频计数器复位, 不建议这么做。

# PIC24FJ128GB204 系列

---

注:

## 13.0 TIMER2/3 和 TIMER4/5

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》中的“定时器”（DS39704）。本数据手册中的信息取代了FRM中的信息。

Timer2/3 和 Timer4/5 模块各为一个 32 位定时器，它们也可以被配置为 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 可以在以下三种模式下工作：

- 两个独立的 16 位定时器，可实现所有 16 位工作模式（异步计数器模式除外）
- 一个 32 位定时器
- 一个 32 位同步计数器

它们还支持以下功能：

- 定时器门控操作
- 可选的预分频比设置
- 空闲和休眠模式下作为定时器工作
- 在 32 位周期寄存器匹配时产生中断
- 触发 A/D 事件（仅 32 位模式的 Timer2/3 和 16 位模式的 Timer3）

在作为 16 位定时器时，所有 4 个定时器都可以单独用作同步定时器或计数器。它们也提供上面所列的功能，但 A/D 事件触发功能除外。该触发仅在 32 位模式的 Timer2/3 和 16 位模式的 Timer3 上实现。工作模式和使能的功能由 T2CON、T3CON、T4CON 和 T5CON 寄存器中的相应位的设置决定。寄存器 13-1 给出了 T2CON 和 T4CON 的一般形式；寄存器 13-2 给出了 T3CON 和 T5CON 的一般形式。

当工作在 32 位定时器/计数器模式时，Timer2 和 Timer4 作为低字，而 Timer3 和 Timer5 则作为高字。

**注：** 当定时器工作在 32 位模式时，将不使用 T3CON 和 T5CON 控制位。只有 T2CON 和 T4CON 控制位用于设置和控制。Timer2 和 Timer4 提供 32 位定时器模块的时钟和门控输入，但由 Timer3 或 Timer5 的中断标志触发产生模块中断。

要将 Timer2/3 或 Timer4/5 配置为 32 位工作模式：

1. 将 T32 或 T45 位置 1 (T2CON<3> 或 T4CON<3> = 1)。
2. 使用 TCKPS<1:0> 位选择 Timer2 或 Timer4 的预分频比。
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，则必须将 RPNRx (TxCK) 配置给可用的 RPN/RPIn 引脚。更多信息，请参见第 11.4 节“外设引脚选择 (PPS)”。
4. 装载定时器周期值。PR3 (或 PR5) 将存放该值的高字 (most significant word, msw)，而 PR2 (或 PR4) 则存放低字 (least significant word, lsw)。
5. 如果需要中断，则将 Timer3/5 中断允许位 T3IE 或 T5IE 置 1。使用 Timer3/5 中断优先级位 T3IP<2:0> 或 T5IP<2:0> 来设置中断的优先级。请注意，虽然 Timer2 或 Timer4 控制定时器，但其中断却表现为 Timer3 或 Timer5 中断。
6. 将 TON 位置 1 (= 1)。

任意时刻的定时器值都将存储在寄存器对 TMR<3:2> (或 TMR<5:4>) 中。TMR3 (TMR5) 始终存放计数值的高字，而 TMR2 (TMR4) 存放低字。

要将任何定时器配置为独立的 16 位工作模式：

1. 将与该定时器对应的位 T32 清零 (对于 Timer2 和 Timer3 来说是 T2CON<3>；对于 Timer4 和 Timer5 来说是 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 11.4 节“外设引脚选择 (PPS)”。
4. 将定时器周期值装入 PRx 寄存器。
5. 如果需要中断，则将 Timerx 中断允许位 TxIE 置 1。使用 Timerx 中断优先级位 TxIP<2:0> 来设置中断的优先级。
6. 将 TON 位置 1 (TxCON<15> = 1)。

# PIC24FJ128GB204 系列

图 13-1: TIMER2/3 和 TIMER4/5 (32 位) 框图

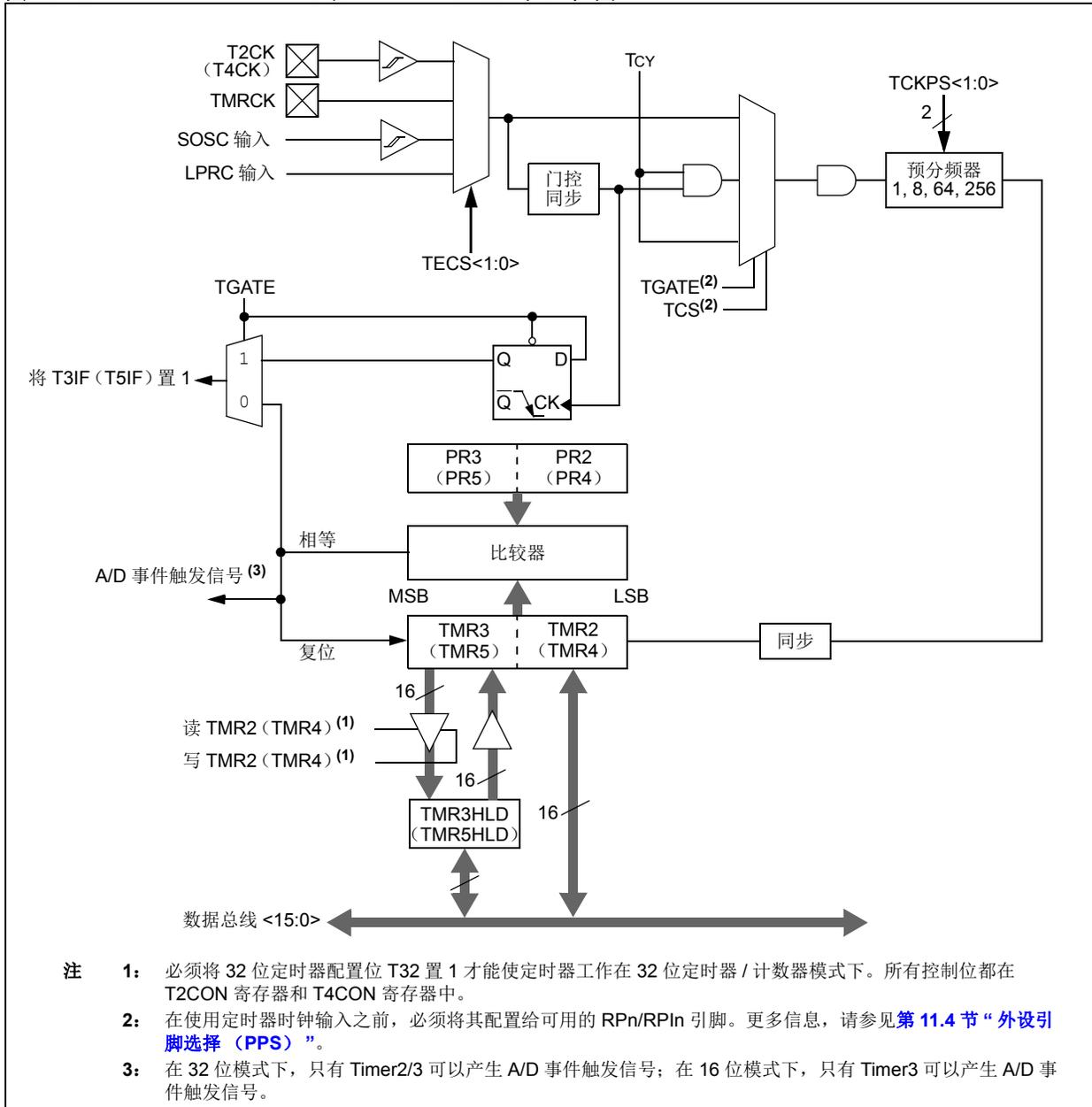


图 13-2: TIMER2 和 TIMER4 (16 位同步) 框图

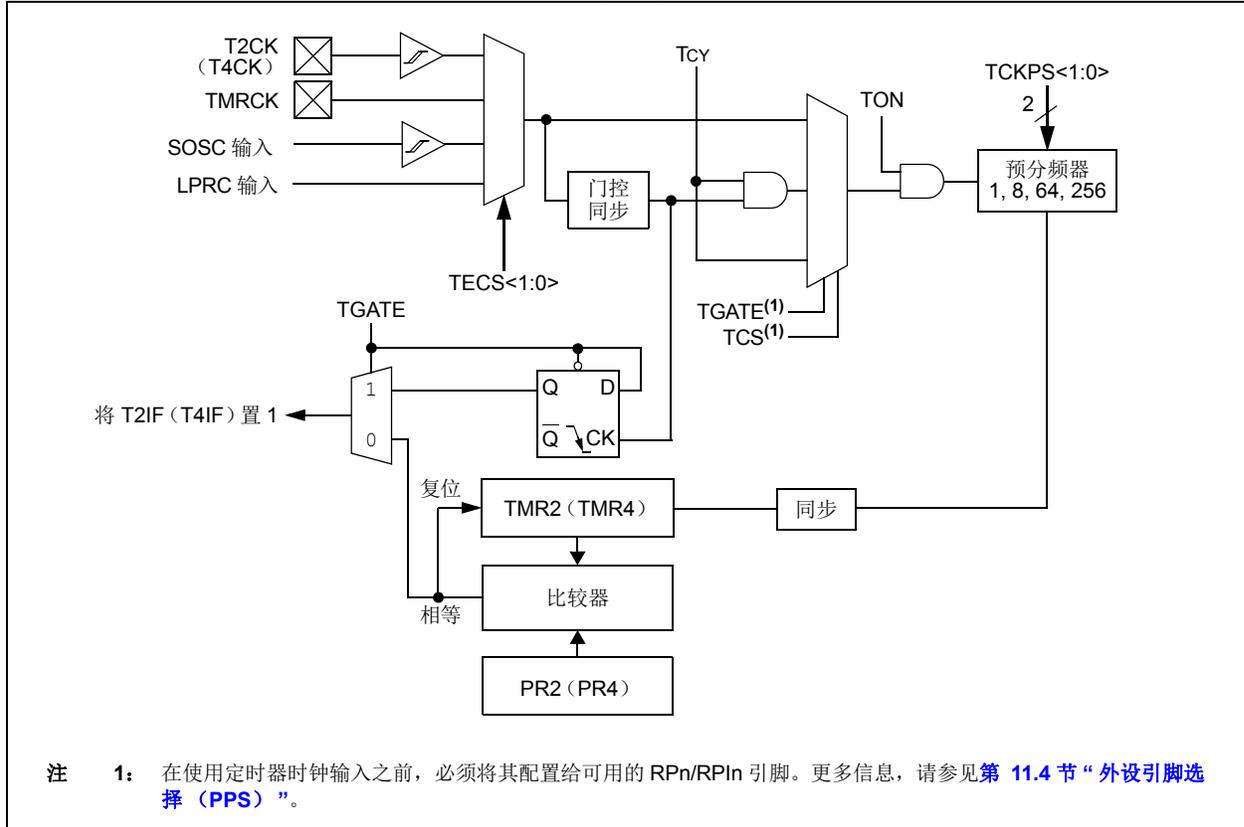
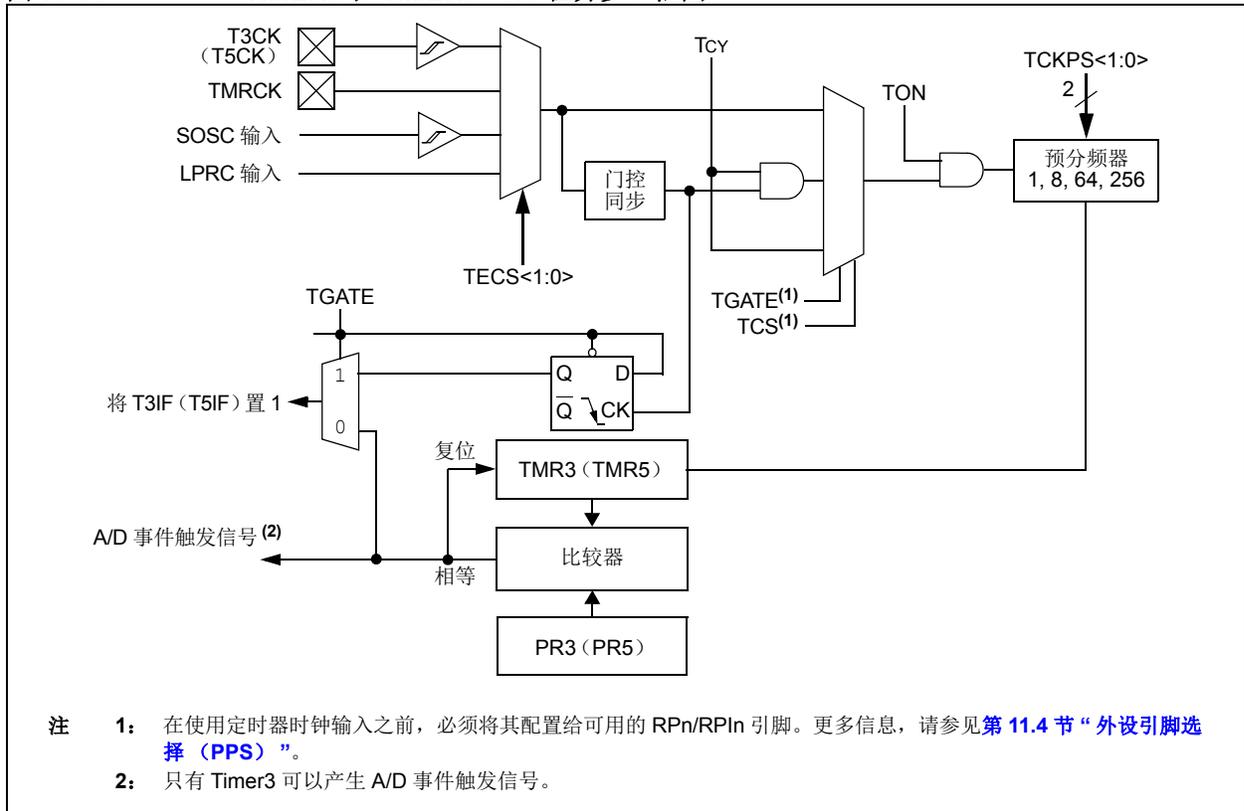


图 13-3: TIMER3 和 TIMER5 (16 位异步) 框图



# PIC24FJ128GB204 系列

寄存器 13-1: TxCON: TIMER2 和 TIMER4 控制寄存器<sup>(1)</sup>

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
TON	—	TSIDL	—	—	—	TECS1 <sup>(2)</sup>	TECS0 <sup>(2)</sup>
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 <sup>(3)</sup>	—	TCS <sup>(2)</sup>	—
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:** Timerx 空闲模式停止位  
                   1 = 当器件进入空闲模式后, 模块停止工作  
                   0 = 模块在空闲模式下继续工作
- bit 12-10        **未实现:** 读为 0
- bit 9-8            **TECS<1:0>:** Timerx 扩展时钟源选择位 (当 TCS = 1 时选择)<sup>(2)</sup>  
                   当 TCS = 1 时:  
                   11 = 通用定时器 (TMRCK) 外部输入  
                   10 = LPRC 振荡器  
                   01 = TxCK 外部时钟输入  
                   00 = SOSOC  
                   当 TCS = 0 时:  
                   忽略这些位; Timerx 定时器将内部系统时钟 (Fosc/2) 作为时钟源。
- bit 7             **未实现:** 读为 0
- bit 6             **TGATE:** Timerx 门控时间累加使能位  
                   当 TCS = 1 时:  
                   该位被忽略。  
                   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

- 注 1:** 定时器运行 (TON = 1) 时改变 TxCON 的值会导致定时器预分频计数器复位, 不建议这么做。
- 注 2:** 如果 TCS = 1 并且 TECS<1:0> = x1, 必须将所选的外部定时器输入 (TMRCK 或 TxCK) 配置给可用的 RPN/RPIN 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。
- 注 3:** 在 T4CON 中, 将实现 T45 位 (而非 T32 位) 来选择 32 位模式。在 32 位模式下, T3CON 或 T5CON 控制位不影响 32 位定时器的的工作。

## 寄存器 13-1: TxCON: TIMER2 和 TIMER4 控制寄存器<sup>(1)</sup> (续)

bit 3	<b>T32:</b> 32 位定时器模式选择位 <sup>(3)</sup> 1 = Timerx 和 Timery 构成一个 32 位定时器 0 = Timerx 和 Timery 作为两个 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	<b>未实现:</b> 读为 0
bit 1	<b>TCS:</b> Timerx 时钟源选择位 <sup>(2)</sup> 1 = 通过 TECS<1:0> 选择定时器源 0 = 内部时钟 (Fosc/2)
bit 0	<b>未实现:</b> 读为 0

- 注**
- 1: 定时器运行 (TON = 1) 时改变 TxCON 的值会导致定时器预分频计数器复位, 不建议这么做。
  - 2: 如果 TCS = 1 并且 TECS<1:0> = x1, 必须将所选的外部定时器输入 (TMRCK 或 TxCK) 配置给可用的 RPN/RPIn 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。
  - 3: 在 T4CON 中, 将实现 T45 位 (而非 T32 位) 来选择 32 位模式。在 32 位模式下, T3CON 或 T5CON 控制位不影响 32 位定时器的  
工作。

# PIC24FJ128GB204 系列

寄存器 13-2: TyCON: TIMER3 和 TIMER5 控制寄存器<sup>(1)</sup>

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
TON <sup>(2)</sup>	—	TSIDL <sup>(2)</sup>	—	—	—	TECS1 <sup>(2,3)</sup>	TECS0 <sup>(2,3)</sup>
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 <sup>(2)</sup>	TCKPS1 <sup>(2)</sup>	TCKPS0 <sup>(2)</sup>	—	—	TCS <sup>(2,3)</sup>	—
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<sup>(2)</sup>**  
 1 = 启动 16 位 Timery  
 0 = 停止 16 位 Timery
- bit 14            **未实现:** 读为 0
- bit 13            **TSIDL: Timery 空闲模式停止位<sup>(2)</sup>**  
 1 = 当器件进入空闲模式后, 模块停止工作  
 0 = 模块在空闲模式下继续工作
- bit 12-10        **未实现:** 读为 0
- bit 9-8           **TECS<1:0>: Timery 扩展时钟源选择位 (当 TCS = 1 时选择)<sup>(2,3)</sup>**  
 11 = 通用定时器 (TMRCK) 外部输入  
 10 = LPRC 振荡器  
 01 = TxCK 外部时钟输入  
 00 = SOSC
- bit 7             **未实现:** 读为 0
- bit 6             **TGATE: Timery 门控时间累加使能位<sup>(2)</sup>**  
当 TCS = 1 时:  
 该位被忽略。  
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<sup>(2)</sup>**  
 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<sup>(2,3)</sup>**  
 1 = TyCK 引脚的外部时钟的上升沿  
 0 = 内部时钟 (Fosc/2)
- bit 0             **未实现:** 读为 0

- 注**
- 1: 定时器运行 (TON = 1) 时改变 TyCON 的值会导致定时器预分频计数器复位, 不建议这么做。
  - 2: 当使能 32 位工作模式时 (T2CON<3> 或 T4CON<3> = 1), 这些位不会对 Timery 的工作产生影响。定时器的所有功能都通过 T2CON 和 T4CON 寄存器设置。
  - 3: 如果 TCS = 1 并且 TECS<1:0> = x1, 必须将所选的外部定时器输入 (TyCK) 配置给可用的 RPn/RPIn 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。

## 14.0 带专用定时器的输入捕捉

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》中的“带专用定时器的输入捕捉”（DS39722）。本数据手册中的信息取代了FRM中的信息。

PIC24FJ128GB204系列器件都具有6个独立的输入捕捉模块。每个模块都为捕捉外部脉冲事件和产生中断提供了各种配置和工作选项。

输入捕捉模块的主要特性包括：

- 通过级联两个相邻的模块，可采用硬件方式配置为所有的32位工作模式
- 输出比较操作有同步和触发两种模式，最多具有30个用户可选的同步/触发源
- 一个4级深的FIFO缓冲区，用于捕捉并保存多个事件的定时器值
- 可配置中断产生
- 为每个模块提供了最多6个时钟源，驱动独立的内部16位计数器

此模块可通过两个寄存器进行控制：ICxCON1（寄存器14-1）和ICxCON2（寄存器14-2）。模块的一般框图如图14-1所示。

## 14.1 一般工作模式

### 14.1.1 同步和触发模式

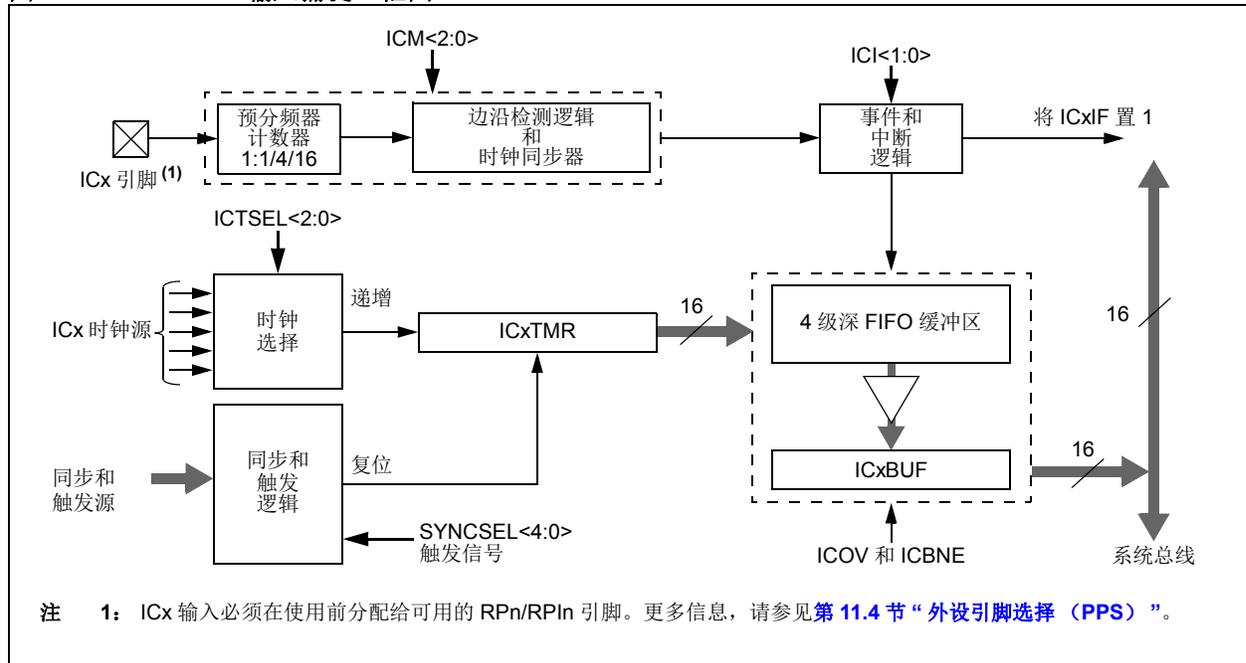
输入捕捉模块工作在自由运行模式下时，内部16位计数器ICxTMR可连续向上计数，并在每次溢出时从FFFh折回000h。其周期与所选的外部时钟源同步。发生捕捉事件时，将内部计数器的当前16位值写入FIFO缓冲区。

在同步模式下，一旦使能了所选时钟源，模块就开始捕捉ICx引脚上的事件。只要所选的同步源发生事件，内部计数器就会复位。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后，才允许内部计数器运行。

通过把SYNCSELx位（ICxCON2<4:0>）设置为00000并清零ICTRIG位（ICxCON2<7>）选择标准的自由运行工作模式。当SYNCSELx位设置为除00000以外的其他值时，将选择同步和触发模式。ICTRIG位选择同步或触发模式；置1该位将选择触发工作模式。在这两种模式下，SYNCSELx位用于决定同步/触发源。

当SYNCSELx位设置为00000且ICTRIG置1时，模块工作在软件触发模式下。在此情况下，可通过手动将TRIGSTAT位（ICxCON2<6>）置1来启动捕捉操作。

图 14-1: 输入捕捉 x 框图



# PIC24FJ128GB204 系列

## 14.1.2 级联（32 位）模式

默认情况下，每个模块都可以使用自己的 16 位定时器独立运行。要提高分辨率，可将相邻的偶数编号的模块和奇数编号的模块配置为一个 32 位模块。（例如，模块 1 和 2 配对，模块 3 和 4 配对等）。奇数编号的输入捕捉模块  $x$  (IC $x$ ) 提供 32 位寄存器对中的低 16 位，而偶数编号的输入捕捉模块  $y$  (IC $y$ ) 提供高 16 位。IC $x$  寄存器的折回操作会导致其相应的 IC $y$  寄存器递增。

通过将两个模块的 IC32 位 (IC $x$ CON2<8>) 置 1 使用硬件来配置级联操作。

## 14.2 捕捉操作

输入捕捉模块可配置为在 IC $x$  信号的上升沿或所有跳变沿处捕捉定时值并产生中断。捕捉可配置为在所有上升沿上发生，或仅在某些上升沿（例如每 4 个或 16 个上升沿）发生。可单独配置中断以在每个事件或一组事件发生时产生中断。

将模块设置为捕捉操作：

1. 将 IC $x$  输入配置为可用的外设引脚选择引脚之一。
2. 如果要使用同步模式，应在继续前禁止同步源。
3. 确保 ICBNE 位 (IC $x$ CON1<3>) 清零之前已通过读 IC $x$ BUF 从 FIFO 中除去了所有以前的数据。
4. 设置 SYNCSEL $x$  位 (IC $x$ CON2<4:0>) 以得到所需的同步 / 触发源。
5. 设置 ICTSEL $x$  位 (IC $x$ CON1<12:10>) 以得到所需的时钟源。
6. 设置 IC $i$  $x$  位 (IC $x$ CON1<6:5>) 以得到所需的中断频率。
7. 选择同步或触发工作模式：
  - a) 检查 SYNCSEL $x$  位是否设置为 00000。
  - b) 对于同步模式，清零 ICTRIG 位 (IC $x$ CON2<7>)。
  - c) 对于触发模式，置 1 ICTRIG 位并清零 TRIGSTAT 位 (IC $x$ CON2<6>)。
8. 设置 ICM $x$  位 (IC $x$ CON1<2:0>) 以得到所需的工作模式。
9. 使能所选的同步 / 触发源。

对于 32 位级联操作，设置过程稍有不同：

1. 将两个模块的 IC32 位 (IC $y$ CON2<8> 和 IC $x$ CON2<8>) 置 1，首先使能偶数编号的模块。这可以确保模块一起启动。
2. 分别设置两个模块的 ICTSEL $x$  和 SYNCSEL $x$  位以选择相同的同步 / 触发模式和时基源。首先设置偶数编号的模块，然后设置奇数编号的模块。两个模块必须使用相同的 ICTSEL $x$  和 SYNCSEL $x$  位设置。
3. 清零偶数编号的模块的 ICTRIG 位 (IC $y$ CON2<7>)。这可以强制该模块与奇数编号的模块运行在同步模式下（不管其触发设置如何）。
4. 使用奇数编号的模块的 IC $i$  $x$  位 (IC $x$ CON1<6:5>) 设置所需的中断频率。
5. 使用奇数编号的模块的 ICTRIG 位 (IC $x$ CON2<7>) 配置触发或同步工作模式。

**注：** 对于同步工作模式，最后一个步骤是使能同步源。使能同步源之前，两个输入捕捉模块都保持复位状态。

6. 使用奇数编号的模块的 ICM $x$  位 (IC $x$ CON1<2:0>) 设置所需的捕捉模式。

使能了时基和同步 / 触发源之后，模块准备就绪，可以捕捉事件了。ICBNE 位 (IC $x$ CON1<3>) 置 1 时，FIFO 中至少有一个捕捉值。从 FIFO 读取输入捕捉值，直到 ICBNE 清零。

对于 32 位操作，读取 IC $x$ BUF 和 IC $y$ BUF 以得到整个 32 位定时器值 (IC $x$ BUF 用于低 16 位，IC $y$ BUF 用于高 16 位)。当奇数编号的模块的 ICBNE 位 (IC $x$ CON1<3>) 置 1 时，FIFO 缓冲区中至少有一个捕捉值。继续读缓冲寄存器直到 ICBNE 清零（由硬件自动执行）。

# PIC24FJ128GB204 系列

**寄存器 14-1: 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, HSC	R-0, HSC	R/W-0	R/W-0	R/W-0
—	ICI1	ICI0	ICOV	ICBNE	ICM2 <sup>(1)</sup>	ICM1 <sup>(1)</sup>	ICM0 <sup>(1)</sup>
bit 7						bit 0	

<b>图注:</b>	HSC = 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 15-14 **未实现:** 读为 0

bit 13 **ICSIDL:** 空闲时输入捕捉 x 模块停止控制位  
 1 = 输入捕捉模块将在 CPU 空闲模式下停止  
 0 = 输入捕捉模块将在 CPU 空闲模式下继续工作

bit 12-10 **ICTSEL<2:0>:** 输入捕捉 x 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>:** 选择每次中断发生的捕捉次数的位  
 11 = 每四次捕捉事件中断一次  
 10 = 每三次捕捉事件中断一次  
 01 = 每两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 = 已发生输入捕捉溢出  
 0 = 未发生输入捕捉溢出

bit 3 **ICBNE:** 输入捕捉 x 缓冲区空状态位 (只读)  
 1 = 输入捕捉缓冲区不为空, 至少可以读取一个以上的捕捉值  
 0 = 输入捕捉缓冲区为空

bit 2-0 **ICM<2:0>:** 输入捕捉 x 模式选择位 <sup>(1)</sup>  
 111 = 中断模式: 只有当器件处于休眠或空闲模式时输入捕捉才可作为中断引脚 (只检测上升沿, 所有其他控制位都不适用)  
 110 = 未使用 (禁止模块)  
 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: 每个沿 (上升和下降) 捕捉一次; ICI<1:0> 位不控制此模式下中断的产生  
 000 = 输入捕捉模块关闭

**注 1:** ICx 输入也必须配置给可用的 RPN/RPIIn 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。

# PIC24FJ128GB204 系列

寄存器 14-2: 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

<b>图注:</b>	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
	U = 未实现位, 读为 0
	0 = 清零
	x = 未知

- bit 15-9 **未实现:** 读为 0
- bit 8 **IC32:** 级联两个输入捕捉模块使能位 (32 位操作)
  - 1 = ICx 和 ICy 以级联的形式作为 32 位模块使用 (两个模块中的此位都必须置 1)
  - 0 = ICx 作为 16 位模块独立工作
- bit 7 **ICTRIG:** 输入捕捉 x 同步 / 触发选择位
  - 1 = 用 SYNCSELx 位指定的源触发 ICx
  - 0 = 使 ICx 与 SYNCSELx 位指定的源同步
- bit 6 **TRIGSTAT:** 定时器触发状态位
  - 1 = 定时器源被触发且正在运行 (该位可用硬件置 1, 也可用软件置 1)
  - 0 = 定时器源未被触发, 保持清零状态
- bit 5 **未实现:** 读为 0

- 注**
- 1: 这些输入仅用作触发源, 不能用作同步源。
  - 2: 请勿通过选择该模式将输入捕捉模块作为自己的触发源使用。

## 寄存器 14-2: ICxCON2: 输入捕捉 x 控制寄存器 2 (续)

bit 4-0 **SYNCSEL<4:0>**: 同步 / 触发源选择位

1111x = 保留  
11101 = 保留  
11100 = CTMU<sup>(1)</sup>  
11011 = A/D<sup>(1)</sup>  
11010 = 比较器 3<sup>(1)</sup>  
11001 = 比较器 2<sup>(1)</sup>  
11000 = 比较器 1<sup>(1)</sup>  
10111 = 保留  
10110 = 保留  
10101 = 输入捕捉 6<sup>(2)</sup>  
10100 = 输入捕捉 5<sup>(2)</sup>  
10011 = 输入捕捉 4<sup>(2)</sup>  
10010 = 输入捕捉 3<sup>(2)</sup>  
10001 = 输入捕捉 2<sup>(2)</sup>  
10000 = 输入捕捉 1<sup>(2)</sup>  
01111 = Timer5  
01110 = Timer4  
01101 = Timer3  
01100 = Timer2  
01011 = Timer1  
01010 = 保留  
01001 = 保留  
01000 = 保留  
00111 = 保留  
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:** 这些输入仅用作触发源，不能用作同步源。  
**注 2:** 请勿通过选择该模式将输入捕捉模块作为自己的触发源使用。

# PIC24FJ128GB204 系列

---

注:

## 15.0 带专用定时器的输出比较

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》中的“带专用定时器的输出比较”（DS70005159）。本数据手册中的信息取代了FRM中的信息。

PIC24FJ128GB204系列中的器件采用了6个独立的输出比较模块。每个模块都为发生器件内部事件时产生连续脉冲信号提供了各种配置和工作选项，还可产生脉宽调制（Pulse-Width Modulated, PWM）波形以驱动功率应用。

输出比较模块的主要特性包括：

- 通过级联两个相邻的模块，可采用硬件方式配置为所有的32位工作模式
- 输出比较操作有同步和触发两种模式，可使用最多31个用户可选的触发/同步源
- 两个单独的周期寄存器（一个主寄存器OCxR和一个辅助寄存器OCxRS），可更加灵活地产生不同宽度的脉冲
- 可配置为在发生输出事件时产生单脉冲信号或连续脉冲信号，或产生连续的PWM波形
- 每个模块最多可提供6种时钟源来驱动独立的内部16位计数器

### 15.1 一般工作模式

#### 15.1.1 同步和触发模式

输出比较模块工作在自由运行模式下时，内部16位计数器OCxTMR可连续向上计数，并在每次溢出时从0xFFFF折回0x0000。它的周期与所选的外部时钟源同步。当内部计数器与其中一个周期寄存器发生匹配时，产生一个比较或PWM事件。

在同步模式下，一旦使能了所选的时钟源，模块就开始执行比较或PWM操作。只要所选的同步源发生事件，模块的内部计数器都会复位。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后，才允许该计数器运行。

默认情况下或SYNCSELx位（OCxCON2<4:0>）设置为00000时，选择自由运行模式。当SYNCSELx位设置为00000之外的其他值时，将选择同步或触发模式。OCTRIG位（OCxCON2<7>）用于选择同步或触发模式；置1该位表示选择触发工作模式。在两种模式下，SYNCSELx位决定同步/触发源。

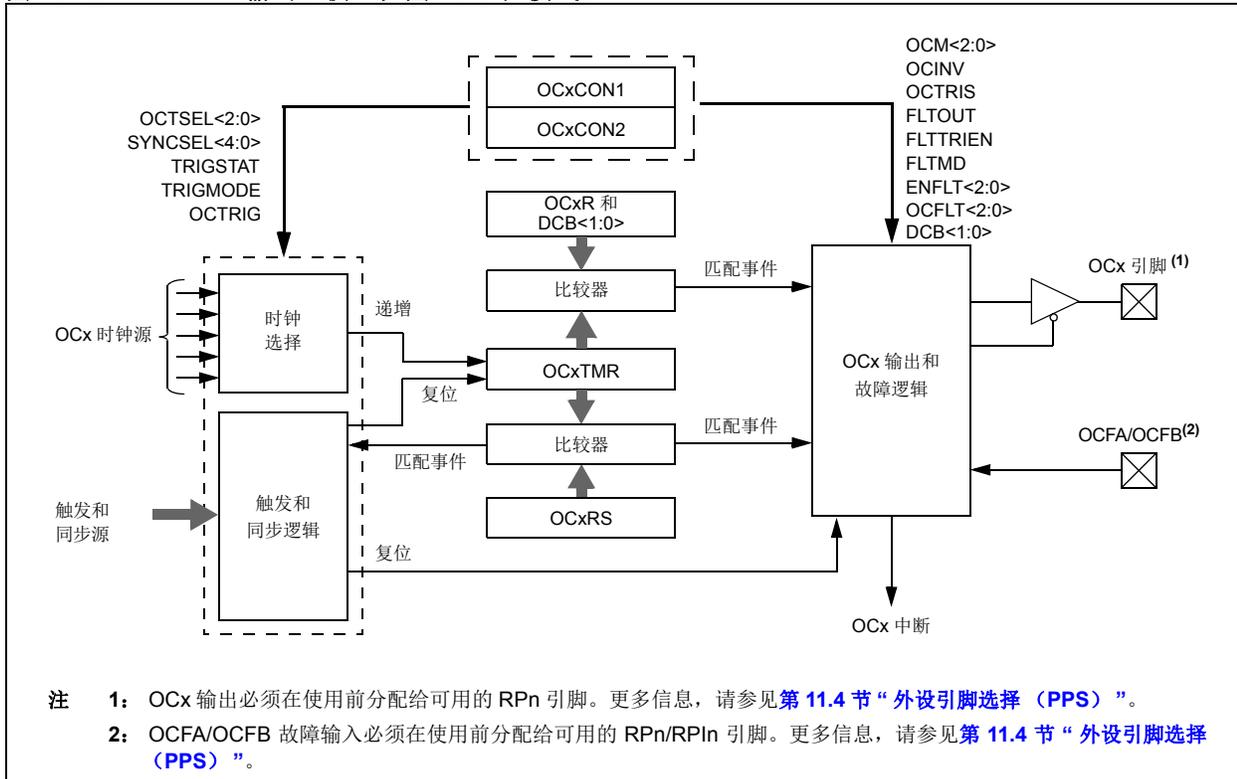
#### 15.1.2 级联（32位）模式

默认情况下，每个模块都可以使用自己的16位定时器和占空比寄存器独立运行。要提高分辨率，可将相邻的偶数编号的模块和奇数编号的模块配置为一个32位模块。（例如，模块1和2配对，模块3和4配对等）。奇数编号的输出比较模块（OCx）提供32位寄存器对中的低16位，而偶数编号的输出比较模块（OCy）提供高16位。OCx寄存器的进位操作会导致其相应的OCy寄存器递增。

通过将两个模块的OC32位（OCxCON2<8>）置1使用硬件来配置级联操作。欲知级联的更多信息，请参见《dsPIC33/PIC24系列参考手册》中的“带专用定时器的输出比较”（DS70005159）。

# PIC24FJ128GB204 系列

图 15-1: 输出比较 x 框图 (16 位模式)



## 15.2 比较操作

在比较模式下 (图 15-1), 输出比较模块可配置为产生单脉冲信号或产生连续脉冲信号。它还可在发生每个定时器事件时重复翻转输出引脚。

要将模块设置为比较操作:

1. 将 OCx 输出配置为其中一个可用的外设引脚选择引脚。
2. 计算 OCxR 和 (用于双比较模式) OCxRS 占空比寄存器所需的值:
  - a) 确定指令时钟周期。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. 将上升沿的值写入 OCxR, 下降沿的值写入 OCxRS。
4. 将定时器周期寄存器 PRy 的值设置为大于或等于 OCxRS 中的值。
5. 设置 OCM<2:0> 位以选择合适的比较操作 (0xx)。
6. 如需触发工作模式, 置 1 OCTRIG 以启动触发模式。置 1 或清零 TRIGMODE 可配置触发工作模式, 置 1 或清零 TRIGSTAT 可选择硬件触发或软件触发。如需同步模式, 清零 OCTRIG。
7. 设置 SYNCSEL<4:0> 位可配置触发或同步源。如需定时器在自由运行模式下工作, 请将 SYNCSELx 位设置为 00000 (无同步/触发源)。
8. 使用 OCTSEL<2:0> 位选择时基源。如有必要, 将使能比较时基进行计数的选定定时器的 TON 位置 1。使能了时基后, 将立即启动同步工作模式; 发生触发源事件后, 启动触发工作模式。

对于 32 位级联操作，还包含以下必需步骤：

1. 将两个寄存器的 OC32 位（即 OCyCON2<8> 和 OCxCON2<8>）置 1。首先使能偶数编号的模块，以确保模块一起启动。
2. 清零偶数编号的模块的 OCTRIG 位（OCyCON2<7>），使其在同步模式下运行。
3. 为 OCy 配置所需的输出和故障设置。
4. 通过清零 OCTRIS 位强制 OCx 的输出引脚为输出状态。
5. 如需触发工作模式，请使用 OCTRIG（OCxCON2<7>）、TRIGMODE（OCxCON1<3>）和 SYNCSELx（OCxCON2<4:0>）位配置触发选项。
6. 首先为 OCy 配置所需的比较或 PWM 工作模式（OCM<2:0>），然后再为 OCx 配置。

根据选择的输出模式，模块保持 OCx 引脚处于默认状态并在 OCxR 中的值与定时器中的值匹配时强制翻转该引脚的状态。在双比较模式下，当定时器中的值与 OCxRS 中的值匹配时，OCx 被强制返回默认状态。在单比较模式下发生定时器与 OCxR 的匹配、以及在双比较模式下每次定时器与 OCxRS 发生匹配后，OCxIF 中断标志都会置 1。

单脉冲事件仅发生一次，但是可通过简单地重写 OCxCON1 寄存器的值使其重复发生。连续脉冲事件在终止之前可无限次地发生。

## 15.3 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 操作都是双缓冲的（缓冲寄存器是模块内部的寄存器，没有映射到 SFR 空间）。

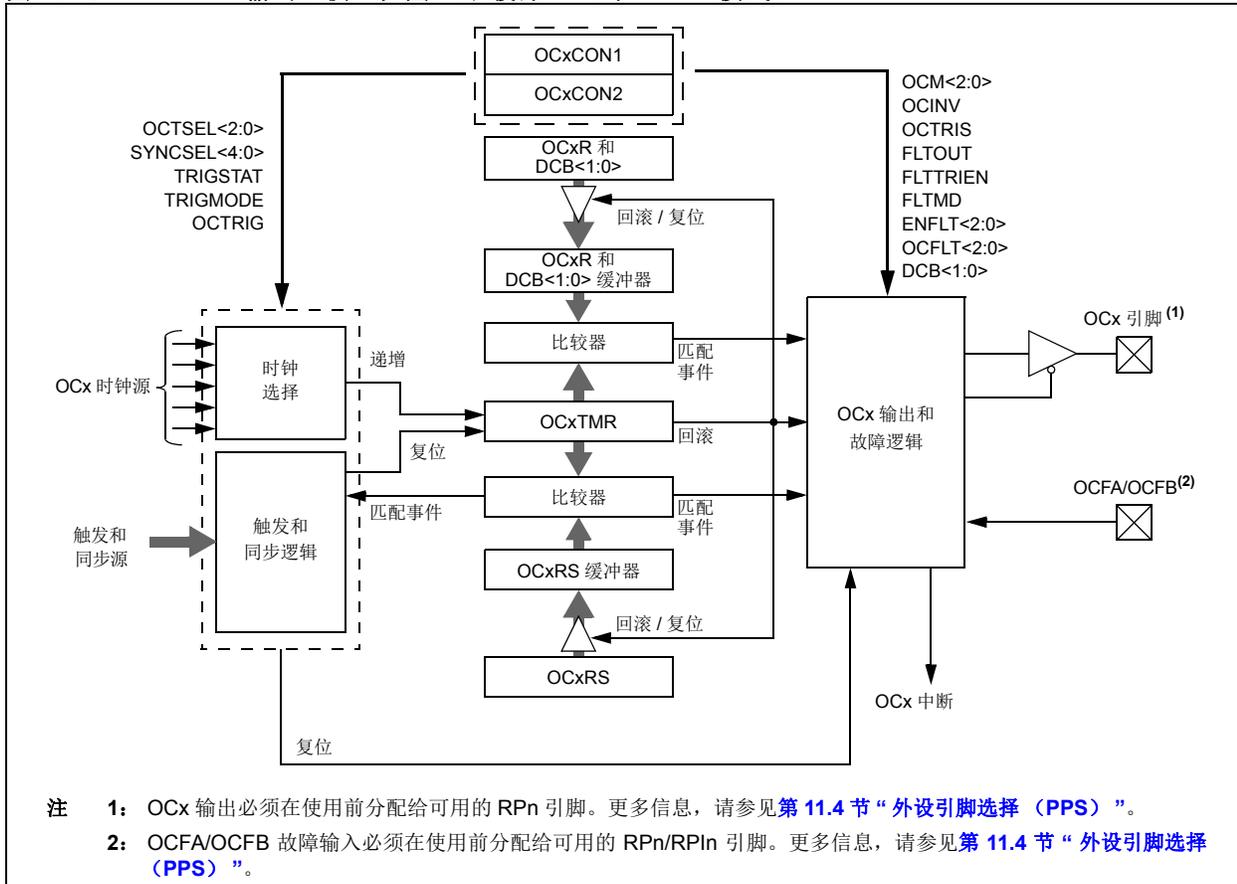
将输出比较模块配置为 PWM 工作模式：

1. 将 OCx 输出配置为其中一个可用的外设引脚选择引脚。
2. 计算所需的占空比并将计算结果装入 OCxR 寄存器。
3. 计算所需的周期并将计算结果装入 OCxRS 寄存器。
4. 选择当前的 OCx 作为同步源，方法是把 0x1F 写入 SYNCSEL<4:0> 位（OCxCON2<4:0>），把 0 写入 OCTRIG 位（OCxCON2<7>）。
5. 通过写 OCTSEL<2:0> 位（OCxCON1<12:10>）选择时钟源。
6. 如果需要的话，允许定时器和输出比较模块中断。使用 PWM 故障引脚时需要输出比较中断。
7. 通过 OCM<2:0>（OCxCON1<2:0>）位选择所需的 PWM 模式。
8. 通过使用 ENFLT<2:0> 位可以使能适当的故障输入，如寄存器 15-1 中所述。
9. 如果定时器被选为时钟源，请设置选定定时器预分频值。选定定时器的预分频器输出将用作 OCx 定时器的时钟输入，而不是选定定时器的输出。

**注：** 本外设包含的输入和输出功能可能需要通过外设引脚选择功能进行配置。更多信息，请参见第 11.4 节“外设引脚选择（PPS）”。

# PIC24FJ128GB204 系列

图 15-2: 输出比较 x 框图 (双缓冲, 16 位 PWM 模式)



## 15.3.1 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 15-1 来计算 PWM 周期。

公式 15-1: 计算 PWM 周期<sup>(1)</sup>

$$PWM \text{ 周期} = [(PRy) + 1] \cdot Tcy \cdot (\text{定时器预分频值})$$

其中:

$$PWM \text{ 频率} = 1/[PWM \text{ 周期}]$$

注 1: 基于  $Tcy = TOSC \cdot 2$ ; 禁止打盹模式和 PLL。

注: 若 PRy 的值为 N, 则会使 PWM 周期为 N + 1 个时基计数周期。例如: 写入 PRy 寄存器的值为 7 将产生由 8 个时基周期组成的 PWM 周期。

## 15.3.2 PWM 占空比

PWM 占空比是通过写 OCxRS 和 OCxR 寄存器指定的。在任何时间都可以写入 OCxRS 和 OCxR 寄存器，但是在 PRy 和 TMRy 发生匹配（即周期完成）前不会锁存占空比值。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。

以下是 PWM 占空比的部分重要边界参数：

- 如果 OCxR、OCxRS 和 PRy 装入 0000h，那么 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%）。

欲知 PWM 模式时序的详细信息，请参见例 15-1。表 15-1 和表 15-2 给出了当器件分别工作在 4 MIPS 和 10 MIPS 时 PWM 频率和分辨率的示例。

**公式 15-2: 计算最大 PWM 分辨率<sup>(1)</sup>**

$$\text{最大 PWM 分辨率 (位)} = \frac{\log_{10} \left( \frac{F_{CY}}{F_{PWM} \cdot (\text{定时器预分频值})} \right)}{\log_{10} 2} \text{ 位}$$

**注 1:** 基于  $F_{CY} = F_{OSC}/2$ ；禁止打盹模式和 PLL。

**例 15-1: 计算 PWM 周期和占空比<sup>(1)</sup>**

1. 在所需的 PWM 频率为 52.08 kHz、 $F_{OSC} = 8 \text{ MHz}$  并带 PLL（32 MHz 器件时钟速率）且 Timer2 预分频值设置为 1:1 的条件下计算定时器周期寄存器的值。

$$T_{CY} = 2 * T_{OSC} = 62.5 \text{ ns}$$

$$\text{PWM 周期} = 1/\text{PWM 频率} = 1/52.08 \text{ kHz} = 19.2 \text{ } \mu\text{s}$$

$$\text{PWM 周期} = (\text{PR2} + 1) \cdot T_{CY} \cdot (\text{Timer2 预分频值})$$

$$19.2 \text{ } \mu\text{s} = (\text{PR2} + 1) \cdot 62.5 \text{ ns} \cdot 1$$

$$\text{PR2} = 306$$

2. 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：

$$\text{PWM 分辨率} = \log_{10}(F_{CY}/F_{PWM})/\log_{10}2 \text{ 位}$$

$$= (\log_{10}(16 \text{ MHz}/52.08 \text{ kHz})/\log_{10}2) \text{ 位}$$

$$= 8.3 \text{ 位}$$

**注 1:** 基于  $T_{CY} = 2 * T_{OSC}$ ，禁止打盹模式和 PLL。

**表 15-1: 器件工作在 4 MIPS ( $F_{CY} = 4 \text{ MHz}$ ) 时的 PWM 频率和分辨率示例<sup>(1)</sup>**

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  $F_{CY} = F_{OSC}/2$ ；禁止打盹模式和 PLL。

**表 15-2: 器件工作在 16 MIPS ( $F_{CY} = 16 \text{ MHz}$ ) 时的 PWM 频率和分辨率示例<sup>(1)</sup>**

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  $F_{CY} = F_{OSC}/2$ ；禁止打盹模式和 PLL。

# PIC24FJ128GB204 系列

寄存器 15-1: 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	R/W-0	R/W-0
—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2 <sup>(2)</sup>	ENFLT1 <sup>(2)</sup>
bit 15						bit 8	
R/W-0	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0	R/W-0	R/W-0	R/W-0
ENFLT0 <sup>(2)</sup>	OCFLT2 <sup>(2,3)</sup>	OCFLT1 <sup>(2,4)</sup>	OCFLT0 <sup>(2,4)</sup>	TRIGMODE	OCM2 <sup>(1)</sup>	OCM1 <sup>(1)</sup>	OCM0 <sup>(1)</sup>
bit 7						bit 0	

**图注:** HSC = 可由硬件置 1/ 清零的位  
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 15-14 **未实现:** 读为 0
- bit 13 **OCSIDL:** 在空闲模式下输出比较 x 停止控制位  
1 = 输出比较 x 在 CPU 空闲模式下停止  
0 = 输出比较 x 将在 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 = 外设时钟 (Fcy)  
110 = 保留  
101 = 保留  
100 = Timer1 时钟 (仅支持同步时钟)  
011 = Timer5 时钟  
010 = Timer4 时钟  
001 = Timer3 时钟  
000 = Timer2 时钟
- bit 9 **ENFLT2:** 故障输入 2 使能位<sup>(2)</sup>  
1 = 使能故障 2 (比较器 1/2/3 输出)<sup>(3)</sup>  
0 = 禁止故障 2
- bit 8 **ENFLT1:** 故障输入 1 使能位<sup>(2)</sup>  
1 = 使能故障 1 (OCFB 引脚)<sup>(4)</sup>  
0 = 禁止故障 1
- bit 7 **ENFLT0:** 故障输入 0 使能位<sup>(2)</sup>  
1 = 使能故障 0 (OCFA 引脚)<sup>(4)</sup>  
0 = 禁止故障 0
- bit 6 **OCFLT2:** 输出比较 x PWM 故障 2 (比较器 1/2/3) 条件状态位<sup>(2,3)</sup>  
1 = 已发生 PWM 故障 2  
0 = 未发生 PWM 故障 2
- bit 5 **OCFLT1:** 输出比较 x PWM 故障 1 (OCFB 引脚) 条件状态位<sup>(2,4)</sup>  
1 = 已发生 PWM 故障 1  
0 = 未发生 PWM 故障 1

**注** 1: OCx 输出也必须配置给可用的 RPN 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。  
2: 当 OCM<2:0> = 111 或 110 时, 故障输入使能位和故障状态位有效。  
3: 比较器 1 输出控制 OC1-OC2 通道; 比较器 2 输出控制 OC3-OC4 通道; 比较器 3 输出控制 OC5-OC6 通道。  
4: OCFA/OCFB 故障输入也必须配置给可用的 RPN/RPN 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。

## 寄存器 15-1: OCxCON1: 输出比较 x 控制寄存器 1 (续)

bit 4	<b>OCFLT0:</b> 输出比较 x PWM 故障 0 (OCFA 引脚) 条件状态位 <sup>(2,4)</sup> 1 = 已发生 PWM 故障 0 0 = 未发生 PWM 故障 0
bit 3	<b>TRIGMODE:</b> 触发状态模式选择位 1 = 当 OCxRS = OCxTMR 时清零 TRIGSTAT (OCxCON2<6>), 或由软件清零 0 = TRIGSTAT 只能由软件清零
bit 2-0	<b>OCM&lt;2:0&gt;:</b> 输出比较 x 模式选择位 <sup>(1)</sup> 111 = OCx 处于中心对齐 PWM 模式 <sup>(2)</sup> 110 = OCx 处于边沿对齐 PWM 模式 <sup>(2)</sup> 101 = 双比较连续脉冲模式: 初始化 OCx 引脚为低电平, 与 OCxR 和 OCxRS 交替匹配时连续翻转 OCx 的状态 100 = 双比较单脉冲模式: 初始化 OCx 引脚为低电平; 在一个周期内与 OCxR 与 OCxRS 匹配时翻转 OCx 的状态 011 = 单比较连续脉冲模式: 比较事件连续翻转 OCx 引脚 010 = 单比较单脉冲模式: 初始化 OCx 引脚为高电平; 比较事件强制 OCx 引脚为低电平 010 = 单比较单脉冲模式: 初始化 OCx 引脚为低电平; 比较事件强制 OCx 引脚为高电平 000 = 禁止输出比较通道

- 注
- 1: OCx 输出也必须配置给可用的 RPN 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。
  - 2: 当 OCM<2:0> = 111 或 110 时, 故障输入使能位和故障状态位有效。
  - 3: 比较器 1 输出控制 OC1-OC2 通道; 比较器 2 输出控制 OC3-OC4 通道; 比较器 3 输出控制 OC5-OC6 通道。
  - 4: OCFA/OCFB 故障输入也必须配置给可用的 RPN/RPN 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。

# PIC24FJ128GB204 系列

寄存器 15-2: 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	R/W-0	R/W-0	R/W-0
FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1 <sup>(3)</sup>	DCB0 <sup>(3)</sup>	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	

<b>图注:</b>	HS = 可由硬件置 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	x = 未知

- bit 15      **FLTMD:** 故障模式选择位  
1 = 故障模式保持, 直到消除故障源并且相应的 OCFLT0 位由软件清零  
0 = 故障模式保持, 直到消除故障源并且启动了新的 PWM 周期
- bit 14      **FLTOUT:** 故障输出位  
1 = 发生故障时 PWM 输出被驱动为高电平  
0 = 发生故障时 PWM 输出被驱动为低电平
- bit 13      **FLTTRIEN:** 故障输出状态选择位  
1 = 发生故障条件时引脚被强制为输出状态  
0 = 引脚 I/O 状态不受故障影响
- bit 12      **OCINV:** 输出比较 x 反相位  
1 = OCx 输出反相  
0 = OCx 输出不反相
- bit 11      **未实现:** 读为 0
- bit 10-9    **DCB<1:0>:** PWM 占空比最低有效位 <sup>(3)</sup>  
11 = OCx 下降沿延迟 ¾ 指令周期  
10 = OCx 下降沿延迟 ½ 指令周期  
01 = OCx 下降沿延迟 ¼ 指令周期  
00 = OCx 下降沿在指令周期一开始就出现
- bit 8        **OC32:** 级联两个输出比较模块使能位 (32 位操作)  
1 = 使能级联模块操作  
0 = 禁止级联模块操作
- bit 7        **OCTRIG:** 输出比较 x 触发 / 同步选择位  
1 = 用 SYNCSELx 位指定的源触发 OCx  
0 = OCx 与 SYNCSELx 位指定的源同步
- bit 6        **TRIGSTAT:** 定时器触发状态位  
1 = 定时器源已被触发并运行  
0 = 定时器源未被触发, 保持清零状态
- bit 5        **OCTRIS:** 输出比较 x 输出引脚方向选择位  
1 = OCx 引脚是三态引脚  
0 = OCx 引脚上连接了输出比较外设 x

注 1: 不要通过以下方式将 OCx 模块作为自己的触发源: 选择这种模式或其他同等 SYNCSELx 设置。  
 2: 这些输入仅用作触发源, 不能用作同步源。  
 3: DCB<1:0> 位只有在 PWM 模式 (OCM<2:0> (OCxCON1<2:0>) = 111、110) 下才是双缓冲位。

## 寄存器 15-2: 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 = 此输出比较模块 <sup>(1)</sup>  
11110 = OCTRIG1 外部输入  
11101 = OCTRIG2 外部输入  
11100 = CTMU <sup>(2)</sup>  
11011 = A/D <sup>(2)</sup>  
11010 = 比较器 3 <sup>(2)</sup>  
11001 = 比较器 2 <sup>(2)</sup>  
11000 = 比较器 1 <sup>(2)</sup>  
10111 = 保留  
10110 = 保留  
10101 = 输入捕捉 6 <sup>(2)</sup>  
10100 = 输入捕捉 5 <sup>(2)</sup>  
10011 = 输入捕捉 4 <sup>(2)</sup>  
10010 = 输入捕捉 3 <sup>(2)</sup>  
10001 = 输入捕捉 2 <sup>(2)</sup>  
10000 = 输入捕捉 1 <sup>(2)</sup>  
01111 = Timer5  
01110 = Timer4  
01101 = Timer3  
01100 = Timer2  
01011 = Timer1  
01010 = 保留  
01001 = 保留  
01000 = 保留  
00111 = 保留  
00110 = 输出比较 6 <sup>(1)</sup>  
00101 = 输出比较 5 <sup>(1)</sup>  
00100 = 输出比较 4 <sup>(1)</sup>  
00011 = 输出比较 3 <sup>(1)</sup>  
00010 = 输出比较 2 <sup>(1)</sup>  
00001 = 输出比较 1 <sup>(1)</sup>  
00000 = 不与任何其他模块同步

- 注**
- 1: 不要通过以下方式将 OCx 模块作为自己的触发源: 选择这种模式或其他同等 SYNCSELx 设置。
  - 2: 这些输入仅用作触发源, 不能用作同步源。
  - 3: DCB<1:0> 位只有在 PWM 模式 (OCM<2:0> (OCxCON1<2:0>) = 111、110) 下才是双缓冲位。

# PIC24FJ128GB204 系列

---

注:

## 16.0 串行外设接口 (SPI)

**注:** 本数据手册总结了 PIC24FJ128GB204 系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息, 请参见 Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 上提供的《dsPIC33/PIC24 系列参考手册》中的“支持音频编解码器的串行外设接口 (SPI)” (DS70005136)。

串行外设接口 (Serial Peripheral Interface, SPI) 模块是用于与其他外设或者单片机进行通信的同步串行接口。这些外设可以是串行 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。PIC24FJ128GB204 系列的所有器件都包含三个 SPI 模块。

该模块可在两种缓冲模式下工作。在标准缓冲模式下, 通过单个串行缓冲器移动数据。在增强型缓冲模式下, 通过 FIFO 缓冲区移动数据。FIFO 深度取决于配置模式。

可发送和接收 2 到 32 位的可变长度数据。

**注:** 无论是在标准还是增强型缓冲模式下, 都不要对 SPIxBUF 寄存器执行读 - 修改 - 写操作 (如位操作指令)。

工作在主模式或从模式下时, 该模块还支持基本的帧 SPI 协议。共支持四种帧 SPI 配置。

该模块还支持音频模式。提供了四种不同的音频模式。

- I<sup>2</sup>S 模式
- 左对齐
- 右对齐
- PCM/DSP

在每个模式下, 串行时钟都自由运行且始终传输音频数据。

如果在两个器件之间进行音频协议数据传输, 那么通常一个器件为主器件, 而另一个器件为从器件。不过, 也可在两个从器件之间传输音频数据。由于音频协议需要自由运行的时钟, 所以主器件可以是第三方控制器。在任一情况下, 主器件都生成两个自由运行的时钟: SCKx 和 LRC (左、右通道时钟 /SSx/FSYNC)。

SPI 串行接口由以下四个引脚组成:

- 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 模块可以被配置为使用 2 个、3 个或 4 个引脚工作。在 3 引脚模式下, 不使用 SSx。在 2 引脚模式下, 不使用 SDOx 和 SSx。

SPI 模块可产生 3 个中断, 反映数据通信期间发生的事件。可产生以下类型的中断:

1. 接收中断由 SPIxRXIF 指示。在以下情况下发生该事件:
  - 接收水印中断
  - SPIROV = 1
  - SPIRBF = 1
  - SPIRBE = 1

前提是已在 SPIxIMSKL/H 中使能相应的掩码位。
2. 发送中断由 SPIxTXIF 指示。在以下情况下发生该事件:
  - 发送水印中断
  - SPITUR = 1
  - SPITBF = 1
  - SPITBE = 1

前提是已在 SPIxIMSKL/H 中使能相应的掩码位。
3. 通用中断由 SPIxIF 指示。在以下情况下发生该事件:
  - FRMERR = 1
  - SPIBUSY = 1
  - SRMT = 1

前提是已在 SPIxIMSKL/H 中使能相应的掩码位。

图 16-1 和图 16-2 所示为标准模式和增强型模式下该模块的框图。

**注:** 在本章中, SPI 模块统称为 SPIx, 或分别称为 SPI1、SPI2 和 SPI3。特殊功能寄存器也使用类似的符号表示。例如, SPIxCON1L 和 SPIxCON1H 表示这 3 个 SPI 模块中任何一个的控制寄存器。

# PIC24FJ128GB204 系列

## 16.1 标准主模式

要将 SPIx 模块设置为工作在标准主模式下，请遵循以下步骤：

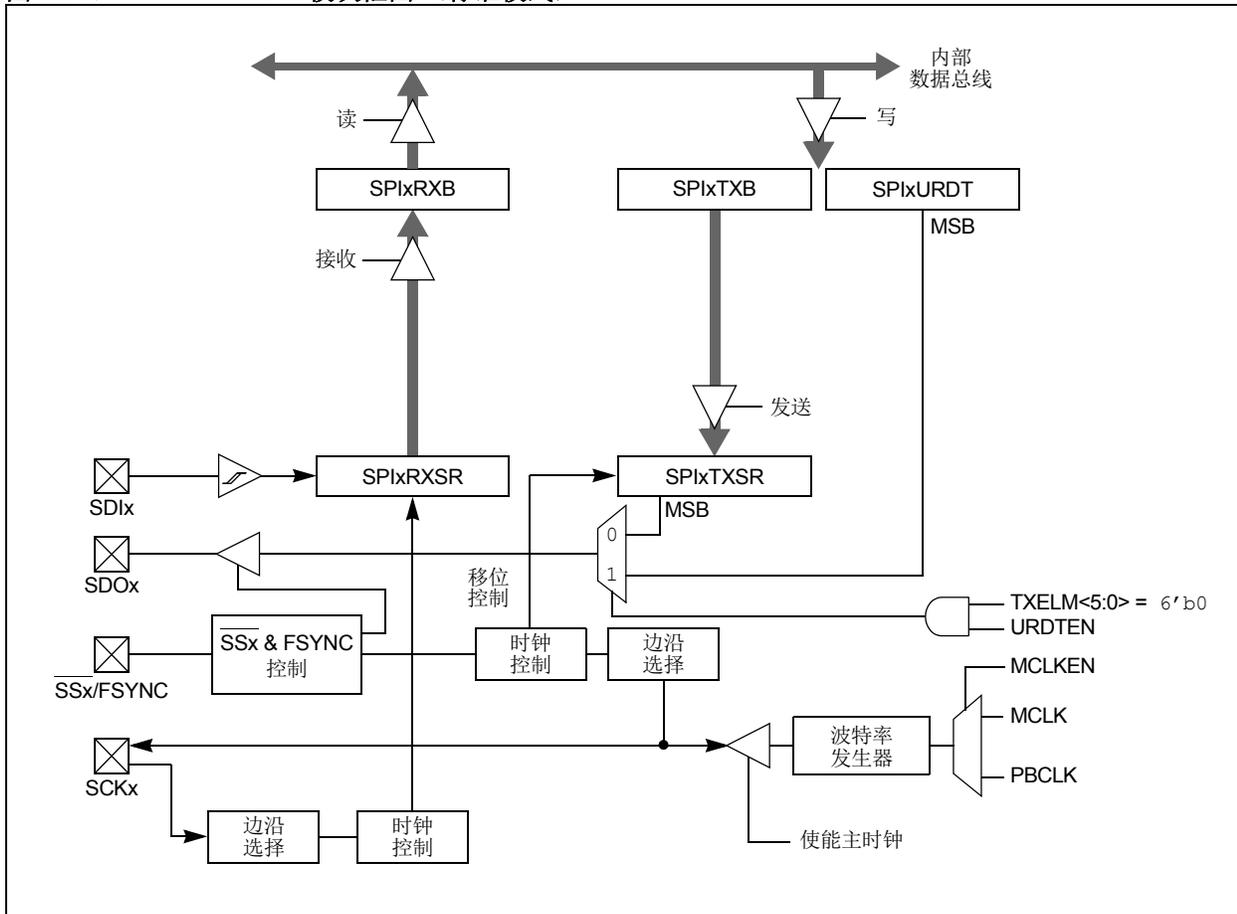
1. 如果使用中断：
  - a) 将相应 IFSx 寄存器中的中断标志位清零。
  - b) 将相应 IECx 寄存器中的中断允许位置 1。
  - c) 通过写相应 IPCx 寄存器中的 SPIxIP<2:0> 位来设置中断优先级。
2. 将所需设置写入 SPIxCON1L 和 SPIxCON1H 寄存器，同时 MSTEN 位 (SPIxCON1L<5>) = 1。
3. 将 SPIROV 位 (SPIxSTATL<6>) 清零。
4. 通过将 SPIEN 位 (SPIxCON1L<15>) 置 1 使能 SPIx 操作。
5. 将待发送数据写入 SPIxBUFL 和 SPIxBUFH 寄存器。数据一写入 SPIxBUFL 和 SPIxBUFH 寄存器，发送（和接收）就会立即开始。

## 16.2 标准从模式

要将 SPIx 模块设置为工作在标准从模式下，请遵循以下步骤：

1. 将 SPIxBUF 寄存器清零。
2. 如果使用中断：
  - a) 将 SPIxBUFL 和 SPIxBUFH 寄存器清零。
  - b) 将相应 IECx 寄存器中的中断允许位置 1。
  - c) 通过写相应 IPCx 寄存器中的 SPIxIP<2:0> 位来设置中断优先级。
3. 将所需设置写入 SPIxCON1L、SPIxCON1H 和 SPIxCON2L 寄存器，同时 MSTEN 位 (SPIxCON1L<5>) = 0。
4. 将 SMP 位清零。
5. 如果将 CKE 位 (SPIxCON1L<8>) 置 1，则 SSEN 位 (SPIxCON1L<7>) 也必须置 1 以使能 SSx 引脚。
6. 将 SPIROV 位 (SPIxSTATL<6>) 清零。
7. 通过将 SPIEN 位 (SPIxCON1L<15>) 置 1 使能 SPIx 操作。

图 16-1: SPIx 模块框图 (标准模式)



## 16.3 增强型从模式

要将 SPIx 模块设置为工作在增强型缓冲主模式下，请遵循以下步骤：

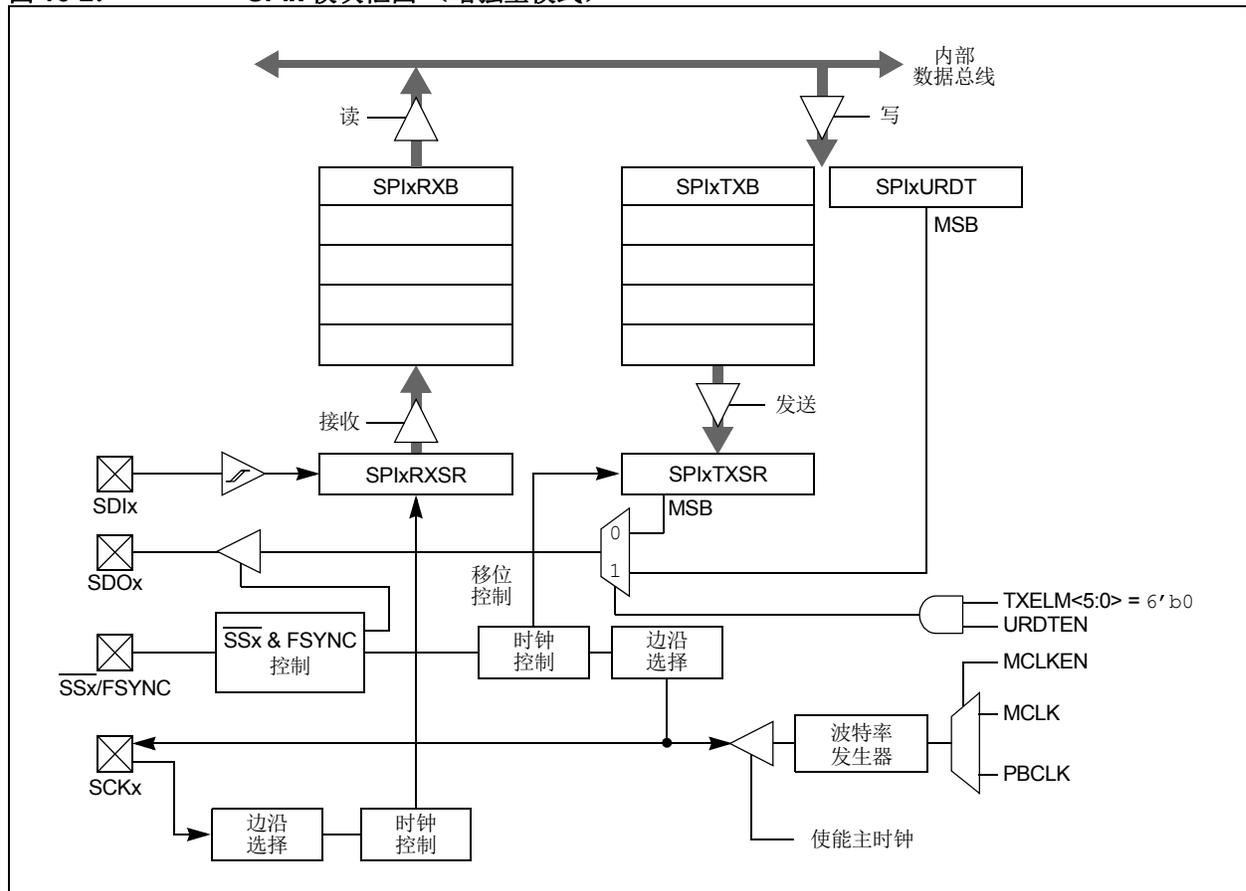
1. 如果使用中断：
  - a) 将相应 IFSx 寄存器中的中断标志位清零。
  - b) 将相应 IECx 寄存器中的中断允许位置 1。
  - c) 写相应 IPCx 寄存器中的 SPIxIP<2:0> 位。
2. 将所需设置写入 SPIxCON1L、SPIxCON1H 和 SPIxCON2L 寄存器，同时 MSTEN (SPIxCON1L<5>) = 1。
3. 将 SPIROV 位 (SPIxSTATL<6>) 清零。
4. 通过将 ENHBUF 位 (SPIxCON1L<0>) 置 1 选择增强型缓冲模式。
5. 通过将 SPIEN 位 (SPIxCON1L<15>) 置 1 使能 SPIx 操作。
6. 将待发送数据写入 SPIxBUFL 和 SPIxBUFH 寄存器。数据一写入 SPIxBUFL 和 SPIxBUFH 寄存器，发送（和接收）就会立即开始。

## 16.4 增强型主模式

要将 SPIx 模块设置为工作在增强型缓冲从模式下，请遵循以下步骤：

1. 将 SPIxBUFL 和 SPIxBUFH 寄存器清零。
2. 如果使用中断：
  - a) 将相应 IFSx 寄存器中的中断标志位清零。
  - b) 将相应 IECx 寄存器中的中断允许位置 1。
  - c) 通过写相应 IPCx 寄存器中的 SPIxIP<2:0> 位来设置中断优先级。
3. 将所需设置写入 SPIxCON1L、SPIxCON1H 和 SPIxCON2L 寄存器，同时 MSTEN 位 (SPIxCON1L<5>) = 0。
4. 将 SMP 位清零。
5. 如果 CKE 位置 1，则 SSEN 位也必须置 1 以使能 SSx 引脚。
6. 将 SPIROV 位 (SPIxSTATL<6>) 清零。
7. 通过将 ENHBUF 位 (SPIxCON1L<0>) 置 1 选择增强型缓冲模式。
8. 通过将 SPIEN 位 (SPIxCON1L<15>) 置 1 使能 SPIx 操作。

图 16-2: SPIx 模块框图（增强型模式）



# PIC24FJ128GB204 系列

---

## 16.5 音频模式

要将 SPIx 模块设置为工作在音频模式下，请遵循以下步骤：

1. 将 SPIxBUFL 和 SPIxBUFH 寄存器清零。
2. 如果使用中断：
  - a) 将相应 IFSx 寄存器中的中断标志位清零。
  - b) 将相应 IECx 寄存器中的中断允许位置 1。
  - c) 通过写相应 IPCx 寄存器中的 SPIxIP<2:0> 位来设置中断优先级。
3. 将所需设置写入 SPIxCON1L、SPIxCON1H 和 SPIxCON2L 寄存器，同时 AUDEN 位 (SPIxCON1H<15>) = 1。
4. 将 SPIROV 位 (SPIxSTATL<6>) 清零。
5. 通过将 SPIEN 位 (SPIxCON1L<15>) 置 1 使能 SPIx 操作。
6. 将待发送数据写入 SPIxBUFL 和 SPIxBUFH 寄存器。数据一写入 SPIxBUFL 和 SPIxBUFH 寄存器，发送（和接收）就会立即开始。

## 16.6 寄存器

SPI 模块由以下特殊功能寄存器（SFR）组成：

- SPIxCON1L、SPIxCON1H 和 SPIxCON2L: SPIx 控制器寄存器（寄存器 16-1 至寄存器 16-3）
- SPIxSTATL 和 SPIxSTATH: SPIx 状态寄存器（寄存器 16-4 和寄存器 16-5）
- SPIxBUFL 和 SPIxBUFH: SPIx 缓冲区寄存器
- SPIxBRGL: SPIx 波特率寄存器
- SPIxIMSKL 和 SPIxIMSKH: SPIx 中断屏蔽寄存器（寄存器 16-6 和寄存器 16-7）
- SPIxURDTL 和 SPIxURDTH: SPIx 下溢数据寄存器

# PIC24FJ128GB204 系列

寄存器 16-1: **SPIxCON1L: SPIx 控制寄存器 1 低位字**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPIEN	—	SPISIDL	DISSDO	MODE32 <sup>(1,4)</sup>	MODE16 <sup>(1,4)</sup>	SMP	CKE <sup>(1)</sup>
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 <sup>(2)</sup>	CKP	MSTEN	DISSDI	DISSCK	MCLKEN <sup>(3)</sup>	SPIFE	ENHBUF
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      **SPIEN:** SPIx 使能位  
 1 = 使能模块  
 0 = 关闭并复位模块, 禁止时钟, 禁止发生中断事件, 允许修改 SFR
- bit 14      **未实现:** 读为 0
- bit 13      **SPISIDL:** SPIx 空闲模式停止位  
 1 = 在 CPU 空闲模式下停止工作  
 0 = 在 CPU 空闲模式下继续工作
- bit 12      **DISSDO:** 禁止 SDOx 输出端口位  
 1 = 模块不使用 SDOx 引脚; 引脚由端口功能控制  
 0 = SDOx 引脚由模块控制
- bit 11-10   **MODE<32,16>:** 串行字长度位 <sup>(1,4)</sup>  
AUDEN = 0:
- |        |        |      |
|--------|--------|------|
| MODE32 | MODE16 | 通信   |
| 1      | x      | 32 位 |
| 0      | 1      | 16 位 |
| 0      | 0      | 8 位  |
- AUDEN = 1:
- |        |        |                                  |
|--------|--------|----------------------------------|
| MODE32 | MODE16 | 通信                               |
| 1      | 1      | 24 位数据, 32 位 FIFO, 32 位通道 /64 位帧 |
| 1      | 0      | 32 位数据, 32 位 FIFO, 32 位通道 /64 位帧 |
| 0      | 1      | 16 位数据, 16 位 FIFO, 32 位通道 /64 位帧 |
| 0      | 0      | 16 位数据, 16 位 FIFO, 16 位通道 /32 位帧 |
- bit 9      **SMP:** SPIx 数据输入采样阶段位  
主模式  
 1 = 在数据输出时间的结尾采样输入数据  
 0 = 在数据输出时间的中间采样输入数据  
从模式  
 始终在数据输出时间的中间采样输入数据, 无论 SMP 设置如何。
- bit 8      **CKE:** SPIx 时钟边沿选择位 <sup>(1)</sup>  
 1 = 在时钟由有效状态变为空闲状态时发送  
 0 = 在时钟由空闲状态变为有效状态时发送

- 注 1: 当 AUDEN = 1 时, 该模块按照 CKE = 0 的情况运行, 而不管 CKE 的实际值为何。  
 2: FRMEN = 1 时, 不使用 SSEN。  
 3: 仅当 SPIEN 位 = 0 时才能写 MCLKEN。  
 4: 该通道对于 DSP/PCM 模式没有意义, 原因是 LRC 遵循 FRMSYPW。

# PIC24FJ128GB204 系列

## 寄存器 16-1: SPIxCON1L: SPIx 控制寄存器 1 低位字 (续)

bit 7	<b>SSEN:</b> 从选择使能 (从模式) 位 <sup>(2)</sup> 1 = $\overline{SSx}$ 引脚由宏用在从模式下; $\overline{SSx}$ 引脚用作从选择输入 0 = $\overline{SSx}$ 引脚不被宏使用 ( $\overline{SSx}$ 引脚将由端口 I/O 控制)
bit 6	<b>CKP:</b> 时钟极性选择位 1 = 时钟信号空闲状态为高电平; 有效状态为低电平 0 = 时钟信号空闲状态为低电平; 有效状态为高电平
bit 5	<b>MSTEN:</b> 主模式使能位 1 = 主模式 0 = 从模式
bit 4	<b>DISSDI:</b> 禁止 SDIx 输入端口位 1 = 模块不使用 SDIx 引脚; 引脚由端口功能控制 0 = SDIx 引脚由模块控制
bit 3	<b>DISSCK:</b> 禁止 SCKx 输出端口位 1 = 模块不使用 SCKx 引脚; 引脚由端口功能控制 0 = SCKx 引脚由模块控制
bit 2	<b>MCLKEN:</b> 主时钟使能位 <sup>(3)</sup> 1 = BRG 使用 MCLK 0 = BRG 使用 PBCLK
bit 1	<b>SPIFE:</b> 帧同步脉冲边沿选择位 1 = 帧同步脉冲 (空闲到有效边沿) 与第一个位时钟同步 0 = 帧同步脉冲 (空闲到有效边沿) 比第一个位时钟超前
bit 0	<b>ENHBUF:</b> 增强型缓冲模式使能位 1 = 使能增强型缓冲模式 0 = 禁止增强型缓冲模式

- 注 1: 当 AUDEN = 1 时, 该模块按照 CKE = 0 的情况运行, 而不管 CKE 的实际值为何。
- 注 2: FRMEN = 1 时, 不使用 SSEN。
- 注 3: 仅当 SPIEN 位 = 0 时才能写 MCLKEN。
- 注 4: 该通道对于 DSP/PCM 模式没有意义, 原因是 LRC 遵循 FRMSYPW。

# PIC24FJ128GB204 系列

寄存器 16-2: SPIxCON1H: SPIx 控制寄存器 1 高位字

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AUDEN <sup>(1)</sup>	SPISGNEXT	IGNROV	IGNTUR	AUDMONO <sup>(2)</sup>	URDTEN <sup>(3)</sup>	AUDMOD1 <sup>(4)</sup>	AUDMOD0 <sup>(4)</sup>
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
FRMEN	FRMSYNC	FRMPOL	MSEN	FRMSYPW	FRMCNT2	FRMCNT1	FRMCNT0
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      **AUDEN:** 使能音频编解码器支持位 <sup>(1)</sup>  
 1 = 使能音频协议; MSTEN 控制 SCKx 和帧 (又称 LRC) 的方向, 且模块按照 FRMEN = 1、FRMSYNC = MSTEN、FRMCNT<2:0> = 001 和 SMP = 0 的情况运行, 而不管它们的实际值为何  
 0 = 禁止音频协议
- bit 14      **SPISGNEXT:** SPIx 符号扩展接收 FIFO 读数据使能位  
 1 = 接收 FIFO 中的数据进行符号扩展  
 0 = 接收 FIFO 中的数据不进行符号扩展
- bit 13      **IGNROV:** 忽略接收溢出位  
 1 = 接收溢出 (Receive Overflow, ROV) 不是关键错误; 在 ROV 期间, FIFO 中的数据不会被接收数据覆盖  
 0 = ROV 是关键错误, 会停止 SPI 操作
- bit 12      **IGNTUR:** 忽略发送数据不足位  
 1 = 发送数据不足 (Transmit Underrun, TUR) 不是关键错误, 在 SPIxTXB 不为空之前将一直发送 URDTEN 指示的数据  
 0 = TUR 是关键错误, 会停止 SPI 操作
- bit 11      **AUDMONO:** 发送音频数据格式位 <sup>(2)</sup>  
 1 = 音频数据为单声道 (例如, 每个数据字在左右声道同时发送)  
 0 = 音频数据为立体声
- bit 10      **URDTEN:** 发送数据不足使能位 <sup>(3)</sup>  
 1 = 遇到发送数据不足 (TUR) 条件时发送 SPIxURDTL/H 寄存器中的数据  
 0 = 遇到发送数据不足条件时发送上一次接收到的数据
- bit 9-8     **AUDMOD<1:0>:** 音频协议模式选择位 <sup>(4)</sup>  
 11 = PCM/DSP 模式  
 10 = 右对齐模式: 该模块按照 SPIFE = 1 的情况运行, 而不管 SPIFE 的实际值为何  
 01 = 左对齐模式: 该模块按照 SPIFE = 1 的情况运行, 而不管 SPIFE 的实际值为何  
 00 = I<sup>2</sup>S 模式: 该模块按照 SPIFE = 0 的情况运行, 而不管 SPIFE 的实际值为何
- bit 7        **FRMEN:** 帧 SPIx 支持位  
 1 = 使能帧 SPIx 支持 (SSx 引脚用作 FSYNC 输入 / 输出)  
 0 = 禁止帧 SPIx 支持

