

深圳市粤原点科技有限公司 ( 丽智电子:www.LZmcu.com )  
(Microchip Authorized Design Partner)指定授权  
总部地址：深圳市福田区福虹路世贸广场C座1103室  
Add：Room 1103,Block C,World Trade Plaza,  
Fuhong Road,Futian District Shen Zhen City  
电话(tel)：86-755-83666320,83685176,83666321,83666325  
传真(fax)：86-755-83666329,83681854  
Web: Http://www.origin-gd.com or Http://www.LZmcu.com  
E-mail：01@LZmcu.com  
联系人：马先生,王小姐  
公司在线咨询：QQ:46885145  
MSN:MSN:action\_tech@hotmail.com  
7x24小时在线产品咨询: 13509674380 13798484366



---

# **dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 数据手册**

高性能  
16 位数字信号控制器

---

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

---

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

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

#### 商标

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

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

Analog-for-the-Digital Age、Application Maestro、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、In-Circuit Serial Programming、ICSP、ICEPIC、Mindi、MiWi、MPASM、MPLAB Certified 徽标、MPLIB、MPLINK、mTouch、PICkit、PICDEM、PICDEM.net、PICtail、PIC<sup>32</sup> 徽标、PowerCal、PowerInfo、PowerMate、PowerTool、REAL ICE、rLAB、Select Mode、Total Endurance、UNI/O、WiperLock 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

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

在此提及的所有其他商标均为各持有公司所有。

© 2008, Microchip Technology Inc. 版权所有。

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

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2002 认证。公司在 PIC<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方面的质量体系流程均符合 ISO/TS-16949:2002。此外, Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

## 高性能 16 位数字信号控制器

### 工作范围:

- 最高 40 MIPS 的工作速度 (3.0-3.6V 时):
  - 工业级温度范围 (-40°C 至 +85°C)
  - 扩展级温度范围 (-40°C 至 +125°C)

### 高性能 DSC CPU:

- 改进型哈佛架构
- C 编译器优化指令集
- 16 位宽数据总线
- 24 位宽指令
- 可寻址最大 4M 指令字的线性程序存储空间
- 可寻址最大 64 KB 的线性数据存储空间
- 83 条基本指令: 多为单字 / 单周期指令
- 两个带舍入和饱和选项的 40 位累加器
- 灵活而强大的寻址模式:
  - 间接寻址
  - 模寻址
  - 位反转寻址
- 软件堆栈
- 16 x 16 位小数 / 整数乘法运算
- 32/16 位和 16/16 位除法运算
- 单周期乘 - 累加运算:
  - DSP 运算的累加器回写操作
  - 双数据取操作
- 可将最多 40 位的数据左移或右移最多 16 位

### 直接存储器访问 (Direct Memory Access, DMA):

- 8 通道硬件 DMA
- 最大 2 KB 的双端口 DMA 缓冲区 (DMA RAM) 用于存储通过 DMA 传输的数据:
  - 允许 CPU 执行代码时在 RAM 和外设之间传输数据 (不占用周期)
- 大部分外设都支持 DMA

### 定时器 / 捕捉 / 比较 / PWM:

- 定时器 / 计数器, 最多 5 个 16 位定时器
  - 可以配对作为最多 2 个 32 位定时器使用
  - 1 个定时器依靠外部 32.768 kHz 振荡器作为实时时钟运行
  - 可编程预分频器
- 输入捕捉 (最多 4 路通道):
  - 上升沿捕捉、下降沿捕捉或上升 / 下降沿捕捉
  - 16 位捕捉输入功能
  - 每路捕捉通道都带有 4 字深度的 FIFO 缓冲区
- 输出比较 (最多 4 路通道):
  - 1 个或 2 个 16 位比较模式
  - 16 位无毛刺 PWM 模式
- 硬件实时时钟 / 日历 (Real-Time Clock/Calendar, RTCC):
  - 提供时钟、日历和闹钟功能

### 中断控制器:

- 中断响应延时为 5 个周期
- 118 个中断向量
- 最多 53 个中断源
- 最多 3 个外部中断
- 7 个可编程优先级
- 5 个处理器异常

### 数字 I/O:

- 外设引脚选择功能
- 最多 35 个可编程数字 I/O 引脚
- 最多 21 个引脚上具有唤醒 / 电平变化中断功能
- 输出引脚可驱动 3.0V 至 3.6V 的电压
- 漏极开路配置、最高 5V 的输出
- 所有数字输入引脚可承受 5V 的电压
- 所有 I/O 引脚的最大拉 / 灌电流为 4 mA

### 片上闪存和 SRAM:

- 闪存程序存储器 (最大 128 KB)
- 数据 SRAM (最大 16 KB)
- 闪存程序存储器的引导、安全和通用安全性

### 系统管理：

- 灵活的时钟选择：
  - 外部振荡器、晶振、谐振器和内部 RC 振荡器
  - 全集成锁相环（Phase-Locked Loop, PLL）
  - 极低抖动 PLL
- 上电延时定时器
- 振荡器起振定时器 / 稳定器
- 自带 RC 振荡器的看门狗定时器
- 故障保护时钟监视器
- 多个复位源

### 功耗管理：

- 片上 2.5V 稳压器
- 实时时钟源切换
- 可快速唤醒的空闲、休眠和打盹模式

### 模数转换器（Analog-to-Digital Converter, ADC）：

- 10 位 1.1 Msps 或 12 位 500 ksps 转换：
  - 2 路或 4 路输入同时采样（10 位 ADC）
  - 最多 9 路带自动扫描功能的输入通道
  - 可手动启动转换或与 4 个触发源中的一个同步
  - 休眠模式下仍可进行转换
  - 最大积分非线性误差为  $\pm 2$  LSB
  - 最大微分非线性误差为  $\pm 1$  LSB

### 音频数模转换器（Digital-to-Analog Converter, DAC）：

- 16 位双通道 DAC 模块
- 最大 100 ksps 采样速率
- 二阶数字  $\Delta$ - $\Sigma$  调制器

### 比较器模块：

- 带可编程输入 / 输出配置的两个模拟比较器

### CMOS 闪存技术：

- 低功耗高速闪存技术
- 全静态设计
- 3.3V ( $\pm 10\%$ ) 工作电压
- 工业级和扩展级温度
- 低功耗

### 电机控制外设：

- 6 通道 16 位电机控制 PWM
  - 3 个占空比发生器
  - 独立或互补模式
  - 可编程死区和输出极性
  - 边沿对齐或中心对齐
  - 手动输出改写控制
  - 1 路故障输入
  - ADC 转换触发
  - 16 位分辨率时的 PWM 频率 (@ 40 MIPS)：边沿对齐模式为 1220 Hz，中心对齐模式为 610 Hz
  - 11 位分辨率时的 PWM 频率 (@ 40 MIPS)：边沿对齐模式为 39.1 kHz，中心对齐模式为 19.55 kHz
- 2 通道 16 位电机控制 PWM
  - 1 个占空比发生器
  - 独立或互补模式
  - 可编程死区和输出极性
  - 边沿对齐或中心对齐
  - 手动输出改写控制
  - 1 路故障输入
  - ADC 转换触发
  - 16 位分辨率时的 PWM 频率 (@ 40 MIPS)：边沿对齐模式为 1220 Hz，中心对齐模式为 610 Hz
  - 11 位分辨率时的 PWM 频率 (@ 40 MIPS)：边沿对齐模式为 39.1 kHz，中心对齐模式为 19.55 kHz
- 双正交编码器接口模块：
  - A 相、B 相和索引脉冲输入
  - 16 位递增 / 递减位置计数器
  - 计数方向状态
  - 位置测量 (x2 和 x4) 模式
  - 输入端上的可编程数字噪声滤波器
  - 备用 16 位定时器 / 计数器模式
  - 位置计数器计满返回 / 下溢中断

## 通信模块:

- 4 线 SPI（最多 2 个模块）：
  - 帧支持与简单编解码器的 I/O 接口
  - 支持 8 位和 16 位数据
  - 支持所有串行时钟格式和采样模式
- I<sup>2</sup>C™：
  - 完全支持多主机从模式
  - 7 位和 10 位寻址
  - 总线冲突检测和仲裁
  - 集成信号调理
  - 从地址掩码
- UART（最多 2 个模块）：
  - 检测到地址位时产生中断
  - 出现 UART 错误时产生中断
  - 检测到启动位时将器件从休眠模式唤醒
  - 4 字符深的发送和接收 FIFO 缓冲区
  - LIN 总线支持
  - 硬件 IrDA® 编解码器
  - 高速波特率模式
  - 使用 CTS 和 RTS 的硬件流控制
- 增强型 CAN（ECAN™ 模块） 2.0B active：
  - 最多 8 个发送缓冲区、最多 32 个接收缓冲区
  - 16 个接收过滤器和 3 个屏蔽器
  - 环回、监听和监听所有报文模式
  - 用于诊断和总线监视的报文模式
  - 接收到 CAN 报文时唤醒
  - 远程发送请求的自动处理
  - 使用 DMA 的 FIFO 模式
  - 支持 DeviceNet™ 寻址模式
- 并行主从端口（PMP/EPSP）：
  - 支持 8 位或 16 位数据
  - 支持 16 根地址线
- 可编程循环冗余校验（Cyclic Redundancy Check, CRC）
  - CRC 发生器多项式的可编程位长度（最大 16 位长度）
  - 用于数据输入的 8 级深 16 位或 16 级深 8 位 FIFO

## 封装:

- 28 引脚 SPDIP/SOIC/QFN-S
- 44 引脚 TQFP/QFN

**注:** 关于每个器件的具体外设特性，请参见相应的器件数据表。

# dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04

## dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 产品系列

下表列出了每个器件的器件名、引脚数、存储容量和可用的外设。表后各页还附有它们的引脚图。

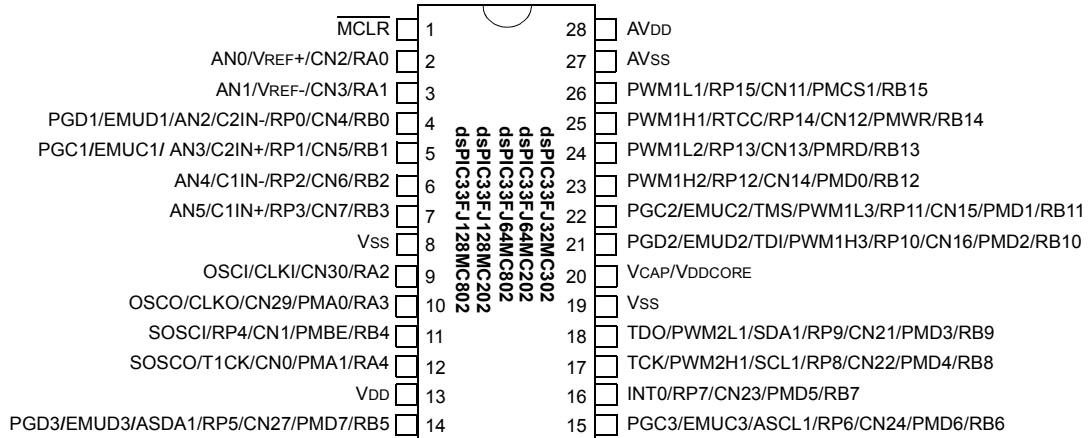
### dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 控制器系列

器件	引脚数	闪存程序存储器 (KB)	RAM (KB) (1)	可重映射的外设								RTCC	I <sup>2</sup> C™	CRC 发生器	10 位/12 位 ADC (通道数)	6 引脚 16 位 DAC	模拟比较器 (2 通道/稳压器)	8 位并行主端口 (地址线)	I/O 引脚	封装	
				可重映射的引脚数	16 位定时器 (2)	输入捕捉	输出比较标准 PWM	电机控制 PWM (通道) (3)	正交编码器接口	UART	SPI										ECAN™
dsPIC33FJ128MC804	44	128	16	26	5	4	4	6, 2	2	2	2	1	1	1	1	9	1	1/1	11	35	QFN TQFP
dsPIC33FJ128MC802	28	128	16	16	5	4	4	6, 2	2	2	2	1	1	1	1	6	0	1/0	2	21	SPDIP SOIC QFN-S
dsPIC33FJ128MC204	44	128	8	26	5	4	4	6, 2	2	2	2	0	1	1	1	9	0	1/1	11	35	QFN TQFP
dsPIC33FJ128MC202	28	128	8	16	5	4	4	6, 2	2	2	2	0	1	1	1	6	0	1/0	2	21	SPDIP SOIC QFN-S
dsPIC33FJ64MC804	44	64	16	26	5	4	4	6, 2	2	2	2	1	1	1	1	9	1	1/1	11	35	QFN TQFP
dsPIC33FJ64MC802	28	64	16	16	5	4	4	6, 2	2	2	2	1	1	1	1	6	0	1/0	2	21	SPDIP SOIC QFN-S
dsPIC33FJ64MC204	44	64	8	26	5	4	4	6, 2	2	2	2	0	1	1	1	9	0	1/1	11	35	QFN TQFP
dsPIC33FJ64MC202	28	64	8	16	5	4	4	6, 2	2	2	2	0	1	1	1	6	0	1/0	2	21	SPDIP SOIC QFN-S
dsPIC33FJ32MC304	44	32	4	26	5	4	4	6, 2	2	2	2	0	1	1	1	9	0	1/1	11	35	QFN TQFP
dsPIC33FJ32MC302	28	32	4	16	5	4	4	6, 2	2	2	2	0	1	1	1	6	0	1/0	2	21	SPDIP SOIC QFN-S

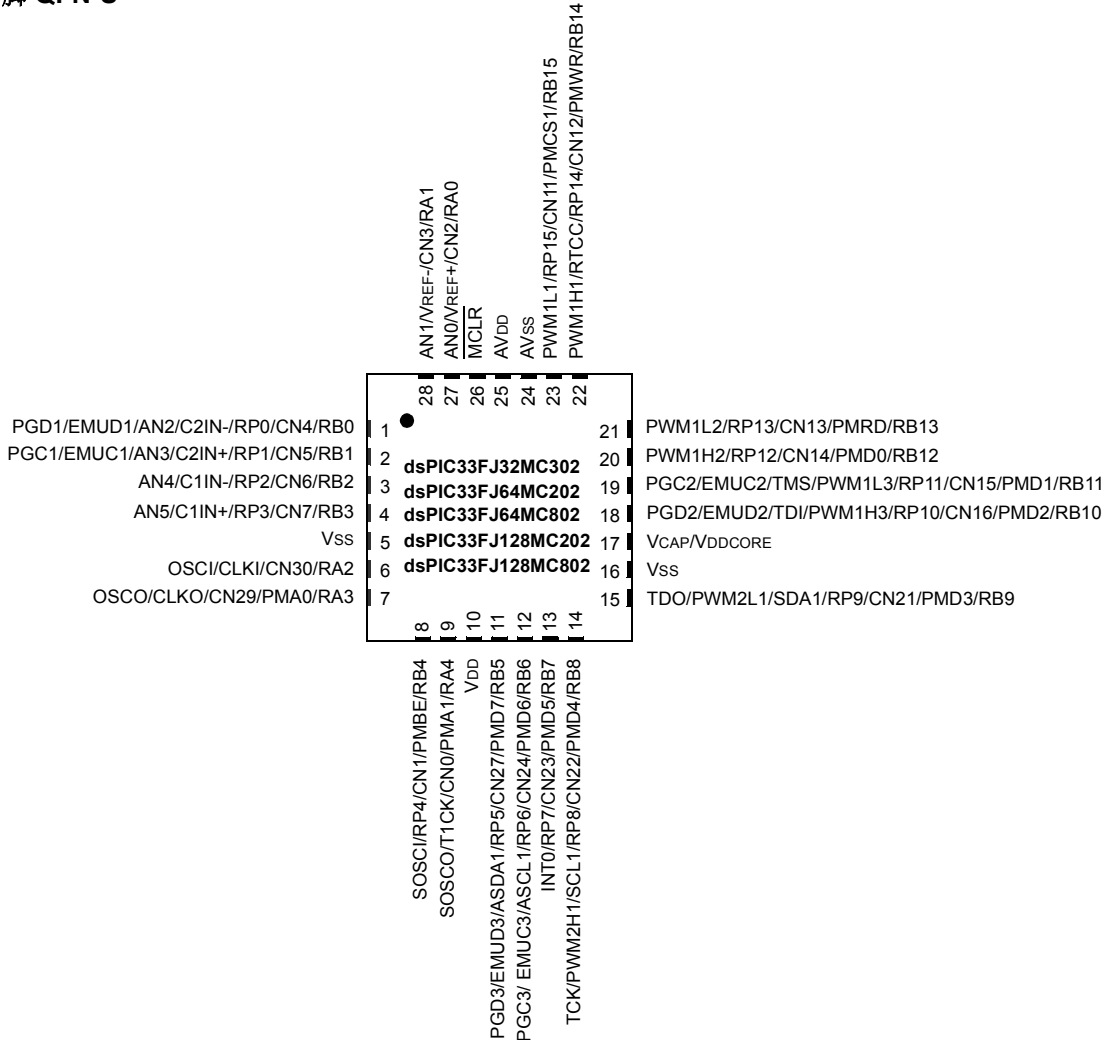
- 注 1: 所有器件的 RAM 大小包括 2 KB 的 DMA RAM, dsPIC33FJ32MC302/304 除外, 它包含 1 KB 的 DMA RAM。  
 2: 5 个定时器中只有 4 个是可重映射的。  
 3: 只有 PWM 故障引脚是可重映射的。

引脚图

28 引脚 SPDIP 和 SOIC

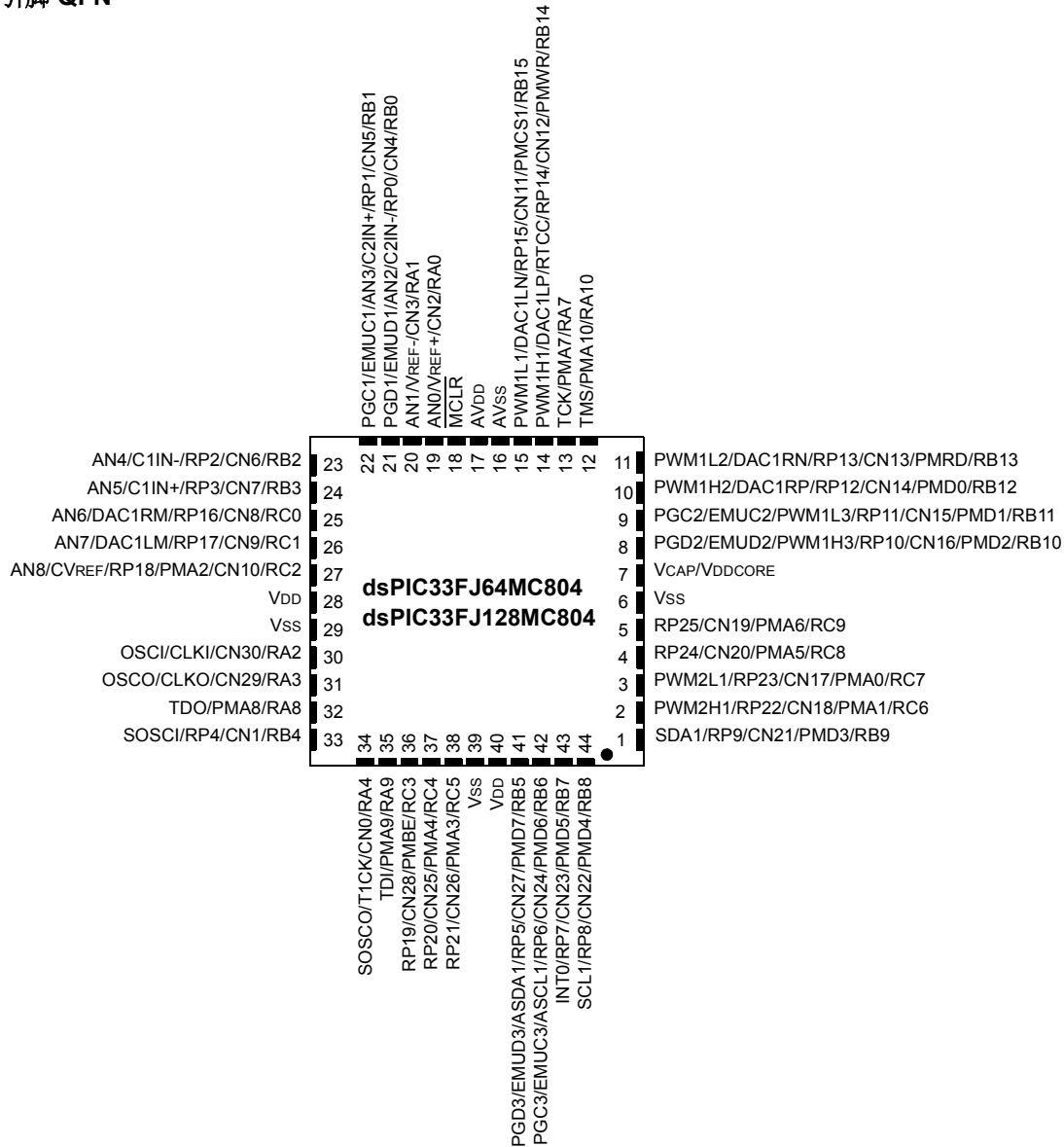


28 引脚 QFN-S



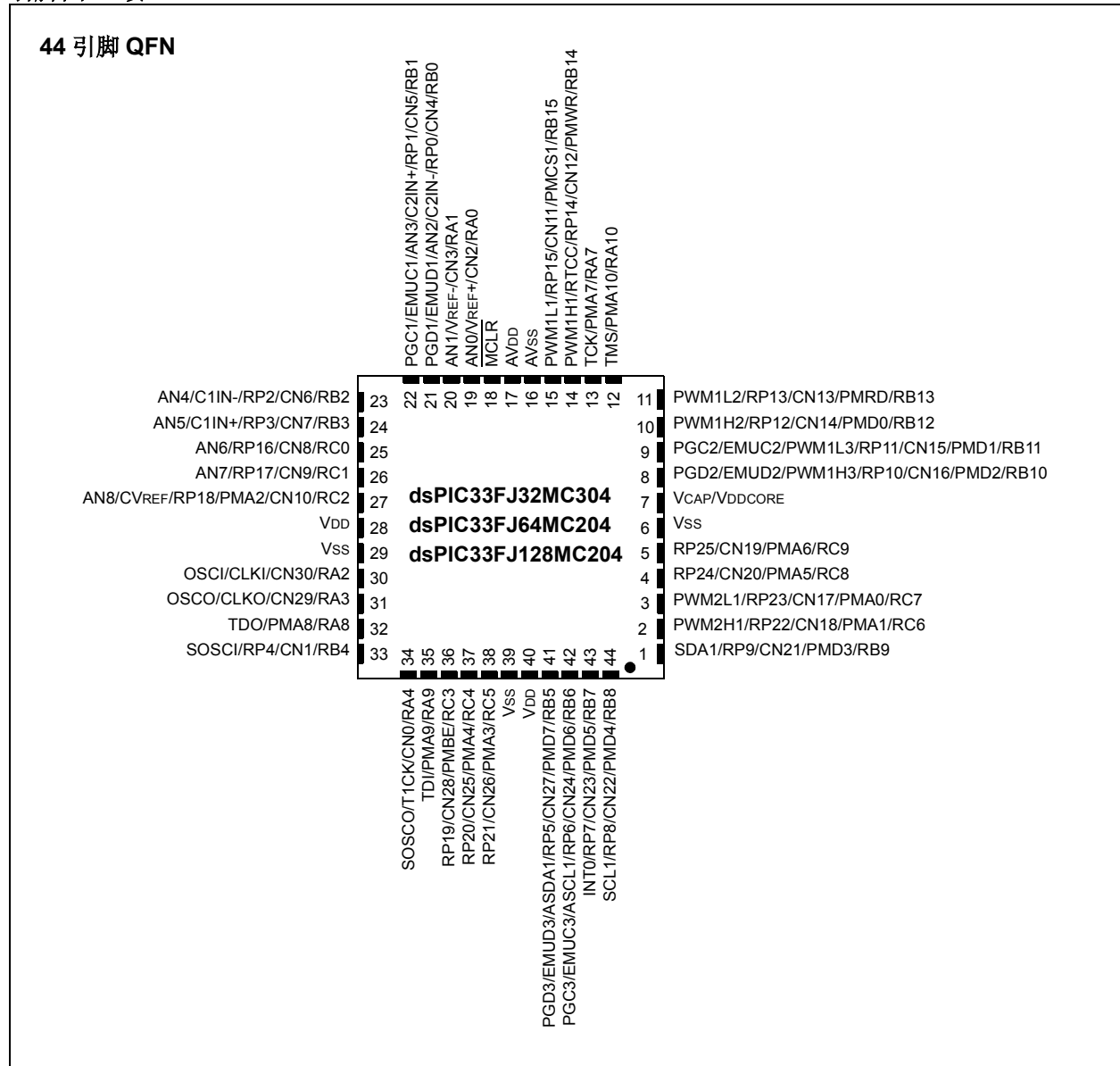
引脚图 (续)

44 引脚 QFN

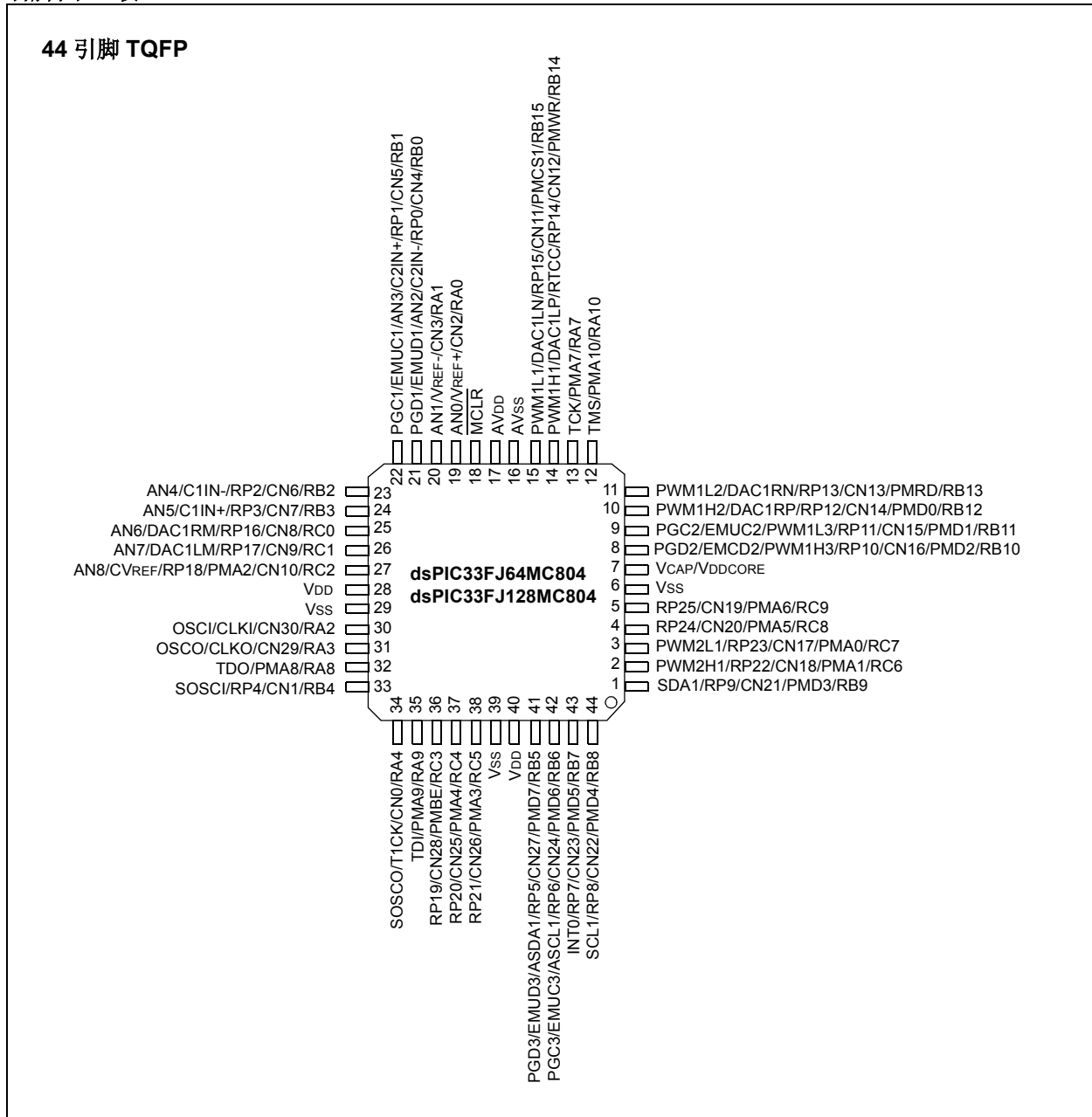




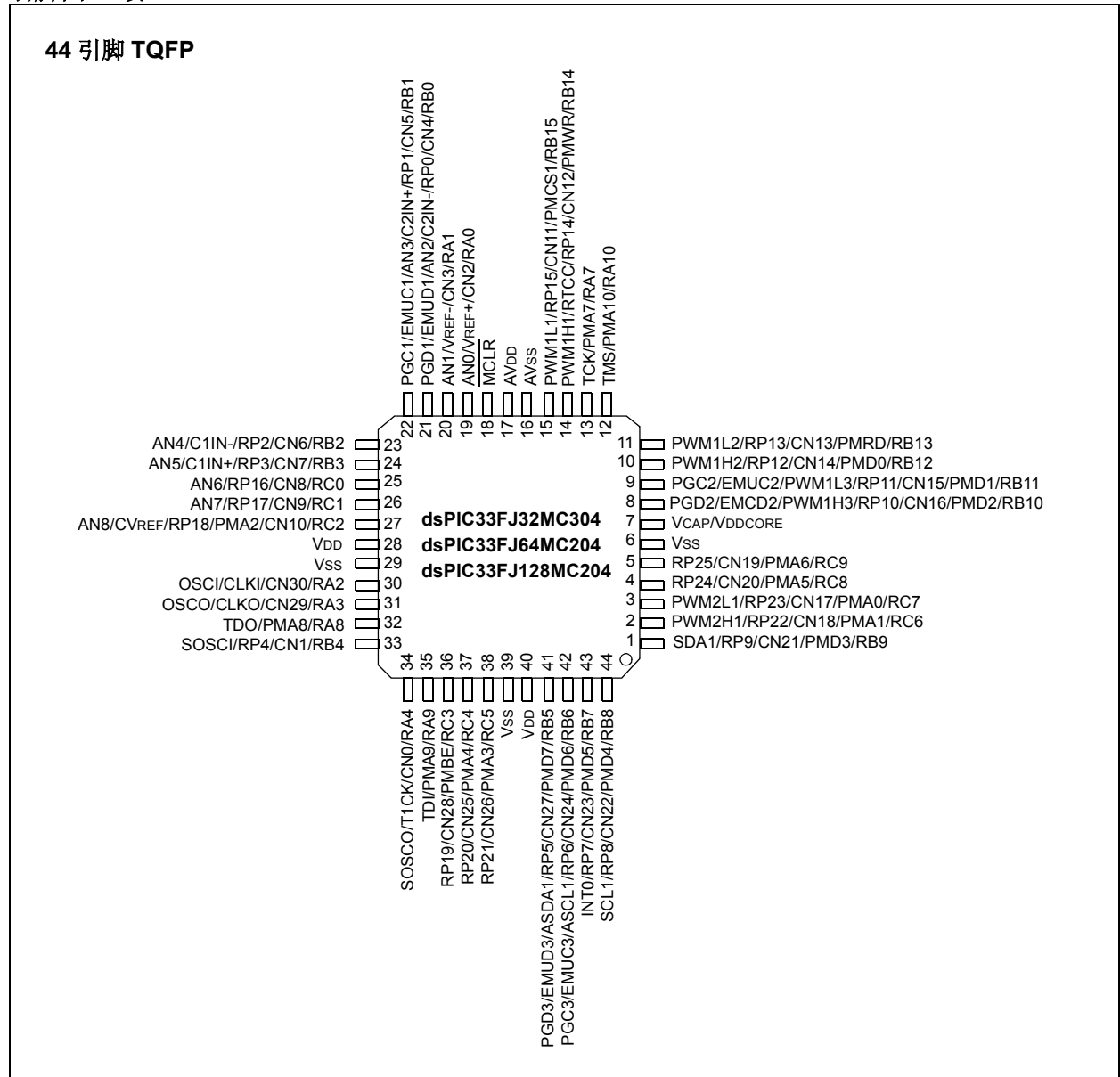
引脚图 (续)



引脚图 (续)



引脚图 (续)



**目录**

dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 产品系列 .....	4
1.0 器件概述 .....	13
2.0 CPU .....	19
3.0 存储器构成 .....	31
4.0 闪存程序存储器 .....	67
5.0 复位 .....	73
6.0 中断控制器 .....	81
7.0 直接存储器访问 (DMA) .....	123
8.0 振荡器配置 .....	135
9.0 节能特性 .....	147
10.0 I/O 端口 .....	149
11.0 Timer2/3 和 Timer4/5 特性 .....	181
12.0 Timer1 .....	187
13.0 输入捕捉 .....	189
14.0 输出比较 .....	191
15.0 电机控制 PWM 模块 .....	195
16.0 正交编码器接口 (QE1) 模块 .....	217
17.0 串行外设接口 (SPI) .....	225
18.0 I <sup>2</sup> C™ .....	233
19.0 通用异步收发器 (UART) .....	243
20.0 增强型 CAN (ECAN™) 模块 .....	251
21.0 10 位/12 位模数转换器 (ADC1) .....	281
22.0 音频数模转换器 (DAC) .....	295
23.0 比较器模块 .....	301
24.0 实时时钟和日历 (RTCC) .....	307
25.0 可编程循环冗余校验 (CRC) 发生器 .....	319
26.0 并行主端口 (PMP) .....	323
27.0 特殊功能 .....	333
28.0 指令集汇总 .....	343
29.0 开发支持 .....	351
30.0 电气特性 .....	355
31.0 封装信息 .....	399
Microchip 网站 .....	413
变更通知客户服务 .....	413
客户支持 .....	413
读者反馈表 .....	414
产品标识体系 .....	415

## 致客户

我们旨在提供最佳文档供客户正确使用 Microchip 产品。为此，我们将不断改进出版物的内容和质量，使之更好地满足您的要求。出版物的质量将随新文档及更新版本的推出而得到提升。

如果您对本出版物有任何问题和建议，请通过电子邮件联系我公司 TRC 经理，电子邮件地址为 [CTRC@microchip.com](mailto:CTRC@microchip.com)，或将本数据手册后附的《读者反馈表》传真到 86-21-5407 5066。我们期待您的反馈。

### 最新数据手册

欲获得本数据手册的最新版本，请查询我公司的网站：

<http://www.microchip.com>

查看数据手册中任意一页下边角处的文献编号即可确定其版本。文献编号中数字串后的字母是版本号，例如：DS30000A是DS30000的A版本。

### 勘误表

现有器件可能带有一份勘误表，描述了实际运行与数据手册中记载内容之间存在的细微差异以及建议的变通方法。一旦我们了解到器件 / 文档存在某些差异时，就会发布勘误表。勘误表上将注明其所适用的硅片版本和文件版本。

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

- Microchip 网站， <http://www.microchip.com>
- 当地 Microchip 销售办事处（见最后一页）

在联络销售办事处时，请说明您所使用的器件型号、硅片版本和数据手册版本（包括文献编号）。

### 客户通知系统

欲及时获知 Microchip 产品的最新信息，请到我公司网站 [www.microchip.com](http://www.microchip.com) 上注册。

注:

## 1.0 器件概述

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“*dsPIC33F Family Reference Manual*”的相关章节，该文档可从 [Microchip 网站 \(www.microchip.com\)](http://www.microchip.com) 下载。

本文档包含 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 数字信号控制器 (Digital Signal Controller, DSC) 器件的特定信息。dsPIC33F 器件在高性能 16 位单片机 (MCU) 架构中，融合了丰富的数字信号处理器 (Digital Signal Processor, DSP) 功能。

图 1-1 给出了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件内核和外设模块的通用框图。表 1-1 列出了引脚图中所示各引脚的功能。

图 1-1: dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 框图

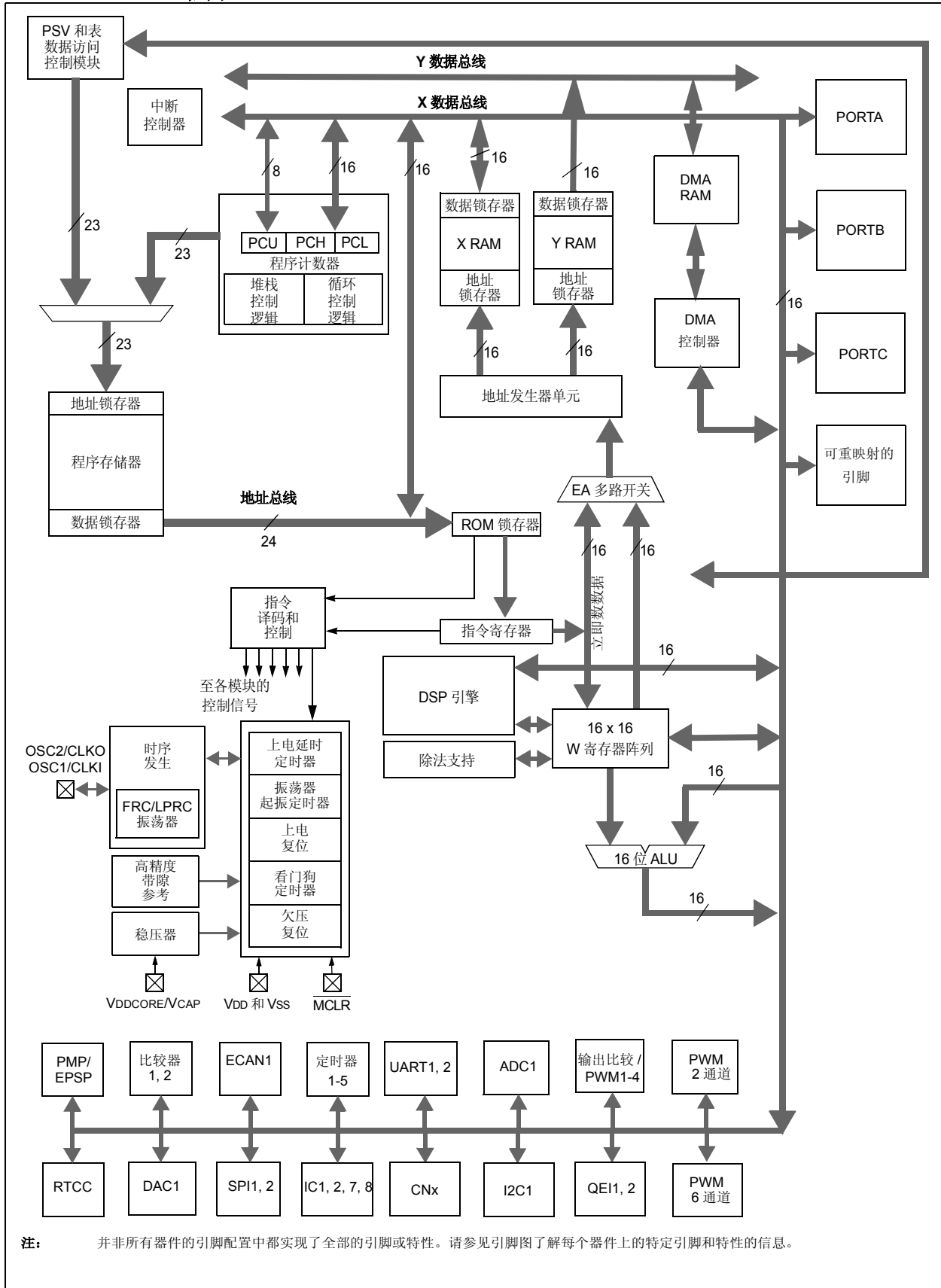




表 1-1: 引脚输入 / 输出说明

引脚名称	引脚类型	缓冲器类型	说明
AN0-AN8	I	Analog	模拟输入通道。
CLKI CLKO	I O	ST/CMOS —	外部时钟源输入。总是与 OSC1 引脚功能相关联。 晶振输出。在晶振模式下，连接到晶振或谐振器。也可选择在 RC 和 EC 模式下用作 CLKO。总是与 OSC2 引脚功能相关联。
OSC1 OSC2	I I/O	ST/CMOS —	晶振输入。配置为 RC 模式时为 ST 缓冲器输入；否则为 CMOS 缓冲器输入。 晶振输出。在晶振模式下，连接到晶振或谐振器。也可选择在 RC 和 EC 模式下用作 CLKO。
SOSCI SOSCO	I O	ST/CMOS —	32.768 kHz 低功耗晶振输入；否则为 CMOS 输入。 32.768 kHz 低功耗晶振输出。
CN0-CN30	I	ST	电平变化通知输入。 可将所有输入用软件编程为内部弱上拉。
IC1-IC2 IC7-IC8	I I	ST ST	捕捉输入 1/2 捕捉输入 7/8。
OCFA OC1-OC4	I O	ST —	比较故障 A 输入（用于比较通道 1、2、3 和 4）。 比较输出 1 至 4。
INT0 INT1 INT2	I I I	ST ST ST	外部中断 0。 外部中断 1。 外部中断 2。
RA0-RA4 RA7-RA10	I/O I/O	ST ST	PORTA 是双向 I/O 端口。 PORTA 是双向 I/O 端口。
RB0-RB15	I/O	ST	PORTB 是双向 I/O 端口。
RC0-RC9	I/O	ST	PORTC 是双向 I/O 端口。
T1CK T2CK T3CK T4CK T5CK	I I I I I	ST ST ST ST ST	Timer1 外部时钟输入。 Timer2 外部时钟输入。 Timer3 外部时钟输入。 Timer4 外部时钟输入。 Timer5 外部时钟输入。
U1CTS U1RTS U1RX U1TX	I O I O	ST — ST —	UART1 允许发送。 UART1 请求发送。 UART1 接收。 UART1 发送。
U2CTS U2RTS U2RX U2TX	I O I O	ST — ST —	UART2 允许发送。 UART2 请求发送。 UART2 接收。 UART2 发送。
SCK1 SDI1 SDO1 SS1	I/O I O I/O	ST ST — ST	SPI1 的同步串行时钟输入 / 输出。 SPI1 数据输入。 SPI1 数据输出。 SPI1 从同步或帧脉冲 I/O。
SCK2 SDI2 SDO2 SS2	I/O I O I/O	ST ST — ST	SPI2 的同步串行时钟输入 / 输出。 SPI2 数据输入。 SPI2 数据输出。 SPI2 从同步或帧脉冲 I/O。
SCL1 SDA1 ASCL1 ASDA1	I/O I/O I/O I/O	ST ST ST ST	I2C1 的同步串行时钟输入 / 输出。 I2C1 的同步串行数据输入 / 输出。 I2C1 的备用同步串行时钟输入 / 输出。 I2C1 的备用同步串行数据输入 / 输出。

图注: CMOS = CMOS 兼容输入或输出  
ST = CMOS 电平的施密特触发器输入

Analog = 模拟输入  
O = 输出

P = 电源  
I = 输入

表 1-1: 引脚输入 / 输出说明 (续)

引脚名称	引脚类型	缓冲器类型	说明
TMS TCK TDI TDO	I I I O	ST ST ST —	JTAG 测试模式选择引脚。 JTAG 测试时钟输入引脚。 JTAG 测试数据输入引脚。 JTAG 测试数据输出引脚。
INDX1 QE1 QEB1 UPDN1	I I I O	ST ST ST CMOS	正交编码器索引 1 脉冲输入。 在 QE1 模式下为正交编码器 A 相输入。 在定时器模式下为辅助定时器外部时钟 / 门控输入。 在 QE1 模式下为正交编码器 B 相输入。 在定时器模式下为辅助定时器外部时钟 / 门控输入。 位置递增 / 递减计数器方向状态。
INDX2 QE2 QEB2 UPDN2	I I I O	ST ST ST CMOS	正交编码器索引 2 脉冲输入。 在 QE2 模式下为正交编码器 A 相输入。 在定时器模式下为辅助定时器外部时钟 / 门控输入。 在 QE2 模式下为正交编码器 B 相输入。 在定时器模式下为辅助定时器外部时钟 / 门控输入。 位置递增 / 递减计数器方向状态。
C1RX C1TX	I O	ST —	ECAN1 总线接收引脚。 ECAN1 总线发送引脚。
RTCC	O	—	实时时钟闹钟输出。
CVREF	O	ANA	比较器参考电压输出。
C1IN- C1IN+ C1OUT	I I O	ANA ANA —	比较器 1 反相输入。 比较器 1 同相输入。 比较器 1 输出。
C2IN- C2IN+ C2OUT	I I O	ANA ANA —	比较器 2 反相输入。 比较器 2 同相输入。 比较器 2 输出。
PMA0 PMA1 PMA2 -PMPA10 PMBE PMCS1 PMD0-PMPD7 PMRD PMWR	I/O I/O O O O I/O O O	ST ST — — — ST — —	并行主端口地址 Bit 0 输入 (缓冲从模式下) 和输出 (主模式下)。 并行主端口地址 Bit 1 输入 (缓冲从模式下) 和输出 (主模式下)。 并行主端口地址线 (非复用主模式)。 并行主端口字节使能选通。 并行主端口片选 1 选通。 并行主端口数据线 (非复用主模式下) 或者地址 / 数据线 (复用主模式下)。 并行主端口读选通。 并行主端口写选通。
DAC1RN DAC1RP DAC1RM	O O O	— — —	DAC1 负输出。 DAC1 正输出。 DAC1 中点电压值输出 (典型值为 1.65V)。
DAC2RN DAC2RP DAC2RM	O O O	— — —	DAC2 负输出。 DAC2 正输出。 DAC2 中点电压值输出 (典型值为 1.65V)。
FLTA1 PWM1L1 PWM1H1 PWM1L2 PWM1H2 PWM1L3 PWM1H3 FLTA2 PWM2L1 PWM2H1	I O O O O O O I O O	ST — — — — — — ST — —	PWM1 故障 A 输入。 PWM1 低端输出 1 PWM1 高端输出 1 PWM1 低端输出 2 PWM1 高端输出 2 PWM1 低端输出 3 PWM1 高端输出 3 PWM2 故障 A 输入。 PWM2 低端输出 1 PWM2 高端输出 1

图注: CMOS = CMOS 兼容输入或输出      Analog = 模拟输入      P = 电源  
ST = CMOS 电平的施密特触发器输入      O = 输出      I = 输入

表 1-1: 引脚输入 / 输出说明 (续)

引脚名称	引脚类型	缓冲器类型	说明
PGD1/EMUD1	I/O	ST	编程 / 调试通信通道 1 使用的数据 I/O 引脚。
PGC1/EMUC1	I	ST	编程 / 调试通信通道 1 使用的时钟输入引脚。
PGD2/EMUD2	I/O	ST	编程 / 调试通信通道 2 使用的数据 I/O 引脚。
PGC2/EMUC2	I	ST	编程 / 调试通信通道 2 使用的时钟输入引脚。
PGD3/EMUD3	I/O	ST	编程 / 调试通信通道 3 使用的数据 I/O 引脚。
PGC3/EMUC3	I	ST	编程 / 调试通信通道 3 使用的时钟输入引脚。
MCLR	I/P	ST	主复位输入。此引脚为低电平有效的器件复位输入。
AVDD	P	P	模拟模块的正电源。
AVSS	P	P	模拟模块的参考地。
VDD	P	—	外设逻辑和 I/O 引脚的正电源。
VDDCORE	P	—	CPU 逻辑滤波电容连接。
Vss	P	—	逻辑和 I/O 引脚的参考地。
VREF+	I	Analog	模拟参考电压 (高电压) 输入。
VREF-	I	Analog	模拟参考电压 (低电压) 输入。

图注: CMOS = CMOS 兼容输入或输出  
 ST = CMOS 电平的施密特触发器输入  
 Analog = 模拟输入  
 O = 输出  
 P = 电源  
 I = 输入

注:

## 2.0 CPU

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 2. CPU”（DS70204），该文档可从 Microchip 网站（www.microchip.com）下载。

### 2.1 概述

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的 CPU 模块采用 16 位（数据）的改进型哈佛架构，具有增强指令集，包括对 DSP 的强大支持。CPU 具有 24 位指令字，指令字带有长度可变的操作码字段。程序计数器（Program Counter, PC）为 23 位宽，可寻址最大 4M x 24 位的用户程序存储空间。实际实现的程序存储容量因器件而异。单周期指令预取机制可帮助维持吞吐量并提供指令的预测执行功能。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 DO 和 REPEAT 指令支持无开销的程序循环结构，这两条指令在任何时间都可以被中断。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 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（Stack Pointer, SP），用于中断和调用。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件具有两类指令：MCU 类指令和 DSP 类指令。这两类指令无缝地集成到单个 CPU 中。指令集包括很多寻址模式，指令的设计可使 C 编译器的效率达到最优。对于大多数指令，dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 能够在每个指令周期内执行一次数据（或程序数据）存储器读取、一次工作寄存器（数据）读取、一次数据存储器写入以及一次程序（指令）存储器读取操作。因此，支持三操作数指令，允许在单个周期内执行  $A + B = C$  这样的操作。

CPU 的框图如图 2-1 所示，dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的编程模型如图 2-2 所示。

### 2.2 数据寻址概述

数据空间可以作为 32K 字或 64 KB 寻址，并被分成两块，称为 X 和 Y 数据存储区。每个存储区有各自独立的地址发生单元（Address Generation Unit, AGU）。MCU 类指令只通过 X 存储空间 AGU 进行操作，可将整个存储器映射作为一个线性数据空间访问。某些 DSP 指令通过 X 和 Y 的 AGU 进行操作以支持双操作数读操作，这样会将数据地址空间分成两个部分。X 和 Y 数据空间的边界视具体器件而定。

X 和 Y 地址空间都支持无开销循环缓冲区（模寻址模式）。模寻址省去了 DSP 算法的软件边界检查开销。此外，X AGU 的循环寻址可以用于任何 MCU 类指令。X AGU 还支持位反转寻址，大幅简化了基为 2 的 FFT 算法对输入或输出数据的重新排序。

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

### 2.3 DSP 引擎概述

DSP 引擎具有一个高速 17 位 x 17 位乘法器、一个 40 位 ALU、两个 40 位饱和累加器和一个 40 位双向桶形移位寄存器。该桶形移位寄存器能在单个周期内将一个 40 位的值右移或左移最多 16 位。DSP 指令可以无缝地与所有其他指令一起操作，且设计为能获得最佳实时性能。MAC 指令和其他相关指令可以在同一个周期内，同时完成从存储器中取两个数据操作数，将两个 W 寄存器相乘并累加，且可选择使结果饱和。这要求 RAM 数据空间对于这些指令拆分为两块，但对于所有其他指令保持线性。数据空间分块是通过将某些工作寄存器专用于每个地址空间，以透明和灵活的方式实现的。

## 2.4 MCU 的特性

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 具有一个由 MCU ALU 和 DSP 引擎共用的 17 位 x 17 位单周期乘法器。此乘法器可以进行有符号、无符号和混合符号的乘法运算。使用 17 位 x 17 位乘法器进行 16 位 x 16 位乘法运算不仅允许您执行混合符号的乘法运算，而且对于 (-1.0) x (-1.0) 这样的特殊运算也可以得到准确结果。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 支持小数和整数的 16/16 位和 32/16 位除法运算。所有的除法指令都是迭代操作。它们必须在一个 REPEAT 循环内执行，总执行时间为 19 个指令周期。在这 19 个周期的任一周期内可以中断除法运算而不会丢失数据。

一个 40 位桶形移位寄存器用于在单个周期内将数据左移或右移最多 16 位。MCU 和 DSP 指令都可以使用该桶形移位寄存器。

图 2-1: dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的 CPU 内核框图

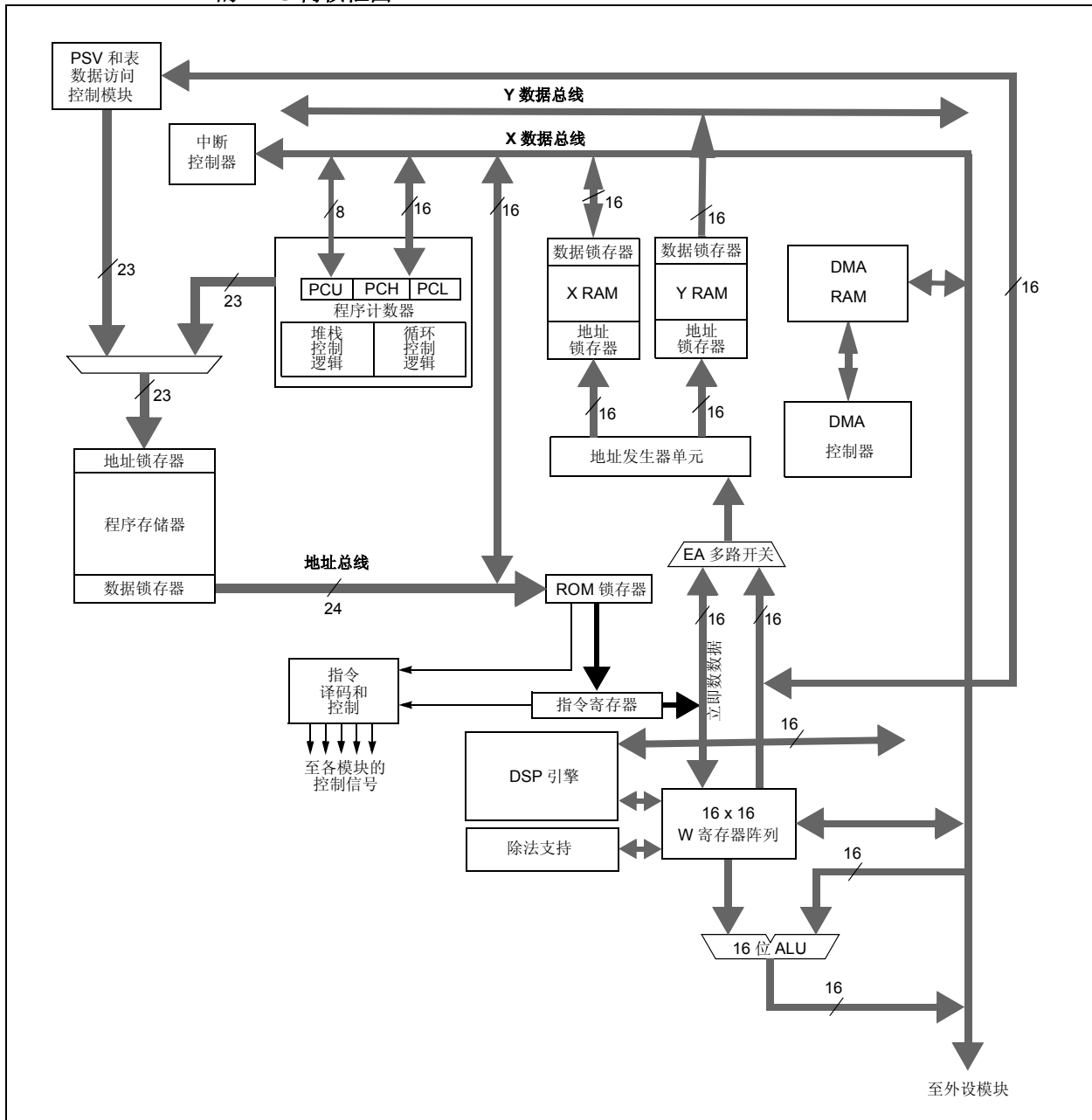
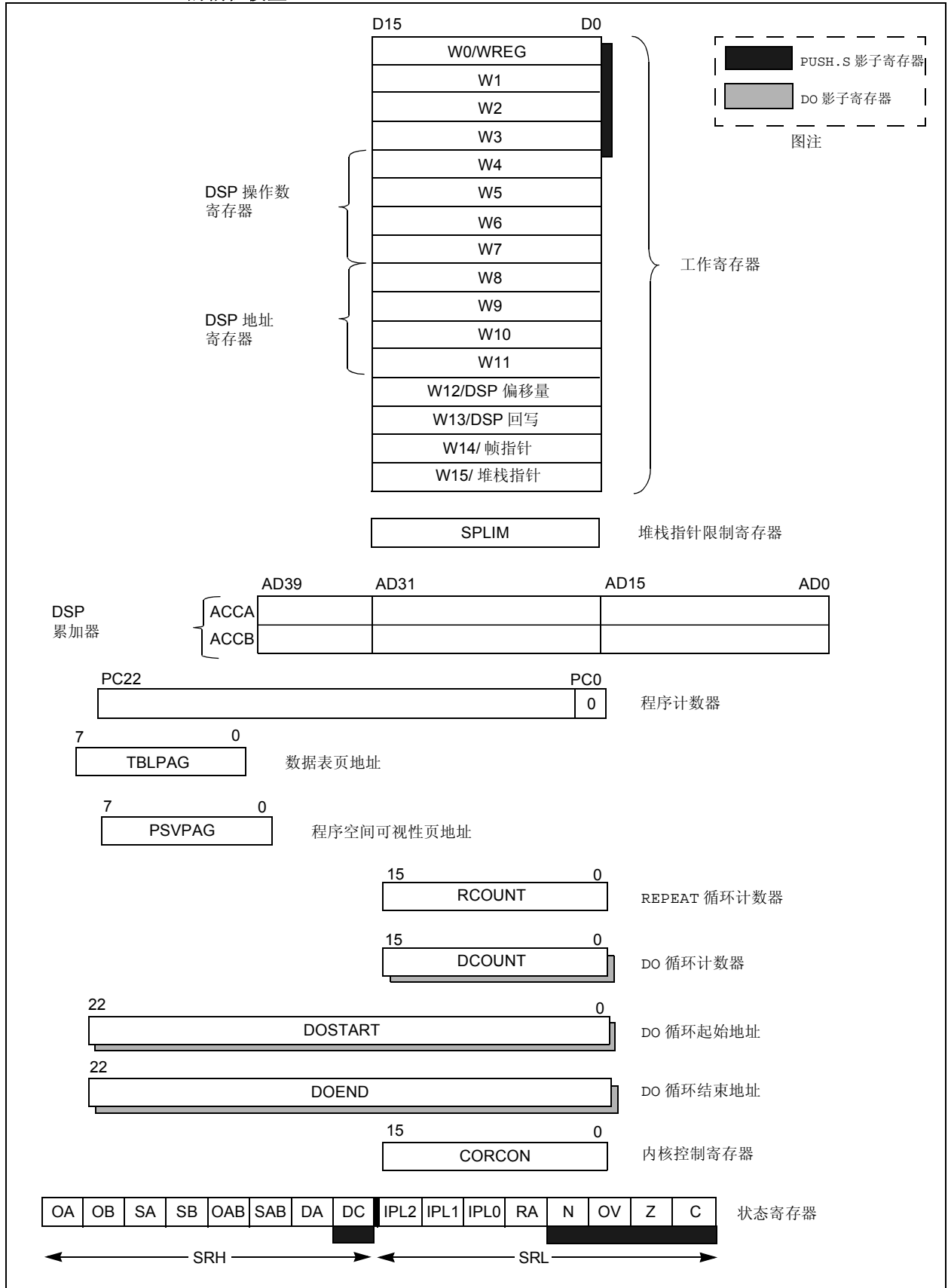


图 2-2: dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的编程模型



## 2.5 CPU 控制寄存器

寄存器 2-1: SR: CPU 状态寄存器

R-0	R-0	R/C-0	R/C-0	R-0	R/C-0	R-0	R/W-0
OA	OB	SA <sup>(1)</sup>	SB <sup>(1)</sup>	OAB	SAB	DA	DC
bit 15						bit 8	
R/W-0 <sup>(3)</sup>	R/W-0 <sup>(3)</sup>	R/W-0 <sup>(3)</sup>	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL<2:0> <sup>(2)</sup>			RA	N	OV	Z	C
bit 7						bit 0	

**图注:**

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

- bit 15     **OA:** 累加器 A 溢出状态位  
1 = 累加器 A 溢出  
0 = 累加器 A 未溢出
- bit 14     **OB:** 累加器 B 溢出状态位  
1 = 累加器 B 溢出  
0 = 累加器 B 未溢出
- bit 13     **SA:** 累加器 A 饱和 “粘住” 状态位 <sup>(1)</sup>  
1 = 累加器 A 饱和或在某时已经饱和  
0 = 累加器 A 未饱和
- bit 12     **SB:** 累加器 B 饱和 “粘住” 状态位 <sup>(1)</sup>  
1 = 累加器 B 饱和或在某时已经饱和  
0 = 累加器 B 未饱和
- bit 11     **OAB:** OA || OB 组合的累加器溢出状态位  
1 = 累加器 A 或 B 已经溢出  
0 = 累加器 A 和 B 都未溢出
- bit 10     **SAB:** SA || SB 组合的累加器饱和 (粘住) 状态位 <sup>(4)</sup>  
1 = 累加器 A 或 B 饱和或在过去某时已经饱和  
0 = 累加器 A 和 B 都未饱和
- bit 9       **DA:** DO 循环活动位  
1 = 正在进行 DO 循环  
0 = 不在进行 DO 循环
- bit 8       **DC:** MCU ALU 半进位 / 借位标志位  
1 = 结果的第 4 个低位 (对于字节大小的数据) 或第 8 个低位 (对于字大小的数据) 发生了进位  
0 = 结果的第 4 个低位 (对于字节大小的数据) 或第 8 个低位 (对于字大小的数据) 未发生进位

- 注**
- 1: 此位可被读取或清零 (但不能置 1)。
  - 2: IPL<2:0> 位与 IPL<3> 位 (CORCON<3>) 组合形成 CPU 中断优先级。如果 IPL<3> = 1, 那么括号中的值表示 IPL。当 IPL<3> = 1 时, 禁止用户中断。
  - 3: 当 NSTDIS (INTCON1<15>) = 1 时, IPL<2:0> 状态位是只读的。
  - 4: 此位可被读取或清零 (但不能置 1)。清零此位的同时将清零 SA 和 SB。



寄存器 2-1: SR: CPU 状态寄存器 (续)

bit 7-5	<b>IPL&lt;2:0&gt;</b> : CPU 中断优先级状态位 <sup>(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	<b>RA</b> : REPEAT 循环活动位 1 = 正在进行 REPEAT 循环 0 = 不在进行 REPEAT 循环
bit 3	<b>N</b> : MCU ALU 负标志位 1 = 结果为负 0 = 结果为非负 (零或正值)
bit 2	<b>OV</b> : MCU ALU 溢出标志位 此位用于有符号的算术运算 (以二进制补码方式进行)。它表示量值上的溢出, 这种溢出将导致符号位改变状态。 1 = 有符号算术运算中发生溢出 (本次运算) 0 = 未发生溢出
bit 1	<b>Z</b> : MCU ALU 全零标志位 1 = 影响 Z 位的任何运算在过去某时已将该位置 1 0 = 影响 Z 位的最近一次运算已将该位清零 (即运算结果非零)
bit 0	<b>C</b> : MCU ALU 进位 / 借位标志位 1 = 结果的最高位发生了进位 0 = 结果的最高位未发生进位

注 1: 此位可被读取或清零 (但不能置 1)。

2: IPL<2:0> 位与 IPL<3> 位 (CORCON<3>) 组合形成 CPU 中断优先级。如果 IPL<3> = 1, 那么括号中的值表示 IPL。当 IPL<3> = 1 时, 禁止用户中断。

3: 当 NSTDIS (INTCON1<15>) = 1 时, IPL<2:0> 状态位是只读的。

4: 此位可被读取或清零 (但不能置 1)。清零此位的同时将清零 SA 和 SB。

**寄存器 2-2: CORCON: 内核控制寄存器**

U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-0	R-0	
—	—	—	US	EDT <sup>(1)</sup>	DL<2:0>			
bit 15								bit 8

R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R/W-0	R/W-0	R/W-0	
SATA	SATB	SATDW	ACCSAT	IPL3 <sup>(2)</sup>	PSV	RND	IF	
bit 7								bit 0

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

- bit 15-13     **未实现:** 读为 0
- bit 12       **US:** DSP 乘法无符号 / 有符号控制位  
1 = DSP 引擎执行无符号乘法运算  
0 = DSP 引擎执行有符号乘法运算
- bit 11       **EDT:** DO 循环提前终止控制位 <sup>(1)</sup>  
1 = 在当前循环迭代结束时终止执行 DO 循环  
0 = 无影响
- bit 10-8     **DL<2:0>:** DO 循环嵌套层级状态位  
111 = 正在进行 7 层 DO 循环嵌套  
·  
·  
·  
001 = 正在进行 1 层 DO 循环嵌套  
000 = 无 DO 循环嵌套
- bit 7        **SATA:** ACCA 饱和和使能位  
1 = 使能累加器 A 饱和  
0 = 禁止累加器 A 饱和
- bit 6        **SATB:** ACCB 饱和和使能位  
1 = 使能累加器 B 饱和  
0 = 禁止累加器 B 饱和
- bit 5        **SATDW:** DSP 引擎的数据空间写饱和和使能位  
1 = 使能数据空间写饱和  
0 = 禁止数据空间写饱和
- bit 4        **ACCSAT:** 累加器饱和模式选择位  
1 = 9.31 饱和 (超饱和)  
0 = 1.31 饱和 (正常饱和)
- bit 3        **IPL3:** CPU 中断优先级状态位 3<sup>(2)</sup>  
1 = CPU 中断优先级大于 7  
0 = CPU 中断优先级等于或小于 7
- bit 2        **PSV:** 数据空间中程序空间可视性使能位  
1 = 程序空间在数据空间中可视  
0 = 程序空间在数据空间中不可视

**注 1:** 此位总是读为 0。  
**注 2:** IPL3 位与 IPL<2:0> 位 (SR<7:5>) 组合形成 CPU 中断优先级。

**寄存器 2-2: CORCON: 内核控制寄存器 (续)**

bit 1      **RND:** 舍入模式选择位  
          1 = 使能有偏 (常规) 舍入  
          0 = 使能无偏 (收敛) 舍入

bit 0      **IF:** 整数或小数乘法器模式选择位  
          1 = 使能 DSP 乘法运算的整数模式  
          0 = 使能 DSP 乘法运算的小数模式

- 注 1:** 此位总是读为 0。  
**2:** IPL3 位与 IPL<2:0> 位 (SR<7:5>) 组合形成 CPU 中断优先级。

## 2.6 算术逻辑单元 (ALU)

dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的 ALU 为 16 位宽, 可进行加法、减法、移位和逻辑运算。除非另外指明, 算术运算实际上采用二进制补码形式进行。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) 和半进位状态位 (DC) 的值。在减法运算中, C 和 DC 状态位分别作为借位标志位和半借位标志位。

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

有关每条指令所影响的 SR 位的信息, 请参见《dsPIC30F/33F 程序员参考手册》(DS70157B\_CN)。

dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的 CPU 融入了对乘法和除法的硬件支持。它包含了专门的硬件乘法器以及支持 16 位除数除法的硬件。

### 2.6.1 乘法器

通过使用 DSP 引擎的高速 17 位 x 17 位乘法器, ALU 支持各种无符号、有符号或混合符号的 MCU 乘法运算:

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

### 2.6.2 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 /16 位有符号除法
2. 32 位无符号 /16 位无符号除法
3. 16 位有符号 /16 位有符号除法
4. 16 位无符号 /16 位无符号除法

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

## 2.7 DSP 引擎

DSP 引擎由一个高速 17 位 x 17 位乘法器、一个桶形移位寄存器和一个 40 位加法器 / 减法器 (两个目标累加器、舍入逻辑和饱和逻辑) 组成。

dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 采用单周期指令流架构; 因此 DSP 引擎的工作不能与 MCU 指令流同时进行。但是, 某些 MCU ALU 和 DSP 引擎资源可由同一条指令 (如 ED 和 EDAC) 同时使用。

DSP 引擎还可以执行固有的不需要其他数据的累加器—累加器操作。这些指令是 ADD、SUB 和 NEG。

通过 CPU 内核控制寄存器 (CORCON) 中的各个位, 可以对 DSP 引擎的操作进行多种选择, 这些选择如下:

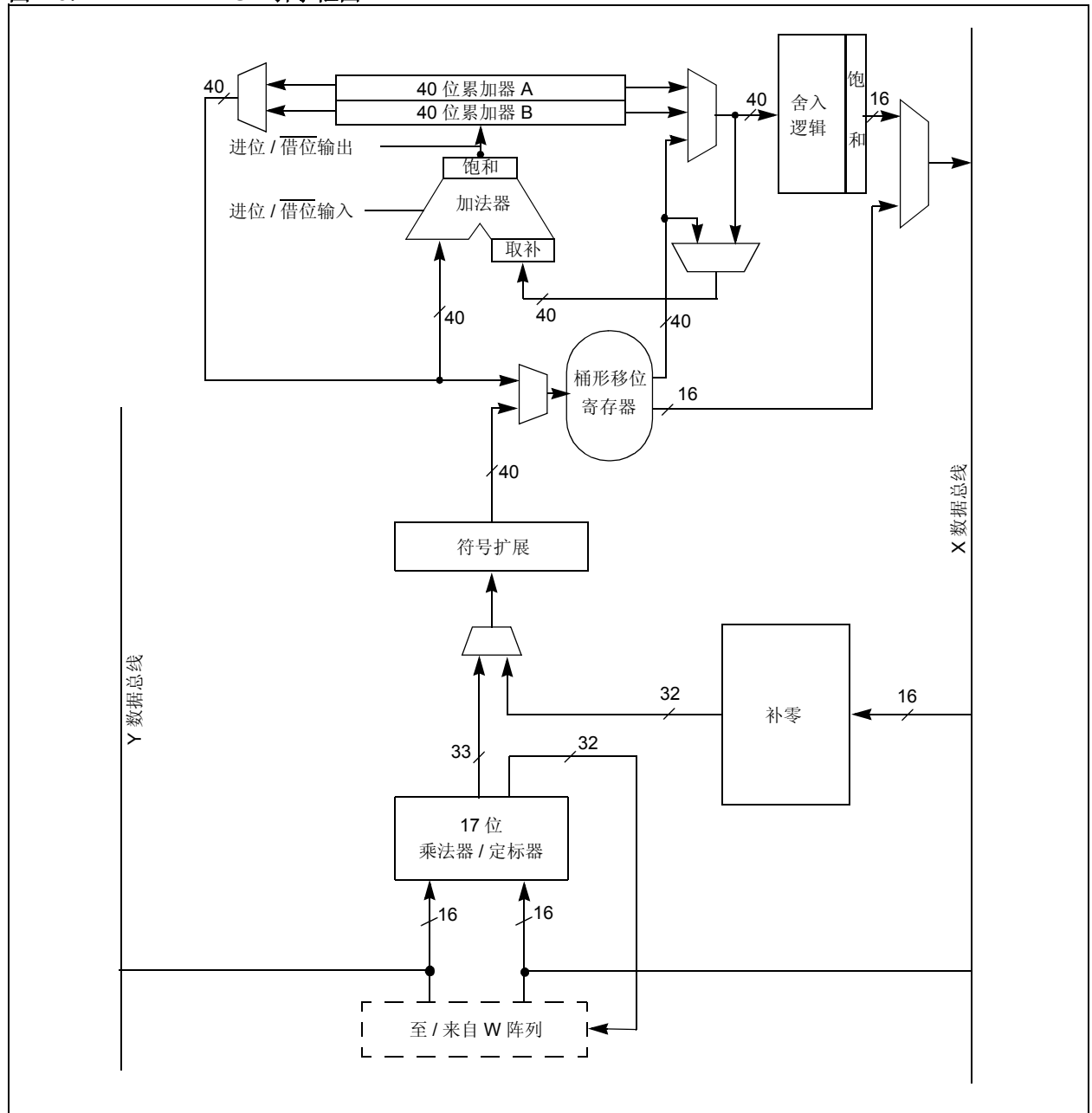
- 小数或整数 DSP 乘法 (IF)
- 有符号或无符号 DSP 乘法 (US)
- 常规或收敛舍入 (RND)
- ACCA 自动饱和和使能 / 禁止 (SATA)
- ACCB 自动饱和和使能 / 禁止 (SATB)
- 对于写数据存储器, 自动饱和和使能 / 禁止 (SATDW)
- 累加器饱和模式选择 (ACCSAT)

DSP 引擎的框图如图 2-3 所示。

表 2-1: DSP 指令汇总

指令	代数运算	ACC 回写
CLR	$A = 0$	有
ED	$A = (x - y)2$	无
EDAC	$A = A + (x - y)2$	无
MAC	$A = A + (x \cdot y)$	有
MAC	$A = A + x2$	无
MOVSAC	A 中内容将不发生改变	有
MPY	$A = x \cdot y$	无
MPY	$A = x2$	无
MPY.N	$A = -x \cdot y$	无
MSC	$A = A - x \cdot y$	有

图 2-3: DSP 引擎框图



### 2.7.1 乘法器

17 位 x17 位乘法器可以进行有符号或无符号运算，其输出经过定标器进行换算后可支持 1.31 小数 (Q31) 或 32 位整数结果。无符号操作数经过零扩展后，送入乘法器输入值的第 17 位。有符号操作数经过符号扩展后，送入乘法器输入值的第 17 位。17 位 x17 位乘法器 / 定标器的输出是 33 位值，它将被符号扩展为 40 位。整型数据的固有表示形式为有符号的二进制补码值，其中最高有效位 (MSb) 定义为符号位。N 位二进制补码整数的范围为  $-2^{N-1}$  到  $2^{N-1} - 1$ 。

- 对于 16 位整数，数据范围为 -32768 (0x8000) 到 32767 (0x7FFF)，包括 0 在内。
- 对于 32 位整数，数据范围为 -2,147,483,648 (0x8000 0000) 到 2,147,483,647 (0x7FFF FFFF)。

当乘法器配置为小数乘法时，数据表示为二进制补码小数，其中 MSb 定义为符号位，小数点暗含在符号位之后 (QX 格式)。暗含小数点的 N 位二进制补码小数的范围为 -1.0 到  $(1 - 2^{1-N})$ 。对于 16 位小数，Q15 数据范围为 -1.0 (0x8000) 到 0.999969482 (0x7FFF)，包括 0 在内，其精度为  $3.01518 \times 10^{-5}$ 。在小数模式下，16 x 16 乘法运算将产生 1.31 乘积，其精度为  $4.65661 \times 10^{-10}$ 。

同一个乘法器还用来支持 MCU 乘法指令，包括整数的 16 位有符号、无符号和混合符号乘法。

MUL 指令可以使用字节或字长度的操作数。字节操作数将产生 16 位结果，而字操作数将产生 32 位结果，结果存放在 W 寄存器阵列的指定寄存器中。

### 2.7.2 数据累加器和加法器 / 减法器

数据累加器包含一个 40 位加法器 / 减法器，它带有自动符号扩展逻辑。它可以选择两个累加器 (A 或 B) 之一作为其累加前的源累加器和累加后的目标累加器。对于 ADD 和 LAC 指令，可选择通过桶形移位器在累加之前将要累加或装入的数据进行换算。

#### 2.7.2.1 加法器 / 减法器、溢出和饱和

加法器 / 减法器是一个 40 位加法器，一侧输入可以选择为零，而另一侧输入可以是原数据或求补后的数据。

- 对于加法，进位 / 借位输入为高电平有效，另一侧输入是原数据 (没有求补的)。
- 对于减法，进位 / 借位输入为低电平有效，另一侧输入是求补后的数据。

加法器 / 减法器产生溢出状态位 SA/SB 和 OA/OB，这些状态位被锁存在状态寄存器中并在其中得到反映。

- 从 bit 39 溢出：这是灾难性溢出，会破坏累加器的符号位。
- 溢出到警戒位 (bit 32 到 bit 39)：这是可恢复的溢出。每当警戒位彼此不完全一致时，就将把这个状态位置 1。

加法器有一个额外的饱和模块，如果选取的话，饱和模块将控制累加器的数据饱和。饱和模块使用加法器的结果、上述的溢出状态位、SAT<A:B> (CORCON<7:6>) 和 ACCSAT (CORCON<4>) 模式控制位，来确定何时饱和、达到何值为饱和。

状态寄存器中有 6 个支持饱和与溢出的位：

- OA: ACCA 溢出到警戒位
- OB: ACCB 溢出到警戒位
- SA: ACCA 已饱和 (bit 31 溢出并饱和)  
或者  
ACCA 溢出到警戒位并饱和 (bit 39 溢出并饱和)
- SB: ACCB 已饱和 (bit 31 溢出并饱和)  
或者  
ACCB 溢出到警戒位并饱和 (bit 39 溢出并饱和)
- OAB: OA 和 OB 的逻辑或 (OR)
- SAB: SA 和 SB 的逻辑或 (OR)

每次数据通过加法器 / 减法器，就会修改 OA 和 OB 位。置 1 时，它们表明最近的操作已溢出到累加器警戒位 (bit 32 到 bit 39)。如果 OA 和 OB 位置 1 而且 INTCON1 寄存器中相应的溢出陷阱标志允许位 (OVATE 和 OVBTE) 置 1 的话，还可以选择用 OA 和 OB 位产生算术警告陷阱 (见第 6.0 节 “中断控制器”)。这使得用户应用程序能够立即采取措施，例如，校正系统增益。

每次数据通过加法器 / 减法器，就会修改 SA 和 SB 位，但用户应用程序只能对它们进行清零。置 1 时，它们表明累加器已溢出其最大范围 (对于 32 位饱和是 bit 31，而 40 位饱和是 bit 39)，并发生饱和 (如果饱和使能的话)。如果没有使能饱和，SA 和 SB 置 1 默认为 bit 39 溢出，以此指示产生了灾难性溢出。如果 INTCON1 寄存器中的 COVTE 位置 1，当饱和被禁止时，SA 和 SB 位将产生算术警告陷阱。

在状态寄存器（SR）中，对于溢出和饱和状态位，可以将 OA 和 OB 的逻辑或形成 OAB 位，将 SA 和 SB 的逻辑或形成 SAB 位。这样，只需检查状态寄存器中的一个位，程序员就能判断是否有累加器溢出；检查状态寄存器中的另一个位，就可以判断是否有累加器饱和。对于通常要使用两个累加器的复数运算而言，这很有用。

器件支持三种饱和与溢出模式：

- **bit 39 溢出和饱和：**  
当发生 bit 39 溢出和饱和时，饱和逻辑将最大的正 9.31 值（0x7FFFFFFF）或最小的负 9.31 值（0x80000000）装入目标累加器。SA 或 SB 位置 1 并保持直到被用户应用程序清零。这称为“超饱和”，为错误数据或不可预期的算法问题（例如，增益计算）提供了保护机制。
- **bit 31 溢出和饱和：**  
当发生 bit 31 溢出和饱和时，饱和逻辑将最大的正 1.31 值（0x007FFFFFFF）或最小的负 1.31 值（0x00800000）装入目标累加器。SA 或 SB 位置 1 并保持直到被用户应用程序清零。当这种饱和模式生效时，不使用警戒位，因此 OA、OB 或 OAB 位不会被置 1。
- **bit 39 灾难性溢出：**  
加法器的 bit 39 溢出状态位用来将 SA 或 SB 位置 1；这两位置 1 后，将保持状态直到被用户应用程序清零。不进行饱和操作，允许累加器溢出（破坏其符号位）。如果 INTCON1 寄存器中的 COVTE 位置 1，灾难性溢出会导致一个陷阱异常。

### 2.7.3 累加器“回写”

MAC 类指令（MPY、MPY.N、ED 和 EDAC 除外）可以选择将累加器高位字（bit 16 到 bit 31）的舍入形式写入数据存储空间，前提是当前指令不对该累加器进行操作。通过 X 总线寻址组合的 X 和 Y 地址空间，执行回写操作。支持以下寻址模式：

- **W13，寄存器直接寻址：**  
非操作目标的累加器的舍入内容以 1.15 小数形式写入 W13。
- **[W13]+ = 2，执行后递增的寄存器间接寻址：**  
非操作目标的累加器的舍入内容以 1.15 小数形式写入 W13 指向的地址。然后 W13 递增 2（对于字写入）。

#### 2.7.3.1 舍入逻辑

舍入逻辑是一个组合模块，在累加器写（存储）过程中执行常规的（有偏）或收敛的（无偏）舍入功能。舍入模式由 CORCON 寄存器中 RND 位的状态决定。它会产生一个 16 位的 1.15 数据值，该值被送到数据空间写饱和和逻辑。如果指令不指明舍入，就会存储一个截取的 1.15 数据值，简单地丢弃低位字。

常规舍入取累加器的 bit 15，对它进行零扩展并将扩展后的值加到 ACCxH 字（累加器的 bit 16 到 bit 31）。

- 如果 ACCxL 字（累加器的 bit 0 到 bit 15）在 0x8000 和 0xFFFF 之间（包括 0x8000），则 ACCxH 递增 1。
- 如果 ACCxL 在 0x0000 和 0x7FFF 之间，则 ACCxH 不变。

此算法的结果经过一系列随机舍入操作，值会稍稍偏大（正偏）。

除非 ACCxL 等于 0x8000，否则收敛的（或无偏）舍入操作方式与常规舍入相同。在这种情况下，要对 ACCxH 的最低位（累加器的 bit 16）进行检测：

- 如果它为 1，ACCxH 递增 1。
- 如果它为 0，ACCxH 不变。

假设 bit 16 本身是随机的，这样的机制将消除任何可能累加的舍入偏差。

通过 X 总线，SAC 和 SAC.R 指令将目标累加器内容的截取（SAC）或舍入（SAC.R）形式存入数据存储空间（这受数据饱和的影响，请参见第 2.7.3.2 节“数据空间写饱和”）。对于 MAC 类指令，累加器回写操作将以同样的方式进行，通过 X 总线寻址组合的 MCU（X 和 Y）数据空间。对于此类指令，数据始终要进行舍入。

### 2.7.3.2 数据空间写饱和

除了加法器 / 减法器饱和，对数据空间进行写操作也会饱和，但不会影响源累加器的内容。数据空间写饱和和逻辑块接受来自舍入逻辑块的一个 16 位的 1.15 小数值作为输入，还接受来自源（累加器）和 16 位舍入加法器的溢出状态。这些输入经过组合用来选择适当的 1.15 小数值作为输出，写入数据存储空间中。

如果 CORCON 寄存器中的 SATDW 位置 1，将检测（经过舍入或截取后的）数据是否溢出，并进行相应的调整

- 如果输入数据大于 0x007FFF，则写入存储器中的数据被强制为最大的正 1.15 值，0x7FFF。
- 如果输入数据小于 0xFF8000，则写入存储器中的数据被强制为最小的负 1.15 值，0x8000。

源累加器的最高位（bit 39）用来决定被检测的操作数的符号。

如果 CORCON 寄存器中的 SATDW 位没有置 1，则输入数据都将通过，在任何情况下都不会被修改。

### 2.7.4 桶形移位寄存器

桶形移位寄存器在单个周期内可将数据算术或逻辑右移或左移最多 16 位。源操作数可以是两个 DSP 累加器中的任何一个或 X 总线（支持寄存器或存储器中数据的多位移位）。

移位寄存器需要一个有符号二进制值，用来确定移位操作的幅度（位数）和方向。正值将操作数右移。负值则将操作数左移。值为 0 则不改变操作数。

桶形移位寄存器为 40 位宽，于是，它为 DSP 移位操作提供了 40 位的结果，而为 MCU 移位操作提供 16 位的结果。来自 X 总线的数据在桶形移位寄存器中的存放方式是：右移则数据存放在 bit 16 到 bit 31，左移则存放在 bit 0 到 bit 15。



### 3.0 存储器构成

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 4. Program Memory”（DS70203），该文档可从 Microchip 网站（www.microchip.com）下载。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 架构具有独立的程序和配置存储空间 / 总线。这一架构还允许在代码执行过程中从配置空间直接访问程序存储器。

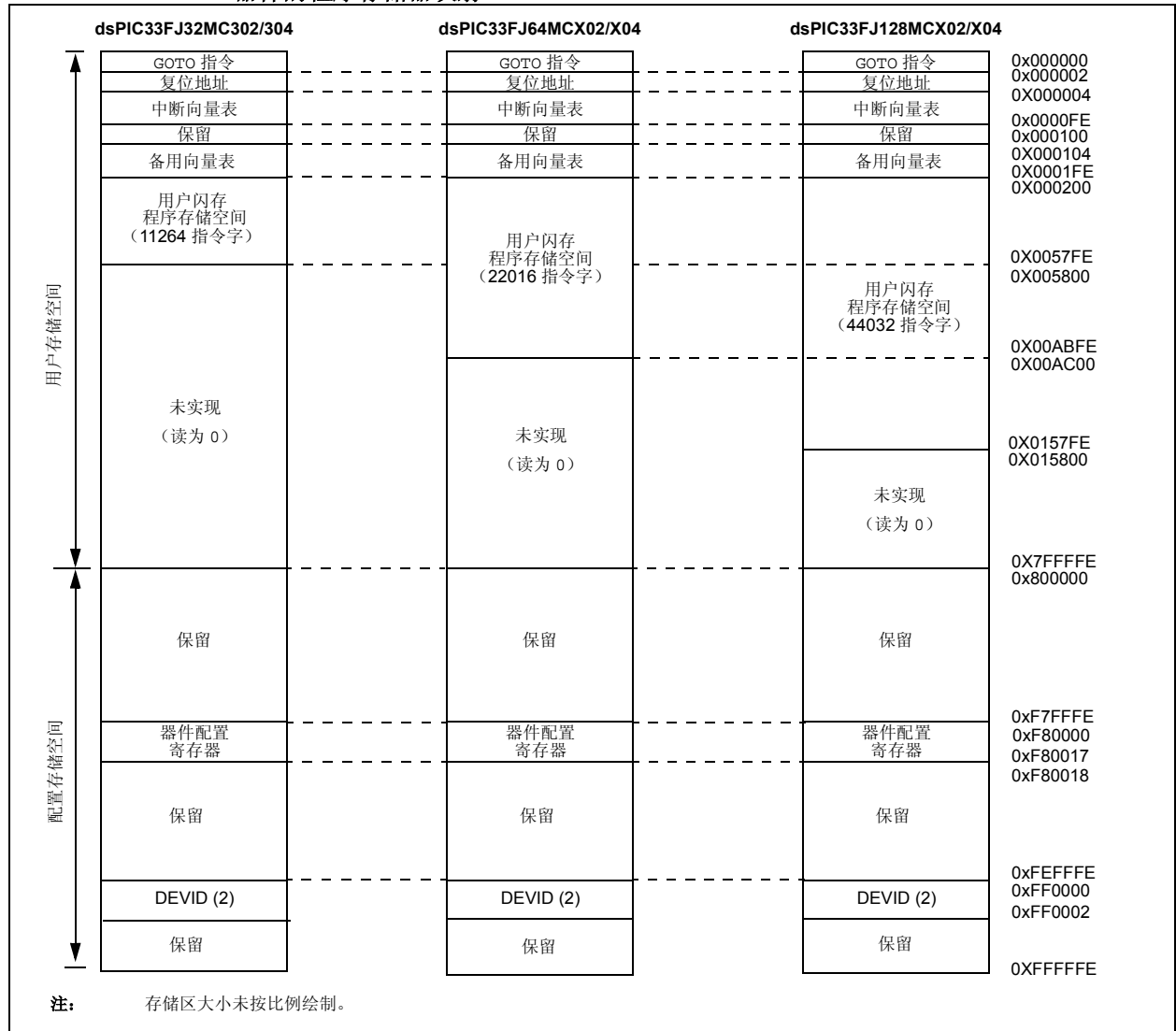
### 3.1 程序地址空间

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件的程序存储空间可存储 4M 个指令字。可通过 24 位值寻址这一空间，24 位值由程序执行过程中的 23 位程序计数器（PC）或第 3.6 节“程序存储空间与数据存储空间的接口”中所述的表操作或数据空间重映射得到。

用户应用程序只能访问程序存储空间的低半地址范围（0x000000 至 0x7FFFFFFF）。使用 TBLRD/TBLWT 指令时，情况有所不同，这两条指令使用 TBLPAG<7> 来允许访问配置存储空间中的配置位和器件 ID。

图 3-1 给出了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件的程序存储器映射。

**图 3-1: dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件的程序存储器映射**



3.1.1 程序存储器构成

程序存储空间由可字寻址的块构成。虽然它被视为24位宽，但将程序存储器的每个地址视作一个低位字和一个高位字的组合更加合理，其中高位字的高字节部分没有实现。低位字的地址始终为偶数，而高位字的地址为奇数（图 3-2）。

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

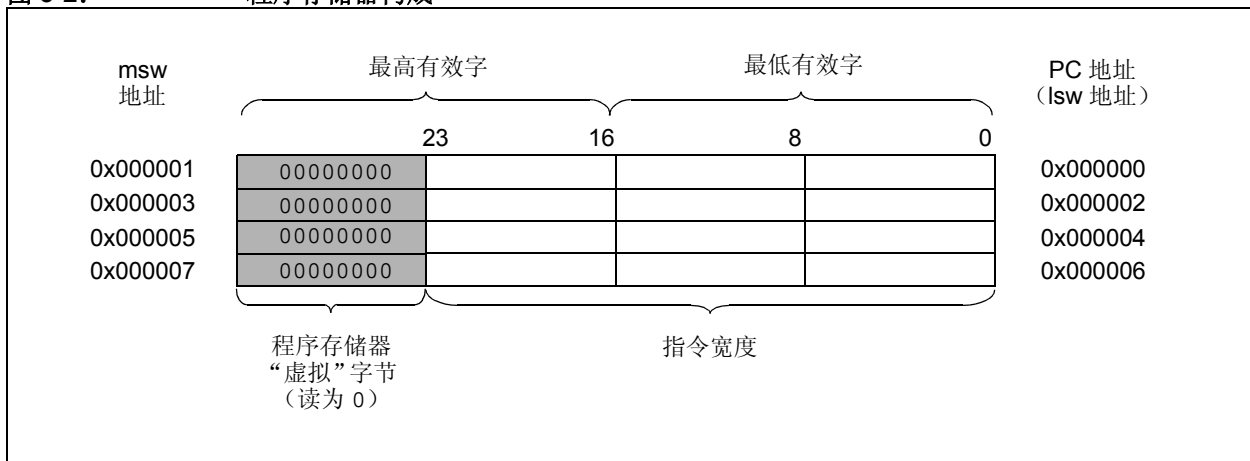
3.1.2 中断向量和陷阱向量

所有 dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件中 0x00000 和 0x000200 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 的默认值重定向到代码实际起始处。用户应用程序可在地址 0x000000 处编写一条 GOTO 指令以将代码的实际起始地址设置为 0x000002。

dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件也具有两个中断向量表，地址分别为从 0x000004 到 0x0000FF 和 0x000100 到 0x0001FF。这两个中断向量表允许使用不同的中断服务程序（Interrupt Service Routines, ISR）处理器件的每个中断源。关于中断向量表更详细的讨论，请参见第 6.1 节“中断向量表”。

图 3-2: 程序存储器构成



## 3.2 数据地址空间

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的 CPU 具有独立的 16 位宽数据存储空间。使用独立的地址发生单元 (AGU) 对数据空间执行读写操作。数据存储映射如图 3-4 所示。

数据存储空间中的所有有效地址 (Effective Adresse, EA) 均为 16 位宽, 并且指向数据空间内的字节。这种构成方式使得数据空间的地址范围为 64 KB 或 32K 字。数据存储空间的低半地址区 (即当  $EA<15> = 0$  时) 用作实现的存储地址, 而高半地址区 ( $EA<15> = 1$ ) 则保留为程序空间可视性 (Program Space Visibility, PSV) 区域 (见第 3.6.3 节 “使用程序空间可视性读程序存储器中的数据”)。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件实现最大 16 KB 的数据存储器。如果 EA 指向了该区域以外的存储单元, 则将返回一个全零的字或字节。

### 3.2.1 数据空间宽度

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

### 3.2.2 数据存储器构成和对齐方式

为维持与 PIC<sup>®</sup>MCU 器件的后向兼容性和提高数据存储空间的使用效率, dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的指令集同时支持字和字节操作。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 EA 的 LSB 来确定要选取的字节, 数据字节读取将读取包含字节的整个字。选定的字节被放在数据路径的 LSB 处。这就是说, 数据存储器和寄存器被组织为两个并行的字节宽的实体, 它们共享 (字) 地址译码, 但写入线独立。数据字节写操作只写入阵列或寄存器中与字节地址匹配的那一侧。

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

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

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

### 3.2.3 SFR 空间

Near 数据空间的前 2 KB 存储单元 (从 0x0000 到 0x07FF) 主要被特殊功能寄存器 (Special Function Registers, SFR) 占用。dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 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; 它们读为 0。

**注:** 不同器件的实际外设功能集和中断也有所不同。请参见相应器件的数据表和引脚图了解特定器件的信息。

### 3.2.4 NEAR 数据空间

在 0x0000 和 0x1FFF 之间的 8 KB 区域被称为 near 数据空间。可以使用所有存储器直接寻址指令中的 13 位绝对地址字段直接寻址这一空间中的存储单元。此外, 还可以使用 MOV 指令寻址整个数据空间, MOV 指令支持使用 16 位地址字段的存储器直接寻址模式或使用工作寄存器作为地址指针的间接寻址模式。

图 3-3: 带 4 KB RAM 的 dsPIC33FJ32MC302/304 器件的数据存储器映射

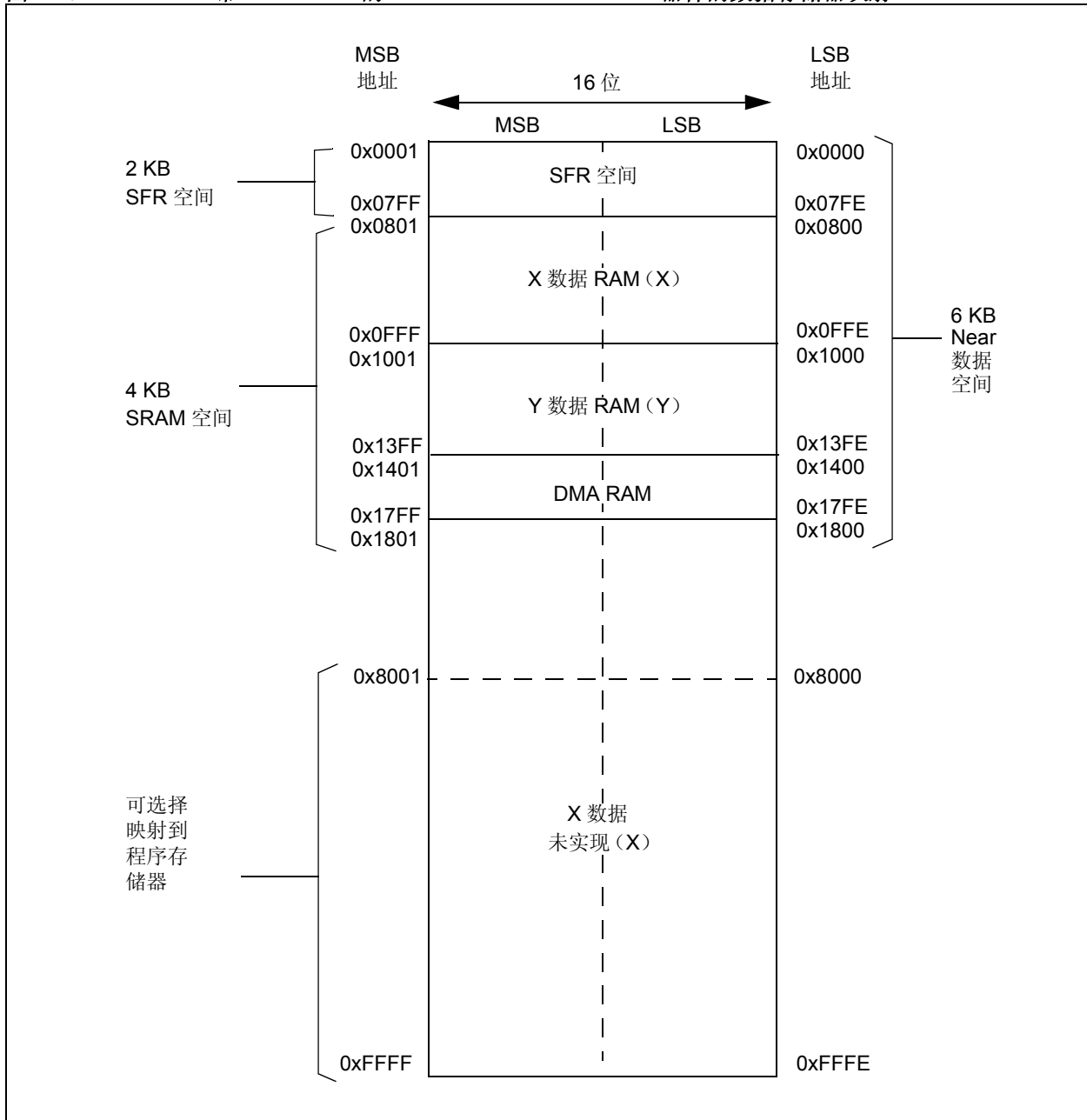


图 3-4: 带 8 KB RAM 的 dsPIC33FJ128MC202/204 和 dsPIC33FJ64MC202/204 器件的数据存储器映射

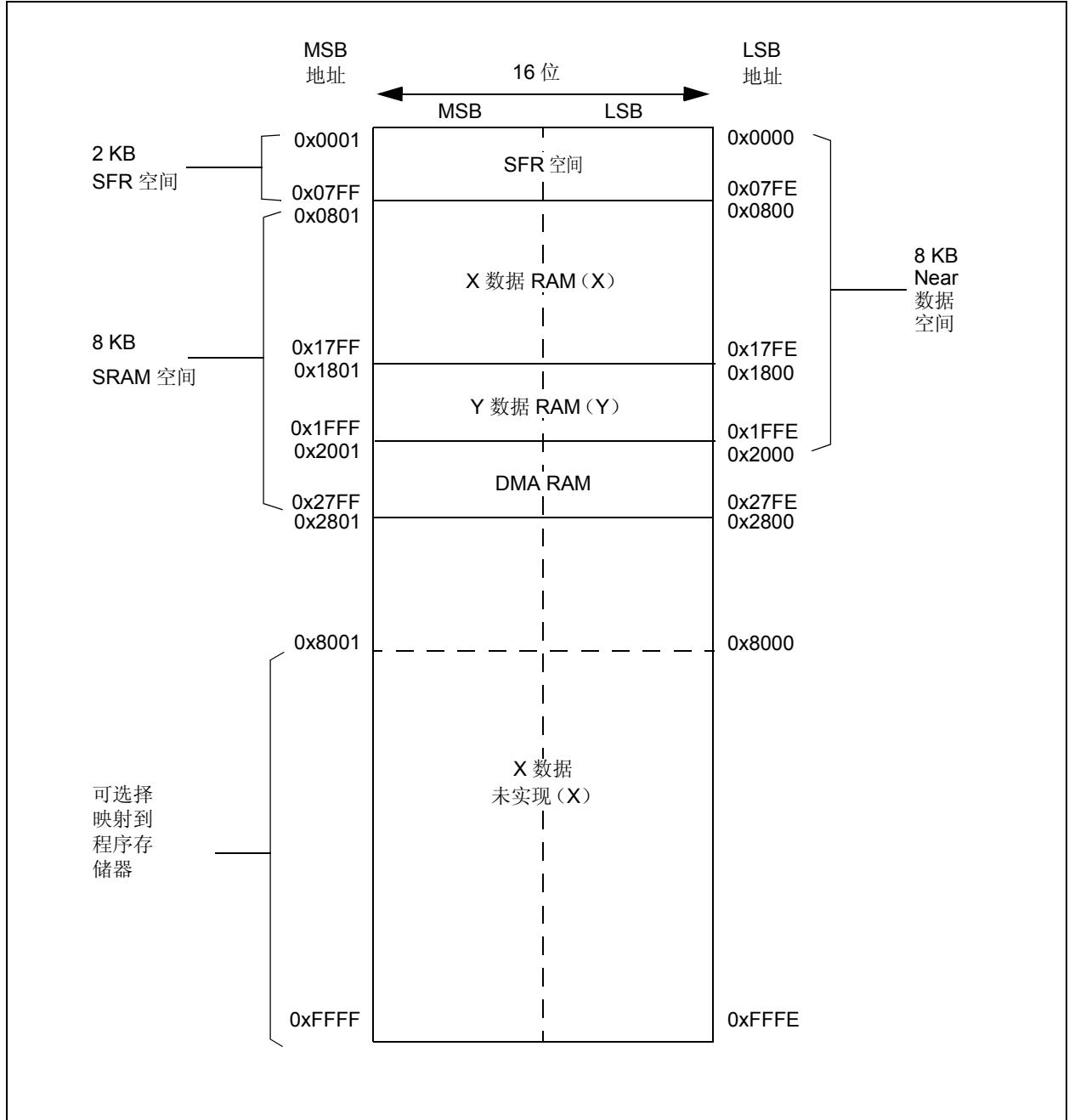
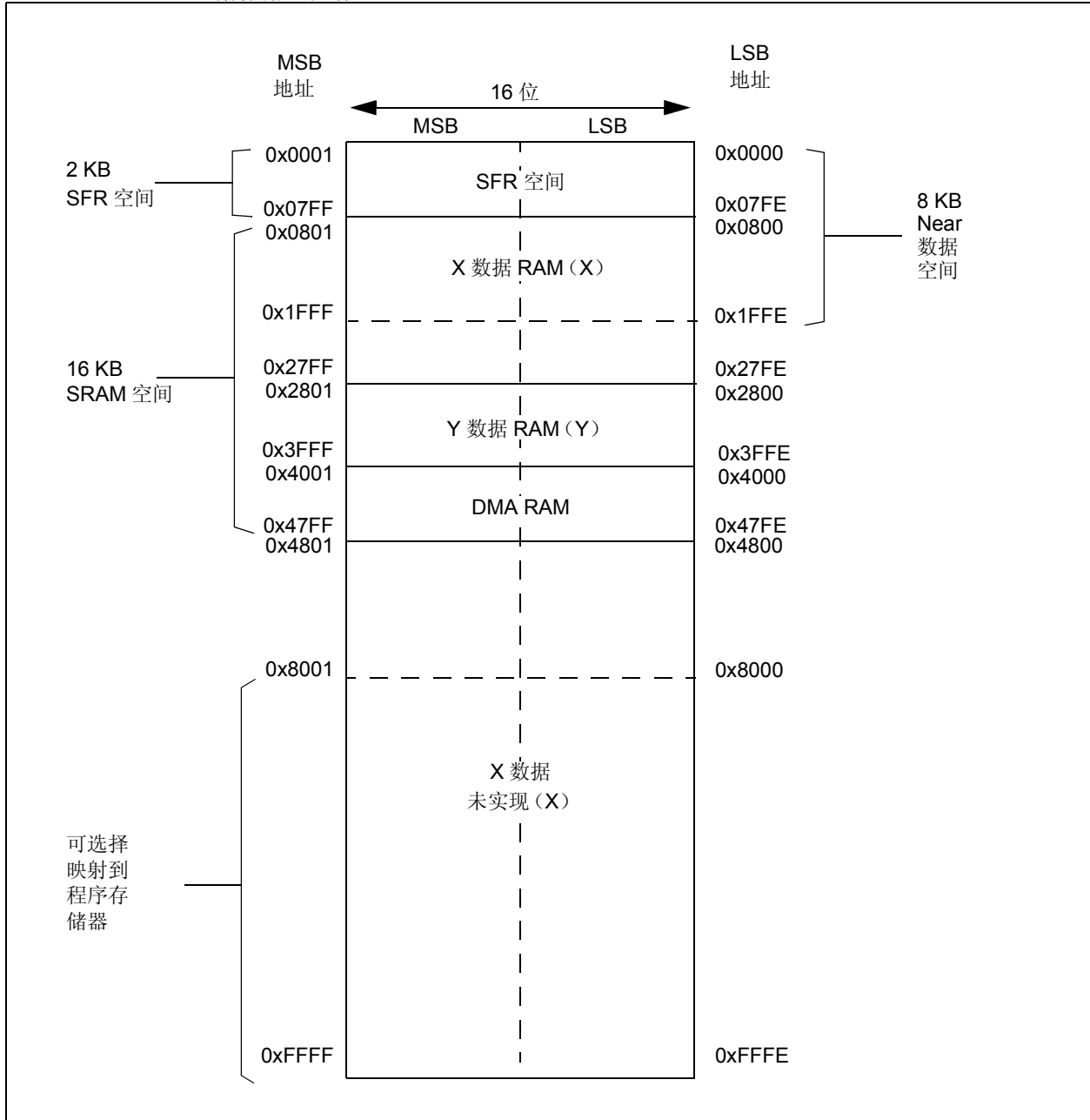


图 3-5: 带 16 KB RAM 的 dsPIC33FJ128MC802/804 和 dsPIC33FJ64MC802/804 器件的数据存储器映射



### 3.2.5 X 和 Y 数据空间

内核有两个数据空间 X 和 Y。这些数据空间可以看作是独立的（对于一些 DSP 指令），或者看作是统一的线性地址范围（对于 MCU 指令）。使用两个地址发生单元（AGU）和独立的数据总线来访问这两个数据空间。此特性允许某些指令同时从 RAM 中取两个字，因此提高了某些 DSP 算法的执行效率，如有限冲激响应（Finite Impulse Response, FIR）滤波算法和快速傅立叶变换（Fast Fourier Transform, FFT）。

X 数据空间可用于所有指令，并且支持所有寻址模式。X 数据空间的读 / 写数据总线相互独立。所有将数据空间视为组合的 X 和 Y 地址空间的指令均将 X 读数据总线作为读数据路径。X 读数据总线也可作为双操作数 DSP 指令（MAC 类）的 X 数据预取路径。

MAC 类指令（CLR、ED、EDAC、MAC、MOVSAC、MPY、MPY.N 和 MSC）同时使用 Y 数据空间与 X 数据空间，从而提供两条可同时对数据进行读操作的路径。

X 和 Y 数据空间都支持所有指令的模寻址，但要受到寻址模式的限制。位反转寻址模式只是在写 X 数据空间时才支持。

所有数据存储寄存器写操作（包括 DSP 指令中的数据存储寄存器写操作）均把数据空间视为组合的 X 和 Y 地址空间。X 和 Y 数据空间的分界取决于具体的器件，且不能由用户编程。

所有有效地址均为 16 位宽并且指向数据空间内的字节。因此，数据空间地址范围为 64 KB 或 32K 字，尽管不同器件上实际实现的存储单元有所不同。

### 3.2.6 DMA RAM

每个 dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件包含最大 2 KB 的双端口 DMA RAM，位于 Y 数据空间的末端。DMA RAM 空间中的存储单元可被 CPU 和 DMA 控制器模块同时访问。DMA 控制器使用 DMA RAM 来存储要使用 DMA 传输到各外设的数据以及使用 DMA 从各外设传输进来的数据。DMA 控制器对 DMA RAM 的访问无须占用 CPU 周期。

当 CPU 和 DMA 控制器尝试同时写同一个 DMA RAM 存储单元时，硬件确保 CPU 具有对 DMA RAM 存储单元的优先访问权。因此，DMA RAM 提供了传输 DMA 数据的可靠方式，而无须暂停 CPU。

<b>注：</b> 在不需要使用 DMA 功能的应用中，DMA RAM 可作为通用数据存储区使用。
---

表 3-1: CPU 内核寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态	
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	堆栈指针限制寄存器																xxxx	
ACCAL	0022	ACCAL																xxxx	
ACCAH	0024	ACCAH																xxxx	
ACCAU	0026	ACCA<39>								ACCAU								xxxx	
ACCBL	0028	ACCBL																xxxx	
ACCBH	002A	ACCBH																xxxx	
ACCBU	002C	ACCB<39>								ACCBU								xxxx	
PCL	002E	程序计数器低位寄存器																xxxx	
PCH	0030	—	—	—	—	—	—	—	—	程序计数器高字节寄存器								0000	
TBLPAG	0032	—	—	—	—	—	—	—	—	表页地址指针寄存器								0000	
PSVPAG	0034	—	—	—	—	—	—	—	—	程序存储器可视性页地址指针寄存器								0000	
RCOUNT	0036	Repeat 循环计数器寄存器																xxxx	
DCOUNT	0038	DCOUNT<15:0>																xxxx	
DOSTARTL	003A	DOSTARTL<15:1>																0	xxxx
DOSTARTH	003C	—	—	—	—	—	—	—	—	—	—	DOSTARTH<5:0>					00xx		
DOENDL	003E	DOENDL<15:1>																0	xxxx
DOENDH	0040	—	—	—	—	—	—	—	—	—	DOENDH								00xx
SR	0042	OA	OB	SA	SB	OAB	SAB	DA	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000	
CORCON	0044	—	—	—	US	EDT	DL<2:0>			SATA	SATB	SATDW	ACCSAT	IPL3	PSV	RND	IF	0000	
MODCON	0046	XMODEN	YMODEN	—	—	BWM<3:0>				YWM<3:0>				XWM<3:0>				0000	

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以 16 进制表示。



表 3-1: CPU 内核寄存器映射 (续)

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态	
XMODSRT	0048	XS<15:1>															0	xxxx	
XMODEND	004A	XE<15:1>															1	xxxx	
YMODSRT	004C	YS<15:1>															0	xxxx	
YMODEND	004E	YE<15:1>															1	xxxx	
XBREV	0050	BREN	XB<14:0>															xxxx	
DISICNT	0052	—	—	禁止中断计数器寄存器															xxxx

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以 16 进制表示。

表 3-2: dsPIC33FJ128MC202/802、dsPIC33FJ64MC202/802 和 dsPIC33FJ32MC302 的电平变化通知寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
CNEN1	0060	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	—	—	—	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	00C2	—	CN30IE	CN29IE	—	CN27IE	—	—	CN24IE	CN23IE	CN22IE	CN21IE	—	—	—	—	CN16IE	0000
CNPU1	0068	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	—	—	—	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	006A	—	CN30PUE	CN29PUE	—	CN27PUE	—	—	CN24PUE	CN23PUE	CN22PUE	CN21PUE	—	—	—	—	CN16PUE	0000

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-3: dsPIC33FJ128MC204/804、dsPIC33FJ64MC204/804 和 dsPIC33FJ32MC304 的电平变化通知寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
CNEN1	0060	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	00C2	—	CN30IE	CN29IE	CN28IE	CN27IE	CN26IE	CN25IE	CN24IE	CN23IE	CN22IE	CN21IE	CN20IE	CN19IE	CN18IE	CN17IE	CN16IE	0000
CNPU1	0068	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	006A	—	CN30PUE	CN29PUE	CN28PUE	CN27PUE	CN26PUE	CN25PUE	CN24PUE	CN23PUE	CN22PUE	CN21PUE	CN20PUE	CN19PUE	CN18PUE	CN17PUE	CN16PUE	0000

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-4: 中断控制器寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
INTCON1	0080	NSTDIS	OVAERR	OVBERR	COVAERR	COVBERR	OVATE	OVBTE	COVTE	SFTACERR	DIV0ERR	DMACERR	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	DMA1IF	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	T3IF	T2IF	OC2IF	IC2IF	DMA0IF	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	DMA2IF	IC8IF	IC7IF	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF	0000
IFS2	0088	—	DMA4IF	PMPIF	—	—	—	—	—	—	—	—	DMA3IF	C1IF <sup>(1)</sup>	C1RXIF <sup>(1)</sup>	SPI2IF	SPI2EIF	0000
IFS3	008A	FLTA1IF	RTCIF	DMA5IF	—	—	QE11IF	PWM1IF	—	—	—	—	—	—	—	—	—	0000
IFS4	008C	DAC1LIF <sup>(2)</sup>	DAC1RIF <sup>(2)</sup>	—	—	QE12IF	FLTA2IF	PWM2IF	—	—	C1TXIF <sup>(1)</sup>	DMA7IF	DMA6IF	CRCIF	U2EIF	U1EIF	—	0000
IEC0	0094	—	DMA1IE	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	T3IE	T2IE	OC2IE	IC2IE	DMA0IE	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	DMA2IE	IC8IE	IC7IE	—	INT1IE	CNIE	CMIE	MI2C1IE	SI2C1IE	0000
IEC2	0098	—	DMA4IE	PMPIE	—	—	—	—	—	—	—	—	DMA3IE	C1IE <sup>(1)</sup>	C1RXIE <sup>(1)</sup>	SPI2IE	SPI2EIE	0000
IEC3	009A	FLTA1IE	RTCIE	DMA5IE	—	—	QE11IE	PWM1IE	—	—	—	—	—	—	—	—	—	0000
IEC4	009C	DAC1LIE <sup>(2)</sup>	DAC1RIE <sup>(2)</sup>	—	—	QE12IE	FLTA2IE	PWM2IE	—	—	C1TXIE <sup>(1)</sup>	DMA7IE	DMA6IE	CRCIE	U2EIE	U1EIE	—	0000
IPC0	00A4	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			—	INT0IP<2:0>			4444
IPC1	00A6	—	T2IP<2:0>			—	OC2IP<2:0>			—	IC2IP<2:0>			—	DMA0IP<2:0>			4444
IPC2	00A8	—	U1RXIP<2:0>			—	SPI1IP<2:0>			—	SPI1EIP<2:0>			—	T3IP<2:0>			4444
IPC3	00AA	—	—	—	—	—	DMA1IP<2:0>			—	AD1IP<2:0>			—	U1TXIP<2:0>			0444
IPC4	00AC	—	CNIP<2:0>			—	CMIP<2:0>			—	MI2C1IP<2:0>			—	SI2C1IP<2:0>			4444
IPC5	00AE	—	IC8IP<2:0>			—	IC7IP<2:0>			—	—	—	—	—	INT1IP<2:0>			4404
IPC6	00B0	—	T4IP<2:0>			—	OC4IP<2:0>			—	OC3IP<2:0>			—	DMA2IP<2:0>			4444
IPC7	00B2	—	U2TXIP<2:0>			—	U2RXIP<2:0>			—	INT2IP<2:0>			—	T5IP<2:0>			4444
IPC8	00B4	—	C1IP<2:0> <sup>(1)</sup>			—	C1RXIP<2:0> <sup>(1)</sup>			—	SPI2IP<2:0>			—	SPI2EIP<2:0>			4444
IPC9	00B6	—	—	—	—	—	—	—	—	—	—	—	—	—	DMA3IP<2:0>			0004
IPC11	00BA	—	—	—	—	—	DMA4IP<2:0>			—	PMPIP<2:0>			—	—	—	—	0440
IPC14	00C0	—	—	—	—	—	QE11IP<2:0>			—	PWM1IP<2:0>			—	—	—	—	0440
IPC15	00C2	—	FLTA1IP<2:0>			—	RTCIP<2:0>			—	DMA5IP<2:0>			—	—	—	—	4440
IPC16	00C4	—	CRCIP<2:0>			—	U2EIP<2:0>			—	U1EIP<2:0>			—	—	—	—	4440
IPC17	00C6	—	—	—	—	—	C1TXIP<2:0> <sup>(1)</sup>			—	DMA7IP<2:0>			—	DMA6IP<2:0>			0444
IPC18	00C8	—	QE12IP<2:0>			—	FLTA2IP<2:0>			—	PWM2IP<2:0>			—	—	—	—	4440
IPC19	00CA	—	DAC1LIP<2:0> <sup>(2)</sup>			—	DAC1RIP<2:0> <sup>(2)</sup>			—	—	—	—	—	—	—	—	4400
INTTREG	00E0	—	—	—	—	ILR<3:0>>				—	VECNUM<6:0>						4444	

