



dsPIC30F1010/202X

数据手册

28/44 引脚高性能
开关电源
数字信号控制器

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

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

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

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

商标

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

AmpLab、FilterLab、Linear Active Thermistor、Migratable Memory、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、FlexROM、fuzzyLAB、In-Circuit Serial Programming、ICSP、ICEPIC、Mindi、MiWi、MPASM、MPLAB Certified 徽标、MPLIB、MPLINK、PICKit、PICDEM、PICDEM.net、PICLAB、PICtail、PowerCal、PowerInfo、PowerMate、PowerTool、REAL ICE、rLAB、Select Mode、Smart Serial、SmartTel、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。

© 2007, 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 and 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。

28/44 引脚 dsPIC30F1010/202X 增强型闪存 SMPS 16 位数字信号控制器

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。欲知有关器件指令集和编程的更多详细信息，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

高性能改进的 RISC CPU

- 改进型哈佛结构
- C 编译器优化的指令集架构
- 83 条具有灵活寻址模式的基本指令
- 24 宽指令，16 位宽数据总线
- 12 KB 片上闪存程序存储空间
- 512 字节片上数据 RAM
- 16 x 16 位工作寄存器阵列
- 工作速度最高为 30 MIPS
 - 两个内部 RC
 - 9.7 和 14.55 MHz ($\pm 1\%$) (工业级温度)
 - 6.4 和 9.7 MHz ($\pm 1\%$) (扩展级温度)
 - 带有 480 MHz VCO 的 32X PLL
 - PLL 输入的变化范围为 $\pm 3\%$
 - 外部 EC 时钟 6.0 至 14.55 MHz
 - HS 晶振模式 6.0 至 14.55 MHz
- 32 个中断源
- 3 个外部中断源
- 每个中断具有 8 个可由用户选择的优先级
- 4 个处理器异常和软件陷阱

DSP 引擎特性：

- 模和位反转寻址模式
- 两个具有可选饱和逻辑的 40 位累加器
- 17 位 x 17 位单周期硬件小数 / 整数乘法器
- 单周期乘加 (MAC) 操作
- 40 级桶形移位寄存器
- 双数据取操作

外设特性：

- 高灌 / 拉电流 I/O 引脚：25 mA/25 mA
- 3 个 16 位定时器 / 计数器；可选择将 16 位定时器配对组成 32 位定时器模块
- 一个 16 位捕捉输入功能引脚
- 两个 16 位比较 / PWM 输出功能引脚
 - 可提供双比较模式
- 3 线 SPI 模块（支持 4 种帧模式）
- I²C™ 模块支持多主器件 / 从模式和 7 位 / 10 位寻址
- UART 模块：
 - 支持 RS-232、RS-485 和 LIN 1.2
 - 支持带有片上硬件编解码器的 IrDA®
 - 起始位自动唤醒
 - 自动波特率检测
 - 4 级 FIFO 缓冲区

电源 PWM 模块特性：

- 4 个具有 8 路输出的 PWM 发生器
- 每个 PWM 发生器具有独立的时基和占空比
- 当工作在 30 MIPS 时占空比分辨率为 1.1 ns
- 每个 PWM 发生器具有独立的死区时间：
 - 工作在 30 MIPS 时死区时间分辨率为 4.2 ns
 - 上升和下降沿的死区时间
- 相移分辨率为 4.2 ns @ 30 MIPS
- 频率分辨率为 8.4 ns @ 30 MIPS
- 支持以下 PWM 模式：
 - 互补
 - 推挽
 - 多相
 - 变相
 - 电流复位
 - 电流限制
- 独立电流限制和故障引脚
- 输出改写控制
- 特殊事件触发信号
- 由 PWM 产生的 ADC 触发信号

dsPIC30F1010/202X

模拟特性:

ADC

- 10 位分辨率
- 2000 Ksps 转换速率
- 最多 12 个输入通道
- “转换组对”允许使用一个触发信号同时转换两个输入（即电流和电压）
- PWM 控制环：
 - 最多可使用 6 对转换
 - 每对转换具有最多 4 个 PWM 和 7 个其他可选择的触发源
- 中断硬件支持每秒中断最多 1M 次。

比较器

- 4 个模拟比较器：
 - 20 ns 响应时间
 - 10 位 DAC 参考电压发生器
 - 可编程输出极性
 - 可选择的输入源
 - ADC 采样和转换触发功能
- PWM 模块接口
 - PWM 占空比控制
 - PWM 周期控制
 - PWM 故障检测
- 特殊事件触发信号
- 由 PWM 产生的 ADC 触发信号

数字信号控制器的特性:

- 增强型闪存程序存储器：
 - 对于工业级温度范围，最少擦写次数 1 万次，典型擦写次数 10 万次。
- 软件控制下，可自行再编程
- 上电复位（Power-on Reset, POR）、上电延时定时器（Power-up Timer, PWRT）以及振荡器起振定时器（Oscillator Start-up Timer, OST）
- 灵活的看门狗定时器（Watchdog Timer, WDT），片内带有低功耗 RC 振荡器，以便可靠地工作
- 故障保护时钟监视器操作
- 检测时钟故障，可切换至片上低功耗 RC 振荡器
- 可编程代码保护
- 在线串行编程（In-Circuit Serial Programming™, ICSP™）
- 可选择的功耗管理模式
 - 休眠、空闲和备用时钟模式

CMOS 技术:

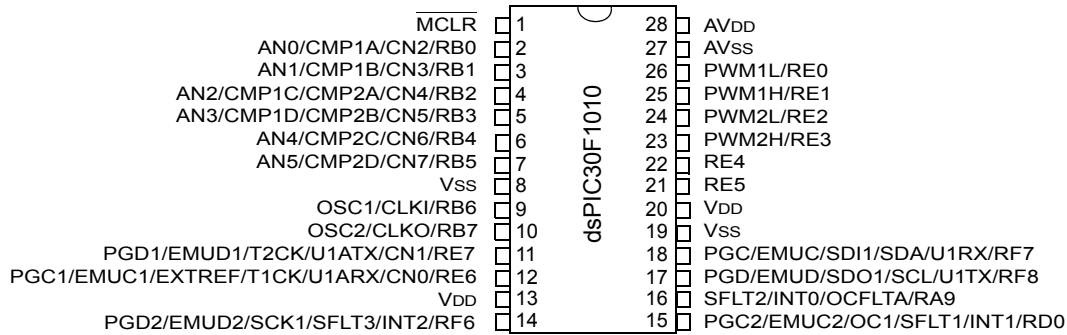
- 低功耗高速闪存技术
- 3.3V 和 5.0V 工作（± 10%）
- 工业级和扩展级温度范围
- 低功耗

dsPIC30F 开关电源系列

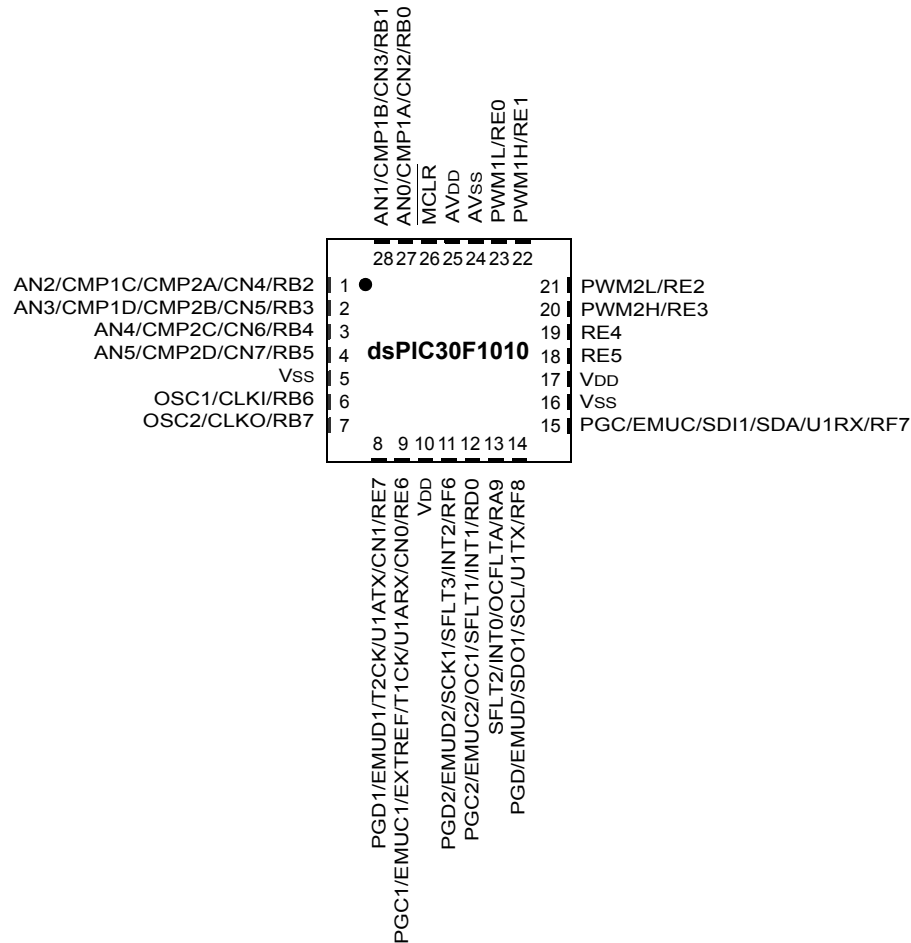
产品	引脚	封装	程序存储器 (字节)	数据 SRAM (字节)	定时器	捕捉	比较	UART	SPI	I ² C™	PWM	ADC	S 和 H	A/D 输入	模拟比较器	GPIO
dsPIC30F1010	28	SDIP	6K	256	2	0	1	1	1	1	2x2	1	3	6 个通道	2	21
dsPIC30F1010	28	SOIC	6K	256	2	0	1	1	1	1	2x2	1	3	6 个通道	2	21
dsPIC30F1010	28	QFN-S	6K	256	2	0	1	1	1	1	2x2	1	3	6 个通道	2	21
dsPIC30F2020	28	SDIP	12K	512	3	1	2	1	1	1	4x2	1	5	8 个通道	4	21
dsPIC30F2020	28	SOIC	12K	512	3	1	2	1	1	1	4x2	1	5	8 个通道	4	21
dsPIC30F2020	28	QFN-S	12K	512	3	1	2	1	1	1	4x2	1	5	8 个通道	4	21
dsPIC30F2023	44	QFN	12K	512	3	1	2	1	1	1	4x2	1	5	12 个通道	4	35
dsPIC30F2023	44	TQFP	12K	512	3	1	2	1	1	1	4x2	1	5	12 个通道	4	35

引脚图

28 引脚 SDIP 和 SOIC

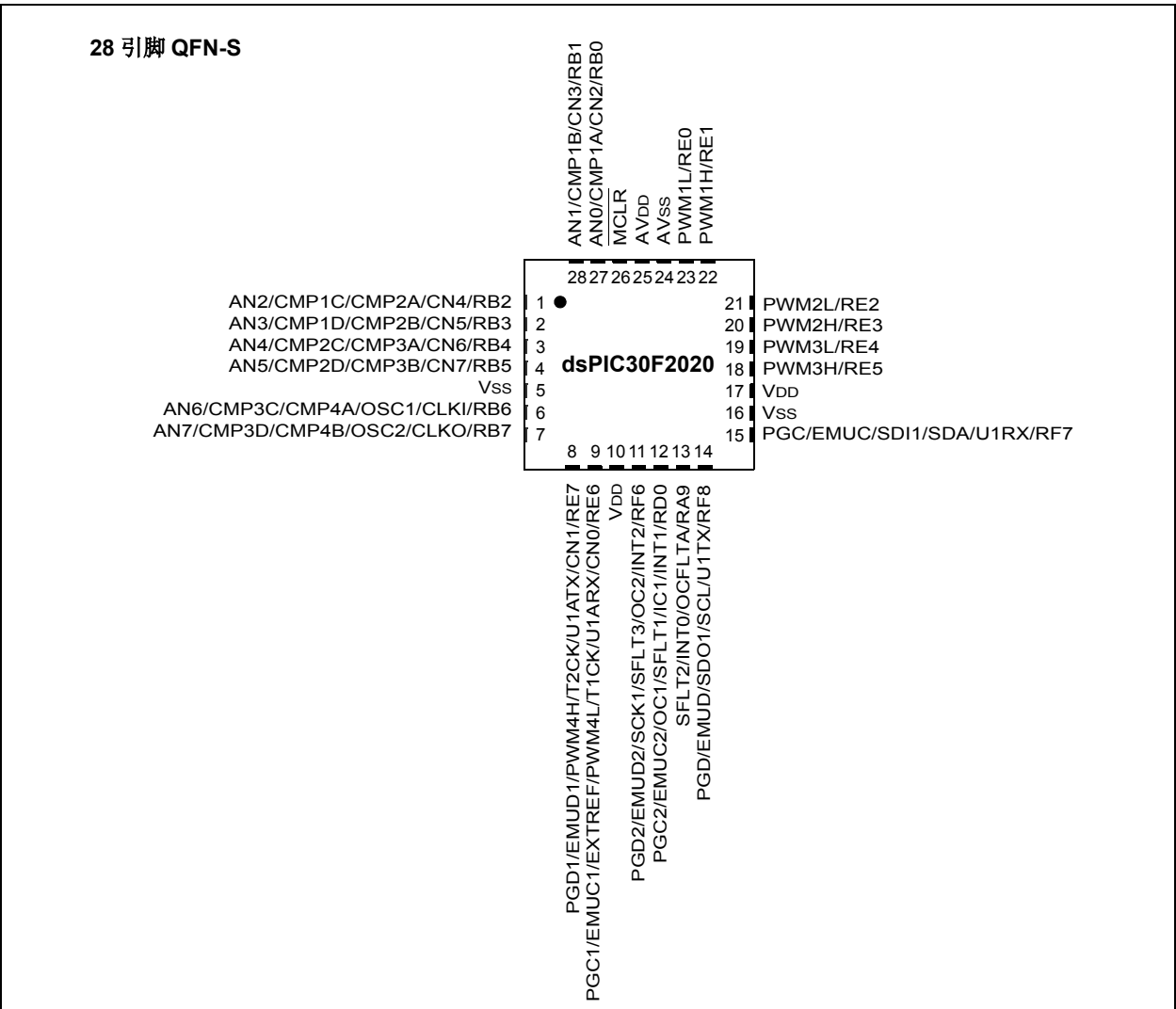
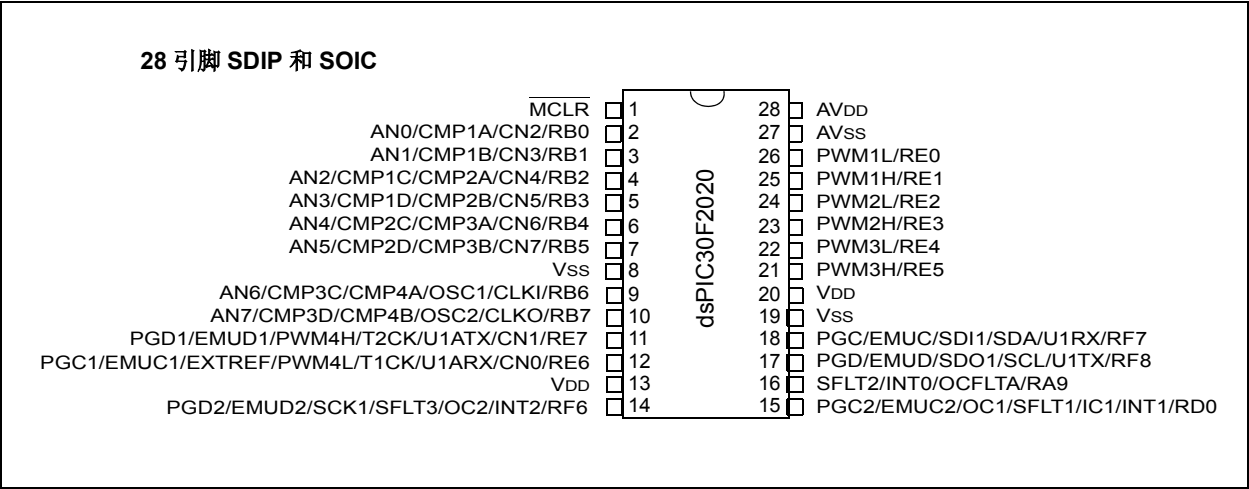


28 引脚 QFN-S



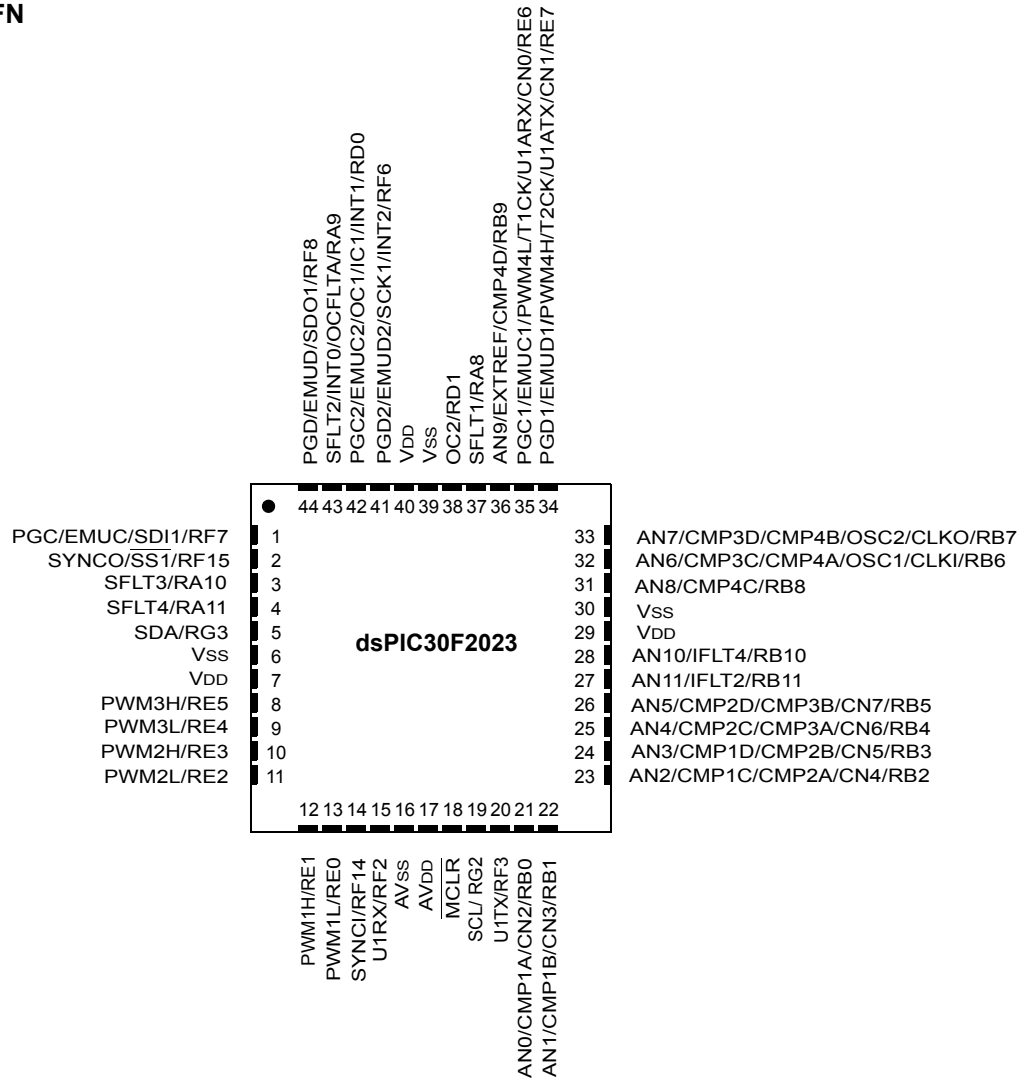
dsPIC30F1010/202X

引脚图



引脚图

44 引脚 QFN



目录

1.0	器件概述	9
2.0	CPU 架构综述	19
3.0	存储器构成	29
4.0	地址发生器单元	41
5.0	中断	47
6.0	I/O 端口	77
7.0	闪存程序存储器	81
8.0	Timer1 模块	87
9.0	Timer2/3 模块	91
10.0	输入捕捉模块	97
11.0	输出比较模块	101
12.0	电源 PWM	107
13.0	串行外设接口 (SPI)	145
14.0	I ² C™ 模块	153
15.0	通用异步收发器 (UART) 模块	161
16.0	10 位 2 Msps 模数转换器 (ADC) 模块	169
17.0	SMPS 比较器模块	191
18.0	系统集成	197
19.0	指令集汇总	219
20.0	开发支持	227
21.0	电气特性	231
22.0	封装标识信息	267

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 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。有关器件指令集和编程的更多信息，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

本文档包含了 dsPIC30F1010/202X SMPS 器件的特定信息。这些器件在高性能 16 位单片机 (MCU) 架构中融合了大量的数字信号处理器 (Digital Signal Processor, DSP) 功能，如下列框图所示。图 1-1 和表 1-1 所示为 dsPIC30F1010 SMPS 器件。图 1-2 和表 1-2 所示为 dsPIC30F2020 器件。图 1-3 和表 1-3 所示为 dsPIC30F2023 SMPS 器件。

dsPIC30F1010/202X

图 1-1: dsPIC30F1010 框图

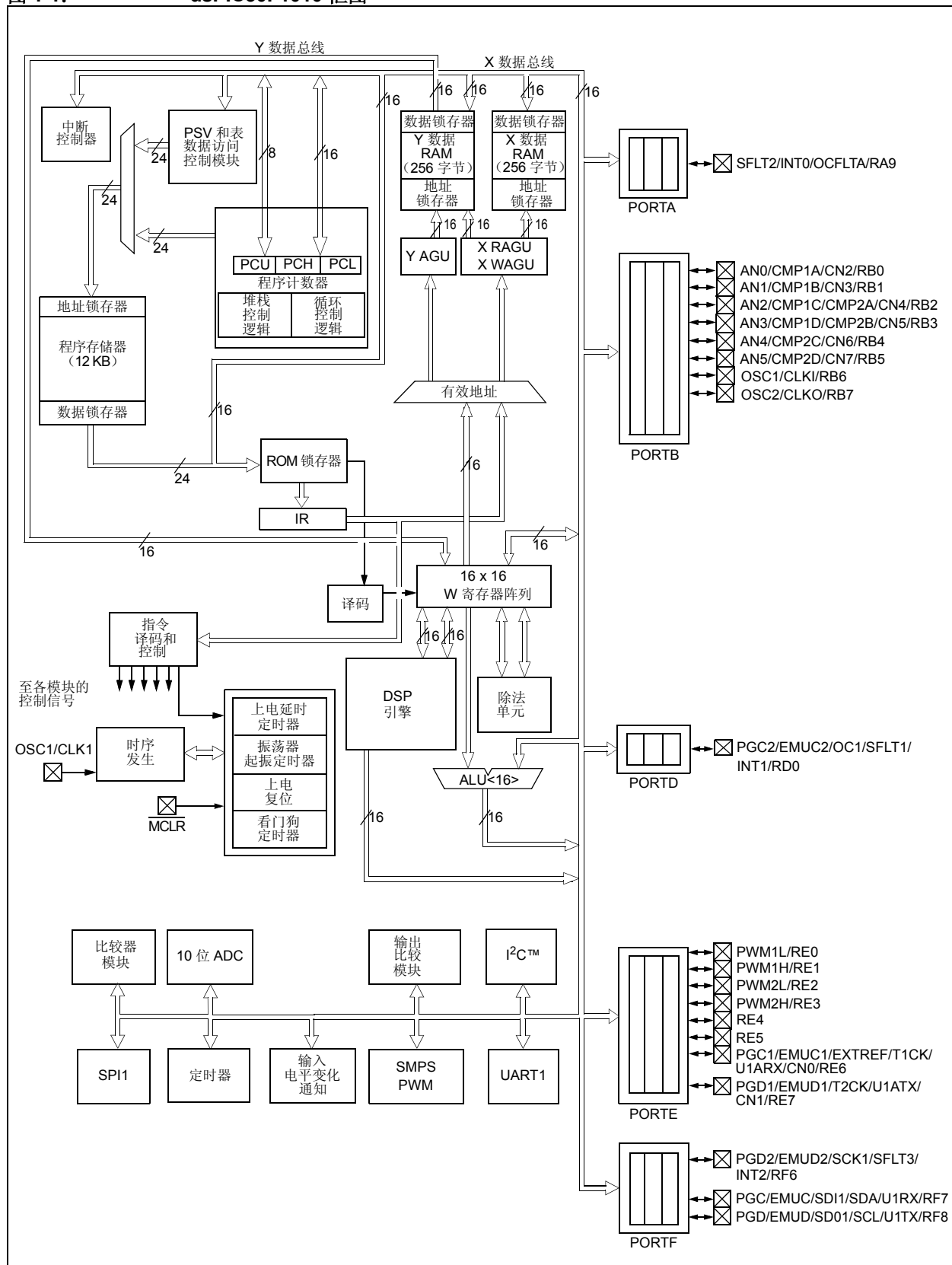


表 1-1 对 dsPIC30F1010 的 I/O 引脚配置和端口引脚的复用功能进行了简要描述。端口引脚可具有多种功能。当发生复用时，外设模块的功能要求可强制替代端口引脚的数据方向。

表 1-1: dsPIC30F1010 I/O 引脚说明

引脚名称	引脚类型	缓冲器类型	说明
AN0-AN5	I	模拟	模拟输入通道。
AVDD	P	P	模拟模块正电源。
AVSS	P	P	模拟模块的参考地。
CLKI CLKO	I O	ST/CMOS —	外部时钟源输入。总是与 OSC1 引脚功能相关联。 振荡器晶振输出。在晶振模式下，连接到晶振或谐振器。在 RC 和 EC 模式下，可选择作为 CLKO。总是与 OSC2 引脚功能相关联。
EMUD EMUC EMUD1 EMUC1 EMUD2 EMUC2	I/O I/O I/O I/O I/O I/O	ST ST ST ST ST ST	ICD 主通信通道数据输入 / 输出引脚。 ICD 主通信通道时钟输入 / 输出引脚。 ICD 第二通信通道数据输入 / 输出引脚。 ICD 第二通信通道时钟输入 / 输出引脚。 ICD 第三通信通道数据输入 / 输出引脚。 ICD 第三通信通道时钟输入 / 输出引脚。
INT0 INT1 INT2	I I I	ST ST ST	外部中断 0。 外部中断 1。 外部中断 2。
SFLT1 SFLT2 SFLT3 PWM1L PWM1H PWM2L PWM2H	I I I O O O O	ST ST ST — — — —	共用故障引脚 1 共用故障引脚 2 共用故障引脚 3 PWM 1 低电平输出 PWM 1 高电平输出 PWM 2 低电平输出 PWM 2 高电平输出
MCLR	I/P	ST	主复位输入或可编程电压输入，此引脚为低电平有效的器件复位输入端。
OC1	O	—	比较输出。
OCFLTA	I	ST	输出比较故障引脚
OSC1 OSC2	I I/O	CMOS —	振荡器晶振输入。 振荡器晶振输出。在晶振模式下，连接到晶振或谐振器。在 RC 和 EC 模式下，可选择作为 CLKO。
PGD PGC PGD1 PGC1 PGD2 PGC2	I/O I I/O I I/O I	ST ST ST ST ST ST	在线串行编程（ICSP™）数据输入 / 输出引脚。 在线串行编程时钟输入引脚。 在线串行编程数据输入 / 输出引脚 1。 在线串行编程时钟输入引脚 1。 在线串行编程数据输入 / 输出引脚 2。 在线串行编程时钟输入引脚 2。
RB0-RB7	I/O	ST	PORTB 是一个双向 I/O 口。
RA9	I/O	ST	PORTA 是一个双向 I/O 口。
RD0	I/O	ST	PORTD 是一个双向 I/O 口。

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

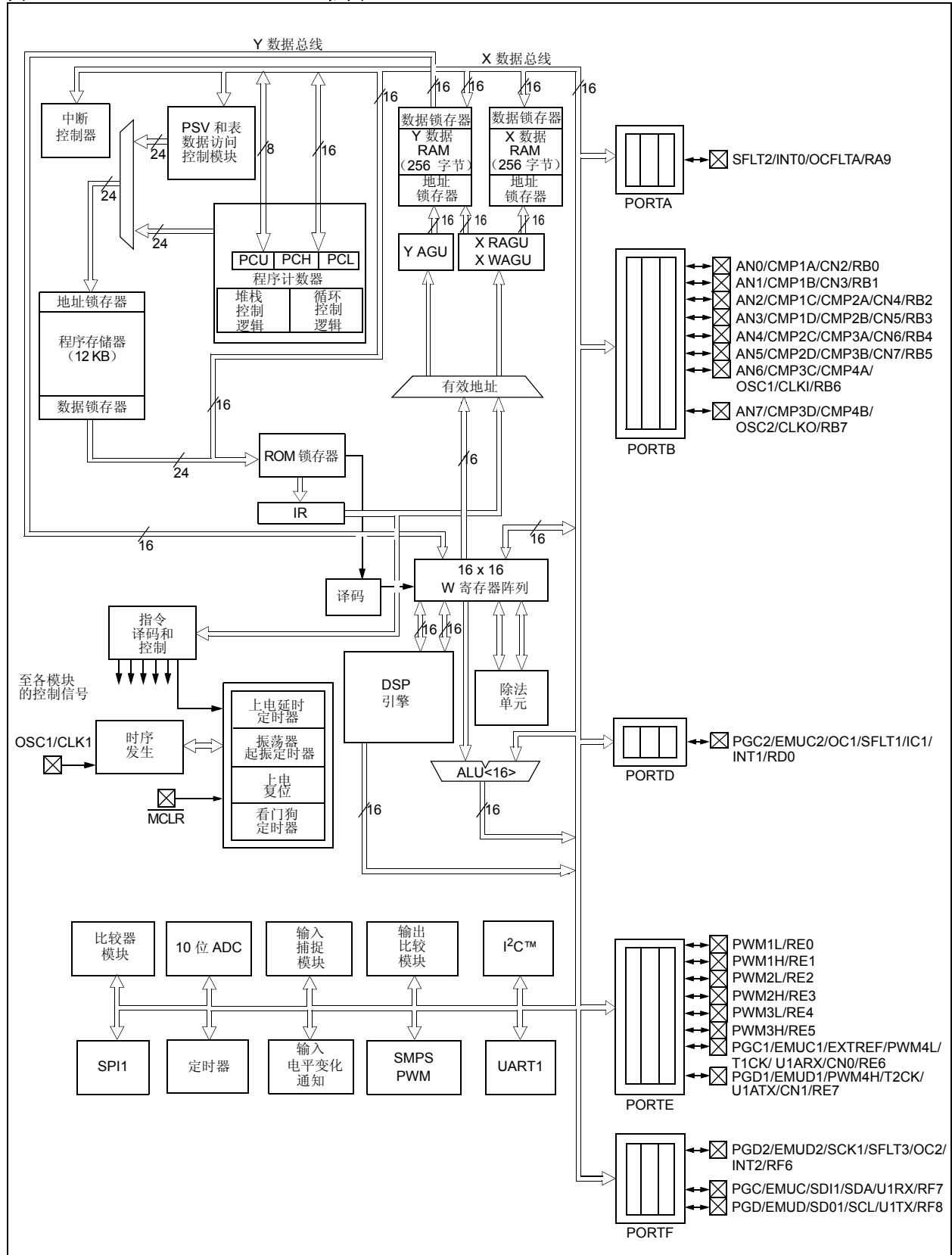
dsPIC30F1010/202X

表 1-1: dsPIC30F1010 I/O 引脚说明 (续)

引脚名称	引脚类型	缓冲器类型	说明
RE0-RE7	I/O	ST	PORTE 是一个双向 I/O 口。
RF6, RF7, RF8	I/O	ST	PORTF 是一个双向 I/O 口。
SCK1	I/O	ST	SPI1 的同步串行时钟输入 / 输出。
SDI1	I	ST	SPI1 数据输入。
SDO1	O	—	SPI1 数据输出。
SCL	I/O	ST	用于 I ² C™ 的同步串行时钟输入 / 输出。
SDA	I/O	ST	用于 I ² C™ 的同步串行数据输入 / 输出。
T1CK	I	ST	Timer1 外部时钟输入。
T2CK	I	ST	Timer2 外部时钟输入。
U1RX	I	ST	UART1 接收。
U1TX	O	—	UART1 发送。
U1ARX	I	ST	备用 UART1 接收。
U1ATX	O	—	备用 UART1 发送。
CMP1A	I	模拟	比较器 1 通道 A
CMP1B	I	模拟	比较器 1 通道 B
CMP1C	I	模拟	比较器 1 通道 C
CMP1D	I	模拟	比较器 1 通道 D
CMP2A	I	模拟	比较器 2 通道 A
CMP2B	I	模拟	比较器 2 通道 B
CMP2C	I	模拟	比较器 2 通道 C
CMP2D	I	模拟	比较器 2 通道 D
CN0-CN7	I	ST	输入电平变化通知输入 可通过软件编程将所有输入设置为内部弱上拉。
VDD	P	—	逻辑和 I/O 引脚的正电源。
VSS	P	—	逻辑和 I/O 引脚的参考地。
EXTREF	I	模拟	至比较器 DAC 的外部参考电压。

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

图 1-2: dsPIC30F2020 框图



dsPIC30F1010/202X

表 1-2 对 dsPIC30F2020 的 I/O 引脚配置和端口引脚的复用功能进行了简要描述。端口引脚可具有多种功能。当发生复用时，外设模块的功能要求可强制替代端口引脚的数据方向。

表 1-2: dsPIC30F2020 I/O 引脚说明

引脚名称	引脚类型	缓冲器类型	说明
AN0-AN7	I	模拟	模拟输入通道。
AVDD	P	P	模拟模块正电源。
AVss	P	P	模拟模块的参考地。
CLKI CLKO	I O	ST/CMOS —	外部时钟源输入。总是与 OSC1 引脚功能相关联。 振荡器晶振输出。在晶振模式下，连接到晶振或谐振器。在 RC 和 EC 模式下，可选择作为 CLKO。总是与 OSC2 引脚功能相关联。
EMUD EMUC EMUD1 EMUC1 EMUD2 EMUC2	I/O I/O I/O I/O I/O I/O	ST ST ST ST ST ST	ICD 主通信通道数据输入 / 输出引脚。 ICD 主通信通道时钟输入 / 输出引脚。 ICD 第二通信通道数据输入 / 输出引脚。 ICD 第二通信通道时钟输入 / 输出引脚。 ICD 第三通信通道数据输入 / 输出引脚。 ICD 第三通信通道时钟输入 / 输出引脚。
IC1	I	ST	捕捉输入。
INT0 INT1 INT2	I I I	ST ST ST	外部中断 0 外部中断 1 外部中断 2
SFLT1 SFLT2 SFLT3 PWM1L PWM1H PWM2L PWM2H PWM3L PWM3H PWM4L PWM4H	I I I O O O O O O O O O	ST ST ST — — — — — — — — —	共享故障引脚 1 共享故障引脚 2 共享故障引脚 3 PWM 1 低电平输出 PWM 1 高电平输出 PWM 2 低电平输出 PWM 2 高电平输出 PWM 3 低电平输出 PWM 3 高电平输出 PWM 4 低电平输出 PWM 4 高电平输出
MCLR	I/P	ST	主复位输入或可编程电压输入，此引脚为低电平有效的器件复位输入端。
OC1-OC2 OCFLTA	O I	—	比较输出。 输出比较故障引脚。
OSC1 OSC2	I I/O	CMOS —	振荡器晶振输入。 振荡器晶振输出。在晶振模式下，连接到晶振或谐振器。在 RC 和 EC 模式下，可选择作为 CLKO。
PGD PGC PGD1 PGC1 PGD2 PGC2	I/O I I/O I I/O I	ST ST ST ST ST ST	在线串行编程数据输入 / 输出引脚。 在线串行编程时钟输入引脚。 在线串行编程数据输入 / 输出引脚 1。 在线串行编程时钟输入引脚 1。 在线串行编程数据输入 / 输出引脚 2。 在线串行编程时钟输入引脚 2。

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

表 1-2: dsPIC30F2020 I/O 引脚说明 (续)

引脚名称	引脚类型	缓冲器类型	说明
RB0-RB7	I/O	ST	PORTB 是一个双向 I/O 口。
RA9	I/O	ST	PORTA 是一个双向 I/O 口。
RD0	I/O	ST	PORTD 是一个双向 I/O 口。
RE0-RE7	I/O	ST	PORTE 是一个双向 I/O 口。
RF6, RF7, RF8	I/O	ST	PORTF 是一个双向 I/O 口。
SCK1 SDI1 SDO1	I/O I O	ST ST —	SPI1 的同步串行时钟输入 / 输出。 SPI1 数据输入。 SPI1 数据输出。
SCL SDA	I/O I/O	ST ST	用于 I ² C™ 的同步串行时钟输入 / 输出。 用于 I ² C™ 的同步串行数据输入 / 输出。
T1CK T2CK	I I	ST ST	Timer1 外部时钟输入。 Timer2 外部时钟输入。
U1RX U1TX U1ARX U1ATX	I O I O	ST — ST O	UART1 接收。 UART1 发送。 备用 UART1 接收。 备用 UART1 发送。
CMP1A CMP1B CMP1C CMP1D CMP2A CMP2B CMP2C CMP2D CMP3A CMP3B CMP3C CMP3D CMP4A CMP4B	I I I I I I I I I I I I I I	模拟 模拟 模拟 模拟 模拟 模拟 模拟 模拟 模拟 模拟 模拟 模拟 模拟 模拟	比较器 1 通道 A 比较器 1 通道 B 比较器 1 通道 C 比较器 1 通道 D 比较器 2 通道 A 比较器 2 通道 B 比较器 2 通道 C 比较器 2 通道 D 比较器 3 通道 A 比较器 3 通道 B 比较器 3 通道 C 比较器 3 通道 D 比较器 4 通道 A 比较器 4 通道 B
CN0-CN7	I	ST	输入电平变化通知输入 可通过软件编程将所有输入设置为内部弱上拉。
VDD	P	—	逻辑和 I/O 引脚的正电源。
VSS	P	—	逻辑和 I/O 引脚的参考地。
EXTREF	I	模拟	至比较器 DAC 的外部参考电压。

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

图 1-3: dsPIC30F2023 框图

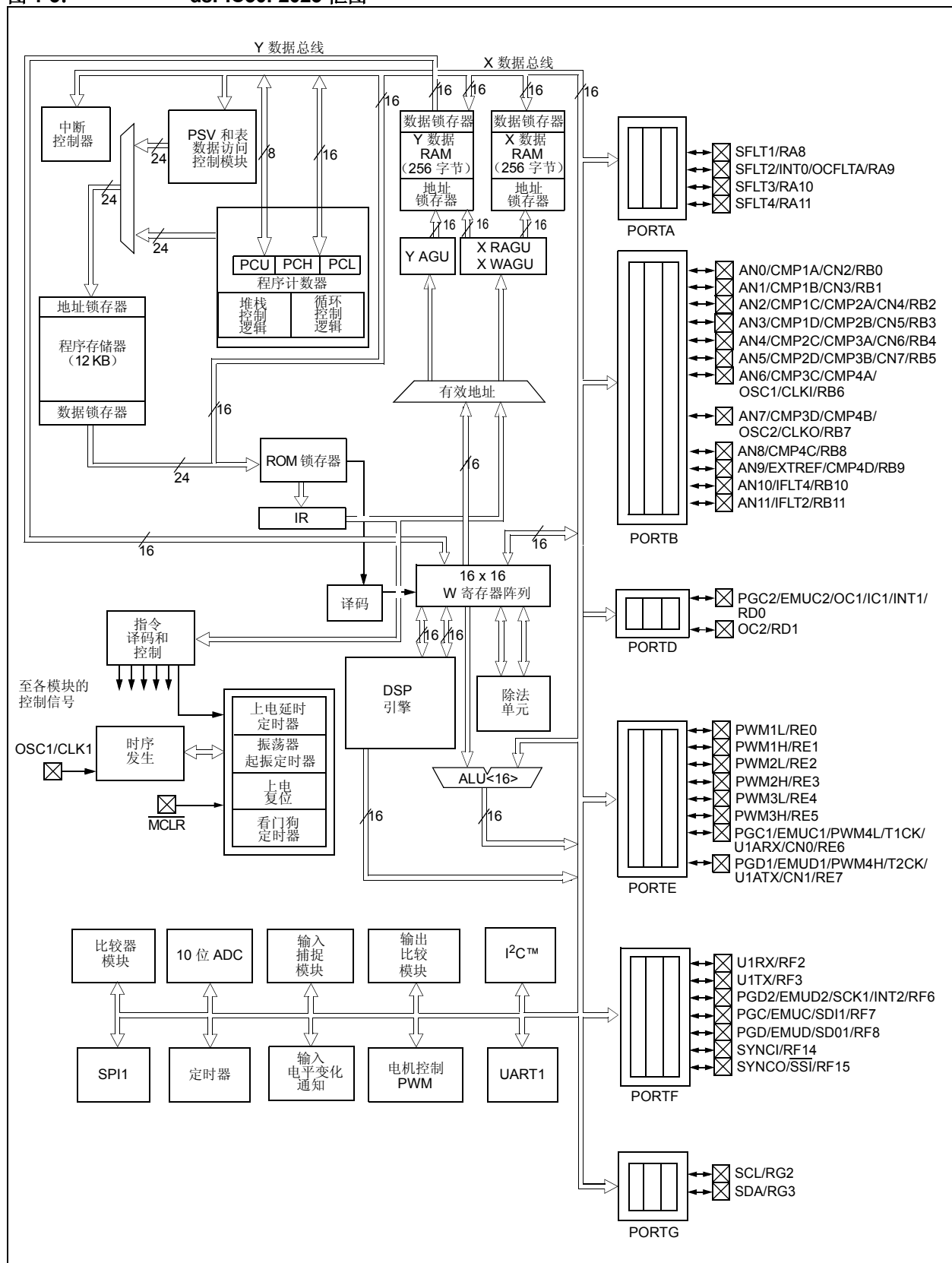


表 1-3 对 dsPIC30F2023 的 I/O 引脚配置和端口引脚的复用功能进行了简要描述。端口引脚可具有多种功能。当发生复用时，外设模块的功能要求可强制替代端口引脚的数据方向。

表 1-3: dsPIC30F2023 I/O 引脚说明

引脚名称	引脚类型	缓冲器类型	说明
AN0-AN11	I	模拟	模拟输入通道。
AVDD	P	P	模拟模块正电源。
AVss	P	P	模拟模块的参考地。
CLKI CLKO	I O	ST/CMOS —	外部时钟源输入。总是与 OSC1 引脚功能相关联。 振荡器晶振输出。在晶振模式下，连接到晶振或谐振器。在 RC 和 EC 模式下，可选择作为 CLKO。总是与 OSC2 引脚功能相关联。
EMUD EMUC EMUD1 EMUC1 EMUD2 EMUC2	I/O I/O I/O I/O I/O I/O	ST ST ST ST ST ST	ICD 主通信通道数据输入 / 输出引脚。 ICD 主通信通道时钟输入 / 输出引脚。 ICD 第二通信通道数据输入 / 输出引脚。 ICD 第二通信通道时钟输入 / 输出引脚。 ICD 第三通信通道数据输入 / 输出引脚。 ICD 第三通信通道时钟输入 / 输出引脚。
IC1	I	ST	捕捉输入。
INT0 INT1 INT2	I I I	ST ST ST	外部中断 0 外部中断 1 外部中断 2
SFLT1 SFLT2 SFLT3 SFLT4 IFLT2 IFLT4 PWM1L PWM1H PWM2L PWM2H PWM3L PWM3H PWM4L PWM4H	I I I I I I O O O O O O O O O	ST ST ST ST ST ST — — — — — — — — —	共享故障引脚 1 共享故障引脚 2 共享故障引脚 3 共享故障引脚 4 独立故障引脚 1 独立故障引脚 2 PWM 1 低电平输出 PWM 1 高电平输出 PWM 2 低电平输出 PWM 2 高电平输出 PWM 3 低电平输出 PWM 3 高电平输出 PWM 4 低电平输出 PWM 4 高电平输出
SYNCO SYNCI	O I	— ST	PWM 同步输出 PWM 同步输入
MCLR	I/P	ST	主复位输入或可编程电压输入，此引脚为低电平有效的器件复位输入端。
OC1-OC2 OCFLTA	O I	— ST	比较输出。 输出比较故障条件。
OSC1 OSC2	I I/O	CMOS —	振荡器晶振输入。 振荡器晶振输出。在晶振模式下，连接到晶振或谐振器。在 RC 和 EC 模式下，可选择作为 CLKO。

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

dsPIC30F1010/202X

表 1-3: dsPIC30F2023 I/O 引脚说明 (续)

引脚名称	引脚类型	缓冲器类型	说明
PGD	I/O	ST	在线串行编程数据输入 / 输出引脚。
PGC	I	ST	在线串行编程时钟输入引脚。
PGD1	I/O	ST	在线串行编程数据输入 / 输出引脚 1。
PGC1	I	ST	在线串行编程时钟输入引脚 1。
PGD2	I/O	ST	在线串行编程数据输入 / 输出引脚 2。
PGC2	I	ST	在线串行编程时钟输入引脚 2。
RA8-RA11	I/O	ST	PORTA 是一个双向 I/O 口。
RB0-RB11	I/O	ST	PORTB 是一个双向 I/O 口。
RD0,RD1	I/O	ST	PORTD 是一个双向 I/O 口。
RE0-RE7	I/O	ST	PORTE 是一个双向 I/O 口。
RF2, RF3, RF6-RF8, RF14, RF15	I/O	ST	PORTF 是一个双向 I/O 口。
RG2, RG3	I/O	ST	PORTG 是一个双向 I/O 口。
SCK1	I/O	ST	SPI1 的同步串行时钟输入 / 输出。
SDI1	I	ST	SPI1 数据输入。
SDO1	O	—	SPI1 数据输出。
SS1	I	ST	SPI1 从动同步。
SCL	I/O	ST	用于 I ² C™ 的同步串行时钟输入 / 输出。
SDA	I/O	ST	用于 I ² C™ 的同步串行数据输入 / 输出。
T1CK	I	ST	Timer1 外部时钟输入。
T2CK	I	ST	Timer2 外部时钟输入。
U1RX	I	ST	UART1 接收。
U1TX	O	—	UART1 发送。
U1ARX	I	ST	备用 UART1 接收。
U1ATX	O	—	备用 UART1 发送。
CMP1A	I	模拟	比较器 1 通道 A
CMP1B	I	模拟	比较器 1 通道 B
CMP1C	I	模拟	比较器 1 通道 C
CMP1D	I	模拟	比较器 1 通道 D
CMP2A	I	模拟	比较器 2 通道 A
CMP2B	I	模拟	比较器 2 通道 B
CMP2C	I	模拟	比较器 2 通道 C
CMP2D	I	模拟	比较器 2 通道 D
CMP3A	I	模拟	比较器 3 通道 A
CMP3B	I	模拟	比较器 3 通道 B
CMP3C	I	模拟	比较器 3 通道 C
CMP3D	I	模拟	比较器 3 通道 D
CMP4A	I	模拟	比较器 4 通道 A
CMP4B	I	模拟	比较器 4 通道 B
CMP4C	I	模拟	比较器 4 通道 C
CMP4D	I	模拟	比较器 4 通道 D
CN0-CN7	I	ST	输入电平变化通知输入 可通过软件编程将所有输入设置为内部弱上拉。
VDD	P	—	逻辑和 I/O 引脚的正电源。
VSS	P	—	逻辑和 I/O 引脚的参考地。
EXTREF	I	Analog	至比较器 DAC 的外部参考电压。

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

2.0 CPU 架构综述

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。有关器件指令集和编程的更多信息，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

2.1 内核概述

内核的指令字长为 24 位。程序计数器 (PC) 为 23 位宽，其最低有效位 (LSb) 始终为 0 (见第 3.1 节“程序地址空间”)，最高有效位 (MSb) 在正常的程序执行期间被忽略，某些专用指令除外。因此，PC 能够寻址最大为 4M 指令字的用户程序空间。使用指令预取机制来帮助维持吞吐量。使用 DO 和 REPEAT 指令，支持无循环计数管理开销的程序循环结构，而 DO 和 REPEAT 这两条指令在任何时候都可被中断。

工作寄存器阵列由 16 个 16 位寄存器组成，每个寄存器均可作为数据、地址或偏移量寄存器。一个工作寄存器 (W15) 用作中断和调用时的软件堆栈指针。

数据空间为 64KB (32K 字)，被分成两块，称为 X 和 Y 数据存储空间。每个存储块有各自独立的地址发生单元 (Address Generation Unit, AGU)。大多数指令只通过 X 存储空间 AGU 进行操作，这样对外界而言数据空间就是单独而统一的。乘法—累加 (Multiply-Accumulate, MAC) 类双源操作数 DSP 指令，通过 X 和 Y 的 AGU 进行操作，这将数据地址空间分成两个部分 (见第 3.2 节“数据地址空间”)。X 和 Y 数据空间边界视具体器件而定，用户不能改变它们。每个数据字由 2 个字节构成，大多数指令可以把数据当作字或字节来进行寻址。

对于存储在程序存储空间中的数据，有两种存取方法：

- 数据存储空间的高 32 KB 可以映射到程序空间的下半部分 (用户空间)，映射的边界可以是任何 16K 程序字边界，边界由 8 位的程序空间可视性页 (Program Space Visibility Page, PSVPAG) 寄存器定义。这使得任何指令都能象存取数据存储空间那样存取程序存储空间，但存取需要一个额外的周期。此外，使用这个方法，只能存取每个指令字的低 16 位。

- 使用任何工作寄存器，通过表读及表写指令，可以对程序空间内的 32K 字页进行线性间接存取。表读和表写指令能够用来存取一个指令字的全部 24 位。

X 和 Y 地址空间都支持无开销循环缓冲区 (模寻址)。模寻址的主要目的是要消除 DSP 算法的循环开销。

X AGU 还支持对目的有效地址进行位反转寻址，对于基 -2 FFT 算法，这极大地简化了输入或输出数据的重新排序。关于模寻址和位反转寻址的详细信息，请参见第 4.0 节“地址发生器单元”。

内核支持固有 (无操作数) 寻址、相对寻址、立即数寻址、存储器直接寻址、寄存器直接寻址、寄存器间接寻址、寄存器偏移量寻址以及立即数偏移量寻址模式。指令与预定义的寻址模式相关联，这取决于其功能需求。

对于大多数指令，内核能够在每个指令周期内执行数据 (或程序数据) 存储器读取、工作寄存器 (数据) 读取、数据存储写入以及程序 (指令) 存储器读取操作。因此，支持 3 操作数指令，允许在单个周期内执行 $C = A + B$ 这样的操作。

内核包含 DSP 引擎，这极大地提高了内核的运算能力和吞吐能力。它具有一个高速 17 位 $\times$ 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。累加器或任何工作寄存器中的数据，可以在单个周期内右移 16 位或左移 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。在把两个 W 寄存器相乘时，MAC 类指令能够从存储器中同时取出两个操作数。为了能够同时取数据操作数，对于 MAC 类指令，数据空间拆分为两块，对所有其他指令数据空间则是线性的。对于 MAC 类指令，这是通过将某些工作寄存器专用于每个地址空间，以透明而灵活的方式实现的。

内核不支持多级指令流水线。不过，为了最大限度地获得可用的执行时间，使用了单级指令预取机制，在指令执行的前一个周期，存取并部分译码指令。除了一些特例外，大部分指令都在单个周期内执行。

内核具有用于陷阱和中断的向量异常处理结构，具有 62 个独立的向量。异常包括最多 8 个陷阱 (其中有 4 个是保留的) 和 54 个中断。根据用户指定的 1 到 7 之间的优先级 (1 为最低优先级，7 为最高优先级)，以及预定义的“自然顺序”，决定每个中断的优先级。陷阱的优先级是固定的，其优先级范围是从 8 到 15。

2.2 编程模型

编程模型如图 2-1 所示，它包括 16 x 16 位工作寄存器（W0 至 W15）、2 x 40 位累加器（ACCA 和 ACCB）、状态寄存器（SR）、数据表页寄存器（TBLPAG）、程序空间可视性页寄存器（PSVPAG）、DO 和 REPEAT 寄存器（DOSTART、DOEND、DCOUNT 和 RCOUNT）以及程序计数器（PC）。工作寄存器可作为数据、地址或偏移量寄存器。全部寄存器都是存储器映射的。W0 用作文件寄存器寻址的 W 寄存器。

这些寄存器中，一些寄存器有与之关联的影子寄存器，如图 2-1 所示。影子寄存器用作临时保存寄存器，事件发生时可在影子寄存器和主寄存器之间传递内容。所有影子寄存器均不能直接存取。寄存器与其影子寄存器之间的数据传递遵守下列规则：

- PUSH.S 和 POP.S
W0、W1、W2、W3 和 SR（仅限 DC、N、OV、Z 和 C 位）与对应的影子寄存器之间进行数据传递。
- DO 指令
循环开始时，DOSTART、DOEND 和 DCOUNT 寄存器的内容压入影子寄存器，在循环结束时其内容从各自的影子寄存器中弹出。

对工作寄存器进行字节操作时，只影响目标寄存器的最低有效字节（LSB）。不过，存储器映射工作寄存器的一个好处是，可以通过字节宽数据存储空间存取来处理最低和最高有效字节（MSB）。

2.2.1 软件堆栈指针 / 帧指针

dsPIC® DSC 器件具备一个软件堆栈。W15 被指定用作软件堆栈指针（Stack Pointer, SP），异常处理以及子程序调用与返回将自动修改 W15。但是，W15 可以被任何指令所引用，引用方式与引用所有其他 W 寄存器相同。这简化了堆栈指针的读、写和处理（例如，创建堆栈帧）。

注： 为了防止出现不对齐的堆栈访问，W15<0> 始终保持为零。

复位时 W15 初始化为 0x0800。在初始化过程中，用户可以重新设定 SP，使它指向数据空间内的任何单元。

W14 被指定用作 LNK 和 ULNK 指令中所定义的堆栈帧指针。但是，W14 可以被任何指令所引用，引用方式与引用所有其他 W 寄存器相同。

2.2.2 状态寄存器

dsPIC DSC 内核有一个 16 位状态寄存器（SR），其 LSB 称为 SR 低字节（SRL），其 MSB 称为 SR 高字节（SRH），SR 寄存器组成见图 2-1。

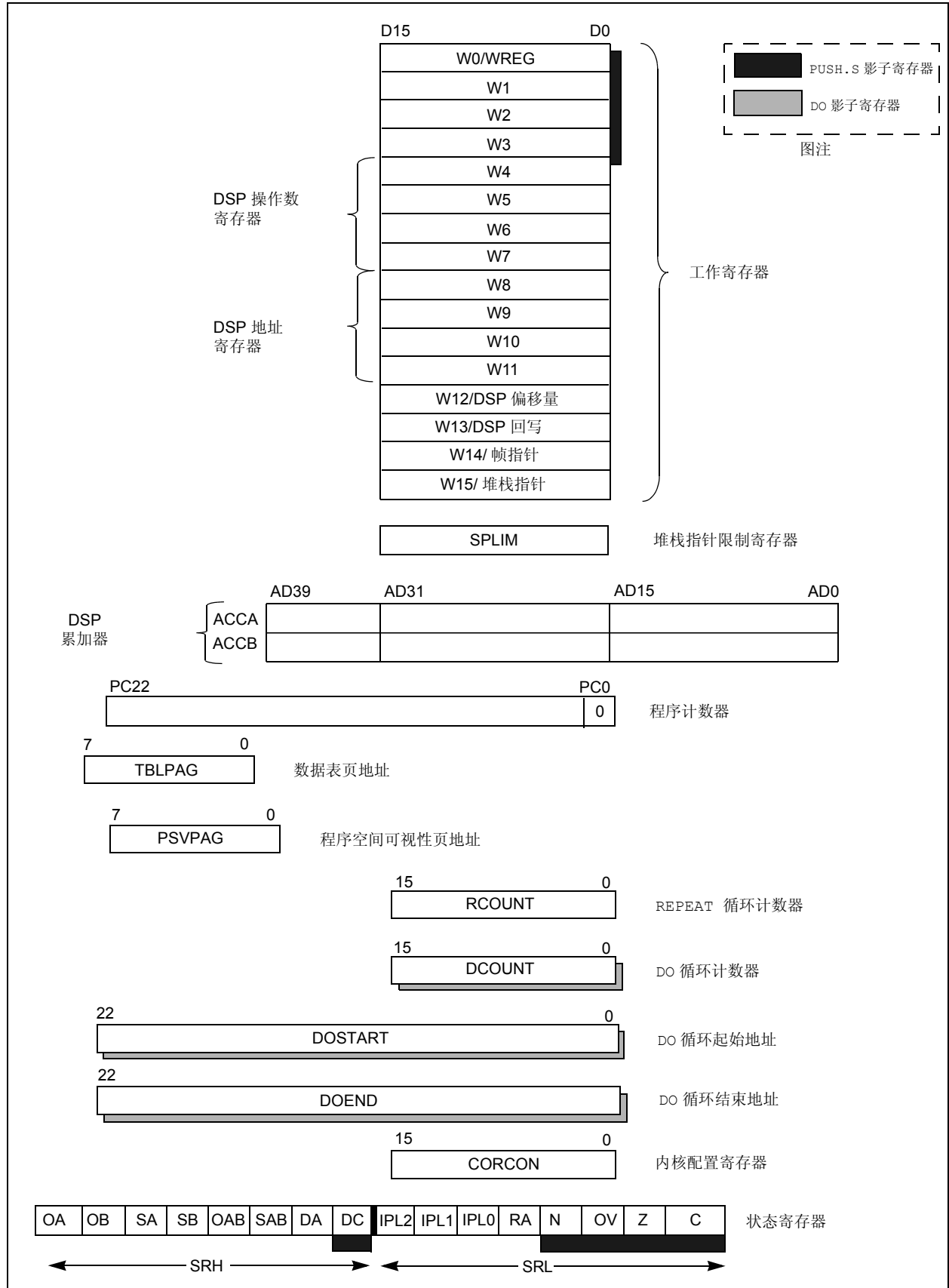
SRL 包含全部的 MCU ALU 运算状态标志位（包括 Z 位），以及 CPU 中断优先级状态位 IPL<2:0> 和 REPEAT 有效状态位 RA。在异常处理期间，SRL 与 PC 的 MSB 连接起来形成一个完整的字值，然后被压入堆栈。

状态寄存器的高字节包含 DSP 加法器 / 减法器状态位、DO 循环有效位（DA）和半进位（DC）状态位。

2.2.3 程序计数器

程序计数器为 23 位宽；bit 0 始终为零。因此，PC 能够最多寻址 4M 指令字。

图 2-1: 编程模型



2.3 除法支持

dsPIC DSC 器件具有这样的特性: 支持 16 位 /16 位有符号小数除法运算, 支持 32 位 /16 位、16 位 /16 位有符号和无符号整数除法运算, 除法形式均为单指令迭代除法。支持以下指令和数据长度:

1. DIVF — 16/16 有符号小数除法
2. DIV.sd — 32/16 有符号除法
3. DIV.ud — 32/16 无符号除法
4. DIV.sw — 16/16 有符号除法
5. DIV.uw — 16/16 无符号除法

16/16 除法类似于 32/16 除法 (它们的迭代次数相同), 但在第一次迭代时, 被除数要进行零扩展或符号扩展。

除法指令必须在一个 REPEAT 循环中执行。任何其他执行方式 (例如, 一系列不连续的除法指令), 将不会正常运行, 因为指令流取决于 RCOUNT。除法指令不会自动设置 RCOUNT 值, 于是 RCOUNT 必须显式地、正确地在 REPEAT 指令中给定, 如表 2-1 所示 (REPEAT 将执行目标指令 { 操作数的值 +1} 次)。必须为 DIV/DIVF 指令的 18 次迭代设置 REPEAT 循环计数。所以, 完整的除法操作需要 19 个周期。

注: 除法流可以被中断。然而, 用户需要恰当地保存中断现场。

表 2-1: 除法指令

指令	功能
DIVF	有符号小数除法: $Wm/Wn \rightarrow W0$; $Rem \rightarrow W1$
DIV.sd	有符号除法: $(Wm + 1:Wm)/Wn \rightarrow W0$; $Rem \rightarrow W1$
DIV.ud	无符号除法: $(Wm + 1:Wm)/Wn \rightarrow W0$; $Rem \rightarrow W1$
DIV.sw	有符号除法: $Wm/Wn \rightarrow W0$; $Rem \rightarrow W1$
DIV.uw	无符号除法: $Wm/Wn \rightarrow W0$; $Rem \rightarrow W1$

2.4 DSP 引擎

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

DSP 引擎能够执行固有的不需要其他数据的累加器至累加器操作。这些指令是 ADD、SUB 和 NEG。

通过 CPU 内核配置寄存器（CORCON）中的各个位，可以有多种 DSP 引擎选择，这些选择如下：

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

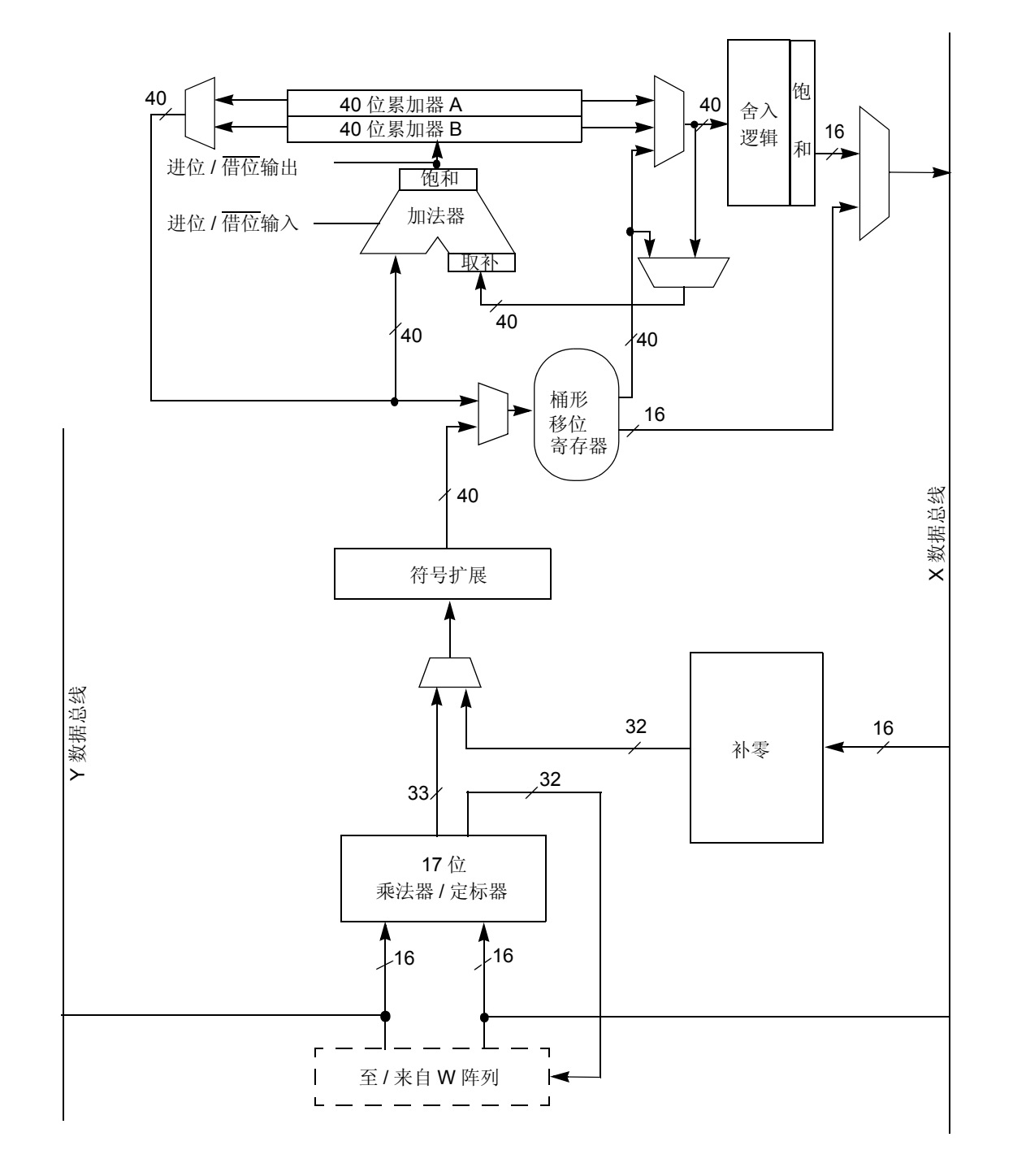
注： CORCON 的构成，见表 3-3。

DSP 引擎的框图，如图 2-4 所示。

表 2-2: DSP 指令汇总

指令	代数运算	ACC 回写?
CLR	$A = 0$	是
ED	$A = (x - y)^2$	否
EDAC	$A = A + (x - y)^2$	否
MAC	$A = A + (x * y)$	是
MAC	$A = A + x^2$	否
MOVSAC	A 中内容将不发生改变	是
MPY	$A = x * y$	否
MPY.N	$A = -x * y$	否
MSC	$A = A - x * y$	是

图 2-4: DSP 引擎框图



2.4.1 乘法器

17位×17位的乘法器可以进行有符号或无符号的运算，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，送入乘法器输入值的第 17 位。17 位×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，MSB 定义为符号位。一般来说，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，数据范围是 -32768 (0x8000) 至 32767 (0x7FFF)，包括 0 在内。对于 32 位整数，数据范围是 -2,147,483,648 (0x8000 0000) 至 2,147,483,645 (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是 -1.0 至 $(1-2^{1-N})$ 。对于 16 位小数，Q15 数据范围是 -1.0 (0x8000) 至 0.999969482 (0x7FFF)，包括 0 在内，其精度为 3.01518×10^{-5} 。在小数方式下，16x16 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、无符号和混和符号乘法。

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

2.4.2 数据累加器和加法器 / 减法器

数据累加器包含一个 40 位的加法器 / 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) 之一作为其预累加的源累加器和后累加的目标累加器。对于 ADD 和 LAC 指令，将被累加或装入的数据可选择通过桶形移位寄存器在累加之前进行换算。

2.4.2.1 加法器 / 减法器，溢出和饱和

加法器 / 减法器是一个 40 位的加法器，一侧输入可以选择为零，而另一侧的输入可以是数据的原码或补码。对于加法，进位 / 借位输入是高有效的，另一侧输入是数据的原码 (没有求补的)；对于减法，进位 / 借位输入是低有效的，另一侧输入是数据的补码。溢出和饱和状态位 SA/SB 及 OA/OB 提供加法器 / 减法器的溢出和饱和状态信息，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。
- 溢出到警戒位 (bit 32 到 bit 39)：这是可恢复的溢出。每当警戒位有任何不一致，就将把这个状态位置 1。

加法器有一个额外的饱和块，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、上述的溢出状态位以及 SATA/B (CORCON<7:6>) 和 ACCSAT (CORCON<4>) 模式控制位，来确定什么时候以及什么值要饱和。

在状态寄存器中有 6 个支持饱和溢出的位，它们是：

1. OA:
ACCA 溢出到警戒位
2. OB:
ACCB 溢出到警戒位
3. 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)
4. 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)
5. OAB:
OA 和 OB 的逻辑或 (OR)
6. 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经溢出到累加器警戒位 (bit 32 到 bit 39)。如果 OA 和 OB 位置 1 而且 INTCON1 寄存器中相应的溢出陷阱标志允许位 (OVATE 和 OVBTE) 置 1 的话，还可以选择用 OA 和 OB 位产生算术警告陷阱 (见第 5.0 节“中断”)。这使得用户能够立即采取措施，例如，改正系统增益。

每次数据通过加法器 / 减法器，就会修改 SA 和 SB 位，但用户只能对它们进行清零。置 1 时，它们表明累加器已经溢出其最大范围（对于 32 位饱和是 bit 31，而 40 位饱和是 bit 39），将发生饱和（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，于是表明产生了灾难性溢出。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。

器件支持三种饱和及溢出模式：

1. **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）或最大的负 9.31 值（0x80000000）装入目标累加器。SA 或 SB 位置 1 并保持置 1 直到被用户清零。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。
2. **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）或最大的负 1.31 值（0x00800000）装入目标累加器。SA 或 SB 位置 1 并保持置 1 直到被用户清零。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）。
3. **bit 39 灾难性溢出：**
来自加法器的 bit 39 溢出状态位，用来将 SA 或 SB 位置 1（置 1 后，这两位将保持置 1，除非用户进行清零）。不进行饱和操作，允许累加器溢出（破坏其符号）。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。

2.4.2.2 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 至 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：

1. **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。
2. **[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）。

2.4.2.3 舍入逻辑

舍入逻辑是一个组合块，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逻辑；如果此指令不指明舍入，就会存储一个截取的 1.15 数据值，简单地丢弃最低有效字（least significant word, lsw）。

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 至 bit 31）。如果 ACCxL 字（累加器的 bit 0 至 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，值将稍稍偏大（正偏）。

除非 ACCxL 等于 0x8000，否则收敛的（或无偏）舍入操作方式与常规舍入相同。ACCxL 等于 0x8000 时，要对 ACCxH 的最低位（累加器的 bit 16）进行检测。如果它为 1，ACCxH 递增 1。如果它为 0，ACCxH 不变。假设 bit 16 本身是随机的，这样的机制将消除任何可能累加的舍入偏差。

通过 X 总线，SAC 和 SAC.R 指令将目标累加器内容的截取（SAC）或舍入（SAC.R）形式存入数据存储空间（这受数据饱和的影响，见第 2.4.2.4 节“数据空间写饱和”）。注意，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 MAC 类指令，数据始终要进行舍入。

2.4.2.4 数据空间写饱和

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

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，并相应地进行调整。如果输入数据大于 0x007FFF，则写入数据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，则写入数据存储器中的数据被强制为最大的负 1.15 值，0x8000。源累加器的最高位位（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。

2.4.3 桶形移位寄存器

桶形移位寄存器在单个周期内最多可算术或逻辑右移 16 位或左移 16 位。源寄存器可以是两个 DSP 累加器中的任何一个，或者是 X 总线（支持寄存器或存储器数据的多位移位）。

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。

桶形移位寄存器是 40 位宽的，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 至 bit 31，左移则存放在 bit 0 至 bit 15。

注:

3.0 存储器构成

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。有关器件指令集和编程的更多信息，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

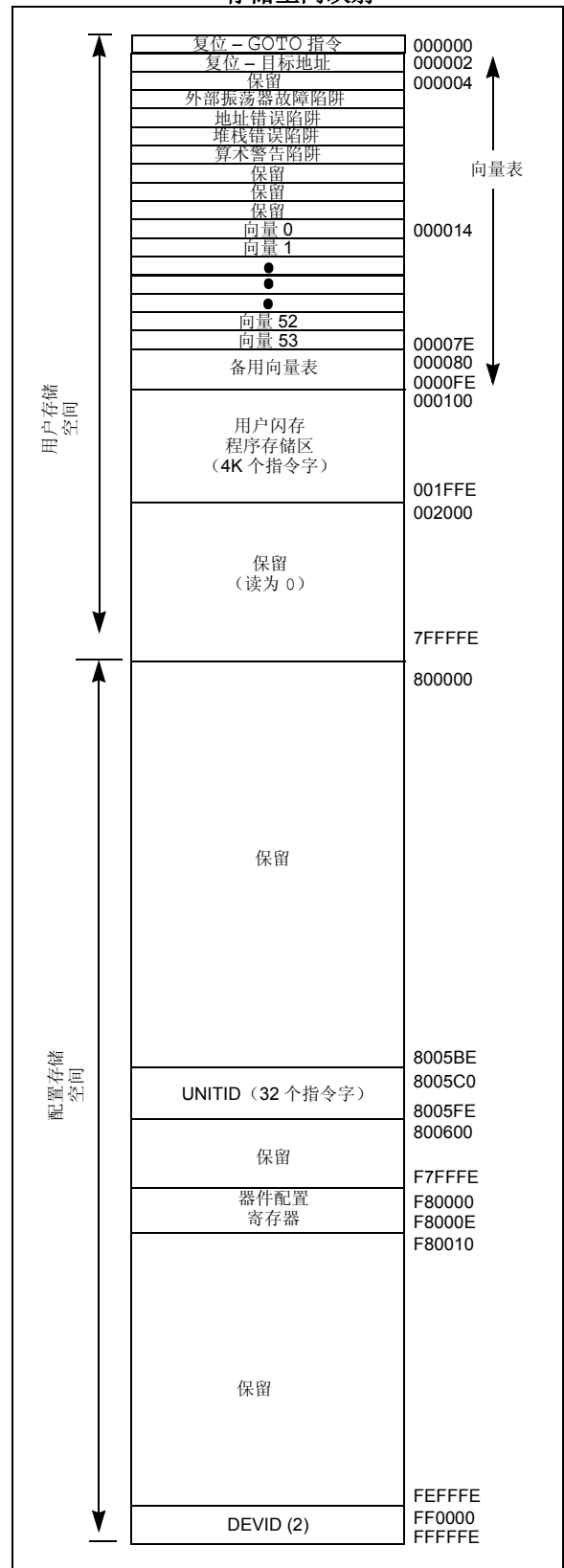
3.1 程序地址空间

程序地址空间为 4M 指令字，它可通过一个 24 位值来寻址，这 24 位值来自 23 位 PC，表指令有效地址 (Effective Address, EA)，或者来自数据空间 EA (当按照表 3-1 定义的那样，把程序空间映射到数据空间时)。注意，为了提供与数据空间寻址的兼容性，在两个连续的程序字之间，程序空间地址递增值为 2。

对于除 TBLRD/TBLWT 外的所有访问，用户程序空间访问限制在低 4M 指令字地址范围 (0x000000 至 0x7FFFFE)；TBLRD/TBLWT 使用 TBLPAG<7>来决定是访问用户空间还是访问配置空间。在表 3-1 “程序空间地址构成”中，bit 23 允许访问器件 ID、用户 ID 和配置位。对于其他情形，bit 23 始终清零。

注：图 3-1 所示的地址映射是概念性的，实际的存储器配置会根据具体器件可用的存储器大小而存在差异。

图 3-1: dsPIC30F1010/202X 的程序存储空间映射

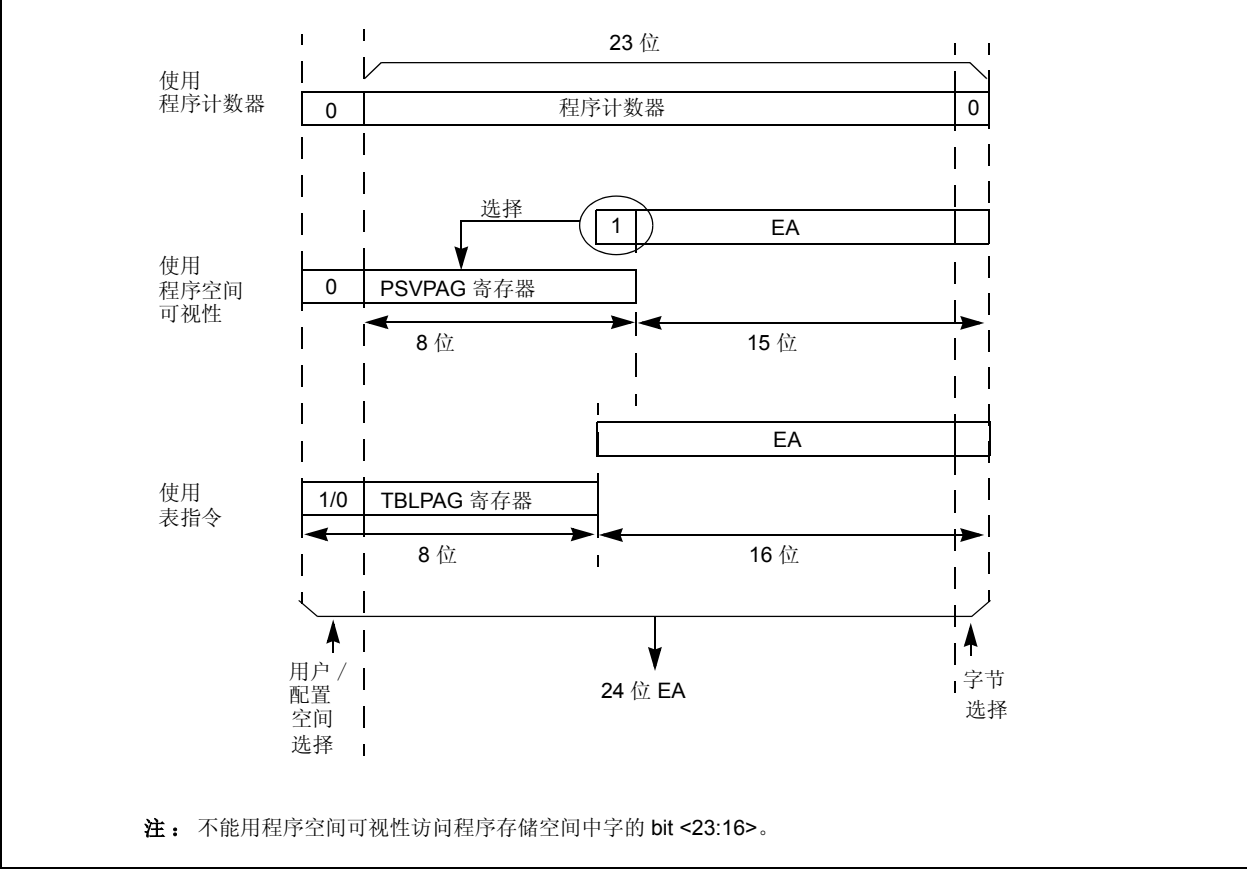


dsPIC30F1010/202X

表 3-1: 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	用户	0	PC<22:1>			0
TBLRD/TBLWT	用户 (TBLPAG<7> = 0)	TBLPAG<7:0>			数据 EA <15:0>	
TBLRD/TBLWT	配置 (TBLPAG<7> = 1)	TBLPAG<7:0>			数据 EA <15:0>	
程序空间可见性	用户	0	PSVPAG<7:0>		数据 EA <14:0>	

图 3-2: 访问程序空间中数据的地址生成



3.1.1 使用表指令访问程序存储器中的数据

本器件架构取指令时是取 24 位宽的程序存储器。因此，指令始终是对齐的。不过，由于器件架构是改进型哈佛架构，数据也可以出现在程序空间中。

有两种方法可以访问程序空间：通过特殊的表指令，或者通过把 16K 字程序空间页重新映射到数据空间的上半部分（见第 3.1.2 节“使用程序空间可视性访问程序存储器中的数据”）。TBLRD L 和 TBLWT L 指令提供了对程序空间内任何地址的最低有效字（lsw）直接读写的方法，而无需通过数据空间。TBLRD H 和 TBLWTH 指令是可以把一个程序空间字的高 8 位作为数据存取的惟一方法。

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

图 3-2 说明了如何为表操作和数据空间访问（PSV = 1）的创建 EA。这里，P<23:0> 指的是程序空间字，而 D<15:0> 指的是数据空间字。

器件提供了一组表指令，用来在程序空间和数据空间之间传送字节或字大小的数据。

- 1. **TBLRD L**: 表读低位
字: 读程序地址的低位字; P<15:0> 映射到 D<15:0>;
字节: 读程序地址的 LSB 中的一个字节;
当字节选择 = 0 时, P<7:0> 映射到目的字节;
当字节选择 = 1 时, P<15:8> 映射到目的字节。
- 2. **TBLWT L**: 表写低位字 (字节) (闪存编程的详细信息请参见第 7.0 节“闪存程序存储器”)。
- 3. **TBLRD H**: 表读高位
字: 读程序地址的高位字; P<23:16> 映射到 D<7:0>; D<15:8> 始终 = 0。
字节: 读程序地址的 MSB 中的一个字节;
当字节选择 = 0 时, P<23:16> 映射到目的字节;
当字节选择 = 1 时, 目的字节始终 = 0。
- 4. **TBLWTH**: 表写高位字 (字节) (闪存编程的详细信息请参见第 7.0 节“闪存程序存储器”)。

图 3-3: 程序数据表访问 (低位字)

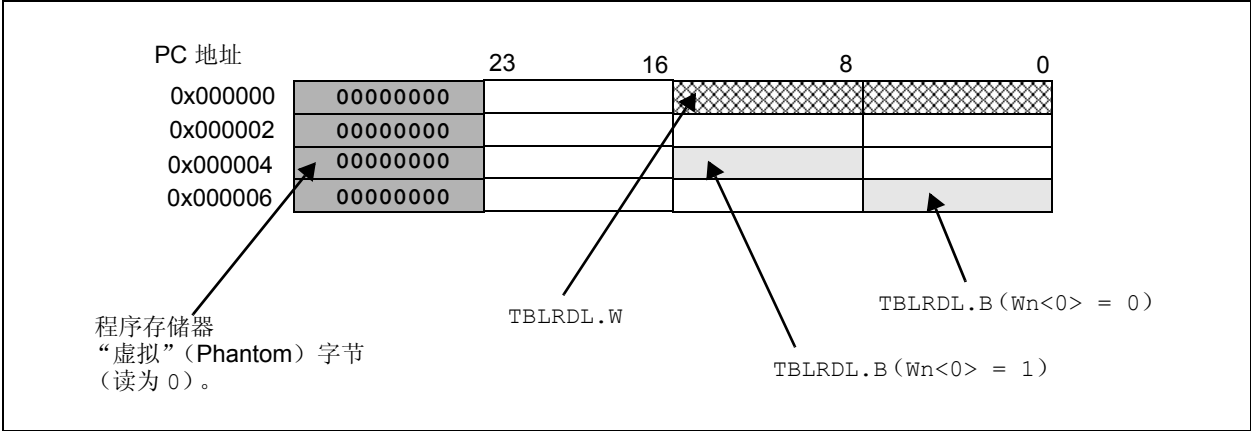
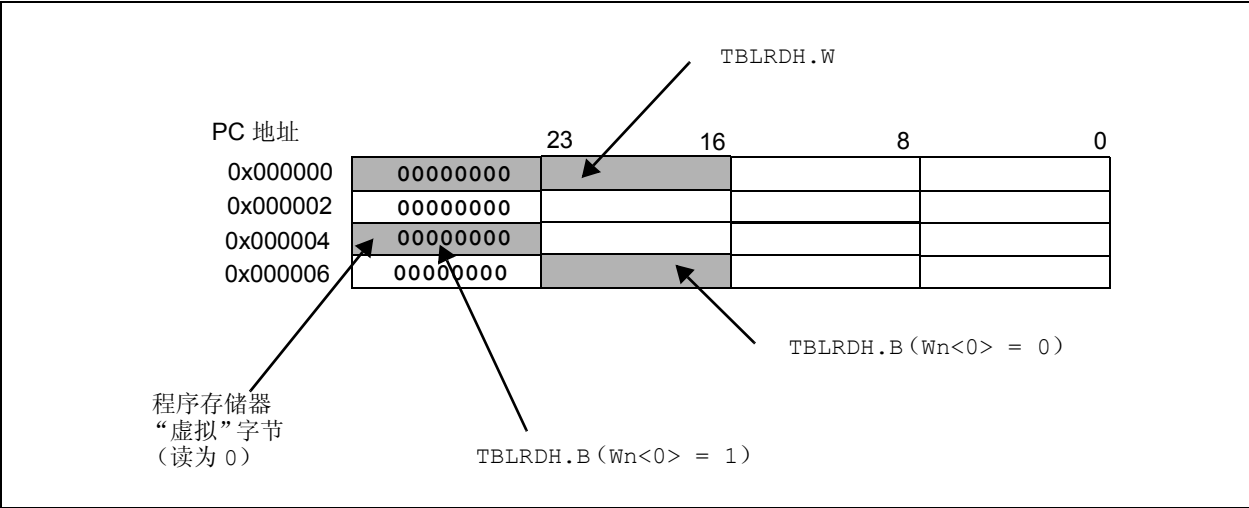