- 注**
- 1: 仅当 SPIEN 位 = 0 时才能写 AUDEN。
  - 2: 仅当 SPIEN 位 = 0 时才能写 AUDMONO, 且 AUDMONO 仅在 AUDEN = 1 时有效。
  - 3: URDTEN 仅在 IGNTUR = 1 时有效。
  - 4: 仅当 SPIEN 位 = 0 时才能写 AUDMOD<1:0>, 且 AUDMOD<1:0> 仅在 AUDEN = 1 时有效。当未工作在 PCM/DSP 模式下时, 该模块按照 FRMSYPW = 1 的情况运行, 而无论 FRMSYPW 的实际值为何。

# PIC24FJ128GB204 系列

## 寄存器 16-2: SPIxCON1H: SPIx 控制寄存器 1 高位字 (续)

bit 6	<b>FRMSYNC:</b> 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 5	<b>FRMPOL:</b> 帧同步 / 从选择极性位 1 = 帧同步脉冲 / 从选择高电平有效 0 = 帧同步脉冲 / 从选择低电平有效
bit 4	<b>MSEN:</b> 主模式从选择使能位 1 = 使能 SPIx 从选择支持, 极性由 FRMPOL 决定 (主模式发送期间自动驱动 $\overline{SSx}$ 引脚) 0 = 禁止 SPIx 从选择支持 ( $\overline{SSx}$ 引脚将由端口 I/O 控制)
bit 3	<b>FRMSYPW:</b> 帧同步脉冲宽度位 1 = 帧同步脉冲为 1 个串行字长宽 (如 MODE<32,16>/WLENGTH<4:0> 所定义) 0 = 帧同步脉冲为 1 个时钟 (SCK) 宽
bit 2-0	<b>FRMCNT&lt;2:0&gt;:</b> 帧同步脉冲计数器位 控制每个同步脉冲发送的串行字数。 111 = 保留 110 = 保留 101 = 每 32 个串行字产生一个帧同步脉冲 100 = 每 16 个串行字产生一个帧同步脉冲 011 = 每 8 个串行字产生一个帧同步脉冲 010 = 每 4 个串行字产生一个帧同步脉冲 001 = 每 2 个串行字产生一个帧同步脉冲 (音频协议使用的值) 000 = 每个串行字产生一个帧同步脉冲

- 注
- 1: 仅当 SPIEN 位 = 0 时才能写 AUDEN。
  - 2: 仅当 SPIEN 位 = 0 时才能写 AUDMONO, 且 AUDMONO 仅在 AUDEN = 1 时有效。
  - 3: URDTEN 仅在 IGNTUR = 1 时有效。
  - 4: 仅当 SPIEN 位 = 0 时才能写 AUDMOD<1:0>, 且 AUDMOD<1:0> 仅在 AUDEN = 1 时有效。当未工作在 PCM/DSP 模式下时, 该模块按照 FRMSYPW = 1 的情况运行, 而无论 FRMSYPW 的实际值为何。

# PIC24FJ128GB204 系列

**寄存器 16-3: SPIxCON2L: SPIx 控制寄存器 2 低位字**

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
—	—	—	WLENGTH<4:0> <sup>(1,2)</sup>				—
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 4-0      **WLENGTH<4:0>:** 可变字长度位 <sup>(1,2)</sup>

- 11111 = 32 位数据
- 11110 = 31 位数据
- 11101 = 30 位数据
- 11100 = 29 位数据
- 11011 = 28 位数据
- 11010 = 27 位数据
- 11001 = 26 位数据
- 11000 = 25 位数据
- 10111 = 24 位数据
- 10110 = 23 位数据
- 10101 = 22 位数据
- 10100 = 21 位数据
- 10011 = 20 位数据
- 10010 = 19 位数据
- 10001 = 18 位数据
- 10000 = 17 位数据
- 01111 = 16 位数据
- 01110 = 15 位数据
- 01101 = 14 位数据
- 01100 = 13 位数据
- 01011 = 12 位数据
- 01010 = 11 位数据
- 01001 = 10 位数据
- 01000 = 9 位数据
- 00111 = 8 位数据
- 00110 = 7 位数据
- 00101 = 6 位数据
- 00100 = 5 位数据
- 00011 = 4 位数据
- 00010 = 3 位数据
- 00001 = 2 位数据
- 00000 = 参见 SPIxCON1L<11:10> 中的 MODE<32,16> 位

- 注 1:** 这些位仅在 AUDEN = 0 时有效。
- 注 2:** 通过更改这些位改变长度不会影响发送 / 接收 FIFO 的深度。

# PIC24FJ128GB204 系列

寄存器 16-4: SPIxSTATL: SPIx 状态寄存器低位字

U-0	U-0	U-0	R/C-0, HS	R-0, HSC	U-0	U-0	R-0, HSC
—	—	—	FRMERR	SPIBUSY	—	—	SPITUR <sup>(1)</sup>
bit 15							bit 8

R-0, HSC	R/C-0, HS	R-1, HSC	U-0	R-1, HSC	U-0	R-0, HSC	R-0, HSC
SRMT	SPIROV	SPIRBE	—	SPITBE	—	SPITBF	SPIRBF
bit 7							bit 0

<b>图注:</b>	C = 可清零位	HS = 可由硬件置 1/ 清零的位
R = 可读位	x = 未知	U = 未实现位, 读为 0
-n = POR 时的值	1 = 置 1	0 = 清零
		HS = 硬件置 1 位

bit 15-13 **未实现:** 读为 0

bit 12 **FRMERR:** SPIx 帧错误状态位  
1 = 检测到帧错误  
0 = 未检测到帧错误

bit 11 **SPIBUSY:** SPIx 活动状态位  
1 = 模块当前正忙于处理一些事务  
0 = 无正在进行的事务 (读取时)

bit 10-9 **未实现:** 读为 0

bit 8 **SPITUR:** SPIx 发送数据不足 (TUR) 状态位<sup>(1)</sup>  
1 = 发送缓冲区遇到发送数据不足条件  
0 = 发送缓冲区未遇到发送数据不足条件

bit 7 **SRMT:** 移位寄存器空状态位 (在增强型缓冲模式下有效)  
1 = 无当前事务或无等待处理的事务 (即, SPIxTXB 或 SPIxTXSR 均不包含要发送的数据)  
0 = 当前事务或等待处理的事务

bit 6 **SPIROV:** SPIx 接收溢出 (ROV) 状态位  
1 = SPIxRXB 满时一个新字节 / 半字 / 字已被完全接收  
0 = 无溢出

bit 5 **SPIRBE:** SPIx 接收缓冲区空状态位  
1 = 接收缓冲区为空  
0 = 接收缓冲区非空

**标准缓冲模式:**

当通过读 SPIxBUF 读 SPIxRXB 时, 该位由硬件自动置 1。当 SPIx 将数据从 SPIxRXSR 传输到 SPIxRXB 时, 该位由硬件自动清零。

**增强型缓冲模式:**

指示 RXELM<5:0> = 6'b000000。

bit 4 **未实现:** 读为 0

bit 3 **SPITBE:** SPIx 发送缓冲区空状态位  
1 = SPIxTXB 为空  
0 = SPIxTXB 非空

**标准缓冲模式:**

当 SPIx 将数据从 SPIxTXB 传输到 SPIxTXSR 时, 该位由硬件自动置 1。当写 SPIxBUF 来装载 SPIxTXB 时, 该位由硬件自动清零。

**增强型缓冲模式:**

指示 TXELM<5:0> = 6'b000000。

**注 1:** SPIEN = 0 时 SPITUR 清零。当 IGNTUR = 1 时, SPITUR 提供数据不足条件的动态状态, 但不停止接收 / 发送操作且不需要由软件清零。

## 寄存器 16-4: SPIxSTATL: SPIx 状态寄存器低位字 (续)

bit 2 未实现: 读为 0

bit 1 **SPITBF:** SPIx 发送缓冲区满状态位

1 = SPIxTXB 已满

0 = SPIxTXB 未满

标准缓冲模式:

当写 SPIxBUF 来装载 SPIxTXB 时, 该位由硬件自动置 1。当 SPIx 将数据从 SPIxTXB 传输到 SPIxTXSR 时, 该位由硬件自动清零。

增强型缓冲模式:

指示 TXELM<5:0> = 6'b111111。

bit 0 **SPIRBF:** SPIx 接收缓冲区满状态位

1 = SPIxRXB 已满

0 = SPIxRXB 未满

标准缓冲模式:

当 SPIx 将数据从 SPIxRXSR 传输到 SPIxRXB 时, 该位由硬件自动置 1。当读 SPIxBUF 来读 SPIxRXB 时, 该位由硬件自动清零。

增强型缓冲模式:

指示 RXELM<5:0> = 6'b111111。

**注 1:** SPIEN = 0 时 SPITUR 清零。当 IGNTUR = 1 时, SPITUR 提供数据不足条件的动态状态, 但不停止接收 / 发送操作且不需要由软件清零。

# PIC24FJ128GB204 系列

寄存器 16-5: SPIxSTATH: SPIx 状态寄存器高位字

U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
—	—	RXELM5 <sup>(3)</sup>	RXELM4 <sup>(2)</sup>	RXELM3 <sup>(1)</sup>	RXELM2	RXELM1	RXELM0
bit 15							bit 8

U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
—	—	TXELM5 <sup>(3)</sup>	TXELM4 <sup>(2)</sup>	TXELM3 <sup>(1)</sup>	TXELM2	TXELM1	TXELM0
bit 7							bit 0

<b>图注:</b>	HSC = 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 15-14 未实现: 读为 0

bit 13-8 **RXELM<5:0>**: 接收缓冲区元素计数位 (在增强型缓冲模式下有效) **(1,2,3)**

bit 7-6 未实现: 读为 0

bit 5-0 **TXELM<5:0>**: 发送缓冲区元素计数位 (在增强型缓冲模式下有效) **(1,2,3)**

- 注
- 1: RXELM3 和 TXELM3 位仅在 FIFODEPTH = 8 或更高时存在。
  - 2: RXELM4 和 TXELM4 位仅在 FIFODEPTH = 16 或更高时存在。
  - 3: RXELM5 和 TXELM5 位仅在 FIFODEPTH = 32 时存在。

# PIC24FJ128GB204 系列

## 寄存器 16-6: SPIxIMSKL: SPIx 中断屏蔽寄存器低位字

U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	
—	—	—	FRMERREN	BUSYEN	—	—	SPITUREN	
bit 15								bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	
SRMTEN	SPIROVEN	SPIRBEN	—	SPITBEN	—	SPITBFEN	SPIRBFEN	
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        **FRMERREN:** 允许通过 FRMERR 产生中断事件位  
                   1 = 帧错误产生中断事件  
                   0 = 帧错误不产生中断事件
- bit 11        **BUSYEN:** 允许通过 SPIBUSY 产生中断事件位  
                   1 = SPIBUSY 产生中断事件  
                   0 = SPIBUSY 不产生中断事件
- bit 10-9     **未实现:** 读为 0
- bit 8         **SPITUREN:** 允许通过 SPITUR 产生中断事件位  
                   1 = 发送数据不足 (TUR) 产生中断事件  
                   0 = 发送数据不足不产生中断事件
- bit 7         **SRMTEN:** 允许通过 SRMT 产生中断事件位  
                   1 = 移位寄存器空 (Shift Register Empty, SRMT) 产生中断事件  
                   0 = 移位寄存器空不产生中断事件
- bit 6         **SPIROVEN:** 允许通过 SPIROV 产生中断事件位  
                   1 = SPIx 接收溢出产生中断事件  
                   0 = SPIx 接收溢出不产生中断事件
- bit 5         **SPIRBEN:** 允许通过 SPIRBE 产生中断事件位  
                   1 = SPIx 接收缓冲区空产生中断事件  
                   0 = SPIx 接收缓冲区空不产生中断事件
- bit 4         **未实现:** 读为 0
- bit 3         **SPITBEN:** 允许通过 SPITBE 产生中断事件位  
                   1 = SPIx 发送缓冲区空产生中断事件  
                   0 = SPIx 发送缓冲区空不产生中断事件
- bit 2         **未实现:** 读为 0
- bit 1         **SPITBFEN:** 允许通过 SPITBF 产生中断事件位  
                   1 = SPIx 发送缓冲区满产生中断事件  
                   0 = SPIx 发送缓冲区满不产生中断事件
- bit 0         **SPIRBFEN:** 允许通过 SPIRBF 产生中断事件位  
                   1 = SPIx 接收缓冲区满产生中断事件  
                   0 = SPIx 接收缓冲区满不产生中断事件

# PIC24FJ128GB204 系列

寄存器 16-7: SPIxIMSKH: SPIx 中断屏蔽寄存器高位字

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RXWIEN	—	RXMSK5 <sup>(1)</sup>	RXMSK4 <sup>(1,4)</sup>	RXMSK3 <sup>(1,3)</sup>	RXMSK2 <sup>(1,2)</sup>	RXMSK1 <sup>(1)</sup>	RXMSK0 <sup>(1)</sup>
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
TXWIEN	—	TXMSK5 <sup>(1)</sup>	TXMSK4 <sup>(1,4)</sup>	TXMSK3 <sup>(1,3)</sup>	TXMSK2 <sup>(1,2)</sup>	TXMSK1 <sup>(1)</sup>	TXMSK0 <sup>(1)</sup>
bit 7							bit 0

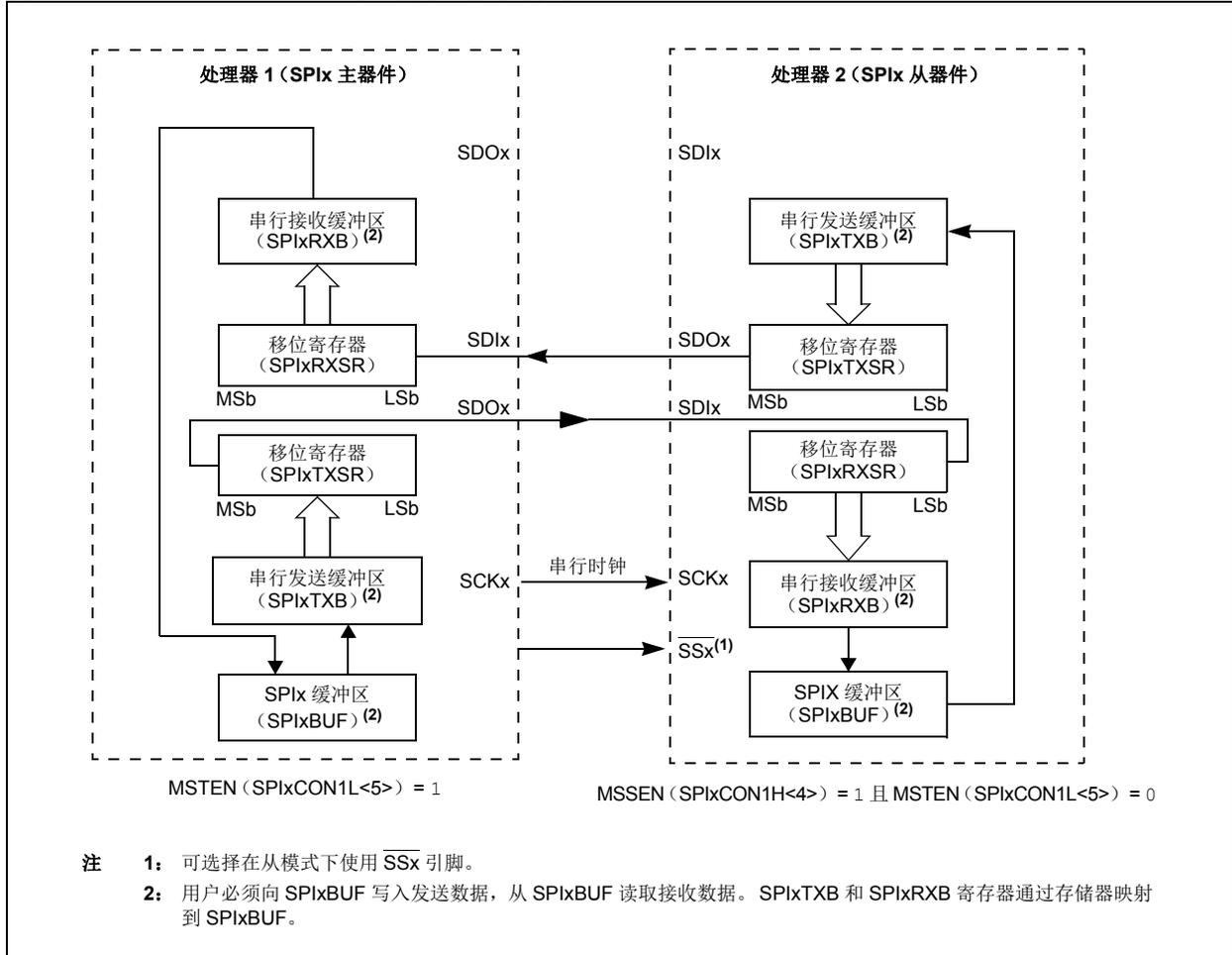
**图注:**

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     **RXWIEN:** 接收水印中断允许位  
 1 = RXMSK<5:0> ≤ RXELM<5:0> 时触发接收缓冲区元素水印中断  
 0 = 禁止接收缓冲区元素水印中断
- bit 14     **未实现:** 读为 0
- bit 13-8   **RXMSK<5:0>:** 接收缓冲区屏蔽位 **(1,2,3,4)**  
 接收屏蔽位, 与 RXWIEN 位配合使用。
- bit 7     **TXWIEN:** 发送水印中断允许位  
 1 = TXMSK<5:0> = TXELM<5:0> 时触发发送缓冲区元素水印中断  
 0 = 禁止发送缓冲区元素水印中断
- bit 6     **未实现:** 读为 0
- bit 5-0   **TXMSK<5:0>:** 发送缓冲区屏蔽位 **(1,2,3,4)**  
 发送屏蔽位, 与 TXWIEN 配合使用。

- 注    **1:** 高于 FIFODEPTH 的屏蔽值无效。在该情况下, 模块将不会触发任何值匹配。  
**2:** RXMSK2 和 TXMSK2 位仅在 FIFODEPTH = 8 或更高时存在。  
**3:** RXMSK3 和 TXMSK3 位仅在 FIFODEPTH = 16 或更高时存在。  
**4:** RXMSK4 和 TXMSK4 位仅在 FIFODEPTH = 32 时存在。

图 16-3: SPIx 主 / 从连接 (标准模式)



# PIC24FJ128GB204 系列

图 16-4: SPIx 主 / 从连接 (增强型缓冲模式)

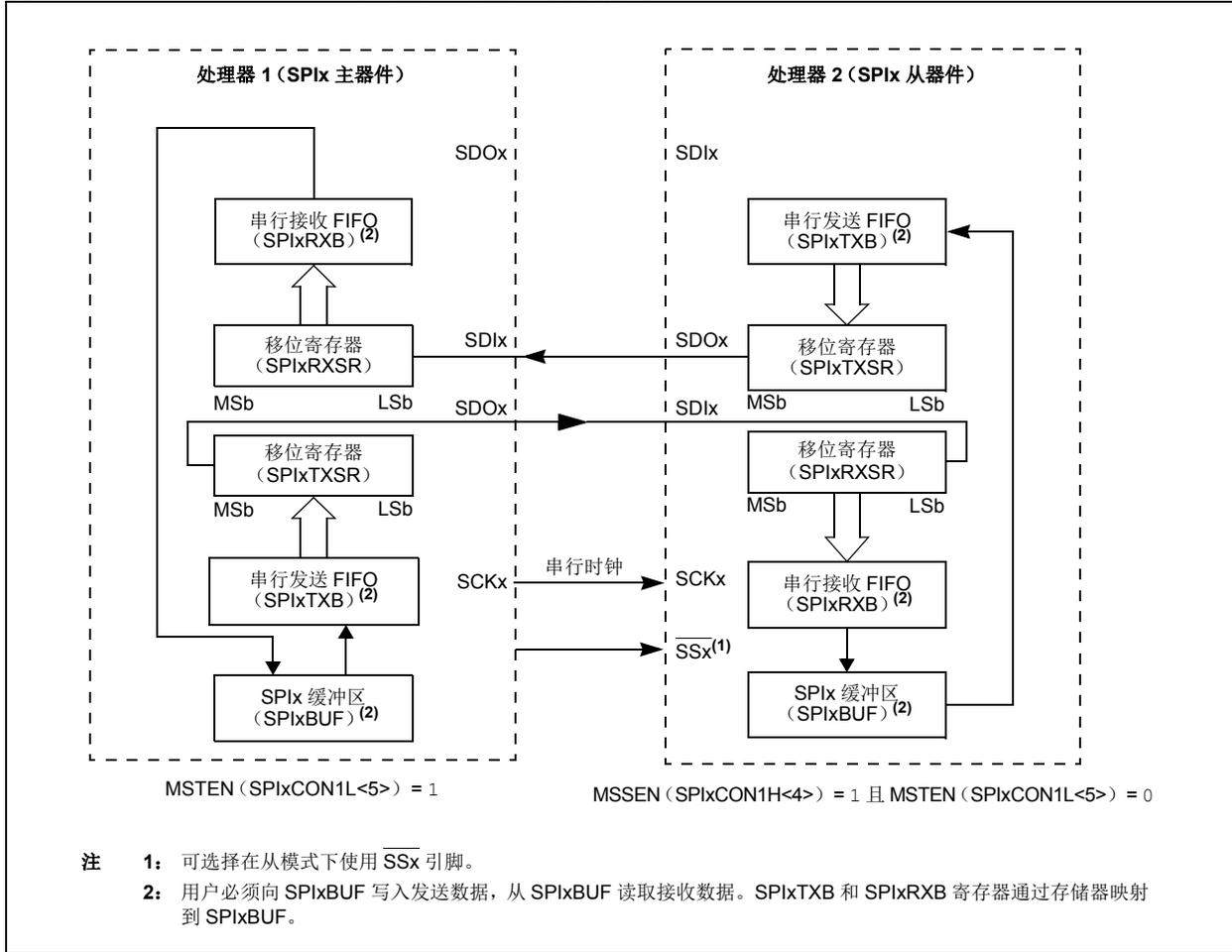


图 16-5: SPIx 主器件、帧主器件连接图

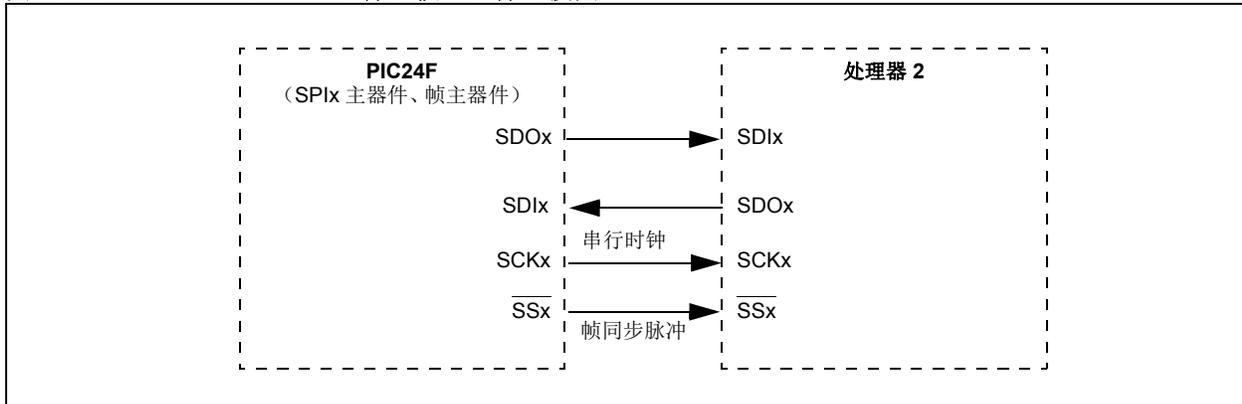


图 16-6: SPIx 主器件、帧从器件连接图

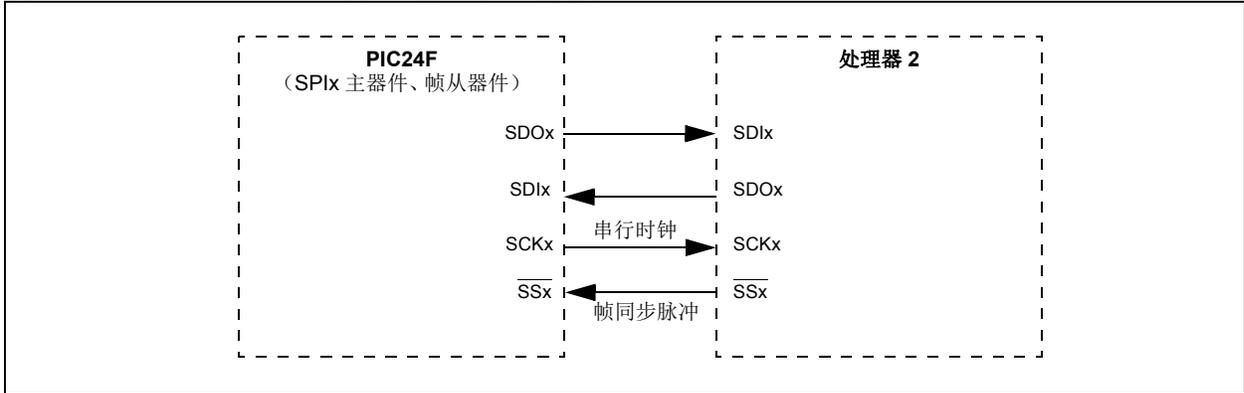


图 16-7: SPIx Slave, Frame Master Connection Diagram

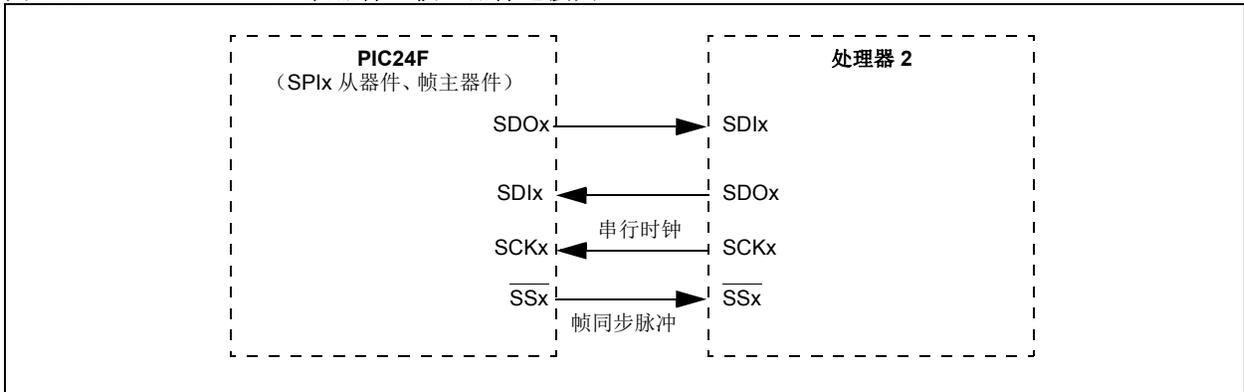
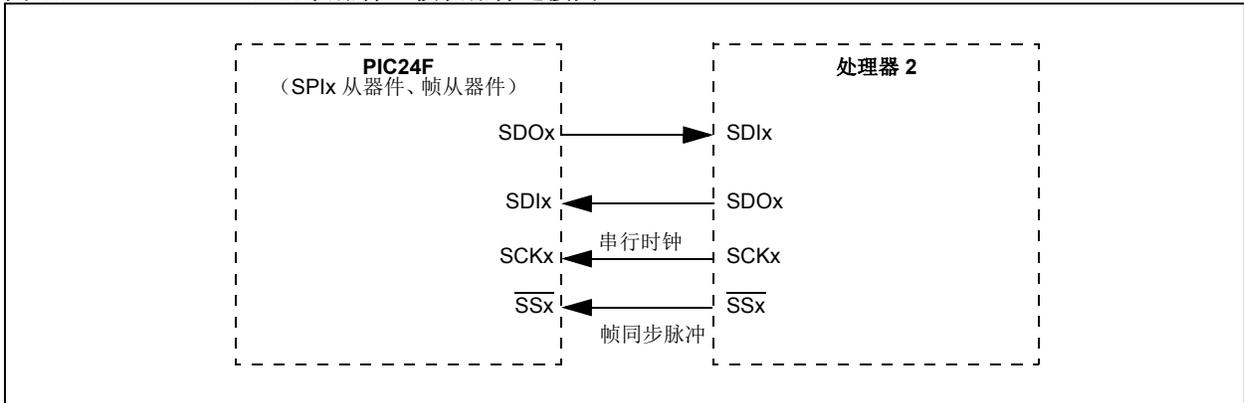


图 16-8: SPIx Slave, Frame Slave Connection Diagram



公式 16-1: 器件速率和 SPIx 时钟速率之间的关系

$$\text{波特率} = \frac{FPB}{(2 * (SPIxBRG + 1))}$$

其中:

FPB 是外设总线时钟频率。

# PIC24FJ128GB204 系列

---

注:

## 17.0 I<sup>2</sup>C™

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24F 系列参考手册》中的“i<sup>2</sup>C™”（DS70000195）。本数据手册中的信息取代了FRM中的信息。

I<sup>2</sup>C 模块是用于与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、显示屏驱动器和 A/D 转换器等。

I<sup>2</sup>C 模块支持以下特性：

- 独立的主 / 从逻辑
- 7 位和 10 位器件地址
- I<sup>2</sup>C 协议中定义的广播呼叫地址
- 时钟延长，为处理器提供延时以响应从器件的数据请求
- 支持 100 kHz 和 400 kHz 两种总线规范
- 可配置的地址掩码
- 多主器件模式，可以防止在仲裁时丢失报文
- 总线重发器模式，允许模块作为从器件接收来自所有地址的所有报文
- 自动 SCL

图 17-1 给出了该模块的框图。

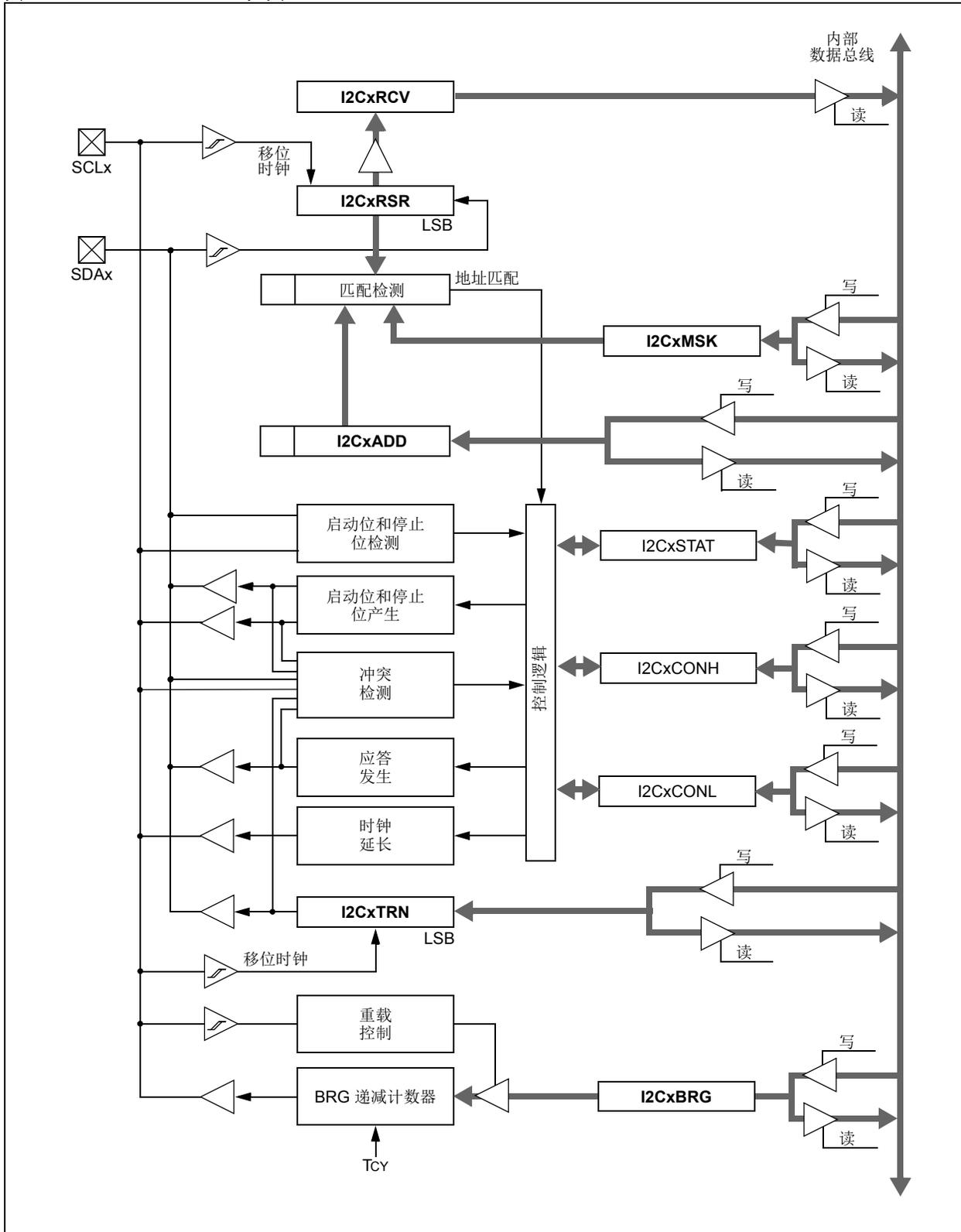
## 17.1 作为主器件在单主器件环境中通信

在主模式下发送报文的细节取决于要进行通信的器件的通信协议。通常，事件序列如下：

1. 在 SDA<sub>x</sub> 和 SCL<sub>x</sub> 上发送一个启动条件。
2. 发送一个 I<sup>2</sup>C 器件地址字节到从器件，表明要执行写操作。
3. 等待并验证从器件的应答。
4. 发送第一个数据字节（有时是命令）到从器件。
5. 等待并验证从器件的应答。
6. 发送串行存储器地址低字节到从器件。
7. 重复步骤 4 和 5，直至发送完所有数据字节。
8. 在 SDA<sub>x</sub> 和 SCL<sub>x</sub> 上发送一个重复启动条件。
9. 发送一个器件地址字节到从器件，表明要执行读操作。
10. 等待并验证从器件的应答。
11. 使能主器件接收以接收串行存储器数据。
12. 在数据字节接收完毕时，产生 ACK 或 NACK 条件。
13. 在 SDA<sub>x</sub> 和 SCL<sub>x</sub> 上产生一个停止条件。

# PIC24FJ128GB204 系列

图 17-1: I<sup>2</sup>C™ 框图



## 17.2 设置用作总线主器件时的波特率

使用公式 17-1 计算波特率发生器的重载值。

公式 17-1: 计算波特率重载值<sup>(1)</sup>

$$I2CxBRG = \left( \left( \frac{1}{F_{SCL}} - PGDX \right) \times \frac{FCY}{2} \right) - 2$$

注 1: 基于  $F_{CY} = F_{OSC}/2$ ; 禁止打盹模式和 PLL。

## 17.3 从地址掩码

I2CxMSK 寄存器 (见寄存器 17-4) 指定了 7 位和 10 位寻址模式下某些“无关”地址位的位置。将 I2CxMSK 寄存器中某个特定位置 1 (= 1), 不论相应的地址位是 0 还是 1, 从模块都会做出响应。例如, 当将 I2CxMSK 设置为 0010000000 时, 从模块将检测两个地址 0000000000 和 0010000000。

为了使能地址掩码, 必须通过将 STRICT 位 (I2CxCONL<11>) 清零来禁止智能外设管理接口 (Intelligent Peripheral Management Interface, IPMI)。

注: 由于修改了 I<sup>2</sup>C™ 协议, 表 17-1 中的地址将被保留, 并且在从模式下不会被应答。这同样适用于包含任意这些地址在内的所有地址掩码设置。

表 17-1: I<sup>2</sup>C™ 保留地址<sup>(1)</sup>

从地址	R/W 位	说明
0000 000	0	广播呼叫地址 <sup>(2)</sup>
0000 000	1	启动字节
0000 001	x	Cbus 地址
0000 01x	x	保留
0000 1xx	x	HS 模式主代码
1111 0xx	x	10 位从地址高字节 <sup>(3)</sup>
1111 1xx	x	保留

- 注 1: 上述地址位永远不会导致地址匹配, 无论地址掩码的设置如何。  
 2: 仅当 GCEN = 1 时, 才会应答该地址。  
 3: 仅在 10 位寻址模式下才会与地址的高字节匹配。

# PIC24FJ128GB204 系列

寄存器 17-1: I2CxCONL: I2Cx 控制寄存器低字节

R/W-0	U-0	R/W-0, HC	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL <sup>(1)</sup>	STRICT	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

<b>图注:</b>	HC = 可由硬件清零的位
R = 可读位	W = 可写位
-n = POR 时的值	1 = 置 1
	U = 未实现位, 读为 0
	0 = 清零
	x = 未知

- bit 15      **I2CEN:** I2Cx 使能位 (仅可从 SW 写入)  
 1 = 使能 I<sup>2</sup>C™ 模块并将 SDAx 和 SCLx 引脚配置为串行端口引脚  
 0 = 禁止 I<sup>2</sup>C 模块; 所有 I<sup>2</sup>C 引脚由端口功能控制
- bit 14      **未实现:** 读为 0
- bit 13      **I2CSIDL:** 空闲模式 I2Cx 停止位  
 1 = 当器件进入空闲模式后, 模块停止工作  
 0 = 模块在空闲模式下继续工作
- bit 12      **SCLREL:** SCLx 释放控制位 (仅适用于 I<sup>2</sup>C 从模式) <sup>(1)</sup>  
 模块复位并且 (I2CEN = 0) 设置 SCLREL = 1。  
如果 STREN = 0: <sup>(2)</sup>  
 1 = 释放时钟  
 0 = 强制时钟为低电平 (时钟延长)  
如果 STREN = 1:  
 1 = 释放时钟  
 0 = 保持时钟为低电平 (时钟延长); 用户可以将该位编程为 0, 时钟在下一个 SCLx 低电平时延长
- bit 11      **STRICT:** I2Cx 严格保留地址规则使能位  
 1 = 强制执行严格保留寻址规则; 关于保留地址, 请参见表 17-1。  
 (在从模式下) —— 器件不响应保留地址空间, 属于该类的地址均不被应答。  
 (在主模式下) —— 允许器件生成具有保留地址空间的地址。  
 0 = 保留寻址将被应答。  
 (在从模式下) —— 器件将响应位于保留地址空间中的地址。当任何保留地址发生匹配时, 器件将产生 ACK。  
 (在主模式下) —— 保留。
- bit 10      **A10M:** 10 位从器件地址标志位  
 1 = I2CxADD 是一个 10 位从地址  
 0 = I2CxADD 是一个 7 位从地址
- bit 9      **DISSLW:** 禁止压摆率控制位  
 1 = 标准速度模式下禁止压摆率控制 (100 kHz, 1 MHz 模式下同样禁止)  
 0 = 高速模式下使能压摆率控制 (400 kHz)
- bit 8      **SMEN:** SMBus 输入电平使能位  
 1 = 使能输入逻辑以使门限值符合 SMBus 规范  
 0 = 禁止 SMBus 特定输入

注 1: 在从器件发送开始时自动清零; 在从器件接收结束时自动清零。  
 2: 在从器件发送开始时自动清零。

## 寄存器 17-1: I2CxCONL: I2Cx 控制寄存器低字节 (续)

- bit 7      **GCEN:** 广播呼叫使能位 (仅适用于 I<sup>2</sup>C 从模式)  
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  
仅在 I<sup>2</sup>C 从模式下与 SCLREL 位一起使用。  
1 = 使能时钟延长  
0 = 禁止时钟延长
- bit 5      **ACKDT:** 应答数据位  
在 I<sup>2</sup>C 主模式下, 在主器件接收模式期间 —— 用户在接收结束时启动应答序列时将发送的值。  
在 I<sup>2</sup>C 从模式下, AHEN = 1 或 DHEN = 1 时 —— 从器件在地址或数据接收结束时启动应答序列时将发送的值。  
1 = 发送 NACK  
0 = 发送 ACK
- bit 4      **ACKEN:** 应答序列使能位  
仅在 I<sup>2</sup>C 主模式下; 适用于主器件接收模式。  
1 = 在 SDAx 和 SCLx 引脚上启动应答序列, 并发送 ACKDT 数据位  
0 = 应答序列空闲
- bit 3      **RCEN:** 接收使能位 (仅适用于 I<sup>2</sup>C 主模式)  
1 = 使能 I<sup>2</sup>C 接收模式。在接收完 8 位数据字节后由硬件自动清零  
0 = 没有启动接收序列
- bit 2      **PEN:** 停止条件使能位 (仅适用于 I<sup>2</sup>C 主模式)  
1 = 在 SDAx 和 SCLx 引脚上发出停止条件  
0 = 停止条件空闲
- bit 1      **RSEN:** 重复启动条件使能位 (仅适用于 I<sup>2</sup>C 主模式)  
1 = 在 SDAx 和 SCLx 引脚上发出重复启动条件  
0 = 重复启动条件空闲
- bit 0      **SEN:** 启动条件使能位 (仅适用于 I<sup>2</sup>C 主模式)  
1 = 在 SDAx 和 SCLx 引脚上发出启动条件  
0 = 启动条件空闲

- 注    1: 在从器件发送开始时自动清零; 在从器件接收结束时自动清零。  
      2: 在从器件发送开始时自动清零。

# PIC24FJ128GB204 系列

寄存器 17-2: I2CxCONH: I2Cx 控制寄存器高字节

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

U-0	R/W-0						
—	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
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        **PCIE:** 停止条件中断允许位 (仅适用于 I<sup>2</sup>C 从模式)  
 1 = 检测到停止条件时允许中断  
 0 = 禁止停止检测中断
- bit 5        **SCIE:** 启动条件中断允许位 (仅适用于 I<sup>2</sup>C 从模式)  
 1 = 检测到启动或重复启动条件时允许中断  
 0 = 禁止启动检测中断
- bit 4        **BOEN:** 缓冲区改写使能位 (仅适用于 I<sup>2</sup>C 从模式)  
 1 = 更新 I2CxRCV, 接收到地址 / 数据字节时生成 ACK, 仅当 RBF 位 = 0 时, 忽略 I2COV 位的状态  
 0 = 仅当 I2COV 清零时才更新 I2CxRCV
- bit 3        **SDAHT:** SDAx 保持时间选择位  
 1 = SCLx 下降沿之后 SDAx 上的最短保持时间为 300 ns  
 0 = SCLx 下降沿之后 SDAx 上的最短保持时间为 100 ns
- bit 2        **SBCDE:** 从模式总线冲突检测允许位 (仅适用于 I<sup>2</sup>C 从模式)  
 如果在 SCLx 的上升沿, 在模块输出高电平状态时采样到 SDAx 为低电平, 则 BCL 位会置 1, 总线会变为空闲状态。该检测模式仅在数据和 ACK 发送序列期间有效。  
 1 = 允许从器件总线冲突中断  
 0 = 禁止从器件总线冲突中断
- bit 1        **AHEN:** 地址保持使能位 (仅适用于 I<sup>2</sup>C 从模式)  
 1 = 在接收匹配地址字节的 SCLx 的第 8 个下降沿后, SCLREL 位 (I2CxCONL<12>) 清零, SCLx 保持低电平  
 0 = 禁止地址保持
- bit 0        **DHEN:** 数据保持使能位 (仅适用于 I<sup>2</sup>C 从模式)  
 1 = 在接收数据字节的 SCLx 的第 8 个下降沿后, 从硬件清零 SCLREL 位 (I2CxCONL<12>), SCLx 保持低电平  
 0 = 禁止数据保持

# PIC24FJ128GB204 系列

## 寄存器 17-3: I2CxSTAT: I2Cx 状态寄存器

R-0, HSC	R-0, HSC	R-0, HSC	U-0	U-0	R/C-0, HSC	R-0, HSC	R-0, HSC
ACKSTAT	TRSTAT	ACKTIM	—	—	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 $\bar{A}$	P	S	R $\bar{W}$	RBF	TBF
bit 7							bit 0

<b>图注:</b>	C = 可清零位	HSC = 可由硬件置 1/ 清零的位
R = 可读位	HS = 可由硬件置 1 的位	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 (在所有主模式和从模式下更新)  
 1 = 未收到来自从器件的应答  
 0 = 收到来自从器件的应答
- bit 14 **TRSTAT:** 发送状态位 (作为 I<sup>2</sup>C 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
- bit 13 **ACKTIM:** 应答时间状态位 (仅在 I<sup>2</sup>C 从模式下有效)  
 1 = 指示 I<sup>2</sup>C 总线处于应答序列中, 在 SCLx 时钟的第 8 个下降沿置 1  
 0 = 非应答序列, 在 SCLx 时钟的第 9 个上升沿清零
- bit 12-11 **未实现:** 读为 0
- bit 10 **BCL:** 总线冲突检测位 (主 / 从模式; 禁止 I<sup>2</sup>C 模块 (I2CEN = 0) 时清零)  
 1 = 主器件或从器件发送操作期间检测到了总线冲突  
 0 = 未检测到总线冲突
- bit 9 **GCSTAT:** 广播呼叫状态位 (在停止检测后清零)  
 1 = 接收到广播呼叫地址  
 0 = 未接收到广播呼叫地址
- bit 8 **ADD10:** 10 位地址状态位 (在停止检测后清零)  
 1 = 10 位地址匹配  
 0 = 10 位地址不匹配
- bit 7 **IWCOL:** I2Cx 写冲突检测位  
 1 = 因为 I<sup>2</sup>C 模块忙, 尝试写 I2CxTRN 寄存器的操作失败; 必须用软件清零  
 0 = 未发生冲突
- bit 6 **I2COV:** I2Cx 接收溢出标志位  
 1 = 当 I2CxRCV 寄存器仍存有前一字节时接收到了新字节; 在发送模式下, I2COV 为“无关”位, 必须由软件清零  
 0 = 无溢出
- bit 5 **D $\bar{A}$ :** 数据 / 地址位 (作为 I<sup>2</sup>C 从器件工作时)  
 1 = 表示上次接收的字节为数据  
 0 = 表示上次接收或发送的字节为地址
- bit 4 **P:** I2Cx 停止位  
 当检测到启动、复位或停止条件时更新; 禁止 I<sup>2</sup>C 模块 (I2CEN = 0) 时清零。  
 1 = 表示上次检测到停止位  
 0 = 上次未检测到停止位

# PIC24FJ128GB204 系列

## 寄存器 17-3: I2CxSTAT: I2Cx 状态寄存器 (续)

- bit 3      **S:** I2Cx 启动位  
 当检测到启动、复位或停止条件时更新；禁止 I<sup>2</sup>C 模块 (I2CEN = 0) 时清零。  
 1 = 表示上次检测到启动位 (或重复启动位)  
 0 = 上次未检测到启动位
- bit 2      **R/W:** 读 / 写信息位 (作为 I<sup>2</sup>C 从器件工作时)  
 1 = 读: 表示数据传输自从器件输出  
 0 = 写: 表示数据传输到从器件
- bit 1      **RBF:** 接收缓冲区满状态位  
 1 = 接收完成, I2CxRCV 已满  
 0 = 接收未完成, I2CxRCV 为空
- bit 0      **TBF:** 发送缓冲区满状态位  
 1 = 发送正在进行, I2CxTRN 已满 (8 位数据)  
 0 = 发送完成, I2CxTRN 为空

## 寄存器 17-4: 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
—	—	—	—	—	—	MSK<9:8>	
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
MSK<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 15-10      **未实现:** 读为 0
- bit 9-0      **MSK<9:0>:** I2Cx 地址位 x 的掩码选择位  
 1 = 使能输入报文地址 bit x 的掩码; 在此位置上不需要位匹配  
 0 = 禁止 bit x 的掩码; 在此位置上需要位匹配

## 18.0 通用异步收发器 (UART)

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24系列参考手册》中的“通用异步收发器 (UART)” (DS70000582)。本数据手册中的信息取代了FRM中的信息。

通用异步收发器 (Universal Synchronous Receiver Transmitter, UART) 模块是 PIC24F 系列器件提供的串行 I/O 模块之一。UART 是可以和外设 (例如个人电脑、LIN/J2602、RS-232 和 RS-485 接口) 通信的全双工异步系统。UART 模块还通过 UxCTS 和 UxRTS 引脚支持硬件流控制。UART 模块还包含符合 ISO 7816 的智能卡和 IrDA<sup>®</sup> 编码器 / 解码器单元。

PIC24FJ128GB204 系列器件具有 4 个 UART 模块，称为 UART1、UART2、UART3 和 UART4。

UARTx 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 引脚实现硬件流控制
- 完全集成的具有 16 位预分频器的波特率发生器
- 当器件工作在 16 MIPS 下 16 倍频模式时，波特率范围为 15 bps 至 1 Mbps

- 当器件工作在 16 MIPS 下 4 倍频模式时，波特率范围为 61 bps 至 4 Mbps
- 4 级深先进先出 (First-In-First-Out, FIFO) 发送数据缓冲区
- 4 级深 FIFO 接收数据缓冲区
- 奇偶校验、帧和缓冲区溢出错误检测
- 支持带地址检测的 9 位模式 (第 9 位 = 1)
- 独立的发送和接收中断
- 支持诊断的环回模式
- 发送和接收线路的极性控制
- 支持同步和间隔字符
- 支持自动波特率检测
- IrDA<sup>®</sup> 编码器和解码器逻辑
- 支持 DMA
- 支持 IrDA 的 16 倍频波特率时钟输出
- 支持智能卡 ISO 7816 (仅 UART1 和 UART2)：
  - T = 0 协议，具有自动错误处理功能
  - T = 1 协议
  - 专用保护时间计数器 (GTC)
  - 专用等待时间计数器 (WTC)

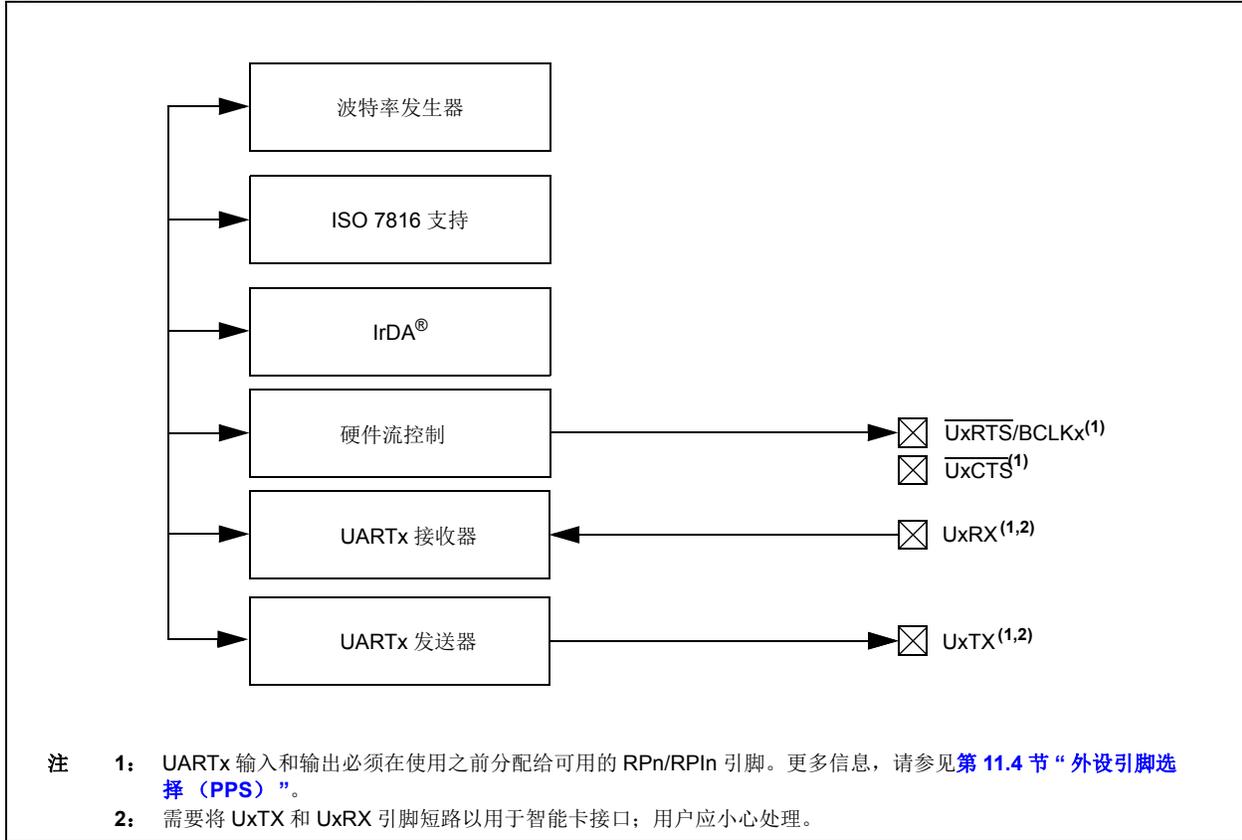
UARTx 模块的简化框图如图 18-1 所示。UARTx 模块由以下重要硬件组成：

- 波特率发生器
- 异步发送器
- 异步接收器

**注：** 在本章中，与特定 UART 模块相关的寄存器名称和位名称的引用一般都使用“x”代替特定模块编号。因此，“UxSTA”可能指 UART1、UART2、UART3 或 UART4 的状态寄存器。

# PIC24FJ128GB204 系列

图 18-1: UARTx 简化框图



## 18.1 UARTx 波特率发生器 (BRG)

UARTx 模块包含一个专用的 16 位波特率发生器。UxBRG 寄存器控制一个独立运行的 16 位定时器的周期。公式 18-1 给出了 BRGH = 0 时计算波特率的公式。

公式 18-1: **BRGH = 0 时的 UARTx 波特率 (1,2)**

$$\text{波特率} = \frac{F_{CY}}{16 \cdot (UxBRG + 1)}$$

$$UxBRG = \frac{F_{CY}}{16 \cdot \text{波特率}} - 1$$

注 1: F<sub>CY</sub> 表示指令周期的时钟频率 (F<sub>OSC</sub>/2)。

注 2: 基于 F<sub>CY</sub> = F<sub>OSC</sub>/2; 禁止打盹模式和 PLL。

例 18-1 给出了下列条件下的波特率误差计算:

- F<sub>CY</sub> = 4 MHz
- 目标波特率 = 9600

例 18-1: **波特率误差计算 (BRGH = 0) (1)**

$$\text{目标波特率} = F_{CY} / (16 (UxBRG + 1))$$

求解 UxBRG 的值:

$$UxBRG = ((F_{CY} / \text{目标波特率}) / 16) - 1$$

$$UxBRG = ((4000000 / 9600) / 16) - 1$$

$$UxBRG = 25$$

计算的波特率 = 4000000 / (16 (25 + 1)) = 9615

误差 =  $\frac{\text{计算的波特率} - \text{目标波特率}}{\text{目标波特率}}$

$$= \frac{9615 - 9600}{9600} = 0.16\%$$

注 1: 基于 F<sub>CY</sub> = F<sub>OSC</sub>/2; 禁止打盹模式和 PLL。

可能的最大波特率 (BRGH = 0) 是 F<sub>CY</sub>/16 (当 UxBRG = 0 时), 可能的最小波特率是 F<sub>CY</sub>/(16 \* 65536)。

公式 18-2 给出了 BRGH = 1 时计算波特率的公式。

公式 18-2: **BRGH = 1 时的 UARTx 波特率 (1,2)**

$$\text{波特率} = \frac{F_{CY}}{4 \cdot (UxBRG + 1)}$$

$$UxBRG = \frac{F_{CY}}{4 \cdot \text{波特率}} - 1$$

注 1: F<sub>CY</sub> 表示指令周期时钟频率。

注 2: 基于 F<sub>CY</sub> = F<sub>OSC</sub>/2; 禁止打盹模式和 PLL。

可能的最大波特率 (BRGH = 1) 是 F<sub>CY</sub>/4 (当 UxBRG = 0 时), 可能的最小波特率是 F<sub>CY</sub>/(4 \* 65536)。

向 UxBRG 寄存器中写入新值会导致 BRG 定时器复位 (清零)。这保证了 BRG 在产生新的波特率之前不需要等待定时器溢出。

## 18.2 8 位数据模式下的发送

1. 设置 UARTx:
  - 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 UARTx。
3. 将 UTXEN 位置 1（置 1 后两个周期内产生发送中断）。
4. 将数据字节写入 UxTXREG 字的低字节。该数据字节将被立即发送给发送移位寄存器（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. 另外，数据字节也可在 UTXEN = 0 时被传送，且随后用户可将 UTXEN 置 1。由于波特率时钟是从清零状态启动的，所以该行为能立即开始发送串行比特流。
6. 将根据中断控制位 UTXISEL<1:0> 的设置产生发送中断。

## 18.3 9 位数据模式下的发送

1. 设置 UARTx（如第 18.2 节“8 位数据模式下的发送”所述）。
2. 使能 UARTx。
3. 将 UTXEN 位置 1（产生发送中断）。
4. 只将 16 位值写入 UxTXREG。
5. 向 UxTXREG 写入一个字可触发 9 位数据向 TSR 传送。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. 将根据中断控制位 URXISEL<1:0> 的设置产生发送中断。

## 18.4 间隔和同步发送序列

下面的序列将发送一个由间隔字符和其后的自动波特率同步字节组成的报文帧头。

1. 将 UARTx 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 寄存器中以启动发送（该值会被忽略）。
4. 向 UxTXREG 写入 55h，这会将同步字符装入发送 FIFO 中。
5. 当间隔字符发送完成后，由硬件将 UTXBRK 位复位。然后开始发送同步字符。

## 18.5 8 位或 9 位数据模式下的接收

1. 设置 UARTx（如第 18.2 节“8 位数据模式下的发送”所述）。
2. 使能 UARTx。
3. 将 URXEN 位（UxSTA<12>）置 1。
4. 当接收到一个或多个数据字符时，将会根据中断控制位 URXISEL<1:0> 的设置产生接收中断。
5. 读 OERR 位以确定是否发生了溢出错误。必须用软件将 OERR 位复位。
6. 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。

## 18.6 UxCTS 和 UxRTS 控制引脚的操作

UARTx 允许发送（UxCTS）和 UARTx 请求发送（UxRTS）是两个与 UARTx 模块有关由硬件控制的引脚。这两个引脚允许 UARTx 运行在单工模式和流控制模式下。这两个引脚控制数据终端设备（Data Terminal Equipment, DTE）之间的数据发送和接收。通过 UxMODE 寄存器的 UEN<1:0> 位来配置这两个引脚。

## 18.7 红外支持

UARTx 模块提供两种类型的红外 UART 支持：一种是 IrDA 时钟输出，支持外部 IrDA 编码器和解码器（支持传统模块）；另一种是完全实现的 IrDA 编码器和解码器。注意，因为 IrDA 模式需要 16 倍频的波特率时钟，所以它们仅在 BRGH 位（UxMODE<3>）为 0 时才能工作。

### 18.7.1 支持外部 IrDA 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（与 UxRTS 引脚相同）配置为产生 16 倍频的波特率时钟。当使能了 UARTx 模块且 UEN<1:0> = 11 时，BCLKx 引脚将输出 16 倍频的波特率时钟，用于支持 IrDA 编解码器芯片。

## 18.7.2 内置 IrDA 编码器和解码器

UARTx 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 (即 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。

## 18.8 智能卡 ISO 7816 支持

图 18-2 给出了一个将具有 UARTx 模块的 PIC24F 单片机用于智能卡数据通信的智能卡子系统。可通过终端或外部电源提供 Vcc 以为智能卡供电。终端还负责为智能

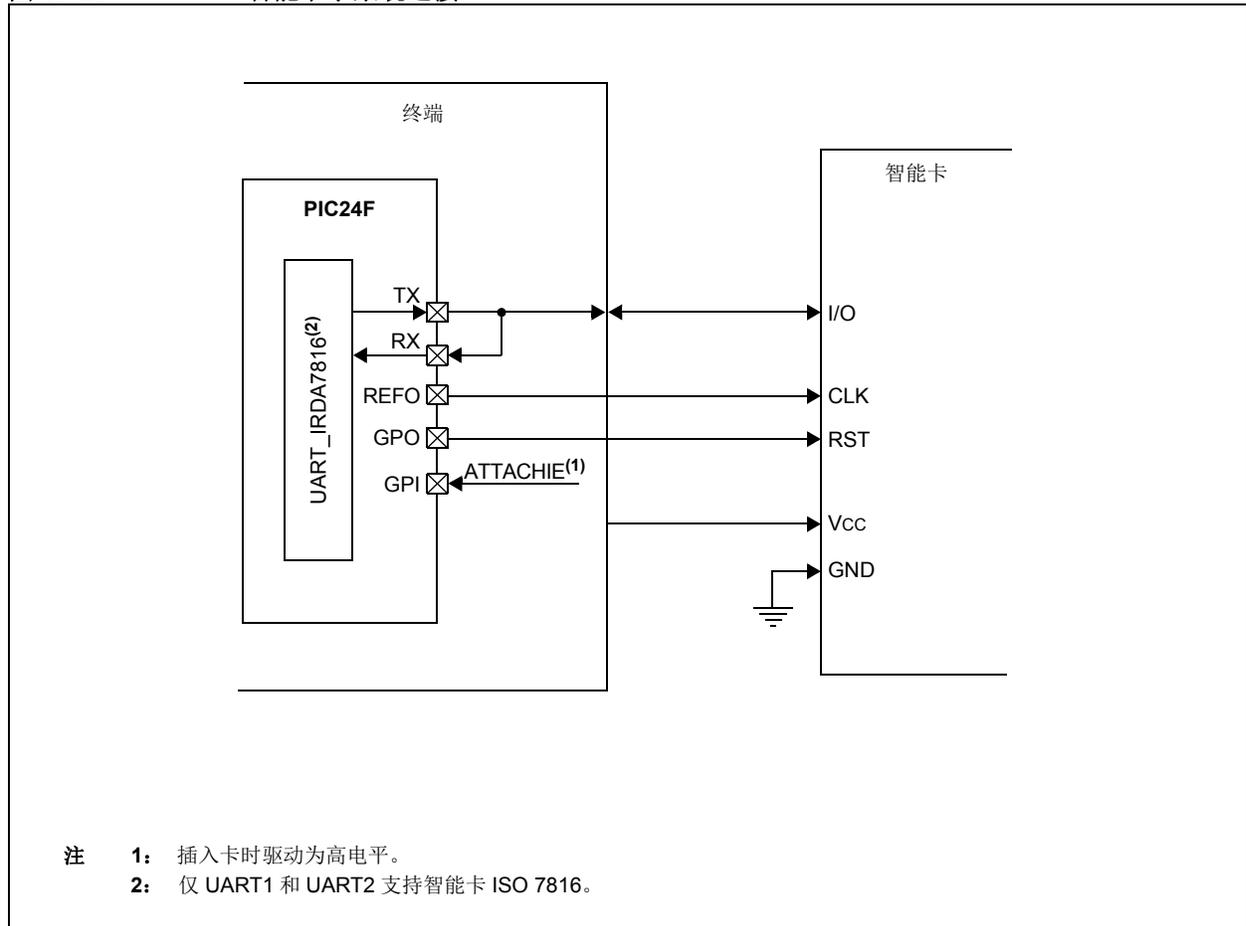
卡提供时钟并复位智能卡。必须外部短接 PIC24F 器件的 TX 和 RX 引脚, 然后连接到智能卡的 I/O 引脚。

有两个广泛用于终端和智能卡之间的智能卡通信的协议:

- T = 0 (异步、半双工、面向字节的协议)
- T = 1 (异步、半双工、面向块的协议)

使用 UxSCCON 寄存器中的 PTRCL 位来选择 T = 0 或 T = 1 协议。

图 18-2: 智能卡子系统连接



# PIC24FJ128GB204 系列

## 18.9 控制寄存器

UART 模块由以下特殊功能寄存器（SFR）组成：

- UxMODE: UARTx 模式寄存器（寄存器 18-1）
- UxSTA: UARTx 状态和控制寄存器（寄存器 18-2）
- UxRXREG: UARTx 接收寄存器
- UxTXREG: UARTx 发送寄存器（只写）（寄存器 18-3）
- UxADMD: UARTx 地址掩码检测寄存器（寄存器 18-4）
- UxBRG: UARTx 波特率寄存器
- UxSCCON: UARTx 智能卡控制寄存器（寄存器 18-5）
- UxSCINT: UARTx 智能卡中断寄存器（寄存器 18-6）
- UxGTC: UARTx 保护时间寄存器
- UxWTCL 和 UxWTCH: UARTx 等待时间计数器低位字 / 高位字

寄存器 18-1: 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 <sup>(1)</sup>	—	USIDL	IREN <sup>(2)</sup>	RTSMD	—	UEN1	UEN0
bit 15							bit 8

R/W-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	URXINV	BRGH	PDSEL1	PDSEL0	STSEL
bit 7							bit 0

<b>图注:</b>	HC = 可由硬件清零的位
R = 可读位	W = 可写位
-n = POR 时的值	1 = 置 1
	U = 未实现位, 读为 0
	0 = 清零
	x = 未知

- bit 15      **UARTEN:** UARTx 使能位 <sup>(1)</sup>  
 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:** UARTx 空闲模式停止位  
 1 = 当器件进入空闲模式后, 模块停止工作  
 0 = 模块在空闲模式下继续工作
- bit 12      **IREN:** IrDA<sup>®</sup> 编码器和解码器使能位 <sup>(2)</sup>  
 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:** UxRTS 引脚模式选择位  
 1 = UxRTS 引脚处于单工模式  
 0 = UxRTS 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 UxTX、UxRX 和 BCLKx 引脚; UxCTS 引脚由端口锁存器控制  
 10 = 使能并使用 UxTX、UxRX、UxCTS 和 UxRTS 引脚  
 01 = 使能并使用 UxTX、UxRX 和 UxRTS 引脚; UxCTS 引脚由端口锁存器控制  
 00 = 使能并使用 UxTX 和 UxRX 引脚; UxCTS 和 UxRTS/BCLKx 引脚由端口锁存器控制

注 1: 如果 UARTEN = 1, 外设的输入和输出必须配置给可用的 RPN/RPIN 引脚。更多信息, 请参见第 11.4 节“**外设引脚选择 (PPS)**”。

注 2: 此功能仅适用于 16 倍 BRG 模式 (BRGH = 0)。

## 寄存器 18-1: UxMODE: UARTx 模式寄存器 (续)

- 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 UxRX 引脚; 在出现下降沿时产生中断, 在随后的上升沿该位由硬件清零  
0 = 未使能唤醒
- 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 = 使能在下一个字符传输过程中对波特率进行测量 —— 需要接收到同步字段 (55h); 完成时由硬件清零  
0 = 禁止或已完成波特率测量
- bit 4      **URXINV:** UARTx 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 (每个位周期内产生 4 个 BRG 时钟信号)  
0 = 标准模式 (每个位周期内产生 16 个 BRG 时钟信号)
- 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`, 外设的输入和输出必须配置给可用的 `RPn/RPIn` 引脚。更多信息, 请参见第 11.4 节“[外设引脚选择 \(PPS\)](#)”。
- 2: 此功能仅适用于 16 倍 BRG 模式 (`BRGH = 0`)。

# PIC24FJ128GB204 系列

寄存器 18-2: UxSTA: UARTx 状态和控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0, HC	R/W-0	R-0, HSC	R-1, HSC
UTXISEL1	UTXINV <sup>(1)</sup>	UTXISEL0	URXEN	UTXBRK	UTXEN <sup>(2)</sup>	UTXBF	TRMT
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	R-1, HSC	R-0, HSC	R-0, HSC	R/C-0, HS	R-0, HSC
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7						bit 0	

<b>图注:</b>	C = 可清零位	HSC = 可硬件置 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
HS = 可由硬件置 1 的位	HC = 可由硬件清零的位	x = 未知

- bit 15,13 **UTXISEL<1:0>**: UARTx 发送中断模式选择位  
 11 = 保留; 不要使用  
 10 = 当一个字符被传送到发送移位寄存器 (TSR) 导致发送缓冲区为空时, 产生中断  
 01 = 当最后一个字符被移出发送移位寄存器时产生中断; 发送操作全部完成  
 00 = 当一个字符被传送到发送移位寄存器 (前提是发送缓冲区中至少还有一个字符可供发送) 时产生中断
- bit 14 **UTXINV**: IrDA<sup>®</sup> 编码器发送极性翻转位 <sup>(1)</sup>  
**IREN = 0:**  
 1 = UxTX 空闲状态为 0  
 0 = UxTX 空闲状态为 1  
**IREN = 1:**  
 1 = UxTX 空闲状态为 1  
 0 = UxTX 空闲状态为 0
- bit 12 **URXEN**: UARTx 接收使能位  
 1 = 使能接收, UARTx 控制 UxRX 引脚  
 0 = 禁止接收, 端口控制 UxRX 引脚
- bit 11 **UTXBRK**: UARTx 发送间隔字符位  
 1 = 在下次发送时发出同步间隔字符 —— 起始位, 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**: UARTx 发送使能位 <sup>(2)</sup>  
 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, 缓冲区复位; 端口控制 UxTX 引脚
- bit 9 **UTXBF**: UARTx 发送缓冲区满状态位 (只读)  
 1 = 发送缓冲区满  
 0 = 发送缓冲区未满, 至少还可写入一个字符
- bit 8 **TRMT**: 发送移位寄存器空位 (只读)  
 1 = 发送移位寄存器为空, 同时发送缓冲区为空 (上一次发送已完成)  
 0 = 发送移位寄存器非空, 发送正在进行或等待发送

注 1: 仅当使能了 IrDA<sup>®</sup> 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, 外设的输入和输出必须配置给可用的 RPN/RPI 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。

## 寄存器 18-2: UxSTA: UARTx 状态和控制寄存器 (续)

bit 7-6	<b>URXISEL&lt;1:0&gt;</b> : UARTx 接收中断模式选择位 11 = 当 RSR 传输使接收缓冲区为满 (即, 有 4 个数据字符) 时, 中断标志位置 1 10 = 当 RSR 传输使接收缓冲区为 3/4 满 (即, 有 3 个数据字符) 时, 中断标志位置 1 0x = 当接收缓冲区接收到来自 RSR 的任何字符时, 中断标志位置 1。接收缓冲区有一个或多个字符
bit 5	<b>ADDEN</b> : 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 (如果未选择 9 位模式, 那么该配置不起作用) 0 = 禁止地址检测模式
bit 4	<b>RIDLE</b> : 接收器空闲位 (只读) 1 = 接收器空闲 0 = 接收器工作
bit 3	<b>PERR</b> : 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 = 没有检测到奇偶校验错误
bit 2	<b>FERR</b> : 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 = 没有检测到帧错误
bit 1	<b>OERR</b> : 接收缓冲区溢出错误状态位 (只读 / 清零) 1 = 接收缓冲区已经溢出 0 = 接收缓冲区没有溢出。清零先前置 1 的 OERR 位 (1 → 0 的跳变) 会使接收缓冲区和 RSR 复位为空状态
bit 0	<b>URXDA</b> : UARTx 接收缓冲区中数据可用位 (只读) 1 = 接收缓冲区中有数据, 有至少一个字符可被读取 0 = 接收缓冲区为空

- 注 1:** 仅当使能了 IrDA<sup>®</sup> 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, 外设的输入和输出必须配置给可用的 RPn/RPIn 引脚。更多信息, 请参见第 11.4 节“**外设引脚选择 (PPS)**”。

# PIC24FJ128GB204 系列

**寄存器 18-3: UxTXREG: UARTx 发送寄存器 (通常为只写)**

W-x	U-0	U-0	U-0	U-0	U-0	U-0	W-x
LAST <sup>(1)</sup>	—	—	—	—	—	—	UxTXREG8
bit 15						bit 8	

W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
UxTXREG<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            **LAST:** 智能卡支持的最后一个字节指示符位 <sup>(1)</sup>
- bit 14-9        **未实现:** 读为 0
- bit 8            **UxTXREG8:** UARTx 发送字符数据位 (9 位模式下)
- bit 7-0         **UxTXREG<7:0>:** UARTx 发送字符数据位

注 1: 该位仅用于 UART1 和 UART2。

**寄存器 18-4: UxADMD: UARTx 地址匹配检测寄存器**

R/W-0							
ADMMASK7	ADMMASK6	ADMMASK5	ADMMASK4	ADMMASK3	ADMMASK2	ADMMASK1	ADMMASK0
bit 15							bit 8

R/W-0							
ADMADDR7	ADMADDR6	ADMADDR5	ADMADDR4	ADMADDR3	ADMADDR2	ADMADDR1	ADMADDR0
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        **ADMMASK<7:0>:** UARTx ADMADDR<7:0> (UxADMD<7:0>) 掩码位  
 对于 ADMMASK<x>:  
 1 = ADMADDR<x> 用来检测地址匹配  
 0 = ADMADDR<x> 不用来检测地址匹配
- bit 7-0         **ADMADDR<7:0>:** UARTx 地址检测任务免除位  
 与 ADMMASK<7:0> 位 (UxADMD<15:8>) 配合, 在地址检测模式下免去处理器检测地址字符的任务。

# PIC24FJ128GB204 系列

寄存器 18-5: UxSCCON: UARTx 智能卡控制寄存器<sup>(1)</sup>

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
—	—	TXRPT1 <sup>(2)</sup>	TXRPT0 <sup>(2)</sup>	CONV	T0PD <sup>(2)</sup>	PTRCL	SCEN
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 5-4      **TXRPT<1:0>:** 重复发送选择位<sup>(2)</sup>  
 11 = 重新发送错误字节 4 次  
 10 = 重新发送错误字节 3 次  
 01 = 重新发送错误字节 2 次  
 00 = 重新发送错误字节 1 次
- bit 3        **CONV:** 逻辑转换选择位  
 1 = 反相逻辑转换  
 0 = 直接逻辑转换
- bit 2        **T0PD:** T = 0 错误处理的下拉持续时间位<sup>(2)</sup>  
 1 = 2 ETU  
 0 = 1 ETU
- bit 1        **PTRCL:** 智能卡协议选择位  
 1 = T = 1  
 0 = T = 0
- bit 0        **SCEN:** 智能卡模式使能位  
 1 = 如果 UARTEN (UxMODE<15>) = 1, 使能智能卡模式  
 0 = 禁止智能卡模式

- 注**    **1:** 该寄存器仅用于 UART1 和 UART2。  
**2:** 这些位仅适应于 T = 0, 请参见 PTRCL (UxSCCON<1>)。

# PIC24FJ128GB204 系列

寄存器 18-6: UxSCINT: UARTx 智能卡中断寄存器<sup>(1)</sup>

U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	—	RXRPTIF <sup>(2)</sup>	TXRPTIF <sup>(2)</sup>	—	—	WTCIF	GTCIF
bit 15						bit 8	

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	PARIE <sup>(2)</sup>	RXRPTIE <sup>(2)</sup>	TXRPTIE <sup>(2)</sup>	—	—	WTCIE	GTCIE
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        **RXRPTIF:** 重复接收中断标志位<sup>(2)</sup>  
 1 = 接收同一字符 5 次 (4 次重新发送) 后, 奇偶校验错误仍存在  
 0 = 清零标志
- bit 12        **TXRPTIF:** 重复发送中断标志位<sup>(2)</sup>  
 1 = 每个 TXRPT<1:0> 设置的最后一次重新发送后检测到线路错误 (见寄存器 18-5)  
 0 = 清零标志
- bit 11-10     **未实现:** 读为 0
- bit 9         **WTCIF:** 等待时间计数器中断标志位  
 1 = 等待时间计数器到达 0  
 0 = 等待时间计数器未到达 0
- bit 8         **GTCIF:** 保护时间计数器中断标志位  
 1 = 保护时间计数器到达 0  
 0 = 保护时间计数器未到达 0
- bit 7         **未实现:** 读为 0
- bit 6         **PARIE:** 奇偶校验中断允许位<sup>(2)</sup>  
 1 = 接收到具有奇偶校验错误的字符时产生中断 (见寄存器 18-2 中的 PERR (UxSTA<3>) 了解中断标志)  
 0 = 禁止中断
- bit 5         **RXRPTIE:** 重复接收中断允许位<sup>(2)</sup>  
 1 = 接收同一字符 5 次 (4 次重新发送) 后, 奇偶校验错误仍存在时产生中断  
 0 = 禁止中断
- bit 4         **TXRPTIE:** 重复发送中断允许位<sup>(2)</sup>  
 1 = 每个 TXRPT<1:0> 设置的最后一次重新发送完成后检测到线路错误时产生中断 (见寄存器 18-5)  
 0 = 禁止中断
- bit 3-2       **未实现:** 读为 0
- bit 1         **WTCIE:** 等待时间计数器中断允许位  
 1 = 允许等待时间计数器中断  
 0 = 禁止等待时间计数器中断
- bit 0         **GTCIE:** 保护时间计数器中断允许位  
 1 = 允许保护时间计数器中断  
 0 = 禁止保护时间计数器中断

注 1: 该寄存器仅用于 UART1 和 UART2。  
 2: 这些位仅适用于 T = 0, 请参见 PTRCL (UxSCCON<1>)。

## 19.0 带 ON-THE-GO 支持的通用串行总线 (USB OTG)

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“USB On-The-Go (OTG)” (DS39721)。本数据手册中的信息取代了FRM中的信息。

PIC24FJ128GB204 系列器件包含一个兼有全速和低速两种工作模式的 On-The-Go (OTG) USB 串行接口引擎 (Serial Interface Engine, SIE)。OTG 功能使得器件可以用作 USB 外设或具有有限主机功能的 USB 嵌入式主机。OTG 功能还可使器件通过 OTG 的主机协商协议 (Host Negotiation Protocol, HNP) 从设备操作动态地切换到主机操作。

更多关于 OTG 操作的详细信息，请参见 USB-IF 发布的“USB 2.0 规范”的“*On-The-Go Supplement*” (On-The-Go 补充信息)。更多关于 USB 操作的详细信息，请参见“*Universal Serial Bus Specification*” (通用串行总线规范) v2.0。

USB OTG 模块提供以下特性：

- 设备和主机模式下的 USB 功能和应用控制模式下的 OTG 功能切换
- 软件可选的模块速度：全速 (12 Mbps) 或低速 (1.5 Mbps，仅在主机模式下可用)
- 支持所有四种 USB 传输类型：控制、中断、批量和同步
- 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
- 集成的片内 USB 收发器
- 使用片内比较器的集成 Vbus 生成，及升压生成
- 片上总线上拉和下拉电阻配置

USB OTG 模块的简化框图如图 19-1 所示。

USB OTG 模块可用作 USB 外设或 USB 主机，并可在软件控制下在设备和主机模式间动态切换。两种模式使用同一数据路径和缓冲区描述符 (Buffer Descriptor, BD) 来收发数据。

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和 USB 之间的数据传输方向。RX (接收) 用于描述数据从 USB 传送到单片机，TX (发送) 用于描述数据从单片机传送到 USB。表 19-1 显示了这种命名法的数据方向和 USB 令牌交换之间的关系。

**表 19-1: USB 主机或目标设备以控制器为中心的数据方向**

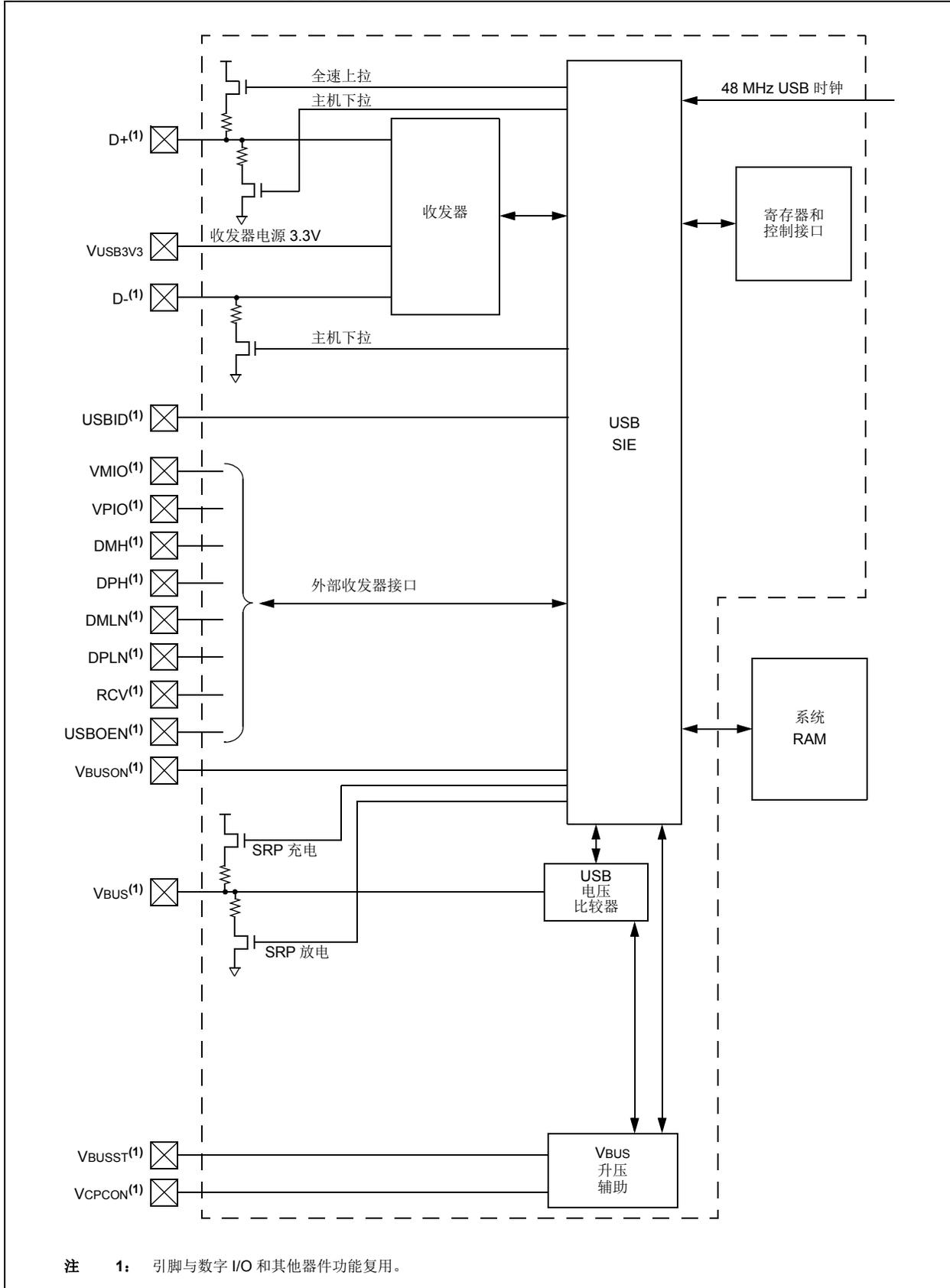
USB 模式	方向	
	RX	TX
设备	输出或设置	输入
主机	输入	输出或设置

