



dsPIC30F5011/5013

数据手册

高性能 16 位
数字信号控制器

请注意以下有关 Microchip 器件代码保护功能的要点：

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

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

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

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

商标

Microchip 的名称和徽标组合、Microchip 徽标、Accuron、dsPIC、KEELOQ、KEELOQ 徽标、microID、MPLAB、PIC、PICmicro、PICSTART、PRO MATE、rfPIC 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、rfLAB、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、非易失性存储器和模拟产品方面的质量体系流程均符合 ISO/TS-16949:2002。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。



dsPIC30F5011/5013

dsPIC30F5011/5013 高性能 数字信号控制器

注: 本数据手册总结了 dsPIC30F 系列器件的功能, 但是不应把本手册当作无所不包的参考手册来使用。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)。

高性能改进型 RISC CPU:

- 改进的哈佛架构
- 优化的 C 编译器指令集架构
- 灵活的寻址模式
- 83 条基本指令
- 24 位宽指令, 16 位宽数据总线
- 66 KB 片上闪存程序空间
- 4 KB 片上数据 RAM
- 1 KB 非易失性数据 EEPROM
- 16 x 16 位工作寄存器阵列
- 最高 30 MIPS 的工作速度:
 - DC 至 40 MHz 外部时钟输入
 - 4 MHz - 10 MHz 振荡器输入, 带 PLL (4 倍频、8 倍频和 16 倍频)
- 最多 41 个中断源:
 - 8 个可由用户选择的中断优先级
 - 5 个外部中断源
 - 4 个处理器陷阱

DSP 特性:

- 双数据取操作
- 模寻址和位反转寻址模式
- 两个 40 位宽的累加器, 具备可选饱和逻辑
- 17 位 x 17 位单周期硬件小数 / 整数乘法器
- 所有 DSP 指令均为单周期指令
 - 乘 - 累加 (MAC) 操作
- 在一个周期内可将数据左右移位 16 位

外设特性:

- 高灌 / 拉电流 I/O 引脚: 25 mA/25 mA
- 5 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
- 16 位输入捕捉功能
- 16 位比较 /PWM 输出功能
- 数据转换器接口 (Data Converter Interface, DCI) 支持常见音频编解码器协议, 包括 I²S 和 AC'97
- 3 线 SPI 模块 (支持 4 种帧模式)
- I²CTM 模块支持多主器件 / 从模式, 支持 7 位 / 10 位寻址
- 两个带 FIFO 缓冲区的可寻址 UART 模块
- 两个与 CAN 2.0B 标准兼容的 CAN 总线模块

模拟特性:

- 12 位模数转换器 (Analog-to-Digital Converter, ADC) 具有以下特性:
 - 转换速率为 200 ksps
 - 最多 16 路输入通道
 - 在休眠和空闲期间可以进行转换
- 可编程低电压检测 (Programmable Low-Voltage Detection, PLVD)
- 可编程欠压检测和复位产生

单片机特性:

- 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
- 数据 EEPROM 存储器:
 - 对于工业级温度范围, 最少擦写次数 10 万次, 典型擦写次数 100 万次
- 可在软件控制下自行再编程
- 上电复位 (Power-on Reset, POR)、上电延时定时器 (Power-up Timer, PWRT) 和振荡器起振定时器 (Oscillator Start-up Timer, OST)
- 灵活的看门狗定时器 (Watchdog Timer, WDT), 带片上低功耗 RC 振荡器以便可靠地工作

dsPIC30F5011/5013

单片机特性（续）：

- 故障保护时钟监视器操作：
 - 检测时钟故障并切换到片上低功耗 RC 振荡器
- 可编程代码保护
- 在线串行编程（In-Circuit Serial Programming, ICSP）功能
- 可选择的功耗管理模式：
 - 休眠、空闲和备用时钟模式

CMOS 技术：

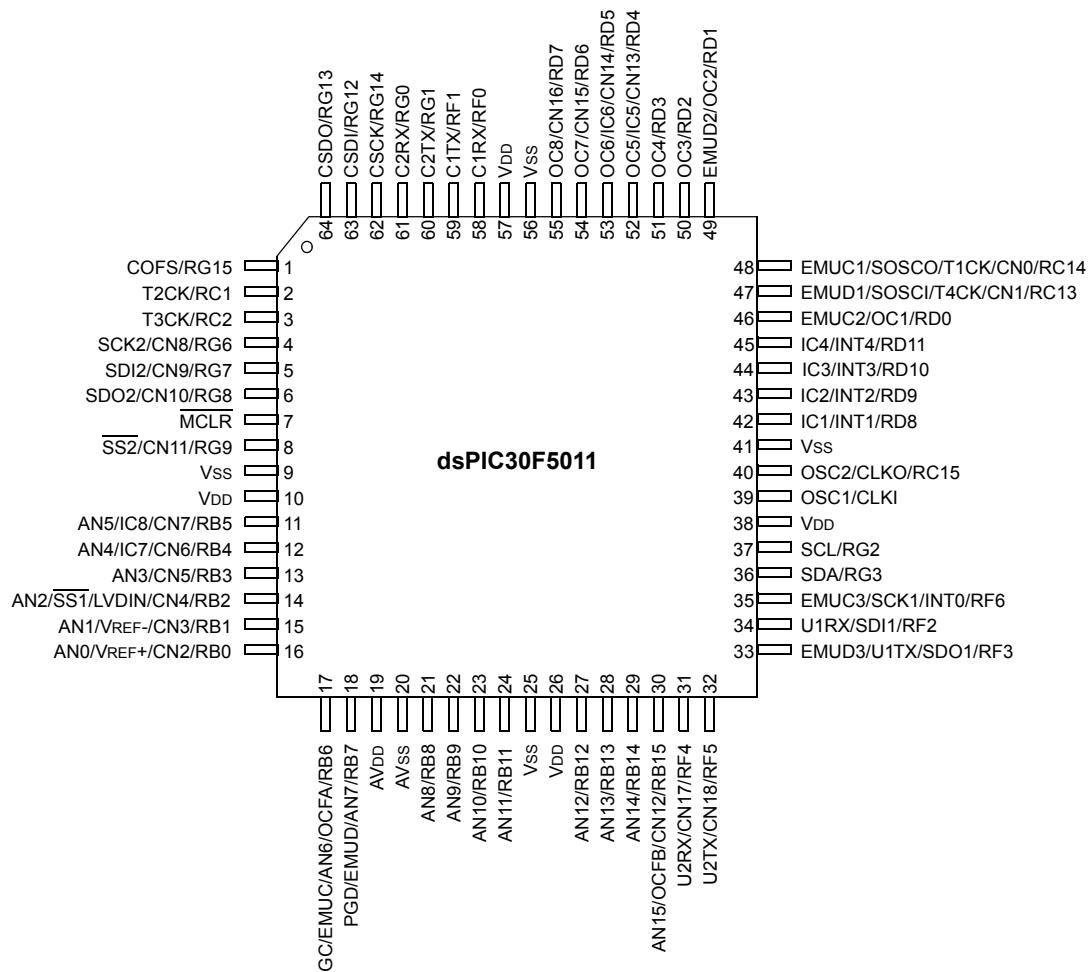
- 低功耗高速闪存技术
- 宽工作电压范围（2.5V 至 5.5V）
- 工业级和扩展级温度范围
- 低功耗

dsPIC30F5011/5013 控制器系列

器件	引脚	程序存储器		SRAM 字节	EEPROM 字节	16 位 定时器	输入 捕捉	输出比较 / 标准 PWM	编解码器 接口	12 位 A/D 200 Msps	UART	SPI	I ² C TM	CAN
		字节	指令											
dsPIC30F5011	64	66K	22K	4096	1024	5	8	8	AC97, I ² S	16 通道	2	2	1	2
dsPIC30F5013	80	66K	22K	4096	1024	5	8	8	AC97, I ² S	16 通道	2	2	1	2

引脚示意图

64 引脚 TQFP

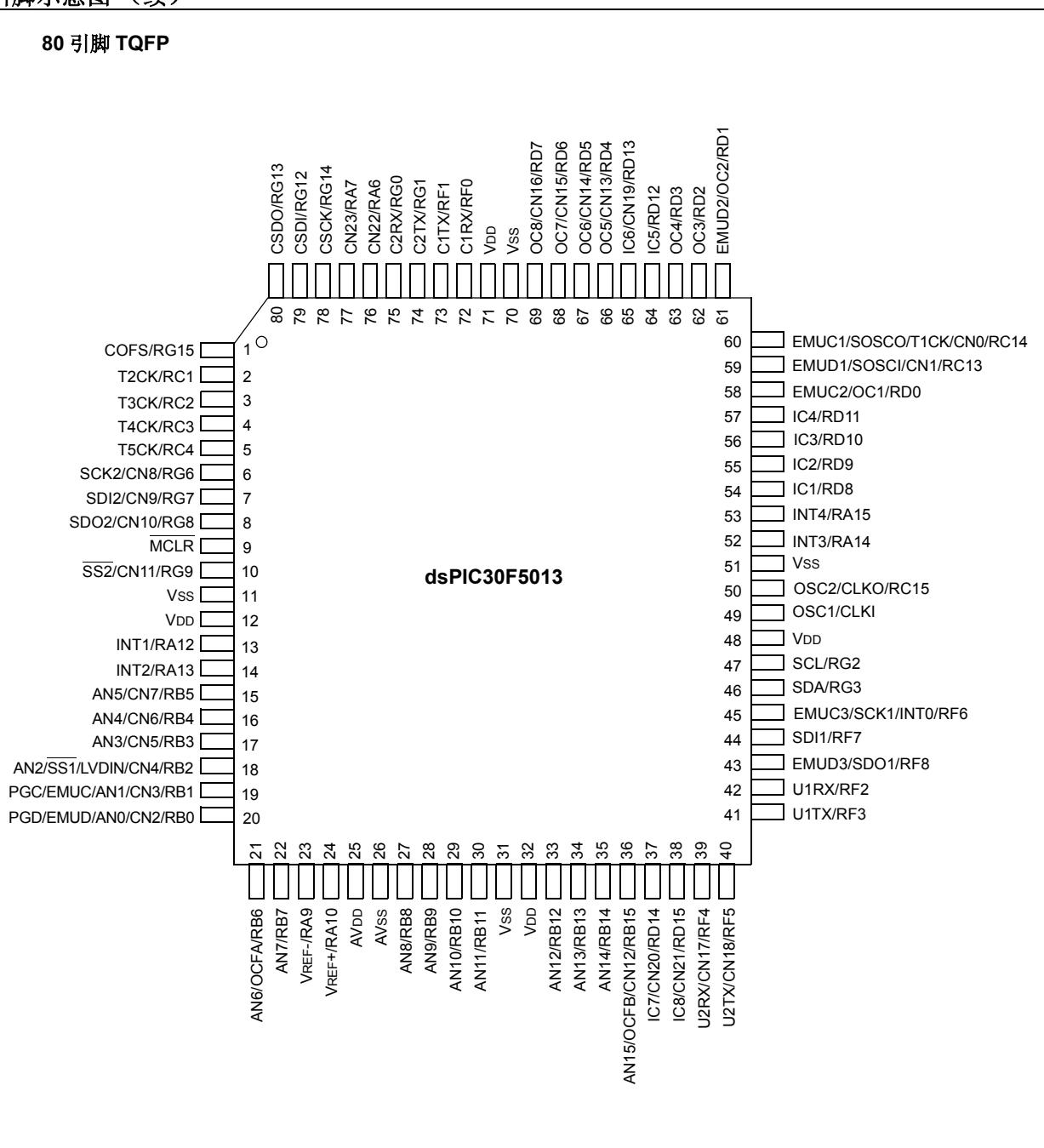


注：有关各个引脚的说明，请参见第 1.0 节“器件概述”。

dsPIC30F5011/5013

引脚示意图（续）

80 引脚 TQFP



注：有关各个引脚的说明，请参见第 1.0 节“器件概述”。

目录

1.0 器件概述	7
2.0 CPU 架构概述	13
3.0 存储器构成	23
4.0 中断	35
5.0 地址发生器单元	41
6.0 闪存程序存储器	47
7.0 数据 EEPROM 存储器	53
8.0 I/O 端口	59
9.0 Timer1 模块	65
10.0 Timer2/3 模块	69
11.0 Timer4/5 模块	75
12.0 输入捕捉模块	79
13.0 输出比较模块	83
14.0 SPI 模块	87
15.0 I ² C TM 模块	91
16.0 通用异步收发器 (UART) 模块	99
17.0 CAN 模块	107
18.0 数据转换器接口 (DCI) 模块	119
19.0 12 位模数转换器 (ADC) 模块	129
20.0 系统集成	139
21.0 指令集汇总	155
22.0 开发支持	163
23.0 电气特性	167
24.0 封装信息	207
索引	213
Microchip 网站	219
变更通知客户服务	219
客户支持	219
读者反馈表	220
产品标识体系	221

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,但是不应把本手册当作无所不包的参考手册来使用。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 dsPIC30F5011/5013 数字信号控制器 (Digital Signal Controller, DSC) 器件的特定信息。dsPIC30F5011/5013 器件在高性能16位单片机 (MCU) 架构中融合了大量数字信号处理器 (Digital Signal Processor, DSP) 功能。图 1-1 和图 1-2 分别给出了 dsPIC30F5011 和 dsPIC30F5013 器件的框图。

dsPIC30F5011/5013

图 1-1: dsPIC30F5011 框图

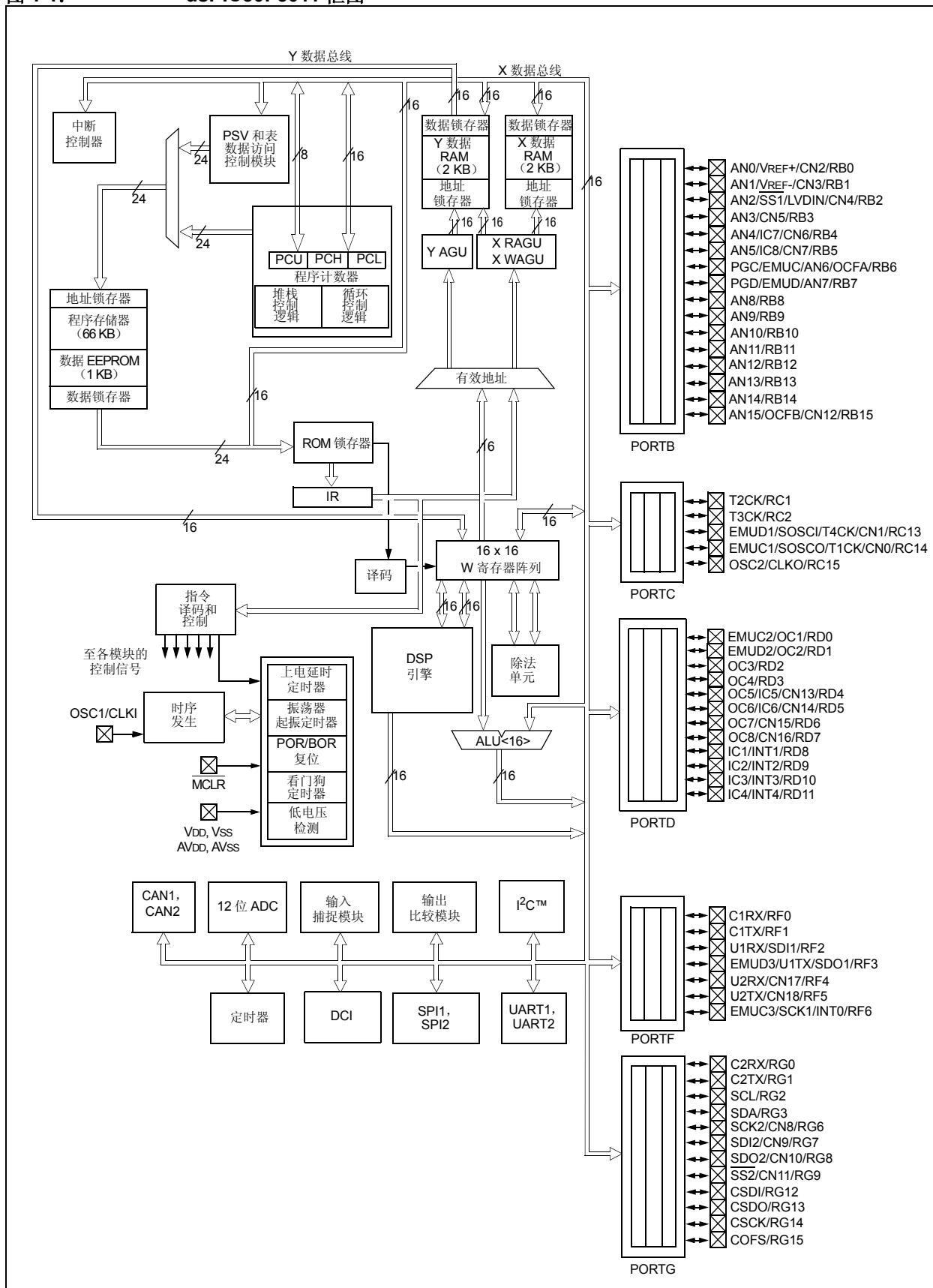
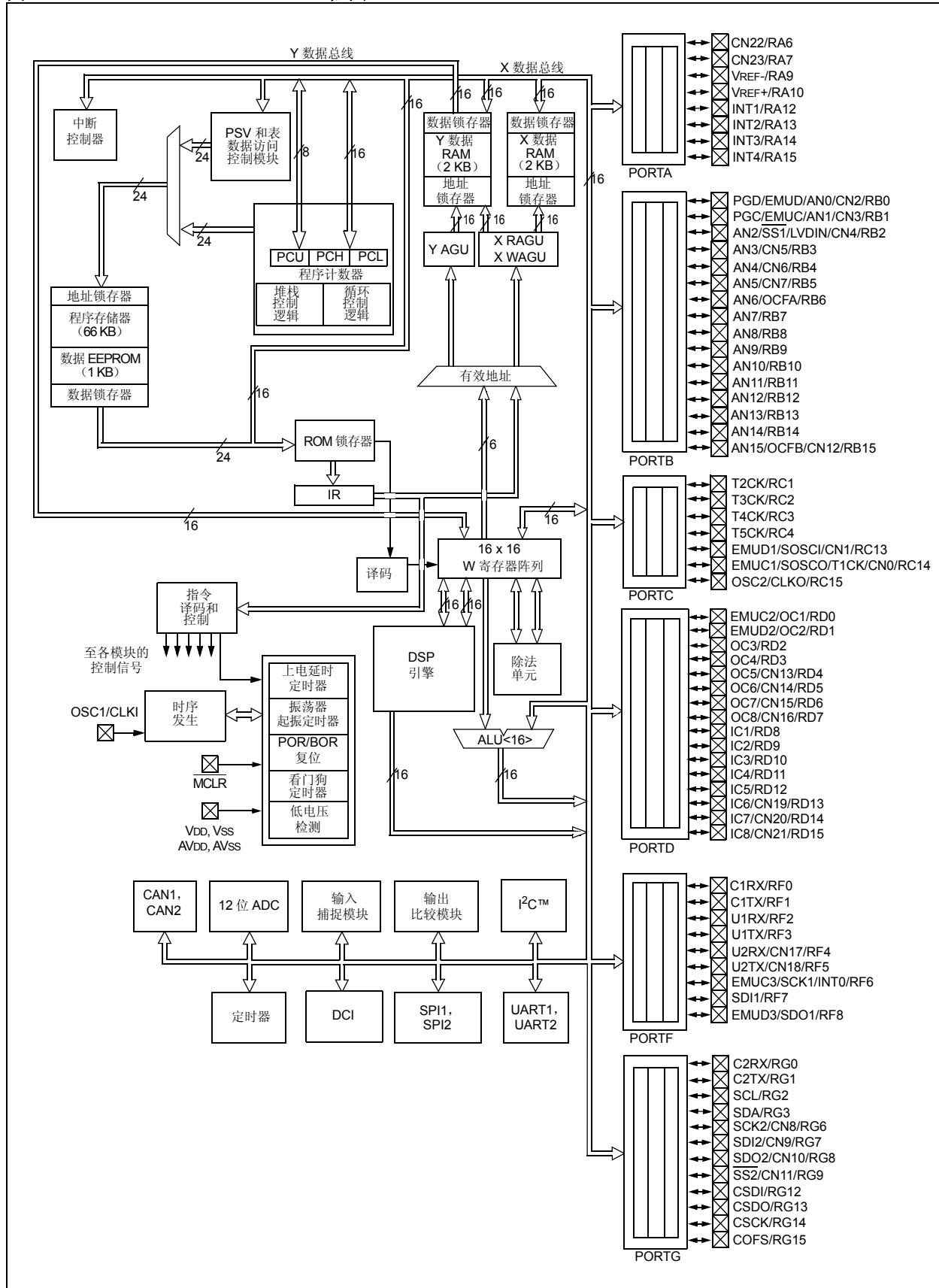


图 1-2: dsPIC30F5013 框图



dsPIC30F5011/5013

表 1-1 对器件的引脚配置和端口引脚的复用功能进行了简要说明。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改写端口引脚的数据方向。

表 1-1：引脚说明

引脚名称	引脚类型	缓冲器类型	说明
AN0-AN15	I	Analog	模拟输入通道。 AN0 和 AN1 也分别用于器件编程数据和时钟输入。
AVDD	P	P	模拟模块的正电源。
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。 晶振输出。在晶振模式下，连接到晶体或谐振器。也可选择在 RC 和 EC 模式下用作 CLKO。总是与 OSC2 引脚功能相关联。
CN0-CN23	I	ST	输入电平变化通知输入。 可将所有输入软件编程为内部弱上拉。
COFS CSCK CSDI CSDO	I/O I/O I O	ST ST ST —	数据转换器接口帧同步引脚。 数据转换器接口串行时钟输入 / 输出引脚。 数据转换器接口串行数据输入引脚。 数据转换器接口串行数据输出引脚。
C1RX C1TX C2RX C2TX	I O I O	ST — ST —	CAN1 总线接收引脚。 CAN1 总线发送引脚。 CAN2 总线接收引脚。 CAN2 总线发送引脚。
EMUD EMUC EMUD1 EMUC1 EMUD2 EMUC2 EMUD3 EMUC3	I/O I/O I/O I/O I/O I/O I/O I/O	ST ST 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。 ICD 第四通信通道数据输入 / 输出引脚。 ICD 第四通信通道时钟输入 / 输出引脚。
IC1-IC8	I	ST	捕捉输入 1 至 8。
INT0 INT1 INT2 INT3 INT4	I	ST	外部中断 0。 外部中断 1。 外部中断 2。 外部中断 3。 外部中断 4。
LVDIN	I	Analog	低电压检测参考电压输入引脚。
MCLR	I/P	ST	主复位输入或编程电压输入。此引脚为低电平有效的器件复位输入端。
OCFA OCFB OC1-OC8	I I O	ST ST —	比较故障 A 输入（对于比较通道 1、2、3 和 4）。 比较故障 B 输入（对于比较通道 5、6、7 和 8）。 比较输出 1 至 8。

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

Analog = 模拟输入
O = 输出
P = 电源

表 1-1：引脚说明（续）

引脚名称	引脚类型	缓冲器类型	说明
OSC1	I	ST/CMOS	晶振输入。配置在 RC 模式时为 ST 缓冲器输入；否则为 CMOS 输入。
OSC2	I/O	—	晶振输出。在晶振模式下，连接到晶体或谐振器。也可选择在 RC 和 EC 模式下用作 CLKO。
PGD PGC	I/O I	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。
RA6-RA7 RA9-RA10 RA12-RA15	I/O I/O I/O	ST ST ST	PORTA 是双向 I/O 端口。
RB0-RB15	I/O	ST	PORTB 是双向 I/O 端口。
RC1-RC4 RC13-RC15	I/O I/O	ST ST	PORTC 是双向 I/O 端口。
RD0-RD15	I/O	ST	PORTD 是双向 I/O 端口。
RF0-RF8	I/O	ST	PORTF 是双向 I/O 端口。
RG0-RG3 RG6-RG9 RG12-RG15	I/O I/O I/O	ST ST ST	PORTG 是双向 I/O 端口。
SCK1 SDI1 SDO1 SS1 SCK2 SDI2 SDO2 SS2	I/O I O I I/O I O I	ST ST — ST ST ST — ST	SPI1 的同步串行时钟输入 / 输出。 SPI1 数据输入。 SPI1 数据输出。 SPI1 从同步。 SPI2 的同步串行时钟输入 / 输出。 SPI2 数据输入。 SPI2 数据输出。 SPI2 从同步。
SCL SDA	I/O I/O	ST ST	I ² C TM 的同步串行时钟输入 / 输出。 I ² C 的同步串行数据输入 / 输出。
SOSCO SOSCI	O I	— ST/CMOS	32 kHz 低功耗晶振输出。 32 kHz 低功耗晶振输入。 配置为 RC 模式时为 ST 缓冲器输入；否则为 CMOS 输入。
T1CK T2CK T3CK T4CK T5CK	I I I I I	ST ST ST ST ST	Timer1 外部时钟输入。 Timer2 外部时钟输入。 Timer3 外部时钟输入。 Timer4 外部时钟输入。 Timer5 外部时钟输入。
U1RX U1TX U1ARX U1ATX U2RX U2TX	I O I O I O	ST — ST — ST —	UART1 接收。 UART1 发送。 UART1 备用接收。 UART1 备用发送。 UART2 接收。 UART2 发送。
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。
VREF+	I	Analog	模拟参考电压（高）输入。
VREF-	I	Analog	模拟参考电压（低）输入。

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

Analog = 模拟输入
 O = 输出
 P = 电源

dsPIC30F5011/5013

注：

2.0 CPU 架构概述

注: 本数据手册总结了 dsPIC30F 系列器件的功能, 但是不应把本手册当作无所不包的参考手册来使用。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

本章概述了 dsPIC30F 的 CPU 架构。有关硬件和编程的更多信息, 请分别参见《dsPIC30F 系列参考手册》(DS70046E_CN) 和《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

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 (Least Significant bit, LSb) 始终清零 (见第 3.1 节 “程序地址空间”), 最高有效位 (Most Significant bit, 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, 并且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, 每个工作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 64 KB (32K 字), 被分成两块, 称为 X 和 Y 数据存储区。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 - 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 个字节组成, 大多数指令可以按字或字节对数据进行寻址。

对于存储在程序存储器中的数据, 有两种访问方法:

- 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像访问数据空间那样访问程序空间, 但访问需要一个额外的周期。此外, 使用这个方法, 只能访问每个指令字的低 16 位。
- 使用任何工作寄存器, 通过表读和表写指令, 也可以对程序空间内的 32K 字页进行线性间接访问。表读和表写指令能够用来访问一个指令字的全部 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 还支持对目标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。有关模寻址和位反转寻址的详细信息, 请参见第 5.0 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、寄存器偏移量寻址和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一次数据 (或程序数据) 存储器读取、一次工作寄存器 (数据) 读取、一次数据存储器写入以及一次程序 (指令) 存储器读取操作。因此, 支持 3 操作数指令, 允许在单个周期内执行 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量。它具有一个高速 17 位 x 17 位乘法器、一个 40 位 ALU、两个 40 位饱和累加器和一个 40 位双向桶形移位寄存器。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 指令可以无缝地与所有其他指令一起操作, 且设计为能获得最佳实时性能。MAC 类指令能够在从存储器取出两个数据操作数的同时, 将两个 W 寄存器相乘。为了能够同时取两个数据操作数, 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取指令并部分译码指令。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，其范围从 8 到 15。

2.2 编程模型

编程模型如图 2-1 所示，它包括 16 个 16 位工作寄存器（W0 至 W15）、2 个 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。所有寄存器都是存储器映射的。W0 用作进行文件寄存器寻址的 W 寄存器。

这些寄存器中，一些寄存器有与之相关的影子寄存器，如图 2-1 所示。影子寄存器用作临时保持寄存器，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访问。寄存器与其影子寄存器之间的数据传递遵循以下规则：

- 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内容传递。

- 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低字节（Least Significant Byte，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访问来操作低字节和高字节（Most Significant Byte，MSB）。

2.2.1 软件堆栈指针 / 帧指针

dsPIC® DSC 器件具有一个软件堆栈。W15 是专用的软件堆栈指针（Stack Pointer，SP），异常处理、子程序调用与返回时将自动修改 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、写和操作（例如，创建堆栈帧）。

注： 为了防止出现不对齐的堆栈访问，
W15<0> 始终清零。

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 是专用的堆栈帧指针，由 LNK 和 ULINK 指令定义。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（SR Low Byte，SRL），其 MSB 称为 SR 高字节（SR High Byte，SRH）。SR 寄存器组成见图 2-1。

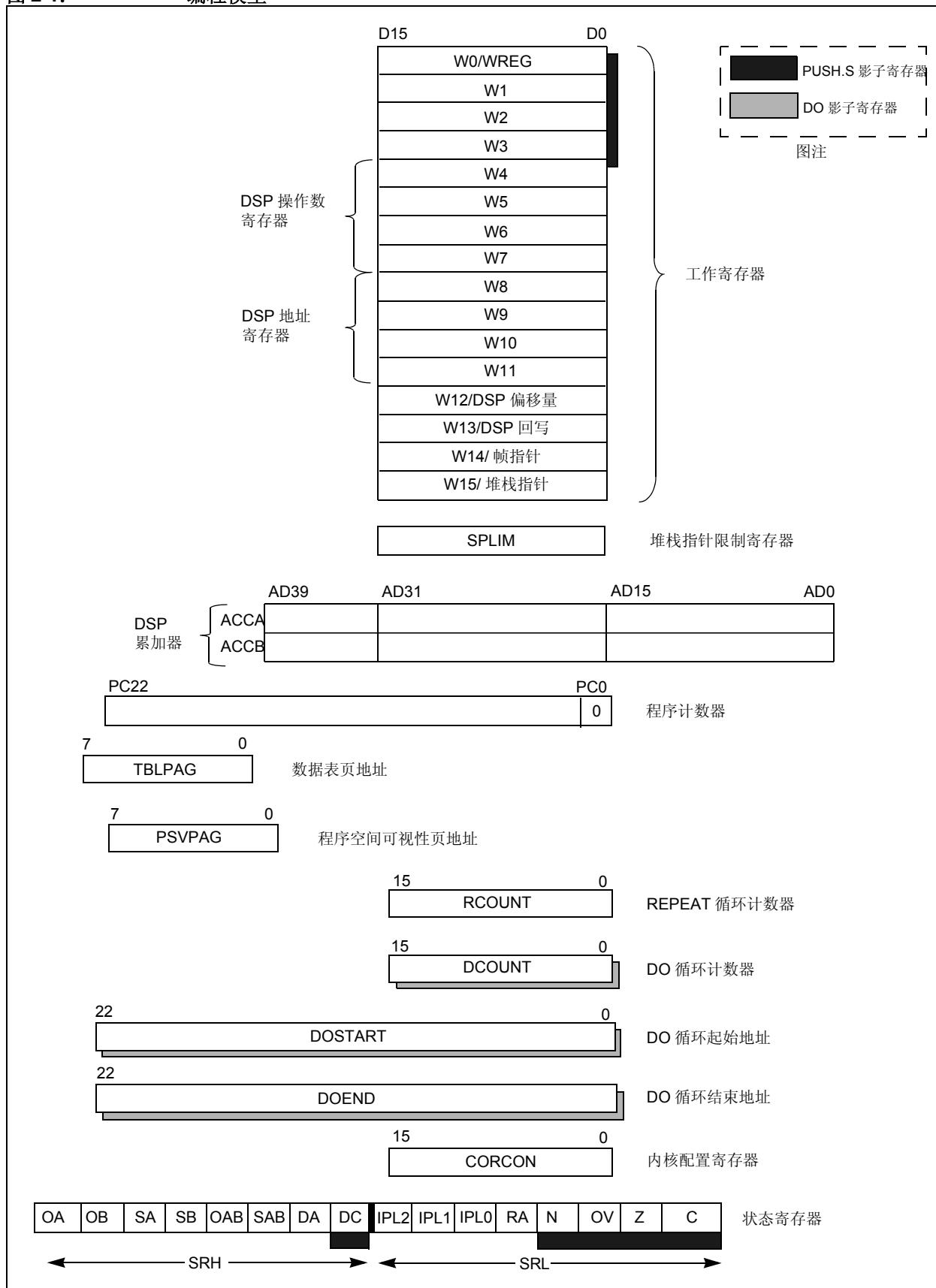
SRL 包含所有的 MCU ALU 运算状态标志位（包括 Z 位），以及 CPU 中断优先级状态位 IPL<2:0> 和 REPEAT 有效状态位 RA。在异常处理期间，SRL 与 PC 的 MSB 连接起来形成一个完整的字值，然后将该字值压入堆栈。

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 始终清零。因此，PC 能够寻址最多 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 16/16 位有符号小数除法运算，以及 32/16 位、 16/16 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 循环内执行。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 指令中指定，如表 2-1 所示（REPEAT 将执行目标指令 { 操作数的值 +1} 次）。DIV/DIVF 指令的 REPEAT 循环计数必须设置为 18 次迭代。所以，完整的除法操作需要 19 个周期。

注： 除法流可以被中断。但是，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 → W0 ; Rem → W1
DIV.sd	有符号除法: (Wm+1:Wm)/Wn → W0 ; Rem → W1
DIV.sw 或 DIV.s	有符号除法: Wm/Wn → W0 ; Rem → W1
DIV.ud	无符号除法: (Wm+1:Wm)/Wn → W0 ; Rem → W1
DIV.uw 或 DIV.u	无符号除法: Wm/Wn → W0 ; Rem → W1

2.4 DSP 引擎

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

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

dsPIC30F 采用单周期指令流架构，因此 DSP 引擎操作和 MCU 指令流不能同时进行。但某些 MCU ALU 和 DSP 引擎资源可被同一条指令同时使用（如 ED 和 EDAC）。

通过 CPU 内核配置寄存器（CORCON）中的各个位，可以选择 DSP 引擎的多种功能，功能如下：

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 自动饱和使能 / 禁止（SATA）。
5. ACCB 自动饱和使能 / 禁止（SATB）。
6. 对于写数据存储器，自动饱和使能/禁止（SATDW）。
7. 累加器饱和模式选择（ACCSAT）。

注： CORCON 的组成见表 3-3。

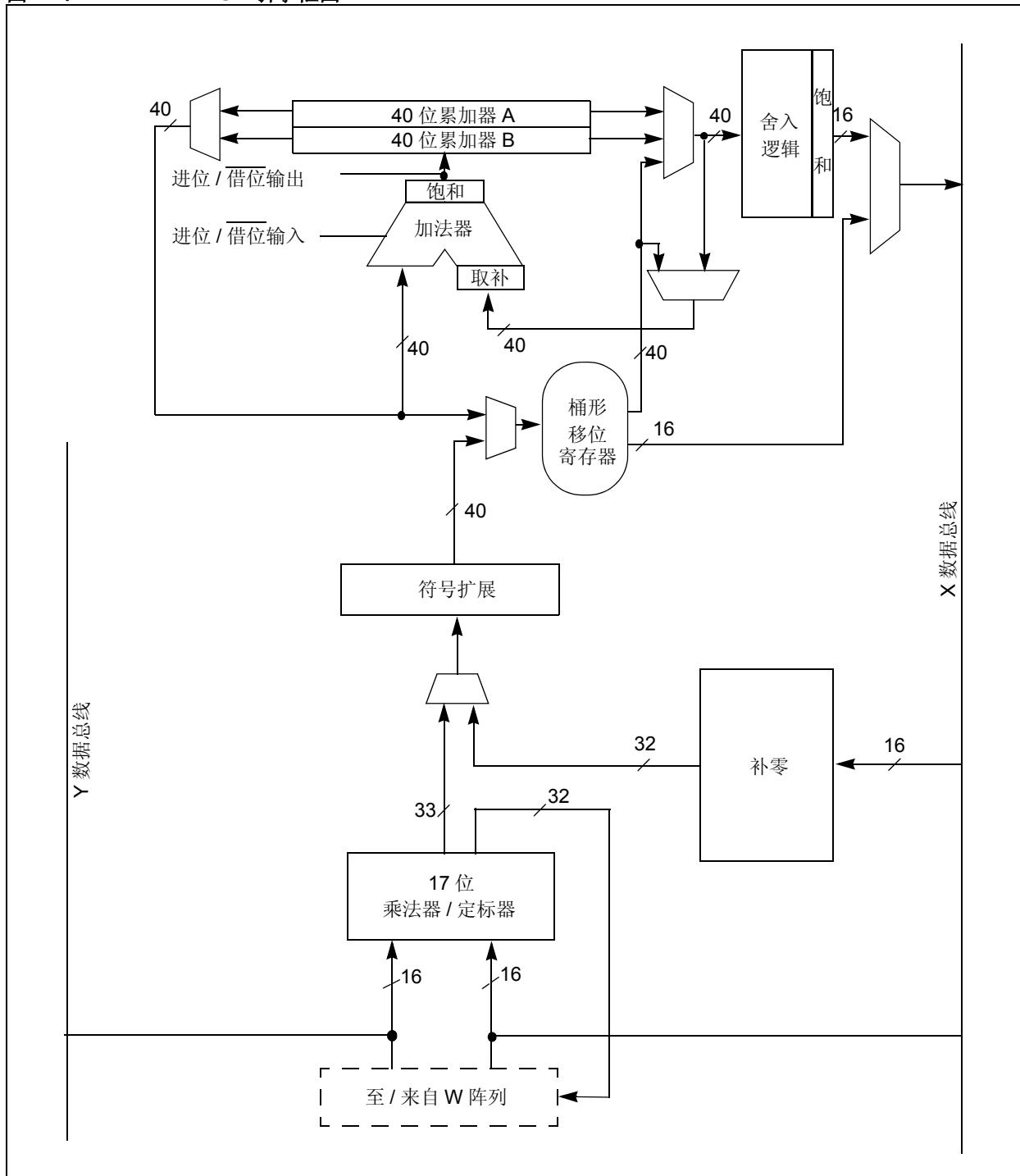
DSP 引擎的框图如图 2-2 所示。

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
MOV SAC	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

dsPIC30F5011/5013

图 2-2: DSP 引擎框图



2.4.1 乘法器

17x17 位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。17x17 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，数据范围为 -32768 (0x8000) 到 32767 (0x7FFF)，包括 0 在内。对于 32 位整数，数据范围为 -2,147,483,648 (0x8000 0000) 到 2,147,483,645 (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为 -1.0 到 $(1 - 2^{1-N})$ 。对于 16 位小数，Q15 数据范围为 -1.0 (0x8000) 到 0.999969482 (0x7FFF)，包括 0 在内，其精度为 3.01518×10^{-5} 。在小数模式下，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、无符号和混合符号乘法。

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

2.4.2 数据累加器和加法器 / 减法器

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

2.4.2.1 加法器 / 减法器、溢出和饱和

加法器 / 减法器是一个 40 位加法器，一侧输入可以选择为零，而另一侧输入可以是原数据或求补后的数据。对于加法，进位 / 借位输入为高电平有效，另一侧输入是原数据（没有求补的）；对于减法，进位 / 借位输入为低电平有效，另一侧输入是求补后的数据。加法器 / 减法器产生溢出状态位 **SA/SB** 和 **OA/OB**，这些状态位被锁存在状态寄存器中并在其中得到反映。

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

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

状态寄存器中有 6 个支持饱和和溢出的位，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
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
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使能位 (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 4.0 节“中断”）。这使得用户能够立即采取措施，例如，校正系统增益。

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

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

器件支持三种饱和和溢出模式：

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（0x8000000000）装入目标累加器。SA 或 SB 位置 1 并保持直到被用户清零。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（0x0080000000）装入目标累加器。SA 或 SB 位置 1 并保持直到被用户清零。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：

加法器的 bit 39 溢出状态位用来将 SA 或 SB 位置 1；这两位置 1 后，将保持该状态直到被用户清零。不进行饱和操作，允许累加器溢出（破坏其符号位）。如果 INTCON1 寄存器中的 COVTE 位置 1，灾难性溢出会导致一个陷阱异常。

2.4.2.2 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 到 bit 31）的舍入形式写入数据存储空间，前提是当前指令不对该累加器进行操作。通过 X 总线寻址组合的 X 和 Y 地址空间，执行回写操作。支持以下寻址模式：

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模块，在累加器写（存储）过程中执行常规的（有偏）或收敛的（无偏）舍入功能。舍入模式由 CORCON 寄存器中 RND 位的状态决定。它会产生一个 16 位的 1.15 数据值，该值被送到数据空间写饱和逻辑。如果指令不指明舍入，就会存储一个截取的 1.15 数据值，简单地丢弃低位字（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 到 bit 31）。如果 ACCxL 字（累加器的 bit 0 到 bit 15）在 0x8000 和 0xFFFF 之间（包括 0x8000），则 ACCxH 递增 1。如果 ACCxL 在 0x0000 和 0x7FFF 之间，则 ACCxH 不变。此算法的结果经过一系列随机舍入操作，值将稍稍偏大（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 Lsb（累加器的 bit 16）进行检测。如果它为 1，ACCxH 递增 1。如果它为 0，ACCxH 不变。假设 bit 16 本身是随机的，这样的机制将消除任何可能累加的舍入偏差。

通过 X 总线，SAC 和 SAC.R 指令将目标累加器内容的截取（SAC）或舍入（SAC.R）形式存入数据存储空间（这受数据饱和的影响，请参见第 2.4.2.4 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。对于此类指令，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适当的 1.15 小数值作为输出，写入数据存储空间中。

如果 CORCON 寄存器中的 SATDW 位置 1，将检测（经过舍入或截取后的）数据是否溢出，并相应地进行调整。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，则写入存储器中的数据被强制为最大的负 1.15 值，0x8000。源累加器的 MSb (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可将数据算术或逻辑右移或左移 16 位。源操作数可以是两个 DSP 累加器中的任何一个或 X 总线（支持寄存器或存储器中数据的多位移位）。

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

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

dsPIC30F5011/5013

注：

3.0 存储器构成

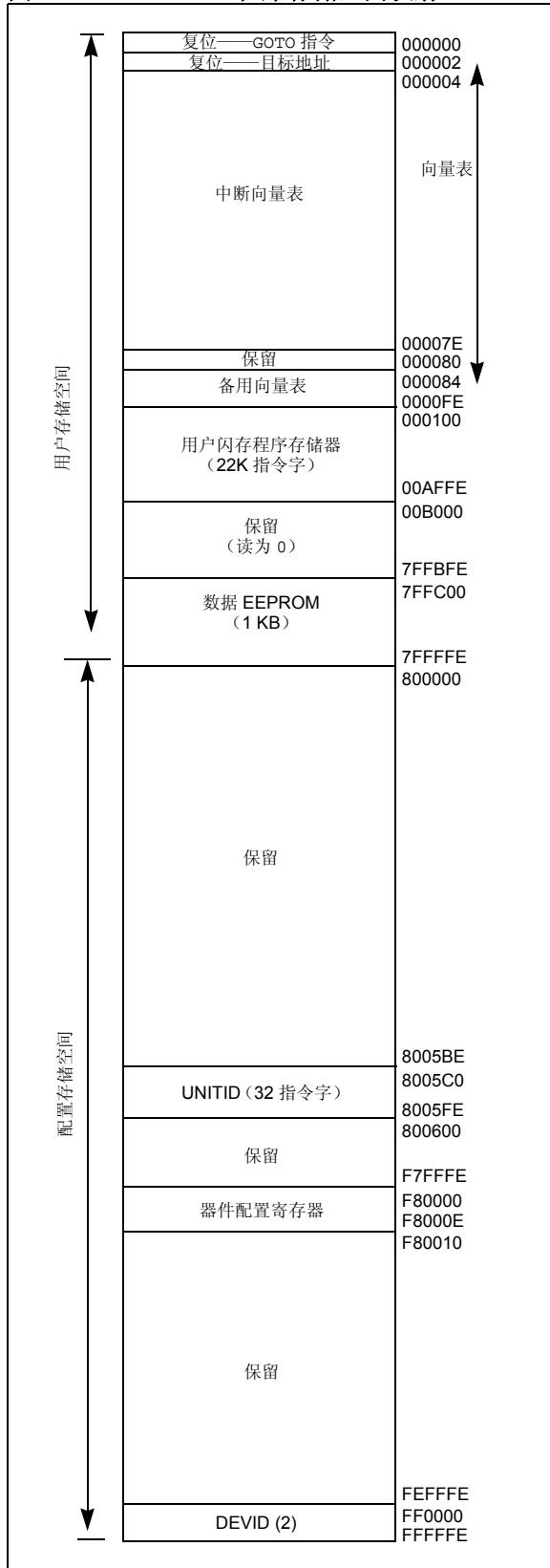
注：本数据手册总结了 dsPIC30F 系列器件的功能，但是不应把本手册当作无所不包的参考手册来使用。有关 CPU、外设、寄存器说明和一般器件功能的更多信息，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。有关器件指令集和编程的更多信息，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

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) 或数据空间 EA (当按照表 3-1 定义的那样，将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，程序空间地址递增量为 2。

除 TBLRD/TBLWT 指令外，对用户程序空间的所有访问限制在低 4M 指令字地址范围 (0x000000 至 0x7FFFFE)；TBLRD/TBLWT 使用 TBLPAG<7> 来决定访问用户空间还是配置空间。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。

图 3-1： 程序存储空间映射

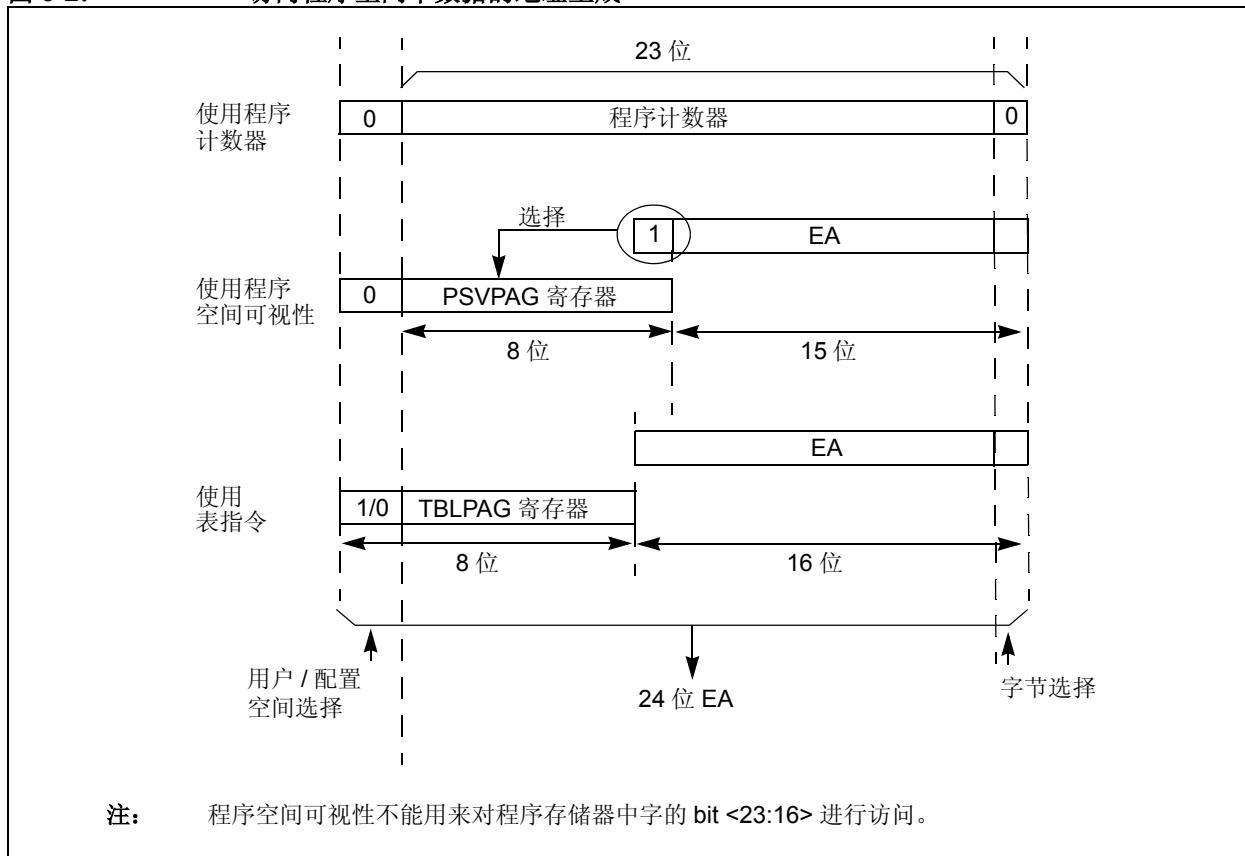


dsPIC30F5011/5013

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

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

图 3-2 说明了如何为表操作和数据空间访问 ($PSV = 1$) 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. TBLRDL：表读低位字
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， $P<7:0>$ 映射到目标字节；
当字节选择 = 1 时， $P<15:8>$ 映射到目标字节。
2. TBLWTL：表写低位字（闪存编程的详细信息，请参见第 6.0 节“闪存程序存储器”）。
3. TBLRDH：表读高位字
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， $P<23:16>$ 映射到目标字节；
当字节选择 = 1 时，目标字节始终 = 0。
4. TBLWTH：表写高位字（闪存编程的详细信息，请参见第 6.0 节“闪存程序存储器”）。