图 3-4: 程序数据表访问 (最高字节)



3.1.2 使用程序空间可视性访问程序存储器中的数据

可选择将数据空间的高 32 KB 映射到任何 16K 字程序空间页。这提供了通过 X 数据空间对存储的常量数据的透明访问，而无需使用特殊指令（即 TBLRDH/H 和 TBLWTL/H 指令）。

如果数据空间 EA 的 MSb 置 1，并且程序空间可视性使能（方法是在内核控制寄存器 CORCON 中把 PSV 位置 1）时，就能通过数据空间访问程序空间。CORCON 功能的讨论请参见第 2.4 节“DSP 引擎”。

正在执行的指令，如果要对这个区域进行数据访问的话，就需要一个额外的指令周期，因为需要进行两次程序空间读取操作。

注意，可寻址数据空间的上半部分始终是 X 数据空间的一部分。于是，当 DSP 操作使用程序空间映射来访问这个存储区域时，Y 数据空间通常应该存放 DSP 操作的状态（可变）数据，而 X 数据空间通常应该存放系数（常量）数据。

尽管每个数据空间地址，0x8000 和更高，直接映射到对应的程序存储器地址（见图 3-6），但只使用 24 位程序字的低 16 位来存放数据。应该进行恰当的设置，使得对高 8 位进行访问的指令都成为非法指令，以维持器件的可靠性。指令编码的细节，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

注意，对于每个程序存储字 PC 都将递增 2，数据空间地址的低 15 位将直接映射到相应程序空间地址的低 15 位。剩下的位由程序空间可视性页寄存器 (PSVPAG<7:0>) 提供，如图 3-5 所示。

注： 在表读 / 表写期间，暂时禁止 PSV 访问。

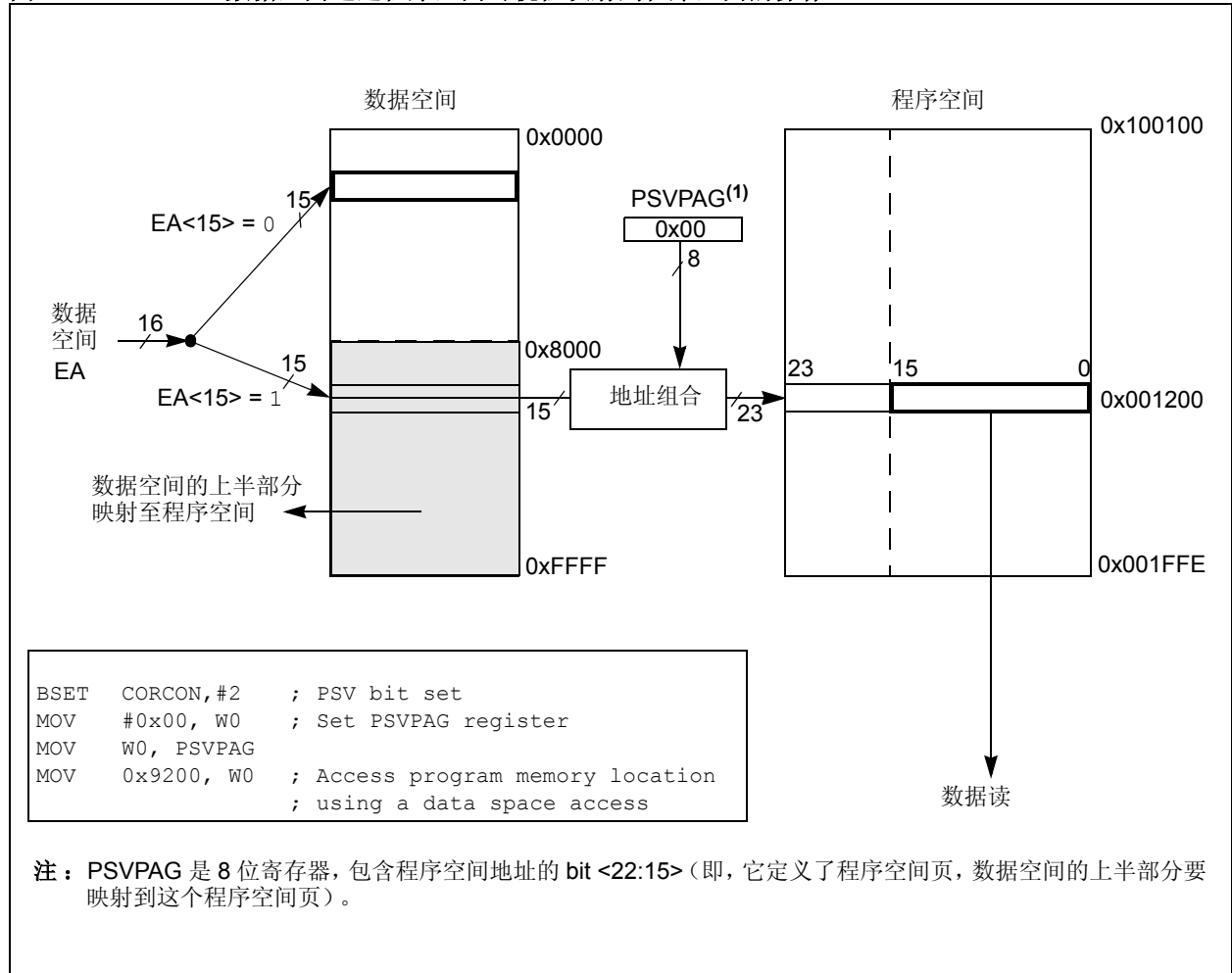
对于使用 PSV 而又在 REPEAT 循环之外执行的指令：

- 下列指令，除了规定的执行时间，还需要一个额外的指令周期：
 - 带数据操作数预取的 MAC 类指令
 - MOV 指令
 - MOV.D 指令
- 其他所有的指令，除了规定的指令执行时间，还需要两个额外的指令周期。

对于使用 PSV 而又在 REPEAT 循环内执行的指令：

- 下列情况，除了规定的指令执行时间，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
 - 中断得到处理后而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 3-5: 数据空间通过程序空间可视性映射到程序空间的操作



3.2 数据地址空间

内核具有两个数据空间。数据空间可以看作是独立的（对于某些 DSP 指令而言），或者看作是统一的线性地址范围（对于 MCU 指令而言）。使用两个地址发生单元（Address Generation Units, AGU）和独立的数据路径访问数据空间。

3.2.1 数据存储空间映射

数据存储空间分为两块，X 和 Y 数据空间。这个架构的关键之处在于，Y 空间是 X 空间的子集，Y 空间完全包含在 X 中。为了提供外在的线性寻址空间，X 和 Y 空间要有连续的地址。

在执行除 MAC 类指令之外的任何指令时，X 块由 256 字节数据地址空间（包括全部 Y 地址）组成。在执行一条 MAC 类指令时，X 块由 256 字节数据地址空间组成，但不包括 Y 地址块（只用来读数据）。换句话说，所有其他指令把整个数据存储空间当作一个复合的地址空间。MAC 类指令把 Y 地址空间从数据空间中抽取出来，使用源自 W10 和 W11 的 EA 来寻址 Y 地址空间。剩下的 X 数据空间则使用 W8 和 W9 来寻址。只有 MAC 类指令才能同时访问两个地址空间。

数据存储空间映射如图 3-6 所示。

dsPIC30F1010/202X

图 3-6: 数据存储空间映射

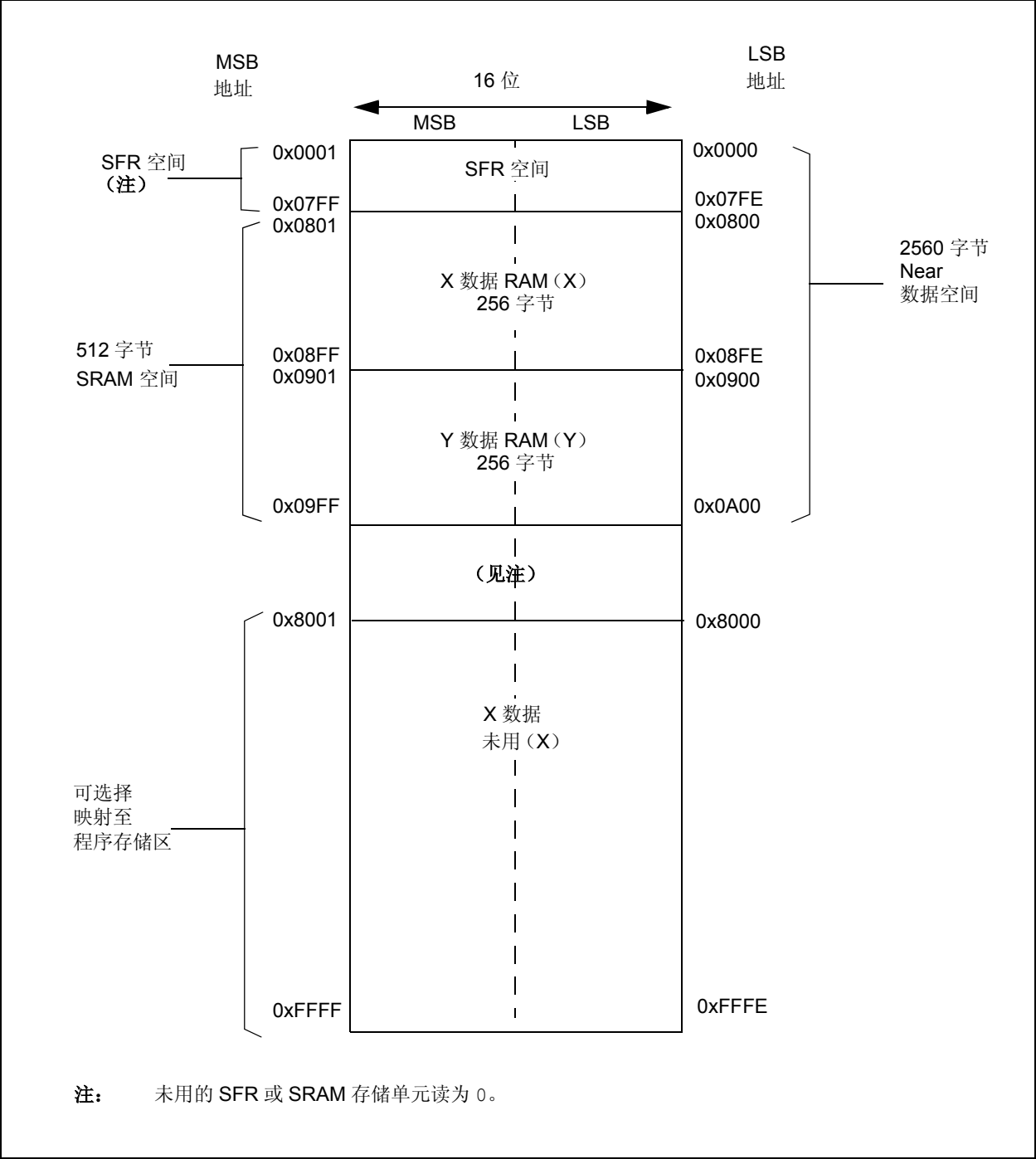
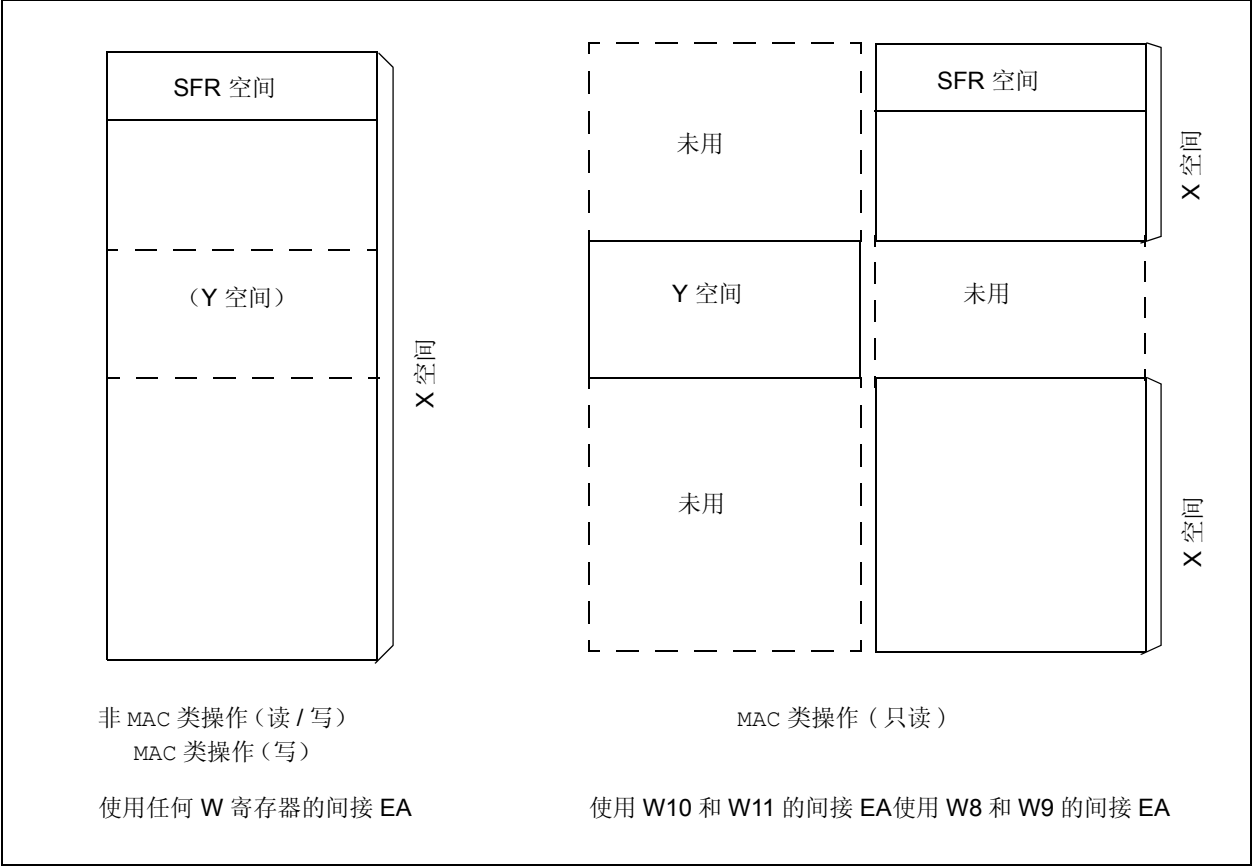


图 3-7: 用于 MCU 和 DSP (MAC 类) 指令的数据空间



3.2.2 数据空间

X 数据空间为全部指令所使用，支持全部的寻址模式。读数据总线和写数据总线是独立的。X 读数据总线是所有指令的返回数据路径，它把数据空间视为复合的 X 和 Y 地址空间。对于双操作数读指令（MAC 类），它还是 X 地址空间的数据路径。对于所有指令而言，X 写数据总线是至数据空间的只写路径。

X 数据空间还支持所有指令的模寻址，但受到寻址模式的限制。位反转寻址只是在写 X 数据空间时才支持。

MAC 类指令（CLR、ED、EDAC、MAC、MOVSAC、MPY、MPY.N 和 MSC）使用的 Y 数据空间，协同 X 数据空间，提供了两条并行的数据读取路径。Y 总线上不会出现写操作。MAC 类指令使用两个专用的 W 寄存器指针，W10 和 W11，它们始终寻址 Y 数据空间，与 X 数据空间独立；而 W8 和 W9 则始终寻址 X 数据空间。注意，在累加器回写期间，数据地址空间视为 X 和 Y 数据空间的组合，从而写操作通过 X 总线进行。所以，可以写整个数据空间中的任何地址。

Y 数据空间仅用于与 MAC 类指令关联的数据预取操作。它 also 支持自动循环缓冲区的模寻址。当然，所有其他指令可以通过 X 数据路径来把 Y 数据地址空间作为复合线性空间的一部分来进行访问。

图 3-6 给出了 X 和 Y 数据空间之间的边界定义，用户不能设定改变边界。如果 EA 指向其所在地址空间之外的数据，或者指向物理存储器之外的存储单元，将返回全零的字 / 字节。例如，尽管 Y 地址空间对于使用任何寻址模式的所有非 MAC 类指令而言，是可见的，如果一条 MAC 类指令试图使用 W8 或 W9（X 空间指针）从 Y 地址空间取数据，将返回 0x0000。

表 3-2: 非法存储器访问的后果

试图进行的操作	返回数据
EA = 未实现的地址	0x0000
在 MAC 类指令中使用 W8 或 W9 访问 Y 数据空间	0x0000
在 MAC 类指令中使用 W10 或 W11 访问 X 数据空间	0x0000

所有有效地址都是 16 位宽的，且指向数据空间内的字节。所以，数据空间地址范围是 64 KB 或 32K 字。

3.2.3 数据空间宽度

内核数据宽度为 16 位。所有内部寄存器都按照 16 位宽的字来组织。数据存储空间以字节可寻址的 16 位宽的块来组织。

3.2.4 数据对齐

为了维持与 PIC® 器件的向后兼容性，并提高数据存储空间的使用效率，dsPIC30F 指令集既支持字操作，也支持字节操作。在数据存储器和寄存器中，数据按照字来对齐，但是，所有数据空间有效地址都将解析为字节。数据字节读取将读取包含字节的整个字，使用任何有效地址（EA）的 LS 位确定要选取的字节。选取的字节将存放到 X 数据路径的 LSB 中（不可能出现来自 Y 数据路径的字节访问，因为 MAC 类指令只能取整字）。这就是说，数据存储器和寄存器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，将只写阵列或寄存器中与字节地址匹配的那一侧。

这种字节可访问能力的结果是，所有的有效地址计算（包括那些由 DSP 操作产生的有效地址，它们只能是字长度的数据）在内部进行换算，以适应字对齐的存储空间。例如，内核将识别执行后修改寄存器间接寻址模式 [Ws++] 的结果，对于字节操作产生的值是 Ws + 1，而对于字操作产生的值是 Ws + 2。

所有的字访问必须按照偶数地址对齐。不支持没有对齐的字数据取，所以在混和字节和字操作时，或者从 8 位 MCU 代码进行转换时，必须要小心。如果试图进行未对齐的读或写，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，都将产生陷阱，从而系统和 / 或用户能够检查地址错误发生之前的机器状态。

图 3-8: 数据对齐

	15	MSB	8	7	LSB	0	
0001	字节 1		字节 0				0000
0003	字节 3		字节 2				0002
0005	字节 5		字节 4				0004

所有载入 W 寄存器的字节都将载入 W 寄存器的 LSB，W 寄存器的 MSB 不变。

提供了一条符号扩展 (SE) 指令，允许用户把 8 位的有符号数据转换为 16 位有符号值。或者，对于 16 位无符号数据，用户可以清零任何 W 寄存器的 MSB，方法是在相应的地址处执行一条零扩展 (ZE) 指令。

尽管大多数指令能够对字或字节大小的数据进行操作，但应该注意的是，一些指令，包括 DSP 指令，只对字大小的数据进行操作。

3.2.5 NEAR 数据空间

X 地址存储空间中，在 0x0000 和 0x1FFF 之间保留了一个 8 KB 的 near 数据空间；在所有的存储器直接寻址指令中，可以通过一个 13 位的绝对地址来直接访问这个数据空间。其余的 X 地址空间和全部的 Y 地址空间都是间接可寻址的。此外，使用 MOV 指令可以寻址整个 X 数据空间，这支持通过 16 位地址字段进行存储器直接寻址。

3.2.6 软件堆栈

dsPIC DSC 器件具备一个软件堆栈。W15 用作堆栈指针。

堆栈指针总是指向堆栈顶部第一个可供使用的字，从低地址到高地址方向增长。堆栈指针在弹出堆栈之前递减，而在压入堆栈后递增，如图 3-9 所示。注意，对于任何 CALL 指令时的 PC 压栈，在压入堆栈之前，PC 的 MSB 要进行零扩展，从而确保了 MSB 始终是清零的。

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

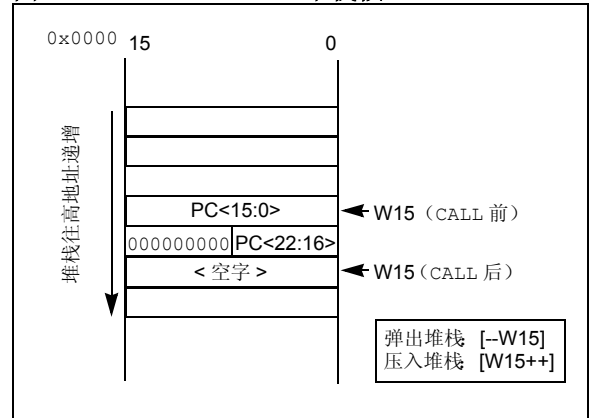
堆栈指针限制寄存器 (SPLIM) 与堆栈指针相关联。复位时 SPLIM 未被初始化。与堆栈指针的情况一样，SPLIM<0> 强制为 0，因为所有的堆栈操作必须是字对齐的。每当使用 W15 作为源指针或目的指针产生有效

地址 EA 时，将与 SPLIM 中的值进行比较。如果堆栈指针 (W15) 与 SPLIM 寄存器的内容相等，则执行压入操作，不产生堆栈错误陷阱。但在随后的压栈操作时将会产生堆栈错误陷阱。这样的话，当堆栈增长超过 RAM 中地址 0x2000 时，如果要想产生堆栈错误陷阱，用值 0x1FFE 来初始化 SPLIM 即可。

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，这避免了堆栈干预特殊功能寄存器 (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 3-9: CALL 堆栈帧



3.2.7 数据 RAM 保护

dsPIC30F1010/202X 器件支持数据 RAM 保护功能，当该功能与引导段代码保护功能配合使用时可以对 RAM 中的段进行保护。BSRAM (用作 BS 的安全 RAM 段) 在使能时只能通过引导段安全闪存代码进行访问。请参见表 3-3 了解 BSRAM SFR。

表 3-3: 内核寄存器映射

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态	
W0	0000	W0/WREG																0000 0000 0000 0000	
W1	0002	W1																0000 0000 0000 0000	
W2	0004	W2																0000 0000 0000 0000	
W3	0006	W3																0000 0000 0000 0000	
W4	0008	W4																0000 0000 0000 0000	
W5	000A	W5																0000 0000 0000 0000	
W6	000C	W6																0000 0000 0000 0000	
W7	000E	W7																0000 0000 0000 0000	
W8	0010	W8																0000 0000 0000 0000	
W9	0012	W9																0000 0000 0000 0000	
W10	0014	W10																0000 0000 0000 0000	
W11	0016	W11																0000 0000 0000 0000	
W12	0018	W12																0000 0000 0000 0000	
W13	001A	W13																0000 0000 0000 0000	
W14	001C	W14																0000 0000 0000 0000	
W15	001E	W15																0000 1000 0000 0000	
SPLIM	0020	SPLIM																0000 0000 0000 0000	
ACCAL	0022	ACCAL																0000 0000 0000 0000	
ACCAH	0024	ACCAH																0000 0000 0000 0000	
ACCAU	0026	符号扩展 (ACCA<39>)								ACCAU								0000 0000 0000 0000	
ACCBH	0028									ACCBH								0000 0000 0000 0000	
ACCBH	002A	ACCBH																0000 0000 0000 0000	
ACCBH	002C	符号扩展 (ACCB<39>)								ACCBH								0000 0000 0000 0000	
PCL	002E									PCL								0000 0000 0000 0000	
PCH	0030	—	—	—	—	—	—	—	—	—	PCH				0000 0000 0000 0000				
TBLPAG	0032	—	—	—	—	—	—	—	—	TBLPAG				0000 0000 0000 0000					
PSVPAG	0034	—	—	—	—	—	—	—	—	PSVPAG				0000 0000 0000 0000					
RCOUNT	0036	RCOUNT																uuuu uuuu uuuu uuuu	
DCOUNT	0038	DCOUNT																uuuu uuuu uuuu uuuu	
DOSTARTL	003A	DOSTARTL																0	
DOSTARTH	003C	—	—	—	—	—	—	—	—	—	DOSTARTH				uuuu uuuu uuuu uuu0				
DOENDL	003E	DOENDL																0	
DOENDH	0040	—	—	—	—	—	—	—	—	—	DOENDH				0000 0000 0uuu uuuu				
SR	0042	OA	OB	SA	SB	OAB	SAB	DA	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000 0000 0000 0000	
CORCON	0044	—	—	—	—	—	US	EDT	DL1	DL0	SATA	SATB	SATDW	ACCSAT	IPL3	PSV	RND	IF	0000 0000 0010 0000

图注: u = 未初始化的位

表 3-3: 内核寄存器映射 (续)

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态				
MODCON	0046	XMODEN	YMODEN	—	—	BWM<3:0>				YWM<3:0>				XWM<3:0>				0000 0000 0000 0000				
XMODSRT	0048																	0				
XMODEND	004A																	1				
YMODSRT	004C																	0				
YMODEND	004E																	1				
XBREV	0050	BREN	XB<14:0>																			
DISICNT	0052	—	—	DISICNT<13:0>																		0000 0000 0000 0000
BSRAM	0750	—	—	—	—	—	—	—	—	—	—	—	—	—	IW_BSR	IR_BSR	RL_BSR	0000 0000 0000 0000				

图注: u = 未初始化的位

注: 有关寄存器位段的描述请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

注:

4.0 地址发生器单元

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。有关器件指令集和编程的更多信息，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

dsPIC DSC 内核包含两个独立的地址发生器单元：X AGU 和 Y AGU。Y AGU 仅支持 DSP MAC 类指令的字数数据读取。dsPIC DSC AGU 支持如下三种数据寻址类型

- 线性寻址
- 模（循环）寻址
- 位反转寻址

线性和模数据寻址模式可应用于数据空间或程序空间。位反转寻址只能用于数据空间地址。

4.1 指令寻址模式

寻址模式经过优化可以支持各指令的具体功能，基本寻址模式如表 4-1 所示。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略有不同。

4.1.1 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者 WREG（MUL 指令除外），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，可以访问整个数据空间。

表 4-1: 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 Wn（递增或递减）。
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 Wn（递增或递减），Wn 的内容形成 EA。
寄存器偏移量的寄存器间接寻址	Wn 和 Wb 的和形成 EA。
立即数偏移量的寄存器间接寻址	Wn 和立即数的和形成 EA。

4.1.2 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 始终是工作寄存器（即，寻址模式只能是寄存器直接寻址），称为 **Wb**。操作数 2 可以是一个 **W** 寄存器，取自数据存储器或一个 5 位立即数。结果位置（操作数 3）可以是 **W** 寄存器或地址单元。MCU 指令支持下列寻址模式：

- 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： 并非所有的指令都支持上面给出的全部寻址模式。各条指令可能支持这些寻址模式的某些模式，指令不同支持的寻址模式可能不同。

4.1.3 传送指令和累加器指令

与其他指令相比，传送和 **DSP** 累加器类指令提供了更为零活的寻址模式。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寄存器偏移量的寄存器间接寻址模式，这也叫做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寄存器和目标寄存器有效地址，可以是不同的。然而，4 位 **Wb**（寄存器偏移量）字段为源寄存器和目标寄存器所共用（但通常只由源寄存器或目标寄存器之一使用）。

概括地说，传送指令和累加器指令支持下列寻址模式：

- 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
- 寄存器偏移量的寄存器间接寻址（变址寻址）
- 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： 并非所有的指令都支持上面给出的全部寻址模式。各条指令可能支持这些寻址模式的某些模式，指令不同所支持的寻址模式可能不同。

4.1.4 MAC 类指令

双源操作数 **DSP** 指令（CLR、ED、EDAC、MAC、MPY、MPY.N、MOVSAAC 和 MSC），也叫做 MAC 类指令，它们使用一组简化的寻址模式，允许用户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，W8 和 W9 总是分配给 **X RAGU**，而 W10 和 W11 则始终用于 **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，仅可用于 W9（在 **X** 空间中）和 W11（在 **Y** 空间中）。

概括地说，MAC 类指令支持下列寻址模式：

- 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）

4.1.5 其他指令

除了上述的各种寻址模式之外，一些指令使用各种大小的立即数。例如，BRA（转移）指令使用 16 位有符号立即数来直接指定转移的目标，而 DISI 指令使用一个 14 位无符号立即数字段。在一些指令中，如 ADD ACC，操作数的来源和运算结果已经暗含在操作码中。某些操作，如 NOP，不含任何操作数。

4.2 模寻址

模寻址模式是一种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 算法中很常见），不需要用软件来执行数据地址边界检查。

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

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

使用限制的惟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进行地址边界检查）。

4.2.1 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，并将其载入 16 位模缓冲区地址寄存器（XMODSRT、XMODEND、YMODSRT 和 YMODEND，见表 3-3）中。

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，相应的起始地址和结束地址之差可以确定长度。循环缓冲区最大长度为 32K 字（64 KB）。

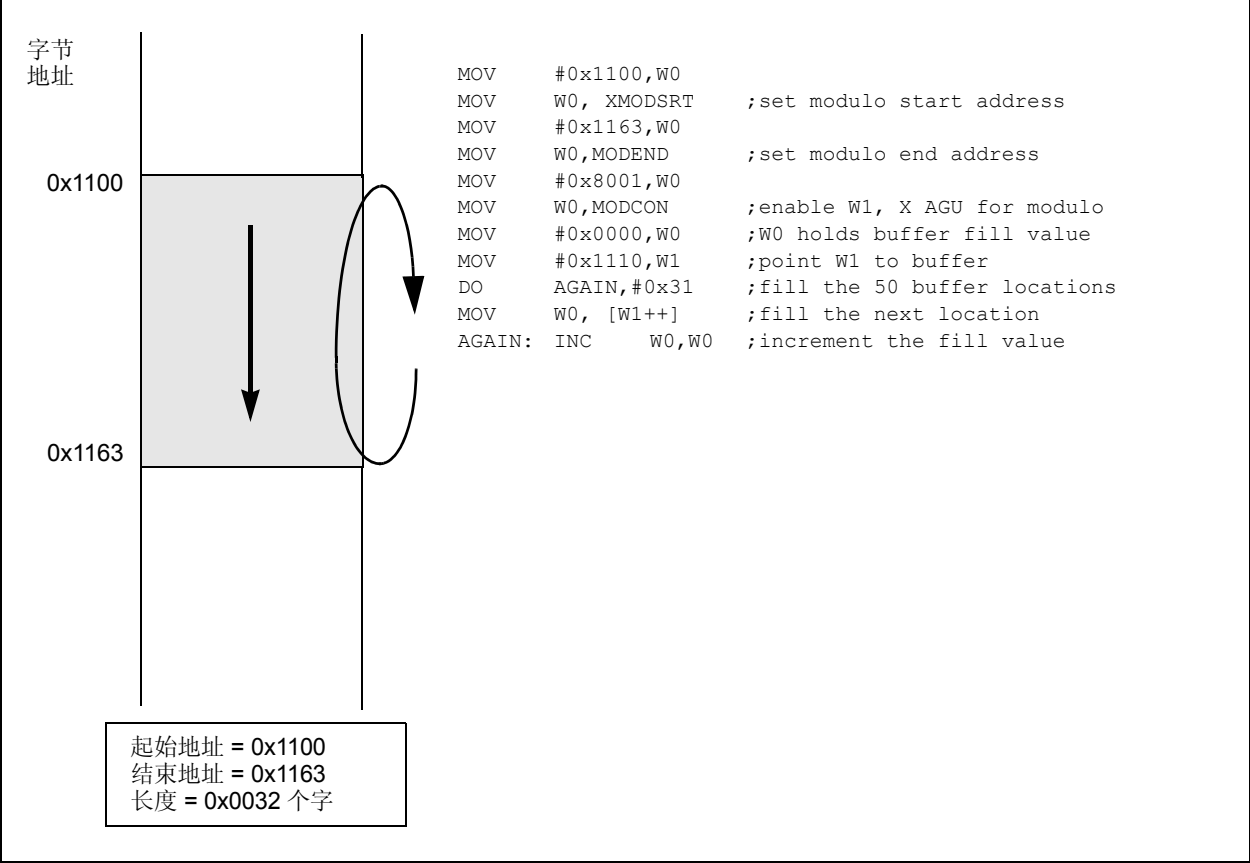
4.2.2 W 地址寄存器选择

模寻址和位反转寻址控制寄存器 MODCON<15:0> 中包含使能标志以及指定 W 地址寄存器的 W 寄存器字段。XWM 和 YWM 字段选择对哪些寄存器进行模寻址。如果 XWM = 15，则禁止 X RAGU 和 X WAGU 模寻址。同理，如果 YWM = 15，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）位于 MODCON<3:0> 中（见表 3-3）。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）位于 MODCON<7:4> 中。当 YWM 被设置为除 15 之外的任何值且 YMODEN 位（MODCON<14>）置 1 时，Y 数据空间的模寻址被使能。

图 4-1: 模寻址操作示例



4.2.3 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。地址边界检查功能不仅会对正好处在地址边界上的地址进行检查, 而且也会对地址小于或大于上限 (对于递增缓冲区)、低于下限 (对于递减缓冲区) 的地址进行检查, 用户认识到这一点是很重要的。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, $[W7 + W2]$), 会进行地址修正, 但寄存器的内容保持不变。

4.3 位反转寻址

位反转寻址用来简化基 2 FFT 算法的数据重新排序。**XWAGU** 仅支持数据写入的位反转寻址。

地址修改量可以是常数或寄存器的内容, 它被视为其位顺序被反转。地址的源和目的仍然是正常的顺序。因此, 惟一需要反转的操作数就是地址修改量。

4.3.1 位反转寻址的实现

位反转寻址的使能方式如下:

1. **MODCON** 寄存器中 **BWM** (**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), 且
2. **XBREV** 寄存器中 **BREN** 位置 1, 且
3. 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是 $M = 2^N$ 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” (pivot point), 通常是一个常数。对于 FFT 计算, 其值等于 FFT 数据缓冲区长度的一半。

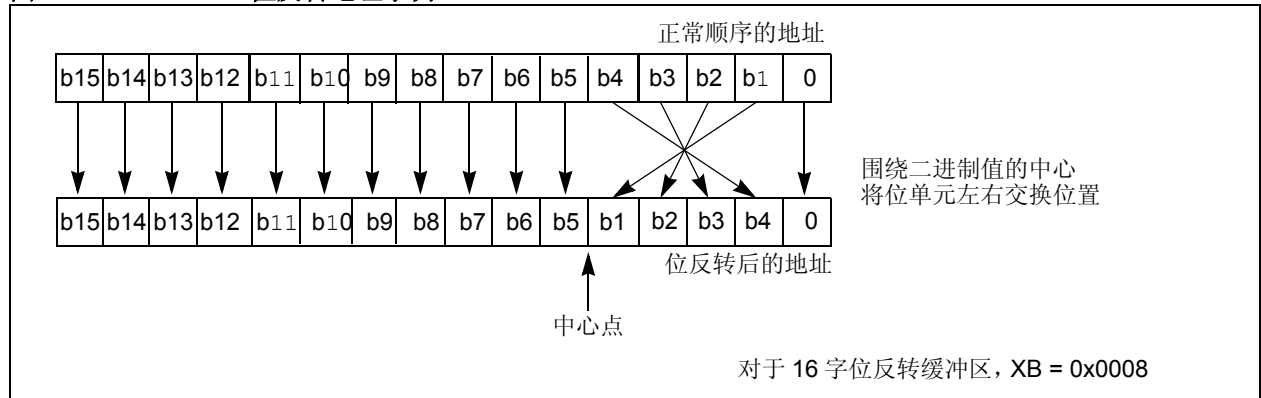
注: 所有位反转 **EA** 的计算都使用字数据 (每个 **EA** 的 **LSb** 始终为零)。为了产生兼容地址 (字节), 要相应地调整 **XB** 的值。

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数据, 不会进行位反转寻址, 而是生成正常的地址。在进行位反转寻址时, **W** 地址指针的增量将始终加上地址修改量 (**XB**), 与寄存器间接寻址模式相关的偏移量将被忽略。此外, 由于要求是字数据, **EA** 的 **LSb** 被忽略 (且始终被清零)。

注: 不应同时使能模寻址和位反转寻址。如果用户试图这么做的话, 对于 **XWAGU**, 位反转寻址将优先, **XWAGU** 模寻址将被禁止。然而, 在 **XRAGU** 中, 模寻址继续起作用。

如果通过置 1 **BREN** (**XBREV<15>**) 位使能了位反转寻址, 那么, 在写 **XBREV** 寄存器之后, 不应马上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 4-2: 位反转地址示例



dsPIC30F1010/202X

表 4-2: 位反转地址序列 (16 个项)

正常地址					位反转后的地址				
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

表 4-3: XBREV 寄存器的位反转地址修改量

缓冲区大小 (字)	XB<14:0> 位反转地址修改量 ⁽¹⁾
32768	0x4000
16384	0x2000
8192	0x1000
4096	0x0800
2048	0x0400
1024	0x0200
512	0x0100
256	0x0080
128	0x0040
64	0x0020
32	0x0010
16	0x0008
8	0x0004
4	0x0002
2	0x0001

注 1: 大于 256 字的修改量将超出 dsPIC30F1010/202X 器件的数据存储空间范围。

5.0 中断

注：本数据手册总结了 dsPIC30 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和通用器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。欲知有关器件指令集和编程的更多详细信息，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

dsPIC30F1010/202X 器件最多具有 35 个中断源和 4 个处理器异常（陷阱），必须依据优先级机制对它们进行仲裁。

CPU 负责读取中断向量表（Interrupt Vector Table, IVT）并将包含在中断向量中的地址传输给程序计数器（PC）。中断向量通过程序计数器输入端的 24 位宽度的多路开关，从程序数据总线传输到程序计数器。

中断向量表（IVT）和备用中断向量表（Alternate Interrupt Vector Table, AIVT）位于程序存储器起始地址附近（0x000004）。IVT 和 AIVT 如图 5-1 所示。

中断和处理器异常传送到处理器内核前，中断控制器负责对它们进行预处理。使用集中式的特殊功能寄存器允许、控制外设中断和陷阱并划分外设中断和陷阱的优先级。

- IFS0<15:0>、IFS1<15:0> 和 IFS2<15:0>
这三个寄存器包含所有中断请求标志位。各中断请求标志位通过相应外设或外部信号置 1，由软件清零。
- IEC0<15:0>、IEC1<15:0> 和 IEC2<15:0>
这三个寄存器包含所有中断允许控制位。这些控制位用于单独允许外设或外部信号中断。
- IPC0<15:0> 至 IPC11<7:0>
与这些中断中的每一个中断有关的用户可分配中断优先级集中存放在这 12 个寄存器中。
- IPL<3:0>
当前 CPU 的优先级明确地储存在 IPL 位中。IPL<3> 位于 CORCON 寄存器中，而 IPL<2:0> 位于处理器内核中的状态寄存器（SR）中。
- INTCON1<15:0> 和 INTCON2<15:0>
全局中断控制功能由这两个寄存器提供。INTCON1 包含处理器异常的控制和状态标志。INTCON2 寄存器控制外部中断请求信号的操作以及备用中断向量表的使用。

- 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>）位字段中。新中断优先级是等待处理的中断的优先级。

注：发生中断时，无论相应的中断允许位的状态如何，中断标志位都将置 1。用户软件应确保在允许一个中断前先将相应的中断标志位清零。

用户可以通过 IPCx 寄存器为所有中断源分配从 1 到 7 的 7 种优先级之一。每个中断源都与一个中断向量关联，如图 5-1 所示。优先级 7 和 1 分别表示最高和最低的可屏蔽优先级。

注：为中断源分配优先级为 0 等同于禁止该中断。

如果 NSTDIS 位（INTCON1<15>）置 1，则禁止中断嵌套。因此，正在处理一个中断时，禁止处理新的中断，即使新中断的优先级比当前正在处理的中断优先级高。

注：一旦 NSTDIS 位置 1，IPL 位就变为只读。

某些中断具有专门的控制位，用于控制诸如边沿或电平触发中断、电平变化中断这样的功能。这些功能的控制仍然由产生中断的外设模块负责。

对于某些指令，在其执行期间如果 DISI 位（INTCON2<14>）保持置 1 的话，那么可以用 DISI 指令来禁止优先级为 6 或更低的中断的处理。

当中断得到服务时，PC 中装入存放在程序存储器中相应中断向量存储单元中的地址。IVT 中有 63 个不同的向量（见图 5-1）。这些向量存放在程序存储器从 0x000004 到 0x0000FE 的存储单元中（见图 5-1）。这些存储单元中存有 24 位地址；为了保持鲁棒性，如果在正常执行期间，PC 试图取任何这些字的话，就会产生地址错误陷阱。这样便可防止由于以下情况而执行随机数据：PC 意外地递减到向量空间、意外地将数据空间地址映射到向量空间或 PC 在到达已实现的程序存储空间的末尾后重新回到 0x000000。执行 GOTO 指令跳转到向量空间，也将产生地址错误陷阱。

5.1 中断优先级

对于每个中断源，用户可分配的中断优先级位（IP<2:0>），位于 IPCx 寄存器中每个半字节的最低 3 位中。每个半字节的 bit 3 未使用，读为 0。这些位定义了用户分配给特定中断的优先级。

注： 用户可选择的优先级为从 0 到 7，0 是最低优先级，7 是最高优先级。

由于某一特定的用户指定优先级会被分配给多个中断请求源，因此提供了一种在给定的用户分配优先级内分配优先级的方法。这种方法称为“自然顺序优先级”，并最终以此为准。

自然优先级取决于中断在向量表中的位置，仅当多个具有相同用户分配优先级的中断同时等待响应时才对中断操作产生影响。

表 5-1 列出了 dsPIC DSC 器件的中断编号和中断源，以及关联的向量编号。

- 注 1：** 在自然顺序优先级中，0 为最高优先级，53 为最低优先级。
- 2：** 自然顺序优先级编号与 INT 编号相同。

用户能够为每个中断分配七个优先级中的一个，这意味着，对于一个具有低自然顺序优先级的中断，用户能够分配一个非常高的总优先级。例如，可为 INTO（外部中断 0）分配中断优先级 1，这样，它的有效优先级将非常低。

表 5-1: dsPIC30F1010/202X 中断向量表

中断编号	向量编号	中断源
最高自然优先级		
0	8	INT0 — 外部中断 0
1	9	IC1 — 输入捕捉 1
2	10	OC1 — 输出比较 1
3	11	T1 — Timer 1
4	12	保留
5	13	OC2 — 输出比较 2
6	14	T2 — Timer 2
7	15	T3 — Timer 3
8	16	SPI1
9	17	U1RX — UART1 接收器
10	18	U1TX — UART1 发送器
11	19	ADC — ADC 转换完成
12	20	NVM — NVM 写完成
13	21	SI2C — I ² C™ 从事件
14	22	MI2C — I ² C 主事件
15	23	保留
16	24	INT1 — 外部中断 1
17	25	INT2 — 外部中断 2
18	26	PWM 特殊事件触发器
19	27	PWM 发生器 1
20	28	PWM 发生器 2
21	29	PWM 发生器 3
22	30	PWM 发生器 4
23	31	保留
24	32	保留
25	33	保留
26	34	保留
27	35	CN — 输入状态变化通知
28	36	保留
29	37	模拟比较器 1
30	38	模拟比较器 2
31	39	模拟比较器 3
32	40	模拟比较器 4
33	41	保留
34	42	保留
35	43	保留
36	44	保留
37	45	ADC 对 0 转换完成
38	46	ADC 对 1 转换完成
39	47	ADC 对 2 转换完成
40	48	ADC 对 3 转换完成
41	49	ADC 对 4 转换完成
42	50	ADC 对 5 转换完成
43	51	保留
44	52	保留
45-53	53-61	保留
最低自然优先级		

5.2 复位过程

复位不是真正的异常，因为复位过程中并不涉及中断控制器。在响应强制清零 PC 的复位时，器件初始化其寄存器。然后，处理器开始在 0x000000 地址处执行程序。一条 GOTO 指令存放在第一个程序存储单元中，紧跟着的是 GOTO 指令的目标地址。处理器执行 GOTO，跳转到指定地址，然后开始指定目标（起始）地址处的操作。

5.2.1 复位源

除外部复位和上电复位（POR）之外，还有 6 个错误条件会导致处理器复位。

- 看门狗定时器超时：
看门狗定时器已经超时，表明处理器不再执行正确的代码流。
- 未初始化的 W 寄存器陷阱：
试图把未初始化的 W 寄存器用作地址指针将导致复位。
- 非法指令陷阱：
试图执行任何未使用的操作码，将产生非法指令陷阱。注意，如果由于指令流改变而使非法指令在执行前就被从流水线中舍弃的话，取非法指令并不产生非法指令陷阱。
- 陷阱锁定：
同时发生多个陷阱条件将导致复位。

5.3 陷阱

可将陷阱看作一个表明软件或硬件错误的不可屏蔽的中断，它遵从图 5-1 所示的预定义的优先级。陷阱旨在为用户提供一种方法，改正正在调试和在应用中工作时的错误操作。

注： 如果用户不想在出现陷阱错误条件时采取纠正措施，那么必须在这些陷阱向量中装入缺省陷阱处理程序的地址，缺省陷阱处理程序仅包含 RESET 指令。否则，如果调用了包含非法地址的陷阱向量，将产生地址错误陷阱。

注意，很多陷阱条件只有在发生时才能检测到。因此，在陷阱异常处理之前允许有问题的指令完成执行。如果用户选择从错误中恢复，可能需将导致陷阱的错误操作结果进行修正。

陷阱有 8 个固定的优先级：8 到 15，这意味着在陷阱处理期间，IPL3 始终置 1。

如果用户目前不在执行陷阱处理，而将 IPL<3:0> 设置为 0111（优先级 7），这样就禁止了所有中断，但仍然能够处理陷阱。

5.3.1 陷阱源

下列陷阱的优先级依次递增。然而，由于所有陷阱都可以嵌套，优先级的作用很小。

算术错误陷阱：

在以下四种情况下产生算术错误陷阱：

1. 如果试图进行以零作除数的除法运算，除法运算将在周期边界处中止，并产生陷阱。
2. 如果已使能，则当累加器 A 或 B 上的算术运算导致从 bit 31 溢出且累加器未使用警戒位时产生算术错误陷阱。
3. 如果已使能，则当累加器 A 或 B 上的算术运算导致从 bit 39 灾难性溢出且已禁止了所有饱和时产生算术错误陷阱。
4. 如果在移位指令中，指定的移位位数大于允许的最大移位位数，将产生陷阱。

当发生任何下列情形时，将产生陷阱：

1. 试图访问不对齐的数据字。
2. 试图从未实现的数据存储单元取数据。
3. 试图访问未实现的程序存储单元。
4. 试图从向量空间取指令。

注: 在 MAC 类指令中, 数据空间被分成 X 数据空间和 Y 数据空间, 未实现的 X 空间包括所有 Y 空间, 而未实现的 Y 空间包括所有 X 空间。

5. 执行 `BRA #literal` 指令或 `GOTO #literal` 指令, 其中 `literal` 是未实现的程序存储器地址。
6. 修改 `PC` 使其指向未实现的程序存储器地址后, 执行指令。通过将值装入堆栈并执行 `RETURN` 指令可以修改 `PC`。

堆栈错误陷阱:

当出现以下条件时，产生此陷阱：

1. 在堆栈指针中装入了一个大于堆栈限制的值（堆栈溢出）；堆栈限制值可由用户编程，存放在 **SPLIM** 寄存器中。
2. 堆栈指针中装入了一个小于 **0x0800** 的值（简单堆栈下溢）。

振荡器故障陷阱:

如果外部振荡器出现故障，器件使用备用的内部 RC 振荡器工作，就会产生振荡器故障陷阱。

5.3.2 硬陷阱和软陷阱

在同一周期内可能会产生多个陷阱（例如，把不对齐的字写入堆栈中的溢出地址）。这时，图 5-1 所示的固定优先级就会起作用；为了完全纠正错误，这可能需要用户检查是否有其他等待处理的陷阱。

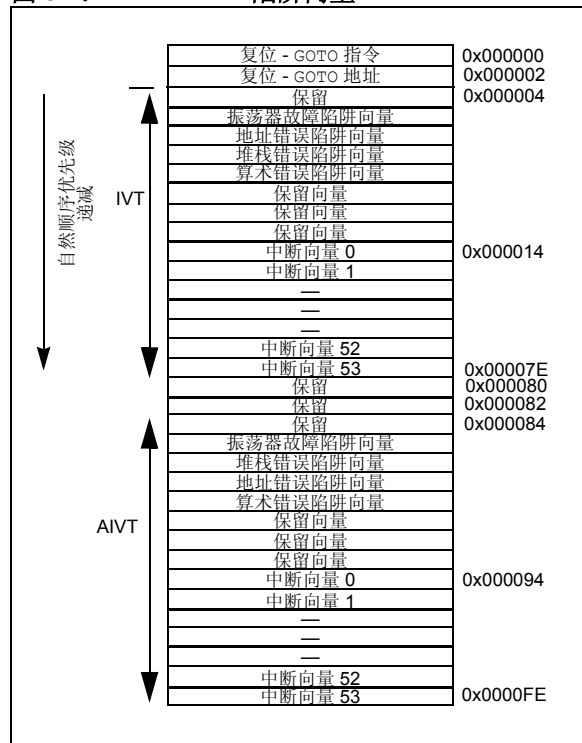
“软”陷阱包括优先级 8 到 11 的异常。算术错误陷阱（优先级 11）就属于这一类陷阱。

“硬”陷阱包括优先级 12 至 15 的异常。地址错误（优先级 12）、堆栈错误（优先级 13）和振荡器错误（优先级 14）陷阱就属于这一类。

每个硬陷阱产生时，在执行任何代码之前，必须先对它进行响应。在优先级较高的陷阱正在等待处理、被响应或正在处理过程中时，如果产生了较低优先级的陷阱，就会产生硬陷阱冲突。

器件在硬陷阱冲突时自动复位。复位发生时，把TRAPR状态位（RCON<15>）置1，从而在软件中可以检测到这一条件。

图 5-1: 陷阱向量



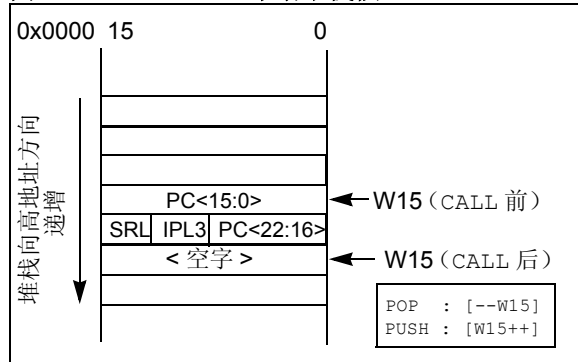
5.4 中断过程

在每个指令周期的开始，IFSx 寄存器都会采样全部的中断事件标志。IFSx 寄存器中标志位等于 1，表明有等待处理的中断请求（IRQ）。如果中断允许（IECx）寄存器中相应的位置 1，IRQ 将会导致中断产生。在 IRQ 采样后余下的指令周期中，将评估所有待处理中断请求的优先级。

如果有待处理的 IRQ，它的优先级比 IPL 位中的当前处理器优先级高，则处理器将被中断。

随即，处理器将把当前程序计数器和处理器状态寄存器的低字节（SRL）压入堆栈，如图 5-2 所示。状态寄存器低字节包含中断周期开始之前的处理器优先级。然后，处理器把中断的优先级装入状态寄存器。这将禁止所有低优先级的中断，直到中断服务程序（Interrupt Service Routine，ISR）结束为止。

图 5-2: 中断堆栈帧



- 注 1:** 用户能始终通过向 SR 写入一个新值以降低优先级。必须在降低处理器中断优先级之前，在中断服务程序中清除 IFSx 寄存器中的中断标志位以避免重复中断。
- 2:** 当处理中断时，IPL3 位（CORCON<3>）总是被清零。只有在进行陷阱处理时，该位才将会被置 1。

RETfie（从中断返回）指令将把程序计数器和状态寄存器内容弹出堆栈，将处理器恢复到中断处理之前的状态。

5.5 备用向量表

在程序存储器中，中断向量表（IVT）之后紧跟备用中断向量表（AIVT），如图 5-1 所示。INTCON2 寄存器中的 ALTIVT 位控制对备用向量表的访问。如果 ALTIVT 位置 1，所有中断和异常处理将使用备用向量而不是默认的向量。备用向量与默认向量的构成相同。AIVT 支持仿真和调试功能，它提供了一种不需要将中断向量再编程就可以在应用和支持环境之间切换的方法。此特性也支持运行时在不同应用之间切换以便评估各种软件算法。

如果不需要 AIVT，分配给 AIVT 的程序存储空间可以作其他用途。AIVT 不是受保护的区域，用户可以对其自由编程。

5.6 快速现场保护

可以使用影子寄存器来保护现场。SR 中的 DC、N、OV、Z 和 C 位以及寄存器 W0 到 W3 均提供有影子寄存器。影子寄存器深度仅为一级。只能使用 PUSH.S 和 POP.S 指令来访问影子寄存器。

当处理器转移到中断向量开始处理中断时，可以使用 PUSH.S 指令，把上述寄存器的当前值保存 to 它们对应的影子寄存器中去。

如果某个优先级的 ISR 使用 PUSH.S 和 POP.S 指令来快速保存现场，那么优先级更高的 ISR 就不应再使用这两条指令。如果优先级较高的 ISR 使用了快速现场保护，那么在优先级较低的中断处理期间，用户必须保存关键寄存器。

5.7 外部中断请求

中断控制器支持三种外部中断请求信号：INT0-INT2。这些输入是边沿敏感的；它们要求一个由低到高或由高到低的跳变以产生一个中断请求。INTCON2 寄存器具有 3 个位（INT0EP-INT2EP），用于选择边沿检测电路的极性。

5.8 从休眠和空闲模式唤醒

如果产生中断时处理器处于休眠或空闲模式，则可使用中断控制器将处理器从休眠或空闲模式唤醒。

如果中断控制器接收到一个具有足够优先级的已允许的中断请求，则会向处理器提交一个标准中断请求。同时，处理器会从休眠或空闲模式唤醒并开始执行处理中断请求所需的中断服务程序（ISR）。

dsPIC30F1010/202X

寄存器 5-1: INTCON1: 中断控制寄存器 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NSTDIS	OVAERR	OVBERR	COVAERR	COVBERR	OVATE	OVBTE	COVTE
bit 15							bit 8

R/W-0	R/W-0	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 = 上电复位时的值

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 = 陷阱由累加器 B 灾难性溢出引起
0 = 陷阱不是由累加器 B 灾难性溢出引起
- bit 10 **OVATE:** 累加器 A 溢出陷阱允许位
1 = 允许累加器 A 溢出陷阱
0 = 禁止陷阱
- bit 9 **OVBTE:** 累加器 B 溢出陷阱允许位
1 = 允许累加器 B 溢出陷阱
0 = 禁止陷阱
- bit 8 **COVTE:** 灾难性溢出陷阱允许位
1 = 允许累加器 A 或 B 的灾难性溢出陷阱
0 = 禁止陷阱
- bit 7 **SFTACERR:** 累加器移位错误状态位
1 = 算术错误陷阱由非法累加器移位引起
0 = 算术错误陷阱不是由非法累加器移位引起
- bit 6 **DIV0ERR:** 算术错误状态位
1 = 算术错误陷阱由被零除引起
0 = 算术错误陷阱不是由被零除引起
- bit 5 未实现: 读为 0
- bit 4 **MATHERR:** 算术错误状态位
1 = 已发生溢出陷阱
0 = 未发生溢出陷阱
- bit 3 **ADDRERR:** 地址错误陷阱状态位
1 = 已发生地址错误陷阱
0 = 未发生地址错误陷阱

寄存器 5-1: 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

dsPIC30F1010/202X

寄存器 5-2: **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 = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

- bit 15

ALTIVT: 备用中断向量表使能位
1 = 使用备用中断向量表
0 = 使用标准 (默认) 向量表
- bit 14

DISI: DISI 指令状态位
1 = 执行了 DISI 指令
0 = 没有执行 DISI 指令
- bit 13-3

未实现: 读为 0
- bit 2

INT2EP: 外部中断 2 边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断
- bit 1

INT1EP: 外部中断 1 边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断
- bit 0

INT0EP: 外部中断 0 边沿检测极性选择位
1 = 下降沿中断
0 = 上升沿中断

寄存器 5-3: IFS0: 中断标志状态寄存器 0

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF
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
T3IF	T2IF	OC2IF	—	T1IF	OC1IF	IC1IF	INT0IF
bit 7							bit 0

图注:

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

- bit 15 **未实现:** 读为 0
- bit 14 **MI2CIF:** I²C 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 13 **SI2CIF:** I²C 从事件中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 12 **NVMIF:** 非易失性存储器中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 11 **ADIF:** ADC 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 10 **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 9 **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 8 **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 7 **T3IF:** Timer3 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 6 **T2IF:** Timer2 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 5 **OC2IF:** 输出比较通道 2 中断标志状态位
1 = 产生了中断请求
0 = 未产生中断请求
- bit 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

dsPIC30F1010/202X

寄存器 5-3: IFS0: 中断标志状态寄存器 0 (续)

bit 2	OC1IF: 输出比较通道 1 中断标志状态位 1 = 产生了中断请求 0 = 未产生中断请求
bit 1	IC1IF: 输入捕捉通道 1 中断标志状态位 1 = 产生了中断请求 0 = 未产生中断请求
bit 0	INT0IF: 外部中断 0 标志状态位 1 = 产生了中断请求 0 = 未产生中断请求

寄存器 5-4: IFS1: 中断标志状态寄存器 1

R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0
AC3IF	AC2IF	AC1IF	—	CNIF	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	PWM4IF	PWM3IF	PWM2IF	PWM1IF	PSEMIF	INT2IF	INT1IF
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **AC3IF:** 模拟比较器 3 中断标志状态位
 1 = 产生了中断请求
 0 = 未产生中断请求
- bit 14 **AC2IF:** 模拟比较器 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 13 **AC1IF:** 模拟比较器 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 12 **未实现:** 读为 0
- bit 11 **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 10-7 **未实现:** 读为 0
- bit 6 **PWM4IF:** 脉宽调制发生器 4 中断标志状态位
 1 = 产生了中断请求
 0 = 未产生中断请求
- bit 5 **PWM3IF:** 脉宽调制发生器 3 中断标志状态位
 1 = 产生了中断请求
 0 = 未产生中断请求
- bit 4 **PWM2IF:** 脉宽调制发生器 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 3 **PWM1IF:** 脉宽调制发生器 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 2 **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 1 **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 0 **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

dsPIC30F1010/202X

寄存器 5-5: IFS2: 中断标志状态寄存器 2

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-00	R/W-0
—	—	—	—	—	ADCP5IF	ADCP4IF	ADCP3IF
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
ADCP2IF	ADCP1IF	ADCP0IF	—	—	—	—	AC4IF
bit 7						bit 0	

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

bit 15-11	未实现: 读为 0
bit 10	ADCP5IF: ADC 对 5 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 9	ADCP4IF: ADC 对 4 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 8	ADCP3IF: ADC 对 3 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 7	ADCP2IF: ADC 对 2 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 6	ADCP1IF: ADC 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 5	ADCP0IF: ADC 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 4-1	未实现: 读为 0
bit 0	AC4IF: 模拟比较器 4 中断标志状态位 1 = 产生了中断请求 0 = 未产生中断请求

寄存器 5-6: IEC0: 中断允许控制寄存器 0

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE
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
T3IE	T2IE	OC2IE	—	T1IE	OC1IE	IC1IE	INT0IE
bit 7							bit 0

图注:

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

- bit 15 **未实现:** 读为 0
- bit 14 **MI2CIE:** I²C 主控事件中中断允许位
 1 = 允许中断请求
 0 = 禁止中断请求
- bit 13 **SI2CIE:** I²C 从动事件中中断允许位
 1 = 允许中断请求
 0 = 禁止中断请求
- bit 12 **NVMIE:** 非易失性存储器中断允许位
 1 = 允许中断请求
 0 = 禁止中断请求
- bit 11 **ADIE:** ADC 转换完成中断允许位
 1 = 允许中断请求
 0 = 禁止中断请求
- bit 10 **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 9 **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 8 **SPI1IE:** SPI1 事件中中断允许位
 1 = 允许中断请求
 0 = 禁止中断请求
- bit 7 **T3IE:** Timer3 中断允许位
 1 = 允许中断请求
 0 = 禁止中断请求
- bit 6 **T2IE:** Timer2 中断允许位
 1 = 允许中断请求
 0 = 禁止中断请求
- bit 5 **OC2IE:** 输出比较通道 2 中断允许位
 1 = 允许中断请求
 0 = 禁止中断请求
- bit 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

dsPIC30F1010/202X

寄存器 5-6: IEC0: 中断允许控制寄存器 0 (续)

bit 2	OC1IE: 输出比较通道 1 中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 1	IC1IE: 输入捕捉通道 1 中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 0	INT0IE: 外部中断 0 允许位 1 = 允许中断请求 0 = 禁止中断请求

寄存器 5-7: IEC1: 中断允许控制寄存器 1

R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0
AC3IE	AC2IE	AC1IE	—	CNIE	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	PWM4IE	PWM3IE	PWM2IE	PWM1IE	PSEMIE	INT2IE	INT1IE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **AC3IE:** 模拟比较器 3 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 14 **AC2IE:** 模拟比较器 2 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 13 **AC1IE:** 模拟比较器 1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 12 **未实现:** 读为 0
- bit 11 **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 10-7 **未实现:** 读为 0
- bit 6 **PWM4IE:** 脉宽调制发生器 4 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 5 **PWM3IE:** 脉宽调制发生器 3 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 4 **PWM2IE:** 脉宽调制发生器 2 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 3 **PWM1IE:** 脉宽调制发生器 1 中断允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 2 **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 1 **INT2IE:** 外部中断 2 允许位
1 = 允许中断请求
0 = 禁止中断请求
- bit 0 **INT1IE:** 外部中断 1 允许位
1 = 允许中断请求
0 = 禁止中断请求

dsPIC30F1010/202X

寄存器 5-8: IEC2: 中断允许控制寄存器 2

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	ADCP5IE	ADCP4IE	ADCP3IE
bit 15					bit 8		

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
ADCP2IE	ADCP1IE	ADCP0IE	—	—	—	—	AC4IE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10 **ADCP5IE:** ADC 对 5 转换完成中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 9 **ADCP4IE:** ADC 对 4 转换完成中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 8 **ADCP3IE:** ADC 对 3 转换完成中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 7 **ADCP2IE:** ADC 对 2 转换完成中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 6 **ADCP1IE:** ADC 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 5 **ADCP0IE:** ADC 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 4-1 未实现: 读为 0

bit 0 **AC4IE:** 模拟比较器 4 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

寄存器 5-9: IPC0: 中断优先级控制寄存器 0

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T1IP<2:0>			—	OC1IP<2:0>		
bit 15				bit 8			

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC1IP<2:0>			—	INT0IP<2:0>		
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

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>**: 输入捕捉通道 1 中断优先级位
 111 = 中断优先级为 7 (最高优先级中断)
 .
 .
 .
 001 = 中断优先级为 1
 000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **IC1IP<2:0>**: 输入捕捉通道 1 中断优先级位
 111 = 中断优先级为 7 (最高优先级中断)
 .
 .
 .
 001 = 中断优先级为 1
 000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 **INT0IP<2:0>**: 外部中断 0 优先级位
 111 = 中断优先级为 7 (最高优先级中断)
 .
 .
 .
 001 = 中断优先级为 1
 000 = 禁止中断源

dsPIC30F1010/202X

寄存器 5-10: **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
—	T3IP<2:0>			—	T2IP<2:0>		
bit 15				bit 8			

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	OC2IP<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 = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15	未实现: 读为 0
bit 14-12	T3IP<2:0>: Timer3 中断优先级位 111 = 中断优先级为 7 (最高优先级中断) . . . 001 = 中断优先级为 1 000 = 禁止中断源
bit 11	未实现: 读为 0
bit 10-8	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 7	未实现: 读为 0
bit 6-4	OC2IP<2:0>: 输出比较通道 2 中断优先级位 111 = 中断优先级为 7 (最高优先级中断) . . . 001 = 中断优先级为 1 000 = 禁止中断源
bit 3-0	未实现: 读为 0

寄存器 5-11: 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
—	ADIP<2:0>			—	U1TXIP<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
—	U1RXIP<2:0>			—	SPI1IP<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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **ADIP<2:0>**: ADC 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 **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

bit 7 未实现: 读为 0

bit 6-4 **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 3 未实现: 读为 0

bit 2-0 **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

dsPIC30F1010/202X

寄存器 5-12: **IPC3: 中断优先级控制寄存器 3**

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	MI2CIP<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
—	SI2CIP<2:0>			—	NVMIP<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 = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15-11	未实现: 读为 0
bit 10-8	MI2CIP<2:0>: I ² C 主事件中断优先级位 111 = 中断优先级为 7 (最高优先级中断) • • • 001 = 中断优先级为 1 000 = 禁止中断源
bit 7	未实现: 读为 0
bit 6-4	SI2CIP<2:0>: I ² C 从事件中断优先级位 111 = 中断优先级为 7 (最高优先级中断) • • • 001 = 中断优先级为 1 000 = 禁止中断源
bit 3	未实现: 读为 0
bit 2-0	NVMIP<2:0>: 非易失性存储器中断优先级位 111 = 中断优先级为 7 (最高优先级中断) • • • 001 = 中断优先级为 1 000 = 禁止中断源

寄存器 5-13: 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
—	PWM1IP<2:0>			—	PSEMIP<2:0>		
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	INT2IP<2:0>			—	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **PWM1IP<2:0>**: PWM 发生器 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 11 未实现: 读为 0

bit 10-8 **PSEMIP<2:0>**: PWM 特殊事件匹配优先级位

111 = 中断优先级为 7 (最高优先级中断)

•
•
•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **INT2IP<2:0>**: 外部中断 2 优先级位

111 = 中断优先级为 7 (最高优先级中断)

•
•
•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 **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

dsPIC30F1010/202X

寄存器 5-14: **IPC5: 中断优先级控制寄存器 5**

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	PWM4IP<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
—	PWM3IP<2:0>			—	PWM2IP<2:0>		
bit 7				bit 0			

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

bit 15-11	未实现: 读为 0
bit 10-8	PWM4IP<2:0>: PWM 发生器 4 中断优先级位 111 = 中断优先级为 7 (最高优先级中断) • • • 001 = 中断优先级为 1 000 = 禁止中断源
bit 7	未实现: 读为 0
bit 6-4	PWM3IP<2:0>: PWM 发生器 3 中断优先级位 111 = 中断优先级为 7 (最高优先级中断) • • • 001 = 中断优先级为 1 000 = 禁止中断源
bit 3	未实现: 读为 0
bit 2-0	PWM2IP<2:0>: PWM 发生器 2 中断优先级位 111 = I 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 5-15: IPC6: 中断优先级控制寄存器 6

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	CNIP<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 = 上电复位时的值	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 11-0 未实现: 读为 0

dsPIC30F1010/202X

寄存器 5-16: **IPC7: 中断优先级控制寄存器 7**

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	AC3IP<2:0>			—	AC2IP<2:0>		
bit 15				bit 8			

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	AC1IP<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 = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15	未实现: 读为 0
bit 14-12	AC3IP<2:0>: 模拟比较器 3 中断优先级位 111 = 中断优先级为 7 (最高优先级中断) • • • 001 = 中断优先级为 1 000 = 禁止中断源
bit 11	未实现: 读为 0
bit 10-8	AC2IP<2:0>: 模拟比较器 2 中断优先级位 111 = 中断优先级为 7 (最高优先级中断) • • • 001 = 中断优先级为 1 000 = 禁止中断源
bit 7	未实现: 读为 0
bit 6-4	AC1IP<2:0>: 模拟比较器 1 中断优先级位 111 = 中断优先级为 7 (最高优先级中断) • • • 001 = 中断优先级为 1 000 = 禁止中断源
bit 3-0	未实现: 读为 0

寄存器 5-17: IPC8: 中断优先级控制寄存器 8

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

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	AC4IP<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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-3

未实现: 读为 0

bit 2-0

AC4IP<2:0>: 模拟比较器 4 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

-
-
-

001 = 中断优先级为 1

000 = 禁止中断源

dsPIC30F1010/202X

寄存器 5-18: **IPC9: 中断优先级控制寄存器 9**

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	ADCP2IP<2:0>			—	ADCP1IP<2:0>		
bit 15				bit 8			

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	ADCP0IP<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 = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15	未实现: 读为 0
bit 14-12	ADCP2IP<2:0>: ADC 对 2 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	ADCP1IP<2:0>: ADC 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 7	未实现: 读为 0
bit 6-4	ADCP0IP<2:0>: ADC 对 0 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 3-0	未实现: 读为 0

寄存器 5-19: IPC10: 中断优先级控制寄存器 10

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	ADCP5IP<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
—	ADCP4IP<2:0>			—	ADCP3IP<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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10 - 8 **ADCP5IP<2:0>**: ADC 对 5 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 7 未实现: 读为 0

bit 6-4 **ADCP4IP<2:0>**: ADC 对 4 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 **ADCP3IP<2:0>**: ADC 对 3 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

dsPIC30F1010/202X

寄存器 5-20: 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 = 上电复位时的值		1 = 置 1		0 = 清零		x = 未知	

bit 15-12	未实现: 读为 0
bit 11-8	ILR: 新的 CPU 中断优先级位 1111 = CPU 中断优先级为 15 • • • 0001 = CPU 中断优先级为 1 0000 = CPU 中断优先级为 0
bit 7	未实现: 读为 0
bit 6-0	VECNUM: 待处理中断向量编号位 0111111 = 待处理中断向量的编号为 135 • • • 0000001 = 待处理中断向量的编号为 9 0000000 = 待处理中断向量的编号为 8

表 5-2: 中断控制器寄存器映射

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态
INTCON1	0080	NSTDIS	OVAERR	OVBERR	COVAERR	COVBERR	OVATE	OVATE	COVTE	SFTACERR	DIV0ERR	—	MATHERR	ADDRERR	STKERR	OSCFail	—	0000 0000 0000 0000
INTCON2	0082	ALTVT	DISI	—	—	—	U1TXIF	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000 0000 0000 0000
IFS0	0084	—	M2CIF	S2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	—	T1IF	OC1IF	INT0IF	INT0IF	0000 0000 0000 0000
IFS1	0086	AC3IF	AC2IF	AC1IF	—	CNIF	—	—	—	—	PWM4IF	PWM3IF	PWM2IF	PWM1IF	PSEMIF	INT2IF	INT1IF	0000 0000 0000 0000
IFS2	0088	—	—	—	—	—	ADCP5IF	ADCP4IF	ADCP3IF	ADCP2IF	ADCP1IF	ADCP0IF	—	—	—	—	AC4IF	0000 0000 0000 0000
IEC0	0094	—	M2CIE	S2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	—	T1IE	OC1IE	INT0IE	INT0IE	0000 0000 0000 0000
IEC1	0096	AC3IE	AC2IE	AC1IE	—	CNIE	—	—	—	—	PWM4IE	PWM3IE	PWM2IE	PWM1IE	PSEMIE	INT2IE	INT1IE	0000 0000 0000 0000
IEC2	0098	—	—	—	—	—	ADCP5IE	ADCP4IE	ADCP3IE	ADCP2IE	ADCP1IE	ADCP0IE	—	—	—	—	AC4IE	0000 0000 0000 0000
IPC0	00A4	—	—	T1IP<2:0>	—	—	OC1IP<2:0>	—	—	—	—	IC1IP<2:0>	—	—	INT0IP<2:0>	—	—	0100 0100 0100 0100
IPC1	00A6	—	—	T31P<2:0>	—	—	T2IP<2:0>	—	—	—	—	OC2IP<2:0>	—	—	—	—	—	0100 0100 0100 0000
IPC2	00A8	—	—	ADIP<2:0>	—	—	U1TXIP<2:0>	—	—	—	—	U1RXIP<2:0>	—	—	—	SPIIP<2:0>	—	0100 0100 0100 0100
IPC3	00AA	—	—	—	—	—	M2CIP<2:0>	—	—	—	—	S2CIP<2:0>	—	—	—	NMIP<2:0>	—	0000 0100 0100 0100
IPC4	00AC	—	—	PWM1IP<2:0>	—	—	PSEMIP<2:0>	—	—	—	—	INT2IP<2:0>	—	—	—	INT1IP<2:0>	—	0100 0100 0100 0100
IPC5	00AE	—	—	—	—	—	PWM4IP<2:0>	—	—	—	—	PWM3IP<2:0>	—	—	—	PWM2IP<2:0>	—	0000 0100 0100 0100
IPC6	00B0	—	—	CNIP<2:0>	—	—	—	—	—	—	—	—	—	—	—	—	—	0100 0000 0000 0000
IPC7	00B2	—	—	AC3IP<2:0>	—	—	AC2IP<2:0>	—	—	—	—	AC1IP<2:0>	—	—	—	—	—	0100 0100 0100 0000
IPC8	00B4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AC4IP<2:0>	—	0000 0000 0000 0100
IPC9	00B6	—	—	ADCP2IP<2:0>	—	—	ADCP1IP<2:0>	—	—	—	—	ADCP0IP<2:0>	—	—	—	—	—	0100 0100 0100 0000
IPC10	00B8	—	—	—	—	—	ADCP5IP<2:0>	—	—	—	—	ADCP4IP<2:0>	—	—	—	ADCP3IP<2:0>	—	0000 0100 0100 0100
INTTREG	00E0	—	—	—	—	—	ILR<3:0>	VECNUM<6:0>				—	—	—	—	—	—	0000 0000 0000 0000

注: 有关寄存器位字段的描述, 请参见《dsPIC30F/33F 系列参考手册》(DS70157B_CN)。

注:

6.0 I/O 端口

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

所有的器件引脚（除 VDD、VSS、MCLR 和 OSC1/CLKI 之外）均为外设与并行 I/O 端口所共用。

所有 I/O 输入端口都是施密特触发器输入，以增强抗干扰性。

6.1 并行 I/O (PIO) 端口

如果外设使能，并且外设正在使用其关联引脚时，引脚将不再作为通用 I/O 引脚使用。这时，可以读 I/O 引脚，但是对应并行端口位的输出驱动器将被禁止。如果外设使能，但外设不在使用某引脚，则该引脚可以被端口驱动。

所有端口引脚都有三个与端口引脚的操作直接相关的寄存器。数据方向寄存器 (TRISx) 决定引脚是输入还是输出。如果数据方向位是 1，则引脚是输入。复位后，

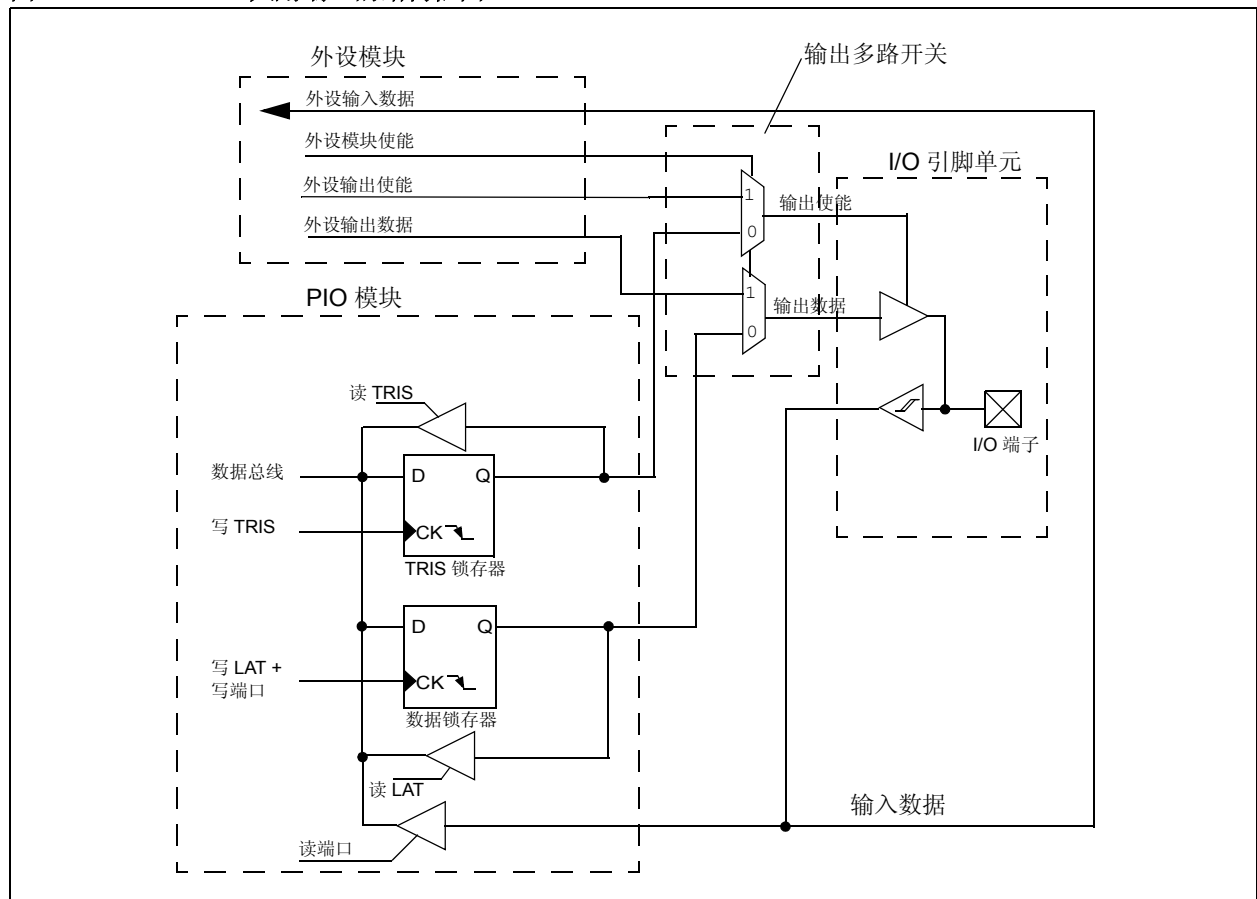
所有端口引脚均定义为输入。读锁存器 (LATx)，读的是锁存值；向锁存器写入，写的是锁存器 (LATx)。读端口 (PORTx)，读的是端口引脚值，写端口引脚，写的是锁存器 (LATx)。

任何位及其关联的数据和控制寄存器，如果对于特定器件而言是无效的话，则将被禁止。这意味着，相应的 LATx 和 TRISx 寄存器以及端口引脚，都将读为零。

当端口引脚与另一个外设共用或与只定义为输入的功能共用时，它将仍然被视为专用端口，因为没有任何其他与之竞争的输出资源。

与外设共用引脚的并行 I/O (PIO) 端口一般屈从于外设。外设的输出缓冲数据和控制信号提供给一对多路开关。多路开关选择外设还是相关的端口将拥有 I/O 引脚输出数据和控制信号的所有权。图 6-1 显示了端口如何与其他外设共用，以及与外设连接的相关 I/O 引脚单元。表 6-1 和表 6-2 分别给出了 dsPIC30F1010/2020 器件中从 PORTA 到 PORTF 的共用端口的寄存器的格式，以及 dsPIC30F2023 器件中从 PORTA 到 PORTG 的共用端口的寄存器的格式。

图 6-1: 共用端口的结构框图



6.2 配置模拟端口引脚

使用 ADPCFG 和 TRIS 寄存器控制 A/D 端口引脚的操作。对于要用作模拟输入的端口引脚，必须将其相应的 TRIS 位置 1（输入）。如果 TRIS 位清零（输出），则将转换数字输出电平（VOH 或 VOL）。

读取端口寄存器时，所有配置为模拟输入通道的引脚，都将读为零（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。当将模拟电压加在任何定义为数字输入的引脚（包括 ANx 引脚）上时，可能导致输入缓冲器消耗的电流超出器件规范中规定的值。

6.2.1 I/O 端口读 / 写时序

在端口方向变化或端口写操作以及同一端口的读操作之间需要一个指令周期的等待时间。通常情况下插入一条 NOP 指令。

例 6-1: 端口读 / 写示例

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

6.3 输入电平变化通知

I/O端口的输入电平变化通知模块使 dsPIC30F1010/202X 器件能够产生中断请求并传递给处理器，以响应所选取输入引脚上的电平变化。即便是在休眠模式下（时钟被禁止）时，该模块也能检测输入的电平变化。要在输入电平变化时产生中断请求，可以选择（使能）的外部信号有 8 个（CN0 到 CN7）。

有两个与 CN 模块相关的控制寄存器。CNEN1 包含每个 CN 输入引脚的中断允许控制位（CNxIE）。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。弱上拉电路作为连接到该引脚的电流源，当连接了按钮或键盘设备时，不再需要使用外部电阻。使用包含每个 CN 引脚的弱上拉使能位（CNxPUE）的 CNPU1 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： 只要端口引脚被配置为数字输出引脚，就应始终将电平变化通知引脚上的弱上拉电路禁止。

表 6-1: dsPIC30F1010/2020 端口寄存器映射

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态
TRISA	02C0	—	—	—	—	—	—	TRISA9	—	—	—	—	—	—	—	—	—	0000 0010 0000 0000
PORTA	02C2	—	—	—	—	—	—	RA9	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATA	02C4	—	—	—	—	—	—	LAT9	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
TRISB	02C6	—	—	—	—	—	—	—	—	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0000 0000 0011 1111
PORTB	02C8	—	—	—	—	—	—	—	—	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	0000 0000 0000 0000
LATB	02CA	—	—	—	—	—	—	—	—	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000 0000 0000 0000
TRISD	02D2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRISD0	0000 0000 0000 0001
PORTD	02D4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RD0	0000 0000 0000 0000
LATD	02D6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LATD0	0000 0000 0000 0000
TRISE	02D8	—	—	—	—	—	—	—	—	TRSE7	TRSE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	0000 0000 1111 1111
PORTE	02DA	—	—	—	—	—	—	—	—	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	0000 0000 0000 0000
LATE	02DC	—	—	—	—	—	—	—	—	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	0000 0000 0000 0000
TRISF	02DE	—	—	—	—	—	—	—	TRISF8	TRISF7	TRISF6	—	—	—	—	—	—	0000 0001 1100 0000
PORTF	02E0	—	—	—	—	—	—	—	RF8	RF7	RF6	—	—	—	—	—	—	0000 0000 0000 0000
LATF	02E2	—	—	—	—	—	—	—	LATF8	LATF7	LATF6	—	—	—	—	—	—	0000 0000 0000 0000