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

- 注 1: 对于不带 ECAN™ 模块的器件禁止此类中断。  
 注 2: 对于不带 DAC 模块的器件禁止此类中断。

表 3-5: 定时器寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
TMR1	0100	Timer1 寄存器																xxxx
PR1	0102	周期寄存器 1																FFFF
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>	—	TSYNC	TCS	—	0000	
TMR2	0106	Timer2 寄存器																xxxx
TMR3HLD	0108	Timer3 保持寄存器 (仅适用于 32 位定时器操作)																xxxx
TMR3	010A	Timer3 寄存器																xxxx
PR2	010C	周期寄存器 2																FFFF
PR3	010E	周期寄存器 3																FFFF
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>	T32	—	TCS	—	0000	
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>	—	—	TCS	—	0000	
TMR4	0114	Timer4 寄存器																xxxx
TMR5HLD	0116	Timer5 保持寄存器 (仅适用于 32 位定时器操作)																xxxx
TMR5	0118	Timer5 寄存器																xxxx
PR4	011A	周期寄存器 4																FFFF
PR5	01C2	周期寄存器 5																FFFF
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>	T32	—	TCS	—	0000	
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>	—	—	TCS	—	0000	

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-6: 输入捕捉寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
IC1BUF	0140	输入捕捉 1 寄存器																xxxx
IC1CON	0142	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>			0000	
IC2BUF	0144	输入捕捉 2 寄存器																xxxx
IC2CON	0146	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>			0000	
IC7BUF	0158	输入捕捉 7 寄存器																xxxx
IC7CON	015A	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>			0000	
IC8BUF	015C	输入捕捉 8 寄存器																xxxx
IC8CON	015E	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>			0000	

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-7: 输出比较寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
OC1RS	0180	输出比较 1 辅助寄存器																xxxx
OC1R	0182	输出比较 1 寄存器																xxxx
OC1CON	0184	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC2RS	0186	输出比较 2 辅助寄存器																xxxx
OC2R	0188	输出比较 2 寄存器																xxxx
OC2CON	018A	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC3RS	018C	输出比较 3 辅助寄存器																xxxx
OC3R	018E	输出比较 3 寄存器																xxxx
OC3CON	0190	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	
OC4RS	0192	输出比较 4 辅助寄存器																xxxx
OC4R	0194	输出比较 4 寄存器																xxxx
OC4CON	0196	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>		0000	

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-8: 6 路输出 PWM1 寄存器映射

SFR 名称	SFR 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	复位时的状态
P1TCON	01C0	PTEN	—	PTSIDL	—	—	—	—	—	PTOPS<3:0>			PTCKPS<1:0>		PTMOD<1:0>		0000	
P1TMR	01C2	PTDIR	PWM 定时器计数值寄存器															0000
P1TPER	01C4	—	PWM 时基周期寄存器															0000
P1SECMP	01C6	SEVTDIR	PWM 特殊事件比较寄存器															0000
PWM1CON1	01C8	—	—	—	—	—	PMOD3	PMOD2	PMOD1	—	PEN3H	PEN2H	PEN1H	—	PEN3L	PEN2L	PEN1L	00FF
PWM1CON2	01CA	—	—	—	—	SEVOPS<3:0>			—	—	—	—	—	IUE	OSYNC	UDIS	0000	
P1DTCON1	01CC	DTBPS<1:0>		DTB<5:0>				DTAPS<1:0>		DTA<5:0>				0000				
P1DTCON2	01CE	—	—	—	—	—	—	—	—	—	—	DTS3A	DTS3I	DTS2A	DTS2I	DTS1A	DTS1I	0000
P1FLTACON	01D0	—	—	FAOV3H	FAOV3L	FAOV2H	FAOV2L	FAOV1H	FAOV1L	FLTAM	—	—	—	FAEN<2:0>			0000	
P1OVDCON	01D4	—	—	POVD3H	POVD3L	POVD2H	POVD2L	POVD1H	POVD1L	—	—	POUT3H	POUT3L	POUT2H	POUT2L	POUT1H	POUT1L	FF00
P1DC1	01D6	PWM 占空比 1 寄存器																0000
P1DC2	01D8	PWM 占空比 2 寄存器																0000
P1DC3	01DA	PWM 占空比 3 寄存器																0000

图注: x = 未初始化位, — = 未实现, 读为 0。

表 3-9: 2 路输出 PWM2 寄存器映射

SFR 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	复位时的状态	
P2TCON	05C0	PTEN	—	PTSIDL	—	—	—	—	—	PTOPS<3:0>			PTCKPS<1:0>		PTMOD<1:0>		0000		
P2TMR	05C2	PTDIR	PWM 定时器计数值寄存器																0000
P2TPER	05C4	—	PWM 时基周期寄存器																0000
P2SECMP	05C6	SEVTDIR	PWM 特殊事件比较寄存器																0000
PWM2CON1	05C8	—	—	—	—	—	—	—	—	PMOD1	—	—	—	PEN1H	—	—	—	PEN1L	00FF
PWM2CON2	05CA	—	—	—	—	SEVOPS<3:0>			—	—	—	—	—	IUE	OSYNC	UDIS	0000		
P2DTCON1	05CC	DTBPS<1:0>		DTB<5:0>					DTAPS<1:0>		DTA<5:0>						0000		
P2DTCON2	05CE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTS1A	DTS1I	0000	
P2FLTACON	05D0	—	—	—	—	—	—	—	FAOV1H	FAOV1L	FLTAM	—	—	—	—	—	—	FAEN1	0000
P2OVDCON	05D4	—	—	—	—	—	—	—	POVD1H	POVD1L	—	—	—	—	—	—	POUT1H	POUT1L	FF00
P2DC1	05D6	PWM 占空比 1 寄存器																0000	

图注: u = 未初始化位, — = 未实现, 读为 0。

表 3-10: QE1 寄存器映射

SFR 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	复位时的状态
QE1CON	01E0	CNTERR	—	QEISIDL	INDX	UPDN	QEIM<2:0>		SWPAB	PCDOUT	TQGATE	TQCKPS<1:0>		POSRES	TQCS	UPDN_SRC	0000	
DFLT1CON	01E2	—	—	—	—	—	IMV<1:0>	CEID	QEOUT	QECK<2:0>			—	—	—	—	0000	
POS1CNT	01E4	位置计数器 <15:0>																0000
MAX1CNT	01E6	最大计数 <15:0>																FFFF

图注: u = 未初始化位, — = 未实现, 读为 0。

表 3-11: QE2 寄存器映射

SFR 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	复位时的状态
QE2CON	01F0	CNTERR	—	QEISIDL	INDX	UPDN	QEIM<2:0>		SWPAB	PCDOUT	TQGATE	TQCKPS<1:0>		POSRES	TQCS	UPDN_SRC	0000	
DFLT2CON	01F2	—	—	—	—	—	IMV<1:0>	CEID	QEOUT	QECK<2:0>			—	—	—	—	0000	
POS2CNT	01F4	位置计数器 <15:0>																0000
MAX2CNT	01F6	最大计数 <15:0>																FFFF

图注: u = 未初始化位, — = 未实现, 读为 0。

表 3-12: I2C 寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
I2C1RCV	0200	—	—	—	—	—	—	—	—	接收寄存器								0000
I2C1TRN	0202	—	—	—	—	—	—	—	—	发送寄存器								00FF
I2C1BRG	0204	—	—	—	—	—	—	—	波特率发生器寄存器								0000	
I2C1CON	0206	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C1STAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF	0000
I2C1ADD	020A	—	—	—	—	—	—	地址寄存器								0000		
I2C1MSK	020C	—	—	—	—	—	—	地址掩码寄存器								0000		

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-13: UART1 寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
U1MODE	0220	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	URXINV	BRGH	PDSEL<1:0>		STSEL	0000
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U1TXREG	0224	—	—	—	—	—	—	—	UTX8	UART 发送寄存器								xxxxx
U1RXREG	0226	—	—	—	—	—	—	—	URX8	UART 接收寄存器								0000
U1BRG	0228	波特率发生器预分频器															0000	

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-14: UART2 寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
U2MODE	0230	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	URXINV	BRGH	PDSEL<1:0>		STSEL	0000
U2STA	0232	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U2TXREG	0234	—	—	—	—	—	—	—	UTX8	UART 发送寄存器								xxxxx
U2RXREG	0236	—	—	—	—	—	—	—	URX8	UART 接收寄存器								0000
U2BRG	0238	波特率发生器预分频器															0000	

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-15: SPI1 寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
SPI1STAT	0240	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000
SPI1CON1	0242	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE<2:0>		PPRE<1:0>		0000	
SPI1CON2	0244	FRMEN	SPIFSD	FRMPOL	—	—	—	—	—	—	—	—	—	—	—	FRMDLY	—	0000
SPI1BUF	0248	SPI1 发送和接收缓冲寄存器																0000

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-16: SPI2 寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
SPI2STAT	0260	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000
SPI2CON1	0262	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE<2:0>		PPRE<1:0>		0000	
SPI2CON2	0264	FRMEN	SPIFSD	FRMPOL	—	—	—	—	—	—	—	—	—	—	—	FRMDLY	—	0000
SPI2BUF	0268	SPI2 发送和接收缓冲寄存器																0000

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-17: dsPIC33FJ64MC202/802、dsPIC33FJ128MC202/802 和 dsPIC33FJ32MC302 的 ADC1 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
ADC1BUF0	0300	ADC 数据缓冲区 0																xxxx
AD1CON1	0320	ADON	—	ADSIDL	ADDMABM	—	AD12B	FORM<1:0>		SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE	0000
AD1CON2	0322	VCFG<2:0>		—	—	CSCNA	CHPS<1:0>		BUFS	—	SMPI<3:0>				BUFM	ALTS	0000	
AD1CON3	0324	ADRC	—	—	SAMC<4:0>				ADCS<7:0>							0000		
AD1CHS123	0326	—	—	—	—	—	CH123NB<1:0>		CH123SB	—	—	—	—	—	CH123NA<1:0>		CH123SA	0000
AD1CHS0	0328	CH0NB	—	—	CH0SB<4:0>				CH0NA	—	—	CH0SA<4:0>						0000
AD1PCFGL	032C	—	—	—	—	—	—	—	—	—	—	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSL	0330	—	—	—	—	—	—	—	—	—	—	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0	0000
AD1CON4	0332	—	—	—	—	—	—	—	—	—	—	—	—	—	DMABL<2:0>		0000	

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-18: dsPIC33FJ64MC204/804、dsPIC33FJ128MC204/804 和 dsPIC33FJ32MC304 的 ADC1 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
ADC1BUF0	0300	ADC 数据缓冲区 0																xxxxx
AD1CON1	0320	ADON	—	ADSIDL	ADDMABM	—	AD12B	FORM<1:0>		SSRC<2:0>		—	SIMSAM	ASAM	SAMP	DONE	0000	
AD1CON2	0322	VCFG<2:0>		—	—	CSCNA	CHPS<1:0>		BUFS	—	SMPI<3:0>			BUFM	ALTS	0000		
AD1CON3	0324	ADRC	—	—	SAMC<4:0>				ADCS<7:0>							0000		
AD1CHS123	0326	—	—	—	—	—	CH123NB<1:0>		CH123SB	—	—	—	—	—	CH123NA<1:0>		CH123SA	0000
AD1CHS0	0328	CH0NB	—	—	CH0SB<4:0>				CH0NA	—	—	CH0SA<4:0>						0000
AD1PCFGL	032C	—	—	—	—	—	—	—	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSL	0330	—	—	—	—	—	—	—	CSS8	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0	0000
AD1CON4	0332	—	—	—	—	—	—	—	—	—	—	—	—	—	DMABL<2:0>		0000	

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-19: dsPIC33FJ128MC804 和 dsPIC33FJ64MC804 的 DAC1 寄存器映射

SFR 名称	SFR 地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
DAC1CON	03F0	DACEN	—	DACSIDL	AMPON	—	—	—	FORM	—	DACFDIV<6:0>						0000	
DAC1STAT	03F2	LOEN	—	LMVOEN	—	—	LITYPE	LFULL	LEMPY	ROEN	—	RMVOEN	—	—	RITYPE	RFULL	REMPY	0000
DAC1DFLT	03F4	DAC1DFLT<15:0>																0000
DAC1RDAT	03F6	DAC1RDAT<15:0>																0000
DAC1LDAT	03F8	DAC1LDAT<15:0>																0000

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。



**表 3-20: 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	所有复位时的状态	
DMA0CON	0380	CHEN	SIZE	DIR	HALF	NULLW	—	—	—	—	—	AMODE<1:0>		—	—	MODE<1:0>		0000	
DMA0REQ	0382	FORCE	—	—	—	—	—	—	—	—	IRQSEL<6:0>								0000
DMA0STA	0384	STA<15:0>																0000	
DMA0STB	0386	STB<15:0>																0000	
DMA0PAD	0388	PAD<15:0>																0000	
DMA0CNT	038A	—	—	—	—	—	—	CNT<9:0>										0000	
DMA1CON	038C	CHEN	SIZE	DIR	HALF	NULLW	—	—	—	—	—	AMODE<1:0>		—	—	MODE<1:0>		0000	
DMA1REQ	038E	FORCE	—	—	—	—	—	—	—	—	IRQSEL<6:0>								0000
DMA1STA	0390	STA<15:0>																0000	
DMA1STB	0392	STB<15:0>																0000	
DMA1PAD	0394	PAD<15:0>																0000	
DMA1CNT	0396	—	—	—	—	—	—	CNT<9:0>										0000	
DMA2CON	0398	CHEN	SIZE	DIR	HALF	NULLW	—	—	—	—	—	AMODE<1:0>		—	—	MODE<1:0>		0000	
DMA2REQ	039A	FORCE	—	—	—	—	—	—	—	—	IRQSEL<6:0>								0000
DMA2STA	039C	STA<15:0>																0000	
DMA2STB	039E	STB<15:0>																0000	
DMA2PAD	03A0	PAD<15:0>																0000	
DMA2CNT	03A2	—	—	—	—	—	—	CNT<9:0>										0000	
DMA3CON	03A4	CHEN	SIZE	DIR	HALF	NULLW	—	—	—	—	—	AMODE<1:0>		—	—	MODE<1:0>		0000	
DMA3REQ	03A6	FORCE	—	—	—	—	—	—	—	—	IRQSEL<6:0>								0000
DMA3STA	03A8	STA<15:0>																0000	
DMA3STB	03AA	STB<15:0>																0000	
DMA3PAD	03AC	PAD<15:0>																0000	
DMA3CNT	03AE	—	—	—	—	—	—	CNT<9:0>										0000	
DMA4CON	03B0	CHEN	SIZE	DIR	HALF	NULLW	—	—	—	—	—	AMODE<1:0>		—	—	MODE<1:0>		0000	
DMA4REQ	03B2	FORCE	—	—	—	—	—	—	—	—	IRQSEL<6:0>								0000
DMA4STA	03B4	STA<15:0>																0000	
DMA4STB	03B6	STB<15:0>																0000	
DMA4PAD	03B8	PAD<15:0>																0000	
DMA4CNT	03BA	—	—	—	—	—	—	CNT<9:0>										0000	
DMA5CON	03BC	CHEN	SIZE	DIR	HALF	NULLW	—	—	—	—	—	AMODE<1:0>		—	—	MODE<1:0>		0000	
DMA5REQ	03BE	FORCE	—	—	—	—	—	—	—	—	IRQSEL<6:0>								0000
DMA5STA	03C0	STA<15:0>																0000	
DMA5STB	03C2	STB<15:0>																0000	

图注: — = 未实现, 读为 0。复位值以十六进制表示。

**表 3-20: DMA 寄存器映射 (续)**

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态	
DMA5PAD	03C4	PAD<15:0>																0000	
DMA5CNT	03C6	—	—	—	—	—	—	CNT<9:0>										0000	
DMA6CON	03C8	CHEN	SIZE	DIR	HALF	NULLW	—	—	—	—	—	AMODE<1:0>		—	—	MODE<1:0>		0000	
DMA6REQ	03CA	FORCE	—	—	—	—	—	—	—	—	IRQSEL<6:0>								0000
DMA6STA	03CC	STA<15:0>																0000	
DMA6STB	03CE	STB<15:0>																0000	
DMA6PAD	03D0	PAD<15:0>																0000	
DMA6CNT	03D2	—	—	—	—	—	—	CNT<9:0>										0000	
DMA7CON	03D4	CHEN	SIZE	DIR	HALF	NULLW	—	—	—	—	—	AMODE<1:0>		—	—	MODE<1:0>		0000	
DMA7REQ	03D6	FORCE	—	—	—	—	—	—	—	—	IRQSEL<6:0>								0000
DMA7STA	03D8	STA<15:0>																0000	
DMA7STB	03DA	STB<15:0>																0000	
DMA7PAD	03DC	PAD<15:0>																0000	
DMA7CNT	03DE	—	—	—	—	—	—	CNT<9:0>										0000	
DMACS0	03E0	PWCOL7	PWCOL6	PWCOL5	PWCOL4	PWCOL3	PWCOL2	PWCOL1	PWCOL0	XWCOL7	XWCOL6	XWCOL5	XWCOL4	XWCOL3	XWCOL2	XWCOL1	XWCOL0	0000	
DMACS1	03E2	—	—	—	—	LSTCH<3:0>				PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0	0000	
DSADR	03E4	DSADR<15:0>																0000	

图注: — = 未实现, 读为 0。复位值以十六进制表示。

表 3-21: 当 C1CTRL1.WIN = 0 或 1 时的 ECAN1 寄存器映射 (对于 dsPIC33FJ128MC802/804 和 dsPIC33FJ64MC802/804)

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
C1CTRL1	0400	—	—	CSIDL	ABAT	CANCK S	REQOP<2:0>			OPMODE<2:0>			—	CANCAP	—	—	WIN	0480
C1CTRL2	0402	—	—	—	—	—	—	—	—	—	—	—	DNCNT<4:0>				0000	
C1VEC	0404	—	—	—	FILHIT<4:0>					—	ICODE<6:0>					0000		
C1FCTRL	0406	DMABS<2:0>			—	—	—	—	—	—	—	—	FSA<4:0>					0000
C1FIFO	0408	—	—	FBP<5:0>						—	—	FNRB<5:0>					0000	
C1INTF	040A	—	—	TXBO	TXBP	RXBP	TXWAR	RXWAR	EWARN	IVRIF	WAKIF	ERRIF	—	FIFOIF	RBOVIF	RBIF	TBIF	0000
C1INTE	040C	—	—	—	—	—	—	—	—	IVRIE	WAKIE	ERRIE	—	FIFOIE	RBOVIE	RBIE	TBIE	0000
C1EC	040E	TERRCNT<7:0>						RERRCNT<7:0>						0000				
C1CFG1	0410	—	—	—	—	—	—	—	—	SJW<1:0>		BRP<5:0>					0000	
C1CFG2	0412	—	WAKFIL	—	—	—	SEG2PH<2:0>			SEG2PHTS	SAM	SEG1PH<2:0>			PRSEG<2:0>		0000	
C1FEN1	0414	FLTEN15	FLTEN14	FLTEN13	FLTEN12	FLTEN11	FLTEN10	FLTEN9	FLTEN8	FLTEN7	FLTEN6	FLTEN5	FLTEN4	FLTEN3	FLTEN2	FLTEN1	FLTEN0	FFFF
C1FMSKSEL1	0418	F7MSK<1:0>		F6MSK<1:0>		F5MSK<1:0>		F4MSK<1:0>		F3MSK<1:0>		F2MSK<1:0>		F1MSK<1:0>		F0MSK<1:0>		0000
C1FMSKSEL2	041A	F15MSK<1:0>		F14MSK<1:0>		F13MSK<1:0>		F12MSK<1:0>		F11MSK<1:0>		F10MSK<1:0>		F9MSK<1:0>		F8MSK<1:0>		0000

图注: — = 未实现, 读为 0。复位值以十六进制表示。

表 3-22: 当 C1CTRL1.WIN = 0 时的 ECAN1 寄存器映射 (对于 dsPIC33FJ128MC802/804 和 dsPIC33FJ64MC802/804)

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
	0400-041E	参见 WIN = x 时的定义																
C1RXFUL1	0420	RXFUL15	RXFUL14	RXFUL13	RXFUL12	RXFUL11	RXFUL10	RXFUL9	RXFUL8	RXFUL7	RXFUL6	RXFUL5	RXFUL4	RXFUL3	RXFUL2	RXFUL1	RXFUL0	0000
C1RXFUL2	0422	RXFUL31	RXFUL30	RXFUL29	RXFUL28	RXFUL27	RXFUL26	RXFUL25	RXFUL24	RXFUL23	RXFUL22	RXFUL21	RXFUL20	RXFUL19	RXFUL18	RXFUL17	RXFUL16	0000
C1RXOVF1	0428	RXOVF15	RXOVF14	RXOVF13	RXOVF12	RXOVF11	RXOVF10	RXOVF9	RXOVF8	RXOVF7	RXOVF6	RXOVF5	RXOVF4	RXOVF3	RXOVF2	RXOVF1	RXOVF0	0000
C1RXOVF2	042A	RXOVF31	RXOVF30	RXOVF29	RXOVF28	RXOVF27	RXOVF26	RXOVF25	RXOVF24	RXOVF23	RXOVF22	RXOVF21	RXOVF20	RXOVF19	RXOVF18	RXOVF17	RXOVF16	0000
C1TR01CON	0430	TXEN1	TXABT1	TXLARB1	TXERR1	TXREQ1	RTREN1	TX1PRI<1:0>		TXEN0	TXABT0	TXLARB0	TXERR0	TXREQ0	RTREN0	TX0PRI<1:0>		0000
C1TR23CON	0432	TXEN3	TXABT3	TXLARB3	TXERR3	TXREQ3	RTREN3	TX3PRI<1:0>		TXEN2	TXABT2	TXLARB2	TXERR2	TXREQ2	RTREN2	TX2PRI<1:0>		0000
C1TR45CON	0434	TXEN5	TXABT5	TXLARB5	TXERR5	TXREQ5	RTREN5	TX5PRI<1:0>		TXEN4	TXABT4	TXLARB4	TXERR4	TXREQ4	RTREN4	TX4PRI<1:0>		0000
C1TR67CON	0436	TXEN7	TXABT7	TXLARB7	TXERR7	TXREQ7	RTREN7	TX7PRI<1:0>		TXEN6	TXABT6	TXLARB6	TXERR6	TXREQ6	RTREN6	TX6PRI<1:0>		0000
C1RXD	0440	接收的数据字																xxxx
C1TXD	0442	发送的数据字																xxxx

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-23: 当 C1CTRL1.WIN = 1 时的 ECAN1 寄存器映射 (对于 dsPIC33FJ128MC802/804 和 dsPIC33FJ64MC802/804)

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
	0400-041E	参见 WIN = x 时的定义																
C1BUFNPNT1	0420	F3BP<3:0>			F2BP<3:0>			F1BP<3:0>			F0BP<3:0>						0000	
C1BUFNPNT2	0422	F7BP<3:0>			F6BP<3:0>			F5BP<3:0>			F4BP<3:0>						0000	
C1BUFNPNT3	0424	F11BP<3:0>			F10BP<3:0>			F9BP<3:0>			F8BP<3:0>						0000	
C1BUFNPNT4	0426	F15BP<3:0>			F14BP<3:0>			F13BP<3:0>			F12BP<3:0>						0000	
C1RXM0SID	0430	SID<10:3>				SID<2:0>				MIDE		EID<17:16>					xxxx	
C1RXM0EID	0432	EID<15:8>				EID<7:0>											xxxx	
C1RXM1SID	0434	SID<10:3>				SID<2:0>				MIDE		EID<17:16>					xxxx	
C1RXM1EID	0436	EID<15:8>				EID<7:0>											xxxx	
C1RXM2SID	0438	SID<10:3>				SID<2:0>				MIDE		EID<17:16>					xxxx	
C1RXM2EID	043A	EID<15:8>				EID<7:0>											xxxx	
C1RXF0SID	0440	SID<10:3>				SID<2:0>				EXIDE		EID<17:16>					xxxx	
C1RXF0EID	0442	EID<15:8>				EID<7:0>											xxxx	
C1RXF1SID	0444	SID<10:3>				SID<2:0>				EXIDE		EID<17:16>					xxxx	
C1RXF1EID	0446	EID<15:8>				EID<7:0>											xxxx	
C1RXF2SID	0448	SID<10:3>				SID<2:0>				EXIDE		EID<17:16>					xxxx	
C1RXF2EID	044A	EID<15:8>				EID<7:0>											xxxx	
C1RXF3SID	044C	SID<10:3>				SID<2:0>				EXIDE		EID<17:16>					xxxx	
C1RXF3EID	044E	EID<15:8>				EID<7:0>											xxxx	
C1RXF4SID	0450	SID<10:3>				SID<2:0>				EXIDE		EID<17:16>					xxxx	
C1RXF4EID	0452	EID<15:8>				EID<7:0>											xxxx	
C1RXF5SID	0454	SID<10:3>				SID<2:0>				EXIDE		EID<17:16>					xxxx	
C1RXF5EID	0456	EID<15:8>				EID<7:0>											xxxx	
C1RXF6SID	0458	SID<10:3>				SID<2:0>				EXIDE		EID<17:16>					xxxx	
C1RXF6EID	045A	EID<15:8>				EID<7:0>											xxxx	
C1RXF7SID	045C	SID<10:3>				SID<2:0>				EXIDE		EID<17:16>					xxxx	
C1RXF7EID	045E	EID<15:8>				EID<7:0>											xxxx	
C1RXF8SID	0460	SID<10:3>				SID<2:0>				EXIDE		EID<17:16>					xxxx	
C1RXF8EID	0462	EID<15:8>				EID<7:0>											xxxx	
C1RXF9SID	0464	SID<10:3>				SID<2:0>				EXIDE		EID<17:16>					xxxx	
C1RXF9EID	0466	EID<15:8>				EID<7:0>											xxxx	
C1RXF10SID	0468	SID<10:3>				SID<2:0>				EXIDE		EID<17:16>					xxxx	
C1RXF10EID	046A	EID<15:8>				EID<7:0>											xxxx	
C1RXF11SID	046C	SID<10:3>				SID<2:0>				EXIDE		EID<17:16>					xxxx	

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-23: 当 C1CTRL1.WIN = 1 时的 ECAN1 寄存器映射 (对于 dsPIC33FJ128MC802/804 和 dsPIC33FJ64MC802/804) (续)

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
C1RXF11EID	046E	EID<15:8>						EID<7:0>								xxxxx		
C1RXF12SID	0470	SID<10:3>						SID<2:0>		—	EXIDE	—	EID<17:16>			xxxxx		
C1RXF12EID	0472	EID<15:8>						EID<7:0>								xxxxx		
C1RXF13SID	0474	SID<10:3>						SID<2:0>		—	EXIDE	—	EID<17:16>			xxxxx		
C1RXF13EID	0476	EID<15:8>						EID<7:0>								xxxxx		
C1RXF14SID	0478	SID<10:3>						SID<2:0>		—	EXIDE	—	EID<17:16>			xxxxx		
C1RXF14EID	047A	EID<15:8>						EID<7:0>								xxxxx		
C1RXF15SID	047C	SID<10:3>						SID<2:0>		—	EXIDE	—	EID<17:16>			xxxxx		
C1RXF15EID	047E	EID<15:8>						EID<7:0>								xxxxx		

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

**表 3-24: 外设引脚选择输入寄存器映射**

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
RPINR0	0680	—	—	—	INT1R<4:0>				—	—	—	—	—	—	—	—	—	1F00
RPINR1	0682	—	—	—	—	—	—	—	—	—	—	—	INT2R<4:0>				001F	
RPINR3	0686	—	—	—	T3CKR<4:0>				—	—	—	T2CKR<4:0>				1F1F		
RPINR4	0688	—	—	—	T5CKR<4:0>				—	—	—	T4CKR<4:0>				1F1F		
RPINR7	068E	—	—	—	IC2R<4:0>				—	—	—	IC1R<4:0>				1F1F		
RPINR10	0694	—	—	—	IC8R<4:0>				—	—	—	IC7R<4:0>				1F1F		
RPINR11	0696	—	—	—	—	—	—	—	—	—	—	—	OCFAR<4:0>				001F	
RPINR12	0698	—	—	—	—	—	—	—	—	—	—	—	FLTA1R<4:0>				001F	
RPINR13	069A	—	—	—	—	—	—	—	—	—	—	—	FLTA2R<4:0>				001F	
RPINR14	069C	—	—	—	QEB1R<4:0>				—	—	—	QEA1R<4:0>				1F1F		
RPINR15	069E	—	—	—	—	—	—	—	—	—	—	—	INDX1R<4:0>				001F	
RPINR16	06A0	—	—	—	QEB2R<4:0>				—	—	—	QEA2R<4:0>				1F1F		
RPINR17	06A2	—	—	—	—	—	—	—	—	—	—	—	INDX2R<4:0>				001F	
RPINR18	06A4	—	—	—	U1CTSR<4:0>				—	—	—	U1RXR<4:0>				1F1F		
RPINR19	06A6	—	—	—	U2CTSR<4:0>				—	—	—	U2RXR<4:0>				1F1F		
RPINR20	06A8	—	—	—	SCK1R<4:0>				—	—	—	SDI1R<4:0>				1F1F		
RPINR21	06AA	—	—	—	—	—	—	—	—	—	—	—	SS1R<4:0>				001F	
RPINR22	06AC	—	—	—	SCK2R<4:0>				—	—	—	SDI2R<4:0>				1F1F		
RPINR23	06AE	—	—	—	—	—	—	—	—	—	—	—	SS2R<4:0>				001F	
RPINR26 <sup>(1)</sup>	06B4	—	—	—	—	—	—	—	—	—	—	—	C1RXR<4:0>				001F	

**图注:** x = 复位时的未知值, — = 未实现, 读为 0。复位值以 16 进制表示。  
**注 1:** 仅 dsPIC33FJ128MC802/804 和 dsPIC33FJ64MC802/804 器件具有该寄存器。

表 3-25: dsPIC33FJ128MC202/802、dsPIC33FJ64MC202/802 和 dsPIC33FJ32MC302 的外设引脚选择输出寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
RPOR0	06C0	—	—	—	RP1R<4:0>						—	—	—	RP0R<4:0>				0000
RPOR1	06C2	—	—	—	RP3R<4:0>						—	—	—	RP2R<4:0>				0000
RPOR2	06C4	—	—	—	RP5R<4:0>						—	—	—	RP4R<4:0>				0000
RPOR3	06C6	—	—	—	RP7R<4:0>						—	—	—	RP6R<4:0>				0000
RPOR4	06C8	—	—	—	RP9R<4:0>						—	—	—	RP8R<4:0>				0000
RPOR5	06CA	—	—	—	RP11R<4:0>						—	—	—	RP10R<4:0>				0000
RPOR6	06CC	—	—	—	RP13R<4:0>						—	—	—	RP12R<4:0>				0000
RPOR7	06CE	—	—	—	RP15R<4:0>						—	—	—	RP14R<4:0>				0000

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-26: dsPIC33FJ128MC204/804、dsPIC33FJ64MC204/804 和 dsPIC33FJ32MC304 的外设引脚选择输出寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
RPOR0	06C0	—	—	—	RP1R<4:0>						—	—	—	RP0R<4:0>				0000
RPOR1	06C2	—	—	—	RP3R<4:0>						—	—	—	RP2R<4:0>				0000
RPOR2	06C4	—	—	—	RP5R<4:0>						—	—	—	RP4R<4:0>				0000
RPOR3	06C6	—	—	—	RP7R<4:0>						—	—	—	RP6R<4:0>				0000
RPOR4	06C8	—	—	—	RP9R<4:0>						—	—	—	RP8R<4:0>				0000
RPOR5	06CA	—	—	—	RP11R<4:0>						—	—	—	RP10R<4:0>				0000
RPOR6	06CC	—	—	—	RP13R<4:0>						—	—	—	RP12R<4:0>				0000
RPOR7	06CE	—	—	—	RP15R<4:0>						—	—	—	RP14R<4:0>				0000
RPOR8	06D0	—	—	—	RP17R<4:0>						—	—	—	RP16R<4:0>				0000
RPOR9	06D2	—	—	—	RP19R<4:0>						—	—	—	RP18R<4:0>				0000
RPOR10	06D4	—	—	—	RP21R<4:0>						—	—	—	RP20R<4:0>				0000
RPOR11	06D6	—	—	—	RP23R<4:0>						—	—	—	RP22R<4:0>				0000
RPOR12	06D8	—	—	—	RP25R<4:0>						—	—	—	RP24R<4:0>				0000

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-27: dsPIC33FJ128MC202/802、dsPIC33FJ64MC202/802 和 dsPIC33FJ32MC302 的并行主 / 从端口寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
PMCON	0600	PMPEN	—	PSIDL	ADRMUX<1:0>		PTBEEN	PTWREN	PTRDEN	CSF1	CSF0	ALP	—	CS1P	BEP	WRSP	RDSP	0000
PMMODE	0602	BUSY	IRQM<1:0>		INCM<1:0>		MODE16	MODE<1:0>		WAITB<1:0>		WAITM<3:0>			WAITE<1:0>		0000	
PMADDR	0604	ADDR15	CS1	ADDR<13:0>														0000
PMDOUT1		并行端口数据输出寄存器 1 (缓冲区 0 和缓冲区 1)																0000
PMDOUT2	0606	并行端口数据输出寄存器 2 (缓冲区 2 和缓冲区 3)																0000
PMDIN1	0608	并行端口数据输入寄存器 1 (缓冲区 0 和缓冲区 1)																0000
PMPDIN2	060A	并行端口数据输入寄存器 2 (缓冲区 2 和缓冲区 3)																0000
PMAEN	060C	—	PTEN14	—	—	—	—	—	—	—	—	—	—	—	—	—	PTEN<1:0>	0000
PMSTAT	060E	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	0000

图注: — = 未实现, 读为 0。复位值以十六进制表示。

表 3-28: dsPIC33FJ128MC204/804、dsPIC33FJ64MC204/804 和 dsPIC33FJ32MC304 的并行主 / 从端口寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
PMCON	0600	PMPEN	—	PSIDL	ADRMUX<1:0>		PTBEEN	PTWREN	PTRDEN	CSF1	CSF0	ALP	—	CS1P	BEP	WRSP	RDSP	0000
PMMODE	0602	BUSY	IRQM<1:0>		INCM<1:0>		MODE16	MODE<1:0>		WAITB<1:0>		WAITM<3:0>			WAITE<1:0>		0000	
PMADDR	0604	ADDR15	CS1	ADDR<13:0>														0000
PMDOUT1		并行端口数据输出寄存器 1 (缓冲区 0 和缓冲区 1)																0000
PMDOUT2	0606	并行端口数据输出寄存器 2 (缓冲区 2 和缓冲区 3)																0000
PMDIN1	0608	并行端口数据输入寄存器 1 (缓冲区 0 和缓冲区 1)																0000
PMPDIN2	060A	并行端口数据输入寄存器 2 (缓冲区 2 和缓冲区 3)																0000
PMAEN	060C	—	PTEN14	—	—	—	PTEN<10:0>										0000	
PMSTAT	060E	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	0000

图注: — = 未实现, 读为 0。复位值以十六进制表示。



表 3-29: 实时时钟和日历寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
ALRMVAL	0620	基于 APTR<1:0> 的闹钟值寄存器窗口																xxxx
ALCFGRPT	0622	ALRMEN	CHIME	AMASK<3:0>				ALRMPTR<1:0>			ARPT<7:0>						0000	
RTCVAL	0624	基于 RTCPTR<1:0> 的 RTCC 值寄存器窗口																xxxx
RCFGCAL	0626	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR<1:0>				CAL<7:0>						0000

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-30: CRC 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
CRCCON	0640	—	—	CSIDL	VWORD<4:0>				CRCFUL	CRCMPT	—	CRCGO	PLEN<3:0>					0000
CRCXOR	0642	X<15:0>																0000
CRCDAT	0644	CRC 数据输入寄存器																0000
CRCWDAT	0646	CRC 结果寄存器																0000

图注: — = 未实现, 读为 0。复位值以十六进制表示。

表 3-31: 双比较器寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
CMCON	0630	CMIDL	—	C2EVT	C1EVT	C2EN	C1EN	C2OUTEN	C1OUTEN	C2OUT	C1OUT	C2INV	C1INV	C2NEG	C2POS	C1NEG	C1POS	0000
CVRCON	0632	—	—	—	—	—	—	—	—	CVREN	CVROE	CVRR	CVRSS	CVR<3:0>				0000

图注: — = 未实现, 读为 0。复位值以十六进制表示。

表 3-32: dsPIC33FJ128MC202/802、dsPIC33FJ64MC202/802 和 dsPIC33FJ32MC302 的 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	所有复位时的状态
TRISA	02C0	—	—	—	—	—	—	—	—	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	079F
PORTA	02C2	—	—	—	—	—	—	—	—	—	—	—	RA4	RA3	RA2	RA1	RA0	xxxx
LATA	02C4	—	—	—	—	—	—	—	—	—	—	—	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx
ODCA	02C6	—	—	—	—	—	—	—	—	—	—	—	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	xxxx

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-33: dsPIC33FJ128MC204/804、dsPIC33FJ64MC204/804 和 dsPIC33FJ32MC304 的 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	所有复位时的状态
TRISA	02C0	—	—	—	—	—	TRISA10	TRISA9	TRISA8	TRISA7	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	079F
PORTA	02C2	—	—	—	—	—	RA10	RA9	RA8	RA7	—	—	RA4	RA3	RA2	RA1	RA0	xxxx
LATA	02C4	—	—	—	—	—	LATA10	LATA9	LATA8	LATA7	—	—	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx
ODCA	02C6	—	—	—	—	—	ODCA10	ODCA9	ODCA8	ODCA7	—	—	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	xxxx

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-34: PORTB 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
TRISB	02C8	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	FFFF
PORTB	02CA	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx
LATB	02CC	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx
ODCB	02CE	ODCB15	ODCB14	ODCB13	ODCB12	ODCB11	ODCB10	ODCB9	ODCB8	ODCB7	ODCB6	ODCB5	ODCB4	ODCB3	ODCB2	ODCB1	ODCB0	xxxx

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-35: dsPIC33FJ128MC204/804、dsPIC33FJ64MC204/804 和 dsPIC33FJ32MC304 的 PORTC 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
TRISC	02D0	—	—	—	—	—	—	TRISC9	TRISC8	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	03FF
PORTC	02D2	—	—	—	—	—	—	RC9	RC8	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx
LATC	02D4	—	—	—	—	—	—	LATC9	LATC8	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx
ODCC	02D6	—	—	—	—	—	—	ODCC9	ODCC8	ODCC7	ODCC6	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0	xxxx

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-36: 系统控制寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态	
RCON	0740	TRAPR	IOPUWR	—	—	—	—	CM	VREGS	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	xxxx <sup>(1)</sup>	
OSCCON	0742	—	COSC<2:0>				—	NOSC<2:0>			CLKLOCK	IOLOCK	LOCK	—	CF	—	LPOSCEN	OSWEN	0300 <sup>(2)</sup>
CLKDIV	0744	ROI	DOZE<2:0>			DOZEN	FRCDIV<2:0>			PLLPOST<1:0>			—	PLLPRE<4:0>				0040	
PLLFBD	0746	—	—	—	—	—	—	—	PLLDIV<8:0>									0030	
OSCTUN	0748	—	—	—	—	—	—	—	—	—	—	TUN<5:0>						0000	
ACLKCON	074A	—	—	SELACLK	AOSCMD<1:0>			APSTSCLR<2:0>			ASRCSEL	—	—	—	—	—	—	0000	

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

- 注 1: RCON 寄存器的复位值取决于复位类型。  
注 2: OSCCON 寄存器的复位值取决于 FOSC 配置位和复位类型。

表 3-37: 仅限 dsPIC33FJ128MC204/804 和 dsPIC33FJ64MC204/804 的安全性寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
BSRAM	0750	—	—	—	—	—	—	—	—	—	—	—	—	—	IW_BSR	IR_BSR	RL_BSR	0000
SSRAM	0752	—	—	—	—	—	—	—	—	—	—	—	—	—	IW_SSR	IR_SSR	RL_SSR	0000

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-38: NVM 寄存器映射

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

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

表 3-39: PMD 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
PMD1	0770	T5MD	T4MD	T3MD	T2MD	T1MD	QE11MD	PWM1MD	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	C1MD	AD1MD	0000
PMD2	0772	IC8MD	IC7MD	—	—	—	—	IC2MD	IC1MD	—	—	—	—	OC4MD	OC3MD	OC2MD	OC1MD	0000
PMD3	0774	—	—	—	—	—	CMPMD	RTCCMD	PMPMD	CRCMD	DAC1MD	QE12MD	PWM2MD	—	—	—	—	0000

图注: x = 复位时的未知值, — = 未实现, 读为 0。复位值以十六进制表示。

### 3.2.7 软件堆栈

除了用作工作寄存器外， dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件中的 W15 寄存器也可用作软件堆栈指针。堆栈指针总是指向第一个可用的空字，并且从低地址向高地址方向增长。它在弹出堆栈之前递减，而在压入堆栈之后递增，如图 3-6 所示。对于执行任何 CALL 指令期间的 PC 压栈，在压入堆栈之前，PC 的 MSb 要进行零扩展，从而确保 MSb 始终是清零的。

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

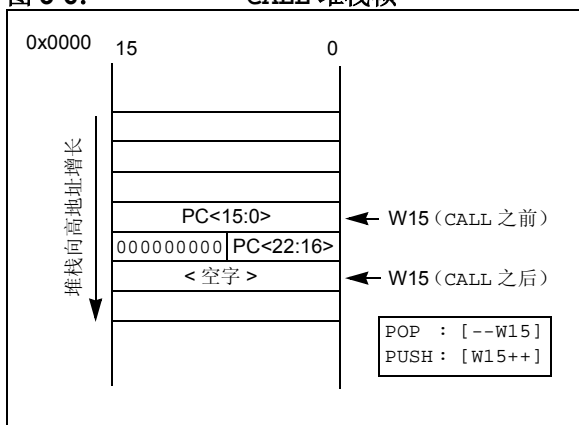
堆栈指针限制寄存器 (SPLIM) 与堆栈指针相关联，它设置堆栈地址上边界的值。复位时 SPLIM 不被初始化。与堆栈指针的情况一样，SPLIM<0> 被强制为 0，因为所有的堆栈操作必须是字对齐的。

每当使用 W15 作为源指针或目标指针生成 EA 时，生成的有效地址会与 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。例如，当堆栈增长超过 RAM 中地址 0x2000 时，如果要想产生堆栈错误陷阱，则需要用值 0x1FFE 来初始化 SPLIM。

类似地，当堆栈指针地址小于 0x0800 时，就会产生堆栈指针下溢 (堆栈错误) 陷阱。这可防止堆栈进入特殊功能寄存器 (SFR) 空间。

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

图 3-6: CALL 堆栈帧



### 3.2.8 数据 RAM 保护功能

dsPIC33F 产品系列支持数据 RAM 保护功能，允许使用引导和安全代码段安全性来保护 RAM 段。BS 的安全 RAM 段 (Secure RAM Segment for BS, BSRAM) 使能时，只能从引导段闪存代码进行访问。RAM 的安全 RAM 段 (Secure RAM Segment for RAM, SSRAM) 使能时，只能从安全段闪存代码进行访问。请参见表 3-37 中对 BSRAM 和 SSRAM SFR 的概括。

## 3.3 指令寻址模式

表 3-40 给出了基本的寻址模式，这些寻址模式经过优化可以支持各指令的具体功能。MAC 类指令中提供的寻址模式与其他指令类型中的寻址模式有所不同。

### 3.3.1 文件寄存器指令

大多数文件寄存器指令使用一个 13 位地址字段 (f) 来直接寻址数据存储器中的前 8192 字节 (Near 数据空间)。大多数文件寄存器指令使用工作寄存器 W0，W0 在这些指令中表示为 WREG。目标寄存器通常是同一个文件寄存器或 WREG (MUL 指令除外，它把结果写入寄存器或寄存器对)。使用 MOV 指令能够获得更大的灵活性，且可以访问整个数据空间。

### 3.3.2 MCU 指令

三操作数 MCU 指令的形式是：

操作数 3 = 操作数 1 < 功能 > 操作数 2

其中，操作数 1 始终是称为 Wb 的工作寄存器 (即，寻址模式只能是寄存器直接寻址)。操作数 2 可以是一个 W 寄存器，取自数据存储器或一个 5 位立即数。结果可以被保存在 W 寄存器或数据存储单元中。MCU 指令支持以下寻址模式：

- 寄存器直接寻址
- 寄存器间接寻址
- 执行后修改的寄存器间接寻址
- 执行前修改的寄存器间接寻址
- 5 位或 10 位立即数寻址

**注：** 并非所有指令都支持上述所有的寻址模式。各条指令可能支持这些寻址模式中的某些模式。

表 3-40: 支持的基本寻址模式

寻址模式	说明
文件寄存器直接寻址	明确指定文件寄存器的地址。
寄存器直接寻址	直接访问寄存器的内容。
寄存器间接寻址	Wn 的内容形成有效地址 (EA)。
执行后修改的寄存器间接寻址	Wn 的内容形成 EA。然后用一个常量值来修改 (递增或递减) Wn。
执行前修改的寄存器间接寻址	先用一个有符号常量值修改 (递增或递减) Wn, 再由此时的 Wn 内容形成 EA。
带寄存器偏移量的寄存器间接寻址 (寄存器变址寻址)	Wn 和 Wb 的和形成 EA。
带立即数偏移量的寄存器间接寻址	Wn 和立即数的和形成 EA。

### 3.3.3 传送指令和累加器指令

与其他指令相比, 传送指令和 DSP 累加器类指令提供了更为灵活的寻址模式。除了大多数 MCU 指令支持的寻址模式以外, 传送和累加器指令还支持带寄存器偏移量的寄存器间接寻址模式, 这也称为寄存器变址寻址模式。

**注:** 对于 MOV 指令, 指令中指定的寻址模式对于源和目标 EA, 可以是不同的。然而, 4 位 Wb (寄存器偏移量) 字段为源和目标 EA 所共用 (但通常只由其中之一使用)。

概括地说, 传送和累加器指令支持以下寻址模式:

- 寄存器直接寻址
- 寄存器间接寻址
- 执行后修改的寄存器间接寻址
- 执行前修改的寄存器间接寻址
- 带寄存器偏移量的寄存器间接寻址 (变址寻址)
- 带立即数偏移量的寄存器间接寻址
- 8 位立即数寻址
- 16 位立即数寻址

**注:** 并非所有指令都支持上述所有的寻址模式。各条指令可能支持这些寻址模式中的某些模式。

### 3.3.4 MAC 指令

双源操作数 DSP 指令 (CLR、ED、EDAC、MAC、MPY、MPY.N、MOVSAC 和 MSC), 也称为 MAC 指令, 它们使用一组简化的寻址模式, 允许用户应用程序通过寄存器间接寻址表高效地对数据指针进行操作。

双源操作数预取寄存器必须是集合 {W8, W9, W10, W11} 的成员。对于数据读取操作, W8 和 W9 始终用于 X RAGU, 而 W10 和 W11 始终用于 Y AGU。从而, 产生的有效地址 (无论是在修改之前还是之后), 对于 W8 和 W9 必须是 X 数据空间中的有效地址, 对于 W10 和 W11 则必须是 Y 数据空间中的有效地址。

**注:** 带寄存器偏移量的寄存器间接寻址模式仅可用于 W9 (在 X 空间中) 和 W11 (在 Y 空间中)。

概括地说, MAC 类指令支持以下寻址模式:

- 寄存器间接寻址
- 执行后修改 (修改量为 2) 的寄存器间接寻址
- 执行后修改 (修改量为 4) 的寄存器间接寻址
- 执行后修改 (修改量为 6) 的寄存器间接寻址
- 带寄存器偏移量的寄存器间接寻址 (变址寻址)

### 3.3.5 其他指令

除上述寻址模式外, 一些指令使用各种大小的立即数常量。例如, BRA (转移) 指令使用 16 位有符号立即数常量来直接指定转移的目标, 而 DISI 指令则使用一个 14 位无符号立即数字段。在一些指令中, 例如 ADD Acc, 操作数的来源和运算结果已经暗含在操作码中。某些操作, 例如 NOP, 没有任何操作数。

### 3.4 模寻址

模寻址模式是一种使用硬件来自动支持循环数据缓冲区的方法。目的是在执行紧密循环代码时（这在许多 DSP 算法中很典型），不需要用软件来执行数据地址边界检查。

可以在数据空间或程序空间中进行模寻址（因为这两种空间的数据指针机制本质上是相同的）。每个 X（也提供指向程序空间的指针）和 Y 数据空间中都可支持一个循环缓冲区。模寻址可以对任何 W 寄存器指针进行操作。然而，最好不要将 W14 或 W15 用于模寻址，因为这两个寄存器分别用作堆栈帧指针和堆栈指针。

总的来说，任何特定的循环缓冲区只能配置为单向工作，因为根据缓冲区的方向，对缓冲区起始地址（对于递增缓冲区）或结束地址（对于递减缓冲区）有某些限制。

使用限制的唯一例外是那些长度为 2 的幂的缓冲区。这些缓冲区满足起始和结束地址判据，它们可以双向工作（即在低地址边界和高地址边界上都将进行地址边界检查）。

#### 3.4.1 起始地址和结束地址

模寻址机制要求指定起始和结束地址，并将它们装入 16 位模缓冲区地址寄存器中：XMODSRT、XMODEND、YMODSRT 和 YMODEND（见表 3-1）。

**注：** Y 空间模寻址的 EA 计算使用字长度的数据（每个 EA 的 LSB 始终清零）。

循环缓冲区的长度没有直接指定。它由相应的起始和结束地址之差决定。循环缓冲区的最大长度为 32K 字（64 KB）。

#### 3.4.2 W 地址寄存器选择

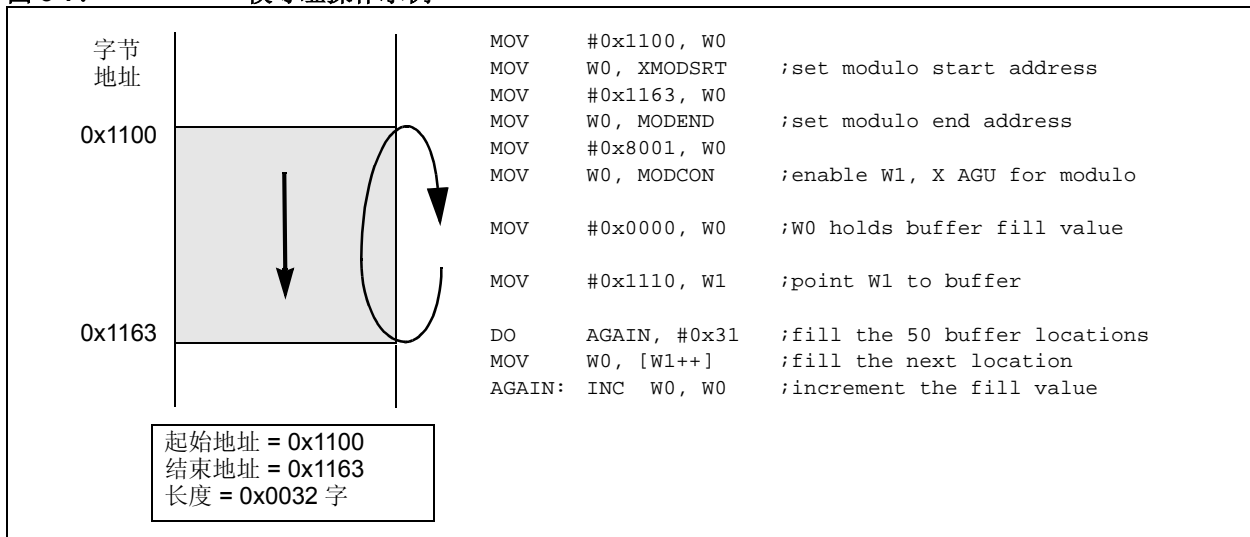
模寻址和位反转寻址控制寄存器 MODCON<15:0> 包含使能标志以及指定 W 地址寄存器的 W 寄存器字段。XWM 和 YWM 字段选择对哪些寄存器进行模寻址：

- 如果 XWM = 15，则禁止 X RAGU 和 X WAGU 模寻址。
- 如果 YWM = 15，则禁止 Y AGU 模寻址。

要进行模寻址的 X 地址空间指针 W 寄存器（XWM）存储在 MODCON<3:0> 中（见表 3-1）。当 XWM 被设置为除 15 之外的任何值且 XMODEN 位（MODCON<15>）置 1 时，X 数据空间的模寻址被使能。

要对其进行模寻址的 Y 地址空间指针 W 寄存器（YWM）存储在 MODCON<7:4> 中。当 YWM 被设置为除 15 之外的任何值且 YMODEN 位（MODCON<14>）置 1 时，Y 数据空间的模寻址被使能。

图 3-7: 模寻址操作示例



### 3.4.3 模寻址的应用

模寻址可以应用于任何与 W 寄存器相关的有效地址 (EA) 计算中。地址边界检查功能等于：

- 递增缓冲器的上边界地址
- 递减缓冲器的下边界地址

重要的是要意识到，地址边界检查功能不仅会检查地址是否正好在地址边界上，而且会检查地址是否小于或大于上限（对于递增缓冲区）、是否低于下限（对于递减缓冲区）。因此，地址变化可能会越过边界，但仍然可以正确调整。

**注：** 只有在使用执行前修改或执行后修改寻址模式来计算有效地址时，模修正有效地址才被写入寄存器。如果使用了地址偏移量（例如，[W7 + W2]），会进行模地址修正，但寄存器的内容保持不变。

### 3.5 位反转寻址

位反转寻址模式用来简化基为 2 的 FFT 算法的数据重新排序。它为 X AGU 所支持，仅限于数据写入。

地址修改量可以是常数或寄存器的内容，可视为将其位顺序反转。源地址和目标地址仍然是正常的顺序。于是，唯一需要反转的操作数就是地址修改量。

#### 3.5.1 位反转寻址的实现

在下列情况下使能位反转寻址模式：

- MODCON 寄存器中 BWM 位（W 寄存器选择）的值是除 15 以外的任何值（不能使用位反转寻址访问堆栈）

- XBREV 寄存器中的 BREN 位置 1
- 使用的寻址模式是预递增或后递增的寄存器间接寻址模式

如果位反转缓冲区的长度为  $M = 2^N$  字节，则数据缓冲区起始地址的最后“N”位必须为零。

**XB<14:0>** 是位反转地址修改量或“中心点”，通常是一个常数。对于 FFT 计算，其值等于 FFT 数据缓冲区长度的一半。

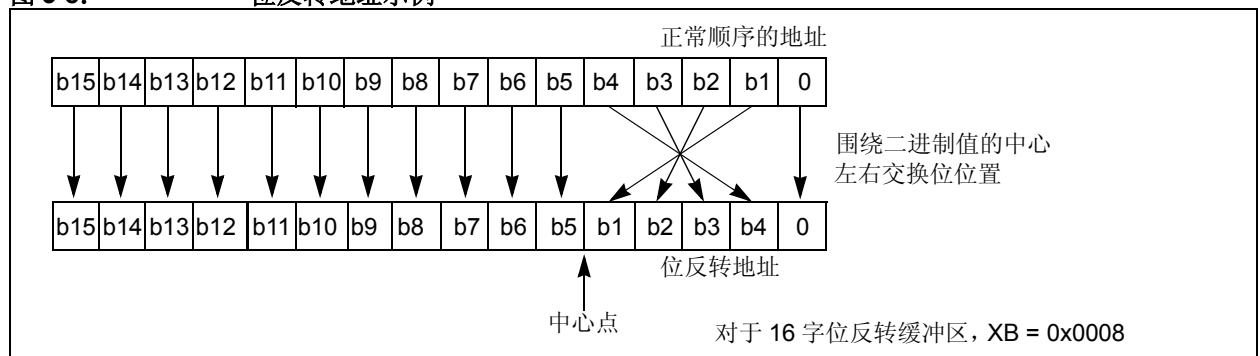
**注：** 所有位反转 EA 的计算都使用字数据（每个 EA 的 LSb 始终清零）。为了产生兼容（字节）地址，要相应地调整 XB 的值。

使能位反转寻址时，仅对预递增或后递增的寄存器间接寻址、且仅对字长度数据写入，才会进行位反转寻址。对于任何其他寻址模式或对于字节长度数据，不会进行位反转寻址，而是生成正常的地址。在进行位反转寻址时，W 地址指针的增量将始终加上地址修改量 (XB)，与寄存器间接寻址模式相关的偏移量将被忽略。此外，由于要求是字数据，EA 的 LSb 被忽略（且始终清零）。

**注：** 不应同时使能模寻址和位反转寻址。如果应用试图这么做的话，对于 X WAGU，位反转寻址将优先，X WAGU 模寻址将被禁止。然而，在 X RAGU 中，模寻址继续起作用。

如果通过将 BREN 位 (XBREV<15>) 置 1 使能了位反转寻址，那么，在写 XBREV 寄存器之后，不应立即进行要使用被指定为位反转指针的 W 寄存器的间接读操作。

图 3-8: 位反转地址示例



**表 3-41: 位反转地址序列 (16 项)**

正常地址					位反转地址				
A3	A2	A1	A0	十进制	A3	A2	A1	A0	十进制
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	8
0	0	1	0	2	0	1	0	0	4
0	0	1	1	3	1	1	0	0	12
0	1	0	0	4	0	0	1	0	2
0	1	0	1	5	1	0	1	0	10
0	1	1	0	6	0	1	1	0	6
0	1	1	1	7	1	1	1	0	14
1	0	0	0	8	0	0	0	1	1
1	0	0	1	9	1	0	0	1	9
1	0	1	0	10	0	1	0	1	5
1	0	1	1	11	1	1	0	1	13
1	1	0	0	12	0	0	1	1	3
1	1	0	1	13	1	0	1	1	11
1	1	1	0	14	0	1	1	1	7
1	1	1	1	15	1	1	1	1	15



### 3.6 程序存储空间与数据存储空间的接口

dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 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。要成功使用这种数据，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，dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 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小块程序存储区。这一功能使得这种方法对于访问需要定期更新的数据表来说非常理想。也可通过表指令访问一个程序字的所有字节。重映射方法允许应用程序访问一大块数据，但只限于读操作，它非常适合于在一个大的静态数据表中进行查找。采用这种方法应用程序只能访问程序字的低位字。

#### 3.6.1 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 位宽，因此需要一个从 16 位数据寄存器创建 23 位或 24 位程序地址的方法。方法取决于所采用的接口方式。

对于表操作，使用 8 位的表页寄存器（TBLPAG）定义程序空间内一个 32K 字的区域。这与 16 位 EA 组合形成一个完整的 24 位程序空间地址。在这种地址形式下，TBLPAG 的最高位用来决定操作是发生在用户存储区中（TBLPAG<7> = 0）还是配置存储区中（TBLPAG<7> = 1）。

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

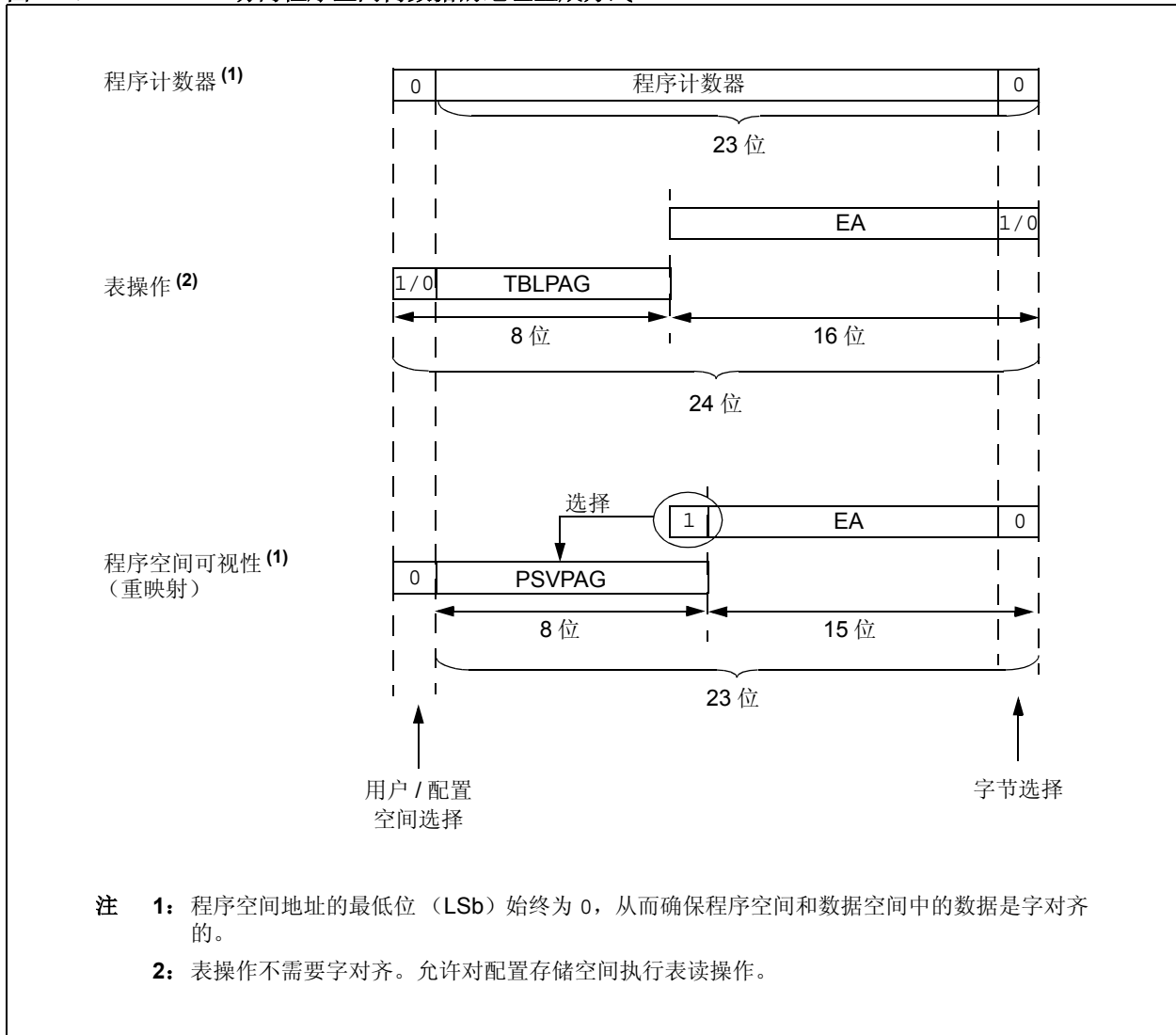
表 3-42 和图 3-9 显示了如何为表操作和重映射访问来从数据 EA 生成程序 EA。本文中，P<23:0> 指程序空间字，而 D<15:0> 指数据空间字。

表 3-42: 程序空间地址构成

访问类型	访问空间	程序空间地址				
		<23>	<22:16>	<15>	<14:1>	<0>
指令访问 (代码执行)	用户	0	PC<22:1>			0
		0xx xxxx xxxx xxxx xxxx 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	PSVPAG<7:0>		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 PSVPAG<0>。

图 3-9: 访问程序空间内数据的地址生成方式



3.6.2 使用表指令访问程序存储器中的数据

TBLRDL 和 TBLWTL 指令提供了直接读或写程序空间内任何地址的低位字的方法，无需通过数据空间。TBLRDH 和 TBLWTH 指令是可以将一个程序空间字的高 8 位作为数据读写的唯一方法。

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

提供了两条表指令来向 / 从程序空间传送字节或字（16 位）大小的数据。两条表指令都可以采用字节或字操作的形式。

- 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。

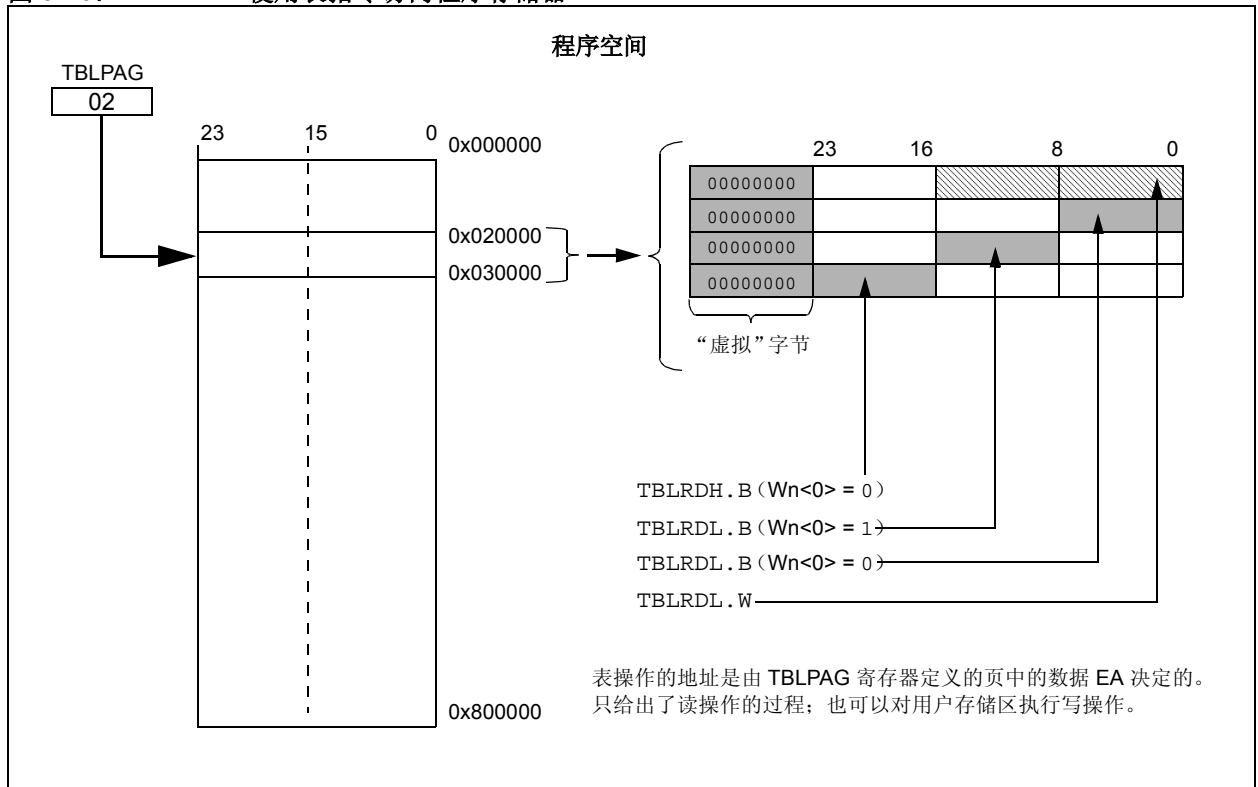
- 在字节模式下，低位程序字的高字节或低字节被映射到数据地址的低字节中。当字节选择位为 1 时映射高字节；当字节选择位为 0 时映射低字节。

- 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。“虚拟”字节（D<15:8>）始终为 0。
  - 在字节模式下，该指令将程序字的高字节或低字节映射到数据地址的 D<7:0> 中，就如同 TBLRDL 指令。当选择高位“虚拟”字节（字节选择位 = 1）时，数据将始终为 0。

两条表指令 TBLWTH 和 TBLWTL 以类似的方式向程序空间地址写入各字节或字。第 4.0 节“闪存程序存储器”对这两条指令的详细操作进行了说明。

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页寄存器（TBLPAG）决定的。TBLPAG 可寻址器件的整个程序存储空间，包括用户应用程序空间和配置空间。当 TBLPAG<7> = 0 时，表页位于用户存储空间中。当 TBLPAG<7> = 1 时，表页位于配置存储空间中。

图 3-10: 使用表指令访问程序存储器



表操作的地址是由 TBLPAG 寄存器定义的页中的数据 EA 决定的。只给出了读操作的过程；也可以对用户存储区执行写操作。

### 3.6.3 使用程序空间可视性读程序存储器中的数据

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

如果数据空间 EA 的最高有效位为 1，并且程序空间可视性使能（通过将内核控制寄存器中的 PSV 位（CORCON<2>）置 1）时，就能通过数据空间访问程序空间。由程序空间可视性页寄存器（PSVPAG）确定要被映射到数据空间中的程序存储空间的位置。这一 8 位的寄存器定义程序空间中 256 个可能的 16K 字页中的任何一个。事实上，PSVPAG 作为程序存储地址的高 8 位，而 EA 的 15 位则作为地址的低位。对于每个程序存储字，PC 都将递增 2，数据空间地址的低 15 位将直接映射到相应程序空间地址的低 15 位。

对该区域进行数据读取的指令，需要一个额外的指令周期，因为这类指令需要对程序存储器进行两次数据取操作。

尽管大于或等于 8000h 的每个数据空间地址直接映射到对应的程序存储器地址（见图 3-11），但只使用 24 位程序字的低 16 位来存放数据。所有用来存放数据的程

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

**注：** 在表读 / 写期间，暂时禁止 PSV 访问。

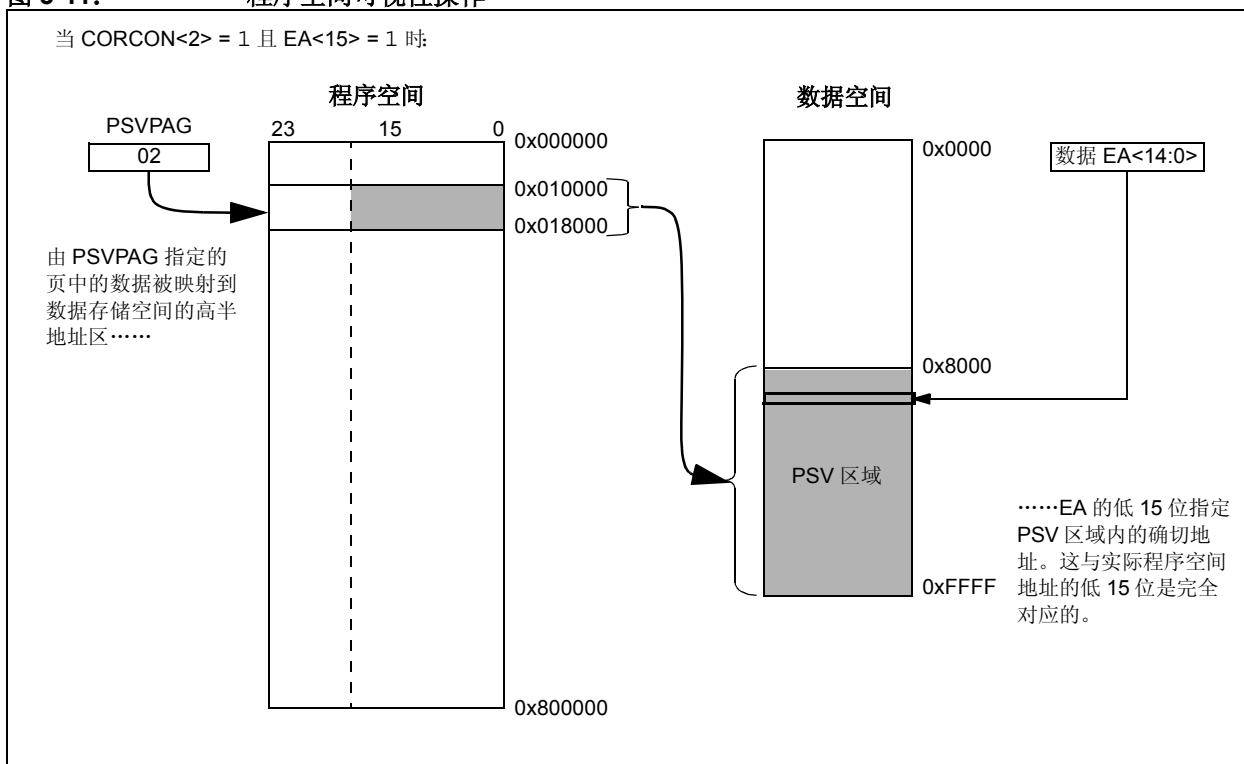
对于使用 PSV 而又在 REPEAT 循环外执行的操作，MOV 和 MOV.D 指令除了规定的执行时间之外，还需要一个额外的指令周期。所有其他指令都需要在规定的执行时间之外额外增加两个指令周期。

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

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

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

图 3-11: 程序空间可视性操作



## 4.0 闪存程序存储器

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 5. Flash Programming”（DS70191），该文档可从 Microchip 网站（www.microchip.com）下载。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件包含用于存储和执行应用代码的内部闪存程序存储器。在整个 V<sub>DD</sub> 范围内，正常操作期间，存储器都是可读写、可擦除的。

可采用两种方式对闪存程序存储器进行编程：

- 在线串行编程（In-Circuit Serial Programming™，ICSP™）功能
- 运行时自编程（Run-Time Self-Programming，RTSP）

ICSP 允许在最终的应用电路中对 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件进行串行编程。编程首先需要两根线作为编程时钟线和编程数据线（以下备用编程引脚对之一：PGC1/PGD1、PGC2/PGD2 或

PGC3/PGD3），还需要三根线作为电源线（V<sub>DD</sub>）、接地线（V<sub>SS</sub>）和主复位线（MCLR）。这允许用户在生产电路板时使用未编程器件，而在产品交付之前才对数字信号控制器进行编程，从而可以使用最新版本的固件或者定制固件进行编程。

使用 TBLRD（表读）和 TBLWT（表写）指令来实现 RTSP。使用 RTSP，用户应用程序可以一次将 64 条指令（192 字节）的块（或“行”）或单个程序存储字写入程序存储器，也可以一次擦除 512 条指令（1536 字节）的块（或“页”）。

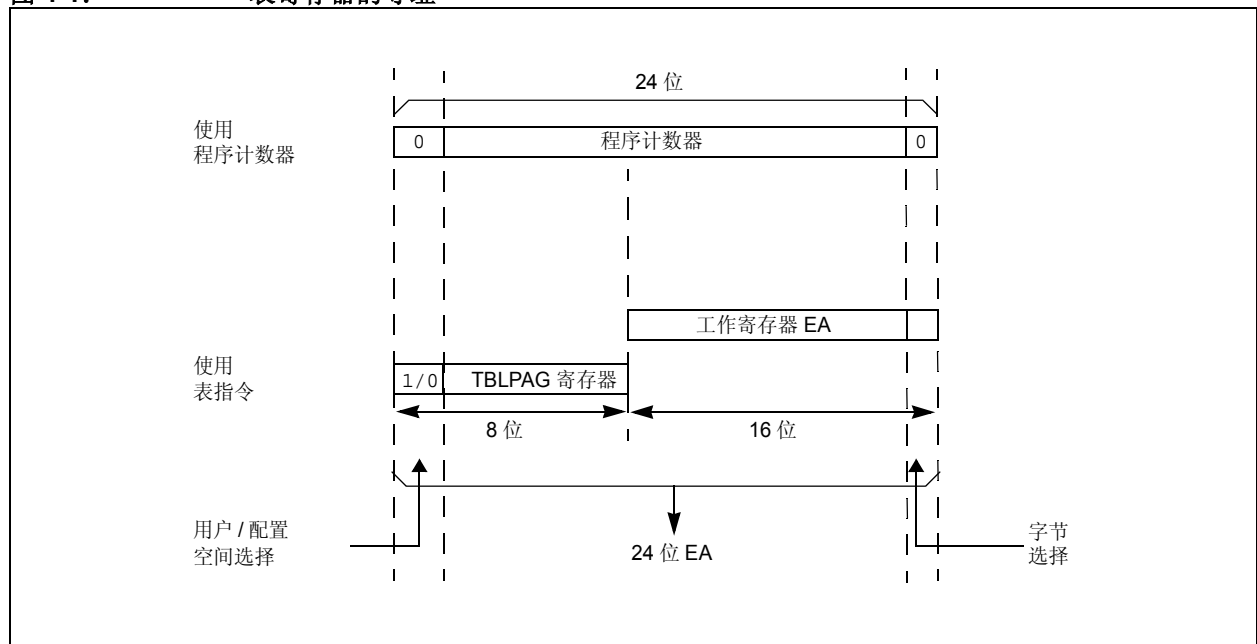
### 4.1 表指令和闪存编程

闪存的编程都是用表读和表写指令实现的，与使用的编程方法无关。这些指令允许器件在正常工作模式下从数据存储器直接读写程序存储空间。程序存储器中 24 位目标地址由 TBLPAG 寄存器的 bit<7:0> 和表指令中指定 W 寄存器中的有效地址（EA）组成，如图 4-1 所示。

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

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

图 4-1: 表寄存器的寻址



## 4.2 RTSP 工作原理

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的闪存程序存储器阵列是由 64 条指令或 192 字节的行组成的。RTSP 允许用户应用程序一次擦除由 8 行（512 条指令）组成的程序存储器页，一次编程一行或一个字。表 30-12 给出了典型的擦除和编程次数。8 行擦除页和单行写入行都是边界对齐的，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。

程序存储器实现了保持缓冲区，它能缓冲 64 条指令的编程数据。在实际编程操作前，必须将待写数据顺序装入缓冲区。要装入的指令字必须总是来自一组 64 个指令字的边界。

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 指令来装载缓冲区。通过设置 NVMCON 寄存器中的控制位来执行编程。装载总共需要 64 条 TBLWTL 和 TBLWTH 指令。

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

## 4.3 控制寄存器

有两个 SFR 用于读写闪存程序存储器：NVMCON 和 NVMKEY。

NVMCON 寄存器（寄存器 4-1）控制要擦除哪些块、要编程的存储器类型以及编程周期的启动。

NVMKEY 是一个只写寄存器，用于写保护。要启动编程或擦除序列，用户应用程序必须按顺序将 0x55 和 0xAA 写入 NVMKEY 寄存器。更多详细信息，请参见第 4.4 节“编程操作”。

## 4.4 编程操作

在 RTSP 模式下，编程或擦除内部闪存需要执行完整的编程序列。编程操作的持续时间通常为 4 ms，在此期间处理器暂停等待操作完成。将 WR 位（NVMCON<15>）置 1 启动编程或擦除操作，当操作完成时 WR 位被自动清零。

寄存器 4-1: **NVMCON: 闪存控制寄存器**

R/SO-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <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	—	—	NVMOP<3:0> <sup>(2)</sup>			
bit 7							bit 0

<b>图注:</b>	SO = 只可置 1 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 **WR:** 写控制位  
1 = 启动闪存编程或擦除操作。该操作是自定时的，一旦操作完成该位即由硬件清零。  
0 = 编程或擦除操作完成，处于停止状态
- bit 14 **WREN:** 写使能位  
1 = 使能闪存编程 / 擦除操作  
0 = 禁止闪存编程 / 擦除操作
- bit 13 **WRERR:** 写序列错误标志位  
1 = 试图执行不合法的编程或擦除序列，或者发生终止（将 WR 位置 1 时自动置 1 该位）  
0 = 编程或擦除操作正常完成
- bit 12-7 **未实现:** 读为 0
- bit 6 **ERASE:** 擦除 / 编程使能位  
1 = 在下一条 WR 命令时执行 NVMOP<3:0> 指定的擦除操作  
0 = 在下一条 WR 命令时执行 NVMOP<3:0> 指定的编程操作
- bit 5-4 **未实现:** 读为 0
- bit 3-0 **NVMOP<3:0>:** NVM 操作选择位 <sup>(2)</sup>  
如果 ERASE = 1:  
1111 = 存储器批量擦除操作  
1110 = 保留  
1101 = 擦除通用段  
1100 = 擦除安全段  
1011 = 保留  
0011 = 无操作  
0010 = 存储器页擦除操作  
0001 = 无操作  
0000 = 擦除单个配置寄存器字节  
  
如果 ERASE = 0:  
1111 = 无操作  
1110 = 保留  
1101 = 无操作  
1100 = 无操作  
1011 = 保留  
0011 = 存储器字编程操作  
0010 = 无操作  
0001 = 存储器行编程操作  
0000 = 编程单个配置寄存器字节

**注 1:** 这些位只能在 POR 时被复位。  
**2:** NVMOP<3:0> 的所有其他组合均未实现。

**寄存器 4-2: NVMKEY: 非易失性存储器密钥寄存器**

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

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
NVMKEY<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      **NVMKEY<7:0>:** 密钥寄存器 (只写) 位



#### 4.4.1 闪存程序存储器的编程算法

程序员一次可以编程闪存程序存储器的一行。要实现该操作，需要擦除包含该行在内的一个 8 行大小的页。一般过程如下：

1. 读取程序存储器的 8 行内容（512 条指令），并存储在数据 RAM 中。
2. 用所需的新数据更新 RAM 中的程序数据。
3. 擦除程序存储器块（见例 4-1）：
  - a) 将 NVMOP 位（NVMCON<3:0>）设置为“0010”，配置为块擦除操作。将 ERASE（NVMCON<6>）和 WREN（NVMCON<14>）位置 1。
  - b) 将要被擦除的页的起始地址写入 TBLPAG 和 W 寄存器。
  - c) 将 0x55 写入 NVMKEY。
  - d) 将 0xAA 写入 NVMKEY。
  - e) 将 WR 位（NVMCON<15>）置 1。擦除周期开始，在擦除周期中 CPU 会暂停。当擦除完成时，WR 位被自动清零。

4. 将数据 RAM 中的前 64 条指令写入程序存储器缓冲区（见例 4-2）。
5. 将程序块写入闪存程序存储器：
  - a) 将 NVMOP 位设置为“0001”，配置为行编程操作。将 ERASE 位清零，将 WREN 位置 1。
  - b) 将 0x55 写入 NVMKEY。
  - c) 将 0xAA 写入 NVMKEY。
  - d) 将 WR 位置 1。编程周期开始，在写周期中 CPU 会暂停。当对闪存程序存储器的写操作完成时，WR 位被自动清零。
6. 通过将 TBLPAG 中的值递增 1，使用数据 RAM 块中下一组 64 条指令重复步骤 4 和 5，直到所有 512 条指令被写回到闪存程序存储器。

为防止意外操作，必须向 NVMKEY 写入写启动序列，用于允许执行擦除或编程操作。在执行编程命令后，用户应用程序必须等待一段编程时间，直至编程完成。紧跟编程启动序列后面的两条指令应为 NOP，如例 4-3 所示。

**例 4-1: 擦除程序存储器页**

```

; Set up NVMCON for block erase operation
MOV    #0x4042, W0          ;
MOV    W0, NVMCON          ; Initialize NVMCON
; Init pointer to row to be ERASED
MOV    #tblpage(PROG_ADDR), W0      ;
MOV    W0, TBLPAG          ; Initialize PM Page Boundary SFR
MOV    #tbloffset(PROG_ADDR), W0    ; Initialize in-page EA[15:0] pointer
TBLWTL W0, [W0]             ; Set base address of erase block
DISI   #5                   ; Block all interrupts with priority <7
                                ; for next 5 instructions

MOV    #0x55, W0
MOV    W0, NVMKEY          ; Write the 55 key
MOV    #0xAA, W1
MOV    W1, NVMKEY          ; Write the AA key
BSET   NVMCON, #WR         ; Start the erase sequence
NOP                                ; Insert two NOPs after the erase
NOP                                ; command is asserted
    
```

**例 4-2: 装载写缓冲区**

```

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

```

**例 4-3: 启动编程序列**

```

DISI    #5                    ; Block all interrupts with priority <7
                                        ; for next 5 instructions
MOV     #0x55, W0              ;
MOV     W0, NVMKEY             ; Write the 55 key
MOV     #0xAA, W1              ;
MOV     W1, NVMKEY             ; Write the AA key
BSET    NVMCON, #WR           ; Start the erase sequence
NOP                                           ; Insert two NOPs after the
NOP                                           ; erase command is asserted

```

## 5.0 复位

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 8. Reset” (DS70192)，该文档可从 Microchip 网站 (www.microchip.com) 下载。

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

- POR: 上电复位
- BOR: 欠压复位
- MCLR: 主复位引脚复位
- SWR: RESET 指令
- WDTO: 看门狗定时器复位
- CM: 配置不匹配复位
- TRAPR: 陷阱冲突复位
- IOPUWR: 非法条件器件复位
  - 非法操作码复位
  - 未初始化的 W 寄存器复位
  - 安全性复位

复位模块的简化框图如图 5-1 所示。

任何有效的复位源都将使  $\overline{\text{SYSRST}}$  信号有效。在系统复位时，某些与 CPU 和外设相关的寄存器被强制为已知的复位状态，而有一些寄存器不受影响。

**注：** 如需了解寄存器复位状态的信息，请参见本手册中特定的外设章节或第 2.0 节“CPU”。

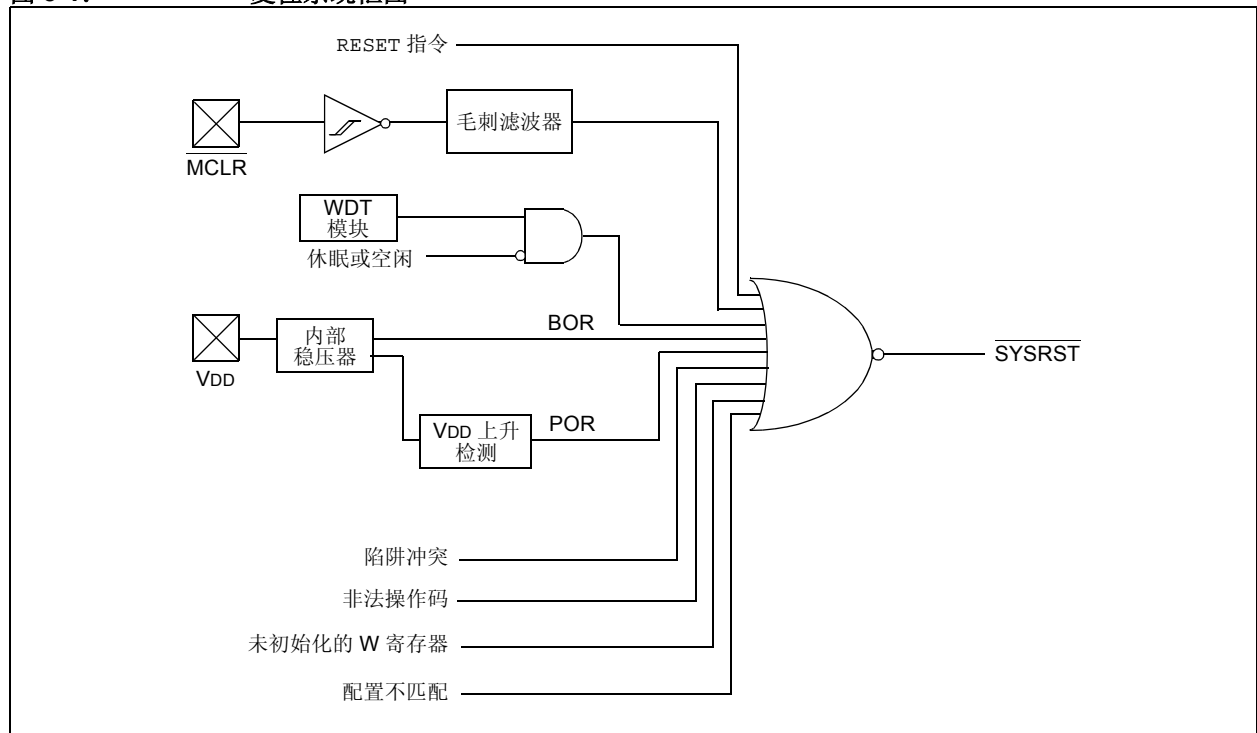
所有类型的器件复位都会将 RCON 寄存器中相应的状态位置 1，以表明复位类型（见寄存器 5-1）。

POR 将清零 RCON 寄存器中除 POR 位 (RCON<0>) 之外的所有位，POR 位在 POR 时置 1。用户应用程序可在代码执行过程中的任何时间置 1 或清零任意位。RCON 寄存器中的位仅用作状态位。用软件将某个复位状态位置 1 不会导致器件发生复位。

RCON 寄存器还包含与看门狗定时器和器件节能状态相关的其他位。本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，这样在器件复位后的下一个 RCON 寄存器值才有意义。

图 5-1: 复位系统框图



寄存器 5-1: RCON: 复位控制寄存器 (1)

R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
TRAPR	IOPUWR	—	—	—	—	CM	VREGS
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	SWR	SWDTEN(2)	WDTO	SLEEP	IDLE	BOR	POR
bit 7						bit 0	

图注:

R = 可读位                      W = 可写位                      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  
 1 = 发生了陷阱冲突复位  
 0 = 未发生陷阱冲突复位
- bit 14        **IOPUWR:** 非法操作码或未初始化的 W 寄存器访问复位标志位  
 1 = 检测到非法操作码、非法地址模式或将未初始化的 W 寄存器用作地址指针从而导致复位  
 0 = 未发生非法操作码或未初始化的 W 寄存器复位
- bit 13-10    **未实现:** 读为 0
- bit 9         **CM:** 配置不匹配标志位  
 1 = 发生了配置不匹配复位  
 0 = 未发生配置不匹配复位
- bit 8         **VREGS:** 休眠模式下稳压器待机位  
 1 = 在休眠模式下稳压器继续工作  
 0 = 在休眠模式下稳压器进入待机模式
- bit 7         **EXTR:** 外部复位 (MCLR) 引脚位  
 1 = 发生主复位 (引脚) 复位  
 0 = 未发生主复位 (引脚) 复位
- bit 6         **SWR:** 软件复位 (指令) 标志位  
 1 = 执行了 RESET 指令  
 0 = 未执行 RESET 指令
- bit 5         **SWDTEN:** 软件使能 / 禁止 WDT 位 (2)  
 1 = 使能 WDT  
 0 = 禁止 WDT
- bit 4         **WDTO:** 看门狗定时器超时标志位  
 1 = 发生了 WDT 超时  
 0 = 未发生 WDT 超时
- bit 3         **SLEEP:** 从休眠模式唤醒标志位  
 1 = 器件处于休眠模式  
 0 = 器件未处于休眠模式
- bit 2         **IDLE:** 从空闲模式唤醒标志位  
 1 = 器件处于空闲模式  
 0 = 器件未处于空闲模式

注 1: 所有复位状态位都可以用软件置 1 或清零。用软件将这些位中的某一位置 1 不会导致器件复位。  
 2: 如果 FWDTEN 配置位为 1 (未编程), 则 WDT 始终使能, 而与 SWDTEN 位的设置无关。

**寄存器 5-1: RCON: 复位控制寄存器<sup>(1)</sup> (续)**

bit 1      **BOR:** 欠压复位标志位  
          1 = 发生了欠压复位  
          0 = 未发生欠压复位

bit 0      **POR:** 上电复位标志位  
          1 = 发生了上电复位  
          0 = 未发生上电复位

- 注 1:** 所有复位状态位都可以用软件置 1 或清零。用软件将这些位中的某一位置 1 不会导致器件复位。
- 2:** 如果 FWDTEN 配置位为 1 (未编程), 则 WDT 始终使能, 而与 SWDTEN 位的设置无关。

## 5.1 系统复位

dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件有两种复位类型：

- 冷复位
- 热复位

上电复位（POR）或欠压复位（BOR）的结果是引起冷复位。在冷复位时，FOSC 器件配置寄存器的 FNOSC 配置位选择器件时钟源。

所有其他复位源（包括 RESET 指令）都引发热复位。在热复位时，器件继续依靠当前时钟源下运行，这个时钟源由振荡器控制寄存器的当前振荡器选择 COSC<2:0>（OSCCON<14:12>）位来指示。

器件将保持在复位状态，直到系统电源稳定在适当的电平并且振荡器时钟准备就绪。其时序在下文详述，如图 5-2 所示。

1. **POR 复位：**当电源开启时，POR 电路将器件保持在复位状态。POR 电路保持有效直到 VDD 超过 VPOR 门限并且 TPOR 延时已经完成。

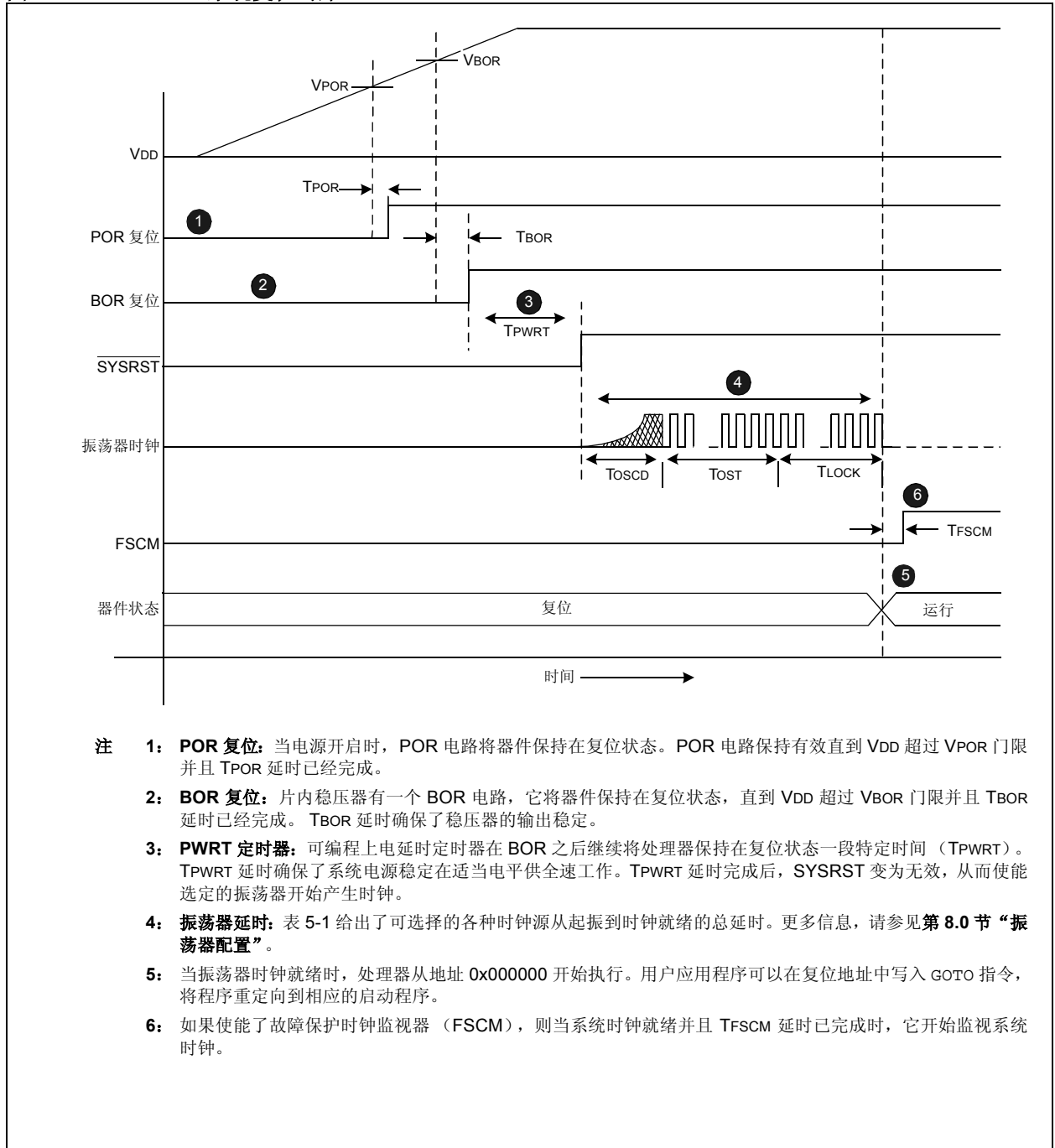
2. **BOR 复位：**片内稳压器有一个 BOR 电路，它将器件保持在复位状态，直到 VDD 超过 VBOR 门限并且 TBOR 延时已经完成。TBOR 延时确保了稳压器的输出稳定。
3. **PWRT 定时器：**可编程上电延时定时器在 BOR 之后继续将处理器保持在复位状态一段特定时间（TPWRT）。TPWRT 延时确保了系统电源稳定在适当电平供全速工作。TPWRT 延时完成后，SYSRST 变为无效，从而使能选定的振荡器开始产生时钟。
4. **振荡器延时：**表 5-1 给出了各种可选择时钟源从起振到时钟就绪之前的总延时。更多信息，请参见第 8.0 节“振荡器配置”。
5. 当振荡器时钟就绪时，处理器从地址 0x000000 开始执行。用户应用程序可以在复位地址中写入 goto 指令，将程序重定向到相应的启动程序。
6. 如果使能了故障保护时钟监视器（Fail-Safe Clock Monitor, FSCM），则当系统时钟就绪并且 TFSCM 延时已完成时，它开始监视系统时钟。

表 5-1: 振荡器延时

振荡器模式	振荡器起振延时	振荡器起振定时器	PLL 锁定时间	总延时
FRC、FRCDIV16 和 FRCDIVN	TOSCD	—	—	TOSCD
FRCPLL	TOSCD	—	TLOCK	TOSCD + TLOCK
XT	TOSCD	TOST	—	TOSCD + TOST
HS	TOSCD	TOST	—	TOSCD + TOST
EC	—	—	—	—
XTPLL	TOSCD	TOST	TLOCK	TOSCD + TOST + TLOCK
HSPLL	TOSCD	TOST	TLOCK	TOSCD + TOST + TLOCK
ECPLL	—	—	TLOCK	TLOCK
SOSC	TOSCD	TOST	—	TOSCD + TOST
LPRC	TOSCD	—	—	TOSCD

- 注 1: TOSCD = 振荡器起振延时（对于 FRC 最大值为 1.1 μs，对于 LPRC 最大值为 70 μs）。晶振起振时间随晶体特性和负载电容等变化。
- 2: TOST = 振荡器起振定时器延时（1024 个振荡器时钟周期）。例如，10 MHz 晶振的 TOST = 102.4 μs，32 kHz 晶振的 TOST = 32 ms。
- 3: 如果 PLL 使能，TLOCK = PLL 锁定时间（标称值为 1.5 ms）。

图 5-2: 系统复位时序



- 注
- 1: POR 复位:** 当电源开启时，POR 电路将器件保持在复位状态。POR 电路保持有效直到 VDD 超过  $V_{POR}$  门限并且  $T_{POR}$  延时已经完成。
  - 2: BOR 复位:** 片内稳压器有一个 BOR 电路，它将器件保持在复位状态，直到 VDD 超过  $V_{BOR}$  门限并且  $T_{BOR}$  延时已经完成。 $T_{BOR}$  延时确保了稳压器的输出稳定。
  - 3: PWRT 定时器:** 可编程上电延时定时器在 BOR 之后继续将处理器保持在复位状态一段特定时间 ( $T_{PWRT}$ )。 $T_{PWRT}$  延时确保了系统电源稳定在适当电平供全速工作。 $T_{PWRT}$  延时完成后，SYSRST 变为无效，从而使能选定的振荡器开始产生时钟。
  - 4: 振荡器延时:** 表 5-1 给出了可选择的各种时钟源从起振到时钟就绪的总延时。更多信息，请参见第 8.0 节“振荡器配置”。
  - 5:** 当振荡器时钟就绪时，处理器从地址 0x000000 开始执行。用户应用程序可以在复位地址中写入 GOTO 指令，将程序重定向到相应的启动程序。
  - 6:** 如果使能了故障保护时钟监视器 (FSCM)，则当系统时钟就绪并且  $T_{FSCM}$  延时已完成时，它开始监视系统时钟。

表 5-2: 振荡器延时

符号	参数	值
VPOR	POR 门限	1.8V 标称值
TPOR	POR 延时时间	30 $\mu$ s 最大值
VBOR	BOR 门限	2.5V 标称值
TBOR	BOR 延时时间	100 $\mu$ s 最大值
TPWRT	可编程上电延时	0-128 ms 标称值
TFSCM	故障保护时钟监视器延时	900 $\mu$ s 最大值

**注:** 当器件退出复位状态（开始正常操作）时，器件工作参数（电压、频率和温度等）必须它们的工作范围内，否则器件可能无法正常工作。用户应用程序必须确保从最开始上电到SYSRST变成无效之间的延时足够长，以使所有工作参数都符合规范。

## 5.2 上电复位（POR）

上电复位（POR）电路确保器件在上电时复位。POR 电路保持有效直到 VDD 超过 VPOR 门限并且 TPOR 延时已经完成。TPOR 延时确保了器件内部偏置电路的稳定。

器件供电电压的特性必须符合规定的起始电压和上升率要求以产生 POR。更多详细信息，请参见第 30.0 节“电气特性”。

复位控制寄存器中的 POR 状态位 POR（RCON<0>）置 1，表示发生上电复位。

### 5.2.1 欠压复位（BOR）和上电延时定时器（PWRT）

片上稳压器有一个欠压复位（BOR）电路，当 VDD 太低（ $VDD < VBOR$ ）器件无法正常工作时将复位器件。BOR 电路将器件保持在复位状态，直到 VDD 超过 VBOR 门限并且 TBOR 延时已经完成。TBOR 延时确保了稳压器的输出稳定。

复位控制寄存器中的 BOR 状态位 BOR（RCON<1>）置 1，表示发生欠压复位。

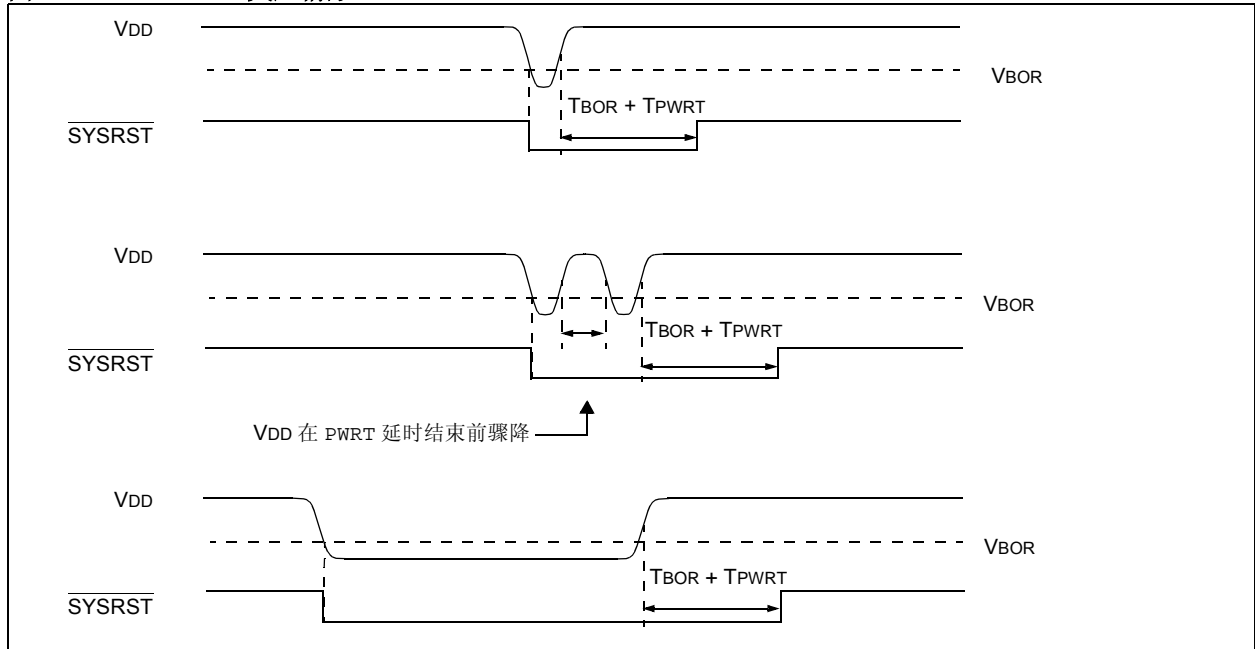
BOR 之后器件不会全速工作，因为全速工作需要 VDD 升至可接受的电平。PWRT 提供上电延时（TPWRT），确保释放SYSRST之前系统电源已稳定在供全速工作的适当电路上。

上电延时定时器的延时（TPWRT）由 POR 配置寄存器中的上电复位定时器值选择位 FPWRT<2:0>（FPOR<2:0>）来编程设定，它提供了 8 种设定（从 0 ms 到 128 ms）。更多详细信息，请参见第 27.0 节“特殊功能”。

图 5-3 给出了典型的欠压情形。复位延时（TBOR + TPWRT）在每次 VDD 上升到超过 VBOR 跳变点时开始。



图 5-3: 欠压情形



### 5.3 外部复位 (EXTR)

通过将 MCLR 引脚驱动为低电平产生外部复位。MCLR 引脚是附带毛刺滤波器的施密特触发器输入。宽度大于最小脉冲宽度的复位脉冲会产生复位。最小脉冲宽度规范请参见第 30.0 节“电气特性”。复位控制 (RCON) 寄存器中的外部复位 (MCLR) 引脚 (EXTR) 位置 1, 表示发生 MCLR 复位。

#### 5.3.1 外部监控电路

许多系统具有外部监控电路, 这种监控电路可产生复位信号将系统中的多个器件复位。可将外部复位信号直接连接到 MCLR 引脚, 使系统的其他部分复位时复位器件。

#### 5.3.2 内部监控电路

当使用内部电源监控电路复位器件时, 外部复位引脚 (MCLR) 应直连或通过电阻连接到 VDD。在这种情况下, MCLR 引脚不用于产生复位。外部复位引脚 (MCLR) 没有内部上拉, 所以不能处于未连接状态。

### 5.4 软件 RESET 指令 (SWR)

每当执行 RESET 指令时, 器件都将产生 SYSRST, 将器件置于特殊的复位状态。此种复位状态不会重新初始化时钟。执行 RESET 指令前使用的时钟源仍然继续使用。SYSRST 在下一个指令周期释放, 并且开始取复位向量。

复位控制寄存器中的软件复位 (指令) 标志位 SWR (RCON<6>) 置 1, 表示发生软件复位。

### 5.5 看门狗超时复位 (WDTR)

只要发生看门狗超时, 器件都将异步产生 SYSRST 信号。时钟源保持不变。在休眠或空闲模式下, WDT 超时将唤醒处理器, 但不会复位处理器。

复位控制寄存器中的看门狗定时器超时标志 WDTO (RCON<4>) 位置 1, 表示发生看门狗复位。更多关于看门狗复位的信息, 请参见第 27.4 节“看门狗定时器 (WDT)”。

### 5.6 陷阱冲突复位

如果在某个高优先级的陷阱正在进行处理时, 发生了低优先级的硬陷阱, 则将发生硬陷阱冲突复位。硬陷阱包括优先级为 13 到 15 的异常。地址错误 (优先级 13) 和振荡器错误 (优先级 14) 陷阱都属于此类。

复位控制寄存器中的陷阱复位标志 TRAPR (RCON<15>) 位置 1, 表示发生陷阱冲突复位。更多关于陷阱冲突复位的信息, 请参见第 6.0 节“中断控制器”。

## 5.7 配置不匹配复位

为保持外设引脚选择控制寄存器的完整性，使用硬件影子寄存器不断监视它们。如果任何寄存器发生了异常改变（例如 ESD 或其他外部事件引起的干扰），将会发生配置不匹配复位。

复位控制寄存器中的配置不匹配标志位 CM (RCON<9>) 置 1，表示发生配置不匹配复位。更多关于配置不匹配复位的信息，请参见第 10.0 节“I/O 端口”。

**注：** 配置不匹配特性和相关的复位标志并非在所有的器件上都可用。

## 5.8 非法条件器件复位

如下原因会发生非法条件器件复位：

- 非法操作码复位
- 未初始化的 W 寄存器复位
- 安全性复位

复位控制寄存器中的非法操作码或访问未初始化的 W 寄存器复位标志 IOPUWR (RCON<14>) 位置 1，表示发生非法条件器件复位。

### 5.8.1 非法操作码复位

如果器件试图执行从程序存储器中取出的非法操作码值，将会产生器件复位。

非法操作码复位功能可以阻止器件执行用于存储常量数据的程序存储器段。要利用非法操作码复位，只能使用每个程序存储器段的低 16 位存储数据值。高 8 位应该被编程为非法操作码值 3Fh。

### 5.8.2 未初始化的 W 寄存器复位

任何试图使用未初始化的 W 寄存器作为地址指针的操作，都将复位器件。所有复位过程中都将清零 W 寄存器阵列 (W15 除外)，并在写入前将 W 寄存器阵列视为未初始化状态。

### 5.8.3 安全性复位

如果程序流变化 (Program Flow Change, PFC) 或向量流变化 (Vector Flow Change, VFC) 的目标是受保护段 (引导和安全段) 中的受限地址，该操作将导致安全性复位。

当调用、跳转、计算跳转、返回、从子程序返回或者其他形式的转移指令导致程序计数器被重载时，将发生 PFC。

当中断或者陷阱向量导致程序计数器被重载时发生 VFC。

更多关于安全性复位的信息，请参见第 27.8 节“代码保护和 CodeGuard 安全性”。

## 5.9 使用 RCON 状态位

任何器件复位后，用户应用程序都可以读取复位控制 (RCON) 寄存器以判断复位原因。

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

表 5-3 提供了复位标志位操作的总结。

**表 5-3: 复位标志位操作**

标志位	置 1 原因	清零原因
TRAPR (RCON<15>)	陷阱冲突事件	POR 和 BOR
IOPWR (RCON<14>)	非法操作码、访问了未初始化的 W 寄存器或安全性复位	POR 和 BOR
CM (RCON<9>)	配置不匹配	POR 和 BOR
EXTR (RCON<7>)	MCLR 复位	POR
SWR (RCON<6>)	RESET 指令	POR 和 BOR
WDTO (RCON<4>)	WDT 超时	PWRSV 指令、CLRWDT 指令、POR 和 BOR
SLEEP (RCON<3>)	PWRSV #SLEEP 指令	POR 和 BOR
IDLE (RCON<2>)	PWRSV #IDLE 指令	POR 和 BOR
BOR (RCON<1>)	POR 和 BOR	
POR (RCON<0>)	POR	

**注：** 所有复位标志位均可由用户软件置 1 或清零。

## 6.0 中断控制器

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 6. Reset” (DS70184)，该文档可从 Microchip 网站 (www.microchip.com) 下载。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的中断控制器将诸多外设中断请求信号缩减为一个送往 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 CPU 的中断请求。

中断控制器具有以下特性：

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

### 6.1 中断向量表

IVT (如图 6-1 所示) 位于程序存储器中，起始存储单元地址是 000004h。IVT 包含 126 个向量，由 8 个不可屏蔽陷阱向量和多达 118 个中断源组成。一般来说，每个中断源都有自己的中断向量。每个中断向量都包含一个 24 位宽的地址。每个中断向量存储单元中设置的值是其相关的中断服务程序 (ISR) 的起始地址。

中断向量根据它们的自然优先级区分优先次序。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。一般而言，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。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件实现了多达 53 个唯一中断和 5 个不可屏蔽陷阱中断。表 6-1 对此作了总结。

#### 6.1.1 备用中断向量表

备用中断向量表 (AIVT) 位于 IVT 之后，如图 6-1 所示。由 ALTIVT 控制位 (INTCON2<15>) 控制对 AIVT 的访问。如果 ALTIVT 位置 1，则所有的中断和异常处理都将使用备用向量，而非默认向量。备用向量与默认向量的组织方式相同。

AIVT 通过提供一种不需要重新编程中断向量就可以在应用程序和支持环境之间切换的方法，来支持调试功能。此特性也支持运行时在应用程序之间切换以便评估各种不同的软件算法。如果不需要 AIVT，则应该用 IVT 中同样的地址编程 AIVT。

## 6.2 复位过程

器件复位不是真正的异常，因为复位过程中并不涉及到中断控制器。作为对复位的响应，dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件清零其寄存器，同时强制 PC 为零。然后数字信号控制器从地址 0x000000 处开始执行程序。在复位地址中写入一条 GOTO 指令，可以将程序执行重定向到相应的启动程序。

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

图 6-1: dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 中断向量表

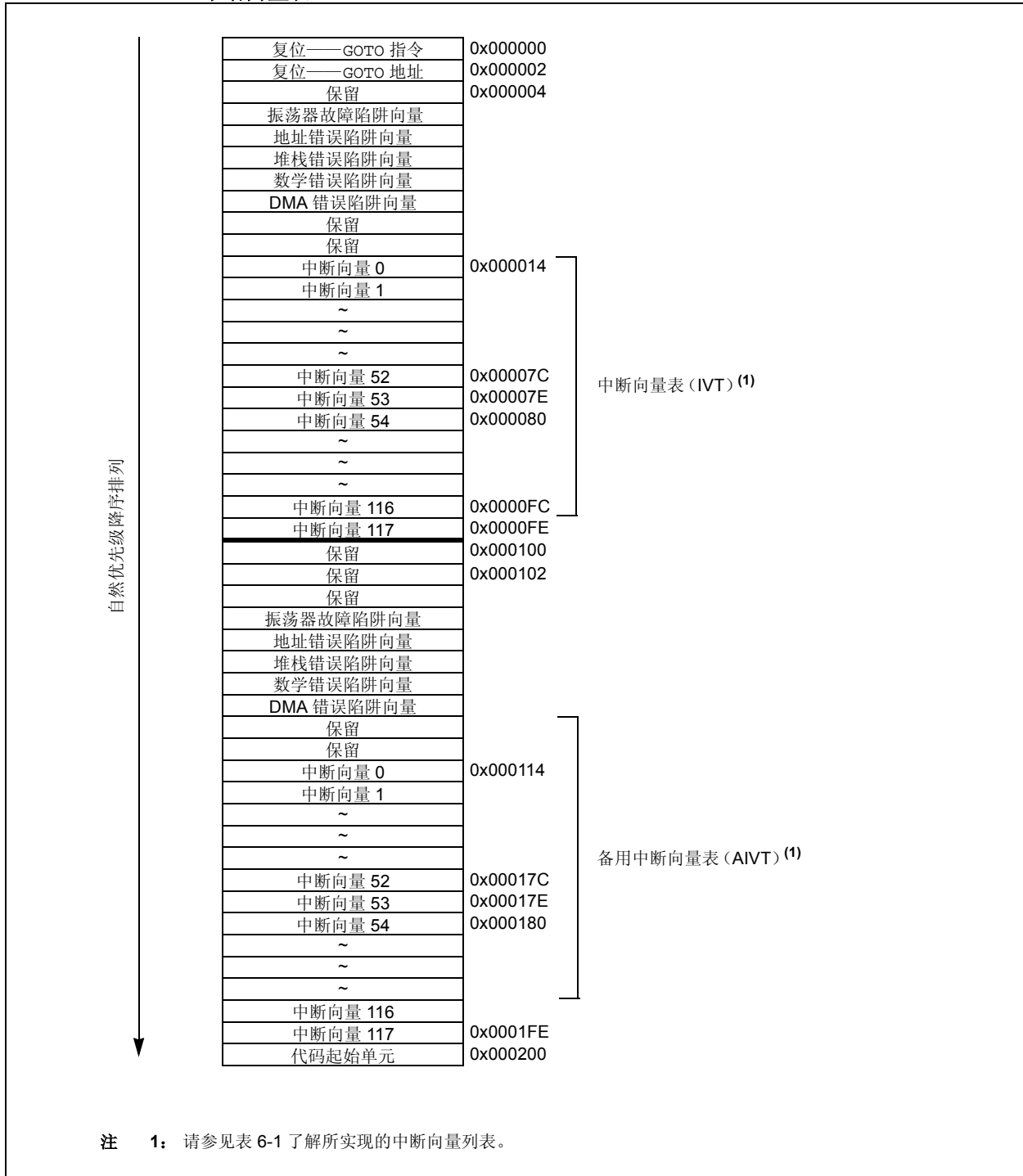


表 6-1: 中断向量

向量编号	IVT 地址	AIVT 地址	中断源
0	0x000004	0x000104	保留
1	0x000006	0x000106	振荡器故障
2	0x000008	0x000108	地址错误
3	0x00000A	0x00010A	堆栈错误
4	0x00000C	0x00010C	数学错误
5	0x00000E	0x00010E	DMA 错误
6	0x000010	0x000110	保留
7	0x000012	0x000112	保留
8	0x000014	0x000114	INT0——外部中断 0
9	0x000016	0x000116	IC1——输入捕捉 1
10	0x000018	0x000118	OC1——输出比较 1
11	0x00001A	0x00011A	T1——Timer1
12	0x00001C	0x00011C	DMA0——DMA 通道 0
13	0x00001E	0x00011E	IC2——输入捕捉 2
14	0x000020	0x000120	OC2——输出比较 2
15	0x000022	0x000122	T2——Timer2
16	0x000024	0x000124	T3——Timer3
17	0x000026	0x000126	SPI1E——SPI1 错误
18	0x000028	0x000128	SPI1——SPI1 传输完成
19	0x00002A	0x00012A	U1RX——UART1 接收器
20	0x00002C	0x00012C	U1TX——UART1 发送器
21	0x00002E	0x00012E	ADC1——ADC 1
22	0x000030	0x000130	DMA1——DMA 通道 1
23	0x000032	0x000132	保留
24	0x000034	0x000134	SI2C1——I2C1 从事件
25	0x000036	0x000136	MI2C1——I2C1 主事件
26	0x000038	0x000138	CM——比较器中断
27	0x00003A	0x00013A	电平变化通知中断
28	0x00003C	0x00013C	INT1——外部中断 1
29	0x00003E	0x00013E	保留
30	0x000040	0x000140	IC7——输入捕捉 7
31	0x000042	0x000142	IC8——输入捕捉 8
32	0x000044	0x000144	DMA2——DMA 通道 2
33	0x000046	0x000146	OC3——输出比较 3
34	0x000048	0x000148	OC4——输出比较 4
35	0x00004A	0x00014A	T4——Timer4
36	0x00004C	0x00014C	T5——Timer5
37	0x00004E	0x00014E	INT2——外部中断 2
38	0x000050	0x000150	U2RX——UART2 接收器
39	0x000052	0x000152	U2TX——UART2 发送器
40	0x000054	0x000154	SPI2E——SPI2 错误
41	0x000056	0x000156	SPI2——SPI2 传输完成
42	0x000058	0x000158	C1RX——ECAN1 接收数据就绪
43	0x00005A	0x00015A	C1——ECAN1 事件
44	0x00005C	0x00015C	DMA3——DMA 通道 3
45	0x00005E	0x00015E	保留
46	0x000060	0x000160	保留

