



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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更安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

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

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

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

商标

Microchip 的名称和徽标组合、Microchip 徽标、Accuron、dsPIC、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、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、PIC³² 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[®] MCU 与 dsPIC[®] DSC、KEELOQ[®] 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品方面的质量体系流程均符合 ISO/TS-16949:2002。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

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)

高性能数字信号控制器 (Digital Signal Controller, 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 位数据左移或右移最多 16 位

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 30 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
- 数字输入引脚可耐受 5V 的电压 (RB5 除外)
- 所有 PWM 引脚的拉 / 灌电流为 16 mA

片上闪存和 SRAM:

- 闪存程序存储器 (最大 16 KB)
- 数据 SRAM (最大 2 KB)
- 闪存程序存储器的引导和通用安全性

外设特性:

- 定时器 / 计数器, 最多 3 个 16 位定时器
 - 最多可以配对成 1 个 32 位定时器
- 输入捕捉 (最多 2 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 级深度的 FIFO 缓冲区
- 输出比较 (最多 2 路通道):
 - 单次或两次 16 位比较模式
 - 16 位无毛刺 PWM 模式
- 4 线 SPI:
 - 帧支持与简单编解码器的 I/O 接口
 - 1 级深度 FIFO 缓冲区
 - 支持 8 位和 16 位数据
 - 支持所有串行时钟格式和采样模式
- I²C™:
 - 支持完全的多主机从模式
 - 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:
 - 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® 编码和解码
 - 高速波特率模式
 - 使用 CTS 和 RTS 的硬件流控制

中断控制器:

- 中断响应延时为 5 个周期
- 118 个中断向量
- 最多 35 个中断源
- 最多 3 个外部中断
- 7 个可编程优先级
- 4 个处理器异常

高速 PWM 模块特性:

- 最多支持 4 个 PWM 发生器, 能提供 4-8 路输出
- 8 路 PWM 输出的每一路都具有独立的时基和占空比
- 上升沿和下降沿死区
- 占空比分辨率为 1.04 ns
- 死区分辨率为 1.04 ns
- 相移分辨率为 1.04 ns
- 频率分辨率为 1.04 ns
- 支持的 PWM 模式:
 - 标准边沿对齐
 - 真正独立输出
 - 互补
 - 中心对齐
 - 推挽
 - 多相
 - 可变相位
 - 固定关断时间
 - 电流复位
 - 限流
- 用于 8 路 PWM 输出的独立故障 / 限流输入
- 输出改写控制
- 特殊事件触发器
- PWM 捕捉功能
- 输入时钟的预分频器
- 由 PWM 模块送至 ADC (analog-to-digital converter, ADC) 的双路触发信号
- PWMxL 和 PWMxH 输出引脚交换
- PWM4H 和 PWM4L 引脚可重映射
- 可在运行过程中更改 PWM 频率、占空比和相移
- 各独立 PWM 发生器的禁止
- 前沿消隐 (Leading-Edge Blanking, LEB) 功能

高速模拟比较器

- 最多 4 个模拟比较器:
 - 20 ns 响应时间
 - 与每个模拟比较器相关的 10 位 DAC
 - 提供 DAC 输出的 DACOUT 引脚
 - 可编程的输出极性
 - 可选择输入信号源
 - ADC 采样和转换功能
- 与 PWM 模块的接口:
 - PWM 占空比控制
 - PWM 周期控制
 - PWM 故障检测

高速 10 位 ADC

- 10 位分辨率
- 最多 12 路输入通道可组成 6 个转换对
- 两个内部参考电压监控输入组成一对内部模拟输入
- 用于并行转换模拟输入对的逐次逼近型 (Successive Approximation Register, SAR) 转换器:
 - 对于带两个 SAR 的器件, 转换速率为 4 Msps
 - 对于带一个 SAR 的器件, 转换速率为 2 Msps
- 用于每路模拟通道的专用结果缓冲区
- 用于每个模拟输入转换对的独立触发源

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

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

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
 - 带 120 MHz 压控振荡器 (Voltage Controlled Oscillator, VCO) 的锁相环 (Phase-Locked Loop, PLL)
 - 3 MHz 至 40 MHz 范围内的主晶振
 - 频率为 32 kHz 的内部低功耗 RC (Low-Power RC, LPRC) 振荡器
 - 频率为 7.37 MHz 的内部快速 RC (Fast RC, FRC) 振荡器
- 上电复位 (Power-on Reset, POR)
- 欠压复位 (Brown-out Reset, BOR)
- 上电延时定时器 (Power-up Timer, PWRT)
- 振荡器起振定时器 (Oscillator Start-up Timer, OST)
- 自带 RC 振荡器的看门狗定时器
- 故障保护时钟监视器 (FSCM)
- 多个复位源
- 在线串行编程 (In-Circuit Serial Programming™, ICSP™)

封装:

- 18 引脚 SOIC
- 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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 控制器系列

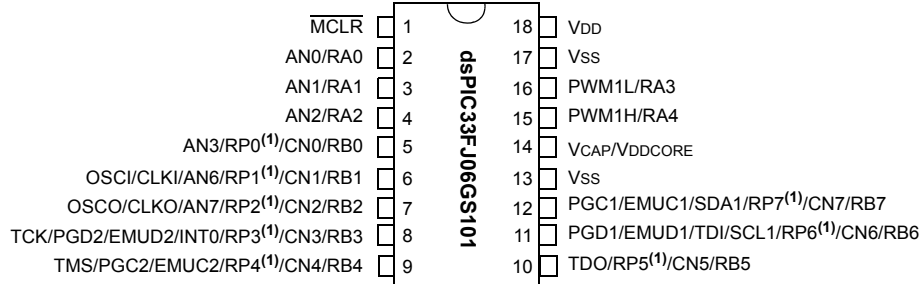
器件	引脚	闪存程序存储器 (KB)	RAM (字节)	可重映射的外设									DAC 输出	I ² C™	ADC			I/O 引脚	封装
				可重映射的引脚数	16 位定时器	输入捕捉	输出比较	UART	SPI	PWM ⁽²⁾	模拟比较器	外部中断 ⁽³⁾			SAR	采样/保持 (S&H) 电路	模数转换输入		
dsPIC33FJ06GS101	18	6	256	8	2	0	1	1	1	2x2 ⁽¹⁾	0	3	0	1	1	3	6	13	SOIC
dsPIC33FJ06GS102	28	6	256	16	2	0	1	1	1	2x2	0	3	0	1	1	3	6	21	SPDIP SOIC QFN-S
dsPIC33FJ06GS202	28	6	1K	16	2	1	1	1	1	2x2	2	3	1	1	1	3	6	21	SPDIP SOIC QFN-S
dsPIC33FJ16GS402	28	16	2K	16	3	2	2	1	1	3x2	0	3	0	1	1	4	8	21	SPDIP SOIC QFN-S
dsPIC33FJ16GS404	44	16	2K	30	3	2	2	1	1	3x2	0	3	0	1	1	4	8	35	QFN TQFP
dsPIC33FJ16GS502	28	16	2K	16	3	2	2	1	1	4x2 ⁽¹⁾	4	3	1	1	2	6	8	21	SPDIP SOIC QFN-S
dsPIC33FJ16GS504	44	16	2K	30	3	2	2	1	1	4x2 ⁽¹⁾	4	3	1	1	2	6	12	35	QFN TQFP

- 注
- 1: PWM4H:PWM4L 引脚是可重映射的。
 - 2: PWM 故障引脚和 PWM 同步引脚是可重映射的。
 - 3: 3 个中断中只有 2 个是可重映射的。

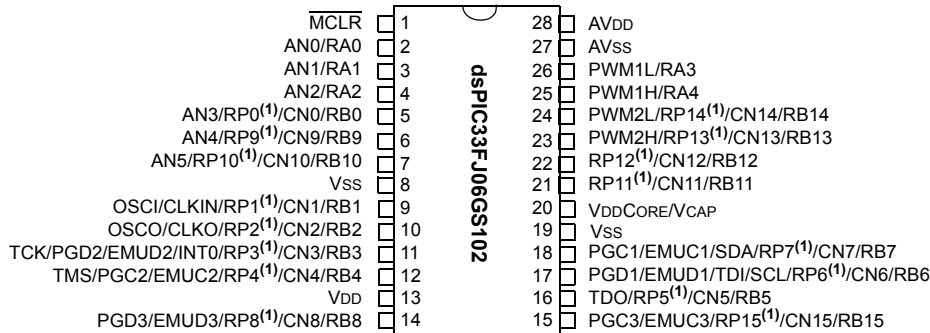
dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

引脚图

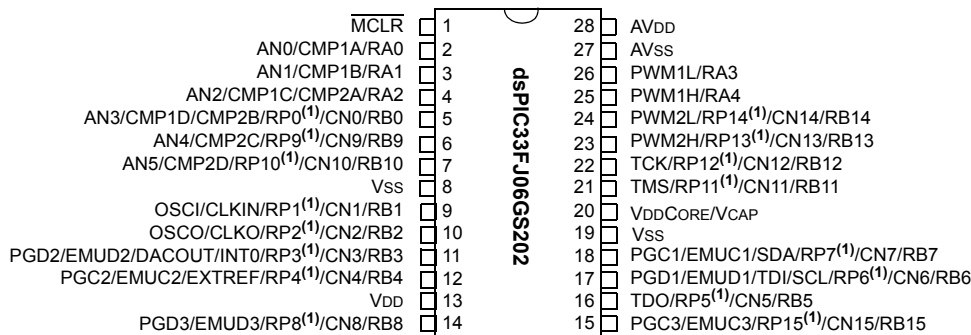
18 引脚 SOIC



28 引脚 SOIC 和 SPDIP



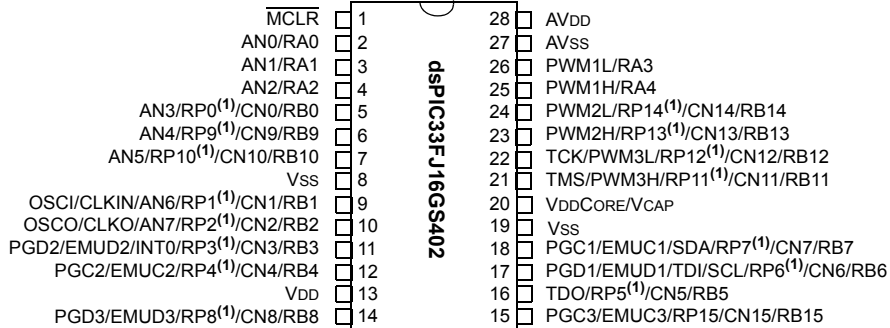
28 引脚 SPDIP 和 SOIC



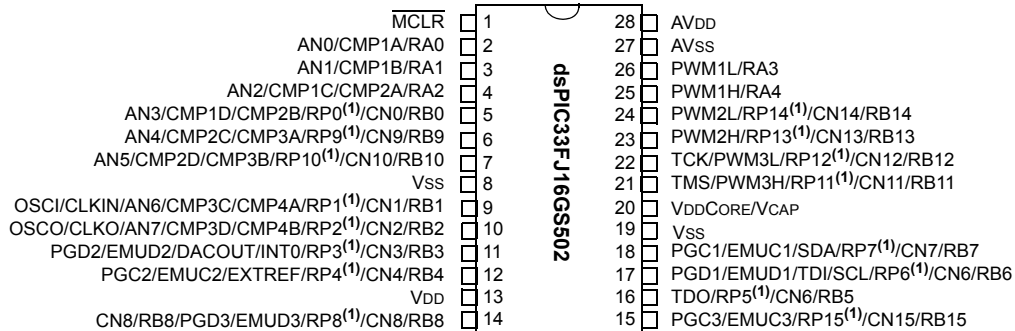
注 1: RPn 引脚可由任何可重映射的外设使用。可用外设的列表, 请参见 “dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 控制器系列” 表。

引脚图（续）

28 引脚 SPDIP 和 SOIC



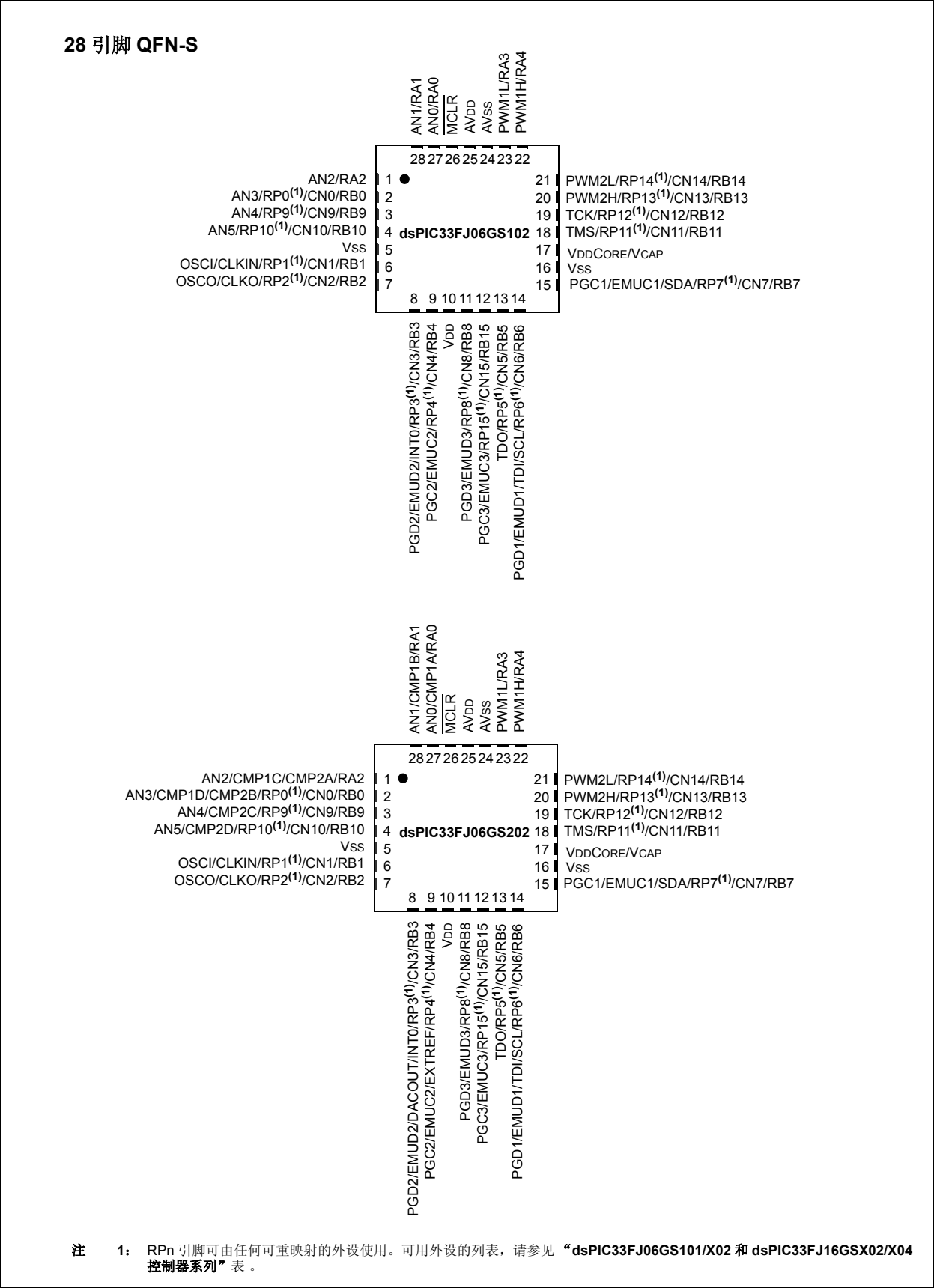
28 引脚 SPDIP 和 SOIC



注 1: RPn 引脚可由任何可重映射的外设使用。可用外设的列表, 请参见 “dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 控制器系列” 表。

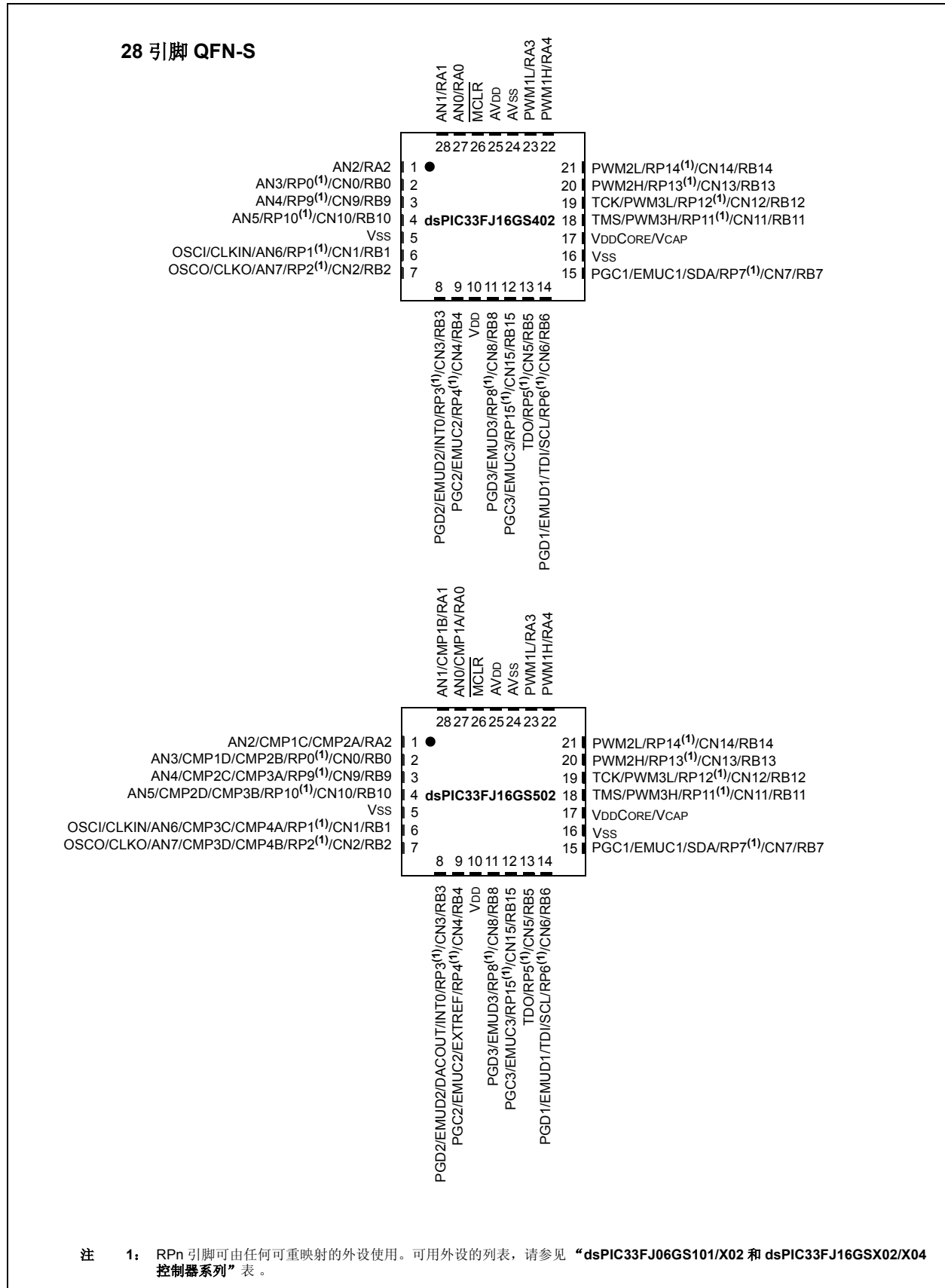
dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

引脚图（续）



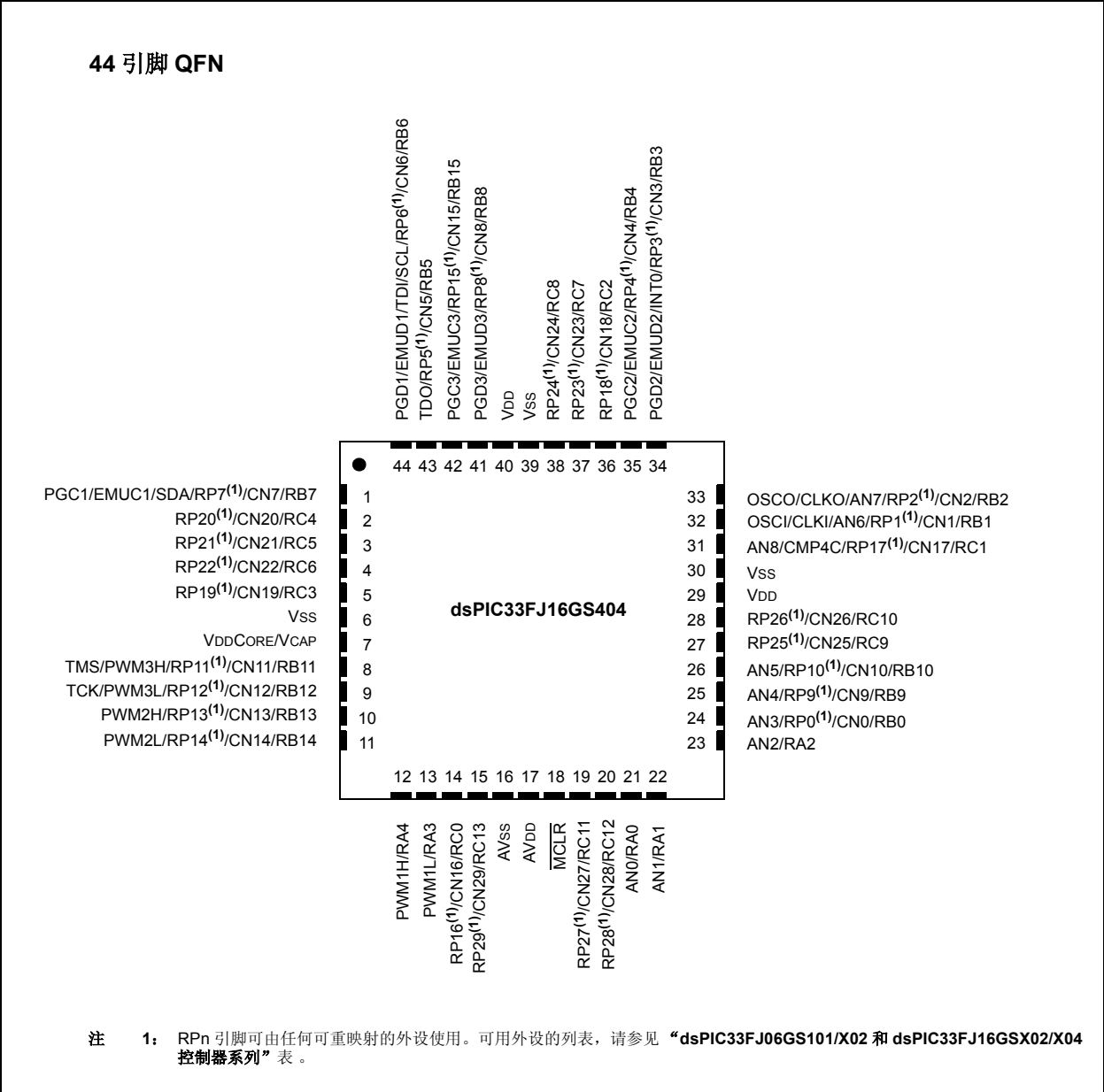
dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

引脚图（续）

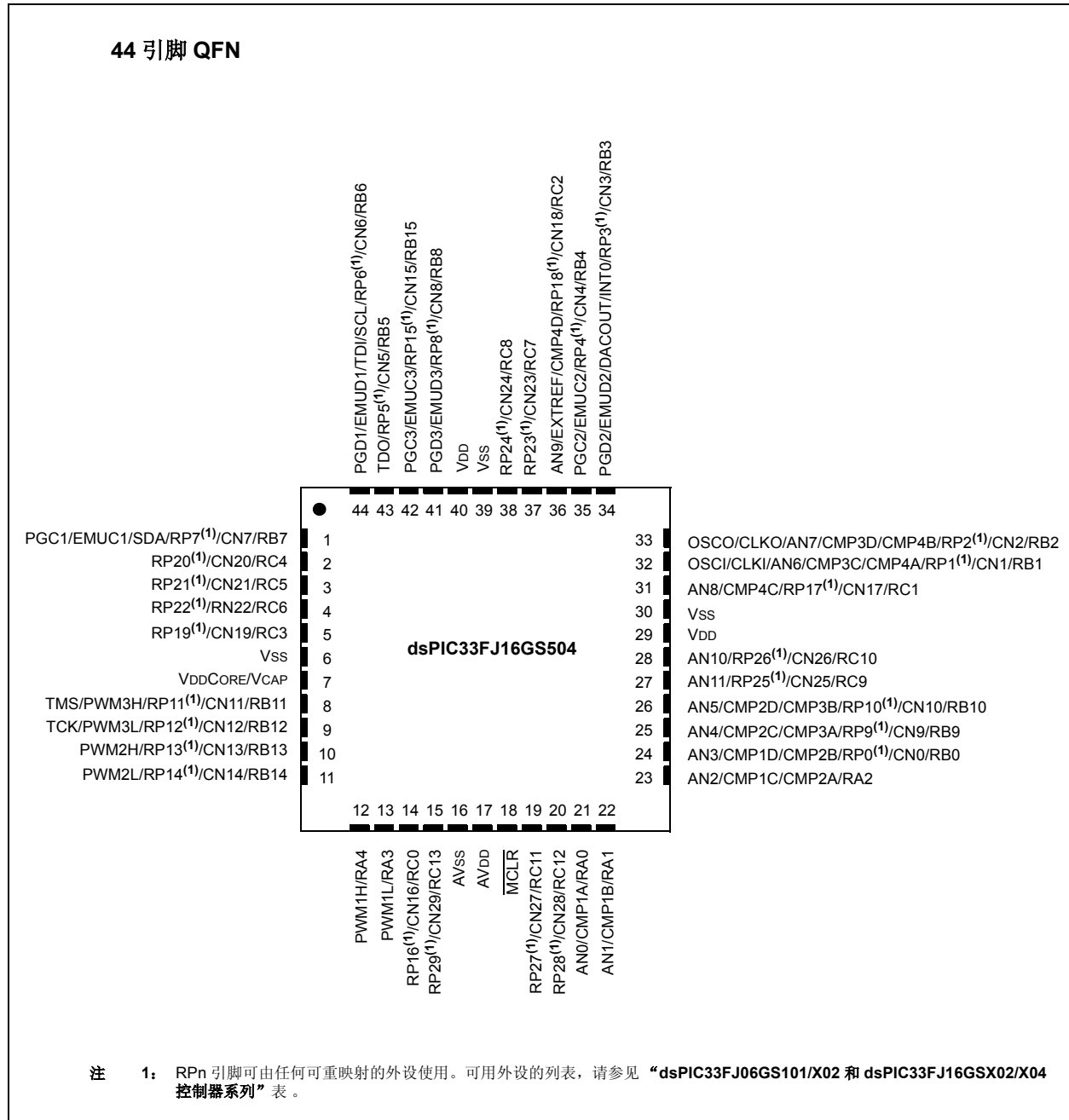


dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

引脚图（续）

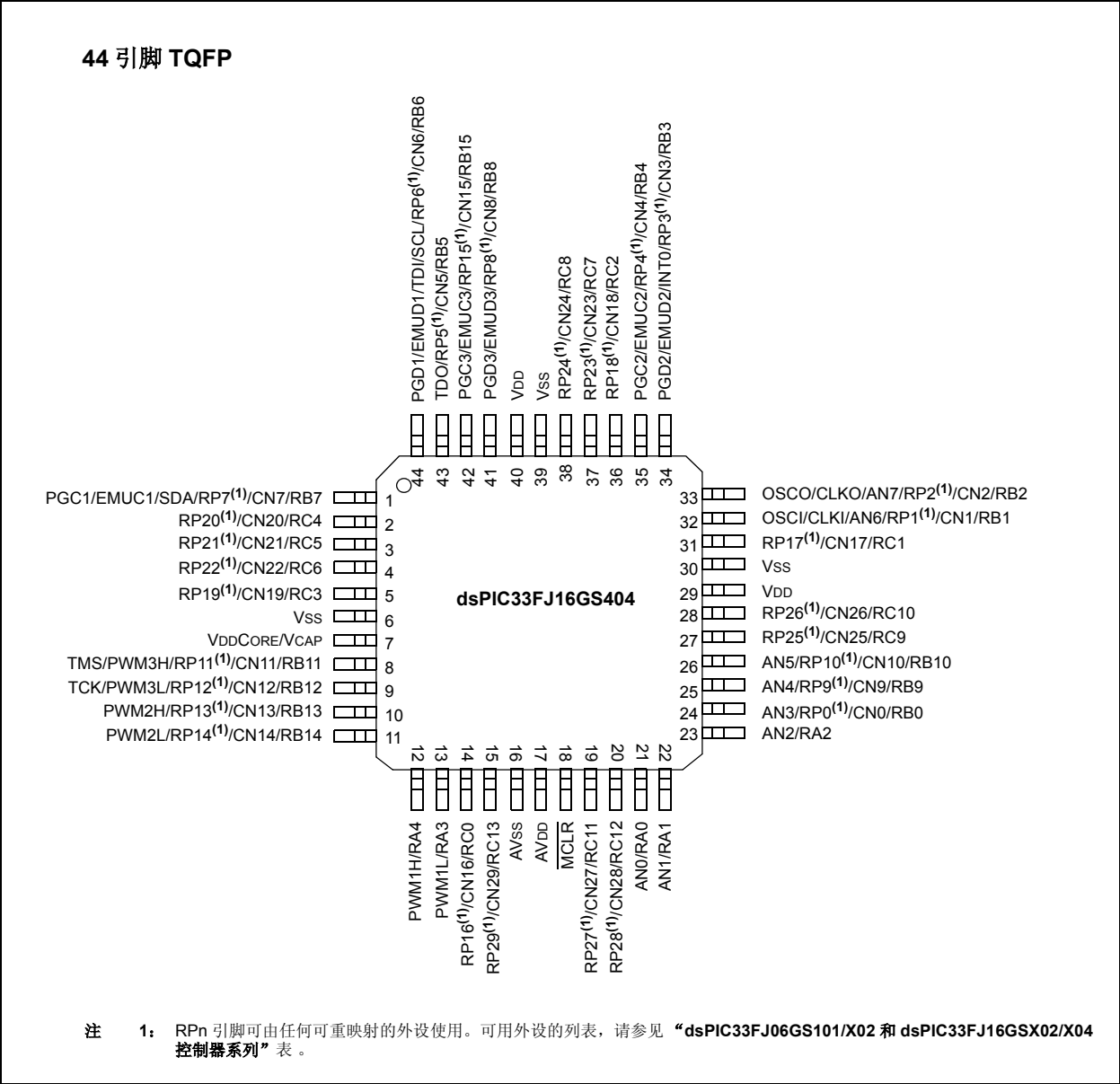


引脚图（续）

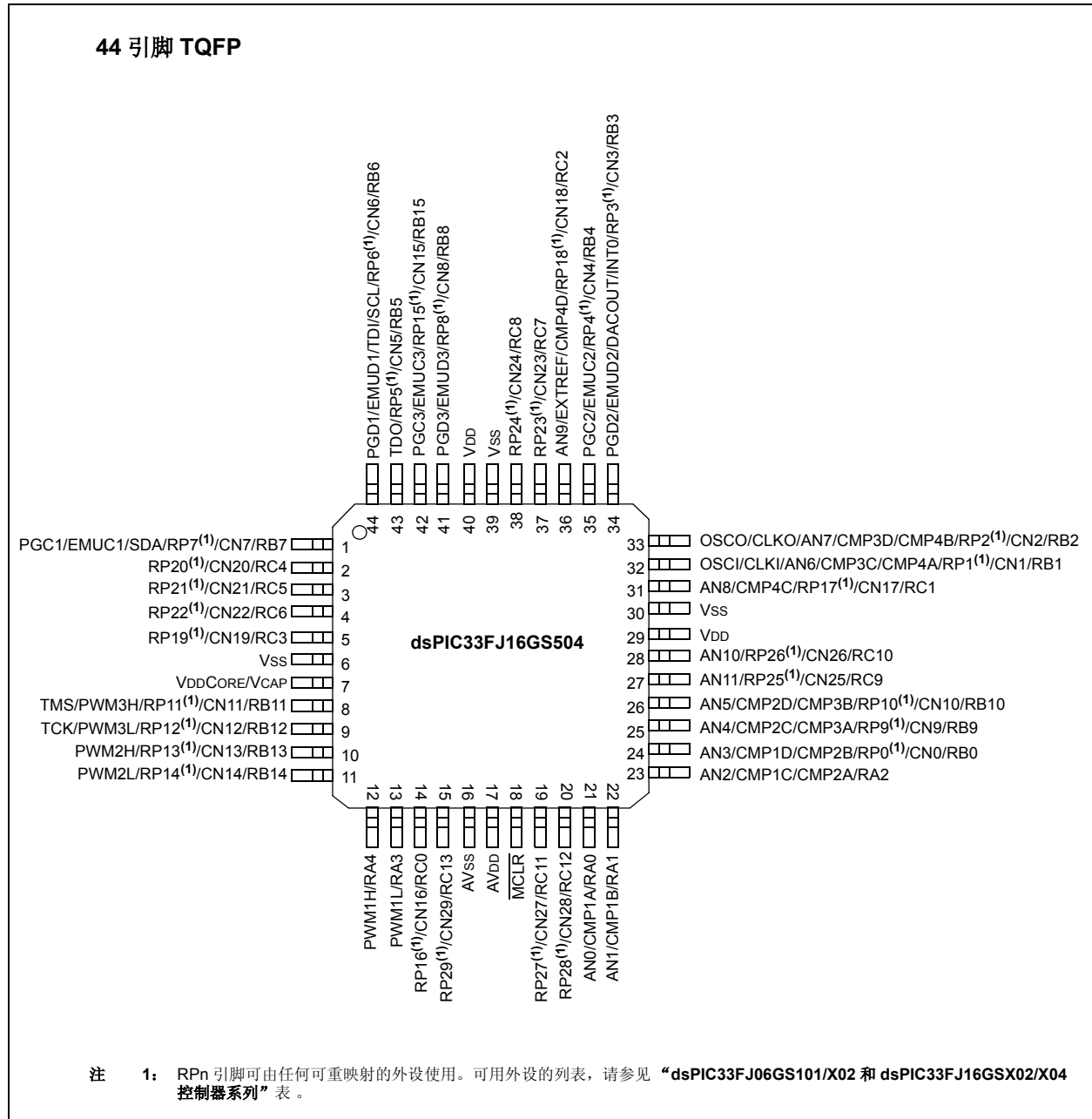


dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

引脚图（续）



引脚图（续）



目录

1.0	器件概述	13
2.0	CPU	17
3.0	存储器构成	29
4.0	闪存程序存储器	71
5.0	复位	77
6.0	中断控制器	85
7.0	振荡器配置	125
8.0	节能特性	135
9.0	I/O 端口	143
10.0	Timer1	173
11.0	Timer2/3 特性	175
12.0	输入捕捉	181
13.0	输出比较	183
14.0	高速 PWM	187
15.0	串行外设接口 (SPI)	207
16.0	I ² C™	213
17.0	通用异步收发器 (UART)	221
18.0	高速 10 位模数转换器 (ADC)	227
19.0	高速模拟比较器	249
20.0	特殊功能	253
21.0	指令集汇总	261
22.0	开发支持	269
23.0	电气特性	273
24.0	封装信息	307
附录 A:	版本历史	319
Microchip 网站		327
变更通知客户服务		327
客户支持		327
读者反馈表		328
产品标识体系		329

致客户

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

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

最新数据手册

欲获得本数据手册的最新版本，请查询我公司的网站：

<http://www.microchip.com>

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

勘误表

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

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

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

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

客户通知系统

欲及时获知 Microchip 产品的最新信息，请到我公司网站 www.microchip.com 上注册。

1.0 器件概述

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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*”。请访问 Microchip 网站（www.microchip.com）了解最新的 “*dsPIC33F Family Reference Manual*” 章节。

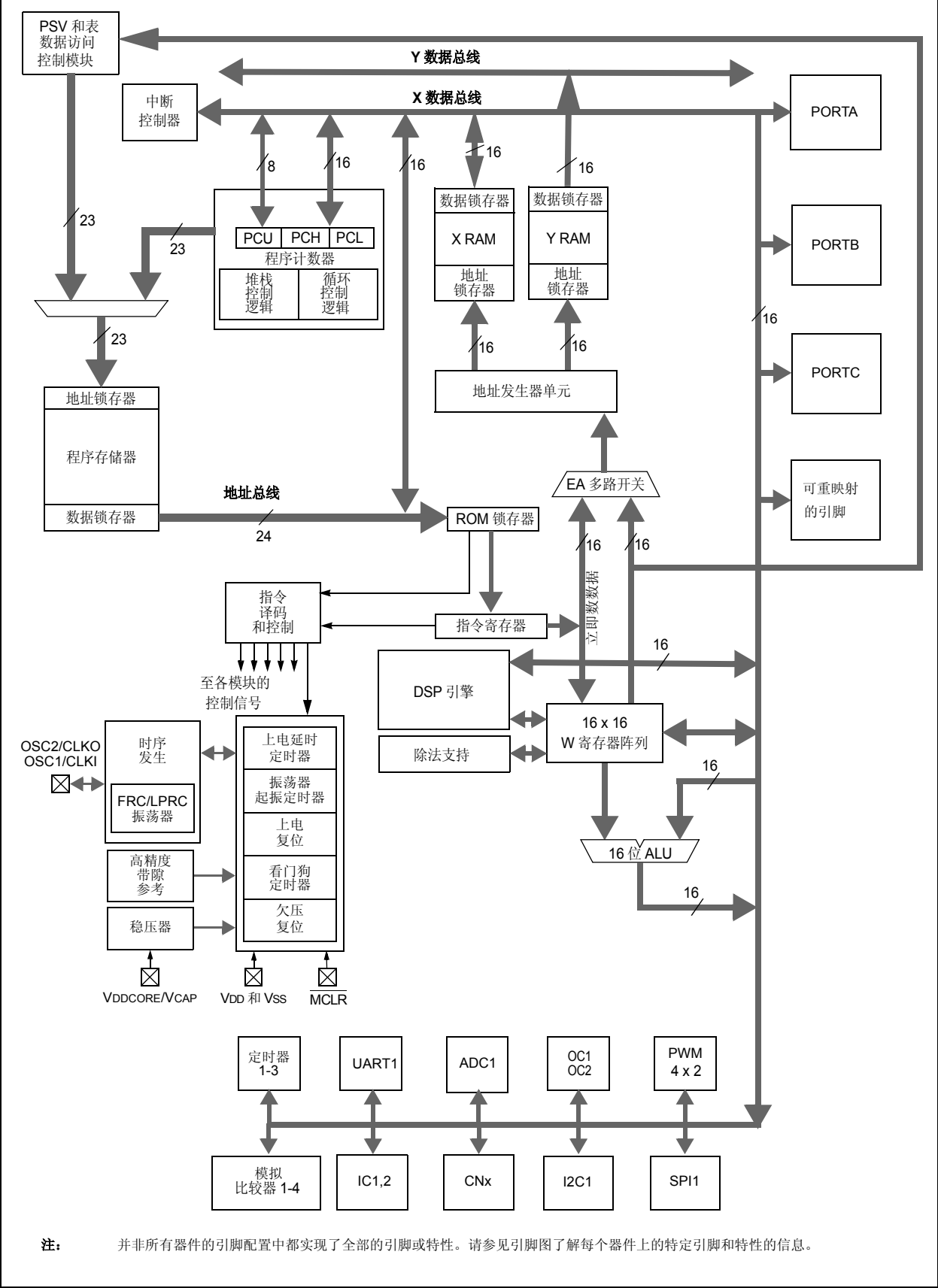
本文档包含如下 dsPIC33F 数字信号控制器（DSC）器件的特定信息：

- dsPIC33FJ06GS101
- dsPIC33FJ06GS102
- dsPIC33FJ06GS202
- dsPIC33FJ16GS402
- dsPIC33FJ16GS404
- dsPIC33FJ16GS502
- dsPIC33FJ16GS504

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件在高性能 16 位单片机（microcontroller，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 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 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: dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 的框图



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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-AN11	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。
CN0-CN29	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	I	ST	捕捉输入 1/2
OCFA OC1-OC2	I O	ST —	比较故障 A 输入（用于比较通道 1 和 2） 比较输出 1 至 2
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	I/O	ST	PORTA 是双向 I/O 端口
RB0-RB15	I/O	ST	PORTB 是双向 I/O 端口
RC0-RC13	I/O	ST	PORTC 是双向 I/O 端口
RP0-RP29	I/O	ST	可重映射的 I/O 引脚
T1CK T2CK T3CK	I I I	ST ST ST	Timer1 外部时钟输入 Timer2 外部时钟输入 Timer3 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
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
SCL1 SDA1	I/O I/O	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
TMS TCK TDI TDO	I I I O	TTL TTL TTL —	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

图注: CMOS = CMOS 兼容输入或输出
ST = CMOS 电平的施密特触发器输入
TTL = 晶体管 - 晶体管逻辑

Analog = 模拟输入
P = 电源

I = 输入
O = 输出

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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
CMP1A	I	Analog	比较器 1 的通道 A
CMP1B	I	Analog	比较器 1 的通道 B
CMP1C	I	Analog	比较器 1 的通道 C
CMP1D	I	Analog	比较器 1 的通道 D
CMP2A	I	Analog	比较器 2 的通道 A
CMP2B	I	Analog	比较器 2 的通道 B
CMP2C	I	Analog	比较器 2 的通道 C
CMP2D	I	Analog	比较器 2 的通道 D
CMP3A	I	Analog	比较器 3 的通道 A
CMP3B	I	Analog	比较器 3 的通道 B
CMP3C	I	Analog	比较器 3 的通道 C
CMP3D	I	Analog	比较器 3 的通道 D
CMP4A	I	Analog	比较器 4 的通道 A
CMP4B	I	Analog	比较器 4 的通道 B
CMP4C	I	Analog	比较器 4 的通道 C
CMP4D	I	Analog	比较器 4 的通道 D
DACOUT	0	—	DAC 输出电压
EXTREF	I	Analog	用于参考 DAC 的外部参考电压输入
REFCLK	0	—	REFCLK 输出信号是对系统时钟后分频得到的
FLT1-FLT8	I	ST	PWM 模块的故障输入
SYNCl1-SYNCl2	I	ST	PWM 主时基的外部同步信号
SYNCO1	O	—	用于外部器件同步的 PWM 主时基输出
PWM1L	O	—	PWM1 低端输出
PWM1H	O	—	PWM1 高端输出
PWM2L	O	—	PWM2 低端输出
PWM2H	O	—	PWM2 高端输出
PWM3L	O	—	PWM3 低端输出
PWM3H	O	—	PWM3 高端输出
PWM4L	O	—	PWM4 低端输出
PWM4H	O	—	PWM4 高端输出
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。此引脚必须始终连接。
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

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

2.0 CPU

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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，其中包括对 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。实际的程序存储器容量因器件而异。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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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，dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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，dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 的编程模型如图 2-2 所示。

2.1 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。

2.2 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。此功能要求 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。

2.3 MCU 的特性

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 $(-1.0) \times (-1.0)$ 这样的特殊运算也可以得到准确结果。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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: dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 的 CPU 内核框图

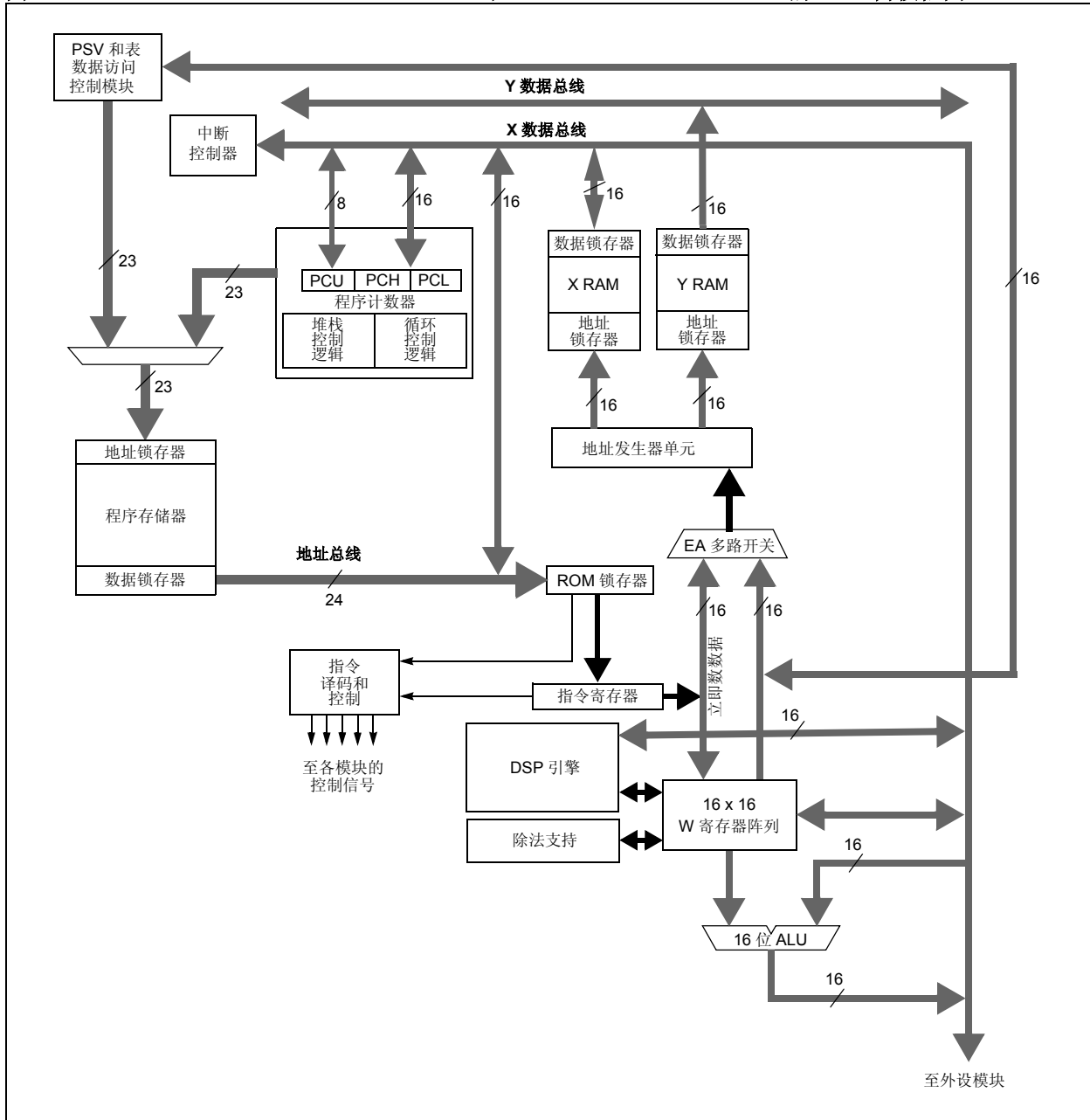
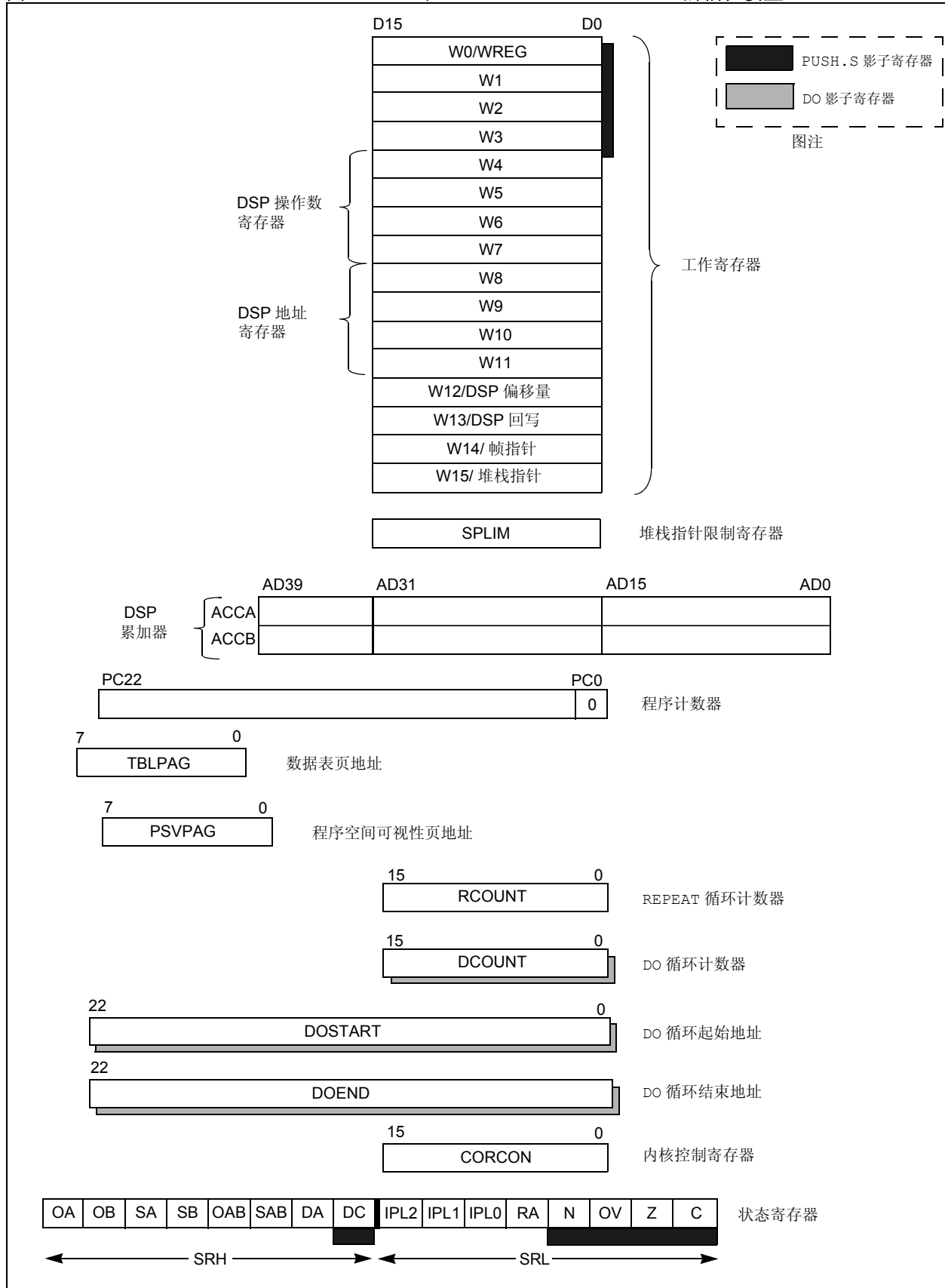


图 2-2: dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 的编程模型



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

2.4 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 ⁽¹⁾	SB ⁽¹⁾	OAB	SAB ^(1,4)	DA	DC
bit 15							bit 8

R/W-0 ⁽²⁾	R/W-0 ⁽³⁾	R/W-0 ⁽³⁾	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL<2:0> ⁽²⁾			RA	N	OV	Z	C
bit 7							bit 0

图注:

C = 可清零位	R = 可读位	U = 未实现位, 读为 0
S = 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 ⁽¹⁾ 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 ⁽¹⁾ 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 ^(1,4) 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 循环有效位 1 = 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 (对于字节长度的数据) 或第 8 个低位 (对于字长度的数据) 发生了进位 0 = 结果的第 4 个低位 (对于字节长度的数据) 或第 8 个低位 (对于字长度的数据) 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 (Interrupt Priority Level, IPL)。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: 清零此位将同时清零 SA 和 SB。

寄存器 2-1: SR: CPU 状态寄存器 (续)

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)

bit 4 **RA**: REPEAT 循环有效位

1 = 在进行 REPEAT 循环
0 = 不在进行 REPEAT 循环

bit 3 **N**: MCU ALU 负标志位

1 = 结果为负
0 = 结果为非负 (零或正值)

bit 2 **OV**: 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 (本次运算) 中发生溢出
0 = 未发生溢出

bit 1 **Z**: 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 **C**: 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 (Interrupt Priority Level, IPL)。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: 清零此位将同时清零 SA 和 SB。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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 ⁽¹⁾	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 ⁽²⁾	PSV	RND	IF
bit 7							bit 0

图注:	C = 可清零位		
R = 可读位	W = 可写位	-n = POR 时的值	1 = 置 1
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 ⁽¹⁾ 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 . . . 001 = 正在进行 1 层 DO 循环 000 = 未进行 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 ⁽²⁾ 1 = CPU 中断优先级大于 7 0 = CPU 中断优先级等于或小于 7
bit 2	PSV: 数据空间中程序空间可视性使能位 1 = 程序空间在数据空间中可视 0 = 程序空间在数据空间中不可视
bit 1	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: 此位将总是读为 0。
2: IPL3 位与 IPL<2:0> 位 (SR<7:5>) 组合形成 CPU 中断优先级。

2.5 算术逻辑单元 (ALU)

dsPIC33FJ06GS101/X02和dsPIC33FJ16GSX02/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)。

dsPIC33FJ06GS101/X02和dsPIC33FJ16GSX02/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。

2.5.1 乘法器

通过使用DSP引擎的高速17位x17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x16位有符号
- 16位x16位无符号
- 16位有符号x5位(立即数)无符号
- 16位无符号x16位无符号
- 16位无符号x5位(立即数)无符号
- 16位无符号x16位有符号
- 8位无符号x8位无符号

2.5.2 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:

- 32位有符号/16位有符号除法
- 32位无符号/16位无符号除法
- 16位有符号/16位有符号除法
- 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。

2.6 DSP引擎

DSP引擎由一个高速17位x17位乘法器、一个桶形移位寄存器和一个40位加法器/减法器(带两个目标累加器、舍入逻辑和饱和逻辑)组成。

dsPIC33FJ06GS101/X02和dsPIC33FJ16GSX02/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:

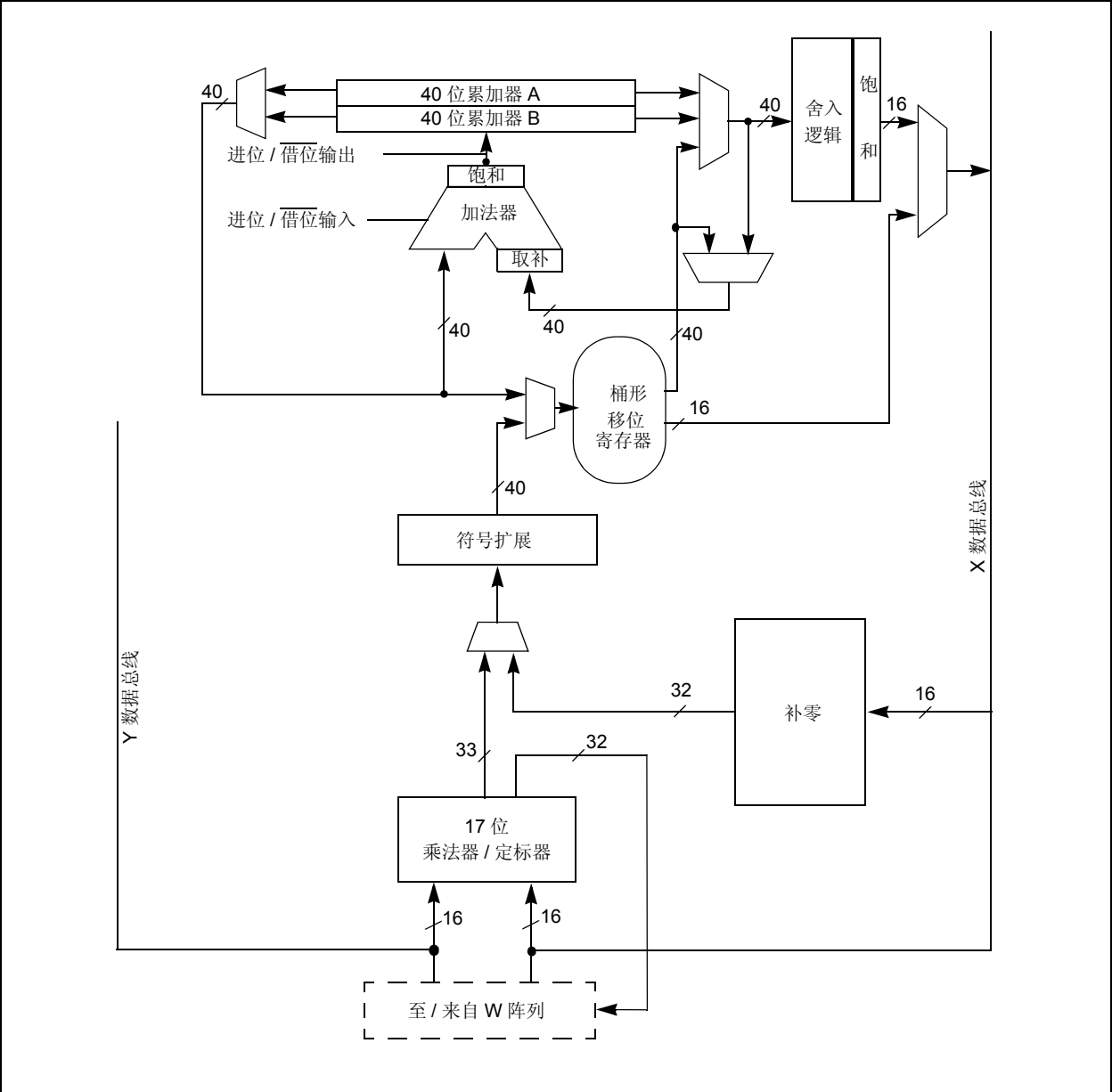
- 小数或整数DSP乘法(IF)
- 有符号或无符号DSP乘法(US)
- 常规或收敛舍入(RND)
- ACCA自动饱和和使能/禁止(SATA)
- ACCB自动饱和和使能/禁止(SATB)
- 对于写数据存储器,自动饱和和使能/禁止(SATDW)
- 累加器饱和模式选择(ACCSAT)

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	$A = A + (x * y)$	有
MAC	$A = A + x2$	无
MOVSAC	A 中内容将不发生改变	有
MPY	$A = x * y$	无
MPY	$A = x2$	无
MPY.N	$A = -x * y$	无
MSC	$A = A - x * y$	有

图 2-3: DSP 引擎框图



2.6.1 乘法器

17 位 x 17 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 x 17 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 3.01518×10^{-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 4.65661×10^{-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运算。

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。

2.6.2 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。

2.6.2.1 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 且 INTCON1 寄存器中相应的溢出陷阱允许位（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。

2.6.3 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）。

2.6.3.1 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）进行检查：

- 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 2.6.3.2 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。

2.6.3.2 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模块接受来自舍入逻辑模块的一个 16 位 1.15 小数值以及来自源（累加器）和 16 位舍入加法器的溢出状态作为输入。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。

2.6.4 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。

注:

3.0 存储器构成

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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”（DS70202），该文档可从 Microchip 网站（www.microchip.com）下载。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 的架构具有独立的程序和数据存储空间/总线。这一架构还允许在代码执行过程中从数据空间直接访问程序存储器。

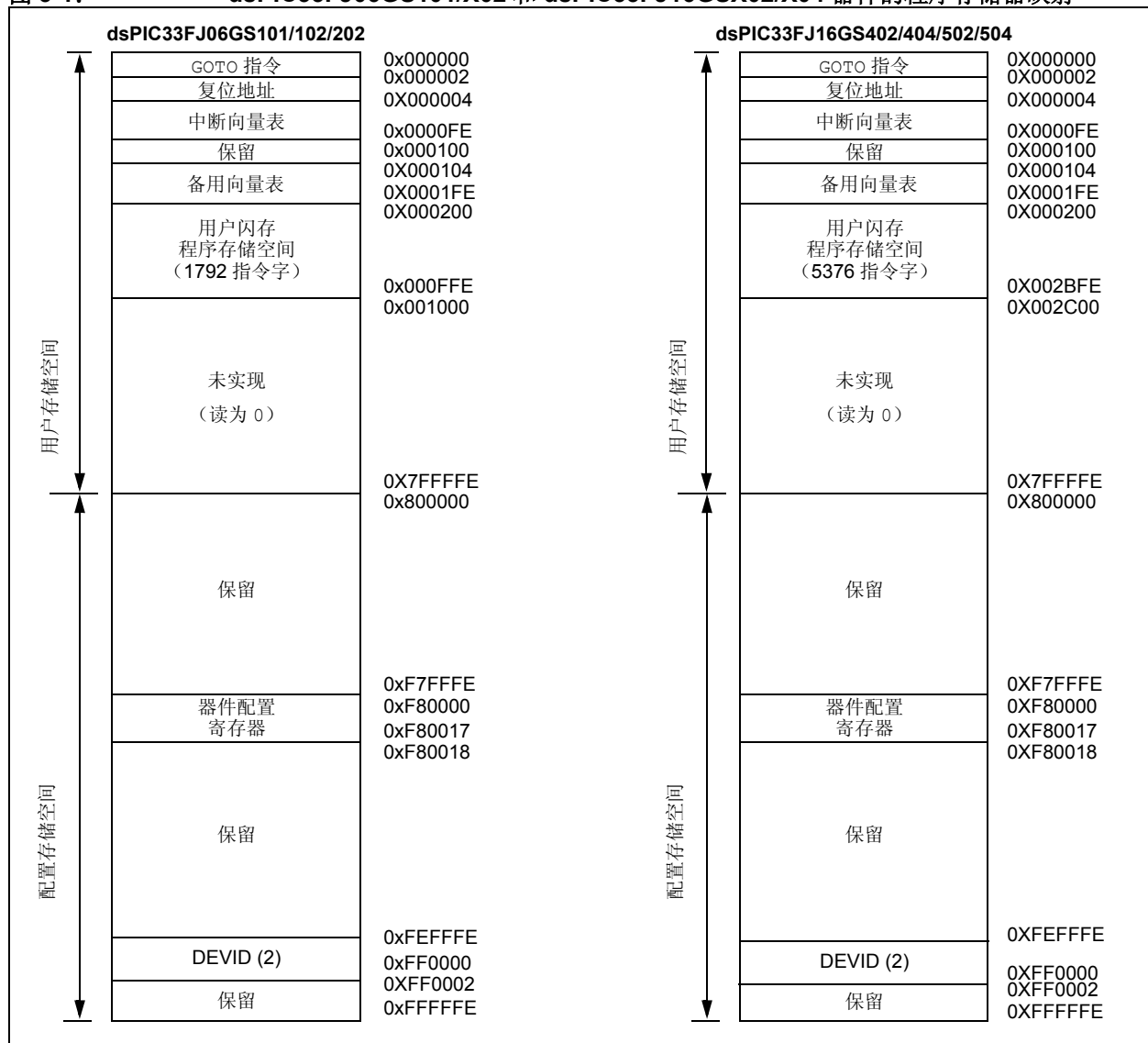
3.1 程序地址空间

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件的程序存储器映射。

图 3-1: dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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（见图 3-2）。

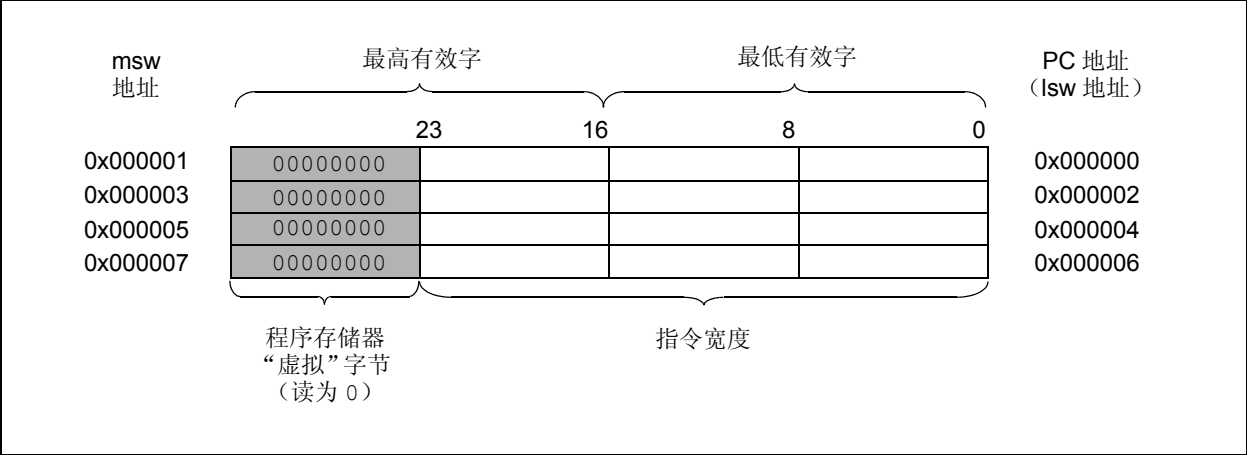
程序存储器地址始终在低位字处按字对齐，并且在代码执行过程中地址将递增或递减 2。这种寻址模式与数据存储空间寻址兼容，且为访问程序存储空间中的数据提供了可能。