本章将介绍在应用中实现 USB OTG 功能所需的最基本操作。对 USB 协议及其 OTG 补充信息的完整详细说明不在本数据手册的讨论范围内。本数据手册建立在用户对 USB 架构以及协议最新版本已有了基本了解的基础之上。

这里并没有给出正确 USB 操作 (例如设备枚举) 所需的全部步骤。建议应用开发人员使用适当的设备驱动程序来实现所有的必需功能。Microchip 提供了许多特定于应用的资源，例如 USB 固件和驱动程序支持。可访问 [www.microchip.com/usb](http://www.microchip.com/usb) 以获取最新的固件和驱动程序支持。

# PIC24FJ128GB204 系列

图 19-1: USB OTG 模块框图



## 19.1 硬件配置

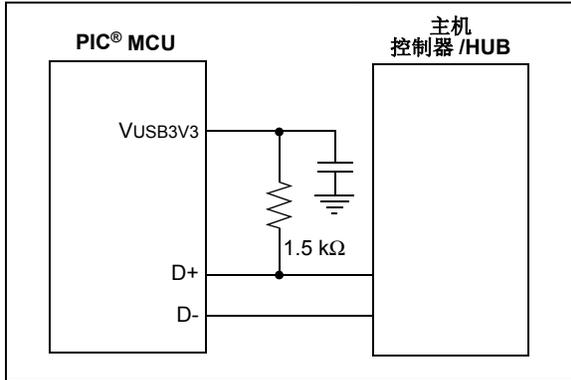
### 19.1.1 设备模式

#### 19.1.1.1 D+ 上拉电阻

PIC24FJ128GB204 系列器件在 D+ 线上有一个内置的 1.5 kΩ 电阻，当单片机工作在设备模式下时，D+ 线可用。这用于以信号形式告知外部主机：器件正工作在全速设备模式下。通过置 1 USBEN 位 (U1CON<0>) 使能该模式。如果 OTGEN 位 (U1OTGCON<2>) 置 1，则通过 DPPULUP 位 (U1OTGCON<7>) 使能 D+ 上拉。

另外，可在 D+ 线上使用外部电阻，如图 19-2 所示。

图 19-2: 全速设备模式下的外部上拉



#### 19.1.1.2 电源模式

许多 USB 应用可能有多组不同的电源要求和配置。最常遇到的电源模式为：

- 仅总线电源模式
- 仅自供电模式
- 双电源且自供电为主模式

实际上，仅总线电源模式（图 19-3）是最简单的方法。应用的所有电源均来自 USB。

为了满足“USB 2.0 Specification”的涌流要求，V<sub>BUS</sub> 和地之间的总有效电容不得大于 10 μF。

在 USB 暂挂模式下，设备从 USB 电缆的 5V V<sub>BUS</sub> 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 19-4）下，USB 应用有自己的电源，由 USB 提供的电源极小。注意，将添加一个连接指示，表明 USB 已连接，主机正在提供 V<sub>BUS</sub> 电源。

要满足规范要求，在主机将 V<sub>BUS</sub> 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）。使用一个能承受 5.5V 电压的 I/O 引脚来实现此目的。

应用不应向 USB 电缆的 5V V<sub>BUS</sub> 引脚汲取任何电流。

双电源且自供电为主的模式（图 19-5）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 V<sub>BUS</sub> 驱动为高电平之前不得使能 USB 模块。

图 19-3: 仅总线电源

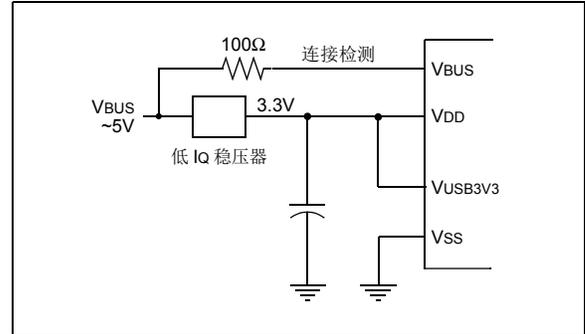


图 19-4: 仅自供电

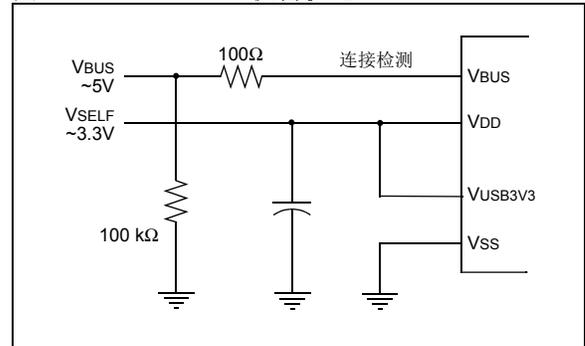
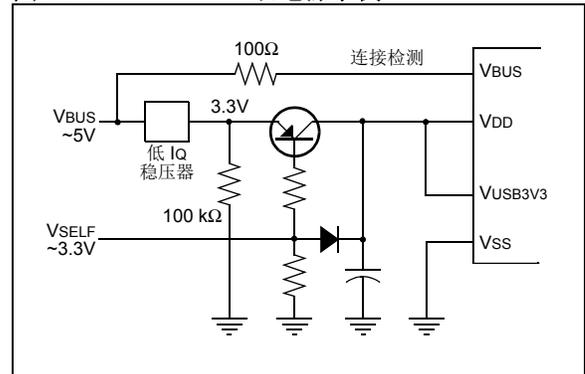


图 19-5: 双电源示例



# PIC24FJ128GB204 系列

## 19.1.2 主机和 OTG 模式

### 19.1.2.1 D+ 和 D- 下拉电阻

PIC24FJ128GB204 系列器件在 D+ 和 D- 线上具有内置的 15 kΩ 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。通过置 1 HOSTEN 位 (U1CON<3>) 使能该模式。如果 OTGEN 位 (U1OTGCON<2>) 置 1，那么将 DPPULDWN 和 DMPULDWN 位 (U1OTGCON<5:4>) 置 1 可使能这两个下拉电阻。

### 19.1.2.2 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”要求主机应用提供 VBUS 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。

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提供 VBUS 并调节总线上的电流 (图 19-6)。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 19-7 给出了使用外部电荷泵的典型示例。

图 19-6: USB 主机接口示例

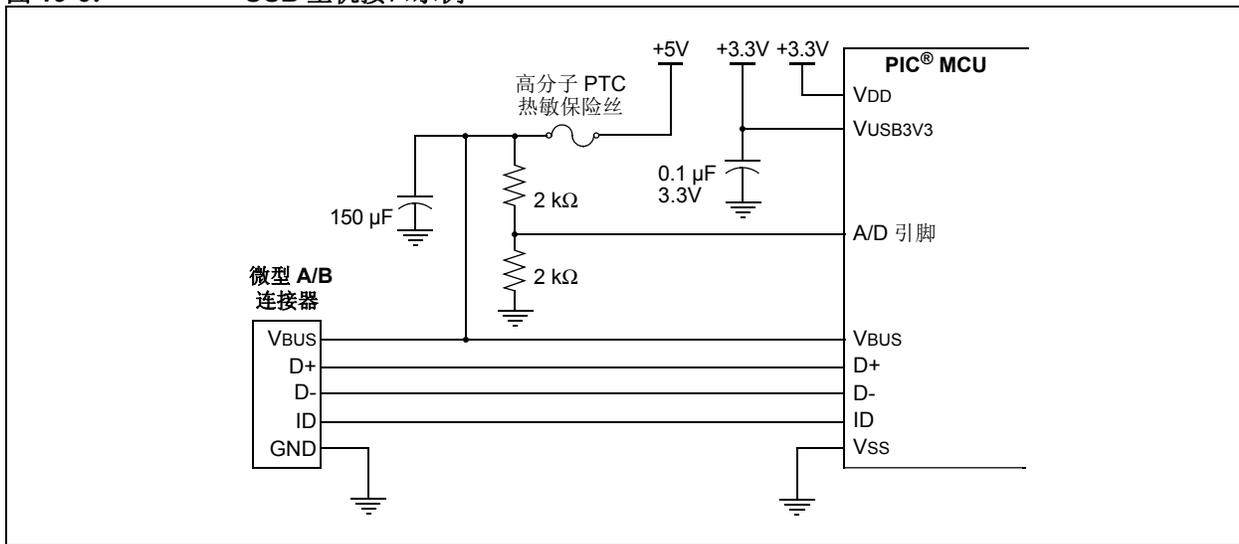
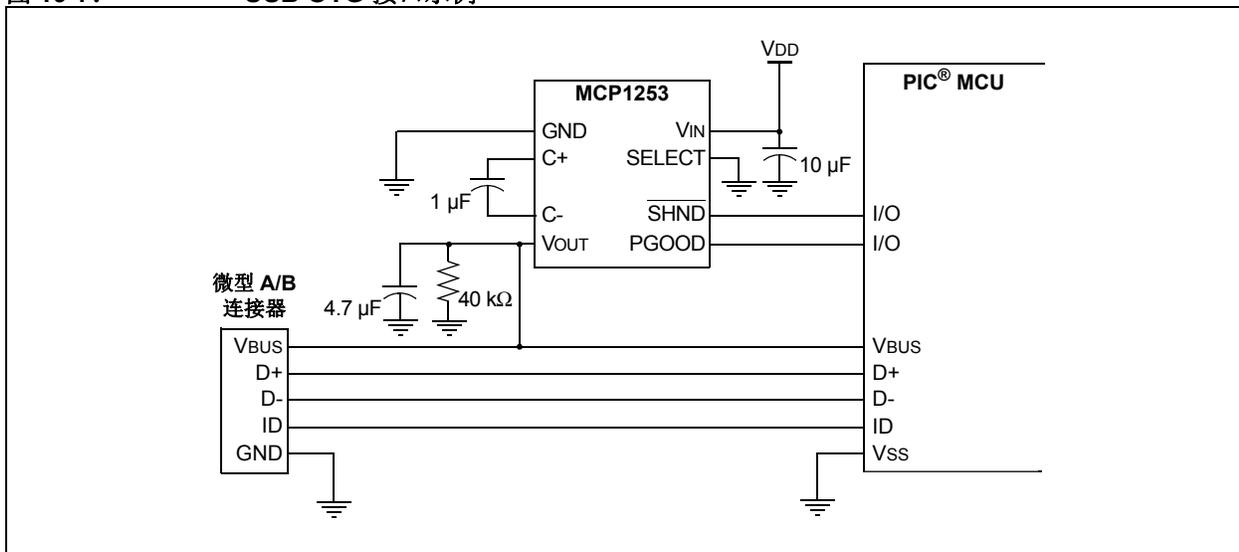


图 19-7: USB OTG 接口示例



## 19.1.2.3 使用外部器件生成 VBus 电压

作为 USB 主机（作为 OTG 配置中的 A 设备或嵌入式主机）工作时，必须为所连接的器件提供 VBus。PIC24FJ128GB204 系列器件具有内部 VBus 升压辅助工具，可帮助通过电路板上提供的电压生成所需的 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，请执行以下操作：

1. 确认 USB 模块已上电（U1PWRC<0> = 1）并且已禁止 VBus 放电（U1OTGCON<0> = 0）。
2. 使用 VBUSCHG 位（U1OTGCON<1>）选择所需的目标电压。
3. 使能 VBus 生成电路（U1OTGCON<3> = 1）。

**注：** 本节描述了 VBus 电压生成和控制的一般步骤。如需更多示例，请参见《dsPIC33/PIC24 系列参考手册》。

## 19.1.3 计算收发器功耗要求

USB 收发器消耗的电流取决于 USB 电缆的特性阻抗、电缆的长度、VUSB3V3 供电电压和 USB 电缆上的实际数据传输模式。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 19-1 可帮助估算在全速应用中实际需要的电流。

关于收发器功耗的全面讨论，请参见《dsPIC33/PIC24 系列参考手册》中的“USB On-The-Go (OTG)” (DS39721)。

### 公式 19-1: 估算 USB 收发器电流消耗

$$IXCVR = \frac{40 \text{ mA} \cdot V_{USB3V3} \cdot P_{ZERO} \cdot P_{IN} \cdot L_{CABLE}}{3.3V \cdot 5m} + IPULLUP$$

**图注：**

- $V_{USB3V3}$ ——施加到 VUSB3V3 引脚的电压（单位为 V，3.0V 至 3.6V）。
- $P_{ZERO}$ ——PIC® 单片机发送的 IN（输入）通信中值为 0 的位所占的百分比（单位为 %）。
- $P_{IN}$ ——用于 IN 通信的总线带宽占总线总带宽的百分比（单位为 %）。
- $L_{CABLE}$ ——USB 电缆的长度（单位为 m）。“USB 2.0 规范”要求全速应用使用的电缆不应长于 5m。
- $IPULLUP$ ——1.5 kΩ 的标称上拉电阻（使能时）必须为 USB 电缆提供的电流。

# PIC24FJ128GB204 系列

## 19.2 USB 缓冲区描述符和 BDT

可通过一个称为缓冲区描述符表 (Buffer Descriptor Table, BDT) 的结构来控制端点缓冲区。用户可使用这个方法灵活地构造和控制各种长度以及各种配置的端点缓冲区。

BDT 可位于数据 RAM 中任何可用的 512 字节对齐的块中。BDT 指针 (U1BDTP1) 包含 BDT 的高地址字节, 用于设置 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 由定义和控制 USB RAM 空间中的实际缓冲区的缓冲区描述符 (BD) 组成。每个 BD 包含两个 16 位的“软” (地址不固定) 寄存器 BDNSTAT 和 BDNADR, 其中 n 表示可能的 64 个 BD (范围为 0 至 63) 中的一个。BDNSTAT 是 BDN 的状态寄存器, 而 BDNADR 指定与 BDN 相关的缓冲区的起始地址。

**注:** 由于 BDNADR 是 16 位寄存器, 仅可通过 USB 模块访问 RAM 的第一个 64 KB。

根据使用的端点缓冲配置, 对于总共 256 字节的 BDT 空间, 最多有 64 组缓冲区描述符。根据最低配置的要求, BDT 必须至少是 8 个字节长。这是因为“USB 2.0 规范”强制每个器件必须提供具有输入 / 输出功能的端点 0, 以确保正确的初始化设置。

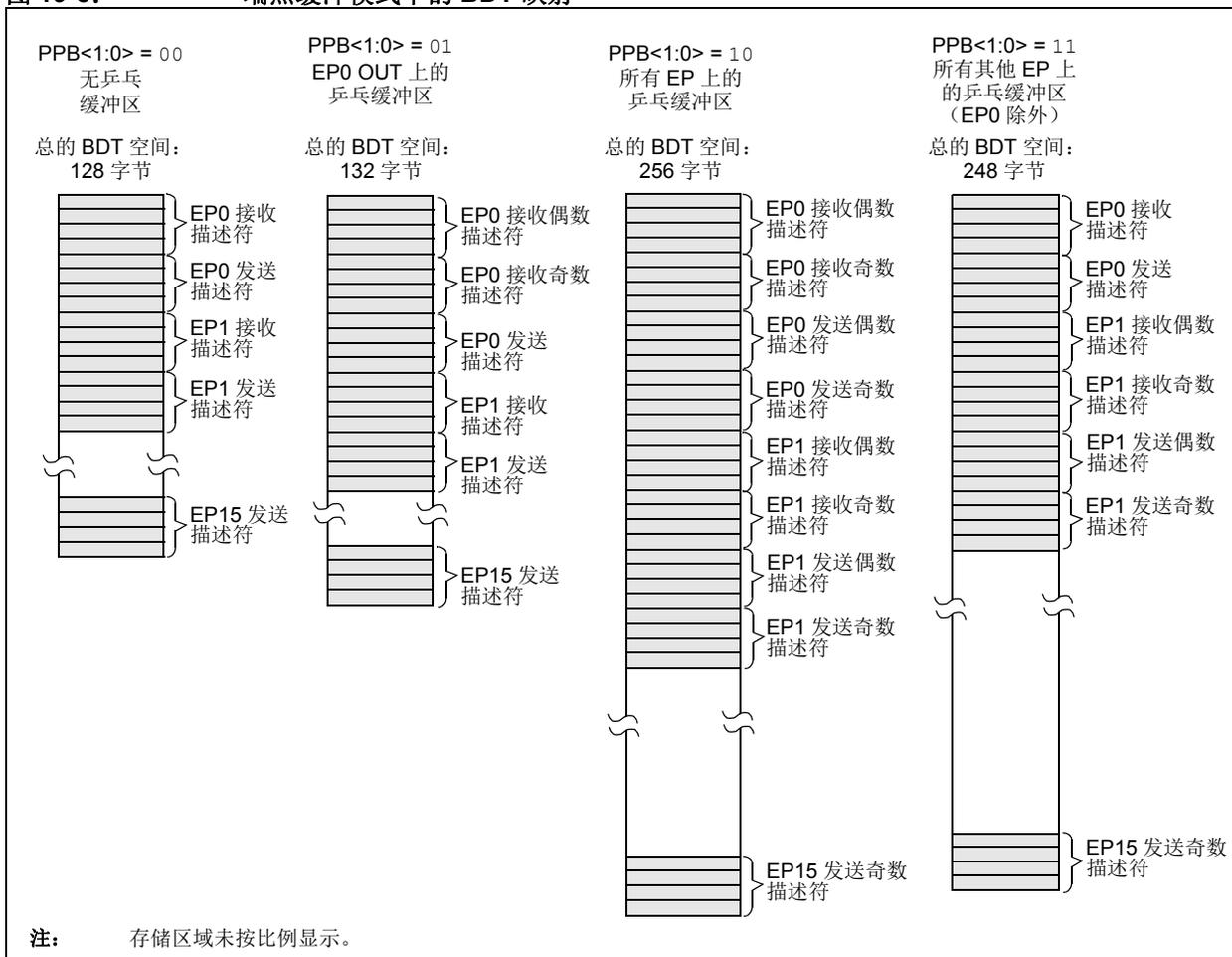
BDT 中的端点映射取决于三个变量:

- 端点号 (0 至 15)
- 端点方向 (RX 或 TX)
- 乒乓设置 (U1CNFG1<1:0>)

图 19-8 演示了如何使用这些变量映射 BDT 中的端点。

在主机模式下, 仅使用了端点 0 的缓冲区描述符。所有传输操作都使用了端点 0 的缓冲区描述符和 USB 端点控制寄存器 (U1EP0)。接收数据包时, 可通过 USB 状态寄存器中的 ENDPT<3:0> (U1STAT<7:4>) 的值指示所连接设备的源端点。发送数据包时, 可通过写入 USB 令牌寄存器 (U1TOK) 中的值指示所连接设备的目标端点。

图 19-8: 端点缓冲模式下的 BDT 映射



BD 与特定端点之间具有固定关系，具体取决于缓冲区配置。表 19-2 提供了 BD 至端点的映射。这种关系还意味着当未连续使能端点时，BDT 中可能会存在间隙。理论上，这意味着禁止端点的 BD 可用作缓冲空间。实际操作中，用户应避免使用 BDT 中的此类空间，除非执行了验证 BD 地址的方法。

## 19.2.1 缓冲区所有权

由于缓冲区及其 BD 是由 CPU 和 USB 模块共用的，所以可将 UOWN 位用作简单的信号机制，用以区分允许哪个模块更新 BD 及其相关的缓冲存储区。UOWN 是 BDnSTAT 的两种配置之间唯一共用的位。

清零 UOWN 时，BD 条目归单片机内核“所有”。UOWN 位置 1 时，BD 条目和缓冲存储区归 USB 外设“所有”。在此期间内，内核不应修改 BD 或其相应的数据缓冲区。

注意，SIE 拥有缓冲区时，单片机内核仍可读取 BDnSTAT，反之亦然。

根据寄存器更新源的不同，缓冲区描述符具有不同的含义。寄存器 19-1 和寄存器 19-2 显示了 BDnSTAT 当前“所有权”不同时的差异。

当 UOWN 置 1 时，用户不再依赖于写入 BD 的值。从此时起，USB 模块只在需要时更新 BD，改写原来的 BD 值。SIE 使用令牌 PID 更新 BDnSTAT 寄存器，同时也会更新传输计数。

## 19.2.2 DMA 接口

USB OTG 模块使用专用的 DMA 访问 BDT 和端点数据缓冲区。因为 DMA 的部分地址空间专用于缓冲区描述符，所以与 DMA 连接的存储器的一部分必须是正确映射的连续地址空间，以供模块访问。

表 19-2: 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 数							
	模式 0 (无乒乓缓冲区)		模式 1 (EPO 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, EPO 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

# PIC24FJ128GB204 系列

寄存器 19-1: **BDnSTAT: 缓冲区描述符 n 状态寄存器原型, USB 模式**  
(BD0STAT 至 BD63STAT)

R/W-x	R/W-x	R/W-x, HSC					
UOWN	DTS	PID3	PID2	PID1	PID0	BC9	BC8
bit 15						bit 8	

R/W-x, HSC							
BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
bit 7						bit 0	

<b>图注:</b>	HSC = 可硬件置 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      **UOWN: USB 所有权位**  
1 = USB 模块拥有 BD 及其对应的缓冲区; CPU 不得修改 BD 或缓冲区
- 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>: 数据包标识位 (由 USB 模块写入)**  
在设备模式下:  
表示上一次传输期间接收到的令牌的 PID。  
在主机模式下:  
表示上一次返回的 PID 或传输状态指示。
- bit 9-0     **BC<9:0>: 字节计数位**  
这表示传输期间要发送的字节数或要接收的最大字节数。完成后, 由 USB 模块使用实际发送或接收的字节数更新该字节计数。

# PIC24FJ128GB204 系列

寄存器 19-2: **BDnSTAT: 缓冲区描述符 n 状态寄存器原型, CPU 模式**  
(BD0STAT 至 BD63STAT)

R/W-x	R/W-x	r-0	r-0	R/W-x	R/W-x	R/W-x, HSC	R/W-x, HSC
UOWN	DTS <sup>(1)</sup>	—	—	DTSEN	BSTALL	BC9	BC8
bit 15						bit 8	

R/W-x, HSC							
BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
bit 7						bit 0	

<b>图注:</b>	r = 保留位	HSC = 可硬件置 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	r = 保留位                      x = 未知

- bit 15      **UOWN:** USB 所有权位  
0 = 单片机内核拥有 BD 及其相应的缓冲区; 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 <sup>(1)</sup>  
1 = 数据 1 数据包  
0 = 数据 0 数据包
- bit 13-12   **保留:** 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; 忽略具有错误同步值的数据包  
0 = 不执行数据翻转同步
- bit 10      **BSTALL:** 缓冲区停止使能位  
1 = 使能缓冲区停止; 如果接收到使用给定存储单元 BD 的令牌, 发出 STALL 握手 (UOWN 位保持置 1, BD 值不变); 任何 STALL 握手都将导致相应的 EPSTALL 位置 1  
0 = 禁止缓冲区停止
- bit 9-0     **BC<9:0>:** 字节计数位  
这表示传输期间要发送的字节数或要接收的最大字节数。完成后, 由 USB 模块使用实际发送或接收的字节数更新该字节计数。

**注 1:** 除非 DTSEN = 1, 否则该位将被忽略。

# PIC24FJ128GB204 系列

## 19.3 USB 中断

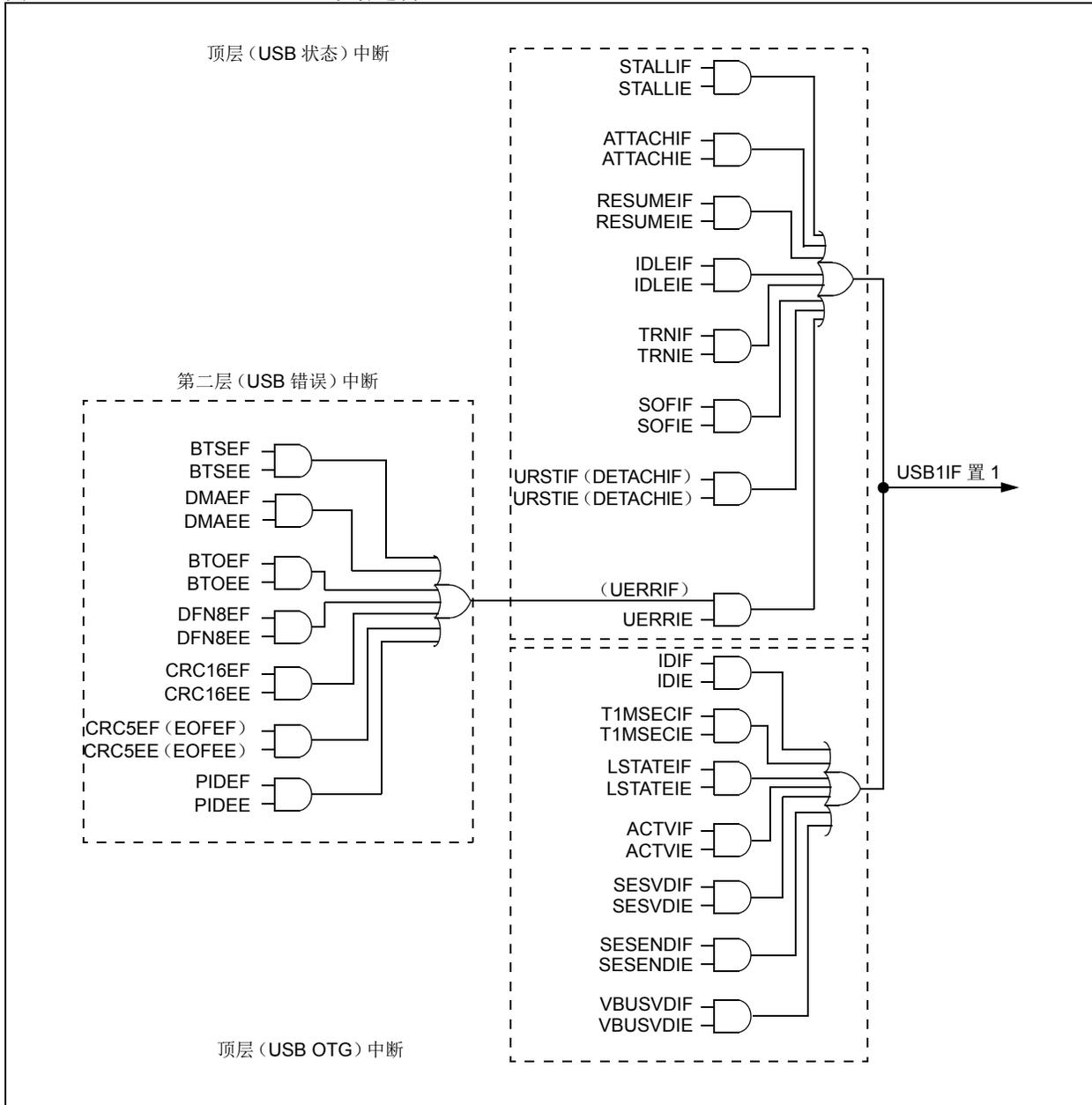
USB OTG 模块在许多条件下可以配置为产生中断。所有中断源都使用同一中断向量。

图 19-9 给出了 USB 模块的中断逻辑。在 USB 模块中有两层中断寄存器。顶层由所有 USB 状态中断组成；这些中断分别在 U1IE 和 U1IR 寄存器中被允许和标记。

第二层由 USB 错误条件组成，它们在 U1EIE 寄存器和 U1EIR 寄存器中被允许和标记。任意一个中断条件都可以触发顶层的 USB 错误中断标志 (UERRIF)。

中断可用于在 USB 事务中产生陷阱。图 19-10 提供了 USB 帧中的一些常见事件及其相应中断。

图 19-9: USB OTG 中断逻辑



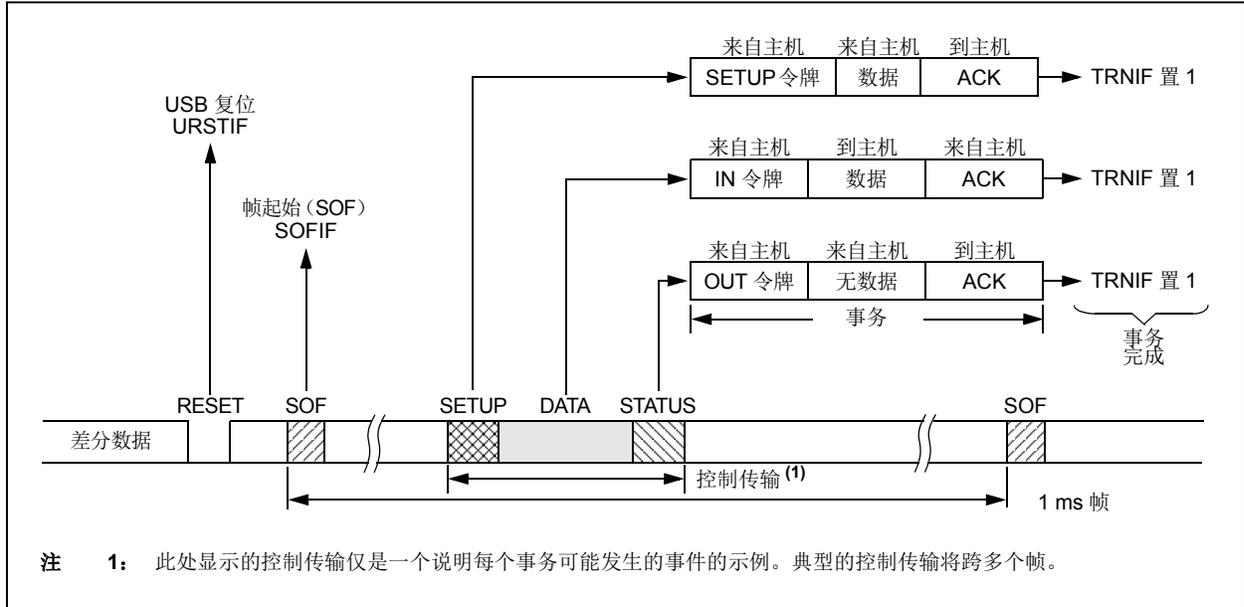
## 19.3.1 清除 USB OTG 中断

与器件级别的中断不同，USB OTG 中断状态标志不能用软件自由写入。所有 USB OTG 标志位都实现为只能由硬件置 1 位。而且，只能通过用软件写入 1 到其存储

单元（即执行 MOV 类型的指令）来清零这些位。写入 0 到标志位（即 BCLR 指令）的操作无效。

**注：** 在本数据手册中，只能通过写入 1 到其存储单元来进行清零的位被称为“写 1 清除”位。在寄存器描述中，此功能用描述符“K”表示。

图 19-10: USB 事务和中断事件的示例



## 19.4 设备模式的操作

接下来的章节描述如何执行常见的设备模式任务。在设备模式下，在传输层执行 USB 传输。USB 模块自动执行传输的状态阶段。

### 19.4.1 使能设备模式

1. 通过先置 1 后清零乒乓缓冲区复位位 PPBRST (U1CON<1>) 来复位乒乓缓冲区指针。
2. 禁止所有中断 (U1IE 和 U1EIE = 00h)。
3. 通过将 FFh 写入 U1IR 和 U1EIR 来清零任何现有中断标志。
4. 确认 VBUS 已提供 (仅限非 OTG 设备)。

5. 通过将 USBEN 位 (U1CON<0>) 置 1 使能 USB 模块。
6. 通过将 OTGEN 位 (U1OTGCON<2>) 置 1 使能 OTG 工作。
7. 通过将端点 0 的 EPRXEN 和 EPHSHK 位置 1 (U1EP0<3,0> = 1) 使能端点 0 缓冲区接收第一个设置数据包。
8. 通过将 USBPWR 位 (U1PWRC<0>) 置 1 使 USB 模块上电。
9. 通过将 DPPULUP 位 (U1OTGCON<7>) 置 1 使能 D+ 上拉电阻发出连接信号。

## 19.4.2 在设备模式下接收 IN 令牌

1. 按照“USB 2.0 规范”第 9 章中的说明连接到 USB 主机并执行枚举。
2. 创建数据缓冲区，并用要发送到主机的数据进行填充。
3. 在所需端点的适当（偶数或奇数）发送 BD 中：
  - a) 使用正确的数据翻转（DATA0/1）值和数据缓冲区的字节计数设置状态寄存器（BDnSTAT）。
  - b) 使用数据缓冲区的起始地址设置地址寄存器（BDnADR）。
  - c) 将状态寄存器的 UOWN 位置 1。
4. 当 USB 模块接收到 IN 令牌时，它自动发送缓冲区中的数据。完成时，模块会更新状态寄存器（BDnSTAT）并将传输完成中断标志 TRNIF（U1IR<3>）置 1。

## 19.4.3 在设备模式下接收 OUT 令牌

1. 按照“USB 2.0 规范”第 9 章中的说明连接到 USB 主机并执行枚举。
2. 使用期望从主机获得的数据量创建数据缓冲区。
3. 在所需端点的适当（偶数或奇数）发送 BD 中：
  - a) 使用正确的数据翻转（DATA0/1）值和数据缓冲区的字节计数设置状态寄存器（BDnSTAT）。
  - b) 使用数据缓冲区的起始地址设置地址寄存器（BDnADR）。
  - c) 将状态寄存器的 UOWN 位置 1。
4. 当 USB 模块接收到 OUT 令牌时，会自动接收主机发送给缓冲区的数据。完成时，模块会更新状态寄存器（BDnSTAT）并将传输完成中断标志 TRNIF（U1IR<3>）置 1。

## 19.5 主机模式的操作

接下来的章节描述如何执行常见的主机模式任务。在主机模式下，USB 传输由主机软件明确调用。主机软件负责传输的应答部分。而且，所有传输都是使用 USB 端点 0 控制寄存器（U1EP0）和缓冲区描述符执行的。

### 19.5.1 使能主机模式并发现连接的设备

1. 通过将 HOSTEN 位（U1CON<3>）置 1 使能主机模式。这使得其他 USB OTG 寄存器中的主机模式控制位可用。
2. 通过置 1 DPPULDWN 和 DMPULDWN 位（U1OTGCON<5:4>）使能 D+ 和 D- 下拉电阻。通过清零 DPPULUP 和 DMPULUP 位（U1OTGCON<7:6>）禁止 D+ 和 D- 上拉电阻。
3. 此时，SOF 计数器装入 12,000 开始帧起始（SOF）生成。通过清零 SOFEN 位（U1CON<0>）禁止帧起始数据包生成，从而消除 USB 上的噪声。
4. 通过将 ATTACHIE（U1IE<6>）置 1 允许所连接设备的中断。
5. 等待所连接设备的中断（U1IR<6> = 1）。这可通过发出以下信号来实现：USB 设备将 D+ 或 D- 的状态从 0 更改为 1（SE0 变为 J 状态）。在此之后，等待 100 ms 的时间，以使设备电源稳定下来。
6. 检查 U1CON 中 JSTATE 和 SE0 位的状态。如果 JSTATE 位（U1CON<7>）为 0，则连接的设备为低速，此时，将 LSPDEN 和 LSPD 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，来复位 USB 设备。50 ms 后，通过清零 USBRST 终止复位。
8. 要避免所连接设备进入暂挂状态，可通过将 SOFEN 位置 1 使能 SOF 数据包生成。
9. 等待 10 ms，使设备从复位中恢复。
10. 按“USB 2.0 规范”第 9 章中的说明执行枚举。

## 19.5.2 完成对所连接设备的控制事务

- 按照第 19.5.1 节“使能主机模式并发现连接的设备”中描述的步骤发现设备。
- 通过将 0Dh 写入 U1EP0（这会将 EPCONDIS、EPTXEN 和 EPHSHK 位置 1）将端点控制寄存器设置为双向控制传输。
- 在存储缓冲区中存储设备框架设置命令的副本。关于设备框架命令集的信息，请参见“USB 2.0 规范”第 9 章。
- 初始化当前（偶数或奇数）发送 EP0 的缓冲区描述符（BD），以传输设备框架命令的 8 字节命令数据（即 GET DEVICE DESCRIPTOR）：
  - 将 BD 数据缓冲区地址（BD0ADR）设置为包含此命令的 8 字节存储缓冲区的起始地址。
  - 将 8008h 写入 BD0STAT（这会将 UOWN 位置 1 并将字节计数设置为 8）。
- 在 USB 地址寄存器（U1ADDR<6:0>）中设置目标设备的 USB 设备地址。在 USB 总线复位后，设备 USB 地址将为零。在枚举操作后，它将设置为一个 1 至 127 之间的值。
- 将 D0h 写入 U1TOK；这是一个到目标设备的默认控制管道端点 0 的 SETUP 令牌。这将在总线上启动 SETUP 令牌，后跟数据包。在包完成后，将返回设备握手到 BD0STAT 的 PID 字段中。当 USB 模块更新了 BD0STAT 时，将产生一个传输完成中断标志（TRNIF 标志置 1）。这就完成了设置事务的设置阶段，如“USB 2.0 规范”第 9 章所述。
- 要启动设置事务的数据阶段（即获取 GET DEVICE DESCRIPTOR 命令所需数据），请在存储器中设置缓冲区以存储接收到的数据。
- 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。
  - 将 C040h 写入 BD0STAT。这会将 UOWN 置 1、将数据翻转包（DTS）配置为 DATA1，并将字节计数设置为数据缓冲区的长度（这里是 64 或 40h）。
  - 将 BD0ADR 设置为数据缓冲区的起始地址。
- 用对于目标设备的默认控制管道端点 0 适当的 IN 或 OUT 令牌写 USB 令牌寄存器（例如，对于 GET DEVICE DESCRIPTOR 命令的 IN 令牌，写入 90h 到 U1TOK）。这会在总线上发起 IN 令牌，后跟从设备发送到主机的数据包。当数据包完成时，将写入 BD0STAT 并产生一个传输完成中断（即 TRNIF 标志置 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 2.0 规范”第 9 章所述。如果需要传输更多数据，请返回步骤 8。
- 要启动设置事务的状态阶段，请在存储器中建立一个缓冲区以接收或发送零长度状态阶段数据包。
- 初始化当前（偶数或奇数）发送 EP0 BD 以传输状态数据：
  - 将 BDT 缓冲区地址字段设置为数据缓冲区的起始地址
  - 将 8000h 写入 BD0STAT（置 1 UOWN 位、将 DTS 配置为 DATA0 并将字节计数设置为 0）。
- 用对于目标设备的默认控制管道端点 0 适当的 IN 或 OUT 令牌写 USB 令牌寄存器（例如，对于 GET DEVICE DESCRIPTOR 命令的 OUT 令牌，写入 01h 到 U1TOK）。这会在总线上发起 OUT 令牌，后跟从主机发送到设备的零长度数据包。当数据包完成时，将来自设备的握手更新 BD，并产生一个传输完成中断（即 TRNIF 标志置 1）。这就完成了设置事务的状态阶段，如“USB 2.0 规范”第 9 章所述。

**注：** 对于每个帧，只能执行一个控制事务。

# PIC24FJ128GB204 系列

## 19.5.3 向目标设备发送全速批量数据传输

1. 按照第 19.5.1 节“使能主机模式并发现连接的设备”和第 19.5.2 节“完成对所连接设备的控制事务”中描述的步骤发现和配置设备。
2. 要在使能握手时使能发送和接收传输，请将 1Dh 写入 U1EP0。如果目标设备为低速设备，还需要将 LSPD 位 (U1EP0<7>) 置 1。如果希望硬件在目标设备在传输中产生 NAK 时无限期自动重试，请清零重试禁止位 RETRYDIS (U1EP0<6>)。
3. 设置当前 (偶数或奇数) 发送 EP0 的 BD，以传输最多 64 字节。
4. 在 USB 地址寄存器 (U1ADDR<6:0>) 中设置目标设备的 USB 设备地址。
5. 将到目标端点的 OUT 令牌写入 U1TOK。该操作会触发模块的发送状态机，开始发送令牌和数据。
6. 等待传输完成中断标志 TRNIF。这表示 BD 所有权已释放给单片机且传输完成。如果重试禁止位 (RETRYDIS) 置 1，则返回握手 (ACK、NAK、STALL 或 ERROR (0Fh)) 到 BD PID 字段。如果发生暂停中断，则从队列中除去暂挂数据包，并清除目标设备中的错误条件。如果发生断开连接中断 (保持 SE0 状态 2.5  $\mu$ s 以上)，则目标已断开连接 (U1IR<0> 置 1)。
7. 发生传输完成中断 (TRNIF 置 1) 后，返回步骤 2 检查 BD 并将下一个数据包放入队列。

**注：** 只能在模块设置阶段配置 USB 速度、收发器和上拉。建议不要在模块使能后更改这些设置。

## 19.6 OTG 工作

### 19.6.1 会话请求协议 (SRP)

当 OTG A 设备没有通过会话请求协议 (Session Request Protocol, SRP) 使用 USB 链路时，它可能决定让 VBUS 电源掉电。软件可通过清零 VBUSON (U1OTGCON<3>) 完成此操作。当 VBUS 电源掉电后，说明 A 设备已结束 USB 会话。

OTG A 设备或嵌入式主机可以随时重新让 VBUS 电源上电 (发起新的会话)。OTG B 设备也可以请求 OTG A 设备重新让 VBUS 电源上电 (发起新的会话)。这可以通过会话请求协议 (SRP) 来完成。

在请求新的会话之前，B 设备必须检查上一个会话是否确实已结束。要完成此操作，B 设备必须检查两个条件：

1. VBUS 电源低于会话有效电压。
2. D+ 和 D- 都至少保持低电平状态 2 ms。

SESENDIF (U1OTGIR<2>) 中断会向 B 设备通知条件 1。软件将必须手动检查条件 2。

**注：** 当 A 设备使 VBUS 电源掉电时，B 设备必须断开其上拉电阻与电源的连接。如果设备是自供电的，它会通过清零 DPPULUP (U1OTGCON<7>) 和 DMPULUP (U1OTGCON<6>) 来执行此操作。

B 设备可以通过一个电阻让 VBUS 电源放电来协助达到条件 1。用软件通过将 VBUSDIS (U1OTGCON<0>) 置 1 可完成此操作。

满足这些初始条件后，B 设备可开始请求新的会话。B 设备通过在 D+ 数据线上发送脉冲信号开始请求。用软件通过将 DPPULUP (U1OTGCON<7>) 置 1 来完成此操作。数据线应保持高电平 5 至 10 ms。

B 设备随后在 VBUS 电源上发送脉冲信号。用软件通过将 PUVBUS (U1CNFG2<4>) 置 1 来完成此操作。当 A 设备检测到 SRP 信号 (通过 ATTACHIF (U1IR<6>) 中断或通过 SESVDIF (U1OTGIR<3>) 中断) 时，A 设备必须通过将 VBUSON (U1OTGCON<3>) 置 1 或设置控制外部电源的 I/O 端口，来恢复 VBUS 电源。

B 设备在发送 VBUS 电源脉冲信号时不会监视 VBUS 电源的状态。当 B 设备确实检测到 VBUS 电源已恢复 (通过 SESVDIF (U1OTGIR<3>) 中断) 时，它必须通过上拉 D+ 或 D- (通过 DPPULUP 或 DMPULUP 位) 重新连接到 USB 链路。

A 设备必须通过驱动 USB 复位信号来完成 SRP。

## 19.6.2 主机协商协议 (HNP)

在 USB OTG 应用中，双角色设备 (Dual Role Device, DRD) 是能够用作主机或外设的设备。所有 OTG DRD 都必须支持主机协商协议 (HNP)。

HNP 允许 OTG B 设备临时用作 USB 主机。A 设备必须先使能 B 设备，以遵守 HNP。更多关于 HNP 的信息，请参见“USB 2.0 规范”的“*On-The-Go Supplement*”。HNP 只能在全速时启用。

在被 A 设备使能以遵守 HNP 后，B 设备请求在 USB 链路处于暂挂状态 (通过指示断开连接) 时用作主机。用软件通过清零 DPPULUP 和 DMPULUP 可完成此操作。当 A 设备检测到断开连接条件 (通过 URSTIF (U1IR<0>) 中断) 时，A 设备允许 B 设备作为主机接管。A 设备通过发出全速连接信号完成该操作。用软件将 DPPULUP 置 1 实现该操作。

如果 A 设备改为使用恢复信号作为响应，则 A 设备仍用作主机。当 B 设备检测到连接条件 (通过 ATTACHIF) 时，它成为主机。B 设备在使用总线之前驱动复位信号。

当 B 设备完成主机角色后，它停止所有总线活动并通过将 DPPULUP 置 1 来使其 D+ 上拉电阻。当 A 设备检测到暂挂条件 (保持空闲 3 ms) 时，A 设备关闭其 D+ 上拉。A 设备还可让 Vbus 电源掉电以结束会话。当 A 设备检测到连接条件 (通过 ATTACHIF) 时，它恢复主机操作并驱动复位信号。

## 19.7 USB OTG 模块寄存器

共有 37 个存储器映射的寄存器与 USB OTG 模块关联。可划分为四个通用类：

- 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。其原型寄存器 (如寄存器 19-1 和寄存器 19-2 所示) 分别在第 19.2 节“USB 缓冲区描述符和 BDT”中进行了说明。

所有 USB OTG 寄存器都只实现了寄存器的低字节。高字节中的位未实现，不起作用。注意，有些寄存器仅在主机模式下有意义，而其他寄存器在设备和主机模式下具有不同的位含义和功能。

下面章节描述的寄存器包含那些具有特殊控制和配置功能的位。以下寄存器仅用于数据或地址值：

- 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

# PIC24FJ128GB204 系列

## 19.7.1 USB OTG 模块控制寄存器

**寄存器 19-3: U1OTGSTAT: USB OTG 状态寄存器 (仅主机模式)**

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

<b>图注:</b>	U = 未实现位, 读为 0		
R = 可读位	W = 可写位	HSC = 可硬件置 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        **ID:** ID 引脚状态指示位  
1 = 未连接插头或 B 型电缆已插入 USB 插座  
0 = A 型插头已插入 USB 插座
- bit 6        **未实现:** 读为 0
- bit 5        **LSTATE:** 线路状态稳定指示位  
1 = USB 线路状态 (由 SE0 和 JSTATE 定义) 已在上 1 ms 达到稳定  
0 = USB 线路状态在上 1 ms 未达到稳定
- 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 设备上的 VA\_SESS\_VLD (如“USB 2.0 规范”所定义)  
0 = VBUS 电压低于 A 或 B 设备上的 VA\_SESS\_VLD
- bit 2        **SESEND:** B 会话结束指示位  
1 = VBUS 电压低于 B 设备上的 VB\_SESS\_VLD (如“USB 2.0 规范”所定义)  
0 = VBUS 电压高于 B 设备上的 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 设备上的 VA\_VBUS\_VLD (如“USB 2.0 规范”所定义)  
0 = VBUS 电压低于 A 设备上的 VA\_VBUS\_VLD

# PIC24FJ128GB204 系列

**寄存器 19-4: U1OTGCON: USB OTG 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 <sup>(1)</sup>	DMPULDWN <sup>(1)</sup>	VBUSON <sup>(1)</sup>	OTGEN <sup>(1)</sup>	VBUSCHG <sup>(1)</sup>	VBUSDIS <sup>(1)</sup>
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 <sup>(1)</sup>  
               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 <sup>(1)</sup>  
               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 <sup>(1)</sup>  
               1 = VBUS 线路上电  
               0 = VBUS 线路未上电
- bit 2         **OTGEN:** OTG 功能使能位 <sup>(1)</sup>  
               1 = 使能 USB OTG 以及所有 D+/D- 上拉和下拉位  
               0 = 禁止 USB OTG; 通过设置 HOSTEN 和 USBEN 位 (U1CON<3,0>) 用硬件控制 D+/D- 上拉和下拉位
- bit 1         **VBUSCHG:** VBUS 充电选择位 <sup>(1)</sup>  
               1 = VBUS 线路设置为充电到 3.3V  
               0 = VBUS 线路设置为充电到 5V
- bit 0         **VBUSDIS:** VBUS 放电使能位 <sup>(1)</sup>  
               1 = VBUS 线路通过电阻放电  
               0 = VBUS 线路未放电

**注 1:** 这些位仅在主机模式下使用; 不可在设备模式下使用。

# PIC24FJ128GB204 系列

寄存器 19-5: 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

<b>图注:</b>	HS = 可由硬件置 1 的位	HC = 可由硬件清零的位
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 15-8      **未实现:** 读为 0
- bit 7        **UACTPND:** USB 活动暂挂位  
1 = 此时不应暂挂模块 (需要 USLPGRD 位置 1)  
0 = 模块可以暂挂或掉电
- bit 6-5      **未实现:** 读为 0
- bit 4        **USLPGRD:** USB 休眠 / 暂挂保护位  
1 = 表示 USB 模块即将被暂挂或掉电  
0 = 未暂挂
- bit 3-2      **未实现:** 读为 0
- bit 1        **USUSPND:** USB 暂挂模式使能位  
1 = USB OTG 模块处于暂挂模式; USB 时钟是门控的, 且收发器处于低功耗状态  
0 = 正常的 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 <sup>(1)</sup>

注 1: 不要清零该位, 除非 HOSTEN、USBEN 和 OTGEN 位 (U1CON<3,0> 和 U1OTGCON<2>) 都被清零。

# PIC24FJ128GB204 系列

**寄存器 19-6: 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 <sup>(1)</sup>	—	—
bit 7						bit 0	

<b>图注:</b>	U = 未实现位, 读为 0
R = 可读位	W = 可写位
-n = POR 时的值	1 = 置 1
	HSC = 可硬件置 1/ 清零位
	0 = 清零
	x = 未知

bit 15-8 **未实现:** 读为 0

bit 7-4 **ENDPT<3:0>:** 上一个活动端点编号位  
(表示上一次 USB 传输更新的 BDT 编号)

1111 = 端点 15

1110 = 端点 14

.

.

.

0001 = 端点 1

0000 = 端点 0

bit 3 **DIR:** 上一个 BD 方向指示位

1 = 上一个事务是发送传输 (TX)

0 = 上一个事务是接收传输 (RX)

bit 2 **PPBI:** 乒乓 BD 指针指示位<sup>(1)</sup>

1 = 上一个事务针对奇数 BD 存储区

0 = 上一个事务针对偶数 BD 存储区

bit 1-0 **未实现:** 读为 0

**注 1:** 此位仅对具有可用偶数和奇数 BD 寄存器的端点有效。

# PIC24FJ128GB204 系列

## 寄存器 19-7: U1CON: USB 控制寄存器 (设备模式)

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

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							bit 0

<b>图注:</b>	U = 未实现位, 读为 0
R = 可读位	W = 可写位
-n = POR 时的值	1 = 置 1
	HSC = 可硬件置 1/ 清零位
	0 = 清零
	x = 未知

- bit 15-7 **未实现:** 读为 0
- bit 6 **SE0:** 有效单端零标志位  
1 = USB 总线上检测到有效的单端零  
0 = 未检测到单端零
- 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 = 使能处理 SIE 令牌和数据包
- 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; 用硬件激活 D+ 和 D- 上的下拉  
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:** 乒乓缓冲区复位位  
1 = 将所有乒乓缓冲区指针复位到偶数 BD 存储区  
0 = 不复位乒乓缓冲区指针
- bit 0 **USBEN:** USB 模块使能位  
1 = 使能 USB 模块和支持电路 (设备已连接); 用硬件激活 D+ 上拉  
0 = 禁止 USB 模块和支持电路 (设备已断开连接)

# PIC24FJ128GB204 系列

**寄存器 19-8: U1CON: USB 控制寄存器 (仅主机模式)**

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	

<b>图注:</b>	U = 未实现位, 读为 0
R = 可读位	W = 可写位
-n = POR 时的值	1 = 置 1
	HSC = 可硬件置 1/ 清零位
	0 = 清零
	x = 未知

- bit 15-8     **未实现:** 读为 0
- bit 7       **JSTATE:** 有效差分接收器 J 状态标志位  
1 = 在 USB 上检测到 J 状态 (低速模式下为差分 0, 全速模式下为差分 1)  
0 = 未检测到 J 状态
- bit 6       **SE0:** 有效单端零标志位  
1 = USB 总线上检测到有效的单端零  
0 = 未检测到单端零
- bit 5       **TOKBUSY:** 令牌忙状态位  
1 = USB 模块在 On-The-Go 状态下执行令牌  
0 = 不执行令牌
- bit 4       **USBRST:** USB 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 10 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; 用硬件激活 D+ 和 D- 上的下拉  
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; 软件必须置 1 该位 10 ms 然后清零, 以使能远程唤醒  
0 = 禁止恢复信号
- bit 1       **PPBRST:** 乒乓缓冲区复位位  
1 = 将所有乒乓缓冲区指针复位到偶数 BD 存储区  
0 = 不复位乒乓缓冲区指针
- bit 0       **SOFEN:** 帧起始使能位  
1 = 每隔 1 ms 发送一次帧起始令牌  
0 = 禁止帧起始令牌

# PIC24FJ128GB204 系列

**寄存器 19-9: 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 <sup>(1)</sup>	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 <sup>(1)</sup>  
               1 = USB 模块工作在低速模式下  
               0 = USB 模块工作在全速模式下  
 bit 6-0      **ADDR<6:0>:** USB 设备地址位

**注 1:** 仅主机模式; 在设备模式下, 该位未实现, 读为 0。

**寄存器 19-10: U1TOK: USB 令牌寄存器 (仅主机模式)**

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 <sup>(1)</sup>  
               1001 = IN (RX) 令牌类型事务 <sup>(1)</sup>  
               0001 = OUT (TX) 令牌类型事务 <sup>(1)</sup>  
 bit 3-0      **EP<3:0>:** 令牌命令端点地址位  
               该值必须指定所连接设备上的有效端点。

**注 1:** 所有其他组合都被保留不用。

# PIC24FJ128GB204 系列

寄存器 19-11: **U1SOF: USB OTG 帧起始计数器寄存器** (仅主机模式)

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	R/W-0	R/W-0
CNT<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 15-8 **未实现:** 读为 0

bit 7-0 **CNT<7:0>:** 帧起始大小位  
值表示 10 + (n 字节的数据包大小)。例如:

0100 1010 = 64 字节包

0010 1010 = 32 字节包

0001 0010 = 8 字节包

# PIC24FJ128GB204 系列

## 寄存器 19-12: 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 <sup>(1)</sup>	—	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  $\overline{OE}$  监视器使能位 <sup>(1)</sup>  
                   1 =  $\overline{OE}$  信号有效; 它表示驱动 D+/D- 线的时间间隔  
                   0 =  $\overline{OE}$  信号无效
- bit 5            **未实现:** 读为 0
- bit 4            **USBSIDL:** 空闲模式下 USB OTG 停止位  
                   1 = 当器件进入空闲模式时, 模块停止工作  
                   0 = 模块在空闲模式下继续工作
- bit 3-2        **未实现:** 读为 0
- bit 1-0        **PPB<1:0>:** 乒乓缓冲区配置位  
                   11 = 为端点 1 至 15 使能偶数 / 奇数乒乓缓冲区  
                   10 = 为所有端点使能偶数 / 奇数乒乓缓冲区  
                   01 = 为 OUT 端点 0 使能偶数 / 奇数乒乓缓冲区  
                   00 = 禁止偶数 / 奇数乒乓缓冲区

**注 1:** 此位仅在 UTRDIS 位 (U1CNFG2<0>) 置 1 时有效。

# PIC24FJ128GB204 系列

## 寄存器 19-13: 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	U-0	R/W-0
—	—	—	PUVBUS	EXTI2CEN	UVBUSDIS <sup>(1)</sup>	—	UTRDIS <sup>(1)</sup>
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 I<sup>2</sup>C™ 接口控制使能位

1 = 通过 I<sup>2</sup>C 接口控制外部模块

0 = 通过专用引脚控制外部模块

bit 2 **UVBUSDIS:** USB 片内 5V 升压稳压器构建器禁止位 <sup>(1)</sup>

1 = 禁止片内升压稳压器构建器; 使能数字输出控制接口

0 = 激活片内升压稳压器构建器

bit 1 **未实现:** 读为 0

bit 0 **UTRDIS:** USB 片内收发器禁止位 <sup>(1)</sup>

1 = 禁止片内收发器; 使能数字收发器接口

0 = 激活片内收发器

注 1: USBPWR 位置 1 (U1PWRC<0> = 1) 时, 绝不要更改这些位。

# PIC24FJ128GB204 系列

## 19.7.2 USB 中断寄存器

寄存器 19-14: **U1OTGIR: USB OTG 中断状态寄存器 (仅主机模式)**

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

<b>图注:</b>	U = 未实现位, 读为 0
R = 可读位	K = 写 1 以清零位
-n = POR 时的值	HS = 可由硬件置 1 的位
	1 = 置 1
	0 = 清零
	x = 未知

- bit 15-8 **未实现:** 读为 0
- bit 7 **IDIF:** ID 状态更改指示位  
1 = 检测到 ID 状态发生了更改  
0 = 未检测到 ID 状态更改
- bit 6 **T1MSECIF:** 1 ms 定时器位  
1 = 1 ms 定时器已超时  
0 = 1 ms 定时器未超时
- bit 5 **LSTATEIF:** 线路状态稳定指示位  
1 = USB 线路状态 (由 SE0 和 JSTATE 定义) 已稳定 1 ms, 但与上次不同  
0 = USB 线路状态稳定未达到 1 ms
- bit 4 **ACTVIF:** 总线活动指示位  
1 = 检测到 D+/D- 线路或 VBUS 上有活动  
0 = 未检测到 D+/D- 线路或 VBUS 上有活动
- bit 3 **SESVDIF:** 会话有效更改指示位  
1 = VBUS 电压已超过  $V_{A\_SESS\_END}$  (如“USB 2.0 规范”所定义) <sup>(1)</sup>  
0 = VBUS 电压未超过  $V_{A\_SESS\_END}$
- bit 2 **SESENDIF:** B 设备 VBUS 变化指示位  
1 = 检测到 B 设备上的 VBUS 发生变化; VBUS 电压已超过  $V_{B\_SESS\_END}$  (如“USB 2.0 规范”所定义) <sup>(1)</sup>  
0 = VBUS 电压未超过  $V_{B\_SESS\_END}$
- bit 1 **未实现:** 读为 0
- bit 0 **VBUSVDIF:** A 设备 VBUS 变化指示位  
1 = 检测到 A 设备上的 VBUS 发生变化; VBUS 电压已超过  $V_{A\_VBUS\_VLD}$  (如“USB 2.0 规范”所定义) <sup>(1)</sup>  
0 = 未检测到 A 设备上的 VBUS 发生变化

**注 1:** 穿越 VBUS 门限值的事件可以发生在电压上升过程也可以发生在电压下降过程。

**注:** 各个位只能通过向该位所在的单元写入 1 (作为对整个寄存器的字写操作的一部分) 来清零。使用布尔指令或位操作指令写单个位所在的单元会导致写过程中整个寄存器中所有置 1 的位都被清零。

# PIC24FJ128GB204 系列

寄存器 19-15: U1OTGIE: USB OTG 中断允许寄存器 (仅主机模式)

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	SESVIE	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 ms 定时器中断允许位  
               1 = 允许中断  
               0 = 禁止中断
- bit 5        **LSTATEIE:** 线路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        **SESVIE:** 会话有效中断允许位  
               1 = 允许中断  
               0 = 禁止中断
- bit 2        **SESENDIE:** B 设备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 设备 VBUS 有效中断允许位  
               1 = 允许中断  
               0 = 禁止中断

# PIC24FJ128GB204 系列

寄存器 19-16: U1IR: USB 中断状态寄存器 (仅设备模式)

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

<b>图注:</b>	U = 未实现位, 读为 0
R = 可读位	K = 写 1 以清零位
-n = POR 时的值	HS = 可由硬件置 1 的位
	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 = 在设备模式下, 外设在事务的握手阶段发送 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 = 观察到 D+ 或 D- 引脚上保持 K 状态 2.5 μs (低速模式下为差分 1, 全速模式下为差分 0)  
0 = 未观察到 K 状态
- bit 4       **IDLEIF:** 空闲检测中断位  
1 = 检测到空闲条件 (保持连续的空闲状态 3 ms 或更多时间)  
0 = 未检测到空闲条件
- bit 3       **TRNIF:** 令牌处理完成中断位  
1 = 完成当前令牌的处理; 从 U1STAT 寄存器中读取端点信息  
0 = 未完成当前令牌的处理; 清零 U1STAT 寄存器或从 U1STAT 装入下个令牌 (清零此位将导致 U1STAT FIFO 递增)
- bit 2       **SOFIF:** 帧起始令牌中断位  
1 = 外设接收到帧起始令牌或主机到达帧起始门限值  
0 = 未接收到帧起始令牌或未到达帧起始门限值
- bit 1       **UERRIF:** USB 错误条件中断位 (只读)  
1 = 发生未屏蔽的错误条件; 仅 U1EIE 寄存器中允许的错误状态才能将此位置 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 μs 的时间; 在此位失效之前必须清零复位状态  
0 = 未发生 USB 复位。各个位只能通过向该位所在的单元写入 1 (作为对整个寄存器的字写操作的一部分) 来清零。使用布尔指令或位操作指令写单个位所在的单元会导致写过程中整个寄存器中所有置 1 的位都被清零。

**注:** 各个位只能通过向该位所在的单元写入 1 (作为对整个寄存器的字写操作的一部分) 来清零。使用布尔指令或位操作指令写单个位所在的单元会导致写过程中整个寄存器中所有置 1 的位都被清零。

# PIC24FJ128GB204 系列

## 寄存器 19-17: U1IR: USB 中断状态寄存器 (仅主机模式)

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

R/K-0, HS							
STALLIF	ATTACHIF	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	DETACHIF
bit 7						bit 0	

<b>图注:</b>	U = 未实现位, 读为 0		
R = 可读位	K = 写 1 以清零位	HS = 可由硬件置 1 的位	
-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 = 在设备模式下, 外设从事务的握手阶段发送 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 且在 2.5 μs 内没有总线活动则置 1 该位  
0 = 未检测到外设连接
- bit 5       **RESUMEIF:** 恢复中断位  
1 = 观察到 D+ 或 D- 引脚上保持 K 状态 2.5 μs (低速模式下为差分 1, 全速模式下为差分 0)  
0 = 未观察到 K 状态
- bit 4       **IDLEIF:** 空闲检测中断位  
1 = 检测到空闲条件 (保持连续的空闲状态 3 ms 或更多时间)  
0 = 未检测到空闲条件
- bit 3       **TRNIF:** 令牌处理完成中断位  
1 = 完成当前令牌的处理; 从 U1STAT 寄存器中读取端点信息  
0 = 未完成当前令牌的处理; 清零 U1STAT 寄存器或从 U1STAT 装入下个令牌
- bit 2       **SOFIF:** 帧起始令牌中断位  
1 = 外设接收到帧起始令牌或主机到达帧起始门限值  
0 = 未接收到帧起始令牌或未到达帧起始门限值
- bit 1       **UERRIF:** USB 错误条件中断位  
1 = 发生未屏蔽的错误条件; 仅 U1EIE 寄存器中允许的错误状态才能将此位置 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; 在此位失效之前必须清零复位状态  
0 = 未检测到外设断开连接。各个位只能通过向该位所在的单元写入 1 (作为对整个寄存器的字写操作的一部分) 来清零。使用布尔指令或位操作指令写单个位所在的单元会导致写过程中整个寄存器中所有置 1 的位都被清零。

**注:** 各个位只能通过向该位所在的单元写入 1 (作为对整个寄存器的字写操作的一部分) 来清零。使用布尔指令或位操作指令写单个位所在的单元会导致写过程中整个寄存器中所有置 1 的位都被清零。

# PIC24FJ128GB204 系列

寄存器 19-18: **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/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS
STALLIE	ATTACHIE <sup>(1)</sup>	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	URSTIE DETACHIE <sup>(1)</sup>
bit 7							bit 0

**图注:** U = 未实现位, 读为 0  
R = 可读位 K = 写 1 以清零位 HS = 可由硬件置 1 的位  
-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 **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 (仅主机模式) <sup>(1)</sup>  
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:** 帧起始令牌中断位  
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 (设备模式) 或 USB 断开连接中断 (主机模式) 允许位 <sup>(1)</sup>  
1 = 允许中断  
0 = 禁止中断

**注 1:** ATTACHIE 和 DETACHIE 位在设备模式下未实现, 读为 0。

# PIC24FJ128GB204 系列

## 寄存器 19-19: 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					
BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF
						EOFEF	
bit 7						bit 0	

<b>图注:</b>	U = 未实现位, 读为 0
R = 可读位	K = 写 1 以清零位
-n = POR 时的值	HS = 可由硬件置 1 的位
	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 = 未检测到位填充错误
- 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所接收的字节数。接收到的数据被截短。  
0 = 没有 DMA 错误
- bit 4 **BTOEF:** 总线周转超时错误标志位  
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 = 数据字段的字节是整数
- bit 2 **CRC16EF:** CRC16 失败标志位  
1 = CRC16 失败  
0 = CRC16 通过
- bit 1 **在设备模式下:**  
**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 = 接受令牌数据包 (无 CRC5 错误)  
**在主机模式下:**  
**EOFEF:** 帧结束错误标志位  
1 = 发生了帧结束错误  
0 = 禁止帧结束中断
- bit 0 **PIDEF:** PID 检查失败标志位  
1 = PID 检查失败  
0 = PID 检查通过

**注:** 各个位只能通过向该位所在的单元写入 1 (作为对整个寄存器的字写操作的一部分) 来清零。使用布尔指令或位操作指令写单个位所在的单元会导致写过程中整个寄存器中所有置 1 的位都被清零。

# PIC24FJ128GB204 系列

## 寄存器 19-20: 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         在设备模式下:  
                  **CRC5EE:** CRC5 主机错误中断允许位  
                  1 = 允许中断  
                  0 = 禁止中断  
                  在主机模式下:  
                  **EOFEE:** 帧结束错误中断允许位  
                  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

## 19.7.3 USB 端点管理寄存器

寄存器 19-21: **U1EPn: USB 端点 n 控制寄存器 (n = 0 至 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 <sup>(1)</sup>	RETRYDIS <sup>(1)</sup>	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSK
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) <sup>(1)</sup>  
             1 = 使能直接连接到低速设备  
             0 = 禁止直接连接到低速设备
- bit 6       **RETRYDIS:** 重试禁止位 (仅限 U1EP0) <sup>(1)</sup>  
             1 = 禁止重试 NAK 事务  
             0 = 使能重试 NAK 事务; 用硬件完成重试
- 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 进行控制传输; 仅允许发送和接收传输  
             0 = 使能端点 n 进行控制 (SETUP) 传输; 也允许发送和接收传输  
             对于 **EPTXEN** 和 **EPRXEN** 的所有其他组合:  
             该位被忽略。
- 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       **EPHSK:** 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, 读为 0。

# PIC24FJ128GB204 系列

---

注:

## 20.0 数据信号调制器 (DSM)

**注：** 本数据手册总结了该 PIC24F 系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“数据信号调制器 (DSM)” (DS39744)。本数据手册中的信息取代了 FRM 中的信息。

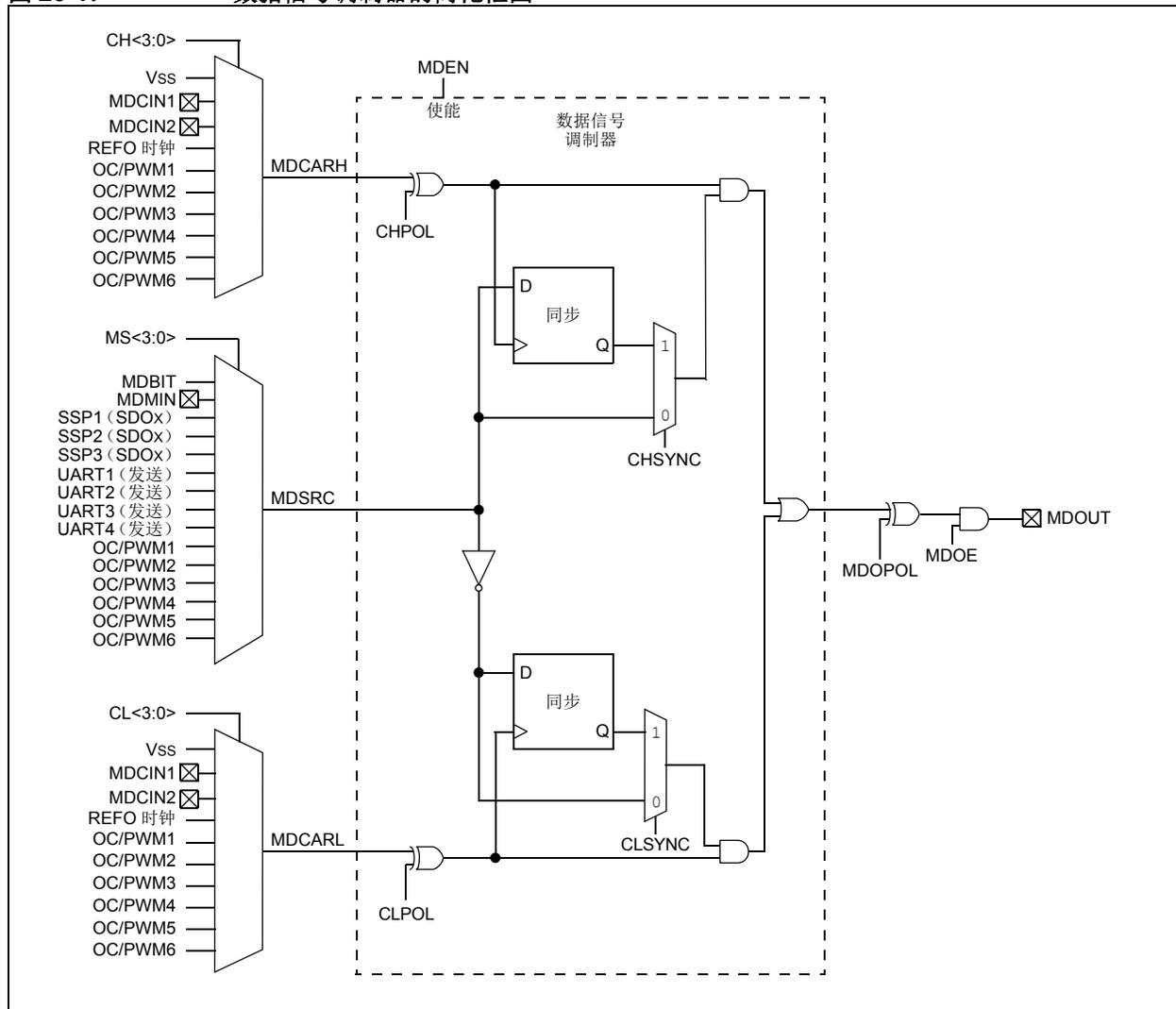
调制输出信号通过对载波信号和调制器信号执行逻辑“与”运算生成，然后提供给 MDOUT 引脚。使用此方法，DSM 可以生成以下类型的键调制方案：

- 频移键控 (Frequency-Shift Keying, FSK)
- 相移键控 (Phase-Shift Keying, PSK)
- 开关键控 (On-Off Keying, OOK)

图 20-1 给出了数据信号调制器外设的简化框图。

数据信号调制器 (DSM) 允许用户将数字数据流 (调制器信号) 与载波信号混合以产生调制输出。载波信号和调制器信号通过外设输出在内部或通过输入引脚从外部提供给 DSM 模块。

图 20-1: 数据信号调制器的简化框图



# PIC24FJ128GB204 系列

寄存器 20-1: MDCON: 数据信号调制器控制寄存器

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
MDEN	—	MDSIDL	—	—	—	—	—
bit 15							bit 8
U-0	R/W-0	R/W-1	R/W-0	U-0	U-0	U-0	R/W-0
—	MDOE	MDSLRL	MDOPOL	—	—	—	MDBIT <sup>(1)</sup>
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            **MDEN:** DSM 模块使能位  
 1 = 使能调制器模块, 并对输入信号进行混合  
 0 = 禁止调制器模块, 不产生任何输出
- bit 14            **未实现:** 读为 0
- bit 13            **MDSIDL:** 空闲模式 DSM 停止位  
 1 = 当器件进入空闲模式后, 模块停止工作  
 0 = 模块在空闲模式下继续工作
- bit 12-7        **未实现:** 读为 0
- bit 6            **MDOE:** DSM 模块引脚输出使能位  
 1 = 使能调制器引脚输出  
 0 = 禁止调制器引脚输出
- bit 5            **MDSLRL:** MDOUT 引脚压摆率限制位  
 1 = 使能 MDOUT 引脚压摆率限制  
 0 = 禁止 MDOUT 引脚压摆率限制
- bit 4            **MDOPOL:** DSM 输出极性选择位  
 1 = 调制器输出信号反相  
 0 = 调制器输出信号不反相
- bit 3-1         **未实现:** 读为 0
- bit 0            **MDBIT:** 手动调制输入位 <sup>(1)</sup>  
 1 = 调制载波  
 0 = 不调制载波

注 1: 必须选择 MDBIT 作为调制源 (MDSRC<3:0> = 0000)。

# PIC24FJ128GB204 系列

寄存器 20-2: MDSRC: 数据信号调制器源控制寄存器

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

R/W-x	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
SODIS <sup>(1)</sup>	—	—	—	MS3 <sup>(2)</sup>	MS2 <sup>(2)</sup>	MS1 <sup>(2)</sup>	MS0 <sup>(2)</sup>
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       **SODIS:** DSM 源输出禁止位 <sup>(1)</sup>  
 1 = 禁止驱动外设输出引脚的输出信号 (通过 MS<3:0> 选择)  
 0 = 使能驱动外设输出引脚的输出信号 (通过 MS<3:0> 选择)

bit 6-4     **未实现:** 读为 0

bit 3-0     **MS<3:0>:** DSM 源选择位 <sup>(2)</sup>  
 1111 = 未实现  
 1110 = SPI3 模块输出 (SDO3)  
 1101 = 输出比较 /PWM 模块 6 输出  
 1100 = 输出比较 /PWM 模块 5 输出  
 1011 = 输出比较 /PWM 模块 4 输出  
 1010 = 输出比较 /PWM 模块 3 输出  
 1001 = 输出比较 /PWM 模块 2 输出  
 1000 = 输出比较 /PWM 模块 1 输出  
 0111 = UART4 发送输出  
 0110 = UART3 发送输出  
 0101 = UART2 发送输出  
 0100 = UART1 发送输出  
 0011 = SPI2 模块输出 (SDO2)  
 0010 = SPI1 模块输出 (SDO1)  
 0001 = MDMIN 引脚上的输入  
 0000 = 使用 MDBIT (MDCON<0>) 手动调制

- 注   **1:** 此位仅受到 POR 影响。  
      **2:** 这些位不受 POR 影响。

# PIC24FJ128GB204 系列

## 寄存器 20-3: MDCAR: 数据信号调制器载波控制寄存器

R/W-x	R/W-x	R/W-x	U-0	R/W-x	R/W-x	R/W-x	R/W-x
CHODIS	CHPOL	CHSYNC	—	CH3 <sup>(1)</sup>	CH2 <sup>(1)</sup>	CH1 <sup>(1)</sup>	CH0 <sup>(1)</sup>
bit 15							bit 8

R/W-0	R/W-x	R/W-x	U-0	R/W-x	R/W-x	R/W-x	R/W-x
CLODIS	CLPOL	CLSYNC	—	CL3 <sup>(1)</sup>	CL2 <sup>(1)</sup>	CL1 <sup>(1)</sup>	CL0 <sup>(1)</sup>
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            **CHODIS:** DSM 高载波输出禁止位  
 1 = 禁止驱动外设输出引脚的输出信号 (通过 CH<3:0> 选择)  
 0 = 使能驱动外设输出引脚的输出信号
- bit 14            **CHPOL:** DSM 高载波极性选择位  
 1 = 选定的高载波信号反相  
 0 = 选定的高载波信号不反相
- bit 13            **CHSYNC:** DSM 高载波同步使能位  
 1 = 调制器在允许切换到低电平载波前等待高电平载波信号上出现下降沿  
 0 = 调制器输出不与高电平载波信号同步 <sup>(1)</sup>
- bit 12            **未实现:** 读为 0
- bit 11-8        **CH<3:0>:** DSM 数据高载波选择位 <sup>(1)</sup>  
 1111  
 ... = 保留  
 1010  
 1001 = 输出比较 /PWM 模块 6 输出  
 1000 = 输出比较 /PWM 模块 5 输出  
 0111 = 输出比较 /PWM 模块 4 输出  
 0110 = 输出比较 /PWM 模块 3 输出  
 0101 = 输出比较 /PWM 模块 2 输出  
 0100 = 输出比较 /PWM 模块 1 输出  
 0011 = 参考时钟输出 (REFO)  
 0010 = MDCIN2 引脚上的输入  
 0001 = MDCIN1 引脚上的输入  
 0000 = Vss
- bit 7            **CLODIS:** DSM 低载波输出禁止位  
 1 = 禁止驱动外设输出引脚的输出信号 (通过 CL<3:0> 选择)  
 0 = 使能驱动外设输出引脚的输出信号
- bit 6            **CLPOL:** DSM 低载波极性选择位  
 1 = 选定的低载波信号反相  
 0 = 选定的低载波信号不反相
- bit 5            **CLSYNC:** DSM 低载波同步使能位  
 1 = 调制器在允许切换到高电平载波前等待低电平载波信号上出现下降沿  
 0 = 调制器输出不与低电平载波信号同步 <sup>(1)</sup>
- bit 4            **未实现:** 读为 0
- bit 3-0        **CL<3:0>:** DSM 数据低载波选择位 <sup>(1)</sup>  
 位设置与 CH<3:0> 的那些位设置相同。

注 1: 如果载波未同步, 信号流中可能发生载波脉冲宽度变窄或毛刺的情况。

## 21.0 增强型并行主端口 (EPMP)

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。更多信息，请参见《dsPIC33/PIC24系列参考手册》中的“**增强型并行主端口 (EPMP)**” (DS39730)。本数据手册中的信息取代了 FRM 中的信息。

增强型并行主端口 (EPMP) 模块提供了并行的 4 位 (仅适用于主模式) 和 8 位 (主模式和从模式) 数据总线接口，用于与片外模块 (例如存储器、FIFO、LCD 控制器和其他单片机) 进行通信。该模块在通信总线上可以用作主器件或从器件。

对于 EPMP 主模式，所有外部地址均可映射到内部扩展数据空间 (EDS)。这通过为每个片选分配 EDS 区域，然后将片选分配到如存储器或外部控制器等特定外部资源实现。该区域不应再被分配给其他器件资源，如 RAM 或 SFR。要对外部资源执行读或写操作，CPU 只需在为 EPMP 分配的地址范围内执行读或写操作。

EPMP 模块的主要特性包括：

- 扩展数据空间 (EDS) 接口支持直接从 CPU 访问
- 最多 10 条可编程地址线
- 最多 2 条片选信号线
- 最多 1 条应答线 (每个片选信号一条)
- 4 位和 8 位宽数据总线
- 可编程选通选项 (每个片选信号)
  - 独立的读和写选通，或；
  - 带使能选通的读 / 写选通
- 可编程地址 / 数据复用
- 可编程地址等待状态
- 可编程数据等待状态 (每个片选信号)
- 控制信号的可编程极性 (每个片选信号)
- 支持传统的并行从端口 (Parallel Slave Port , PSP)
- 支持增强型并行从端口
  - 地址支持
  - 4 字节深自动递增缓冲区

### 21.1 不同模式下存储器寻址

器件可寻址的存储空间取决于地址 / 数据复用选择；范围为 1K 至 2 MB。关于不同存储器寻址模式，请参见表 21-1。

# PIC24FJ128GB204 系列

表 21-1: 不同模式下的存储器寻址

数据端口大小	PMA<9:8>	PMA<7:0>	PMD<7:4>	PMD<3:0>	可访问的存储空间
<b>解复用的地址 (ADRMUX&lt;1:0&gt; = 00)</b>					
8 位 (PTSZ<1:0> = 00)	地址 <9:8>	地址 <7:0>	数据		1K
4 位 (PTSZ<1:0> = 01)	地址 <9:8>	地址 <7:0>	—	数据	1K
<b>1 个地址段 (ADRMUX&lt;1:0&gt; = 01)</b>					
8 位 (PTSZ<1:0> = 00)	—	PMALL	地址 <7:0> 数据		1K
4 位 (PTSZ<1:0> = 01)	地址 <9:8>	PMALL	地址 <7:4>	地址 <3:0>	1K
			—	数据 (1)	
<b>2 个地址段 (ADRMUX&lt;1:0&gt; = 10)</b>					
8 位 (PTSZ<1:0> = 00)	—	PMALL	地址 <7:0>		64K
		PMALH	地址 <15:8>		
		—	数据		
4 位 (PTSZ<1:0> = 01)	地址 <9:8>	PMALL	地址 <3:0>		1K
		PMALH	地址 <7:4>		
		—	数据		
<b>3 个地址段 (ADRMUX&lt;1:0&gt; = 11)</b>					
8 位 (PTSZ<1:0> = 00)	—	PMALL	地址 <7:0>		2 MB
		PMALH	地址 <15:8>		
		PMALU	地址 <22:16>		
		—	数据		
4 位 (PTSZ<1:0> = 01)	地址 <13:12>	PMALL	地址 <3:0>		16K
		PMALH	地址 <7:4>		
		PMALU	地址 <11:8>		
		—	数据		