图 3-3： 程序数据表访问（低位字）

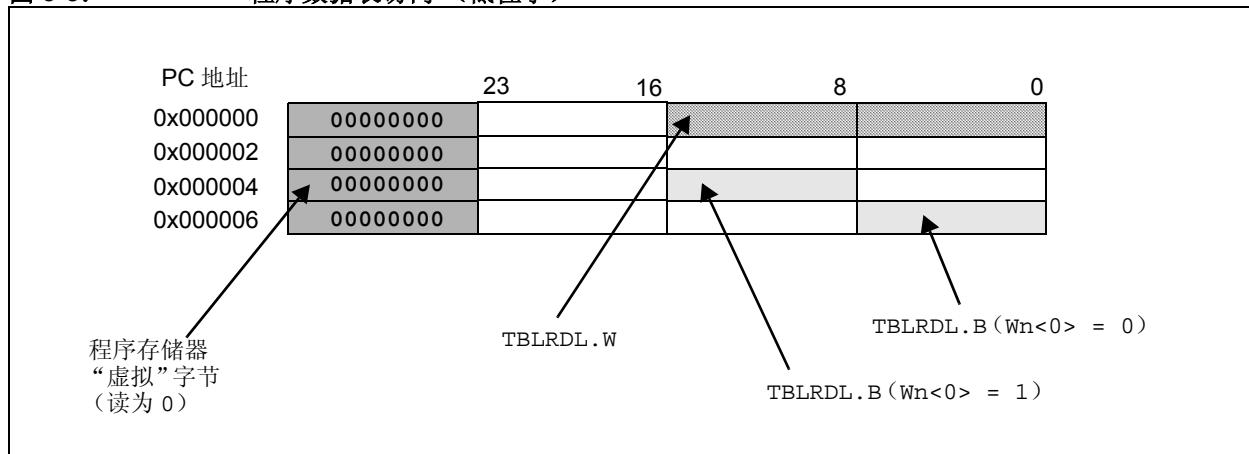
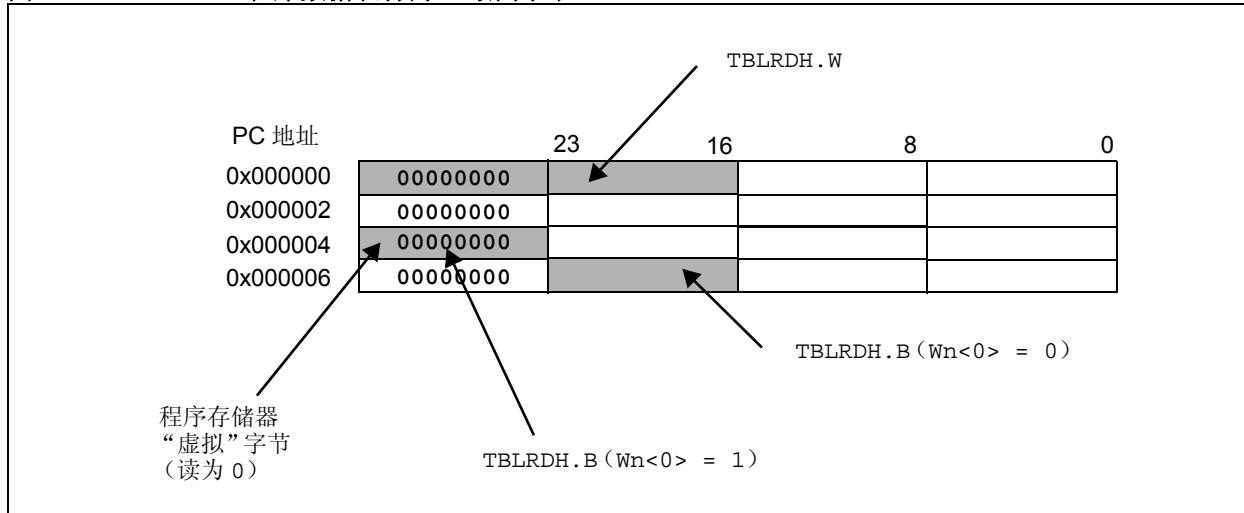


图 3-4: 程序数据表访问 (最高字节)



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 TBLRDL/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，就需要增加一个额外的指令周期，因为需要进行两次程序存储空间读取操作。

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。

尽管大于或等于 0x8000 的每个数据空间地址直接映射到对应的程序存储器地址（见图 3-5），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。指令编码的详细信息，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 / 写期间，暂时禁止 PSV 访问。

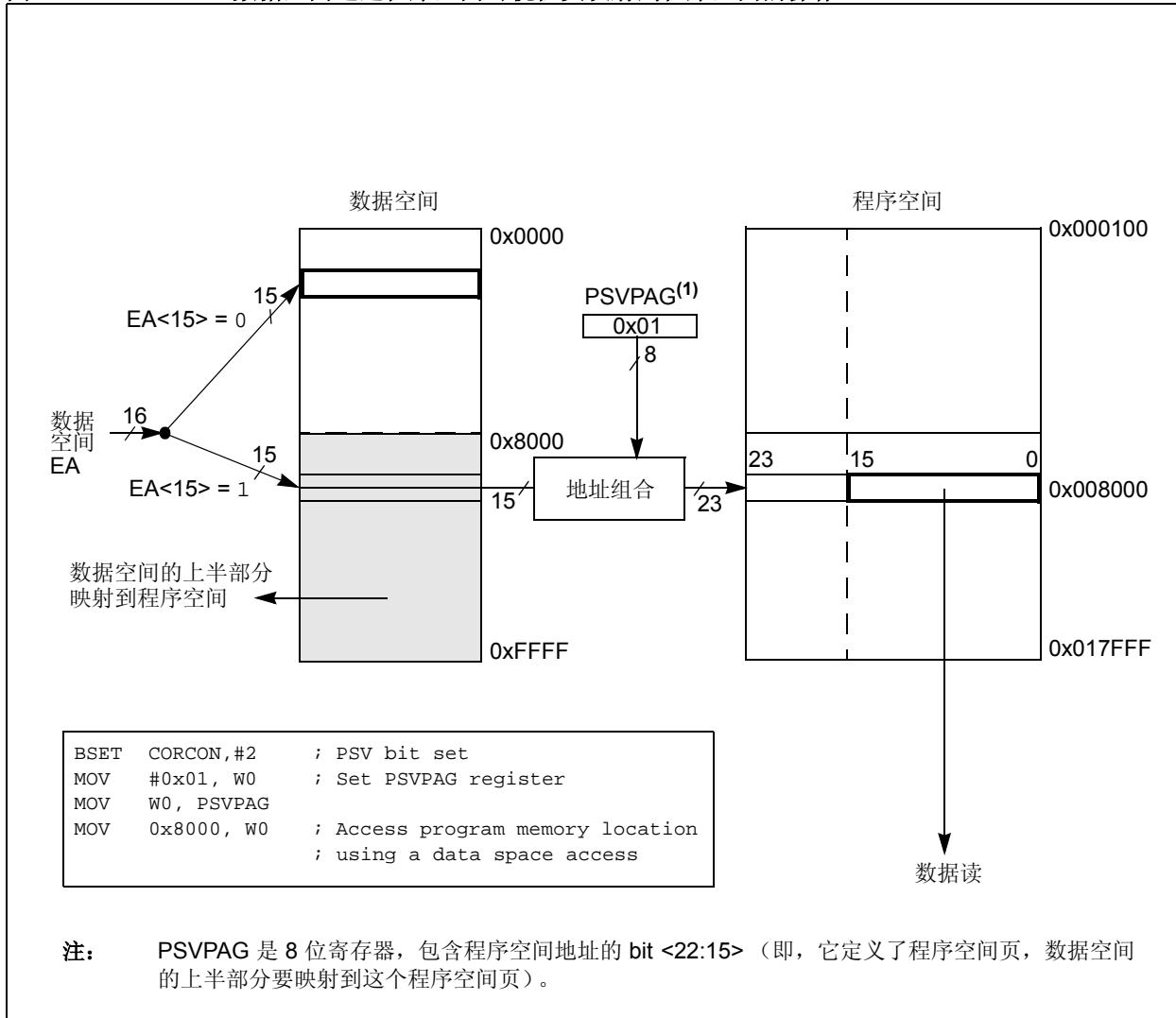
对于使用 PSV 而又在 REPEAT 循环之外执行的指令：

- 以下指令都需要在规定的执行时间之外额外增加一个指令周期：
 - 带数据操作数预取的 MAC 类指令
 - MOV 指令
 - MOV.D 指令
- 其他所有指令都需要在规定的指令执行时间之外额外增加两个指令周期。

对于使用 PSV 而又在 REPEAT 循环之内执行的指令：

- 以下情况都需要在规定的指令执行时间之外额外增加两个指令周期：
 - 在第一次迭代中执行的指令
 - 在最后一次迭代中执行的指令
 - 由于中断而退出循环之前执行的指令
 - 中断得到处理后再次进入循环时执行的指令
- REPEAT 循环的所有其他各次迭代，都允许使用 PSV 访问数据的指令在一个周期内执行。

图 3-5: 数据空间通过程序空间可视性页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（对于某些 MCU 指令而言）。使用两个地址发生单元（AGU）和独立的数据路径访问这两个数据空间。

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 空间中。为了提供表面看似的线性寻址空间，X 和 Y 空间要有连续的地址。

dsPIC30F5011/5013

在执行除 MAC 类指令之外的任何指令时，X 块由 64 KB 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 64 KB 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 对其进行寻址。剩下的 X 数据空间则使用 W8 和 W9 来寻址。只有通过 MAC 类指令才能同时访问这两个地址空间。

数据存储空间映射如图 3-6 所示。X 数据空间可用于所有指令，并且支持所有寻址模式，如图 3-7 所示。

图 3-6: 数据存储空间映射

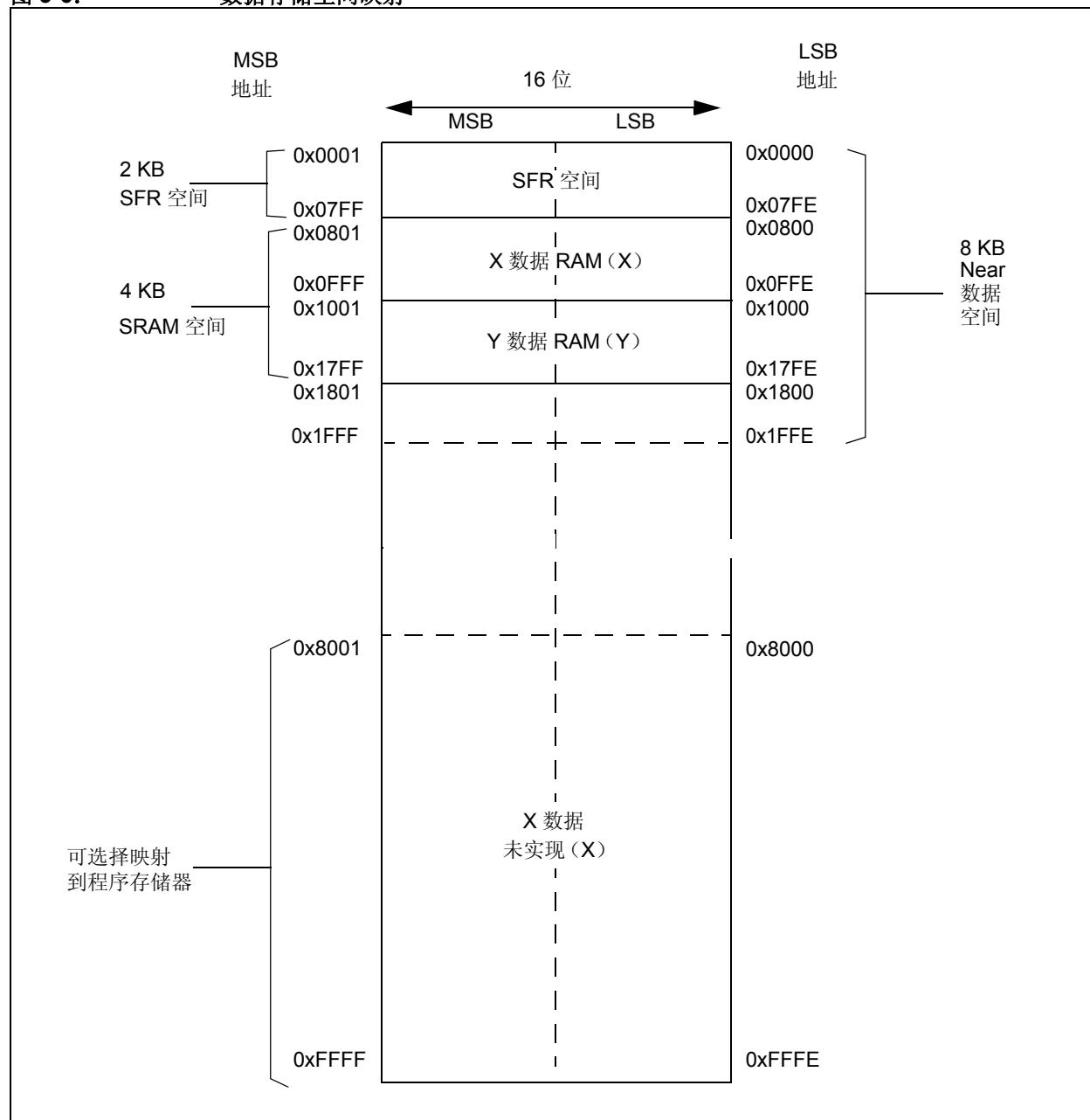
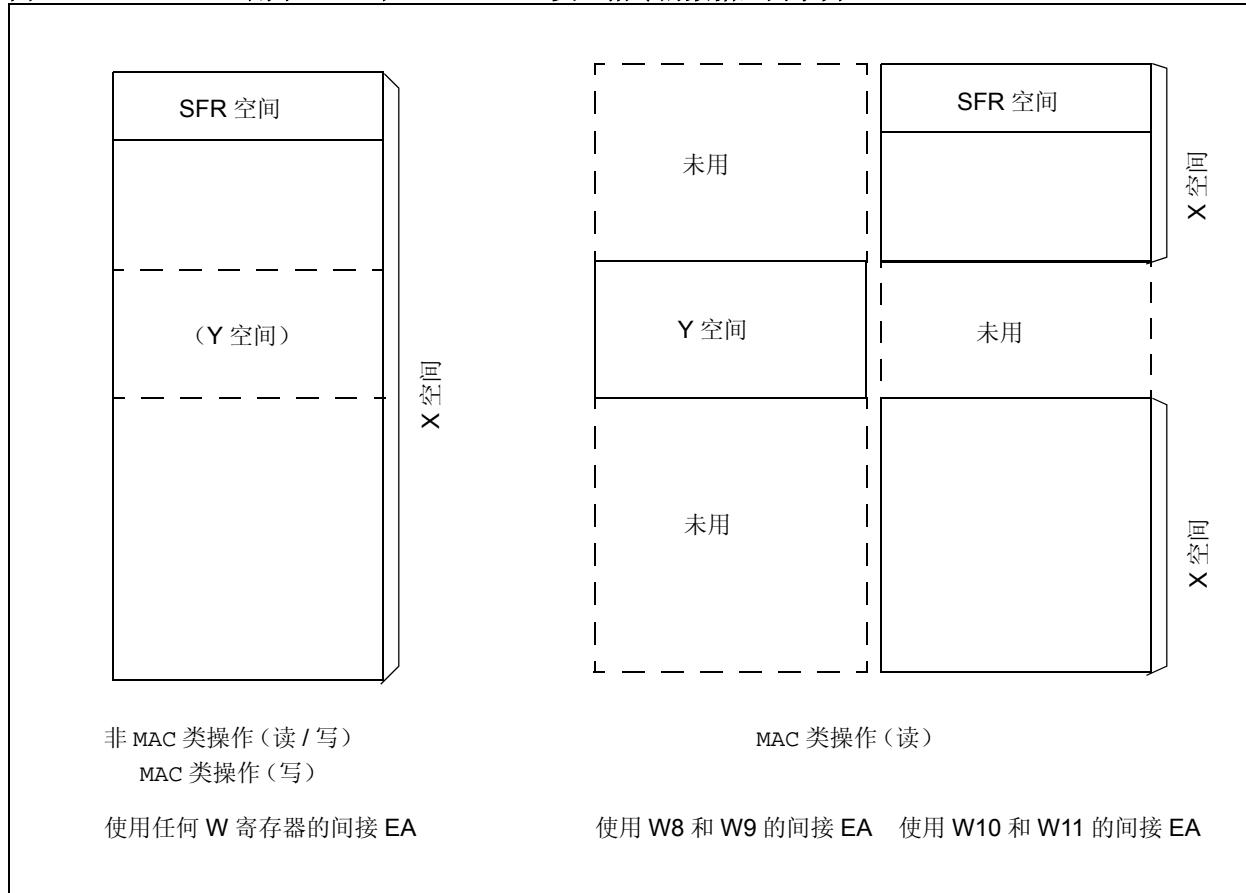


图 3-7：用于 MCU 和 DSP (MAC 类) 指令的数据空间示例



3.2.2 数据空间

X 数据空间可用于所有指令，并且支持所有寻址模式。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，它把数据空间视为组合的 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。X 写数据总线是所有指令对数据空间进行写操作的唯一路径。

X 数据空间还支持所有指令的模寻址，不过要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）将 Y 数据空间与 X 数据空间配合使用，提供了两条并行的数据读取路径。不能通过 Y 总线进行任何写操作。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，数据地址空间被视为 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 类指令相关的数据预取操作。它也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 指令都是可见的，但如果一条 MAC 指令试图使用 W8 或 W9（X 空间指针）从该空间取数据，则将返回 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 指令中使用 W8 或 W9 访问 Y 数据空间	0x0000
在 MAC 指令中使用 W10 或 W11 访问 X 数据空间	0x0000

所有有效地址均为 16 位宽，并且指向数据空间内的字节。所以，数据空间地址范围为 64 KB 或 32K 字。

3.2.3 数据空间宽度

内核数据宽度为 16 位。所有内部寄存器均按 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® MCU 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 EA 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 EA 的 LSB 来确定要选取的字节。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被组织为两个并行的字节宽的实体，它们共享（字）地址译码，但写入线独立。数据字节写操作只写阵列或寄存器中与字节地址匹配的那一侧。

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。

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

图 3-8：数据对齐

	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

所有装入 W 寄存器的字节都将被装入相应的 LSB 中。相应的 MSB 不变。

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

尽管大多数指令能够对字或字节大小的数据进行操作，但要注意的是，部分指令（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，在所有存储器直接寻址指令中，可以通过一个 13 位的绝对地址字段来直接访问这个数据空间。其余的 X 地址空间和所有 Y 地址空间都是间接寻址的。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。

堆栈指针总是指向第一个可用的空字，并且从低地址向高地址方向增长。它在弹出堆栈之前递减，而在压入堆栈之后递增，如图 3-9 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，在将 PC 压入堆栈之前，要先将 PC 的 MSB 与 SRL 寄存器组合在一起。

堆栈指针限制寄存器 (SPLIM) 与堆栈指针相关。复位时 SPLIM 不被初始化。与堆栈指针的情况一样，SPLIM<0> 被强制为 0，因为所有的堆栈操作必须是字对齐的。每当使用 W15 作为源指针或目标指针产生有效地址 (EA) 时，将与 SPLIM 中的值进行比较。如果堆栈指针 (W15) 的内容与 SPLIM 寄存器的内容相等，则执行压栈操作时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 0xFFE 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，从而避免了堆栈干扰特殊功能寄存器（Special Function Register，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

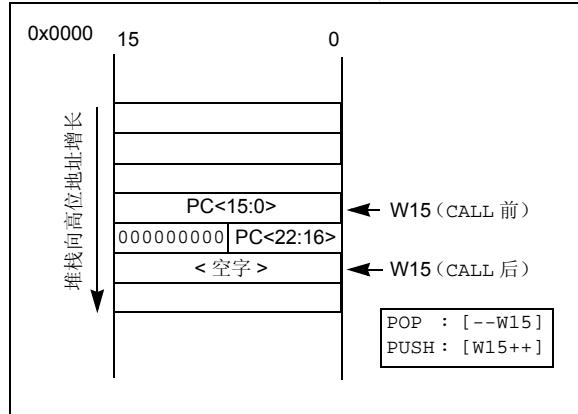


表 3-3: 内核寄存器映射

SFR 名称	地址 (首地址)	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																0000 0000 0000 0000	
W1	0002																0000 0000 0000 0000	
W2	0004																0000 0000 0000 0000	
W3	0006																0000 0000 0000 0000	
W4	0008																0000 0000 0000 0000	
W5	000A																0000 0000 0000 0000	
W6	000C																0000 0000 0000 0000	
W7	000E																0000 0000 0000 0000	
W8	0010																0000 0000 0000 0000	
W9	0012																0000 0000 0000 0000	
W10	0014																0000 0000 0000 0000	
W11	0016																0000 0000 0000 0000	
W12	0018																0000 0000 0000 0000	
W13	001A																0000 0000 0000 0000	
W14	001C																0000 0000 0000 0000	
W15	001E																0000 1000 0000 0000	
SPLIM	0020																0000 0000 0000 0000	
ACCAL	0022																0000 0000 0000 0000	
ACCAH	0024																0000 0000 0000 0000	
ACCAU	0026																0000 0000 0000 0000	
ACCBL	0028																0000 0000 0000 0000	
ACCBH	002A																0000 0000 0000 0000	
ACCBU	002C																0000 0000 0000 0000	
PCL	002E																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																RRRR RRRR RRRR RRRR	
DCOUNT	0038																RRRR RRRR RRRR RRRR	
DOSTARTL	003A																DOSTARTL	0000 0000 0000 0000
DOSTARTH	003C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DOSTARTH	0000 0000 0000 0000
DOENDL	003E																DOENDL	0000 0000 0000 0000
DOENDH	0040	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DOENDH	0000 0000 0000 0000
SR	0042	OA	OB	SA	SB	OAB	SAB	DA	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000 0000 0000 0000

图注: u = 未初始化位

表 3-3: 内核寄存器映射 (续)

SFR 名称	地址 (首地址)	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
CORCON	0044	—	—	—	US	EDT	DL2	DL1	DL0	SATA	SATB	SATDW	ACCSAT	IPL3	PSV	RND	IF	0000 0000 0010 0000
MODCON	0046	XMODEN	YMODEN	—	—	BWM<3:0>				YWM<3:0>				XWM<3:0>			0000 0000 0000 0000	
XMODSRT	0048							XS<15:1>								0	uuuu uuuu uuuu uuu0	
XMODEND	004A							XE<15:1>								1	uuuu uuuu uuuu uuu1	
YMODSRT	004C							YS<15:1>								0	uuuu uuuu uuuu uuu0	
YMODEND	004E							YE<15:1>								1	uuuu uuuu uuuu uuu1	
XBREV	0050	BREN							XB<14:0>								uuuu uuuu uuuu uuuu	
DISICNT	0052	—	—							DISICNT<13:0>							0000 0000 0000 0000	
BSRAM	0750	—	—	—	—	—	—	—	—	—	—	—	—	IW_BSR	IR_BSR	RL_BSR	0000 0000 0000 0000	
SSRAM	0752	—	—	—	—	—	—	—	—	—	—	—	—	IW_SSR	IR_SSR	RL_SSR	0000 0000 0000 0000	

图注: u = 未初始化位

注: 有关寄存器各位的说明, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

dsPIC30F5011/5013

注：

4.0 中断

注: 本数据手册总结了 dsPIC30F 系列器件的功能, 但是不应把本手册当作无所不包的参考手册来使用。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 传感器和通用系列具有最多 41 个中断源和 4 个处理器异常（陷阱），所有中断和异常的处理必须依据优先级机制进行仲裁。

CPU 负责读取中断向量表 (Interrupt Vector Table, IVT), 并把中断向量中的地址传送给程序计数器。中断向量从程序数据总线向程序计数器的传送, 是经由程序计数器输入侧 24 位宽的多路开关来进行的。

中断向量表 (IVT) 和备用中断向量表 (AIVT) 的位置靠近程序存储器的开始 (0x000004)。IVT 和 AIVT 如图 4-1 所示。

在中断和处理器异常提交给处理器内核之前, 中断控制器负责它们的预处理。使用集中式的特殊功能寄存器对外设中断和陷阱进行允许、优先级分配和控制:

- IFS0<15:0>、IFS1<15:0> 和 IFS2<15:0>
这三个寄存器中保存所有的中断请求标志。这些标志由其对应的外设或外部信号置 1, 标志的清除则通过软件完成。
- IEC0<15:0>、IEC1<15:0> 和 IEC2<15:0>
这三个寄存器中保存所有的中断允许控制位。这些控制位用来单独允许来自外设或外部信号的中断。
- IPC0<15:0>...IPC10<7:0>
与 41 个中断中的每一个中断有关的用户可分配优先级, 集中存放在这 12 个寄存器中。
- IPL<3:0>
当前 CPU 优先级显式地存放在 IPL 位中。IPL<3>在 CORCON 寄存器中, 而 IPL<2:0> 则存放在处理器内核的状态寄存器 (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行为和备用向量表的使用。

• INTTREG<15:0>

相关的中断向量编号和新的 CPU 中断优先级被锁存到 INTTREG 寄存器中的向量编号 (VECNUM<5:0>) 和中断优先级 (ILR<3:0>) 位域。新的中断优先级是待处理中断的优先级。

注: 当中断条件产生时, 不管相应的中断允许位状态如何, 中断标志位都将置 1。用户软件应在允许一个中断前, 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 7 个优先级中的一个, 优先级从 1 到 7。每个中断源都与一个中断向量相关, 如表 4-1 所示。优先级 7 和 1 分别代表最高和最低的可屏蔽优先级。

注: 中断源优先级指定为 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, 如果正在处理一个中断, 将禁止处理新的中断, 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 位置为 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。

处理中断时, 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 (见表 4-1)。这些向量存放在程序存储器从 0x000004 到 0x00000FE 的存储单元中 (见表 4-1)。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。这避免了执行随机数据; PC 递减时意外进入向量空间、意外地把数据空间地址映射到了向量空间, 或者 PC 到达可用程序存储空间的末尾继而返回到 0x000000, 都将执行随机数据。执行 GOTO 指令跳转到向量空间, 也将产生地址错误陷阱。

4.1 中断优先级

对于每个中断源，用户可分配中断优先级位 (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。

自然顺序优先级由中断在向量表中的位置决定；只有在具有相同用户分配优先级的多个中断同时等待处理时，才会影响中断操作。

表 4-1 列出了 dsPIC DSC 器件的中断编号和中断源，以及相关的向量编号。

注 1： 自然顺序优先级机制以 0 为最高优先级，以 53 为最低优先级。

2： 自然顺序优先级编号与 INT 编号相同。

用户可以为每个中断分配 7 个优先级之一，这意味着用户可以为具有较低自然顺序优先级的中断分配一个极高的总优先级。例如，可将 PLVD（低电压检测）的优先级分配为 7。INT0（外部中断 0）可以分配为优先级 1，这样，它的有效优先级将非常低。

表 4-1：中断向量表

INT 编号	向量编号	中断源
最高自然顺序优先级		
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——Timer1
4	12	IC2——输入捕捉 2
5	13	OC2——输出比较 2
6	14	T2——Timer2
7	15	T3——Timer3
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 写入完成
13	21	SI2C——I ² C TM 从中断
14	22	MI2C——I ² C 主中断
15	23	输入电平变化中断
16	24	INT1——外部中断 1
17	25	IC7——输入捕捉 7
18	26	IC8——输入捕捉 8
19	27	OC3——输出比较 3
20	28	OC4——输出比较 4
21	29	T4——Timer4
22	30	T5——Timer5
23	31	INT2——外部中断 2
24	32	U2RX——UART2 接收器
25	33	U2TX——UART2 发送器
26	34	SPI2
27	35	C1——CAN1 的组合 IRQ
28	36	IC3——输入捕捉 3
29	37	IC4——输入捕捉 4
30	38	IC5——输入捕捉 5
31	39	IC6——输入捕捉 6
32	40	OC5——输出比较 5
33	41	OC6——输出比较 6
34	42	OC7——输出比较 7
35	43	OC8——输出比较 8
36	44	INT3——外部中断 3
37	45	INT4——外部中断 4
38	46	C2——CAN2 的组合 IRQ
39-40	47-48	保留
41	49	DCI——编解码器传送完成
42	50	LVD——低电压检测
43-53	51-61	保留
最低自然顺序优先级		

4.2 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，处理器初始化其寄存器。然后，处理器从单元 0x000000 开始执行程序。一条 GOTO 指令存放在第一个程序存储单元中，紧跟着的是 GOTO 指令的地址目标。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。

4.2.1 复位的来源

除了外部复位和上电复位（POR）外，还有 6 个错误条件会引起器件复位。

- 看门狗定时器超时：
看门狗定时器已超时，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在非法指令执行之前就被从指令流中舍弃的话，取非法指令并不产生非法指令陷阱。
- 欠压复位（BOR）：
检测到可能导致器件故障的电源电压短暂降低。
- 陷阱锁定：
多个并发的陷阱条件将导致复位。

4.3 陷阱

可以将陷阱看作不可屏蔽中断，表明出现软件或硬件错误，陷阱遵循预定义的优先级，如图 4-1 所示。陷阱旨在为用户提供一种方法，修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，则必须在这些向量中装入只包含 RESET 指令的默认陷阱处理程序的地址。否则，如果调用了包含非法地址的向量，将产生地址错误陷阱。

注意，很多陷阱条件只有在发生的时候才能检测到。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，可能不得不修正导致陷阱的错误操作的结果。

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

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

4.3.1 陷阱的来源

下列陷阱的优先级依次递增。但是，由于所有陷阱都可以嵌套，优先级的作用很小。

数学错误陷阱：

在以下四种情况下，产生数学错误陷阱：

1. 如果试图进行以零作除数的除法操作，它将在周期边界处中止，并产生陷阱。
2. 如果使能的话，当对累加器 A 或 B 进行的算术操作导致 bit 31 溢出且没有使用累加器警戒位时，将产生数学错误陷阱。
3. 如果使能的话，当对累加器 A 或 B 进行的算术操作导致 bit 39 灾难性溢出且所有饱和被禁止时，将产生数学错误陷阱。
4. 如果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。

地址错误陷阱：

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. 试图从未实现的程序存储单元取数据。
4. 试图从向量空间取指令。

注： 在 MAC 类指令中，数据空间被分为 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. 在将 PC 修改为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：

该陷阱在以下条件下产生：

1. 在堆栈指针中装入了一个大于堆栈限制的值（堆栈溢出）；堆栈限制值用户可编程，存放在 SPLIM 寄存器中。
2. 堆栈指针装入了一个小于 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。

4.3.2 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 4-2 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。

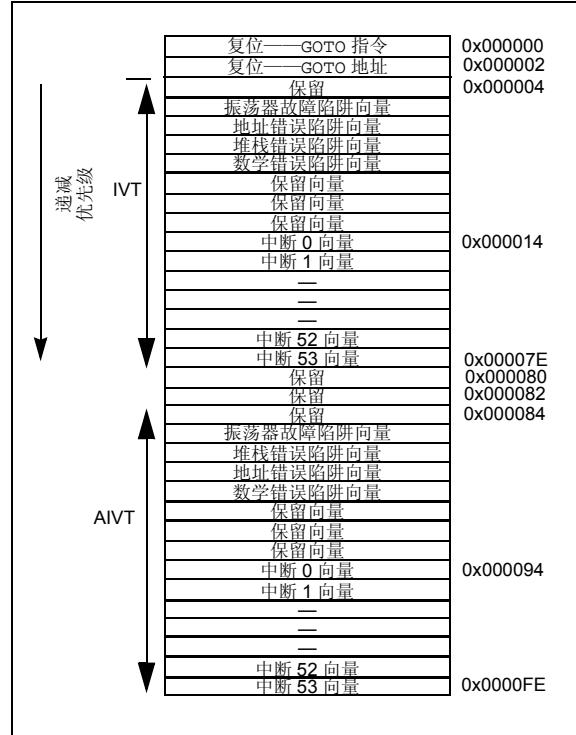
“软”陷阱包括优先级 8 到 11 的异常（含）。算术错误陷阱（优先级 11）就属于这一类陷阱。

“硬”陷阱包括优先级 12 到 15 的异常（含）。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，如果产生了优先级较低的硬陷阱，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，从而可用软件检测到该条件。

图 4-1：陷阱向量

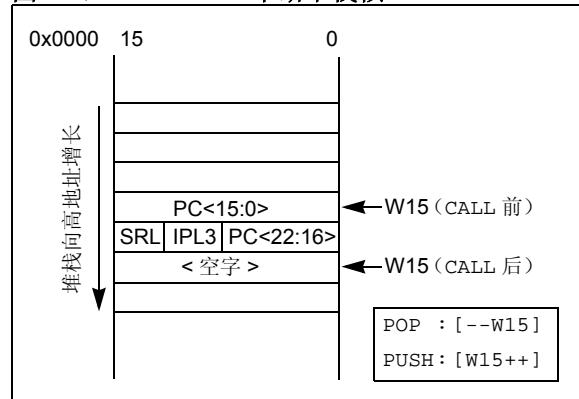


4.4 中断过程

在每个指令周期的开始，IFSx 寄存器都会采样所有的中断事件标志。在 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，则 IRQ 会导致中断产生。在余下的指令周期中，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 4-2 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）完成为止。

图 4-2： 中断堆栈帧



- 注 1：** 用户始终能通过把一个新值写入 SR 来降低优先级。在降低处理器中断优先级之前，为了避免重复中断，中断服务程序必须把 IFSx 寄存器中的中断标志位清零。
- 2：** 在处理中断时，IPL3 位（CORCON<3>）始终被清零。它仅在陷阱执行期间置 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，使处理器返回到中断处理之前的状态。

4.5 备用向量表

在程序存储器中，中断向量表（IVT）之后是备用中断向量表（AIVT），如图 4-1 所示。INTCON2 寄存器中的 ALTVT 位控制对备用向量表的访问。如果 ALTVT 位置 1，则所有的中断和异常处理都将使用备用向量，而不是默认向量。备用向量与默认向量的组织方式相同。AIVT 提供了一种在应用和支持环境之间切换的方法，从而支持仿真和调试功能，而无需重新设定中断向量。这个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。

如果不使用 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。

4.6 快速现场保护

可以使用影子寄存器来保存现场。为 SR 中的 DC、N、OV、Z 和 C 位，以及寄存器 W0 到 W3 提供了影子寄存器。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到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。

4.7 外部中断请求

中断控制器支持最多 5 个外部中断请求信号，INT0-INT4。这些输入是边沿敏感的；它们需要从低至高或从高至低的跳变来产生中断请求。INTCON2 寄存器的 5 个位（INT0EP-INT4EP）用于选择边沿检测电路的极性。

4.8 从休眠和空闲中唤醒

如果产生中断时器件处于休眠或空闲模式，中断控制器可以用来把处理器从休眠或空闲模式中唤醒。

如果中断控制器收到已允许的中断请求，且中断请求优先级足够高的话，则标准中断请求将提交给处理器。同时，处理器将从休眠或空闲中唤醒，开始执行处理中断请求所需的中断服务程序（ISR）。

表 4-2：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	—	—	—	—	OVATE	OVBTE	COVTE	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000 0000 0000 0000	
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP	0000 0000 0000 0000	
IFS0	0084	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT0IF	
IFS1	0086	IC6IF	IC5IF	IC4IF	IC3IF	C1IF	SPI2IF	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	IC8IF	IC7IF	INT1IF	
IFS2	0088	—	—	—	—	—	LVDIF	DCIIF	—	—	C2IF	INT4IF	INT3IF	OC8IF	OC7IF	OC6IF	OC5IF	
IEC0	008C	CNIE	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE	
IEC1	008E	IC6IE	IC5IE	IC4IE	IC3IE	C1IE	SPI2IE	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	IC8IE	IC7IE	INT1IE	
IEC2	0090	—	—	—	—	—	LVDIE	DCIIIE	—	—	C2IE	INT4IE	INT3IE	OC8IE	OC7IE	OC6IE	OC5IE	
IPC0	0094	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			INT0IP<2:0>			0100 0100 0100 0100	
IPC1	0096	—	T31P<2:0>			—	T2IP<2:0>			—	OC2IP<2:0>			IC2IP<2:0>			0100 0100 0100 0100	
IPC2	0098	—	ADIP<2:0>			—	U1TXIP<2:0>			—	U1RXIP<2:0>			SPI1IP<2:0>			0100 0100 0100 0100	
IPC3	009A	—	CNIP<2:0>			—	MI2CIP<2:0>			—	SI2CIP<2:0>			NVMIP<2:0>			0100 0100 0100 0100	
IPC4	009C	—	OC3IP<2:0>			—	IC8IP<2:0>			—	IC7IP<2:0>			INT1IP<2:0>			0100 0100 0100 0100	
IPC5	009E	—	INT2IP<2:0>			—	T5IP<2:0>			—	T4IP<2:0>			OC4IP<2:0>			0100 0100 0100 0100	
IPC6	00A0	—	C1IP<2:0>			—	SPI2IP<2:0>			—	U2TXIP<2:0>			U2RXIP<2:0>			0100 0100 0100 0100	
IPC7	00A2	—	IC6IP<2:0>			—	IC5IP<2:0>			—	IC4IP<2:0>			IC3IP<2:0>			0100 0100 0100 0100	
IPC8	00A4	—	OC8IP<2:0>			—	OC7IP<2:0>			—	OC6IP<2:0>			OC5IP<2:0>			0100 0100 0100 0100	
IPC9	00A6	—	—	—	—	—	C2IP<2:0>			—	INT41IP<2:0>			INT3IP<2:0>			0000 0100 0100 0100	
IPC10	00A8	—	—	—	—	—	LVDIP<2:0>			—	DCIIP<2:0>			—	—	—	0000 0100 0100 0000	
INTTREG	00B0	—	—	—	—	ILR<3:0>			—	—	VECNUM<5:0>			0000 0000 0000 0000			0000 0000 0000 0000	

图注： u = 未初始化位

注： 有关寄存器各位的说明，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

5.0 地址发生器单元

注: 本数据手册总结了 dsPIC30F 系列器件的功能,但是不应把本手册当作无所不包的参考手册来使用。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)。

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 类指令的字长度数据读取。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。

5.1 指令寻址模式

表 5-1 给出了基本的寻址模式,这些寻址模式经过优化可以支持各指令的具体功能。MAC 类指令中提供的寻址模式与其他指令类型中的寻址模式略有不同。

5.1.1 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器中的前 8192 字节(Near 数据空间)。大多数文件寄存器指令使用工作寄存器 W0, W0 在这些指令中表示为 WREG。目标寄存器通常是同一个文件寄存器或 WREG(MUL 指令除外),它把结果写入寄存器或寄存器对。使用 MOV 指令能够获得更大的灵活性,可以在文件寄存器操作期间访问整个数据空间。

5.1.2 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。结果位置可以是 W 寄存器或地址单元。MCU 指令支持以下寻址模式:

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

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

表 5-1: 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 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),再由此时的 Wn 内容形成 EA。
带寄存器偏移量的寄存器间接寻址	Wn 和 Wb 的和形成 EA。
带立即数偏移量的寄存器间接寻址	Wn 和立即数的和形成 EA。

5.1.3 传送指令和累加器类指令

与其他指令相比，传送指令和 DSP 累加器类指令提供了更为灵活的寻址模式。除了大多数 MCU 指令支持的寻址模式以外，传送和累加器指令还支持带寄存器偏移量的寄存器间接寻址模式，这也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 EA，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（但通常只由其中之一使用）。

概括地说，传送和累加器指令支持以下寻址模式：

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

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

5.1.4 MAC 指令

双源操作数 DSP 指令（CLR、ED、EDAC、MAC、MPY、MPY.N、MOVSAC 和 MSC）也叫做 MAC 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有效地对数据指针进行操作。

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

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

概括地说，MAC 类指令支持以下寻址模式：

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

5.1.5 其他指令

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

5.2 模寻址

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

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

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

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

5.2.1 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，并将它们装入 16 位模缓冲区地址寄存器中：XMODSRT、XMODEND、YMODSRT 和 YMODEND（见表 3-3）。

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

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

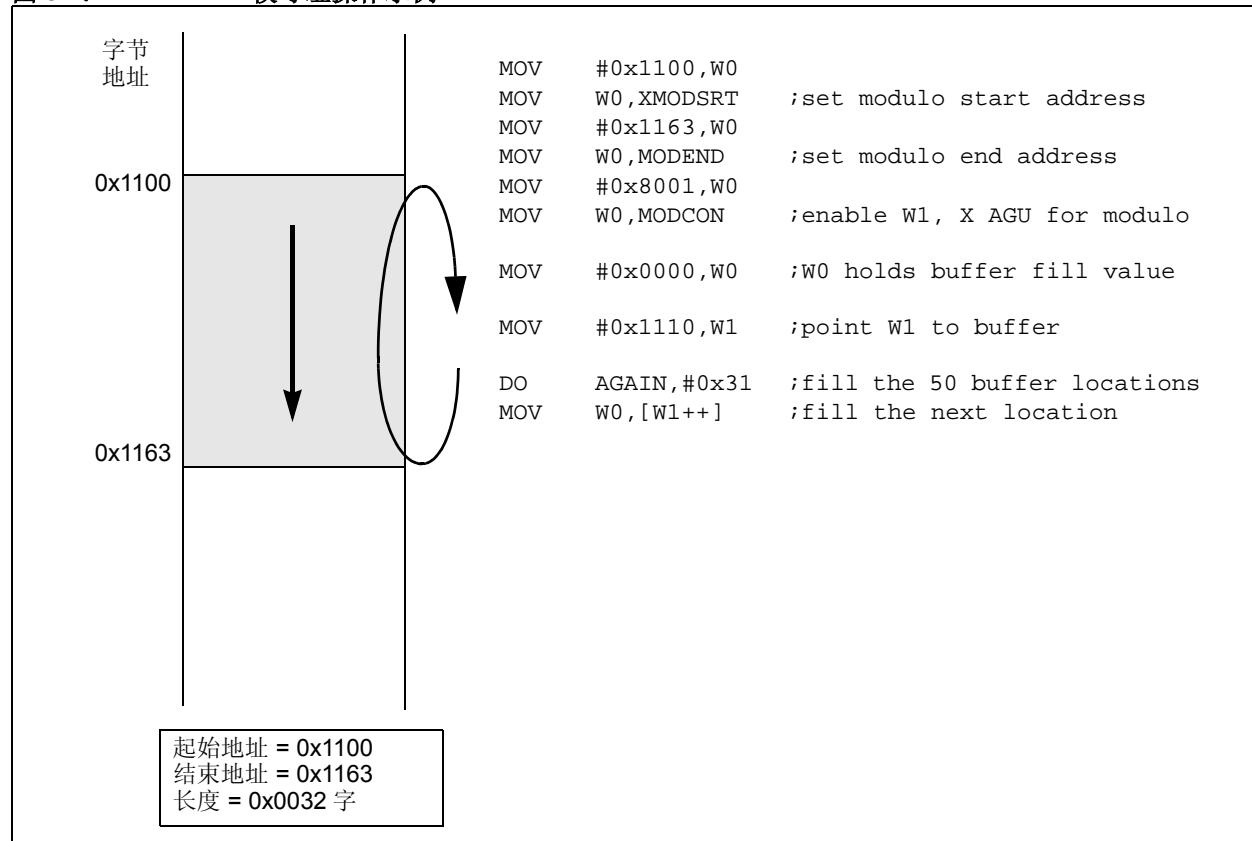
5.2.2 W 地址寄存器选择

模寻址和位反转寻址控制寄存器 MODCON<15:0> 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。类似地，如果 YWM = 15，则禁止 Y AGU 模寻址。

要对其进行模寻址的 X 地址空间指针 W 寄存器 (XWM) 存储在 MODCON<3:0> 中（见表 3-3）。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。

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

图 5-1： 模寻址操作示例



5.2.3 模寻址的应用

模寻址可以应用于任何与 W 寄存器相关的有效地址 (EA) 计算中。重要的是要意识到，地址边界检查功能不仅会检查地址是否正好在地址边界上，而且会检查地址是否小于或大于上限（对于递增缓冲区）、是否低于下限（对于递减缓冲区）。因此，地址变化可能会越过边界，但仍然可以正确调整。

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

5.3 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。它为 X AGU 所支持，仅限于数据写入。

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

5.3.1 位反转寻址的实现

当发生以下情况时，使能位反转寻址：

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 寄存器中的 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为 $M = 2^N$ 字节，则数据缓冲区起始地址的最后 “N” 位必须为零。

XB<14:0> 是位反转地址修改量或“中心点”(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 的计算都使用字长度数据（每个 EA 的 Lsb 始终清零）。为了产生兼容（字节）地址，要相应地调整 XB 的值。

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

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

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

图 5-2：位反转地址示例

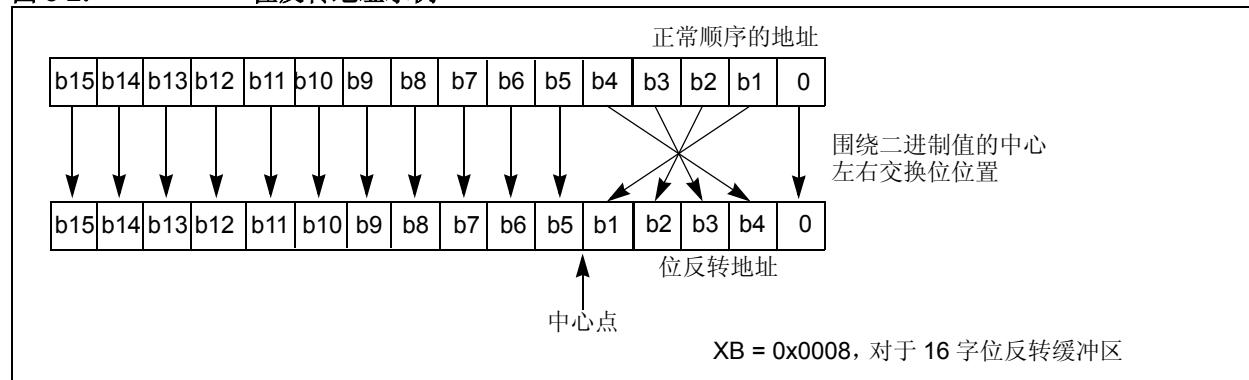


表 5-2: 位反转地址序列 (16 项)