3.1.2 中断向量和陷阱向量

所有 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。数据存储单元映射如图 3-3 所示。

数据存储空间中的所有有效地址（Effective Address, EA）均为 16 位宽，并且指向数据空间内的字节。这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”）。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件实现最大 30 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 and 寄存器中的数据是以 16 位字为单位对齐的，但所有数据空间 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® MCU 器件的后向兼容性和提高数据存储空间的使用效率，dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 and 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。dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 256 字节 RAM 的 dsPIC33FJ06GS101/102 器件的数据存储器映射

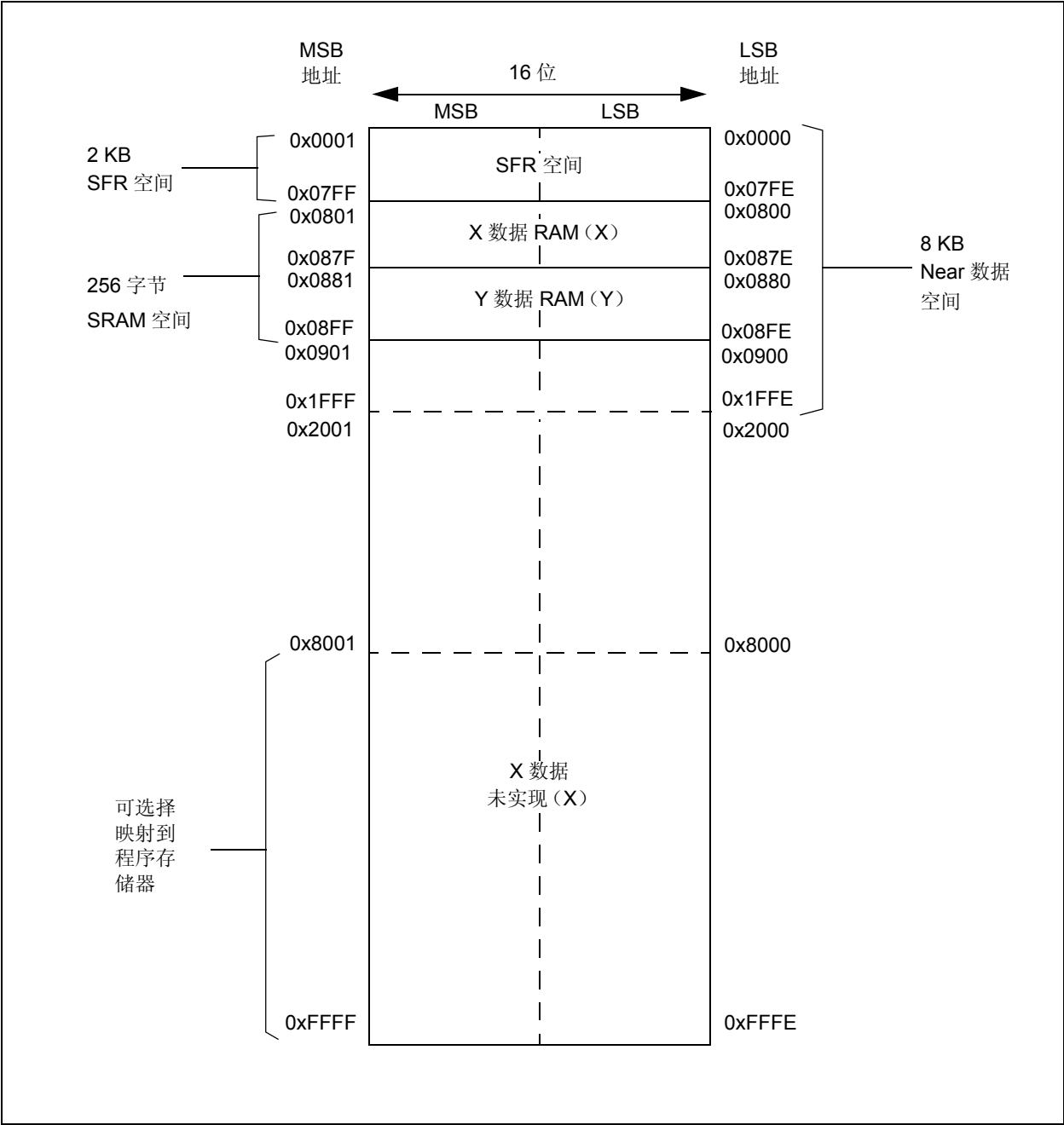


图 3-4: 带 1 KB RAM 的 dsPIC33FJ06GS202 器件的数据存储器映射

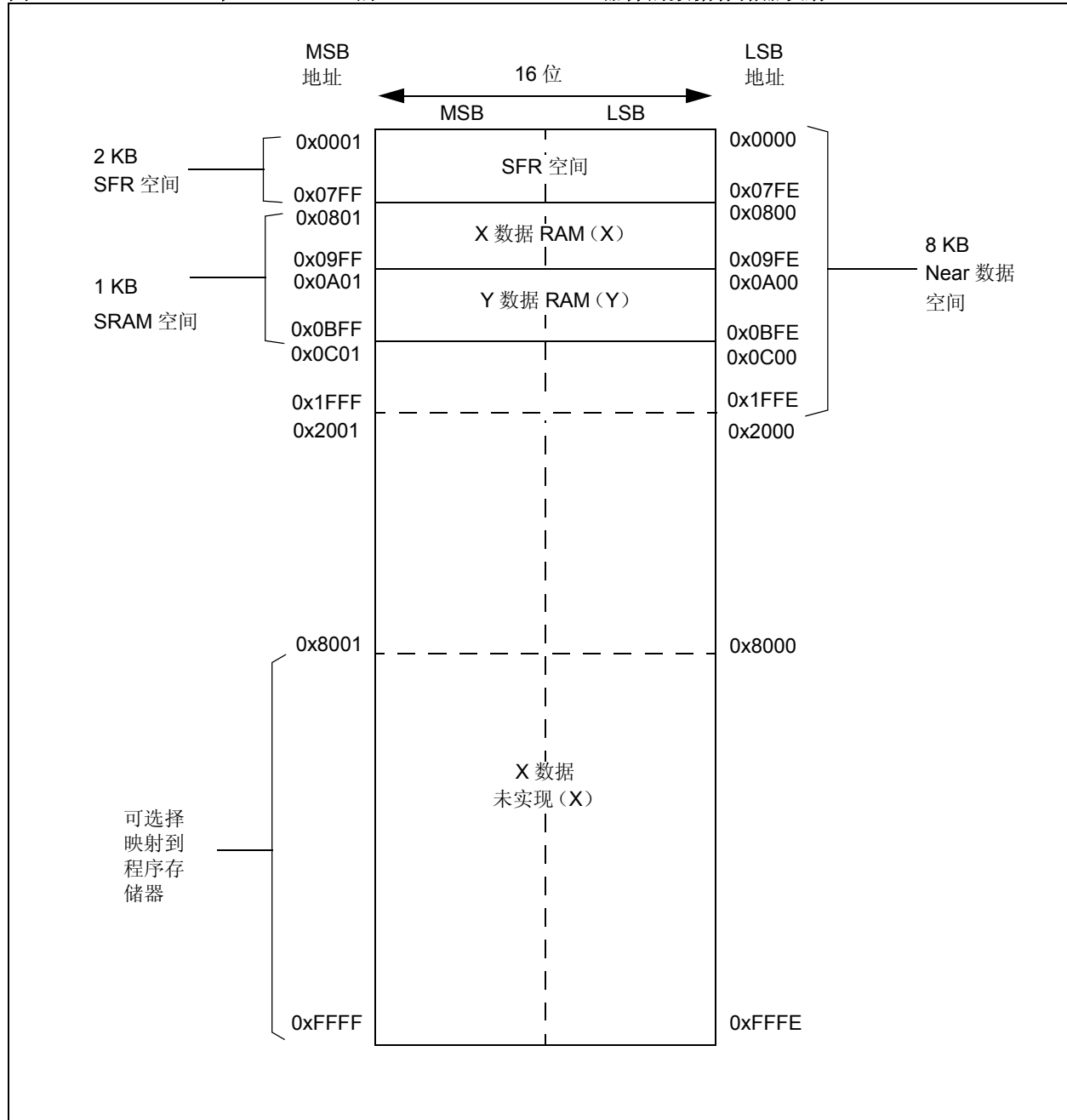
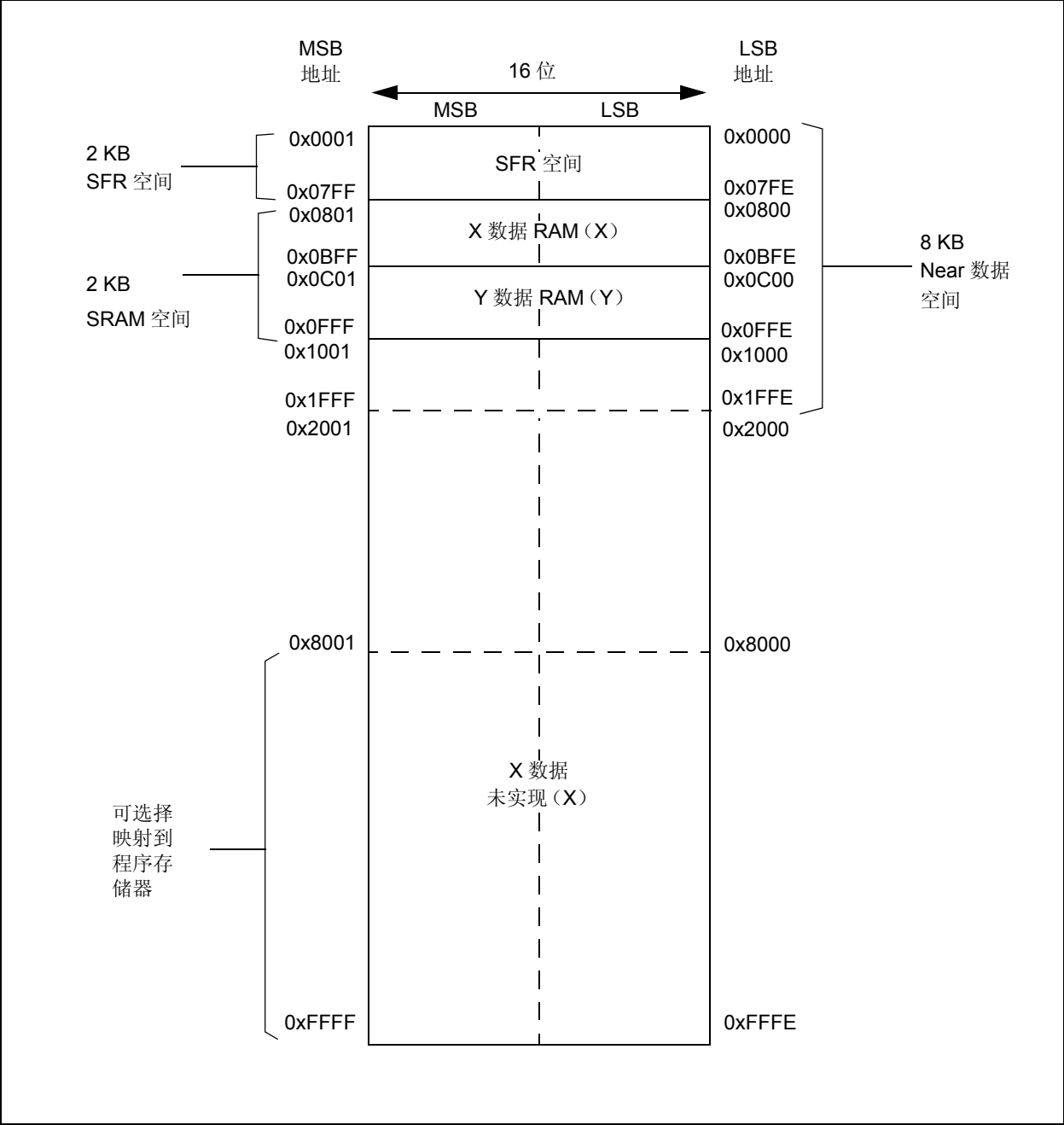


图 3-5: 带 2 KB RAM 的 dsPIC33FJ16GS402/404/502/504 器件的数据存储器映射



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）算法。

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。

所有数据存储器写操作（包括 DSP 指令中的数据存储器写操作）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。

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>	ACCA<39>	ACCA<39>	ACCA<39>	ACCA<39>	ACCA<39>	ACCA<39>	ACCA<39>	ACCAU								xxxx	
ACCB L	0028	ACCB L																xxxx	
ACCBH	002A	ACCBH																xxxx	
ACCBU	002C	ACCB<39>	ACCB<39>	ACCB<39>	ACCB<39>	ACCB<39>	ACCB<39>	ACCB<39>	ACCB<39>	ACCBU								xxxx	
PCL	002E	程序计数器低位字节寄存器																0000	
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	0020	
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。复位值以十六进制表示。

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。复位值以十六进制表示。

表 3-2: dsPIC33FJ06GS101 的电平变化通知寄存器映射

文件寄存器名称	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	—	—	—	—	—	—	—	—	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNPU1	0068	—	—	—	—	—	—	—	—	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	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: dsPIC33FJ06GS102、dsPIC33FJ06GS202、dsPIC33FJ16GS402 和 dsPIC33FJ16GS502 的电平变化通知寄存器映射

文件寄存器名称	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
CNPU1	0068	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	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: dsPIC33FJ16GS404 和 dsPIC33FJ16GS504 的电平变化通知寄存器映射

文件寄存器名称	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	0062	—	—	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	—	—	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 3-5: 仅 dsPIC33FJ06GS101 器件的中断控制器寄存器映射

文件寄存器名称	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	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	—	ADIF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	—	T2IF	—	—	—	T1IF	OC1IF	—	INT0IF	0000
IFS1	0086	—	—	INT2IF	—	—	—	—	—	—	—	—	INT1IF	CNIF	—	MI2C1IF	SI2C1IF	0000
IFS3	008A	—	—	—	—	—	—	PSEMIF	—	—	—	—	—	—	—	—	—	0000
IFS4	008C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIF	—	0000
IFS5	008E	—	PWM1IF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IFS6	0090	ADCP1IF	ADCP0IF	—	—	—	—	—	—	—	—	—	—	—	—	PWM4IF	—	0000
IFS7	0092	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP3IF	—	0000
IEC0	0094	—	—	ADIE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	—	T2IE	—	—	—	T1IE	OC1IE	—	INT0IE	0000
IEC1	0096	—	—	INT2IE	—	—	—	—	—	—	—	—	INT1IE	CNIE	—	MI2C1IE	SI2C1IE	0000
IEC2	0098	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IEC3	009A	—	—	—	—	—	—	PSEMIE	—	—	—	—	—	—	—	—	—	0000
IEC4	009C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIE	—	0000
IEC5	009E	—	PWM1IE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IEC6	00A0	ADCP1IE	ADCP0IE	—	—	—	—	—	—	—	—	—	—	—	—	PWM4IE	—	0000
IEC7	00A2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP3IE	—	0000
IPC0	00A4	—	T1IP<2:0>			—	OC1IP<2:0>			—	—	—	—	—	INT0IP<2:0>			4404
IPC1	00A6	—	T2IP<2:0>			—	—	—	—	—	—	—	—	—	—	—	—	4000
IPC2	00A8	—	U1RXIP<2:0>			—	SPI1IP<2:0>			—	SPI1EIP<2:0>			—	—	—	—	4440
IPC3	00AA	—	—	—	—	—	—	—	—	—	ADIP<2:0>			—	U1TXIP<2:0>			0044
IPC4	00AC	—	CNIP<2:0>			—	—	—	—	—	MI2C1IP<2:0>			—	SI2C1IP<2:0>			4044
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP<2:0>			0004
IPC7	00B2	—	—	—	—	—	—	—	—	—	INT2IP<2:0>			—	—	—	—	0040
IPC14	00C0	—	—	—	—	—	—	—	—	—	PSEMIP<2:0>			—	—	—	—	0040
IPC16	00C4	—	—	—	—	—	—	—	—	—	U1EIP<2:0>			—	—	—	—	0400
IPC23	00D2	—	—	—	—	—	PWM1IP<2:0>			—	—	—	—	—	—	—	—	0040
IPC24	00D4	—	—	—	—	—	—	—	—	—	PWM4IP<2:0>			—	—	—	—	4400
IPC27	00DA	—	ADCP1IP<2:0>			—	ADCP0IP<2:0>			—	—	—	—	—	—	—	—	0040
IPC28	00DC	—	—	—	—	—	—	—	—	—	ADCP3IP<2:0>			—	—	—	—	0000
INTTREG	00E0	—	—	—	—	ILR<3:0>				—	VECNUM<6:0>							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 3-6: 仅 dsPIC33FJ06GS102 器件的中断控制器寄存器映射

文件寄存器名称	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	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	—	ADIF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	—	T2IF	—	—	—	T1IF	OC1IF	—	INT0IF	0000
IFS1	0086	—	—	INT2IF	—	—	—	—	—	—	—	—	INT1IF	CNIF	—	MI2C1IF	SI2C1IF	0000
IFS3	008A	—	—	—	—	—	—	PSEMIF	—	—	—	—	—	—	—	—	—	0000
IFS4	008C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIF	—	0000
IFS5	008E	PWM2IF	PWM1IF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IFS6	0090	ADCP1IF	ADCP0IF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IFS7	0092	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP2IF	0000
IEC0	0094	—	—	ADIE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	—	T2IE	—	—	—	T1IE	OC1IE	—	INT0IE	0000
IEC1	0096	—	—	INT2IE	—	—	—	—	—	—	—	—	INT1IE	CNIE	—	MI2C1IE	SI2C1IE	0000
IEC3	009A	—	—	—	—	—	—	PSEMIE	—	—	—	—	—	—	—	—	—	0000
IEC4	009C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIE	—	0000
IEC5	009E	PWM2IE	PWM1IE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IEC6	00A0	ADCP1IE	ADCP0IE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IEC7	00A2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP2IE	0000
IPC0	00A4	—	T1IP<2:0>			—	OC1IP<2:0>			—	—	—	—	—	INT0IP<2:0>			4404
IPC1	00A6	—	T2IP<2:0>			—	—	—	—	—	—	—	—	—	—	—	—	4000
IPC2	00A8	—	U1RXIP<2:0>			—	SPI1IP<2:0>			—	SPI1EIP<2:0>			—	—	—	—	4440
IPC3	00AA	—	—	—	—	—	—	—	—	—	ADIP<2:0>			—	U1TXIP<2:0>			0044
IPC4	00AC	—	CNIP<2:0>			—	—	—	—	—	MI2C1IP<2:0>			—	SI2C1IP<2:0>			4044
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP<2:0>			0004
IPC7	00B2	—	—	—	—	—	—	—	—	—	INT2IP<2:0>			—	—	—	—	0040
IPC14	00C0	—	—	—	—	—	—	—	—	—	PSEMIP<2:0>			—	—	—	—	0040
IPC16	00C4	—	—	—	—	—	—	—	—	—	U1EIP<2:0>			—	—	—	—	0040
IPC23	00D2	—	PWM2IP<2:0>			—	PWM1IP<2:0>			—	—	—	—	—	—	—	—	4400
IPC27	00DA	—	ADCP1IP<2:0>			—	ADCP0IP<2:0>			—	—	—	—	—	—	—	—	4400
IPC28	00DC	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP2IP<2:0>			0004
INTTREG	00E0	—	—	—	—	ILR<3:0>				—	VECNUM<6:0>							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: 仅 dsPIC33FJ06GS202 器件的中断控制器寄存器映射

文件寄存器名称	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	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	—	ADIF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	—	T2IF	—	—	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	—	—	INT2IF	—	—	—	—	—	—	—	—	INT1IF	CNIF	AC1IF	MI2C1IF	SI2C1IF	0000
IFS3	008A	—	—	—	—	—	—	PSEMIF	—	—	—	—	—	—	—	—	—	0000
IFS4	008C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIF	—	0000
IFS5	008E	PWM2IF	PWM1IF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IFS6	0090	ADCP1IF	ADCP0IF	—	—	—	—	—	—	AC2IF	—	—	—	—	—	—	—	0000
IFS7	0092	—	—	—	—	—	—	—	—	—	—	—	ADCP6IF	—	—	—	ADCP2IF	0000
IEC0	0094	—	—	ADIE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	—	T2IE	—	—	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	—	—	INT2IE	—	—	—	—	—	—	—	—	INT1IE	CNIE	AC1IE	MI2C1IE	SI2C1IE	0000
IEC3	009A	—	—	—	—	—	—	PSEMIE	—	—	—	—	—	—	—	—	—	0000
IEC4	009C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIE	—	0000
IEC5	009E	PWM2IE	PWM1IE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IEC6	00A0	ADCP1IE	ADCP0IE	—	—	—	—	—	—	AC2IE	—	—	—	—	—	—	—	0000
IEC7	00A2	—	—	—	—	—	—	—	—	—	—	—	ADCP6IE	—	—	—	ADCP2IE	0000
IPC0	00A4	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			—	INT0IP<2:0>			4444
IPC1	00A6	—	T2IP<2:0>			—	—	—	—	—	—	—	—	—	—	—	—	4000
IPC2	00A8	—	U1RXIP<2:0>			—	SPI1IP<2:0>			—	SPI1EIP<2:0>			—	—	—	—	4440
IPC3	00AA	—	—	—	—	—	—	—	—	—	ADIP<2:0>			—	U1TXIP<2:0>			0044
IPC4	00AC	—	CNIP<2:0>			—	AC1IP<2:0>			—	MI2C1IP<2:0>			—	SI2C1IP<2:0>			4444
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP<2:0>			0004
IPC7	00B2	—	—	—	—	—	—	—	—	—	INT2IP<2:0>			—	—	—	—	0040
IPC14	00C0	—	—	—	—	—	—	—	—	—	PSEMIP<2:0>			—	—	—	—	0040
IPC16	00C4	—	—	—	—	—	—	—	—	—	U1EIP<2:0>			—	—	—	—	0040
IPC23	00D2	—	PWM2IP<2:0>			—	PWM1IP<2:0>			—	—	—	—	—	—	—	—	4400
IPC25	00D6	—	AC2IP<2:0>			—	—	—	—	—	—	—	—	—	—	—	—	4000
IPC27	00DA	—	ADCP1IP<2:0>			—	ADCP0IP<2:0>			—	—	—	—	—	—	—	—	4400
IPC28	00DC	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP2IP<2:0>			0004
IPC29	00DE	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP6IP<2:0>			0004
INTTREG	00E0	—	—	—	—	ILR<3:0>				—	VECNUM<6:0>							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: 仅 dsPIC33FJ16GS402/404 器件的中断控制器寄存器映射

文件寄存器名称	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	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	—	ADIF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	T3IF	T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	—	—	INT2IF	—	—	—	—	—	—	—	—	INT1IF	CNIF	—	MI2C1IF	SI2C1IF	0000
IFS3	008A	—	—	—	—	—	—	PSEMIF	—	—	—	—	—	—	—	—	—	0000
IFS4	008C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIF	—	0000
IFS5	008E	PWM2IF	PWM1IF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IFS6	0090	ADCP1IF	ADCP0IF	—	—	—	—	—	—	—	—	—	—	—	—	—	PWM3IF	0000
IFS7	0092	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP3IF	ADCP2IF	0000
IEC0	0094	—	—	ADIE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	T3IE	T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	—	—	INT2IE	—	—	—	—	—	—	—	—	INT1IE	CNIE	—	MI2C1IE	SI2C1IE	0000
IEC3	009A	—	—	—	—	—	—	PSEMIE	—	—	—	—	—	—	—	—	—	0000
IEC4	009C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIE	—	0000
IEC5	009E	PWM2IE	PWM1IE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IEC6	00A0	ADCP1IE	ADCP0IE	—	—	—	—	—	—	—	—	—	—	—	—	—	PWM3IE	0000
IEC7	00A2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP3IE	ADCP2IE	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>			—	—	—	—	4440
IPC2	00A8	—	U1RXIP<2:0>			—	SPI1IP<2:0>			—	SPI1EIP<2:0>			—	T3IP<2:0>			4444
IPC3	00AA	—	—	—	—	—	—	—	—	—	ADIP<2:0>			—	U1TXIP<2:0>			0044
IPC4	00AC	—	CNIP<2:0>			—	—	—	—	—	MI2C1IP<2:0>			—	SI2C1IP<2:0>			4044
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP<2:0>			0004
IPC7	00B2	—	—	—	—	—	—	—	—	—	INT2IP<2:0>			—	—	—	—	0040
IPC14	00C0	—	—	—	—	—	—	—	—	—	PSEMIP<2:0>			—	—	—	—	0040
IPC16	00C4	—	—	—	—	—	—	—	—	—	U1EIP<2:0>			—	—	—	—	0040
IPC23	00D2	—	PWM2IP<2:0>			—	PWM1IP<2:0>			—	—			—	—	—	—	4400
IPC24	00D4	—	—	—	—	—	—	—	—	—	—	—	—	—	PWM3IP<2:0>			0004
IPC27	00DA	—	ADCP1IP<2:0>			—	ADCP0IP<2:0>			—	—			—	—	—	—	4400
IPC28	00DC	—	—	—	—	—	—	—	—	—	ADCP3IP<2:0>			—	ADCP2IP<2:0>			0044
INTTREG	00E0	—	—	—	—	ILR<3:0>				—	VECNUM<6:0>							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 3-9: 仅 dsPIC33FJ16GS502 器件的中断控制器寄存器映射

文件寄存器名称	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	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	—	ADIF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	T3IF	T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	—	—	INT2IF	—	—	—	—	—	—	—	—	INT1IF	CNIF	AC1IF	MI2C1IF	SI2C1IF	0000
IFS3	008A	—	—	—	—	—	—	PSEMIF	—	—	—	—	—	—	—	—	—	0000
IFS4	008C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIF	—	0000
IFS5	008E	PWM2IF	PWM1IF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IFS6	0090	ADCP1IF	ADCP0IF	—	—	—	—	AC4IF	AC3IF	AC2IF	—	—	—	—	—	PWM4IF	PWM3IF	0000
IFS7	0092	—	—	—	—	—	—	—	—	—	—	—	ADCP6IF	—	—	ADCP3IF	ADCP2IF	0000
IEC0	0094	—	—	ADIE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	T3IE	T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	—	—	INT2IE	—	—	—	—	—	—	—	—	INT1IE	CNIE	AC1IE	MI2C1IE	SI2C1IE	0000
IEC3	009A	—	—	—	—	—	—	PSEMIE	—	—	—	—	—	—	—	—	—	0000
IEC4	009C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIE	—	0000
IEC5	009E	PWM2IE	PWM1IE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IEC6	00A0	ADCP1IE	ADCP0IE	—	—	—	—	AC4IE	AC3IE	AC2IE	—	—	—	—	—	PWM4IE	PWM3IE	0000
IEC7	00A2	—	—	—	—	—	—	—	—	—	—	—	ADCP6IE	—	—	ADCP3IE	ADCP2IE	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>			—	—	—	—	4440
IPC2	00A8	—	U1RXIP<2:0>			—	SPI1IP<2:0>			—	SPI1EIP<2:0>			—	T3IP<2:0>			4444
IPC3	00AA	—	—	—	—	—	—			—	ADIP<2:0>			—	U1TXIP<2:0>			0044
IPC4	00AC	—	CNIP<2:0>			—	AC1IP<2:0>			—	MI2C1IP<2:0>			—	SI2C1IP<2:0>			4444
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP<2:0>			0004
IPC7	00B2	—	—	—	—	—	—	—	—	—	INT2IP<2:0>			—	—	—	—	0040
IPC14	00C0	—	—	—	—	—	—	—	—	—	PSEMIP<2:0>			—	—	—	—	0040
IPC16	00C4	—	—	—	—	—	—	—	—	—	U1EIP<2:0>			—	—	—	—	0040
IPC23	00D2	—	PWM2IP<2:0>			—	PWM1IP<2:0>			—	—	—	—	—	—	—	—	4400
IPC24	00D4	—	—	—	—	—	—	—	—	—	PWM4IP<2:0>			—	PWM3IP<2:0>			0044
IPC25	00D6	—	AC2IP<2:0>			—	—	—	—	—	—	—	—	—	—	—	—	4000
IPC26	00D8	—	—	—	—	—	—	—	—	—	AC4IP<2:0>			—	AC3IP<2:0>			0044
IPC27	00DA	—	ADCP1IP<2:0>			—	ADCP0IP<2:0>			—	—			—	—	—	—	4400
IPC28	00DC	—	—	—	—	—	—	—	—	—	ADCP3IP<2:0>			—	ADCP2IP<2:0>			0044
IPC29	00DE	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP6IP<2:0>			0004
INTTREG	00E0	—	—	—	—	ILR<3:0>				—	VECNUM<6:0>							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 3-10: 仅 dsPIC33FJ16GS504 器件的中断控制器寄存器映射

文件寄存器名称	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	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	—	ADIF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	T3IF	T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	—	—	INT2IF	—	—	—	—	—	—	—	—	INT1IF	CNIF	AC1IF	MI2C1IF	SI2C1IF	0000
IFS3	008A	—	—	—	—	—	—	PSEMIF	—	—	—	—	—	—	—	—	—	0000
IFS4	008C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIF	—	0000
IFS5	008E	PWM2IF	PWM1IF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IFS6	0090	ADCP1IF	ADCP0IF	—	—	—	—	AC4IF	AC3IF	AC2IF	—	—	—	—	—	PWM4IF	PWM3IF	0000
IFS7	0092	—	—	—	—	—	—	—	—	—	—	—	ADCP6IF	ADCP5IF	ADCP4IF	ADCP3IF	ADCP2IF	0000
IEC0	0094	—	—	ADIE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	T3IE	T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	—	—	INT2IE	—	—	—	—	—	—	—	—	INT1IE	CNIE	AC1IE	MI2C1IE	SI2C1IE	0000
IEC3	009A	—	—	—	—	—	—	PSEMIE	—	—	—	—	—	—	—	—	—	0000
IEC4	009C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	U1EIE	—	0000
IEC5	009E	PWM2IE	PWM1IE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
IEC6	00A0	ADCP1IE	ADCP0IE	—	—	—	—	AC4IE	AC3IE	AC2IE	—	—	—	—	—	PWM4IE	PWM3IE	0000
IEC7	00A2	—	—	—	—	—	—	—	—	—	—	—	ADCP6IE	ADCP5IE	ADCP4IE	ADCP3IE	ADCP2IE	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>			—	—	—	—	4440
IPC2	00A8	—	U1RXIP<2:0>			—	SPI1IP<2:0>			—	SPI1EIP<2:0>			—	T3IP<2:0>			4444
IPC3	00AA	—	—	—	—	—	—	—	—	—	ADIP<2:0>			—	U1TXIP<2:0>			0044
IPC4	00AC	—	CNIP<2:0>			—	AC1IP<2:0>			—	MI2C1IP<2:0>			—	SI2C1IP<2:0>			4444
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP<2:0>			0004
IPC7	00B2	—	—	—	—	—	—	—	—	—	INT2IP<2:0>			—	—	—	—	0040
IPC14	00C0	—	—	—	—	—	—	—	—	—	PSEMIP<2:0>			—	—	—	—	0040
IPC16	00C4	—	—	—	—	—	—	—	—	—	U1EIP<2:0>			—	—	—	—	0040
IPC23	00D2	—	PWM2IP<2:0>			—	PWM1IP<2:0>			—	—	—	—	—	—	—	—	4400
IPC24	00D4	—	—	—	—	—	—	—	—	—	PWM4IP<2:0>			—	PWM3IP<2:0>			0044
IPC25	00D6	—	AC2IP<2:0>			—	—	—	—	—	—	—	—	—	—	—	—	4000
IPC26	00D8	—	—	—	—	—	—	—	—	—	AC4IP<2:0>			—	AC3IP<2:0>			0440
IPC27	00DA	—	ADCP1IP<2:0>			—	ADCP0IP<2:0>			—	—	—	—	—	—	—	—	4400
IPC28	00DC	—	ADCP5IP<2:0>			—	ADCP4IP<2:0>			—	ADCP3IP<2:0>			—	ADCP2IP<2:0>			4444
IPC29	00DE	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCP6IP<2:0>			0004
INTTREG	00E0	—	—	—	—	ILR<3:0>				—	VECNUM<6:0>							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 3-11: dsPIC33FJ06GS101 和 dsPIC33FJ06GSX02 的定时器寄存器映射

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
PR2	010C	周期寄存器 2																FFFF
T2CON	0110	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 3-12: dsPIC33FJ16GSX02 和 dsPIC33FJ16GSX04 的定时器寄存器映射

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	

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: dsPIC33FJ06GS202 的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	

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: dsPIC33FJ16GSX02 和 dsPIC33FJ16GSX04 的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	

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: dsPIC33FJ06GS101 和 dsPIC33FJ06GSX02 的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

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: dsPIC33FJ16GSX02 和 dsPIC33FJ06GSX04 的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																xxxxx
OC2CON	018A	—	—	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 3-17: 高速 PWM 模块寄存器映射

文件寄存器名称	地址偏移	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	
PTCON	0400	PTEN	—	PTSIDL	SESTAT	SEIEN	EIPU	SYNCPOL	SYNCOEN	SYNCEN	—	SYNCSRC<1:0>		SEVTPS<3:0>				0000	
PTCON2	0402	—	—	—	—	—	—	—	—	—	—	—	—	—	PCLKDIV<2:0>			0000	
PTPER	0404	PTPER<15:0>																FFF8	
SEVTCMP	0406	SEVTCMP<15:3>														—	—	—	0000
MDC	040A	MDC<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 3-18: 高速 PWM 发生器 1 寄存器映射

文件寄存器名称	地址偏移	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	
PWMCON1	0420	FLTSTAT	CLSTAT	TRGSTAT	FLTIEEN	CLIEEN	TRGIEEN	ITB	MDCS	DTC<1:0>		—	—	—	CAM	XPRES	IUE	0000	
IOCON1	0422	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		SWAP	OSYNC	0000	
FCLCON1	0424	IFLTMOD	CLSRC<4:0>					CLPOL	CLMOD	FLTSRC<4:0>					FLTPOL	FLTMOD<1:0>		0000	
PDC1	0426	PDC1<15:0>																	0000
PHASE1	0428	PHASE1<15:0>																	0000
DTR1	042A	—	—	DTR1<13:0>														0000	
ALTDTR1	042C	—	—	ALTDTR1<13:0>														0000	
SDC1	042E	SDC1<15:0>																	0000
SPHASE1	0430	SPHASE1<15:0>																	0000
TRIG1	0432	TRGCMP<15:3>													—	—	—	0000	
TRGCON1	0434	TRGDIV<3:0>				—	—	—	—	DTM	—	TRGSTRT<5:0>						0000	
STRIG1	0436	STRGCMP<15:3>													—	—	—	0000	
PWMCAP1	0438	PWMCAP1<15:3>													—	—	—	0000	
LEBCON1	043A	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>						—	—	—	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: 仅 dsPIC33FJ06GS102/202 和 dsPIC33FJ16GSX02/X04 器件的高速 PWM 发生器 2 寄存器映射

文件寄存器名称	地址偏移	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	
PWMCON2	0440	FLTSTAT	CLSTAT	TRGSTAT	FLTIE	CLIE	TRGIE	ITB	MDCS	DTC<1:0>		—	—	—	CAM	XPRES	IUE	0000	
IOCON2	0442	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		SWAP	OSYNC	0000	
FCLCON2	0444	IFLTMOD	CLSRC<4:0>					CLPOL	CLMOD	FLTSRC<4:0>					FLTPOL	FLTMOD<1:0>		0000	
PDC2	0446	PDC2<15:0>																	0000
PHASE2	0448	PHASE2<15:0>																	0000
DTR2	044A	—	—	DTR2<13:0>														0000	
ALTDTR2	044C	—	—	ALTDTR2<13:0>														0000	
SDC2	044E	SDC2<15:0>																	0000
SPHASE2	0450	SPHASE2<15:0>																	0000
TRIG2	0452	TRGCMP<15:3>													—	—	—	0000	
TRGCON2	0454	TRGDIV<3:0>				—	—	—	—	DTM	—	TRGSTRT<5:0>							0000
STRIG2	0456	STRGCMP<15:3>													—	—	—	0000	
PWMCAP2	0458	PWMCAP2<15:3>													—	—	—	0000	
LEBCON2	045A	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>							—	—	—	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: 仅 dsPIC33FJ16GSX02/X04 器件的高速 PWM 发生器 3 寄存器映射

文件寄存器名称	地址偏移	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
PWMCON3	0460	FLTSTAT	CLSTAT	TRGSTAT	FLTIE	CLIE	TRGIE	ITB	MDCS	DTC<1:0>		—	—	—	CAM	XPRES	IUE	0000
IOCON3	0462	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		SWAP	OSYNC	0000
FCLCON3	0464	IFLTMOD	CLSRC<4:0>					CLPOL	CLMOD	FLTSRC<4:0>					FLTPOL	FLTMOD<1:0>		0000
PDC3	0466	PDC3<15:0>																0000
PHASE3	0468	PHASE3<15:0>																0000
DTR3	046C	—	—	DTR3<13:0>														0000
ALTDTR3	046C	—	—	ALTDTR3<13:0>														0000
SDC3	046E	SDC3<15:0>																0000
SPHASE3	0470	SPHASE3<15:0>																0000
TRIG3	0472	TRGCMP<15:3>													—	—	—	0000
TRGCON3	0474	TRGDIV<3:0>				—	—	—	—	DTM	—	TRGSTRT<5:0>						0000
STRIG3	0476	STRGCMP<15:3>													—	—	—	0000
PWMCAP3	0478	PWMCAP3<15:3>													—	—	—	0000
LEBCON3	047A	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>						—	—	—	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 3-21: 仅 dsPIC33FJ06GS101 和 dsPIC33FJ16GS50X 器件的高速 PWM 发生器 4 寄存器映射

文件寄存器名称	地址偏移	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
PWMCON4	0480	FLTSTAT	CLSTAT	TRGSTAT	FLTIE	CLIE	TRGIE	ITB	MDCS	DTC<1:0>		—	—	—	CAM	XPRES	IUE	0000
IOCON4	0482	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		SWAP	OSYNC	0000
FCLCON4	0484	IFLTMOD	CLSRC<4:0>					CLPOL	CLMOD	FLTSRC<4:0>					FLTPOL	FLTMOD<1:0>		0000
PDC4	0486	PDC4<15:0>																0000
PHASE4	0488	PHASE4<15:0>																0000
DTR4	048A	—	—	DTR4<13:0>														0000
ALTDTR4	048A	—	—	ALTDTR4<13:0>														0000
SDC4	048E	SDC4<15:0>																0000
SPHASE4	0490	SPHASE4<15:0>																0000
TRIG4	0492	TRGCMP<15:3>													—	—	—	0000
TRGCON4	0494	TRGDIV<3:0>				—	—	—	—	DTM	—	TRGSTRT<5:0>						0000
STRIG4	0496	STRGCMP<15:3>													—	—	—	0000
PWMCAP4	0498	PWMCAP4<15:3>													—	—	—	0000
LEBCON4	049A	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>						—	—	—	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 3-22: I2C1 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	—	—	—	—	—	—	AMSK<9:0>										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 3-23: 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	RTSMO	—	UEN1	UEN0	WAKE	LPBACK	ABAUO	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	—	—	—	—	—	—	—	UART 发送寄存器									xxxx
U1RXREG	0226	—	—	—	—	—	—	—	UART 接收寄存器									0000
U1BRG	0228	波特率发生器预分频比																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 3-24: 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 3-25: 仅 dsPIC33FJ06GS101 器件的高速 10 位 ADC 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	
ADCON	0300	ADON	—	ADSIDL	SLOWCLK	—	GSWTRG	—	FORM	EIE	ORDER	SEQSAMP	ASYNCSAMP	—	ADCS<2:0>			0000	
ADPCFG	0302	—	—	—	—	—	—	—	—	PCFG7	PCFG6	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0000	
ADSTAT	0306	—	—	—	—	—	—	—	—	—	—	—	—	P3RDY	—	P1RDY	P0RDY	0000	
ADBASE	0308	ADBASE<15:1>															—	0000	
ADCPC0	030A	IRQEN1	PEND1	SWTRG1	TRGSRC1<4:0>					IRQEN0	PEND0	SWTRG0	TRGSRC0<4:0>					0000	
ADCPC1	030C	IRQEN3	PEND3	SWTRG3	TRGSRC3<4:0>					—	—	—	—	—	—	—	—	—	0000
ADCBUF0	0320	ADC 数据缓冲区 0																xxxx	
ADCBUF1	0322	ADC 数据缓冲区 1																xxxx	
ADCBUF2	0324	ADC 数据缓冲区 2																xxxx	
ADCBUF3	0326	ADC 数据缓冲区 3																xxxx	
ADCBUF6	032C	ADC 数据缓冲区 6																xxxx	
ADCBUF7	032E	ADC 数据缓冲区 7																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 3-26: 仅 dsPIC33FJ06GS102 器件的高速 10 位 ADC 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
ADCON	0300	ADON	—	ADSIDL	SLOWCLK	—	GSWTRG	—	FORM	EIE	ORDER	SEQSAMP	ASYNC SAMP	—	ADCS<2:0>			0003
ADPCFG	0302	—	—	—	—	—	—	—	—	—	—	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
ADSTAT	0306	—	—	—	—	—	—	—	—	—	—	—	—	—	P2RDY	P1RDY	P0RDY	0000
ADBASE	0308	ADBASE<15:1>															—	0000
ADCPC0	030A	IRQEN1	PEND1	SWTRG1	TRGSRC1<4:0>					IRQEN0	PEND0	SWTRG0	TRGSRC0<4:0>					0000
ADCPC1	030C	—	—	—	—	—	—	—	—	IRQEN2	PEND2	SWTRG2	TRGSRC2<4:0>					0000
ADCBUF0	0320	ADC 数据缓冲区 0																xxxx
ADCBUF1	0322	ADC 数据缓冲区 1																xxxx
ADCBUF2	0324	ADC 数据缓冲区 2																xxxx
ADCBUF3	0326	ADC 数据缓冲区 3																xxxx
ADCBUF4	0328	ADC 数据缓冲区 4																xxxx
ADCBUF5	032A	ADC 数据缓冲区 5																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 3-27: 仅 dsPIC33FJ06GS202 器件的高速 10 位 ADC 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	
ADCON	0300	ADON	—	ADSIDL	SLOWCLK	—	GSWTRG	—	FORM	EIE	ORDER	SEQSAMP	ASYNC SAMP	—	ADCS<2:0>			0003	
ADPCFG	0302	—	—	—	—	—	—	—	—	—	—	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000	
ADSTAT	0306	—	—	—	—	—	—	—	—	—	P6RDY	—	—	—	P2RDY	P1RDY	P0RDY	0000	
ADBASE	0308	ADBASE<15:1>																—	0000
ADCPC0	030A	IRQEN1	PEND1	SWTRG1	TRGSRC1<4:0>					IRQEN0	PEND0	SWTRG0	TRGSRC0<4:0>					0000	
ADCPC1	030C	—	—	—	—	—	—	—	—	IRQEN2	PEND2	SWTRG2	TRGSRC2<4:0>					0000	
ADCPC3	0310	—	—	—	—	—	—	—	—	IRQEN6	PEND6	SWTRG6	TRGSRC6<4:0>					0000	
ADCBUF0	0320	ADC 数据缓冲区 0																	xxxx
ADCBUF1	0322	ADC 数据缓冲区 1																	xxxx
ADCBUF2	0324	ADC 数据缓冲区 2																	xxxx
ADCBUF3	0326	ADC 数据缓冲区 3																	xxxx
ADCBUF4	0328	ADC 数据缓冲区 4																	xxxx
ADCBUF5	032A	ADC 数据缓冲区 5																	xxxx
ADCBUF12	0338	ADC 数据缓冲区 12																	xxxx
ADCBUF13	033A	ADC 数据缓冲区 13																	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 3-28: 仅 dsPIC33FJ16GS402/404 器件的高速 10 位 ADC 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
ADCON	0300	ADON	—	ADSIDL	SLOWCLK	—	GSWTRG	—	FORM	EIE	ORDER	SEQSAMP	ASYNCSAMP	—	ADCS<2:0>			0003
ADPCFG	0302	—	—	—	—	—	—	—	—	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
ADSTAT	0306	—	—	—	—	—	—	—	—	—	—	—	—	P3RDY	P2RDY	P1RDY	P0RDY	0000
ADBASE	0308	ADBASE<15:1>															—	0000
ADCPC0	030A	IRQEN1	PEND1	SWTRG1	TRGSRC1<4:0>					IRQEN0	PEND0	SWTRG0	TRGSRC0<4:0>					0000
ADCPC1	030C	IRQEN3	PEND3	SWTRG3	TRGSRC3<4:0>					IRQEN2	PEND2	SWTRG2	TRGSRC2<4:0>					0000
ADCBUF0	0320	ADC 数据缓冲区 0																xxxx
ADCBUF1	0322	ADC 数据缓冲区 1																xxxx
ADCBUF2	0324	ADC 数据缓冲区 2																xxxx
ADCBUF3	0326	ADC 数据缓冲区 3																xxxx
ADCBUF4	0328	ADC 数据缓冲区 4																xxxx
ADCBUF5	032A	ADC 数据缓冲区 5																xxxx
ADCBUF6	032C	ADC 数据缓冲区 6																xxxx
ADCBUF7	032E	ADC 数据缓冲区 7																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 3-29: 仅 dsPIC33FJ16GS502 器件的高速 10 位 ADC 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	
ADCON	0300	ADON	—	ADSIDL	SLOWCLK	—	GSWTRG	—	FORM	EIE	ORDER	SEQSAMP	ASYNCSAMP	—	ADCS<2:0>			0003	
ADPCFG	0302	—	—	—	—	—	—	—	—	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000	
ADSTAT	0306	—	—	—	—	—	—	—	—	—	P6RDY	—	—	P3RDY	P2RDY	P1RDY	P0RDY	0000	
ADBASE	0308	ADBASE<15:1>															—	0000	
ADCPC0	030A	IRQEN1	PEND1	SWTRG1	TRGSRC1<4:0>					IRQEN0	PEND0	SWTRG0	TRGSRC0<4:0>					0000	
ADCPC1	030C	IRQEN3	PEND3	SWTRG3	TRGSRC3<4:0>					IRQEN2	PEND2	SWTRG2	TRGSRC2<4:0>					0000	
ADCPC3	0310	—	—	—	—	—	—	—	—	IRQEN6	PEND6	SWTRG6	TRGSRC6<4:0>					0000	
ADCBUF0	0320	ADC 数据缓冲区 0																	xxxx
ADCBUF1	0322	ADC 数据缓冲区 1																	xxxx
ADCBUF2	0324	ADC 数据缓冲区 2																	xxxx
ADCBUF3	0326	ADC 数据缓冲区 3																	xxxx
ADCBUF4	0328	ADC 数据缓冲区 4																	xxxx
ADCBUF5	032A	ADC 数据缓冲区 5																	xxxx
ADCBUF6	032C	ADC 数据缓冲区 6																	xxxx
ADCBUF7	032E	ADC 数据缓冲区 7																	xxxx
ADCBUF12	0338	ADC 数据缓冲区 12																	xxxx
ADCBUF13	033A	ADC 数据缓冲区 13																	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 3-30: 仅 dsPIC33FJ16GS504 器件的高速 10 位 ADC 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	
ADCON	0300	ADON	—	ADSIDL	SLOWCLK	—	GSWTRG	—	FORM	EIE	ORDER	SEQSAMP	ASYNCSAMP	—	ADCS<2:0>			0003	
ADPCFG	0302	—	—	—	—	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000	
ADSTAT	0306	—	—	—	—	—	—	—	—	—	P6RDY	P5RDY	P4RDY	P3RDY	P2RDY	P1RDY	P0RDY	0000	
ADBASE	0308	ADBASE<15:1>																—	0000
ADCPC0	030A	IRQEN1	PEND1	SWTRG1	TRGSRC1<4:0>					IRQEN0	PEND0	SWTRG0	TRGSRC0<4:0>					0000	
ADCPC1	030C	IRQEN3	PEND3	SWTRG3	TRGSRC3<4:0>					IRQEN2	PEND2	SWTRG2	TRGSRC2<4:0>					0000	
ADCPC2	030E	IRQEN5	PEND5	SWTRG5	TRGSRC5<4:0>					IRQEN4	PEND4	SWTRG4	TRGSRC4<4:0>					0000	
ADCPC3	0310	—	—	—	—	—	—	—	—	IRQEN6	PEND6	SWTRG6	TRGSRC6<4:0>					0000	
ADCBUF0	0320	ADC 数据缓冲区 0																xxxx	
ADCBUF1	0322	ADC 数据缓冲区 1																xxxx	
ADCBUF2	0324	ADC 数据缓冲区 2																xxxx	
ADCBUF3	0326	ADC 数据缓冲区 3																xxxx	
ADCBUF4	0328	ADC 数据缓冲区 4																xxxx	
ADCBUF5	032A	ADC 数据缓冲区 5																xxxx	
ADCBUF6	032C	ADC 数据缓冲区 6																xxxx	
ADCBUF7	032E	ADC 数据缓冲区 7																xxxx	
ADCBUF8	0330	ADC 数据缓冲区 8																xxxx	
ADCBUF9	0332	ADC 数据缓冲区 9																xxxx	
ADCBUF10	0334	ADC 数据缓冲区 10																xxxx	
ADCBUF11	0336	ADC 数据缓冲区 11																xxxx	
ADCBUF12	0338	ADC 数据缓冲区 12																xxxx	
ADCBUF13	033A	ADC 数据缓冲区 13																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 3-31: 仅 dsPIC33FJ06GS202 器件的模拟比较器控制寄存器映射

文件寄存器名称	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
CMPCON1	0540	CMPON	—	CMPSIDL	—	—	—	—	DACOE	INSEL<1:0>		EXTREF	—	CMPSTAT	—	CMPPOL	RANGE	0000
CMPDAC1	0542	—	—	—	—	—	—	CMREF<9:0>										0000
CMPCON2	0544	CMPON	—	CMPSIDL	—	—	—	—	DACOE	INSEL<1:0>		EXTREF	—	CMPSTAT	—	CMPPOL	RANGE	0000
CMPDAC2	0546	—	—	—	—	—	—	CMREF<9:0>										0000

表 3-32: 仅 dsPIC33FJ16GS502/504 器件的模拟比较器控制寄存器映射

文件寄存器名称	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
CMPCON1	0540	CMPON	—	CMPSIDL	—	—	—	—	DACOE	INSEL<1:0>		EXTREF	—	CMPSTAT	—	CMPPOL	RANGE	0000
CMPDAC1	0542	—	—	-	—	—	—	CMREF<9:0>										0000
CMPCON2	0544	CMPON	—	CMPSIDL	—	—	—	—	DACOE	INSEL<1:0>		EXTREF	—	CMPSTAT	—	CMPPOL	RANGE	0000
CMPDAC2	0546	—	—	-	—	—	—	CMREF<9:0>										0000
CMPCON3	0548	CMPON	—	CMPSIDL	—	—	—	—	DACOE	INSEL<1:0>		EXTREF	—	CMPSTAT	—	CMPPOL	RANGE	0000
CMPDAC3	054A	—	—	-	—	—	—	CMREF<9:0>										0000
CMPCON4	054C	CMPON	—	CMPSIDL	—	—	—	—	DACOE	INSEL<1:0>		EXTREF	—	CMPSTAT	—	CMPPOL	RANGE	0000
CMPDAC4	054E	—	—	—	—	—	—	CMREF<9:0>										0000

表 3-33: 外设引脚选择输入寄存器映射

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
RPINR0	0680	—	—	INT1R<5:0>						—	—	—	—	—	—	—	—	3F00
RPINR1	0682	—	—	—	—	—	—	—	—	—	—	INT2R<5:0>						003F
RPINR2	0684	—	—	T1CKR<5:0>						—	—	—	—	—	—	—	—	0000
RPINR3	0686	—	—	T3CKR<5:0>						—	—	T2CKR<5:0>						3F3F
RPINR7	068E	—	—	IC2R<5:0>						—	—	IC1R<5:0>						3F3F
RPINR11	0696	—	—	—	—	—	—	—	—	—	—	OCFAR<5:0>						3F3F
RPINR18	06A4	—	—	U1CTSR<5:0>						—	—	U1RXR<5:0>						003F
RPINR20	06A8	—	—	SCK1R<5:0>						—	—	SDI1R<5:0>						3F3F
RPINR21	06AA	—	—	—	—	—	—	—	—	—	—	SS1R<5:0>						0000
RPINR29	06BA	—	—	FLT1R<5:0>						—	—	—	—	—	—	—	—	3F00
RPINR30	06BC	—	—	FLT3R<5:0>						—	—	FLT2R<5:0>						3F3F
RPINR31	06BE	—	—	FLT5R<5:0>						—	—	FLT4R<5:0>						3F3F
RPINR32	06C0	—	—	FLT7R<5:0>						—	—	FLT6R<5:0>						3F3F
RPINR33	06C2	—	—	SYNC1R<5:0>						—	—	FLT8R<5:0>						3F3F
RPINR34	06C4	—	—	—	—	—	—	—	—	—	—	SYNCI2R<5:0>						3F3F

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: dsPIC33FJ06GS101 器件的外设引脚选择输出寄存器映射

文件寄存器名称	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	06D0	—	—	RP1R<5:0>						—	—	RP0R<5:0>						0000
RPOR1	06D2	—	—	RP3R<5:0>						—	—	RP2R<5:0>						0000
RPOR2	06D4	—	—	RP5R<5:0>						—	—	RP4R<5:0>						0000
RPOR3	06D6	—	—	RP7R<5:0>						—	—	RP6R<5:0>						0000
RPOR16	06F0	—	—	RP33<5:0>						—	—	RP32<5:0>						0000
RPOR17	06F2	—	—	RP35<5:0>						—	—	RP34<5:0>						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 3-35: dsPIC33FJ06GS102、dsPIC33FJ06GS202、dsPIC33FJ16GS402 和 dsPIC33FJ16GS502 器件的外设引脚选择输出寄存器映射

文件寄存器名称	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	06D0	—	—	RP1R<5:0>						—	—	RP0R<5:0>						0000
RPOR1	06D2	—	—	RP3R<5:0>						—	—	RP2R<5:0>						0000
RPOR2	06D4	—	—	RP5R<5:0>						—	—	RP4R<5:0>						0000
RPOR3	06D6	—	—	RP7R<5:0>						—	—	RP6R<5:0>						0000
RPOR4	06D8	—	—	RP9R<5:0>						—	—	RP8R<5:0>						0000
RPOR5	06DA	—	—	RP11R<5:0>						—	—	RP10R<5:0>						0000
RPOR6	06DC	—	—	RP13R<5:0>						—	—	RP12R<5:0>						0000
RPOR7	06DE	—	—	RP15R<5:0>						—	—	RP14R<5:0>						0000
RPOR16	06F0	—	—	RP33<5:0>						—	—	RP32<5:0>						0000
RPOR17	06F2	—	—	RP35<5:0>						—	—	RP34<5:0>						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 3-36: dsPIC33FJ16GS404 和 dsPIC33FJ16GS504 器件的外设引脚选择输出寄存器映射

文件寄存器名称	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	06D0	—	—	RP1R<5:0>						—	—	RP0R<5:0>						0000
RPOR1	06D2	—	—	RP3R<5:0>						—	—	RP2R<5:0>						0000
RPOR2	06D4	—	—	RP5R<5:0>						—	—	RP4R<5:0>						0000
RPOR3	06D6	—	—	RP7R<5:0>						—	—	RP6R<5:0>						0000
RPOR4	06D8	—	—	RP9R<5:0>						—	—	RP8R<5:0>						0000
RPOR5	06DA	—	—	RP11R<5:0>						—	—	RP10R<5:0>						0000
RPOR6	06DC	—	—	RP13R<5:0>						—	—	RP12R<5:0>						0000
RPOR7	06DE	—	—	RP15R<5:0>						—	—	RP14R<5:0>						0000
RPOR8	06E0	—	—	RP17R<5:0>						—	—	RP16R<5:0>						0000
RPOR9	06E2	—	—	RP19R<5:0>						—	—	RP18R<5:0>						0000
RPOR10	06E4	—	—	RP21R<5:0>						—	—	RP20R<5:0>						0000
RPOR11	06E6	—	—	RP23R<5:0>						—	—	RP22R<5:0>						0000
RPOR12	06E8	—	—	RP25R<5:0>						—	—	RP24R<5:0>						0000
RPOR13	06EA	—	—	RP27R<5:0>						—	—	RP26R<5:0>						0000
RPOR14	06EC	—	—	RP29R<5:0>						—	—	RP28R<5:0>						0000
RPOR16	06F0	—	—	RP33<5:0>						—	—	RP32<5:0>						0000
RPOR17	06F2	—	—	RP35<5:0>						—	—	RP34<5:0>						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 3-37: PORTA 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
TRISA	02C0	—	—	—	—	—	—	—	—	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	001F
PORTA	02C2	—	—	—	—	—	—	—	—	—	—	—	RA4	RA3	RA2	RA1	RA0	xxxx
LATA	02C4	—	—	—	—	—	—	—	—	—	—	—	LATA4	LATA3	LATA2	LATA1	LATA0	0000
ODCA	02C6	—	—	—	—	—	—	—	—	—	—	—	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	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: dsPIC33FJ06GS101 器件的 PORTB 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
TRISB	02C8	—	—	—	—	—	—	—	—	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	00FF
PORTB	02CA	—	—	—	—	—	—	—	—	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx
LATB	02CC	—	—	—	—	—	—	—	—	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000
ODCB	02CE	—	—	—	—	—	—	—	—	ODCB7	ODCB6	ODCB5	ODCB4	ODCB3	ODCB2	ODCB1	ODCB0	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: dsPIC33FJ06GS102、dsPIC33FJ06GS202、dsPIC33FJ16GS402、dsPIC33FJ16GS404、dsPIC33FJ16GS502 和 dsPIC33FJ16GS504 器件的 PORTB 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
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	0000
ODCB	02CE	ODCB15	ODCB14	ODCB13	ODCB12	ODCB11	ODCB10	ODCB9	ODCB8	ODCB7	ODCB6	ODCB5	ODCB4	ODCB3	ODCB2	ODCB1	ODCB0	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 3-40: dsPIC33FJ16GS404 和 dsPIC33FJ16GS504 器件的 PORTC 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
TRISC	02D0	—	—	TRISC13	TRISC12	TRISC11	TRISC10	TRISC9	TRISC8	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	3FFF
PORTC	02D2	—	—	RC13	RC12	RC11	RC10	RC9	RC8	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx
LATC	02D4	—	—	LATC13	LATC12	LATC11	LATC10	LATC9	LATC8	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	0000
ODCC	02D6	—	—	ODCC13	ODCC12	ODCC11	ODCC10	ODCC9	ODCC8	ODCC7	ODCC6	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0	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 3-41: 系统控制寄存器映射

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
RCON	0740	TRAPR	IOPUWR	—	—	—	—	CM	VREGS	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	xxxx ⁽¹⁾
OSCCON	0742	—	COSC<2:0>			—	NOSC<2:0>			CLKLOCK	IOLOCK	LOCK	—	CF	—	—	OSWEN	0300 ⁽²⁾
CLKDIV	0744	ROI	DOZE<2:0>			DOZEN	FRCDIV<2:0>			PLLPOST<1:0>		—	PLLPRE<4:0>					3040
PLLFBD	0746	—	—	—	—	—	—	—	PLLDIV<8:0>									0030
REFOCON	074E	ROON	—	ROSIDL	ROSEL	RODIV<3:0>				—	—	—	—	—	—	—	—	0000
OSCTUN	0748											TUN<5:0>						0000
ACLKCON	0750	ENAPLL	APLLCK	SELACLK	—	—	APSTSCCLR<2:0>			ASRCSEL	FRCSEL	—	—	—	—	—	—	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。

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

表 3-42: NVM 寄存器映射

文件寄存器名称	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 ⁽¹⁾
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。

- 注 1: 复位值仅表示 POR 时的状态值。其他复位状态时的值取决于复位时存储器写或者擦除操作的状态。

表 3-43: 仅 dsPIC33FJ06GS101 和 dsPIC33FJ06GS102 器件的 PMD 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
PMD1	0770	—	—	T3MD	T2MD	T1MD	—	PWMMD	—	I2C1MD	—	U1MD	—	SPI1MD	—	—	ADCMD	0000
PMD2	0772	—	—	—	—	—	—	IC2MD	IC1MD	—	—	—	—	—	—	OC2MD	OC1MD	0000
PMD3	0774	—	—	—	—	—	CMPMD	—	—	—	—	—	—	—	—	—	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	—	—	—	—	REFOMD	—	—	—	0000
PMD6	077A	—	—	—	—	PWM4MD	PWM3MD	PWM2MD	PWM1MD	—	—	—	—	—	—	—	—	0000
PMD7	077C	—	—	—	—	CMP4MD	CMP3MD	CMP2MD	CMP1MD	—	—	—	—	—	—	—	—	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 3-44: 仅 dsPIC33FJ06GS202 器件的 PMD 寄存器映射

SFR Name	SFR Addr	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	All Resets
PMD1	0770	—	—	—	T2MD	T1MD	—	PWMMD	—	I2C1MD	—	U1MD	—	SPI1MD	—	—	ADCMD	0000
PMD2	0772	—	—	—	—	—	—	—	IC1MD	—	—	—	—	—	—	—	OC1MD	0000
PMD3	0774	—	—	—	—	—	CMPMD	—	—	—	—	—	—	—	—	—	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	—	—	—	—	REFOMD	—	—	—	0000
PMD6	077A	—	—	—	—	—	—	PWM2MD	PWM1MD	—	—	—	—	—	—	—	—	0000
PMD7	077C	—	—	—	—	—	—	CMP2MD	CMP1MD	—	—	—	—	—	—	—	—	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 3-45: 仅 dsPIC33FJ16GS402 和 dsPIC33FJ16GS404 器件的 PMD 寄存器映射

SFR Name	SFR Addr	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	All Resets
PMD1	0770	—	—	T3MD	T2MD	T1MD	—	PWMMD	—	I2C1MD	—	U1MD	—	SPI1MD	—	—	ADCMD	0000
PMD2	0772	—	—	—	—	—	—	IC2MD	IC1MD	—	—	—	—	—	—	OC2MD	OC1MD	0000
PMD3	0774	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	—	—	—	—	REFOMD	—	—	—	0000
PMD6	077A	—	—	—	—	—	PWM3MD	PWM2MD	PWM1MD	—	—	—	—	—	—	—	—	0000
PMD7	077C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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 3-46: 仅 dsPIC33FJ16GS502 和 dsPIC33FJ16GS504 器件的 PMD 寄存器映射

SFR Name	SFR Addr	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	All Resets
PMD1	0770	—	—	T3MD	T2MD	T1MD	—	PWMMD	—	I2C1MD	—	U1MD	—	SPI1MD	—	—	ADCMD	0000
PMD2	0772	—	—	—	—	—	—	IC2MD	IC1MD	—	—	—	—	—	—	OC2MD	OC1MD	0000
PMD3	0774	—	—	—	—	—	CMPMD	—	—	—	—	—	—	—	—	—	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	—	—	—	—	REFOMD	—	—	—	0000
PMD6	077A	—	—	—	—	PWM4MD	PWM3MD	PWM2MD	PWM1MD	—	—	—	—	—	—	—	—	0000
PMD7	077C	—	—	—	—	CMP4MD	CMP3MD	CMP2MD	CMP1MD	—	—	—	—	—	—	—	—	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。

3.2.6 软件堆栈

除了用作工作寄存器外，dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。

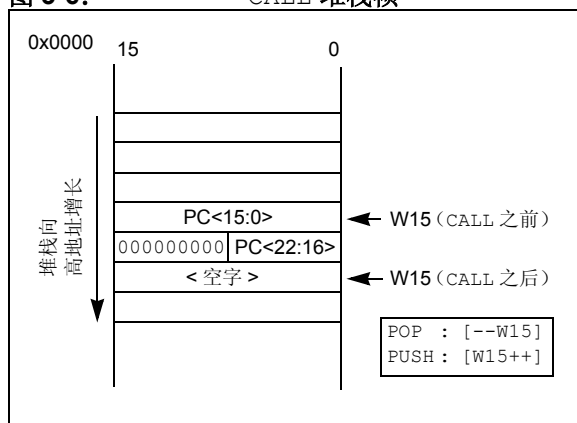
堆栈指针限制寄存器 (SPLIM) 与堆栈指针相关联，它设置堆栈地址上边界的值。复位时 SPLIM 不被初始化。与堆栈指针的情况一样，SPLIM<0> 被强制为 0，因为所有的堆栈操作必须是字对齐的。