表 21-2: 增强型并行主端口引脚说明

引脚名称 (备用功能)	类型	说明
PMA<14> (PMCS1)	O	地址总线位 bit 14
	I/O	数据总线位 bit 14 (带地址复用的 16 位端口)
	O	片选 1 (备用位置)
PMA<9:3>	O	地址总线位 bit<9:3>
PMA<2> (PMALU)	O	地址总线位 bit 2
	O	复用地址的地址锁存器最高字节选通
PMA<1> (PMALH)	I/O	地址总线位 bit 1
	O	复用地址的地址锁存器高字节选通
PMA<0> (PMALL)	I/O	地址总线位 bit 0
	O	复用地址的地址锁存器低字节选通
PMD<7:0>	I/O	数据总线位 bit<7:0>, 数据位 bit<15:8>
	O	地址总线位 bit<7:0>
PMCS1	I/O	片选 1
PMCS2	I/O	片选 2
PMWR	I/O	写选通
PMRD	I/O	读选通
PMBE1	O	字节指示
PMBE0	O	半字节或字节指示
PMACK1	I	应答信号 1

# PIC24FJ128GB204 系列

## 寄存器 21-1: PMCON1: EPMP 控制寄存器 1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	—	MODE1	MODE0
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSF1	CSF0	ALP	ALMODE	—	BUSKEEP	IRQM1	IRQM0
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            **PMPEN:** 增强型并行主端口使能位  
 1 = 使能 EPMP  
 0 = 禁止 EPMP
- bit 14            **未实现:** 读为 0
- bit 13            **PSIDL:** 空闲模式 PMP 停止位  
 1 = 当器件进入空闲模式后, 模块停止工作  
 0 = 模块在空闲模式下继续工作
- bit 12-11        **ADRMUX<1:0>:** 地址 / 数据复用选择位  
 11 = 低地址位使用 3 个地址段与数据位复用  
 10 = 低地址位使用 2 个地址段与数据位复用  
 01 = 低地址位使用 1 个地址段与数据位复用  
 00 = 地址和数据使用独立的引脚
- bit 10            **未实现:** 读为 0
- bit 9-8          **MODE<1:0>:** 并行端口模式选择位  
 11 = 主模式  
 10 = 增强型 PSP: 所使用的引脚为 PMRD、PMWR、PMCS、PMD<7:0> 和 PMA<1:0>  
 01 = 缓冲 PSP: 所使用的引脚为 PMRD、PMWR、PMCS 和 PMD<7:0>  
 00 = 传统并行从端口: 所使用的引脚为 PMRD、PMWR、PMCS 和 PMD<7:0>
- bit 7-6          **CSF<1:0>:** 片选功能位  
 11 = 保留  
 10 = PMA<14> 用作片选 1  
 01 = 保留  
 00 = PMCS1 用作片选 1
- bit 5            **ALP:** 地址锁存器极性位  
 1 = 高电平有效 (PMALL、PMALH 和 PMALU)  
 0 = 低电平有效 (PMALL、PMALH 和 PMALU)
- bit 4            **ALMODE:** 地址锁存器选通模式位  
 1 = 使能“智能”地址选通 (仅在当前访问将导致锁存地址不同于先前地址时, 才存在各地址段)  
 0 = 禁止“智能”地址选通
- bit 3            **未实现:** 读为 0
- bit 2            **BUSKEEP:** 总线保持单元位  
 1 = 当未驱动数据总线时, 保持其最后一个值  
 0 = 当未驱动数据总线时, 其处于高阻态
- bit 1-0         **IRQM<1:0>:** 中断请求模式位  
 11 = 当读取读缓冲区 3 或写入写缓冲区 3 时产生中断 (缓冲的 PSP 模式), 或当 PMA<1:0> = 11 时  
       执行读或写操作时产生中断 (仅限可寻址 PSP 模式)  
 10 = 保留  
 01 = 在读 / 写周期结束时产生中断  
 00 = 不产生中断

# PIC24FJ128GB204 系列

## 寄存器 21-2: PMCON2: EPMP 控制寄存器 2

R-0, HSC	U-0	R/C-0, HS	R/C-0, HS	U-0	U-0	U-0	U-0
PMPBUSY	—	ERROR	TIMEOUT	—	—	—	—
bit 15							bit 8

R/W-0							
RADDR23 <sup>(1)</sup>	RADDR22 <sup>(1)</sup>	RADDR21 <sup>(1)</sup>	RADDR20 <sup>(1)</sup>	RADDR19 <sup>(1)</sup>	RADDR18 <sup>(1)</sup>	RADDR17 <sup>(1)</sup>	RADDR16 <sup>(1)</sup>
bit 7							bit 0

<b>图注:</b>	HS = 可由硬件置 1 的位	HSC = 硬件置 1/ 清零位
R = 可读位	W = 可写位	U = 未实现, 读为 0
-n = POR 时的值	1 = 置 1	0 = 清零
		C = 可清零位

- bit 15      **PMPBUSY:** PMP 忙位 (仅用于主模式)  
1 = 端口忙  
0 = 端口不忙
- bit 14      **未实现:** 读为 0
- bit 13      **ERROR:** PMP 错误位  
1 = 传输错误 (请求非法传输)  
0 = 传输成功完成
- bit 12      **TIMEOUT:** PMP 超时位  
1 = 传输超时  
0 = 传输成功完成
- bit 11-8    **未实现:** 读为 0
- bit 7-0     **RADDR<23:16>:** 并行主端口保留地址空间位 <sup>(1)</sup>

**注 1:** 如果 RADDR<23:16> = 00000000, 则片选 2 的最后一个 EDS 地址将为 FFFFFFFh。

# PIC24FJ128GB204 系列

## 寄存器 21-3: PMCON3: EPMP 控制寄存器 3

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PTWREN	PTRDEN	PTBE1EN	PTBE0EN	—	AWAITM1	AWAITM0	AWAITE
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        **PTWREN:** PMP 写 / 使能选通端口使能位  
 1 = 使能 PMWR 端口  
 0 = 禁止 PMWR 端口
- bit 14        **PTRDEN:** PMP 读 / 写选通端口使能位  
 1 = 使能 PMRD/PMWR 端口  
 0 = 禁止 PMRD/PMWR 端口
- bit 13        **PTBE1EN:** PMP 高半字节 / 字节使能端口使能位  
 1 = 使能 PMBE1 端口  
 0 = 禁止 PMBE1 端口
- bit 12        **PTBE0EN:** PMP 低半字节 / 字节使能端口使能位  
 1 = 使能 PMBE0 端口  
 0 = 禁止 PMBE0 端口
- bit 11        **未实现:** 读为 0
- bit 10-9      **AWAITM<1:0>:** 地址锁存器选通等待状态位  
 11 = 等待 3½ 个 Tcy  
 10 = 等待 2½ 个 Tcy  
 01 = 等待 1½ 个 Tcy  
 00 = 等待 ½ 个 Tcy
- bit 8        **AWAITE:** 地址锁存器选通后地址保持等待状态位  
 1 = 等待 ¼ 个 Tcy  
 0 = 等待 ¼ 个 Tcy
- bit 7-0      **未实现:** 读为 0

# PIC24FJ128GB204 系列

## 寄存器 21-4: PMCON4: EPMP 控制寄存器 4

U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	PTEN14	—	—	—	—	PTEN<9:8>	
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
PTEN<7:3>					PTEN<2:0>		
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      **未实现:** 读为 0

bit 14      **PTEN14:** PMA14 端口使能位  
 1 = PMA14 用作地址线 14 或片选 1  
 0 = PMA14 用作端口 I/O

bit 13-10    **未实现:** 读为 0

bit 9-3      **PTEN<9:3>:** EPMP 地址端口使能位  
 1 = PMA<9:3> 用作 EPMP 地址线  
 0 = PMA<9:3> 用作端口 I/O

bit 2-0      **PTEN<2:0>:** PMALU/PMALH/PMALL 选通使能位  
 1 = PMA<2:0> 用作地址线或地址锁存器选通  
 0 = PMA<2:0> 用作端口 I/O

# PIC24FJ128GB204 系列

寄存器 21-5: PMCSxCF: EPMP 片选 x 配置寄存器

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSDIS	CSP	CSPTEN	BEP	—	WRSP	RDSP	SM
bit 15							bit 8
R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
ACKP	PTSZ1	PTSZ0	—	—	—	—	—
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        **CSDIS:** 片选 x 禁止位  
 1 = 禁止片选 x 功能  
 0 = 使能片选 x 功能
- bit 14        **CSP:** 片选 x 极性位  
 1 = 高电平有效 (PMCSx)  
 0 = 低电平有效 (PMCSx)
- bit 13        **CSPTEN:** PMCSx 端口使能位  
 1 = 使能 PMCSx 端口  
 0 = 禁止 PMCSx 端口
- bit 12        **BEP:** 片选 x 半字节 / 字节使能极性位  
 1 = 半字节 / 字节使能高电平有效 (PMBE0 和 PMBE1)  
 0 = 半字节 / 字节使能低电平有效 (PMBE0 和 PMBE1)
- bit 11        **未实现:** 读为 0
- bit 10        **WRSP:** 片选 x 写选通极性位  
 对于从模式和主模式 (SM = 0):  
 1 = 写选通高电平有效 (PMWR)  
 0 = 写选通低电平有效 (PMWR)  
 对于主模式 (SM = 1):  
 1 = 使能选通高电平有效  
 0 = 使能选通低电平有效
- bit 9         **RDSP:** 片选 x 读选通极性位  
 对于从模式和主模式 (SM = 0):  
 1 = 读选通高电平有效 (PMRD)  
 0 = 读选通低电平有效 (PMRD)  
 对于主模式 (SM = 1):  
 1 = 读 / 写选通高电平有效 (PMRD/PMWR)  
 0 = 读 / 写选通低电平有效 (PMRD/PMWR)
- bit 8         **SM:** 片选 x 选通模式位  
 1 = 读 / 写和使能选通 (PMRD/PMWR)  
 0 = 读和写选通 (PMRD 和 PMWR)
- bit 7         **ACKP:** 片选 x 应答极性位  
 1 = ACK 高电平有效 (PMACK1)  
 0 = ACK 低电平有效 (PMACK1)
- bit 6-5       **PTSZ<1:0>:** 片选 x 端口大小位  
 11 = 保留  
 10 = 保留  
 01 = 4 位端口大小 (PMD<3:0>)  
 00 = 8 位端口大小 (PMD<7:0>)
- bit 4-0       **未实现:** 读为 0

# PIC24FJ128GB204 系列

寄存器 21-6: **PMCSxBS: EPMP 片选 x 基址寄存器** <sup>(2)</sup>

R/W <sup>(1)</sup>							
BASE<23:16>							
bit 15							bit 8

R/W <sup>(1)</sup>	U-0	U-0	U-0	R/W <sup>(1)</sup>	U-0	U-0	U-0
BASE15	—	—	—	BASE11	—	—	—
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 **BASE<23:15>**: 片选 x 基址位 <sup>(1)</sup>

bit 6-4 **未实现**: 读为 0

bit 3 **BASE11**: 片选 x 基址位 <sup>(1)</sup>

bit 2-0 **未实现**: 读为 0

注 1: POR 时的值为 0080h (对于 PMCS1BS) 和 0880h (对于 PMCS2BS)。

2: 如果对整个 PMCS2BS 寄存器写入 0x0000, 则片选 1 的最后一个 EDS 地址将为 FFFFFFFh。在这种情况下, 不应使用片选 2。PMCS1BS 不具有此类功能。

# PIC24FJ128GB204 系列

## 寄存器 21-7: PMCSxMD: EPMP 片选 x 模式寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
ACKM1	ACKM0	AMWAIT2	AMWAIT1	AMWAIT0	—	—	—
bit 15							bit 8

R/W-0							
DWAITB1	DWAITB0	DWAITM3	DWAITM2	DWAITM1	DWAITM0	DWAITE1	DWAITE0
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    **ACKM<1:0>**: 片选 x 应答模式位  
 11 = 保留  
 10 = PMACKx 用于确定何时完成读 / 写操作  
 01 = PMACKx 用于确定何时完成读 / 写操作且超时  
     (如果 DWAITM<3:0> = 0000, 那么最大超时时间为 255 Tcy, 其他情况为 DWAITM<3:0> 对应的周期。)  
 00 = 未使用 PMACKx
- bit 13-11    **AMWAIT<2:0>**: 片选 x 备用主等待状态位  
 111 = 等待 10 个备用主周期  
 .  
 .  
 .  
 001 = 等待 4 个备用主周期  
 000 = 等待 3 个备用主周期
- bit 10-8     **未实现**: 读为 0
- bit 7-6      **DWAITB<1:0>**: 读 / 写选通前片选 x 数据建立等待状态位  
 11 = 等待 3¼ 个 Tcy  
 10 = 等待 2¼ 个 Tcy  
 01 = 等待 1¼ 个 Tcy  
 00 = 等待 ¼ 个 Tcy
- bit 5-2      **DWAITM<3:0>**: 片选 x 数据读 / 写选通等待状态位  
对于写操作:  
 1111 = 等待 15½ 个 Tcy  
 .  
 .  
 .  
 0001 = 等待 1½ 个 Tcy  
 0000 = 等待 ½ 个 Tcy  
对于读操作:  
 1111 = 等待 15¼ 个 Tcy  
 .  
 .  
 .  
 0001 = 等待 1¼ 个 Tcy  
 0000 = 等待 ¾ 个 Tcy

## 寄存器 21-7: PMCSxMD: EPMP 片选 x 模式寄存器 (续)

bit 1-0 **DWAITE<1:0>**: 读 / 写选通后片选 x 数据保持等待状态位

对于写操作:

11 = 等待 3¼ 个 TcY

10 = 等待 2¼ 个 TcY

01 = 等待 1¼ 个 TcY

00 = 等待 ¼ 个 TcY

对于读操作:

11 = 等待 3 个 TcY

10 = 等待 2 个 TcY

01 = 等待 1 个 TcY

00 = 等待 0 个 TcY

## 寄存器 21-8: PMSTAT: EPMP 状态寄存器 (仅从模式)

R-0, HSC	R/W-0, HS	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IBF	IBOV	—	—	IB3F <sup>(1)</sup>	IB2F <sup>(1)</sup>	IB1F <sup>(1)</sup>	IB0F <sup>(1)</sup>
bit 15				bit 8			

R-1, HSC	R/W-0, HS	U-0	U-0	R-1, HSC	R-1, HSC	R-1, HSC	R-1, HSC
OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E
bit 7				bit 0			

<b>图注:</b>	HS = 可由硬件置 1 的位	HSC = 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 **IBF**: 输入缓冲区满状态位  
 1 = 所有可写的输入缓冲寄存器已满  
 0 = 部分或所有可写的输入缓冲寄存器为空

bit 14 **IBOV**: 输入缓冲区溢出状态位  
 1 = 尝试对已满的输入缓冲寄存器执行写操作 (必须由软件清零)  
 0 = 未发生溢出

bit 13-12 **未实现**: 读为 0

bit 11-8 **IB3F:IB0F**: 输入缓冲区 x 满状态位 <sup>(1)</sup>  
 1 = 输入缓冲区包含尚未读取的数据 (读缓冲区将清零此位)  
 0 = 输入缓冲区不包含任何未读取的数据

bit 7 **OBE**: 输出缓冲区空状态位  
 1 = 所有可读的输出缓冲寄存器均为空  
 0 = 部分或所有可读的输出缓冲寄存器已满

bit 6 **OBUF**: 输出缓冲区下溢状态位  
 1 = 对空的输出缓冲寄存器执行了读操作 (必须由软件清零)  
 0 = 未发生下溢

bit 5-4 **未实现**: 读为 0

bit 3-0 **OB3E:OB0E**: 输出缓冲区 x 空状态位  
 1 = 输出缓冲区寄存器为空 (将数据写入该缓冲区会将该位清零)  
 0 = 输出缓冲区寄存器包含未发送的数据

**注 1:** 即使某个位表示缓冲区中的字节, 在进行字节读取时与字 (字节 0 和 1 或字节 2 和 3) 对应的位也被清零。

# PIC24FJ128GB204 系列

寄存器 21-9: PADCFG1: 焊盘配置控制寄存器

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

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-1      **未实现:** 读为 0

bit 0      **PMPTTL:** EPMP 模块 TTL 输入缓冲区选择位

1 = EPMP 模块输入 (PMDx 和 PMCS1) 使用 TTL 输入缓冲器

0 = EPMP 模块输入使用施密特触发器输入缓冲器

## 22.0 实时时钟和日历 (RTCC)

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。关于实时时钟和日历的更多信息，请参见《dsPIC33/PIC24系列参考手册》中的“具有外部电源控制的RTCC”（DS39745）。

RTCC 为用户提供了可校准的实时时钟和日历 (Real-Time Clock and Calendar, RTCC) 功能。

RTCC 模块的主要特性包括：

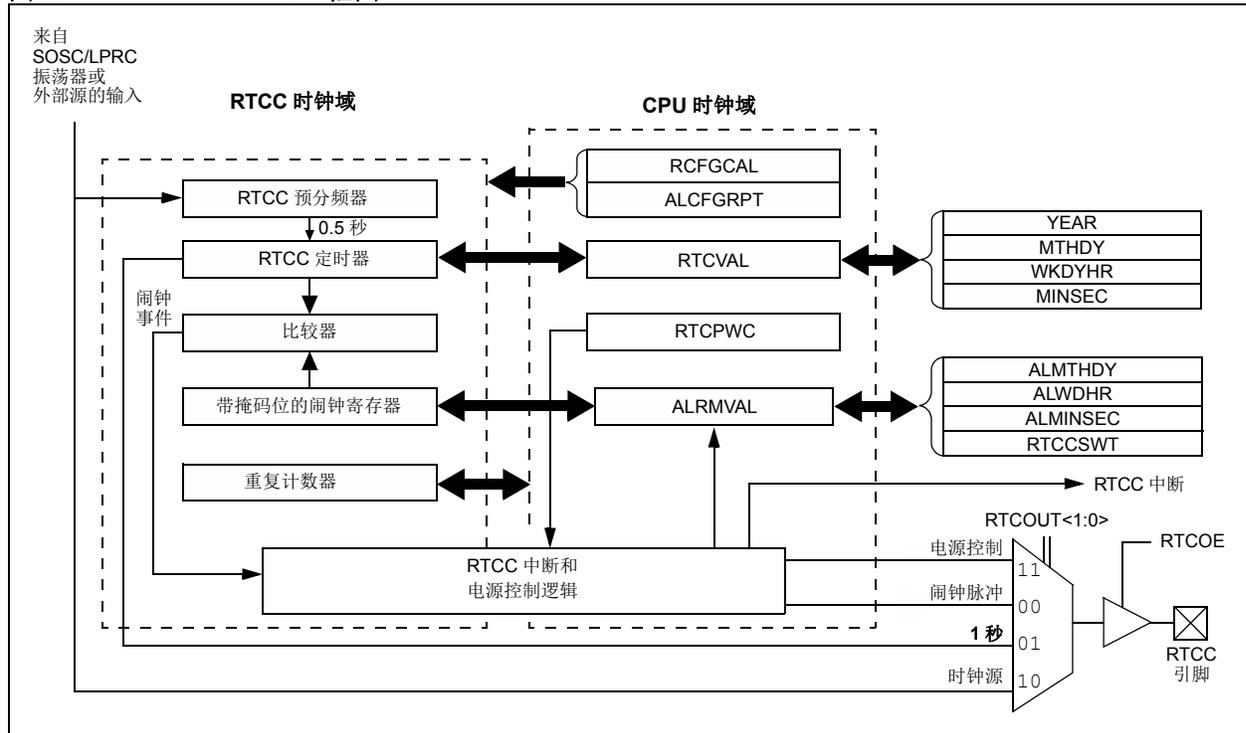
- 可在深度休眠模式下工作
- 可选的时钟源
- 采用 24 小时格式提供小时、分和秒
- 可分辨半秒的时长
- 提供日历：星期、日期、月和年
- 闹钟可配置为 0.5 秒、1 秒、10 秒、1 分、10 分、1 小时、1 天、1 周、1 月或 1 年
- 闹钟使用递减计数器进行重复
- 闹钟具有无限重复响铃
- 年份为 2000 至 2099，带闰年修正

- 采用 BCD 格式，以减小软件开销
- 为长期电池工作进行了优化
- 32.768 kHz 时钟晶振的用户校准 / 带周期性自动调整的 32 kHz INTRC 频率
- 为长期电池工作进行了优化
- 小数秒同步
- 校准每月误差在  $\pm 2.64$  秒内
- 校准最高 260 ppm 的晶振误差
- 可定期唤醒外部器件，而无需 CPU 干预（外部电源控制）
- 电源控制输出用于控制外部电路
- 校准每 15 秒生效
- 可依靠以下任何时钟运行：
  - 32.768 kHz 的外部实时时钟 (RTC)
  - 内部 31.25 kHz LPRC 时钟
  - 50 Hz 或 60 Hz 外部输入

### 22.1 RTCC 源时钟

用户可选择 SOSC 晶振、LPRC 内部振荡器或外部 50 Hz/60 Hz 电源线输入作为 RTCC 模块的参考时钟。用户可以基于系统总体的需要，通过此方法使系统成本、精度和功耗之间达到平衡。

图 22-1: RTCC 框图



# PIC24FJ128GB204 系列

## 22.2 RTCC 模块寄存器

RTCC 模块寄存器分为三类：

- RTCC 控制寄存器
- RTCC 值寄存器
- 闹钟值寄存器

### 22.2.1 寄存器映射

要限制寄存器接口，则需通过相应的寄存器指针访问 RTCC 定时器和闹钟时间寄存器。RTCC 值寄存器窗口 (RTCVALH 和 RTCVALL) 使用 RTCPTRx 位 (RCFGCAL<9:8>) 来选择需要的定时器寄存器对 (见表 22-1)。

通过写 RTCVALH 字节，使 RTCPTR<1:0> 位 (RTCC 指针值) 逐次递减 1 直至 00。一旦其达到 00，在手动更改指针值前，可通过 RTCVALH 和 RTCVALL 来访问分钟和秒值。

表 22-1: RTCC 值寄存器映射

RTCPTR<1:0>	RTCC 值寄存器窗口	
	RTCVAL<15:8>	RTCVAL<7:0>
00	分钟数	秒数
01	星期	小时数
10	月	日
11	—	年

闹钟值寄存器窗口 (ALRMVALH 和 ALRMVALL) 使用 ALRMPTRx 位 (ALCFGRPT<9:8>) 来选择所需的闹钟寄存器对 (见表 22-2)。

通过写 ALRMVALH 字节，使闹钟指针值 (ALRMPTR<1:0> 位) 逐次递减 1 直至 00。一旦其达到 00，在手动更改指针值前，可通过 ALRMVALH 和 ALRMVALL 来访问 ALRMMIN 和 ALRMSEC 的值。

例 22-1: 将 RTCWREN 位置 1

```
asm volatile("push w7");
asm volatile("push w8");
asm volatile("disi #5");
asm volatile("mov #0x55, w7");
asm volatile("mov w7, _NVMKEY");
asm volatile("mov #0xAA, w8");
asm volatile("mov w8, _NVMKEY");
asm volatile("bset _RCFGCAL, #13"); //set the RTCWREN bit
asm volatile("pop w8");
asm volatile("pop w7");
```

表 22-2: 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 位内核不能区分 8 位和 16 位读操作，用户必须注意读 ALRMVALH 或 ALRMVALL 字节会递减 ALRMPTR<1:0> 的值。同样，读 RTCVALH 或 RTCVALL 字节会使 RTCPTR<1:0> 的值递减。

**注：** 这仅适用于读操作，而不适用于写操作。

### 22.2.2 写锁定

为了对任何 RTCC 定时器寄存器执行写操作，必须将 RTCWREN 位 (RCFGCAL<13>) 置 1 (见例 22-1)。

**注：** 为避免意外写入定时器，建议在除要特意执行写操作之外的任何时候都保持 RTCWREN 位 (RCFGCAL<13>) 清零。对于要使 RTCWREN 位置 1 的操作，由于在 55h/AA 序列和 RTCWREN 置 1 之间仅允许有一个指令周期的时间；因此，建议按照例 22-1 中的过程执行代码。

### 22.2.3 选择 RTCC 时钟源

使用 RTCPWC 寄存器中的 RTCLK<1:0> 位选择 RTCC 模块的时钟源。该位设置为 00 时，辅助振荡器 (SOSC) 用作参考时钟；该位为 01 时，LPRC 用作参考时钟。RTCLK<1:0> = 10 和 11 时，外部电源线 (50 Hz 和 60 Hz) 用作时钟源。

## 22.3 寄存器

### 22.3.1 RTCC 控制寄存器

寄存器 22-1: **RCFGCAL: RTCC 校准 / 配置寄存器 (1)**

R/W-0	U-0	R/W-0	R-0, HSC	R-0, HSC	R/W-0	R/W-0	R/W-0
RTCEN <sup>(2)</sup>	—	RTCWREN	RTCSYNC	HALFSEC <sup>(3)</sup>	RTCOE	RTCPTR1	RTCPTR0
bit 15						bit 8	

R/W-0							
CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
bit 7						bit 0	

<b>图注:</b>	HSC = 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 **RTCEN:** RTCC 使能位 <sup>(2)</sup>

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 寄存器可被用户写入  
0 = RTCVALH 和 RTCVALL 寄存器被锁定不允许用户写入

bit 12 **RTCSYNC:** RTCC 值寄存器读同步位

1 = 由于计满返回, RTCVALH、RTCVALL 和 ALCFGRPT 寄存器可能会在读取期间变化, 从而导致读到的数据无效。如果两次读取寄存器得到的数据相同, 则认为数据有效。  
0 = RTCVALH、RTCVALL 或 ALCFGRPT 寄存器可被读取, 而不用担心计满返回

bit 11 **HALFSEC:** 半秒状态位 <sup>(3)</sup>

1 = 一秒的后半周期  
0 = 一秒的前半周期

bit 10 **RTCOE:** RTCC 输出使能位

1 = 使能 RTCC 输出  
0 = 禁止 RTCC 输出

bit 9-8 **RTCPTR<1:0>:** RTCC 值寄存器窗口指针位

当读 RTCVALH 和 RTCVALL 寄存器时, 指向相应的 RTCC 值寄存器; 每次读或写 RTCVALH 时, RTCPTR<1:0> 的值都会减 1, 直至 00。

RTCVAL<15:8>:

11 = 保留  
10 = 月  
01 = 星期  
00 = 分钟数

RTCVAL<7:0>:

11 = 年  
10 = 日  
01 = 小时数  
00 = 秒数

注 1: RCFGCAL 寄存器仅受到 POR 影响。

2: 仅当 RTCWREN = 1 时才允许写入 RTCEN 位。

3: 该位是只读位。当写入 MINSEC 寄存器的低半部分时, 该位清 0。

# PIC24FJ128GB204 系列

---

## 寄存器 22-1: RCFGAL: RTCC 校准 / 配置寄存器<sup>(1)</sup> (续)

bit 7-0      **CAL<7:0>**: RTC 漂移校准位

01111111 = 最大正向调整; 每 15 秒增加 127 个 RTC 时钟脉冲

•

•

00000001 = 最小正向调整; 每 15 秒增加 1 个 RTC 时钟脉冲

00000000 = 无调整

11111111 = 最小负向调整; 每 15 秒减少 1 个 RTC 时钟脉冲

•

•

10000000 = 最大负向调整; 每 15 秒减少 128 个 RTC 时钟脉冲

- 注
- 1: RCFGAL 寄存器仅受到 POR 影响。
  - 2: 仅当 RTCWREN = 1 时才允许写入 RTCEN 位。
  - 3: 该位是只读位。当写入 MINSEC 寄存器的低半部分时, 该位清 0。

# PIC24FJ128GB204 系列

## 寄存器 22-2: RTCPWC: RTCC 电源控制寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PWCEN	PWCPOL	PWCPRE	PWSPRE	RTCLK1 <sup>(2)</sup>	RTCLK0 <sup>(2)</sup>	RTCOUT1	RTCOUT0
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     **PWCEN:** 电源控制使能位  
           1 = 使能电源控制  
           0 = 禁止电源控制
- bit 14     **PWCPOL:** 电源控制极性位  
           1 = 电源控制输出高电平有效  
           0 = 电源控制输出低电平有效
- bit 13     **PWCPRE:** 电源控制 / 稳定预分频比位  
           1 = PWC 稳定窗口时钟为 RTCC 时钟源的 2 分频  
           0 = PWC 稳定窗口时钟为 RTCC 时钟源的 1 分频
- bit 12     **PWSPRE:** 电源控制采样预分频比位  
           1 = PWC 采样窗口时钟为 RTCC 时钟源的 2 分频  
           0 = PWC 采样窗口时钟为 RTCC 时钟源的 1 分频
- bit 11-10   **RTCLK<1:0>:** RTCC 时钟源选择位<sup>(2)</sup>  
           11 = 外部电源线电源 (60 Hz)  
           10 = 外部电源线电源 (50 Hz)  
           01 = 内部 LPRC 振荡器  
           00 = 外部辅助振荡器 (SOSC)
- bit 9-8     **RTCOUT<1:0>:** RTCC 输出源选择位  
           11 = 电源控制  
           10 = RTCC 时钟  
           01 = RTCC 秒时钟  
           00 = RTCC 闹钟脉冲
- bit 7-0     **未实现:** 读为 0

注 1: RTCPWC 寄存器仅受到 POR 影响。

2: 当新值被写入这些寄存器位时, 也应被写入 MINSEC 寄存器的低半部分以正确重新设置 RTCC 中的时钟预分频比。

# PIC24FJ128GB204 系列

## 寄存器 22-3: 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 = 使能闹钟 (只要 ARPT<7:0> = 00h 且 CHIME = 0, 就会在闹钟事件后自动清零)  
 0 = 禁止闹钟
- bit 14        **CHIME:** 响铃使能位  
 1 = 使能响铃; 允许 ARPT<7:0> 位从 00h 返回到 FFh  
 0 = 禁止响铃; ARPT<7:0> 位到达 00h 时停止
- 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 = 每年一次 (除配置为 2 月 29 日, 亦即每 4 年一次外)  
 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  
 当读 ALRMVALH 和 ALRMVALL 寄存器时指向相应的闹钟值寄存器。每次读或写 ALRMVALH 时, ALRMPTR<1:0> 的值都会减 1, 直至 00。  
ALRMVAL<15:8>:  
 00 = ALRMMIN  
 01 = ALRMWD  
 10 = ALRMMNTH  
 11 = PWCSTAB  
ALRMVAL<7:0>:  
 00 = ALRMSEC  
 01 = ALRMHR  
 10 = ALRMDAY  
 11 = PWCSAMP
- bit 7-0        **ARPT<7:0>:** 闹钟重复计数器值位  
 11111111 = 闹钟将重复 255 次  
 .  
 .  
 .  
 00000000 = 闹钟不重复  
 每发生一次闹钟事件, 计数器就减 1; 只有在 CHIME = 1 的情况下, 计数器才会在递减到 00h 后返回 FFh。

# PIC24FJ128GB204 系列

## 22.3.2 RTCVAL 寄存器映射

### 寄存器 22-4: YEAR: 年值寄存器<sup>(1)</sup>

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>:** 年的十位数字的二 - 十进制码 (Binary Coded Decimal, BCD) 值位  
包含从 0 到 9 的值。
- bit 3-0      **YRONE<3:0>:** 年的个位数字的二 - 十进制码值位  
包含从 0 到 9 的值。

注 1: 仅当 RTCWREN = 1 时才允许写入 YEAR 寄存器。

### 寄存器 22-5: MTHDY: 月和日值寄存器<sup>(1)</sup>

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MHTTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 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      **MHTTEN0:** 月的十位数字的二 - 十进制码值位  
包含为 0 或 1 的值。
- bit 11-8    **MTHONE<3:0>:** 月的个位数字的二 - 十进制码值位  
包含从 0 到 9 的值。
- bit 7-6      **未实现:** 读为 0
- bit 5-4      **DAYTEN<1:0>:** 日的十位数字的二 - 十进制码值位  
包含从 0 到 3 的值。
- bit 3-0      **DAYONE<3:0>:** 日的个位数字的二 - 十进制码值位  
包含从 0 到 9 的值。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

# PIC24FJ128GB204 系列

## 寄存器 22-6: WKDYHR: 星期和小时值寄存器<sup>(1)</sup>

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>:** 星期数字的二 - 十进制码值位  
包含从 0 到 6 的值。
- bit 7-6        **未实现:** 读为 0
- bit 5-4        **HRTEN<1:0>:** 小时的十位数字的二 - 十进制码值位  
包含从 0 到 2 的值。
- bit 3-0        **HRONE<3:0>:** 小时的个位数字的二 - 十进制码值位  
包含从 0 到 9 的值。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

## 寄存器 22-7: 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	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>:** 分钟的十位数字的二 - 十进制码值位  
包含从 0 到 5 的值。
- bit 11-8     **MINONE<3:0>:** 分钟的个位数字的二 - 十进制码值位  
包含从 0 到 9 的值。
- bit 7         **未实现:** 读为 0
- bit 6-4      **SECTEN<2:0>:** 秒的十位数字的二 - 十进制码值位  
包含从 0 到 5 的值。
- bit 3-0      **SECONE<3:0>:** 秒的个位数字的二 - 十进制码值位  
包含从 0 到 9 的值。

## 22.3.3 ALRMVAL 寄存器映射

### 寄存器 22-8: ALMTHDY: 闹钟月和日值寄存器<sup>(1)</sup>

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MHTTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 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        **MHTTEN0:** 月的十位数字的二 - 十进制码值位  
包含为 0 或 1 的值。
- bit 11-8     **MTHONE<3:0>:** 月的个位数字的二 - 十进制码值位  
包含从 0 到 9 的值。
- bit 7-6      **未实现:** 读为 0
- bit 5-4      **DAYTEN<1:0>:** 日的十位数字的二 - 十进制码值位  
包含从 0 到 3 的值。
- bit 3-0      **DAYONE<3:0>:** 日的个位数字的二 - 十进制码值位  
包含从 0 到 9 的值。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

### 寄存器 22-9: ALWDHR: 闹钟星期和小时值寄存器<sup>(1)</sup>

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>:** 星期数字的二 - 十进制码值位  
包含从 0 到 6 的值。
- bit 7-6      **未实现:** 读为 0
- bit 5-4      **HRTEN<1:0>:** 小时的十位数字的二 - 十进制码值位  
包含从 0 到 2 的值。
- bit 3-0      **HRONE<3:0>:** 小时的个位数字的二 - 十进制码值位  
包含从 0 到 9 的值。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

# PIC24FJ128GB204 系列

## 寄存器 22-10: 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>:** 分钟的十位数字的二 - 十进制码值位  
包含从 0 到 5 的值。
- bit 11-8     **MINONE<3:0>:** 分钟的个位数字的二 - 十进制码值位  
包含从 0 到 9 的值。
- bit 7        **未实现:** 读为 0
- bit 6-4      **SECTEN<2:0>:** 秒的十位数字的二 - 十进制码值位  
包含从 0 到 5 的值。
- bit 3-0      **SECONE<3:0>:** 秒的个位数字的二 - 十进制码值位  
包含从 0 到 9 的值。

# PIC24FJ128GB204 系列

寄存器 22-11: **RTCCSWT: RTCC 电源控制和采样窗口定时器寄存器** <sup>(1)</sup>

R/W-x							
PWCSTAB7	PWCSTAB6	PWCSTAB5	PWCSTAB4	PWCSTAB3	PWCSTAB2	PWCSTAB1	PWCSTAB0
bit 15							bit 8

R/W-x							
PWCSAMP7 <sup>(2)</sup>	PWCSAMP6 <sup>(2)</sup>	PWCSAMP5 <sup>(2)</sup>	PWCSAMP4 <sup>(2)</sup>	PWCSAMP3 <sup>(2)</sup>	PWCSAMP2 <sup>(2)</sup>	PWCSAMP1 <sup>(2)</sup>	PWCSAMP0 <sup>(2)</sup>
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

**PWCSTAB<7:0>:** 电源控制稳定窗口定时器位

11111111 = 稳定窗口为 255 个 TPWCCLK 时钟周期

11111110 = 稳定窗口为 254 个 TPWCCLK 时钟周期

•  
•  
•

00000001 = 稳定窗口为 1 个 TPWCCLK 时钟周期

00000000 = 无稳定窗口; 触发闹钟事件时, 采样窗口启动

bit 7-0

**PWCSAMP<7:0>:** 电源控制采样窗口定时器位 <sup>(2)</sup>

11111111 = 始终使能采样窗口, 即使 PWCEN = 0

11111110 = 采样窗口为 254 个 TPWCCLK 时钟周期

•  
•  
•

00000001 = 采样窗口为 1 个 TPWCCLK 时钟周期

00000000 = 无采样窗口

注

**1:** 仅当 RTCWREN = 1 时才允许写入该寄存器。

**2:** 当稳定窗口定时器超时时, 始终启动采样窗口, 当其初始值为 00h 时除外。

# PIC24FJ128GB204 系列

## 22.4 校准

实时晶振输入可使用周期性自动调整功能来校准。正确校准后，RTCC 可实现每月小于 3 秒的误差。这可通过确定误差时钟脉冲数并将其存储到 RCFGAL 寄存器的低半字节完成。装入到 RCFGAL 低半字节的 8 位有符号值与 4 相乘，然后每分钟与 RTCC 定时器的值相加或相减一次。请参见以下步骤校准 RTCC：

1. 使用器件上的其他定时器资源，用户必须确定 32.768 kHz 晶振的误差。
2. 一旦误差已知，必须将其转换为每分钟的误差时钟脉冲数。
3. a) 如果振荡器频率高于理想值（步骤 2 中的计算结果为负），那么 RCFGAL 寄存器的值就必须为负。这样每分钟会在定时器计数值上减去指定的时钟脉冲数。  
b) 如果振荡器频率低于理想值（步骤 2 中的计算结果为正），那么 RCFGAL 寄存器的值就必须为正。这样每分钟会在定时器计数值上加上指定的时钟脉冲数。

### 公式 22-1:

$$\begin{aligned} & (\text{理想频率} \uparrow - \text{测得频率}) * 60 = \text{每分钟的时钟数} \\ & \uparrow \text{理想频率} = 32,768 \text{ Hz} \end{aligned}$$

仅当定时器关闭或者紧跟在秒脉冲的上升沿 (SECONDS = 00、15、30 或 45 时除外) 后，才可写入 RCFGAL 寄存器的低半部分。这是因为 RTCC 每 15 秒会自动调整一次。

**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。

## 22.5 闹钟

- 闹钟的时间间隔可配置为从半秒到一年
- 使用 ALRMEN 位 (ALCFGRPT<15>) 使能
- 可选择一次性闹钟和重复闹钟

### 22.5.1 配置闹钟

使用 ALRMEN 位使能闹钟功能。闹钟事件发生后该位清零。只有在 ALRMEN = 0 时才允许写入 ALRMVAL。

如图 22-2 所示，可通过 AMASKx 位 (ALCFGRPT<13:10>) 配置闹钟时间间隔。这些位确定要触发闹钟，闹钟的哪些位、多少位必须和时钟值匹配。

也可配置闹钟根据预配置的时间间隔重复。一旦使能了闹钟，其重复次数就会被存储到 ARPT<7:0> 位 (ALCFGRPT<7:0>)。当 ARPTx 位的值等于 00h 且 CHIME 位 (ALCFGRPT<14>) 清零时，禁止重复功能，仅发生单次闹钟。通过将 FFh 装入 ARPT<7:0>，可使闹钟重复最多 255 次。

每发生一次闹钟事件，ARPTx 位的值都会减 1。一旦该值达到 00h，就会发生最后一次闹钟事件，然后 ALRMEN 位自动清零，关闭闹钟。

如果 CHIME 位 = 1，闹钟可无限次重复发生。在这种情况下，当 ARPTx 位的值达到 00h 时，不会禁止闹钟，而是返回 FFh 并继续无限次地计数。

### 22.5.2 闹钟中断

每一次闹钟事件都会产生一个中断。此外，还将输出一个频率为闹钟频率一半的闹钟脉冲序列。该输出与 RTCC 时钟完全同步，且可用作其他外设的触发时钟。

**注：** 当使能闹钟 (ALRMEN = 1) 时，更改除 RCFGAL 和 ALCFGRPT 寄存器外的任何寄存器和 CHIME 位都将导致错误的闹钟事件，从而引起错误的闹钟中断。为避免错误闹钟事件，只能在禁止闹钟 (ALRMEN = 0) 时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 22-2: 闹钟掩码设置

闹钟掩码设置 (AMASK<3:0>)	星期值	月	日	小时	分钟	秒
0000 - 每半秒 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"/> 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"/> 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"/> s <input type="checkbox"/> s
0110 - 每日	<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"/> m <input type="checkbox"/> m	<input type="checkbox"/> s <input type="checkbox"/> s
0111 - 每周	<input type="checkbox"/> d	<input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> h <input type="checkbox"/> h	<input type="checkbox"/> m <input type="checkbox"/> m	<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"/> h <input type="checkbox"/> h	<input type="checkbox"/> m <input type="checkbox"/> m	<input type="checkbox"/> s <input type="checkbox"/> s
1001 - 每年 <sup>(1)</sup>	<input type="checkbox"/>	<input type="checkbox"/> m <input type="checkbox"/> m	<input type="checkbox"/> d <input type="checkbox"/> d	<input type="checkbox"/> h <input type="checkbox"/> h	<input type="checkbox"/> m <input type="checkbox"/> m	<input type="checkbox"/> s <input type="checkbox"/> s

**注 1:** 每年，除非配置为 2 月 29 日。

## 22.6 电源控制

RTCC 包含电源控制功能，使器件可定期唤醒外部器件，在从该器件采样唤醒事件前等待器件稳定，然后关闭外部器件。这可完全由 RTCC 自动完成，而无需从当前低功耗模式（休眠模式和深度休眠模式等）唤醒。

要使用该功能：

1. 使能 RTCC（如果 RTCEN = 1）。
2. 将 PWCEN 位（RTCPWC<15>）置 1。
3. 配置 RTCC 引脚以驱动 PWC 控制信号（RTCOE = 1 和 RTCOUT<1:0> = 11）。

可使用 PWCPOL 位（RTCPWC<14>）选择 PWC 控制信号的极性。可将低电平有效或高电平有效信号用于相应的外部开关来打开或关闭一个或多个外部器件的电

源。低电平有效设置也可与 RTCC 引脚的漏极开路设置结合使用。可以直接驱动外部器件的接地引脚（相应的外部 VDD 上拉器件），而无需外部开关。最后，应将 CHIME 位置 1 以定期使能 PWC。

## 22.7 RTCC VBAT 操作

当 VDD 引脚上有功率损耗时，RTCC 可工作在 VBAT 模式下。如果为 VBAT 引脚供电（通常连接到电池），RTCC 将继续工作。

**注：** 如果未使用 VBAT 模式（未连接至电池），建议将 VBAT 引脚连接至 VDD。

# PIC24FJ128GB204 系列

---

注:

## 23.0 加密引擎

**注：** 本数据手册总结了 PIC24FJ128GB204 系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见 Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 上提供的《dsPIC33/PIC24 系列参考手册》中的“加密引擎” (DS70005133)。

加密引擎提供一组新的数据安全选项。该引擎使用自己的独立状态机，可独立于 CPU 执行 NIS 标准数据加密和解密。从而无需担心在加强应用程序的安全性时，加密和解密额外需要的过多 CPU 或程序存储器开销。

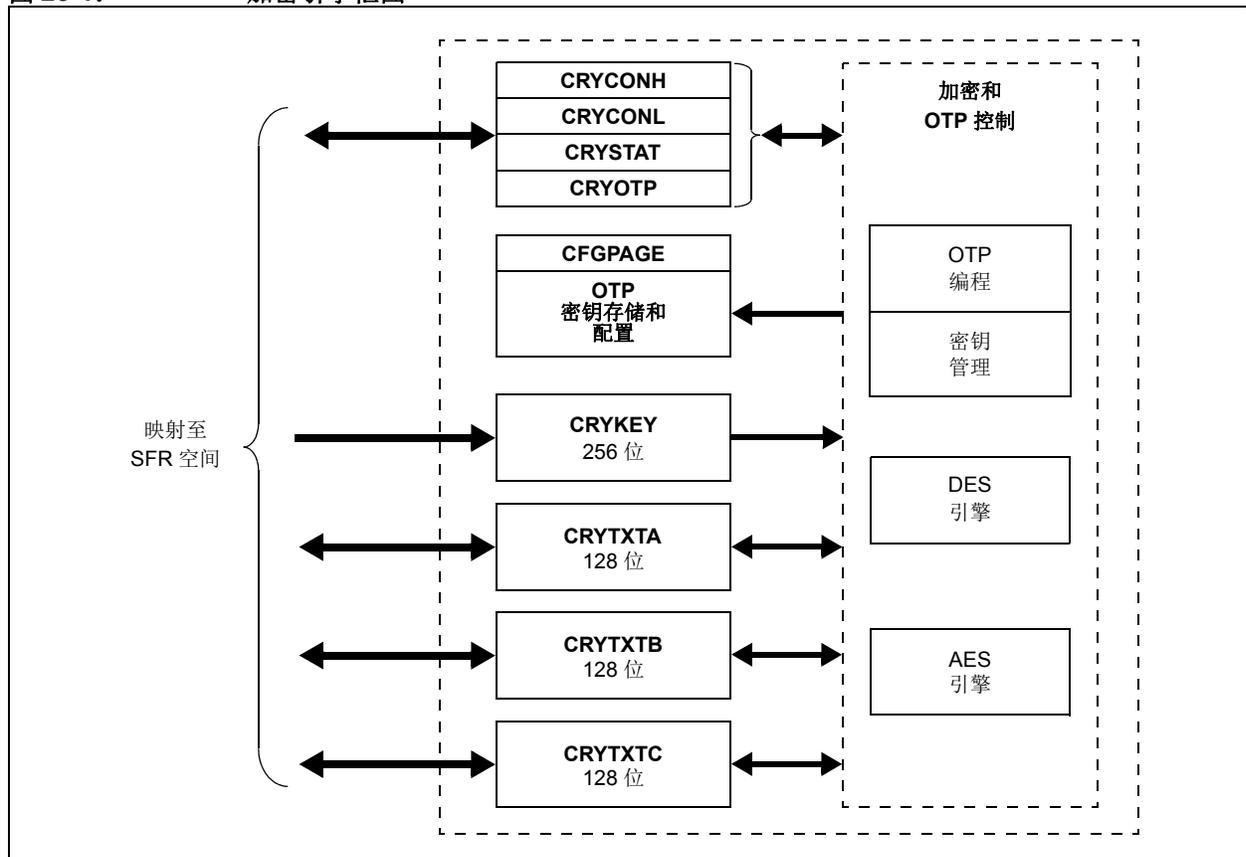
加密引擎的主要特性包括：

- 存储器映射的 128 位和 256 位存储空间，用于加密 / 解密数据
- 多个密钥存储、选择和管理选项

- 支持内部现场保护
- 会话密钥加密和加载
- 半双工操作
- DES 和三重 DES (Triple DES, 3DES) 加密和解密 (64 位块大小)：
  - 支持 64 位密钥和 2 密钥或 3 密钥三重 DES
- AES 加密和解密 (128 位块大小)：
  - 支持 128、192 或 256 位密钥大小
- 支持 DES 和 AES 标准的 ECB、CBC、CFB、OFB 和 CTR 模式
- 可编程安全密钥存储：
  - 512 位 OTP 阵列用于密钥存储，不可从其他存储空间读取
  - 32 位配置页
  - 简单模块内编程接口
  - 支持密钥加密密钥 (Key Encryption Key, KEK)
- 支持真随机数和伪随机数发生 (PRNG) (符合 NIST SP800-90)

加密引擎的简化框图如图 23-1 所示。

图 23-1: 加密引擎框图



# PIC24FJ128GB204 系列

## 23.1 数据寄存器空间

有 4 个寄存器空间用于加密数据和密钥存储：

- CRYTXTA
- CRYXTB
- CRYTTC
- CRYKEY

虽然所有这些数据空间均映射至 SFR 空间，但实际实现为 128 位或 256 位宽阵列，而不是 16 位宽数据寄存器组。和 SFR 空间中的任何其他寄存器一样，对这些阵列的读和写操作均自动执行。

CRYTXTA 至 CRYTTC 为 128 位宽空间，用于对加密引擎写入 / 读取数据。另外，它们还用于存储加密 / 解密操作的中间结果。当模块在执行操作 (CRYGO = 1) 时，不能写入这些寄存器。

CRYTXTA 和 CRYXTB 通常充当加密 / 解密过程的输入。

CRYTXTA 通常包含要加密的初始明文或要解密的初始密文。根据操作模式，CRYXTB 可能包含密文输出或中间密文数据。在某些操作中它还可充当长度可编程的计数器。

CRYTTC 主要用于存储加密 / 解密操作的最终输出。它还用作输入寄存器，包含要编程到安全 OTP 阵列中的数据。

CRYKEY 为 256 位宽的空间，用于存储所选操作的加密密钥。可从 SFR 空间和安全 OTP 阵列对其进行写操作。虽然 CRYKEY 映射到 SFR 空间，但为只写存储区域；此处存储的任何数据，无论其来源为何，均无法通过任何运行时操作读回。该特性有助于确保所有密钥数据的安全。

## 23.2 操作模式

加密引擎支持以下操作模式（通过 OPMOD<3:0> 位确定）：

- 块加密
- 块解密
- AES 解密密钥扩展
- 随机数发生
- 会话密钥生成
- 会话密钥加密
- 会话密钥加载

OPMODx 位可在 CRYON 置 1 时发生更改。它们仅应在加密操作未完成 (CRYGO = 0) 时发生更改。

选择加密操作和相应的有效密钥配置后，通过将 CRYGO 位置 1 来执行该操作。当操作完成时，该位由硬件自动清零。也可用软件手动清零 CRYGO 位，这可

使任何正在进行中的操作立即终止。用软件清零该位还会将 CRYABRT 位 (CRYSTAT<5>) 置 1。

对于大多数操作，仅当 OTP 操作未在执行且未出现任何其他错误情况时，才可将 CRYGO 置 1。CRYREAD、CRYWR、CRYABRT、ROLLOVR、MODFAIL 和 KEYFAIL 必须全为 0。

同时置 1 CRYWR 和 CRYGO 将不会启动 OTP 编程操作或任何其他操作。模块禁止 (CRYON = 0) 时将 CRYGO 置 1 也无任何作用。

## 23.3 使能引擎

通过将 CRYON 位置 1 使能加密引擎。清零该位将同时禁止 DES 和 AES 引擎，并导致以下寄存器位保持复位状态：

- CRYGO (CRYCONL<8>)
- TXTABSY (CRYSTAT<6>)
- CRYWR (CRYOTP<0>)

CRYON = 0 时，可读取和写入所有其他寄存器位和寄存器。

## 23.4 休眠和空闲模式下的操作

### 23.4.1 休眠模式下的操作

每当器件进入任何休眠或深度休眠模式时，所有操作引擎状态机都会复位。通过丢弃任何可用于破解密钥的中间文本，该特性有助于保存完整性或保护被加密或解密的所有数据。

进入休眠模式时，正在进行的任何 OTP 编程操作也将停止。这可能导致永久丧失一个存储单元或可能使用整个安全 OTP 阵列，具体取决于所编程的内容。建议用户仅在禁止进入节能模式时执行 OTP 编程。

<b>注：</b>	OTP 编程错误，无论来源为何，均不可恢复。用户应确保禁止编程操作的所有可见中断，包括器件中断和进入节能模式。
-----------	---

### 23.4.2 空闲模式下的操作

当 CRYSIDL 位 (CRYCONL<13>) 为 0 时，器件进入空闲模式后，引擎将继续任何正在进行的操作，而不会中断。

当 CRYSIDL 为 1 时，模块的行为类似于在休眠模式下。

## 23.5 具体加密操作

本节分步详细介绍了加密引擎提供的每种操作类型。

### 23.5.1 加密数据

1. 如果 CRYON 位尚未置 1，请将其置 1。
2. 按需配置 CPHRSEL、CPHRMODx、KEYMODx 和 KEYSRCx 位以选择适当的模式和密钥长度。
3. 将 OPMOD<3:0> 设置为 0000。
4. 如果正在使用软件密钥，请将其写入 CRYKEY 寄存器。只需写 CRYKEY 的低  $n$  位作为密钥长度  $n$ ，所有未使用的 CRYKEY 位均忽略。
5. 读 KEYFAIL 位。该位为 1，表示选择了一个非法配置，将不会执行加密操作。
6. 将要加密的数据写入相应的 CRYTXT 寄存器。对于单个 DES 加密操作，只需写入低 64 位。然而，对于小于块大小（DES 为 64 位，AES 为 128 位）的数据，由软件对块内的高位进行适当填补。
7. 将 CRYGO 置 1。
8. 在 ECB 和 CBC 模式下，将 FREEIE 位（CRYCONL<10>）置 1 以允许可选 CRYTXTA 中断，指示可加载下一个明文块。
9. 查询 CRYGO 位直至其清零，或等待 CRYDNIF 模块中断（DONEIE 必须置 1）。如果允许其他加密引擎中断，将需要查询 CRYGO 位以验证中断源。
10. 从相应的 CRYTXT 寄存器读加密块。
11. 重复步骤 5 到 8 以使用相同的密钥加密消息中的后续块。

### 23.5.2 解密数据

1. 如果 CRYON 位尚未置 1，请将其置 1。
2. 按需配置 CPHRSEL、CPHRMODx、KEYMODx 和 KEYSRCx 位以选择适当的模式和密钥长度。
3. 将 OPMOD<3:0> 设置为 0001。
4. 如果正在使用软件密钥，请写入 CRYKEY 寄存器。只需写 CRYKEY 的低  $n$  位作为密钥长度  $n$ ，所有未使用的 CRYKEY 位均忽略。
5. 如果正在执行 AES-ECB 或 AES-CBC 模式解密，必须首先执行 AES 解密密钥扩展操作。
6. 读 KEYFAIL 状态位。该位为 1，表示选择了一个非法配置，将不会执行解密操作。
7. 将要解密的数据写入相应的文本 / 数据寄存器。对于 DES 解密操作，只需写入 CRYTXTB 的低 64 位。
8. 将 CRYGO 置 1。
9. 如果这是复位后的第一个解密操作，或者完成上一个解密操作后执行了密钥存储器编程操作，或者 KEYMODx 或 KEYSRCx 域发生更改，引擎将执行新的密钥扩展操作。这将导致需要额外的时钟周期来执行解密操作，但无论如何对应用来说，该操作都是透明的（即，仅当密钥扩展和解密操作已完成，CRYGO 位才会清零）。
10. 在 ECB 和 CBC 模式下，将 FREEIE 位（CRYCONL<10>）置 1 以允许可选 CRYTXTA 中断，指示可加载下一个明文块。
11. 查询 CRYGO 位直至其清零，或等待 CRYDNIF 模块中断（DONEIE 必须置 1）。如果允许其他加密引擎中断，将需要查询 CRYGO 位以验证中断源。
12. 从相应的文本 / 数据寄存器读取解密块。
13. 重复步骤 6 到 10 以使用相同的密钥解密消息中的后续块。

# PIC24FJ128GB204 系列

## 23.5.3 加密会话密钥

**注：** ECB 和 CBC 模式限制为仅 128 位会话密钥。

1. 如果 CRYON 位尚未置 1，请将其置 1。
2. 如果尚未将 SKEYEN 位设定为 1，请设定。

**注：** 将 SKEYEN 永久置 1 使 Key #1 仅用作密钥加密密钥。该操作后，它不可用于其他加密或解密操作。

3. 将 OPMOD<3:0> 设置为 1110。
4. 按需配置 CPHRSEL、CPHRMOD<2:0> 和 KEYMOD<1:0> 寄存器位域，将 SKEYSEL 设置为 0。
5. 读 KEYFAIL 状态位。该位为 1，表示选择了一个非法配置，将不会执行加密操作。
6. 将软件生成的会话密钥写入 CRYKEY 寄存器，或生成随机密钥并写入 CRYKEY 寄存器。只需写入 CRYKEY 的低  $n$  位作为密钥长度  $n$ ，所有未使用的密钥位均忽略。
7. 将 CRYGO 置 1。查询该位直至其由硬件清零；或者，将 DONEIE 位 (CRYCONL<11>) 置 1 以在加密完成时产生中断。
8. 从相应的 CRYTXT 寄存器中读取加密的会话密钥。
9. 对于总长度大于 128 位的密钥，将 SKEYSEL 设置为 1 并重复步骤 6 和 7。
10. 将 KEYSRC<3:0> 设置为 0000 以使用会话密钥来加密数据。

## 23.5.4 接收会话密钥

**注：** ECB 和 CBC 模式限制为仅 128 位会话密钥。

1. 如果 CRYON 位尚未置 1，请将其置 1。
2. 如果尚未将 SKEYEN 位设定为 1，请设定。

**注：** 将 SKEYEN 永久置 1 使 Key #1 仅用作密钥加密密钥。该操作后，它不可用于其他加密或解密操作。它还会永久禁止软件将会话密钥解密到 CRYXTA 寄存器，从而破坏程序安全性（即，软件可读取未加密的密钥）。

3. 将 OPMOD<3:0> 设置为 1111。
4. 按需配置 CPHRSEL、CPHRMOD<2:0> 和 KEYMOD<1:0> 寄存器位域，将 SKEYSEL 设置为 0。
5. 读 KEYFAIL 状态位。该位为 1，表示选择了一个非法配置，将不会执行加密操作。
6. 将接收到的加密会话密钥写入到相应的 CRYTXT 寄存器。
7. 将 CRYGO 置 1。查询该位直至其由硬件清零；或者，将 DONEIE 位 (CRYCONL<11>) 置 1 以在此过程完成时产生中断。
8. 对于总长度大于 128 位的密钥，将 SKEYSEL 设置为 1 并重复步骤 6 和 7。
9. 将 KEYSRC<3:0> 设置为 0000 以使用新生成的会话密钥来加密和解密数据。

## 23.5.5 生成伪随机数 (PRN)

对于需要伪随机数 (PRN) 的操作, 可以采用 NIST SP800-90 中概述的方法有效利用加密引擎。该方法使用 CTR 模式的 AES 算法以最少的 CPU 开销创建 PRN。以此方式生成的 PRN 可用于加密或主机应用可能需要的任何其他目的。

可从任何方便用户应用的源获取随机数作为初始种子。如果可能, 应使用不确定性随机数源。

**注:** 当禁止软件密钥 (SWKYDIS = 1) 时, 不可生成 PRN。

要执行初始重新设置种子操作, 以及在重新设置种子时间间隔到期后执行后续重新设置种子, 请执行下列操作:

1. 将随机数 (128 位) 存储在 CRYTXTA 中。
2. 仅对于初始生成, 使用密钥值 0h (128 位) 以及计数器值 0h。
3. 将引擎配置为 AES 加密、CTR 模式 (OPMOD<3:0> = 0000, CPHRSEL = 1, CPHMOD<2:0> = 100)。
4. 通过将 CRYGO 置 1, 执行加密操作。
5. 将 CRYTXXTC 中的结果移入 RAM。这是新密钥值 (NEW\_KEY)。
6. 将另一个随机数 (128 位) 存储在 CRYTXTA 中。
7. 按照步骤 3, 将模块配置为加密。
8. 通过将 CRYGO 置 1, 执行加密操作。
9. 将该值存储在 RAM 中。这是新计数器值 (NEW\_CTR)。
10. 对于后续重新设置种子操作, 使用 NEW\_KEY 和 NEW\_CTR 获取起始密钥和计数器值。

要生成伪随机数, 请执行下列操作:

1. 将 NEW\_KEY 值从 RAM 装入 CRYKEY。
2. 将 NEW\_CTR 值从 RAM 装入 CRYXTB。
3. 将 0h (128 位) 装入 CRYXTA。
4. 将引擎配置为 AES 加密、CTR 模式 (OPMOD<3:0> = 0000, CPHRSEL = 1, CPHMOD<2:0> = 100)。
5. 通过将 CRYGO 置 1, 执行加密操作。
6. 将 CRYTXXTC 中生成的 PRN (PRNG\_VALUE) 复制到 RAM。
7. 重复加密操作。
8. 将本轮的 CRYTXXTC 值存储为 NEW\_KEY 的新值。
9. 重复加密操作。
10. 将本轮的 CRYTXXTC 值存储为 NEW\_CTR 的新值。

可重复本过程生成后续 PRN, 直至重新设置种子时间间隔到期。此时, 使用 NEW\_KEY 和 NEW\_CTR 的存储值执行重新设置种子操作。

## 23.5.6 生成随机数

1. 使能加密模式 (CRYON (CRYCONL<15>) = 1)。
2. 设置 OPMOD<3:0> 位为 1010。
3. 将 CRYGO 位 (CRYCONL<8>) 设置为 1, 启动请求。
4. 等待 CRYGO 位被硬件清 0。
5. 从 CRYTXTA 寄存器读取随机数。

## 23.5.7 测试密钥源配置

可始终通过写相应的寄存器位, 然后读取 KEYFAIL 寄存器位来测试密钥源配置的有效性。执行该检查无需启动任何操作; 甚至不需要使能模块。

# PIC24FJ128GB204 系列

## 23.5.8 编程 CFGPAGE (PAGE 0) 配置位

1. 如果 CRYON 位尚未置 1, 请将其置 1。将 KEYPG<3:0> 设置为 0000。
2. 读取 PGMFAIL 状态位。该位为 1, 表示选择了一个非法配置, 将不会执行编程操作。
3. 将要编程到配置页的数据写入 CRYTXTC<31:0>。任何置 1 的位将被永久编程, 而任何清 0 的位将不会被编程, 并可在以后编程。
4. 将 CRYWR 位置 1。查询该位直至其清零; 或者, 将 OTPIE 位 (CRYOTP<6>) 置 1 以允许可选 OTP 完成中断。
5. 完成所有编程后, 将 CRYREAD 位置 1 以从片上存储器重新加载值。必须执行一个读操作来完成编程。

**注:** CRYREAD 位置 1 时, 请勿清零 CRYON 位; 这将导致不完整的读操作和不可用的密钥数据。要恢复, 请将 CRYON 和 CRYREAD 置 1 并允许读操作完全完成。

6. 查询 CRYREAD 位直至其清零; 或者, 将 OTPIE 位 (CRYOTP<6>) 置 1 以允许可选 OTP 完成中断。
7. 对于生产编程, 可将 TSTPGM 位置 1 以指示编程操作成功完成。TSTPGM 置 1 时, PGMST 位 (CRYOTP<7>) 也将置 1, 允许用户通过对 OTP 阵列执行读操作来查看该阵列的状态。

**注:** 如果在 OTP 编程期间器件进入休眠模式, OTP 阵列的内容可能会被破坏。这不是可恢复的错误。用户必须确保在执行 OTP 编程前禁止了进入节能模式。

## 23.5.9 编程密钥

1. 如果 CRYON 位尚未置 1, 请将其置 1。
2. 将 KEYPG<3:0> 配置为您要编程的页。
3. 读取 PGMFAIL 状态位。该位为 1, 表示选择了一个非法配置, 将不会执行编程操作。
4. 将要编程到配置页的数据写入 CRYTXTC<63:0>。任何置 1 的位将被永久编程, 而任何清 0 的位将不会被编程, 并可在以后编程。
5. 将 CRYWR 位置 1。查询该位直至其清零; 或者, 将 OTPIE 位 (CRYOTP<6>) 置 1 以允许可选 OTP 完成中断。
6. 针对每个要编程的 OTP 阵列页, 重复步骤 2 到 5。
7. 完成所有编程后, 将 CRYREAD 位置 1 以从片上存储器重新加载值。必须执行一个读操作来完成编程。

**注:** CRYREAD 位置 1 时, 请勿清零 CRYON 位; 这将导致不完整的读操作和不可用的密钥数据。要恢复, 请将 CRYON 和 CRYREAD 置 1 并允许读操作完全完成。

8. 查询 CRYREAD 位直至其清零; 或者, 将 OTPIE 位 (CRYOTP<6>) 置 1 以允许可选 OTP 完成中断。
9. 对于生产编程, 可将 TSTPGM 位置 1 以指示编程操作成功完成。TSTPGM 置 1 时, PGMST 位 (CRYOTP<7>) 也将置 1, 允许用户通过对 OTP 阵列执行读操作来查看该阵列的状态。

**注:** 如果在 OTP 编程期间器件进入休眠模式, OTP 阵列的内容可能会被破坏。这不是可恢复的错误。用户必须确保在执行 OTP 编程前禁止了进入节能模式。

## 23.5.10 验证编程的密钥

为了保持密钥安全性, 安全 OTP 阵列不支持以任何操作模式将其数据读回任何用户可访问的存储空间。因此, 没有方法可直接验证编程的数据。验证数据已正确编程的唯一方法是: 针对每个编程的密钥, 对已知的明文 / 密文对执行加密操作。

## 23.6 控制寄存器

寄存器 23-1: **CRYCONL**: 加密控制寄存器低位字

R/W-0	U-0	R/W-0	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	U-0	R/W-0, HC <sup>(1)</sup>
CRYON	—	CRYSIDL <sup>(3)</sup>	ROLLIE	DONEIE	FREEIE	—	CRYGO
bit 15							bit 8

R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>				
OPMOD3 <sup>(2)</sup>	OPMOD2 <sup>(2)</sup>	OPMOD1 <sup>(2)</sup>	OPMOD0 <sup>(2)</sup>	CPHRSEL <sup>(2)</sup>	CPHRMOD2 <sup>(2)</sup>	CPHRMOD1 <sup>(2)</sup>	CPHRMOD0 <sup>(2)</sup>
bit 7							bit 0

<b>图注:</b>	HC = 可由硬件清零的位
R = 可读位	W = 可写位
-n = POR 时的值	1 = 置 1
	0 = 清零
	x = 未知
	U = 未实现位, 读为 0

- bit 15 **CRYON**: 加密使能位  
1 = 使能模块  
0 = 禁止模块
- bit 14 **未实现**: 读为 0
- bit 13 **CRYSIDL**: 加密空闲模式停止控制位<sup>(3)</sup>  
1 = 在空闲模式下停止模块操作  
0 = 在空闲模式下继续模块操作
- bit 12 **ROLLIE**: CRYTXTB 计满返回中断允许位<sup>(1)</sup>  
1 = 当 CRYTXTB 的计数器部分计满返回到 0 时发生中断事件  
0 = 当 CRYTXTB 的计数器部分计满返回到 0 时不发生中断事件
- bit 11 **DONEIE**: 操作完成中断允许位<sup>(1)</sup>  
1 = 当前加密操作完成时发生中断事件  
0 = 当前加密操作完成时不发生中断事件; 软件必须查询 CRYGO 或 CRYBSY 位以确定当前加密操作何时完成
- bit 10 **FREEIE**: 输入文本中断允许位<sup>(1)</sup>  
1 = 当前加密操作期间使用输入文本 (明文或密文) 时发生中断事件  
0 = 使用输入文本时不发生中断事件
- bit 9 **未实现**: 读为 0
- bit 8 **CRYGO**: 加密引擎启动位<sup>(1)</sup>  
1 = 启动 OPMOD<3:0> 指定的操作 (操作完成时自动清零)  
0 = 停止当前操作 (由软件清零时); 还指示当前操作已完成 (由硬件清零时)

- 注**
- 1: 这些位在系统复位或 CRYMD 位置 1 时复位。
  - 2: 只要某个操作正在进行中 (CRYGO 位置 1), 对这些位域的写操作就会被锁定。
  - 3: 如果当 CRYSIDL = 1 时器件进入空闲模式, 模块将停止其当前操作。在 OTP 写操作正在进行时进入空闲模式将导致对 OTP 不可撤销的损坏。

# PIC24FJ128GB204 系列

---

## 寄存器 23-1: CRYCONL: 加密控制寄存器低位字 (续)

bit 7-4	<b>OPMOD&lt;3:0&gt;</b> : 加密操作模式选择位 (1,2) 1111 = 加载会话密钥 (使用密钥加密密钥解密 CRYTXTA/CRYXTB 中的会话密钥并写入 CRYKEY) 1110 = 加密会话密钥 (使用密钥加密密钥加密 CRYKEY 中的会话密钥并写入 CRYTXTA/CRYXTB) 1011 = 保留 1010 = 产生随机数 1001 = 1000 = . . = 保留 . 0011 = 0010 = AES 解密密钥扩展 0001 = 解密 0000 = 加密
bit 3	<b>CPHRSEL</b> : 加密引擎选择位 (1,2) 1 = AES 引擎 0 = DES 引擎
bit 2-0	<b>CPHRMOD&lt;2:0&gt;</b> : 加密模式位 (1,2) 11x = 保留 101 = 保留 100 = 计数器 (Counter, CTR) 模式 011 = 输出反馈 (Output Feedback, OFB) 模式 010 = 加密反馈 (Cipher Feedback, CFB) 模式 001 = 加密块链接 (Cipher Block Chaining, CBC) 模式 000 = 电子密码本 (Electronic Codebook, ECB) 模式

- 注
- 1: 这些位在系统复位或 CRYMD 位置 1 时复位。
  - 2: 只要某个操作正在进行中 (CRYGO 位置 1), 对这些位域的写操作就会被锁定。
  - 3: 如果当 CRYSIDL = 1 时器件进入空闲模式, 模块将停止其当前操作。在 OTP 写操作正在进行时进入空闲模式将导致对 OTP 不可撤销的损坏。

# PIC24FJ128GB204 系列

## 寄存器 23-2: CRYCONH: 加密控制寄存器高位字

U-0	R/W-0 <sup>(1)</sup>						
—	CTRSIZE6 <sup>(2,3)</sup>	CTRSIZE5 <sup>(2,3)</sup>	CTRSIZE4 <sup>(2,3)</sup>	CTRSIZE3 <sup>(2,3)</sup>	CTRSIZE2 <sup>(2,3)</sup>	CTRSIZE1 <sup>(2,3)</sup>	CTRSIZE0 <sup>(2,3)</sup>
bit 15							bit 8

R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	U-0	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>
SKEYSEL	KEYMOD1 <sup>(2)</sup>	KEYMOD0 <sup>(2)</sup>	—	KEYSRC3 <sup>(2)</sup>	KEYSRC2 <sup>(2)</sup>	KEYSRC1 <sup>(2)</sup>	KEYSRC0 <sup>(2)</sup>
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 14-8 **CTRSIZE<6:0>:** 计数器大小选择位 <sup>(1,2,3)</sup>

计数器定义为 CRYXTB<n:0>, 其中 n = CTRSIZEx。每个操作后计数器递增 1, 并在计数器从 (2<sup>n-1</sup>-1) 计满返回到 0 时发生计满返回事件。

11111111 = 128 位 (CRYXTB<127:0>)

11111110 = 127 位 (CRYXTB<126:0>)

•

•

0000010 = 3 位 (CRYXTB<2:0>)

0000001 = 2 位 (CRYXTB<1:0>)

0000000 = 1 位 (CRYXTB<0>); 当 CRYXTB<0> 从 1 切换到 0 时发生计满返回事件

bit 7 **SKEYSEL:** 会话密钥选择位 <sup>(1)</sup>

1 = 使用 CRYKEY<255:128> 执行密钥生成 / 加密 / 加载

0 = 使用 CRYKEY<127:0> 执行密钥生成 / 加密 / 加载

bit 6-5 **KEYMOD<1:0>:** AES/DES 加密 / 解密密钥模式 / 密钥长度选择位 <sup>(1,2)</sup>

对于 DES 加密 / 解密操作 (CPHRSEL = 0):

11 = 64 位, 3 密钥 3DES

10 = 保留

01 = 64 位, 标准 2 密钥 3DES

00 = 64 位 DES

对于 AES 加密 / 解密操作 (CPHRSEL = 1):

11 = 保留

10 = 256 位 AES

01 = 192 位 AES

00 = 128 位 AES

bit 4 **未实现:** 读为 0

bit 3-0 **KEYSRC<3:0>:** 加密密钥源位 <sup>(1,2)</sup>

请参见表 23-1 和表 23-2 获取 KEYSRC<3:0> 值。

- 注
- 1: 这些位在系统复位或 CRYMD 位置 1 时复位。
  - 2: 只要某个操作正在进行中 (CRYGO 位置 1), 对这些位域的写操作就会被锁定。
  - 3: 仅在 CRYXTB 用作计数器时在 CTR 操作中使用; 否则, 这些位无任何作用。

# PIC24FJ128GB204 系列

## 寄存器 23-3: CRYSTAT: 加密状态寄存器

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

R/HSC-x <sup>(1)</sup>	R/HSC-0 <sup>(1)</sup>	R/C-0, HS <sup>(2)</sup>	R/C-0, HS <sup>(2)</sup>	U-0	R/HSC-0 <sup>(1)</sup>	R/HSC-x <sup>(1)</sup>	R/HSC-x <sup>(1)</sup>
CRYBSY <sup>(4)</sup>	TXTABSY	CRYABRT <sup>(5)</sup>	ROLLOVR	—	MODFAIL <sup>(3)</sup>	KEYFAIL <sup>(3,4)</sup>	PGMFAIL <sup>(3,4)</sup>
bit 7						bit 0	

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
HS = 可由硬件置 1 的位	C = 可清零位	HSC = 硬件置 1/ 清零位
-n = POR 时的值	1 = 置 1	0 = 清零
		x = 复位状态条件位

- bit 15-8 **未实现:** 读为 0
- bit 7 **CRYBSY:** 加密 OTP 阵列忙状态位 <sup>(1,4)</sup>  
 1 = 加密模块正在执行加密操作或 OTP 操作  
 0 = 模块当前未在执行任何操作
- bit 6 **TXTABSY:** CRYTXTA 忙状态位 <sup>(1)</sup>  
 1 = CRYTXTA 寄存器在忙, 且不可写入  
 0 = CRYTXTA 空闲且可写入
- bit 5 **CRYABRT:** 加密操作中止状态位 <sup>(2,5)</sup>  
 1 = 用软件清零 CRYGO 位中止上一个操作  
 0 = 上一个操作正常完成 (CRYGO 由硬件清零)
- bit 4 **ROLLOVR:** 计数器计满返回状态位 <sup>(2)</sup>  
 1 = 执行上一个 CTR 模式操作时 CRYXTTB 计数器计满返回。该位置 1 后, 必须由软件清零, 然后才可再次将 CRYGO 位置 1  
 0 = 未发生计满返回事件
- bit 3 **未实现:** 读为 0
- bit 2 **MODFAIL:** 模式配置故障标志位 <sup>(1,3)</sup>  
 1 = 当前所选操作和密码模式配置无效; 在选择有效模式前无法将 CRYWR 位置 1 (使用任何有效配置时, 由硬件自动清零)  
 0 = 当前所选操作和密码模式配置有效
- bit 1 **KEYFAIL:** 密钥配置故障状态位 <sup>(1,3,4)</sup>  
 请参见表 23-1 和表 23-2 了解无效密钥配置。  
 1 = 当前所选密钥和模式配置无效; 在选择有效模式前无法将 CRYWR 位置 1 (使用任何有效配置时, 由硬件自动清零)  
 0 = 当前所选配置有效
- bit 0 **PGMFAIL:** 密钥存储 / 配置编程故障标志位 <sup>(1,3,4)</sup>  
 1 = KEYPG<3:0> 指示的页保留或锁定; 无法将 CRYWR 位置 1 且不可启动任何编程操作  
 0 = 可对 KEYPG<3:0> 指示的页编程

- 注**
- 1: 这些位在系统复位或 CRYMD 位置 1 时复位。
  - 2: 这些位在系统复位、CRYMD 位置 1 或 CRYGO 清零时复位。
  - 3: 即使模块禁止 (CRYON = 0), 这些位仍起作用; 这允许在使能模块前验证模式配置的兼容性。
  - 4: 这些位在所有 OTP 读操作期间自动置 1, 包括 POR 时的初始读操作。读操作完成后, 该位呈现适当的状态, 反映当前配置。
  - 5: 如果该位置 1, 无法执行加密操作。

# PIC24FJ128GB204 系列

## 寄存器 23-4: CRYOTP: 加密 OTP 页编程控制寄存器

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

R/HSC-x <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/S/HC-1	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/S/HC-0 <sup>(2)</sup>
PGMTST	OTPIE	CRYREAD <sup>(3,4)</sup>	KEYPG3	KEYPG2	KEYPG1	KEYPG0	CRYWR <sup>(3,4)</sup>
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
S = 可置 1 位	HC = 可由硬件清零的位	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 **PGMTST:** 密钥存储 / 配置编程测试位 <sup>(1)</sup>  
 该位反映 TSTPGM 位的状态, 并在编程后用于测试对安全 OTP 阵列的编程。  
 1 = TSTPGM (CFGPAGE<30>) 已编程 (1)  
 0 = TSTPGM 未编程 (0)

bit 6 **OTPIE:** 密钥存储 / 配置编程中断允许位 <sup>(1)</sup>  
 1 = 当前编程或读操作完成时产生中断  
 0 = 当前编程或读操作完成时不产生中断; 软件必须查询 CRYWR、CRYREAD 或 CRYBSY 位以确定当前编程操作何时完成

bit 5 **CRYREAD:** 加密密钥存储 / 配置读取位 <sup>(3,4)</sup>  
 1 = 将该位置 1 以启动读操作; 该位置 1 且 CRYGO = 1 时, 表示读操作正在进行中  
 0 = 读操作已完成

bit 4-1 **KEYPG<3:0>:** 密钥存储 / 配置编程页选择位 <sup>(1)</sup>  
 1111 =  
 .  
 . = 保留  
 .  
 1001 =  
 1000 = OTP Page 8  
 0111 = OTP Page 7  
 0110 = OTP Page 6  
 0101 = OTP Page 5  
 0100 = OTP Page 4  
 0011 = OTP Page 3  
 0010 = OTP Page 2  
 0001 = OTP Page 1  
 0000 = 配置页 (CFGPAGE, OTP Page 0)

bit 0 **CRYWR:** 加密密钥存储 / 配置编程位 <sup>(2,3,4)</sup>  
 1 = 使用 CRYTTC<63:0> 中的值编程密钥存储 / 配置位  
 0 = 编程操作已完成

- 注
- 1: 这些位在系统复位或 CRYMD 位置 1 时复位。
  - 2: 这些位在系统复位、CRYMD 位置 1 或 CRYGO 清零时复位。
  - 3: 仅当 CRYON = 1 且 CRYGO = 0 时, 该位才置 1。不要在任何给定时间将 CRYREAD 或 CRYWR 同时置 1。
  - 4: 当 CRYON 或这些位置 1 时, 请勿将其清零; 始终允许硬件操作完成并自动清零该位。

# PIC24FJ128GB204 系列

寄存器 23-5: CFGPAGE: 安全阵列配置位 (OTP PAGE 0) 寄存器

r-x	R/PO-x	U-x	U-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x
—	TSTPGM <sup>(1)</sup>	—	—	KEY4TYPE1	KEY4TYPE0	KEY3TYPE1	KEY3TYPE0
bit 31				bit 24			

R/PO-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x
KEY2TYPE1	KEY2TYPE0	KEY1TYPE1	KEY1TYPE0	SKEYEN	LKYSRC7	LKYSRC6	LKYSRC5
bit 23				bit 16			

R/PO-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x
LKYSRC4	LKYSRC3	LKYSRC2	LKYSRC1	LKYSRC0	SRCLK	WRLOCK8	WRLOCK7
bit 15				bit 8			

R/PO-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x	R/PO-x
WRLOCK6	WRLOCK5	WRLOCK74	WRLOCK3	WRLOCK2	WRLOCK1	WRLOCK0	SWKYDIS
bit 7				bit 0			

<b>图注:</b>	r = 保留位		
R = 可读位	PO = 一次编程位	U = 未实现位, 读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 31      **保留:** 不要修改
- bit 30      **TSTPGM:** 客户编程测试位 <sup>(1)</sup>  
1 = 已编程 CFGPAGE  
0 = 未编程 CFGPAGE
- bit 29-28    **未实现:** 读为 0
- bit 27-26    **KEY4TYPE<1:0>:** OTP Page 7 和 8 的密钥类型位  
00 = 这些页中的密钥仅用于 DES/DES2 操作  
01 = 这些页中的密钥仅用于 DES3 操作  
10 = 这些页中的密钥仅用于 128 位 AES 操作  
11 = 这些页中的密钥仅用于 192 位 /256 位 AES 操作
- bit 25-24    **KEY3TYPE<1:0>:** OTP Page 5 和 6 的密钥类型位  
00 = 这些页中的密钥仅用于 DES/DES2 操作  
01 = 这些页中的密钥仅用于 DES3 操作  
10 = 这些页中的密钥仅用于 128 位 AES 操作  
11 = 这些页中的密钥仅用于 192 位 /256 位 AES 操作
- bit 23-22    **KEY2TYPE<1:0>:** OTP Page 3 和 4 的密钥类型位  
00 = 这些页中的密钥仅用于 DES/DES2 操作  
01 = 这些页中的密钥仅用于 DES3 操作  
10 = 这些页中的密钥仅用于 128 位 AES 操作  
11 = 这些页中的密钥仅用于 192 位 /256 位 AES 操作
- bit 21-20    **KEY1TYPE<1:0>:** OTP Page 1 和 2 的密钥类型位  
00 = 这些页中的密钥仅用于 DES/DES2 操作  
01 = 这些页中的密钥仅用于 DES3 操作  
10 = 这些页中的密钥仅用于 128 位 AES 操作  
11 = 这些页中的密钥仅用于 192 位 /256 位 AES 操作