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

表 5-3: 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
2048	0x0400
1024	0x0200
512	0x0100
256	0x0080
128	0x0040
64	0x0020
32	0x0010
16	0x0008
8	0x0004
4	0x0002
2	0x0001

dsPIC30F5011/5013

注：

6.0 闪存程序存储器

注：本数据手册总结了 dsPIC30F 系列器件的功能，但是不应把本手册当作无所不包的参考手册来使用。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)。

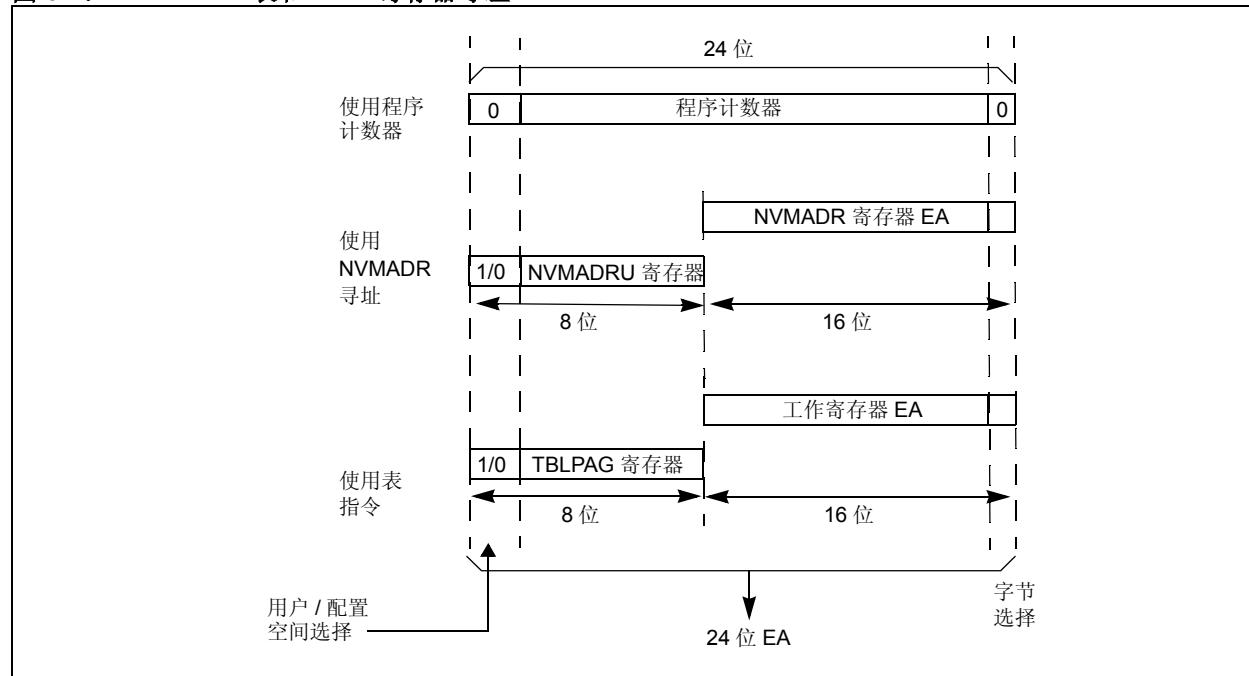
dsPIC30F 系列器件包含用于执行用户代码的内部闪存程序存储器。用户可以使用以下两种方法对闪存程序存储器编程：

1. 运行时自编程 (RTSP)
2. 在线串行编程 (ICSP)

6.1 在线串行编程 (ICSP)

dsPIC30F 器件可以在最终的应用电路中进行串行编程。只需要 5 根线即可完成这一操作，其中编程时钟线和编程数据线（名称分别为 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)。这允许用户在生产电路板时使用未编程器件，仅在产品交付之前才对数字信号控制器进行编程，从而可以使用最新版本的固件或者定制固件进行编程。

图 6-1：表和 NVM 寄存器寻址



6.2 运行时自编程 (RTSP)

使用 TBLRD (表读) 和 TBLWT (表写) 指令来实现 RTSP。

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。

6.3 表指令操作汇总

TBLRDL 和 TBLWTL 指令用来读或写程序存储器的 bit<15:0>。TBLRDL 和 TBLWTL 可以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 可以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 (EA)，形成 24 位的程序存储器地址，如图 6-1 所示。

6.4 RTSP 操作

dsPIC30F 闪存程序存储器组织成行和板（panel）。每行由 32 个指令字（96 字节）组成。每个板由 128 行（ $4K \times 24$ 个指令字）组成。RTSP 允许用户每次擦除一行（32 个指令字）、每次编程 4 条指令。RTSP 可以用来对多个程序存储器板进行编程，但在每个板的边界处必须更改表指针。

程序存储器的每个板都包含写锁存器，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入到写锁存器中：指令 0，指令 1，依此类推。装入的指令字必须始终来自 32 个指令字的地址边界。

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，装入写锁存器。通过把 NVMCON 寄存器中的特定位置 1，进行编程。装入 32 条指令需要 32 条 TBLWTL 和 4 条 TBLWTH 指令。如果需要对多个板进行编程，应该为下一组要写入的多个写锁存器修改表指针。

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）。编程每一行需要一个编程周期。

在整个 VDD 范围内，正常操作期间，闪存程序存储器都是可读写可擦除的。

6.5 控制寄存器

4 个特殊功能寄存器用来读、写闪存程序存储器，它们是：

- NVMCON
- NVMAADR
- NVMAADRU
- NVMKEY

6.5.1 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。

6.5.2 NVMAADR 寄存器

NVMAADR 寄存器用来存放有效地址的两个低字节。它捕捉已执行的上一条表指令的 EA<15:0>，选择要写的行。

6.5.3 NVMAADRU 寄存器

NVMAADRU 寄存器用来存放有效地址的高字节。它捕捉已执行的上一条表指令的 EA<23:16>。

6.5.4 NVMKEY 寄存器

NVMKEY 是只写寄存器，用于写保护。要启动编程或擦除过程，用户必须把 0x55 和 0xAA 连续写入 NVMKEY 寄存器。更多详细信息，请参见第 6.6 节“编程操作”。

注： 用户也可直接写 NVMAADR 和 NVMAADRU 寄存器，以指定擦除或编程的程序存储器地址。

6.6 编程操作

在 RTSP 模式下，对内部闪存进行编程或擦除需要完整的编程过程。编程操作持续时间的标称值为 2 ms，编程操作结束之前，处理器将暂停（等待）。将 WR 位（NVMCON<15>）置 1 开始编程操作，操作结束后 WR 位会自动清零。

6.6.1 闪存程序存储器的编程算法

用户一次可以擦除或编程闪存程序存储器的一行。一般过程如下：

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行。
 - 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/NVMADR。
 - 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。

例 6-1：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          ; ; Init NVMCON SFR
    MOV    W0,NVMCON
; Init pointer to row to be ERASED
    MOV    #tblpage(PROG_ADDR),W0   ; ; Initialize PM Page Boundary SFR
    MOV    W0,NVMADRU            ; ; Intialize in-page EA[15:0] pointer
    MOV    #tbloffset(PROG_ADDR),W0 ; ; Initialize NVMADR SFR
    MOV    W0,NVMADR             ; ; Block all interrupts with priority <7 for
    DISI   #5                   ; ; next 5 instructions
                                ; ; Write the 0x55 key
    MOV    #0x55,W0
    MOV    W0,NVMKEY
    MOV    #0xAA,W1
    MOV    W1,NVMKEY
    BSET  NVMCON,#WR           ; ; Start the erase sequence
    NOP
    NOP                         ; ; Insert two NOPs after the erase
                                ; ; command is asserted
```

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。

6.6.2 擦除程序存储器的一行

例 6-1 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）。

dsPIC30F5011/5013

6.6.3 装入写锁存器

例 6-2 所示的指令序列可以用来装入 96 字节的写锁存器。装入由表指针选择的写锁存器，需要 32 条 TBLWTL 指令和 32 条 TBLWTH 指令。

例 6-2: 装入写锁存器

```
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
    MOV      #0x0000,W0          ;
    MOV      W0,TBLPG           ; 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_BYT_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_BYT_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_BYT_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_BYT_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 6-2 中，W3 高位字节的内容将不起作用。

6.6.4 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，用户必须等待一段编程时间，直至编程完成。启动编程序列开始后紧跟的两条指令应该为 NOP。

例 6-3: 启动编程过程

```
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                ; Insert two NOPs after the erase
NOP                ; command is asserted
```

表 6-1: 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	所有复位时的状态
NVMCON	0760	WR	WREN	WRERR	—	—	—	—	TWRI	—	PROGOP<6:0>							0000 0000 0000 0000
NVMADR	0762										NVMADR<15:0>							uuuu uuuu uuuu uuuu
NVMADRU	0764	—	—	—	—	—	—	—	—		NVMADR<23:16>							0000 0000 uuuu uuuu
NVMKEY	0766	—	—	—	—	—	—	—	—		KEY<7:0>							0000 0000 0000 0000

图注: u = 未初始化位

注: 有关寄存器各位的说明, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

dsPIC30F5011/5013

注：

7.0 数据 EEPROM 存储器

注: 本数据手册总结了 dsPIC30F 系列器件的功能, 但是不应把本手册当作无所不包的参考手册来使用。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)。

在整个 V_{DD} 范围内的正常操作期间, 数据 EEPROM 存储器是可读写的。数据 EEPROM 存储器直接映射到程序存储器地址空间。

用来读写闪存程序存储器的 4 个 SFR, 也用来访问数据 EEPROM 存储器。如在第 6.5 节“控制寄存器”中所述, 这些寄存器是:

- NVMCON
- NVMADR
- NVMADRU
- NVMKEY

EEPROM 数据存储器允许读写单个字和 16 字长的块。针对数据存储器时, NVMADR 与 NVMADRU 寄存器配合使用, 用来寻址要访问的 EEPROM 存储单元。TBLRDL 和 TBLWTL 指令用来读写数据 EEPROM。dsPIC30F 器件有最多 8 KB (4K 字) 的数据 EEPROM, 地址范围从 0x7FF000 到 0xFFFFE。

字写操作之前, 应该先擦除相应的存储单元。写操作完成通常需要 2 ms, 不过它将随电压和温度的变化而变化。

对数据 EEPROM 进行编程或擦除操作, 不会停止指令流。在启动另一次数据 EEPROM 写 / 擦除操作之前, 用户需要等待一段适当的时间。在编程或擦除操作进行过程中, 尝试读数据 EEPROM 将返回不确定的数据。

控制位 WR 启动写操作, 这与闪存程序存储器写类似。用软件只能将该位置 1 而无法清零。写操作完成时, 由硬件将其清零。由于无法用软件将 WR 位清零, 可避免意外或提早结束写操作。

当 WREN 位置 1 时, 允许进行写操作。上电时, WREN 位被清零。当写操作被正常工作期间的 MCLR 复位或 WDT 超时复位所中断时, WRERR 位置 1。在这些情况下, 复位后用户可以检查 WRERR 位并重写存储单元。地址寄存器 NVMADR 保持不变。

注: 在写操作完成时, IFS0 寄存器中的中断标志位 NVMIF 将置 1。它必须用软件清零。

7.1 读数据 EEPROM

TBLRD 指令读取位于当前程序字地址的字。下面的示例使用 W0 作为指向数据 EEPROM 的指针。结果存放在寄存器 W4 中, 如例 7-1 所示。

例 7-1: 读数据 EEPROM

```
MOV    #LOW_ADDR_WORD,W0 ; Init Pointer  
MOV    #HIGH_ADDR_WORD,W1  
MOV    W1,TBLPAG  
TBLRDL [ W0 ], W4           ; read data EEPROM
```

7.2 擦除数据 EEPROM

7.2.1 擦除数据 EEPROM 的一块

为了擦除数据 EEPROM 的一块，首先 NVMADR 和 NVMADRU 寄存器必须指向要擦除的存储块。将 NVMCON 配置为擦除数据 EEPROM 块，并将 NVMCON 寄存器中的 ERASE 位和 WREN 位置 1。将 WR 位置 1，启动擦除操作，如例 7-2 所示。

例 7-2：擦除数据 EEPROM 的一块

```
; Select data EEPROM block, ERASE, WREN bits
MOV    #0x4045,W0
MOV    W0,NVMCON           ; Initialize NVMCON SFR

; Start erase cycle by setting WR after writing key sequence
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          ; Initiate erase sequence
NOP
NOP

; Erase cycle will complete in 2mS.CPU is not stalled for the Data Erase Cycle
; User can poll WR bit, use NVMIF or Timer IRQ to determine erasure complete
```

7.2.2 擦除数据 EEPROM 的一个字

TBLPAG 和 NVMADR 寄存器必须指向存储块。在 NVMCON 寄存器中，选择擦除闪存数据块，将 ERASE 位和 WREN 位置 1。将 WR 位置 1，启动擦除操作，如例 7-3 所示。

例 7-3：擦除数据 EEPROM 的一个字

```
; Select data EEPROM word, ERASE, WREN bits
MOV    #0x4044,W0
MOV    W0,NVMCON

; Start erase cycle by setting WR after writing key sequence
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          ; Initiate erase sequence
NOP
NOP

; Erase cycle will complete in 2mS.CPU is not stalled for the Data Erase Cycle
; User can poll WR bit, use NVMIF or Timer IRQ to determine erasure complete
```

7.3 写数据 EEPROM

要写 EEPROM 数据存储单元，必须按照下列顺序进行：

1. 擦除数据 EEPROM 的字。
 - a) 在 NVMCON 寄存器中，选择字擦除数据 EEPROM，并将 WREN 位置 1。
 - b) 把要擦除的字的地址写入 NVMADR。
 - c) 允许 NVM 中断（可选）。
 - d) 把“55”写入 NVMKEY。
 - e) 把“AA”写入 NVMKEY。
 - f) 将 WR 位置 1。这将开始擦除周期。
 - g) 查询 NVMIF 位，或者等待 NVMIF 中断。
 - h) 当擦除周期结束时 WR 位会被清零。
2. 把数据字写入数据 EEPROM 的写锁存器中。
3. 把 1 个数据字编程到数据 EEPROM 中。
 - a) 在 NVMCON 寄存器中，选择字编程数据 EEPROM，并将 WREN 位置 1。
 - b) 允许 NVM 写完成中断（可选）。
 - c) 把“55”写入 NVMKEY。
 - d) 把“AA”写入 NVMKEY。
 - e) 将 WR 位置 1。这将开始编程周期。
 - f) 查询 NVMIF 位，或者等待 NVM 中断。
 - g) 当写周期结束时 WR 位会被清零。

例 7-4: 数据 EEPROM 字写入

```

; Point to data memory
MOV      #LOW_ADDR_WORD,W0           ; Init pointer
MOV      #HIGH_ADDR_WORD,W1
MOV     W1,TBLPAG
MOV      #LOW(WORD),W2               ; Get data
TBLWTL   W2,[ W0 ]                 ; Write data

; The NVMADR captures last table access address
; Select data EEPROM for 1 word op
MOV      #0x4004,W0
MOV      W0,NVMCON

; Operate key to allow write operation
DISI    #5                         ; Block all interrupts with priority <7 for
                                    ; next 5 instructions
MOV      #0x55,W0
MOV      W0,NVMKEY
MOV      #0xAA,W1
MOV      W1,NVMKEY
BSET    NVMCON,#WR                 ; Write the 0x55 key
                                    ; Write the 0xAA key
                                    ; Initiate program sequence
NOP
NOP

; Write cycle will complete in 2mS.CPU is not stalled for the Data Write Cycle
; User can poll WR bit, use NVMIF or Timer IRQ to determine write complete

```

对于每个字，如果上述序列（把 0x55 写入 NVMKEY，把 0xAA 写入 NVMCON，然后将 WR 位置 1）没有严格遵守的话，写操作不会启动。强烈建议在这段代码执行期间禁止中断。

此外，必须将 NVMCON 中的 WREN 位置 1 来使能写操作。这种机制可防止由于意外执行代码导致误写数据 EEPROM。除非在更新 EEPROM 时，否则 WREN 位应始终保持清零。WREN 位不能用硬件清零。

在写过程启动之后，清零 WREN 位将不会影响当前的写周期。除非将 WREN 位置 1，否则将禁止 WR 位被置 1。WREN 位必须在前一条指令中被置 1。WR 和 WREN 不能在同一条指令中置 1。

写周期结束时，硬件清零 WR 位，非易失性存储器写完成中断标志位（NVMIF）置 1。用户可以允许此中断或查询此位。NVMIF 必须用软件清零。

7.3.1 写数据 EEPROM 的一个字

一旦用户擦除了要编程的字，则表写指令用来写写锁存器，如例 7-4 所示。

7.3.2 写数据 EEPROM 的一个存储块

要写入数据 EEPROM 的一个存储块，应首先写入所有 16 个锁存器，然后设置 NVMCON 寄存器并对存储块进行编程。

例 7-5: 数据 EEPROM 块写入

```
MOV      #LOW_ADDR_WORD,W0 ; Init pointer
MOV      #HIGH_ADDR_WORD,W1
MOV      W1,TBLPG
MOV      #data1,W2          ; Get 1st data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data2,W2          ; Get 2nd data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data3,W2          ; Get 3rd data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data4,W2          ; Get 4th data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data5,W2          ; Get 5th data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data6,W2          ; Get 6th data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data7,W2          ; Get 7th data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data8,W2          ; Get 8th data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data9,W2          ; Get 9th data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data10,W2         ; Get 10th data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data11,W2         ; Get 11th data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data12,W2         ; Get 12th data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data13,W2         ; Get 13th data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data14,W2         ; Get 14th data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data15,W2         ; Get 15th data
TBLWTL  W2,[ W0]++        ; write data
MOV      #data16,W2         ; Get 16th data
TBLWTL  W2,[ W0]++        ; write data.The NVMADR captures last table access address.
MOV      #0x400A,W0
MOV      W0,NVMCON          ; Select data EEPROM for multi word op
MOV      W0,NVMCON          ; Operate Key to allow program operation
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 write cycle
NOP
NOP
```

7.4 写校验

根据具体应用，将写入存储器的值对照原始值进行校验是一个很好的编程习惯。在应用中，如果某些位的写次数接近规定极限值，就应该进行写校验。

7.5 防止误写操作的保护措施

有些情况下，器件不宜向数据 EEPROM 存储器写入数据。器件内置了各种机制来防止误写入 EEPROM。上电时，**WREN** 位清零；而且，上电延时定时器禁止 EEPROM 写操作。

在欠压、电源故障或软件故障期间，写操作的启动序列以及 **WREN** 位可共同防止意外写操作的发生。

dsPIC30F5011/5013

注：

8.0 I/O 端口

注: 本数据手册总结了 dsPIC30F 系列器件的功能，但是不应把本手册当作无所不包的参考手册来使用。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/CLK1 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，以便增强抗干扰性。

8.1 并行 I/O (PIO) 端口

如果外设使能，并且外设正在使用相关引脚时，该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。

所有的端口引脚，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；写锁存值，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)。

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

当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，它将被视为专用端口，因为没有任何其他竞争的输出源。INT4 引脚就是一个例子。

PORTA 的寄存器的格式，如表 8-1 所示。

TRISA (数据方向控制) 寄存器控制 RA<7:0> 引脚的方向，以及 INTx 引脚和 VREF 引脚的方向。LATA 寄存器提供数据到输出，它是可读 / 可写的。读 PORTA 寄存器得到输入引脚的状态，而写 PORTA 寄存器修改 LATA 寄存器的内容。

与外设共用的引脚的并行 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 8-2 显示了端口是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 8-2 至表 8-9 给出了从 PORTB 到 PORTG 的共用端口的寄存器的格式。

注: 不同器件实际使用的位有所不同。

图 8-1: 专用端口结构框图

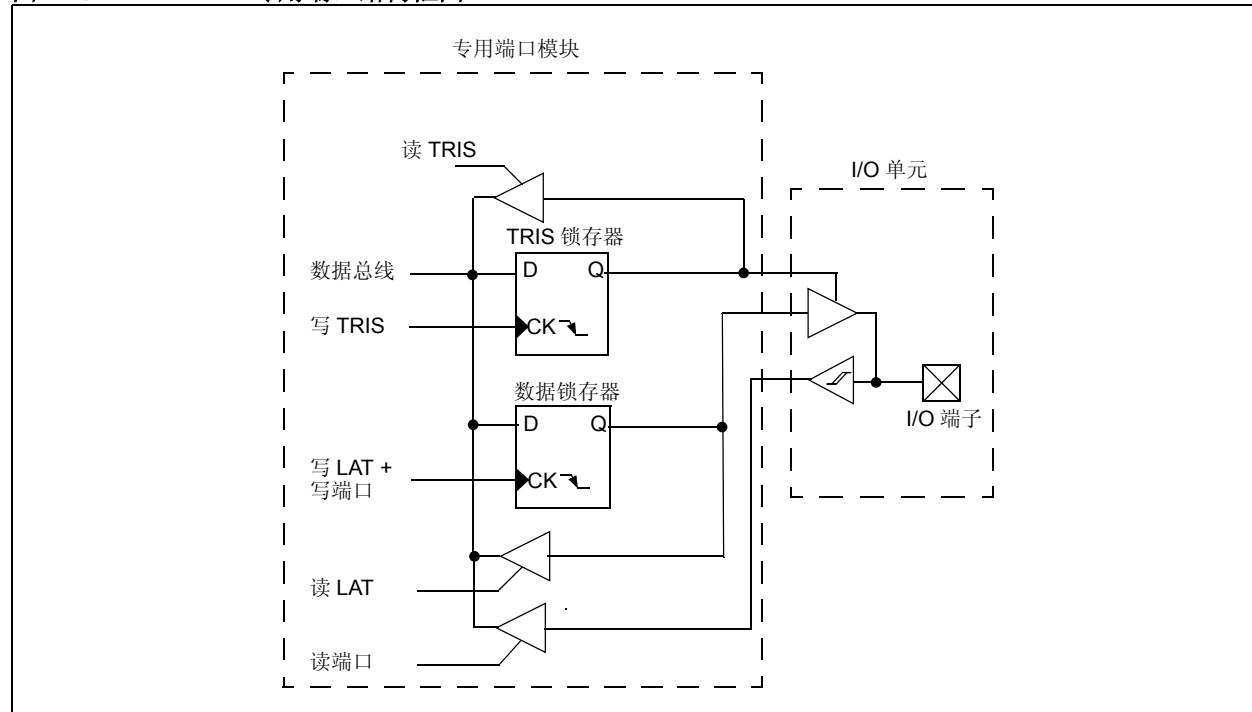
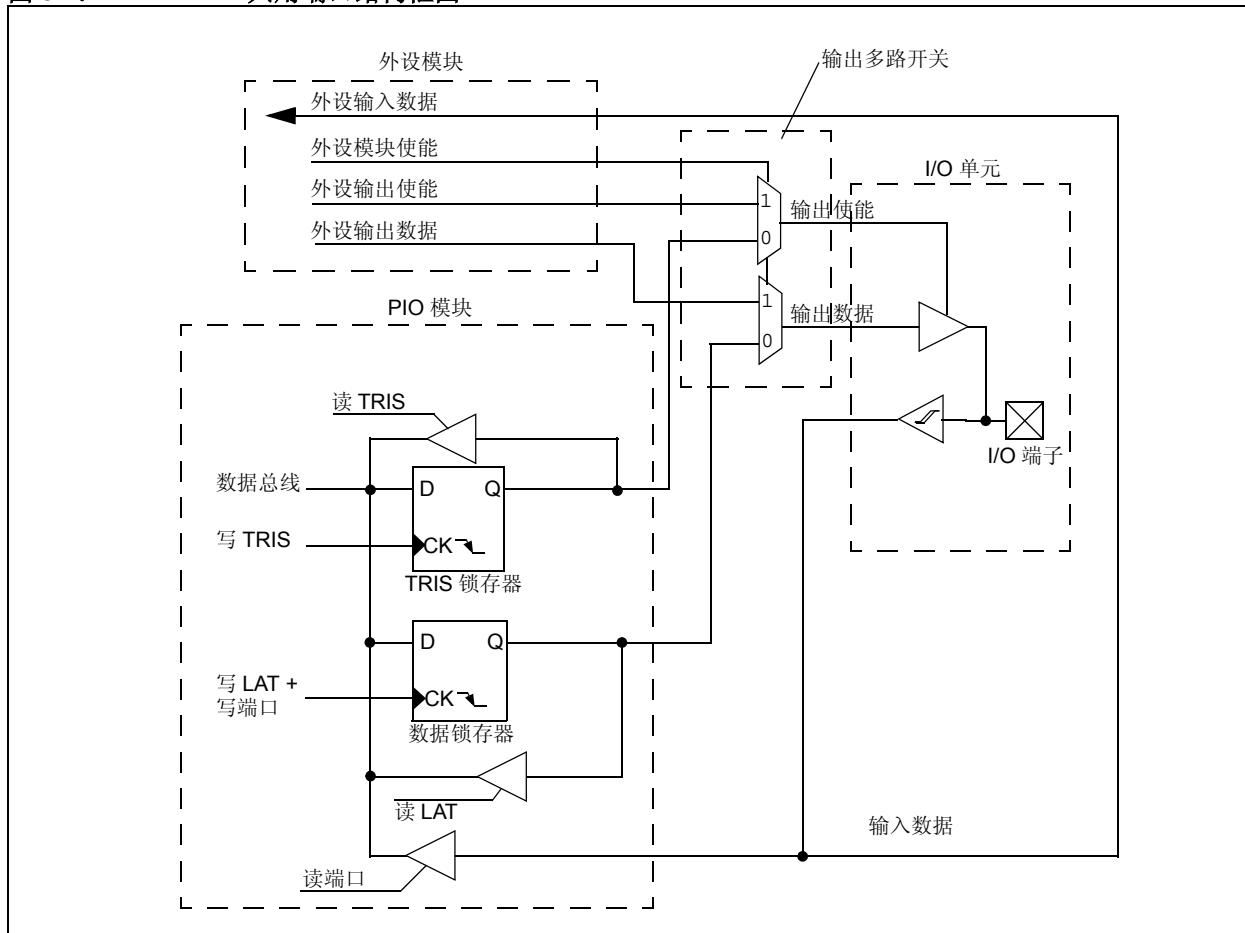


图 8-2: 共用端口结构框图



8.2 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。如果要将端口引脚用作模拟输入，则对应的 TRIS 位必须置 1（输入）。如果将 TRIS 位清零（输出），则数字输出电平（VOH 或 VOL）将被转换。

当读取端口寄存器时，所有配置为模拟输入通道的引脚均读为零（低电平）。

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

8.2.1 I/O 端口写 / 读时序

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

例 8-1: 端口写 / 读示例

```

MOV 0xFF00, W0 ; Configure PORTB<15:8>
; as inputs
MOV W0, TRISB ; and PORTB<7:0> as outputs
NOP           ; additional instruction
               ; cycle
btss PORTB, #13 ; bit test RB13 and skip if
                 ; set

```

表 8-1: dsPIC30F5013 的 PORTA 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	TRISA15	TRISA14	TRISA13	TRISA12	—	TRISA10	TRISA9	—	TRISA7	TRISA6	—	—	—	—	—	1111 0110 1100 0000	
PORTA	02C2	RA15	RA14	RA13	RA12	—	RA10	RA9	—	RA7	RA6	—	—	—	—	—	0000 0000 0000 0000	
LATA	02C4	LATA15	LATA14	LATA13	LATA12	—	LATA10	LATA9	—	LATA7	LATA6	—	—	—	—	—	0000 0000 0000 0000	

图注: u = 未初始化位

注: PORTA 在 dsPIC30F5011 器件中未实现。

表 8-2: dsPIC30F5011/5013 的 PORTB 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
TRISB	02C6	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111 1111 1111
PORTB	02C8	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	0000 0000 0000 0000
LATB	02CB	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000 0000 0000 0000

图注: u = 未初始化位

表 8-3: dsPIC30F5011 的 PORTC 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
TRISC	02CC	TRISC15	TRISC14	TRISC13	—	—	—	—	—	—	—	—	—	TRISC2	TRISC1	—	1110 0000 0000 0110	
PORTC	02CE	RC15	RC14	RC13	—	—	—	—	—	—	—	—	—	RC2	RC1	—	0000 0000 0000 0000	
LATC	02D0	LATC15	LATC14	LATC13	—	—	—	—	—	—	—	—	—	LATC2	LATC1	—	0000 0000 0000 0000	

图注: u = 未初始化位

表 8-4: dsPIC30F5013 的 PORTC 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
TRISC	02CC	TRISC15	TRISC14	TRISC13	—	—	—	—	—	—	—	—	TRISC4	TRISC3	TRISC2	TRISC1	—	1110 0000 0001 1110
PORTC	02CE	RC15	RC14	RC13	—	—	—	—	—	—	—	—	RC4	RC3	RC2	RC1	—	0000 0000 0000 0000
LATC	02D0	LATC15	LATC14	LATC13	—	—	—	—	—	—	—	—	LATC4	LATC3	LATC2	LATC1	—	0000 0000 0000 0000

图注: u = 未初始化位

表 8-5: dsPIC30F5011 的 PORTD 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
TRISD	02D2	—	—	—	—	TRISD11	TRISD10	TRISD9	TRISD8	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	0000 1111 1111 1111
PORTD	02D4	—	—	—	—	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	0000 0000 0000 0000
LATD	02D6	—	—	—	—	LATD11	LATD10	LATD9	LATD8	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	0000 0000 0000 0000

图注: u = 未初始化位

表 8-6: dsPIC30F5013 的 PORTD 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
TRISD	02D2	TRISD15	TRISD14	TRISD13	TRISD12	TRISD11	TRISD10	TRISD9	TRISD8	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111 1111 1111
PORTD	02D4	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	0000 0000 0000 0000
LATD	02D6	LATD15	LATD14	LATD13	LATD12	LATD11	LATD10	LATD9	LATD8	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	0000 0000 0000 0000

图注: u = 未初始化位

表 8-7: dsPIC30F5011 的 PORTF 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
TRISF	02DE	—	—	—	—	—	—	—	—	—	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	0000 0000 0111 1111
PORTF	02E0	—	—	—	—	—	—	—	—	—	RF6	RF5	RF4	RF3	RF2	RF1	RF0	0000 0000 0000 0000
LATF	02E2	—	—	—	—	—	—	—	—	—	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	0000 0000 0000 0000

图注: u = 未初始化位

表 8-8: dsPIC30F5013 的 PORTF 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	
TRISF	02DE	—	—	—	—	—	—	—	—	TRISF8	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	0000 0001 1111 1111
PORTF	02E0	—	—	—	—	—	—	—	—	RF8	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0	0000 0000 0000 0000
LATF	02E2	—	—	—	—	—	—	—	—	LATF8	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	0000 0000 0000 0000

图注: u = 未初始化位

表 8-9: dsPIC30F5011/5013 的 PORTG 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
TRISG	02E4	TRISG15	TRISG14	TRISG13	TRISG12	—	—	TRISG9	TRISG8	TRISG7	TRISG6	—	—	TRISG3	TRISG2	TRISG1	TRISG0	1111 0011 1100 1111
PORTG	02E6	RG15	RG14	RG13	RG12	—	—	RG9	RG8	RG7	RG6	—	—	RG3	RG2	RG1	RG0	0000 0000 0000 0000
LATG	02E8	LATG15	LATG14	LATG13	LATG12	—	—	LATG9	LATG8	LATG7	LATG6	—	—	LATG3	LATG2	LATG1	LATG0	0000 0000 0000 0000

图注: u = 未初始化位

注: 有关寄存器各位的描述, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

8.3 输入电平变化通知模块

输入电平变化通知模块使 dsPIC30F 器件能够向处理器发出中断请求，以响应所选输入引脚上的状态变化。即使是在休眠模式下、时钟被禁止时，该模块也能检测输入的状态变化。对于在状态变化时产生中断请求，可以选择（使能）的外部信号多达 24 个（CN0 到 CN23）。

表 8-10: dsPIC30F5011 的输入电平变化通知寄存器映射 (BIT 15-8)

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	复位状态
CNEN1	00C0	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	0000 0000 0000 0000
CNEN2	00C2	—	—	—	—	—	—	—	—	0000 0000 0000 0000
CNPU1	00C4	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	0000 0000 0000 0000
CNPU2	00C6	—	—	—	—	—	—	—	—	0000 0000 0000 0000

图注： u = 未初始化位

表 8-11: dsPIC30F5011 的输入电平变化通知寄存器映射 (BIT 7-0)

SFR 名称	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态
CNEN1	00C0	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000 0000 0000 0000
CNEN2	00C2	—	—	—	—	—	CN18IE	CN17IE	CN16IE	0000 0000 0000 0000
CNPU1	00C4	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000 0000 0000 0000
CNPU2	00C6	—	—	—	—	—	CN18PUE	CN17PUE	CN16PUE	0000 0000 0000 0000

图注： u = 未初始化位

表 8-12: dsPIC30F5013 的输入电平变化通知寄存器映射 (BIT 15-8)

SFR 名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	复位状态
CNEN1	00C0	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	0000 0000 0000 0000
CNEN2	00C2	—	—	—	—	—	—	—	—	0000 0000 0000 0000
CNPU1	00C4	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	0000 0000 0000 0000
CNPU2	00C6	—	—	—	—	—	—	—	—	0000 0000 0000 0000

图注： u = 未初始化位

表 8-13: dsPIC30F5013 的输入电平变化通知寄存器映射 (BIT 7-0)

SFR 名称	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位状态
CNEN1	00C0	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000 0000 0000 0000
CNEN2	00C2	CN23IE	CN22IE	CN21IE	CN20IE	CN19IE	CN18IE	CN17IE	CN16IE	0000 0000 0000 0000
CNPU1	00C4	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000 0000 0000 0000
CNPU2	00C6	CN23PUE	CN22PUE	CN21PUE	CN20PUE	CN19PUE	CN18PUE	CN17PUE	CN16PUE	0000 0000 0000 0000

图注： u = 未初始化位

注： 有关寄存器各位的说明，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

dsPIC30F5011/5013

注：

9.0 TIMER1 模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但是不应把本手册当作无所不包的参考手册来使用。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)。

本章介绍了 16 位通用 (GP) Timer1 模块以及相关的工作模式。图 9-1 给出了 16 位 Timer1 模块的简化框图。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以及相应的框图。

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

此外，支持如下工作特性：

- 定时器门控操作
- 可选择的预分频比设置
- 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或外部门控信号的下降沿产生中断

通过设定 16 位 SFR T1CON 中的相应位来决定这些工作模式。图 9-1 给出了 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，定时器在每个指令周期递增，直到与预先装入周期寄存器 PR1 中的值匹配，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 TSDL (T1CON<13>) 位 = 0。如果 TSDL = 1，定时器模块逻辑将继续递增，直到 CPU 空闲模式终止。

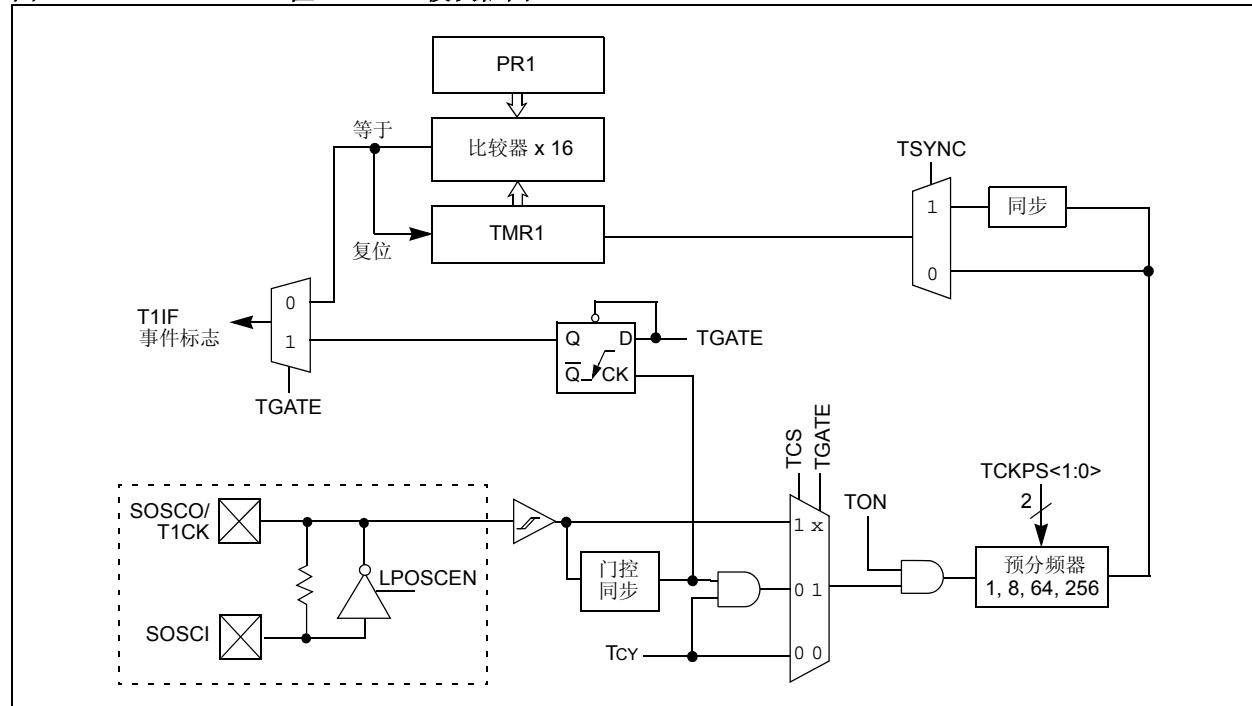
16 位同步计数器模式：在 16 位同步计数器模式下，定时器将在外部时钟信号的上升沿递增，外部时钟信号与内部相位时钟同步。定时器计数，直到等于 PR1 中预先装入的值，然后复位至 0，继续计数。

当 CPU 进入空闲模式时，定时器将停止递增，除非相应的 TSDL 位 = 0。如果 TSDL = 1，定时器模块逻辑将继续递增，直到 CPU 空闲模式终止。

16 位异步计数器模式：在 16 位异步计数器模式下，定时器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，CPU 进入空闲模式，如果 TSDL = 1，则定时器将停止递增。

图 9-1：16 位 TIMER1 模块框图



9.1 定时器门控操作

16位定时器可以置为门控时间累加模式。该模式使得当门控输入信号（T1CK 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 (T1CON<6>)。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)。

当 CPU 进入空闲模式时，定时器将停止递增，除非 TSIDL = 0。如果 TSIDL = 1，则定时器将在 CPU 空闲模式终止后继续递增。

9.2 定时器预分频器

16位定时器的输入时钟 (Fosc/4 或外部时钟)，有 1:1、1:8、1:64 或 1:256 的预分频比可供选择；通过控制位 TCKPS<1:0> (T1CON<5:4>) 来选择预分频比。当出现下列事件之一时，预分频器计数器将清零：

- 对 TMR1 寄存器进行写操作
- 对 T1CON 寄存器进行写操作
- 器件复位，如 POR 和 BOR

但是，如果定时器被禁止 (TON = 0)，由于预分频器的时钟停止了工作，定时器预分频器将不会被复位。

当写 T1CON 时，TMR1 不会被清零。当写 TMR1 寄存器时，TMR1 将被清零。

9.3 休眠模式下的定时器操作

在 CPU 休眠模式期间，定时器将继续工作，如果：

- 定时器模块被使能 (TON = 1)，且
- 定时器时钟源选用外部时钟 (TCS = 1)，且
- TSYNC 位 (T1CON<2>) 为逻辑 0，这将外部时钟源定义为异步的。

如果所有这三个条件都得到满足，定时器将继续计数，直到等于周期寄存器中的值，然后复位至 0x0000。

当定时器与周期寄存器之间匹配时，如果相应的定时器中断允许位置 1 的话，将产生中断。

9.4 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 位必须用软件清零。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。

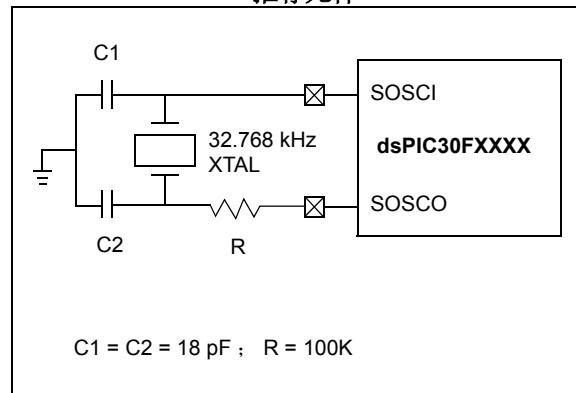
9.5 实时时钟

当 Timer1 工作在实时时钟 (RTC) 模式下时，提供了当天时间和事件时间标记的能力。RTC 的主要工作特性如下：

- 使用 32 kHz LP 振荡器工作
- 8 位预分频器
- 低功耗
- 实时时钟中断

这些工作模式由 T1CON 控制寄存器中相应位的设置决定。

图 9-2: TIMER1 LP 振荡器 RTC 的推荐元件



9.5.1 RTC 振荡器操作

当 **TON** = 1、**TCS** = 1 且 **TGATE** = 0 时，定时器在 32 kHz LP 振荡器输出信号的上升沿递增，直至递增到周期寄存器中指定的值，然后复位至 0。

为确保正确工作，**TSYNC** 位必须置为逻辑 0（异步模式）。

使能 **LPOSCEN** (**OSCCON<1>**) 将禁止正常的定时器和计数器模式，而使能定时器溢出唤醒事件。

当 CPU 进入休眠模式时，如果 32 kHz 外部晶体振荡器处于工作状态、且控制位没有改变的话，则 RTC 将继续工作。要使 RTC 在空闲模式下继续工作，**TSIDL** 位应当清零。

9.5.2 RTC 中断

当中断事件发生时，相应的中断标志 **T1IF** 将置 1，如果允许中断的话就将产生中断。**T1IF** 位必须用软件清零。相应的定时器中断标志 **T1IF** 位于中断控制器的 **IFS0** 状态寄存器中。

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

表 9-1: 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																uuuu uuuu uuuu uuuu	
PR1	0102																1111 1111 1111 1111	
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000 0000 0000 0000	

图注: u = 未初始化位

注: 有关寄存器各位的说明, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

10.0 TIMER2/3 模块

注: 本数据手册总结了 dsPIC30F 系列器件的功能, 但是不应把本手册当作无所不包的参考手册来使用。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)。

本章介绍了 32 位通用 (GP) 定时器模块 (Timer2/3) 以及相关的工作模式。图 10-1 给出了 32 位 Timer2/3 模块的简化框图。图 10-2 和图 10-3 分别为将 Timer2/3 配置成的两个独立 16 位定时器 Timer2 和 Timer3 的框图。

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, 下面的小节将详细说明其设置和控制寄存器以及相应的框图。

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置
- 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, 而 Timer3 是高位字。

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。这两种工作模式的详细信息, 请参见第 9.0 节 “Timer1 模块”。

Timer2 和 Timer3 之间唯一的功能差异是, 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 的内容将被传送并锁存到 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相位时钟同步。定时器计数, 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。

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

dsPIC30F5011/5013

图 10-1： 32 位 TIMER2/3 框图

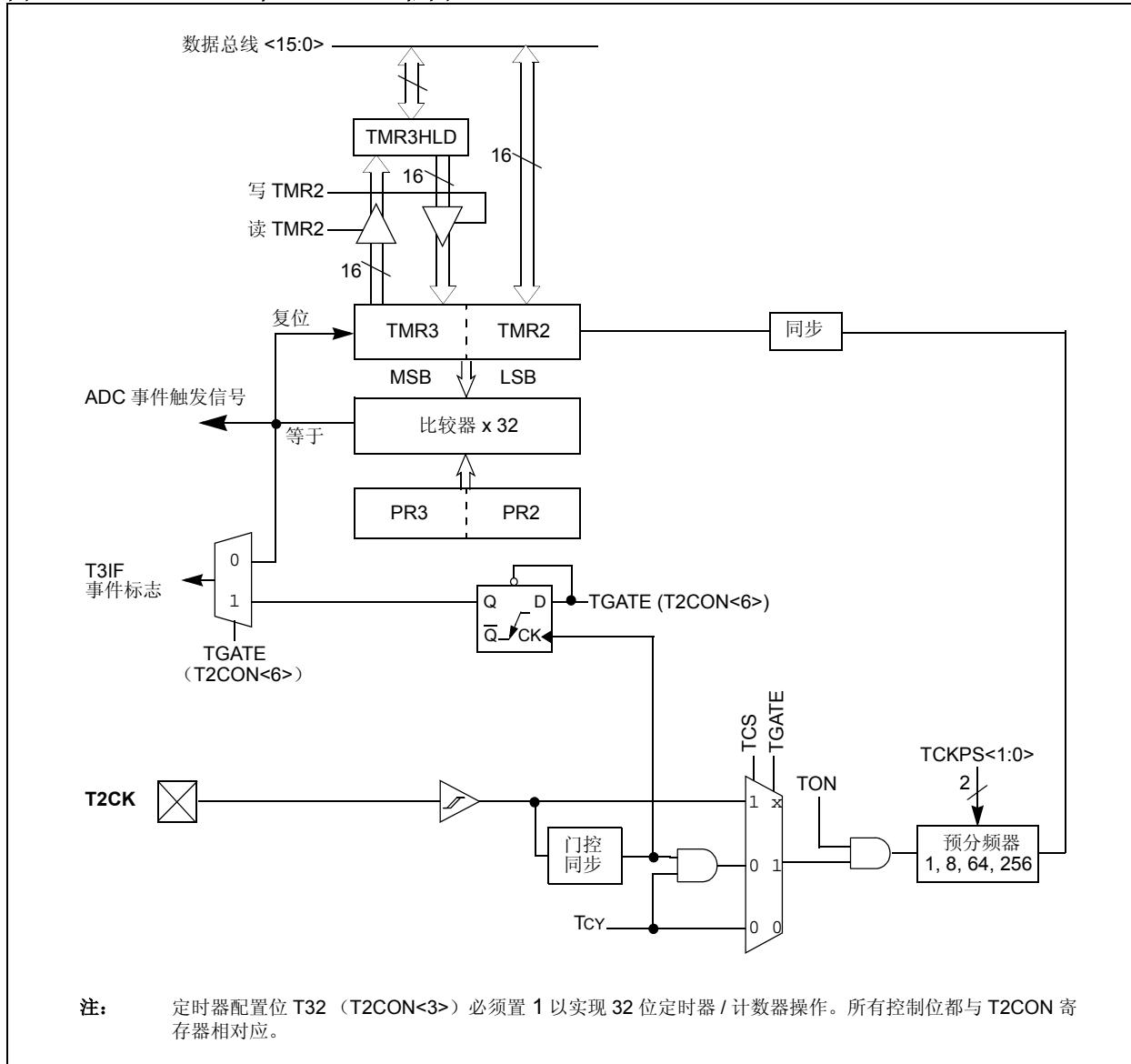


图 10-2: 16 位 TIMER2 框图

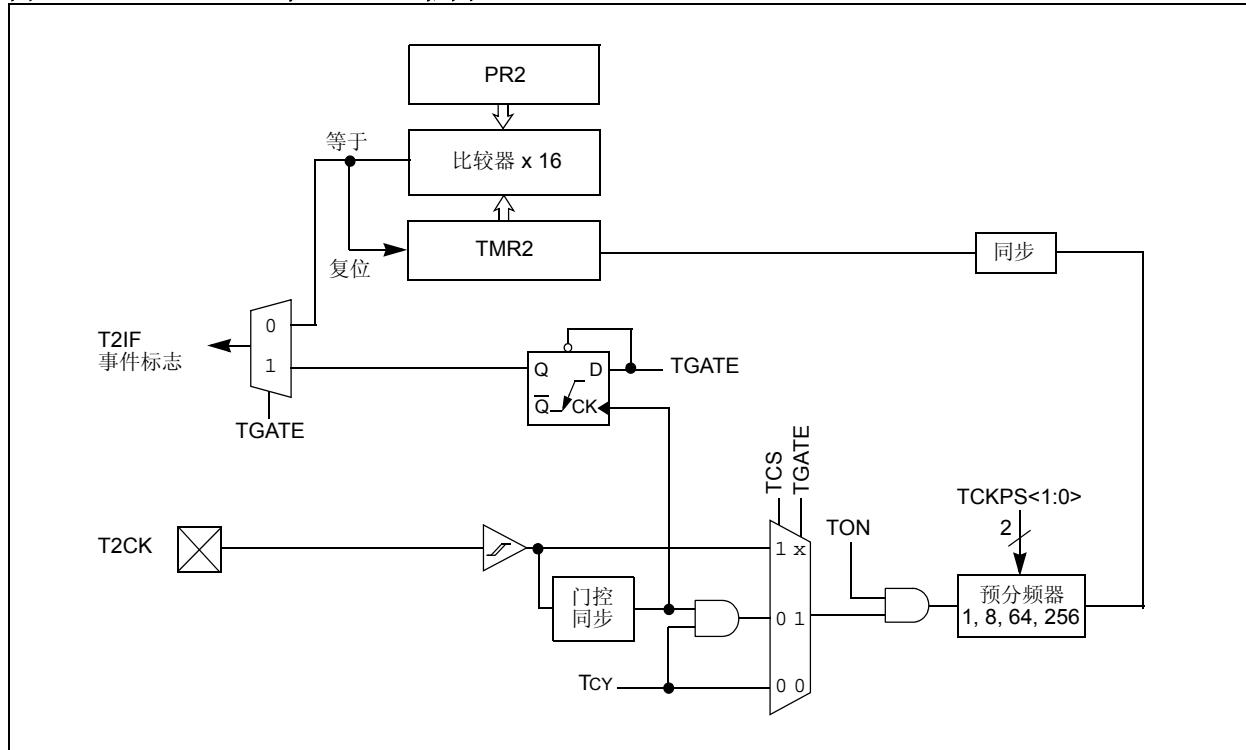
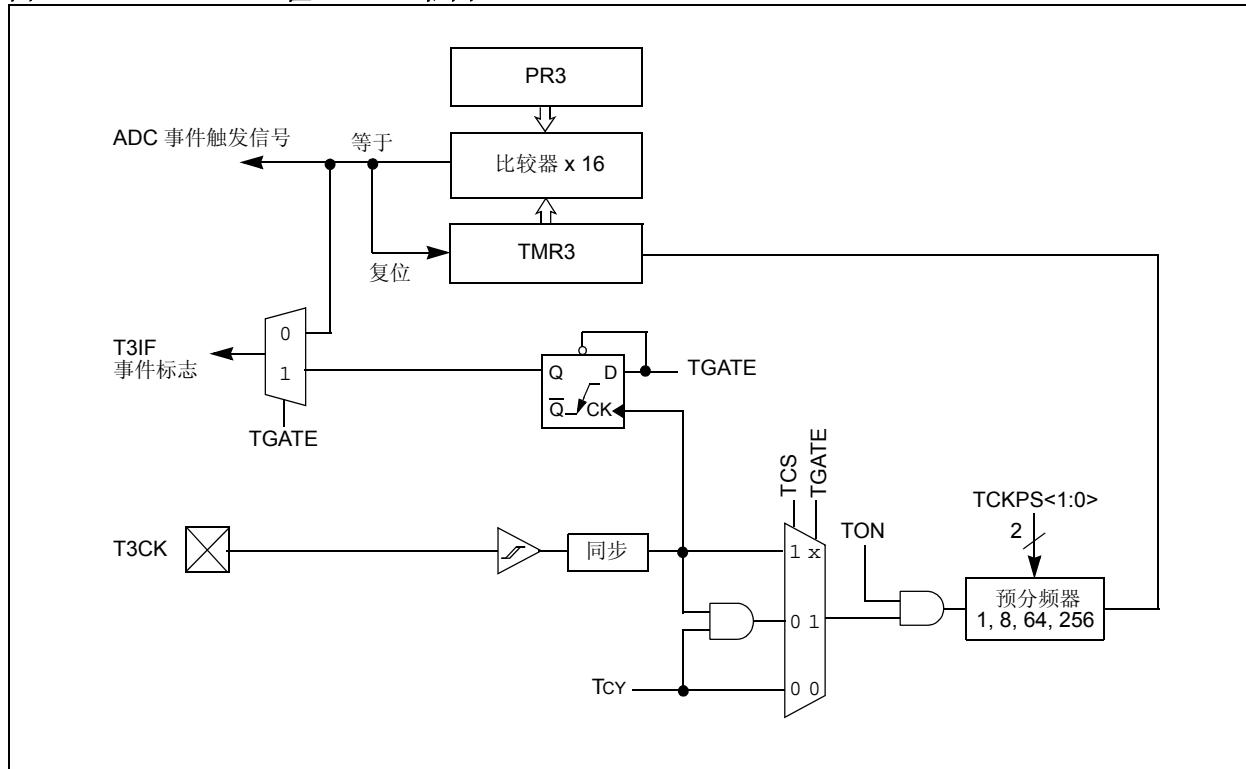