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，不应紧跟着使用 W15 进行间接读操作的指令。

图 3-6: CALL 堆栈帧



3.2.7 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 3-1 中对 BSRAM 和 SSRAM SFR 的概括。

3.3 指令寻址模式

表 3-47 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单元中的前 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，取自数据存储单元或一个 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 3-47: 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，一些指令使用各种长度的立即数常量。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，操作数的来源或操作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）。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：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

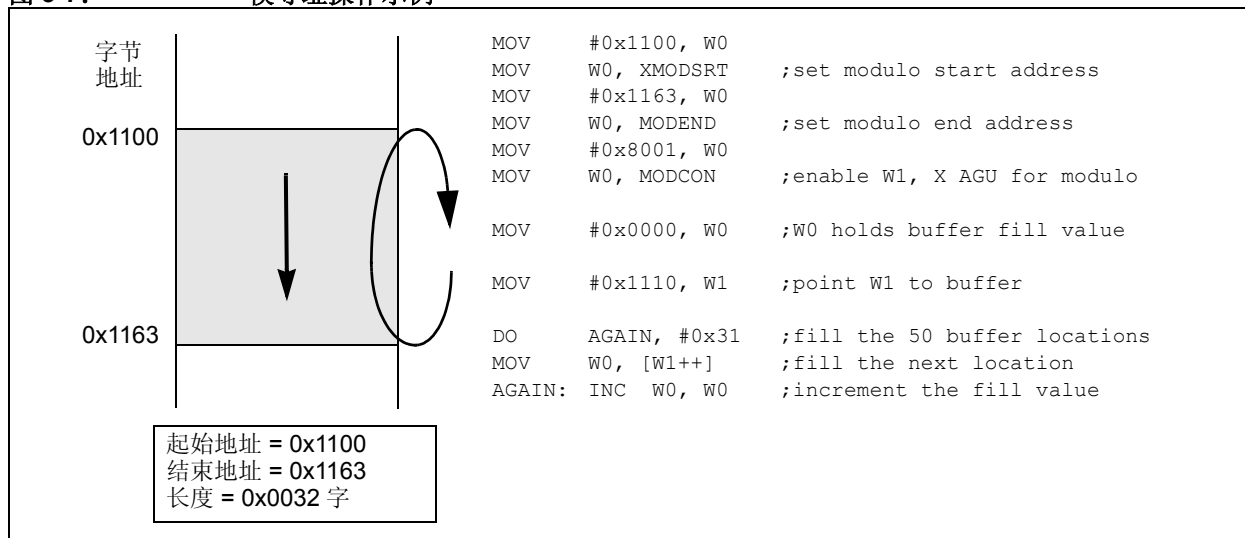
模寻址和位反转寻址控制寄存器 MODCON<15:0> 包含使能标志以及指定 W 地址寄存器的 W 寄存器字段。XWM 和 YWM 字段选择将对哪些寄存器进行模寻址：

- 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）由 MODCON<3:0> 确定（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。

要进行模寻址的 Y 地址空间指针 W 寄存器（YWM）由 MODCON<7:4> 确定。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

模寻址可以应用于与任何 **W** 寄存器相关的有效地址 (**EA**) 计算中。地址边界检查功能检查:

- 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边界地址上, 而且会检查地址是否大于上边界地址 (对于递增缓冲区)、是否小于下边界地址 (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回寄存器。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

当满足下列条件时, 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** 的计算都使用字长度数据 (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, 由于要求是字长度数据, **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。如果应用程序试图这么做的话, 对于 **X WAGU**, 位反转寻址将优先 (如果位反转寻址有效的话), **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

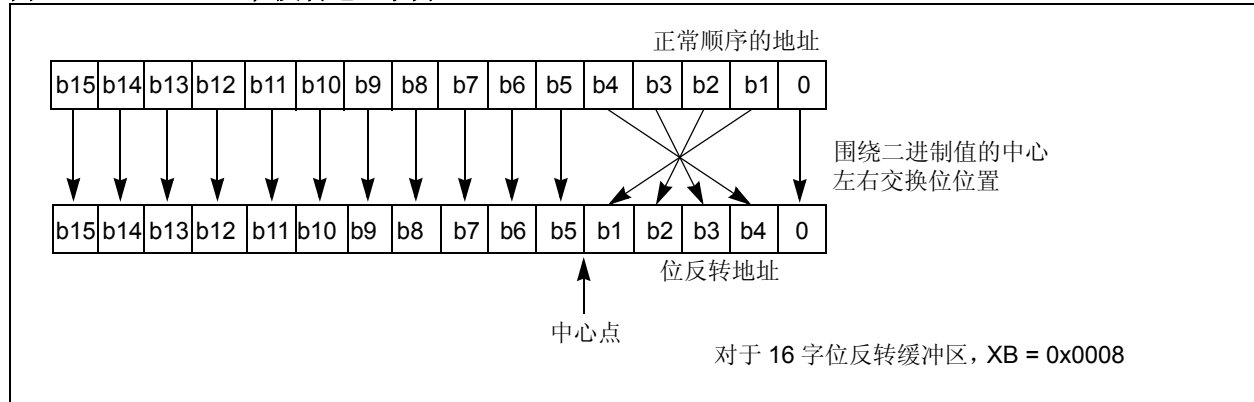


表 3-48: 位反转地址序列 (16 项)

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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 的架构采用 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，dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 的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。此功能也允许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

由于数据空间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。

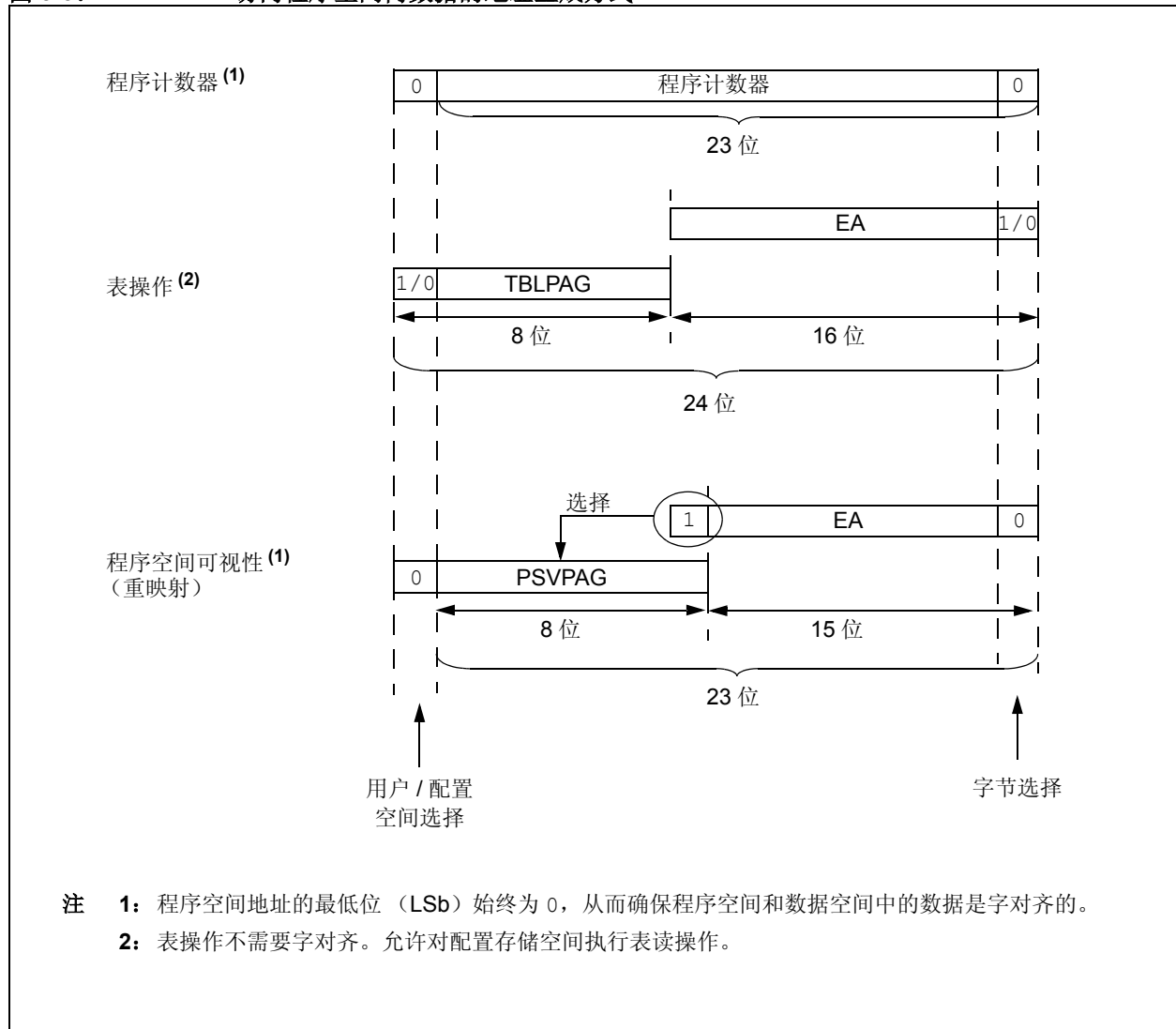
表 3-49 和图 3-9 显示了如何为表操作和重映射访问来从数据 EA 生成程序 EA。本文中，P<23:0> 指程序空间字，而 D<15:0> 指数据空间字。

表 3-49：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> ⁽¹⁾	
		0	xxxx xxxx		xxx xxxx xxxx xxxx	

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

图 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

TBLRD_L 和 TBLWT_L 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。TBLRD_H 和 TBLWT_H 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 16 位宽的字地址空间，它们并排放置，具有相同的地址范围。TBLRD_L 和 TBLWT_L 访问存放低位数据字的空间，而 TBLRD_H 和 TBLWT_H 则访问存有高位数据字节的空间。

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

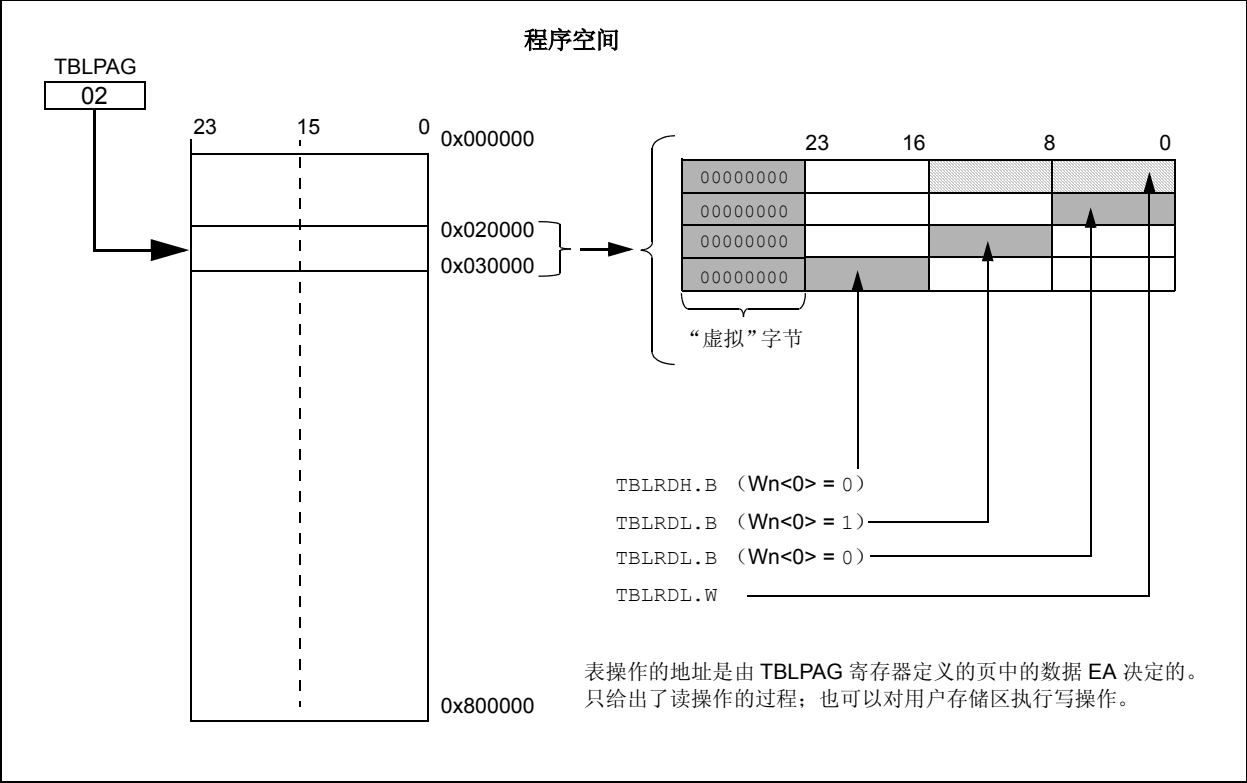
- TBLRD_L（表读低位字）：
 - 在字模式下，该指令将程序空间地址的低位字（P<15:0>）映射到数据地址（D<15:0>）中。

- 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。
- TBLRD_H（表读高位字）：
 - 在字模式下，该指令将程序地址的整个高位字（P<23:16>）映射到数据地址中。注意 D<15:8>（“虚拟字节”）总是为 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 TBLRD_L 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。

同样地，两条表指令 TBLWT_H 和 TBLWT_L 用于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



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 TBLRDH/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。

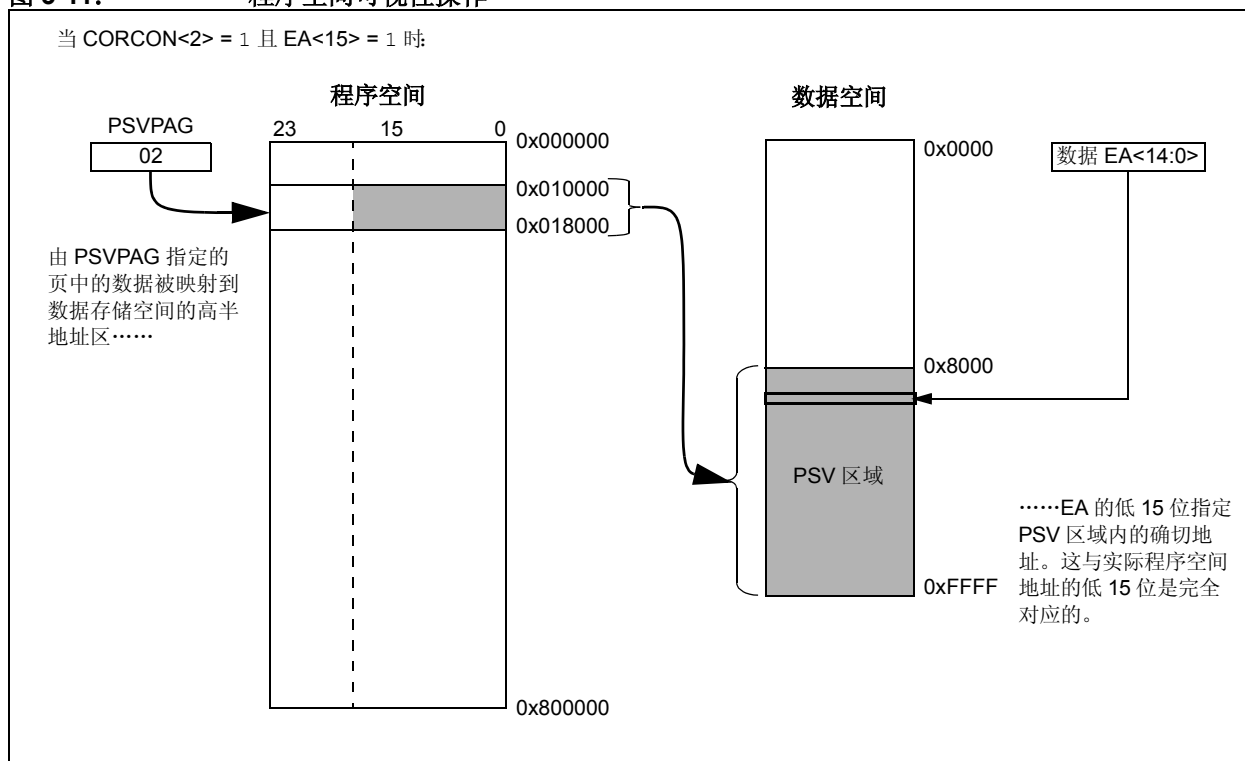
对于使用 PSV 而又在 REPEAT 循环外执行的操作，MOV 和 MOV.D 指令除了规定的执行时间之外，还需要一个额外的指令周期。所有其他指令都需要在规定的执行时间之外额外增加两个指令周期。

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



注:

4.0 闪存程序存储器

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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_{DD} 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（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 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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_{DD} ）、

接地线（ V_{SS} ）和主复位线（ $\overline{MCLR}$ ）。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，用户应用程序写程序存储数据时，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”）。

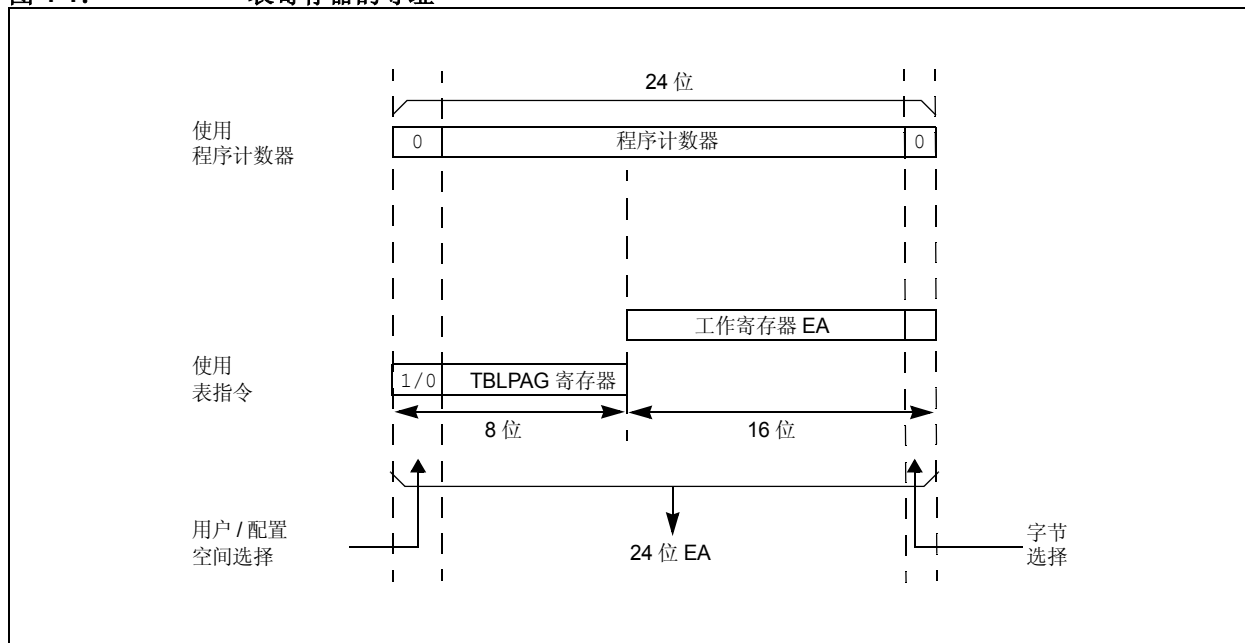
4.1 表指令和闪存编程

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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 23-12 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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 4-1: NVMCON: 闪存控制寄存器

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

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

图注:

R = 可读位

-n = POR 时的值

SO = 只可置 1 位

W = 可写位

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 ⁽²⁾
如果 ERASE = 1:
1111 = 存储器批量擦除操作
1101 = 擦除通用段
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
1101 = 无操作
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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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

bit 15-8未实现: 读为 0

bit 7-0NVMKEY<7:0>: 密钥寄存器位（只写）

4.4.1 闪存程序存储器的编程算法

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
; Init pointer to row to be ERASED
MOV    #tblpage(PROG_ADDR), W0
MOV    W0, TBLPAG
MOV    #tbloffset(PROG_ADDR), W0
TBLWTL W0, [W0]
DISI   #5

MOV    #0x55, W0
MOV    W0, NVMKEY
MOV    #0xAA, W1
MOV    W1, NVMKEY
BSET   NVMCON, #WR
NOP
NOP

```

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

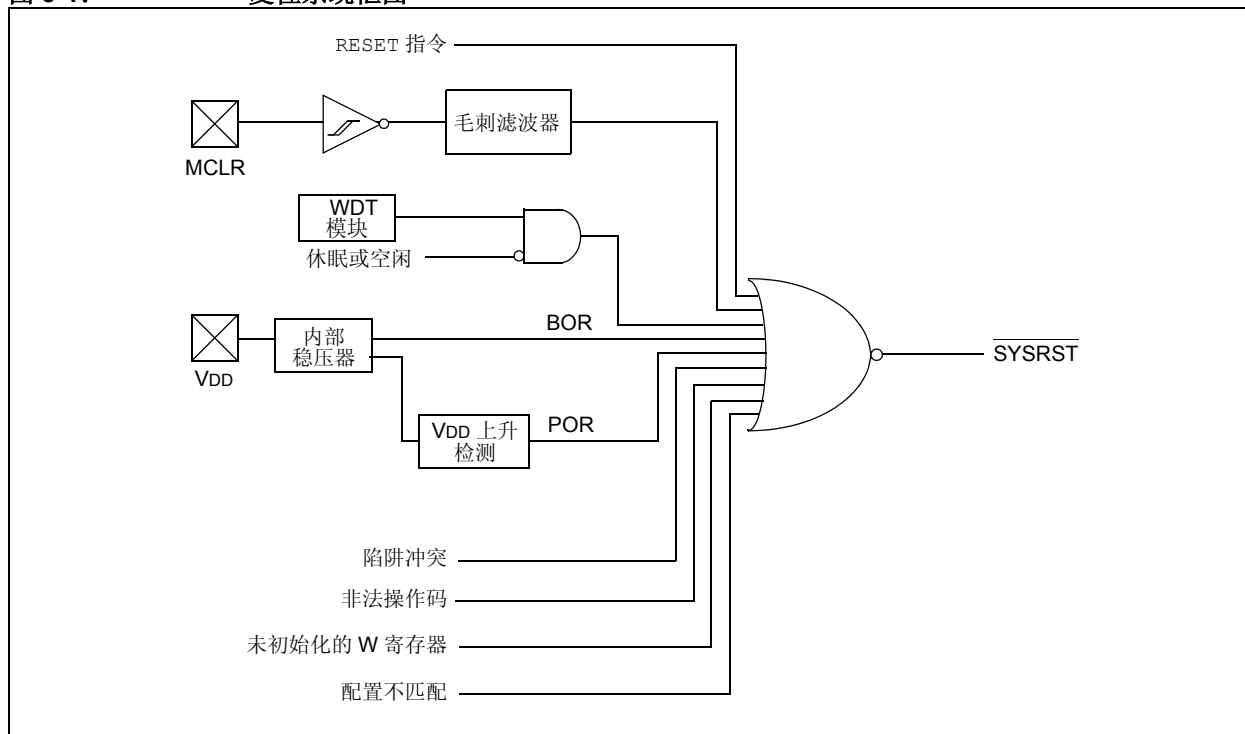
注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 系列器件的特性。但是不应把本手册当作无所不包的参考手册来使用。如需了解本数据手册的补充信息，请参考“dsPIC33F Family Reference Manual”的“Section 8. Reset” (DS70192)，该文档可从 Microchip 网站 (www.microchip.com) 下载。

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：软件 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。

图 5-1： 复位系统框图



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，请参见本数据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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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:** 外部复位引脚 (MCLR) 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⁽¹⁾ (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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 阈值并且 TPOR 延时已经结束。

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 阈值并且 TBOR 延时已经结束。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 延时确保系统电源稳定在适当电平供全速工作。TPWRT 延时结束后，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 7.0 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），则当系统时钟就绪且 TFSCM 延时结束时，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 ⁽¹⁾	—	—	TOSCD ⁽¹⁾
FRCPLL	TOSCD ⁽¹⁾	—	TLOCK ⁽³⁾	TOSCD + TLOCK ^(1,3)
XT	TOSCD ⁽¹⁾	TOST ⁽²⁾	—	TOSCD + TOST ^(1,2)
HS	TOSCD ⁽¹⁾	TOST ⁽²⁾	—	TOSCD + TOST ^(1,2)
EC	—	—	—	—
XTPLL	TOSCD ⁽¹⁾	TOST ⁽²⁾	TLOCK ⁽³⁾	TOSCD + TOST + TLOCK ^(1,2,3)
HSPLL	TOSCD ⁽¹⁾	TOST ⁽²⁾	TLOCK ⁽³⁾	TOSCD + TOST + TLOCK ^(1,2,3)
ECPLL	—	—	TLOCK ⁽³⁾	TLOCK ⁽³⁾
LPRC	TOSCD ⁽¹⁾	—	—	TOSCD ⁽¹⁾

- 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

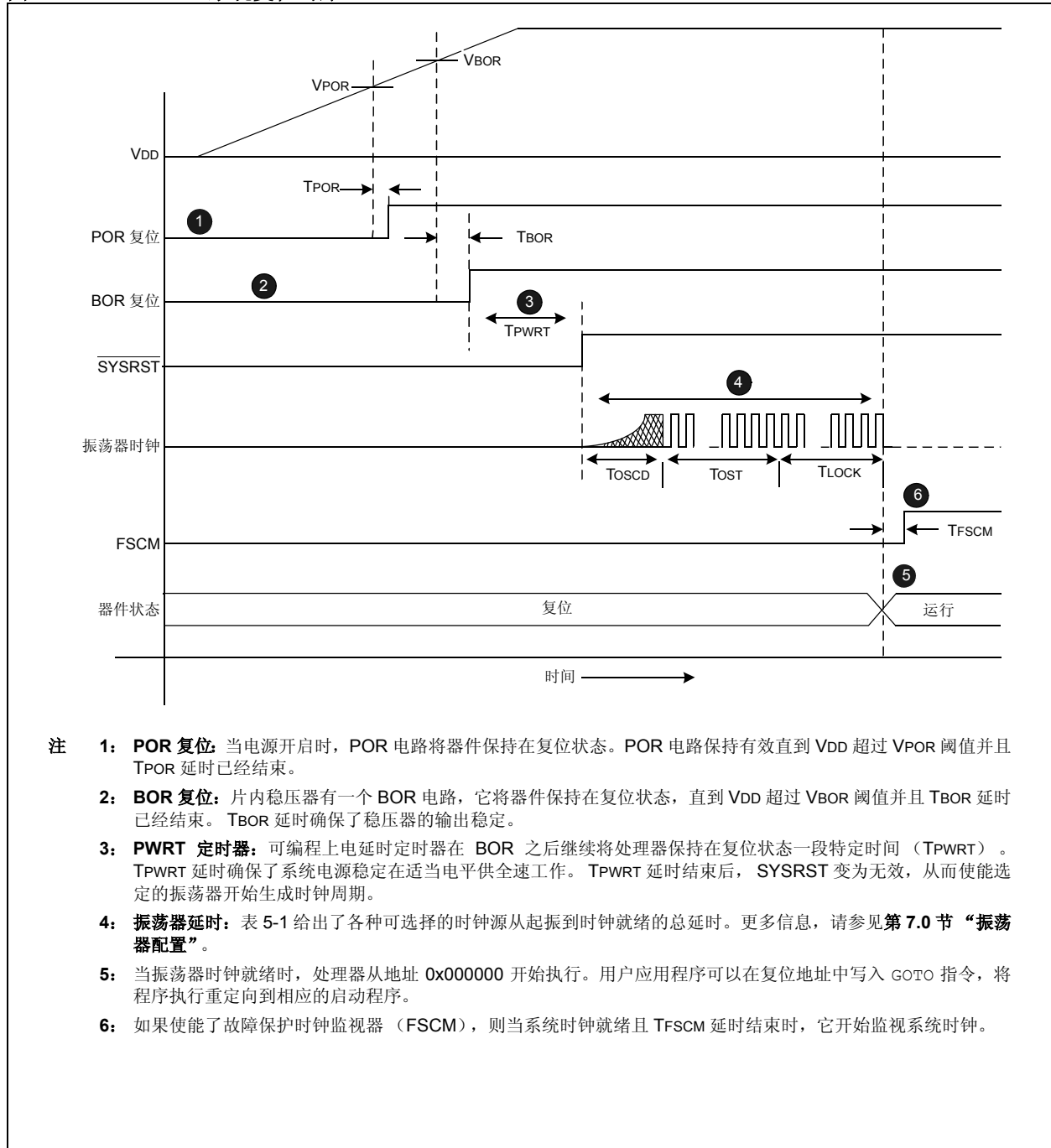


表 5-2: 振荡器延时

符号	参数	值
VPOR	POR 阈值	1.8V 标称值
TPOR	POR 延时时间	30 μ s 最大值
VBOR	BOR 阈值	2.5V 标称值
TBOR	BOR 延时时间	100 μ 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 μ s 最大值

注: 当器件退出复位状态（开始正常工作）时，器件工作参数（电压、频率和温度等）必须在其工作范围内，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 阈值并且 TPOR 延时已经结束。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。详细信息，请参见第 23.0 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 电路使器件保持在复位状态，直到 VDD 超过 VBOR 阈值且 TBOR 延时已结束。TBOR 延时确保了稳压器的输出稳定。

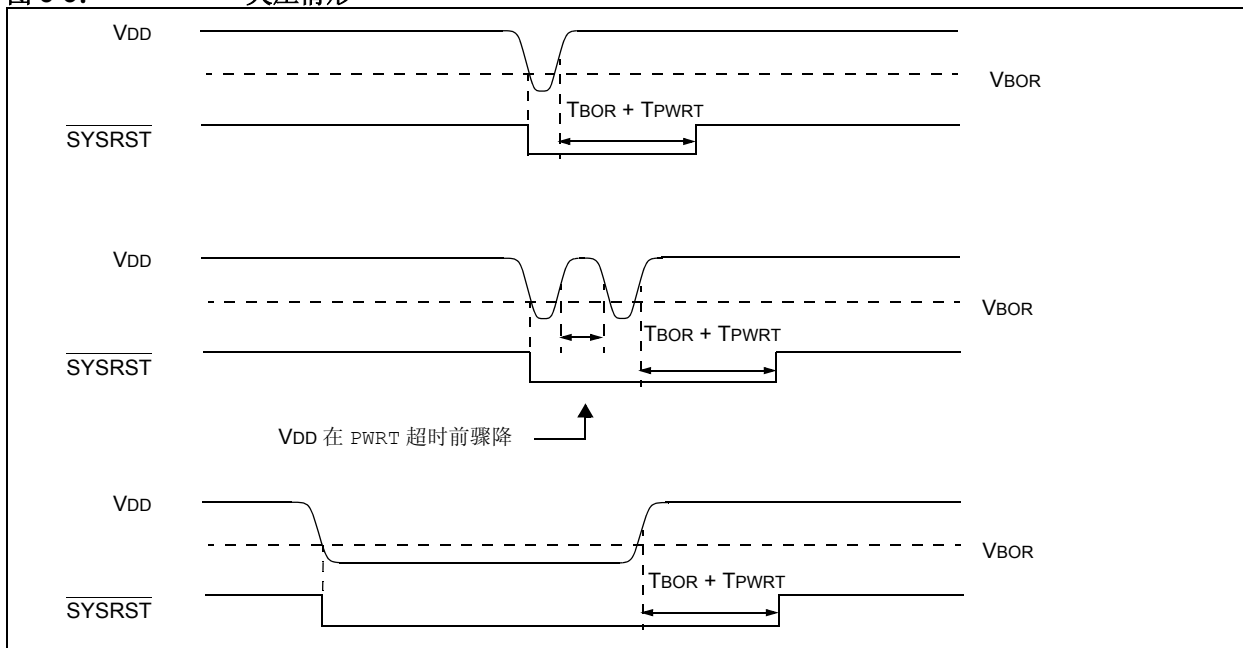
复位控制寄存器中的 BOR 状态位 BOR（RCON<1>）置 1，表示发生欠压复位。

BOR 之后器件不会全速工作，因为全速工作需要 VDD 上升至可接受的电平。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平上。

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 种设置（从 0 ms 到 128 ms）。更多详细信息，请参见第 20.0 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 23.0 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。

5.3.0.1 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。可将此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。

5.3.0.2 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 (SWR) (指令) 标志位 (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 (WDTO) 标志 (RCON<4>) 位置 1, 表示发生看门狗复位。关于看门狗复位的更多信息, 请参见第 20.4 节“看门狗定时器 (WDT)”。

5.6 陷阱冲突复位

如果正在处理某个高优先级的陷阱时, 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 (TRAPR) 标志 (RCON<15>) 位置 1, 表示发生陷阱冲突复位。关于陷阱冲突复位的更多信息, 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（CM）标志（RCON<9>）置 1，表示发生配置不匹配复位。关于配置不匹配复位的更多信息，请参见第 9.0 节“I/O 端口”。

注：配置不匹配复位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未初始化的 W 寄存器访问复位（IOPUWR）标志（RCON<14>）置 1，表示发生非法条件器件复位。

5.8.0.1 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段。要利用非法操作码复位，只能使用每个程序存储段的低 16 位存储数据值。高 8 位应该被编程为非法操作码值 3Fh。

5.8.0.2 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 寄存器阵列视作未初始化状态。

5.8.0.3 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重载时发生 VFC。

关于安全性复位的更多信息，请参见第 20.8 节“代码保护和 CodeGuard™ 安全”。

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 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 41. Interrupts (Part IV)” (DS70300)，该文档可从 Microchip 网站 (www.microchip.com) 下载。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 CPU 的中断请求。它具有以下特性：

- 多达 8 个处理器异常和软件陷阱
- 7 个可由用户选择的优先级
- 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

图 6-1 所示为中断向量表 (IVT)。IVT 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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件实现了多达 35 个唯一中断和 4 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。

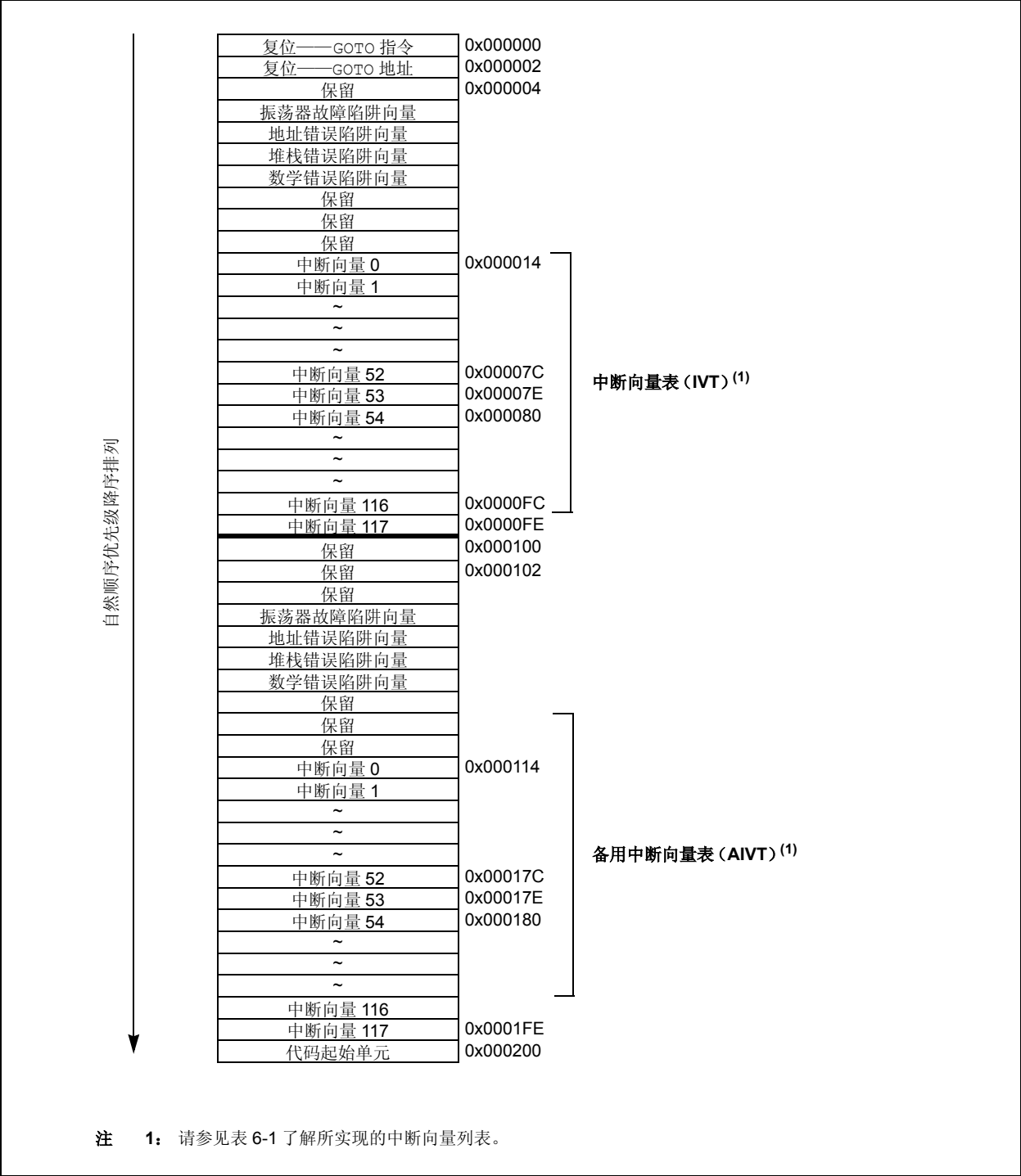
AIVT 通过提供一种不需要重新编程中断向量就可以在应用程序和支持环境之间切换的方法，来支持调试功能。此特性也支持运行时在应用程序之间切换以便评估各种不同的软件算法。如果不需要 AIVT，则应该用 IVT 中同样的地址编程 AIVT。

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，dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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: dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 的中断向量表



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 6-1: 中断向量

向量编号	中断请求 (IQR)	IVT 地址	AIVT 地址	中断源
最高自然顺序优先级				
8	0	0x000014	0x000114	INT0 —— 外部中断 0
9	1	0x000016	0x000116	IC2 —— 输入捕捉 1
10	2	0x000018	0x000118	OC1 —— 输出比较 1
11	3	0x00001A	0x00011A	T1 —— Timer1
12	4	0x00001C	0x00011C	保留
13	5	0x00001E	0x00011E	IC2 —— 输入捕捉 2
14	6	0x000020	0x000120	OC2 —— 输出比较 2
15	7	0x000022	0x000122	T2 —— Timer2
16	8	0x000024	0x000124	T3 —— Timer3
17	9	0x000026	0x000126	SPI1E —— SPI1 错误
18	10	0x000028	0x000128	SPI1 —— SPI1 传输完成
19	11	0x00002A	0x00012A	U1RX —— UART1 接收器
20	12	0x00002C	0x00012C	U1TX —— UART1 发送器
21	13	0x00002E	0x00012E	ADC —— ADC 组转换完成
22	14	0x000030	0x000130	保留
23	15	0x000032	0x000132	保留
24	16	0x000034	0x000134	SI2C1 —— I ² C1 从事件
25	17	0x000036	0x000136	MI2C1 —— I ² C1 主事件
26	18	0x000038	0x000138	CMP1 —— 模拟比较器 1 中断
27	19	0x00003A	0x00013A	CN —— 输入电平变化通知中断
28	20	0x00003C	0x00013C	INT1 —— 外部中断 1
29	21	0x00003E	0x00013E	保留
30	22	0x000040	0x000140	保留
31	23	0x000042	0x000142	保留
32	24	0x000044	0x000144	保留
33	25	0x000046	0x000146	保留
34	26	0x000048	0x000148	保留
35	27	0x00004A	0x00014A	保留
36	28	0x00004C	0x00014C	保留
37	29	0x00004E	0x00014E	INT2 —— 外部中断 2
38-64	30-56			保留
65	57	0x000086	0x000186	PSEM —— PWM 特殊事件匹配
66-72	58-64			保留
73	65	0x000096	0x000196	U1E —— UART1 错误中断
74-101	66-93			保留
102	94	0x0000D0	0x0001D0	PWM 1 —— PWM1 中断
103	95	0x0000D2	0x0001D2	PWM 2 —— PWM2 中断
104	96	0x0000D4	0x0001D4	PWM 3 —— PWM3 中断
105	97	0x0000D6	0x0001D6	PWM 4 —— PWM4 中断
106	98	0x0000D8	0x0001D8	保留
107	99	0x0000DA	0x0001DA	保留
108	100	0x0000DC	0x0001DC	保留
109	101	0x0000DE	0x0001DE	保留
110	102	0x0000E0	0x0001E0	保留
111	103	0x0000E2	0x0001E2	CMP2 —— 模拟比较器 2

表 6-1: 中断向量 (续)

向量编号	中断请求 (IQR)	IVT 地址	AIVT 地址	中断源
112	104	0x0000E4	0x0001E4	CMP3 —— 模拟比较器 3
113	105	0x0000E6	0x0001E6	CMP4 —— 模拟比较器 4
114	106	0x0000E8	0x0001E8	保留
115	107	0x0000EA	0x0001EA	保留
116	108	0x0000EC	0x0001EC	保留
117	109	0x0000EE	0x0001EE	保留
118	110	0x0000F0	0x0001F0	ADC 输入对 0 转换完成
119	111	0x0000F2	0x0001F2	ADC 输入对 1 转换完成
120	112	0x0000F4	0x0001F4	ADC 输入对 2 转换完成
121	113	0x0000F6	0x0001F6	ADC 输入对 3 转换完成
122	114	0x0000F8	0x0001F8	ADC 输入对 4 转换完成
123	115	0x0000FA	0x0001FA	ADC 输入对 5 转换完成
124	116	0x0000FC	0x0001FC	ADC 输入对 6 转换完成
125	117	0x0000FE	0x0001FE	保留
最低自然顺序优先级				

6.3 中断控制和状态寄存器

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件实现了 27 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 IFSx

IFSx 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 IECx

IECx 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 IPCx

IPCx 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、自然顺序优先级为 0 的外部中断。所以 INTOIF 位位于 IFS0<0>，INT0IE 位位于 IEC0<0>，INT0IP 位在 IPC0 的第一个位置（IPC0<2:0>）中。

6.3.6 状态 / 控制寄存器

尽管两个 CPU 控制寄存器不是中断控制硬件的特定组成部分，但它们仍包含控制中断功能的位。

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

在下面各页中的寄存器 6-1 到寄存器 6-35 描述了所有的中断寄存器。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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

图注:

C = 可清零位	R = 可读位	U = 未实现位, 读为 0
S = 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: 如需了解完整的寄存器详细信息, 请参见寄存器 2-1。
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。

寄存器 6-2: CORCON: 内核控制寄存器 ⁽¹⁾

U-0	U-0	U-0	U-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 ⁽²⁾	PSV	RND	IF
bit 7							bit 0

图注:

C = 可清零位			
R = 可读位	W = 可写位	-n = POR 时的值	1 = 置 1
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 ⁽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: 如需了解完整的寄存器详细信息, 请参见寄存器 2-2。
2: IPL3 位与 IPL<2:0> 位 (SR<7:5>) 组合形成 CPU 的中断优先级。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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	OVATE	OVATE
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
SFTACERR	DIV0ERR	—	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 = 禁止累加器 A 溢出陷阱
- bit 9 **OVATE:** 累加器 B 溢出陷阱允许位
1 = 允许累加器 B 溢出陷阱
0 = 禁止累加器 B 溢出陷阱
- 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 = 禁止累加器 A 或 B 的灾难性溢出陷阱
- 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 **未实现:** 读为 0
- 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
- bit 3 **ADDRERR:** 地址错误陷阱状态位
1 = 发生了地址错误陷阱
0 = 未发生地址错误陷阱

寄存器 6-3: INTCON1: 中断控制寄存器 1 （续）

- bit 2 **STKERR:** 堆栈错误陷阱状态位
 1 = 发生了堆栈错误陷阱
 0 = 未发生堆栈错误陷阱
- bit 1 **OSCFAIL:** 振荡器故障陷阱状态位
 1 = 发生了振荡器故障陷阱
 0 = 未发生振荡器故障陷阱
- bit 0 **未实现:** 读为 0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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 = DISI 指令执行中
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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-5: IFS0: 中断标志状态寄存器 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	ADIF	U1TXIF	U1RXIF	SPI1IF	SPI1EIF	T3IF ^(1,2)
bit 15							bit 8
R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IF ^(1,2)	OC2IF ^(1,2)	IC2IF	—	T1IF	OC1IF	IC1IF ⁽¹⁾	INT0IF
bit 7							bit 0

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

- bit 15-14 未实现: 读为 0
- bit 13 ADIF: ADC 组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 ^(1,2)
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 ^(1,2)
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 ^(1,2)
1 = 产生了中断请求
0 = 未产生中断请求
- bit 5 IC2IF: 输入捕捉通道 2 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 4 未实现: 读为 0
- bit 3 T1IF: Timer1 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求

注 1: dsPIC33FJ06GS101/102 器件未实现此位。
2: dsPIC33FJ06GS202 器件未实现这些位。

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⁾ 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

注 1: dsPIC33FJ06GS101/102 器件未实现此位。

2: dsPIC33FJ06GS202 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-6: IFS1: 中断标志状态寄存器 1

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	—	INT2IF	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT1IF	CNIF	AC1IF ⁽¹⁾	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 15-14

未实现: 读为 0
- 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 12-5

未实现: 读为 0
- bit 4

INT1IF: 外部中断 1 标志状态位

1 = 产生了中断请求

0 = 未产生中断请求
- bit 3

CNIF: 输入电平变化通知中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求
- bit 2

AC1IF: 模拟比较器 1 中断标志状态位 ⁽¹⁾

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

注 1: dsPIC33FJ16GS402/404 和 dsPIC33FJ06GS101/102 器件未实现此位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-7: IFS3: 中断标志状态寄存器 3

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

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-10 未实现: 读为 0

bit 9 **PSEMIF**: PWM 特殊事件匹配中断标志状态位

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 6-8: IFS4: 中断标志状态寄存器 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	U-0	R/W-0	U-0
—	—	—	—	—	—	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 15-2 未实现: 读为 0

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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-9: IFS5: 中断标志状态寄存器 5

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
PWM2IF ⁽¹⁾	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

PWM2IF: PWM2 中断标志状态位 ⁽¹⁾
1 = 产生了中断请求
0 = 未产生中断请求
- bit 14

PWM1IF: PWM1 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 13-0

未实现: 读为 0

注 1: dsPIC33FJ06GS101/102 器件未实现此位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-10: IFS6: 中断标志状态寄存器 6

R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
ADCP1IF	ADCP0IF	—	—	—	—	AC4IF ^(1,2)	AC3IF ^(1,2)
bit 15						bit 8	

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
AC2IF ⁽²⁾	—	—	—	—	—	PWM4IF ^(1,3)	PWM3IF ⁽⁴⁾
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **ADCP1IF:** ADC 输入对 1 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 14 **ADCP0IF:** ADC 输入对 0 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 13-10 **未实现:** 读为 0

bit 9 **AC4IF:** 模拟比较器 4 中断标志状态位 ^(1,2)

1 = 产生了中断请求

0 = 未产生中断请求

bit 8 **AC3IF:** 模拟比较器 3 中断标志状态位 ^(1,2)

1 = 产生了中断请求

0 = 未产生中断请求

bit 7 **AC2IF:** 模拟比较器 2 中断标志状态位 ⁽²⁾

1 = 产生了中断请求

0 = 未产生中断请求

bit 6-2 **未实现:** 读为 0

bit 1 **PWM4IF:** PWM4 中断标志状态位 ^(1,3)

1 = 产生了中断请求

0 = 未产生中断请求

bit 0 **PWM3IF:** PWM3 中断标志状态位 ⁽⁴⁾

1 = 产生了中断请求

0 = 未产生中断请求

注 1: dsPIC33FJ06GS202 器件未实现这些位。

2: dsPIC33FJ06GS101 和 dsPIC33FJ16GS502 器件未实现这些位。

3: dsPIC33FJ16GS402/404/502 器件未实现这些位。

4: dsPIC33FJ06101/102/202 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-11: IFS7: 中断标志状态寄存器 7

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	ADCP6IF	ADCP5IF ⁽¹⁾	ADCP4IF ⁽¹⁾	ADCP3IF ⁽²⁾	ADCP2IF ⁽³⁾
bit 7							bit 0

图注:							
R = 可读位		W = 可写位		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 **ADCP6IF:** ADC 输入对 6 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 3 **ADCP5IF:** ADC 输入对 5 转换完成中断标志状态位 ⁽¹⁾
1 = 产生了中断请求
0 = 未产生中断请求
- bit 2 **ADCP4IF:** ADC 输入对 4 转换完成中断标志状态位 ⁽¹⁾
1 = 产生了中断请求
0 = 未产生中断请求
- bit 1 **ADCP3IF:** ADC 输入对 3 转换完成中断标志状态位 ⁽²⁾
1 = 产生了中断请求
0 = 未产生中断请求
- bit 0 **ADCP2IF:** ADC 输入对 2 转换完成中断标志状态位 ⁽³⁾
1 = 产生了中断请求
0 = 未产生中断请求

注 1: dsPIC33FJ06GS101/102/202 和 dsPIC33FJ16GS402/404/502 器件未实现这些位。
2: dsPIC33FJ06GS102/202 器件未实现此位。
3: dsPIC33FJ06GS101 器件未实现此位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-12: IEC0: 中断允许控制寄存器 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	ADIE	U1TXIE	U1RXIE	SPI1IE	SPI1EIE	T3IE ^(1,2)
bit 15							bit 8
R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IE	OC2IE ^(1,2)	IC2IE ^(1,2)	—	T1IE	OC1IE	IC1IE ⁽¹⁾	INT0IE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14	未实现: 读为 0
bit 13	ADIE: 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 ^(1,2) 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 ^(1,2) 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 ^(1,2) 1 = 允许中断请求 0 = 禁止中断请求
bit 4	未实现: 读为 0
bit 3	T1IE: Timer1 中断允许位 1 = 允许中断请求 0 = 禁止中断请求

注 1: dsPIC33FJ06GS101/102 器件未实现这些位。

2: dsPIC33FJ06GS202 器件未实现这些位。

寄存器 6-12: 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⁾ 1 = 允许中断请求 0 = 禁止中断请求
bit 0	INT0IE: 外部中断 0 允许位 1 = 允许中断请求 0 = 禁止中断请求

注 1: dsPIC33FJ06GS101/102 器件未实现这些位。
2: dsPIC33FJ06GS202 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-13: IEC1: 中断允许控制寄存器 1

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	—	INT2IE	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT1IE	CNIE	AC1IE ⁽¹⁾	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 15-14 未实现: 读为 0

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 12-5 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

bit 2 **AC1IE:** 模拟比较器 1 中断允许位 ⁽¹⁾

1 = 允许中断请求

0 = 禁止中断请求

bit 1 **MI2C1IE:** I2C1 主事件中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 0 **SI2C1IE:** I2C1 从事件中断允许位

1 = 允许中断请求

0 = 禁止中断请求

注 1: dsPIC33FJ06GS101/102 和 dsPIC33FJ16GS402/404 器件未实现此位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-14: IEC3: 中断允许控制寄存器 3

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
—	—	—	—	—	—	PSEMIE	—
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15-10 未实现: 读为 0
bit 9 PSEMIE: PWM 特殊事件匹配中断允许位
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 6-15: IEC4: 中断允许控制寄存器 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	U-0	R/W-0	U-0
—	—	—	—	—	—	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 15-2 未实现: 读为 0
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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-16: IEC5: 中断允许控制寄存器 5

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
PWM2IE ⁽¹⁾	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 **PWM2IE:** PWM2 中断允许位 ⁽¹⁾

1 = 允许中断请求

0 = 禁止中断请求

bit 14 **PWM1IE:** PWM1 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 13-0 **未实现:** 读为 0

注 1: dsPIC33FJ06GS101/102 器件未实现此位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-17: IEC6: 中断允许控制寄存器 6

R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
ADCP1IE	ADCP0IE	—	—	—	—	AC4IE ^(1,2)	AC3IE ^(1,2)
bit 15						bit 8	

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
AC2IE ⁽²⁾	—	—	—	—	—	PWM4IE ^(1,3)	PWM3IE ⁽⁴⁾
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **ADCP1IE:** ADC 输入对 1 转换完成中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 14 **ADCP0IE:** ADC 输入对 0 转换完成中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 13-10 **未实现:** 读为 0

bit 9 **AC4IE:** 模拟比较器 4 中断允许位 ^(1,2)

1 = 允许中断请求

0 = 禁止中断请求

bit 8 **AC3IE:** 模拟比较器 3 中断允许位 ^(1,2)

1 = 允许中断请求

0 = 禁止中断请求

bit 7 **AC2IE:** 模拟比较器 2 中断允许位 ⁽²⁾

1 = 允许中断请求

0 = 禁止中断请求

bit 6-2 **未实现:** 读为 0

bit 1 **PWM4IE:** PWM4 中断允许位 ^(1,3)

1 = 允许中断请求

0 = 禁止中断请求

bit 0 **PWM3IE:** PWM3 中断允许位 ⁽⁴⁾

1 = 允许中断请求

0 = 禁止中断请求

注 1: dsPIC33FJ06GS202 器件未实现这些位。

2: dsPIC33FJ06GS101 和 dsPIC33FJ16GS502 器件未实现这些位。

3: dsPIC33FJ16GS402/404/502 器件未实现这些位。

4: dsPIC33FJ06101/102/202 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-18: IEC7: 中断允许控制寄存器 7

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

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	ADCP6IE ⁽³⁾	ADCP5IE ⁽¹⁾	ADCP4IE ⁽¹⁾	ADCP3IE ⁽²⁾	ADCP2IE ⁽³⁾
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

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 **ADCP6IE:** ADC 输入对 6 转换完成中断允许位 ⁽³⁾

1 = 允许中断请求

0 = 禁止中断请求

bit 3 **ADCP5IE:** ADC 输入对 5 转换完成中断允许位 ⁽¹⁾

1 = 允许中断请求

0 = 禁止中断请求

bit 2 **ADCP4IE:** ADC 输入对 4 转换完成中断允许位 ⁽¹⁾

1 = 允许中断请求

0 = 禁止中断请求

bit 1 **ADCP3IE:** ADC 输入对 3 转换完成中断允许位 ⁽²⁾

1 = 允许中断请求

0 = 禁止中断请求

bit 0 **ADCP2IE:** ADC 输入对 2 转换完成中断允许位 ⁽³⁾

1 = 允许中断请求

0 = 禁止中断请求

注 1: dsPIC33FJ06GS101/102/202 和 dsPIC33FJ16GS402/404/502 器件未实现这些位。

2: dsPIC33FJ06GS102/202 器件未实现此位。

3: dsPIC33FJ06GS101 器件未实现此位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-19: **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> ⁽¹⁾			—	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 ⁽¹⁾
 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

注 1: dsPIC33FJ06GS101/102 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-20: 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> ⁽¹⁾		
bit 15				bit 8			

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	IC2IP<2:0> ^(1,2)			—	—	—	—
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **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 ⁽¹⁾
 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 ^(1,2)
 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

注 1: dsPIC33FJ06GS101/202 器件未实现这些位。

注 2: dsPIC33FJ06GS102 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-21: 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> ⁽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 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 ⁽¹⁾
 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: dsPIC33FJ06GS101/102/202 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-22: IPC3: 中断优先级控制寄存器 3

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	ADIP<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 15-7 未实现: 读为 0

bit 6-4 **ADIP<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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-23: 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>			—	AC1IP<2:0> ⁽¹⁾		
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 **AC1IP<2:0>**: 模拟比较器 1 中断优先级位 ⁽¹⁾
 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: dsPIC33FJ06GS101/102 和 dsPIC33FJ16GS402/404 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-24: IPC5: 中断优先级控制寄存器 5

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15						bit 8	
U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT1IP<2:0>		
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-3 未实现: 读为 0

bit 2-0 **INT1IP<2:0>**: 外部中断 1 优先级位
 111 = 中断优先级为 7 (最高优先级中断)
 .
 .
 .
 001 = 中断优先级为 1
 000 = 禁止中断源

寄存器 6-25: IPC7: 中断优先级控制寄存器 7

U-0	U-1	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	INT2IP<2:0>			—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-7 未实现: 读为 0

bit 6-4 **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 3-0 未实现: 读为 0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	PSEMIP<2:0>			—	—	—	—
bit 7							bit 0

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

bit 15-7 未实现: 读为 0
bit 6-4 **PSEMIP<2:0>**: PWM 特殊事件匹配中断优先级位
 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: IPC16: 中断优先级控制寄存器 16

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

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	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 15-7 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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-28: IPC23: 中断优先级控制寄存器 23

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	PWM2IP ⁽¹⁾			—	PWM1IP<2:0>		
bit 15				bit 8			

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **未实现:** 读为 0

bit 14-12 **PWM2IP<2:0>:** PWM2 中断优先级位 ⁽¹⁾

111 = 中断优先级为 7 (最高优先级)

•

•

•

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 **PWM1IP<2:0>:** PWM1 中断优先级位

111 = 中断优先级为 7 (最高优先级)

•

•

•

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: dsPIC33FJ06GS101/102 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-29: IPC24: 中断优先级控制寄存器 24

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	PWM4IP ⁽¹⁾			—	PWM3IP<2:0> ⁽²⁾		
bit 7							bit 0

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

bit 15-7 未实现：读为 0

bit 6-4 **PWM4IP<2:0>**: PWM4 中断优先级位 ⁽¹⁾

 111 = 中断优先级为 7 （最高优先级）

 •

 •

 •

 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 **PWM3IP<2:0>**: PWM3 中断优先级位 ⁽²⁾

 111 = 中断优先级为 7 （最高优先级）

 •

 •

 •

 001 = 中断优先级为 1

 000 = 禁止中断源

注 1: dsPIC33FJ06GS202 和 dsPIC33FJ16GS402/404 器件未实现这些位。

 2: dsPIC33FJ06101/102/202 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-30: **IPC25: 中断优先级控制寄存器 25**

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	AC2IP<2:0> ⁽¹⁾			—	—	—	—
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 **AC2IP<2:0>:** 模拟比较器 2 中断优先级位 ⁽¹⁾

 111 = 中断优先级为 7 (最高优先级)

 •

 •

 •

 001 = 中断优先级为 1

 000 = 禁止中断源

bit 11-01 **未实现:** 读为 0

注 1: dsPIC33FJ06GS101/102 和 dsPIC33FJ16GS402/404 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-31: IPC26: 中断优先级控制寄存器 26

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	AC4IP<2:0> ⁽¹⁾			—	AC3IP<2:0> ^(1,2)		
bit 7							bit 0

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

bit 15-7 未实现：读为 0

bit 6-4 **AC4IP<2:0>**：模拟比较器 4 中断优先级位 ⁽¹⁾

 111 = 中断优先级为 7 （最高优先级）

 •

 •

 •

 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 **AC3IP<2:0>**：模拟比较器 3 中断优先级位 ^(1,2)

 111 = 中断优先级为 7 （最高优先级）

 •

 •

 •

 001 = 中断优先级为 1

 000 = 禁止中断源

注 1: dsPIC33FJ06GS202 和 dsPIC33FJ16GS402/404 器件未实现这些位。

 2: dsPIC33FJ06GS101/102 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-32: IPC27: 中断优先级控制寄存器 27

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	ADCP1IP<2:0>			—	ADCP0IP<2:0>		
bit 15				bit 8			

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **ADCP1IP<2:0>**: ADC 输入对 1 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 11 未实现: 读为 0

bit 10-8 **ADCP0IP<2:0>**: ADC 输入对 0 转换完成中断优先级位 ⁽¹⁾

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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-33: IPC28: 中断优先级控制寄存器 28

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	ADCP5IP<2:0> ⁽⁴⁾			—	ADCP4IP<2:0> ⁽⁴⁾		
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
—	ADCP3IP<2:0> ^(2,3)			—	ADCP2IP<2:0> ⁽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	未实现: 读为 0
bit 14-12	ADCP5IP<2:0> : ADC 输入对 5 转换完成中断优先级位 ⁽⁴⁾ 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	ADCP4IP<2:0> : ADC 输入对 4 转换完成中断优先级位 ⁽⁴⁾ 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	ADCP3IP<2:0> : ADC 输入对 3 转换完成中断优先级位 ^(2,3) 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	ADCP2IP<2:0> : ADC 输入对 2 转换完成中断优先级位 ⁽¹⁾ 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: dsPIC33FJ06GS101 器件未实现这些位。
 2: dsPIC33FJ06GS102 器件未实现这些位。
 3: dsPIC33FJ06GS202 器件未实现这些位。
 4: 仅 dsPIC33FJ16GS504 器件实现了这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-34: IPC29: 中断优先级控制寄存器 29

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

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	ADCP6IP<2:0> ⁽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 15-3 未实现: 读为 0

bit 2-0 **ADCP6IP<2:0>**: ADC 输入对 6 转换完成中断优先级位 ⁽¹⁾

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: dsPIC33FJ06GS202 器件未实现这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 6-35: 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<3:0>: 新的 CPU 中断优先级位

1111 = CPU 中断优先级为 15

•

•

•

0001 = CPU 中断优先级为 1

0000 = CPU 中断优先级为 0
- bit 7

未实现: 读为 0
- bit 6-0

VECNUM<6:0>: 待处理中断向量编号位

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需完成下列步骤：

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

下列步骤概括了禁止所有用户中断的过程：

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

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。

注:

7.0 振荡器配置

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 42. Oscillator (Part IV)” (DS70307)，该文档可从 Microchip 网站 (www.microchip.com) 下载。

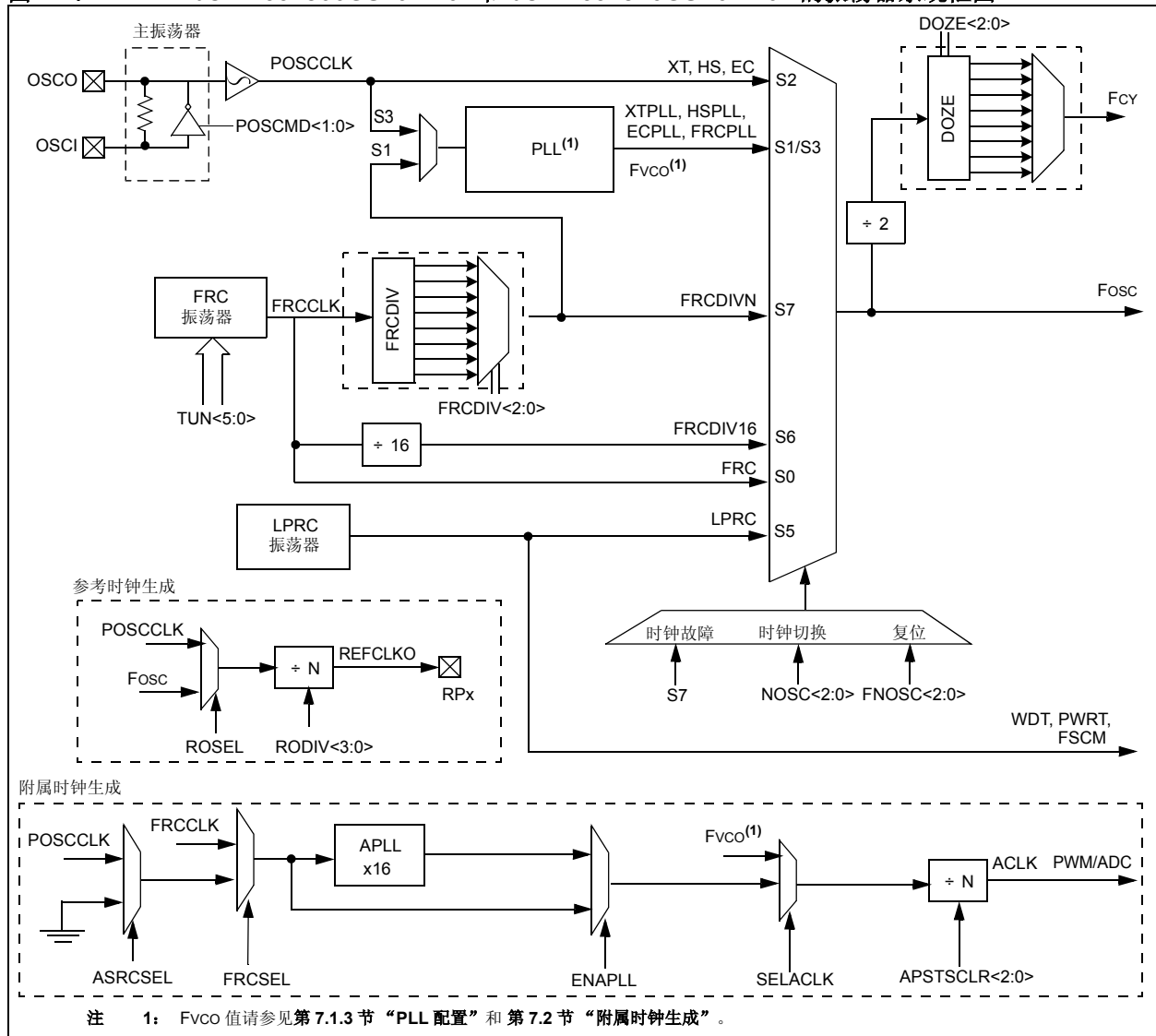
dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 的振荡器系统提供：

- 可选择多种外部和内部振荡器作为时钟源

- 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
- 用于 ADC 和 PWM 的附属 PLL

图 7-1 给出了振荡器系统的简化框图。

图 7-1: dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 的振荡器系统框图



7.1 CPU 时钟系统

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件提供了 6 种系统时钟选择：

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

7.1.1 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：

- XT（晶振）：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。
- HS（高速晶振）：10 MHz 至 40 MHz 范围内的晶振。晶振连接在 OSC1 和 OSC2 引脚之间。
- EC（外部时钟）：外部时钟信号直接施加到 OSC1 引脚。

LPRC 内部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 7.1.3 节“PLL 配置”中讲述。

7.1.2 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 20.1 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 7-1 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，dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 按公式 7-1 计算。

公式 7-1： 器件工作频率

Fcy = Fosc/2

表 7-1： 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
保留	保留	xx	100	—
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。

7.1.3 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 7-2 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 (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 ($PLLFBF<8:0>$) 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} 按公式 7-2 计算:

公式 7-2: F_{osc} 计算

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

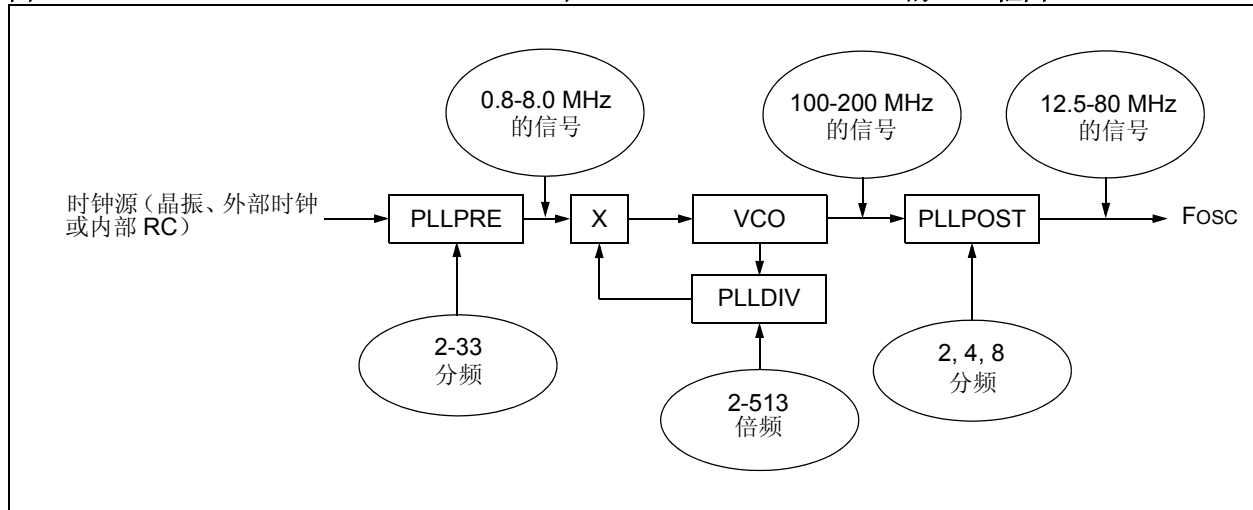
例如, 假设正在使用 10 MHz 晶振, 所选的振荡器模式为带 PLL 的 XT (见公式 7-3)。

- 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 7-3: 带 PLL 的 XT 模式示例

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

图 7-2: dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 的 PLL 框图



7.2 附属时钟生成

附属时钟生成可用于需要在与系统时钟无关的频率下工作的外设, 例如 PWM 或 ADC。

注: 为获得 1.04 ns 的 PWM 分辨率, 必须将附属时钟设置为 120 MHz。

主振荡器和内部 FRC 振荡器源可与附属 PLL 配合使用来获得附属时钟。附属 PLL 具有固定的 16 倍频因子。

注: 如果使用了主 PLL 作为附属时钟的时钟源, 那么应将主 PLL 配置为小于等于 30 MIPS 的工作速度。

7.3 参考时钟生成

参考时钟输出逻辑为用户提供基于系统时钟或器件引脚上外接的晶振输出时钟信号的功能。用户应用程序可指定范围较宽的时钟频率缩放比例来产生参考时钟输出。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 7-1: **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	U-0	R/W-0
CLKLOCK	IOLOCK	LOCK	—	CF	—	—	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 = 保留

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 = 保留

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, 起振定时器在进行延时或 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 已检测到时钟故障

0 = FSCM 未检测到时钟故障

bit 2-1 未实现: 读为 0

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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 7-2: CLKDIV: 时钟分频比寄存器

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
ROI	DOZE<2:0>			DOZEN ⁽¹⁾	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:

R = 可读位

W = 可写位

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⁽¹⁾

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 被 1 分频 (默认)
001 = FRC 被 2 分频
010 = FRC 被 4 分频
011 = FRC 被 8 分频
100 = FRC 被 16 分频
101 = FRC 被 32 分频
110 = FRC 被 64 分频
111 = FRC 被 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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 7-3: 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
—	—	—	—	—	—	—	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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 7-4: 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% (8.23 MHz)

011110 = 中心频率 +11.25% (8.20 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)

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 7-5: **ACLKCON: 附属时钟分频器控制寄存器**

R/W-0	R-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
ENAPLL	APLLCK	SELACLK	—	—	APSTSCLR<2:0>		
bit 15							bit 0

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
ASRCSEL	FRCSEL	—	—	—	—	—	—
bit 7							

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

- bit 15

ENAPLL: 附属 PLL 使能位
1 = 使能 APLL
0 = 禁止 APLL
- bit 14

APLLCK: APLL 锁定状态位 (只读)
1 = 表明附属 PLL 处于锁定状态
0 = 表明附属 PLL 未处于锁定状态
- bit 13

SELACLK: 为附属时钟分频器选择附属时钟源位
1 = 附属振荡器为附属时钟分频器提供时钟源
0 = 主 PLL (Fvco) 为附属时钟分频器提供时钟源
- bit 12-11

未实现: 读为 0
- bit 10-8

APSTSCLR<2:0>: 附属时钟输出分频比位
111 = 被 1 分频
110 = 被 2 分频
101 = 被 4 分频
100 = 被 8 分频
011 = 被 16 分频
010 = 被 32 分频
001 = 被 64 分频
000 = 被 256 分频
- bit 7

ASRCSEL: 为附属时钟选择参考时钟源位
1 = 主振荡器是时钟源
0 = 未选择时钟输入
- bit 6

FRCSEL: 为附属 PLL 选择参考时钟源位
1 = 为附属 PLL 选择 FRC 时钟
0 = 输入时钟源由 ASRCSEL 位的设置决定
- bit 5-0

未实现: 读为 0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 7-6: REFOCON: 参考振荡器控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROON	—	ROSIDL	ROSEL	RODIV<3:0>			
bit 15				bit 8			
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **ROON:** 参考振荡器输出使能位
1 = 使能 REFO 引脚上的参考振荡器输出
0 = 禁止参考振荡器输出
- bit 14 **未实现:** 读为 0
- bit 13 **ROSIDL:** 参考振荡器在休眠模式下运行情况位
1 = 参考振荡器输出在休眠模式下继续运行
0 = 参考振荡器输出在休眠模式下禁止
- bit 12 **ROSEL:** 参考振荡器源选择位
1 = 使用晶振作为参考时钟
0 = 使用系统时钟作为参考时钟
- bit 11-8 **RODIV<3:0>:** 参考振荡器分频比位
1111 = 参考时钟被 32,768 分频
1110 = 参考时钟被 16,384 分频
1101 = 参考时钟被 8,192 分频
1100 = 参考时钟被 4,096 分频
1011 = 参考时钟被 2,048 分频
1010 = 参考时钟被 1,024 分频
1001 = 参考时钟被 512 分频
1000 = 参考时钟被 256 分频
0111 = 参考时钟被 128 分频
0110 = 参考时钟被 64 分频
0101 = 参考时钟被 32 分频
0100 = 参考时钟被 16 分频
0011 = 参考时钟被 8 分频
0010 = 参考时钟被 4 分频
0001 = 参考时钟被 2 分频
0000 = 参考时钟
- bit 7-0 **未实现:** 读为 0

7.4 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，dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。

7.4.1 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 20.1 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。

7.4.2 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。

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。

7.5 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。

8.0 节能特性

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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驱动电路的数目可降低功耗。dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件可通过如下四种不同的方式管理功耗：

- 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。

8.1 时钟频率和时钟切换

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 7.0 节“**振荡器配置**”中进行更详细的讨论。

8.2 基于指令的节能模式

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 暂停工作并暂停代码执行，但是允许外设模块继续工作。例 8-1 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”。

8.2.1 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。这包括 I/O 端口上的输入电平变化通知功能或使用外部时钟输入的外设等。
- 任何需要使用系统时钟源来工作的外设将在休眠模式下都被禁止。

当发生以下任何事件时，器件将被从休眠模式唤醒：

- 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 8-1: PWRSAV 指令语法

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

8.2.2 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 8.4 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，器件将被从空闲模式唤醒：

- 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。

8.2.3 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。

8.3 打盹模式

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。但在有些情况下可能不可行。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停止通信。

打盹模式是另一种简单有效的节能方法，它可以在器件仍然执行代码的情况下降低功耗。在此模式下，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，并已基于这一工作速度将 CAN 模块的速度配置为 500 kbps。如果将器件置于时钟频率比为 1:4 的打盹模式下，那么 CAN 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。

8.4 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将不起作用，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® DSC 器件支持外设时，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。

注： 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）。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 8-1: **PMD1: 外设模块禁止控制寄存器 1**

U-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	—	T3MD	T2MD	T1MD	—	PWMMD	—
bit 15						bit 8	

R/W-0	U-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0
I2C1MD	—	U1MD	—	SPI1MD	—	—	ADCMD
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

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 **T3MD:** Timer3 模块禁止位
1 = 禁止 Timer3 模块
0 = 使能 Timer3 模块
- bit 12 **T2MD:** Timer2 模块禁止位
1 = 禁止 Timer2 模块
0 = 使能 Timer2 模块
- bit 11 **T1MD:** Timer1 模块禁止位
1 = 禁止 Timer1 模块
0 = 使能 Timer1 模块
- bit 10 **未实现:** 读为 0
- bit 9 **PWMMD:** PWM 模块禁止位
1 = 禁止 PWM 模块
0 = 使能 PWM 模块
- bit 8 **未实现:** 读为 0
- bit 7 **I2C1MD:** I2C1 模块禁止位
1 = 禁止 I2C1 模块
0 = 使能 I2C1 模块
- bit 6 **未实现:** 读为 0
- bit 5 **U1MD:** UART1 模块禁止位
1 = 禁止 UART1 模块
0 = 使能 UART1 模块
- bit 4 **未实现:** 读为 0
- bit 3 **SPI1MD:** SPI1 模块禁止位
1 = 禁止 SPI1 模块
0 = 使能 SPI1 模块
- bit 2-1 **未实现:** 读为 0
- bit 0 **ADCMD:** ADC 模块禁止位
1 = 禁止 ADC 模块
0 = 使能 ADC 模块

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 8-2: PMD2: 外设模块禁止控制寄存器 2

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	IC2MD	IC1MD
bit 15						bit 8	
U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	OC2MD	OC1MD
bit 7						bit 0	

图注:

R = 可读位	W = 可写位	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 **IC2MD:** 输入捕捉 2 模块禁止位
 1 = 禁止输入捕捉 2 模块
 0 = 使能输入捕捉 2 模块
- bit 8 **IC1MD:** 输入捕捉 1 模块禁止位
 1 = 禁止输入捕捉 1 模块
 0 = 使能输入捕捉 1 模块
- bit 7-2 **未实现:** 读为 0
- bit 1 **OC2MD:** 输出比较 2 模块禁止位
 1 = 禁止输出比较 2 模块
 0 = 使能输出比较 2 模块
- bit 0 **OC1MD:** 输出比较 1 模块禁止位
 1 = 禁止输出比较 1 模块
 0 = 使能输出比较 1 模块

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 8-3: PMD3: 外设模块禁止控制寄存器 3

U-0	U-0	U-0	U-0	U-0	R/W-0	U-0	U-0
—	—	—	—	—	CMPMD	—	—
bit 15						bit 8	
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10 **CMPMD**: 模拟比较器模块禁止位

1 = 禁止模拟比较器模块

0 = 使能模拟比较器模块

bit 9-0 未实现: 读为 0

寄存器 8-4: PMD4: 外设模块禁止控制寄存器 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	R/W-0	U-0	U-0	U-0
—	—	—	—	REFOMD	—	—	—
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-4 未实现: 读为 0

bit 3 **REFOMD**: 参考时钟发生器模块禁止位

1 = 禁止参考时钟发生器模块

0 = 使能参考时钟发生器模块

bit 2-0 未实现: 读为 0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 8-5: PMD6: 外设模块禁止控制寄存器 6

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PWM4MD	PWM3MD	PWM2MD	PWM1MD
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 15-12

未实现：读为 0
- bit 11

PWM4MD: PWM 发生器 4 模块禁止位

1 = 禁止 PWM 发生器 4 模块

0 = 使能 PWM 发生器 4 模块
- bit 10

PWM3MD: PWM 发生器 3 模块禁止位

1 = 禁止 PWM 发生器 3 模块

0 = 使能 PWM 发生器 3 模块
- bit 9

PWM2MD: PWM 发生器 2 模块禁止位

1 = 禁止 PWM 发生器 2 模块

0 = 使能 PWM 发生器 2 模块
- bit 8

PWM1MD: PWM 发生器 1 模块禁止位

1 = 禁止 PWM 发生器 1 模块

0 = 使能 PWM 发生器 1 模块
- bit 7-0

未实现：读为 0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 8-6: PMD7: 外设模块禁止控制寄存器 7

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	CMP4MD	CMP3MD	CMP2MD	CMP1MD
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 15-12 未实现: 读为 0

bit 11 **CMP4MD**: 模拟比较器 4 模块禁止位

1 = 禁止模拟比较器 4 模块

0 = 使能模拟比较器 4 模块

bit 10 **CMP3MD**: 模拟比较器 3 模块禁止位

1 = 禁止模拟比较器 3 模块

0 = 使能模拟比较器 3 模块

bit 9 **CMP2MD**: 模拟比较器 2 模块禁止位

1 = 禁止模拟比较器 2 模块

0 = 使能模拟比较器 2 模块

bit 8 **CMP1MD**: 模拟比较器 1 模块禁止位

1 = 禁止模拟比较器 1 模块

0 = 使能模拟比较器 1 模块

bit 7-0 未实现: 读为 0

注:

9.0 I/O 端口

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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、 $\overline{\text{MCLR}}$ 和 OSC1/CLKI 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。

9.1 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。外设的输出缓冲器数据和控制信号提供给一对多路开关。这对多路开关用于选择 I/O 引脚的输出数据和控制信号是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 9-1 显示了端口是如何与其他外设复用的以及与之相连接的相关 I/O 引脚。

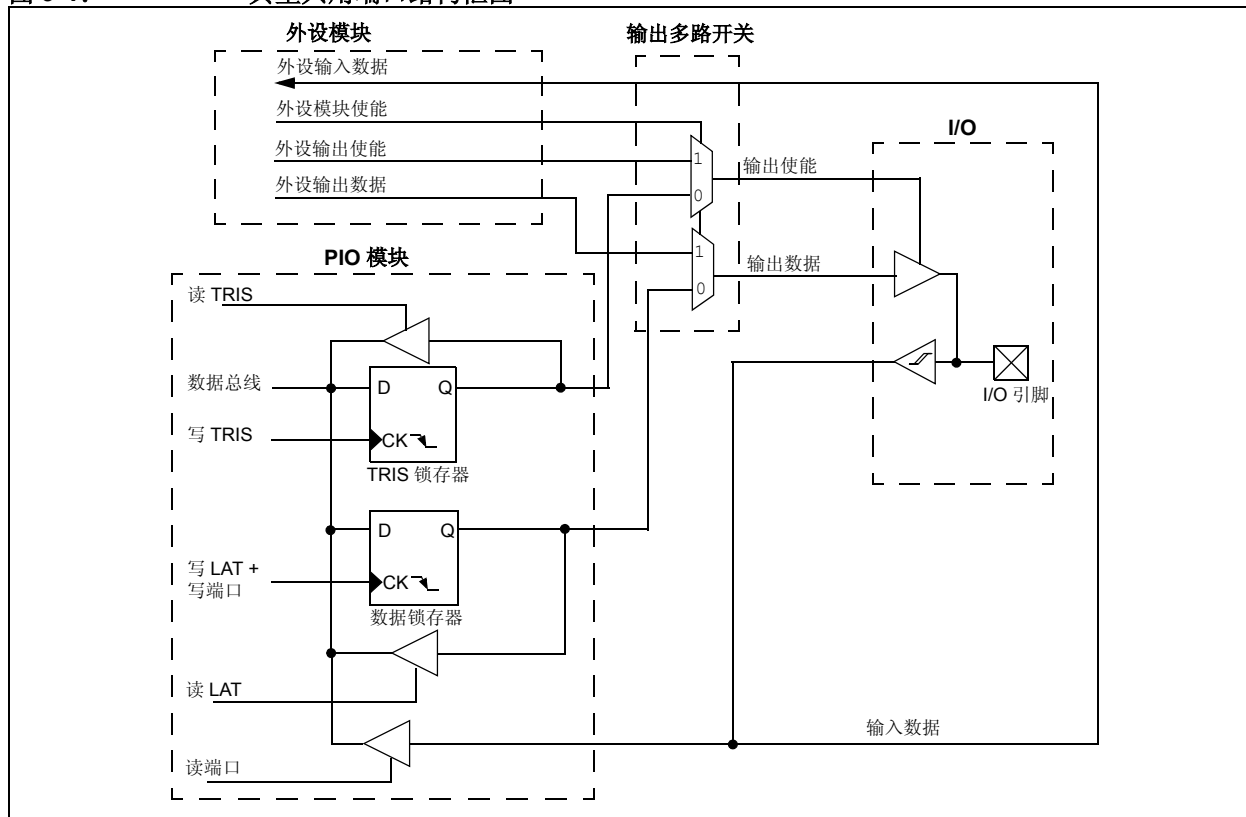
如果外设使能，并且外设正在驱动相关引脚时，将禁止将该引脚用作通用输出引脚。可以读该 I/O 引脚，但并行端口位的输出驱动器将被禁止。如果使能某外设但该外设没有驱动相应的引脚时，则该引脚可由一个端口驱动。

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 9-1: 典型共用端口结构框图



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

9.1.1 漏极开路配置

除 PORT、LAT 和 TRIS 寄存器用于数据控制外，一些仅用作数字功能的端口引脚也可被单独地配置为数字或漏极开路输出。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。允许的最大开漏电压与最大 VIH 规范相同。

某些 I/O 引脚可能具有内部模拟功能，这在器件的引脚图上并未标出。这些引脚必须作为模拟引脚处理。表9-1列出了所有可用的引脚和它们的功能。

表 9-1: 可用 I/O 引脚和功能

I/O 引脚	仅数字引脚 / 5V 耐压	拉 / 灌电流能力 (mA)
RA0	否	4
RA1	否	4
RA2	否	4
RA3	是	16
RA4	是	16
RB0	否	4
RB1	否	4
RB2	否	4
RB3	是	16
RB4	是 ⁽¹⁾	16
RB5	否	4
RB6	是	4
RB7	是	4
RB8	是	4
RB9	否	4
RB10	否	4
RB11	是	16
RB12	是	16
RB13	是	16
RB14	是	16
RB15	是	4
RC0	是	8
RC1	否	4
RC2	否	4
RC3	是	8
RC4	是	8
RC5	是	8
RC6	是	8
RC7	是	8
RC8	是	8
RC9	否	4
RC10	否	4
RC11	是	8
RC12	是	8
RC13	是	8

注 1: dsPIC33FJ06GS202 和 dsPIC33FJ16GS502 器件的 RB4 I/O 引脚不是 5V 耐压的。

9.2 配置模拟端口引脚

ADPCFG 和 TRIS 寄存器用于控制模数 (Analog-to-Digital, A/D) 端口引脚的操作。如果要将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。

9.2.1 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。这通常通过一条 NOP 指令来实现。例 9-1 中给出了一个示例。

9.3 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 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件在选定输入引脚的状态变化 (COS) 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) 30 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, 不再需要使用外部上拉电阻。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。

公式 9-1: 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
```