注: 有关寄存器位字段的描述, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

表 6-2: dsPIC30F2023 端口寄存器映射

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态
TRISA	02C0	—	—	—	—	TRISA11	TRISA10	TRIS9	TRISA8	—	—	—	—	—	—	—	—	0000 1111 0000 0000
PORTA	02C2	—	—	—	—	RA11	RA10	RA9	RA8	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATA	02C4	—	—	—	—	LATA11	LATA10	LATA9	LATA8	—	—	—	—	—	—	—	—	0000 0000 0000 0000
TRISB	02C6	—	—	—	—	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRIS6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0000 1111 1111 1111
PORTB	02C8	—	—	—	—	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	0000 0000 0000 0000
LATB	02CA	—	—	—	—	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000 0000 0000 0000
TRISD	02D2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRISD1	TRISD0	0000 0000 0000 0011
PORTD	02D4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RD1	RD0	0000 0000 0000 0000
LATD	02D6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LATD1	LATD0	0000 0000 0000 0000
TRISE	02D8	—	—	—	—	—	—	—	—	TRSE7	TRSE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	0000 0000 1111 1111
PORTE	02DA	—	—	—	—	—	—	—	—	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	0000 0000 0000 0000
LATE	02DC	—	—	—	—	—	—	—	—	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	0000 0000 0000 0000
TRISF	02DE	TRISF15	TRISF14	—	—	—	—	—	TRISF8	TRISF7	TRISF6	—	—	TRISF3	TRISF ₂	—	—	1100 0001 1100 1100
PORTF	02E0	RF15	RF14	—	—	—	—	—	RF8	RF7	RF6	—	—	RF3	RF2	—	—	0000 0000 0000 0000
LATF	02E2	LATF15	LATF14	—	—	—	—	—	LATF8	LATF7	LATF6	—	—	LATF3	LATF2	—	—	0000 0000 0000 0000
TRISG	02E4	—	—	—	—	—	—	—	—	—	—	—	—	TRISG3	TRISG ₂	—	—	0000 0000 0000 1100
PORTG	02E6	—	—	—	—	—	—	—	—	—	—	—	—	RG3	RG2	—	—	0000 0000 0000 0000
LATG	02E8	—	—	—	—	—	—	—	—	—	—	—	—	LATG3	LATG2	—	—	0000 0000 0000 0000

注: 有关寄存器位字段的描述, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

表 6-3: dsPIC30F1010/202X 输入电平变化通知寄存器映射

SFR Name	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	Reset State
CNEN1	0060	—	—	—	—	—	—	—	—	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000 0000 0000 0000
CNPU1	0064	—	—	—	—	—	—	—	—	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000 0000 0000 0000

注: 有关寄存器位字段的描述, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

7.0 闪存程序存储器

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。有关器件指令集和编程的更多信息，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

dsPIC30F 系列器件包含用以执行用户代码的内部闪存程序存储器。用户可使用以下两种方法对该存储器编程：

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

7.1 在线串行编程 (ICSP)

可以在最终的应用电路中对 dsPIC30F 器件进行串行编程。只需要 5 根线即可实现这一操作，其中编程时钟线和编程数据线（分别称为 PGC 和 PGD）各一根，其余 3 根分别是电源线 (VDD)、接地线 (Vss) 和主复位线 (MCLR)。这使用户可在制造电路板时使用未编程器件，而仅在产品交付前才对数字信号控制器进行编程。这样还能将最新版本的固件或者定制固件烧写到器件中。

7.2 运行时自编程 (RTSP)

运行时自编程 (RTSP) 使用 TBLRD (表读) 和 TBLWT (表写) 指令实现。

使用 RTSP，用户可以一次擦除 32 个指令字 (96 字节) 的程序存储单元，一次可写入 32 个指令字 (96 字节) 的程序存储数据。

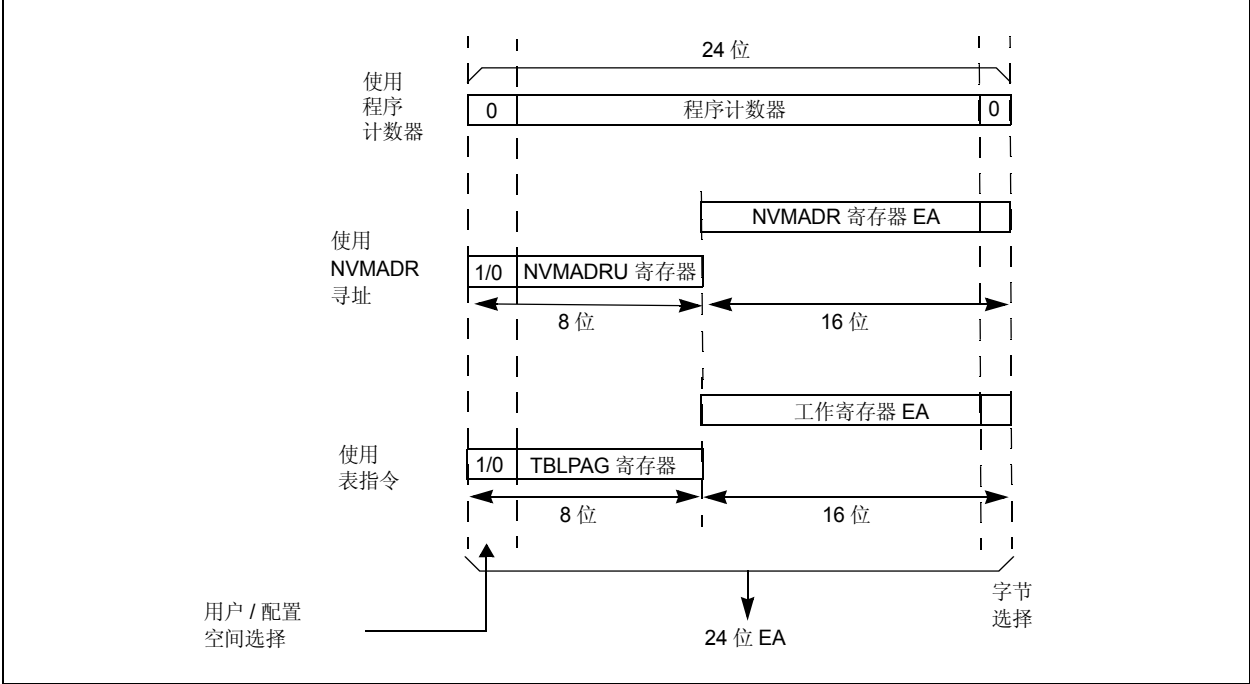
7.3 表指令操作概述

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

使用 TBLPAG 寄存器的 bit<7:0> 和来自表指令中指定的 W 寄存器的有效地址 (Effective Address, EA) 构成 24 位的程序存储器地址，如图 7-1 所示。

图 7-1: 表和 NVM 寄存器寻址



7.4 RTSP 操作

dsPIC30F 闪存程序存储器以行 (row) 和板 (panel) 为单位构成。每行由 32 条指令或 96 字节组成。每板由 128 行或 4K x 24 个指令字组成。RTSP 允许用户每次擦除一行 (32 条指令)、每次编程 32 条指令。RTSP 可用于编程多个程序存储器板, 但必须在每个板的边界处更改表指针。

程序存储器的每板都包含写锁存器, 它能够保存 32 个指令字的编程数据。在实际编程操作前, 等待写入的数据必须先装入板的写锁存器。要写入板的数据按顺序装载到写锁存器中: 指令 0, 指令 1, 依此类推。装载的指令字必须始终来自 32 位指令字的地址边界。

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, 装载写锁存器。通过设置 NVMCON 寄存器中的特定位启动编程。装载 32 条指令需要 32 条 TBLWTL 和 32 条 TBLWTH 指令。

由于只需要写表锁存器, 所以所有表写操作都是单字写入 (2 个指令周期)。每编程一行需要一个编程周期。

在整个 VDD 范围内, 正常操作期间, 闪存程序存储器是可读、可写且可擦除的。

7.5 控制寄存器

有四个 SFR 可用于读写闪存程序存储器, 它们是:

- NVMCON
- NVMADR
- NVMADRU
- NVMKEY

7.5.1 NVMCON 寄存器

NVMCON 寄存器控制要擦除的存储块、要编程的存储器类型以及编程周期的启动。

7.5.2 NVMADR 寄存器

NVMADR 寄存器用于存放有效地址的两个低字节。它捕捉已被执行的上一条表指令的 EA<15:0>, 选择要写入的行。

7.5.3 NVMADRU 寄存器

NVMADRU 寄存器用来保持有效地址的高字节。它捕捉已执行的上一条表指令的 EA<23:16>。

7.5.4 NVMKEY 寄存器

NVMKEY 是一个用于写保护的只写寄存器。要启动编程或擦除过程, 用户必须把 0x55 和 0xAA 连续写入 NVMKEY 寄存器。更多详细信息请参见第 7.6 节“编程操作”。

注: 用户也可直接写 NVMADR 和 NVMADRU 寄存器, 以指定待擦除或编程的程序存储器的地址。

7.6 编程操作

在 RTSP 模式下对内部闪存进行编程或擦除需要一个完整的编程序列。编程操作持续时间的标称值为 2 ms，在此期间处理器暂停，等待操作完成。将 WR 位 (NVMCON<15>) 置 1 启动编程操作，操作结束后将自动清零 WR 位。

7.6.1 闪存程序存储器的编程算法

用户可以一次擦除或编程闪存程序存储器的一行。一般步骤如下：

1. 读一行闪存程序存储器（32 个指令字），把读出的数据作为数据“镜像”保存到数据 RAM 中。
2. 用期望的新数据更新数据镜像。
3. 擦除闪存程序存储器。
 - a) 设置 NVMCON 寄存器以允许对闪存程序存储器进行多字擦除，并将 WREN 位置 1。
 - b) 把要擦除的行地址写入寄存器 NVMADRU/ NVMDR。
 - c) 把“55”写入 NVMKEY。
 - d) 把“AA”写入 NVMKEY。
 - e) 将 WR 位置 1。这将开始擦除周期。
 - f) 在擦除周期中 CPU 将暂停。
 - g) 当擦除周期结束时 WR 位会被清零。

4. 从数据 RAM “镜像”中把 32 个指令字的数据写入闪存程序存储器写锁存器。
5. 将 32 个指令字写入闪存程序存储器。
 - a) 设置 NVMCON 寄存器以允许对闪存程序存储器进行多字编程，并将 WREN 位置 1。
 - b) 把“55”写入 NVMKEY。
 - c) 把“AA”写入 NVMKEY。
 - d) 将 WR 位置 1。这将开始编程周期。
 - e) 在编程周期中 CPU 将暂停。
 - f) 当编程周期结束时，WR 位会被硬件清零。
6. 根据需要重复步骤 1 到 5，对所需的闪存程序存储器进行编程。

7.6.2 擦除程序存储器的一行

例 7-1 所示的代码序列可以用来擦除程序存储器的一行（32 个指令字）。

例 7-1: 擦除程序存储器的一行

```
; Setup NVMCON for erase operation, multi word write
; program memory selected, and writes enabled
      MOV    #0x4041,W0                ;
      MOV    W0,NVMCON                 ; Init NVMCON SFR
; Init pointer to row to be ERASED
      MOV    #tblpage(PROG_ADDR),W0    ;
      MOV    W0,NVMADRU                ; Initialize PM Page Boundary SFR
      MOV    #tbloffset(PROG_ADDR),W0  ; Intialize in-page EA[15:0] pointer
      MOV    W0, NVMDR                 ; Intialize NVMADR SFR
      DISI   #5                        ; Block all interrupts with priority <7
                                           ; for next 5 instructions

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

7.6.3 装载写锁存器

例 7-2 给出了可用于装载 96 字节的写锁存器的指令序列。需要 32 条 TBLWTL 指令和 32 条 TBLWTH 指令来装载由表指针选定的写锁存器。

例 7-2: 装载写锁存器

```
;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
    .
    .
    .
; 31st_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
```

注：在例 7-2 中，W3 高字节的内容将不起作用。

7.6.4 启动编程序列

出于保护的目的，必须使用 NVMKEY 的写启动序列，以便允许进行任何擦除或编程操作。在执行编程命令后，用户必须等待一段时间（编程时间），直至编程完成。编程开始后紧跟的两条指令应该为 NOP 指令。

例 7-3: 启动编程序列

```
DISI    #5                          ; Block all interrupts with priority <7
                                              ; for next 5 instructions

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

表 7-1: NVM 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态
NVMCON	0760	WR	WREN	WRERR	—	—	—	—	TWRI	—	—	—	—	—	—	—	—	0000 0000 0000 0000
NVMADR	0762	NVMADR<15:0>																0000 0000 0000 0000
NVMADRU	0764	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
NVMKEY	0766	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000

图注: u = 未初始化的位
注: 有关寄存器位字段的描述请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

注:

8.0 TIMER1 模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

本节介绍了 16 位通用 Timer1 模块以及相关的工作模式。图 8-1 所示为 16 位 Timer1 模块的简化框图。

注：Timer1 是 A 类定时器。请参见本文档第 21.0 节“电气特性”中 A 类定时器的规范。

下面各小节将详细说明定时器的工作模式，包括其设置、控制寄存器以及相应的框图。

Timer1 模块是 16 位的定时器，可以用作实时时钟的时间计数器，或者用作自由运行的时段定时器 / 计数器。16 位定时器有下列模式：

- 16 位定时器
- 16 位同步计数器
- 16 位异步计数器

而且，支持下列操作特征：

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

通过设定 16 位 SFR T1CON 中的相应位来决定这些工作模式。图 8-1 所示为 16 位定时器模块的框图。

16 位定时器模式：在 16 位定时器模式下，定时器将在每个指令周期递增，直到等于周期寄存器 PR1 中预先装入的值，然后复位至 0，继续计数。

当 CPU 进入空闲模式时，定时器将停止递增，除非 TSIDL (T1CON<13>) 位 = 0。如果 TSIDL = 1，则定时器模块逻辑将在 CPU 空闲模式结束后继续递增。

16 位同步计数器模式：在 16 位同步计数器模式下，定时器将在外部时钟信号的上升沿递增，外部时钟与内部时钟同步。定时器计数，直到等于 PR1 中预先装入的值，然后复位至 0，继续计数。

当 CPU 进入空闲模式时，定时器将停止递增，除非对应的 TSIDL 位 = 0。如果 TSIDL = 1，则定时器模块逻辑将在 CPU 空闲模式结束后继续递增。

16 位异步计数器模式：在 16 位异步计数器模式下，定时器将在外部时钟信号的上升沿递增。定时器计数，直到等于 PR1 中预先装入的值，然后复位至 0，继续计数。

当定时器配置为异步工作模式时，且 CPU 进入空闲模式的话，如果 TSIDL = 1，则定时器将停止递增。

8.4 定时器中断

16 位定时器具有在周期匹配时产生中断的能力。当定时器计数与周期寄存器相匹配时，**T1IF** 位将置 1，如果允许中断的话，将产生中断。**T1IF** 位必须在软件中清零。定时器中断标志 **T1IF** 位于中断控制器的 **IFS0** 控制寄存器中。

当使能了门控时间累加模式时，还将在门控信号的下降沿（累加周期的末尾）产生中断。

通过相应的定时器中断允许位 **T1IE** 来允许定时器中断。定时器中断允许位位于中断控制器的 **IEC0** 控制寄存器中。

表 8-1: TIMER1 寄存器映射

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态
TMR1	0100	Timer 1 寄存器																uuuu uuuu uuuu uuuu
PR1	0102	周期寄存器 1																1111 1111 1111 1111
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>	—	—	TSYNC	TCS	—	0000 0000 0000 0000

图注: u = 未初始化的位
有关寄存器位字段的描述, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

9.0 TIMER2/3 模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参阅《dsPIC30F 系列参考手册》(DS70046E_CN)。

本节介绍了 32 位通用定时器模块 (Timer2/3) 以及相关的工作模式。图 9-1 给出了 32 位 Timer2/3 模块的简化框图。图 9-2 和图 9-3 分别为将 Timer2/3 配置成的两个独立 16 位定时器的 Timer2 和 Timer3 的框图。

注：dsPIC30F1010 器件不具备 Timer3。Timer2 是 B 类定时器，Timer3 是 C 类定时器。请参见本文档第 21.0 节“电气特性”中相应类型定时器的规范。

Timer2/3 模块是 32 位定时器（可配置为两个 16 位定时器），具有可选择的工作模式。这些定时器为其他外设模块所使用，例如：

- 输入捕捉
- 输出比较 / 简单 PWM

对于定时器的工作模式，下面的各小节将详细说明其设置和控制寄存器及相应的框图。

32 位定时器有下列模式：

- 两个独立的 16 位定时器 (Timer2 和 Timer3)，支持所有 16 位工作模式（异步计数器模式除外）
- 一个 32 位定时器
- 一个 32 位同步计数器

此外，支持如下工作特性：

- ADC 事件触发信号
- 定时器门控操作
- 可选择的预分频比设定
- 空闲和休眠模式下的定时器操作
- 在 32 位周期寄存器匹配时产生中断

通过设定 16 位 T2CON 和 T3CON SFR 中的相应位来决定这些工作模式。

对于 32 位定时器 / 计数器操作，Timer2 是 32 位定时器的低位字 (LS 字)，而 Timer3 是高位字 (MS 字)。

注：对于 32 位定时器操作，T3CON 控制位将被忽略。设置和控制只使用 T2CON 控制位。对于 32 位定时器模块，使用的是 Timer2 时钟和门控输入，但是，中断产生由 Timer3 中断标志位 (T3IF) 反映，且中断通过 Timer3 中断允许位 (T3IE) 来允许。

16 位定时器模式：在 16 位模式下，Timer2 和 Timer3 可以配置为两个独立的 16 位定时器。每个定时器均可设置为 16 位定时器模式或 16 位同步计数器模式。这两种工作模式的细节，请参见第 8.0 节“Timer1 模块”。

Timer2 和 Timer3 之间惟一的功能差异是，Timer2 提供了时钟预分频器输出同步，这对于高频外部时钟输入很有用。

32 位定时器模式：在 32 位定时器模式下，定时器在每个指令周期递增，一直递增到与预先装入组合 32 位周期寄存器 PR3/PR2 中的值匹配，然后复位至 0，继续计数。

对于 Timer2/Timer3 对的同步 32 位读操作，读低位字 (TMR2 寄存器) 将导致高位字 (msw) 被读取并锁存到一个 16 位保持寄存器 (称作 TMR3HLD)。

对于同步 32 位写操作，必须首先写入保持寄存器 (TMR3HLD)。如果后续操作是写 TMR2 寄存器的话，则 TMR3HLD 的内容将传送并锁存到 32 位定时器 (TMR3) 的 MSB 中。

32 位同步计数器模式：在 32 位同步计数器模式下，定时器将在外部时钟信号的上升沿递增，外部时钟信号与内部时钟同步。定时器将递增计数，直到与预先装入组合的 32 位周期寄存器 PR3/PR2 中的值匹配，然后复位至 0，并继续计数。

当定时器配置为同步计数器工作模式时，当 CPU 进入空闲模式时，定时器将停止递增，除非 TSIDL (T2CON<13>) 位 = 0。如果 TSIDL = 1，则定时器模块逻辑将继续递增，直到 CPU 空闲模式终止。

图 9-1: 32 位 TIMER2/3 框图

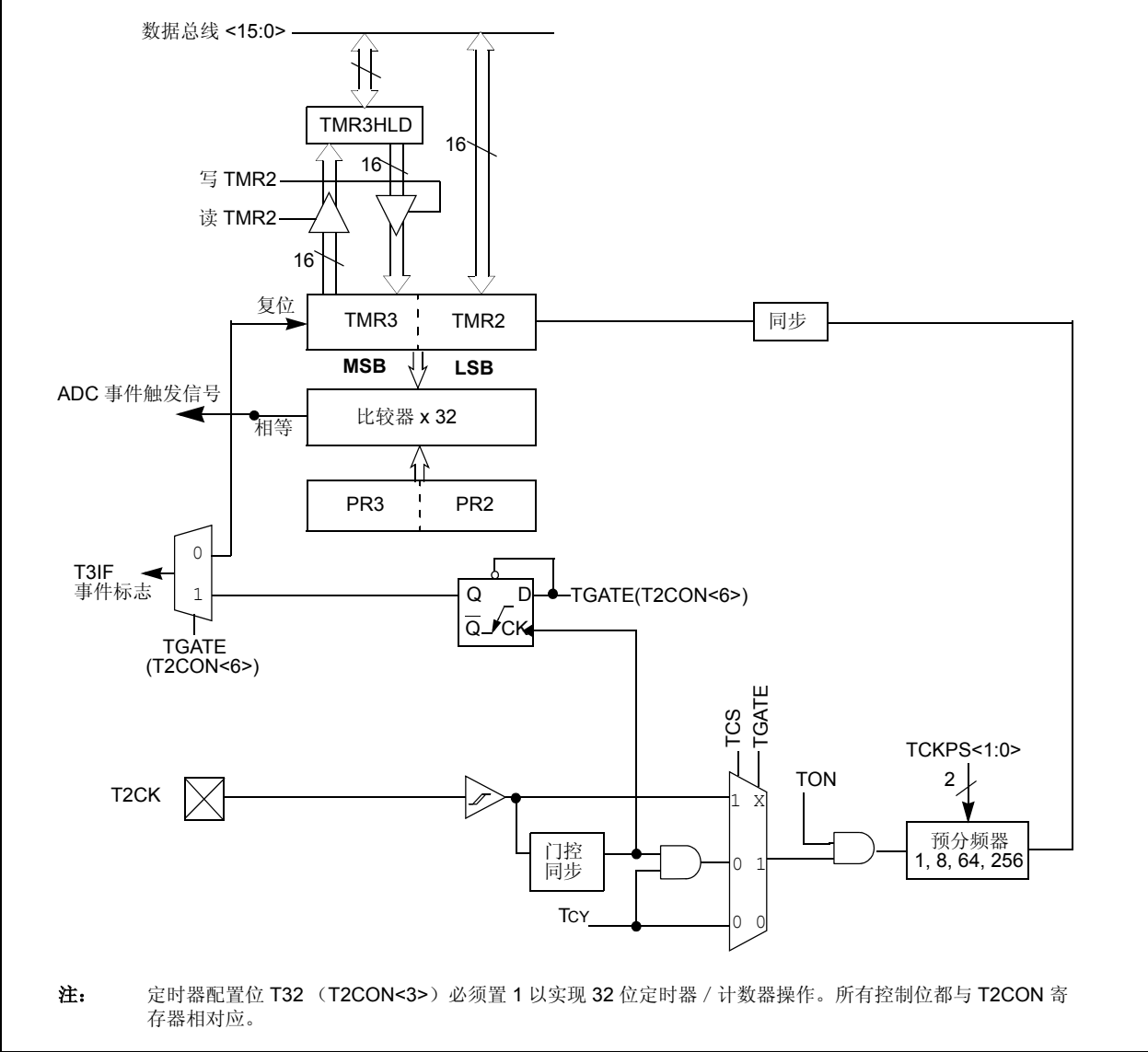


图 9-2: 16 位 TIMER2 框图

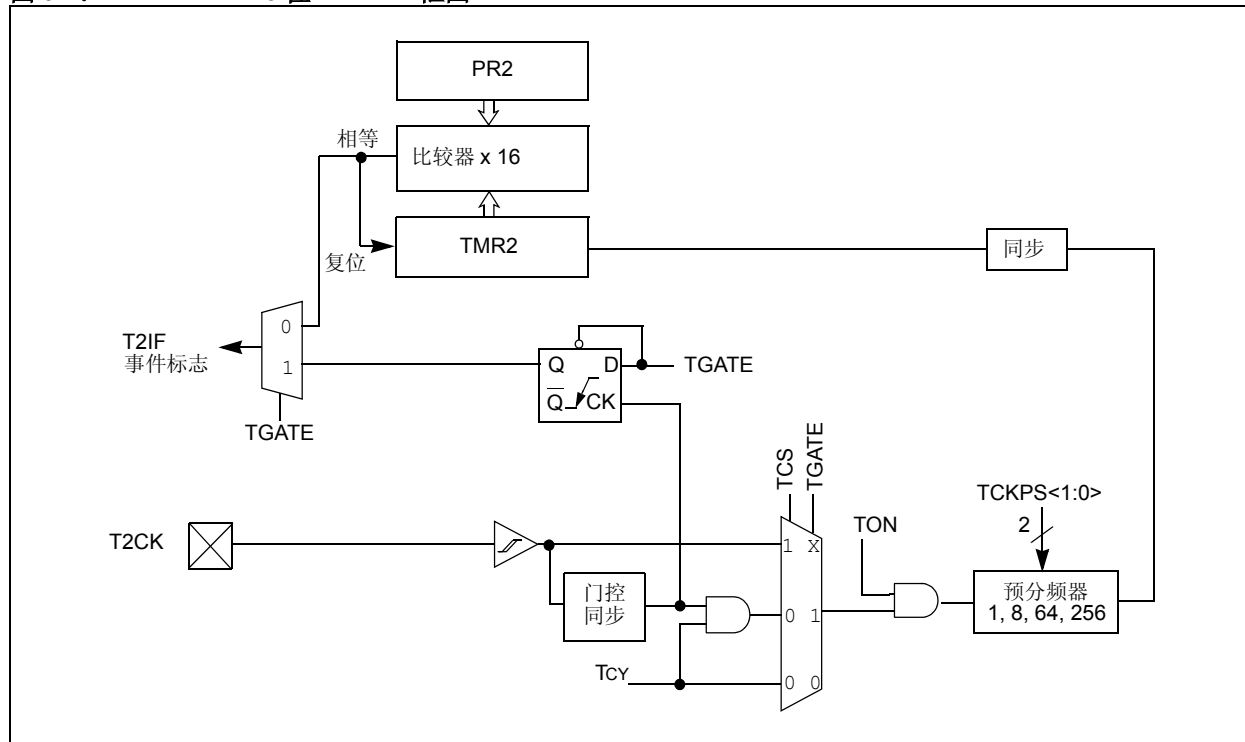
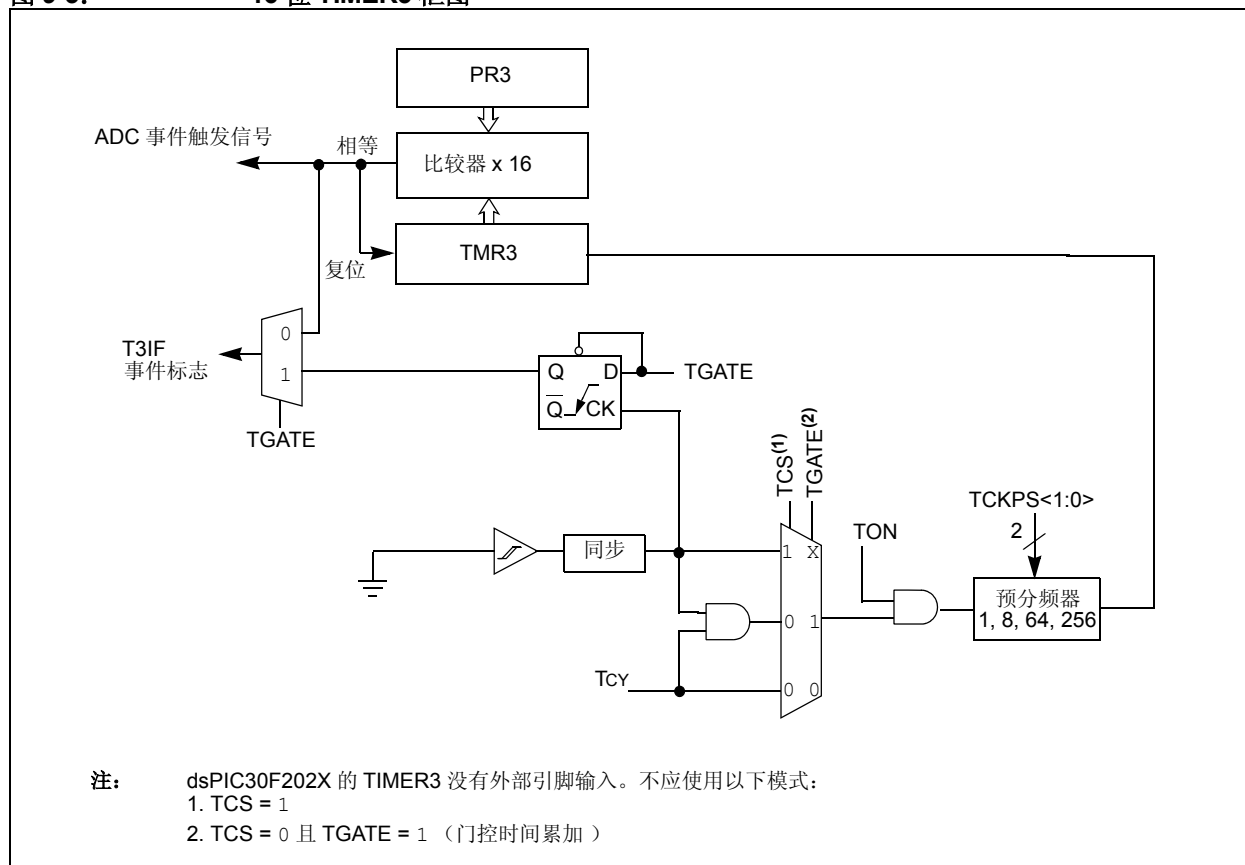


图 9-3: 16 位 TIMER3 框图



9.1 定时器门控操作

32 位定时器可被置于门控时间累加模式。该模式使得当门控输入信号（T2CK 引脚）为高电平时，内部 Tcy 能够递增相应的定时器。要使能该模式，必须置 1 控制位 TGATE（T2CON<6>）。在此模式下，Timer2 是发起时钟源。对于 Timer3，TGATE 设定被忽略。必须使能定时器（TON = 1），且定时器时钟源设置为内部时钟源（TCS = 0）。

外部信号的下降沿将终止计数操作，但不会复位定时器。要让定时器从零开始计数，用户必须先将其复位。

9.2 ADC 事件触发信号

当 32 位定时器（TMR3/TMR2）与 32 位组合周期寄存器（PR3/PR2）匹配时，Timer3 将生成 ADC 特殊事件触发信号。

9.3 定时器预分频器

定时器的输入时钟（Fosc/2 或外部时钟），有 1:1、1:8、1:64 或 1:256 的预分频选择；可通过控制位 TCKPS<1:0>（T2CON<5:4> 和 T3CON<5:4>）来选择预分频比。对于 32 位定时器操作，发起时钟源的是 Timer2。在此模式下不能对 Timer3 的预分频器进行操作。当出现下列事件之一时，预分频器的计数器将清零：

- 写 TMR2/TMR3 寄存器
- 清零任一 TON 位（T2CON<15> 或 T3CON<15>）
- 器件复位，如 POR

但是，如果定时器被禁止（TON = 0），那么 Timer2 预分频器将不会被复位，因为预分频器的时钟停止了。

当写 T2CON/T3CON 时，TMR2/TMR3 不会被清零。

9.4 休眠模式下的定时器操作

在 CPU 休眠模式期间，定时器不工作，因为内部时钟被禁止。

9.5 定时器中断

32 位定时器模块具有在周期匹配时，或在外部门控信号的下降沿产生中断的能力。当 32 位定时器计数与相应的周期寄存器相匹配时，或者检测到外部“门控”信号的下降沿时，T3IF 位（IFS0<7>）将置 1，如果允许中断则将产生中断。在此模式下，T3IF 中断标志用作中断源。T3IF 位必须在软件中清零。

通过相应的定时器中断允许位 T3IE（IEC0<7>）来允许中断。

表 9-1: TIMER2/3 寄存器映射

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态	
TMR2	0106	Timer2 寄存器																	uuuu uuuu uuuu uuuu
TMR3HLD	0108	Timer3 保持寄存器（仅适用于 32 位定时器操作）																	uuuu uuuu uuuu uuuu
TMR3	010A	Timer3 寄存器																	uuuu uuuu uuuu uuuu
PR2	010C	周期寄存器 2																	1111 1111 1111 1111
PR3	010E	周期寄存器 3																	1111 1111 1111 1111
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>	T32	—	—	TCS	—	0000 0000 0000 0000	
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS<1:0>	—	—	—	TCS	—	0000 0000 0000 0000	

图注：u = 未初始化的位

注：有关寄存器位字段的描述，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

注:

10.0 输入捕捉模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

本节对输入捕捉模块和相关的工作模式进行了介绍。此模块提供的工作特性可用于需要进行频率（周期）和脉冲测量的应用场合。图 10-1 所示为输入捕捉模块的框图。输入捕捉模式可用于以下场合：

- 频率 / 周期 / 脉冲测量
- 作为额外的外部中断源

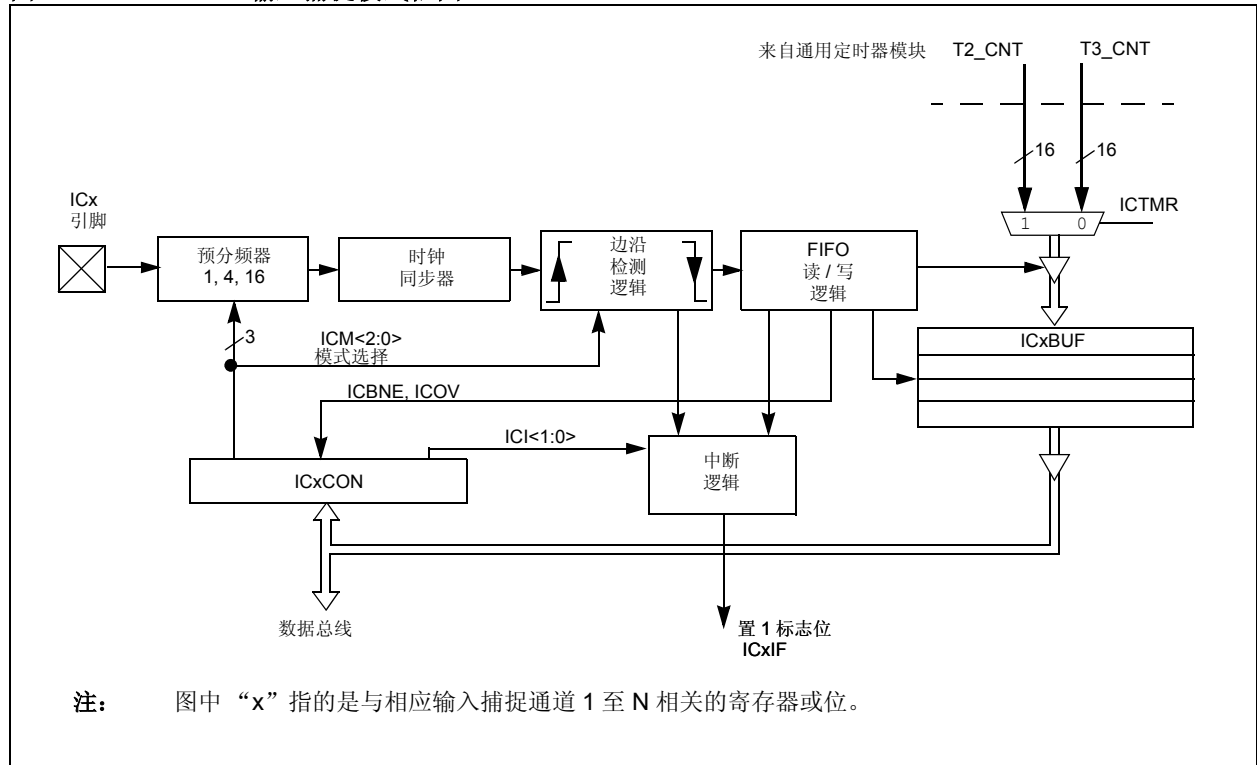
输入捕捉模块的关键工作特性是：

- 简单捕捉事件模式
- Timer2 和 Timer3 模式选择
- 输入捕捉事件发生时产生中断

通过在 ICxCON 寄存器（其中， $x = 1, 2, \dots, N$ ）中设置相应的位，选择上述工作模式。dsPIC DSC 器件具有最多 8 个捕捉通道（即，N 的最大值为 8）。

注： dsPIC30F1010 器件不具备输入捕捉模块。dsPIC30F202X 器件具有一个捕捉输入——IC1。此捕捉通道的命名意在保持与其他 dsPIC DSC 器件的软件兼容性。

图 10-1: 输入捕捉模式框图



10.1 简单捕捉事件模式

dsPIC30F 产品系列中的简单捕捉事件是：

- 在每个下降沿进行捕捉
- 在每个上升沿进行捕捉
- 在每第 4 个上升沿进行捕捉
- 在每第 16 个上升沿进行捕捉
- 在每个上升沿和下降沿进行捕捉

通过设置适当的位 ICM<2:0> (ICxCON<2:0>)，选择上述简单输入捕捉模式。

10.1.1 捕捉预分频器

有四种输入捕捉预分频器设置，它们由位 ICM<2:0> (ICxCON<2:0>) 指定。每当捕捉通道关闭时，预分频器计数器将清零。另外，任何复位都将清零预分频器计数器。

10.1.2 捕捉缓冲器操作

每个捕捉通道都有一个关联的 FIFO 缓冲器，缓冲器宽 16 位，深度为四级。有两个状态标志，提供 FIFO 缓冲器的状态：

- ICBFNE——输入捕捉缓冲器非空
- ICOV——输入捕捉溢出

ICBFNE 将在第一个输入捕捉事件发生时被置 1，并且一直保持置 1 状态，直到所有的捕捉事件都已从 FIFO 中读出。每次从 FIFO 中读出一个字，余下的字将在缓冲器中前进一个位置。

当 FIFO 已满 (FIFO 中有四个捕捉事件) 时，如果在读 FIFO 之前发生第五个捕捉事件，则将产生溢出，且 ICOV 位将被设置成逻辑 1。第五个捕捉事件将丢失，不会存入 FIFO。而且在从缓冲器读出全部四个事件之前不会再捕捉任何事件。

如果 FIFO 已空、且还没有收到新的捕捉事件，这时读 FIFO 将产生不确定的结果。

10.1.3 TIMER2 和 TIMER3 选择模式

输入捕捉模块有多达 8 个输入捕捉通道。每个通道均可选择两个定时器 (Timer2 和 Timer3) 之一作为时基。

通过特殊功能寄存器位 ICTMR (ICxCON<7>)，来进行定时器资源的选择。对于输入捕捉模块，Timer3 是缺省的可用定时器资源。

10.1.4 霍尔传感器模式

当输入捕捉模块设置为在每个边沿 (上升沿和下降沿，ICM<2:0> = 001) 进行捕捉时，输入捕捉逻辑将执行下列操作：

- 输入捕捉中断标志在每个边沿 (上升沿和下降沿) 置 1。
- 捕捉时产生中断模式设置位 ICI<1:0> 被忽略，因为每次捕捉都产生中断。
- 在这个模式下，不产生捕捉溢出。

10.2 休眠和空闲模式期间的输入捕捉工作

当器件在 CPU 空闲或休眠模式下时，输入捕捉事件将使器件唤醒。如果允许中断，还将产生中断。

如果 $ICM<2:0> = 111$ 且中断允许位置 1，那么当捕捉事件发生时，输入捕捉模块将使 CPU 从休眠或空闲模式中唤醒，这与使能的时钟无关。如果处理中断的条件已经得到满足的话，前述的唤醒还将产生中断。唤醒特性可以用来增加额外的外部引脚中断。

10.2.1 CPU 休眠模式下的输入捕捉

在 CPU 休眠模式下输入捕捉模块可以工作，但功能有所限制。在 CPU 休眠模式下， $ICI<1:0>$ 位不可用，输入捕捉模块只能用作外部中断源。

当器件处于休眠模式下时，为了使用输入捕捉模块，必须把捕捉模块设置成仅在上升沿产生中断（ $ICM<2:0> = 111$ ）。在此模式下，4:1 或 16:1 的预分频比设定不可用。

10.2.2 CPU 空闲模式下的输入捕捉

在 CPU 休眠模式下输入捕捉模块可以工作，但功能有所限制。在 CPU 空闲模式下，通过 $ICI<1:0>$ 位选择的的中断模式可用，由控制位 $ICM<2:0>$ 定义的 4:1 和 16:1 捕捉预分频比设定也同样可用。这个模式要求使能所选择的定时器。此外， $ICSIDL$ 位必须置为逻辑 0。

在 CPU 空闲模式下，如果输入捕捉模块定义为 $ICM<2:0> = 111$ ，则输入捕捉引脚仅用作外部中断引脚。

10.3 输入捕捉中断

输入捕捉通道具有根据选定的捕捉事件数来产生中断的能力。可通过控制位 $ICI<1:0>$ （ $ICxCON<6:5>$ ）来设定选定的捕捉事件数。

每个通道都有一个中断标志位（ $ICxIF$ ）。各捕捉通道中断变址位于相应的 $IFSx$ 状态寄存器中。

通过相应捕捉通道中断允许位（ $ICxIE$ ）来允许中断；捕捉通道中断允许位位于相应的 IEC 控制寄存器中。

表 10-1: 输入捕捉寄存器映射

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态
IC1BUF	0140	输入捕捉 1 寄存器																uuuu uuuu uuuu uuuu
IC1CON	0142	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>				0000 0000 0000 0000

图注: u = 未初始化的位
注: 有关寄存器字段的描述, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

11.0 输出比较模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

本节对输出比较模块和相关工作模式进行了介绍。此模块提供的特性适用于需要以下工作模式的应用：

- 产生可变宽度输出脉冲
- 功率因数校正

图 11-1 所示为输出比较模块的框图。

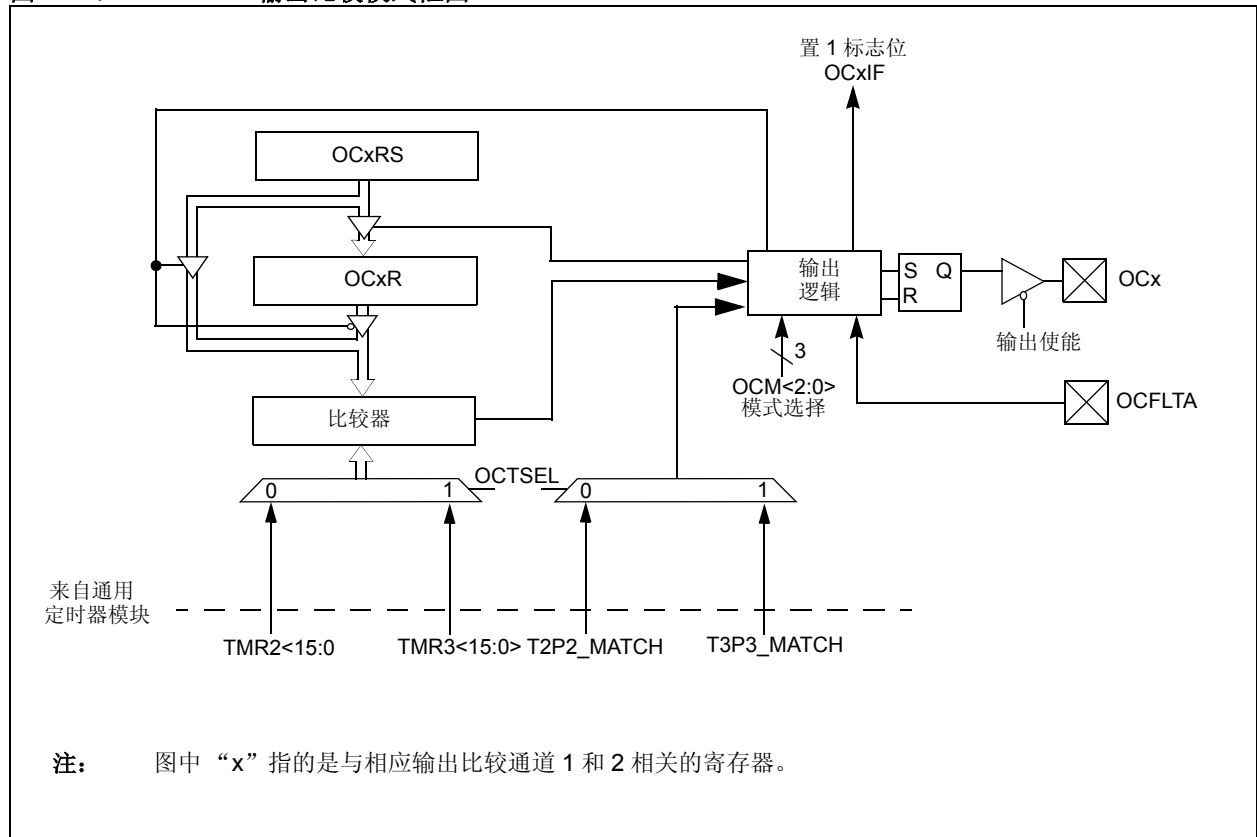
输出比较模块的主要工作特性包括：

- Timer2 和 Timer3 模式选择
- 简单输出比较匹配模式
- 双输出比较匹配模式
- 简单 PWM 模式
- 休眠和空闲模式下的输出比较
- 发生输出比较 /PWM 事件时将产生中断

通过设置 16 位 OCxCON SFR（其中 $x = 1$ 或 2 ）中相应的位可确定这些操作模式。

图中的 OCxRS 和 OCxR 表示双比较寄存器。在双比较模式下，OCxR 寄存器用于第一次比较，而 OCxRS 用于第二次比较。

图 11-1: 输出比较模式框图



11.1 Timer2 和 Timer3 选择模式

每个输出比较通道均可选择两个 16 位定时器 Timer2 和 Timer3 中的一个。

通过 OCTSEL 位 (OCxCON<3>) 来控制定时器的选择。Timer2 是输出比较模块默认的定时器资源。

11.2 简单输出比较匹配模式

当控制位 OCM<2:0> (OCxCON<2:0>) = 001、010 或 011 时, 所选输出比较通道被配置为下列三种简单输出比较匹配模式之一:

- 比较强制 I/O 引脚为低电平
- 比较强制 I/O 引脚为高电平
- 比较使 I/O 引脚状态翻转

上述模式中使用 OCxR 寄存器。OCxR 寄存器将装入一个值, 然后与所选的递增定时器的计数值作比较。进行比较时, 将出现上述比较匹配模式之一。如果在达到 OCxR 中的值之前, 计数器复位为零, 则 OCx 引脚的状态保持不变。

11.3 双输出比较匹配模式

当控制位 <2:0> (OCxCON<2:0>) = 100 或 101 时, 所选输出比较通道被配置为下列两种双输出比较匹配模式之一:

- 单输出脉冲模式
- 连续输出脉冲模式

11.3.1 单脉冲模式

用户将模块配置为生成单输出脉冲, 需要执行下列步骤 (假设定时器已关闭):

- 确定指令周期时间 Tcy。
- 根据 Tcy 计算所需的脉冲宽度值。
- 计算从定时器起始值 0x0000 到开始输出脉冲的时间。
- 把脉冲宽度开始和停止时间分别写入 OCxR 和 OCxRS 比较寄存器 (x 表示通道 1 或 2)。
- 设置定时器周期寄存器的值, 使之等于或大于 OCxRS 比较寄存器中的值。
- 设置 OCM<2:0> = 100。
- 使能定时器, TON (TxCON<15>) = 1。

要启动另一个单脉冲, 进行另一次写操作设置 OCM<2:0> = 100 即可。

11.3.2 连续脉冲模式

用户将模块配置为生成连续输出脉冲, 则需要执行下列步骤:

- 确定指令周期时间 Tcy。
- 根据 Tcy 计算所需的脉冲宽度值。
- 计算从定时器起始值 0x0000 到开始输出脉冲的时间。
- 把脉冲宽度开始和停止时间分别写入 OCxR 和 OCxRS 比较寄存器 (x 表示通道 1 或 2)。
- 设置定时器周期寄存器的值, 使之等于或大于 OCxRS 比较寄存器中的值。
- 设置 OCM<2:0> = 101。
- 使能定时器, TON (TxCON<15>) = 1。

11.4 简单 PWM 模式

当控制位 OCM<2:0> (OCxCON<2:0>) = 110 或 111 时, 所选输出比较通道配置为 PWM 工作模式。当配置为 PWM 工作模式时, OCxR 是主锁存器 (只读), OCxRS 是辅助锁存器。这样能使 PWM 在电平跳变时不会产生毛刺。

为了把输出比较模块配置为 PWM 工作模式, 用户必须执行下列步骤:

1. 写适当的周期寄存器, 设置 PWM 周期。
2. 写 OCxRS 寄存器, 设置 PWM 占空比。
3. 把输出比较模块配置为 PWM 工作模式。
4. 设置 TMRx 预分频比, 使能定时器, TON (TxCON<15>) = 1。

11.4.1 PWM 周期

通过写 PRx 寄存器可以指定 PWM 周期。PWM 周期可使用公式 11-1 计算。

公式 11-1: PWM 周期

$$\text{PWM 周期} = [(PRx) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMRx 预分频比})$$

PWM 频率定义为 $1/[\text{PWM 周期}]$ 。

当所选 TMRx 等于其对应的周期寄存器 PRx 时，在下一个递增周期将发生如下四个事件：

- TMRx 清零。
- OCx 引脚置 1。
 - 例外 1: 如果 PWM 占空比为 0x0000，则 OCx 引脚将保持为低电平。
 - 例外 2: 如果占空比大于 PRx，则引脚将保持为高电平。
- 把 PWM 占空比从 OCxRS 锁存到 OCxR 中。
- 相应的定时器中断标志置 1。

请参见图 11-1 获取关键的 PWM 周期比较。为了清晰起见，图中引用的是 Timer3。

11.4.2 带故障保护输入引脚的 PWM

当控制位 OCM<2:0> (OCxCON<2:0>) = 111 时，通过 OCFLTA 引脚使能故障保护功能。若在 OCFLTA 引脚上检测到逻辑 0，输出引脚将被置于高阻态。此状态将保持，直到发生了下列事件：

- 外部故障条件已消除，且
- 通过写相应的控制位重新使能了 PWM 模式

此故障条件将导致 OCxIF 中断标志位置 1，若允许中断，还将产生中断。当检测到故障条件时，OCxCON 寄存器中的 OCFLTx 位置 1。此位为只读位，只要清除外部故障条件，它就会被清零，通过写相应的模式位 (OCxCON 寄存器中的 OCM<2:0>) 可重新使能 PWM 模式。

11.5 CPU 休眠模式期间的输出比较操作

当 CPU 进入休眠模式时，所有的内部时钟都将停止工作。因此，当 CPU 进入休眠模式时，输出比较通道将把引脚驱动为 CPU 进入休眠模式之前的有效状态。

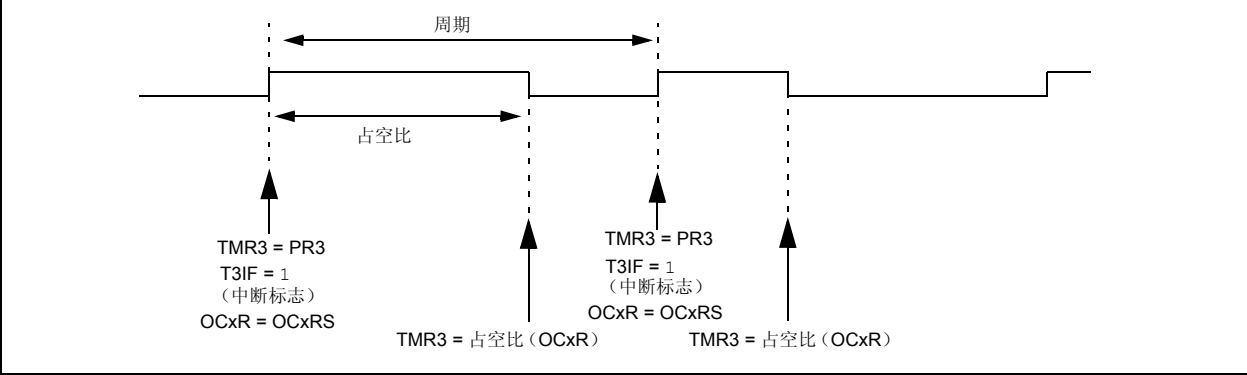
例如，当 CPU 进入休眠状态时，如果引脚为高电平，则引脚将继续保持为高电平。同理，当 CPU 进入休眠状态时，如果引脚为低电平，它将继续保持为低电平。在这两种情况下，当器件唤醒时，输出比较模块都将恢复工作。

11.6 CPU 空闲模式期间的输出比较操作

当 CPU 进入空闲模式时，输出比较模块仍然能够全功能工作。

如果 OCSIDL 位 (OCxCON<13>) 为逻辑 0、所选时基 (Timer2 或 Timer3) 使能且所选定时器的 TSIDL 位设置为逻辑 0，输出比较通道将在 CPU 空闲模式期间工作。

图 11-1: PWM 输出时序



11.7 输出比较中断

输出比较通道具有在比较匹配时产生中断的能力，而与选择的匹配模式无关。

对于除 PWM 之外的所有模式，当发生比较事件时，相应的中断标志（OCxIF）置 1，如果允许中断的话还将产生中断。OCxIF 位于相应的 IFS 状态寄存器中，且必须用软件清零。通过相应的比较中断允许位（OCxIE）来允许中断；OCxIE 位位于相应的 IEC 控制寄存器中。

对于 PWM 模式，当事件发生时，对应的定时器中断标志位（T2IF 或 T3IF）置 1，如果允许中断的话还将产生中断。IF 位在 IFS0 状态寄存器中，且必须用软件清零。通过位于 IEC0 控制寄存器中的定时器中断允许位（T2IE 或 T3IE），来允许相应的中断。在 PWM 工作模式下，输出比较中断标志始终不会置 1。

表 11-1: 输出比较寄存器映射

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态	
OC1RS	0180	输出比较 1 从寄存器																	0000 0000 0000 0000
OC1R	0182	输出比较 1 主寄存器																	0000 0000 0000 0000
OC1CON	0184	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000 0000 0000 0000	
OC2RS	0186	输出比较 2 从寄存器																	0000 0000 0000 0000
OC2R	0188	输出比较 2 主寄存器																	0000 0000 0000 0000
OC2CON	018A	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>			0000 0000 0000 0000	

注: 有关寄存器位字段的描述, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

注:

12.0 电源 PWM

dsPIC30F1010/202X 器件中的电源 PWM (Power Supply PWM, PS PWM) 模块支持多种 PWM 模式和输出格式。PWM 模块适用于电源转换应用, 例如:

- DC/DC 转换器
- AC/DC 电源
- 不间断电源 (Uninterruptible Power Supplies, UPS)

12.1 特性概述

PS PWM 模块具有下列特性:

- 带 8 个 I/O 引脚的 4 个 PWM 发生器
- 4 个独立的时基
- 30 MIPS 时, 占空比分辨率为 1.1 ns
- 30 MIPS 时, 死区时间分辨率为 4.2 ns
- 30 MIPS 时, 相移分辨率为 4.2 ns
- 30 MIPS 时, 频率分辨率为 8.4 ns
- 支持的 PWM 模式:
 - 标准边沿对齐的 PWM
 - 互补 PWM
 - 推挽式 PWM
 - 多相 PWM
 - 可变相位的 PWM
 - 固定关断时间的 PWM
 - 电流复位 PWM
 - 限流 PWM
 - 独立时基 PWM
- 实时修改:
 - PWM 频率
 - PWM 占空比
 - PWM 相移
- 输出改写控制
- 独立的限流和故障输入
- 用于调度其他外设事件的特殊事件比较器
- 每个 PWM 发生器都有一个触发 ADC 转换的比较器。

图 12-1 以简化框图的形式概括了 PWM 模块。图 12-2 说明了如何在互补 PWM 模式下将模块硬件分配给各对 PWM 输出。PWM 模块的各功能单元将在后续章节中讨论。

PWM 模块包含 4 个 PWM 发生器。此模块有 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 引脚分成 4 组 H/L 对。

12.2 说明

PWM 模块是为需要以下功能的应用而设计的: (a) 高 PWM 频率时为高分辨率、(b) 能驱动标准推挽式或半桥式转换器或 (c) 能创建多相 PWM 输出。

有两种常见的中等功耗的转换器结构, 分别为推挽式和半桥式。此设计要求 PWM 输出信号可在备用引脚之间切换, 如推挽式 PWM 模式中提供的那样。

相移 PWM 描述了各 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 通常用于提高 DC-DC 转换器负载的瞬态响应, 降低输出滤波器的电容和电感值。多个 DC/DC 转换器通常同时工作, 但在时间上有一个移相。工作在 250 kHz 的单个 PWM 输出的周期为 4 μ s。但由 4 个 PWM 通道级成的阵列 (每个 PWM 之间的相移为 1 μ s) 可产生 1 MHz 的有效切换频率。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。这时 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电流。

注: 要使 PS PWM 模块工作必须先使能 PLL。使用 FOSCSEL 配置寄存器中的 FNOSC<1:0> 位可使能 PLL。

图 12-1: 电源 PWM 的概念性简化框图

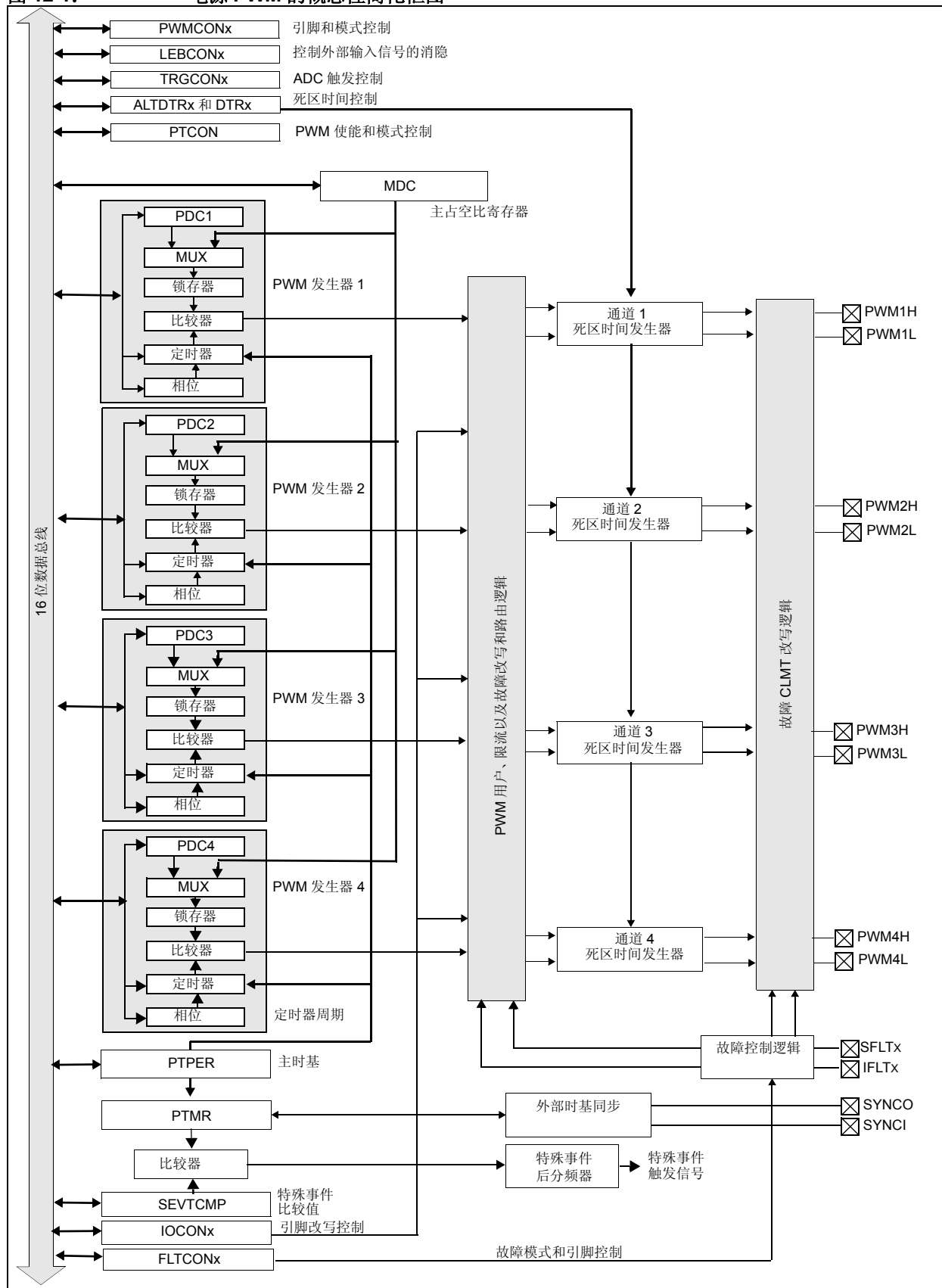
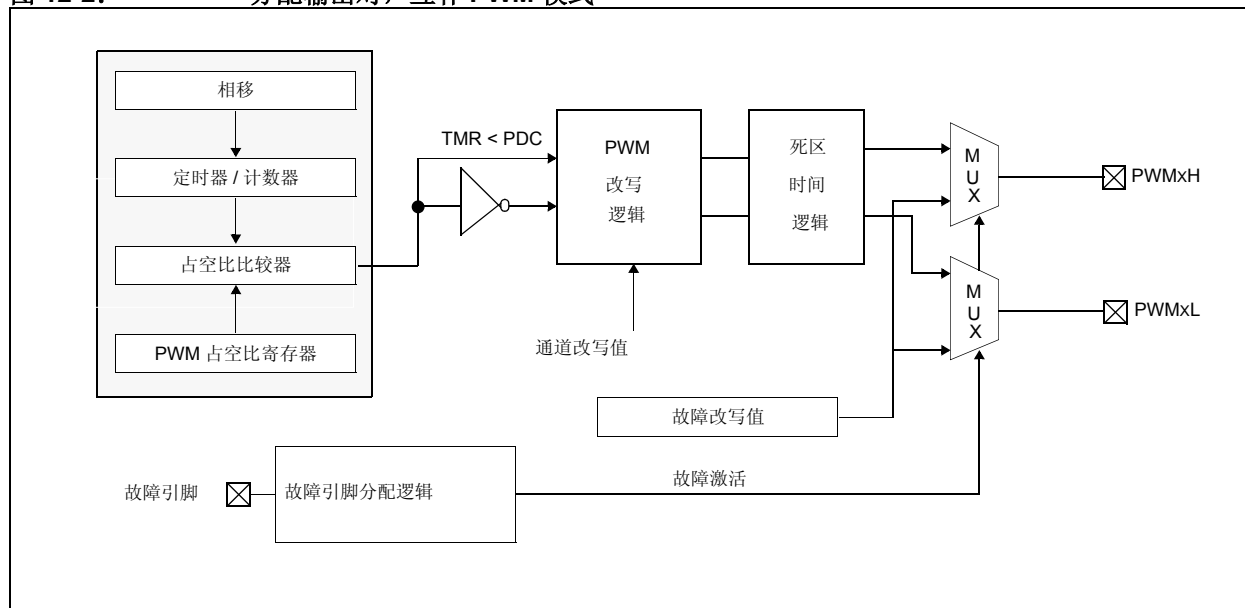


图 12-2: 分配输出对, 互补 PWM 模式



12.3 控制寄存器

下列寄存器控制电源 PWM 模块的工作。

- PTCN: PWM 时基控制寄存器
- PTPER: 主时基寄存器
- SEVTCMP: PWM 特殊事件比较寄存器
- MDC: PWM 主占空比寄存器
- PWMCONx: PWM 控制寄存器
- PDCx: PWM 发生器占空比寄存器
- 相位 x: PWM 相移寄存器 (当模块配置为单周期模式时为 PWM 周期寄存器)
- DTRx: PWM 死区时间寄存器
- ALTDTRx: PWM 备用死区时间寄存器
- TRGCONx: PWM 触发控制寄存器
- IOCONx: PWM I/O 控制寄存器
- FCLCONx: PWM 故障限流控制寄存器
- TRIGx: PWM 触发比较值寄存器
- LEBCONx: 前沿消隐控制寄存器

dsPIC30F1010/202X

寄存器 12-1: PTCON: PWM 时基控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN	—	PTSIDL	SESTAT	SEIEN	EIPU	SYNCPOL	SYNCOEN
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
SYNCEN	SYNCSRC<2:0>			SEVTPS<3:0>			
bit 7							bit 0

图注:			
R = 可读位	W = 可写位	U = 未实现, 读为 0	
-n = 上电复位时的值	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: 空闲模式时 PWM 时基停止位 1 = 在 CPU 空闲模式下, PWM 时基停止 0 = 在 CPU 空闲模式时, PWM 时基运行
bit 12	SESTAT: 特殊事件中断状态位 1 = 特殊事件中断等待处理 0 = 无特殊事件中断等待处理
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: 周期即时更新使能位 1 = 立即更新有效周期寄存器 0 = 在 PWM 周期边界上发生有效周期寄存器更新
bit 9	SYNCPOL: 同步输入极性位 1 = 翻转 SYNCIN 的极性 (低电平有效) 0 = SYNCIN 高电平有效
bit 8	SYNCOEN: 主时基同步使能位 1 = 使能 SYNCO 输出 0 = 禁止 SYNCO 输出
bit 7	SYNCCEN: 外部时基同步使能位 1 = 使能主时基外部同步 0 = 禁止主时基外部同步
bit 6-4	SYNCSRC<2:0>: 同步源选择位 000 = SYNCI 001 = 保留 . . 111 = 保留
bit 3-0	SEVTPS<3:0>: PWM 特殊事件触发输出后分频比选择位 0000 = 1:1 后分频比 0001 = 1:2 后分频比 1111 = 1:16 后分频比

寄存器 12-2: PTPER: 主时基寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTPER <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
PTPER <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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-3 主时基 (PTMR) 周期值位

bit 2-0 未实现: 读为 0

寄存器 12-3: 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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-3 特殊事件比较计数值位

bit 2-0 未实现: 读为 0

dsPIC30F1010/202X

寄存器 12-4: 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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-0 主 PWM 占空比值位 ⁽¹⁾

注 1: 此寄存器的最小值为 0x0008, 最大值为 0xFFEF。

寄存器 12-5: PWMCONx: PWM 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	CLSTAT	TRGSTAT	FLTIE	CLIE	TRGIE	ITB	MDCS
bit 15				bit 0			

R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
DTC<1:0>		—	—	—	—	XPRES	IUE
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **FLTSTAT:** 故障中断状态
1 = 故障中断正在等待处理
0 = 无故障中断在等待处理
通过设置 FLTIE = 0 来清零此位。

注: 软件必须清零该中断状态位和中断控制器中相应的 IFS 位。

bit 14 **CLSTAT:** 限流中断状态位
1 = 限流中断正在等待处理
0 = 无限流中断在等待处理
通过设置 CLIE = 0 来清零此位。

注: 软件必须清零该中断状态位和中断控制器中相应的 IFS 位。

bit 13 **TRGSTAT:** 触发中断状态位
1 = 触发中断正在等待处理
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

寄存器 12-5: PWMCONx: PWM 控制寄存器 (续)

bit 11	CLIEN: 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	TRGIEN: 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 1 = 相位 x 寄存器为 PWM 发生器提供时基周期 0 = 主时基为 PWM 发生器提供时序
bit 8	MDCS: 主占空比寄存器选择位 1 = MDC 寄存器为 PWM 发生器提供占空比信息 0 = PDCx 寄存器为 PWM 发生器提供占空比信息
bit 7-6	DTC<1:0>: 死区时间控制位 00 = 正死区时间施加给所有输出模式 01 = 负死区时间施加给所有输出模式 10 = 禁止死区功能 11 = 保留
bit 5-2	未实现: 读为 0
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 PWM 发生器的时基 0 = 外部引脚不影响 PWM 时基
bit 0	IUE: 即时更新使能位 1 = 立即对有效的 PDC 寄存器进行更新 0 = 对有效的 PDC 寄存器的更新与 PWM 时基同步

寄存器 12-6: PDCx: PWM 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 = 上电复位时的值	1 = 置 1	0 = 清零
		x = 未知

bit 15-0 PWM 发生器 x 占空比值位 ⁽¹⁾

注 1: 此寄存器的最小值为 0x0008, 最大值为 0xFFEF。

dsPIC30F1010/202X

寄存器 12-7: 相位 x: PWM 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	U-0	U-0
PHASEx<7:2>						—	—
bit 7							bit 0

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

bit 15-2 **PHASEx<15:2>**: PWM 相移值位或 PWM 发生器的独立时基周期位

注: 若用作独立时基, 则不能使用 bit <3:2>。

bit 1-0 未实现: 读为 0

寄存器 12-8: DTRx: PWM 死区时间寄存器

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	U-0	U-0
DTRx<7:2>						—	—
bit 7							bit 0

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

bit 15-14 未实现: 读为 0

bit 13-2 **DTRx<13:2>**: PWMx 死区时间单元的 12 位无符号死区时间值位

bit 1-0 未实现: 读为 0

寄存器 12-9: ALTDTRx: PWM 备用死区时间寄存器

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	U-0	U-0
ALTDTR <7:2>						—	—
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-2 **ALTDTRx<13:2>**: PWMx 死区时间单元的 12 位无符号死区时间值

bit 1-0 未实现: 读为 0

寄存器 12-10: TRGCONx: PWM 触发控制寄存器

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TRGDIV<2:0>			—	—	—	—	—
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
—	—	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-13 **TRGDIV<2:0>**: 触发信号输出分频比

000 = 每 1 次触发事件输出一次触发信号

001 = 每 2 次触发事件输出一次触发信号

010 = 每 3 次触发事件输出一次触发信号

011 = 每 4 次触发事件输出一次触发信号

100 = 每 5 次触发事件输出一次触发信号

101 = 每 6 次触发事件输出一次触发信号

110 = 每 7 次触发事件输出一次触发信号

111 = 每 8 次触发事件输出一次触发信号

bit 12-6 未实现: 读为 0

bit 5-0 **TRGSTRT<5:0>**: 触发后分频器启动使能选择位

此值指定匹配所需的 POLL 计数器值, 匹配后将使能触发后分频器逻辑对触发事件进行计数。

dsPIC30F1010/202X

寄存器 12-11: IOCONx: PWM 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	R/W-0
PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		—	OSYNC
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **PENH:** PWMH 输出引脚所有权位

1 = PWM 模块控制 PWMxH 引脚

0 = GPIO 模块控制 PWMxH 引脚

bit 14 **PENL:** PWML 输出引脚所有权位

1 = PWM 模块控制 PWMxL 引脚

0 = 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 x I/O 引脚模式位

00 = PWM I/O 引脚对处于互补输出模式

01 = PWM I/O 引脚对处于独立输出模式

10 = PWM I/O 引脚对处于推挽式输出模式

11 = 保留

bit 9 **OVRENH:** PWMxH 引脚的改写使能位

1 = OVRDAT<1> 为 PWMxH 引脚提供输出数据

0 = PWM 发生器为 PWMxH 引脚提供数据

bit 8 **OVRENL:** PWMxL 引脚的改写使能位

1 = OVRDAT<0> 为 PWMxL 引脚提供输出数据

0 = PWM 发生器为 PWMxL 引脚提供数据

bit 7-6 **OVRDAT<1:0>:** PWMxH 和 PWMxL 引脚的数据位 (若使能了改写功能)

若 OVERENH = 1, 则 OVRDAT<1> 为 PWMxH 提供数据

若 OVERENL = 1, 则 OVRDAT<0> 为 PWMxL 提供数据

bit 5-4 **FLTDAT<1:0>:** PWMxH 和 PWMxL 引脚的数据位 (若使能了 FLTMODE)

若发生故障, 则 FLTDAT<1> 为 PWMxH 提供数据

若发生故障, 则 FLTDAT<0> 为 PWMxL 提供数据

bit 3-2 **CLDAT<1:0>:** PWMxH 和 PWMxL 引脚的数据位 (若使能了 CLMODE)

若限流有效, 则 CLDAT<1> 为 PWMxH 提供数据

若限流有效, 则 CLDAT<0> 为 PWMxL 提供数据

bit 1 **未实现:** 读为 0

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 OVDDAT<1:0> 位进行的输出改写在下一个时钟的边界处发生

寄存器 12-12: FCLCONx: PWM 故障限流控制寄存器

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CLSRC<3:0>				CLPOL
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
CLMODE	FLTSRC<3:0>				FLTPOL	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-13 **未实现:** 读为 0bit 12-9 **CLSRC<3:0>:** PWM X 发生器的限流控制信号源选择位

0000 = 模拟比较器 1

0001 = 模拟比较器 2

0010 = 模拟比较器 3

0011 = 模拟比较器 4

0100 = 保留

0101 = 保留

0110 = 保留

0111 = 保留

1000 = 共享故障 1 (SFLT1)

1001 = 共享故障 2 (SFLT2)

1010 = 共享故障 3 (SFLT3)

1011 = 共享故障 4 (SFLT4)

1100 = 保留

1101 = 独立故障 2 (IFLT2)

1110 = 保留

1111 = 独立故障 4 (IFLT4)

bit 8 **CLPOL:** PWM 发生器 X 限流极性位

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 7 **CLMODE:** PWM 发生器 X 限流模式使能位

1 = 使能限流功能

0 = 禁止限流功能

dsPIC30F1010/202X

寄存器 12-12: FCLCONx: PWM 故障限流控制寄存器 (续)

bit 6-3	FLTSRC<3:0>: PWM 发生器 X 的故障控制信号源选择位
	0000 = 模拟比较器 1
	0001 = 模拟比较器 2
	0010 = 模拟比较器 3
	0011 = 模拟比较器 4
	0100 = 保留
	0101 = 保留
	0110 = 保留
	0111 = 保留
	1000 = 共享故障 1 (SFLT1)
	1001 = 共享故障 2 (SFLT2)
	1010 = 共享故障 3 (SFLT3)
	1011 = 共享故障 4 (SFLT4)
	1100 = 保留
	1101 = 独立故障 2 (IFLT2)
	1110 = 保留
	1111 = 独立故障 4 (IFLT4)
bit 2	FLTPOL: PWM 发生器 X 故障极性位
	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 故障模式位
	00 = 所选故障源强制 PWMxH 和 PWMxL 引脚为 FLTDAT 值 (锁存条件)
	01 = 所选故障源强制 PWMxH 和 PWMxL 引脚为 FLTDAT 值 (周期)
	10 = 保留
	11 = 禁止故障引脚

寄存器 12-13: TRIGx: PWM 触发比较值寄存器

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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-3 TRGCMP<15:3>: 触发控制值位 ⁽¹⁾

寄存器包含 PWMx 时基的比较值, 用于产生触发信号到 ADC 模块以启动采样和转换过程, 或产生触发中断。

bit 2-0 未实现: 读为 0

注 1: 此寄存器的最小可用值为 0x0008

值 0x0000 不会产生触发信号。

若根据占空比值计算 TRIGx 值, 则必须确保始终将最小的 TRIGx 值写入寄存器。

dsPIC30F1010/202X

寄存器 12-14: 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 = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知

bit 15	PHR: PWMH 上升沿触发使能位 1 = PWMH 上升沿触发 LEB 计数器 0 = LEB 忽略 PWMH 的上升沿
bit 14	PHL: PWMH 下降沿触发使能位 1 = PWMH 下降沿触发 LEB 计数器 0 = LEB 忽略 PWMH 的下降沿
bit 13	PLR: PWML 上升沿触发使能位 1 = PWML 上升沿触发 LEB 计数器 0 = LEB 忽略 PWML 的上升沿
bit 12	PLF: PWML 下降沿触发使能位 1 = PWML 下降沿触发 LEB 计数器 0 = LEB 忽略 PWML 的下降沿
bit 11	FLTLEBEN: 故障输入前沿消隐使能位 1 = 前沿消隐应用于所选故障输入 0 = 前沿消隐不应用于所选故障输入
bit 10	CLLEBEN: 限流前沿消隐使能位 1 = 前沿消隐应用于所选限流输入 0 = 前沿消隐不应用于所选限流输入
bit 9-3	LEB: 限流和故障输入的前沿消隐位 值以 8 ns 为单位递增
bit 2-0	未实现: 读为 0

12.4 模块功能

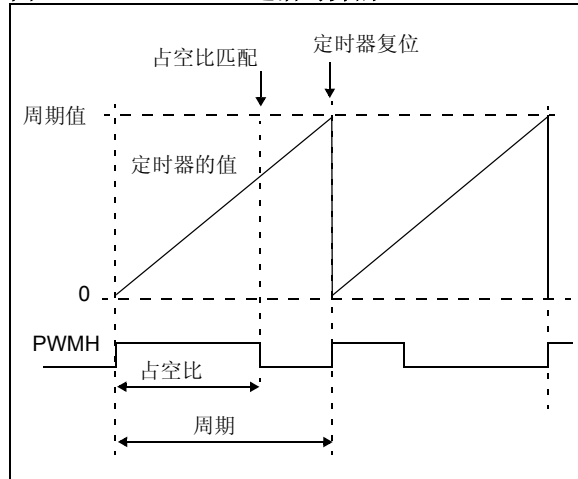
PS PWM 模块是一种极高速设计，提供了其他 PWM 发生器所无法提供的功能。此模块支持下列 PWM 模式：

- 标准边沿对齐 PWM 模式
- 互补 PWM 模式
- 推挽式 PWM 模式
- 多相 PWM 模式
- 可变相位 PWM 模式
- 限流 PWM 模式
- 固定关断时间 PWM 模式
- 电流复位 PWM 模式
- 独立时基 PWM 模式

12.4.1 标准边沿对齐 PWM 模式

标准边沿对齐模式（图 12-3）是许多电源转换器拓扑结构（如“降压型”、“升压型”和“正向激励型”等）所采用的基本 PWM 模式。要创建边沿对齐的 PWM，定时器 / 计数器电路应从零向上计数，直到指定的周期最大值。另一个寄存器包含占空比值，此值不断地与定时器（周期）值进行比较。当定时器 / 计数器值小于或等于占空比值时，PWM 输出信号有效。定时器值大于占空比值时，PWM 信号无效。当定时器的值大于周期值时，定时器复位，并重复上述过程。

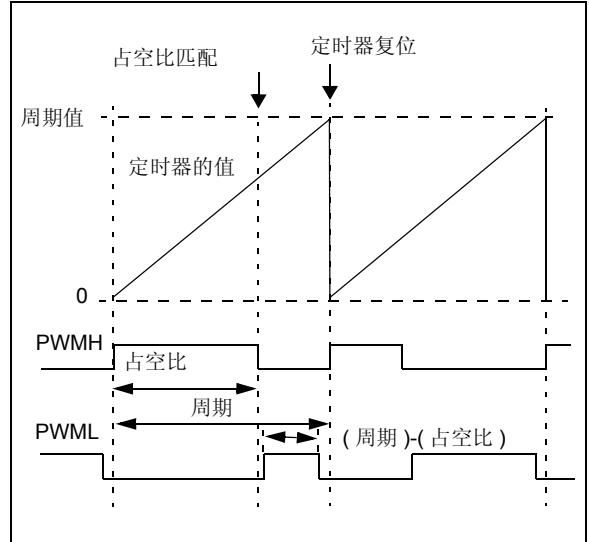
图 12-3: 边沿对齐的 PWM



12.4.2 互补 PWM 模式

互补 PWM 的生成方式与标准边沿对齐的 PWM 类似。互补模式在 PWML 引脚上提供第二个 PWM 输出信号，此信号与主 PWM 信号（PWMH）互补。互补模式 PWM 如图 12-4 所示。

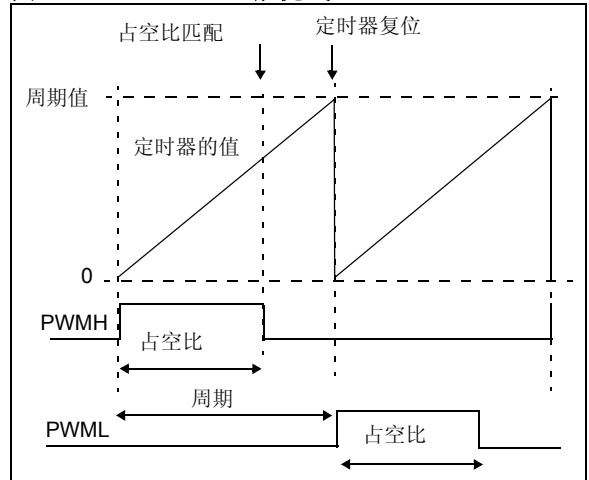
图 12-4: 互补 PWM



12.4.3 推挽式 PWM 模式

图 12-5 中所示的推挽式模式是标准边沿对齐 PWM 模式的一种形式，此模式下，有效 PWM 信号可在两个 PWM 引脚之一上输出。此模式无互补 PWM 输出，它对于基于变压器的电源转换器很有用。基于变压器的电路必须避免任何会导致其铁芯饱和的直流。推挽模式保证了两相的占空比相等，从而产生净值为零的 DC 偏置电压。

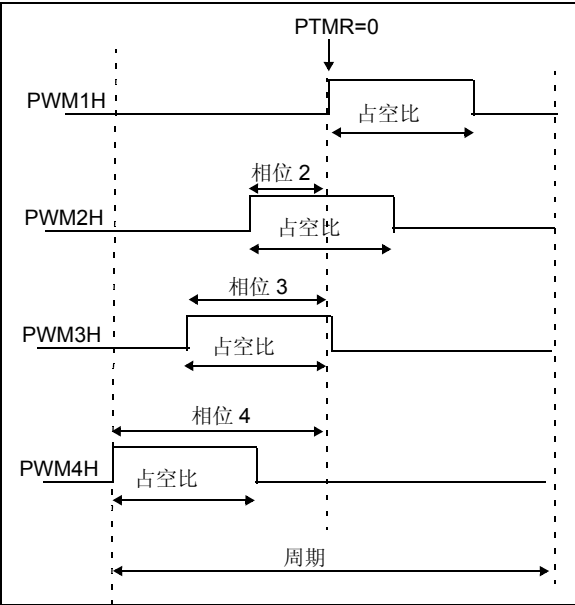
图 12-5: 推挽式 PWM



12.4.4 多相 PWM 模式

如图 12-6 所示，多相 PWM 使用相位寄存器中的相移值以主时基为基准平移 PWM 输出。由于相移值被加到主时基上，因此移相的输出早于指定为零相移的 PWM 通道输出而发生。在多相模式下，指定相移在应用设计时被固定。

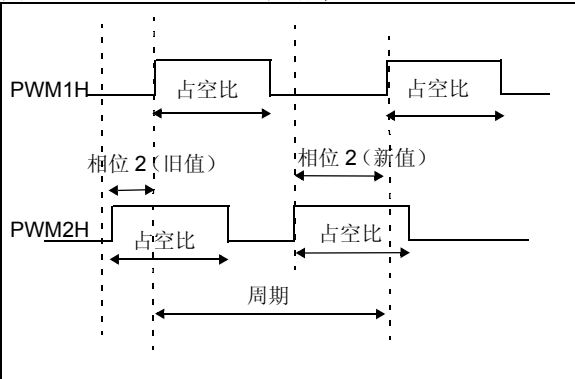
图 12-6: 多相 PWM



12.4.5 可变相位 PWM 模式

图 12-7 给出了可变相位 PWM 的波形图。电源转换器电路不断改变 PWM 通道之间的相移，作为控制能流的一种方法，与多数通过改变 PWM 信号的占空比来控制能流的 PWM 电路形成对比。通常，在可变相位应用中，PWM 占空比保持在 50%。若 PWM 信号无效，应更新相移值。可变相位模式中有互补输出。

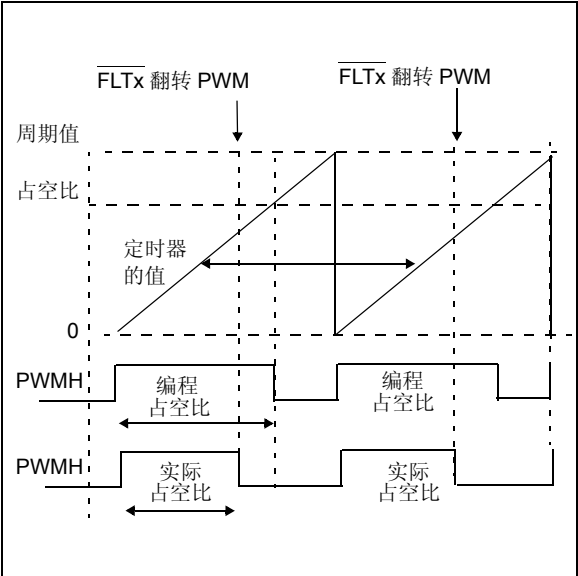
图 12-7: 可变相位 PWM



12.4.6 限流 PWM 模式

图 12-8 所示为逐周期限流模式。当所选外部故障信号有效时，此模式截断有效的 PWM 信号。PWM 输出值由 IOCONx 寄存器中的故障改写位 (FLTDAT<1:0>) 指定。改写输出保持有效，直到下一个 PWM 周期的开始。有时，此模式可用于功率因数校正 (Power Factor Correction, PFC) 电路，此电路中由电感电流实时控制 PWM。此为固定频率的 PWM 模式。