图 10-3: 16 位 TIMER3 框图



10.1 定时器门控操作

32位定时器可以置为门控时间累加模式。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）。

外部信号的下降沿终止计数操作，但不会复位定时器。要从零开始计数，用户必须复位定时器。

10.2 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，或当 16 位定时器 TMR3 与 16 位周期寄存器 PR3 匹配时，Timer3 将生成 ADC 特殊事件触发信号。

10.3 定时器预分频器

定时器的输入时钟（Fosc/4 或外部时钟），有 1:1、1:8、1:64 或 1:256 的预分频比选择；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，发起时钟源是 Timer2。在此模式下不能进行 Timer3 的预分频器操作。当出现下列事件之一时，预分频器计数器将清零：

- 写 TMR2/TMR3 寄存器
- 对 T2CON/T3CON 寄存器进行写操作
- 器件复位，如 POR 和 BOR

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

当写 T2CON/T3CON 时，TMR2/TMR3 不会被清零。

10.4 休眠模式下的定时器操作

在 CPU 休眠模式期间，由于禁止了内部时钟，所以定时器不工作。

10.5 定时器中断

32 位定时器模块具有在周期匹配时，或者在外部门控信号下降沿产生中断的能力。当 32 位定时器计数与相应的 32 位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，如果允许中断的话则将产生中断。在此模式下，T3IF 中断标志用作中断源。T3IF 位必须用软件清零。

通过相应的定时器中断允许位 T3IE（IEC0<7>）来允许中断。

表 10-1: 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	TCKPS1	TCKPS0	T32	—	TCS	—	0000 0000 0000 0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000 0000 0000 0000

图注: — = 未初始化位

注: 有关寄存器各位的说明, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

dsPIC30F5011/5013

注：

11.0 TIMER4/5 模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但是不应把本手册当作无所不包的参考手册来使用。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)。

本章介绍了另一个 32 位通用 (GP) 定时器模块 (Timer4/5) 以及相关的工作模式。图 11-1 给出了 32 位 Timer4/5 模块的简化框图。图 11-2 和图 11-3 分别为将 Timer4/5 配置成的两个独立 16 位定时器 Timer4 和 Timer5 的框图。

Timer4/5 模块的工作与 Timer2/3 模块相似。但是，也存在如下区别：

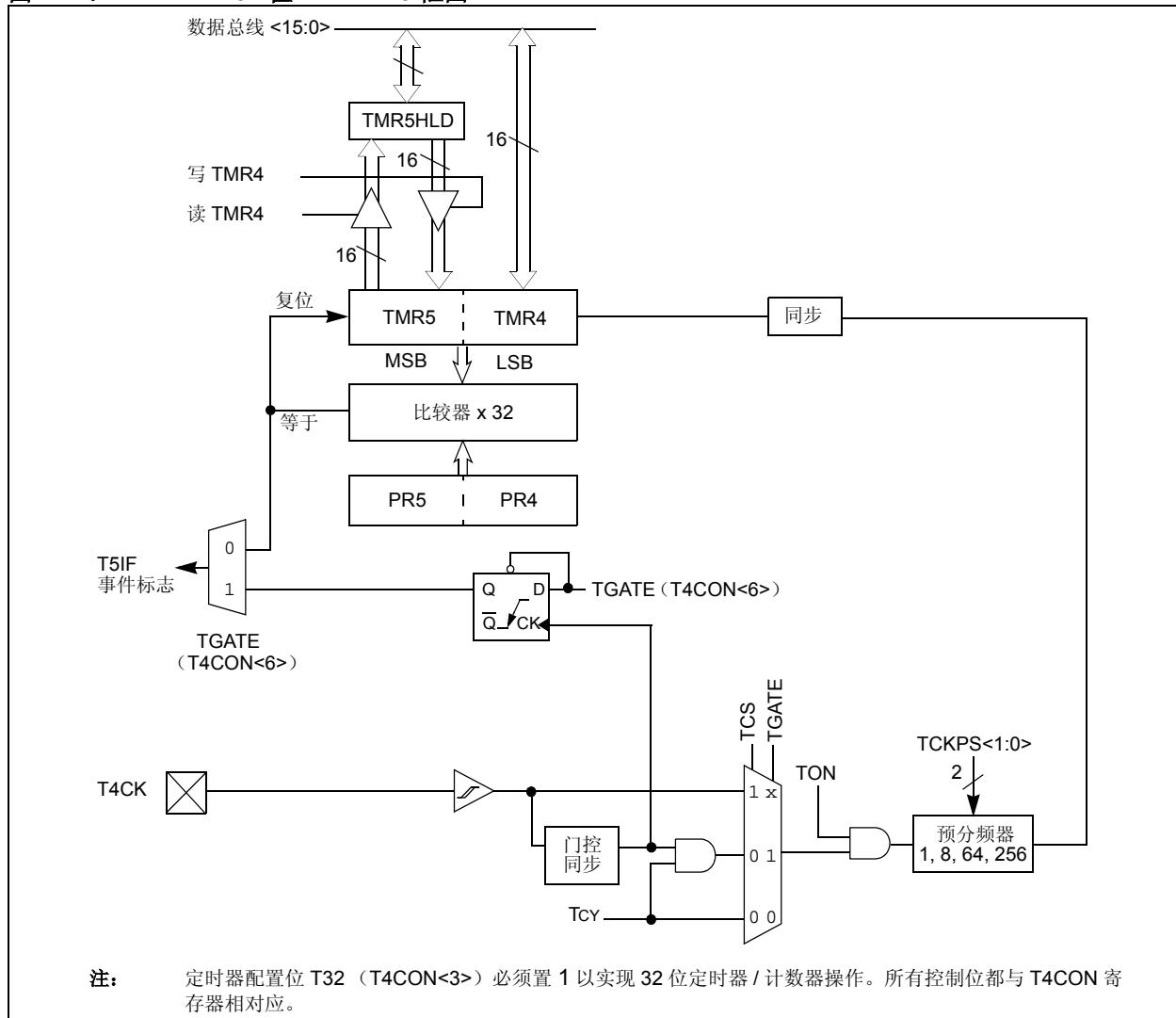
- Timer4/5 模块不支持 ADC 事件触发功能
- Timer4/5 不能为诸如输入捕捉和输出比较的其他外设模块所使用

通过设定 16 位 T4CON 和 T5CON SFR 中的相应位来决定 Timer4/5 模块的工作模式。

对于 32 位定时器 / 计数器操作，Timer4 是 32 位定时器的低位字，而 Timer5 是高位字。

注：对于 32 位定时器操作，T5CON 控制位将被忽略。设置和控制只使用 T4CON 控制位。对于 32 位定时器模块，使用的是 Timer4 时钟和门控输入，但是，中断产生由 Timer5 中断标志位 (T5IF) 反映，且中断通过 Timer5 中断允许位 (T5IE) 来允许。

图 11-1： 32 位 TIMER4/5 框图



dsPIC30F5011/5013

图 11-2: 16 位 TIMER4 框图

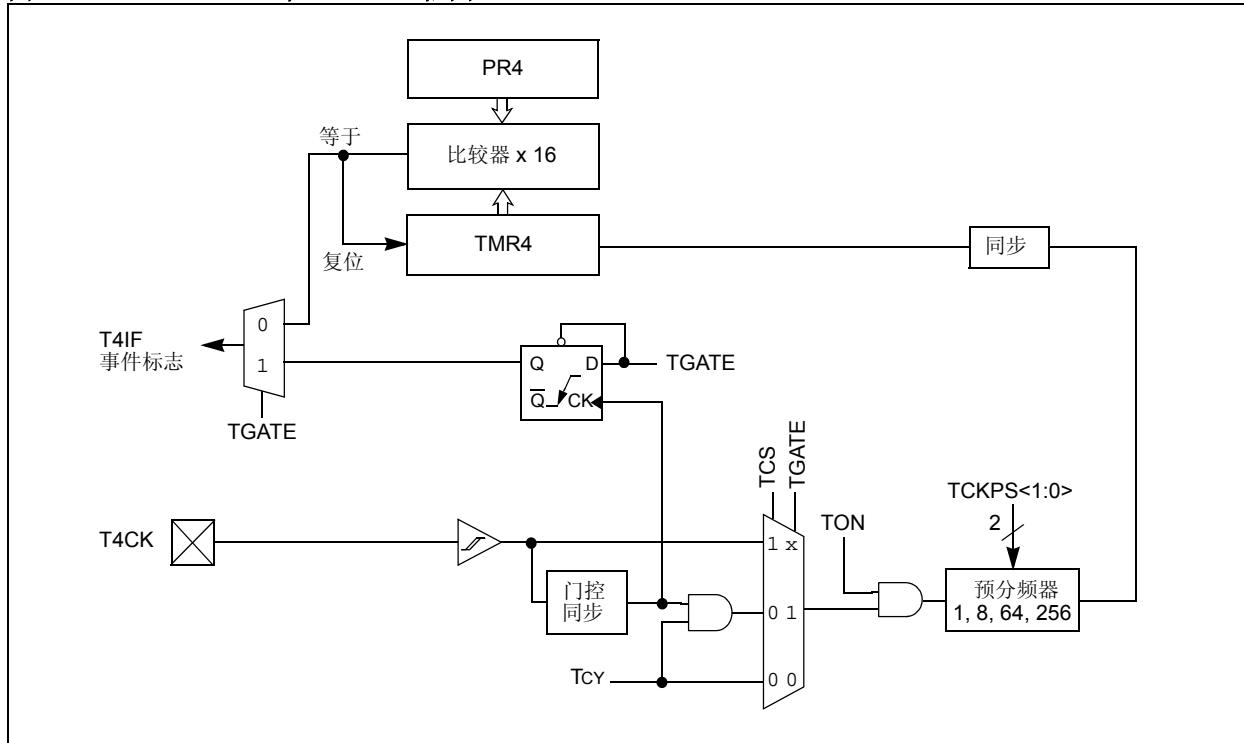
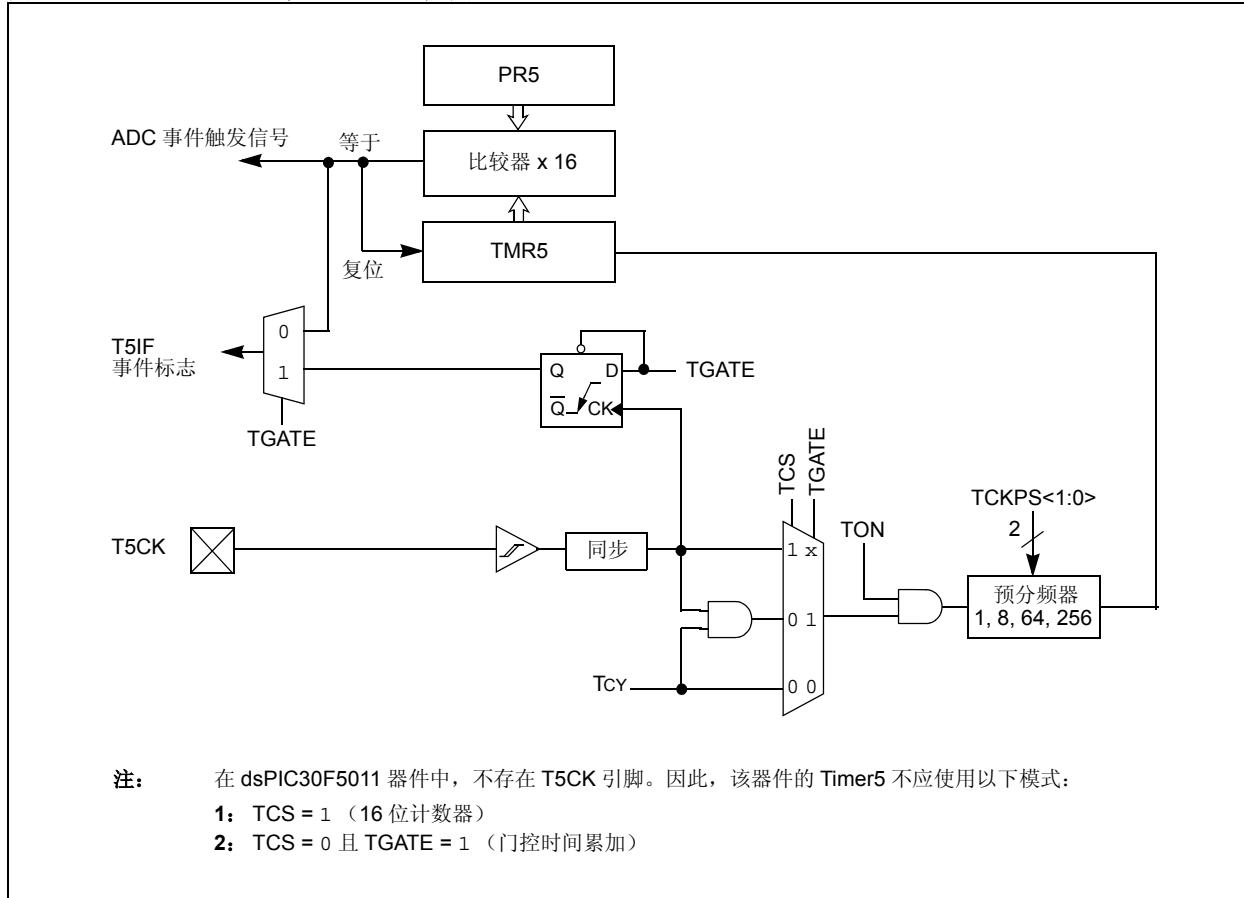


图 11-3: 16 位 TIMER5 框图



注：在 dsPIC30F5011 器件中，不存在 T5CK 引脚。因此，该器件的 Timer5 不应使用以下模式：

- 1: TCS = 1 (16 位计数器)
- 2: TCS = 0 且 TGATE = 1 (门控时间累加)

表 11-1: TIMER4/5 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
TMR4	0114	Timer4 寄存器														uuuu uuuu uuuu uuuu		
TMR5HLD	0116	Timer5 保持寄存器（仅适用于 32 位定时器操作）														uuuu uuuu uuuu uuuu		
TMR5	0118	Timer5 寄存器														uuuu uuuu uuuu uuuu		
PR4	011A	周期寄存器 4														1111 1111 1111 1111		
PR5	011C	周期寄存器 5														1111 1111 1111 1111		
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T45	—	TCS	—	0000 0000 0000 0000
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000 0000 0000 0000

图注: u = 未初始化位

dsPIC30F5011/5013

注：

12.0 输入捕捉模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但是不应把本手册当作无所不包的参考手册来使用。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)。

本章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 12-1 给出了输入捕捉模块的框图。输入捕捉功能可用于以下模式：

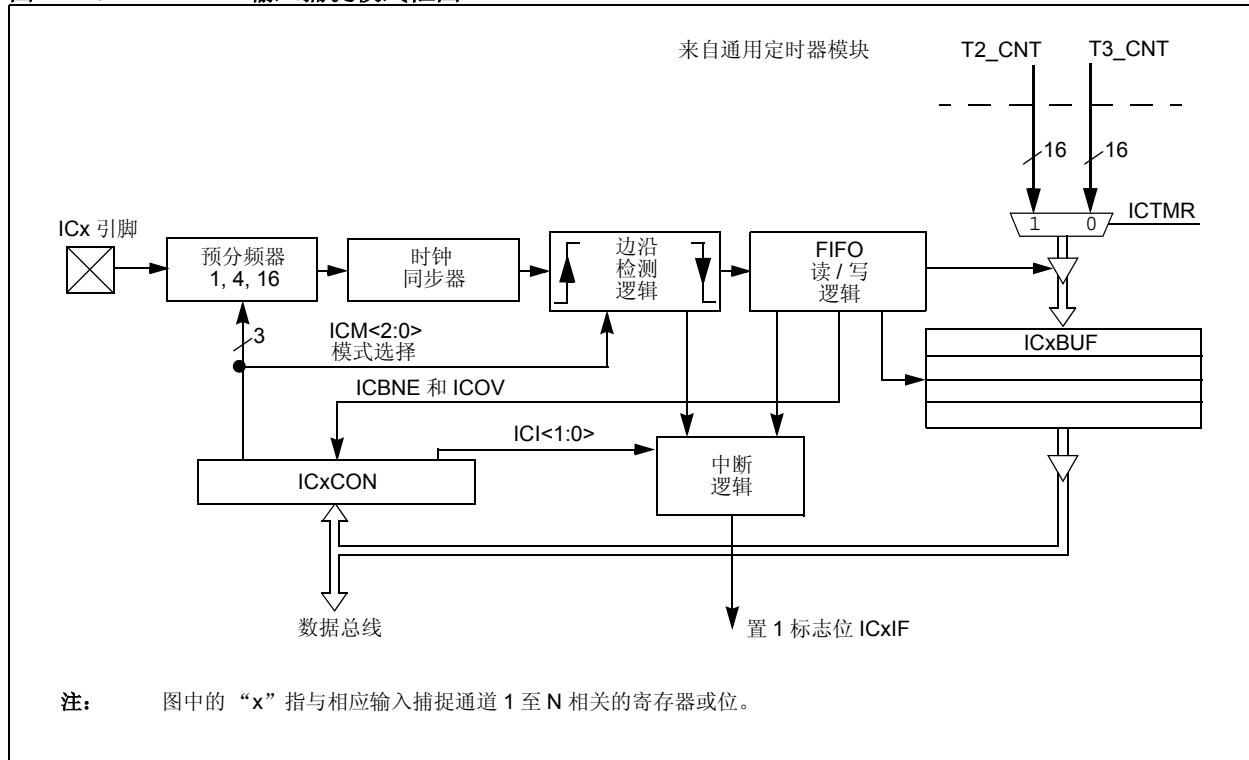
- 频率 / 周期 / 脉冲测量
- 其他外部中断源

输入捕捉模块的主要工作特性是：

- 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 IC_xCON 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 路捕捉通道（即，N 的最大值为 8）。

图 12-1： 输入捕捉模式框图



12.1 简单捕捉事件模式

dsPIC30F 产品系列中的简单捕捉事件有：

- 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> (IC_xCON<2:0>)，选择上述简单输入捕捉模式。

12.1.1 捕捉预分频器

有 4 种输入捕捉预分频器设置，它们由 ICM<2:0> (IC_xCON<2:0>) 位指定。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。

12.1.2 捕捉缓冲器操作

每个捕捉通道都有一个关联的 FIFO 缓冲器，该缓冲器宽 16 位，深度为 4 字。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。

如果 FIFO 已被 4 个捕捉事件填满，在读 FIFO 之前又发生第 5 个捕捉事件，则将产生溢出，且 ICOV 位将被设置为逻辑 1。第 5 个捕捉事件将丢失，不会存入 FIFO。在将 4 个事件全部从缓冲器中读出之前，不会再捕捉任何事件。

如果 FIFO 已全部读完，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。

12.1.3 TIMER2 和 TIMER3 选择模式

输入捕捉模块有最多 8 路输入捕捉通道。每个通道均可选择两个定时器（Timer2 或 Timer3）之一作为时基。

通过 SFR 位 ICTMR（ICxCON<7>）来选择定时器资源。对于输入捕捉模块，Timer3 是默认的可用定时器资源。

12.1.4 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以下操作：

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

12.2 休眠和空闲模式下的输入捕捉工作

当器件处于 CPU 空闲或休眠模式下时，输入捕捉事件将唤醒器件或产生中断（如果允许中断的话）。

如果 $ICM<2:0> = 111$ 且中断允许位置 1，那么当发生捕捉事件时，输入捕捉模块将使 CPU 从休眠或空闲模式唤醒，这与使能的定时器无关。如果已满足处理中断的条件，上述唤醒还将产生中断。唤醒特性可以用来增加额外的外部引脚中断。

12.2.1 CPU 休眠模式下的输入捕捉

CPU 休眠模式允许输入捕捉模块以功能受到限制的方式工作。在 CPU 休眠模式下， $ICI<1:0>$ 位不适用，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，必须将模块配置为仅在上升沿产生中断 ($ICM<2:0> = 111$)。在此模式下，4:1 或 16:1 的预分频比设置不适用。

12.2.2 CPU 空闲模式下的输入捕捉

CPU 空闲模式允许输入捕捉模块以全功能方式工作。在 CPU 空闲模式下，可以通过 $ICI<1:0>$ 位选择中断模式，还可以通过控制位 $ICM<2:0>$ 选择 4:1 和 16:1 捕捉预分频比设置。此模式要求使能所选择的定时器。此外，还必须将 ICSIDL 位设置为逻辑 0。

在 CPU 空闲模式下，如果将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。

12.3 输入捕捉中断

输入捕捉通道具有根据所选择的捕捉事件数来产生中断的能力。所选择的捕捉事件数可以通过控制位 $ICI<1:0>$ ($ICxCON<6:5>$) 进行设置。

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

通过各捕捉通道中断允许位 (TcxIE) 来允许中断。捕捉中断允许位位于相应的 IEC 控制寄存器中。

表 12-1：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																uuuu uuuu uuuu uuuu	
IC1CON	0142	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000	
IC2BUF	0144																uuuu uuuu uuuu uuuu	
IC2CON	0146	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000	
IC3BUF	0148																uuuu uuuu uuuu uuuu	
IC3CON	014A	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000	
IC4BUF	014C																uuuu uuuu uuuu uuuu	
IC4CON	014E	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000	
IC5BUF	0150																uuuu uuuu uuuu uuuu	
IC5CON	0152	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000	
IC6BUF	0154																uuuu uuuu uuuu uuuu	
IC6CON	0156	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000	
IC7BUF	0158																uuuu uuuu uuuu uuuu	
IC7CON	015A	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000	
IC8BUF	015C																uuuu uuuu uuuu uuuu	
IC8CON	015E	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>	ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000	

图注： u = 未初始化位

注： 有关寄存器各位的说明，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

dsPIC30F5011/5013

注：

13.0 输出比较模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但是不应把本手册当作无所不包的参考手册来使用。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)。

本章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 13-1 给出了输出比较模块的框图。

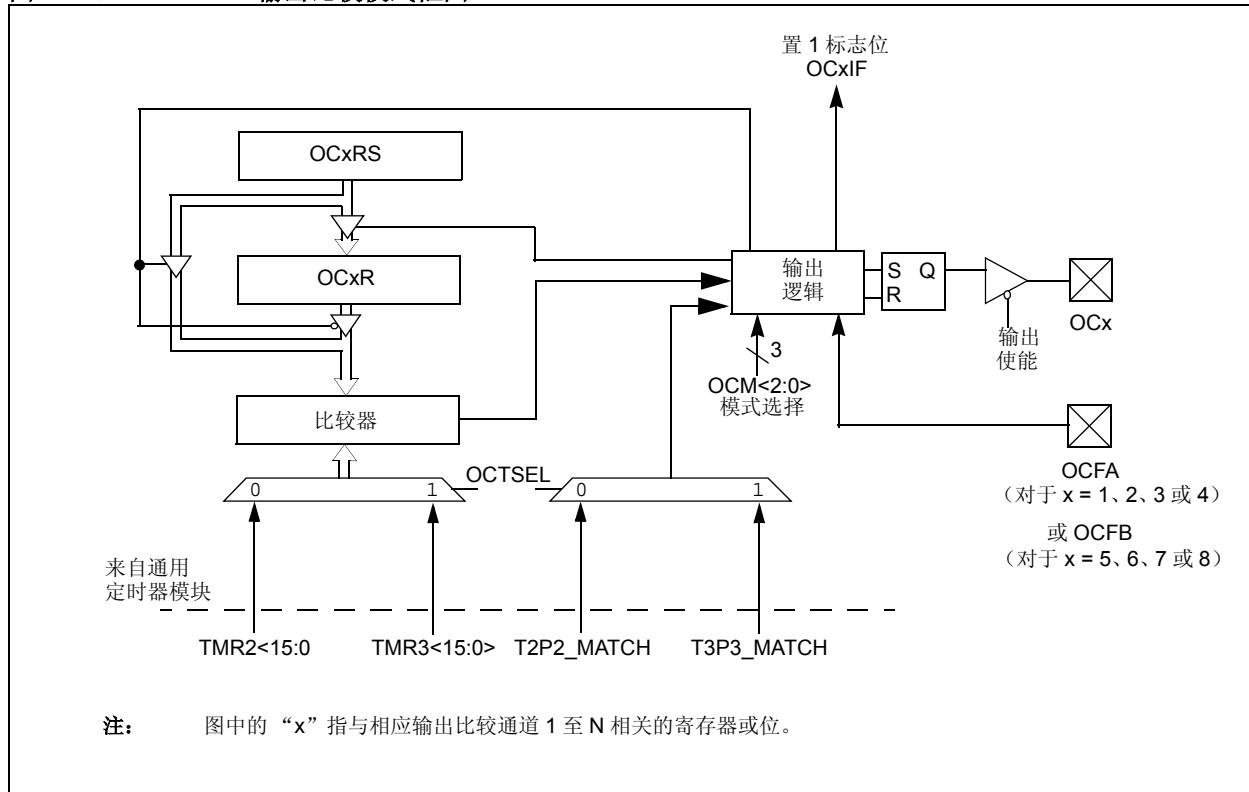
输出比较模块的主要工作特性如下：

- Timer2 和 Timer3 选择模式
- 简单输出比较匹配模式
- 双输出比较匹配模式
- 简单 PWM 模式
- 休眠和空闲模式期间的输出比较
- 发生输出比较 /PWM 事件时产生中断

通过设置 16 位 OCxCON SFR (其中 $x = 1, 2, 3, \dots, N$) 中的相应位来决定这些工作模式。dsPIC DSC 器件具有最多 8 路比较通道 (即, N 的最大值为 8)。

图 13-1 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 13-1：输出比较模式框图



13.1 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 或 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。

13.2 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以下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

上述模式中使用了 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。

13.3 双输出比较匹配模式

当控制位 OCM<2:0>（OCxCON<2:0>）= 100 或 101 时，所选输出比较通道被配置为以下两种双输出比较模式之一：

- 单输出脉冲模式
- 连续输出脉冲模式

13.3.1 单脉冲模式

用户将模块配置为生成单输出脉冲，需要执行以下步骤（假定定时器已关闭）：

- 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 到脉冲开始的时间。
- 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, 2, ..., N）。
- 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。

13.3.2 连续脉冲模式

用户将模块配置为生成连续输出脉冲，需要执行以下步骤：

- 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 到脉冲开始的时间。
- 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, 2, ..., N）。
- 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。

13.4 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 信号跳变时不会产生毛刺。

为将输出比较模块配置为 PWM 工作模式，用户必须执行以下步骤：

1. 通过写适当的周期寄存器，设置 PWM 周期。
2. 通过写 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 预分频比并使能定时器，
TON（TxCON<15>）= 1。

13.4.1 PWM 的输入引脚故障保护

当控制位 OCM<2:0>（OCxCON<2:0>）= 111 时，所选输出比较通道还是配置为 PWM 工作模式，但具有输入故障保护的附加功能。在此模式下，如果在 OCFA/B 引脚上检测到逻辑 0，则对应的 PWM 输出引脚将被置于高阻输入状态。OCFLT 位（OCxCON<4>）表明是否产生故障条件。此状态将保持到以下事件都已发生：

- 外部故障条件已经消除。
- 通过写适当的控制位，重新使能了 PWM 模式。

13.4.2 PWM 周期

通过写入 PR_x 寄存器可以指定 PWM 周期。可使用公式 13-1 计算 PWM 周期。

公式 13-1:

$$\text{PWM 周期} = \frac{[(\text{PR}_x) + 1] \cdot 4 \cdot \text{TOSC}}{(\text{TMR}_x \text{ 预分频值})}$$

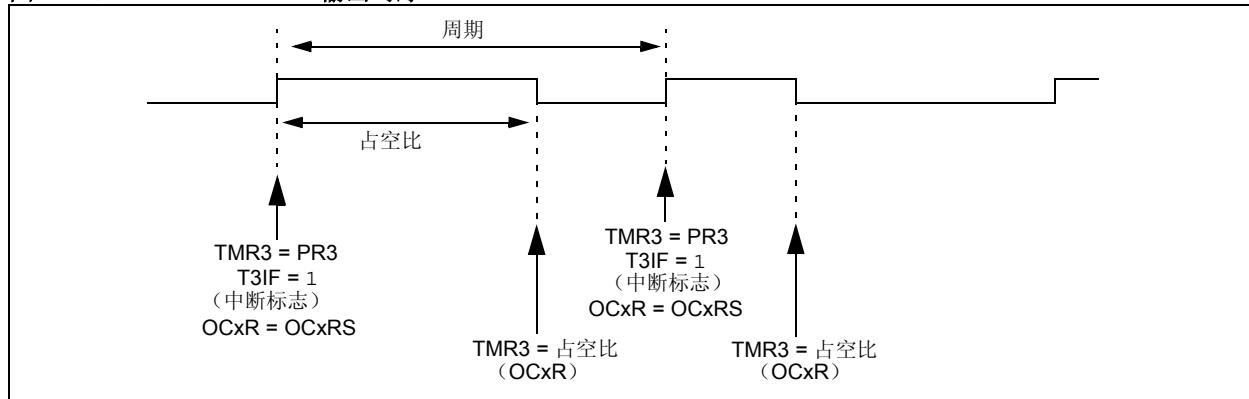
PWM 频率定义为 $1/\text{[PWM 周期]}$ 。

当所选 TMR_x 等于其对应的周期寄存器 PR_x 时，在下一个递增周期将发生以下 4 个事件：

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

关键 PWM 周期的比较，请参见图 13-2。为了清晰起见，图 13-2 中引用的是 Timer3。

图 13-2: PWM 输出时序



13.5 CPU 休眠模式期间的输出比较工作

当 CPU 进入休眠模式时，所有的内部时钟都将停止。从而，当 CPU 进入休眠状态时，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。类似地，当 CPU 进入休眠状态时，如果引脚为低电平，则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。

13.6 CPU 空闲模式期间的输出比较工作

当 CPU 进入空闲模式时，输出比较模块仍然能够以全功能工作。

如果 OCSIDL 位 ($\text{OC}_x\text{CON}<13>$) 为逻辑 0，所选时基 (Timer2 或 Timer3) 使能且所选定时器的 TSIDL 位为逻辑 0，则输出比较通道将在 CPU 空闲模式期间工作。

13.7 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 模式之外的所有模式，当比较事件发生时，相应的中断标志 (OC_xIF) 置 1，如果允许中断的话还将产生中断。 OC_xIF 位位于相应的 IFS 状态寄存器中，它必须用软件清零。通过相应的比较中断允许 (OC_xIE) 位来允许中断；该位位于相应的 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 (T2IF 或 T3IF) 置 1，如果允许中断的话还将产生中断。 IF 位位于 IFS0 状态寄存器中，必须用软件对它清零。通过相应的定时器中断允许位 (T2IE 或 T3IE) 来允许中断；该位位于 IEC0 控制寄存器中。在 PWM 工作模式下，输出比较中断标志始终不会置 1。

表 13-1：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																0000 0000 0000 0000	
OC1R	0182																0000 0000 0000 0000	
OC1CON	0184	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>	0000 0000 0000 0000		
OC2RS	0186																0000 0000 0000 0000	
OC2R	0188																0000 0000 0000 0000	
OC2CON	018A	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSE	OCM<2:0>	0000 0000 0000 0000		
OC3RS	018C																0000 0000 0000 0000	
OC3R	018E																0000 0000 0000 0000	
OC3CON	0190	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>	0000 0000 0000 0000		
OC4RS	0192																0000 0000 0000 0000	
OC4R	0194																0000 0000 0000 0000	
OC4CON	0196	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>	0000 0000 0000 0000		
OC5RS	0198																0000 0000 0000 0000	
OC5R	019A																0000 0000 0000 0000	
OC5CON	019C	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>	0000 0000 0000 0000		
OC6RS	019E																0000 0000 0000 0000	
OC6R	01A0																0000 0000 0000 0000	
OC6CON	01A2	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>	0000 0000 0000 0000		
OC7RS	01A4																0000 0000 0000 0000	
OC7R	01A6																0000 0000 0000 0000	
OC7CON	01A8	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>	0000 0000 0000 0000		
OC8RS	01AA																0000 0000 0000 0000	
OC8R	01AC																0000 0000 0000 0000	
OC8CON	01AE	—	—	OCSIDL	—	—	—	—	—	—	—	—	OCFLT	OCTSEL	OCM<2:0>	0000 0000 0000 0000		

图注：— = 未初始化位

注：有关寄存器各位的说明，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

14.0 SPI 模块

注: 本数据手册总结了 dsPIC30F 系列器件的功能,但是不应把本手册当作无所不包的参考手册来使用。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)。

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。

14.1 功能说明

每个 SPI 模块由一个用于将数据移入和移出的 16 位移位寄存器 SPIxSR (其中 $x = 1$ 或 2) 和一个缓冲器寄存器 SPIxBUF 组成。控制寄存器 SPIxCON 用来配置模块。此外,还有一个状态寄存器 SPIxSTAT 用来表明各种状态条件。

串行接口由 4 个引脚组成: SDIx (串行数据输入)、SDOx (串行数据输出)、SCKx (移位时钟输入或输出) 和 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 位,表明产生溢出条件。数据从 SPIxSR 向 SPIxBUF 的传输不会完成,新数据将丢失。当 SPIROV 为 1 时,模块将不会对 SCKx 电平的跳变进行响应;实际上会禁止模块,直到用户软件读 SPIxBUF 为止。

发送写操作同样也是双缓冲的。用户写入 SPIxBUF。当主器件或从器件传输完成后,移位寄存器 (SPIxSR) 的内容将被传送给接收缓冲器。如果已经向缓冲器寄存器写了任何发送数据,发送缓冲器的内容将被传送给 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。

在主模式下,通过对系统时钟进行预分频来产生时钟。只要数据写入 SPIxBUF 就会被发送。在最后一个位传输的过程中产生中断。

在从模式下, SCKx 上出现外部时钟脉冲时,进行数据的发送和接收。同样,当最后一个位被锁存时产生中断。如果使能了 SSx 控制,那么仅当 SSx 为低电平时使能发送和接收。在 SSx 模式下,当 SSx 为高电平时 SDOx 输出将被禁止。

提供给模块的时钟的频率为 Fosc/4。而后,使用主预分频因子 (PPRE<1:0>) 和辅助预分频因子 (SPRE<2:0>),对这个时钟进行预分频。CKE 位决定发送的时钟边沿,是在时钟从有效状态跳变到空闲状态进行发送,还是反之。CKP 位选择时钟的空闲状态 (高电平或低电平)。

14.1.1 字和字节通信

控制位 MODE16 (SPIxCON<10>) 允许模块在 8 位或 16 位模式下通信。16 位通信,除了发送的位数是 16 而不是 8 以外,与 8 位通信相同。

在更改 MODE16 位之前,用户软件必须禁止模块。用户更改 MODE16 位时, SPI 模块被复位。

8 位和 16 位工作之间的基本区别是: 8 位工作模式下数据是从 SPIxSR 的 bit 7 开始发送的;而在 16 位工作模式下,则是从 SPIxSR 的 bit 15 开始发送。在两种模式下,数据都会移入 SPIxSR 的 bit 0。

14.1.2 SDOx 禁止

SPIxCON 寄存器中的控制位 DISSDO,用来禁止 SDOx 输出。这样可以允许 SPI 模块按仅输入配置进行连接。SDO 也可用作一般 I/O。

14.2 帧 SPI 支持

当工作在主模式或从模式时,模块支持基本的帧 SPI 协议。控制位 FRMEN 使能帧 SPI 支持,并使 SSx 引脚用作帧同步脉冲 (FSYNC) 功能。控制位 SPIFSD 决定 SSx 引脚是输入还是输出(即,模块是接收还是产生帧同步脉冲)。帧脉冲在一个 SPI 时钟周期内是高电平有效脉冲。当帧同步使能时,仅在 SPI 时钟的下一个发送边沿开始发送数据。

dsPIC30F5011/5013

图 14-1: SPI 框图

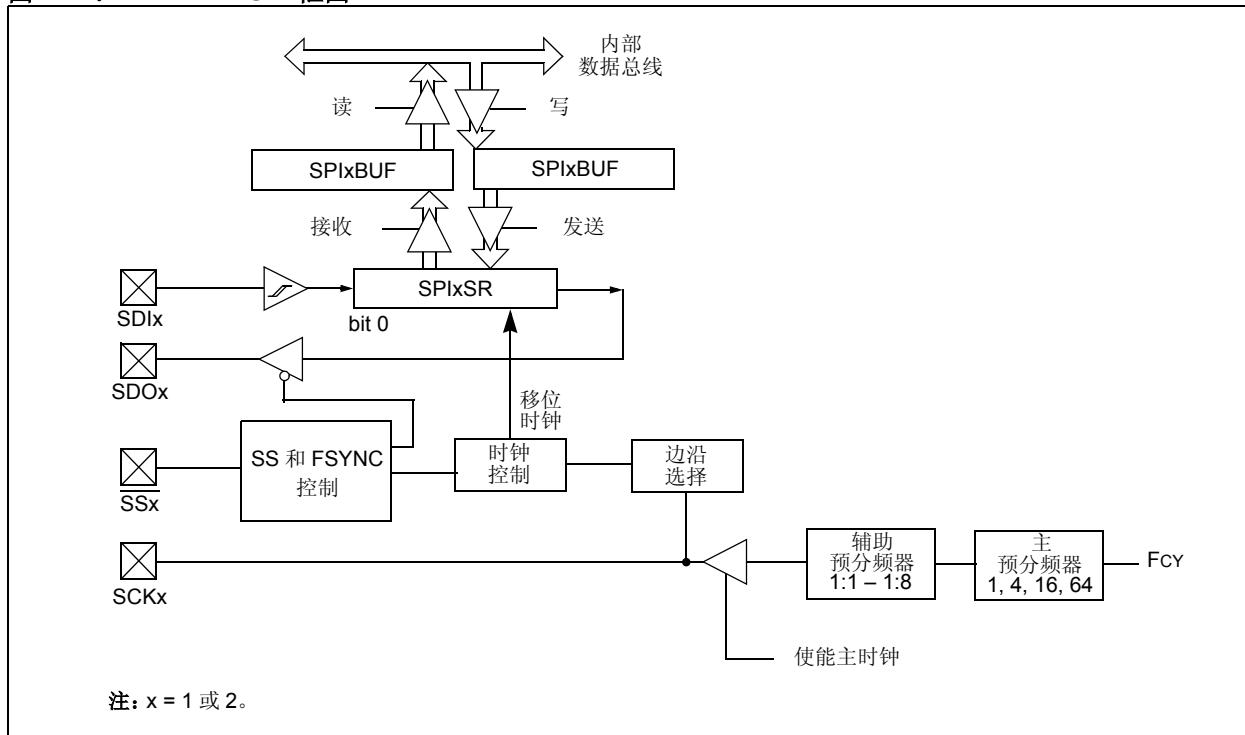
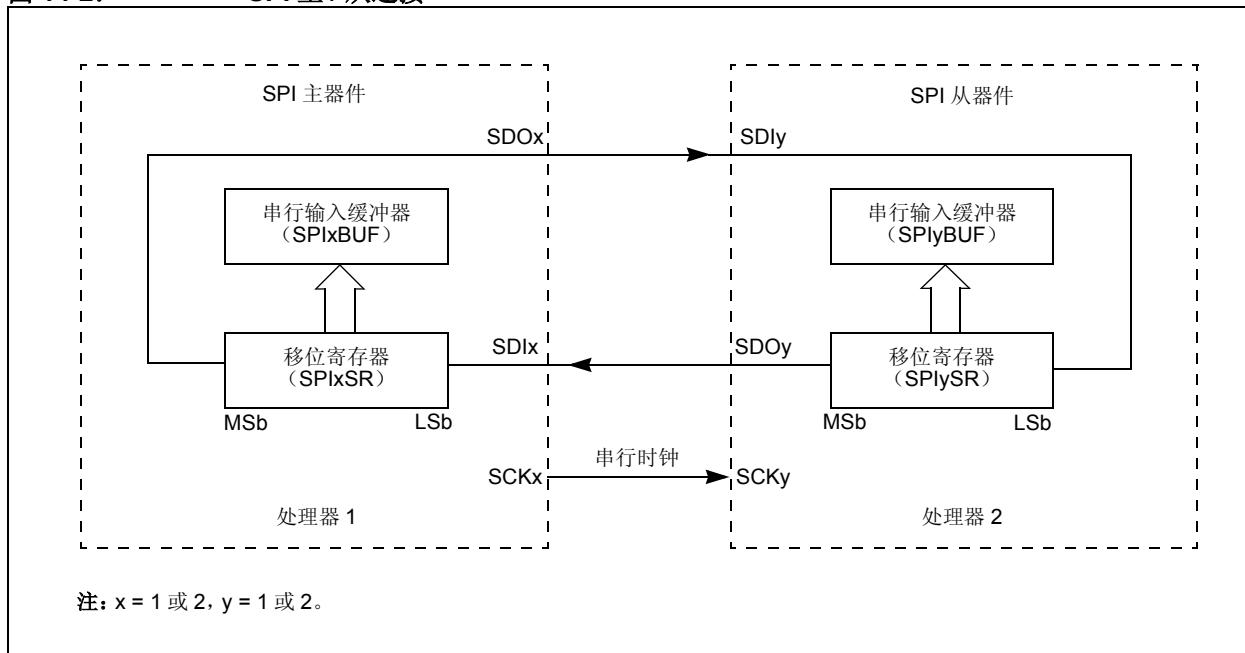


图 14-2: SPI 主 / 从连接



注: $x = 1$ 或 2 , $y = 1$ 或 2 。

14.3 从选择同步

SSx 引脚允许同步从模式。必须使用 SSx 引脚控制使能位 (**SSEN** = 1) 将 SPI 配置为 SPI 从模式。当 SSx 引脚为低电平时，使能发送和接收且驱动 SDOx 引脚。当 SSx 引脚变为高电平时，将不再驱动 SDOx 引脚。另外，可以重新同步 SPI 模块，所有的计数器 / 控制电路都被复位。于是，当 SSx 引脚再次为低电平时，发送 / 接收又将从 MSb 开始，即使 SSx 在发送 / 接收过程中变为高电平。

14.4 CPU 休眠模式期间的 SPI 工作

在休眠模式期间，SPI 模块关闭。如果 CPU 在 SPI 正在进行处理时进入休眠模式，则发送和接收将中止。

在休眠模式下，发送器和接收器将停止。但进入或退出休眠模式不影响寄存器的内容。

14.5 CPU 空闲模式期间的 SPI 工作

当器件进入空闲模式时，所有的时钟源继续工作。**SPISIDL** 位 (**SPIxSTAT<13>**) 选择 SPI 模块在空闲模式下是停止工作还是继续工作。如果 **SPISIDL** = 0，则在 CPU 进入空闲模式时，模块将继续工作。如果 **SPISIDL** = 1，则在 CPU 进入空闲模式时，模块将停止工作。

表 14-1: 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	0220	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000 0000 0000 0000
SPI1CON	0222	—	FRMEN	SPIFSD	—	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000 0000 0000 0000
SPI1BUF	0224	发送和接收缓冲器														0000 0000 0000 0000		

图注: u = 未初始化位

表 14-2: SPI2 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
SPI2STAT	0226	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000 0000 0000 0000
SPI2CON	0228	—	FRMEN	SPIFSD	—	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000 0000 0000 0000
SPI2BUF	022A	发送和接收缓冲器														0000 0000 0000 0000		