9.4 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 / 或输出单独地映射到这些 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。一旦建立了外设引脚选择，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。

9.4.1 可用的引脚

外设引脚选择功能可在最多30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。

9.4.2 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。

9.4.2.1 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该外设要映射到的引脚。RPINRx 寄存器用来配置外设输入映射（见寄存器 9-1 至寄存器 9-14）。每个寄存器包含 6 位位域组，每组都与可重映射外设之一相关。用适当的 6 位值编程给定外设的位域，会将与该值相对应的 RPn 引脚映射到此外设。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 9-2 说明了 U1RX 输入的可重映射引脚选择。

图 9-2: U1RX 的可重映射多路开关（MUX）输入

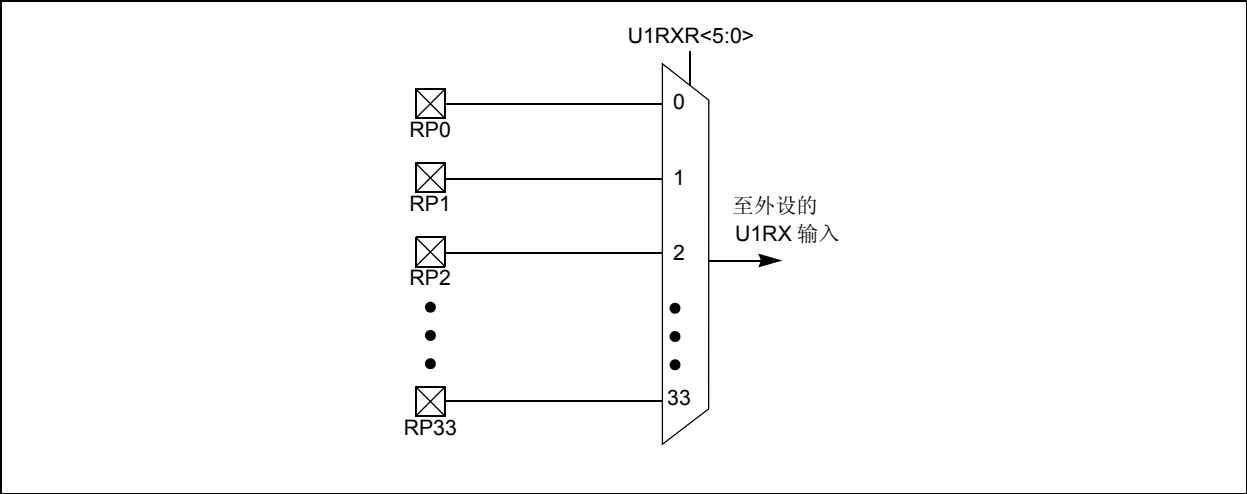


表 9-2: 可选择的输入源（将输入映射到功能）

输入名称	功能名称	寄存器	配置位
外部中断 1	INT1	RPINR0	INT1R<5:0>
外部中断 2	INT2	RPINR1	INT2R<5:0>
Timer1 外部时钟	T1CK	RPINR2	T1CKR<5:0>
Timer2 外部时钟	T2CK	RPINR3	T2CKR<5:0>
Timer3 外部时钟	T3CK	RPINR3	T3CKR<5:0>
输入捕捉 1	IC1	RPINR7	IC1R<5:0>
输入捕捉 2	IC2	RPINR7	IC2R<5:0>
输出比较故障 A	OCFA	RPINR11	OCFAR<5:0>
UART1 接收	U1RX	RPINR18	U1RXR<5:0>
UART1 允许发送	U1CTS	RPINR18	U1CTSR<5:0>
SPI 数据输入 1	SDI1	RPINR20	SDI1R<5:0>
SPI 时钟输入 1	SCK1IN	RPINR20	SCK1R<5:0>
SPI 从选择输入 1	SS1IN	RPINR21	SS1R<5:0>
PWM 故障输入 PWM1	FLT1	RPINR29	FLT1R<5:0>
PWM 故障输入 PWM2	FLT2	RPINR30	FLT2R<5:0>
PWM 故障输入 PWM3	FLT3	RPINR30	FLT3R<5:0>
PWM 故障输入 PWM4	FLT4	RPINR31	FLT4R<5:0>
PWM 故障输入 PWM5	FLT5	RPINR31	FLT5R<5:0>
PWM 故障输入 PWM6	FLT6	RPINR32	FLT6R<5:0>
PWM 故障输入 PWM7	FLT7	RPINR32	FLT7R<5:0>
PWM 故障输入 PWM8	FLT8	RPINR33	FLT8R<5:0>
至 PWM 主时基的外部同步信号	SYNCI1	RPINR33	SYNCI1R<5:0>
至 PWM 主时基的外部同步信号	SYNCI2	RPINR34	SYNCI2R<5:0>

9.4.2.2 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 6 位位域组，每组都与一个 RPN 引脚相关（见寄存器 9-15 至寄存器 9-31）。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 9-3 和图 9-3）。

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 9-3: RPn 的可重映射输出的多路选择功能

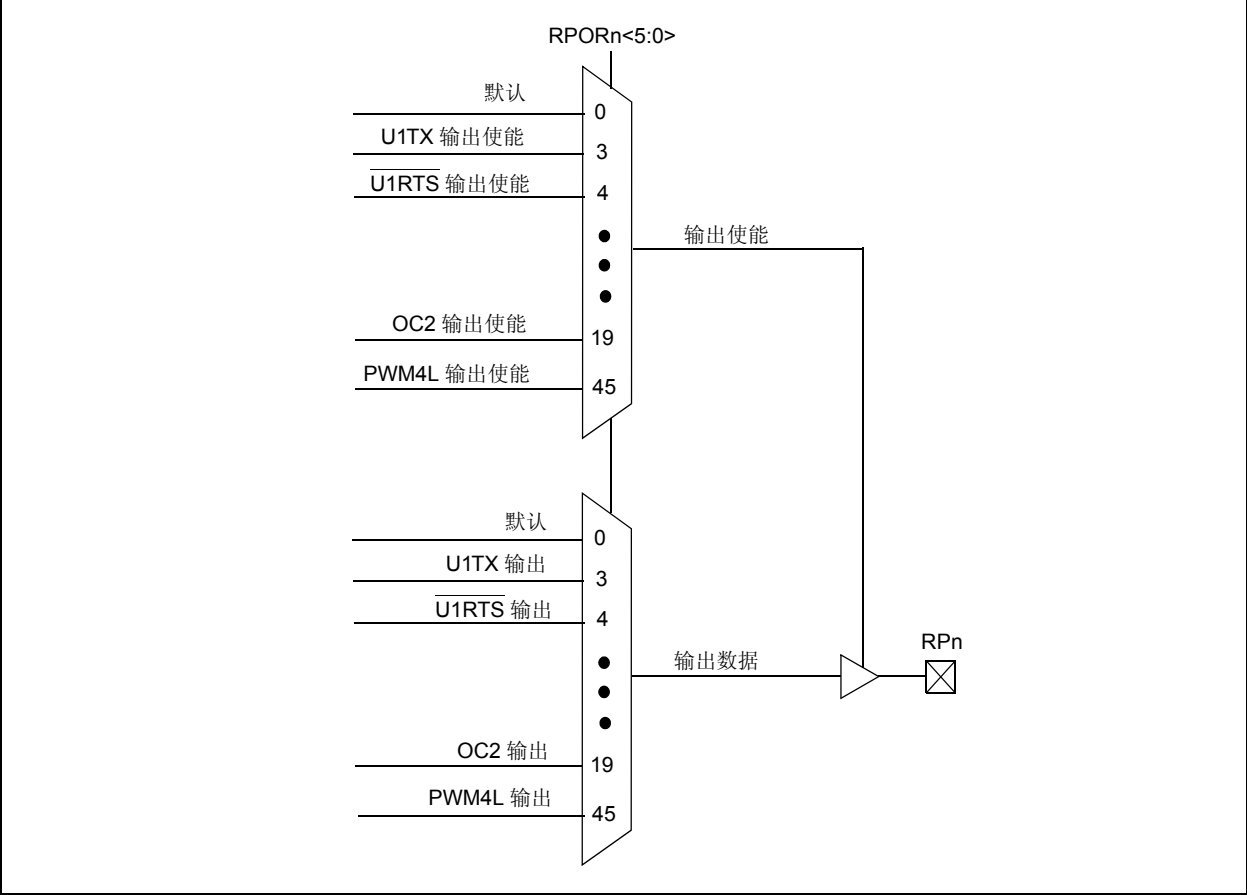


表 9-3: 可重映射引脚 (RPn) 的输出选择

功能	RPORn<5:0>	输出名称
NULL	000000	RPn 连接到默认端口引脚
U1TX	000011	RPn 连接到 UART1 发送
U1RTS	000100	RPn 连接到 UART1 请求发送
SDO1	000111	RPn 连接到 SPI1 数据输出
SCK1OUT	001000	RPn 连接到 SPI1 时钟输出
SS1OUT	001001	RPn 连接到 SPI1 从选择输出
OC1	010010	RPn 连接到输出比较 1
OC2	010011	RPn 连接到输出比较 2
SYNCO1	100101	RPn 通过 PWM 主时基连接到外部器件同步信号
REFCLKO	100110	REFCLK 输出信号
ACMP1	100111	RPn 连接到模拟比较器输出 1
ACMP2	101000	RPn 连接到模拟比较器输出 2
ACMP3	101001	RPn 连接到模拟比较器输出 3
ACMP4	100010	RPn 连接到模拟比较器输出 4
PWM4H	101100	RPn 连接到 PWM 发生器 4 的 PWM 输出引脚
PWM4L	101101	RPn 连接到 PWM 发生器 4 的 PWM 输出引脚

9.4.2.3 虚拟引脚

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件支持 4 个虚拟 RPn 引脚（RP32、RP33、RP34 和 RP35），它们在功能上与所有其他 RPn 引脚相同，只是引脚排列不同。这四个引脚是器件内部的，不连接到器件的物理引脚上。

这些引脚提供了一种简单的方法，可以不使用物理引脚来实现外设间互连。例如，模拟比较器的输出可以连接到 RP32，同时 PWM 故障输入可以配置为 RP32。这种配置可以通过模拟比较器来触发 PWM 故障，而不需要使用器件的实际物理引脚。

9.4.3 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 个功能来防止对外设映射的更改：

- 控制寄存器锁定序列
- 连续状态监视
- 配置位引脚选择锁定

9.4.3.1 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. 执行对 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 C30 帮助文件以获取更多信息。

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。

9.4.3.2 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。

9.4.3.3 配置位引脚选择锁定

为了进一步确保安全，可以将器件配置为防止对 RPINRx 和 RPORx 寄存器进行多于一次写会话。IOL1WAY（FOSC<5>）配置位会阻止 IOLOCK 位在置 1 后被清零。如果 IOLOCK 保持置 1，则不会执行寄存器解锁过程，且不能写入外设引脚选择控制寄存器。清零该位并重新使能外设重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次写会话。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。

9.5 外设引脚选择寄存器

dsPIC33FJ06GS101/X02 和dsPIC33FJ16GSX02/X04系列器件实现了 34 个用于可重映射外设配置的寄存器：

- 15 个输入可重映射外设寄存器
- 19 个输出可重映射外设寄存器

注： 仅当 OSCCON<IOLOCK> = 0 时才能更改输入和输出寄存器值。具体的命令序列请参见第 9.4.3.1 节 “控制寄存器锁定”。

并非所有的器件都实现了所有输出可重映射外设寄存器。更多信息请参见特定寄存器的寄存器描述。

寄存器 9-1: RPINR0: 外设引脚选择输入寄存器 0

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT1R<5:0>					
bit 15		bit 8					
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7		bit 0					

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

bit 15-14 未实现：读为 0
bit 13-8 INT1R<5:0>：将外部中断 1（INT1）分配给对应 RPn 引脚的位
 111111 = 输入连接到 Vss
 100011 = 输入连接到 RP35
 100010 = 输入连接到 RP34
 100001 = 输入连接到 RP33
 100000 = 输入连接到 RP32
 .
 .
 .
 00000 = 输入连接到 RP0
bit 7-0 未实现：读为 0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-2: **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	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT2R<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

INT2R<5:0>: 将外部中断 2 (INT2) 分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

•

•

•

000000 = 输入连接到 RP0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-3: RPINR3: 外设引脚选择输入寄存器 3

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T3CKR<5:0>					
bit 15		bit 8					

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T2CKR<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

未实现: 读为 0
- bit 13-8

T3CKR<5:0>: 将 Timer3 外部时钟 (T3CK) 分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

.

.

.

000000 = 输入连接到 RP0
- bit 7-6

未实现: 读为 0
- bit 5-0

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

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

.

.

.

000000 = 输入连接到 RP0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-4: **RPINR7: 外设引脚选择输入寄存器 7**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC2R<5:0>					
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC1R<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 **未实现:** 读为 0

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

111111 = 输入连接到 Vss
 100011 = 输入连接到 RP35
 100010 = 输入连接到 RP34
 100001 = 输入连接到 RP33
 100000 = 输入连接到 RP32

•
•
•

00000 = 输入连接到 RP0

bit 7-6 **未实现:** 读为 0

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

111111 = 输入连接到 Vss
 100011 = 输入连接到 RP35
 100010 = 输入连接到 RP34
 100001 = 输入连接到 RP33
 100000 = 输入连接到 RP32

•
•
•

00000 = 输入连接到 RP0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-5: 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	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCFAR<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 **OCFAR<5:0>**: 将输出比较故障 A (OCFA) 分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

•

•

•

000000 = 输入连接到 RP0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-6: **RPINR18: 外设引脚选择输入寄存器 18**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1CTSR<5:0>					
bit 15		bit 8					

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1RXR<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 **未实现:** 读为 0

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

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

•

•

•

00000 = 输入连接到 RP0

bit 7-6 **未实现:** 读为 0

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

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

•

•

•

00000 = 输入连接到 RP0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-7: **RPINR20: 外设引脚选择输入寄存器 20**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK1R<5:0>					
bit 15							bit 8
U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI1R<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

未实现: 读为 0
- bit 13-8

SCK1R<5:0>: 将 SPI1 时钟输入 (SCK1IN) 分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

•

•

•

000000 = 输入连接到 RP0
- bit 7-6

未实现: 读为 0
- bit 5-0

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

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

•

•

•

000000 = 输入连接到 RP0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-8: **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	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS1R<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 **SS1R<5:0>:** 将 SPI1 从选择输入 (SS1IN) 分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss
100011 = 输入连接到 RP35
100010 = 输入连接到 RP34
100001 = 输入连接到 RP33
100000 = 输入连接到 RP32
.
.
.
000000 = 输入连接到 RP0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-9: RPINR29: 外设引脚选择输入寄存器 29

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	FLT1R<5:0>					
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

图注:

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

bit 15-14 未实现：读为 0

bit 13-8 **FLT1R<5:0>**: 将 PWM 故障输入 1（FLT1）分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

•

•

•

000000 = 输入连接到 RP0

bit 7-0 未实现：读为 0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-10: **RPINR30: 外设引脚选择输入寄存器 30**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	FLT3R<5:0>					
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	FLT2R<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 **未实现:** 读为 0

bit 13-8 **FLT3R<5:0>:** 将 PWM 故障输入 3 (FLT3) 分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss
 100011 = 输入连接到 RP35
 100010 = 输入连接到 RP34
 100001 = 输入连接到 RP33
 100000 = 输入连接到 RP32

•
•
•

00000 = 输入连接到 RP0

bit 7-6 **未实现:** 读为 0

bit 5-0 **FLT2R<5:0>:** 将 PWM 故障输入 2 (FLT2) 分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss
 100011 = 输入连接到 RP35
 100010 = 输入连接到 RP34
 100001 = 输入连接到 RP33
 100000 = 输入连接到 RP32

•
•
•

00000 = 输入连接到 RP0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-11: **RPINR31: 外设引脚选择输入寄存器 31**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	FLT5R<5:0>					
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	FLT4R<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

未实现: 读为 0
- bit 13-8

FLT5R<5:0>: 将 PWM 故障输入 5 (FLT5) 分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

•

•

•

000000 = 输入连接到 RP0
- bit 7-6

未实现: 读为 0
- bit 5-0

FLT4R<5:0>: 将 PWM 故障输入 4 (FLT4) 分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

•

•

•

000000 = 输入连接到 RP0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-12: **RPINR32: 外设引脚选择输入寄存器 32**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	FLT7R<5:0>					
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	FLT6R<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 **未实现:** 读为 0

bit 13-8 **FLT7R<5:0>:** 将 PWM 故障输入 7 (FLT7) 分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

•

•

•

000000 = 输入连接到 RP0

bit 7-6 **未实现:** 读为 0

bit 5-0 **FLT6R<5:0>:** 将 PWM 故障输入 6 (FLT6) 分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

•

•

•

000000 = 输入连接到 RP0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-13: **RPINR33: 外设引脚选择输入寄存器 33**

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SYNCI1R<5:0>					
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	FLT8R<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

未实现: 读为 0
- bit 13-8

SYNCI1R<5:0>: 将 PWM 主时基外部同步信号分配给对应 RPn 引脚的位
111111 = 输入连接到 Vss
100011 = 输入连接到 RP35
100010 = 输入连接到 RP34
100001 = 输入连接到 RP33
100000 = 输入连接到 RP32
.
.
.
000000 = 输入连接到 RP0
- bit 7-6

未实现: 读为 0
- bit 5-0

FLT8R<5:0>: 将 PWM 故障输入 8 (FLT8) 分配给对应 RPn 引脚的位
111111 = 输入连接到 Vss
100011 = 输入连接到 RP35
100010 = 输入连接到 RP34
100001 = 输入连接到 RP33
100000 = 输入连接到 RP32
.
.
.
000000 = 输入连接到 RP0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-14: RPINR34: 外设引脚选择输入寄存器 34

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SYNCl2R<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 **SYNCl2R<5:0>**: 将 PWM 主时基外部同步信号分配给对应 RPn 引脚的位

111111 = 输入连接到 Vss

100011 = 输入连接到 RP35

100010 = 输入连接到 RP34

100001 = 输入连接到 RP33

100000 = 输入连接到 RP32

•

•

•

000000 = 输入连接到 RP0

寄存器 9-15: RPOR0: 外设引脚选择输出寄存器 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP1R<5:0>					
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
—	—	RP0R<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 未实现: 读为 0

bit 13-8 **RP1R<5:0>**: 将外设输出功能分配给 RP1 输出引脚的位 (请参见表 9-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP0R<5:0>**: 将外设输出功能分配给 RP0 输出引脚的位 (请参见表 9-3 了解外设功能编号)

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-16: RPOR1: 外设引脚选择输出寄存器 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP3R<5:0>					
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
—	—	RP2R<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 未实现: 读为 0
- bit 13-8 **RP3R<5:0>**: 将外设输出功能分配给 RP3 输出引脚的位 (请参见表 9-3 了解外设功能编号)
- bit 7-6 未实现: 读为 0
- bit 5-0 **RP2R<5:0>**: 将外设输出功能分配给 RP2 输出引脚的位 (请参见表 9-3 了解外设功能编号)

寄存器 9-17: RPOR2: 外设引脚选择输出寄存器 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP5R<5:0>					
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP4R<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 未实现: 读为 0
- bit 13-8 **RP5R<5:0>**: 将外设输出功能分配给 RP5 输出引脚的位 (请参见表 9-3 了解外设功能编号)
- bit 7-6 未实现: 读为 0
- bit 5-0 **RP4R<5:0>**: 将外设输出功能分配给 RP4 输出引脚的位 (请参见表 9-3 了解外设功能编号)

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-18: RPOR3: 外设引脚选择输出寄存器 3

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP7R<5:0>					
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
—	—	RP6R<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 未实现: 读为 0

bit 13-8 **RP7R<5:0>**: 将外设输出功能分配给 RP7 输出引脚的位 (请参见表 9-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP6R<5:0>**: 将外设输出功能分配给 RP6 输出引脚的位 (请参见表 9-3 了解外设功能编号)

寄存器 9-19: RPOR4: 外设引脚选择输出寄存器 4

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP9R<5:0>					
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
—	—	RP8R<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 未实现: 读为 0

bit 13-8 **RP9R<5:0>**: 将外设输出功能分配给 RP9 输出引脚的位 (请参见表 9-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP8R<5:0>**: 将外设输出功能分配给 RP8 输出引脚的位 (请参见表 9-3 了解外设功能编号)

注 1: 此寄存器在 dsPIC33FJ06GS101 器件上未实现。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-20: RPOR5: 外设引脚选择输出寄存器 5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP11R<5:0>					
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
—	—	RP10R<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 未实现: 读为 0
- bit 13-8 **RP11R<5:0>**: 将外设输出功能分配给 RP11 输出引脚的位 (请参见表 9-3 了解外设功能编号)
- bit 7-6 未实现: 读为 0
- bit 5-0 **RP10R<5:0>**: 将外设输出功能分配给 RP10 输出引脚的位 (请参见表 9-3 了解外设功能编号)

注 1: 此寄存器在 dsPIC33FJ06GS101 器件上未实现。

寄存器 9-21: RPOR6: 外设引脚选择输出寄存器 6

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP13R<5:0>					
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP12R<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 未实现: 读为 0
- bit 13-8 **RP13R<5:0>**: 将外设输出功能分配给 RP13 输出引脚的位 (请参见表 9-3 了解外设功能编号)
- bit 7-6 未实现: 读为 0
- bit 5-0 **RP12R<5:0>**: 将外设输出功能分配给 RP12 输出引脚的位 (请参见表 9-3 了解外设功能编号)

注 1: 此寄存器在 dsPIC33FJ06GS101 器件上未实现。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-22: RPOR7: 外设引脚选择输出寄存器 7

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP15R<5:0>					
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
—	—	RP14R<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 未实现: 读为 0

bit 13-8 **RP15R<5:0>**: 将外设输出功能分配给 RP15 输出引脚的位 (请参见表 9-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP14R<5:0>**: 将外设输出功能分配给 RP14 输出引脚的位 (请参见表 9-3 了解外设功能编号)

注 1: 此寄存器在 dsPIC33FJ06GS101 器件上未实现。

寄存器 9-23: RPOR8: 外设引脚选择输出寄存器 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP17R<5:0>					
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
—	—	RP16R<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 未实现: 读为 0

bit 13-8 **RP17R<5:0>**: 将外设输出功能分配给 RP17 输出引脚的位 (请参见表 9-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP16R<5:0>**: 将外设输出功能分配给 RP16 输出引脚的位 (请参见表 9-3 了解外设功能编号)

注 1: 仅 dsPIC33FJ16GS404 和 dsPIC33FJ16GS504 器件实现了此寄存器。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-24: RPOR9: 外设引脚选择输出寄存器 9

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP19R<5:0>					
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
—	—	RP18R<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 未实现: 读为 0
- bit 13-8 **RP19R<5:0>**: 将外设输出功能分配给 RP19 输出引脚的位 (请参见表 9-3 了解外设功能编号)
- bit 7-6 未实现: 读为 0
- bit 5-0 **RP18R<5:0>**: 将外设输出功能分配给 RP18 输出引脚的位 (请参见表 9-3 了解外设功能编号)

注 1: 仅 dsPIC33FJ16GS404 和 dsPIC33FJ16GS504 器件实现了此寄存器。

寄存器 9-25: RPOR10: 外设引脚选择输出寄存器 10

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP21R<5:0>					
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
—	—	RP20R<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 未实现: 读为 0
- bit 13-8 **RP21R<5:0>**: 将外设输出功能分配给 RP21 输出引脚的位 (请参见表 9-3 了解外设功能编号)
- bit 7-6 未实现: 读为 0
- bit 5-0 **RP20R<5:0>**: 将外设输出功能分配给 RP20 输出引脚的位 (请参见表 9-3 了解外设功能编号)

注 1: 仅 dsPIC33FJ16GS404 和 dsPIC33FJ16GS504 器件实现了此寄存器。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-26: **RPOR11: 外设引脚选择输出寄存器 11**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP23R<5:0>					
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
—	—	RP22R<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 **未实现:** 读为 0

bit 13-8 **RP23R<5:0>:** 将外设输出功能分配给 RP23 输出引脚的位 (请参见表 9-3 了解外设功能编号)

bit 7-6 **未实现:** 读为 0

bit 5-0 **RP22R<5:0>:** 将外设输出功能分配给 RP22 输出引脚的位 (请参见表 9-3 了解外设功能编号)

注 1: 仅 dsPIC33FJ16GS404 和 dsPIC33FJ16GS504 器件实现了此寄存器。

寄存器 9-27: **RPOR12: 外设引脚选择输出寄存器 12**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP25R<5:0>					
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
—	—	RP24R<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 **未实现:** 读为 0

bit 13-8 **RP25R<5:0>:** 将外设输出功能分配给 RP25 输出引脚的位 (请参见表 9-3 了解外设功能编号)

bit 7-6 **未实现:** 读为 0

bit 5-0 **RP24R<5:0>:** 将外设输出功能分配给 RP24 输出引脚的位 (请参见表 9-3 了解外设功能编号)

注 1: 仅 dsPIC33FJ16GS404 和 dsPIC33FJ16GS504 器件实现了此寄存器。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-28: **RPOR13: 外设引脚选择输出寄存器 13**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP27R<5:0>					
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
—	—	RP26R<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 未实现: 读为 0
- bit 13-8 **RP27R<5:0>**: 将外设输出功能分配给 RP27 输出引脚的位 (请参见表 9-3 了解外设功能编号)
- bit 7-6 未实现: 读为 0
- bit 5-0 **RP26R<5:0>**: 将外设输出功能分配给 RP26 输出引脚的位 (请参见表 9-3 了解外设功能编号)

注 1: 仅 dsPIC33FJ16GS404 和 dsPIC33FJ16GS504 器件实现了此寄存器。

寄存器 9-29: **RPOR14: 外设引脚选择输出寄存器 14**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP29R<5:0>					
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
—	—	RP28R<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 未实现: 读为 0
- bit 13-8 **RP29R<5:0>**: 将外设输出功能分配给 RP29 输出引脚的位 (请参见表 9-3 了解外设功能编号)
- bit 7-6 未实现: 读为 0
- bit 5-0 **RP28R<5:0>**: 将外设输出功能分配给 RP28 输出引脚的位 (请参见表 9-3 了解外设功能编号)

注 1: 仅 dsPIC33FJ16GS404 和 dsPIC33FJ16GS504 器件实现了此寄存器。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 9-30: RPOR16: 外设引脚选择输出寄存器 16

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP33R<5:0>					
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
—	—	RP32R<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 未实现: 读为 0

bit 13-8 **RP33R<5:0>**: 将外设输出功能分配给 RP33 输出引脚的位 (请参见表 9-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP32R<5:0>**: 将外设输出功能分配给 RP32 输出引脚的位 (请参见表 9-3 了解外设功能编号)

寄存器 9-31: RPOR17: 外设引脚选择输出寄存器 17

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP35R<5:0>					
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
—	—	RP34R<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 未实现: 读为 0

bit 13-8 **RP35R<5:0>**: 将外设输出功能分配给 RP35 输出引脚的位 (请参见表 9-3 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP34R<5:0>**: 将外设输出功能分配给 RP34 输出引脚的位 (请参见表 9-3 了解外设功能编号)

注:

10.0 TIMER1

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 (Real-Time Clock, RTC) 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 (RTC) 应用。图 10-1 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：

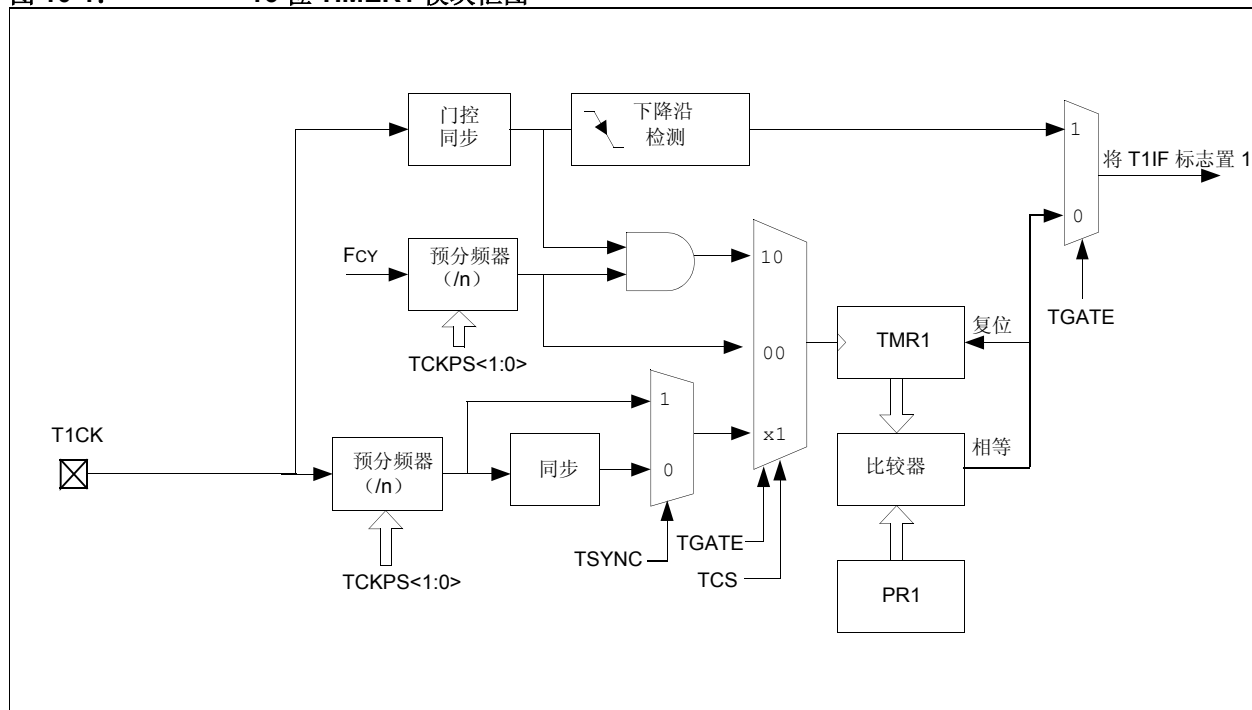
- 定时器时钟源控制位 (TCS)：T1CON<1>
- 定时器同步控制位 (TSYNC)：T1CON<2>
- 定时器门控位 (TGATE)：T1CON<6>

表 10-1 给出了不同工作模式下的定时器控制位的设置。

表 10-1: 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 10-1: 16 位 TIMER1 模块框图



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 10-1: 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

11.0 TIMER2/3 特性

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。

Timer2 是 B 类定时器，具有以下主要特性：

- B 类定时器可以与 C 类定时器组合成一个 32 位定时器
- 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 是 C 类定时器，具有以下主要特性：

- 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。

注： 在所有器件中 Timer3 不可用。

图 11-1: B 类定时器框图 (x = 2)

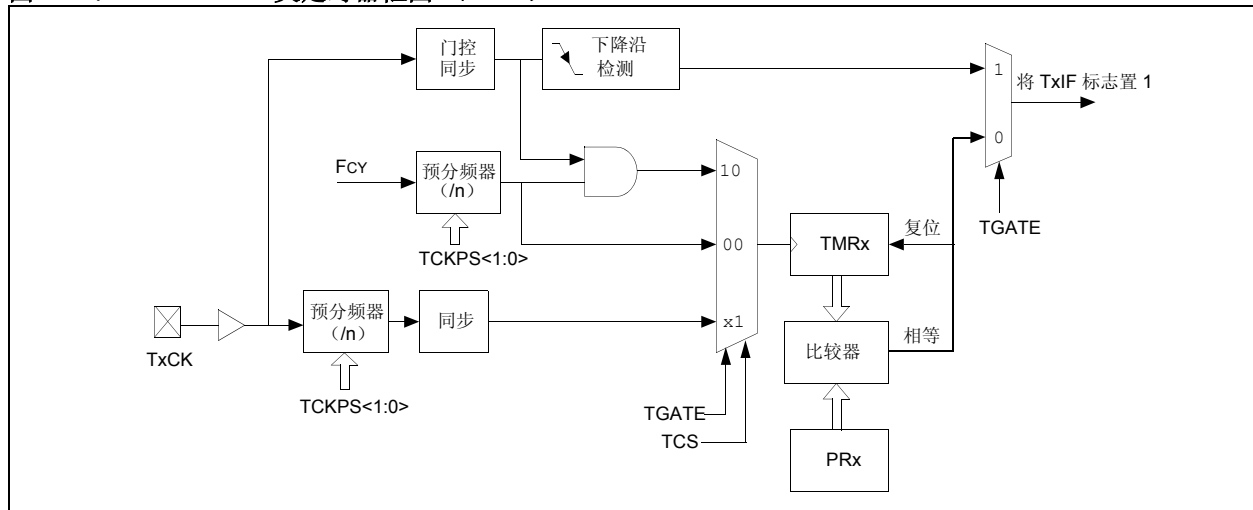
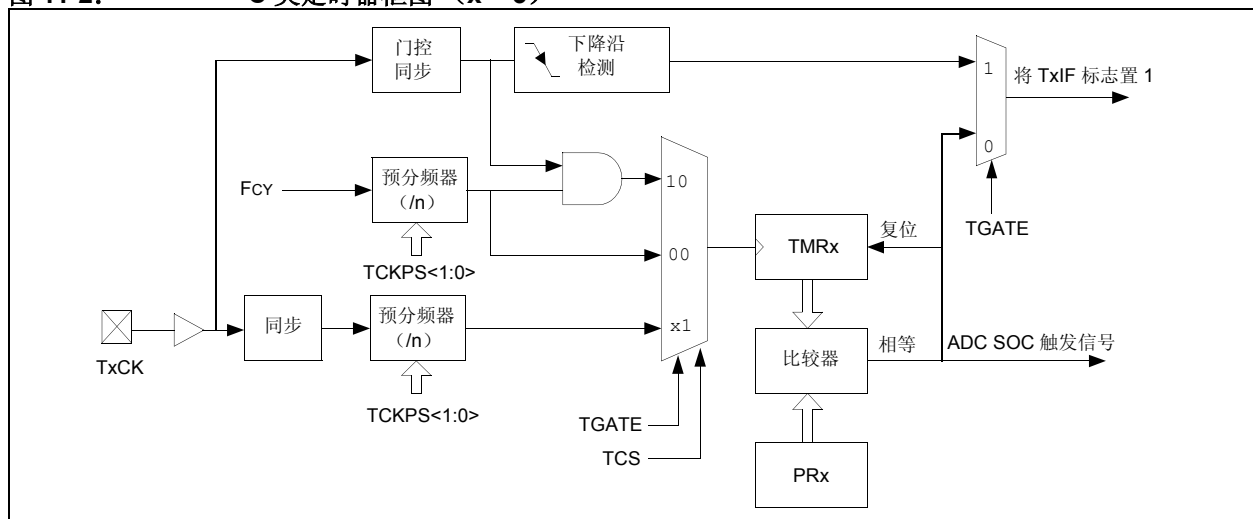


图 11-2: C 类定时器框图 (x = 3)



Timer2/3 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

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。

11.2 32 位工作

可以将 B 类和 C 类 16 位定时模块组合成 32 位定时模块。对于 32 位定时工作，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）寄存器的位来进行设置和控制，而忽略 C 类定时器控制寄存器的位（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 Timer2 和 Timer3。

表 11-2：32 位定时器

B 类定时器（lsw）	C 类定时器（msw）
Timer2	Timer3

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

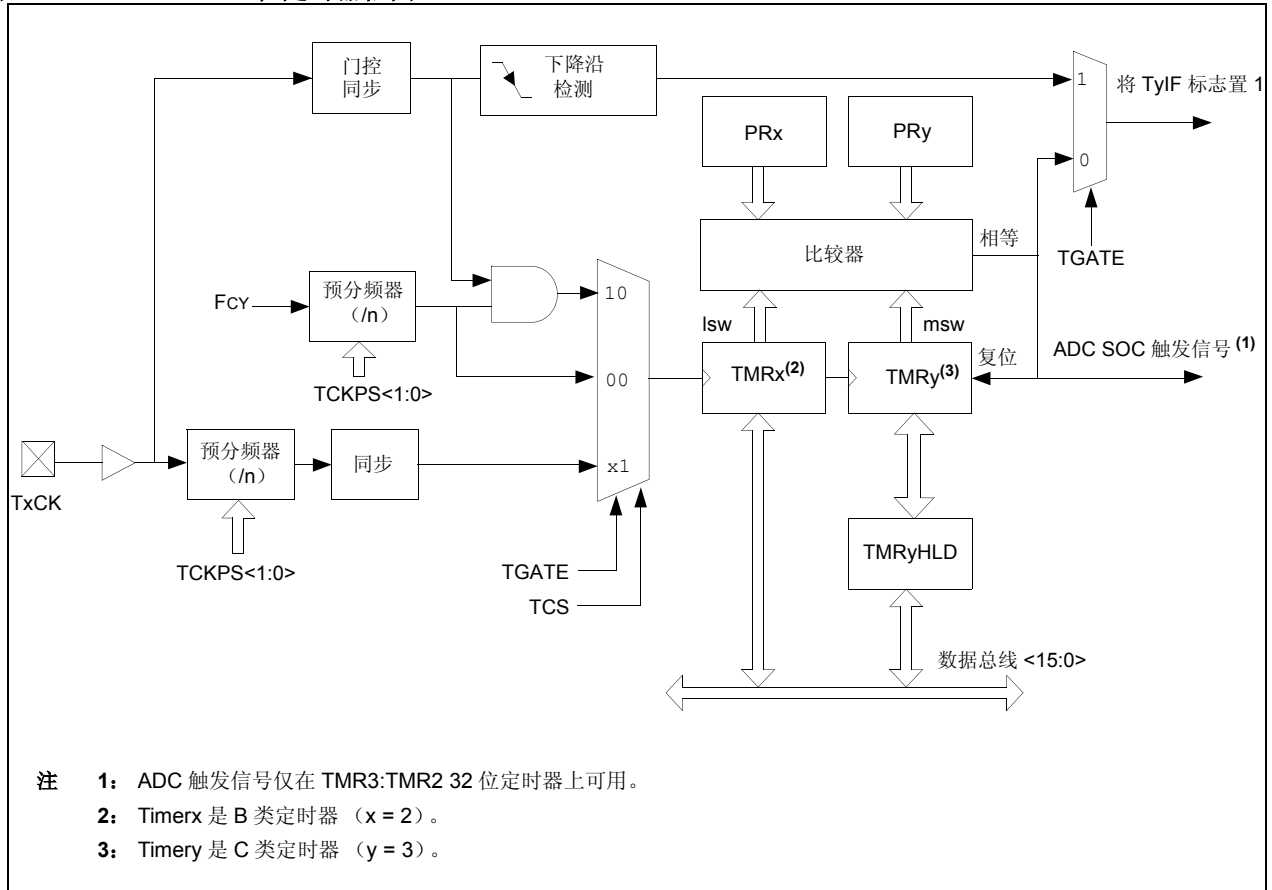
- 定时器模式
- 门控定时器模式
- 同步计数器模式

要将 Timer2/3 配置为 32 位工作：

1. 将 T32 控制位置 1。
2. 使用 TCKPS<1:0> 位为 Timer2 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 包含值的高位字，而 PR2 包含低位字。
5. 如果需要中断，将中断允许位 T3IE 置 1。使用优先级位 T3IP<2:0> 来设置中断优先级。当 Timer2 控制定时器时，中断用作 Timer3 中断。
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 中，TMR3 总是包含计数值的高位字，而 TMR2 包含低位字。

图 11-3: 32 位定时器框图



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 11-1: TxCON: 定时器控制寄存器 (x = 2)

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 ⁽¹⁾	—	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 15TON: 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 13TSIDL: 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 6TGATE: 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 5-4TCKPS<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 3T32: 32 位定时器模式选择位 ⁽¹⁾
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 1TCS: 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 寄存器中的控制位不影响 32 位定时器的工作。

寄存器 11-2: TyCON: 定时器控制寄存器 (y = 3)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON ⁽²⁾	—	TSIDL ⁽¹⁾	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE ⁽²⁾	TCKPS<1:0> ⁽²⁾		—	—	TCS ⁽²⁾	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **TON: Timery 使能位⁽²⁾**
 1 = 启动 16 位 Timery
 0 = 停止 16 位 Timery
- bit 14 **未实现:** 读为 0
- bit 13 **TSIDL: 空闲模式停止位⁽¹⁾**
 1 = 当器件进入空闲模式时, 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⁽²⁾**
 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⁽²⁾**
 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⁽²⁾**
 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中使能 32 位定时器工作时 (T32 (TxCON<3>) = 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中使能 32 位定时器工作时 (T32 (TxCON<3>) = 1), 这些位将不起作用。

注:

12.0 输入捕捉

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件支持最多 2 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：

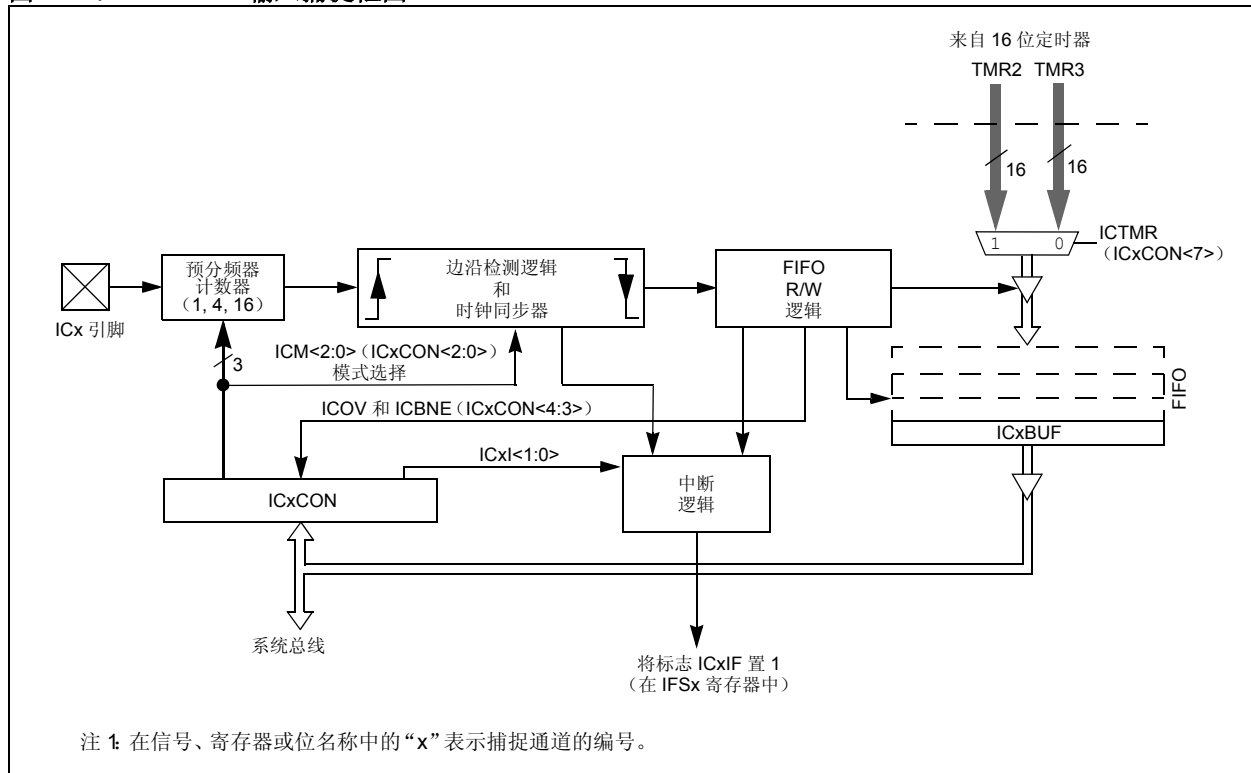
- 简单捕捉事件模式：
 - 每当 ICx 引脚上的输入信号出现下降沿时捕捉定时器值
 - 每当 ICx 引脚上的输入信号出现上升沿时捕捉定时器值
- 在每个边沿（上升沿和下降沿）都捕捉定时器值
- 预分频捕捉事件模式：
 - ICx 引脚上的输入信号每出现 4 个上升沿捕捉一次定时器值
 - ICx 引脚上的输入信号每出现 16 个上升沿捕捉一次定时器值