表 6-1: 中断向量 (续)

向量编号	IVT 地址	AIVT 地址	中断源
47	0x000062	0x000162	保留
48	0x000064	0x000164	保留
49	0x000066	0x000166	保留
50	0x000068	0x000168	保留
51	0x00006A	0x00016A	保留
52	0x00006C	0x00016C	保留
53	0x00006E	0x00016E	PMP——并行主端口
54	0x000070	0x000170	DMA4——DMA 通道 4
55	0x000072	0x000172	保留
56	0x000074	0x000174	保留
57	0x000076	0x000176	保留
58	0x000078	0x000178	保留
59	0x00007A	0x00017A	保留
60	0x00007C	0x00017C	保留
61	0x00007E	0x00017E	保留
62	0x000080	0x000180	保留
63	0x000082	0x000182	保留
64	0x000084	0x000184	保留
65	0x000086	0x000186	PWM1——PWM1 周期匹配
66	0x000088	0x000188	QE11——位置计数器比较
67	0x00008A	0x00018A	保留
68	0x00008C	0x00018C	保留
69	0x00008E	0x00018E	DMA5——DMA 通道 5
70	0x000090	0x000190	RTCC——实时时钟
71	0x000092	0x000192	FLTA1——PWM1 故障 A
72	0x000094	0x000194	保留
73	0x000096	0x000196	U1E——UART1 错误
74	0x000098	0x000198	U2E——UART2 错误
75	0x00009A	0x00019A	CRC——CRC 发生器中断
76	0x00009C	0x00019C	DMA6——DMA 通道 6
77	0x00009E	0x00019E	DMA7——DMA 通道 7
78	0x0000A0	0x0001A0	C1TX——ECAN1 发送数据请求
79	0x0000A2	0x0001A2	保留
80	0x0000A4	0x0001A4	保留
81	0x0000A6	0x0001A6	PWM2——PWM2 周期匹配
82	0x0000A8	0x0001A8	FLTA2——PWM2 故障 A
83	0x0000AA	0x0001AA	QE12——位置计数器比较
84	0x0000AC	0x0001AC	保留
85	0x0000AE	0x0001AE	保留
86	0x0000B0	0x0001B0	DAC1R——DAC1 右声道数据请求
87	0x0000B2	0x0001B2	DAC1L——DAC1 左声道数据请求
88-126	0x0000B4-0x0000FE	0x0001B4-0x0001FE	保留

## 6.3 中断控制和状态寄存器

dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件共有 30 个用于中断控制器的寄存器：

- INTCON1
- INTCON2
- IFSx
- IECx
- IPCx
- INTTREG

### 6.3.1 INTCON1 和 INTCON2

INTCON1 和 INTCON2 控制全局中断。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以及备用中断向量表的使用。

### 6.3.2 IFS0–IFS4

IFS 寄存器维护所有中断请求标志。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，而由软件清零。

### 6.3.3 IEC0–IEC4

IEC 寄存器维护所有中断允许位。这些控制位用于单独允许外设或外部信号中断。

### 6.3.4 IPC0–IPC19

IPC 寄存器用于设置每个中断源的中断优先级。可以将每个用户中断源分配为 8 个优先级之一。

### 6.3.5 INTTREG

INTTREG 寄存器包含相关的中断向量编号和新的 CPU 中断优先级，分别锁存在 INTTREG 寄存器中的向量编号 (VECNUM<6:0>) 和中断优先级 (ILR<3:0>) 位域中。新的中断优先级是等待处理中断的优先级。

中断源按表 6-1 中的顺序分配给 IFSx、IECx 和 IPCx 寄存器。例如，INT0 (外部中断 0) 表示为向量编号为 8、自然优先级为 0 的外部中断。所以 INTOIF 位在 IFS0<0> 中，INT0IE 位在 IEC0<0> 中，INT0IP 位在 IPC0 的第一个位位置 (IPC0<2:0>) 中。

### 6.3.6 状态 / 控制寄存器

尽管这两个 CPU 控制寄存器不是中断控制硬件的特定组成部分，但它们仍包含控制中断功能的位。

- CPU 状态寄存器 SR 包含 IPL<2:0> 位 (SR<7:5>)。这些位表示当前 CPU 中断优先级。用户软件可以通过写 IPL 位来更改当前 CPU 优先级。
- CORCON 寄存器包含 IPL3 位，这个位与 IPL<2:0> 位一起表示当前 CPU 优先级。IPL3 是只读位，所以用户软件无法屏蔽陷阱事件。

在下面各页中的寄存器 6-1 到寄存器 6-32 描述了所有的中断寄存器。

**寄存器 6-1: SR: CPU 状态寄存器 (1)**

R-0	R-0	R/C-0	R/C-0	R-0	R/C-0	R-0	R/W-0
OA	OB	SA	SB	OAB	SAB	DA	DC
bit 15						bit 8	
R/W-0 <sup>(3)</sup>	R/W-0 <sup>(3)</sup>	R/W-0 <sup>(3)</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:**

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

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

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: 如需了解整个寄存器的详细信息, 请参见寄存器 2-1: “SR: CPU 状态寄存器”。
- 2: IPL<2:0> 位与 IPL<3> 位 (CORCON<3>) 组合形成 CPU 中断优先级。如果 IPL<3> = 1, 那么括号中的值表示 IPL。当 IPL<3> = 1 时, 禁止用户中断。
- 3: 当 NSTDIS (INTCON1<15>) = 1 时, IPL<2:0> 状态位是只读的。

**寄存器 6-2: CORCON: 内核控制寄存器 (1)**

U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-0	R-0
—	—	—	US	EDT	DL<2:0>		
bit 15						bit 8	
R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R/W-0	R/W-0	R/W-0
SATA	SATB	SATDW	ACCSAT	IPL3 <sup>(2)</sup>	PSV	RND	IF
bit 7						bit 0	

**图注:**

R = 可读位	C = 只可清零位	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 3 **IPL3**: CPU 中断优先级状态位 3(2)

1 = CPU 中断优先级大于 7  
 0 = CPU 中断优先级等于或小于 7

- 注 1: 如需了解整个寄存器的详细信息, 请参见寄存器 2-2: “CORCON: 内核控制寄存器”。
- 2: IPL3 位与 IPL<2:0> 位 (SR<7:5>) 组合形成 CPU 的中断优先级。



寄存器 6-3: INTCON1: 中断控制寄存器 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
NSTDIS	OVAERR	OVBERR	COVAERR	COVBERR	OVATE	OVBTE	COVTE
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
SFTACERR	DIV0ERR	DMACERR	MATHERR	ADDRERR	STKERR	OSCFAIL	—
bit 7							bit 0

图注:

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

- bit 15        **NSTDIS:** 中断嵌套禁止位  
 1 = 禁止中断嵌套  
 0 = 使能中断嵌套
- bit 14        **OVAERR:** 累加器 A 溢出陷阱标志位  
 1 = 陷阱是由累加器 A 溢出引起  
 0 = 陷阱不是由累加器 A 溢出引起
- bit 13        **OVBERR:** 累加器 B 溢出陷阱标志位  
 1 = 陷阱是由累加器 B 溢出引起  
 0 = 陷阱不是由累加器 B 溢出引起
- bit 12        **COVAERR:** 累加器 A 灾难性溢出陷阱标志位  
 1 = 陷阱是由累加器 A 灾难性溢出引起  
 0 = 陷阱不是由累加器 A 灾难性溢出引起
- bit 11        **COVBERR:** 累加器 B 灾难性溢出陷阱标志位  
 1 = 陷阱是由累加器 B 灾难性溢出引起  
 0 = 陷阱不是由累加器 B 灾难性溢出引起
- bit 10        **OVATE:** 累加器 A 溢出陷阱允许位  
 1 = 允许累加器 A 溢出陷阱  
 0 = 禁止陷阱
- bit 9         **OVBTE:** 累加器 B 溢出陷阱允许位  
 1 = 允许累加器 B 溢出陷阱  
 0 = 禁止陷阱
- bit 8         **COVTE:** 灾难性溢出陷阱允许位  
 1 = 允许累加器 A 或 B 的灾难性溢出陷阱  
 0 = 禁止陷阱
- bit 7         **SFTACERR:** 累加器移位错误状态位  
 1 = 数学错误陷阱是由非法的累加器移位引起的  
 0 = 数学错误陷阱不是由非法的累加器移位引起的
- bit 6         **DIV0ERR:** 算术错误状态位  
 1 = 数学错误陷阱是由被零除引起的  
 0 = 数学错误陷阱不是由被零除引起的
- bit 5         **DMACERR:** DMA 控制器错误状态位  
 1 = 发生了 DMA 控制器错误陷阱  
 0 = 未发生 DMA 控制器错误陷阱
- bit 4         **MATHERR:** 算术错误状态位  
 1 = 发生了数学错误陷阱  
 0 = 未发生数学错误陷阱

**寄存器 6-3: INTCON1: 中断控制寄存器 1 (续)**

bit 3	<b>ADDRERR:</b> 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	<b>STKERR:</b> 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	<b>OSCFAIL:</b> 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	<b>未实现:</b> 读为 0

寄存器 6-4: INTCON2: 中断控制寄存器 2

R/W-0	R-0	U-0	U-0	U-0	U-0	U-0	U-0
ALTIVT	DISI	—	—	—	—	—	—
bit 15						bit 8	
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	INT2EP	INT1EP	INT0EP
bit 7						bit 0	

图注:

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

- bit 15            **ALTIVT:** 备用中断向量表使能位  
 1 = 使用备用向量表  
 0 = 使用标准 (默认) 向量表
- bit 14            **DISI:** DISI 指令状态位  
 1 = 执行了 DISI 指令  
 0 = 未执行 DISI 指令
- bit 13-3        **未实现:** 读为 0
- 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

**寄存器 6-5: 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	SPI1IF	SPI1EIF	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 数据传输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 13        **AD1IF:** ADC1 转换完成中断标志状态位  
               1 = 产生了中断请求  
               0 = 未产生中断请求
- bit 12        **U1TXIF:** UART1 发送器中断标志状态位  
               1 = 产生了中断请求  
               0 = 未产生中断请求
- bit 11        **U1RXIF:** UART1 接收器中断标志状态位  
               1 = 产生了中断请求  
               0 = 未产生中断请求
- bit 10        **SPI1IF:** SPI1 事件中断标志状态位  
               1 = 产生了中断请求  
               0 = 未产生中断请求
- bit 9         **SPI1EIF:** SPI1 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 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 数据传输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 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 6-5: 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

**寄存器 6-6: 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
R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IC8IF	IC7IF	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF
bit 7							bit 0

**图注:**

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

- bit 15        **U2TXIF:** UART2 发送器中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 14        **U2RXIF:** UART2 接收器中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 13        **INT2IF:** 外部中断 2 标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 12        **T5IF:** Timer5 中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 11        **T4IF:** Timer4 中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 10        **OC4IF:** 输出比较通道 4 中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 9         **OC3IF:** 输出比较通道 3 中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 8         **DMA2IF:** DMA 通道 2 数据传输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         **IC8IF:** 输入捕捉通道 8 中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 6         **IC7IF:** 输入捕捉通道 7 中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 5         **未实现:** 读为 0
- bit 4         **INT1IF:** 外部中断 1 标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 3         **CNIF:** 输入电平变化通知中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求

**寄存器 6-6: 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

**寄存器 6-7: IFS2: 中断标志状态寄存器 2**

U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	DMA4IF	PMPIF	—	—	—	—	—
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
—	—	—	DMA3IF	C1IF <sup>(1)</sup>	C1RXIF <sup>(1)</sup>	SPI2IF	SPI2EIF
bit 7						bit 0	

**图注:**

R = 可读位                      W = 可写位                      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 数据传输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 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 12-5        **未实现:** 读为 0
- bit 4             **DMA3IF:** DMA 通道 3 数据传输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 3             **C1IF:** ECAN1 事件中断标志状态位 <sup>(1)</sup>  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 2             **C1RXIF:** ECAN1 接收数据就绪中断标志状态位 <sup>(1)</sup>  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 1             **SPI2IF:** SPI2 事件中断标志状态位  
 1 = 产生了中断请求  
 0 = 未产生中断请求
- bit 0             **SPI2EIF:** SPI2 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

**注 1:** 对于不带 ECAN™ 模块的器件禁止此类中断。



寄存器 6-8: IFS3: 中断标志状态寄存器 3

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
FLTA1IF	RTCIF	DMA5IF	—	—	QE11IF	PWM1IF	—
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            **FLTA1IF:** PWM1 故障 A 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            **RTCIF:** 实时时钟 / 日历中断标志状态位  
                  1 = 产生了中断请求  
                  0 = 未产生中断请求
- bit 13            **DMA5IF:** DMA 通道 5 数据传输完成中断标志状态位  
                  1 = 产生了中断请求  
                  0 = 未产生中断请求
- bit 12-11        **未实现:** 读为 0
- bit 10            **QE11IF:** QE11 事件中断标志状态位  
                  1 = 产生了中断请求  
                  0 = 未产生中断请求
- bit 9             **PWM1IF:** PWM1 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 8-0          **未实现:** 读为 0

**寄存器 6-9: IFS4: 中断标志状态寄存器 4**

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
DAC1LIF <sup>(2)</sup>	DAC1RIF <sup>(2)</sup>	—	—	QEI2IF	FLTA2IF	PWM2IF	—
bit 15							bit 8
U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	C1TXIF <sup>(1)</sup>	DMA7IF	DMA6IF	CRCIF	U2EIF	U1EIF	—
bit 7							bit 0

**图注:**

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

- bit 15      **DAC1LIF:** DAC 左声道中断标志状态位 <sup>(2)</sup>  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 14      **DAC1RIF:** DAC 右声道中断标志状态位 <sup>(2)</sup>  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 13-12   **未实现:** 读为 0
- bit 11      **QEI2IF:** QEI2 事件中断标志状态位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 10      **FLTA2IF:** PWM2 故障 A 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        **PWM2IF:** PWM2 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 8-7     **未实现:** 读为 0
- bit 6        **C1TXIF:** ECAN1 接收数据就绪中断标志状态位 <sup>(1)</sup>  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 5        **DMA7IF:** DMA 通道 7 数据传输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 4        **DMA6IF:** DMA 通道 6 数据传输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 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        **U2EIF:** 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        **U1EIF:** 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

**注 1:** 对于不带 ECAN™ 模块的器件禁止此类中断。  
**注 2:** 对于不带 DAC 模块的器件禁止此类中断。

寄存器 6-10: 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	SPI1IE	SPI1EIE	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 数据传输完成中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 13            **AD1IE:** ADC1 转换完成中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 12            **U1TXIE:** UART1 发送器中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 11            **U1RXIE:** UART1 接收器中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 10            **SPI1IE:** SPI1 事件中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 9             **SPI1EIE:** SPI1 错误中断允许位  
 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 数据传输完成中断允许位  
 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

**寄存器 6-10: 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 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求

寄存器 6-11: 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	T5IE	T4IE	OC4IE	OC3IE	DMA2IE
bit 15							bit 8
R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IC8IE	IC7IE	—	INT1IE	CNIE	CMIE	MI2C1IE	SI2C1IE
bit 7							bit 0

图注:

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

- bit 15        **U2TXIE:** UART2 发送器中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 14        **U2RXIE:** UART2 接收器中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 13        **INT2IE:** 外部中断 2 允许位  
 1 = 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 数据传输完成中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 7         **IC8IE:** 输入捕捉通道 8 中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 6         **IC7IE:** 输入捕捉通道 7 中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 5         **未实现:** 读为 0
- bit 4         **INT1IE:** 外部中断 1 允许位  
 1 = 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

**寄存器 6-11: IEC1: 中断允许控制寄存器 1 (续)**

- 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
- bit 1      **MI2C1IE:** I2C1 主事件中中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 0      **SI2C1IE:** I2C1 从事件中中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求

寄存器 6-12: IEC2: 中断允许控制寄存器 2

U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	DMA4IE	PMPIE	—	—	—	—	—
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
—	—	—	DMA3IE	C1IE <sup>(1)</sup>	C1RXIE <sup>(1)</sup>	SPI2IE	SPI2EIE
bit 7						bit 0	

图注:

R = 可读位                      W = 可写位                      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 数据传输完成中断允许位  
 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 12-5        **未实现:** 读为 0
- bit 4             **DMA3IE:** DMA 通道 3 数据传输完成中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 3             **C1IE:** ECAN1 事件中断允许位 <sup>(1)</sup>  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 2             **C1RXIE:** ECAN1 接收数据就绪中断允许位 <sup>(1)</sup>  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 1             **SPI2IE:** SPI2 事件中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 0             **SPI2EIE:** SPI2 错误中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求

注 1: 对于不带 ECAN™ 模块的器件禁止此类中断。

**寄存器 6-13: IEC3: 中断允许控制寄存器 3**

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
FLTA1IE	RTCIE	DMA5IE	—	—	QE11IE	PWM1IE	—
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        **FLTA1IE:** PWM1 故障 A 中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 14        **RTCIE:** 实时时钟 / 日历中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 13        **DMA5IE:** DMA 通道 5 数据传输完成中断允许位  
 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        **QE11IE:** QE11 事件中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 9         **PWM1IE:** PWM1 错误中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 8-0       **未实现:** 读为 0



**寄存器 6-14: IEC4: 中断允许控制寄存器 4**

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
DAC1LIE <sup>(2)</sup>	DAC1RIE <sup>(2)</sup>	—	—	QE12IE	FLTA2IE	PWM2IE	—
bit 15							bit 8
U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	C1TXIE <sup>(1)</sup>	DMA7IE	DMA6IE	CRCIE	U2EIE	U1EIE	—
bit 7							bit 0

**图注:**

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

- bit 15            **DAC1LIE:** DAC 左声道中断允许位 <sup>(2)</sup>  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 14            **DAC1RIE:** DAC 右声道中断允许位 <sup>(2)</sup>  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 13-12        **未实现:** 读为 0
- bit 11            **QE12IE:** QE12 事件中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 10            **FLTA2IE:** PWM2 故障 A 中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 9             **PWM2IE:** PWM2 错误中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 8-7          **未实现:** 读为 0
- bit 6             **C1TXIE:** ECAN1 接收数据就绪中断允许位 <sup>(1)</sup>  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 5             **DMA7IE:** DMA 通道 7 数据传输完成中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 4             **DMA6IE:** DMA 通道 6 数据传输完成中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- 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             **U2EIE:** UART2 错误中断允许位  
 1 = 允许中断请求  
 0 = 禁止中断请求
- bit 1             **U1EIE:** 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

**注 1:** 对于不带 ECAN™ 模块的器件禁止此类中断。  
**注 2:** 对于不带 DAC 模块的器件禁止此类中断。

**寄存器 6-15: 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
—	T1IP<2:0>			—	OC1IP<2:0>		
bit 15				bit 8			

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC1IP<2:0>			—	INT0IP<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 14-12       **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

寄存器 6-16: 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
—	T2IP<2:0>			—	OC2IP<2:0>		
bit 15				bit 8			

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC2IP<2:0>			—	DMA0IP<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 14-12       **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 数据传输完成中断优先级位
  - 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

**寄存器 6-17: 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
—	U1RXIP<2:0>			—	SPI1IP<2:0>		
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPI1EIP<2:0>			—	T3IP<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 14-12       **U1RXIP<2:0>:** UART1 接收器中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源
- bit 11           **未实现:** 读为 0
- bit 10-8        **SPI1IP<2:0>:** SPI1 事件中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源
- bit 7            **未实现:** 读为 0
- bit 6-4         **SPI1EIP<2:0>:** SPI1 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            **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

寄存器 6-18: 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
—	—	—	—	—	DMA1IP<2:0>		
bit 15					bit 8		
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	AD1IP<2:0>		—	U1TXIP<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 15-11      **未实现:** 读为 0
- bit 10-8      **DMA1IP<2:0>:** DMA 通道 1 数据传输完成中断优先级位
  - 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 7          **未实现:** 读为 0
- bit 6-4      **AD1IP<2:0>:** ADC1 转换完成中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源
- bit 3          **未实现:** 读为 0
- bit 2-0      **U1TXIP<2:0>:** UART1 发送器中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源

**寄存器 6-19: 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
—	CNIP<2:0>			—	CMIP<2:0>		
bit 15				bit 8			
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	MI2C1IP<2:0>			—	SI2C1IP<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 14-12       **CNIP<2:0>:** 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

寄存器 6-20: IPC5: 中断优先级控制寄存器 5

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC8IP<2:0>			—	IC7IP<2:0>		
bit 15				bit 8			

U-0	U-1	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            未实现: 读为 0
- bit 14-12       **IC8IP<2:0>**: 输入捕捉通道 8 中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源
- bit 11           未实现: 读为 0
- bit 10-8        **IC7IP<2:0>**: 输入捕捉通道 7 中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源
- bit 7-3         未实现: 读为 0
- bit 2-0         **INT1IP<2:0>**: 外部中断 1 优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源

**寄存器 6-21: 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
—	T4IP<2:0>			—	OC4IP<2:0>		
bit 15				bit 8			
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC3IP<2:0>			—	DMA2IP<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 14-12       **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 数据传输完成中断优先级位
  - 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



寄存器 6-22: 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
—	U2TXIP<2:0>			—	U2RXIP<2:0>		
bit 15				bit 8			
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	INT2IP<2:0>			—	T5IP<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 14-12                **U2TXIP<2:0>:** UART2 发送器中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源
- bit 11                      **未实现:** 读为 0
- bit 10-8                **U2RXIP<2:0>:** UART2 接收器中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源
- bit 7                      **未实现:** 读为 0
- bit 6-4                **INT2IP<2:0>:** 外部中断 2 优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源
- bit 3                      **未实现:** 读为 0
- bit 2-0                **T5IP<2:0>:** Timer5 中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源

**寄存器 6-23: 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
—	C1IP<2:0> <sup>(1)</sup>			—	C1RXIP<2:0> <sup>(1)</sup>		
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
—	SPI2IP<2:0>			—	SPI2EIP<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 14-12       **C1IP<2:0>:** ECAN1 事件中断优先级位 <sup>(1)</sup>  
 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        **C1RXIP<2:0>:** ECAN1 接收数据就绪中断优先级位 <sup>(1)</sup>  
 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         **SPI2IP<2:0>:** SPI2 事件中断优先级位  
 111 = 中断优先级为 7 (最高优先级中断)  
 .  
 .  
 .  
 001 = 中断优先级为 1  
 000 = 禁止中断源
- bit 3            **未实现:** 读为 0
- bit 2-0         **SPI2EIP<2:0>:** SPI12 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

**注 1:** 对于不带 ECAN™ 模块的器件禁止此类中断。

寄存器 6-24: IPC9: 中断优先级控制寄存器 9

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
—	—	—	—	—	DMA3IP<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      **DMA3IP<2:0>:** DMA 通道 3 数据传输完成中断优先级位  
               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

**寄存器 6-25: 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
—	—	—	—	—	DMA4IP<2:0>		
bit 15					bit 8		
U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	PMPPIP<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 15-11      **未实现:** 读为 0
- bit 10-8      **DMA4IP<2:0>:** DMA 通道 4 数据传输完成中断优先级位
  - 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 7          **未实现:** 读为 0
- bit 6-4      **PMPPIP<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

寄存器 6-26: IPC14: 中断优先级控制寄存器 14

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	QE11IP<2:0>		
bit 15					bit 8		
U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	PWM1IP<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 15-11      **未实现:** 读为 0
- bit 10-8      **QE11IP<2:0>:** QE11 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      **PWM1IP<2:0>:** PWM1 中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源
- bit 3-0      **未实现:** 读为 0

**寄存器 6-27: IPC15: 中断优先级控制寄存器 15**

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	FLTA1IP<2:0>			—	RTCIP<2:0>		
bit 15				bit 8			

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	DMA5IP<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 14-12       **FLTA1IP<2:0>:** PWM1 故障 A 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        **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 数据传输完成中断优先级位  
                   111 = 中断优先级为 7 (最高优先级中断)  
                   .  
                   .  
                   .  
                   001 = 中断优先级为 1  
                   000 = 禁止中断源

bit 3-0           **未实现:** 读为 0

寄存器 6-28: 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
—	CRCIP<2:0>			—	U2EIP<2:0>		
bit 15				bit 8			
U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1EIP<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 14-12                **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                **U2EIP<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                **U1EIP<2:0>**: UART1 错误中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源
- bit 3-0                 未实现: 读为 0

**寄存器 6-29: IPC17: 中断优先级控制寄存器 17**

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	C1TXIP<2:0> <sup>(1)</sup>		
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
—	DMA7IP<2:0>			—	DMA6IP<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 15-11      **未实现:** 读为 0
- bit 10-8      **C1TXIP<2:0>:** ECAN1 发送数据请求中断优先级位 <sup>(1)</sup>
  - 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      **DMA7IP<2:0>:** DMA 通道 7 数据传输完成中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源
- bit 3          **未实现:** 读为 0
- bit 2-0      **DMA6IP<2:0>:** DMA 通道 6 数据传输完成中断优先级位
  - 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

**注 1:** 对于不带 ECAN™ 模块的器件禁止此类中断。



寄存器 6-30: IPC18: 中断优先级控制寄存器 18

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	QEI2IP<2:0>			—	FLTA2IP<2:0>		
bit 15				bit 8			

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	PWM2IP<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 14-12                **QEI2IP<2:0>**: QEI2 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                **FLTA2IP<2:0>**: PWM2 故障 A 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                **PWM2IP<2:0>**: PWM2 中断优先级位
  - 111 = 中断优先级为 7 (最高优先级中断)
  - 
  - 
  - 
  - 001 = 中断优先级为 1
  - 000 = 禁止中断源
- bit 3-0                 未实现: 读为 0

**寄存器 6-31: IPC19: 中断优先级控制寄存器 19**

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	DAC1LIP<2:0> <sup>(1)</sup>			—	DAC1RIP<2:0> <sup>(1)</sup>		
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            **未实现:** 读为 0
- bit 14-12        **DAC1LIP<2:0>:** DAC 左声道中断优先级位 <sup>(1)</sup>
  - 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         **DAC1RIP<2:0>:** DAC 右声道中断优先级位 <sup>(1)</sup>
  - 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

**注 1:** 对于不带 DAC 模块的器件禁止此类中断。

寄存器 6-32: INTTREG: 中断控制和状态寄存器

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
—	—	—	—	ILR<3:0>			
bit 15				bit 8			
U-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
—	VECNUM<6:0>						
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 15-12      **未实现:** 读为 0

bit 11-8      **ILR:** 新的 CPU 中断优先级位  
 1111 = CPU 中断优先级为 15  
 .  
 .  
 .  
 0001 = CPU 中断优先级为 1  
 0000 = CPU 中断优先级为 0

bit 7          **未实现:** 读为 0

bit 6-0        **VECNUM:** 待处理中断向量编号位  
 0111111 = 待处理中断向量的编号为 135  
 .  
 .  
 .  
 0000001 = 待处理中断向量的编号为 9  
 0000000 = 待处理中断向量的编号为 8

## 6.4 中断设置步骤

### 6.4.1 初始化

要在初始化时配置中断源：

1. 如果不需要嵌套中断，则将 `NSTDIS` 位 (`INTCON1<15>`) 置 1。
2. 通过写相应 `IPCx` 寄存器中的控制位为中断源选择由用户分配的优先级。优先级将取决于具体的应用和中断源类型。如果不需要多个优先级，则可以将所有允许中断源的 `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。

### 6.4.2 中断服务程序

如何声明 `ISR` 以及怎样使用正确的向量地址初始化 `IVT`，将依赖于编程语言（C 语言或汇编语言）和用于开发应用程序的语言开发工具包。

一般情况下，用户应用程序必须将相应 `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 `ISR` 程序后将立即再次进入 `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。

### 6.4.3 陷阱服务程序

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

### 6.4.4 中断禁止

可以使用以下步骤禁止所有用户中断：

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

要允许用户中断，则可以使用 `POP` 指令恢复先前的 `SR` 值。

**注：** 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。

## 7.0 直接存储器访问 (DMA)

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参见“dsPIC33F Family Reference Manual”的“Section 22. Direct Memory Access (DMA)” (DS70182)，该文档可从 Microchip 网站 (www.microchip.com) 下载。

直接存储器访问 (DMA) 是在外设 SFR (如 UART 接收寄存器和输入捕捉 1 缓冲区) 和 RAM 中缓冲区或存储在 RAM 中的变量间复制数据的非常高效的机制，它极少需要 CPU 干预。每次产生外设中断时，DMA 控制器能自动复制整块数据，无需用户软件读或写外设特殊功能寄存器 (SFR)。DMA 控制器使用专用的总线传输数据，因此，不会占用 CPU 的代码执行周期。要使用 DMA 功能，相应的用户缓冲区或变量必须位于 DMA RAM 中。

表 7-1 中列出了可以使用 DMA 的 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 外设。

表 7-1: DMA 通道与外设的关系

外设与 DMA 的关系	DMAxREQ 寄存器 IRQSEL<6:0> 位	要从外设读取的 DMAxPAD 寄存器值	要写入外设的 DMAxPAD 寄存器值
INT0——外部中断 0	0000000	—	—
IC2——输入捕捉 1	0000001	0x0140 (IC1BUF)	—
OC1——输出比较 1 数据	0000010	—	0x0182 (OC1R)
OC1——输出比较 1 辅助数据	0000010	—	0x0180 (OC1RS)
IC2——输入捕捉 2	0000101	0x0144 (IC2BUF)	—
OC2——输出比较 2 数据	0000110	—	0x0188 (OC2R)
OC2——输出比较 2 辅助数据	0000110	—	0x0186 (OC2RS)
TMR2——Timer2	0000111	—	—
TMR3——Timer3	0001000	—	—
SPI1——传输完成	0001010	0x0248 (SPI1BUF)	0x0248 (SPI1BUF)
UART1RX——UART1 接收器	0001011	0x0226 (U1RXREG)	—
UART1TX——UART1 发送器	0001100	—	0x0224 (U1TXREG)
ADC1——ADC1 转换完成	0001101	0x0300 (ADC1BUF0)	—
UART2RX——UART2 接收器	0011110	0x0236 (U2RXREG)	—
UART2TX——UART2 发送器	0011111	—	0x0234 (U2TXREG)
SPI2——传输完成	0100001	0x0268 (SPI2BUF)	0x0268 (SPI2BUF)
ECAN1——接收数据就绪	0100010	0x0440 (C1RXD)	—
PMP——主控数据传输	0101101	0x0608 (PMDIN1)	0x0604 (PMDOUT1)
ECAN1——发送数据就绪	1000110	—	0x0442 (C1TXD)
DAC1——右声道数据输出	1001110	—	0x3F6 (DAC1RDAT)
DAC2——左声道数据输出	1001111	—	0x03F8 (DAC1LDAT)

DMA 控制器具有 8 路相同的数据传输通道。

每路通道都有一组控制和状态寄存器。每路 DMA 通道可配置为将数据从双端口 DMA RAM 中的缓冲区复制到外设 SFR 中，或从外设 SFR 复制到 DMA RAM 中的缓冲区中。

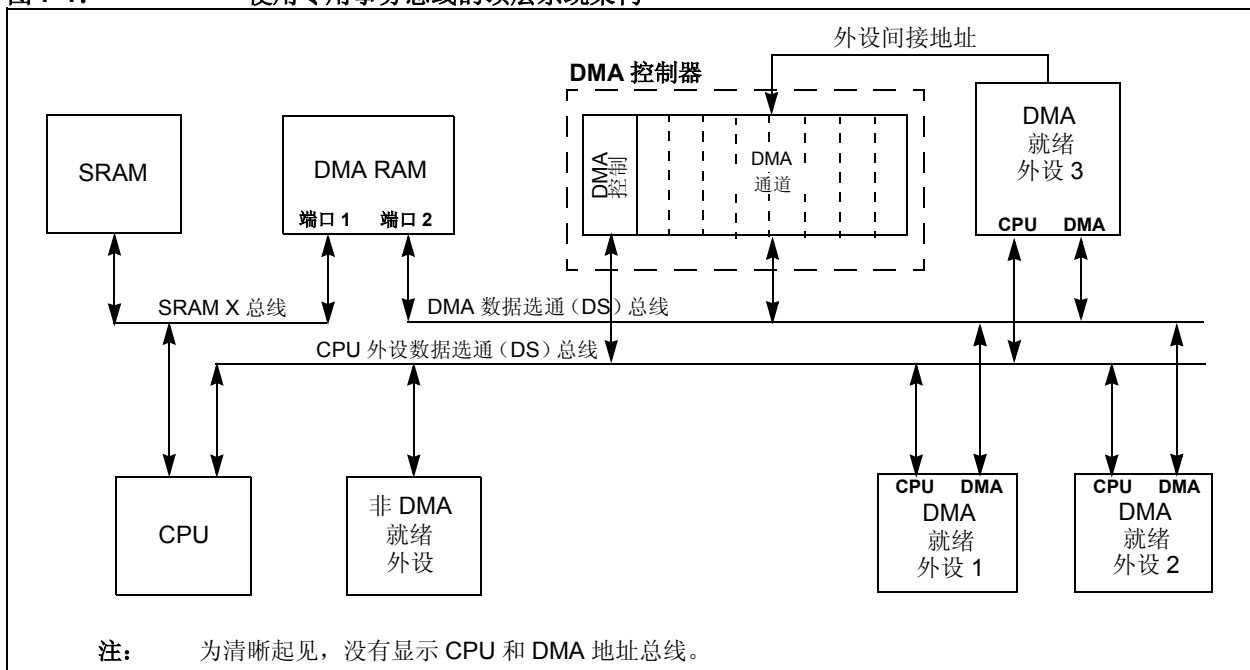
DMA 控制器支持以下功能：

- 8 路 DMA 通道
- 带后递增的寄存器间接寻址模式
- 不带后递增的寄存器间接寻址模式
- 外设间接寻址模式（外设生成目标地址）
- 在传输完一半或整个数据块后产生 CPU 中断

- 字节或字传输
- 固定优先级通道仲裁
- 手动（软件）或自动（外设 DMA 请求）启动传输
- 单数据块或自动重复数据块传输模式
- “乒乓”（Ping-Pong）模式（每个数据块传输完成后，在两个 DPSRAM 起始地址之间进行自动切换）
- 每路通道的 DMA 请求可从任何支持的中断源中选择
- 调试支持功能

对于每路 DMA 通道，在传输完一整块数据后产生一个 DMA 中断请求。也可在填充完整块数据的一半后产生中断。

图 7-1: 使用专用事务总线的顶层系统架构



## 7.1 DMAC 寄存器

每路 DMAC 通道  $x$  ( $x = 0, 1, 2, 3, 4, 5, 6$  或  $7$ ) 均包含以下寄存器:

- 16 位 DMA 通道控制寄存器 (DMAxCON)
- 16 位 DMA 通道 IRQ 选择寄存器 (DMAxREQ)
- 16 位 DMA RAM 主起始地址寄存器 (DMAxSTA)
- 16 位 DMA RAM 辅助起始地址寄存器 (DMAxSTB)
- 16 位 DMA 外设地址寄存器 (DMAxPAD)
- 10 位 DMA 传输计数寄存器 (DMAxCNT)

还有一对状态寄存器 DMACS0 和 DMACS1 是所有 DMAC 通道共用的。DMACS0 包含 DMA RAM 和 SFR 写冲突标志, 分别为 XWCOLx 和 PWCOLx。DMACS1 指示 DMA 通道和乒乓模式的状态。

DMAxCON、DMAxREQ、DMAxPAD 和 DMAxCNT 都是常规的读 / 写寄存器。读 DMAxSTA 或 DMAxSTB 会读取 DMA RAM 地址寄存器的内容。写 DMAxSTA 或 DMAxSTB 会直接写入这两个寄存器。这使用户可以在任何时候确定 DMA 缓冲区的指针值 (地址)。

中断标志 (DMAxIF) 位于中断控制器的 IFSx 寄存器中。对应的中断允许控制位 (DMAxIE) 位于中断控制器的 IECx 寄存器中, 对应的中断优先级控制位 (DMAxIP) 位于中断控制器的 IPCx 寄存器中。

**寄存器 7-1: DMAxCON: DMA 通道 x 控制寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
CHEN	SIZE	DIR	HALF	NULLW	—	—	—
bit 15							bit 8
U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	—	AMODE<1:0>		—	—	MODE<1:0>	
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        **CHEN:** 通道使能位  
               1 = 使能通道  
               0 = 禁止通道
- bit 14        **SIZE:** 数据传输大小位  
               1 = 字节  
               0 = 字
- bit 13        **DIR:** 传输方向位 (源 / 目标总线选择)  
               1 = 从 DMA RAM 地址读取, 写入外设地址  
               0 = 从外设地址读取, 写入 DMA RAM 地址
- bit 12        **HALF:** 数据块传输完成中断选择位  
               1 = 当传送了一半数据时, 发出数据块传输完成中断  
               0 = 当传送了所有数据时, 发出数据块传输完成中断
- bit 11        **NULLW:** 空数据外设写模式选择位  
               1 = 除将外设 SFR 中的数据写入 DMA RAM 外, 还将空数据写入外设 SFR (DIR 位也必须清零)  
               0 = 正常工作
- bit 10-6     **未实现:** 读为 0
- bit 5-4        **AMODE<1:0>:** DMA 通道工作模式选择位  
               11 = 保留 (作为外设间接寻址模式)  
               10 = 外设间接寻址模式  
               01 = 不带后递增的寄存器间接寻址模式  
               00 = 带后递增的寄存器间接寻址模式
- bit 3-2        **未实现:** 读为 0
- bit 1-0        **MODE<1:0>:** DMA 通道工作模式选择位  
               11 = 使能单数据块乒乓模式 (从 / 向每个 DMA RAM 缓冲区传输一块数据)  
               10 = 使能连续数据块乒乓模式  
               01 = 禁止单数据块乒乓模式  
               00 = 禁止连续数据块乒乓模式



寄存器 7-2: DMAxREQ: DMA 通道 x IRQ 选择寄存器

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
FORCE <sup>(1)</sup>	—	—	—	—	—	—	—
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
—	IRQSEL6<6:0> <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            **FORCE:** 强制 DMA 传输位 <sup>(1)</sup>  
 1 = 强制进行单次 DMA 传输 (手动模式)  
 0 = 自动按照 DMA 请求进行 DMA 传输
- bit 14-7        **未实现:** 读为 0
- bit 6-0         **IRQSEL<6:0>:** DMA 外设 IRQ 编号选择位 <sup>(2)</sup>  
 0000000-1111111 = 可选择作为通道 DMAREQ 的 DMAIRQ0-DMAIRQ127

- 注 1: FORCE 位不能被用户清零。当强制的 DMA 传输完成时, FORCE 位由硬件清零。
- 注 2: 请参见表 6-1 获得所有中断源的 IRQ 编号的完整列表。

**寄存器 7-3: DMAxSTA: DMA 通道 x RAM 起始地址寄存器 A<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
STA<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	R/W-0
STA<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 15-0 **STA<15:0>**: 主 DMA RAM 起始地址位 (源地址或目标地址)

**注 1:** 读该地址寄存器会返回 DMA RAM 地址寄存器的当前内容, 而非写入 STA<15:0> 的内容。如果使能了通道 (即通道处于工作状态), 写入该寄存器可能导致 DMA 通道的行为不可预测, 应该避免。

**寄存器 7-4: DMAxSTB: DMA 通道 x RAM 起始地址寄存器 B<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
STB<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	R/W-0
STB<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 15-0 **STB<15:0>**: 辅助 DMA RAM 起始地址位 (源地址或目标地址)

**注 1:** 读该地址寄存器会返回 DMA RAM 地址寄存器的当前内容, 而非写入 STB<15:0> 的内容。如果使能了通道 (即通道处于工作状态), 写入该寄存器可能导致 DMA 通道的行为不可预测, 应该避免。

**寄存器 7-5: DMAxPAD: DMA 通道 x 外设地址寄存器<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
PAD<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	R/W-0
PAD<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 15-0                **PAD<15:0>**: 外设地址寄存器位

**注 1:** 如果使能了通道 (即通道处于工作状态), 写入该寄存器可能导致 DMA 通道的行为不可预测, 应该避免。

**寄存器 7-6: DMAxCNT: DMA 通道 x 传输计数寄存器<sup>(1)</sup>**

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—						CNT<9:8> <sup>(2)</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
CNT<7:0> <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 15-10                **未实现:** 读为 0

bit 9-0                **CNT<9:0>**: DMA 传输计数寄存器位<sup>(2)</sup>

**注 1:** 如果使能了通道 (即通道处于工作状态), 写入该寄存器可能导致 DMA 通道的行为不可预测, 应该避免。

**2:** DMA 传输的次数 = CNT<9:0> + 1。

**寄存器 7-7: DMACS0: DMA 控制器状态寄存器 0**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
PWCOL7	PWCOL6	PWCOL5	PWCOL4	PWCOL3	PWCOL2	PWCOL1	PWCOL0
bit 15							bit 8

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
XWCOL7	XWCOL6	XWCOL5	XWCOL4	XWCOL3	XWCOL2	XWCOL1	XWCOL0
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      **PWCOL7:** 通道 7 外设写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 14      **PWCOL6:** 通道 6 外设写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 13      **PWCOL5:** 通道 5 外设写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 12      **PWCOL4:** 通道 4 外设写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 11      **PWCOL3:** 通道 3 外设写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 10      **PWCOL2:** 通道 2 外设写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 9        **PWCOL1:** 通道 1 外设写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 8        **PWCOL0:** 通道 0 外设写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 7        **XWCOL7:** 通道 7 DMA RAM 写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 6        **XWCOL6:** 通道 6 DMA RAM 写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 5        **XWCOL5:** 通道 5 DMA RAM 写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 4        **XWCOL4:** 通道 4 DMA RAM 写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突

**寄存器 7-7: DMACS0: DMA 控制器状态寄存器 0 (续)**

- bit 3      **XWCOL3:** 通道 3 DMA RAM 写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 2      **XWCOL2:** 通道 2 DMA RAM 写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 1      **XWCOL1:** 通道 1 DMA RAM 写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突
- bit 0      **XWCOL0:** 通道 0 DMA RAM 写冲突标志位  
1 = 检测到写冲突  
0 = 未检测到写冲突

寄存器 7-8: **DMACS1: DMA 控制器状态寄存器 1**

U-0	U-0	U-0	U-0	R-1	R-1	R-1	R-1
—	—	—	—	LSTCH<3:0>			
bit 15				bit 8			

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      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 11-8     **LSTCH<3:0>:** 上一次工作的 DMA 通道位
  - 1111 = 自系统复位以来没有发生 DMA 传输
  - 1110-1000 = 保留
  - 0111 = 上次数据传输是通过 DMA 通道 7 进行的
  - 0110 = 上次数据传输是通过 DMA 通道 6 进行的
  - 0101 = 上次数据传输是通过 DMA 通道 5 进行的
  - 0100 = 上次数据传输是通过 DMA 通道 4 进行的
  - 0011 = 上次数据传输是通过 DMA 通道 3 进行的
  - 0010 = 上次数据传输是通过 DMA 通道 2 进行的
  - 0001 = 上次数据传输是通过 DMA 通道 1 进行的
  - 0000 = 上次数据传输是通过 DMA 通道 0 进行的
- bit 7        **PPST7:** 通道 7 乒乓模式状态标志位
  - 1 = 选择 DMA7STB 寄存器
  - 0 = 选择 DMA7STA 寄存器
- bit 6        **PPST6:** 通道 6 乒乓模式状态标志位
  - 1 = 选择 DMA6STB 寄存器
  - 0 = 选择 DMA6STA 寄存器
- bit 5        **PPST5:** 通道 5 乒乓模式状态标志位
  - 1 = 选择 DMA5STB 寄存器
  - 0 = 选择 DMA5STA 寄存器
- bit 4        **PPST4:** 通道 4 乒乓模式状态标志位
  - 1 = 选择 DMA4STB 寄存器
  - 0 = 选择 DMA4STA 寄存器
- bit 3        **PPST3:** 通道 3 乒乓模式状态标志位
  - 1 = 选择 DMA3STB 寄存器
  - 0 = 选择 DMA3STA 寄存器
- bit 2        **PPST2:** 通道 2 乒乓模式状态标志位
  - 1 = 选择 DMA2STB 寄存器
  - 0 = 选择 DMA2STA 寄存器
- bit 1        **PPST1:** 通道 1 乒乓模式状态标志位
  - 1 = 选择 DMA1STB 寄存器
  - 0 = 选择 DMA1STA 寄存器
- bit 0        **PPST0:** 通道 0 乒乓模式状态标志位
  - 1 = 选择 DMA0STB 寄存器
  - 0 = 选择 DMA0STA 寄存器

**寄存器 7-9: DSADR: 最近的 DMA RAM 地址**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
DSADR<15:8>							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
DSADR<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 15-0 **DSADR<15:0>**: DMA 控制器最近访问的 DMA RAM 地址位

注:



## 8.0 振荡器配置

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 7. Oscillator”（DS70186），该文档可从 Microchip 网站（www.microchip.com）下载。

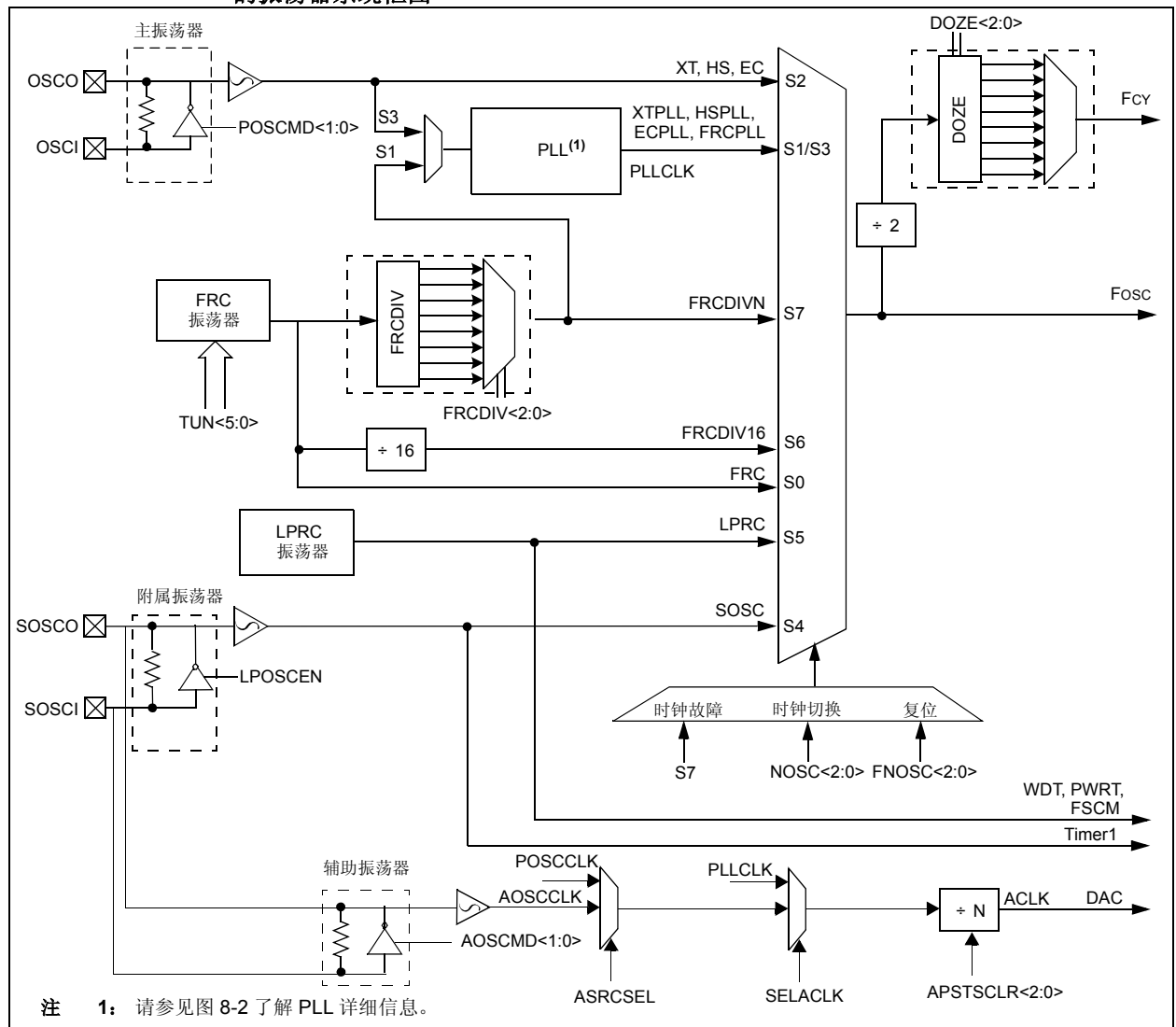
- 片上锁相环（PLL）可将内部工作频率调整为所要求的系统时钟频率
- 内部 FRC 振荡器也可使用 PLL，因此允许在没有任何外部时钟产生硬件的情况下全速工作
- 各种时钟源之间的时钟切换
- 有助于节省系统功耗的可编程时钟后分频器
- 故障保护时钟监视器（FSCM），可检测时钟故障并采取故障保护措施
- 一个时钟控制寄存器（OSCCON）
- 用于主振荡器选择的非易失性配置位
- 一个用于音频 DAC 的附属晶振

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的振荡器系统提供：

- 可选择多种外部和内部振荡器作为时钟源

图 8-1 给出了振荡器系统的简化框图。

**图 8-1: dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的振荡器系统框图**



## 8.1 CPU 时钟系统

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件提供 7 种系统时钟选择：

- 快速 RC (FRC) 振荡器
- 带锁相环 (PLL) 的 FRC 振荡器
- 主 (XT、HS 或 EC) 振荡器
- 带 PLL 的主振荡器
- 辅助 (LP) 振荡器
- 低功耗 RC (LPRC) 振荡器
- 带后分频器的 FRC 振荡器

### 8.1.1 系统时钟源

内部快速 RC (FRC) 振荡器工作频率的标称值为 7.37 MHz。用户软件可以调节 FRC 频率。用户软件能够有选择地指定 FRC 时钟的分频比 (从 1:2 至 1:256)。使用  $FRCDIV<2:0>$  (CLKDIV<10:8>) 位来选择该分频比。

主振荡器能以下列任一时钟作为其时钟源：

- 晶振 (XT)：3 MHz 至 10 MHz 范围内的晶振和陶瓷谐振器。晶振连接在 OSC1 和 OSC2 引脚之间。
- 高速晶振 (HS)：10 MHz 至 40 MHz 范围内的晶振。晶振连接在 OSC1 和 OSC2 引脚之间。
- 外部时钟 (EC)：0.8 MHz 至 64 MHz 范围内的外部时钟信号。外部时钟信号直接施加到 OSC1 引脚。

辅助 (LP) 振荡器是为低功耗运行而设计的，它使用 32.768 kHz 晶振或陶瓷谐振器。LP 振荡器使用 SOSC1 和 SOSCO 引脚。

内部低功耗 RC (LPRC) 振荡器工作频率的标称值为 32.768 kHz。它也用作看门狗定时器 (WDT) 和故障保护时钟监视器 (FSCM) 的参考时钟。

可选择将 FRC 和主振荡器产生的时钟信号加到片上锁相环 (PLL)，为器件工作提供宽范围的输出频率。PLL 配置在第 8.1.4 节 “PLL 配置” 中讲述。

### 8.1.2 系统时钟选择

通过设置配置位可选择器件发生上电复位事件时使用的振荡器源。振荡器配置位设置通过程序存储器中的配置寄存器进行。(更多详细信息，请参见第 27.1 节 “配置位”。) 初始振荡器选择配置位  $FNOSC<2:0>$  ( $FOSCSEL<2:0>$ ) 和主振荡器模式选择配置位  $POSCMD<1:0>$  ( $FOSC<1:0>$ ) 选择在上电复位时使用的振荡器源。FRC 主振荡器是默认的 (未编程) 选择。

配置位允许用户在 12 种不同的时钟模式之间进行选择，如表 8-1 所示。

振荡器的输出 (或当选择了 PLL 模式时 PLL 的输出)  $Fosc$  被 2 分频以产生器件指令时钟 ( $Fcy$ )。Fcy 定义器件的工作速度，dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 架构可支持最高 40 MHz 的工作速度。

指令执行速度或器件工作频率  $Fcy$  按如下公式计算：

#### 公式 8-1: 器件工作频率

$$Fcy = Fosc/2$$

### 8.1.3 附属振荡器

附属振荡器 (AOSC) 可用于需要在与系统时钟无关的频率下工作的外设，例如 DAC。

附属振荡器能以下列任一时钟作为其时钟源：

晶振 (XT)：3 MHz 至 10 MHz 范围内的晶振和陶瓷谐振器。晶振连接在 SOSC1 和 SOSCO 引脚之间。

高速晶振 (HS)：10 MHz 至 40 MHz 范围内的晶振。晶振连接在 SOSC1 和 SOSCO 引脚之间。

外部时钟 (EC)：外部时钟信号频率最高为 64 MHz。外部时钟信号直接施加到 SOSC1 引脚。

8.1.4 PLL 配置

主振荡器和内部 FRC 振荡器可选择使用片上 PLL 来获取更高的工作速度。PLL 在选择器件工作速度方面提供很大的灵活性。图 8-2 给出了 PLL 的框图。

以  $F_{IN}$  表示的主振荡器或 FRC 的输出在提供给 PLL 的压控振荡器 (Voltage Controlled Oscillator, VCO) 之前被预分频因子 ( $N1$ ) 2、3、……或 33 分频。VCO 的输入必须在 0.8 MHz 到 8 MHz 的范围内进行选择。使用 PLLPRE<4:0> 位 (CLKDIV<4:0>) 来选择预分频因子 “ $N1$ ”。

可由 PLLDIV<8:0> 位 (PLLFBD<8:0>) 选择 PLL 反馈倍频比, 提供可使 VCO 的输入倍频的因子 “ $M$ ”。必须选择该因子以使产生的 VCO 输出频率在 100 MHz 到 200 MHz 范围内。

VCO 输出进一步被后分频因子 “ $N2$ ” 分频。使用 PLLPOST<1:0> 位 (CLKDIV<7:6>) 来选择该因子。“ $N2$ ” 可以是 2、4 或 8, 必须选择该因子, 以使 PLL 输出频率 ( $F_{osc}$ ) 在 12.5 MHz 到 80 MHz 范围内, 以产生 6.25-40 MIPS 的器件工作速度。

对于主振荡器或 FRC 振荡器, 输出为  $F_{IN}$ , 则 PLL 的输出  $F_{osc}$  按以下公式计算:

公式 8-2:  $F_{osc}$  计算

$$F_{osc} = F_{IN} \cdot \left( \frac{M}{N1 \cdot N2} \right)$$

例如, 假设正在使用 10 MHz 晶振, 所选的振荡器模式为带 PLL 的 XT。

- 如果 PLLPRE<4:0> = 0, 那么  $N1 = 2$ 。这将产生频率为  $10/2 = 5$  MHz 的 VCO 输入, 这一频率在 0.8-8 MHz 的可接受范围内。
- 如果 PLLDIV<8:0> = 0x1E, 那么  $M = 32$ 。这将产生频率为  $5 \times 32 = 160$  MHz 的 VCO 输出, 这一频率在所需的 100-200 MHz 范围内。
- 如果 PLLPOST<1:0> = 0, 那么  $N2 = 2$ 。这提供  $160/2 = 80$  MHz 的  $F_{osc}$ 。产生的器件工作速度为  $80/2 = 40$  MIPS。

公式 8-3: 带 PLL 的 XT 模式示例

$$F_{CY} = \frac{F_{osc}}{2} = \frac{1}{2} \left( \frac{10000000 \cdot 32}{2 \cdot 2} \right) = 40 \text{ MIPS}$$

图 8-2: dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的 PLL 框图

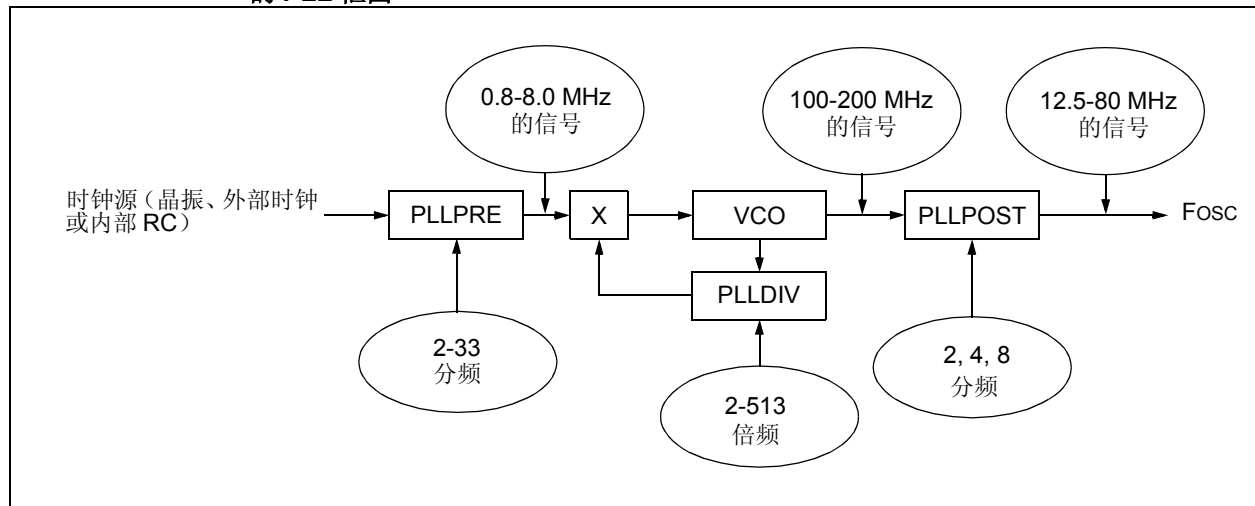


表 8-1: 时钟选择的配置位值

振荡器模式	振荡器源	POSCMD<1:0>	FNOSC<2:0>	注
带 N 分频的快速 RC 振荡器 (FRCDIVN)	内部	xx	111	1, 2
带 16 分频的快速 RC 振荡器 (FRCDIV16)	内部	xx	110	1
低功耗 RC 振荡器 (LPRC)	内部	xx	101	1
辅助 (Timer1) 振荡器 (SOSC)	辅助	xx	100	1
带 PLL 的主振荡器 (HS) (HSPLL)	主	10	011	
带 PLL 的主振荡器 (XT) (XTPLL)	主	01	011	
带 PLL 的主振荡器 (EC) (ECPLL)	主	00	011	1
主振荡器 (HS)	主	10	010	
主振荡器 (XT)	主	01	010	
主振荡器 (EC)	主	00	010	1
带 PLL 的快速 RC 振荡器 (FRCPLL)	内部	xx	001	1
快速 RC 振荡器 (FRC)	内部	xx	000	1

注 1: OSC2 引脚功能由 OSCIOFNC 配置位决定。

2: 对于未编程 (已擦除) 器件, 这是默认的振荡器模式。

寄存器 8-1: **OSCCON: 振荡器控制寄存器**

U-0	R-0	R-0	R-0	U-0	R/W-y	R/W-y	R/W-y
—	COSC<2:0>			—	NOSC<2:0>		
bit 15				bit 8			
R/W-0	R/W-0	R-0	U-0	R/C-0	U-0	R/W-0	R/W-0
CLKLOCK	IOLOCK	LOCK	—	CF	—	LPOSCEN	OSWEN
bit 7				bit 0			

**图注:** y = 在 POR 时由配置位设置的值  
 R = 可读位                      W = 可写位                      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 (只读)
  - 000 = 快速 RC 振荡器 (FRC)
  - 001 = 带 PLL 的快速 RC 振荡器 (FRC)
  - 010 = 主振荡器 (XT、HS 和 EC)
  - 011 = 带 PLL 的主振荡器 (XT、HS 和 EC)
  - 100 = 辅助振荡器 (SOSC)
  - 101 = 低功耗 RC 振荡器 (LPRC)
  - 110 = 带 16 分频的快速 RC 振荡器 (FRC)
  - 111 = 带 n 分频的快速 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
  - 000 = 快速 RC 振荡器 (FRC)
  - 001 = 带 PLL 的快速 RC 振荡器 (FRC)
  - 010 = 主振荡器 (XT、HS 和 EC)
  - 011 = 带 PLL 的主振荡器 (XT、HS 和 EC)
  - 100 = 辅助振荡器 (SOSC)
  - 101 = 低功耗 RC 振荡器 (LPRC)
  - 110 = 带 16 分频的快速 RC 振荡器 (FRC)
  - 111 = 带 n 分频的快速 RC 振荡器 (FRC)
- bit 7            **CLKLOCK:** 时钟锁定使能位  
 如果时钟切换被使能且 FSCM 被禁止 (FOSC<FCKSM> = 0b01)
  - 1 = 时钟切换被禁止, 系统时钟源被锁定
  - 0 = 时钟切换被使能, 系统时钟源可被时钟切换更改
- bit 6            **IOLOCK:** 外设引脚选择锁定位
  - 1 = 外设引脚选择被锁定, 不允许写入外设引脚选择寄存器
  - 0 = 外设引脚选择未锁定, 允许写入外设引脚选择寄存器
- bit 5            **LOCK:** PLL 锁定状态位 (只读)
  - 1 = 表示 PLL 处于锁定状态, 或 PLL 起振定时器延时结束
  - 0 = 表示 PLL 处于失锁状态, 起振定时器在运行或 PLL 被禁止
- bit 4            **未实现:** 读为 0
- bit 3            **CF:** 时钟故障检测位 (由应用程序读 / 清零)
  - 1 = FSCM 检测到时钟故障
  - 0 = FSCM 未检测到时钟故障
- bit 2            **未实现:** 读为 0

**寄存器 8-1: OSCCON: 振荡器控制寄存器 (续)**

- bit 1      **LPOSCEN:** 辅助 (LP) 振荡器使能位  
1 = 使能辅助振荡器  
0 = 禁止辅助振荡器
- bit 0      **OSWEN:** 振荡器切换使能位  
1 = 请求切换到由 NOSC<2:0> 位指定的振荡器  
0 = 振荡器切换完成

寄存器 8-2: CLKDIV: 时钟分频比寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0
ROI	DOZE<2:0>			DOZEN <sup>(1)</sup>	FRCDIV<2:0>		
bit 15				bit 8			
R/W-0	R/W-1	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PLLPOST<1:0>		—	PLLPRE<4:0>				
bit 7				bit 0			

**图注:** y = 在 POR 时由配置位设置的值  
 R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
 -n = POR 时的值                1 = 置 1                              0 = 清零                              x = 未知

- bit 15            **ROI:** 中断恢复位  
 1 = 中断将清零 DOZEN 位, 并且处理器时钟 / 外设时钟频率比被设置为 1:1  
 0 = 中断对 DOZEN 位无影响
- bit 14-12       **DOZE<2:0>:** 处理器时钟分频比选择位  
 000 = Fcy/1  
 001 = Fcy/2  
 010 = Fcy/4  
 011 = Fcy/8 (默认)  
 100 = Fcy/16  
 101 = Fcy/32  
 110 = Fcy/64  
 111 = Fcy/128
- bit 11           **DOZEN:** 打盹模式使能位 <sup>(1)</sup>  
 1 = DOZE<2:0> 位域指定外设时钟与处理器时钟之间的频率比  
 0 = 处理器时钟 / 外设时钟频率比强制为 1:1
- bit 10-8        **FRCDIV<2:0>:** 内部快速 RC 振荡器后分频比位  
 000 = FRC 1 分频 (默认)  
 001 = FRC 2 分频  
 010 = FRC 4 分频  
 011 = FRC 8 分频  
 100 = FRC 16 分频  
 101 = FRC 32 分频  
 110 = FRC 64 分频  
 111 = FRC 256 分频
- bit 7-6         **PLLPOST<1:0>:** PLL VCO 输出分频比选择位 (也表示为 “N2”, PLL 后分频比)  
 00 = 输出 /2  
 01 = 输出 /4 (默认)  
 10 = 保留  
 11 = 输出 /8
- bit 5            **未实现:** 读为 0
- bit 4-0         **PLLPRE<4:0>:** PLL 相位检测器输入分频比位 (也表示为 “N1”, PLL 预分频比)  
 00000 = 输入 /2 (默认)  
 00001 = 输入 /3  
 .  
 .  
 .  
 11111 = 输入 /33

注 1: 该位在 ROI 位置 1 和产生中断时清零。

**寄存器 8-3: PLLFBD: PLL 反馈倍频比寄存器**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0 <sup>(1)</sup>
—	—	—	—	—	—	—	PLLDIV<8>
bit 15							bit 8

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
PLLDIV<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 15-9        **未实现:** 读为 0  
 bit 8-0        **PLLDIV<8:0>:** PLL 反馈倍频比位 (也表示为 “M”, PLL 倍频比)  
                  000000000 = 2  
                  000000001 = 3  
                  000000010 = 4  
                  .  
                  .  
                  .  
                  000110000 = 50 (默认)  
                  .  
                  .  
                  .  
                  111111111 = 513



寄存器 8-4: OSCTUN: FRC 振荡器调节寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN<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        **TUN<5:0>**: FRC 振荡器调节位  
               011111 = 中心频率 +11.625%  
               011110 = 中心频率 +11.25% (8.23 MHz)  
               •  
               •  
               •  
               000001 = 中心频率 +0.375% (7.40 MHz)  
               000000 = 中心频率 (标称值 7.37 MHz)  
               111111 = 中心频率 -0.375% (7.345 MHz)  
               •  
               •  
               •  
               100001 = 中心频率 -11.625% (6.52 MHz)  
               100000 = 中心频率 -12% (6.49 MHz)

**寄存器 8-5: ACLKCON: 附属时钟分频器控制寄存器**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	SELACLK	AOSCMD<1:0>		APSTSCLR<2:0>			
bit 15								bit 8
R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	
ASRCSEL	—	—	—	—	—	—	—	
bit 7								bit 0

**图注:**

R = 可读位                      W = 可写位                      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        **SELACLK:** 为附属时钟分频器选择附属时钟源  
 1 = 附属振荡器为附属时钟分频器提供时钟源  
 0 = 带 PLL 的 FRC 为附属时钟分频器提供时钟源
- bit 12-11    **AOSCMD<1:0>:** 附属振荡器模式  
 11 = EC 外部时钟模式选择  
 10 = XT 振荡器模式选择  
 01 = HS 振荡器模式选择  
 00 = 禁止附属振荡器 (默认)
- bit 10-8     **APSTSCLR<2:0>:** 附属时钟输出分频值  
 111 = 1 分频  
 110 = 2 分频  
 101 = 4 分频  
 100 = 8 分频  
 111 = 16 分频  
 010 = 32 分频  
 011 = 64 分频  
 000 = 256 分频 (默认)
- bit 7         **ASRCSEL:** 为附属时钟选择参考时钟源  
 1 = 主振荡器是时钟源  
 0 = 附属振荡器是时钟源
- bit 6-0      **未实现:** 读为 0

## 8.2 时钟切换工作原理

在软件控制下，应用可以在任何时候在四个时钟源（主振荡器、LP、FRC 和 LPRC）之间自由切换。为限制这种灵活性可能产生的负面影响，dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件的时钟切换过程带有安全锁定。

**注：** 主振荡器模式有三种不同的子模式（XT、HS 和 EC），这三种子模式由 POSCMD<1:0> 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。

### 8.2.1 使能时钟切换

要使能时钟切换，配置寄存器中的 FCKSM1 配置位必须编程为 0。（更多详细信息，请参见第 27.1 节“配置位”。）如果 FCKSM1 配置位未被编程（1），则时钟切换功能和故障保护时钟监视器功能被禁止。这是默认设置。

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

在时钟切换被禁止时，OSWEN 控制位（OSCCON<0>）无效。它总是保持为 0。

### 8.2.2 振荡器切换过程

执行时钟切换至少需要以下基本过程：

1. 如果需要，读 COSC 位（OSCCON<14:12>）以确定当前的振荡器源。
2. 执行解锁序列以允许写入 OSCCON 寄存器的高字节。
3. 将适当的值写入 NOSC 控制位（OSCCON<10:8>）选择新振荡器源。
4. 执行解锁序列以允许写入 OSCCON 寄存器的低字节。
5. 将 OSWEN 位（OSCCON<0>）置 1 以启动振荡器切换。

一旦基本过程完成，系统时钟硬件将自动进行如下响应：

1. 时钟切换硬件将 NOSC 控制位的新值与 COSC 状态位进行比较。如果它们相同，则时钟切换是多余的操作。在这种情况下，OSWEN 位自动清零，时钟切换中止。
2. 如果启动了有效的时钟切换，则 LOCK（OSCCON<5>）和 CF（OSCCON<3>）状态位清零。
3. 如果新振荡器现在不在运行，则硬件会将它开启。如果开启的是晶振，则硬件将等待直到振荡器起振定时器（OST）超时。如果新的振荡器源使用 PLL，则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 位表示时钟切换成功。此外，NOSC 位的值被传送到 COSC 状态位。
6. 此时旧时钟源被关闭，LPRC（如果 WDT 或 FSCM 被使能）或 LP（如果 LPOSCEN 保持置 1）除外。

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

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

## 8.3 故障保护时钟监视器（FSCM）

故障保护时钟监视器（FSCM）允许器件在即使振荡器发生故障的情况下仍能继续运行。通过编程使能 FSCM 功能。如果使能了 FSCM 功能，LPRC 内部振荡器将总是运行（休眠模式下除外），并且不受看门狗定时器的控制。

在发生振荡器故障时，FSCM 会产生时钟故障陷阱事件，并将系统时钟切换到 FRC 振荡器。然后应用程序可尝试重新启动振荡器或执行受控关闭。只需将复位地址装入振荡器故障陷阱向量，就可将陷阱作为一个热复位来处理。

如果使用 PLL 倍频器来对系统时钟倍频，则时钟发生故障时内部 FRC 也会被以相同的倍频比倍频。实际上时钟发生故障时器件会切换到带 PLL 的 FRC。

注:

## 9.0 节能特性

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“*dsPIC33F Family Reference Manual*”的“Section 9. Watchdog Timer and Power Savings Modes”（DS70196），该文档可从 Microchip 网站（www.microchip.com）下载。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件提供了管理功耗的功能，该功能是通过有选择地管理 CPU 和外设的时钟来实现的。一般来说，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。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件通过以下四种方法管理功耗：

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

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

### 9.1 时钟频率和时钟切换

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件提供的时钟频率范围较宽，用户可根据应用需要进行选择。如果未锁定系统时钟配置，用户只需更改 NOSC 位（OSCCON<10:8>）即可选择低功耗或高精度振荡器。在工作期间更改系统时钟的过程以及相应的限制，将在第 8.0 节“振荡器配置”中进行更详细的讨论。

## 9.2 基于指令的节能模式

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件有两种特殊的节能模式，通过执行特殊的 PWRSAV 指令可以进入这两种模式。休眠模式下时钟停止工作并暂停所有代码执行；空闲模式下 CPU 停止工作并停止代码执行，但是允许外设模块继续工作。例 9-1 中所示为 PWRSAV 指令的汇编语法。

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

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

### 9.2.1 休眠模式

休眠模式具有下列特征：

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

当发生以下任何事件时，器件将从休眠模式唤醒：

- 产生任何被单独允许的中断
- 任何形式的器件复位
- WDT 超时

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

#### 例 9-1: PWRSAV 指令语法

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

## 9.2.2 空闲模式

空闲模式具有下列特征：

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

当发生以下任何事件时，器件将从空闲模式唤醒：

- 产生任何被单独允许的中断
- 任何器件复位
- WDT 超时

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

## 9.2.3 在节能指令执行期间的中断

在执行 PWRSAV 指令时产生的中断都将延迟到进入休眠或空闲模式后才起作用，并导致器件从休眠或空闲模式中唤醒。

## 9.3 打盹模式

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

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

通过将 DOZEN 位 (CLKDIV<11>) 置 1 使能打盹模式。外设与内核的时钟速度之比是由 DOZE<2:0> 位 (CLKDIV<14:12>) 决定的。有八种可能的配置，从 1:1 到 1:128，其中 1:1 是默认设置。

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

例如，假设器件的工作速度为 20 MIPS，并已根据这一速度将 ECAN 模块的速度配置为 500 kbps。如果现在将器件置于时钟频率比为 1:4 的打盹模式下，那么 ECAN 模块将继续按要求的 500 kbps 比特率通信，而 CPU 则以 5 MIPS 的速度开始执行指令。

## 9.4 外设模块禁止

外设模块禁止 (Peripheral Module Disable, PMD) 寄存器通过停止所有提供给外设模块的时钟源提供一种禁止外设模块的方法。当通过相应的 PMD 控制位禁止外设时，外设就进入了功耗最低的状态。与外设相关的控制寄存器和状态寄存器也被禁止，因此写入这些寄存器不起作用，且读取值无效。

只有在 PMD 寄存器中的相应位被清零且特定的 dsPIC<sup>®</sup> DSC 器件支持某个外设时，才能使能相应的外设模块。如果外设存在于器件中，则默认情况下，它是通过 PMD 寄存器使能的。

<b>注：</b>	如果 PMD 位置 1，则对应的模块将在一个指令周期的延时后被禁止。类似地，如果某个 PMD 位清零，则对应的模块将在一个指令周期的延时后被使能（假设已将模块控制寄存器配置为使能模块的工作）。
-----------	--

## 10.0 I/O 端口

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 10. I/O Ports”（DS70193），该文档可从 Microchip 网站（www.microchip.com）下载。

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

### 10.1 并行 I/O（PIO）端口

与某个外设共用一个引脚的并行 I/O 端口通常服从于该外设。外设的输出缓冲区数据和控制信号提供给一对多路开关。这对多路开关用于选择 I/O 引脚的输出数据和控制信号是来自外设还是相应的端口。该逻辑电路同时会阻止“环回进入”（loop through），即一个端口的数

字输出可以驱动共用同一个引脚的外设的输入。图 10-1 显示了端口是如何与其他外设复用的以及相关的 I/O 引脚。

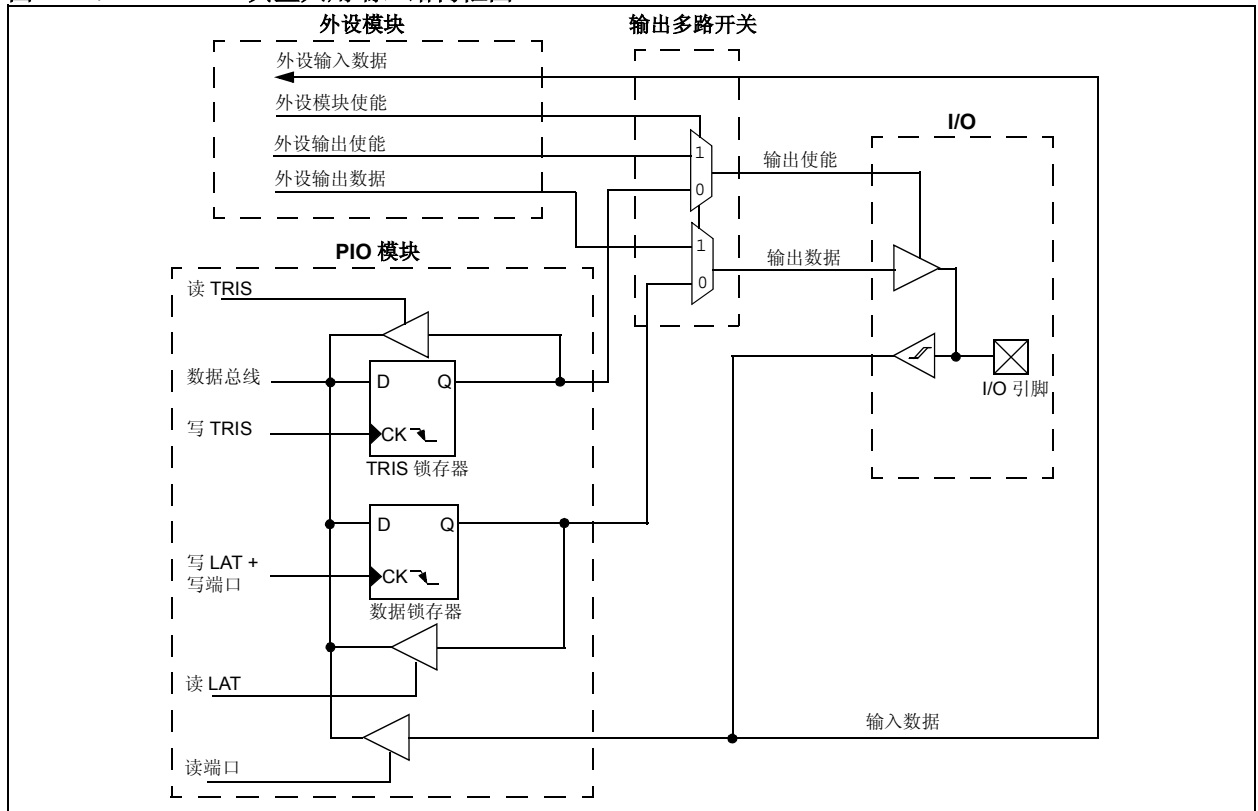
如果外设使能，并且外设正在驱动相关引脚时，将禁止将该引脚用作通用输出引脚。可以读该 I/O 引脚，但并行端口位的输出驱动器将被禁止。如果使能某外设但该外设没有驱动相应的引脚时，则该引脚可由一个端口驱动。

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

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

当端口引脚与另一个外设共用或与定义为仅输入的功能共用时，它将被视为专用端口，因为没有任何其他竞争的输出源。

图 10-1: 典型共用端口结构框图



10.1.1 漏极开路配置

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

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

10.2 配置模拟端口引脚

AD1PCFGL 和 TRIS 寄存器用于控制模数（Analog-to-Digital, A/D）端口引脚的操作。如果要使端口引脚用作模拟输入，则对应的 TRIS 位必须置 1（输入）。如果将 TRIS 位清零（输出），则数字输出电平（VOH 或 VOL）将被转换。

当读取端口寄存器时，所有配置为模拟输入通道的引脚均读为零（低电平）。

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

10.2.1 I/O 端口写 / 读时序

端口方向改变或端口写操作与同一端口的读操作之间需要一个指令周期。这通常通过执行一条 NOP 指令来实现，如例 10-1 所示。

例 10-1: 端口写 / 读示例

```
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
```

10.3 输入电平变化通知

I/O 端口的输入电平变化通知功能允许 dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件在选定输入引脚的状态变化时，向处理器发出中断请求。即使在休眠模式下、禁止时钟时，该特性也可检测输入状态改变。根据器件的引脚数，最多可以选择（允许）21 个外部信号（CNx 引脚）在输入状态发生变化时产生中断请求。

有 4 个与 CN 模块相关的控制寄存器。CNEN1 和 CNEN2 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电路。弱上拉电路充当连接到该引脚的电流源，当连接了按钮或键盘设备时，不再需要使用外部电阻。使用包含每个 CN 引脚控制位的 CNPU1 和 CNPU2 寄存器可分别使能各个上拉电路。将任一控制位置 1 均可使能相应引脚的弱上拉功能。

**注：** 当端口引脚被配置为数字输出时，电平变化通知引脚的上拉应始终被禁止。



## 10.4 外设引脚选择

外设引脚选择配置使得可以在较宽的 I/O 引脚范围内选择和配置外设功能部件。通过增加特定器件上可用的引脚配置选项，程序员可以让单片机更好地适合他们的整个应用，而不是通过修改应用来适应器件。

外设引脚选择配置功能在数字 I/O 引脚的固定子集下工作。程序员可以将大多数数字外设的输入和 I 或输出独立地映射到这些 I/O 引脚中的任何一个。用软件来进行外设引脚选择，通常不需要对器件进行再编程。一旦建立外设引脚选择，就同时包含了硬件保护，以防止对外设映射的意外或错误更改。

### 10.4.1 可用的引脚

外设引脚选择功能可在最多 26 个引脚的范围内使用。可用引脚的数目取决于具体器件及其引脚数。支持外设引脚选择功能的引脚在它们的引脚全称中包含名称“RPn”，其中“RP”表示可重映射的外设，“n”是可重映射的引脚编号。

### 10.4.2 可用的外设

由外设引脚选择功能管理的外设都是仅用作数字功能的外设。这些外设包括：

- 一般串行通信（UART、SPI 和 ECAN）
- 通用定时器时钟输入
- 与定时器相关的外设（输入捕捉和输出比较）
- 比较器模块
- 电平变化中断输入

相比较而言，某些仅用作数字功能的外设模块不能使用外设引脚选择功能。这是因为外设功能需要特定端口上的特殊 I/O 电路，且不能容易地连接到多个引脚。这样的模块有 I<sup>2</sup>C。类似的要求排除了所有带模拟输入的模块，例如模数转换器（ADC）。

可重映射的外设与默认的 I/O 引脚无关。必须在使用外设前始终将其分配给特定的 I/O 引脚。相反，不可重映射的外设始终在默认引脚上可用，假设该外设在工作且与其他外设没有冲突。

### 10.4.2.1 外设引脚选择功能优先级

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

### 10.4.3 控制外设引脚选择

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

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

#### 10.4.3.1 输入映射

外设引脚选择选项的输入在外设基础上进行映射。与外设相关的控制寄存器指示要映射到的引脚。RPINRx 寄存器用来配置外设输入映射（见寄存器 10-1 到寄存器 10-20）。每个寄存器包含 5 位位域组，每组都与可重映射外设之一相关。用适当的 5 位值编程给定外设的位域，会将具有该值的 RPn 引脚映射到该外设。对于任何给定的器件，任何位域的值的有效范围与器件所支持的外设引脚选择的数目相对应。

图 10-2 说明了 U1RX 输入的可重映射引脚选择。

图 10-2: U1RX 的可重映射多路开关 (MUX) 输入

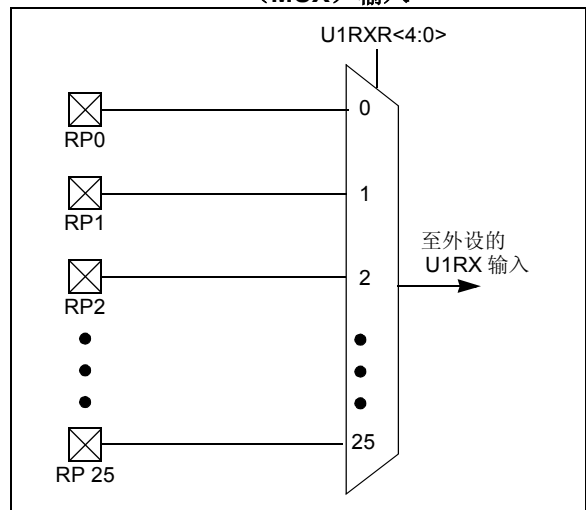


表 10-1: 可选的输入源 (将输入映射到功能) (1)

输入名称	功能名称	寄存器	配置位
外部中断 1	INT1	RPINR0	INT1R<4:0>
外部中断 2	INT2	RPINR1	INT2R<4:0>
Timer2 外部时钟	T2CK	RPINR3	T2CKR<4:0>
Timer3 外部时钟	T3CK	RPINR3	T3CKR<4:0>
Timer4 外部时钟	T4CK	RPINR4	T4CKR<4:0>
Timer5 外部时钟	T5CK	RPINR4	T5CKR<4:0>
输入捕捉 1	IC1	RPINR7	IC1R<4:0>
输入捕捉 2	IC2	RPINR7	IC2R<4:0>
输入捕捉 7	IC7	RPINR10	IC7R<4:0>
输入捕捉 8	IC8	RPINR10	IC8R<4:0>
输出比较故障 A	OCFA	RPINR11	OCFAR<4:0>
PWM1 故障	FLTA1	RPINR12	FLTA1R<4:0>
PWM2 故障	FLTA2	RPINR13	FLTA2R<4:0>
QE1 的 A 相	QEA1	RPINR14	QEA1R<4:0>
QE1 的 B 相	QEB1	RPINR14	QEB1R<4:0>
QE1 索引	INDX1	RPINR15	INDX1R<4:0>
QE2 的 A 相	QEA2	RPINR16	QEA2R<4:0>
QE2 的 B 相	QEB2	RPINR16	QEB2R<4:0>
QE2 索引	INDX2	RPINR17	INDX2R<4:0>
UART1 接收	U1RX	RPINR18	U1RXR<4:0>
UART1 允许发送	U1CTS	RPINR18	U1CTSR<4:0>
UART2 接收	U2RX	RPINR19	U2RXR<4:0>
UART2 允许发送	U2CTS	RPINR19	U2CTSR<4:0>
SPI1 数据输入	SDI1	RPINR20	SDI1R<4:0>
SPI1 时钟输入	SCK1	RPINR20	SCK1R<4:0>
SPI1 从选择输入	SS1	RPINR21	SS1R<4:0>
SPI2 数据输入	SDI2	RPINR22	SDI2R<4:0>
SPI2 时钟输入	SCK2	RPINR22	SCK2R<4:0>
SPI2 从选择输入	SS2	RPINR23	SS2R<4:0>
ECAN1 接收	CIRX	RPINR26	CIRXR<4:0>

注 1: 除非另外声明, 否则所有输入都使用施密特输入缓冲器。

10.4.3.2 输出映射

与输入相比，外设引脚选择选项的输出在引脚基础上进行映射。在这种情况下，与特定引脚相关的控制寄存器指示要被映射的外设输出。RPORx 寄存器用来控制输出映射。像 RPINRx 寄存器一样，每个寄存器包含 5 位位域组，每组都与一个 RPn 引脚相关（见寄存器 10-21 到寄存器 10-33）。位域的值与外设之一相对应，并且该外设的输出被映射到引脚（见表 10-2 和图 10-3）。

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

图 10-3: RPn 的可重映射输出的多路选择功能

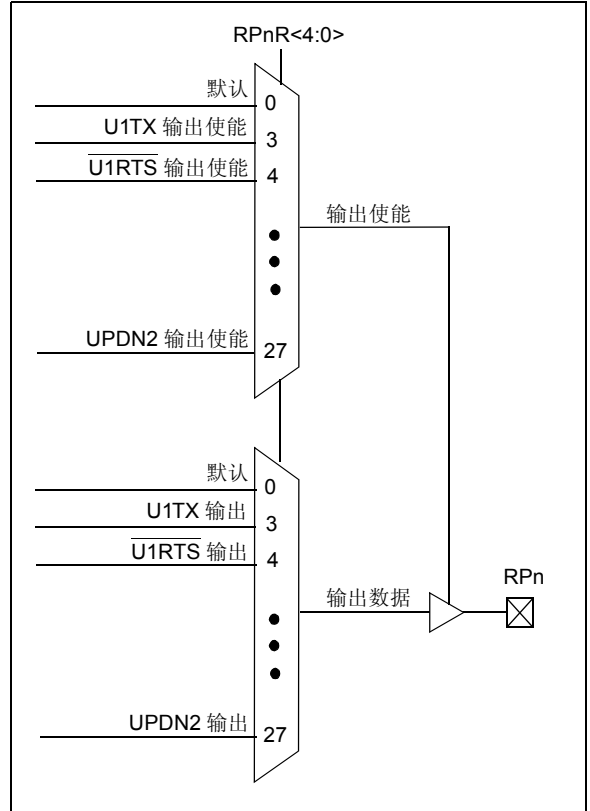


表 10-2: 可重映射引脚 (RPn) 的输出选择

功能	RPnR<4:0>	输出名称
NULL	00000	RPn 连接到默认端口引脚
C1OUT	00001	RPn 连接到比较器 1 输出
C2OUT	00010	RPn 连接到比较器 2 输出
U1TX	00011	RPn 连接到 UART1 发送
$\overline{U1RTS}$	00100	RPn 连接到 UART1 请求发送
U2TX	00101	RPn 连接到 UART2 发送
$\overline{U2RTS}$	00110	RPn 连接到 UART2 请求发送
SDO1	00111	RPn 连接到 SPI1 数据输出
SCK1OUT	01000	RPn 连接到 SPI1 时钟输出
SS1OUT	01001	RPn 连接到 SPI1 从选择输出
SDO2	01010	RPn 连接到 SPI2 数据输出
SCK2OUT	01011	RPn 连接到 SPI2 时钟输出
SS2OUT	01100	RPn 连接到 SPI2 从选择输出
C1TX	10000	RPn 连接到 ECAN1 发送
OC1	10010	RPn 连接到输出比较 1
OC2	10011	RPn 连接到输出比较 2
OC3	10100	RPn 连接到输出比较 3
OC4	10101	RPn 连接到输出比较 4
UPDN1	11010	RPn 连接到 QE11 方向 (UPDN) 状态
UPDN2	11011	RPn 连接到 QE12 方向 (UPDN) 状态

### 10.4.3.3 外设映射

外设选择引脚的控制机制不局限于固定外设配置的小范围内。在任何外设映射 SFR 之间没有相互的或硬件强制的锁定。也就是说，任何或所有 RPn 引脚上的外设映射的任何组合都是可能的。这包括外设输入和输出对引脚的多对一或一对多映射。

从配置观点来看，这类映射在技术上是可行的，但在电气方面可能不被支持。

### 10.4.4 控制配置改变

由于可以在运行时更改外设的重映射，因此需要对外设重映射进行一些限制以防意外更改配置。dsPIC33F 器件具有 3 个功能以防对外设映射的更改：

- 控制寄存器锁定序列
- 连续状态监视
- 配置位引脚选择锁定

#### 10.4.4.1 控制寄存器锁定

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

要置 1 或清零 IOLOCK，必须执行特定的命令序列：

1. 将 0x46 写入 OSCCON<7:0>。
2. 将 0x57 写入 OSCCON<7:0>。
3. 执行将 IOLOCK 清零（或置 1）的单个操作。

**注：** MPLAB® C30 提供了用于解锁 OSCCON 寄存器的内建 C 语言函数：

```
__builtin_write_OSCCONL(value)
__builtin_write_OSCCONH(value)
```

请参见 MPLAB IDE 帮助以获取更多信息。

与振荡器的 LOCK 位的类似序列不同，IOLOCK 会保持一种状态直到被更改。这允许所有的外设引脚选择均可被配置为：在单个解锁序列后紧跟对所有控制寄存器的更新，然后用第二个锁定序列进行锁定。

### 10.4.4.2 连续状态监视

除了防止直接写操作，RPINRx 和 RPORx 寄存器的内容一直由硬件通过影子寄存器监视。如果任何寄存器发生了意外改变（例如 ESD 或其他外部事件引起的干扰），将会触发配置不匹配复位。

#### 10.4.4.3 配置位引脚选择锁定

为了进一步确保安全，可以将器件配置为防止对 RPINRx 和 RPORx 寄存器进行多于一个写会话。IOL1WAY (FOSC<IOL1WAY>) 配置位会阻止 IOLOCK 位在被置 1 一次后被清零。如果 IOLOCK 保持置 1，则不会执行寄存器解锁过程，且不能写入外设引脚选择控制寄存器。清零该位并重新使能外设映射的唯一方法是执行器件复位。

在默认（未编程）状态下，IOL1WAY 被置 1，将用户限制为只能进行一个写会话。对 IOL1WAY 编程可允许用户应用程序（通过正确使用解锁序列）对外设引脚选择寄存器不受限制的访问。

### 10.4.5 外设引脚选择的注意事项

在应用设计中使用控制外设引脚选择功能有一些值得注意的事项，包括几个只能作为可重映射外设的常见外设。

#### 10.4.5.1 初始化和锁定

主要的注意事项是在器件的默认（复位）状态下，外设引脚选择在默认引脚上不可用。尤其是，由于所有 RPINRx 和 RPORx 寄存器被复位为 0000h，这意味着所有的外设引脚选择输入都连接到 RP0，而所有的外设引脚选择输出处于未连接状态。这意味着在执行任何其他应用代码前，用户应用程序必须用适当的外设配置初始化器件。

由于 IOLOCK 位在解锁状态下复位，因此在器件复位结束后不必执行解锁序列。然而，基于应用安全考虑，在写入控制寄存器后将 IOLOCK 置 1 并锁定配置是一个好的建议。

解锁序列对时序要求比较严格，因此必须以与更改振荡器配置相同的方式，通过编写汇编语言程序来执行。如果应用程序是用 C 语言或其他高级语言编写的，则解锁序列应通过写行内汇编来执行。

#### 10.4.5.2 选择配置

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

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

#### 10.4.5.3 引脚操作

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

#### 10.4.5.4 模拟功能

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

#### 10.4.5.5 配置示例

例 10-2 给出了为使用 UART1 的带流控制双向通信所作的配置。使用了以下输入和输出功能：

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

#### 例 10-2: 配置 UART1 输入和输出功能

```

//*****
// Unlock Registers
//*****
asm volatile ( "mov #OSCCONL, w1  \n"
               "mov #0x46, w2    \n"
               "mov #0x57, w3    \n"
               "mov.b w2, [w1]   \n"
               "mov.b w3, [w1]   \n"
               "bclr OSCCON, 6");

//*****
// Configure Input Functions
// (See Table 10-1)
//*****
//*****
// Assign U1Rx To Pin RP0
//*****
RPINR18bits.U1RXR = 0;

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

//*****
// Configure Output Functions
// (See Table 10-2)
//*****
//*****
// Assign U1Tx To Pin RP2
//*****
RPOR1bits.RP2R = 3;

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

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

```



寄存器 10-2: RPINR1: 外设引脚选择输入寄存器 1

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-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	INT2R<4:0>				
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 15-5      **未实现:** 读为 0  
 bit 4-0      **INTR2R<4:0>:** 将外部中断 2 (INTR2) 分配给对应 RPn 引脚的位  
               11111 = 输入连接到 Vss  
               11001 = 输入连接到 RP25  
               •  
               •  
               •  
               00001 = 输入连接到 RP1  
               00000 = 输入连接到 RP0

**寄存器 10-3: RPINR3: 外设引脚选择输入寄存器 3**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T3CKR<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T2CKR<4:0>				
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 12-8      **T3CKR<4:0>:** 将 Timer3 外部时钟 (T3CK) 分配给对应 RPn 引脚的位
  - 11111 = 输入连接到 Vss
  - 11001 = 输入连接到 RP25
  - 
  - 
  - 
  - 00001 = 输入连接到 RP1
  - 00000 = 输入连接到 RP0
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **T2CKR<4:0>:** 将 Timer2 外部时钟 (T2CK) 分配给对应 RPn 引脚的位
  - 11111 = 输入连接到 Vss
  - 11001 = 输入连接到 RP25
  - 
  - 
  - 
  - 00001 = 输入连接到 RP1
  - 00000 = 输入连接到 RP0



寄存器 10-4: **RPINR4: 外设引脚选择输入寄存器 4**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T5CKR<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T4CKR<4:0>				
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 12-8      **T5CKR<4:0>:** 将 Timer5 外部时钟 (T5CK) 分配给对应 RPn 引脚的位
  - 11111 = 输入连接到 Vss
  - 11001 = 输入连接到 RP25
  - 
  - 
  - 
  - 00001 = 输入连接到 RP1
  - 00000 = 输入连接到 RP0
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **T4CKR<4:0>:** 将 Timer4 外部时钟 (T4CK) 分配给对应 RPn 引脚的位
  - 11111 = 输入连接到 Vss
  - 11001 = 输入连接到 RP25
  - 
  - 
  - 
  - 00001 = 输入连接到 RP1
  - 00000 = 输入连接到 RP0

**寄存器 10-5: RPINR7: 外设引脚选择输入寄存器 7**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC2R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC1R<4:0>				
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 12-8      **IC2R<4:0>:** 将输入捕捉 2 (IC2) 分配给对应 RPn 引脚的位

- 11111 = 输入连接到 Vss
- 11001 = 输入连接到 RP25
- 
- 
- 

- 00001 = 输入连接到 RP1
- 00000 = 输入连接到 RP0

bit 7-5      **未实现:** 读为 0

bit 4-0      **IC1R<4:0>:** 将输入捕捉 1 (IC1) 分配给对应 RPn 引脚的位

- 11111 = 输入连接到 Vss
- 11001 = 输入连接到 RP25
- 
- 
- 

- 00001 = 输入连接到 RP1
- 00000 = 输入连接到 RP0

寄存器 10-6: **RPINR10: 外设引脚选择输入寄存器 10**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC8R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC7R<4:0>				
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 12-8      **IC8R<4:0>:** 将输入捕捉 8 (IC8) 分配给对应 RPn 引脚的位

- 11111 = 输入连接到 Vss
- 11001 = 输入连接到 RP25
- 
- 
- 

- 00001 = 输入连接到 RP1
- 00000 = 输入连接到 RP0

bit 7-5      **未实现:** 读为 0

bit 4-0      **IC7R<4:0>:** 将输入捕捉 7 (IC7) 分配给对应 RPn 引脚的位

- 11111 = 输入连接到 Vss
- 11001 = 输入连接到 RP25
- 
- 
- 

- 00001 = 输入连接到 RP1
- 00000 = 输入连接到 RP0

**寄存器 10-7: RPINR11: 外设引脚选择输入寄存器 11**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	OCFAR<4:0>				
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 15-5        **未实现:** 读为 0  
 bit 4-0        **OCFAR<4:0>:** 将输出比较故障 A (OCFA) 分配给对应 RPn 引脚的位  
                  11111 = 输入连接到 Vss  
                  11001 = 输入连接到 RP25  
                  ·  
                  ·  
                  ·  
                  00001 = 输入连接到 RP1  
                  00000 = 输入连接到 RP0

**寄存器 10-8: RPINR12: 外设引脚选择输入寄存器 12**

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-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	FLTA1R<4:0>				
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 15-5        **未实现:** 读为 0  
 bit 4-0        **FLTA1R<4:0>:** 将 PWM1 故障 (FLTA1) 分配给对应 RPn 引脚的位  
                  11111 = 输入连接到 Vss  
                  11001 = 输入连接到 RP25  
                  ·  
                  ·  
                  ·  
                  00001 = 输入连接到 RP1  
                  00000 = 输入连接到 RP0

寄存器 10-9: **RPINR13: 外设引脚选择输入寄存器 13**

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-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	FLTA2R<4:0>				—
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 15-5      **未实现:** 读为 0
- bit 4-0      **FLTA2R<4:0>:** 将 PWM2 故障 (FLTA2) 分配给对应 RPn 引脚的位
  - 11111 = 输入连接到 Vss
  - 11001 = 输入连接到 RP25
  - 
  - 
  - 
  - 00001 = 输入连接到 RP1
  - 00000 = 输入连接到 RP0

**寄存器 10-10: RPINR14: 外设引脚选择输入寄存器 14**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	QEB1R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	QEA1R<4:0>				
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 12-8      **QEB1R<4:0>:** 将 QE1 的 B 相输入 (QEB1) 分配给对应引脚的位  
 11111 = 输入连接到 Vss  
 11001 = 输入连接到 RP25  
 •  
 •  
 •  
 00001 = 输入连接到 RP1  
 00000 = 输入连接到 RP0

bit 7-5      **未实现:** 读为 0

bit 4-0      **QEA1R<4:0>:** 将 QE1 的 A 相输入 (QEA1) 分配给对应引脚的位  
 11111 = 输入连接到 Vss  
 11001 = 输入连接到 RP25  
 •  
 •  
 •  
 00001 = 输入连接到 RP1  
 00000 = 输入连接到 RP0

寄存器 10-11: **RPINR15: 外设引脚选择输入寄存器 15**

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-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	INDX1R<4:0>				
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 15-5 **未实现:** 读为 0

bit 4-0 **INDX1R<4:0>:** 将 QE11 索引 (INDX1) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11001 = 输入连接到 RP25

•

•

•

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0

**寄存器 10-12: RPINR16: 外设引脚选择输入寄存器 16**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	QEB2R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	QEA2R<4:0>				
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 12-8      **QEB2R<4:0>:** 将 QEI2 的 B 相输入 (QEB2) 分配给对应引脚的位  
 11111 = 输入连接到 Vss  
 11001 = 输入连接到 RP25  
 •  
 •  
 •  
 00001 = 输入连接到 RP1  
 00000 = 输入连接到 RP0

bit 7-5      **未实现:** 读为 0

bit 4-0      **QEA2R<4:0>:** 将 QEI2 的 A 相输入 (QEA2) 分配给对应引脚的位  
 11111 = 输入连接到 Vss  
 11001 = 输入连接到 RP25  
 •  
 •  
 •  
 00001 = 输入连接到 RP1  
 00000 = 输入连接到 RP0



寄存器 10-13: RPINR17: 外设引脚选择输入寄存器 17

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-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	INDX2R<4:0>				—
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 15-5      未实现: 读为 0  
 bit 4-0      **INDX2R<4:0>**: 将 QEI2 索引 (INDX2) 分配给对应 RPn 引脚的位  
               11111 = 输入连接到 Vss  
               11001 = 输入连接到 RP25  
               •  
               •  
               •  
               00001 = 输入连接到 RP1  
               00000 = 输入连接到 RP0

**寄存器 10-14: RPINR18: 外设引脚选择输入寄存器 18**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U1CTSR<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U1RXR<4:0>				
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 12-8      **U1CTSR<4:0>:** 将 UART1 允许发送 (U1CTS) 分配给对应 RPn 引脚的位

- 11111 = 输入连接到 Vss
- 11001 = 输入连接到 RP25
- 
- 
- 

- 00001 = 输入连接到 RP1
- 00000 = 输入连接到 RP0

bit 7-5      **未实现:** 读为 0

bit 4-0      **U1RXR<4:0>:** 将 UART1 接收 (U1RX) 分配给对应 RPn 引脚的位

- 11111 = 输入连接到 Vss
- 11001 = 输入连接到 RP25
- 
- 
- 

- 00001 = 输入连接到 RP1
- 00000 = 输入连接到 RP0

寄存器 10-15: **RPINR19: 外设引脚选择输入寄存器 19**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U2CTSR<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U2RXR<4:0>				
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 12-8     **U2CTSR<4:0>:** 将 UART2 允许发送 (U2CTS) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss  
 11001 = 输入连接到 RP25

•  
•  
•

00001 = 输入连接到 RP1  
 00000 = 输入连接到 RP0

bit 7-5     **未实现:** 读为 0

bit 4-0     **U2RXR<4:0>:** 将 UART2 接收 (U2RX) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss  
 11001 = 输入连接到 RP25

•  
•  
•

00001 = 输入连接到 RP1  
 00000 = 输入连接到 RP0

**寄存器 10-16: RPINR20: 外设引脚选择输入寄存器 20**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SCK1R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SDI1R<4:0>				
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 12-8      **SCK1R<4:0>:** 将 SPI1 时钟输入 (SCK1) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss  
 11001 = 输入连接到 RP25

•  
•  
•

00001 = 输入连接到 RP1  
 00000 = 输入连接到 RP0

bit 7-5      **未实现:** 读为 0

bit 4-0      **SDI1R<4:0>:** 将 SPI1 数据输入 (SDI1) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss  
 11001 = 输入连接到 RP25

•  
•  
•

00001 = 输入连接到 RP1  
 00000 = 输入连接到 RP0

寄存器 10-17: **RPINR21: 外设引脚选择输入寄存器 21**

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-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SS1R<4:0>				
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 15-5      **未实现:** 读为 0  
 bit 4-0      **SS1R<4:0>:** 将 SPI1 从选择输入 (SS1) 分配给对应 RPn 引脚的位  
               11111 = 输入连接到 Vss  
               11001 = 输入连接到 RP25  
               •  
               •  
               •  
               00001 = 输入连接到 RP1  
               00000 = 输入连接到 RP0

**寄存器 10-18: RPINR22: 外设引脚选择输入寄存器 22**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SCK2R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SDI2R<4:0>				
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 12-8      **SCK2R<4:0>:** 将 SPI2 时钟输入 (SCK2) 分配给对应 RPn 引脚的位  
 11111 = 输入连接到 Vss  
 11001 = 输入连接到 RP25  
 .  
 .  
 .  
 00001 = 输入连接到 RP1  
 00000 = 输入连接到 RP0

bit 7-5      **未实现:** 读为 0

bit 4-0      **SDI2R<4:0>:** 将 SPI2 数据输入 (SDI2) 分配给对应 RPn 引脚的位  
 11111 = 输入连接到 Vss  
 11001 = 输入连接到 RP25  
 .  
 .  
 .  
 00001 = 输入连接到 RP1  
 00000 = 输入连接到 RP0

**寄存器 10-19: RPINR23: 外设引脚选择输入寄存器 23**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SS2R<4:0>				
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 15-5        **未实现:** 读为 0  
 bit 4-0        **SS2R<4:0>:** 将 SPI2 从选择输入 (SS2) 分配给对应 RPn 引脚的位  
                  11111 = 输入连接到 Vss  
                  11001 = 输入连接到 RP25  
                  ·  
                  ·  
                  ·  
                  00001 = 输入连接到 RP1  
                  00000 = 输入连接到 RP0

**寄存器 10-20: RPINR26: 外设引脚选择输入寄存器 26<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	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	C1RXR<4:0>				
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 15-5        **未实现:** 读为 0  
 bit 4-0        **U1RXR<4:0>:** 将 ECAN1 接收 (C1RX) 分配给对应 RPn 引脚的位  
                  11111 = 输入连接到 Vss  
                  11001 = 输入连接到 RP25  
                  ·  
                  ·  
                  ·  
                  00001 = 输入连接到 RP1  
                  00000 = 输入连接到 RP0

**注 1:** 对于不带 ECAN™ 模块的器件禁止此寄存器。

**寄存器 10-21: RPOR0: 外设引脚选择输出寄存器 0**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP1R<4:0>				
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
—	—	—	RP0R<4:0>				
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 12-8      **RP1R<4:0>:** 将外设输出功能分配给 RP1 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP0R<4:0>:** 将外设输出功能分配给 RP0 输出引脚的位 (请参见表 10-2 了解外设功能编号)

**寄存器 10-22: RPOR1: 外设引脚选择输出寄存器 1**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP3R<4:0>				
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
—	—	—	RP2R<4:0>				
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 12-8      **RP3R<4:0>:** 将外设输出功能分配给 RP3 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP2R<4:0>:** 将外设输出功能分配给 RP2 输出引脚的位 (请参见表 10-2 了解外设功能编号)



**寄存器 10-23: RPOR2: 外设引脚选择输出寄存器 2**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP5R<4:0>				
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
—	—	—	RP4R<4:0>				
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 12-8      **RP5R<4:0>:** 将外设输出功能分配给 RP5 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP4R<4:0>:** 将外设输出功能分配给 RP4 输出引脚的位 (请参见表 10-2 了解外设功能编号)

**寄存器 10-24: RPOR3: 外设引脚选择输出寄存器 3**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP7R<4:0>				
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
—	—	—	RP6R<4:0>				
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 12-8      **RP7R<4:0>:** 将外设输出功能分配给 RP7 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP6R<4:0>:** 将外设输出功能分配给 RP6 输出引脚的位 (请参见表 10-2 了解外设功能编号)

**寄存器 10-25: RPOR4: 外设引脚选择输出寄存器 4**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP9R<4:0>				
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
—	—	—	RP8R<4:0>				
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 12-8      **RP9R<4:0>:** 将外设输出功能分配给 RP9 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP8R<4:0>:** 将外设输出功能分配给 RP8 输出引脚的位 (请参见表 10-2 了解外设功能编号)

**寄存器 10-26: RPOR5: 外设引脚选择输出寄存器 5**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP11R<4:0>				
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
—	—	—	RP10R<4:0>				
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 12-8      **RP11R<4:0>:** 将外设输出功能分配给 RP11 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP10R<4:0>:** 将外设输出功能分配给 RP10 输出引脚的位 (请参见表 10-2 了解外设功能编号)

**寄存器 10-27: RPOR6: 外设引脚选择输出寄存器 6**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP13R<4:0>				
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
—	—	—	RP12R<4:0>				
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 12-8      **RP13R<4:0>:** 将外设输出功能分配给 RP13 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP12R<4:0>:** 将外设输出功能分配给 RP12 输出引脚的位 (请参见表 10-2 了解外设功能编号)

**寄存器 10-28: RPOR7: 外设引脚选择输出寄存器 7**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP15R<4:0>				
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
—	—	—	RP14R<4:0>				
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 12-8      **RP15R<4:0>:** 将外设输出功能分配给 RP15 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP14R<4:0>:** 将外设输出功能分配给 RP14 输出引脚的位 (请参见表 10-2 了解外设功能编号)

**寄存器 10-29: RPOR8: 外设引脚选择输出寄存器 8<sup>(1)</sup>**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP17R<4:0>				
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
—	—	—	RP16R<4:0>				
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 12-8      **RP17R<4:0>:** 将外设输出功能分配给 RP17 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP16R<4:0>:** 将外设输出功能分配给 RP16 输出引脚的位 (请参见表 10-2 了解外设功能编号)

**注 1:** 仅 44 引脚器件上实现了此寄存器。

**寄存器 10-30: RPOR9: 外设引脚选择输出寄存器 9<sup>(1)</sup>**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP19R<4:0>				
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
—	—	—	RP18R<4:0>				
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 12-8      **RP19R<4:0>:** 将外设输出功能分配给 RP19 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP18R<4:0>:** 将外设输出功能分配给 RP18 输出引脚的位 (请参见表 10-2 了解外设功能编号)

**注 1:** 仅 44 引脚器件上实现了此寄存器。

寄存器 10-31: **RPOR10: 外设引脚选择输出寄存器 10<sup>(1)</sup>**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP21R<4:0>				
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
—	—	—	RP20R<4:0>				
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 12-8      **RP21R<4:0>:** 将外设输出功能分配给 RP21 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP20R<4:0>:** 将外设输出功能分配给 RP20 输出引脚的位 (请参见表 10-2 了解外设功能编号)

注 1: 仅 44 引脚器件上实现了此寄存器。

寄存器 10-32: **RPOR11: 外设引脚选择输出寄存器 11<sup>(1)</sup>**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP23R<4:0>				
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
—	—	—	RP22R<4:0>				
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 12-8      **RP23R<4:0>:** 将外设输出功能分配给 RP23 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP22R<4:0>:** 将外设输出功能分配给 RP22 输出引脚的位 (请参见表 10-2 了解外设功能编号)

注 1: 仅 44 引脚器件上实现了此寄存器。

**寄存器 10-33: RPOR12: 外设引脚选择输出寄存器 12<sup>(1)</sup>**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP25R<4:0>				
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
—	—	—	RP24R<4:0>				
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 12-8      **RP25R<4:0>:** 将外设输出功能分配给 RP25 输出引脚的位 (请参见表 10-2 了解外设功能编号)
- bit 7-5        **未实现:** 读为 0
- bit 4-0        **RP24R<4:0>:** 将外设输出功能分配给 RP24 输出引脚的位 (请参见表 10-2 了解外设功能编号)

**注 1:** 仅 44 引脚器件上实现了此寄存器。

### 11.0 TIMER2/3 和 TIMER4/5 特性

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 11. Timers”（DS70205），该文档可从 Microchip 网站（www.microchip.com）下载。

- 外部时钟输入（TxCK）始终与器件内部时钟同步，时钟同步在预分频之后进行

图 11-1 给出了 B 类定时器的框图。

Timer3 和 Timer5 是 C 类定时器，具有以下特性：

- C 类定时器可以与 B 类定时器组合成一个 32 位定时器
- 至少有一个 C 类定时器能触发 A/D 转换
- 外部时钟输入（TxCK）始终与器件内部时钟同步，时钟同步在预分频之前进行

图 11-2 给出了 C 类定时器的框图。

Timer2 和 Timer4 是 B 类定时器，具有以下特性：

- B 类定时器可以与 C 类定时器组合成一个 32 位定时器

图 11-1: B 类定时器框图 (x = 2 或 4)