图注: u = 未初始化位

注: 有关寄存器各位的说明, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

15.0 I²C 模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但是不应把本手册当作无所不包的参考手册来使用。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多主器件模式提供完全的硬件支持。

此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。

15.1 功能说明

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

从而，在 I²C 总线上，I²C 模块既可作为主器件工作，也可作为从器件工作。

15.1.1 各种 I²C 模式

支持 I²C 模块工作在以下模式：

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

I²C 编程模型，请参见图 15-1。

15.1.2 I²C 模式下的引脚配置

I²C 有一个双引脚接口：SCL 引脚是时钟线，SDA 引脚是数据线。

15.1.3 I²C 寄存器

I2CCON 和 I2CSTAT 分别是控制寄存器和状态寄存器。I2CCON 寄存器是可读写的。I2CSTAT 的低 6 位是只读的，它的其余位是可读写的。

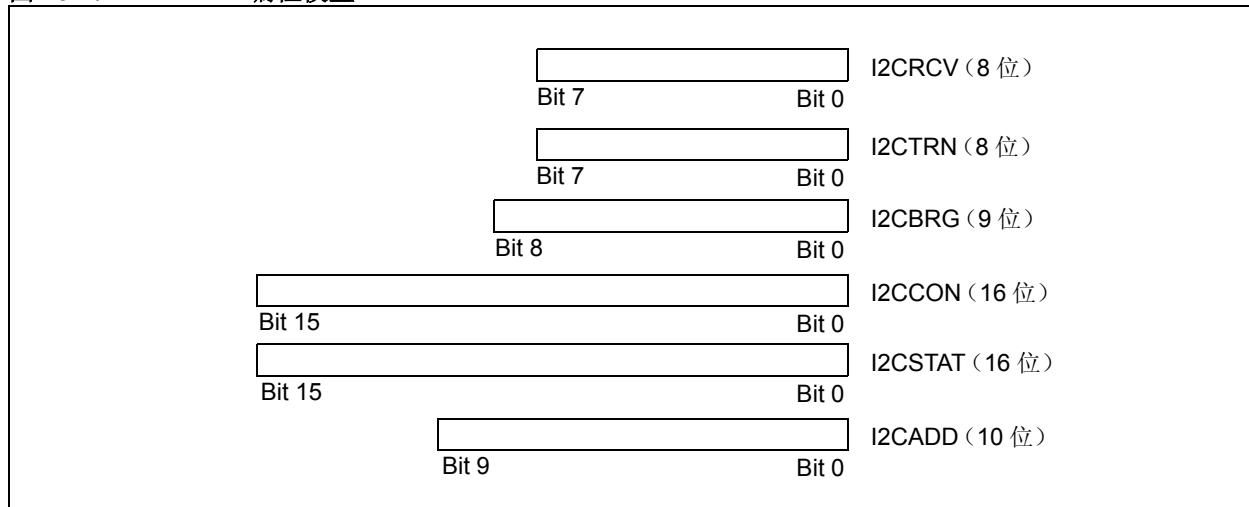
I2CRSR 是移位寄存器，用来对数据进行移位。I2CRCV 是缓冲寄存器，可写入、读出数据字节，如图 15-1 所示。I2CTRN 是发送寄存器，发送操作期间，字节将会写入 I2CTRN，如图 15-2 所示。

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。

在进行接收时，I2CRSR 和 I2CRCV 一起形成双缓冲接收器。当 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。

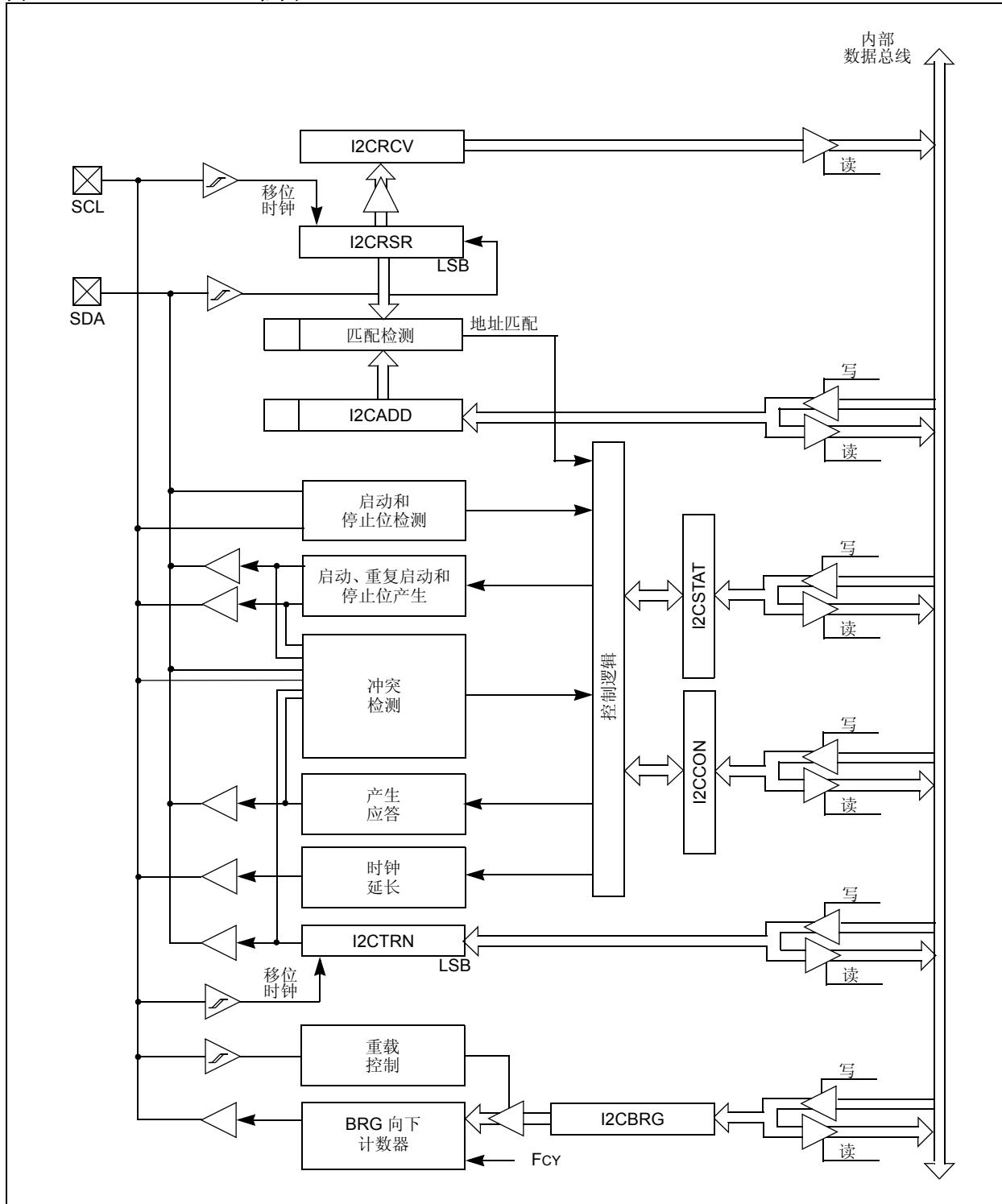
注： 在 10 位模式下，在重复启动条件之后，用户只需匹配前 7 位地址。

图 15-1：编程模型



dsPIC30F5011/5013

图 15-2: I²CTM 框图



15.2 I²C 模块地址

I2CADD 寄存器包含从模式地址。这是一个 10 位寄存器。如果 A10M 位 (I2CCON<10>) 为 0，则模块把地址解释为 7 位地址。当接收到地址时，将收到的地址与 I2CADD 寄存器中的低 7 位进行比较。

如果 A10M 位为 1，则将地址视为 10 位地址。当接收地址时，将收到的地址与二进制值“11110 A9 A8”（其中 A9 和 A8 是 I2CADD 的高 2 位）进行比较。如果值匹配，下一个地址将与 I2CADD 的低 8 位进行比较，这是 10 位寻址协议规定的。

表 15-1: dsPIC30F 支持的 7 位 I²C™ 从地址

0x00	广播呼叫地址或启动字节
0x01-0x03	保留
0x04-0x07	Hs 模式主机码
0x04-0x77	有效 7 位地址
0x78-0x7b	有效 10 位地址（低 7 位）
0x7c-0x7f	保留

15.3 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 位模式下 (A10M = 0)，位 I2CADD<6:0> 与 I2CRSR<7:1> 进行比较，而 I2CRSR<0> 是 R_W 位。所有进入的位都是在 SCL 的上升沿采样的。

如果地址匹配的话，将发送一个应答，并且在第 9 个位 (ACK) 的下降沿将从事件中断标志 (SI2CIF) 置 1。地址匹配不会影响 I2CRCV 缓冲器的内容或 RBF 位。

15.3.1 从模式发送

如果接收到的 R_W 位为 1，串行端口将进入发送模式。模块将在第 9 个位发送 ACK，并使 SCL 保持为 0，直到 CPU 写 I2CTRN 进行响应为止。通过将 SCLREL 位置 1 释放 SCL，而后数据的 8 个位被移出。数据位在 SCL 的下降沿移出，因此 SCL 为高电平时 SDA 有效。将在第 9 个时钟脉冲的下降沿发出中断脉冲，与从主器件接收到的 ACK 状态无关。

15.3.2 从模式接收

如果地址匹配时接收到的 R_W 位为 0，就将启动接收模式。在 SCL 上升沿采样进入的位。接收到 8 个位之后，如果 I2CRCV 未满或 I2COV 没有置 1，I2CRSR 中的数据将被送至 I2CRCV。在第 9 个时钟发送 ACK。

如果 RBF 标志置 1，表明 I2CRCV 仍然存有前一次接收到的数据 (RBF = 1)，则不发送 ACK；但是，仍将产生中断脉冲。如果发生溢出，I2CRSR 中的内容不会被装入 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，用户不会清零 I2COV 位。不发送应答 (ACK = 1)，但 I2CRCV 会被更新。

15.4 I²C 10 位从模式的工作

在 10 位模式下，基本的接收和发送操作与 7 位模式相同。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（默认 10 位地址）进行比较，而 I2CRSR<2:1> 与 I2CADD<9:8> 进行比较。如果发生匹配且 R_W = 0，则发出中断脉冲。ADD10 位被清零，表明地址部分匹配。如果匹配失败或 R_W = 1，则 ADD10 位被清零，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。如果发生地址匹配，则产生中断脉冲并将 ADD10 位置 1，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。

15.4.1 10 位从模式发送

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

15.4.2 10 位从模式接收

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，并将 R_W 位置 1，但不产生停止位，从而开始进行从器件发送操作。

15.5 自动时钟延长

在从模式下，模块可以通过时钟延长同步读缓冲器和写主器件。

15.5.1 发送时钟延长

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

在从发送模式下，始终进行时钟延长，与 STREN 位无关。

在发送序列的第 9 个时钟之后，发生时钟同步。如果器件在第 9 个时钟的下降沿采样到 ACK，并且 TBF 位仍然清零的话，则 SCLREL 位自动清零。SCLREL 的清零将使 SCL 线置为低电平。在允许继续发送之前，用户中断服务程序必须将 SCLREL 位置 1。通过使 SCL 线保持为低电平，在主器件启动另一个发送序列之前，用户就有时间执行中断服务程序并装入 I2CTRN 的内容。

- 注 1:** 如果用户装入 I2CTRN 的内容，在第 9 个时钟下降沿之前将 TBF 位置 1，则 SCLREL 位将不会被清零，也不会发生时钟延长。
- 2:** 不管 TBF 位的状态如何，SCLREL 位都可以用软件置 1。

15.5.2 接收时钟延长

在从接收模式下，I2CCON 寄存器中的 STREN 位可用来使能时钟延长。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。

15.5.3 7 位寻址时的时钟延长（STREN = 1）

在从接收模式下，当 STREN 位置 1，SCL 线在缓冲寄存器已满时保持为低电平。对于 7 位和 10 位寻址模式，延长 SCL 输出的方法相同。

时钟延长将在接收序列的第 9 个时钟后进行。在 ACK 序列末尾的第 9 个时钟的下降沿，如果 RBF 位置 1，SCLREL 位将自动清零，这将强制 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将 SCLREL 位置 1。通过保持 SCL 线为低电平，用户可以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。这可以防止缓冲器溢出。

- 注 1:** 如果用户读 I2CRCV 的内容，在第 9 个时钟下降沿之前将 RBF 位清零，则 SCLREL 位将不会被清零，也不会发生时钟延长。
- 2:** 不管 RBF 位的状态如何，SCLREL 位都可以用软件置 1。为避免溢出，在下一个接收序列开始之前，用户要注意在中断服务程序中清零 RBF 位。

15.5.4 10 位寻址时的时钟延长（STREN = 1）

在寻址过程中会自动进行时钟延长。由于模块有一个寄存器用于存放完整的地址，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。

15.6 软件控制的时钟延长（STREN = 1）

当 STREN 位为 1 时，可以用软件清零 SCLREL 位，从而实现对时钟延长的软件控制。逻辑将对 SCLREL 位的写操作与 SCL 时钟进行同步。清零 SCLREL 位不会使 SCL 输出保持为低电平，直到模块检测到 SCL 输出的下降沿且采样到 SCL 为低电平。在 SCL 线采样为低电平时，如果用户清零 SCLREL 位，则 SCL 输出将保持为低电平。SCL 输出将保持为低电平，直到 SCLREL 位置 1，并且 I²C 总线上所有其他器件都已释放了 SCL。这确保了对 SCLREL 位的写操作不会违反 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。

15.7 中断

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

15.8 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。

15.9 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, 模块将接受并对所有地址执行操作。

15.10 广播呼叫地址支持

广播呼叫地址能寻址所有器件。当使用此地址时, 理论上所有器件都应该以应答信号作出响应。

广播呼叫地址是 I²C 协议为特定目的保留的 8 个地址之一。它由全 0 地址位组成, 且 R_W = 0。

当广播呼叫使能位 (GCEN) 置 1 (I2CCON<7> = 1) 时, 将识别广播呼叫地址。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, I2CRSR 在第 8 个时钟之后被送至 I2CRCV, RBF 标志在第 9 位 (ACK 位) 的下降沿置 1, 主事件中断标志 (MI2CIF) 置 1。

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

15.11 I²C 主模式支持

用作主器件时, 支持以下 6 种操作:

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

15.12 I²C 主模式的工作

主器件产生所有的串行时钟脉冲、启动和停止条件。出现停止条件或重复启动条件时, 传输结束。因为重复启动条件也是下一次串行传输的开始, 此时不会释放 I²C 总线。

在主发送模式下, 串行数据通过 SDA 输出, 而 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。一次发送 8 位串行数据。在每个字节发送完之后, 接收到一个 ACK 位。输出启动和停止条件, 以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, 发送的第一个字节是一个 7 位从地址, 后跟一个表明接收的 1。通过 SDA 接收串行数据, 而 SCL 输出串行时钟。一次接收 8 位串行数据。在每个字节接收之后, 发送一个 ACK 位。启动和停止条件分别表明发送的开始和结束。

15.12.1 I²C 主发送

只需将值写入 I2CTRN 寄存器, 便可以进行数据字节、7 位地址或 10 位地址后半部分的发送。当模块处于等待状态时, 用户应该只写 I2CTRN。此操作将使缓冲器满标志 (TBF) 置 1, 允许波特率发生器开始计数, 并开始下一次发送。在 SCL 的下降沿后, 地址 / 数据的每一个位都将被移出至 SDA 引脚上。发送状态标志 TRSTAT (I2CSTAT<14>) 用来表明主器件是否正在进行发送。

15.12.2 I²C 主接收

通过设置接收使能位 RCEN (I2CCON<3>) 使能主控模式接收。在 RCEN 位置 1 之前, I²C 模块必须处于空闲状态, 否则将忽略 RCEN 位。波特率发生器开始计数, 每次计满返回时, 在每个时钟的上升沿 SCL 引脚 ACK 的状态和数据被移入 I2CRSR 中。

15.12.3 波特率发生器

在 I²C 主模式下，BRG 的重载值位于 I2CBRG 寄存器中。BRG 装入该值后，BRG 将递减计数到 0 并停止，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 BRG。

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

公式 15-1：串行时钟速率

$$I2CBRG = \left(\frac{FCY}{FSCK} - \frac{FCY}{1,111,111} \right) - 1$$

15.12.4 时钟仲裁

在任何接收、发送或重复启动 / 停止条件期间，当主器件释放了 SCL 引脚（允许 SCL 悬空为高电平）时，就会发生时钟仲裁。当允许 SCL 引脚悬空为高电平时，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入 I2CBRG 的内容并开始计数。当外部器件使时钟保持为低电平时，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。

15.12.5 多主器件通信、总线冲突和总线仲裁

通过总线仲裁实现了对多主器件工作的支持。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一个 1，而另一个主器件要输出 0，就会发生仲裁。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 MI2CIF 脉冲置 1，并且将 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，清零 TBF 标志，释放 SDA 和 SCL 线，而后可将值写入 I2CTRН。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。

当总线冲突发生在启动、重复启动、停止或应答条件期间，将中止这些条件，释放 SDA 和 SCL 线，清零 I2CCON 寄存器中对应的控制位。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，如果停止条件出现，将 MI2CIF 位置 1。

写 I2CTRН 将从第一个数据位开始数据的发送，无论发生总线冲突时发送器停在何处。

在多主器件环境中，在检测到启动和停止条件时产生中断，可以确定总线何时空闲。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。

15.13 CPU 休眠和空闲模式期间 I²C 模块的工作

15.13.1 CPU 休眠模式期间 I²C 的工作

当器件进入休眠模式时，模块的所有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，则发送将中止。类似地，如果休眠出现在接收过程中，则接收也将中止。

15.13.2 CPU 空闲模式期间 I²C 的工作

对于 I²C，I2CSIDL 位用来选择空闲时模块是停止还是继续工作。如果 I2CSIDL = 0，则在空闲模式下模块将继续工作。如果 I2CSIDL = 1，则在空闲模式下模块将停止工作。

表 15-2: I²CTM 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		

图注: — = 未初始化位

注: 有关寄存器各位的说明, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

dsPIC30F5011/5013

注：

16.0 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，但是不应把本手册当作无所不包的参考手册来使用。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)。

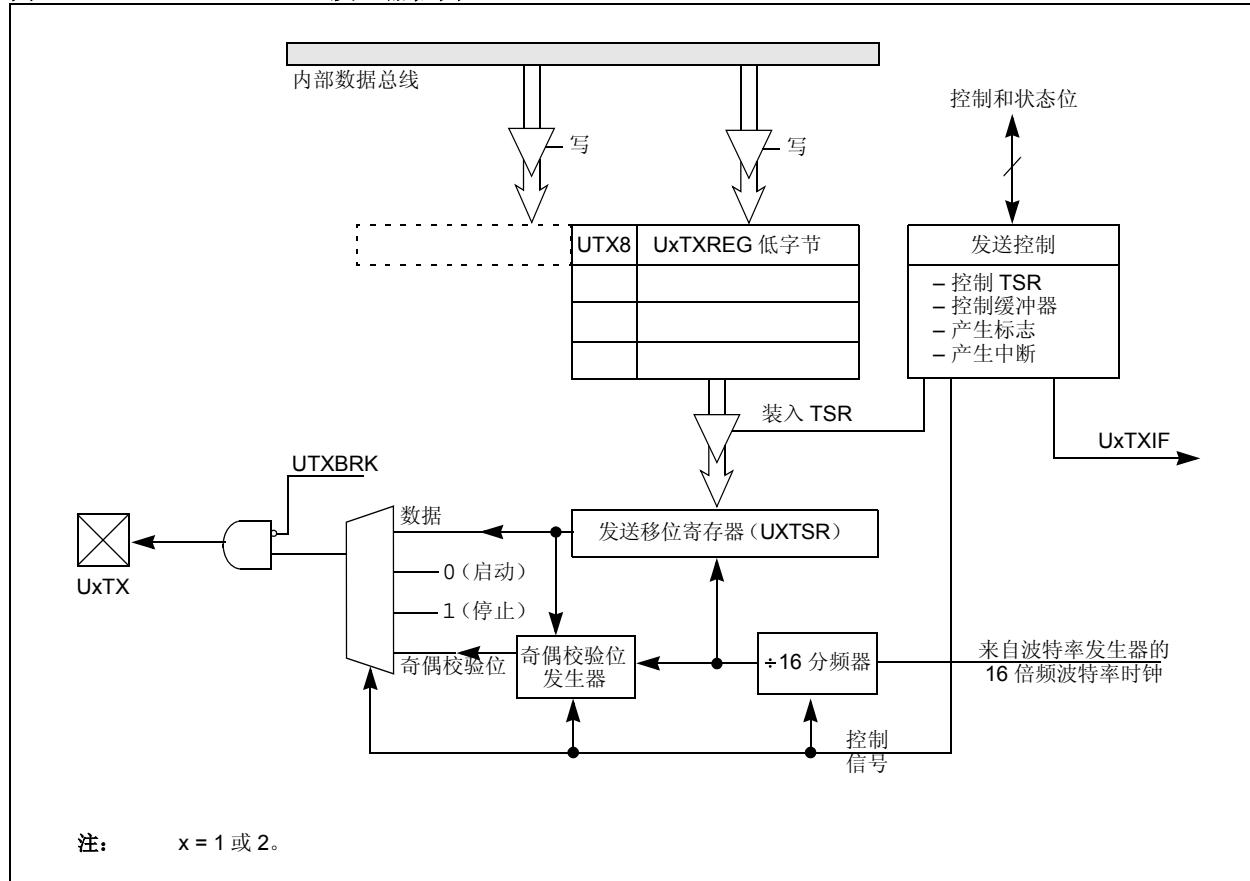
本章介绍了通用异步接收器 / 发送器通信模块。

16.1 UART 模块概述

UART 模块的主要特性是：

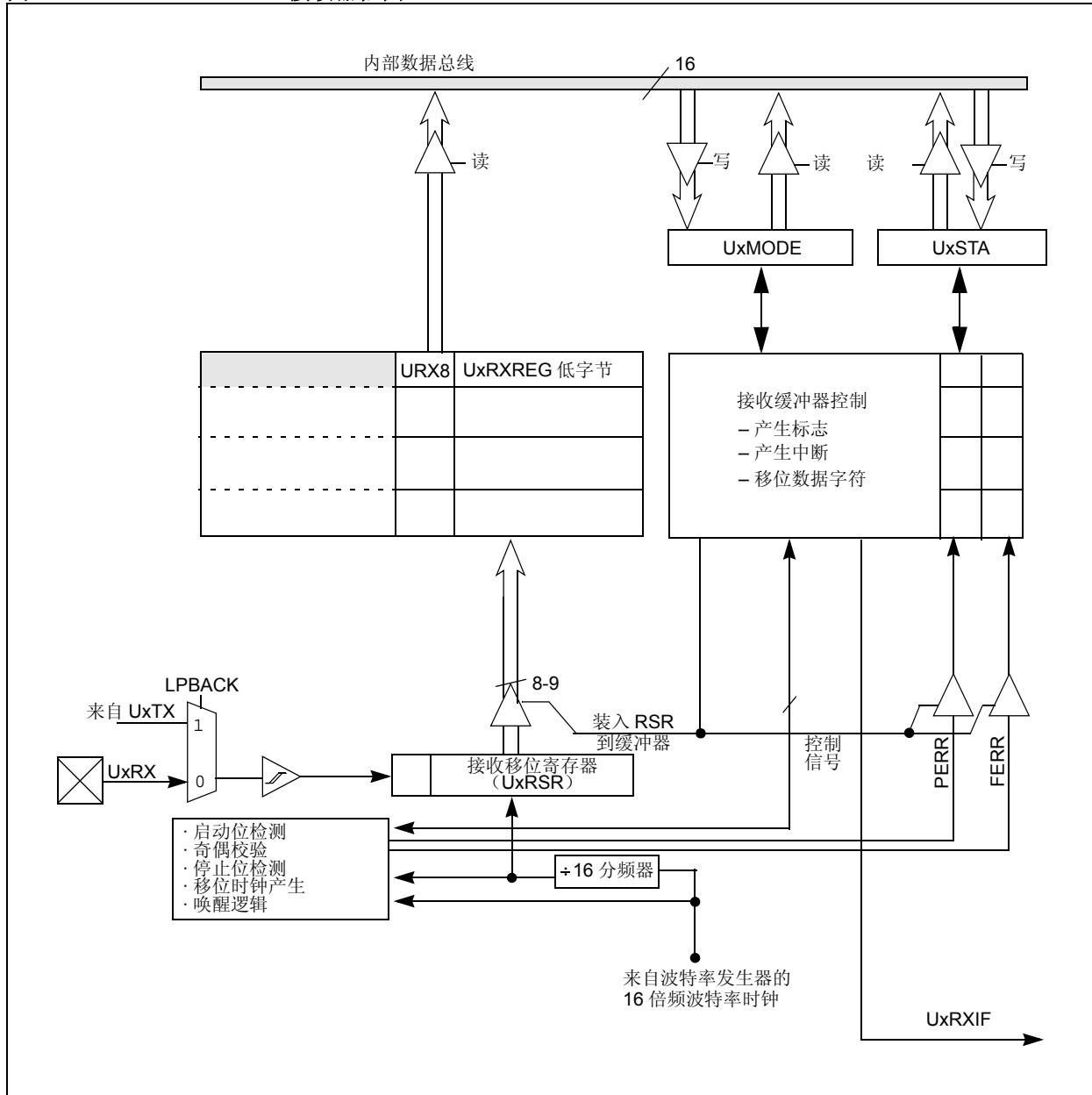
- 全双工 8 位或 9 位数据通信
- 偶校验、奇校验或无奇偶校验选项（对于 8 位数据）
- 一个或两个停止位
- 完全集成的波特率发生器，具有 16 位预分频器
- 在 30 MHz 指令速率时，波特率范围为 38 bps 到 1.875 Mbps
- 4 字深度发送数据缓冲器
- 4 字深度接收数据缓冲器
- 奇偶校验错误、帧错误和缓冲器溢出错误检测
- 支持仅在地址检测（第 9 位 = 1）时产生中断
- 独立的发送和接收中断
- 用于诊断支持的环回模式

图 16-1：UART 发送器框图



dsPIC30F5011/5013

图 16-2: UART 接收器框图



16.2 使能和设置 UART

16.2.1 使能 UART

通过将 **UxMODE** 寄存器（其中 $x = 1$ 或 2 ）中的 **UARTEN** 位置 1，可以使能 **UART** 模块。一旦使能，**UxTX** 和 **UxRX** 引脚被分别配置为输出和输入，这改写了相应 **I/O** 端口引脚的 **TRIS** 和 **LATCH** 寄存器位的设置。在没有传输发生时，**UxTX** 引脚状态为逻辑 1。

16.2.2 禁止 UART

通过清零 **UxMODE** 寄存器中的 **UARTEN** 位，可以禁止 **UART** 模块。这是任何复位后的默认状态。如果 **UART** 被禁止，那么所有的 **I/O** 引脚均用作端口引脚，受相应端口引脚的锁存器和 **TRIS** 位的控制。

禁止 **UART** 模块会将缓冲器复位为空状态。缓冲器中的所有数字字符都将丢失，同时波特率计数器也将复位。

当 **UART** 模块被禁止时，所有与之相关的错误和状态标志都将复位。**URXDA**、**OERR**、**FERR**、**PERR**、**UTXEN**、**UTXBRK** 和 **UTXBF** 位被清零，而 **RIDLE** 和 **TRMT** 被置 1。其他控制位，包括 **ADDEN**、**URXISEL<1:0>** 和 **UTXISEL**，以及 **UxMODE** 和 **UxBRG** 寄存器则不受影响。

当 **UART** 处于工作状态时，对 **UARTEN** 位清零将中止所有等待的发送和接收，同时还将象上面定义的那样复位模块。再次使能 **UART** 将使用同样的配置重新启动 **UART**。

16.2.3 设置数据、奇偶校验和停止位选择

UxMODE 寄存器中的控制位 **PDSEL<1:0>** 用来选择发送时使用的数据长度和奇偶校验。数据长度可以是 8 位，具有偶校验位或奇校验位或没有奇偶校验位，数据长度也可以是没有奇偶校验位的 9 位。

STSEL 位决定在数据发送期间，是使用一个还是两个停止位。

UART 的默认（上电）设置是 8 位、无奇偶校验位以及 1 个停止位（通常表示为 8, N, 1）。

16.3 发送数据

16.3.1 8 位数据模式下的发送

为了发送 8 位数据，必须执行以下步骤：

1. 设置 **UART**:

首先，必须选择数据长度、奇偶校验和停止位的位数。然后，在 **UxMODE** 和 **UxSTA** 寄存器中设置发送和接收中断允许位以及优先级位。另外，必须向 **UxBRG** 寄存器中写入适当的波特率值。

2. 通过将 **UARTEN** (**UxMODE<15>**) 位置 1，使能 **UART**。

3. 将 **UTXEN** 位 (**UxSTA<10>**) 置 1，从而使能发送。

4. 将要发送的字节写入 **UxTXREG** 的低字节。此值将被立即传输到发送移位寄存器 (**UxTSR**)，在波特率时钟的下一个上升沿期间，串行位流将开始移出移位寄存器。或者，可以在 **UTXEN** = 0 时写入数据字节，而后用户可以将 **UTXEN** 置 1。这将使串行位流立即开始，因为波特率时钟将从清零状态启动。

5. 将产生发送中断，这取决于中断控制位 **UTXISEL** (**UxSTA<15>**) 的值。

16.3.2 9 位数据模式下的发送

9 位数据发送涉及的步骤与 8 位数据发送类似，但是必须将一个 16 位数据字（其中高 7 位始终为零）写入 **UxTXREG** 寄存器。

16.3.3 发送缓冲器 (**UxTXB**)

发送缓冲器为 9 位宽，4 字符深度。加上发送移位寄存器 (**UxTSR**)，用户实际上有一个 5 级深度的 FIFO（先进先出）缓冲器。**UTXBF** 状态位 (**UxSTA<9>**) 表明发送缓冲器是否已满。

如果用户试图写已满的缓冲器，则 FIFO 不会接收新数据，缓冲器内也不会发生数据移位。这确保了能够从缓冲器溢出条件中恢复。

FIFO 在任何器件复位时复位，但当器件进入省电模式或从省电模式唤醒时，FIFO 不受影响。

16.3.4 发送中断

发送中断标志（U1TXIF 或 U2TXIF）位于相应的中断标志寄存器中。

发送器通过产生一个边沿将 UxTXIF 位置 1。产生中断的条件取决于 UTXISEL 控制位：

- a) 如果 UTXISEL = 0，当从发送缓冲器向发送移位寄存器（UxTSR）传输一个字时，将产生中断。这意味着发送缓冲器中至少有一个空字。
- b) 如果 UTXISEL = 1，当从发送缓冲器向发送移位寄存器（UxTSR）传输一个字并且发送缓冲器为空时，将产生中断。

工作期间可以在这两种中断模式之间切换，有时这提供了更大的灵活性。

16.3.5 发送间隔字符

将 UTXBRK 位（UxSTA<11>）置 1 将使 UxTX 线被驱动为逻辑 0。UTXBRK 位改写了所有的发送活动。从而，通常用户应该在将 UTXBRK 置 1 之前等待发送器空闲。

要发送间隔字符，必须用软件将 UTXBRK 置 1，并且保持该位置 1 至少 13 个波特率时钟周期。然后用软件将 UTXBRK 位清零，产生停止位。在重新装入 UxTXB 或开始其他发送活动之前，用户必须等待至少一或两个波特率时钟周期，以确保产生有效的停止位。发送间隔字符不会产生发送中断。

16.4 接收数据

16.4.1 8 位或 9 位数据模式下的接收

在接收 8 位或 9 位数据时，必须执行以下步骤：

1. 设置 UART（见第 16.3.1 节“8 位数据模式下的发送”）。
2. 使能 UART（见第 16.3.1 节“8 位数据模式下的发送”）。
3. 当接收到一个或多个数据字时，将产生接收中断，这取决于由 URXISEL 位（UxSTA<7:6>）指定的接收中断设置。
4. 读 OERR 位，确定是否发生了溢出错误。OERR 位必须用软件复位。
5. 从 UxRXREG 读取接收到的数据。读 UxRXREG 会将下一个字移动到接收 FIFO 的顶端，并且将更新 PERR 和 FERR 值。

16.4.2 接收缓冲器（UxRXB）

接收缓冲器为 4 字深度。加上接收移位寄存器（UxRSR），用户实际上有一个 5 字深度的 FIFO 缓冲器。

URXDA（UxSTA<0>）= 1 表明接收缓冲器中还有数据。URXDA = 0 意味着缓冲器为空。如果用户试图读空缓冲器，读取的值是缓冲器中的原有值，在 FIFO 中不会发生数据移位。

任何器件复位时，FIFO 都将复位。当器件进入省电模式或从省电模式唤醒时，FIFO 不受影响。

16.4.3 接收中断

可以从相应的中断标志寄存器中读取接收中断标志（U1RXIF 或 U2RXIF）。中断标志由接收器产生的边沿置 1。将接收中断标志置 1 的条件，取决于由 URXISEL<1:0>（UxSTA<7:6>）控制位指定的设置。

- a) 如果 URXISEL<1:0> = 00 或 01，每当从接收移位寄存器（UxRSR）向接收缓冲器传输一个字时，就会产生中断。接收缓冲器可以有一个或多个字符。
- b) 如果 URXISEL<1:0> = 10，当从接收移位寄存器（UxRSR）向接收缓冲器传输一个字，使得接收缓冲器中有 3 个字符时，就会产生中断。
- c) 如果 URXISEL<1:0> = 11，当从接收移位寄存器（UxRSR）向接收缓冲器传输一个字，使得接收缓冲器中有 4 个字符（即缓冲器满）时，就会产生中断。

工作期间可以在这两种中断模式之间切换，尽管正常工作期间这样做并不可取。

16.5 接收错误处理

16.5.1 接收缓冲器溢出错误（OERR 位）

如果满足以下所有条件，OERR 位（UxSTA<1>）将置 1：

- a) 接收缓冲器满。
- b) 接收移位寄存器满，但不能把字符送到接收缓冲器。
- c) 检测到 UxRSR 中字符的停止位，这表明 UxRSR 需要把字符送到缓冲器。

一旦 OERR 置 1，就不会有数据移入 UxRSR（除非用软件清零 OERR 位或发生复位）。存放在 UxRSR 和 UxRXREG 中的数据继续有效。

16.5.2 帧错误 (FERR)

如果检测到 0 而不是停止位，则 FERR 位 ($\text{UxSTA}\langle 2 \rangle$) 置 1。如果选用两个停止位，则两个停止位都必须为 1，否则 FERR 也将置 1。FERR 位是只读的，它将和接收到的数据一起被送入缓冲器。此位在发生任何复位时被清零。

16.5.3 奇偶校验错误 (PERR)

如果接收到的字奇偶校验错误的话，PERR 位 ($\text{UxSTA}\langle 3 \rangle$) 将置 1。此错误位仅在选取了奇偶校验模式（奇校验或偶校验）时适用。PERR 位是只读的，它将和接收到的数据一起被送入缓冲器。此位在发生任何复位时被清零。

16.5.4 空闲状态

当接收器处于活动状态（即，在最初检测到启动位与停止位完成之间）时，RIDLE 位 ($\text{UxSTA}\langle 4 \rangle$) 为 0。在停止位完成与检测到下一个启动位之间时，RIDLE 位为 1，表明 UART 空闲。

16.5.5 接收间隔字符

接收器会根据在 PDSEL ($\text{UxMODE}\langle 2:1 \rangle$) 和 STSEL ($\text{UxMODE}\langle 0 \rangle$) 位中设置的值，计数并等待一定的位时间数。

如果间隔超过 13 个位时间，则在 PDSEL 和 STSEL 指定的位时间数之后，认为接收完成。URXDA 位置 1，FERR 置 1，接收 FIFO 中装入零，同时产生中断（如果允许中断的话），并且 RIDLE 位置 1。

当模块接收到长间隔信号，同时接收器检测到启动位、数据位和无效的停止位（这将把 FERR 置 1），那么接收器在找寻下一个启动位之前，必须等待有效停止位出现。接收器不能将线上的间隔条件看作是下一个启动位。

间隔字符被视为一个全 0 的字符，且 FERR 位置 1。间隔字符被装入缓冲器中。只有在接收到停止位之后，才会继续进行接收。注意，当还没有接收到停止位时，RIDLE 将变为高电平。

16.6 地址检测模式

通过将 ADDEN 位 ($\text{UxSTA}\langle 5 \rangle$) 置 1 可以使能此特殊模式，在此模式下，如果第 9 位 (URX8) 值为 1，则表明接收到的字是一个地址而不是数据。此模式仅适用于 9 位数据通信。在此模式下，URXISEL 控制位对中断产生没有任何影响，因为中断（如果允许的话）仅在接收到的字的第 9 位置 1 时产生。

16.7 环回模式

通过将 LPBACK 位置 1 可以使能此特殊模式，在该模式下， UxTX 引脚在内部连接到 UxRX 引脚。当配置为环回模式时， UxRX 引脚与内部 UART 接收逻辑断开。但是， UxTX 引脚仍然正常工作。

要选择该模式：

- 将 UART 配置为所需的工作模式。
- 设置 LPBACK = 1，使能环回模式。
- 使能发送，如同第 16.3 节“发送数据”中定义的那样。

16.8 波特率发生器

UART 有一个 16 位波特率发生器，以便在波特率发生时获得最大的灵活性。波特率发生器寄存器 (UxBRG) 是可读写的。波特率的计算如下：

$$\text{BRG} = \text{UxBRG} \text{ 寄存器中的 16 位值 (0 到 65535)}$$

$$\text{FCY} = \text{指令时钟速度 (1/TCY)}$$

波特率由公式 16-1 给出。

公式 16-1: 波特率

$$\text{波特率} = \text{FCY} / (16 * (\text{BRG} + 1))$$

从而，可能的最大波特率是：

$$\text{FCY} / 16 \text{ (如果 BRG = 0),}$$

可能的最小波特率是：

$$\text{FCY} / (16 * 65536)。$$

对于在 30 MIPS 下工作的 16 位波特率发生器，能够达到的最小波特率是 28.5 bps。

16.9 自动波特率支持

为了让系统确定接收字符的波特率，可选择将输入连接到所选择的捕捉输入（UART1 连接到 IC1，UART2 连接到 IC2）。要使能该模式，用户必须通过对输入捕捉模块进行设置，以检测启动位的上升沿和下降沿。

16.10 UART 在 CPU 休眠和空闲模式下的工作

16.10.1 UART 在 CPU 休眠模式下的工作

当器件进入休眠模式时，模块的所有时钟源都将关闭并保持在逻辑 0。如果在发送进行期间进入休眠模式的话，则发送将中止。**UxTX** 引脚被驱动为逻辑 1。类似地，如果在接收进行期间进入休眠模式，则接收将中止。休眠模式不影响 **UxSTA**、**UxMODE**、发送和接收寄存器和缓冲器以及 **UxBRG** 寄存器。

在器件进入休眠模式前，如果 **WAKE** 位 (**UxMODE<7>**) 置 1，则 **UxRX** 引脚上的下降沿将产生接收中断。接收中断选择模式位 (**URXISEL**) 对该功能没有影响。如果允许接收中断，那么这会将器件从休眠中唤醒。为了产生唤醒中断，必须将 **UARTEN** 位置 1。

16.10.2 UART 在 CPU 空闲模式下的工作

对于 **UART**，**USIDL** 位选择当器件进入空闲模式时，模块是停止工作还是继续工作。如果 **USIDL** = 0，则模块将在空闲模式下继续工作。如果 **USIDL** = 1，则模块将在空闲模式下停止工作。

表 16-1: UART1 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
U1MODE	020C	UARTEN	—	USIDL	—	—	—	—	—	WAKE	LPBACK	ABAUD	—	—	PDSEL1	PDSEL0	STSEL	0000 0000 0000 0000
U1STA	020E	UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0000 0001 0001 0000
U1TXREG	0210	—	—	—	—	—	—	—	UTX8								发送寄存器	0000 000u uuuu uuuu
U1RXREG	0212	—	—	—	—	—	—	—	URX8								接收寄存器	0000 0000 0000 0000
U1BRG	0214																波特率发生器预分频器	0000 0000 0000 0000

图注: u = 未初始化位

表 16-2: UART2 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
U2MODE	0216	UARTEN	—	USIDL	—	—	—	—	—	WAKE	LPBACK	ABAUD	—	—	PDSEL1	PDSEL0	STSEL	0000 0000 0000 0000
U2STA	0218	UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0000 0001 0001 0000
U2TXREG	021A	—	—	—	—	—	—	—	UTX8								发送寄存器	0000 000u uuuu uuuu
U2RXREG	021C	—	—	—	—	—	—	—	URX8								接收寄存器	0000 0000 0000 0000
U2BRG	021E																波特率发生器预分频器	0000 0000 0000 0000

图注: u = 未初始化位

注: 有关寄存器各位的说明, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

dsPIC30F5011/5013

注：

17.0 CAN 模块

注: 本数据手册总结了 dsPIC30F 系列器件的功能,但是不应把本手册当作无所不包的参考手册来使用。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)。

17.1 概述

控制器局域网 (Controller Area Network, CAN) 模块是一个串行接口,用来与其他 CAN 模块或单片机进行通信。此接口/协议设计为允许在噪声环境下进行通信。

CAN 模块是一个通信控制器,实现了 BOSCH 规范中定义的 CAN 2.0 A/B 协议。模块支持协议的 CAN 1.2、CAN 2.0A、CAN 2.0B Passive 以及 CAN2.0B Active 版本。模块实现的是完整的 CAN 系统。本数据手册不讨论 CAN 规范。更多详细信息,读者可参见 BOSCH CAN 规范。

模块具有以下特性:

- 实现了 CAN 协议 CAN 1.2、CAN 2.0A 和 CAN 2.0B
- 标准和扩展数据帧
- 0-8 字节数据长度
- 可编程比特率,最高为 1 Mbps
- 支持远程帧
- 双缓冲接收器,带两个区分优先级的接收报文存储缓冲器(每个缓冲器可存放最多 8 字节的数据)
- 6 个完全(标准/扩展标识符)接收过滤器,其中 2 个与高优先级接收缓冲器相关联,4 个与低优先级接收缓冲器相关联
- 2 个完全接收过滤屏蔽寄存器,分别与高优先级接收缓冲器和低优先级接收缓冲器相关联
- 3 个发送缓冲器,能够由应用指定优先级,并具有中止能力(每个缓冲器可存放最多 8 字节的数据)
- 可编程唤醒功能,集成了低通滤波器
- 可编程环回模式,支持自检操作
- 通过中断功能,为所有 CAN 接收器和发送器的错误状态发送信号
- 可编程时钟源
- 到输入捕捉模块(对于 CAN1 和 CAN2 为 IC2)的可编程链接,可进行时间标记和网络同步
- 低功耗休眠和空闲模式

CAN 总线模块由协议引擎与报文缓冲 / 控制组成。CAN 协议引擎处理 CAN 总线上报文接收和发送的所有功能。发送报文时,首先将报文装入恰当的数据寄存器。通过读取恰当的寄存器,可以检查状态和错误。在 CAN 总线检测到任何报文时,首先检查是否有错,然后与过滤器进行匹配,决定是否接收报文并把它存放到两个接收寄存器之一。

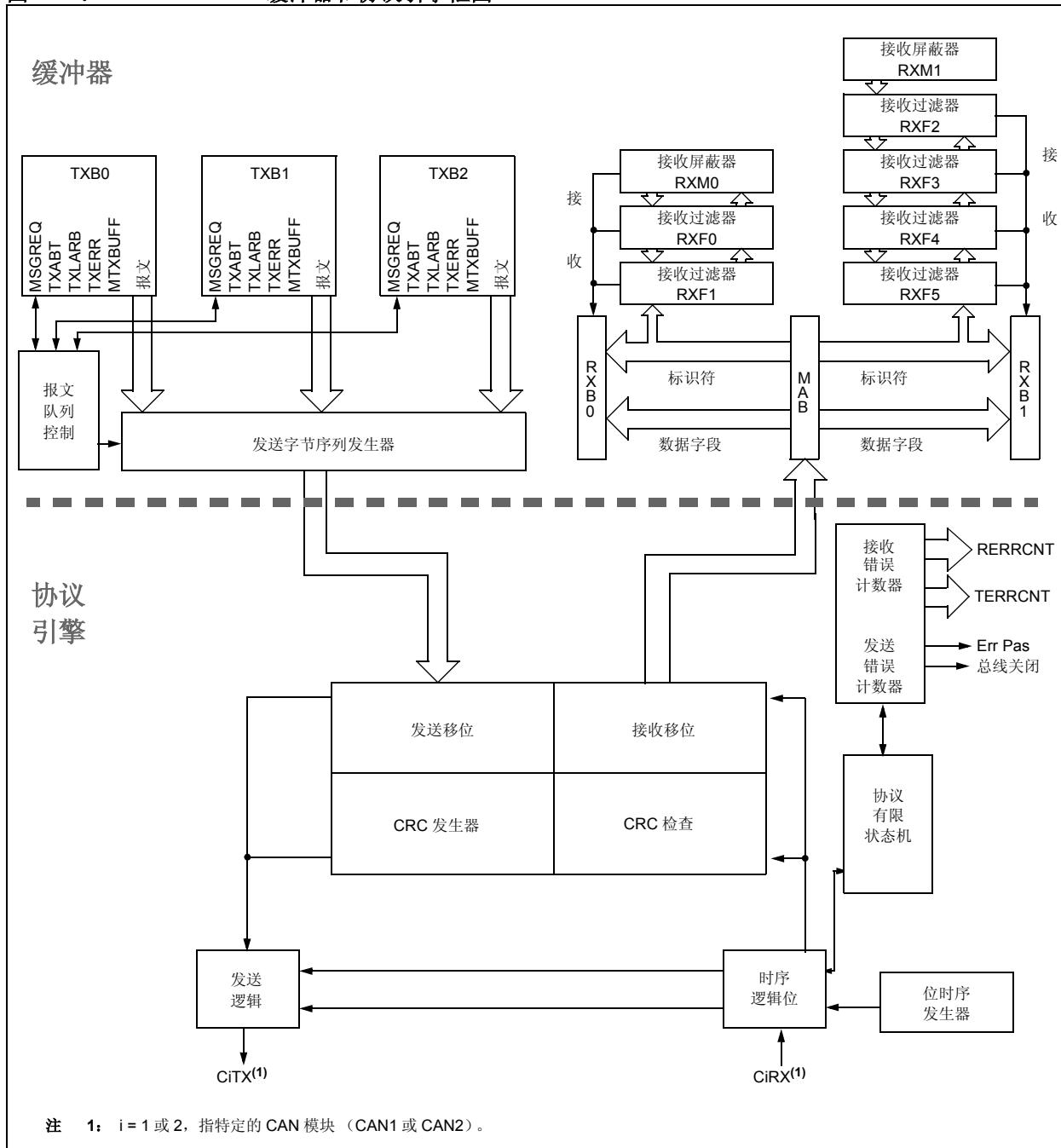
17.2 帧类型

CAN 模块发送各种类型的帧,包括用户发起的数据报文或远程发送请求,以及用于控制目的而自动生成的帧。支持以下帧类型:

- **标准数据帧:**
当节点希望发送数据时,就将生成标准数据帧。它包括一个 11 位标准标识符 (SID),但不包括 18 位扩展标识符 (EID)。
- **扩展数据帧:**
扩展数据帧与标准数据帧类似,但多了扩展标识符。
- **远程帧:**
目标节点向源节点请求发送数据是可能的。要做到这一点,目标节点需要发送一个远程帧,其标识符与所需数据帧的标识符相匹配。随后,相应的数据源节点会发送一个数据帧,作为对远程请求的响应。
- **错误帧:**
节点检测到总线错误,就会生成错误帧。一个错误帧由 2 个字段构成: 错误标志字段和错误界定符字段。
- **过载帧:**
如果出现两种情况之一,节点将生成过载帧。第一种情形,节点检测到帧间间隔期间出现显性位,而这是非法状态。第二种情形,由于内部状态,节点还不能开始接收下一条报文。节点可生成最多 2 个连续的过载帧,以便延迟接收下一条报文。
- **帧间间隔:**
帧间间隔将当前帧(无论何种类型)与其后的数据帧或远程帧分隔开来。

dsPIC30F5011/5013

图 17-1: CAN 缓冲器和协议引擎框图



注 1: i = 1 或 2, 指特定的 CAN 模块 (CAN1 或 CAN2)。

17.3 工作模式

用户可以选择 CAN 模块工作在以下几种工作模式之一。这些模式包括：

- 初始化模式
- 禁止模式
- 正常工作模式
- 监听模式
- 环回模式
- 错误识别模式

通过设置 $\text{REQOP}_{<2:0>} = 000$ 位 ($\text{CiCTRL}_{<10:8>} = 000$) 可选择所需模式。通过监视 $\text{OPMODE}_{<2:0>} = 001$ 位 ($\text{CiCTRL}_{<7:5>} = 001$) 可以确认进入的模式。除非模式改变是可接受的，通常是在由至少 11 个连续隐性位确定的总线空闲时间内，否则模块不会改变模式和 OPMODE 位。

17.3.1 初始化模式

在初始化模式下，模块不进行发送或接收。错误计数器清零，中断标志保持不变。编程器能够访问配置寄存器，而这在其他模式下是受到限制的。模块会防止用户因为编程错误而意外违反 CAN 协议。当模块在线时，不能修改所有控制模块配置的寄存器。如果正在发送，不允许 CAN 模块进入配置模式。配置模式就像是一把锁，保护着下列寄存器。

- 所有模块控制寄存器
- 波特率和中断配置寄存器
- 总线时序寄存器
- 标识符接收过滤寄存器
- 标识符接收屏蔽寄存器

17.3.2 禁止模式

在禁止模式下，模块不会进行发送或接收。由于总线活动，模块能够置 1 WAKIF 位，但是，等待处理的中断将继续等待，且错误计数器的值也将保持不变。

如果 $\text{REQOP}_{<2:0>} = 001$ ，模块将进入模块禁止模式。如果该模块处于活动状态，它将等候 CAN 总线上的 11 位隐性位，检测到总线空闲状态，然后接受模块禁止命令。当 $\text{OPMODE}_{<2:0>} = 001$ 时，这表明模块成功进入了模块禁止模式。当模块在模块禁止模式下时，I/O 引脚将恢复为普通 I/O 功能。

当模块或 CPU 处于休眠模式时，可以设置模块，在 CiRX 输入线上应用低通滤波器功能。由 WAKFIL 位 ($\text{CiCFG2}_{<14>} = 1$) 使能或禁止滤波器。

注：通常，如果允许 CAN 模块在某种工作模式下发送，并且在 CAN 模块进入该模式后立即被要求发送，则模块将在开始发送前等待总线上出现 11 个连续隐性位。如果用户在此 11 位期间切换到禁止模式，那么发送会被中止，同时相应的 TXABT 位置 1， TXREQ 位清零。

17.3.3 正常工作模式

当 $\text{REQOP}_{<2:0>} = 000$ 时，选择正常工作模式。在此模式下，模块被激活，I/O 引脚用作 CAN 总线功能。模块将通过 CxTX 和 CxRX 引脚，发送和接收 CAN 总线报文。

17.3.4 监听模式

如果激活监听模式，则 CAN 总线上的模块是被动的。发送器缓冲器恢复为端口 I/O 功能。接收引脚仍然为输入。对于接收器，不发出错误标志或应答信号。该状态下，错误计数器失效。监听模式可用来检测 CAN 总线上的波特率。要使用监听模式，必须有至少 2 个可以互相通信的节点。

17.3.5 监听所有报文模式

可以设置模块，使它忽略所有的错误并接收任何报文。通过设置 $\text{REQOP}_{<2:0>} = 111$ ，激活监听所有报文模式。在此模式下，报文组合缓冲器中一直到错误发生前的数据，将被复制到接收缓冲器中，可通过 CPU 接口读取。

17.3.6 环回模式

如果环回模式被激活，模块将在模块边界，把内部发送信号连接到内部接收信号。发送和接收引脚将转换到其 I/O 端口功能。

17.4 报文接收

17.4.1 接收缓冲器

CAN 总线模块有 3 个接收缓冲器。但是，其中一个缓冲器总是用于监视总线是否有进入的报文。这个缓冲器叫做报文组合缓冲器 (Message Assembly Buffer, MAB)。因此只有 2 个可见的接收缓冲器 (RXB0 和 RXB1)，基本上可以即时接收来自协议引擎的完整报文。

MAB 将组合所有接收到的报文，这些报文只有在满足接收过滤器判据时才被传送到 RXBn 缓冲器。当接收到报文时，RXnIF 标志 (CiINTF<0> 或 CiINRF<1>) 将置 1。此位仅在接收到报文时才能被模块置 1。该位在 CPU 处理完缓冲器中的报文后由 CPU 清零。如果 RXnIE 位 (CiINTE<0> 或 CiINTE<1>) 置 1，当接收到报文时将会产生一个中断。

RXF0 和 RXF1 过滤器，以及 RXM0 屏蔽器与 RXB0 相关联。过滤器 RXF2、RXF3、RXF4 和 RXF5，以及屏蔽器 RXM1 与 RXB1 关联。

17.4.2 报文接收过滤器

报文接收过滤器和屏蔽寄存器用于决定报文组合缓冲器中的一条报文是否应该被装入接收缓冲器中的一个。一旦一条有效报文被接收进入 MAB，就会将该报文的标识符字段与过滤值进行比较。如果匹配的话，该报文就会被装入相应的接收缓冲器。

接收过滤器检查进入的报文，根据 RXIDE 位 (CiRXnSID<0>) 决定如何比较标识符。如果 RXIDE 位清零，报文是标准帧，只与 EXIDE 位 (CiRXFnSID<0>) 清零的过滤器比较。如果 RXIDE 位置 1，报文是扩展帧，只与 EXIDE 位置 1 的过滤器比较。将 RXM<1:0> 位配置为 01 或 10，可以改写 EXIDE 位。

17.4.3 报文接收过滤屏蔽

屏蔽位基本上决定了哪些位应进行过滤。如果某屏蔽位设置为零，则该位将被自动接收，而与过滤位无关。有 2 个可编程接收过滤屏蔽器与接收缓冲器关联，每个缓冲器有一个过滤屏蔽器与其关联。

17.4.4 接收溢出

当 MAB 组合了一个有效的接收报文后，该报文通过接收过滤器接收，而此时与该过滤器关联的接收缓冲器还未指定将前一次接收的报文清除时，就会发生溢出。

溢出错误标志 RXnOVR (CiINTF<15> 或 CiINTF<14>) 和 ERRIF 位 (CiINTF<5>) 将置 1，MAB 中的报文会被丢弃。

如果 DBEN 位清零，则 RXB1 和 RXB0 独立工作。在这种情况下，如果 RXB0 中有未读的报文，而且 RXOOVR 位置 1 的话，则要存入 RXB0 的报文将不会被转存到 RXB1。

如果 DBEN 位置 1，处理 RXB0 溢出的方法就不同了。如果接收到一条将要存入 RXB0 的有效报文，且 RXFUL = 1 (RXB0 满)、RXFUL = 0 (RXB1 空)，RXB0 的报文将装入 RXB1。此时不会为 RXB0 产生溢出错误。如果接收到一条将要存入 RXB0 的有效报文且 RXFUL = 1 (表明 RXB0 和 RXB1 都为满)，这样报文将被丢弃并产生 RXB1 溢出。

17.4.5 接收错误

CAN 模块将会检测到以下接收错误：

- 循环冗余校验 (Cyclic Redundancy Check , CRC) 错误
- 位填充错误
- 无效报文接收错误

这些接收错误不会产生中断。然而，当发生上述错误之一时，接收错误计数器会递增 1。置 1 RXWAR 位 (CiINTF<9>)，表明接收错误计数器已经达到 CPU 警告上限值 96，接着产生中断。

17.4.6 接收中断

接收中断可划分为三大类，每类包括各种产生中断的条件：

- 接收中断：
报文已被成功接收并被装入一个接收缓冲器中。接收到帧结束 (End-of-Frame, EOF) 字段后，中断立即被激活。读 RXnIF 标志位可知哪个接收缓冲器引起了中断。
- 唤醒中断：
CAN 模块已经从禁止模式中唤醒，或者器件已经从休眠模式中唤醒。

- 接收错误中断:

接收错误中断由 **ERRIF** 位表明。该位表明产生了错误条件。通过检查 CAN 中断状态寄存器中的位 **CiINTF**, 可以决定错误的来源。

- 无效的报文接收:

如果在最后一条报文的接收过程中, 产生了任何类型的错误, **IVRIF** 位将表明出现错误。

- 接收器溢出:

RXnOVR 位表明是否出现溢出。

- 接收器警告:

RXWAR 位表明, 是否接收错误计数器 (**RERRCNT<7:0>**) 已经到达 96 的警告上限。

- 接收器错误被动状态:

RXEPE 位表明, 是否接收错误计数器已经超过了 127 的错误被动上限, 模块已经进入错误被动状态。

17.5 报文发送

17.5.1 发送缓冲器

CAN 模块有 3 个发送缓冲器。每个缓冲器可容纳 14 字节的数据。其中 8 个字节用于存放发送的报文 (最大 8 个字节)。另外 5 个字节用来存放标准或扩展标识符和其他报文仲裁信息。

17.5.2 发送报文优先级

发送优先级指在各个节点内待发送报文的优先级。发送优先级有 4 级。如果某个报文缓冲器的 **TXPRI<1:0>** (**CiTXnCON<1:0>**, 其中 $n = 0, 1$ 或 2 , 代表具体的发送缓冲器) 置为 11, 则该缓冲器具有最高优先级。如果某个报文缓冲器的 **TXPRI<1:0>** 被置为 10 或 01, 则该缓冲器具有中等优先级。如果某个报文缓冲器的 **TXPRI<1:0>** 是 00, 则该缓冲器优先级最低。

17.5.3 发送过程

必须把 **TXREQ** 位 (**CiTXnCON<3>**) 置 1 来开始发送报文。CAN 总线模块解决了由 **TXREQ** 位设定与帧起始 (**Start-of-Frame, SOF**) 之间的任何时序冲突, 确保在优先级改变时, 能在发送 **SOF** 之前正确解决时序冲突。当 **TXREQ** 置 1 时, **TXABT** (**CiTXnCON<6>**)、**TXLARB** (**CiTXnCON<5>**) 和 **TXERR** (**CiTXnCON<4>**) 标志位将自动清零。

置 1 **TXREQ** 位, 仅仅表示报文缓冲器正在排队以等待发送。当模块检测到总线可用时, 就开始发送具有最高优先级的报文。

如果发送首次尝试就成功完成, **TXREQ** 位将自动清零, 如果 **TXnIE** 位置 1 的话还将产生中断。

如果报文发送失败, 一个错误状态标志位将置 1, **TXREQ** 位将保持置 1, 表示该报文仍然等待发送。如果在报文尝试发送期间遇到出错情况, **TXERR** 位将置 1, 出错情况可能会引起中断。如果报文尝试发送期间仲裁失败, **TXLARB** 位将置 1。没有中断产生表明仲裁失败。

17.5.4 中止报文发送

通过清零与各个报文缓冲器关联的 **TXREQ** 位, 系统能中止报文发送。置 1 **ABAT** 位 (**CiCTRL<12>**) 将请求中止所有待发送的报文。如果报文还未开始发送, 或者报文已开始发送但由于仲裁失败或错误而被中断, 那么将被中止。当模块置 1 **TXABT** 位时, 表明发生了中止, **TXnIF** 标志不会被自动置 1。

17.5.5 发送错误

CAN 模块将会检测到以下发送错误:

- 应答错误
- 格式错误
- 位错误

这些发送错误不一定会产生中断, 但是发送错误计数器将累计数以表示有错误发生。每个错误将引起错误计数器的值递增 1。一旦错误计数器的值超过 96, **ERRIF** (**CiINTF<5>**) 和 **TXWAR** 位 (**CiINTF<10>**) 将置 1。一旦错误计数器的值超过 96, 将产生中断, 错误标志寄存器中的 **TXWAR** 位将置 1。

17.5.6 发送中断

发送中断可划分为两大类，每类包括各种产生中断的条件：

- **发送中断：**

三个发送缓冲器中至少有一个为空（未预定），并且可以装入预定发送的报文。读 TXnIF 标志位可知哪一个发送缓冲器可用并引起了中断。

- **发送错误中断：**

发送错误中断由 ERRIF 标志位表明。该标志位表明产生了错误条件。通过检查 CAN 中断状态寄存器 CiINTF 中的错误标志，可以决定错误的来源。该寄存器中的标志位与接收和发送错误有关。

- **发送器警告中断：**

TXWAR 位表明，是否发送错误计数器已经到达 CPU 警告的上限值 96。

- **发送器错误被动状态：**

TXEP 位 (CiINTF<12>) 表明，是否发送错误计数器已经超过了 127 的错误被动上限，模块已经进入了错误被动状态。

- **总线关断：**

TXBO 位 (CiINTF<13>) 表明，是否发送错误计数器已经超过了 255，模块已经进入了总线关断状态。

17.6 波特率设置

任意特定 CAN 总线上的所有节点必须有相同的标称比特率。为了设置波特率，必须初始化下列参数：

- 同步跳转宽度位
- 波特率预分频器位
- 相位段位
- 相位段 2 的长度决定位
- 采样点位
- 传播段位

17.6.1 位时序

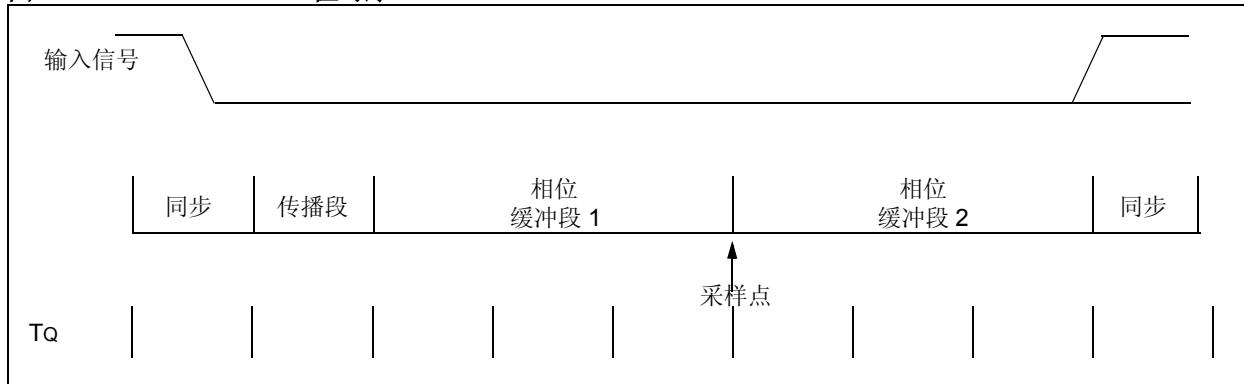
CAN 总线上的所有控制器都必须有相同的波特率和位长度。然而，不同的控制器并不要求使用相同的主振荡器时钟。如果各控制器的时钟频率不同，就不得不通过调节每个段的时间份额来调节波特率。

可以认为，标称位时间划分成几个互不重叠的时间段。这些段如图 17-2 所示。

- 同步段 (Sync Seg)
- 传播时间段 (Prop Seg)
- 相位缓冲段 1 (Phase1 Seg)
- 相位缓冲段 2 (Phase2 Seg)

时间段以及标称位时间由整数个时间单元组成，这些单元称作时间份额或 TQ。根据定义，标称位时间最小由 8 个 TQ 组成，最大由 25 TQ 组成。同样根据定义，最小标称位时间是 1 μs，对应最大 1 MHz 的比特率。

图 17-2: CAN 位时序



17.6.2 预分频器设置

有一个可编程预分频器，除了固定的二分频用于时钟发生以外，其整数预分频比范围为 1 至 64。时间份额 (T_Q) 是从振荡器周期得出的固定时间单元，由公式 17-1 给出，其中 F_{CAN} 为 F_{CY} （如果 CANCKS 位置 1）或 $4F_{CY}$ （如果 CANCKS 位清零）。

注： F_{CAN} 不能超出 30 MHz。如果 CANCKS = 0，那么 F_{CY} 不能超出 7.5 MHz。

公式 17-1：时钟发生的时间份额

$$T_Q = 2(BRP<5:0> + 1) / F_{CAN}$$

17.6.3 传播段

这部分的位时间用来补偿网络内的物理延时。这些延时包括总线线路上的信号传播时间以及节点的内部延时。通过设置 PRSEG<2:0> 位 (CiCFG2<2:0>)，可以设置传播段为 1 T_Q 到 8 T_Q 。

17.6.4 相位段

相位段用于在发送位时间内定位最佳的接收位采样。采样点在相位缓冲段 1 与相位缓冲段 2 之间。这两个段可以通过重新同步加长或缩短。相位缓冲段 1 的末尾决定在一个位周期内的采样点。该段可以编程为 1 T_Q 到 8 T_Q 。相位缓冲段 2 为下一个发送数据跳变提供一个延时。该段可以编程为 1 T_Q 到 8 T_Q ，也可将其定义为相位缓冲段 1 的时间份额与信息处理时间 (2 T_Q) 两者中的较大者。通过设置 SEG1PH<2:0> (CiCFG2<5:3>) 位初始化相位缓冲段 1，通过设置 SEG2PH<2:0> (CiCFG2<10:8>) 位初始化相位缓冲段 2。

在设置相位缓冲段的长度时，必须满足下面的要求：

$$\text{传播段} + \text{相位缓冲段 1} \geq \text{相位缓冲段 2}$$

17.6.5 采样点

采样点是个时间点，在此时间点将读取总线电平并将其解释为对应位的值。它的位置在相位缓冲段 1 的末尾。若位时序较慢而且包含很多 T_Q ，可以在同一个采样点指定总线线路的多个采样。由 CAN 总线决定的电平将对应于三值择多决策的结果。择多采样在采样点进行，且前两次采样相隔 $T_Q/2$ 。CAN 模块允许用户选择在同一点采样 3 次或 1 次。这可以通过置 1 或清零 SAM 位 (CiCFG2<6>) 实现。

通常，位采样应该发生在位时间的 60-70% 处，取决于系统参数。

17.6.6 同步

为了补偿不同总线站振荡器频率的相移，每个 CAN 控制器必须要能够与进入信号的相关信号沿同步。当检测到发送数据中的一个沿时，逻辑会将该沿的位置与预期时间（同步段）比较。然后电路将调节相位缓冲段 1 和相位缓冲段 2 的值。有两种机制用来同步。

17.6.6.1 硬同步

硬同步仅当总线空闲期间，有一个从“隐性”转变到“显性”的边沿时（表明报文开始），才会进行。硬同步后，位时间计数器从同步段重新开始计数。硬同步把引起硬同步的沿强制地置于重新开始的位时间同步段之内。如果完成一个硬同步，在该位时间内将不会再有重新同步。

17.6.6.2 重新同步

重新同步可能使相位缓冲段 1 加长或相位缓冲段 2 缩短。相位缓冲段加长或缩短的量有一个上限值，称作同步跳转宽度，由 SJW<1:0> 位 (CiCFG1<7:6>) 指定。同步跳转宽度的值将被加到相位缓冲段 1 或从相位缓冲段 2 减去。重新同步跳转宽度可以设置为 1 T_Q 到 4 T_Q 。

在设置 SJW<1:0> 位时，必须满足下面的要求：

$$\text{相位缓冲段 2} > \text{同步跳转宽度}$$

表 17-1: CAN1 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		
C1RXF0SID	0300	—	—	—	接收过滤器 0 标准标识符 <10:0>										—	EXIDE	000u uuuu uuuu uu0u			
C1RXF0EIDH	0302	—	—	—	—	接收过滤器 0 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu			
C1RXF0EIDL	0304	接收过滤器 0 扩展标识符 <5:0>				—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000			
C1RXF1SID	0308	—	—	—	接收过滤器 1 标准标识符 <10:0>										—	EXIDE	000u uuuu uuuu uu0u			
C1RXF1EIDH	030A	—	—	—	—	接收过滤器 1 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu			
C1RXF1EIDL	030C	接收过滤器 1 扩展标识符 <5:0>				—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000			
C1RXF2SID	0310	—	—	—	接收过滤器 2 标准标识符 <10:0>										—	EXIDE	000u uuuu uuuu uu0u			
C1RXF2EIDH	0312	—	—	—	—	接收过滤器 2 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu			
C1RXF2EIDL	0314	接收过滤器 2 扩展标识符 <5:0>				—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000			
C1RXF3SID	0318	—	—	—	接收过滤器 3 标准标识符 <10:0>										—	EXIDE	000u uuuu uuuu uu0u			
C1RXF3EIDH	031A	—	—	—	—	接收过滤器 3 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu			
C1RXF3EIDL	031C	接收过滤器 3 扩展标识符 <5:0>				—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000			
C1RXF4SID	0320	—	—	—	接收过滤器 4 标准标识符 <10:0>										—	EXIDE	000u uuuu uuuu uu0u			
C1RXF4EIDH	0322	—	—	—	—	接收过滤器 4 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu			
C1RXF4EIDL	0324	接收过滤器 4 扩展标识符 <5:0>				—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000			
C1RXF5SID	0328	—	—	—	接收过滤器 5 标准标识符 <10:0>										—	EXIDE	000u uuuu uuuu uu0u			
C1RXF5EIDH	032A	—	—	—	—	接收过滤器 5 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu			
C1RXF5EIDL	032C	接收过滤器 5 扩展标识符 <5:0>				—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000			
C1RXM0SID	0330	—	—	—	接收屏蔽器 0 标准标识符 <10:0>										—	MIDE	000u uuuu uuuu uu0u			
C1RXM0EIDH	0332	—	—	—	—	接收屏蔽器 0 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu			
C1RXM0EIDL	0334	接收屏蔽器 0 扩展标识符 <5:0>				—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000			
C1RXM1SID	0338	—	—	—	接收屏蔽器 1 标准标识符 <10:0>										—	MIDE	000u uuuu uuuu uu0u			
C1RXM1EIDH	033A	—	—	—	—	接收屏蔽器 1 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu			
C1RXM1EIDL	033C	接收屏蔽器 1 扩展标识符 <5:0>				—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000			
C1TX2SID	0340	发送缓冲器 2 标准标识符 <10:6>				—	—	—	发送缓冲器 2 标准标识符 <5:0>					SRR	TXIDE	uuuu u000 uuuu uuuu				
C1TX2EID	0342	发送缓冲器 2 扩展标识符 <17:14>				—	—	—	—	发送缓冲器 2 扩展标识符 <13:6>					—	uuuu 0000 uuuu uuuu				
C1TX2DLC	0344	发送缓冲器 2 扩展标识符 <5:0>				TXRTR	TXRB1	TXRB0	DLC<3:0>				—	—	—	—	uuuu uuuu uuuu u000			
C1TX2B1	0346	发送缓冲器 2 字节 1								发送缓冲器 2 字节 0								uuuu uuuu uuuu uuuu		
C1TX2B2	0348	发送缓冲器 2 字节 3								发送缓冲器 2 字节 2								uuuu uuuu uuuu uuuu		
C1TX2B3	034A	发送缓冲器 2 字节 5								发送缓冲器 2 字节 4								uuuu uuuu uuuu uuuu		
C1TX2B4	034C	发送缓冲器 2 字节 7								发送缓冲器 2 字节 6								uuuu uuuu uuuu uuuu		
C1TX2CON	034E	—	—	—	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI<1:0>	0000 0000 0000 0000	—	—		
C1TX1SID	0350	发送缓冲器 1 标准标识符 <10:6>				—	—	—	发送缓冲器 1 标准标识符 <5:0>					SRR	TXIDE	uuuu u000 uuuu uuuu	—	—		
C1TX1EID	0352	发送缓冲器 1 扩展标识符 <17:14>				—	—	—	发送缓冲器 1 扩展标识符 <13:6>					—	—	uuuu 0000 uuuu uuuu	—	—		
C1TX1DLC	0354	发送缓冲器 1 扩展标识符 <5:0>				TXRTR	TXRB1	TXRB0	DLC<3:0>				—	—	—	—	uuuu uuuu uuuu u000	—	—	
C1TX1B1	0356	发送缓冲器 1 字节 1								发送缓冲器 1 字节 0								uuuu uuuu uuuu uuuu	—	—

图注: u = 未初始化位

表 17-1: CAN1 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		
C1TX1B2	0358	发送缓冲器 1 字节 3								发送缓冲器 1 字节 2								uuuu uuuu uuuu uuuu		
C1TX1B3	035A	发送缓冲器 1 字节 5								发送缓冲器 1 字节 4								uuuu uuuu uuuu uuuu		
C1TX1B4	035C	发送缓冲器 1 字节 7								发送缓冲器 1 字节 6								uuuu uuuu uuuu uuuu		
C1TX1CON	035E	—	—	—	—	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI<1:0>	0000 0000 0000 0000			
C1TX0SID	0360	发送缓冲器 0 标准标识符 <10:6>								发送缓冲器 0 标准标识符 <5:0>								SRR TXIDE	uuuu u000 uuuu uuuu	
C1TX0EID	0362	发送缓冲器 0 扩展标识符 <17:14>				—	—	—	—	发送缓冲器 0 扩展标识符 <13:6>								uuuu 0000 uuuu uuuu		
C1TX0DLC	0364	发送缓冲器 0 扩展标识符 <5:0>								TXRTR	TXRB1	TXRB0	DLC<3:0>				—	—	—	uuuu uuuu uuuu u000
C1TX0B1	0366	发送缓冲器 0 字节 1								发送缓冲器 0 字节 0								uuuu uuuu uuuu uuuu		
C1TX0B2	0368	发送缓冲器 0 字节 3								发送缓冲器 0 字节 2								uuuu uuuu uuuu uuuu		
C1TX0B3	036A	发送缓冲器 0 字节 5								发送缓冲器 0 字节 4								uuuu uuuu uuuu uuuu		
C1TX0B4	036C	发送缓冲器 0 字节 7								发送缓冲器 0 字节 6								uuuu uuuu uuuu uuuu		
C1TX0CON	036E	—	—	—	—	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI<1:0>	0000 0000 0000 0000			
C1RX1SID	0370	—	—	—	接收缓冲器 1 标准标识符 <10:0>								SRR RXIDE	000u uuuu uuuu uuuu						
C1RX1EID	0372	—	—	—	—	接收缓冲器 1 扩展标识符 <17:6>								0000 uuuu uuuu uuuu						
C1RX1DLC	0374	接收缓冲器 1 扩展标识符 <5:0>								RXRTR	RXRB1	—	—	—	RXRB0	DLC<3:0>				uuuu uuuu 000 uuuu
C1RX1B1	0376	接收缓冲器 1 字节 1								接收缓冲器 1 字节 0								uuuu uuuu uuuu uuuu		
C1RX1B2	0378	接收缓冲器 1 字节 3								接收缓冲器 1 字节 2								uuuu uuuu uuuu uuuu		
C1RX1B3	037A	接收缓冲器 1 字节 5								接收缓冲器 1 字节 4								uuuu uuuu uuuu uuuu		
C1RX1B4	037C	接收缓冲器 1 字节 7								接收缓冲器 1 字节 6								uuuu uuuu uuuu uuuu		
C1RX1CON	037E	—	—	—	—	—	—	—	—	RXFUL	—	—	—	RXRTRRO	FILHIT<2:0>		0000 0000 0000 0000			
C1RX0SID	0380	—	—	—	接收缓冲器 0 标准标识符 <10:0>								SRR RXIDE	000u uuuu uuuu uuuu						
C1RX0EID	0382	—	—	—	—	接收缓冲器 0 扩展标识符 <17:6>								0000 uuuu uuuu uuuu						
C1RX0DLC	0384	接收缓冲器 0 扩展标识符 <5:0>								RXRTR	RXRB1	—	—	—	RXRB0	DLC<3:0>				uuuu uuuu 000 uuuu
C1RX0B1	0386	接收缓冲器 0 字节 1								接收缓冲器 0 字节 0								uuuu uuuu uuuu uuuu		
C1RX0B2	0388	接收缓冲器 0 字节 3								接收缓冲器 0 字节 2								uuuu uuuu uuuu uuuu		
C1RX0B3	038A	接收缓冲器 0 字节 5								接收缓冲器 0 字节 4								uuuu uuuu uuuu uuuu		
C1RX0B4	038C	接收缓冲器 0 字节 7								接收缓冲器 0 字节 6								uuuu uuuu uuuu uuuu		
C1RX0CON	038E	—	—	—	—	—	—	—	—	RXFUL	—	—	—	RXRTRRO	DBEN JTOFF FILHITO	0000 0000 0000 0000				
C1CTRL	0390	CANCAP	—	CSIDLE	ABAT	CANCKS	REQOP<2:0>				OPMODE<2:0>				—	ICODE<2:0>	—	0000 0100 1000 0000		
C1CFG1	0392	—	—	—	—	—	—	—	—	SJW<1:0>		BRP<5:0>						0000 0000 0000 0000		
C1CFG2	0394	—	WAKFIL	—	—	—	SEG2PH<2:0>				SEG2PHTS	SAM	SEG1PH<2:0>				PRSEG<2:0>	0u00 0uuu uuuu uuuu		
C1INTF	0396	RX0OVR	RX1OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN	IVRIF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF	0000 0000 0000 0000		
C1INTE	0398	—	—	—	—	—	—	—	—	IVRIE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1IE	RX0IE	0000 0000 0000 0000		
C1EC	039A	发送错误计数寄存器								接收错误计数寄存器								0000 0000 0000 0000		

图注: u = 未初始化位

注: 有关寄存器各位的说明, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

表 17-2: CAN2 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					
C2RXF0SID	03C0	—	—	—	接收过滤器 0 标准标识符 <10:0>										—	EXIDE	000u uuuu uuuu uu0u						
C2RXF0EIDH	03C2	—	—	—	—	接收过滤器 0 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu						
C2RXF0EIDL	03C4	接收过滤器 0 扩展标识符 <5:0>					—	—	—	—	—	—	—	—	—	—	uuuu uuu0 0000 0000						
C2RXF1SID	03C8	—	—	—	接收过滤器 1 标准标识符 <10:0>										—	EXIDE	000u uuuu uuuu uu0u						
C2RXF1EIDH	03CA	—	—	—	—	接收过滤器 1 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu						
C2RXF1EIDL	03CC	接收过滤器 1 扩展标识符 <5:0>					—	—	—	—	—	—	—	—	—	—	uuuu uuu0 0000 0000						
C2RXF2SID	03D0	—	—	—	接收过滤器 2 标准标识符 <10:0>										—	EXIDE	000u uuuu uuuu uu0u						
C2RXF2EIDH	03D2	—	—	—	—	接收过滤器 2 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu						
C2RXF2EIDL	03D4	接收过滤器 2 扩展标识符 <5:0>					—	—	—	—	—	—	—	—	—	—	uuuu uuu0 0000 0000						
C2RXF3SID	03D8	—	—	—	接收过滤器 3 标准标识符 <10:0>										—	EXIDE	000u uuuu uuuu uu0u						
C2RXF3EIDH	03DA	—	—	—	—	接收过滤器 3 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu						
C2RXF3EIDL	03DC	接收过滤器 3 扩展标识符 <5:0>					—	—	—	—	—	—	—	—	—	—	uuuu uuu0 0000 0000						
C2RXF4SID	03E0	—	—	—	接收过滤器 4 标准标识符 <10:0>										—	EXIDE	000u uuuu uuuu uu0u						
C2RXF4EIDH	03E2	—	—	—	—	接收过滤器 4 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu						
C2RXF4EIDL	03E4	接收过滤器 4 扩展标识符 <5:0>					—	—	—	—	—	—	—	—	—	—	uuuu uuu0 0000 0000						
C2RXF5SID	03E8	—	—	—	接收过滤器 5 标准标识符 <10:0>										—	EXIDE	000u uuuu uuuu uu0u						
C2RXF5EIDH	03EA	—	—	—	—	接收过滤器 5 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu						
C2RXF5EIDL	03EC	接收过滤器 5 扩展标识符 <5:0>					—	—	—	—	—	—	—	—	—	—	uuuu uuu0 0000 0000						
C2RXM0SID	03F0	—	—	—	接收屏蔽器 0 标准标识符 <10:0>										—	MIDE	000u uuuu uuuu uu0u						
C2RXM0EIDH	03F2	—	—	—	—	接收屏蔽器 0 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu						
C2RXM0EIDL	03F4	接收屏蔽器 0 扩展标识符 <5:0>					—	—	—	—	—	—	—	—	—	—	uuuu uuu0 0000 0000						
C2RXM1SID	03F8	—	—	—	接收屏蔽器 1 标准标识符 <10:0>										—	MIDE	000u uuuu uuuu uu0u						
C2RXM1EIDH	03FA	—	—	—	—	接收屏蔽器 1 扩展标识符 <17:6>										—	0000 uuuu uuuu uuuu						
C2RXM1EIDL	03FC	接收屏蔽器 1 扩展标识符 <5:0>					—	—	—	—	—	—	—	—	—	—	uuuu uuu0 0000 0000						
C2TX2SID	0400	发送缓冲器 2 标准标识符 <10:6>					—	—	—	发送缓冲器 2 标准标识符 <5:0>										SRR TXIDE	uuuu u000 uuuu uuuu		
C2TX2EID	0402	发送缓冲器 2 扩展标识符 <17:14>					—	—	—	发送缓冲器 2 扩展标识符 <13:6>										uuuu 0000 uuuu uuuu			
C2TX2DLC	0404	发送缓冲器 2 扩展标识符 <5:0>					TXRTR	TXRB1	TXRB0	DLC<3:0>					—	—	—	uuuu uuuu uuuu u000					
C2TX2B1	0406	发送缓冲器 2 字节 1										发送缓冲器 2 字节 0								uuuu uuuu uuuu uuuu			
C2TX2B2	0408	发送缓冲器 2 字节 3										发送缓冲器 2 字节 2								uuuu uuuu uuuu uuuu			
C2TX2B3	040A	发送缓冲器 2 字节 5										发送缓冲器 2 字节 4								uuuu uuuu uuuu uuuu			
C2TX2B4	040C	发送缓冲器 2 字节 7										发送缓冲器 2 字节 6								uuuu uuuu uuuu uuuu			
C2TX2CON	040E	—	—	—	—	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI<1:0>		0000 0000 0000 0000					
C2TX1SID	0410	发送缓冲器 1 标准标识符 <10:6>					—	—	—	发送缓冲器 1 标准标识符 <5:0>										SRR TXIDE	uuuu u000 uuuu uuuu		
C2TX1EID	0412	发送缓冲器 1 扩展标识符 <17:14>					—	—	—	发送缓冲器 1 扩展标识符 <13:6>										uuuu 0000 uuuu uuuu			
C2TX1DLC	0414	发送缓冲器 1 扩展标识符 <5:0>					TXRTR	TXRB1	TXRB0	DLC<3:0>					—	—	—	uuuu uuuu uuuu u000					

图注: u = 未初始化位

表 17-2: CAN2 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			
C2TX1B1	0416	发送缓冲器 1 字节 1								发送缓冲器 1 字节 0								uuuu uuuu uuuu uuuu			
C2TX1B2	0418	发送缓冲器 1 字节 3								发送缓冲器 1 字节 2								uuuu uuuu uuuu uuuu			
C2TX1B3	041A	发送缓冲器 1 字节 5								发送缓冲器 1 字节 4								uuuu uuuu uuuu uuuu			
C2TX1B4	041C	发送缓冲器 1 字节 7								发送缓冲器 1 字节 6								uuuu uuuu uuuu uuuu			
C2TX1CON	041E	—	—	—	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI<1:0>		0000 0000 0000 0000				
C2TX0SID	0420	发送缓冲器 0 标准标识符 <10:6>								发送缓冲器 0 标准标识符 <5:0>								SRR TXIDE	uuuu u000 uuuu uuuu		
C2TX0EID	0422	发送缓冲器 0 扩展标识符 <17:14>								—	—	—	发送缓冲器 0 扩展标识符 <13:6>								uuuu 0000 uuuu uuuu
C2TX0DLC	0424	发送缓冲器 0 扩展标识符 <5:0>								TXRTR	TXRB1	TXRB0	DLC<3:0>				—	—	—	uuuu uuuu uuuu u000	
C2TX0B1	0426	发送缓冲器 0 字节 1								发送缓冲器 0 字节 0								uuuu uuuu uuuu uuuu			
C2TX0B2	0428	发送缓冲器 0 字节 3								发送缓冲器 0 字节 2								uuuu uuuu uuuu uuuu			
C2TX0B3	042A	发送缓冲器 0 字节 5								发送缓冲器 0 字节 4								uuuu uuuu uuuu uuuu			
C2TX0B4	042C	发送缓冲器 0 字节 7								发送缓冲器 0 字节 6								uuuu uuuu uuuu uuuu			
C2TX0CON	042E	—	—	—	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI<1:0>		0000 0000 0000 0000				
C2RX1SID	0430	—	—	—	接收缓冲器 1 标准标识符 <10:0>								SRR RXIDE	000u uuuu uuuu uuuu							
C2RX1EID	0432	—	—	—	—	接收缓冲器 1 扩展标识符 <17:6>												0000 uuuu uuuu uuuu			
C2RX1DLC	0434	接收缓冲器 1 扩展标识符 <5:0>								RXRTR	RXRB1	—	—	—	RXRBO	DLC<3:0>				uuuu uuuu 000u uuuu	
C2RX1B1	0436	接收缓冲器 1 字节 1								接收缓冲器 1 字节 0								uuuu uuuu uuuu uuuu			
C2RX1B2	0438	接收缓冲器 1 字节 3								接收缓冲器 1 字节 2								uuuu uuuu uuuu uuuu			
C2RX1B3	043A	接收缓冲器 1 字节 5								接收缓冲器 1 字节 4								uuuu uuuu uuuu uuuu			
C2RX1B4	043C	接收缓冲器 1 字节 7								接收缓冲器 1 字节 6								uuuu uuuu uuuu uuuu			
C2RX1CON	043E	—	—	—	—	—	—	—	—	RXFUL	—	—	—	RXRTRRO	FILHIT<2:0>		0000 0000 0000 0000				
C2RX0SID	0440	—	—	—	接收缓冲器 0 标准标识符 <10:0>								SRR RXIDE	000u uuuu uuuu uuuu							
C2RX0EID	0442	—	—	—	—	接收缓冲器 0 扩展标识符 <17:6>												0000 uuuu uuuu uuuu			
C2RX0DLC	0444	接收缓冲器 0 扩展标识符 <5:0>								RXRTR	RXRB1	—	—	—	RXRBO	DLC<3:0>				uuuu uuuu 000u uuuu	
C2RX0B1	0446	接收缓冲器 0 字节 1								接收缓冲器 0 字节 0								uuuu uuuu uuuu uuuu			
C2RX0B2	0448	接收缓冲器 0 字节 3								接收缓冲器 0 字节 2								uuuu uuuu uuuu uuuu			
C2RX0B3	044A	接收缓冲器 0 字节 5								接收缓冲器 0 字节 4								uuuu uuuu uuuu uuuu			
C2RX0B4	044C	接收缓冲器 0 字节 7								接收缓冲器 0 字节 6								uuuu uuuu uuuu uuuu			
C2RX0CON	044E	—	—	—	—	—	—	—	—	RXFUL	—	—	—	RXRTRRO	DBEN	JTOFF	FILHITO	0000 0000 0000 0000			
C2CTRL	0450	CANCAP	—	CSIDLE	ABAT	CANCKS	REQOP<2:0>			OPMODE<2:0>				—	ICODE<2:0>			—	0000 0100 1000 0000		
C2CFG1	0452	—	—	—	—	—	—	—	—	SJW<1:0>		BRP<5:0>						0000 0000 0000 0000			
C2CFG2	0454	—	WAKFIL	—	—	—	SEG2PH<2:0>			SEG2PHTS	SAM	SEG1PH<2:0>			PRSEG<2:0>			0u00 0uuu uuuu uuuu			
C2INTF	0456	RX0OVR	RX1OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN	IVRIF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF	0000 0000 0000 0000			
C2INTE	0458	—	—	—	—	—	—	—	—	IVRIE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1IE	RX0IE	0000 0000 0000 0000			
C2EC	045A	发送错误计数寄存器								接收错误计数寄存器										0000 0000 0000 0000	

图注: u = 未初始化位

注: 有关寄存器各位的说明, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

dsPIC30F5011/5013

注：

18.0 数据转换器接口（DCI）模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但是不应把本手册当作无所不包的参考手册来使用。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)。

18.1 模块简介

dsPIC30F 数据转换器接口 (Data Converter Interface, DCI) 模块允许与如下器件的简单接口：音频编码器 / 解码器 (coder/decoder, codec)、A/D 转换器和 D/A 转换器等。支持以下接口：

- 帧同步串行传输 (单通道或多通道)
- Inter-IC Sound (I²S) 接口
- AC-Link 兼容模式

DCI 模块提供下列一般特性：

- 可设置的字长度，最大为 16 位
- 支持最大 16 时隙，帧最长为 256 位
- 数据缓冲最高为 4 个采样，无 CPU 开销

18.2 模块 I/O 引脚

有 4 个 I/O 引脚与模块相关。一旦使能，模块控制这 4 个引脚的数据方向。

18.2.1 CSCK 引脚

CSCK 引脚为 DCI 模块提供串行时钟。使用 DCICON2 SFR 中的 CSCKD 控制位，可以把 CSCK 引脚配置为输入或输出。当 CSCK 引脚配置为输出时，串行时钟由 dsPIC30F 器件提供。当 CSCK 引脚配置为输入时，串行时钟必须由外部器件提供。

18.2.2 CSDO 引脚

当模块使能时，串行数据输出 (CSDO) 引脚被配置为只能输出的引脚。无论何时要发送数据，CSDO 引脚就会驱动串行总线。在不发送数据的 CSCK 周期内，CSDO 引脚可以为三态或驱动为 0，这取决于 CSDOM 控制位的状态。这样，其他器件就能在 DCI 模块不使用的发送周期内，把数据放到串行总线上去。

18.2.3 CSDI 引脚

当模块使能时，串行数据输入 (CSDI) 引脚被配置为只能输入的引脚。

18.2.3.1 COFS 引脚

编解码器帧同步 (COFS) 引脚用来同步发生在 CSDO 和 CSDI 引脚上的数据传输。COFS 引脚可以配置为输入或输出。COFS 引脚的数据方向由 DCICON1 寄存器中的 COFSD 控制位决定。

当 CPU 正在访问存储器映射的缓冲器寄存器时，DCI 模块访问影子寄存器。

18.2.4 缓冲器数据对齐

数据值在缓冲器中总是以左对齐方式存储，因为大多数编解码器数据是用有符号二进制补码小数表示的。如果接收到的字长度小于 16 位，则接收缓冲寄存器中未使用的 LSb 会被模块置为 0。如果发送字长小于 16 位，模块将忽略发送缓冲寄存器中未使用的 LSb。本文档的后续小节将说明字长的设置。

18.2.5 发送 / 接收移位寄存器

DCI 模块具备一个 16 位移位寄存器，用来把串行数据移入和移出模块。数据首先移入 / 移出移位寄存器的 MSb，因为音频 PCM 数据是以有符号二进制补码格式发送的。

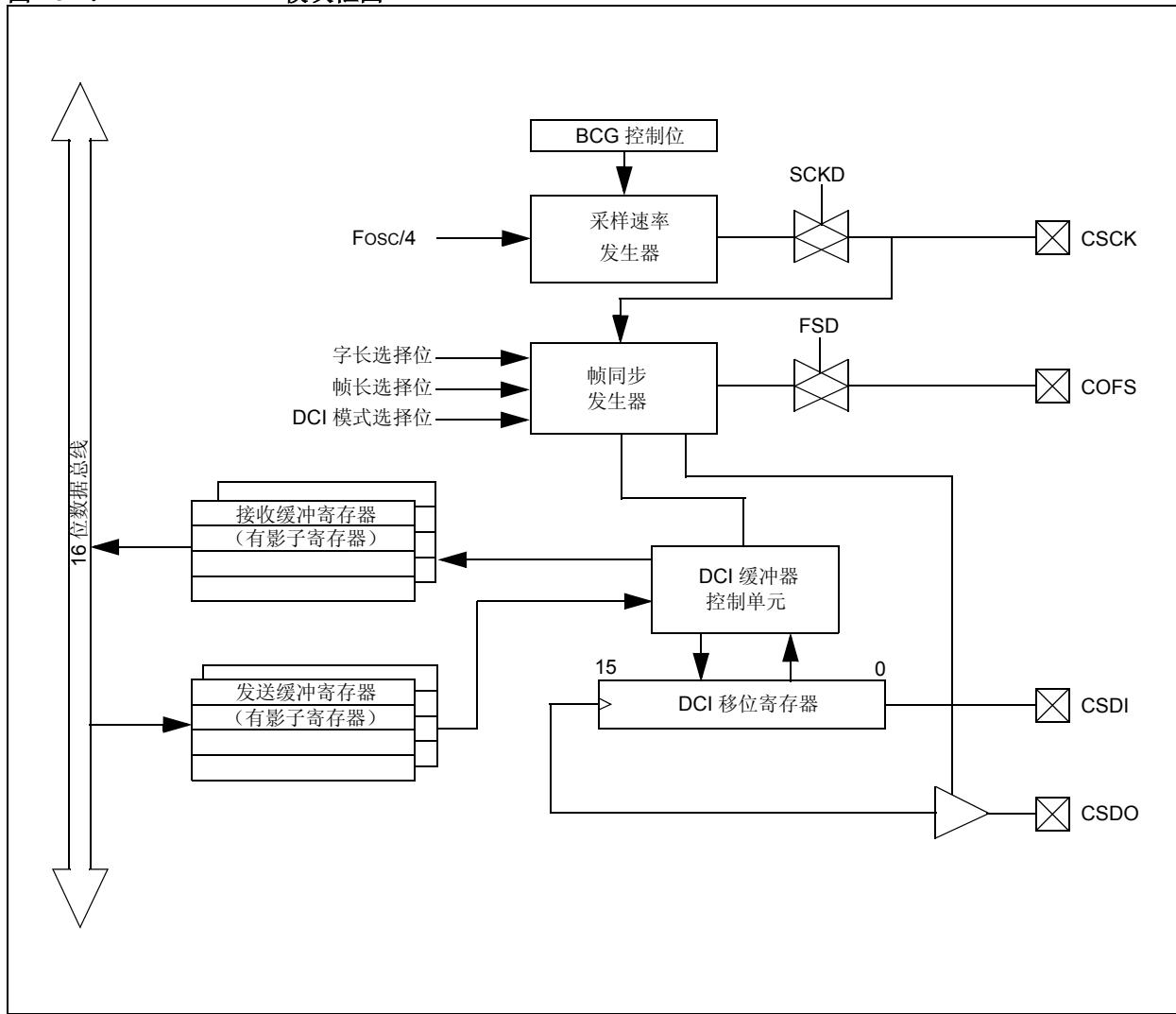
18.2.6 DCI 缓冲器控制

DCI 模块包含一个缓冲器控制单元，用来在影子缓冲存储区和串行移位寄存器之间传输数据。缓冲器控制单元是个简单的 2 位地址计数器，指向影子缓冲存储区中的字存储单元。对于接收存储空间 (DCI 缓冲存储区的高地址部分)，地址计数器值作为低位，与 MSb 位的 0 组合形成 3 位宽的地址。对于发送存储空间 (DCI 缓冲存储区的高地址部分)，地址计数器值作为低位，与 MSb 位的 1 组合形成 3 位宽的地址。