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

图 12-1: 输入捕捉框图



12.1 输入捕捉寄存器

寄存器 12-1: **ICxCON**: 输入捕捉 x 控制寄存器 (x = 1 或 2)

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

R/W-0	R/W-0	R/W-0	R-0, HC	R-0, HC	R/W-0	R/W-0	R/W-0
ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>		
bit 7							bit 0

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

- bit 15-14 **未实现:** 读为 0
- bit 13 **ICSIDL:** 输入捕捉模块在空闲模式下停止的控制位
 1 = 输入捕捉模块在 CPU 空闲模式下将停止工作
 0 = 输入捕捉模块在 CPU 空闲模式下将继续工作
- bit 12-8 **未实现:** 读为 0
- bit 7 **ICTMR:** 输入捕捉定时器选择位
 1 = 发生捕捉事件时捕捉 TMR2 的内容
 0 = 发生捕捉事件时捕捉 TMR3 的内容
- bit 6-5 **ICI<1:0>:** 选择每次产生中断捕捉的次数的位
 11 = 每 4 次捕捉事件产生一次中断
 10 = 每 3 次捕捉事件产生一次中断
 01 = 每 2 次捕捉事件产生一次中断
 00 = 每次捕捉事件产生一次中断
- bit 4 **ICOV:** 输入捕捉溢出状态标志位 (只读)
 1 = 发生了输入捕捉溢出
 0 = 未发生输入捕捉溢出
- bit 3 **ICBNE:** 输入捕捉缓冲区空状态位 (只读)
 1 = 输入捕捉缓冲区非空, 至少可以再读一个捕捉值
 0 = 输入捕捉缓冲区为空
- bit 2-0 **ICM<2:0>:** 输入捕捉模式选择位
 111 = 仅当器件处于休眠或空闲模式时, 输入捕捉才用作中断引脚。仅上升沿检测, 所有其他控制位都不适用。
 110 = 未使用 (模块被禁止)
 101 = 捕捉模式, 每 16 个上升沿捕捉一次
 100 = 捕捉模式, 每 4 个上升沿捕捉一次
 011 = 捕捉模式, 每个上升沿捕捉一次
 010 = 捕捉模式, 每个下降沿捕捉一次
 001 = 捕捉模式, 每个边沿 (上升沿和下降沿) 捕捉一次。ICI<1:0> 位不控制此模式下的中断产生。
 000 = 输入捕捉模块关闭

13.0 输出比较

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。

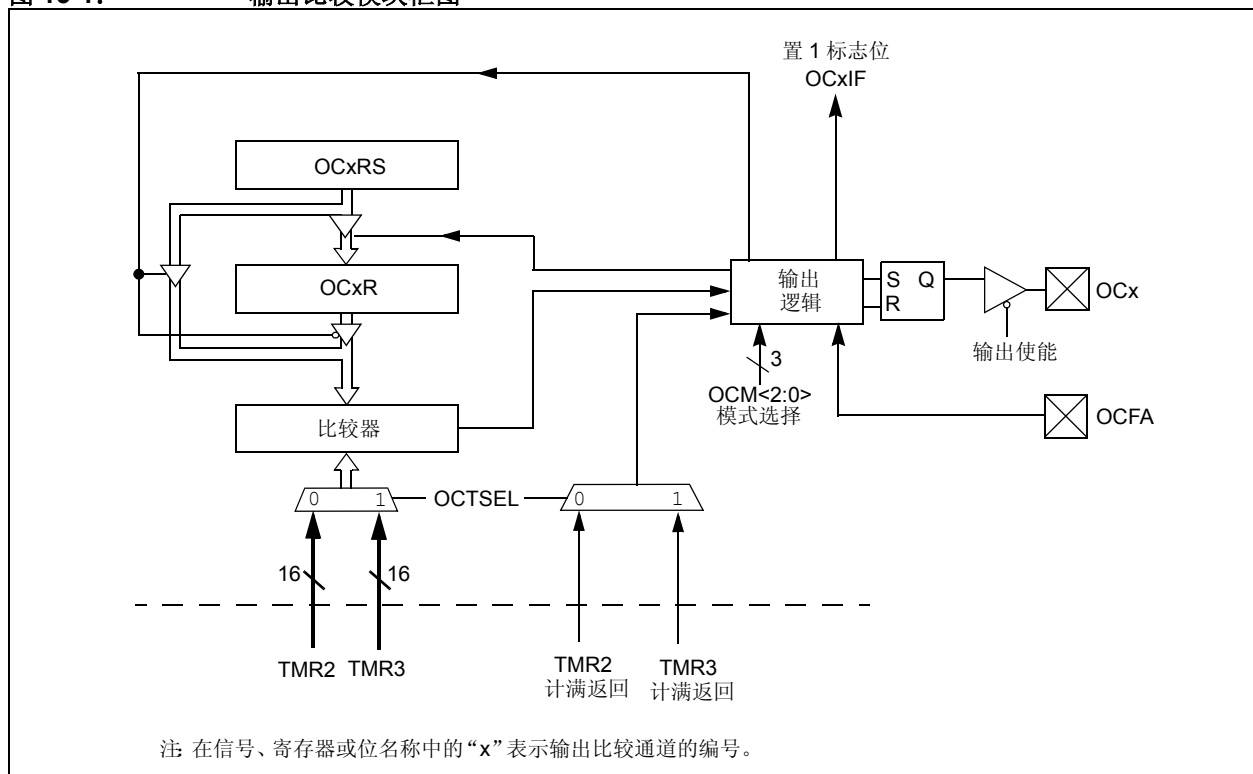
输出比较模块可以选择 Timer2 或 Timer3 作为其时基。模块将定时器的值与一个或两个比较寄存器的值（取决于所选的工作模式）作比较。当定时器值与比较寄存器

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

图 13-1: 输出比较模块框图



13.1 输出比较模式

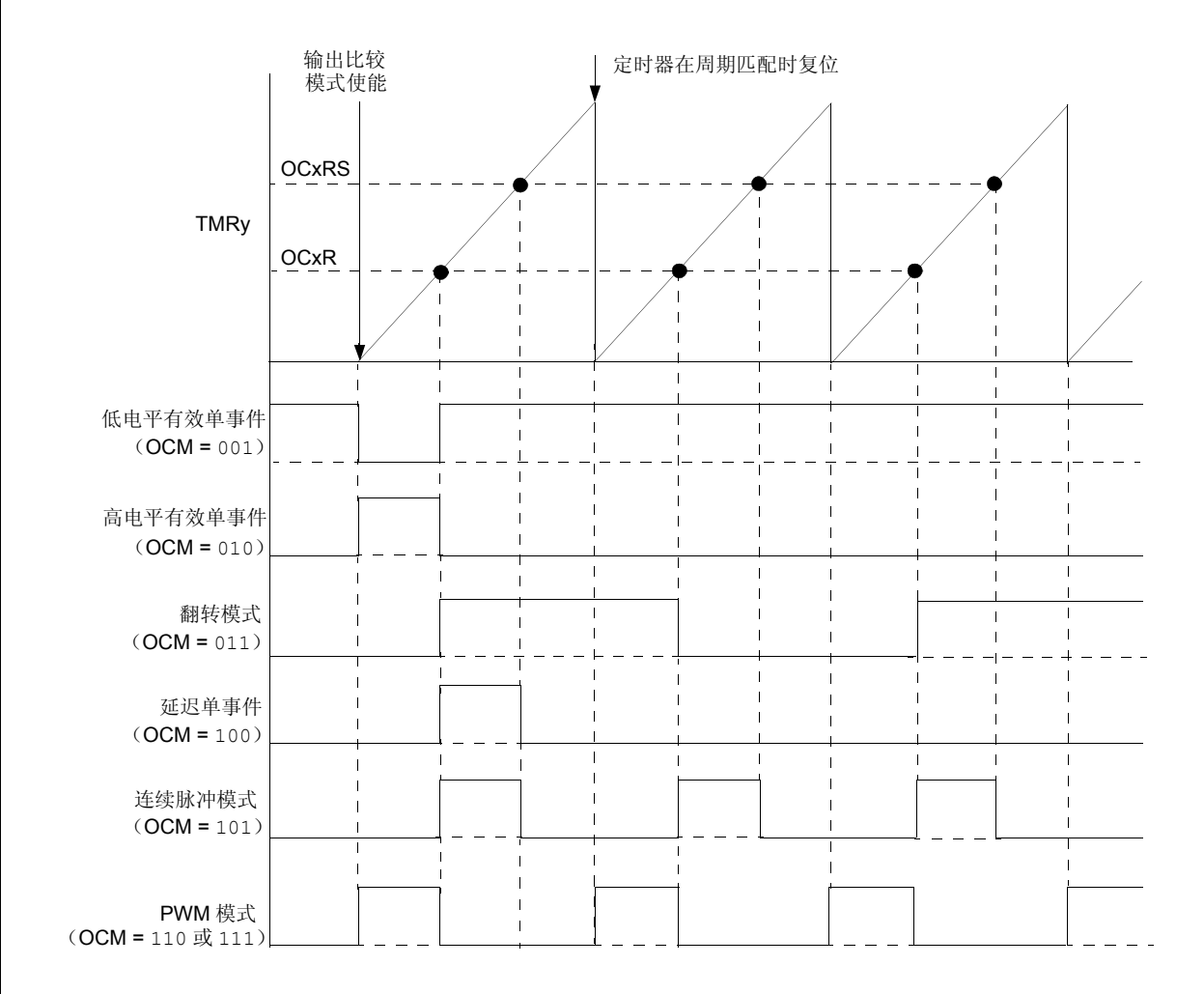
通过设置输出比较控制寄存器中的相应输出比较模式位 OCM<2:0>（OCxCON<2:0>），配置输出比较模式。表 13-1 列出了输出比较模式的不同位设置。图 13-2 说

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 13-1： 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	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	0	OCX 下降沿
110	不带故障保护的 PWM	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	0，如果 OCxR 为零 1，如果 OCxR 非零	OC1 至 OC4 的 OCFA 下降沿

图 13-2： 输出比较操作



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 13-1: **OCxCON: 输出比较 x 控制寄存器 (x = 1 或 2)**

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	

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

bit 15-14 **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:

14.0 高速 PWM

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 43. High-Speed PWM”（DS70323），该文档可从 Microchip 网站（www.microchip.com）下载。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件的高速 PWM 模块支持多种 PWM 模式和输出形式。此 PWM 模块非常适用于电源转换应用，如：

- AC/DC 转换器
- DC/DC 转换器
- 功率因数校正
- 不间断电源（Uninterruptible Power Supply, UPS）
- 逆变器
- 电池充电器
- 数字照明

14.1 特性概述

高速 PWM 模块具有以下特性：

- 2 至 4 个 PWM 发生器，可提供 4 至 8 路输出
- 8 路 PWM 输出中的每一路都具有独立的时基和占空比
- 上升沿和下降沿的死区：
- 40 MIPS 时，占空比分辨率为 1.04 ns
- 40 MIPS 时，死区分辨率为 1.04 ns
- 40 MIPS 时，相移分辨率为 1.04 ns
- 40 MIPS 时，频率分辨率为 1.04 ns
- 支持的 PWM 模式：
 - 标准边沿对齐
 - 真正独立输出
 - 互补
 - 中心对齐
 - 推挽式
 - 多相
 - 可变相位
 - 固定关断时间
 - 电流复位
 - 限流
- 为 8 路 PWM 输出的每一路提供了独立的故障 / 限流输入
- 输出改写控制
- 特殊事件触发器
- PWM 捕捉功能
- 输入时钟预分频器

- 由 PWM 送至 ADC 的双路触发信号
- PWMxH 和 PWMxL 输出引脚交换
- PWM4H 和 PWM4L 引脚可重映射
- 可在运行过程中更改 PWM 频率、占空比和相移
- 可禁止独立的 PWM 发生器，以降低功耗
- 前沿消隐（Leading-Edge Blanking, LEB）功能

注： 在中心对齐 PWM 模式下，占空比、死区、相移和频率的分辨率都是 8.32 ns。

图 14-1 给出了 PWM 模块的简化框图。图 14-2 说明了在互补 PWM 模式下，是如何将模块硬件分配给每个 PWM 输出对的。PWM 模块的各功能单元将在后续章节中讨论。

PWM 模块具有 4 个 PWM 发生器。此模块具有最多 8 个 PWM 输出引脚：PWM1H、PWM1L、PWM2H、PWM2L、PWM3H、PWM3L、PWM4H 和 PWM4L。对于互补输出，这 8 个 I/O 引脚被分成高端 / 低端（H/L）对。

14.2 特性描述

PWM 模块是为有以下需求的应用设计的：

- 高 PWM 频率时具有高分辨率
- 具有驱动标准边沿对齐、中心对齐互补模式和推挽模式输出的能力
- 具有生成多相 PWM 输出的能力

对于中心对齐模式，占空比、相移和死区分辨率都是 8 ns。

两种常见的中等功率电源转换器拓扑结构是推挽和半桥转换器。这两种设计要求 PWM 输出信号在两个引脚之间交替输出，如同推挽 PWM 模式所提供的那样。

相移 PWM 描述了每个 PWM 发生器提供输出的情况，但发生器输出之间的相位关系是可指定且可变的。

多相 PWM 通常用于改善 DC/DC 转换器的负载瞬态响应，以及缩小输出滤波电容和电感的体积。多个 DC/DC 转换器通常并行工作，但在时间上有一个相移。工作在 250 kHz 的单 PWM 输出的周期为 4 μ s，但由 4 个 PWM 通道组成的一个阵列（PWM 通道之间的相移为 1 μ s）可产生 1 MHz 的有效开关频率。多相 PWM 应用通常使用固定相位关系。

可变相位 PWM 在零电压转换（Zero Voltage Transition, ZVT）电源转换器中很有用。此模式下 PWM 占空比总是为 50%，可通过改变两个 PWM 发生器之间的相对相移来控制功率流。

图 14-1: 高速 PWM 模块的简化概念框图

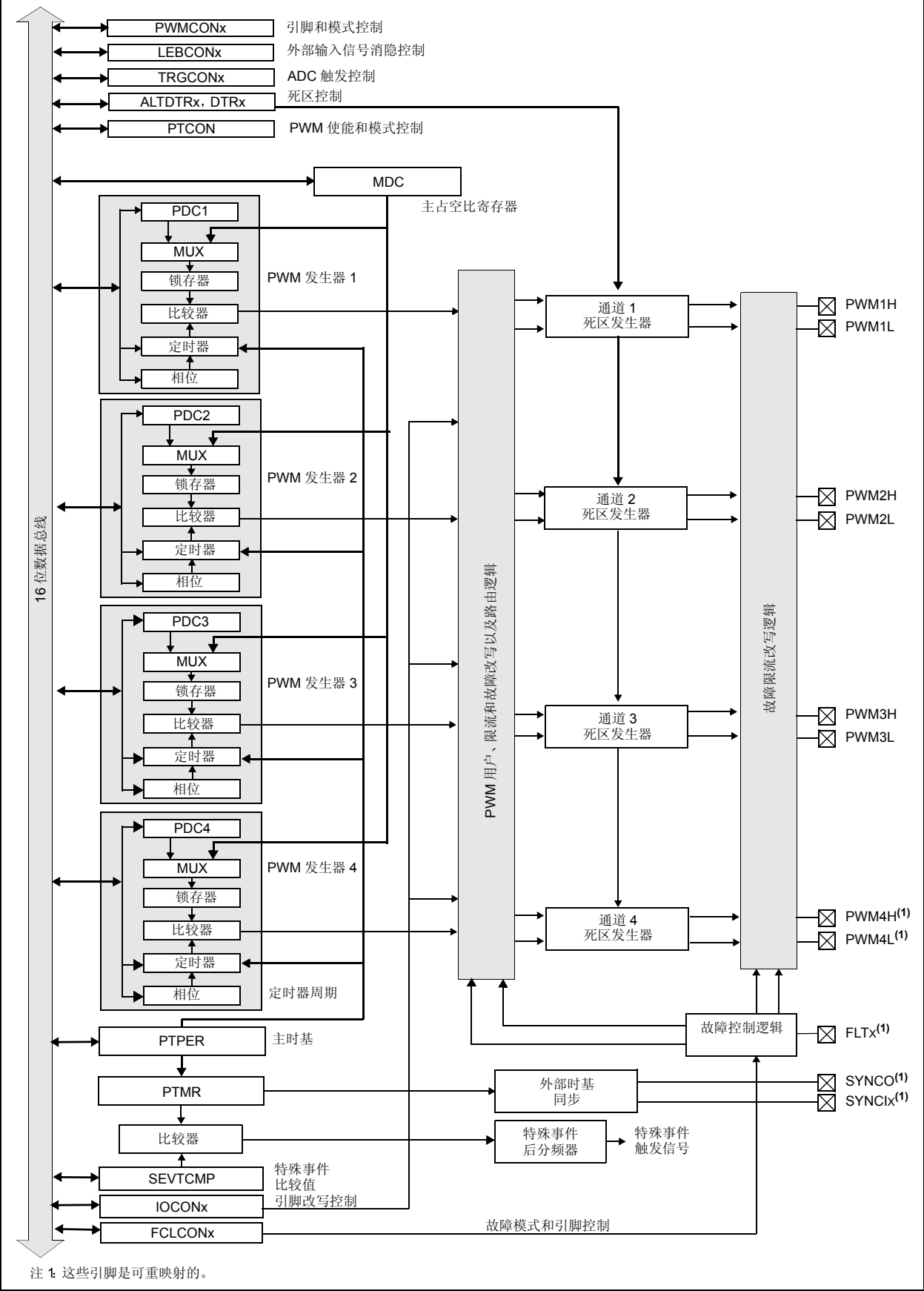
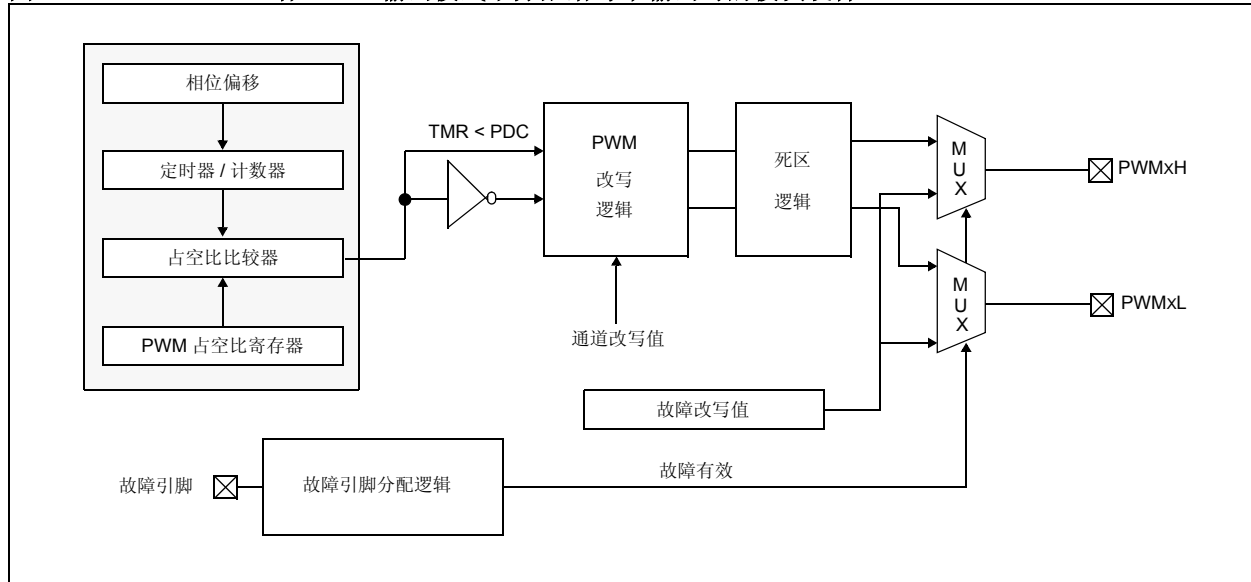


图 14-2: 互补 PWM 输出模式下分配给每个输出对的模块硬件



14.3 控制寄存器

下列寄存器控制高速 PWM 模块的工作：

- PTCON: PWM 时基控制寄存器
- PTCON2: PWM 时钟分频比选择寄存器
- PTPER: PWM 主时基寄存器 (1)
- SEVTCMP: PWM 特殊事件比较寄存器
- MDC: PWM 主占空比寄存器
- PWMCONx: PWMx 控制寄存器
- PDCx: PWMx 发生器占空比寄存器
- PHASEx: PWMx 主相移寄存器 (PHASEx 寄存器为 PWMxH 提供本地时基周期)
- DTRx: PWMx 死区寄存器
- ALTDTRx: PWMx 备用死区寄存器
- SDCx: PWMx 辅助占空比寄存器
- SPHASEx: PWMx 辅助相移寄存器 (为 PWMxL 提供本地时基周期)
- TRGCONx: PWMx 触发器控制寄存器
- IOCONx: PWMx I/O 控制寄存器
- FCLCONx: PWMx 故障限流控制寄存器
- TRIGx: PWMx 主触发器比较值寄存器
- STRIGx: PWMx 辅助触发器比较值寄存器
- LEBCONx: 前沿消隐控制寄存器
- PWMCAPx: 主 PWMx 时基捕捉寄存器

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-1: PTCON: PWM 时基控制寄存器

R/W-0	U-0	R/W-0	HS/HC-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN	—	PTSIDL	SESTAT	SEIEN	EIPU ⁽¹⁾	SYNCPOL ⁽¹⁾	SYNCOEN ⁽¹⁾
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SYNCEN ⁽¹⁾	—	SYNCSRC<1:0> ⁽¹⁾		SEVTPS<3:0> ⁽¹⁾			
bit 7		bit 0					

图注:	HC = 可硬件清零位	HS = 可硬件置 1 位
R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = POR 时的值	1 = 置 1	0 = 清零
		x = 未知

- bit 15 **PTEN:** PWM 模块使能位
1 = 使能 PWM 模块
0 = 禁止 PWM 模块
- 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 **SESTAT:** 特殊事件中断标志状态位
1 = 特殊事件中断在等待处理
0 = 无特殊事件中断在等待处理
- bit 11 **SEIEN:** 特殊事件中断允许位
1 = 允许特殊事件中断
0 = 禁止特殊事件中断
- bit 10 **EIPU:** 立即更新周期使能位 ⁽¹⁾
1 = 立即更新有效周期寄存器
0 = 有效周期寄存器更新发生在 PWM 周期边界
- bit 9 **SYNCPOL:** 同步输入 / 输出极性位 ⁽¹⁾
1 = SYNCIx 和 SYNCO 的极性翻转 (低电平有效)
0 = SYNCIx 和 SYNCO 为高电平有效
- bit 8 **SYNCOEN:** 主时基同步使能位 ⁽¹⁾
1 = 使能 SYNCO 输出
0 = 禁止 SYNCO 输出
- bit 7 **SYNCEN:** 外部时基同步使能位 ⁽¹⁾
1 = 使能主时基外部同步
0 = 禁止主时基外部同步
- bit 6 **未实现:** 读为 0
- bit 5-4 **SYNCSRC<1:0>:** 同步源选择位 ⁽¹⁾
00 = SYNCI1
01 = SYNCI2
10 = 保留
11 = 保留
- bit 3-0 **SEVTPS<3:0>:** PWM 特殊事件触发信号输出后分频比选择位 ⁽¹⁾
0000 = 1:1 每个比较匹配事件, 后分频器产生一个特殊事件触发信号
0001 = 1:2 每 2 个比较匹配事件, 后分频器产生一个特殊事件触发信号
•
•
•
1111 = 1:16 每 16 个比较匹配事件, 后分频器产生一个特殊事件触发信号

注 1: 仅当 PTEN = 0 时才能改变这些位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-2: PTCON2: PWM 时钟分频比选择寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PCLKDIV<2:0> ⁽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 15-3 未实现: 读为 0

bit 2-0 **PCLKDIV<2:0>**: PWM 输入时钟预分频比选择位 ⁽¹⁾

000 = 1 分频, 最大 PWM 时序分辨率 (上电默认值)

001 = 2 分频, 最大 PWM 时序分辨率

010 = 4 分频, 最大 PWM 时序分辨率

011 = 8 分频, 最大 PWM 时序分辨率

100 = 16 分频, 最大 PWM 时序分辨率

101 = 32 分频, 最大 PWM 时序分辨率

110 = 64 分频, 最大 PWM 时序分辨率

111 = 保留

注 1: 仅当 PTEN = 0 时才能改变这些位。在工作期间改变时钟选择将产生不可预知的结果。

寄存器 14-3: PTPER: PWM 主时基寄存器 ⁽¹⁾

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PTPER<15:8>							
bit 15							bit 8
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	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 15-0 **PTPER<15:0>**: PWM 主时基 (PMTMR) 周期值位

注 1: 可装入 PTPER 寄存器的最小值是 0x0010, 最大值是 0xFFFF8。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-4: SEVTCMP: PWM 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
SEVTCMP <15:8>							
bit 15							
bit 8							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
SEVTCMP <7:3>					—	—	—
bit 7					bit 0		

图注:			
R = 可读位	W = 可写位	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 SEVTCMP<15:3>: 特殊事件比较计数值位
bit 2-0 未实现: 读为 0

寄存器 14-5: MDC: PWM 主占空比寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MDC<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
MDC<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 MDC<15:0>: 主 PWM 占空比值位

注	1: 对应于值 0x0008, 可在 PWM 输出上产生最小脉冲宽度; 同时对应于周期值 -0x0008, 可以产生最大脉冲宽度。
2:	当占空比接近 PWM 周期的 0% 或者 100% 时 (0 ns-40 ns, 取决于工作模式), PWM 占空比分辨率从 1 LSB 变为 3 LSB。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-6: PWMCONx: PWMx 控制寄存器

HS/HC-0	HS/HC-0	HS/HC-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FLTSTAT ⁽¹⁾	CLSTAT ⁽¹⁾	TRGSTAT	FLTIE	CLIE	TRGIE	ITB ⁽³⁾	MDCS ⁽³⁾
bit 15						bit 8	

R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
DTC<1:0>		—	—	—	CAM ^(2,3)	XPRES	IUE
bit 7						bit 0	

图注:	HC = 可硬件清零位	HS = 可硬件置 1 位
R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = POR 时的值	1 = 置 1	0 = 清零
		x = 未知

- bit 15 **FLTSTAT:** 故障中断状态位 ⁽¹⁾
 1 = 故障中断在等待处理
 0 = 无故障中断在等待处理。通过设置 FLTIE = 0 来清零此位。
- bit 14 **CLSTAT:** 限流中断状态位 ⁽¹⁾
 1 = 限流中断在等待处理
 0 = 无限流中断在等待处理。通过设置 CLIE = 0 来清零此位。
- bit 13 **TRGSTAT:** 触发中断状态位
 1 = 触发中断在等待处理
 0 = 无触发中断在等待处理。通过设置 TRGIE = 0 来清零此位。
- bit 12 **FLTIE:** 故障中断允许位
 1 = 允许故障中断
 0 = 禁止故障中断, 且清零 FLTSTAT 位
- bit 11 **CLIE:** 限流中断允许位
 1 = 允许限流中断
 0 = 禁止限流中断, 且清零 CLSTAT 位
- bit 10 **TRGIE:** 触发中断允许位
 1 = 触发事件产生中断请求
 0 = 禁止触发事件中断, 且清零 TRGSTAT 位
- bit 9 **ITB:** 独立时基模式位 ⁽³⁾
 1 = PHASEx/SPHASEx 寄存器为此 PWM 发生器提供时基周期
 0 = PTPER 寄存器为此 PWM 发生器提供时序
- bit 8 **MDCS:** 主占空比寄存器选择位 ⁽³⁾
 1 = MDC 寄存器为此 PWM 发生器提供占空比信息
 0 = PDCx/SDCx 寄存器为此 PWM 发生器提供占空比信息
- bit 7-6 **DTC<1:0>:** 死区控制位
 00 = 对于所有输出模式施加正死区
 01 = 对于所有输出模式施加负死区
 10 = 禁止死区功能
 11 = 保留
- bit 5-3 **未实现:** 读为 0

- 注** 1: 在这里, 必须用软件清零中断控制器中相应 IFSx 寄存器中的中断标志状态位。
 2: 要使用中心对齐模式, 必须使能独立时基模式 (ITB = 1)。如果 ITB = 0, 则 CAM 位被忽略。
 3: 仅当 PTEN = 0 时才能改变这些位。在工作期间改变时钟选择将产生不可预知的结果。

寄存器 14-6: PWMCONx: PWMx 控制寄存器 (续)

bit 2	CAM: 中心对齐模式使能位 (2,3) 1 = 使能中心对齐模式 0 = 禁止中心对齐模式
bit 1	XPRES: 外部 PWM 复位控制位 1 = 在独立的时基模式下, 由限流源复位此 PWM 发生器的时基 0 = 外部引脚不影响 PWM 时基
bit 0	IUE: 立即更新使能位 1 = 立即更新有效 MDC/PDCx/SDCx 寄存器 0 = 对有效 MDC/PDCx/SDCx 寄存器的更新与 PWM 时基同步

- 注 1: 在这里, 必须用软件清零中断控制器中相应 IFSx 寄存器中的中断标志状态位。
- 2: 要使用中心对齐模式, 必须使能独立时基模式 (ITB = 1)。如果 ITB = 0, 则 CAM 位被忽略。
- 3: 仅当 PTEN = 0 时才能改变这些位。在工作期间改变时钟选择将产生不可预知的结果。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-7: PDCx: PWMx 发生器占空比寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PDCx<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
PDCx<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 PDCx<15:0>: PWM 发生器 x 占空比值位

- 注 1:** 在独立 PWM 模式下, PDCx 寄存器仅控制 PWMxH 占空比。在互补、冗余和推挽模式下, PDCx 寄存器控制 PWMxH 和 PWMxL 占空比。对应于值 0x0008, 可在 PWM 输出上产生最小脉冲宽度; 同时对应于值 0xFFEF, 可以产生最大脉冲宽度。
- 2:** 当占空比接近 PWM 周期的 0% 或者 100% 时 (0 ns-40 ns, 取决于工作模式), PWM 占空比分辨率从 1 LSB 变为 3 LSB。

寄存器 14-8: SDCx: PWMx 辅助占空比寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SDCx<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
SDCx<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 SDCx<15:0>: 用于 PWMxL 输出引脚辅助占空比的位

- 注 1:** SDCx 寄存器仅在独立 PWM 模式下使用。当在独立 PWM 模式下使用时, SDCx 寄存器控制 PWMxL 占空比。对应于值 0x0008, 可在 PWM 输出上产生最小脉冲宽度; 同时对应于值 0xFFEF, 可以产生最大脉冲宽度。
- 2:** 当占空比接近 PWM 周期的 0% 或者 100% 时 (0 ns-40 ns, 取决于工作模式), PWM 占空比分辨率从 1 LSB 变为 3 LSB。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-9: PHASEx: PWMx 主相移寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PHASEx<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
PHASEx<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 PHASEx<15:0>: 此 PWM 发生器的 PWM 相移值或独立时基周期位

注:	如果 PWMCONx<ITB> = 0, 根据不同工作模式适用下列情况:
	• 互补、冗余和推挽输出模式 (IOCONx<PMOD> = 00、01 或 10)
	PHASEx<15:0> = 用于 PWMxH 和 PWMxL 输出的相移值
	• 真正独立输出模式 (IOCONx<PMOD> = 11)
	PHASEx<15:0> = 仅用于 PWMxH 相移值
	如果 PWMCONx<ITB> = 1, 根据不同工作模式适用下列情况:
	• 互补、冗余和推挽输出模式 (IOCONx<PMOD> = 00、01 或 10)
	PHASEx<15:0> = 用于 PWMxH 和 PWMxL 的独立时基周期值
	• 真正独立输出模式 (IOCONx<PMOD> = 11)
	PHASEx<15:0> = 仅用于 PWMxH 的独立时基周期值

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-10: **SPHASEx: PWMx 辅助相移寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPHASEx<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
SPHASEx<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 **SPHASEx<15:0>: PWMxL 输出引脚的辅助相位偏移位** (仅在独立 PWM 模式下使用)

- 注:** 如果 PWMCONx<ITB> = 0, 根据不同工作模式适用下列情况:
- 互补、冗余和推挽输出模式 (IOCONx<PMOD> = 00、01 或 10)
 SPHASEx<15:0> = 不使用
 - 真正独立输出模式 (IOCONx<PMOD> = 11)
 PHASEx<15:0> = 仅用于 PWMxL 的相移值
- 如果 PWMCONx<ITB> = 1, 根据不同工作模式适用下列情况:
- 互补、冗余和推挽输出模式 (IOCONx<PMOD> = 00、01 或 10)
 SPHASEx<15:0> = 不使用
 - 真正独立输出模式 (IOCONx<PMOD> = 11)
 PHASEx<15:0> = 用于 PWMxL 的独立时基周期值

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-11: DTRx: PWMx 死区寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DTRx<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
DTRx<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 15-14 未实现: 读为 0
bit 13-0 DTRx<13:0>: PWMx 死区单元的无符号 14 位死区值位

寄存器 14-12: ALTDTRx: PWMx 备用死区寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	ALTDTRx<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
ALTDTR <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 15-14 未实现: 读为 0
bit 13-0 ALTDTRx<13:0>: PWMx 死区单元的无符号 14 位死区值位

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-13: TRGCONx: PWMx 触发器控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
TRGDIV<3:0>				—	—	—	—
bit 15				bit 8			

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DTM ⁽¹⁾	—	TRGSTRT<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 15-12 **TRGDIV<3:0>**: 触发信号输出分频比位

0000 = 每个触发事件输出一次触发信号
 0001 = 每 2 个触发事件输出一次触发信号
 0010 = 每 3 个触发事件输出一次触发信号
 0011 = 每 4 个触发事件输出一次触发信号
 0100 = 每 5 个触发事件输出一次触发信号
 0101 = 每 6 个触发事件输出一次触发信号
 0110 = 每 7 个触发事件输出一次触发信号
 0111 = 每 8 个触发事件输出一次触发信号
 1000 = 每 9 个触发事件输出一次触发信号
 1001 = 每 10 个触发事件输出一次触发信号
 1010 = 每 11 个触发事件输出一次触发信号
 1011 = 每 12 个触发事件输出一次触发信号
 1100 = 每 13 个触发事件输出一次触发信号
 1101 = 每 14 个触发事件输出一次触发信号
 1110 = 每 15 个触发事件输出一次触发信号
 1111 = 每 16 个触发事件输出一次触发信号

bit 11-8 **未实现**: 读为 0

bit 7 **DTM**: 双触发器模式位 ⁽¹⁾

1 = 辅助触发事件和主触发事件一起产生 PWM 触发信号。

0 = 辅助触发事件不同主触发事件一起产生 PWM 触发信号。而是产生两个独立的 PWM 触发信号。

bit 6 **未实现**: 读为 0

bit 5-0 **TRGSTRT<5:0>**: 触发器后分频比启动使能选择位

000000 = 在模块使能后等待 0 个 PWM 周期产生第一个触发事件
 000001 = 在模块使能后等待 1 个 PWM 周期产生第一个触发事件
 000010 = 在模块使能后等待 1 个 PWM 周期产生第一个触发事件

•
•
•

111111 = 在模块使能后等待 63 个 PWM 周期产生第一个触发事件

注 1: 辅助发生器不能产生 PWM 触发中断。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-14: IOCONx: PWMx I/O 控制寄存器

R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0			
PENH		PENL		POLH		POLL		PMOD<1:0> ⁽¹⁾				OVRENH		OVRENL	
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
OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		SWAP	OSYNC
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **PENH:** PWMH 输出引脚所有权位
1 = 由 PWM 模块控制 PWMxH 引脚
0 = 由 GPIO 模块控制 PWMxH 引脚
- bit 14 **PENL:** PWML 输出引脚所有权位
1 = 由 PWM 模块控制 PWMxL 引脚
0 = 由 GPIO 模块控制 PWMxL 引脚
- bit 13 **POLH:** PWMH 输出引脚极性位
1 = PWMxH 引脚低电平有效
0 = PWMxH 引脚高电平有效
- bit 12 **POLL:** PWML 输出引脚极性位
1 = PWMxL 引脚低电平有效
0 = PWMxL 引脚高电平有效
- bit 11-10 **PMOD<1:0>:** PWM I/O 引脚模式位 ⁽¹⁾
00 = PWM I/O 引脚对处于互补输出模式
01 = PWM I/O 引脚对处于冗余输出模式
10 = PWM I/O 引脚对处于推挽输出模式
11 = PWM I/O 引脚对处于真正独立 PWM 输出模式
- bit 9 **OVRENH:** PWMxH 引脚改写使能位
1 = OVRDAT<1> 为 PWMxH 引脚提供输出数据
0 = 由 PWM 发生器为 PWMxH 引脚提供数据
- bit 8 **OVRENL:** PWMxL 引脚改写使能位
1 = OVRDAT<0> 为 PWMxL 引脚提供输出数据
0 = 由 PWM 发生器为 PWMxL 引脚提供数据
- bit 7-6 **OVRDAT<1:0>:** 使能改写时 PWMxH 和 PWMxL 引脚的数据位
如果 OVRRENH = 1, 则 OVRDAT<1> 为 PWMxH 提供数据。
如果 OVRRENL = 1, 则 OVRDAT<0> 为 PWMxL 提供数据。
- bit 5-4 **FLTDAT<1:0>:** 使能 FLTMOD 时 PWMxH 和 PWMxL 引脚的数据位
FCLCONx<IFLTMOD> = 0: 普通故障模式:
如果故障有效, 则 FLTDAT<1> 为 PWMxH 提供数据。
如果故障有效, 则 FLTDAT<0> 为 PWMxL 提供数据。
FCLCONx<IFLTMOD> = 1: 独立故障模式:
如果限流有效, 则 FLTDAT<1> 为 PWMxH 提供数据。
如果故障有效, 则 FLTDAT<0> 为 PWMxL 提供数据。

注 1: 仅当 PTEN = 0 时才能改变这些位。在工作期间改变时钟选择将产生不可预知的结果。

寄存器 14-14: IOCONx: PWMx I/O 控制寄存器 (续)

- bit 3-2 **CLDAT<1:0>**: 使能 CLMODE 时 PWMxH 和 PWMxL 引脚的数据位
FCLCONx<IFLTMOD> = 0: 普通故障模式:
如果限流有效, 则 CLDAT<1> 为 PWMxH 提供数据。
如果限流有效, 则 CLDAT<0> 为 PWMxL 提供数据。
FCLCONx<IFLTMOD> = 1: 独立故障模式:
CLDAT<1:0> 被忽略。
- bit 1 **SWAP<1:0>**: 交换 PWMxH 和 PWMxL 引脚
1 = PWMxH 输出信号和 PWMxL 引脚连接, PWMxL 输出信号和 PWMxH 引脚连接
0 = PWMxH 和 PWMxL 引脚被映射到它们各自的引脚上
- bit 0 **OSYNC**: 输出改写同步位
1 = 通过 OVRDAT<1:0> 位进行的输出改写与 PWM 时基同步
0 = 通过 OVRDAT<1:0> 位进行的输出改写在下一个 CPU 时钟的边界处发生

注 1: 仅当 PTEN = 0 时才能改变这些位。在工作期间改变时钟选择将产生不可预知的结果。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-15: FCLCONx: PWMx 故障限流控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IFLTMOD	CLSRC<4:0> ^(2,3)					CLPOL ⁽¹⁾	CLMOD
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
FLTSRC<4:0> ^(2,3)					FLTPOL ⁽¹⁾	FLTMOD<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

IFLTMOD: 独立故障模式允许位
1 = 独立故障模式: 限流输入将 FLTDAT<1> 映射到 PWMxH 输出, 故障输入将 FLTDAT<0> 映射到 PWMxL 输出。CLDAT<1:0> 位不用作改写功能。
0 = 普通故障模式: 限流特性将 CLDAT<1:0> 映射到 PWMxH 和 PWMxL 输出。PWM 故障特性将 FLTDAT<1:0> 映射到 PWMxH 和 PWMxL 输出。
- bit 14-10

CLSRC<4:0>: PWM 发生器 x 的限流控制信号源选择位 ^(2,3)
00000 = 故障 1
00001 = 故障 2
00010 = 故障 3
00011 = 故障 4
00100 = 故障 5
00101 = 故障 6
00110 = 故障 7
00111 = 故障 8

01000 = 保留
•
•
•

11111 = 保留
- bit 9

CLPOL: PWM 发生器 x 的限流极性位 ⁽¹⁾
1 = 所选限流源为低电平有效
0 = 所选限流源为高电平有效
- bit 8

CLMOD: PWM 发生器 x 的限流模式使能位
1 = 允许限流功能
0 = 禁止限流功能

注 1: 仅当 PTEN = 0 时才能改变这些位。在工作期间改变时钟选择将产生不可预知的结果。

2: 当使能独立故障模式 (IFLTMOD = 1) 并且故障 1 用于限流模式 (CLSRC<4:0> = b0000) 时, 故障控制源选择位 (FLTSRC<4:0>) 必须设置为未使用故障源, 用于防止故障 1 同时禁止了 PWMxL 和 PWMxH 输出。

3: 当使能独立故障模式 (IFLTMOD = 1) 并且故障 1 用于故障模式 (FLTSRC<4:0> = b0000) 时, 限流控制源选择位 (CLSRC<4:0>) 必须设置为未使用限流源, 用于防止限流源同时禁止了 PWMxH 和 PWMxL 输出。

寄存器 14-15: FCLCONx: PWMx 故障限流控制寄存器 (续)

bit 7-3	FLTSRC<4:0> : PWM 发生器 x 的故障控制信号源选择位 (2,3) 00000 = 故障 1 00001 = 故障 2 00010 = 故障 3 00011 = 故障 4 00100 = 故障 5 00101 = 故障 6 00110 = 故障 7 00111 = 故障 8 01000 = 保留 . . . 11111 = 保留
bit 2	FLTPOL : PWM 发生器 x 的故障极性位 (1) 1 = 所选故障源为低电平有效 0 = 所选故障源为高电平有效
bit 1-0	FLTMOD<1:0> : PWM 发生器 x 的故障模式位 00 = 所选故障源强制 PWMxH 和 PWMxL 引脚为 FLTDAT 值 (锁定模式) 01 = 所选故障源强制 PWMxH 和 PWMxL 引脚为 FLTDAT 值 (周期模式) 10 = 保留 11 = 禁止故障输入

- 注**
- 1: 仅当 PTEN = 0 时才能改变这些位。在工作期间改变时钟选择将产生不可预知的结果。
 - 2: 当使能独立故障模式 (IFLTMOD = 1) 并且故障 1 用于限流模式 (CLSRC<4:0> = b0000) 时, 故障控制源选择位 (FLTSRC<4:0>) 必须设置为未使用故障源, 用于防止故障 1 同时禁止了 PWMxL 和 PWMxH 输出。
 - 3: 当使能独立故障模式 (IFLTMOD = 1) 并且故障 1 用于故障模式 (FLTSRC<4:0> = b0000) 时, 限流控制源选择位 (CLSRC<4:0>) 必须设置为未使用限流源, 用于防止限流源同时禁止了 PWMxH 和 PWMxL 输出。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-16: TRIGx: PWMx 主触发器比较值寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TRGCMP<15:8>							
bit 15							
bit 8							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
TRGCMP<7:3>					—	—	—
bit 7							
bit 0							

图注:							
R = 可读位		W = 可写位		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 **TRGCMP<15:3>**: 触发器控制值位
当主 PWM 工作在本地时基时, 此寄存器包含比较值, 可以触发 ADC 模块。

bit 2-0 未实现: 读为 0

寄存器 14-17: STRIGx: PWMx 辅助触发器比较值寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STRGCMP<15:8>							
bit 15							
bit 8							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
STRGCMP<7:3>					—	—	—
bit 7							
bit 0							

图注:							
R = 可读位		W = 可写位		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 **STRGCMP<15:3>**: 辅助触发器控制值位
当辅助 PWM 工作在本地时基时, 此寄存器包含比较值, 可以触发 ADC 模块。

bit 2-0 未实现: 读为 0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-18: LEBCONx: 前沿消隐控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:8>	
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
LEB<7:3>					—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **PHR:** PWMxH 上升沿触发使能位
1 = PWMxH 上升沿触发 LEB 计数器
0 = LEB 忽略 PWMxH 上升沿
- bit 14 **PHF:** PWMxH 下降沿触发使能位
1 = PWMxH 下降沿触发 LEB 计数器
0 = LEB 忽略 PWMxH 下降沿
- bit 13 **PLR:** PWMxL 上升沿触发使能位
1 = PWMxL 上升沿触发 LEB 计数器
0 = LEB 忽略 PWMxL 上升沿
- bit 12 **PLF:** PWMxL 下降沿触发使能位
1 = PWMxL 下降沿触发 LEB 计数器
0 = LEB 忽略 PWMxL 下降沿
- bit 11 **FLTLEBEN:** 故障输入 LEB 使能位
1 = 将前沿消隐应用于所选故障输入
0 = 前沿消隐未用于所选故障输入
- bit 10 **CLLEBEN:** 限流 LEB 使能位
1 = 将前沿消隐应用于所选限流输入
0 = 前沿消隐未用于所选限流输入
- bit 9-3 **LEB:** 限流输入和故障输入的前沿消隐位
值以 8 ns 为单位进行递增。
- bit 2-0 **未实现:** 读为 0

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 14-19: PWMCAPx: 主 PWMx 时基捕捉寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PWMCAP<15:8> ^(1,2)							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	U-0	U-0	U-0
PWMCAP<7:3> ^(1,2)					—	—	—
bit 7							bit 0

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

bit 15-3 **PWMCAP<15:3>:** 捕捉的 PWM 时基值位 ^(1,2)
此寄存器中的值表示当检测到限流输入的一个前沿时捕捉到的 PWM 时基值。

bit 2-0 **未实现:** 读为 0

- 注 **1:** 此捕捉特性仅在主输出 (PWMxH) 上可用。
- 2:** 此特性仅在限流输入信号的前沿消隐处理结束后才有效。

15.0 串行外设接口 (SPI)

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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) 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。

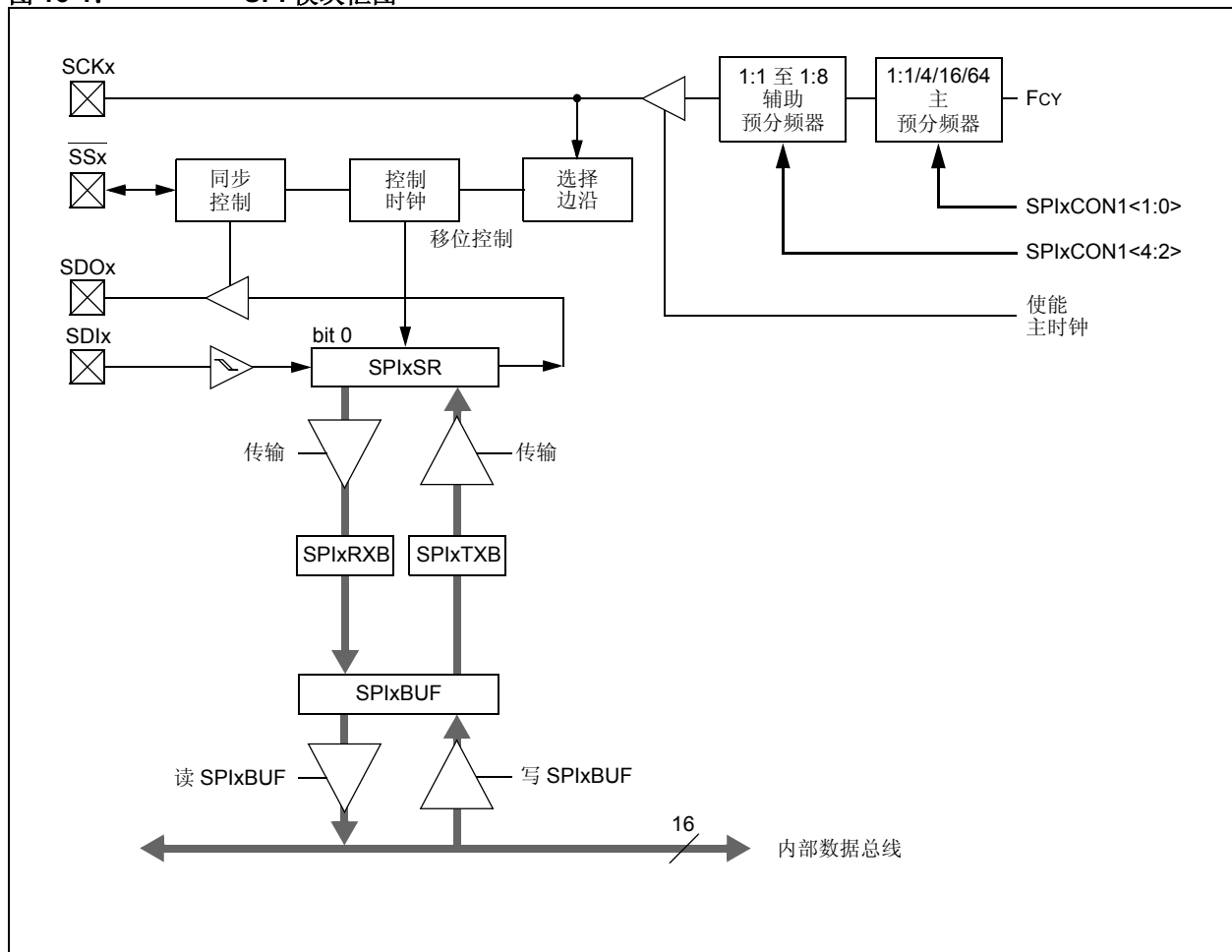
SPI 模块由一个用于将数据移入和移出的 16 位移位寄存器 SPIxSR (其中 $x = 1$) 和一个缓冲寄存器 SPIxBUF 组成。控制寄存器 SPIxCON 用来配置模块。另外，还有一个状态寄存器 SPIxSTAT 用来表明各种状态条件。

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；在从模式下工作时，SCK 是时钟输入。

图 15-1: SPI 模块框图



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 15-1: SPIxSTAT: SPIx 状态和控制寄存器

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

- bit 15

SPIEN: SPIx 使能位
1 = 使能模块并将 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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 15-2: SPIxCON1: SPIx 控制寄存器 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE ⁽¹⁾
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 ⁽¹⁾

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 = SSx 引脚用于从模式

0 = 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 15-2: SPIxCON1: SPIx 控制寄存器 1 (续)

bit 4-2 **SPRE<2:0>**: 辅助预分频比位 (主模式)
 111 = 辅助预分频比 1:1
 110 = 辅助预分频比 2:1
 .
 .
 .
 000 = 辅助预分频比 8:1

bit 1-0 **PPRE<1:0>**: 主预分频比位 (主模式)
 11 = 主预分频比 1:1
 10 = 主预分频比 4:1
 01 = 主预分频比 16:1
 00 = 主预分频比 64:1

注 1: 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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 15-3: SPIxCON2: SPIx 控制寄存器 2

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	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 ($\overline{SSx}$ 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。

注:

16.0 I²C™

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 (I²C™)**” (DS70195)，该文档可从 Microchip 网站 (www.microchip.com) 下载。

16 位接口的 I²C 模块为 I²C 串行通信标准的从模式和主器件模式提供完全的硬件支持。

I²C 模块有一个双引脚接口：

- SCLx 引脚是时钟线。
- SDAx 引脚是数据线。

I²C 模块提供以下主要特性：

- I²C 接口支持主 / 从工作模式。
- I²C 从模式支持 7 位和 10 位寻址。
- I²C 主模式支持 7 位和 10 位寻址。
- I²C 端口允许主器件和从器件之间的双向传输。
- I²C 端口的串行时钟同步可以用作握手机制来暂停和恢复串行传输 (SCLREL 控制)。
- I²C 支持多主器件工作；检测总线冲突并相应地进行仲裁。

16.1 工作模式

硬件完全实现了 I²C 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 I²C 总线上，I²C 模块既可作为主器件工作，也可作为从器件工作。

支持 I²C 模块工作在以下模式：

- 带 7 位寻址的 I²C 从模式
- 带 10 位寻址的 I²C 从模式
- 带 7 位或 10 位寻址的 I²C 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) 获得最新的“*dsPIC33F Family Reference Manual*”章节。

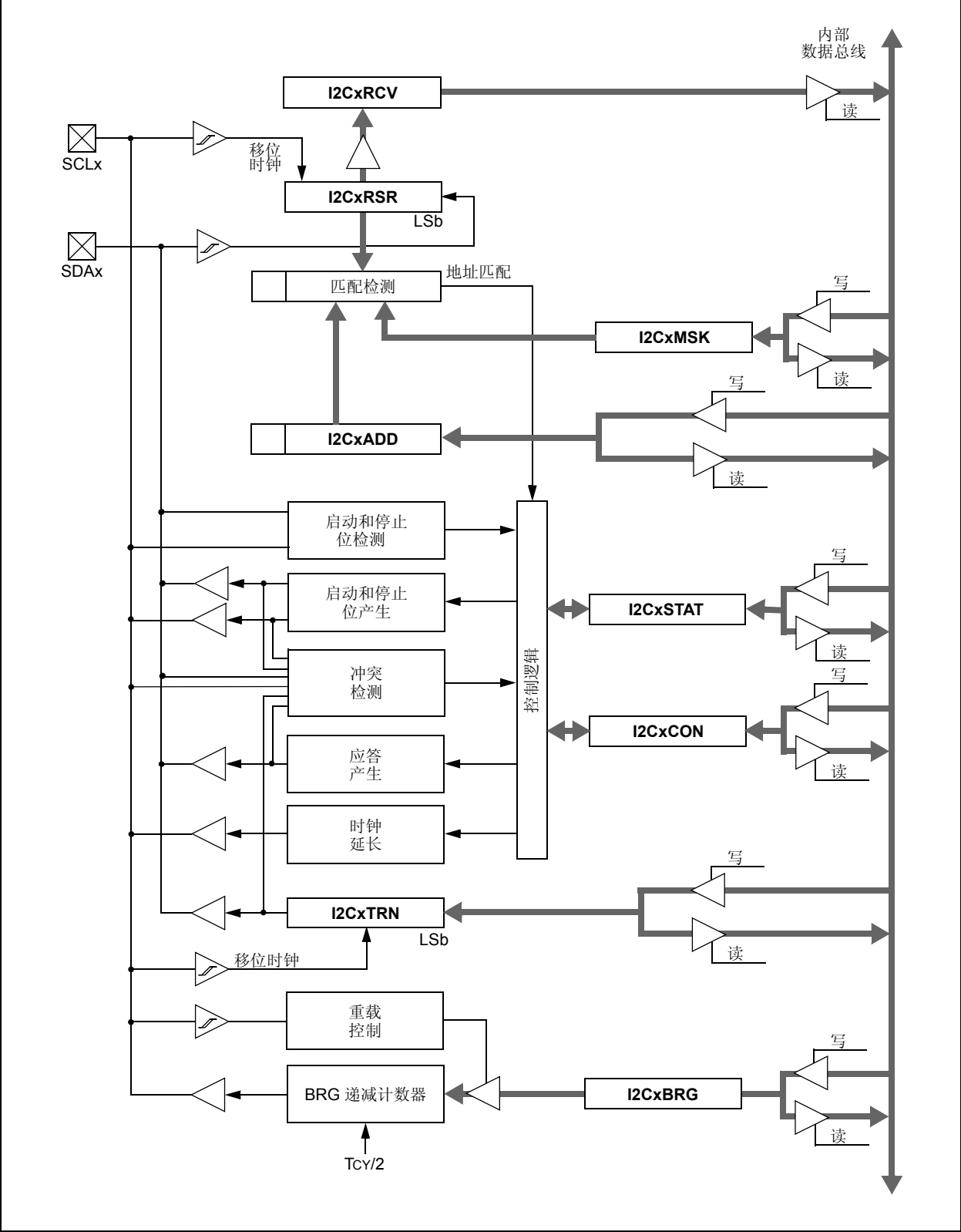
16.2 I²C 寄存器

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。

图 16-1: I²C™ 框图 (x = 1)



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 16-1: I2CxCON: I2Cx 控制寄存器

R/W-0	U-0	R/W-0	R/W-1, HC	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0, HC	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

图注:	U = 未实现位, 读为 0		
R = 可读位	W = 可写位	HS = 可硬件置 1 位	HC = 可硬件清零位
-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 ² 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 ² 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 位从地址位 1 = I2CxADD 为 10 位从地址 0 = I2CxADD 为 7 位从地址
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 引脚阈值 0 = 禁止 SMBus 输入阈值
bit 7	GCEN: 广播呼叫使能位 (作为 I ² 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 ² 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 16-1: I2CxCON: I2Cx 控制寄存器 (续)

- bit 5 **ACKDT:** 应答数据位 (作为 I²C 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 I²C 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 I²C 主器件工作时)
1 = 使能 I²C 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 I²C 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 I²C 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 I²C 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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 16-2: I2CxSTAT: I2Cx 状态寄存器

R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HSC	R-0, HSC	R-0, HSC
ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
bit 15						bit 8	

R/C-0, HS	R/C-0, HS	R-0, HSC	R/C-0, HSC	R/C-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF
bit 7						bit 0	

图注:	U = 未实现位, 读为 0		
R = 可读位	W = 可写位	HS = 可硬件置 1 位	HSC = 可硬件置 1/ 清零位
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

- bit 15 ACKSTAT:** 应答状态位
(作为 I²C 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 I²C 主器件工作时, 适用于主器件发送操作)
1 = 主器件正在进行发送 (8 位 + ACK)
0 = 主器件不在进行发送
在主器件发送开始时由硬件置 1。在从器件应答结束时由硬件清零。
- bit 13-11 未实现:** 读为 0
- bit 10 BCL:** 主器件总线冲突检测位
1 = 主器件工作期间检测到了总线冲突
0 = 未发生冲突
检测到总线冲突时由硬件置 1。
- bit 9 GCSTAT:** 广播呼叫状态位
1 = 接收到广播呼叫地址
0 = 未接收到广播呼叫地址
当地址与广播呼叫地址匹配时由硬件置 1。当检测到停止条件时由硬件清零。
- bit 8 ADD10:** 10 位地址状态位
1 = 10 位地址匹配
0 = 10 位地址不匹配
当与匹配的 10 位地址的第 2 个字节匹配时由硬件置 1。当检测到停止条件时由硬件清零。
- bit 7 IWCOL:** 写冲突检测位
1 = 因为 I²C 模块忙, 尝试写 I2CxTRN 寄存器失败
0 = 未发生冲突
当模块忙时写 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 I²C 从器件工作时)
1 = 表示上次接收的字节为数据
0 = 表示上次接收的字节为器件地址
器件地址匹配时由硬件清零。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 16-2: 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 I²C 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 I²C 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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 16-3: I2CxMSK: I2Cx 从模式地址掩码寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	AMSK<9:8>	
bit 15						bit 8	
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AMSK<7:0>							
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-10

未实现: 读为 0

bit 9-0

AMSK<9:0>: 地址中 bit x 的掩码选择位

1 = 使能输入报文的地址中 bit x 的掩码; 在此位置上不需要位匹配

0 = 禁止 bit x 的掩码; 在此位置上需要位匹配

注:

17.0 通用异步收发器 (UART)

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件系列提供的串行 I/O 模块之一。UART 是可以与外设 (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：

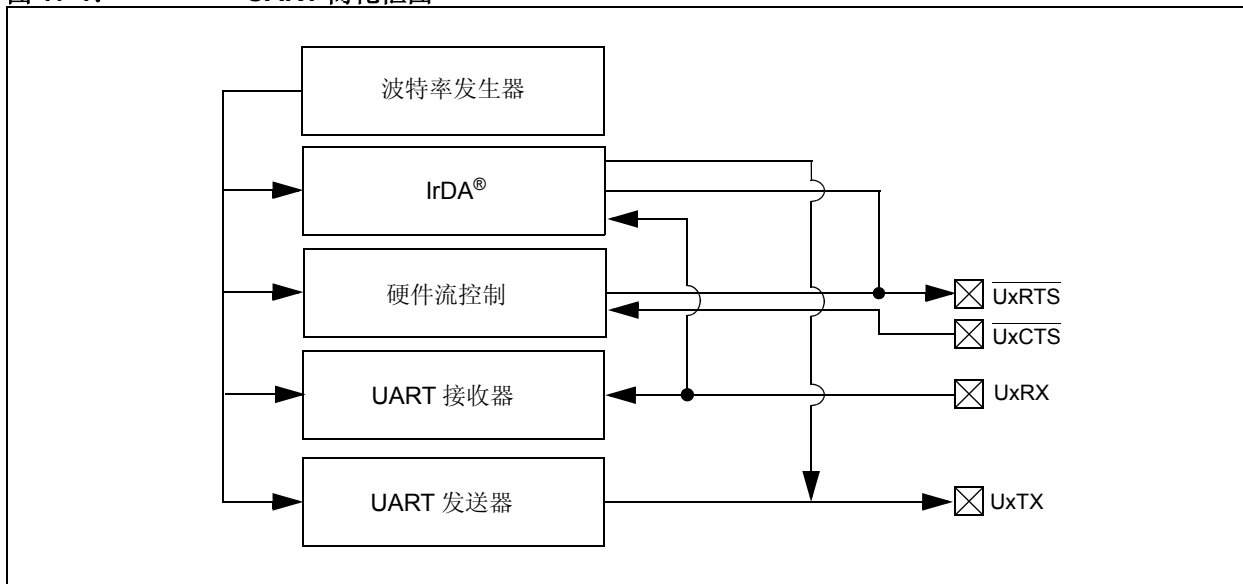
- 通过 UxTX 和 UxRX 引脚进行全双工 8 位或 9 位数据传输
- 偶校验、奇校验或无奇偶校验选项 (对于 8 位数据)
- 一个或两个停止位

- 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 40 MIPS 下 16x 模式时，波特率范围为从 1 Mbps 到 15 Mbps
- 当器件工作在 40 MIPS 下 4x 模式时，波特率范围从 4 Mbps 到 61 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
- 支持同步字符和间隔字符
- 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 16x 波特率时钟输出
- 支持 DMA

图 17-1 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 17-1: UART 简化框图



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 17-1: UxMODE: UARTx 模式寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
UARTEN	—	USIDL	IREN ⁽¹⁾	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	

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

- bit 15 **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 ⁽¹⁾
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 16x BRG 模式 (BRGH = 0) 下使用。

寄存器 17-1: 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 (4x 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 (16x 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 16x BRG 模式 (BRGH = 0) 下使用。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 17-2: UxSTA: UARTx 状态和控制寄存器

R/W-0	R/W-0	R/W-0	U-0	R/W-0, HC	R/W-0	R-0	R-1
UTXISEL1	UTXINV	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	

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

bit 15,13	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 : 发送极性翻转位 1 = UxTX 空闲状态为 1 0 = UxTX 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, 缓冲区被复位; 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 = 当接收到任何一个字符且将字符从 UxRSR 传输到接收缓冲区时, 中断标志位置 1; 接收缓冲区有一个或多个字符。
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
bit 4	RIDLE : 接收器空闲位 (只读) 1 = 接收器空闲 0 = 接收器工作

寄存器 17-2: UxSTA: UARTx 状态和控制寄存器 (续)

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)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, 至少还有一个字符可被读取 0 = 接收缓冲区为空

注:

18.0 高速 10 位模数转换器 (ADC)

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 44. High-Speed 10-bit Analog-to-Digital Converter (ADC)**” (DS70321)，该文档可从 Microchip 网站 (www.microchip.com) 下载。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件提供高速的逐次逼近模数转换，支持 AC/DC 和 DC/DC 电源转换器等应用。

18.1 特性概述

ADC 模块具有以下特性：

- 10 位分辨率
- 单极性输入
- 最多两个逐次逼近寄存器 (Successive Approximation Register, SAR)
- 最多 12 路外部输入通道
- 最多两路内部模拟输入
- 用于每路模拟输入的专用结果寄存器
- 3.3V 时精度为 ± 1 LSB
- 单电源工作
- 3.3V 时转换速率为 4 Msps (对于带两个 SAR 的器件)
- 3.3V 时转换速率为 2 Msps (对于带一个 SAR 的器件)
- 低功耗 CMOS 技术

18.2 模块说明

此 ADC 模块是为要求从请求转换到产生输出数据之间的延时比较低的应用设计的。典型应用包括：

- AC/DC 电源
- DC/DC 转换器
- 功率因数校正 (Power Factor Correction, PFC)

此 ADC 在需要高频控制环的电源控制应用中与高速 PWM 模块配合使用。当使用两个 SAR 时，此模块可在 0.5 微秒内对两路模拟输入进行采样和转换。这一较小的转换延时减小了从测量到控制系统响应之间的“相位滞后”。

可同时对最多五路输入进行采样 (其中四路输入来自专用的采样 / 保持电路，一路输入来自共用的采样 / 保持电路)。如果有多路输入请求转换，ADC 将从编号最低的输入开始按顺序进行转换。