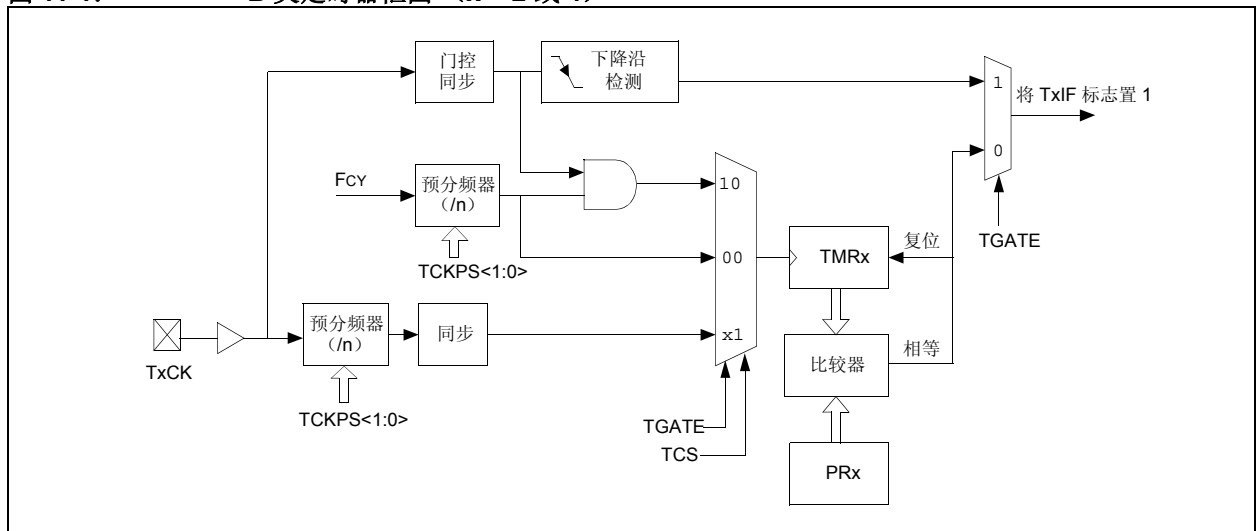
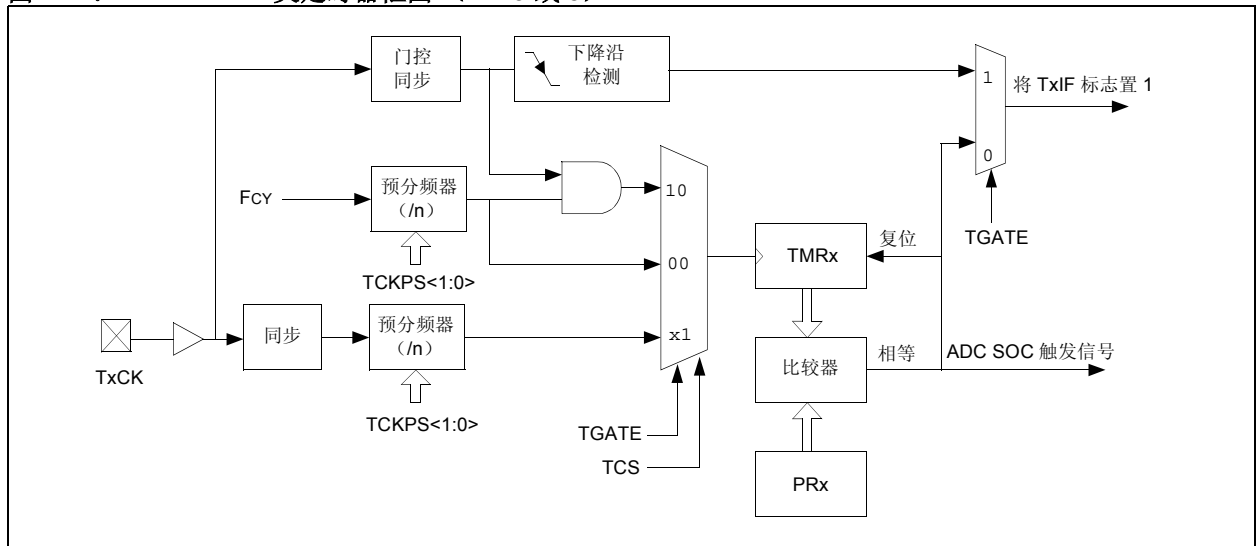


图 11-2: C 类定时器框图 (x = 3 或 5)



Timer2/3 和 Timer4/5 模块都可以工作于以下模式之一：

- 定时器模式
- 门控定时器模式
- 同步计数器模式

在定时器和门控定时器模式下，输入时钟来自于内部指令周期时钟（Fcy）。在同步计数器模式下，输入时钟来自于 TxCK 引脚上的外部时钟输入。

定时器模式由以下位决定：

- TCS (TxCON<1>)：定时器时钟源控制位
- TGATE (TxCON<6>)：定时器门控位

表 11-1 给出了不同工作模式下的定时器控制位的设置。

表 11-1: 定时器模式设置

模式	TCS	TGATE
定时器	0	0
门控定时器	0	1
同步计数器	1	x

### 11.1 16 位工作

要将任一定时器配置为独立的 16 位工作：

1. 清零与该定时器对应的 T32 位。
2. 使用 TCKPS<1:0> 位选择定时器预分频比。
3. 使用 TCS 和 TGATE 位设置时钟和门控模式。
4. 将定时器的周期值装入 PRx 寄存器。
5. 如果需要中断，将中断允许位 TxIE 置 1。使用优先级位 TxIP<2:0> 来设置中断优先级。
6. 将 TON 位置 1。

**注：** 仅Timer2和Timer3能触发DMA数据传输。

### 11.2 32 位工作

可以将 B 类和 C 类 16 位定时器模块组合成 32 位定时器模块。对于 32 位定时器工作，B 类定时器控制寄存器中的 T32 控制位 (TxCON<3>) 必须置 1。对于 32 位工作，C 类定时器保存高位字 (msw)，而 B 类定时器保存低位字 (lsw)。

当配置为 32 位工作时，只需要 B 类定时器控制 (TxCON) 寄存器位来进行设置和控制。C 类定时器控制寄存器位被忽略 (TSIDL 位除外)。

组合的 32 位定时器使用 C 类定时器的中断允许、中断标志和中断优先级控制位进行中断控制。32 位定时器工作时 B 类定时器的中断控制和状态位被忽略。

表 11-2 列出了可以组合成 32 位定时器的 B 类和 C 类定时器。

表 11-2: 32 位定时器

B 类定时器 (lsw)	C 类定时器 (msw)
Timer2	Timer3
Timer4	Timer5

图 11-3 给出了 32 位定时器模块的框图。32 位定时器模块可以工作于以下模式之一：

- 定时器模式
- 门控定时器模式
- 同步计数器模式

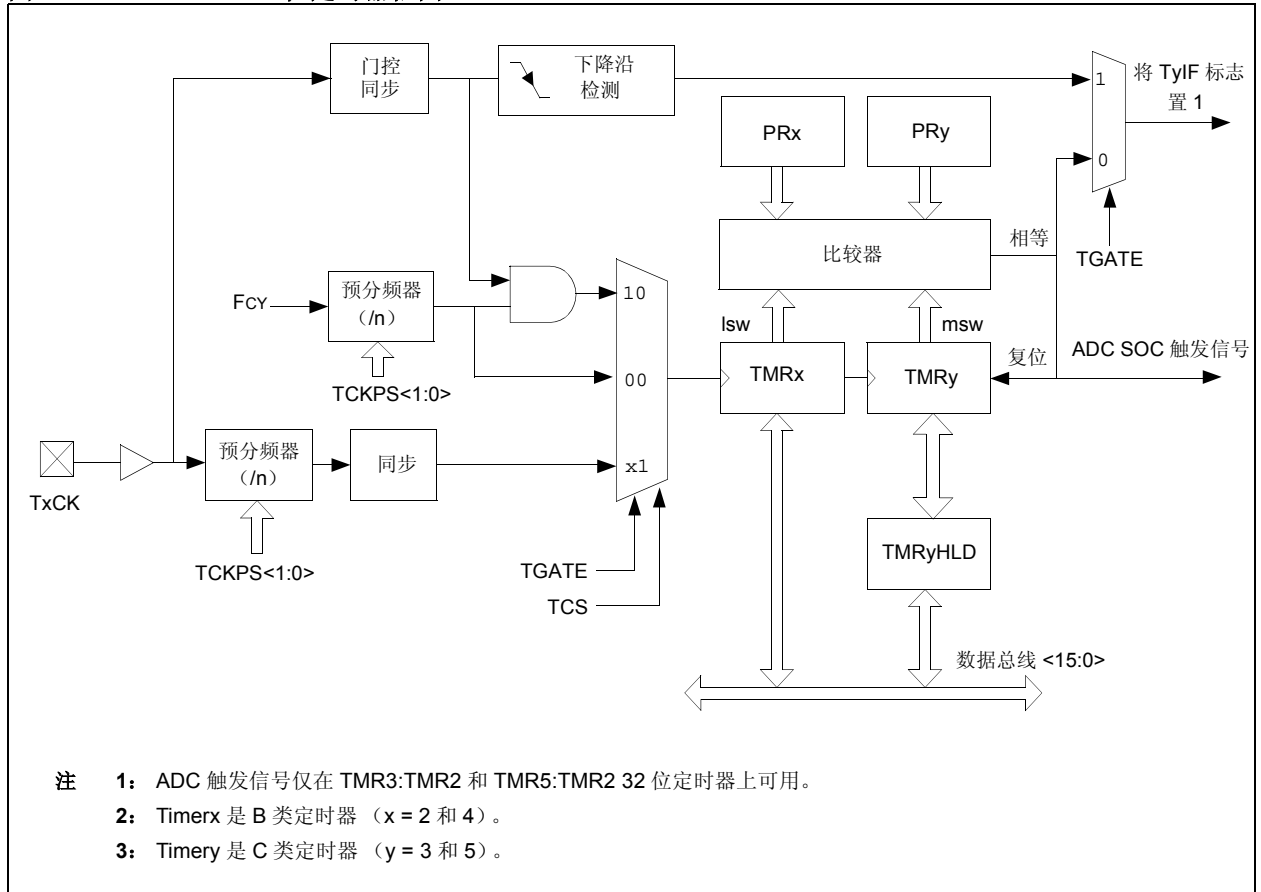
要将 Timer2/3 或 Timer4/5 配置为 32 位工作：

1. 将 T32 控制位置 1。
2. 使用 TCKPS<1:0> 位为 Timer2 或 Timer4 选择预分频比。
3. 使用相应的 TCS 和 TGATE 位设置时钟和门控模式。
4. 装入定时器的周期值。PR3 或 PR5 包含值的高位字，而 PR2 或 PR4 包含低位字。
5. 如果需要中断，将中断允许位 T3IE 或 T5IE 置 1。使用优先级位 T3IP<2:0> 或 T5IP<2:0> 设置中断优先级。Timer2 或 Timer4 控制定时器，而中断由 Timer3 或 Timer5 产生。
6. 将相应的 TON 位置 1。

任意时刻定时器的值被存储在寄存器对 TMR3:TMR2 或 TMR5:TMR4 中，TMR3 或 TMR5 总是包含计数值的高位字，而 TMR2 或 TMR4 包含低位字。



图 11-3: 32 位定时器框图



**寄存器 11-1: TxCON: 定时器控制寄存器 (x = 2 或 4)**

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8
U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		T32 <sup>(1)</sup>	—	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:** Timerx 使能位  
             当 T32 = 1 时 (在 32 位定时器模式下):  
             1 = 启动 32 位 TMRx:TMRy 定时器对  
             0 = 停止 32 位 TMRx:TMRy 定时器对  
             当 T32 = 0 时 (在 16 位定时器模式下):  
             1 = 启动 16 位定时器  
             0 = 停止 16 位定时器
- bit 14        **未实现:** 读为 0
- bit 13        **TSIDL:** 空闲模式停止位  
             1 = 当器件进入空闲模式时, 定时器停止工作  
             0 = 在空闲模式下定时器继续工作
- bit 12-7     **未实现:** 读为 0
- bit 6        **TGATE:** Timerx 门控时间累加使能位  
             当 TCS = 1 时:  
             该位被忽略。  
             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 预分频比  
             10 = 1:64 预分频比  
             01 = 1:8 预分频比  
             00 = 1:1 预分频比
- bit 3        **T32:** 32 位 Timerx 模式选择位 <sup>(1)</sup>  
             1 = Timerx 和 Timery 组成一个 32 位定时器  
             0 = Timerx 和 Timery 分别作为独立的 16 位定时器
- bit 2        **未实现:** 读为 0
- bit 1        **TCS:** Timerx 时钟源选择位  
             1 = 来自 TxCK 引脚的外部时钟  
             0 = 内部时钟 (Fosc/2)
- bit 0        **未实现:** 读为 0

**注 1:** 在 32 位模式下, TyCON 中的控制位不影响 32 位定时器的的工作。

寄存器 11-2: TyCON: 定时器控制寄存器 (y = 3 或 5)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON <sup>(2)</sup>	—	TSIDL <sup>(1)</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>	TCKPS<1:0> <sup>(2)</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:** 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:** 空闲模式停止位 <sup>(1)</sup>  
 1 = 当器件进入空闲模式时, 定时器停止工作  
 0 = 在空闲模式下定时器继续工作
- bit 12-7        **未实现:** 读为 0
- bit 6            **TGATE:** Timery 门控时间累加使能位 <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 预分频比  
 10 = 1:64 预分频比  
 01 = 1:8 预分频比  
 00 = 1:1 预分频比
- bit 3-2         **未实现:** 读为 0
- bit 1            **TCS:** Timery 时钟源选择位 <sup>(2)</sup>  
 1 = 来自 TyCK 引脚的外部时钟  
 0 = 内部时钟 (Fosc/2)
- bit 0            **未实现:** 读为 0

- 注 1: 当在定时器控制寄存器 (TxCON<3>) 中使能 32 位定时器工作时 (T32 = 1), TSIDL 位必须清零以在空闲模式下运行 32 位定时器。
- 2: 当在定时器控制寄存器 (TxCON<3>) 中使能 32 位定时器工作时 (T32 = 1), 这些位将不起作用。

注:

## 12.0 TIMER1

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 11. Timers”（DS70205），该文档可从 Microchip 网站（www.microchip.com）下载。

Timer1 模块是一个 16 位定时器，可作为实时时钟的时间计数器，或作为自由运行的间隔定时器 / 计数器。

Timer1 模块具有不同于其他定时器的以下独特特性：

- 可依靠器件上的低功耗 32 kHz 晶振工作。
- 可依靠外部时钟源以异步计数器模式工作。
- 外部时钟输入（T1CK）可以选择与内部器件时钟同步，时钟同步在预分频之后执行。

Timer1 的独特特性使其可以用于实时时钟（Real Time Clock, RTC）应用。图 12-1 给出了 Timer1 的框图。

Timer1 模块可工作于以下模式之一：

- 定时器模式
- 门控定时器模式
- 同步计数器模式
- 异步计数器模式

在定时器和门控定时器模式下，输入时钟来自于内部指令周期时钟（FcY）。在同步和异步计数器模式下，输入时钟来自于 T1CK 引脚上的外部时钟输入。

定时器模式由以下位决定：

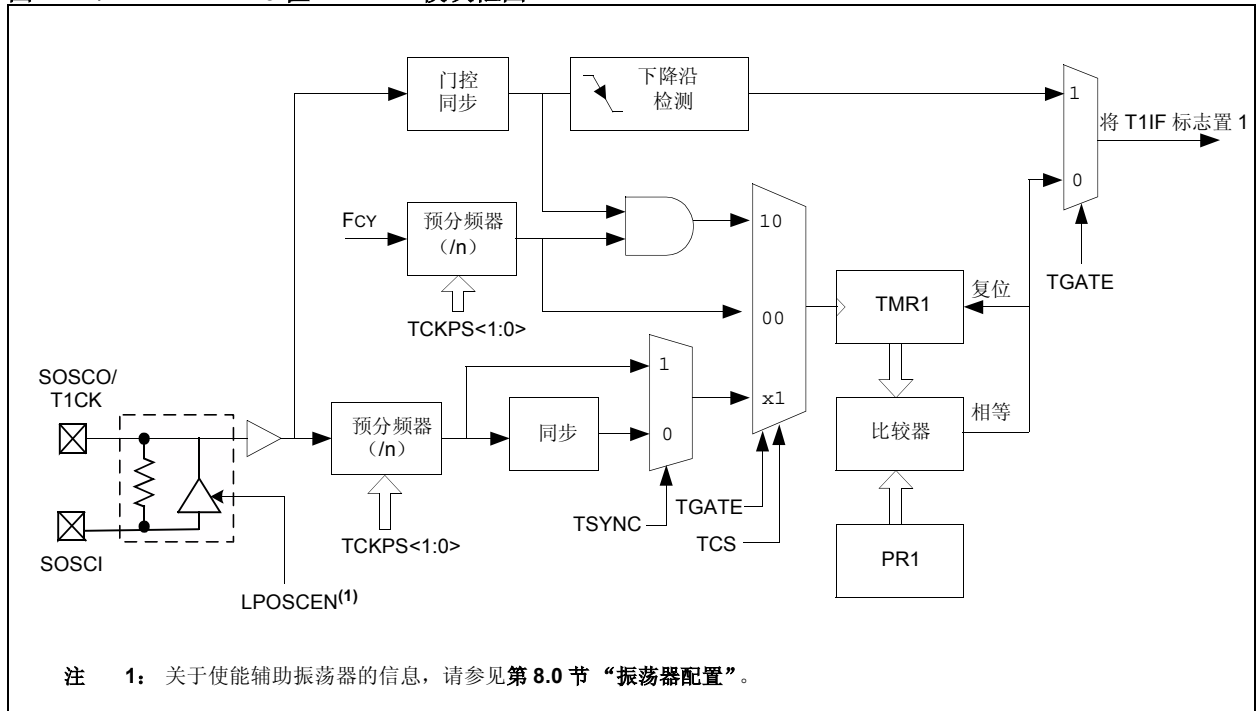
- 定时器时钟源控制位（TCS）：T1CON<1>
- 定时器同步控制位（TSYNC）：T1CON<2>
- 定时器门控位（TGATE）：T1CON<6>

表 12-1 给出了不同工作模式下的定时器控制位的设置。

**表 12-1: 定时器模式设置**

模式	TCS	TGATE	TSYNC
定时器	0	0	x
门控定时器	0	1	x
同步计数器	1	x	1
异步计数器	1	x	0

**图 12-1: 16 位 TIMER1 模块框图**



**寄存器 12-1: T1CON: TIMER1 控制寄存器**

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8
U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		—	TSYNC	TCS	—
bit 7							bit 0

**图注:**

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

- bit 15        **TON:** Timer1 使能位  
               1 = 启动 16 位 Timer1  
               0 = 停止 16 位 Timer1
- bit 14        **未实现:** 读为 0
- bit 13        **TSIDL:** 空闲模式停止位  
               1 = 当器件进入空闲模式时, 模块停止工作  
               0 = 在空闲模式下模块继续工作
- bit 12-7     **未实现:** 读为 0
- bit 6        **TGATE:** Timer1 门控时间累加使能位  
               当 T1CS = 1 时:  
               该位被忽略。  
               当 T1CS = 0 时:  
               1 = 使能门控时间累加  
               0 = 禁止门控时间累加
- bit 5-4     **TCKPS<1:0>:** Timer1 输入时钟预分频比选择位  
               11 = 1:256  
               10 = 1:64  
               01 = 1:8  
               00 = 1:1
- bit 3        **未实现:** 读为 0
- bit 2        **TSYNC:** Timer1 外部时钟输入同步选择位  
               当 TCS = 1 时:  
               1 = 与外部时钟输入同步  
               0 = 与外部时钟输入不同步  
               当 TCS = 0 时:  
               该位被忽略。
- bit 1        **TCS:** Timer1 时钟源选择位  
               1 = 来自 T1CK 引脚的外部时钟 (上升沿触发计数)  
               0 = 内部时钟 (Fcy)
- bit 0        **未实现:** 读为 0

### 13.0 输入捕捉

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 12. Input Capture”（DS70198），该文档可从 Microchip 网站（www.microchip.com）下载。

输入捕捉模块在需要频率（周期）和脉冲测量的应用中很有用。dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件支持最多 4 路输入捕捉通道。

当 ICx 引脚上有事件发生时，输入捕捉模块捕捉所选择时基寄存器的 16 位值。导致发生捕捉的事件分为以下三类：

1. 简单捕捉事件模式：
  - 每当 ICx 引脚上的输入信号出现下降沿时捕捉定时器值
  - 每当 ICx 引脚上的输入信号出现上升沿时捕捉定时器值

2. 在每个边沿（上升沿和下降沿）都捕捉定时器值
3. 预分频捕捉事件模式：
  - ICx 引脚上的输入信号每出现 4 个上升沿捕捉一次定时器值
  - ICx 引脚上的输入信号每出现 16 个上升沿捕捉一次定时器值

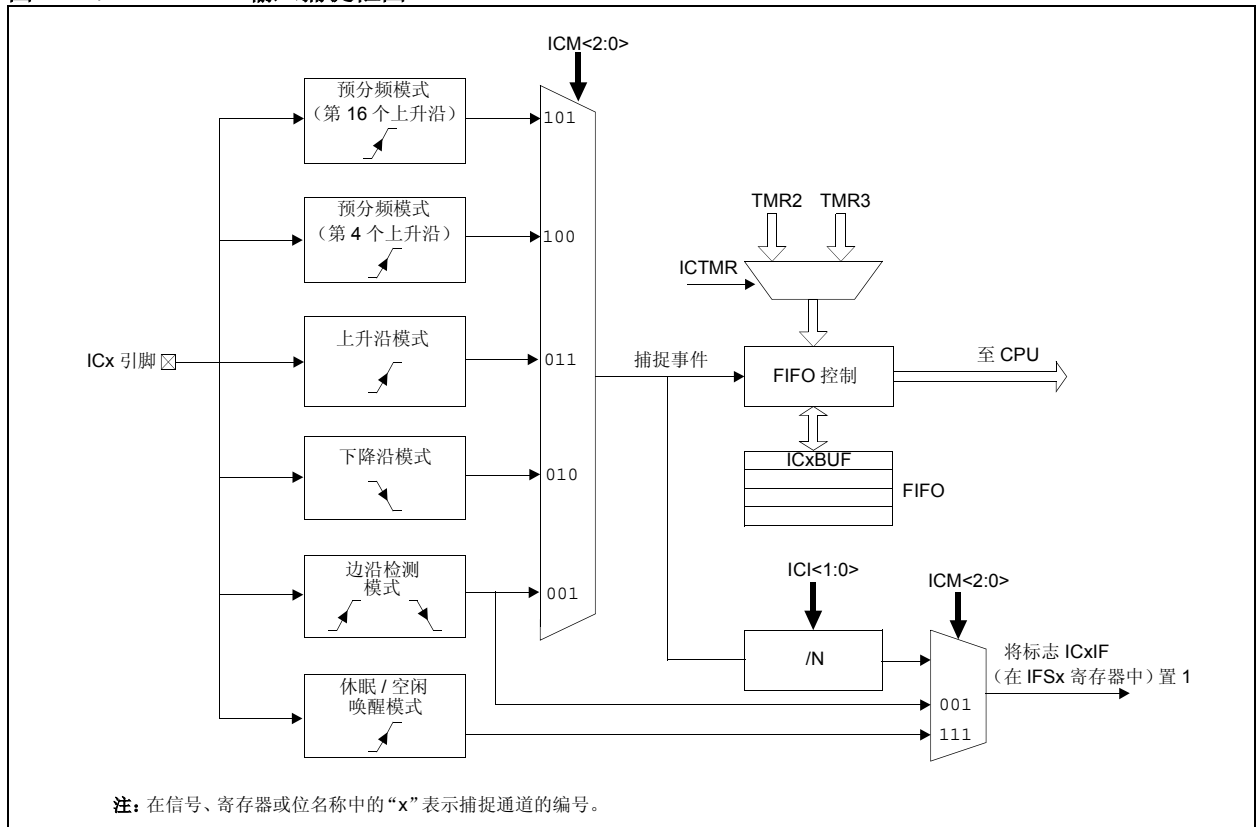
每路输入捕捉通道都可以选择两个 16 位定时器（Timer2 或 Timer3）之一作为时基。所选择定时器可以使用内部或外部时钟。

其他工作特性包括：

- 当 CPU 在休眠和空闲模式时通过捕捉引脚上的信号将器件唤醒
- 输入捕捉事件中断
- 用于存储捕捉值的 4 字 FIFO 缓冲区
  - 可选择在填充完 1、2、3 或 4 个缓冲单元后产生中断
- 输入捕捉也可用来提供额外的外部中断源

**注：** 仅 IC1 和 IC2 能触发 DMA 数据传输。如果需要 DMA 数据传输，FIFO 缓冲区大小必须设置为 1（ICI<1:0> = 00）。

图 13-1: 输入捕捉框图







## 14.0 输出比较

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 13. Output Compare”（DS70209），该文档可从 Microchip 网站（www.microchip.com）下载。

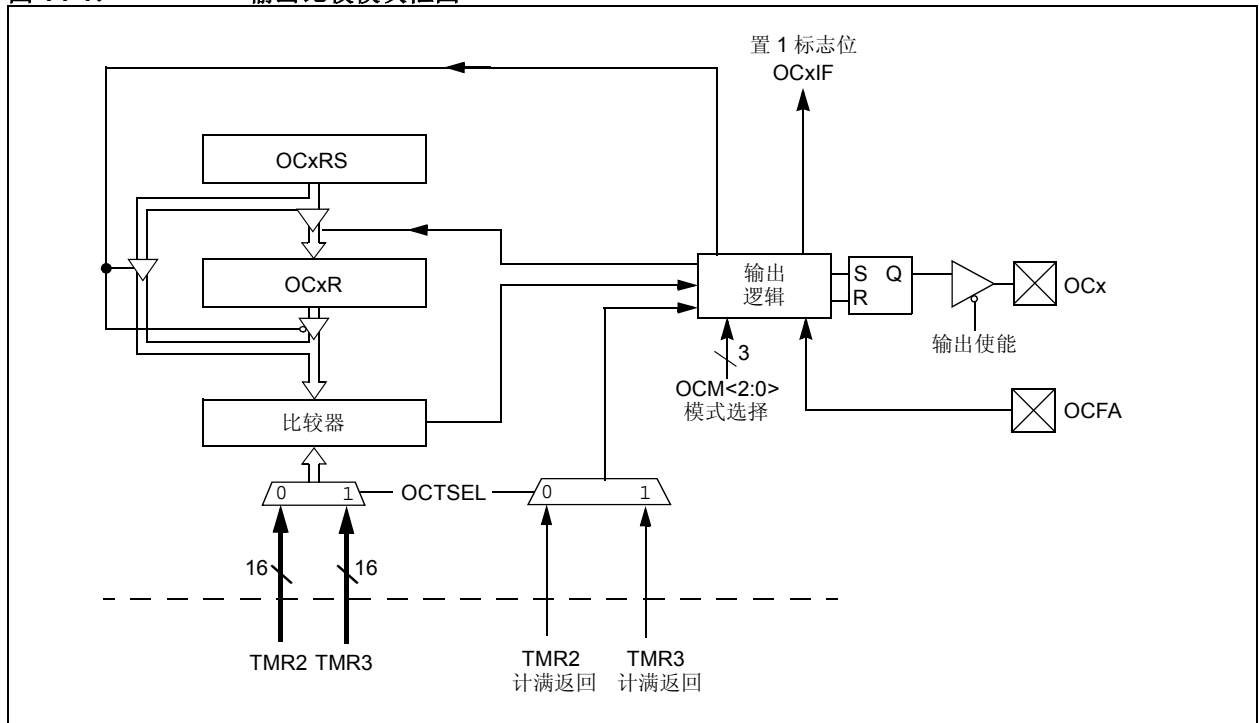
值匹配时，输出引脚的状态发生改变。输出比较模块通过在发生比较匹配事件时改变输出引脚的状态，产生单个输出脉冲或连续输出脉冲。输出比较模块还能在发生比较匹配事件时产生中断。

输出比较模块具有多种工作模式：

- 低电平有效单事件模式
- 高电平有效单事件模式
- 翻转模式
- 延迟单事件模式
- 连续脉冲模式
- 不带故障保护的 PWM 模式
- 带故障保护的 PWM 模式

输出比较模块可以选择 Timer2 或 Timer3 作为其时基。模块将定时器的值与一个或两个比较寄存器的值（取决于所选的工作模式）作比较。当定时器值与比较寄存器

图 14-1: 输出比较模块框图



### 14.1 输出比较模式

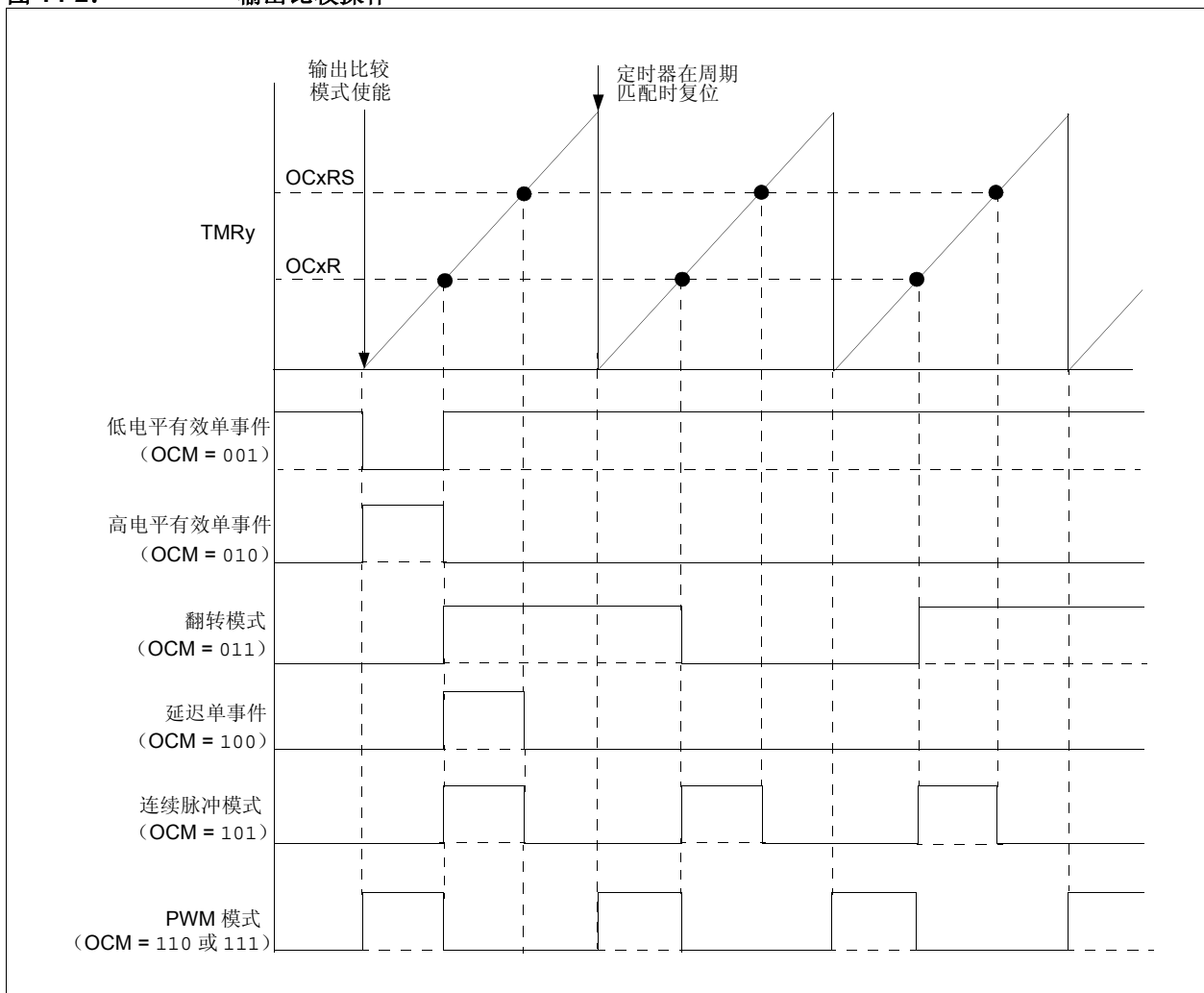
通过设置输出比较控制寄存器中的相应输出比较模式位 OCM<2:0> (OCxCON<2:0>), 配置输出比较模式。表 14-1 列出了输出比较模式的不同位设置。图 14-2 说

明了各种模式的输出比较操作。当写入输出比较控制寄存器时, 用户应用程序必须禁止相关的定时器以避免发生故障。

表 14-1: 输出比较模式

OCM<2:0>	模式	OCx 引脚初始状态	OCx 中断产生
000	禁止模块	由 GPIO 寄存器控制	—
001	低电平有效单事件	0	OCX 上升沿
010	高电平有效单事件	1	OCX 下降沿
011	翻转模式	保持当前输出	OCx 上升沿和下降沿
100	延迟单事件	0	OCX 下降沿
101	连续脉冲模式	0	OCX 下降沿
110	不带故障保护的 PWM 模式	0, 如果 OCxR 为零 1, 如果 OCxR 非零	不产生中断
111	带故障保护的 PWM 模式	0, 如果 OCxR 为零 1, 如果 OCxR 非零	OC1 至 OC4 的 OCFA 下降沿

图 14-2: 输出比较操作



寄存器 14-1: OCxCON: 输出比较 x 控制寄存器 (x = 1、2、3 或 4)

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	—	OCSIDL	—	—	—	—	—
bit 15						bit 8	
U-0	U-0	U-0	R-0 HC	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OCFLT	OCTSEL	OCM<2:0>		
bit 7						bit 0	

<b>图注:</b>	HC = 由硬件清零	HS = 由硬件置 1
R = 可读位	W = 可写位	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:** 在空闲模式下停止输出比较控制位  
 1 = 输出比较 x 在 CPU 空闲模式下停止工作  
 0 = 输出比较 x 在 CPU 空闲模式下继续工作

bit 12-5      **未实现:** 读为 0

bit 4      **OCFLT:** PWM 故障条件状态位  
 1 = 产生了 PWM 故障条件 (仅可由硬件清零)  
 0 = 未产生 PWM 故障条件  
 (仅当 OCM<2:0> = 111 时, 才使用该位。)

bit 3      **OCTSEL:** 输出比较定时器选择位  
 1 = Timer3 是比较 x 的时钟源  
 0 = Timer2 是比较 x 的时钟源

bit 2-0      **OCM<2:0>:** 输出比较模式选择位  
 111 = OCx 处于 PWM 模式, 使能故障引脚  
 110 = OCx 处于 PWM 模式, 禁止故障引脚  
 101 = 将 OCx 引脚初始化为低电平, 在 OCx 引脚上产生连续输出脉冲  
 100 = 将 OCx 引脚初始化为低电平, 在 OCx 引脚上产生单个输出脉冲  
 011 = 比较事件使 OCx 引脚的电平翻转  
 010 = 将 OCx 引脚初始化为高电平, 比较事件强制 OCx 引脚为低电平  
 001 = 将 OCx 引脚初始化为低电平, 比较事件强制 OCx 引脚为高电平  
 000 = 禁止输出比较通道

注:

## 15.0 电机控制 PWM 模块

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“*dsPIC33F Family Reference Manual*”的“Section 14. Motor Control PWM”（DS70187），该文档可从 Microchip 网站（www.microchip.com）下载。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件支持最多两个专用的脉宽调制（Pulse Width Modulation, PWM）模块。PWM1 模块是一个 6 通道 PWM 发生器，而 PWM2 模块是一个 2 通道 PWM 发生器。

PWM 模块具有以下特性：

- 最高 16 位分辨率
- 可在运行过程中更改 PWM 频率
- 边沿和中心对齐输出模式
- 单脉冲生成模式
- 在中心对齐模式下，为不对称更新提供了中断支持
- 为电子换相电机（Electrically Commutative Motor, ECM）操作或无刷直流（Brushless DC, BLDC）电机提供了输出改写控制
- 用于调度其他外设事件的“特殊事件”比较器
- 故障引脚可选择将每个 PWM 输出引脚驱动为定义的状态
- 占空比更新可配置为立即更新或与 PWM 时基同步

## 15.1 PWM1: 6 通道 PWM 模块

此模块简化了产生多个同步 PWM 输出的任务。PWM 模块还能支持以下电源和电机控制应用：

- 三相交流感应电机
- 开关磁阻（Switched Reluctance, SR）电机
- 无刷直流（BLDC）电机
- 不间断电源（Uninterruptible Power Supply, UPS）

此模块包含 3 个占空比发生器，编号从 1 到 3。此模块具有 6 个 PWM 输出引脚，编号从 PWM1H1/PWM1L1 到 PWM1H3/PWM1L3。6 个 I/O 引脚可组合为 3 个高/低端引脚对，分别以后缀 H 或 L 表示。对于互补的负载，低端 PWM 引脚的状态总是与对应高端 I/O 引脚的状态相反。

## 15.2 PWM2: 2 通道 PWM 模块

此模块提供了一对额外的互补 PWM 输出，可用于：

- 电机系统中独立的 PFC 校正
- 感应烹饪

此模块包含 1 个占空比发生器，提供两个 PWM 输出，编号为 PWM2H1/PWM2L1。

图 15-1: 6 通道 PWM 模块框图 (PWM1)

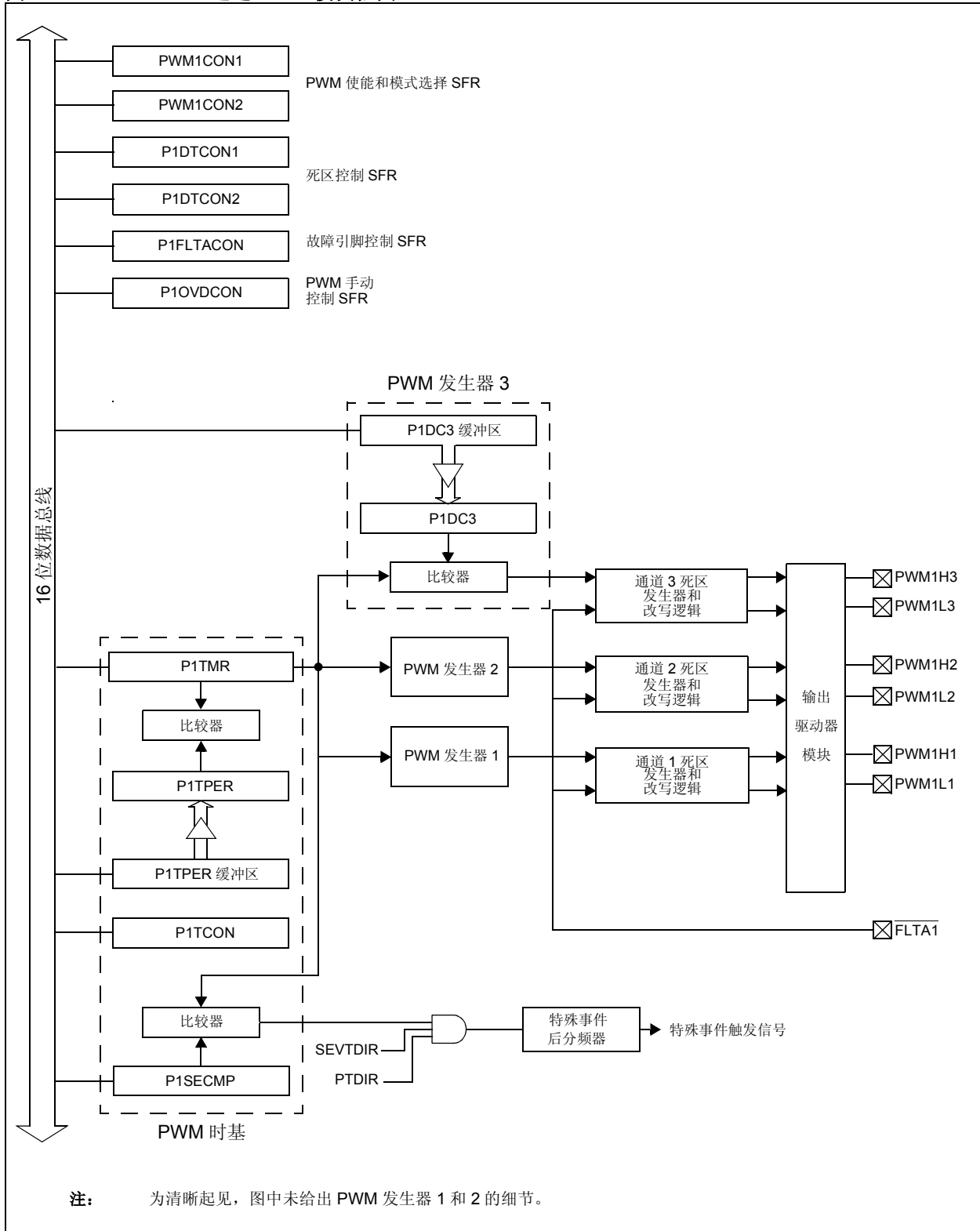
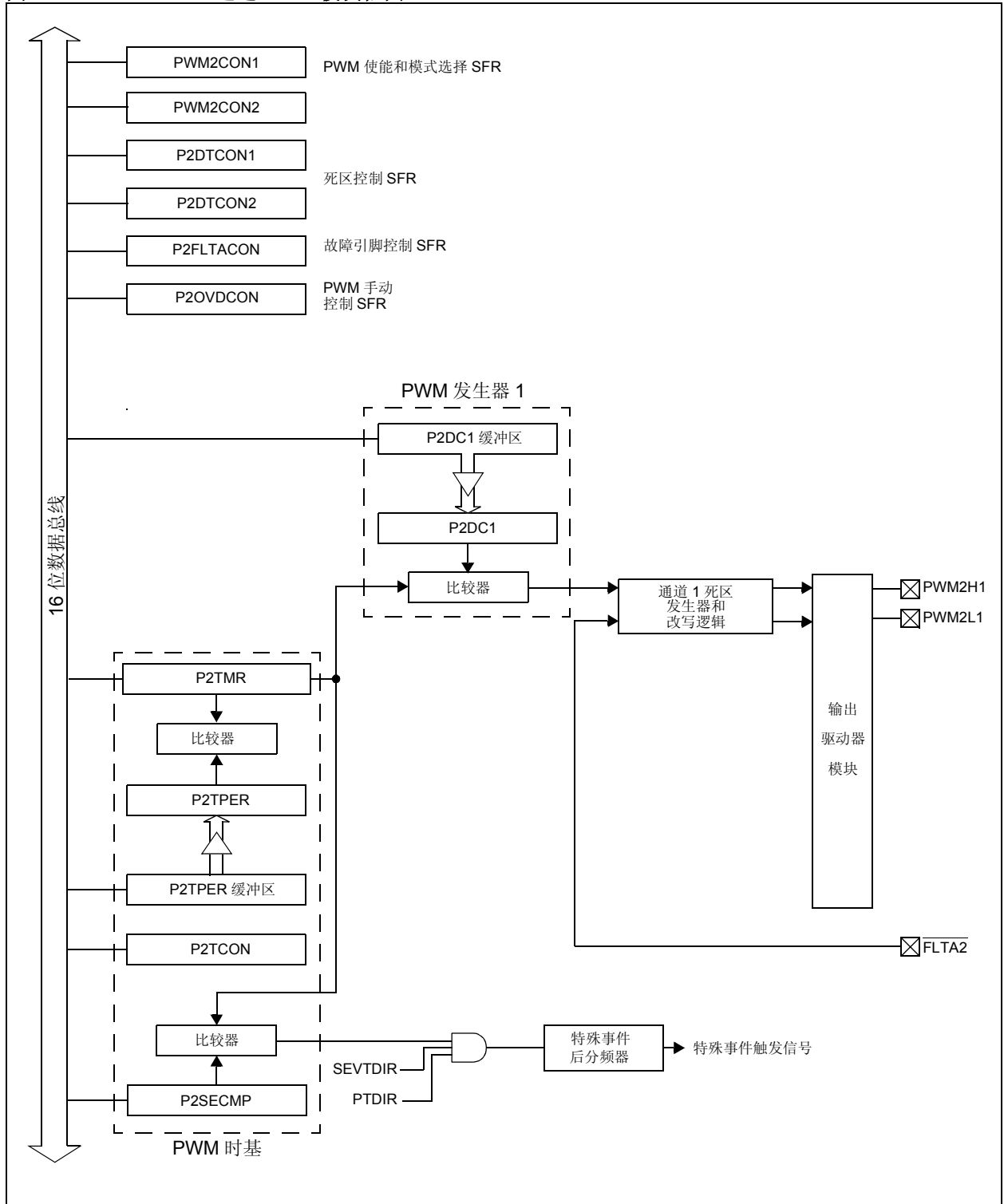


图 15-2: 2 通道 PWM 模块框图 (PWM2)



### 15.3 PWM 时基

PWM 时基由带预分频器和后分频器的 15 位定时器提供。可通过 PxTMR SFR 访问该时基。PxTMR<15> 为只读状态位 PTDIR，指示 PWM 时基当前的计数方向。

- 如果 PTDIR 清零，表示 PxTMR 正在进行递增计数。
- 如果 PTDIR 置 1，表示 PxTMR 正在进行递减计数。

通过 PxTCON SFR 对 PWM 时基进行配置。通过置 1/ 清零 PxTCON SFR 中的 PTEN 位来使能 / 禁止时基。当用软件清零 PTEN 位时，PxTMR 将不会被清零。

PxTPER SFR 设置 PxTMR 的计数周期。用户应用程序必须向 PxTPER<14:0> 中写入一个 15 位的值。当 PxTMR<14:0> 中的值与 PxTPER<14:0> 中的值匹配时，时基将复位为 0，或者在下一个时钟周期反转计数方向。所采取的操作取决于时基的工作模式。

**注：** 如果将 PWM 周期寄存器设置为 0x0000，则定时器将停止计数，且即使特殊事件值也是 0x0000，也不会产生中断和特殊事件触发信号。如果 PWM 周期寄存器已为 0x0000，则模块将不更新 PWM 周期寄存器；因此，用户应用程序必须禁止该模块以更新 PWM 周期寄存器。

PWM 时基可配置为 4 种不同的工作模式：

- 自由运行模式
- 单事件模式
- 连续递增 / 递减计数模式
- 带两次更新中断的连续递增 / 递减计数模式

通过 PxTCON SFR 中的 PTMOD<1:0> 位可选择这 4 种模式。递增 / 递减计数模式支持产生中心对齐的 PWM。单事件模式使 PWM 模块可支持某些电子换相电机 (ECM) 的脉冲控制。

PWM 时基产生的中断信号取决于 PxTCON SFR 中的模式选择位 (PTMOD<1:0>) 和后分频比位 (PTOPS<3:0>)。

#### 15.3.1 自由运行模式

在自由运行模式下，PWM 时基进行递增计数直到与 PWM 时基周期寄存器 (PxTPER) 中的值发生匹配。PxTMR 寄存器在接下来的输入时钟边沿复位，只要 PTEN 位保持置 1，时基将继续进行递增计数。

当 PWM 时基处于自由运行模式 (PTMOD<1:0> = 00) 时，每当计数值与 PxTPER 寄存器匹配时将产生中断事件，且 PxTMR 寄存器将被复位为零。在此定时器模式下，可使用后分频比选择位来降低中断事件的频率。

#### 15.3.2 单事件模式

在单事件模式下，PWM 时基在 PTEN 位置 1 时将开始递增计数。当 PxTMR 寄存器中的值与 PxTPER 寄存器匹配时，PxTMR 寄存器在接下来的输入时钟边沿复位，并且 PTEN 位将由硬件清零以暂停时基。

当 PWM 时基处于单事件模式 (PTMOD<1:0> = 01) 时，在发生与 PxTPER 寄存器匹配时将产生中断事件。PxTMR 寄存器在接下来的输入时钟边沿被复位为零，并且 PTEN 位被清零。在此定时器模式下，后分频比选择位将不起作用。

#### 15.3.3 连续递增 / 递减计数模式

在连续递增 / 递减计数模式下，PWM 时基将进行递增计数直到与 PxTPER 寄存器中的值匹配。定时器将在接下来的输入时钟边沿开始递减计数。PxTMR SFR 中的 PTDIR 位是只读位，指示计数方向。当定时器进行递减计数时，PTDIR 位将被置 1。

在递增 / 递减计数模式 (PTMOD<1:0> = 10) 下，每当 PxTMR 寄存器的值变为零时都会产生中断事件，这时 PWM 时基开始递增计数。在此定时器模式下，可使用后分频比选择位来降低中断事件的频率。



### 15.3.4 两次更新模式

在两次更新模式 (PTMOD<1:0> = 11) 下, 每当 PxTMR 寄存器等于零, 以及每当发生周期匹配时都产生中断。在此定时器模式下, 后分频比选择位将不起作用。

两次更新模式提供了两种额外的功能:

- 由于每个周期中可更新两次 PWM 占空比, 因此控制环带宽加倍。
- 可产生不对称的中心对齐 PWM 波形, 可在某些电机控制应用中用于最大限度地减少输出波形失真。

**注:** 将 PWM 周期寄存器编程设置为 0x0001 可产生连续的中断脉冲, 因此应避免这样操作。

### 15.3.5 PWM 时基预分频器

PxTMR 的输入时钟 (Fosc/4) 具有以下预分频比选项: 1:1、1:4、1:16 或 1:64, 通过 PxTCON SFR 中的控制位 PTCKPS<1:0> 选择。当出现下列事件之一时, 预分频器计数器将清零:

- 对 PxTMR 寄存器进行写操作
- 对 PxTCON 寄存器进行写操作
- 任何器件复位

写 PxTCON 时 PxTMR 寄存器不会被清零。

### 15.3.6 PWM 时基后分频器

可选择通过一个 4 位后分频器 (分频比从 1:1 到 1:16) 对 PxTMR 的匹配输出进行后分频。

当出现以下事件之一时, 后分频器计数器将清零:

- 对 PxTMR 寄存器进行写操作
- 对 PxTCON 寄存器进行写操作
- 任何器件复位

写 PxTCON 时 PxTMR 寄存器不会被清零。

## 15.4 PWM 周期

PxTPER 是一个 15 位寄存器, 用于设置 PWM 时基的计数周期。PxTPER 是一个双缓冲寄存器。PxTPER 缓冲区的內容在发生以下情形时被装入 PxTPER 寄存器:

- 自由运行和单事件模式: 当 PxTMR 寄存器在与 PxTPER 寄存器匹配后被复位为零时。
- 递增 / 递减计数模式: 当 PxTMR 寄存器为零时。

当 PWM 时基被禁止 (PTEN = 0) 时, PxTPER 缓冲区中保存的值被自动装入 PxTPER 寄存器。

PWM 周期可通过公式 15-1 确定:

### 公式 15-1: PWM 周期

$$T_{PWM} = T_{CY} \cdot (PxTPER + 1) \cdot (PxTMR \text{ 预分频值})$$

如果 PWM 时基配置为递增 / 递减计数模式之一, 则 PWM 周期为公式 15-1 得出的值的两倍。

给定器件振荡器的最大分辨率 (以位为单位) 和 PWM 频率可通过公式 15-2 确定:

### 公式 15-2: PWM 分辨率

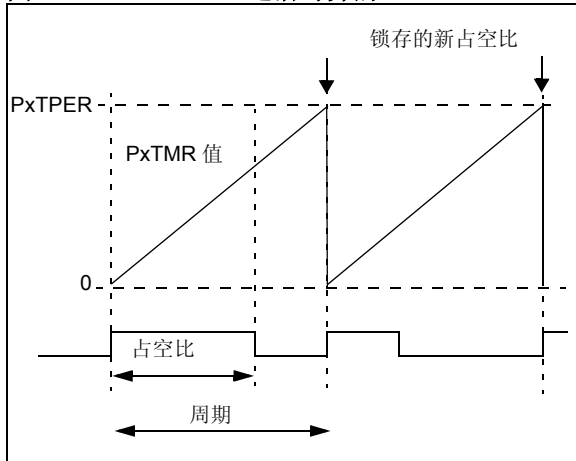
$$\text{分辨率} = \frac{\log(2 \cdot T_{PWM}/T_{CY})}{\log(2)}$$

## 15.5 边沿对齐的 PWM

当 PWM 时基处于自由运行或单事件模式时, 模块将产生边沿对齐的 PWM 信号。对于边沿对齐的 PWM 输出, 输出的周期由 PxTPER 中的值指定, 占空比由相应的占空比寄存器指定 (见图 15-3)。PWM 输出在周期开始 (PxTMR = 0) 时被驱动为有效状态; 而当占空比寄存器中的值与 PxTMR 发生匹配时, PWM 输出被驱动为无效状态。

如果特定占空比寄存器中的值为零, 则相应 PWM 引脚的输出在整个 PWM 周期内都将为无效状态。此外, 如果占空比寄存器中的值大于 PxTPER 寄存器中保存的值, 则 PWM 引脚上的输出在整个 PWM 周期内都将为有效状态。

图 15-3: 边沿对齐的 PWM



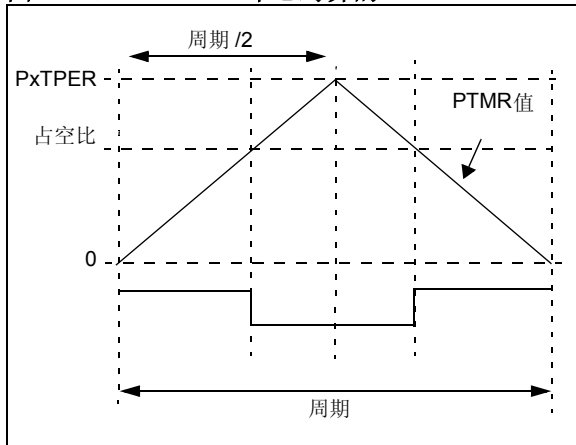
## 15.6 中心对齐的 PWM

当 PWM 时基配置为递增/递减计数模式时，模块将产生中心对齐的 PWM 信号（见图 15-4）。

当占空比寄存器的值与 PxTMR 的值匹配，并且 PWM 时基进行递减计数（PTDIR = 1）时，PWM 比较输出将被驱动为有效状态。当 PWM 时基进行递增计数（PTDIR = 0），并且 PxTMR 寄存器中的值与占空比值匹配时，PWM 比较输出将被驱动为无效状态。

如果某个占空比寄存器中的值为零，则相应 PWM 引脚的输出在整个 PWM 周期内都将为无效状态。此外，如果占空比寄存器中的值等于 PxTPER 寄存器中保存的值，则 PWM 引脚上的输出在整个 PWM 周期内都将为有效状态。

图 15-4: 中心对齐的 PWM



## 15.7 PWM 占空比比较单元

有 3 个 16 位特殊功能寄存器（PxDC1、PxDC2 和 PxDC3）用于指定 PWM 模块的占空比值。

每个占空比寄存器中的值确定 PWM 输出处于有效状态的时间。占空比寄存器为 16 位宽。占空比寄存器的最低有效位（LSb）确定是否在开始时出现 PWM 边沿。因此，PWM 分辨率实际上翻倍。

### 15.7.1 占空比寄存器缓冲区

3 个 PWM 占空比寄存器都是双缓冲的，以使 PWM 输出更新时不会产生毛刺。对于每个占空比，都有可由用户应用程序访问的占空比寄存器和保存当前 PWM 周期中所使用的实际比较值的占空比寄存器。

对于边沿对齐的 PWM 输出，当计数值与 PxTPER 寄存器值匹配且 PxTMR 被复位时，将更新占空比值。当 PWM 时基被禁止（PTEN = 0）且 PWMxCON2 中的 UDIS 位被清零时，占空比缓冲区的内容会被自动装入占空比寄存器。

如果 PWM 时基处于递增/递减计数模式，则当 PxTMR 寄存器的值为零且 PWM 时基开始递增计数时，将更新占空比值。当 PWM 时基被禁止（PTEN = 0）时，占空比缓冲区的内容会被自动装入占空比寄存器。

如果 PWM 时基处于带两次更新功能的递增/递减计数模式，则当 PxTMR 寄存器的值为零或 PxTMR 寄存器的值与 PxTPER 寄存器中的值匹配时，都将更新占空比值。当 PWM 时基被禁止（PTEN = 0）时，占空比缓冲区的内容会被自动装入占空比寄存器。

### 15.7.2 占空比立即更新

当立即更新使能位置 1 (IUE = 1) 时, 任何对占空比寄存器的写操作均将立即更新占空比值。此功能为程序员提供了一种选择, 使其可以立即更新有效的 PWM 占空比寄存器, 而不必等到当前时基周期结束。占空比更新作用如下:

- 如果新占空比写入时 PWM 输出有效, 且新占空比小于当前时基值, 则 PWM 脉宽将变窄。
- 如果新占空比写入时 PWM 输出有效, 且新占空比大于当前时基值, 则 PWM 脉宽将变宽。
- 如果新占空比写入时 PWM 输出无效, 且新占空比大于当前时基值, 则 PWM 输出将立即变为有效, 并对新写入的占空比值保持有效。

在闭环伺服应用中, 当使能立即更新 (IUE = 1) 时, 可通过缩短检测到系统故障和发出系统校正命令之间的延时而提高系统稳定性。

### 15.8 互补 PWM 操作

在互补工作模式下, 每对 PWM 输出引脚都将输出互补的 PWM 信号。可选择在器件开关过程中插入一个死区, 使得在这段短暂的时间内两个输出均为无效。(见第 15.9 节“死区发生器”)。

在互补模式下, 占空比较单元被分配给 PWM 输出使用, 如下所示:

- PxDC1 寄存器控制 PWM1H/PWM1L 输出
- PxDC2 寄存器控制 PWM2H/PWM2L 输出
- PxDC3 寄存器控制 PWM3H/PWM3L 输出

通过将 PWMxCON1 SFR 中相应的 PMODx 位清零, 可为每个 PWM I/O 引脚对选择互补模式。在器件复位时, PWM I/O 引脚默认设置为互补模式。

### 15.9 死区发生器

当任一 PWM I/O 引脚对工作于互补输出模式时, 可以使用死区发生功能。PWM 输出采用推挽驱动电路。功率输出器件不能瞬时完成切换, 因此在一对互补 PWM 输出中一个开关晶体管的关断和另一个晶体管的导通之间必须间隔一定的时间。

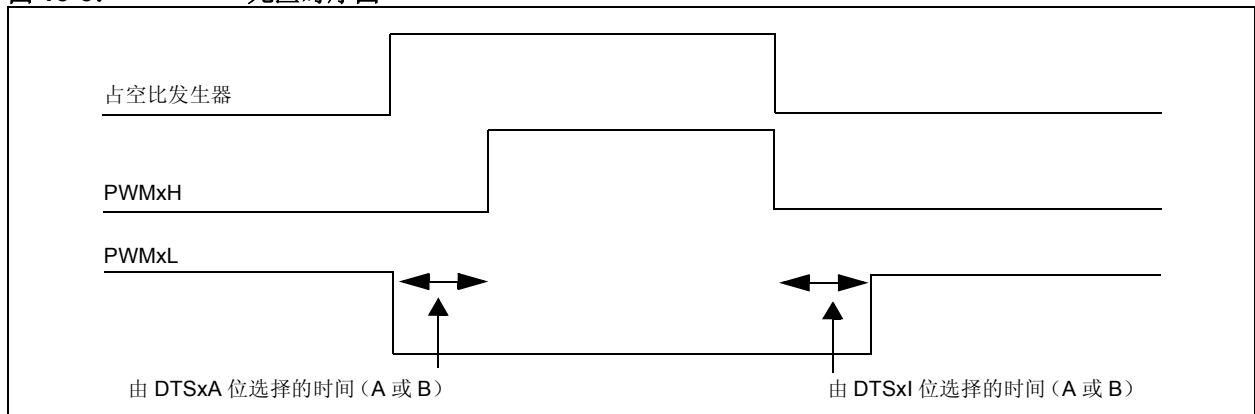
PWM 模块允许编程设定两个不同的死区。这两个死区可以用以下两种方法之一来提高用户灵活性:

- 可以对 PWM 输出信号进行优化使一对互补晶体管中的高端和低端晶体管的关断时间不同。在互补对低端晶体管的关断事件和高端晶体管的导通事件之间插入第一个死区。在高端晶体管的关断事件和低端晶体管的导通事件之间插入第二个死区。
- 可将两个死区分配给各 PWM I/O 引脚对。此工作模式使 PWM 模块可以用每一个互补的 PWM I/O 引脚对驱动不同的晶体管 / 负载组合。

#### 15.9.1 死区发生器

PWM 模块的每一个互补输出对都有一个 6 位的递减计数器, 用于插入死区。如图 15-5 所示, 每个死区单元都有与占空比较输出相连的上升沿和下降沿检测器。

图 15-5: 死区时序图



**15.9.2 死区分配**

PxDTCN2 SFR 包含可以将死区分配给每个互补输出对的控制位。表 15-1 总结了每个死区选择控制位的功能。

**表 15-1: 死区选择位**

位	功能
DTS1A	选择 PWMxL1/PWMxH1 有效边沿死区。
DTS1I	选择 PWMxL1/PWMxH1 无效边沿死区。
DTS2A	选择 PWMxL2/PWMxH2 有效边沿死区。
DTS2I	选择 PWMxL2/PWMxH2 无效边沿死区。
DTS3A	选择 PWMxL3/PWMxH3 有效边沿死区。
DTS3I	选择 PWMxL3/PWMxH3 无效边沿死区。

**15.9.3 死区范围**

每个死区单元提供的死区大小是通过指定输入时钟预分频值和一个 6 位无符号值来选择的。可以单独设置每个死区单元所提供的死区大小。

通过所提供的 4 种输入时钟预分频比选择，用户可以根据器件的工作频率选择适当的死区范围。可以为两个死区值之一单独选择时钟预分频比选项。死区时钟预分频值是使用 PxDTCN1 SFR 中的 DTAPS<1:0> 和 DTBPS<1:0> 控制位选择的。每个死区值可选择 4 个时钟预分频选项之一 (Tcy、2 Tcy、4 Tcy 或 8 Tcy)。

选择预分频值后，通过向 PxDTCN1 SFR 中装入两个 6 位无符号值，可对每个单元的死区进行调整。

死区单元预分频器在发生以下事件时清零：

- 由于发生占空比比较边沿事件而装入递减定时器时。
- 对 PxDTCN1 或 PxDTCN2 寄存器进行写操作时。
- 任何器件复位时。

**注：** 当 PWM 模块正在工作 (PTEN = 1) 时，用户应用程序不应修改 PxDTCN1 或 PxDTCN2 的值。因为这样做可能造成预料不到的结果。

## 15.10 独立的 PWM 输出

在驱动某些类型的负载时，需要采用独立的 PWM 输出模式。当 PWMxCON1 寄存器中相应的 PMODx 位置 1 时，则相应的 PWM 输出对就处于独立输出模式。当模块工作于独立的 PWM 输出模式时，相邻 PWM I/O 引脚之间不实现死区控制，允许两个 I/O 引脚同时处于有效状态。

在独立的 PWM 输出模式下，每个占空比发生器同时连接到 PWM 输出对中的两个 PWM I/O 引脚。通过使用相关的占空比寄存器和 PxOVDCON 寄存器中的相应位，程序员可以为工作于该模式下的每个 PWM I/O 引脚选择以下信号输出选项：

- I/O 引脚输出 PWM 信号
- I/O 引脚处于无效状态
- I/O 引脚处于有效状态

## 15.11 单脉冲 PWM 操作

当 PxTCON 控制位 PTMOD<1:0> = 10 时，PWM 模块产生单脉冲输出。在单脉冲模式下，只能产生边沿对齐的输出。在单脉冲模式下，当 PTEN 位置 1 时，PWM I/O 引脚被驱动为有效状态。当计数值与占空比寄存器发生匹配时，PWM I/O 引脚被驱动为无效状态。当计数值与 PxTPER 寄存器发生匹配时，PxTMR 寄存器清零，所有的有效 PWM I/O 引脚被驱动为无效状态，PTEN 位清零并产生中断。

## 15.12 PWM 输出改写

PWM 输出改写位允许用户应用程序手动将 PWM I/O 引脚驱动为指定的逻辑状态，而不受占空比比较单元的影响。

所有与 PWM 输出改写功能相关的控制位都包含在 PxOVDCON 寄存器中。PxOVDCON 寄存器的高字节包含 8 个位 (POVDxH<4:1> 和 POVDxL<4:1>)，这些位确定改写哪些 PWM I/O 引脚。PxOVDCON 寄存器的低字节也包含 8 个位 (POUTxH<4:1> 和 POUTxL<4:1>)，当通过 POVD 位改写某个输出时，可以使用这些位确定 PWM I/O 引脚的状态。

### 15.12.1 互补输出模式

当通过 PxOVDCON 寄存器将 PWMxL 引脚驱动为有效状态时，输出信号将被强制为与输出对应对应 PWMxH 引脚输出状态相反的状态。当手动改写 PWM 通道时，仍然执行死区插入。

### 15.12.2 改写同步

如果 PWMxCON2 寄存器中的 OSYNC 位置 1，则所有通过 PxOVDCON 寄存器执行的输出改写将与 PWM 时钟同步。同步输出改写发生在以下时间：

- 边沿对齐模式——当 PxTMR 为零时
- 中心对齐模式——当 PxTMR 为零时，且 PxTMR 值与 PxTPER 匹配时

## 15.13 PWM 输出和极性控制

3 个与 PWM 模块相关的器件配置位用来提供 PWM 输出引脚控制：

- HPOL 配置位
- LPOL 配置位
- PWMPIN 配置位

FPOR 配置寄存器中的这 3 个位（见第 27.0 节“特殊功能”）与位于 PWMxCON1 SFR 中的 8 个 PWM 使能位 (PENxH<4:1> 和 PENxL<4:1>) 配合工作。配置位和 PWM 使能位确保在发生器件复位后，PWM 引脚处于正确的状态。

通过 PWMPIN 配置熔丝，可以选择在器件复位时使能 PWM 模块输出。如果 PWMPIN = 0，PWM 输出在复位时被驱动为无效状态。如果 PWMPIN = 1（默认），PWM 输出将为三态。HPOL 位指定 PWMxH 输出的极性，而 LPOL 位指定 PWMxL 输出的极性。

### 15.13.1 输出引脚控制

PWMxCON1 SFR 中的 PENxH<4:1> 和 PENxL<4:1> 控制位分别用于使能每个高端 PWM 输出引脚和每个低端 PWM 输出引脚。如果没有使能某个 PWM 输出引脚，则将其视为通用 I/O 引脚。

## 15.14 PWM 故障引脚

有一个与 PWM 模块相关的故障引脚 ( $\overline{\text{FLTAX}}$ )。当有效时，可选择通过该引脚将每个 PWM I/O 引脚驱动为定义的状态。

### 15.14.1 故障引脚使能位

PxFLTACON SFR 具有 4 个控制位，用来决定特定 PWM I/O 引脚对是否由故障输入引脚控制。要使能某一 PWM I/O 引脚对的故障改写功能，需要将 PxFLTACON 寄存器中的相应位置 1。

如果 PxFLTACON 寄存器中所有的使能位都被清零，则相应的故障输入引脚对 PWM 模块没有影响，可以将引脚用作通用中断或 I/O 引脚。

**注：** 故障引脚逻辑可以独立于 PWM 逻辑进行工作。如果将 PxFLTACON 寄存器中所有的使能位都清零，则可以将故障引脚用作通用中断引脚。每个故障引脚都具有与其相关的中断向量、中断标志位和中断优先级位。

### 15.14.2 故障状态

PxFLTACON 特殊功能寄存器具有 8 个位，用来决定每个 PWM I/O 引脚在被故障输入改写时的状态。当这些位清零时，PWM I/O 引脚被驱动为无效状态。当相应位置 1 时，PWM I/O 引脚被驱动为有效状态。有效和无效状态与为每个 PWM I/O 引脚定义的极性（通过 HPOL 和 LPOL 极性控制位设置）相对应。

当 PWM 模块的一个 I/O 对处于互补模式，并且两个引脚都编程为在产生故障条件时驱动为有效时，存在一种特殊情况。在互补模式下，PWMxH 引脚始终优先，因此两个 I/O 引脚不能同时被驱动为有效状态。

### 15.14.3 故障引脚优先级

如果两个故障输入引脚都被分配为控制某一个 PWM I/O 引脚，则为故障 A 输入引脚设置的故障状态将优先于故障 B 输入引脚。

### 15.14.4 故障输入模式

每个故障输入引脚都有两种工作模式：

- **锁定模式：**当故障引脚驱动为低电平时，PWM 输出将进入 PxFLTACON 寄存器定义的状态。PWM 输出将保持在此状态，直到故障引脚被驱动为高电平并且相应的中断标志用软件清零。当这两个操作都发生后，PWM 输出将在下一个 PWM 周期开始时或半周期边界返回到正常工作状态。如果中断标志在故障条件结束前清零，则 PWM 模块将等到故障引脚不再有效时才恢复输出。
- **逐周期模式：**当故障输入引脚被驱动为低电平时，只要故障引脚保持为低电平，PWM 输出就会一直保持定义的故障状态。在故障引脚被驱动为高电平后，PWM 输出将在下一个 PWM 周期开始时或半周期边界返回正常工作状态。

每个故障输入引脚的工作模式可通过 PxFLTACON 特殊功能寄存器中的 FLTAM 控制位进行选择。

可以用软件对各故障引脚进行手动控制。

## 15.15 PWM 更新锁定

对于复杂 PWM 应用，用户应用可能需要在给定时间对最多 3 个占空比寄存器以及 PWM 时基周期寄存器 PxTPER 进行写操作。在某些应用中，在对模块装入新占空比和周期值之前写入所有的缓冲寄存器是很重要的。

通过将 PWM1CON2 SFR 中的 UDIS 控制位置 1 可启用 PWM 更新锁定功能。UDIS 位会影响所有的占空比缓冲寄存器和 PWM 时基周期寄存器 PxTPER。当 UDIS = 1 时，占空比更改或周期值更改都不会生效。

如果将 IUE 位置 1，则不管 UDIS 位的状态如何，对占空比寄存器的任何更改都将立即得到更新。PWM 周期寄存器（PxTPER）更新不受 IUE 控制位的影响。

## 15.16 PWM 特殊事件触发器

PWM 模块有一个特殊事件触发器，可以使 ADC 转换与 PWM 时基同步。可以将 ADC 采样和转换时间编程为在 PWM 周期内的任何时间发生。特殊事件触发器可以使程序员将获得 ADC 转换结果的时间与占空比值更新的时间之间的延时减到最小。

PWM 特殊事件触发器使用一个称为 PxSECMP 的 SFR 和 5 个控制位对其操作进行控制。用于产生特殊事件触发信号的 PxTMR 值被装入 PxSECMP 寄存器中。

当 PWM 时基处于递增 / 递减计数模式时，还需要一个控制位指定特殊事件触发信号的计数阶段。计数阶段通过 PxSECMP SFR 中的 SEVTDIR 控制位进行选择：

- 如果 SEVTDIR 位清零，则特殊事件触发信号将在 PWM 时基的递增计数周期产生。
- 如果 SEVTDIR 位置 1，则特殊事件触发信号将在 PWM 时基的递减计数周期产生。

如果不将 PWM 时基配置为递增 / 递减计数模式，则 SEVTDIR 控制位不起作用。

### 15.16.1 特殊事件触发器后分频器

PWM 特殊事件触发器有一个允许后分频比为 1:1 到 1:16 的后分频器。通过写 PWMxCON2 SFR 中的 SEVOPS<3:0> 控制位配置后分频器。

特殊事件输出后分频器在发生以下事件时清零：

- 对 PxSECMP 寄存器的任何写操作
- 任何器件复位

## 15.17 CPU 休眠模式下的 PWM 操作

故障 A 和故障 B 输入引脚能够将 CPU 从休眠模式唤醒。在休眠模式下，如果任一故障引脚被驱动为低电平，则 PWM 模块将产生中断。

## 15.18 CPU 空闲模式下的 PWM 操作

PxTCON SFR 包含 PTSIDL 控制位。该位用于确定当器件进入空闲模式时 PWM 模块是继续工作还是停止工作。如果 PTSIDL = 0，则模块将继续工作。如果 PTSIDL = 1，只要 CPU 仍然处于空闲模式，模块将停止工作。

**寄存器 15-1: P<sub>x</sub>TCON: PWM 时基控制寄存器**

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
PTEN	—	PTSIDL	—	—	—	—	—
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
PTOPS<3:0>				PTCKPS<1:0>		PTMOD<1:0>	
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            **PTEN: PWM 时基定时器使能位**  
 1 = PWM 时基开启  
 0 = PWM 时基关闭
- bit 14            **未实现:** 读为 0
- bit 13            **PTSIDL: PWM 时基空闲模式停止位**  
 1 = PWM 时基在 CPU 空闲模式下暂停  
 0 = PWM 时基在 CPU 空闲模式下运行
- bit 12-8        **未实现:** 读为 0
- bit 7-4           **PTOPS<3:0>: PWM 时基输出后分频比选择位**  
 1111 = 1:16 后分频比  
 .  
 .  
 .  
 0001 = 1:2 后分频比  
 0000 = 1:1 后分频比
- bit 3-2          **PTCKPS<1:0>: PWM 时基输入时钟预分频比选择位**  
 11 = PWM 时基输入时钟周期为 64 个 T<sub>CY</sub> (1:64 预分频比)  
 10 = PWM 时基输入时钟周期为 16 个 T<sub>CY</sub> (1:16 预分频比)  
 01 = PWM 时基输入时钟周期为 4 个 T<sub>CY</sub> (1:4 预分频比)  
 00 = PWM 时基输入时钟周期为 T<sub>CY</sub> (1:1 预分频比)
- bit 1-0          **PTMOD<1:0>: PWM 时基模式选择位**  
 11 = PWM 时基在连续递增 / 递减计数模式下工作, 且在双 PWM 更新时产生中断  
 10 = PWM 时基在连续递增 / 递减计数模式下工作  
 01 = PWM 时基在单脉冲模式下工作  
 00 = PWM 时基在自由运行模式下工作



**寄存器 15-2: PxTMR: PWM 定时器计数值寄存器**

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
PTDIR	PTMR<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	
PTMR<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            **PTDIR:** PWM 时基计数方向状态位 (只读)  
                   1 = PWM 时基递减计数  
                   0 = PWM 时基递增计数  
 bit 14-0        **PTMR <14:0>:** PWM 时基寄存器计数值位

**寄存器 15-3: PxTPER: PWM 时基周期寄存器**

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	PTPER<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	
PTPER<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 14-0        **PTPER<14:0>:** PWM 时基周期值位

**寄存器 15-4: PxSECMP: 特殊事件比较寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SEVTDIR <sup>(1)</sup>	SEVTCMP<14:8> <sup>(2)</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
SEVTCMP<7:0> <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      **SEVTDIR:** 特殊事件触发信号时基方向位 <sup>(1)</sup>  
 1 = 当 PWM 时基递减计数时, 产生特殊事件触发信号  
 0 = 当 PWM 时基递增计数时, 产生特殊事件触发信号

bit 14-0    **SEVTCMP<14:0>:** 特殊事件比较值位 <sup>(2)</sup>

- 注 1: SEVTDIR 与 PTDIR (PxTMR<15>) 作比较, 以产生特殊事件触发信号。  
 2: PxSECMP<14:0> 与 PxTMR<14:0> 作比较, 以产生特殊事件触发信号。

寄存器 15-5: PWMxCON1: PWM 控制寄存器 1<sup>(2)</sup>

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PMOD3	PMOD2	PMOD1
bit 15					bit 8		
U-0	R/W-1	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1
—	PEN3H <sup>(1)</sup>	PEN2H <sup>(1)</sup>	PEN1H <sup>(1)</sup>	—	PEN3L <sup>(1)</sup>	PEN2L <sup>(1)</sup>	PEN1L <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 15-11      **未实现:** 读为 0
- bit 10-8      **PMOD4:PMOD1:** PWM I/O 对模式位  
 1 = PWM I/O 引脚对处于独立 PWM 输出模式  
 0 = PWM I/O 引脚对处于互补输出模式
- bit 7          **未实现:** 读为 0
- bit 6-4      **PEN3H:PEN1H:** PWMxH I/O 使能位 <sup>(1)</sup>  
 1 = 使能 PWMxH 引脚用于 PWM 输出  
 0 = 禁止 PWMxH 引脚, I/O 引脚变为通用 I/O
- bit 3          **未实现:** 读为 0
- bit 2-0      **PEN3L:PEN1L:** PWMxL I/O 使能位 <sup>(1)</sup>  
 1 = 使能 PWMxL 引脚用于 PWM 输出  
 0 = 禁止 PWMxL 引脚, I/O 引脚变为通用 I/O

- 注 1: PENxH 和 PENxL 位的复位状态取决于 FPOR 配置寄存器中 PWMPIN 配置位的值。
- 2: PWM2 仅支持 1 个 PWM I/O 引脚对。

**寄存器 15-6: PWMxCON2: PWM 控制寄存器 2**

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	SEVOPS<3:0>			
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	IUE	OSYNC	UDIS
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      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 11-8     **SEVOPS<3:0>:** PWM 特殊事件触发信号输出后分频比选择位
  - 1111 = 1:16 后分频比
  - 
  - 
  - 
  - 0001 = 1:2 后分频比
  - 0000 = 1:1 后分频比
- bit 7-3     **未实现:** 读为 0
- bit 2        **IUE:** 立即更新使能位
  - 1 = 立即更新有效 PxDC 寄存器
  - 0 = 有效 PxDC 寄存器的更新与 PWM 时基同步
- bit 1        **OSYNC:** 输出改写同步位
  - 1 = 通过 PxOVDCON 寄存器进行的输出改写与 PWM 时基同步
  - 0 = 通过 PxOVDCON 寄存器进行的输出改写在下一个 Tcy 边界发生
- bit 0        **UDIS:** PWM 更新禁止位
  - 1 = 禁止来自占空比寄存器和周期缓冲寄存器的更新
  - 0 = 使能来自占空比寄存器和周期缓冲寄存器的更新

寄存器 15-7: PxDTCN1: 死区控制寄存器 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
DTBPS<1:0>		DTB<5:0>					
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
DTAPS<1:0>		DTA<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 15-14     **DTBPS<1:0>**: 死区单元 B 预分频比选择位  
 11 = 死区单元 B 的时钟周期为 8 个 T<sub>cY</sub>  
 10 = 死区单元 B 的时钟周期为 4 个 T<sub>cY</sub>  
 01 = 死区单元 B 的时钟周期为 2 个 T<sub>cY</sub>  
 00 = 死区单元 B 的时钟周期为 T<sub>cY</sub>
- bit 13-8     **DTB<5:0>**: 死区单元 B 的无符号 6 位死区值位
- bit 7-6      **DTAPS<1:0>**: 死区单元 A 预分频比选择位  
 11 = 死区单元 A 的时钟周期为 8 个 T<sub>cY</sub>  
 10 = 死区单元 A 的时钟周期为 4 个 T<sub>cY</sub>  
 01 = 死区单元 A 的时钟周期为 2 个 T<sub>cY</sub>  
 00 = 死区单元 A 的时钟周期为 T<sub>cY</sub>
- bit 5-0      **DTA<5:0>**: 死区单元 A 的无符号 6 位死区值位

**寄存器 15-8: PxDTCON2: 死区控制寄存器 2<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
—	—	DTS3A	DTS3I	DTS2A	DTS2I	DTS1A	DTS1I
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      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            **DTS3A:** PWMxH3 信号变为有效的死区选择位  
                   1 = 死区由单元 B 提供  
                   0 = 死区由单元 A 提供
- bit 4            **DTS3I:** PWMxL3 信号变为无效的死区选择位  
                   1 = 死区由单元 B 提供  
                   0 = 死区由单元 A 提供
- bit 3            **DTS2A:** PWMxH2 信号变为有效的死区选择位  
                   1 = 死区由单元 B 提供  
                   0 = 死区由单元 A 提供
- bit 2            **DTS2I:** PWMxL2 信号变为无效的死区选择位  
                   1 = 死区由单元 B 提供  
                   0 = 死区由单元 A 提供
- bit 1            **DTS1A:** PWMxH1 信号变为有效的死区选择位  
                   1 = 死区由单元 B 提供  
                   0 = 死区由单元 A 提供
- bit 0            **DTS1I:** PWMxL1 信号变为无效的死区选择位  
                   1 = 死区由单元 B 提供  
                   0 = 死区由单元 A 提供

**注 1:** PWM2 仅支持 1 个 PWM I/O 引脚对。

寄存器 15-9: P<sub>x</sub>FLTACON: 故障 A 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
—	—	FAOV3H	FAOV3L	FAOV2H	FAOV2L	FAOV1H	FAOV1L
bit 15						bit 8	

R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
FLTAM	—	—	—	—	FAEN3	FAEN2	FAEN1
bit 7						bit 0	

图注:

R = 可读位                      W = 可写位                      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      **FAOV<sub>x</sub>H<3:1>:FAOV<sub>x</sub>L<3:1>**: 故障输入 A PWM 改写值位  
 1 = 在发生外部故障输入事件时, PWM 输出引脚被驱动为有效状态  
 0 = 在发生外部故障输入事件时, PWM 输出引脚被驱动为无效状态
- bit 7          **FLTAM:** 故障 A 模式位  
 1 = 故障 A 输入引脚在逐周期模式下起作用  
 0 = 故障 A 输入引脚将所有控制引脚锁定为 P<sub>x</sub>FLTACON<13:8> 中编程设定的状态
- bit 6-3      **未实现:** 读为 0
- bit 2          **FAEN3:** 故障输入 A 使能位  
 1 = PWM<sub>x</sub>H3/PWM<sub>x</sub>L3 引脚对由故障输入 A 控制  
 0 = PWM<sub>x</sub>H3/PWM<sub>x</sub>L3 引脚对不受故障输入 A 控制
- bit 1          **FAEN2:** 故障输入 A 使能位  
 1 = PWM<sub>x</sub>H2/PWM<sub>x</sub>L2 引脚对由故障输入 A 控制  
 0 = PWM<sub>x</sub>H2/PWM<sub>x</sub>L2 引脚对不受故障输入 A 控制
- bit 0          **FAEN1:** 故障输入 A 使能位  
 1 = PWM<sub>x</sub>H1/PWM<sub>x</sub>L1 引脚对由故障输入 A 控制  
 0 = PWM<sub>x</sub>H1/PWM<sub>x</sub>L1 引脚对不受故障输入 A 控制

注 1: PWM2 仅支持 1 个 PWM I/O 引脚对。

**寄存器 15-10: PxOVDCON: 改写控制寄存器 (1)**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	POVD3H	POVD3L	POVD2H	POVD2L	POVD1H	POVD1L
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
—	—	POUT3H	POUT3L	POUT2H	POUT2L	POUT1H	POUT1L
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      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      **POVDxH<3:1>:POVDxL<3:1>:** PWM 输出改写位  
                     1 = PWMx I/O 引脚上的输出由 PWM 发生器控制  
                     0 = PWMx I/O 引脚上的输出由对应 POUTxH:POUTxL 位中的值控制
- bit 7-6        **未实现:** 读为 0
- bit 5-0        **POUTxH<3:1>:POUTxL<3:1>:** PWM 手动输出位  
                     1 = 当对应的 POVDxH:POVDxL 位清零时, PWMx I/O 引脚被驱动为有效状态  
                     0 = 当对应的 POVDxH:POVDxL 位清零时, PWMx I/O 引脚被驱动为无效状态

**注 1:** PWM2 仅支持 1 个 PWM I/O 引脚对。



**寄存器 15-11: PxDC1: PWM 占空比寄存器 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
PDC1<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	R/W-0
PDC1<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 15-0            **PDC1<15:0>**: PWM 占空比 1 值位

**寄存器 15-12: P1DC2: PWM 占空比寄存器 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
PDC2<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	R/W-0
PDC2<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 15-0            **PDC2<15:0>**: PWM 占空比 2 值位

**寄存器 15-13: P1DC3: PWM 占空比寄存器 3**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PDC3<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	R/W-0
PDC3<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 15-0            **PDC3<15:0>**: PWM 占空比 3 值位

注:

## 16.0 正交编码器接口 (QEI) 模块

**注:** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息,请参考“dsPIC33F Family Reference Manual”的“Section 15. Quadrature Encoder Interface (QEI)” (DS70208), 该文档可从 Microchip 网站 (www.microchip.com) 下载。

本章介绍了正交编码器接口 (Quadrature Encoder Interface, QEI) 模块和相关的工作模式。QEI 模块提供了与用于获得机械位置数据的增量式编码器的接口。

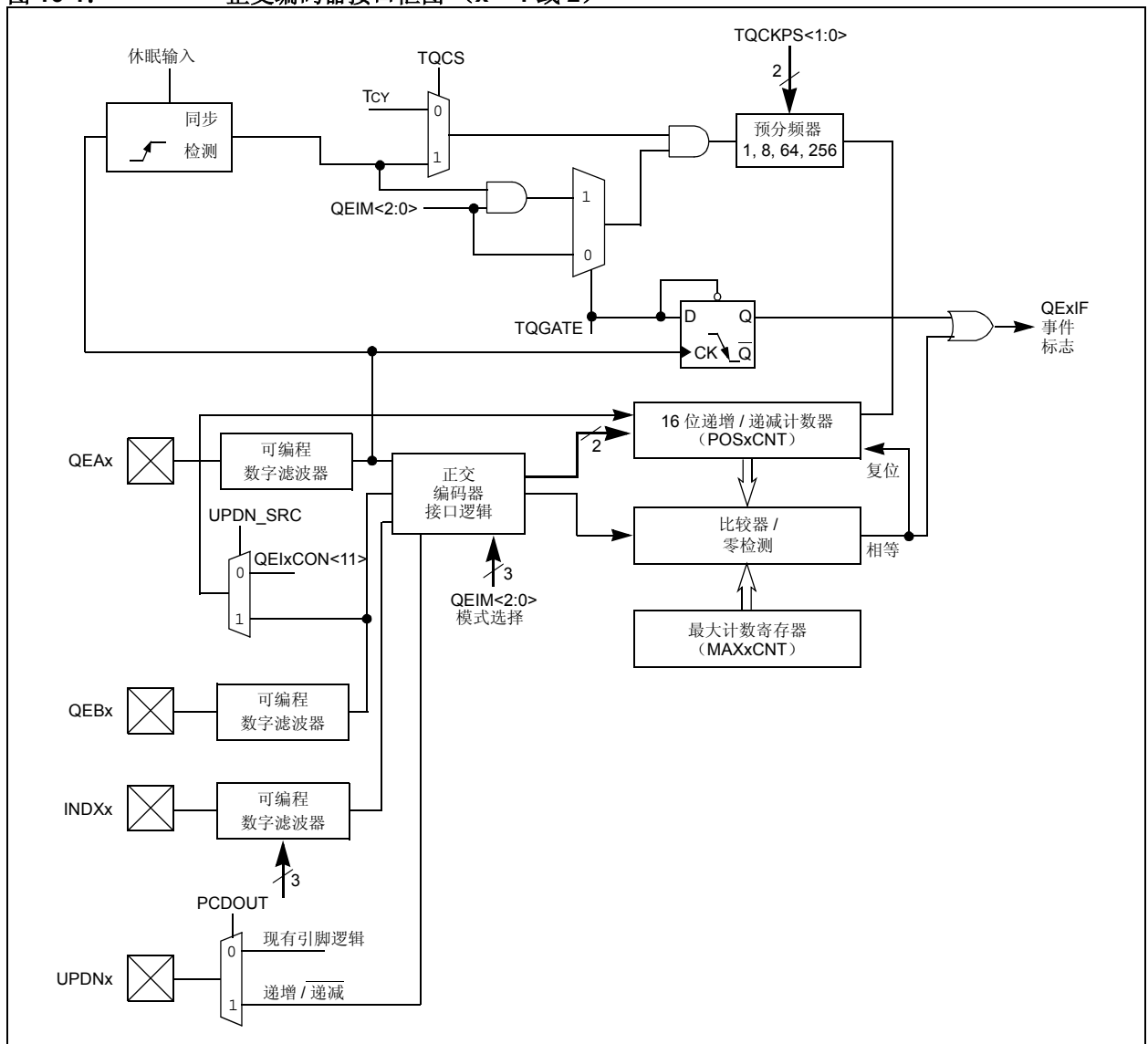
QEI 的工作特性包括:

- 3 路输入通道, 分别为两相信号和索引脉冲输入
- 16 位递增 / 递减位置计数器
- 计数方向状态
- 位置测量 (x2 和 x4) 模式
- 输入端上的可编程数字噪声滤波器
- 备用 16 位定时器 / 计数器模式
- 正交编码器接口中断

通过设置适当的位 QEIM<2:0> (QEIxCON<10:8>) 来决定这些工作模式。图 16-1 给出了正交编码器接口的框图。

**注:** 引脚名、控制 / 状态位名和寄存器名中的“x”表示某个正交编码器接口 (QEI) 模块的编号 (x = 1 或 2)。

图 16-1: 正交编码器接口框图 (x = 1 或 2)



## 16.1 正交编码器接口逻辑

典型的增量式（也称为光电式）编码器具有 3 路输出：A 相、B 相和索引脉冲。在交流感应电机（ACIM）和开关磁阻（SR）电机的位置和速度控制中，这三个信号非常有用，且通常是必需的。

A 相（QEAx）和 B 相（QEBx）这两个通道具有独特的关系。如果 A 相超前 B 相，那么认为电机正向旋转。如果 A 相滞后于 B 相，那么认为电机反向旋转。

第三个通道称为索引脉冲（INDXx），每转一圈产生一个脉冲，作为参考用来确定绝对位置。索引脉冲与 A 相和 B 相一致，皆为低电平。

## 16.2 16 位递增 / 递减位置计数器模式

16 位递增 / 递减计数器在每一个计数脉冲时递增或递减计数，计数脉冲信号由 A 相和 B 相输入信号之间的相位差确定。计数器作为一个积分器，其计数值与位置成比例关系。计数方向由 UPDN 信号决定，该信号由正交编码器接口逻辑产生。

### 16.2.1 位置计数器错误检查

QEI 中提供了位置计数器错误检查功能，这由 CNTERR 位（QEIXCON<15>）指示。只有当位置计数器配置为在索引脉冲模式下复位（QEIM<2:0> = 110 或 100）时，才能支持错误检查功能。在这些模式下，POSxCNT 寄存器的内容与数值 0xFFFF 或 MAXxCNT + 1（取决于方向）进行比较。

如果检测到这些值，通过将 CNTERR 位置 1 产生错误条件，并产生 QEI 计数器错误中断。通过将 CEID 位（DFLTxCNTCON<8>）置 1 可以禁止 QEI 计数器错误中断。

检测到错误后，位置计数器仍将继续对编码器边沿进行计数。POSxCNT 寄存器继续递增 / 递减计数，直到发生自然计满返回 / 下溢。发生自然计满返回 / 下溢事件时不产生中断。

CNTERR 位是读 / 写位，由用户应用程序用软件复位。

### 16.2.2 位置计数器复位

位置计数器复位使能位 POSRES（QEI<2>）控制当检测到索引脉冲时是否对位置计数器进行复位。只有当 QEIM<2:0> = 100 或 110 时，该位才适用。

如果 POSRES 位设置为 1，则位置计数器在检测到索引脉冲时将被复位。如果 POSRES 位设置为 0，则位置计数器在检测到索引脉冲时不会被复位。位置计数器将继续进行递增或递减计数，在发生计满返回或下溢时复位。

在检测到索引脉冲时仍将产生中断，但在位置计数器发生溢出 / 下溢时不产生中断。

### 16.2.3 计数方向状态

QEI 逻辑根据 A 相和 B 相之间的关系产生 UPDN 信号。除输出引脚外，还将此内部 UPDN 信号的状态传送给 SFR 的 UPDN 位（QEIXCON<11>），该位是只读位。要将信号状态送至 I/O 引脚，则必须将 SFR 的 PCDOUT 位（QEIXCON<6>）置为 1。

## 16.3 位置测量模式

QEI 支持两种测量模式 x2 和 x4。这些模式通过位于 SFR QEIXCON<10:8> 中的 QEIM<2:0> 模式选择位进行选择。

当控制位 QEIM<2:0> = 100 或 101 时，将选择 x2 测量模式，此时 QEI 逻辑将只通过 A 相输入信号来确定位置计数器递增速率。A 相信号的每个上升沿和下降沿都会导致位置计数器递增或递减。B 相信号仍用于计数器方向的确定。

在 x2 测量模式下，存在导致位置计数器复位的两种不同原因：

- 通过检测到索引脉冲将位置计数器复位，QEIM<2:0> = 100。
- 通过计数值与 MAXxCNT 匹配将位置计数器复位，QEIM<2:0> = 101。

当控制位 QEIM<2:0> = 110 或 111 时，将选择 x4 测量模式，此时 QEI 逻辑将通过 A 相和 B 相输入信号的每个边沿来确定位置计数器递增速率。A 相和 B 相输入信号的每个边沿都会导致位置计数器递增或递减。

在 x4 测量模式下，可用两种方式复位位置计数器：

- 通过检测到索引脉冲将位置计数器复位，QEIM<2:0> = 110。
- 通过计数值与 MAXxCNT 匹配将位置计数器复位，QEIM<2:0> = 111。

x4 测量模式为确定电机位置提供了更高分辨率的测量数据（更多的位置计数）。

## 16.4 可编程数字噪声滤波器

模块的数字噪声滤波器部分负责抑制输入捕捉或正交信号中的噪声。施密特触发器输入和三时钟周期延时滤波器共同用来抑制低电平噪声和瞬时噪声尖峰。这些噪声干扰经常出现在易受噪声干扰的应用中，例如电机系统。

该滤波器可以确保在 3 个连续的时钟周期内都获得同一个稳定值之后，才允许经过滤波的输出信号发生变化。

对于 QEAx、QEBx 和 INDXx 引脚，数字滤波器的时钟分频由 QECK<2:0> (DFLTxCN<6:4>) 位设定，该频率源自基本指令周期 Tcy。

要使能通道 QEAx、QEBx 和 INDXx 的滤波器输出，必须将 QEOUT 位置 1。在 POR 时，所有通道的滤波器网络将被禁止。

## 16.5 备用 16 位定时器 / 计数器

当 QEI 模块未配置为 QEI 模式，即 QEIM<2:0> = 001 时，可以将模块配置为简单的 16 位定时器 / 计数器。辅助定时器的设置和控制通过 QEIXCON SFR 寄存器实现。此定时器功能与 Timer1 相同。QEAx 引脚被用作定时器时钟输入。

当配置为定时器时，POSxCNT 寄存器作为定时器计数寄存器，而 MAXxCNT 寄存器作为周期寄存器。当发生定时器 / 周期寄存器匹配时，QEI 中断标志将被置 1。

此定时器和通用定时器的唯一区别在于，此定时器增加了外部递增 / 递减输入选择功能。当 UPDN 引脚为高电平时，定时器将进行递增计数。当 UPDN 引脚为低电平时，定时器将进行递减计数。

**注：** 工作模式的改变（例如，从 QEI 改变到定时器模式或反之）将不会影响定时器 / 位置计数寄存器的内容。

UPDN 控制 / 状态位 (QEIXCON<11>) 可用来选择定时器寄存器的计数方向状态。当 UPDN = 1 时，定时器将进行递增计数。当 UPDN = 0 时，定时器将进行递减计数。

此外，控制位 UPDN\_SRC (QEIXCON<0>) 确定定时器计数方向状态是基于写入 UPDN 控制 / 状态位 (QEIXCON<11>) 的逻辑状态还是 QEBx 引脚的状态：

- 当 UPDN\_SRC = 1 时，定时器计数方向由 QEBx 引脚控制。
- 当 UPDN\_SRC = 0 时，定时器计数方向由 UPDN 位控制。

**注：** 此备用定时器不支持外部异步计数器工作模式。如果程序使用外部时钟源，时钟将自动与内部指令周期同步。

## 16.6 CPU 休眠模式下的 QEI 模块操作

在 CPU 休眠模式期间，QEI 模块将按如下操作：

- QEI 模块暂停工作。
- 由于禁止了内部时钟，所以定时器不工作。

## 16.7 CPU 空闲模式下的 QEI 模块操作

由于 QEI 模块可以用作正交编码器接口或 16 位定时器，本节将描述这两种模式下的 QEI 模块操作。

### 16.7.1 CPU 空闲模式下的 QEI 操作

当 CPU 处于空闲模式时，如果 QEISIDL 位 (QEIXCON<13>) = 0，则 QEI 模块将工作。在执行 POR 时，该位默认为逻辑 0。要在 CPU 空闲模式下暂停 QEI 模块，应将 QEISIDL 设置为 1。

### 16.7.2 CPU 空闲模式下的定时器操作

当 CPU 处于空闲模式且 QEI 模块配置为 16 位定时器模式时，如果 QEISIDL 位 (QEIxCON<13>) = 0，则 16 位定时器将工作。在执行 POR 时，该位默认为逻辑 0。要在 CPU 空闲模式下暂停定时器模块，应将 QEISIDL 设置为 1。

如果 QEISIDL 位被清零，定时器将正常工作，就像 CPU 没有进入空闲模式一样。

### 16.8 正交编码器接口中断

正交编码器接口能够在发生以下事件时产生中断：

- 16 位递增 / 递减位置计数器发生计满返回 / 下溢
- 检测到合格的索引脉冲
- CNTERR 位置 1
- 定时器周期匹配事件 (溢出 / 下溢)
- 门控累加事件

在发生以上任一事件时，IFS3 和 IFS4 寄存器中的 QEI 中断标志位 QEIXIF 将被置 1。QEIXIF 位必须用软件清零。

通过 IEC3 和 IEC4 寄存器中对应的中断允许位 QEIXIE 可以允许中断。

### 16.9 控制和状态寄存器

QEI 模块有 4 个用户可访问的寄存器，可通过字节或字模式进行访问：

- 控制 / 状态寄存器 (QEIXCON) —— 允许控制 QEI 操作和用来指示模块状态的状态标志。
- 数字滤波器控制寄存器 (DFLTxCON) —— 允许控制数字输入滤波器操作。
- 位置计数寄存器 (POSxCNT) —— 允许读写 16 位位置计数器。
- 最大计数寄存器 (MAXxCNT) —— 保存在某些操作中用来与 POSxCNT 计数器进行比较的值。

**注：** POSxCNT 寄存器允许字节访问。但是，在字节模式下读寄存器可能导致在后续读取时值被部分更新。或者使用字模式读 / 写，或者确保在字节操作期间计数器不计数。

寄存器 16-1: QEIxCON: QEIx 控制寄存器 (x = 1 或 2)

R/W-0	U-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
CNTERR	—	QEISIDL	INDEX	UPDN	QEIM<2:0>		
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
SWPAB	PCDOUT	TQGATE	TQCKPS<1:0>		POSRES	TQCS	UPDN_SRC
bit 7				bit 0			

图注:

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

- bit 15            **CNTERR:** 计数错误状态标志位 <sup>(1)</sup>  
 1 = 发生了位置计数错误  
 0 = 未发生位置计数错误
- bit 14            **未实现:** 读为 0
- bit 13            **QEISIDL:** 空闲模式停止位  
 1 = 当器件进入空闲模式时, 模块停止工作  
 0 = 在空闲模式下模块继续工作
- bit 12            **INDEX:** 索引引脚状态位 (只读)  
 1 = 索引引脚为高电平  
 0 = 索引引脚为低电平
- bit 11            **UPDN:** 位置计数器方向状态位 <sup>(2)</sup>  
 1 = 位置计数器方向为正向 (+)  
 0 = 位置计数器方向为反向 (-)
- bit 10-8         **QEIM<2:0>:** 正交编码器接口模式选择位  
 111 = 使能正交编码器接口 (x4 模式), 匹配时复位位置计数器 (MAXxCNT)  
 110 = 使能正交编码器接口 (x4 模式), 索引脉冲复位位置计数器  
 101 = 使能正交编码器接口 (x2 模式), 匹配时复位位置计数器 (MAXxCNT)  
 100 = 使能正交编码器接口 (x2 模式), 索引脉冲复位位置计数器  
 011 = 未使用 (模块被禁止)  
 010 = 未使用 (模块被禁止)  
 001 = 启动 16 位定时器  
 000 = 正交编码器接口 / 定时器关闭
- bit 7             **SWPAB:** A 相和 B 相输入交换选择位  
 1 = A 相和 B 相输入交换  
 0 = A 相和 B 相输入不交换
- bit 6             **PCDOUT:** 位置计数器方向状态输出使能位  
 1 = 使能位置计数器方向状态输出 (QEI 逻辑控制 I/O 引脚的状态)  
 0 = 禁止位置计数器方向状态输出 (正常 I/O 引脚操作)

- 注 1: 仅当 QEIM<2:0> = 110 或 100 时, 可以使用 CNTERR 标志。
- 2: 当 QEIM<2:0> = 1xx 时为只读位。当 QEIM<2:0> = 001 时为可读写位。
- 3: 预分频器仅用于 16 位定时器模式。
- 4: 当 QEIM<2:0> = 100 或 110 时, 才能使用该位。
- 5: 当配置为 QEI 模式时, 该控制位为“无关位”。

**寄存器 16-1: QEIxCON: QEIx 控制寄存器 (x = 1 或 2) (续)**

bit 5	<b>TQGATE:</b> 定时器门控时间累加使能位 1 = 使能定时器门控时间累加 0 = 禁止定时器门控时间累加
bit 4-3	<b>TQCKPS&lt;1:0&gt;:</b> 定时器输入时钟预分频比选择位 <sup>(3)</sup> 11 = 1:256 预分频比 10 = 1:64 预分频比 01 = 1:8 预分频比 00 = 1:1 预分频比
bit 2	<b>POSRES:</b> 位置计数器复位使能位 <sup>(4)</sup> 1 = 索引脉冲复位位置计数器 0 = 索引脉冲不复位位置计数器
bit 1	<b>TQCS:</b> 定时器时钟源选择位 1 = 来自 QEAx 引脚的外部时钟 (上升沿触发计数) 0 = 内部时钟 (Tcy)
bit 0	<b>UPDN_SRC:</b> 位置计数器方向选择控制位 <sup>(5)</sup> 1 = QEBx 引脚状态定义位置计数器方向 0 = 控制 / 状态位 UPDN (QEIxCON<11>) 定义定时器计数器 (POSxCNT) 的方向

- 注 1:** 仅当 QEIM<2:0> = 110 或 100 时, 可以使用 CNTERR 标志。
- 2:** 当 QEIM<2:0> = 1xx 时为只读位。当 QEIM<2:0> = 001 时为可读写位。
- 3:** 预分频器仅用于 16 位定时器模式。
- 4:** 当 QEIM<2:0> = 100 或 110 时, 才能使用该位。
- 5:** 当配置为 QEI 模式时, 该控制位为“无关位”。



寄存器 16-2: DFLT<sub>x</sub>CON: 数字滤波器控制寄存器

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	IMV<2:0>		CEID
bit 15							bit 8
R/W-0	R/W-0		U-0	U-0	U-0	U-0	
QEOUT	QECK<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 15-11      未实现: 读为 0

bit 10-9      **IMV<1:0>**: 索引匹配值位——当 POS<sub>x</sub>CNT 寄存器将被复位时, 这些位允许用户应用程序在索引脉冲期间指定 QE<sub>Ax</sub> 和 QE<sub>Bx</sub> 输入引脚的状态。

在 x4 正交计数模式下:

IMV1 = 索引脉冲匹配所要求的 B 相输入信号的状态

IMV0 = 索引脉冲匹配所要求的 A 相输入信号的状态

在 x2 正交计数模式下:

IMV1 = 为索引状态匹配选择的相输入信号 (0 = A 相, 1 = B 相)

IMV0 = 索引脉冲匹配要求的所选相输入信号的状态

bit 8          **CEID**: 计数错误中断禁止位  
 1 = 禁止计数错误引起的中断  
 0 = 允许计数错误引起的中断

bit 7          **QEOUT**: QE<sub>Ax</sub>/QE<sub>Bx</sub>/IND<sub>x</sub> 引脚数字滤波器输出使能位  
 1 = 使能数字滤波器输出  
 0 = 禁止数字滤波器输出 (正常引脚操作)

bit 6-4      **QECK<2:0>**: QE<sub>Ax</sub>/QE<sub>Bx</sub>/IND<sub>x</sub> 数字滤波器时钟分频选择位  
 111 = 1:256 时钟分频  
 110 = 1:128 时钟分频  
 101 = 1:64 时钟分频  
 100 = 1:32 时钟分频  
 011 = 1:16 时钟分频  
 010 = 1:4 时钟分频  
 001 = 1:2 时钟分频  
 000 = 1:1 时钟分频

bit 3-0      未实现: 读为 0

注:

## 17.0 串行外设接口 (SPI)

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参见“*dsPIC33F Family Reference Manual*”的“Section 18. Serial Peripheral Interface (SPI)” (DS70206)，该文档可从 [Microchip 网站 \(www.microchip.com\)](http://www.microchip.com) 下载。

串行外设接口 (Serial Peripheral Interface, SPI) 模块是用于同其他外设或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、显示驱动器和模数转换器等。SPI 模块与 Motorola® 的 SPI 和 SIOP 兼容。

每个 SPI 模块由一个用于将数据移入和移出的 16 位移位寄存器 SPIxSR (其中 x = 1 或 2) 和一个缓冲寄存器 SPIxBUF 组成。控制寄存器 SPIxCON 用来配置模块。另外，还有一个状态寄存器 SPIxSTAT 用来表明各种状态条件。

串行接口由 4 个引脚组成：

- SDIx (串行数据输入)
- SDOx (串行数据输出)
- SCKx (移位时钟输入或输出)
- SSx (低电平有效从选择)

在主模式下工作时，SCK 是时钟输出。在从模式下时，SCK 是时钟输入。

### 17.1 中断

一组 8 或 16 个时钟脉冲将数据位从 SPIxSR 移出到 SDOx 引脚，同时将 SDIx 引脚的数据移入 SPIxSR。当传输完成时产生一个中断，相应的中断标志位 (SPI1IF) 置 1。通过中断允许位 (SPI1IE) 可以禁止该中断。

### 17.2 接收操作

接收操作是双重缓冲的。当接收完字节时，将字节从 SPIxSR 送到 SPIxBUF。

当从 SPIxSR 传输新数据到 SPIxBUF 时，如果接收缓冲区已满，模块会将 SPIROV 位置 1，表明产生了溢出条件。数据从 SPIxSR 向 SPIxBUF 的传输不会完成，新数据将丢失。当 SPIROV 为 1 时，模块将不会对 SCK 电平的跳变进行响应；实际上会禁止模块，直到用户软件读 SPIxBUF 为止。

## 17.3 发送操作

发送写操作同样也是双重缓冲的。用户应用程序写入 SPIxBUF。当主模式或从模式传输完成后，移位寄存器 (SPIxSR) 的内容将被送入接收缓冲区。如果已经向缓冲寄存器写了任何发送数据，发送缓冲区的内容将被送入 SPIxSR。于是，接收到的数据存放在 SPIxBUF 中，而 SPIxSR 中的发送数据已经准备就绪，可进行下一次传输。

**注：** 发送缓冲区 (SPIxTXB) 和接收缓冲区 (SPIxRXB) 都映射到相同的寄存器地址 SPIxBUF。请勿对 SPIxBUF 寄存器执行读 - 修改 - 写操作 (例如针对位的指令)。

## 17.4 SPI 设置：主模式

要将 SPI 模块设置为工作在主模式下：

1. 如果使用中断：
  - a) 将相应 IFSn 寄存器中的 SPIxIF 位清零。
  - b) 将相应 IECn 寄存器中的 SPIxIE 位置 1。
  - c) 通过写相应 IPCn 寄存器中的 SPIxIP 位来设置中断优先级。
2. 将要求的设置写入 SPIxCON 寄存器，且使 MSTEN (SPIxCON1<5>) = 1。
3. 将 SPIROV 位 (SPIxSTAT<6>) 清零。
4. 通过将 SPIEN 位 (SPIxSTAT<15>) 置 1 使能 SPI 工作。
5. 将待发送数据写入 SPIxBUF 寄存器。发送 (和接收) 在数据写入 SPIxBUF 寄存器时立即开始。

### 17.5 SPI 设置：从模式

要将 SPI 模块设置为工作在从模式下：

1. 将 SPIxBUF 寄存器清零。
2. 如果使用中断：
  - a) 将相应 IFSn 寄存器中的 SPIxIF 位清零。
  - b) 将相应 IECn 寄存器中的 SPIxIE 位置 1。
  - c) 通过写相应 IPCn 寄存器中的 SPIxIP 位来设置中断优先级。
3. 将要求的设置写入 SPIxCON1 和 SPIxCON2 寄存器，且使 MSTEN (SPIxCON1<5>) = 0。
4. 将 SMP 位清零。
5. 如果 CKE 位置 1，则必须将 SSx 位 (SPIxCON1<7>) 置 1 来使能 SSx 引脚。
6. 将 SPIROV 位 (SPIxSTAT<6>) 清零。
7. 通过将 SPIEN 位 (SPIxSTAT<15>) 置 1 使能 SPI 工作。

SPI 模块会产生一个中断表明字节或字传输完成，还会为所有的 SPI 错误条件产生单独的中断。

图 17-1: SPI 模块框图

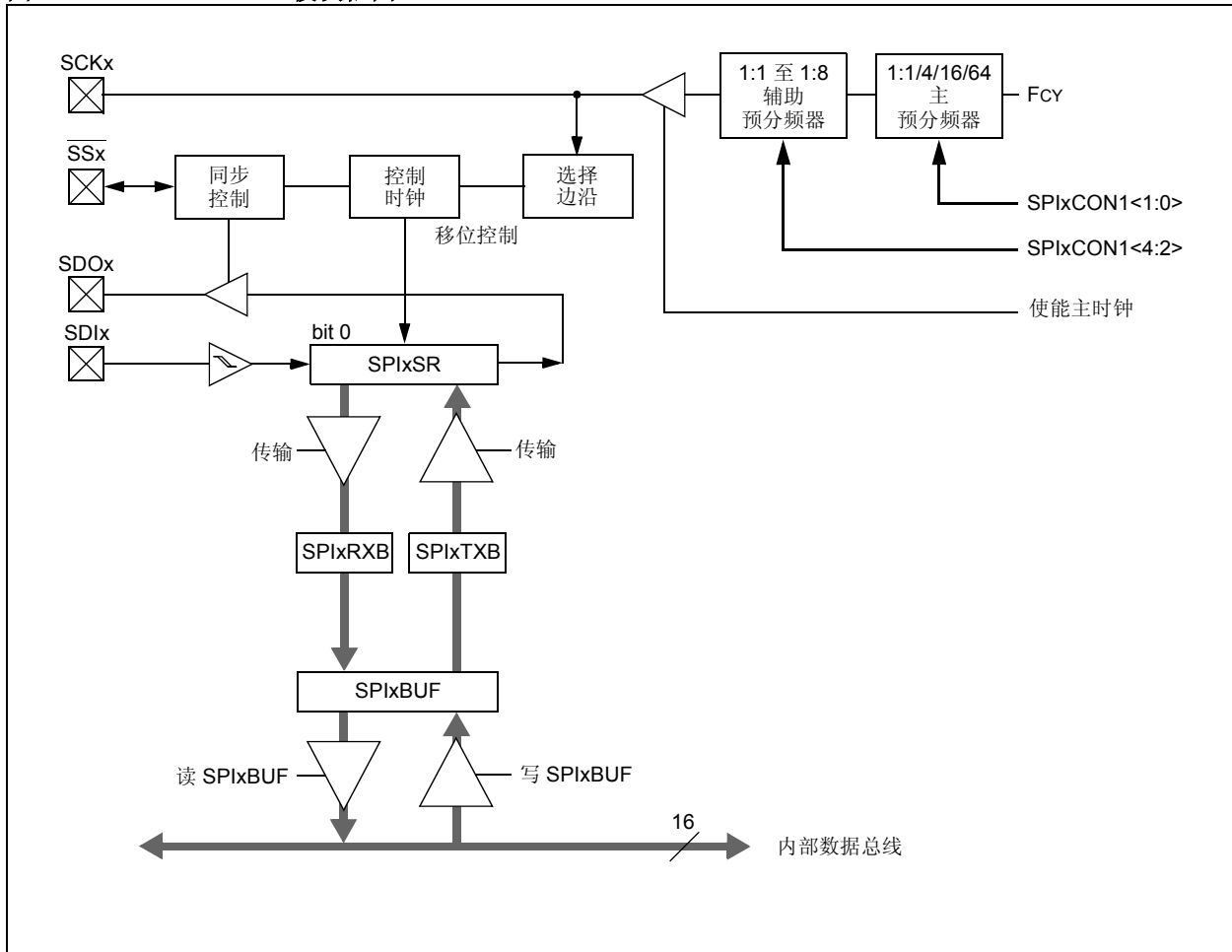


图 17-2: SPI 主 / 从连接

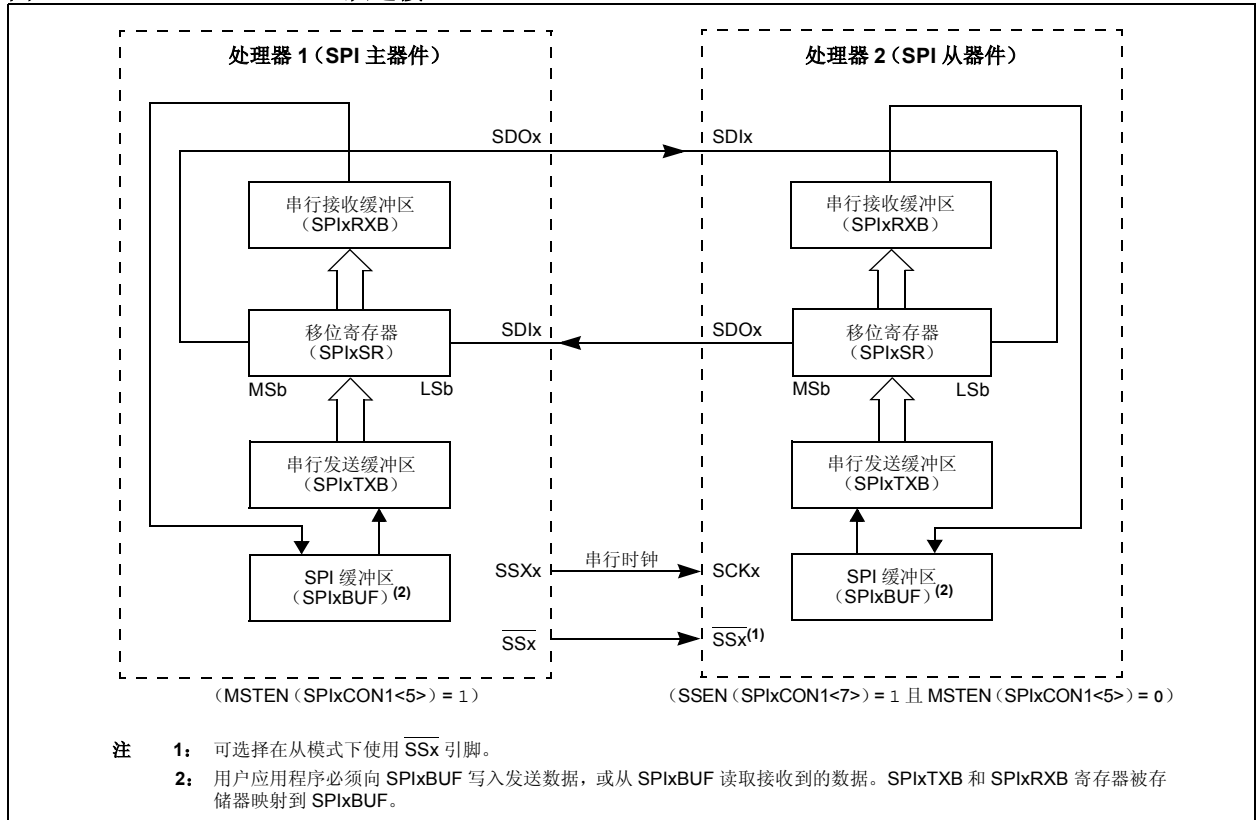


图 17-3: SPI 主 - 帧主模式连接图

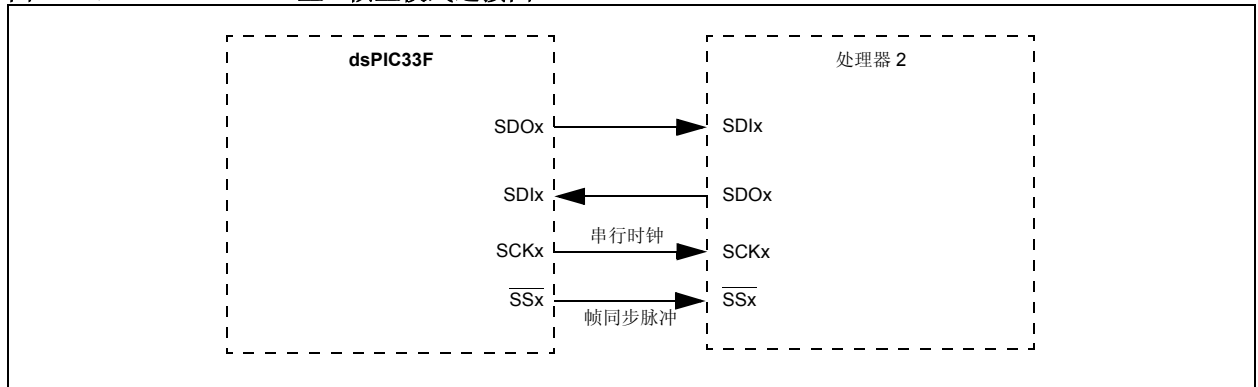


图 17-4: SPI 主 - 帧从模式连接图

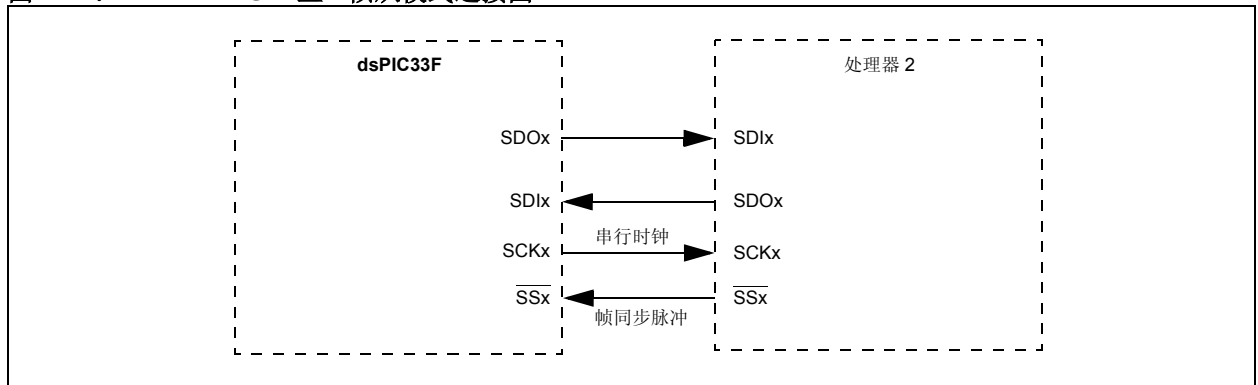


图 17-5: SPI 从-帧主模式连接图

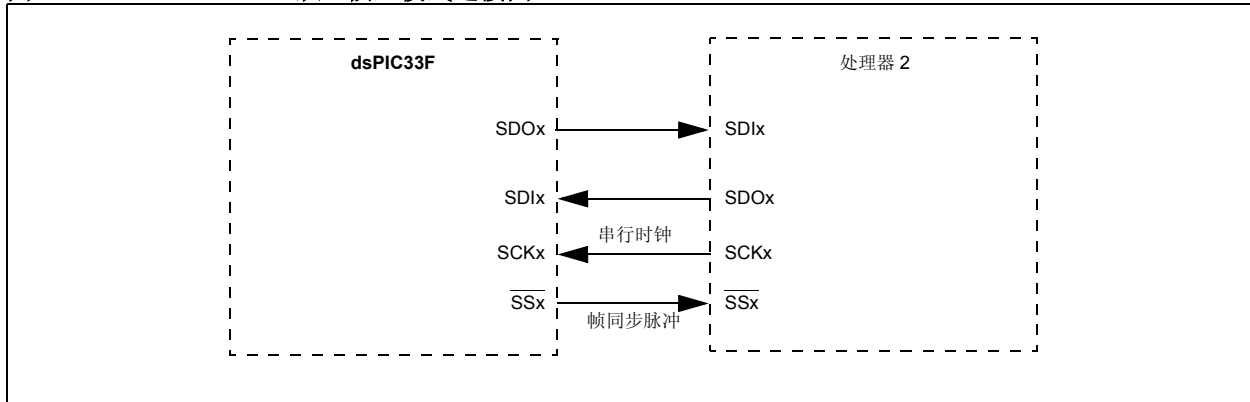
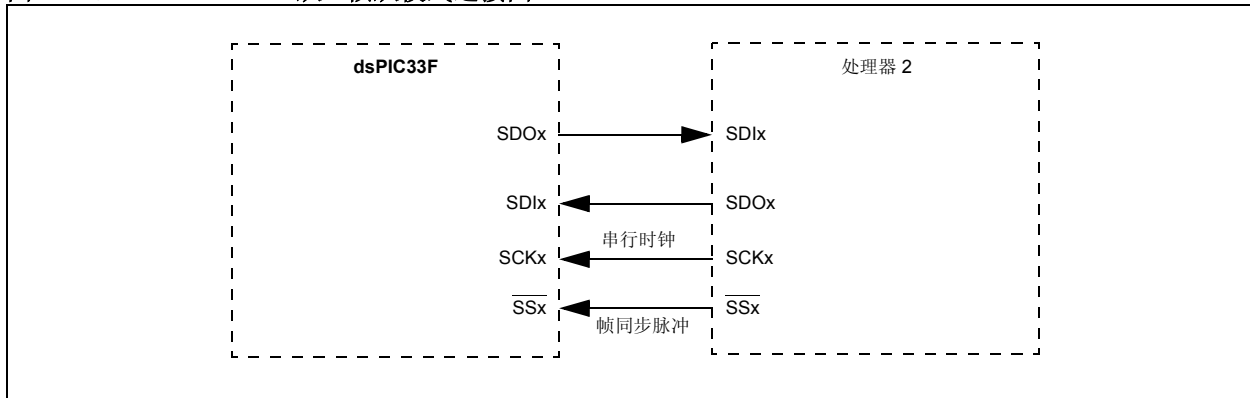