注： DCI 缓冲器控制单元始终访问发送和接收缓冲器中相同的相对位置，所以器件只提供了一个地址计数器。

dsPIC30F5011/5013

图 18-1: DCI 模块框图



18.3 DCI 模块的工作

18.3.1 模块使能

DCI 模块的使能和禁止是通过置 1/清零 DCICON1 SFR 中的 DCIEN 控制位来进行的。清零 DCIEN 控制位将复位模块。尤其是，与 CSCK 发生、帧同步和 DCI 缓冲器控制单元相关的所有计数器都将复位。

当 DCIEN 位清零时，DCI 时钟关闭。

当 DCI 使能时，它控制与模块相关的 4 个 I/O 引脚的数据方向。当 DCIEN 位置 1 时，这些 I/O 引脚的端口、LAT 和 TRIS 寄存器的值都会被 DCI 模块改写。

当使能了位时钟发生器时，单独改写 CSCK 引脚的上述寄存器值也是允许的。这可以无需使能 DCI 模块的其余部分，就能让位时钟发生器工作。

18.3.2 字长选择位

DCICON2 SFR 中的 WS<3:0>字长选择位决定每个 DCI 数据字的位数。本质上，WS<3:0>位决定 4 位计数器的计数周期（计数器时钟来自 CSCK 信号）。

可以选择任何长度的数据字长，字长最大 16 位。装入到 WS<3:0>位中的值，要比期望的字长小 1。例如，如果 WS<3:0> = 1111，则选择 16 位数据字长。

注： WS<3:0> 控制位仅用在多通道和 I²S 模式中。它们在 AC-Link 模式中没有作用，因为此时协议规定数据间隙的大小固定不变。

18.3.3 帧同步发生器

帧同步发生器（Frame Sync Generator, FSG）是一个 4 位计数器，它设置数据字中的帧长度。帧同步发生器在每次字长计数器复位时（第 18.3.2 节“字长选择位”）递增。通过写 DCICON2 SFR 中的 COFSG<3:0>控制位，来设置帧同步发生器的周期。时钟周期中的 COFSG 周期由以下公式决定：

公式 18-1： COFSG 周期

$$\text{帧长} = \text{字长 (FSG 值 + 1)}$$

可以选择的帧长度最大为 16 个数据字。CSCK 周期中的帧长度会随选择的字长而变化，最大可以为 256。

注： COFSG 控制位在 AC-Link 模式中不起作用，因为该协议已经将帧长度设置为 256 个 CSCK 周期。

18.3.4 帧同步模式控制位

使用 DCICON1 SFR 中的帧同步模式控制位（COFSM<1:0>），来选择帧同步信号的类型。可选择以下工作模式：

- 多通道模式
- I²S 模式
- AC-Link 模式（16 位）
- AC-Link 模式（20 位）

COFSM 控制位的工作，取决于 DCI 模块是作为主器件产生帧同步信号，还是作为从器件接收帧同步信号。

DSP/Codec 对中的主器件是产生帧同步信号的器件。帧同步信号启动 CSDI 和 CSDO 引脚上的数据传输，通常与数据采样速率（COFS）具有相同的频率。

如果 COFSD 控制位清零，则 DCI 模块是帧同步主器件；如果 COFSD 控制位置 1，则 DCI 模块是帧同步从器件。

18.3.5 主器件帧同步的工作

当 DCI 模块作为帧同步主器件（COFSD = 0）工作时，COFSM 模式位决定帧同步脉冲的类型；帧同步脉冲由帧同步发生器逻辑产生。

当帧同步发生器复位至 0 时，将产生新的 COFS 信号。

在多通道模式下，帧同步脉冲被驱动为高电平一个 CSCK 周期，从而启动数据传输。相继帧同步脉冲之间的 CSCK 周期数，取决于字长和帧同步发生器控制位。多通道模式下帧同步信号的时序图如图 18-2 所示。

在 AC-Link 工作模式下，帧同步信号有固定的周期和占空比。AC-Link 帧同步信号为高的时间为 16 个 CSCK 周期，为低的时间为 240 个 CSCK 周期。时序图以及 AC-Link 帧起始处的时序细节，如图 18-3 所示。

在 I²S 模式下，产生占空比为 50% 的帧同步信号。I²S 帧同步信号的周期（表示为 CSCK 周期数）由字长和帧同步发生器控制位决定。COFS 引脚上电平从高至低或从低至高的跳变，标记出了新的 I²S 数据传输的边界。

18.3.6 从器件帧同步的工作

当 DCI 模块作为帧同步从器件工作时 ($\text{COFS} = 1$)，数据传输由连接到DCI模块的Codec器件控制。COFSM 控制位控制 DCI 模块如何响应进入的 COFS 信号。

在多通道模式下，在 COFS 引脚采样为高电平后的一个 CSCK 周期，将会开始新的数据帧传输（见图 18-2）。COFS 引脚上的脉冲会复位帧同步发生器逻辑。

在 I^2S 模式下，在 COFS 引脚上采样到电平从低至高或从高至低跳变后的一个 CSCK 周期，会传输新的数据字。COFS 引脚的上升或下降沿将复位帧同步发生器逻辑。

在 AC-Link 模式下，当在 COFS 引脚采样到高电平后的一个 CSCK 周期，将传输下一个帧的标记间隙和接下来的数据间隙。

当模块工作在从模式时，必须配置 COFSG 和 WS 位，提供恰当的帧长度。一旦模块在 COFS 引脚上采样到了有效的帧同步脉冲，就会发生一个完整的数据帧传输。在当前数据帧传输完成之前，模块都不会响应其他帧同步脉冲。

图 18-2：帧同步时序，多通道模式

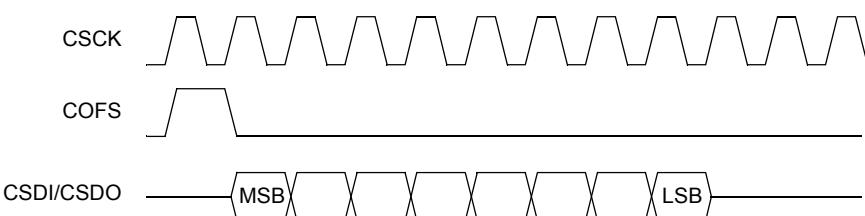


图 18-3：帧同步时序，AC-LINK 帧起始

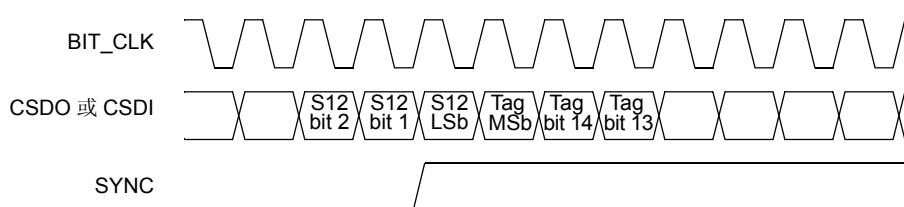
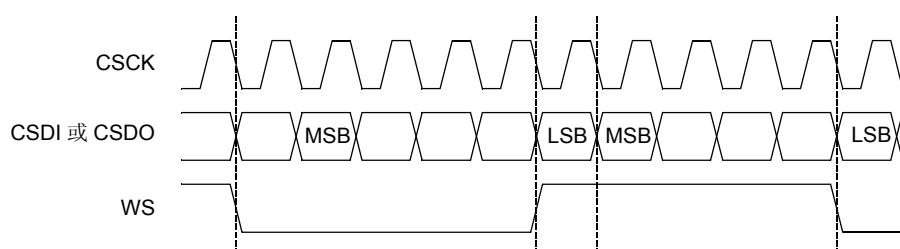


图 18-4： I^2S 接口帧同步时序



注：为便于说明，此处显示了 5 位传输。 I^2S 协议没有规定字长，字长取决于系统。

18.3.7 位时钟发生器

DCI 模块有一个专用的 12 位时基，用来产生位时钟。通过写一个非零 12 位值到 DCICON3 SFR 中的 BCG<11:0> 控制位，设置位时钟的速率（周期）。

当 BCG<11:0> 位设置为零时，位时钟将禁止。如果把 BCG<11:0> 位设置为非零值，位时钟发生器被使能。如果 DCI 的串行时钟来自外部器件，这些位应该置为 0，且 CSCKD 位置为 1。

位时钟频率的公式由公式 18-2 给出。

公式 18-2: 位时钟频率

$$FBCK = \frac{FCY}{2 \cdot (BCG + 1)}$$

所需的位时钟频率将取决于系统采样速率和帧大小。根据数据转换器和使用的通信协议，典型的位时钟频率范围在 16 倍到 512 倍转换器采样速率之间。

为了获得与常见音频采样速率相关的位时钟频率，用户需要选择的晶体频率应该是“偶”数二进制值。表 18-1 列出了这样的晶体频率的例子。

表 18-1: 用于常见 CODEC CSCK 频率的器件频率

Fs (KHz)	Fcsck/Fs	Fcsck (MHz) ⁽¹⁾	Fosc (MHz)	PLL	Fcyc (MIPS)	BCG ⁽²⁾
8	256	2.048	8.192	4	8.192	1
12	256	3.072	6.144	8	12.288	1
32	32	1,024	8.192	8	16.384	7
44.1	32	1.4112	5.6448	8	11.2896	3
48	64	3.072	6.144	16	24.576	3

注 1: 当 CSCK 信号来自外部时 (CSCKD = 1)，外部时钟的高低电平时间必须满足器件时序要求。

2: 当 CSCK 信号来自外部时 (CSCKD = 1)，BCG<11:0> 位对 DCI 模块的工作不产生影响。

18.3.8 采样时钟边沿控制位

采样时钟边沿（**CSCKE**）控制位决定 **CSCK** 信号的采样边沿。如果 **CSCK** 位清零（默认），将在 **CSCK** 信号的下降沿采样数据。**AC-Link** 协议和大部分多通道格式都要求在 **CSCK** 信号的下降沿采样数据。如果 **CSCK** 位置 1，将在 **CSCK** 的上升沿采样数据。**I²S** 协议要求在 **CSCK** 信号的上升沿采样数据。

18.3.9 数据对齐控制位

在大多数应用中，在 **COFS** 信号采样为高时，将在一个 **CSCK** 周期后开始数据传输。这是 **DCI** 模块的默认配置。在 **DCICON1 SFR** 中把 **DJST** 控制位置 1，可以选择另一种数据对齐方式。当 **DJST = 1** 时，如果 **COFS** 信号采样为高，将在当前 **CSCK** 周期中开始数据传输。

18.3.10 发送时隙使能位

TSCON SFR 中有可用来使能发送时隙的控制位，最多 16 个发送时隙。这些控制位是 **TSE<15:0>** 位。每个时隙的大小由 **WS<3:0>** 字长选择位决定，最大为 16 位。

如果通过某个 **TSE** 位 (**TSE_x = 1**) 使能了发送时隙，当前发送影子缓冲器单元中的内容就会被装入 **CSDO** 移位寄存器，同时 **DCI** 缓冲器控制单元将递增以指向下一个单元。

在未用的发送时隙中，**CSDO** 引脚将驱动为 0，或者在所有被禁止的时隙中呈现三态，这取决于 **DCICON1 SFR** 中 **CSDOM** 位的状态。

数据帧长度（按位计）由所选的数据字长、帧中数据字的个数决定。如果帧中数据字的个数少于 16，则多出的时隙使能位没有作用。

每个发送数据字以左对齐的方式写入 16 位发送缓冲器中。如果所选字长小于 16 位，发送缓冲存储区的 **Lsb** 对发送数据没有作用。用户应该向每个发送缓冲器单元中未使用的 **Lsb** 写入 0。

18.3.11 接收时隙使能位

RSCON SFR 中有可用来使能接收时隙的控制位，最多 16 个接收时隙。这些控制位是 **RSE<15:0>** 位。每个接收时隙的大小由 **WS<3:0>** 字长选择位决定，大小可以从 1 位到 16 位。

如果通过某个 **RSE** 位 (**RSE_x = 1**) 使能了接收时隙，移位寄存器内容将被写入当前 **DCI** 接收影子缓冲器单元中，同时缓冲器控制单元将递增以指向下一个缓冲器单元。

如果所选的字长小于 16 位，数据在接收存储器缓冲单元中不被打包。每个接收时隙数据字都存储在一个单独的 16 位缓冲器单元中。数据总是以左对齐的格式存储在接收存储缓冲器中。

18.3.12 帧同步时隙使能位的工作

TSE 和 **RSE** 控制位与 **DCI** 帧同步发生器协同工作。在主模式下，一旦帧同步发生器复位，就将产生 **COFS** 信号。在从模式下，一旦接收到 **COFS** 脉冲，帧同步发生器就将复位。

TSE 和 **RSE** 控制位允许使能最多 16 个连续的发送或接收时隙。在最后一个使能时隙发送 / 接收时，**DCI** 将停止缓冲数据，一直到下一次 **COFS** 脉冲产生。

18.3.13 同步数据传输

一旦使能了给定的发送或接收时隙，**DCI** 缓冲器控制单元将递增一个字单元。大多数情况下，数据输入和输出传输是同步的，这就是说，对于给定通道，数据采样的接收与发送同时进行。于是，当产生 **DCI** 中断时，发送和接收缓冲器中将填入相同数量的数据。

在一些情况下，一个数据帧内发送、接收的数据量可能不相等。例如，假设使用的是双字数据帧。此外，假设仅在时隙 0 中接收数据，但在时隙 0 和时隙 1 中发送数据。在这种情况下，在数据帧中，缓冲器控制单元计数器将递增两次，但只有一个接收寄存器单元将填入数据。

18.3.14 缓冲器长度控制

两次中断之间缓冲的数据量，由 DCICON1 SFR 中的缓冲器长度（BLEN<1:0>）控制位决定。使用 BLEN 控制位，可以设置发送和接收缓冲器的长度为 1 至 4 个数据字。BLEN 控制位将与 DCI 缓冲器控制单元地址计数器的当前值进行比较。当 DCI 地址计数器的最低 2 位与 BLEN<1:0> 的值匹配时，缓冲器控制单元将复位为 0。此外，接收影子寄存器的内容将传递到接收缓冲寄存器中，发送缓冲寄存器的内容将传递到发送影子寄存器中。

18.3.15 缓冲器与数据帧的对齐

AGU 地址指针位置与数据帧边界之间，没有直接的耦合关系。这意味着，每个发送和接收缓冲器存在隐含的分配，这与 BLEN 控制位和使能的时隙数量（使能的时隙数量通过 TSE 和 RSE 控制位来设置）有关。

例如，假设选择了 4 字长的数据帧，我们打算在帧中传输全部 4 个时隙。通过设置 TSCON SFR 中的 TSE0、TSE1、TSE2 和 TSE3 控制位，可以得到这个配置。这样设置好模块后，TXBUFO 寄存器很自然地就会被分配给时隙 0，TXBUF1 寄存器很自然地就会被分配给时隙 1，依此类推。

注：如果在一个数据帧中，有多于 4 个时隙有效，用户代码必须跟踪每个中断时读 / 写了哪些时隙。在一些情况下，发送 / 接收缓冲器与对应时隙分配之间的对齐可能会丢失。仿真断点或硬件陷阱就是这样的例子。在这些情况下，用户应该查询 SLOT 状态位，决定在缓冲寄存器中装入什么数据，从而使软件与 DCI 模块重新同步。

18.3.16 发送状态位

在 DCISTAT SFR 中有两个发送状态位。

当发送缓冲寄存器内容被传递到发送影子寄存器中时，TMPTY 位被置 1。在软件中可以查询 TMPTY 位，决定何时可以写发送缓冲寄存器。如果对 4 个发送缓冲器之一进行了写入，硬件将自动清零 TMPTY 位。

TUNF 位是只读的，它表明正在使用的发送缓冲寄存器中至少有一个发生了下溢。发送缓冲寄存器内容被传递到发送影子寄存器中时，TUNF 位被置 1。当 CPU 对下溢的缓冲寄存器进行了写入时，将自动清零 TUNF 状态位。

注：发送状态位仅用来指明模块所使用缓冲器单元的状态。例如，如果设置的缓冲器长度小于 4 个字，则未使用的缓冲器单元将不影响发送状态位。

18.3.17 接收状态位

在 DCISTAT SFR 中有两个接收状态位。

RFUL 状态位是只读的，它表明在接收缓冲器中是否有新数据。当 CPU 读取正在使用的全部接收缓冲器后，RFUL 位自动清零。

ROV 状态位是只读的，它表明接收缓冲器单元中至少有一个产生了接收溢出。如果在新数据从影子寄存器传递到缓冲器单元之前，CPU 没有读取缓冲器单元，就将产生接收溢出。如果 CPU 读取了造成溢出的缓冲寄存器，将自动清零 ROV 状态位。

对于特定的缓冲器单元，当发生接收溢出时，缓冲器中的原有内容将被改写。

注：接收状态位仅用来指明模块所使用缓冲器单元的状态。例如，如果设置的缓冲器长度小于 4 个字，则未使用的缓冲器单元将不影响接收状态位。

18.3.18 SLOT 状态位

DCISTAT SFR 中的 SLOT_{<3:0>} 状态位，指明当前有效时隙。这些位与帧同步发生器计数器的值相对应。当产生 DCI 中断时，用户可以在软件中查询这些状态位，从而确定哪个时隙数据是最后接收到的以及哪个时隙数据应该被装入 TXBUF 寄存器。

18.3.19 CSDO 模式位

CSDOM 控制位控制 CSDO 引脚在未使用发送时隙中的状态。如果某个发送时隙在 TSCON SFR 中相应的 TSE_x 位被清零，此发送时隙就是未使用的。

如果 CSDOM 位被清零（默认），在未使用时隙周期中，CSDO 引脚将为低电平。此模式用在仅有两个器件连接到串行总线时。

如果 CSDOM 位被置 1，CSDO 引脚在未使用的时隙周期中将呈现三态。在多通道应用中，此模式允许多个器件共用同一根 CSDO 线。此时应将 CSDO 线上的每个器件配置为只在特定时隙内发送数据，不允许有两个器件在同一时隙时发送数据。

18.3.20 数字环回模式

通过置 1 DCICON1 SFR 中的 DLOOP 控制位，使能数字环回模式。当 DLOOP 位置 1 时，模块在内部将 CSDO 信号连接到 CSDI。在数字环回模式下，将忽略 CSDI I/O 引脚上实际输入的数据。

18.3.21 下溢模式控制位

当发生下溢时，根据 DCICON1 SFR 中下溢模式（UNFM）控制位的状态，模块会执行两种操作之一。如果 UNFM 位清零（默认），在缓冲器单元有效时隙期间，模块将在 CSDO 引脚上发送 0。在此工作模式下，对于连接到 DCI 模块的编解码器件，输出给它的将仅是数字“静音”。如果 UNFM 控制位置 1，模块将发送最后写入缓冲器单元的数据。该工作模式允许用户发送连续的数据值到编解码器件，而无需 CPU 开销。

18.4 DCI 模块中断

DCI 模块中断的频率取决于 DCICON2 SFR 中的 BLEN_{<1:0>} 控制位。每当达到设定的缓冲器长度且发生影子寄存器数据传送时，就会产生中断传递给 CPU。影子寄存器传送的定义是：上一次写入 TXBUF 的值被传送到发送影子寄存器，而接收影子寄存器中新接收的值被传送到 RXBUF 寄存器。

18.5 CPU 休眠和空闲模式期间 DCI 模块的工作

18.5.1 CPU 休眠模式期间 DCI 模块的工作

当 CSCK 信号由外部器件提供（CSCKD = 1）时，DCI 模块在 CPU 处于休眠模式时能够工作并唤醒 CPU。当 DCI 缓冲器传送完成且 CPU 处于休眠模式时，DCI 模块将产生一个异步中断。

18.5.2 CPU 空闲模式期间 DCI 模块的工作

如果 DCISIDL 控制位清零（默认），则模块将在空闲模式下继续正常工作。如果 DCISIDL 位置 1，当 CPU 进入空闲模式时，模块将停止工作。

18.6 AC-Link 模式的工作

AC-Link 协议数据帧是 256 位的，它可以被分成一个 16 位数据间隙和其后的 12 个 20 位数据间隙。DCI 模块有两种针对 AC-Link 协议的工作模式。使用 DCICON1 SFR 中的 COFSM_{<1:0>} 控制位来选择这两种工作模式。第一种 AC-Link 模式称为“16 位 AC-Link 模式”，通过设置 COFSM_{<1:0>} = 10 来选择。第二种 AC-Link 模式称为“20 位 AC-Link 模式”，通过设置 COFSM_{<1:0>} = 11 来选择。

18.6.1 16 位 AC-LINK 模式

在 16 位 AC-Link 模式下，数据字长度被限制为 16 位。注意此限制仅影响 AC-Link 协议的 20 位数据时隙。对于接收时隙，输入数据将被简单地截取为 16 位。对于输出时隙，模块将数据字的最低 4 位置为 0。时隙的这种截取，使得 A/D 和 DAC 数据被限制为 16 位，但允许 TXBUF 和 RXBUF 寄存器中恰当的数据对齐。每个 RXBUF 和 TXBUF 寄存器将存放一个数据时隙值。

18.6.2 20 位 AC-LINK 模式

20 位 AC-Link 模式允许发送和接收数据时隙中的所有位，但不会维护 TXBUF 和 RXBUF 寄存器中的数据对齐。

除了帧同步信号的占空比不同之外，20 位 AC-Link 模式的工作方式类似于 DCI 模块的多通道模式。AC-Link 帧同步信号应该保持 16 个 CSCK 周期的高电平，并在接着的 240 个周期中保持低电平。

20 位模式把每个 256 位 AC-Link 帧当作 16 个 16 位时隙处理。在 20 位 AC-Link 模式下，模块就好像是在 COFSG<3:0> = 1111 且 WS<3:0> = 1111 的条件下工作。20 位数据间隙的数据对齐被忽略。例如，通过将 TSCON 和 RSCON SFR 中的所有位置 1，整个 AC-Link 数据帧能够以打包的方式发送和接收。由于可用缓冲器的总长度为 64 位，所以传输该 AC-Link 帧需要经过 4 次连续的中断。应用软件必须跟踪当前 AC-Link 帧段。

18.7 I²S 模式的工作

向 DCICON1 SFR 中的 COFSM<1:0> 控制位写入 01，就将把 DCI 模块配置为 I²S 模式。工作在 I²S 模式下时，DCI 模块将产生占空比为 50% 的帧同步信号。帧同步信号的每个边沿，标记着一个新数据字传输的边界。

用户必须选择帧长度和数据字大小，使用 DCICON2 SFR 中的 COFSG 和 WS 控制位来进行选择。

18.7.1 I²S 帧和数据字长选择

设置 WS 和 COFSG 控制位，以便产生 I²S 数据帧长一半的周期。即，帧长是左对齐或右对齐数据字传输所需的总的 CSCK 周期数。

必须设置 BLEN 位，从而得到所需的缓冲器长度。设置 BLEN<1:0> = 01 将产生 CPU 中断，每 I²S 帧产生一次中断。

18.7.2 I²S 数据对齐

根据 I²S 规范，在默认情况下，将在 WS 信号跳变后过一个 CSCK 周期开始数据字的传输。可以使用 DCICON1 SFR 中的 DJST 控制位选择“MSb 左对齐”选项。

如果 DJST = 1，则 I²S 数据传输是 MSb 左对齐的。在 COFS 信号的上升或下降沿的 CSCK 周期期间，数据字的 MSb 将出现在 CSDO 引脚上。在数据字发送后，CSDO 引脚为三态。

表 18-2: DCI 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
DCICON1	0240	DCIEN	—	DCISIDL	—	DLOOP	CSCKD	CSCKE	COFSID	UNFM	CSDOM	DJST	—	—	—	COFSM1	COFSM0	0000 0000 0000 0000
DCICON2	0242	—	—	—	—	BLEN1	BLEN0	—	COFSG<3:0>			—	WS<3:0>			0000 0000 0000 0000		
DCICON3	0244	—	—	—	—	BCG<11:0>												0000 0000 0000 0000
DCISTAT	0246	—	—	—	—	SLOT3	SLOT2	SLOT1	SLOT0	—	—	—	—	ROV	RFUL	TUNF	TMPTY	0000 0000 0000 0000
TSCON	0248	TSE15	TSE14	TSE13	TSE12	TSE11	TSE10	TSE9	TSE8	TSE7	TSE6	TSE5	TSE4	TSE3	TSE2	TSE1	TSE0	0000 0000 0000 0000
RSCON	024C	RSE15	RSE14	RSE13	RSE12	RSE11	RSE10	RSE9	RSE8	RSE7	RSE6	RSE5	RSE4	RSE3	RSE2	RSE1	RSE0	0000 0000 0000 0000
RXBUFO	0250	接收缓冲器 #0 数据寄存器															0000 0000 0000 0000	
RXBUF1	0252	接收缓冲器 #1 数据寄存器															0000 0000 0000 0000	
RXBUF2	0254	接收缓冲器 #2 数据寄存器															0000 0000 0000 0000	
RXBUF3	0256	接收缓冲器 #3 数据寄存器															0000 0000 0000 0000	
TXBUFO	0258	发送缓冲器 #0 数据寄存器															0000 0000 0000 0000	
TXBUF1	025A	发送缓冲器 #1 数据寄存器															0000 0000 0000 0000	
TXBUF2	025C	发送缓冲器 #2 数据寄存器															0000 0000 0000 0000	
TXBUF3	025E	发送缓冲器 #3 数据寄存器															0000 0000 0000 0000	

图注: u = 未初始化位

注: 有关寄存器各位的说明, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

19.0 12 位模数转换器 (ADC) 模块

注：本数据手册总结了 dsPIC30F 系列器件的功能，但是不应把本手册当作无所不包的参考手册来使用。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)。

12 位模数转换器可将模拟输入信号转换为 12 位数字值。该模块基于逐次逼近寄存器 (Successive Approximation Register, SAR) 结构，可实现最大为 200 kspS 的采样速率。ADC 模块拥有最多 16 路模拟输入通道，经过多路开关连接到 1 个采样 / 保持放大器。采样 / 保持电路的输出输入到转换器，经转换器转换后得到数字结果。可通过软件选择模拟参考电压为器件供电电压 (AVDD/AVSS) 或 (VREF+/VREF-) 引脚上的电压。ADC 具有一个独特的功能啊；当选择 RC 振荡器时，即使器件处于休眠模式，ADC 仍能工作。

ADC 模块具有 6 个 16 位寄存器：

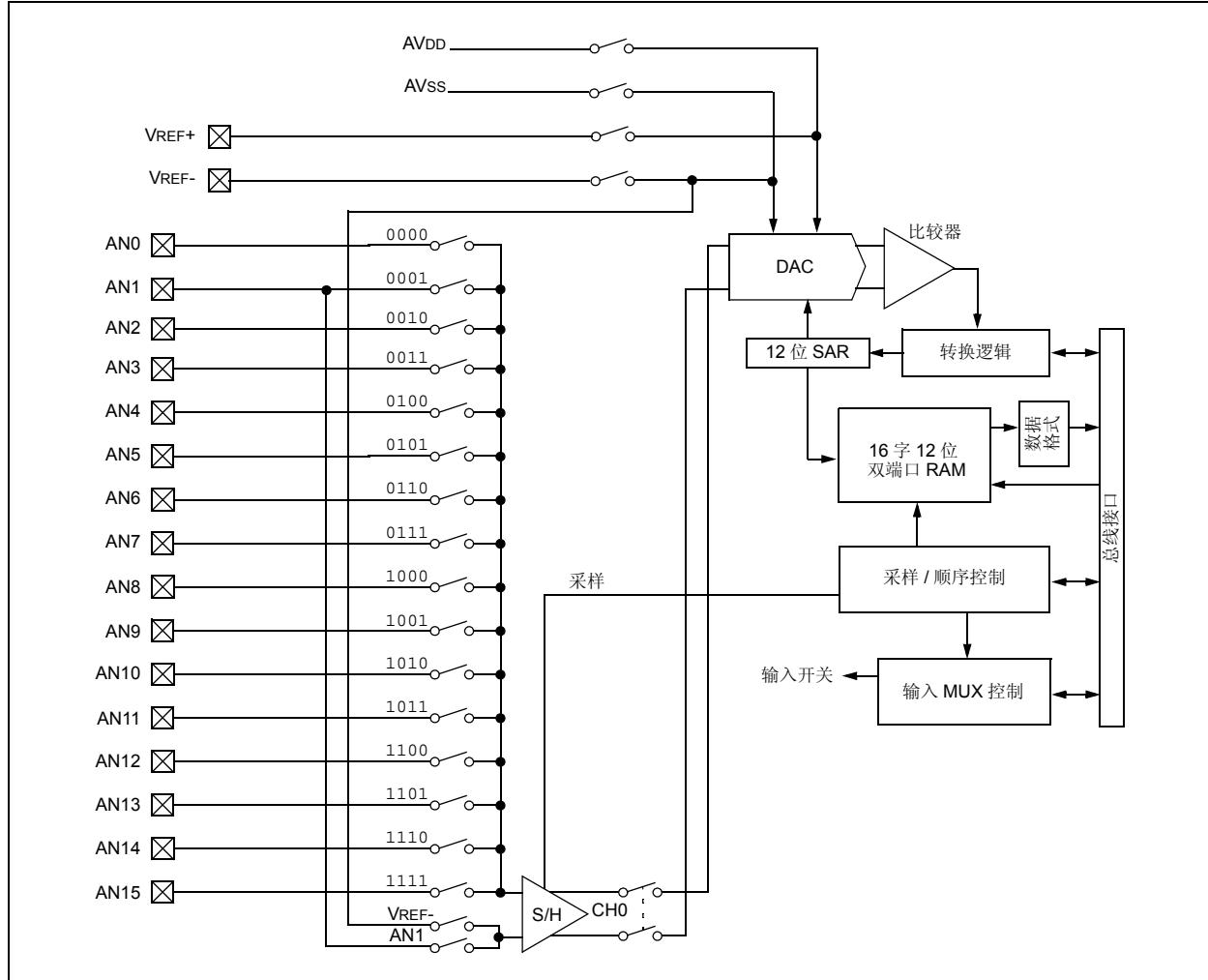
- ADC 控制寄存器 1 (ADCON1)
- ADC 控制寄存器 2 (ADCON2)
- ADC 控制寄存器 3 (ADCON3)
- ADC 输入选择寄存器 (ADCHS)
- ADC 端口配置寄存器 (ADPCFG)
- ADC 输入扫描选择寄存器 (ADCSSL)

ADCON1、ADCON2 和 ADCON3 寄存器控制 ADC 模块的工作。ADCHS 寄存器选择要转换的输入通道。ADPCFG 寄存器用于将端口引脚配置为模拟输入或数字 I/O。ADCSSL 寄存器用于选择要扫描的输入通道。

注：当 ADON = 1 时，不应对 SSRC<2:0>、ASAM、SMPI<3:0>、BUFM 和 ALTS 位以及 ADCON3 和 ADCSSL 寄存器进行写入操作，否则将导致不确定的结果。

图 19-1 给出了 12 位 ADC 模块的框图。

图 19-1： 12 位 ADC 功能框图



19.1 ADC 结果缓冲器

ADC 模块包含一个称为 ADCBUF0...ADCBUFF 的 16 字双端口只读缓冲器，对 ADC 结果进行缓冲。RAM 为 12 位宽，但其存储的数据可为 4 种不同格式的 16 位数据。用户软件不能对 16 个 ADC 结果缓冲寄存器 (ADCBUF0 到 ADCBUFF) 的内容进行写入操作。

19.2 转换操作

当 ADC 模块配置完毕后，把 SAMP 位置 1 就将启动采样操作。可编程位、定时器超时和外部事件等触发源将终止采集并启动转换。当 A/D 转换完成时，结果将被装入 ADCBUF0 至 ADCBUFF，且在 SMPI 位指定的采样数之后 DONE 位和 A/D 中断标志位 ADIF 将被置 1。如第 19.3 节“选择转换顺序”所述，ADC 模块可被配置为以不同的频率中断。

进行 A/D 转换时，应遵循以下步骤：

1. 配置 ADC 模块：
 - 配置模拟引脚、参考电压和数字 I/O
 - 选择 ADC 输入通道
 - 选择 ADC 转换时钟
 - 选择 ADC 转换触发源
 - 开启 ADC 模块
2. 配置 ADC 中断（如需要）：
 - 清零 ADIF 位
 - 选择 ADC 中断优先级
3. 启动采样。
4. 等待所需的采集时间。
5. 触发采集结束，启动转换：
6. 等待 ADC 转换完成，可通过以下方式实现：
 - 等待 ADC 中断，或
 - 等待 DONE 位被置 1。
7. 读 ADC 结果缓冲器，需要的话清零 ADIF。

19.3 选择转换顺序

几组控制位可用来对 ADC 将输入连接到采样 / 保持通道、转换通道、写入缓冲存储区和产生中断的过程进行选择控制。

该过程将由采样时钟进行控制。

SMPI 位用于选择在中断发生前进行采集/转换的次数。该数值可为每次中断进行采样 1 到 16 次。

当置 1 时，BUFM 位将 16 字结果缓冲器分为两个 8 字缓冲器。每个中断事件将交替写入这两个 8 字缓冲器。

BUFM 位的使用取决于中断之后有多少时间将数据从缓冲器内移出。

如果处理器可在对一个通道进行采集和转换的时间内快速卸空一个满的缓冲器，则 BUFM 位可为 0 且每次中断可进行最多 16 次转换（对应于 16 路输入通道）。处理器在一次采样和转换的时间内移出 16 个转换结果。

如果在采集和转换时间内处理器不能完全卸空一个满的缓冲器，BUFM 位应为 1。例如，如果 SMPI<3:0> (ADCON2<5:2>) = 0111，则先将 8 个转换结果装入其中一个 8 字缓冲器，此后将产生中断。再将另外的 8 个转换结果装入另一个 8 字缓冲器。处理器将利用相邻两次中断之间的全部时间来移出 8 个转换结果。

ALTS 位用来在采样过程中轮换选择的输入。输入多路开关具有两组采样输入：MUX A 和 MUX B。如果 ALTS 位为 0，只选择 MUX A 输入进行采样。如果 ALTS 位为 1 且 SMPI<3:0> = 0000，在第一个采样/转换过程将选择 MUX A 输入通道，而在下一个采样/转换过程将选择 MUX B 输入通道。

CSCNA 位 (ADCON2<10>) 允许对多路开关输入 MUX A 组的选定数量的模拟输入进行顺序扫描。ADCSSL 寄存器选择要扫描的输入。如果 ADCSSL 寄存器中的某个位为 1，则将选择相应的输入。将在每次中断之后，始终按照输入编号从低到高的顺序对输入进行扫描。如果选择的输入数目大于每次中断进行的采样数，则编号高于采样数的输入将不会被使用。

19.4 转换触发的编程设定

转换触发将终止数据采集并开始进行要求的转换。

SSRC<2:0> 位用于选择转换触发源。 **SSRC** 位提供了多达 4 个备用的转换触发源。

当 **SSRC<2:0> = 000** 时，转换触发由软件进行控制。清零 **SAMP** 位将导致触发转换。

当 **SSRC<2:0> = 111**（自动启动模式）时，转换触发由 A/D 时钟进行控制。**SAMC** 位用于选择采样开始到转换开始之间的 ADC 时钟数。这将提供最快的多通道转换速率。**SAMC** 必须至少为 1 个时钟周期。

其他触发源可能来自定时器模块或外部中断。

19.5 中止转换

转换过程中清零 **ADON** 位将中止当前的转换并停止采样过程，直到下一个采样触发为止。不会用部分完成的 A/D 转换采样来更新 **ADCBUF**。即，**ADCBUF** 将继续保留上一次转换完成后的值（或上一次写入 **ADCBUF** 寄存器的值）。

如果清零 **ADON** 位与自动启动同时发生，则清零 **ADON** 位具有更高的优先级，且不会启动新的转换。

19.6 选择 ADC 转换时钟

ADC 转换需要 14 个 TAD。通过软件方式使用 6 位计数器可选择 ADC 转换的时钟源。TAD 存在 64 种可能的选择。

公式 19-1: ADC 转换时钟

$$TAD = TCY * (0.5 * (ADCS<5:0> + 1))$$

通过将 **ADRC** 位置 1 可选择内部 RC 振荡器。

为使 ADC 转换正确，必须选择合适的 ADC 转换时钟 (TAD) 以确保最小 334 ns（当 VDD = 5V 时）的 TAD 时间。其他工作条件下的最小 TAD，请参见“电气规范”章节。

例 19-1 给出了 **ADCS<5:0>** 位的计算示例，假设器件工作速度为 30 MIPS。

例 19-1: ADC 转换时钟和采样速率计算

$$\begin{aligned} \text{最小 } TAD &= 334 \text{ ns} \\ TCY &= 33.33 \text{ ns (30 MIPS)} \end{aligned}$$

$$\begin{aligned} ADCS<5:0> &= 2 \frac{TAD}{TCY} - 1 \\ &= 2 \cdot \frac{334 \text{ ns}}{33.33 \text{ ns}} - 1 \\ &= 19 \end{aligned}$$

因此，

$$\text{设置 } ADCS<5:0> = 19$$

$$\begin{aligned} \text{实际 } TAD &= \frac{TCY}{2} (ADCS<5:0> + 1) \\ &= \frac{33.33 \text{ ns}}{2} (19 + 1) \\ &= 334 \text{ ns} \end{aligned}$$

如果 **SSRC<2:0> = 111** 且 **SAMC<4:0> = 00001**

因为，

$$\begin{aligned} \text{采样时间} &= \text{采集时间} + \text{转换时间} \\ &= 1 \text{ TAD} + 14 \text{ TAD} \\ &= 15 \times 334 \text{ ns} \end{aligned}$$

因此，

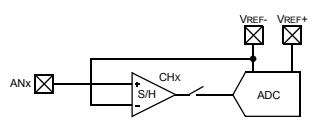
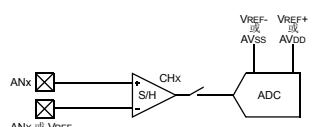
$$\begin{aligned} \text{采样速率} &= \frac{1}{(15 \times 334 \text{ ns})} \\ &= \sim 200 \text{ kHz} \end{aligned}$$

dsPIC30F5011/5013

19.7 ADC 速度

dsPIC30F 的 12 位 ADC 规范允许最大 200 kspS 的采样速率。下表汇总了 dsPIC30F 12 位 ADC 的转换速度及其要求的工作条件。

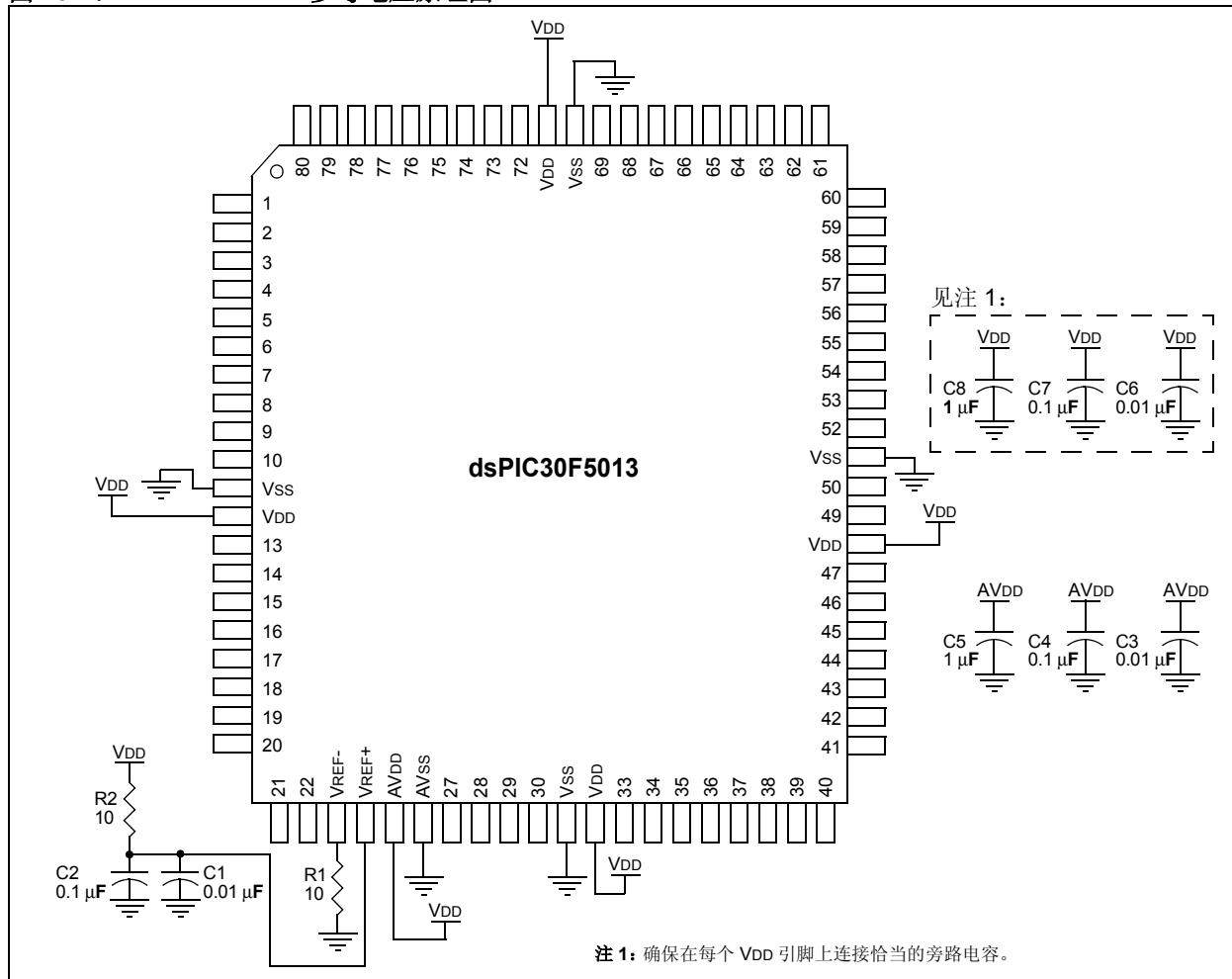
表 19-1：12 位 ADC 扩展转换速率

dsPIC30F 12 位 ADC 转换速率						
速度	TAD 最小值	采样时间 最小值	R _s 最大值	V _{DD}	温度	通道配置
最大 200 kspS ⁽¹⁾	334 ns	1 TAD	2.5 kΩ	4.5V 至 5.5V	-40°C 至 +85°C	
最大 100 kspS	668 ns	1 TAD	2.5 kΩ	3.0V 至 5.5V	-40°C 至 +125°C	

注 1：为确保正常工作，必须使用外部 V_{REF}- 和 V_{REF}+ 引脚。推荐电路请参见图 19-2。

下图给出了转换速率高于 100 kspS 时的推荐电路。以 dsPIC30F5013 为例。

图 19-2: ADC 参考电压原理图



以下配置指导给出了转换速度高于 100 kspS 时要求的设置值。

19.7.1 200 KSPS 时的配置指南

为获得 200 kspS 的转换速率，需要进行如下配置。

- 符合表 19-2 中提供的条件。
- 按照图 19-2 所示的推荐电路，连接外部 VREF+ 和 VREF- 引脚。
- 设置 ADCON1 寄存器中的 SSRC<2.0> = 111，使能自动转换选项。
- 把 ADCON1 寄存器中的 ASAM 控制位置 1，使能自动采样。
- 写 ADCON2 寄存器中的 SMPI<3.0> 控制位，设置两次中断之间所需的转换次数。

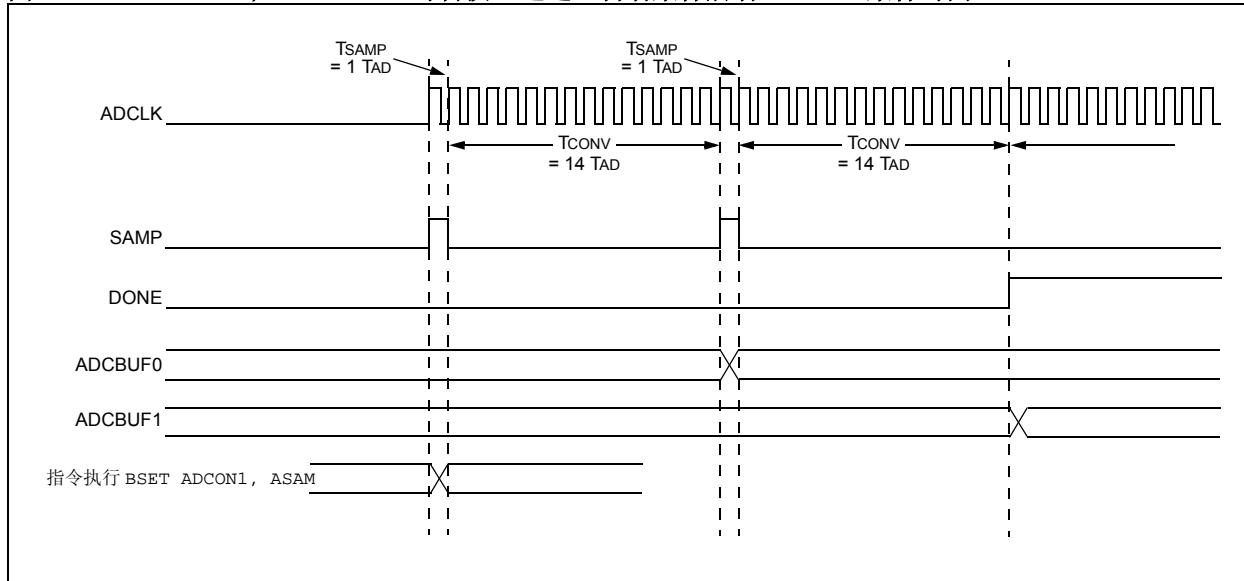
- 通过写 ADCON3 寄存器中的 ADCS<5:0> 控制位，将 ADC 时钟周期配置为：

$$\frac{1}{(14 + 1) \times 200,000} = 334 \text{ ns}$$

- 通过写 SAMC<4:0> = 00001，将采样时间配置为 1 TAD。

下图给出了 ADC 运行在 200 kspS 时的时序图。按照上述指南选择的 TAD 允许转换速度达到 200 kspS。例 19-1 给出了代码示例。

图 19-3: 在 200 KSPS 时转换 1 通道, 自动采样启动, 1 TAD 采样时间



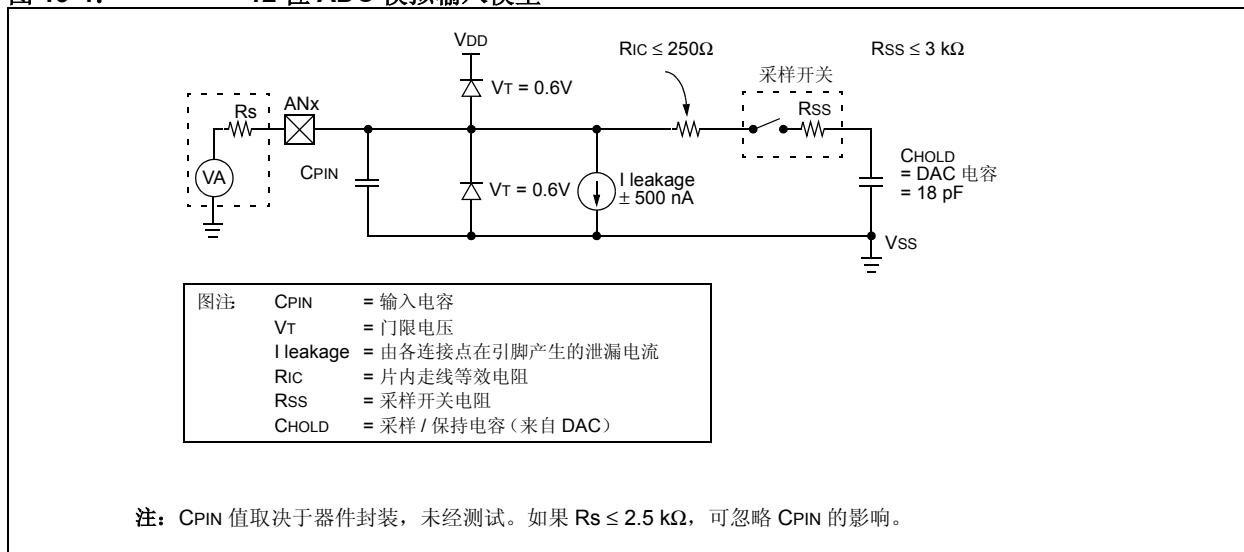
19.8 ADC 采集要求

图 19-4 给出了 12 位 ADC 的模拟输入模型。ADC 的总采样时间是内部放大器稳定时间和保持电容充电时间的函数。

为了使 ADC 达到规定的精度, 必须让充电保持电容 (CHOLD) 充分充电至模拟输入引脚上的电平。信号源阻抗 (R_s)、片内走线等效电阻 (R_{IC}) 和内部采样开