此 ADC 设计为每对模拟输入 ((AN1,AN0), (AN3,AN2),...) 提供了从最多 16 个不同触发源中指定自身触发源的功能。此功能允许此 ADC 对与依靠独立时基工作的 PWM 发生器相关联的模拟输入进行采样和转换。

用户应用通常要求模拟数据采样与到应用电路的 PWM 输出同步。此 ADC 模块的极高速工作允许即时获得数据。

此外，为外设接口添加了几个硬件特性，来提高典型 DSP 应用的实时性能。

- 结果对齐选项
- 自动采样
- 外部转换启动控制
- 用于监控 1.2V 内部参考电压和 EXTREF 输入信号的两路内部输入

ADC 模块的框图如图 18-6 所示。

18.3 模块功能

高速 10 位 ADC 设计为与高速 PWM 模块配合使用来支持电源转换应用。ADC 可能具有一个或两个 SAR 模块，这取决于具体的器件型号。如果器件中包含两个 SAR，则可同时处理两路转换，达到 4 Msps 的转换速率。如果器件中只包含一个 SAR，则一次只能处理一路转换，仅能达到 2 Msps 的转换速率。高速 10 位 ADC 可在 0.5 微秒内产生两个 10 位转换结果。

ADC 模块支持最多 12 路外部模拟输入以及两路内部模拟输入。为监控参考电压，将两路内部输入 AN12 和 AN13 分别连接到 EXTREF 和内部带隙电压（1.2V）。模拟参考电压定义为器件的供电电压（AVDD / AVSS）。

ADC 模块使用以下控制和状态寄存器：

- ADCON：A/D 控制寄存器
- ADSTAT：A/D 状态寄存器
- ADBASE：A/D 基址寄存器 (1,2)
- ADPCFG：A/D 端口配置寄存器
- ADCPC0：A/D 转换对控制寄存器 0
- ADCPC1：A/D 转换对控制寄存器 1
- ADCPC2：A/D 转换对控制寄存器 2(1)
- ADCPC3：A/D 转换对控制寄存器 3(1)

ADCON 寄存器控制 ADC 模块的工作。ADSTAT 寄存器用于显示转换过程的状态。ADPCFG 寄存器将端口引脚配置为模拟输入或数字 I/O。ADCPCx 寄存器控制 ADC 转换的触发。详细的位配置请参见寄存器 18-1 到寄存器 18-8。

注： ADC 模块的一个独特特性是其以异步方式对输入进行采样的能力。可对各采样 / 保持电路进行独立触发。

图 18-1: 带一个 SAR 的 dsPIC33FJ06GS101 器件的 ADC 框图

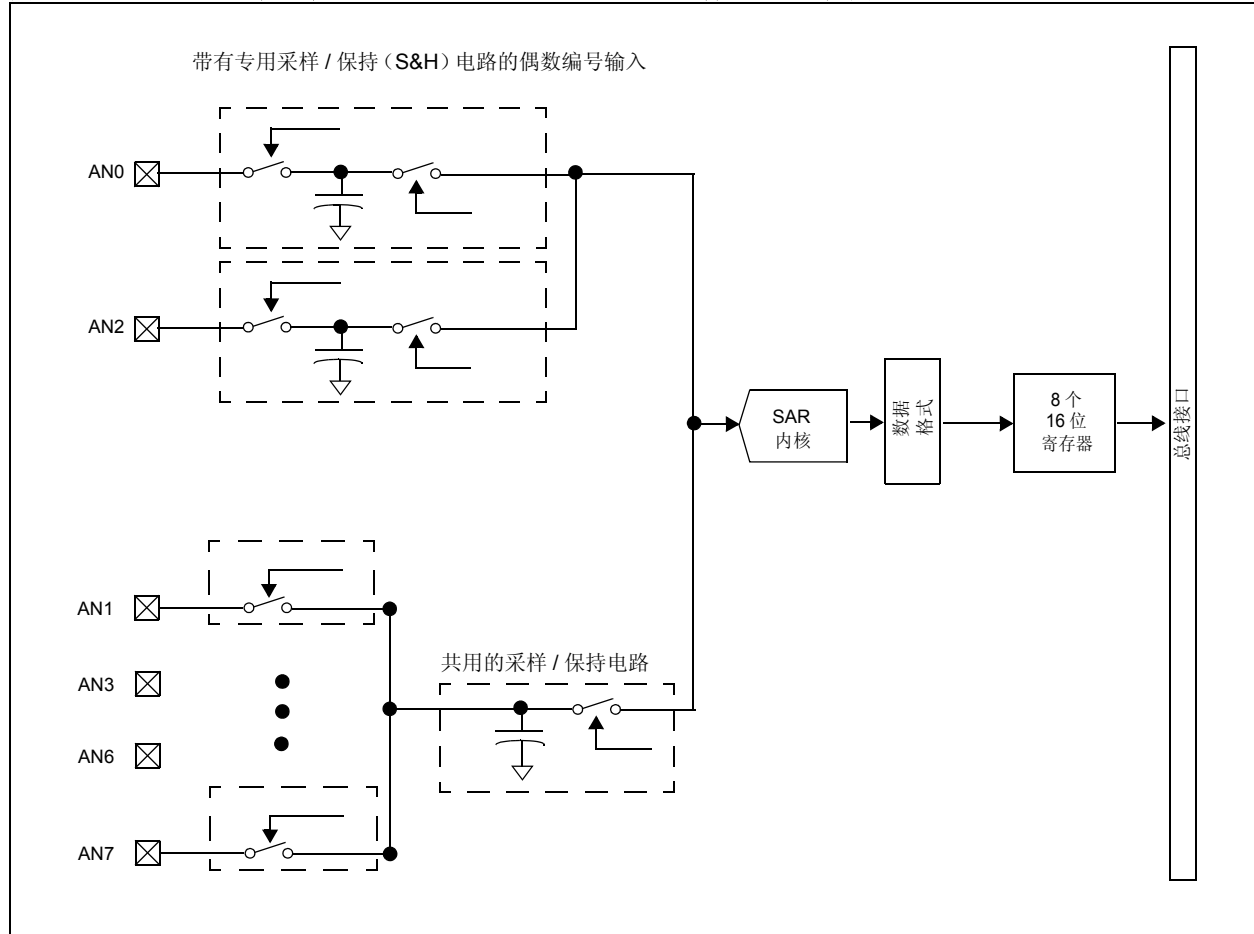


图 18-2: 带一个 SAR 的 dsPIC33FJ06GS102 器件的 ADC 框图

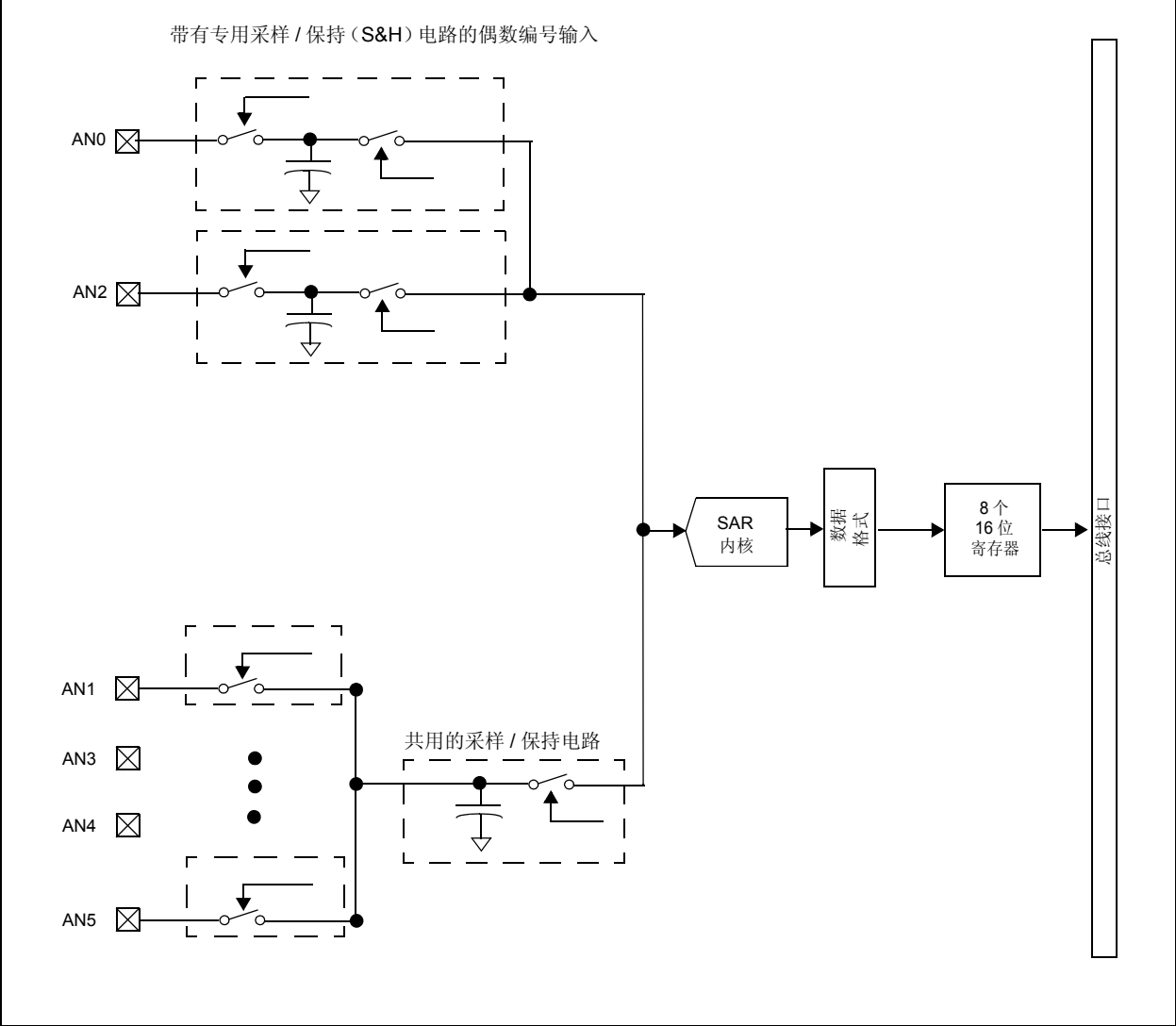


图 18-3: 带一个 SAR 的 dsPIC33FJ06GS202 器件的 ADC 框图

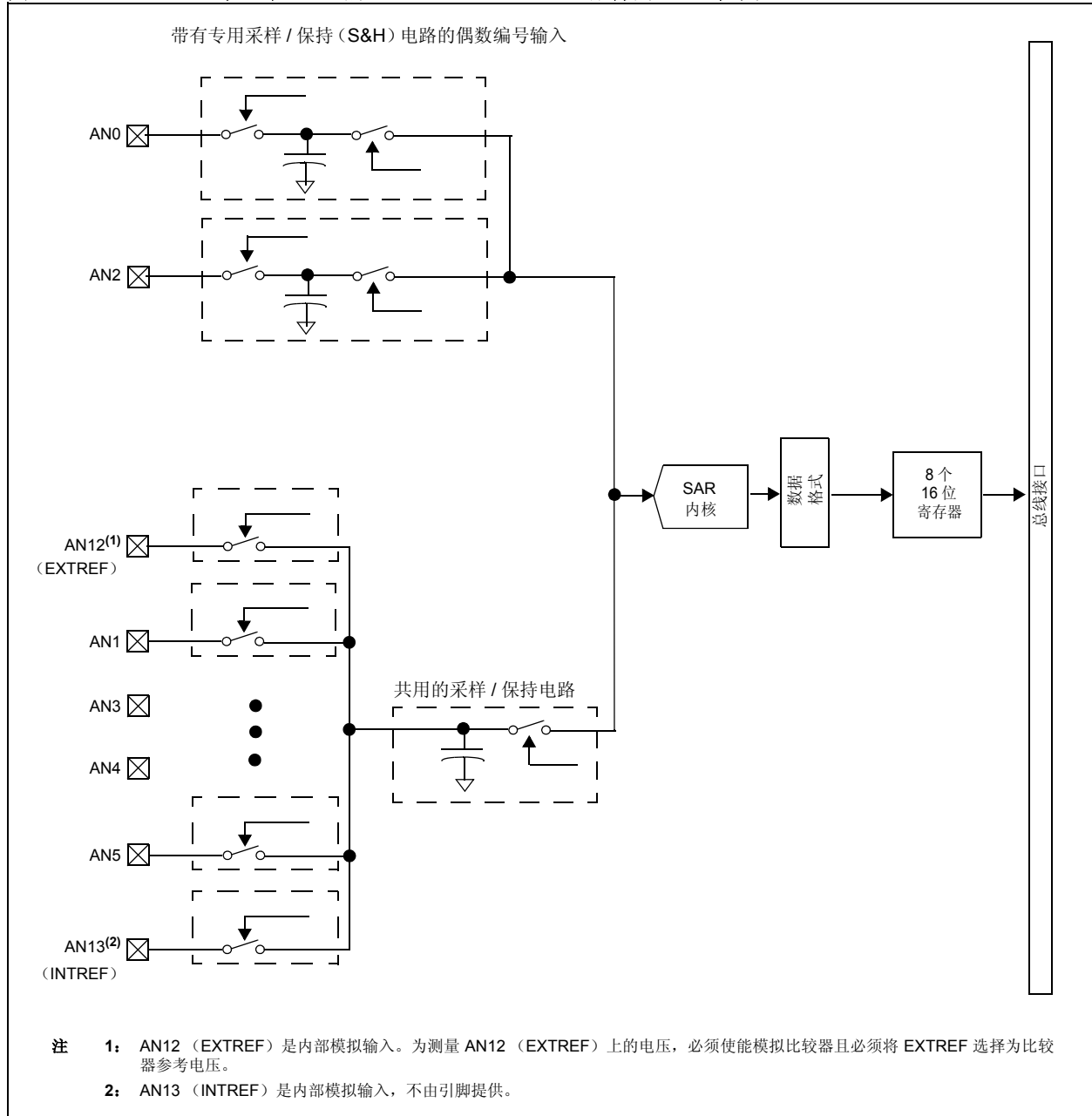


图 18-4: 带一个 SAR 的 dsPIC33FJ16GS402/404 器件的 ADC 框图

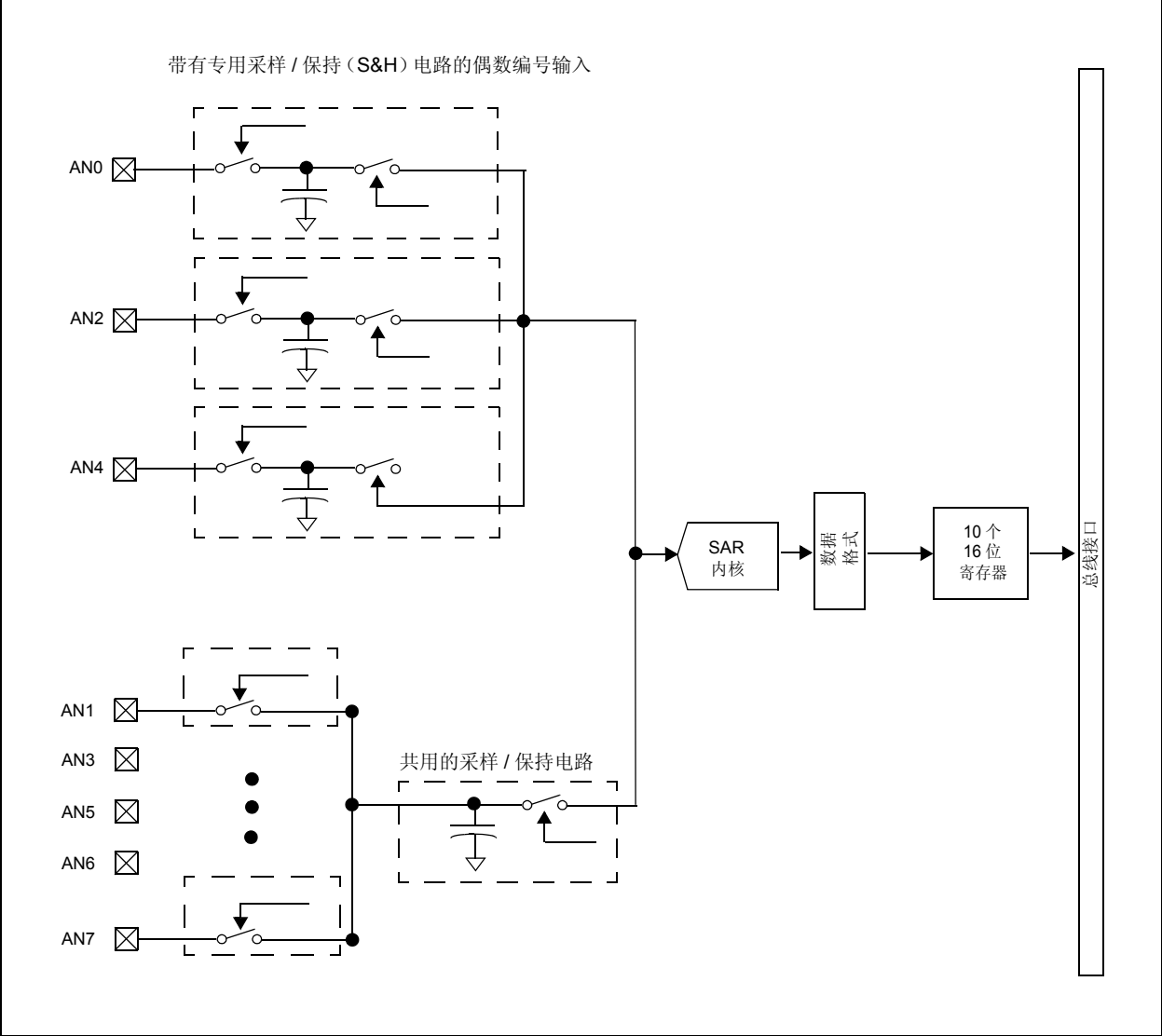


图 18-5: 带两个 SAR 的 dsPIC33FJ16GS502 器件的 ADC 框图

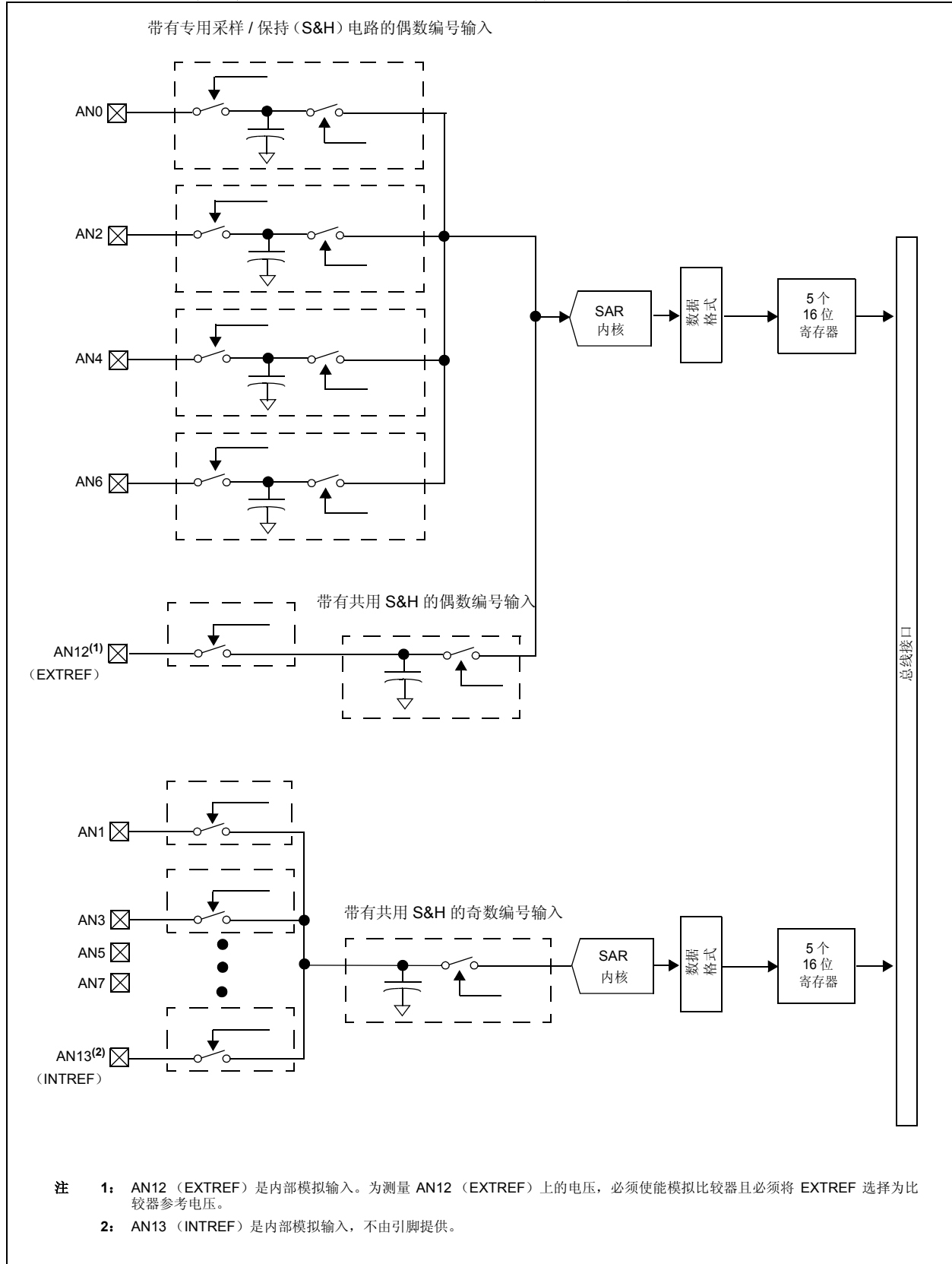
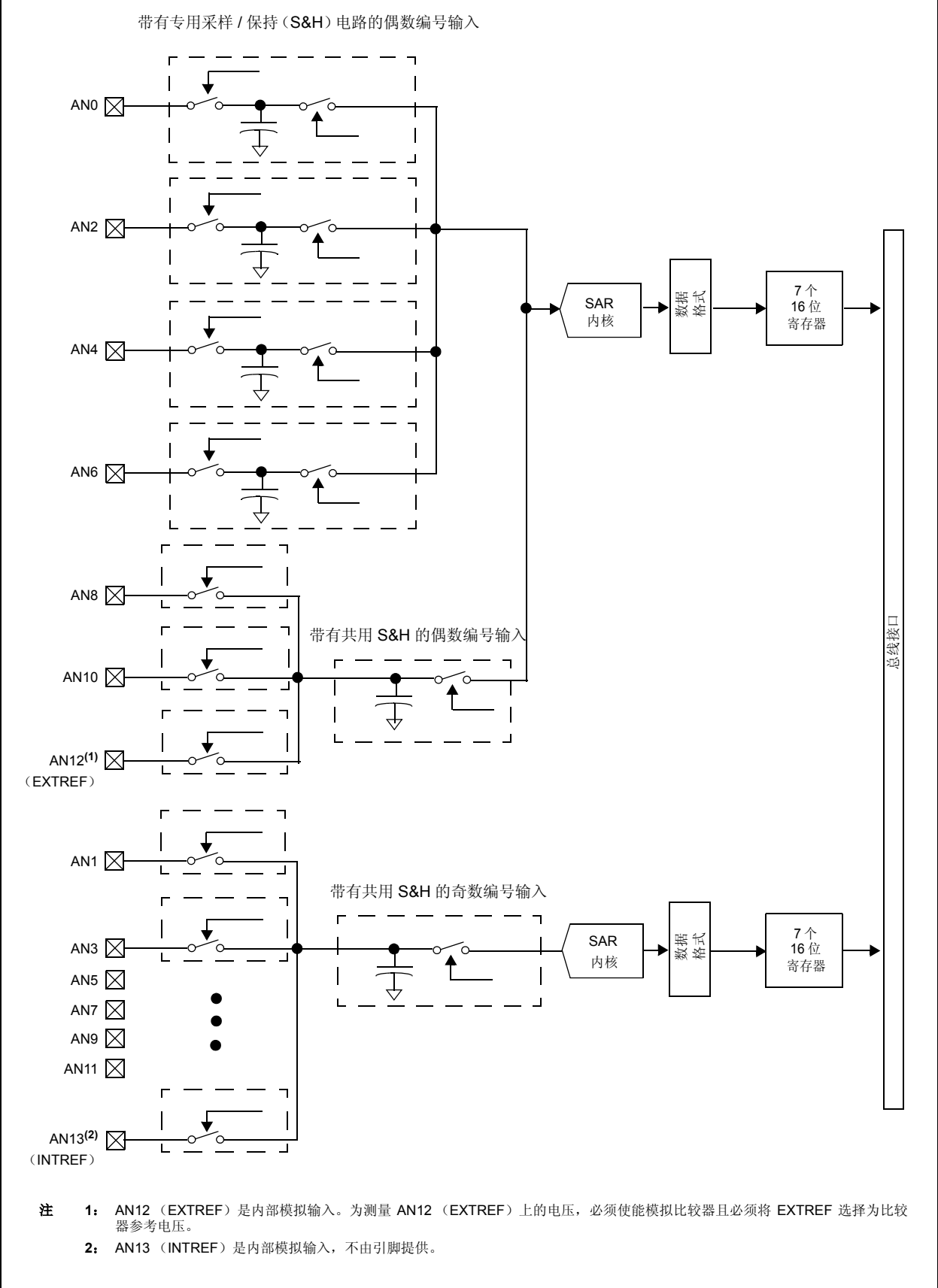


图 18-6: 带两个 SAR 的 dsPIC33FJ16GS504 器件的 ADC 框图



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 18-1: **ADCON: A/D 控制寄存器**

R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0
ADON	—	ADSIDL	SLOWCLK ⁽¹⁾	—	GSWTRG	—	FORM ⁽¹⁾
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-1	R/W-1
EIE ⁽¹⁾	ORDER ⁽¹⁾	SEQSAMP ⁽¹⁾	ASYNCSAMP ⁽¹⁾	—	ADCS<2:0> ⁽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 **ADON: A/D 工作模式位**
 1 = A/D 转换器模块在工作
 0 = A/D 转换器关闭
- 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 **SLOWCLK: 低频时钟分频比使能位 ⁽¹⁾**
 1 = 使用针对低频 ADC 输入时钟的时钟分频比。此分频比提供低频分频比用来在较慢的输入时钟下获得速度最快的 ADC 操作。
 0 = 使用针对高频 ADC 输入时钟的时钟分频比。此分频比提供高分频比。
- bit 11 **未实现:** 读为 0
- bit 10 **GSWTRG: 全局软件触发位**
 当此位被用户置 1 时, 如果通过 ADCPCx 寄存器中的 TRGSRC<4:0> 位选择了全局软件触发源, 将触发转换。在启用另一个全局触发源之前, 此位必须由用户清零 (即, 此位不是自动清零的)。
- bit 9 **未实现:** 读为 0
- bit 8 **FORM: 数据输出格式位 ⁽¹⁾**
 1 = 小数 (DOUT = dddd dddd dd00 0000)
 0 = 整数 (DOUT = 0000 00dd dddd dddd)
- bit 7 **EIE: 提前中断允许位 ⁽¹⁾**
 1 = 在第一次转换完成后产生中断
 0 = 在第二次转换完成后产生中断
- bit 6 **ORDER: 转换顺序位 ⁽¹⁾**
 1 = 先转换奇数编号的模拟输入, 然后转换偶数编号的输入
 0 = 先转换偶数编号的模拟输入, 然后转换奇数编号的输入
- bit 5 **SEQSAMP: 顺序采样使能位 ⁽¹⁾**
 1 = 如果 ORDER = 0, 在第二次转换开始时对共用采样 / 保持 (S&H) 电路进行采样。如果 ORDER = 1, 则在第一次转换开始时对共用 S&H 电路进行采样。
 0 = 如果在现有转换过程中共用 S&H 不处于忙状态, 则在采样专用 S&H 的同时对共用 S&H 进行采样。如果在采样专用 S&H 时共用 S&H 处于忙状态, 则在新转换周期开始时对共用 S&H 进行采样。

注 1: 仅可在禁止 ADC (ADON = 0) 时更改此控制位。

寄存器 18-1: **ADCON: A/D 控制寄存器** (续)

bit 4 **ASYNCSAMP:** 异步专用 S&H 采样使能位 ⁽¹⁾

1 = 专用 S&H 在不断采样，一经检测到触发脉冲即终止采样。

0 = 当检测到触发事件时专用 S&H 开始采样，并在两个 ADC 时钟周期内完成采样过程。

bit 3 **未实现:** 读为 0

bit 2-0 **ADCS<2:0>:** A/D 转换时钟分频值选择位 ⁽¹⁾

如果 SLOWCLK = 0，ADC 时钟分频值如下:

111 = FADC/7

110 = FADC/6.5

101 = FADC/6

100 = FADC/5.5

011 = FADC/5 (默认)

010 = FADC/4.5

001 = FADC/4

000 = FADC/3.5

如果 SLOWCLK = 1，ADC 时钟分频值如下:

111 = FADC/8

110 = FADC/7

101 = FADC/6

100 = FADC/5

011 = FADC/4 (默认)

010 = FADC/3

001 = FADC/2

000 = FADC/1

注 1: 仅可在禁止 ADC (ADON = 0) 时更改此控制位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 18-2: ADSTAT: A/D 状态寄存器

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

U-0	R/C-0, HS	R/C-0, HS	R/C-0, HS	R/C-0, HS	R/C-0, HS	R/C-0, HS	R/C-0, HS
—	P6RDY ⁽¹⁾	P5RDY ⁽²⁾	P4RDY ⁽²⁾	P3RDY ⁽³⁾	P2RDY ⁽⁴⁾	P1RDY	P0RDY
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

C = 可清零位

HS = 可硬件置 1 位

bit 15-7 **未实现:** 读为 0

bit 6 **P6RDY:** 输入对 6 转换数据就绪位 ⁽¹⁾

当缓冲区中数据就绪时此位被置 1, 当写入 0 时此位被清零。

bit 5 **P5RDY:** 输入对 5 转换数据就绪位 ⁽²⁾

当缓冲区中数据就绪时此位被置 1, 当写入 0 时此位被清零。

bit 4 **P4RDY:** 输入对 4 转换数据就绪位 ⁽²⁾

当缓冲区中数据就绪时此位被置 1, 当写入 0 时此位被清零。

bit 3 **P3RDY:** 输入对 3 转换数据就绪位 ⁽³⁾

当缓冲区中数据就绪时此位被置 1, 当写入 0 时此位被清零。

bit 2 **P2RDY:** 输入对 2 转换数据就绪位 ⁽⁴⁾

当缓冲区中数据就绪时此位被置 1, 当写入 0 时此位被清零。

bit 1 **P1RDY:** 输入对 1 转换数据就绪位

当缓冲区中数据就绪时此位被置 1, 当写入 0 时此位被清零。

bit 0 **P0RDY:** 输入对 0 转换数据就绪位

当缓冲区中数据就绪时此位被置 1, 当写入 0 时此位被清零。

注 1: 此位仅在 dsPIC33FJ16GS502、dsPIC33FJ16GS504 和 dsPIC33FJ06GS202 器件中可用。

2: 此位仅在 dsPIC33FJ16GS504 器件中可用。

3: 此位仅在 dsPIC33FJ16GS402/404、dsPIC33FJ16GS502、dsPIC33FJ16GS504 和 dsPIC33FJ06GS101 器件中可用。

4: 此位仅在 dsPIC33FJ16GS504 和 dsPIC33FJ16GS502 器件中可用。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 18-3: **ADBASE: A/D 基址寄存器** ^(1,2)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADBASE<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
ADBASE<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 **ADBASE<15:1>**: 此寄存器包含用户 ADC 中断服务程序跳转表的基地址。当读此寄存器时, 此寄存器包含 ADBASE 寄存器内容与 PxRDY 状态位编码值的和。

编码器逻辑提供最高优先级 PxRDY 位的位编号, 其中 P0RDY 为最高优先级, P6RDY 为最低优先级。

bit 0 **未实现**: 读为 0

- 注 1: 编码结果被左移两位, 因此结果的 bit 1-0 始终为零。
 2: 作为使用 ADBASE 寄存器调用 ADC 中断服务程序的替代方案, ADCP0-6 ADC 输入对转换完成中断也可用于调用各 ADC 输入对的 A/D 转换完成中断服务程序。

寄存器 18-4: **ADPCFG: A/D 端口配置寄存器**

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PCFG11 ⁽¹⁾	PCFG10 ⁽¹⁾	PCFG9 ⁽¹⁾	PCFG8 ⁽¹⁾
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 ⁽²⁾	PCFG6 ⁽³⁾	PCFG5 ⁽⁴⁾	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 15-12 **未实现**: 读为 0

bit 11-0 **PCFG<11:0>**: A/D 端口配置控制位 ^(1,2,3,4)

1 = 端口引脚处于数字模式, 使能端口读输入, A/D 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, A/D 采样引脚电压

- 注 1: 此位仅在 dsPIC33FJ16GS504 器件中可用。
 2: 此位仅在 dsPIC33FJ16GS402/404、dsPIC33FJ16GS502 和 dsPIC33FJ06GS101 器件中可用。
 3: 此位仅在 dsPIC33FJ06GS101、dsPIC33FJ16GS504、dsPIC33FJ16GS502 和 dsPIC33FJ16GS402/404 器件中可用。
 4: 此位仅在 dsPIC33FJ16GS504、dsPIC33FJ16GS502、dsPIC33FJ06GS102 和 dsPIC33FJ06GS202 器件中可用。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 18-5: ADCPC0: A/D 转换对控制寄存器 0

R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0	
IRQEN1		PEND1		SWTRG1		TRGSRC1<4:0>							
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
IRQEN0	PEND0	SWTRG0	TRGSRC0<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 IRQEN1: 中断请求允许 1 位

1 = 当通道 AN3 和 AN2 上请求的转换完成时允许产生中断请求
0 = 不产生中断请求

bit 14 PEND1: 等待转换状态 1 位

1 = 通道 AN3 和 AN2 在等待转换。该位在所选触发源有效时被置 1
0 = 转换已完成

bit 13 SWTRG1: 软件触发 1 位

1 = 启动对 AN3 和 AN2 的转换 (如果通过 TRGSRC 位选择了触发源) ⁽¹⁾

bit 12-8 TRGSRC1<4:0>: 触发 1 触发源选择位

为模拟通道 AN3 和 AN2 的转换选择触发源。

00000 = 不使能转换

00001 = 选择独立软件触发

00010 = 选择全局软件触发

00011 = 选择 PWM 特殊事件触发

00100 = 选择 PWM 发生器 1 主触发

00101 = 选择 PWM 发生器 2 主触发

00110 = 选择 PWM 发生器 3 主触发

00111 = 选择 PWM 发生器 4 主触发

01000 = 保留

.

.

01100 = Timer1 周期匹配

01101 = 保留

01110 = 选择 PWM 发生器 1 辅助触发

01111 = 选择 PWM 发生器 2 辅助触发

10000 = 选择 PWM 发生器 3 辅助触发

10001 = 选择 PWM 发生器 4 辅助触发

10010 = 保留

.

.

10110 = 保留

10111 = PWM 发生器 1 限流 ADC 触发

11000 = PWM 发生器 2 限流 ADC 触发

11001 = PWM 发生器 3 限流 ADC 触发

11010 = PWM 发生器 4 限流 ADC 触发

11011 = 保留

.

.

11111 = Timer2 周期匹配

注 1: 如果在进行其他转换, 当转换资源可用时将执行转换。当 PEND 位置 1 时该位将被复位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 18-5: ADCPC0: A/D 转换对控制寄存器 0 (续)

bit 7	IRQEN0: 中断请求允许 0 位 1 = 当通道 AN1 和 AN0 上请求的转换完成时允许产生中断请求 0 = 不产生中断请求
bit 6	PEND0: 等待转换状态 0 位 1 = 通道 AN1 和 AN0 在等待转换; 当所选触发源有效时置 1 0 = 转换已完成
bit 5	SWTRG0: 软件触发 0 位 1 = 启动对 AN1 和 AN0 的转换 (如果通过 TRGSRC 位选择了触发源) (1)
bit 4-0	TRGSRC0<4:0>: 触发 0 触发源选择位 为模拟通道 AN1 和 AN0 的转换选择触发源。 00000 = 不使能转换 00001 = 选择独立软件触发 00010 = 选择全局软件触发 00011 = 选择 PWM 特殊事件触发 00100 = 选择 PWM 发生器 1 主触发 00101 = 选择 PWM 发生器 2 主触发 00110 = 选择 PWM 发生器 3 主触发 00111 = 选择 PWM 发生器 4 主触发 01000 = 保留 . . . 01100 = Timer1 周期匹配 01101 = 保留 01110 = 选择 PWM 发生器 1 辅助触发 01111 = 选择 PWM 发生器 2 辅助触发 10000 = 选择 PWM 发生器 3 辅助触发 10001 = 选择 PWM 发生器 4 辅助触发 10010 = 保留 . . . 10110 = 保留 10111 = PWM 发生器 1 限流 ADC 触发 11000 = PWM 发生器 2 限流 ADC 触发 11001 = PWM 发生器 3 限流 ADC 触发 11010 = PWM 发生器 4 限流 ADC 触发 11011 = 保留 . . . 11111 = Timer2 周期匹配

注 1: 如果在进行其他转换, 当转换资源可用时将执行转换。当 PEND 位置 1 时该位将被复位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 18-6: ADCPC1: A/D 转换对控制寄存器 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
IRQEN3 ⁽¹⁾	PEND3 ⁽¹⁾	SWTRG3 ⁽¹⁾	TRGSRC3<4:0> ⁽¹⁾				
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
IRQEN2 ⁽²⁾	PEND2 ⁽²⁾	SWTRG2 ⁽²⁾	TRGSRC2<4:0> ⁽²⁾				
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **IRQEN3:** 中断请求允许 3 位 ⁽¹⁾
1 = 当通道 AN7 和 AN6 上请求的转换完成时允许产生中断请求
0 = 不产生中断请求
- bit 14 **PEND3:** 等待转换状态 3 位 ⁽¹⁾
1 = 通道 AN7 和 AN6 在等待转换。该位在所选触发源有效时被置 1
0 = 转换已完成
- bit 13 **SWTRG3:** 软件触发 3 位 ⁽¹⁾
1 = 启动对 AN7 和 AN6 的转换 (如果通过 TRGSRC 位选择了触发源) ⁽³⁾

- 注 1: 这些位仅在 dsPIC33FJ16GS402/404、dsPIC33FJ16GS504、dsPIC33FJ16GS502 和 dsPIC33FJ06GS101 器件中可用。
- 2: 这些位仅在 dsPIC33FJ16GS502、dsPIC33FJ16GS504、dsPIC33FJ06GS102、dsPIC33FJ06GS202 和 dsPIC33FJ16GS402/404 器件中可用。
- 3: 如果在进行其他转换, 当转换资源可用时将执行转换。当 PEND 位置 1 时该位将被复位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 18-6: ADCPC1: A/D 转换对控制寄存器 1 (续)

bit 12-8	TRGSRC3<4:0> : 触发 3 触发源选择位 ⁽¹⁾ 为模拟通道 AN7 和 AN6 的转换选择触发源。 00000 = 不使能转换 00001 = 选择独立软件触发 00010 = 选择全局软件触发 00011 = 选择 PWM 特殊事件触发 00100 = 选择 PWM 发生器 1 主触发 00101 = 选择 PWM 发生器 2 主触发 00110 = 选择 PWM 发生器 3 主触发 00111 = 选择 PWM 发生器 4 主触发 01000 = 保留 . . . 01100 = Timer1 周期匹配 01101 = 保留 01110 = 选择 PWM 发生器 1 辅助触发 01111 = 选择 PWM 发生器 2 辅助触发 10000 = 选择 PWM 发生器 3 辅助触发 10001 = 选择 PWM 发生器 4 辅助触发 10010 = 保留 . . . 10110 = 保留 10111 = PWM 发生器 1 限流 ADC 触发 11000 = PWM 发生器 2 限流 ADC 触发 11001 = PWM 发生器 3 限流 ADC 触发 11010 = PWM 发生器 4 限流 ADC 触发 11011 = 保留 . . . 11111 = Timer2 周期匹配
bit 7	IRQEN2 : 中断请求允许 2 位 ⁽²⁾ 1 = 当通道 AN5 和 AN4 上请求的转换完成时允许产生中断请求 (IRQ) 0 = 不产生中断请求 (IRQ)
bit 6	PEND2 : 等待转换状态 2 位 ⁽²⁾ 1 = 通道 AN5 和 AN4 在等待转换; 当所选触发源有效时置 1。 0 = 转换已完成
bit 5	SWTRG2 : 软件触发 2 位 ⁽²⁾ 1 = 启动对 AN5 和 AN4 的转换 (如果通过 TRGSRC 位选择了触发源) ⁽³⁾

- 注 1: 这些位仅在 dsPIC33FJ16GS402/404、dsPIC33FJ16GS504、dsPIC33FJ16GS502 和 dsPIC33FJ06GS101 器件中可用。
- 2: 这些位仅在 dsPIC33FJ16GS502、dsPIC33FJ16GS504、dsPIC33FJ06GS102、dsPIC33FJ06GS202 和 dsPIC33FJ16GS402/404 器件中可用。
- 3: 如果在进行其他转换, 当转换资源可用时将执行转换。当 PEND 位置 1 时该位将被复位。

寄存器 18-6: ADCPC1: A/D 转换对控制寄存器 1 (续)

bit 4-0

TRGSRC2<4:0>: 触发 2 触发源选择位
为模拟通道 AN5 和 AN4 的转换选择触发源。
00000 = 不使能转换
00001 = 选择独立软件触发
00010 = 选择全局软件触发
00011 = 选择 PWM 特殊事件触发
00100 = 选择 PWM 发生器 1 主触发
00101 = 选择 PWM 发生器 2 主触发
00110 = 选择 PWM 发生器 3 主触发
00111 = 选择 PWM 发生器 4 主触发
01000 = 保留
.
.
.
01100 = Timer1 周期匹配
01101 = 保留
01110 = 选择 PWM 发生器 1 辅助触发
01111 = 选择 PWM 发生器 2 辅助触发
10000 = 选择 PWM 发生器 3 辅助触发
10001 = 选择 PWM 发生器 4 辅助触发
10010 = 保留
.
.
.
10110 = 保留
10111 = PWM 发生器 1 限流 ADC 触发
11000 = PWM 发生器 2 限流 ADC 触发
11001 = PWM 发生器 3 限流 ADC 触发
11010 = PWM 发生器 4 限流 ADC 触发
11011 = 保留
.
.
.
11111 = Timer2 周期匹配

- 注 1: 这些位仅在 dsPIC33FJ16GS402/404、dsPIC33FJ16GS504、dsPIC33FJ16GS502 和 dsPIC33FJ06GS101 器件中可用。
- 2: 这些位仅在 dsPIC33FJ16GS502、dsPIC33FJ16GS504、dsPIC33FJ06GS102、dsPIC33FJ06GS202 和 dsPIC33FJ16GS402/404 器件中可用。
- 3: 如果在进行其他转换，当转换资源可用时将执行转换。当 PEND 位置 1 时该位将被复位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 18-7: **ADPC2: A/D 转换对控制寄存器 2⁽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
IRQEN5	PEND5	SWTRG5	TRGSRC5<4:0>				
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
IRQEN4	PEND4	SWTRG4	TRGSRC4<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 **IRQEN5:** 中断请求允许 5 位
1 = 当通道 AN11 和 AN10 上请求的转换完成时允许产生中断请求
0 = 不产生中断请求
- bit 14 **PEND5:** 等待转换状态 5 位
1 = 通道 AN11 和 AN10 在等待转换; 当所选触发源有效时置 1
0 = 转换已完成
- bit 13 **SWTRG5:** 软件触发 5 位
1 = 启动对 AN11 和 AN10 的转换 (如果通过 TRGSRC 位选择了触发源) ⁽²⁾
- bit 12-8 **TRGSRC5<4:0>:** 触发 5 触发源选择位
为模拟通道 AN11 和 AN10 的转换选择触发源。
00000 = 不使能转换
00001 = 选择独立软件触发
00010 = 选择全局软件触发
00011 = 选择 PWM 特殊事件触发
00100 = 选择 PWM 发生器 1 主触发
00101 = 选择 PWM 发生器 2 主触发
00110 = 选择 PWM 发生器 3 主触发
00111 = 选择 PWM 发生器 4 主触发
01000 = 保留
.
.
.
01100 = Timer1 周期匹配
01101 = 保留
01110 = 选择 PWM 发生器 1 辅助触发
01111 = 选择 PWM 发生器 2 辅助触发
10000 = 选择 PWM 发生器 3 辅助触发
10001 = 选择 PWM 发生器 4 辅助触发
10010 = 保留
.
.
.
10110 = 保留
10111 = PWM 发生器 1 限流 ADC 触发
11000 = PWM 发生器 2 限流 ADC 触发
11001 = PWM 发生器 3 限流 ADC 触发
11010 = PWM 发生器 4 限流 ADC 触发
11011 = 保留
.
.
.
11111 = Timer2 周期匹配

注 1: 仅 dsPIC33FJ16GS504 器件实现了此寄存器。
 2: 如果在进行其他转换, 当转换资源可用时将执行转换。当 PEND 位置 1 时该位将被复位。

寄存器 18-7: ADCPC2: A/D 转换对控制寄存器 2⁽¹⁾ (续)

bit 7	IRQEN4: 中断请求允许 4 位 1 = 当通道 AN9 和 AN8 上请求的转换完成时允许产生中断请求 0 = 不产生中断请求
bit 6	PEND4: 等待转换状态 4 位 1 = 通道 AN9 和 AN8 在等待转换; 当所选触发源有效时置 1 0 = 转换已完成
bit 5	SWTRG4: 软件触发 4 位 1 = 启动对 AN9 和 AN8 的转换 (如果通过 TRGSRC 位选择了触发源) ⁽²⁾
bit 4-0	TRGSRC4<4:0>: 触发 4 触发源选择位 为模拟通道 AN9 和 AN8 的转换选择触发源。 00000 = 不使能转换 00001 = 选择独立软件触发 00010 = 选择全局软件触发 00011 = 选择 PWM 特殊事件触发 00100 = 选择 PWM 发生器 1 主触发 00101 = 选择 PWM 发生器 2 主触发 00110 = 选择 PWM 发生器 3 主触发 00111 = 选择 PWM 发生器 4 主触发 01000 = 保留 . . . 01100 = Timer1 周期匹配 01101 = 保留 01110 = 选择 PWM 发生器 1 辅助触发 01111 = 选择 PWM 发生器 2 辅助触发 10000 = 选择 PWM 发生器 3 辅助触发 10001 = 选择 PWM 发生器 4 辅助触发 10010 = 保留 . . . 10110 = 保留 10111 = PWM 发生器 1 限流 ADC 触发 11000 = PWM 发生器 2 限流 ADC 触发 11001 = PWM 发生器 3 限流 ADC 触发 11010 = PWM 发生器 4 限流 ADC 触发 11011 = 保留 . . . 11111 = Timer2 周期匹配

注 1: 仅 dsPIC33FJ16GS504 器件实现了此寄存器。

2: 如果在进行其他转换, 当转换资源可用时将执行转换。当 PEND 位置 1 时该位将被复位。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

寄存器 18-8: ADCPC3: A/D 转换对控制寄存器 3⁽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
IRQEN6	PEND6	SWTRG6	TRGSRC6<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 15-8

未实现：读为 0
- bit 7

IRQEN6: 中断请求允许 6 位

1 = 当通道 AN13 和 AN12 上请求的转换完成时允许产生中断请求

0 = 不产生中断请求
- bit 6

PEND6: 等待转换状态 6 位

1 = 通道 AN13 和 AN12 在等待转换；当所选触发源有效时置 1

0 = 转换已完成
- bit 5

SWTRG6: 软件触发 6 位

1 = 如果通过 TRGSRC 位选择了触发源，启动对 AN13（INTREF）和 AN12（EXTREF）的转换

如果在进行其他转换，当转换资源可用时将执行转换。当 PEND 位置 1 时该位将被复位。

注 1: 仅 dsPIC33FJ16GS502 和 dsPIC33FJ16GS504 器件实现了此寄存器。

寄存器 18-8: ADCPC3: A/D 转换对控制寄存器 3⁽¹⁾ (续)

bit 4-0 **TRGSRC6<4:0>**: 触发 6 触发源选择位
为模拟通道 AN13 和 AN12 的转换选择触发源。

00000 = 不使能转换
00001 = 选择独立软件触发
00010 = 选择全局软件触发
00011 = 选择 PWM 特殊事件触发
00100 = 选择 PWM 发生器 1 主触发
00101 = 选择 PWM 发生器 2 主触发
00110 = 选择 PWM 发生器 3 主触发
00111 = 选择 PWM 发生器 4 主触发
01000 = 保留
.
.
.
01100 = Timer1 周期匹配
01101 = 保留
01110 = 选择 PWM 发生器 1 辅助触发
01111 = 选择 PWM 发生器 2 辅助触发
10000 = 选择 PWM 发生器 3 辅助触发
10001 = 选择 PWM 发生器 4 辅助触发
10010 = 保留
.
.
.
10110 = 保留
10111 = PWM 发生器 1 限流 ADC 触发
11000 = PWM 发生器 2 限流 ADC 触发
11001 = PWM 发生器 3 限流 ADC 触发
11010 = PWM 发生器 4 限流 ADC 触发
11011 = 保留
.
.
.
11111 = Timer2 周期匹配

注 1: 仅 dsPIC33FJ16GS502 和 dsPIC33FJ16GS504 器件实现了此寄存器。

注:

19.0 高速模拟比较器

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 45. High-Speed Analog Comparator” (DS70296)，该文档可从 Microchip 网站 (www.microchip.com) 下载。

dsPIC33F SMPS 比较器模块对瞬态电流和 / 或瞬态电压进行监测，由于瞬态电流和瞬态电压的速度太快，使得 CPU 和 ADC 可能无法捕捉到。

19.1 特性概述

SMPS 比较器模块具有以下主要特性：

- 16 个可选比较器输入
- 最多 4 个模拟比较器
- 与每个模拟比较器相关的 10 位 DAC
- 可编程的输出极性
- 中断产生功能
- 提供 DAC 输出的 DACOUT 引脚

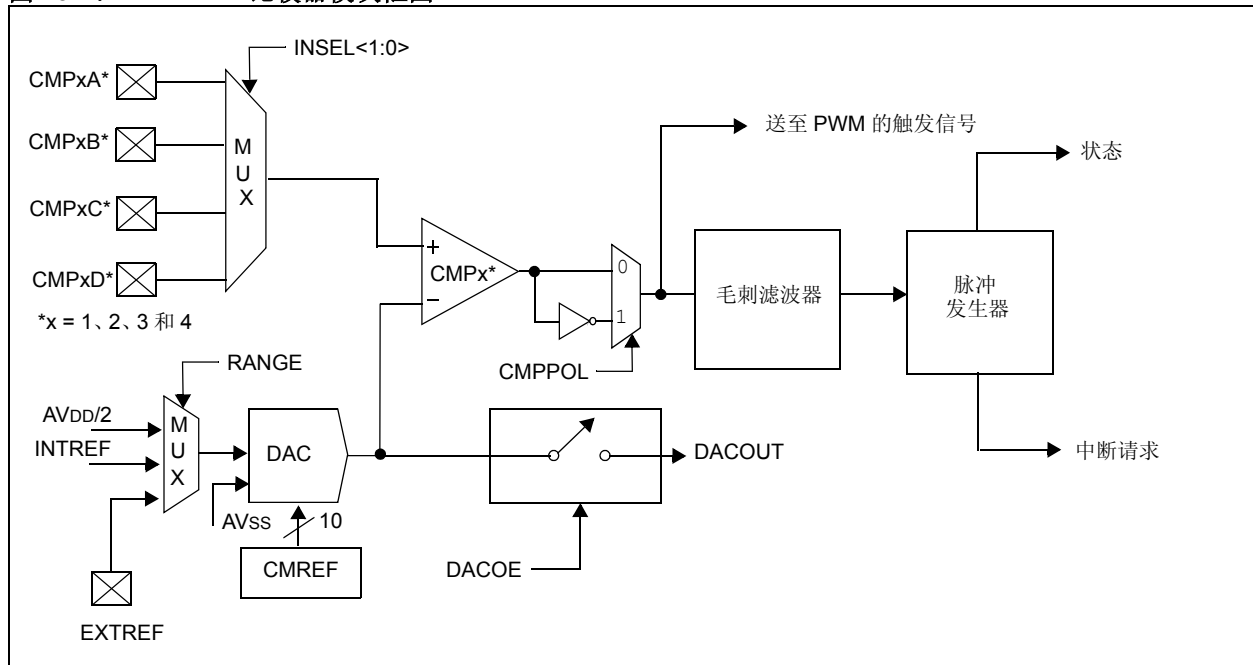
- DAC 具有三个工作范围：
 - $AVDD / 2$
 - 内部参考电压 $1.2V \pm 1\%$
 - 外部参考电压 $< (AVDD - 1.6V)$
- ADC 采样和转换触发功能
- 禁止功能降低功耗
- 对 PWM 模块的功能支持：
 - PWM 占空比控制
 - PWM 周期控制
 - PWM 故障检测

19.2 模块说明

图 19-1 所示为 SMPS 比较器模块中的一个模拟比较器的功能框图。模拟比较器提供高速操作，典型延时为 20 ns。比较器的典型失调电压为 $\pm 5\text{ mV}$ 。比较器的反相输入端总是连接到 DAC 电路。比较器的同相输入端连接到一个用于选择所需信号源引脚的模拟多路开关。

模拟比较器的输入引脚通常与模数转换器 (ADC) 模块使用的引脚共用。比较器和 ADC 可同时使用相同的引脚。此功能允许用户用 ADC 测量输入电压，同时用比较器检测瞬态电压。

图 19-1: 比较器模块框图



19.3 模块的应用

该模块为 SMPS dsPIC DSC 器件在电源转换应用中监测电压和电流提供了一种方法。检测瞬态条件并触发 dsPIC DSC 处理器和 / 或外设进行相应处理，而不需要处理器和 ADC 持续监测电压或电流，这种能力减轻了 dsPIC DSC 的负担，使之可以同时执行其他任务。

比较器模块具备一个高速比较器和一个相关的 10 位 DAC，此 DAC 提供可编程的参考电压连接到比较器的反相输入端。比较器输出的极性可由用户编程。模块的输出可用于以下模式：

- 产生中断
- 触发 ADC 采样和转换过程
- 截断 PWM 信号（限流）
- 截断 PWM 周期（最小电流）
- 禁止 PWM 输出（故障锁定）

比较器模块的输出可同时用于多种模式，如：(1) 产生中断；(2) 使 ADC 进行采样和转换；(3) 当检测到电压超出预期值时截断 PWM 输出。

当模拟输入电压超出编程设定的阈值电压时，还可将比较器模块用于将系统从休眠或空闲模式唤醒。

19.4 DAC

DAC 的范围通过模拟多路开关控制，模拟多路开关选择 $AV_{DD}/2$ 、内部 $1.2V \pm 1\%$ 参考电压或外部参考电压源 EXTREF。当所选输入信号源引脚与 ADC 共用时，通常使用 DAC 的整个范围 ($AV_{DD}/2$)。当使用电流检测电阻监测电流大小时很可能使用减小的范围选项 (INTREF)。通常，在这种应用中测量的电压比较小 ($<1.25V$)；因此，对比较器使用减小参考电压范围的选项扩展了这些应用中可获得的 DAC 分辨率。使用外部参考电压允许用户连接更适合其应用的参考电压。

图 19-1 中所示的 DACOUT 在一个给定的时刻仅能与一个比较器相关联。

注： 应该在软件中确保不会将多个 DACOE 位同时置 1。如果多个比较器使能了 DAC 输出，DACOUT 引脚的输出将不确定。

19.5 对 I/O 缓冲器的影响

如果使能了比较器模块且将一个引脚选择为比较器的信号源，那么所选 I/O 引脚必须禁止与其相关联的数字输入缓冲器，以防止因模拟输入电压使数字输入缓冲器中产生过大的电流。

19.6 数字逻辑

CMPCONx 寄存器（见寄存器 19-1）提供了对比较器模块进行配置的控制逻辑。数字逻辑为比较器输出提供了一个毛刺滤波器，可滤除小于两个指令周期的瞬态信号。在休眠或空闲模式下，毛刺滤波器被旁路，从而使能从比较器到中断控制器的异步路径。此异步路径可用于将处理器从休眠或空闲模式唤醒。

如果 CMPSIDL 位置 1，则在空闲模式下可禁止比较器。如果器件具有多个比较器，且任一 CMPSIDL 位置 1，那么在空闲模式下整个比较器组都将被禁止。此操作降低了此模块的时钟控制逻辑设计的复杂度。

数字逻辑还提供了一个 Tcy 宽度脉冲发生器，用于触发 ADC 以及产生中断请求。

CMPDACx 寄存器（见寄存器 19-2）为参考 DAC 提供数字输入值。

如果模块被禁止，DAC 和比较器将被禁止以降低功耗。

19.7 比较器输入范围

比较器的输入共模电压范围（Common Mode Range, CMR）有一个限制值，其典型值为 $(AV_{DD} - 1.5V)$ 。这意味着其两个输入都不能超过这个范围。只要一个输入在共模电压范围内，比较器输出就将是正确的。但是，任一输入超过 CMR 极限值都将导致比较器输入饱和。

如果两个输入都超过 CMR，比较器输出将不确定。

19.8 DAC 输出范围

DAC 的最大参考电压输入有一个限制值，为 $(AV_{DD} - 1.6)V$ 。外部参考电压输入不应超过这个值，否则 DAC 输出将变得不确定。

19.9 比较器寄存器

比较器模块由以下寄存器控制：

- CMPCONx：比较器控制寄存器
- CMPDACx：比较器 DAC 控制寄存器

寄存器 19-1: CMPCONx: 比较器控制寄存器

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
CMPON	—	CMPSIDL	—	—	—	—	DACOE
bit 15							bit 8
R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0
INSEL<1:0>		EXTREF	—	CMPSTAT	—	CMPPOL	RANGE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **CMPON:** 比较器工作模式位
1 = 使能比较器模块
0 = 禁止比较器模块 (降低功耗)
- bit 14 **未实现:** 读为 0
- bit 13 **CMPSIDL:** 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
如果器件具有多个比较器, 将任一 CMPSIDL 位置为 1, 在空闲模式下将禁止所有比较器。
- bit 12-9 **保留:** 读为 0
- bit 8 **DACOE:** DAC 输出使能位
1 = DAC 模拟电压输出到 DACOUT 引脚⁽¹⁾
0 = DAC 模拟电压未连接到 DACOUT 引脚
- bit 7-6 **INSEL<1:0>:** 比较器的输入信号源选择位
00 = 选择 CMPxA 输入引脚
01 = 选择 CMPxB 输入引脚
10 = 选择 CMPxC 输入引脚
11 = 选择 CMPxD 输入引脚
- bit 5 **EXTREF:** 使能外部参考电压源的位
1 = 外部参考电压源为 DAC 提供参考电压 (最大 DAC 电压由外部电压源决定)
0 = 内部参考电压源为 DAC 提供参考电压 (最大 DAC 电压由 RANGE 位的设置决定)
- bit 4 **保留:** 读为 0
- bit 3 **CMPSTAT:** 包含 CMPPOL 选择的比较器输出当前状态位
- bit 2 **保留:** 读为 0
- bit 1 **CMPPOL:** 比较器输出极性控制位
1 = 输出反相
0 = 输出不反相
- bit 0 **RANGE:** 选择 DAC 输出电压范围的位
1 = 高电压范围: 最大 DAC 值 = $AV_{DD} / 2$, AV_{DD} 为 3.3V 时为 1.65V
0 = 低电压范围: 最大 DAC 值 = INTREF, $1.2V \pm 1\%$

注 1: DACOUT 在任何给定时刻仅能与一个比较器相关联。必须通过软件设置相应的 DACOE 位来确保不会同时使能多个比较器的 DAC 输出。

寄存器 19-2: CMPDACx: 比较器 DAC 控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	CMREF<9:8>	
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMREF<7:0>							
bit 7							bit 0

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

bit 15-10 保留：读为 0

bit 9-0 **CMREF<9:0>**: 比较器参考电压选择位

1111111111 = (CMREF * INTREF/1024) 或 (CMREF * (AVDD/2)/1024)V （取决于 RANGE 位），或

(CMREF * EXTREF/1024) （当 EXTREF 置 1 时）

•

•

•

0000000000 = 0.0V

20.0 特殊功能

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 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*”。请访问 Microchip 网站（www.microchip.com）了解最新的“*dsPIC33F Family Reference Manual*”章节。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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
- JTAG 边界扫描接口
- 在线串行编程（ICSP™）
- 在线仿真
- 欠压复位（Brown-out Reset, BOR）

20.1 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 20-2 给出了 FBS、FGS、FOSCSEL、FOSC、FWDT、FPOR 和 FICD 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（0xF80000-0xFFFFF）。

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。在上电周期内对一个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 20-1 所示。

表 20-1: 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	—	—	—	—	BSS<2:0>			BWRP
0xF80002	保留	保留 ⁽¹⁾							
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	—	—	—	—	BOREN	FPWRT<2:0>		
0xF8000E	FICD	BKBUG	COE	JTAGEN	—	—	—	ICS<1:0>	
0xF80010	FUID0	用户部件 ID 字节 0							
0xF80012	FUID1	用户部件 ID 字节 1							

注 1： 当读这些位时，这些位读为 1。当写这些位时，将这些位设置为 1。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 20-2: 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位 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容量位 x11 = 无引导程序闪存段 引导空间为 256 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 0x0003FE 010 = 高安全性；引导程序闪存段结束于 0x0003FE 引导空间为 768 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 0x0007FE 001 = 高安全性；引导程序闪存段结束于 0x0007FE 引导空间为 1792 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 0x000FFE 000 = 高安全性；引导程序闪存段结束于 0x000FFE
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	外设引脚选择配置位 1 = 只允许一次重新配置 0 = 允许多次重新配置
OSCIOFNC	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

表 20-2: dsPIC33F 配置位的说明 (续)

位域	寄存器	说明
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 = 看门狗定时器处于非窗口模式下 0 = 看门狗定时器处于窗口模式下
WDTPRE	FWDT	看门狗定时器预分频比位 1 = 1:128 0 = 1:32
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
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
BOREN	FPOR	欠压复位使能位 1 = 使能 BOR 0 = 禁止 BOR
BKBUG	FICD	后台调试使能位 1 = 器件复位至用户模式 0 = 器件复位至调试模式
COE	FICD	调试器 / 仿真器使能位 1 = 器件复位至正常工作模式 0 = 器件复位至 Clip-On 仿真模式
JTAGEN	FICD	JTAG 使能位 1 = 使能 JTAG 0 = 禁止 JTAG
ICS<1:0>	FICD	ICD 通信通道选择使能位 11 = 通过 PGC1/EMUC1 和 PGD1/EMUD1 引脚进行通信 10 = 通过 PGC2/EMUC2 和 PGD2/EMUD2 引脚进行通信 01 = 通过 PGC3/EMUC3 和 PGD3/EMUD3 引脚进行通信 00 = 保留, 不要使用

20.2 片上稳压器

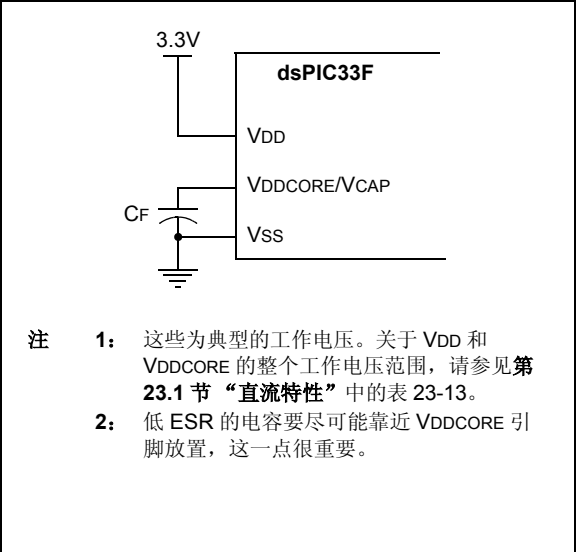
dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件使用标称值 2.5V 的电压为其内核数字逻辑供电。对于需要工作在更高典型电压值（如 3.3V）的设计中，这可能会产生冲突。为简化系统设计，dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 系列中的所有器件均包含一个片上稳压器，可使器件内核逻辑在 VDD 下工作。

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 20-1）。这可帮助维持稳压器的稳定性。滤波电容的推荐值在第 23.1 节“直流特性”的表 23-13 中提供。