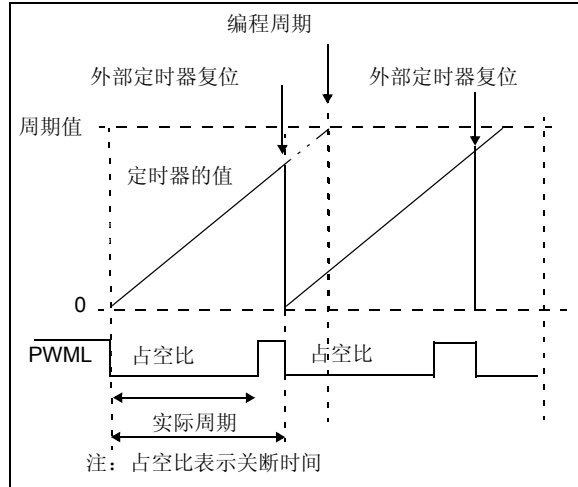
图 12-8: 逐周期限流 PWM 模式



12.4.7 固定关断时间 PWM

固定关断时间模式如图 12-9 所示。固定关断时间 PWM 为可变频模式，此时实际 PWM 周期小于或等于指定的周期值。PWM 时基在达到 PWM 信号占空比值后的某时通过外部条件复位。通过使能导通时间 PWM 模式（电流复位模式）并使用互补输出实现此模式。

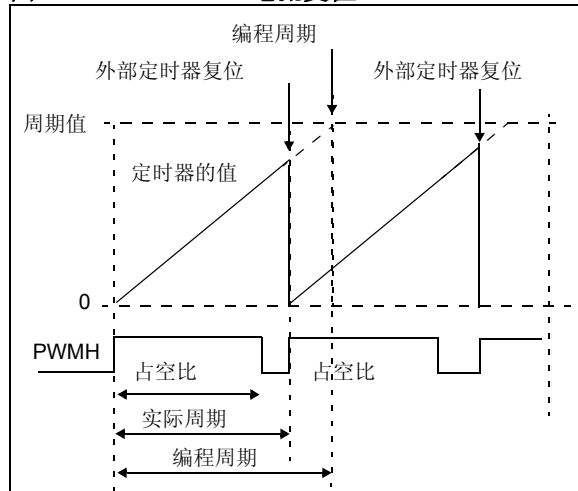
图 12-9: 固定关断时间 PWM



12.4.8 电流复位 PWM 模式

电流复位 PWM 如图 12-10 所示。电流复位 PWM 使用变频模式，此模式下，实际 PWM 周期小于或等于指定的周期值。PWM 时基在达到 PWM 信号占空比值且 PWM 信号无效后的某时通过外部条件复位。电流复位 PWM 为固定导通时间的 PWM 模式。

图 12-10: 电流复位 PWM

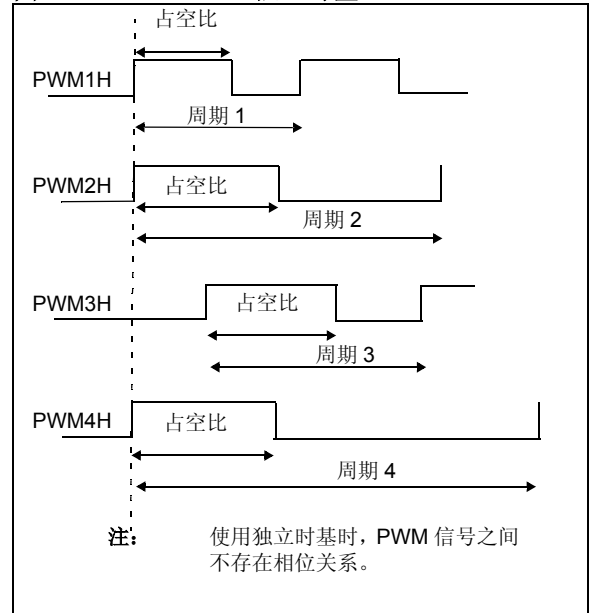


典型情况下，在转换器应用中，当 PWM 信号有效时，储能电感由电流充电，当 PWM 信号无效时，由负载对该电感放电。在电流复位 PWM 应用中，外部电流测量电路决定电感何时放电以及何时产生供 PWM 模块复位时基计数器的信号。在电流复位模式中，互补输出可用。

12.4.9 独立时基 PWM

图 12-11 给出的独立时基 PWM 通常使用在需要 dsPIC DSC 控制不同电源转换器支路的应用中（例如，功率因数校正电路可能使用 100 kHz PWM，而全桥正激转换器部分可能使用 250 kHz PWM）。

图 12-11: 独立时基 PWM

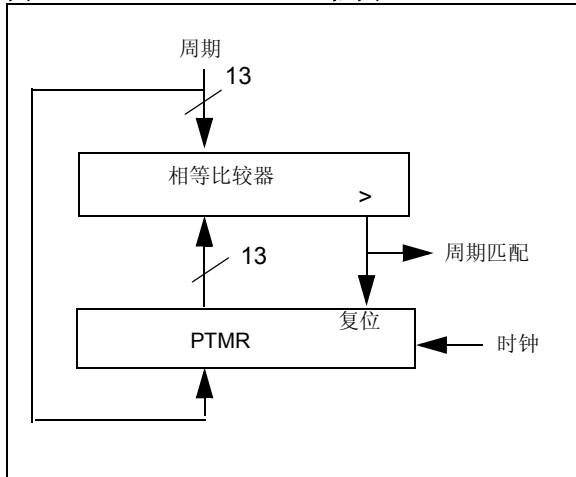


12.5 主 PWM 时基

对于整个 PWM 模块有一个主时基（PTMR）计数器，另外，每个 PWM 发生器都有独立的时基计数器。

PTMR 决定独立时基计数器何时更新其占空比和相移寄存器。主时基还负载产生特殊事件触发信号和基于定时器的中断。图 12-12 给出了主时基逻辑的框图。

图 12-12: PTMR 框图



主时基可由 PTCN 寄存器中的 SYNC SRC<2:0> 位所指定的外部信号复位。外部复位功能通过 PTCN 寄存器中的 SYNCEN 位使能。主时基复位功能支持主时基与其他 SMPS dsPIC DSC 器件或用户应用中的其他电路同步。发生周期匹配时，主时基逻辑还提供输出信号，此信号可用于同步外部器件，例如，其他 SMPS dsPIC DSC。

12.5.1 PTMR 同步

由于绝对同步是不可能的，因此用户应编程辅助（从）器件的时基周期，使之稍大于主器件时基，以确保两个时基能同时复位。

12.6 主 PWM 时基 ROLL 计数器

主时基还有一个额外的 6 位计数器，用来计数与主时基周期匹配的次數。此 ROLL 计数器可使 PWM 发生器向 ADC 模块提供在时间上有相移的多个触发事件。计数器不可读。每个 PWM 发生器在 TRGCONx 寄存器中都有 6 个与之相关的位（TRGSTRT<5:0>）。这些位用于指定由 TRGCONx 寄存器中的 TRGDIV<2:0> 位控制的每个 TRIGx 后分频器的使能。

TRGDIV 位指定产生触发脉冲的频率，ROLL 位指定此脉冲序列的开始时间。一旦使能了 TRIG 后分频器，ROLL 位和 TRGSTRT 位就不再起作用了，直到 PWM 模块禁止后再重新使能。

ROLL 计数器和 TRGSTRT 位是为了使用户能够在一系列 PWM 周期之间分配系统的工作负荷。

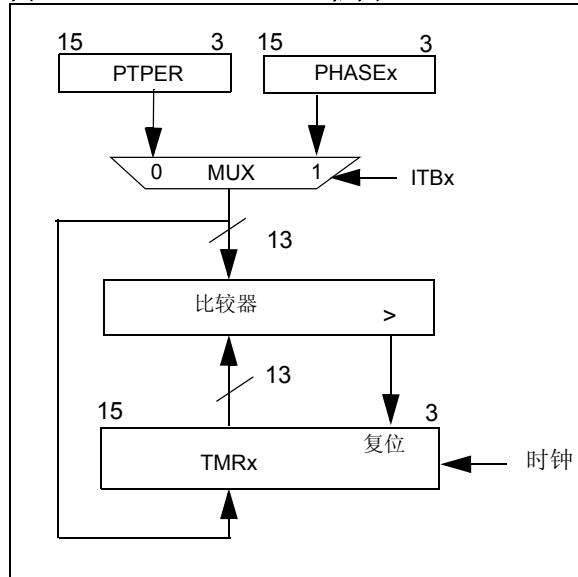
ROLL 计数器的另外一个用途是允许内部 FRC 振荡器根据 PWM 周期变化，以减小在功率转换应用中开关晶体管时产生的 EMI 尖峰脉冲。

禁止 PWM 模块（PTEN = 0）时，清零 ROLL 计数器，并禁止 TRIGx 后分频器，需要一个相对于 TRGSTRT 匹配的新 ROLL 以开始下一次计数。

12.7 独立 PWM 时基

每个 PWM 发生器还有自己的时基。图 12-13 给出了独立时基电路的框图。使用各 PWM 发生器的时基，PWM 模块可以产生彼此之间有相移的或完全独立的 PWM 输出。独立 PWM 定时器（TMRx）提供与占空比寄存器进行比较的时基值，以创建 PWM 信号。用户可通过相移寄存器在操作之前或操作过程中初始化这些独立时基计数器。用户不能读主时基（PTMR）和独立定时器（TMRx）。

图 12-13: TMRx 框图



通常，主时基（PTMR）提供对各定时器 / 计数器的同步控制，使它们能同步计数。

若使用了 PWM 相移功能，PTMR 将为各定时器 / 计数器提供同步信号，使它们用各相移值重新初始化。

若 PWM 发生器以独立时基模式工作，各定时器 / 计数器将向上计数，直到计数值与存储在相位寄存器中的值相匹配，之后定时器 / 计数器复位，重复此过程。

主时基和独立时基实现为 13 位计数器。30 MIPS 时，定时器 / 计数器的时钟速度为 120 MHz，提供 8.4 ns 的频率分辨率。

通过置 1/ 清零 PTCN SFR 中的 PTEN 位来使能 / 禁止所有定时器 / 计数器。当 PTEN 位用软件清零时，定时器清零。

PTPER 寄存器设置 PTMR 的计数周期。用户必须将 13 位值写入 PTPER<15:3>。PTMR<15:3> 中的值与 PTPER<15:3> 中的值匹配时，主时基复位为 0，且独立时基被重新初始化为其相位值（独立时基模式除外）。

12.8 PWM 周期

PTPER 保存指定主 PWM 时基计数周期的 13 位值。用户可随时更新定时器周期。可使用以下公式计算 PWM 周期：

$$\text{周期时间} = (\text{PTPER} + 1) / 120 \text{ MHz @ 30 MIPS}$$

12.9 PWM 频率和占空比分辨率

工作在 30 MIPS 时，PWM 占空比的分辨率为每 1 个 LSB 1.05 ns。30 MIPS 时，PWM 周期分辨率为 8.4 ns。表 12-1 显示了 30 MIPS 执行速度时占空比分辨率和 PWM 频率的关系。

表 12-1: 30 MIPS 时可用的 PWM 频率和分辨率

MIPS	PWM 占空比分辨率	PWM 频率
30	16 位	14.6 kHz
30	15 位	29.3 kHz
30	14 位	58.6 kHz
30	13 位	117.2 kHz
30	12 位	234.4 kHz
30	11 位	468.9 kHz
30	10 位	937.9 kHz
30	9 位	1.87 MHz
30	8 位	3.75 MHz

表 12-2: 20 MIPS 时可用的 PWM 频率和分辨率

MIPS	PWM 占空比分辨率	PWM 频率
20	14 位	39 kHz
20	12 位	156 kHz
20	10 位	624 kHz
20	8 位	2.5 MHz

注意，给定 PWM 频率可用分辨率的降低是因为时钟速率降低和占空比分辨率的 LSB 来源于固定延时元件的事实。工作频率低于 30 MIPS 时，固定延时元件对输出分辨率的影响小于 1 LSB。

对于频率谐振模式功率转换应用，知道可用的 PWM 频率分辨率是理想的。可用频率分辨率随 PWM 频率的变化而变化。30 MIPS 时，PWM 时基时钟为 120 MHz。下列公式提供不同 PWM 周期下的频率分辨率：

$$\text{频率分辨率} = 120 \text{ MHz} / (\text{周期})$$

$$\text{其中，周期} = \text{PTPER} < 15:3 >$$

12.10 PWM 占空比比较单元

PWM 模块有 2 至 4 个占空比发生器。有 5 个 16 位特殊功能寄存器用于指定 PWM 模块的占空比值：

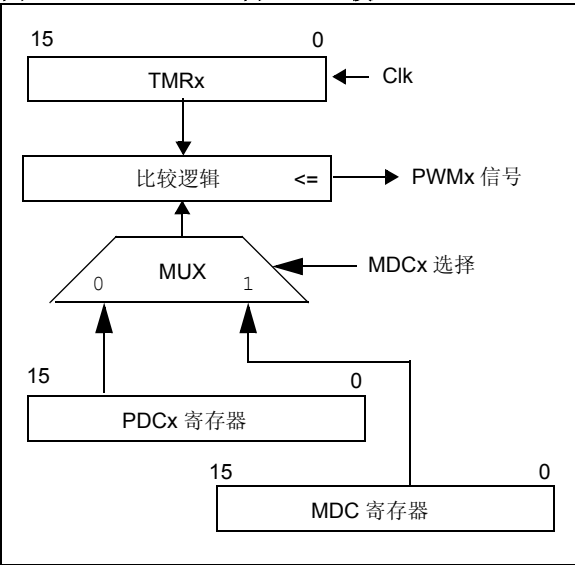
- MDC（主占空比）
- PDC1 至 PDC4（占空比）

各 PWM 发生器都具有自己的占空比寄存器（PDCx），还有一个主占空比（Master Duty Cycle，MDC）寄存器。MDC 寄存器可代替各占空比寄存器。MDC 寄存器使多个 PWM 发生器能共享一个通用的占空比寄存器，以降低更新多个占空比寄存器时所需的 CPU 开销。多相电源转换器即为一个应用，此时使用 MDC 功能可节省宝贵的处理器时间。

每个占空比寄存器中的值决定 PWM 输出保持在有效状态的时间。PWM 时基计数器宽度为 13 位，且在每个指令周期递增两次。当计数器 / 定时器小于或等于占空比寄存器值的 13 个最高有效位时，PWM 输出有效。各占空比寄存器允许指定 16 位占空比。将占空比寄存器的 3 个最低有效位发送到附加逻辑，以对 PWM 信号边沿进行进一步的调整。

图 12-14 为占空比比较单元的框图。

图 12-14: 占空比比较



可随时更新占空比值。可选择将更新的占空比值保持到主时基在其有效之前的下一次计满归零。

12.11 互补 PWM 输出

互补 PWM 输出模式在 PWM 输出引脚对上提供同相和反相 PWM 输出。通过将当前的 PWM 信号翻转产生互补 PWM 信号。通常，互补输出在除推挽式 PWM 和独立 PWM 输出模式之外的所有 PWM 模式中可用。

12.12 独立 PWM 输出

独立 PWM 输出模式只复制与 PWM 发生器相关的两个输出引脚上的有效 PWM 输出信号。

12.13 占空比限制

占空比发生器的值被限定在允许的范围内。值 0x0008 为能产生输出脉冲的最小占空比值。30 MIPS 时，此值表示 8.4 ns。由于 PWM 输出缓冲器、外部 FET 驱动器和功率晶体管的转换率限制，在实际应用中此最小范围限制并不是一个问题。应用控制环需要较大的占空比值以达到晶体管最小导通时间。

最大占空比值被限制为 0xFFEF。

将占空比值限制在 0x0008 到 0xFFEF 的允许范围内是用户的责任。

注： 0x0000 占空比将产生零 PWM 输出，而 0xFFFF 占空比值将产生高 PWM 输出。

12.14 死区时间发生

死区时间指由死区时间寄存器（DTR）或 ALTDTR 寄存器指定的可编程周期，在互补 PWM 信号保持指定时间的无效之前，死区时间阻止主 PWM 输出有效。图 12-15 显示了在互补 PWM 输出对插入死区时间。图 12-16 显示了 4 个有各自死区时间值的死区时间单元。

在任一 PWM I/O 引脚对工作在任何输出模式下时都可提供死区时间。

由于功率晶体管不能瞬间完成通断，因此许多电源转换器电路需要死区时间。要阻止电流“直通”，必须在关闭互补对中的一路 PWM 输出和导通另一个晶体管之间提供一定的时间。

PWM 模块还可以提供负死区时间。负死区时间为 PWMH 和 PWML 信号的强制重叠。存在一些需要限量电流“直通”的转换器技术。

可禁止每个 PWM 发生器的死区时间功能。由 PWMCON 寄存器中的 DTC<1:0> 位控制死区时间功能。

注： 若需要零死区时间，则必须在 PWMCON 寄存器中的 DTC<1:0> 位中明确禁止死区时间功能。

图 12-15: 互补 PWM 的死区时间插入

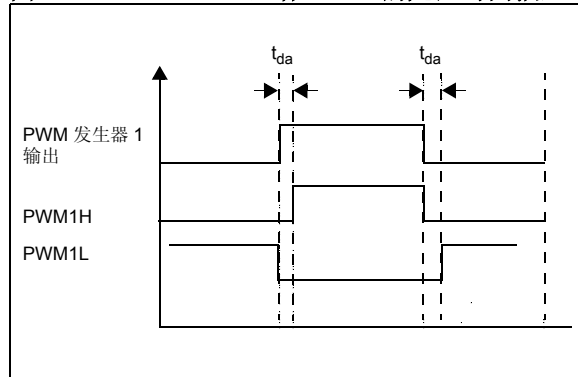
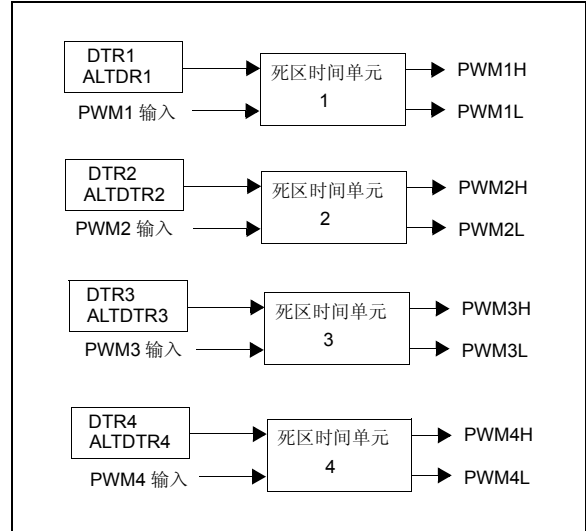


图 12-16: 死区时间控制单元框图



12.14.1 死区时间发生器

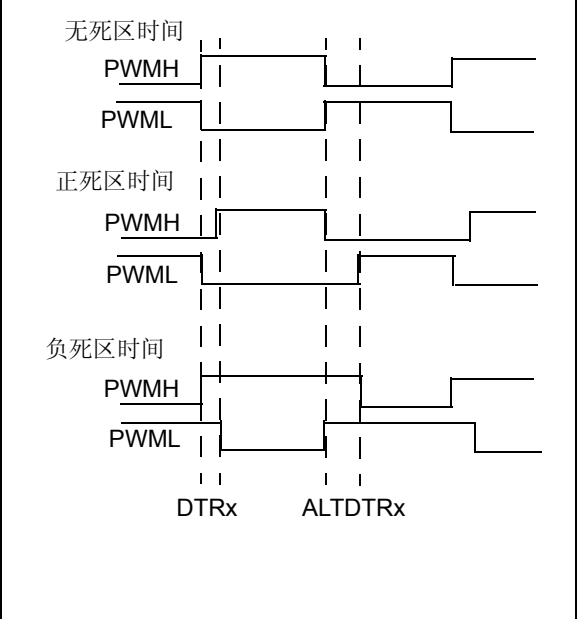
PWM 模块的每对互补输出都有一个 12 位的向下计数器，用于插入死区时间。每个死区时间单元都有与占空比比比较输出相连的上升沿和下降沿检测器。

根据是上升沿还是下降沿，互补输出对中一路信号的转换将会延时直到定时器计数减至零。图 12-15 显示了将死区时间插入一对 PWM 输出的时序图。

12.14.2 备用死区时间源

备用死区时间是指由 ALTDTR 寄存器指定的死区时间，应用于互补 PWM 输出。图 12-17 显示了使用 ALTDTR 寄存器的双死区时间插入。

图 12-17: 双死区时间波形



12.14.3 死区时间范围

通过在 DTRx 寄存器中指定一个 12 位无符号值来选择每个死区时间单元提供的死区时间量。12 位死区时间计数器的时钟速率是指令执行速率的 4 倍。死区时间值的最低有效位由精密调整 (Fine Adjust) PWM 模块进行处理。

表 12-3 显示了死区时间范围随器件工作频率变化的示例。

表 12-3: 死区时间范围示例

MIPS	分辨率	死区时间范围
30	4.16 ns	0-17.03 μ s
20	6.25 ns	0-25.59 μ s

12.14.4 死区时间插入时序

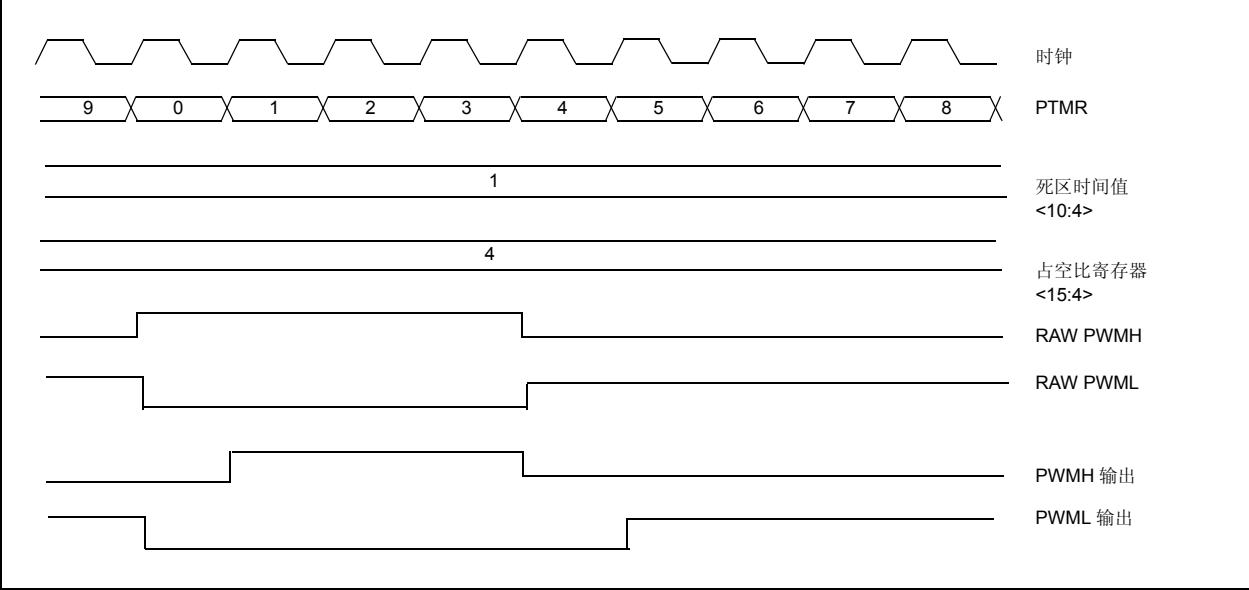
图 12-18 显示了如何在互补信号中插入死区时间。

12.14.5 死区时间失真

对于 PWM 占空比较小的情况，死区时间相对于有效 PWM 时间的比例可能会变大。在这种情况下，插入的死区时间可能会导致 PWM 模块产生的波形失真。通过保持 PWM 占空比至少是死区时间的三倍以上，用户可以确保死区时间失真最小。

当占空比等于或接近 100% 的情况下，也会出现同样的情况。在实际应用中使用的最大占空比应该选择使信号的最小无效时间至少比死区时间大三倍以上。

图 12-18: 死区时间插入 (PWM 输出信号可能会延时)



12.15 配置 PWM 通道

例 12-1 所示代码示例将 PWM 通道 1 配置为互补模式，工作频率为 400 kHz，死区时间的近似值为 64 ns。假设 dsPIC30F1010/202X 工作在内部快速 RC 振荡器，并带有高频范围 PLL（14.55 MHz 输入到 PLL，假设为工业温度级器件）。

12.16 PWM 输出速度限制电路

PWM 输出 I/O 缓冲器以及诸如 FET 驱动器和功率 FET 等附加电路的转换率有限。对于非常小的 PWM 占空比，PWM 输出信号被低通滤波，因而没有脉冲通过所有电路。

接近 100% 的占空比也会发生类似的情况。在占空比达到 100% 之前，输出 PWM 信号呈现 100% 饱和。

用户在应用中需要考虑此种情况。在正常功率转换应用中，应避免接近 0% 或 100% 的占空比值，因为要达到这些值，器件将工作在非连续模式或饱和模式下，而此时控制环不起作用。

12.17 PWM 特殊事件触发信号

PWM 模块具有特殊事件触发功能，可以使 A/D 转换与 PWM 时基同步。可以将 A/D 采样和转换时间编程为在 PWM 周期中的任何时刻发生。特殊事件触发功能使用户能够将获取 A/D 转换结果和更新占空比值之间的延时降至最短。

特殊事件触发基于主 PWM 时基工作。

PWM 特殊事件触发信号有一个寄存器（SEVTCMP），还有 4 个控制位（PTCON 中的 SEVTPS<3:0>）用于控制其工作。产生特殊事件触发信号的 PTMR 值被装入 SEVTCMP 寄存器。

12.17.1 使能特殊事件触发信号

PWM 模块始终产生特殊事件触发脉冲。可选择将此信号用于 ADC 模块。

12.17.2 特殊事件触发器分频器

PWM 特殊事件触发信号有一个后分频比为 1:1 至 1:16 的后分频器。通过写 PTCN 寄存器中的 SEVTPS<3:0> 控制位来配置后分频器。

特殊事件输出后分频器在发生下列事件时清零：

- 对 SEVTCMP 寄存器的任何写入。
- 任何器件复位。

12.18 单个 PWM 触发信号

PWM 模块还具有为每个 PWM 发生器另外产生 ADC 触发输出的功能。当 PWM 发生器工作在独立时基模式时，此功能非常有用。

触发电路的框图如图 12-19 所示。用户在 TRIGx 寄存器中指定一个匹配值。当本地时基计数器值与 TRIGx 值匹配时，将产生一个 ADC 触发信号。

只要 TRIGx 的值小于或等于本地时基的 PWM 周期值，就会产生触发信号，而与 TRIGx 的具体值无关。若在 PWMCONx 寄存器中置 1 TRGIEN 位，将产生中断请求。

根据 TRGCONx 寄存器中的 TRGDIV<2:0> 位的设置，对各触发输出进行分频，这允许每 1、2、3 至 7 个触发事件产生一次 ADC 触发信号。

触发分频器允许用户调节 ADC 采样速率以满足控制环要求。

例 12-1: 配置 PWM 通道 1 的代码示例

注: 此代码示例未说明 PWM 模块各故障模式的配置。它仅作为设置 PWM 模块的快速入门指南。

```
mov #0x0400, w0          ; PWM Module is disabled, continue operation in
mov w0, PTCON             ; idle mode, special event interrupt disabled,
                          ; immediate period updates enabled, no external
                          ; synchronization

; Set the PWM Period
mov #0x094D, w0           ; Select period to be approximately 2.5usec
mov w0, PTPER             ; PLL Frequency is ~480MHz. This equates to a
                          ; clocke period of 2.1nsec. The PWM period and
                          ; duty cycle registers are triggered on both +ve
                          ; and -ve edges of the PLL clock. Therefore,
                          ; one count of the PTPER and PDCx registers
                          ; equals 1.05nsec.
                          ; So, to achieve a PWM period of 2.5usec, we
                          ; choose PTPER = 0x094D

mov #0x0000, w0           ; no phase shift for this PWM Channel
mov w0, PHASE1            ; This register is used for generating variable
                          ; phase PWM

; Select individual Duty Cycle Control
mov #0x0001, w0           ; Fault interrupt disabled, Current Limit
mov w0, PWMCON1           ; interrupt disabled, trigger interrupt,
                          ; disabled, Primary time base provides timing,
                          ; DC1 provides duty cycle information, positive
                          ; dead time applied, no external PWM reset,
                          ; Enable immediate duty cycle updates

; Code for PWM Current Limit and Fault Inputs
mov #0x0003, w0
mov w0, FCLCON1           ; Disable current limit and fault inputs

; Code for PWM Output Control
mov #0xC000, w0           ; PWM1H and PWM1L is controlled by PWM module
mov w0, IOCON1            ; Output polarities are active high, override
                          ; disabled

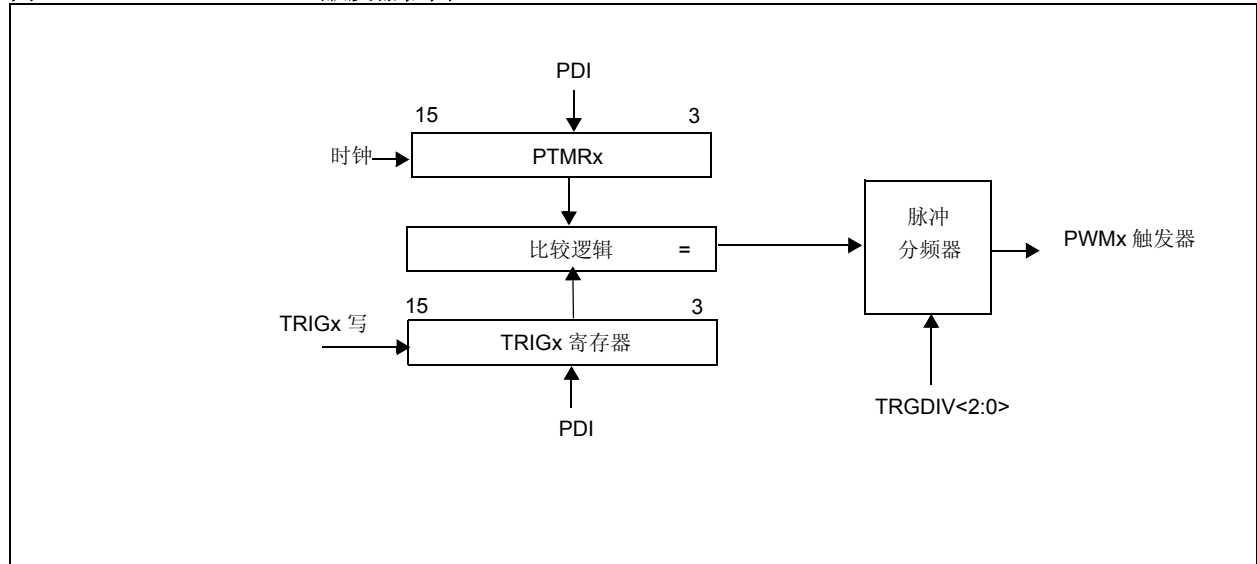
; Duty Cycle Setting
mov #0x04A6, w0           ; To achieve a duty cycle of 50%, we choose
mov w0, PDC1              ; the PDC1 value = 0.5*(PWM Period)
                          ; The ON time for the PWM = 1.25usec
                          ; The Duty Cycle Register will provide
                          ; positive duty cycle to the PWMxH outputs
                          ; when output polarities are active high
                          ; (see IOCON1 register)

; Dead Time Setting
mov #0x0040, w0           ; Dead time ~ 67nsec
mov w0, DTR1              ; Hex(40) = decimal(64)
                          ; So, Dead time = 64*1.05nsec = 67.2nsec
                          ; Note that the last 2 bits are unimplemented,
                          ; therefore the dead time register can achieve a
                          ; a resolution of about 4nsec.

mov w0, ALTDTR1           ; Load the same value in ALTDTR1 register

bset PTCON, #15           ; turn ON PWM module
```

图 12-19: PWM 触发器框图



12.19 PWM 中断

PWM 模块可根据内部时序或通过限流和故障输入的外部信号产生中断。发生特殊事件时，主时基模块可产生中断请求。每个 PWM 发生器模块有自己到中断控制器的中断请求信号。每个 PWM 发生器的中断为模块的触发事件中断请求、限流输入事件以及故障输入事件的逻辑或。

发生特殊事件时，有 4 个用于中断控制的中断请求信号和一个来自主时基的中断请求。

12.20 PWM 时基中断

PWM 模块可根据主时基和/或各 PWM 发生器的时基产生中断。中断时序由主时基的特殊事件比较寄存器 (SEVTCMP) 和 PWM 发生器模块各时基的 TRIGx 寄存器指定。

通过 PTCN 寄存器中的 SEIEN 位允许主时基特殊事件中断。各 PWM 发生器中的触发逻辑产生的独立时基中断由 PWMCONx 寄存器中的 TRGIEN 位控制。

12.21 PWM 故障和限流引脚

PWM 模块支持每个 PWM 发生器具有多个故障引脚。这些引脚称为 SFLT_x (共享故障) 或 IFLT_x (独立故障)。共享故障引脚可被任何 PWM 发生器识别和使用。独立故障引脚由特定 PWM 发生器使用。

各 PWM 发生器可有一个引脚用作逐周期限流引脚，另一个引脚用作逐周期限流或锁存电流故障禁止功能。

12.22 前沿消隐

各 PWM 发生器通过 LEBCONx 寄存器中的 LEB<9:3>、PHR、PHF、PLR、PLF、FLTLEBEN 以及 CLLEBEN 位支持限流和故障引脚的“前沿消隐”功能。前沿消隐是为了屏蔽当功率晶体管导通或关断时在应用印刷电路板上产生的瞬态信号。

LEB 位支持在 PWMH 和 PWML 粗略 (coarse) 信号的任何指定上升沿或下降沿后的 0 到 1024 ns 的周期 (以 8.4 ns 的增量递增) 内，消隐 (忽略) 限流和故障输入。非精细 PWM 信号 (在 PWM 精细调整之前的信号) 分辨率为 8.4 ns (30 MIPS 时)，与 LEB 计数器的时间分辨率相同。

PHR、PHF、PLR 和 PLF 位用于选择 PWMH 和 PWML 信号的哪个边沿启动消隐定时器。若当自上一次选定 PWM 边沿后定时器仍有效时，新选择的边沿触发了 LEB 定时器，此定时器将重新初始化并继续计数。

FLTLEBEN 和 CLLEBEN 位可使能消隐周期的应用给选定的故障和限流输入。

30 MIPS 时, LEB 持续时间 = $(LEB<9:3> + 1)/120 \text{ MHz}$ 。

此时有一个 8.4 ns 的消隐周期偏移量。因此, $LEB<9:3>$ 的零值将产生一个 8.4 ns 的有效消隐周期。

若在上一个 PWM 周期结束时限流或故障输入有效, 且在新 PWM 周期的开始仍然有效, 而死区时间为非零, 则将检测到故障或限流, 而与 LEB 计数器的配置无关。

12.23 PWM 故障引脚

各 PWM 发生器可从最多 12 个故障/限流引脚中选择其自己的故障输入源。在 FCLCONx 寄存器中, 各 PWM 发生器均具有指定其故障输入信号源的控制位。这些控制位为 FLTSRC<3:0> 位。另外, 各 PWM 发生器在 PWM CONx 寄存器中有一个允许产生故障输入请求的 FLTLEN 位。各 PWM 发生器在 FCLCONx 寄存器中有一个选择所选故障输入的有效电平的相关故障极性位 (FLTPOL)。

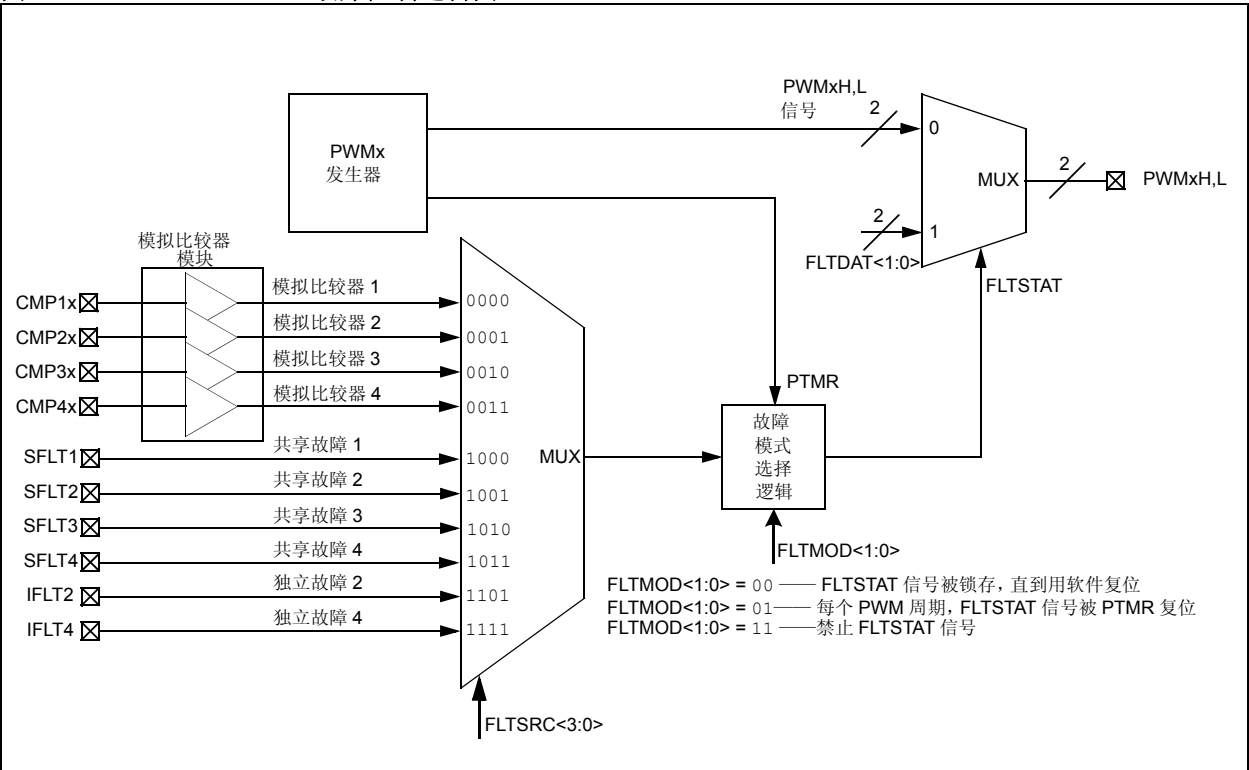
故障引脚实际上具有两种不同的用途。第一种用途是, 为 PWM 输出生成故障改写功能。改写 PWM 输出的操作和产生中断的操作由硬件异步执行, 以便快速管理故障事件。第二种用途是, 故障引脚输入可用于实现限流 PWM 模式或电流强制模式。

PWM 故障条件状态保存在 PWMCONx 寄存器中的 FLTSTAT 位中。若 FIE 位置 1, FLTSTAT 位显示故障 IRQ 锁存器。若未允许故障中断, FSTATx 位将以正逻辑格式显示所选 FLT_x 输入的状态。当故障输入引脚未与 PWM 发生器一起使用时, 这些引脚将作为通用 I/O 或中断输入引脚。

FLT_x 引脚通常为高电平有效。若将 FCLCONx 寄存器中的 FLTPOL 位置 1, 将翻转选定的故障输入信号, 使 FLT_x 引脚为低电平有效。

使能 PWM 模块时, 还可通过端口 I/O 逻辑读故障引脚。这允许用户用软件查询故障引脚的状态。图 12-20 为 PWM 故障控制逻辑的框图。

图 12-20: PWM 故障控制逻辑图



12.23.1 故障中断

PWMCONx 寄存器中的 FLTIEIx 位决定当 FLT_x 引脚拉高时是否产生中断。FCLCONx 寄存器中的 FLTMOD 位决定 PWM 发生器及其输出如何响应选定的故障输入引脚。IOCONx 寄存器中的 FLTDAT<1:0> 位在 PWMxH 和 PWMxL 引脚出现故障时为其提供数据值。

故障引脚逻辑可作为外部中断引脚独立于 PWM 逻辑工作。若在 FCLCONx 寄存器中禁止故障对 PWM 发生器的影响，则故障引脚可用作通用中断引脚。

12.23.2 故障状态

IOCONx 寄存器有两个位，用于决定当各 PWMx I/O 引脚被故障输入改写时的状态。当这些位清零时，PWM I/O 引脚将被驱动为无效状态。若这些位置 1，PWM I/O 引脚将被驱动为有效状态。有效和无效状态与为每个 PWM I/O 引脚定义的极性有关（通过 HPOL 和 LPOL 极性控制位设置）。

12.23.3 故障输入模式

故障输入引脚有两种操作模式：

- **锁存模式：**当故障引脚有效时，PWM 输出将进入 IOCONx 寄存器中的 FLTDAT 位所定义的状态。PWM 输出将保持在此状态，直到故障引脚无效且相应的中断标志由软件清零。当这两个动作都发生后，PWM 输出将在下一个 PWM 周期边界开始时返回到正常工作状态。如果 FLTSTAT 位在故障条件结束前清零，PWM 模块将等到故障引脚不再有效时才恢复输出。通过将零写入 FLTIEI 位，可用软件清零 FLTSTAT。
- **逐周期模式：**只要故障输入引脚保持有效，PWM 输出就会保持有效的 PWM 状态。对于互补输出模式，PWMH 为低电平（无效），PWML 为高电平（有效）。故障引脚驱动为高电平后，PWM 输出将在下一个 PWM 周期开始时返回正常工作状态。

用 FCLCONx 寄存器中的 FLTMOD<1:0> 控制位选择各故障输入引脚的操作模式。

12.23.4 进入故障

PWM 引脚对故障输入引脚的响应总是与器件时钟信号异步。也就是说，PWM 输出应立即转到在 FLTDAT 寄存器中定义的状态，而无需 dsPIC DSC 器件或软件的任何干预。

有关数据灵敏度和对限流或故障事件的响应行为的信息，请参见第 12.28 节“死区时间逻辑的故障和限流改写问题”。

12.23.5 退出故障

在故障条件结束之后恢复 PWM 信号的操作必须发生在 PWM 周期边界，以确保 PWM 信号边沿和手动改写准确同步。当 PTMRx 值为零时，开始下一个 PWM 周期。

12.23.6 PTMR 禁止时退出故障

当禁止 PWM 时基时（PTEN = 0），此为退出故障条件的特例。当故障输入被编程为逐周期模式时，故障输入引脚无效时，PWM 输出将立即恢复正常操作。PWM 输出应返回到其默认编程值。（禁止时基，使得没有理由等待下一个 PWM 周期开始。）

当故障输入被编程为锁定模式时，若故障输入引脚无效且 FSTAT 位由软件清零，则应立即恢复 PWM 输出。

12.23.7 故障引脚软件控制

故障引脚可通过软件手动控制。因为故障输入与端口 I/O 引脚共用，通过将相应的 TRIS 位清零，可以将端口引脚配置为输出。当引脚的 PORT 位清零时会激活故障输入。

注：	当用软件控制故障输入时，用户应小心使用。如果故障引脚的 TRIS 位清零，且 PORT 位置 1，就无法从外部驱动故障输入。
-----------	--

12.24 PWM 限流引脚

各 PWM 发生器可从最多 12 个限流 / 故障引脚中选择自己的限流输入源。在 FCLCONx 寄存器中，各 PWM 发生器具有为其限流输入信号指定输入源的控制位 (CLSRC<3:0>)。另外，各 PWM 发生器在 PWMCONx 寄存器中有允许产生限流中断请求的 CLIEN 位。各 PWM 发生器在 FCLCONx 寄存器中有相应的故障极性位 CLPOL。图 12-21 为 PWM 限流控制逻辑的框图。

限流引脚实际上有两种用途。可用于实现限流 PWM 模式或电流复位 PWM 模式。

1. 在 PWMCONx 寄存器中置 1 CLIEN 位时，若所选限流输入信号有效，则 PWMxH 和 PWMxL 输出将被强制为 IOCONx 寄存器中 CLDAT<1:0> 位指定的值。
2. 当 CLMOD 位为 0，PWMCONx 寄存器中的 XPRES 位为 01，且 PWM 发生器处于独立时基模式 (ITB = 1) 时，限流信号将复位受影响的 PWM 发生器的时基。此行为称为电流复位模式，用在一些功率因数校正 (PFC) 应用中。

12.24.1 限流中断

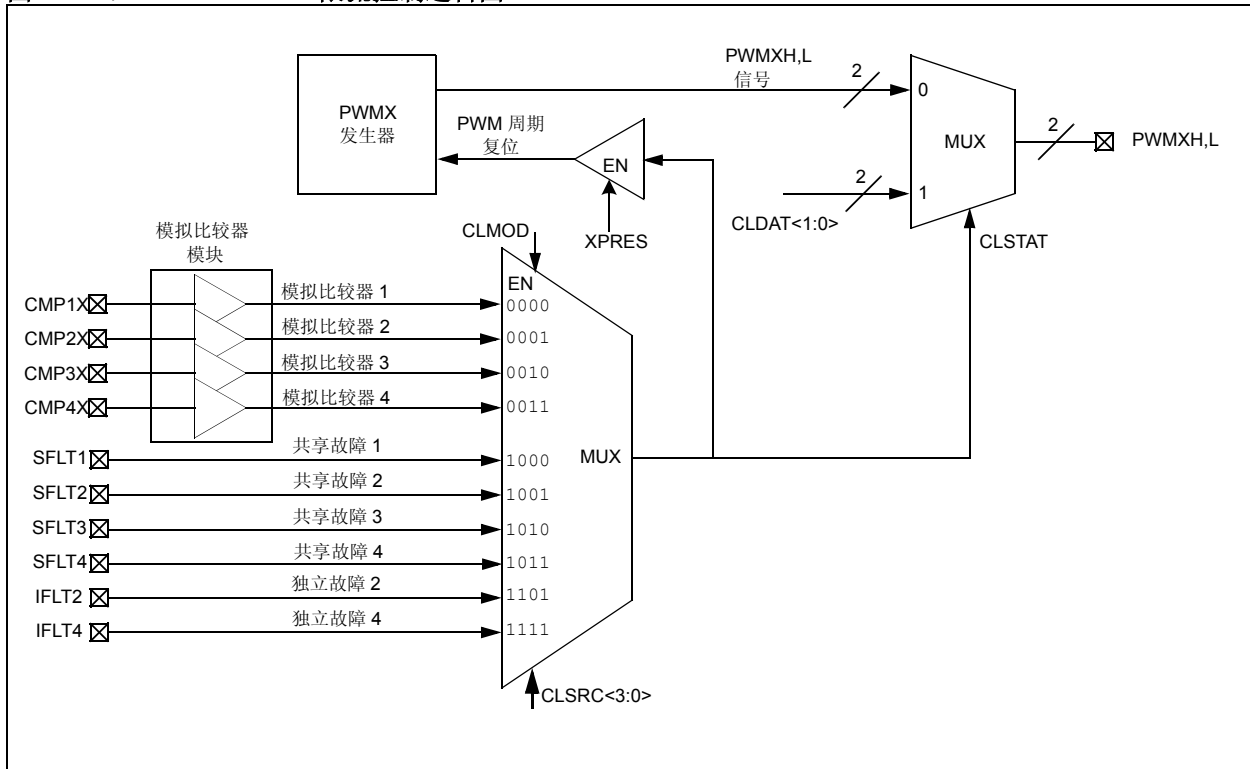
PWM 限流条件的状态可从 PWMCONx 寄存器中的 CLSTAT 位中得到。若 CLIEN 位置 1，CLSTAT 位显示限流 IRQ 标志。若未允许限流中断，CLSTAT 位将以正逻辑格式显示所选限流输入的状态。若未使用与 PWM 发生器相关的限流输入引脚，则此引脚将作为通用 I/O 或中断输入引脚。

限流引脚通常为高电平有效。若将 FCLCONx 寄存器中的 CLPOL 位置 1，将会将选定限流输入信号翻转为有效的高电平。

由所选限流信号产生的中断组合在一起创建到中断控制器的单个中断请求信号，中断控制器包含与中断相关的中断向量、中断标志位、中断允许位和中断优先级位。

使能 PWM 模块时，还可通过端口 I/O 逻辑读故障引脚。这允许用户用软件查询故障引脚的状态。

图 12-21: PWM 限流控制逻辑图



12.25 PWM 故障和限流同时发生

若使能并激活了限流改写功能，除非使能并激活了故障功能，否则限流改写功能将把 PWMxH 和 PWMxL 引脚强制为 IOCONx 寄存器中 CLDAT<1:0> 位指定的值。若所选故障输入有效，PWMxH 和 PWMxL 输出将采用 IOCONx 寄存器中 FLTDAT<1:0> 位指定的值。

12.26 PWM 故障和限流触发信号输出到 ADC

FCLCONx 寄存器中的故障和限流源选择字段 (FLTSRC<3:0> 和 CLSRC<3:0>) 控制各 PWM 发生器模块中的多路开关。控制多路开关为相应的模块选择期望的故障和限流信号。所选故障和限流信号也可作为触发信号供 ADC 模块使用，以启动 ADC 采样和转换操作。

12.27 PWM 输出改写优先级

若使能了 PWM 模块，PWMx 引脚所拥有的优先级为：

1. PWM 发生器（最低优先级）
2. 输出改写
3. 限流改写
4. 故障改写
5. PENx（GPIO/PWM）所有权（最高优先级）

若禁止了 PWM 模块，则 GPIO 模块控制 PWMx 引脚。

12.28 死区时间逻辑的故障和限流改写问题

发生限流或故障事件时，PWMxH 和 PWMxL 输出被立即驱动为低电平（无效），具体由根据 CLDAT<1:0> 和 FLTDAT<1:0> 位指定。

改写数据被门控，而 PWM 信号进入死区时间逻辑模块，此改写数据处于 PWM 模块的输出且刚好在 PWM 引脚输出缓冲器之前。

许多应用要求对精确电流控制的电流关断和 / 或限制电路对故障电流的破坏做出快速响应。

在同步整流器设计中，有些应用会在发生故障或限流事件时将互补 PWM 输出设置为高电平。若 CLDAT 或 FLTDAT 位被置 1，且发生与它们相关的事件，那么死区时间电路中的时钟逻辑将使这些有效的输出延迟。

12.29 通过限流使输出有效

可使用 CLDAT 位使 PWMxH 和 PWMxL 输出有效以响应限流事件。这种行为可用作电流“强制”功能，以响应外部电流或电压测量值中所表明的电源转换器输出上的负载骤增。强制 PWM “开”可视为“前馈”项，允许系统快速响应不期望的负载增加，而无需等待数字逻辑控制环响应。

12.30 PWM 立即更新

对于高性能 PWM 控制环应用，用户可能希望强制占空比立即更新。置 1 PWMCONx 寄存器中的 IUE 位将使能此功能。

在闭环控制应用中，系统状态的检测和随后输出的驱动应用的 PWM 控制信号之间的任何延时都会降低环的稳定性。置 1 IUE 位会降低写占空比寄存器和 PWM 发生器响应更改之间的延时。

12.31 PWM 输出改写

所有与 PWM 输出改写功能相关的控制位都包含在 IOCONx 寄存器中。

若 PENH 和 PENL 位置 1，则 PWM 模块控制 PWMx 输出引脚。

PWM 输出改写位可以让用户手动将 PWM I/O 引脚驱动为指定的逻辑状态，而与占空比比较单元无关。

IOCONx 寄存器中的 OVRDAT<1:0> 位决定当通过 OVRENH 和 OVRENL 位改写特定输出时 PWM I/O 引脚的状态。

OVRENH 和 OVRENL 位为高电平有效控制位。当 OVREN 位置 1 时，相应的 OVRDAT 位改写 PWM 发生器的 PWM 输出。

12.31.1 互补输出模式

当 PWM 在互补输出模式下时，死区时间发生器在改写时仍有效。输出改写和故障改写将产生死区时间单元使用的控制信号，用于根据请求设置输出，包括死区时间。

当手动改写 PWM 通道时，可插入死区时间。

12.31.2 改写同步

若置1 IOCONx寄存器中的OSYNC位，通过OVRENH、OVRENL 和 OVDDAT<1:0> 位执行的输出改写与 PWM 时基同步。当时基为 0 时，发生同步输出改写。

若 PTEN = 0 意味着定时器未运行，写 IOCON 的操作在下一 Tcy 边界生效。

12.32 功能异常

12.32.1 电源复位条件

上电复位时，所有与 PWM 模块相关的寄存器都复位为表 12-4 中给定的状态。器件复位时，PWM 输出引脚为三态。

12.32.2 休眠模式

所选故障输入引脚具有将CPU从休眠模式唤醒的功能。若休眠时任何选定故障引脚被驱动为低电平，PWM 模块将产生一个异步中断。

建议用户在进入休眠模式之前禁止PWM输出。若PWM模块正在控制电源转换应用，则将器件置于休眠模式的操作将导致禁止所有控制环，且多数应用可能会遇到问题，除非这些应用明确设计为工作在开环模式下。

12.32.3 CPU 空闲模式

此 dsPIC30F202X 模块在 PTCON 寄存器中有一个 PTSIDL 控制位。此位决定当器件进入空闲模式时 PWM 模块是继续工作还是停止。停止空闲模式功能与休眠模式类似，故障引脚被异步激活。

- PTSIDL = 1（模块在空闲模式下停止）
- PTSIDL = 0（模块在空闲模式下继续工作）

建议用户在进入空闲模式之前禁止PWM输出。若PWM模块正在控制电源转换应用，则将器件置于空闲模式的操作将导致禁止所有控制环，且多数应用可能会遇到问题，除非这些应用明确设计为工作在开环模式下。

12.33 寄存器位对齐

第 142 页中的表 12-4 显示了 PS PWM 模块的寄存器。此模块的所有时基数据都是相对于时间位对齐的。例如：假设器件工作在 30 MIPS，周期寄存器、占空比寄存器、死区时间寄存器、触发寄存器和相位寄存器中的 bit 3 总是表示值 8.4 ns。寄存器的未用单元总是读为 0。

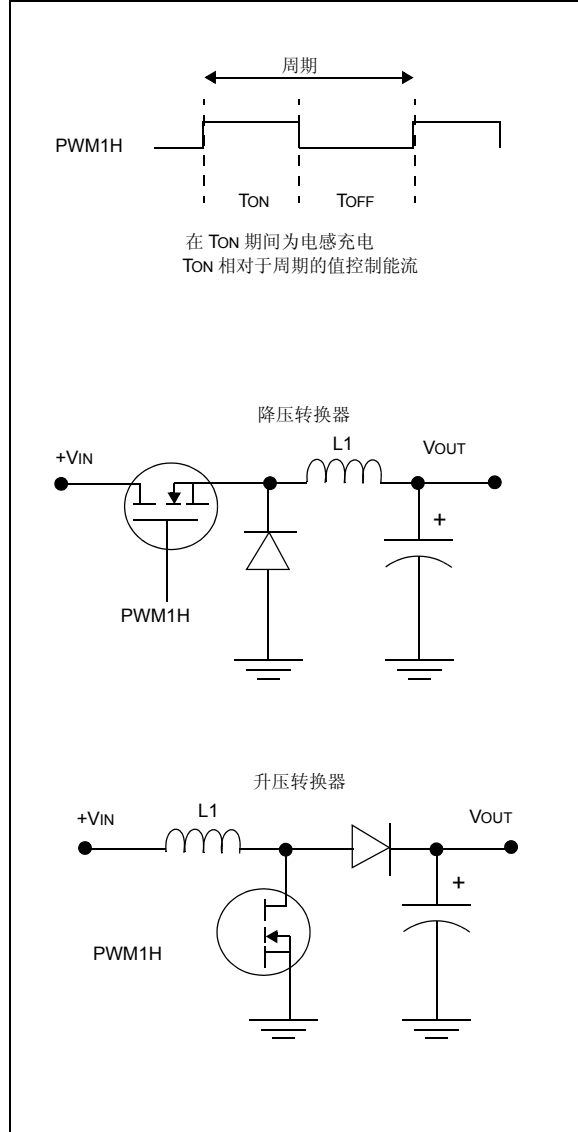
由于使用数据对齐消除了将时间值移入寄存器的必要，因此便于软件编写。此外数据对齐还便于在一个 PWM 周期内计算和理解时间分配。

12.34 应用示例:

12.34.1 标准 PWM 模式

在标准 PWM 模式下，PWM 输出通常连接到为电感充电的一个晶体管，如图 12-22 所示。降压和升压转换器通常使用标准 PWM 模式。

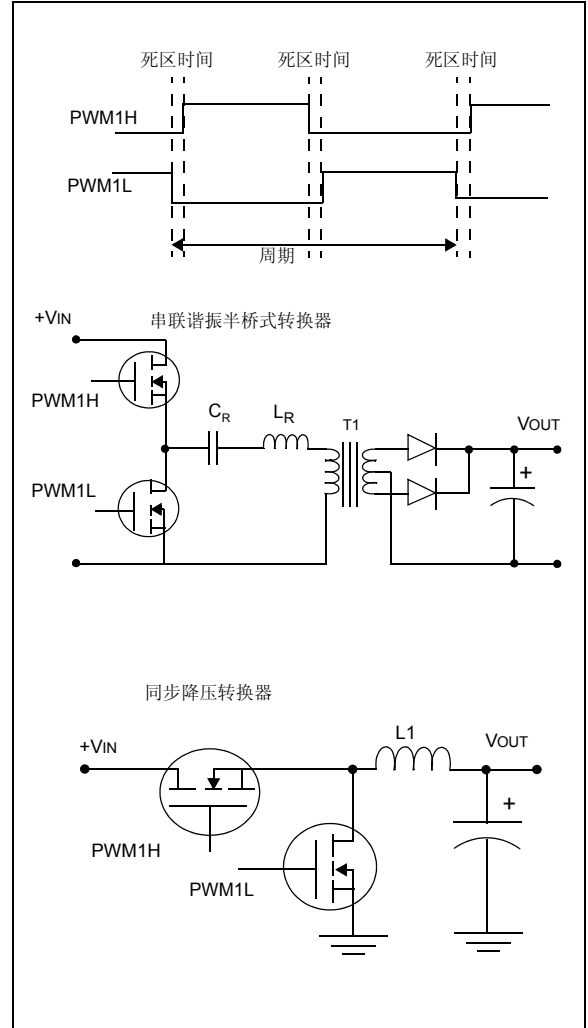
图 12-22: 标准 PWM 模式的应用



12.34.2 互补 PWM 模式的应用

互补 PWM 模式通常用在使用两个桥式配置的晶体管（不使用变压器）的电路中，如图 12-23 所示。若使用了变压器，则必须提供一些方法以确保无净直流电流通过变压器以防止磁芯饱和。

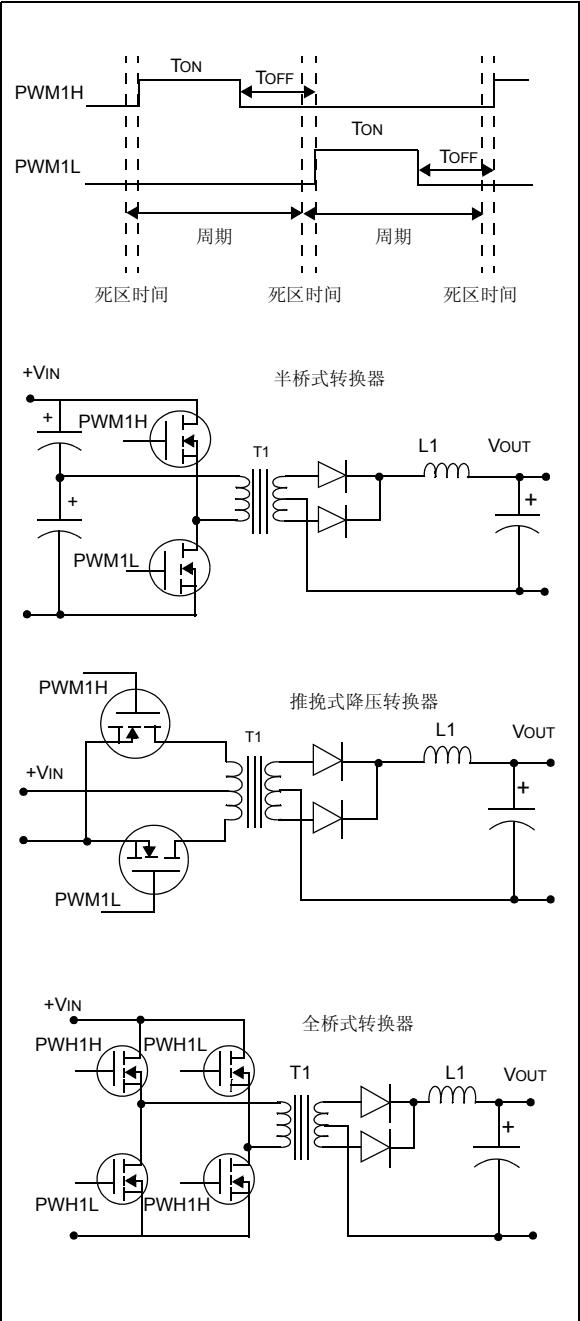
图 12-23: 互补 PWM 模式的应用



12.34.3 推挽式 PWM 模式的应用

推挽式 PWM 模式通常用于变压器耦合电路以确保无净直流电流通过变压器。推挽模式确保将相同占空比的 PWM 脉冲施加到相反方向上的变压器绕组上，如图 12-24 所示。

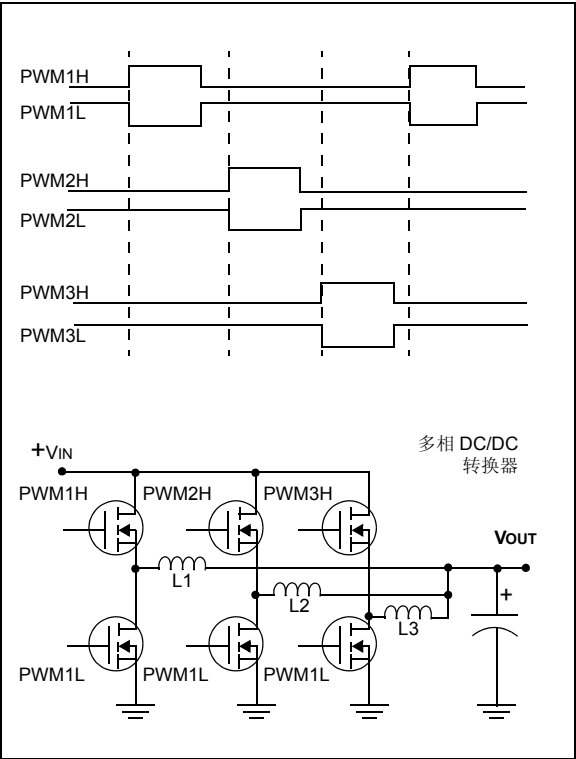
图 12-24: 推挽式 PWM 模式的应用



12.34.4 多相 PWM 模式的应用

多相 PWM 模式通常用于必须处理快速变化的瞬态负载电流并置于紧凑空间的 DC/DC 转换器。多相转换器实际上是一组并联的降压转换器，工作时相互之间有少许不同步，如图 12-25 所示。多个相所生成的有效转换速度等于各转换器的转换速度之和。若单个相位的 PWM 工作频率为 333 kHz，那么此电路的有效切换频率即为 1 MHz。高切换频率极大地减少了输出电容大小要求，并提高了负载瞬态响应。

图 12-25: 多相 PWM 模式的应用

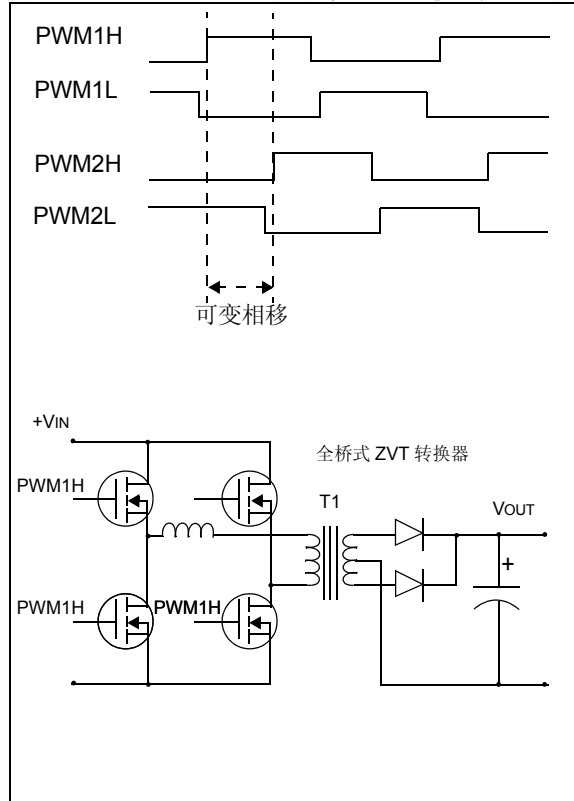


12.34.5 可变相位 PWM 模式的应用

可变相位 PWM 用于为减少开关损耗而设计的新型电源转换拓扑结构中。在标准 PWM 方法中，每次晶体管在导通状态和非导通状态之间（反之亦然）切换时，在导通或关断晶体管的时间周期内，晶体管均工作在全电流和电压条件下。功率损耗（ $V \cdot I \cdot T_{sw} \cdot F_{PWM}$ ）在高频时相当大。零电压开关（Zero Voltage Switching, ZVS）和零电流开关（Zero Current Switching, ZCS）电路拓扑结构试图使用准谐振（quasi-resonant）技术平移彼此相关的电压或电流波形。此行为使电压或电流在晶体管导通或关断时为零。若电流或电压为零，则不产生开关损耗。

在可变相位 PWM 模式下，占空比固定为 50%，能流通过改变 PWM 通道之间的相位关系控制，如图 12-26 所示。

图 12-26: 可变相位 PWM 模式的应用

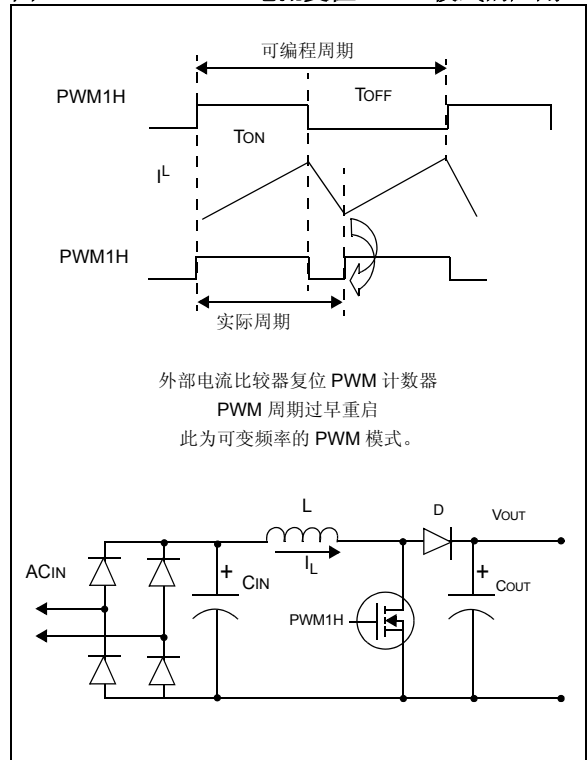


12.34.6 电流复位 PWM 模式的应用

在电流复位 PWM 模式下，PWM 频率随负载电流不同而不同。此模式与多数 PWM 模式不同，因为此模式下由用户设置最大 PWM 周期，而由外部电路测量电感电流。当电感电流降至指定值以下时，外部电流比较器电路产生一个信号来复位 PWM 时基计数器。用户指定 PWM “工作”时间，PWM 信号变为无效后的一段时间后，电感电流降至指定值以下，PWM 计数器在达到可编程 PWM 周期之前复位。此模式有时称为持续工作时间。

不应将此模式与逐周期限流 PWM 模式混淆，在后者中 PWM 无效、外部电路产生电流故障，且 PWM 信号在编程的占空比正常关断此信号之前关断。如图 12-27 所示，此模式下每个时基周期的 PWM 频率固定。

图 12-27: 电流复位 PWM 模式的应用



12.35 降低 EMI 的方法

其目标是及时移动 PWM 边沿以使 EMI 能量均分到一定的频率范围内，从而在 EMI 测量过程（测量长期平均值）中降低任何给定频率处的峰值能量。

EMI 测量过程将 EMI 能量在 9 kHz 宽的频率范围内积分。假设载波（PWM）频率为 150 kHz，6% 的抖动将得到一个 9 kHz 宽的抖动。

12.35.1 方法 1：可编程 FRC 抖动

此方法使所有 PWM 输出和系统时钟抖动。此方法的优点是不需要 CPU 资源。一旦设置就会自动进行。用户可周期性的更新这些值以模拟一个更随机的频率模式。

12.35.2 方法 2：软件控制的抖动

此方法用软件通过缩放占空比和周期来抖动各 PWM 通道。此方法消耗 CPU 资源：

假设：

在速率为 150 kHz 时，更新 4 个 PWM 通道：

600 kHz x (5 个时钟 (2 条 mul 指令、1 条 tblrdl 指令和 1 条 mov 指令))

= 3 MIPS 附加工作负荷

12.35.3 方法 3：时基周期的软件缩放

此方法使用软件缩放时基周期。假设抖动速率相对较慢（大约 250 Hz），应用控制环应能对 PWM 周期的变化进行补偿，并相应地调整占空比。

12.35.4 方法 4：频率调制

此方法在改变 PWM 周期（抖动）时改变频率。当分析小时间窗口时，频率调制过程与相位调制相似（从数学上讲）。

PWM 模块具有通过相移寄存器对 PWM 信号进行相位调制的能力。相位调制具有软件简单且速度快的优点，因为多次乘法运算（用于通过缩放周期和占空比抖动频率）被较少次数的加法或对相位寄存器中相移值的简单更新所取代。

此方法也有下列优点：

1. 仍可创建多相和可变相位 PWM 模式。
2. PWM 发生器仍可使用共用时基，这可简化对何时是“静止时间”作出判断以进行电流测量。

此方法有一个缺点：相位调制必须工作在相对较高的更新速率下，以达到可用的频率分布范围。

12.35.5 独立 PWM 通道抖动问题：

使用独立输出抖动的多相或可变相设计必须考虑下列问题：

1. 这些相位不再对齐。
2. 对相位之间共享电流的控制更困难。

12.36 外部同步功能

在大型电源转换系统中，使多个电源控制器同步通常是很理想的，因为这可以确保系统中不产生“拍频”，或作为确保产生能进行电流和电压测量的“静止”周期的方法。

dsPIC30F202X 器件（28 引脚封装除外）的输入和 / 或输出引脚提供使 SMPS dsPIC DSC 器件与外部器件同步，或使外部器件与 SMPS dsPIC DSC 同步的功能。这些同步功能通过 PWM 模块的 PTCN 控制寄存器中的 SYNCIEN 和 SYNCOEN 位使能。

PTCN 寄存器中的 SYNCPOL 位选择 SYNCI 信号的上升沿还是下降沿为有效沿。PTCN 寄存器中的 SYNCPOL 位还选择 SYNCO 输出脉冲是低电平有效还是高电平有效。

PTCN 寄存器中的 SYNCSRC<2:0> 位指定 SYNCI 信号的源。

若使能了 SYNCI 功能，则当检测到有效 SYNCI 边沿时，主时基计数器复位。若使能了 SYNCO 功能，则在 PWM 周期末主时基计数器计满归零时产生输出脉冲。

建议的 SYNCI 脉冲宽度应大于 100 ns。预期的 SYNCO 输出脉冲宽度接近 100 ns。

使用 SYNCI 功能时，建议用户用比外部同步输入信号的预期周期稍大的周期值编程周期寄存器。这能够在由于噪声或外部元件故障造成 SYNCI 信号接收失败的情况下提供保护。将合理的周期值编程到 PTPER 寄存器中时，即使未接收到全局同步信号，本地电源转换过程也应继续进行。

12.37 CPU 负载切换

SMPS dsPIC DSC 具有切换各触发比较操作的能力。此功能有助于平均分配处理器工作量以将处理器超载的情况降至最少。

假设有一种情况由 4 个 PWM 通道控制 4 个独立的电压输出。进一步假设各 PWM 发生器工作频率为 1000 kHz（周期为 1 μs），且各控制环的工作频率为 125 kHz（8 μs）。

各 TRGCONx 寄存器中的 TRGDIV<2:0> 位设置为 111，这将选择每第 8 个触发比较匹配将产生一个 ADC 触发信号以捕捉数据并开始转换过程。

若不存在时间偏移（stagger-in-time）功能，所有 PWM 触发寄存器的所有请求可能会同时发生。若发生了“堆积”，某些数据采样将会超过处理所有 4 个通道的数据的时间。

有了此时间偏移功能，触发信号在后续 PWM 周期内根据时间排列，以使所有数据能有序处理。

ROLL 计数器是一种连接到主时基计数器的计数器。主时基计数器每计数到终值（计满），ROLL 计数器就递增一次。

时间偏移功能由 TRGCONx 寄存器中的 TRGSTRT<5:0> 位控制。TRGSTRT<5:0> 位指定 ROLL 计数器的计数值，此值必须在各 PWM 发生器中的独立触发比较模块开始计数触发比较事件（由 PWMCONx 寄存器中的 TRGDIV<2:0> 位指定）之前匹配。

因此，在 4 个 PWM 发生器的示例中，第一个 PWM 的 TRGSTRT<5:0> 位应为 000，第二个 PWM TRGSTRT 位应置为 010，第三个 PWM TRGSTRT 位应置为 100，第四个 PWM TRGSTRT 位应置为 110。所以，在总共 8 个 PWM 周期中，4 个独立的控制环应分别以自己的 2 μs 时间周期工作。

12.38 外部触发明隐

使用 LEBCONx 寄存器中的 LEB<9:3> 位，PWM 模块能在 0 到 1024 ns 的周期内消隐（忽略）外部电流和故障输入。当功率晶体管导通时所感应的瞬态电流使在 PWM 周期开始时电流检测困难的情况下，此功能很有用。

表 12-4: 电源 PWM 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时			
PTCON	0400	PTEN	---	PTSIDL	SESTAT	SEIEN	EIPU	SYNCPOL	SYNCOEN	SYNCHEN	SYNCSRC<2:0>							SEVTPS<3:0>			0000
PTPER	0402	PTPER<15:3>																	FFF0		
MDC	0404	MDC<15:0>																	0000		
SEVTCMP	0406	SEVTCMP<15:3>																	0000		
PWMCON1	0408	FLTSTAT	CLSTAT	TRGSTAT	FLTEN	CLLEN	TRGIEN	ITB	MDCS	DTC<1:0>	---		---	---	---	XPRES	IUE	0000			
IOCON1	040A	PENH	PENL	POLH	POLL	PMOD<1:0>	OVRENH	OVRENH	OVRENH	OVRENH	CLPOL	CLMOD	CLMOD	CLMOD	CLMOD	CLMOD	OSYNC	0000			
FCLCON1	040C	---	---	---	---	CLSRC<3:0>		CLPOL		CLMOD		FLTSRC<3:0>		FLTPOL		FLTMOD<1:0>		0000			
PDC1	040E	PDC1<15:0>																	0000		
PHASE1	0410	PHASE1<15:2>																	0000		
DTR1	0412	---	---	DTR1<13:2>														---		---	0000
ALTDTR1	0414	---	---	ALTDTR1<13:2>														---		---	0000
TRIG1	0416	TRIG<15:3>																	0000		
TRGCON1	0418	TRGDIV<2:0>		---	---	---	---	---	---	---	---	---	---	TRGSTRT<5:0>			0000				
LEBCON1	041A	PHR	PHF	PLR	PLF	FLTLBEN	CLLBEN	LEB<9:3>		---								---	0000		
PWMCON2	041C	FLTSTAT	CLSTAT	TRGSTAT	FLTEN	CLLEN	TRGIEN	ITB	MDCS	DTC<1:0>	---		---	---	---	XPRES	IUE	0000			
IOCON2	041E	PENH	PENL	POLH	POLL	PMOD<1:0>	OVRENH	OVRENH	OVRENH	OVRENH	CLPOL	CLMOD	CLMOD	CLMOD	CLMOD	CLMOD	OSYNC	0000			
FCLCON2	0420	---	---	---	---	CLSRC<3:0>		CLPOL		CLMOD		FLTSRC<3:0>		FLTPOL		FLTMOD<1:0>		0000			
PDC2	0422	PDC2<15:0>																	0000		
PHASE2	0424	PHASE2<15:2>																	0000		
DTR2	0426	---	---	DTR2<13:2>														---		---	0000
ALTDTR2	0428	---	---	ALTDTR2<13:2>														---		---	0000
TRIG2	042A	TRIG<15:3>																	0000		
TRGCON2	042C	TRGDIV<2:0>		---	---	---	---	---	---	---	---	---	---	TRGSTRT<5:0>			0000				
LEBCON2	042E	PHR	PHF	PLR	PLF	FLTLBEN	CLLBEN	LEB<9:3>		---								---	0000		
PWMCON3	0430	FLTSTAT	CLSTAT	TRGSTAT	FLTEN	CLLEN	TRGIEN	ITB	MDCS	DTC<1:0>	---		---	---	---	XPRES	IUE	0000			
IOCON3	0432	PENH	PENL	POLH	POLL	PMOD<1:0>	OVRENH	OVRENH	OVRENH	OVRENH	CLPOL	CLMOD	CLMOD	CLMOD	CLMOD	CLMOD	OSYNC	0000			
FCLCON3	0434	---	---	---	---	CLSRC<3:0>		CLPOL		CLMOD		FLTSRC<3:0>		FLTPOL		FLTMOD<1:0>		0000			
PDC3	0436	PDC3<15:0>																	0000		
PHASE3	0438	PHASE3<15:2>																	0000		
DTR3	043A	---	---	DTR3<13:2>														---		---	0000
ALTDTR3	043C	---	---	ALTDTR3<13:2>														---		---	0000
TRIG3	043E	TRIG<15:3>																	0000		
TRGCON3	0440	TRGDIV<2:0>		---	---	---	---	---	---	---	---	---	---	TRGSTRT<5:0>			0000				
LEBCON3	0442	PHR	PHF	PLR	PLF	FLTLBEN	CLLBEN	LEB<9:3>		---								---	0000		
PWMCON4	0444	FLTSTAT	CLSTAT	TRGSTAT	FLTEN	CLLEN	TRGIEN	ITB	MDCS	DTC<1:0>	---		---	---	---	XPRES	IUE	0000			
IOCON4	0446	PENH	PENL	POLH	POLL	PMOD<1:0>	OVRENH	OVRENH	OVRENH	OVRENH	CLPOL	CLMOD	CLMOD	CLMOD	CLMOD	CLMOD	OSYNC	0000			

表 12-4: 电源 PWM 寄存器映射 (续)

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复 位时
FCLCON4	0448	---	---	---		CLSRC<3:0>			CLPOL	CLMODE		FLTSRC<3:0>			FLTPOL	FLTMOD<1:0>		0000
PDC4	044A							PDC4<15:0>										0000
PHASE4	044C							PHASE4<15:2>										0000
DTR4	044E	---	---					DTR4<13:2>										0000
ALTDTR4	0450	---	---					ALTDTR4<13:2>										0000
TRIG4	0452							TRIG<15:3>										0000
TRGCON4	0454	TRGDIV<2:0>													TRGSTRT<5:0>			0000
LEBCON4	0456	PHR	PHF	PLR	PLF	FLTLBEN	CLLBEN			LEB<9:3>								0000
保留	0458-47F	---	---	---	---	---	---											0000

注:

13.0 串行外设接口 (SPI)

注： 本数据手册总结了 dsPIC30F1010/202X 系列器件的功能，但并不作为参考大全使用。要了解本数据手册的补充信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

串行外设接口 (Serial Peripheral Interface, SPI) 模块是同步串行接口，可用于与其他外设 (如 EEPROM、移位寄存器、显示驱动器和 A/D 转换器等) 或者单片机进行通信。SPI 模块与 Motorola® 的 SPI 和 SIOP 接口兼容。

注： dsPIC30F101/202X 系列只有一个 SPI。所有对 $x = 2$ 的引用是为了在软件上与其他 dsPIC DSC 器件兼容。

SPI 模块由一个 16 位移位寄存器 (SPIxSR, 其中 $x = 1$ 或 2) 和一个缓冲器寄存器 (SPIxBUF) 组成；移位寄存器用来移入和移出数据。用两个控制寄存器 SPIxCON1 和 SPIxCON2 来配置模块。用户软件不能访问 SPIxSR 寄存器。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) 以及 SSx (低电平有效从选择)。

在主模式下工作时，SCK 是时钟输出，但在从模式下它是时钟输入。

一组 8 或 16 个时钟脉冲将数据位从 SPIxSR 移出到 SDOx 引脚，同时将 SDIx 引脚的数据位移入 SPIxSR。当传输完成后将产生中断，相应的中断标志位 (SPI1IF 或 SPI2IF) 置 1。通过中断允许位 (SPI1IE 或 SPI2IE) 可以禁止该中断。

接收操作是双缓冲的。当接收完字节时，将字节从 SPIxSR 送到 SPIxBUF。

当从 SPIxSR 传输新数据到 SPIxBUF 时，如果接收缓冲器满，模块将置 1 SPIROV (SPIxSTAT<6>) 位，表明产生溢出条件。数据从 SPIxSR 向 SPIxBUF 的传送不会完成，新数据将丢失。当 SPIROV (SPIxSTAT<6>) 为 1 时，模块将不会响应 SCKx 引脚上电平的跳变，这样就有效地禁止了模块，直到用户软件读 SPIxBUF 为止。

发送写操作同样也是双缓冲的。用户软件写入 SPIxBUF。当主器件或从器件传输完成后，移位寄存器 (SPIxSR) 的内容将送至接收缓冲器。如果已经向缓冲

器寄存器写了任何发送数据，发送缓冲器的内容都将送至 SPIxSR。于是，接收到的数据存放在 SPIxBUF 中，而 SPIxSR 中的发送数据已经准备就绪，可进行下一次传输。

注： 发送缓冲器 (SPIxTXB) 和接收缓冲器 (SPIxRXB) 都映射到相同的寄存器地址 SPIxBUF。不要对 SPIxBUF 寄存器执行读 - 修改 - 写操作 (如针对位操作的指令)。