**注 1:** 该位的状态通过 PGMST 位 (CRYOTP<7>) 反映。

## 寄存器 23-5: CFGPAGE: 安全阵列配置位 (OTP PAGE 0) 寄存器 (续)

- bit 19 **SKEYEN:** 会话密钥使能位  
1 = 存储的 Key #1 仅可用作密钥加密密钥  
0 = 存储的 Key #1 可用于任何操作
- bit 18-11 **LKYSRC<7:0>:** 锁定密钥源配置位  
如果 SRCLCK = 1:  
1xxxxxxx = 密钥源和 KEYSRC<3:0> = 1111 时一样  
01xxxxxx = 密钥源和 KEYSRC<3:0> = 0111 时一样  
001xxxxx = 密钥源和 KEYSRC<3:0> = 0110 时一样  
0001xxxx = 密钥源和 KEYSRC<3:0> = 0101 时一样  
00001xxx = 密钥源和 KEYSRC<3:0> = 0100 时一样  
000001xx = 密钥源和 KEYSRC<3:0> = 0011 时一样  
0000001x = 密钥源和 KEYSRC<3:0> = 0010 时一样  
00000001 = 密钥源和 KEYSRC<3:0> = 0001 时一样  
00000000 = 密钥源和 KEYSRC<3:0> = 0000 时一样  
如果 SRCLCK = 0:  
忽略这些位。
- bit 10 **SRCLCK:** 密钥源锁定位  
1 = 密钥源由 KEYSRC<3:0> 位 (CRYCONH<3:0>) 决定 (禁止软件密钥选择)  
0 = 密钥源由 KEYSRC<3:0> 位 (CRYCONH<3:0>) 决定 (禁止锁定密钥选择)
- bit 9-1 **WRLOCK<8:0>:** 写锁定页使能位  
对于 OTP Page 0 (CFGPAGE) 至 8:  
1 = OTP 页永久锁定且不可编程  
0 = OTP 页未锁定且可编程
- bit 0 **SWKYDIS:** 软件密钥禁止位  
1 = 禁止软件密钥 (CRYKEY 寄存器); 当 KEYSRC<3:0> = 0000 时, KEYFAIL 状态位将置 1 且不启动任何加密 / 解密 / 会话密钥操作直至 KEYSRC<3:0> 位更改为 0000 以外的某个值  
0 = 当 KEYSRC<3:0> = 0000 时软件密钥 (CRYKEY 寄存器) 可用作密钥源

注 1: 该位的状态通过 PGMTST 位 (CRYOTP<7>) 反映。

# PIC24FJ128GB204 系列

表 23-1: DES/3DES 密钥源选择

操作模式	KEYMOD<1:0>	KEYSRC<3:0>	会话密钥源		OTP 阵列地址
			SKEYEN = 0	SKEYEN = 1	
64 位 DES	00	0000 <sup>(1)</sup>	CRYKEY<63:0>		—
		0001	DES Key #1	密钥配置错误 <sup>(2)</sup>	<63:0>
		0010	DES Key #2		<127:64>
		0011	DES Key #3		<191:128>
		0100	DES Key #4		<255:192>
		0101	DES Key #5		<319:256>
		0110	DES Key #6		<383:320>
		0111	DES Key #7		<447:384>
		1111	保留 <sup>(2)</sup>		—
		所有其他值	密钥配置错误 <sup>(2)</sup>		—
64 位 2 密钥 3DES (标准 2 密钥 E-D-E/D-E-D)	01	0000 <sup>(1)</sup>	CRYKEY<63:0> (第 1 个 / 第 3 个) CRYKEY<127:64> (第 2 个)		—
		0001	DES Key #1 (第 1 个 / 第 3 个) DES Key #2 (第 2 个)	密钥配置错误 <sup>(2)</sup>	<63:0> <127:64>
		0010	DES Key #3 (第 1 个 / 第 3 个) DES Key #4 (第 2 个)		<191:128> <255:192>
		0011	DES Key #5 (第 1 个 / 第 3 个) DES Key #6 (第 2 个)		<319:256> <383:320>
		0100	DES Key #7 (第 1 个 / 第 3 个) DES Key #8 (第 2 个)		<447:384> <511:448>
		1111	保留 <sup>(2)</sup>		—
		所有其他值	密钥配置错误 <sup>(2)</sup>		—
(保留)	10	xxxx	密钥配置错误 <sup>(2)</sup>		—
64 位 3 密钥 3DES	11	0000 <sup>(1)</sup>	CRYKEY<63:0> (第 1 次迭代) CRYKEY<127:64> (第 2 次迭代) CRYKEY<191:128> (第 3 次迭代)		—
		0001	DES Key #1 (第 1 个) DES Key #2 (第 2 个) DES Key #3 (第 3 个)	密钥配置错误 <sup>(2)</sup>	<63:0> <127:64> <191:128>
		0010	DES Key #4 (第 1 个) DES Key #5 (第 2 个) DES Key #6 (第 3 个)		<255:192> <319:256> <383:320>
		1111	保留 <sup>(2)</sup>		—
		所有其他值	密钥配置错误 <sup>(2)</sup>		—

注 1: 当 SWKYDIS 也置 1 时, 该配置视为密钥配置错误 (KEYFAIL 位置 1)。

注 2: 选择这些配置时, KEYFAIL 位 (CRYSTAT<1>) 置 1, 并且保持置 1 直至选择了有效配置。

表 23-2: AES 密钥模式 / 源选择

操作模式	KEYMOD<1:0>	KEYSRC<3:0>	密钥源		OTP 地址
			SKEYEN = 0	SKEYEN = 1	
128 位 AES	00	0000 <sup>(1)</sup>	CRYKEY<127:0>		—
		0001	AES Key #1	密钥配置错误 <sup>(2)</sup>	<127:0>
		0010	AES Key #2		<255:128>
		0011	AES Key #3		<383:256>
		0100	AES Key #4		<511:384>
		1111	保留 <sup>(2)</sup>		—
		所有其他值	密钥配置错误 <sup>(2)</sup>		—
192 位 AES	01	0000 <sup>(1)</sup>	CRYKEY<191:0>		—
		0001	AES Key #1	密钥配置错误 <sup>(2)</sup>	<191:0>
		0010	AES Key #2		<383:192>
		1111	保留 <sup>(2)</sup>		—
		所有其他值	密钥配置错误 <sup>(2)</sup>		—
256 位 AES	10	0000 <sup>(1)</sup>	CRYKEY<255:0>		—
		0001	AES Key #1	密钥配置错误 <sup>(2)</sup>	<255:0>
		0010	AES Key #2		<511:256>
		1111	保留 <sup>(2)</sup>		—
		所有其他值	密钥配置错误 <sup>(2)</sup>		—
(保留)	11	xxxx	密钥配置错误 <sup>(2)</sup>		—

注 1: 当 SWKYDIS 也置 1 时, 该配置视为密钥配置错误 (KEYFAIL 位置 1)。

注 2: 选择这些配置时, KEYFAIL 位 (CRYSTAT<1>) 置 1, 并且保持置 1 直至选择了有效配置。

# PIC24FJ128GB204 系列

---

注:

## 24.0 32 位可编程循环冗余校验 (CRC) 发生器

**注:** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息, 请参见《dsPIC33/PIC24 系列参考手册》中的“32 位可编程循环冗余校验 (CRC)” (DS30009729)。本数据手册中的信息取代了 FRM 中的信息。

32 位可编程 CRC 发生器为各种网络和安全应用提供快速产生校验和的硬件实现方法。它提供以下特性:

- 用户可编程的 CRC 多项式方程 (最多 32 位)
- 可编程移位方向 (小尾数或大尾数格式)
- 独立的数据和多项式长度
- 可配置中断输出
- 数据 FIFO

CRC 发生器的简化框图如图 24-1 所示。图 24-2 中给出了 CRC 移位引擎的简化形式。

图 24-1: CRC 模块框图

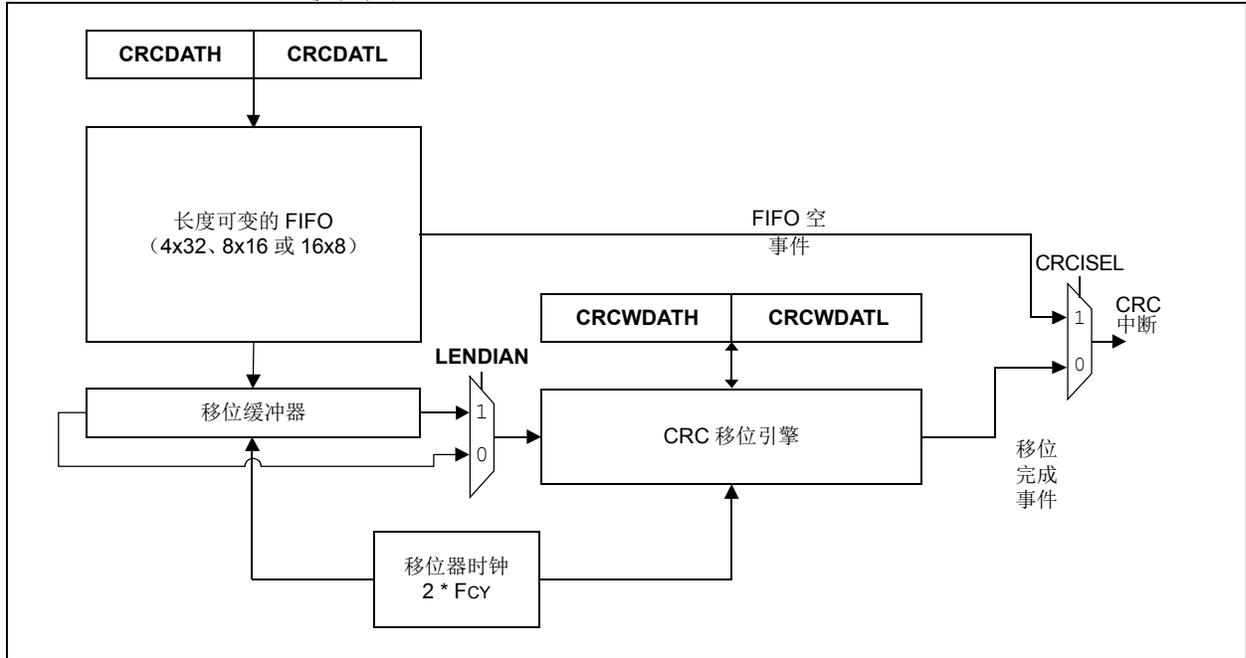
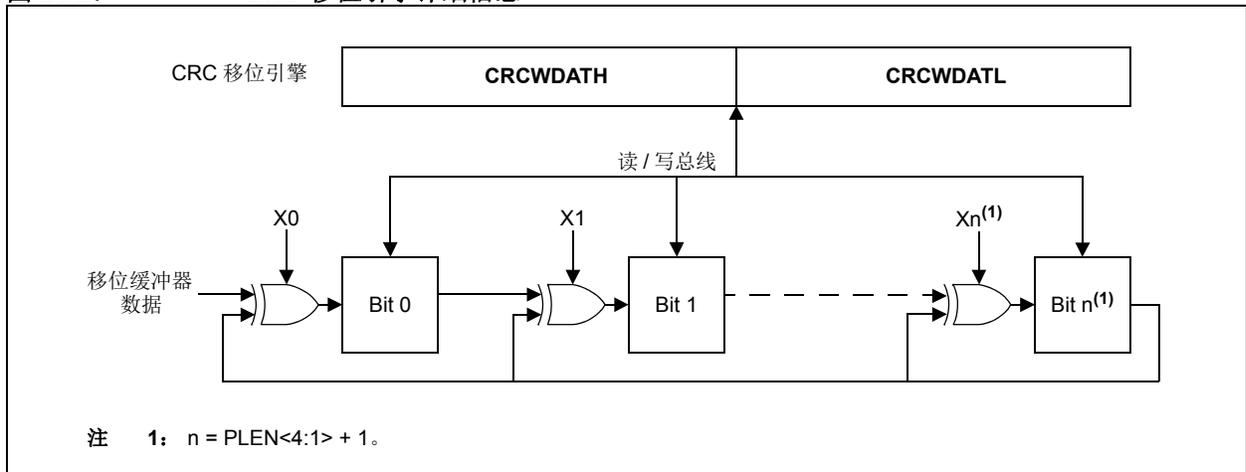


图 24-2: CRC 移位引擎详细信息



# PIC24FJ128GB204 系列

## 24.1 用户接口

### 24.1.1 多项式接口

可使用最多 32 位，将 CRC 模块编程为生成最多 32 阶的 CRC 多项式。

通过 PLEN<4:0> 位 (CRCCON2<4:0>) 选择反映方程中最高次幂的多项式长度。

CRCXORL 和 CRCXORH 寄存器控制方程中包含的幂项。将寄存器中的某个位置 1 会在方程中包含相应的幂项，在功能上相当于包含在 CRC 引擎的相应位上执行的异或操作。清零该位会禁止此异或操作。

例如，考虑两个 CRC 多项式，一个是 16 位方程，另一个是 32 位方程：

**公式 24-1: 16 位和 32 位 CRC 多项式**

$$\begin{aligned}
 &X^{16} + X^{12} + X^5 + 1 \\
 &\quad \text{和} \\
 &X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + \\
 &\quad X^8 + X^7 + X^5 + X^4 + X^2 + X + 1
 \end{aligned}$$

要将这些多项式编程到 CRC 发生器中，请设置表 24-1 中的寄存器位。

注意，请将适当的存储单元置 1 以指示在方程中使用了它们（例如 X<sup>26</sup> 和 X<sup>23</sup>）。方程所需的 0 位始终为异或操作，X<sup>0</sup> 表示无关位。对于长度为 32 的多项式，假设将使用第 32 位。因此，X<31:1> 位没有第 32 位。

### 24.1.2 数据接口

CRC 模块具有可缓存各种数据宽度的 FIFO。可使用 DWIDTH<4:0> 位 (CRCCON2<12:8>) 将输入数据宽度配置为 1 到 32 位之间的任何数值。当数据宽度大于 15 时，FIFO 为 4 字深。当 DWIDTHx 值在 8 至 15 之间时，FIFO 为 8 字深。当 DWIDTHx 值小于 8 时，FIFO 为 16 字深。

首先将要进行 CRC 计算所需的数据写入 FIFO。即使数据宽度小于 8，可被写入 FIFO 的最小数据元素也是 1 个字节。例如，如果 DWIDTHx 位的值为 5，那么数据大小为 DWIDTH<4:0> + 1，即 6。数据作为整个字节写入，CRC 模块忽略了两个未使用的高位。

数据一旦被写入 CRCDAT 寄存器的最高位（由数据宽度定义），VWORD<4:0> 位 (CRCCON1<12:8>) 的值将递增 1。例如，如果 DWIDTHx 值为 24，则写入 CRCDATH 的 bit 7 时，VWORDx 位递增 1。因此，CRCDATL 必须始终先于 CRCDATH 写入。

当 CRCGO 位置 1 且 VWORDx 的值大于零时，CRC 引擎开始移入数据。

每个字都被从 FIFO 复制到缓冲寄存器（这会导致 VWORDx 位减 1）。然后从缓冲器中移出数据。CRC 引擎会以每指令周期两个位的速率继续移动数据直到 VWORDx 位达到 0。因此对于给定的数据宽度，每个字完成计算所需的指令周期数为数据宽度的一半。例如，完成 32 位数据单字的 CRC 计算需要 16 个周期。

当 VWORDx 位达到所配置 DWIDTHx 值（4、8 或 16）的最大值时，CRCFUL 位置 1。当 VWORDx 位达到零时，CRCMPT 位置 1。当 CRCEN 为 0 时，FIFO 为空且 VWORD<4:0> 设置为 00000。

在写入 CRCWDAT 之后，必须至少经过一个指令周期才可以读 VWORDx 位。

**表 24-1: CRC 设置示例（16 和 32 位多项式）**

CRC 控制位	位值	
	16 位多项式	32 位多项式
PLEN<4:0>	01111	11111
X<31:16>	0000 0000 0000 0001	0000 0100 1100 0001
X<15:0>	0001 0000 0010 000x	0001 1101 1011 011x

## 24.1.3 数据移位方向

LENDIAN 位 (CRCCON1<3>) 用于控制移位方向。默认情况下, CRC 通过引擎从最高有效位开始移入数据。将 LENDIAN 置 1 (= 1) 可导致 CRC 从最低有效位开始移入数据。此设置可使 CRC 与各种通信机制更好地集成, 并省去了用软件使位反序的开销。注意, 这只是改变了将数据移入引擎的方向。CRC 计算的结果仍然是正常的 CRC 结果, 不是反序的 CRC 结果。

## 24.1.4 中断操作

用户可以配置该模块在以下两种条件下产生中断。

如果 CRCISEL 为 0, 当 VWORD<4:0> 位的值从 1 跳变到 0 时产生中断。如果 CRCISEL 为 1, 在 CRC 操作完成且模块将 CRCGO 位设置为 0 时产生中断。手动将 CRCGO 设置为 0 不会产生中断。请注意, 产生中断时, CRC 计算尚未完成。产生中断后, 模块将仍需要 (PLEN + 1)/2 个时钟周期, 才会完成 CRC 计算。

## 24.1.5 典型操作

要使用模块进行典型 CRC 计算, 请执行以下步骤:

1. 置 1 CRCEN 位以使能模块。
2. 配置模块以实现所需的操作:
  - a) 使用 CRCXORL 和 CRCXORH 寄存器以及 PLEN<4:0> 位对所需的多项式编程。
  - b) 使用 DWIDTHx 和 LENDIAN 位配置数据宽度和移位方向。
  - c) 使用 CRCISEL 位选择所需的中断模式。
3. 在 CRCFUL 位置 1 之前或 FIFO 中无数据时, 通过写入 CRCDATL 和 CRCDATH 寄存器来预装载 FIFO。

4. 通过写入 00h 到 CRCWDATL 和 CRCWDATH 来清除旧数据。CRCWDAT 寄存器也可以保持不变以继续上一个暂停的计算。
5. 置 1 CRCGO 位以启动计算。
6. 在 FIFO 有可用空间时将剩余数据写入 FIFO 中。
7. 计算完成时, CRCGO 自动清零。如果 CRCISEL = 1, 则将产生中断。
8. 读取 CRCWDATL 和 CRCWDATH 以获得计算结果。

有 8 个寄存器用于控制可编程 CRC 操作:

- CRCCON1
- CRCCON2
- CRCXORL
- CRCXORH
- CRCDATL
- CRCDATH
- CRCWDATL
- CRCWDATH

CRCCON1 和 CRCCON2 寄存器 (寄存器 24-1 和寄存器 24-2) 用于控制模块的操作和配置各种设置。

CRCXORL/H 寄存器 (寄存器 24-3 和寄存器 24-4) 用于选择 CRC 方程中要使用的多项式项。CRCDAT 和 CRCWDAT 寄存器是分别用作双字输入数据和 CRC 处理结果输出的缓冲器的寄存器对。

# PIC24FJ128GB204 系列

寄存器 24-1: CRCCON1: CRC 控制寄存器 1

R/W-0	U-0	R/W-0	R-0, HSC				
CRCCEN	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0
bit 15							bit 8

R-0, HSC	R-1, HSC	R/W-0	R/W-0, HC	R/W-0	U-0	U-0	U-0
CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	—	—	—
bit 7							bit 0

<b>图注:</b>	HC = 可由硬件清零的位	HSC = 可由硬件置 1/ 清零的位
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      **CRCCEN:** CRC 使能位  
1 = 使能模块  
0 = 禁止模块; 所有状态机、指针和 CRCWDAT/CRCDATH 寄存器都复位; 其他 SFR 不复位
- bit 14      **未实现:** 读为 0
- bit 13      **CSIDL:** 空闲模式 CRC 停止位  
1 = 当器件进入空闲模式时, 模块停止工作  
0 = 模块在空闲模式下继续工作
- bit 12-8    **VWORD<4:0>:** 指针值位  
指出 FIFO 中有效字的个数。当 PLEN<4:0> ≥ 7 时, 最大值为 8, 当 PLEN<4:0> ≤ 7 时, 最大值为 16。
- bit 7        **CRCFUL:** CRC FIFO 满位  
1 = FIFO 为满  
0 = FIFO 未滿
- bit 6        **CRCMPT:** CRC FIFO 空位  
1 = FIFO 为空  
0 = FIFO 非空
- bit 5        **CRCISEL:** CRC 中断选择位  
1 = FIFO 空时中断; 仍在通过 CRC 移位最后的数据字  
0 = 移位完成时中断且结果已计算好
- bit 4        **CRCGO:** 启动 CRC 位  
1 = 启动 CRC 串行移位器  
0 = 关闭 CRC 串行移位器
- bit 3        **LENDIAN:** 数据移位方向选择位  
1 = 数据字从最低有效位开始移入 FIFO (小尾数格式)  
0 = 数据字从最高有效位开始移入 FIFO (大尾数格式)
- bit 2-0     **未实现:** 读为 0

# PIC24FJ128GB204 系列

## 寄存器 24-2: CRCCON2: CRC 控制寄存器 2

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0
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
—	—	—	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0
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 12-8     **DWIDTH<4:0>:** 数据字宽度配置位  
配置数据字的宽度 (数据字宽度 - 1)。

bit 7-5     **未实现:** 读为 0

bit 4-0     **PLEN<4:0>:** 多项式长度配置位  
配置多项式的长度 (多项式长度 - 1)。

# PIC24FJ128GB204 系列

## 寄存器 24-3: CRCXORL: CRC 多项式异或操作寄存器 (低位字)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X<15:8>							
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	U-0
X<7:1>							—
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 15-1      **X<15:1>**: 多项式的项  $x^n$  的异或操作使能位  
 bit 0        未实现: 读为 0

## 寄存器 24-4: CRCXORH: CRC 多项式异或操作寄存器 (高位字)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X<31:24>							
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
X<23:16>							
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      **X<31:16>**: 多项式的项  $x^n$  的异或操作使能位

## 25.0 带阈值检测功能的 12 位 A/D 转换器

**注：** 本数据手册总结了该 PIC24F 系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解 12 位 A/D 转换器的更多信息，请参见《dsPIC33/PIC24 系列参考手册》的“带阈值检测功能的 12 位 A/D 转换器” (DS39739)。

12 位 A/D 转换器具有以下主要特性：

- 逐次逼近寄存器 (Successive Approximation Register, SAR) 转换
- 转换速度最高可达 200 ksp/s
- 最多 19 个模拟输入通道 (内部和外部)
- 可选择 10 位或 12 位 (默认) 转换分辨率
- 多个内部参考电压输入通道
- 外部参考电压输入引脚
- 单极型差分采样 / 保持 (Sample-and-Hold, S/H) 放大器
- 用以预先评估转换结果的自动阈值扫描和比较操作
- 可选择转换触发源
- 固定长度 (每通道一个字) 的可配置转换结果缓冲区
- 4 个结果对齐选项
- 可配置中断产生
- 采用间接地址发生的增强型 DMA 操作
- 可在 CPU 休眠和空闲模式下工作

12 位 A/D 转换器模块对早期 PIC24 器件中提供的 10 位模块进行了改进。该转换器为逐次逼近寄存器 (SAR) 转换器，对以下功能进行了改进：提供 12 位分辨率、提供一系列更广泛的自动采样选项、更紧密地与其他模拟模块进行集成，以及具有可配置的结果缓冲区。

该模块还包含了一种独特的阈值检测功能，让模块自身可以基于转换结果做出简单决定，以及通过外设间接寻址 (Peripheral Indirect Addressing, PIA) 与 DMA 控制器实现了增强操作。

图 25-1 给出了该模块的简化框图。

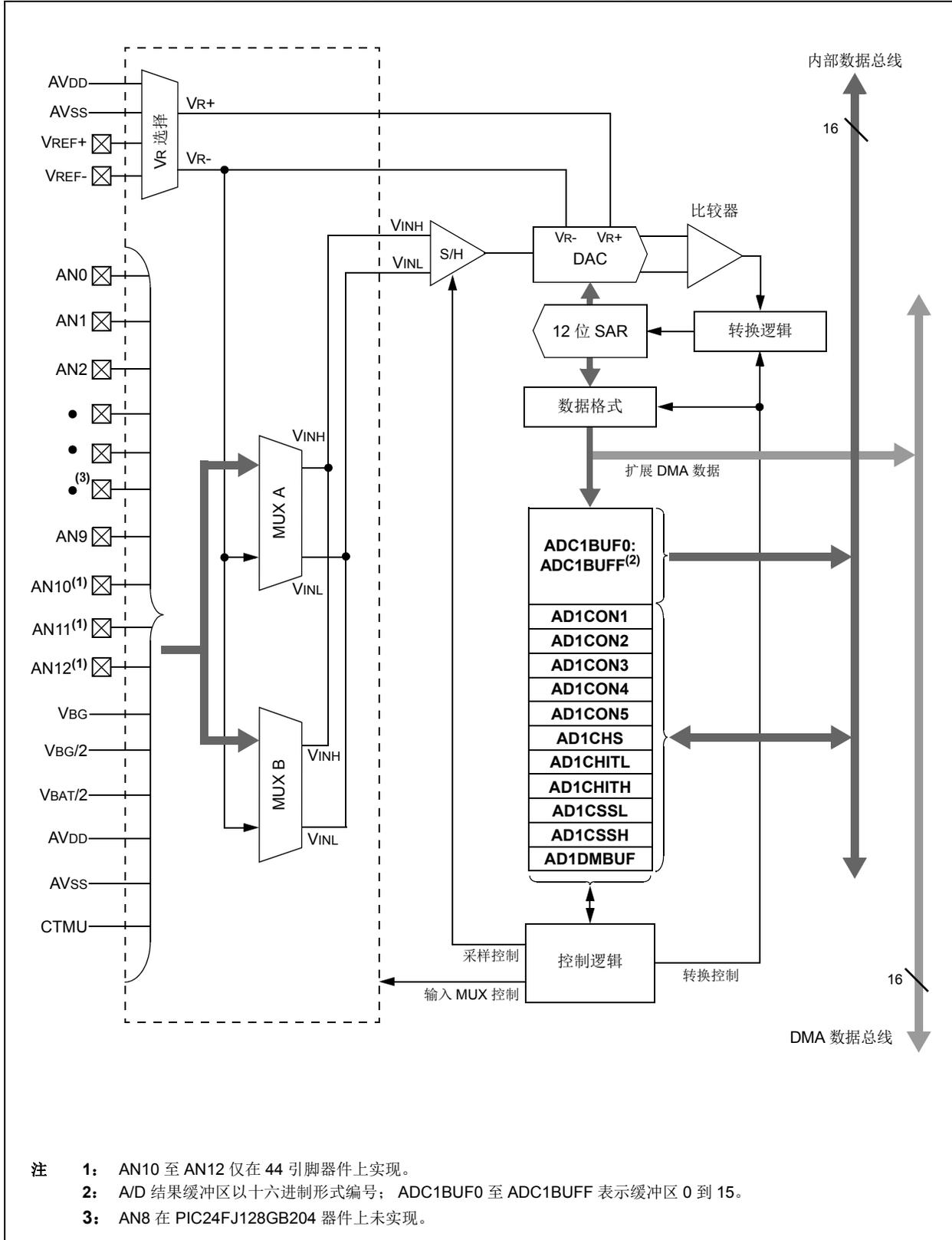
## 25.1 基本操作

要执行一个标准 A/D 转换：

1. 配置 A/D 模块：
  - a) 通过将 ANSx 寄存器中的相应位置 1 将端口引脚配置为模拟输入 (更多信息，请参见第 11.2 节“配置模拟端口引脚 (ANSx)”)。
  - b) 选择参考电压源以匹配模拟输入的预期范围 (AD1CON2<15:13>)。
  - c) 选择每个通道的同相和反相多路开关输入 (AD1CHS<15:0>)。
  - d) 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>)。
  - e) 选择适当的采样/转换序列 (AD1CON1<7:4> 位和 AD1CON3<12:8> 位)。
  - f) 对于通道 A 扫描操作，选择要包含的同相通道 (AD1CSSH 和 AD1CSSL 寄存器)。
  - g) 选择转换结果在缓冲区中的存储方式 (AD1CON1<9:8> 和 AD1CON5 寄存器)。
  - h) 选择中断发生的频率 (AD1CON2<6:2>)。
  - i) 启动 A/D 模块 (AD1CON1<15>)。
2. 配置 A/D 中断 (如果需要)：
  - a) 清零 AD1IF 位 (IFS0<13>)。
  - b) 允许 AD1IE 中断 (IEC0<13>)。
  - c) 选择 A/D 中断优先级 (IPC3<6:4>)。
3. 如果模块配置为手动采样，将 SAMP 位 (AD1CON1<1>) 置 1 以开始采样。

# PIC24FJ128GB204 系列

图 25-1: 12 位 A/D 转换器框图 (PIC24FJ128GB204 系列)



## 25.2 扩展 DMA 操作

除了所有 12 位 A/D 转换器提供的标准功能外，PIC24FJ128GB204 系列器件实现了 DMA 功能的有限扩展。该扩展增加的功能可与器件的 DMA 控制器配合使用来扩大 A/D 模块的数据存储能力，使之不仅包含模块的内置缓冲区。

该扩展 DMA 功能通过 DMAEN 位 (AD1CON1<11>) 控制，将该位置 1 将使能该功能。DMABM 位 (AD1CON1<12>) 用于配置 DMA 功能的工作方式。

### 25.2.1 扩展缓冲区模式

扩展缓冲区模式 (DMABM = 1) 对于存储通道结果非常有用。该模式还可用于将任一 A/D 通道上的转换结果存储在数据 RAM 中任何实现的地址中。

在扩展缓冲区模式下，将 A/D 缓冲寄存器 and 编号在 26 以上的通道中的所有数据映射到数据 RAM。转换数据写入 DMA 控制器指定的目标单元，具体由 DMADSTn 寄存器指定。这允许用户从数据存储读取编号在 26 以上的通道的转换结果，这些通道没有自己的存储器映射的 A/D 缓冲区单元。

使用扩展缓冲区模式时，始终置 1 BUFREGEN 位以禁止 FIFO 操作。此外，通过清零 BUFM 位禁止分割缓冲区模式。

### 25.2.2 PIA 模式

DMABM = 0 时，A/D 模块配置为与 DMA 控制器配合工作以实现外设间接寻址 (PIA) 模式操作。在该模式下，A/D 模块产生一个 11 位间接地址 (Indirect Address, IA)。该 IA 与 DMA 控制器中的目标地址进行逻辑或运算来定义存储 A/D 转换数据的位置。

在 PIA 模式下，缓冲区空间创建为一系列连续的较小缓冲区，每个模拟通道分配一个缓冲区。通道缓冲区的大小决定了可容纳的模拟通道的数量。缓冲区大小通过 DMABLx 位 (AD1CON4<2:0>) 选择。缓冲区大小选择范围为从单个字到 128 个字。每个通道均分配该大小的缓冲区，无论将来是否会对具体的通道执行转换。

通过将通道缓冲区中的基址与标识通道的 3 到 5 位 (具体取决于缓冲区大小) 组合来创建 IA。基址的宽度范围为 0 到 7 位，具体取决于缓冲区大小。地址右边补 0 以保持地址在数据空间中对齐。然后，合并的通道和基址位根据需要在左边补 0 以构成 11 位 IA。

IA 配置为在写操作期间使用 SMP1x 位 (AD1CON2<6:2>) 自动递增。

和任何使能 DMA 模块的 PIA 操作一样，必须适当地掩码 DMADSTn 寄存器中的目标基址以包含 IA。表 25-1 给出了构成完整地址的方式。请注意，选择的缓冲区大小不同，地址掩码也会有所不同。由于掩码需要，某些地址范围可能不可用于某些缓冲区大小。用户应验证 DMA 基址是否与所选的缓冲区大小兼容。

图 25-2 给出了地址的各部分如何定义数据存储中的缓冲区单元。在这种情况下，模块将 256 字节的数据 RAM (1000h 至 1100h) 分配给 32 个缓冲区，每个缓冲区 4 个字。然而，这不是硬分配，这些单元还可用于其他目的。例如，在当前情况下，如果模拟通道 1、3 和 8 被采样和转换，转换数据将仅写入分别起始于 1008h、1018h 和 1040h 的通道缓冲区。PIA 缓冲空间中的空存储单元可用于任何其他目的。用户有必要跟踪缓冲区单元并防止数据改写。

## 25.3 采用 VBAT 的 A/D 操作

其中一个 A/D 通道连接到 VBAT 引脚以监视 VBAT 电压。这允许在不进行任何外部连接的情况下监视 VBAT 引脚电压 (电池电压)。使用 A/D VBAT 监视器测得的电压为  $V_{BAT}/2$ 。对于 10 位 A/D，根据读数  $A/D = ((V_{BAT}/2)/V_{DD}) * 1024$  计算该电压，而对于 12 位 A/D，根据读数  $((V_{BAT}/2)/V_{DD}) * 4096$  计算。

使用 VBAT A/D 监视器时：

- 将 A/D 通道接地以对采样电容放电。
- 由于 VBAT 的高阻抗，选择较大的采样时间可获取更准确的读数。

由于 VBAT 引脚在采样期间连接到 A/D 以延长 VBAT 电池寿命，因此建议仅在需要时选择 VBAT 通道。

# PIC24FJ128GB204 系列

## 25.4 寄存器

12 位 A/D 转换器通过总共 11 个寄存器进行控制:

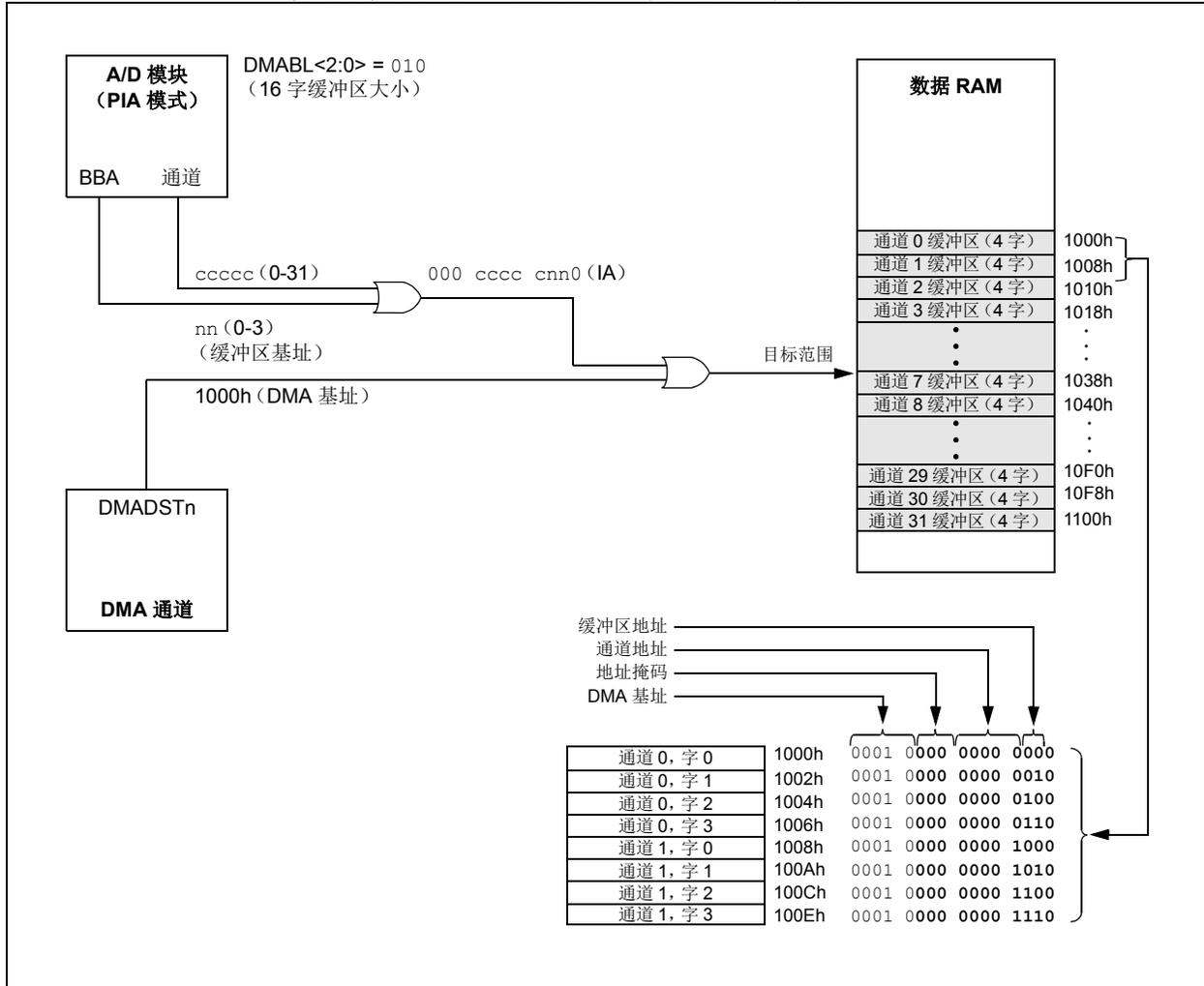
- AD1CON1 至 AD1CON5 (寄存器 25-1 至寄存器 25-5)
- AD1CHS (寄存器 25-6)
- AD1CHITL (寄存器 25-8)
- AD1CSSH 和 AD1CSSL (寄存器 25-9 和寄存器 25-10)
- AD1CTMENL (寄存器 25-11)
- AD1DMBUF (未显示) —— 用于扩展缓冲区模式的 16 位转换缓冲区

表 25-1: PIA 模式下的间接地址生成

DMABL<2:0>	每个通道的缓冲区大小 (字)	生成的偏移地址 (低 11 位)	可用输入通道数	允许的 DMADSTn 地址
000	1	000 00cc ccc0	32	xxxx xxxx xx00 0000
001	2	000 0ccc ccn0	32	xxxx xxxx x000 0000
010	4	000 cccc cnn0	32	xxxx xxxx 0000 0000
011	8	00c cccc nnn0	32	xxxx xxx0 0000 0000
100	16	0cc cccn nnn0	32	xxxx xx00 0000 0000
101	32	ccc ccnn nnn0	32	xxxx x000 0000 0000
110	64	ccc cnnn nnn0	16	xxxx x000 0000 0000
111	128	ccc nnnn nnn0	8	xxxx x000 0000 0000

图注: ccc = 通道数 (3 至 5 位); n = 缓冲区基址 (0 至 7 位);  
x = 用于基址的 DMADSTn 的用户定义范围; 0 = 用于 IA 的 DMADSTn 掩码位。

图 25-2: PIA 模式下缓冲区地址生成示例 (每个通道 4 字缓冲区)



# PIC24FJ128GB204 系列

## 寄存器 25-1: AD1CON1: ADC1 控制寄存器 1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADON	—	ADSIDL	DMABM <sup>(1)</sup>	DMAEN	MODE12	FORM1	FORM0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0, HSC	R/C-0, HSC
SSRC3	SSRC2	SSRC1	SSRC0	—	ASAM	SAMP	DONE
bit 7							bit 0

<b>图注:</b>	C = 可清零位	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      **ADON:** ADC1 工作模式位  
 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      **ADSIDL:** ADC1 空闲模式停止位  
 1 = 当器件进入空闲模式后, 模块停止工作  
 0 = 模块在空闲模式下继续工作
- bit 12      **DMABM:** 扩展 DMA 缓冲区模式选择位 <sup>(1)</sup>  
 1 = 扩展缓冲区模式: 缓冲区地址由 DMADSTn 寄存器定义  
 0 = PIA 模式: 缓冲区地址由 DMA 控制器和 AD1CON4<2:0> 定义
- bit 11      **DMAEN:** 扩展 DMA/ 缓冲区使能位  
 1 = 使能扩展 DMA 和缓冲区功能  
 0 = 禁止扩展功能
- bit 10      **MODE12:** ADC1 12 位工作模式位  
 1 = 12 位 A/D 工作  
 0 = 10 位 A/D 工作
- bit 9-8      **FORM<1:0>:** 数据输出格式位 (见以下格式)  
 11 = 小数结果, 有符号, 左对齐  
 10 = 绝对小数结果, 无符号, 左对齐  
 01 = 十进制结果, 有符号, 右对齐  
 00 = 绝对十进制结果, 无符号, 右对齐
- bit 7-4      **SSRC<3:0>:** 采样时钟源选择位  
 1xxx = 未实现, 不要使用  
 0111 = 内部计数器结束采样并开始转换 (自动转换); 不要在自动扫描模式下使用  
 0110 = 未实现  
 0101 = TMR1  
 0100 = CTMU  
 0011 = TMR5  
 0010 = TMR3  
 0001 = INTO  
 0000 = 必须由软件清零 SAMP 位以开始转换
- bit 3      **未实现:** 读为 0
- bit 2      **ASAM:** ADC1 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 = 手动将 SAMP 位置 1 时开始采样

注 1: 该位仅在扩展 DMA/ 缓冲区功能有效 (DMAEN = 1) 时可用。

**寄存器 25-1: AD1CON1: ADC1 控制寄存器 1 (续)**

- bit 1     **SAMP:** ADC1 采样使能位  
          1 = A/D 采样 / 保持放大器正在采样  
          0 = A/D 采样 / 保持放大器在保持采样结果
- bit 0     **DONE:** ADC1 转换状态位  
          1 = A/D 转换完成  
          0 = A/D 转换尚未开始或在进行中

**注 1:** 该位仅在扩展 DMA/ 缓冲区功能有效 (DMAEN = 1) 时可用。

# PIC24FJ128GB204 系列

## 寄存器 25-2: AD1CON2: ADC1 控制寄存器 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
PVCFG1	PVCFG0	NVCFG0	OFFCAL	BUFREGEN	CSCNA	—	—
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
BUFS <sup>(1)</sup>	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM <sup>(1)</sup>	ALTS
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      **PVCFG<1:0>**: 转换器正参考电压配置位

1x = 未实现, 不要使用  
 01 = 外部 VREF+  
 00 = AVDD

bit 13      **NVCFG0**: 转换器负参考电压配置位

1 = 外部 VREF-  
 0 = AVSS

bit 12      **OFFCAL**: 失调校准模式选择位

1 = 采样 / 保持通道的反相和同相输入与 AVSS 连接  
 0 = 采样 / 保持通道的反相和同相输入与一般输入连接

bit 11      **BUFREGEN**: ADC1 缓冲寄存器使能位

1 = 转换结果装入根据转换通道确定的缓冲单元  
 0 = A/D 结果缓冲区视为 FIFO 缓冲区

bit 10      **CSCNA**: 采样 A 期间 CH0+ 的扫描输入选择位

1 = 扫描输入  
 0 = 不扫描输入

bit 9-8      **未实现**: 读为 0

bit 7      **BUFS**: 缓冲区填充状态位 <sup>(1)</sup>

1 = A/D 正在填充 ADC1BUF8-ADC1BUFF, 用户应该访问 ADC1BUF0-ADC1BUF7 中的数据  
 0 = A/D 正在填充 ADC1BUF0-ADC1BUF7, 用户应该访问 ADC1BUF8-ADC1BUFF 中的数据

注 1: 这些位仅在缓冲区采用 FIFO 模式 (BUFREGEN = 0) 时适用。此外, BUFS 仅在 BUFM = 1 时使用。

## 寄存器 25-2: AD1CON2: ADC1 控制寄存器 2 (续)

bit 6-2 **SMPI<4:0>**: 中断采样 /DMA 递增速率选择位

当 **DMAEN = 1** 时:

11111 = 完成 32 次采样 / 转换操作后递增 DMA 地址

11110 = 完成 31 次采样 / 转换操作后递增 DMA 地址

•

•

•

00001 = 完成 2 次采样 / 转换操作后递增 DMA 地址

00000 = 完成 1 次采样 / 转换操作后递增 DMA 地址

当 **DMAEN = 0** 时:

11111 = 对每 32 次采样进行转换后中断

11110 = 对每 31 次采样进行转换后中断

•

•

•

00001 = 对每 2 次采样进行转换后中断

00000 = 对每次采样进行转换后中断

bit 1 **BUF<sub>M</sub>**: 缓冲区填充模式选择位 <sup>(1)</sup>

1 = 第一次中断时从 **ADC1BUF0** 处开始填充缓冲区, 下一次中断时从 **ADC1BUF8** 处开始填充

0 = 始终从 **ADC1BUF0** 处开始填充缓冲区

bit 0 **ALTS**: 交替输入采样模式选择位

1 = 在第一次采样时使用采样多路开关 **A** 选择的输入通道, 而在下一次采样时使用采样多路开关 **B** 选择的输入通道

0 = 总是使用采样多路开关 **A** 选择的输入通道

注 1: 这些位仅在缓冲区采用 FIFO 模式 (**BUFREGEN = 0**) 时适用。此外, **BUFS** 仅在 **BUF<sub>M</sub> = 1** 时使用。

# PIC24FJ128GB204 系列

## 寄存器 25-3: AD1CON3: ADC1 控制寄存器 3

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	EXTSAM	PUMPEN	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 = 可读位                      W = 可写位                      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:** ADC1 转换时钟源位  
 1 = RC 时钟  
 0 = 时钟由系统时钟产生
- bit 14        **EXTSAM:** 扩展采样时间位  
 1 = A/D 在 SAMP = 0 之后仍然在采样  
 0 = A/D 完成采样
- bit 13        **PUMPEN:** 电荷泵使能位  
 1 = 使能开关的电荷泵  
 0 = 禁止开关的电荷泵
- bit 12-8     **SAMC<4:0>:** 自动采样时间选择位  
 11111       = 31 TAD  
 .  
 .  
 .  
 00001       = 1 TAD  
 00000       = 0 TAD
- bit 7-0      **ADCS<7:0>:** ADC1 转换时钟选择位  
 11111111   = 256 • T<sub>CY</sub> = TAD  
 .  
 .  
 .  
 00000001   = 2 • T<sub>CY</sub> = TAD  
 00000000   = T<sub>CY</sub> = TAD

# PIC24FJ128GB204 系列

## 寄存器 25-4: AD1CON4: ADC1 控制寄存器 4

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-0	R/W-0	R/W-0
—	—	—	—	—	DMABL<2:0> <sup>(1)</sup>		
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      **DMABL<2:0>:** DMA 缓冲区大小选择位 <sup>(1)</sup>

111 = 给每个模拟输入分配 128 字的缓冲区

110 = 给每个模拟输入分配 64 字的缓冲区

101 = 给每个模拟输入分配 32 字的缓冲区

100 = 给每个模拟输入分配 16 字的缓冲区

011 = 给每个模拟输入分配 8 字的缓冲区

010 = 给每个模拟输入分配 4 字的缓冲区

001 = 给每个模拟输入分配 2 字的缓冲区

000 = 给每个模拟输入分配 1 字的缓冲区

**注 1:** DMABL<2:0> 位仅在 AD1CON1<11> = 1 且 AD1CON<12> = 0 时使用; 否则, 将忽略这些位的值。

# PIC24FJ128GB204 系列

## 寄存器 25-5: AD1CON5: ADC1 控制寄存器 5

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
ASEN	LPEN	CTMREQ	BGREQ	—	—	ASINT1	ASINT0
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	WM1	WM0	CM1	CM0
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        **ASEN:** 自动扫描使能位  
 1 = 使能自动扫描  
 0 = 禁止自动扫描
- bit 14        **LPEN:** 低功耗使能位  
 1 = 扫描之后使能低功耗  
 0 = 扫描之后使能全功耗
- bit 13        **CTMREQ:** CTMU 请求位  
 1 = 在 A/D 使能并处于活动状态时使能 CTMU  
 0 = 不通过 A/D 使能 CTMU
- bit 12        **BGREQ:** 带隙请求位  
 1 = 在 A/D 使能并处于活动状态时使能带隙  
 0 = 不通过 A/D 使能带隙
- bit 11-10     **未实现:** 读为 0
- bit 9-8        **ASINT<1:0>:** 自动扫描 (阈值检测) 中断模式位  
 11 = 在阈值检测序列完成且发生有效比较之后产生中断  
 10 = 在发生有效比较之后产生中断  
 01 = 在阈值检测序列完成之后产生中断  
 00 = 不产生中断
- bit 7-4        **未实现:** 读为 0
- bit 3-2        **WM<1:0>:** 写模式位  
 11 = 保留  
 10 = 仅自动比较 (不保存转换结果, 但在发生由 CMx 和 ASINTx 位定义的有效匹配时产生中断)  
 01 = 转换并保存 (当发生由 CMx 位定义的匹配时, 转换结果保存到由寄存器位决定的存储单元中)  
 00 = 传统操作 (转换数据保存到由缓冲寄存器位决定的存储单元中)
- bit 1-0        **CM<1:0>:** 比较模式位  
 11 = 窗外模式 (如果转换结果超出相应缓冲区对定义的窗口, 则发生有效匹配)  
 10 = 窗内模式 (如果转换结果处于相应缓冲区对定义的窗口中, 则发生有效匹配)  
 01 = 大于模式 (如果结果大于相应缓冲寄存器中的值, 则发生有效匹配)  
 00 = 小于模式 (如果结果小于相应缓冲寄存器中的值, 则发生有效匹配)

# PIC24FJ128GB204 系列

## 寄存器 25-6: AD1CHS: ADC1 采样选择寄存器

R/W-0							
CH0NB2	CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0
bit 15							bit 8

R/W-0							
CH0NA2	CH0NA1	CH0NA0	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 15-13 **CH0NB<2:0>**: 采样多路开关 B 的通道 0 的反相输入选择位

1xx = 未实现

011 = 未实现

010 = AN1

001 = 未实现

000 = VREF-/AVSS

bit 12-8 **CH0SB<4:0>**: 采样多路开关 B 的通道 0 的同相输入选择位

11111 = VBAT/2<sup>(1)</sup>

11110 = AVDD<sup>(1)</sup>

11101 = AVSS<sup>(1)</sup>

11100 = 带隙参考电压 (VBG) <sup>(1)</sup>

11011 = VBG/2<sup>(1)</sup>

01110 = CTMU

01101 = CTMU 温度传感器输入 (不需要设置 AD1CTMENL<12>)

01100 = AN12<sup>(2)</sup>

01011 = AN11<sup>(2)</sup>

01010 = AN10<sup>(2)</sup>

01001 = AN9

01000 = 未实现

00111 = AN7

00110 = AN6

00101 = AN5

00100 = AN4

00011 = AN3

00010 = AN2

00001 = AN1

00000 = AN0

bit 7-5 **CH0NA<2:0>**: 采样多路开关 A 的通道 0 的反相输入选择位  
与 CH0NB<2:0> 定义相同。

bit 4-0 **CH0SA<4:0>**: 采样多路开关 A 的通道 0 的同相输入选择位  
与 CH0SB<4:0> 定义相同。

注 1: 这些输入通道没有相应的存储器映射的结果缓冲区。

2: 这些通道在 28 引脚器件上未实现。

# PIC24FJ128GB204 系列

寄存器 25-7: ANCFG: A/D 带隙参考配置寄存器

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	R/W-0	R/W-0	R/W-0
—	—	—	—	—	—	VBG2EN	VBGEN
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      **VBG2EN:** A/D 输入 V<sub>BG/2</sub> 使能位

1 = 使能二分之一带隙参考电压 (V<sub>BG/2</sub>)

0 = 禁止二分之一带隙参考电压 (V<sub>BG/2</sub>)

bit 0      **VBGEN:** A/D 输入 V<sub>BG</sub> 使能位

1 = 使能带隙参考电压 (V<sub>BG</sub>)

0 = 禁止带隙参考电压 (V<sub>BG</sub>)

# PIC24FJ128GB204 系列

寄存器 25-8: **AD1CHITL: ADC1 扫描比较命中寄存器 (低位字)**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CHH<12:9> <sup>(1)</sup>				—
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
CHH<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 15-13    **未实现:** 读为 0
- bit 12-9    **CHH<12:9>:** ADC1 比较命中位 <sup>(1)</sup>  
如果 CM<1:0> = 11:  
 1 = A/D 结果缓冲区 n 已写入数据或已发生匹配  
 0 = A/D 结果缓冲区 n 尚未写入数据  
对于 CM<1:0> 的所有其他值:  
 1 = A/D 结果通道 n 上发生了匹配  
 0 = A/D 结果通道 n 上未发生匹配
- bit 8        **未实现:** 读为 0
- bit 7-0     **CHH<8:0>:** ADC1 比较命中位  
如果 CM<1:0> = 11:  
 1 = A/D 结果缓冲区 n 已写入数据或已发生匹配  
 0 = A/D 结果缓冲区 n 尚未写入数据  
对于 CM<1:0> 的所有其他值:  
 1 = A/D 结果通道 n 上发生了匹配  
 0 = A/D 结果通道 n 上未发生匹配

**注 1:** CHH<12:10> 位在 28 引脚器件上未实现, 读为 0。

# PIC24FJ128GB204 系列

## 寄存器 25-9: AD1CSSH: ADC1 输入扫描选择寄存器 (高位字)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
CSS<31:27>					—	—	—
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      **CSS<31:27>**: ADC1 输入扫描选择位

1 = 输入扫描时包含相应的通道  
 0 = 输入扫描时跳过通道

bit 10-0      **未实现**: 读为 0

## 寄存器 25-10: AD1CSSL: ADC1 输入扫描选择寄存器 (低位字)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	CSS<14:9> <sup>(1)</sup>					—	
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
CSS<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      **未实现**: 读为 0

bit 14-9      **CSS<14:9>**: ADC1 输入扫描选择位 <sup>(1)</sup>

1 = 输入扫描时包含相应的通道  
 0 = 输入扫描时跳过通道

bit 8      **未实现**: 读为 0

bit 7-0      **CSS<7:0>**: ADC1 输入扫描选择位

1 = 输入扫描时包含相应的通道  
 0 = 输入扫描时跳过通道

注 1: CSS<12:10> 位在 28 引脚器件上未实现, 读为 0。

# PIC24FJ128GB204 系列

寄存器 25-11: **AD1CTMENL: ADC1 CTMU 使能寄存器 (低位字) (1)**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	CTMEN<12:9>(2)				—
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
CTMEN<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 15-13 **未实现:** 读为 0

bit 12-9 **CTMEN<12:9>:** 转换期间 CTMU 使能位 (2)  
 1 = 在转换期间使能 CTMU 并连接到选定的通道  
 0 = CTMU 不连接到该通道

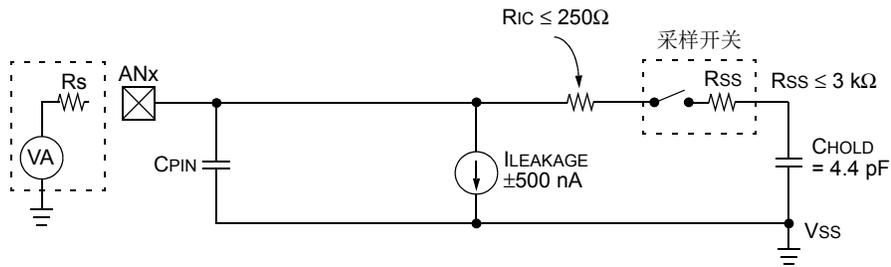
bit 8 **未实现:** 读为 0

bit 7-0 **CTMEN<7:0>:** 转换期间 CTMU 使能位  
 1 = 在转换期间使能 CTMU 并连接到选定的通道  
 0 = CTMU 不连接到该通道

- 注 1:** 实际的可用通道数取决于具体器件上实现了哪些通道。更多信息, 请参见第 1.0 节“器件概述”中的表 1-1 和表 1-2。未实现的通道, 读为 0。
- 2:** CTMEN<12:10> 位在 28 引脚器件上未实现, 读为 0。

# PIC24FJ128GB204 系列

图 25-3: 10 位 A/D 转换器模拟输入模型



图注	CPIN	= 输入电容 <sup>(1)</sup>
	VT	= 阈值电压
	ILEAKAGE	= 由各结点在引脚上产生的泄漏电流
	RIC	= 片内走线等效电阻
	RSS	= 采样开关电阻
	CHOLD	= 采样 / 保持电容 (来自 DAC)

注 1: CPIN 值取决于器件封装, 未经测试。如果  $R_s \leq 5 \text{ k}\Omega$ , 可忽略 CPIN 的影响。

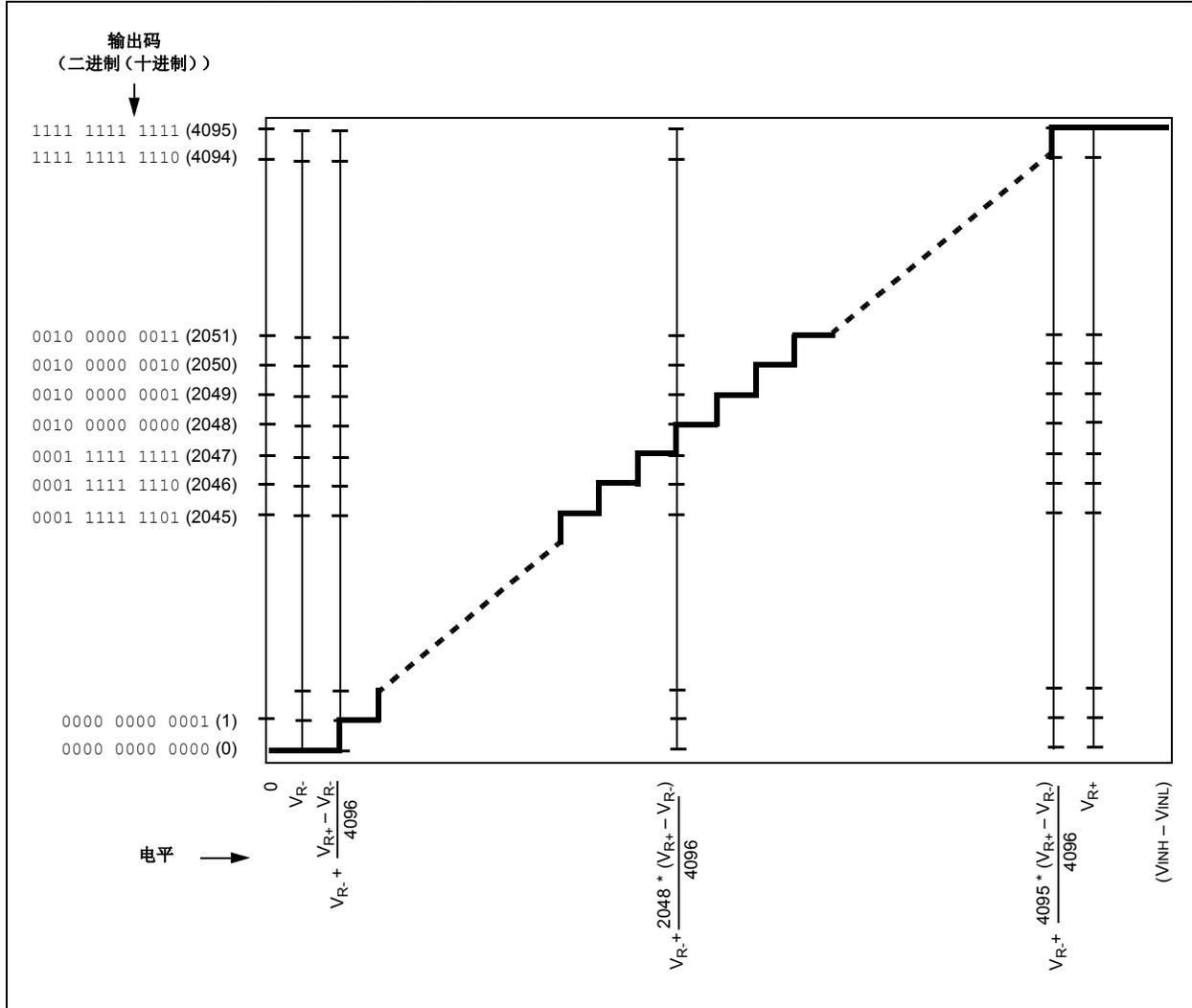
公式 25-1: A/D 转换时钟周期

$$T_{AD} = T_{CY} (ADCS + 1)$$

$$ADCS = \frac{T_{AD}}{T_{CY}} - 1$$

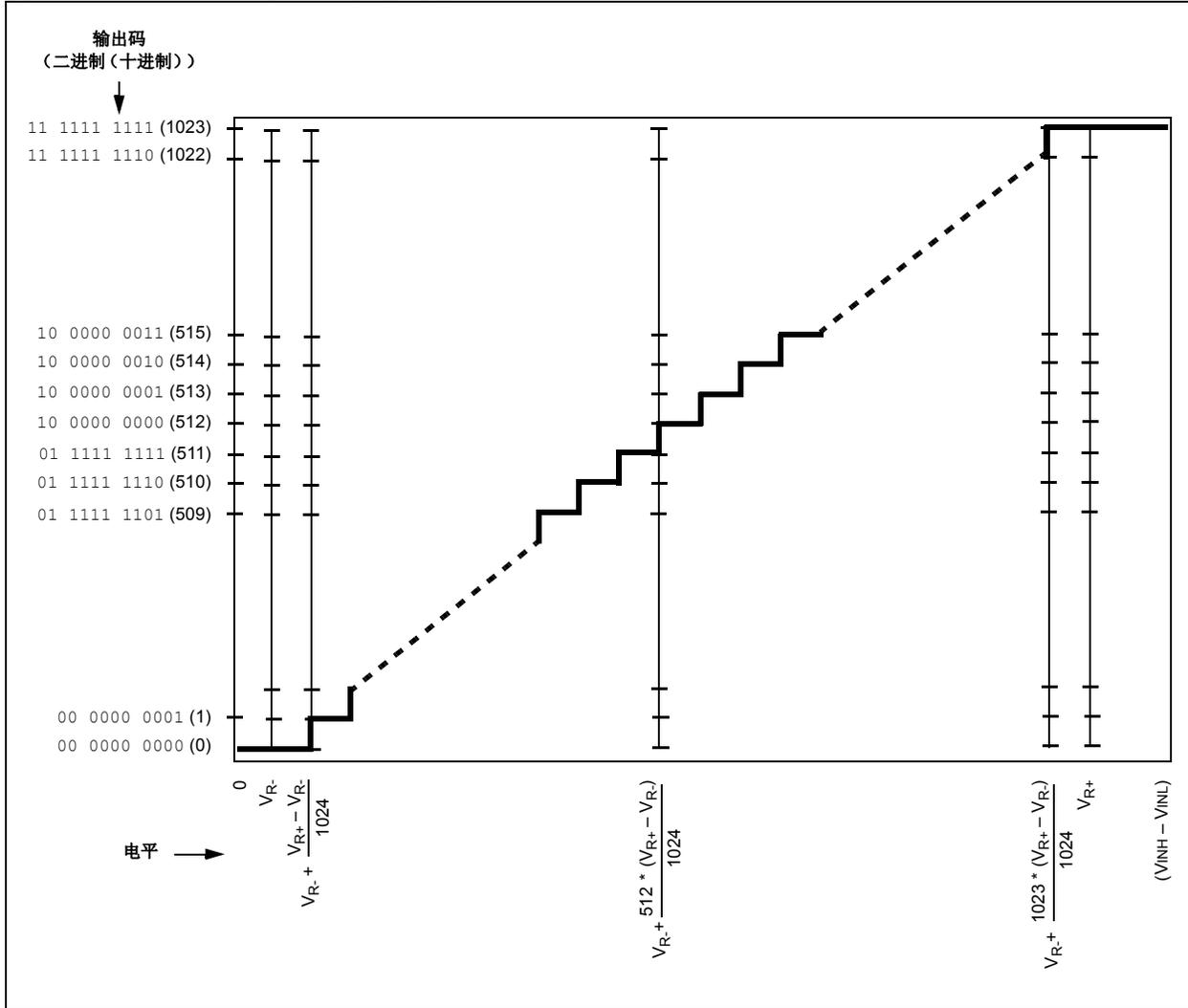
注: 基于  $T_{CY} = 2/F_{OSC}$ ; 打吨模式和 PLL 被禁止。

图 25-4: 12 位 A/D 传递函数



# PIC24FJ128GB204 系列

图 25-5: 10 位 A/D 传递函数



## 26.0 三比较器模块

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本参考手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“可扩展比较器模块”（DS39734）。本数据手册中的信息取代了FRM中的信息。

三比较器模块提供三个双输入比较器。至比较器的输入可配置为使用五个外部模拟输入（CxINA、CxINB、CxINC、CxIND 和 VREF+）之一，且参考电压输入可以

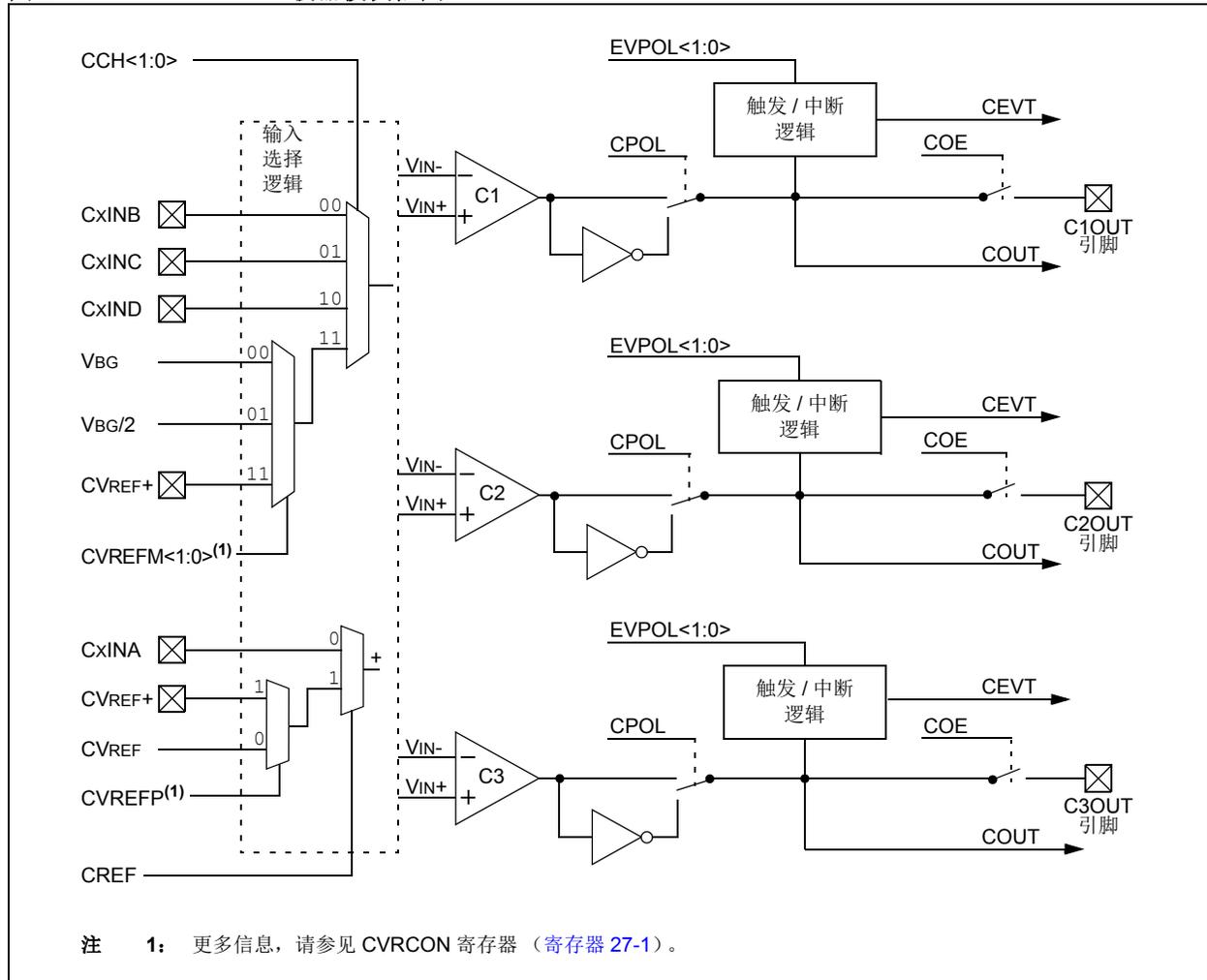
来自其中一个内部带隙参考或比较器参考电压发生器（VBG、VBG/2 和 CVREF）。

比较器输出可直接连接到 CxOUT 引脚。当对应的 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。

模块的简化框图如图 26-1 所示。图 26-2、图 26-3 和图 26-4 中给出了各种可能的比较器配置图。

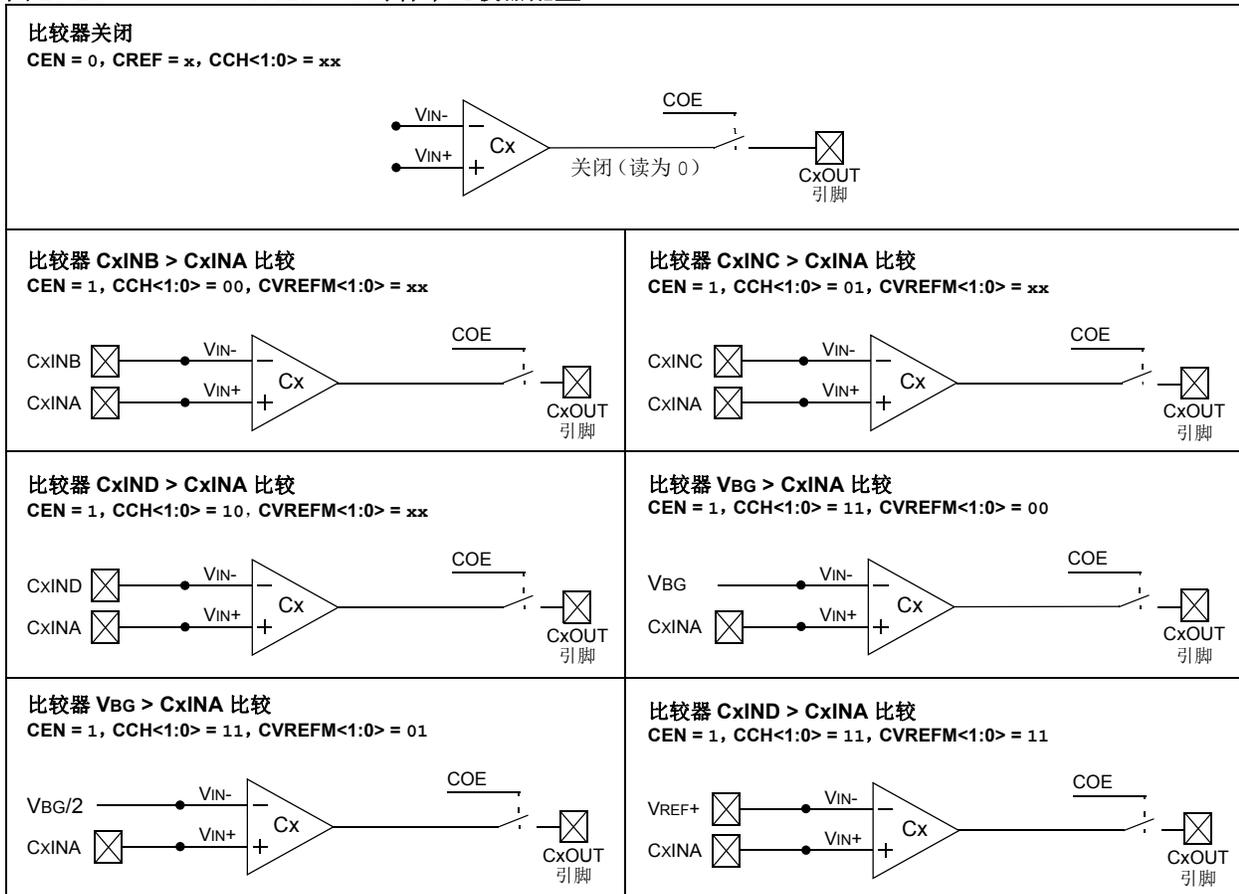
每个比较器都有自己的控制寄存器 CMxCON（寄存器 26-1），用于使能和配置其操作。所有三个比较器的输出和事件状态都在 CMSTAT 寄存器（寄存器 26-2）中给出。

图 26-1: 三比较器模块框图

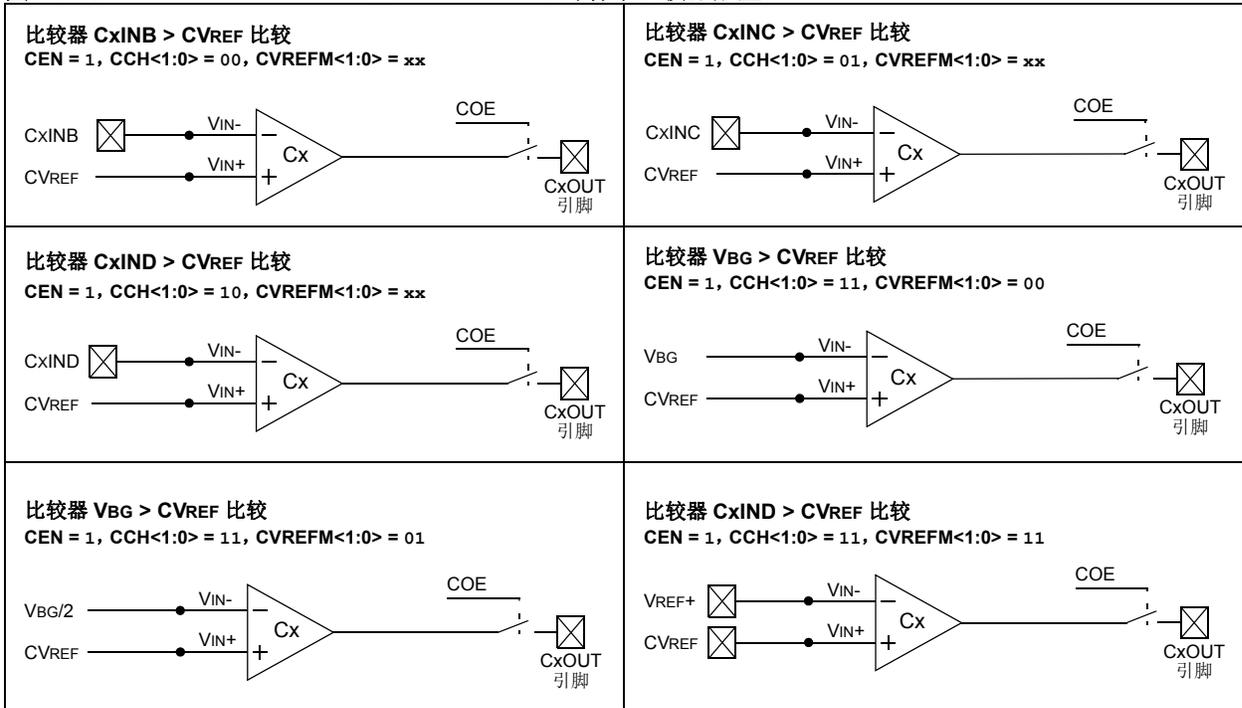


# PIC24FJ128GB204 系列

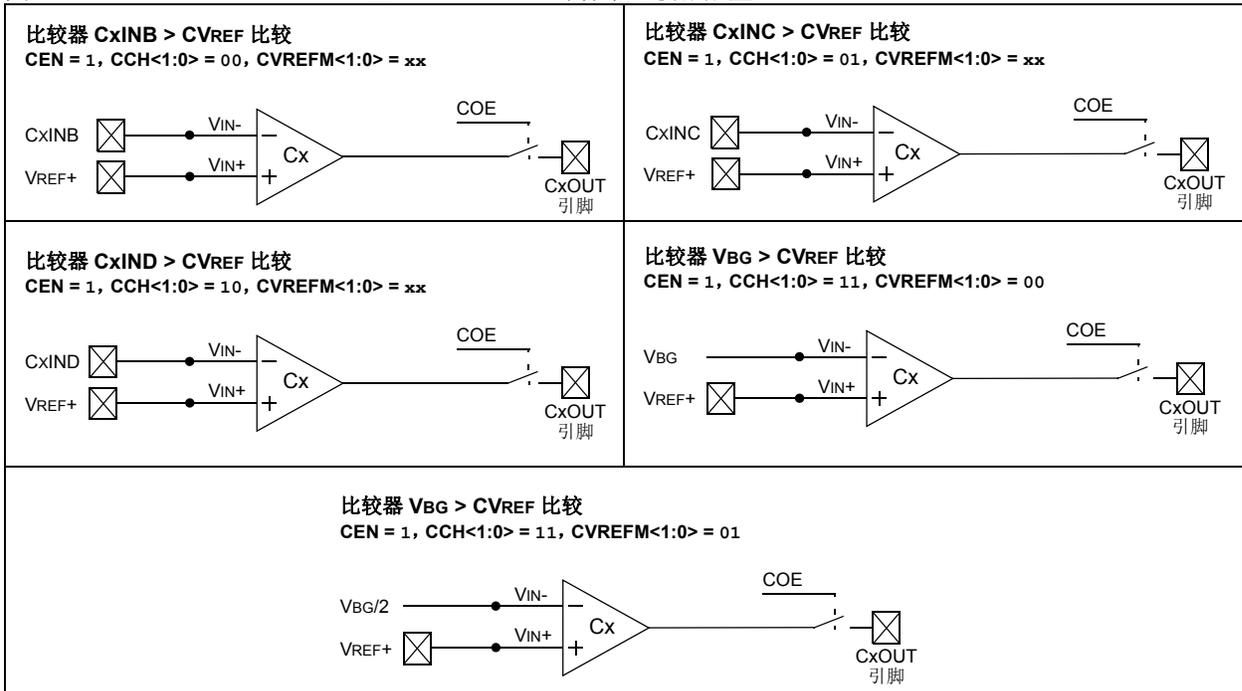
图 26-2: CREF = 0 时各个比较器配置



**图 26-3: CREF = 1 且 CVREFP = 0 时各个比较器配置**



**图 26-4: CREF = 1 且 CVREFP = 1 时各个比较器配置**



# PIC24FJ128GB204 系列

寄存器 26-1: **CMxCON: 比较器 x 控制寄存器 (比较器 1 - 3)**

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0, HS	R-0, HSC
CON	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 <sup>(1)</sup>	EVPOL0 <sup>(1)</sup>	—	CREF	—	—	CCH1	CCH0
bit 7						bit 0	

<b>图注:</b>	HS = 可由硬件置 1 的位	HSC = 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      **CON:** 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 = 在 CxOUT 引脚上提供比较器输出  
0 = 比较器输出仅供内部使用
- 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 = 发生由 EVPOL<1:0> 位定义的比较器事件; 该位清零之前禁止后续触发和中断  
0 = 未发生比较器事件
- bit 8        **COUT:** 比较器输出位  
当 CPOL = 0 时:  
1 =  $V_{IN+} > V_{IN-}$   
0 =  $V_{IN+} < V_{IN-}$   
当 CPOL = 1 时:  
1 =  $V_{IN+} < V_{IN-}$   
0 =  $V_{IN+} > V_{IN-}$
- bit 7-6     **EVPOL<1:0>:** 触发 / 事件 / 中断极性选择位 <sup>(1)</sup>  
11 = 比较器输出的任何变化导致发生触发 / 事件 / 中断 (当 CEVT = 0 时)  
10 = 比较器输出的高到低跳变导致发生触发 / 事件 / 中断:  
01 = 比较器输出的低到高跳变导致发生触发 / 事件 / 中断:  
00 = 禁止发生触发 / 事件 / 中断
- bit 5        **未实现:** 读为 0
- 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

**注 1:** 如果 EVPOL<1:0> 位被设置为非 00 值, 那么 COUT 的任何跳变均将产生第一个中断。后续中断将根据 EVPOLx 位的设置产生。

## 寄存器 26-1: CMxCON: 比较器 x 控制寄存器 (比较器 1 - 3) (续)

bit 1-0 **CCH<1:0>**: 比较器通道选择位  
 11 = 比较器反相输入连接到 CVRCON 寄存器中的 CVREFM<1:0> 位指定的内部可选参考电压  
 10 = 比较器反相输入连接到 CxIND 引脚  
 10 = 比较器反相输入连接到 CxINC 引脚  
 10 = 比较器反相输入连接到 CxINB 引脚

**注 1:** 如果 EVPOL<1:0> 位被设置为非 00 值, 那么 COUT 的任何跳变均将产生第一个中断。后续中断将根据 EVPOLx 位的设置产生。

## 寄存器 26-2: CMSTAT: 比较器模块状态寄存器

R/W-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
CMIDL	—	—	—	—	C3EVT	C2EVT	C1EVT
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
—	—	—	—	—	C3OUT	C2OUT	C1OUT
bit 7							bit 0

<b>图注:</b>	HSC = 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 **CMIDL**: 比较器空闲模式停止位  
 1 = 当器件进入空闲模式时, 所有比较器停止工作  
 0 = 所有使能的比较器在空闲模式下继续工作

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 的当前事件状态 (CM3CON<9>)。

bit 9 **C2EVT**: 比较器 2 事件状态位 (只读)  
 显示比较器 2 的当前事件状态 (CM2CON<9>)。

bit 8 **C1EVT**: 比较器 1 事件状态位 (只读)  
 显示比较器 1 的当前事件状态 (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 的当前输出 (CM3CON<8>)。

bit 1 **C2OUT**: 比较器 2 输出状态位 (只读)  
 显示比较器 2 的当前输出 (CM2CON<8>)。

bit 0 **C1OUT**: 比较器 1 输出状态位 (只读)  
 显示比较器 1 的当前输出 (CM1CON<8>)。

# PIC24FJ128GB204 系列

---

注:

## 27.0 比较器参考电压

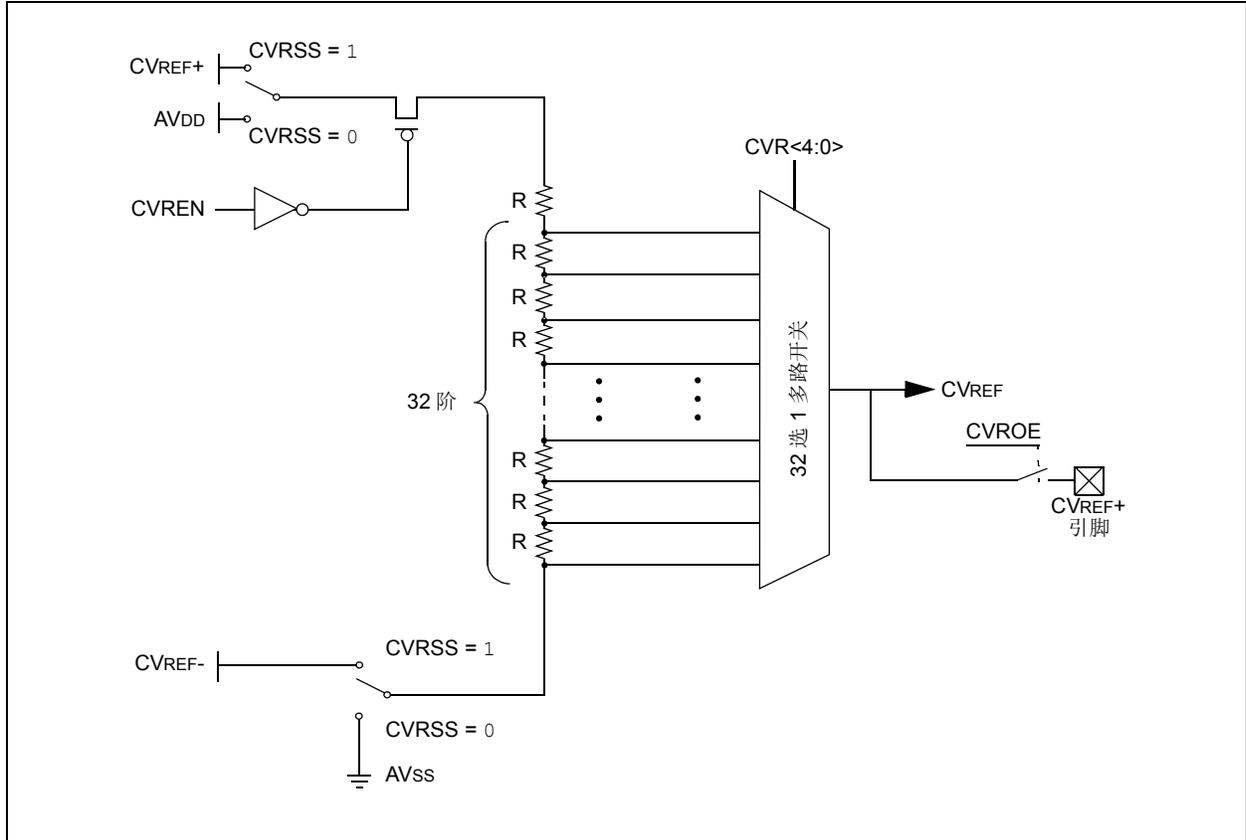
**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。如需了解更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“比较器参考电压”（DS39709）。本数据手册中的信息取代了FRM中的信息。

## 27.1 配置比较器参考电压

比较器参考电压模块由CVRCON寄存器（寄存器27-1）控制，它能提供一系列的输出电压，具有 32 种不同的电平。比较器参考电压模块的电压可以来自 VDD 和 VSS 或外部 CVREF+ 和 CVREF- 引脚。电压源通过 CVRSS 位（CVRCON<5>）选择。

在改变 CVREF 输出值时，必须考虑到比较器参考电压的稳定时间。

图 27-1: 比较器参考电压框图



# PIC24FJ128GB204 系列

寄存器 27-1: **CVRCON**: 比较器参考电压控制寄存器

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CVREFP	CVREFM1	CVREFM0
bit 15						bit 8	

R/W-0							
CVREN	CVROE	CVRSS	CVR4	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 15-11     **未实现:** 读为 0
- bit 10       **CVREFP:** 参考电压选择位 (仅当 CREF 为 1 时有效)
  - 1 = VREF+ 用作比较器的参考电压
  - 0 = 此模块内的 CVR<4:0> 位 (4 位 DAC) 为比较器提供参考电压
- bit 9-8       **CVREFM<1:0>:** 带隙参考源选择位 (仅当 CCH<1:0> = 11 时有效)
  - 00 = 提供带隙电压作为比较器的输入
  - 01 = 提供二分之一带隙电压作为比较器的输入
  - 10 = 保留
  - 11 = 提供 VREF+ 引脚作为比较器的输入
- bit 7         **CVREN:** 比较器参考电压使能位
  - 1 = CVREF 电路上电
  - 0 = CVREF 电路掉电
- bit 6         **CVROE:** 比较器 VREF 输出使能位
  - 1 = CVREF 电压从 CVREF 引脚输出
  - 0 = CVREF 电压与 CVREF 引脚断开
- bit 5         **CVRSS:** 比较器 VREF 源选择位
  - 1 = 比较器参考源 CVRSRC = VREF+ – VREF-
  - 0 = 比较器参考源 CVRSRC = AVDD – AVSS
- bit 4-0       **CVR<4:0>:** 比较器 VREF 值选择位
  - CVREF = (CVR<4:0>/32) • (CVRSS)

## 28.0 充电时间测量单元 (CTMU)

**注：** 本数据手册总结了该 PIC24F 系列器件的特性。但是不应把本参考手册当作无所不包的参考资料来使用。如需了解充电时间测量单元的更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“带阈值检测功能的充电时间测量单元 (CTMU)” (DS39743)。本数据手册中的信息替代 FRM 中的信息。

充电时间测量单元 (CTMU) 是一个灵活的模拟模块，它提供充电时间测量、脉冲源之间的精确时间差测量，以及异步脉冲生成功能。它的主要特性包括：

- 13 个外部边沿输入触发源
- 每个边沿源的极性控制
- 控制边沿序列
- 控制对边沿电平或边沿跳变的响应
- 时间测量分辨率为 1 ns
- 适合电容测量的精确电流源

CTMU 可与其他片内模拟模块一起使用，以精确地测量时间、电容以及电容的相对变化或生成不依赖于系统时钟的输出脉冲。CTMU 模块是与电容式触摸传感器接口的理想选择。

CTMU 可通过三个寄存器进行控制：CTMUCON1、CTMUCON2 和 CTMUICON。CTMUCON1 用于使能模块、控制 CTMU 的工作模式，以及控制边沿序列。CTMUCON2 用于控制边沿源选择和边沿源极性选择。CTMUICON 寄存器选择电流源范围并微调电流。

## 28.1 测量电容

CTMU 模块通过在两个单独的输入通道上产生输出脉冲（脉冲宽度等于边沿事件之间的时间间隔）来测量电容。这两个输入通道上的脉冲边沿事件源有四种选择：两个内部外设模块 (OC1 和 Timer1) 和最多 13 个外部引脚 (CTED1 到 CTED13)。该脉冲和模块的精确电流源一起使用，可根据以下关系计算电容：

公式 28-1:

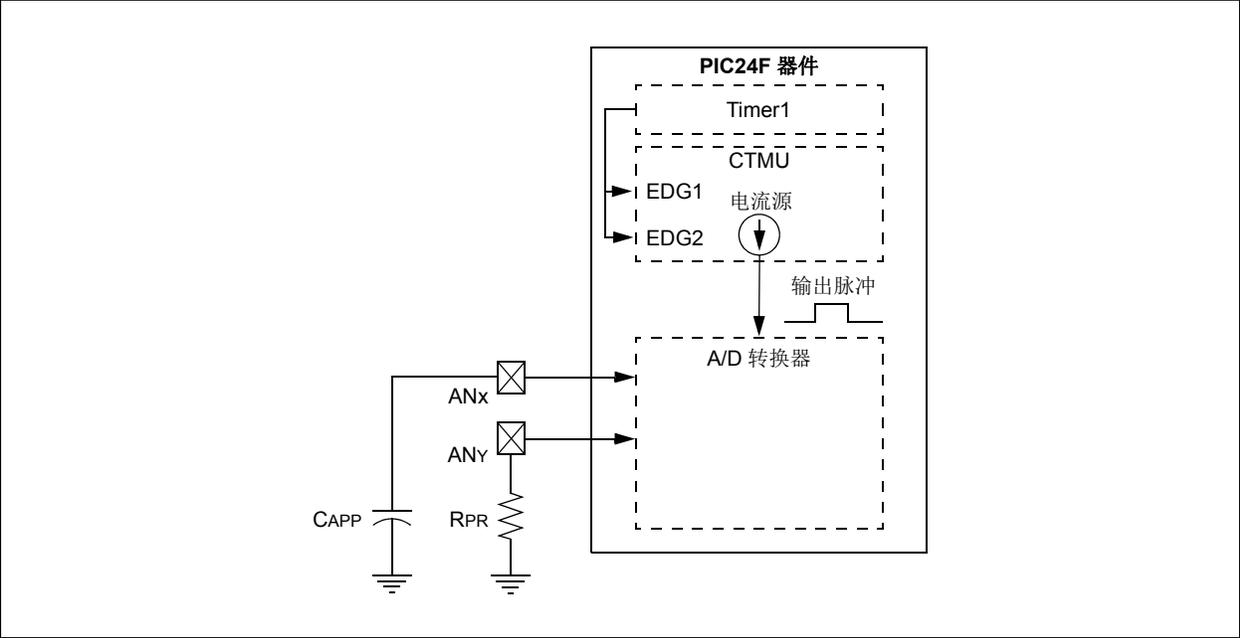
$$I = C \cdot \frac{dV}{dT}$$

测量电容时，A/D 转换器在 CTMU 输出脉冲信号之后对其某个输入通道采样外部电容 (CAPP)。由第二个 A/D 通道上的精确电阻 (RPR) 对电流源进行校准。脉冲信号结束后，转换器确定电容上的电压。电容的实际计算在软件中由应用程序执行。

图 28-1 给出了电容测量使用的外部连接以及此应用中 CTMU 和 A/D 模块的关系。此示例还展示了来自 Timer1 的边沿事件，但使用外部边沿源的其他配置也是可能的。关于使用 CTMU 模块测量电容和时间的详细说明，请参见《dsPIC33/PIC24 系列参考手册》中的“带阈值检测功能的充电时间测量单元 (CTMU)” (DS39743)。

# PIC24FJ128GB204 系列

图 28-1: 电容测量的典型连接和内部配置



## 28.2 测量时间

对脉冲宽度的时间测量也可类似地执行，需要使用 A/D 模块的内部电容（CAD）和用于校准电流的高精度电阻。图 28-2 给出了用于时间测量的外部连接以及此应用中 CTMU 和 A/D 模块的关系。此示例还显示了来自外部 CTED 引脚的边沿事件，但使用内部边沿源的其他配置也是可能的。

## 28.3 脉冲生成和延时

CTMU 模块也可生成边沿与器件的系统时钟异步的输出脉冲。更明确地说，它可以生成和该模块边沿事件输入之间存在可编程延时的脉冲。

当通过置 1 TGEN 位（CTMUCON1<12>）将模块配置为生成脉冲延时，将内部电流源连接到比较器 2 的 B 输入。将电容（CDELAY）连接到比较器 2 引脚 C2INB，且比较器参考电压 CVREF 连接到 C2INA。CVREF 随后被配置为特定跳变点。当检测到边沿事件时，模块开始对 CDELAY 充电。当 CDELAY 充电到超过 CVREF 跳变点时，在 CTPLS 上输出脉冲信号。脉冲延时的时间长度由 CDELAY 和 CVREF 跳变点的值决定。

图 28-3 给出了生成脉冲延时的外部连接，以及所需的不同模拟模块之间的关系。CTED1 显示为输入脉冲源时，其他选项可用。关于使用 CTMU 模块生成脉冲的详细说明，请参见《dsPIC33/PIC24 系列参考手册》。

图 28-2: 时间测量的典型连接和内部配置

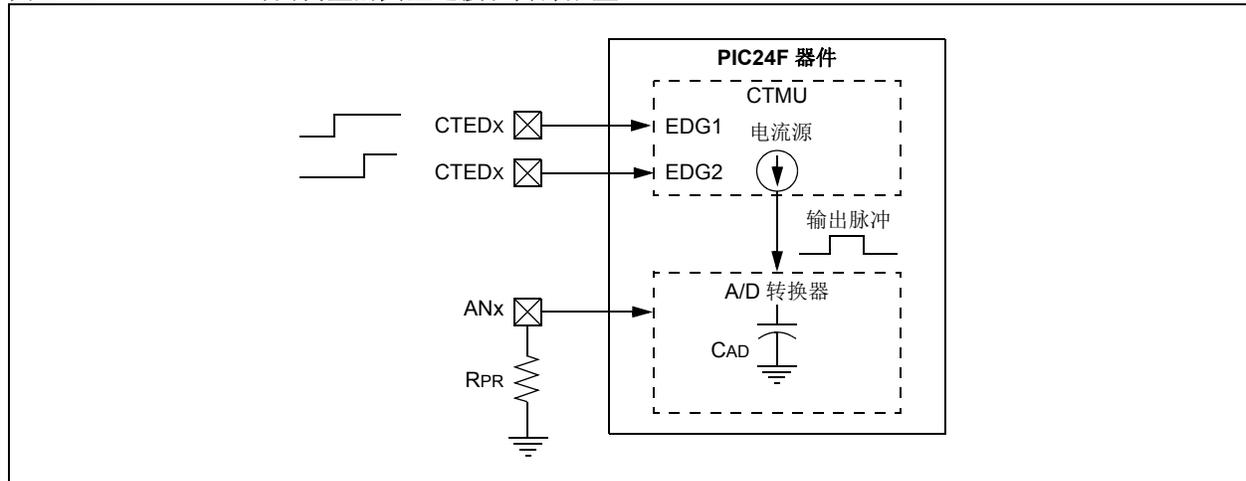
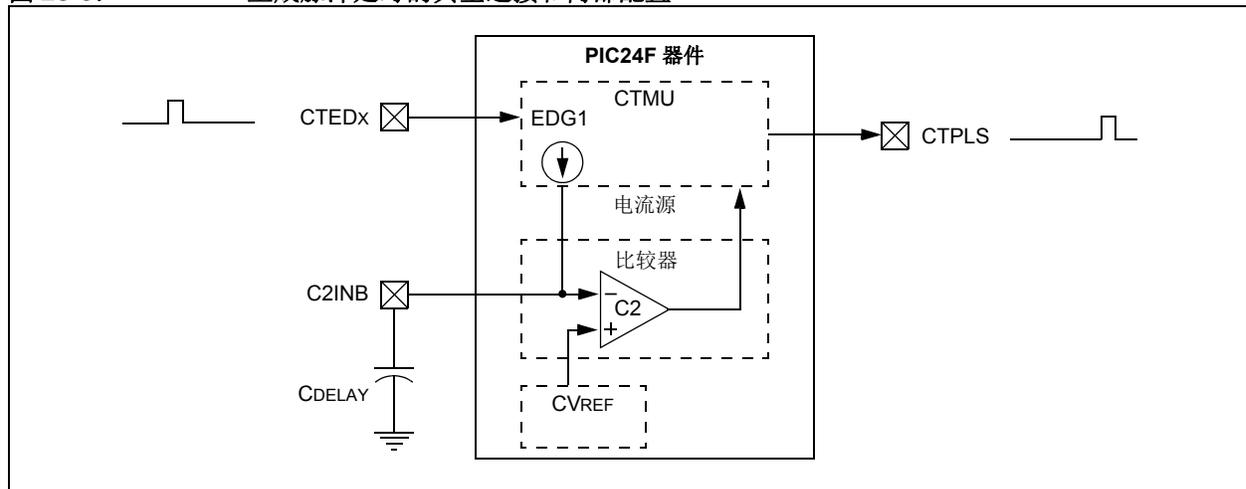


图 28-3: 生成脉冲延时的典型连接和内部配置



# PIC24FJ128GB204 系列

寄存器 28-1: CTMUCON1: CTMU 控制寄存器 1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
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        **CTMUEN:** 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 CTMU 停止位  
               1 = 当器件进入空闲模式后, 模块停止工作  
               0 = 模块在空闲模式下继续工作
- bit 12        **TGEN:** 延时生成使能位  
               1 = 使能边沿延时生成  
               0 = 禁止边沿延时生成
- bit 11        **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:** CTMU 触发器控制位  
               1 = 使能触发器输出  
               0 = 禁止触发器输出
- bit 7-0       **未实现:** 读为 0

# PIC24FJ128GB204 系列

## 寄存器 28-2: CTMUCON2: CTMU 控制寄存器 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EDG1MOD	EDG1POL	EDG1SEL3	EDG1SEL2	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
EDG2MOD	EDG2POL	EDG2SEL3	EDG2SEL2	EDG2SEL1	EDG2SEL0	—	—
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     **EDG1MOD:** 边沿 1 边沿敏感选择位  
           1 = 输入边沿敏感  
           0 = 输入电平敏感
- bit 14     **EDG1POL:** 边沿 1 极性选择位  
           1 = 边沿 1 设定为正边沿响应  
           0 = 边沿 1 设定为负边沿响应
- bit 13-10   **EDG1SEL<3:0>:** 边沿 1 源选择位  
           1111 = 边沿 1 源为比较器 3 输出  
           1110 = 边沿 1 源为比较器 2 输出  
           1101 = 边沿 1 源为比较器 1 输出  
           1100 = 边沿 1 源为 IC3  
           1011 = 边沿 1 源为 IC2  
           1010 = 边沿 1 源为 IC1  
           1001 = 边沿 1 源为 CTED8  
           1000 = 边沿 1 源为 CTED7<sup>(1)</sup>  
           0111 = 边沿 1 源为 CTED6  
           0110 = 边沿 1 源为 CTED5  
           0101 = 边沿 1 源为 CTED4  
           0100 = 边沿 1 源为 CTED3  
           0011 = 边沿 1 源为 CTED1  
           0010 = 边沿 1 源为 CTED2  
           0001 = 边沿 1 源为 OC1  
           0000 = 边沿 1 源为 Timer1
- bit 9     **EDG2STAT:** 边沿 2 状态位  
           指示边沿 2 的状态, 可写入以控制电流源。  
           1 = 已发生边沿 2  
           0 = 未发生边沿 2
- bit 8     **EDG1STAT:** 边沿 1 状态位  
           指示边沿 1 的状态, 可写入以控制电流源。  
           1 = 已发生边沿 1  
           0 = 未发生边沿 1
- bit 7     **EDG2MOD:** 边沿 2 边沿敏感选择位  
           1 = 输入边沿敏感  
           0 = 输入电平敏感
- bit 6     **EDG2POL:** 边沿 2 极性选择位  
           1 = 边沿 2 设定为正边沿  
           0 = 边沿 2 设定为负边沿

注 1: 边沿源 CTED7 在 28 引脚封装器件上不可用。

# PIC24FJ128GB204 系列

---

## 寄存器 28-2: CTMUCON2: CTMU 控制寄存器 2 (续)

bit 5-2 **EDG2SEL<3:0>**: 边沿 2 源选择位

1111 = 边沿 2 源为比较器 3 输出  
1110 = 边沿 2 源为比较器 2 输出  
1101 = 边沿 2 源为比较器 1 输出  
1100 = 未实现; 不要使用  
1011 = 边沿 2 源为 IC3  
1010 = 边沿 2 源为 IC2  
1001 = 边沿 2 源为 IC1  
1000 = 边沿 2 源为 CTED13  
0111 = 边沿 2 源为 CTED12  
0110 = 边沿 2 源为 CTED11  
0101 = 边沿 2 源为 CTED10  
0100 = 边沿 2 源为 CTED9  
0011 = 边沿 2 源为 CTED1  
0010 = 边沿 2 源为 CTED2  
0001 = 边沿 2 源为 OC1  
0000 = 边沿 2 源为 Timer1

bit 1-0 **未实现**: 读为 0

**注 1:** 边沿源 CTED7 在 28 引脚封装器件上不可用。

# PIC24FJ128GB204 系列

## 寄存器 28-3: 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 = 对标称电流的最大正向调整

011110

.

.

.

000001 = 对标称电流的最小正向调整

000000 = IRNG<1:0> 指定的标称电流输出

111111 = 对标称电流的最小负向调整

.

.

.

100010

100001 = 对标称电流的最大负向调整

bit 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 (标称值为 0.55 μA)

00 = 1000 × 基本电流

bit 7-0

**未实现**: 读为 0

# PIC24FJ128GB204 系列

---

注:

## 29.0 高 / 低电压检测 (HLVD)

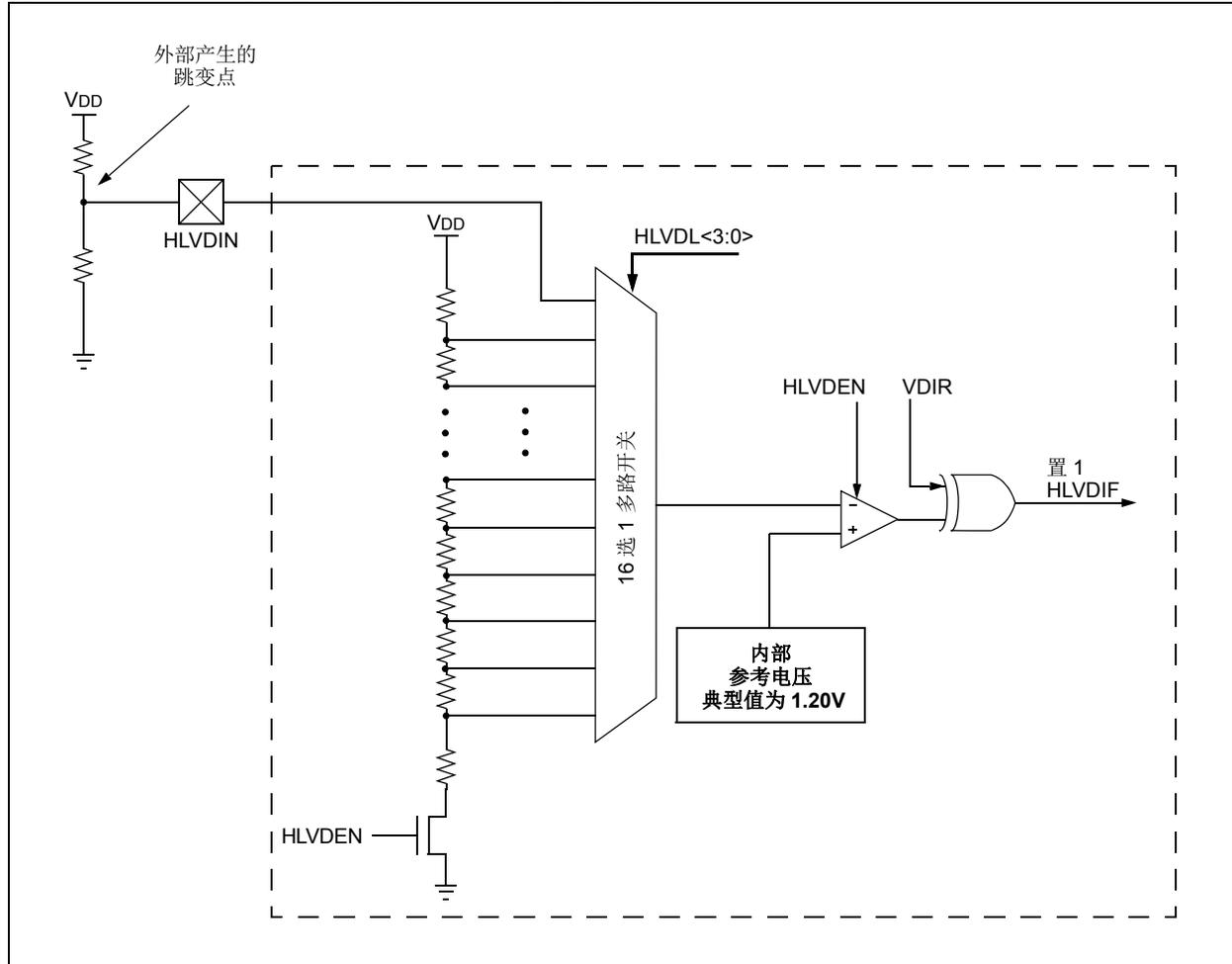
**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。关于高 / 低电压检测的更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“高度集成的可编程高 / 低电压检测 (HLVD)” (DS39725)。

如果器件电压按照特定的变化方向相对于跳变点发生了偏离，就会将中断标志位置 1。如果允许了中断，程序将跳转到中断向量地址处执行，由软件响应该中断。

HLVD 控制寄存器（见寄存器 29-1）完全控制 HLVD 模块的操作。用户可通过软件控制该寄存器将电路“关闭”，从而使器件的电流消耗降至最低。

高 / 低电压检测 (HLVD) 模块是一个可编程的电路，它允许用户指定器件的电压跳变点和电压变化方向。

图 29-1: 高 / 低电压检测 (HLVD) 模块框图



# PIC24FJ128GB204 系列

寄存器 29-1: HLVDCON: 高 / 低电压检测控制寄存器

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
HLVDEN	—	LSIDL	—	—	—	—	—
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
VDIR	BGVST	IRVST	—	HLVDL3	HLVDL2	HLVDL1	HLVDL0
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            **HLVDEN:** 高 / 低电压检测模块使能位  
                   1 = 使能 HLVD  
                   0 = 禁止 HLVD
- bit 14            **未实现:** 读为 0
- bit 13            **LSIDL:** 空闲模式 HLVD 停止位  
                   1 = 当器件进入空闲模式后, 模块停止工作  
                   0 = 模块在空闲模式下继续工作
- bit 12-8        **未实现:** 读为 0
- bit 7            **VDIR:** 电压变化方向选择位  
                   1 = 当电压等于或超过跳变点 (HLVDL<3:0>) 时发生事件  
                   0 = 当电压等于或低于跳变点 (HLVDL<3:0>) 时发生事件
- bit 6            **BGVST:** 带隙电压稳定标志位  
                   1 = 表示带隙电压稳定  
                   0 = 表示带隙电压不稳定
- bit 5            **IRVST:** 内部参考电压稳定标志位  
                   1 = 内部参考电压稳定; 高电压检测逻辑在指定的电压范围内产生中断标志  
                   0 = 内部参考电压不稳定; 高电压检测逻辑在指定的电压范围内不产生中断标志, 且不应允许 HLVD 中断
- bit 4            **未实现:** 读为 0
- bit 3-0        **HLVDL<3:0>:** 高 / 低电压检测限制位  
                   1111 = 使用外部模拟输入 (输入来自于 HLVDIN 引脚)  
                   1110 = 跳变点 1<sup>(1)</sup>  
                   1101 = 跳变点 2<sup>(1)</sup>  
                   1100 = 跳变点 3<sup>(1)</sup>  
                   .  
                   .  
                   .  
                   0100 = 跳变点 11<sup>(1)</sup>  
                   00xx = 未用

注 1: 关于实际跳变点, 请参见第 33.0 节“电气特性”。

## 30.0 特殊功能

**注：** 本数据手册总结了该PIC24F系列器件的特性。但是不应把本数据手册当作无所不包的参考资料来使用。更多信息，请参见《dsPIC33/PIC24 系列参考手册》的以下章节。本数据手册中的信息取代了FRM中的信息。

- “看门狗定时器 (WDT)” (DS39697)
- “高级器件集成” (DS39719)
- “编程和诊断” (DS39716)

PIC24FJ128GB204 系列器件具有几项特殊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。这些功能包括：

- 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 (ICSP™)
- 在线仿真 (ICE)

### 30.1 配置位

可以通过对配置位编程（读为0）或不编程（读为1）来选择不同的器件配置。这些配置位被映射到程序存储器中从F80000h开始的单元中。寄存器30-1到寄存器30-6详细说明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的范围。事实上，它属于配置存储空间（800000h-FFFFFFh），这一空间仅能通过表读和表写进行访问。

#### 30.1.1 配置 PIC24FJ128GB204 系列器件的注意事项

在PIC24FJ128GB204系列器件中，配置字节以易失性存储方式实现。这就意味着在器件每次上电时都必须对配置数据进行编程。配置数据存储在片上程序存储空间顶部的4个字中，这些字被称为闪存配置字。表30-1给出了它们的具体位置。这些字是实际器件配置位的紧凑表现形式，这些配置位实际上散布在配置空间的几个单元中。器件复位时，配置数据会被自动从闪存配置字装入相应的配置寄存器中。

**注：** 所有类型的器件复位都会重新装载配置数据。

当为这些器件创建应用程序时，用户应始终为配置数据分配特定的闪存配置字单元。此操作是为了确保在编译代码时不会把程序代码存储在该地址中。

程序存储器中的所有闪存配置字的高字节应始终为0000 0000。这样当这些单元被远程事件意外执行时，会被当作一条NOP指令来执行。由于配置位并未真正保存在对应的单元内，因此向这些单元写0不会影响器件工作。

**注：** 对程序存储器最后一页执行页擦除操作会清零闪存配置字，从而使能代码保护。因此，用户应避免对程序存储器最后一页执行页擦除操作。

**表 30-1: PIC24FJ128GB204 系列器件的闪存配置字位置**

器件	配置字地址			
	1	2	3	4
PIC24FJ64GB2XX	ABFEh	ABFCh	ABFAh	ABF8h
PIC24FJ128GB2XX	157FEh	157FCh	157FAh	157F8h

# PIC24FJ128GB204 系列

## 寄存器 30-1: 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						
—	JTAGEN	GCP	GWRP	DEBUG	LPCFG	ICS1	ICS0
bit 15						bit 8	

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
FWDTEN1	FWDTEN0	WINDIS	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7						bit 0	

<b>图注:</b>	r = 保留位	PO = 一次编程位
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 23-16 **未实现:** 读为 1
- bit 15 **保留:** 该值未知; 编程为 0
- bit 14 **JTAGEN:** JTAG 端口使能位  
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 = 对整个程序存储空间使能代码保护
- bit 12 **GWRP:** 通用段代码闪存写保护位  
1 = 允许写程序存储器  
0 = 禁止写程序存储器
- bit 11 **DEBUG:** 后台调试器使能位  
1 = 器件复位至正常工作模式  
0 = 器件复位至调试模式
- bit 10 **LPCFG:** 低压 / 保持稳压器配置位  
1 = 始终禁止低压 / 保持稳压器  
0 = 使能低功耗低压 / 保持稳压器并由固件通过 RETEN 位控制
- bit 9-8 **ICS<1:0>:** 仿真器引脚位置选择位  
11 = 仿真器功能与 PGEC1/PGED1 复用  
10 = 仿真器功能与 PGEC2/PGED2 复用  
01 = 仿真器功能与 PGEC3/PGED3 复用  
00 = 保留; 不要使用
- bit 7-6 **FWDTEN<1:0>:** 看门狗定时器配置位  
11 = 始终使能 WDT; SWDTEN 位不起作用  
10 = 使能 WDT 并由固件通过 SWDTEN 位控制  
01 = WDT 仅在运行模式下使能, 在休眠模式下禁止; 禁止 SWDTEN 位  
00 = 禁止 WDT; 禁止 SWDTEN 位
- bit 5 **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<1:0> 不得为 00)

## 寄存器 30-1: CW1: 闪存配置字 1 (续)

bit 4	<b>FWPSA:</b> WDT 预分频比选择位 1 = 预分频比为 1:128 0 = 预分频比为 1:32
bit 3-0	<b>WDTPS&lt;3:0&gt;:</b> 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

# PIC24FJ128GB204 系列

## 寄存器 30-2: 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-0	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
IESO	—	WDTCMX	ALTCMPI	ALTRB6 <sup>(2)</sup>	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	OSCIOFCN	WDTCLK1	WDTCLK0	—	POSCMD1	POSCMD0
bit 7							bit 0

<b>图注:</b>	r = 保留位	PO = 一次编程位
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 23-16 **未实现:** 读为 1

bit 15 **IESO:** 内部 / 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 **保留:** 读为 0

bit 13 **WDTCMX:** WDT 时钟多路复用控制位  
1 = 由 WDTCLK<1:0> 配置位决定 WDT 时钟源  
0 = WDT 始终使用 LPRC 作为其时钟源

bit 12 **ALTCMPI:** 备用比较器输入位  
1 = C1INC 位于 RB13 上, C2INC 位于 RB9 上且 C3INC 位于 RA0 上  
0 = C1INC、C2INC 和 C3INC 均位于 RB9 上

bit 11 **ALTRB6:** 备用 RB6 引脚功能使能位 <sup>(2)</sup>  
1 = 将 RB6 的 RP6/ASCL1/PMD6 功能附加到 RA1 引脚功能  
0 = 保持 RB6 的 RP6/ASCL1/PMD6 功能

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

**注 1:** 当选择窗口 WDT 模式且 LPRC 未用作系统时钟时, 使用 31 kHz FRC 时钟源。当器件在休眠模式下以及所有其他情况下, 使用 LPRC。

**2:** 使用 Vbus 功能时, 该配置位必须编程为 1。

## 寄存器 30-2: CW2: 闪存配置字 2 (续)

- bit 5      **OSCIOFCN:** OSCO 引脚配置位  
如果 **POSCMD<1:0> = 11 或 00:**  
1 = OSCO/CLKO/RA3 用作 CLKO ( $F_{osc}/2$ )  
0 = OSCO/CLKO/RA3 用作端口 I/O (RA3)  
如果 **POSCMD<1:0> = 10 或 01:**  
OSCIOFCN 对 OSCO/CLKO/RA3 没有影响。
- bit 4-3    **WDTCLK<1:0>:** WDT 时钟源选择位  
当 **WDTCMX = 1** 时:  
11 = LPRC  
10 = 31 kHz FRC 时钟源或 LPRC, 具体取决于器件配置 <sup>(1)</sup>  
01 = SOSC 输入  
00 = 器件工作时为系统时钟, 休眠模式下为 LPRC  
当 **WDTCMX = 0** 时:  
LPRC 始终为 WDT 时钟源。
- bit 2      **保留:** 配置为 1
- bit 1-0    **POSCMD<1:0>:** 主振荡器配置位  
11 = 禁止主振荡器模式  
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

- 注 1:** 当选择窗口 WDT 模式且 LPRC 未用作系统时钟时, 使用 31 kHz FRC 时钟源。当器件在休眠模式下以及所有其他情况下, 使用 LPRC。
- 2:** 使用 VBUS 功能时, 该配置位必须编程为 1。

# PIC24FJ128GB204 系列

## 寄存器 30-3: CW3: 闪存配置字 3

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
WPEND	WPCFG	WPDIS	BOREN	PLLSS <sup>(4)</sup>	WDTWIN1	WDTWIN0	SOSCSEL
bit 15							bit 8

r-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
—	WFPF6 <sup>(3)</sup>	WFPF5	WFPF4	WFPF3	WFPF2	WFPF1	WFPF0
bit 7							bit 0

<b>图注:</b>	r = 保留位	PO = 一次编程位
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 23-16 **未实现:** 读为 1

bit 15 **WPEND:** 段写保护结束页选择位

1 = 受保护的程序存储器代码段上边界是程序存储器的最后一页; 下边界是 WFPF<6:0> 指定的代码页  
0 = 受保护的程序存储器代码段下边界是程序存储器的末尾 (000000h); 上边界是 WFPF<6:0> 指定的代码页

bit 14 **WPCFG:** 配置字代码页写保护选择位

1 = 最后一页 (位于程序存储器开头) 和闪存配置字不受写保护<sup>(1)</sup>  
0 = 对最后一页和闪存配置字进行写保护, 前提是 WPDIS = 0

bit 13 **WPDIS:** 段写保护禁止位

1 = 禁止段程序存储器写保护  
0 = 使能段程序存储器写保护; 受保护的段由 WPEND、WPCFG 和 WFPFx 配置位定义

bit 12 **BOREN:** 欠压复位使能位

1 = 使能 BOR (除深度休眠外的所有模式)  
0 = 禁止 BOR

bit 11 **PLLSS:** PLL 辅助选择配置位<sup>(4)</sup>

1 = PLL 由主振荡器提供时钟  
0 = PLL 由片内快速 RC (FRC) 振荡器提供时钟

bit 10-9 **WDTWIN<1:0>:** 看门狗定时器窗口宽度选择位

11 = 25%  
10 = 37.5%  
01 = 50%  
00 = 75%

bit 8 **SOSCSEL:** SOSC 选择位

1 = 选择 SOSC 电路  
0 = 数字 (SCLKI) 模式<sup>(2)</sup>

**注 1:** 无论 WPCFG 状态如何, 如果 WPEND = 1 或 WFPF<6:0> 位对应配置字页, 将保护配置字页。

**注 2:** 请确保在使用该配置时, SCLKI 引脚作为数字输入引脚 (见表 11-1)。

**注 3:** 对于 64K 器件 (PIC24FJ64GB2XX), 保持 WFPF6 为 0。

**注 4:** 该配置位仅在未使用 PLL 时起作用。

## 寄存器 30-3: CW3: 闪存配置字 3 (续)

bit 7           **保留:** 始终保持为 1

bit 6-0       **WFPF<6:0>:** 写保护代码段边界页位 <sup>(3)</sup>

指定受保护代码段的 512 指令字页边界。

如果 WPEND = 1:

指定受保护代码段的下页边界; 最后一页为器件中最后实现的页。

如果 WPEND = 0:

指定受保护代码段的上页边界; Page0 为下边界。

- 注**
- 1: 无论 WPCFG 状态如何, 如果 WPEND = 1 或 WFPF<6:0> 位对应配置字页, 将保护配置字页。
  - 2: 请确保在使用该配置时, SCLKI 引脚作为数字输入引脚 (见表 11-1)。
  - 3: 对于 64K 器件 (PIC24FJ64GB2XX), 保持 WFPF6 为 0。
  - 4: 该配置位仅在未使用 PLL 时起作用。

# PIC24FJ128GB204 系列

## 寄存器 30-4: CW4: 闪存配置字 4

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-1	R/PO-1
IOL1WAY	I2C1SEL	PLLDIV3	PLLDIV2	PLLDIV1	PLLDIV0	—	DSSWEN
bit 15						bit 8	

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DSWDTEN	DSBOREN	DSWDTOSC	DSWDTPS4	DSWDTPS3	DSWDTPS2	DSWDTPS1	DSWDTPS0
bit 7						bit 0	

<b>图注:</b>	r = 保留位	PO = 一次编程位
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 23-16 **未实现:** 读为 1
- bit 15 **IOL1WAY:** IOLOCK 单次置 1 使能位
  - 1 = 解锁序列完成后, 可将 IOLOCK 位 (OSCCON<6>) 置 1 一次; 一旦置 1, 就不能再次写入外设引脚选择寄存器。
  - 0 = 解锁序列完成后, 可根据需要将 IOLOCK 位置 1 或清零。
- bit 14 **I2C1SEL:** 备用 I2C1 位置选择位
  - 1 = I2C1 使用 SCL1 和 SDA1 引脚
  - 0 = I2C1 使用 ASCL1 和 ASDA1 引脚
- bit 13-10 **PLLDIV<3:0>:** USB 96 MHz PLL 预分频比选择位
  - 1111 = 禁止 PLL
  - 1110 = 选择 8x PLL
  - 1101 = 选择 6x PLL
  - 1100 = 选择 4x PLL
  - 1011
  - .... = 保留; 不要使用
  - 1000
  - 0111 = 振荡器输入的 12 分频 (48 MHz 输入)
  - 0110 = 振荡器输入的 8 分频 (32 MHz 输入)
  - 0101 = 振荡器输入的 6 分频 (24 MHz 输入)
  - 0100 = 振荡器输入的 5 分频 (20 MHz 输入)
  - 0011 = 振荡器输入的 4 分频 (16 MHz 输入)
  - 0010 = 振荡器输入的 3 分频 (12 MHz 输入)
  - 0001 = 振荡器输入的 2 分频 (8 MHz 输入)
  - 0000 = 直接使用振荡器输入 (4 MHz 输入)
- bit 9 **保留:** 始终保持为 1
- bit 8 **DSSWEN:** 深度休眠软件控制选择位
  - 1 = 使能深度休眠操作并通过 DSEN 位控制
  - 0 = 禁止深度休眠操作
- bit 7 **DSWDTEN:** 深度休眠看门狗定时器使能位
  - 1 = 使能深度休眠 WDT
  - 0 = 禁止深度休眠 WDT

## 寄存器 30-4: CW4: 闪存配置字 4 (续)

bit 6	<b>DSBOREN:</b> 深度休眠欠压复位使能位 1 = 深度休眠模式下使能 BOR 0 = 深度休眠模式下禁止 BOR (在其他休眠模式下保持活动)
bit 5	<b>DSWDTOSC:</b> 深度休眠看门狗定时器时钟选择位 1 = 时钟源为 LPRC 0 = 时钟源为 SOSC
bit 4-0	<b>DSWDTPS&lt;4:0&gt;:</b> 深度休眠看门狗定时器后分频比选择位 11111 = 1:68,719,476,736 (25.7 天) 11110 = 1:34,359,738,368 (12.8 天) 11101 = 1:17,179,869,184 (6.4 天) 11100 = 1:8,589,934,592 (77.0 小时) 11011 = 1:4,294,967,296 (38.5 小时) 11010 = 1:2,147,483,648 (19.2 小时) 11001 = 1:1,073,741,824 (9.6 小时) 11000 = 1:536,870,912 (4.8 小时) 10111 = 1:268,435,456 (2.4 小时) 10110 = 1:134,217,728 (72.2 分钟) 10101 = 1:67,108,864 (36.1 分钟) 10100 = 1:33,554,432 (18.0 分钟) 10011 = 1:16,777,216 (9.0 分钟) 10010 = 1:8,388,608 (4.5 分钟) 10001 = 1:4,194,304 (135.3 秒) 10000 = 1:2,097,152 (67.7 秒) 01111 = 1:1,048,576 (33.825 秒) 01110 = 1:524,288 (16.912 秒) 01101 = 1:262,144 (8.456 秒) 01100 = 1:131,072 (4.228 秒) 01011 = 1:65,536 (2.114 秒) 01010 = 1:32,768 (1.057 秒) 01001 = 1:16,384 (528.5 ms) 01000 = 1:8,192 (264.3 ms) 00111 = 1:4,096 (132.1 ms) 00110 = 1:2,048 (66.1 ms) 00101 = 1:1,024 (33 ms) 00100 = 1:512 (16.5 ms) 00011 = 1:256 (8.3 ms) 00010 = 1:128 (4.1 ms) 00001 = 1:64 (2.1 ms) 00000 = 1:32 (1 ms)

# PIC24FJ128GB204 系列

## 寄存器 30-5: DEVID: 器件 ID 寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R	R	R	R	R	R	R	R
FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2	FAMID1	FAMID0
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV7	DEV6	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
bit 7							bit 0

图注: R = 可读位 U = 未实现位, 读为 1

bit 23-16 未实现: 读为 1

bit 15-8 **FAMID<7:0>**: 器件系列标识符位  
0100 1100 = PIC24FJ128GB204 系列

bit 7-0 **DEV<7:0>**: 单个器件标识符位  
0101 1000 = PIC24FJ64GB202  
0101 1010 = PIC24FJ128GB202  
0101 1001 = PIC24FJ64GB204  
0101 1011 = PIC24FJ128GB204

## 寄存器 30-6: DEVREV: 器件版本寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

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	R	R	R
—	—	—	—	REV3	REV2	REV1	REV0
bit 7							bit 0

图注: R = 可读位 U = 未实现位, 读为 1

bit 23-4 未实现: 读为 1

bit 3-0 **REV<3:0>**: 器件版本标识符位

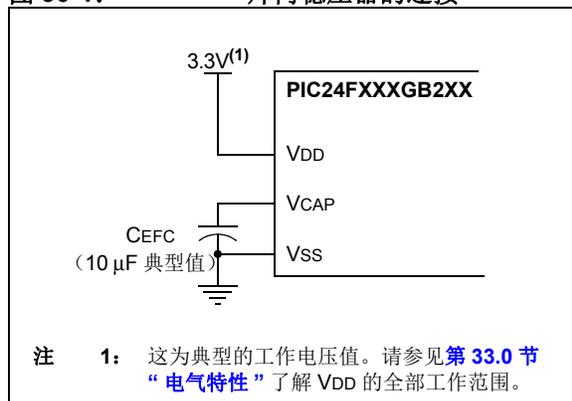
## 30.2 片内稳压器

所有 PIC24FJ128GB204 系列器件使用标称值为 1.8V 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）的设计来讲，这可能会带来问题。为简化系统设计，PIC24FJ128GB204 系列中的所有器件均包含一个片内稳压器，可使器件内核逻辑工作在 VDD 下。

始终使能该稳压器。稳压器从约为 2.1V 直至器件的 VDDMAX 的 VDD 电压，为数字内核逻辑提供标称值为 1.8V 的恒定电压。该稳压器无法将低于 2.1V 的 VDD 电压提高。为防止出现“欠压”情况，当电压对于稳压器过低时，会发生欠压复位。稳压器输出跟随 VDD，通常比 VDD 小 300 mV。

必须在 VCAP 引脚上连接一个低 ESR 电容（例如陶瓷电容）（图 30-1）。这有利于保持稳压器的稳定性。第 33.1 节“直流特性”中提供了该滤波电容（CEFC）的推荐值。

图 30-1: 片内稳压器的连接



### 30.2.1 片内稳压器和 POR

稳压器需要少量的时间来从禁止或待机状态转换为正常工作模式。在这段称为 TVREG 的时间内，禁止代码执行。每次器件在掉电（包括休眠模式）后恢复工作时都需要 TVREG。TVREG 由 VREGS 位（RCON<8>）的状态决定。更多有关 TVREG 的信息，请参见第 33.0 节“电气特性”。

**注:** 更多信息，请参见第 33.0 节“电气特性”。本数据手册中的信息将取代《dsPIC33/PIC24 系列参考手册》中的信息。

### 30.2.2 稳压器待机模式

片内稳压器除消耗 IDD/IPD 外，还总是会额外消耗一个小的电流，器件工作在休眠模式下也是如此，尽管此时内核数字逻辑并不需要耗能。为了在供电紧张的应用中节省更多的功耗，可设置稳压器在器件进入休眠模式时自动进入待机模式。VREGS（RCON<8>）位控制此特性。清零 VREGS 位，使能待机模式。稳压器需要等待 TVREG 时间才能从待机模式唤醒。

### 30.2.3 低压 / 保持稳压器

使能节能模式（例如休眠）时，PIC24FJ128GB204 系列器件可使用独立的低功耗低压 / 保持稳压器为主要电路供电。当所有其他内核数字逻辑掉电时，该稳压器以 1.2V 标称电压工作，保持为数据 RAM 和 RTCC 供电。它仅在休眠和 VBAT 模式下工作。

第 10.1.3 节“低压 / 保持稳压器”对低压 / 保持稳压器进行了详细说明。

# PIC24FJ128GB204 系列

## 30.3 看门狗定时器 (WDT)

PIC24FJ128GB204 系列器件的 WDT 是由 LPRC 振荡器驱动的。当使能 WDT 时，也将同时使能该时钟源。

由 LPRC 提供的 WDT 时钟源的频率标称值为 31 kHz。将此时钟源提供给可配置为 5 位 (32 分频) 或 7 位 (128 分频) 工作模式的预分频器。分频比由 FWPSA 配置位设置。31 kHz 的输入使预分频器生成标称的 WDT 超时周期 (TWDT) —— 5 位模式下为 1 ms, 7 位模式下为 4 ms。

分频比可变的后分频器对 WDT 预分频器的输出进行分频，从而获得更大范围的超时周期。后分频比由 WDTPS<3:0> 配置位 (CW1<3:0>) 控制，该配置位共允许选择 16 种设置，从 1:1 到 1:32,768。使用预分频器和后分频器后，可获得 1 ms 到 131 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
- 在时钟切换完成时，无论时钟切换是由软件（即改变 NOSC<sub>x</sub> 位后将 OSWEN 位置 1）或是硬件（即故障保护时钟监视器）引起
- 执行 PWRSAV 指令时（即进入休眠模式或空闲模式）
- 当器件退出休眠模式或空闲模式恢复正常工作时
- 在正常执行过程中，执行 CLRWDT 指令

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，将唤醒器件并且代码将从 PWRSAV 指令处继续执行。器件被唤醒后，需要用软件将相应的 SLEEP 或 IDLE 位 (RCON<3:2>) 清零。

WDT 标志位 WDTO (RCON<4>) 不会在 WDT 超时后自动清零。要检测后续的 WDT 事件，必须用软件将该标志清零。

**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。

### 30.3.1 窗口操作

看门狗定时器具有可选的固定窗口工作模式。在此窗口模式下，CLRWDT 指令只能在编程的窗口宽度内 (WDT 周期的 25%、37.5%、50% 或 75%) 复位 WDT，窗口宽度由 WDTWIN<1:0> 配置位 (CW3<10:9>) 控制。在该窗口前执行 CLRWDT 指令会导致 WDT 复位，这与 WDT 超时类似。

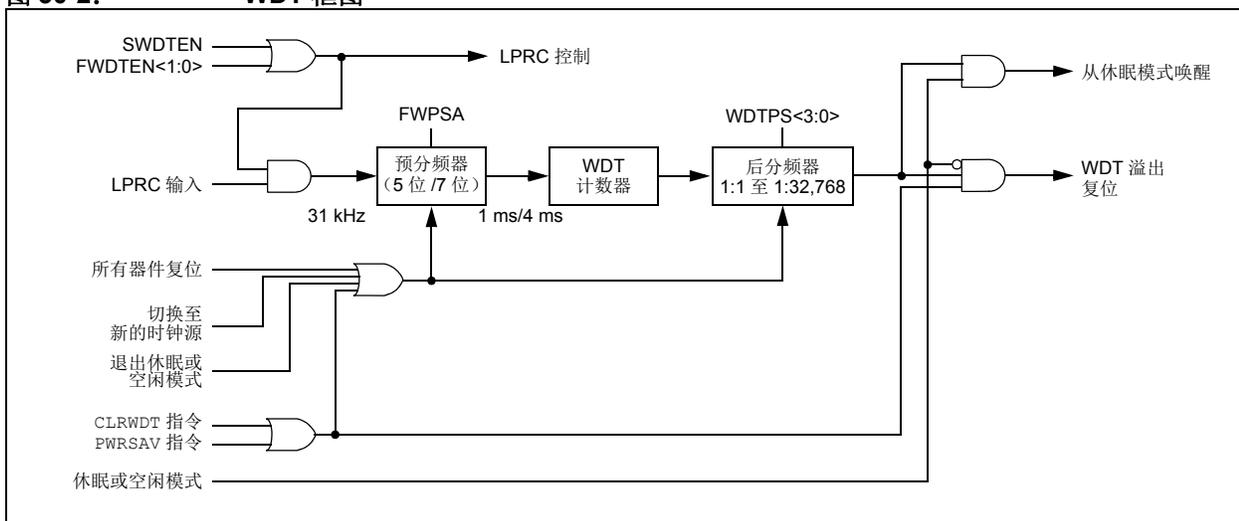
将 WINDIS 配置位 (CW1<5>) 编程为 0 使能 WDT 窗口模式。

### 30.3.2 控制寄存器

通过 FWDTEN<1:0> 配置位使能或禁止 WDT。当配置位 FWDTEN<1:0> = 11 时，始终使能 WDT。

当配置位 FWDTEN<1:0> = 10 时，可选择用软件控制 WDT。FWDTEN<1:0> = 00 时，始终禁止看门狗定时器。通过在软件中将 SWDTEN 控制位 (RCON<5>) 置 1 来使能 WDT。任何器件复位都会使 SWDTEN 控制位清零。软件 WDT 选项允许用户在关键代码段使能 WDT 并在非关键代码段禁止 WDT，以最大限度地降低功耗。

图 30-2: WDT 框图



## 30.4 程序校验和代码保护

PIC24FJ128GB204 系列器件提供了两种补充方法防止应用代码被改写和擦除。这两种方法也有助于在运行时防止器件配置被意外更改。

### 30.4.1 通用段保护

PIC24FJ128GB204 系列中所有器件的片内程序存储空间被视作一个存储区，即通用段（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写操作。但对正常的执行模式没有直接影响。

写保护由配置字 1 中的 GWRP 位控制。当 GWRP 被编程为 0 时，还禁止在内部对程序存储器执行写和擦除操作。

### 30.4.2 代码段保护

除了全局通用段保护外，还可以单独保护程序存储空间的一部分区域以防止写和擦除操作。在需要单独对某个存储区的代码进行擦除和写保护的应用程序（例如自举应用程序）中，该区域有很多用途。与常见的引导区实现方案不同，PIC24FJ128GB204 系列器件中特定的受保护段可被用户放在程序空间的任何位置并配置为各种大小。

代码段（CS）保护通过在写或擦除地址位于指定范围内时禁止 NVM 安全互锁，以此向程序存储器的指定区域提供了新增的保护等级。它不会改写 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。

段代码保护范围的大小和类型由配置字 3 的 WFPFx、WPEND、WPCFG 和 WPDIS 位配置。将 WPDIS 位编程为 0 使能代码段保护。WFPFx 位通过指定 512 字代码页（即受保护段的开始或结束位置）指定受保护的段大小。因为包含指定的存储区，所以，此页也将受到保护。

WPEND 位决定受保护段是使用程序空间的开头还是末尾作为边界。将 WPEND 编程为 0 可设置程序存储器的末尾（000000h）作为受保护段的下边界。保留 WPEND 未编程（即 = 1）将保护指定页到已实现程序存储器的最后一页（包含配置字单元）。

独立位 WPCFG 用于单独保护程序空间的最后一页（包含闪存配置字）。将 WPCFG 编程为 0 可保护最后一页，以及由 WPEND 和 WFPFx<6:0> 位设置所选择的页。这可用在需要对存储器末尾的代码段以及闪存配置字进行写保护的情况下。

表 30-2 中给出了段代码保护的各种选项。

表 30-2: 代码段保护配置选项

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都由 GCP 和 GWRP 配置。
0	1	x	擦除 / 写保护从 WFPFx<6:0> 定义的代码页的第一个地址到实现的程序存储器的末尾（包含末尾）的地址范围，包括闪存配置字。
0	0	1	擦除 / 写保护从地址 000000h 到 WFPFx<6:0> 定义的代码页的最后一个地址（包含该地址）的范围。
0	0	0	擦除 / 写保护从地址 000000h 到 WFPFx<6:0> 定义的代码页的最后一个地址（包含该地址）以及最后一页（包含闪存配置字）的范围。

# PIC24FJ128GB204 系列

---

## 30.4.3 配置寄存器保护

有两种方法保护配置寄存器使其免遭无意的或不期望的更改或读取。主要的保护方法与保护 RP 寄存器的方法相同——影子寄存器中包含了一个基准值，持续将该值与实际的值进行比较。

出于从防范不可预见事件方面的考虑，由于电池故障（如 ESD 事件）引起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，使用适当的代码段保护设置也可以保护器件配置。

## 30.5 JTAG 接口

PIC24FJ128GB204 系列器件实现了 JTAG 接口，该接口支持边界扫描器件测试和编程。

## 30.6 在线串行编程

PIC24FJ128GB204 系列单片机可以在最终的应用电路中进行串行编程。只需要 5 根线即可完成这一操作，其中时钟线（PGECx）和数据线（PGEDx）各一根，其余 3 根分别是电源线（VDD）、接地线（VSS）和 MCLR。这允许用户使用未编程器件制造电路板，仅在产品交付前才对单片机进行编程，从而可以将最新版本的固件或定制固件烧写到单片机中。

## 30.7 在线调试器

当选择 MPLAB® ICD 3 作为调试器时，使能在线调试功能。这一功能允许结合 MPLAB X IDE 进行一些简单的调试。通过 PGECx（仿真/调试时钟）和 PGEDx（仿真/调试数据）引脚控制调试功能。

要使用器件的在线调试功能，在设计中必须实现至 MCLR、VDD、VSS 和 ICSx 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 个字节和两个 I/O 引脚。

## 31.0 开发支持

一系列软件及硬件开发工具对 PIC® 单片机 (MCU) 和 dsPIC® 数字信号控制器 (DSC) 提供支持:

- 集成开发环境
  - MPLAB® X IDE 软件
- 编译器 / 汇编器 / 链接器
  - MPLAB XC 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 X SIM 软件模拟器
- 仿真器
  - MPLAB REAL ICE™ 在线仿真器
- 在线调试器 / 编程器
  - MPLAB ICD 3
  - PICKit™ 3
- 器件编程器
  - 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
- 第三方开发工具

## 31.1 MPLAB X 集成开发环境软件

MPLAB X IDE 是适用于 Microchip 和第三方软硬件开发工具统一的通用图形用户界面, 可以在 Windows®、Linux 和 Mac OS® X 上运行。MPLAB X IDE 是一款全新的 IDE, 它基于 NetBeans IDE, 包含许多免费的软件组件和插件, 适用于高性能的应用程序开发和调试。通过这一无缝交互的用户界面, 在不同工具之间的迁移以及从软件模拟器到硬件调试和编程工具的升级都变得极为简便。

MPLAB X IDE 具有完善的项目管理、可视化的调用图、可配置的观察窗口以及包含代码补全功能和上下文菜单的功能丰富编辑器, 因此对于新用户来说非常灵活和友好。MPLAB X IDE 支持对多个项目使用多个工具和同时调试, 因此也完全可以满足经验丰富用户的需求。

功能丰富的编辑器:

- 彩色高亮显示语法
- 智能代码补全功能, 在输入代码时提供建议和提示
- 基于用户定义规则, 代码自动格式化
- 即时解析

用户友好的可定制界面:

- 完全可定制界面: 工具栏、工具栏图标、窗口和窗口放置等
- 调用图窗口

基于项目的工作空间:

- 多个项目
- 多个工具
- 多种配置
- 同时调试会话

文件历史和错误跟踪:

- 本地文件历史功能
- 内建对 Bugzilla 缺陷跟踪系统的支持

# PIC24FJ128GB204 系列

## 31.2 MPLAB XC 编译器

MPLAB XC编译器是适用于Microchip所有8位、16位和32位MCU以及DSC器件的完全ANSI C编译器。这些编译器提供强大的集成功能以及出色的代码优化功能，且易于使用。MPLAB XC编译器可在Windows、Linux或Mac OS X上运行。

为方便进行源代码级调试，编译器提供了已针对MPLAB X IDE优化的调试信息。

MPLAB XC编译器的免费版支持所有器件和命令，没有时间或存储容量限制，且为大多数应用程序提供了充分的代码优化。

MPLAB XC编译器包含汇编器、链接器和实用程序。汇编器生成可重定位目标文件，然后通过链接器将生成的可重定位目标文件与其他可重定位目标文件或归档文件归档或链接在一起，进而生成可执行文件。MPLAB XC编译器使用汇编器来生成目标文件。汇编器具有如下突出特性：

- 支持全部器件指令集
- 支持定点和浮点数据
- 命令行接口
- 丰富的伪指令集
- 灵活的宏语言
- 与MPLAB X IDE兼容

## 31.3 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 X 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

## 31.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK目标链接器组合由MPASM汇编器生成的可重定位目标文件。通过使用链接器脚本中的伪指令，它还可链接预编译库中的可重定位目标文件。

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

## 31.5 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和PIC32 MCU以及dsPIC DSC器件从符号汇编语言生成可重定位机器码。MPLAB XC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
- 支持定点和浮点数据
- 命令行接口
- 丰富的指令集
- 与MPLAB X IDE兼容

## 31.6 MPLAB X SIM 软件模拟器

MPLAB X 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 X SIM 软件模拟器完全支持使用 MPLAB XC 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。

## 31.7 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 器件推出的新一代高速仿真器。结合 MPLAB X IDE 易于使用且功能强大的图形用户界面，该仿真器可对所有 8 位、16 位和 32 位 MCU 及 DSC 器件进行调试和编程。

该仿真器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与在线调试器系统兼容的连接器和新型抗噪声、高速低压差分信号 (LVDS) 互连电缆 (CAT5) 与目标板相连。

可通过 MPLAB X IDE 下载将来版本的固件，对该仿真器进行现场升级。在同类仿真器中，MPLAB REAL ICE 的优势十分明显：全速仿真、运行时变量观察、跟踪分析、复杂断点、逻辑探针、耐用的探针接口及较长（长达 3 米）的互连电缆。

## 31.8 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 的闪存 DSC 和 MCU 器件。结合 MPLAB X IDE 功能强大但易于使用的图形用户界面，该调试器可对 PIC 闪存单片机和 dsPIC DSC 进行调试和编程。

MPLAB ICD 3 在线调试器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与 MPLAB ICD 2 或 MPLAB REAL ICE 系统兼容的连接器和目标板相连。MPLAB ICD 3 支持所有 MPLAB ICD 2 连接器。

## 31.9 PICKit 3 在线调试器 / 编程器

结合 MPLAB X IDE 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 调试连接器 (RJ-11) (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 (ICSP)。

PICKit 3 Debug Express 包括 PICKit 3、演示板和单片机、连接电缆和光盘 (内含用户指南、课程、教程、编译器 and MPLAB IDE 软件)。

## 31.10 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款符合 CE 规范的通用器件编程器，在 VDDMIN 和 VDDMAX 点对其可编程电压进行校验以确保可靠性最高。它有一个用来显示菜单和错误消息的大 LCD 显示器 (128 x 64)，以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP 电缆。在单机模式下，MPLAB PM3 器件编程器不必与 PC 相连即可对 PIC MCU 和 dsPIC DSC 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。

# PIC24FJ128GB204 系列

---

## 31.11 演示 / 开发板、评估工具包及入门工具包

有许多演示、开发和评估板可用于各种 PIC MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于检查和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、RS-232 接口、LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 和 dsPICDEM™ 演示 / 开发板系列电路外，Microchip 还有一系列评估工具包和演示软件，适用于模拟滤波器设计、KEELOQ® 数据安全产品 IC、CAN、IrDA®、PowerSmart 电池管理、SEEVAL® 评估系统、 $\Sigma$ - $\Delta$  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](http://www.microchip.com))。

## 31.12 第三方开发工具

Microchip 还提供一些来自第三方供应商的优秀开发工具。这些工具均经过精心挑选，功能独特，物有所值。

- SoftLog 和 CCS 等公司提供的器件编程器和量产编程器
- Gimpel 和 Trace Systems 等公司提供的软件工具
- Saleae 和 Total Phase 等公司提供的协议分析器
- MikroElektronika、Digilent® 和 Olimex 等公司提供的演示板
- EZ Web Lynx、WIZnet 和 IPLogika® 等公司提供的嵌入式以太网解决方案

## 32.0 指令集汇总

**注：** 本章是 PIC24F 指令集架构 (ISA) 的简要汇总，并不能作为详尽的参考资料使用。

PIC24F 指令集与以前的 PIC<sup>®</sup> MCU 指令集相比，添加了许多增强功能，并保持了易于从以前的 PIC MCU 指令集移植的特点。大部分指令只占用一个程序存储字。只有 3 条指令需要两个程序存储单元。

每条单字指令都是一个 24 位字，由一个 8 位的操作码 (指明指令类型) 和一个或多个操作数 (指定指令具体操作) 组成。整个指令集具有高度的正交性，分为以下 4 种基本类型：

- 面向字或字节的操作类指令
- 面向位的操作类指令
- 立即数操作类指令
- 控制操作类指令

表 32-1 给出了在说明指令时要用到的通用符号。表 32-2 中的 PIC24F 指令集汇总列出了所有指令及每条指令影响的状态标志。

大部分面向字或字节的 W 寄存器指令 (包含桶形移位寄存器指令) 含有三个操作数：

- 第一个源操作数通常是不带地址修改符的 “Wb” 寄存器
- 第二个源操作数通常是带或不带地址修改符的 “Ws” 寄存器
- 结果的目标地址通常是带或不带地址修改符的 “Wd” 寄存器

而面向字或字节的文件寄存器指令具有两个操作数：

- 文件寄存器 (由 “f” 值指定)
- 目标寄存器 (可以是文件寄存器 “f”，也可以是记作 “WREG” 的 W0 寄存器)

大部分面向位的操作类指令 (包括简单循环/移位指令) 含有两个操作数：

- W 寄存器 (带或不带地址修改符) 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 (由立即数直接指定或由 “Wb” 寄存器中的内容间接指定)

涉及数据传送的立即数指令可能使用以下操作数：

- 将被装载到 W 寄存器或文件寄存器的立即数值 (由 “k” 的值指定)
- 将要装载立即数值的 W 寄存器或文件寄存器 (由 “Wb” 或 “f” 指定)

而涉及算术或逻辑运算的立即数指令使用以下操作数：

- 第一个源操作数通常是不带地址修改符的 “Wb” 寄存器
- 第二个源操作数是立即数
- 结果的目标地址 (只有在与第一个源操作数不同的情况下) 通常为带或不带地址修改符的 “Wd” 寄存器

控制操作类指令可使用以下操作数：

- 程序存储器地址
- 表读和表写指令的模式

除某些双字指令外所有指令都是单字指令。双字指令中所有必需的信息都在这 48 位中，第二个字的高 8 位全为 0。如果指令自身将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。

除非条件测试结果为 true 或者指令执行后改变了程序计数器的值，否则执行大多数单字指令都只需要一个指令周期。对于上述两种特殊情况，指令执行需要两个指令周期，在第二个指令周期中执行一条 NOP 指令。值得注意的特殊指令有：BRA (无条件 / 计算转移指令)、间接 CALL/GOTO 指令、所有表读和表写指令以及 RETURN/RETFIE 指令，这些指令都是单字指令，但执行起来需要 2 或 3 个指令周期。

某些涉及跳过下一条指令的指令，在执行跳过时需要两或三个指令周期，具体周期数取决于被跳过的指令是单字指令还是双字指令。此外需要传送两个字的指令需要两个周期。执行双字指令需要两个指令周期。

# PIC24FJ128GB204 系列

表 32-1: 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“地址为 text 的单元”
{ }	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（默认情况）
bit4	4 位位选择字段（用于字寻址指令） $\in \{0...15\}$
C、DC、N、OV 和 Z	MCU 状态位：进位、半进位、负标志、溢出标志和全零标志
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，字节模式下， $\in \{0...255\}$ ，字模式下 $\in \{0:1023\}$
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	该字段不必有输入项，可以为空白
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 $\in \{Wns, [Wns], [Wns++] , [Wns--], [++Wns], [--Wns], [Wns+Wb] \}$

# PIC24FJ128GB204 系列

表 32-2: 指令集概述

汇编指令 助记符	汇编语法	说明	字数	周期数	受影响的状态标志
ADD	ADD f	$f = f + WREG$	1	1	C、DC、N、OV 和 Z
	ADD f, WREG	$WREG = f + WREG$	1	1	C、DC、N、OV 和 Z
	ADD #lit10, Wn	$Wd = lit10 + Wd$	1	1	C、DC、N、OV 和 Z
	ADD Wb, Ws, Wd	$Wd = Wb + Ws$	1	1	C、DC、N、OV 和 Z
	ADD Wb, #lit5, Wd	$Wd = Wb + lit5$	1	1	C、DC、N、OV 和 Z
ADDC	ADDC f	$f = f + WREG + (C)$	1	1	C、DC、N、OV 和 Z
	ADDC f, WREG	$WREG = f + WREG + (C)$	1	1	C、DC、N、OV 和 Z
	ADDC #lit10, Wn	$Wd = lit10 + Wd + (C)$	1	1	C、DC、N、OV 和 Z
	ADDC Wb, Ws, Wd	$Wd = Wb + Ws + (C)$	1	1	C、DC、N、OV 和 Z
	ADDC Wb, #lit5, Wd	$Wd = Wb + lit5 + (C)$	1	1	C、DC、N、OV 和 Z
AND	AND f	$f = f .AND. WREG$	1	1	N 和 Z
	AND f, WREG	$WREG = f .AND. WREG$	1	1	N 和 Z
	AND #lit10, Wn	$Wd = lit10 .AND. Wd$	1	1	N 和 Z
	AND Wb, Ws, Wd	$Wd = Wb .AND. Ws$	1	1	N 和 Z
	AND Wb, #lit5, Wd	$Wd = Wb .AND. lit5$	1	1	N 和 Z
ASR	ASR f	f = 算术右移 f	1	1	C、N、OV 和 Z
	ASR f, WREG	WREG = 算术右移 f	1	1	C、N、OV 和 Z
	ASR Ws, Wd	Wd = 算术右移 Ws	1	1	C、N、OV 和 Z
	ASR Wb, Wns, Wnd	Wnd = 将 Wb 算术右移 Wns 位	1	1	N 和 Z
	ASR Wb, #lit5, Wnd	Wnd = 将 Wb 算术右移 lit5 位	1	1	N 和 Z
BCLR	BCLR f, #bit4	将 f 中的某位清零	1	1	无
	BCLR Ws, #bit4	将 Ws 中的指定位清零	1	1	无
BRA	BRA C, Expr	进位则跳转	1	1 (2)	无
	BRA GE, Expr	如果大于或等于则跳转	1	1 (2)	无
	BRA GEU, Expr	如果无符号大于或等于则跳转	1	1 (2)	无
	BRA GT, Expr	如果大于则跳转	1	1 (2)	无
	BRA GTU, Expr	如果无符号大于则跳转	1	1 (2)	无
	BRA LE, Expr	如果小于或等于则跳转	1	1 (2)	无
	BRA LEU, Expr	如果无符号小于或等于则跳转	1	1 (2)	无
	BRA LT, Expr	如果小于则跳转	1	1 (2)	无
	BRA LTU, Expr	如果无符号小于则跳转	1	1 (2)	无
	BRA N, Expr	为负则跳转	1	1 (2)	无
	BRA NC, Expr	无进位则跳转	1	1 (2)	无
	BRA NN, Expr	不为负则跳转	1	1 (2)	无
	BRA NOV, Expr	不溢出则跳转	1	1 (2)	无
	BRA NZ, Expr	不为零则跳转	1	1 (2)	无
	BRA OV, Expr	溢出则跳转	1	1 (2)	无
	BRA Expr	无条件跳转	1	2	无
	BRA Z, Expr	为零则跳转	1	1 (2)	无
BRA Wn	相对跳转	1	2	无	
BSET	BSET f, #bit4	将 f 中的某位置 1	1	1	无
	BSET Ws, #bit4	将 Ws 中的某位置 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 中的某位取反	1	1	无
	BTG Ws, #bit4	将 Ws 中的某位取反	1	1	无
BTSC	BTSC f, #bit4	对 f 中的某位进行检测, 如果为零则跳过	1	1 (2 或 3)	无
	BTSC Ws, #bit4	对 Ws 中的某位进行检测, 如果为零则跳过	1	1 (2 或 3)	无

# PIC24FJ128GB204 系列

表 32-2: 指令集概述 (续)

汇编指令助记符	汇编语法	说明	字数	周期数	受影响的状态标志
BTSS	BTSS $f, \#bit4$	对 $f$ 中的某位进行检测, 如果为 1 则跳过	1	1 (2 或 3)	无
	BTSS $Ws, \#bit4$	对 $Ws$ 中的某位进行检测, 如果为 1 则跳过	1	1 (2 或 3)	无
BTST	BTST $f, \#bit4$	对 $f$ 中的某位进行检测	1	1	Z
	BTST.C $Ws, \#bit4$	对 $Ws$ 中的某位进行检测, 并将结果存储到进位标志位 C	1	1	C
	BTST.Z $Ws, \#bit4$	对 $Ws$ 中的某位进行检测, 并将结果存储到全零标志位 Z	1	1	Z
	BTST.C $Ws, Wb$	对 $Ws<Wb>$ 位进行检测, 并将结果存储到进位标志位 C	1	1	C
	BTST.Z $Ws, Wb$	对 $Ws<Wb>$ 位进行检测, 并将结果存储到全零标志位 Z	1	1	Z
BTSTS	BTSTS $f, \#bit4$	对 $f$ 中的某位进行检测, 并将该位置 1	1	1	Z
	BTSTS.C $Ws, \#bit4$	对 $Ws$ 中的某位进行检测, 并将结果存储到进位标志位 C 中, 然后将 $Ws$ 中的该位置 1	1	1	C
	BTSTS.Z $Ws, \#bit4$	对 $Ws$ 中的某位进行检测, 并将结果存储到全零标志位 Z 中, 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
CLRWDTC	CLRWDTC	将看门狗定时器清零	1	1	WDTO 和 Sleep
COM	COM $f$	$f = \bar{f}$	1	1	N 和 Z
	COM $f, WREG$	WREG = $\bar{f}$	1	1	N 和 Z
	COM $Ws, Wd$	$Wd = \bar{Ws}$	1	1	N 和 Z
CP	CP $f$	比较 $f$ 和 WREG	1	1	C、DC、N、OV 和 Z
	CP $Wb, \#lit5$	比较 $Wb$ 和 $lit5$	1	1	C、DC、N、OV 和 Z
	CP $Wb, Ws$	比较 $Wb$ 和 $Ws$ ( $Wb - Ws$ )	1	1	C、DC、N、OV 和 Z
CP0	CP0 $f$	比较 $f$ 和 0x0000	1	1	C、DC、N、OV 和 Z
	CP0 $Ws$	比较 $Ws$ 和 0x0000	1	1	C、DC、N、OV 和 Z
CPB	CPB $f$	比较 $f$ 和 WREG (通过带借位减法实现)	1	1	C、DC、N、OV 和 Z
	CPB $Wb, \#lit5$	比较 $Wb$ 和 $lit5$ (通过带借位减法实现)	1	1	C、DC、N、OV 和 Z
	CPB $Wb, Ws$	比较 $Wb$ 和 $Ws$ (通过带借位减法实现) ( $Wb - Ws - C$ )	1	1	C、DC、N、OV 和 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 $Wn$	$Wn =$ 对 $Wn$ 进行十进制调整	1	1	C
DEC	DEC $f$	$f = f - 1$	1	1	C、DC、N、OV 和 Z
	DEC $f, WREG$	WREG = $f - 1$	1	1	C、DC、N、OV 和 Z
	DEC $Ws, Wd$	$Wd = Ws - 1$	1	1	C、DC、N、OV 和 Z
DEC2	DEC2 $f$	$f = f - 2$	1	1	C、DC、N、OV 和 Z
	DEC2 $f, WREG$	WREG = $f - 2$	1	1	C、DC、N、OV 和 Z
	DEC2 $Ws, Wd$	$Wd = Ws - 2$	1	1	C、DC、N、OV 和 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 和 OV
	DIV.SD $Wm, Wn$	有符号 32/16 位整数除法	1	18	N、Z、C 和 OV
	DIV.UW $Wm, Wn$	无符号 16/16 位整数除法	1	18	N、Z、C 和 OV
	DIV.UD $Wm, Wn$	无符号 32/16 位整数除法	1	18	N、Z、C 和 OV
EXCH	EXCH $Wns, Wnd$	将 $Wns$ 和 $Wnd$ 交换	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 (MSb) 查找第一个 1	1	1	C

# PIC24FJ128GB204 系列

表 32-2: 指令集概述 (续)

汇编指令助记符	汇编语法	说明	字数	周期数	受影响的状态标志
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 和 Z
	INC f, WREG	$WREG = f + 1$	1	1	C、DC、N、OV 和 Z
	INC Ws, Wd	$Wd = Ws + 1$	1	1	C、DC、N、OV 和 Z
INC2	INC2 f	$f = f + 2$	1	1	C、DC、N、OV 和 Z
	INC2 f, WREG	$WREG = f + 2$	1	1	C、DC、N、OV 和 Z
	INC2 Ws, Wd	$Wd = Ws + 2$	1	1	C、DC、N、OV 和 Z
IOR	IOR f	$f = f .IOR. WREG$	1	1	N 和 Z
	IOR f, WREG	$WREG = f .IOR. WREG$	1	1	N 和 Z
	IOR #lit10, Wn	$Wd = lit10 .IOR. Wd$	1	1	N 和 Z
	IOR Wb, Ws, Wd	$Wd = Wb .IOR. Ws$	1	1	N 和 Z
	IOR Wb, #lit5, Wd	$Wd = Wb .IOR. lit5$	1	1	N 和 Z
LNK	LNK #lit14	链接帧指针	1	1	无
LSR	LSR f	$f =$ 逻辑右移 $f$	1	1	C、N、OV 和 Z
	LSR f, WREG	$WREG =$ 逻辑右移 $f$	1	1	C、N、OV 和 Z
	LSR Ws, Wd	$Wd =$ 逻辑右移 $Ws$	1	1	C、N、OV 和 Z
	LSR Wb, Wns, Wnd	$Wnd =$ 将 $Wb$ 逻辑右移 $Wns$ 位	1	1	N 和 Z
	LSR Wb, #lit5, Wnd	$Wnd =$ 将 $Wb$ 逻辑右移 $lit5$ 位	1	1	N 和 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	将源寄存器的内容送入目标寄存器	1	1	N 和 Z
	MOV f, WREG	将 $f$ 中的内容送入 $WREG$	1	1	N 和 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 和 Z
	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	$f = \bar{f} + 1$	1	1	C、DC、N、OV 和 Z
	NEG f, WREG	$WREG = \bar{f} + 1$	1	1	C、DC、N、OV 和 Z
	NEG Ws, Wd	$Wd = \overline{Ws} + 1$	1	1	C、DC、N、OV 和 Z
NOP	NOP	空操作	1	1	无
	NOPR	空操作	1	1	无
POP	POP f	从栈顶 (TOS) 弹出 $f$ 寄存器的内容	1	1	无
	POP Wdo	将栈顶 (TOS) 的内容弹出到 $Wdo$ 中	1	1	无
	POP.D Wnd	将栈顶 (TOS) 的内容弹出到 $W(nd):W(nd+1)$ 中	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 $W(ns):W(ns + 1)$ 中的内容压入栈顶 (TOS)	1	2	无
	PUSH.S	将主寄存器中的内容压入影子寄存器	1	1	无

# PIC24FJ128GB204 系列

表 32-2: 指令集概述 (续)

汇编指令助记符	汇编语法	说明	字数	周期数	受影响的状态标志
PWRSVAV	PWRSVAV #lit1	进入休眠或空闲模式	1	1	WDTO 和 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	用软件使器件复位	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 和 Z
	RLC f, WREG	WREG = 对 f 执行带进位的循环左移	1	1	C、N 和 Z
	RLC Ws, Wd	Wd = 对 Ws 执行带进位的循环左移	1	1	C、N 和 Z
RLNC	RLNC f	f = 循环左移 f (不带进位)	1	1	N 和 Z
	RLNC f, WREG	WREG = 循环左移 f (不带进位)	1	1	N 和 Z
	RLNC Ws, Wd	Wd = 循环左移 Ws (不带进位)	1	1	N 和 Z
RRC	RRC f	f = 对 f 执行带进位的循环右移	1	1	C、N 和 Z
	RRC f, WREG	WREG = 对 f 执行带进位的循环右移	1	1	C、N 和 Z
	RRC Ws, Wd	Wd = 对 Ws 执行带进位的循环右移	1	1	C、N 和 Z
RRNC	RRNC f	f = 循环右移 f (不带进位)	1	1	N 和 Z
	RRNC f, WREG	WREG = 循环右移 f (不带进位)	1	1	N 和 Z
	RRNC Ws, Wd	Wd = 循环右移 Ws (不带进位)	1	1	N 和 Z
SE	SE Ws, Wnd	Wnd = 对 Ws 进行符号扩展	1	1	C、N 和 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 和 Z
	SL f, WREG	WREG = 左移 f	1	1	C、N、OV 和 Z
	SL Ws, Wd	Wd = 左移 Ws	1	1	C、N、OV 和 Z
	SL Wb, Wns, Wnd	Wnd = 将 Wb 左移 Wns 位	1	1	N 和 Z
	SL Wb, #lit5, Wnd	Wnd = 将 Wb 左移 lit5 位	1	1	N 和 Z
SUB	SUB f	f = f - WREG	1	1	C、DC、N、OV 和 Z
	SUB f, WREG	WREG = f - WREG	1	1	C、DC、N、OV 和 Z
	SUB #lit10, Wn	Wn = Wn - lit10	1	1	C、DC、N、OV 和 Z
	SUB Wb, Ws, Wd	Wd = Wb - Ws	1	1	C、DC、N、OV 和 Z
	SUB Wb, #lit5, Wd	Wd = Wb - lit5	1	1	C、DC、N、OV 和 Z
SUBB	SUBB f	f = f - WREG - ( $\bar{C}$ )	1	1	C、DC、N、OV 和 Z
	SUBB f, WREG	WREG = f - WREG - ( $\bar{C}$ )	1	1	C、DC、N、OV 和 Z
	SUBB #lit10, Wn	Wn = Wn - lit10 - ( $\bar{C}$ )	1	1	C、DC、N、OV 和 Z
	SUBB Wb, Ws, Wd	Wd = Wb - Ws - ( $\bar{C}$ )	1	1	C、DC、N、OV 和 Z
	SUBB Wb, #lit5, Wd	Wd = Wb - lit5 - ( $\bar{C}$ )	1	1	C、DC、N、OV 和 Z
SUBR	SUBR f	f = WREG - f	1	1	C、DC、N、OV 和 Z
	SUBR f, WREG	WREG = WREG - f	1	1	C、DC、N、OV 和 Z
	SUBR Wb, Ws, Wd	Wd = Ws - Wb	1	1	C、DC、N、OV 和 Z
	SUBR Wb, #lit5, Wd	Wd = lit5 - Wb	1	1	C、DC、N、OV 和 Z
SUBBR	SUBBR f	f = WREG - f - ( $\bar{C}$ )	1	1	C、DC、N、OV 和 Z
	SUBBR f, WREG	WREG = WREG - f - ( $\bar{C}$ )	1	1	C、DC、N、OV 和 Z
	SUBBR Wb, Ws, Wd	Wd = Ws - Wb - ( $\bar{C}$ )	1	1	C、DC、N、OV 和 Z
	SUBBR Wb, #lit5, Wd	Wd = lit5 - Wb - ( $\bar{C}$ )	1	1	C、DC、N、OV 和 Z
SWAP	SWAP.b Wn	Wn = 将 Wn 的两个半字节相交换	1	1	无
	SWAP Wn	Wn = 将 Wn 的两个字节相交换	1	1	无