图 17-6: SPI 从-帧从模式连接图



公式 17-1: 器件工作频率和 SPI 时钟速度之间的关系

$$F_{SCK} = \frac{F_{CY}}{\text{主预分频比} \cdot \text{辅助预分频比}}$$

表 17-1: SCKx 频率示例

F <sub>cy</sub> = 40 MHz		辅助预分频比设置				
		1:1	2:1	4:1	6:1	8:1
主预分频比设置	1:1	无效	无效	10000	6666.67	5000
	4:1	10000	5000	2500	1666.67	1250
	16:1	2500	1250	625	416.67	312.50
	64:1	625	312.5	156.25	104.17	78.125
F <sub>cy</sub> = 5 MHz						
主预分频比设置	1:1	5000	2500	1250	833	625
	4:1	1250	625	313	208	156
	16:1	313	156	78	52	39
	64:1	78	39	20	13	10

注: 表中 SCKx 频率的单位为 kHz。

寄存器 17-1: SPIxSTAT: SPIx 状态和控制寄存器

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
SPIEN	—	SPISIDL	—	—	—	—	—
bit 15						bit 8	
U-0	R/C-0	U-0	U-0	U-0	U-0	R-0	R-0
—	SPIROV	—	—	—	—	SPITBF	SPIRBF
bit 7						bit 0	

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

- bit 15      **SPIEN:** SPIx 使能位  
 1 = 使能模块并将 SCKx、SDOx、SDIx 和  $\overline{SSx}$  配置为串口引脚  
 0 = 禁止模块
- bit 14      **未实现:** 读为 0
- bit 13      **SPISIDL:** 空闲模式停止位  
 1 = 当器件进入空闲模式时, 模块停止工作  
 0 = 在空闲模式下模块继续工作
- bit 12-7   **未实现:** 读为 0
- bit 6        **SPIROV:** 接收溢出标志位  
 1 = 一个新字节 / 字已完全接收并丢弃。用户软件没有读先前保存在 SPIxBUF 寄存器中的数据。  
 0 = 未发生溢出。
- bit 5-2     **未实现:** 读为 0
- bit 1        **SPITBF:** SPIx 发送缓冲区满状态位  
 1 = 发送尚未开始, SPIxTXB 为满  
 0 = 发送已开始, SPIxTXB 为空  
 当 CPU 写 SPIxBUF 存储单元并装入 SPIxTXB 时, 该位由硬件自动置 1  
 当 SPIx 模块将数据从 SPIxTXB 传输到 SPIxSR 时, 该位由硬件自动清零
- bit 0        **SPIRBF:** SPIx 接收缓冲区满状态位  
 1 = 接收完成, SPIxRXB 为满  
 0 = 接收未完成, SPIxRXB 为空  
 当 SPIx 将数据从 SPIxSR 传输到 SPIxRXB 时, 该位由硬件自动置 1  
 当内核通过读 SPIxBUF 存储单元读 SPIxRXB 时, 该位由硬件自动清零

寄存器 17-2: SPIxCON1: SPIx 控制寄存器 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DISSCK	DISSDO	MODE16	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	CKP	MSTEN	SPRE<2:0>			PPRE<1:0>	
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        **DISSCK:** 禁止 SCKx 引脚位 (仅限 SPI 主模式)  
 1 = 禁止内部 SPI 时钟, 引脚用作 I/O  
 0 = 使能内部 SPI 时钟
- bit 11        **DISSDO:** 禁止 SDOx 引脚位  
 1 = 模块不使用 SDOx 引脚; 引脚用作 I/O  
 0 = SDOx 引脚由模块控制
- bit 10        **MODE16:** 字 / 字节通信选择位  
 1 = 采用字宽 (16 位) 通信  
 0 = 采用字节宽 (8 位) 通信
- bit 9         **SMP:** SPIx 数据输入采样阶段位  
主模式:  
 1 = 在数据输出时间的末端采样输入数据  
 0 = 在数据输出时间的中间采样输入数据  
从模式:  
 当在从模式下使用 SPIx 时, 必须将 SMP 清零。
- bit 8         **CKE:** SPIx 时钟边沿选择位 <sup>(1)</sup>  
 1 = 串行输出数据在时钟从工作状态转变为空闲状态时变化 (见 bit 6)  
 0 = 串行输出数据在时钟从空闲状态转变为工作状态时变化 (见 bit 6)
- bit 7         **SSEN:** 从选择使能位 (从模式)  
 1 =  $\overline{SSx}$  引脚用于从模式  
 0 =  $\overline{SSx}$  引脚不被模块使用。引脚由端口功能控制。
- bit 6         **CKP:** 时钟极性选择位  
 1 = 空闲状态时钟信号为高电平; 工作状态为低电平  
 0 = 空闲状态时钟信号为低电平; 工作状态为高电平
- bit 5         **MSTEN:** 主模式使能位  
 1 = 主模式  
 0 = 从模式

注 1: 在帧 SPI 模式下不使用 CKE 位。在帧 SPI 模式 (FRMEN = 1) 下, 将该位编程为 0。



**寄存器 17-2: SPIxCON1: SPIx 控制寄存器 1 (续)**

bit 4-2      **SPRE<2:0>**: 辅助预分频比位 (主模式)  
111 = 辅助预分频比 1:1  
110 = 辅助预分频比 2:1  
.  
.  
000 = 辅助预分频比 8:1

bit 1-0      **PPRE<1:0>**: 主预分频比位 (主模式)  
11 = 主预分频比 1:1  
10 = 主预分频比 4:1  
01 = 主预分频比 16:1  
00 = 主预分频比 64:1

**注 1:** 在帧 SPI 模式下不使用 CKE 位。在帧 SPI 模式 (FRMEN = 1) 下, 将该位编程为 0。

**寄存器 17-3: SPIxCON2: SPIx 控制寄存器 2**

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	FRMPOL	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
—	—	—	—	—	—	FRMDLY	—
bit 7							bit 0

**图注:**

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

- bit 15        **FRMEN:** 帧 SPIx 支持位  
 1 = 使能帧 SPIx 支持 (SSx 引脚用作帧同步脉冲输入 / 输出)  
 0 = 禁止帧 SPIx 支持
- bit 14        **SPIFSD:** 帧同步脉冲方向控制位  
 1 = 帧同步脉冲输入 (从器件)  
 0 = 帧同步脉冲输出 (主器件)
- bit 13        **FRMPOL:** 帧同步脉冲极性位  
 1 = 帧同步脉冲为高电平有效  
 0 = 帧同步脉冲为低电平有效
- bit 12-2      **未实现:** 读为 0
- bit 1         **FRMDLY:** 帧同步脉冲边沿选择位  
 1 = 帧同步脉冲与第一个位时钟一致  
 0 = 帧同步脉冲比第一个位时钟提前
- bit 0         **未实现:** 禁止用户应用程序将该位设置为 1。

## 18.0 I<sup>2</sup>C™

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“*dsPIC33F Family Reference Manual*”的“Section 19. Inter-Integrated Circuit (I<sup>2</sup>C™)” (DS70195)，该文档可从 Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 下载。

16 位接口的 I<sup>2</sup>C 模块为 I<sup>2</sup>C 串行通信标准的从模式和多主器件模式提供完全的硬件支持。

I<sup>2</sup>C 模块有一个双引脚接口：

- SCLx 引脚是时钟线。
- SDAx 引脚是数据线。

I<sup>2</sup>C 模块提供以下主要特性：

- I<sup>2</sup>C 接口支持主 / 从工作模式。
- I<sup>2</sup>C 从模式支持 7 位和 10 位地址。
- I<sup>2</sup>C 主模式支持 7 位和 10 位地址。
- I<sup>2</sup>C 端口允许主器件和从器件之间的双向传输。
- I<sup>2</sup>C 端口的串行时钟同步可以用作握手机制来暂停和继续串行传输 (SCLREL 控制)。
- I<sup>2</sup>C 支持多主器件工作；检测总线冲突并相应地进行仲裁。

### 18.1 工作模式

硬件完全实现了 I<sup>2</sup>C 标准模式和快速模式规范的所有主从功能，以及 7 位和 10 位寻址。

在 I<sup>2</sup>C 总线上，I<sup>2</sup>C 模块既可作为主器件工作，也可作为从器件工作。

支持 I<sup>2</sup>C 模块工作在以下模式：

- 带 7 位地址的 I<sup>2</sup>C 从模式
- 带 10 位地址的 I<sup>2</sup>C 从模式
- 带 7 位或 10 位地址的 I<sup>2</sup>C 主模式

关于每种模式的通信时序的详细信息，请参见“*dsPIC33F Family Reference Manual*”。请访问 Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 获得最新的“*dsPIC33F Family Reference Manual*”章节。

## 18.2 I<sup>2</sup>C 寄存器

I2CxCON 和 I2CxSTAT 分别是控制寄存器和状态寄存器。I2CxCON 寄存器是可读写的。I2CxSTAT 的低 6 位是只读的，它的其余位是可读写的：

- I2CxRSR 是用于将数据从内部移位到模块的移位寄存器，用户应用程序无权访问它。
- I2CxRCV 是接收缓冲寄存器，是一个用来写入和读出数据字节的寄存器。
- I2CxTRN 是发送寄存器，发送操作期间，字节将会写入 I2CxTRN。
- I2CxADD 寄存器存放从地址。
- 状态位 ADD10 表明是否为 10 位地址模式。
- I2CxBRG 用来保存波特率发生器 (Baud Rate Generator, BRG) 的重载值。

在进行接收时，I2CxRSR 和 I2CxRCV 一起形成双重缓冲接收器。当 I2CxRSR 接收到一个完整的字节后，字节被送至 I2CxRCV 并产生中断脉冲。

### 18.3 I<sup>2</sup>C 中断

I<sup>2</sup>C 模块产生两个中断：

- MI2CxIF (I<sup>2</sup>C 主事件中断标志)
- SI2CxIF (I<sup>2</sup>C 从事件中断标志)

在所有 I<sup>2</sup>C 错误条件下都会产生一个单独的中断。

### 18.4 波特率发生器

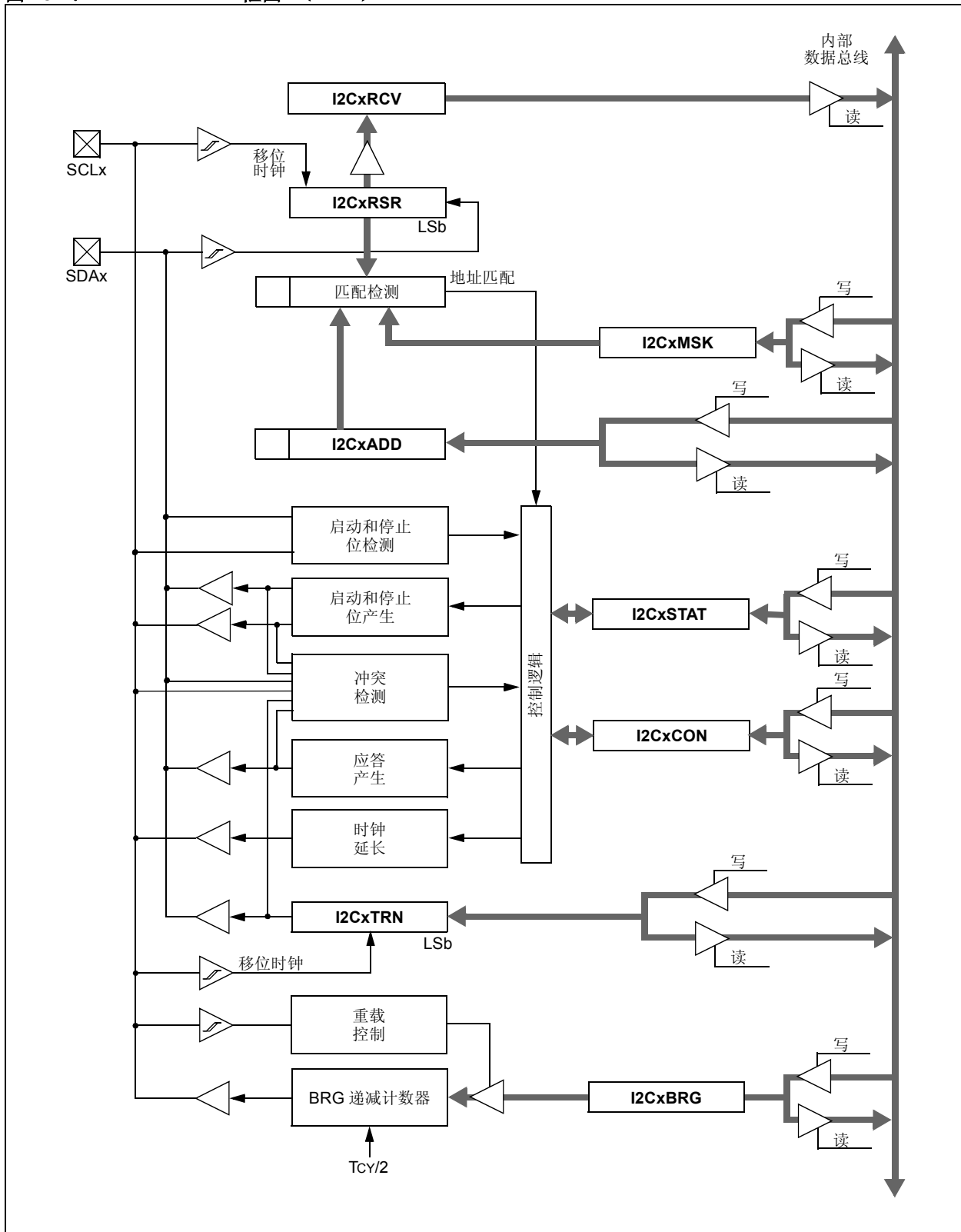
在 I<sup>2</sup>C 主模式下，波特率发生器 (BRG) 的重载值位于 I2CxBRG 寄存器中。BRG 装入该值后，BRG 将递减计数至 0 并停止，直到再次发生重载。如果发生时钟仲裁，例如 SCLx 引脚采样为高电平时，将重载 BRG。

根据 I<sup>2</sup>C 标准，F<sub>SCL</sub> 可以是 100 kHz 或 400 kHz。然而，用户应用程序可指定最高为 1 MHz 的任何波特率。I2CxBRG 的值为 0 或 1 是非法的。

**公式 18-1: 串行时钟速率**

$$I2CxBRG = \left( \frac{FCY}{F_{SCL}} - \frac{FCY}{10,000,000} \right) - 1$$

图 18-1: I<sup>2</sup>C™ 框图 (x = 1)



## 18.5 I<sup>2</sup>C 模块地址

10 位 I2CxADD 寄存器包含从模式地址。

如果 A10M 位 (I2CxCON<10>) 为 0, 则模块把地址解释为 7 位地址。当接收到地址时, 将收到的地址与 I2CxADD 寄存器中的低 7 位进行比较。

如果 A10M 位为 1, 则将地址视为 10 位地址。当接收到地址时, 将收到的地址与二进制值“11110 A9 A8” (其中 A9 和 A8 是 I2CxADD 的高两位) 进行比较。如果值匹配, 下一个地址将与 I2CxADD 的低 8 位进行比较, 这是 10 位寻址协议规定的。

表 18-1: 支持的 7 位 I<sup>2</sup>C™ 从地址

地址	说明
0x00	广播呼叫地址或启动字节
0x01-0x03	保留
0x04-0x07	HS 模式主机码
0x08-0x77	有效 7 位地址
0x78-0x7B	有效 10 位地址 (低 7 位)
0x7C-0x7F	保留

## 18.6 从地址掩码

I2CxMSK 寄存器 (寄存器 18-3) 将 7 位和 10 位地址模式下地址中的某些位指定为“无关位”。将 I2CxMSK 的某个位置 1 (= 1), 不论相应的地址位的值是 0 还是 1, 工作在从模式下的模块都会作出响应。例如, 当将 I2CxMSK 设置为 00100000 时, 工作在从模式下的模块将同时检测两个地址 00000000 和 00100000。

要启用地址掩码, 必须通过将 IPMIEN 位 (I2CxCON<11>) 清零来禁止智能外设管理接口 (Intelligent Peripheral Management Interface, IPMI)。

## 18.7 IPMI 支持

控制位 IPMIEN 允许模块支持智能外设管理接口 (IPMI)。当此位置 1 时, 模块将接受所有地址并对所有地址执行操作。

## 18.8 广播呼叫地址支持

广播呼叫地址能寻址所有器件。当使用此地址时, 理论上所有器件都应该以应答信号作出响应。

广播呼叫地址是 I<sup>2</sup>C 协议为特定目的保留的 8 个地址之一。它由全 0 地址位组成, 且 R\_W = 0。

当广播呼叫使能 (GCEN) 位置 1 (I2CxCON<7> = 1) 时, 将识别广播呼叫地址。在响应中断时, 通过读 I2CxRCV 的内容可以检查中断的来源, 从而确定地址是特定于器件的还是广播呼叫地址。

## 18.9 自动时钟延长

在从模式下, 模块可以通过时钟延长来将读缓冲区与写主器件同步。

### 18.9.1 发送时钟延长

在 10 位和 7 位发送模式下, 如果 TBF 位清零 (表明缓冲区为空), 则在第 9 个时钟的下降沿后将 SCLREL 位清零来实现时钟延长。

在从发送模式下, 始终进行时钟延长, 与 STREN 位无关。在允许继续发送之前, 用户 ISR 必须将 SCLREL 位置 1。通过使 SCLx 线保持为低电平, 在主器件启动另一个发送序列之前, 用户应用程序就有时间执行 ISR 并装入 I2CxTRN 的内容。

### 18.9.2 接收时钟延长

在从接收模式下, I2CxCON 寄存器中的 STREN 位用来使能时钟延长。当 STREN 位置 1 时, SCLx 引脚将在每个数据接收序列结束时保持为低电平。

在允许继续接收之前, 用户 ISR 必须将 SCLREL 位置 1。通过使 SCLx 线保持为低电平, 在主器件启动另一个接收序列之前, 用户应用程序就有时间执行 ISR 并读出 I2CxRCV 的内容。这可以防止缓冲区溢出。

## 18.10 软件控制的时钟延长 (STREN = 1)

当 STREN 位为 1 时, 可以用软件清零 SCLREL 位, 从而实现对时钟延长的软件控制。

如果 STREN 位为 0, 软件对 SCLREL 位的写操作将被忽略, 不会对 SCLREL 位产生影响。

### 18.11 斜率控制

对于快速模式（400 kHz），I<sup>2</sup>C 标准要求对 SDAx 和 SCLx 信号进行斜率控制。如果需要的话，用户应用程序可通过控制位 DISSLW 来禁止斜率控制。对于 1 MHz 模式，必须禁止斜率控制。

### 18.12 时钟仲裁

在任何接收、发送或重复启动 / 停止条件期间，当主器件释放了 SCLx 引脚（允许 SCLx 悬空为高电平）时，就会发生时钟仲裁。当允许 SCLx 引脚悬空为高电平时，波特率发生器（BRG）暂停计数，直到 SCLx 引脚被实际采样为高电平为止。当 SCLx 引脚被采样为高电平时，BRG 重新装入 I2CxBRG 的内容并开始计数。这可以保证当外部器件将时钟拉为低电平时，SCLx 高电平时间始终至少为一个 BRG 计满返回计数周期。

### 18.13 多主器件通信、总线冲突和总线仲裁

通过总线仲裁实现了对多主器件模式的支持。当主器件在 SDAx 引脚上输出地址 / 数据位时，如果一个主器件使 SDAx 悬空为高电平从而输出一个 1，而另一个主器件要输出 0，就会发生仲裁。当 SCLx 引脚悬空为高电平时，数据应该是稳定的。如果 SDAx 上预期数据是 1，但从 SDAx 引脚采样到的数据是 0，那么就发生了总线冲突。主器件将 I<sup>2</sup>C 主事件中标志位置 1，并且将 I<sup>2</sup>C 端口的主器件部分复位到空闲状态。

### 18.14 外设引脚选择限制

I<sup>2</sup>C 模块具有有限的外设引脚选择功能。当将 FPOR 配置寄存器中的 ALTI2C 位设置为 1 时，I<sup>2</sup>C 模块使用 SDAx/SCLx 引脚。当 ALTI2C 位为 0 时，I<sup>2</sup>C 模块使用 ASDAx/ASCLx 引脚。

寄存器 18-1: I2CxCON: I2Cx 控制寄存器

R/W-0	U-0	R/W-0	R/W-1 HC	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0 HC	R/W-0 HC	R/W-0 HC	R/W-0 HC	R/W-0 HC
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

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

- bit 15      **I2CEN:** I2Cx 使能位  
1 = 使能 I2Cx 模块, 并将 SDAx 和 SCLx 引脚配置为串口引脚  
0 = 禁止 I2Cx 模块。所有 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:** 空闲模式停止位  
1 = 当器件进入空闲模式时, 模块停止工作  
0 = 在空闲模式下模块继续工作
- bit 12      **SCLREL:** SCLx 释放控制位 (作为 I<sup>2</sup>C 从器件工作时)  
1 = 释放 SCLx 时钟  
0 = 保持 SCLx 时钟为低电平 (时钟延长)  
如果 STREN = 1:  
该位可读写 (即软件可以写入 0 来启动时钟延长或写入 1 来释放时钟)。在从器件发送开始时由硬件清零。在从器件接收结束时由硬件清零。  
如果 STREN = 0:  
该位可读且可被置 1 (即软件只能写入 1 来释放时钟)。在从器件发送开始时由硬件清零。
- bit 11      **IPMIEN:** 智能外设管理接口 (IPMI) 使能位  
1 = 使能 IPMI 模式; 应答所有地址  
0 = 禁止 IPMI 模式
- bit 10      **A10M:** 10 位从器件地址位  
1 = I2CxADD 为 10 位从器件地址  
0 = I2CxADD 为 7 位从器件地址
- bit 9        **DISSLW:** 禁止斜率控制位  
1 = 禁止斜率控制  
0 = 使能斜率控制
- bit 8        **SMEN:** SMBus 输入电平位  
1 = 使能符合 SMBus 规范的 I/O 引脚门限值  
0 = 禁止 SMBus 输入门限值
- bit 7        **GCEN:** 广播呼叫使能位 (作为 I<sup>2</sup>C 从器件工作时)  
1 = 允许在 I2CxRSR 接收到广播呼叫地址时产生中断 (已使能模块接收)  
0 = 禁止广播呼叫地址
- bit 6        **STREN:** SCLx 时钟延长使能位 (作为 I<sup>2</sup>C 从器件工作时)  
与 SCLREL 位配合使用。  
1 = 使能软件或接收时钟延长  
0 = 禁止软件或接收时钟延长

**寄存器 18-1: I2CxCON: I2Cx 控制寄存器 (续)**

- bit 5      **ACKDT:** 应答数据位 (作为 I<sup>2</sup>C 主器件工作时, 适用于主器件接收操作)  
当软件启动应答序列时将发送的值。  
1 = 在应答时发送 NACK  
0 = 在应答时发送 ACK
- bit 4      **ACKEN:** 应答序列使能位 (作为 I<sup>2</sup>C 主器件工作时, 适用于主器件接收操作)  
1 = 在 SDAx 和 SCLx 引脚上发出应答序列, 并发送 ACKDT 数据位。在主器件应答序列结束时由硬件清零。  
0 = 应答序列不在进行中
- bit 3      **RCEN:** 接收使能位 (作为 I<sup>2</sup>C 主器件工作时)  
1 = 使能 I<sup>2</sup>C 接收模式。在主器件接收完数据字节的第 8 位时由硬件清零。  
0 = 接收序列不在进行中
- bit 2      **PEN:** 停止条件使能位 (作为 I<sup>2</sup>C 主器件工作时)  
1 = 在 SDAx 和 SCLx 引脚上发出停止条件。在主器件停止序列结束时由硬件清零。  
0 = 停止条件不在进行中
- bit 1      **RSEN:** 重复启动条件使能位 (作为 I<sup>2</sup>C 主器件工作时)  
1 = 在 SDAx 和 SCLx 引脚上发出重复启动条件。在主器件重复启动序列结束时由硬件清零。  
0 = 重复启动条件不在进行中
- bit 0      **SEN:** 启动条件使能位 (作为 I<sup>2</sup>C 主器件工作时)  
1 = 在 SDAx 和 SCLx 引脚上发出启动条件。在主器件启动序列结束时由硬件清零。  
0 = 启动条件不在进行中



寄存器 18-2: I2CxSTAT: I2Cx 状态寄存器

R-0 HSC	R-0 HSC	U-0	U-0	U-0	R/C-0 HS	R-0 HSC	R-0 HSC
ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
bit 15						bit 8	
R/C-0 HS	R/C-0 HS	R-0 HSC	R/C-0 HSC	R/C-0 HSC	R-0 HSC	R-0 HSC	R-0 HSC
IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF
bit 7						bit 0	

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

- bit 15      **ACKSTAT:** 应答状态位 (作为 I<sup>2</sup>C 主器件工作时, 适用于主器件发送操作)  
1 = 接收到来自从器件的 NACK  
0 = 接收到来自从器件的 ACK  
在从器件应答结束时由硬件置 1 或清零。
- 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  
在主器件发送开始时由硬件置 1。在从器件应答结束时由硬件清零。
- bit 13-11    **未实现:** 读为 0
- bit 10      **BCL:** 主器件总线冲突检测位  
1 = 主器件工作期间检测到了总线冲突  
0 = 未发生冲突  
检测到总线冲突时由硬件置 1。
- bit 9        **GCSTAT:** 广播呼叫状态位  
1 = 接收到广播呼叫地址  
0 = 未接收到广播呼叫地址  
当地址与广播呼叫地址匹配时由硬件置 1。当检测到停止条件时由硬件清零。
- bit 8        **ADD10:** 10 位地址状态位  
1 = 10 位地址匹配  
0 = 10 位地址不匹配  
当与匹配的 10 位地址的第 2 个字节匹配时由硬件置 1。当检测到停止条件时由硬件清零。
- bit 7        **IWCOL:** 写冲突检测位  
1 = 因为 I<sup>2</sup>C 模块忙, 尝试写 I2CxTRN 寄存器失败  
0 = 未发生冲突  
当总线忙时写 I2CxTRN 会使硬件将该位置 1 (由软件清零)。
- bit 6        **I2COV:** 接收溢出标志位  
1 = 当 I2CxRCV 寄存器仍然保存前一个字节时接收到了新字节  
0 = 未溢出  
尝试将数据从 I2CxRSR 传输到 I2CxRCV 时由硬件置 1 (由软件清零)。
- bit 5        **D\_A:** 数据 / 地址位 (作为 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 = 表示上次接收的字节为器件地址  
器件地址匹配时由硬件清零。在作为从器件接收到数据字节时由硬件置 1。
- bit 4        **P:** 停止位  
1 = 表示上一次检测到停止位  
0 = 上一次未检测到停止位  
当检测到启动、重复启动或停止条件时由硬件置 1 或清零。

**寄存器 18-2: I2CxSTAT: I2Cx 状态寄存器 (续)**

- bit 3      **S:** 启动位  
1 = 表示上次检测到启动 (或重复启动) 位  
0 = 上一次未检测到启动位  
当检测到启动、重复启动或停止条件时由硬件置 1 或清零。
- bit 2      **R\_W:** 读 / 写信息位 (作为 I<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——表示数据传输输入到从器件  
接收到 I<sup>2</sup>C 器件地址字节后由硬件置 1 或清零。
- bit 1      **RBF:** 接收缓冲区满状态位  
1 = 接收完成, I2CxRCV 为满  
0 = 接收未完成, I2CxRCV 为空  
用接收到的字节写 I2CxRCV 时由硬件置 1。当软件读 I2CxRCV 时由硬件清零。
- bit 0      **TBF:** 发送缓冲区满状态位  
1 = 发送正在进行中, I2CxTRN 为满  
0 = 发送完成, I2CxTRN 为空  
当软件写 I2CxTRN 时由硬件置 1。数据发送完成时由硬件清零。

寄存器 18-3: I2CxMSK: I2Cx 从模式地址掩码寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	AMSK9	AMSK8
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AMSK7	AMSK6	AMSK5	AMSK4	AMSK3	AMSK2	AMSK1	AMSK0
bit 7						bit 0	

图注:

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

bit 15-10      未实现: 读为 0

bit 9-0        **AMSKx**: 地址中 bit x 的掩码选择位

1 = 使能输入报文的地址中 bit x 的掩码; 在此位置上不需要位匹配

0 = 禁止 bit x 的掩码; 在此位置上需要位匹配

注:

## 19.0 通用异步收发器 (UART)

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 17. UART” (DS70188)，该文档可从 Microchip 网站 (www.microchip.com) 下载。

通用异步收发器 (Universal Asynchronous Receiver Transmitter, UART) 模块是 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件系列提供的串行 I/O 模块之一。UART 是可以与外设 (例如个人电脑、LIN、RS-232 和 RS-485 接口) 通信的全双工异步系统。模块还通过 UxCTS 和 UxRTS 引脚支持硬件流控制选项，其中还包括 IrDA® 编码器和解码器。

UART 模块的主要特性有：

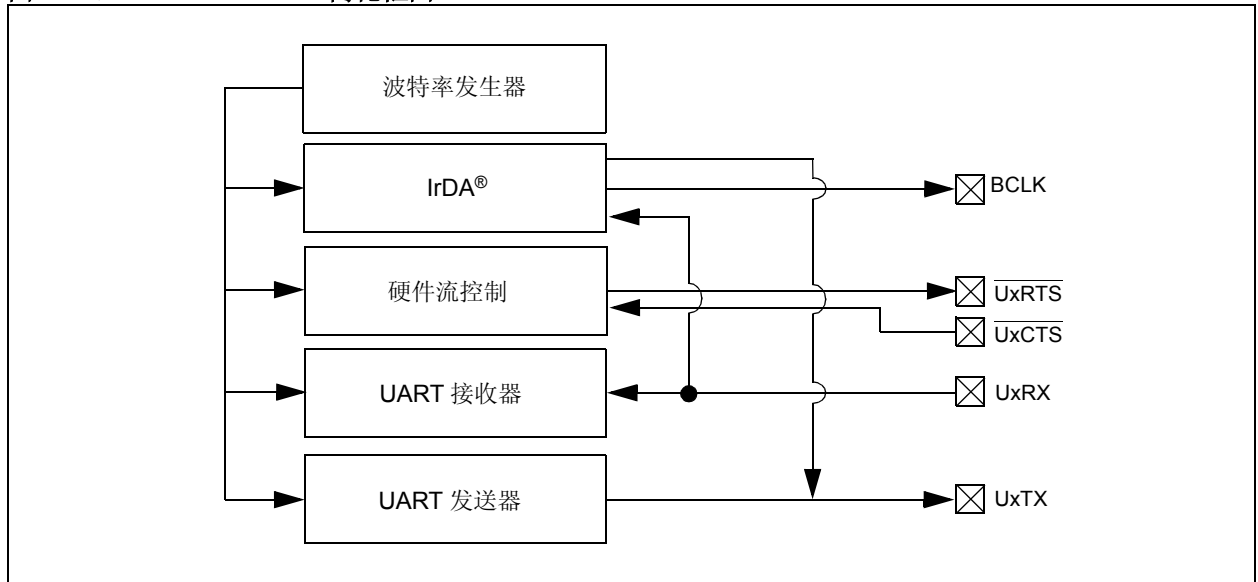
- 通过 UxTX 和 UxRX 引脚进行全双工 8 位或 9 位数据传输

- 偶校验、奇校验或无奇偶校验选项 (对于 8 位数据)
- 一个或两个停止位
- 通过 UxCTS 和 UxRTS 引脚支持硬件流控制选项
- 完全集成的波特率发生器，具有 16 位预分频器
- 当器件工作在 16 MIPS 时，波特率范围从 1 Mbps 到 15 Mbps
- 4 级深度先进先出 (First-In-First-Out, FIFO) 发送数据缓冲区
- 4 级深度 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)
- 发送和接收中断
- 所有 UART 错误条件下可产生单独的中断
- 用于诊断支持的环回模式
- 支持同步和间隔字符
- 支持自动波特率检测
- IrDA 编码器和解码器逻辑
- 用于 IrDA 支持的 16 倍频波特率时钟输出

图 19-1 给出了 UART 模块的简化框图。UART 模块由以下至关重要的硬件组件组成：

- 波特率发生器
- 异步发送器
- 异步接收器

图 19-1: UART 简化框图



### 19.1 UART 波特率发生器 (BRG)

UART模块包含一个专用的16位波特率发生器(BRG)。BRGx寄存器控制一个自由运行的16位定时器的周期。公式 19-1 给出了 BRGH = 0 时计算波特率的公式。

公式 19-1: UART 波特率 (BRGH = 0)

$$\text{波特率} = \frac{F_{CY}}{16 \cdot (BRGx + 1)}$$

$$BRGx = \frac{F_{CY}}{16 \cdot \text{波特率}} - 1$$

注: F<sub>CY</sub> 表示指令周期时钟频率 (F<sub>OSC</sub>/2)。

例 19-1 给出了如下条件下的波特率误差计算:

- F<sub>CY</sub> = 4 MHz
- 目标波特率 = 9600 bps

例 19-1: 波特率误差计算 (BRGH = 0)

目标波特率	=	F <sub>CY</sub> /(16 (BRGx + 1))
求解 BRGx 值:		
BRGx	=	((F <sub>CY</sub> / 目标波特率)/16) - 1
BRGx	=	((4000000/9600)/16) - 1
BRGx	=	25
计算波特率	=	4000000/(16 (25 + 1))
	=	9615
误差	=	( 计算波特率 - 目标波特率 )/目标波特率
	=	(9615 - 9600)/9600
	=	0.16%

最大可能波特率 (BRGH = 0) 是 F<sub>CY</sub>/16 (当 BRGx = 0 时), 最小可能波特率是 F<sub>CY</sub>/(16 • 65536)。

公式 19-2 给出了 BRGH = 1 时计算波特率的公式。

公式 19-2: UART 波特率 (BRGH = 1)

$$\text{波特率} = \frac{F_{CY}}{4 \cdot (BRGx + 1)}$$

$$BRGx = \frac{F_{CY}}{4 \cdot \text{波特率}} - 1$$

注: F<sub>CY</sub> 表示指令周期时钟频率 (F<sub>OSC</sub>/2)。

最大可能波特率 (BRGH = 1) 是 F<sub>CY</sub>/4 (当 BRGx = 0 时), 最小可能波特率是 F<sub>CY</sub>/(4 • 65536)。

向 BRGx 寄存器写入新值会使 BRG 定时器复位 (清零)。这可以确保 BRG 无需等待定时器溢出就可以产生新的波特率。

## 19.2 8 位数据模式下的发送

1. 设置 UART:
  - a) 将适当的值写入数据位、奇偶校验位和停止位。
  - b) 将适当的波特率值写入 BRGx 寄存器。
  - c) 设置发送和接收中断允许位和优先级位。
2. 使能 UART。
3. 将 UTXEN 位置 1（产生发送中断）。
4. 将数据字节写入 UxTXREG 字的低字节。该数据字节值将被立即传输给发送移位寄存器（TSR），且在波特率时钟的下一个上升沿开始移出串行比特流。

或者，当 UTXEN = 0 时，数据字节也可被发送，且随后用户应用程序可将 UTXEN 置 1。由于波特率时钟将从清零状态启动，这将立即开始发送串行比特流。

根据中断控制位 UTXISEL<1:0> 的设置决定何时产生发送中断。

## 19.3 9 位数据模式下的发送

1. 设置 UART（如第 19.2 节“8 位数据模式下的发送”所述）。
2. 使能 UART。
3. 将 UTXEN 位置 1（产生发送中断）。
4. 仅向 UxTXREG 写入一个 16 位的值。
5. 向 UxTXREG 写入一个字可触发 9 位数据向 TSR 的传输。串行比特流将会在波特率时钟的第一个上升沿开始移出。

根据控制位 UTXISEL<1:0> 的设置决定何时产生发送中断。

## 19.4 间隔和同步发送序列

下述序列会发送一个报文帧头，包括一个间隔字符和其后的一个自动波特率同步字节。

1. 将 UART 配置为所需的模式。
2. 将 UTXEN 和 UTXBRK 置 1，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（值被忽略）。
4. 向 UxTXREG 写入 0x55，将同步字符装入发送 FIFO 中。间隔字符发送后，硬件会将 UTXBRK 位复位。

然后开始发送同步字符。

## 19.5 8 位或 9 位数据模式下的接收

1. 设置 UART（如第 19.2 节“8 位数据模式下的发送”所述）。
2. 使能 UART
 

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。
3. 读 OERR 位以确定是否发生了溢出错误。OERR 位必须用软件复位。
4. 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。

## 19.6 使用 UxCTS 和 UxRTS 引脚的流控制

UARTx 允许发送 (UxCTS) 和 UARTx 请求发送 (UxRTS) 是两个与 UART 模块相关、由硬件控制的低电平有效引脚。UxMODE 寄存器中的 UEN<1:0> 位用来配置这两个引脚。

这两个引脚允许 UART 运行在单工模式和流控制模式下。它们用于控制数据终端设备 (Data Terminal Equipment, DTE) 之间的发送和接收。

## 19.7 红外支持

UART 模块提供两种类型的红外 UART 支持：

- IrDA 时钟输出支持外部 IrDA 编码器和解码器（传统模块支持）
- 完全实现了 IrDA 编码器和解码器

### 19.7.1 外部 IrDA 支持——IrDA 时钟输出

为了支持外部 IrDA 编码器和解码器，可将 BCLK 引脚（和 UxRTS 引脚相同）配置为产生 16 倍频的波特率时钟。当 UEN<1:0> = 11 时，如果使能了 UART 模块，BCLK 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。

### 19.7.2 内置 IrDA 编码器和解码器

UART 模块在其内部完全实现了 IrDA 编码器和解码器。内置 IrDA 编码器和解码器的功能可通过 IREN 位 (UxMODE<12>) 来使能。当使能 (IREN = 1) 时，接收引脚 (UxRX) 可作为红外接收器的输入引脚。发送引脚 (UxTX) 可作为红外发送器的输出引脚。

**寄存器 19-1: 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	—	USIDL	IREN <sup>(1)</sup>	RTSMD	—	UEN<1:0>	
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	PDSEL<1:0>		STSEL
bit 7							bit 0

<b>图注:</b>	HC = 由硬件清零
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  
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; 此时 UARTx 的功耗最小
- bit 14      **未实现:** 读为 0
- bit 13      **USIDL:** 空闲模式停止位  
1 = 当器件进入空闲模式时, 模块停止工作  
0 = 在空闲模式下模块继续工作
- bit 12      **IREN:** IrDA 编码器和解码器使能位 <sup>(1)</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 BCLK 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/BCLK 引脚由端口锁存器控制
- bit 7      **WAKE:** 在休眠模式下检测到启动位唤醒使能位  
1 = UARTx 将继续采样 UxRX 引脚; 在出现下降沿时产生中断; 在紧接着的上升沿由硬件清零该位  
0 = 禁止唤醒
- bit 6      **LPBACK:** UARTx 环回模式选择位  
1 = 使能环回模式  
0 = 禁止环回模式
- bit 5      **ABAUD:** 自动波特率使能位  
1 = 使能对下一个字符的波特率测量——需要在接收其他数据前接收同步字段 (55h); 完成时由硬件清零  
0 = 禁止波特率测量或测量已完成
- bit 4      **URXINV:** 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

**注 1:** 此功能只能在 16 倍频 BRG 模式 (BRGH = 0) 下使用。



**寄存器 19-1: UxMODE: UARTx 模式寄存器 (续)**

- bit 3            **BRGH:** 高波特率使能位  
1 = BRG 在每个位周期内产生 4 个时钟信号 (4 倍频波特率时钟, 高速模式)  
0 = BRG 在每个位周期内产生 16 个时钟信号 (16 倍频波特率时钟, 标准模式)
- 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:** 此功能只能在 16 倍频 BRG 模式 (BRGH = 0) 下使用。

寄存器 19-2: UxSTA: UARTx 状态和控制寄存器

R/W-0	R/W-0	R/W-0	U-0	R/W-0 HC	R/W-0	R-0	R-1
UTXISEL1	UTXINV <sup>(1)</sup>	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0
URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

<b>图注:</b>	HC = 由硬件清零
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 **UTXISEL<1:0>**: 发送中断模式选择位
  - 11 = 保留; 不要使用
  - 10 = 当一个字符被传输到发送移位寄存器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; 所有发送操作执行完毕时产生中断
  - 00 = 当一个字符被传输到发送移位寄存器 (这意味着发送缓冲区中至少还有一个字符) 时产生中断
- bit 14 **UTXINV**: IrDA<sup>®</sup> 编码器发送极性翻转位 <sup>(1)</sup>
  - 1 = 使能 IrDA 编码器功能时 UxTX 的空闲状态为 1
  - 0 = 使能 IrDA 编码器功能时 UxTX 的空闲状态为 0
- bit 12 **未实现**: 读为 0
- bit 11 **UTXBRK**: 发送间隔位
  - 1 = 在下次发送时发出同步间隔字符——启动位, 后跟 12 个 0 位, 然后是停止位; 完成时由硬件清零
  - 0 = 禁止或已完成同步间隔字符的发送
- bit 10 **UTXEN**: 发送使能位
  - 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。由端口控制 UxTX 引脚。
- bit 9 **UTXBF**: 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, 至少还可写入一个或多个字符
- bit 8 **TRMT**: 发送移位寄存器空位 (只读)
  - 1 = 发送移位寄存器为空, 同时发送缓冲区为空 (上一次发送已完成)
  - 0 = 发送移位寄存器非空, 发送在进行中或在发送缓冲区中排队
- bit 7-6 **URXISEL<1:0>**: 接收中断模式选择位
  - 11 = 当 UxRSR 传输使接收缓冲区为满时 (即有 4 个数据字符), 中断标志位置 1
  - 10 = 当 UxRSR 传输使接收缓冲区 3/4 满时 (即有 3 个数据字符), 中断标志位置 1
  - 0x = 当接收到一个字符且该字符从 UxRSR 传输给接收缓冲区, 使接收缓冲区有一个或多个字符时, 中断标志位置 1。
- bit 5 **ADDEN**: 地址字符检测位 (接收数据的 bit 8 = 1)
  - 1 = 使能地址检测模式。如果没有选择 9 位模式, 这个控制位将无效。
  - 0 = 禁止地址检测模式

注 1: 仅当 IrDA 编码器使能时 (IREN = 1), 该位的值才影响模块的发送属性。

**寄存器 19-2: UxSTA: UARTx 状态和控制寄存器 (续)**

- bit 4           **RIDLE:** 接收器空闲位 (只读)  
1 = 接收器空闲  
0 = 接收器工作
- bit 3           **PERR:** 奇偶校验错误状态位 (只读)  
1 = 检测到当前字符 (接收 FIFO 顶部的字符) 的奇偶校验错误  
0 = 未检测到奇偶校验错误
- bit 2           **FERR:** 帧错误状态位 (只读)  
1 = 检测到当前字符 (接收 FIFO 顶部的字符) 的帧错误  
0 = 未检测到帧错误
- bit 1           **OERR:** 接收缓冲区溢出错误状态位 (只能读或清零)  
1 = 接收缓冲区已溢出  
0 = 接收缓冲区未溢出。清零原来置 1 的 OERR 位 (1 → 0 的跳变) 将使接收缓冲区和 UxRSR 复位为空状态。
- bit 0           **URXDA:** 接收缓冲区中是否有数据标志位 (只读)  
1 = 接收缓冲区中有数据, 有至少一个或多个字符可被读取  
0 = 接收缓冲区为空

**注 1:** 仅当 IrDA 编码器使能时 (IREN = 1), 该位的值才影响模块的发送属性。

注:

## 20.0 增强型 CAN (ECAN™) 模块

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 21. Enhanced Controller Area Network (ECAN™)” (DS70185)，该文档可从 Microchip 网站 (www.microchip.com) 下载。

### 20.1 概述

增强型控制器局域网 (Enhanced Controller Area Network, ECAN) 模块是一个串行接口，用于同其他 CAN 模块或单片机器件之间进行通信。这个接口 / 协议设计为可以在噪声环境下进行通信。dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件最多具有 2 个 ECAN 模块。

ECAN 模块是一个实现了 BOSCH CAN 规范中定义的 CAN 2.0 A/B 协议的通信控制器。此模块支持 CAN 1.2、CAN 2.0A、CAN 2.0B Passive 和 CAN 2.0B Active 版本的协议。模块实现了完整的 CAN 系统。但是在本数据手册中不讨论 CAN 规范。读者可以参考 BOSCH CAN 规范获取更多的详细信息。

此模块有如下特性：

- 实现 CAN 协议，包括 CAN 1.2、CAN 2.0A 和 CAN 2.0B
- 实现了标准和扩展的数据帧
- 0-8 字节数据宽度
- 最高 1 Mb/s 的可编程比特率
- 远程发送请求的自动响应
- 最多 8 个发送缓冲区，应用程序可以指定优先级和中止功能（每个缓冲区包含最多 8 个字节的数据）
- 最多 32 个接收缓冲区（每个缓冲区最多包含 8 字节数据）
- 最多 16 个完全（标准 / 扩展标识符）接收过滤器
- 3 个完全接收过滤屏蔽寄存器
- 支持 DeviceNet™ 寻址模式
- 集成了低通滤波器的可编程唤醒功能
- 支持自检操作的可编程环回模式
- 对于 CAN 接收器和发送器的所有错误状态，可通过中断功能发出信号
- 可编程时钟源

- 到输入捕捉模块（对于 CAN1 是 IC2）的可编程连接，可用于时间标记和网络同步功能
- 低功耗休眠和空闲模式

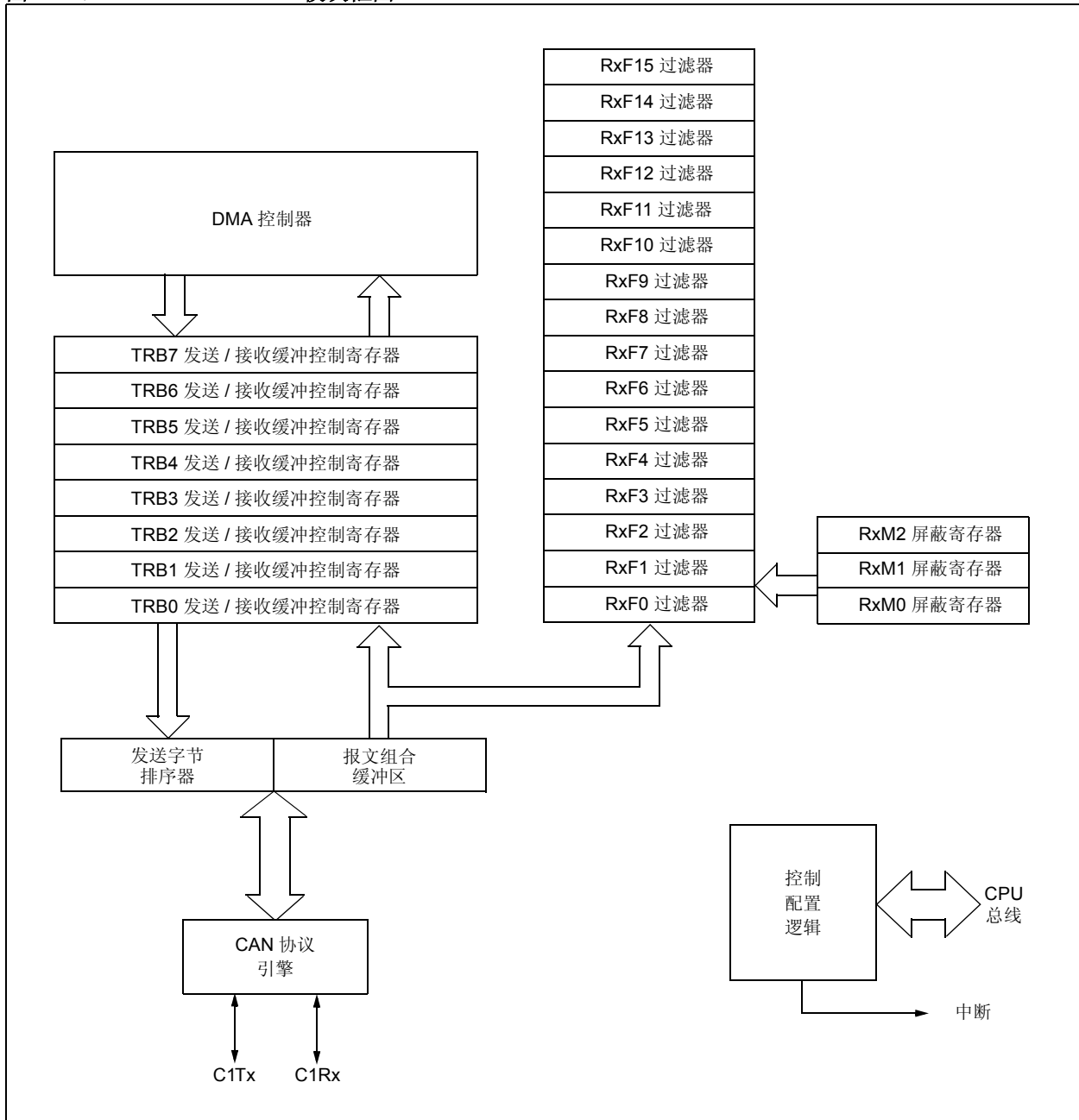
CAN 总线模块由协议引擎与报文缓冲 / 控制组成。CAN 协议引擎处理在 CAN 总线上收发报文的所有功能。通过将报文装载到适当的数据寄存器来发送报文。通过读取相应的寄存器可以检测状态和错误。对于 CAN 总线上检测到的任何报文都会首先检查错误，然后与过滤器进行匹配以判断是否应该将报文接收和存储到接收寄存器之一中。

### 20.2 帧类型

ECAN 模块可发送包括数据报文、用户发出的远程发送请求在内多种类型的帧，以及出于控制目的自动生成的其他帧。支持下列帧类型：

- **标准数据帧：**  
当节点要发送数据时会产生一个标准数据帧。它包含一个 11 位的标准标识符 (Standard Identifier, SID)，而不是 18 位的扩展标识符 (Extended Identifier, EID)。
- **扩展数据帧：**  
扩展数据帧与标准数据帧相似，但还包含一个扩展标识符。
- **远程帧：**  
从目标节点向源节点请求数据是可行的。出于这个目的，目标节点必须发送一个其标识符与所需的数据帧匹配的远程帧。随后相应的数据源节点发送一个数据帧作为对这个远程请求的响应。
- **错误帧：**  
任何节点检测到总线错误将产生错误帧。错误帧包含两个字段：错误标志字段和错误定界符字段。
- **过载帧：**  
节点在两种条件下会产生过载帧。第一，节点在帧间间隔内检测到一个显性位，这是一种非法的条件。第二，由于内部原因节点尚无法开始接收下一条报文。节点最多可产生 2 个连续过载帧来延迟下一条报文的接收。
- **帧间间隔：**  
帧间间隔将前一个帧（无论何种类型）与随后的数据帧或远程帧分隔开来。

图 20-1: ECAN™ 模块框图



## 20.3 工作模式

用户可以选择ECAN模块工作在以下几种工作模式之一。这些模式包括：

- 初始化模式
- 禁止模式
- 正常工作模式
- 监听模式
- 监听所有报文模式
- 环回模式

通过设置 REQOP<2:0> 位 (CiCTRL1<10:8>) 选择所需模式。通过监视 OPMODE<2:0> 位 (CiCTRL1<7:5>) 可以确认进入的模式。在允许进行模式改变之前，模块不会改变工作模式和 OPMODE 位。模式改变通常发生在总线空闲时，即出现了至少 11 个连续隐性位时。

### 20.3.1 初始化模式

在初始化模式下，模块不会进行发送或接收。错误计数器被清零且中断标志位保持不变。用户应用程序可访问在其他模式下不可访问的配置寄存器。模块会防止用户由于编程错误而意外地违反 CAN 协议。当模块在线时，不能修改控制模块配置的所有寄存器。当进行发送的时候，不允许 ECAN 模块进入配置模式。配置模式作为锁来保护以下寄存器：

- 所有的模块控制寄存器
- 波特率和中断配置寄存器
- 总线时序寄存器
- 标识符接收过滤寄存器
- 标识符接收屏蔽寄存器

### 20.3.2 禁止模式

在禁止模式下，模块不会进行发送或接收。由于总线活动，模块能够将 WAKIF 位置 1，但是等待处理的中断将继续等待，且错误计数器的值也保持不变。

如果 REQOP<2:0> 位 (CiCTRL1<10:8>) = 001，模块将进入模块禁止模式。如果模块正处于工作状态，它将等待 CAN 总线上出现 11 个隐性位，表明总线空闲，然后执行模块禁止命令。当 OPMODE<2:0> 位 (CiCTRL1<7:5>) = 001 时，表明模块成功进入了模块禁止模式。当模块处于模块禁止模式时，I/O 引脚恢复普通 I/O 功能。

可将模块编程设置为，当模块或 CPU 处于休眠模式时对 CiRX 输入线应用低通滤波器功能。WAKFIL 位 (CiCFG2<14>) 用于使能和禁止滤波器。

**注：** 通常，如果允许 ECAN 模块在某种工作模式下发送，并且在 ECAN 模块进入该模式后要求立即发送，则模块将在启动发送前等待总线上出现 11 个连续隐性位。如果用户在此 11 个隐性位期间切换到禁止模式，则发送会被中止，同时相应的 TXABT 位置 1，TXREQ 位清零。

### 20.3.3 正常工作模式

当 REQOP<2:0> = 000 时选择正常工作模式。在此模式下，模块激活且 I/O 引脚实现 CAN 总线功能。模块通过 CiTX 和 CiRX 引脚发送和接收 CAN 总线报文。

### 20.3.4 监听模式

如果激活了监听模式，CAN 总线上的模块处于被动状态。发送缓冲区恢复为 I/O 端口功能。接收引脚保持输入功能。对于接收器，不会发出错误标志和应答信号。此状态下错误计数器不工作。监听模式可用于检测 CAN 总线的波特率。要使用这个功能，必须有 2 个以上可以互相通信的节点。

### 20.3.5 监听所有报文模式

此模块能够设置为忽略所有错误并接受所有报文。设置 REQOP<2:0> = 111 可激活监听所有报文模式。在此模式下，报文组合缓冲区中的数据将被复制到接收缓冲区并可通过 CPU 接口读取，直到错误发生。

### 20.3.6 环回模式

如果激活了环回模式，模块将在模块边界处把内部发送信号连接到内部接收信号。发送和接收引脚恢复为 I/O 端口功能。

## 20.4 报文接收

### 20.4.1 接收缓冲区

CAN 总线模块具有最多 32 个接收缓冲区，位于 DMA RAM 中。通过清零 CiTRmnCON 寄存器中的相应的发送 / 接收缓冲区选择位 (TXENn)，可将前 8 个缓冲区配置为接收缓冲区。用户可通过定义 DMABS<2:0> 位 (CiFCTRL<15:13>) 来选择 DMA RAM 中的 CAN 缓冲区的总大小。前 16 个缓冲区可分配为接收过滤器，其余的只能用作 FIFO 缓冲区。

另外有一个缓冲区总是用于监视总线上的输入报文。此缓冲区称为报文组合缓冲区（Message Assembly Buffer, MAB）。

MAB 组合所有报文，但只有符合接收过滤器条件的报文，才被传送到缓冲区中。当收到一个报文时，RBIF 标志（CiINTF<1>）被置 1。然后用户应用程序需要检查 CiVEC 和 / 或 CiRXFUL1 寄存器来判定导致中断产生的过滤器和缓冲区。只有接收到报文时模块才将 RBIF 位置 1。用户完成缓冲区中报文的处理后将该位清零。如果 RBIE 位置 1，当收到报文时将产生中断。

## 20.4.2 FIFO 缓冲区模式

如果过滤器的缓冲区指针的值为 1111，那么 ECAN 模块提供 FIFO 缓冲功能。在此模式下，命中此缓冲区的结果将写入 FIFO 的下一个可用的缓冲单元。

CiFCTRL 寄存器定义 FIFO 的大小。该寄存器中的 FSA<4:0> 位定义 FIFO 缓冲区的起始位置。如果 DMA 使能，则 DMABS<2:0> 位定义 FIFO 的结束位置。因此，支持的 FIFO 最大为 32 个缓冲区大小。

## 20.4.3 报文接收过滤器

报文接收过滤器和屏蔽寄存器用于决定报文组合缓冲区中的报文是否应该被装入某个接收缓冲区中。一旦一个有效报文被接收到报文组合缓冲区（MAB），报文的标识符字段就会与过滤器值进行比较。如果匹配，则此报文将被装入相应的接收缓冲区。每个过滤器都与一个缓冲区指针（FnBP<3:0>）相关联，此指针用于将过滤器和 16 个接收缓冲区中的一个相关联。

接收过滤器检查进入报文中的 IDE 位（CiTRBnSID<0>）来决定如何比较标识符。如果 IDE 位置 1，则报文是标准帧，仅与 EXIDE 位（CiRXFnSID<3>）清零的过滤器进行比较。如果 IDE 位置 0，则报文是扩展帧，仅与 EXIDE 位置 1 的过滤器进行比较。

## 20.4.4 报文接收过滤屏蔽寄存器

屏蔽位主要决定将对哪些位应用过滤器。如果任何屏蔽位被设置为零，无论过滤位为何值该位都会被自动接收。有 3 个与接收缓冲区相关的可编程接收过滤屏蔽寄存器。通过设置相应 CiFMSKSELn 寄存器中的 FnMSK<1:0> 位来选择所需的屏蔽寄存器，可以将这 3 个屏蔽寄存器中的任意一个与每个过滤器关联。

## 20.4.5 接收错误

ECAN 模块可检测到以下接收错误：

- 循环冗余校验（CRC）错误
- 位填充错误
- 无效报文接收错误

这些接收错误不会产生中断。但是，当上述错误之一发生时，接收错误计数器会递增 1。RXWAR 位（CiINTF<9>）指示接收错误计数器已经达到 CPU 警告限制值 96，并产生一个中断。

## 20.4.6 接收中断

接收中断主要分为 3 组，每组包含不同的中断产生条件：

- 接收中断：  
报文已成功接收并装入其中一个接收缓冲区。收到帧结束（End-of-Frame, EOF）字段后立即产生中断。读取 RXnIF 标志可知哪一个接收缓冲区引发了中断。
- 唤醒中断：  
ECAN 模块从禁止模式中唤醒，或器件从休眠模式中唤醒。
- 接收错误中断：  
接收错误中断由 ERRIF 位指示。此位表示有错误条件发生。通过检查 CAN 中断标志寄存器 CiINTF 中的相应位，可以确定错误源。
  - 无效报文接收：  
如果在上一次报文接收时发生了任何类型的错误，IVRIF 位都将指示有错误发生。
  - 接收器溢出：  
RBOVIF 位（CiINTF<2>）指示有溢出条件发生。
  - 接收器警告：  
RXWAR 位指示接收错误计数器（RERRCNT<7:0>）已经达到 CPU 警告限制值 96。
  - 接收器错误被动：  
RXEP 位指示接收错误计数器已经超过了错误被动的限制值 127，且模块已经进入错误被动状态。



## 20.5 报文发送

### 20.5.1 发送缓冲区

ECAN 模块具有最多 8 个发送缓冲区，位于 DMA RAM 中。需要将 CiTRmnCON 寄存器中相应的发送 / 接收缓冲区选择位 (TXENn 或 TXENm) 置 1，把这 8 个缓冲区配置为发送缓冲区。用户可通过定义 DMABS<2:0> 位 (CiFCTRL<15:13>) 来选择 DMA RAM 中的 CAN 缓冲区的总大小。

每个发送缓冲区占用 16 个字节。其中的 8 个字节存放待发送的最多 8 字节的报文。5 个字节保存标准帧或扩展帧标识符和其他报文仲裁信息。末尾的一个字节不使用。

### 20.5.2 发送报文优先级

发送优先级指在各个节点内待发送报文的优先级。发送优先级有 4 级。如果指定的报文缓冲区的 TXnPRI<1:0> 位 (在 CiTRmnCON 中) 被设置为 11，则此缓冲区具有最高优先级。如果指定的报文缓冲区的 TXnPRI<1:0> 位被设置为 10 或 01，则此缓冲区具有中等优先级。如果指定的报文缓冲区的 TXnPRI<1:0> 位被设置为 00，则此缓冲区具有最低优先级。如果两个或更多等待发送的报文具有相同优先级，报文以缓冲区编号递减的顺序发送。

### 20.5.3 发送过程

要启动报文发送必须将 TXREQn 位 (在 CiTRmnCON 中) 置 1。CAN 总线模块解决了将 TXREQn 位置 1 和帧起始 (Start-of-Frame, SOF) 之间的时序冲突，确保当优先级改变时，能在 SOF 产生之前正确解决时序冲突。当 TXREQn 位置 1 时，TXABTn、TXLARbn 和 TXERRn 标志位自动清零。

将 TXREQn 位置 1 仅仅标志报文缓冲区排序等待发送。当模块检测到总线可用时，模块开始发送具有最高优先级的报文。

如果第一次尝试就成功发送了报文，TXREQn 位将自动清零。同时如果 TXnIE 置 1 则产生一个中断。

如果报文发送失败，错误条件标志位之一将置 1，TXREQn 位将保持为 1，表示该报文仍然等待发送。如果报文在尝试发送过程中遇到错误条件，TXERRn 位将被置 1，错误条件会导致产生一个中断。如果报文在尝试发送过程中仲裁失败，TXLARbn 位将置 1。仲裁失败不会产生中断。

### 20.5.4 远程发送请求的自动处理

如果指定发送缓冲区的 RTRENn 位 (在 CiTRmnCON 寄存器中) 置 1，硬件自动发送在该缓冲区中的数据，以响应与指向该缓冲区的过滤器相匹配的远程发送请求。此种情况下用户无需手动开始发送。

### 20.5.5 中止报文发送

通过清零与每个报文缓冲区相关的 TXREQ 位，系统能中止报文发送。将 ABAT 位 (CiCTRL1<12>) 置 1 将请求中止所有等待发送的报文。如果报文还未开始发送或者报文已开始发送但由于仲裁失败或错误而被中断，那么将会执行中止。当模块将 TXABT 位置 1 且 TXnIF 标志位未自动置 1 时，表明发生了中止。

### 20.5.6 发送错误

ECAN 模块可检测到以下发送错误：

- 应答错误
- 格式错误
- 位错误

这些发送错误无需产生中断，但是发送错误计数器会表明有错误发生。每个发送错误都会引起发送错误计数器的值递增 1。一旦错误计数器的值超过 96，ERRIF 位 (CiINTF<5>) 和 TXWAR 位 (CiINTF<10>) 都将被置 1。一旦错误计数器的值超过 96，就会产生一个中断，同时中断标志寄存器的 TXWAR 位将被置 1。

20.5.7 发送中断

发送中断主要分为2组，每组包含不同的中断产生条件:

- 发送中断:
 

三个发送缓冲区中至少有一个为空（未预定）并且可以装入报文来预定发送。读取 TXnIF 标志可知哪一个发送缓冲区可用并引发了中断。
- 发送错误中断:
 

发送错误中断由 ERRIF 标志位指示。此标志位表示有错误条件发生。通过检查 CAN 中断标志寄存器 CiINTF 中的错误标志位，可以确定错误源。此寄存器中的标志位与收发错误有关。

  - 发送器警告中断:
 

TXWAR 位指示发送错误计数器已经达到CPU警告限制值 96。
  - 发送器错误被动:
 

TXEP 位 (CiINTF<12>) 指示发送错误计数器已经超过了错误被动的限制值 127，且该模块已经进入错误被动状态。
  - 总线关闭:
 

TXBO 位 (CiINTF<13>) 指示发送错误计数器已经超过了 255，且该模块已经进入总线关闭状态。

**注:** ECAN1 可触发 DMA 数据传输。如果 C1TX 或 C1RX 被选作 DMA 的 IRQ 源，在 ECAN1 发送或接收导致 C1TXIF 或 C1RXIF 位置 1 时，会发生 DMA 传输。

20.6 波特率设置

任何特定 CAN 总线上的所有节点必须具有相同的标称比特率。为了设置波特率，必须初始化以下参数:

- 同步跳转宽度
- 波特率预分频比
- 相位缓冲段
- 相位缓冲段 2 的长度确定
- 采样点
- 传播时间段位

20.6.1 位时序

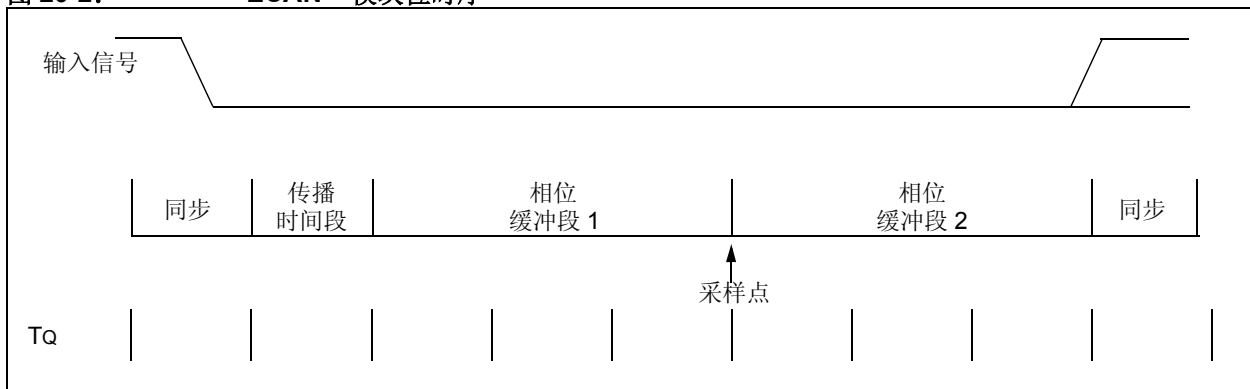
CAN 总线上的所有控制器必须具有相同的波特率和位长度。但是不同的控制器并不需要使用相同的主振荡器时钟。由于各个控制器的时钟频率不同，必须通过调节各段的时间份额数来调整波特率。

可以设想标称位时间被划分为几个互不重叠的时间段。这些段如图 20-2 所示。

- 同步段 (Sync Seg)
- 传播时间段 (Prop Seg)
- 相位缓冲段 1 (Phase1 Seg)
- 相位缓冲段 2 (Phase2 Seg)

时间段以及标称位时间由整数个时间单元组成，这些单元称作时间份额或 Tq。根据定义，标称位时间最少由 8 个 Tq 组成，最多由 25 个 Tq 组成。同样根据定义，最小标称位时间是 1 μs，对应最大 1 MHz 的比特率。

图 20-2: ECAN™ 模块位时序



### 20.6.2 预分频比设定

模块带有一个可编程预分频器，除对时钟进行固定的二分频外还提供其他时钟分频选择，其整数预分频值范围为 1 至 64。时间份额（Tq）是由振荡器周期得到的固定时间单元，如公式 20-1 所示。

**注：** F<sub>CAN</sub> 不能超过 40 MHz。如果 CANCKS = 0，那么 F<sub>CY</sub> 不能超过 20 MHz。

#### 公式 20-1: 时钟发生的时间份额

$$TQ = 2 (BRP<5:0> + 1) / F_{CAN}$$

### 20.6.3 传播时间段

这部分时间用于补偿网络内的物理延时。这些延时包括总线线路上的信号传播时间以及节点的内部延时。通过设置 PRSEG<2:0> 位（CiCFG2<2:0>），传播时间段长度可以编程为 1 Tq 到 8 Tq。

### 20.6.4 相位缓冲段

相位缓冲段用于在发送位时间内将接收位的采样点放置到最佳位置。采样点在相位缓冲段 1 和相位缓冲段 2 之间。这些段可通过重新同步来延长或缩短。相位缓冲段 1 的末尾决定了一个位周期内的采样点。相位缓冲段 1 可编程为 1 Tq 到 8 Tq。相位缓冲段 2 提供到下一次数据发送前的延时，相位缓冲段 2 可编程为 1 Tq 到 8 Tq，或者定义为等于相位缓冲段 1 和信息处理时间（2 Tq）两者中的较大者。通过将 SEG1PH<2:0>（CiCFG2<5:3>）位置 1 来初始化相位缓冲段 1，将 SEG2PH<2:0>（CiCFG2<10:8>）位置 1 来初始化相位缓冲段 2。

在设置相位缓冲段长度时，必须符合下列要求：（传播时间段 + 相位缓冲段 1 ≥ 相位缓冲段 2）。

### 20.6.5 采样点

采样点是一个时间点，在这个时间点读取总线电平并解释为相应位的值。采样点发生在相位缓冲段 1 的末尾。若位时序较慢而且包含很多 Tq，可以在采样点对总线线路进行多次采样。由 CAN 总线确定的电平对应于三个值中进行择多判定的结果。在采样点进行多次采样，且前两次采样的时间相隔 Tq/2。通过将 SAM 位（CiCFG2<6>）置 1 或清零，ECAN 模块允许用户选择同一点采样三次或一次。

通常，位采样应该发生在位时间的 60-70% 处，这取决于系统参数。

### 20.6.6 同步

为了补偿不同总线节点振荡器频率的相移，每个 CAN 控制器必须要能够与输入信号的相关信号沿同步。当检测到发送数据中的一个沿时，逻辑会将该沿的位置与预期时间（同步段）比较。然后电路将调节相位缓冲段 1 和相位缓冲段 2 的值。有两种机制用于同步。

#### 20.6.6.1 硬同步

硬同步仅在总线空闲期间，当一个从“隐性”转变到“显性”的边沿时（表明报文开始）才会进行。硬同步后，位时间计数器从同步段重新开始计数。硬同步把引起硬同步的沿强制地置于重新开始的位时间同步段之内。如果完成了一个硬同步，则在该位时间内将不会再有重新同步。

#### 20.6.6.2 重新同步

重新同步可以延长相位缓冲段 1 或缩短相位缓冲段 2。相位缓冲段延长或缩短的量有一个上限值，称作同步跳转宽度，由 SJW<1:0> 位（CiCFG1<7:6>）指定。同步跳转宽度的值将被加到相位缓冲段 1 或从相位缓冲段 2 减去。重新同步跳转宽度可以编程为 1 Tq 到 4 Tq。

在设置 SJW<1:0> 时必须符合下列要求：相位缓冲段 2 > 同步跳转宽度。

**注：** 在下面的寄存器说明中，寄存器标识符中的“i”表示 ECAN1 模块。  
寄存器标识符中的“n”表示缓冲区、过滤器或屏蔽寄存器的编号。  
寄存器标识符中的“m”表示特定 CAN 数据字段中的字数。

寄存器 20-1: CiCTRL1: ECAN™ 控制寄存器 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	
—	—	CSIDL	ABAT	CANCKS	REQOP<2:0>			
bit 15								bit 8
R-1	R-0	R-0	U-0	R/W-0	U-0	U-0	R/W-0	
OPMODE<2:0>			—	CANCAP	—	—	WIN	
bit 7								bit 0

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
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 15-14     **未实现:** 读为 0
- bit 13       **CSIDL:** 空闲模式停止位  
1 = 当器件进入空闲模式时, 模块停止工作  
0 = 在空闲模式下模块继续工作
- bit 12       **ABAT:** 中止所有等待处理的发送的位  
1 = 通知所有发送缓冲区中止发送  
0 = 当所有发送都中止时模块将清零此位
- bit 11       **CANCKS:** CAN 主时钟选择位  
1 = CAN F<sub>CAN</sub> 时钟为 F<sub>CY</sub>  
0 = CAN F<sub>CAN</sub> 时钟为 F<sub>OSC</sub>
- bit 10-8     **REQOP<2:0>:** 请求工作模式位  
000 = 设置正常工作模式  
001 = 设置禁止模式  
010 = 设置环回模式  
011 = 设置监听模式  
100 = 设置配置模式  
101 = 保留  
110 = 保留  
111 = 设置监听所有报文模式
- bit 7-5      **OPMODE<2:0>:** 工作模式位  
000 = 模块工作在正常工作模式下  
001 = 模块工作在禁止模式下  
010 = 模块工作在环回模式下  
011 = 模块工作在监听模式下  
100 = 模块工作在配置模式下  
101 = 保留  
110 = 保留  
111 = 模块工作在监听所有报文模式下
- bit 4        **未实现:** 读为 0
- bit 3        **CANCAP:** CAN 报文接收定时器捕捉事件使能位  
1 = 使能基于 CAN 报文接收的输入捕捉  
0 = 禁止 CAN 捕捉
- bit 2-1      **未实现:** 读为 0
- bit 0        **WIN:** SFR 映射窗口选择位  
1 = 使用过滤器窗口  
0 = 使用缓冲区窗口

寄存器 20-2: CiCTRL2: ECAN™ 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-0	R-0	R-0	R-0
—	—	—	DNCNT<4:0>				
bit 7						bit 0	

图注:

C = 可写位, 但是只能写 0 来清零此位			
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 4-0

**DNCNT<4:0>**: DeviceNet™ 过滤器位编号位

10010-11111 = 无效选择

10001 = 最多可将数据字节 3 的 bit 6 与 EID<17> 作比较

•

•

•

00001 = 最多可将数据字节 1 的 bit 7 与 EID<0> 作比较

00000 = 不比较数据字节

**寄存器 20-3: CIVEC: ECAN™ 中断编码寄存器**

U-0	U-0	U-0	R-0	R-0	R-0	R-0	R-0
—	—	—	FILHIT<4:0>				
bit 15							bit 8
U-0	R-1	R-0	R-0	R-0	R-0	R-0	R-0
—	ICODE<6:0>						
bit 7							bit 0

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
R = 可读位	W = 可写位
-n = POR 时的值	U = 未实现位, 读为 0
	0 = 清零
	1 = 置 1
	x = 未知

- bit 15-13     **未实现:** 读为 0
- bit 12-8     **FILHIT<4:0>:** 命中过滤器的编号位
  - 10000-11111 = 保留
  - 01111 = 过滤器 15
  - 
  - 
  - 
  - 00001 = 过滤器 1
  - 00000 = 过滤器 0
- bit 7         **未实现:** 读为 0
- bit 6-0     **ICODE<6:0>:** 中断标志编码位
  - 1000101-1111111 = 保留
  - 1000100 = FIFO 几乎满中断
  - 1000011 = 接收器溢出中断
  - 1000010 = 唤醒中断
  - 1000001 = 错误中断
  - 1000000 = 无中断
  - 
  - 
  - 
  - 0010000-0111111 = 保留
  - 0001111 = RB15 缓冲区中断
  - 
  - 
  - 
  - 0001001 = RB9 缓冲区中断
  - 0001000 = RB8 缓冲区中断
  - 0000111 = TRB7 缓冲区中断
  - 0000110 = TRB6 缓冲区中断
  - 0000101 = TRB5 缓冲区中断
  - 0000100 = TRB4 缓冲区中断
  - 0000011 = TRB3 缓冲区中断
  - 0000010 = TRB2 缓冲区中断
  - 0000001 = TRB1 缓冲区中断
  - 0000000 = TRB0 缓冲区中断

寄存器 20-4: CiFCTRL: ECAN™ FIFO 控制寄存器

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	
DMABS<2:0>			—	—	—	—	—	
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	
—	—	—	FSA<4:0>					
bit 7								bit 0

**图注:** C = 可写位, 但是只能写 0 来清零此位  
 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 15-13 **DMABS<2:0>**: DMA 缓冲区大小位  
 111 = 保留  
 110 = 使用 DMA RAM 中 32 个缓冲区  
 101 = 使用 DMA RAM 中 24 个缓冲区  
 100 = 使用 DMA RAM 中 16 个缓冲区  
 011 = 使用 DMA RAM 中 12 个缓冲区  
 010 = 使用 DMA RAM 中 8 个缓冲区  
 001 = 使用 DMA RAM 中 6 个缓冲区  
 000 = 使用 DMA RAM 中 4 个缓冲区

bit 12-5 **未实现**: 读为 0

bit 4-0 **FSA<4:0>**: FIFO 区域从哪个缓冲区开始位  
 11111 = 读缓冲区 RB31  
 11110 = 读缓冲区 RB30  
 .  
 .  
 .  
 00001 = 发送 / 接收缓冲区 TRB1  
 00000 = 发送 / 接收缓冲区 TRB0

**寄存器 20-5: CiFIFO: ECAN™ FIFO 状态寄存器**

U-0	U-0	R-0	R-0	R-0	R-0	R-0	R-0
—	—	FBP<5:0>					—
bit 15						bit 8	
U-0	U-0	R-0	R-0	R-0	R-0	R-0	R-0
—	—	FNRB<5:0>					—
bit 7						bit 0	

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
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      **FBP<5:0>:** FIFO 缓冲区指针位  
 011111 = RB31 缓冲区  
 011110 = RB30 缓冲区  
 .  
 .  
 .  
 000001 = TRB1 缓冲区  
 000000 = TRB0 缓冲区

bit 7-6      **未实现:** 读为 0

bit 5-0      **FNRB<5:0>:** FIFO 下一个读缓冲区指针位  
 011111 = RB31 缓冲区  
 011110 = RB30 缓冲区  
 .  
 .  
 .  
 000001 = TRB1 缓冲区  
 000000 = TRB0 缓冲区



寄存器 20-6: **CiINTF: ECAN™ 中断标志寄存器**

U-0	U-0	R-0	R-0	R-0	R-0	R-0	R-0	
—	—	TXBO	TXBP	RXBP	TXWAR	RXWAR	EWARN	
bit 15								bit 8

R/C-0	R/C-0	R/C-0	U-0	R/C-0	R/C-0	R/C-0	R/C-0	
IVRIF	WAKIF	ERRIF	—	FIFOIF	RBOVIF	RBIF	TBIF	
bit 7								bit 0

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
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 15-14 **未实现:** 读为 0
- bit 13 **TXBO:** 发送器处于错误状态总线关闭位  
1 = 发送器处于总线关闭状态  
0 = 发送器不处于总线关闭状态
- bit 12 **TXBP:** 发送器处于错误状态总线被动位  
1 = 发送器处于总线被动状态  
0 = 发送器不处于总线被动状态
- bit 11 **RXBP:** 接收器处于错误状态总线被动位  
1 = 接收器处于总线被动状态  
0 = 接收器不处于总线被动状态
- bit 10 **TXWAR:** 发送器处于错误状态警告位  
1 = 发送器处于错误警告状态  
0 = 发送器不处于错误警告状态
- bit 9 **RXWAR:** 接收器处于错误状态警告位  
1 = 接收器处于错误警告状态  
0 = 接收器不处于错误警告状态
- bit 8 **EWARN:** 发送器或接收器处于错误状态警告位  
1 = 发送器或接收器处于错误状态警告状态  
0 = 发送器或接收器不处于错误状态警告状态
- bit 7 **IVRIF:** 收到无效报文中断标志位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 6 **WAKIF:** 总线唤醒活动中断标志位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 5 **ERRIF:** 错误中断标志位 (CiINTF<13:8> 寄存器中的多个中断源)  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 4 **未实现:** 读为 0
- bit 3 **FIFOIF:** FIFO 几乎满中断标志位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 2 **RBOVIF:** 接收缓冲区溢出中断标志位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 1 **RBIF:** 接收缓冲区中断标志位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 0 **TBIF:** 发送缓冲区中断标志位  
1 = 产生了中断请求  
0 = 未产生中断请求

寄存器 20-7: **CiINTE: ECAN™ 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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IVRIE	WAKIE	ERRIE	—	FIFOIE	RBOVIE	RBIE	TBIE
bit 7							bit 0

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
R = 可读位	W = 可写位
-n = POR 时的值	U = 未实现位, 读为 0
	0 = 清零
	1 = 置 1
	x = 未知

- bit 15-8      **未实现:** 读为 0
- bit 7        **IVRIE:** 收到无效报文中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 6        **WAKIE:** 总线唤醒活动中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 5        **ERRIE:** 错误中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 4        **未实现:** 读为 0
- bit 3        **FIFOIE:** FIFO 几乎满中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 2        **RBOVIE:** 接收缓冲区溢出中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 1        **RBIE:** 接收缓冲区中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 0        **TBIE:** 发送缓冲区中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求

**寄存器 20-8: CiEC: ECAN™ 发送 / 接收错误计数寄存器**

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TERRCNT<7:0>							
bit 15				bit 8			
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RERRCNT<7:0>							
bit 7				bit 0			

**图注:** C = 可写位, 但是只能写 0 来清零此位  
 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 **TERRCNT<7:0>**: 发送错误计数位  
 bit 7-0 **RERRCNT<7:0>**: 接收错误计数位

**寄存器 20-9: CiCFG1: ECAN™ 波特率配置寄存器 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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SJW<1:0>		BRP<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 15-8 **未实现:** 读为 0  
 bit 7-6 **SJW<1:0>**: 同步跳转宽度位  
 11 = 长度为 4 x T<sub>Q</sub>  
 10 = 长度为 3 x T<sub>Q</sub>  
 01 = 长度为 2 x T<sub>Q</sub>  
 00 = 长度为 1 x T<sub>Q</sub>  
 bit 5-0 **BRP<5:0>**: 波特率预分频比位  
 11 1111 = T<sub>Q</sub> = 2 x 64 x 1/FCAN  
 .  
 .  
 .  
 00 0010 = T<sub>Q</sub> = 2 x 3 x 1/FCAN  
 00 0001 = T<sub>Q</sub> = 2 x 2 x 1/FCAN  
 00 0000 = T<sub>Q</sub> = 2 x 1 x 1/FCAN

**寄存器 20-10: CiCFG2: ECAN™ 波特率配置寄存器 2**

U-0	R/W-x	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	WAKFIL	—	—	—	SEG2PH<2:0>		
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SEG2PHTS	SAM	SEG1PH<2:0>			PRSEG<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            **WAKFIL:** 选择将 CAN 总线滤波器用于唤醒的位  
 1 = 将 CAN 总线滤波器用于唤醒  
 0 = 不将 CAN 总线滤波器用于唤醒
- bit 13-11        **未实现:** 读为 0
- bit 10-8         **SEG2PH<2:0>:** 相位缓冲段 2 位  
 111 = 长度为 8 x T<sub>Q</sub>  
 .  
 .  
 .  
 000 = 长度为 1 x T<sub>Q</sub>
- bit 7             **SEG2PHTS:** 相位缓冲段 2 时间选择位  
 1 = 可自由编程  
 0 = SEG1PH 位的最大值与信息处理时间 (Information Processing Time, IPT) 中的较大者
- bit 6             **SAM:** CAN 总线采样位  
 1 = 总线在采样点被采样三次  
 0 = 总线在采样点被采样一次
- bit 5-3          **SEG1PH<2:0>:** 相位缓冲段 1 位  
 111 = 长度为 8 x T<sub>Q</sub>  
 .  
 .  
 .  
 000 = 长度为 1 x T<sub>Q</sub>
- bit 2-0          **PRSEG<2:0>:** 传播时间段位  
 111 = 长度为 8 x T<sub>Q</sub>  
 .  
 .  
 .  
 000 = 长度为 1 x T<sub>Q</sub>

**寄存器 20-11: CiFEN1: ECAN™ 接收过滤器使能寄存器**

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
FLTEN15	FLTEN14	FLTEN13	FLTEN12	FLTEN11	FLTEN10	FLTEN9	FLTEN8
bit 15							bit 8
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
FLTEN7	FLTEN6	FLTEN5	FLTEN4	FLTEN3	FLTEN2	FLTEN1	FLTEN0
bit 7							bit 0

**图注:** C = 可写位, 但是只能写 0 来清零此位  
 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 15-0 **FLTENn:** 使能过滤器 n 接收报文的位  
 1 = 使能过滤器 n  
 0 = 禁止过滤器 n

**寄存器 20-12: CiBUFPNT1: ECAN™ 过滤器 0-3 缓冲区指针寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
F3BP<3:0>				F2BP<3:0>			
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
F1BP<3:0>				F0BP<3:0>			
bit 7							bit 0

**图注:** C = 可写位, 但是只能写 0 来清零此位  
 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 15-12 **F3BP<3:0>:** 过滤器 3 的接收缓冲区屏蔽寄存器  
 1111 = 命中过滤条件的数据被接收到接收 FIFO 缓冲区中  
 1110 = 命中过滤条件的数据被接收到接收缓冲区 14 中  
 .  
 .  
 .  
 0001 = 命中过滤条件的数据被接收到接收缓冲区 1 中  
 0000 = 命中过滤条件的数据被接收到接收缓冲区 0 中

bit 11-8 **F2BP<3:0>:** 过滤器 2 的接收缓冲区屏蔽寄存器 (与 bit 15-12 的值相同)  
 bit 7-4 **F1BP<3:0>:** 过滤器 1 的接收缓冲区屏蔽寄存器 (与 bit 15-12 的值相同)  
 bit 3-0 **F0BP<3:0>:** 过滤器 0 的接收缓冲区屏蔽寄存器 (与 bit 15-12 的值相同)

**寄存器 20-13: CiBUFNT2: ECAN™ 过滤器 4-7 缓冲区指针寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
F7BP<3:0>				F6BP<3:0>			
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
F5BP<3:0>				F4BP<3:0>			
bit 7				bit 0			

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
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 15-12      **F7BP<3:0>**: 过滤器 7 的接收缓冲区屏蔽寄存器  
 1111 = 命中过滤条件的数据被接收到接收 FIFO 缓冲区中  
 1110 = 命中过滤条件的数据被接收到接收缓冲区 14 中  
 .  
 .  
 .  
 0001 = 命中过滤条件的数据被接收到接收缓冲区 1 中  
 0000 = 命中过滤条件的数据被接收到接收缓冲区 0 中
- bit 11-8      **F6BP<3:0>**: 过滤器 6 的接收缓冲区屏蔽寄存器 (与 bit 15-12 的值相同)
- bit 7-4        **F5BP<3:0>**: 过滤器 5 的接收缓冲区屏蔽寄存器 (与 bit 15-12 的值相同)
- bit 3-0        **F4BP<3:0>**: 过滤器 4 的接收缓冲区屏蔽寄存器 (与 bit 15-12 的值相同)

**寄存器 20-14: CiBUFNT3: ECAN™ 过滤器 8-11 缓冲区指针寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
F11BP<3:0>				F10BP<3:0>			
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
F9BP<3:0>				F8BP<3:0>			
bit 7				bit 0			

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
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 15-12      **F11BP<3:0>**: 过滤器 11 的接收缓冲区屏蔽寄存器  
 1111 = 命中过滤条件的数据被接收到接收 FIFO 缓冲区中  
 1110 = 命中过滤条件的数据被接收到接收缓冲区 14 中  
 .  
 .  
 .  
 0001 = 命中过滤条件的数据被接收到接收缓冲区 1 中  
 0000 = 命中过滤条件的数据被接收到接收缓冲区 0 中
- bit 11-8      **F10BP<3:0>**: 过滤器 10 的接收缓冲区屏蔽寄存器 (与 bit 15-12 的值相同)
- bit 7-4        **F9BP<3:0>**: 过滤器 9 的接收缓冲区屏蔽寄存器 (与 bit 15-12 的值相同)
- bit 3-0        **F8BP<3:0>**: 过滤器 8 的接收缓冲区屏蔽寄存器 (与 bit 15-12 的值相同)

寄存器 20-15: CiBUFPNT4: ECAN™ 过滤器 12-15 缓冲区指针寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
F15BP<3:0>				F14BP<3:0>			
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
F13BP<3:0>				F12BP<3:0>			
bit 7				bit 0			

图注:

C = 可写位, 但是只能写 0 来清零此位			
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 15-12 **F15BP<3:0>**: 过滤器 15 的接收缓冲区屏蔽寄存器
  - 1111 = 命中过滤条件的数据被接收到接收 FIFO 缓冲区中
  - 1110 = 命中过滤条件的数据被接收到接收缓冲区 14 中
  - 
  - 
  - 
  - 0001 = 命中过滤条件的数据被接收到接收缓冲区 1 中
  - 0000 = 命中过滤条件的数据被接收到接收缓冲区 0 中
- bit 11-8 **F14BP<3:0>**: 过滤器 14 的接收缓冲区屏蔽寄存器 (与 bit 15-12 的值相同)
- bit 7-4 **F13BP<3:0>**: 过滤器 13 的接收缓冲区屏蔽寄存器 (与 bit 15-12 的值相同)
- bit 3-0 **F12BP<3:0>**: 过滤器 12 的接收缓冲区屏蔽寄存器 (与 bit 15-12 的值相同)

**寄存器 20-16: CiRXFnSID: ECAN™ 接收过滤器 n 标准标识符寄存器 (n = 0-15)**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
bit 15							bit 8

R/W-x	R/W-x	R/W-x	U-0	R/W-x	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDE	—	EID17	EID16
bit 7							bit 0

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
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 15-5      **SID<10:0>:** 标准标识符位  
 1 = 报文地址位 SIDx 必须为 1 才能与过滤器匹配  
 0 = 报文地址位 SIDx 必须为 0 才能与过滤器匹配
- bit 4        **未实现:** 读为 0
- bit 3        **EXIDE:** 扩展标识符使能位  
 如果 MIDE = 1:  
 1 = 只与带有扩展标识符地址的报文匹配  
 0 = 只与带有标准标识符地址的报文匹配  
 如果 MIDE = 0:  
 忽略 EXIDE 位。
- bit 2        **未实现:** 读为 0
- bit 1-0      **EID<17:16>:** 扩展标识符位  
 1 = 报文地址位 EIDx 必须为 1 才能与过滤器匹配  
 0 = 报文地址位 EIDx 必须为 0 才能与过滤器匹配



**寄存器 20-17: CiRXFnEID: ECAN™ 接收过滤器 n 扩展标识符寄存器 (n = 0-15)**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
bit 7							bit 0

**图注:** C = 可写位, 但是只能写 0 来清零此位  
 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 15-0 **EID<15:0>**: 扩展标识符位  
 1 = 报文地址位 EIDx 必须为 1 才能与过滤器匹配  
 0 = 报文地址位 EIDx 必须为 0 才能与过滤器匹配

**寄存器 20-18: CiFMSKSEL1: ECAN™ 过滤器 7-0 屏蔽选择寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
F7MSK<1:0>		F6MSK<1:0>		F5MSK<1:0>		F4MSK<1:0>	
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
F3MSK<1:0>		F2MSK<1:0>		F1MSK<1:0>		F0MSK<1:0>	
bit 7							bit 0

**图注:** C = 可写位, 但是只能写 0 来清零此位  
 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 **F7MSK<1:0>**: 过滤器 7 的屏蔽源位  
 11 = 无屏蔽  
 10 = 接收屏蔽寄存器 2 包含屏蔽值  
 01 = 接收屏蔽寄存器 1 包含屏蔽值  
 00 = 接收屏蔽寄存器 0 包含屏蔽值

bit 13-12 **F6MSK<1:0>**: 过滤器 6 的屏蔽源位 (与 bit 15-14 的值相同)  
 bit 11-10 **F5MSK<1:0>**: 过滤器 5 的屏蔽源位 (与 bit 15-14 的值相同)  
 bit 9-8 **F4MSK<1:0>**: 过滤器 4 的屏蔽源位 (与 bit 15-14 的值相同)  
 bit 7-6 **F3MSK<1:0>**: 过滤器 3 的屏蔽源位 (与 bit 15-14 的值相同)  
 bit 5-4 **F2MSK<1:0>**: 过滤器 2 的屏蔽源位 (与 bit 15-14 的值相同)  
 bit 3-2 **F1MSK<1:0>**: 过滤器 1 的屏蔽源位 (与 bit 15-14 的值相同)  
 bit 1-0 **F0MSK<1:0>**: 过滤器 0 的屏蔽源位 (与 bit 15-14 的值相同)

**寄存器 20-19: CiFMSKSEL2: ECAN™ 过滤器 15-8 屏蔽选择寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
F15MSK<1:0>		F14MSK<1:0>		F13MSK<1:0>		F12MSK<1:0>	
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
F11MSK<1:0>		F10MSK<1:0>		F9MSK<1:0>		F8MSK<1:0>	
bit 7						bit 0	

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
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 15-14     **F15MSK<1:0>**: 过滤器 15 的屏蔽源位  
 11 = 无屏蔽  
 10 = 接收屏蔽寄存器 2 包含屏蔽值  
 01 = 接收屏蔽寄存器 1 包含屏蔽值  
 00 = 接收屏蔽寄存器 0 包含屏蔽值
- bit 13-12    **F14MSK<1:0>**: 过滤器 14 的屏蔽源位 (与 bit 15-14 的值相同)
- bit 11-10    **F13MSK<1:0>**: 过滤器 13 的屏蔽源位 (与 bit 15-14 的值相同)
- bit 9-8       **F12MSK<1:0>**: 过滤器 12 的屏蔽源位 (与 bit 15-14 的值相同)
- bit 7-6       **F11MSK<1:0>**: 过滤器 11 的屏蔽源位 (与 bit 15-14 的值相同)
- bit 5-4       **F10MSK<1:0>**: 过滤器 10 的屏蔽源位 (与 bit 15-14 的值相同)
- bit 3-2       **F9MSK<1:0>**: 过滤器 9 的屏蔽源位 (与 bit 15-14 的值相同)
- bit 1-0       **F8MSK<1:0>**: 过滤器 8 的屏蔽源位 (与 bit 15-14 的值相同)

**寄存器 20-20: CiRXMnSID: ECAN™接收过滤屏蔽寄存器 n 标准标识符寄存器 (n = 0-2)**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
bit 15							bit 8
R/W-x	R/W-x	R/W-x	U-0	R/W-x	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	MIDE	—	EID17	EID16
bit 7							bit 0

**图注:** C = 可写位, 但是只能写 0 来清零此位  
 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 **SID<10:0>**: 标准标识符位  
 1 = 过滤器比较操作包含 SIDx 位  
 0 = 过滤器比较操作与 SIDx 位无关
- bit 4 **未实现**: 读为 0
- bit 3 **MIDE**: 标识符接收模式位  
 1 = 只匹配与过滤器中 EXIDE 位对应的报文类型 (标准或扩展地址)  
 0 = 如果过滤器匹配则与标准或扩展地址报文匹配  
 (即, 如果 (过滤器 SID) = (报文 SID) 或 (过滤器 SID/EID) = (报文 SID/EID))
- bit 2 **未实现**: 读为 0
- bit 1-0 **EID<17:16>**: 扩展标识符位  
 1 = 过滤器比较操作包含 EIDx 位  
 0 = 过滤器比较操作与 EIDx 位无关

**寄存器 20-21: CiRXMnEID: ECAN™接收过滤屏蔽寄存器 n 扩展标识符寄存器 (n = 0-2)**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 15							bit 8
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
bit 7							bit 0

**图注:** C = 可写位, 但是只能写 0 来清零此位  
 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 15-0 **EID<15:0>**: 扩展标识符位  
 1 = 过滤器比较操作包含 EIDx 位  
 0 = 过滤器比较操作与 EIDx 位无关

**寄存器 20-22: CiRXFUL1: ECAN™ 接收缓冲区满寄存器 1**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXFUL15	RXFUL14	RXFUL13	RXFUL12	RXFUL11	RXFUL10	RXFUL9	RXFUL8
bit 15							bit 8

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXFUL7	RXFUL6	RXFUL5	RXFUL4	RXFUL3	RXFUL2	RXFUL1	RXFUL0
bit 7							bit 0

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
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 15-0      **RXFUL<15:0>:** 接收缓冲区 n 满状态位  
 1 = 缓冲区为满 (由模块置 1)  
 0 = 缓冲区为空

**寄存器 20-23: CiRXFUL2: ECAN™ 接收缓冲区满寄存器 2**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXFUL31	RXFUL30	RXFUL29	RXFUL28	RXFUL27	RXFUL26	RXFUL25	RXFUL24
bit 15							bit 8

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXFUL23	RXFUL22	RXFUL21	RXFUL20	RXFUL19	RXFUL18	RXFUL17	RXFUL16
bit 7							bit 0

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
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 15-0      **RXFUL<31:16>:** 接收缓冲区 n 满状态位  
 1 = 缓冲区为满 (由模块置 1)  
 0 = 缓冲区为空

**寄存器 20-24: CiRXOVF1: ECAN™ 接收缓冲区溢出寄存器 1**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXOVF15	RXOVF14	RXOVF13	RXOVF12	RXOVF11	RXOVF10	RXOVF9	RXOVF8
bit 15							bit 8

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXOVF7	RXOVF6	RXOVF5	RXOVF4	RXOVF3	RXOVF2	RXOVF1	RXOVF0
bit 7							bit 0

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
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 15-0 **RXOVF<15:0>**: 接收缓冲区 n 溢出状态位  
 1 = 模块尝试写一个已满的缓冲区 (由模块置 1)  
 0 = 无溢出条件

**寄存器 20-25: CiRXOVF2: ECAN™ 接收缓冲区溢出寄存器 2**

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXOVF31	RXOVF30	RXOVF29	RXOVF28	RXOVF27	RXOVF26	RXOVF25	RXOVF24
bit 15							bit 8

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXOVF23	RXOVF22	RXOVF21	RXOVF20	RXOVF19	RXOVF18	RXOVF17	RXOVF16
bit 7							bit 0

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
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 15-0 **RXOVF<31:16>**: 接收缓冲区 n 溢出状态位  
 1 = 模块尝试写一个已满的缓冲区 (由模块置 1)  
 0 = 无溢出条件

**寄存器 20-26: CiTRmnCON: ECAN™ 发送 / 接收缓冲区 m 控制寄存器**  
 (m = 0,2,4,6 ; n = 1,3,5,7)

R/W-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
TXENn	TXABTn	TXLARBn	TXERRn	TXREQn	RTRENn	TXnPRI<1:0>	
bit 15							bit 8

R/W-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
TXENm	TXABTm <sup>(1)</sup>	TXLARBm <sup>(1)</sup>	TXERRm <sup>(1)</sup>	TXREQm	RTRENm	TXmPRI<1:0>	
bit 7							bit 0

<b>图注:</b>	C = 可写位, 但是只能写 0 来清零此位
R = 可读位	W = 可写位
-n = POR 时的值	U = 未实现位, 读为 0
	0 = 清零
	1 = 置 1
	x = 未知

- bit 15-8 请参见控制缓冲区 n 中 bit 7-0 的定义
- bit 7 **TXENm:** 发送 / 接收缓冲区选择位  
 1 = 缓冲区 TRBn 是发送缓冲区  
 0 = 缓冲区 TRBn 是接收缓冲区
- bit 6 **TXABTm:** 报文中止位 <sup>(1)</sup>  
 1 = 中止报文  
 0 = 成功完成报文发送
- bit 5 **TXLARBm:** 报文仲裁失败位 <sup>(1)</sup>  
 1 = 报文在发送过程中仲裁失败  
 0 = 报文在发送过程中无仲裁失败
- bit 4 **TXERRm:** 发送过程中错误检测位 <sup>(1)</sup>  
 1 = 报文发送时发生总线错误  
 0 = 报文发送时未发生总线错误
- bit 3 **TXREQm:** 报文发送请求位  
 1 = 请求发送报文。当报文发送成功后, 此位自动清零。  
 0 = 当该位为 1 时清零该位将请求中止报文。
- bit 2 **RTRENm:** 自动远程发送使能位  
 1 = 当收到远程发送时, 将 TXREQ 置 1  
 0 = 当收到远程发送时, TXREQ 不受影响
- bit 1-0 **TXmPRI<1:0>:** 报文发送优先级位  
 11 = 最高报文优先级  
 10 = 中高报文优先级  
 01 = 中低报文优先级  
 00 = 最低报文优先级

**注 1:** 当 TXREQ 置 1 时此位被清零。

**注:** 缓冲区、SID、EID、DLC、数据字段和接收状态寄存器位于 DMA RAM 中。

## 20.7 ECAN 报文缓冲区

ECAN 报文缓冲区是 DMA RAM 存储区的一部分。它们不是 ECAN 特殊功能寄存器。用户应用程序必须直接写入为 ECAN 报文缓冲区配置的 DMA RAM 区域。缓冲区的地址和大小由用户应用程序定义。

### 缓冲区 20-1: ECAN™ 报文缓冲区字 0

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	
—	—	—	SID10	SID9	SID8	SID7	SID6	
bit 15								bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID5	SID4	SID3	SID2	SID1	SID0	SRR	IDE
bit 7						bit 0	

**图注:**

R = 可读位                      W = 可写位                      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 12-2    **SID<10:0>:** 标准标识符位  
 bit 1        **SRR:** 替代远程请求位  
               1 = 报文将请求远程发送  
               0 = 正常报文  
 bit 0        **IDE:** 扩展标识符位  
               1 = 报文将发送扩展标识符  
               0 = 报文将发送标准标识符

### 缓冲区 20-2: ECAN™ 报文缓冲区字 1

U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	
—	—	—	—	EID17	EID16	EID15	EID14	
bit 15								bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID13	EID12	EID11	EID10	EID9	EID8	EID7	EID6
bit 7						bit 0	

**图注:**

R = 可读位                      W = 可写位                      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 11-0    **EID<17:6>:** 扩展标识符位

**缓冲区 20-3: ECAN™ 报文缓冲区字 2**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID5	EID4	EID3	EID2	EID1	EID0	RTR	RB1
bit 15							bit 8
U-x	U-x	U-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	RB0	DLC3	DLC2	DLC1	DLC0
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      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      **EID<5:0>**: 扩展标识符位
- bit 9            **RTR**: 远程发送请求位  
                  1 = 报文将请求远程发送  
                  0 = 正常报文
- bit 8            **RB1**: 保留的 Bit 1  
                  用户必须按 CAN 协议将此位设置为 0。
- bit 7-5        **未实现**: 读为 0
- bit 4            **RB0**: 保留的 Bit 0  
                  用户必须按 CAN 协议将此位设置为 0。
- bit 3-0        **DLC<3:0>**: 数据长度编码位

**缓冲区 20-4: ECAN™ 报文缓冲区字 3**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Byte 1							
bit 15							bit 8
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Byte 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        **Byte 1<15:8>**: ECAN™ 报文字节 1
- bit 7-0        **Byte 0<7:0>**: ECAN 报文字节 0



**缓冲区 20-5: ECAN™ 报文缓冲区字 4**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Byte 3							
bit 15				bit 8			
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Byte 2							
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 15-8      **Byte 3<15:8>**: ECAN™ 报文字节 3

bit 7-0      **Byte 2<7:0>**: ECAN 报文字节 2

**缓冲区 20-6: ECAN™ 报文缓冲区字 5**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Byte 5							
bit 15				bit 8			
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Byte 4							
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 15-8      **Byte 5<15:8>**: ECAN™ 报文字节 5

bit 7-0      **Byte 4<7:0>**: ECAN 报文字节 4

**缓冲区 20-7: ECAN™ 报文缓冲区字 6**

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Byte 7							
bit 15							bit 8
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Byte 6							
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 15-8      **Byte 7<15:8>:** ECAN™ 报文字节 7

bit 7-0      **Byte 6<7:0>:** ECAN 报文字节 6

**缓冲区 20-8: ECAN™ 报文缓冲区字 7**

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	FILHIT<4:0> <sup>(1)</sup>				
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-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 15-13      **未实现:** 读为 0

bit 12-8      **FILHIT<4:0>:** 命中过滤器的编号位 <sup>(1)</sup>  
对导致写入此缓冲区的过滤器的编号进行编码。

bit 7-0      **未实现:** 读为 0

**注 1:** 模块只能对接收缓冲区执行写操作, 不用于发送缓冲区。

## 21.0 10 位 /12 位模数转换器 (ADC1)

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“*dsPIC33F Family Reference Manual*”的“Section 16. Analog-to-Digital Converter (ADC)” (DS70183)，该文档可从 Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 下载。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件具有最多 9 路 ADC 输入通道。

AD12B 位 (AD1CON1<10>) 允许用户对每个 ADC 模块进行配置，可以配置为 10 位、4 采样 / 保持 ADC (默认配置) 或 12 位、1 采样 / 保持 ADC。

**注：** 在修改 AD12B 位之前必须禁止 ADC 模块。

### 21.1 关键特性

10 位 ADC 配置具有以下关键特性：

- 逐次逼近 (Successive Approximation Register, SAR) 型转换
- 转换速度最高为 1.1 Msps
- 最多 9 个模拟输入引脚
- 外部参考电压输入引脚
- 可同时采样最多 4 个模拟输入引脚
- 自动通道扫描模式
- 可选的转换触发源
- 可选的缓冲区填充模式
- 四种结果对齐选项 (有符号 / 无符号, 小数 / 整数)
- 可在 CPU 休眠和空闲模式下工作

12 位 ADC 配置支持所有上述特性，但以下特性除外：

- 在 12 位配置中，支持最高 500 ksps 的转换速度
- 在 12 位配置中只有 1 个采样 / 保持放大器，因此不支持多通道同时采样。

根据特定的器件引脚配置，ADC 最多有 9 个模拟输入引脚，指定为 AN0 到 AN8。此外，有两个可用于外部参考电压连接的模拟输入引脚。这两个参考电压输入可以与其他模拟输入引脚共用。实际的模拟输入引脚数和外部参考电压输入配置取决于具体的器件。

图 21-1 和图 21-2 给出了 ADC 模块的框图。

### 21.2 ADC 初始化

应该按照下面的步骤来进行配置。

1. 配置 ADC 模块：
  - a) 选择端口引脚作为模拟输入引脚 (AD1PCFGH<15:0>或AD1PCFGL<15:0>)
  - b) 选择参考电压源以匹配预期的模拟输入范围 (AD1CON2<15:13>)
  - c) 选择模拟转换时钟以使期望的数据速率与处理器时钟匹配 (AD1CON3<7:0>)
  - d) 确定使用多少路采样 / 保持通道 (AD1CON2<9:8>和AD1PCFGH<15:0>或AD1PCFGL<15:0>)
  - e) 选择适当的采样 / 转换序列 (AD1CON1<7:5>和AD1CON3<12:8>)
  - f) 选择转换结果在缓冲区中的存储方式 (AD1CON1<9:8>)
  - g) 开启 ADC 模块 (AD1CON1<15>)
2. 配置 ADC 中断 (如需要)：
  - a) 清零 AD1IF 位
  - b) 选择 ADC 中断优先级

### 21.3 ADC 和 DMA

如果触发中断前需要缓冲多个转换结果，可使用 DMA 数据传输。ADC1 可触发 DMA 数据传输。如果选择 ADC1 作为 DMA 的 IRQ 源，则当 ADC1 采样转换过程完成使 AD1IF 位置 1 时将发生 DMA 传输。

SMPI<3:0> 位 (AD1CON2<5:2>) 用于选择 DMA RAM 缓冲区指针递增的频率。

ADDMABM 位 (AD1CON1<12>) 确定如何将转换结果填充到 ADC 使用的 DMA RAM 缓冲区。如果该位置 1，则以转换顺序来写入 DMA 缓冲区。模块向 DMA 通道提供的地址与用于非 DMA 独立缓冲区的地址相同。如果该位清零，则以分散 / 集中模式写入 DMA 缓冲区。模块根据模拟输入编号和 DMA 缓冲区的大小向 DMA 通道提供分散 / 集中地址。