遵循以下步骤可将 SPI 模块设置为工作在主模式下：

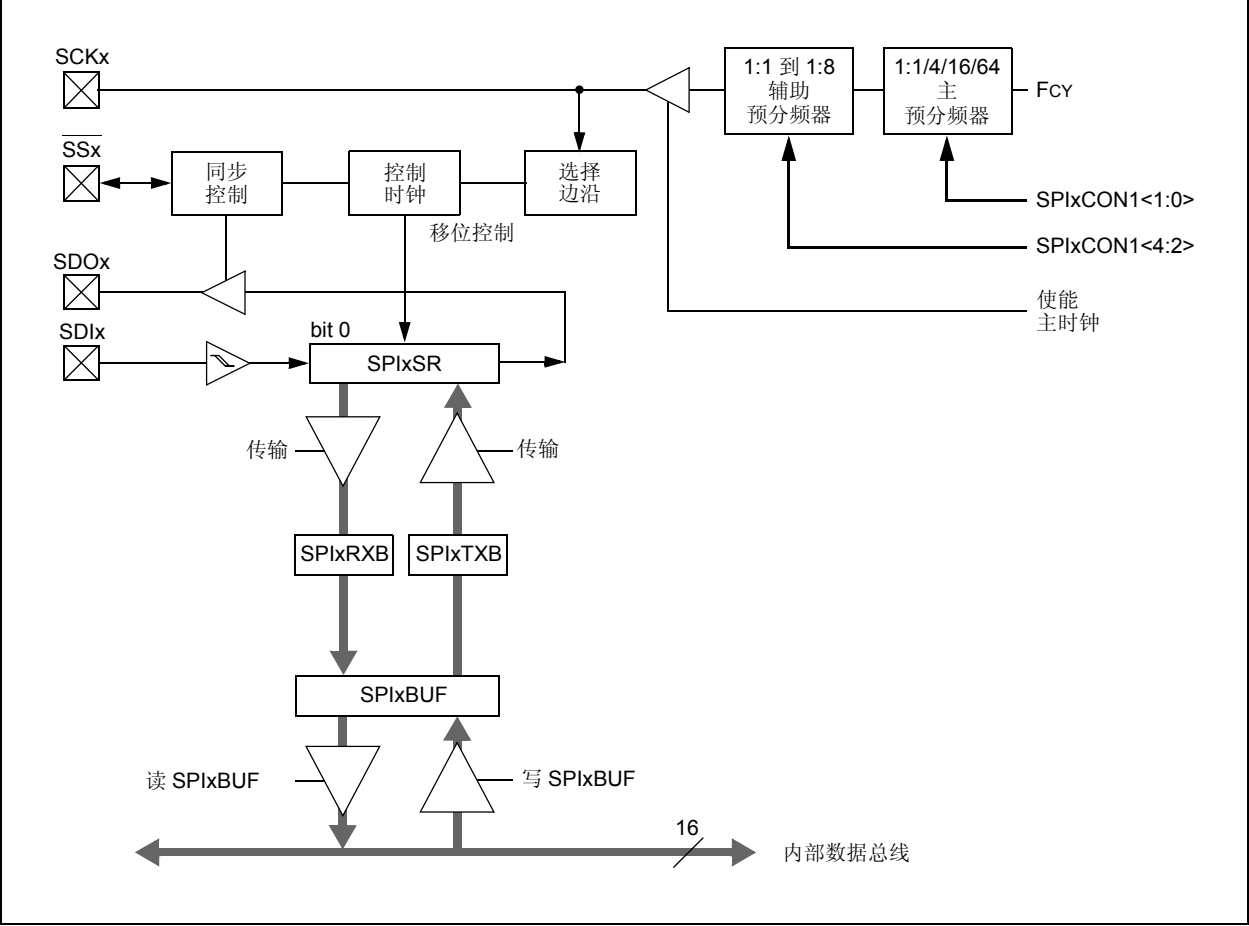
1. 如果使用中断：
 - a) 将相应 IFSn 寄存器中的 SPIxIF 位清零。
 - b) 将相应 IECn 寄存器中的 SPIxIE 位置 1。
 - c) 通过写相应 IPCn 寄存器中的 SPIxIP 位来设置中断的优先级。
2. 将要求的设置写入 SPIxCON 寄存器，且 MSTEN (SPIxCON1<5>) = 1。
3. 将 SPIROV 位 (SPIxSTAT<6>) 清零。
4. 通过将 SPIEN 位 (SPIxSTAT<15>) 置 1 使能 SPI 工作。
5. 将待发送数据写入 SPIxBUF 寄存器。数据一写入 SPIxBUF 寄存器发送 (和接收) 就会立即开始。

遵循以下步骤可将 SPI 模块设置为工作在从模式下：

1. 将 SPIxBUF 寄存器清零。
2. 如果使用中断：
 - a) 将相应 IFSn 寄存器中的 SPIxIF 位清零。
 - b) 将相应 IECn 寄存器中的 SPIxIE 位置 1。
 - c) 通过写相应 IPCn 寄存器中的 SPIxIP 位来设置中断的优先级。
3. 将要求的设置写入 SPIxCON1 和 SPIxCON2 寄存器，且 MSTEN (SPIxCON1<5>) = 0。
4. 将 SMP 位 (SPIxCON1<9>) 清零。
5. 如果将 CKE 位 (SPIxCON1<8>) 置 1，然后必须将 SSx 位 (SPIxCON1<7>) 置 1 来使能 SSx 引脚。
6. 将 SPIROV 位 (SPIxSTAT<6>) 清零。
7. 通过将 SPIEN 位 (SPIxSTAT<15>) 置 1 使能 SPI 工作。

SPI 模块会产生一个中断表明字节或字传输完成，还会为所有的 SPI 错误条件产生单独的中断。

图 13-1: SPI 模块框图



注： dsPIC30F1010/2020 器件不包含 $\overline{\text{SS1}}$ 引脚。因此，在这些器件上不能使用从选择和帧同步功能。但 dsPIC30F2023 器件可以使用这些功能。

图 13-2: SPI 主 / 从连接

dsPIC30F1010/202X

图 13-5: SPI 从动—帧主模式连接图

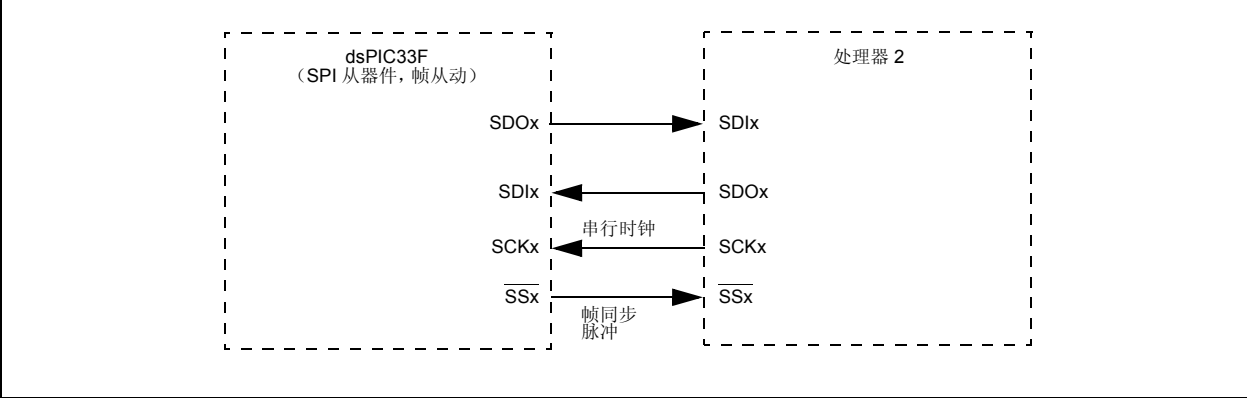
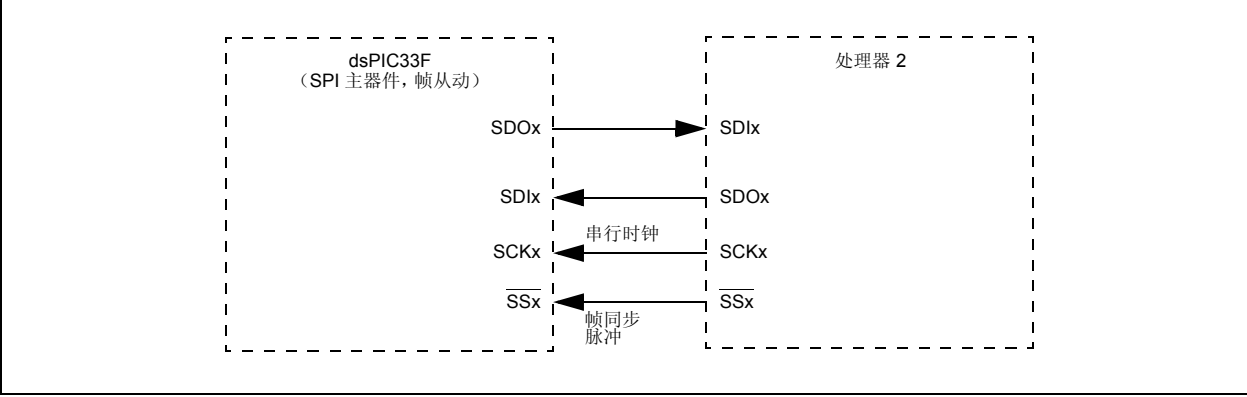


图 13-6: SPI 从动—帧从模式连接图



公式 13-1: 器件工作频率和 SPI 时钟速度之间的关系

$$F_{SCK} = \frac{F_{CY}}{\text{主预分频比} * \text{辅助预分频比}}$$

表 13-1: 采样 SCKx 频率

Fcy = 40 MHz		辅助预分频比设置				
		1:1	2:1	4:1	6:1	8:1
主预分频比设置	1:1	无效	无效	7500	5000	3750
	4:1	7500	3750	1875	1250	937.5
	16:1	1875	937.5	469	312.5	234.4
	64:1	469	234.4	117	78.1	58.6
Fcy = 5 MHz						
主预分频比设置	1:1	5000	2500	1250	833	625
	4:1	1250	625	313	208	156
	16:1	313	156	78	52	39
	64:1	78	39	20	13	10

注：表中 SCKx 频率的单位为 kHz。

寄存器 13-1: 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 = 上电复位时的值	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 $\overline{SSx}$ 配置为串行端口引脚
0 = 禁止模块
- bit 14 **未实现:** 读为 0
- bit 13 **SPISIDL:** 空闲模式下的停止位
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 发送缓冲器满状态位
1 = 发送尚未开始, SPIxTXB 为满
0 = 开始发送, SPIxTXB 为空
当 CPU 写 SPIxBUF 地址单元并装载 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 接收缓冲器满状态位
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 地址单元读 SPIxRXB 时, 该位由硬件自动清零。

dsPIC30F1010/202X

寄存器 13-2: 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 = 上电复位时的值

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, 引脚作为 I/O 口使用

0 = 使能内部 SPI 时钟

bit 11 **DISSDO:** SDOx 引脚禁止位

1 = SDOx 引脚未由模块使用; 引脚作为 I/O 口使用

0 = SDOx 引脚由模块控制

bit 10 **MODE16:** 字 / 字节通信选择位

1 = 通信为字宽 (16 位)

0 = 通信为字节宽 (8 位)

bit 9 **SMP:** SPIx 数据输入采样相位位

主模式:

1 = 输入数据在数据输出时间末尾采样

0 = 输入数据在数据输出时间中间采样

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:** 从动选择使能 (从模式) 位

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

bit 4-2 **SPRE<2:0>:** 辅助预分频比 (主模式) 位

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 (主模式) 位

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, 用户应该将该位编程为 0。

寄存器 13-3: 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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **FRMEN:** 帧 SPIx 支持位
 1 = 使能帧 SPIx 支持 ($\overline{\text{SSx}}$ 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:** 禁止使用用户应用程序将该位置 1。

表 13-2: SPI1 寄存器映射

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态
SPI1STAT	0240	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000 0000 0000 0000
SPI1CON	0242	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	—	SPRE<2:0>	—	PPRE<1:0>	—	0000 0000 0000 0000
SPI1CON2	0244	FRMEN	SPIFSD	FRMPOL	—	—	—	—	—	—	—	—	—	—	—	FRMDLY	—	0000 0000 0000 0000
SPI1BUF	0246	发送和接收缓冲器																0000 0000 0000 0000

图注: u = 未初始化的位

注: 有关寄存器位字段的描述, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

14.0 I²C™ 模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

I²C 模块（16 位接口）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。

模块具有以下主要特性：

- I²C 接口支持主、从模式工作。
- I²C 从模式支持 7 位和 10 位地址。
- I²C 主模式支持 7 位和 10 位地址。
- 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继续串行传输（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。

14.1 功能说明

硬件完全实现了 I²C 标准和快速模式规范的全部主从功能，以及 7 位和 10 位寻址。

从而，I²C 模块既可作为 I²C 总线上的主器件工作，也可作为 I²C 总线上的从器件工作。

14.1.1 各种 I²C 模式

支持 I²C 模块工作在以下模式：

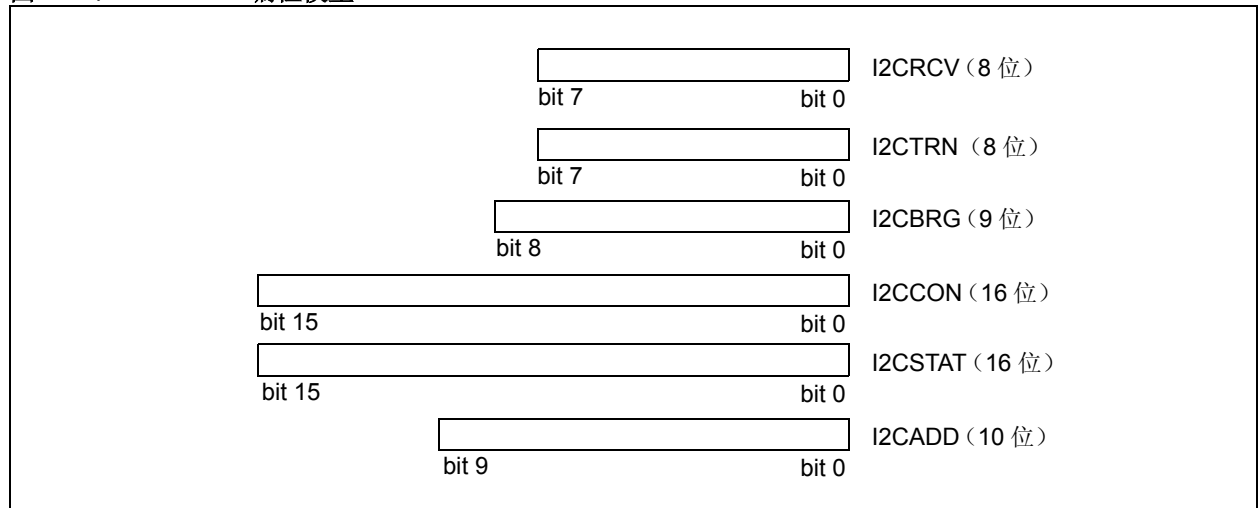
- 7 位或 10 位地址的 I²C 从模式
- 7 位或 10 位地址的 I²C 主模式

请参见图 14-1 中的 I²C 编程模型。

14.1.2 I²C 模式下的引脚配置

I²C 有一个双引脚接口：SCL 引脚为时钟线，而 SDA 引脚为数据线。

图 14-1: 编程模型



14.1.3 I²C 寄存器

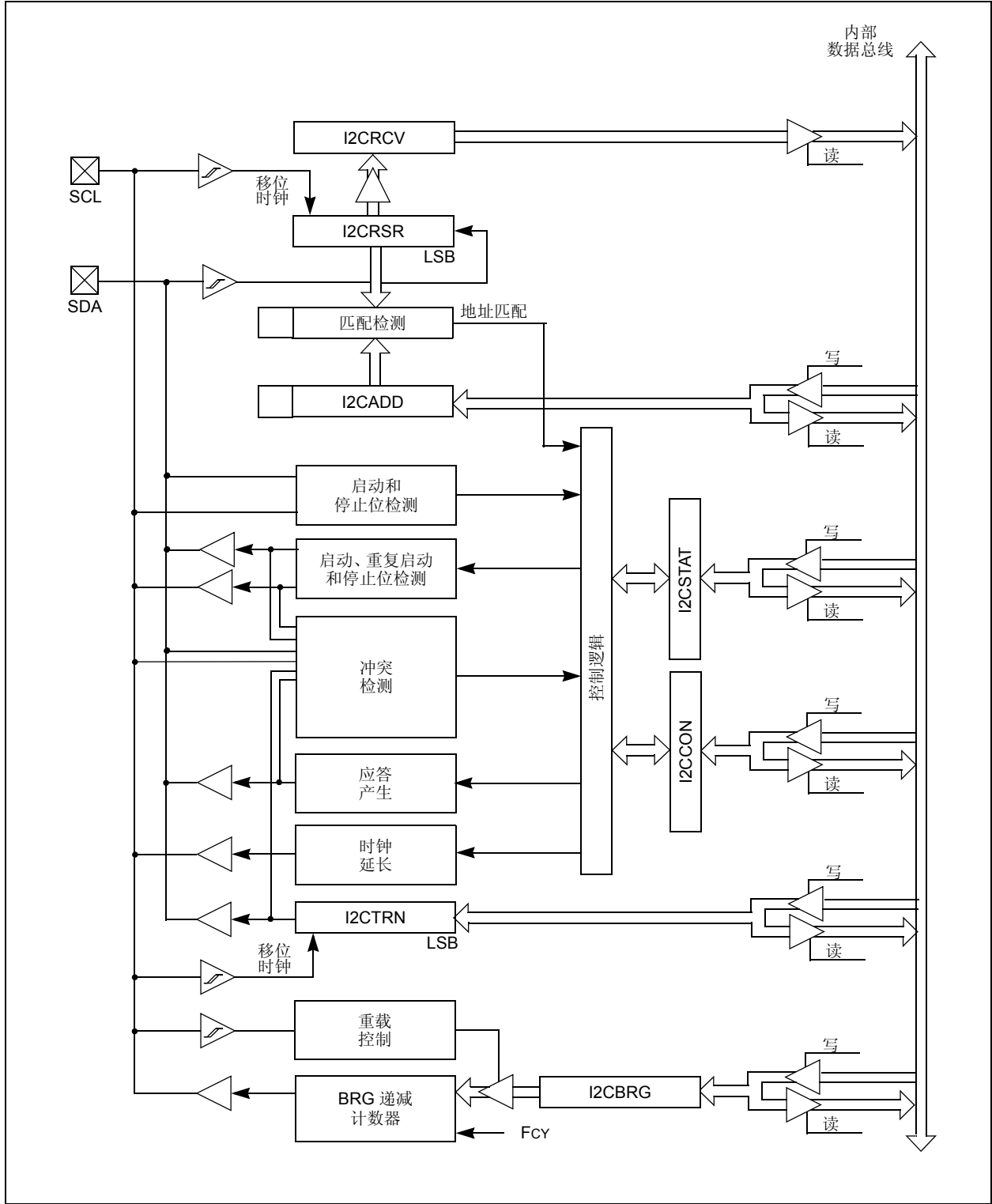
I2CCON 和 I2CSTAT 分别是控制寄存器和状态寄存器。I2CxCON 是可读写的寄存器。I2CSTAT 的低 6 位是只读的，它的其余位则是可读且可写的。

I2CRSR 是移位寄存器，用来对数据进行移位，而 I2CRCV 是接收缓冲寄存器，可写入、读出数据字节，如图 14-1 所示。I2CTRN 是发送寄存器，发送操作期间，字节将会写入 I2CTRN，如图 14-2 所示。

I2CADD 寄存器存放从地址。状态位 ADD10 表明是否为 10 位地址模式。I2CBRG 用来保存波特率发生器（Baud Rate Generator, BRG）的重载值。

接收时，I2CRSR 和 I2CRCV 共同形成双缓冲接收器。在 I2CRSR 接收到一个完整的字节后，字节被送至 I2CRCV 并产生中断脉冲。在发送期间，I2CTRN 不是双缓冲的。

注：在 10 位寻址方式下，在重复启动条件之后，用户只需匹配前 7 位地址。

图 14-2: I²C™ 框图

14.2 I²C 模块地址

I2CADD 寄存器包含从模式地址，它是 10 位宽的寄存器。

如果 A10M 位 (I2CCON<10>) 为 0，模块把地址解释为 7 位地址。接收地址时，将接收到的地址与 I2CADD 寄存器的低 7 位进行比较。

如果 A10M 位为 1，将认为地址是 10 位地址。在接收地址时，将收到的地址与二进制数 “1 1 1 1 0 A9 A8” (其中 A9 和 A8 是 I2CADD 的最高两位) 进行比较。如果匹配，则按 10 位寻址协议的规定，下一地址同 I2CxADD 的低 8 位比较。

14.3 I²C 7 位从模式的工作

一旦使能 (I2CEN = 1)，从模块将等待一个启动位出现 (即，I²C 模块 “空闲”)。检测到启动位之后，8 个位将移入 I2CRSR，然后地址将与 I2CADD 进行比较。在 7 位寻址方式下 (A10M = 0)，位 I2CADD<6:0> 与位 I2CRSR<7:1> 进行比较，而 I2CRSR<0> 是 R_W 位。所有输入位都是在 SCL 的上升沿采样到的。

如果地址匹配的话，将发送一个应答，在第九个位 (ACK) 的下降沿，从事件中断标志 (SI2CIF) 置 1。地址匹配不影响 I2CRCV 缓冲器或 RBF 位的内容。

14.3.1 从模式发送

如果收到的 R_W 位为 1，串行端口将进入发送模式。模块将在第九位发送 ACK 信号，并使 SCL 保持为 0，直到 CPU 写 I2CTRNL 进行响应为止。通过把 SCLREL 置 1 释放 SCL，而后移出数据的 8 个位。数据位在 SCL 下降沿移出，因此 SCL 为高 (见时序图) 时 SDA 有效。将在第九个时钟脉冲的下降沿发出中断脉冲，而与从主器件接收到的 ACK 状态无关。

14.3.2 从模式接收

如果地址匹配时，接收到的 R_W 位是 0，就将启动接收模式。在 SCL 上升沿采样输入位。接收到 8 个位之后，如果 I2CRCV 未满足或者 I2COV 没有置 1，I2CRSR 中的数据将被送至 I2CRCV。在第九个时钟发送 ACK。

如果 RBF 标志置 1，表明 I2CRCV 仍然存有前一次收到的数据 (RBF = 1)，则不发送 ACK；然而，仍将产生中断脉冲。如果出现溢出，I2CRSR 的内容不会被装载到 I2CRCV 中。

注： 如果 I2COV 位为 1 且 RBF 标志为 0，将装入 I2CRCV。在这种情况下，要对 I2CRCV 进行读操作，但在下一次接收操作发生之前，用户不会清除 I2COV 位的状态，不发送应答 (ACK = 1)，但 I2CRCV 位会被更新。

14.4 I²C 10 位从模式的工作

在 10 位寻址方式下，基本的接收和发送操作与 7 位寻址方式下的操作相同。不过，地址匹配的判据更加复杂。

I²C 规范要求，对于写操作，必须在一个启动位后跟两个地址字节来寻址从器件。

A10M 控制位置 1 表明 I2CADD 中的地址是 10 位地址，而不是 7 位地址。报文地址首字节的地址检测协议，对于 7 位和 10 位报文地址而言，是完全相同的，但首字节各位进行比较的机制不同。

I2CADD 存放完整的 10 位地址。在接收到启动位后的地址时，I2CRSR <7:3> 与立即数 11110 (缺省 10 位地址) 进行比较，I2CRSR<2:1> 与 I2CADD<9:8> 进行比较。如果出现匹配且 R_W = 0，则发出中断脉冲。ADD10 位清零，表明地址部分匹配。如果地址不匹配，或 R_W = 1，则 ADD10 位清零，模块返回到空闲状态。

然后，接收地址的低字节，并与 I2CADD<7:0> 进行比较。如果出现匹配，则产生中断脉冲且置 1 ADD10 位，表明 10 位地址完全匹配。如果地址不匹配，ADD10 位清零，模块返回到空闲状态。

14.4.1 10 位地址从模式接收

一旦以上述方式使用完整的 10 位地址寻址到从器件时 (我们把这种状态叫做 “PRIOR_ADDR_MATCH”)，主器件就可以开始发送数据字节，供从器件接收。

14.4.2 10 位地址从模式发送

一旦寻址到从器件，主器件就可以产生重复的启动位，复位地址的高字节，并置 1 R_W 位，但不产生停止位，于是开始进行从器件发送操作。

14.5 自动时钟延长

当模块工作在从模式时，通过时钟延长，模块能够同步读缓冲器与写主器件。

14.5.1 发送时钟延长

在 10 位和 7 位发送模式下，如果 TBF 位被清零（表明缓冲器为空），则在第 9 个时钟的下降沿后清零 SCLREL 位，即实现了时钟延长。

在从发送模式下，始终进行时钟延长，而与 STREN 位的状态无关。

在发送序列的第九个时钟之后，发生时钟同步。如果器件在第九个时钟的下降沿采样到 ACK，并且 TBF 位仍然清零的话，则 SCLREL 位将自动清零，从而使 SCL 线变为低电平。在允许继续发送之前，用户中断服务程序必须置 1 SCLREL 位。通过使 SCL 线保持为低电平，使用户在主器件启动另一个发送序列之前有时间执行中断服务程序，并装载 I2CTRN 的内容。

- 注 1:** 如果用户载入 I2CTRN 的内容，在第九个时钟下降沿之前置 1 TBF 位，则 SCLREL 位将不会被清零，也不会发生时钟延长。
- 2:** 无论 TBF 位的状态为何都可在软件中将 SCLREL 位置 1。

14.5.2 接收时钟延长

I2CCON 寄存器中的 STREN 位可以用来使能从接收模式下的时钟延长。当 STREN 位置 1 时，SCL 引脚将在每个数据接收序列结束后保持为低电平。

14.5.3 7 位寻址的时钟延长（STREN = 1）

从接收模式下，当 STREN 置 1 时，缓冲器已满时 SCL 线保持为低电平。对于 7 位和 10 位寻址模式，延长 SCL 输出的方法是相同的。

在接收序列的第九个时钟之后发生时钟延长。在 ACK 序列末端的第九个时钟的下降沿，如果 RBF 位置 1，SCLREL 位将自动清零，从而强制将 SCL 输出保持为低电平。在允许继续接收之前，用户中断服务程序必须置 1 SCLREL 位。通过使 SCL 线保持为低电平，使用户在主器件启动另一个接收序列之前有时间执行中断服务程序，并读出 I2CRCV 的内容，这将防止发生缓冲器溢出。

- 注 1:** 如果用户读 I2CRCV 的内容，并在第九个时钟下降沿之前置 1 RBF 位，则 SCLREL 位将不会被清零，也不会发生时钟延长。
- 2:** 可以用软件置 1 SCLREL 位，而与 RBF 位的状态无关。在下一个接收序列之前，用户应该在中断服务程序中清零 RBF 位，以避免溢出。

14.5.4 10 位寻址的时钟延长（STREN = 1）

在寻址过程中自动发生时钟延长。由于模块有寄存器存放完整的地址，协议不必等待地址更新。

寻址过程完成后，每个数据接收或发送序列都要进行时钟延长，如前所述。

14.6 软件控制的时钟延长（STREN = 1）

当 STREN 位为 1 时，可用软件清零 SCLREL 位，以允许通过软件控制时钟延长。思路是把写 SCLREL 位的操作与 SCL 时钟进行同步。清零 SCLREL 位不会使 SCL 输出保持为低，除非模块检测到 SCL 输出的下降沿且采样到 SCL 为低。在 SCL 线采样为低时，如果用户清零 SCLREL 位，则 SCL 输出将保持为低。SCL 输出将保持为低，一直到 SCLREL 位置 1，且 I²C 总线上所有其他器件都已经释放了 SCL。这确保了写 SCLREL 位的操作，不会违背 SCL 的最小高电平时间要求。

如果 STREN 位为 0，软件对 SCLREL 位的写操作将被忽略，不会对 SCLREL 位产生影响。

14.7 中断

I²C 模块可产生两个中断标志：MI2CIF（I²C 主中断标志）和 SI2CIF（I²C 从中断标志）。MI2CIF 中断标志，会在主器件报文事件完成时置 1。SI2CIF 中断标志，在检测到发往从器件的报文时置 1。

14.8 斜率控制

对于快速模式（400 kHz），I²C 标准要求对 SDA 和 SCL 信号进行斜率控制。如果需要的话，用户可通过控制位 DISSLW 来禁止斜率控制。对于 1 MHz 模式，必须禁止斜率控制。

14.9 IPMI 支持

控制位 IPMIEN 允许模块支持智能外设管理接口（Intelligent Peripheral Management Interface, IPMI）。当此位置 1 时，模块将接受并操作所有的地址。

14.10 广播呼叫地址支持

广播呼叫地址能寻址所有器件。当使用广播呼叫地址时，理论上所有器件都应对此信号作出应答。

广播呼叫地址是 I²C 协议为特定目的保留的八个地址之一。它由全 0 组成，且 R_W = 0。

当广播呼叫使能位 GCEN 置 1（I2CCON<7> = 1）时，识别广播呼叫地址。在检测到启动位后，将 8 位移入 I2CRSR，并将地址与 I2CADD 进行比较，同时也与固化在硬件中的广播呼叫地址进行比较。

如果广播呼叫地址匹配，第八个时钟之后 I2CRSR 被送至 I2CRCV，在第九个位（ACK 位）的下降沿将 RBF 标志置 1，主事件中中断标志（MI2CIF）置 1。

在响应中断时，通过读 I2CRCV 的内容可以检测到中断源，从而确定地址是特定于器件的地址还是广播呼叫地址。

14.11 I²C 主模式支持

用作主器件时，支持下列 6 个操作：

- 在 SDA 和 SCL 上产生启动条件。
- 在 SDA 和 SCL 上产生重复启动条件。
- 写 I2CTRN 寄存器以启动数据 / 地址的发送。
- 在 SDA 和 SCL 上产生停止条件。
- 配置 I²C 端口，用于接收数据。
- 在数据字节接收完成后产生 ACK 条件。

14.12 I²C 主模式下的工作

主器件产生所有的串行时钟脉冲以及启动和停止条件。用停止条件或重复的启动条件结束传输。由于重复的启动条件也是下一次串行传输的开始，所以此时不会释放 I²C 总线。

在主发送模式下，由 SDA 输出串行数据，SCL 输出串行时钟。发送的第一个字节包含接收器件的从地址（7 位）和数据方向位。此时，数据方向位（R_W）为逻辑 0。每次发送 8 位串行数据。在每个字节发送完之后，接收到一个 ACK 位。输出启动条件和停止条件，来表明串行传输的开始和结束。

在主接收模式下，发送的第一个字节包含发送器件的从地址（7 位）和数据方向位。此时，数据方向位（R_W）为逻辑 1。这样，发送的第一个字节为一个 7 位从地址，后跟一个表示接收的位 1。通过 SDA 接收串行数据，SCL 输出串行时钟。每次接收 8 位串行数据。在每个字节接收完之后，都要发送一个 ACK 位。启动条件和停止条件表明发送的开始和结束。

14.12.1 I²C 主发送

数据字节、7 位地址或 10 地址后半部分的发送，只需把值写入 I2CTRN 寄存器就能完成。当模块处于等待状态时，用户应只写 I2CTRN。此操作将置 1 缓冲器满标志（TBF），允许波特率发生器开始计数，并开始下一次发送。在 SCL 出现下降沿后，地址 / 数据的每一位都将被移出至 SDA 引脚上。发送状态标志位 TRSTAT（I2CSTAT<14>）用来表明主器件是否正在进行发送。

14.12.2 I²C 主接收

通过设置接收使能位 RCEN（I2CCON<3>）可以使能主模式接收。在 RCEN 置 1 之前，I²C 模块必须是空闲的，否则，RCEN 位被忽略。波特率发生器开始计数，每次计满返回到零时，在每个时钟的上升沿，SCL 引脚状态发生翻转，数据移入 I2CRSR。

14.12.3 波特率发生器

在 I²C 主模式下，波特率发生器的重载值位于 I2CBRG 寄存器中。波特率发生器装入该值后，发生器递减计数到 0，然后停止，直到发生再次装入。如果发生时钟仲裁，例如 SCL 引脚采样为高电平时，将重载波特率发生器。

根据 I²C 标准，FSCK 可以是 100 kHz 或 400 kHz。但用户可以指定任何波特率，最高为 1 MHz。I2CBRG 的值不能是 0 或 1。

公式 14-1: I2CBRG 值

$$I2CBRG = \left(\frac{F_{CY}}{F_{SCL}} - \frac{F_{CY}}{1,111,111} \right) - 1$$

14.12.4 时钟仲裁

在任何接收、发送或重复启动 / 停止条件期间，当主器件释放 SCL 引脚（允许 SCL 悬空为高电平）时，就会出现时钟仲裁。当释放 SCL 引脚时，波特率发生器暂停计数，直到 SCL 引脚被实际采样到高电平为止。当 SCL 引脚被采样到高电平时，波特率发生器重新装载 I2CBRG 的内容并开始计数。在外部器件使时钟保持为低时，这可以始终保证 SCL 高电平时间至少为一个 BRG 计满返回计数周期。

14.12.5 多主器件通信、总线冲突与总线仲裁

多主器件工作的支持，是通过总线仲裁来实现的。当主器件在 SDA 引脚上输出地址 / 数据位时，第一个主器件使 SDA 悬空为高电平从而输出 1，而另一个主器件要输出 0，就会发生仲裁。当 SCL 引脚悬空为高时，数据应该是稳定的。如果 SDA 上预期数据是 1，但从 SDA 引脚采样到的数据是 0，那么就发生了总线冲突。主器件将置 1 MI2CIF 脉冲，并把 I²C 端口的主器件部分复位到空闲状态。

如果正在进行发送时产生总线冲突，将停止发送，清除 TBF 标志，释放 SDA 和 SCL 线，而后可向 I2CTRN 写入数据。当用户执行 I²C 主事件中断服务程序时，如果 I²C 总线空闲（即 P 位置 1），用户可以通过产生一个启动条件继续通信。

若在启动、重复启动、停止或应答条件期间发生总线冲突，将中止这些条件，释放 SDA 和 SCL 线，清零 I2CCON 寄存器中相应的控制位。当用户执行总线冲突中断服务程序时，如果 I²C 总线空闲，用户可以通过产生一个启动条件继续通信。

主器件将继续监视 SDA 和 SCL 引脚，如果出现停止条件，将置 1 MI2CIF 位。

写 I2CTRN 将从第一个数据位开始数据的发送，而与发生总线冲突时发送器的状态无关。

在多主器件环境中，在检测到启动和停止条件时产生中断，能够确定总线何时空闲。当 I2CSTAT 寄存器中的 P 位置 1 时，器件可以控制 I²C 总线，否则，总线空闲，且 S 和 P 位清零。

14.13 CPU 休眠和空闲模式期间 I²C 模块的工作

14.13.1 CPU 休眠模式期间 I²C 的工作

当器件进入休眠模式时，模块的全部时钟源都将关闭，并保持在逻辑 0。如果休眠发生在发送过程中，且时钟停止时状态机部分进入发送，发送将中止。类似地，如果休眠出现在接收过程中，接收也将中止。

14.13.2 CPU 空闲模式期间 I²C 的工作

对于 I²C，I2CSIDL 位选择 CPU 空闲时模块是停止还是继续工作。如果 I2CSIDL = 0，模块将继续工作；如果 I2CSIDL = 1，模块将停止工作。

表 14-1: I²C™ 寄存器映射

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态	
I2CRCV	0200	—	—	—	—	—	—	—	—	接收寄存器									0000 0000 0000 0000
I2CTRN	0202	—	—	—	—	—	—	—	—	发送寄存器									0000 0000 1111 1111
I2CBRG	0204	—	—	—	—	—	—	—	波特率发生器									0000 0000 0000 0000	
I2CCON	0206	I2CEN	-	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0001 0000 0000 0000	
I2CSTAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF	0000 0000 0000 0000	
I2CADD	020A	—	—	—	—	—	—	地址寄存器										0000 0000 0000 0000	

注： 有关寄存器位字段的描述，请参见《dsPIC30F 系列参考手册》（DS70046E_CN）。

注:

15.0 通用异步收发器 (UART) 模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册 I》(DS70046E_CN)。

通用异步收发器 (Universal Asynchronous Receiver Transmitter, UART) 模块是 dsPIC30F1010/202X 系列器件提供的串行 I/O 模块之一。UART 是可以和外设 (例如个人电脑、LIN、RS-232 和 RS-485 接口) 通信的全双工异步系统。此模块还包含 IrDA 编码器和解码器功能。

UART 模块的主要特性是：

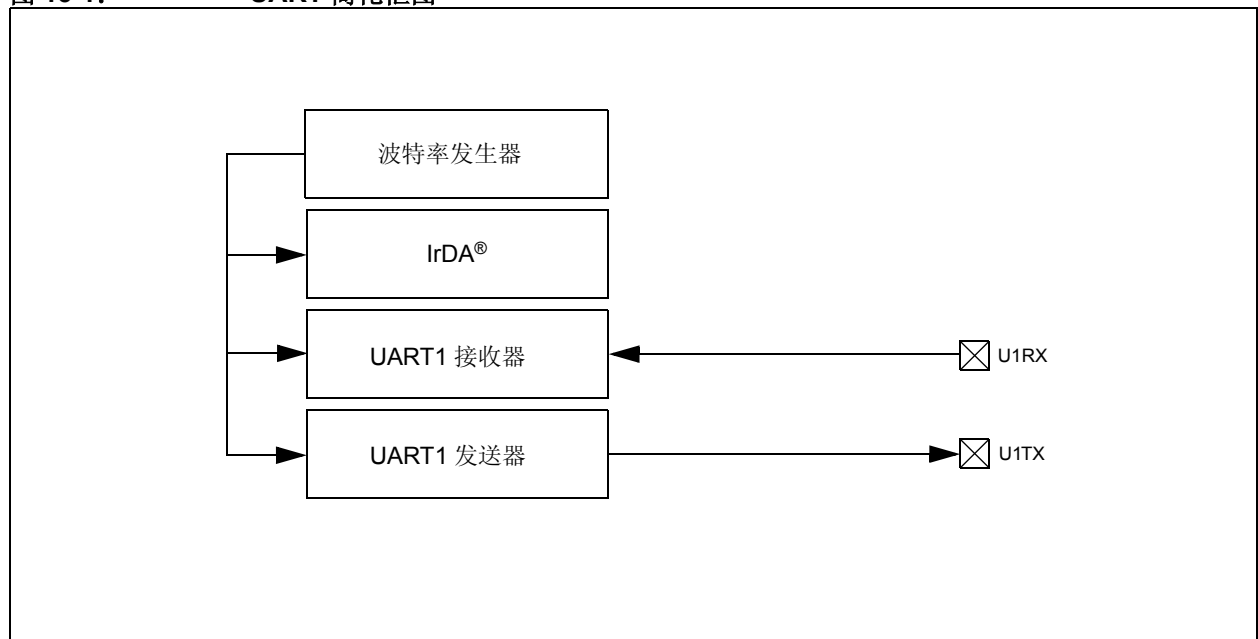
- 通过 U1TX 和 U1RX 引脚进行全双工 8 位或 9 位数据传输
- 偶校验、奇校验或无奇偶校验选项 (对于 8 位数据)
- 一个或两个停止位
- 完全集成的具有 16 位预分频器的波特率发生器

- 当器件工作在 16 MIPS 时，波特率范围从 15 bps 到 1 Mbps
- 4 字深度先进先出 (First-In-First-Out, FIFO) 发送数据缓冲器
- 4 字深度 FIFO 接收数据缓冲器
- 奇偶、帧和缓冲器溢出错误检测
- 支持带地址检测的 9 位模式 (第 9 位 = 1)
- 发送和接收中断
- 用于诊断支持的环回模式
- 支持同步和间隔字符
- 支持自动波特率检测
- IrDA 编码器和解码器逻辑
- 用于 IrDA 支持的 16 倍频波特率时钟输出

图 15-1 给出了 UART 的简化框图。UART 模块由以下至关重要的硬件组成：

- 波特率发生器
- 异步发送器
- 异步接收器

图 15-1: UART 简化框图



15.1 UART 波特率发生器 (BRG)

UART 模块包含一个专用的 16 位波特率发生器。U1BRG 寄存器控制一个自由运行的 16 位定时器的周期。公式 15-1 给出了 BRGH = 0 时计算波特率的公式。

公式 15-1: BRGH = 0 时的 UART 波特率 (1,2,3)

$$\text{波特率} = \frac{F_{CY}}{16 \cdot (U1BRG + 1)}$$

$$U1BRG = \frac{F_{CY}}{16 \cdot \text{波特率}} - 1$$

- 注 1:** F_{CY}表示指令周期时钟频率 (F_{OSC}/2)。
- 2:** 若外部振荡器频率为 15 MHz, 且 PLL 禁止, 则 F_{CY} 为 7.5 MHz。
- 3:** 若外部振荡器频率为 15 MHz, 且 PLL 使能, 则 F_{CY} 为 30 MHz。

例 15-1 给出了在下列条件下的波特率误差计算:

- F_{CY} = 7.5 MHz
- 目标波特率 = 9600

例 15-1: 波特率误差计算 (BRGH = 0) (1)

$$\text{目标波特率} = F_{CY} / (16 (U1BRG + 1))$$

U1BRG 值的计算方法:

$$U1BRG = ((F_{CY} / \text{目标波特率}) / 16) - 1$$

$$U1BRG = ((7500000 / 9600) / 16) - 1$$

$$U1BRG = 48$$

$$\text{计算波特率} = 7500000 / (16 (48 + 1))$$

$$= 9566$$

$$\text{误差} = \frac{(\text{计算波特率} - \text{目标波特率})}{\text{目标波特率}}$$

$$= (9566 - 9600) / 9600$$

$$= -0.35\%$$

注 1: 由于 T_{CY} = 2/F_{OSC}, 禁止 PLL。

最大可能波特率 (BRGH = 0) 是 F_{CY}/16 (当 U1BRG = 0 时), 最小可能波特率是 F_{CY}/(16 * 65536)。

公式 15-2 给出了 BRGH = 1 时计算波特率的公式。

公式 15-2: BRGH = 1 时的 UART 波特率 (1,2,3)

$$\text{波特率} = \frac{F_{CY}}{4 \cdot (U1BRG + 1)}$$

$$U1BRG = \frac{F_{CY}}{4 \cdot \text{波特率}} - 1$$

- 注 1:** F_{CY} 表示指令周期时钟频率。
- 2:** 若外部振荡器频率为 15 MHz, 且 PLL 禁止, 则 F_{CY} 为 7.5 MHz。
- 3:** 若外部振荡器频率为 15 MHz, 且 PLL 使能, 则 F_{CY} 为 30 MHz。

最大可能波特率 (BRGH = 1) 是 F_{CY}/4 (当 U1BRG = 0 时), 可能达到的最小波特率是 F_{CY}/(4 * 65536)。

向 U1BRG 寄存器中写入新值会导致 BRG 定时器复位 (清零)。这保证了 BRG 在产生新的波特率之前不需要等待定时器溢出。

15.2 在 8 位数据模式下发送

1. 设置 UART:
 - a) 将适当的值写入数据位、奇偶校验位和停止位。
 - b) 将适当的波特率值写入 U1BRG 寄存器。
 - c) 设置发送和接收中断允许位和优先级位。
2. 使能 UART。
3. 将 UTXEN 位置 1（产生发送中断）。
4. 将数据字节写入 TXxREG 字的低字节。它将被立即送往发送移位寄存器（TSR），且在波特时钟的下一个上升沿开始移出串行比特流。
5. 或者，当 UTXEN = 0 时数据字节也可以被发送，随后，用户可以将 UTXEN 置 1。由于波特率时钟将从清零状态启动，这将立即开始发送串行比特流。
6. 根据中断控制位 UTXISELx 的设置产生发送中断。

15.3 在 9 位数据模式下发送

1. 设置 UART（如第 15.2 节“在 8 位数据模式下发送”中所述）。
2. 使能 UART。
3. 将 UTXEN 位置 1（产生发送中断）。
4. 仅向 TXxREG 写入一个 16 位的值。
5. 向 TxTXREG 写入一个字可触发 9 位数据向 TSR 的传输。串行比特流将会在波特率时钟的第一个上升沿开始移出。
6. 根据中断控制位 UTXISELx 的设置产生发送中断。

15.4 间隔和同步发送过程

下面的过程将发送一个由间隔字符组成的报文帧头，其后紧跟一个自动波特率同步字节。

1. 将 UART 配置为所需的模式。
2. 将 UTXEN 和 UTXBRK 置 1——设置间隔字符。
3. 将一个虚拟字符装入 TXxREG 寄存器以启动发送（值被忽略）。
4. 向 TXxREG 写入“55h”——将同步字符装载到发送 FIFO 中。
5. 当间隔字符发送完成后，由硬件将 UTXBRK 位复位。然后开始发送同步字符。

15.5 在 8 位或 9 位数据模式下接收

1. 设置 UART（如第 15.2 节“在 8 位数据模式下发送”中所述）。
2. 使能 UART。
3. 当接收到一个或多个数据字符时，将会根据中断控制位 URXISELx 的设置产生接收中断。
4. 读 OERR 位，判断是否产生溢出错误。OERR 位必须由软件复位。
5. 读 RXxREG。

读取 RXxREG 字符的行为会将下一个字符传送到接收 FIFO 的顶部，其中包含一组新的 PERR 和 FERR 值。

15.6 内置 IrDA 编码器和解码器

UART 在其内部完全实现了 IrDA 编码器和解码器功能。内置 IrDA 编/解码器功能由 IREN 位（U1MODE<12>）使能。使能时（IREN = 1），接收引脚（U1RX）作为红外接收器的输入端，发送引脚（U1TX）作为红外发送器的输出端。

15.7 备用 UART I/O 引脚

备用 I/O 引脚 U1ATX 和 U1ARX 可用于通信。当 UART 的主引脚与其他外设共用时，备用 UART 引脚就有用了。通过置 1 UxMODE 寄存器中的 ALTIO 位可使能备用 I/O 引脚。若 ALTIO = 1，UART 模块使用 U1ATX 和 U1ARX 引脚，而不使用 U1TX 和 U1RX 引脚。若 ALTIO = 0，UART 模块将使用 U1TX 和 U1RX 引脚。

dsPIC30F1010/202X

寄存器 15-1: U1MODE: UART1 模式寄存器

R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	U-0	U-0
UARTEN	—	USIDL	IREN	—	ALTIO	—	—
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	RXINV	BRGH	PDSEL1	PDSEL0	STSEL
bit 7							bit 0

图注: U = 未实现位, 读为 0

R = 可读位

W = 可写位

HC = 由硬件清零

HS = 硬件选择

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 **UARTEN:** UART1 使能位
1 = 使能 UART1; UART1 根据 UEN<1:0> 的定义控制所有 UART1 引脚
0 = 禁止 UART1; 由端口锁存器控制所有 UART1 引脚; UART1 的功耗最小
- bit 14 **未实现:** 读为 0
- bit 13 **USIDL:** 空闲模式下的停止位
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 = 使能 IrDA 编码器和解码器
0 = 禁止 IrDA 编码器和解码器
注: 此功能只在 16x BRG 模式 (BRGH = 0) 下有效。
- bit 11 **未实现:** 读为 0
- bit 10 **ALTIO:** UART 备用 I/O 选择位
1 = UART 使用 U1ATX 和 U1ARX I/O 引脚进行通信
0 = UART 使用 U1TX 和 U1RX I/O 引脚进行通信
- bit 9-8 **未实现:** 读为 0
- bit 7 **WAKE:** 在休眠模式下检测到启动位唤醒使能位
1 = UART1 将继续采样 U1RX 引脚; 在出现下降沿时产生中断, 在出现上升沿时由硬件将该位清零
0 = 禁止唤醒
- bit 6 **LPBACK:** UART1 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——需要接收到同步字段 (55h); 完成时由硬件清零
0 = 禁止或已完成波特率测量
- bit 4 **RXINV:** 接收极性翻转位
1 = U1RX 空闲状态为 0
0 = U1RX 空闲状态为 1
- bit 3 **BRGH:** 高波特率使能位
1 = BRG 在每个位周期内产生 4 个时钟信号 (4 倍频波特率时钟, 高速模式)
0 = BRG 在每个位周期内产生 16 个时钟信号 (16 倍频波特率时钟, 标准模式)

寄存器 15-1: U1MODE: UART1 模式寄存器 (续)

bit 2-1 PDSEL1:PDSEL0: 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

dsPIC30F1010/202X

寄存器 15-2: U1STA: UART1 状态和控制寄存器

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
UTXISEL1	UTXINV ⁽¹⁾	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT
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
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

图注: U = 未实现位, 读为 0

R = 可读位

W = 可写位

HS = 由硬件置 1

HC = 由硬件清零

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15, 13 **UTXISEL1:UTXISEL0:** 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器为空时, 产生中断
 - 01 = 当最后一个字符被移出发送移位寄存器, 所有发送操作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 (暗指发送缓冲器中至少还有一个字符) 时产生中断
- bit 14 **UTXINV:** IrDA 编码器发送极性翻转位 ⁽¹⁾
- 1 = IrDA 编码的 U1TX 空闲状态为 1
 - 0 = IrDA 编码的 U1TX 空闲状态为 0
- 注 1: 仅当使能了 IrDA 编码器 (IREN = 1) 时, 该位的值才影响模块的发送属性。
- 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, UART1 控制 U1TX 引脚
 - 0 = 禁止发送, 中止所有等待的发送, 缓冲器复位。由端口控制 U1TX 引脚。
- bit 9 **UTXBF:** 发送缓冲器满状态位 (只读)
- 1 = 发送缓冲器满
 - 0 = 发送缓冲器未满, 至少还可写入一个或多个字符
- bit 8 **TRMT:** 发送移位寄存器空位 (只读)
- 1 = 发送移位寄存器为空, 同时发送缓冲器为空 (上一次发送已完成)。
 - 0 = 发送移位寄存器非空, 发送在进行中或在发送缓冲器中排队
- bit 7-6 **URXISEL1:URXISEL0:** 接收中断模式选择位
- 11 = 当 RSR 传输使接收缓冲器为满时 (即, 有 4 个数据字符), 中断标志位置 1
 - 10 = 当 RSR 传输使接收缓冲器为 3/4 满时 (即, 有 3 个数据字符), 中断标志位置 1
 - 0x = 当接收到一个字符且 RSR 的内容被传输给接收缓冲器时, 中断标志位置 1。接收缓冲器有一个或多个字符。
- bit 5 **ADDEN:** 地址字符检测位 (接收数据的第 8 位 = 1)
- 1 = 使能地址检测模式。若未选择 9 位模式, 此控制位将无效。
 - 0 = 禁止地址检测模式

寄存器 15-2: U1STA: UART1 状态和控制寄存器 (续)

bit 4	RIDLE: 接收器空闲位 (只读) 1 = 接收器空闲 0 = 接收器工作
bit 3	PERR: 奇偶校验错误状态位 (只读) 1 = 检测到当前字符的奇偶校验错误 (在接收 FIFO 顶部的字符) 0 = 未检测到奇偶校验错误
bit 2	FERR: 帧错误状态位 (只读) 1 = 检测到当前字符的帧错误 (在接收 FIFO 顶部的字符) 0 = 未检测到帧错误
bit 1	OERR: 接收缓冲器溢出错误状态位 (只读 / 清零) 1 = 接收缓冲器已溢出 0 = 接收缓冲器没有溢出。清除原来置1的OERR位(1 → 0的转换)将使接收缓冲器复位并使RSR为空。
bit 0	URXDA: 接收缓冲器中是否有数据位 (只读) 1 = 接收缓冲器中有数据, 有至少一个或多个字符可被读取 0 = 接收缓冲器为空

表 15-1: 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	复位状态
U1MODE	0220	UARTEN	—	USIDL	IREN	—	ALTIO	—	—	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL<1:0>	STSEL	0000	UART 发送寄存器
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL<1:0>	—	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	
U1TXREG	0224	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
U1RXREG	0226	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
U1BRG	0228	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
波特率发生器预分频器																		
UART 接收寄存器																		

波特率发生器预分频器

UART 发送寄存器

UART 接收寄存器

图注: x = 复位时未知的值, — = 未实现, 读为 0。所示复位值为十六进制。

16.0 10 位 2 MSPS 模数转换器 (ADC) 模块

dsPIC30F1010/202X 器件具有高速逐次逼近模数转换模块，可支持如 AC/DC 和 DC/DC 电源转换器等应用。

16.1 特性

- 10 位分辨率
- 单极性输入
- 最多 12 个输入通道
- ± 1 LSB 精度
- 单电源工作
- 5V 电压下的转换速率为 2000 ksps
- 3V 电压下的转换速率为 1000 ksps
- 低功耗 CMOS 技术

16.2 描述

此 ADC 模块设计用于要求在请求转换和输出结果数据间具有较短延时的应用。典型应用包括：

- AC/DC 电源
- DC/DC 转换器
- 功率因数校正

此 ADC 与需要高频控制环的电源控制应用的电源 PWM 模块协同工作。此模块可在 1 微秒内采样并转换两个模拟输入。转换延时为 1 微秒，减轻了测量和控制系统响应间“相位滞后”的程度。

每次最多可采样 4 个输入，每次最多可有 12 个输入请求转换。如果多个输入请求转换，ADC 将从最低顺序的输入开始依次对其进行转换。

这种 ADC 设计为每对模拟输入 (AN1 和 AN0) 和 (AN3 和 AN2) 等提供了最多可从 16 个不同触发源中指定其自身触发源的能力。此功能允许 ADC 采样和转换与运行在独立时基上的 PWM 发生器相关的模拟输入。

该模块在休眠模式下不工作。用户应用通常要求模拟数据采样和到应用电路的 PWM 输出间保持同步。此 ADC 模块的高速工作使得“按需供应数据 (data on demand)”成为可能。

另外，还向外设接口添加了几个硬件功能，用于提高基于 DSP 的典型应用的实时性能。

1. 结果对齐选项
2. 自动采样
3. 外部转换启动控制

图 16-1 给出了 ADC 模块的框图。

16.3 模块功能

10 位 2 Msps ADC 设计用于和电源 PWM 模块协同工作以支持电源转换应用。10 位 2 Msps ADC 一次最多采样 N ($N \leq 12$) 个输入，然后每次转换两个采样到的输入。采样保持电路的品质取决于器件要求。10 位 2 Msps ADC 在 1 微秒内可产生两个 10 位转换结果。

ADC 模块最多支持 12 个模拟输入。采样到的输入通过多路开关与转换器连接。

模拟参考电压定义为器件供电电压 ($AVDD / AVSS$)。

ADC 模块使用以下控制和状态寄存器：

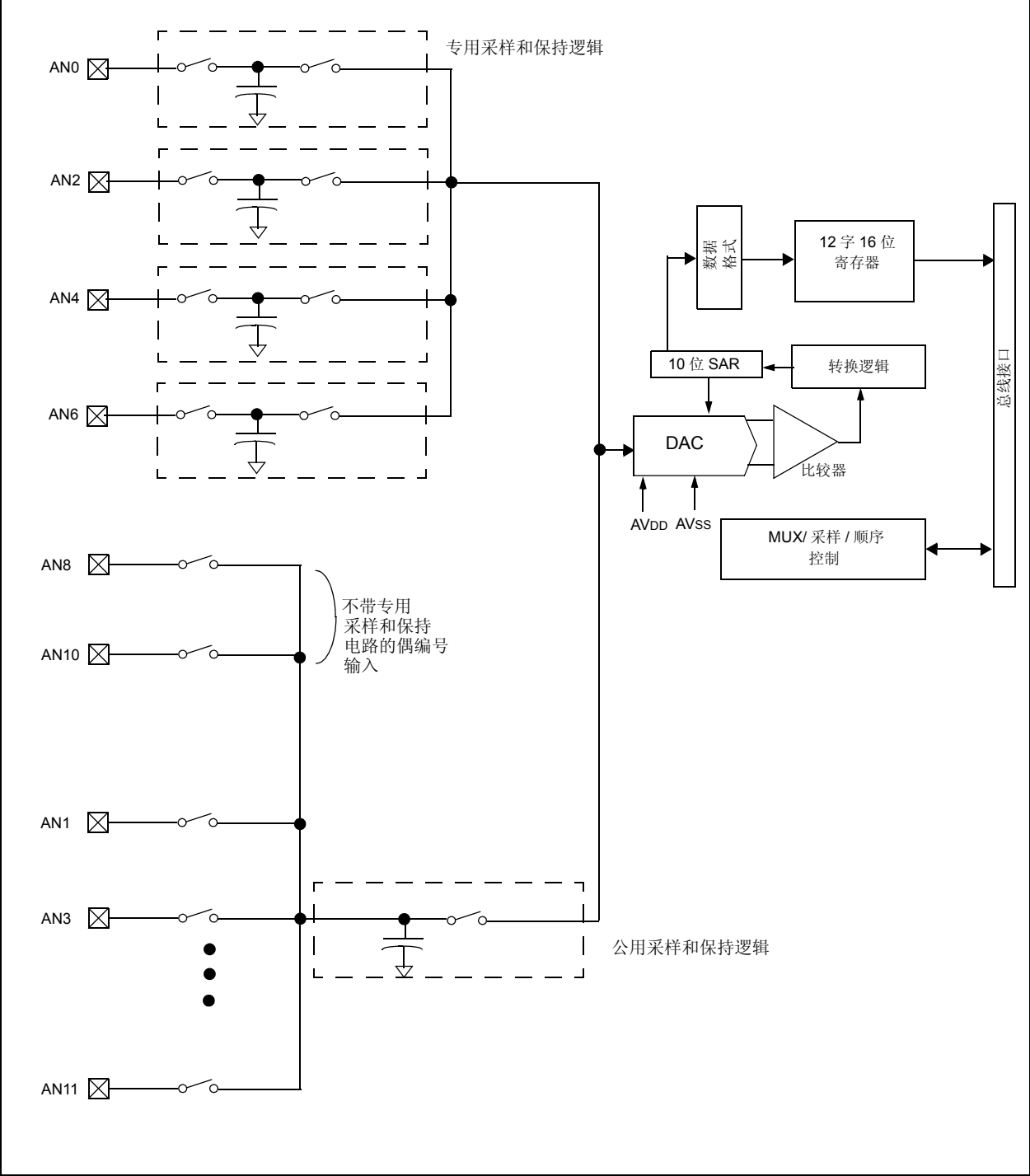
- A/D 控制寄存器 (ADCON)
- A/D 状态寄存器 (ADSTA)
- A/D 基本寄存器 (ADBASE)
- A/D 端口配置寄存器 (ADPCFG)
- A/D 转换器对控制寄存器 0 (ADCPC0)
- A/D 转换器对控制寄存器 1 (ADCPC1)
- A/D 转换器对控制寄存器 2 (ADCPC2)

ADCON 寄存器控制 ADC 模块的工作。ADSTAT 寄存器显示转换过程的状态。ADPCFG 寄存器用于将端口引脚配置为模拟输入或数字 I/O。CPC 寄存器控制 ADC 转换的触发。(有关详细的位配置，请参见寄存器 16-1 至寄存器 16-7)。

注：ADC 模块的独特特点是能以异步方式采样输入。各采样保持电路的触发可以是相互独立的。

注：要使 ADC 模块正常工作，必须使能 PLL。可使用 FOSCSEL 配置寄存器中的 FNOSC<1:0> 位使能 PLL。

图 16-1: ADC 框图



寄存器 16-1: A/D 控制寄存器 (ADCON)

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	U-0	R/W-0
ADON	—	ADSIDL	—	—	GSWTRG	—	FORM
bit 15							bit 8
R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-1	R/W-1
EIE	ORDER	SEQSAMP	—	—	ADCS<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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **ADON:** A/D 工作模式选择位

1 = A/D 转换器模块工作

0 = A/D 转换器模块关闭

bit 14 **未实现:** 读为 0bit 13 **ADSIDL:** 空闲模式停止位

1 = 器件进入空闲模式时模块停止工作

0 = 器件进入空闲模式时模块继续工作

bit 12-11 **未实现:** 读为 0bit 10 **GSWTRG:** 全局软件触发位

当用户置 1 此位时, 将触发由 ADCPCx 寄存器中的 TRGSRC<4:0> 位所选择的转换。在启动另一个全局触发前用户必须清零此位 (即, 此位不能自动清零)。

bit 9 **未实现:** 读为 0bit 8 **FORM:** 数据输出格式位

1 = 小数 (DOUT = dddd dddd dd00 0000)

0 = 整数 (DOUT = 0000 00dd dddd dddd)

bit 7 **EIE:** 早期中断允许位

1 = 第一次转换完成时产生中断

0 = 第二次转换完成时产生中断

注: 仅当禁止 ADC (ADON = 0) 时可更改此控制位。bit 6 **ORDER:** 转换顺序位

1 = 先转换奇数编号的模拟输入, 后转换偶数编号的输入

0 = 先转换偶数编号的模拟输入, 后转换奇数编号的输入

注: 仅当禁止 ADC (ADON = 0) 时可更改此控制位。bit 5 **SEQSAMP:** 顺序采样使能位

1 = 如果 ORDER = 0, 则在第二次转换开始时采样共用的采样保持 (S&H) 逻辑。如果 ORDER = 1, 则在第一次转换开始时采样共用的采样保持逻辑。

0 = 如果共用的采样保持逻辑当前不在处理转换, 则同时采样共用的采样保持逻辑与专用的采样保持逻辑。如果在采样专用的采样保持逻辑时共用的采样保持逻辑忙碌, 则将在新的转换周期开始时采样共用的采样保持逻辑。

bit 4-3 **未实现:** 读为 0

寄存器 16-1: A/D 控制寄存器 (ADCON) (续)

bit 2-0

ADCS<2:0>: A/D 转换时钟分频比选择位

如果使能了 PLL (假设 15 MHz 外部时钟作为时钟源):

111 = $F_{ADC}/18 = 13.3 \text{ MHz}$ (在 30 MIPS 条件下)

110 = $F_{ADC}/16 = 15.0 \text{ MHz}$ (在 30 MIPS 条件下)

101 = $F_{ADC}/14 = 17.1 \text{ MHz}$ (在 30 MIPS 条件下)

100 = $F_{ADC}/12 = 20.0 \text{ MHz}$ (在 30 MIPS 条件下)

011 = $F_{ADC}/10 = 24.0 \text{ MHz}$ (在 30 MIPS 条件下)

110 = $F_{ADC}/8 = 30.0 \text{ MHz}$ (在 30 MIPS 条件下)

001 = $F_{ADC}/6$ = 保留, 默认为 30 MHz (在 30 MIPS 条件下)

000 = $F_{ADC}/4$ = 保留, 默认为 30 MHz (在 30 MIPS 条件下)

如果禁止了 PLL (假设 15 MHz 外部时钟作为时钟源):

111 = $F_{ADC}/18 = 0.83 \text{ MHz}$ (在 7.5 MIPS 条件下)

110 = $F_{ADC}/16 = 0.93 \text{ MHz}$ (在 7.5 MIPS 条件下)

101 = $F_{ADC}/14 = 1.07 \text{ MHz}$ (在 7.5 MIPS 条件下)

100 = $F_{ADC}/12 = 1.25 \text{ MHz}$ (在 7.5 MIPS 条件下)

011 = $F_{ADC}/10 = 1.5 \text{ MHz}$ (在 7.5 MIPS 条件下)

110 = $F_{ADC}/8 = 1.87 \text{ MHz}$ (在 7.5 MIPS 条件下)

001 = $F_{ADC}/6 = 2.5 \text{ MHz}$ (在 7.5 MIPS 条件下)

000 = $F_{ADC}/4 = 3.75 \text{ MHz}$ (在 7.5 MIPS 条件下)

注: 有关 ADC 的时钟源, 请参见图 18-2。

寄存器 16-2: A/D 状态寄存器 (ADSTA)

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/C-0 H-S	R/C-0 H-S	R/C-0 H-S	R/C-0 H-S	R/C-0 H-S	R/C-0 H-S
—	—	P5RDY	P4RDY	P3RDY	P2RDY	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

C = 由软件清零

H-S = 由硬件清零

bit 15-6 未实现: 读为 0

bit 5 **P5RDY**: 对 5 转换数据就绪位
当缓冲器中的数据就绪时, 置 1, 当向此位写入 0 时清零。

bit 4 **P4RDY**: 对 4 转换数据就绪位
当缓冲器中的数据就绪时, 置 1, 当向此位写入 0 时清零。

bit 3 **P3RDY**: 对 3 转换数据就绪位
当缓冲器中的数据就绪时, 置 1, 当向此位写入 0 时清零。

bit 2 **P2RDY**: 对 2 转换数据就绪位
当缓冲器中的数据就绪时, 置 1, 当向此位写入 0 时清零。

bit 1 **P1RDY**: 对 1 转换数据就绪位
当缓冲器中的数据就绪时, 置 1, 当向此位写入 0 时清零。

bit 0 **P0RDY**: 对 0 转换数据就绪位
当缓冲器中的数据就绪时, 置 1, 当向此位写入 0 时清零。

dsPIC30F1010/202X

寄存器 16-3: A/D 基本寄存器 (ADBASE)

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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-1

ADC 基本寄存器: 此寄存器包括用户 ADC 中断服务程序跳转表的基地址。读此寄存器时, 读到的是 ADBASE 寄存器的内容和 PxRDY 状态位的编码值的和。

编码器逻辑提供最高优先级 PxRDY 位的位数 (其中, P0RDY 为最高优先级, P5RDY 为最低优先级)。

注: 编码结果左移两位, 因此其 bit 1-0 总是为 0。

bit 0

未实现: 读为 0

注: 作为 ADBASE 寄存器的备用寄存器, ADCP0-5 ADC 对转换完成中断 (中断 37-42) 可用于调用各 ADC 输入对的转换完成子程序。请参见第 16.9 节 “单对中断”。

寄存器 16-4: A/D 端口配置寄存器 (ADPCFG)

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PCFG11	PCFG10	PCFG2	PCFG8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

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 = 端口引脚处于数字模式, 使能端口读输入, A/D 输入多路开关与 AVss 连接。

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

寄存器 16-5: A/D 转换器对控制寄存器 0 (ADCPC0)

R/W-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 = 上电复位时的值

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 请求的转换完成时, 允许产生 IRQ
0 = 不产生 IRQ
- bit 14 **PEND1:** 等待转换状态 1 位
1 = AN3 和 AN2 通道正在等待转换。当所选的触发发生时置 1
0 = 转换完成
- bit 13 **SWTRG1:** 软件触发 1 位
1 = 开始转换 AN3 和 AN2 (如果已在 TRGSRC 位选择)。如果正在进行其他转换, 则当转换资源可用时开始执行转换。当 PEND 位置 1 时, 复位此位。
- bit 12-8 **TRGSRC1<4:0>:** 触发 1 源选择位
选择转换模拟通道 AN3 和 AN2 的触发源。
00000 = 未使能转换
00001 = 选择单个软件触发
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 触发
00101 = 选择 PWM 发生器 2 触发
00110 = 选择 PWM 发生器 3 触发
00111 = 选择 PWM 发生器 4 触发
01100 = Timer1 周期匹配
01101 = Timer2 周期匹配
00110 = PWM 发生器 1 电流限制 ADC 触发
01111 = PWM 发生器 2 电流限制 ADC 触发
10000 = PWM 发生器 3 电流限制 ADC 触发
10001 = PWM 发生器 4 电流限制 ADC 触发
10110 = PWM 发生器 1 故障 ADC 触发
10111 = PWM 发生器 2 故障 ADC 触发
11000 = PWM 发生器 3 故障 ADC 触发
11001 = PWM 发生器 4 故障 ADC 触发
- bit 7 **IRQEN0:** 中断请求允许 0 位
1 = 当通道 AN1 和 AN0 请求的转换完成时, 允许产生 IRQ
0 = 不产生 IRQ
- bit 6 **PEND0:** 等待转换状态 0 位
1 = AN1 和 AN0 通道正在等待转换。当所选的触发发生时置 1。
0 = 转换完成
- bit 5 **SWTRG0:** 软件触发 0 位
1 = 开始转换 AN1 和 AN0 (如果已在 TRGSRC 位选择)。如果正在进行其他转换, 则当转换资源可用时开始执行转换。当 PEND 位置 1 时, 复位此位。

寄存器 16-5: A/D 转换器对控制寄存器 0 (ADCP0) (续)

bit 4-0

TRGSRC0<4:0>: 触发 0 源选择位

选择转换模拟通道 AN1 和 AN0 的触发源。

00000 = 未使能转换

00001 = 选择单个软件触发

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 触发

00101 = 选择 PWM 发生器 2 触发

00110 = 选择 PWM 发生器 3 触发

00111 = 选择 PWM 发生器 4 触发

01100 = Timer1 周期匹配

01101 = Timer2 周期匹配

00110 = PWM 发生器 1 电流限制 ADC 触发

01111 = PWM 发生器 2 电流限制 ADC 触发

10000 = PWM 发生器 3 电流限制 ADC 触发

10001 = PWM 发生器 4 电流限制 ADC 触发

10110 = PWM 发生器 1 故障 ADC 触发

10111 = PWM 发生器 2 故障 ADC 触发

11000 = PWM 发生器 3 故障 ADC 触发

11001 = PWM 发生器 4 故障 ADC 触发

寄存器 16-6: A/D 转换器对控制寄存器 1 (ADCPC1)

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	PEND3	SWTRG3	TRGSRC3<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
IRQEN2	PEND2	SWTRG2	TRGSRC2<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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15

IRQEN3: 中断请求允许 3 位

1 = 当通道 AN7 和 AN6 请求的转换完成时, 允许产生 IRQ。

0 = 不产生 IRQ

bit 14

PEND3: 等待转换状态 3 位

1 = AN7 和 AN6 通道正在等待转换。当所选的触发发生时置 1。

0 = 转换完成

bit 13

SWTRG3: 软件触发 3 位

1 = 开始转换 AN7 和 AN6 (如果已在 TRGSRC 位选择)。如果正在进行其他转换, 则当转换资源可用时开始执行转换。当 PEND 位置 1 时, 复位此位。

bit 12-8

TRGSRC3<4:0>: 触发 3 源选择位

选择转换模拟通道 AN7 和 AN6 的触发源。

00000 = 未使能转换

00001 = 选择单个软件触发

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 触发

00101 = 选择 PWM 发生器 2 触发

00110 = 选择 PWM 发生器 3 触发

00111 = 选择 PWM 发生器 4 触发

01100 = Timer1 周期匹配

01101 = Timer2 周期匹配

00110 = PWM 发生器 1 电流限制 ADC 触发

01111 = PWM 发生器 2 电流限制 ADC 触发

10000 = PWM 发生器 3 电流限制 ADC 触发

10001 = PWM 发生器 4 电流限制 ADC 触发

10110 = PWM 发生器 1 故障 ADC 触发

10111 = PWM 发生器 2 故障 ADC 触发

11000 = PWM 发生器 3 故障 ADC 触发

11001 = PWM 发生器 4 故障 ADC 触发

bit 7

IRQEN2: 中断请求允许 2 位

1 = 当通道 AN5 和 AN4 请求的转换完成时, 允许产生 IRQ。

0 = 不产生 IRQ

bit 6

PEND2: 等待转换状态 2 位

1 = AN5 和 AN4 通道正在等待转换。当所选的触发发生时置 1

0 = 转换完成

bit 5

SWTRG2: 软件触发 2 位

1 = 开始转换 AN5 和 AN4 (如果已在 TRGSRC 位选择)。如果正在进行其他转换, 则当转换资源可用时开始执行转换。当 PEND 位置 1 时, 复位此位

寄存器 16-6: A/D 转换器对控制寄存器 1 (ADCP1) (续)

bit 4-0

TRGSRC2<4:0>: 触发 2 源选择位

选择转换模拟通道 AN5 和 AN4 的触发源。

00000 = 未使能转换

00001 = 选择单一软件触发

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 触发

00101 = 选择 PWM 发生器 2 触发

00110 = 选择 PWM 发生器 3 触发

00111 = 选择 PWM 发生器 4 触发

01100 = Timer1 周期匹配

01101 = Timer2 周期匹配

00110 = PWM 发生器 1 电流限制 ADC 触发

01111 = PWM 发生器 2 电流限制 ADC 触发

10000 = PWM 发生器 3 电流限制 ADC 触发

10001 = PWM 发生器 4 电流限制 ADC 触发

10110 = PWM 发生器 1 故障 ADC 触发

10111 = PWM 发生器 2 故障 ADC 触发

11000 = PWM 发生器 3 故障 ADC 触发

11001 = PWM 发生器 4 故障 ADC 触发

寄存器 16-7: A/D 转换器对控制寄存器 2 (ADCPC2)

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 = 上电复位时的值

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 请求的转换完成时, 允许产生 IRQ

0 = 不产生 IRQ

bit 14

PEND5: 等待转换状态 5 位

1 = AN11 和 AN10 通道正在等待转换。当所选的触发发生时置 1

0 = 转换完成

bit 13

SWTRG5: 软件触发 5 位

1 = 开始转换 AN11 和 AN10 (如果已在 TRGSRC 位选择)。如果正在进行其他转换, 则当转换资源可用时开始执行转换。当 PEND 位置 1 时, 复位此位。

bit 12-8

TRGSRC5<4:0>: 触发源选择 5 位

选择转换模拟通道 AN11 和 AN10 时的触发源。

00000 = 未使能转换

00001 = 选择单一软件触发

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 触发

00101 = 选择 PWM 发生器 2 触发

00110 = 选择 PWM 发生器 3 触发

00111 = 选择 PWM 发生器 4 触发

01100 = Timer 1 周期匹配

01101 = Timer 2 周期匹配

00110 = PWM 发生器 1 电流限制 ADC 触发

01111 = PWM 发生器 2 电流限制 ADC 触发

10000 = PWM 发生器 3 电流限制 ADC 触发

10001 = PWM 发生器 4 电流限制 ADC 触发

10110 = PWM 发生器 1 故障 ADC 触发

10111 = PWM 发生器 2 故障 ADC 触发

11000 = PWM 发生器 3 故障 ADC 触发

11001 = PWM 发生器 4 故障 ADC 触发

bit 7

IRQEN4: 中断请求允许 4 位

1 = 当通道 AN9 和 AN8 请求的转换完成时, 允许产生 IRQ

0 = 不产生 IRQ

bit 6

PEND4: 等待转换状态 4 位

1 = AN9 和 AN8 通道正在等待转换。当所选的触发发生时置 1。

0 = 转换完成

bit 5

SWTRG4: 软件触发 4 位

1 = 开始转换 AN9 和 AN8 (如果已在 TRGSRC 位选择)。如果正在进行其他转换, 则当转换资源可用时开始执行转换。当 PEND 位置 1 时, 复位此位。

寄存器 16-7: A/D 转换器对控制寄存器 2 (ADCP2) (续)

bit 4-0

TRGSRC4<4:0>: 触发源选择 4 位

选择转换模拟通道 AN9 和 AN8 的触发源。

00000 = 未使能转换

00001 = 选择单一软件触发

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 触发

00101 = 选择 PWM 发生器 2 触发

00110 = 选择 PWM 发生器 3 触发

00111 = 选择 PWM 发生器 4 触发

01100 = Timer 1 周期匹配

01101 = Timer 2 周期匹配

00110 = PWM 发生器 1 电流限制 ADC 触发

01111 = PWM 发生器 2 电流限制 ADC 触发

10000 = PWM 发生器 3 电流限制 ADC 触发

10001 = PWM 发生器 4 电流限制 ADC 触发

10110 = PWM 发生器 1 故障 ADC 触发

10111 = PWM 发生器 2 故障 ADC 触发

11000 = PWM 发生器 3 故障 ADC 触发

11001 = PWM 发生器 4 故障 ADC 触发

16.4 ADC 结果缓冲器

ADC 模块最多包含 12 个用于存储 A/D 结果的数据输出寄存器，表示为 $\text{ADCBUF}<11:0>$ 。寄存器为 10 位宽，但读出格式为 16 位的字。这些缓冲器是只读的。

各模拟输入具有一个与之对应的数据输出寄存器。

该模块不包括循环数据缓冲区或 FIFO。因为转换结果的产生顺序是任意的，无法确定特定单元中的数据，故上述方案行不通。

SAR 写入到缓冲器的过程与 ADC 时钟是同步的。只要已处理了数据就绪中断，则读缓冲器总是能读到有效的数据。

如果软件尚未读缓冲单元，但 SAR 需要改写该单元，则之前的数据就会丢失。

从结果缓冲器中读出的数据会被传递给数据格式化程序（data formatter）。10 位结果数据被格式化为 16 位的字。

16.5 应用信息

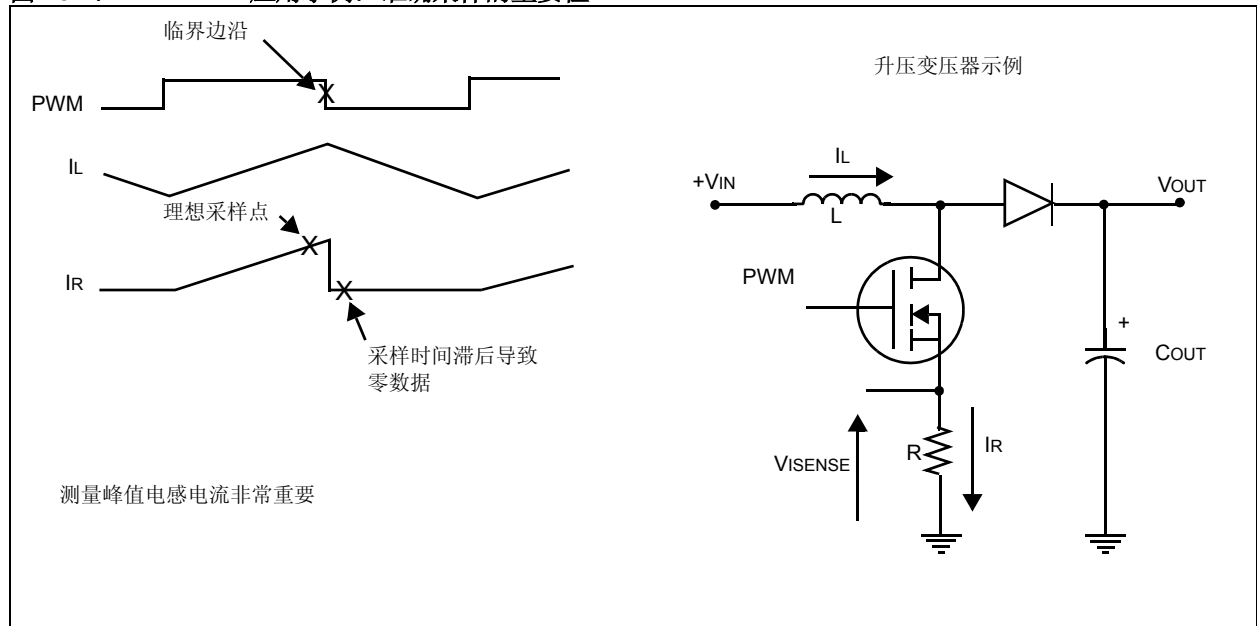
ADC 模块实现了一个基于“成对转换”的概念。在电源转换应用中，需要测量各 PWM 控制环的电压和电流。ADC 模块使每对转换的采样和转换过程可精确按 PWM 信号定时。

在用户应用电路中，PWM 信号可使能一个晶体管，因而可使电感充电到所需的电流值。PWM 信号持续时间越长，电感充电时间就越长，因此电感电流在 PWM 信号结尾处达到最大值。通常，用户希望在此时测量电流和电压值。

图 16-2 显示了一个典型的电源转换应用（升压变压器），其中对电感电流的检测是通过监视与功率晶体管串联的电阻两端的压降完成的，该功率晶体管用于为电感充电。此图的一个显著的特点是，如果比理想的采样时刻稍慢一点采样电阻电压，则读到的数据将为 0。在大多数应用中这是不容接受的。ADC 模块始终按规定的时间采样模拟电压，而不论 ADC 转换器是否在“忙”。

电源 PWM 模块支持 2 至 4 个独立 PWM 通道和 2 至 4 个触发信号（每个 PWM 发生器具有一个触发信号）。用户可在 PWM 周期的适当时刻配置这些通道以启动对所选输入对的 ADC 转换。电源 PWM 模块也提供一个额外的触发信号（特殊事件触发信号），可对其编程使之在主时基计数周期的一个指定时间发生。

图 16-2: 应用示例：准确采样的重要性



16.6 逆转转换顺序

ADCON 寄存器中的 ORDER 控制位，当置 1 时，可逆转一对输入转换过程的顺序。通常 (ORDER = 0)，先转换某对输入中偶数编号的输入，后转换奇数编号的输入。如果 ORDER = 1，则先转换奇数输入，随后转换偶数引脚。

当使用电压控制模式和使用早期中断功能 (EIE = 1) 时，此功能很有用。此功能允许用户最大程度地减少从实际采集到反馈 (ADC) 数据到更新控制输出 (PWM) 间的时间周期。这一从控制系统的输入到输出的时间决定了控制系统的整体稳定性。

16.7 同时和顺序采样一对输入

输入具有专用的采样和保持 (S&H) 电路，当特定的触发事件发生时对输入进行采样。共用采样和保持电路的输入按如下方式采样：

1. 如果 SEQSAMP 位 = 0，且共用采样保持电路不“忙”，则共用 S&H 与专用 S&H 同时采样其指定输入。此行为提供了“同时”采样和保持功能。
2. 如果 SEQSAMP 位 = 0，且共用 S&H 当前正在处理一个转换，则共用 S&H 将尽快采样（在输入对新转换过程开始时）。
3. 如果 SEQSAMP 位 = 1，则共用 S&H 将在该输入的转换过程开始时采样。例如：如果 ORDER 位 = 0，则共用 S&H 将在第二个输入的转换过程开始时采样。如果 ORDER = 1，则共用采样和保持电路 (S&H) 将在第一个输入转换开始时采样。

对一些希望最大程度地缩短采样事件和转换采样间的时间的应用而言，SEQSAMP 位很有用。

当 SEQSAMP = 0 时，如果有可用的资源，则逻辑电路试图同时对一对输入采样。通过控制启动转换的触发信号的时序，用户通常可确保 ADC 不忙于处理前一转换。

16.8 组中断的产生

ADC 模块提供了一个公用或“组”中断请求，它是模块内部所有已允许的中断源的逻辑或 (OR)。各 CPC 寄存器具有两个 IRQENx 位，每个模拟输入对各具有一个。如果 IRQEN 位置 1，则所请求的转换完成时，会发出中断请求到中断控制器。当产生中断时，ADSTAT 寄存器中的相关 PxRDY 位将置 1。PxRDY 位由用户清零。用户软件可检查 ADSTAT 寄存器的 PxRDY 位以确定是否已完成了更多的已请求中断。

对一个使用公用软件子程序处理多对模拟输入的 ADC 中断的应用而言，组中断很有用。此方法在概念上比较传统。

注： 用户必须在清零 PxRDY 位前先清零中断控制器中与 ADC 相关的 IFS 位。否则会导致中断丢失。其原因在于 ADC 可能具有另一个等待处理的中断。ADC 可能在用户清零 PxRDY 位后产生另一个中断请求，但如果随后清零了 IFS 位，则该中断请求将被清除。

16.9 单对中断

ADC 模块也为各对模拟输入提供单独的中断输出。在模块内始终允许这些中断。可通过 IEC 寄存器中的相关中断允许位分别允许和禁止这些中断。

使用组中断可能要求中断服务程序确定产生中断的中断源。对使用独立软件任务处理 ADC 数据的应用而言，公用中断向量可能会导致性能瓶颈。

与使用组中断处理多个中断源相比，使用单对中断可节省很多时钟周期。单对中断支持基于任务进行响应和组织的应用程序软件结构。

无论使用单对中断还是全局中断来响应来自 ADC 转换的中断请求，ADSTAT 寄存器的 PxRDY 位的工作方式都相同。

与组中断中的固定优先级结构相比，使用单对中断允许用户更改单个 ADC 通道（对）的中断优先级。

注意：使用单独中断不会影响 ADC 优先级结构中每对输入转换的顺序。

使用单独中断可避免在处理和清除当前中断时出现意外“丢失”一个等待处理的中断的问题。

16.10 早期中断的产生

ADCON 寄存器中的 EIE 控制位允许第一个转换完成后产生中断，而不是等到输入对的两个输入都完成转换时才产生中断。即使第二个输入仍在转换过程中，也可编写软件使用第一个数据值执行一些计算，并且此时第二个输入也完成了转换。