**表 32-2: 指令集概述 (续)**

汇编指令 助记符	汇编语法	说明	字数	周期数	受影响的状态标志
TBLRDH	TBLRDH <i>Ws, Wd</i>	将程序计数器的 <23:16> 读入 <i>Wd</i> <7:0>	1	2	无
TBLRDL	TBLRDL <i>Ws, Wd</i>	将程序计数器的 <15:0> 读入 <i>Wd</i>	1	2	无
TBLWTH	TBLWTH <i>Ws, Wd</i>	将 <i>Ws</i> <7:0> 写入程序计数器的 <23:16>	1	2	无
TBLWTL	TBLWTL <i>Ws, Wd</i>	将 <i>Ws</i> 写入程序计数器的 <15:0>	1	2	无
ULNK	ULNK	释放堆栈帧	1	1	无
XOR	XOR <i>f</i>	$f = f .XOR. WREG$	1	1	N 和 Z
	XOR <i>f, WREG</i>	$WREG = f .XOR. WREG$	1	1	N 和 Z
	XOR #lit10, <i>Wn</i>	$Wd = lit10 .XOR. Wd$	1	1	N 和 Z
	XOR <i>Wb, Ws, Wd</i>	$Wd = Wb .XOR. Ws$	1	1	N 和 Z
	XOR <i>Wb, #lit5, Wd</i>	$Wd = Wb .XOR. lit5$	1	1	N 和 Z
ZE	ZE <i>Ws, Wnd</i>	<i>Wnd</i> = 对 <i>Ws</i> 进行零扩展	1	1	C、Z 和 N

# PIC24FJ128GB204 系列

---

注:

## 33.0 电气特性

本章提供 PIC24FJ128GB204 系列器件电气特性的概述。在文档后续版本中会添加其他信息。

下面列出了 PIC24FJ128GB204 系列器件的绝对最大值。器件长时间工作在最大值条件下，其稳定性会受到影响。我们建议不要使器件在该规范规定的参数范围外工作。

### 绝对最大值 †

环境温度.....	-40°C 至 +100°C
存储温度.....	-65°C 至 +150°C
V <sub>DD</sub> 引脚相对于 V <sub>SS</sub> 的电压.....	-0.3V 至 +4.0V
任何不能承受 5.5V 电压的通用数字或模拟引脚相对于 V <sub>SS</sub> 的电压.....	-0.3V 至 (V <sub>DD</sub> + 0.3V)
任何可承受 5.5V 电压的通用数字或模拟引脚（包括 <u>MCLR</u> ）相对于 V <sub>SS</sub> 的电压:	
当 V <sub>DD</sub> = 0V 时: .....	-0.3V 至 +4.0V
当 V <sub>DD</sub> ≥ 2.0V 时: .....	-0.3V 至 +6.0V
AV <sub>DD</sub> 引脚相对于 V <sub>SS</sub> 的电压.....	(V <sub>DD</sub> - 0.3V) 至 (4.0V 或 (V <sub>DD</sub> + 0.3V) 中的较小者)
AV <sub>SS</sub> 引脚相对于 V <sub>SS</sub> 的电压.....	-0.3V 至 +0.3V
VBAT 引脚相对于 V <sub>SS</sub> 的电压.....	-0.3V 至 +4.0V
V <sub>USB3V3</sub> 引脚相对于 V <sub>SS</sub> 的电压.....	(V <sub>CAP</sub> - 0.3V) 至 +4.0V
V <sub>BUS</sub> 引脚相对于 V <sub>SS</sub> 的电压.....	-0.3V 至 +6.0V
D+ 或 D- 引脚相对于 V <sub>SS</sub> 的电压:	
(信号源阻抗为 0Ω) (注 1) .....	-0.5V 至 (V <sub>USB3V3</sub> + 0.5V)
(信号源阻抗 ≥ 28Ω, V <sub>USB3V3</sub> ≥ 3.0V) .....	-1.0V 至 +4.6V
V <sub>SS</sub> 引脚的最大输出电流.....	300 mA
V <sub>DD</sub> 引脚的最大输入电流 (注 2) .....	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 2) .....	200 mA

- 注 1:** 原始 USB 2.0 规范指示 USB 器件应能够承受 D+ 或 D- 到 V<sub>BUS</sub> 之间电压的 24 小时短路。后来，作为 USB 规范的补充材料，工程变更通知 (Engineering Change Notice, ECN) 删除了该要求，并取代了原始规范。PIC24FJ128GB204 系列器件通常能够承受这种短路测试，但是建议遵守此处规定的绝对最大值以避免损坏器件。
- 注 2:** 允许的最大电流由器件的最大功耗决定 (见表 33-1)。

† 注: 如果器件工作条件超过上述“绝对最大值”，可能引起器件永久性损坏。上述值仅为运行条件最大值，我们建议不要使器件在该规范规定的范围以外运行。器件长时间工作在最大值条件下，其稳定性可能会受到影响。

# PIC24FJ128GB204 系列

## 33.1 直流特性

图 33-1: PIC24FJ128GB204 系列器件电压——频率关系图（工业级）

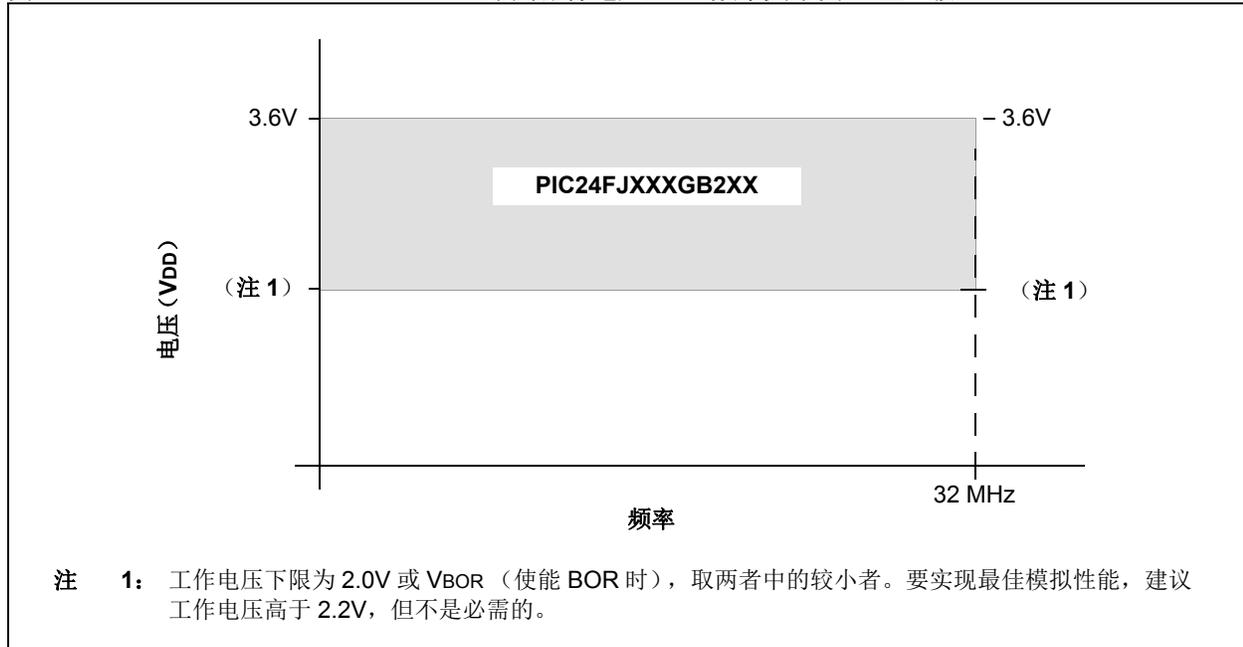


表 33-1: 热工作条件

额定值	符号	最小值	典型值	最大值	单位
PIC24FJ128GB204:					
工作结温范围	T <sub>J</sub>	-40	—	+125	°C
工作环境温度范围	T <sub>A</sub>	-40	—	+85	°C
功耗:					
内部芯片功耗: P <sub>INT</sub> = V <sub>DD</sub> × (I <sub>DD</sub> - Σ I <sub>OH</sub> )	P <sub>D</sub>	P <sub>INT</sub> + P <sub>I/O</sub>			W
I/O 引脚功耗: P <sub>I/O</sub> = Σ ({V <sub>DD</sub> - V <sub>OH</sub> } × I <sub>OH</sub> ) + Σ (V <sub>OL</sub> × I <sub>OL</sub> )					
允许的最大功耗	P <sub>D</sub> MAX	(T <sub>J</sub> - T <sub>A</sub> )/θ <sub>JA</sub>			W

表 33-2: 封装热阻特性

特性	符号	典型值	最大值	单位	注
封装热阻, 7.50 mm 28 引脚 SOIC	θ <sub>JA</sub>	49	—	°C/W	(注 1)
封装热阻, 6x6x0.9 mm 28 引脚 QFN-S	θ <sub>JA</sub>	33.7	—	°C/W	(注 1)
封装热阻, 8x8 mm 44 引脚 QFN	θ <sub>JA</sub>	28	—	°C/W	(注 1)
封装热阻, 10x10x1 mm 44 引脚 TQFP	θ <sub>JA</sub>	39.3	—	°C/W	(注 1)
封装热阻, 5.30 mm 28 引脚 SSOP	θ <sub>JA</sub>	—	—	°C/W	(注 1)
封装热阻, 300 mil 28 引脚 SPDIP	θ <sub>JA</sub>	—	—	°C/W	(注 1)

注 1: 通过封装模拟获得结点与环境的热阻值 Theta-JA (θ<sub>JA</sub>)。

# PIC24FJ128GB204 系列

表 33-3: 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 ≤ T <sub>A</sub> ≤ +85°C (工业级)				
			-40°C ≤ T <sub>A</sub> ≤ +125°C (扩展级)				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
工作电压							
DC10	V <sub>DD</sub>	电源电压	2.0	—	3.6	V	禁止 BOR
			V <sub>BOR</sub>	—	3.6	V	使能 BOR
DC12	V <sub>DR</sub>	RAM 数据保持电压 <sup>(1)</sup>	取如下二者中的较大值: V <sub>PORREL</sub> 或 V <sub>BOR</sub>	—	—	V	仅当使能 BOR (BOREN = 1) 时使用 V <sub>BOR</sub>
DC16	V <sub>POR</sub>	V <sub>DD</sub> 启动电压 确保内部上电复位信号	V <sub>SS</sub>	—	—	V	(注 2)
DC16A	V <sub>PORREL</sub>	V <sub>DD</sub> 上电复位释放电压	1.80	1.88	1.95	V	(注 3)
DC17A	SRV <sub>DD</sub>	建议的 V <sub>DD</sub> 上升速率 确保内部上电复位信号	0.05	—	—	V/ms	66 ms 内电压变化范围为 0-3.3V 50 ms 内电压变化范围为 0-2.5V (注 2)
DC17B	V <sub>BOR</sub>	V <sub>DD</sub> 由高到低跳变时的 欠压复位电压	2.0	2.1	2.2	V	(注 3)

注 1: 这是 V<sub>DD</sub> 的下限值, 将始终保持 RAM 内容。

注 2: 如果无法满足 V<sub>POR</sub> 或 SRV<sub>DD</sub> 参数, 或应用遇到缓慢掉电 V<sub>DD</sub> 速率, 建议使能和使用 BOR。

注 3: 在上升的 V<sub>DD</sub> 上电序列中, 应用固件以 V<sub>PORREL</sub> 或 V<sub>BOR</sub> 电平 (BOREN = 1 时) 中的较高者开始执行。

# PIC24FJ128GB204 系列

表 33-4: 直流特性: 工作电流 (IDD)

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	典型值 (1)	最大值	单位	工作温度	VDD	条件
<b>工作电流 (IDD) (2)</b>						
DC19	0.20	0.28	mA	-40°C 至 +125°C	2.0V	0.5 MIPS, Fosc = 1 MHz
DC20A	0.21	0.28	mA	-40°C 至 +125°C	3.3V	
DC20	0.38	0.52	mA	-40°C 至 +125°C	2.0V	1 MIPS, Fosc = 2 MHz
	0.39	0.52	mA	-40°C 至 +125°C	3.3V	
DC23	1.5	2.0	mA	-40°C 至 +125°C	2.0V	4 MIPS, Fosc = 8 MHz
	1.5	2.0	mA	-40°C 至 +125°C	3.3V	
DC24	5.6	7.6	mA	-40°C 至 +125°C	2.0V	16 MIPS, Fosc = 32 MHz
	5.7	7.6	mA	-40°C 至 +125°C	3.3V	
DC31	23	78	μA	-40°C 至 +85°C	2.0V	LPRC (15.5 KIPS), Fosc = 31 kHz
	—	98	μA	+125°C	2.0V	
	25	80	μA	-40°C 至 +85°C	3.3V	
	—	100	μA	+125°C	3.3V	

注 1: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

注 2: 所有 IDD 测量的测试条件为: 使用满幅的外部方波驱动 OSC1。所有 I/O 引脚配置为输出且驱动为 Vss。MCLR = VDD, WDT 和 FSCM 被禁止。CPU、程序存储器和数据存储器正常工作。外设模块均不工作; 但锁定每个外设 (PMDx 位全部清零)。

# PIC24FJ128GB204 系列

表 33-5: 直流特性: 空闲电流 (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)			
			-40°C ≤ TA ≤ +125°C (扩展级)			
参数编号	典型值 <sup>(1)</sup>	最大值	单位	工作温度	VDD	条件
<b>空闲电流 (IDLE)<sup>(2)</sup></b>						
DC40	116	150	μA	-40°C 至 +85°C	2.0V	1 MIPS, Fosc = 2 MHz
	—	170	μA	+125°C	2.0V	
	123	160	μA	-40°C 至 +85°C	3.3V	
	—	180	μA	+125°C	3.3V	
DC43	0.39	0.5	mA	-40°C 至 +85°C	2.0V	4 MIPS, Fosc = 8 MHz
	—	0.52	mA	+125°C	2.0V	
	0.41	0.54	mA	-40°C 至 +85°C	3.3V	
	—	0.56	mA	+125°C	3.3V	
DC47	1.5	1.9	mA	-40°C 至 +85°C	2.0V	16 MIPS, Fosc = 32 MHz
	—	2	mA	+125°C	2.0V	
	1.6	2.0	mA	-40°C 至 +85°C	3.3V	
	—	2.1	mA	+125°C	3.3V	
DC50	0.54	0.61	mA	-40°C 至 +85°C	2.0V	4 MIPS (FRC), Fosc = 8 MHz
	0.54	0.64	mA	-40°C 至 +85°C	3.3V	
DC51	17	78	μA	-40°C 至 +85°C	2.0V	LPRC (15.5 KIPS), Fosc = 31 kHz
	—	128	μA	+125°C	2.0V	
	18	80	μA	-40°C 至 +85°C	3.3V	
	—	130	μA	+125°C	3.3V	

注 1: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

注 2: 基本 IDLE 电流是在内核关闭、使能时钟以及所有模块关闭的条件下测得的。外设模块禁止 SFR 寄存器为零。所有 I/O 引脚都配置为输入且被拉到 VSS。

# PIC24FJ128GB204 系列

表 33-6: 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)			
			-40°C ≤ Ta ≤ +125°C (扩展级)			
参数编号	典型值 (1)	最大值	单位	工作温度	VDD	条件
<b>掉电电流 (IPD) (5,6)</b>						
DC60	2.9	17	μA	-40°C	2.0V	休眠 (2)
	4.3	17	μA	+25°C		
	8.3	27.5	μA	+60°C		
	20	27.5	μA	+85°C		
	—	79	μA	+125°C		
	2.9	18	μA	-40°C	3.3V	
	4.3	18	μA	+25°C		
	8.4	28	μA	+60°C		
	20.5	28	μA	+85°C		
	—	80	μA	+125°C		
DC61	0.07	—	μA	-40°C	2.0V	低电压休眠 (3)
	0.38	—	μA	+25°C		
	2.6	—	μA	+60°C		
	9.0	—	μA	+125°C		
	0.09	—	μA	-40°C	3.3V	
	0.42	—	μA	+25°C		
	2.75	—	μA	+60°C		
	9.0	—	μA	+125°C		
DC70	0.1	700	nA	-40°C	2.0V	深度休眠
	18	700	nA	+25°C		
	230	1700	nA	+60°C		
	1.8	3.0	μA	+85°C		
	—	24	μA	+125°C		
	5	900	nA	-40°C	3.3V	
	75	900	nA	+25°C		
	540	3450	nA	+60°C		
	1.5	6.0	μA	+85°C		
—	48	μA	+125°C			
DC74	0.4	2.0	μA	-40°C 至 +125°C	0V	带 VBAT 模式的 RTCC (LPRC/SOSC) (4)

- 注 1: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。
- 2: 禁止保持低电压稳压器; RETEN (RCON<12>) = 0,  $\overline{\text{LPCFG}}$  (CW1<10>) = 1。
- 3: 使能保持低电压稳压器; RETEN (RCON<12>) = 1,  $\overline{\text{LPCFG}}$  (CW1<10>) = 0。
- 4: VBAT 引脚连接至电池, RTCC 在 VDD = 0 时继续运行。
- 5: 基本 IPD 是在所有外设和时钟关闭的情况下测得的。所有 I/O 都配置为输入且被拉到 Vss。WDT 等也都被关闭。
- 6: 这些电流是在该系列中具有最大存储器的器件上测得的。

# PIC24FJ128GB204 系列

表 33-7: 直流特性:  $\Delta$  电流 (BOR、WDT、DSBOR 和 DSWDT) (4)

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	典型值 (1)	最大值	单位	工作温度	VDD	条件
欠压复位时的电流增量 ( $\Delta$ BOR) (2)						
DC25	3.1	5.0	μA	-40°C 至 +125°C	2.0V	$\Delta$ BOR(2)
	4.3	6.0	μA	-40°C 至 +125°C	3.3V	
看门狗定时器复位时的电流增量 ( $\Delta$ WDT) (2)						
DC71	0.8	1.5	μA	-40°C 至 +125°C	2.0V	$\Delta$ WDT(2)
	0.8	1.5	μA	-40°C 至 +125°C	3.3V	
高 / 低电压检测时的电流增量 ( $\Delta$ HLVD) (2)						
DC75	4.2	15	μA	-40°C 至 +125°C	2.0V	$\Delta$ HLVD(2)
	4.2	15	μA	-40°C 至 +125°C	3.3V	
实时时钟和日历运行时的电流增量 ( $\Delta$ RTCC) (2)						
DC77	0.3	1.0	μA	-40°C 至 +125°C	2.0V	$\Delta$ RTCC (带 SOSC) (2)
	0.35	1.0	μA	-40°C 至 +125°C	3.3V	
DC77A	0.3	1.0	μA	-40°C 至 +125°C	2.0V	$\Delta$ RTCC (带 LPRC) (2)
	0.35	1.0	μA	-40°C 至 +125°C	3.3V	
深度休眠 BOR 时的电流增量 ( $\Delta$ DSBOR) (2)						
DC81	0.11	0.40	μA	-40°C 至 +125°C	2.0V	$\Delta$ 深度休眠 BOR(2)
	0.12	0.40	μA	-40°C 至 +125°C	3.3V	
深度休眠看门狗定时器复位时的电流增量 ( $\Delta$ DSWDT) (2)						
DC80	0.24	0.40	μA	-40°C 至 +125°C	2.0V	$\Delta$ 深度休眠 WDT(2)
	0.24	0.40	μA	-40°C 至 +125°C	3.3V	
VBAT A/D 监视器 (3)						
DC91	1.5	—	μA	-40°C 至 +125°C	3.3V	V <sub>BAT</sub> = 2V
	4	—	μA	-40°C 至 +125°C	3.3V	V <sub>BAT</sub> = 3.3V

- 注 1: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。
- 2: 模块使能和运行时的电流增量。
- 3: A/D 通道从内部连接至 V<sub>BAT</sub> 引脚; 这是 A/D V<sub>BAT</sub> 工作期间的电流。
- 4:  $\Delta$  电流为模块使能时额外消耗的电流。掉电时外设模块的电流消耗是这一电流与基本 I<sub>PD</sub> 电流之和。

# PIC24FJ128GB204 系列

表 33-8: 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) -40°C ≤ TA ≤ +125°C (扩展级)					
参数编号	符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件
	V <sub>IL</sub>	<b>输入低电压<sup>(3)</sup></b>					
DI10		具有 ST 缓冲器的 I/O 引脚	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
DI11		具有 TTL 缓冲器的 I/O 引脚	V <sub>SS</sub>	—	0.15 V <sub>DD</sub>	V	
DI15		MCLR	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
DI16		OSC1 (XT 模式)	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
DI17		OSC1 (HS 模式)	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
DI18		具有 I <sup>2</sup> C™ 缓冲器的 I/O 引脚	V <sub>SS</sub>	—	0.3 V <sub>DD</sub>	V	
DI19		具有 SMBus 缓冲器的 I/O 引脚	V <sub>SS</sub>	—	0.8	V	使能 SMBus
	V <sub>IH</sub>	<b>输入高电压<sup>(3)</sup></b>					
DI20		具有 ST 缓冲器的 I/O 引脚: 具有模拟功能 仅有数字功能	0.8 V <sub>DD</sub> 0.8 V <sub>DD</sub>	— —	V <sub>DD</sub> 5.5	V V	
DI21		具有 TTL 缓冲器的 I/O 引脚: 具有模拟功能 仅有数字功能	0.25 V <sub>DD</sub> + 0.8 0.25 V <sub>DD</sub> + 0.8	— —	V <sub>DD</sub> 5.5	V V	
DI25		MCLR	0.8 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
DI26		OSC1 (XT 模式)	0.7 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
DI27		OSC1 (HS 模式)	0.7 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
DI28		具有 I <sup>2</sup> C 缓冲器的 I/O 引脚: 具有模拟功能 仅有数字功能	0.7 V <sub>DD</sub> 0.7 V <sub>DD</sub>	— —	V <sub>DD</sub> 5.5	V V	
DI29		具有 SMBus 缓冲器的 I/O 引脚: 具有模拟功能 仅有数字功能	2.1 2.1	— —	V <sub>DD</sub> 5.5	V V	2.5V ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub>
DI30	ICNPU	<b>CNxx 上拉电流</b>	150	340	550	μA	V <sub>DD</sub> = 3.3V, V <sub>PIN</sub> = V <sub>SS</sub>
DI30A	ICNPD	<b>CNxx 下拉电流</b>	150	310	550	μA	V <sub>DD</sub> = 3.3V, V <sub>PIN</sub> = V <sub>DD</sub>
	I <sub>IL</sub>	<b>输入泄漏电流<sup>(2)</sup></b>					
DI50		I/O 端口	—	—	±1	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , 引脚处于高阻态
DI51		模拟输入引脚	—	—	±1	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , 引脚处于高阻态
DI55		MCLR	—	—	±1	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub>
DI56		OSCI/CLKI	—	—	±1	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , EC、XT 和 HS 模式

注 1: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

注 2: 负电流定义为从引脚流出的电流。

注 3: I/O 引脚缓冲器类型请参见表 1-3。

# PIC24FJ128GB204 系列

表 33-9: 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)					
			-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	
DO10	VOL	输出低电压	—	—	0.4	V	IOL = 6.6 mA, VDD = 3.6V	
		I/O 端口	—	—	0.4	V		IOL = 5.0 mA, VDD = 2V
DO16		OSCO/CLKO	—	—	0.4	V	IOL = 6.6 mA, VDD = 3.6V	
			—	—	0.4	V	IOL = 5.0 mA, VDD = 2V	
DO20	VOH	输出高电压	3.0	—	—	V	IOH = -3.0 mA, VDD = 3.6V	
		I/O 端口	2.4	—	—	V		IOH = -6.0 mA, VDD = 3.6V
			1.65	—	—	V		IOH = -1.0 mA, VDD = 2V
DO26		OSCO/CLKO	1.4	—	—	V	IOH = -3.0 mA, VDD = 2V	
			2.4	—	—	V	IOH = -6.0 mA, VDD = 3.6V	
			1.4	—	—	V	IOH = -1.0 mA, VDD = 2V	

注 1: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

表 33-10: 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	闪存程序存储器	20000	—	—	E/W	-40°C 至 +125°C V <sub>MIN</sub> = 最小工作电压 V <sub>MIN</sub> = 最小工作电压 假定未违反其他规范
D131	VPR	单元耐擦写能力	V <sub>MIN</sub>	—	3.6	V	
D132B		读操作使用的 VDD	V <sub>MIN</sub>	—	3.6	V	
D133A	TIW	自定时写操作使用的 VDD	—	20	—	μs	
		自定时字写周期时间	—	1.5	—	ms	
D133B	TIE	自定时行写周期时间	20	—	40	ms	
D134	TRETD	自定时页擦除时间	20	—	—	年	
D135	IDDP	特性保持时间	—	5	—	mA	
D136	VOTP	编程时的供电电流	3.1	—	3.6	V	
D137	TOTP	OTP 编程	—	500	—	μs	
		OTP 存储器写位	—	500	—	μs	

注 1: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 的条件下给出的。

# PIC24FJ128GB204 系列

表 33-11: 内部稳压器规范

工作条件: 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (扩展级)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
DVR	TVREG	稳压器启动时间	—	10	—	$\mu\text{s}$	VREGS = 1 以及任何 POR 或 BOR
DVR10	VBG	内部带隙参考电压	—	1.2	—	V	
DVR11	TBG	带隙参考启动时间	—	1	—	ms	
DVR20	VRGOUT	稳压器输出电压	—	1.8	—	V	VDD > 1.9V
DVR21	CEFC	外部滤波器电容值	4.7	10	—	$\mu\text{F}$	建议串联一个阻抗 < 3 $\Omega$ 的电阻; 要求串连电阻的阻抗 < 5 $\Omega$
DVR30	VLVR	低电压稳压器输出电压	—	1.2	—	V	RETEN = 1, $\overline{\text{LPCFG}} = 0$

表 33-12: 高 / 低压检测特性

工作条件: 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (扩展级)								
参数编号	符号	特性	最小值	典型值	最大值	单位	条件	
DC18	VHLVD	VDD 跳变时的 HLVD 电压	HLVDL<3:0> = 0100 <sup>(1)</sup>	3.45	3.59	3.74	V	
			HLVDL<3:0> = 0101	3.33	3.45	3.58	V	
			HLVDL<3:0> = 0110	3.0	3.125	3.25	V	
			HLVDL<3:0> = 0111	2.8	2.92	3.04	V	
			HLVDL<3:0> = 1000	2.7	2.81	2.93	V	
			HLVDL<3:0> = 1001	2.50	2.6	2.70	V	
			HLVDL<3:0> = 1010	2.4	2.52	2.64	V	
			HLVDL<3:0> = 1011	2.30	2.4	2.50	V	
			HLVDL<3:0> = 1100	2.20	2.29	2.39	V	
			HLVDL<3:0> = 1101	2.1	2.19	2.28	V	
		HLVDL<3:0> = 1110	2.0	2.08	2.17	V		
DC101	VTHL	HLVDIN 引脚跳变时的 HLVD 电压	HLVDL<3:0> = 1111	—	1.2	—	V	

注 1: HLVD<3:0> 从 0000 至 0011 的值的跳变点未实现。

**表 33-13: 比较器直流规范**

工作条件:		2.0V < VDD < 3.6V					
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	最大值	单位	备注
D300	VIOFF	输入失调电压	—	20	±40	mV	(注 1)
D301	VICM	输入共模电压	0	—	VDD	V	(注 1)
D302	CMRR	共模抑制比	55	—	—	dB	(注 1)
D306	IQCOMP	每个比较器的 AVDD 静态电流	—	27	—	μs	使能比较器
D307	TRESP	响应时间	—	300	—	ns	(注 2)
D308	TMC2OV	比较器模式更改到输出有效的时间	—	—	10	μs	

注 1: 这些参数为特征值, 未经测试。

注 2: 响应时间是在比较器的一个输入端电压为 VDD/2 而另一个输入端从 VSS 变化到 VDD、阶跃电压为 40 mV 和过驱动电压为 15 mV 的条件下测得的。

**表 33-14: 比较器参考电压直流规范**

工作条件:		2.0V < VDD < 3.6V					
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	最大值	单位	备注
VR310	TSET	稳定时间	—	—	10	μs	(注 1)
VRD311	CVRAA	绝对精度	-100	—	100	mV	
VRD312	CVRUR	单位电阻值 (R)	—	4.5	—	kΩ	

注 1: 测量 CVR<4:0> 从 11111 变化到 00000 的时间间隔。

# PIC24FJ128GB204 系列

表 33-15: VBAT 工作电压规范

参数编号	符号	特性	最小值	典型值	最大值	单位	备注
DVB01	VBT	工作电压	1.6	—	3.6	V	电池连接至 VBAT 引脚
DVB10	VBTADC	VBAT A/D 监视电压规范 (1)	1.6	—	3.6	V	A/D 使用内部 A/D 通道监视 VBAT 引脚

注 1: 对于 12 位 A/D, 使用该模块测得的 A/D 值可通过下列公式表示: 测量电压 = ((VBAT/2)/VDD) \* 4096。

表 33-16: CTMU 电流源规范

直流特性		标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°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 (3)	单位	备注	条件
DCT10	IOUT1	CTMU 电流源, 基本范围	208	550	797	nA	CTMUICON<9:8> = 00	2.5V < VDD < VDDMAX
DCT11	IOUT2	CTMU 电流源, 10x 范围	3.32	5.5	7.67	μA	CTMUICON<9:8> = 01	
DCT12	IOUT3	CTMU 电流源, 100x 范围	32.22	55	77.78	μA	CTMUICON<9:8> = 10	
DCT13	IOUT4	CTMU 电流源, 1000x 范围	322	550	777	μA	CTMUICON<9:8> = 11 (2)	
DCT21	VΔ	每摄氏度温度二极管电压变化	—	-3	—	mV/°C		

- 注 1: 电流微调范围的中点为标称值 (CTMUICON<15:10> = 000000)。  
 2: 不要对温度传感二极管使用该电流范围。  
 3: 最大值在 +85°C 条件下测得。

表 33-17: USB ON-THE-GO 模块规范

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	最大值	单位	备注
工作电压							
DUS01	VUSB3V3	USB 电源电压	取如下二者中的较大值: 3.0 或 (VDD - 0.3V)	3.3	3.6	V	使能 USB 模块
			(VDD - 0.3V) (1)	—	3.6	V	禁止 USB, D+/D- 未使用并外部拉低或保留为高阻态
			(VDD - 0.3V)	VDD	3.6	V	禁止 USB, D+/D- 用作通用 I/O

注 1: VUSB3V3 引脚在这些情况下也会保留为高阻态。但是, 如果电压悬空且低于 (VDD - 0.3V), 这可能导致 IPD 电流比规定值高。

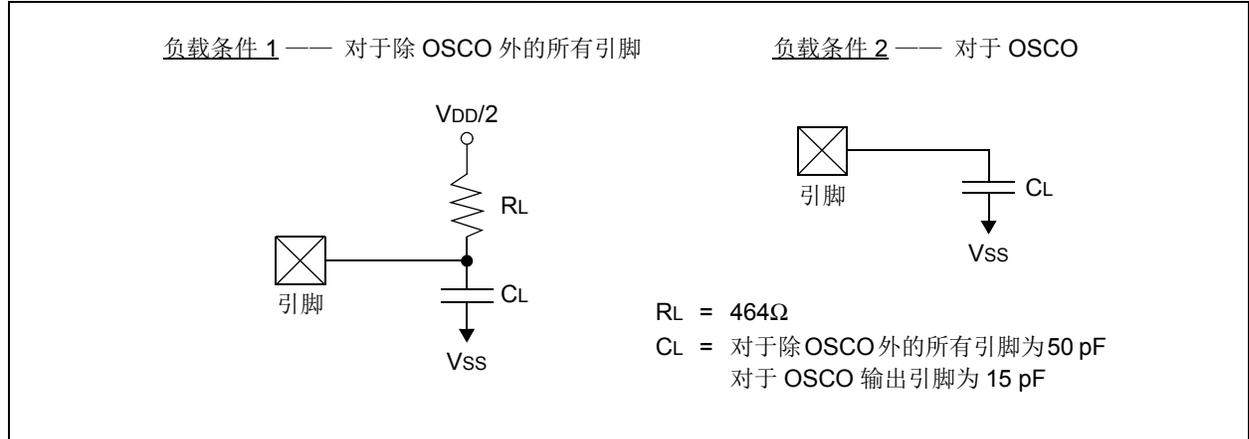
## 33.2 交流特性和时序参数

本节包含的信息说明了 PIC24FJ128GB204 系列器件的交流特性和时序参数。

**表 33-18: 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 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)
	$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (扩展级)
	工作电压 $V_{DD}$ 范围如第 33.1 节“直流特性”所示。

**图 33-2: 器件时序规范的负载条件**



**表 33-19: 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 <sup>(1)</sup>	最大值	单位	条件
DO50	Cosco	OSCO/CLKO 引脚	—	—	15	pF	当外部时钟用于驱动 OSCI 时, 处于 XT 和 HS 模式
DO56	Cio	所有 I/O 引脚和 OSCO	—	—	50	pF	EC 模式
DO58	CB	SCLx 和 SDAx	—	—	400	pF	在 I <sup>2</sup> C™ 模式下

**注 1:** 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

# PIC24FJ128GB204 系列

图 33-3: 外部时钟时序

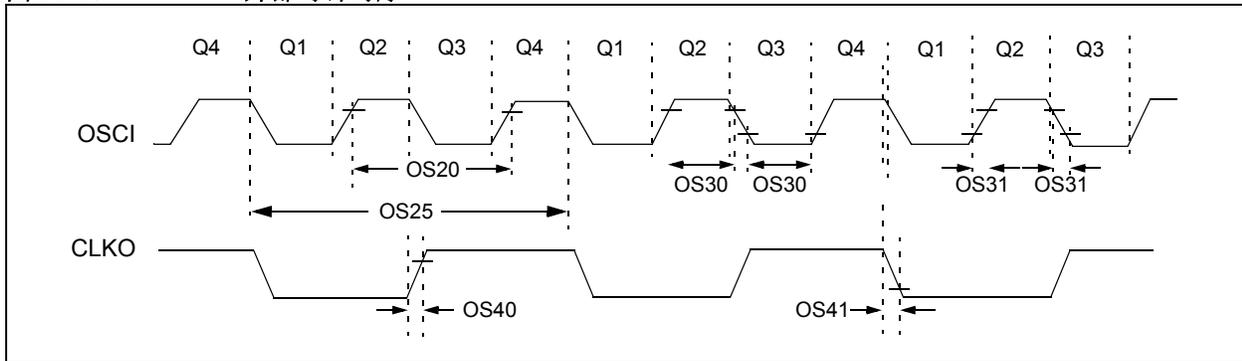


表 33-20: 外部时钟时序要求

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° (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 <sup>(1)</sup>	最大值	单位	条件
OS10	Fosc	外部 CLKI 频率 (仅在 EC 模式下允许 使用外部时钟)	DC 4	— —	32 48	MHz MHz	EC ECPLL (注 2)
		振荡器频率	3.5	—	10	MHz	XT
			4	—	8	MHz	XTPLL
			10	—	32	MHz	HS
			12 31	— —	32 33	MHz kHz	HSPLL SOSC
OS20	Tosc	Tosc = 1/Fosc	—	—	—	Fosc 的值请参见参数 OS10	
OS25	Tcy	指令周期 <sup>(3)</sup>	62.5	—	DC	ns	
OS30	TosL, TosH	外部时钟输入 (OSCI) 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) 上升或下降时间	—	—	20
OS40	TckR	CLKO 上升时间 <sup>(4)</sup>	—	6	10	ns	
OS41	TckF	CLKO 下降时间 <sup>(4)</sup>	—	6	10	ns	

- 注 1: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。
- 2: 表示系统时钟预分频器的输入。必须仍然配置了 PLL 分频器和后分频器, 从而系统时钟频率不会超出图 33-1 所示的最大频率。
- 3: 指令周期 (Tcy) 等于输入振荡器时基周期的 2 倍。所有规定值均基于标准工作条件下, 器件执行代码时对应特定振荡器类型的特征数据。超过规范值可导致振荡器运行不稳定和 / 或使电流消耗超过预期值。所有器件在测试“最小值”时, 均在 OSCI/CLKI 引脚接入了外部时钟。当使用外部时钟输入时, 所有器件的“最大”周期时间限制为“DC” (无时钟)。
- 4: 测量在 EC 模式下进行。CLKO 信号是在 OSCO 引脚上测得的。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。

# PIC24FJ128GB204 系列

表 33-21: PLL 时钟时序规范 (VDD = 1.8V 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) -40°C ≤ TA ≤ +125°C (扩展级)					
参数编号	符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件
OS50	FPLLI	USB PLL 输入频率范围	2	4	4	MHz	ECPLL 模式
			2	4	4	MHz	HSPLL 模式
			2	4	4	MHz	XTPLL 模式
OS52	TLOCK	USB PLL 启动时间 (锁定时间)	—	—	128	μs	
OS53	DCLK	CLKO 稳定性 (去抖动性能)	-0.25	—	0.25	%	
OS54	F4xPLL	4x PLL 输入频率范围	2	—	8	MHz	4xPLL
OS55	F6xPLL	6x PLL 输入频率范围	2	—	5	MHz	6xPLL
OS56	F8xPLL	8x PLL 输入频率范围	2	—	4	MHz	8xPLL
OS57	TxPLL-LOCK	PLL 启动时间 (锁定时间)	—	—	24	μs	
OS58	DxPLL-CLK	PLL CLKO 稳定性 (去抖动性能)	-2	—	2	%	

注 1: 这些参数为特征值, 但未经生产测试。

表 33-22: 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) -40°C ≤ TA ≤ +125°C (扩展级)					
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	±0.15	1	%	2.0V ≤ VDD ≤ 3.6V, 0°C ≤ TA ≤ +85°C (注 1)	
		1.5	—	1.5	%	2.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ 0°C	
		-0.20	±0.05	-0.20	%	2.0V ≤ VDD ≤ 3.6V, -40°C ≤ TA ≤ +85°C, 自调节使能并锁定 (注 2)	
		—	3	5	%	2.0V ≤ VDD ≤ 3.6V, TA = +125°C	
F21	LPRC @ 31 kHz	-20	—	20	%	VCAP 输出电压 = 1.8V	
F22	OSCTUN 步长	—	0.05	—	%/位		
F23	FRC 自调节锁定时间	—	<5	8	ms	(注 3)	

注 1: 要达到这一精度, 单片机封装所受到的物理应力 (例如: 弯曲 PCB) 必须保持最小。

2: 精度使用参考源测量。

3: 从参考时钟稳定, 且在正常范围内, 到在 F20 指定的范围内进行 FRC 调节 (自调节) 的时间。

# PIC24FJ128GB204 系列

表 33-23: RC 振荡器起振时间

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)				
			-40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
FR0	TfRC	FRC 振荡器起振时间	—	15	—	μs	
FR1	TLPRC	低功耗 RC 振荡器起振时间	—	50	—	μs	

图 33-4: CLKO 和 I/O 时序特性

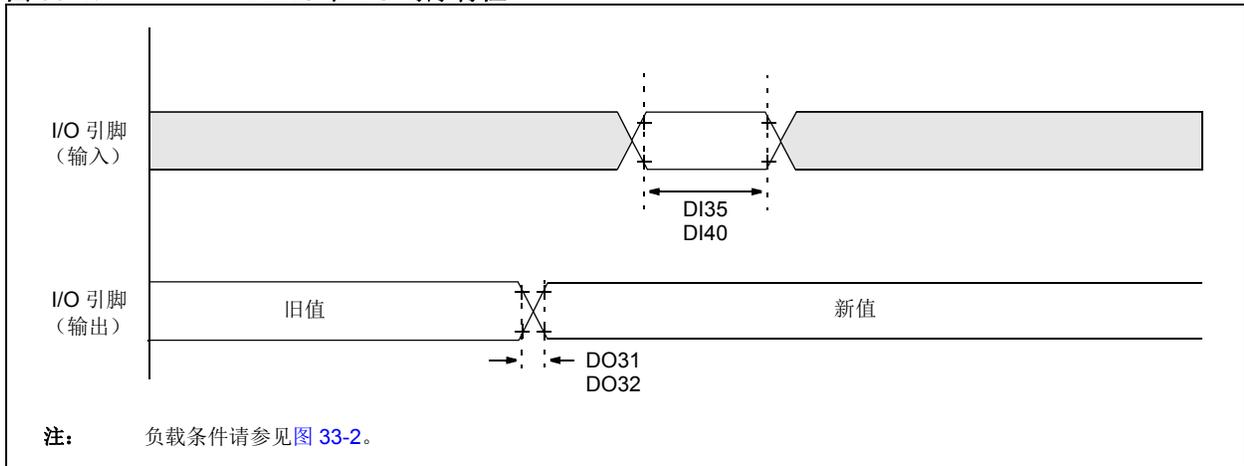


表 33-24: CLKO 和 I/O 时序要求

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)				
			-40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件
DO31	TioR	端口输出上升时间	—	10	25	ns	
DO32	TioF	端口输出下降时间	—	10	25	ns	
DI35	TINP	INTx 引脚高电平或低电平时间 (输入)	20	—	—	ns	
DI40	TRBP	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”栏中的数据都是在 3.3V、+25°C 的条件下给出的。

# PIC24FJ128GB204 系列

表 33-25: 复位和欠压复位要求

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)				
			-40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
SY10	TMCL	MCLR 脉冲宽度 (低电平)	2	—	—	μs	
SY12	TPOR	上电复位延时	—	2	—	μs	
SY13	TIOZ	自 MCLR 低电平或看门狗定时器复位起 I/O 处于高阻态的时间	取 (3 Tcy + 2) 或 700 中的 较小者		(3 Tcy + 2)	μs	
SY25	TBOR	欠压复位脉冲宽度	1	—	—	μs	VDD ≤ VBOR
SY45	TRST	内部状态复位时间	—	50	—	μs	
SY70	TDSWU	深度休眠唤醒时间	—	200	—	μs	VCAP 在唤醒前完全放电
SY71	TPM	程序存储器唤醒时间	—	20	—	μs	从休眠模式唤醒, VREGS = 0
			—	1	—	μs	从休眠模式唤醒, VREGS = 1
SY72	TLVR	低电压稳压器唤醒时间	—	90	—	μs	从休眠模式唤醒, VREGS = 0
			—	70	—	μs	从休眠模式唤醒, VREGS = 1

# PIC24FJ128GB204 系列

图 33-5: TIMER1、2、3、4 和 5 外部时钟时序特性

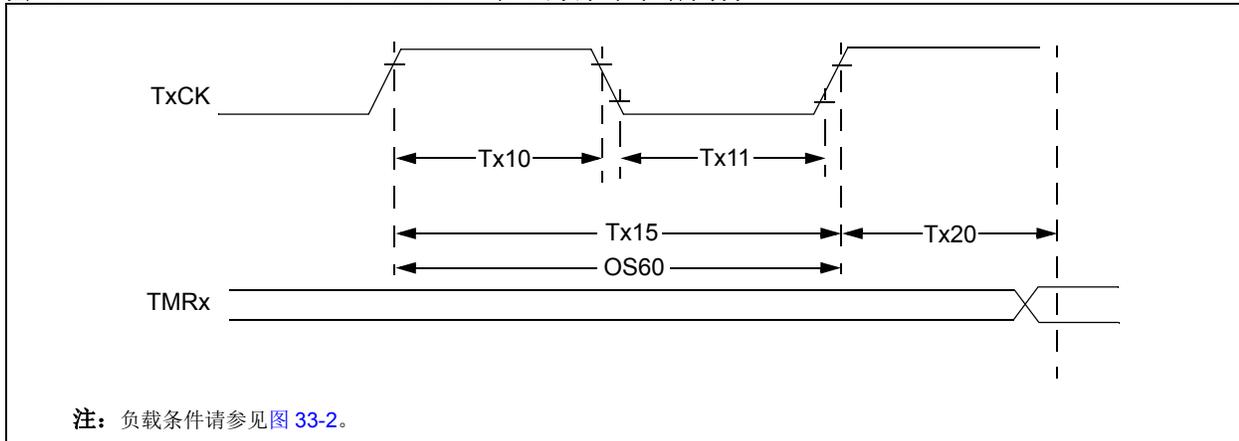


表 33-26: TIMER1 外部时钟时序要求<sup>(1)</sup>

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) -40°C ≤ TA ≤ +125°C (扩展级)						
参数编号	符号	特性		最小值	典型值	最大值	单位	条件
TA10	TtXH	T1CK 高电平时间	同步, 无预分频器	0.5 Tcy + 20	—	—	ns	还必须满足参数 TA15
			同步, 带预分频器	10	—	—	ns	
			异步	10	—	—	ns	
TA11	TtXL	T1CK 低电平时间	同步, 无预分频器	0.5 Tcy + 20	—	—	ns	还必须满足参数 TA15
			同步, 带预分频器	10	—	—	ns	
			异步	10	—	—	ns	
TA15	TtXP	T1CK 输入周期	同步, 无预分频器	Tcy + 40	—	—	ns	N = 预分频值 (1、8、64 或 256)
			同步, 带预分频器	取 20 ns 或 (Tcy + 40)/N 的较大值	—	—	—	
			异步	20	—	—	ns	
OS60	Ft1	SOSC1/T1CK 振荡器输入频率范围 (通过将 TCS (T1CON<1>) 置 1 使能振荡器)		DC	—	50	kHz	
TA20	TckEXTMRL	外部 T1CK 时钟边沿到定时器递增的延时		0.5 Tcy	—	1.5 Tcy	—	

注 1: Timer1 为 A 型。

# PIC24FJ128GB204 系列

表 33-27: TIMER2 和 TIMER4 外部时钟时序要求

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) -40°C ≤ TA ≤ +125°C (扩展级)						
参数编号	符号	特性		最小值	典型值	最大值	单位	条件
TB10	TtxH	TxCK 高电平时间	同步, 无预分频器	0.5 Tcy + 20	—	—	ns	还必须满足参数 TB15
			同步, 带预分频器	10	—	—	ns	
TB11	TtxL	TxCK 低电平时间	同步, 无预分频器	0.5 Tcy + 20	—	—	ns	还必须满足参数 TB15
			同步, 带预分频器	10	—	—	ns	
TB15	TtxP	TxCK 输入周期	同步, 无预分频器	Tcy + 40	—	—	ns	N = 预分频值 (1、8、64 或 256)
			同步, 带预分频器	取 20 ns 和 (Tcy + 40)/N 中的较大值				
TB20	Tckextmrl	外部 TxCK 时钟边沿到定时器递增的延时		0.5 Tcy	—	1.5 Tcy	—	

表 33-28: TIMER3 和 TIMER5 外部时钟时序要求

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) -40°C ≤ TA ≤ +125°C (扩展级)						
参数编号	符号	特性		最小值	典型值	最大值	单位	条件
TC10	TtxH	TxCK 高电平时间	同步	0.5 Tcy + 20	—	—	ns	还必须满足参数 TC15
TC11	TtxL	TxCK 低电平时间	同步	0.5 Tcy + 20	—	—	ns	还必须满足参数 TC15
TC15	TtxP	TxCK 输入周期	同步, 无预分频器	Tcy + 40	—	—	ns	N = 预分频值 (1、8、64 或 256)
			同步, 带预分频器	取 20 ns 和 (Tcy + 40)/N 中的较大值				
TC20	Tckextmrl	外部 TxCK 时钟边沿到定时器递增的延时		0.5 Tcy	—	1.5 Tcy	—	

# PIC24FJ128GB204 系列

图 33-6: 输入捕捉 x (ICx) 时序特性

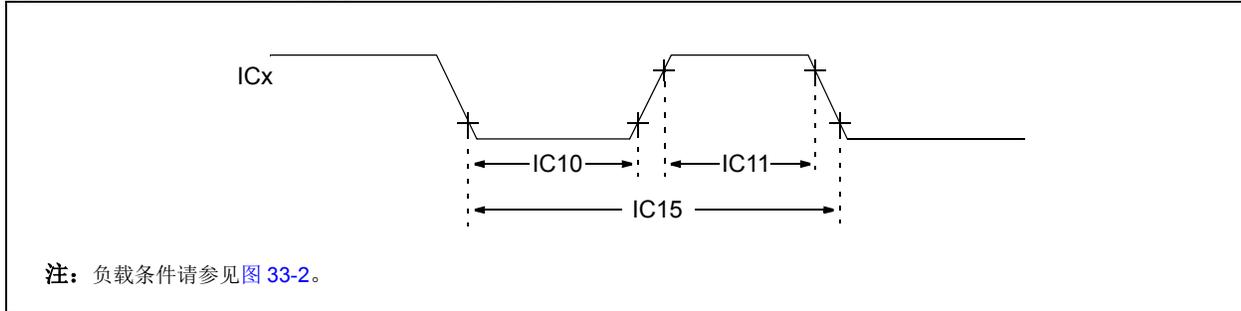


表 33-29: 输入捕捉 x 时序要求

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 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (扩展级)				
参数编号	符号	特性 <sup>(1)</sup>	最小值	最大值	单位	条件
IC10	TccL	ICx 输入低电平时间	无预分频器	$0.5 T_{CY} + 20$	—	ns
		有预分频器	10	—	ns	
IC11	TccH	ICx 输入高电平时间	无预分频器	$0.5 T_{CY} + 20$	—	ns
		有预分频器	10	—	ns	
IC15	TccP	ICx 输入周期	$(T_{CY} + 40)/N$	—	ns	N = 预分频值 (1、4 和 16)

注 1: 这些参数仅为特征值, 在生产时未经测试。

图 33-7: 输出比较 x 模块 (OCx) 时序特性

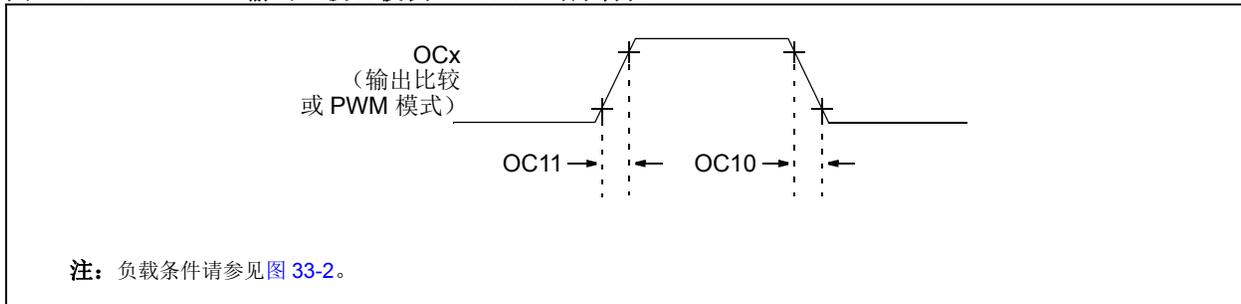


表 33-30: 输出比较 x 模块时序要求

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 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (扩展级)					
参数编号	符号	特性 <sup>(1)</sup>	最小值	典型值	最大值	单位	条件
OC10	TccF	OCx 输出下降时间	—	—	—	ns	见参数 DO32
OC11	TccR	OCx 输出上升时间	—	—	—	ns	见参数 DO31

注 1: 这些参数仅为特征值, 在生产时未经测试。

图 33-8: OCx/PWM 模块时序特性

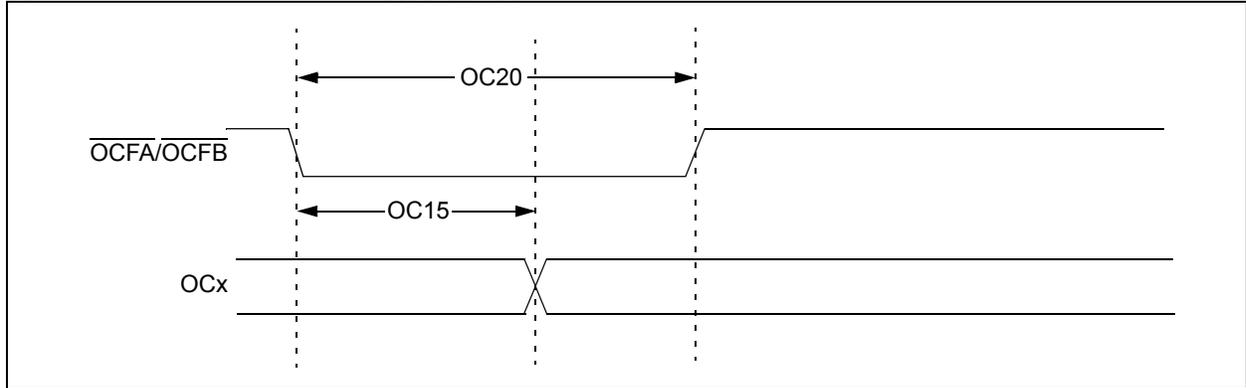


表 33-31: 简单 OCx/PWM 模式时序要求

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) -40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性 <sup>(1)</sup>	最小值	典型值	最大值	单位	条件
OC15	TFD	故障输入到 PWM I/O 改变的时间	—	—	50	ns	
OC20	TFLT	故障输入脉冲宽度	50	—	—	ns	

注 1: 这些参数仅为特征值, 在生产时未经测试。

# PIC24FJ128GB204 系列

图 33-9: SPIx 模块主模式 (CKE = 0) 时序特性

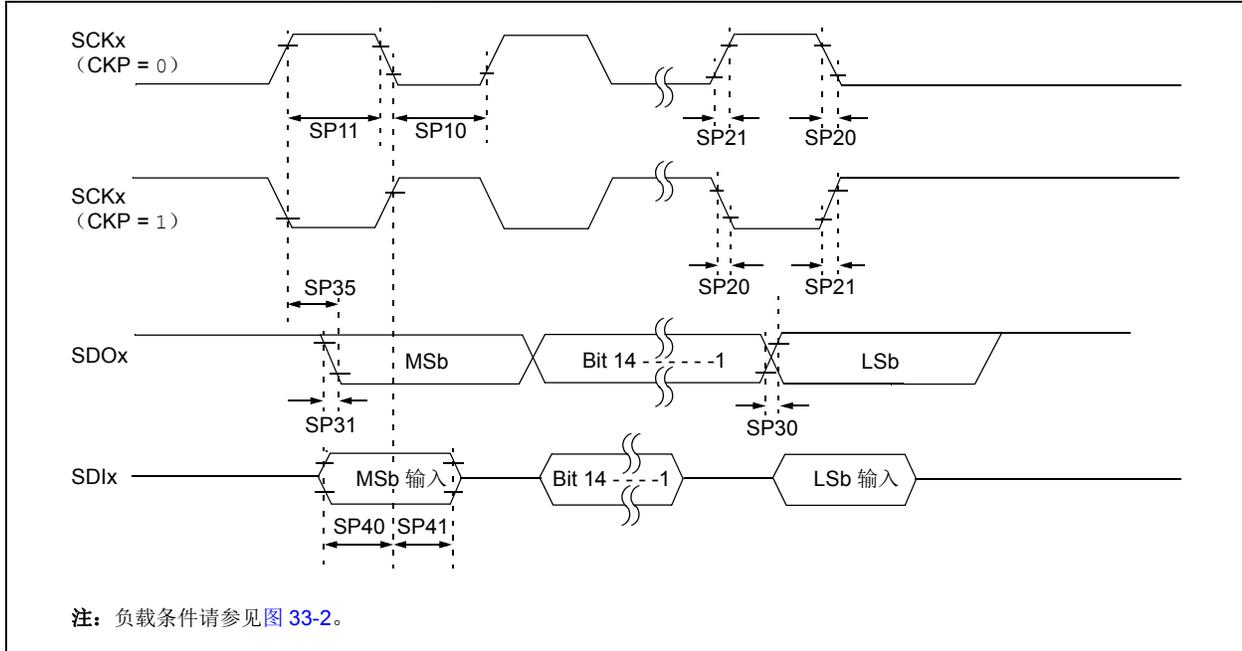


表 33-32: SPIx 主模式 (CKE = 0) 时序要求

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) -40°C ≤ TA ≤ +125°C (扩展级)					
参数编号	符号	特性 (1)	最小值	典型值 (2)	最大值	单位	条件
SP10	TscL	SCKx 输出低电平时间	Tcy/2	—	—	ns	(注 3)
SP11	TscH	SCKx 输出高电平时间	Tcy/2	—	—	ns	(注 3)
SP20	TscF	SCKx 输出下降时间	—	—	—	ns	见参数 DO32 (注 4)
SP21	TscR	SCKx 输出上升时间	—	—	—	ns	见参数 DO31 (注 4)
SP30	TdoF	SDOx 数据输出下降时间	—	—	—	ns	见参数 DO32 (注 4)
SP31	TdoR	SDOx 数据输出上升时间	—	—	—	ns	见参数 DO31 (注 4)
SP35	Tsch2doV, TscL2doV	SCKx 边沿后 SDOx 数据输出有效的 时间	—	6	20	ns	
SP40	TdiV2scH, TdiV2scL	SDIx 数据输入到 SCKx 边沿的 建立时间	23	—	—	ns	
SP41	Tsch2diL, TscL2diL	SDIx 数据输入到 SCKx 边沿的 保持时间	30	—	—	ns	

- 注 1: 这些参数仅为特征值, 在生产时未经测试。  
 注 2: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 条件下测得的。  
 注 3: SCKx 的最小时钟周期为 100 ns。因此, 主模式下产生的时钟必定符合该规范。  
 注 4: 假设所有 SPIx 引脚上的负载均为 50 pF。

图 33-10: SPIx 模块主模式 (CKE = 1) 时序特性

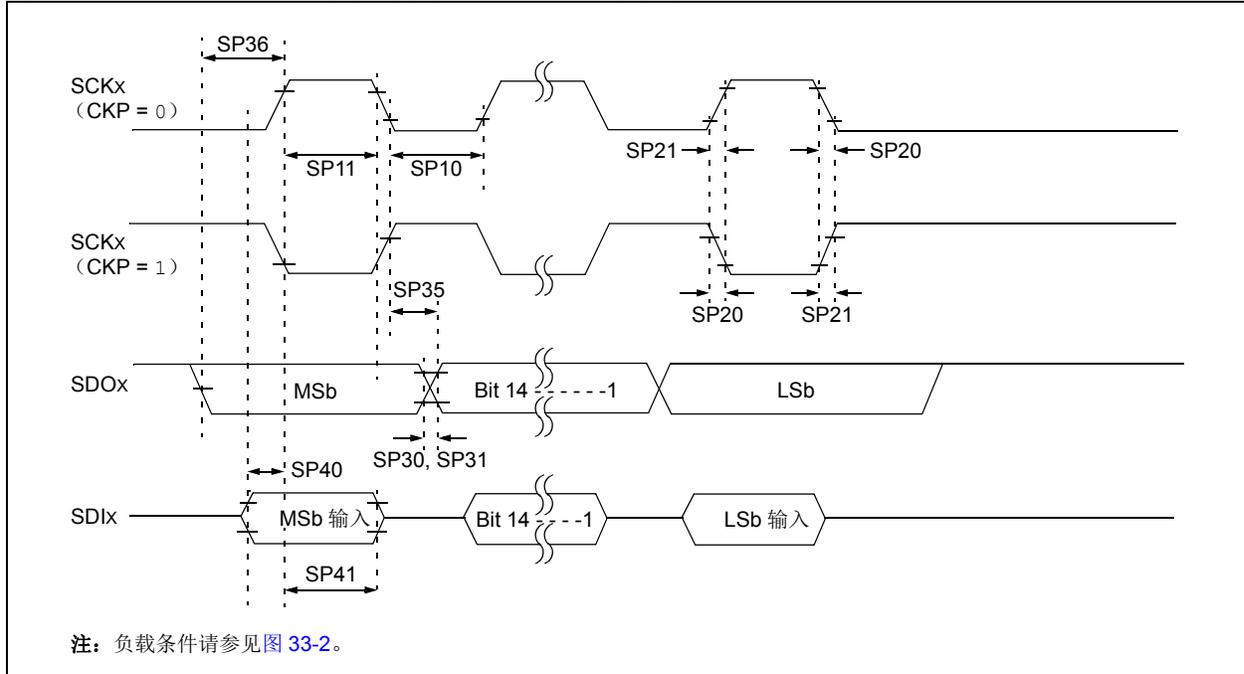


表 33-33: SPIx 模块主模式 (CKE = 1) 时序要求

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) -40°C ≤ Ta ≤ +125°C (扩展级)				
参数编号	符号	特性 (1)	最小值	典型值 (2)	最大值	单位	条件
SP10	TscL	SCKx 输出低电平时间 (3)	Tcy/2	—	—	ns	
SP11	Tsch	SCKx 输出高电平时间 (3)	Tcy/2	—	—	ns	
SP20	TscF	SCKx 输出下降时间 (4)	—	—	—	ns	见参数 DO32
SP21	TscR	SCKx 输出上升时间 (4)	—	—	—	ns	见参数 DO31
SP30	TdoF	SDOx 数据输出下降时间 (4)	—	—	—	ns	见参数 DO32
SP31	TdoR	SDOx 数据输出上升时间 (4)	—	—	—	ns	见参数 DO31
SP35	Tsch2doV, TscL2doV	SCKx 边沿后 SDOx 数据输出有效的时间	—	6	20	ns	
SP36	TdoV2sc, TdoV2scL	SDOx 数据输出建立到 SCKx 第一个边沿的时间	30	—	—	ns	
SP40	TdiV2scH, TdiV2scL	SDIx 数据输入到 SCKx 边沿的建立时间	23	—	—	ns	
SP41	Tsch2diL, TscL2diL	SDIx 数据输入到 SCKx 边沿的保持时间	30	—	—	ns	

注 1: 这些参数仅为特征值, 在生产时未经测试。

2: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 条件下测得的。

3: SCKx 的最小时钟周期为 100 ns。因此, 主模式下产生的时钟必定符合该规范。

4: 假设所有 SPIx 引脚上的负载均为 50 pF。

# PIC24FJ128GB204 系列

图 33-11: SPIx 模块从模式 (CKE = 0) 时序特性

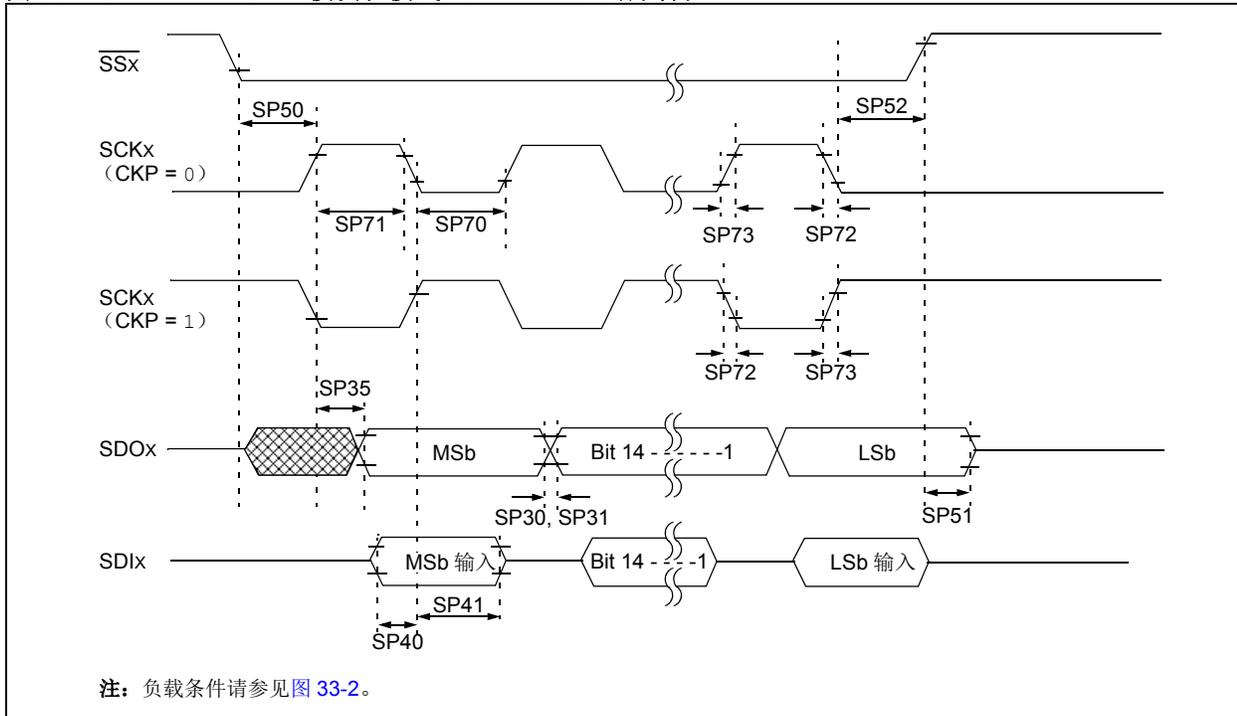
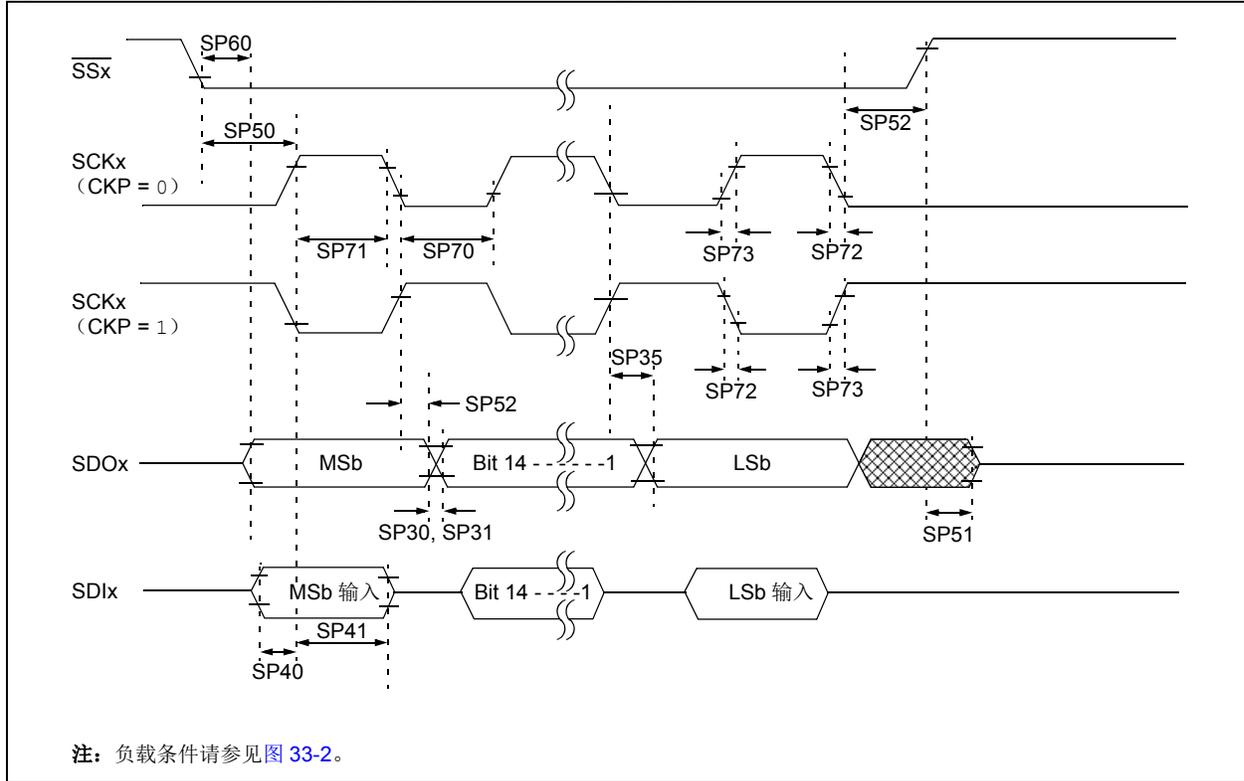


表 33-34: SPIx 模块从模式 (CKE = 0) 时序要求

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) -40°C ≤ TA ≤ +125°C (扩展级)					
参数编号	符号	特性 <sup>(1)</sup>	最小值	典型值 <sup>(2)</sup>	最大值	单位	条件
SP70	TscL	SCKx 输入低电平时间	30	—	—	ns	
SP71	Tsch	SCKx 输入高电平时间	30	—	—	ns	
SP72	TscF	SCKx 输入下降时间 <sup>(3)</sup>	—	10	25	ns	
SP73	TscR	SCKx 输入上升时间 <sup>(3)</sup>	—	10	25	ns	
SP30	TdoF	SDOx 数据输出下降时间 <sup>(3)</sup>	—	—	—	ns	见参数 DO32
SP31	TdoR	SDOx 数据输出上升时间 <sup>(3)</sup>	—	—	—	ns	见参数 DO31
SP35	Tsch2doV, TscL2doV	SCKx 边沿后 SDOx 数据输出有效的时间	—	—	30	ns	
SP40	TdiV2scH, TdiV2scL	SDIx 数据输入到 SCKx 边沿的建立时间	20	—	—	ns	
SP41	Tsch2diL, TscL2diL	SDIx 数据输入到 SCKx 边沿的保持时间	20	—	—	ns	
SP50	TssL2scH, TssL2scL	SSx ↓ 到 SCKx ↑ 或 SCKx 输入的时间	120	—	—	ns	
SP51	TssH2doZ	SSx ↑ 到 SDOx 输出呈高阻抗的时间 <sup>(3)</sup>	10	—	50	ns	
SP52	Tsch2ssH, TscL2ssH	SCKx 边沿后 SSx 有效的时间	1.5 Tcy + 40	—	—	ns	

- 注 1: 这些参数仅为特征值, 在生产时未经测试。  
 2: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 条件下测得的。  
 3: 假设所有 SPIx 引脚上的负载均为 50 pF。

图 33-12: SPIx 模块从模式 (CKE = 1) 时序特性



# PIC24FJ128GB204 系列

表 33-35: SPIx 模块从模式 (CKE = 1) 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) -40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性 (1)	最小值	典型值 (2)	最大值	单位	条件
SP70	TscL	SCKx 输入低电平时间	30	—	—	ns	
SP71	TscH	SCKx 输入高电平时间	30	—	—	ns	
SP72	TscF	SCKx 输入下降时间 (3)	—	10	25	ns	
SP73	TscR	SCKx 输入上升时间 (3)	—	10	25	ns	
SP30	TdoF	SDOx 数据输出下降时间 (3)	—	—	—	ns	见参数 DO32
SP31	TdoR	SDOx 数据输出上升时间 (3)	—	—	—	ns	见参数 DO31
SP35	TscH2doV, TscL2doV	SCKx 边沿后 SDOx 数据输出有效的时间	—	—	30	ns	
SP40	TdiV2scH, TdiV2scL	SDIx 数据输入到 SCKx 边沿的建立时间	20	—	—	ns	
SP41	Tsch2diL, TscL2diL	SDIx 数据输入到 SCKx 边沿的保持时间	20	—	—	ns	
SP50	TssL2scH, TssL2scL	SSx ↓ 到 SCKx ↓ 或 SCKx ↑ 输入的时间	120	—	—	ns	
SP51	TssH2doZ	SSx ↑ 到 SDOx 输出呈现高阻抗的时间 (4)	10	—	50	ns	
SP52	Tsch2ssH, TscL2ssH	SCKx 边沿后 SSx ↑ 的时间	1.5 TCY + 40	—	—	ns	
SP60	TssL2doV	SSx 边沿后 SDOx 数据输出有效的时间	—	—	50	ns	

- 注 1: 这些参数仅为特征值, 在生产时未经测试。  
 2: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、+25°C 条件下测得的。  
 3: SCKx 的最小时钟周期为 100 ns。因此, 主模式下产生的时钟必定符合该规范。  
 4: 假设所有 SPIx 引脚上的负载均为 50 pF。

图 33-13: I<sup>2</sup>C™ 总线启动 / 停止位时序特性 (主模式)

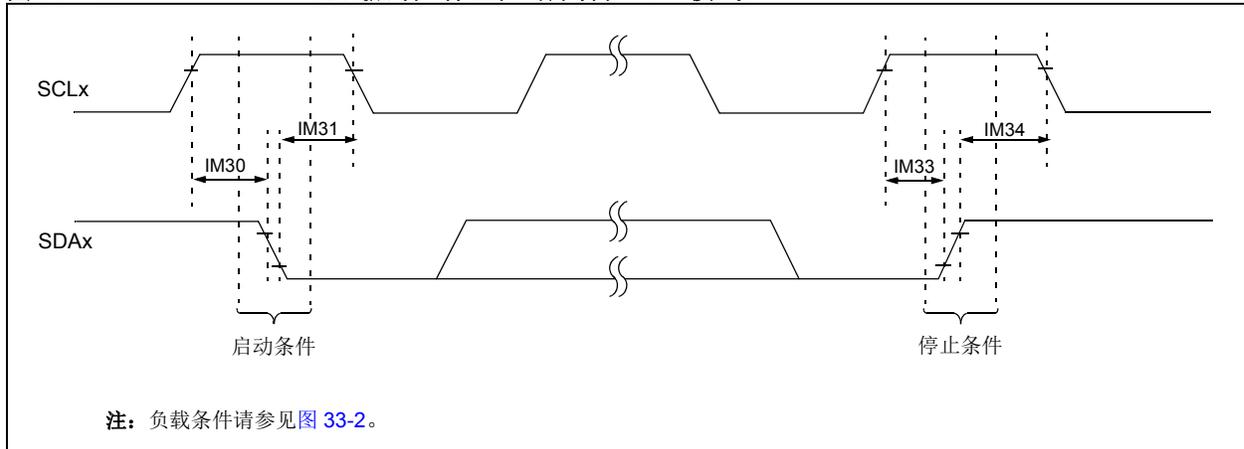


表 33-36: I<sup>2</sup>C™ 总线启动 / 停止位时序要求 (主模式)

交流特性				标准工作条件: 2.0V 至 3.6V (除非另外声明)			条件
				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 <sup>(1)</sup>	最大值	单位		
IM30	TSU:STA	启动条件建立时间	100 kHz 模式	T <sub>CY</sub> (BRG + 1)	—	μs	仅与重复启动条件相关
			400 kHz 模式	T <sub>CY</sub> (BRG + 1)	—	μs	
			1 MHz 模式 <sup>(2)</sup>	T <sub>CY</sub> (BRG + 1)	—	μs	
IM31	THD:STA	启动条件保持时间	100 kHz 模式	T <sub>CY</sub> (BRG + 1)	—	μs	该周期后, 产生第一个时钟脉冲
			400 kHz 模式	T <sub>CY</sub> (BRG + 1)	—	μs	
			1 MHz 模式 <sup>(2)</sup>	T <sub>CY</sub> (BRG + 1)	—	μs	
IM33	TSU:STO	停止条件建立时间	100 kHz 模式	T <sub>CY</sub> (BRG + 1)	—	μs	
			400 kHz 模式	T <sub>CY</sub> (BRG + 1)	—	μs	
			1 MHz 模式 <sup>(2)</sup>	T <sub>CY</sub> (BRG + 1)	—	μs	
IM34	THD:STO	停止条件保持时间	100 kHz 模式	T <sub>CY</sub> (BRG + 1)	—	ns	
			400 kHz 模式	T <sub>CY</sub> (BRG + 1)	—	ns	
			1 MHz 模式 <sup>(2)</sup>	T <sub>CY</sub> (BRG + 1)	—	ns	

注 1: BRG 是 I<sup>2</sup>C™ 波特率发生器的值。详情请参见第 17.2 节“设置用作总线主器件时的波特率”。

注 2: 对于所有 I<sup>2</sup>C 引脚, 最大引脚电容 = 10 pF (仅针对 1 MHz 模式)。

# PIC24FJ128GB204 系列

图 33-14: I<sup>2</sup>C™ 总线数据时序特性 (主模式)

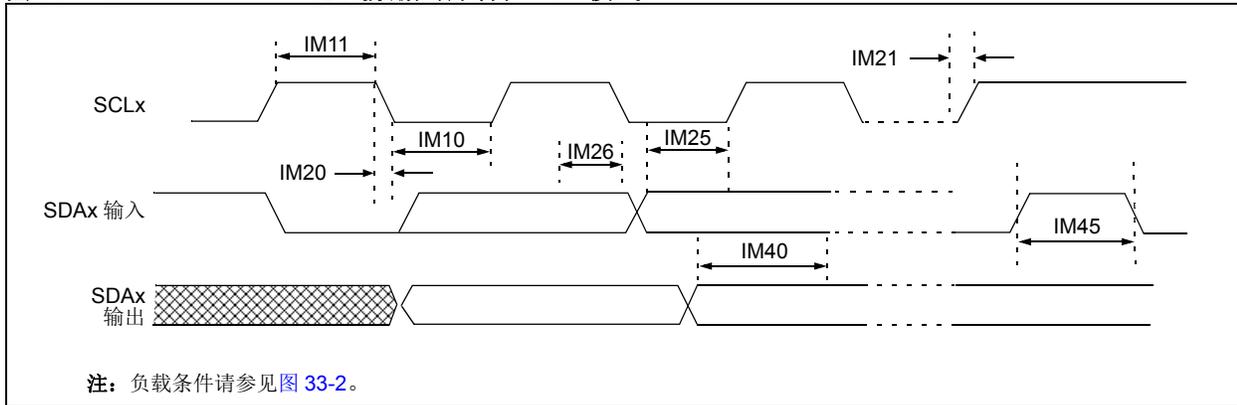


表 33-37: I<sup>2</sup>C™ 总线数据时序要求 (主模式)

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 ≤ T <sub>A</sub> ≤ +85°C (工业级) -40°C ≤ T <sub>A</sub> ≤ +125°C (扩展级)					
参数编号	符号	特性		最小值 <sup>(1)</sup>	最大值	单位	条件
IM10	TLO:SCL	时钟低电平时间	100 kHz 模式	T <sub>cy</sub> (BRG + 1)	—	μs	
			400 kHz 模式	T <sub>cy</sub> (BRG + 1)	—	μs	
			1 MHz 模式 <sup>(2)</sup>	T <sub>cy</sub> (BRG + 1)	—	μs	
IM11	THI:SCL	时钟高电平时间	100 kHz 模式	T <sub>cy</sub> (BRG + 1)	—	μs	
			400 kHz 模式	T <sub>cy</sub> (BRG + 1)	—	μs	
			1 MHz 模式 <sup>(2)</sup>	T <sub>cy</sub> (BRG + 1)	—	μs	
IM20	TF:SCL	SDAx 和 SCLx 下降时间	100 kHz 模式	—	300	ns	规定 C <sub>b</sub> 在 10 至 400 pF 之间
			400 kHz 模式	20 + 0.1 C <sub>b</sub>	300	ns	
			1 MHz 模式 <sup>(2)</sup>	—	100	ns	
IM21	TR:SCL	SDAx 和 SCLx 上升时间	100 kHz 模式	—	1000	ns	规定 C <sub>b</sub> 在 10 至 400 pF 之间
			400 kHz 模式	20 + 0.1 C <sub>b</sub>	300	ns	
			1 MHz 模式 <sup>(2)</sup>	—	300	ns	
IM25	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns	
			400 kHz 模式	100	—	ns	
			1 MHz 模式 <sup>(2)</sup>	40	—	ns	
IM26	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns	
			400 kHz 模式	0	0.9	μs	
			1 MHz 模式 <sup>(2)</sup>	0.2	—	ns	
IM40	TAA:SCL	时钟输出有效的 时间	100 kHz 模式	—	3500	ns	
			400 kHz 模式	—	1000	ns	
			1 MHz 模式 <sup>(2)</sup>	—	400	ns	
IM45	TBF:SDA	总线空闲时间	100 kHz 模式	4.7	—	μs	在新的传输开始前, 总线必须保持空闲的时间
			400 kHz 模式	1.3	—	μs	
			1 MHz 模式 <sup>(2)</sup>	0.5	—	μs	
IM50	C <sub>b</sub>	总线容性负载		—	400	pF	

注 1: BRG 是 I<sup>2</sup>C™ 波特率发生器的值。详情请参见第 17.2 节“设置用作总线主器件时的波特率”。

注 2: 对于所有 I<sup>2</sup>C 引脚, 最大引脚电容 = 10 pF (仅针对 1 MHz 模式)。

图 33-15: I<sup>2</sup>C™ 总线启动 / 停止位时序特性 (从模式)

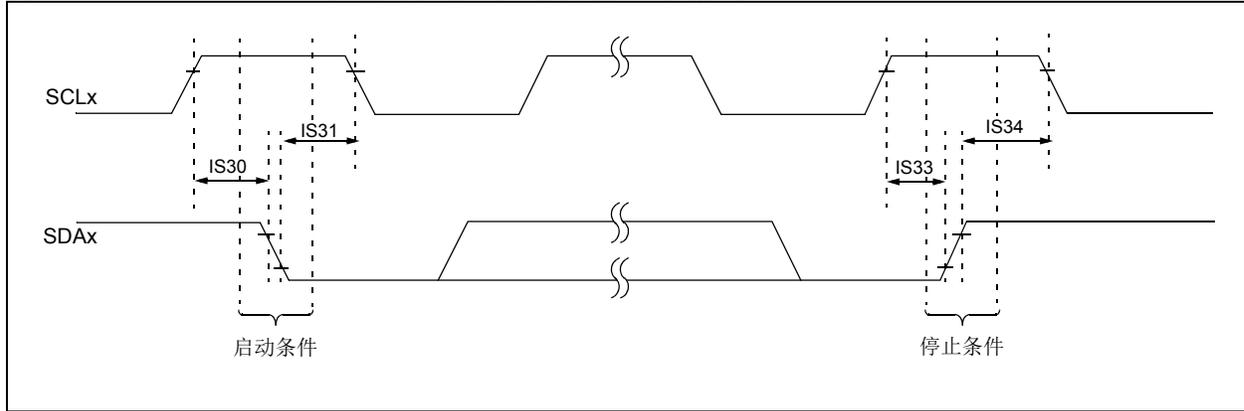


表 33-38: I<sup>2</sup>C™ 总线启动 / 停止位时序要求 (从模式)

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) -40°C ≤ TA ≤ +125°C (扩展级)					
参数编号	符号	特性	最小值	最大值	单位	条件	
IS30	TSU:STA	启动条件建立时间	100 kHz 模式	4.7	—	μs	仅与重复启动条件相关
			400 kHz 模式	0.6	—	μs	
			1 MHz 模式 <sup>(1)</sup>	0.25	—	μs	
IS31	THD:STA	启动条件保持时间	100 kHz 模式	4.0	—	μs	该周期后, 产生第一个时钟脉冲
			400 kHz 模式	0.6	—	μs	
			1 MHz 模式 <sup>(1)</sup>	0.25	—	μs	
IS33	TSU:STO	停止条件建立时间	100 kHz 模式	4.7	—	μs	
			400 kHz 模式	0.6	—	μs	
			1 MHz 模式 <sup>(1)</sup>	0.6	—	μs	
IS34	THD:STO	停止条件保持时间	100 kHz 模式	4000	—	ns	
			400 kHz 模式	600	—	ns	
			1 MHz 模式 <sup>(1)</sup>	250	—	ns	