图 21-1: ADC 模块框图

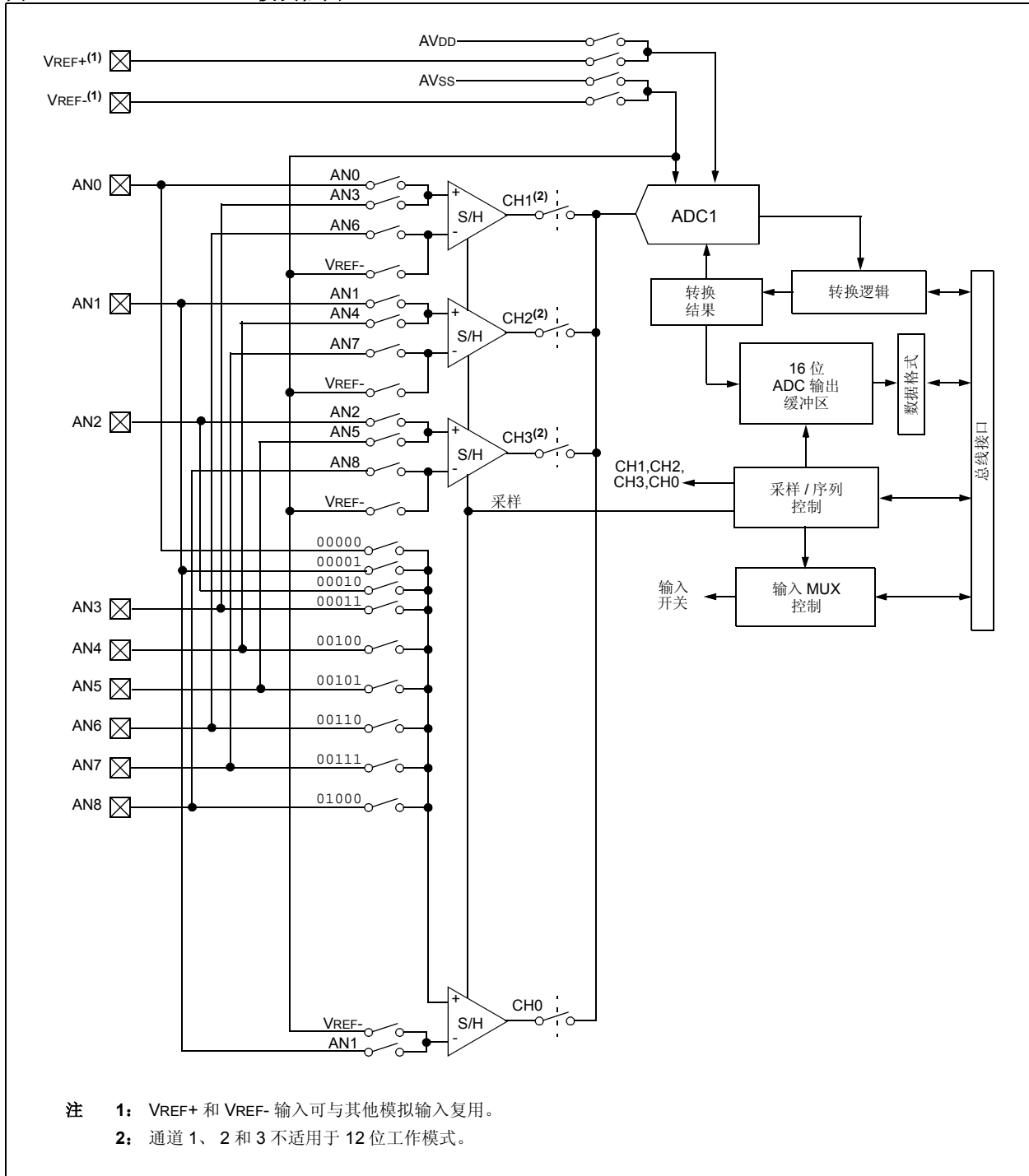
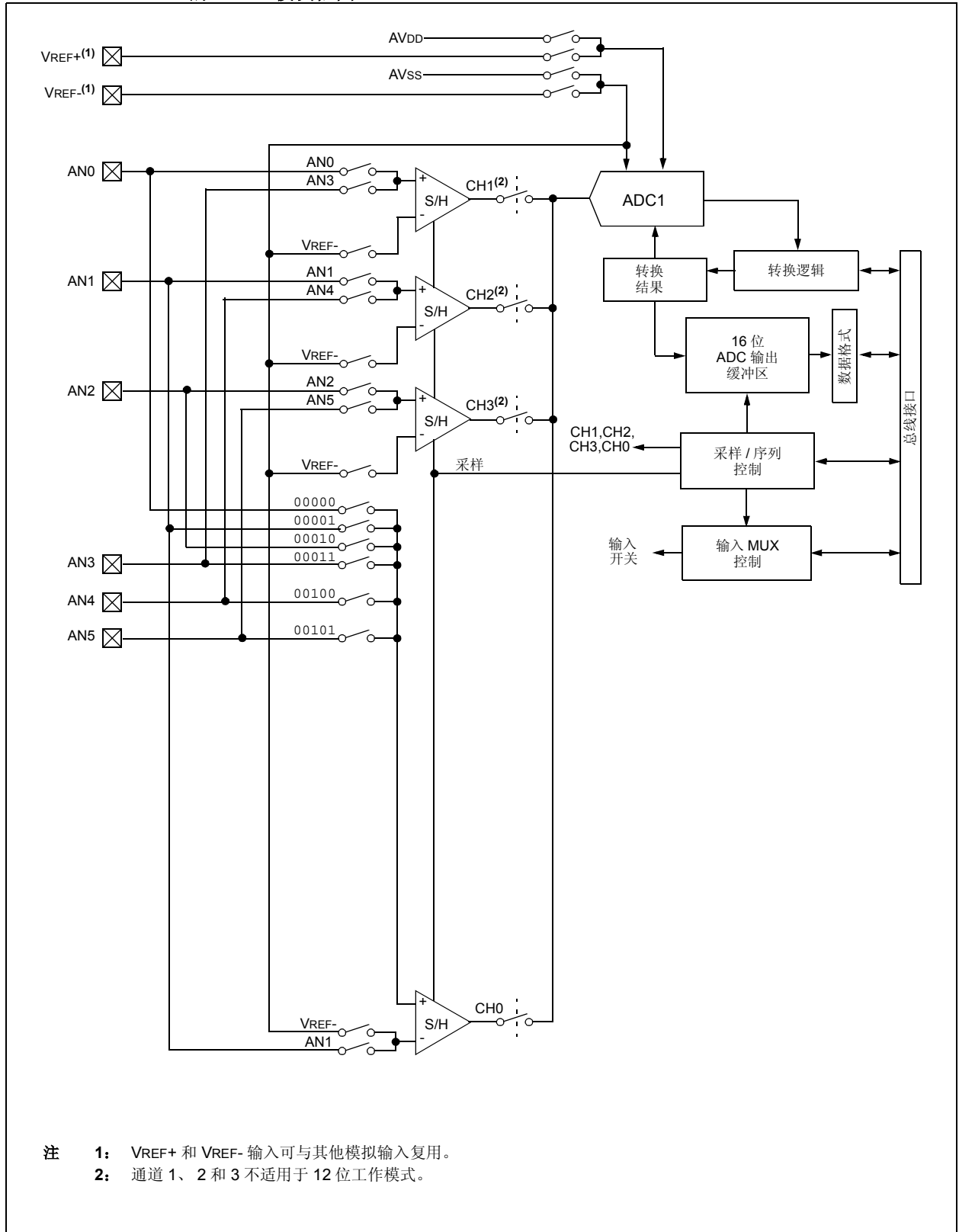


图 21-2: dsPIC33FJ32MC302、dsPIC33FJ64MC202/802、dsPIC33FJ128MC202/802 器件的 ADC1 模块框图



公式 21-1: ADC 转换时钟周期

$$T_{AD} = T_{CY}(ADCS + 1)$$

$$ADCS = \frac{T_{AD}}{T_{CY}} - 1$$

图 21-3: ADC 传递函数 (以 10 位 ADC 为例)

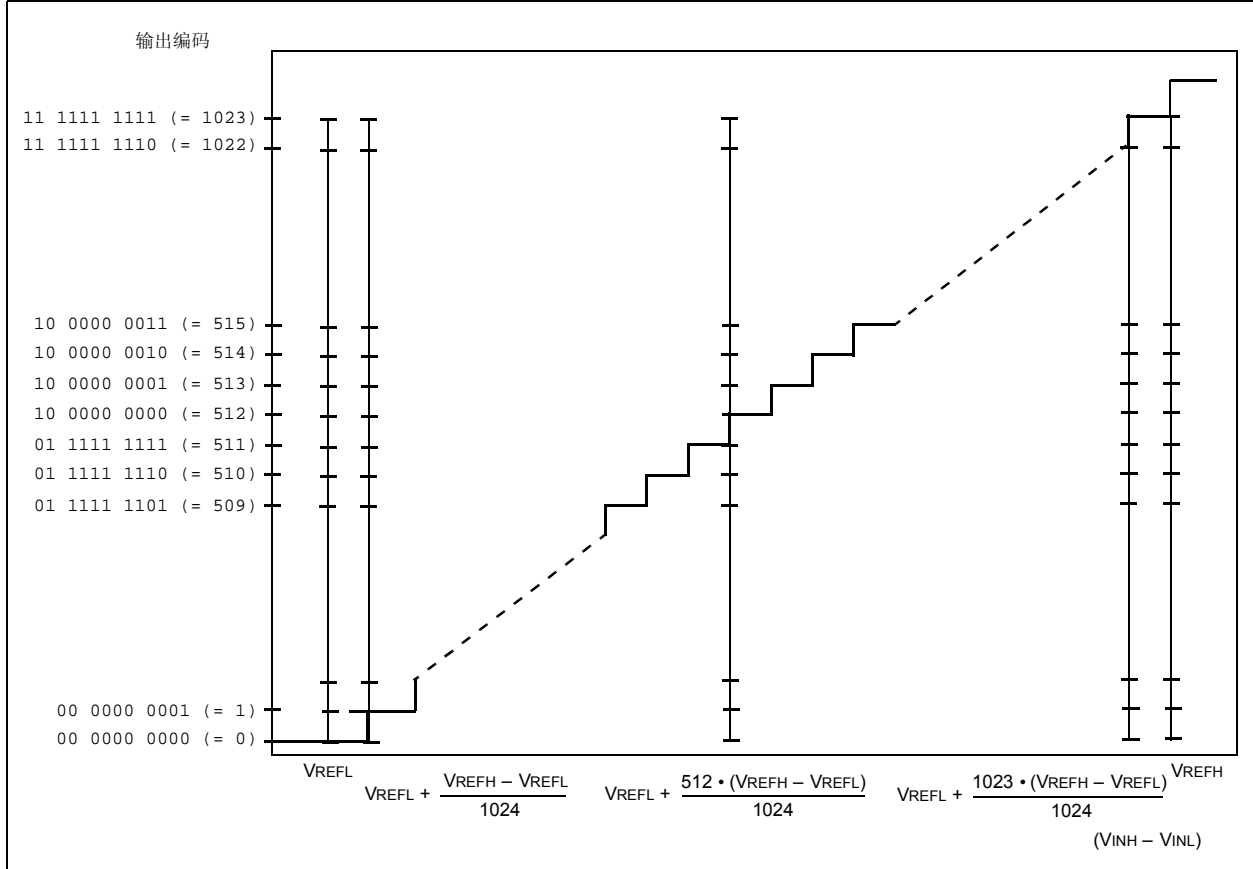
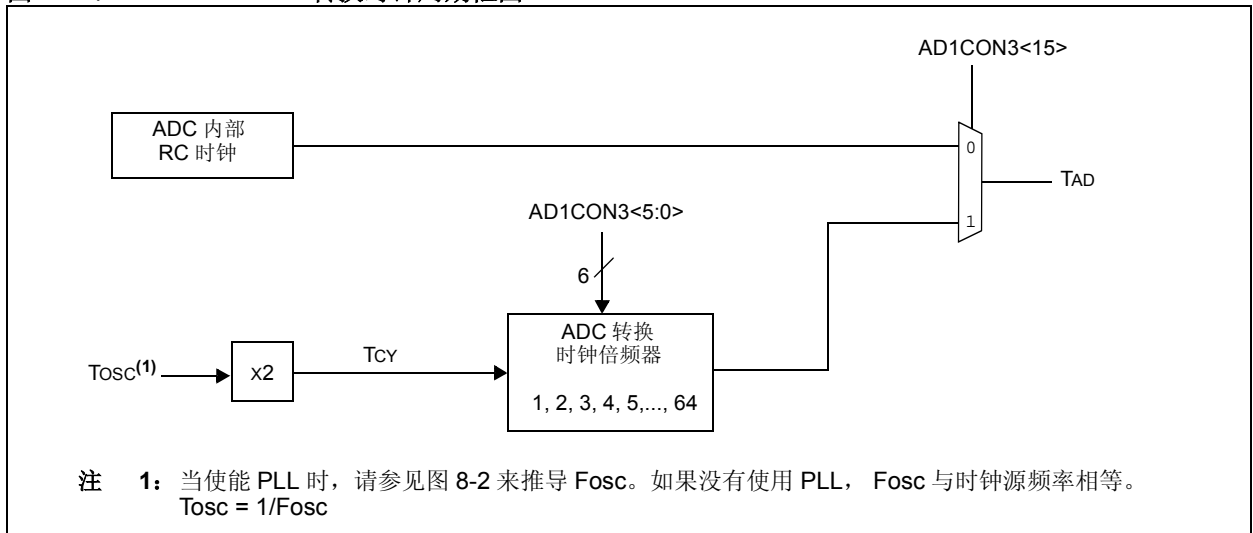


图 21-4: ADC 转换时钟周期框图



寄存器 21-1: AD1CON1: ADC1 控制寄存器 1

R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
ADON	—	ADSIDL	ADDMABM	—	AD12B	FORM<1:0>	
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/C-0
SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE
bit 7						bit 0	

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

- bit 15      **ADON:** ADC 工作模式位  
1 = ADC 模块正在工作  
0 = ADC 模块关闭
- bit 14      **未实现:** 读为 0
- bit 13      **ADSIDL:** 在空闲模式下停止位  
1 = 当器件进入空闲模式时, 模块停止工作  
0 = 在空闲模式下模块继续工作
- bit 12      **ADDMABM:** DMA 缓冲区构建模式位  
1 = 以转换顺序写入 DMA 缓冲区。模块向 DMA 通道提供的地址与用于非 DMA 独立缓冲区的地址相同。  
0 = 以分散 / 集中模式写入 DMA 缓冲区。模块根据模拟输入编号和 DMA 缓冲区的大小向 DMA 通道提供分散 / 集中地址。
- bit 11      **未实现:** 读为 0
- bit 10      **AD12B:** 10 位或 12 位工作模式位  
1 = 12 位 1 通道 ADC 工作  
0 = 10 位 4 通道 ADC 工作
- bit 9-8      **FORM<1:0>:** 数据输出格式位  
对于 10 位工作模式:  
11 = 有符号小数 (DOUT = sddd dddd dd00 0000, 其中 s = d<9> 取反)  
10 = 小数 (DOUT = dddd dddd dd00 0000)  
01 = 有符号整数 (DOUT = ssss sssd dddd dddd, 其中 s = d<9> 取反)  
00 = 整数 (DOUT = 0000 00dd dddd dddd)  
对于 12 位工作模式:  
11 = 有符号小数 (DOUT = sddd dddd dddd 0000, 其中 s = d<11> 取反)  
10 = 小数 (DOUT = dddd dddd dddd 0000)  
01 = 有符号整数 (DOUT = ssss sddd dddd dddd, 其中 s = d<11> 取反)  
00 = 整数 (DOUT = 0000 dddd dddd dddd)
- bit 7-5      **SSRC<2:0>:** 采样时钟源选择位  
111 = 由内部计数器结束采样并启动转换 (自动转换)  
110 = 保留  
101 = 由电机控制 PWM2 间隔结束采样并启动转换  
100 = 由 GP 定时器 (ADC1 采用 Timer5) 比较结束采样并启动转换  
011 = 由电机控制 PWM1 间隔结束采样并启动转换  
010 = 由 GP 定时器 (ADC1 采用 Timer3) 比较结束采样并启动转换  
001 = 由 INT 引脚的有效电平跳变结束采样并启动转换  
000 = 由清零采样位结束采样并启动转换
- bit 4      **未实现:** 读为 0

**寄存器 21-1: AD1CON1: ADC1 控制寄存器 1 (续)**

- bit 3 **SIMSAM:** 同步采样选择位 (仅当 CHPS<1:0> = 01 或 1x 时适用)  
当 AD12B = 1 时, SIMSAM 为: U-0, 未实现, 读为 0  
1 = 同时采样 CH0、CH1、CH2 和 CH3 (当 CHPS<1:0> = 1x 时) 或者  
同时采样 CH0 和 CH1 (当 CHPS<1:0> = 01 时)  
0 = 按顺序依次采样多路通道
- bit 2 **ASAM:** ADC 采样自动启动位  
1 = 上次转换结束后立即开始采样。SAMP 位自动置 1。  
0 = SAMP 位置 1 时开始采样
- bit 1 **SAMP:** ADC 采样使能位  
1 = ADC 采样 / 保持放大器正在采样  
0 = ADC 采样 / 保持放大器在保持采样结果  
如果 ASAM = 0, 由软件写入 1 开始采样。如果 ASAM = 1, 该位由硬件自动置 1。  
如果 SSRC = 000, 由软件写入 0 结束采样并启动转换。如果 SSRC ≠ 000,  
由硬件自动清零来结束采样并启动转换。
- bit 0 **DONE:** ADC 转换状态位  
1 = ADC 转换周期完成。  
0 = ADC 转换尚未开始或在进行中  
当 ADC 转换完成时, 由硬件自动置 1。可由软件写入 0 来清零 DONE 状态位 (不允许由软件写入 1)。  
清零该位不会影响进行中的任何操作。在新的转换开始时由硬件自动清零。



寄存器 21-2: AD1CON2: ADC1 控制寄存器 2

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	
VCFG<2:0>			—	—	CSCNA	CHPS<1:0>		
bit 15								bit 8

R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
BUFS	—	SMPI<3:0>				BUFM	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 15-13      **VCFG<2:0>**: 转换器参考电压配置位

	ADREF+	ADREF-
000	AVDD	AVSS
001	外部 VREF+	AVSS
010	AVDD	外部 VREF-
011	外部 VREF+	外部 VREF-
1xx	AVDD	Avss

bit 12-11      未实现: 读为 0

bit 10          **CSCNA**: 选择是否在使用采样多路开关 A 时扫描 CH0+ 输入的位

1 = 扫描输入  
 0 = 不扫描输入

bit 9-8        **CHPS<1:0>**: 选择要使用通道的位

当 AD12B = 1 时, CHPS<1:0> 为: U-0, 未实现, 读为 0

1x = 转换 CH0、CH1、CH2 和 CH3  
 01 = 转换 CH0 和 CH1  
 00 = 转换 CH0

bit 7          **BUFS**: 缓冲区填充状态位 (仅当 BUFM = 1 时有效)

1 = ADC 当前在填充缓冲区的 0x8-0xF, 用户应用程序应访问 0x0-0x7 中的数据  
 0 = ADC 当前在填充缓冲区的 0x0-0x7, 用户应用程序应访问 0x8-0xF 中的数据

bit 6          未实现: 读为 0

bit 5-2        **SMPI<3:0>**: 选择 DMA 地址递增速率或每次产生中断时采样 / 转换操作次数的位

1111 = 每完成 16 次采样 / 转换操作后递增 DMA 地址或产生中断  
 1110 = 每完成 15 次采样 / 转换操作后递增 DMA 地址或产生中断  
 .  
 .  
 .  
 0001 = 每完成 2 次采样 / 转换操作后递增 DMA 地址或产生中断  
 0000 = 每完成 1 次采样 / 转换操作后递增 DMA 地址或产生中断

bit 1          **BUFM**: 缓冲区填充模式选择位

1 = 第一次产生中断时从地址 0x0 开始填充缓冲区, 而下一次产生中断时从地址 0x8 开始填充缓冲区  
 0 = 总是从地址 0x0 开始填充缓冲区

bit 0          **ALTS**: 备用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

**寄存器 21-3: AD1CON3: ADC1 控制寄存器 3**

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	—	—	SAMC<4:0>				
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
ADCS<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            **ADRC:** ADC 转换时钟源位  
                   1 = ADC 内部 RC 时钟  
                   0 = 时钟由系统时钟产生

bit 14-13        **未实现:** 读为 0

bit 12-8        **SAMC<4:0>:** 自动采样时间位  
                   11111 = 31 TAD  
                   .  
                   .  
                   .  
                   00001 = 1 TAD  
                   00000 = 0 TAD

bit 7-0         **ADCS<7:0>:** ADC 转换时钟选择位  
                   11111111 = T<sub>CY</sub> · (ADCS<7:0> + 1) = 256 · T<sub>CY</sub> = TAD  
                   .  
                   .  
                   .  
                   00000010 = T<sub>CY</sub> · (ADCS<7:0> + 1) = 3 · T<sub>CY</sub> = TAD  
                   00000001 = T<sub>CY</sub> · (ADCS<7:0> + 1) = 2 · T<sub>CY</sub> = TAD  
                   00000000 = T<sub>CY</sub> · (ADCS<7:0> + 1) = 1 · T<sub>CY</sub> = TAD

寄存器 21-4: 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>		
bit 7						bit 0	

图注:

R = 可读位                      W = 可写位                      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>**: 选择为每个模拟输入分配的 DMA 缓冲区存储单元数

- 111 = 为每个模拟输入分配 128 字的 DMA 缓冲区
- 110 = 为每个模拟输入分配 64 字的 DMA 缓冲区
- 101 = 为每个模拟输入分配 32 字的 DMA 缓冲区
- 100 = 为每个模拟输入分配 16 字的 DMA 缓冲区
- 011 = 为每个模拟输入分配 8 字的 DMA 缓冲区
- 010 = 为每个模拟输入分配 4 字的 DMA 缓冲区
- 001 = 为每个模拟输入分配 2 字的 DMA 缓冲区
- 000 = 为每个模拟输入分配 1 字的 DMA 缓冲区

**寄存器 21-5: AD1CHS123: ADC1 输入通道 1、2 和 3 选择寄存器**

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CH123NB<1:0>		CH123SB
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CH123NA<1:0>		CH123SA
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      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 10-9      **CH123NB<1:0>:** 采样多路开关 B 的通道 1、2 和 3 的反相输入选择位  
 仅限 dsPIC33FJ32MC302、 dsPIC33FJ64MC202/802 和 dsPIC33FJ128MC202/802 器件:  
如果 AD12B = 1:  
 11 = 保留  
 10 = 保留  
 01 = 保留  
 00 = 保留

如果 AD12B = 0:  
 11 = 保留  
 10 = 保留  
 01 = CH1、 CH2 和 CH3 的反相输入为 VREF-  
 00 = CH1、 CH2 和 CH3 的反相输入为 VREF-

仅限 dsPIC33FJ32MC304、 dsPIC33FJ64MC204/804 和 dsPIC33FJ128MC204/804 器件:  
如果 AD12B = 1:  
 11 = 保留  
 10 = 保留  
 01 = 保留  
 00 = 保留

如果 AD12B = 0:  
 11 = 保留  
 10 = CH1 的反相输入为 AN6, CH2 的反相输入为 AN7, CH3 的反相输入为 AN8  
 01 = CH1、 CH2 和 CH3 的反相输入为 VREF-  
 00 = CH1、 CH2 和 CH3 的反相输入为 VREF-

bit 8      **CH123SB:** 采样多路开关 B 的通道 1、2 和 3 的同相输入选择位

如果 AD12B = 1:  
 1 = 保留  
 0 = 保留

如果 AD12B = 0:  
 1 = CH1 的同相输入为 AN3, CH2 的同相输入为 AN4, CH3 的同相输入为 AN5  
 0 = CH1 的同相输入为 AN0, CH2 的同相输入为 AN1, CH3 的同相输入为 AN2

bit 7-3      **未实现:** 读为 0

**寄存器 21-5: AD1CHS123: ADC1 输入通道 1、2 和 3 选择寄存器 (续)**

bit 2-1

**CH123NA<1:0>**: 采样多路开关 A 的通道 1、2 和 3 的反相输入选择位

仅限 dsPIC33FJ32MC302、 dsPIC33FJ64MC202/802 和 dsPIC33FJ128MC202/802 器件:

如果 AD12B = 1:

11 = 保留  
10 = 保留  
01 = 保留  
00 = 保留

如果 AD12B = 0:

11 = 保留  
10 = 保留  
01 = CH1、CH2 和 CH3 的反相输入为 VREF-  
00 = CH1、CH2 和 CH3 的反相输入为 VREF-

仅限 dsPIC33FJ32MC304、 dsPIC33FJ64MC204/804 和 dsPIC33FJ128MC204/804 器件:

如果 AD12B = 1:

11 = 保留  
10 = 保留  
01 = 保留  
00 = 保留

如果 AD12B = 0:

11 = 保留  
10 = CH1 的反相输入为 AN6, CH2 的反相输入为 AN7, CH3 的反相输入为 AN8  
01 = CH1、CH2 和 CH3 的反相输入为 VREF-  
00 = CH1、CH2 和 CH3 的反相输入为 VREF-

bit 0

**CH123SA**: 采样多路开关 A 的通道 1、2 和 3 的同相输入选择位

如果 AD12B = 1:

1 = 保留  
0 = 保留

如果 AD12B = 0:

1 = CH1 的同相输入为 AN3, CH2 的同相输入为 AN4, CH3 的同相输入为 AN5  
0 = CH1 的同相输入为 AN0, CH2 的同相输入为 AN1, CH3 的同相输入为 AN2

**寄存器 21-6: AD1CHS0: ADC1 输入通道 0 选择寄存器**

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB	—	—	CH0SB<4:0>				
bit 15							bit 8

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA	—	—	CH0SA<4:0>				
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            **CH0NB:** 采样多路开关 B 的通道 0 的反相输入选择位  
 1 = 通道 0 的反相输入为 AN1  
 0 = 通道 0 的反相输入为 VREF-
- bit 14-13        **未实现:** 读为 0
- 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  
 仅限 dsPIC33FJ32MC304、 dsPIC33FJ64MC204/804 和 dsPIC33FJ128MC204/804 器件:  
 01000 = 通道 0 的同相输入为 AN8  
 .  
 .  
 .  
 00010 = 通道 0 的同相输入为 AN2  
 00001 = 通道 0 的同相输入为 AN1  
 00000 = 通道 0 的同相输入为 AN0  
  
 仅限 dsPIC33FJ32MC302、 dsPIC33FJ64MC202/802 和 dsPIC33FJ128MC202/802 器件:  
 00101 = 通道 0 的同相输入为 AN5  
 .  
 .  
 .  
 00010 = 通道 0 的同相输入为 AN2  
 00001 = 通道 0 的同相输入为 AN1  
 00000 = 通道 0 的同相输入为 AN0
- bit 7            **CH0NA:** 采样多路开关 A 的通道 0 的反相输入选择位  
 1 = 通道 0 的反相输入为 AN1  
 0 = 通道 0 的反相输入为 VREF-
- bit 6-5        **未实现:** 读为 0

**寄存器 21-6: AD1CHS0: ADC1 输入通道 0 选择寄存器 (续)**

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

仅限 dsPIC33FJ32MC304、dsPIC33FJ64MC204/804 和 dsPIC33FJ128MC204/804 器件:

01000 = 通道 0 的同相输入为 AN8

•

•

•

00010 = 通道 0 的同相输入为 AN2

00001 = 通道 0 的同相输入为 AN1

00000 = 通道 0 的同相输入为 AN0

仅限 dsPIC33FJ32MC302、dsPIC33FJ64MC202/802 和 dsPIC33FJ128MC202/802 器件:

00101 = 通道 0 的同相输入为 AN5

•

•

•

00010 = 通道 0 的同相输入为 AN2

00001 = 通道 0 的同相输入为 AN1

00000 = 通道 0 的同相输入为 AN0

**寄存器 21-7: AD1CSSL: ADC1 输入扫描选择寄存器的低位字 (1)**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	CSS8
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
CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      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 8-0            **CSS<8:0>:** ADC 输入扫描选择位  
                     1 = 选择对 ANx 进行输入扫描  
                     0 = 输入扫描时跳过 ANx

**注 1:** 对于没有 9 路模拟输入的器件, 用户应用程序可选择所有的 AD1CSSL 位。但是, 如果选择了器件上没有的对应输入通道进行扫描, 则将转换 ADREF-。

**寄存器 21-8: AD1PCFGL: ADC1 端口配置寄存器的低位字 (1)**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PCFG8
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
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

**图注:**

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

bit 15-9            **未实现:** 读为 0  
 bit 8-0            **PCFG<8:0>:** ADC 端口配置控制位  
                     1 = 端口引脚处于数字模式, 使能端口读输入, ADC 输入多路开关连接到 AVss  
                     0 = 端口引脚处于模拟模式, 禁止端口读输入, ADC 采样引脚电压

**注 1:** 对于没有 9 路模拟输入的器件, 用户可读 / 写所有的 PCFG 位。但是, 如果器件不具备相应的输入, PCFG 位被忽略。



## 22.0 音频数模转换器 (DAC)

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 33. Audio Digital-to-Analog Converter (DAC)”，该文档可从 Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 下载。

音频数模转换器 (Digital-to-Analog Converter, DAC) 模块是为音频应用设计的 16 位  $\Delta$ - $\Sigma$  信号转换器。它具有左右两个输出声道，用于支持立体声应用。对于 dsPIC33FJ64MC804 和 dsPIC33FJ128MC804 器件，每个 DAC 输出声道提供三种电压输出：正 DAC 输出、负 DAC 输出和中点电压输出。dsPIC33FJ64MC802 和 dsPIC33FJ128MC802 器件提供正 DAC 输出电压和负 DAC 输出电压。正 DAC 输出和负 DAC 输出相对于约 1.65V 的中点电压成差分关系，用于驱动采用桥接式负载 (Bridge-Tied Load, BTL) 配置的扬声器。

### 22.1 关键特性

- 16 位分辨率 (14 位精度)
- 二阶数字  $\Delta$ - $\Sigma$  调制器
- 256 倍过采样比
- 128 抽头 FIR 电流控制型模拟重构滤波器
- 最大 100 ksp/s 采样速率
- 可由用户控制的采样时钟
- 输入频率最大为 45 kHz
- 差分模拟输出
- 信噪比: 90 dB
- 4 级深度输入缓冲区
- 16 位处理器 I/O 和 DMA 接口

### 22.2 DAC 模块工作原理

图 22-1 给出了音频 DAC 模块的功能框图。音频 DAC 模块为每个输出声道提供 4 级深度的数据输入 FIFO 缓冲区。如果 DMA 模块和 / 或处理器不能及时提供输出数据，使 FIFO 变为空，DAC 将从 DAC 默认数据寄存器 (DACDFLT) 接收数据。这种安全特性对于由 DAC 输出控制重要处理器或机器的工业控制应用很有用。应该用“安全”的输出值对 DACDFLT 寄存器进行初始化。安全的输出值通常为中点值 (0x8000) 或零值 (0x0000)。

数字插值器对输入信号进行上采样，采用的过采样比为 256 倍，在用户提供的数据点之间生成数据点。插值器还包含由数字滤波器进行的处理，来提供“噪声整形”来将转换器的噪声转移到高于 20 kHz (通带的上限) 频段。插值器的输出驱动  $\Sigma$ - $\Delta$  调制器，来自  $\Sigma$ - $\Delta$  调制器的串行数据比特流由重构滤波器进行处理。重构滤波器的差分输出由运放进行放大，向 1k 欧姆负载提供所需的 2V 峰-峰电压摆幅。

### 22.3 DAC 输出格式

DAC 输出数据流可为二进制补码有符号数格式或无符号数格式。

音频 DAC 模块能接收二进制补码有符号数格式或无符号数格式的 16 位输入数据。数据格式由 DAC1CON 寄存器中的数据格式控制 (FORM<8>) 位控制。支持的格式有：

- 1 = 有符号 (二进制补码格式)
- 0 = 无符号

如果 FORM 位配置为“无符号数”，用户输入数据具有如下特征：

- 0xFFFF = 最大正输出电压
- 0x8000 = 中点输出电压
- 0x7FFF = 低于中点电压的最大值
- 0x0000 = 最小输出电压

如果 FORM 位配置为“有符号数”，用户输入数据具有如下特征：

- 0x7FFF = 最大正输出电压
- 0x0000 = 中点输出电压
- 0xFFFF = 低于中点电压的最大值
- 0x8000 = 最小输出电压

音频 DAC 提供与数字输入值成比例的模拟输出。每秒 10 万次采样 (100 ksp/s) 的最大更新速率可提供高品质的音频再现。

音频 DAC 提供差分模拟输出，供电电压为 3.3V 时其共模输出电压的标称值为 1.65V。电压摆幅约为相对于 1.65V 中点电压  $\pm 1V$ ，或者向 1k 欧姆负载提供约 0.65V 到 2.65V 的电压。

### 22.4 DAC 时钟

DAC 时钟信号为音频 DAC 模块的内部逻辑提供时钟。音频 DAC 的数据采样速率是 DAC 时钟频率的整数倍。DAC 时钟由时钟分频电路产生，时钟分频电路接受来自附属振荡器的附属时钟。分频比通过 DAC 控制寄存器

(DAC1CON) 中的时钟分频位 (DACFDIV<6:0>) 设定。产生的 DAC 时钟不能超过 25.6 MHz。如果使用较低的采样速率，则可降低 DAC 滤波器的时钟频率来降低功耗。DAC 时钟频率为采样频率的 256 倍。

图 22-1: 音频数模转换器 (DAC) 的框图

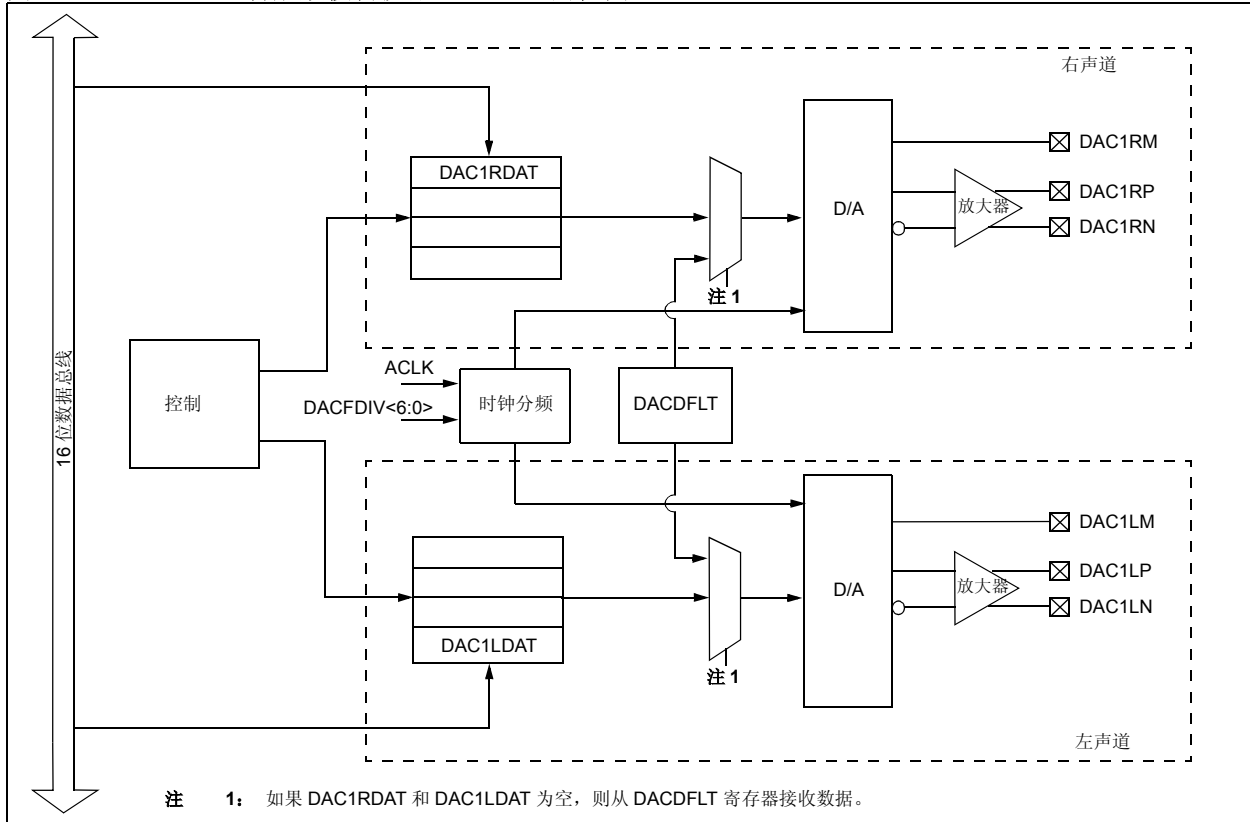
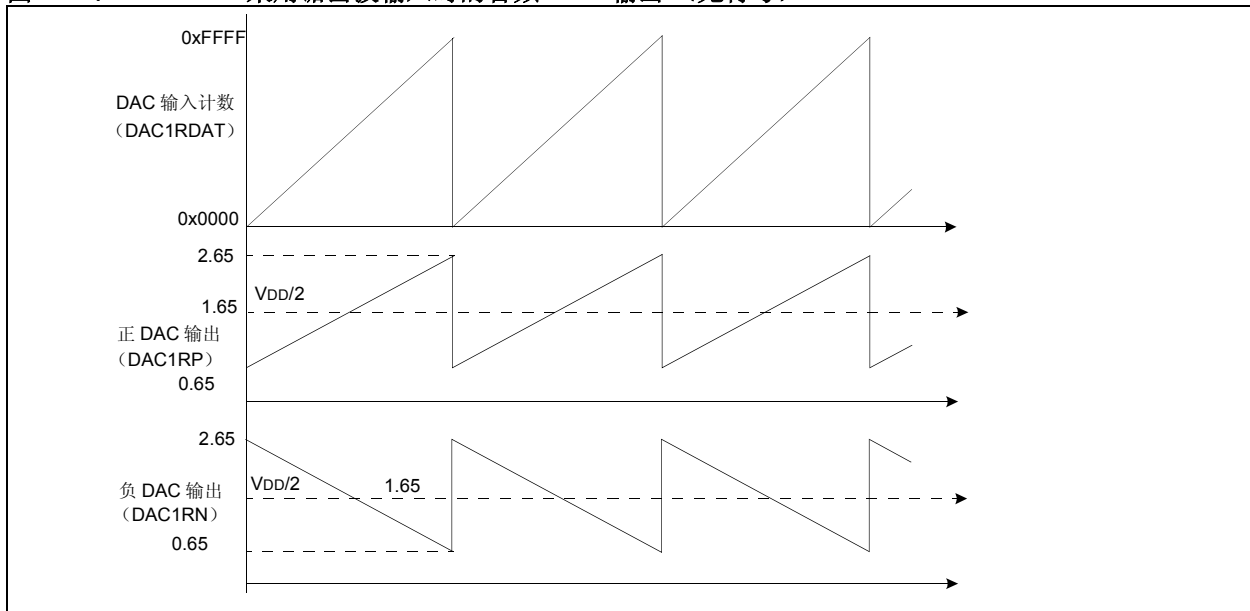


图 22-2: 采用锯齿波输入时的音频 DAC 输出 (无符号)



寄存器 22-1: DAC1CON: DAC 控制寄存器

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
DACEN	—	DACSIDL	AMPON	—	—	—	FORM
bit 15							bit 8
U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1
—	DACFDIV<6:0>						—
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            **DACEN:** DAC1 使能位  
                  1 = 使能模块  
                  0 = 禁止模块
- bit 14            **未实现:** 读为 0
- bit 13            **DACSIDL:** 在空闲模式下停止位  
                  1 = 当器件进入空闲模式时, 模块停止工作  
                  0 = 在空闲模式下模块继续工作
- bit 12            **AMPON:** 在休眠 / 空闲模式下使能模拟输出放大器的位  
                  1 = 在休眠 / 空闲模式下使能模拟输出放大器  
                  0 = 在休眠 / 空闲模式下禁止模拟输出放大器
- bit 11-9         **未实现:** 读为 0
- bit 8             **FORM:** 数据格式选择位  
                  1 = 有符号整数  
                  0 = 无符号整数
- bit 7             **未实现:** 读为 0
- bit 6-0          **DACFDIV<6:0>:** DAC 时钟分频比:  
                  11111111 = 将输入时钟 128 分频  
                  .  
                  .  
                  .  
                  0000101 = 将输入时钟 6 分频 (默认)  
                  .  
                  .  
                  .  
                  0000010 = 将输入时钟 3 分频  
                  0000001 = 将输入时钟 2 分频  
                  0000000 = 将输入时钟 1 分频 (不分频)

**寄存器 22-2: DAC1STAT: DAC 状态寄存器**

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R-0	R-0
LOEN	—	LMVOEN	—	—	LITYPE	LFULL	LEMPY
bit 15						bit 8	

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R-0	R-0
ROEN	—	RMVOEN	—	—	RITYPE	RFULL	REMPY
bit 7						bit 0	

**图注:**

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

- bit 15        **LOEN:** 左声道 DAC 输出使能位  
 1 = 使能正、负 DAC 输出  
 0 = 禁止 DAC 输出
- bit 14        **未实现:** 读为 0
- bit 13        **LMVOEN:** 左声道中点 DAC 输出电压使能位  
 1 = 使能中点 DAC 输出  
 0 = 禁止中点 DAC 输出
- bit 12-11    **未实现:** 读为 0
- bit 10        **LITYPE:** 左声道中断类型位  
 1 = 如果 FIFO 为空产生中断  
 0 = 如果 FIFO 非满产生中断
- bit 9         **LFULL:** 左声道数据输入 FIFO 满状态位  
 1 = FIFO 已满  
 0 = FIFO 未满
- bit 8         **LEMPY:** 左声道数据输入 FIFO 空状态位  
 1 = FIFO 为空  
 0 = FIFO 非空
- bit 7         **ROEN:** 右声道 DAC 输出使能位  
 1 = 使能正、负 DAC 输出  
 0 = 禁止 DAC 输出
- bit 6         **未实现:** 读为 0
- bit 5         **RMVOEN:** 右声道中点 DAC 输出电压使能位  
 1 = 使能中点 DAC 输出  
 0 = 禁止中点 DAC 输出
- bit 4-3       **未实现:** 读为 0
- bit 2         **RITYPE:** 右声道中断类型位  
 1 = 如果 FIFO 为空产生中断  
 0 = 如果 FIFO 非满产生中断
- bit 1         **RFULL:** 右声道数据输入 FIFO 满状态位  
 1 = FIFO 已满  
 0 = FIFO 未满
- bit 0         **REMPY:** 右声道数据输入 FIFO 空状态位  
 1 = FIFO 为空  
 0 = FIFO 非空

**寄存器 22-3: DAC1DFLT: DAC 默认数据寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DAC1DFLT<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	R/W-0
DAC1DFLT<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 15-0            **DACDFLT:** DAC 默认值

**寄存器 22-4: DAC1LDAT: DAC 左声道数据寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DAC1LDAT<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	R/W-0
DAC1LDAT<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 15-0            **DACL DAT:** 左声道数据端口

**寄存器 22-5: DAC1RDAT: DAC 右声道数据寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DAC1RDAT<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	R/W-0
DAC1RDAT<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 15-0            **DACRDAT:** 右声道数据端口

注:

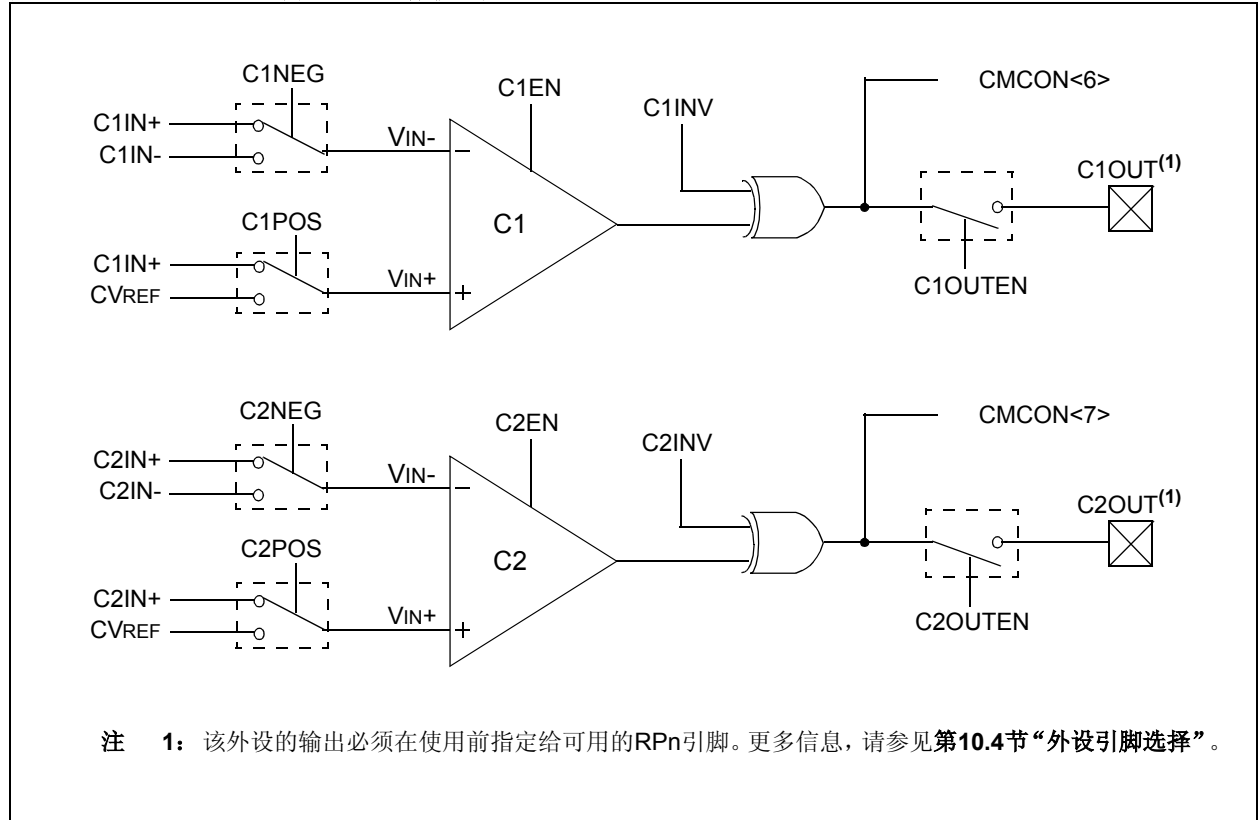
### 23.0 比较器模块

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 34. Comparator”，该文档可从 Microchip 网站 (www.microchip.com) 下载。

比较器模块提供了一组双输入比较器。比较器的输入可以配置为使用 4 个输入引脚之一 (C1IN+、C1IN-、C2IN+ 和 C2IN-)，以及比较器参考电压输入 (CVREF)。

**注：** 该外设包含输出功能，可能需要通过外设引脚选择特性对比较器的输出功能进行配置。更多信息，请参见第 10.4 节“外设引脚选择”。

图 23-1: 比较器 I/O 工作模式



**寄存器 23-1: CMCON: 比较器控制寄存器**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMIDL	—	C2EVT	C1EVT	C2EN	C1EN	C2OUTEN <sup>(1)</sup>	C1OUTEN <sup>(2)</sup>
bit 15							bit 8

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
C2OUT	C1OUT	C2INV	C1INV	C2NEG	C2POS	C1NEG	C1POS
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      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:** 空闲模式停止位  
 1 = 当器件进入空闲模式时, 模块不产生中断。模块仍然使能。  
 0 = 在空闲模式下模块继续正常工作
- bit 14        **未实现:** 读为 0
- bit 13        **C2EVT:** 比较器 2 事件位  
 1 = 比较器输出改变了状态  
 0 = 比较器输出未改变状态
- bit 12        **C1EVT:** 比较器 1 事件位  
 1 = 比较器输出改变了状态  
 0 = 比较器输出未改变状态
- bit 11        **C2EN:** 比较器 2 使能位  
 1 = 比较器使能  
 0 = 比较器禁止
- bit 10        **C1EN:** 比较器 1 使能位  
 1 = 比较器使能  
 0 = 比较器禁止
- bit 9         **C2OUTEN:** 比较器 2 输出使能位 <sup>(1)</sup>  
 1 = 比较器输出连接到输出焊盘  
 0 = 比较器输出不连接到输出焊盘
- bit 8         **C1OUTEN:** 比较器 1 输出使能位 <sup>(2)</sup>  
 1 = 比较器输出连接到输出焊盘  
 0 = 比较器输出不连接到输出焊盘
- bit 7         **C2OUT:** 比较器 2 输出位  
 当 C2INV = 0 时:  
 1 = C2 VIN+ > C2 VIN-  
 0 = C2 VIN+ < C2 VIN-  
 当 C2INV = 1 时:  
 0 = C2 VIN+ > C2 VIN-  
 1 = C2 VIN+ < C2 VIN-

- 注 1: 如果 C2OUTEN = 1, 则 C2OUT 外设输出必须配置给可用的 RPx 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”
- 2: 如果 C1OUTEN = 1, 则 C1OUT 外设输出必须配置给可用的 RPx 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”



寄存器 23-1: **CMCON: 比较器控制寄存器 (续)**

bit 6	<p><b>C1OUT:</b> 比较器 1 输出位</p> <p>当 <b>C1INV = 0</b> 时:</p> <p>1 = C1 VIN+ &gt; C1 VIN-</p> <p>0 = C1 VIN+ &lt; C1 VIN-</p> <p>当 <b>C1INV = 1</b> 时:</p> <p>0 = C1 VIN+ &gt; C1 VIN-</p> <p>1 = C1 VIN+ &lt; C1 VIN-</p>
bit 5	<p><b>C2INV:</b> 比较器 2 输出反相位</p> <p>1 = C2 输出反相</p> <p>0 = C2 输出不反相</p>
bit 4	<p><b>C1INV:</b> 比较器 1 输出反相位</p> <p>1 = C1 输出反相</p> <p>0 = C1 输出不反相</p>
bit 3	<p><b>C2NEG:</b> 比较器 2 反相输入配置位</p> <p>1 = 输入连接到 VIN+</p> <p>0 = 输入连接到 VIN-</p> <p>请参见图 23-1 了解比较器模式。</p>
bit 2	<p><b>C2POS:</b> 比较器 2 同相输入配置位</p> <p>1 = 输入连接到 VIN+</p> <p>0 = 输入连接到 CVREF</p> <p>请参见图 23-1 了解比较器模式。</p>
bit 1	<p><b>C1NEG:</b> 比较器 1 反相输入配置位</p> <p>1 = 输入连接到 VIN+</p> <p>0 = 输入连接到 VIN-</p> <p>请参见图 23-1 了解比较器模式。</p>
bit 0	<p><b>C1POS:</b> 比较器 1 同相输入配置位</p> <p>1 = 输入连接到 VIN+</p> <p>0 = 输入连接到 CVREF</p> <p>请参见图 23-1 了解比较器模式。</p>

- 注 1: 如果 C2OUTEN = 1, 则 C2OUT 外设输出必须配置给可用的 RPx 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”
- 注 2: 如果 C1OUTEN = 1, 则 C1OUT 外设输出必须配置给可用的 RPx 引脚。更多信息, 请参见第 10.4 节 “外设引脚选择”

### 23.1 比较器参考电压

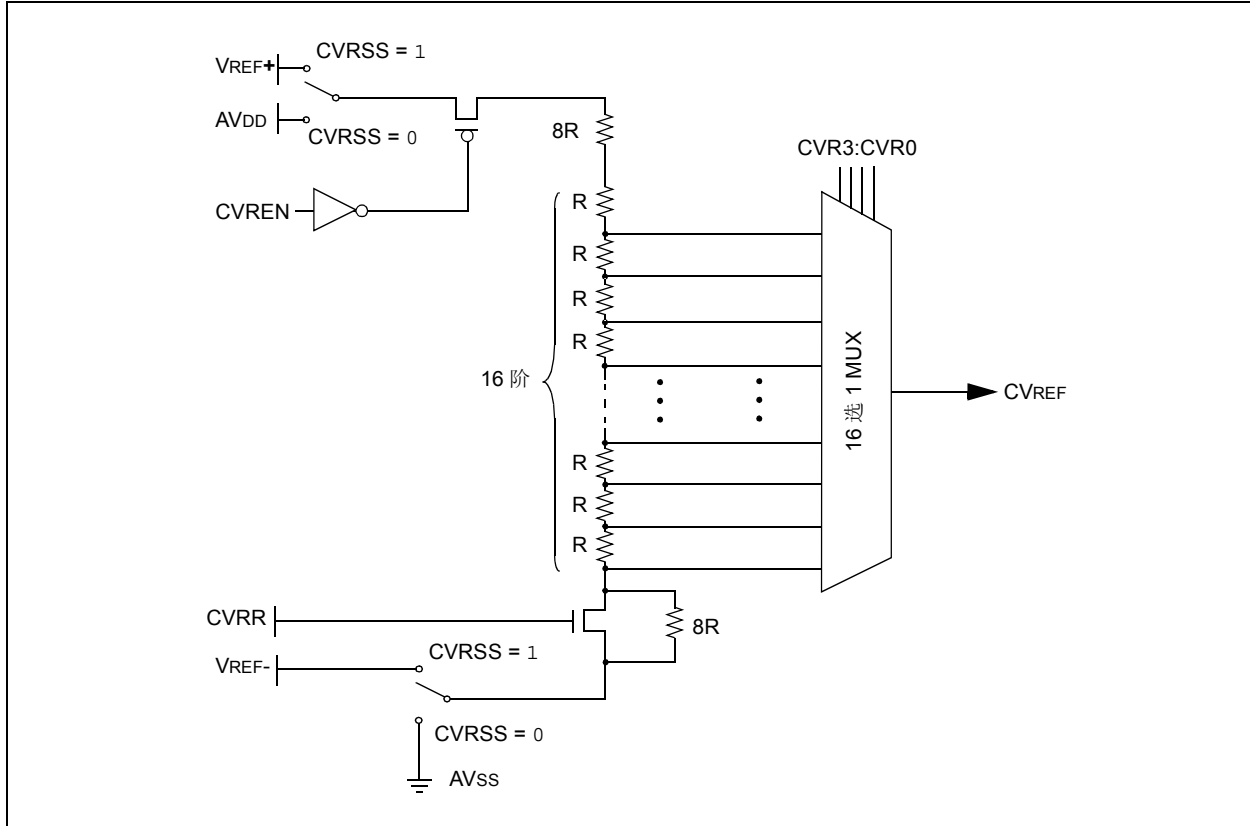
#### 23.1.1 配置比较器参考电压

参考电压模块是通过 CVRCON 寄存器（寄存器 23-2）控制的。比较器参考电压模块提供两种范围的输出电压，每种范围都具有 16 个不同的电平。通过 CVRR 位（CVRCON<5>）选择使用的输出电压范围。这两种范围的主要区别在于其电压值之间的步长不同（其中一种范围可提供较高的分辨率），该步长由 CVREF 选择位（CVR3:CVR0）进行选择。

比较器参考电压源可以来自 VDD 和 VSS，也可以来自外部 VREF+ 和 VREF-。通过 CVRSS 位（CVRCON<4>）选择参考电压源。

在更改 CVREF 输出时必须考虑比较器参考电压的稳定时间。

图 23-2: 比较器参考电压框图



寄存器 23-2: **CVRCON: 比较器参考电压控制寄存器**

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE	CVRR	CVRSS	CVR<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 15-8      **未实现:** 读为 0
- bit 7        **CVREN:** 比较器参考电压使能位  
 1 = CVREF 电路上电  
 0 = CVREF 电路断电
- bit 6        **CVROE:** 比较器 VREF 输出使能位  
 1 = CVREF 电压输出到 CVREF 引脚  
 0 = CVREF 电压与 CVREF 引脚断开连接
- bit 5        **CVRR:** 比较器 VREF 范围选择位  
 1 = CVRSRC 范围应为从 0 至 0.625 CVRSRC, 步长为 CVRSRC/24  
 0 = CVRSRC 范围应为从 0.25 至 0.719 CVRSRC, 步长为 CVRSRC/32
- bit 4        **CVRSS:** 比较器 VREF 源选择位  
 1 = 比较器参考电压源 CVRSRC = VREF+ – VREF-  
 0 = 比较器参考电压源 CVRSRC = AVDD – AVSS
- bit 3-0      **CVR<3:0>:** 比较器 VREF 值选择位,  $0 \leq \text{CVR}<3:0> \leq 15$   
 当 CVRR = 1 时:  
 $\text{CVREF} = (\text{CVR}<3:0>/24) \cdot (\text{CVRSRC})$   
 当 CVRR = 0 时:  
 $\text{CVREF} = 1/4 \cdot (\text{CVRSRC}) + (\text{CVR}<3:0>/32) \cdot (\text{CVRSRC})$

注:

## 24.0 实时时钟和日历 (RTCC)

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 37. Real-Time Clock and Calendar (RTCC)”，该文档可从 Microchip 网站 (www.microchip.com) 下载。

本章讨论 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件提供的实时时钟和日历 (RTCC) 模块及其操作。下面列出了该模块的一些关键特性：

- 时间：小时、分钟和秒
- 24 小时格式（军用时间）
- 日历：年、月、日和星期

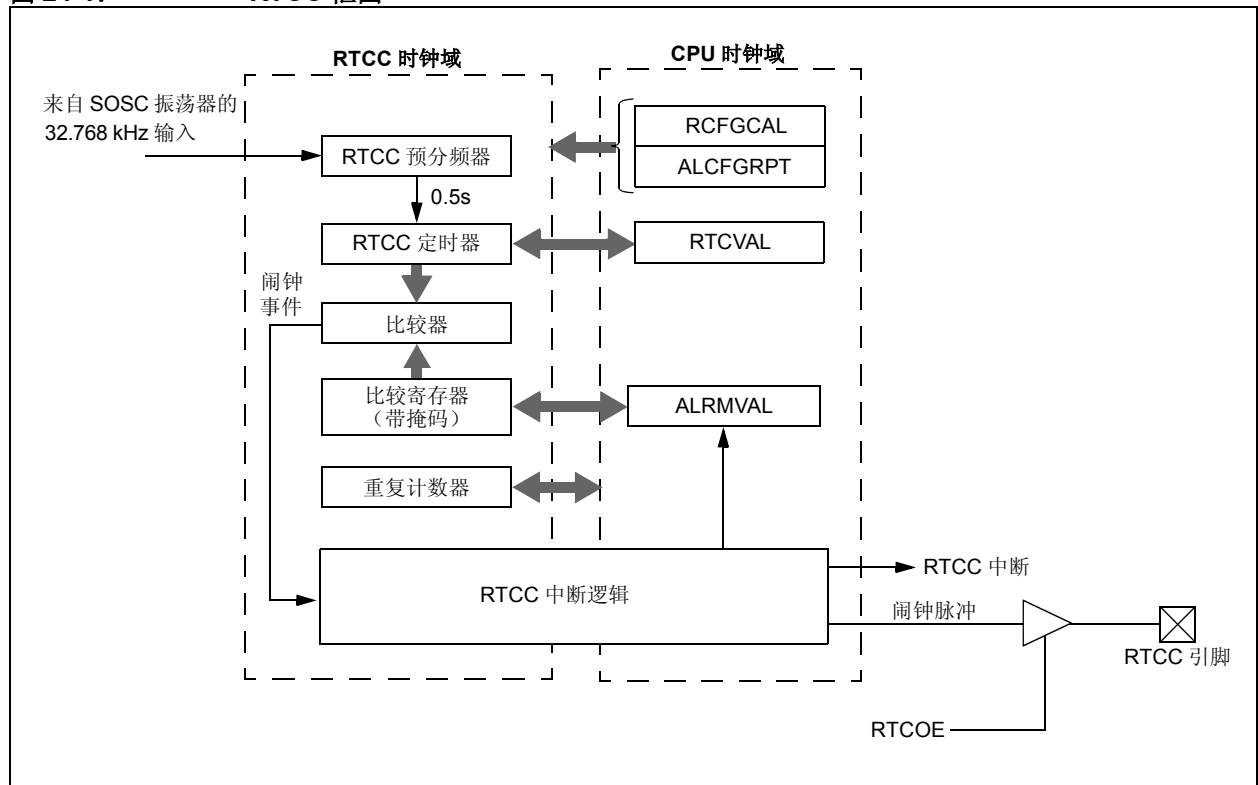
- 可配置闹钟
- 年份范围：2000 至 2099
- 闰年修正
- 用于获得紧凑固件的 BCD 格式
- 为低功耗操作进行了优化
- 带自动调节的用户校准
- 校准范围：每月  $\pm 2.64$  秒误差
- 要求：外部提供 32.768 kHz 时钟晶振
- RTCC 引脚上的闹钟脉冲或秒时钟输出

RTCC 模块是为需要长时间维持精确时间的应用设计的，无需或极少需要 CPU 干预。RTCC 模块为低功耗使用作了优化，以便在跟踪时间时延长电池寿命。

RTCC 模块是百年时钟和日历，带有闰年自动检测功能。时钟范围为从 2000 年 1 月 1 日 00:00:00（午夜）到 2099 年 12 月 31 日 23:59:59。

小时以 24 小时（军用时间）格式提供。该时钟提供一秒的时间粒度，用户可看到半秒的时间间隔。

图 24-1: RTCC 框图



## 24.1 RTCC 模块寄存器

RTCC 模块寄存器可分为以下三类：

- RTCC 控制寄存器
- RTCC 值寄存器
- 闹钟值寄存器

### 24.1.1 寄存器映射

为限制寄存器接口数量，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 RTCPTR 位 (RCFGCAL<9:8>) 选择所需定时器寄存器对 (见表 24-1)。

通过写 RTCVALH 字节，RTCC 指针值 RTCPTR<1:0> 位递减 1，直到达到 00。一旦达到 00，将可通过 RTCVALH 和 RTCVALL 访问分钟和秒值，直到手动更改指针值。

**表 24-1: 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	秒
01	星期	小时
10	月	日
11	—	年

闹钟值寄存器窗口 (ALRMVALH 和 ALRMVALL) 使用 ALRMPTR 位 (ALCFGRPT<9:8>) 选择所需闹钟寄存器对 (见表 24-2)。

### 例 24-1: 将 RTCWREN 位置 1

```

MOV    #NVMKEY, W1      ;move the address of NVMKEY into W1
MOV    #0x55, W2
MOV    #0xAA, W3
MOV    W2, [W1]         ;start 55/AA sequence
MOV    W3, [W1]
BSET   RCFGCAL, #13     ;set the RTCWREN bit
    
```

通过写 ALRMVALH 字节，闹钟指针值 ALRMPTR<1:0> 位递减 1，直到达到 00。一旦达到 00，将可通过 ALRMVALH 和 ALRMVALL 访问 ALRMMIN 和 ALRMSEC 值，直到手动更改指针值。

**表 24-2: ALRMVAL 寄存器映射**

ALRMPTR <1:0>	闹钟值寄存器窗口	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

考虑到 16 位内核并不区分 8 位和 16 位读操作，用户必须注意，读 ALRMVALH 或 ALRMVALL 字节时都会使 ALRMPTR<1:0> 值递减 1。这同样也适用于 RTCVALH 或 RTCVALL 字节，读取它们时会使 RTCPTR<1:0> 递减 1。

**注：** 这只适用于读操作，不适用于写操作。

### 24.1.2 写锁定

要对任何 RTCC 定时器寄存器执行写操作，必须先将 RTCWREN 位 (RCFGCAL<13>) 置 1 (见例 24-1)。

**注：** 为避免意外写入定时器，建议其他任何时候 RTCWREN 位 (RCFGCAL<13>) 都保持清零。要将 RTCWREN 位置 1，在 55h/AA 序列和 RTCWREN 置 1 之间只允许 1 个指令周期的时间段；因此，建议遵循例 24-1 中的顺序编写代码。

寄存器 24-1: RCFGAL: RTCC 校准和配置寄存器 (1)

R/W-0	U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0
RTCEN <sup>(2)</sup>	—	RTCWREN	RTCSYNC	HALFSEC <sup>(3)</sup>	RTCOE	RTCPTR<1:0>	
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
CAL<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            **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 寄存器可由用户写入  
 0 = RTCVALH 和 RTCVALL 寄存器已被锁定, 不可由用户写入
- bit 12            **RTCSYNC:** RTCC 值寄存器读同步位  
 1 = 由于出现计满返回, 在读 RTCVALH、RTCVALL 和 ALCFGRPT 寄存器过程中这些寄存器的值可能改变, 从而导致读取的数据无效。如果两次读取寄存器得到的数据相同, 可认为数据是有效的。  
 0 = RTCVALH、RTCVALL 或 ALCFGRPT 寄存器在读取时无需考虑计满返回
- bit 11            **HALFSEC:** 半秒状态位 <sup>(3)</sup>  
 1 = 一秒的后半秒  
 0 = 一秒的前半秒
- 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  
 读取 RTCVALH 和 RTCVALL 寄存器时, 指向相应的 RTCC 值寄存器; 每当读或写 RTCVALH 时 RTCPTR<1:0> 的值就递减 1, 直到达到 00。  
RTCVAL<15:8>:  
 00 = 分钟  
 01 = 星期  
 10 = 月  
 11 = 保留  
RTCVAL<7:0>:  
 00 = 秒  
 01 = 小时  
 10 = 日  
 11 = 年

- 注 1: RCFGAL 寄存器仅受 POR 影响。  
 2: 仅当 RTCWREN = 1 时才允许写入 RTCEN 位。  
 3: 该位是只读的。写入 MINSEC 寄存器的低半部分时, 它被清零。

**寄存器 24-1: RCFGAL: RTCC 校准和配置寄存器<sup>(1)</sup> (续)**

bit 7-0      **CAL<7:0>**: RTC 漂移校准位

01111111 = 最大正向调整; 每分钟增加 508 个 RTC 时钟脉冲

•

•

•

01111111 = 最小正向调整; 每分钟增加 4 个 RTC 时钟脉冲

00000000 = 无调整

11111111 = 最小负向调整; 每分钟减少 4 个 RTC 时钟脉冲

•

•

•

10000000 = 最大负向调整; 每分钟减少 512 个 RTC 时钟脉冲

- 注 1:** RCFGAL 寄存器仅受 POR 影响。
- 2:** 仅当 RTCWREN = 1 时才允许写入 RTCEN 位。
- 3:** 该位是只读的。写入 MINSEC 寄存器的低半部分时, 它被清零。



寄存器 24-2: 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	R/W-0	R/W-0
—	—	—	—	—	—	RTSECSEL <sup>(1)</sup>	PMP TTL
bit 7						bit 0	

图注:

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

- bit 15-2      **未实现:** 读为 0
- bit 1        **RTSECSEL:** RTCC 秒时钟输出选择位 <sup>(1)</sup>
  - 1 = 选择 RTCC 引脚输出 RTCC 秒时钟
  - 0 = 选择 RTCC 引脚输出 RTCC 闹钟
- bit 0        **PMP TTL:** PMP 模块 TTL 输入缓冲器选择位
  - 1 = PMP 模块使用 TTL 输入缓冲器
  - 0 = PMP 模块使用施密特触发器输入缓冲器

注 1: 要启用实际 RTCC 输出, 需要将 RTCOE (RCFGCAL) 位置 1。

**寄存器 24-3: 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	AMASK<3:0>			ALRMPTR<1:0>		
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
ARPT<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        **ALRMEN:** 闹钟使能位  
 1 = 使能闹钟 (每当 ARPT<7:0> = 00h 且 CHIME = 0 时, 发生闹钟事件后都自动清零)  
 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 就停止
- bit 13-10    **AMASK<3:0>:** 闹钟掩码配置位  
 0000 = 每半秒  
 0001 = 每秒  
 0010 = 每 10 秒  
 0011 = 每分钟  
 0100 = 每 10 分钟  
 0101 = 每小时  
 0110 = 一天一次  
 0111 = 一周一次  
 1000 = 一月一次  
 1001 = 一年一次 (配置在 2 月 29 日时例外, 这时每 4 年一次)  
 101x = 保留——不要使用  
 11xx = 保留——不要使用
- bit 9-8        **ALRMPTR<1:0>:** 闹钟值寄存器窗口指针位  
 读取 ALRMVALH 和 ALRMVALL 寄存器时, 指向相应的闹钟值寄存器; 每当读或写 ALRMVALH 时 ALRMPTR<1:0> 的值就递减 1, 直到达到 00。  
ALRMVAL<15:8>:  
 00 = ALRMMIN  
 01 = ALRMWD  
 10 = ALRMMNTH  
 11 = 未实现  
ALRMVAL<7:0>:  
 00 = ALRMSEC  
 01 = ALRMHR  
 10 = ALRMDAY  
 11 = 未实现
- bit 7-0        **ARPT<7:0>:** 闹钟重复计数器值位  
 11111111 = 闹钟将再重复 255 次  
 .  
 .  
 .  
 00000000 = 闹钟将不重复  
 每当发生闹钟事件时计数器就递减 1。除非 CHIME = 1, 否则计数器不能从 00h 返回到 FFh。

寄存器 24-4: **RTCVAL** (当 **RTCPtr<1:0> = 11** 时): 年份值寄存器<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	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
YRTEN<3:0>				YRONE<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 15-8            **未实现:** 读为 0  
 bit 7-4            **YRTEN<3:0>:** 年份的十位数的二 - 十进制码 (Binary Coded Decimal, BCD) 值; 值为 0 到 9  
 bit 3-0            **YRONE<3:0>:** 年份的个位数的 BCD 值; 值为 0 到 9

**注 1:** 仅当 **RTCWREN = 1** 时才允许写入年份寄存器。

寄存器 24-5: **RTCVAL** (当 **RTCPtr<1:0> = 10** 时): 月和日值寄存器<sup>(1)</sup>

U-0	U-0	U-0	R-x	R-x	R-x	R-x	R-x
—	—	—	MHTTEN0	MTHONE<3:0>			
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
—	—	DAYTEN<1:0>		DAYONE<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           **未实现:** 读为 0  
 bit 12            **MHTTEN0:** 月份的十位数的 BCD 值; 值为 0 或 1  
 bit 11-8          **MTHONE<3:0>:** 月份的个位数的 BCD 值; 值为 0 到 9  
 bit 7-6            **未实现:** 读为 0  
 bit 5-4            **DAYTEN<1:0>:** 日的十位数的 BCD 值; 值为 0 到 3  
 bit 3-0            **DAYONE<3:0>:** 日的个位数的 BCD 值; 值为 0 到 9

**注 1:** 仅当 **RTCWREN = 1** 时才允许写入该寄存器。

**寄存器 24-6: RTCVAL (当 RTCPTR<1:0> = 01 时): 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
—	—	—	—	—	WDAY<2:0>		
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
—	—	HRTEN<1:0>		HRONE<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 15-11      未实现: 读为 0
- bit 10-8      **WDAY<2:0>**: 星期的 BCD 值; 值为 0 到 6
- bit 7-6        未实现: 读为 0
- bit 5-4        **HRTEN<1:0>**: 小时的十位数的 BCD 值; 值为 0 到 2
- bit 3-0        **HRONE<3:0>**: 小时的个位数的 BCD 值; 值为 0 到 9

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

**寄存器 24-7: RTCVAL (当 RTCPTR<1:0> = 00 时): 分钟和秒值寄存器**

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN<2:0>			MINONE<3:0>			
bit 15					bit 8		
U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN<2:0>			SECONE<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        未实现: 读为 0
- bit 14-12    **MINTEN<2:0>**: 分钟的十位数的 BCD 值; 值为 0 到 5
- bit 11-8     **MINONE<3:0>**: 分钟的个位数的 BCD 值; 值为 0 到 9
- bit 7         未实现: 读为 0
- bit 6-4      **SECTEN<2:0>**: 秒的十位数的 BCD 值; 值为 0 到 5
- bit 3-0      **SECONE<3:0>**: 秒的个位数的 BCD 值; 值为 0 到 9

寄存器 24-8: ALRMVAL (当 ALRMPTR<1:0> = 10 时): 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	MTHONE<3:0>				
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	
—	—	DAYTEN<1:0>		DAYONE<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      未实现: 读为 0
- bit 12        **MHTTEN0**: 月份的十位数的 BCD 值; 值为 0 或 1
- bit 11-8      **MTHONE<3:0>**: 月份的个位数的 BCD 值; 值为 0 到 9
- bit 7-6        未实现: 读为 0
- bit 5-4        **DAYTEN<1:0>**: 日的十位数的 BCD 值; 值为 0 到 3
- bit 3-0        **DAYONE<3:0>**: 日的个位数的 BCD 值; 值为 0 到 9

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

寄存器 24-9: ALRMVAL (当 ALRMPTR<1:0> = 01 时): 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	
—	—	HRTEN<1:0>		HRONE<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 15-11     未实现: 读为 0
- bit 10-8     **WDAY<2:0>**: 星期的 BCD 值; 值为 0 到 6
- bit 7-6       未实现: 读为 0
- bit 5-4       **HRTEN<1:0>**: 小时的十位数的 BCD 值; 值为 0 到 2
- bit 3-0       **HRONE<3:0>**: 小时的个位数的 BCD 值; 值为 0 到 9

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

**寄存器 24-10: ALRMVAL (当 ALRMPTR<1:0> = 00 时): 闹钟分钟和秒值寄存器**

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN<2:0>			MINONE<3:0>			
bit 15							bit 8

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN<2:0>			SECONE<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                      **未实现:** 读为 0
- bit 14-12                **MINTEN<2:0>:** 分钟的十位数的 BCD 值; 值为 0 到 5
- bit 11-8                 **MINONE<3:0>:** 分钟的个位数的 BCD 值; 值为 0 到 9
- bit 7                      **未实现:** 读为 0
- bit 6-4                 **SECTEN<2:0>:** 秒的十位数的 BCD 值; 值为 0 到 5
- bit 3-0                 **SECONE<3:0>:** 秒的个位数的 BCD 值; 值为 0 到 9

## 24.2 校准

实时晶振输入可用周期性自动调节功能校准。正确校准后，RTCC 可提供小于每月 3 秒的误差。这是通过找到误差时钟数，将该值存储到 RCFGAL 寄存器的低字节实现的。装入 RCFGAL 低字节的 8 位有符号值乘以 4，每分钟一次加到 RTCC 定时器或从该定时器中减去。关于 RTCC 校准，请参见以下步骤：

1. 用户必须通过器件上的其他定时器资源找出 32.768 kHz 晶振的误差。
2. 知道误差后，必须将它转换为每分钟误差时钟脉冲数。

### 公式 24-1: 时钟频率

$$\begin{aligned} (\text{理想频率} \pm \text{测量频率}) \cdot 60 &= \text{每分钟误差时钟数} \\ \pm \text{理想频率} &= 32,768 \text{ Hz} \end{aligned}$$

3. a) 如果振荡器频率快于理想频率（从步骤 2 得到负的结果），RCFGAL 寄存器值必须为负。这导致每分钟从定时器计数器减去指定的时钟脉冲数一次。  
b) 如果振荡器频率慢于理想频率（从步骤 2 得到正的结果），则 RCFGAL 寄存器值必须为正。这导致每分钟向定时器计数器加上指定的时钟脉冲数一次。
4. 将每分钟误差时钟数除以 4 得到正确的 CAL 值，并将正确的值装入 RCFGAL 寄存器。

（CAL 中每 1 位递增会增加或减去 4 个脉冲）。

只有在定时器关闭时或紧接秒脉冲上升沿后，才应发生对 RCFGAL 寄存器低字节的写入。

**注：** 是否在误差值中包含晶振初始误差、温度造成的漂移和晶振老化造成的漂移，由用户自行决定。

## 24.3 闹钟

- 可在半秒到一年的范围内配置
- 使用 ALRMEN 位（寄存器 24-3 的 ALCFGRPT<7>）使能
- 具有一次性闹钟和重复闹钟选项

### 24.3.1 配置闹钟

闹钟功能用 ALRMEN 位使能。发出闹钟后该位清零。仅当 ALRMEN = 0 时，才应发生对 ALRMVALH:ALRMVALL 的写操作。

闹钟的间隔选择通过 AMASK 位（ALCFGRPT<13:10>）配置，如图 24-2 所示。这些位确定了触发闹钟，闹钟的哪些位、多少位必须和时钟值匹配。也可以将闹钟配置为根据预先配置的间隔重复。闹钟使能后发生的总次数存储到 ALCFGRPT 寄存器的低字节。

当 ALCFGRPT = 00 且 CHIME 位（ALCFGRPT<14>）= 0 时，重复功能被禁止，只发生单次闹钟。通过将 FFh 装入 ALCFGRPT 寄存器的低字节，闹钟可重复最多 255 次。

每次发出闹钟后，ALCFGRPT 寄存器都递减 1。寄存器达到 00 后，将最后一次发出闹钟，此后 ALRMEN 位将自动清零，闹钟将关闭。如果 CHIME 位 = 1，闹钟可以无限重复。当 CHIME = 1 时，ALCFGRPT 寄存器达到 00 时不会禁止闹钟，而是返回到 FF，继续无限计数。

### 24.3.2 闹钟中断

每个闹钟事件发生时，都会产生中断。此外会提供闹钟脉冲输出，其频率是闹钟频率的一半。该输出与 RTCC 时钟完全同步，可用作其他外设的触发时钟。

**注：** 闹钟使能时（ALRMEN = 1），更改除 RCFGAL 和 ALCFGRPT 寄存器以外的任何寄存器以及 CHIME 位，都会导致错误闹钟事件，进而导致错误的闹钟中断。为避免错误闹钟事件，只应在闹钟禁止时（ALRMEN = 0）更改定时器值和闹钟值。建议在 RTCSYNC = 0 时更改 ALCFGRPT 寄存器和 CHIME 位。

图 24-2: 闹钟掩码设置

闹钟掩码设置 (AMASK3:AMASK0)	星期	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——每天	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> h <input type="checkbox"/> h	<input type="checkbox"/> m <input type="checkbox"/> m	<input type="checkbox"/> s <input type="checkbox"/> s
0111——每星期	<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，除非将闹钟事件配置为在 2 月 29 日发生。



## 25.0 可编程循环冗余校验 (CRC) 发生器

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 36. Programmable Cyclic Redundancy Check (CRC)”，该文档可从 Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 下载。

可编程 CRC 发生器具有以下特性：

- 用户可编程多项式 CRC 公式
- 中断输出
- 数据 FIFO

## 25.1 概述

该模块实现了可用软件配置的 CRC 发生器。多项式的项及其长度可分别用 CRCXOR (X<15:1>) 位及 CRCCON (PLEN<3:0>) 位设定。

### 公式 25-1: CRC 公式

$$x^{16} + x^{12} + x^5 + 1$$

要将该多项式编程到 CRC 发生器中，CRC 寄存器位应按表 25-1 所示进行设置。

表 25-1: CRC 设置示例

位名称	位值
PLEN<3:0>	1111
X<15:1>	000100000010000

对于 X<15:1> 的值，第 12 位和第 5 位按照 CRC 公式要求设置为 1。CRC 公式要求的第 0 位总是经过逻辑异或运算的。对于 16 位多项式，第 16 位也总是被认为经过逻辑异或运算的；因此 X<15:1> 位没有第 0 位，也没有第 16 位。

图 25-2 给出了标准 CRC 发生器的拓扑结构。

图 25-1: CRC 移位寄存器详细信息

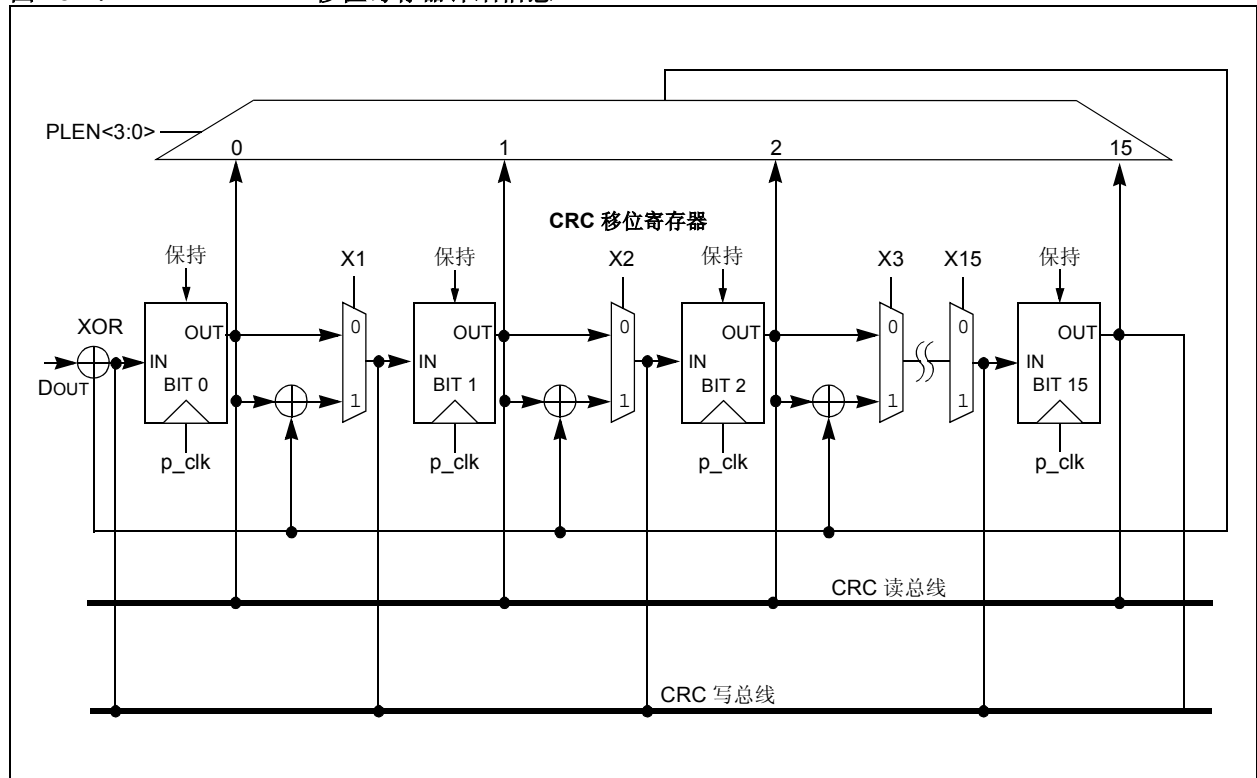
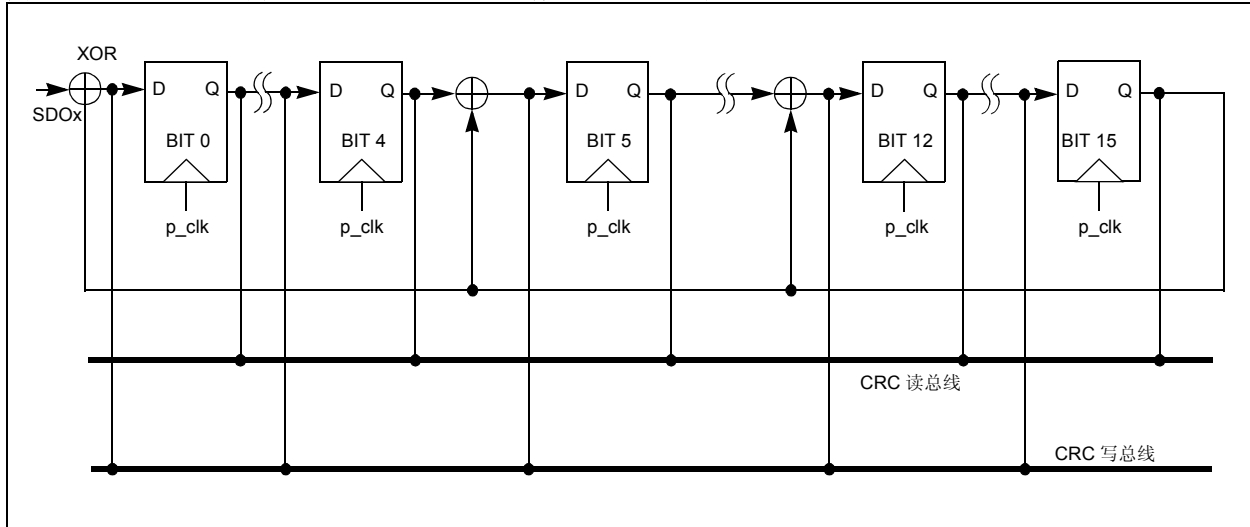


图 25-2: 为  $x^{16} + x^{12} + x^5 + 1$  重新配置的 CRC 发生器

## 25.2 用户接口

### 25.2.1 数据接口

要开始串行移位，必须向 CRCGO 位写入 1。

该模块包含的 FIFO 在 PLEN (PLEN<3:0>) > 7 时为 8 级深，其他情况下为 16 级深。必须先将需要进行 CRC 计算的数据写入 FIFO。可写入 FIFO 的最小数据元素是 1 个字节。例如，如果 PLEN = 5，则数据大小是 PLEN + 1 = 6。必须按如下方式写入数据：

```
data[5:0] = crc_input[5:0]
data[7:6] = `bxxx
```

数据一旦写入 CRCWDAT 的 MSb (由 PLEN 定义)，VWORD (VWORD<4:0>) 的值就递增 1。当 CRCGO = 1 且 VWORD > 0 时，串行移位寄存器开始将数据移入 CRC 引擎。当 MSb 被移出时，VWORD 递减 1。串行移位寄存器继续移位，直到 VWORD 达到 0。因此，对给定的 PLEN 值，需要 (PLEN + 1) \* VWORD 个时钟周期来完成 CRC 计算。

当 VWORD 达到 8 (或 16) 时，CRCFUL 位将置 1。当 VWORD 达到 0 时，CRCMPT 位将置 1。

要将数据连续送到 CRC 引擎中，建议的工作模式是预先向 FIFO 装入足够的字数，这样下个字节可写入之前不会产生中断。完成后，通过将 CRCGO 位置为 1 来启动 CRC。从此时开始应查询 VWORD 位。如果读到的值小于 8 或 16，将把另一个字写入 FIFO。

要清空已写入 FIFO 的字，CRCGO 位必须置为 1 并允许 CRC 移位寄存器运行，直到 CRCMPT 位被置 1。

同时，为了正确地读 CRC，必须等待 CRCMPT 位置 1 才能读 CRCWDAT 寄存器。

如果 CRCFUL 位置 1 时写入一个字，VWORD 指针将计满返回到 0。硬件的操作就像 FIFO 为空时一样。但是，产生中断的条件并不满足；因此，不会产生中断 (见第 25.2.2 节“中断操作”)。

写入 CRCWDAT 后必须经过至少一个指令周期才能读 VWORD 位。

### 25.2.2 中断操作

当 VWORD4:VWORD0 位的值从 1 变为 0 时，将产生中断。

## 25.3 节能模式下的操作

### 25.3.1 休眠模式

如果模块工作时进入休眠模式，模块将在当前的状态暂停，直到时钟恢复执行。

### 25.3.2 空闲模式

要在空闲模式下使模块继续全功能工作，必须在进入该模式前将 CSIDL 位清零。

如果 CSIDL = 1，模块的操作与在休眠模式下相同；等待处理的中断事件将继续进行，即使模块时钟不可用。

## 25.4 寄存器

CRC 模块提供以下寄存器：

- CRC 控制寄存器
- CRC 异或多项式寄存器

**寄存器 25-1: CRCCON: CRC 控制寄存器**

U-0	U-0	R/W-0	R-0	R-0	R-0	R-0	R-0	
—	—	CSIDL	VWORD<4:0>					
bit 15								bit 8

R-0	R-1	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
CRCFUL	CRCMPT	—	CRCGO	PLEN<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 15-14 **未实现：** 读为 0
- bit 13 **CSIDL:** CRC 在空闲模式停止位  
1 = 当器件进入空闲模式时，模块停止工作  
0 = 在空闲模式下模块继续工作
- bit 12-8 **VWORD<4:0>:** 指针值位  
表示 FIFO 中的有效字数。当 PLEN<3:0> > 7 时最大值是 8；PLEN<3:0> ≤ 7 时最大值是 16。
- bit 7 **CRCFUL:** FIFO 满位  
1 = FIFO 已满  
0 = FIFO 未滿
- bit 6 **CRCMPT:** FIFO 空位  
1 = FIFO 为空  
0 = FIFO 非空
- bit 5 **未实现：** 读为 0
- bit 4 **CRCGO:** 启动 CRC 位  
1 = 启动 CRC 串行移位寄存器  
0 = CRC 串行移位寄存器关闭
- bit 3-0 **PLEN<3:0>:** 多项式长度位  
表示要生成的多项式的长度减去 1。

**寄存器 25-2: CRCXOR: CRC 异或多项式寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
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  $X^n$  的异或使能位

bit 0        **未实现**: 读为 0

## 26.0 并行主端口 (PMP)

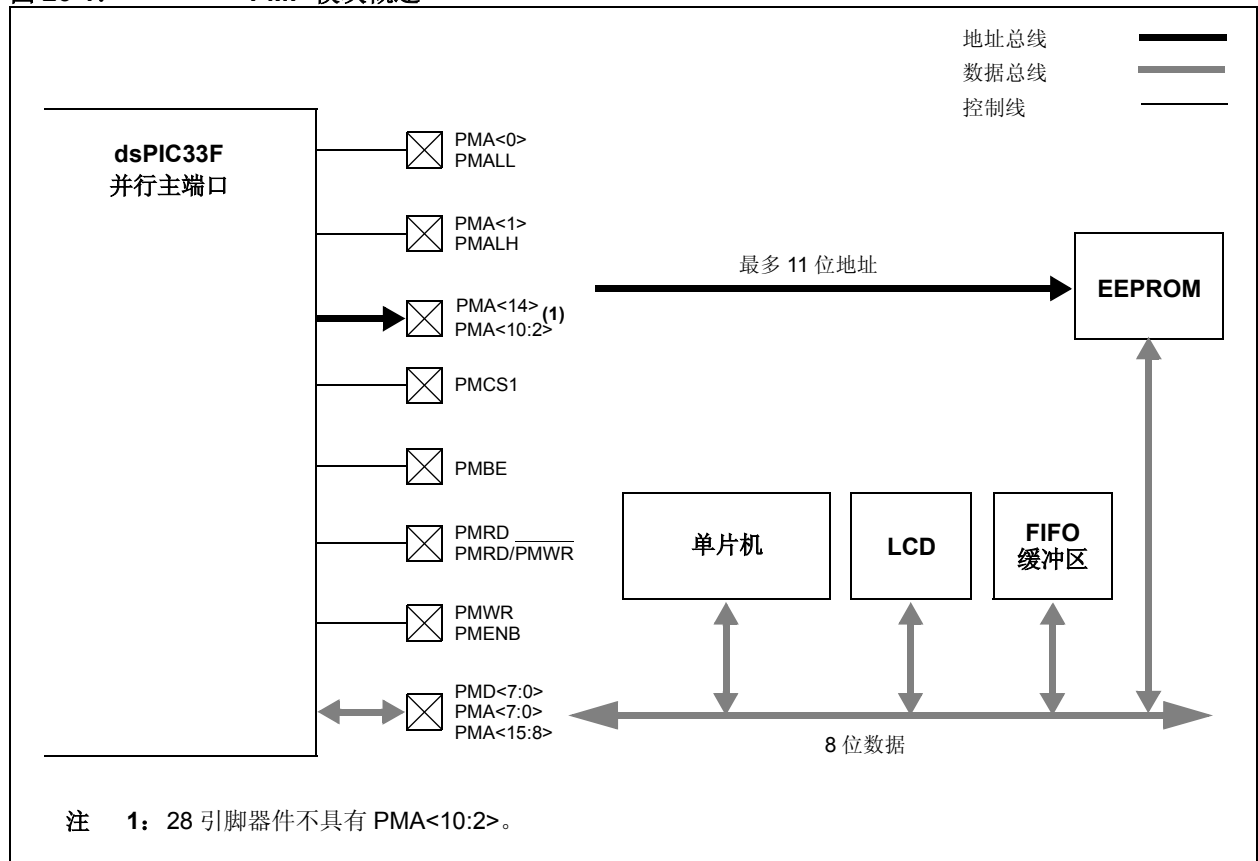
**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 35. Parallel Master Port (PMP)”，该文档可从 Microchip 网站 (www.microchip.com) 下载。

并行主端口 (Parallel Master Port, PMP) 模块是一个并行 8 位 I/O 模块，专用于与多种并口器件进行通信，例如通信外设、LCD、外部存储器及单片机。由于并行外设的接口的多样化，PMP 具有高度可配置性。

PMP 模块的关键特性包括：

- 完全复用的地址 / 数据模式
  - 16 位地址
- 不复用或部分复用的地址 / 数据模式
  - 带单根片选线时最多 11 根地址线
  - 无片选线时最多 12 根地址线
- 单根片选线
- 可编程选通选项
  - 单独读和写选通
  - 带使能选通的读 / 写选通
- 地址自动递增 / 自动递减
- 可编程地址 / 数据复用
- 控制信号的可编程极性
- 支持传统并行从端口
- 支持增强型并行从端口
  - 地址支持
  - 4 字节深的自动递增缓冲区
- 可编程等待状态
- 可选择的输入电平

图 26-1: PMP 模块概述



**寄存器 26-1: PMCON: 并行端口控制寄存器**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	PTBEEN	PTWREN	PTRDEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0 <sup>(1)</sup>	U-0	R/W-0 <sup>(1)</sup>	R/W-0	R/W-0	R/W-0
CSF1	CSF0	ALP	—	CS1P	BEP	WRSP	RDSP
bit 7							bit 0

**图注:**

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

- bit 15        **PMPEN:** 并行主端口使能位  
 1 = 使能 PMP  
 0 = 禁止 PMP, 不执行片外访问
- bit 14        **未实现:** 读为 0
- bit 13        **PSIDL:** 空闲模式停止位  
 1 = 当器件进入空闲模式时, 模块停止工作  
 0 = 在空闲模式下模块继续工作
- bit 12-11    **ADRMUX1:ADRMUX0:** 地址 / 数据复用选择位 <sup>(1)</sup>  
 11 = 保留  
 10 = 地址的所有 16 位与 PMD<7:0> 引脚复用  
 01 = 地址的低 8 位与 PMD<7:0> 引脚复用, 高 3 位与 PMA<10:8> 引脚复用  
 00 = 地址和数据出现在独立的引脚上
- bit 10        **PTBEEN:** 字节使能端口使能位 (16 位主模式)  
 1 = 使能 PMBE 端口  
 0 = 禁止 PMBE 端口
- bit 9         **PTWREN:** 写使能选通端口使能位  
 1 = 使能 PMWR/PMENB 端口  
 0 = 禁止 PMWR/PMENB 端口
- bit 8         **PTRDEN:** 读 / 写选通端口使能位  
 1 = 使能 PMRD/PMWR 端口  
 0 = 禁止 PMRD/PMWR 端口
- bit 7-6       **CSF1:CSF0:** 片选功能位  
 11 = 保留  
 10 = PMCS1 用作片选  
 0x = PMCS1 用作地址的 bit 14
- bit 5         **ALP:** 地址锁存器极性位 <sup>(2)</sup>  
 1 = 高电平有效 (PMALL 和 PMALH)  
 0 = 低电平有效 (PMALL 和 PMALH)
- bit 4         **未实现:** 读为 0
- bit 3         **CS1P:** 片选 1 极性位 <sup>(2)</sup>  
 1 = 高电平有效 (PMCS1/PMCS1)  
 0 = 低电平有效 (PMCS1/ PMCS1)

注 1: 28 引脚器件不具有 PMA<10:2>。  
 注 2: 这些位在相应引脚用作地址线时无效。

寄存器 26-1: PMCON: 并行端口控制寄存器 (续)

bit 2	<b>BEP:</b> 字节使能极性位 1 = 字节使能高电平有效 (PMBE) 0 = 字节使能低电平有效 (PMBE)
bit 1	<b>WRSP:</b> 写选通极性位 <u>对于从模式和主模式 2 (PMMODE&lt;9:8&gt; = 00、01 和 10):</u> 1 = 写选通高电平有效 (PMWR) 0 = 写选通低电平有效 (PMWR) <u>对于主模式 1 (PMMODE&lt;9:8&gt; = 11):</u> 1 = 使能选通高电平有效 (PMENB) 0 = 使能选通低电平有效 (PMENB)
bit 0	<b>RDSP:</b> 读选通极性位 <u>对于从模式和主模式 2 (PMMODE&lt;9:8&gt; = 00、01 和 10):</u> 1 = 读选通高电平有效 (PMRD) 0 = 读选通低电平有效 (PMRD) <u>对于主模式 1 (PMMODE&lt;9:8&gt; = 11):</u> 1 = 读 / 写选通高电平有效 (PMRD/PMWR) 0 = 读 / 写选通低电平有效 (PMRD/PMWR)

- 注 1: 28 引脚器件不具有 PMA<10:2>。  
2: 这些位在相应引脚用作地址线时无效。

**寄存器 26-2: PMMODE: 并行端口模式寄存器**

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUSY	IRQM<1:0>		INCM<1:0>		MODE16	MODE<1:0>	
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
WAITB<1:0> <sup>(1)</sup>		WAITM<3:0>				WAITE<1:0> <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        **BUSY:** 忙状态位 (仅限主模式)  
 1 = 端口处于忙状态 (当处理器暂停状态激活时无效)  
 0 = 端口未处于忙状态
- bit 14-13    **IRQM<1:0>:** 中断请求模式位  
 11 = 当对读缓冲区 3 执行读操作或对写缓冲区 3 执行写操作时产生中断 (缓冲 PSP 模式), 或者当 PMA<1:0> = 11 (仅限可寻址 PSP 模式) 时执行读 / 写操作产生中断  
 10 = 不产生中断, 处理器暂停状态激活  
 01 = 读 / 写周期末尾产生中断  
 00 = 不产生中断
- bit 12-11    **INCM<1:0>:** 递增模式位  
 11 = PSP 读和写缓冲区自动递增 (仅限传统 PSP 模式)  
 10 = 每个读 / 写周期 ADDR<10:0> 递减 1  
 01 = 每个读 / 写周期 ADDR<10:0> 递增 1  
 00 = 无地址递增或递减
- bit 10        **MODE16:** 8/16 位模式位  
 1 = 16 位模式: 数据寄存器为 16 位, 读或写数据寄存器调用两次 8 位传输  
 0 = 8 位模式: 数据寄存器为 8 位, 读或写数据寄存器调用一次 8 位传输
- bit 9-8       **MODE<1:0>:** 并行端口模式选择位  
 11 = 主模式 1 (PMCS1、PMRD/PMWR、PMENB、PMBE、PMA<x:0> 和 PMD<7:0>)  
 10 = 主模式 2 (PMCS1、PMRD、PMWR、PMBE、PMA<x:0> 和 PMD<7:0>)  
 01 = 增强型 PSP, 控制信号 (PMRD、PMWR、PMCS1、PMD<7:0> 和 PMA<1:0>)  
 00 = 传统并行从端口, 控制信号 (PMRD、PMWR、PMCS1 和 PMD<7:0>)
- bit 7-6       **WAITB<1:0>:** 从数据建立到执行读 / 写操作的等待状态配置位 <sup>(1)</sup>  
 11 = 数据等待 4 个 Tcy; 复用地址阶段等待 4 个 Tcy  
 10 = 数据等待 3 个 Tcy; 复用地址阶段等待 3 个 Tcy  
 01 = 数据等待 2 个 Tcy; 复用地址阶段等待 2 个 Tcy  
 00 = 数据等待 1 个 Tcy; 复用地址阶段等待 1 个 Tcy
- bit 5-2       **WAITM<3:0>:** 从读操作到字节使能选通的等待状态配置位  
 1111 = 额外等待 15 个 Tcy  
 .  
 .  
 .  
 0001 = 额外等待 1 个 Tcy  
 0000 = 无额外等待周期 (强制操作在 1 个 Tcy 内执行)
- bit 1-0       **WAITE<1:0>:** 选通后数据保持的等待状态配置位 <sup>(1)</sup>  
 11 = 等待 4 个 Tcy  
 10 = 等待 3 个 Tcy  
 01 = 等待 2 个 Tcy  
 00 = 等待 1 个 Tcy

**注 1:** 每当 WAITM3:WAITM0 = 0000 时, WAITB 和 WAITE 位就被忽略。



**寄存器 26-3: PMADDR: 并行端口地址寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADDR15	CS1	ADDR<13:8>					
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADDR<7:0>							
bit 7							bit 0

**图注:**

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

bit 15            **ADDR15:** 并行端口目标地址位  
 bit 14            **CS1:** 片选 1 位  
                   1 = 片选 1 有效  
                   0 = 片选 1 无效  
 bit 13-0        **ADDR13:ADDR0:** 并行端口目标地址位

**寄存器 26-4: PMAEN: 并行端口使能寄存器**

U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	PTEN14	—	—	—	PTEN<10:8> <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
PTEN<7:2> <sup>(1)</sup>						PTEN<1:0>	
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:** PMCS1 选通使能位  
                   1 = PMA14 作为 PMA<14> 位或者 PMCS1  
                   0 = PMA14 引脚作为端口 I/O  
 bit 13-11       **未实现:** 读为 0  
 bit 10-2        **PTEN<10:2>:** PMP 地址端口使能位 <sup>(1)</sup>  
                   1 = PMA<10:2> 作为 PMP 地址线  
                   0 = PMA<10:2> 作为端口 I/O  
 bit 1-0         **PTEN<1:0>:** PMALH/PMALL 选通使能位  
                   1 = PMA1 和 PMA0 作为 PMA<1:0> 或者 PMALH 和 PMALL  
                   0 = PMA1 和 PMA0 引脚作为端口 I/O

**注 1:** 28 引脚器件不具有 PMA<10:2>。

**寄存器 26-5: PMSTAT: 并行端口状态寄存器**

R-0	R/W-0, HS	U-0	U-0	R-0	R-0	R-0	R-0
IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F
bit 15						bit 8	

R-1	R/W-0, HS	U-0	U-0	R-1	R-1	R-1	R-1
OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E
bit 7						bit 0	

<b>图注:</b>	HS = 由硬件置 1		
R = 可读位	W = 可写位	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 = 所有可写输入缓冲寄存器均已满  
0 = 部分或所有可写输入缓冲寄存器为空
- bit 14      **IBOV:** 输入缓冲区溢出状态位  
1 = 尝试对已满的输入字节寄存器进行写操作 (必须用软件清零)  
0 = 未发生溢出
- bit 13-12    **未实现:** 读为 0
- bit 11-8     **IB3F:IB0F:** 输入缓冲区 x 状态满位  
1 = 输入缓冲区中有尚未读取的数据 (读缓冲区将清零该位)  
0 = 输入缓冲区不包含任何未读数据
- bit 7        **OBE:** 输出缓冲区空状态位  
1 = 所有可读输出缓冲寄存器为空  
0 = 部分或所有可读输出缓冲寄存器为满
- bit 6        **OBUF:** 输出缓冲区下溢状态位  
1 = 对空的输出字节寄存器执行了读操作 (必须用软件清零)  
0 = 未发生下溢
- bit 5-4      **未实现:** 读为 0
- bit 3-0      **OB3E:OB0E:** 输出缓冲区 x 状态空位  
1 = 输出缓冲区为空 (向缓冲区写数据将清零该位)  
0 = 输出缓冲区包含尚未发送的数据

寄存器 26-6: 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	R/W-0	R/W-0
—	—	—	—	—	—	RTSECSEL <sup>(1)</sup>	PMP TTL
bit 7						bit 0	

图注:

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

- bit 15-2      **未实现:** 读为 0
- bit 1        **RTSECSEL:** RTCC 秒时钟输出选择位 <sup>(1)</sup>
  - 1 = 选择 RTCC 引脚输出 RTCC 秒时钟
  - 0 = 选择 RTCC 引脚输出 RTCC 闹钟脉冲
- bit 0        **PMP TTL:** PMP 模块 TTL 输入缓冲器选择位
  - 1 = PMP 模块使用 TTL 输入缓冲器
  - 0 = PMP 模块使用施密特触发器输入缓冲器

注 1: 要启用实际 RTCC 输出, 需要将 RTCOE (RCFGCAL) 位置 1。

图 26-2: 传统并行从端口示例

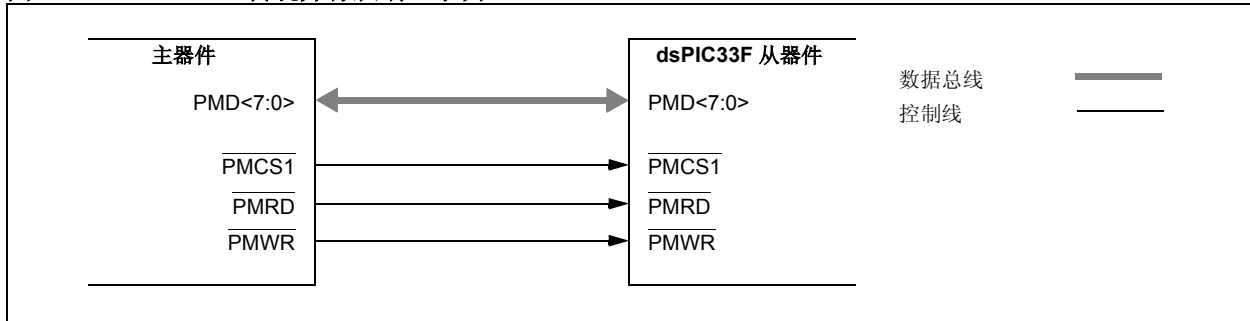


图 26-3: 可寻址并行从端口示例

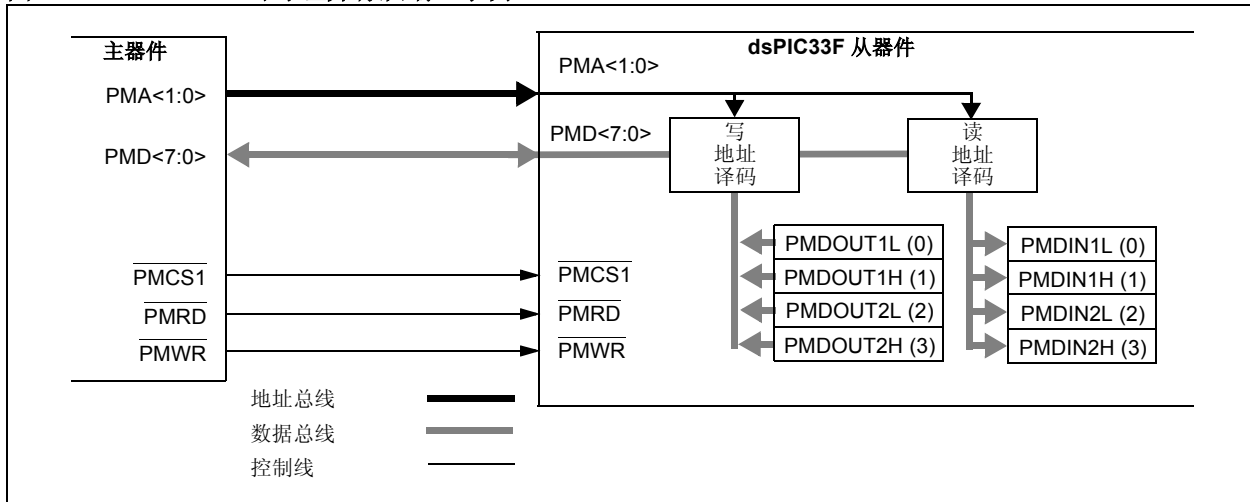


表 26-1: 从模式地址解析

PMA<1:0>	输出寄存器 (缓冲区)	输入寄存器 (缓冲区)
00	PMDOUT1<7:0> (0)	PMDIN1<7:0> (0)
01	PMDOUT1<15:8> (1)	PMDIN1<15:8> (1)
10	PMDOUT2<7:0> (2)	PMDIN2<7:0> (2)
11	PMDOUT2<15:8> (3)	PMDIN2<15:8> (3)

图 26-4: 主模式，非复用寻址 (独立的读和写选通，单片选)

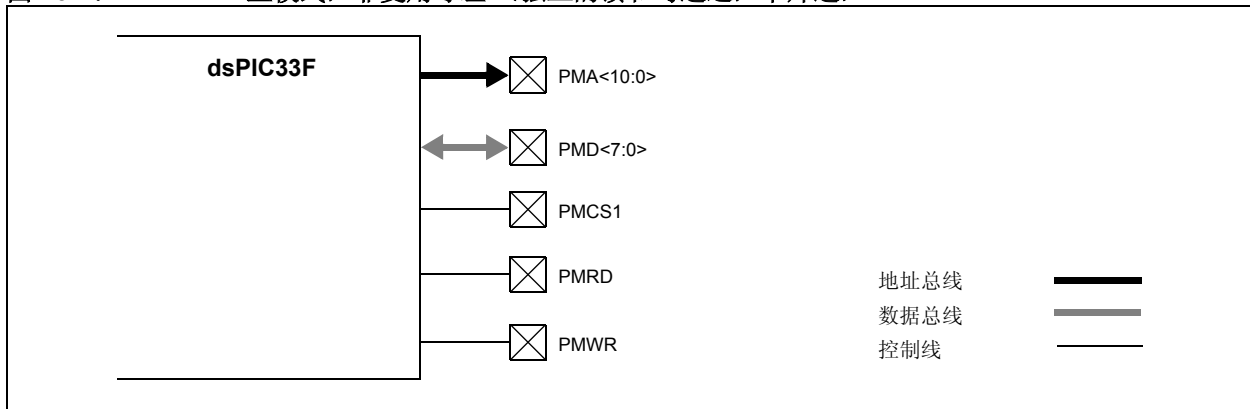


图 26-5: 主模式，部分复用的寻址（独立的读和写选通，单片选）

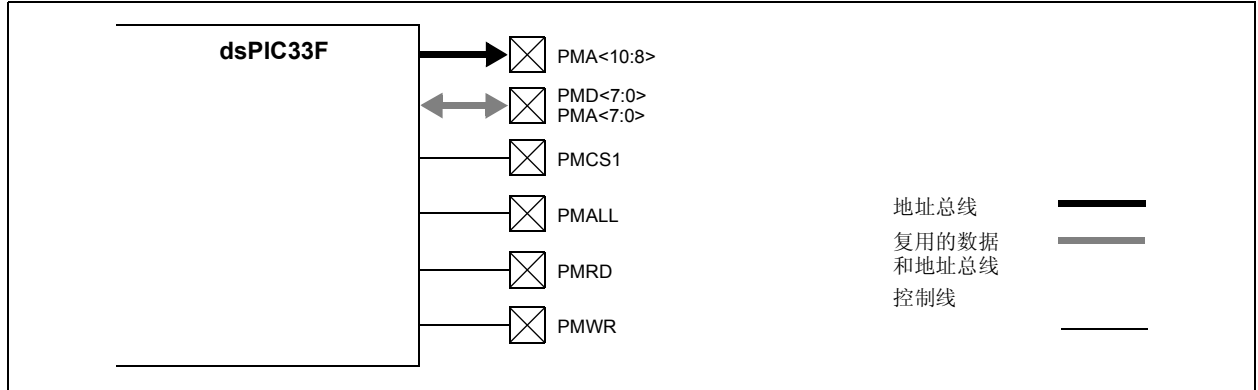


图 26-6: 主模式，完全复用的寻址（独立的读和写选通，单片选）

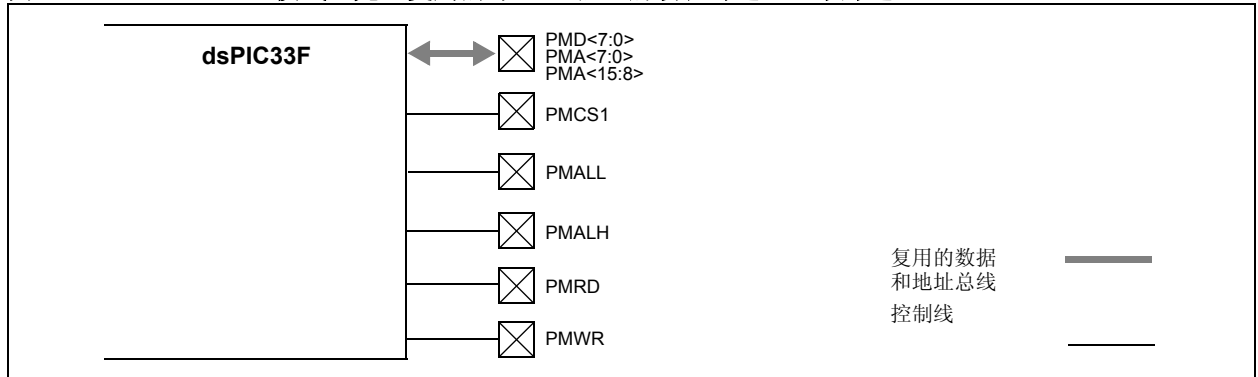


图 26-7: 复用寻址应用示例

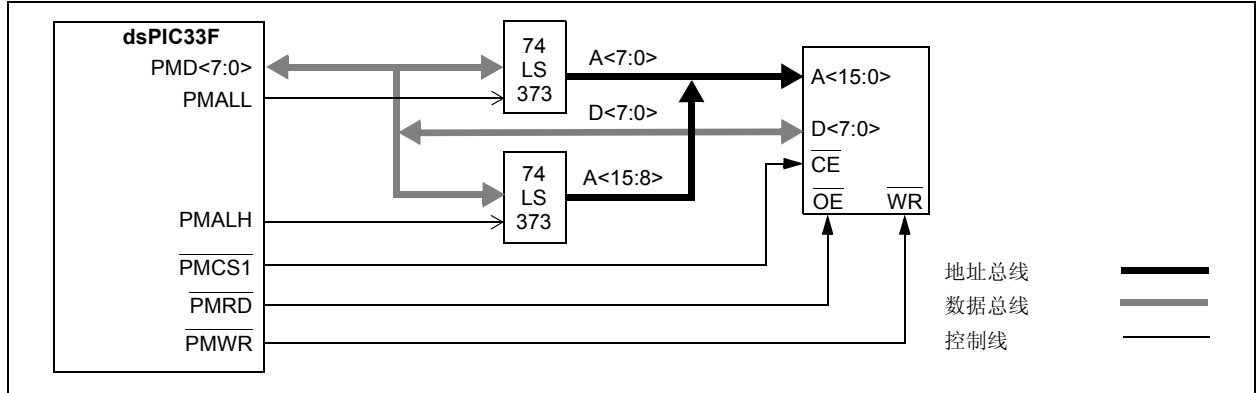


图 26-8: 部分复用寻址应用示例

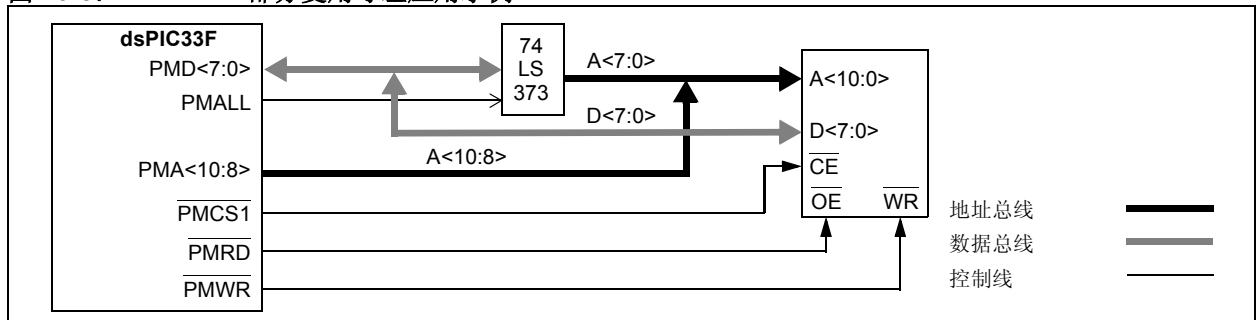


图 26-9: 8 位地址和数据复用应用示例

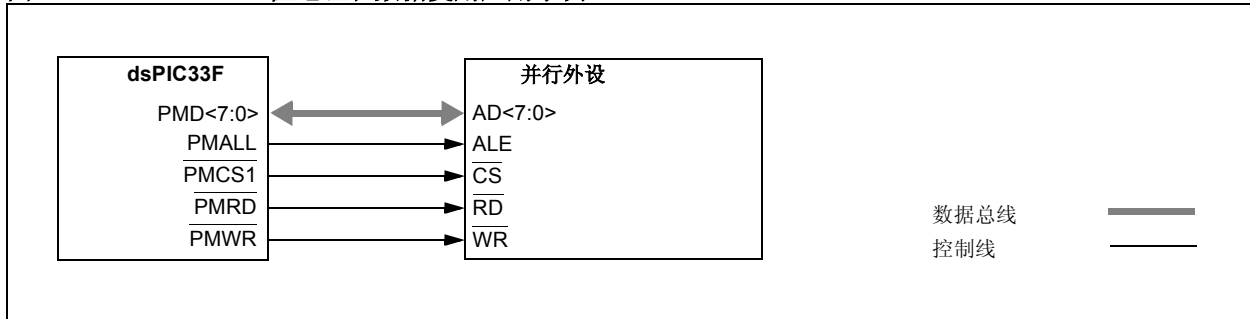


图 26-10: 并行 EEPROM 示例（最多 11 位地址和 8 位数据）

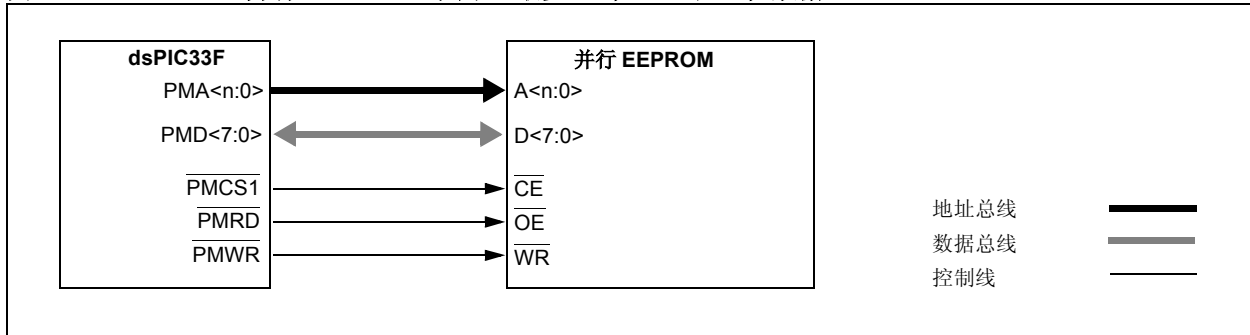


图 26-11: 并行 EEPROM 示例（最多 11 位地址和 16 位数据）

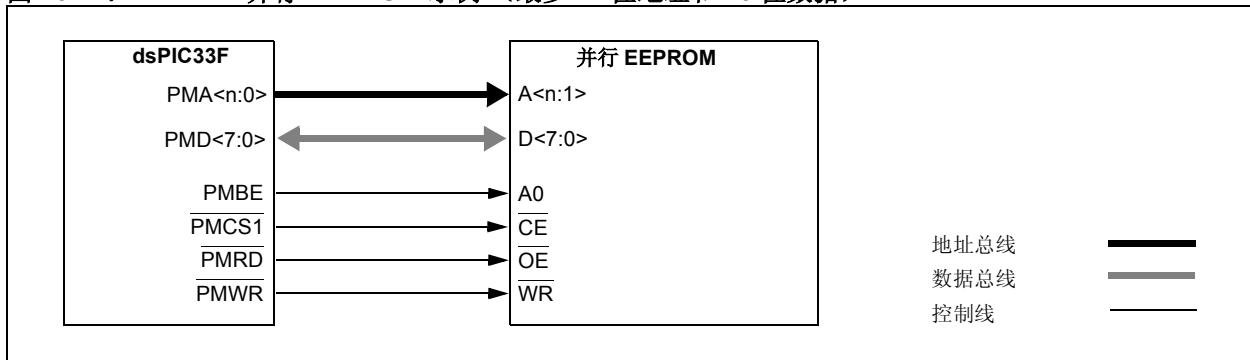
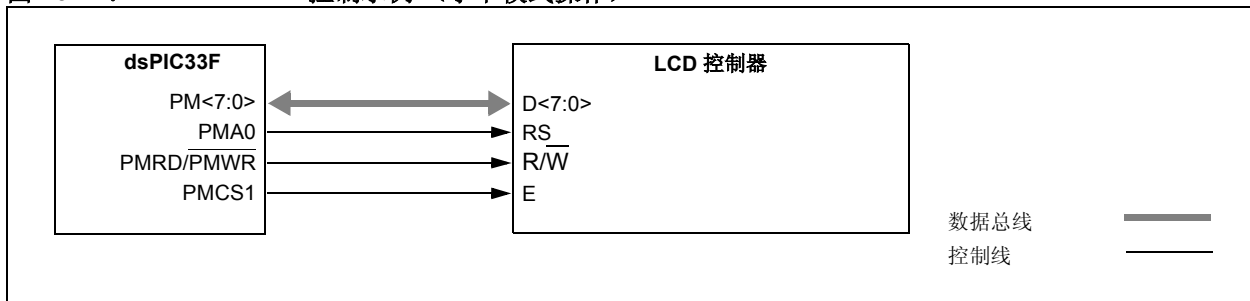