使用第二个数据前，可编写用户软件计时第二个输入的 500 ns 的转换周期，或者，用户可查询 ADCPCx 寄存器中的 PEND 位。

在一对中两个输入的转换都完成前，PEND 位保持置 1。第一个转换完成时，ADSTAT 寄存器中相关中断的 PxRDY 位置 1，且保持置 1 直到由用户清零。

16.11 冲突解决

如果同一时间有多个有效的转换对请求，则 ADC 控制逻辑将以从上自下的方式处理请求，从模拟对 0 (AN1/AN0) 开始，到模拟对 5 (AN11/AN10) 结束。这并不是一个“轮叫 (round-robin)”过程。

16.12 故意冲突

如果用户为多对“转换”指定相同的转换触发源，则 ADC 模块与其他 dsPIC30F ADC 模块功能相同；即，模块按顺序（以对为单位）处理所请求的转换直到序列结束为止。

注：一旦触发，ADC 模块将不重复循环。各转换序列需要一个或多个触发源。

16.13 ADC 时钟选择

ADCON 寄存器中的 ADCS<2:0> 位指定 ADC 时钟产生逻辑的时钟分频比。当 PLL 工作时，到 ADC 时钟分频器的输入为系统时钟（30 MIPS 下为 240MHz）。此高频时钟提供所需的时序分辨率以产生一个 24 MHz 的 ADC 时钟信号，从而在 1 微秒内完成两次 ADC 转换。

16.14 ADC 基本寄存器

用户应用可能需要 ADC 模块每秒产生 500,000 次中断。为加速对 ADSTAT 寄存器中 PxRDY 位的评估，ADC 模块具有读写寄存器：ADBASE。读 ADBASE 寄存器时，它提供 ADBASE 寄存器中的内容和 ADSTAT 寄存器中 PxRDY 位的编码之和。

ADBASE 寄存器的最低有效位被强制为零，以确保所有 (ADBASE + PxRDY) 结果在指令边界上。

PxRDY 位为二进制优先级编码；P0RDY 为最高优先级，P5RDY 为最低优先级。编码的优先级结果左移两位并与 ADBASE 寄存器的内容相加。这样，优先级编码产生了在两个指令字边界上的地址。

用户通常将“跳转”表的基地址装入 ADBASE 寄存器，该“跳转”表包含相应 ISR 的地址或到相应 ISR 地址的转移。设置经过编码的 PxRDY 值以保留跳转表中每项的两个指令字。希望用户软件使用一个指令字将标识符装入 W 寄存器，用另一条指令转移到相应的 ISR。

dsPIC30F1010/202X

例 16-1 所示为使用 ADBASE 寄存器实现 ADC 成对输入中断处理的代码序列。当读 ADBASE 寄存器时，它包含跳转表的基地址与已编码的 ADC 通道数左移两位后的值之和。

例如，如果使用值 0x0360 初始化 ADBASE，则第 1 对通道中断将导致读到 ADBASE 的值为 0x0364 (0x360 + 0b00000100)。第 3 对通道中断将导致读到 ADBASE 的值为 0x036C (0x360 + 0b00001100)。

例 16-1: ADC 基本寄存器代码

```
; Initialize and enable the ADC interrupt

MOV    #handle(JMP_TBL),W0    ; Load the base address of the ISR Jump
MOVW0, ADBASE                  ; table in ADBASE.

BSET    IPC2,#12                ; Set up the interrupt priority
BSET    IPC2,#13
BSET    IPC2,#14

BCLR    IFS0,#11                ; Clear any pending interrupts
BCLR    ADSTAT                  ; Clear the ADC pair interrupts as well

BSET    IEC0,#11                ; Enable the interrupt

; Code to Initialize the rest of the ADC registers

...
...
...

; ADC Interrupt Handler
__ADCInterrupt:

    PUSH.S                      ; Save W0-W3 and SR registers

    BCLR    IFS0,#11            ; Clear the interrupt
    MOV     ADBASE, W0          ; ADBASE contains the encoded jump address
    GOTO    W0                  ; within JMP_TBL

; Here's the Jump Table
; Note: It is important to clear the individual IRQ flags in the ADC AFTER the IRQ flags
in the interrupt controller. Failure to do so may cause interrupt requests to be lost

JMP_TBL:

    BCLR    ADSTAT,#0           ; Clear the IRQ flag in the ADC
    BRA     ADC_PAIR0_PROC      ; Actual Pair 0 Conversion Interrupt Handler

    BCLR    ADSTAT,#1           ; Clear the IRQ flag in the ADC
    BRA     ADC_PAIR1_PROC      ; Actual Pair 1 Conversion Interrupt Handler

    BCLR    ADSTAT,#2           ; Clear the IRQ flag in the ADC
    BRA     ADC_PAIR2_PROC      ; Actual Pair 2 Conversion Interrupt Handler

    BCLR    ADSTAT,#3           ; Clear the IRQ flag in the ADC
    BRA     ADC_PAIR3_PROC      ; Actual Pair 3 Conversion Interrupt Handler

    BCLR    ADSTAT,#4           ; Clear the IRQ flag in the ADC
    BRA     ADC_PAIR4_PROC      ; Actual Pair 4 Conversion Interrupt Handler
```

例 16-1: ADC 基本寄存器代码 (续)

```

; The actual pair conversion interrupt handler
; Don't forget to pop the stack when done and return from interrupt

ADC_PAIR0_PROC:

    ...                ; The ADC pair 0 conversion complete handler
    POP.S              ; Restore W0-W3 and SR registers
    RETFIE             ; Return from Interrupt

ADC_PAIR1_PROC:

    ...                ; The ADC pair 1 conversion complete handler
    POP.S              ; Restore W0-W3 and SR registers
    RETFIE             ; Return from Interrupt

ADC_PAIR2_PROC:

    ...                ; The ADC pair 2 conversion complete handler
    POP.S              ; Restore W0-W3 and SR registers
    RETFIE             ; Return from Interrupt

ADC_PAIR3_PROC:

    ...                ; The ADC pair 3 conversion complete handler
    POP.S              ; Restore W0-W3 and SR registers
    RETFIE             ; Return from Interrupt

ADC_PAIR4_PROC:

    ...                ; The ADC pair 4 conversion complete handler
    POP.S              ; Restore W0-W3 and SR registers
    RETFIE             ; Return from Interrupt

ADC_PAIR5_PROC:

    ...                ; The ADC pair 5 conversion complete handler
    POP.S              ; Restore W0-W3 and SR registers
    RETFIE             ; Return from Interrupt

```

16.15 更改 A/D 时钟

通常, 当 $ADON = 1$ 时, ADC 不接受对 ADC 时钟分频比的更改。如果用户在 $ADON = 1$ 时更改 A/D 时钟, 则会导致不确定的结果。

16.16 采样和转换

ADC 模块始终为采样过程分配两个 ADC 时钟周期。当每个通道工作在最大为 2 Msps 的转换速率时, 采样周期为:

$$2 \times 41.6 \text{ ns} = 83.3 \text{ ns}$$

当所选择的触发事件发生时, 在 ADCPx 寄存器中指定的各对 ADC 就会开始一次采样。当资源可用时, 就会被采样到的模拟数据进行转换。

在通道的前一次采样和转换请求被处理前, 如果同一通道上又发生了新的触发事件, 则新请求将被忽略。用户应负责使触发事件发生的频率不超出模块的转换速率。

实际转换过程需要额外的 10 个 ADC 时钟。转换是有序进行的, 先处理 bit 9, 随后为 bit 8, 最后为 bit 0。当转换完成时存储结果。

16.17 A/D 采样和转换时序

分配给输入引脚的采样保持电路具有其自身的时序逻辑，当产生外部采样和转换请求（来自 PWM 或 TMR）时就会被触发。采样保持电路具有固定的两个时钟数据

采样周期。当采集样本时，会通知 ADC 控制逻辑有一个等待处理的请求，随后当转换资源可用时，开始执行转换。

ADC 模块总是转换成对的模拟输入通道，因此典型的转换过程需要 24 个时钟周期。

图 16-3: 详细的转换序列时序，SEQSAMP = 0，空闲

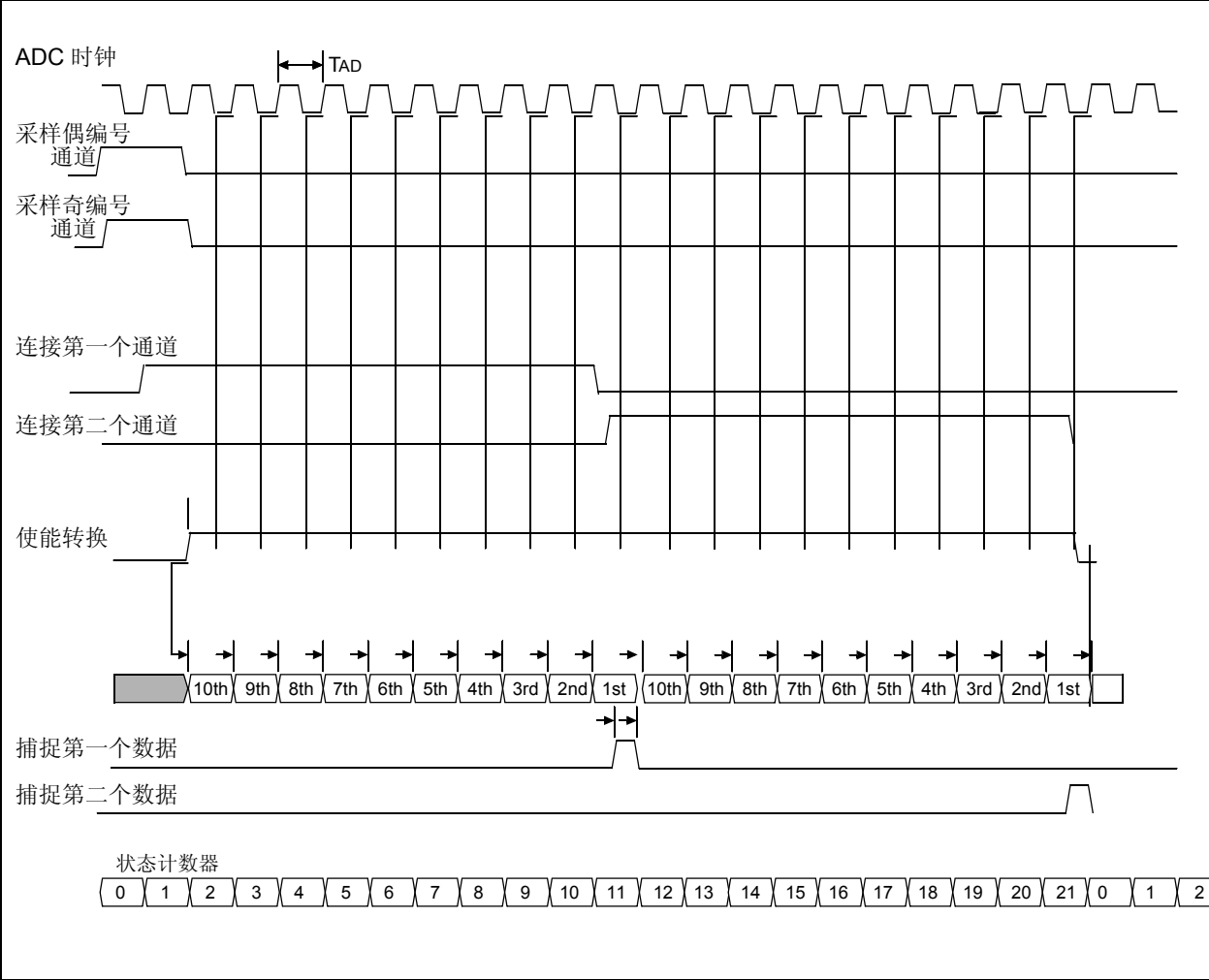
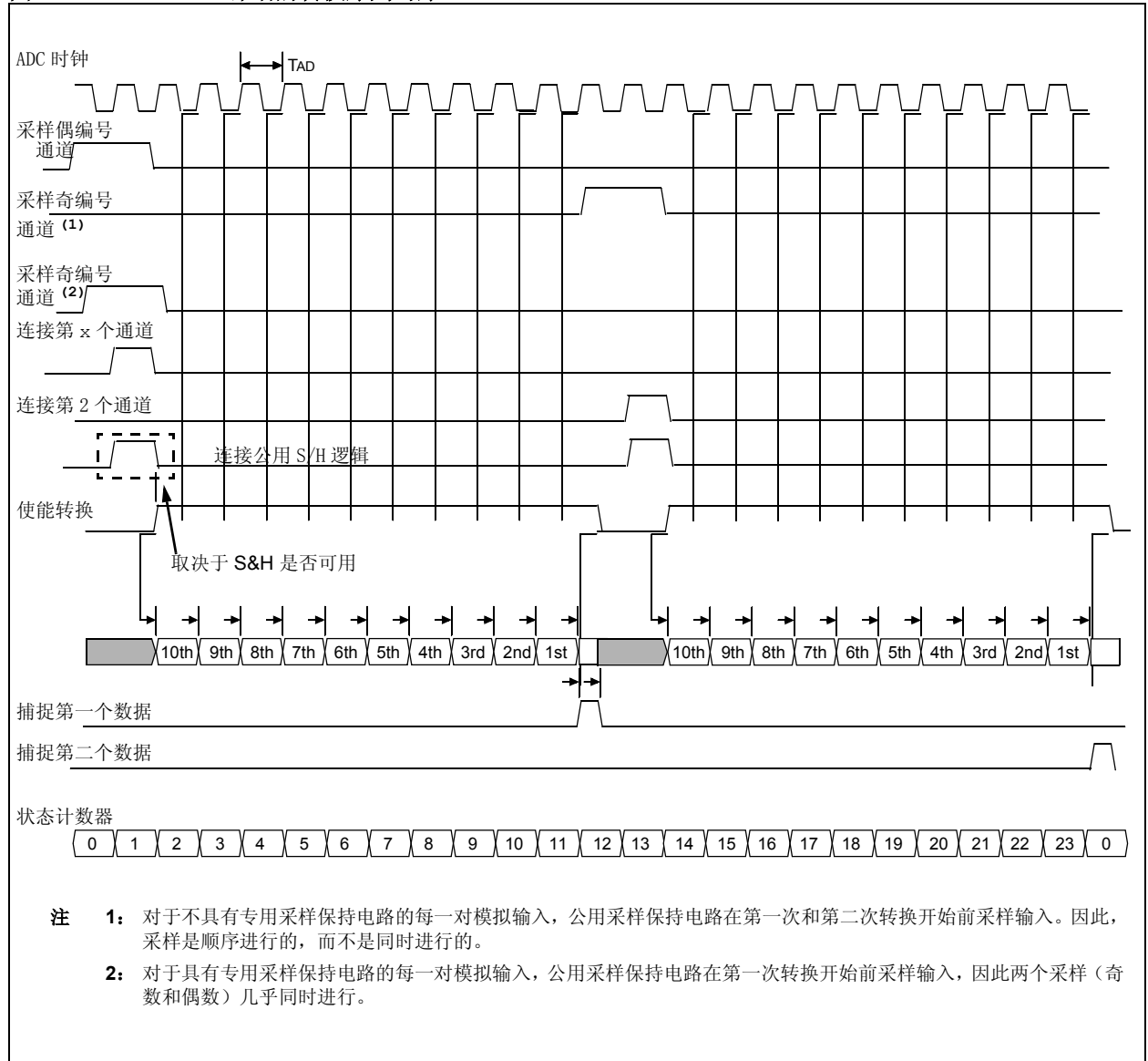


图 16-4: 详细的转换序列时序, **SEQSAMP = 1**



16.18 模块掉电模式

模块具有 2 种内部功耗模式。

当 ADON 位为 1 时，模块处于工作模式，并处于完全供电和工作状态。

当 ADON 为 0 时，模块将处于关断模式。模块的状态机复位，所有等待处理的转换请求也复位。

为从关断模式返回工作模式，用户必须等待偏置发生器进入稳定状态。稳定时间已在电气规范中指定。

16.19 复位的影响

器件复位强制所有寄存器进入复位状态。这将迫使 ADC 模块关闭并中止所有正在进行的转换和采样。ADCBUFx 寄存器中的值将不会被修改。

上电复位时，ADCBUFx 寄存器将包含未知数据。

16.20 配置模拟端口引脚

ADPCFG 和 TRIS 寄存器用于控制 A/D 端口引脚的操作。

当端口引脚作为模拟输入时，必须将其对应的 TRIS 位置 1（输入）。如果 TRIS 位被清零（输出），将对数字输出电平（V_{OH} 或 V_{OL}）进行转换。

若要将端口引脚配置为模拟输入，必须将其对应的 ADPCFG 位清零。这会将端口配置为禁止数字输入缓冲器。对 ADPCFG<n> = 1 的引脚施加模拟电平可能会导致数字输入缓冲器消耗过多的电流。

如果某一引脚未被配置为模拟输入 ADPCFG<n> = 1，则模拟输入被强制为 AV_{ss}，而对该输入进行转换不会产生有意义的结果。

当对 PORT 寄存器进行读操作时，所有配置为模拟输入（ADPCFG<n> = 0）的引脚都读为 0。

A/D 操作不受输入选择位以及 TRIS 位状态的影响。

16.21 输出格式

A/D 转换的结果为 10 位宽。数据缓冲器 RAM 为 16 位宽。可用两种格式之一读 ADC 数据，如图 16-5 所示。FORM 位选择格式。各输出格式将读出数据转换为数据总线上的 16 位结果。

图 16-5: A/D 输出数据格式

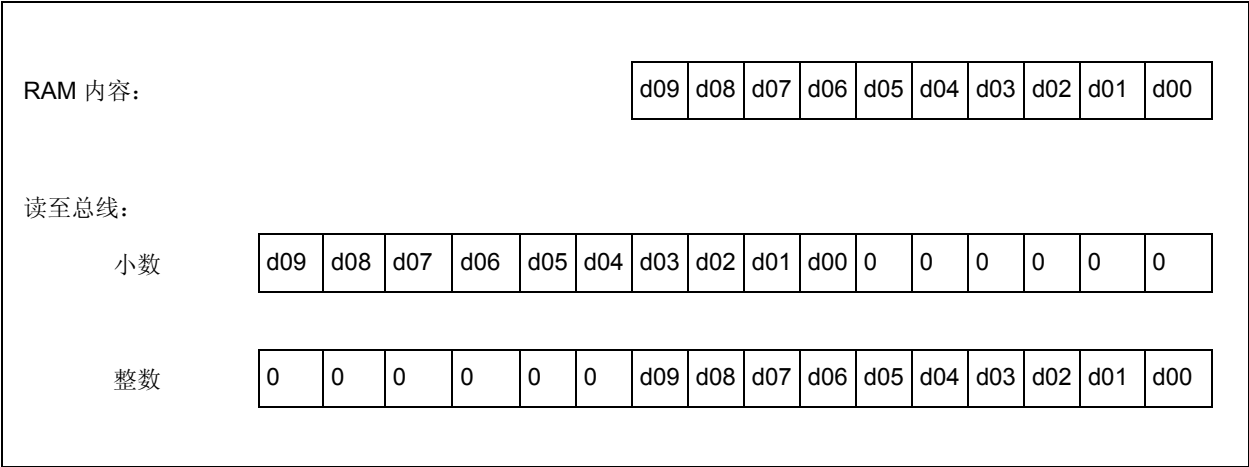


表 16-1: ADC 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
ADCON	0300	ADON	—	ADSIDL	—	—	GSWTRG	—	FORM	EIE	ORDER	SEQSAMP	—	—	ADCS<2:0>			0009
ADPCFG	0302	—	—	—	—	PCFG11	PCFG10	PCFG2	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
保留	0304	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
ADSTAT	0306	—	—	—	—	—	—	—	—	—	—	P5RDY	P4RDY	P3RDY	P2RDY	P1RDY	PORDY	0000
ADBASE	0308	ADBASE<15:1>																0000
ADCP0	030A	IRQEN1	PEND1	SWTRG1	—	—	TRGSR0<4:0>	—	—	IRQEN0	PEND0	SWTRG0	—	—	TRGSR0<4:0>			0000
ADCP1	030C	IRQEN3	PEND3	SWTRG3	—	—	TRGSR3<4:0>	—	—	IRQEN2	PEND2	SWTRG2	—	—	TRGSR2<4:0>			0000
ADCP2	030E	IRQEN5	PEND5	SWTRG5	—	—	TRGSR5<4:0>	—	—	IRQEN4	PEND4	SWTRG4	—	—	TRGSR4<4:0>			0000
保留	0310 - 031E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
ADCBUF0	0320	—	—	—	—	—	—	—	—	—	—	ADC 数据缓冲器 0						xxxx
ADCBUF1	0322	—	—	—	—	—	—	—	—	—	—	ADC 数据缓冲器 1						xxxx
ADCBUF2	0324	—	—	—	—	—	—	—	—	—	—	ADC 数据缓冲器 2						xxxx
ADCBUF3	0326	—	—	—	—	—	—	—	—	—	—	ADC 数据缓冲器 3						xxxx
ADCBUF4	0328	—	—	—	—	—	—	—	—	—	—	ADC 数据缓冲器 4						xxxx
ADCBUF5	032A	—	—	—	—	—	—	—	—	—	—	ADC 数据缓冲器 5						xxxx
ADCBUF6	032C	—	—	—	—	—	—	—	—	—	—	ADC 数据缓冲器 6						xxxx
ADCBUF7	032E	—	—	—	—	—	—	—	—	—	—	ADC 数据缓冲器 7						xxxx
ADCBUF8	0330	—	—	—	—	—	—	—	—	—	—	ADC 数据缓冲器 8						xxxx
ADCBUF9	0332	—	—	—	—	—	—	—	—	—	—	ADC 数据缓冲器 9						xxxx
ADCBUF10	0334	—	—	—	—	—	—	—	—	—	—	ADC 数据缓冲器 10						xxxx
ADCBUF11	0336	—	—	—	—	—	—	—	—	—	—	ADC 数据缓冲器 11						xxxx
保留	0338 - 037E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000

17.0 SMPS 比较器模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

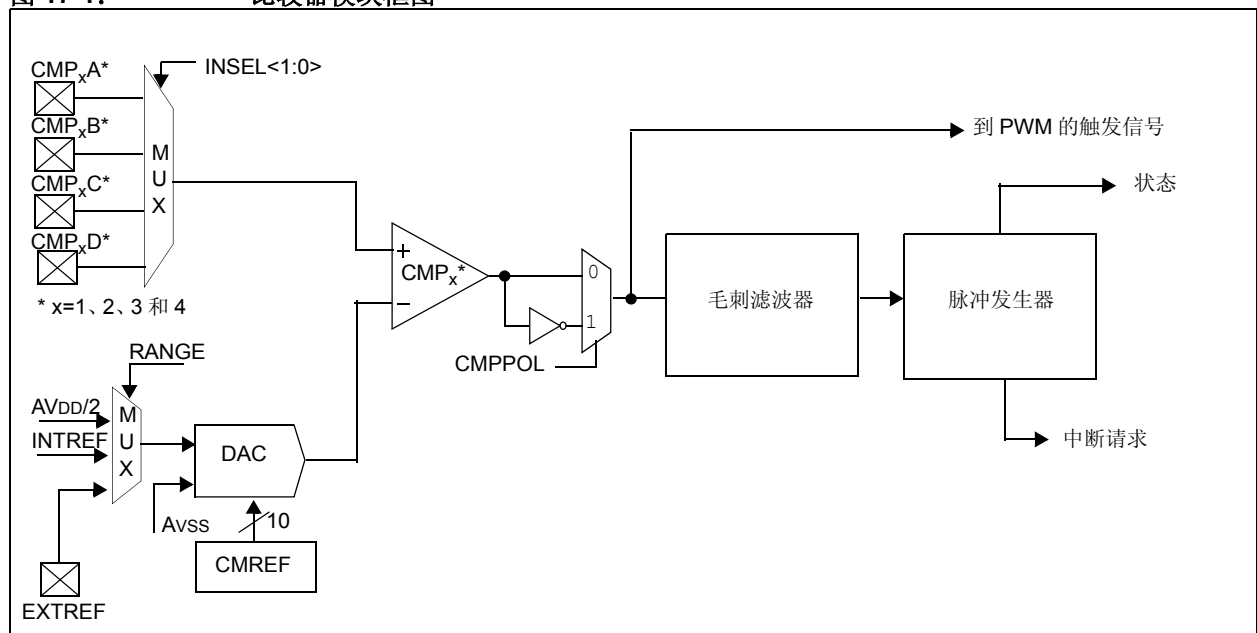
dsPIC30F SMPS 比较器模块可监视电流和 / 或电压的瞬态变化，这种变化对于 CPU 和 ADC 来说可能过快而无法捕捉。

17.1 特性概览

- 16 个比较器输入
- 10 位 DAC 提供参考电压

- 可编程输出极性
- 中断产生功能
- 可选择的输入源
- DAC 具有三种输出电压范围：
 - $AV_{DD} / 2$
 - 内部参考电压 $1.2V \pm 1\%$
 - 外部参考电压 $< (AV_{DD} - 1.6V)$
- ADC 采样和转换器触发功能
- 可被禁止以降低功耗
- 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

图 17-1: 比较器模块框图



17.2 模块应用

该模块向 SMPS dsPIC DSC 器件提供了在功率转换应用中监视电压和电流的方法。由于无需处理器和 ADC 持续监视电压或电流，便能够检测瞬态条件并激励 dsPIC DSC 处理器和 / 或外设，使 dsPIC DSC 可分身执行其他任务。

比较器模块具有一个高速比较器和一个与之相关的 10 位 DAC，该 DAC 用于为比较器的一个输入端提供可编程的参考电压。可由用户对比较器输出的极性进行编程。比较器的输出可以用在以下模式中：

- 产生中断
- 触发 ADC 采样和转换过程
- 截短 PWM 信号（限电）
- 截短 PWM 周期（电流最小）

- 禁止 PWM 输出（故障闭锁）

比较器模块的输出可以同时用在多个模式中，比如：(1) 产生中断，(2) 使 ADC 开始采样并转换采样到的信号以及 (3) 截小 PWM 输出的幅度以对检测到电压超过期望值作出响应。

当模拟输入电压超过编程设定的门限值时，比较器模块还能将系统从休眠或空闲模式唤醒。

17.3 模块说明

该比较器模块使用一个 20 ns 的比较器。比较器失调电压的典型值为 ± 5 mV。比较器的反向输入端始终与 DAC 电路连接。正向输入端与模拟多路开关相连，该多路开关用于选择所需的源引脚。

17.4 DAC

通过模拟多路开关来控制 DAC 的输出范围，可以选择 $AV_{DD}/2$ 、内部 $1.2V \pm 1\%$ 参考电压或外部参考电压源 EXTREF。当选中的输入源引脚与 ADC 共用时，通常使用 DAC 的满量程范围 ($AV_{DD}/2$)。当通过在 CLx 引脚上使用电流检测电阻来监视电平时，很可能使用略小的范围 (INTREF)。在此类应用中的待测电压通常较小 ($<1.25V$)，因此对比较器使用较小的参考电压范围可以增加这些应用中 DAC 的分辨率。使用外部参考电压可允许用户连接适合其应用的参考电压。

17.5 与 I/O 缓冲器交互

若使能比较器模块并选择了一个用作比较器源的引脚，那必须禁止该选定 I/O 引脚的数字输入缓冲器以避免由于模拟输入电压造成该缓冲器中的电流过大。

17.6 数字逻辑

CMPCONx 寄存器（见寄存器 17-1）提供了配置比较器模块的控制逻辑。数字逻辑为比较器提供了毛刺滤波器以屏蔽 2 个 T_{CY} (66 ns) 时间内的瞬态信号。在休眠或空闲模式下，毛刺滤波器被旁路以提供从比较器到中断控制器的异步通道。可使用该异步通道将处理器从休眠或空闲模式唤醒。

若 CMPSIDL 位置 1，则可在空闲模式下禁止比较器。若器件具有多个比较器，当任一 CMPSIDL 位置 1 时，整个比较器组都将在空闲模式下被禁止。这一功能降低了为该模块设计时钟控制逻辑的难度。

数字逻辑还提供了一个能产生 T_{CY} 宽度的脉冲的发生器以触发 ADC 和产生中断请求。

CMPDACx（见寄存器 17-2）寄存器提供到参考 DAC 的数字输入值。

禁止模块也将禁止 DAC 和比较器以降低功耗。

17.7 比较器输入范围

比较器将输入共模范围 (Common Mode Range, CMR) 限制在约 3.5V ($AV_{DD} - 1.5V$)。这意味着两个输入都不能超过该值，否则比较器的输出将变得无法确定。只要比较器的输入在共模范围内，比较器的输出就是正确的。在 CMR 区域内的输入漂移不会对比比较器的输出造成破坏，但却会使比较器的输入饱和。

17.8 DAC 输出范围

DAC 将最大参考电压输入范围限制在 ($AV_{DD} - 1.6$) V 内。外部参考电压输入不应超过该值，否则参考 DAC 的输出将变得无法确定。

17.9 比较器寄存器

比较器模块由以下寄存器控制：

- 比较器控制寄存器 x (CMPCONx)
- 比较器 DAC 控制寄存器 x (CMPDACx)

寄存器 17-1: 比较器控制寄存器 x (CMPCONx)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
CMPON	—	CMPSIDL	—	—	—	—	—
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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15

CMPON: A/D 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: 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。

如果器件具有多个比较器, 任何 CMPSIDL 位置 1 (= 1) 将在空闲模式下禁止所有比较器。

bit 12-8

保留: 读为 0

bit 7-6

INSEL<1:0>: 比较器的输入源选择位

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: 外部参考电压使能位

1 = 外部源给 DAC 提供参考电压

0 = 内部源给 DAC 提供参考电压

bit 4

保留: 读为 0

bit 3

CMPSTAT: 比较器输出的当前状态, 包含 CMPPOL 选择位

bit 2

保留: 读为 0

bit 1

CMPPOL: 比较器输出极性控制位

1 = 输出翻转

0 = 输出不翻转

bit 0

RANGE: DAC 输出电压范围选择位1 = 高电压范围: 最大 DAC 值 = $AV_{DD} / 2$, 2.5V (在 V_{DD} 为 5V 时)0 = 低电压范围: 最大 DAC 值 = INTREF, $1.2V \pm 1\%$

dsPIC30F1010/202X

寄存器 17-2: 比较器 DAC 控制寄存器 x (CMPDACx)

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 = 上电复位时的值		1 = 置 1		0 = 清零		x = 未知	

bit 15-10 **保留:** 读为 0
 这些位保留以供将来可能将 DAC 从 10 位扩展到更多位时使用。

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

 0000000000 = 0.0 V

表 17-1: 模拟比较器控制寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态
CMPCON1	04C0	CM PON	—	CMPSIDL	—	—	—	—	—	INSEL<1:0>	EXTREF	—	—	CMPPOL	—	—	—	0000
CMPDAC1	04C2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
CMPCON2	04C4	CM PON	—	CMPSIDL	—	—	—	—	—	INSEL<1:0>	EXTREF	—	—	CMPPOL	—	—	—	0000
CMPDAC2	04C6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
CMPCON3	04C8	CM PON	—	CMPSIDL	—	—	—	—	—	INSEL<1:0>	EXTREF	—	—	CMPPOL	—	—	—	0000
CMPDAC3	04CA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
CMPCON4	04CC	CM PON	—	CMPSIDL	—	—	—	—	—	INSEL<1:0>	EXTREF	—	—	CMPPOL	—	—	—	0000
CMPDAC4	04CE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000

注:

18.0 系统集成

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

有关器件指令集和编程的更多信息，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

本系列器件的下列功能，旨在最大限度地提高系统可靠性，通过减少外部元件的使用来降低成本，提供省电工作模式以及提供代码保护：

- 振荡器选择
- 复位：
 - 上电复位 (POR)
 - 上电延时定时器 (PWRT)
 - 振荡器起振定时器 (OST)
- 看门狗定时器 (WDT)
- 省电模式 (休眠和空闲)
- 代码保护
- 器件 ID 存储单元
- 在线串行编程 (ICSP) 功能

dsPIC30F 器件具备一个看门狗定时器，它可以通过配置位永久使能，或用软件进行控制。它依靠自身的 RC 振荡器运行，以便提升可靠性。两个定时器提供了上电时必须的延时：一个是振荡器起振定时器 (OST)，用来使芯片保持复位，直到晶体振荡器稳定为止；另一个是上电延时定时器 (PWRT)，仅在上电时提供延时，用来在电源稳定过程中使器件保持在复位状态。有了这两个片内定时器，大多数应用不再需要外部复位电路。

休眠模式，旨在提供极低电流的掉电模式。通过外部复位、看门狗定时器唤醒或中断，用户可将器件从休眠中唤醒。一些振荡器功能仍然可用，从而器件能够适用于范围宽广的各种应用。在空闲模式下，时钟源仍然处于工作状态，但 CPU 停止。RC 振荡器可降低系统成本，而 LP 晶振功能可降低功耗。

18.1 振荡器系统概述

dsPIC30F 振荡器系统包含以下模块和功能：

- 可选择多种外部和内部振荡器作为时钟源
- 片上 PLL，可提高内部工作频率
- 在各种时钟源之间进行切换的时钟切换机制
- 可编程时钟后分频器，用于节省系统功耗
- 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
- 时钟控制寄存器 OSCCON
- 用于主振荡器选择的配置位

配置位用于在上电复位 (POR) 时确定时钟源。此后，可在许可的时钟源之间切换时钟源。OSCCON 寄存器用于控制时钟切换并包含与系统时钟相关的状态位。

注：在 dsPIC30F1010/202X 器件上，32 kHz 晶振操作未被使能。

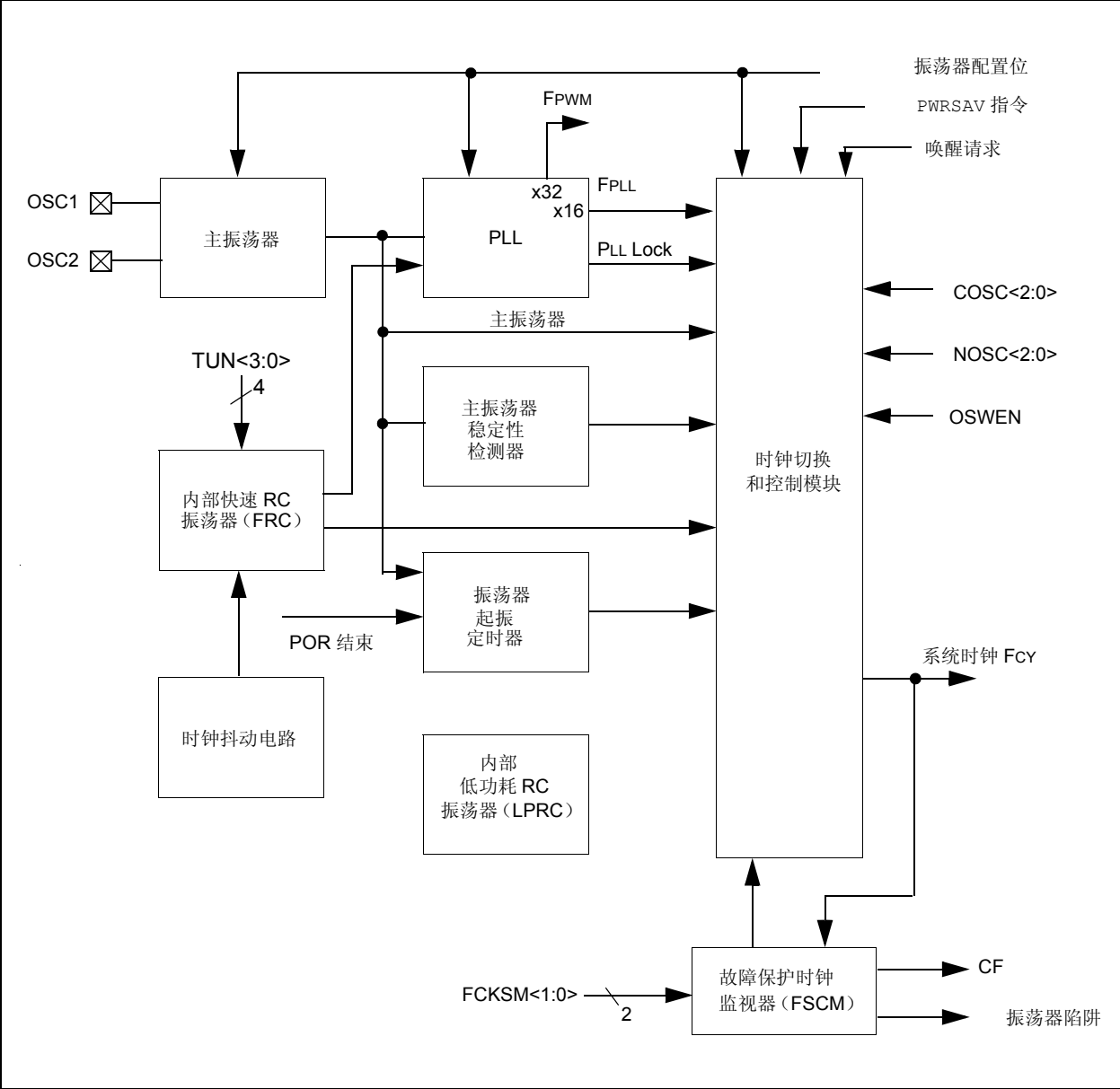
图 18-1 给出了振荡器系统的简化框图。

18.2 振荡器控制寄存器

振荡器是由下列寄存器控制的：

- OSCCON：振荡器控制寄存器
- OSCTUN2：振荡器调节寄存器 2
- LFSR：线性反馈移位寄存器
- FOSCSEL：振荡器选择配置位
- FOSC：振荡器选择配置位

图 18-1: 振荡器系统框图



寄存器 18-1: OSCCON: 振荡器控制寄存器

U-0	R-y HS,HC	R-y HS,HC	R-y HS,HC	U-0	R/W-y	R/W-y	R/W-y
—	COSC<2:0>			—	NOSC<2:0>		
bit 15				bit 8			

R/W-0	U-0	R-0 HS,HC	R/W-0	R/C-0 HS,HC	R/W-0	U-0	R/W-0 HC
CLKLOCK	—	LOCK	PRCDEN	CF	TSEQEN	—	OSWEN
bit 7				bit 0			

图注:	x = 未知位	
R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零
HC = 由硬件清零	HS = 由硬件置 1	-y = 上电复位时由配置位设置的值

bit 15 **未实现:** 读为 0

bit 14-12 **COSC<2:0>:** 电流振荡器组选择位 (只读)
 000 = 快速 RC 振荡器 (FRC)
 001 = 带 PLL 模块的快速 RC 振荡器 (FRC)
 010 = 主振荡器 (HS 或 EC)
 011 = 带 PLL 模块的主振荡器 (HS 或 EC)
 100 = 保留
 101 = 保留
 110 = 保留
 111 = 保留
 下列情况时该位复位:
 上电复位时设置为 FRC 的值 (000)
 在成功完成时钟切换时, 装入 NOSC<2:0> 的值
 当 FSCM 检测到故障并将时钟切换为 FRC 时, 设置为 FRC 的值 (000)

bit 11 **未实现:** 读为 0

bit 10-8 **NOSC<2:0>:** 新振荡器组选择位
 000 = 快速 RC 振荡器 (FRC)
 001 = 带 PLL 模块的快速 RC 振荡器 (FRC)
 010 = 主振荡器 (HS 或 EC)
 011 = 带 PLL 模块的主振荡器 (HS 或 EC)
 100 = 保留
 101 = 保留
 110 = 保留
 111 = 保留

bit 7 **CLKLOCK:** 时钟锁定使能位
 1 = 如果 (FCKSM1 = 1), 时钟和 PLL 配置被锁定
 如果 (FCKSM1 = 0), 时钟和 PLL 配置可以被修改
 0 = 未锁定时钟和 PLL 选择, 可以修改配置

注: 一旦置 1, 此位只能通过复位清零。

bit 6 **未实现:** 读为 0

dsPIC30F1010/202X

寄存器 18-1: OSCCON: 振荡器控制寄存器 (续)

bit 5	LOCK: PLL 锁定状态位 (只读) 1 = 表示 PLL 处于锁定状态 0 = 表示 PLL 处于失锁 (或禁止) 状态 下列情况时该位复位: 上电复位时复位 由时钟切换状态机启动一个有效的时钟切换序列时复位 当 PLL 启动后达到 PLL 锁定状态时置 1 锁定丢失时复位 PLL 未被用作第 1 组系统时钟时读为 0
bit 4	PRCDEN: 伪随机时钟抖动使能位 1 = 使能伪随机时钟抖动 0 = 禁止伪随机时钟抖动
bit 3	CF: 时钟故障检测位 (读 / 可由应用程序清零) 1 = FSCM 已检测到时钟故障 0 = FSCM 未检测到时钟故障 下列情况时该位复位: 上电复位时复位 由时钟切换状态机启动一个有效的时钟切换序列时复位 检测到时钟故障后置 1
bit 2	TSEQEN: FRC 调谐定序器 (Tune Sequencer) 使能位 1 = OSCTUN和OSCTUN2寄存器中的TUN<3:0>和TSEQ1<3:0>至TSEQ7<3:0>位依次调谐FRC振荡器。由 PWM 模块的 ROLL<2:0> 信号依次选择每个字段。 0 = 由 OSCTUN 寄存器的 TUN<3:0> 位调整 FRC 振荡器
bit 1	未实现: 读为 0
bit 0	OSWEN: 振荡器切换使能位 1 = 请求振荡器切换到由 NOSC<1:0> 指定的选择 0 = 完成振荡器切换 下列情况时该位复位: 上电复位时复位 时钟切换成功后复位 冗余时钟切换后复位 FSCM 将振荡器切换到 (第 3 组) FRC 后复位

寄存器 18-2: OSCTUN: 振荡器调节寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TSEQ3<3:0>				TSEQ2<3:0>			
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TSEQ1<3:0>				TUN<3:0>			
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知位

bit 15-12 **TSEQ3<3:0>**: 调整序列值 3 位

当 PWM ROLL<2:0> = 011 时, 使用此字段替代 TUN<3:0> 对 FRC 进行频率校准。

bit 11-8 **TSEQ2<3:0>**: 调整序列值 2 位

当 PWM ROLL<2:0> = 010 时, 使用此字段替代 TUN<3:0> 对 FRC 进行频率校准。

bit 7-4 **TSEQ1<3:0>**: 调整序列值 1 位

当 PWM ROLL<2:0> = 001 时, 使用此字段替代 TUN<3:0> 对 FRC 进行频率校准。

bit 3-0 **TUN<3:0>**: 指定用户对内部快速 RC 振荡器的频率调整范围。如果 OSCCON 寄存器中的 TSEQEN 位被置 1, 此字段将与 TSEQ1 至 TSEQ7 位一起按顺序调整 FRC 振荡器的频率值。

0111 = 最大频率

0110 =

0101 =

0100 =

0011 =

0010 =

0001 =

0000 = 中心频率, 振荡器运行在校准频率下

1111 =

1110 =

1101 =

1100 =

1011 =

1010 =

1001 =

1000 = 最小频率

dsPIC30F1010/202X

寄存器 18-3: OSCTUN2: 振荡器调节寄存器 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TSEQ7<3:0>				TSEQ6<3:0>			
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TSEQ5<3:0>				TSEQ4<3:0>			
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知位

- bit 15-12 **TSEQ7<3:0>**: 调整序列值 7 位
当 PWM ROLL<2:0> = 111 时, 使用此字段替代 TUN<3:0> 对 FRC 进行频率校准。
- bit 11-8 **TSEQ6<3:0>**: 调整序列值 6 位
当 PWM ROLL<2:0> = 110 时, 使用此字段替代 TUN<3:0> 对 FRC 进行频率校准。
- bit 7-4 **TSEQ5<3:0>**: 调整序列值 5 位
当 PWM ROLL<2:0> = 101 时, 使用此字段替代 TUN<3:0> 对 FRC 进行频率校准。
- bit 3-0 **TSEQ4<3:0>**: 调整序列值 4 位
当 PWM ROLL<2:0> = 100 时, 使用此字段替代 TUN<3:0> 对 FRC 进行频率校准。

寄存器 18-4: LFSR: 线性反馈移位寄存器

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	LFSR<14:8>						
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LFSR<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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知位

- bit 15 **未实现**: 读为 0
当 PWM ROLL<2:0> = 111 时, 使用此字段替代 TUN<3:0> 对 FRC 进行频率校准。
- bit 14-8 **LFSR <14:8>**: 伪随机 FRC 调整值位的高 7 位
- bit 7-0 **LFSR <7:0>**: 伪随机 FRC 调整值位的低 8 位

寄存器 18-5: FOSCSEL: 振荡器选择配置位

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23				bit 16			

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

U-0	U-0	U-0	U-0	U-0	U-0	R/P	R/P
—	—	—	—	—	—	FNOSC1	FNOSC0
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知位

bit 23-2 未实现: 读为 0

bit 1-0 **FNOSC<1:0>**: 上电复位时启动振荡器组选择位

00 = 快速 RC 振荡器 (FRC)

01 = 带 PLL 模块的快速 RC 振荡器 (FRC)

10 = 主振荡器 (HS 或 EC)

11 = 带 PLL 模块的主振荡器 (HS 或 EC)

dsPIC30F1010/202X

寄存器 18-6: FOSC: 振荡器选择配置位

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23						bit 16	

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

R/P	R/P	R/P	U-0	U-0	R/P	R/P	R/P
FCKSM<1:0>		FRANGE	—	—	OSCIOFNC	POSCMD<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 = 上电复位时的值	1 = 置 1	0 = 清零	x = 未知位

bit 23-8

未实现：读为 0

bit 7-6

FCKSM<1:0>：时钟切换和监视选择配置位

1x = 禁止时钟切换，禁止故障保护时钟监视

01 = 使能时钟切换，禁止故障保护时钟监视

00 = 使能时钟切换，使能故障保护时钟监视

bit 5

FRANGE：FRC 和 PLL 的频率范围选择位

与“换挡（Gear Shift）”功能类似，其允许 dsPIC DSC 器件在供电电压降低时（3.3V），工作在降低的 MIPS。

FRANGE 位值	温度 额定值	FRC 频率 （标称值）	PLL VCO （标称值）
1 = 较大范围	工业级 扩展级	14.55 MHz 9.7 MHz	466 MHz（480 MHz 最大值） 310 MHz（320 MHz 最大值）
0 = 较小范围	工业级 扩展级	9.7 MHz 6.4 MHz	310 MHz（320 MHz 最大值） 205 MHz（211 MHz 最大值）

bit 4-3

未实现：读为 0

bit 3

OSCIOFNC：OSC2 I/O 引脚使能位

1 = OSCO 引脚的 CLKO 输出信号有效

0 = 禁止 CLKO 输出

bit 1-0

POSCMD<1:0>：主振荡器模式

11 = 禁止主振荡器

10 = 选择 HS 振荡器模式

01 = 保留

00 = 选择外部时钟模式

18.2.1 防止意外写入

由于 OSCCON 寄存器控制时钟切换和时钟分频，因此有意将写入 OSCCON 寄存器的操作过程设计得较为困难。要写入 OSCCON 低位字节，必须执行以下代码序列且不应在其中插入任何其他指令：

- 写入字节 “46h” 到 OSCCON 低位字节
- 写入字节 “57h” 到 OSCCON 低位字节
- 允许在一个指令周期内对 OSCCONL 进行字节写操作
`mov.b W0, OSCCON`

要写入 OSCCON 的高字节，必须执行以下代码序列且不应在其中插入任何其他指令：

- 写入字节 “78h” 到 OSCCON 的低位字节
- 写入字节 “9Ah” 到 OSCCON 的低位字节
- 允许在一个指令周期内对 OSCCONL 进行字节写操作
`mov.b W0, OSCCON + 1`

18.3 振荡器配置

图 18-2 给出了系统时钟 Fcy 的来源。图 18-1 中的 PLL 输出的最大频率为 480 MHz（带 PLL 的高速 FRC 选项，工业级温度范围器件，且将 TUN<3:0> 位设置为 0111）。此信号供电源 PWM 模块使用，是 PLL 输入频率的 32 倍。

假设使用额定工业级温度范围的器件，选择高速 FRC 选项，对 480 MHz PLL 时钟信号进行 2 分频，从而提供驱动 ADC 模块的 240 MHz 信号。同一 480 MHz 信号还被 8 分频产生 60 MHz 信号，作为 Fcy 倍频器的输入之一。倍频器的另一个输入是 FOSC 输入时钟源（主振荡器或 FRC）的 2 分频值。使能 PLL 时，Fcy = FPLL/16。当禁止 PLL 时，Fcy = Fosc/2。

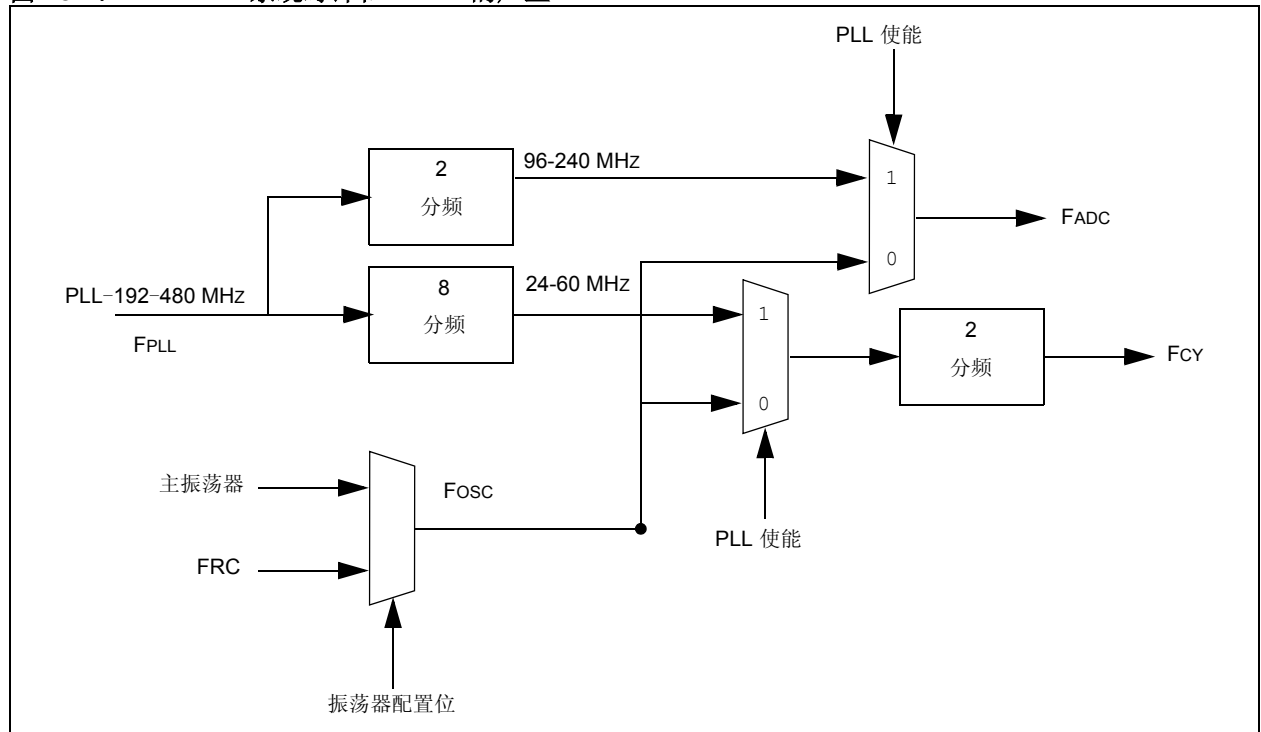
此方法源于 480 MHz 时钟：

- 带大范围选项且 TUN<3:0> = 0111 时的 FRC 时钟为 15 MHz
- 使能 PLL
- PWM 时钟 = 15 x 32 = 480 MHz
- Fcy = 480 MHz/16 = 30 MHz = 30 MIPS

如果禁止 PLL，

- FRC 时钟（带大范围选项且 TUN<3:0> = 0111 时）为 15 MHz
- Fcy = 15 MHz/2 = 7.5 MHz = 7.5 MIPS

图 18-2: 系统时钟和 FADC 的产生



dsPIC30F1010/202X

18.3.1 初始时钟源选择

当退出上电复位时，器件根据以下原则进行时钟源的选择：

- 通过 **FNOSC<1:0>** 配置位选择三种振荡器组中的一种（HS、EC 或 FRC）
- POSCMD1<1:0>** 配置位选择主振荡器模式
- OSCIOFNC** 选择 **OSC2** 引脚作为 I/O 或是时钟输出

表 18-1 所示为时钟选择。

表 18-1: 时钟选择的配置位值

振荡器模式	振荡器源	FNOSC<1:0>		POSCMD<1:0>		OSCIOFNC	OSC2 功能	OSC1 功能
		Bit 1	Bit 0	Bit 1	Bit 0			
HS w/PLL 32x	PLL	1	1	1	0	N/A	CLKO ⁽¹⁾	CLKI
FRC w/PLL 32x	PLL	0	1	1	1	1	CLKO	I/O
FRC w/PLL 32x	PLL	0	1	1	1	0	I/O	I/O
EC w/PLL 32x	PLL	1	1	0	0	1	CLKO	CLKI
EC w/PLL 32x	PLL	1	1	0	0	0	I/O	CLKI
EC ⁽²⁾	外部	1	0	0	0	1	CLKO	CLKI
EC ⁽²⁾	外部	1	0	0	0	0	I/O	CLKI
HS ⁽²⁾	外部	1	0	1	0	N/A	CLKO ⁽¹⁾	CLKI
FRC ⁽²⁾	内部 RC	0	0	1	1	0	I/O	I/O
FRC ⁽²⁾	内部 RC	0	0	1	1	1	CLKO	I/O

注 1：不建议使用 **CLKO** 驱动外部电路。

2：不建议某些应用使用此模式；禁止 32x PLL 将不允许进行高速 ADC 和 PWM 操作。

18.3.2 振荡器起振定时器（OST）

为确保晶振（或陶瓷振荡器）已起振并达到稳定状态，振荡器中包含一个振荡器起振定时器。该定时器仅仅是一个 10 位计数器，在释放振荡器时钟使其用于系统其他部分之前，该计数器将计数 1024 个 **Tosc** 周期。超时周期称为 **Tost**。每当振荡器需要重启时（即上电复位和由休眠模式唤醒时）需进行 **Tost** 时间的定时。振荡器起振定时器适用于主振荡器的 **HS** 振荡器模式（从休眠模式唤醒和上电复位时）。

18.3.3 锁相环（PLL）

PLL 可将主振荡器产生的时钟信号进行倍频。PLL 仅可选择 x32 的增益。表 18-2 中汇总了输入和输出频率范围。

表 18-2: PLL 频率范围

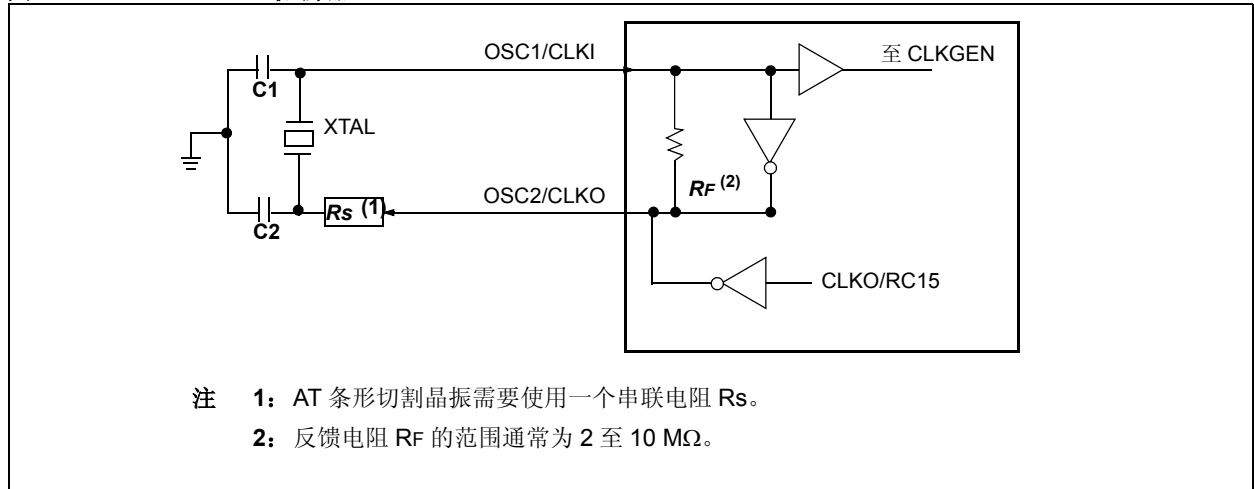
F _{IN}	PLL 倍频	F _{OUT}
6.4 MHz	x32	205 MHz
9.7 MHz	x32	310 MHz
14.55 MHz	x32	466 MHz

PLL 具有锁定输出，当 PLL 进入锁相状态时，锁定输出有效。如果锁相环锁定失败（例如由噪声导致），锁定信号将为无效。此信号的状态将通过 **OSCCON** 寄存器中的只读位 **LOCK** 反映。

18.4 OSC1/OSC2 引脚的主振荡器:

图 18-3 给出了主振荡器的用法。

图 18-3: 主振荡器



18.5 外部时钟输入

有 2 个主振荡器模式使用外部时钟，它们是 EC 和带 IO 的 EC 模式。

在 EC 模式下（图 18-4），OSC1 引脚可以由 CMOS 驱动器驱动。在此模式下，OSC1 引脚为高阻态且 OSC2 引脚为时钟输出（ $F_{osc}/2$ ）。此输出时钟对于测试或同步非常有用。

在带 IO 的 EC 模式下（图 18-5），OSC1 引脚可由 CMOS 驱动器驱动。此模式下，OSC1 引脚为高阻态，且 OSC2 引脚为通用 I/O 引脚。OSC1 和 OSC2 之间的反馈器件被关断以节省电流。

图 18-4: 外部时钟输入操作（EC 振荡器配置）

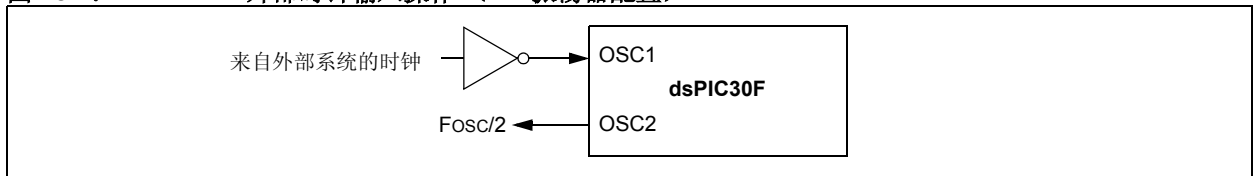
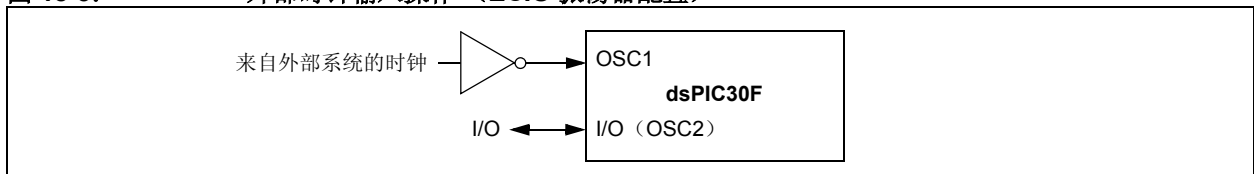


图 18-5: 外部时钟输入操作（ECIO 振荡器配置）



18.6 内部快速 RC 振荡器 (FRC)

FRC 是一种快速并具有精确频率的内部 RC 振荡器。FRC 振荡器运行在 6.4/9.7/14.55 MHz (误差 $<\pm 2\%$) 的频率下。FRC 振荡器旨在具有足够的精度以提供所需的时钟频率, 该时钟频率用于保持串行数据传输的波特率容差。用户可以在 $\pm 3\%$ 的范围内调整 FRC 频率。

FRC 振荡器工作在下列情况:

- 任何未选中 EC 或 HS 振荡器模式的情况。
- 当使能故障保护时钟监视器时, 且检测到时钟故障时, 强制切换到 FRC。

18.6.1 频率范围选择

FRC 模块具有可以选择小范围 (用于工业级温度范围内的器件频率为 9.7 MHz, 用于扩展级温度范围内的器件频率为 6.4 MHz) 或大范围 (用于工业级温度范围内的器件频率为 14.55 MHz, 用于扩展级温度范围内的器件频率为 9.7 MHz) 的“变速”控制信号。此功能允许 dsPIC DSC 器件在 3.3V 时的最大工作速率为 20 MIPS 或在 5.0V 时的最大工作速率为 30 MIPS, 这些值都在系统规范的范围之内。

18.6.2 标称频率值

选择小范围选项时, 对于工作在工业级温度范围的器件, FRC 模块被校准为标称值 9.7 MHz, 对于工作在扩展级温度范围的器件, FRC 模块被校准为标称值 6.4 MHz, 选择大范围选择时, 对于工作在工业级温度范围的器件, FRC 模块被校准为标称值 14.55 MHz, 对于工作在扩展级温度范围的器件, FRC 模块被校准为标称值 9.7 MHz。此功能允许用户在 $\pm 3\%$ 的范围内“调整”FRC 频率, 且仍将频率限制在系统规范内。

18.6.3 用户调整 FRC 频率

出厂时将 FRC 校准为标称值 6.4/9.7/14.55 MHz。用户可使用 OSCTUN 寄存器中的 TUN<3:0> 字段在应用中调整 FRC 振荡器的频率。

由 OSCTUN 或 OSCTUN2 寄存器提供此 4 位调整控制信号, 具体是哪个寄存器是由 OSCCON 寄存器中的 TSEQEN 位决定的。

14.55 MHz 振荡器的调整范围为标称值 ± 0.45 MHz ($\pm 3\%$)。

可在用户应用中调整基本频率。此频率调整功能允许用户偏离出厂时的校准值。用户可通过写 OSCTUN 寄存器的 TUN<3:0> 位调整频率。

18.6.4 时钟抖动逻辑

在电源转换应用中, 设计人员要消除的主要电气噪声是来在 PWM 频率上切换时的功率晶体管。改变 SMPS dsPIC DSC 的系统时钟频率时, PWM 频率将随之改变, 这将减小 EMI 的峰值, 此时噪声的覆盖范围变大了。

通常频率变化范围很小。dsPIC30F1010/202X 根据 PWM 循环基可以提供两种改变系统时钟频率的方法。它们分别为频率序列模式和伪随机时钟抖动模式, 表 18-8 给出了实现这 2 种方法的详细信息。

18.6.5 频率序列模式

频率序列模式允许 PWM 模块选择 8 个不同的 FRC TUN 序列值, 以在每个主 PWM 时基翻转时改变系统频率。当 OSCCON 寄存器的 TSEQEN 位被置 1 时, OSCTUN 和 OSCTUN2 寄存器允许用户指定 8 个序列调整值。如果 TSEQEN 位为零, 则只有 TUN 位会影响 FRC 频率。

带 8 个输入的 4 位宽倍频器由 TUN 和 TSEQx 位字段选择调整频率。倍频器是由 PWM 模块的 ROLL<5:3> 计数器控制的。每达到一个周期值后, 主时基发生翻转时, ROLL<5:3> 计数器递增。

18.6.6 伪随机时钟抖动模式

伪随机时钟抖动 (Pseudo Random Clock Dither, PRCD) 逻辑是由 15 位 LFSR 实现的 (线性反馈移位寄存器), 它是一个带有几个逻辑异或门的移位寄存器。LFSR 的低 4 位提供了 FRC 的 TUNE 位。通过设置 OSCCON 寄存器中的 PRCDEN 位使能 PRCD 功能。LSFR 则在每次 ROLL<3> 位状态改变时“随时钟变化 (clocked)” (使能为随时钟变化), ROLL<3> 位每 8 个 PWM 周期改变一次状态。

18.6.7 故障保护时钟监视器

故障保护时钟监视器 (FSCM), 使得器件能够继续工作, 即便是振荡器出现故障。恰当地设置 FOSC 器件配置寄存器中的 FCKSM 配置位 (时钟切换和监视器选择位), 可使能 FSCM 功能

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。之后, 用户即可选择尝试重新启动振荡器, 或者执行受控关闭操作。

用户可将陷阱当作热复位来处理，这只要把复位地址载入振荡器陷阱向量即可。出现故障时，一旦识别出时钟故障，CF（时钟故障）状态位（OSCCON<3>）也将置 1。

出现时钟故障时，WDT 不受影响，继续靠 LPRC 时钟运行。

从 POR 或休眠退出后，如果振荡器起振时间非常慢的话，则可能出现这样的情况，在振荡器启动之前 PWRT 定时器就已到期。在这样的情形下，FSCM 将被激活，FSCM 将启动一个时钟故障陷阱，FRC 振荡器选择将被载入 COSC<2:0> 位中。这实际上是关闭了先前正在试图启动的振荡器。

在时钟故障陷阱 ISR 中，用户可以检测到此种情况并重启振荡器。

检测到时钟故障时，FSCM 模块将启动至 FRC 振荡器的时钟切换，方法如下：

1. 把 FRC 振荡器选择位的值载入 COSC 位（OSCCON<14:12>）中。
2. CF 位（OSCCON<3>）置 1。
3. OSWEN 控制位（OSCCON<0>）清零。

出于时钟切换的目的，时钟源划分为两组：

1. 主组
2. 内部 FRC 组

用户可以在这些功能组之间进行切换，但不能在同一个组内进行切换。如果选择了主组的话，则在主组内的选择始终由 FNOSC<1:0> 配置位决定。

与时钟切换相关的控制和状态位在 OSCCON 寄存器中。如果配置位 FCKSM<1:0> = 1x，那么时钟切换和故障保护时钟监视器功能被禁止。这是缺省的配置位设置。

如果时钟切换禁止，则 FNOSC<1:0> 和 POSCMD<1:0> 位直接控制振荡器选择，COSC<2:0> 位不控制时钟选择。然而，这些位将反映时钟源选择。

注： 使能故障保护时钟监视器时，在应用中不应尝试将时钟频率切换至 100 KHz 以下。如果执行此了这样的时钟切换，器件可能会产生振荡器故障陷阱并切换至快速 RC 振荡器。

18.7 复位

PIC18F1220/1320 区分下列各种复位：

- a) 上电复位（POR）
- b) 正常工作期间的 MCLR 复位
- c) 休眠期间的 MCLR 复位
- d) 看门狗定时器（WDT）复位（在正常工作期间）
- e) RESET 指令
- f) 由陷阱锁定（TRAPR）导致的复位
- g) 由非法操作码导致的复位，或者是由于把未初始化的 W 寄存器用作地址指针（IOPUWR）而导致的复位

各种复位条件以不同的方式影响不同的寄存器。大多数寄存器不受 WDT 唤醒的影响，因为这被视为是正常工作的继续。在不同的复位条件下，将以不同的方式置 1 或清零 RCON 寄存器中的状态位，如表 18-3 所示。软件中使用这些位来确定复位的性质。

片内复位电路的框图，如图 18-7 所示。

在 MCLR 复位路径上，提供了一个 MCLR 噪声滤波器。该滤波器检测并滤除小脉冲。

内部产生的复位不会将 MCLR 引脚驱动为低电平。

图 18-6: FRC 调整抖动逻辑框图

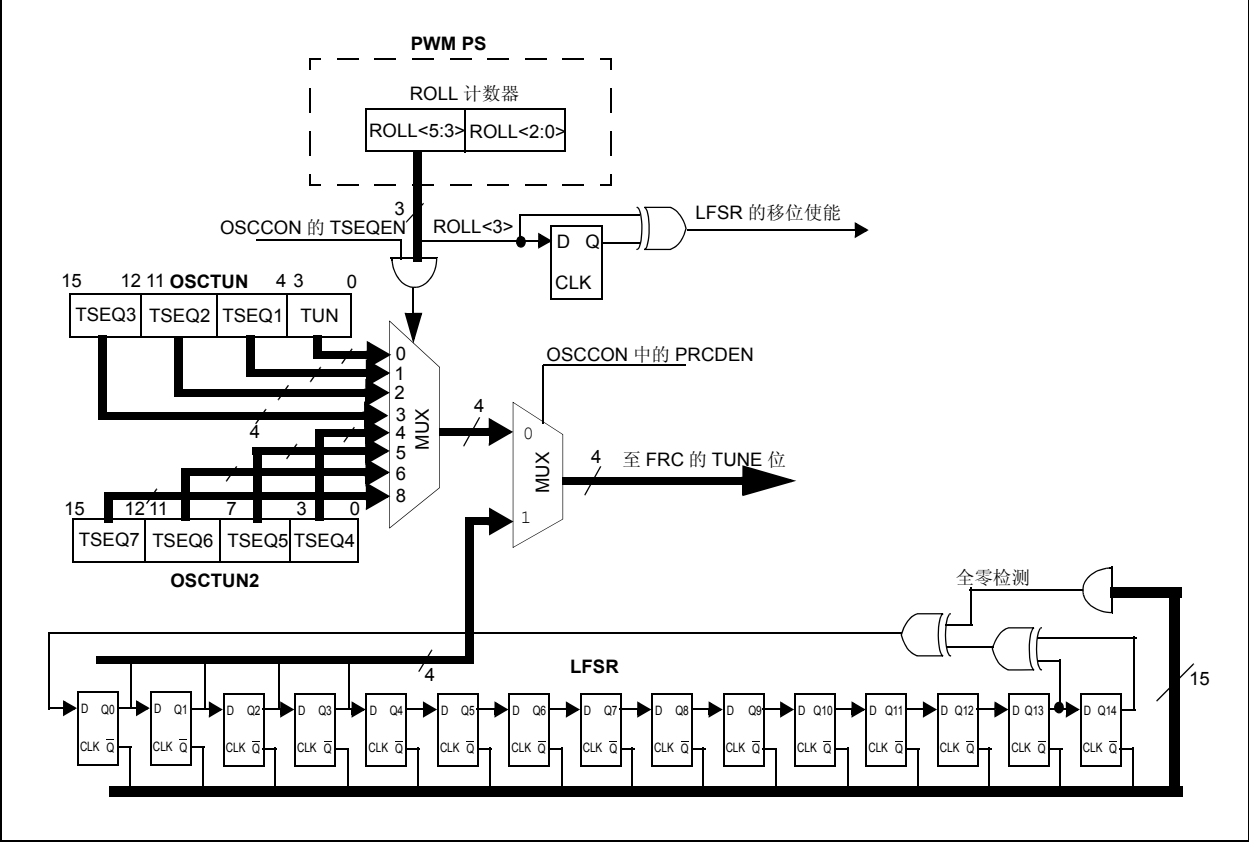
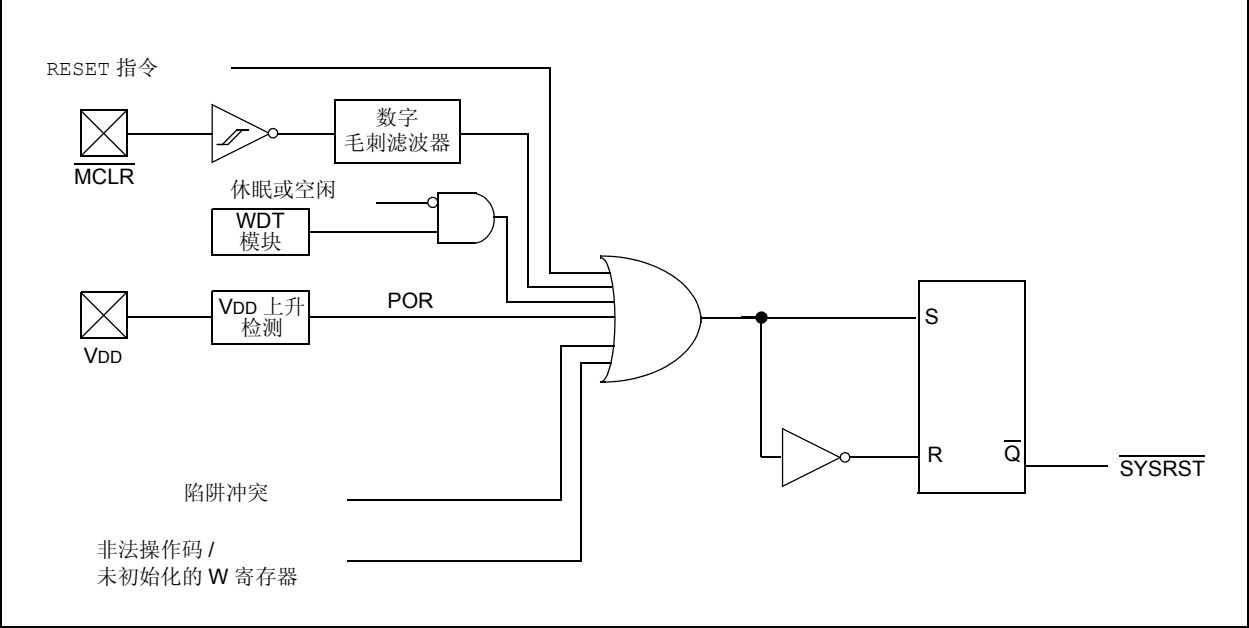


图 18-7: 复位系统框图



18.7.1 POR: 上电复位

当检测到 V_{DD} 上升时，上电事件将产生内部 POR 脉冲。复位脉冲将在 POR 电路阈值电压 (V_{POR}) 处产生，标称阈值为 1.85V。器件电源电压特性曲线必须满足规定的起始电压和上升速度要求。POR 脉冲将复位 POR 定时器，并使器件进入复位状态。POR 还将选择器件时钟源，时钟源由振荡器配置位指定。

POR 电路将插入一个标称值为 $10\ \mu\text{s}$ 的小延时 $TPOR$ ，确保器件偏压电路是稳定的。此外，还要施加用户选择的上电延时 ($TPWRT$)。 $TPWRT$ 参数由器件配置位决定的，可为 0 ms（无延时）、4 ms、16 ms 或 64 ms。器件上电的总延时为 $TPOR + TPWRT$ 。在这些延时结束后， $\overline{\text{SYSRST}}$ 将在 Q1 时钟的下一个上升沿反相，PC 将跳转至复位向量。

$\overline{\text{SYSRST}}$ 信号的时序如图 18-8 至图 18-10 所示。

图 18-8: 上电过程中的延时时序 ($\overline{\text{MCLR}}$ 连接至 V_{DD})

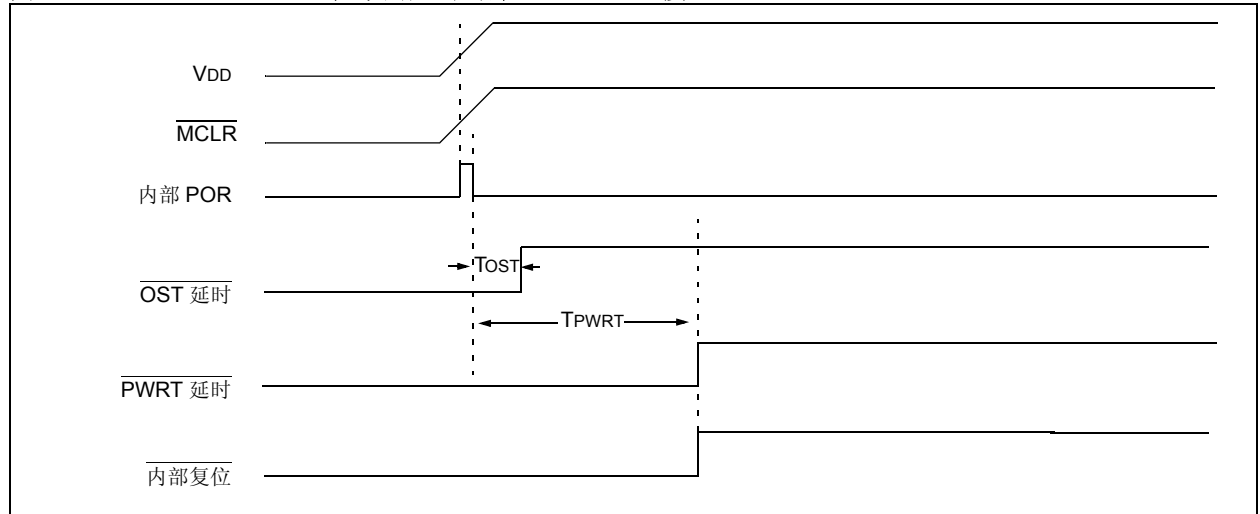


图 18-9: 上电过程中的延时时序 ($\overline{\text{MCLR}}$ 未连接至 V_{DD}): 情形 1

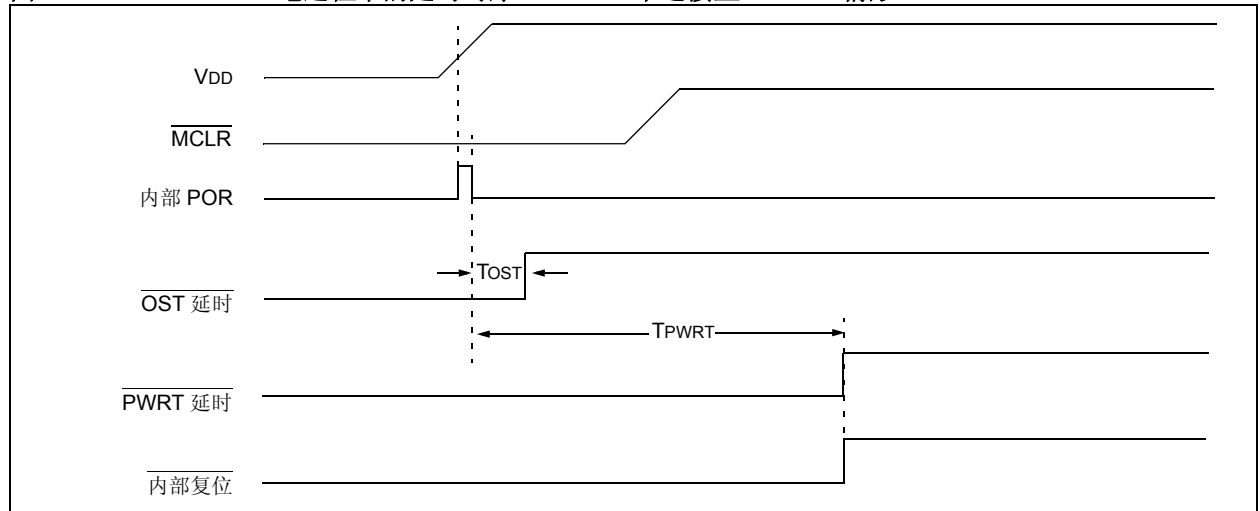
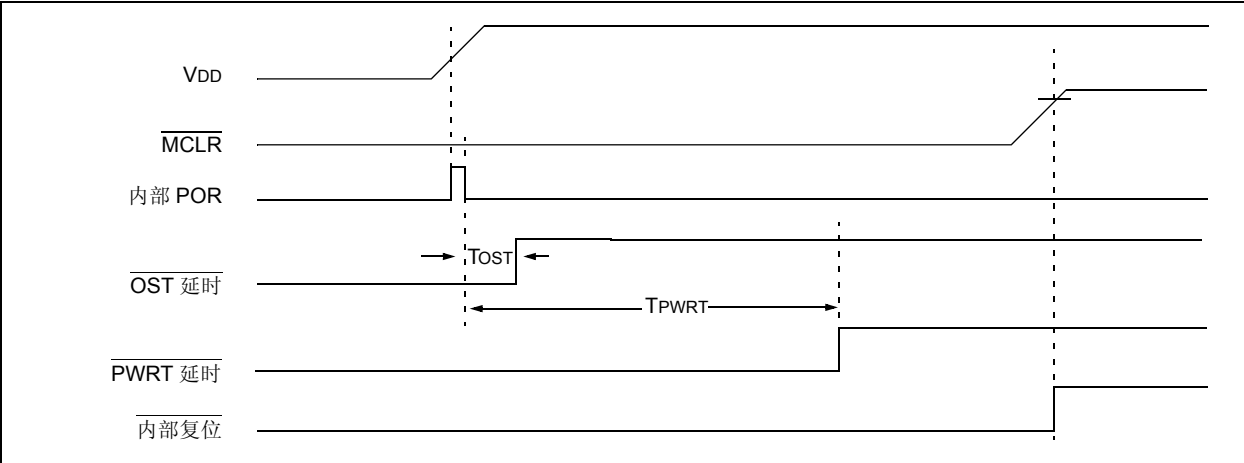


图 18-10: 上电过程中的延时时序 ($\overline{\text{MCLR}}$ 未连接至 V_{DD}): 情形 2



18.7.1.1 具有长晶振起振时间的 POR (FSCM 使能)

振荡器起振电路没有连接到 POR 电路。一些晶振电路 (尤其是低频晶振) 具有相对长的起振时间。从而, 在 POR 定时器和 PWRT 超时后, 可能会出现下列情况:

- 振荡器电路还没有开始振荡。
- 振荡器起振定时器还没有超时 (如果使用晶振的话)。
- PLL 还未锁定 (如果使用了 PLL 的话)。

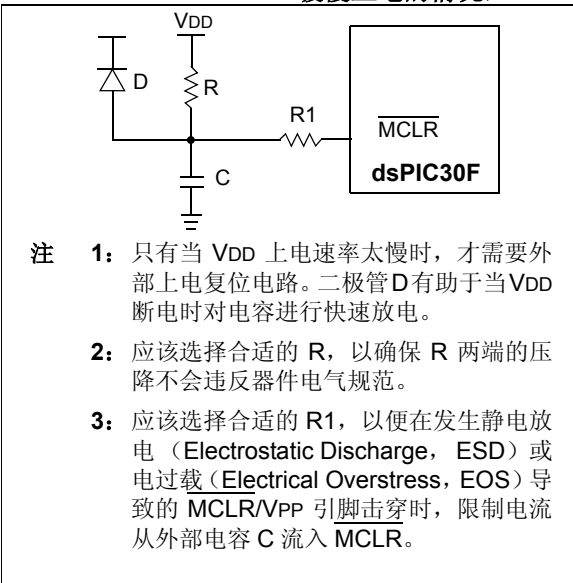
如果 FSCM 使能且出现上述条件之一时, 就将产生时钟故障陷阱。器件将自动切换至 FRC 振荡器, 而用户可在陷阱 ISR 中切换至期望的晶体振荡器。

18.7.1.2 FSCM 和 PWRT 禁止时的的工作

如果 FSCM 禁止、且上电延时定时器 (PWRT) 也禁止的话, 上电时器件将快速退出复位状态。如果时钟源是 FRC 或 EC, 它将立即激活。

如果 FSCM 禁止而系统时钟还没有启动, 器件将在复位向量处将处于冻结状态, 直至系统时钟启动为止。从用户的角度看, 器件看上去仍处于复位状态, 直至系统时钟可用为止。

图 18-11: 外部上电复位电路 (对于 V_{DD} 缓慢上电的情况)



- 注
- 1: 只有当 V_{DD} 上电速率太慢时, 才需要外部上电复位电路。二极管 D 有助于当 V_{DD} 断电时对电容进行快速放电。
 - 2: 应该选择合适的 R , 以确保 R 两端的压降不会违反器件电气规范。
 - 3: 应该选择合适的 $R1$, 以便在发生静电放电 (Electrostatic Discharge, ESD) 或电过载 (Electrical Overstress, EOS) 导致的 $\overline{\text{MCLR}}/\text{VPP}$ 引脚击穿时, 限制电流从外部电容 C 流入 $\overline{\text{MCLR}}$ 。

注: 专用监视器件, 如 MCP1XX 和 MCP8XX 也可用作外部上电复位电路。

表 18-3 所示为 RCON 寄存器的复位状态。由于 RCON 寄存器中的控制位都是可读写的，表中的信息意味着：所有位状态是“条件”这一列中指定操作之前的位状态的反码。