注：低 ESR 的电容要尽可能靠近 VDDCORE 引脚放置，这一点很重要。

在 POR 时，片上稳压器需要约 20 μs 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 20-1：片上稳压器的连接 (1.2)



20.3 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>）为 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。BOR 电路在休眠或空闲模式下继续工作，当 VDD 下降到 BOR 阈值电压以下时将复位器件。

20.4 看门狗定时器 (WDT)

对于 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件, WDT 由 LPRC 振荡器驱动。当使能 WDT 时, 时钟源也将被使能。

20.4.1 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 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 (FWDTEN<3:0>) 控制, 该配置位允许选择 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
- 在完成时钟切换时, 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执行 CLRWDWT 指令时

注: 执行 CLRWDWT 和 PWRSAV 指令会将预分频器和后分频器的计数值清零。

20.4.2 休眠和空闲模式

如果使能了 WDT, 它将在休眠或空闲模式下继续运行。当发生 WDT 超时, 将唤醒器件并且将从执行 PWRSAV 指令处继续执行代码。唤醒器件后, 需要用软件将相应的 SLEEP 或 IDLE 位 (RCON<3:2>) 清零。

20.4.3 使能 WDT

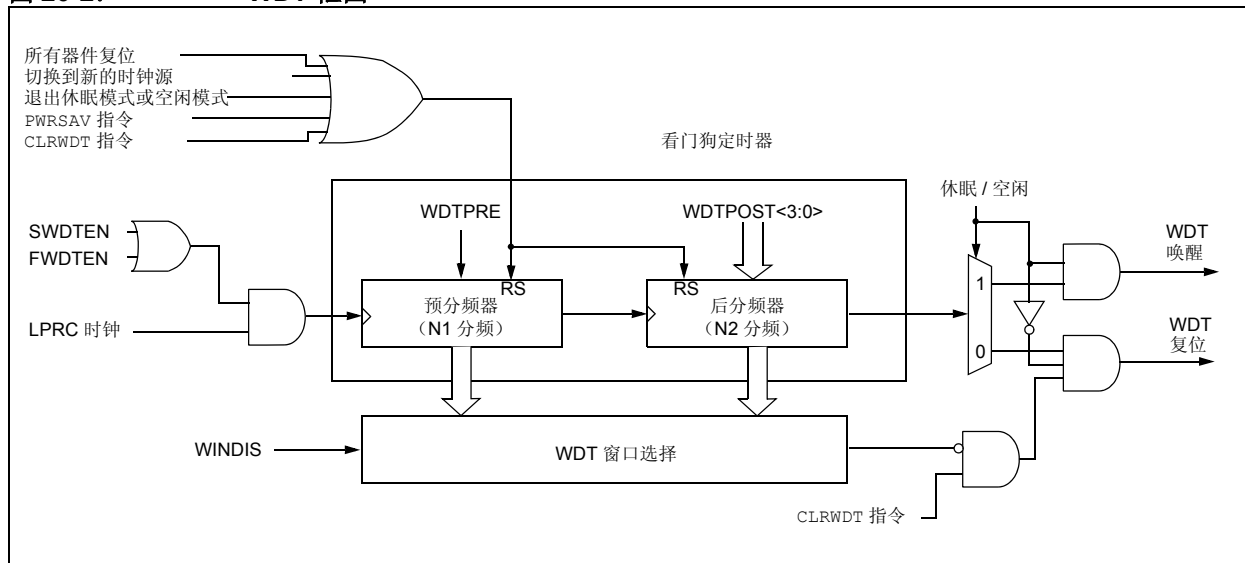
WDT 的使能或禁止由 FWDTEN 配置寄存器中的 FWDTEN 配置位控制。当 FWDTEN 配置位置 1 时, WDT 始终是使能的。

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 (FWDTEN<6>) 清零, 应用软件应仅在 WDT 周期的最后 1/4 周期中执行 CLRWDWT 指令。该 CLRWDWT 窗口可通过使用定时器确定。如果在该窗口之前执行 CLRWDWT 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 20-2: WDT 框图



20.5 JTAG 接口

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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。关于该接口的详细信息将会在文档的将来版本中提供。

20.6 在线串行编程

可在最终的应用电路中对 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 系列数字信号控制器进行串行编程。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

20.7 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、VDD、VSS、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。

20.8 代码保护和 CodeGuard™ 安全

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件提供了 CodeGuard™ 安全的中等级别实现。CodeGuard 安全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™ 安全来安全地更新闪存。

代码保护功能由如下配置寄存器控制：FBS 和 FGS。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 器件未实现安全段和 RAM 保护。

注： 请参考《CodeGuard™ 安全》参考手册（DS70180A_CN）获得有关 CodeGuard 安全的使用、配置和操作方面的更多信息。

表 20-3: 6 KB 器件的代码闪存安全段容量

配置位		
BSS<2:0> = x11 0K	VS = 256 IW	000000h 0001FEh 000200h 0003FEh 000400h 0007FEh 000800h 000FEh 001000h
	GS = 1792 IW	002BFEh
BSS<2:0> = x10 256	VS = 256 IW	000000h 0001FEh 000200h 0003FEh 000400h 0007FEh 000800h 000FEh 001000h
	BS = 256 IW	002BFEh
	GS = 1536 IW	
BSS<2:0> = x01 768	VS = 256 IW	000000h 0001FEh 000200h 0003FEh 000400h 0007FEh 000800h 000FEh 001000h
	BS = 768 IW	002BFEh
	GS = 1024 IW	
BSS<2:0> = x00 1792	VS = 256 IW	000000h 0001FEh 000200h 0003FEh 000400h 0007FEh 000800h 000FEh 001000h
	BS = 1792 IW	002BFEh

表 20-4: 16 KB 器件的代码闪存安全段容量

配置位		
BSS<2:0> = x11 0K	VS = 256 IW	000000h 0001FEh 000200h 0003FEh 000400h 0007FEh 000800h 000FEh 001000h
	GS = 5376 IW	002BFEh
BSS<2:0> = x10 256	VS = 256 IW	000000h 0001FEh 000200h 0003FEh 000400h 0007FEh 000800h 000FEh 001000h
	BS = 256 IW	002BFEh
	GS = 5120 IW	
BSS<2:0> = x01 768	VS = 256 IW	000000h 0001FEh 000200h 0003FEh 000400h 0007FEh 000800h 000FEh 001000h
	BS = 768 IW	002BFEh
	GS = 4608 IW	
BSS<2:0> = x00 1792	VS = 256 IW	000000h 0001FEh 000200h 0003FEh 000400h 0007FEh 000800h 000FEh 001000h
	BS = 1792 IW	002BFEh
	GS = 3584 IW	

注:

21.0 指令集汇总

注： 本数据手册总结了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 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*”。请访问 Microchip 网站 (www.microchip.com) 了解最新的“*dsPIC33F Family Reference Manual*”章节。

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 21-1 给出了在描述指令时使用的通用符号。

表 21-2 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
- 存放操作结果的目标寄存器，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，可以使用下列部分操作数：

- 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，使用下列部分操作数：

- 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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

大部分指令是单字指令。几条双字指令的设计，是为了要用 48 位来提供全部所需的信息。在第二个字中，8 个 MSb 全为 0。如果指令自身把第二个字当作一条指令来执行的话，它将作为一条 NOP 指令来执行。

双字指令执行需要两个指令周期。

大多数单字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 21-1: 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)

表 21-1: 操作码说明中使用的符号 (续)

字段	说明
Wm*Wm	用于平方指令的被乘数和乘数工作寄存器对 $\in \{W4 * W4, W5 * W5, W6 * W6, W7 * W7\}$
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 $\in \{[W8] + = 6, [W8] + = 4, [W8] + = 2, [W8], [W8] - = 6, [W8] - = 4, [W8] - = 2, [W9] + = 6, [W9] + = 4, [W9] + = 2, [W9], [W9] - = 6, [W9] - = 4, [W9] - = 2, [W9 + W12], \text{none}\}$
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 $\in \{[W10] + = 6, [W10] + = 4, [W10] + = 2, [W10], [W10] - = 6, [W10] - = 4, [W10] - = 2, [W11] + = 6, [W11] + = 4, [W11] + = 2, [W11], [W11] - = 6, [W11] - = 4, [W11] - = 2, [W11 + W12], \text{none}\}$
Wyd	用于 DSP 指令的 Y 数据空间预取目标寄存器 $\in \{W4..W7\}$

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 21-2: 指令集概述

基本指令 #	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
		ADD Wso, #Slit4, Acc	将 16 位有符号立即数加到累加器	1	1	OA,OB,SA,SB
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	1	N,Z
		AND f, WREG	$WREG = f .AND. WREG$	1	1	N,Z
		AND #lit10, Wn	$Wd = lit10 .AND. Wd$	1	1	N,Z
		AND Wb, Ws, Wd	$Wd = Wb .AND. Ws$	1	1	N,Z
		AND Wb, #lit5, Wd	$Wd = Wb .AND. lit5$	1	1	N,Z
4	ASR	ASR f	$f =$ 算术右移 f	1	1	C,N,OV,Z
		ASR f, WREG	$WREG =$ 算术右移 f	1	1	C,N,OV,Z
		ASR Ws, Wd	$Wd =$ 算术右移 Ws	1	1	C,N,OV,Z
		ASR Wb, Wns, Wnd	$Wnd =$ 将 Wb 算术右移 Wns 位	1	1	N,Z
		ASR Wb, #lit5, Wnd	$Wnd =$ 将 Wb 算术右移 lit5 位	1	1	N,Z
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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 21-2: 指令集概述 (续)

基本指令 #	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. Ws, #bit4 C	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. Ws, #bit4 Z	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	CLRWDI	CLRWDI	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 = $\bar{Ws}$	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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 21-2: 指令集概述 (续)

基本指令 #	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
47	MOVSAC	MOVSAC Ws, Wnd	将 Ws 中的双字内容送入 W(nd + 1):W(nd)	1	2	无
		MOVSAC Acc, Wx, Wxd, Wy, Wyd, AWB	预取操作数并保存累加器	1	1	无

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 21-2: 指令集概述 (续)

基本指令 #	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	$\{Wnd + 1, Wnd\} = \text{有符号}(Wb) * \text{有符号}(Ws)$	1	1	无
		MUL.SU Wb, Ws, Wnd	$\{Wnd + 1, Wnd\} = \text{有符号}(Wb) * \text{无符号}(Ws)$	1	1	无
		MUL.US Wb, Ws, Wnd	$\{Wnd + 1, Wnd\} = \text{无符号}(Wb) * \text{有符号}(Ws)$	1	1	无
		MUL.UU Wb, Ws, Wnd	$\{Wnd + 1, Wnd\} = \text{无符号}(Wb) * \text{无符号}(Ws)$	1	1	无
		MUL.SU $Wb, \#lit5, Wnd$	$\{Wnd + 1, Wnd\} = \text{有符号}(Wb) * \text{无符号}(lit5)$	1	1	无
		MUL.UU $Wb, \#lit5, Wnd$	$\{Wnd + 1, Wnd\} = \text{无符号}(Wb) * \text{无符号}(lit5)$	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	$Wd = \overline{Ws} + 1$	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	PWRSABV	PWRSABV $\#lit1$	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

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 21-2: 指令集概述 (续)

基本指令 #	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>	<i>f</i> = 循环右移 <i>f</i> (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>	<i>Wd</i> = 循环右移 <i>Ws</i> (不带进位)	1	1	N,Z
67	SAC	SAC <i>Acc</i> , # <i>Slit4</i> , <i>Wdo</i>	保存累加器内容	1	1	无
		SAC.R <i>Acc</i> , # <i>Slit4</i> , <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>	<i>Wnd</i> = 符号扩展后的 <i>Ws</i>	1	1	C,N,Z
69	SETM	SETM <i>f</i>	<i>f</i> = 0xFFFF	1	1	无
		SETM WREG	WREG = 0xFFFF	1	1	无
		SETM <i>Ws</i>	<i>Ws</i> = 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> , # <i>Slit6</i>	对累加器算术移位 <i>Slit6</i> 次	1	1	OA,OB,OAB,SA,SB,SAB
71	SL	SL <i>f</i>	<i>f</i> = 左移 <i>f</i>	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>	<i>Wd</i> = 左移 <i>Ws</i>	1	1	C,N,OV,Z
		SL <i>Wb</i> , <i>Wns</i> , <i>Wnd</i>	<i>Wnd</i> = 将 <i>Wb</i> 左移 <i>Wns</i> 位	1	1	N,Z
		SL <i>Wb</i> , # <i>lit5</i> , <i>Wnd</i>	<i>Wnd</i> = 将 <i>Wb</i> 左移 <i>lit5</i> 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>	<i>f</i> = <i>f</i> - WREG	1	1	C,DC,N,OV,Z
		SUB <i>f</i> , WREG	WREG = <i>f</i> - WREG	1	1	C,DC,N,OV,Z
		SUB # <i>lit10</i> , <i>Wn</i>	<i>Wn</i> = <i>Wn</i> - <i>lit10</i>	1	1	C,DC,N,OV,Z
		SUB <i>Wb</i> , <i>Ws</i> , <i>Wd</i>	<i>Wd</i> = <i>Wb</i> - <i>Ws</i>	1	1	C,DC,N,OV,Z
		SUB <i>Wb</i> , # <i>lit5</i> , <i>Wd</i>	<i>Wd</i> = <i>Wb</i> - <i>lit5</i>	1	1	C,DC,N,OV,Z
73	SUBB	SUBB <i>f</i>	<i>f</i> = <i>f</i> - WREG - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB <i>f</i> , WREG	WREG = <i>f</i> - WREG - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB # <i>lit10</i> , <i>Wn</i>	<i>Wn</i> = <i>Wn</i> - <i>lit10</i> - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB <i>Wb</i> , <i>Ws</i> , <i>Wd</i>	<i>Wd</i> = <i>Wb</i> - <i>Ws</i> - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB <i>Wb</i> , # <i>lit5</i> , <i>Wd</i>	<i>Wd</i> = <i>Wb</i> - <i>lit5</i> - ($\overline{C}$)	1	1	C,DC,N,OV,Z
74	SUBR	SUBR <i>f</i>	<i>f</i> = WREG - <i>f</i>	1	1	C,DC,N,OV,Z
		SUBR <i>f</i> , WREG	WREG = WREG - <i>f</i>	1	1	C,DC,N,OV,Z
		SUBR <i>Wb</i> , <i>Ws</i> , <i>Wd</i>	<i>Wd</i> = <i>Ws</i> - <i>Wb</i>	1	1	C,DC,N,OV,Z
		SUBR <i>Wb</i> , # <i>lit5</i> , <i>Wd</i>	<i>Wd</i> = <i>lit5</i> - <i>Wb</i>	1	1	C,DC,N,OV,Z
75	SUBBR	SUBBR <i>f</i>	<i>f</i> = WREG - <i>f</i> - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR <i>f</i> , WREG	WREG = WREG - <i>f</i> - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR <i>Wb</i> , <i>Ws</i> , <i>Wd</i>	<i>Wd</i> = <i>Ws</i> - <i>Wb</i> - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR <i>Wb</i> , # <i>lit5</i> , <i>Wd</i>	<i>Wd</i> = <i>lit5</i> - <i>Wb</i> - ($\overline{C}$)	1	1	C,DC,N,OV,Z
76	SWAP	SWAP.b <i>Wn</i>	<i>Wn</i> = 半字节交换 <i>Wn</i> 内容	1	1	无
		SWAP <i>Wn</i>	<i>Wn</i> = 将 <i>Wn</i> 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 <i>Wd</i> <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 <i>Wd</i>	1	2	无
79	TBLWTH	TBLWTH <i>Ws</i> , <i>Wd</i>	将 <i>Ws</i> <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 <i>Ws</i> 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>	<i>f</i> = <i>f</i> .XOR.WREG	1	1	N,Z
		XOR <i>f</i> , WREG	WREG = <i>f</i> .XOR.WREG	1	1	N,Z
		XOR # <i>lit10</i> , <i>Wn</i>	<i>Wd</i> = <i>lit10</i> .XOR. <i>Wd</i>	1	1	N,Z
		XOR <i>Wb</i> , <i>Ws</i> , <i>Wd</i>	<i>Wd</i> = <i>Wb</i> .XOR. <i>Ws</i>	1	1	N,Z
		XOR <i>Wb</i> , # <i>lit5</i> , <i>Wd</i>	<i>Wd</i> = <i>Wb</i> .XOR. <i>lit5</i>	1	1	N,Z
83	ZE	ZE <i>Ws</i> , <i>Wnd</i>	<i>Wnd</i> = 零扩展后的 <i>Ws</i>	1	1	C,Z,N

22.0 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

22.1 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。

22.2 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

22.3 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。

22.4 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来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

22.5 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

22.6 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。

22.7 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。

22.8 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。

22.9 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。

22.10 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。

22.11 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。

22.12 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。

22.13 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、 $\Sigma\text{-}\Delta$ 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)。

23.0 电气特性

本章将对 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/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⁽¹⁾

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 ⁽²⁾	250 mA
任一 I/O 引脚的最大输出灌电流 ⁽³⁾	4 mA
任一 I/O 引脚的最大输出拉电流 ⁽³⁾	4 mA
所有端口的最大灌电流.....	200 mA
所有端口的最大拉电流 ⁽²⁾	200 mA
非可重映射 PWM 引脚的最大输出灌电流.....	16 mA
非可重映射 PWM 引脚的最大输出拉电流.....	16mA

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 23-2）。

3: 例外情况是 PWMxL 和 PWMxH 引脚，其最大拉/灌电流为 16 mA；以及数字引脚，其最大拉/灌电流为 8 mA。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

23.1 直流特性

表 23-1: 工作 MIPS 与电压

特性	VDD 范围 (单位: V)	温度范围 (单位: °C)	最大 MIPS
			dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04
	3.0-3.6V	-40°C 至 +85°C	40
	3.0-3.6V	-40°C 至 +125°C	40

表 23-2: 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: $P_{INT} = V_{DD} \times (I_{DD} - \Sigma I_{OH})$ I/O 引脚功耗: $I/O = \Sigma (\{V_{DD} - V_{OH}\} \times I_{OH}) + \Sigma (V_{OL} \times I_{OL})$	PD	PINT + PI/O			W
最大允许功耗	PDMAX	$(T_J - T_A)/\theta_{JA}$			W

表 23-3: 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	θ_{JA}	31	—	°C/W	1
封装热阻, 44 引脚 TQFP	θ_{JA}	43	—	°C/W	1
封装热阻, 28 引脚 SPDIP	θ_{JA}	45	—	°C/W	1
封装热阻, 28 引脚 SOIC	θ_{JA}	50	—	°C/W	1
封装热阻, 28 引脚 QFN-S	θ_{JA}	33	—	°C/W	1
封装热阻, 18 引脚 SOIC	θ_{JA}	76	—	°C/W	1

注 1: 通过封装模拟获得结点与环境的热阻值 (θ_{JA})。

表 23-4: 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 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (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 ⁽¹⁾	最大值	单位	条件
工作电压							
	供电电压						
DC10	VDD		3.0	—	3.6	V	工业级和扩展级
DC12	VDR	RAM 数据保持电压 ⁽²⁾	1.1	—	1.8	V	
DC16	VPOR	确保内部上电复位信号的 VDD 启动电压 ⁽³⁾	—	—	VSS	V	
DC17	SVDD	确保内部上电复位信号的 VDD 上升率 ⁽³⁾	0.03	—	—	V/ms	0-3.0V/0.1s
DC18	VCORE	VDD 内核内部稳压器电压	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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 23-5: 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	55	—	mA	-40°C	3.3V	10 MIPS 见注 2	
DC20a	55	—	mA	+25°C			
DC20b	55	—	mA	+85°C			
DC20c	55	—	mA	+125°C			
DC21d	68	—	mA	-40°C	3.3V	16 MIPS 见注 2 和注 3	
DC21a	68	—	mA	+25°C			
DC21b	68	—	mA	+85°C			
DC21c	68	—	mA	+125°C			
DC22d	78	—	mA	-40°C	3.3V	20 MIPS 见注 2 和注 3	
DC22a	78	—	mA	+25°C			
DC22b	78	—	mA	+85°C			
DC22c	78	—	mA	+125°C			
DC23d	88	—	mA	-40°C	3.3V	30 MIPS 见注 2 和注 3	
DC23a	88	—	mA	+25°C			
DC23b	88	—	mA	+85°C			
DC23c	88	—	mA	+125°C			
DC24d	98	—	mA	-40°C	3.3V	40 MIPS 见注 2	
DC24a	98	—	mA	+25°C			
DC24b	98	—	mA	+85°C			
DC24c	98	—	mA	+125°C			
DC25d	128	—	mA	-40°C	3.3V	40 MIPS 见注 2, 除了 PWM 工作在最大速度 (PTCON2 = 0x0000)	
DC25a	125	—	mA	+25°C			
DC25b	121	—	mA	+85°C			
DC25c	119	—	mA	+125°C			
DC26d	115	—	mA	-40°C	3.3V	40 MIPS 见注 2, 除了 PWM 工作在 1/2 速度 (PTCON2 = 0x0001)	
DC26a	112	—	mA	+25°C			
DC26b	110	—	mA	+85°C			
DC26c	108	—	mA	+125°C			
DC27d	111	—	mA	-40°C	3.3V	40 MIPS 见注 2, 除了 PWM 工作在 1/4 速度 (PTCON2 = 0x0002)	
DC27a	108	—	mA	+25°C			
DC27b	105	—	mA	+85°C			
DC27c	103	—	mA	+125°C			
DC28d	102	—	mA	-40°C	3.3V	40 MIPS 见注 2, 除了 PWM 工作在 1/8 速度 (PTCON2 = 0x0003)	
DC28a	100	—	mA	+25°C			
DC28b	100	—	mA	+85°C			
DC28c	100	—	mA	+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。
- 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 (PMD 的所有位全部置 1)。
- 3: 这些参数为特性值, 但生产时未经测试。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 23-6: 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 ⁽¹⁾	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 ⁽²⁾					
DC40d	80	—	mA	-40°C	3.3V 10 MIPS
DC40a	80	—	mA	+25°C	
DC40b	80	—	mA	+85°C	
DC40c	80	—	mA	+125°C	
DC41d	81	—	mA	-40°C	3.3V 16 MIPS ⁽³⁾
DC41a	81	—	mA	+25°C	
DC41b	81	—	mA	+85°C	
DC41c	81	—	mA	+125°C	
DC42d	82	—	mA	-40°C	3.3V 20 MIPS ⁽³⁾
DC42a	82	—	mA	+25°C	
DC42b	82	—	mA	+85°C	
DC42c	82	—	mA	+125°C	
DC43d	84	—	mA	-40°C	3.3V 30 MIPS ⁽³⁾
DC43a	84	—	mA	+25°C	
DC43b	84	—	mA	+85°C	
DC43c	84	—	mA	+125°C	
DC44d	86	—	mA	-40°C	3.3V 40 MIPS
DC44a	86	—	mA	+25°C	
DC44b	86	—	mA	+85°C	
DC44c	86	—	mA	+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。
- 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。
- 3: 这些参数为特性值, 但生产时未经测试。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 23-7: 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 -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	
掉电电流 (IPD) ^(2,4)					
DC60d	304	—	μA	-40°C	3.3V 基本掉电电流
DC60a	317	—	μA	+25°C	
DC60b	321	—	μA	+85°C	
DC60c	391	—	μA	+125°C	
DC61d	40	—	μA	-40°C	3.3V 看门狗定时器电流: ΔI _{WDT} ⁽³⁾
DC61a	40	—	μA	+25°C	
DC61b	40	—	μA	+85°C	
DC61c	80	—	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。
- 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, 并且 VREGS (RCON<8>) = 1。
- 3: Δ 电流为 WDT 模块使能时额外消耗的电流。此电流应被加到基本 IPD 电流。
- 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 23-8: 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-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	打盹模式 时钟分频 比	单位	条件
DC73a	86	—	1:2	mA	-40°C 3.3V 40 MIPS
DC73f	86	—	1:64	mA	
DC73g	86	—	1:128	mA	
DC70a	86	—	1:2	mA	+25°C 3.3V 40 MIPS
DC70f	86	—	1:64	mA	
DC70g	86	—	1:128	mA	
DC71a	86	—	1:2	mA	+85°C 3.3V 40 MIPS
DC71f	86	—	1:64	mA	
DC71g	86	—	1:128	mA	
DC72a	86	—	1:2	mA	+125°C 3.3V 40 MIPS
DC72f	86	—	1:64	mA	
DC72g	86	—	1:128	mA	

- 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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 23-9: 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
DI10	V _{IL}	输入低电压					
DI15		I/O 引脚	V _{SS}	—	0.2 V _{DD}	V	
DI16		MCLR	V _{SS}	—	0.2 V _{DD}	V	
DI17		OSC1 (XT 模式)	V _{SS}	—	0.2 V _{DD}	V	
DI18		OSC1 (HS 模式)	V _{SS}	—	0.2 V _{DD}	V	
DI19		SDAx 和 SCLx	V _{SS}	—	0.3 V _{DD}	V	禁止 SMBus
		SDAx 和 SCLx	V _{SS}	—	0.2 V _{DD}	V	使能 SMBus
DI20	V _{IH}	输入高电压					
		I/O 引脚:					
		带模拟功能 ⁽⁴⁾	0.8 V _{DD}	—	V _{DD}	V	
		仅数字功能 ⁽⁴⁾	0.8 V _{DD}	—	5.5	V	
DI25		MCLR	0.8 V _{DD}	—	V _{DD}	V	
DI26		OSC1 (XT 模式)	0.7 V _{DD}	—	V _{DD}	V	
DI27		OSC1 (HS 模式)	0.7 V _{DD}	—	V _{DD}	V	
DI28		SDAx 和 SCLx	0.7 V _{DD}	—	V _{DD}	V	禁止 SMBus
DI29		SDAx 和 SCLx	0.8 V _{DD}	—	V _{DD}	V	使能 SMBus
DI30	ICNPU	CNx 上拉电流	—	250	—	μA	V _{DD} = 3.3V, V _{PIN} = V _{SS}
DI50	I _{IL}	输入漏电流 ^(2,3,4)					
		I/O 端口:					
		4 mA 拉 / 灌电流能力	—	±2	—	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态
		8 mA 拉 / 灌电流能力	—	±4	—	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态
		16 mA 拉 / 灌电流能力	—	±8	—	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态
DI51		模拟输入引脚	—	—	±1	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态, -40°C ≤ TA ≤ +85°C
DI51b		模拟输入引脚	—	—	±3.5	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 引脚处于高阻态, -40°C ≤ TA ≤ +125°C
DI55		MCLR	—	—	±2	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD}
DI56		OSC1	—	—	±2	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , 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 引脚上的漏电流主要取决于所施加电压。指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漏电流。
- 3: 负电流定义为引脚的拉电流。
- 4: 请参见表 9-1 了解仅数字引脚和模拟引脚的列表以及拉 / 灌电流能力。

表 23-10: 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	VOL	输出低电压					
		I/O 端口:					
		4 mA 拉 / 灌电流能力	—	0.4	—	V	IOL = 4 mA, VDD = 3.3V
		8 mA 拉 / 灌电流能力	—	0.4	—	V	IOL = 8 mA, VDD = 3.3V
	DO16	16 mA 拉 / 灌电流能力	—	0.4	—	V	IOL = 16 mA, VDD = 3.3V
		OSC2/CLKO	—	0.4	—	V	IOL = 2 mA, VDD = 3.3V
DO20	VOH	输出高电压					
		I/O 端口:					
		4 mA 拉 / 灌电流能力	—	2.40	—	V	IOH = -4 mA, VDD = 3.3V
		8 mA 拉 / 灌电流能力	—	2.40	—	V	IOH = -8 mA, VDD = 3.3V
	DO26	16 mA 拉 / 灌电流能力	—	2.40	—	V	IOH = -16 mA, VDD = 3.3V
		OSC2/CLKO	—	2.41	—	V	IOH = -1.3 mA, VDD = 3.3V

表 23-11: 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 (1)	典型值	最大值	单位	条件
BO10	VBOR	当 VDD 由高变低时的 BOR 事件 BOR 事件与 VDD 内核电压下降关联	2.55	—	2.79	V	

注 1: 这些参数仅供设计参考, 生产时未经测试。

表 23-12: 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 ⁽¹⁾	最大值	单位	条件
D130	EP	闪存程序存储器					
D131	VPR	单元耐擦写能力	10,000	—	—	E/W	-40°C 至 +125°C
D132B	VPEW	读操作时的 VDD	V _{MIN}	—	3.6	V	V _{MIN} = 最小工作电压
D134	TRETD	自定时写的 VDD	V _{MIN}	—	3.6	V	V _{MIN} = 最小工作电压
D135	IDDP	特性保持时间	20	—	—	年	假设没有违反其他规范, -40°C 至 +125°C
D136	TRW	编程时的供电电流	—	10	—	mA	
D137	TPE	行写入时间	1.3	1.5	1.8	ms	
D138	TWW	页擦除时间	20	23	27	ms	
		字写周期时间	42	48	56	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 23-13: 内部稳压器规范

工作温度: -40°C ≤ Ta ≤ +85°C (工业级) -40°C ≤ Ta ≤ +125°C (扩展级)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
	CEFC	外部滤波电容值	1	10	—	μF	电容必须具有较低的等效串联阻抗 (< 5 欧姆)

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

23.2 交流特性和时序参数

本节给出了 dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 系列器件的交流特性和时序参数。

表 23-14: 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°C ≤ TA ≤ +85°C (工业级)
		-40°C ≤ TA ≤ +125°C (扩展级)
工作电压 VDD 范围如第 23.0 节 “电气特性” 中所介绍。		

图 23-1: 器件时序规范的负载条件

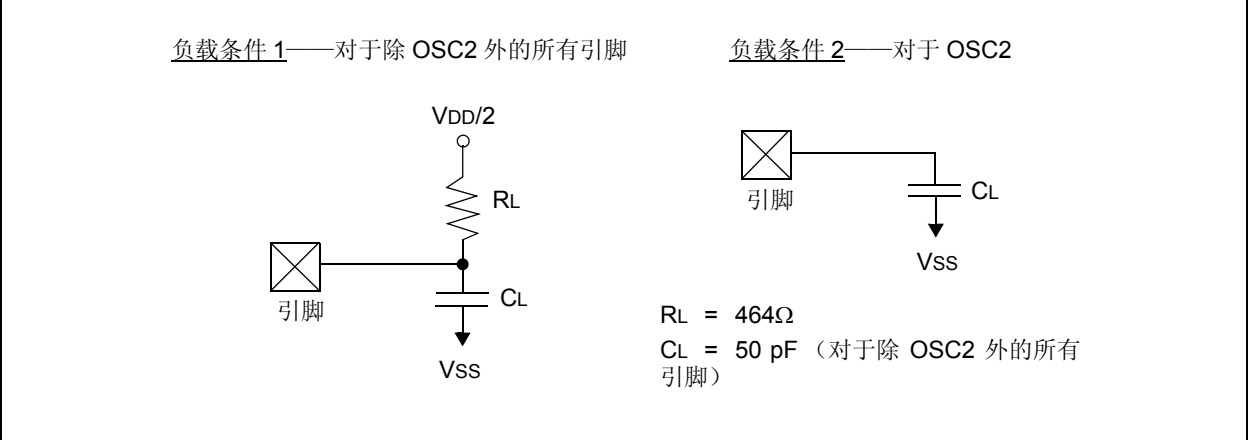


表 23-15: 输出引脚上的容性负载要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件
DO50	Cosc2	OSC2 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 ² C™ 模式下

图 23-2: 外部时钟时序

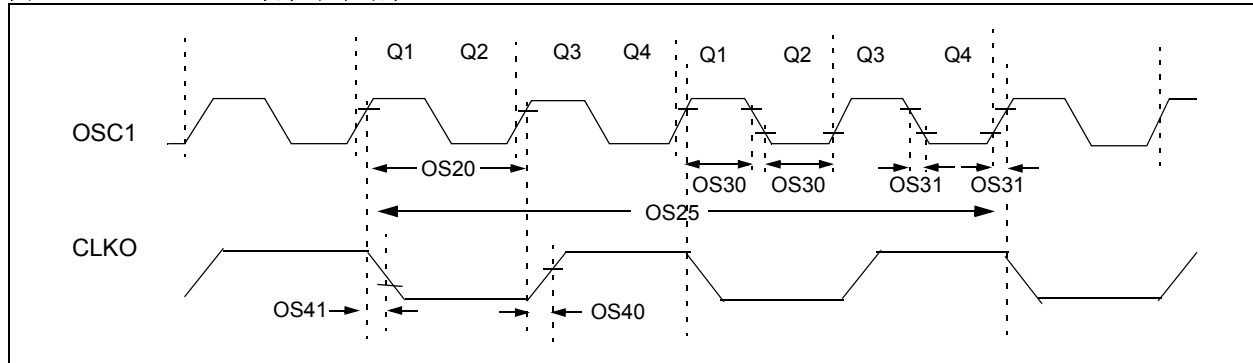


表 23-16: 外部时钟时序要求

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 ⁽¹⁾	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	MHz MHz	XT HS
OS20	TOSC	TOSC = 1/FOSC	12.5	—	DC	ns	
OS25	TCY	指令周期 ⁽²⁾	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 TOSC	—	0.625 x TOSC	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 ⁽³⁾	—	5.2	—	ns	
OS41	TckF	CLKO 下降时间 ⁽³⁾	—	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 CLK0 信号。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 23-17: 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 (1)	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, 生产时未经测试。

表 23-18: 附属 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 (1)	最大值	单位	条件
	FHPOUT	片上 16 倍频 PLL CCO 频率	105	120	135	MHz	
	FHPIN	片上 16 倍频 PLL 相位检测器输入频率	6.56	7.5	8.44	MHz	
	TSU	频率发生器锁定时间	—	—	10	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。这些参数仅供设计参考, 生产时未经测试。

表 23-19: 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） -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	
	FRC 频率 = 7.37 MHz ^(1,2) 时的内部 FRC 精度						
F20	FRC	—	±2	—	%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
	FRC	—	±5	—	%	-40°C ≤ TA ≤ +125°C	VDD = 3.0-3.6V

注 1: 频率在 +25°C、3.3V 条件下校准。TUN 位可用于补偿温度漂移。

2: FRC 被设置为 +25°C 下的初始频率 7.37 MHz (±2%)。

表 23-20: 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	
	LPRC 频率 = 32.768 kHz ⁽¹⁾						
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 23-3: I/O 时序特性

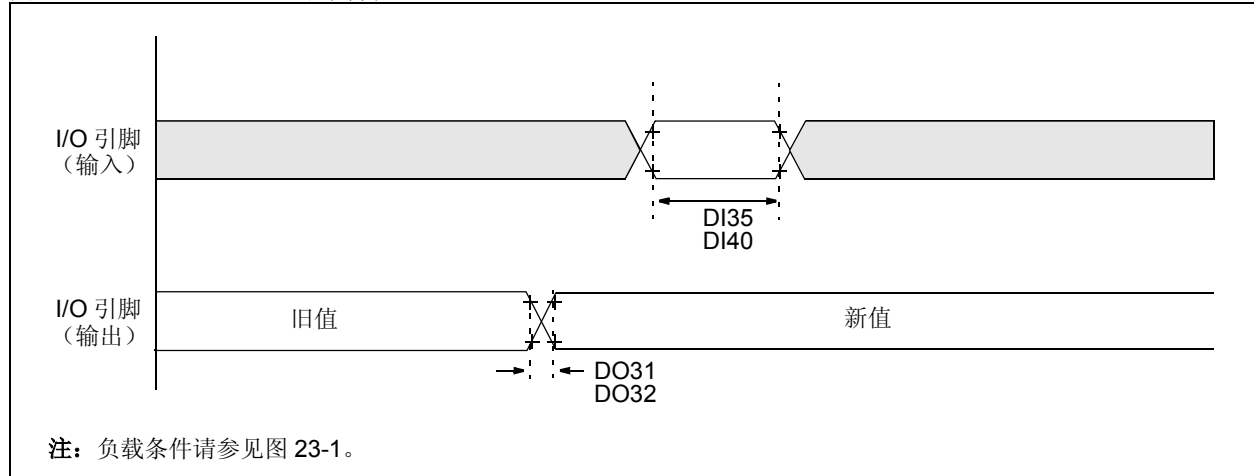


表 23-21: I/O 时序要求

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 ⁽¹⁾	最大值	单位	条件
DO31	TioR	端口输出上升时间	—	10	25	ns	测试条件请参见图 23-1
DO32	TioF	端口输出下降时间	—	10	25	ns	测试条件请参见图 23-1
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 23-4: 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

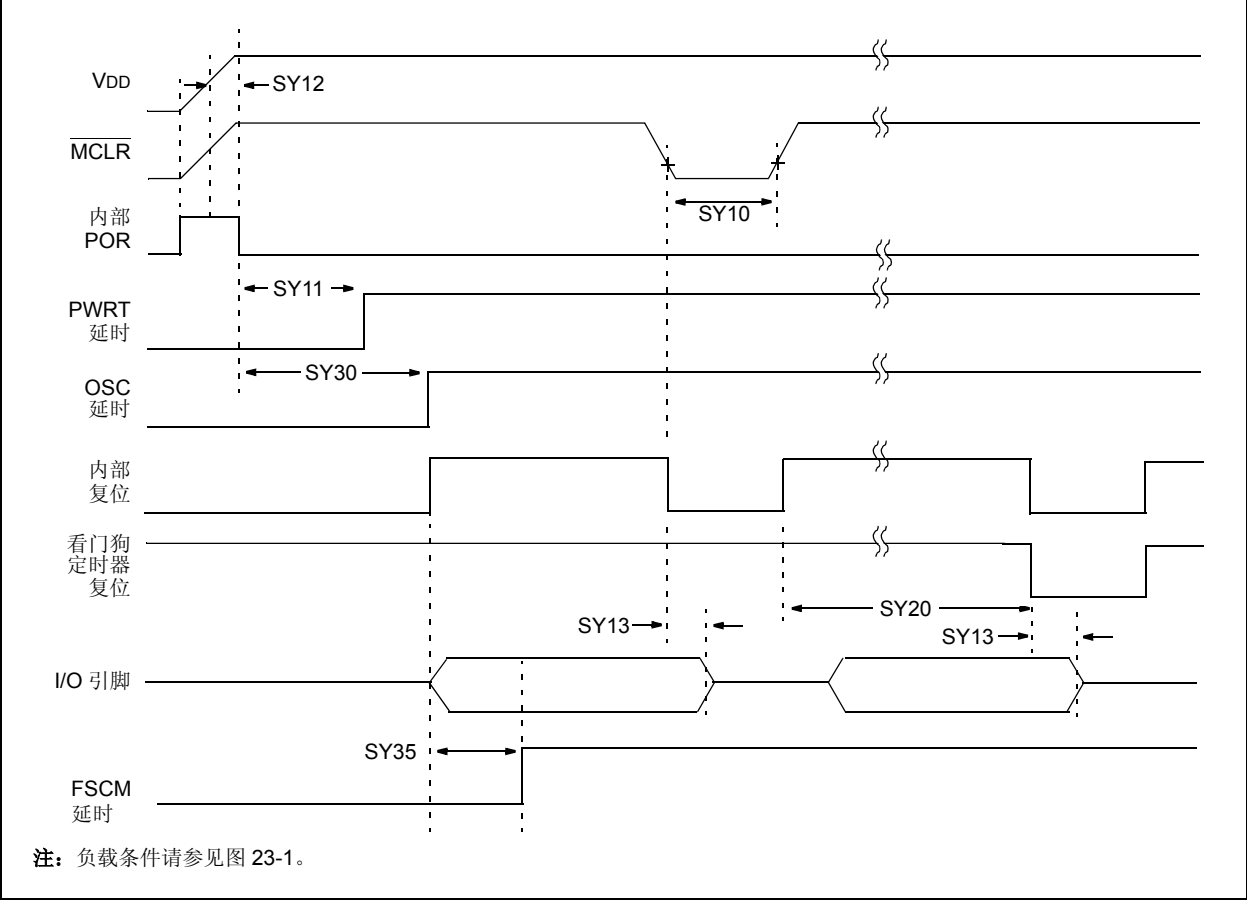


表 23-22: 复位、看门狗定时器、振荡器起振定时器和上电延时定时器时序要求

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	—	—	—	ms	见第 20.4 节 “看门狗定时器 (WDT)” 和 LPRC 参数 F21 (表 23-20)。
SY30	TOST	振荡器起振时间	—	1024 TOSC	—	—	TOSC = OSC1 周期

注 1: 这些参数为特性值, 但生产时未经测试。

2: 除非另外声明, 否则 “典型值” 栏中的数据均为 3.3V、+25°C 条件下的值。

图 23-5: TIMER1、TIMER2 和 TIMER3 外部时钟时序特性

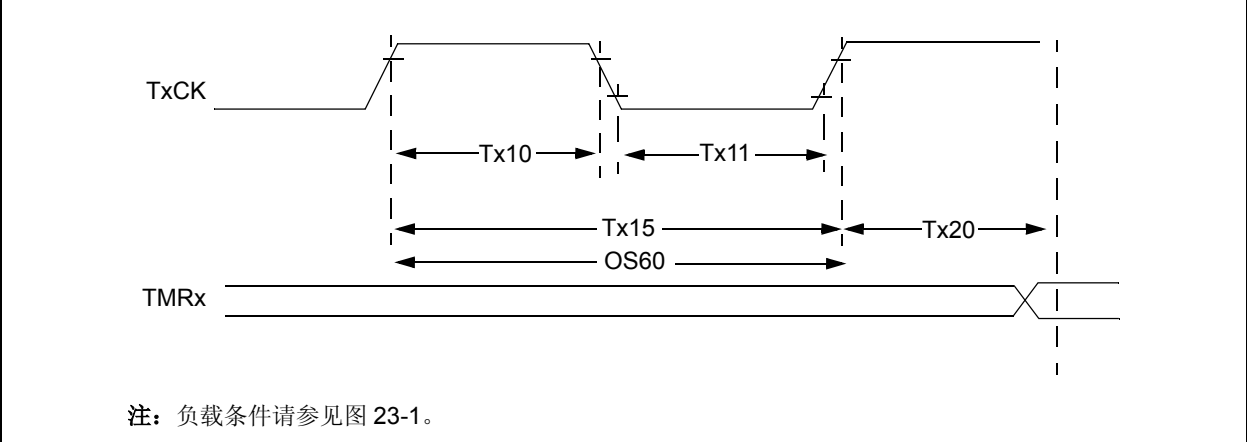


表 23-23: TIMER1 外部时钟时序要求⁽¹⁾

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	TtxH	TxCK 高电平时间	同步, 无预分频器	0.5 Tcy +20	—	—	ns	也必须满足参数 TA15
			同步, 带预分频器	10	—	—	ns	
			异步	10	—	—	ns	
TA11	TtxL	TxCK 低电平时间	同步, 无预分频器	0.5 Tcy +20	—	—	ns	也必须满足参数 TA15
			同步, 带预分频器	10	—	—	ns	
			异步	10	—	—	ns	
TA15	TtxP	TxCK 输入周期	同步, 无预分频器	Tcy +40	—	—	ns	N = 预分频值 (1、8、64 或 256)
			同步, 带预分频器	20 ns 和 (Tcy + 40)/N 中的较大值	—	—	—	
			异步	20	—	—	ns	
OS60	Ft1	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	TCKEXTMRL	从外部 TxCK 时钟边沿到定时器递增之间的延时		0.5 Tcy		1.5 Tcy	—	

注 1: Timer1 属于 A 类定时器。

表 23-24: 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 ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)	
参数编号	符号	特性		最小值	典型值	最大值	单位	条件	
TB10	T _{TXH}	TxCK 高电平时间	同步, 无预分频器	0.5 T _{CY} + 20	—	—	ns	也必须满足参数 TB15	
			同步, 带预分频器	10	—	—	ns		
TB11	T _{TXL}	TxCK 低电平时间	同步, 无预分频器	0.5 T _{CY} + 20	—	—	ns	也必须满足参数 TB15	
			同步, 带预分频器	10	—	—	ns		
TB15	T _{TXP}	TxCK 输入周期	同步, 无预分频器	T _{CY} + 40	—	—	ns	N = 预分频值 (1、8、64 或 256)	
			同步, 带预分频器	20 ns 和 (T _{CY} + 40)/N 中的较大值					
TB20	T _{CKEXTMRL}	从外部 TxCK 时钟边沿到定时器递增之间的延时		0.5 T _{CY}	—	1.5 T _{CY}	—		

表 23-25: 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 ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)	
参数编号	符号	特性		最小值	典型值	最大值	单位	条件	
TC10	T _{TXH}	TxCK 高电平时间	同步	0.5 T _{CY} + 20	—	—	ns	也必须满足参数 TC15	
TC11	T _{TXL}	TxCK 低电平时间	同步	0.5 T _{CY} + 20	—	—	ns	也必须满足参数 TC15	
TC15	T _{TXP}	TxCK 输入周期	同步, 无预分频器	T _{CY} + 40	—	—	ns	N = 预分频值 (1、8、64 或 256)	
			同步, 带预分频器	20 ns 和 (T _{CY} + 40)/N 中的较大值					
TC20	T _{CKEXTMRL}	从外部 TxCK 时钟边沿到定时器递增之间的延时		0.5 T _{CY}	—	1.5 T _{CY}	—		

图 23-6: 输入捕捉 (CAPx) 时序特性

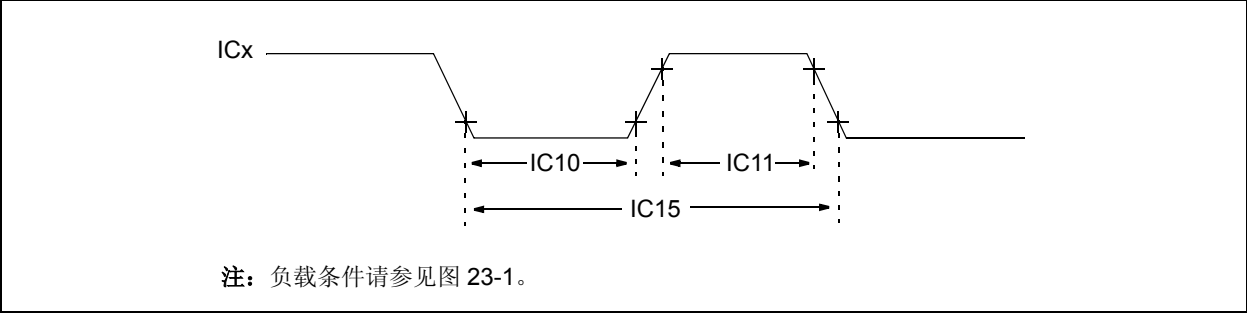


表 23-26: 输入捕捉时序要求

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	—	
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	—	
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 23-7: 输出比较模块 (OCx) 时序特性

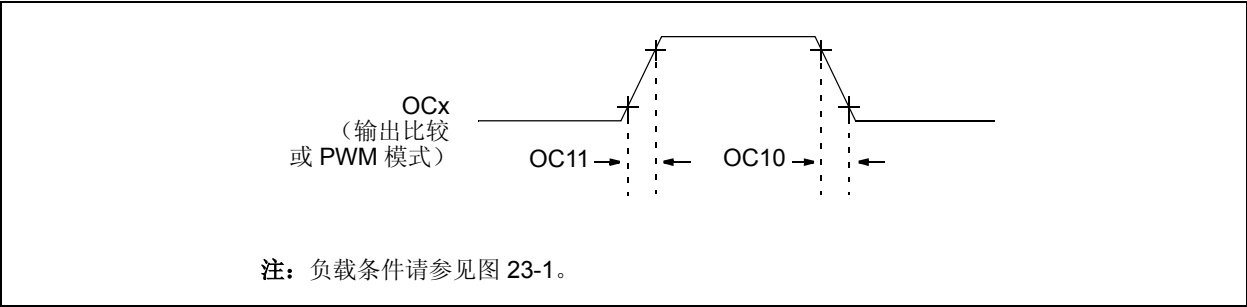


表 23-27: 输出比较模块时序要求

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 23-8: 输出比较 /PWM 模块时序特性

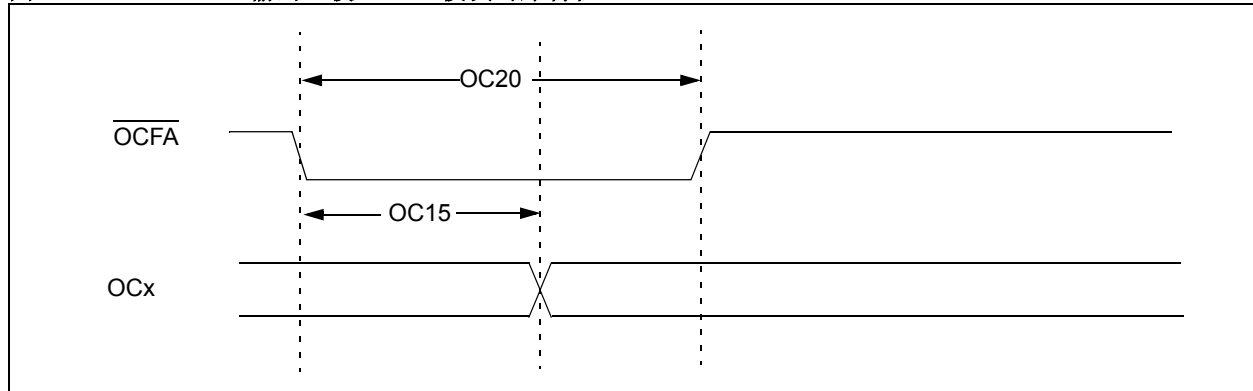


表 23-28: 简单输出比较 /PWM 模式时序要求

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 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (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 ⁽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 23-9: 高速 PWM 模块故障时序特性

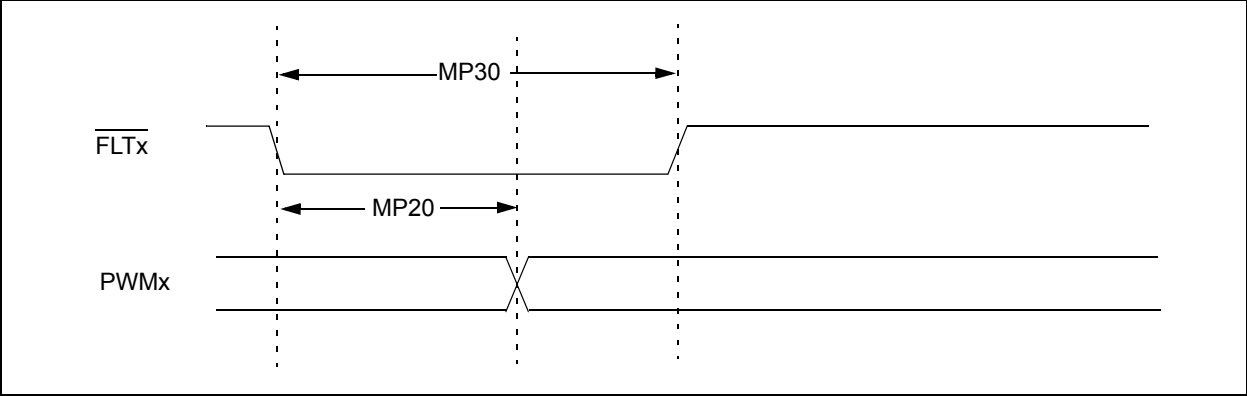
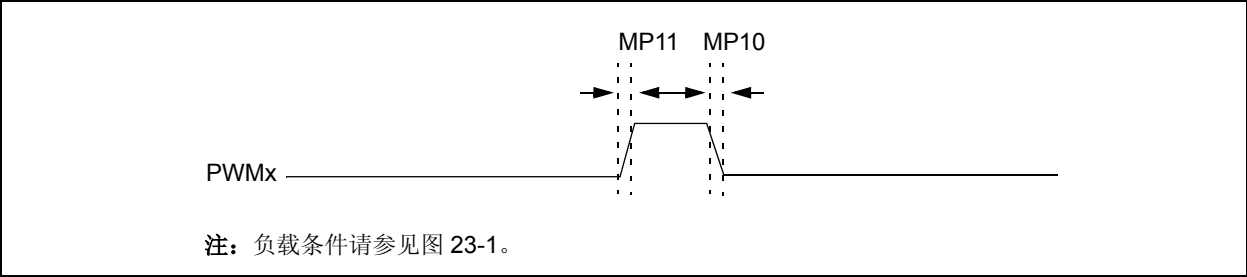


图 23-10: 高速 PWM 模块时序特性



注：负载条件请参见图 23-1。

表 23-29: 高速 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	—	2.5	—	ns	见参数 D032
MP11	TRPWM	PWM 输出上升时间	—	2.5	—	ns	见参数 D031
MP20	TFD	故障输入 ↓ 到 PWM I/O 变化的时间	—	—	15	ns	
MP30	TFH	最小脉冲宽度	—	8	—	ns	
	TPDLY	控制延时	—	1.04	—	ns	

注 1: 这些参数为特性值, 但生产时未经测试。

图 23-11: SPIx 模块主模式 (CKE = 0) 时序特性

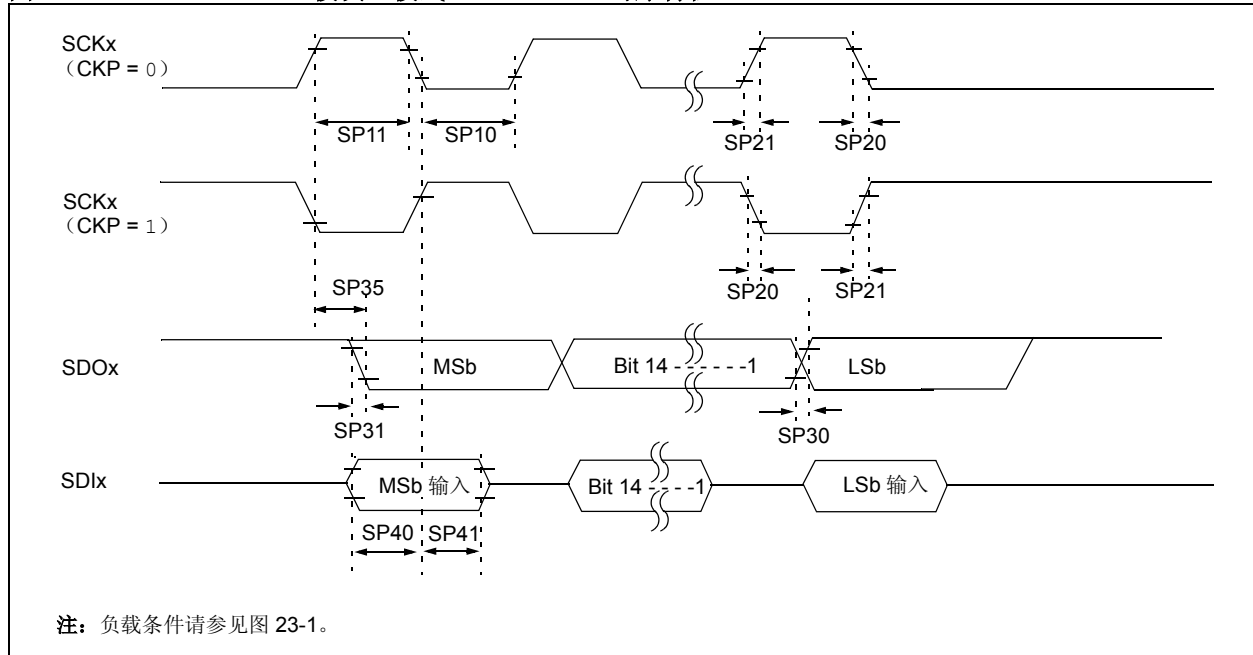


表 23-30: SPIx 主模式 (CKE = 0) 时序要求

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 ⁽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	
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。
 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。
 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。
 4: 假定所有 SPIx 引脚上的负载均为 50 pF。

图 23-12: SPIx 模块主模式 (CKE = 1) 时序特性

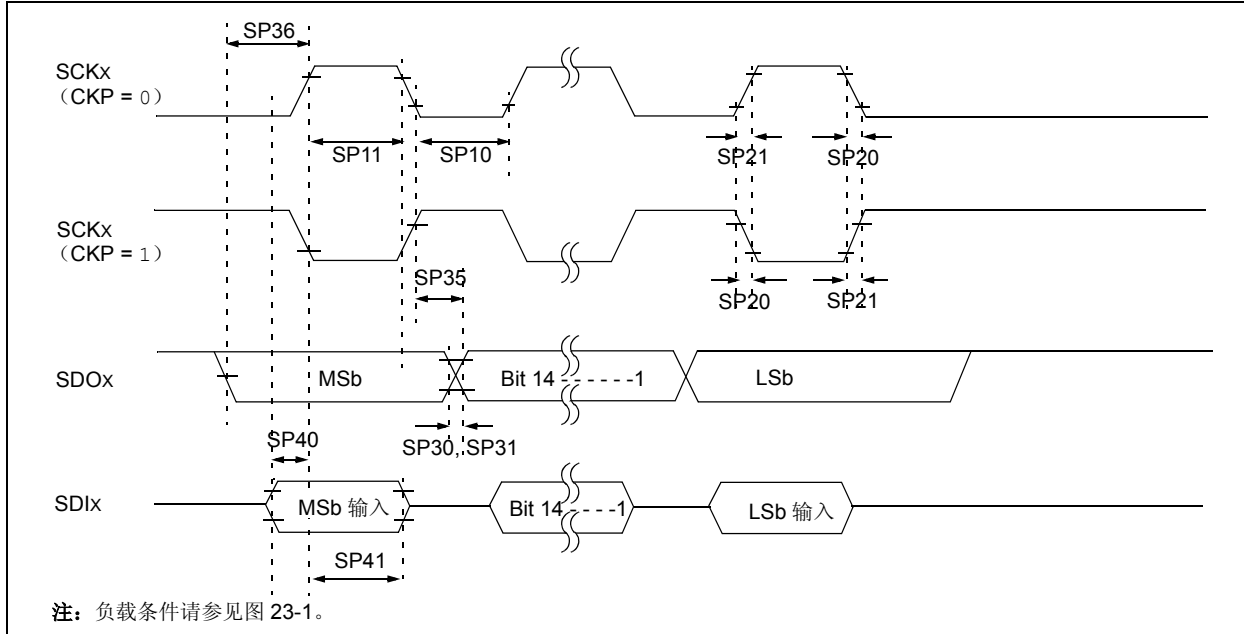


表 23-31: SPIx 模块主模式 (CKE = 1) 时序要求

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。
- 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。
- 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。
- 4: 假定所有 SPIx 引脚上的负载均为 50 pF。

图 23-13: SPIx 模块从模式 (CKE = 0) 时序特性

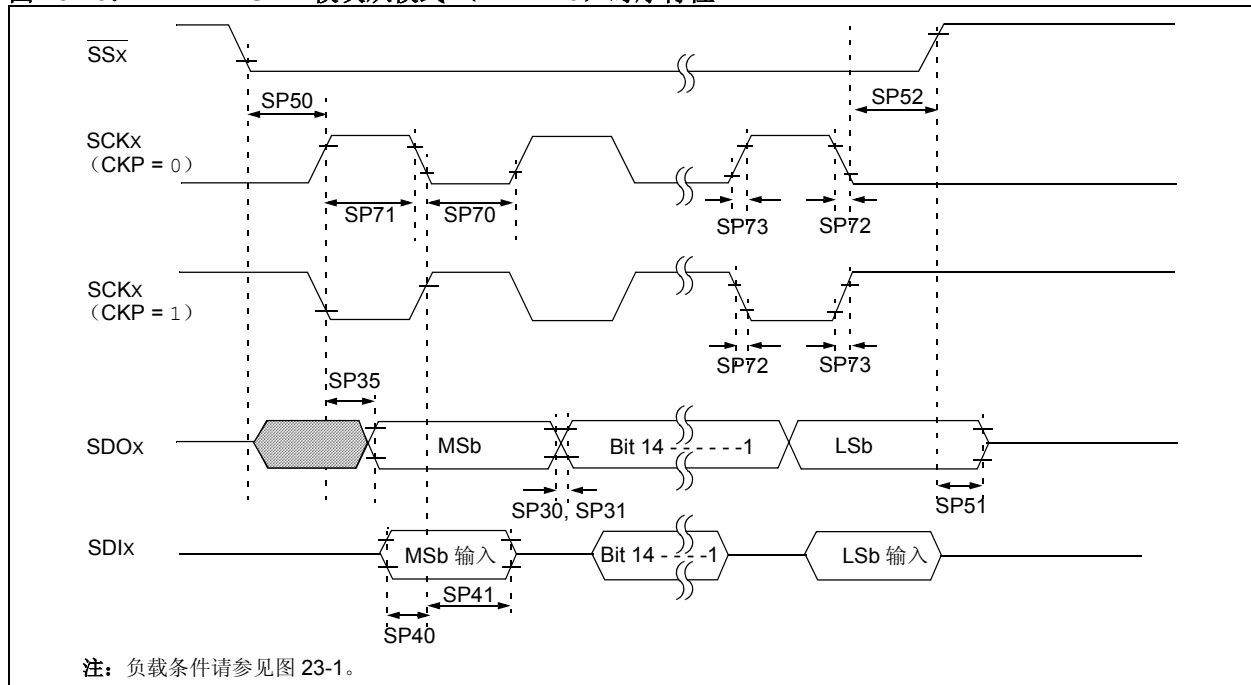


表 23-32: SPIx 模块从模式 (CKE = 0) 时序要求

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。

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。

3: 假定所有 SPIx 引脚上的负载均为 50 pF。

图 23-14: SPIx 模块从模式 (CKE = 1) 时序特性

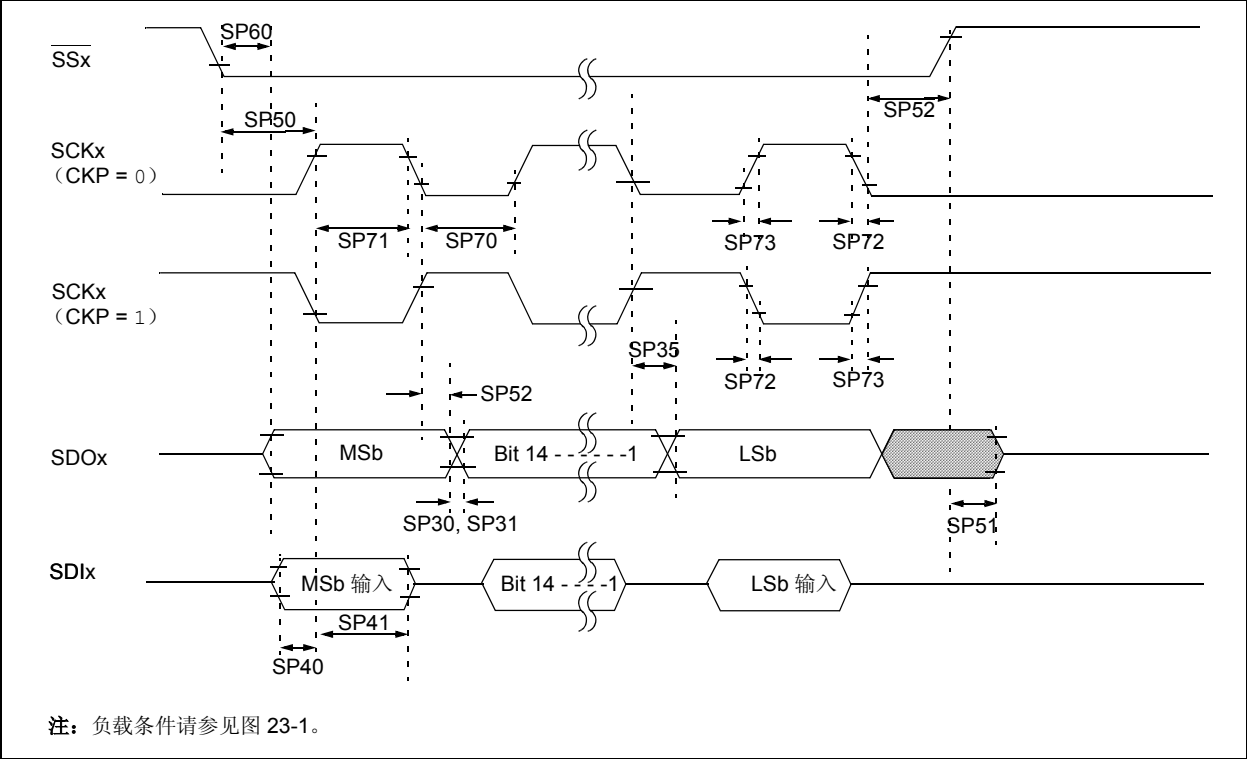


表 23-33: SPIx 模块从模式 (CKE = 1) 时序要求

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	$\overline{SSx}$ ↓ 到 SCKx ↓ 或 SCKx ↑ 输 入的时间	120	—	—	ns	
SP51	TssH2doZ	$\overline{SSx}$ ↑ 到 SDOx 输出高阻态的 时间	10	—	50	ns	见注 4
SP52	Tsch2ssH, TscL2ssH	SCKx 边沿后 $\overline{SSx}$ ↑ 有效的时 间	1.5 Tcy + 40	—	—	ns	
SP60	TssL2doV	在 $\overline{SSx}$ 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。