图 26-12: LCD 控制示例（字节模式操作）



## 27.0 特殊功能

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“*dsPIC33F Family Reference Manual*”的相关章节，该文档可从 [Microchip 网站 \(www.microchip.com\)](http://www.microchip.com) 下载。

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件具有一些特殊的功能，旨在最大限度地提高应用的灵活性和可靠性，并通过减少外部元件的使用将成本降至最低。这些特殊功能包括：

- 灵活的配置
- 看门狗定时器 (WDT)
- 代码保护和 CodeGuard™ 安全性
- JTAG 边界扫描接口
- 在线串行编程 (In-Circuit Serial Programming™, ICSP™)
- 在线仿真

**表 27-1: 器件配置寄存器映射**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xF80000	FBS	RBS<1:0>		—	—	BSS<2:0>		BWRP	
0xF80002	FSS	RSS<1:0>		—	—	SSS<2:0>		SWRP	
0xF80004	FGS	—	—	—	—	—	GSS<1:0>		GWRP
0xF80006	FOSCSEL	IESO	—	—	—		FNOSC<2:0>		
0xF80008	FOSC	FCKSM<1:0>		IOL1WAY	—	—	OSCIOFNC	POSCMD<1:0>	
0xF8000A	FWDT	FWDTEN	WINDIS	—	WDTPRE	WDTPOST<3:0>			
0xF8000C	FPOR	PWMPIN	HPOL	LPOL	ALTI2C	—	FPWRT<2:0>		
0xF8000E	保留	保留 (1)							
0xF80010	FUID0	用户部件 ID 字节 0							
0xF80012	FUID1	用户部件 ID 字节 1							
0xF80014	FUID2	用户部件 ID 字节 2							
0xF80016	FUID3	用户部件 ID 字节 3							

**注 1：** 这些保留的位读为 1 并且必须被编程为 1。

## 27.1 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以 0xF80000 开始的存储单元中。

表 27-2 给出了 FBS、FGS、FOSCSEL、FOSC、FWDT 和 FPOR 配置寄存器中各配置位的说明。

注意，地址 0xF80000 超出了用户程序存储空间。它属于只能使用表读和表写访问的配置存储空间（0x800000-0xFFFFF）。

所有器件配置寄存器的高字节应该总为 1111 1111。这使得在极少情况下意外执行这些存储单元时将其作为 NOP 指令来执行。由于没有在相应的存储单元中实现这些配置位，因此向这些存储单元写入 1 不会影响器件工作。

为了避免在代码执行期间配置被无意中改变，所有的可编程配置位只可被写入一次。在上电周期内对位进行初始化编程之后就不能再次写该位了。改变器件的配置需要对器件重复上电。

器件配置寄存器的映射如表 27-1 所示。

表 27-2: dsPIC33F 配置位的说明

位域	寄存器	说明
BWRP	FBS	引导段程序闪存写保护 1 = 引导段可写 0 = 引导段被写保护
BSS<2:0>	FBS	引导段程序闪存代码保护大小 X11 = 无引导程序闪存段  引导空间为 1K 指令字（中断向量除外） 110 = 标准安全性；引导程序闪存段结束于 0x0007FE 010 = 高安全性；引导程序闪存段结束于 0x0007FE  引导空间为 4K 指令字（中断向量除外） 101 = 标准安全性；引导程序闪存段结束于 0x001FFE 001 = 高安全性；引导程序闪存段结束于 0x001FFE  引导空间为 8K 指令字（中断向量除外） 100 = 标准安全性；引导程序闪存段结束于 0x003FFE 000 = 高安全性；引导程序闪存段结束于 0x003FFE
RBS<1:0> <sup>(1)</sup>	FBS	引导段 RAM 代码保护大小 11 = 未定义引导 RAM 10 = 引导 RAM 为 128 字节 01 = 引导 RAM 为 256 字节 00 = 引导 RAM 为 1024 字节
SWRP	FSS	安全段程序闪存写保护位 1 = 安全段可写 0 = 安全段被写保护
SSS<2:0>	FSS	安全段程序闪存代码保护大小 (安全段在 32K 器件上未实现) X11 = 无安全程序闪存段  安全空间为 4K 指令字减去 BS 大小 110 = 标准安全性；安全程序闪存段开始于 BS 末端，结束于 0x001FFE 010 = 高安全性；安全程序闪存段开始于 BS 末端，结束于 0x001FFE  安全空间为 8K 指令字减去 BS 大小 101 = 标准安全性；安全程序闪存段开始于 BS 末端，结束于 0x003FFE 001 = 高安全性；安全程序闪存段开始于 BS 末端，结束于 0x003FFE  安全空间为 16K 指令字减去 BS 大小 100 = 标准安全性；安全程序闪存段开始于 BS 末端，结束于 007FFFEh 000 = 高安全性；安全程序闪存段开始于 BS 末端，结束于 0x007FFE
RSS<1:0> <sup>(1)</sup>	FSS	安全段 RAM 代码保护 10 = 未定义安全 RAM 10 = 安全 RAM 为 256 字节减去 BS RAM 大小 01 = 安全 RAM 为 2048 字节减去 BS RAM 大小 00 = 安全 RAM 为 4096 字节减去 BS RAM 大小

注 1: RAM 代码保护仅在 64K 和 128K 器件上可用，在 32K 器件上未实现。



表 27-2: dsPIC33F 配置位的说明 (续)

位域	寄存器	说明
GSS<1:0>	FGS	通用段代码保护位 11 = 用户程序存储区未被代码保护 10 = 标准安全性 0x = 高安全性
GWRP	FGS	通用段写保护位 1 = 用户程序存储区未被写保护 0 = 用户程序存储区被写保护
IESO	FOSCSEL	双速振荡器启动使能位 1 = 使用 FRC 启动器件, 然后自动切换到就绪的用户选择的振荡器源 0 = 使用用户选择的振荡器源启动器件
FNOSC<2:0>	FOSCSEL	初始振荡器源选择位 111 = 带后分频器的内部快速 RC (FRC) 振荡器 110 = 带 16 分频的内部快速 RC (FRC) 振荡器 101 = LPRC 振荡器 100 = 辅助 (LP) 振荡器 011 = 带 PLL 的主 (XT、HS 或 EC) 振荡器 010 = 主 (XT、HS 或 EC) 振荡器 001 = 带 PLL 的内部快速 RC (FRC) 振荡器 000 = FRC 振荡器
FCKSM<1:0>	FOSC	时钟切换模式位 1x = 禁止时钟切换, 禁止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, 使能故障保护时钟监视器
IOL1WAY	FOSC	外设引脚选择配置 1 = 只允许一次重新配置 0 = 允许多次重新配置
OSCI0FNC	FOSC	OSC2 引脚功能位 (XT 和 HS 模式除外) 1 = OSC2 为时钟输出 0 = OSC2 为通用数字 I/O 引脚
POSCMD<1:0>	FOSC	主振荡器模式选择位 11 = 禁止主振荡器 10 = HS 晶振模式 01 = XT 晶振模式 00 = EC (外部时钟) 模式
FWDTEN	FWDT	看门狗定时器使能位 1 = 始终使能看门狗定时器 (不能禁止 LPRC 振荡器。将 RCON 寄存器中的 SWDTEN 位清零不会产生任何影响。) 0 = 通过用户软件使能 / 禁止看门狗定时器 (可通过清零 RCON 寄存器中的 SWDTEN 位来禁止 LPRC)
WINDIS	FWDT	看门狗定时器窗口使能位 1 = 非窗口模式下的看门狗定时器 0 = 窗口模式下的看门狗定时器
WDTPRE	FWDT	看门狗定时器预分频比位 1 = 1:128 0 = 1:32

注 1: RAM 代码保护仅在 64K 和 128K 器件上可用, 在 32K 器件上未实现。

表 27-2: dsPIC33F 配置位的说明 (续)

位域	寄存器	说明
WDTPOST<3:0>	FWDT	看门狗定时器后分频比位 1111 = 1:32,768 1110 = 1:16,384 . . . 0001 = 1:2 0000 = 1:1
PWMPIN	FPOR	电机控制 PWM 模块引脚模式位 1 = PWM 模块引脚在器件复位时由端口寄存器控制 (三态) 0 = PWM 模块引脚在器件复位时由 PWM 模块控制 (配置为输出引脚)
HPOL	FPOR	电机控制 PWM 高端极性位 1 = PWM 模块高端输出引脚具有高电平有效的输出极性 0 = PWM 模块高端输出引脚具有低电平有效的输出极性
LPOL	FPOR	电机控制 PWM 低端极性位 1 = PWM 模块低端输出引脚具有高电平有效的输出极性 0 = PWM 模块低端输出引脚具有低电平有效的输出极性
FPWRT<2:0>	FPOR	上电复位定时器值选择位 111 = PWRT = 128 ms 110 = PWRT = 64 ms 101 = PWRT = 32 ms 100 = PWRT = 16 ms 011 = PWRT = 8 ms 010 = PWRT = 4 ms 001 = PWRT = 2 ms 000 = PWRT = 禁止
ALTI2C	FPOR	备用 I <sup>2</sup> C™ 引脚 1 = I <sup>2</sup> C 被映射到 SDA1/SCL1 引脚 0 = I <sup>2</sup> C 被映射到 ASDA1/ASCL1 引脚

注 1: RAM 代码保护仅在 64K 和 128K 器件上可用, 在 32K 器件上未实现。

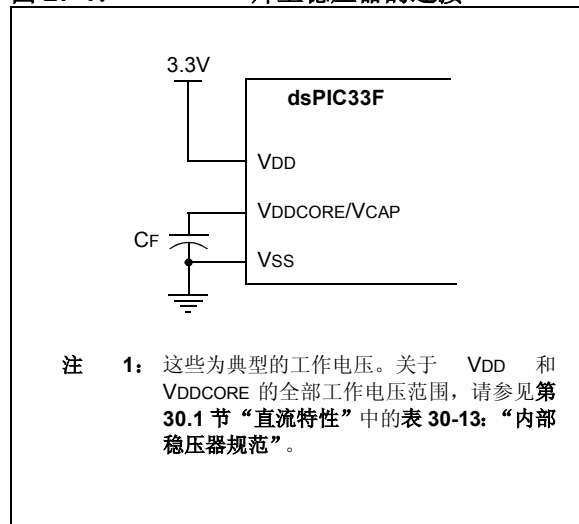
## 27.2 片上稳压器

所有 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件使用标称值 2.5V 的电压为其内核数字逻辑供电。对于需要工作在一个更高的典型电压值（如 3.3V）的设计中，这可能会产生冲突。为简化系统设计，dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列中的所有器件均包含一个片上稳压器，可使器件内核逻辑在 VDD 下工作。

稳压器通过其他 VDD 引脚为内核供电。当使能了稳压器时，必须将一个低 ESR（小于 5 欧姆）电容（如钽电容或陶瓷电容）连接到 VDDCORE/VCAP 引脚（图 27-1）。这可帮助维持稳压器的稳定性。滤波电容的推荐值在第 30.0 节“电气特性”的表 30-13 中提供。

在 POR 时，片上稳压器需要约 20  $\mu$ s 的时间来产生输出电压。这段时间被称为 TSTARTUP，在此期间禁止代码执行。在每次掉电后器件恢复工作的过程中都将有一段 TSTARTUP 时间。

图 27-1: 片上稳压器的连接<sup>(1)</sup>



## 27.3 BOR: 欠压复位

欠压复位（BOR）模块基于内部参考电压电路，该电路监视经过稳压后的电压 VDDCORE。BOR 模块的主要用途是在发生欠压条件时产生器件复位。欠压条件通常由交流电源线上的干扰信号（例如，由于电源传输线路不良造成的交流周期波形部分丢失，或者由于接入大感性负载时电流消耗过大造成电压下降）产生的。

BOR 将产生复位器件的复位脉冲。BOR 会根据器件配置位（FNOSC<2:0> 和 POSCMD<1:0>）的值选择时钟源。

如果选择了振荡器模式，BOR 将激活振荡器起振定时器（OST）。系统时钟将保持到 OST 超时。如果使用了 PLL，则时钟将被保持到 LOCK 位（OSCCON<5>）置 1。

同时，将在内部复位释放前应用 PWRT 延时（TPWRT）。如果 TPWRT = 0 且使用了晶振，那么会应用 TFSCM = 100 的标称延时。这种情况下总的延时为 TFSCM。

BOR 状态位（RCON<1>）置 1，表明发生了 BOR。如果使能了 BOR 电路，该电路在休眠或空闲模式下继续工作，当 VDD 下降到 BOR 门限电压以下时将复位器件。

## 27.4 看门狗定时器 (WDT)

对于 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件，WDT 由 LPRC 振荡器驱动。当使能 WDT 时，时钟源也将使能。

### 27.4.1 预分频器 / 后分频器

由 LPRC 提供的 WDT 时钟源的频率标称值为 32 kHz。此信号输入给可配置为 5 位 (32 分频) 或 7 位 (128 分频) 操作的预分频器。预分频比由 WDTPRE 配置位设置。使用 32 kHz 输入，预分频器在 5 位模式下将产生一个 1 ms 的标称 WDT 超时周期 (TWDT)，在 7 位模式下产生的超时周期为 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，并扩展超时周期范围。后分频比由 WDTPOST<3:0> 配置位 (FWDT<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，可以使超时周期的范围扩展到 1 ms 至 131 秒。

WDT、预分频器和后分频器在以下条件下复位：

- 任何器件复位时
- 在完成时钟切换后，无论时钟切换是由软件（即在更改 NOSC 位后将 OSWEN 位置 1）引起还是由硬件（即，故障保护时钟监视器）引起
- 当执行 PWRSAV 指令时（即，进入休眠或空闲模式）
- 当器件退出休眠或空闲模式恢复正常工作时
- 当在正常执行过程中使用 CLRWDT 指令时

**注：** 执行 CLRWDT 和 PWRSAV 指令会将预分频器和后分频器的计数值清零。

### 27.4.2 休眠和空闲模式

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，将唤醒器件并且将继续从 PWRSAV 指令处执行代码。当器件唤醒后，需要用软件将相应的 SLEEP 或 IDLE 位 (RCON<3,2>) 清零。

### 27.4.3 使能 WDT

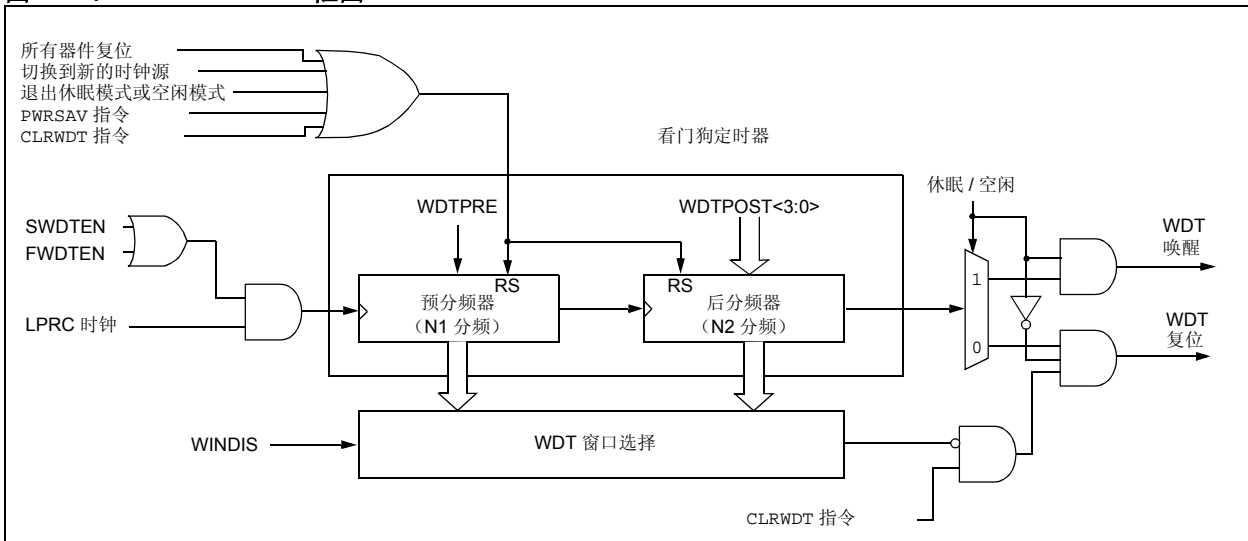
WDT 的使能或禁止由 FWDT 配置寄存器中的 FWDTEN 配置位控制。当 FWDTEN 配置位置 1 时，WDT 始终是使能的。

当 FWDTEN 配置位被编程为 0 时，可以选择用软件控制 WDT。用软件通过将 SWDTEN 控制位 (RCON<5>) 置 1 来使能 WDT。SWDTEN 控制位在任何器件复位时清零。软件 WDT 选项允许用户应用程序在关键代码段使能 WDT 并在非关键代码段禁止 WDT，从而最大限度地降低功耗。

**注：** 如果 WINDIS 位 (FWDT<6>) 清零，应用软件应仅在 WDT 周期的最后 1/4 周期中执行 CLRWDT 指令。该 CLRWDT 窗口可通过使用定时器确定。如果在该窗口之前执行 CLRWDT 指令，仍将发生 WDT 复位。

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位清零。

图 27-2: WDT 框图



## 27.5 JTAG 接口

dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件实现了一个 JTAG 接口，该接口支持边界扫描器件测试以及在线编程。有关该接口的详细信息将会在文档的将来版本中提供。

**注：** 请参考“*dsPIC33F Family Reference Manual*”的“**Section 24. Programming and Diagnostics**” (DS70207) 获得有关 JTAG 接口使用、配置和操作方面的更多信息。

## 27.6 在线串行编程

可在最终的应用电路中对 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件进行串行编程。只需要 5 根线即可实现这一操作，其中时钟线、数据线各一根，其余 3 根分别是电源线、接地线和编程电压线。串行编程允许用户在生产电路板时使用未编程器件，而在产品交付之前才对数字信号控制器进行编程，从而可以使用最新版本的固件或者定制固件进行编程。请参见《dsPIC33F/PIC24H 闪存编程规范》(DS70152C\_CN) 了解有关在线串行编程 (ICSP) 的详细信息。

可使用 3 对编程时钟 / 数据引脚中的任意一对：

- PGC1/EMUC1 和 PGD1/EMUD1
- PGC2/EMUC2 和 PGD2/EMUD2
- PGC3/EMUC3 和 PGD3/EMUD3

## 27.7 在线调试器

当选择 MPLAB® ICD 2 作为调试器时，使能在线调试功能。该功能允许与 MPLAB IDE 配合使用来进行简单的调试。通过 EMUCx（仿真 / 调试时钟）和 EMUDx（仿真 / 调试数据）引脚功能控制调试功能。

可使用 3 对调试时钟 / 数据引脚中的任意一对：

- PGC1/EMUC1 和 PGD1/EMUD1
- PGC2/EMUC2 和 PGD2/EMUD2
- PGC3/EMUC3 和 PGD3/EMUD3

要使用器件的在线调试功能，就必须在设计中正确对 MCLR、V<sub>DD</sub>、V<sub>SS</sub>、PGC、PGD 和 EMUDx/EMUCx 引脚对进行 ICSP 连接。此外，当使能该功能时，某些资源就不能用于一般用途了。这些资源包括数据 RAM 的前 80 字节和两个 I/O 引脚。

## 27.8 代码保护和 CodeGuard 安全性

dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件提供了支持 BS、SS 和 GS 的 CodeGuard 安全性的高级实现，而 dsPIC33FJ32MC302/304 器件则提供了仅支持 BS 和 GS 的 CodeGuard 安全性的中等实现。CodeGuard 安全性允许多方安全地共享同一个芯片上的资源（存储器、中断和外设）。这一功能有助于在协作系统设计中保护各方的知识产权（Intellectual Property, IP）。

CodeGuard 安全性与软件加密函数库配合使用时，即使在单个芯片上存在多个知识产权（IP），也可以使用 CodeGuard 安全性来安全地更新闪存。代码保护功能的实现随实际所使用的 dsPIC33F 器件而有所不同。后续章节将对这些功能进行概括介绍。

dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 器件实现了安全段和 RAM 保护。dsPIC33FJ32MC302/304 器件不支持安全段和 RAM 保护。

**注：** 请参考“*dsPIC33F Family Reference Manual*”的“**Section 23. CodeGuard™ Security**” (DS70199) 获得有关 CodeGuard 安全性使用、配置和操作方面的更多信息。

**表 27-3: 32 KB 器件的代码闪存安全段大小**

配置位	BSS<2:0>=x11 0K	BSS<2:0>=x10 1K	BSS<2:0>=x01 4K	BSS<2:0>=x00 8K
<b>SSS&lt;2:0&gt; = x11</b>  <b>0K</b>	VS = 256 IW 000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh	VS = 256 IW 000000h 0001FEh BS = 768 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh	VS = 256 IW 000000h 0001FEh BS = 3840 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh	VS = 256 IW 000000h 0001FEh BS = 7936 IW 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 0057FEh
	GS = 11008 IW	GS = 10240 IW	GS = 7168 IW	GS = 3072 IW
	0157FEh	0157FEh	0157FEh	0157FEh

表 27-4: 64 KB 器件的代码闪存安全段大小

配置位	BSS<2:0>=x11 0K		BSS<2:0>=x10 1K		BSS<2:0>=x01 4K		BSS<2:0>=x00 8K						
SSS<2:0> = x11  0K	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh			
	BS = 768 IW		BS = 768 IW		BS = 3840 IW		BS = 7936 IW		GS = 21760 IW	GS = 20992 IW	GS = 17920 IW	GS = 13824 IW	GS = 13824 IW
	GS = 21760 IW	0157FEh	GS = 20992 IW	0157FEh	GS = 17920 IW	0157FEh	GS = 13824 IW	0157FEh					
SSS<2:0> = x10  4K	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh			
	SS = 3840 IW		SS = 3072 IW		SS = 3840 IW		SS = 7936 IW		GS = 17920 IW	GS = 17920 IW	GS = 17920 IW	GS = 13824 IW	GS = 13824 IW
	GS = 17920 IW	0157FEh	GS = 17920 IW	0157FEh	GS = 17920 IW	0157FEh	GS = 13824 IW	0157FEh					
SSS<2:0> = x01  8K	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh			
	SS = 7936 IW		SS = 7168 IW		SS = 3840 IW		SS = 7936 IW		GS = 13824 IW	GS = 13824 IW	GS = 13824 IW	GS = 13824 IW	GS = 13824 IW
	GS = 13824 IW	0157FEh	GS = 13824 IW	0157FEh	GS = 13824 IW	0157FEh	GS = 13824 IW	0157FEh					
SSS<2:0> = x00  16K	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh			
	SS = 16128 IW		SS = 15360 IW		SS = 3840 IW		SS = 7936 IW		GS = 5632 IW	GS = 5632 IW	GS = 5632 IW	GS = 8192 IW	GS = 5632 IW
	GS = 5632 IW	0157FEh	GS = 5632 IW	0157FEh	GS = 5632 IW	0157FEh	GS = 5632 IW	0157FEh					

表 27-5: 128 KB 器件的代码闪存安全段大小

配置位	BSS<2:0>=x11 0K		BSS<2:0>=x10 1K		BSS<2:0>=x01 4K		BSS<2:0>=x00 8K																			
SSS<2:0> = x11  0K	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	BS = 768 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	BS = 3840 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	BS = 7936 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	GS = 43776 IW	GS = 43008 IW	GS = 39936 IW	GS = 35840 IW	0157FEh	0157FEh	0157FEh	0157FEh				
	GS = 43776 IW	0157FEh	GS = 43008 IW	0157FEh	GS = 39936 IW	0157FEh	GS = 39936 IW	0157FEh	GS = 35840 IW	0157FEh	GS = 35840 IW	0157FEh	GS = 35840 IW	0157FEh												
SSS<2:0> = x10  4K	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	BS = 768 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	SS = 3072 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	BS = 3840 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	SS = 3840 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00ABFEh	GS = 39936 IW	GS = 39936 IW	GS = 39936 IW	GS = 35840 IW	0157FEh	0157FEh	0157FEh	0157FEh		
	GS = 39936 IW	0157FEh	GS = 39936 IW	0157FEh	GS = 39936 IW	0157FEh	GS = 39936 IW	0157FEh	GS = 35840 IW	0157FEh	GS = 35840 IW	0157FEh	GS = 35840 IW	0157FEh												
SSS<2:0> = x01  8K	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	BS = 768 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	SS = 7168 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	BS = 3840 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	SS = 4096 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	BS = 7936 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	GS = 35840 IW	GS = 35840 IW	GS = 35840 IW	GS = 35840 IW	0157FEh	0157FEh	0157FEh	0157FEh
	GS = 35840 IW	0157FEh	GS = 35840 IW	0157FEh	GS = 35840 IW	0157FEh	GS = 35840 IW	0157FEh	GS = 35840 IW	0157FEh	GS = 35840 IW	0157FEh	GS = 35840 IW	0157FEh												
SSS<2:0> = x00  16K	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	BS = 768 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	SS = 15360 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	BS = 3840 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	SS = 12288 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	VS = 256 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	BS = 7936 IW	000000h 0001FEh 000200h 0007FEh 000800h 001FFEh 002000h 003FFEh 004000h 007FFEh 008000h 00FFFEh 010000h	GS = 27648 IW	GS = 27648 IW	GS = 27648 IW	GS = 27648 IW	0157FEh	0157FEh	0157FEh	0157FEh
	GS = 27648 IW	0157FEh	GS = 27648 IW	0157FEh	GS = 27648 IW	0157FEh	GS = 27648 IW	0157FEh	GS = 27648 IW	0157FEh	GS = 27648 IW	0157FEh	GS = 27648 IW	0157FEh												



## 28.0 指令集汇总

**注：** 本数据手册总结了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“*dsPIC33F Family Reference Manual*”的相关章节，该文档可从 [Microchip 网站 \(www.microchip.com\)](http://www.microchip.com) 下载。

dsPIC33F 的指令集与 dsPIC30F 的指令集相同。

大部分指令的长度为一个程序存储字（24 位）。只有三条指令需要两个程序存储单元。

每一条单字指令长 24 位，分为一个指明指令类型的 8 位操作码和进一步指明指令操作的一个或多个操作数。

指令集是高度正交的，分为 5 个基本类别：

- 针对字或字节的操作
- 针对位的操作
- 立即数操作
- DSP 操作
- 控制操作

表 28-1 给出了在描述指令时使用的通用符号。

表 28-2 中的 dsPIC33F 指令集汇总列出了所有指令，并给出了每条指令影响的状态标志。

大多数针对字或字节的 W 寄存器指令（包括桶形移位指令）有三个操作数：

- 第一个源操作数通常是寄存器 “Wb”，不带任何地址修改量
- 第二个源操作数通常是寄存器 “Ws”，带或不带地址修改量
- 存放运算结果的目标寄存器，通常是寄存器 “Wd”，带或不带地址修改量

不过，针对字或字节的文件寄存器指令只有两个操作数：

- 文件寄存器，由 “f” 值指定
- 目标寄存器，可以是文件寄存器 “f” 或 W0 寄存器（用 “WREG” 表示）

大多数位操作指令（包括简单的循环 / 移位指令）有两个操作数：

- W 寄存器（带或不带地址修改量）或文件寄存器（由 “Ws” 或 “f” 的值指定）
- W 寄存器或者文件寄存器中的位（由一个立即数指定，或者由寄存器 “Wb” 的内容间接指定）

涉及数据传送的立即数指令，可以使用下列操作数：

- 要被装入到 W 寄存器或文件寄存器中的立即数（由 “k” 指定）
- 要装入立即数的 W 寄存器或文件寄存器（由 “Wb” 或 “f” 指定）

但是涉及算术或逻辑操作的立即数指令，使用下列操作数：

- 第一个源操作数是不带任何地址修改量的寄存器 “Wb”
- 第二个源操作数是立即数
- 存放操作结果的目标寄存器（仅在与第一个源操作数不同时）通常是带或不带地址修改量的寄存器 “Wd”

MAC 类 DSP 指令可以使用下列部分操作数：

- 要使用的累加器（A 或 B）（必需的操作数）
- 要用作两个操作数的 W 寄存器
- X 和 Y 地址空间预取操作
- X 和 Y 地址空间预取目标寄存器
- 累加器回写目标寄存器

与乘法无关的其他 DSP 指令使用的操作数可能包括：

- 要使用的累加器（必需）
- 源操作数或目标操作数（分别由 Wso 或 Wdo 指定），带或不带地址修改量
- 移位位数，由 W 寄存器 “Wn” 或立即数指定

控制指令可以使用下列操作数：

- 程序存储区地址
- 表读和表写指令的模式

大部分指令是单字指令。几个双字指令的设计，是为了要用 48 位来提供所需全部信息。在第二个字中，8 个 MSb 全为 0。如果指令自身把第二个字当作一条指令来执行的话，它作为一条 NOP 指令来执行。

双字指令执行需要两个指令周期。

大多数单字长指令都在一个指令周期内执行，除非条件测试为真或者指令执行结果改变了程序计数器。对于上述两种特殊情况，指令执行需要两个指令周期，在第二个指令周期中执行一条 NOP 指令。值得注意的例外是

BRA（无条件 / 计算转移）、间接 CALL/GOTO、所有的表读和表写以及 RETURN/RETFIE 指令，它们是单字长指令，但执行需要两个或三个周期。某些涉及跳过后续指令的指令，如果要执行跳过的话，需要两个或三个周期，这取决于被跳过的指令是一条单字指令还是双字指令。此外，双字传送需要两个周期。

**注：** 有关指令集的更多详细信息，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B CN)。

**表 28-1: 操作码说明中使用的符号**

字段	说明
#text	表示由 text 定义的立即数
(text)	表示 text 的内容
[text]	表示由 text 寻址的存储单元
{ }	可选字段或操作
<n:m>	寄存器位域
.b	字节模式选择
.d	双字模式选择
.S	影子寄存器选择
.w	字模式选择（默认）
Acc	两个累加器 {A, B} 之一
AWB	累加器回写目标地址寄存器 $\in \{W13, [W13]+2\}$
bit4	4 位位选择字段（用于字寻址指令） $\in \{0...15\}$
C, DC, N, OV, Z	MCU 状态位：进位、半进位、负、溢出和全零标志位
Expr	绝对地址、标号或表达式（由链接器解析）
f	文件寄存器地址 $\in \{0x0000...0x1FFF\}$
lit1	1 位无符号立即数 $\in \{0,1\}$
lit4	4 位无符号立即数 $\in \{0...15\}$
lit5	5 位无符号立即数 $\in \{0...31\}$
lit8	8 位无符号立即数 $\in \{0...255\}$
lit10	10 位无符号立即数，对于字节模式 $\in \{0...255\}$ ，对于字模式 $\in \{0:1023\}$
lit14	14 位无符号立即数 $\in \{0...16384\}$
lit16	16 位无符号立即数 $\in \{0...65535\}$
lit23	23 位无符号立即数 $\in \{0...8388608\}$ ；LSb 必须为 0
None	字段无需内容，可为空
OA, OB, SA, SB	DSP 状态位：ACCA 溢出、ACCB 溢出、ACCA 饱和与 ACCB 饱和
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）
Wm*Wm	用于平方指令的被乘数和乘数工作寄存器对 $\in \{W4 * W4, W5 * W5, W6 * W6, W7 * W7\}$

表 28-1: 操作码说明中使用的符号 (续)

字段	说明
Wm*Wn	用于 DSP 指令的被乘数和乘数工作寄存器对 $\in \{W4 * W5, W4 * W6, W4 * W7, W5 * W6, W5 * W7, W6 * W7\}$
Wn	16 个工作寄存器中之一 $\in \{W0..W15\}$
Wnd	16 个目标工作寄存器中之一 $\in \{W0..W15\}$
Wns	16 个源工作寄存器中之一 $\in \{W0..W15\}$
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 $\in \{Wns, [Wns], [Wns++] , [Wns--], [++Wns], [--Wns], [Wns+Wb]\}$
Wx	用于 DSP 指令的 X 数据空间预取地址寄存器 $\in \{[W8] + = 6, [W8] + = 4, [W8] + = 2, [W8], [W8] - = 6, [W8] - = 4, [W8] - = 2, [W9] + = 6, [W9] + = 4, [W9] + = 2, [W9], [W9] - = 6, [W9] - = 4, [W9] - = 2, [W9 + W12], none\}$
Wxd	用于 DSP 指令的 X 数据空间预取目标寄存器 $\in \{W4..W7\}$
Wy	用于 DSP 指令的 Y 数据空间预取地址寄存器 $\in \{[W10] + = 6, [W10] + = 4, [W10] + = 2, [W10], [W10] - = 6, [W10] - = 4, [W10] - = 2, [W11] + = 6, [W11] + = 4, [W11] + = 2, [W11], [W11] - = 6, [W11] - = 4, [W11] - = 2, [W11 + W12], 无\}$
Wyd	用于 DSP 指令的 Y 数据空间预取目标寄存器 $\in \{W4..W7\}$

表 28-2: 指令集概述

基本指令编号	汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
1	ADD	ADD Acc	累加器相加	1	1	OA,OB,SA,SB
		ADD f	$f = f + WREG$	1	1	C,DC,N,OV,Z
		ADD f, WREG	$WREG = f + WREG$	1	1	C,DC,N,OV,Z
		ADD #lit10, Wn	$Wd = lit10 + Wd$	1	1	C,DC,N,OV,Z
		ADD Wb, Ws, Wd	$Wd = Wb + Ws$	1	1	C,DC,N,OV,Z
		ADD Wb, #lit5, Wd	$Wd = Wb + lit5$	1	1	C,DC,N,OV,Z
2	ADDC	ADDC f	$f = f + WREG + (C)$	1	1	C,DC,N,OV,Z
		ADDC f, WREG	$WREG = f + WREG + (C)$	1	1	C,DC,N,OV,Z
		ADDC #lit10, Wn	$Wd = lit10 + Wd + (C)$	1	1	C,DC,N,OV,Z
		ADDC Wb, Ws, Wd	$Wd = Wb + Ws + (C)$	1	1	C,DC,N,OV,Z
		ADDC Wb, #lit5, Wd	$Wd = Wb + lit5 + (C)$	1	1	C,DC,N,OV,Z
		3	AND	AND f	$f = f .AND. WREG$	1
AND f, WREG	$WREG = f .AND. WREG$			1	1	N,Z
AND #lit10, Wn	$Wd = lit10 .AND. Wd$			1	1	N,Z
AND Wb, Ws, Wd	$Wd = Wb .AND. Ws$			1	1	N,Z
AND Wb, #lit5, Wd	$Wd = Wb .AND. lit5$			1	1	N,Z
4	ASR			ASR f	f = 算术右移 f	1
		ASR f, WREG	WREG = 算术右移 f	1	1	C,N,OV,Z
		ASR Ws, Wd	Wd = 算术右移 Ws	1	1	C,N,OV,Z
		ASR Wb, Wns, Wnd	Wnd = 将 Wb 算术右移 Wns 位	1	1	N,Z
		ASR Wb, #lit5, Wnd	Wnd = 将 Wb 算术右移 lit5 位	1	1	N,Z
5	BCLR	BCLR f, #bit4	将 f 中的指定位清零	1	1	无
		BCLR Ws, #bit4	将 Ws 中的指定位清零	1	1	无
6	BRA	BRA C, Expr	如果有进位则转移	1	1 (2)	无
		BRA GE, Expr	如果大于或等于则转移	1	1 (2)	无
		BRA GEU, Expr	如果无符号大于或等于则转移	1	1 (2)	无
		BRA GT, Expr	如果大于则转移	1	1 (2)	无
		BRA GTU, Expr	如果无符号大于则转移	1	1 (2)	无
		BRA LE, Expr	如果小于或等于则转移	1	1 (2)	无
		BRA LEU, Expr	如果无符号小于或等于则转移	1	1 (2)	无
		BRA LT, Expr	如果小于则转移	1	1 (2)	无
		BRA LTU, Expr	如果无符号小于则转移	1	1 (2)	无
		BRA N, Expr	如果为负则转移	1	1 (2)	无
		BRA NC, Expr	如果没有进位则转移	1	1 (2)	无
		BRA NN, Expr	如果不为负则转移	1	1 (2)	无
		BRA NOV, Expr	如果不溢出则转移	1	1 (2)	无
		BRA NZ, Expr	如果不为零则转移	1	1 (2)	无
		BRA OA, Expr	如果累加器 A 溢出则转移	1	1 (2)	无
		BRA OB, Expr	如果累加器 B 溢出则转移	1	1 (2)	无
		BRA OV, Expr	如果溢出则转移	1	1 (2)	无
		BRA SA, Expr	如果累加器 A 饱和则转移	1	1 (2)	无
		BRA SB, Expr	如果累加器 B 饱和则转移	1	1 (2)	无
		BRA Expr	无条件转移	1	2	无
BRA Z, Expr	如果为零则转移	1	1 (2)	无		
BRA Wn	计算转移	1	2	无		
7	BSET	BSET f, #bit4	将 f 中的指定位置 1	1	1	无
		BSET Ws, #bit4	将 Ws 中的指定位置 1	1	1	无
8	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
9	BTG	BTG f, #bit4	将 f 中的指定位翻转	1	1	无
		BTG Ws, #bit4	将 Ws 中的指定位翻转	1	1	无

表 28-2: 指令集概述 (续)

基本指令编号	汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
10	BTSC	BTSC $f, \#bit4$	对 $f$ 中的指定位进行测试, 如果为零则跳过	1	1 (2 或 3)	无
		BTSC $Ws, \#bit4$	对 $Ws$ 中的指定位进行测试, 如果为零则跳过	1	1 (2 或 3)	无
11	BTSS	BTSS $f, \#bit4$	对 $f$ 中的指定位进行测试, 如果为 1 则跳过	1	1 (2 或 3)	无
		BTSS $Ws, \#bit4$	对 $Ws$ 中的指定位进行测试, 如果为 1 则跳过	1	1 (2 或 3)	无
12	BTST	BTST $f, \#bit4$	对 $f$ 中的指定位进行测试	1	1	Z
		BTST.C $Ws, \#bit4$	对 $Ws$ 中的指定位进行测试, 并将其值存储到 C	1	1	C
		BTST.Z $Ws, \#bit4$	对 $Ws$ 中的指定位进行测试, 并将其反码存储到 Z	1	1	Z
		BTST.C $Ws, Wb$	对 $Ws < Wb >$ 位进行测试, 并将其值存储到 C	1	1	C
		BTST.Z $Ws, Wb$	对 $Ws < Wb >$ 位进行测试, 并将其反码存储到 Z	1	1	Z
13	BTSTS	BTSTS $f, \#bit4$	对 $f$ 中的指定位进行测试, 并将 $f$ 中的该位置 1	1	1	Z
		BTSTS.C $Ws, \#bit4$	对 $Ws$ 中的指定位进行测试, 并将其值存储到 C, 然后将 $Ws$ 中的该位置 1	1	1	C
		BTSTS.Z $Ws, \#bit4$	对 $Ws$ 中的指定位进行测试, 并将其反码存储到 Z, 然后将 $Ws$ 中的该位置 1	1	1	Z
14	CALL	CALL $lit23$	调用子程序	2	2	无
		CALL $Wn$	间接调用子程序	1	2	无
15	CLR	CLR $f$	$f = 0x0000$	1	1	无
		CLR WREG	WREG = $0x0000$	1	1	无
		CLR $Ws$	$Ws = 0x0000$	1	1	无
		CLR $Acc, Wx, Wxd, Wy, Wyd, AWB$	清零累加器	1	1	OA, OB, SA, SB
16	CLRWDT	CLRWDT	将看门狗定时器清零	1	1	WDTO, Sleep
17	COM	COM $f$	$f = \bar{f}$	1	1	N, Z
		COM $f, WREG$	WREG = $\bar{f}$	1	1	N, Z
		COM $Ws, Wd$	$Wd = Ws$	1	1	N, Z
18	CP	CP $f$	比较 $f$ 和 WREG	1	1	C, DC, N, OV, Z
		CP $Wb, \#lit5$	比较 $Wb$ 和 $lit5$	1	1	C, DC, N, OV, Z
		CP $Wb, Ws$	比较 $Wb$ 和 $Ws$ ( $Wb - Ws$ )	1	1	C, DC, N, OV, Z
19	CP0	CP0 $f$	比较 $f$ 和 $0x0000$	1	1	C, DC, N, OV, Z
		CP0 $Ws$	比较 $Ws$ 和 $0x0000$	1	1	C, DC, N, OV, Z
20	CPB	CPB $f$	带借位比较 $f$ 和 WREG	1	1	C, DC, N, OV, Z
		CPB $Wb, \#lit5$	带借位比较 $Wb$ 和 $lit5$	1	1	C, DC, N, OV, Z
		CPB $Wb, Ws$	带借位比较 $Wb$ 和 $Ws$ ( $Wb - Ws - C$ )	1	1	C, DC, N, OV, Z
21	CPSEQ	CPSEQ $Wb, Wn$	比较 $Wb$ 和 $Wn$ , 如果相等则跳过	1	1 (2 或 3)	无
22	CPSGT	CPSGT $Wb, Wn$	比较 $Wb$ 和 $Wn$ , 如果大于则跳过	1	1 (2 或 3)	无
23	CPSLT	CPSLT $Wb, Wn$	比较 $Wb$ 和 $Wn$ , 如果小于则跳过	1	1 (2 或 3)	无
24	CPSNE	CPSNE $Wb, Wn$	比较 $Wb$ 和 $Wn$ , 如果不相等则跳过	1	1 (2 或 3)	无
25	DAW	DAW $Wn$	$Wn =$ 十进制调整 $Wn$	1	1	C
26	DEC	DEC $f$	$f = f - 1$	1	1	C, DC, N, OV, Z
		DEC $f, WREG$	WREG = $f - 1$	1	1	C, DC, N, OV, Z
		DEC $Ws, Wd$	$Wd = Ws - 1$	1	1	C, DC, N, OV, Z
27	DEC2	DEC2 $f$	$f = f - 2$	1	1	C, DC, N, OV, Z
		DEC2 $f, WREG$	WREG = $f - 2$	1	1	C, DC, N, OV, Z
		DEC2 $Ws, Wd$	$Wd = Ws - 2$	1	1	C, DC, N, OV, Z
28	DISI	DISI $\#lit14$	在 $k$ 个指令周期内禁止中断	1	1	无

表 28-2: 指令集概述 (续)

基本指令编号	汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
29	DIV	DIV.S Wm, Wn	有符号 16/16 位整数除法	1	18	N,Z,C,OV
		DIV.SD Wm, Wn	有符号 32/16 位整数除法	1	18	N,Z,C,OV
		DIV.U Wm, Wn	无符号 16/16 位整数除法	1	18	N,Z,C,OV
		DIV.UD Wm, Wn	无符号 32/16 位整数除法	1	18	N,Z,C,OV
30	DIVF	DIVF Wm, Wn	有符号 16/16 位小数除法	1	18	N,Z,C,OV
31	DO	DO #lit14, Expr	执行 DO 循环代码到 PC + Expr。执行次数为 lit14 + 1 次	2	2	无
		DO Wn, Expr	执行 DO 循环代码到 PC + Expr。执行次数为 (Wn) + 1 次	2	2	无
32	ED	ED Wm*Wm, Acc, Wx, Wy, Wxd	欧几里德距离 (无累加)	1	1	OA,OB,OAB,SA,SB,SAB
33	EDAC	EDAC Wm*Wm, Acc, Wx, Wy, Wxd	欧几里德距离	1	1	OA,OB,OAB,SA,SB,SAB
34	EXCH	EXCH Wns, Wnd	交换 Wns 和 Wnd 的内容	1	1	无
35	FBCL	FBCL Ws, Wnd	从左边 (MSb) 查找第一个位变化	1	1	C
36	FF1L	FF1L Ws, Wnd	从左边 (MSb) 查找第一个 1	1	1	C
37	FF1R	FF1R Ws, Wnd	从右边 (LSb) 查找第一个 1	1	1	C
38	GOTO	GOTO Expr	转移到地址	2	2	无
		GOTO Wn	间接转移到地址	1	2	无
39	INC	INC f	f = f + 1	1	1	C,DC,N,OV,Z
		INC f, WREG	WREG = f + 1	1	1	C,DC,N,OV,Z
		INC Ws, Wd	Wd = Ws + 1	1	1	C,DC,N,OV,Z
40	INC2	INC2 f	f = f + 2	1	1	C,DC,N,OV,Z
		INC2 f, WREG	WREG = f + 2	1	1	C,DC,N,OV,Z
		INC2 Ws, Wd	Wd = Ws + 2	1	1	C,DC,N,OV,Z
41	IOR	IOR f	f = f .IOR.WREG	1	1	N,Z
		IOR f, WREG	WREG = f .IOR.WREG	1	1	N,Z
		IOR #lit10, Wn	Wd = lit10 .IOR.Wd	1	1	N,Z
		IOR Wb, Ws, Wd	Wd = Wb .IOR.Ws	1	1	N,Z
		IOR Wb, #lit5, Wd	Wd = Wb .IOR.lit5	1	1	N,Z
42	LAC	LAC Wso, #Slit4, Acc	装载累加器	1	1	OA,OB,OAB,SA,SB,SAB
43	LNK	LNK #lit14	分配堆栈帧	1	1	无
44	LSR	LSR f	f = 逻辑右移 f	1	1	C,N,OV,Z
		LSR f, WREG	WREG = 逻辑右移 f	1	1	C,N,OV,Z
		LSR Ws, Wd	Wd = 逻辑右移 Ws	1	1	C,N,OV,Z
		LSR Wb, Wns, Wnd	Wnd = 将 Wb 逻辑右移 Wns 位	1	1	N,Z
		LSR Wb, #lit5, Wnd	Wnd = 将 Wb 逻辑右移 lit5 位	1	1	N,Z
45	MAC	MAC Wm*Wn, Acc, Wx, Wxd, Wy, Wyd, AWB	相乘并累加	1	1	OA,OB,OAB,SA,SB,SAB
		MAC Wm*Wm, Acc, Wx, Wxd, Wy, Wyd	平方并累加	1	1	OA,OB,OAB,SA,SB,SAB
46	MOV	MOV f, Wn	将 f 中的内容送入 Wn	1	1	无
		MOV f	将 f 中的内容送入目标文件寄存器	1	1	N,Z
		MOV f, WREG	将 f 中的内容送入 WREG	1	1	N,Z
		MOV #lit16, Wn	将 16 位立即数送入 Wn	1	1	无
		MOV.b #lit8, Wn	将 8 位立即数送入 Wn	1	1	无
		MOV Wn, f	将 Wn 中的内容送入 f	1	1	无
		MOV Wso, Wdo	将 Ws 中的内容送入 Wd	1	1	无
		MOV WREG, f	将 WREG 中的内容送入 f	1	1	N,Z
		MOV.D 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
47	MOVSAC	MOVSAC Acc, Wx, Wxd, Wy, Wyd, AWB	预取操作数并保存累加器	1	1	无

表 28-2: 指令集概述 (续)

基本指令编号	汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
48	MPY	MPY Wm*Wn, Acc, Wx, Wxd, Wy, Wyd	Wm 与 Wn 相乘, 结果存入累加器	1	1	OA,OB,OAB,SA,SB,SAB
		MPY Wm*Wm, Acc, Wx, Wxd, Wy, Wyd	Wm 求平方, 结果存入累加器	1	1	OA,OB,OAB,SA,SB,SAB
49	MPY.N	MPY.N Wm*Wn, Acc, Wx, Wxd, Wy, Wyd	Wm 与 Wn 相乘并取反, 结果存入累加器	1	1	无
50	MSC	MSC Wm*Wm, Acc, Wx, Wxd, Wy, Wyd , AWB	相乘再从累加器中减去	1	1	OA,OB,OAB,SA,SB,SAB
51	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
52	NEG	NEG Acc	将累加器内容求补	1	1	OA,OB,OAB,SA,SB,SAB
		NEG f	$f = \bar{f} + 1$	1	1	C,DC,N,OV,Z
		NEG f, WREG	$WREG = \bar{f} + 1$	1	1	C,DC,N,OV,Z
		NEG Ws, Wd	$Wd = \bar{Ws} + 1$	1	1	C,DC,N,OV,Z
53	NOP	NOP	空操作	1	1	无
		NOPR	空操作	1	1	无
54	POP	POP f	将栈顶 (TOS) 的内容弹出到 f	1	1	无
		POP Wdo	将栈顶 (TOS) 的内容弹出到 Wdo	1	1	无
		POP.D Wnd	将栈顶 (TOS) 的内容弹出到 W(nd):W(nd + 1)	1	2	无
		POP.S	将影子寄存器的内容弹出到主寄存器	1	1	全部
55	PUSH	PUSH f	将 f 的内容压入栈顶 (TOS)	1	1	无
		PUSH Wso	将 Wso 的内容压入栈顶 (TOS)	1	1	无
		PUSH.D Wns	将 W(ns):W(ns + 1) 的双字内容压入栈顶 (TOS)	1	2	无
		PUSH.S	将主寄存器中的双字内容压入影子寄存器	1	1	无
56	PWRSVAV	PWRSVAV #lit1	进入休眠或空闲模式	1	1	WDTO,Sleep
57	RCALL	RCALL Expr	相对调用	1	2	无
		RCALL Wn	计算调用	1	2	无
58	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
59	RESET	RESET	软件器件复位	1	1	无
60	RETFIE	RETFIE	从中断返回	1	3 (2)	无
61	RETLW	RETLW #lit10, Wn	返回并将立即数存入 Wn	1	3 (2)	无
62	RETURN	RETURN	从子程序返回	1	3 (2)	无
63	RLC	RLC f	f = 对 f 执行带进位的循环左移	1	1	C,N,Z
		RLC f, WREG	WREG = 对 f 执行带进位的循环左移	1	1	C,N,Z
		RLC Ws, Wd	Wd = 对 Ws 执行带进位的循环左移	1	1	C,N,Z
64	RLNC	RLNC f	f = 循环左移 f (不带进位)	1	1	N,Z
		RLNC f, WREG	WREG = 循环左移 f (不带进位)	1	1	N,Z
		RLNC Ws, Wd	Wd = 循环左移 Ws (不带进位)	1	1	N,Z
65	RRC	RRC f	f = 对 f 执行带进位的循环右移	1	1	C,N,Z
		RRC f, WREG	WREG = 对 f 执行带进位的循环右移	1	1	C,N,Z
		RRC Ws, Wd	Wd = 对 Ws 执行带进位的循环右移	1	1	C,N,Z

表 28-2: 指令集概述 (续)

基本指令编号	汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
66	RRNC	RRNC <i>f</i>	$f = \text{循环右移 } f \text{ (不带进位)}$	1	1	N,Z
		RRNC <i>f</i> , WREG	WREG = 循环右移 <i>f</i> (不带进位)	1	1	N,Z
		RRNC <i>Ws</i> , <i>Wd</i>	$Wd = \text{循环右移 } Ws \text{ (不带进位)}$	1	1	N,Z
67	SAC	SAC <i>Acc</i> , #Slit4, <i>Wdo</i>	保存累加器内容	1	1	无
		SAC.R <i>Acc</i> , #Slit4, <i>Wdo</i>	保存舍入后的累加器内容	1	1	无
68	SE	SE <i>Ws</i> , <i>Wnd</i>	$Wnd = \text{符号扩展后的 } Ws$	1	1	C,N,Z
69	SETM	SETM <i>f</i>	$f = 0xFFFF$	1	1	无
		SETM WREG	WREG = 0xFFFF	1	1	无
		SETM <i>Ws</i>	$Ws = 0xFFFF$	1	1	无
70	SFTAC	SFTAC <i>Acc</i> , <i>Wn</i>	对累加器算术移位 ( <i>Wn</i> ) 次	1	1	OA,OB,OAB,SA,SB,SAB
		SFTAC <i>Acc</i> , #Slit6	对累加器算术移位 Slit6 次	1	1	OA,OB,OAB,SA,SB,SAB
71	SL	SL <i>f</i>	$f = \text{左移 } f$	1	1	C,N,OV,Z
		SL <i>f</i> , WREG	WREG = 左移 <i>f</i>	1	1	C,N,OV,Z
		SL <i>Ws</i> , <i>Wd</i>	$Wd = \text{左移 } Ws$	1	1	C,N,OV,Z
		SL <i>Wb</i> , <i>Wns</i> , <i>Wnd</i>	$Wnd = \text{将 } Wb \text{ 左移 } Wns \text{ 位}$	1	1	N,Z
		SL <i>Wb</i> , #lit5, <i>Wnd</i>	$Wnd = \text{将 } Wb \text{ 左移 lit5 位}$	1	1	N,Z
72	SUB	SUB <i>Acc</i>	从累加器减去	1	1	OA,OB,OAB,SA,SB,SAB
		SUB <i>f</i>	$f = f - WREG$	1	1	C,DC,N,OV,Z
		SUB <i>f</i> , WREG	WREG = $f - WREG$	1	1	C,DC,N,OV,Z
		SUB #lit10, <i>Wn</i>	$Wn = Wn - \text{lit10}$	1	1	C,DC,N,OV,Z
		SUB <i>Wb</i> , <i>Ws</i> , <i>Wd</i>	$Wd = Wb - Ws$	1	1	C,DC,N,OV,Z
		SUB <i>Wb</i> , #lit5, <i>Wd</i>	$Wd = Wb - \text{lit5}$	1	1	C,DC,N,OV,Z
73	SUBB	SUBB <i>f</i>	$f = f - WREG - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBB <i>f</i> , WREG	WREG = $f - WREG - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBB #lit10, <i>Wn</i>	$Wn = Wn - \text{lit10} - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBB <i>Wb</i> , <i>Ws</i> , <i>Wd</i>	$Wd = Wb - Ws - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBB <i>Wb</i> , #lit5, <i>Wd</i>	$Wd = Wb - \text{lit5} - (\overline{C})$	1	1	C,DC,N,OV,Z
74	SUBR	SUBR <i>f</i>	$f = WREG - f$	1	1	C,DC,N,OV,Z
		SUBR <i>f</i> , WREG	WREG = $WREG - f$	1	1	C,DC,N,OV,Z
		SUBR <i>Wb</i> , <i>Ws</i> , <i>Wd</i>	$Wd = Ws - Wb$	1	1	C,DC,N,OV,Z
		SUBR <i>Wb</i> , #lit5, <i>Wd</i>	$Wd = \text{lit5} - Wb$	1	1	C,DC,N,OV,Z
75	SUBBR	SUBBR <i>f</i>	$f = WREG - f - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBBR <i>f</i> , WREG	WREG = $WREG - f - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBBR <i>Wb</i> , <i>Ws</i> , <i>Wd</i>	$Wd = Ws - Wb - (\overline{C})$	1	1	C,DC,N,OV,Z
		SUBBR <i>Wb</i> , #lit5, <i>Wd</i>	$Wd = \text{lit5} - Wb - (\overline{C})$	1	1	C,DC,N,OV,Z
76	SWAP	SWAP.b <i>Wn</i>	$Wn = \text{半字节交换 } Wn \text{ 内容}$	1	1	无
		SWAP <i>Wn</i>	$Wn = \text{将 } Wn \text{ 的两个字节相交换}$	1	1	无
77	TBLRDH	TBLRDH <i>Ws</i> , <i>Wd</i>	将程序存储单元的 <23:16> 读入 $Wd<7:0>$	1	2	无
78	TBLRDL	TBLRDL <i>Ws</i> , <i>Wd</i>	将程序存储单元的 <15:0> 读入 $Wd$	1	2	无
79	TBLWTH	TBLWTH <i>Ws</i> , <i>Wd</i>	将 $Ws<7:0>$ 写入程序存储单元的 <23:16>	1	2	无
80	TBLWTL	TBLWTL <i>Ws</i> , <i>Wd</i>	将 $Ws$ 写入程序存储单元的 <15:0>	1	2	无
81	ULNK	ULNK	释放堆栈帧	1	1	无
82	XOR	XOR <i>f</i>	$f = f .XOR.WREG$	1	1	N,Z
		XOR <i>f</i> , WREG	WREG = $f .XOR.WREG$	1	1	N,Z
		XOR #lit10, <i>Wn</i>	$Wd = \text{lit10} .XOR.Wd$	1	1	N,Z
		XOR <i>Wb</i> , <i>Ws</i> , <i>Wd</i>	$Wd = Wb .XOR.Ws$	1	1	N,Z
		XOR <i>Wb</i> , #lit5, <i>Wd</i>	$Wd = Wb .XOR. \text{lit5}$	1	1	N,Z
83	ZE	ZE <i>Ws</i> , <i>Wnd</i>	$Wnd = \text{零扩展后的 } Ws$	1	1	C,Z,N



## 29.0 开发支持

一系列硬件及软件开发工具对 PIC® 单片机提供支持：

- 集成开发环境
  - MPLAB® IDE 软件
- 汇编器 / 编译器 / 链接器
  - MPASM™ 汇编器
  - MPLAB C18 和 MPLAB C30 C 编译器
  - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
  - MPLAB ASM30 汇编器 / 链接器 / 库
- 模拟器
  - MPLAB SIM 软件模拟器
- 仿真器
  - MPLAB ICE 2000 在线仿真器
  - MPLAB REAL ICE™ 在线仿真器
- 在线调试器
  - MPLAB ICD 2
- 器件编程器
  - PICSTART® Plus 开发编程器
  - MPLAB PM3 器件编程器
  - PICKit™ 2 开发编程器
- 低成本演示和开发板及评估工具包

## 29.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16 位单片机市场提供了前所未有的易于使用的软件开发平台。MPLAB IDE 是基于 Windows® 操作系统的应用软件，包括：

- 一个包含所有调试工具的图形界面
  - 模拟器
  - 编程器（单独销售）
  - 仿真器（单独销售）
  - 在线调试器（单独销售）
- 具有彩色上下文代码显示的全功能编辑器
- 多项目管理器
- 内容可直接编辑的可定制式数据窗口
- 高级源代码调试
- 可视化器件初始化程序，便于进行寄存器的初始化
- 鼠标停留在变量上进行查看的功能
- 通过拖放把变量从源代码窗口拉到观察窗口
- 丰富的在线帮助
- 集成了可选的第三方工具，如 HI-TECH 软件 C 编译器和 IAR C 编译器

MPLAB IDE 可以让您：

- 编辑源文件（汇编语言或 C 语言）
- 点击一次即可完成汇编（或编译）并将代码下载到 PIC MCU 仿真器和模拟器工具中（自动更新所有项目信息）
- 可使用如下各项进行调试：
  - 源文件（汇编语言或 C 语言）
  - 混合汇编语言和 C 语言
  - 机器码

MPLAB IDE 在单个开发范例中支持使用多种调试工具，包括从成本效益高的模拟器到低成本的在线调试器，再到全功能的仿真器。这样缩短了用户升级到更加灵活而功能更强大的工具时的学习时间。

## 29.2 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于所有的 PIC MCU。

MPASM 汇编器可生成用于 MPLINK 目标链接器的可重定位目标文件、Intel® 标准 HEX 文件、详细描述存储器使用状况和符号参考的 MAP 文件、包含源代码行及生成机器码的绝对 LST 文件以及用于调试的 COFF 文件。

MPASM 汇编器具有如下特征：

- 集成在 MPLAB IDE 项目中
- 用户定义的宏可简化汇编代码
- 对多用途源文件进行条件汇编
- 允许完全控制汇编过程的指令

## 29.3 MPLAB C18 和 MPLAB C30 C 编译器

MPLAB C18 和 MPLAB C30 代码开发系统是完全的 ANSI C 编译器，分别适用于 Microchip 的 PIC18 和 PIC24 系列单片机及 dsPIC30F 和 dsPIC33 系列数字信号控制器。这些编译器可提供其他编译器并不具备的强大的集成功能和出众的代码优化能力，且使用方便。

为便于源代码调试，编译器提供了针对 MPLAB IDE 调试器的优化符号信息。

## 29.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器包含了由 MPASM 汇编器、MPLAB C18 C 编译器产生的可重定位目标。通过使用链接器脚本中的指令，它还可链接预编译库中的可重定位目标。

MPLIB 目标库管理器管理预编译代码库文件的创建和修改。当从源文件调用库中的一段子程序时，只有包含此子程序的模块被链接到应用程序。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征：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

## 29.5 MPLAB ASM30 汇编器、链接器和库管理器

MPLAB ASM30 汇编器为 dsPIC30F 器件提供转换自符号汇编语言的可重定位机器码。MPLAB C30 C 编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特征：

- 支持整个 dsPIC30F 指令集
- 支持定点数据和浮点数据
- 命令行界面
- 丰富的指令集
- 灵活的宏语言
- MPLAB IDE 兼容性

## 29.6 MPLAB SIM 软件模拟器

MPLAB SIM 软件模拟器在指令级对 PIC MCU 和 dsPIC® DSC 进行模拟，使得用户可以在 PC 主机的环境下进行代码开发。对于任何给定的指令，用户均可对数据区进行检查或修改，并通过各种触发机制来产生激励。可以将各寄存器的情况记录在文件中，以便进行进一步地运行时分析。跟踪缓冲器和逻辑分析器的显示使模拟器还能记录和跟踪程序的执行、I/O 的动作、大部分的外设及内部寄存器的状况。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C18 和 MPLAB C30 C 编译器以及 MPASM 和 MPLAB ASM30 汇编器的符号调试。该软件模拟器可用于在硬件实验室环境外灵活地开发和调试代码，是一款完美且经济的软件开发工具。

## 29.7 MPLAB ICE 2000 高性能在线仿真器

MPLAB ICE 2000 在线仿真器旨在为产品开发工程师提供一整套用于 PIC 单片机的设计工具。MPLAB ICE 2000 在线仿真器的软件控制由 MPLAB 集成开发环境平台提供，它允许在单一环境下进行编辑、编译、下载以及源代码调试。

MPLAB ICE 2000 是全功能仿真器系统，它具有增强的跟踪、触发和数据监控功能。处理器模块可插拔，使系统可轻松进行重新配置以适应各种不同处理器的仿真需要。MPLAB ICE 2000 在线仿真器的架构允许对其进行扩展以支持新的 PIC 单片机。

MPLAB ICE 2000 在线仿真器系统设计为一款实时仿真系统，该仿真系统具备通常只有昂贵的开发工具中才有的高级功能。选择 PC 平台和 Microsoft® Windows® 32 位操作系统可使这些功能在一个简单而统一的应用中得到很好的利用。

## 29.8 MPLAB REAL ICE 在线仿真器系统

MPLAB REAL ICE 在线仿真器系统是 Microchip 针对其闪存 DSC 和 MCU 器件而推出的新一代高速仿真器。结合 MPLAB 集成开发环境 (IDE) 所具有的易于使用且功能强大的图形用户界面，该仿真器可对 PIC® 闪存 MCU 和 dsPIC® DSC 进行调试和编程。IDE 是随每个工具包一起提供的。

MPLAB REAL ICE 探针通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与常用 MPLAB ICD 2 系统兼容的连接器 (RJ11) 或新型抗噪声、高速低压差分信号 (LVDS) 互连电缆 (CAT5) 与目标板相连。

可通过 MPLAB IDE 下载将来版本的固件，对 MPLAB REAL ICE 进行现场升级。在即将推出的 MPLAB IDE 版本中，会支持许多新器件，还将增加一些新特性，如软件断点和汇编代码跟踪等。在同类仿真器中，MPLAB REAL ICE 的优势十分明显：低成本、高速仿真、实时变量监视、跟踪分析、复杂断点、耐用的探针接口及较长 (长达 3 米) 的互连电缆。

## 29.9 MPLAB ICD 2 在线调试器

Microchip 的在线调试器 MPLAB ICD 2 是一款功能强大而成本低廉的运行时开发工具，通过 RS-232 或高速 USB 接口与 PC 主机相连。该工具基于闪存 PIC MCU，可用于开发本系列及其他 PIC MCU 和 dsPIC DSC。MPLAB ICD 2 使用了闪存器件中内建的在线调试功能。该功能结合 Microchip 的在线串行编程 (In-Circuit Serial Programming™, ICSP™) 协议，可在 MPLAB 集成开发环境的图形用户界面上提供成本效益很高的在线闪存调试。这使设计人员可通过设置断点、单步运行以及对变量、CPU 状态以及外设寄存器进行监视的方法实现源代码的开发和调试。其全速运行特性可对硬件和应用进行实时测试。MPLAB ICD 2 还可用作某些 PIC 器件的开发编程器。

## 29.10 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款通用的、符合 CE 规范的器件编程器，其可编程电压设置在 VDDMIN 和 VDDMAX 之间时可靠性最高。它有一个用来显示菜单和错误信息的大 LCD 显示器 (128 x 64)，以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP™ 电缆。在单机模式下，MPLAB PM3 器件编程器不必与 PC 相连即可对 PIC 器件进行读取、验证和编程。在该模式下它还可设置代码保护。MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。MPLAB PM3 具备高速通信能力以及优化算法，可对存储器很大的器件进行快速编程，它还采用 SD/MMC 卡用作文件存储及数据安全应用。

### 29.11 PICSTART Plus 开发编程器

PICSTART Plus 开发编程器是一款易于使用而成本低廉的原型编程器。它通过 COM (RS-232) 端口与 PC 相连。MPLAB 集成开发环境软件使得该编程器的使用简便、高效。PICSTART Plus 开发编程器支持采用 DIP 封装的大部分 PIC 器件，其引脚数最多可达 40 个。引脚数更多的器件，如 PIC16C92X 和 PIC17C76X，可通过连接一个转接插槽来获得支持。PICSTART Plus 开发编程器符合 CE 规范。

### 29.12 PICkit 2 开发编程器

PICkit™ 2 开发编程器是一个低成本编程器；对于某些选定闪存器件，它也是一个调试器，通过其易于使用的接口可对众多 Microchip 的低档、中档和 PIC18F 系列闪存单片机进行编程。PICkit 2 入门工具包中包含一个有实验布线区的开发板、十二堂系列课程、软件和 HI-TECH 的 PICC™ Lite C 编译器，有助于用户快速掌握 PIC® 单片机的使用。这一工具包为使用 Microchip 功能强大的中档闪存系列单片机进行编程、评估和应用开发，提供了所需的一切。

### 29.13 演示、开发和评估板

有许多演示、开发和评估板可用于各种 PIC MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于测试和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、RS-232 接口、LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 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。

有关演示、开发和评估工具包的完整列表，请查阅 Microchip 公司网页 ([www.microchip.com](http://www.microchip.com))。

### 30.0 电气特性

本章将对 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 的电气特性进行概括介绍。其余信息在本文档的将来版本中提供。

下面列出了 dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的绝对最大额定值。器件长时间工作在最大额定值条件下可能会影响其可靠性。我们不建议使器件在或超过本规范指定的最大额定值条件下运行。

#### 绝对最大额定值<sup>(1)</sup>

环境温度	-40°C 至 +125°C
储存温度	-65°C 至 +150°C
VDD 引脚相对于 VSS 的电压	-0.3V 至 +4.0V
任一模拟 / 数字引脚和 MCLR 引脚相对于 VSS 的电压	-0.3V 至 (VDD + 0.3V)
任一仅用作数字功能的引脚相对于 VSS 的电压	-0.3V 至 +5.6V
VDDCORE 引脚相对于 VSS 的电压	2.25V 至 2.75V
流出 VSS 引脚的最大电流	300 mA
流入 VDD 引脚的最大电流 <sup>(2)</sup>	250 mA
任一 I/O 引脚的最大输出灌电流 <sup>(3)</sup>	4 mA
任一 I/O 引脚的最大输出拉电流 <sup>(3)</sup>	4 mA
所有端口的最大灌电流	200 mA
所有端口的最大拉电流 <sup>(2)</sup>	200 mA

**注 1:** 如果器件工作条件超过上述“绝对最大额定值”，可能引起器件永久性损坏。这仅是极限参数，我们不建议器件工作在极限值甚至超过上述极限值。器件长时间工作在极限条件下可能会影响其可靠性。

**2:** 允许的最大电流由器件最大功耗决定（见表 30-2）。

**3:** CLKOUT 引脚例外，其灌 / 拉电流为 25 mA，另外 VREF+、VREF-、SCLx、SDAx、PGCx 和 PGDx 引脚的灌 / 拉电流为 12 mA。

30.1 直流特性

表 30-1: 工作 MIPS 与电压

特性	VDD 范围 (单位: V)	温度范围 (单位: °C)	最大 MIPS
			dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04
	3.0-3.6V	-40°C 至 +85°C	40
	3.0-3.6V	-40°C 至 +125°C	35

表 30-2: 热工作条件

参数	符号	最小值	典型值	最大值	单位
工业级温度器件					
工作结温范围	TJ	-40	—	+125	°C
工作环境温度范围	TA	-40	—	+85	°C
扩展级温度器件					
工作结温范围	TJ	-40	—	+140	°C
工作环境温度范围	TA	-40	—	+125	°C
功耗: 芯片内部功耗: $P_{INT} = V_{DD} \times (I_{DD} - \sum I_{OH})$ I/O 引脚功耗: $I/O = \sum (\{V_{DD} - V_{OH}\} \times I_{OH}) + \sum (V_{OL} \times I_{OL})$	PD	PINT + PI/O			W
最大允许功耗	PDMAX	$(T_J - T_A)/\theta_{JA}$			W

表 30-3: 热封装特性

特性	符号	典型值	最大值	单位	注
封装热阻, 44 引脚 QFN	$\theta_{JA}$	62.4	—	°C/W	1
封装热阻, 44 引脚 TFQP	$\theta_{JA}$	60	—	°C/W	1
封装热阻, 28 引脚 SPDIP	$\theta_{JA}$	108	—	°C/W	1
封装热阻, 28 引脚 SOIC	$\theta_{JA}$	80.2	—	°C/W	1
封装热阻, 28 引脚 QFN-S	$\theta_{JA}$	32	—	°C/W	1

注 1: 通过封装模拟获得结点与环境的热阻值 ( $\theta_{JA}$ )。

表 30-4: 直流特性的温度和电压规范

直流特性		标准工作条件: 3.0V 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
工作电压							
DC10	电源电压						
	VDD		3.0	—	3.6	V	工业级和扩展级
DC12	VDR	RAM 数据保持电压 <sup>(2)</sup>	1.1	1.3	1.8	V	
DC16	VPOR	VDD 启动电压 (确保内部上电复位信号)	—	—	VSS	V	
DC17	SVDD	VDD 上升率 (确保内部上电复位信号)	0.03	—	—	V/ms	0-3.0V/0.1s
DC18	VCORE	VDD 内核 <sup>(3)</sup> 内部稳压器电压	2.25	—	2.75	V	电压取决于负载、温度和 VDD

- 注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。  
 2: 这是在不丢失 RAM 数据的前提下, VDD 的下限值。  
 3: 这些参数为特性值, 但生产时未经测试。

表 30-5: 直流特性: 工作电流 (IDD)

直流特性		标准工作条件: 3.0V 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	条件	
工作电流 (IDD) (2)					
DC20d	24	30	mA	-40°C	3.3V 10 MIPS
DC20a	27	30	mA	+25°C	
DC20b	27	30	mA	+85°C	
DC20c	27	35	mA	+125°C	
DC21d	30	40	mA	-40°C	3.3V 16 MIPS
DC21a	37	40	mA	+25°C	
DC21b	32	45	mA	+85°C	
DC21c	33	45	mA	+125°C	3.3V 20 MIPS
DC22d	35	50	mA	-40°C	
DC22a	38	50	mA	+25°C	
DC22b	38	55	mA	+85°C	
DC22c	39	55	mA	+125°C	3.3V 30 MIPS
DC23d	47	70	mA	-40°C	
DC23a	48	70	mA	+25°C	
DC23b	48	70	mA	+85°C	
DC23c	48	70	mA	+125°C	3.3V 40 MIPS
DC24d	56	90	mA	-40°C	
DC24a	56	90	mA	+25°C	
DC24b	54	90	mA	+85°C	
DC24c	54	80	mA	+125°C	3.3V 35 MIPS

- 注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。
- 注 2: 供电电流主要受工作电压和频率的影响。其他因素如 I/O 引脚负载和开关速率、振荡器类型、内部代码执行模式以及温度也对电流消耗有影响。所有 IDD 测量的测试条件为: OSC1 使用满幅的外部方波进行驱动。所有 I/O 引脚配置为输入且被拉到 VSS。MCLR = VDD, WDT 和 FSCM 被禁止。CPU、SRAM、程序存储器 and 数据存储器处于工作状态。所有外设模块都不工作; 但是, 仍然为每个外设提供时钟 (PMD 的所有位均被清零)。



表 30-6: 直流特性: 空闲电流 (IDLE)

直流特性			标准工作条件: 3.0V 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	条件		
空闲电流 (IDLE): 内核不工作、时钟工作时的基本电流 <sup>(2)</sup>						
DC40d	3	25	mA	-40°C	3.3V	10 MIPS
DC40a	3	25	mA	+25°C		
DC40b	3	25	mA	+85°C		
DC40c	3	25	mA	+125°C		
DC41d	4	25	mA	-40°C	3.3V	16 MIPS
DC41a	4	25	mA	+25°C		
DC41b	5	25	mA	+85°C		
DC41c	5	25	mA	+125°C	3.3V	20 MIPS
DC42d	6	25	mA	-40°C		
DC42a	6	25	mA	+25°C		
DC42b	7	25	mA	+85°C		
DC42c	7	25	mA	+125°C	3.3V	30 MIPS
DC43d	9	25	mA	-40°C		
DC43a	9	25	mA	+25°C		
DC43b	9	25	mA	+85°C		
DC43c	9	25	mA	+125°C	3.3V	40 MIPS
DC44d	10	25	mA	-40°C		
DC44a	10	25	mA	+25°C		
DC44b	16	25	mA	+85°C		
DC44c	10	25	mA	+125°C	3.3V	35 MIPS

注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。

注 2: 基本 IDLE 电流的测量是在内核不工作、时钟工作而所有外设模块关闭的条件下进行的。外设模块禁止 SFR 寄存器为全零。所有 I/O 引脚配置为输入且被拉到 Vss。

表 30-7: 直流特性: 掉电电流 (IPD)

直流特性		标准工作条件: 3.0V 至 3.6V (除非另外声明) 工作温度		
参数编号	典型值 <sup>(1)</sup>	最大值	单位	条件
掉电电流 (IPD) <sup>(2)</sup>				
DC60d	55	500	μA	-40°C
DC60a	63	500	μA	+25°C
DC60b	85	500	μA	+85°C
DC60c	146	100	μA	+125°C
DC61d	8	13	μA	-40°C
DC61a	10	15	μA	+25°C
DC61b	12	20	μA	+85°C
DC61c	13	25	μA	+125°C

- 注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。  
 注 2: 基本 IPD 是在所有外设和时钟都关闭的条件下进行测量的。所有 I/O 引脚配置为输入且被拉到 Vss。WDT 等外设也都被关闭。  
 注 3: Δ 电流为模块使能时额外消耗的电流。此电流应被加到基本 IPD 电流。  
 注 4: 这些电流是针对该系列中存储容量最大的器件测得的。

表 30-8: 直流特性: 打盹电流 (IDOZE)

直流特性		标准工作条件: 3.0V 至 3.6V (除非另外声明) 工作温度			
参数编号	典型值 <sup>(1)</sup>	最大值	打盹模式时 钟分频比	单位	条件
DC73a	25	32	1:2	mA	-40°C 3.3V 40 MIPS
DC73f	23	27	1:64		
DC73g	23	26	1:128		
DC70a	42	47	1:2	mA	+25°C 3.3V 40 MIPS
DC70f	26	27	1:64		
DC70g	25	27	1:128		
DC71a	41	48	1:2	mA	+85°C 3.3V 40 MIPS
DC71f	25	28	1:64		
DC71g	24	28	1:128		
DC72a	42	49	1:2	mA	+125°C 3.3V 35 MIPS
DC72f	26	29	1:64		
DC72g	25	28	1:128		

- 注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。

表 30-9: 直流特性: I/O 引脚输入规范

直流特性		标准工作条件: 3.0V 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
D110	VIL	输入低电压					
		I/O 引脚	VSS	—	0.2 VDD	V	
D115		MCLR	VSS	—	0.2 VDD	V	
D116		OSC1 (XT 模式)	VSS	—	0.2 VDD	V	
D117		OSC1 (HS 模式)	VSS	—	0.2 VDD	V	
D118		SDAx 和 SCLx	VSS	—	.3 VDD	V	禁止 SMBus
D119		SDAx 和 SCLx	VSS	—	0.2 VDD	V	使能 SMBus
D120	VIH	输入高电压					
		I/O 引脚:					
		带模拟功能	0.8 VDD	—	VDD	V	
		仅数字功能	0.8 VDD	—	5.5	V	
D125		MCLR	0.8 VDD	—	VDD	V	
D126		OSC1 (XT 模式)	0.7 VDD	—	VDD	V	
D127		OSC1 (HS 模式)	0.7 VDD	—	VDD	V	
D128	SDAx 和 SCLx	0.7 VDD	—	VDD	V	禁止 SMBus	
D129	SDAx 和 SCLx	0.8 VDD	—	VDD	V	使能 SMBus	
D130	ICNPU	CNx 上拉电流	50	250	400	μA	VDD = 3.3V, VPIN = VSS
D150	IIL	输入泄漏电流 (2)(3)					
		I/O 端口	—	—	±2	μA	VSS ≤ VPIN ≤ VDD, 引脚处于高阻态
D151		模拟输入引脚	—	—	±1	μA	VSS ≤ VPIN ≤ VDD, 引脚处于高阻态, 40°C ≤ TA ≤ +85°C
D151a		模拟输入引脚	—	—	±2	μA	模拟引脚与外部参考电压引脚共用, 40°C ≤ TA ≤ +85°C
D151b		模拟输入引脚	—	—	±3.5	μA	VSS ≤ VPIN ≤ VDD, 引脚处于高阻态 -40°C ≤ TA ≤ +125°C
D151c		模拟输入引脚	—	—	±8	μA	模拟引脚与外部参考电压引脚共用, -40°C ≤ TA ≤ +125°C
D155		MCLR	—	—	±2	μA	VSS ≤ VPIN ≤ VDD
D156	OSC1	—	—	±2	μA	VSS ≤ VPIN ≤ VDD, XT 和 HS 模式	

- 注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。
- 2: MCLR 引脚上的泄漏电流主要取决于所施加电压。规定电压为正常工作条件下的电压。在不同的输入电压下可能测得更高的泄漏电流。
- 3: 负电流定义为引脚的拉电流。

表 30-10: 直流特性: I/O 引脚输出规范

直流特性			标准工作条件: 3.0V 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
DO10 DO16	VOL	输出低电压 I/O 端口 OSC2/CLKO	— —	— —	0.4 0.4	V V	IOL = 2 mA, VDD = 3.3V IOL = 2 mA, VDD = 3.3V
DO20 DO26	VOH	输出高电压 I/O 端口 OSC2/CLKO	2.40 2.41	— —	— —	V V	IOH = -2.3 mA, VDD = 3.3V IOH = -1.3 mA, VDD = 3.3V

表 30-11: 电气特性: BOR

直流特性			标准工作条件: 3.0V 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 <sup>(1)</sup>	典型值	最大值	单位	条件
BO10	VBOR	当 VDD 从高压变为低电压时的 BOR 事件 BOR 事件与 VDD 内核电压下降关联	2.40	—	2.55	V	

注 1: 这些参数仅供设计参考, 生产时未经测试。

表 30-12: 直流特性: 程序存储器

直流特性			标准工作条件: 3.0V 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
D130	EP	闪存程序存储器 单元耐擦写能力	10,000	—	—	E/W	-40°C 至 +125°C
D131	VPR	读操作时的 VDD	V <sub>MIN</sub>	—	3.6	V	V <sub>MIN</sub> = 最小工作电压
D132B	VPEW	自定时写的 VDD	V <sub>MIN</sub>	—	3.6	V	V <sub>MIN</sub> = 最小工作电压
D134	TRETD	特性保持时间	20	—	—	年	假设没有违反其他规范, -40°C 至 +125°C
D135	IDDP	编程时的供电电流	—	10	—	mA	
D136	TRW	行写入时间	—	1.6	—	ms	
D137	TPE	页擦除时间	—	20	—	ms	
D138	TWW	字写周期时间	20	—	40	μs	

注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。

表 30-13: 内部稳压器规范

工作条件: -40°C < TA < +85°C (除非另外声明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
	CEFC	外部滤波电容值	1	10	—	μF	电容必须具有较低值的等效串联阻抗 (< 5 欧姆)

### 30.2 交流特性和时序参数

本节给出了 dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04 和 dsPIC33FJ128MCX02/X04 系列器件的交流特性和时序参数。

表 30-14: 温度和电压规范——交流

交流特性	标准工作条件: 3.0V 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 30.0 节“电气特性”中所介绍。	

图 30-1: 器件时序规范的负载条件

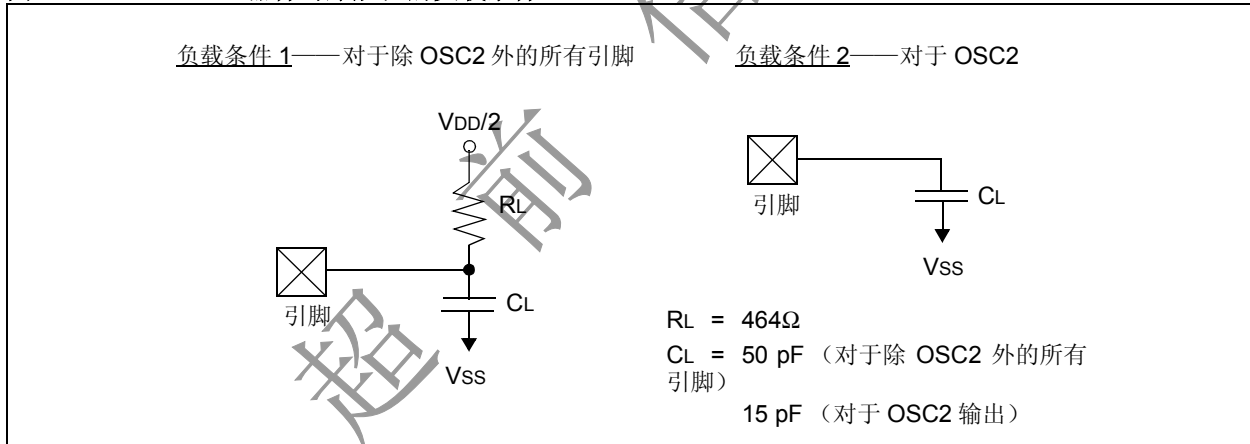


表 30-15: 输出引脚上的容性负载要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
DO50	COsc2	OSC2/SOSC2 引脚	—	—	15	pF	当外部时钟用于驱动 OSC1 时处于 XT 和 HS 模式下
DO56	CIo	所有 I/O 引脚和 OSC2	—	—	50	pF	EC 模式
DO58	CB	SCLx 和 SDAx	—	—	400	pF	在 I <sup>2</sup> C™ 模式下

图 30-2: 外部时钟时序

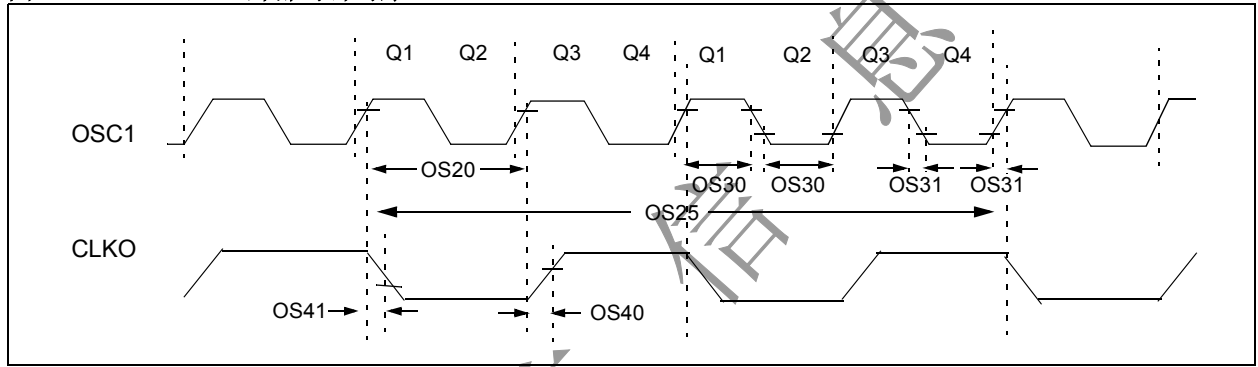


表 30-16: 外部时钟时序要求

交流特性			标准工作条件: 3.0V 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
OS10	FIN	外部 CLKI 频率 (外部时钟仅允许运行于 EC 和 ECPLL 模式)	DC	—	40	MHz	EC
		晶振频率	3.5 10	— —	10 40 33	MHz MHz kHz	XT HS SOSC
OS20	Tosc	Tosc = 1/Fosc	12.5	—	DC	ns	
OS25	Tcy	指令周期 <sup>(2)</sup>	25	—	DC	ns	
OS30	TosL, TosH	外部时钟输入 (OSC1) 高电平或低电平时间	0.375 x Tsc	—	0.625 x Tsc	ns	EC
OS31	TosR, TosF	外部时钟输入 (OSC1) 上升或下降时间	—	—	20	ns	EC
OS40	TckR	CLKO 上升时间 <sup>(3)</sup>	—	5.2	—	ns	
OS41	TckF	CLKO 下降时间 <sup>(3)</sup>	—	5.2	—	ns	

- 注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。
- 注 2: 指令周期 (Tcy) 等于输入振荡器时基周期的两倍。所有规定值均为基于特定振荡器类型, 器件在标准工作条件下执行代码时的特性数据。超出这些规定的限定值, 可能导致振荡器运行不稳定和 / 或导致电流消耗超出预期值。所有器件在测试“最小”值时, 都在 OSC1/CLKI 引脚连接了外部时钟。当使用了外部时钟输入时, 所有器件的“最大”周期时间限制为“DC”(无时钟)。
- 注 3: 测量在 EC 模式下进行。在 OSC2 引脚上测量 CLKO 信号。

表 30-17: PLL 时钟时序规范 (VDD = 3.0V 至 3.6V)

交流特性		标准工作条件: 3.0V 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	PLL 压控振荡器 (VCO) 的输入频率范围	0.8	—	8	MHz	ECPLL 和 XTPLL 模式
OS51	Fsys	片上 VCO 系统频率	100	—	200	MHz	
OS52	TLOCK	PLL 起振时间 (锁定时间)	0.9	1.5	3.1	mS	
OS53	DCLK	CLKO 稳定性 (抗抖动性)	-3	0.5	3	%	在 100 ms 时间段内测量

注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。这些参数仅供设计参考, 未经测试。

表 30-18: 交流特性: 内部 RC 精度

交流特性		标准工作条件: 3.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)					
参数编号	特性	最小值	典型值	最大值	单位	条件	
<b>FRC 频率 = 7.37 MHz<sup>(1,2)</sup> 时的内部 FRC 精度</b>							
F20	FRC	-2	—	+2	%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
	FRC	-5	—	+5	%	-40°C ≤ TA ≤ +125°C	VDD = 3.0-3.6V

注 1: 频率在 25°C 和 3.3V 条件下校准。TUN 位可用于补偿温度漂移。

注 2: FRC 被设置为 25°C 下的初始频率 7.37 MHz (±2%)。

表 30-19: 内部 RC 精度

交流特性		标准工作条件: 3.0V 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	
<b>LPRC 频率 = 32.768 kHz<sup>(1)</sup></b>							
F21	LPRC	-20	±6	+20	%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
	LPRC	-70	—	+70	%	-40°C ≤ TA ≤ +125°C	VDD = 3.0-3.6V

注 1: LPRC 频率将随 VDD 的变化而变化。



图 30-3: I/O 时序特性

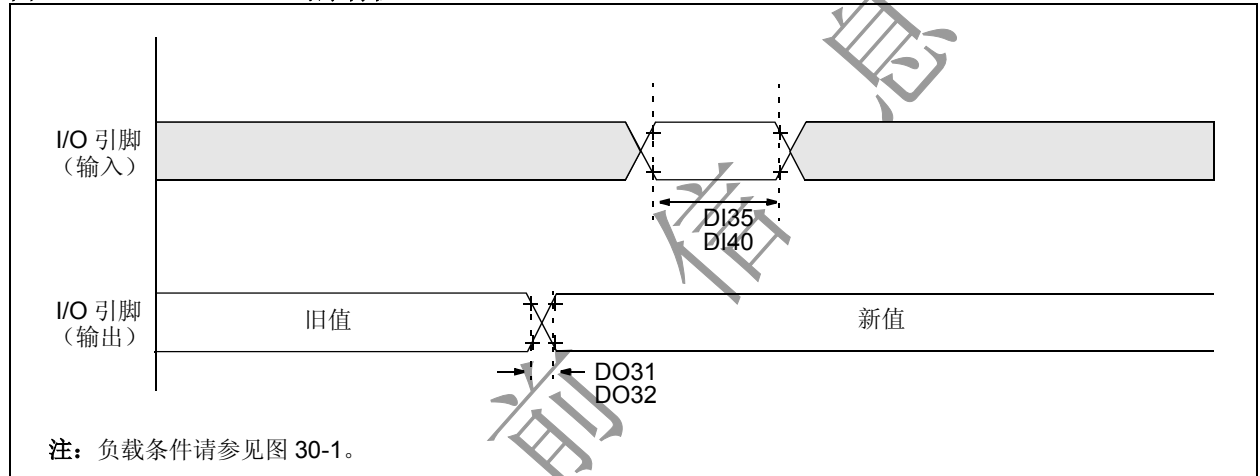


表 30-20: I/O 时序要求

交流特性		标准工作条件: 3.0V 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 (输出)	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”栏中的数据均为 3.3V、25°C 条件下的值。

图 30-4: 复位、看门狗定时器、振荡器起振定时器和上电延时定时器时序特性

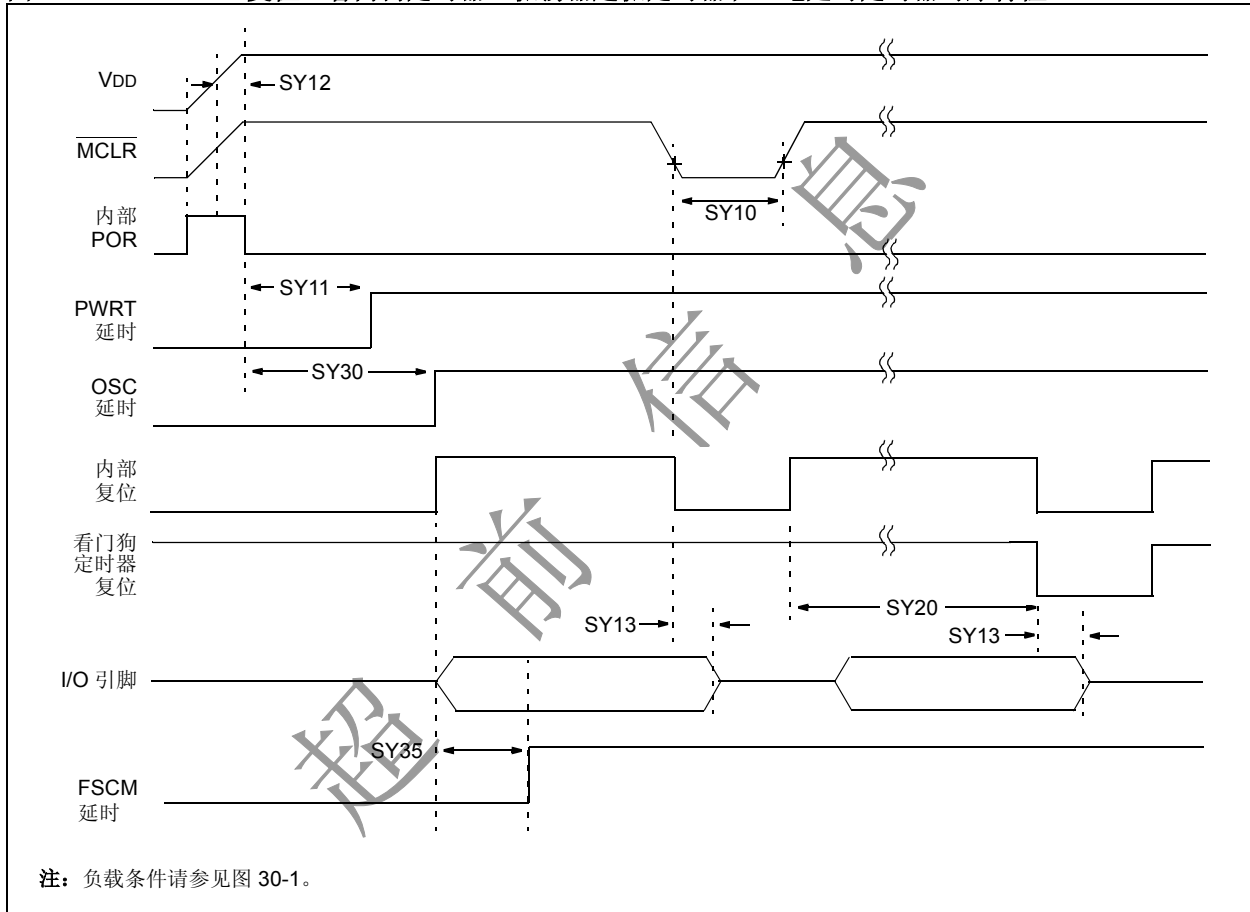


表 30-21: 复位、看门狗定时器、振荡器起振定时器和上电延时定时器时序要求

交流特性		标准工作条件: 3.0V 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
SY10	TMCL	MCLR 脉冲宽度 (低电平)	2	—	—	μs	-40°C 至 +85°C
SY11	TPWRT	上电延时定时器周期	—	2 4 8 16 32 64 128	—	ms	-40°C 至 +85°C 用户可编程
SY12	TPOR	上电复位延时	3	10	30	μs	-40°C 至 +85°C
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	0.68	0.72	1.2	μs	
SY20	TWDT1	看门狗定时器超时周期 (无预分频器)	1.7	2.1	2.6	ms	VDD = 3V, -40°C 至 +85°C
SY30	TOST	振荡器起振时间	—	1024 TOSC	—	—	TOSC = OSC1 周期
SY35	TFSCM	故障保护时钟监视器延时	—	500	900	μs	-40°C 至 +85°C

注 1: 这些参数为特性值, 但生产时未经测试。

注 2: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。

图 30-5: TIMER1、TIMER2 和 TIMER3 外部时钟时序特性

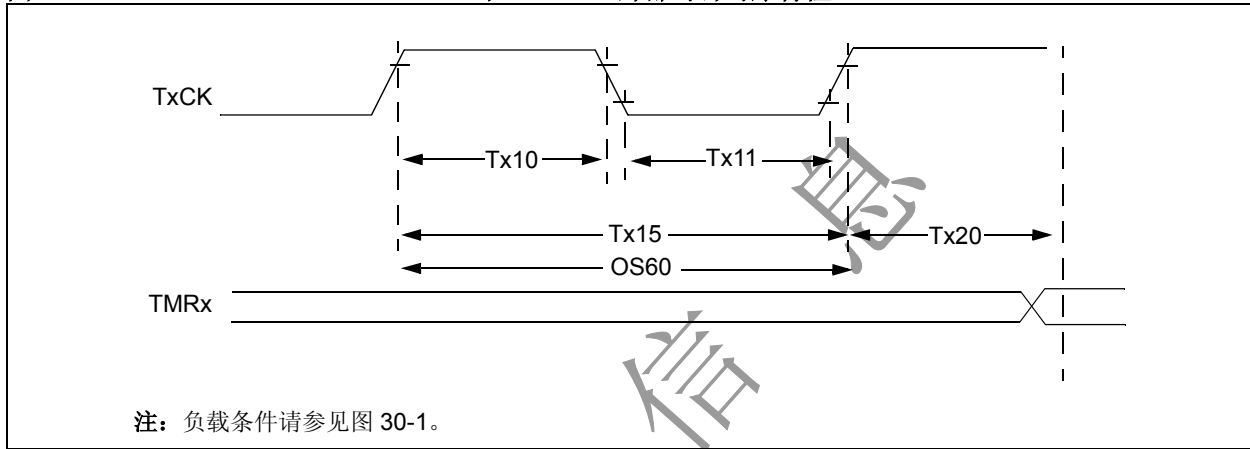


表 30-22: TIMER1 外部时钟时序要求 (1)

交流特性		标准工作条件: 3.0V 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	T <sub>TXH</sub>	TxCK 高电平时间	同步, 无预分频器	0.5 T <sub>CY</sub> + 20	—	—	ns	也必须满足参数 TA15
		同步, 带预分频器	10	—	—	ns		
		异步	10	—	—	ns		
TA11	T <sub>TXL</sub>	TxCK 低电平时间	同步, 无预分频器	0.5 T <sub>CY</sub> + 20	—	—	ns	也必须满足参数 TA15
		同步, 带预分频器	10	—	—	ns		
		异步	10	—	—	ns		
TA15	T <sub>TXP</sub>	TxCK 输入周期	同步, 无预分频器	T <sub>CY</sub> + 40	—	—	ns	
		同步, 带预分频器	20 ns 和 (T <sub>CY</sub> + 40)/N 中的较大值	—	—	—	N = 预分频值 (1、8、64 或 256)	
		异步	20	—	—	ns		
OS60	F <sub>t1</sub>	SOSC1/T1CK 振荡器输入频率范围 (通过置 1 TCS (T1CON<1>) 位 使能振荡器)	DC	—	50	kHz		
TA20	T <sub>CKEXTMRL</sub>	从外部 TxCK 时钟边沿到定时器递增之间的延时	0.5 T <sub>CY</sub>		1.5 T <sub>CY</sub>	—		

注 1: Timer1 属于 A 类定时器。

表 30-23: TIMER2 外部时钟时序要求

交流特性				标准工作条件: 3.0V 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
TB10	TtxH	TxCK 高电平时间	同步, 无预分频器	0.5 T <sub>CY</sub> + 20	—	—	ns	也必须满足参数 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 T <sub>CY</sub> + 20	—	—	ns	也必须满足参数 TB15
			同步, 带预分频器	10	—	—	ns	
TB15	TtxP	TxCK 输入周期	同步, 无预分频器	T <sub>CY</sub> + 40	—	—	ns	N = 预分频值 (1、8、64 或 256)
			同步, 带预分频器	20 ns 和 (T <sub>CY</sub> + 40)/N 中的较大值	—	—		
TB20	TCKEXTMRL	从外部 TxCK 时钟边沿到定时器递增之间的延时		0.5 T <sub>CY</sub>	—	1.5 T <sub>CY</sub>	—	

表 30-24: TIMER3 外部时钟时序要求

交流特性				标准工作条件: 3.0V 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
TC10	TtxH	TxCK 高电平时间	同步	0.5 T <sub>CY</sub> + 20	—	—	ns	也必须满足参数 TC15
TC11	TtxL	TxCK 低电平时间	同步	0.5 T <sub>CY</sub> + 20	—	—	ns	也必须满足参数 TC15
TC15	TtxP	TxCK 输入周期	同步, 无预分频器	T <sub>CY</sub> + 40	—	—	ns	N = 预分频值 (1、8、64 或 256)
			同步, 带预分频器	20 ns 和 (T <sub>CY</sub> + 40)/N 中的较大值	—	—		
TC20	TCKEXTMRL	从外部 TxCK 时钟边沿到定时器递增之间的延时		0.5 T <sub>CY</sub>	—	1.5 T <sub>CY</sub>	—	

图 30-6: TIMERQ (QEI 模块) 外部时钟时序特性

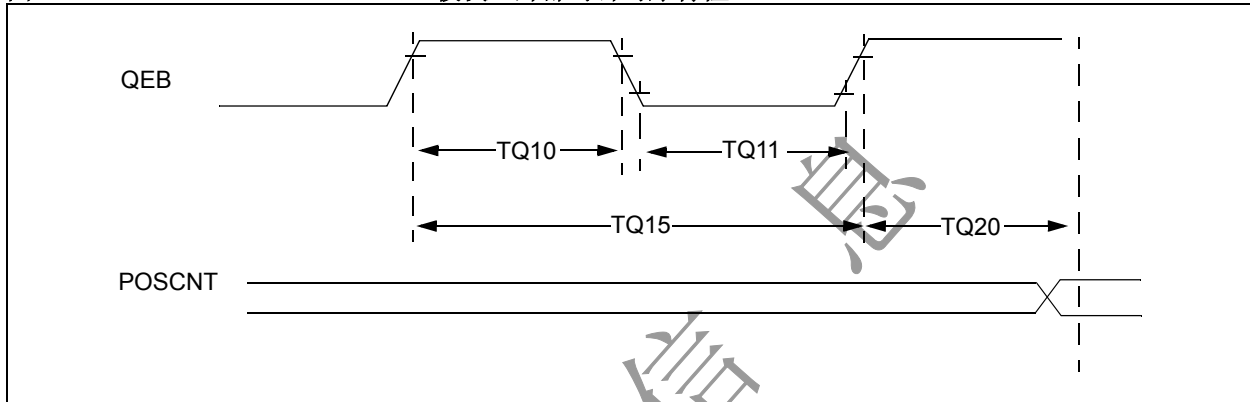


表 30-25: QEI 模块外部时钟时序要求

交流特性		标准工作条件: 3.0V 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	最大值	单位	条件
TQ10	TtQH	TQCK 高电平时间	同步, 带预分频器	T <sub>cy</sub> + 20		—	ns	还必须满足参数 TQ15
TQ11	TtQL	TQCK 低电平时间	同步, 带预分频器	T <sub>cy</sub> + 20		—	ns	还必须满足参数 TQ15
TQ15	TtQP	TQCP 输入周期	同步, 带预分频器	2 * T <sub>cy</sub> + 40		—	ns	—
TQ20	TCKEXTMRL	从外部 TxCK 时钟边沿到定时器递增之间的延时		0.5 T <sub>cy</sub>		1.5 T <sub>cy</sub>	—	—

注 1: 这些参数为特性值, 但生产时未经测试。

图 30-7: 输入捕捉 (CAPx) 时序特性

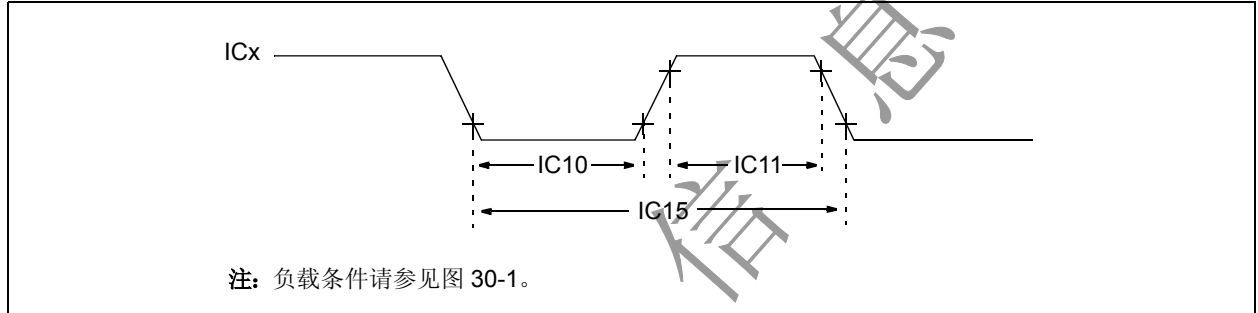


表 30-26: 输入捕捉时序要求

交流特性		标准工作条件: 3.0V 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	最大值	单位	条件
IC10	TcCL	ICx 输入低电平时间	无预分频器	0.5 Tcy + 20	—	ns	
			带预分频器	10	—	ns	
IC11	TcCH	ICx 输入高电平时间	无预分频器	0.5 Tcy + 20	—	ns	
			带预分频器	10	—	ns	
IC15	TcCP	ICx 输入周期		(Tcy + 40)/N	—	ns	N = 预分频值 (1、4 或 16)

注 1: 这些参数为特性值, 但生产时未经测试。

图 30-8: 输出比较模块 (OCx) 时序特性

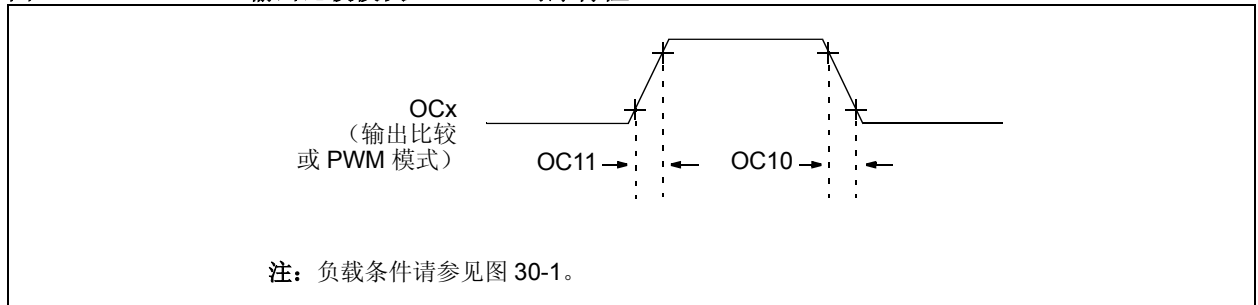


表 30-27: 输出比较模块时序要求

交流特性		标准工作条件: 3.0V 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	最大值	单位	条件
OC10	TcCF	OCx 输出下降时间	—	—	—	ns	见参数 D032
OC11	TcCR	OCx 输出上升时间	—	—	—	ns	见参数 D031

注 1: 这些参数为特性值, 但生产时未经测试。

图 30-9: 输出比较 /PWM 模块时序特性

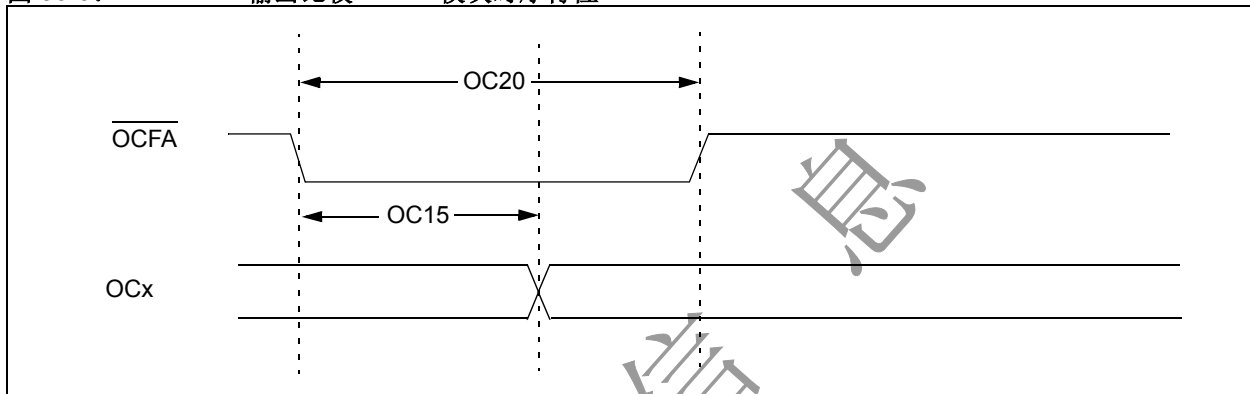


表 30-28: 简单输出比较 /PWM 模式时序要求

交流特性			标准工作条件: 3.0V 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	最大值	单位	条件
OC15	TFD	故障输入到 PWM I/O 发生变化的时间	—	—	50	ns	—
OC20	TFLT	故障输入脉冲宽度	50	—	—	ns	—

注 1: 这些参数为特性值, 但生产时未经测试。



图 30-10: 电机控制 PWM 模块故障时序特性

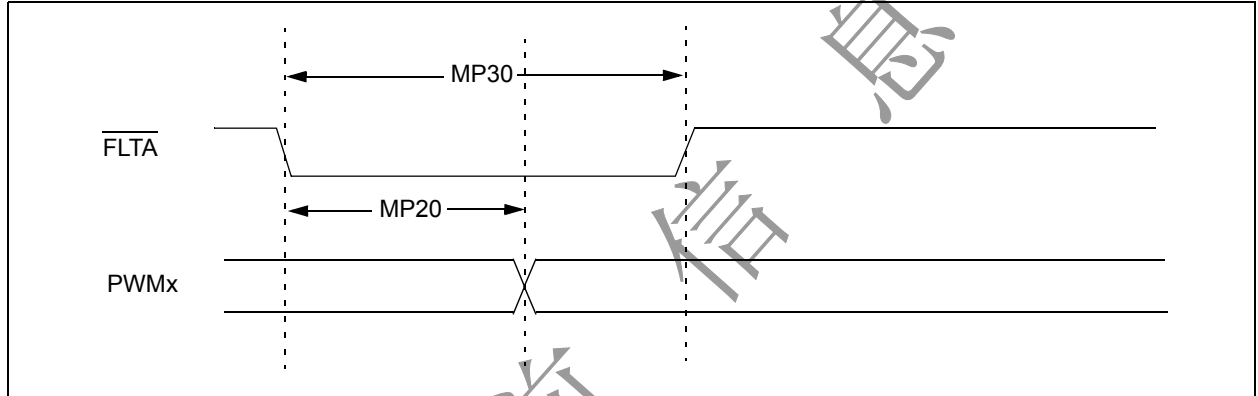


图 30-11: 电机控制 PWM 模块时序特性

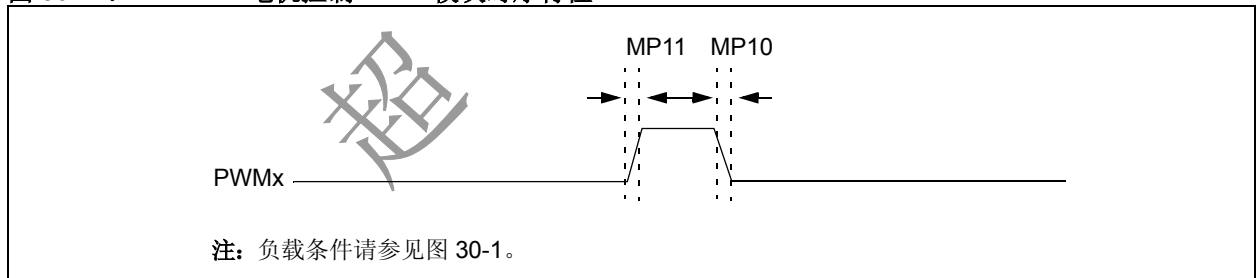


表 30-29: 电机控制 PWM 模块时序要求

交流特性			标准工作条件: 3.0V 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	最大值	单位	条件
MP10	TFPWM	PWM 输出下降时间	—	—	—	ns	见参数 D032
MP11	TRPWM	PWM 输出上升时间	—	—	—	ns	见参数 D031
MP20	TFD	故障输入 ↓ 到 PWM I/O 变化的时间	—	—	50	ns	—
MP30	TFH	最小脉冲宽度	50	—	—	ns	—