表 18-3: RCON 寄存器的初始状态，情形 1

条件	程序计数器	TRAPR	IOPUWR	EXTR	SWR	WDTO	空闲	休眠	POR
上电复位	0x000000	0	0	0	0	0	0	0	1
正常工作期间的 MCLR 复位	0x000000	0	0	1	0	0	0	0	0
正常工作期间的软件复位	0x000000	0	0	0	1	0	0	0	0
休眠模式下的 MCLR 复位	0x000000	0	0	1	0	0	0	1	0
空闲模式下的 MCLR 复位	0x000000	0	0	1	0	0	1	0	0
WDT 超时复位	0x000000	0	0	0	0	1	0	0	0
WDT 唤醒	PC + 2	0	0	0	0	1	0	1	0
被中断从休眠状态唤醒	PC + 2 ⁽¹⁾	0	0	0	0	0	0	1	0
时钟故障陷阱	0x000004	0	0	0	0	0	0	0	0
陷阱复位	0x000000	1	0	0	0	0	0	0	0
非法操作陷阱	0x000000	0	1	0	0	0	0	0	0

注 1: 如果唤醒由允许的中断导致，PC 将装入相应的中断向量。

表 18-4 显示了 RCON 寄存器位状态的第二个示例。在该情形中，并未假定在条件栏指定的操作之前用户已置 1 或清零指定的位。

表 18-4: RCON 寄存器的初始状态，情形 2

条件	程序计数器	TRAPR	IOPUWR	EXTR	SWR	WDTO	空闲	休眠	POR
上电复位	0x000000	0	0	0	0	0	0	0	1
正常工作期间的 MCLR 复位	0x000000	u	u	1	0	0	0	0	u
正常工作期间的软件复位	0x000000	u	u	0	1	0	0	0	u
休眠模式下的 MCLR 复位	0x000000	u	u	1	u	0	0	1	u
空闲模式下的 MCLR 复位	0x000000	u	u	1	u	0	1	0	u
WDT 超时复位	0x000000	u	u	0	0	1	0	0	u
WDT 唤醒	PC + 2	u	u	u	u	1	u	1	u
被中断从休眠状态唤醒	PC + 2 ⁽¹⁾	u	u	u	u	u	u	1	u
时钟故障陷阱	0x000004	u	u	u	u	u	u	u	u
陷阱复位	0x000000	1	u	u	u	u	u	u	u
非法操作复位	0x000000	u	1	u	u	u	u	u	u

图注: u = 不变

注 1: 如果唤醒由允许的中断导致，PC 将装入相应的中断向量。

18.8 看门狗定时器 (WDT)

18.8.1 看门狗定时器工作原理

看门狗定时器 (WDT) 的主要功能是在软件故障时使处理器复位。WDT 是独立运行的定时器，它使用片内 RC 振荡器，不需要外部元件。于是，即使主处理器时钟（例如，晶振）出现故障，WDT 定时器将继续工作。

18.8.2 使能和禁止 WDT

看门狗定时器的使能或禁止，只能通过配置寄存器 FWDTE 中的配置位 (FWDTE) 来进行。

设置 FWDTE = 1，使能看门狗定时器。使能是在对器件进行编程时完成的。缺省情况下，在擦除芯片后，FWDTE 位 = 1。任何能够对 dsPIC30F 器件进行编程的编程器，都能够对这个配置位及其他配置位进行编程。

如果使能，WDT 将递增，直到它溢出或“超时”。WDT 超时将强制器件复位（休眠模式时除外）。要防止 WDT 超时，用户必须使用 CLRWDTE 指令清零看门狗定时器。

如果休眠期间 WDT 超时，将唤醒器件。RCON 寄存器中的 WDTO 位清零，表明 WDT 超时导致唤醒。

设置 FWDTE = 0，允许用户软件通过 SWDTE 控制位 (RCON<5>) 使能 / 禁止看门狗定时器。

18.9 省电模式

通过执行一条特殊的指令 PWRSAV，可以进入两个省电模式。

这两个模式是：休眠模式和空闲模式。

PWRSAV 指令的格式如下：

PWRSAV< parameter>, 其中 “parameter” 定义是空闲模式还是休眠模式。

18.9.1 休眠模式

在休眠模式下，CPU 和外设的时钟关闭。如果使用了片上振荡器，它也被关闭。

休眠期间，故障保护时钟监视器不起作用，因为没有时钟需要监视。不过，如果休眠期间 WDT 工作的话，LPRC 时钟将保持为活动状态。

发生以下任一事件时，处理器将从休眠模式唤醒：

- 任何中断，如果已被允许且满足所需的优先级
- 任何复位 (POR 和 MCLR)
- WDT 超时

由休眠模式唤醒后，处理器将重新启动进入休眠模式之前处于活动状态的时钟。使能时钟切换时，COSC<2:0>位将决定唤醒时所使用的振荡器源。如果时钟切换禁止，则只有一个系统时钟。

注： 如果发生上电复位，则振荡器的选择将基于 OSC<2:0> 和 FOSCSEL<1:0> 配置位。

如果时钟源是振荡器，器件的时钟将一直关闭，直至 OST 超时为止（这表明振荡器已经稳定）。如果使用了 PLL，系统时钟关闭，直至 LOCK = 1（这表明 PLL 已经稳定）。不管是哪种情况，都要加上 TPOR、TLOCK 和 TPWRT 延时。

如果使用了 EC 或 FRC 振荡器，那么要加上 TPOR（~10 μs）。这是从休眠中唤醒时的最小延时。

此外，如果在休眠期间 LP 振荡器处于工作状态，并且 LP 为唤醒时使用的振荡器，那么起振延时等于 TPOR。不会有 PWRT 延时和 OST 定时器延时。为了使从休眠中唤醒的起振延时最小，在进入休眠之前，应该选择这些快速唤醒选项之一。

任何中断，如果被单独允许（通过相应的 IE 位）且优先级占先的话，将唤醒处理器。处理器将处理中断，转移至 ISR。唤醒时 RCON 寄存器中的休眠状态位将置 1。

注： 尽管使用了各种不同的延时 (TPOR、TLOCK 和 TPWRT)，在延时结束时晶振（和 PLL）可能并未处于工作状态（例如，对于低频晶振）。在这种情况下，如果 FSCM 被使能，则器件将此情况检测为时钟故障并将处理时钟故障陷阱，FRC 振荡器将被使能而用户必须重新使能晶振。如果 FSCM 未使能，器件只是暂停代码执行直至时钟稳定，并且将在振荡器时钟启动前继续处于休眠状态。

所有复位将使处理器从休眠模式唤醒。除 POR 之外的任何复位都将置 1 休眠状态位。在 POR 时，休眠状态位将被清零。

如果看门狗定时器被使能，则处理器将在发生 WDT 超时时从休眠模式唤醒。休眠和 WDTO 状态位都将被置 1。

18.9.2 空闲模式

在空闲模式下，CPU 的时钟关闭，而外设继续工作。与休眠模式不同的是，时钟源继续保持为工作状态。

对于一些外设而言，每个模块中有一个与外设对应的控制位，控制外设空闲期间是否工作。

如果时钟故障检测使能的话，LPRC 故障保护时钟继续保持为工作状态。

在发生以下事件之一时，处理器将从空闲模式唤醒：

- 任何中断，如果已允许（IE 位为 1）并且满足所需的优先级
- 任何复位（POR 或 MCLR）
- WDT 超时

在从空闲模式唤醒时，时钟再次供 CPU 使用且指令立即开始执行，从 PWRSAV 指令之后的第一条指令处开始执行。

任何中断，如果已允许（使用相应的 IE 位）并且优先级占先的话，将唤醒处理器。处理器将处理中断，转移至 ISR。唤醒时 RCON 寄存器中的空闲状态位将置 1。

除上电复位外的任何复位都会将空闲状态位置 1。上电复位时，空闲状态位清零。

如果看门狗定时器使能，则处理器在 WDT 超时时将从空闲模式中唤醒。空闲和 WDTO 状态位都将置 1。

与从休眠中唤醒不同的，从空闲中唤醒时不存在延时。

18.10 器件配置寄存器

每个器件配置寄存器中的配置位都指定某些器件模式，配置位通过器件编程器或者器件的在线串行编程（ICSP）功能来进行编程。虽然每个器件配置寄存器都是 24 位寄存器，但只有低 16 位可用来保存配置数据。有六个器件配置寄存器可供用户使用：

1. FBS（0xF80000）：引导代码段配置寄存器
2. FGS（0xF80004）：通用代码段配置寄存器
3. FGS（0xF80006）：振荡器配置寄存器
4. FGS（0xF80008）：振荡器配置寄存器
5. FWDT（0xF8000A）：看门狗定时器配置寄存器
6. FPOR（0xF8000C）：上电复位配置寄存器

如果在器件编程器中选择了器件，配置位的位置将被自动处理。可以在源代码（这取决于使用的语言工具）中，或者通过编程接口来指定配置位的期望状态。在对器件编程之后，应用软件可以通过表读指令读取配置位的值。更多信息，请参见器件的编程规范。

注： 如果已经编程了代码保护配置控制位（FGS 寄存器中的 GSS<1:0> 和 GWRP），那么只有在电压 VDD ≥ 4.5V 时才可以擦除整个代码保护器件。

表 18-5 给出了 dsPIC30F1010 器件的 FGS 和 FBS 寄存器的位描述。表 18-6 给出了 dsPIC30F202x 器件的 FGS 和 FBS 寄存器的位描述。表 18-7 给出了 dsPIC30F1010/202X 器件的 FWDT 和 FPOR 寄存器的位描述。

dsPIC30F1010/202X

表 18-5: dsPIC30F1010 的 FGS 和 FBS 位描述

位字段	寄存器	说明
BWRP	FBS	引导段程序闪存写保护 1 = 引导段可写 0 = 引导段被写保护
BSS<2:0>	FBS	引导段程序闪存代码保护大小 x11 = 无引导程序闪存段 x00 = 无引导程序闪存段 x01 = 无引导程序闪存段 110 = 标准安全性；小容量引导段；引导程序闪存段开始于中断向量段的末端，结束于 0003FFH 010 = 高安全性；小容量引导段；引导程序闪存段开始于中断向量段的末端，结束于 0003FFH
GRWP	FGS	通用段程序闪存写保护 1 = 通用段可写 0 = 通用段被写保护
GSS<1:0>	FGS	通用段程序闪存代码保护 11 = 无保护 10 = 标准安全性；通用程序闪存段开始于引导段的末端，结束于程序闪存的末端 0x = 保留

表 18-6: dsPIC30F202X 的 FGS 和 FBS 位描述

位字段	寄存器	说明
BWRP	FBS	引导段程序闪存写保护 1 = 引导段可写 0 = 引导段被写保护
BSS<2:0>	FBS	引导段程序闪存代码保护大小 x11 = 无引导程序闪存段 x00 = 无引导程序闪存段 110 = 标准安全性；小容量引导段；引导程序闪存段开始于中断向量段的末端，结束于 0003FFH 010 = 高安全性；小容量引导段；引导程序闪存段开始于中断向量段的末端，结束于 0003FFH 101 = 标准安全性；中等容量引导段；引导程序闪存段开始于中断向量段的末端，结束于 000FFFH 001 = 高安全性；中等容量引导段；引导程序闪存段开始于中断向量段的末端，结束于 000FFFH
GWRP	FGS	通用段程序闪存写保护 1 = 通用段可写 0 = 通用段被写保护
GSS<1:0>	FGS	通用段程序闪存代码保护 11 = 无保护 10 = 标准安全性；通用程序闪存段开始于引导段的末端，结束于程序闪存的末端 0x = 保留

表 18-7: dsPIC30F1010/202X 的 FWDT 和 FPOR 位描述

位字段	寄存器	说明
FWDTEN	FWDT	看门狗定时器使能位 1 = 总是使能看门狗定时器。(不能禁止 LPRC 振荡器。清零 RCON 寄存器中的 SWDTEN 位不产生影响。) 0 = 由用户软件使能 / 禁止看门狗定时器 (通过清零 RCON 寄存器中的 SWDTEN 位可以禁止 LPRC)
WWDTEN	FWDT	看门狗定时器窗口使能位 1 = 看门狗定时器处于无窗口模式 0 = 看门狗定时器处于窗口模式
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

18.11 在线调试器

如果选择 MPLAB® ICD 2 作为调试器, 则使能在线调试功能。与 MPLAB IDE 一起使用时, 在线调试功能允许简单的调试功能。当使能器件的这个功能时, 某些资源不能用于通常的用途。这些资源包括数据 RAM 的前 80 字节和两个 I/O 引脚。

用户使用 MPLAB IDE 中的配置选项, 可选择四对调试 I/O 引脚中的一对。这些引脚为 EMUD/EMUC、EMUD1/EMUC1 和 EMUD2/EMUC2。

无论选取哪对调试引脚, 选取的 EMUD 引脚都是仿真 / 调试数据线, 而 EMUC 引脚则是仿真 / 调试时钟线。这些引脚将与 Microchip 提供的 MPLAB ICD 2 模块接口。MPLAB ICD 2 使用选取的调试 I/O 引脚发送命令和接收响应以及发送和接收数据。要使用器件的在线调试功能, 应用设计必须实现至 MCLR、VDD、VSS、PGC、PGD 以及选取的 EMUDx/EMUCx 引脚对的 ICSP 连接。

这样就产生了两种可能性:

1. 如果选取 EMUD/EMUC 作为调试 I/O 引脚对, 那么只需要一个 5 引脚的接口, 这是因为, 在所有的 dsPIC30F 器件上, EMUD 和 EMUC 引脚功能均可与 PGD 和 PGC 引脚功能复用。
2. 如果选取了 EMUD1/EMUC1 或 EMUD2/EMUC2 作为调试 I/O 引脚对, 那么就需要一个 7 引脚的接口, 这是因为, EMUDx/EMUCx 引脚功能 (x = 1 或 2) 不能与 PGD 和 PGC 引脚功能复用。

表 18-8: dsPIC30F202X 的系统集成寄存器映射

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态	
RCON	0740	TRAPR	IOPUWR	—	—	—	—	—	—	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	—	POR	取决于复位类型	
OSCCON	0742	—	COSC<2:0>			—	NOSC<2:0>			CLKLOCK	—	LOCK	PRCDEN	CF	TSEQEN	—	OSWEN	取决于配置位。	
OSCTUN	0748	TSEQ3<3:0>			TSEQ2<3:0>						TSEQ1<3:0>			TUN<3:0>					0000 0000 0000 0000
OSCTUN2	074A	TSEQ7<3:0>			TSEQ6<3:0>			TSEQ5<3:0>			TSEQ4<3:0>								0000 0000 0000 0000
LFSR	074C	—	LFSR<14:0>														0000 0000 0000 0000		
PMD1	0770	—	—	T3MD	T2MD	T1MD	—	PWMMD	—	I2CMD	—	U1MD	—	SP1MD	—	—	ADCMD	0000 0000 0000 0000	
PMD2	0772	—	—	—	—	—	—	—	IC1MD	—	—	—	—	—	—	OC2MD	OC1MD	0000 0000 0000 0000	
PMD3	0774	—	—	—	—	CMP_PSMD	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000	

注： 有关寄存器位字段的描述，请参见《dsPIC30F 系列参考手册》（DS70046E_CN）。

表 18-9: 器件配置寄存器映射

文件寄存器名称	地址	Bits 23-16	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
FBS	F80000	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BSS<2:0>	—	BWRP
FGS	F80004	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GSS1	GSS0	GWRP
FOSCSSEL	F80006	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FNOSC<1:0>	
FOSC	F80008	—	—	—	—	—	—	—	—	—	—	—	FRANGE	—	—	OSCIOFNC	POSCMD<1:0>	
FWDT	F8000A	—	—	—	—	—	—	—	—	—	—	FCKSM<1:0>	FWDTEN	VWDTEN	—	WDTPRE	WDTPOST<3:0>	
FPOR	F8000C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FPWRT<2:0>	

注： 有关寄存器位字段的描述，请参见《dsPIC30F 系列参考手册》（DS70046E_CN）。

19.0 指令集汇总

注：本数据手册总结了 dsPIC30F 系列器件的功能，但并不作为参考大全使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。有关器件指令集和编程的更多信息，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

dsPIC30F 指令集的众多增强性能扩展了以往的 PIC[®] MCU 指令集，同时可以方便地从 PIC MCU 指令集移植到 dsPIC30F 指令集。

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，分为一个指定指令类型的 8 位操作码和进一步指定指令操作的一个或多个操作数。

指令集是高度正交的，分为五个基本类别：

- 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 19-1 给出了在说明指令时使用的通用符号。

表 19-2 是 dsPIC30F 指令集汇总，还给出了每条指令影响的状态标志位。

大多数针对字或字节的 W 寄存器指令（包括桶形移位指令）有三个操作数：

- 第一个源操作数通常是寄存器 Wb，不带任何地址修改量
- 第二个源操作数通常是寄存器 Ws，带或不带地址修改量
- 运算结果的目的寄存器，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
- 目的寄存器可以是文件寄存器 f 或 W0 寄存器（用 WREG 表示）

大多数位操作类指令（包括简单的循环 / 移位指令）有两个操作数：

- W 寄存器（带或不带地址修改量）或文件寄存器（由 Ws 或 f 的值指定）
- W 寄存器或文件寄存器中的位（由一个立即数指定，或者由 Wb 寄存器的内容间接指定）

涉及数据传送的立即数指令，可以使用下列操作数：

- 要被装入到 W 寄存器或文件寄存器中的立即数（由 k 值指定）
- 要装入立即数的 W 寄存器或文件寄存器（由 Wb 或 f 指定）

然而，涉及算术或逻辑操作的立即数指令，使用如下的操作数：

- 第一个源操作数是寄存器 Wb，不带任何地址修改量
- 第二个源操作数是立即数
- 操作结果的目的寄存器（仅在与第一个源操作数不同时）通常是寄存器 Wd（带或不带地址修改量）

MAC 类 DSP 指令可使用下列操作数：

- 要使用的累加器（A 或 B）（必须的操作数）
- 要用作第二个操作数的 W 寄存器
- X 和 Y 地址空间预取操作
- X 和 Y 地址空间预取目的寄存器
- 累加器回写目的寄存器

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（必须）
- 源操作数或目的操作数（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：

- 程序存储地址
- 表读和表写指令的模式

除了某些双字指令外，所有指令都是单字指令；双字指令之所以是双字长的（48 位），是因为要用 48 位来提供所需信息。在第二个字中，8 个 MSb 全为 0。如果指令自身把第二个字当作一条指令来执行的话，它将作为一条 NOP 指令来执行。

dsPIC30F1010/202X

除非条件测试结果为“true”或者指令执行的结果改变了程序计数器的值，否则执行大部分单字指令都只需要一个指令周期。对于上述两种特殊情况，执行指令需要两个指令周期，第二个指令周期执行一条 NOP 指令。值得注意的是：BRA（无条件 / 计算转移）、间接 CALL/GOTO、所有表读和表写以及 RETURN/RETFIE 指令都是单字指令，但是却需要两个或三个指令周期来执行。对于那些涉及跳过后续指令的某些指令，如果执行了跳

过，则需要两个到三个指令周期，具体取决于被跳过的指令是单字指令还是双字指令。此外，双字传送指令需要两个指令周期。双字指令执行需两个指令周期。

注： 欲知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 19-1: 操作码说明中使用的符号

字段	说明
#text	表示由“text”定义的立即数
(text)	表示“text 的内容”
[text]	表示“地址为 text 的单元”
{ }	可选的位域或操作
<n:m>	寄存器位域
.b	字节模式选择
.d	双字模式选择
.s	影子寄存器选择
.w	字模式选择（默认情况）
Acc	累加器 A 或累加器 B
AWB	累加器回写目的地址寄存器 $\in \{W13, [W13]+2\}$
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	不需要内容，可为空白
OA, OB, SA, SB	DSP 状态位：ACCA 溢出、ACCB 溢出、ACCA 饱和及 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 $\in \{-16...16\}$

表 19-1: 操作码说明中使用的符号 (续)

字段	说明
Wb	基准 W 寄存器 $\in \{W0..W15\}$
Wd	目的 W 寄存器 $\in \{Wd, [Wd], [Wd++] , [Wd--], [++Wd], [--Wd]\}$
Wdo	目的 W 寄存器 $\in \{Wnd, [Wnd], [Wnd++] , [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb]\}$
Wm, Wn	被除数 / 除数工作寄存器对 (直接寻址)
Wm*Wm	平方指令中的被乘数和乘数工作寄存器对 $\in \{W4 * W4, W5 * W5, W6 * W6, W7 * W7\}$
Wm*Wn	DSP 指令中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 个工作寄存器之一 $\in \{W0..W15\}$
Wnd	16 个目的工作寄存器之一 $\in \{W0..W15\}$
Wns	16 个源工作寄存器之一 $\in \{W0..W15\}$
WREG	W0 (文件寄存器指令中使用的工作寄存器)
Ws	源 W 寄存器 $\in \{Ws, [Ws], [Ws++] , [Ws--], [++Ws], [--Ws]\}$
Wso	源 W 寄存器 $\in \{Wns, [Wns], [Wns++] , [Wns--], [++Wns], [--Wns], [Wns+Wb]\}$
Wx	DSP 指令中用于 X 数据空间预取操作的地址寄存器 $\in \{[W8] += 6, [W8] += 4, [W8] += 2, [W8], [W8] -= 6, [W8] -= 4, [W8] -= 2, [W9] += 6, [W9] += 4, [W9] += 2, [W9], [W9] -= 6, [W9] -= 4, [W9] -= 2, [W9 + W12], \text{无}\}$
Wxd	DSP 指令中用于 X 数据空间预取操作的目的寄存器 $\in \{W4..W7\}$
Wy	DSP 指令中用于 Y 数据空间预取操作的地址寄存器 $\in \{[W10] += 6, [W10] += 4, [W10] += 2, [W10], [W10] -= 6, [W10] -= 4, [W10] -= 2, [W11] += 6, [W11] += 4, [W11] += 2, [W11], [W11] -= 6, [W11] -= 4, [W11] -= 2, [W11 + W12], \text{无}\}$
Wyd	DSP 指令中用于 Y 数据空间预取操作的目的寄存器 $\in \{W4..W7\}$

dsPIC30F1010/202X

表 19-2: 指令集汇总

基本指令编号	汇编助记符	汇编语法	说明	指令字数	周期数	受影响的状态标志位
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	$f = \text{算术右移 } f$	1	1	C,N,OV,Z
		ASR f, WREG	$WREG = \text{算术右移 } f$	1	1	C,N,OV,Z
		ASR Ws, Wd	$Wd = \text{算术右移 } Ws$	1	1	C,N,OV,Z
		ASR Wb, Wns, Wnd	$Wnd = \text{将 } Wb \text{ 算术右移 } Wns \text{ 位}$	1	1	N,Z
		ASR Wb, #lit5, Wnd	$Wnd = \text{将 } Wb \text{ 算术右移 } lit5 \text{ 位}$	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	如果进位位为 1 则转移	1	1 (2)	无
		BRA GE, Expr	如果有符号大于或等于则转移	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	如果有符号大于则转移	1	1 (2)	无
		BRA GTU, Expr	如果无符号大于则转移	1	1 (2)	无
		BRA LE, Expr	如果有符号小于或等于则转移	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	如果有符号小于则转移	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	如果进位位为 0 则转移	1	1 (2)	无
		BRA NN, Expr	如果非负则转移	1	1 (2)	无
		BRA NOV, Expr	如果未溢出则转移	1	1 (2)	无
		BRA NZ, Expr	如果非零则转移	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
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

表 19-2: 指令集汇总 (续)

基本指令编号	汇编助记符	汇编语法	说明	指令字数	周期数	受影响的状态标志位
11	BTSS	BTSS $f, \#bit4$	对 f 中的指定位进行测试, 如果为 1 则跳过	1	1 (2 或 3)	无
		BTSS $Ws, \#bit4$	对 Ws 中的指定位进行测试, 如果为 1 则跳过	1	1 (2 或 3)	无
12	BTST	BTST $f, \#bit4$	对 f 中的指定位进行测试	1	1	Z
		BTST.C $Ws, \#bit4$	对 Ws 中的指定位进行测试, 并将其值存储到 C	1	1	C
		BTST.Z $Ws, \#bit4$	对 Ws 中的指定位进行测试, 并将其反码存储到 Z	1	1	Z
		BTST.C Ws, Wb	对 $Ws<Wb>$ 位进行测试, 并将其值存储到 C	1	1	C
		BTST.Z Ws, Wb	对 $Ws<Wb>$ 位进行测试, 并将其反码存储到 Z	1	1	Z
13	BTSTS	BTSTS $f, \#bit4$	对 f 中的指定位进行测试, 并将 f 中的该位置 1	1	1	Z
		BTSTS.C $Ws, \#bit4$	对 Ws 中的指定位进行测试, 并将其值存储到 C, 然后将 Ws 中的该位置 1	1	1	C
		BTSTS.Z $Ws, \#bit4$	对 Ws 中的指定位进行测试, 并将其反码存储到 Z, 然后将 Ws 中的该位置 1	1	1	Z
14	CALL	CALL $lit23$	调用子程序	2	2	无
		CALL Wn	间接调用子程序	1	2	无
15	CLR	CLR f	$f = 0x0000$	1	1	无
		CLR WREG	WREG = 0x0000	1	1	无
		CLR Ws	$Ws = 0x0000$	1	1	无
		CLR Acc, Wx, Wxd, Wy, W yd, AWB	将累加器清零	1	1	OA,OB,SA,SB
16	CLRWD	CLRWD	将看门狗定时器清零	1	1	WDTO,Sleep
17	COM	COM f	$f = f$	1	1	N,Z
		COM $f, WREG$	WREG = f	1	1	N,Z
		COM Ws, Wd	$Wd = Ws$	1	1	N,Z
18	CP	CP f	比较 f 和 WREG	1	1	C,DC,N,OV,Z
		CP $Wb, \#lit5$	比较 Wb 和 $lit5$	1	1	C,DC,N,OV,Z
		CP Wb, Ws	比较 Wb 和 $Ws(Wb - Ws)$	1	1	C,DC,N,OV,Z
19	CP0	CP0 f	比较 f 和 0x0000	1	1	C,DC,N,OV,Z
		CP0 Ws	比较 Ws 和 0x0000	1	1	C,DC,N,OV,Z
20	CPB	CPB f	带借位比较 f 和 WREG	1	1	C,DC,N,OV,Z
		CPB $Wb, \#lit5$	带借位比较 Wb 和 $lit5$	1	1	C,DC,N,OV,Z
		CPB Wb, Ws	带借位比较 Wb 和 $Ws(Wb - Ws - \overline{C})$	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
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 $Wm*Wm, Acc, Wx, Wy$ $, Wxd$	欧几里德距离的内容	1	1	OA,OB,OAB, SA,SB,SAB

dsPIC30F1010/202X

表 19-2: 指令集汇总 (续)

基本指令编号	汇编助记符	汇编语法	说明	指令字数	周期数	受影响的状态标志位
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 (MSb) 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	分配帧指针	1	1	无
44	LSR	LSR f	$f = \text{逻辑右移 } f$	1	1	C, N, OV, Z
		LSR f, WREG	$WREG = \text{逻辑右移 } f$	1	1	C, N, OV, Z
		LSR Ws, Wd	$Wd = \text{逻辑右移 } Ws$	1	1	C, N, OV, Z
		LSR Wb, Wns, Wnd	$Wnd = \text{将 } Wb \text{ 逻辑右移 } Wns \text{ 位}$	1	1	N, Z
		LSR Wb, #lit5, Wnd	$Wnd = \text{将 } Wb \text{ 逻辑右移 } lit5 \text{ 位}$	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 中的内容送入目的寄存器	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	MOV.SAC	MOV.SAC Ws, Wnd	将 Ws 中的双字内容送入 W(nd + 1):W(nd)	1	2	无
		MOV.SAC Acc, Wx, Wxd, Wy, Wyd, AWB	预取操作数并保存累加器	1	1	无
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 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	$\{Wnd+1, Wnd\} = \text{signed}(Wb) * \text{signed}(Ws)$	1	1	无
		MUL.SU Wb, Ws, Wnd	$\{Wnd+1, Wnd\} = \text{signed}(Wb) * \text{unsigned}(Ws)$	1	1	无
		MUL.US Wb, Ws, Wnd	$\{Wnd+1, Wnd\} = \text{unsigned}(Wb) * \text{signed}(Ws)$	1	1	无
		MUL.UU Wb, Ws, Wnd	$\{Wnd+1, Wnd\} = \text{unsigned}(Wb) * \text{unsigned}(Ws)$	1	1	无
		MUL.SU Wb, #lit5, Wnd	$\{Wnd+1, Wnd\} = \text{signed}(Wb) * \text{unsigned}(lit5)$	1	1	无
		MUL.UU Wb, #lit5, Wnd	$\{Wnd+1, Wnd\} = \text{unsigned}(Wb) * \text{unsigned}(lit5)$	1	1	无
		MUL f	$W3:W2 = f * WREG$	1	1	无

表 19-2: 指令集汇总 (续)

基本指令编号	汇编助记符	汇编语法	说明	指令字数	周期数	受影响的状态标志位
52	NEG	NEG <i>Acc</i>	将累加器内容求补	1	1	OA,OB,OAB,SA,SB,SAB
		NEG <i>f</i>	$f = f + 1$	1	1	C,DC,N,OV,Z
		NEG <i>f</i> , WREG	WREG = $f + 1$	1	1	C,DC,N,OV,Z
		NEG <i>Ws</i> , <i>Wd</i>	$Wd = Ws + 1$	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 <i>f</i>	将栈顶 (TOS) 的内容弹出到 <i>f</i>	1	1	无
		POP <i>Wdo</i>	将栈顶 (TOS) 的内容弹出到 <i>Wdo</i>	1	1	无
		POP.D <i>Wnd</i>	将栈顶 (TOS) 的内容弹出到 <i>W(nd):W(nd+1)</i>	1	2	无
		POP.S	将影子寄存器的内容弹出到主寄存器	1	1	复位
55	PUSH	PUSH <i>f</i>	将 <i>f</i> 的内容压入栈顶 (TOS)	1	1	无
		PUSH <i>Wso</i>	将 <i>Wso</i> 的内容压入栈顶 (TOS)	1	1	无
		PUSH.D <i>Wns</i>	将 <i>W(ns):W(ns + 1)</i> 中的双字压入栈顶 (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	PWRSV	PWRSV <i>#lit1</i>	进入休眠或空闲模式	1	1	WDTO,Sleep
57	RCALL	RCALL <i>Expr</i>	相对调用	1	2	无
		RCALL <i>Wn</i>	计算调用	1	2	无
58	REPEAT	REPEAT <i>#lit14</i>	将下一条指令重复执行 <i>lit14 + 1</i> 次	1	1	无
		REPEAT <i>Wn</i>	将下一条指令重复执行 (<i>Wn</i>) + 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 <i>#lit10</i> , <i>Wn</i>	返回并将立即数存入 <i>Wn</i>	1	3 (2)	无
62	RETURN	RETURN	从子程序返回	1	3 (2)	无
63	RLC	RLC <i>f</i>	$f =$ 对 <i>f</i> 执行带进位的循环左移	1	1	C,N,Z
		RLC <i>f</i> , WREG	WREG = 对 <i>f</i> 执行带进位的循环左移	1	1	C,N,Z
		RLC <i>Ws</i> , <i>Wd</i>	$Wd =$ 对 <i>Ws</i> 执行带进位的循环左移	1	1	C,N,Z
64	RLNC	RLNC <i>f</i>	$f =$ 循环左移 <i>f</i> (不带进位)	1	1	N,Z
		RLNC <i>f</i> , WREG	WREG = 循环左移 <i>f</i> (不带进位)	1	1	N,Z
		RLNC <i>Ws</i> , <i>Wd</i>	$Wd =$ 循环左移 <i>Ws</i> (不带进位)	1	1	N,Z
65	RRC	RRC <i>f</i>	$f =$ 对 <i>f</i> 执行带进位的循环右移	1	1	C,N,Z
		RRC <i>f</i> , WREG	WREG = 对 <i>f</i> 执行带进位的循环右移	1	1	C,N,Z
		RRC <i>Ws</i> , <i>Wd</i>	$Wd =$ 对 <i>Ws</i> 执行带进位的循环右移	1	1	C,N,Z
66	RRNC	RRNC <i>f</i>	$f =$ 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>	$Wd =$ 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>	$Wnd =$ 对 <i>Ws</i> 进行符号扩展	1	1	C,N,Z
69	SETM	SETM <i>f</i>	$f = 0xFFFF$	1	1	无
		SETM WREG	WREG = $0xFFFF$	1	1	无
		SETM <i>Ws</i>	$Ws = 0xFFFF$	1	1	无
70	SFTAC	SFTAC <i>Acc</i> , <i>Wn</i>	对累加器算术移位 (<i>Wn</i>) 次	1	1	OA,OB,OAB,SA,SB,SAB
		SFTAC <i>Acc</i> , <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>	$f =$ 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>	$Wd =$ 左移 <i>Ws</i>	1	1	C,N,OV,Z
		SL <i>Wb</i> , <i>Wns</i> , <i>Wnd</i>	$Wnd =$ 将 <i>Wb</i> 左移 <i>Wns</i> 位	1	1	N,Z
		SL <i>Wb</i> , <i>#lit5</i> , <i>Wnd</i>	$Wnd =$ 将 <i>Wb</i> 左移 <i>lit5</i> 位	1	1	N,Z

dsPIC30F1010/202X

表 19-2: 指令集汇总 (续)

基本指令编号	汇编助记符	汇编语法	说明	指令字数	周期数	受影响的状态标志位
72	SUB	SUB <i>Acc</i>	从累加器减去	1	1	OA,OB,OAB,SA,SB,SAB
		SUB <i>f</i>	$f = f - \text{WREG}$	1	1	C,DC,N,OV,Z
		SUB <i>f</i> , WREG	$\text{WREG} = f - \text{WREG}$	1	1	C,DC,N,OV,Z
		SUB #lit10, Wn	$\text{Wn} = \text{Wn} - \text{lit10}$	1	1	C,DC,N,OV,Z
		SUB Wb, Ws, Wd	$\text{Wd} = \text{Wb} - \text{Ws}$	1	1	C,DC,N,OV,Z
		SUB Wb, #lit5, Wd	$\text{Wd} = \text{Wb} - \text{lit5}$	1	1	C,DC,N,OV,Z
73	SUBB	SUBB <i>f</i>	$f = f - \text{WREG} - (\text{C})$	1	1	C,DC,N,OV,Z
		SUBB <i>f</i> , WREG	$\text{WREG} = f - \text{WREG} - (\text{C})$	1	1	C,DC,N,OV,Z
		SUBB #lit10, Wn	$\text{Wn} = \text{Wn} - \text{lit10} - (\text{C})$	1	1	C,DC,N,OV,Z
		SUBB Wb, Ws, Wd	$\text{Wd} = \text{Wb} - \text{Ws} - (\text{C})$	1	1	C,DC,N,OV,Z
		SUBB Wb, #lit5, Wd	$\text{Wd} = \text{Wb} - \text{lit5} - (\text{C})$	1	1	C,DC,N,OV,Z
74	SUBR	SUBR <i>f</i>	$f = \text{WREG} - f$	1	1	C,DC,N,OV,Z
		SUBR <i>f</i> , WREG	$\text{WREG} = \text{WREG} - f$	1	1	C,DC,N,OV,Z
		SUBR Wb, Ws, Wd	$\text{Wd} = \text{Ws} - \text{Wb}$	1	1	C,DC,N,OV,Z
		SUBR Wb, #lit5, Wd	$\text{Wd} = \text{lit5} - \text{Wb}$	1	1	C,DC,N,OV,Z
75	SUBBR	SUBBR <i>f</i>	$f = \text{WREG} - f - (\text{C})$	1	1	C,DC,N,OV,Z
		SUBBR <i>f</i> , WREG	$\text{WREG} = \text{WREG} - f - (\text{C})$	1	1	C,DC,N,OV,Z
		SUBBR Wb, Ws, Wd	$\text{Wd} = \text{Ws} - \text{Wb} - (\text{C})$	1	1	C,DC,N,OV,Z
		SUBBR Wb, #lit5, Wd	$\text{Wd} = \text{lit5} - \text{Wb} - (\text{C})$	1	1	C,DC,N,OV,Z
76	SWAP	SWAP.b Wn	Wn = 半字节交换 Wn 内容	1	1	无
		SWAP Wn	Wn = 将 Wn 的两个字节相交换	1	1	无
77	TBLRDH	TBLRDH Ws, Wd	将程序存储单元的 <23:16> 读入 Wd<7:0>	1	2	无
78	TBLRDL	TBLRDL Ws, Wd	将程序存储单元的 <15:0> 读入 Wd	1	2	无
79	TBLWTH	TBLWTH Ws, Wd	将 Ws<7:0> 写入程序存储单元的 <23:16>	1	2	无
80	TBLWTL	TBLWTL Ws, Wd	将 Ws 写入程序存储单元的 <15:0>	1	2	无
81	ULNK	ULNK	释放帧指针	1	1	无
82	XOR	XOR <i>f</i>	$f = f .\text{XOR} .\text{WREG}$	1	1	N,Z
		XOR <i>f</i> , WREG	$\text{WREG} = f .\text{XOR} .\text{WREG}$	1	1	N,Z
		XOR #lit10, Wn	$\text{Wd} = \text{lit10} .\text{XOR} .\text{Wd}$	1	1	N,Z
		XOR Wb, Ws, Wd	$\text{Wd} = \text{Wb} .\text{XOR} .\text{Ws}$	1	1	N,Z
		XOR Wb, #lit5, Wd	$\text{Wd} = \text{Wb} .\text{XOR} .\text{lit5}$	1	1	N,Z
83	ZE	ZE Ws, Wnd	Wnd = 对 Ws 进行零扩展	1	1	C,Z,N

20.0 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

20.1 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。

20.2 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

20.3 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。

20.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器包含了由 MPASM 汇编器、MPLAB C18 C 编译器产生的可重定位目标。通过使用链接器脚本中的指令，它还可链接预编译库中的可重定位目标。

MPLIB 目标库管理器管理预编译代码库文件的创建和修改。当从源文件调用库中的一段子程序时，只有包含此子程序的模块被链接到应用中。这样可使大型库在许多不同应用中被高效地利用。

目标链接器 / 库管理器具有如下特征：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

20.5 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

20.6 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。

20.7 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。

20.8 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® 和 dsPIC® 闪存单片机进行调试和编程。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。

20.9 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。

20.10 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。

20.11 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。

20.12 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。

20.13 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、SEEEVAL® 评估系统、 $\Sigma\Delta$ ADC、流速传感器，等等。

有关演示、开发和评估工具包的完整列表，请查阅 Microchip 公司网页 (www.microchip.com) 以及最新的 “*Product Selector Guide (产品选型指南)*” (DS00148)。

21.0 电气特性

本节将对 dsPIC30F 系列器件的电气特性进行概括介绍。其余信息将在该文档的后续版本中给出。

有关 dsPIC30F 架构和内核的详细信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

下面列出了该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可能影响其可靠性。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^(†)

环境温度.....	-40°C 至 +125°C
存储温度.....	-65°C 至 +150°C
任一引脚相对于 V _{SS} 的电压（除 V _{DD} 和 $\overline{\text{MCLR}}$ 外） ⁽¹⁾	-0.3V 至 (V _{DD} + 0.3V)
V _{DD} 引脚相对于 V _{SS} 的电压.....	-0.3V 至 +5.5V
$\overline{\text{MCLR}}$ 引脚相对于 V _{SS} 的电压 ⁽¹⁾	0.3 至 (V _{DD} + 0.3V)
V _{SS} 引脚的最大电流.....	300 mA
V _{DD} 引脚的最大电流 ⁽²⁾	300 mA
输入箝位电流，I _{IK} (V _I < 0 或 V _I > V _{DD}).....	±20 mA
输出箝位电流，I _{OK} (V _O < 0 或 V _O > V _{DD}).....	±20 mA
任一 I/O 引脚最大输出灌电流.....	25 mA
任一 I/O 引脚最大输出拉电流.....	25 mA
全部端口总的灌电流.....	200 mA
全部端口总的拉电流 ⁽²⁾	200 mA

注 1: $\overline{\text{MCLR}}/\text{VPP}$ 引脚上的尖峰电压低于 V_{SS}，导致感应电流超过 80 mA 时，可能引起锁死。“低”电平加到 $\overline{\text{MCLR}}/\text{VPP}$ 引脚上时，应串联一个 50-100Ω 的电阻，而不要将这个引脚直接接到 V_{SS}。

2: 最大可允许电流由器件最大功耗决定，见表 21-2。

[†]注：如果器件工作条件超过上述“绝对最大额定值”，可能引起器件永久性损坏。这仅是极限参数，我们不建议器件工作在极限值甚至超过上述极限值。器件长时间工作在极限条件下可能会影响其可靠性。

21.1 DC 特性

表 21-1: 工作 MIPS 与电压

V _{DD} 范围	温度范围	最大 MIPS	
		dsPIC30FXXX-30I	dsPIC30FXXX-20E
4.5-5.5V	-40°C 至 85°C	30	—
4.5-5.5V	-40 °C 至 125 °C	—	20
3.0-3.6V	-40°C 至 85°C	20	—
3.0-3.6V	-40 °C 至 125 °C	—	15

dsPIC30F1010/202X

表 21-2: 热工作条件

额定值	符号	最小值	典型值	最大值	单位
dsPIC30F1010/202X-30I					
工作结温范围	T _J	-40		+125	°C
工作环境温度范围	T _A	-40		+85	°C
dsPIC30F1010/202X-20E					
工作结温范围	T _J	-40		+150	°C
工作环境温度范围	T _A	-40		+125	°C
功耗: 内部芯片功耗: $P_{INT} = V_{DD} \times (I_{DD} - \sum I_{OH})$ I/O 引脚功耗: $P_{I/O} = \sum (\{V_{DD} - V_{OH}\} \times I_{OH}) + \sum (V_{OL} \times I_{OL})$	P _D	P _{INT} + P _{I/O}			W
最大允许功耗	P _D MAX	(T _J - T _A) / θ _{JA}			W

表 21-3: 热封装特性

特性	符号	典型值	最大值	单位	注
封装热阻, 28 引脚 SOIC (SO)	θ _{JA}	48.3		°C/W	1, 2
封装热阻, 28 引脚 QFN	θ _{JA}	33.7		°C/W	1, 2
封装热阻, 28 引脚 SPDIP (SP)	θ _{JA}	42		°C/W	1, 2
封装热阻, 44 引脚 QFN	θ _{JA}	28		°C/W	1, 2
封装热阻, 44 引脚 TQFP	θ _{JA}	39.3		°C/W	1, 2

- 注 1: 通过封装模拟获得结点与环境的热阻值 θ_{JA}。
 2: 根据具体的工作条件, 可能要求通风以改善热性能。

表 21-4: DC 温度和电压规范

DC 特性		标准工作条件: 3.3V 和 5.0V (±10%) (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 ⁽¹⁾	最大值	单位	条件
工作温度 ⁽²⁾							
DC10	V _{DD}	电源电压	3.0	—	5.5	V	工业级温度
DC11	V _{DD}	电源电压	3.0	—	5.5	V	扩展级温度
DC12	V _{DR}	RAM 数据保持电压 ⁽³⁾	—	1.5	—	V	
DC16	V _{POR}	确保能产生内部上电复位信号的 V _{DD} 启动电压	—	V _{SS}	—	V	
DC17	S _{VDD}	确保能产生内部上电复位信号的 V _{DD} 上升率	0.05			V/ms	0.1 s 内可上升 0-5V 60 ms 内可上升 0-3.3V

- 注 1: 除非另外声明, 否则“典型值”一列中的数据都是在 5V、25 °C 的条件下给出的。这些参数仅供设计参考, 未经测试。
 2: 这些参数为特性值, 但生产时未经测试。
 3: 这是在不丢失 RAM 数据的前提下, V_{DD} 的下限值。

表 21-5: DC 特性: 工作电流 (I_{DD})

DC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (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⁾					
DC20a	13	16	mA	25°C	3.3V FRC 3.2 MIPS, 禁止 PLL
DC20b	14	16	mA	85°C	
DC20c	14	17	mA	125°C	
DC20d	22	26	mA	25°C	
DC20e	22	26	mA	85°C	
DC20f	22	27	mA	125°C	
DC22a	19	22	mA	25°C	3.3V FRC 4.9 MIPS, 禁止 PLL
DC22b	19	23	mA	85°C	
DC22c	19	23	mA	125°C	
DC22d	30	36	mA	25°C	
DC22e	30	37	mA	85°C	
DC22f	31	37	mA	125°C	
DC23a	27	33	mA	25°C	3.3V FRC 7.3 MIPS, 禁止 PLL
DC23b	28	33	mA	85°C	
DC23c	28	34	mA	125°C	
DC23d	44	53	mA	25°C	
DC23e	45	53	mA	85°C	
DC23f	45	54	mA	125°C	
DC24a	66	79	mA	25°C	3.3V FRC 13 MIPS, 使能 PLL
DC24b	67	80	mA	85°C	
DC24c	68	81	mA	125°C	
DC24d	108	129	mA	25°C	
DC24e	109	130	mA	85°C	
DC24f	110	131	mA	125°C	
DC26a	98	118	mA	25°C	3.3V FRC 20 MIPS, 使能 PLL
DC26b	99	118	mA	85°C	
DC26d	159	191	mA	25°C	
DC26e	160	192	mA	85°C	
DC26f	161	193	mA	125°C	
DC27d	222	267	mA	25°C	
DC27e	223	267	mA	85°C	5V FRC 30 MIPS, 使能 PLL

注 1: 除非另外声明, 否则“典型值”一列中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

2: 供电电流主要受工作电压和频率的影响。其他因素如 I/O 引脚负载和开关频率、振荡器类型、内部代码执行模式和温度也对电流消耗有影响。所有 I_{DD} 测量的测试条件为:

- 所有 I/O 引脚配置为输入且被拉到 V_{SS}。
- $MCLR = V_{DD}$, WDT 和 FSCM 被禁止。
- CPU、SRAM、程序存储器和数据存储器处于工作状态。
- 外设模块均不工作。

dsPIC30F1010/202X

表 21-5: DC 特性: 工作电流 (IDD) (续)

DC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (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)					
DC28a	96	116	mA	25°C	3.3V EC 20 MIPS, 使能 PLL
DC28b	97	116	mA	85°C	
DC28d	157	188	mA	25°C	
DC28e	158	189	mA	85°C	
DE28f	159	191	mA	125°C	5V EC 30 MIPS, 使能 PLL
DC29d	227	273	mA	25°C	
DC29e	228	273	mA	85°C	

- 注 1: 除非另外声明, 否则 “典型值” 一系列中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。
- 2: 供电电流主要受工作电压和频率的影响。其他因素如 I/O 引脚负载和开关频率、振荡器类型、内部代码执行模式和温度也对电流消耗有影响。所有 IDD 测量的测试条件为:
- 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、程序存储器和数据存储器处于工作状态。
 - 外设模块均不工作。

表 21-6: DC 特性: 空闲电流 (IDLE)

DC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (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): 内核不工作且时钟工作时的基本电流 ⁽²⁾					
DC40a	8	9	mA	25°C	3.3V FRC 3.2 MIPS, 禁止 PLL
DC40b	8	9	mA	85°C	
DC40c	8	10	mA	125°C	
DC40d	12	15	mA	25°C	
DC40e	13	15	mA	85°C	
DC40f	13	16	mA	125°C	
DC42a	10	12	mA	25°C	3.3V FRC 4.9 MIPS, 禁止 PLL
DC42b	11	13	mA	85°C	
DC42c	11	13	mA	125°C	
DC42d	17	20	mA	25°C	
DC42e	17	21	mA	85°C	
DC42f	18	21	mA	125°C	
DC43a	15	18	mA	25°C	3.3V FRC 7.3 MIPS, 禁止 PLL
DC43b	15	18	mA	85°C	
DC43c	15	18	mA	125°C	
DC43d	24	29	mA	25°C	
DC43e	24	29	mA	85°C	
DC43f	25	30	mA	125°C	
DC44a	44	53	mA	25°C	3.3V FRC 13 MIPS, 使能 PLL
DC44b	45	54	mA	85°C	
DC44c	46	55	mA	125°C	
DC44d	72	87	mA	25°C	
DC44e	73	88	mA	85°C	
DC44f	74	89	mA	25°C	
DC46a	66	79	mA	85°C	3.3V FRC 20 MIPS, 使能 PLL
DC46b	67	80	mA	125°C	
DC46d	108	129	mA	25°C	
DC46e	109	131	mA	85°C	
DC45f	110	132	mA	25°C	
DC47d	152	182	mA	85°C	
DC47e	153	183	mA	125°C	5V FRC 30 MIPS, 使能 PLL

注 1: 除非另外声明, 否则“典型值”一列中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

2: 基本 IDLE 电流测量是在内核不工作、时钟工作而所有模块都关闭的条件下进行的。所有 I/O 都配置为输入且被拉至高电平。WDT 等都关闭。

dsPIC30F1010/202X

表 21-6: DC 特性: 空闲电流 (I_{IDLE}) (续)

DC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (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): 内核不工作且时钟工作时的基本电流 ⁽²⁾					
DC48a	65	78	mA	25°C	3.3V EC 20 MIPS, 使能 PLL
DC48b	66	79	mA	85°C	
DC48d	105	127	mA	125°C	
DC48e	107	128	mA	25°C	
DC48f	108	130	mA	85°C	5V EC 30 MIPS, 使能 PLL
DC49d	155	186	mA	125°C	
DC49e	156	187	mA	25°C	

- 注 1: 除非另外声明, 否则 “典型值” 一行中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。
- 2: 基本 I_{IDLE} 电流测量是在内核不工作、时钟工作而所有模块都关闭的条件下进行的。所有 I/O 都配置为输入且被拉至高电平。WDT 等都关闭。

表 21-7: DC 特性: 掉电电流 (IPD)

DC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (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)							
DC60a	1.2	2.4	mA	25°C	3.3V	基本掉电电流 ⁽²⁾	
DC60b	1.2	2.4	mA	85°C			
DC60c	1.3	2.6	mA	125°C			
DC60e	2.1	4.2	mA	25°C	5V		
DC60f	2.1	4.2	mA	85°C			
DC60g	2.3	4.6	mA	125°C			
DC61a	15	30	μA	25°C	3.3V	看门狗定时器电流: ΔI _{WDT} ⁽³⁾	
DC61b	14	30	μA	85°C			
DC61c	14	30	μA	125°C			
DC61e	30	60	μA	25°C	5V		
DC61f	29	60	μA	85°C			
DC61g	30	60	μA	125°C			

- 注 1: 除非另外声明, 否则“典型值”一列中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。
- 2: 基本 IPD 是在所有外设和时钟都关断的条件下进行测量的。所有 I/O 都配置为输入且被拉至高电平。WDT 等都关闭。
- 3: Δ 电流为当模块使能时额外消耗的电流。此电流应被加到基本 IPD 电流。

dsPIC30F1010/202X

表 21-8: DC 特性: I/O 引脚输入规范

DC 特性			标准工作条件: 3.3V 和 5.0V ($\pm 10\%$) (除非另外声明) 工作温度 -40°C $\leq$ TA $\leq$ +85°C (工业级) -40°C $\leq$ TA $\leq$ +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}	输入低电平电压 ⁽²⁾ I/O 引脚: 带施密特触发缓冲器	V _{SS}	—	0.2 V _{DD}	V	禁止 SM 总线 使能 SM 总线
DI15		MCLR	V _{SS}	—	0.2 V _{DD}	V	
DI16		OSC1 (在 HS 模式下)	V _{SS}	—	0.2 V _{DD}	V	
DI18		SDA 和 SCL	V _{SS}	—	0.3 V _{DD}	V	
DI19		SDA 和 SCL	V _{SS}	—	0.2 V _{DD}	V	
DI20	V _{IH}	输入高电平电压 ⁽²⁾ I/O 引脚: 带施密特触发缓冲器	0.8 V _{DD}	—	V _{DD}	V	禁止 SM 总线 使能 SM 总线
DI25		MCLR	0.8 V _{DD}	—	V _{DD}	V	
DI26		OSC1 (在 HS 模式下)	0.7 V _{DD}	—	V _{DD}	V	
DI28		SDA 和 SCL	0.7 V _{DD}	—	V _{DD}	V	
DI29		SDA 和 SCL	0.8 V _{DD}	—	V _{DD}	V	
DI50	I _{IL}	输入泄漏电流 ⁽²⁾⁽³⁾⁽⁴⁾ I/O 端口	—	0.01	± 1	μ A	V _{SS} $\leq$ V _{PIN} $\leq$ V _{DD} , 引脚处于高阻态
DI51		模拟输入引脚	—	0.50	—	μ A	V _{SS} $\leq$ V _{PIN} $\leq$ V _{DD} , 引脚处于高阻态
DI55		MCLR	—	0.05	± 5	μ A	V _{SS} $\leq$ V _{PIN} $\leq$ V _{DD}
DI56		OSC1	—	0.05	± 5	μ A	V _{SS} $\leq$ V _{PIN} $\leq$ V _{DD} , HS 振荡模式

- 注 1: 除非另外声明, 否则“典型值”一列中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。
- 2: 这些参数为特性值, 但生产时未经测试。
- 3: MCLR 引脚上的泄漏电流主要取决于所加电平。表中给定的电平表示正常工作条件。在不同输入电压条件下可能测得更高的泄漏电流。
- 4: 负电流定义为引脚的拉电流。

表 21-9: DC 特性: I/O 引脚输出规范

DC 特性			标准工作条件: 3.3V 和 5.0V ($\pm 10\%$) (除非另外声明) 工作温度 -40°C $\leq$ Ta $\leq$ +85°C (工业级) -40°C $\leq$ Ta $\leq$ +125°C (扩展级)				
参数编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
DO10	VOL	输出低电平电压 ⁽²⁾ I/O 端口	—	—	0.6	V	IOL = 8.5 mA, VDD = 5V IOL = 2.0 mA, VDD = 3.3V
DO16		OSC2/CLKO (RC 或 EC 振荡模式)	—	—	TBD	V	
			—	—	0.6	V	
			—	—	TBD	V	
DO20	VOH	输出高电平电压 ⁽²⁾ I/O 端口	VDD - 0.7	—	—	V	IOH = -3.0 mA, VDD = 5V IOH = -2.0 mA, VDD = 3.3V
			TBD	—	—	V	
DO26		OSC2/CLKO (RC 或 EC 振荡模式)	VDD - 0.7	—	—	V	
			TBD	—	—	V	
DO50	Cosc2	输出引脚的容性负载规范 ⁽²⁾ OSC2 引脚	—	—	15	pF	在 HS 模式下, 使用外部时钟驱动 OSC1 时。 RC 或 EC 振荡模式 在 I ² C 模式下
DO56	CIO	所有 I/O 引脚和 OSC2	—	—	50	pF	
DO58	CB	SCL 和 SDA	—	—	400	pF	

图注: TBD = 待定

注 1: 除非另外声明, 否则“典型值”一列中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

2: 这些参数为特性值, 但生产时未经测试。

表 21-10: DC 特性: 程序存储器和 EEPROM

DC 特性			标准工作条件: 3.3V 和 5.0V ($\pm 10\%$) (除非另外声明) 工作温度 -40°C $\leq$ Ta $\leq$ +85°C (工业级) -40°C $\leq$ Ta $\leq$ +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 ⁽²⁾ 单元耐擦写能力	10K	100K	—	E/W	VMIN = 最小工作电压
D131	VPR	读操作时的 VDD	VMIN	—	5.5	V	
D132	VEB	块擦除时的 VDD	4.5	—	5.5	V	
D133	VPEW	擦 / 写操作时的 VDD	3.0	—	5.5	V	
D134	TPEW	擦 / 写周期时间	—	2	—	ms	假定未违反其他规范
D135	TRETD	特性保持时间	40	100	—	年	
D136	TEB	ICSP 块擦除时间	—	4	—	ms	
D137	IPEW	编程时的 IDD	—	10	30	mA	
D138	IEB	编程时的 IDD	—	10	30	mA	

注 1: 除非另外声明, 否则“典型值”一列中的数据都是在 5V、25°C 的条件下给出的。

2: 这些参数为特性值, 但生产时未经测试。

21.2 AC 特性和时序参数

本节包含的信息说明了 dsPIC30F 系列器件的 AC 特性和时序参数。

表 21-11: 温度和电压规范——AC

AC 特性	标准工作条件: 3.3V 和 5.0V (±10%) (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 DC 规范第 21.0 节中所述。	

图 21-1: 器件时序规范的负载条件

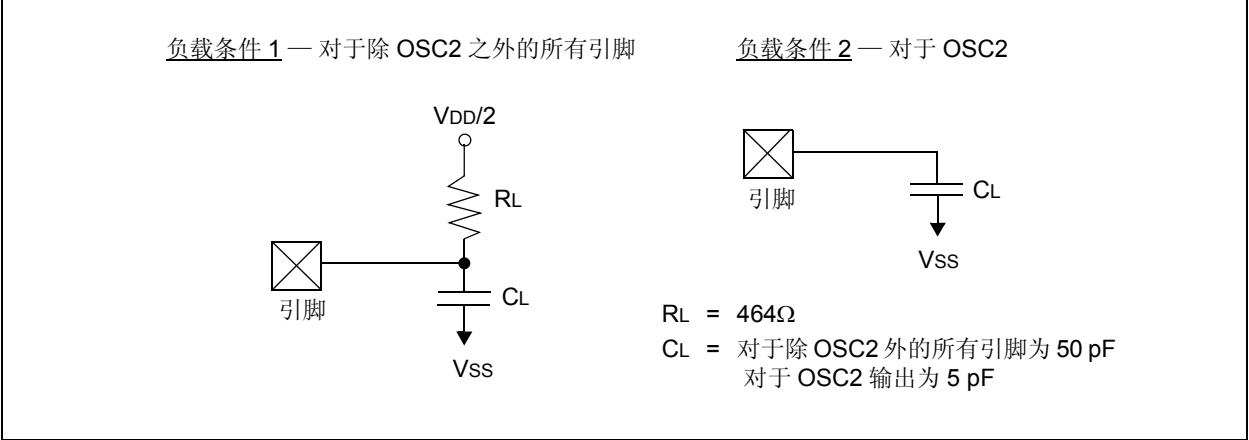


图 21-2: 外部时钟时序

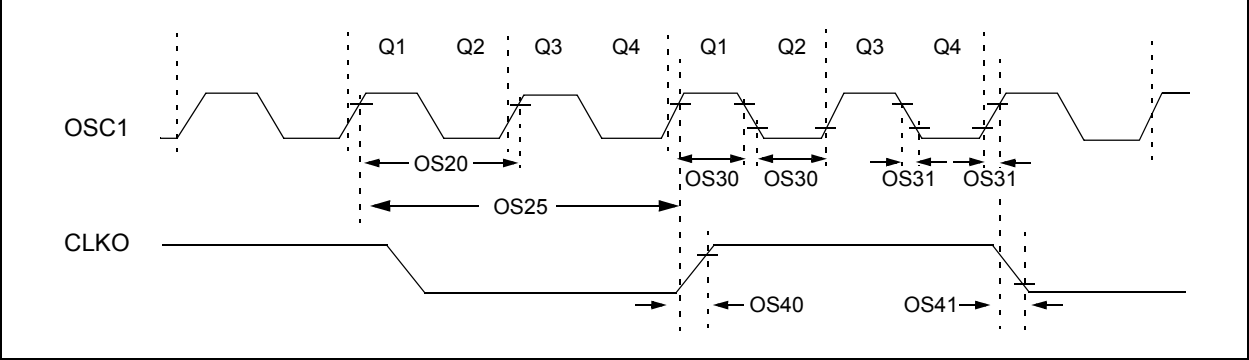


表 21-12: 外部时钟时序要求

AC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (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 (2) (外部时钟仅允许运行于 EC 模式)	6 6	— —	15.00 15.00	MHz MHz	EC EC, 带有 32x PLL
		振荡器频率 (2)	6 6	— —	15.00 15.00	MHz MHz	HS 内部 FRC
OS20	TOSC	TOSC = 1/FOSC (3)	16.5	—	DC	ns	
OS25	Tcy	指令周期时间 (2)(4)	33	—	DC	ns	
OS30	TosL, TosH	外部时钟 (2) 输入 (OSC1) 高电平或低电平时间	.45 x TOSC	—	—	ns	EC
OS31	TosR, TosF	外部时钟 (2) 输入 (OSC1) 上升或下降时间	—	—	20	ns	EC
OS40	TckR	CLKO 上升时间 (2)(5)	—	6	10	ns	
OS41	TckF	CLKO 下降时间 (2)(5)	—	6	10	ns	

- 注 1: 除非另外声明, 否则“典型值”一列中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。
- 2: 这些参数为特性值, 但生产时未经测试。
- 3: 当禁止 PLL 时, 振荡器频率 (Fosc) 与 FIN 相等。当使能 PLL 时, Fosc 等于 FIN 的 4 倍。
- 4: 指令周期 (Tcy) 等于输入振荡器时基周期的 2 倍。所有规定值均基于标准运行条件下, 器件执行代码时对应特定振荡器类型的特征数据。超过规定值可导致振荡器运行不稳定, 并 / 或使电流消耗超过预期。所有器件在测试“最小”值时, 均在 OSC1/CLKI 引脚接入了外部时钟。当使用外部时钟输入时, 所有器件的“最大”周期时间极限为“DC”(无时钟)。
- 5: 测量在 EC 模式下进行的。在 OSC2 引脚上测量 CLKO 信号。

dsPIC30F1010/202X

表 21-13: PLL 时钟时序规范 (VDD = 3.0 和 5.0V)

AC 特性		标准工作条件: 3.3V 和 5.0V (±10%) (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
OS50	FPLLI	PLL 输入频率范围 ⁽²⁾	6	—	15	MHz	EC, HS 模式, 带 32x PLL
OS51	FSYS	片内 PLL 输出 ⁽²⁾	192	—	480	MHz	EC, HS 模式, 带 32x PLL
OS52	TLOC	PLL 启动时间 (锁定时间)	—	20	50	μs	
OS53	DCLK	CLKO 稳定性 (抗抖动)	—	—	1	%	在 100 ms 周期内测量

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”一列中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

表 21-14: 内部时钟时序示例

时钟振荡器模式	F _{IN} (MHz) ⁽¹⁾	T _{cy} (μs) ⁽²⁾	MIPS ⁽³⁾ 无 PLL	MIPS ⁽⁴⁾ PLL x32
EC	10	0.2	5.0	20
	15	0.133	7.5	30
HS	10	0.2	5.0	20
	15	0.133	7.5	30

注 1: 假设振荡器后分频器分频比为 1。

2: 指令执行周期时间: T_{cy} = 1 / MIPS。

3: 没有 PLL 时的指令执行频率: MIPS = F_{IN}/2 (因为每个指令周期有 2 个 Q 时钟)。

4: 有 PLL 时的指令执行频率: MIPS = (F_{IN} * 2)。

表 21-15: AC 特性: 内部 RC 精度

AC 特性		标准工作条件: 3.3V 和 5.0V (±10%) (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 = 6.4 MHz ⁽¹⁾ 时的内部 FRC 精度						
	FRC	-0.06	—	+0.06	%	+25°C	VDD = 3.0-3.6V
		-0.06	—	+0.06	%	+25°C	VDD = 4.5-5.5V
		-1	—	+1	%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
		-1	—	+1	%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
		-1	—	+1	%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V
	FRC 频率 = 9.7 MHz ⁽¹⁾ 时的内部 FRC 精度						
	FRC	-0.06	—	+0.06	%	+25°C	VDD = 3.0-3.6V
		-0.06	—	+0.06	%	+25°C	VDD = 4.5-5.5V
		-1	—	+1	%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
		-1	—	+1	%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
		-1	—	+1	%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V
	FRC 频率 = 14.55 MHz ⁽¹⁾ 时的内部 FRC 精度						
	FRC	-0.06	—	+0.06	%	+25°C	VDD = 3.0-3.6V
		-0.06	—	+0.06	%	+25°C	VDD = 4.5-5.5V
		-1	—	+1	%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
		-1	—	+1	%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
		-1	—	+1	%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V

注 1: 在 25°C 和 5V 条件下进行频率校准。TUN 位可用来补偿温度漂移。

dsPIC30F1010/202X

表 21-16: AC 特性: 内部 RC 抖动

AC 特性		标准工作条件: 3.3V 和 5.0V (±10%) (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 = 6.4 MHz ⁽¹⁾ 时的内部 FRC 抖动						
	FRC	-1	—	+1	%	+25°C	VDD = 3.0-3.6V
		-1	—	+1	%	+25°C	VDD = 4.5-5.5V
		-1	—	+1	%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
		-1	—	+1	%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
		-1	—	+1	%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V
	FRC 频率 = 9.7 MHz ⁽¹⁾ 时的内部 FRC 抖动						
	FRC	-1	—	+1	%	+25°C	VDD = 3.0-3.6V
		-1	—	+1	%	+25°C	VDD = 4.5-5.5V
		-1	—	+1	%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
		-1	—	+1	%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
		-1	—	+1	%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V
	FRC 频率 = 14.55 MHz ⁽¹⁾ 时的内部 FRC 抖动						
	FRC	-1	—	+1	%	+25°C	VDD = 3.0-3.6V
		-1	—	+1	%	+25°C	VDD = 4.5-5.5V
		-1	—	+1	%	-40°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
		-1	—	+1	%	-40°C ≤ TA ≤ +85°C	VDD = 4.5-5.5V
		-1	—	+1	%	-40°C ≤ TA ≤ +125°C	VDD = 4.5-5.5V

注 1: 在 25°C 和 5V 条件下进行频率校准。TUN 位可用来补偿温度漂移。

图 21-3: CLK0 和 I/O 时序特性

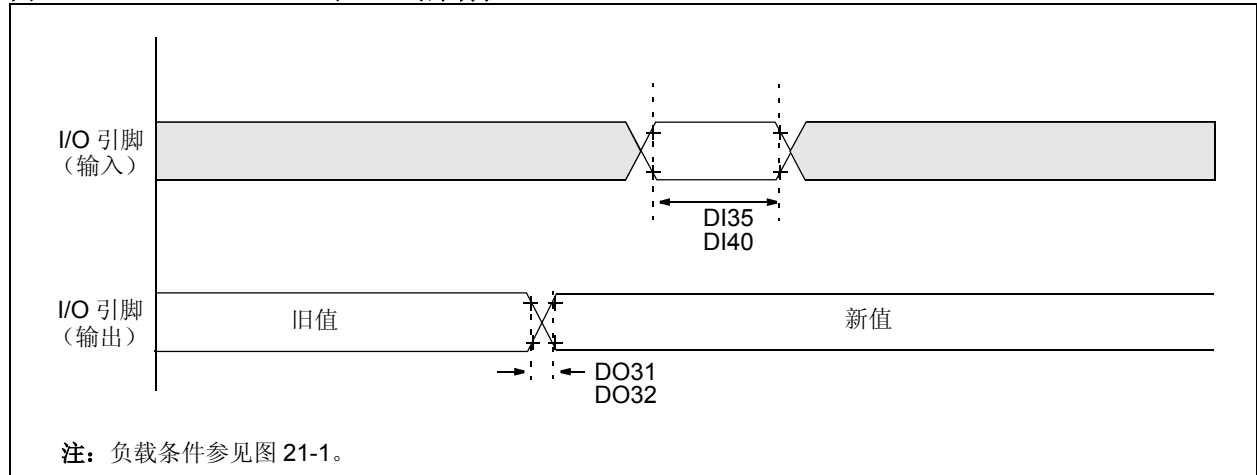


表 21-17: CLK0 和 I/O 时序要求

AC 特性		标准工作条件: 3.3V 和 5.0V ($\pm 10\%$) (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)(2)	最小值	典型值 (3)	最大值	单位	条件
DO31	TioR	端口输出上升时间	—	10	25	ns	—
DO32	TioF	端口输出下降时间	—	10	25	ns	—
DI35	TINP	INTx 引脚高电平或低电平时间 (输出)	20	—	—	ns	—
DI40	TRBP	CNx 高电平或低电平时间 (输入)	2 Tcy	—	—	ns	—

- 注 1: 这些参数为异步事件, 与任何内部时钟边沿无关。
 2: 这些参数为特性值, 但生产时未经测试。
 3: 除非另外声明, 否则“典型值”一列中的数据都是在 5V、 25°C 的条件下给出的。

图 21-4: 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

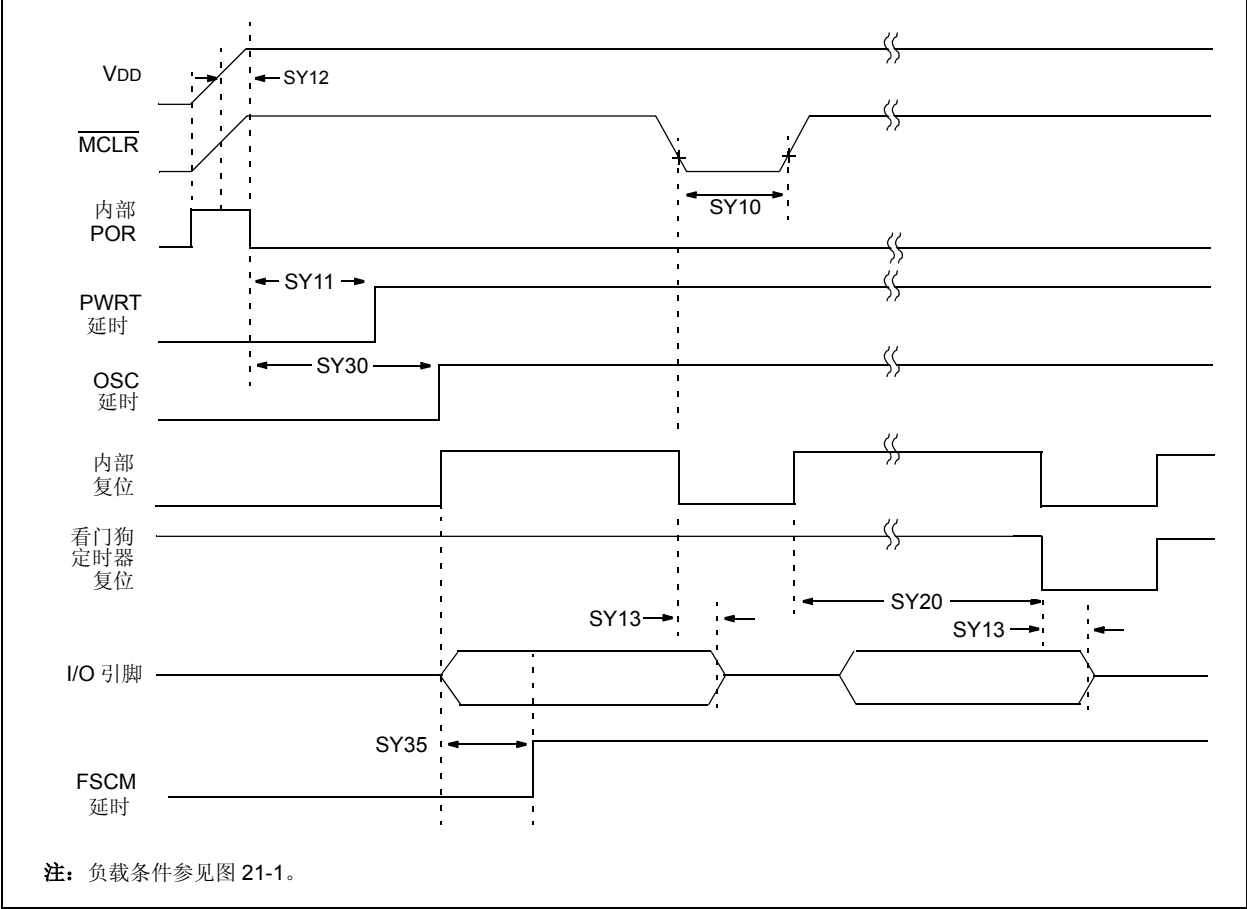


表 21-18: 复位、看门狗定时器、振荡器起振定时器、上电延时定时器时序要求

AC 特性			标准工作条件: 3.3V 和 5.0V ($\pm 10\%$) (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 (2)	最大值	单位	条件
SY10	TmCL	MCLR 脉冲宽度 (低电平)	2	—	—	μs	-40°C 至 125°C
SY11	TPWRT	上电延时定时器周期	0.75 1.5 3 6 12 24 48 96	1 2 4 8 16 32 64 128	1.25 2.5 5 10 20 40 80 160	ms	-40°C 至 125°C 用户可编程
SY12	TPOR	上电复位延时	3	10	30	μs	-40°C 至 125°C
SY13	TIOZ	MCLR 低电平或看门狗定时器复位引起 I/O 高阻态	—	0.8	1.0	μs	
SY20	TWDT1	看门狗定时器超时周期 (无预分频器)	1.4	2.1	2.8	ms	$V_{DD} = 5\text{V}$, -40°C 至 $+125^{\circ}\text{C}$
	TWDT2		1.4	2.1	2.8	ms	$V_{DD} = 3.3\text{V}$, -40°C 至 $+125^{\circ}\text{C}$
SY30	TOST	振荡器起振定时器周期	—	1024 T_{OSC}	—	—	$T_{OSC} = OSC1$ 周期
SY35	TFSCM	故障保护时钟监视器延时	—	500	—	μs	-40°C 至 125°C

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”一列中的数据都是在 5V 、 25°C 的条件下给出的。

图 21-5: 带隙启动时间特性

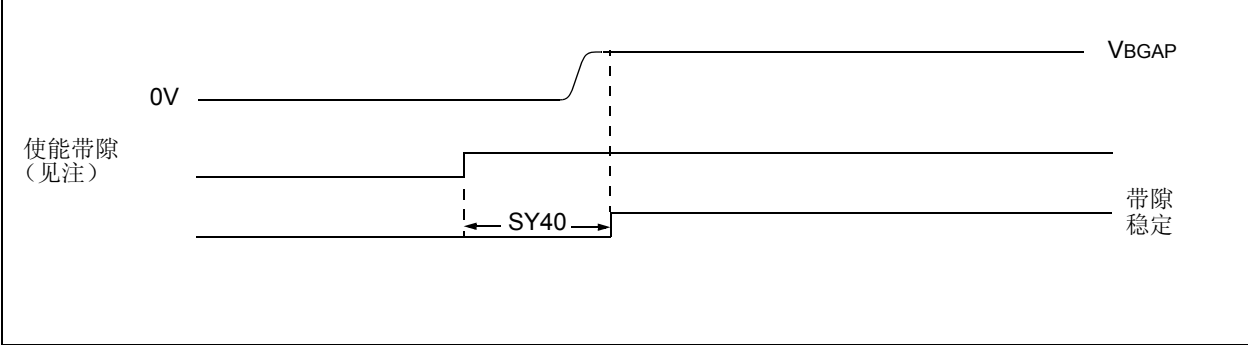


表 21-19: 带隙启动时间要求

AC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (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
SY40	TBGAP	带隙启动时间	—	40	65	µs	定义为带隙功能使能和带隙参考电压稳定之间的时间。RCON<13> 状态位。

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” 一行中的数据都是在 5V、25 °C 的条件下给出的。

图 21-6: 定时器外部时钟时序特性

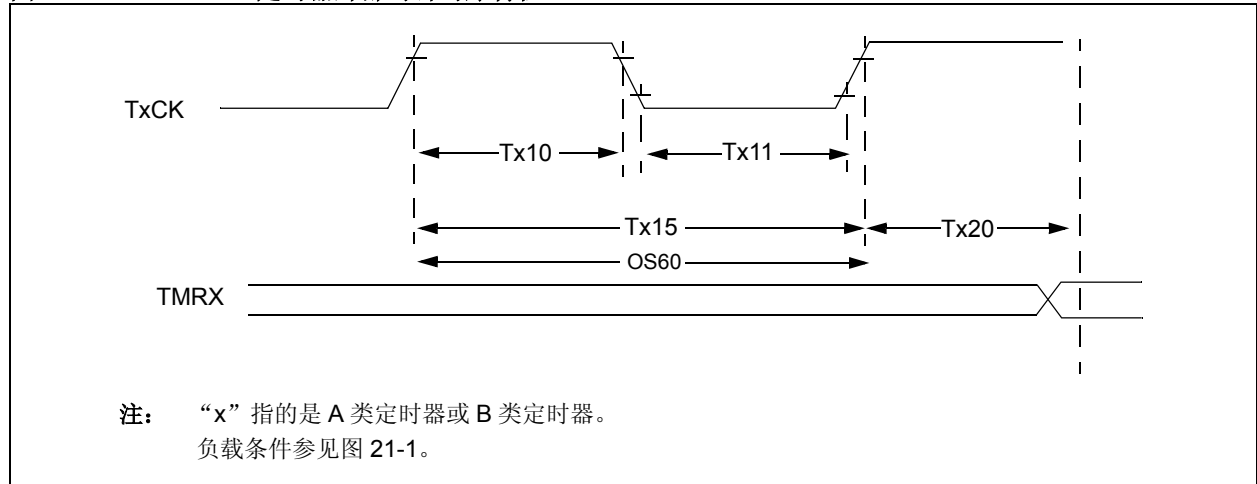


表 21-20: TIMER1 外部时钟时序要求

AC 特性		标准工作条件: 3.3V 和 5.0V (±10%) (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	—	—	也必须满足参数 TA15
			同步, 带预分频器	10	—	—	
			异步	10	—	—	
TA11	TTxL	TxCK 低电平时间	同步, 无预分频器	0.5 Tcy + 20	—	—	也必须满足参数 TA15
			同步, 带预分频器	10	—	—	
			异步	10	—	—	
TA15	TTxP	TxCK 输入周期	同步, 无预分频器	Tcy + 10	—	—	N = 预分频值 (1, 8, 64, 256)
			同步, 带预分频器	取 20 ns 或 (Tcy + 40)/N 中的较大值	—	—	
			异步	20	—	—	
OS60	Ft1	SOSC1/T1CK 振荡器输入频率范围 (通过置 1 TCS 位 (T1CON 的 bit 1) 使能振荡器)	DC	—	50	kHz	
TA20	TCKEXTMRL	从外部 TxCK 时钟边沿至定时器递增之间的延时	0.5 Tcy		1.5 Tcy	—	

dsPIC30F1010/202X

表 21-21: TIMER2 外部时钟时序要求

AC 特性				标准工作条件: 3.3V 和 5.0V (±10%) (除非另外声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (扩展级)				
参数编号	符号	特性		最小值	典型值	最大值	单位	条件
TB10	TtxH	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	TtxL	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	TtxP	TxCK 输入周期	同步, 无预分频器	$T_{CY} + 10$	—	—	ns	N = 预分频值 (1, 8, 64)
			同步, 带预分频器	取 20 ns 或 $(T_{CY} + 40)/N$ 中的较大值				
TB20	TCKEXTMRL	从外部 TxCK 时钟边沿至定时器递增之间的延时		$0.5 T_{CY}$	—	$1.5 T_{CY}$	—	

表 21-22: TIMER3 外部时钟时序要求

AC 特性				标准工作条件: 3.3V 和 5.0V (±10%) (除非另外声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (扩展级)				
参数编号	符号	特性		最小值	典型值	最大值	单位	条件
TC10	TtxH	TxCK 高电平时间	同步	$0.5 T_{CY} + 20$	—	—	ns	也必须满足参数 TC15
TC11	TtxL	TxCK 低电平时间	同步	$0.5 T_{CY} + 20$	—	—	ns	也必须满足参数 TC15
TC15	TtxP	TxCK 输入周期	同步, 无预分频器	$T_{CY} + 10$	—	—	ns	N = 预分频值 (1, 8, 64, 256)
			同步, 带预分频器	取 20 ns 或 $(T_{CY} + 40)/N$ 中的较大值				
TC20	TCKEXTMRL	从外部 TxCK 时钟边沿至定时器递增之间的延时		$0.5 T_{CY}$	—	$1.5 T_{CY}$	—	

图 21-7: 输入捕捉 (CAPx) 时序特性

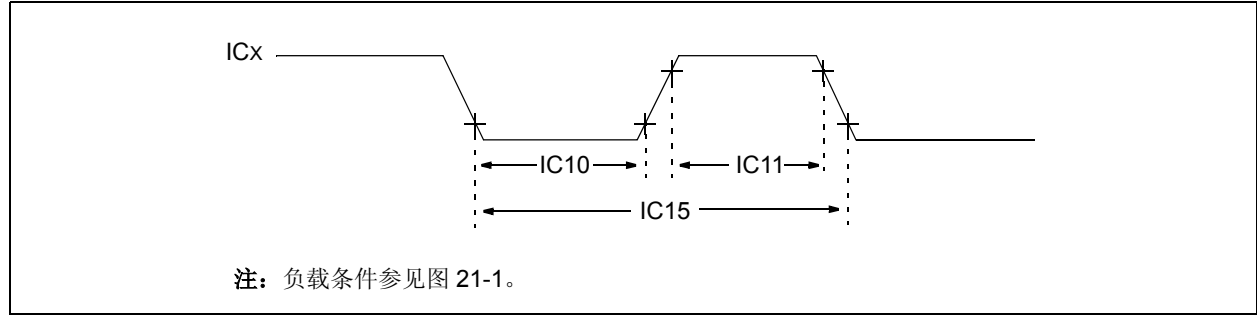


表 21-23: 输入捕捉时序要求

AC 特性		标准工作条件: 3.3V 和 5.0V (±10%) (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性 ⁽¹⁾	最小值	最大值	单位	条件
IC10	TccL	ICx 输入低电平时间	(无预分频器)	0.5 Tcy + 20	—	ns
		带预分频器	10	—	ns	
IC11	TccH	ICx 输入高电平时间	(无预分频器)	0.5 Tcy + 20	—	ns
		带预分频器	10	—	ns	
IC15	TccP	ICx 输入周期	(2 Tcy + 40) / N	—	ns	N = 预分频值 (1, 4, 16)

注 1: 这些参数为特性值, 但生产时未经测试。

图 21-8: 输出比较模块 (OCx) 时序特性

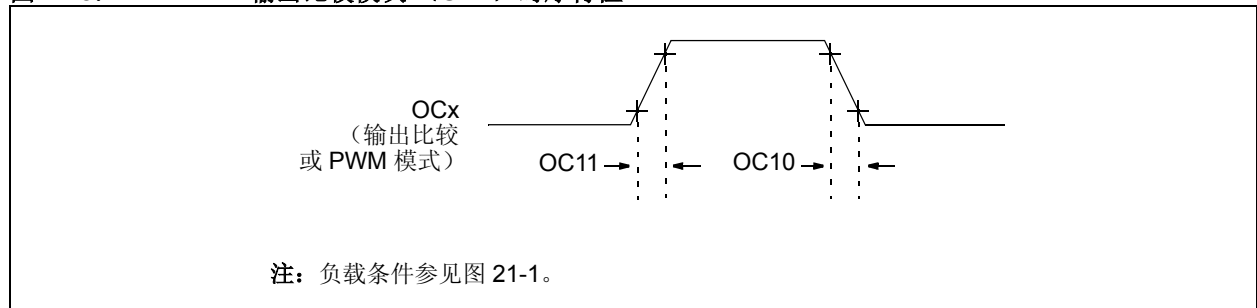


表 21-24: 输出比较模块时序要求

AC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (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
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。