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。

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。

4: 假定所有 SPIx 引脚上的负载均为 50 pF。

图 23-15: I2Cx 总线启动位 / 停止位时序特性（主模式）

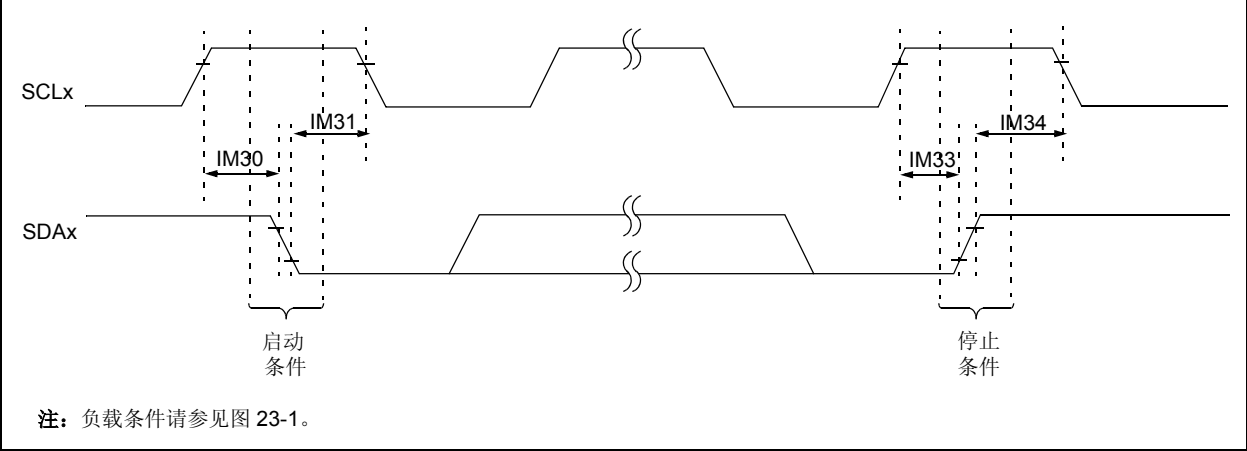
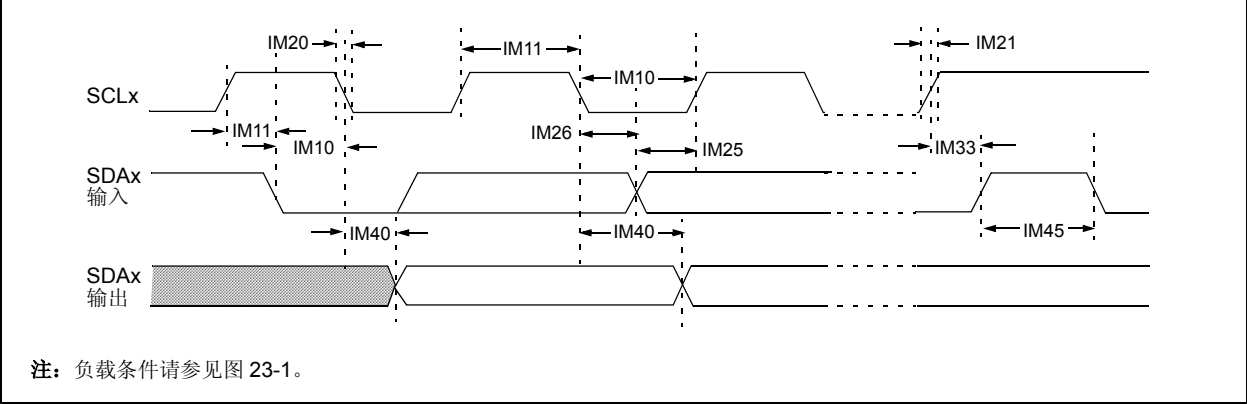


图 23-16: I2Cx 总线数据时序特性（主模式）



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 23-34: 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 ⁽¹⁾	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 ⁽²⁾	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 ⁽²⁾	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 ⁽²⁾	—	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 ⁽²⁾	—	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 ⁽²⁾	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 ⁽²⁾	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 ⁽²⁾	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 ⁽²⁾	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 ⁽²⁾	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 ⁽²⁾	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 ⁽²⁾	—	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 ⁽²⁾	0.5	—	μs	
IM50	CB	总线容性负载		—	400	pF	

注 1: BRG 为 I²C™ 波特率发生器的值。请参见 “dsPIC33F Family Reference Manual” 的 “Section 19. Inter-Integrated Circuit (I²C™)” (DS70195)，该文档可从 Microchip 网站下载。

2: 所有 I2Cx 引脚的最大引脚电容为 10 pF（仅限 1 MHz 模式）。

图 23-17: I2Cx 总线启动位 / 停止位时序特性 (从模式)

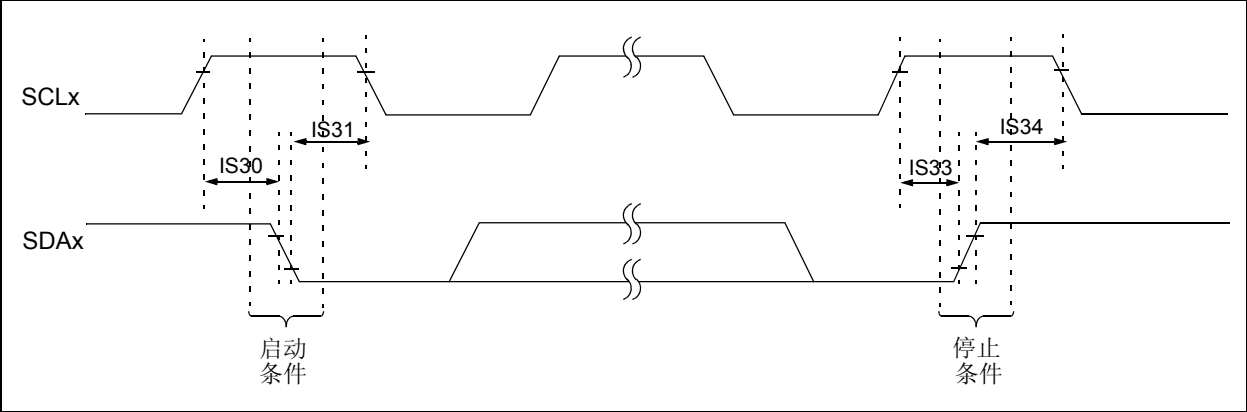
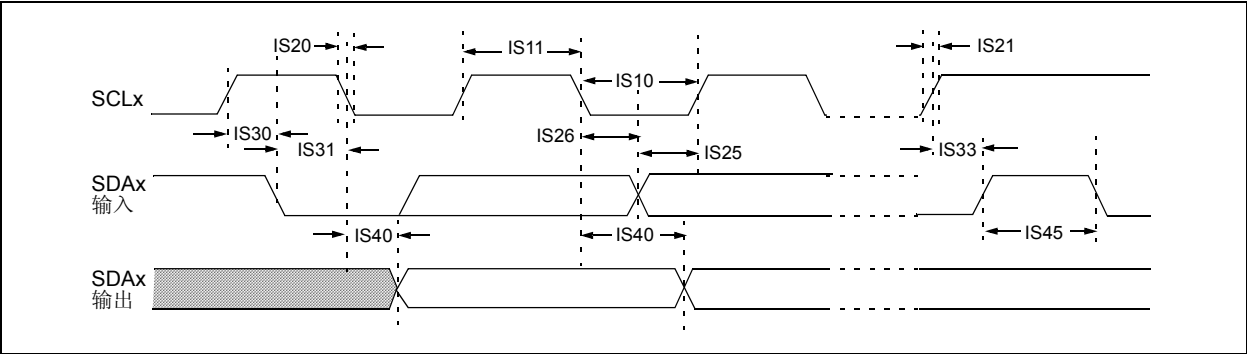


图 23-18: I2Cx 总线数据时序特性 (从模式)



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 23-35: 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 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (扩展级)			
参数	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	$20 + 0.1 C_B$	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	$20 + 0.1 C_B$	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)。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 23-36: 10 位高速 A/D 模块规范

交流特性			标准工作条件: 3.0V 和 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	
模拟输入							
AD10	VINH-VINL	满幅输入范围	VSS		VDD	V	
AD11	VIN	绝对输入电压	AVSS		AVDD	V	
AD13	—	漏电流	—	±0.001	±0.610	μA	VINL = AVSS = 0V, AVDD = 3.3V 信号源阻抗 = 100 Ω
AD17	RIN	模拟输入信号源的推荐阻抗	—		100	Ω	
直流精度							
AD20	Nr	分辨率	10 个数据位			位	
AD21A	INL	积分非线性误差	—	±0.5	<±2	LSb	VINL = AVSS = 0V, AVDD = 3.3V
AD22A	DNL	微分非线性误差	—	±0.5	<±1	LSb	VINL = AVSS = 0V, AVDD = 3.3V
AD23A	GERR	增益误差	—	±0.75	<±3.0	LSb	VINL = AVSS = 0V, AVDD = 3.3V
AD24A	E _{OFF}	失调误差	—	±2.0	<±5.0	LSb	VINL = AVSS = VSS = 0V, AVDD = VDD = 3.3V
AD25	—	单调性 ⁽¹⁾	—	—	—	—	保证
动态性能							
AD30	THD	总谐波失真	—	-73	—	dB	
AD31	SINAD	信噪比和失真	—	58	—	dB	
AD32	SFDR	无杂散动态范围	—	-73	—	dB	
AD33	FNYQ	输入信号带宽	—	—	0.5	MHz	
AD34	ENOB	有效位数	—	9.4	—	位	

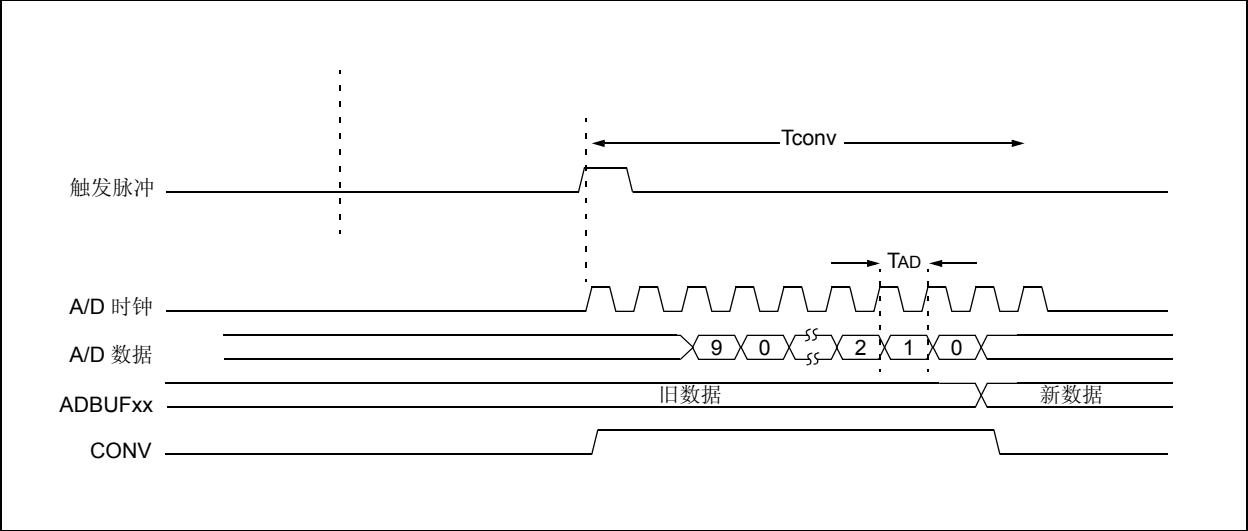
注 1: A/D 转换结果不会随着输入电压的升高而减小, 且不会丢失编码。

表 23-37: 10 位高速 A/D 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	最小值	典型值 (1)	最大值	单位	条件
时钟参数							
AD50b	TAD	ADC 时钟周期	41.6	—	—	ns	
转换速率							
AD55b	tCONV	转换时间	—	12 TAD	—	—	
AD56b	FCNV	吞吐量					
		带单个 SAR 的器件	2.0	—	—	Msp/s	
		带两个 SAR 的器件	4.0	—	—	Msp/s	
AD57b	TSAMP	采样时间	2.0 TAD	—	—	—	
时序参数							
AD60b	tPCS	从采样触发到转换开始的时间 (2)	—	—	1.0 TAD	—	
AD63b	tDPU	从 ADC 关闭到 ADC 启动, 用于稳定模拟级的时间 (1)	1.0	—	10	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。

图 23-19: 每路输入的 A/D 转换时序



dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

表 23-38: 比较器的交流和直流规范

			标准工作条件（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	备注
	VIOFF	输入失调电压		±5	±15	mV	
	VICM	输入共模电压范围 ⁽¹⁾	0	—	AVDD – 1.5	V	
	VGAIN	开环增益 ⁽¹⁾	90	—	—	db	
	CMRR	共模抑制比 ⁽¹⁾	70	—	—	db	
	TRESP	大信号响应		20	30	ns	V- 输入保持在 AVDD/2 时的 V+ 输入步长为 100 mv。测量从模拟输入引脚到 PWM 输出引脚的延时。

注 1: 这些参数仅供设计参考，生产时未经测试。

表 23-39: DAC 直流规范

			标准工作条件（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	备注
	CVRSRC	外部参考电压 ⁽¹⁾	0		AVDD – 1.6	V	
	CVRES	分辨率		10		位	
	INL	传递函数精度	—	±1.0	—	LSB	AVDD = 3.3V, DACREF = (AVDD/2)V
	DNL	积分非线性误差	—	±0.8	—	LSB	
	EOFF	微分非线性误差	—	±2.0	—	LSB	
	EG	失调误差	—	±2.0	—	LSB	

注 1: 这些参数仅供设计参考，生产时未经测试。

表 23-40: DAC 交流规范

			标准工作条件（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	备注
	TSET	稳定时间 ⁽¹⁾			650	ns	在 RANGE = 1（高电压范围）且 CMREF<9:0> 从 0x1FF 变为 0x300 的情况下进行测量。

注 1: 这些参数仅供设计参考，生产时未经测试。

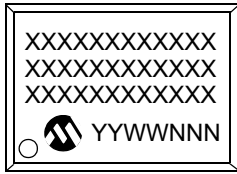
表 23-41: DAC 输出缓冲器直流规范

			标准工作条件（除非另外声明）				
			工作温度: -40°C ≤ T _A ≤ +85°C（工业级）				
			-40°C ≤ T _A ≤ +125°C（扩展级）				
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
	RLOAD	阻性输出负载阻抗	3K	—	—	Ω	
	CLOAD	输出负载电容值	—	20	35	pF	
	IOUT	输出电流驱动能力	200	300	400	μA	拉电流和灌电流
	VRANGE	满额输出驱动能力电压范围	AVSS + 250 mV	—	AVDD – 900 mV	V	
	VLRANGE	降额电流驱动（50 μA）时的输出驱动电压范围	AVSS + 50 mV	—	AVDD – 500 mV	V	
	IDD	模块使能时消耗的电流，高功耗模式	—	—	1.3 x IOUT	μA	即使输出不连接负载，模块也始终消耗这一电流
	RIN	输入阻抗	10 ⁹	—	—	Ω	
	ROUTON	模块使能时的输出阻抗	—	—	10	Ω	闭环输出阻抗
	ROUTOFF	模块禁止时的输出阻抗	10 ⁷	—	—	Ω	buf_enable = 0

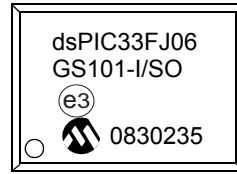
注:

24.0 封装信息

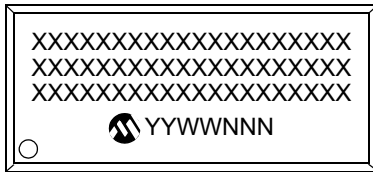
18 引脚 SOIC (.300")



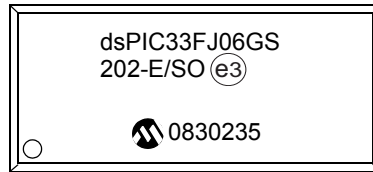
示例



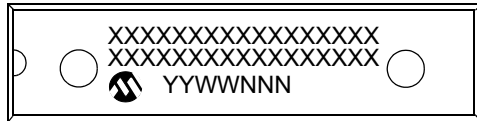
28 引脚 SOIC



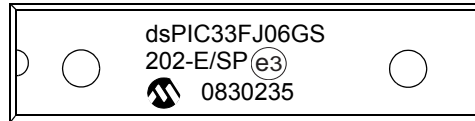
示例



28 引脚 SPDIP



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
*	本封装为无铅封装。JEDEC 无铅标志 (e3) 标示于此种封装的外包装上。

注: Microchip 元器件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户信息的字符数。

24.1 封装标识信息（续）

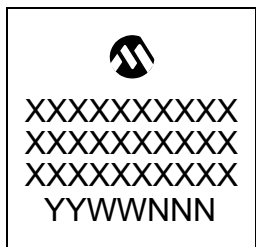
28 引脚 QFN-S



示例



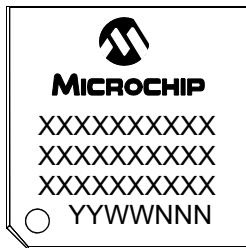
44 引脚 QFN



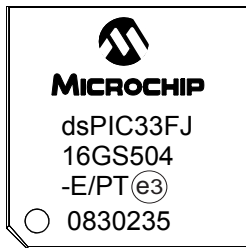
示例



44 引脚 TQFP



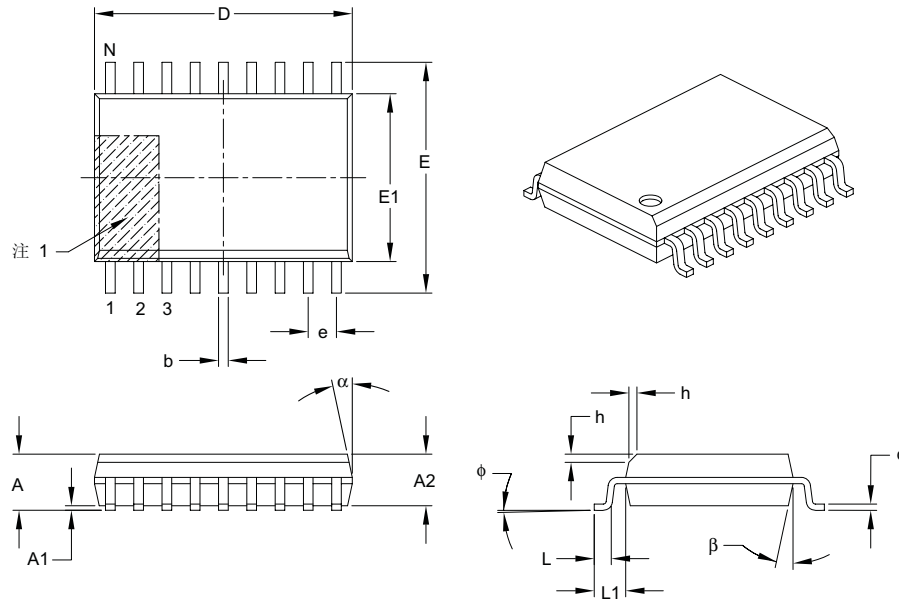
示例



24.2 封装详细信息

18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	18		
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	11.55 BSC		
斜面（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	φ	0°	—	8°
引脚厚度	c	0.20	—	0.33
引脚宽度	b	0.31	—	0.51
塑模顶部锥度	α	5°	—	15°
塑模底部锥度	β	5°	—	15°

注：

1. 引脚1的可见定位功能可能不同，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. 尺寸和公差请参见ASME Y14.5M。

BSC：基本尺寸。显示的是没有公差的理论精确值。

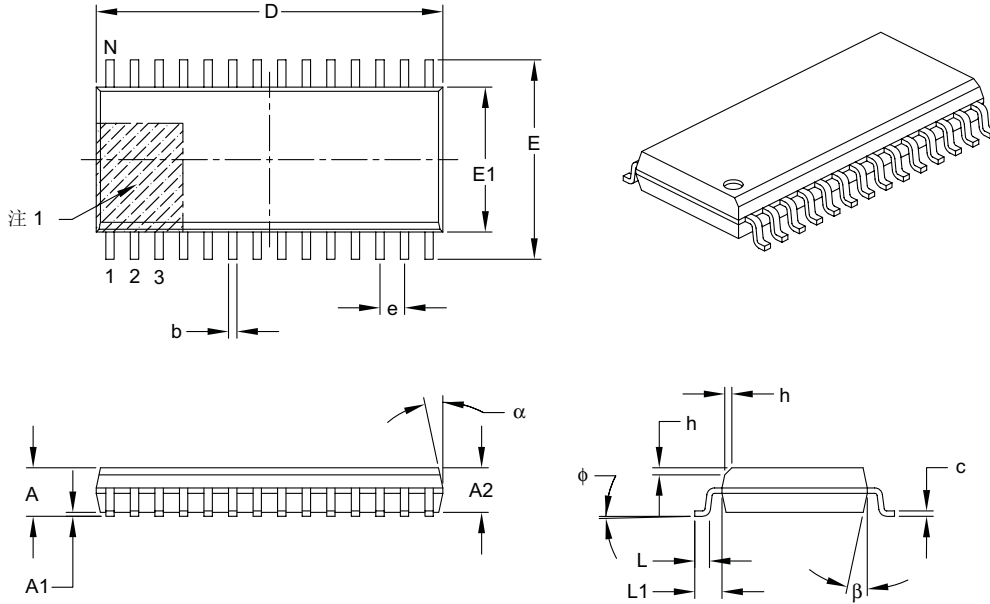
REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology图号 C04-051B

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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。



单位	尺寸范围	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	φ	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	α	5°	—	15°
塑模底部锥度	β	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。

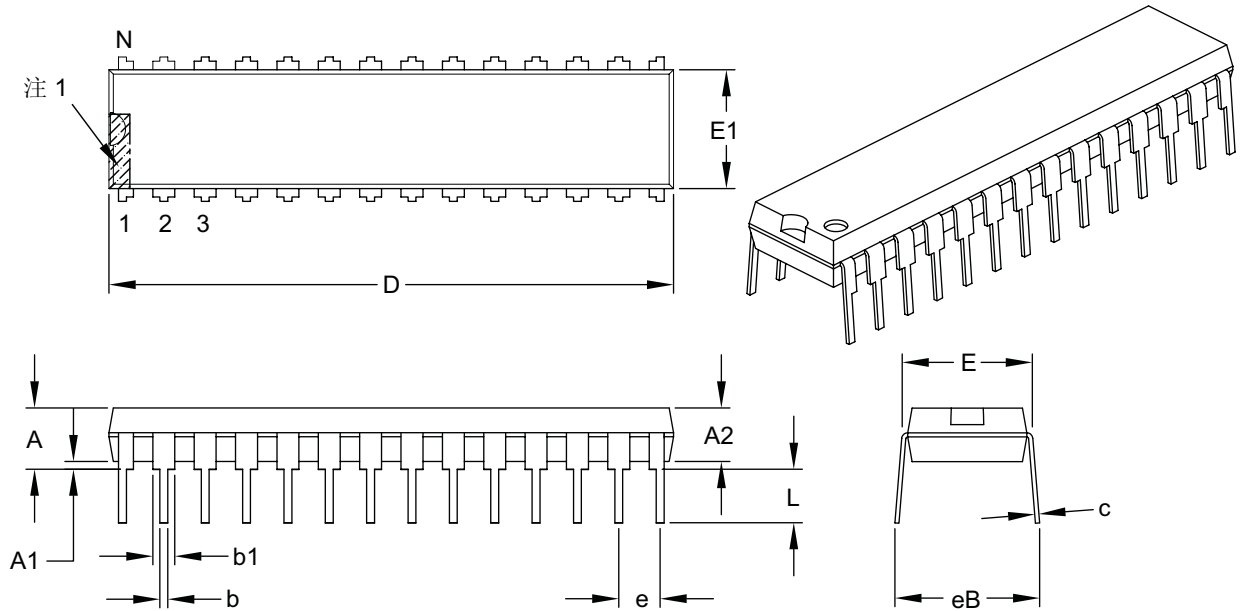
REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology图号C04-052B

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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。



	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。

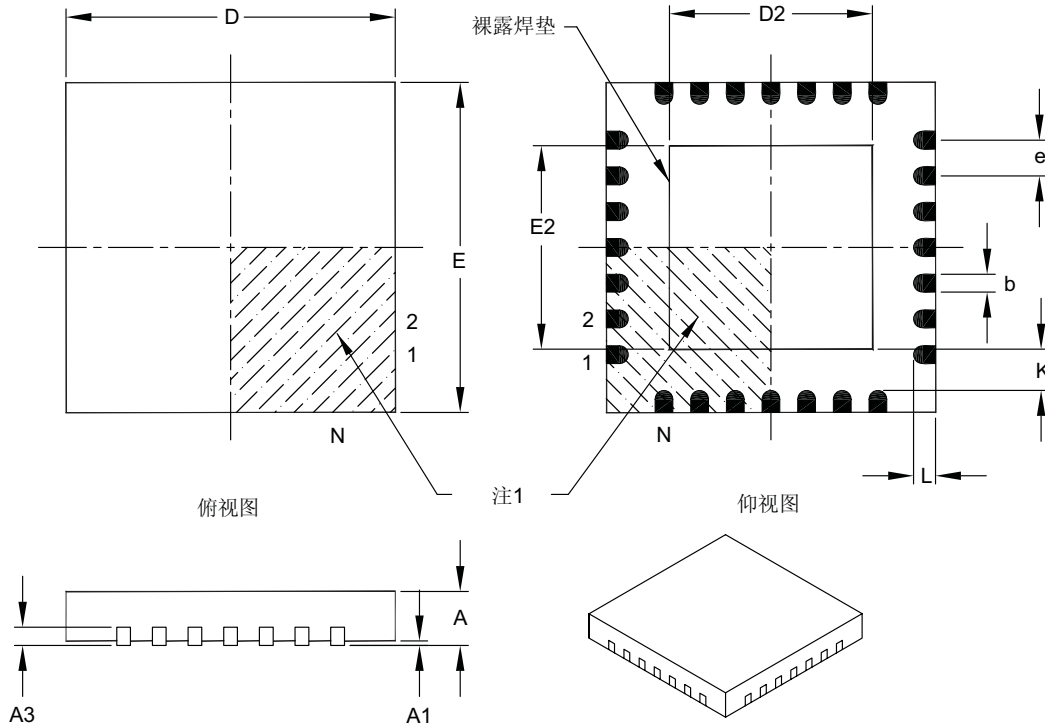
BSC：基本尺寸。显示的是没有公差的理论精确值。

Microchip Technology图号C04-070B

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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。



单位		毫米		
尺寸范围		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。

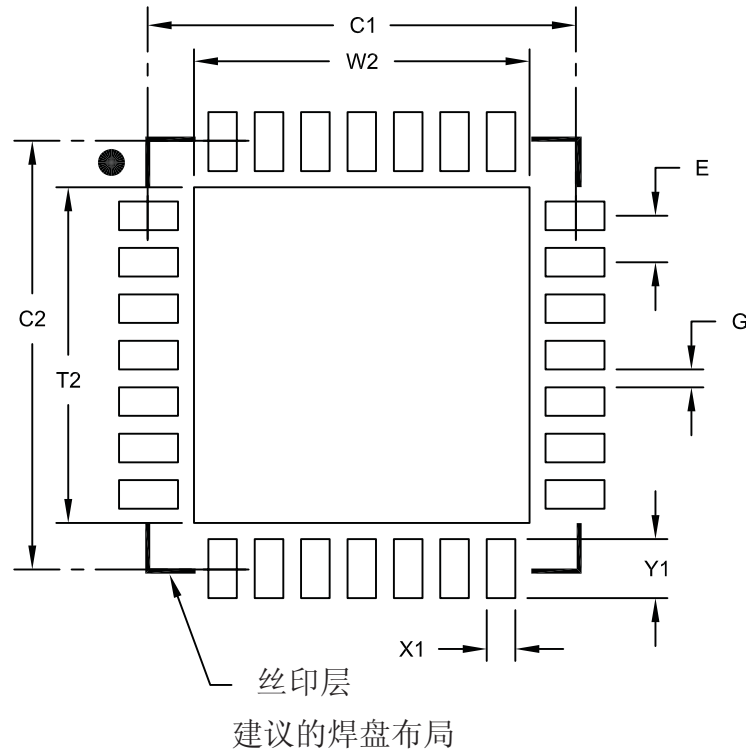
BSC：基本尺寸。显示的是没有公差的理论精确值。

REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology 图号C04-124B

28 引脚塑封四方扁平无引线封装（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。



单位		毫米		
尺寸范围		最小	正常	最大
触点间距	E	0.65 BSC		
可选的中心焊盘宽度	W2			4.70
可选的中心焊盘长度	T2			4.70
触点焊盘空间	C1		6.00	
触点焊盘空间	C2		6.00	
触点焊盘宽度 (X28)	X1			0.40
触点焊盘长度 (X28)	Y1			0.85
焊盘间的距离	G	0.25		

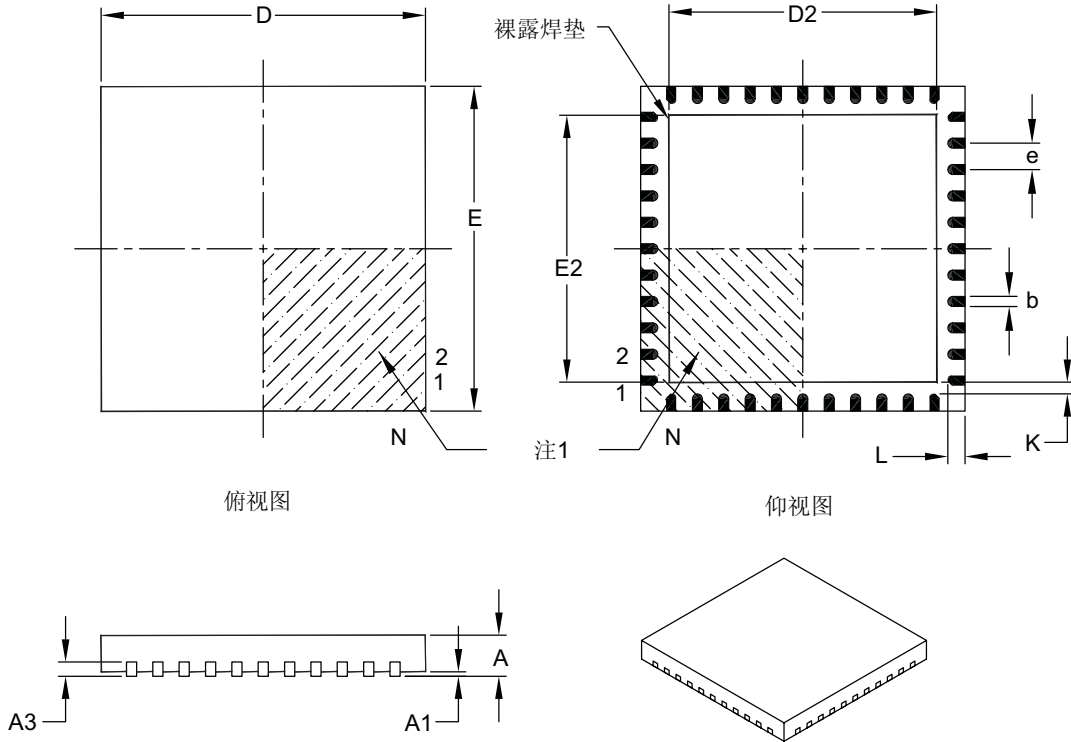
注：

1. 尺寸和公差遵循ASME Y14.5M
BSC：基本尺寸。理论精确值，不含公差。

Microchip Technology图号C04-2124A

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。



单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N	44		
引脚间距	e	0.65 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。

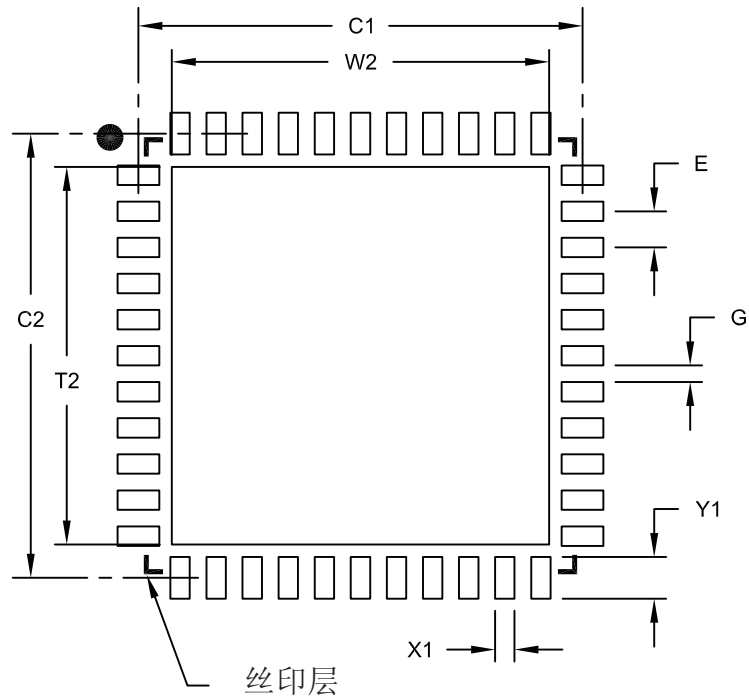
BSC：基本尺寸。显示的是没有公差的理论精确值。

REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology 图号C04-103B

44 引脚塑封四方扁平无引线封装（ML）——8x8x8mm 主体（QFN）

注：最新封装图请至 <http://www.microchip.com/packageing> 查看 Microchip 封装规范。



建议的焊盘布局

单位		毫米		
尺寸范围		最小	正常	最大
触点间距	E	0.65 BSC		
可选的中心焊盘宽度	W2			6.80
可选的中心焊盘长度	T2			6.80
触点焊盘空间	C1		8.00	
触点焊盘空间	C2		8.00	
触点焊盘宽度 (X44)	X1			0.35
触点焊盘长度 (X44)	Y1			0.80
焊盘间的距离	G	0.25		

注：

1. 尺寸和公差遵循ASME Y14.5M

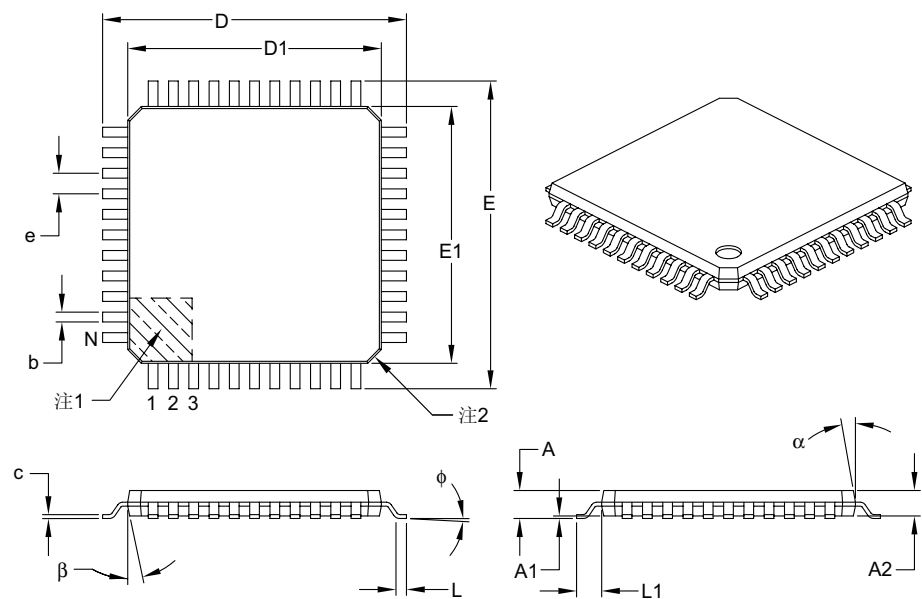
BSC: 基本尺寸。理论精确值，不含公差。

Microchip Technology图号C04-2103A

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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。



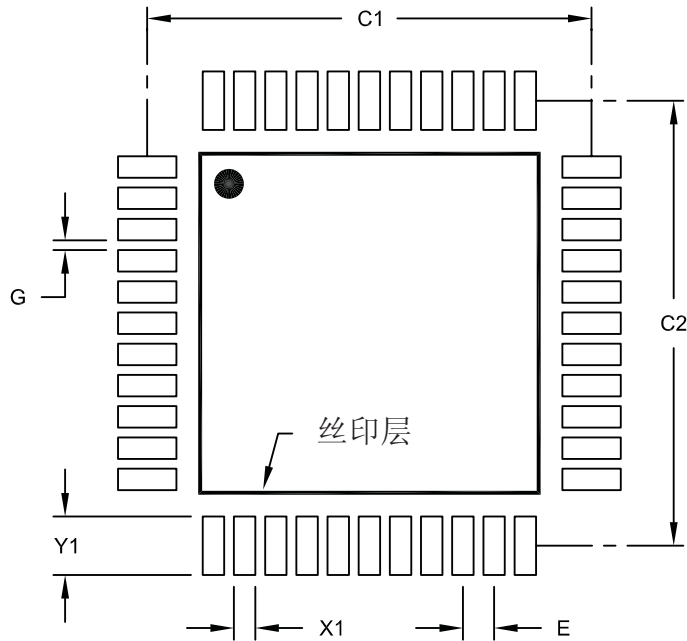
单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N	44		
引脚间距	e	0.80 BSC		
总高度	A	—	—	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	—	0.15
底脚长度	L	0.45	0.60	0.75
引脚投影长度	L1	1.00 REF		
底脚倾斜角	φ	0°	3.5°	7°
总宽度	E	12.00 BSC		
总长度	D	12.00 BSC		
塑模封装宽度	E1	10.00 BSC		
塑模封装长度	D1	10.00 BSC		
引脚厚度	c	0.09	—	0.20
引脚宽度	b	0.30	0.37	0.45
塑模顶部锥度	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 引脚塑封薄型四方扁平封装（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。



建议的焊盘布局

尺寸范围	单位	毫米		
		最小	正常	最大
触点间距	E	0.80 BSC		
触点焊盘空间	C1		11.40	
触点焊盘空间	C2		11.40	
触点焊盘宽度 (X44)	X1			0.55
触点焊盘长度 (X44)	Y1			1.50
焊盘间的距离	G	0.25		

注：

1. 尺寸和公差遵循ASME Y14.5M
BSC：基本尺寸。理论精确值，不含公差。

Microchip Technology图号C04-2076A

注:

附录 A：版本历史

版本 B（2008 年 6 月）

版本 A（2008 年 1 月）

本文档的初始版本。

此版本对整个数据手册的打字、语法和格式进行了较少的更改。此外，删除了一些冗余信息，这些信息已在“*dsPIC33F Family Reference Manual*”的各章节中提供，此参考手册可从 Microchip 网站（www.microchip.com）获得。

下表列出了主要的更改，并指明了相应的章节。

表 A-1：主要章节更新

章节名	更新说明
“高性能 16 位数字信号控制器”	移动了注 1（RPN 引脚）附注的位置（见“引脚图”）。
第 3.0 节“存储器构成”	更新了 CORCON CPU 内核寄存器映射 SFR 的复位值（见表 3-1）。 删除了中断控制器寄存器映射 SFR IPC29，并更新了 IPC0、IPC1、IPC14、IPC16、IPC23、IPC24、IPC27 和 IPC28 的复位值（见表 3-5）。 删除了中断控制器寄存器映射 SFR IPC24 和 IPC29，并更新了 IPC0、IPC1、IPC2、IPC14、IPC16、IPC23、IPC27 和 IPC28 的复位值（见表 3-6）。 删除了中断控制器寄存器映射 SFR IPC24，并更新了 IPC1、IPC2、IPC4、IPC14、IPC16、IPC23、IPC24、IPC27 和 IPC28 的复位值（见表 3-7）。 更新了中断控制器寄存器映射 SFR IPC1、IPC14、IPC16、IPC23、IPC24、IPC27 和 IPC28 的复位值（见表 3-8）。 更新了中断控制器寄存器映射 SFR IPC1、IPC14、IPC16、IPC23、IPC24、IPC25、IPC26、IPC27、IPC28 和 IPC29 的复位值（见表 3-9）。 更新了中断控制器寄存器映射 SFR IPC1、IPC4、IPC14、IPC16、IPC23、IPC24、IPC25、IPC26、IPC27、IPC28 和 IPC29 的复位值（见表 3-10）。 添加了 RPOR16 和 RPOR17 的 SFR 定义（见表 3-34、表 3-35 和表 3-36）。 更新了 PORTA、PORTB 和 PORTC SFR 的位定义（ODCA、ODCB 和 ODCC）（见表 3-37、表 3-38、表 3-39 和表 3-40）。 更新了系统控制寄存器映射 SFR CLKDIV 的位定义和复位值（见表 3-41）。 为 PMD 寄存器映射的标题添加了针对器件的信息（见表 3-46）。 添加了针对器件的 PMD 寄存器映射（见表 3-45、表 3-44 和表 3-43）。
第 7.0 节“振荡器配置”	在第 7.1.1 节“系统时钟源”中删除了第三个时钟源项（外部时钟）的第一句 更新了时钟分频比寄存器中 DOZE 和 FRCDIV 的默认位值（见寄存器 7-2）。

表 A-1: 主要章节更新 (续)

章节名	更新说明
第 8.0 节 “节能特性”	添加了如下 6 个寄存器： • “PMD1: 外设模块禁止控制寄存器 1” • “PMD2: 外设模块禁止控制寄存器 2” • “PMD3: 外设模块禁止控制寄存器 3” • “PMD4: 外设模块禁止控制寄存器 4” • “PMD6: 外设模块禁止控制寄存器 6” • “PMD7: 外设模块禁止控制寄存器 7”
第 9.0 节 “I/O 端口”	向第 9.1.1 节 “漏极开路配置” 添加了段落和表 9-1，以提供 I/O 引脚及其功能的详细信息。 删除了 9.1.2 “5V 耐压”。 图 9-2 中更新了 MUX 的范围并删除了虚拟引脚的详细信息。 表 9-1 中更新了 PWM 输入名称的说明。 添加了第 9.4.2.3 节 “虚拟引脚”。 更新了所有外设选择输入寄存器中的位值（见寄存器 9-1 至寄存器 9-14）。 更新了外设选择输出寄存器 RPOR16 和 RPOR17 中的位名称信息（见寄存器 9-30 和寄存器 9-31）。 添加了如下 2 个寄存器： • “RPOR16: 外设引脚选择输出寄存器 16” • “RPOR17: 外设引脚选择输出寄存器 17” 删除了如下章节： • 9.4.2 “可用外设” • 9.4.3.2 “虚拟输入引脚” • 9.4.3.4 “外设映射” • 9.4.5 “外设引脚选择的注意事项”（及其所有子章节）
第 14.0 节 “高速 PWM”	向图 14-1 添加了注 1（可重映射引脚附注）。 向 PWM 主占空比寄存器（寄存器 14-5）、PWM 发生器占空比寄存器（寄存器 14-7）和 PWM 辅助占空比寄存器（寄存器 14-8）添加了注 2（占空比分辨率）。 为 PWM 故障限流控制寄存器（寄存器 14-15）中的 CLSRC 和 FLTSRC 添加了注 2 和注 3，并更新了位信息。
第 15.0 节 “串行外设接口 (SPI)”	删除了如下章节，这些章节已在 “<i>dsPIC33F Family Reference Manual</i>” 的相关章节中提供： • 15.1 “中断” • 15.2 “接收操作” • 15.3 “发送操作” • 15.4 “SPI 设置”（保留了图 14-1：SPI 模块框图）

表 A-1: 主要章节更新 (续)

章节名	更新说明
第 16.0 节 “I ² C™”	删除了如下章节，这些章节已在 “<i>dsPIC33F Family Reference Manual</i>” 的相关章节中提供： • 16.3 “I²C 中断” • 16.4 “波特率发生器”（保留了图 15-1: I²C 框图） • 16.5 “I²C 模块地址” • 16.6 “从地址掩码” • 16.7 “IPMI 支持” • 16.8 “广播呼叫地址支持” • 16.9 “自动时钟延长” • 16.10 “软件控制的时钟延长（STREN = 1）” • 16.11 “斜率控制” • 16.12 “时钟仲裁” • 16.13 “多主器件通信、总线冲突和总线仲裁”
第 17.0 节 “通用异步收发器 (UART)”	删除了如下章节，这些章节已在 “<i>PIC24H Family Reference Manual</i>” 的相关章节中提供： • 17.1 “UART 波特率发生器” • 17.2 “8 位数据模式下的发送” • 17.3 “9 位数据模式下的发送” • 17.4 “间隔和同步字符发送过程” • 17.5 “8 位或 9 位数据模式下的接收” • 17.6 “使用 UxCTS 和 UxRTS 引脚的流控制” • 17.7 “红外支持” 删除了 IrDA 参考信息和注 1，并更新了 UARTx 状态和控制寄存器（见寄存器 17-2）中 UTXINV（UxSTA<14>）的位和位值说明。
第 18.0 节 “高速 10 位模数转换器 (ADC)”	更新了 A/D 控制寄存器（见寄存器 18-1）的位值信息。 更新了 A/D 转换对控制寄存器 3（见寄存器 18-8）中 Timer1 周期匹配的 TRGSRC6 位值。

表 A-1: 主要章节更新 (续)

章节名	更新说明
第 23.0 节 “电气特性”	更新了热封装特性 (表 23-3) 的典型值。 删除了 DC 温度和电压规范参数 DC12 (表 23-4) 的典型值。 更新了 DC 特性的所有典型值和条件。工作电流 (IDD)，更新了注 2 中的最后一句话。 更新了 DC 特性的所有典型值：空闲电流 (IDLE) (见表 23-6)。 更新了 DC 特性的所有典型值：掉电电流 (IPD) (见表 23-7)。 更新了 DC 特性的所有典型值：打盹电流 (IDOZE) (见表 23-8)。 在 I/O 引脚输入规范 (表 23-9) 中添加了注 4 (对包含仅数字和模拟引脚信息的新表格以及灌 / 拉电流能力的附注)。 更新了 BOR 电气特性参数 BO10 的最大值 (见表 23-11)。 交换了程序存储器参数 D136 和 D137 的最小值和典型值 (表 23-12)。 更新了内部 RC 精度参数 F20 的典型值，并向表标题 (见表 23-19) 添加了扩展温度范围。 删除了复位、看门狗定时器、振荡器起振定时器和上电延时定时器参数 SY20 的所有值并更新了条件，请参见第 20.4 节 “看门狗定时器 (WDT)” 和 LPRC 参数 F21 (见表 23-22)。 为高速 PWM 模块时序需求添加了控制延时 (Tap Delay) 规范 (表 23-29)。 更新了 10 位高速 A/D 模块参数 AD01 和 AD11 的最小值和最大值 (见表 23-36)。 更新了 DAC AC 规范的最大值和测量单位 (见表 23-40)。

索引

A

A/D 227

B

版本历史 319
 备用向量表 (AIVT) 85
 比较器 DAC 控制寄存器 (CMPDACx) 252
 比较器控制寄存器 (CMPCONx) 251
 变更通知客户服务 327

C

C 编译器

MPLAB C18 270
 MPLAB C30 270

CPU

控制寄存器 20
 CPU 的特殊功能 253
 CPU 时钟系统 126
 PLL 配置 127
 选择 126
 源 126

操作码说明中使用的符号 262
 程序存储空间与数据存储空间的接口 66

程序存储器

复位向量 30
 构成 30
 中断向量 30
 程序地址空间 29

表读指令

TBLRDH 68
 TBLRDL 68

存储器映射 29
 访问程序空间中数据的地址生成 67
 构成 66
 可视性操作 69
 使用表指令访问程序存储器中的数据 68
 使用程序空间可视性访问程序存储器中的数据 69

串行外设接口 (SPI) 207

存储器构成 29

D

打盹模式 136

DSP 引擎 23

乘法器 25

代码保护 253, 259

代码示例

擦除程序存储器页 75
 端口写 / 读 145
 PWRSV 指令语法 135
 启动编程序列 76
 装载写缓冲区 76

电气特性 273

BOR 280

交流特性和时序参数 282

读者反馈表 328

E

EBCONx (前沿消隐控制寄存器) 205

F

封装 307

18 引脚 SOIC 310

28 引脚 QFN-S 312

28 引脚 QFN-S 封装的焊盘布局 313

28 引脚 SPDIP 311

44 引脚 QFN 314

44 引脚 QFN 封装的焊盘布局 314

44 引脚 TQFP 314

44 引脚 TQFP 封装的焊盘布局 314

标识 307

复位 77

非法操作码 77, 84

未初始化的 W 寄存器 77, 84

陷阱冲突 83, 84

复位过程 85

G

高速 PWM 187

故障保护时钟监视器 134

H

汇编器

MPASM 汇编器 270

I

I/O 端口 143

并行 I/O (PIO) 143

写 / 读时序 145

I²C

工作模式 213

寄存器 213

J

寄存器

..... 205

A/D 端口配置寄存器 (ADPCFG) 238

A/D 控制寄存器 (ADCON) 235

A/D 转换对控制寄存器 0 (ADCPC0) 239

A/D 转换对控制寄存器 1 (ADCPC1) 241

A/D 转换对控制寄存器 2 (ADCPC2) 244

A/D 转换对控制寄存器 3 (ADCPC3) 246

A/D 状态寄存器 (ADSTAT) 237

ALTDTRx (PWM 备用死区寄存器) 198

CLKDIV (时钟分频比) 129

CMPCONx 251

CMPDACx 252

CORCON (内核控制) 22

DTRx (PWMx 死区寄存器) 198

FCLCONx (PWM 故障限流控制寄存器) 202

I2CxCON (I2Cx 控制寄存器) 215

I2CxMSK (I2Cx 从模式地址掩码寄存器) 219

I2CxSTAT (I2Cx 状态寄存器) 217

ICxCON (输入捕捉 x 控制) 182

IEC0 (中断允许控制寄存器 0) 101

IEC1 (中断允许控制寄存器 1) 103

IEC3 (中断允许控制寄存器 3) 104

IEC4 (中断允许控制寄存器 4) 104

IEC5 (中断允许控制寄存器 5) 105

IFS0 (中断标志状态寄存器 0) 94

IFS1 (中断标志状态寄存器 1) 96

IFS3 (中断标志状态寄存器 3) 97

IFS4 (中断标志状态寄存器 4) 97

IFS5 (中断标志状态寄存器 5) 98

IFS6 (中断标志状态寄存器 6) 99, 106

IFS7 (中断标志状态寄存器 7) 100, 107

INTCON1 (中断控制寄存器 1) 91

INTTREG (中断控制和状态寄存器)	122
IOCONx (PWM I/O 控制寄存器)	200
IPC0 (中断优先级控制寄存器 0)	108
IPC14 (中断优先级控制寄存器 14)	114
IPC16 (中断优先级控制寄存器 16)	114
IPC1 (中断优先级控制寄存器 1)	109
IPC23 (中断优先级控制寄存器 23)	115
IPC24 (中断优先级控制寄存器 24)	116
IPC25 (中断优先级控制寄存器 25)	117
IPC26 (中断优先级控制寄存器 26)	118
IPC27 (中断优先级控制寄存器 27)	119
IPC28 (中断优先级控制寄存器 28)	120
IPC29 (中断优先级控制寄存器 29)	121
IPC2 (中断优先级控制寄存器 2)	110
IPC3 (中断优先级控制寄存器 3)	111
IPC4 (中断优先级控制寄存器 4)	112
IPC5 (中断优先级控制寄存器 5)	113
IPC7 (中断优先级控制寄存器 7)	113
MDC (PWM 主占空比寄存器)	192
NVMCON (闪存控制寄存器)	73
NVMKEY (非易失性存储器密钥寄存器)	74
OCxCON (输出比较 x 控制)	185
OSCCON (振荡器控制)	128
PLLFBD (PLL 反馈倍频比)	130
PMD1 (外设模块禁止控制寄存器 1)	137
PMD2 (外设模块禁止控制寄存器 2)	138
PMD3 (外设模块禁止控制寄存器 3)	139
PMD4 (外设模块禁止控制寄存器 4)	139
PMD6 (外设模块禁止控制寄存器 6)	140
PMD7 (外设模块禁止控制寄存器 7)	141
PTCON (PWM 时基控制寄存器)	190
PWMCAPx (主 PWMx 时基捕捉寄存器)	206
PWMCONx (PWMx 控制寄存器)	193
RCON (复位控制)	78
RPINR0 (外设引脚选择输入寄存器 0)	150
RPINR11 (外设引脚选择输入寄存器 11)	154
RPINR18 (外设引脚选择输入寄存器 18)	155
RPINR1 (外设引脚选择输入寄存器 1)	151
RPINR20 (外设引脚选择输入寄存器 20)	156
RPINR21 (外设引脚选择输入寄存器 21)	157
RPINR29 (外设引脚选择输入寄存器 29)	158
RPINR30 (外设引脚选择输入寄存器 30)	159
RPINR31 (外设引脚选择输入寄存器 31)	160
RPINR32 (外设引脚选择输入寄存器 32)	161
RPINR33 (外设引脚选择输入寄存器 33)	162
RPINR34 (外设引脚选择输入寄存器 34)	163
RPINR3 (外设引脚选择输入寄存器 3)	152
RPINR7 (外设引脚选择输入寄存器 7)	153
RPOR0 (外设引脚选择输出寄存器 0)	163
RPOR10 (外设引脚选择输出寄存器 10)	168
RPOR11 (外设引脚选择输出寄存器 11)	169
RPOR12 (外设引脚选择输出寄存器 12)	169
RPOR13 (外设引脚选择输出寄存器 13)	170
RPOR14 (外设引脚选择输出寄存器 14)	170
RPOR16 (外设引脚选择输出寄存器 16)	171
RPOR17 (外设引脚选择输出寄存器 17)	171
RPOR1 (外设引脚选择输出寄存器 1)	164
RPOR2 (外设引脚选择输出寄存器 2)	164
RPOR3 (外设引脚选择输出寄存器 3)	165
RPOR4 (外设引脚选择输出寄存器 4)	165
RPOR5 (外设引脚选择输出寄存器 5)	166
RPOR6 (外设引脚选择输出寄存器 6)	166
RPOR7 (外设引脚选择输出寄存器 7)	167
RPOR8 (外设引脚选择输出寄存器 8)	167
RPOR9 (外设引脚选择输出寄存器 9)	168

SEVTCMP (PWM 特殊事件比较寄存器)	192
SPIxCON1 (SPIx 控制寄存器 1)	209
SPIxCON2 (SPIx 控制寄存器 2)	211
SPIxSTAT (SPIx 状态和控制)	208
SR (CPU 状态)	20, 90
STRIGx (PWMx 辅助触发器比较值寄存器)	204
T1CON (Timer1 控制)	174
TRGCONx (PWMx 触发器控制寄存器)	199
TRIGx (PWMx 主触发器比较值寄存器)	204
TxCON (B 类定时器时基控制)	178
TyCON (C 类定时器时基控制)	179
UxMODE (UARTx 模式)	222
UxSTA (UARTx 状态和控制寄存器)	224
JTAG 边界扫描接口	253
JTAG 接口	258
基于指令的节能模式	135
空闲	136
休眠	135
交流特性	282
负载条件	282
内部 RC 精度	284
节能特性	135
时钟频率和切换	135

K

开发支持	269
看门狗超时复位 (WDTR)	83
看门狗定时器 (WDT)	253, 257
编程注意事项	257
勘误表	12
客户通知服务	327
客户支持	327
框图	
16 位 Timer1 模块	173
DSP 引擎	24
复位系统	77
高速 PWM 的简化概念	188
共用端口结构	143
看门狗定时器 (WDT)	257
PLL	127
片上稳压器的连接	256
器件时钟	125
SPI	207
输出比较	183
输入捕捉	181
Timer2/3 (32 位)	177
Timer2 (16 位)	175
UART	221

L

灵活的配置	253
漏极开路配置	144

M

Microchip 因特网网站	327
MPLAB ASM30 汇编器、链接器和库管理器	270
MPLAB ICD 2 在线调试器	271
MPLAB ICE 2000 高性能通用在线仿真器	271
MPLAB PM3 器件编程器	271
MPLAB REAL ICE 在线仿真器系统	271
MPLAB 集成开发环境软件	269
MPLINK 目标链接器 /MPLIB 目标库管理器	270
模寻址	63
操作示例	63
起始地址和结束地址	63
W 地址寄存器选择	63

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

应用	64
N	
内部 RC 振荡器 与 WDT 配合使用	257
P	
PICSTART 2 开发编程器	272
PICSTART Plus 开发编程器	272
PLL 框图	127
配置寄存器映射	253
配置模拟端口引脚	145
配置位	253
R	
软件堆栈指针, 帧指针 CALL 堆栈帧	61
软件模拟器 (MPLAB SIM)	270
软件 RESET 指令 (SWR)	83
S	
闪存程序存储器	71
编程算法	75
表指令	71
操作	72
控制寄存器	72
RTSP 工作原理	72
上电复位 (POR)	82
时序规范	
10 位 A/D 转换要求	303
复位、看门狗定时器、振荡器起振定时器、上电延时 定时器和欠压复位要求	287
高速 PWM 需求	292
I2Cx 总线数据要求 (从模式)	301
I2Cx 总线数据要求 (主模式)	299
简单 OC/PWM 模式要求	291
PLL 时钟	284
SPIx 从模式 (CKE = 0) 要求	295
SPIx 从模式 (CKE = 1) 要求	297
SPIx 主模式 (CKE = 0) 要求	293
SPIx 主模式 (CKE = 1) 要求	294
输出比较要求	290
Timer2 外部时钟要求	289
Timer3 外部时钟要求	289
Timer1 外部时钟要求	288
时序图	
复位、看门狗定时器、振荡器起振定时器和上电延时 定时器	286
高速 PWM	292
高速 PWM 故障	292
I/O	285
I2Cx 总线启动位 / 停止位 (从模式)	300
I2Cx 总线启动位 / 停止位 (主模式)	298
I2Cx 总线数据 (从模式)	300
I2Cx 总线数据 (主模式)	298
每路输入的 A/D 转换	303
欠压情形	83
SPIx 从模式 (CKE = 0)	295
SPIx 从模式 (CKE = 1)	296
SPIx 主模式 (CKE = 0)	293
SPIx 主模式 (CKE = 1)	294
输出比较 / PWM	291
输出比较 (OCx)	290
输入捕捉 (CAPx)	290
Timer1、Timer2 和 Timer3 外部时钟	288
外部时钟	283
时序要求	

I/O	285
输入捕捉	290
外部时钟	283
使用 RCON 状态位	84
时钟切换	134
过程	134
使能	134
输出比较	183
数据地址空间	31
带 1 KB RAM 的 dsPIC33FJ06GS202 器件的存储器 映射	33
带 2 KB RAM 的 dsPIC33FJ16GS402/404/502/504 器件的存储器映射	34
带 256 KB RAM 的 dsPIC33FJ06GS101/102 器件的 存储器映射	32
对齐	31
宽度	31
Near 数据空间	31
软件堆栈	61
数据累加器和加法器 / 减法器	25
回写	26
舍入逻辑	26
数据空间写饱和	27
溢出和饱和	25
输入捕捉	181
寄存器	182
输入电平变化通知	145
算术逻辑单元 (ALU)	23
T	
Timer2/3	175
Timer1	173
桶形移位寄存器	27
通用异步收发器 (UART)	221
W	
WWW 地址	327
WWW, 在线支持	12
外设模块禁止 (PMD)	136
位反转寻址	64
示例	65
实现	64
序列表 (16 项)	65
温度和电压规范	
AC	282
稳压器 (片上)	256
Y	
引脚输入 / 输出说明 (表)	15
因特网地址	327
Z	
在节能指令执行期间的中断	136
在线串行编程 (ICSP)	253, 258
在线调试器	258
在线仿真	253
振荡器配置	125
指令集	
概述	264
汇总	261
指令寻址模式	61
MAC 指令	62
MCU 指令	61
其他指令	62
文件寄存器指令	61
支持的基本模式	62
传送指令和累加器指令	62

直流特性 274

 程序存储器 281

 打盹电流 (IDOZE) 278

 掉电电流 (IPD) 278

 工作电流 (IDD) 276

 I/O 引脚输出规范 280

 I/O 引脚输入规范 279

 空闲电流 (IIDL) 277

 温度和电压规范 275

中断控制和状态寄存器 89

 IECx 89

 IFSx 89

 INTCON1 89

 INTCON2 89

 IPCx 89

中断设置过程 123

 初始化 123

 陷阱服务程序 123

 中断服务程序 123

 中断禁止 123

中断向量表 (IVT) 85

MICROCHIP 网站

Microchip 网站 (www.microchip.com) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的因特网浏览器即可访问。网站提供以下信息:

- **产品支持**——数据手册和勘误表、应用笔记和样本程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及存档软件
- **一般技术支持**——常见问题 (FAQ)、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时, 收到电子邮件通知。

欲注册, 请登录 Microchip 网站 www.microchip.com, 点击“变更通知客户 (Customer Change Notification)”服务后按照注册说明完成注册。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助:

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

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。

dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04

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: dsPIC33FJ06GS101/X02 和 dsPIC33FJ16GSX02/X04 文献编号: DS70318B_CN

问题

1. 本文档中哪些部分最有特色?

2. 本文档是否满足了您的软硬件开发要求? 如何满足的?

3. 您认为本文档的组织结构便于理解吗? 如果不便于理解, 那么问题何在?

4. 您认为本文档应该添加哪些内容以改善其结构和主题?

5. 您认为本文档中可以删减哪些内容, 而又不会影响整体使用效果?

6. 本文档中是否存在错误或误导信息? 如果存在, 请指出是什么信息及其具体页数。

7. 您认为本文档还有哪些方面有待改进?

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

		dsPIC 33 FJ 06 GS1 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) dsPIC33FJ06GS102-E/SP:
SMPS dsPIC33F, 6 KB 程序存储器, 28 引脚, 扩展级温度, 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:	GS1	=	开关电源 (SMPS) 系列
	GS2	=	开关电源 (SMPS) 系列
	GS4	=	开关电源 (SMPS) 系列
	GS5	=	开关电源 (SMPS) 系列
引脚数:	01	=	18 引脚
	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:	SO	=	塑封小外形封装——宽条, 7.5 mm 主体 (SOIC)
	SP	=	窄型塑封双列直插式封装——300 mil 主体 (SPDIP)
	ML	=	塑封正方无脚封装——8x8 mm 主体 (QFN)
	ML	=	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)

全球销售及服务中心

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://support.microchip.com>
网址: www.microchip.com

亚特兰大 **Atlanta**

Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

波士顿 **Boston**

Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 **Chicago**

Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

达拉斯 **Dallas**

Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 **Detroit**

Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

科科莫 **Kokomo**

Kokomo, IN
Tel: 1-765-864-8360
Fax: 1-765-864-8387

洛杉矶 **Los Angeles**

Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

圣克拉拉 **Santa Clara**

Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

加拿大多伦多 **Toronto**

Mississauga, Ontario,
Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 **Asia Pacific Office**

Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 北京

Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

中国 - 成都

Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 香港特别行政区

Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 南京

Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛

Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海

Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳

Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳

Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

中国 - 武汉

Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 厦门

Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 西安

Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 珠海

Tel: 86-756-321-0040
Fax: 86-756-321-0049

台湾地区 - 高雄

Tel: 886-7-536-4818
Fax: 886-7-536-4803

台湾地区 - 台北

Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

台湾地区 - 新竹

Tel: 886-3-572-9526
Fax: 886-3-572-6459

亚太地区

澳大利亚 **Australia - Sydney**

Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 **India - Bangalore**

Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

印度 **India - New Delhi**

Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 **India - Pune**

Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

日本 **Japan - Yokohama**

Tel: 81-45-471-6166
Fax: 81-45-471-6122

韩国 **Korea - Daegu**

Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 **Korea - Seoul**

Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 **Malaysia - Kuala Lumpur**

Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 **Malaysia - Penang**

Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 **Philippines - Manila**

Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 **Singapore**

Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 **Thailand - Bangkok**

Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 **Austria - Wels**

Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 **Denmark-Copenhagen**

Tel: 45-4450-2828
Fax: 45-4485-2829

法国 **France - Paris**

Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 **Germany - Munich**

Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 **Italy - Milan**

Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 **Netherlands - Drunen**

Tel: 31-416-690399
Fax: 31-416-690340

西班牙 **Spain - Madrid**

Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

英国 **UK - Wokingham**

Tel: 44-118-921-5869
Fax: 44-118-921-5820