注 1: 对于所有 I<sup>2</sup>C 引脚, 最大引脚电容 = 10 pF (仅针对 1 MHz 模式)。

# PIC24FJ128GB204 系列

图 33-16: I<sup>2</sup>C™ 总线数据时序特性 (从模式)

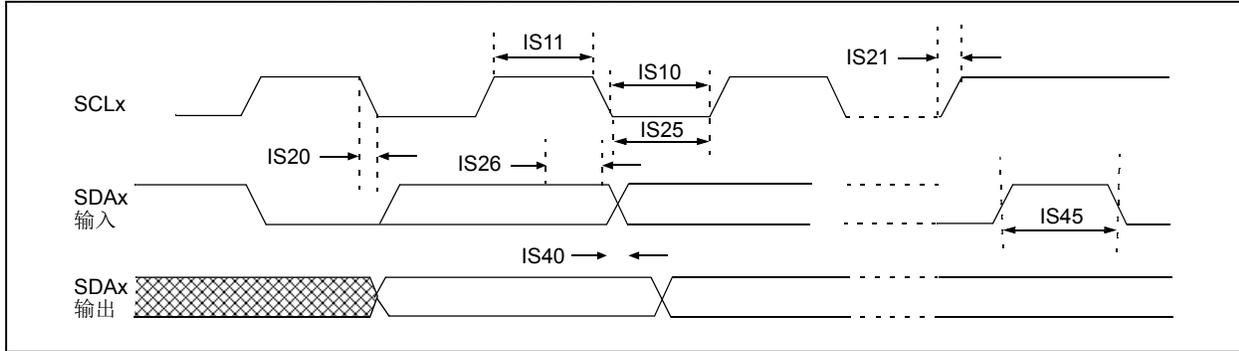


表 33-39: I<sup>2</sup>C™ 总线数据时序要求 (从模式)

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 ≤ T <sub>A</sub> ≤ +85°C (工业级) -40°C ≤ T <sub>A</sub> ≤ +125°C (扩展级)						
参数编号	符号	特性	最小值	最大值	单位	条件		
IS10	TLO:SCL	时钟低电平时间	100 kHz 模式	4.7	—	μs	器件必须在至少 1.5 MHz 频率下运行	
			400 kHz 模式	1.3	—	μs		器件必须在至少 10 MHz 频率下运行
			1 MHz 模式 <sup>(1)</sup>	0.5	—	μs		
IS11	THI:SCL	时钟高电平时间	100 kHz 模式	4.0	—	μs	器件必须在至少 1.5 MHz 频率下运行	
			400 kHz 模式	0.6	—	μs		器件必须在至少 10 MHz 频率下运行
			1 MHz 模式 <sup>(1)</sup>	0.5	—	μs		
IS20	TF:SCL	SDAx和SCLx下降时间	100 kHz 模式	—	300	ns	规定 C <sub>B</sub> 在 10 至 400 pF 之间	
			400 kHz 模式	20 + 0.1 C <sub>B</sub>	300	ns		
			1 MHz 模式 <sup>(1)</sup>	—	100	ns		
IS21	TR:SCL	SDAx和SCLx上升时间	100 kHz 模式	—	1000	ns	规定 C <sub>B</sub> 在 10 至 400 pF 之间	
			400 kHz 模式	20 + 0.1 C <sub>B</sub>	300	ns		
			1 MHz 模式 <sup>(1)</sup>	—	300	ns		
IS25	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns		
			400 kHz 模式	100	—	ns		
			1 MHz 模式 <sup>(1)</sup>	100	—	ns		
IS26	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns		
			400 kHz 模式	0	0.9	μs		
			1 MHz 模式 <sup>(1)</sup>	0	0.3	μs		
IS40	TAA:SCL	时钟输出有效时间	100 kHz 模式	0	3500	ns		
			400 kHz 模式	0	1000	ns		
			1 MHz 模式 <sup>(1)</sup>	0	350	ns		
IS45	TBF:SDA	总线空闲时间	100 kHz 模式	4.7	—	μs	在新的传输开始前, 总线必须保持空闲的时间	
			400 kHz 模式	1.3	—	μs		
			1 MHz 模式 <sup>(1)</sup>	0.5	—	μs		
IS50	C <sub>B</sub>	总线容性负载	—	400	pF			

注 1: 对于所有 I<sup>2</sup>C 引脚, 最大引脚电容 = 10 pF (仅针对 1 MHz 模式)。

# PIC24FJ128GB204 系列

表 33-40: A/D 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 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (扩展级)					
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
<b>器件电源</b>							
AD01	AVDD	模块电源 VDD	取 VDD - 0.3 或 2.2 中的 较大值	—	取 VDD + 0.3 或 3.6 中的 较小值	V	
AD02	AVSS	模块电源 VSS	VSS - 0.3	—	VSS + 0.3	V	
<b>参考输入</b>							
AD05	VREFH	参考电压高电压	AVSS + 1.7	—	AVDD	V	
AD06	VREFL	参考电压低电压	AVSS	—	AVDD - 1.7	V	
AD07	VREF	绝对参考电压	AVSS - 0.3	—	AVDD + 0.3	V	
<b>模拟输入</b>							
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	绝对输入电压 VINL	AVSS - 0.3	—	AVDD/3	V	
AD13		泄漏电流	—	$\pm 1.0$	$\pm 610$	nA	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V, 源阻抗 = 2.5 k $\Omega$
AD17	RIN	模拟电压源的推荐阻抗	—	—	2.5K	$\Omega$	10 位
<b>A/D 精度</b>							
AD20B	Nr	分辨率	—	12	—	位	
AD21B	INL	积分非线性误差	—	$\pm 1$	$< \pm 2$	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22B	DNL	微分非线性误差	—	—	$< \pm 1$	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23B	GERR	增益误差	—	$\pm 1$	$\pm 3$	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD24B	Eoff	失调误差	—	$\pm 1$	$\pm 2$	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD25B		单调性 (1)	—	—	—	—	保证

注 1: A/D 转换结果不会因输入电压的增加而减小, 并且不会丢失代码。

2: 测量是在使用外部 VREF+ 和 VREF- 作为 A/D 参考电压的情况下进行的。

# PIC24FJ128GB204 系列

表 33-41: A/D 转换时序要求 <sup>(1)</sup>

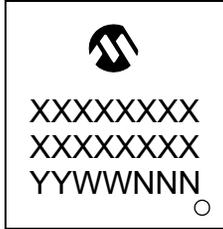
交流特性			标准工作条件: 2.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ Ta ≤ +85°C (工业级) -40°C ≤ Ta ≤ +125°C (扩展级)				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
<b>时钟参数</b>							
AD50	TAD	A/D 时钟周期	75	—	—	ns	Tcy = 75 ns, AD1CON3 处于默认状态
AD51	tRC	A/D 内部 RC 振荡器周期	—	250	—	ns	
<b>转换速率</b>							
AD55	tCONV	转换时间	—	14	—	TAD	
AD56	FCNV	吞吐率	—	—	200	ksps	AVDD > 2.7V
AD57	tSAMP	采样时间	—	1	—	TAD	
<b>时钟参数</b>							
AD61	tpSS	从采样位 (SAMP) 置 1 到采样开始的时间	2	—	3	TAD	

注 1: 因为采样电容最终将无法保持电荷, 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。

## 34.0 封装信息

### 34.1 封装标识信息

28 引脚 QFN-S



示例



28 引脚 SOIC (.300")



示例



28 引脚 SPDIP



示例



28 引脚 SSOP



示例



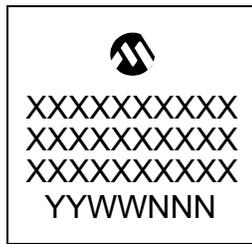
<b>图注:</b>	XX...X	客户指定信息
	Y	年份代码 (日历年的最后一位数字)
	YY	年份代码 (日历年的最后两位数字)
	WW	星期代码 (一月一日的星期代码为“01”)
	NNN	以字母数字排序的追踪代码
<b>注:</b>	Microchip 部件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户信息的字符数。	

# PIC24FJ128GB204 系列

---

## 34.1 封装标识信息 (续)

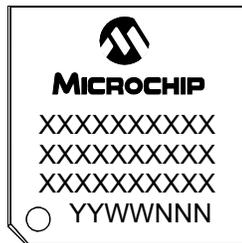
44 引脚 QFN



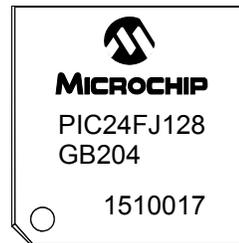
示例



44 引脚 TQFP



示例



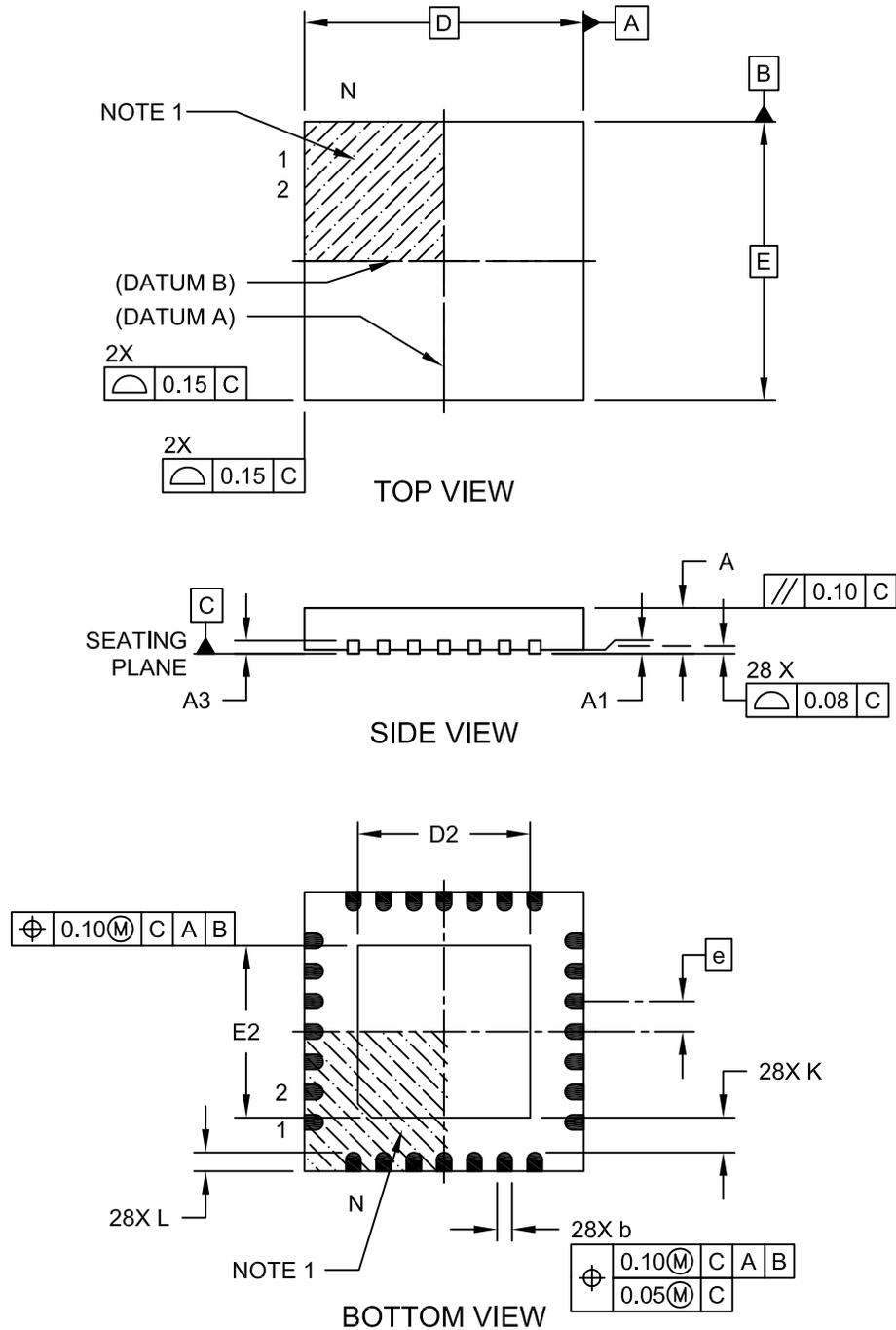
# PIC24FJ128GB204 系列

## 34.2 封装详细信息

以下部分将介绍各种封装的技术细节。

**28 引脚塑封四方扁平无脚封装 (MM) —— 主体 6x6x0.9 mm [QFN-S], 接线端子长度为 0.40 mm**

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。

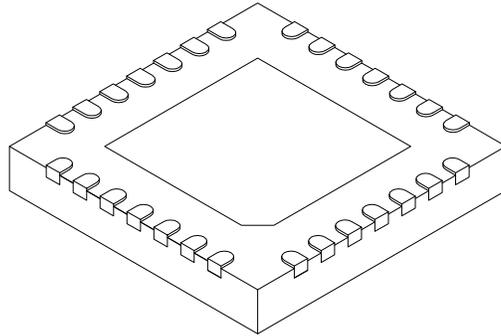


Microchip Technology Drawing C04-124C Sheet 1 of 2

# PIC24FJ128GB204 系列

28 引脚塑封四方扁平无脚封装 (MM) —— 主体 6x6x0.9 mm [QFN-S], 接线端子长度为 0.40 mm

注: 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3	0.20 REF		
Overall Width	E	6.00 BSC		
Exposed Pad Width	E2	3.65	3.70	4.70
Overall Length	D	6.00 BSC		
Exposed Pad Length	D2	3.65	3.70	4.70
Terminal Width	b	0.23	0.30	0.35
Terminal Length	L	0.30	0.40	0.50
Terminal-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.

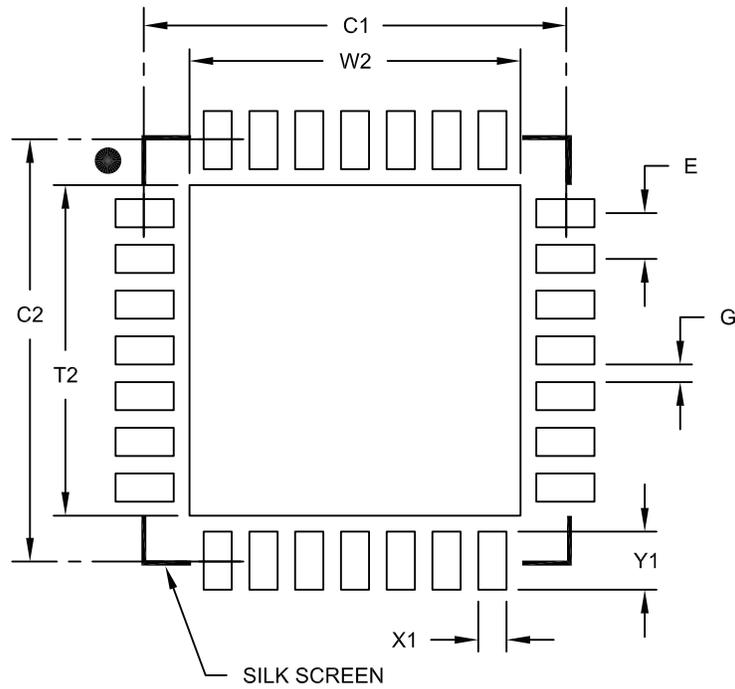
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-124C Sheet 2 of 2

# PIC24FJ128GB204 系列

28 引脚塑封四方扁平无脚封装 (MM) —— 主体 6x6x0.9 mm [QFN-S], 接线端子长度为 0.40 mm

注: 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			4.70
Optional Center Pad Length	T2			4.70
Contact Pad Spacing	C1		6.00	
Contact Pad Spacing	C2		6.00	
Contact Pad Width (X28)	X1			0.40
Contact Pad Length (X28)	Y1			0.85
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

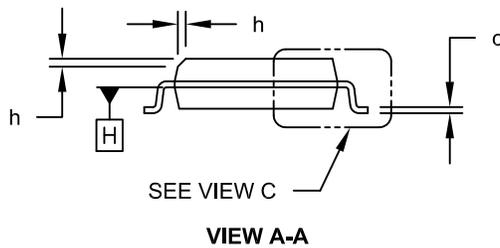
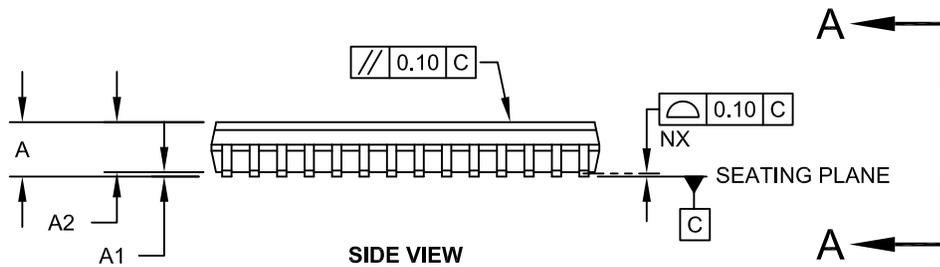
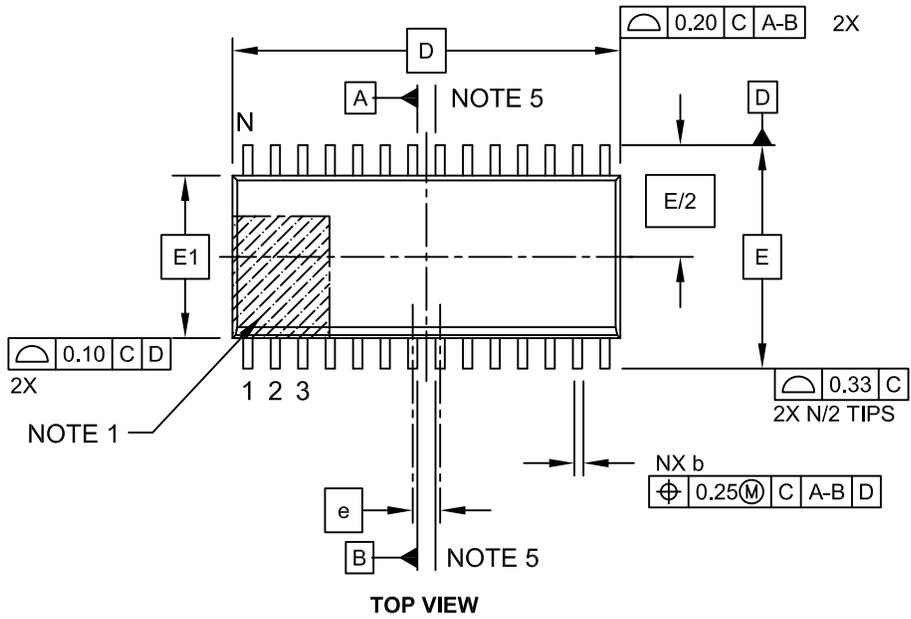
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2124A

# PIC24FJ128GB204 系列

## 28 引脚塑封宽条小外形封装 (SO) —— 主体 7.50 mm [SOIC]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。

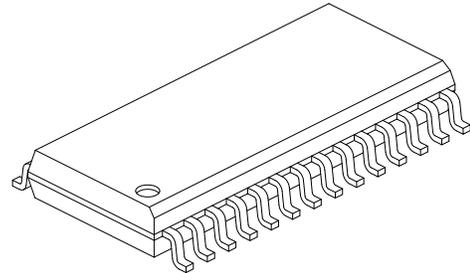
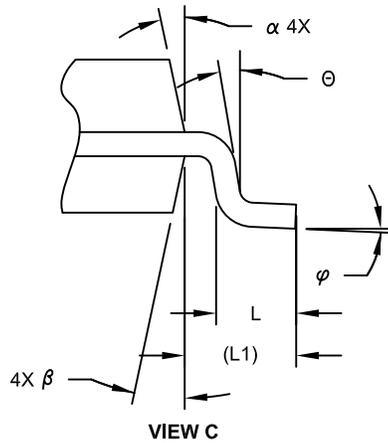


Microchip Technology Drawing C04-052C Sheet 1 of 2

# PIC24FJ128GB204 系列

## 28 引脚塑封宽条小外形封装 (SO) —— 主体 7.50 mm [SOIC]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	17.90 BSC		
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1	1.40 REF		
Lead Angle	$\theta$	0°	-	-
Foot Angle	$\varphi$	0°	-	8°
Lead Thickness	c	0.18	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	$\alpha$	5°	-	15°
Mold Draft Angle Bottom	$\beta$	5°	-	15°

### Notes:

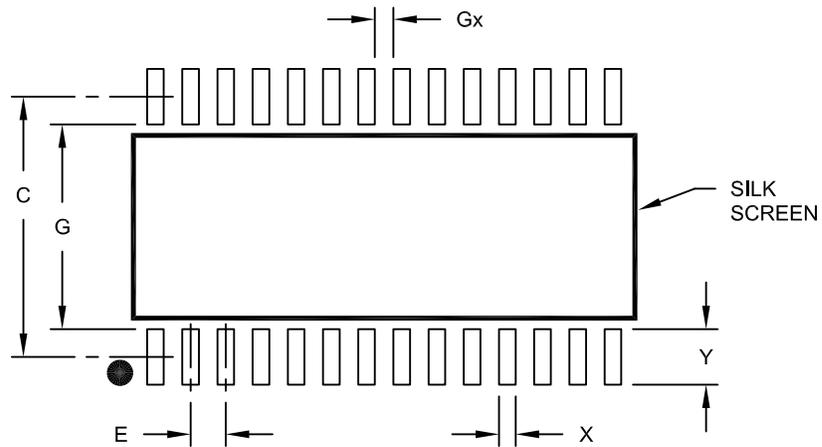
- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

# PIC24FJ128GB204 系列

## 28 引脚塑封宽条小外形封装 (SO) —— 主体 7.50 mm [SOIC]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



### RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

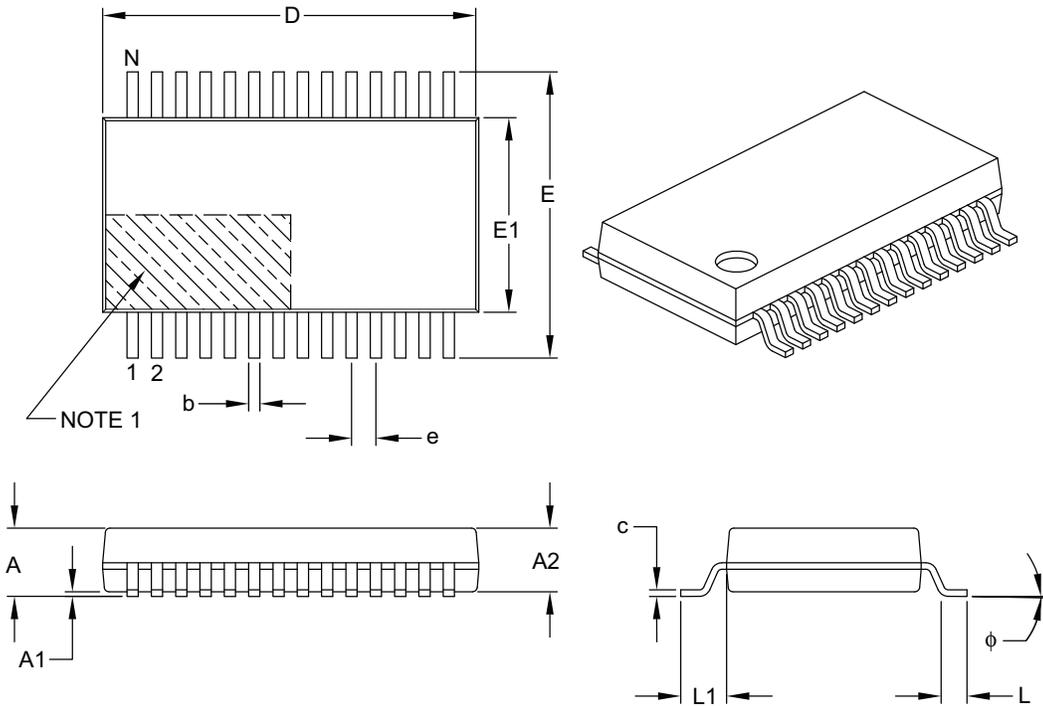
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

# PIC24FJ128GB204 系列

## 28 引脚塑封缩小型小外形封装 (SS) —— 主体 5.30 mm [SSOP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	c	0.09	–	0.25
Foot Angle	φ	0°	4°	8°
Lead Width	b	0.22	–	0.38

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

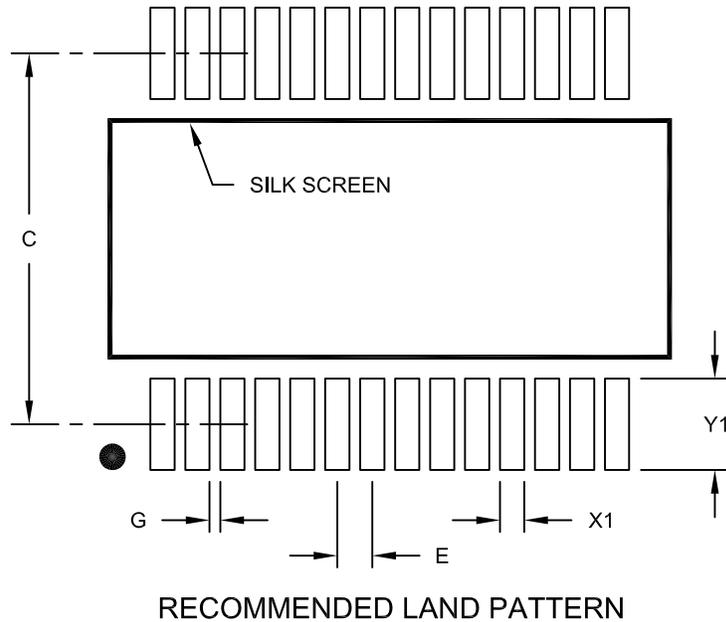
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

# PIC24FJ128GB204 系列

## 28 引脚塑封缩小型小外形封装 (SS) —— 主体 5.30 mm [SSOP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

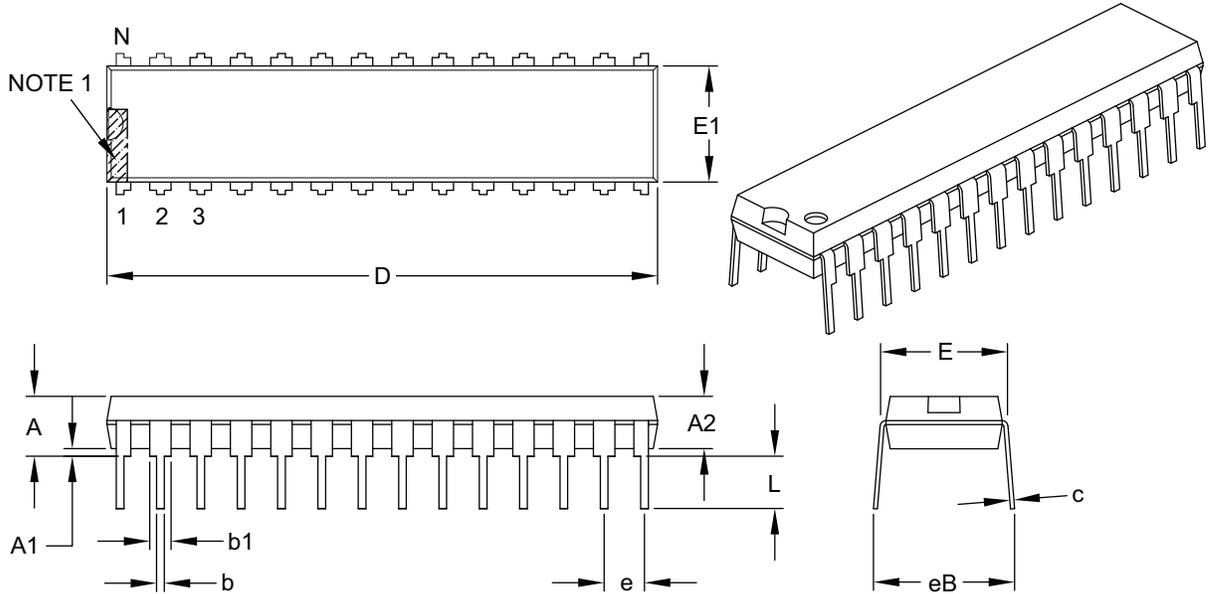
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2073A

# PIC24FJ128GB204 系列

## 28 引脚窄条塑封双列直插式封装 (SP) —— 主体 300 mil [SPDIP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

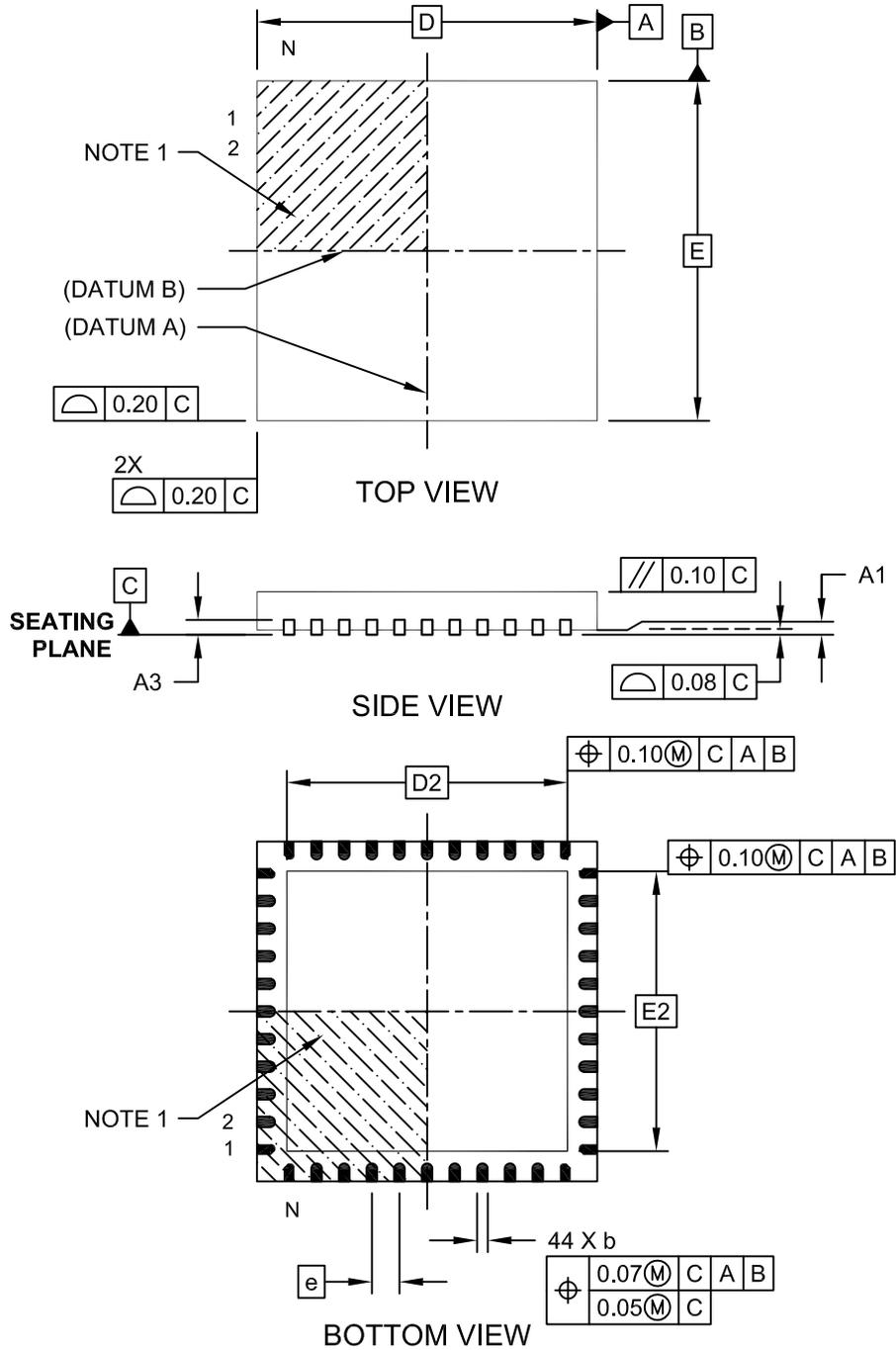
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

# PIC24FJ128GB204 系列

## 44 引脚塑封四方扁平无脚封装 (ML) —— 主体 8x8 mm [QFN]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。

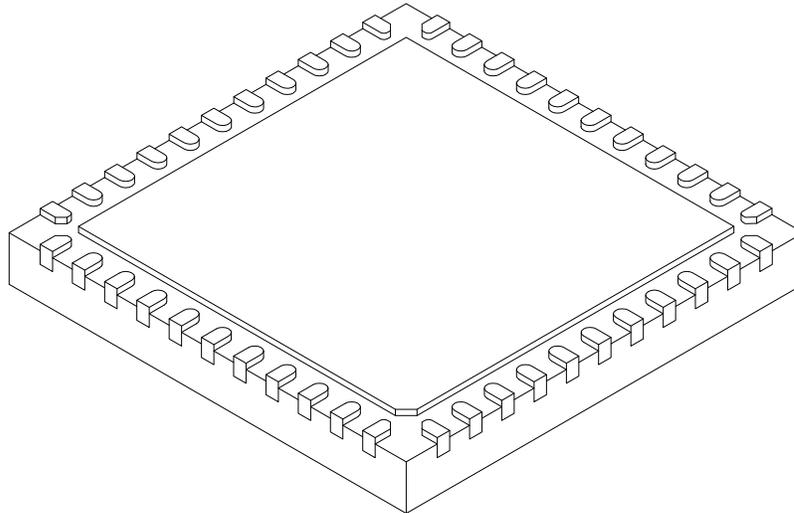


Microchip Technology Drawing C04-103C Sheet 1 of 2

# PIC24FJ128GB204 系列

## 44 引脚塑封四方扁平无脚封装 (ML) —— 主体 8x8 mm [QFN]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	44		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3	0.20 REF		
Overall Width	E	8.00 BSC		
Exposed Pad Width	E2	6.25	6.45	6.60
Overall Length	D	8.00 BSC		
Exposed Pad Length	D2	6.25	6.45	6.60
Terminal Width	b	0.20	0.30	0.35
Terminal Length	L	0.30	0.40	0.50
Terminal-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.

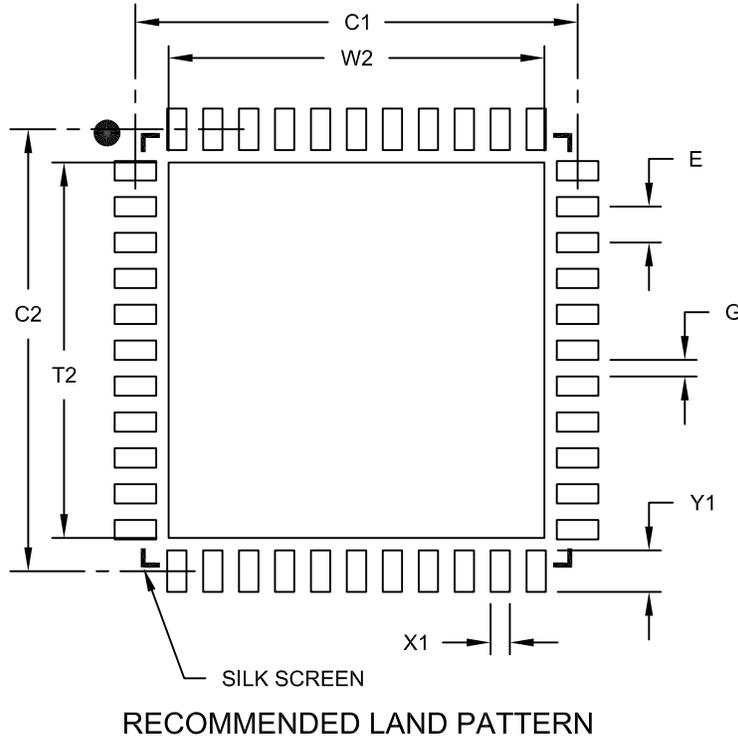
REF: Reference Dimension. usually without tolerance. for information purposes only.

Microchip Technology Drawing C04-103C Sheet 2 of 2

# PIC24FJ128GB204 系列

## 44 引脚塑封四方扁平无脚封装 (ML) —— 主体 8x8 mm [QFN]

注: 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			6.60
Optional Center Pad Length	T2			6.60
Contact Pad Spacing	C1		8.00	
Contact Pad Spacing	C2		8.00	
Contact Pad Width (X44)	X1			0.35
Contact Pad Length (Y44)	Y1			0.85
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

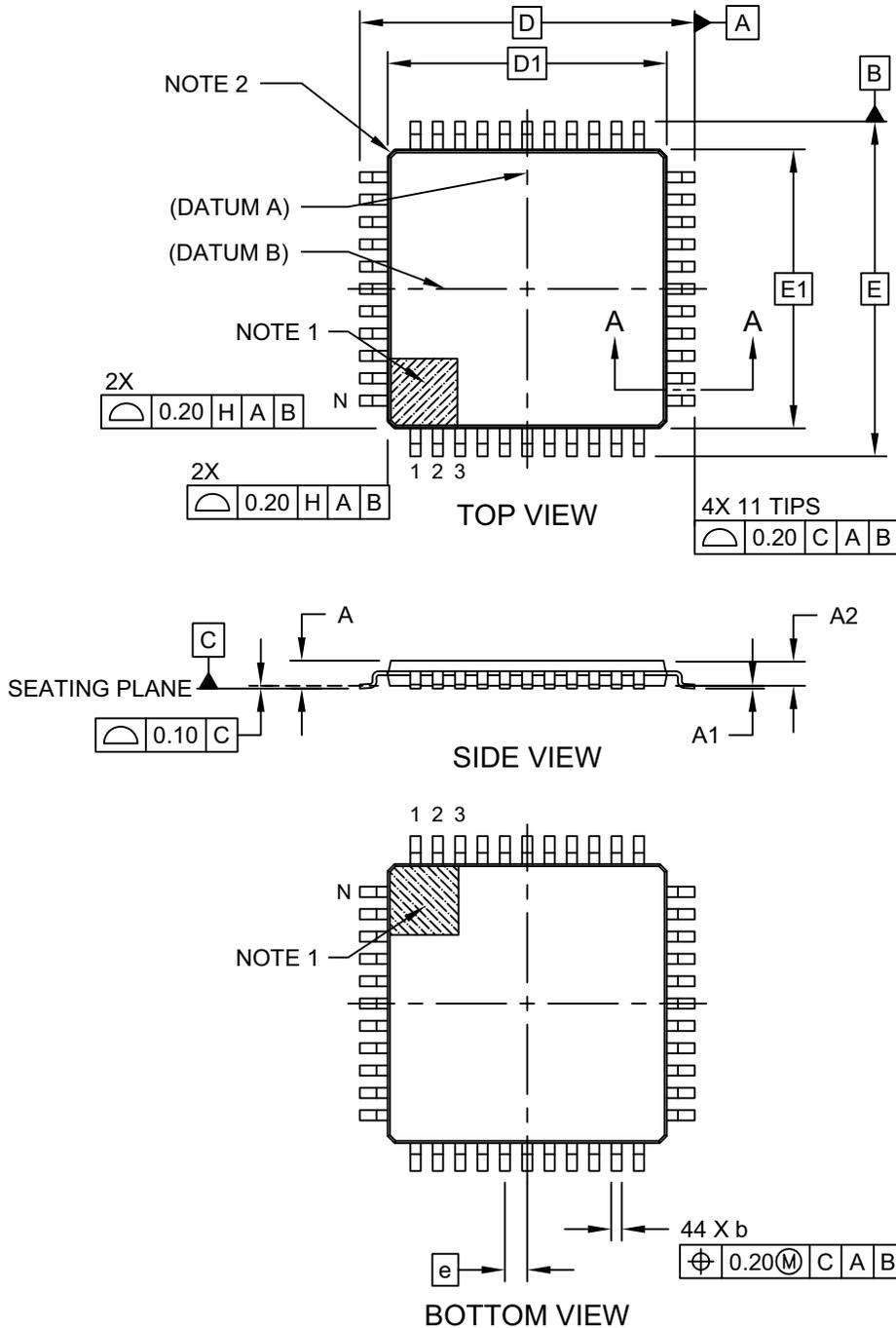
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2103B

# PIC24FJ128GB204 系列

## 44 引脚塑封薄型四方扁平封装 (PT) —— 主体 10x10x1 mm [TQFP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。

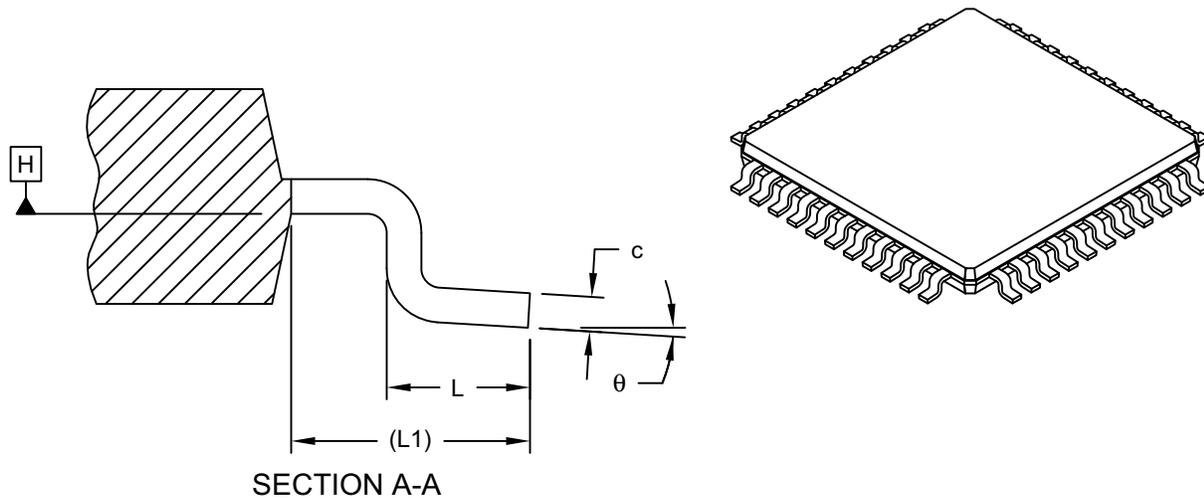


Microchip Technology Drawing C04-076C Sheet 1 of 2

# PIC24FJ128GB204 系列

## 44 引脚塑封薄型四方扁平封装 (PT) —— 主体 10x10x1 mm [TQFP]

注： 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Leads	N	44		
Lead Pitch	e	0.80 BSC		
Overall Height	A	-	-	1.20
Standoff	A1	0.05	-	0.15
Molded Package Thickness	A2	0.95	1.00	1.05
Overall Width	E	12.00 BSC		
Molded Package Width	E1	10.00 BSC		
Overall Length	D	12.00 BSC		
Molded Package Length	D1	10.00 BSC		
Lead Width	b	0.30	0.37	0.45
Lead Thickness	c	0.09	-	0.20
Lead Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	$\theta$	0°	3.5°	7°

Notes:

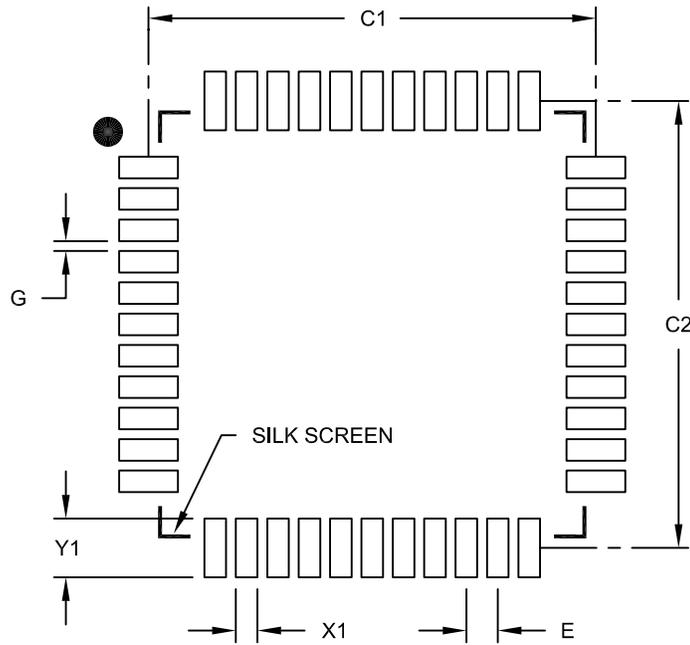
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Exact shape of each corner is optional.
3. Dimensioning and tolerancing per ASME Y14.5M  
 BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
 REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-076C Sheet 2 of 2

# PIC24FJ128GB204 系列

## 44 引脚塑封薄型四方扁平封装 (PT) —— 主体 10x10x1 mm [TQFP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.80 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X44)	X1			0.55
Contact Pad Length (X44)	Y1			1.50
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2076B

# PIC24FJ128GB204 系列

---

注:

## 附录 A： 版本历史

### 版本 A（2013 年 5 月）

PIC24FJ128GB204 系列器件的初始数据手册。

### 版本 B（2014 年 5 月）

该版本包含以下更新：

- 章节：
  - 在“加密引擎”中插入新的条目
  - 更新了“模拟特性”中的单元
  - 更新了第 16.0 节“串行外设接口（SPI）”、第 17.0 节“ $I^2C$ ™”、第 18.0 节“通用异步收发器（UART）”、第 23.0 节“加密引擎”和第 27.0 节“比较器参考电压”中的注
  - 更新了第 17.3 节“从地址掩码”
  - 更新了第 23.0 节“加密引擎”
  - 插入新章节第 23.5.6 节“生成随机数”
  - 更新了第 30.3.1 节“窗口操作”
  - 更新了第 34.0 节“封装信息”中的封装信息
- 寄存器：
  - 更新了寄存器 8-45、寄存器 16-1、寄存器 16-4、寄存器 17-1、寄存器 18-1、寄存器 20-1、寄存器 23-1 和寄存器 23-5
  - 更新了寄存器 18-2 和寄存器 18-4 的标题
  - 更新了寄存器 23-5 中 bit 10 的寄存器描述
- 表：
  - 更新了表 1-3、表 4-5、表 4-9、表 4-10、表 4-11、表 4-12、表 4-13、表 4-14、表 4-29、表 33-1、表 33-3、表 33-4、表 33-5、表 33-6、表 33-7、表 33-8、表 33-10、表 33-12、表 33-13、表 33-14、表 33-15、表 33-16 和表 33-21
  - 添加了表 33-26 至表 33-39
- 图：
  - 添加了图 9-1、图 33-5、图 33-6、图 33-7、图 33-8、图 33-9、图 33-10、图 33-11 和图 33-12
- 示例：
  - 例 22-1

### 版本 C（2015 年 3 月）

本版本包含以下更新：

- 寄存器：
  - 寄存器 26-1
- 表：
  - 表 33-4、表 33-5、表 33-6 和表 33-21
  - 更新了第 34.0 节“封装信息”的封装标识示例和封装图

# PIC24FJ128GB204 系列

---

注:

## 索引

### 数字

16 位 MCU 入门 ..... 23

### A

#### A/D

操作 ..... 351  
 寄存器 ..... 354  
 扩展 DMA 操作 ..... 353  
 传递函数  
     10 位 ..... 370  
     12 位 ..... 369

#### AC 特性

I<sup>2</sup>C 总线数据 (主模式) ..... 442  
 I<sup>2</sup>C 总线数据 (从模式) ..... 444  
 I<sup>2</sup>C 总线启动 / 停止位时序要求 (主模式) ..... 441  
 I<sup>2</sup>C 总线启动 / 停止位 (从模式) ..... 443  
 输入捕捉 x 时序要求 ..... 434  
 输出比较要求 ..... 434  
 简单 OCx/PWM 模式要求 ..... 435  
 SPIx 主模式 (CKE = 0) 要求 ..... 436  
 SPIx 主模式 (CKE = 1) 要求 ..... 437  
 SPIx 从模式 (CKE = 0) 要求 ..... 438  
 SPIx 从模式 (CKE = 1) 要求 ..... 440  
 Timer1 外部时钟要求 ..... 432

AD1CHITL (A/D 扫描比较命中寄存器,  
 低位字) ..... 365  
 AD1CHS (A/D 采样选择寄存器) ..... 363  
 AD1CON5 (A/D 控制寄存器) ..... 362  
 AD1CSSH (A/D 输入扫描选择寄存器,  
 高位字) ..... 366  
 AD1CSSL (A/D 输入扫描选择寄存器,  
 低位字) ..... 366  
 AD1CTMENL (CTMU 使能寄存器,  
 低位字) ..... 367  
 ALCFGRPT (闹钟配置寄存器) ..... 320  
 ANCFG (A/D 带隙参考  
 配置寄存器) ..... 364

### B

版本历史 ..... 465  
 备用中断向量表  
     (AIVT) ..... 91  
 比较器参考电压 ..... 377  
     配置 ..... 377  
 比较器参考电压规范 ..... 425  
 变更通知客户服务 ..... 473

### C

C 编译器  
     MPLAB C18 ..... 404  
 CPU  
     编程模型 ..... 29  
     控制寄存器 ..... 32  
     内核寄存器 ..... 30  
     算术逻辑单元 (ALU) ..... 34  
 CRC  
     16 和 32 位多项式设置示例 ..... 346  
     多项式 ..... 346  
     用户接口 ..... 346  
 CTMU

测量电容 ..... 379  
 测量时间 ..... 381  
 脉冲生成和延时 ..... 381  
 产品标识体系 ..... 475  
 程序存储空间与数据存储空间接口 ..... 65  
 程序存储器  
     存储器硬编码向量 ..... 36  
     存储器映射 ..... 35  
     地址构成 ..... 65  
     地址空间 ..... 35  
     构成 ..... 36  
     闪存配置字 ..... 36  
     使用表指令访问 ..... 67  
     使用 EDS 从程序存储器读数据 ..... 68  
 程序校验 ..... 401  
 充电时间测量单元。见 CTMU。  
 串行外设接口。见 SPI。  
 串行外设接口 (SPI) ..... 227  
 存储器构成 ..... 35

### D

DMA ..... 71  
     操作汇总 ..... 72  
     典型的设置 ..... 74  
     通道触发源 ..... 78  
     外设模块禁止 (PMD) ..... 74  
     传输类型 ..... 73  
 DMA 控制器 ..... 10  
 代码保护 ..... 401  
     代码段保护 ..... 401  
         配置选项 ..... 401  
     配置寄存器保护 ..... 402  
     通用段保护 ..... 401  
 代码示例  
     编程闪存存储器单字 ..... 84  
     编程闪存存储器的一个单字  
         ("C" 语言) ..... 84  
     擦除程序存储器块  
         (C 语言代码) ..... 83  
     擦除程序存储器块 (汇编) ..... 82  
     从程序存储器进行 EDS 读操作的代码  
         (汇编语言) ..... 68  
     端口读 / 写 (C 语言) ..... 177  
     端口读 / 写 (汇编语言) ..... 177  
     EDS 读 (汇编语言) ..... 62  
     EDS 写 (汇编语言) ..... 63  
     基本时钟切换序列 ..... 154  
     将 RTCWREN 位置 1 ..... 316  
     PWRSV 指令语法 ..... 162  
     配置 UART1 输入 / 输出功能 ..... 182  
     启动编程序列 ..... 83  
     重复序列 ..... 164  
     装载写缓冲器 ..... 83  
 低电压 / 保持稳压器 ..... 163  
 电气特性  
     绝对最大值 ..... 415  
     V/F 图 (工业级) ..... 416  
     调制器。请参见数据信号调制器 ..... 299  
     读者反馈表 ..... 474  
     端口或引脚可承受的输入电压说明 ..... 174

# PIC24FJ128GB204 系列

<b>E</b>		
EPMP		
不同模式下存储器寻址	.....	303
主要特性	.....	303
<b>F</b>		
封装	.....	449
标识	.....	449
详细信息	.....	451
复位		
BOR (欠压复位)	.....	85
CM (配置不匹配复位)	.....	85
IOPUWR (非法操作码复位)	.....	85
MCLR (引脚复位)	.....	85
POR (上电复位)	.....	85
器件时间	.....	89
欠压复位 (BOR)	.....	89
RCON 标志, 操作	.....	88
SFR 状态	.....	89
SWR (RESET 指令)	.....	85
时钟源选择	.....	89
TRAPR (陷阱冲突复位)	.....	85
WDT (看门狗定时器复位)	.....	85
UWR (未初始化的 W 寄存器复位)	.....	85
延时时间	.....	90
<b>G</b>		
高 / 低压检测 (HLVD)	.....	387
公式		
16 位和 32 位 CRC 多项式	.....	346
A/D 转换时钟周期	.....	368
BRGH = 0 时的 UARTx 波特率	.....	255
BRGH = 1 时的 UARTx 波特率	.....	255
波特率重载值	.....	247
估算 USB 收发器电流损耗	.....	269
计算 PWM 周期	.....	220
计算最大 PWM 分辨率	.....	221
器件速率和 SPIx 时钟速率之间的关系	.....	243
<b>H</b>		
汇编器		
MPASM 汇编器	.....	404
<b>I</b>		
I/O 端口		
并行 (PIO)	.....	173
可选输出源	.....	180
可选输入源	.....	179
漏极开路配置	.....	174
模拟端口引脚配置 (ANSx)	.....	174
配置 I/O 引脚的模拟 / 数字功能	.....	174
上拉和下拉	.....	177
输入电平变化通知 (ICN)	.....	177
外设引脚选择	.....	178
写 / 读时序	.....	174
I <sup>2</sup> C		
保留地址	.....	247
从地址掩码	.....	247
设置用作总线主器件时的波特率	.....	247
作为主器件在单主器件环境中通信	.....	245
<b>J</b>		
寄存器		
AD1CON1 (A/D 控制 1)	.....	356
AD1CON2 (A/D 控制 2)	.....	358
AD1CON3 (A/D 控制 3)	.....	360
AD1CON4 (A/D 控制 4)	.....	361
ALCFGRPT (闹钟配置)	.....	320
ALMINSEC (闹钟分钟和秒值)	.....	324
ALMTHDY (闹钟月和日值)	.....	323
ALWDHR (闹钟星期和小时值)	.....	323
ANSA (PORTA 模拟功能选择)	.....	175
ANSB (PORTB 模拟功能选择)	.....	175
ANSC (PORTC 模拟功能选择)	.....	176
BDnSTAT (缓冲区描述符 n 状态原型, CPU 模式)	.....	273
BDnSTAT (缓冲区描述符 n 状态原型, USB 模式)	.....	272
CFGPAGE (安全阵列配置位)	.....	340
CLKDIV (时钟分频器)	.....	151
CMSTAT (比较器状态)	.....	375
CMxCON (比较器 x 控制, 比较器 1-3)	.....	374
CORCON (CPU 内核控制)	.....	33, 97
CRCCON1 (CRC 控制 1)	.....	348
CRCCON2 (CRC 控制 2)	.....	349
CRCXORH (CRC 异或操作, 高字节)	.....	350
CRCXORL (CRC 多项式异或操作, 低字节)	.....	350
CRYCONH (加密控制高字节)	.....	337
CRYCONL (加密控制低字节)	.....	335
CRYOTP (加密 OTP 页编程控制)	.....	339
CRYSTAT (加密状态)	.....	338
CTMUCON1 (CTMU 控制 1)	.....	382
CTMUCON2 (CTMU 控制 2)	.....	383
CTMUICON (CTMU 电流控制)	.....	385
CW1 (闪存配置字 1)	.....	390
CW2 (闪存配置字 2)	.....	392
CW3 (闪存配置字 3)	.....	394
CW4 (闪存配置字 4)	.....	396
CVRCON (比较器参考 电压控制)	.....	378
DEVID (器件 ID)	.....	398
DEVREV (器件版本)	.....	398
DMACHn (DMA 通道 n 控制)	.....	76
DMACON (DMA 引擎控制)	.....	75
DMAINTn (DMA 通道 n 中断)	.....	77
DSCON (深度休眠控制)	.....	168
DSWAKE (深度休眠唤醒源)	.....	169
HLVDCON (高 / 低压检测控制)	.....	388
I2C1CONH (I2C1 控制高字节)	.....	248
I2CxCONL (I2Cx 控制低字节)	.....	250
I2CxMSK (I2Cx 从模式地址掩码)	.....	252
I2CxSTAT (I2Cx 状态)	.....	251
ICxCON1 (输入捕捉 x 控制 1)	.....	213
ICxCON2 (输入捕捉 x 控制 2)	.....	214
IEC0 (中断允许控制 0)	.....	111
IEC1 (中断允许控制 1)	.....	113
IEC2 (中断允许控制 2)	.....	115
IEC3 (中断允许控制 3)	.....	117
IEC4 (中断允许控制 4)	.....	119
IEC5 (中断允许控制 5)	.....	120

# PIC24FJ128GB204 系列

IEC6 (中断允许控制 6)	121	RCON2 (复位和系统控制 2)	170
IEC7 (中断允许控制 7)	121	RCON2 (复位控制 2)	88
IFS0 (中断标志状态 0)	100	REFOCONH	
IFS1 (中断标志状态 1)	102	(参考振荡器控制高)	158
IFS2 (中断标志状态 2)	104	REFOCONL	
IFS3 (中断标志状态 3)	106	(参考振荡器控制低)	157
IFS4 (中断标志状态 4)	108	REFOTRIML	
IFS5 (中断标志状态 5)	109	(参考振荡器调节)	159
IFS6 (中断标志状态 6)	110	RPINR0 (PPS 输入 0)	183
IFS7 (中断标志状态 7)	110	RPINR1 (PPS 输入 1)	183
INTCON1 (中断控制 1)	98	RPINR11 (PPS 输入 11)	186
INTCON2 (中断控制 2)	99	RPINR17 (PPS 输入 17)	186
INTTREG (中断控制器测试)	144	RPINR18 (PPS 输入 18)	187
IPC0 (中断优先级控制 0)	122	RPINR19 (PPS 输入 19)	187
IPC1 (中断优先级控制 1)	123	RPINR2 (PPS 输入 2)	184
IPC10 (中断优先级控制 10)	132	RPINR20 (PPS 输入 20)	188
IPC10 (中断优先级控制 26)	143	RPINR21 (PPS 输入 21)	188
IPC11 (中断优先级控制 11)	133	RPINR22 (PPS 输入 22)	189
IPC12 (中断优先级控制 12)	134	RPINR23 (PPS 输入 23)	189
IPC13 (中断优先级控制 13)	135	RPINR27 (PPS 输入 27)	190
IPC14 (中断优先级控制 14)	136	RPINR28 (PPS 输入 28)	190
IPC15 (中断优先级控制 15)	137	RPINR29 (PPS 输入 29)	191
IPC16 (中断优先级控制 16)	138	RPINR30 (PPS 输入 30)	191
IPC18 (中断优先级控制 18)	139	RPINR31 (PPS 输入 31)	192
IPC19 (中断优先级控制 19)	139	RPINR7 (PPS 输入 7)	184
IPC2 (中断优先级控制 2)	124	RPINR8 (PPS 输入 8)	185
IPC20 (中断优先级控制 20)	140	RPINR9 (PPS 输入 9)	185
IPC21 (中断优先级控制 21)	141	RPOR0 (PPS 输出 0)	193
IPC22 (中断优先级控制 22)	142	RPOR1 (PPS 输出 1)	193
IPC29 (中断优先级控制 29)	143	RPOR10 (PPS 输出 10)	198
IPC3 (中断优先级控制 3)	125	RPOR11 (PPS 输出 11)	198
IPC4 (中断优先级控制 4)	126	RPOR12 (PPS 输出 12)	199
IPC5 (中断优先级控制 5)	127	RPOR2 (PPS 输出 2)	194
IPC6 (中断优先级控制 6)	128	RPOR3 (PPS 输出 3)	194
IPC7 (中断优先级控制 7)	129	RPOR4 (PPS 输出 4)	195
IPC8 (中断优先级控制		RPOR5 (PPS 输出 5)	195
寄存器 8)	130	RPOR6 (PPS 输出 6)	196
IPC9 (中断优先级控制 9)	131	RPOR7 (PPS 输出 7)	196
MDCAR (调制器载波控制)	302	RPOR8 (PPS 输出 8)	197
MDCON (调制器控制)	300	RPOR9 (PPS 输出 9)	197
MDSRC (调制器源控制)	301	RTCCSWT (RTCC 电源控制	
MINSEC (RTCC 分钟和秒值)	322	和采样窗口定时器)	325
MTHDY (RTCC 月和日值)	321	RTCPWC	
NVMCON (闪存控制)	81	(RTCC 电源控制)	319
OCxCON1 (输出比较 x		SPIxCON1H(SPIx 控制 1 高)	233
控制 1)	222	SPIxCON1L (SPIx 控制 1 低)	231
OCxCON2 (输出比较 x		SPIxCON2L (SPIx 控制 2 低)	235
控制 2)	224	SPIxIMSKH (SPIx 中断掩码高)	240
OSCCON (振荡器控制)	149	SPIxSTATH (SPIx 状态高)	238
OSCTUN (FRC 振荡器调节)	152	SPIxSTATL (SPIx 状态低)	236
PADCFG1 (填充配置控制)	314	SR (ALU 状态)	32, 96
PMCON1 (EPMP 控制 1)	306	T1CON (Timer1 控制)	202
PMCON2 (EPMP 控制 2)	307	TxCON (Timer2 和	
PMCON3 (EPMP 控制 3)	308	Timer4 控制)	208
PMCON4 (EPMP 控制 4)	309	TyCON (Timer3 和	
PMCSxBBS (片选 x 基地址)	311	Timer5 控制)	210
PMCSxCF (片选 x 配置)	310	U1ADDR (USB 地址)	286
PMCSxMD (片选 x 模式)	312	U1CNFG1 (USB 配置 1)	288
PMSTAT (EPMP 状态,		U1CNFG2 (USB 配置 2)	289
从模式)	313	U1CON (USB 控制,	
RCFGCAL (RTCC		设备模式)	284
校准和配置)	317	U1CON (USB 控制,	
RCON (复位控制)	86	主机模式)	285

# PIC24FJ128GB204 系列

U1EIE (USB 错误 中断允许) .....	296	中断控制器 .....	41
U1EIR (USB 错误中断状态) .....	295	JTAG 接口 .....	402
U1EPn (USB 端点 n 控制) .....	297	加密引擎 .....	10, 329
U1IE (USB 中断允许, 所有模式) .....	294	随机数发生 编程 .....	333
U1IR (USB 中断状态, 仅设备模式) .....	292	CFGPAGE 配置位 .....	334
U1IR (USB 中断状态, 仅主机模式) .....	293	密钥 .....	334
U1OTGCON (USB OTG 控制) .....	281	验证密钥 .....	334
U1OTGIE (USB OTG 中断允许, 仅主机模式) .....	291	操作模式 .....	330
U1OTGIR (USB OTG 中断状态, 仅主机模式) .....	290	空闲 .....	330
U1OTGSTAT (USB OTG 状态, 主机模式) .....	280	休眠 .....	330
U1PWRC (USB 电源控制) .....	282	测试密钥源配置 .....	333
U1SOF (USB OTG 帧起始计数, 主机模式) .....	287	会话密钥 .....	
U1STAT (USB 状态) .....	283	加密 .....	332
U1TOK (USB 令牌, 仅主机模式) .....	286	接收 .....	332
WKDYHR (RTCC 星期和小时值) .....	322	加密数据 .....	331
UxMODE (UARTx 模式) .....	258	解密数据 .....	331
UxSCCON (UARTx 智能卡控制) .....	263	使能 .....	330
UxSCINT (UARTx 智能卡中断) .....	264	数据寄存器空间 .....	330
UxSTAL (UARTx 状态 和控制低字节) .....	260	交流特性 .....	
UxTXREG (UARTx 发送) .....	262	A/D 规范 .....	445
YEAR (RTCC 年值) .....	321	A/D 转换时间 .....	446
寄存器映射 .....		CLKO 和 I/O 时序要求 .....	430
A/D 转换器 .....	51	复位和欠压复位要求 .....	431
比较器 .....	56	和时序参数 .....	427
并行主 / 从端口 .....	55	内部 RC 精度 .....	429
CPU 内核 .....	39	PLL 时钟时序规范 .....	429
CRC .....	57	RC 振荡器起振时间 .....	430
CRYPTO SFR .....	59	时序规范的负载条件和要求 .....	427
CTMU .....	52	输出引脚上的容性负载要求 .....	427
DMA .....	53	外部时钟时序要求 .....	428
定时器 .....	43	节能特性 .....	161
I <sup>2</sup> C .....	46	打盹模式 .....	171
ICN .....	40	基于指令的模式 .....	162
模拟配置 .....	52	空闲 .....	163
NVM .....	59	深度休眠 .....	164
PORTA .....	50	休眠 .....	163
PORTB .....	50	模式概述 .....	161
PORTC .....	50	时钟频率, 时钟切换 .....	171
SPI1 .....	48	VBAT 模式 .....	166
SPI2 .....	48	<b>K</b> .....	
SPI3 .....	49	开发支持 .....	403
深度休眠 .....	59	看门狗定时器 (WDT) .....	400
实时时钟和日历 (RTCC) .....	56	窗口操作 .....	400
输出比较 .....	45	控制寄存器 .....	400
数据信号调制器 (DSM) .....	56	勘误表 .....	7
输入捕捉 .....	44	客户通知服务 .....	473
UART .....	47	客户支持 .....	473
UxADMD (UARTx 地址匹配检测) .....	262	框图 .....	
USB OTG .....	54	10 位 A/D 转换器模拟输入模型 .....	368
外设模块禁止 (PMD) .....	60	12 位 A/D 转换器 .....	352
外设引脚选择 (PPS) .....	57	16 位 Timer1 模块 .....	201
系统控制 (时钟和复位) .....	58	16 位异步 Timer3 和 Timer5 .....	207
引脚配置 (PADCFG1) .....	50	比较器参考电压模块 .....	377
		表寄存器寻址 .....	79
		CALL 堆栈帧 .....	64
		CPU 编程模型 .....	31
		CRC 模块 .....	345
		CRC 移位引擎详细信息 .....	345
		CREF = 0 时各个比较器配置 .....	372
		产生脉冲延时的 CTMU 典型连接和内部配置 .....	381
		电容测量的 CTMU 连接和内部配置 .....	380

读操作的 EDS 地址生成 .....	62
端点缓冲模式下的 BDT 映射 .....	270
访问程序空间内数据的地址生成方式 .....	66
复位系统 .....	85
高 / 低压检测 (HLVD) .....	387
各个比较器配置, CREF = 1, CVREFF = 0 .....	373
各个比较器配置, CREF = 1, CVREFF = 1 .....	373
各个比较器配置 (CREF = 0) .....	372
共用 I/O 端口的结构 .....	173
I <sup>2</sup> C 模块 .....	246
加密引擎 .....	329
仅 USB OTG 自供电 .....	267
仅 USB OTG 总线电源 .....	267
仅总线电源 .....	267
看门狗定时器 (WDT) .....	400
扩展数据空间 (EDS) .....	61
PIA 模式下的缓冲区地址生成 .....	355
PIC24F CPU 内核 .....	30
PIC24FJ128GB204 系列 (通用) .....	14
PSV 操作访问 (低字位) .....	69
PSV 操作访问 (高位字) .....	69
片内稳压器的连接 .....	399
全速设备模式下的 USB OTG 外部上拉 .....	267
全速设备模式下的外部上拉 .....	267
RTCC 模块 .....	315
SPIx 从器件、帧从器件连接 .....	243
SPIx 从器件、帧主器件连接 .....	243
SPIx 模块 (标准模式) .....	228
SPIx 模块 (增强型模式) .....	229
SPIx 主 / 从连接 (标准模式) .....	241
SPIx 主 / 从连接 (增强型缓冲模式) .....	242
SPIx 主器件、帧从器件连接 .....	243
SPIx 主器件、帧主器件连接 .....	242
三比较器模块 .....	371
时间测量的 CTMU 典型连接 和内部配置 .....	381
使用表指令访问程序空间 .....	67
输出比较 (16 位模式) .....	218
输出比较 (双缓冲, 16 位 PWM 模式) .....	220
数据信号调制器 .....	299
输入捕捉 .....	211
Timer2/3 和 Timer4/5 (32 位) .....	206
UARTx (简化) .....	254
USB OTG 接口示例 .....	268
USB OTG 模块 .....	266
USB OTG 双电源示例 .....	267
USB OTG 中断逻辑 .....	274
USB OTG 主机接口示例 .....	268
USB PLL .....	155
系统时钟 .....	147
写操作的 EDS 地址生成 .....	63
直接存储器访问 (DMA) .....	71
智能卡子系统连接 .....	257
扩展数据空间 (EDS) .....	303

## M

Microchip 因特网网站 .....	473
MPLAB ASM30 汇编器、链接器和库管理器 .....	404
MPLAB PM3 器件编程器 .....	405
MPLAB REAL ICE 在线仿真器系统 .....	405
MPLAB 集成开发环境软件 .....	403
MPLINK 目标链接器 /MPLIB 目标库管理器 .....	404
脉宽调制。见 PWM。	
脉宽调制 (PWM) 模式 .....	219

## N

Near 数据空间 .....	38
内部集成电路。请参见 I <sup>2</sup> C .....	245
内核特性 .....	9

## P

PWM	
占空比和周期 .....	220
配置位 .....	389
片上稳压器 .....	399
待机模式 .....	399
POR .....	399

## Q

器件特性	
28 引脚器件 .....	13
44 引脚器件 .....	12
器件引脚配置说明 .....	15

## R

RTCC	
电源控制 .....	327
寄存器映射 .....	316
模块寄存器 .....	316
闹钟配置 .....	326
闹钟屏蔽设置 (图) .....	327
时钟源选择 .....	316
VBAT 操作 .....	327
校准 .....	326
写锁定 .....	316
源时钟 .....	315
软件堆栈 .....	64
软件模拟器 (MPLAB SIM) .....	405

## S

SFR 空间 .....	38
SPI .....	227
控制寄存器 .....	230
增强型主模式 .....	229
标准从模式 .....	228
三比较器 .....	371
三比较器模块 .....	371
闪存程序存储器 .....	79
编程操作 .....	80
编程单字 .....	84
编程算法 .....	82
和表指令 .....	79
JTAG 操作 .....	80
RTSP 操作 .....	80
增强型 ICSP 操作 .....	80
闪存配置字 .....	36
闪存配置字位置 .....	389
实时时钟和日历	

# PIC24FJ128GB204 系列

(RTCC) .....	315	UxCTS 和 UxRTS 控制	
时序图		引脚的操作 .....	256
CLKO 和 I/O 特性 .....	430	智能卡 ISO 7816 支持 .....	257
外部时钟 .....	428	USB On-The-Go (OTG) .....	10
I <sup>2</sup> C 总线数据 (主模式) .....	442	USB OTG .....	265
I <sup>2</sup> C 总线数据 (从模式) .....	444	操作 .....	278
I <sup>2</sup> C 总线启动 / 停止位		会话请求协议	
(主模式) .....	441	(SRP) .....	278
I <sup>2</sup> C 总线启动 / 停止位		DMA 接口 .....	271
(从模式) .....	443	缓冲区描述符和 BDT .....	270
输入捕捉 x (ICx) .....	434	寄存器 .....	279
OCx/PWM .....	435	设备模式的操作 .....	275
SPIx 主模式 (CKE = 0) .....	436	Vbus 电压生成 .....	269
SPIx 主模式 (CKE = 1) .....	437	USB 主机或目标设备以	
SPIx 从模式 (CKE = 0) .....	438	控制器为中心的数据方向 .....	265
SPIx 从模式 (CKE = 1) .....	439	硬件	
Timer1, 2, 3, 4, 5 外部时钟 .....	432	计算	
使用专用定时器的输出比较 .....	217	收发器功耗需求 .....	269
使用专用定时器的输入捕捉 .....	211	硬件配置 .....	267
输出比较		电源模式	
32 位级联模式 .....	217	仅自供电 .....	267
同步和触发模式 .....	217	仅总线电源 .....	267
数据存储		双电源且自供电为主 .....	267
存储器映射 .....	37	设备模式 .....	267
地址空间 .....	37	Vbus 电压生成 .....	269
空间构成, 对齐方式 .....	38	主机和 OTG 模式 .....	268
扩展数据空间		中断 .....	274
(EDS) .....	61	主机模式的操作 .....	276
Near 数据空间 .....	38	WWW 地址 .....	473
软件堆栈 .....	64	外设引脚选择 (PPS) .....	178
SFR 空间 .....	38	寄存器 .....	183
数据信号调制器 .....	299	可用外设和引脚 .....	178
数据信号调制器		配置控制 .....	181
(DSM) .....	299	使用注意事项 .....	182
输入捕捉		输出映射 .....	180
32 位级联模式 .....	212	输入映射 .....	179
操作 .....	212	外设优先级 .....	178
同步和触发模式 .....	211	映射例外 .....	181
<b>T</b>		<b>X</b>	
Timer2/3 和 Timer4/5 .....	205	选择性外设模块控制 .....	171
Timer1 .....	201	<b>Y</b>	
特性 .....	10	因特网地址 .....	473
通用串行总线		<b>Z</b>	
缓冲区描述符		增强型并行主端口。见 EPMP。 .....	303
在不同缓冲模式下的分配 .....	271	增强型并行主端口 (EPMP) .....	303
中断		振荡器	
和 USB 事务 .....	275	参考时钟输出 .....	156
通用串行总线。见 USB OTG。		初始 CPU 时钟机制 .....	148
通用异步收发器。见 UART。		FRC 自调节 .....	154
<b>W</b>		片内 PLL .....	159
UART .....	253	上电复位时的初始配置 .....	148
波特率发生器 (BRG) .....	255	时钟切换 .....	153
发送		序列 .....	153
8 位数据模式 .....	256	USB 工作 .....	155
9 位数据模式 .....	256	特殊注意事项 .....	156
间隔或同步序列 .....	256	用于时钟选择的配置位值 .....	148
红外支持 .....	256	直接存储器访问控制器。参见 DMA。	
接收		指令集	
8 位或 9 位数据模式 .....	256	操作码说明中使用的符号 .....	408
控制寄存器 .....	258		

# PIC24FJ128GB204 系列

概述 .....	409	中断	
汇总 .....	407	复位过程 .....	91
直流特性		控制和状态寄存器 .....	95
比较器规范 .....	425	设置和服务过程 .....	145
CTMU 电流源规范 .....	426	实现的向量 .....	93
程序存储器 .....	423	陷阱向量 .....	92
$\Delta$ 电流 (BOR、WDT、DSBOR、DSWDT) .....	421	向量表 .....	92
掉电电流 (IPD) .....	420	中断向量表 (IVT) .....	91
封装热阻 .....	416	主要特性 .....	389
高 / 低压检测 .....	424		
工作电流 (IDD) .....	418		
I/O 引脚输出规范 .....	423		
I/O 引脚输入规范 .....	422		
空闲电流 (IIDL) .....	419		
内部稳压器规范 .....	424		
热工作条件 .....	416		
Vbat 操作电压规范 .....	426		
USB On-The-Go 模块规范 .....	426		
温度和电压规范 .....	417		

# PIC24FJ128GB204 系列

---

注:

## MICROCHIP 网站

Microchip 网站 ([www.microchip.com](http://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。只要使用常用的互联网浏览器即可访问。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归档软件
- **一般技术支持**——常见问题解答 (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](http://www.microchip.com)。在“支持” (Support) 下，点击“变更通知客户 (Customer Change Notification)” 服务后按照注册说明完成注册。

## 客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

客户应联系其代理商、代表或应用工程师 (FAE) 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 <http://microchip.com/support> 获得网上技术支持。

# PIC24FJ128GB204 系列

---

注:

## 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。

PIC 24 FJ 128 GB2 04 I - I / PT - XXX	
Microchip 商标	_____
架构	_____
闪存系列	_____
程序存储器大小 (KB)	_____
产品组	_____
引脚数	_____
卷带标志 (如果适用)	_____
温度范围	_____
封装	_____
定制编号	_____

架构	24 = 不带 DSP 的 16 位改进型哈佛架构
闪存系列	FJ = 闪存程序存储器
产品组	GB2 = 带 USB On-The-Go (OTG) 的通用单片机
引脚数	02 = 28 引脚 04 = 44 引脚
温度范围	I = -40°C 至 +85°C (工业级) E = -40°C 至 +125°C (扩展级)
封装	MM= 28 引脚 (6x6x0.9 mm) QFN-S (塑封四方扁平封装) ML = 44 引脚 (8x8 mm) QFN (塑封四方扁平封装) PT = 44 引脚 (10x10x1 mm) TQFP (薄型正方扁平封装) SO = 28 引脚 (7.50 mm) SOIC (宽条小外形封装) SP = 28 引脚 (300 mil) SPDIP (窄条塑封双列直插式封装) SS = 28 引脚 (5.30 mm) SSOP (塑封缩小外形封装)
定制编号	3 位 QTP、SQTP、编码或特殊要求 (否则为空白) ES = 工程样片

**示例:**

a) PIC24FJ128GB202-I/MM  
带 USB On-The-Go 的 PIC24F 器件、128 KB 程序存储器、8 KB 数据存储器、28 引脚、工业级温度、QFN-S 封装。

b) PIC24FJ128GB204-I/PT:  
带 USB On-The-Go 的 PIC24F 器件、128 KB 程序存储器、8 KB 数据存储器、44 引脚、工业级温度、TQFP 封装。

# PIC24FJ128GB204 系列

---

注:

---

请注意以下有关 Microchip 器件代码保护功能的要点：

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

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

---

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

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。在 Microchip 知识产权保护下，不得暗中以其他方式转让任何许可证。除非另外声明，在 Microchip 知识产权保护下，不得暗中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、FlashFlex、flexPWR、JukeBlox、KEELOQ、KEELOQ 徽标、Kleer、LANCheck、MediaLB、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、PICSTART、PIC<sup>32</sup> 徽标、RightTouch、SpyNIC、SST、SST 徽标、SuperFlash 及 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

The Embedded Control Solutions Company 和 mTouch 为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、ECAN、In-Circuit Serial Programming、ICSP、Inter-Chip Connectivity、KleerNet、KleerNet 徽标、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、RightTouch 徽标、REAL ICE、SQI、Serial Quad I/O、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

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

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2013-2015, Microchip Technology Inc. 版权所有。

ISBN: 978-1-63277-560-3

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949 ==**

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC<sup>®</sup> MCU 与 dsPIC<sup>®</sup> DSC、KEELOQ<sup>®</sup> 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

## 全球销售及及服务网点

### 美洲

公司总部 **Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 1-480-792-7200  
Fax: 1-480-792-7277

技术支持:  
<http://www.microchip.com/support>

网址: [www.microchip.com](http://www.microchip.com)

**亚特兰大 Atlanta**  
Duluth, GA  
Tel: 1-678-957-9614  
Fax: 1-678-957-1455

**奥斯汀 Austin, TX**  
Tel: 1-512-257-3370

**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**  
Novi, MI  
Tel: 1-248-848-4000

**休斯敦 Houston, TX**  
Tel: 1-281-894-5983

**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

**纽约 New York, NY**  
Tel: 1-631-435-6000

**圣何塞 San Jose, CA**  
Tel: 1-408-735-9110

**加拿大多伦多 Toronto**  
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: 852-2943-5100  
Fax: 852-2401-3431

**中国 - 北京**  
Tel: 86-10-8569-7000  
Fax: 86-10-8528-2104

**中国 - 成都**  
Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

**中国 - 重庆**  
Tel: 86-23-8980-9588  
Fax: 86-23-8980-9500

**中国 - 东莞**  
Tel: 86-769-8702-9880

**中国 - 杭州**  
Tel: 86-571-8792-8115  
Fax: 86-571-8792-8116

**中国 - 香港特别行政区**  
Tel: 852-2943-5100  
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: 86-755-8864-2200  
Fax: 86-755-8203-1760

**中国 - 武汉**  
Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

**中国 - 西安**  
Tel: 86-29-8833-7252  
Fax: 86-29-8833-7256

**中国 - 厦门**  
Tel: 86-592-238-8138  
Fax: 86-592-238-8130

**中国 - 珠海**  
Tel: 86-756-321-0040  
Fax: 86-756-321-0049

### 亚太地区

**台湾地区 - 高雄**  
Tel: 886-7-213-7828

**台湾地区 - 台北**  
Tel: 886-2-2508-8600  
Fax: 886-2-2508-0102

**台湾地区 - 新竹**  
Tel: 886-3-5778-3666  
Fax: 886-3-5770-955

**澳大利亚 Australia - Sydney**  
Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

**印度 India - Bangalore**  
Tel: 91-80-3090-4444  
Fax: 91-80-3090-4123

**印度 India - New Delhi**  
Tel: 91-11-4160-8631  
Fax: 91-11-4160-8632

**印度 India - Pune**  
Tel: 91-20-3019-1500

**日本 Japan - Osaka**  
Tel: 81-6-6152-7160  
Fax: 81-6-6152-9310

**日本 Japan - Tokyo**  
Tel: 81-3-6880-3770  
Fax: 81-3-6880-3771

**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 - Dusseldorf**  
Tel: 49-2129-3766400

**德国 Germany - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

**德国 Germany - Pforzheim**  
Tel: 49-7231-424750

**意大利 Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**意大利 Italy - Venice**  
Tel: 39-049-7625286

**荷兰 Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**波兰 Poland - Warsaw**  
Tel: 48-22-3325737

**西班牙 Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

**瑞典 Sweden - Stockholm**  
Tel: 46-8-5090-4654

**英国 UK - Wokingham**  
Tel: 44-118-921-5800  
Fax: 44-118-921-5820