2: 除非另外声明, 否则“典型值”一列中的数据都是在 5V、25 °C 的条件下给出的。这些参数仅供设计参考, 未经测试。

dsPIC30F1010/202X

图 21-9: 输出比较 /PWM 模块时序特性

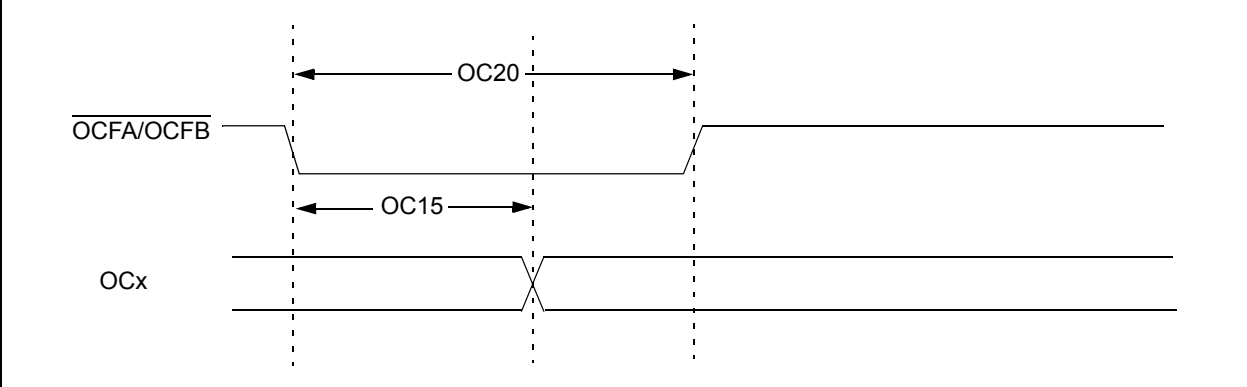


表 21-25: 简单输出比较 /PWM 模式时序要求

AC 特性				标准工作条件: 3.3V 和 5.0V (±10%) (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	
OC15	TFD	故障输入至 PWM I/O 发生变化的时间	—	—	25	ns	VDD = 3.3V	-40°C 至 85°C
					TBD	ns	VDD = 5V	
OC20	TFLT	故障输入脉冲宽度	—	—	50	ns	VDD = 0.33V	-40°C 至 85°C
					TBD	ns	VDD = 5V	

图注: TBD = 待定

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” 一行中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

图 21-10: 电源 PWM 模块故障时序特性

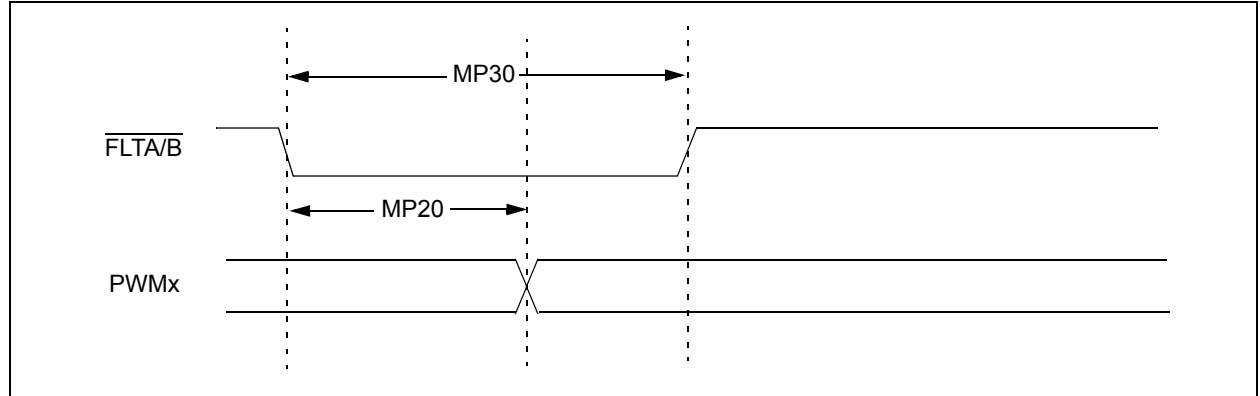


图 21-11: 电源 PWM 模块时序特性

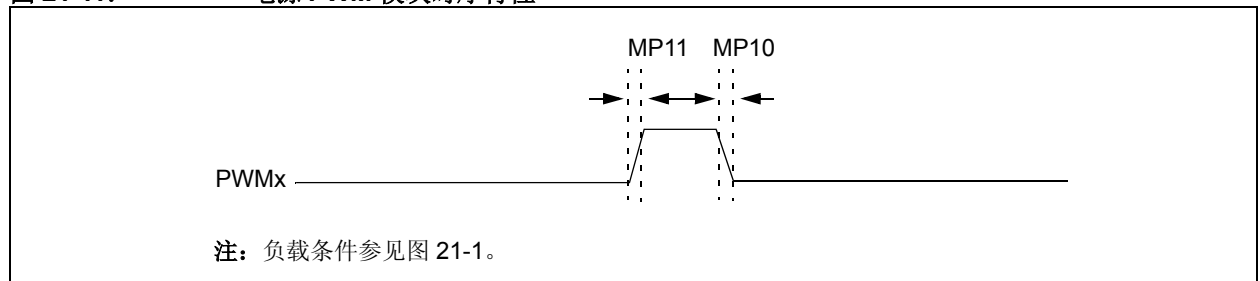


表 21-26: 电源 PWM 模块时序要求

AC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (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
MP10	TfPWM	PWM 输出下降时间	—	10	25	ns	VDD = 5V
MP11	TrPWM	PWM 输出上升时间	—	10	25	ns	VDD = 5V
MP12	TfPWM	PWM 输出下降时间	—	TBD	TBD	ns	VDD = 3.3V
MP13	TrPWM	PWM 输出上升时间	—	TBD	TBD	ns	VDD = 3.3V
MP20	TFD	故障输入 ↓ 至 PWM I/O 变化	—	—	TBD	ns	VDD = 3.3V
					25	ns	VDD = 5V
MP30	TFH	最小脉冲宽度	—	—	TBD	ns	VDD = 3.3V
					50	ns	VDD = 5V

图注: TBD = 待定

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”一列中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

dsPIC30F1010/202X

图 21-12: SPI 模块主模式 (CKE = 0) 时序特性

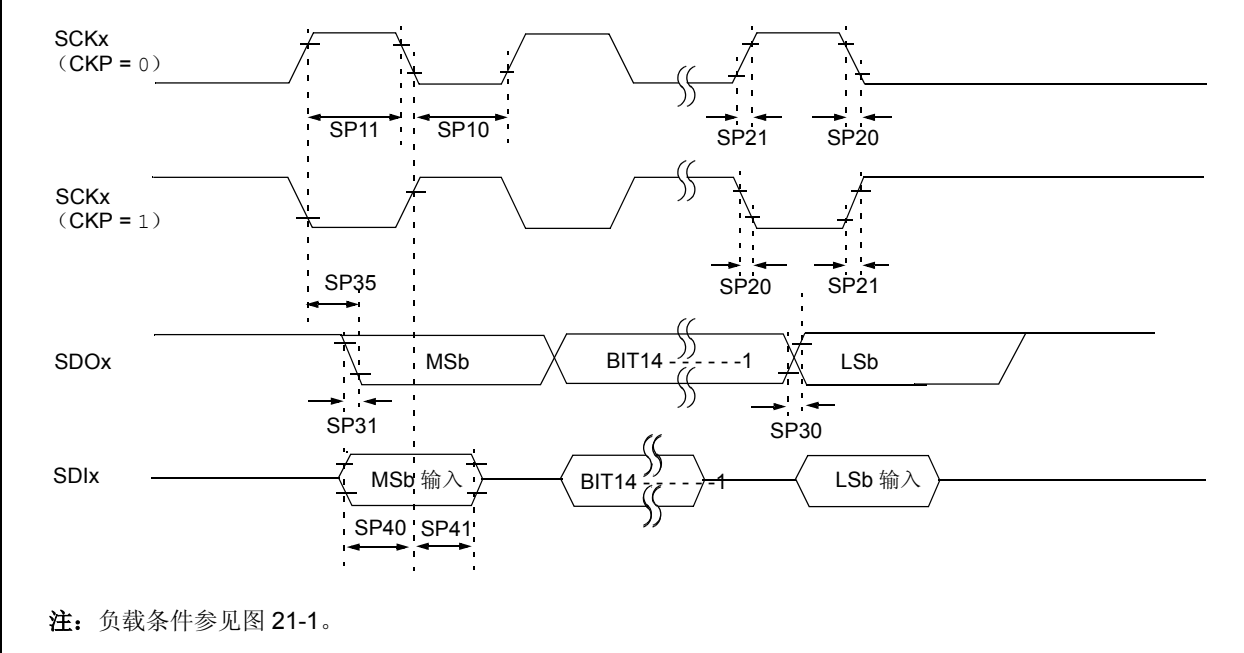


表 21-27: SPI 主模式 (CKE = 0) 时序要求

AC 特性		标准工作条件: 3.3V 和 5.0V (±10%) (除非另外声明) 工作温度					
		-40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)					
参数编号	符号	特性 (1)	最小值	典型值 (2)	最大值	单位	条件
SP10	TscL	SCKx 输出低电平时间 (3)	Tcy/2	—	—	ns	—
SP11	TscH	SCKx 输出高电平时间 (3)	Tcy/2	—	—	ns	—
SP20	TscF	SCKx 输出下降时间 (4)	—	—	—	ns	见参数 D032
SP21	TscR	SCKx 输出上升时间 (4)	—	—	—	ns	见参数 D031
SP30	TdoF	SDOx 数据输出下降时间 (4)	—	—	—	ns	见参数 D032
SP31	TdoR	SDOx 数据输出上升时间 (4)	—	—	—	ns	见参数 D031
SP35	Tsch2doV, TscL2doV	SCKx 边沿后的 SDOx 数据输出有效时间	—	—	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	—

- 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” 一列中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。
- 3: SCK 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 SPI 引脚上为 50 pF 负载。

图 21-13: SPI 模块主模式 (CKE = 1) 时序特性

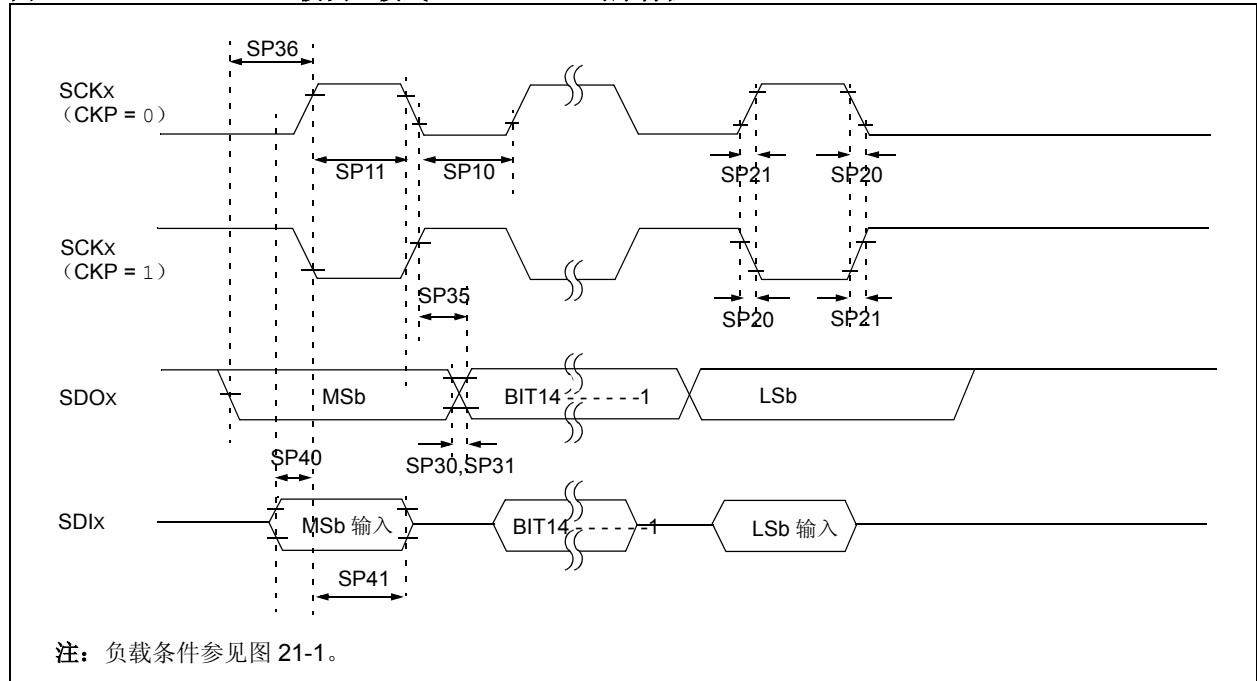


表 21-28: SPI 模块主模式 (CKE = 1) 时序要求

AC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性 (1)	最小值	典型值 (2)	最大值	单位	条件
SP10	TscL	SCKx 输出低电平时间 (3)	Tcy / 2	—	—	ns	—
SP11	TscH	SCKx 输出高电平时间 (3)	Tcy / 2	—	—	ns	—
SP20	TscF	SCKx 输出下降时间 (4)	—	—	—	ns	见参数 D032
SP21	TscR	SCKx 输出上升时间 (4)	—	—	—	ns	见参数 D031
SP30	TdoF	SDOx 数据输出下降时间 (4)	—	—	—	ns	见参数 D032
SP31	TdoR	SDOx 数据输出上升时间 (4)	—	—	—	ns	见参数 D031
SP35	Tsch2doV, TscL2doV	SCKx 边沿后的 SDOx 数据输出有效时间	—	—	30	ns	—
SP36	TdoV2sc, TdoV2scL	SDOx 数据输出建立到第一个 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	20	—	—	ns	—
SP41	Tsch2diL, TscL2diL	SDIx 数据输入到 SCKx 边沿的保持时间	20	—	—	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”一列中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

3: SCK 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 SPI 引脚上为 50 pF 负载。

图 21-14: SPI 模块从模式 (CKE = 0) 时序特性

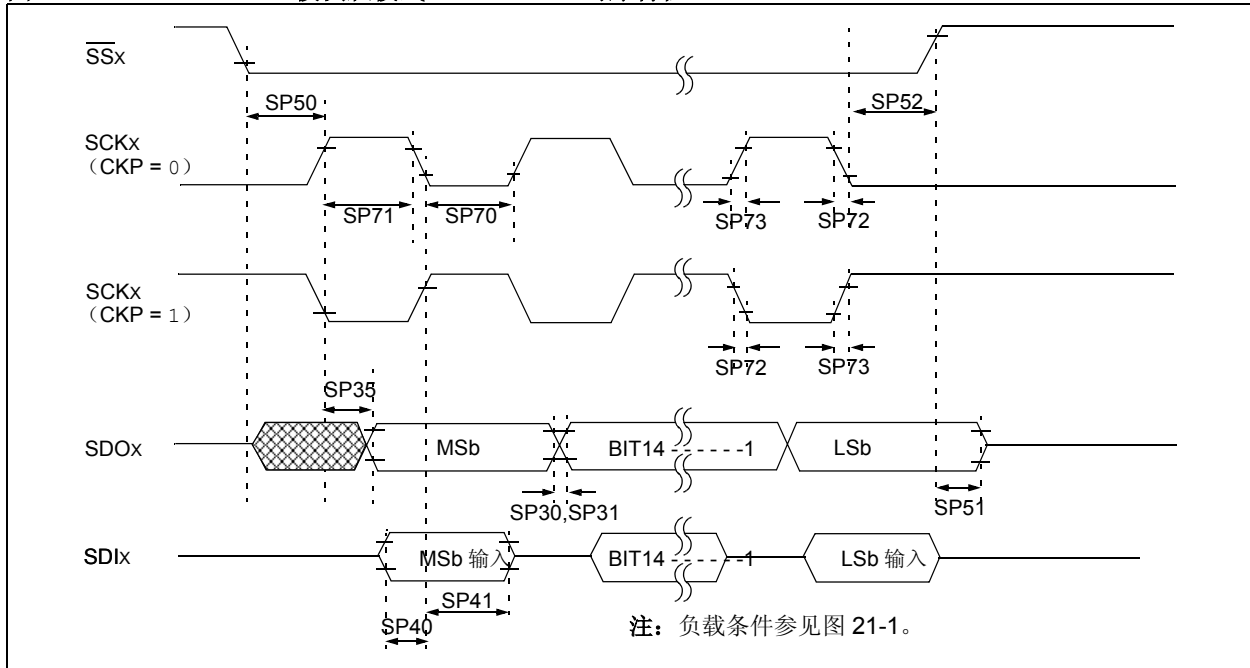


表 21-29: SPI 模块从模式 (CKE = 0) 时序要求

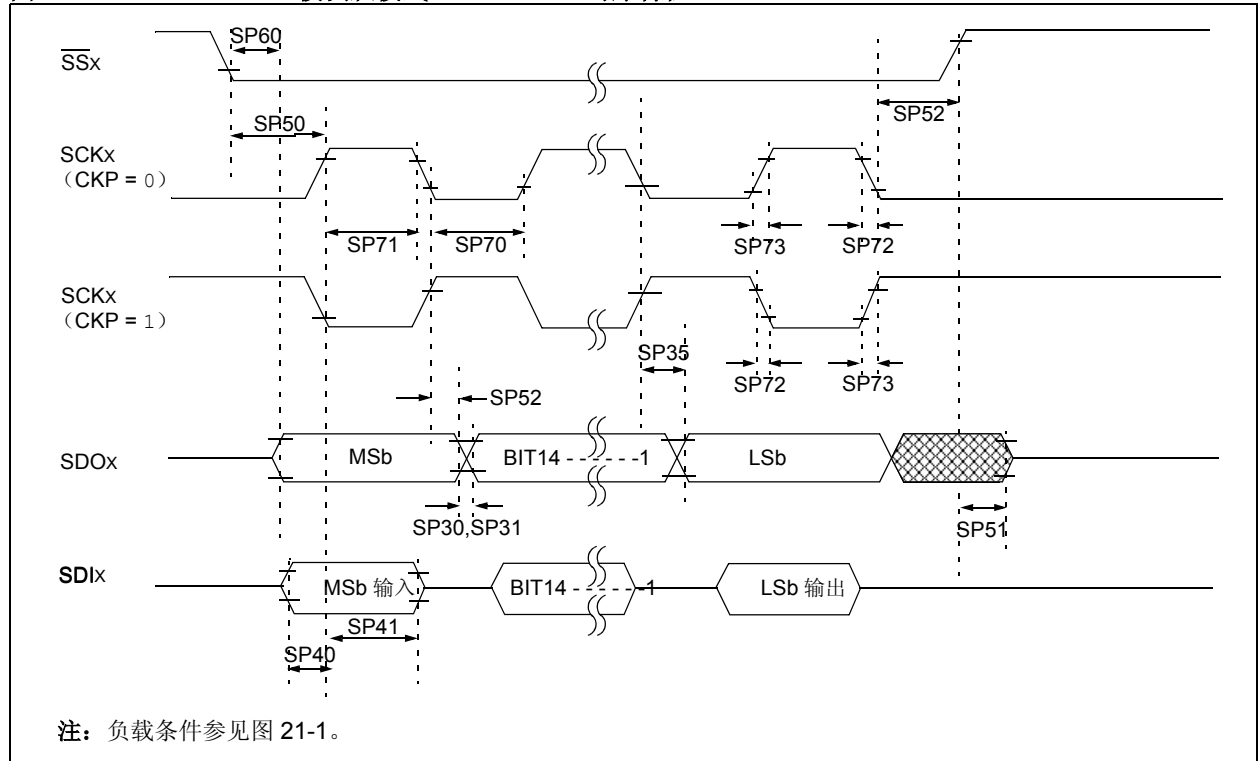
AC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性 (1)	最小值	典型值 (2)	最大值	单位	条件
SP70	TscL	SCKx 输入低电平时间	30	—	—	ns	—
SP71	TscH	SCKx 输入高电平时间	30	—	—	ns	—
SP72	TscF	SCKx 输入下降时间 (3)	—	10	25	ns	—
SP73	TscR	SCKx 输入上升时间 (3)	—	10	25	ns	—
SP30	TdoF	SDOx 数据输出下降时间 (3)	—	—	—	ns	见参数 D032
SP31	TdoR	SDOx 数据输出上升时间 (3)	—	—	—	ns	见参数 D031
SP35	Tsch2doV TscL2doV	SCKx 边沿后的 SDOx 数据输出有效时间	—	—	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 (3)	10	—	50	ns	—
SP52	Tsch2ssH TscL2ssH	SCK 边沿后出现 SSx 有效边沿的时间	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”一列中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

3: 假定所有 SPI 引脚上为 50 pF 负载。

图 21-15: SPI 模块从模式 (CKE = 1) 时序特性



dsPIC30F1010/202X

表 21-30: SPI 模块从模式 (CKE = 1) 时序要求

AC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性 (1)	最小值	典型值 (2)	最大值	单位	条件
SP70	TscL	SCKx 输入低电平时间	30	—	—	ns	—
SP71	TscH	SCKx 输入高电平时间	30	—	—	ns	—
SP72	TscF	SCKx 输入下降时间 (3)	—	10	25	ns	—
SP73	TscR	SCKx 输入上升时间 (3)	—	10	25	ns	—
SP30	TdoF	SDOx 数据输出下降时间 (3)	—	—	—	ns	见参数 D032
SP31	TdoR	SDOx 数据输出上升时间 (3)	—	—	—	ns	见参数 D031
SP35	Tsch2doV, TscL2doV	SCKx 边沿后的 SDOx 数据输出有效时间	—	—	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	$\overline{SS}$ ↑ 到 SDOx 输出高阻态的时间 (4)	10	—	50	ns	—
SP52	Tsch2ssH, TscL2ssH	SCKx 边沿后出现 $\overline{SSx}$ ↑ 的时间	1.5 Tcy + 40	—	—	ns	—
SP60	TssL2doV	$\overline{SSx}$ 边沿后 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” 一行中的数据都是在 5V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。
- 3: SCK 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 SPI 引脚上为 50 pF 负载。

图 21-16: I²C™ 总线启动 / 停止位时序特性 (主模式)

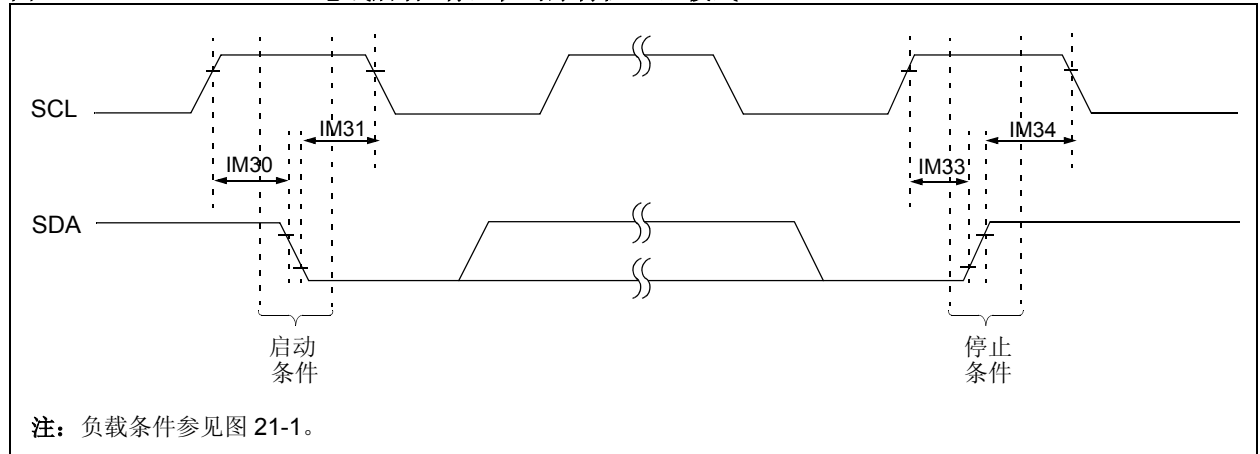
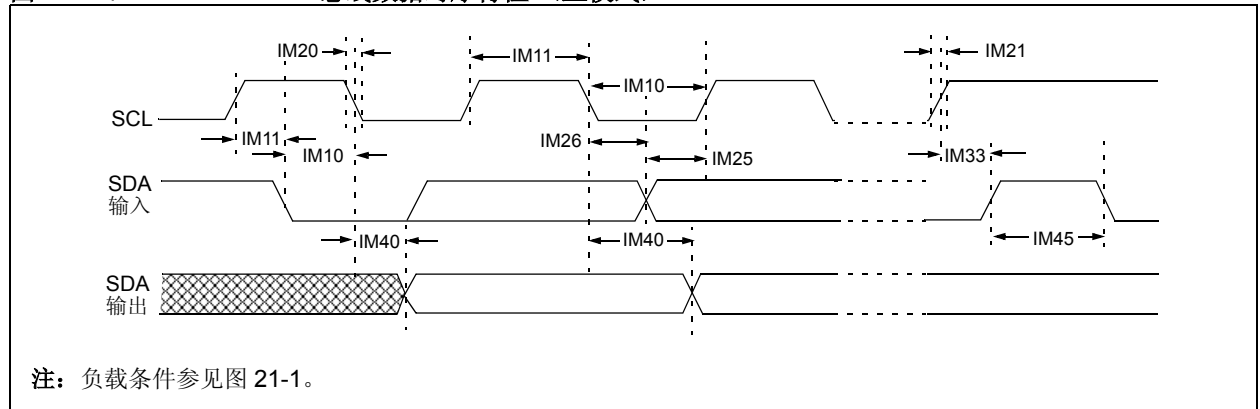


图 21-17: I²C™ 总线数据时序特性 (主模式)



dsPIC30F1010/202X

表 21-31: I²C™ 总线数据时序要求（主模式）

AC 特性				标准工作条件: 3.3V 和 5.0V (±10%) (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 ⁽¹⁾	最大值	单位	条件
IM10	TLO:SCL	时钟低电平时间	100 kHz 模式	T _{CY} / 2 (BRG + 1)	—	μs	—
			400 kHz 模式	T _{CY} / 2 (BRG + 1)	—	μs	—
			1 MHz 模式 ⁽²⁾	T _{CY} / 2 (BRG + 1)	—	μs	—
IM11	THI:SCL	时钟高电平时间	100 kHz 模式	T _{CY} / 2 (BRG + 1)	—	μs	—
			400 kHz 模式	T _{CY} / 2 (BRG + 1)	—	μs	—
			1 MHz 模式 ⁽²⁾	T _{CY} / 2 (BRG + 1)	—	μs	—
IM20	TF:SCL	SDA 和 SCL 下降时间	100 kHz 模式	—	300	ns	规定 C _B 的值在 10 至 400 pF 之间
			400 kHz 模式	20 + 0.1 C _B	300	ns	
			1 MHz 模式 ⁽²⁾	—	100	ns	
IM21	TR:SCL	SDA 和 SCL 上升时间	100 kHz 模式	—	1000	ns	规定 C _B 的值在 10 至 400 pF 之间
			400 kHz 模式	20 + 0.1 C _B	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⁾	TBD	—	ns	
IM26	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns	—
			400 kHz 模式	0	0.9	μs	
			1 MHz 模式 ⁽²⁾	TBD	—	ns	
IM30	TSU:STA	启动条件建立时间	100 kHz 模式	T _{CY} / 2 (BRG + 1)	—	μs	仅与重复启动条件相关
			400 kHz 模式	T _{CY} / 2 (BRG + 1)	—	μs	
			1 MHz 模式 ⁽²⁾	T _{CY} / 2 (BRG + 1)	—	μs	
IM31	THD:STA	启动条件保持时间	100 kHz 模式	T _{CY} / 2 (BRG + 1)	—	μs	在此时间之后将产生第一个时钟脉冲
			400 kHz 模式	T _{CY} / 2 (BRG + 1)	—	μs	
			1 MHz 模式 ⁽²⁾	T _{CY} / 2 (BRG + 1)	—	μs	
IM33	TSU:STO	停止条件建立时间	100 kHz 模式	T _{CY} / 2 (BRG + 1)	—	μs	—
			400 kHz 模式	T _{CY} / 2 (BRG + 1)	—	μs	
			1 MHz 模式 ⁽²⁾	T _{CY} / 2 (BRG + 1)	—	μs	
IM34	THD:STO	停止条件保持时间	100 kHz 模式	T _{CY} / 2 (BRG + 1)	—	ns	—
			400 kHz 模式	T _{CY} / 2 (BRG + 1)	—	ns	
			1 MHz 模式 ⁽²⁾	T _{CY} / 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⁾	—	—	ns	—
IM45	TBF:SDA	总线空闲时间	100 kHz 模式	4.7	—	μs	在新的发送操作启动之前总线必须保持空闲的时间
			400 kHz 模式	1.3	—	μs	
			1 MHz 模式 ⁽²⁾	TBD	—	μs	

图注: TBD = 待定

注 1: BRG 为 I²C™ 波特率发生器的取值。请参见《dsPIC30F 系列参考手册》(DS70046E_CN) 中的 “I²C” 一节。

2: 对于所有 I²C 引脚, 最大引脚电容 = 10 pF (仅对于 1 MHz 模式)。

表 21-31: I²C™ 总线数据时序要求（主模式）（续）

AC 特性			标准工作条件: 3.3V 和 5.0V (±10%) (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 ⁽¹⁾	最大值	单位	条件
IM50	C _B	总线容性负载	0	400	pF	

图注: TBD = 待定

注 1: BRG 为 I²C™ 波特率发生器的取值。请参见《dsPIC30F 系列参考手册》(DS70046E_CN) 中的 “I²C” 一节。

2: 对于所有 I²C 引脚, 最大引脚电容 = 10 pF (仅对于 1 MHz 模式)。

图 21-18: I²C™ 总线启动 / 停止位时序特性（从模式）

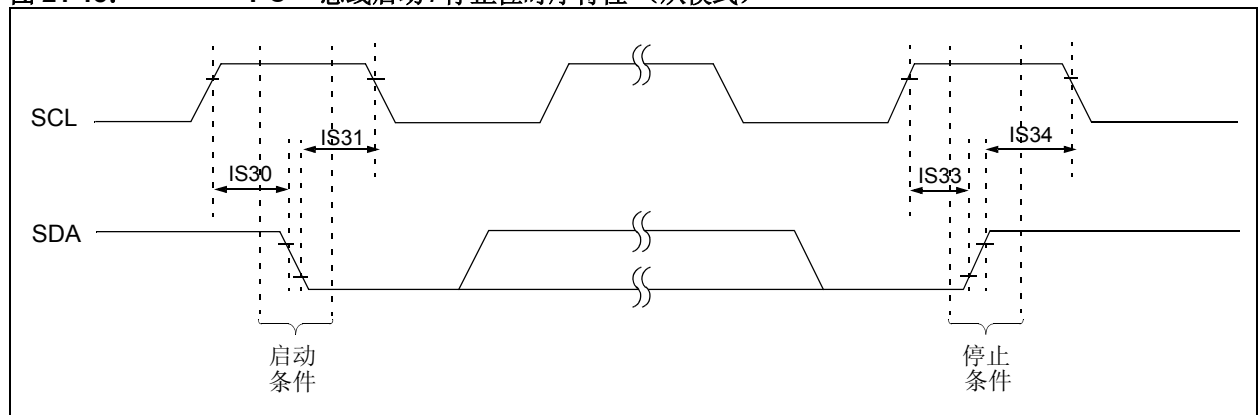
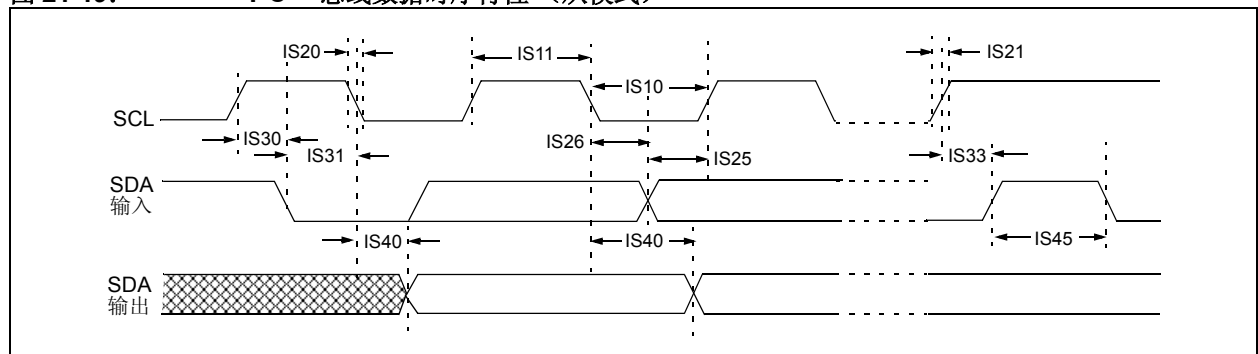


图 21-19: I²C™ 总线数据时序特性（从模式）



dsPIC30F1010/202X

表 21-32: I²C™ 总线数据时序要求（从模式）

AC 特性		标准工作条件: 3.3V 和 5.0V (±10%) (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)					
参数编号	符号	特性	最小值	最大值	单位	条件	
IS10	TLO:SCL	时钟低电平时间	100 kHz 模式	4.7	—	μs	器件工作频率不能低于 1.5 MHz
			400 kHz 模式	1.3	—	μs	器件工作频率不能低于 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	器件工作频率不能低于 1.5 MHz
			400 kHz 模式	0.6	—	μs	器件工作频率不能低于 10 MHz
			1 MHz 模式 (1)	0.5	—	μs	—
IS20	TF:SCL	SDA 和 SCL 下降时间	100 kHz 模式	—	300	ns	规定 CB 的值在 10 至 400 pF 之间
			400 kHz 模式	20 + 0.1 CB	300	ns	
			1 MHz 模式 (1)	—	100	ns	
IS21	TR:SCL	SDA 和 SCL 上升时间	100 kHz 模式	—	1000	ns	规定 CB 的值在 10 至 400 pF 之间
			400 kHz 模式	20 + 0.1 CB	300	ns	
			1 MHz 模式 (1)	—	300	ns	
IS25	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns	—
			400 kHz 模式	100	—	ns	
			1 MHz 模式 (1)	100	—	ns	
IS26	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns	—
			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	在此时间之后将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	在新的发送操作启动之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: 对于所有 I²C™ 引脚, 最大引脚电容 = 10 pF (仅对于 1 MHz 模式)。

表 21-33: 10 位高速 A/D 模块规范

AC 特性			标准工作条件: 3.3V 和 5.0V ($\pm 10\%$) (除非另外声明) 工作温度 -40°C $\leq$ TA $\leq$ +85°C (工业级) -40°C $\leq$ TA $\leq$ +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 VDD 电源	VDD - 0.3 或 2.7 中的较大值		VDD + 0.3 或 5.5 中的较小值	V	—
AD02	AVSS	模块 VSS 电源	VSS - 0.3		VSS + 0.3	V	—
模拟输入							
AD10	VINH-VINL	满量程输入范围	VSS		VDD	V	—
AD11	VIN	绝对输入电压	AVSS - 0.3		AVDD + 0.3	V	—
AD12	—	泄漏电流	—	± 0.001	± 0.244	μA	VINL = AVSS = 0V, AVDD = 5V 源阻抗 = 1 k Ω
AD13	—	泄漏电流	—	± 0.001	± 0.244	μA	VINL = AVSS = 0V, AVDD = 3.3V 源阻抗 = 1 k Ω
AD17	RIN	模拟电压源的建议阻抗	—		1K	Ω	—
DC 精度							
AD20	Nr	分辨率	10 个数据位			位	—
AD21	INL	积分非线性误差	—	± 0.5	$< \pm 1$	LSb	VINL = AVSS = 0V AVDD = 5V
AD21A	INL	积分非线性误差	—	± 0.5	$< \pm 1$	LSb	VINL = AVSS = 0V AVDD = 3.3V
AD22	DNL	微分非线性误差	—	± 0.5	$< \pm 1$	LSb	VINL = AVSS = 0V AVDD = 5V
AD22A	DNL	微分非线性误差	—	± 0.5	$< \pm 1$	LSb	VINL = AVSS = 0V AVDD = 3.3V
AD23	GERR	增益误差	—	± 0.75	$< \pm 4.0$	LSb	VINL = AVSS = 0V AVDD = 5V
AD23A	GERR	增益误差	—	± 0.75	$< \pm 3.0$	LSb	VINL = AVSS = 0V AVDD = 3.3V

注 1: 因为采样电容最终将无法保持电荷, 因此低于 10 kHz 的时钟频率可能影响线性性能, 尤其是在温度较高时。

2: A/D 转换结果将不会随着输入电压的增加而减小, 而且不会丢失代码。

dsPIC30F1010/202X

表 21-33: 10 位高速 A/D 模块规范 (续)

AC 特性			标准工作条件: 3.3V 和 5.0V ($\pm 10\%$) (除非另外声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级) $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ (扩展级)				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
AD24	E _{OFF}	偏移误差	—	± 0.75	$\leq \pm 2.0$	LSb	$V_{\text{INL}} = A_{\text{VSS}} = V_{\text{SS}} = 0\text{V}$, $A_{\text{VDD}} = V_{\text{DD}} = 5\text{V}$
AD24A	E _{OFF}	偏移误差	—	± 0.75	$\leq \pm 2.0$	LSb	$V_{\text{INL}} = A_{\text{VSS}} = V_{\text{SS}} = 0\text{V}$, $A_{\text{VDD}} = V_{\text{DD}} = 3.3\text{V}$
AD25	—	单调性 (2)	—	—	—	—	保证
动态性能							
AD30	THD	总谐波失真	-77	-73	-68	dB	—
AD31	SINAD	信噪比和失真	—	58	—	dB	—
AD32	SFDR	伪空闲动态范围	—	-73	—	dB	—
AD33	F _{NYQ}	输入信号带宽	—	6	0.5	MHz	—
AD34	ENOB	有效位数	—	9.4	—	位	—

注 1: 因为采样电容最终将无法保持电荷, 因此低于 10 kHz 的时钟频率可能影响线性性能, 尤其是在温度较高时。

2: A/D 转换结果将不会随着输入电压的增加而减小, 而且不会丢失代码。

图 21-20: 每次中断的 A/D 转换时序

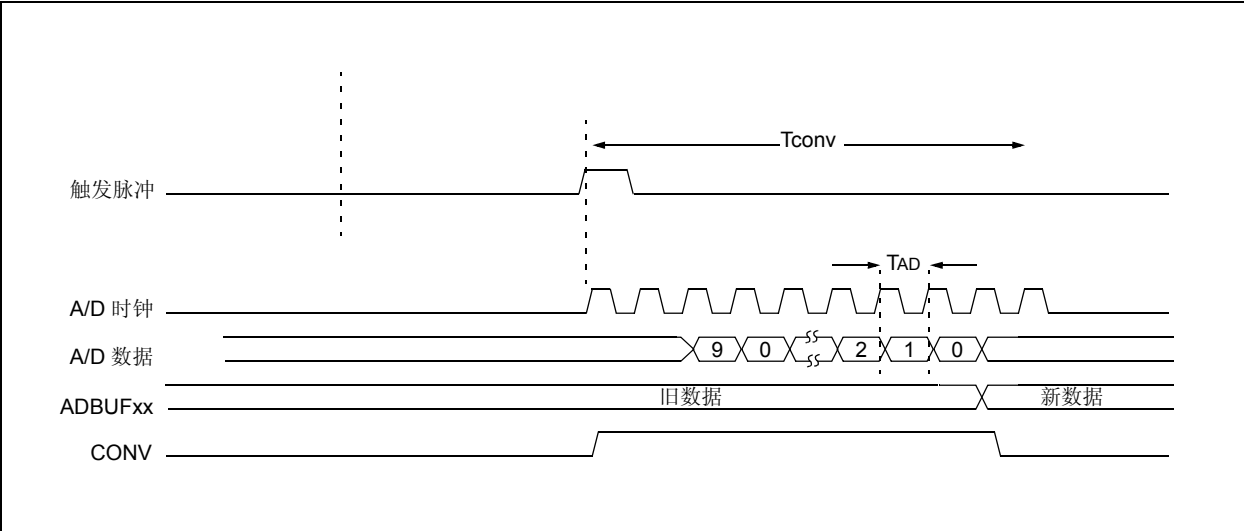


表 21-34: 比较器工作条件

符号	特性	最小值	典型值	最大值	单位	备注
VDD	电压范围	3.0	—	3.6	V	工作范围 3.0 V-3.6V
VDD	电压范围	4.5	—	5.5	V	工作范围 4.5 V-5.5V
TEMP	温度范围	-40	—	105	°C	注意在这样的环境条件下，结温可以超过 125°C

表 21-35: 比较器 AC 和 DC 规范

		标准工作条件（除非另外声明） 工作温度：-40°C ≤ TA ≤ +105°C				
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	0		VDD - 1.5	V	
VGAIN	开环增益	90			db	
CMRR	共模抑制比	70			db	
TRESP	大信号响应		20	30	ns	当 V- 输入端保持在 AVDD/2 时，V+ 输入的阶跃响应为 100mv。测量的是从模拟输入引脚有信号到 PWM 输出引脚有输出之间的延时。

表 21-36: DAC DC 规范

		标准工作条件（除非另外声明） 工作温度：-40°C ≤ TA ≤ +105°C				
符号	特性	最小值	典型值	最大值	单位	备注
CVRSRC	输入参考电压	0		AVDD - 1.6	V	
CVRES	分辨率		10		位	
INL	传递函数精度	—	—	±1	LSB	AVDD = 5 V, DACREF = (AVDD/2) V
DNL	积分非线性误差	—	—	±0.8	LSB	
	微分非线性误差	—	—	±2	LSB	
	失调误差	—	—	±2.0	LSB	
	增益误差	—	—			

图注: TBD = 待定

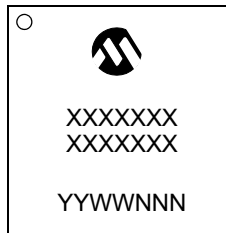
表 21-37: DAC AC 规范

		标准工作条件（除非另外声明） 工作温度：-40°C ≤ TA ≤ +125°C				
符号	特性	最小值	典型值	最大值	单位	备注
TSET	稳定时间			2.0	μs	是在 range = 1（高范围）且 cmref<9:0> 从 0x1FF 跳变到 0x300 时测得的。

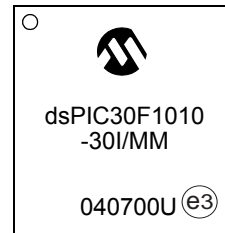
注:

22.0 封装标识信息

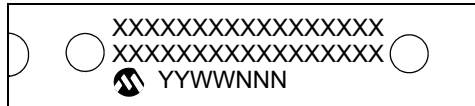
28 引脚 QFN-S



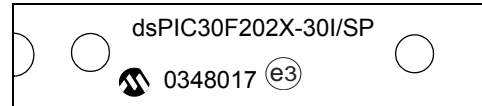
示例



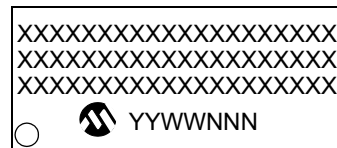
28 引脚 PDIP (窄条 DIP)



示例



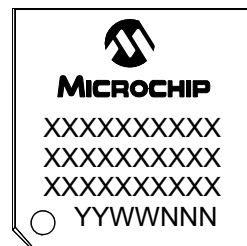
28 引脚 SOIC



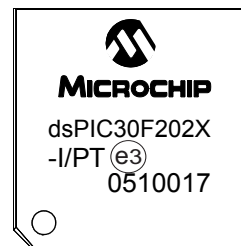
示例



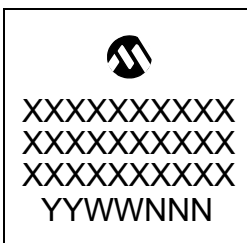
44 引脚 TQFP



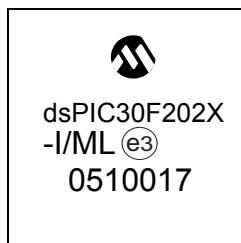
示例



44 引脚 QFN



示例

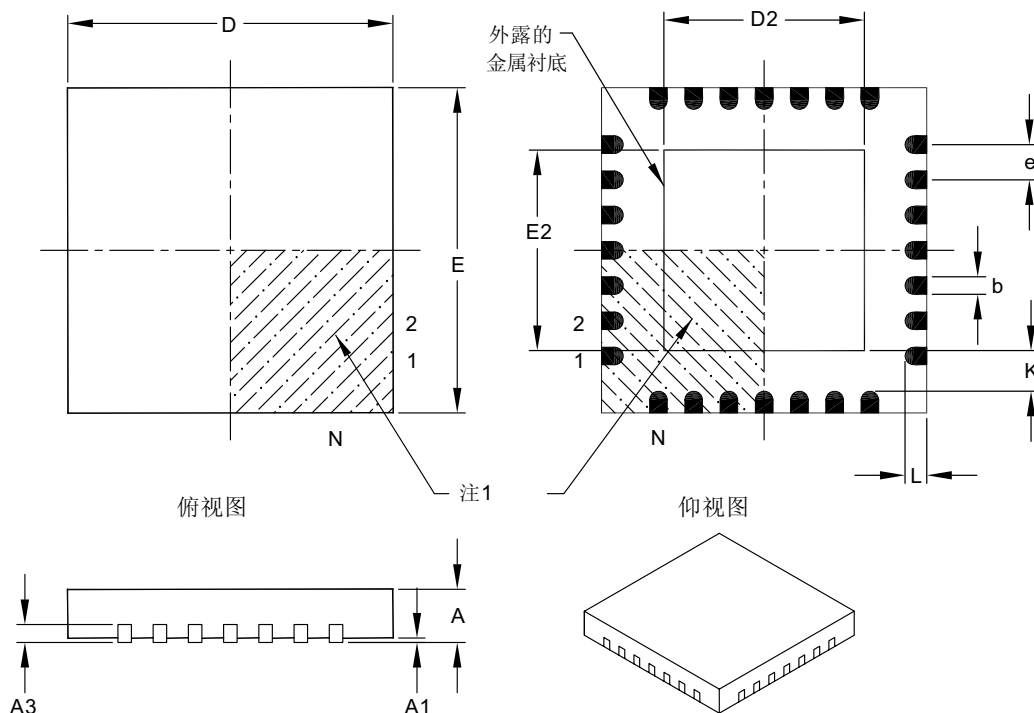


图注: XX...X 客户信息
 (e3) 年份代码 (日历年的最后一位数字)
 YY 年份代码 (日历年的最后两位数字)
 WW 星期代码 (一月一日的星期代码为“01”)
 NNN 以字母数字排序的追踪代码
 * 雾锡 (Matte Tin, Sn) 的 JEDEC 无铅标志
 表示无铅封装。JEDEC 无铅标志 ((e3)) 标示于
 此种封装的外包装上。

dsPIC30F1010/202X

28 引脚塑封四方扁平无引线封装（MM）——6x6x0.9 mm 主体（QFN-S）触点长度为 0.40 mm

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



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		
外露金属衬底宽度	E2	3.65	3.70	4.70
总长度	D	6.00 BSC		
外露金属衬底长度	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
触点到外露金属衬底的距离 §	K	0.20	—	—

注：

1. 引脚1的可视索引特征可能会有变化，但必须位于阴影区域内。
2. § 重要特性
3. 封装为切割分离
4. 尺寸和公差参考ASME Y14.5M

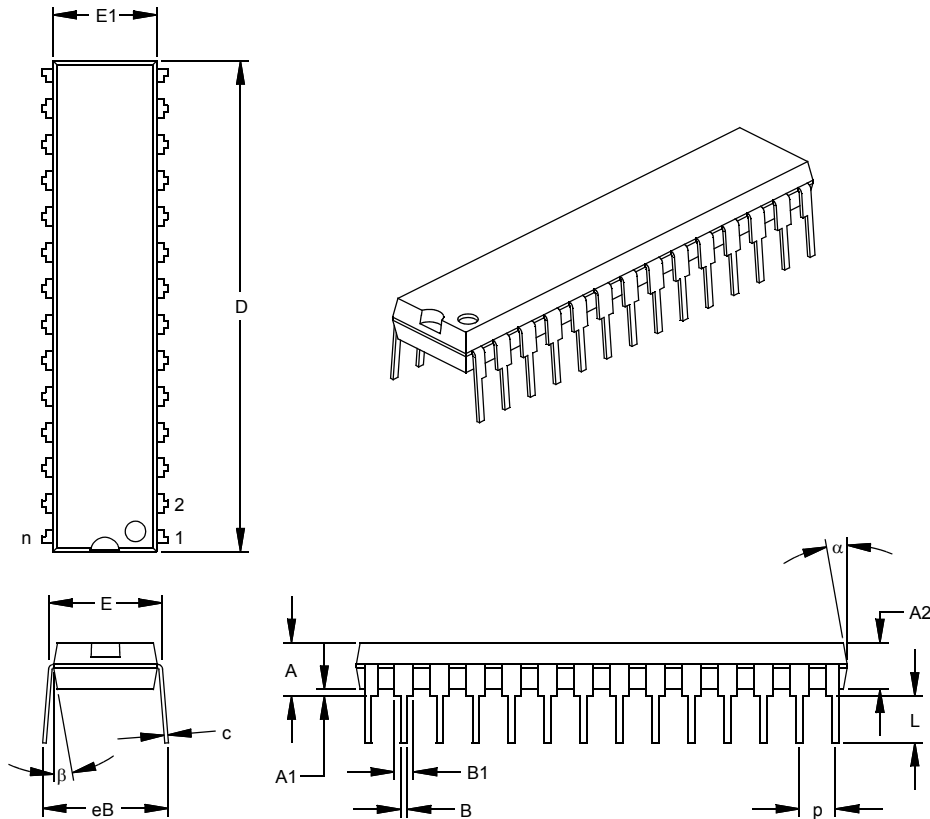
BSC：基本尺寸。给出的是理论上的精确值，没有公差。

REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology，图号：C04-124，2006年9月8日

28 引脚窄条塑封双列直插式封装（SP）——300 mil 主体（PDIP）

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	英寸*			毫米		
		最小	正常	最大	最小	正常	最大
引脚数	n	28			28		
引脚间距	p		.100			2.54	
顶端到固定面高度	A	.140	.150	.160	3.56	3.81	4.06
塑模封装厚度	A2	.125	.130	.135	3.18	3.30	3.43
底端到固定面高度	A1	.015			0.38		
肩到肩宽度	E	.300	.310	.325	7.62	7.87	8.26
塑模封装宽度	E1	.275	.285	.295	6.99	7.24	7.49
总长度	D	1.345	1.365	1.385	34.16	34.67	35.18
引脚尖到固定面高度	L	.125	.130	.135	3.18	3.30	3.43
引脚厚度	c	.008	.012	.015	0.20	0.29	0.38
引脚上部宽度	B1	.040	.053	.065	1.02	1.33	1.65
引脚下部宽度	B	.016	.019	.022	0.41	0.48	0.56
总排列间距	eB	.320	.350	.430	8.13	8.89	10.92
塑模顶部锥度	α	5	10	15	5	10	15
塑模底部锥度	β	5	10	15	5	10	15

* 控制参数

§ 重要特性

注：

尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸 (0.254mm)。

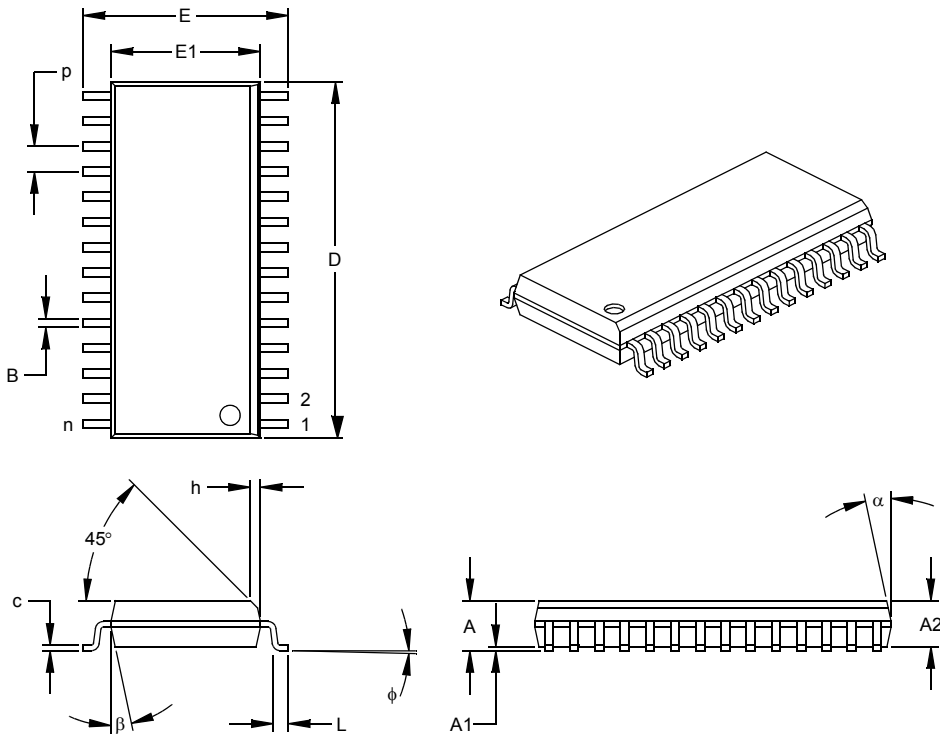
等同于 JEDEC 号：MO-095

图号：C04-070

dsPIC30F1010/202X

28 引脚塑封宽条小外形封装（SO）—— 300 mil 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		英寸*			毫米		
尺寸范围		最小	正常	最大	最小	正常	最大
引脚数	n	28			28		
引脚间距	p		.050			1.27	
总高度	A	.093	.099	.104	2.36	2.50	2.64
塑模封装厚度	A2	.088	.091	.094	2.24	2.31	2.39
悬空间隙	A1	.004	.008	.012	0.10	0.20	0.30
总宽度	E	.394	.407	.420	10.01	10.34	10.67
塑模封装宽度	E1	.288	.295	.299	7.32	7.49	7.59
总长度	D	.695	.704	.712	17.65	17.87	18.08
斜面投影距离	h	.010	.020	.029	0.25	0.50	0.74
底脚长度	L	.016	.033	.050	0.41	0.84	1.27
底脚倾角	φ	0	4	8	0	4	8
引脚厚度	c	.009	.011	.013	0.23	0.28	0.33
引脚宽度	B	.014	.017	.020	0.36	0.42	0.51
塑模顶部锥度	α	0	12	15	0	12	15
塑模底部锥度	β	0	12	15	0	12	15

* 控制参数

§ 重要特性

注

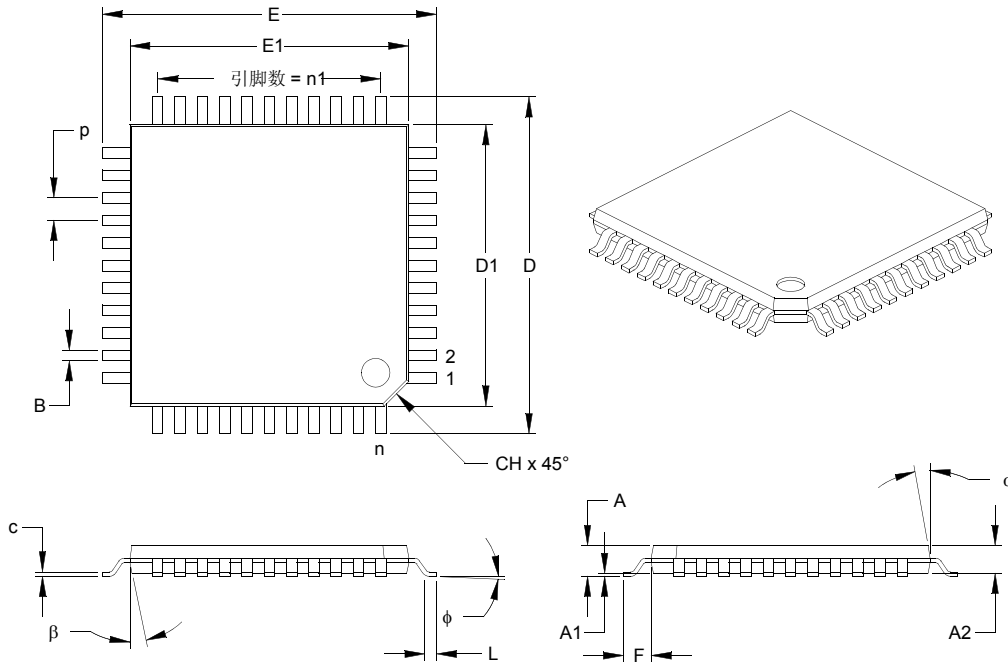
尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸（0.254mm）。

等同于 JEDEC 号：MS-013

图号：C04-052

44 引脚塑封薄型四方扁平封装（PT），10x10x1 mm 主体，1.0/0.10 mm 引脚形式（TQFP）

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	英寸			毫米*		
		最小	正常	最大	最小	正常	最大
引脚数	n	44			44		
引脚间距	p	.031			0.80		
每边引脚数	n1	11			11		
总高度	A	.039	.043	.047	1.00	1.10	1.20
塑模封装厚度	A2	.037	.039	.041	0.95	1.00	1.05
悬空间隙	A1	.002	.004	.006	0.05	0.10	0.15
底脚长度	L	.018	.024	.030	0.45	0.60	0.75
投影距离（参考）	F	.039 REF.			1.00 REF.		
底脚倾角	φ	0	3.5	7	0	3.5	7
总宽度	E	.463	.472	.482	11.75	12.00	12.25
总长度	D	.463	.472	.482	11.75	12.00	12.25
塑模封装宽度	E1	.390	.394	.398	9.90	10.00	10.10
塑模封装长度	D1	.390	.394	.398	9.90	10.00	10.10
引脚厚度	c	.004	.006	.008	0.09	0.15	0.20
引脚宽度	B	.012	.015	.017	0.30	0.38	0.44
引脚 1 切角斜面	CH	.025	.035	.045	0.64	0.89	1.14
塑模顶部锥度	α	5	10	15	5	10	15
塑模底部锥度	β	5	10	15	5	10	15

* 控制参数

注

尺寸 D 和 E1 不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过 0.010 英寸（0.254mm）。

REF: 参考尺寸，通常无公差，仅供参考。

参见 ASME Y14.5M

等同于 JEDEC 号: MS-026

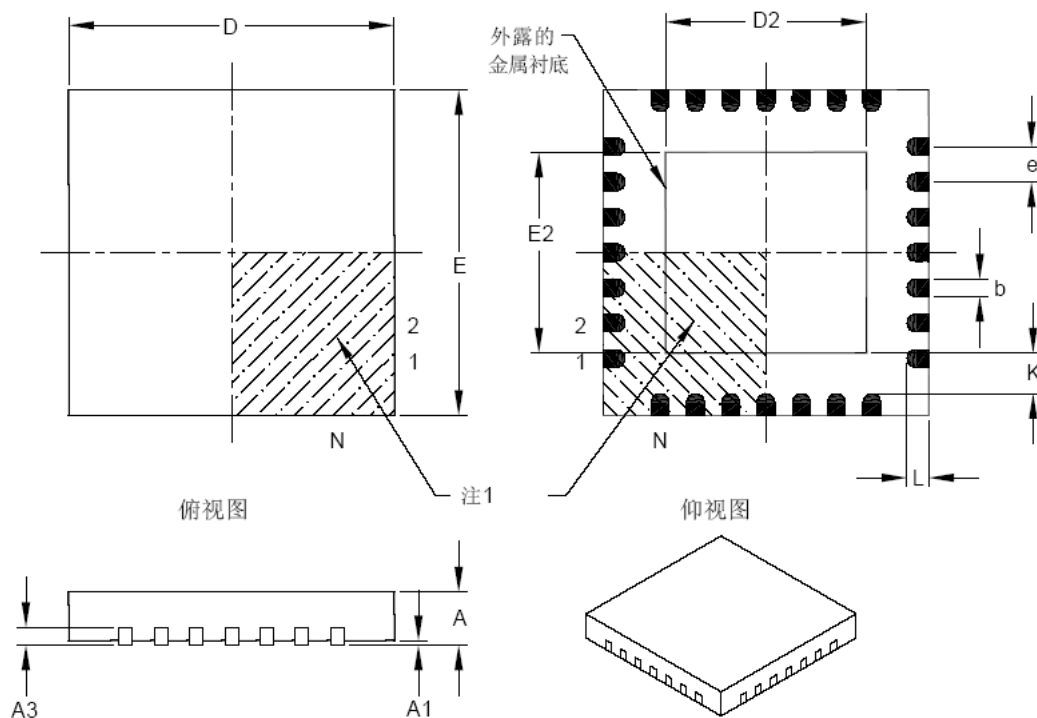
图号: C04-076

修改日期: 07-22-05

dsPIC30F1010/202X

44 引脚塑封四方扁平无引线封装（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		
外露金属衬底宽度	E2	6.30	6.45	6.80
总长度	D	8.00 BSC		
外露金属衬底长度	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
触点到外露金属衬底的距离 §	K	0.20	—	—

注：

1. 引脚1的可视索引特征可能会有变化，但必须位于阴影区域内。
2. § 重要特性
3. 封装为切割分离
4. 尺寸和公差参考ASME Y14.5M

BSC：基本尺寸。给出的是理论上的精确值，没有公差。

REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology, 图号: C04-103, 2006年9月8日

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。

dsPIC30F1010/202X

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: dsPIC30F1010/202X 文献编号: DS70178C_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?

附录 A： 版本历史

版本 A（2006 年 6 月）

- 本文档的初始版本。

版本 B（2006 年 8 月）

本版本包括：

第 5.0 节“中断”已经过更新包含了 INTTREG 寄存器。

器件配置寄存器已经过更新包含了 FBS 引导代码段和 FOSCEL 振荡器选择配置寄存器（见**第 18.0 节“系统集成”**）。

更新了电气特性：

- I_{IDLE} 参数 DC43f 最大值修改为 87 mA（见表 21-6）

排版错误校正：

- dsPIC30F1010/2020 端口寄存器（见表 6-1）
 - TRISA SFR bit 9 校正为 “TRISA9”
 - TRISE SFR 复位状态校正为
0000 0000 0000 0011
- dsPIC30F2023 端口寄存器（见表 6-2）
 - TRISA SFR bit 0 校正为 “未使用”
 - PORTA SFR bit 0 校正为 “未使用”
 - LATA SFR bit 0 校正为 “未使用”
 - TRISD SFR bit 0 校正为 “TRISD0”
 - PORTD SFR bit 0 校正为 “RD0”
 - LATD SFR bit 0 校正为 “LATD0”
 - TRISD SFR 复位状态校正为
0000 0000 0000 0011
- dsPIC30F1010/202X CNEN1 SFR 复位状态校正为 0000 0000 0000 0000（见表 6-3）
- PWMCONx（见寄存器 12-5）
 - Bit 13 内容校正为 “TRGSTAT”
 - Bit 10 内容校正为 “TRGIEN”
- ALTDTRx（见寄存器 12-9）
 - Bit 15-14 校正为 “未使用”
- ADCPC1（见寄存器 16-6）
 - TRGSRC2<4:0> 校正为包括 bit 4

版本 C（2006 年 11 月）

本版本包括：

更新了用于工业级和扩展级温度的 RC、EC 和 HS 晶振工作频率。

修改了 SPI 部分以反映最新的工作频率（见**第 13.0 节“串行外设接口（SPI）”**）。

修改了振荡器配置（见**第 18.3 节“振荡器配置”**）。

更新了电气特性：

- 供电电压参数 DC11 最小值更改为 3.0V（见表 21-4）。
- 工作电流（I_{DD}）（见表 21-5）
- 空闲电流（I_{IDLE}）（见表 21-6）
- I/O 引脚输入规范（见表 21-8）
- I/O 引脚输出规范（见表 21-9）
- 外部时钟时序（见图 21-2 和表 21-12）
- PLL 时钟时序（见表 21-13）
- 内部 RC 精度（见表 21-15）
- 上电延时定时器周期（见表 21-18）

注:

索引

数字

10 位高速模数转换器。参见 A/D

A

A/D	169
配置模拟端口	188
A/D 端口配置寄存器 (ADPCFG)	174
A/D 控制寄存器 (ADCON)	171
A/D 转换对控制寄存器 0 (ADCPC0)	175
A/D 转换对控制寄存器 1 (ADCPC1)	177
A/D 转换对控制寄存器 2 (ADCPC2)	179
A/D 状态寄存器 (ADSTAT)	173
AC 特性	240
负载条件	240
AC 温度和电压规范	240
ADC 寄存器映射	190

B

备用向量表	51
比较器 DAC 控制寄存器 (CMPDACx)	194
比较器控制寄存器 (CMPCONx)	193
编程操作	83
擦除程序存储器的一行	83
对闪存程序存储器编程的算法	83
启动编程序列	84
装载写锁存器	84
编程模型	20
图	21
编程, 器件指令	219
变更通知客户服务	273
波特率误差计算 (BRGH = 0)	162

C

C 编译器	
MPLAB C18	228
MPLAB C30	228
CLKO 和 I/O 时序	
特性	245
要求	245
CPU 空闲模式期间的输出比较操作	103
CPU 休眠模式期间的输出比较操作	103
操作码说明中使用的符号	220
产品标识体系	284
程序存储器和 EEPROM 特性	239
程序地址空间	29
表指令	
TBLRDH	31
TBLRDL	31
TBLWTH	31
TBLWTL	31
存储器映射	29
构成	30
使用表指令访问程序存储器中的数据	31
访问程序空间中数据的地址生成	30
程序计数器	20
使用程序空间可视性访问	
程序存储器中的数据	32
数据空间通过程序空间可视性	
映射到程序空间的操作	33
程序数据表访问	32
除法支持	22
串行外设接口 (SPI)	145
从休眠和空闲模式唤醒	51

从休眠中唤醒	197
存储器构成	29

D

DC 特性	
程序存储器和 EEPROM	239
掉电电流 (IPD)	237
工作电流 (IDD)	233
I/O 引脚输出规范	239
I/O 引脚输入规范	238
空闲电流 (IDLE)	235
dsPIC30F2020 框图	13
DSP 引擎	23
乘法器	25
代码保护	197
代码示例	
擦除程序存储器的一行	83
启动编程序列	84
装载写锁存器	84
带隙启动时间	
时序特性	248
要求	248
地址发生器单元	41
电气特性	231
AC	240
电源 PWM	107
电源 PWM 寄存器映射	142
电源 PWM 模块	
时序要求	253
掉电电流 (IPD)	237
读者反馈表	274
端口读 / 写示例	78
端口寄存器映射 (dsPIC30F2023)	80
端口寄存器映射 (dsPIC30F1010/2020)	79

F

封装信息	
标识	267
复位	197, 209
具有长晶振起振时间的 POR	212
POR	211
POR, FSCM 和 PWRT 禁止时的工作	212
复位过程	49
复位源	49
复位时序特性	246
复位时序要求	247
负载条件	240

G

公式	
BRGH = 1 时的 UART 波特率	162
BRGH = 0 时的 UART 波特率	162
I ² C	158
器件速率和 SPI 时钟速度之间的关系	148
工作电流 (IDD)	233
固件指令	219

H

汇编器	
MPASM 汇编器	228

I

I/O 端口

dsPIC30F1010/202X

并行 I/O (PIO)	77	INTCON2	54
I/O 引脚规范		INTTREG	74
输出	239	IOCONx	116
输入	239	IPC0	63
I ² C	153	IPC1	64
I ² C 10 位从模式的工作	155	IPC10	73
发送	155	IPC2	65
接收	155	IPC3	66
I ² C 7 位从模式的工作	155	IPC4	67
发送	155	IPC5	68
接收	155	IPC6	69
I ² C 寄存器映射	159	IPC7	70
I ² C 模块		IPC8	71
编程模型	153	IPC9	72
地址	155	LEBCONx	120
各种模式	153	LFSR	202
功能说明	153	MDC	112
广播呼叫地址支持	157	OSCCON	199
寄存器	153	OSCTUN	201
IPMI 支持	157	OSCTUN2	202
软件控制的时钟延长 (STREN = 1)	156	PHASEx	114
斜率控制	157	PTCON	110
引脚配置	153	PTPER	111
在 CPU 休眠和空闲模式期间的工作	158	PWMCONx	112
中断	156	SEVTCMP	111
主模式下的工作	157	SPIxCON1 (SPIx 控制寄存器 1)	150
主模式支持	157	SPIxCON2 (SPIx 控制寄存器 2)	151
总线启动 / 停止位时序特性		SPIxSTAT (SPIx 状态和控制寄存器)	149
从模式	261	TRGCONx	115
主模式	259	TRIGx	119
总线数据时序特性		U1MODE	164
从模式	261	U1STA	166
主模式	259		
总线数据时序要求		寄存器映射	
从模式	262	ADC 寄存器	190
主模式	260	电源 PWM 寄存器	142
I ² C 主模式		端口寄存器 (dsPIC30F1010/2020)	79
波特率发生器	158	端口寄存器 (dsPIC30F2023)	80
多主器件通信、总线冲突与总线仲裁	158	I ² C 寄存器	159
发送	157	模拟比较器控制寄存器	195
接收	157	内核寄存器	38
时钟仲裁	158	NVM 寄存器	85
Internet 地址	273	器件配置寄存器	218
		SPI1 寄存器	152
J		输出比较寄存器	105
寄存器		输入捕捉寄存器	100
ADCON	171	输入状态变化通知寄存器	80
ADCPC@	179	Timer 1 寄存器	90
ADCPC0	175	Timer2/3 寄存器	95
ADCPC1	177	UART1 寄存器	168
ADPCFG	174	系统集成寄存器 (dsPIC30F202X)	218
ADSTAT	173	中断控制器寄存器	75
ALTDTRx	115	简单捕捉事件模式	
CMPCONx	193	捕捉缓冲器操作	98
CMPDACx	194	捕捉预分频器	98
DTRx	114	CPU 空闲模式下的输入捕捉	99
FCLCONx	117	霍尔传感器模式	98
FOSC	204	Timer2 和 Timer3 选择模式	98
FOSCSEL	203	简单 PWM 模式	102
IEC1	61	周期	103
IEC2	62	简单输出比较 / PWM 模式时序要求	252
IFS1	57	简单输出比较匹配模式	102
IFS2	58	节能模式	214
IFSO	55	空闲	215
INTCON1	52	休眠	214
		节能模式 (休眠和空闲)	197

K

开发支持	227
看门狗定时器	
时序特性	246
时序要求	247
看门狗定时器 (WDT)	197, 214
工作	214
使能和禁止	214
勘误表	8
客户通知服务	273
客户支持	273
空闲电流 (I _{IDLE})	235
控制寄存器	
NVMADR	82
NVMADRU	82
NVMCON	82
NVMKEY	82
快速现场保护	51
框图	
16 位 Timer1 模块	88
dsPIC30F1010	10
dsPIC30F2020	13
dsPIC30F2023	16
DSP 引擎	24
复位系统	210
共用端口结构	77
I ² C	154
SPI	146
输出比较模式	101
输入捕捉模式	97
UART	161
外部上电复位电路	212
振荡器系统	198

M

MPLAB ASM30 汇编器、链接器和库管理器	228
MPLAB ICD 2 在线调试器	229
MPLAB ICE 2000 高性能通用在线仿真器	229
MPLAB PM3 器件编程器	229
MPLAB REAL ICE 在线仿真器系统	229
MPLAB 集成开发环境软件	227
MPLINK 目标链接器 / MPLIB 目标库管理器	228
模拟比较器控制寄存器映射	195
模寻址	43
操作示例	44
起始和结束地址	43
W 地址寄存器选择	43
应用	45

N

NVM 寄存器映射	85
内部时钟时序示例	242
内核架构	
概述	19

P

PICSTART 2 开发编程器	230
PICSTART Plus 开发编程器	230
PLL 时钟时序规范	242
POR. 参见上电复位	
PWM I/O 控制寄存器 (IOCONx)	116
PWM 备用死区时间寄存器 (ALTDTRx)	115
PWM 触发比较值寄存器 ((TRIGx)	119
PWM 触发控制寄存器 (TRGCONx)	115

PWM 故障电流限制控制寄存器 (FCLCONx)	117
PWM 控制寄存器 (PWMCONx)	112
PWM 时基控制寄存器 (PTCON)	110
PWM 死区时间寄存器 (DTRx)	114
PWM 相移寄存器 (PHASEx)	114
PWM 主占空比寄存器 (MDC)	112
配置模拟端口引脚	78

Q

器件概述	9
器件 ID 单元	197
器件配置寄存器	215
器件配置寄存器映射	218
前沿消隐控制寄存器 (LEBCONx)	120

R

RCON 寄存器的初始状态, 情形 1	213
RCON 寄存器的初始状态, 情形 2	213
RTSP 操作	82
软件堆栈指针, 帧指针	20
CALL 堆栈帧	37
软件模拟器 (MPLAB SIM)	228

S

SPI	
从模式、帧主模式连接	148
从模式、帧从模式连接	148
主 / 从连接	147
主模式、帧主模式连接	147
SPI1 寄存器映射	152
SPI 模式	
SPI1 寄存器映射	152
SPI 模块	
时序特性	
从模式 (CKE = 1)	256, 257
主模式 (CKE = 0)	254
主模式 (CKE = 1)	255
时序要求	
从模式 (CKE = 0)	256
从模式 (CKE = 1)	258
主模式 (CKE = 0)	254
主模式 (CKE = 1)	255
闪存程序存储器	81
表指令操作汇总	81
运行时自编程 (RTSP)	81
在线串行编程 (ICSP)	81
上电复位 (POR)	197
上电延时定时器 (PWRT)	197
振荡器起振定时器 (OST)	197
上电延时定时器	
时序特性	246
时序要求	247
时序规范	
PLL 时钟	242
时序图	
PWM 输出	104
上电过程中的延时时序 ($\overline{\text{MCLR}}$ 接至 V _{DD})	211
上电过程中的延时时序	
(MCLR 未接至 V _{DD}), 情形 1	211
上电过程中的延时时序	
(MCLR 未接至 V _{DD}), 情形 2	212
时序特性	
A/D 转换	
10 位高速 A/D 转换时序特性 (CHPS = 01, SIMSAM = 0, ASAM = 0, SSR = 000)	264
A、B 和 C 类定时器外部时钟	249

dsPIC30F1010/202X

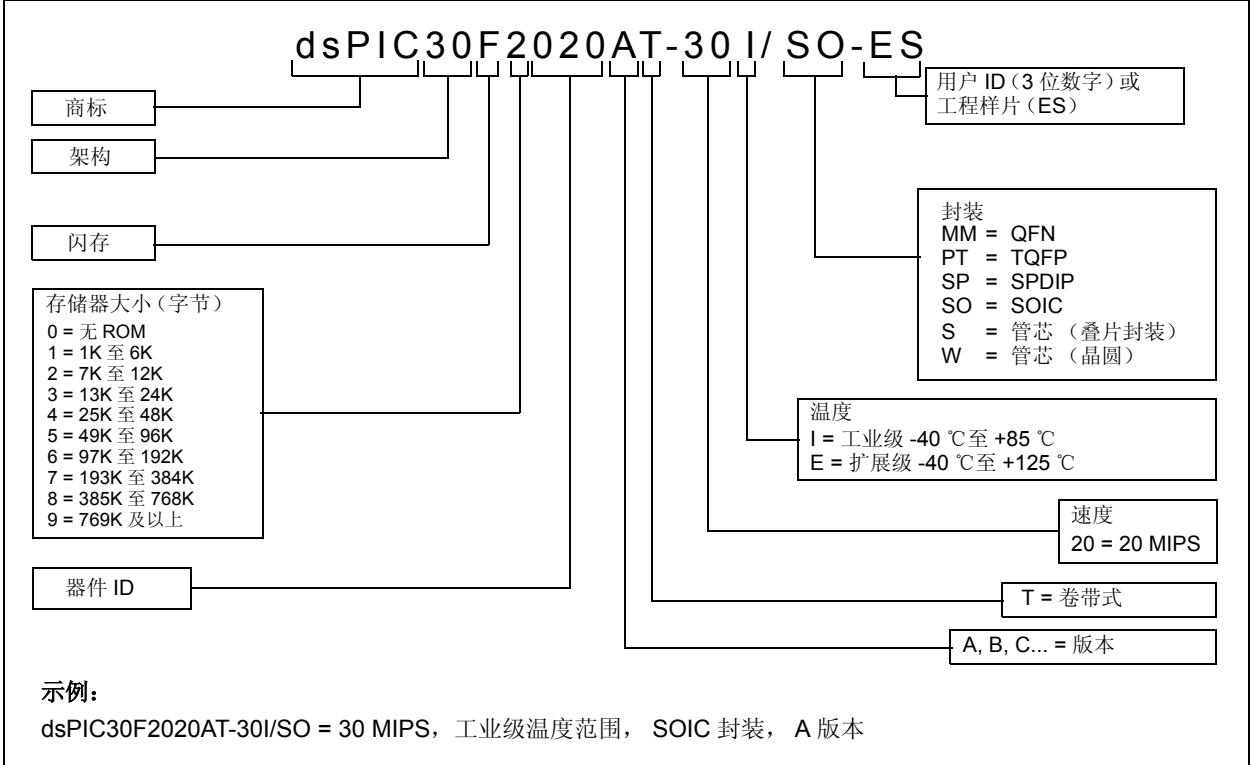
CLKO 和 I/O	245	Near 数据空间	37
带隙启动时间	248	软件堆栈	37
电机控制 PWM 模块	253	数据累加器和加法器 / 减法器	25
电机控制 PWM 模块故障	253	回写	26
复位	246	舍入逻辑	26
I ² C 总线启动 / 停止位		数据空间写饱和	27
从模式	261	溢出和饱和	25
主模式	259	使用程序空间可视性访问程序存储器中的数据	32
I ² C 总线数据		输入捕捉寄存器映射	100
从模式	261	输入捕捉时序要求	251
主模式	259	输入捕捉中断	99
看门狗定时器	246	输入捕捉 (CAPx) 时序特性	251
SPI 模块		输入状态变化通知寄存器映射	80
从模式 (CKE = 0)	256	输入状态变化通知	78
从模式 (CKE = 1)	257	双输出比较匹配模式	102
主模式 (CKE = 0)	254	单脉冲模式	102
主模式 (CKE = 1)	255	连续脉冲模式	102
上电延时定时器	246	T	
输出比较 /PWM 模块	252	Timer2/3 寄存器映射	95
输出比较模块	251	Timer2/3 模块	91
输入捕捉 (CAPx)	251	16 位定时器模式	91
外部时钟	240	32 位定时器模式	91
振荡器起振定时器	246	32 位同步计数器模式	91
时序图和规范		ADC 事件触发器	94
DC 特性 —— 内部 RC 精度	242	定时器预分频器	94
时序图。见时序特性		门控操作	94
时序要求		休眠模式下的定时器操作	94
A 类定时器外部时钟	249	中断	94
B 类定时器外部时钟	250	Timer2 和 Timer3 选择模式	102
CLKO 和 I/O	245	Timer1 寄存器映射	90
C 类定时器外部时钟	250	Timer1 模块	87
带隙启动时间	248	16 位定时器模式	87
电机控制 PWM 模块	253	16 位同步计数器模式	87
复位	247	16 位异步计数器模式	87
I ² C 总线数据 (从模式)	262	门控操作	88
I ² C 总线数据 (主模式)	260	休眠模式下的操作	88
简单输出比较 /PWM 模式	252	预分频器	88
看门狗定时器	247	中断	89
欠压复位	247	特殊事件比较寄存器 (SEVTCMP)	111
SPI 模块		桶形移位寄存器	27
从模式 (CKE = 0)	256	通用异步收发器。见 UART	
从模式 (CKE = 1)	258	U	
主模式 (CKE = 0)	254	UART	
主模式 (CKE = 1)	255	波特率发生器 (BRG)	162
上电延时定时器	247	发送	
输出比较模块	251	8 位数据模式	163
输入捕捉	251	9 位数据模式	163
外部时钟	241	间隔和同步序列	163
振荡器起振定时器	247	IrDA	
输出比较 /PWM 模块时序特性	252	内置编码器和解码器	163
输出比较寄存器映射	105	接收	
输出比较模块	101	8 位或 9 位数据模式	163
时序特性	251	使能和设置 UART	162
时序要求	251	UART1 寄存器映射	168
输出比较中断	104	UART1 模式寄存器 (U1MODE)	164
输入捕捉模块	97	UART1 状态和控制寄存器 (U1STA)	166
简单捕捉事件模式	98	W	
休眠和空闲模式	99	WWW 在线支持	8
数据地址空间		WWW 地址	273
存储器映射	33, 34	外部时钟时序特性	
对齐	36	A、B 和 C 类定时器	249
对齐 (图)		外部时钟输入	207
空间	36	外部时钟时序要求	241
宽度	36		
MCU 和 DSP (MAC 类) 指令	35		

A 类定时器	249	中断控制寄存器 1 (INTCON1)	52
B 类定时器	250	中断控制寄存器 2 (INTCON2)	54
C 类定时器	250	中断优先级控制寄存器 0 (IPC0)	63
外部中断请求	51	中断优先级控制寄存器 10 (IPC10)	73
位反转寻址	45	中断优先级控制寄存器 1 (IPC1)	64
示例	45	中断优先级控制寄存器 2 (IPC2)	65
实现	45	中断优先级控制寄存器 3 (IPC3)	66
修改量表	46	中断优先级控制寄存器 4 (IPC4)	67
序列表 (16 个项)	46	中断优先级控制寄存器 5 (IPC5)	68
温度和电压规范		中断优先级控制寄存器 6 (IPC6)	69
AC	240	中断优先级控制寄存器 7 (IPC7)	70
X		中断优先级控制寄存器 8 (IPC8)	71
系统集成	197	中断优先级控制寄存器 9 (IPC9)	72
系统集成寄存器 (dsPIC30F202X)	218	中断标志状态寄存器 0 (IFS0)	55
陷阱		中断标志状态寄存器 1 (IFS1)	57
陷阱源	49	中断标志状态寄存器 2 (IFS2)	58
线性反馈寄存器 (LFSR)	202	主时基寄存器 (PTPER)	111
销售和支持	284	状态寄存器	20
Y		自动时钟延长	156
引脚说明	11, 14, 17	10 位寻址模式 (STREN = 1)	156
Z		7 位寻址模式 (STREN = 1)	156
在 CPU 休眠模式下的	97	发送模式	156
简单捕捉事件模式	98	接收模式	156
休眠和空闲。	99		
在线串行编程 (ICSP) 功能	197		
在线调试器	217		
振荡器			
系统概述	197		
振荡器调节寄存器 2 (OSCTUN2)	202		
振荡器调节寄存器 (OSCTUN)	201		
振荡器控制寄存器 (OSCCON)	199		
振荡器配置	205		
初始时钟源选择	206		
故障保护时钟监视器	208		
锁相环 (PLL)	206		
振荡器起振定时器 (OST)	206		
振荡器起振定时器			
时序特性	246		
时序要求	247		
振荡器选择	197		
振荡器选择配置位 (FOSCSEL)	203		
振荡器选择配置位 (FOSC)	204		
指令集	219		
指令集概述	222		
指令寻址模式	41		
MAC 指令	42		
MCU 指令	42		
其他指令	42		
文件寄存器指令	41		
支持的基本模式	41		
传送和累加器指令	42		
中断	47		
陷阱	49		
中断标志状态寄存器 0 (IFS0)	55		
中断标志状态寄存器 1 (IFS1)	57		
中断标志状态寄存器 2 (IFS2)	58		
中断过程	51		
中断堆栈帧	51		
中断控制和状态寄存器 (INTTREG)	74		
中断控制器寄存器映射	75		
中断优先级	48		
中断允许控制寄存器 1 (IEC1)	61		
中断允许控制寄存器 2 (IEC2)	62		

注：

产品标识体系

欲订货，或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。





MICROCHIP

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 - 福州
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

中国 - 香港特别行政区
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 - 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 - 顺德
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

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

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

台湾地区 - 高雄
Tel: 886-7-536-4818
Fax: 886-7-536-4803

台湾地区 - 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-572-9526
Fax: 886-3-572-6459

亚太地区

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-4182-8400
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

10/05/07