注 1: 这些参数为特性值, 但生产时未经测试。

图 30-12: QEA/QEB 输入特性

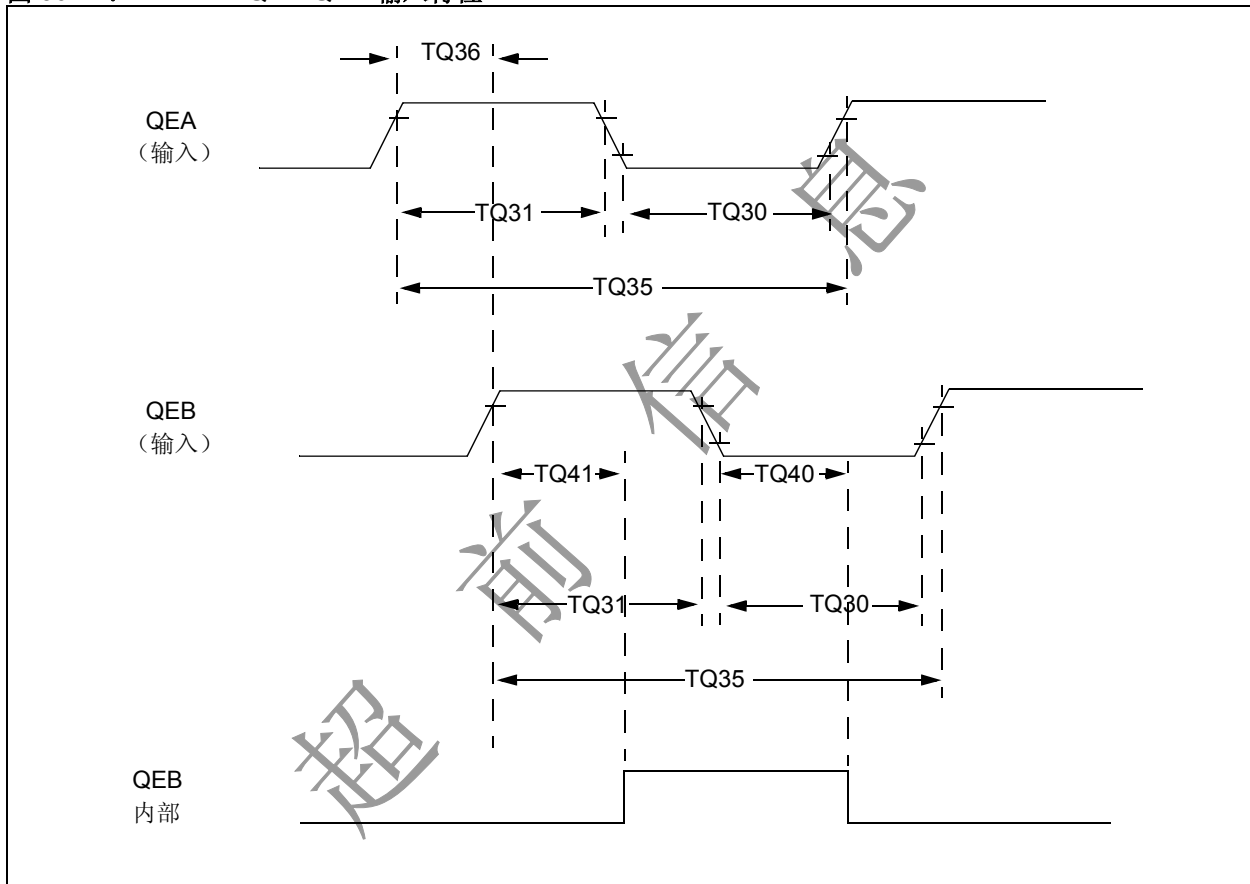


表 30-30: 正交解码器时序要求

交流特性		标准工作条件: 3.0V 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)	典型值 (2)	最大值	单位	条件
TQ30	TQuL	正交输入低电平时间	6 Tcy	—	ns	—
TQ31	TQuH	正交输入高电平时间	6 Tcy	—	ns	—
TQ35	TQuIN	正交输入周期	12 Tcy	—	ns	—
TQ36	TQuP	正交相周期	3 Tcy	—	ns	—
TQ40	TQuFL	数字滤波器确认低电平的时间	3 * N * Tcy	—	ns	N = 1、2、4、16、32、64、128 和 256 (注 3)
TQ41	TQuFH	数字滤波器确认高电平的时间	3 * N * Tcy	—	ns	N = 1、2、4、16、32、64、128 和 256 (注 3)

- 注 1: 这些参数为特性值, 但生产时未经测试。  
 注 2: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。这些参数仅供设计参考, 未经测试。  
 注 3: N = 索引通道数字滤波器时钟分频选择位。请参见“dsPIC33F Family Reference Manual”中的“Section 15. Quadrature Encoder Interface (QEI)”。请参见 Microchip 网站了解最新的“dsPIC33F Family Reference Manual”章节。

图 30-13: QEI 模块索引脉冲时序特性

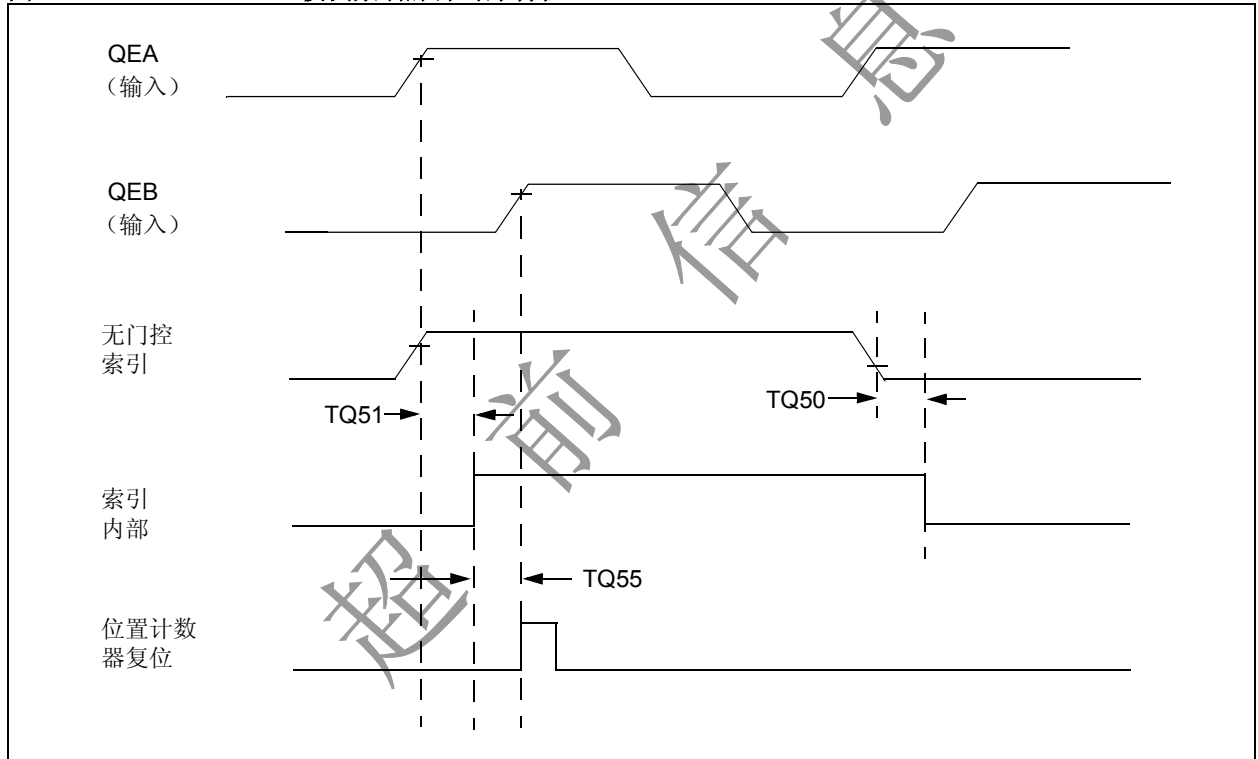


表 30-31: QEI 索引脉冲时序要求

交流特性		标准工作条件: 3.0V 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	最大值	单位	条件
TQ50	TqiL	数字滤波器确认低电平的时间	3 * N * TcY	—	ns	N = 1、2、4、16、32、64、128 和 256 (注 2)
TQ51	TqiH	数字滤波器确认高电平的时间	3 * N * TcY	—	ns	N = 1、2、4、16、32、64、128 和 256 (注 2)
TQ55	Tqidxr	索引脉冲确认至位置计数器复位的时间 (无门控索引)	3 TcY	—	ns	—

注 1: 这些参数为特性值, 但生产时未经测试。

注 2: 给出的索引脉冲与 QEA 和 QEB 对齐仅适用于位置计数器复位时序和正向旋转方式 (QEA 超前 QEB)。同样的时序适用于反向旋转的情形 (QEA 滞后于 QEB), 但索引脉冲识别出现在下降沿。

图 30-14: SPIx 模块主模式 (CKE = 0) 时序特性

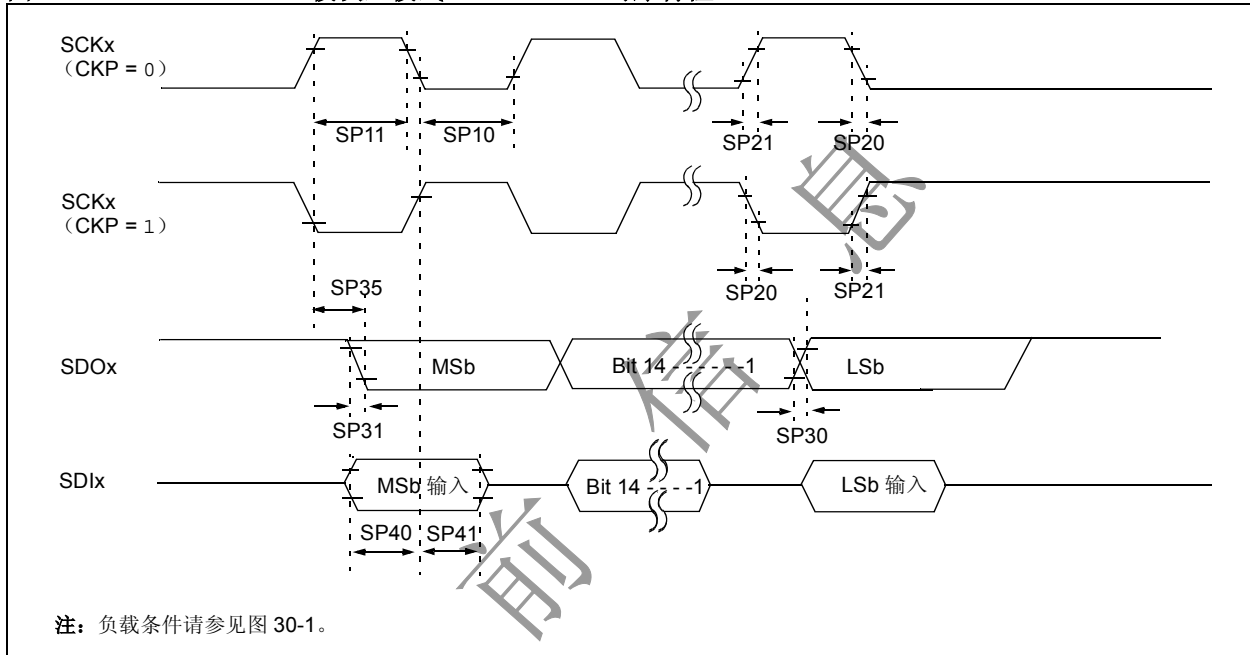


表 30-32: SPIx 主模式 (CKE = 0) 时序要求

交流特性			标准工作条件: 3.0V 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
SP10	TscL	SCKx 输出低电平时间	Tcy/2	—	—	ns	见注 3
SP11	TscH	SCKx 输出高电平时间	Tcy/2	—	—	ns	见注 3
SP20	TscF	SCKx 输出下降时间	—	—	—	ns	见参数 D032 和注 4
SP21	TscR	SCKx 输出上升时间	—	—	—	ns	见参数 D031 和注 4
SP30	TdoF	SDOx 数据输出下降时间	—	—	—	ns	见参数 D032 和注 4
SP31	TdoR	SDOx 数据输出上升时间	—	—	—	ns	见参数 D031 和注 4
SP35	Tsch2doV, TscL2doV	在 SCKx 边沿之后 SDOx 数据输出有效的的时间	—	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: 这些参数为特性值, 但生产时未经测试。  
 注 2: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。  
 注 3: SCKx 的最小时钟周期为 100 ns。因此, 主模式下产生的时钟不应违反此规范。  
 注 4: 假定所有 SPIx 引脚上的负载均为 50 pF。

图 30-15: SPIx 模块主模式 (CKE = 1) 时序特性

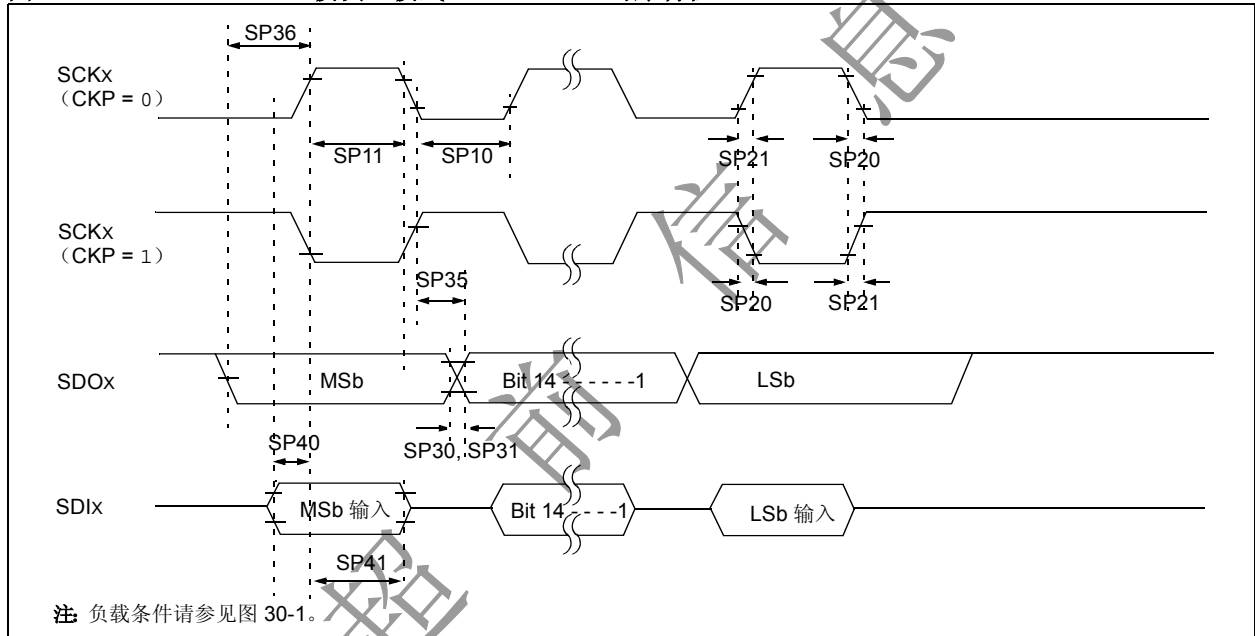


表 30-33: SPIx 模块主模式 (CKE = 1) 时序要求

交流特性		标准工作条件: 3.0V 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	见注 3
SP11	TscH	SCKx 输出高电平时间	Tcy/2	—	—	ns	见注 3
SP20	TscF	SCKx 输出下降时间	—	—	—	ns	见参数 D032 和注 4
SP21	TscR	SCKx 输出上升时间	—	—	—	ns	见参数 D031 和注 4
SP30	TdoF	SDOx 数据输出下降时间	—	—	—	ns	见参数 D032 和注 4
SP31	TdoR	SDOx 数据输出上升时间	—	—	—	ns	见参数 D031 和注 4
SP35	Tsch2doV, TscL2doV	在 SCKx 边沿之后 SDOx 数据输出有效的时间	—	6	20	ns	—
SP36	TdoV2sc, TdoV2scL	SDOx 数据输出建立到出现第一个 SCKx 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: 这些参数为特性值, 但生产时未经测试。  
 注 2: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。  
 注 3: SCKx 的最小时钟周期为 100 ns。主模式下产生的时钟不应违反此规范。  
 注 4: 假定所有 SPIx 引脚上的负载均为 50 pF。

图 30-16: SPIx 模块从模式 (CKE = 0) 时序特性

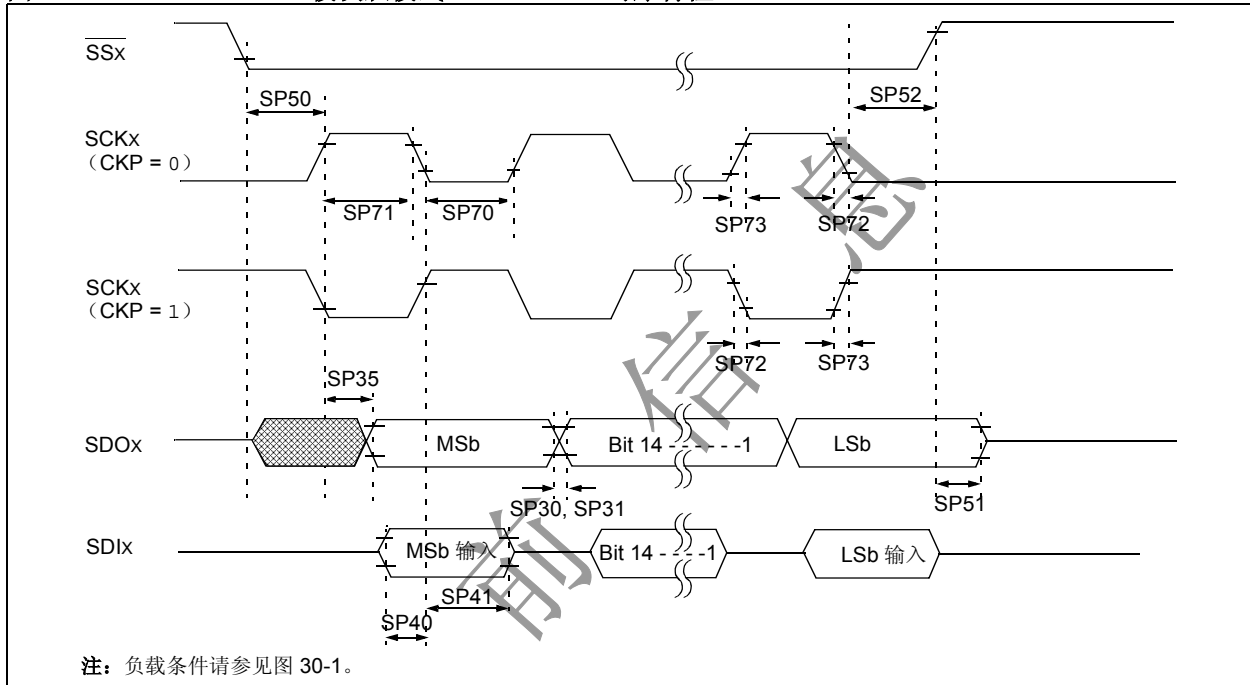


表 30-34: SPIx 模块从模式 (CKE = 0) 时序要求

交流特性			标准工作条件: 3.0V 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	—	10	25	ns	见注 3
SP73	TscR	SCKx 输入上升时间	—	10	25	ns	见注 3
SP30	TdoF	SDOx 数据输出下降时间	—	—	—	ns	见参数 D032 和注 3
SP31	TdoR	SDOx 数据输出上升时间	—	—	—	ns	见参数 D031 和注 3
SP35	Tsch2doV, TscL2doV	在 SCKx 边沿之后 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 输出高阻态的时间	10	—	50	ns	见注 3
SP52	Tsch2ssH TscL2ssH	SCKx 边沿后 SSx 有效的的时间	1.5 Tcy +40	—	—	ns	—

- 注 1: 这些参数为特性值, 但生产时未经测试。  
 注 2: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。  
 注 3: 假定所有 SPIx 引脚上的负载均为 50 pF。

图 30-17: SPIx 模块从模式 (CKE = 1) 时序特性

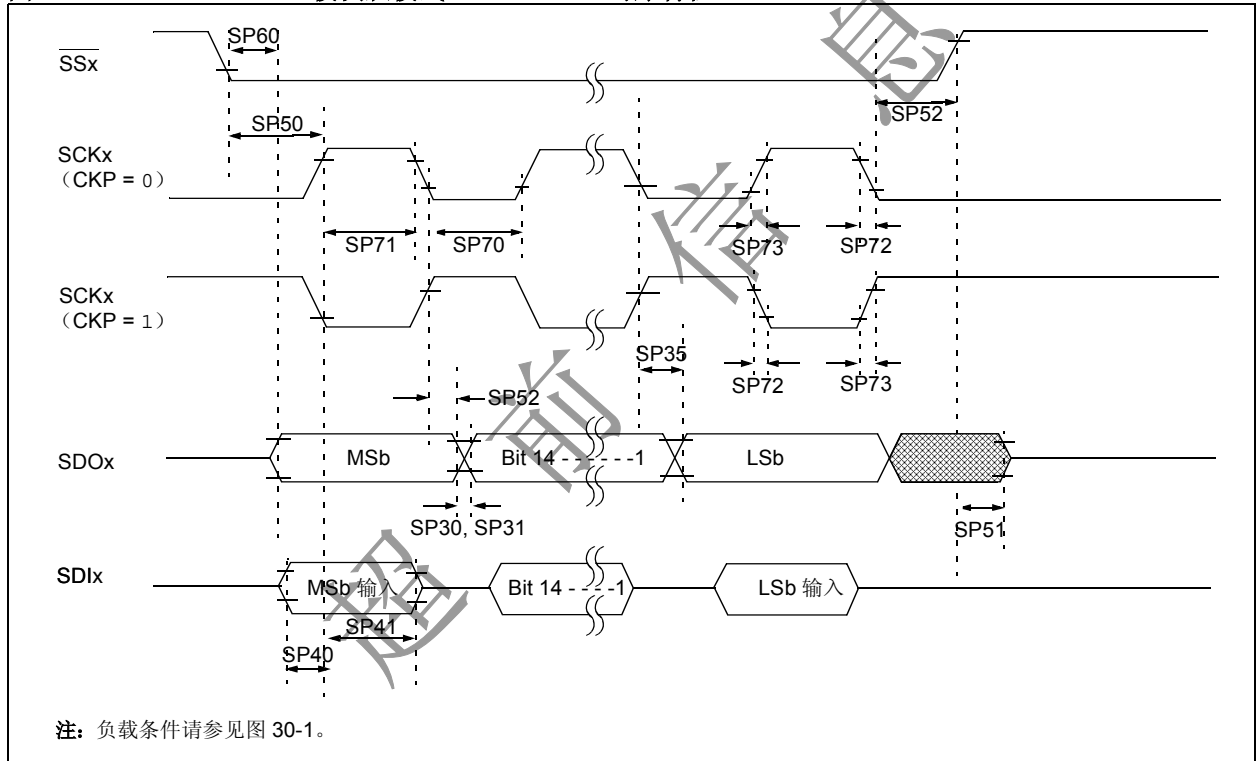


表 30-35: SPIx 模块从模式 (CKE = 1) 时序要求

交流特性			标准工作条件: 3.0V 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	—	10	25	ns	见注 3
SP73	TscR	SCKx 输入上升时间	—	10	25	ns	见注 3
SP30	TdoF	SDOx 数据输出下降时间	—	—	—	ns	见参数 D032 和注 3
SP31	TdoR	SDOx 数据输出上升时间	—	—	—	ns	见参数 D031 和注 3
SP35	TscH2doV, TscL2doV	在 SCKx 边沿之后 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 输出高阻态的时间	10	—	50	ns	见注 4
SP52	TscH2ssH TscL2ssH	SCKx 边沿后 SSx ↑ 有效的时间	1.5 TCY + 40	—	—	ns	—
SP60	TssL2doV	在 SSx 边沿之后 SDOx 数据输出有效的时间	—	—	50	ns	—

- 注 1: 这些参数为特性值, 但生产时未经测试。  
 注 2: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。  
 注 3: SCKx 的最小时钟周期为 100 ns。主模式下产生的时钟不应违反此规范。  
 注 4: 假定所有 SPIx 引脚上的负载均为 50 pF。



图 30-18: I2Cx 总线启动位 / 停止位时序特性 (主模式)

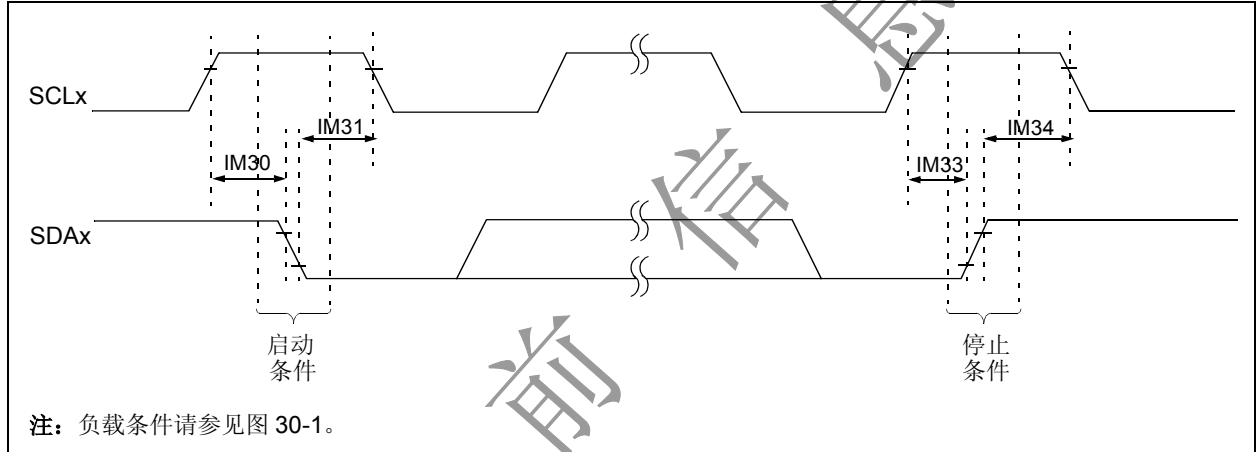


图 30-19: I2Cx 总线数据时序特性 (主模式)

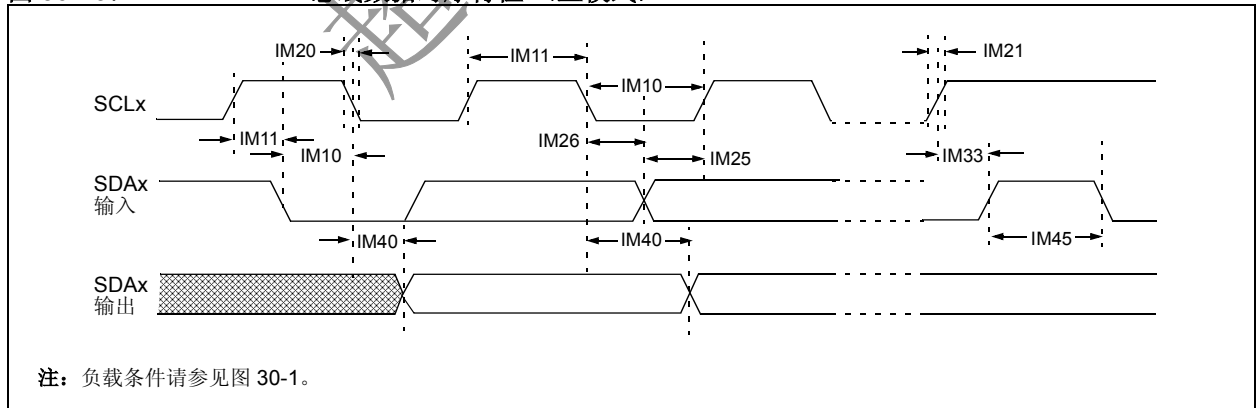


表 30-36: I2Cx 总线数据时序要求 (主模式)

交流特性				标准工作条件: 3.0V 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 <sup>(1)</sup>	最大值	单位	条件	
IM10	TLO:SCL	时钟低电平时间	100 kHz 模式	Tcy/2 (BRG + 1)	—	μs	—
			400 kHz 模式	Tcy/2 (BRG + 1)	—	μs	—
			1 MHz 模式 <sup>(2)</sup>	Tcy/2 (BRG + 1)	—	μs	—
IM11	THI:SCL	时钟高电平时间	100 kHz 模式	Tcy/2 (BRG + 1)	—	μs	—
			400 kHz 模式	Tcy/2 (BRG + 1)	—	μs	—
			1 MHz 模式 <sup>(2)</sup>	Tcy/2 (BRG + 1)	—	μs	—
IM20	TF:SCL	SDAx 和 SCLx 下降时间	100 kHz 模式	—	300	ns	Cb 值规定在 10 至 400 pF 之间
			400 kHz 模式	20 + 0.1 Cb	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	Cb 值规定在 10 至 400 pF 之间
			400 kHz 模式	20 + 0.1 Cb	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	—	μs	—
			400 kHz 模式	0	0.9	μs	
			1 MHz 模式 <sup>(2)</sup>	0.2	—	μs	
IM30	TSU:STA	启动条件建立时间	100 kHz 模式	Tcy/2 (BRG + 1)	—	μs	仅与重复启动条件相关
			400 kHz 模式	Tcy/2 (BRG + 1)	—	μs	
			1 MHz 模式 <sup>(2)</sup>	Tcy/2 (BRG + 1)	—	μs	
IM31	THD:STA	启动条件保持时间	100 kHz 模式	Tcy/2 (BRG + 1)	—	μs	这个周期后产生第一个时钟脉冲
			400 kHz 模式	Tcy/2 (BRG + 1)	—	μs	
			1 MHz 模式 <sup>(2)</sup>	Tcy/2 (BRG + 1)	—	μs	
IM33	TSU:STO	停止条件建立时间	100 kHz 模式	Tcy/2 (BRG + 1)	—	μs	—
			400 kHz 模式	Tcy/2 (BRG + 1)	—	μs	
			1 MHz 模式 <sup>(2)</sup>	Tcy/2 (BRG + 1)	—	μs	
IM34	THD:STO	停止条件保持时间	100 kHz 模式	Tcy/2 (BRG + 1)	—	ns	—
			400 kHz 模式	Tcy/2 (BRG + 1)	—	ns	
			1 MHz 模式 <sup>(2)</sup>	Tcy/2 (BRG + 1)	—	ns	
IM40	TAA:SCL	自时钟边沿到输出有效的的时间	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	在启动一个新的传输前总线必须保持空闲的时间
			400 kHz 模式	1.3	—	μs	
			1 MHz 模式 <sup>(2)</sup>	0.5	—	μs	
IM50	Cb	总线容性负载	—	400	pF	—	

注 1: BRG 为 I<sup>2</sup>C 波特率发生器的值。请参见 “dsPIC33F Family Reference Manual” 的 “Section 19. Inter-Integrated Circuit (I<sup>2</sup>C™)”。请访问 Microchip 网站了解最新的 “dsPIC33F Family Reference Manual” 章节。

2: 所有 I2Cx 引脚的最大引脚电容为 10 pF (仅限 1 MHz 模式)。

图 30-20: I2Cx 总线启动位 / 停止位时序特性 (从模式)

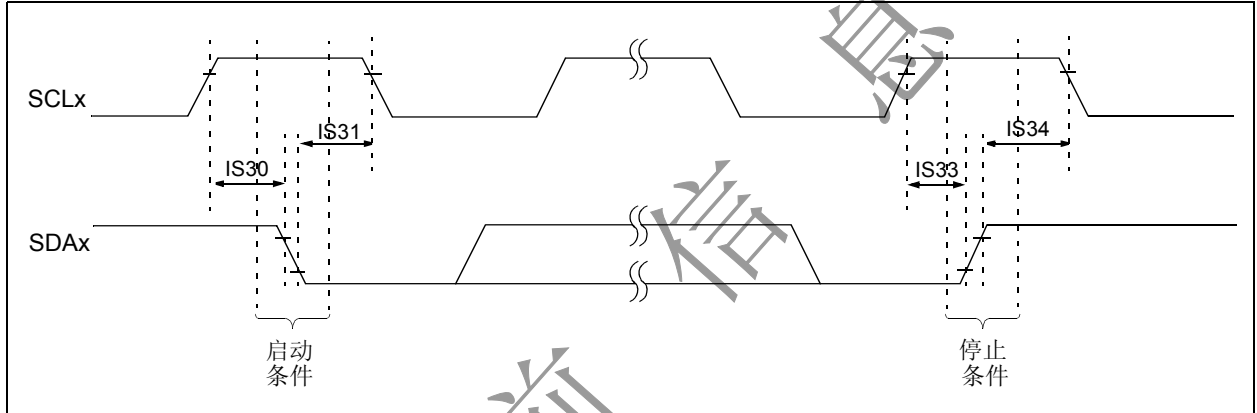


图 30-21: I2Cx 总线数据时序特性 (从模式)

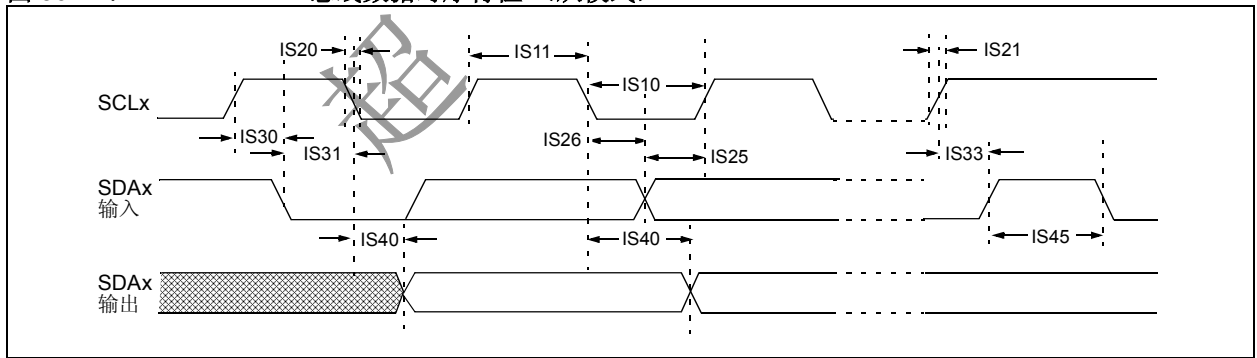


表 30-37: I2Cx 总线数据时序要求 (从模式)

交流特性				标准工作条件: 3.0V 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	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	器件工作频率不得低于 1.5 MHz
			400 kHz 模式	1.3	—	μs	器件工作频率不得低于 10 MHz
			1 MHz 模式 (1)	0.5	—	μs	—
IS11	THI:SCL	时钟高电平时间	100 kHz 模式	4.0	—	μs	器件工作频率不得低于 1.5 MHz
			400 kHz 模式	0.6	—	μs	器件工作频率不得低于 10 MHz
			1 MHz 模式 (1)	0.5	—	μs	—
IS20	TF:SCL	SDAx 和 SCLx 下降时间	100 kHz 模式	—	300	ns	Cb 值规定在 10 至 400pF 之间
			400 kHz 模式	20 + 0.1 Cb	300	ns	
			1 MHz 模式 (1)	—	100	ns	
IS21	TR:SCL	SDAx 和 SCLx 上升时间	100 kHz 模式	—	1000	ns	Cb 值规定在 10 至 400pF 之间
			400 kHz 模式	20 + 0.1 Cb	300	ns	
			1 MHz 模式 (1)	—	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 (1)	100	—	ns	
IS26	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	μs	—
			400 kHz 模式	0	0.9	μs	
			1 MHz 模式 (1)	0	0.3	μs	
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 (1)	0.25	—	μs	
IS31	THD:STA	启动条件保持时间	100 kHz 模式	4.0	—	μs	这个周期后产生第一个时钟脉冲
			400 kHz 模式	0.6	—	μs	
			1 MHz 模式 (1)	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 (1)	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 (1)	250	—	ns	
IS40	TAA:SCL	自时钟边沿到输出有效的的时间	100 kHz 模式	0	3500	ns	—
			400 kHz 模式	0	1000	ns	
			1 MHz 模式 (1)	0	350	ns	
IS45	TBF:SDA	总线空闲时间	100 kHz 模式	4.7	—	μs	在启动一个新的传输前总线必须保持空闲的时间
			400 kHz 模式	1.3	—	μs	
			1 MHz 模式 (1)	0.5	—	μs	
IS50	Cb	总线容性负载	—	400	pF	—	

注 1: 所有 I2Cx 引脚的最大引脚电容为 10 pF (仅对于 1 MHz 模式)。

图 30-22: ECAN 模块 I/O 时序特性

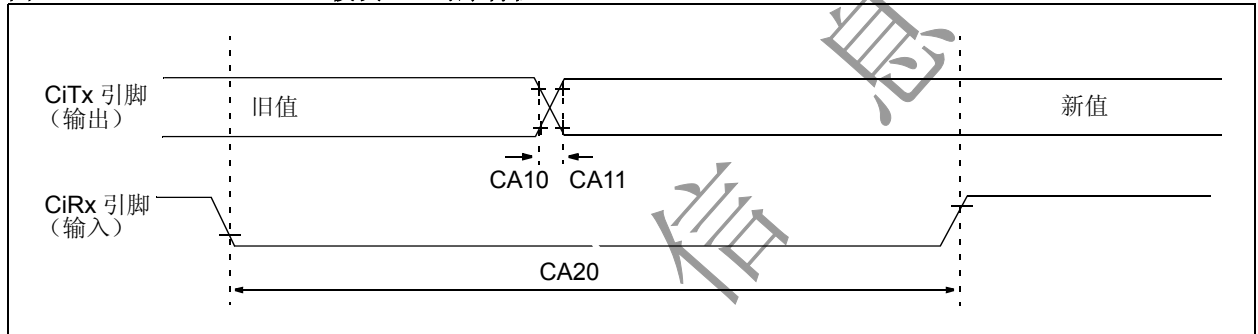


表 30-38: ECAN 模块 I/O 时序要求

交流特性		标准工作条件: 3.0V 至 3.6V (除非另外声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$					
参数编号	符号	特性 (1)	最小值	典型值 (2)	最大值	单位	条件
CA10	TioF	端口输出下降时间	—	—	—	ns	见参数 D032
CA11	TioR	端口输出上升时间	—	—	—	ns	见参数 D031
CA20	Tcwf	触发 CAN 唤醒滤波器的脉冲宽度	120			ns	—

注 1: 这些参数为特性值, 但生产时未经测试。

注 2: 除非另外声明, 否则“典型值”栏中的数据均为 5V 和  $25^{\circ}\text{C}$  条件下的值。这些参数仅供设计参考, 未经测试。

表 30-39: ADC 模块规范

交流特性		标准工作条件: 3.0V 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
器件电源							
AD01	AVDD	模块电源 VDD	VDD - 0.3 和 3.0 中的 较大值	—	VDD + 0.3 和 3.6 中的 较小值	V	—
AD02	AVSS	模块电源 VSS	VSS - 0.3	—	VSS + 0.3	V	—
参考输入							
AD05	VREFH	参考电压高电平	AVSS + 2.7	—	AVDD	V	见注 1
AD05a			3.0	—	3.6	V	VREFH = AVDD VREFL = AVSS = 0
AD06	VREFL	参考电压低电平	AVSS	—	AVDD - 2.7	V	见注 1
AD06a			0	—	0	V	VREFH = AVDD VREFL = AVSS = 0
AD07	VREF	绝对参考电压	2.7	—	3.6	V	VREF = VREFH - VREFL
AD08	IREF	电流消耗	—	400	550	μA	ADC 工作
				—	10	μA	ADC 关闭
模拟输入							
AD12	VINH	输入电压范围 VINH	VINL	—	VREFH	V	该电压反映采样 / 保持通道 0、1、2 和 3 (CH0-CH3) 的同相输入
AD13	VINL	输入电压范围 VINL	VREFL	—	AVSS + 1V	V	该电压反映采样 / 保持通道 0、1、2 和 3 (CH0-CH3) 的反相输入
AD17	RIN	模拟输入信号源的推荐阻抗	—	—	200	Ω	10 位 ADC
			—	—	200	Ω	12 位 ADC

注 1: 这些参数不是特性值, 或生产时未经测试。

表 30-40: ADC 模块规范 (12 位模式)

交流特性		标准工作条件: 3.0V 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>ADC 精度 (12 位模式) —— 测量采用外部 VREF+/VREF-</b>							
AD20a	Nr	分辨率	12 个数据位			位	
AD21a	INL	积分非线性误差	-2	—	+2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD22a	DNL	微分非线性误差	>-1	—	<1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD23a	GERR	增益误差	1.25	1.5	3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD24a	E <sub>OFF</sub>	失调误差	1.25	1.52	2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD25a	—	单调性	—	—	—	—	保证
<b>ADC 精度 (12 位模式) —— 测量采用内部 VREF+/VREF-</b>							
AD20a	Nr	分辨率	12 个数据位			位	
AD21a	INL	积分非线性误差	-2	—	+2	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD22a	DNL	微分非线性误差	>-1	—	<1	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD23a	GERR	增益误差	2	3	7	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD24a	E <sub>OFF</sub>	失调误差	2	3	5	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD25a	—	单调性	—	—	—	—	保证
<b>动态性能 (12 位模式)</b>							
AD30a	THD	总谐波失真	-77	-69	-61	dB	—
AD31a	SINAD	信噪比和失真	59	63	64	dB	—
AD32a	SFDR	无杂散动态范围	63	72	74	dB	—
AD33a	FNYQ	输入信号带宽	—	—	250	kHz	—
AD34a	ENOB	有效位数	10.95	11.1	—	位	—

表 30-41: ADC 模块规范 (10 位模式)

交流特性		标准工作条件: 3.0V 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>ADC 精度 (10 位模式) —— 测量采用外部 VREF+/VREF-</b>							
AD20b	Nr	分辨率	10 个数据位			位	
AD21b	INL	积分非线性误差	-1.5	—	+1.5	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD22b	DNL	微分非线性误差	>-1	—	<1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD23b	GERR	增益误差	1	3	6	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD24b	Eoff	失调误差	1	2	5	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3.6V
AD25b	—	单调性	—	—	—	—	保证
<b>ADC 精度 (10 位模式) —— 测量采用内部 VREF+/VREF-</b>							
AD20b	Nr	分辨率	10 个数据位			位	
AD21b	INL	积分非线性误差	-1	—	+1	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD22b	DNL	微分非线性误差	>-1	—	<1	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD23b	GERR	增益误差	1	5	6	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD24b	Eoff	失调误差	1	2	3	LSb	VINL = AVSS = 0V, AVDD = 3.6V
AD25b	—	单调性	—	—	—	—	保证
<b>动态性能 (10 位模式)</b>							
AD30b	THD	总谐波失真	—	-64	-67	dB	—
AD31b	SINAD	信噪比和失真	—	57	58	dB	—
AD32b	SFDR	无杂散动态范围	—	60	62	dB	—
AD33b	FNYQ	输入信号带宽	—	—	550	kHz	—
AD34b	ENOB	有效位数	9.1	9.7	9.8	位	—



图 30-23: ADC 转换（12 位模式）时序特性  
(ASAM = 0, SSRC<2:0> = 000)

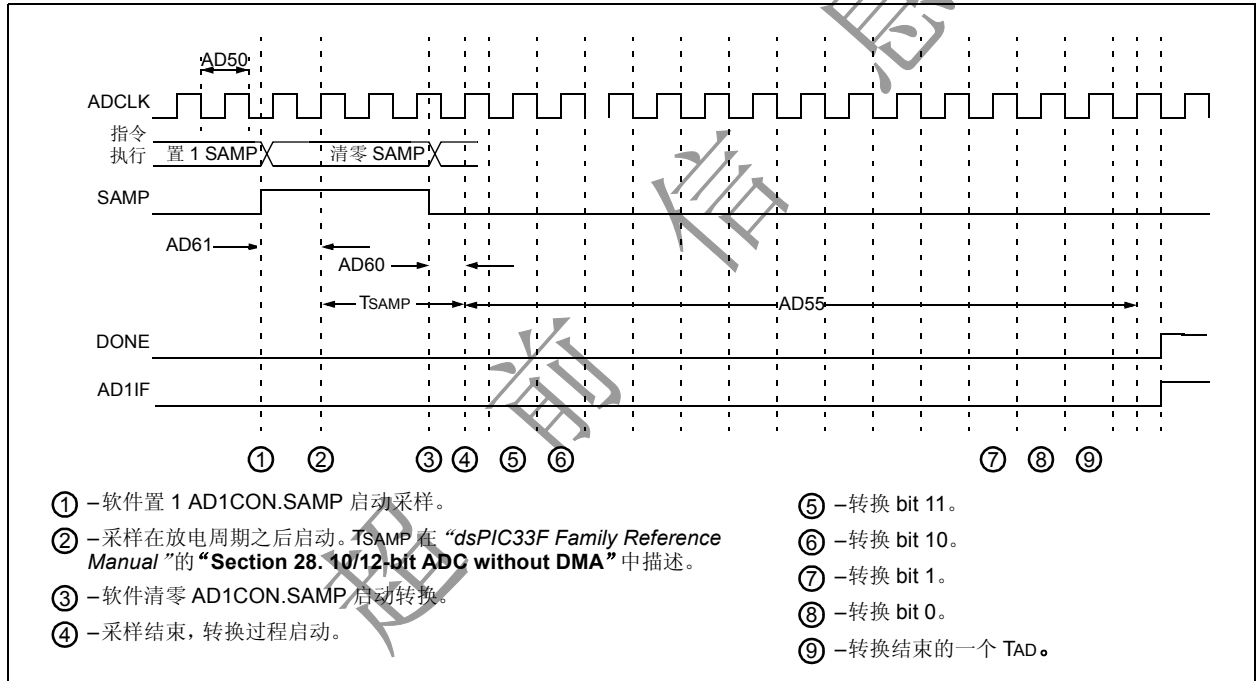


表 30-42: ADC 转换（12 位模式）时序要求

交流特性		标准工作条件: 3.0V 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	ADC 时钟周期	117.6	—	—	ns	
AD51	tRC	ADC 内部 RC 振荡器周期	—	250	—	ns	
<b>转换速率</b>							
AD55	tCONV	转换时间	—	14 TAD		ns	
AD56	FCNV	吞吐率	—	—	500	Ksps	
AD57	TSAMP	采样时间	3 TAD	—	—	—	
<b>时序参数</b>							
AD60	tPCS	从采样触发到转换开始的时间 (2)	—	1.0 TAD	—	—	未选择自动转换触发功能
AD61	tpSS	从采样位 (SAMP) 置 1 到采样启动的时间 (2)	0.5 TAD	—	1.5 TAD	—	—
AD62	tcSS	转换结束到采样启动 (ASAM = 1) 的时间 (2)	—	0.5 TAD	—	—	—
AD63	tDPU	从 ADC 关闭到 ADC 启动, 用于稳定模拟级的时间 (2)	—	20	—	μs	—

注 1: 因为采样电容最终将释放电荷, 因此低于 10 kHz 的时钟频率可能影响线性性能, 尤其是在温度较高时。  
注 2: 这些参数为特性值, 但生产时未经测试。

图 30-24: ADC 转换 (10 位模式) 时序特性  
(CHPS<1:0> = 01, SIMSAM = 0, ASAM = 0, SSRC<2:0> = 000)

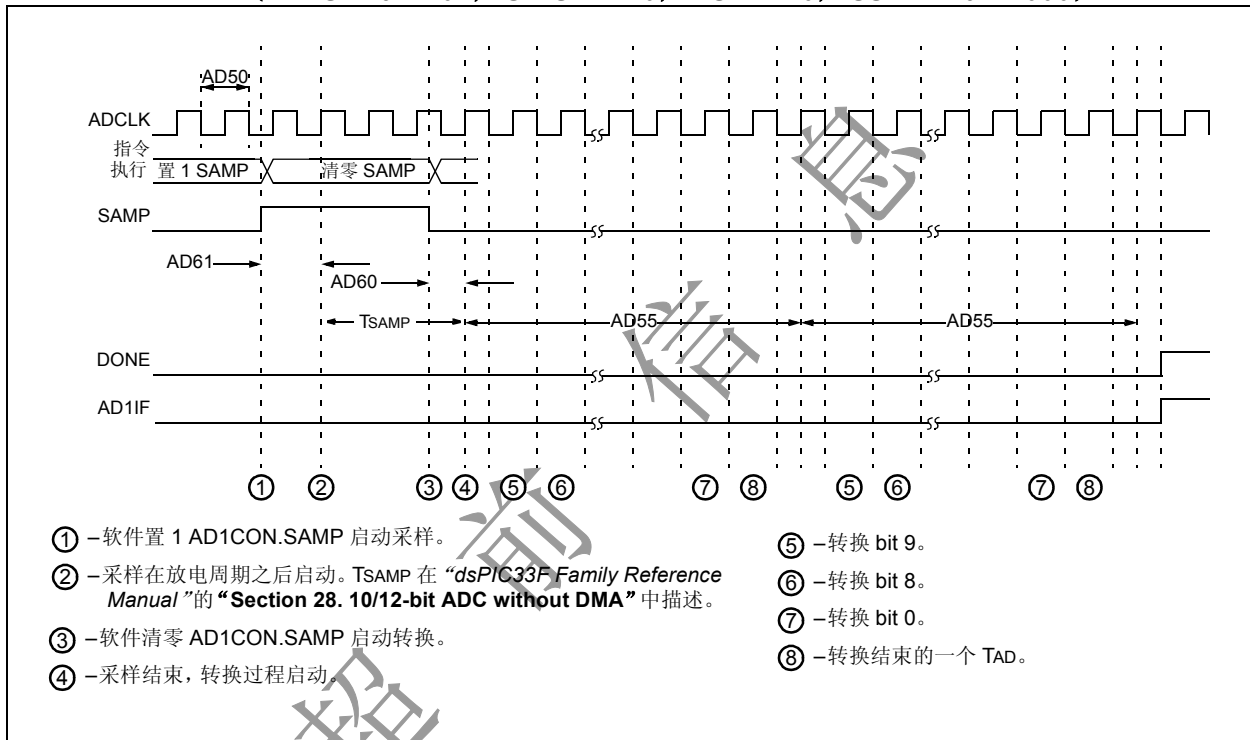


图 30-25: ADC 转换 (10 位模式) 时序特性 (CHPS<1:0> = 01, SIMSAM = 0, ASAM = 1, SSRC<2:0> = 111, SAMC<4:0> = 00001)

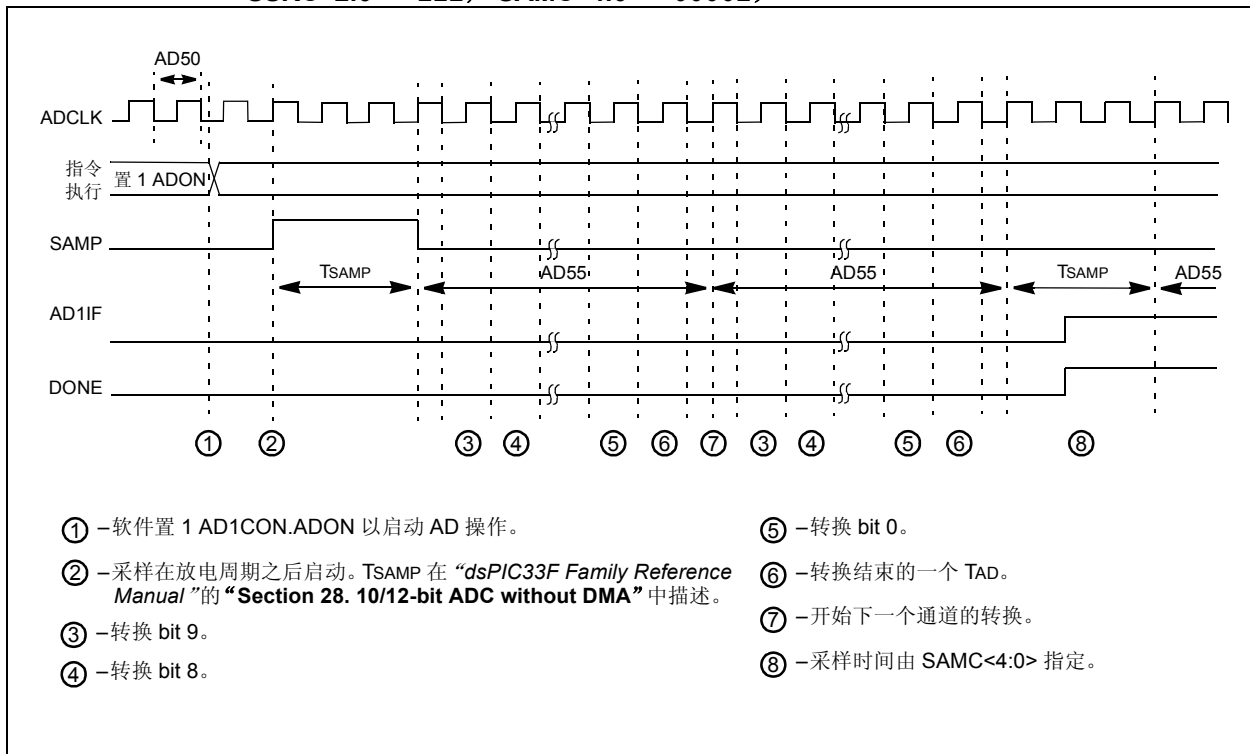


表 30-43: ADC 转换 (10 位模式) 时序要求

交流特性		标准工作条件: 3.0V 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
<b>时钟参数</b>							
AD50	TAD	ADC 时钟周期	76	—	—	ns	
AD51	tRC	ADC 内部 RC 振荡器周期	—	250	—	ns	
<b>转换速率</b>							
AD55	tCONV	转换时间	—	12 TAD	—	—	
AD56	FCNV	吞吐率	—	—	1.1	Msp/s	
AD57	TSAMP	采样时间	2 TAD	—	—	—	
<b>时序参数</b>							
AD60	tPCS	从采样触发到转换开始的时间 <sup>(1)</sup>	—	1.0 TAD	—	—	未选择自动转换触发功能
AD61	tPSS	从采样位 (SAMP) 置 1 到采样启动的时间 <sup>(1)</sup>	0.5 TAD	—	1.5 TAD	—	—
AD62	tcSS	转换结束到采样启动 (ASAM = 1) 的时间 <sup>(1)</sup>	—	0.5 TAD	—	—	—
AD63	tDPU	从 ADC 关闭到 ADC 启动, 用于稳定模拟级的时间 <sup>(1)</sup>	—	20	—	μs	—

注 1: 这些参数为特性值, 但生产时未经测试。

2: 因为采样电容最终将释放电荷, 因此低于 10 kHz 的时钟频率可能影响线性性能, 尤其是在温度较高时。

表 30-44: DAC 模块规范

交流特性		标准工作条件: 3.0V 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>							
	VOMAX	最大输出电压	2.65	—	—	V	
	VOMIN	最小输出电压	—	—	0.65	V	
	VRES	分辨率	—	16 位	—	—	
		增益误差	—	-2%	—	—	
		失调误差	—	—	±30	MV	取决于参考电压稳定性
		微分非线性 (DNL) 误差	—	±0.1	—	LSB	相对于 14 位精度
		积分非线性 (INL) 误差	—	±0.2	—	LSB	相对于 14 位精度

表 30-45: DAC 模块规范

交流特性		标准工作条件: 3.0V 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
时钟参数							
		时钟频率	—	—	25.6	MHz	时钟
		采样速率	0	—	100	kHz	
		输入数据频率	0	—	45	kHz	采样频率 = 100 kHz
		初始化周期	1024	—	—	Cks	第一次采样之前的时间
		信噪比	—	90	—	dB	采样频率 = 96 kHz
		信噪比和失真比	82	—	—	dB	采样频率 = 96 kHz

表 30-46: 比较器时序规范

交流特性		标准工作条件: 3.0V 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
300	TRESP	响应时间 (1,2)	—	150	400	ns	
301	TMC2OV	比较器模式改变到输出有效的时间 (1)	—	—	10	$\mu\text{s}$	

注 1: 这些参数为特性值, 但未经测试。

2: 当比较器的一个输入为  $(V_{DD} - 1.5)/2$  而另一个输入从  $V_{SS}$  跳变到  $V_{DD}$  时, 测量响应时间。

表 30-47: 比较器模块规范

直流特性		标准工作条件: 3.0V 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
D300	VIOFF	输入失调电压 (1)	—	$\pm 10$	—	mV	
D301	VICM	输入共模电压 (1)	0	—	$A_{VDD}-1.5V$	V	
D302	CMRR	共模抑制比 (1)	-54	—	—	dB	

注 1: 这些参数为特性值, 但未经测试。

表 30-48: 比较器参考电压稳定时间规范

交流特性			标准工作条件: 3.0V 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
VR310	TSET	稳定时间 (1)	—	—	10	μs	

注 1: 稳定时间是在 CVRR = 1 且 CVR3:CVR0 位从 0000 跳变到 1111 时测得的。

表 30-49: 比较器参考电压规范

直流特性			标准工作条件: 3.0V 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
VRD310	CVRES	分辨率	CVRSRC/24	—	CVRSRC/32	LSb	
VRD311	CVRAA	绝对精度	—	—	0.5	LSb	
VRD312	CVRUR	单位电阻值 (R)	—	2k	—	Ω	

图 30-26: 并行从端口时序图

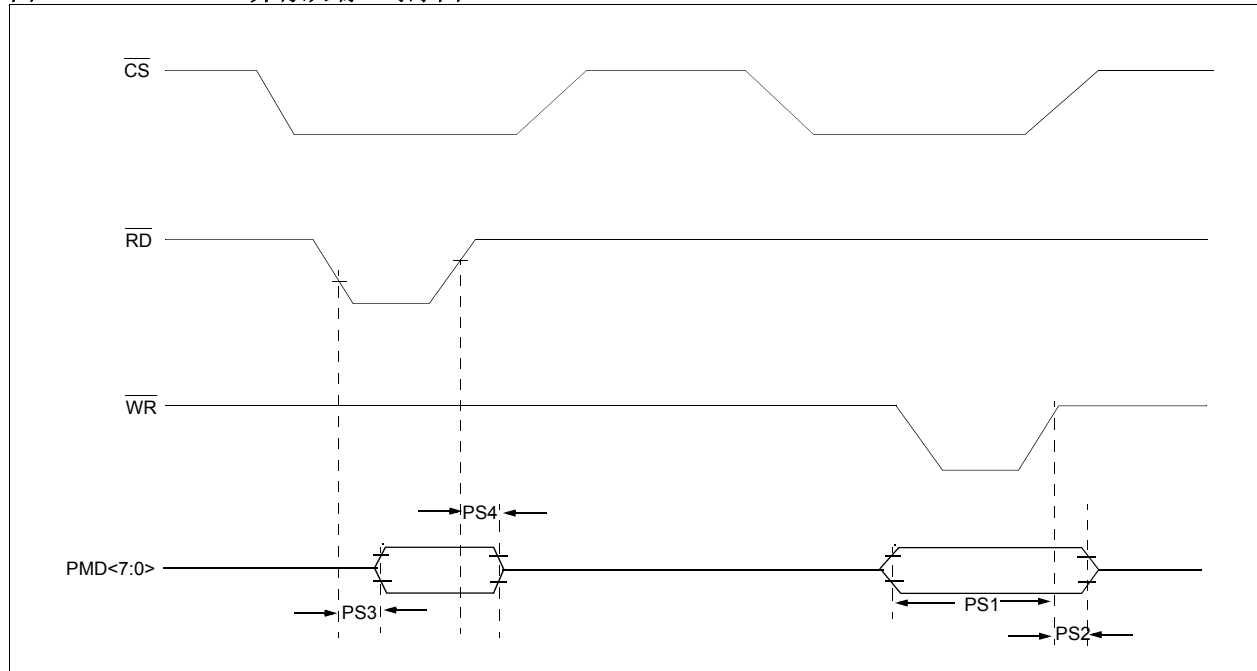


表 30-50: 稳定时间规范

交流特性			标准工作条件: 3.0V 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
PS1	TdtV2wrH	在 $\overline{WR}$ 或 $\overline{CS}$ 无效之前数据输入有效的时间 (建立时间)	20	—	—	ns	
PS2	TwrH2dtl	$\overline{WR}$ 或 $\overline{CS}$ 无效到数据输入无效的时间 (保持时间)	20	—	—	ns	
PS3	TrdL2dtV	$\overline{RD}$ 和 $\overline{CS}$ 有效到数据输出有效的时间	—	—	80	ns	
PS4	TrdL2dtl	$\overline{RD}$ 有效或 $\overline{CS}$ 无效到数据输出无效的时间	10	—	30	ns	

图 30-27: 并行主端口读时序图

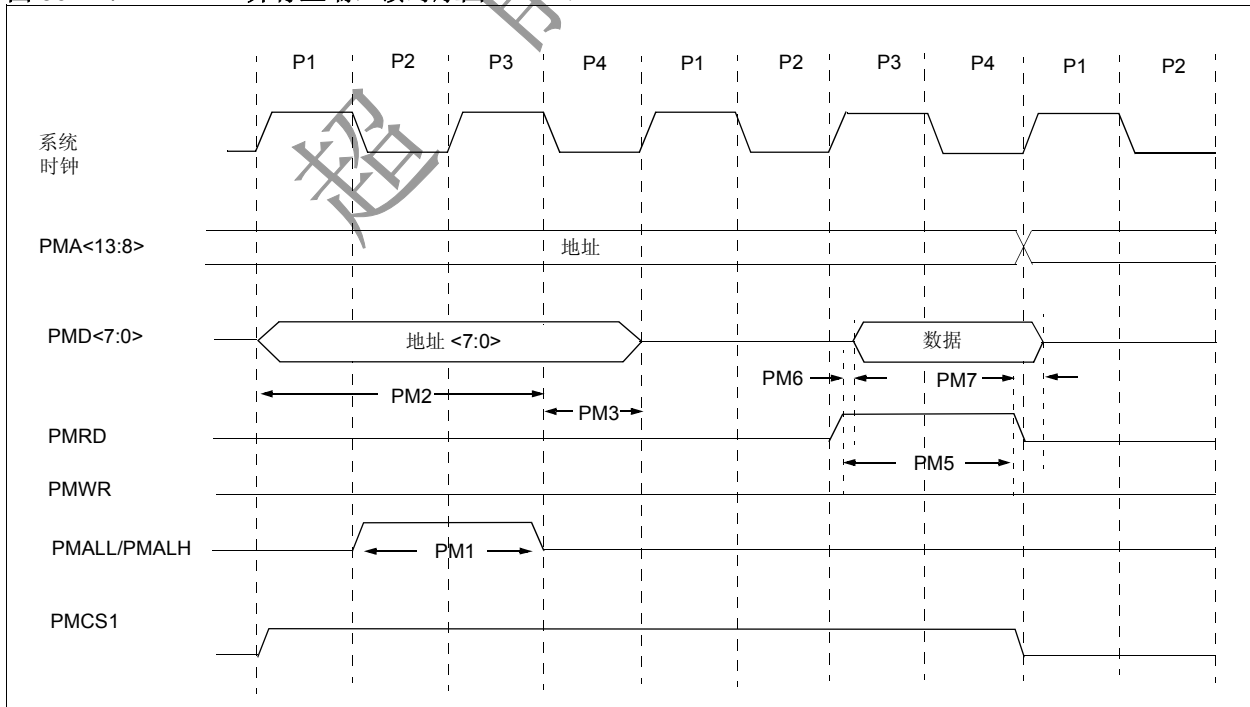
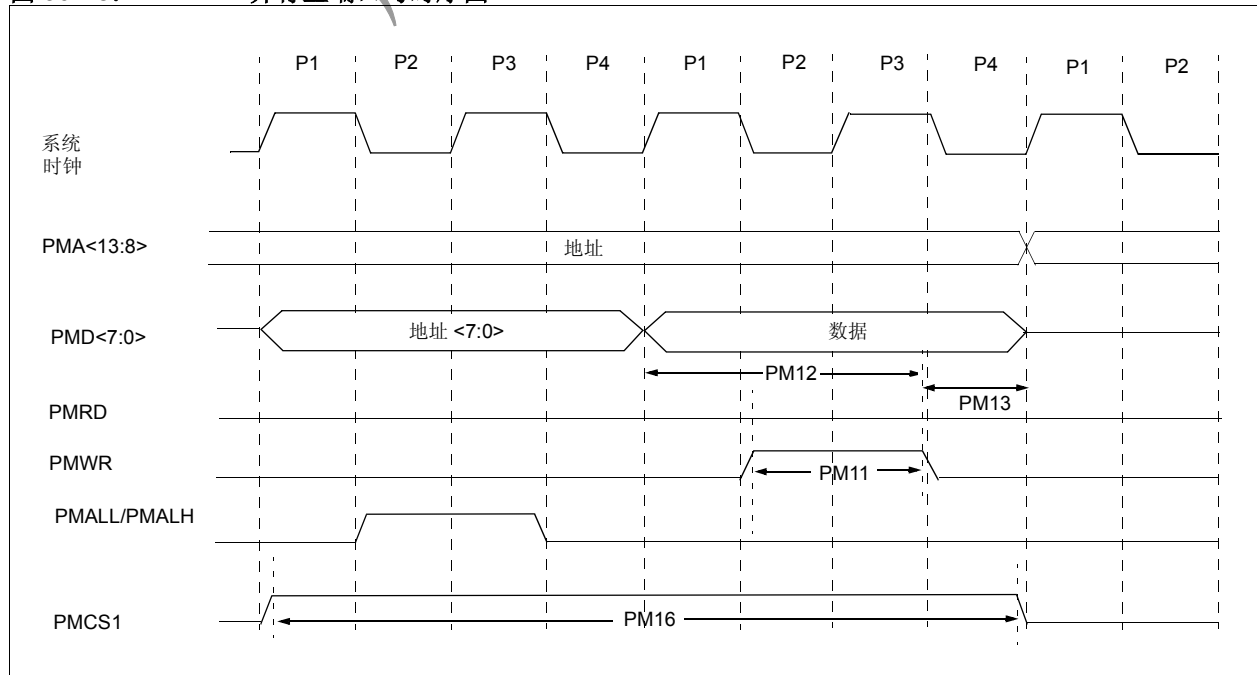


表 30-51: 并行主端口读时序要求

交流特性		标准工作条件: 3.0V 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
PM1	PMALL/PMALH 脉冲宽度	—	0.5 Tcy	—	ns	
PM2	地址输出有效到 PMALL/PMALH 无效的时间 (地址建立时间)	—	0.75 Tcy	—	ns	
PM3	PMALL/PMALH 无效到地址输出无效的时间 (地址保持时间)	—	0.25 Tcy	—	ns	
PM5	PMRD 脉冲宽度	—	0.5 Tcy	—	ns	
PM6	PMRD 或 PMENB 有效到数据输入有效的的时间 (数据建立时间)	—	—	—	ns	
PM7	PMRD 或 PMENB 无效到数据输入无效的时间 (数据保持时间)	—	—	—	ns	

图 30-28: 并行主端口写时序图



**表 30-52: 并行主端口写时序要求**

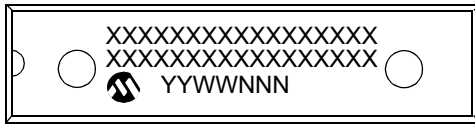
交流特性		标准工作条件: 3.0V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	特性	最小值	典型值	最大值	单位	条件
PM11	PMWR 脉冲宽度	—	0.5 T <sub>CY</sub>	—	ns	
PM12	PMWR 或 PMENB 变为无效之前数据输出有效的时间 (数据建立时间)	—	—	—	ns	
PM13	PMWR 或 PMEMB 无效到数据输出无效的时间 (数据保持时间)	—	—	—	ns	
PM16	PMCSx 脉冲宽度	T <sub>CY</sub> - 5	—	—	ns	

超前

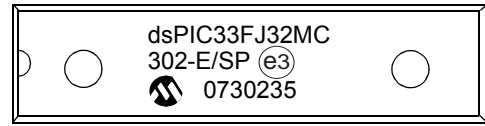


### 31.0 封装信息

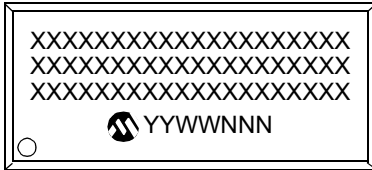
28 引脚 SPDIP



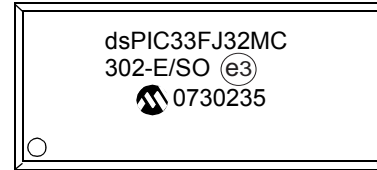
示例



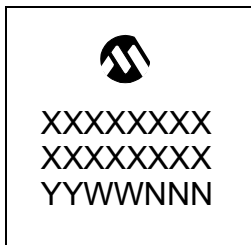
28 引脚 SOIC (.300")



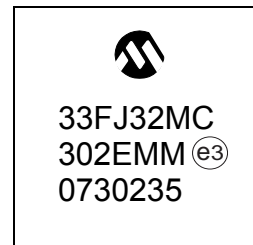
示例



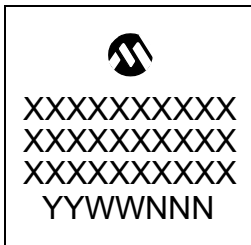
28 引脚 QFN-S



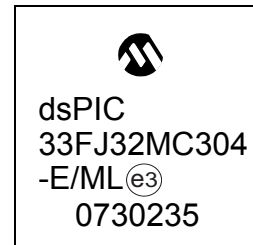
示例



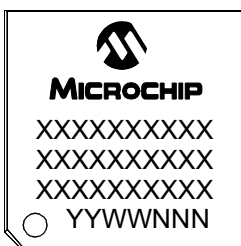
44 引脚 QFN



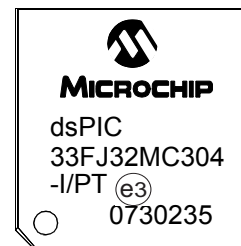
示例



44 引脚 TQFP



示例



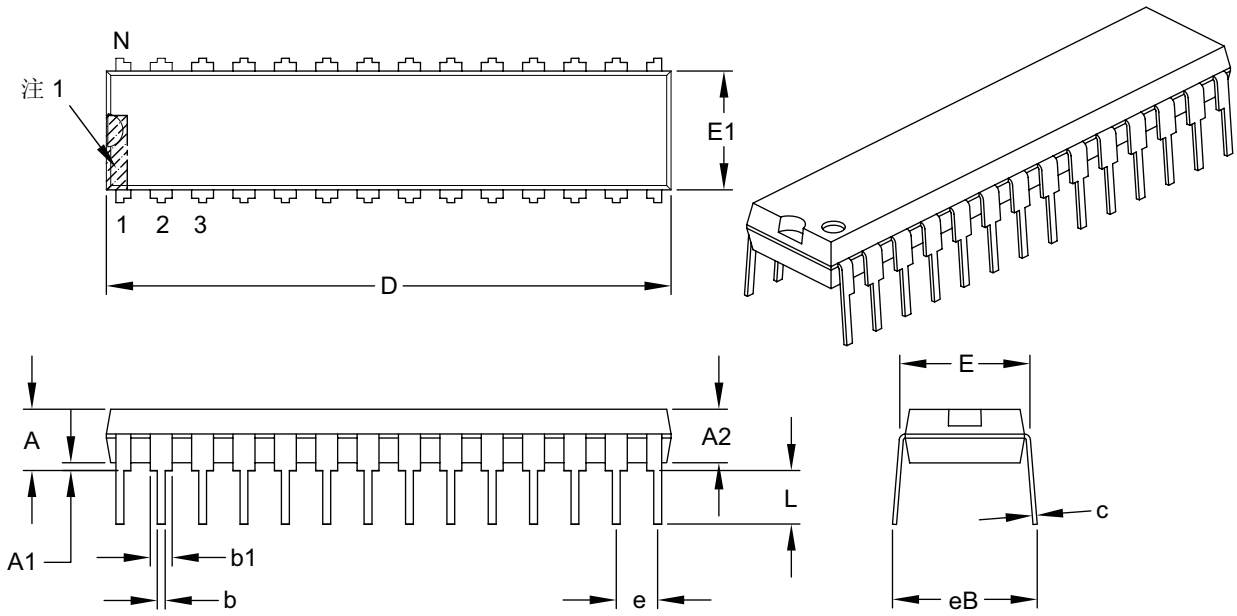
**图注:** XX...X 客户信息  
 Y 年份代码 (日历年的最后一位数字)  
 YY 年份代码 (日历年的最后两位数字)  
 WW 星期代码 (一月一日的星期代码为“01”)  
 NNN 以字母数字排序的追踪代码  
 (e3) 雾锡 (Matte Tin, Sn) 的 JEDEC 无铅标志  
 \* 表示无铅封装。JEDEC 无铅标志 (e3) 标示于此种封装的外包装上。

**注:** Microchip 元器件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户信息的字符数。

31.1 封装详细信息

28引脚窄型塑封双列直插式封装（SP）——300 mil主体 [SPDIP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



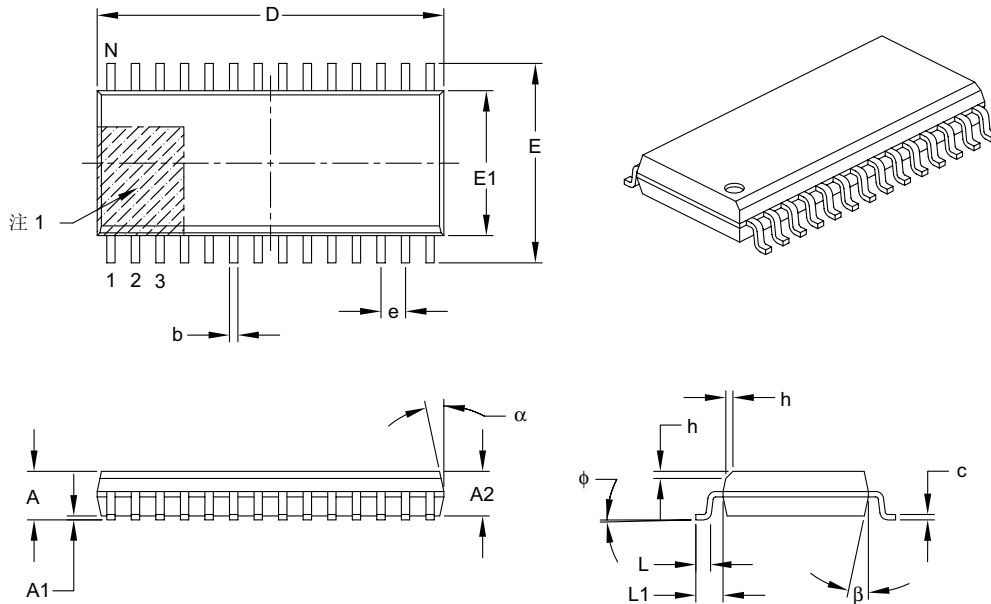
尺寸范围	单位	英寸		
		最小	正常	最大
引脚数	N	28		
引脚间距	e	.100 BSC		
顶端到固定面高度	A	-	-	.200
塑模封装厚度	A2	.120	.135	.150
塑模底面到固定面高度	A1	.015	-	-
肩到肩宽度	E	.290	.310	.335
塑模封装宽度	E1	.240	.285	.295
总长度	D	1.345	1.365	1.400
引脚尖到固定面高度	L	.110	.130	.150
引脚厚度	c	.008	.010	.015
引脚上部宽度	b1	.040	.050	.070
引脚下部宽度	b	.014	.018	.022
总排列间距 §	eB	-	-	.430

- 注：
1. 引脚1的可视定位标记可能不同，但必须在阴影区域内。
  2. § 重要特性。
  3. 尺寸D和E1不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过0.010英寸。
  4. 尺寸和公差遵循ASME Y14.5M。  
BSC：基本尺寸。显示的是没有公差的理论精确值。

Microchip Technology图号C04-070B

28引脚塑封小外形封装（SO）——宽条，7.50 mm主体 [SOIC]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



尺寸范围	单位	毫米		
		最小	正常	最大
引脚数	N	28		
引脚间距	e	1.27 BSC		
总高度	A	—	—	2.65
塑模封装厚度	A2	2.05	—	—
悬空间隙 §	A1	0.10	—	0.30
总宽度	E	10.30 BSC		
塑模封装宽度	E1	7.50 BSC		
总长度	D	17.90 BSC		
斜面投影距离（可选）	h	0.25	—	0.75
底脚长度	L	0.40	—	1.27
引脚投影长度	L1	1.40 REF		
底脚倾斜角	$\phi$	0°	—	8°
引脚厚度	c	0.18	—	0.33
引脚宽度	b	0.31	—	0.51
塑模顶部锥度	$\alpha$	5°	—	15°
塑模底部锥度	$\beta$	5°	—	15°

注：

1. 引脚1的可视定位标记可能不同，但必须在阴影区域内。
2. § 重要特性。
3. 尺寸D和E1不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过0.15毫米。
4. 尺寸和公差遵循ASME Y14.5M。

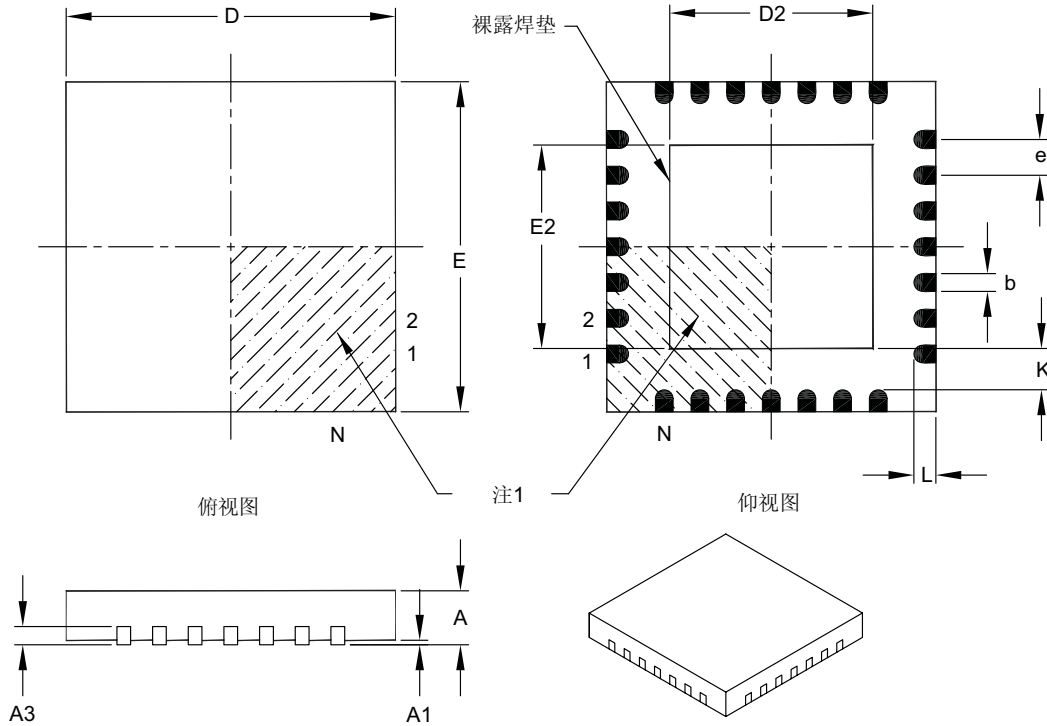
BSC：基本尺寸。显示的是没有公差的理论精确值。

REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology图号C04-052B

28引脚塑封正方扁平无脚封装（MM）——6x6x0.9 mm主体 [QFN-S]，0.40 mm触点长度

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



尺寸范围	单位	毫米		
		最小	正常	最大
引脚数	N	28		
引脚间距	e	0.65 BSC		
总高度	A	0.80	0.90	1.00
悬空间隙	A1	0.00	0.02	0.05
触点厚度	A3	0.20 REF		
总宽度	E	6.00 BSC		
裸露金属焊垫宽度	E2	3.65	3.70	4.70
总长度	D	6.00 BSC		
裸露金属焊垫长度	D2	3.65	3.70	4.70
触点宽度	b	0.23	0.38	0.43
触点长度	L	0.30	0.40	0.50
触点到裸露金属焊垫的距离	K	0.20	-	-

注：

1. 引脚1的可视定位标记可能不同，但必须在阴影区域内。
2. 封装为切割分离。
3. 尺寸和公差遵循ASME Y14.5M。

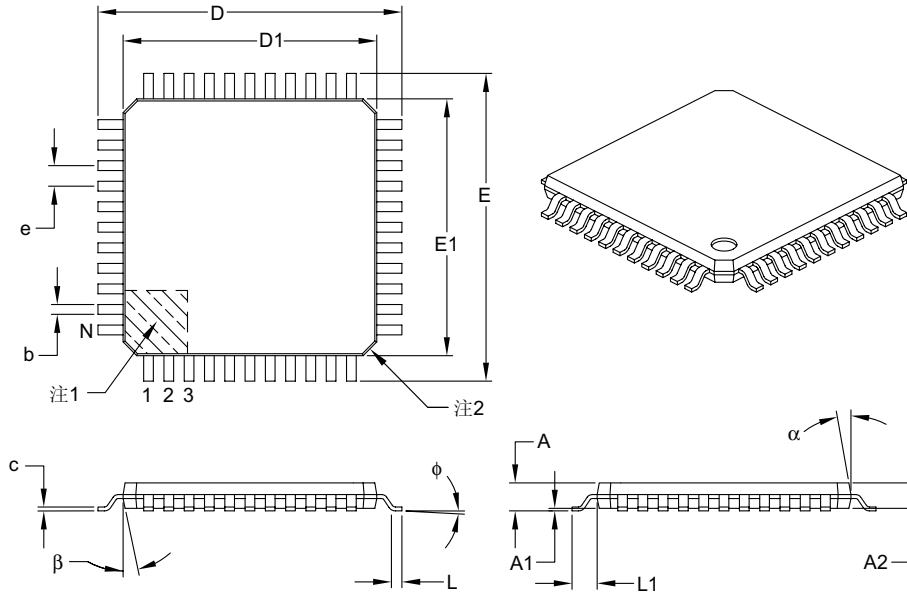
BSC：基本尺寸。显示的是没有公差的理论精确值。

REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology 图号C04-124B

44引脚塑封薄型正方扁平封装（PT）——10x10x1 mm主体，2.00 mm引脚投影长度 [TQFP]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N	44		
引脚间距	e	0.80 BSC		
总高度	A	—	—	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	—	0.15
底脚长度	L	0.45	0.60	0.75
引脚投影长度	L1	1.00 REF		
底脚倾斜角	φ	0°	3.5°	7°
总宽度	E	12.00 BSC		
总长度	D	12.00 BSC		
塑模封装宽度	E1	10.00 BSC		
塑模封装长度	D1	10.00 BSC		
引脚厚度	c	0.09	—	0.20
引脚宽度	b	0.30	0.37	0.45
塑模顶部锥度	α	11°	12°	13°
塑模底部锥度	β	11°	12°	13°

注：

1. 引脚1的可视定位标记可能不同，但必须在阴影区域内。
2. 切角处的斜面是可选的；尺寸可能有所不同。
3. 尺寸D1和E1不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过0.25毫米。
4. 尺寸和公差遵循ASME Y14.5M。

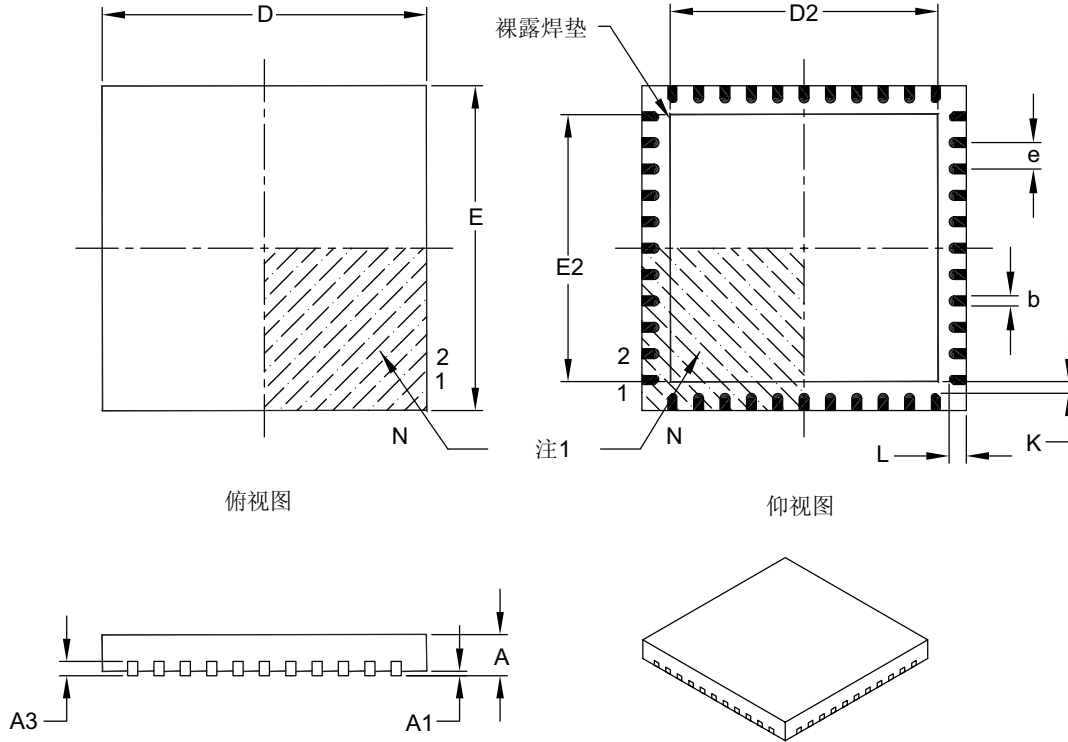
BSC：基本尺寸。显示的是没有公差的理论精确值。

REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology 图号C04-076B

44引脚塑封正方扁平无脚封装（ML）——8x8 mm主体 [QFN]

注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。



尺寸范围	单位	毫米		
		最小	正常	最大
引脚数	N	44		
引脚间距	e	0.65 BSC		
总高度	A	0.80	0.90	1.00
悬空间隙	A1	0.00	0.02	0.05
触点厚度	A3	0.20 REF		
总宽度	E	8.00 BSC		
裸露金属焊垫宽度	E2	6.30	6.45	6.80
总长度	D	8.00 BSC		
裸露金属焊垫长度	D2	6.30	6.45	6.80
触点宽度	b	0.25	0.30	0.38
触点长度	L	0.30	0.40	0.50
触点到裸露金属焊垫的距离	K	0.20	-	-

注：

1. 引脚1的可视定位标记可能不同，但必须在阴影区域内。
2. 封装为切割分离。
3. 尺寸和公差遵循ASME Y14.5M。

BSC: 基本尺寸。显示的是没有公差的理论精确值。

REF: 参考尺寸，通常无公差，仅供参考。

Microchip Technology 图号C04-103B

**附录 A: 版本历史**

**版本 A (2007 年 8 月)**

本文档的初始版本。

注:



## 索引

## A

A/D 转换器 .....	281
初始化 .....	281
DMA .....	281
关键特性 .....	281
ADC 模块 .....	
ADC1 寄存器映射 .....	45, 46

## B

备用向量表 (AIVT) .....	81
比较器模块 .....	295
变更通知客户服务 .....	413

## C

C 编译器 .....	
MPLAB C18 .....	352
MPLAB C30 .....	352

## CPU

控制寄存器 .....	22
CPU 的特殊功能 .....	333
CPU 时钟系统 .....	136
PLL 配置 .....	137
选择 .....	136
源 .....	136
操作码说明中使用的符号 .....	344

## 程序存储器

复位向量 .....	32
构成 .....	32
中断向量 .....	32
程序地址空间 .....	31
表读指令 .....	
TBLRDH .....	65
TBLRDL .....	65

存储器映射 .....	31
访问程序空间中数据的地址生成 .....	64
构成 .....	63
可视性操作 .....	66
使用表指令访问程序存储器中的数据 .....	65
使用程序空间可视性访问程序存储器中的数据 .....	66
串行外设接口 (SPI) .....	225
存储器构成 .....	31

## D

打盹模式 .....	148
DMAC 寄存器 .....	125
DMAxCNT .....	125
DMAxCON .....	125
DMAxPAD .....	125
DMAxREQ .....	125
DMAxSTA .....	125
DMAxSTB .....	125

## DMA 模块

DMA 寄存器映射 .....	47
-----------------	----

DSP 引擎 .....	26
乘法器 .....	28

代码保护 .....	333, 339
------------	----------

## 代码示例

擦除程序存储器页 .....	71
端口写 / 读 .....	150
PWRSAV 指令语法 .....	147
启动编程序列 .....	72
装载写缓冲区 .....	72
电机控制 PWM .....	195
电机控制 PWM 模块 .....	
2 路输出寄存器映射 .....	43

6 路输出寄存器映射 .....	42
电气特性 .....	355
AC .....	364
读者反馈表 .....	414

## E

ECAN 发送 / 接收错误计数寄存器 (CiEC) .....	265
ECAN 发送 / 接收缓冲区 m 控制寄存器 (CiTRmnCON) .....	276

## ECAN 寄存器

波特率配置寄存器 1 (CiCFG1) .....	265
波特率配置寄存器 2 (CiCFG2) .....	266
FIFO 控制寄存器 (CiFCTRL) .....	261
FIFO 状态寄存器 (CiFIFO) .....	262
过滤器 0-3 缓冲区指针寄存器 (CiBUFNT1) .....	267
过滤器 4-7 缓冲区指针寄存器 (CiBUFNT2) .....	268
过滤器 7-0 屏蔽选择寄存器 (CiFMSKSEL1) .....	271
过滤器 8-11 缓冲区指针寄存器 (CiBUFNT3) .....	268
过滤器 12-15 缓冲区指针寄存器 (CiBUFNT4) .....	269
过滤器 15-8 屏蔽选择寄存器 (CiFMSKSEL2) .....	272
接收过滤屏蔽寄存器标准标识符寄存器 n (CiRXMnSID) .....	273
接收过滤屏蔽寄存器扩展标识符寄存器 n (CiRXMnEID) .....	273
接收过滤器标准标识符寄存器 n (CiRXFnSID) .....	270
接收过滤器扩展标识符寄存器 n (CiRXFnEID) .....	271
接收过滤器使能寄存器 (CiFEN1) .....	267
接收缓冲区满寄存器 1 (CiRXFUL1) .....	274
接收缓冲区满寄存器 2 (CiRXFUL2) .....	274
接收缓冲区溢出寄存器 1 (CiRXOVF1) .....	275
接收缓冲区溢出寄存器 2 (CiRXOVF2) .....	275
控制寄存器 1 (CiCTRL1) .....	258
控制寄存器 2 (CiCTRL2) .....	259
中断编码寄存器 (CiVEC) .....	260
中断标志寄存器 (CiINTF) .....	263
中断允许寄存器 (CiINTE) .....	264

## ECAN 模块

报文发送 .....	255
报文接收 .....	253
波特率设置 .....	256
CiBUFNT1 寄存器 .....	267
CiBUFNT2 寄存器 .....	268
CiBUFNT3 寄存器 .....	268
CiBUFNT4 寄存器 .....	269
CiCFG1 寄存器 .....	265
CiCFG2 寄存器 .....	266
CiCTRL1 寄存器 .....	258
CiCTRL2 寄存器 .....	259
CiEC 寄存器 .....	265
CiFCTRL 寄存器 .....	261
CiFEN1 寄存器 .....	267
CiFIFO 寄存器 .....	262
CiFMSKSEL1 寄存器 .....	271
CiFMSKSEL2 寄存器 .....	272
CiINTE 寄存器 .....	264
CiINTF 寄存器 .....	263
CiRXFnEID 寄存器 .....	271
CiRXFnSID 寄存器 .....	270
CiRXFUL1 寄存器 .....	274
CiRXFUL2 寄存器 .....	274
CiRXMnEID 寄存器 .....	273
CiRXMnSID 寄存器 .....	273
CiRXOVF1 寄存器 .....	275
CiRXOVF2 寄存器 .....	275
CiTRmnCON 寄存器 .....	276
CiVEC 寄存器 .....	260

ECAN1 寄存器映射 (C1CTRL1.WIN = 0 或 1) .....	49	AD1CON2 (ADC1 控制寄存器 2) .....	287
ECAN1 寄存器映射 (C1CTRL1.WIN = 0) .....	49	AD1CON3 (ADC1 控制寄存器 3) .....	288
ECAN1 寄存器映射 (C1CTRL1.WIN = 1) .....	50	AD1CON4 (ADC1 控制寄存器 4) .....	289
概述 .....	251	AD1CSSL (ADC1 输入扫描选择寄存器的低位字) .....	294
工作模式 .....	253	AD1PCFGL (ADC1 端口配置寄存器的低位字) ....	294
帧类型 .....	251	CiBUFNT1 (ECAN 过滤器 0-3 缓冲区指针 寄存器) .....	267
<b>F</b>		CiBUFNT2 (ECAN 过滤器 4-7 缓冲区指针 寄存器) .....	268
封装 .....	399	CiBUFNT3 (ECAN 过滤器 8-11 缓冲区指针 寄存器) .....	268
标识 .....	399	CiBUFNT4 (ECAN 过滤器 12-15 缓冲区指针 寄存器) .....	269
详细信息 .....	400	CiCFG1 (ECAN 波特率配置寄存器 1) .....	265
复位 .....	73	CiCFG2 (ECAN 波特率配置寄存器 2) .....	266
非法操作码 .....	73, 80	CiCTRL1 (ECAN 控制寄存器 1) .....	258
未初始化的 W 寄存器 .....	73, 80	CiCTRL2 (ECAN 控制寄存器 2) .....	259
陷阱冲突 .....	79, 80	CiEC (ECAN 发送 / 接收错误计数寄存器) .....	265
复位过程 .....	81	CiFCTRL (ECAN FIFO 控制寄存器) .....	261
<b>G</b>		CiFEN1 (ECAN 接收过滤器使能寄存器) .....	267
公式		CiFIFO (ECAN FIFO 状态寄存器) .....	262
A/D 转换时钟周期 .....	284	CiFMSKSEL1 (ECAN 过滤器 7-0 屏蔽选择 寄存器) .....	271, 272
串行时钟速率 .....	233	CiINTE (ECAN 中断允许寄存器) .....	264
PWM 分辨率 .....	199	CiINTF (ECAN 中断标志寄存器) .....	263
PWM 周期 .....	199	CiRXFnEID (ECAN 接收过滤器 n 扩展标识符 寄存器) .....	271
器件工作频率 .....	136	CiRXFnSID (ECAN 接收过滤器 n 标准标识符 寄存器) .....	270
器件和 SPI 时钟速度之间的关系 .....	228	CiRXFUL1 (ECAN 接收缓冲区满寄存器 1) .....	274
时钟发生的时间份额 .....	257	CiRXFUL2 (ECAN 接收缓冲区满寄存器 2) .....	274
UART 波特率 (BRGH = 0) .....	244	CiRXMnEID (ECAN 接收过滤屏蔽寄存器 n 扩展 标识符寄存器) .....	273
UART 波特率 (BRGH = 1) .....	244	CiRXMnSID (ECAN 接收过滤屏蔽寄存器 n 标准 标识符寄存器) .....	273
工作模式		CiRXOVF1 (ECAN 接收缓冲区溢出寄存器 1) .....	275
初始化 .....	253	CiRXOVF2 (ECAN 接收缓冲区溢出寄存器 2) .....	275
环回 .....	253	CiTRBnSID (ECAN 缓冲区 n 标准标识符 寄存器) .....	277, 278, 280
监听 .....	253	CiTRmCON (ECAN 发送 / 接收缓冲区 m 控制 寄存器) .....	276
监听所有报文 .....	253	CiVEC (ECAN 中断编码寄存器) .....	260
禁止 .....	253	CLKDIV (时钟分频比) .....	141
正常工作 .....	253	CORCON (内核控制) .....	24, 86
<b>H</b>		DFLTCON (QEI 控制寄存器) .....	223
红外支持		DMACS0 (DMA 控制器状态 0) .....	130
内置 IrDA 编码器和解码器 .....	245	DMACS1 (DMA 控制器状态寄存器 1) .....	132
外部 IrDA, IrDA 时钟输出 .....	245	DMAxCNT (DMA 通道 x 传输计数) .....	129
汇编器		DMAxCON (MA 通道 x 控制) .....	126
MPASM 汇编器 .....	352	DMAxPAD (DMA 通道 x 外设地址) .....	129
<b>I</b>		DMAxREQ (DMA 通道 x IRQ 选择) .....	127
I/O 端口 .....	149	DMAxSTA (DMA 通道 x RAM 起始地址 A) .....	128
并行 I/O (PIO) .....	149	DMAxSTB (DMA 通道 x RAM 起始地址 B) .....	128
写 / 读时序 .....	150	DSADR (最近的 DMA RAM 地址) .....	133
I <sup>2</sup> C		I2CxCON (I2Cx 控制寄存器) .....	237
波特率发生器 .....	233	I2CxMSK (I2Cx 从模式地址掩码寄存器) .....	241
从地址掩码 .....	235	I2CxSTAT (I2Cx 状态寄存器) .....	239
地址 .....	235	IFS0 (中断标志状态寄存器 0) .....	90, 97
工作模式 .....	233	IFS1 (中断标志状态寄存器 1) .....	92, 99
广播呼叫地址支持 .....	235	IFS2 (中断标志状态寄存器 2) .....	94, 101
寄存器 .....	233	IFS3 (中断标志状态寄存器 3) .....	95, 102
IPMI 支持 .....	235	IFS4 (中断标志状态寄存器 4) .....	96, 103
软件控制的时钟延长 (STREN = 1) .....	235	INTCON1 (中断控制寄存器 1) .....	87
斜率控制 .....	236	INTCON2 (中断控制寄存器 2) .....	89
中断 .....	233	INTTREG 中断控制和状态寄存器 .....	121
主模式工作		IPC0 (中断优先级控制寄存器 0) .....	104
多主器件通信、总线冲突和总线仲裁 .....	236	IPC11 (中断优先级控制寄存器 11) .....	114
时钟仲裁 .....	236		
<b>J</b>			
寄存器			
AD1CHS0 (ADC1 输入通道 0 选择) .....	292		
AD1CHS123 (ADC1 输入通道 1、2 和 3 选择) ...	290		
AD1CON1 (ADC1 控制寄存器 1) .....	285		

IPC14 (中断优先级控制寄存器 14) .....	115	勘误表 .....	11
IPC15 (中断优先级控制寄存器 15) .....	116	客户通知服务 .....	413
IPC16 (中断优先级控制寄存器 16) .....	117	客户支持 .....	413
IPC17 (中断优先级控制寄存器 17) .....	118	框图	
IPC18 (中断优先级控制寄存器 18) .....	119, 120	16 位 Timer1 模块 .....	187
IPC1 (中断优先级控制寄存器 1) .....	105	A/D 模块 .....	282, 283
IPC2 (中断优先级控制寄存器 2) .....	106	dsPIC33FJ32MC302/304、 dsPIC33FJ64MCX02/X04	
IPC3 (中断优先级控制寄存器 3) .....	107	和 dsPIC33FJ128MCX02/X04 的 CPU 内核 .....	20
IPC4 (中断优先级控制寄存器 4) .....	108	DSP 引擎 .....	27
IPC5 (中断优先级控制寄存器 5) .....	109	ECAN 模块 .....	252
IPC6 (中断优先级控制寄存器 6) .....	110	复位系统 .....	73
IPC7 (中断优先级控制寄存器 7) .....	111	共用端口结构 .....	149
IPC8 (中断优先级控制寄存器 8) .....	112	看门狗定时器 (WDT) .....	338
IPC9 (中断优先级控制寄存器 9) .....	113	PLL .....	137
NVMCON (闪存控制寄存器) .....	69	PWM 模块 .....	196, 197
NVMKEY (非易失性存储器密钥寄存器) .....	70	片上稳压器的连接 .....	337
OCxCON (输出比较 x 控制) .....	193	器件时钟 .....	135, 137
OSCCON (振荡器控制) .....	139	SPI .....	226
OSCTUN (FRC 振荡器调节) .....	143	输出比较 .....	191
P1DC2 (PWM 占空比 2) .....	215	输入捕捉 .....	189
P1DC3 (PWM 占空比 3) .....	215	Timer2/3 (32 位) .....	183
PxDC1 (PWM 占空比 1) .....	215	Timer2 (16 位) .....	181
PLLFB (PLL 反馈频比) .....	142	UART .....	243
PTCON (PWM 时基控制) .....	206	正交编码器接口 .....	217
PWMxCON1 (PWM 控制 1) .....	209	<b>L</b>	
PWMxCON2 (PWM 控制 2) .....	210	灵活的配置 .....	333
PxDTCN1 (死区控制 1) .....	211	漏极开路配置 .....	150
PxDTCN2 (死区控制 2) .....	212	<b>M</b>	
PxFLTA (故障 A 控制) .....	213	MPLAB ASM30 汇编器、链接器和库管理器 .....	352
PxOVD (改写控制) .....	214	MPLAB ICD 2 在线调试器 .....	353
PxSECM (特殊事件比较) .....	208	MPLAB ICE 2000 高性能通用在线仿真器 .....	353
PxTCON (PWM 时基控制) .....	297, 298, 299	MPLAB PM3 器件编程器 .....	353
PxTMR (PWM 定时器计数值) .....	207	MPLAB REAL ICE 在线仿真器系统 .....	353
PxTPER (PWM 时基周期) .....	207	MPLAB 集成开发环境软件 .....	351
QEICON (QEI 控制寄存器) .....	221	MPLINK 目标链接器 / MPLIB 目标库管理器 .....	352
RCON (复位控制) .....	74	模寻址 .....	60
SPIxCON1 (SPIx 控制寄存器 1) .....	230	操作示例 .....	60
SPIxCON2 (SPIx 控制寄存器 2) .....	232	起始地址和结束地址 .....	60
SPIxSTAT (SPIx 状态和控制) .....	229	W 地址寄存器选择 .....	60
SR (CPU 状态) .....	22, 86	应用 .....	61
T1CON (Timer1 控制) .....	188	<b>N</b>	
TCxCON (输入捕捉 x 控制) .....	190	NVM 模块	
TxCON (B 类定时器时基控制) .....	184	寄存器映射 .....	57
TyCON (C 类定时器时基控制) .....	185	内部 RC 振荡器	
UxMODE (UARTx 模式) .....	246	与 WDT 配合使用 .....	338
UxSTA (UARTx 状态和控制寄存器) .....	248	<b>P</b>	
寄存器映射		PICSTART 2 开发编程器 .....	354
并行主 / 从端口 .....	54	PICSTART Plus 开发编程器 .....	354
CRC .....	55	PMD 模块	
实时时钟和日历 .....	55	寄存器映射 .....	57
双比较器 .....	55	PORTA	
JTAG 边界扫描接口 .....	333	寄存器映射 .....	55, 56
JTAG 接口 .....	339	PORTB	
基于指令的节能模式 .....	147	寄存器映射 .....	56
空闲 .....	148	PWM	
休眠 .....	147	边沿对齐 .....	199
交流特性 .....	364	CPU 空闲模式期间的工作 .....	205
负载条件 .....	364	CPU 休眠模式期间的工作 .....	205
内部 RC 精度 .....	366	单脉冲模式 .....	203
节能特性 .....	147	独立输出模式 .....	203
时钟频率和切换 .....	147	互补模式 .....	201
<b>K</b>		互补输出模式 .....	203
开发支持 .....	351	输出改写 .....	203
看门狗超时复位 (WDTR) .....	79		
看门狗定时器 (WDT) .....	333, 338		
编程注意事项 .....	338		

输出改写同步 .....	203	控制寄存器 .....	68
中心对齐 .....	200	RTSP 工作原理 .....	68
周期 .....	199	上电复位 (POR) .....	78
PWM 更新锁定 .....	204	时序规范	
PWM 故障引脚 .....	203	10 位 ADC 转换要求 .....	393
故障状态 .....	204	12 位 ADC 转换要求 .....	391
使能位 .....	204	CAN I/O 要求 .....	387
输入模式 .....	204	电机控制 PWM 要求 .....	375
锁定 .....	204	复位、看门狗定时器、振荡器起振定时器、上电延时	
逐周期 .....	204	定时器和欠压复位要求 .....	369
优先级 .....	204	I2Cx 总线数据要求 (从模式) .....	386
PWM 时基 .....	198	I2Cx 总线数据要求 (主模式) .....	384
单事件模式 .....	198	简单 OC/PWM 模式要求 .....	374
后分频器 .....	199	PLL 时钟 .....	366
连续递增 / 递减计数模式 .....	198	QE1 索引脉冲要求 .....	377
两次更新模式 .....	199	QE1 外部时钟要求 .....	372
预分频器 .....	199	SPIx 从模式 (CKE = 0) 要求 .....	380
自由运行模式 .....	198	SPIx 从模式 (CKE = 1) 要求 .....	382
PWM 输出和极性控制 .....	203	SPIx 主模式 (CKE = 0) 要求 .....	378
输出引脚控制 .....	203	SPIx 主模式 (CKE = 1) 要求 .....	379
PWM 死区发生器 .....	201	输出比较要求 .....	373
范围 .....	202	Timer1 外部时钟要求 .....	370
分配 .....	202	Timer2 外部时钟要求 .....	371
选择位 (表) .....	202	Timer3 外部时钟要求 .....	371
PWM 特殊事件触发器 .....	205	正交解码器要求 .....	376
后分频器 .....	205	时序特性	
PWM 占空比 .....		CLKO 和 I/O .....	367
比较单元 .....	200	时序图	
寄存器缓冲区 .....	200	10 位 ADC 转换 (CHPS = 01, SIMSAM = 0,	
立即更新 .....	201	ASAM = 0, SSRC = 000) .....	392
配置寄存器映射 .....	333	10 位 ADC 转换 (CHPS = 01, SIMSAM = 0,	
配置模拟端口引脚 .....	150	ASAM = 1, SSRC = 111, SAMC = 00001) .....	392
配置位 .....	333	12 位 ADC 转换 (ASAM = 0, SSRC = 000) .....	391
<b>Q</b>		边沿对齐的 PWM .....	200
QE1		电机控制 PWM .....	375
16 位递增 / 递减位置计数器模式 .....	218	电机控制 PWM 故障 .....	375
备用 16 位定时器 / 计数器 .....	219	ECAN I/O .....	387
CPU 空闲模式期间的操作 .....	219	ECAN 位 .....	256
CPU 空闲模式下的定时器操作 .....	220	复位、看门狗定时器、振荡器起振定时器和上电延时	
CPU 休眠模式期间的操作 .....	219	定时器 .....	368
CPU 休眠模式下的定时器操作 .....	219	I2Cx 总线启动位 / 停止位 (从模式) .....	385
错误检查 .....	218	I2Cx 总线启动位 / 停止位 (主模式) .....	383
计数方向状态 .....	218	I2Cx 总线数据 (从模式) .....	385
可编程数字噪声滤波器 .....	219	I2Cx 总线数据 (主模式) .....	383
逻辑 .....	218	QE1/QE2 输入 .....	376
位置测量模式 .....	218	QE1 模块索引脉冲 .....	377
中断 .....	220	欠压情形 .....	79
<b>R</b>		SPIx 从模式 (CKE = 0) .....	380
软件堆栈指针, 帧指针		SPIx 从模式 (CKE = 1) .....	381
CALL 堆栈帧 .....	58	SPIx 主模式 (CKE = 0) .....	378
软件模拟器 (MPLAB SIM) .....	352	SPIx 主模式 (CKE = 1) .....	379
软件 RESET 指令 (SWR) .....	79	输出比较 / PWM .....	374
<b>S</b>		输出比较 (OCx) .....	373
SPI		输入捕捉 (CAPx) .....	373
从 - 帧从模式连接 .....	228	死区 .....	201
从 - 帧主模式连接 .....	228	TimerQ (QE1 模块) 外部时钟 .....	372
主 / 从连接 .....	227	Timer1、Timer2 和 Timer3 外部时钟 .....	370
主 - 帧主模式连接 .....	227	外部时钟 .....	365
SPI 模块		中心对齐的 PWM .....	200
SPI1 寄存器映射 .....	45	时序要求	
闪存程序存储器 .....	67	CLKO 和 I/O .....	367
编程算法 .....	71	输入捕捉 .....	373
表指令 .....	67	外部时钟 .....	365
操作 .....	68	使用 RCON 状态位 .....	80
		时钟切换 .....	145
		过程 .....	145
		使能 .....	145

输出比较.....	191	指令集	
数据地址空间 .....	33	概述 .....	346
带 16 KB RAM 的 dsPIC33FJ128MC802/804		汇总 .....	343
和 dsPIC33FJ64MC802/804 器件的存储器映射	36	指令寻址模式 .....	58
带 4 KB RAM 的 dsPIC33FJ32MC302/304 器件的		MAC 指令 .....	59
存储器映射 .....	34	MCU 指令 .....	58
带 8 KB RAM 的 dsPIC33FJ128MC202/204		其他指令 .....	59
和 dsPIC33FJ64MC202/204 器件的存储器映射	35	文件寄存器指令 .....	58
对齐 .....	33	支持的基本模式 .....	59
宽度 .....	33	传送指令和累加器指令 .....	59
Near 数据空间 .....	33	直流特性 .....	356
软件堆栈 .....	58	程序存储器 .....	363
数据累加器和加法器 / 减法器 .....	28	打盹电流 (IDOZE) .....	360
回写 .....	29	掉电电流 (IPD) .....	360
舍入逻辑 .....	29	工作电流 (IDD) .....	358
数据空间写饱和 .....	30	I/O 引脚输出规范 .....	362
溢出和饱和 .....	28	I/O 引脚输入规范 .....	361
输入捕捉 .....	189	空闲电流 (IDLE) .....	359
寄存器 .....	190	温度和电压规范 .....	357
输入电平变化通知 .....	150	中断控制和状态寄存器 .....	85
算术逻辑单元 (ALU) .....	26	IECx .....	85
<b>T</b>		IFSx .....	85
Timer1 .....	187	INTCON1 .....	85
Timer2/3 .....	181	INTCON2 .....	85
桶形移位寄存器 .....	30	IPCx .....	85
通用异步收发器 (UART) .....	243	中断设置过程 .....	122
<b>U</b>		初始化 .....	122
UART		陷阱服务程序 .....	122
8 位或 9 位数据模式下的接收 .....	245	中断服务程序 .....	122
8 位数据模式下的发送 .....	245	中断禁止 .....	122
9 位数据模式下的发送 .....	245	中断向量表 (IVT) .....	81
波特率		自动时钟延长 .....	235
发生器 (BRG) .....	244	发送模式 .....	235
间隔和同步发送序列 .....	245	接收模式 .....	235
使用 UxCTS 和 UxRTS 引脚的流控制 .....	245		
UART 模块			
UART1 寄存器映射 .....	44		
<b>W</b>			
WWW, 在线支持 .....	11		
外设模块禁止 (PMD) .....	148		
位反转寻址 .....	61		
示例 .....	61		
实现 .....	61		
序列列表 (16 项) .....	62		
温度和电压规范			
AC .....	364		
稳压器 (片上) .....	337		
<b>X</b>			
系统控制			
寄存器映射 .....	56, 57		
<b>Y</b>			
引脚输入 / 输出说明 (表) .....	15		
<b>Z</b>			
在节能指令执行期间的中断 .....	148		
在线串行编程 (ICSP) .....	333, 339		
在线调试器 .....	339		
在线仿真 .....	333		
增强型 CAN 模块 .....	251		
正交编码器接口 (QE1) .....	217		
正交编码器接口 (QE1) 模块			
寄存器映射 .....	43		

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。只要使用常用的因特网浏览器即可访问。网站提供以下信息:

- **产品支持**——数据手册和勘误表、应用笔记和样本程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及存档软件
- **一般技术支持**——常见问题 (FAQ)、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

## 变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时, 收到电子邮件通知。

欲注册, 请登录 Microchip 网站 [www.microchip.com](http://www.microchip.com), 点击“变更通知客户 (Customer Change Notification)”服务后按照注册说明完成注册。

## 客户支持

Microchip 产品的用户可通过以下渠道获得帮助:

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

客户应联系其代理商、代表或应用工程师 (FAE) 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过<http://support.microchip.com>获得网上技术支持。

## 读者反馈表

我们努力为您提供最佳文档，以确保您能够成功使用 Microchip 产品。如果您对文档的组织、条理性、主题及其他有助于提高文档质量的方面有任何意见或建议，请填写本反馈表并传真给我公司 TRC 经理，传真号码为 86-21-5407-5066。请填写以下信息，并从下面各方面提出您对本文档的意见。

致: TRC 经理 总页数 \_\_\_\_\_  
关于: 读者反馈  
发自: 姓名 \_\_\_\_\_  
公司 \_\_\_\_\_  
地址 \_\_\_\_\_  
国家 / 省份 / 城市 / 邮编 \_\_\_\_\_  
电话 (\_\_\_\_\_) \_\_\_\_\_ 传真 (\_\_\_\_\_) \_\_\_\_\_

应用 (选填):

您希望收到回复吗? 是\_\_\_\_ 否\_\_\_\_

器件: dsPIC33FJ32MC302/304、dsPIC33FJ64MCX02/X04 文献编号: DS70291A\_CN  
和 dsPIC33FJ128MCX02/X04

问题

1. 本文档中哪些部分最有特色?

---

---

2. 本文档是否满足了您的软硬件开发要求? 如何满足的?

---

---

3. 您认为本文档的组织结构便于理解吗? 如果不便于理解, 那么问题何在?

---

---

4. 您认为本文档应该添加哪些内容以改善其结构和主题?

---

---

5. 您认为本文档中可以删减哪些内容, 而又不会影响整体使用效果?

---

---

6. 本文档中是否存在错误或误导信息? 如果存在, 请指出是什么信息及其具体页数。

---

---

7. 您认为本文档还有哪些方面有待改进?

---

---



产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

**dsPIC 33 FJ 32 MC3 02 T E / SP - XXX**

**Microchip 的商标**

**架构**

**闪存系列**

**程序存储器容量 (KB)**

**产品组**

**引脚数**

**卷带标志 (如果适用)**

**温度范围**

**封装**

**模式**

**示例:**

a) dsPIC33FJ32MC302-E/SP:  
电机控制 dsPIC33, 32 KB 程序存储器, 28 引脚, 扩展级温度范围, SPDIP 封装。

架构:	33	=	16 位数字信号控制器
闪存系列:	FJ	=	闪存程序存储器, 3.3V
产品组:	MC2	=	电机控制系列
	MC3	=	电机控制系列
	MC8	=	电机控制系列
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:	SP	=	窄型塑封双列直插式封装——300 mil 主体 (SPDIP)
	SO	=	塑封小外形封装——宽条, 7.5 mm 主体 (SOIC)
	ML	=	塑封正方无脚封装——8x8 mm 主体 (QFN)
	MM	=	塑封正方无脚封装——6x6x0.9 mm 主体 (QFN-S)
	PT	=	塑封薄型正方扁平封装——10x10x1 mm 主体 (TQFP)

深圳市粤原点科技有限公司 ( 丽智电子

:www.LZmcu.com )

(Microchip Authorized Design Partner)指定授权

总部地址：深圳市福田区福虹路世贸广场C座1103室

Add：Room 1103,Block C,World Trade Plaza,

Fuhong Road,Futian District Shen Zhen City

电话(tel)：86-755-83666320,83685176,83666321,83666325

传真(fax)：86-755-83666329,83681854

Web: [Http://www.origin-gd.com](http://www.origin-gd.com) or

[Http://www.LZmcu.com](http://www.LZmcu.com)

E-mail：01@LZmcu.com

联系人：马先生,王小姐

公司在线咨询：QQ:46885145

MSN:MSN:action\_tech@hotmail.com

7x24小时在线产品咨询: 13509674380 13798484366