关阻抗 (R_{SS}) 共同地直接影响电容 CHOLD 充电所需的时间。所以模拟信号源的总阻抗应足够小, 以便在选择的采样时间内对保持电容充分充电。为了将引脚泄漏电流对 ADC 精度的影响降到最低, 建议使用的大信号源阻抗 R_s 为 $2.5 \text{ k}\Omega$ 。选择 (改变) 了模拟输入通道后, 采样工作必须在启动转换前完成。在每次采样操作前, 内部保持电容将处于放电状态。

图 19-4: 12 位 ADC 模拟输入模型



19.9 模块掉电模式

模块具有两种内部功耗模式。

当 ADON 位为 1 时，模块处于工作模式，模块处于完全供电和工作状态。

当 ADON 为 0 时，模块处于关闭模式。电路的数字和模拟部分将被禁止以最大程度地省电。

为了从关闭模式返回到工作模式，用户必须等待 ADC 电路进入稳定状态。

19.10 CPU 休眠和空闲模式期间的 ADC 工作

19.10.1 CPU 休眠模式期间的 ADC 工作

当器件进入休眠模式时，模块的所有时钟源被关闭并保持为逻辑 0。

如果在转换过程中进入休眠状态，转换将中止。从休眠模式退出时，转换器将不会继续进行部分完成的转换。

器件进入或退出休眠模式将不会影响寄存器的内容。

如果将 ADC 时钟源设置为 RC (ADRC = 1)，则 ADC 模块可以在休眠模式下工作。当选择 RC 时钟源时，ADC 模块在启动转换之前要先等待一个指令周期。其间可执行 SLEEP 指令，这将消除转换过程中所有的数字开关噪声。当转换完成后，DONE 位将置 1 且转换结果将被装入 ADCBUF 寄存器。

如果允许 ADC 中断，则器件将从休眠状态唤醒。如果未允许 A/D 中断，尽管 ADON 位保持置 1，ADC 模块也将被关闭。

19.10.2 CPU 空闲模式期间的 A/D 工作

ADSIDL 位选择 ADC 模块在空闲模式下是停止还是继续工作。如果 ADSIDL = 0，则模块将在空闲模式时继续工作。如果 ADSIDL = 1，则模块将在空闲模式下停止工作。

19.11 复位的影响

器件复位将强制所有寄存器为复位状态。这将迫使 ADC 模块关闭并中止任何正在进行的转换和采样过程。ADCBUF 寄存器中的值不会被修改。上电复位后，ADC 结果寄存器将包含未知数据。

19.12 输出格式

ADC 结果为 12 位宽，数据缓冲器 RAM 也是 12 位宽。12 位数据可以用以下四种格式之一读取。FORM<1:0> 位用于选择格式。每一种输出格式将数据转换为数据总线上的 16 位结果。

图 19-5: ADC 输出数据格式

RAM 内容:	<table border="1"><tr><td>d11</td><td>d10</td><td>d09</td><td>d08</td><td>d07</td><td>d06</td><td>d05</td><td>d04</td><td>d03</td><td>d02</td><td>d01</td><td>d00</td></tr></table>	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00				
d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00						
读至总线:																	
有符号小数	<table border="1"><tr><td>d11</td><td>d10</td><td>d09</td><td>d08</td><td>d07</td><td>d06</td><td>d05</td><td>d04</td><td>d03</td><td>d02</td><td>d01</td><td>d00</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0
d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0		
小数	<table border="1"><tr><td>d11</td><td>d10</td><td>d09</td><td>d08</td><td>d07</td><td>d06</td><td>d05</td><td>d04</td><td>d03</td><td>d02</td><td>d01</td><td>d00</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0
d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0		
有符号整数	<table border="1"><tr><td>d11</td><td>d11</td><td>d11</td><td>d11</td><td>d11</td><td>d10</td><td>d09</td><td>d08</td><td>d07</td><td>d06</td><td>d05</td><td>d04</td><td>d03</td><td>d02</td><td>d01</td><td>d00</td></tr></table>	d11	d11	d11	d11	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
d11	d11	d11	d11	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00		
整数	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>d11</td><td>d10</td><td>d09</td><td>d08</td><td>d07</td><td>d06</td><td>d05</td><td>d04</td><td>d03</td><td>d02</td><td>d01</td><td>d00</td></tr></table>	0	0	0	0	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
0	0	0	0	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00		

19.13 配置模拟端口引脚

ADPCFG 和 TRIS 寄存器用于控制 ADC 端口引脚的操作。如果要将端口引脚用作模拟输入，则对应的 TRIS 位必须置 1（输入）。如果将 TRIS 位清零（输出），则数字输出电平（VOH 或 VOL）将被转换。

ADC 操作与 CH0SA<3:0>/CH0SB<3:0> 位和 TRIS 位的状态无关。

在读取端口寄存器时，所有配置为模拟输入通道的引脚均读为零。

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

19.14 连接注意事项

模拟输入在 VDD 和 Vss 之间连接有二极管进行 ESD 保护。这就要求模拟输入电压必须介于 VDD 和 Vss 之间。如果输入电压超出此范围 0.3V 以上（任一方向上），就会有一个二极管正向偏置，而且如果超过输入电流规范可能会损坏器件。

有时可外接 RC 滤波器来对输入信号进行抗混叠滤波。R 元件的选择要确保满足采样时间要求。任何通过高阻抗连接到模拟输入引脚上的外部元件（如电容和齐纳二极管等）在引脚上的泄漏电流都应极小。

表 19-2: A/D 转换器寄存器映射

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
ADCBUF0	0280	—	—	—	—	ADC 数据缓冲器 0										0000 uuuu uuuu uuuu		
ADCBUF1	0282	—	—	—	—	ADC 数据缓冲器 1										0000 uuuu uuuu uuuu		
ADCBUF2	0284	—	—	—	—	ADC 数据缓冲器 2										0000 uuuu uuuu uuuu		
ADCBUF3	0286	—	—	—	—	ADC 数据缓冲器 3										0000 uuuu uuuu uuuu		
ADCBUF4	0288	—	—	—	—	ADC 数据缓冲器 4										0000 uuuu uuuu uuuu		
ADCBUF5	028A	—	—	—	—	ADC 数据缓冲器 5										0000 uuuu uuuu uuuu		
ADCBUF6	028C	—	—	—	—	ADC 数据缓冲器 6										0000 uuuu uuuu uuuu		
ADCBUF7	028E	—	—	—	—	ADC 数据缓冲器 7										0000 uuuu uuuu uuuu		
ADCBUF8	0290	—	—	—	—	ADC 数据缓冲器 8										0000 uuuu uuuu uuuu		
ADCBUF9	0292	—	—	—	—	ADC 数据缓冲器 9										0000 uuuu uuuu uuuu		
ADCBUFA	0294	—	—	—	—	ADC 数据缓冲器 10										0000 uuuu uuuu uuuu		
ADCBUFB	0296	—	—	—	—	ADC 数据缓冲器 11										0000 uuuu uuuu uuuu		
ADCBUFC	0298	—	—	—	—	ADC 数据缓冲器 12										0000 uuuu uuuu uuuu		
ADCBUFD	029A	—	—	—	—	ADC 数据缓冲器 13										0000 uuuu uuuu uuuu		
ADCBUFE	029C	—	—	—	—	ADC 数据缓冲器 14										0000 uuuu uuuu uuuu		
ADCBUFF	029E	—	—	—	—	ADC 数据缓冲器 15										0000 uuuu uuuu uuuu		
ADCON1	02A0	ADON	—	ADSIDL	—	—	—	FORM<1:0>		SSRC<2:0>			—	—	ASAM	SAMP	DONE	0000 0000 0000 0000
ADCON2	02A2	VCFG<2:0>			—	—	CSCNA	—	—	BUFS	—	SMPI<3:0>			BUFM	ALTS	0000 0000 0000 0000	
ADCON3	02A4	—	—	—	SAMC<4:0>				ADRC	—	ADCS<5:0>						0000 0000 0000 0000	
ADCHS	02A6	—	—	—	CH0NB	CH0SB<3:0>				—	—	—	CH0NA	CH0SA<3:0>			0000 0000 0000 0000	
ADPCFG	02A8	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000 0000 0000
ADCSSL	02AA	CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000 0000 0000 0000

图注: u = 未初始化位

注: 有关寄存器各位的说明, 请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

dsPIC30F5011/5013

注：

20.0 系统集成

注: 本数据手册总结了 dsPIC30F 系列器件的功能,但是不应把本手册当作无所不包的参考手册来使用。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)。

下列系统集成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)
 - 可编程欠压复位 (BOR)
- 看门狗定时器 (WDT)
- 低电压检测
- 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)

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需的延时。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。有了这两个片上定时器,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,用户应用可将器件从休眠状态唤醒。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 停止工作。RC 振荡器可降低系统成本,而 LP 晶振功能可降低功耗。

20.1 振荡器系统概述

dsPIC30F 振荡器系统包含以下功能:

- 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,可节省系统功耗
- 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

配置位用于决定上电复位 (POR) 和欠压复位 (BOR) 时的时钟源。此后,可在允许的时钟源间切换时钟源。OSCCON 寄存器控制时钟切换并包含与系统时钟相关的状态位。

表 20-1 汇总了 dsPIC30F 的振荡器工作模式。图 20-1 给出了振荡器系统的简化框图。

dsPIC30F5011/5013

表 20-1：振荡器工作模式

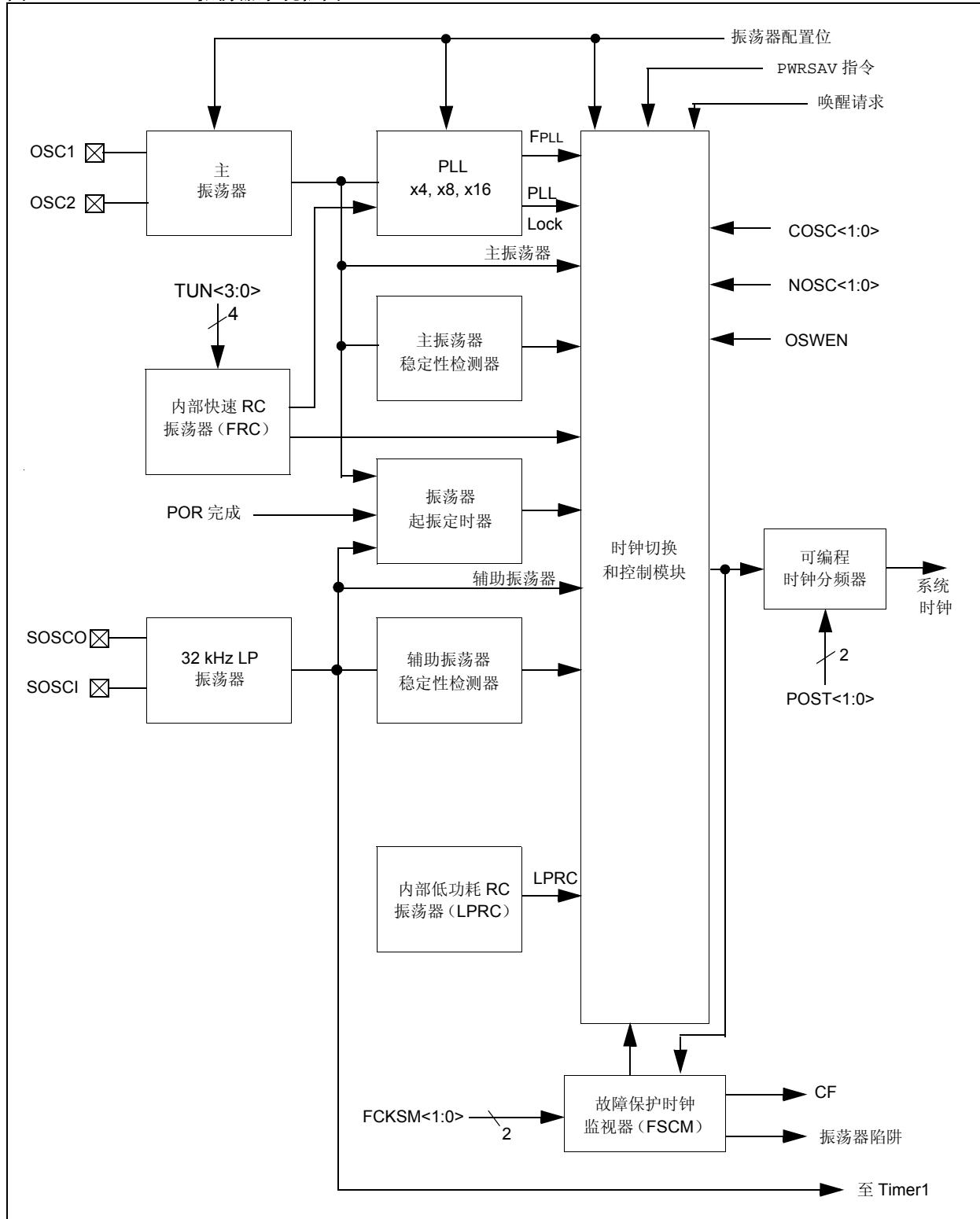
振荡器模式	说明
XTL	OSC1:OSC2 上的晶振频率为 200 kHz-4 MHz。
XT	OSC1:OSC2 上的晶振频率为 4 MHz -10 MHz。
XT, PLL 4x	OSC1:OSC2 上的晶振频率为 4 MHz-10 MHz, 使能 4x PLL。
XT, PLL 8x	OSC1:OSC2 上的晶振频率为 4 MHz-10 MHz, 使能 8x PLL。
XT, PLL 16x	OSC1:OSC2 上的晶振频率为 4 MHz-10 MHz, 使能 16x PLL ⁽¹⁾ 。
LP	SOSCO:SOSCI 上的晶振频率为 32 kHz ⁽²⁾ 。
HS	晶振频率为 10MHz-25MHz。
EC	外部时钟输入 (0-40 MHz)。
ECIO	外部时钟输入 (0-40 MHz), OSC2 引脚为 I/O。
EC, PLL 4x	外部时钟输入 (4-10 MHz), OSC2 引脚为 I/O, 使能 4x PLL ⁽¹⁾ 。
EC, PLL 8x	外部时钟输入 (4-10 MHz), OSC2 引脚为 I/O, 使能 8x PLL ⁽¹⁾ 。
EC, PLL 16x	外部时钟输入 (4-7.5 MHz), OSC2 引脚为 I/O, 使能 16x PLL ⁽¹⁾ 。
ERC	外部 RC 振荡器, OSC2 引脚为 Fosc/4 输出 ⁽³⁾ 。
ERCIO	外部 RC 振荡器, OSC2 引脚为 I/O ⁽³⁾ 。
FRC	7.37 MHz 内部 RC 振荡器。
FRC, PLL 4x	7.37 MHz 内部 RC 振荡器, 使能 4x PLL。
FRC, PLL 8x	7.37 MHz 内部 RC 振荡器, 使能 8x PLL。
FRC, PLL 16x	7.37 MHz 内部 RC 振荡器, 使能 16x PLL。
LPRC	512 kHz 内部 RC 振荡器。

注 1: 必须满足 dsPIC30F 最大工作频率为 120 MHz 的要求。

2: LP 振荡器可方便地为系统时钟和 Timer1 实时时钟所共用。

3: 需要外部 R 和 C 元件。工作频率最高为 4 MHz。

图 20-1: 振荡器系统框图



20.2 振荡器配置

20.2.1 初始时钟源选择

当退出上电复位或欠压复位状态时，器件根据以下原则进行时钟源的选择：

- 通过 FOS<1:0> 配置位选择四个振荡器组中的一组，
- 通过 FPR<3:0> 配置位选择主振荡器组中的 13 个振荡器之一。

表 20-2 给出了时钟选择的配置位值。

表 20-2：时钟选择的配置位值

振荡器模式	振荡器源	FOS1	FOS0	FPR3	FPR2	FPR1	FPR0	OSC2 功能
EC	主	1	1	1	0	1	1	CLKO
ECIO	主	1	1	1	1	0	0	I/O
EC, PLL 4x	主	1	1	1	1	0	1	I/O
EC, PLL 8x	主	1	1	1	1	1	0	I/O
EC, PLL 16x	主	1	1	1	1	1	1	I/O
ERC	主	1	1	1	0	0	1	CLKO
ERCIO	主	1	1	1	0	0	0	I/O
XT	主	1	1	0	1	0	0	OSC2
XT, PLL 4x	主	1	1	0	1	0	1	OSC2
XT, PLL 8x	主	1	1	0	1	1	0	OSC2
XT, PLL 16x	主	1	1	0	1	1	1	OSC2
XTL	主	1	1	0	0	0	0	OSC2
FRC, PLL 4x	内部 FRC	1	1	0	0	0	1	I/O
FRC, PLL 8x	内部 FRC	1	1	1	0	1	0	I/O
FRC, PLL 16x	内部 FRC	1	1	0	0	1	1	I/O
HS	主	1	1	0	0	1	0	OSC2
LP	辅助	0	0	—	—	—	—	(注 1, 2)
FRC	内部 FRC	0	1	x	x	x	x	(注 1, 2)
LPRC	内部 LPRC	1	0	—	—	—	—	(注 1, 2)

注 1: OSC2 引脚功能由主振荡器模式选择位 (FPR<3:0>) 决定。

2: 即使始终选择辅助振荡器或内部时钟源，OSC1 引脚仍不能用作 I/O 引脚。

20.2.2 振荡器起振定时器 (OST)

为了确保晶振（或陶瓷谐振器）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，在允许振荡器时钟应用于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。每当振荡器重启（即当 POR、BOR 和从休眠模式唤醒时）需进行 TOST 时间的延时。振荡器起振定时器适用于 LP 振荡器和主振荡器的 XT、XTL 和 HS 模式（从休眠模式唤醒、POR 和 BOR 时）。

20.2.3 LP 振荡器控制

通过以下各位可使能 LP 振荡器：

- 当前振荡器组控制位 COSC<1:0>。
- LPOSCEN 位 (OSCON 寄存器)。

如果 LPOSCEN = 1，则 LP 振荡器开启（即使处于休眠模式）。在以下条件下，LP 振荡器将作为器件时钟：

- COSC<1:0> = 00（选择 LP 作为主振荡器）且
- LPOSCEN = 1

保持 LP 振荡器始终开启可快速切换至 32 kHz 系统时钟以实现低功耗运行。返回更快速的主振荡器仍需要经过起振时间。

20.2.4 锁相环 (PLL)

PLL 可将主振荡器或快速 RC 振荡器产生的时钟信号进行倍频。PLL 的可选增益有 x4、x8 和 x16。表 20-3 汇总了输入和输出频率范围。

表 20-3: PLL 频率范围

F _{IN}	PLL 倍频	F _{OUT}
4 MHz-10 MHz	x4	16 MHz-40 MHz
4 MHz-10 MHz	x8	32 MHz-80 MHz
4 MHz-7.5 MHz	x16	64 MHz-120 MHz

PLL 具有一个锁定输出，当 PLL 进入相位锁定状态，锁定输出有效。如果锁相环锁定失败（例如由噪音导致），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。

20.2.5 快速 RC 振荡器 (FRC)

FRC 振荡器是一种快速（标称值为 $7.37 \text{ MHz} \pm 2\%$ ）的内部 RC 振荡器。该振荡器用于提供合理的器件工作速度而无需使用外部晶振、陶瓷谐振器或 RC 网络。FRC 振荡器可与 PLL 一同使用以获得更高的时钟频率。

一旦 OSCCON 寄存器中的当前振荡器选择控制位 (OSCCON<13:12>) 设置为 01，dsPIC30F 将使用 FRC 振荡器进行工作。

由 TUN<3:0> (OSCCON<15:14> 和 OSCCON<11:10>) 指定的四位位域允许用户调节内部快速 RC 振荡器（标称值为 7.37 MHz ）。用户可在厂商校准设定值的 -12% (840 kHz) 至 +10.5% (960 kHz) 范围内对 FRC 振荡器进行调节，单步调节的幅度为 1.50%（见表 20-4）。

如果 OSCCON<13:12> 设置为 11，且将 FPR<3:0> 设置为 0001、1010 或 0011，则 PLL 倍频比将分别为 4、8 或 16。

注: 使用 16 倍频 PLL 时，不能将 FRC 的频率调节为大于 7.5 MHz。

表 20-4: FRC 调节

TUN<3:0> 位	FRC 频率
0111	+ 10.5%
0110	+ 9.0%
0101	+ 7.5%
0100	+ 6.0%
0011	+ 4.5%
0010	+ 3.0%
0001	+ 1.5%
0000	中心频率（振荡器运行于校准频率）
1111	- 1.5%
1110	- 3.0%
1101	- 4.5%
1100	- 6.0%
1011	- 7.5%
1010	- 9.0%
1001	- 10.5%
1000	- 12.0%

20.2.6 低功耗 RC 振荡器 (LPRC)

LPRC 振荡器是看门狗定时器 (WDT) 的组成部分，其标称振荡频率为 512 kHz。LPRC 振荡器是上电延时定时器 (PWRT) 电路、WDT 和时钟监视器电路的时钟源。它也可用于低功耗要求高但时序精度要求不高的应用场合作为低频时钟源。

由于 LPRC 振荡器是 PWRT 的时钟源，因此上电复位时该振荡器总是被使能。PWRT 超时后，如果以下任何一个条件为真，LPRC 振荡器将保持为开启状态：

- 已使能故障保护时钟监视器
- 已使能 WDT
- 已通过设置 OSCCON 寄存器中的 COSC<1:0> 控制位选择 LPRC 振荡器作为系统时钟

如果以上任何一个条件都不为真，则 PWRT 超时之后 LPRC 将被关闭。

注 1: OSC2 引脚功能由主振荡器模式选择位 (FPR<3:0>) 决定。

2: 即使始终选择辅助振荡器或内部时钟源，OSC1 引脚仍不能用作 I/O 引脚。

20.2.7 故障保护时钟监视器

故障保护时钟监视器（FSCM）允许器件在即使振荡器发生故障的情况下仍能继续运行。可通过正确地设置 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）使能 FSCM 功能。如果使能了 FSCM 功能，LPRC 内部振荡器将始终保持运行（休眠模式除外），并且不再受 SWDTEN 位的控制。

在发生振荡器故障时，FSCM 会产生时钟故障陷阱事件，并将系统时钟切换到 FRC 振荡器。然后用户可以选择尝试重启振荡器，或执行受控关闭。用户可以把陷阱当作热复位来处理，这只要把复位地址装入振荡器陷阱向量即可。这种情况下，一旦识别到时钟故障，CF（时钟故障）状态位（OSCCON<3>）也将置 1。

发生时钟故障时，WDT 不受影响，且继续靠 LPRC 时钟运行。

从 POR、BOR 或休眠退出后，如果振荡器起振非常慢的话，则可能出现这样的情况，在振荡器起振之前 PWRT 定时器就已超时。在这样的情形下，FSCM 被激活，且 FSCM 会启动一个时钟故障陷阱，FRC 振荡器选择将装入 COSC<1:0> 位中。这实际上是关闭了正在尝试起振的原振荡器。

在时钟故障陷阱 ISR 中，用户可以检测到这样的情况并重启振荡器。

检测到时钟故障时，FSCM 模块将按如下步骤将时钟切换至 FRC 振荡器：

1. 把 FRC 振荡器选择位的值装入 COSC 位（OSCCON<13:12>）中。
2. 将 CF 位（OSCCON<3>）置 1。
3. 清零 OSWEN 控制位（OSCCON<0>）。

出于时钟切换的目的，可将时钟源分为以下四组：

1. 主组
2. 辅助组
3. 内部 FRC 组
4. 内部 LPRC 组

用户可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，则组内的选择始终由 FPR<3:0> 配置位决定。

OSCCON 寄存器包含与时钟切换相关的控制和状态位。

- COSC<1:0>：只读状态位，始终反映当前有效的振荡器组。
- NOSC<1:0>：控制位，写此位表明选择了新的振荡器组。
 - 在POR和BOR时，将把配置位FOS<1:0>的值装入 COSC<1:0> 和 NOSC<1:0> 中。
- LOCK：LOCK 状态位，表明 PLL 是否锁定。
- CF：只读状态位，表明是否检测到时钟故障。
- OSWEN：控制位，当启动时钟转换序列时，从 0 变为 1。清零 OSWEN 控制位将中止正在进行中的时钟转换（用于挂起情形）。

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。这是默认的配置位设置。

如果禁止时钟切换，则 FOS<1:0> 和 FPR<3:0> 位直接控制振荡器选择，COSC<1:0> 位不控制时钟选择。然而，这些位将反映时钟源选择。

注：当使能故障保护时钟监视器时，在应用中不应试图将时钟频率切换到低于 100 kHz。如果执行了这样的时钟切换，则器件可能产生振荡器故障陷阱并切换至快速 RC 振荡器。

20.2.8 防止意外写入 OSCCON

由于 OSCCON 寄存器控制时钟切换和时钟分频，因此有意地将其写入操作过程设计得较为困难。

要写入 OSCCON 低字节，必须执行以下代码序列且不应在其中插入任何其他指令：

向 OSCCON 的低字节写入字节 “0x46”
向 OSCCON 的低字节写入字节 “0x57”

允许在一个指令周期内进行字节写操作。写入期望值或使用位操作指令。

要写入 OSCCON 高字节，必须执行以下指令且不应在其中插入任何其他指令：

向 OSCCON 的高字节写入字节 “0x78”
向 OSCCON 的高字节写入字节 “0x9A”

允许在一个指令周期内进行字节写操作。写入期望值或使用位操作指令。

20.3 复位

dsPIC30F5011/5013 有以下几种不同类型的复位：

- 上电复位 (POR)
- 正常工作期间的 MCLR 复位
- 休眠期间的 MCLR 复位
- 看门狗定时器 (WDT) 复位 (正常工作期间)
- 可编程欠压复位 (BOR)
- RESET 指令
- 由陷阱锁定 (TRAPR) 导致的复位
- 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

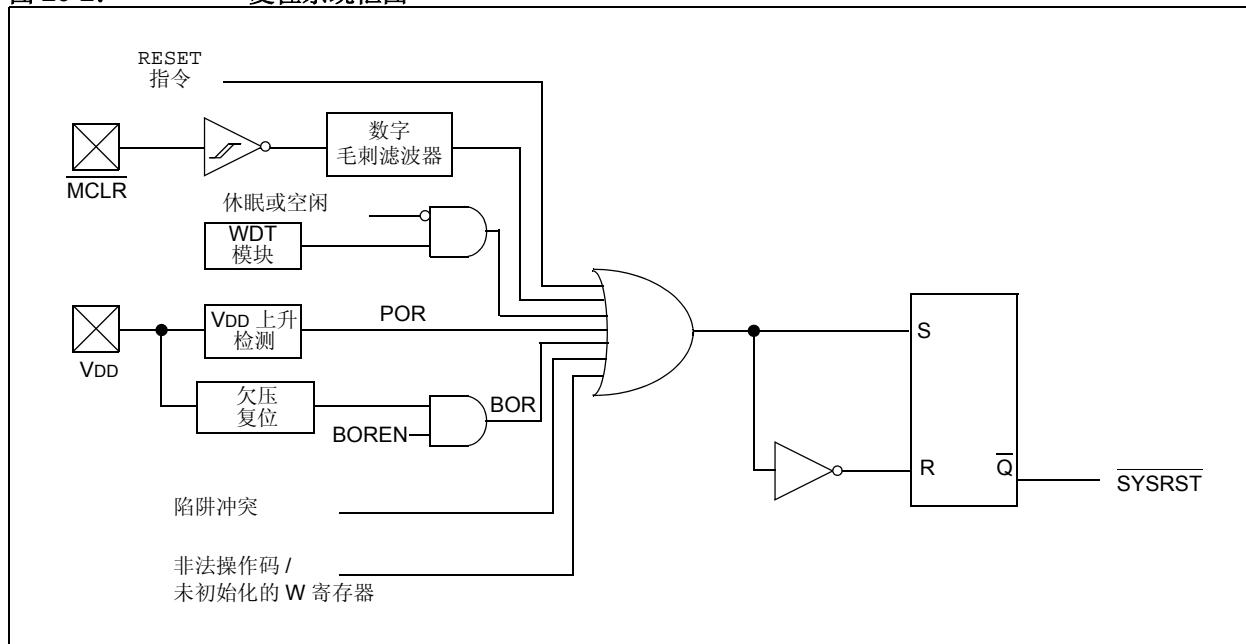
各种复位条件以不同的方式影响不同的寄存器。大多数寄存器不受 WDT 唤醒的影响，因为这被视为是正常工作的继续。在不同的复位条件下，将以不同的方式置 1 或清零 RCON 寄存器中的状态位，如表 20-5 所示。在软件中使用这些位来确定复位的性质。

片上复位电路的框图，如图 20-2 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 20-2：复位系统框图



20.3.1 POR: 上电复位

当检测到 VDD 上升时，上电事件将产生内部 POR 脉冲。复位脉冲将在 POR 电路门限电压 (VPOR) 处产生，标称门限值是 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熔丝位指定。

POR 电路将插入一个标称值为 10 μ 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 参数由器件配置位设定，可以是 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，SYSRST 将在 Q1 时钟的下一个上升沿反相，PC 将跳转到复位向量。

SYSRST 信号时序如图 20-3 至图 20-5 所示。

dsPIC30F5011/5013

图 20-3：上电过程中的延时时序（MCLR 连接至 VDD）

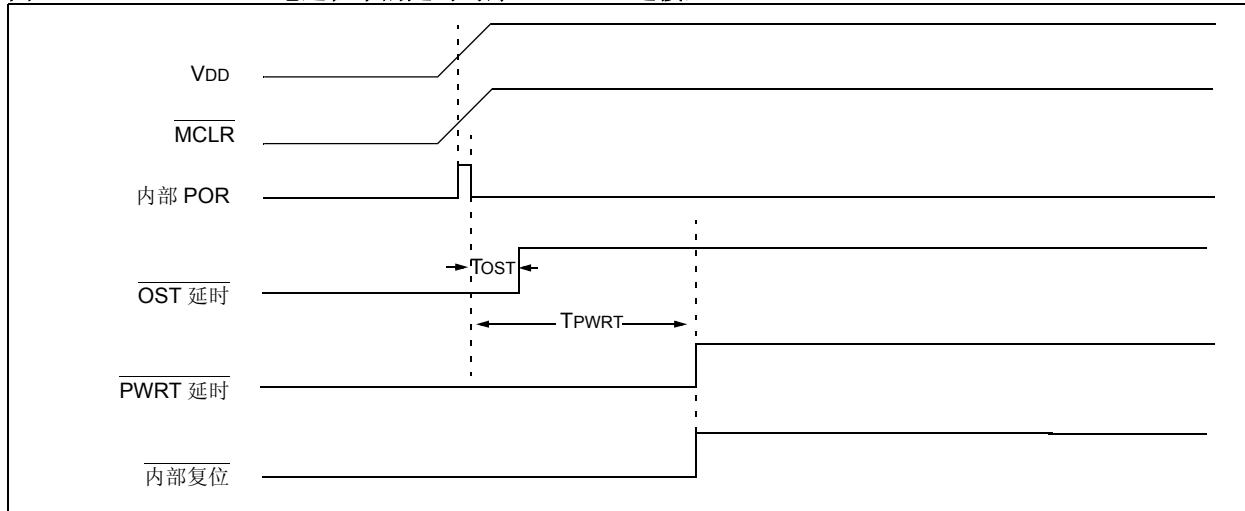


图 20-4：上电过程中的延时时序（MCLR 未连接至 VDD）：情形 1

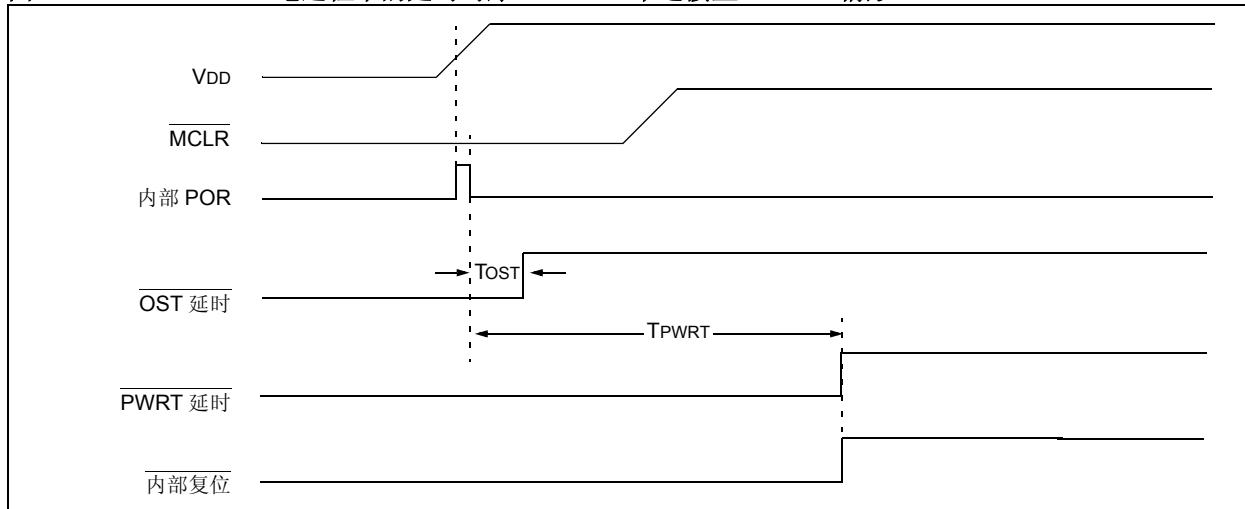
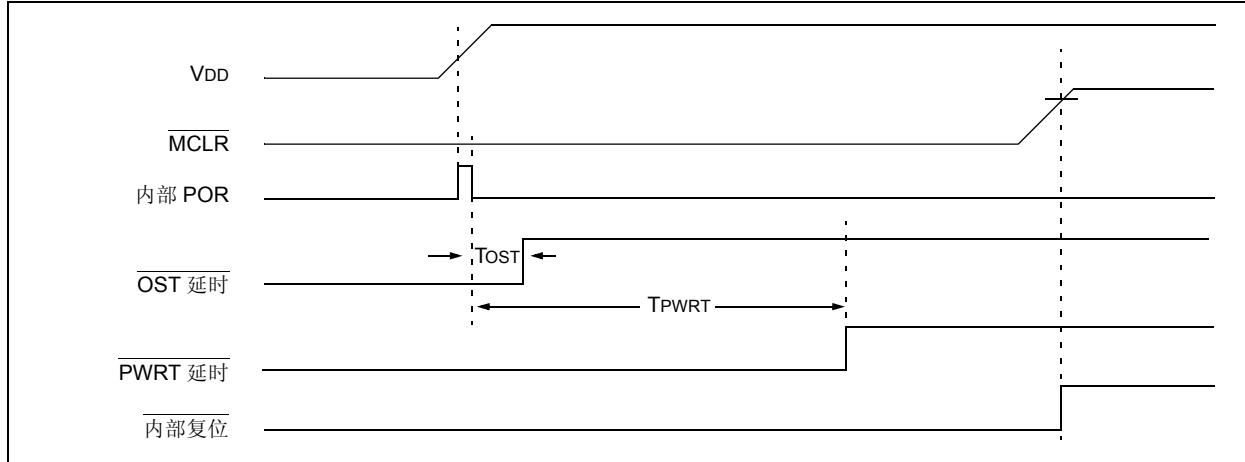


图 20-5：上电过程中的延时时序（MCLR 未连接至 VDD）：情形 2



20.3.1.1 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）的起振时间会相对较长。从而，在 POR 定时器和 PWRT 超时后，可能会出现下列情况：

- 振荡器还没有开始振荡。
- 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。器件将自动切换到 FRC 振荡器，而用户可在陷阱 ISR 中切换到期望的晶体振荡器。

20.3.1.2 FSCM 和 PWRT 禁止时的工作

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、LPRC、EXT RC 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处于冻结状态，直到系统时钟启动为止。从用户的角度看，器件看上去仍然在复位状态，直到系统时钟可用为止。

20.3.2 BOR：可编程欠压复位

BOR（欠压复位）模块是基于内部参考电压电路的。BOR 模块的主要用途是在发生欠压条件时产生器件复位。欠压条件通常由交流电源线上的干扰信号（即，由于电源传输线路不良造成的交流周期波形部分丢失，或者由于接入大感性负载时电流消耗过大造成电压下降）产生的。

BOR 模块允许选择以下电压跳变点之一（见表 23-11）：

- 2.6V-2.71V
- 4.1V-4.4V
- 4.58V-4.73V

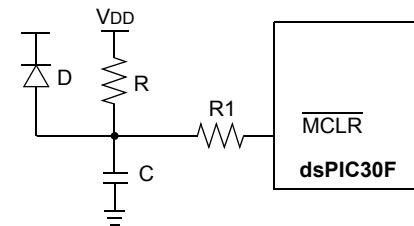
注：这里所述的 BOR 电压跳变点都是标称值，仅供设计参考。有关 BOR 电压限制规范，请参见具体器件数据手册中的“电气规范”章节。

BOR 将产生复位器件的复位脉冲。BOR 会根据器件配置位（FOS<1:0> 和 FPR<3:0>）的值选择时钟源。此外，如果选择了振荡器模式，BOR 将激活振荡器起振定时器（OST）。系统时钟将保持到 OST 超时。如果使用了 PLL，则时钟将被保持到 LOCK 位（OSCCON<5>）置 1。

同时，在内部复位信号释放之前，还要施加 POR 延时（TPOR）和 PWRT 延时（Tpwrt）。如果 Tpwrt = 0 且正在使用的是晶体振荡器的话，那么将施加 Tfscm = 100 μs 的标称延时。这样，总延时为（TPOR + Tfscm）。

BOR 状态位（RCON<1>）将置 1，以表明发生了 BOR。使能时，BOR 电路将在休眠或空闲模式下继续工作，当 VDD 下降到 BOR 门限电压以下时将复位器件。

图 20-6：外部上电复位电路（对于较慢的 VDD 上电过程）



- 注**
- 1:** 仅当 VDD 上电速率过慢时，才需要外部上电复位电路。二极管 D 有助于在 VDD 掉电时使电容迅速放电。
 - 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，这样在由于静电放电（Electrostatic Discharge, ESD）或电过载（Electrical Overstress, EOS）而导致 MCLR/VPP 引脚损坏时，R1 将对从外部电容 C 流入 MCLR 引脚的电流进行限流。

注：专用监视器件，如 MCP1XX 和 MCP8XX 等，也可用作外部上电复位电路。

dsPIC30F5011/5013

表 20-5 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：所有位状态是在“条件”这一列指定操作之前的位状态的反码。

表 20-5： RCON 寄存器的初始化条件：情形 1

条件	程序计数器	TRAPR	IOPUWR	EXTR	SWR	WDTO	IDLE	SLEEP	POR	BOR
上电复位	0x000000	0	0	0	0	0	0	0	1	1
欠压复位	0x000000	0	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	0
正常工作期间的软件复位	0x000000	0	0	0	1	0	0	0	0	0
休眠期间的 MCLR 复位	0x000000	0	0	1	0	0	0	1	0	0
空闲期间的 MCLR 复位	0x000000	0	0	1	0	0	1	0	0	0
WDT 超时复位	0x000000	0	0	0	0	1	0	0	0	0
WDT 唤醒	PC + 2	0	0	0	0	1	0	1	0	0
被中断从休眠状态唤醒	PC + 2 ⁽¹⁾	0	0	0	0	0	0	1	0	0
时钟故障陷阱	0x000004	0	0	0	0	0	0	0	0	0
陷阱复位	0x000000	1	0	0	0	0	0	0	0	0
非法操作陷阱	0x000000	0	1	0	0	0	0	0	0	0

图注： u = 不变， x = 未知， - = 未实现位，读为 0

注 1： 如果唤醒由允许的中断引起， PC 将装入相应的中断向量。

表 20-6 显示了 RCON 寄存器位状态的第二个示例。在此情形中，假设用户在条件列中所指定的操作前未置 1/清零特定位。

表 20-6： RCON 寄存器的初始化条件：情形 2

条件	程序计数器	TRAPR	IOPUWR	EXTR	SWR	WDTO	IDLE	SLEEP	POR	BOR
上电复位	0x000000	0	0	0	0	0	0	0	1	1
欠压复位	0x000000	u	u	u	u	u	u	u	0	1
正常工作期间的 MCLR 复位	0x000000	u	u	1	0	0	0	0	u	u
正常工作期间的软件复位	0x000000	u	u	0	1	0	0	0	u	u
休眠期间的 MCLR 复位	0x000000	u	u	1	u	0	0	1	u	u
空闲期间的 MCLR 复位	0x000000	u	u	1	u	0	1	0	u	u
WDT 超时复位	0x000000	u	u	0	0	1	0	0	u	u
WDT 唤醒	PC + 2	u	u	u	u	1	u	1	u	u
被中断从休眠状态唤醒	PC + 2 ⁽¹⁾	u	u	u	u	u	u	1	u	u
时钟故障陷阱	0x000004	u	u	u	u	u	u	u	u	u
陷阱复位	0x000000	1	u	u	u	u	u	u	u	u
非法操作复位	0x000000	u	1	u	u	u	u	u	u	u

图注： u = 不变， x = 未知， - = 未实现位，读为 0

注 1： 如果唤醒由允许的中断引起， PC 将装入相应的中断向量。

20.4 看门狗定时器 (WDT)

20.4.1 看门狗定时器的工作

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，它使用片上 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 定时器仍将继续工作。

20.4.2 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 FWDT 中的配置位 (FWDTEN) 来进行。

设置 FWDTEN = 1，使能看门狗定时器。使能是在对器件进行编程时完成的。默认情况下，在擦除芯片后，FWDTEN 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。

如果使能的话，WDT 将递增，直到它溢出或“超时”。WDT 超时将强制器件复位（除非是在休眠期间）。要防止 WDT 超时，用户必须使用 CLRWDT 指令清零看门狗定时器。

如果休眠期间 WDT 超时，器件将唤醒。RCON 寄存器中的 WDTO 位将清零，表明 WDT 超时导致唤醒。

设置 FWDTEN = 0，允许用户软件通过 SWDTEN (RCON<5>) 控制位来使能 / 禁止看门狗定时器。

20.5 低电压检测

低电压检测 (LVD) 模块用于检测器件的 VDD 何时降至低于门限电压 VLVD，该电压由 LVDL<3:0> 位 (RCON<11:8>) 确定，因此是用户可编程的。内部电压参考电路要求一段标称稳定时间，而 BGST 位 (RCON<13>) 表明参考电压何时稳定。

在一些器件中，可在 LVDIN 引脚外部施加 LVD 门限电压。

通过将 LVDEN 位 (RCON<12>) 置 1 来使能 LVD 模块。

20.6 省电模式

通过执行一条特殊的指令 PWRSAV，可以进入两个省电状态；这两个状态是休眠和空闲。

PWRSAV 指令的格式如下：

PWRSAV <parameter>，其中“parameter”定义是空闲还是休眠模式。

20.6.1 休眠模式

在休眠模式下，CPU 和外设的时钟关闭。如果在使用片上振荡器，它也将关闭。

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。

欠压保护电路和低电压检测电路，如果使能的话，在休眠期间仍然起作用。

发生以下任一事件时，处理器将从休眠模式唤醒：

- 任何中断，如果已单独允许并且满足所需的优先级
- 任何复位 (POR、BOR 和 MCLR)
- WDT 超时

从休眠模式唤醒时，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。当时钟切换使能时，COSC<1:0> 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 POR 或 BOR，则根据 FOS<1:0> 和 FPR<3:0> 配置位来选择振荡器。

如果时钟源是振荡器，器件的时钟将一直关闭，直到 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，直到 LOCK = 1 为止（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、FRC、LPRC 或 EXTRC 振荡器，那么要加上 Tpor 的延时 (~ 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 是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）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SLEEP状态位将置1。

注： 尽管要加上各种延时（Tpor、TLOCK和Tpwr），在延时结束后晶振（和PLL）可能并未处于工作状态（例如，对于低频晶振）。在这样的情形下，如果FSCM被使能，则器件将这种情况作为时钟故障并处理时钟故障陷阱，FRC振荡器将被使能，而用户必须重新使能晶振。如果FSCM没有使能，器件将暂停代码执行直至时钟稳定，且将继续处于休眠模式直至振荡器时钟启动。

所有复位都将把处理器从休眠模式中唤醒。除POR之外的任何复位，都将置1SLEEP状态位。POR时，SLEEP状态位将被清零。

如果看门狗定时器被使能，则在WDT超时时处理器将从休眠模式中唤醒。SLEEP和WDTO状态位都将被置1。

20.6.2 空闲模式

在空闲模式下，CPU的时钟关闭，而外设继续运行。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，控制外设在空闲期间是否工作。

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。

发生以下任一事件时，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、BOR和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，如果已单独允许（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IDLE状态位将置1。

除POR之外的任何复位，都将把IDLE状态位置1。POR时，IDLE状态位将被清零。

如果看门狗定时器被使能，则处理器在WDT超时时将从空闲模式中唤醒。IDLE和WDTO状态位都将被置1。与从休眠中唤醒不同的是，从空闲中唤醒时不存在延时。

20.7 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6个器件配置寄存器可供用户使用：

1. Fosc（0xF80000）：振荡器配置寄存器
2. FWDT（0xF80002）：看门狗定时器配置寄存器
3. FBORPOR（0xF80004）：BOR和POR配置寄存器
4. FBS（0xF80006）：引导代码段配置寄存器
5. FSS（0xF80008）：安全代码段配置寄存器
6. FGS（0xF8000A）：通用代码段配置寄存器

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《dsPIC30F闪存编程规范》（DS70102G_CN）、《dsPIC30F系列参考手册》（DS70046E_CN）和“CodeGuard™ Security”章节（DS70180）。

注：

1. 如果已编程了代码保护熔丝位（FBS<BSS<2:0>、FSS<SSS<2:0>、FGS<GCP>和FGS<GWRP>），则只能在电压 $VDD \geq 4.5V$ 下时才能擦除整个代码保护的器件。
2. 该器件支持CodeGuard™安全性的高级实现。有关如何在应用中使用CodeGuard安全性的信息，请参见“CodeGuard Security”章节（DS70180）。

20.8 外设模块禁止（PMD）寄存器

外设模块禁止（Peripheral Module Disable, PMD）寄存器提供了一种禁止外设模块的方法，即，停止提供给模块的所有时钟源。通过对 PMD 控制位禁止某个外设时，外设将处于最低功耗状态。与外设相关的控制和状态寄存器也会被禁止，所以读写这些寄存器的操作都无效。

外设模块使能的条件是：PMD 寄存器中相关位清零，且特定的 dsPIC DSC 器件支持该外设。如果器件中存在相应的外设模块，则默认情况下在 PMD 寄存器中使能外设。

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

20.9 在线调试器

如果选择 MPLAB® ICD2 作为调试器，则使能在线调试功能。与 MPLAB IDE 一起使用时，该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 4 对调试 I/O 引脚中的一对。这些引脚为 EMUD/EMUC、EMUD1/EMUC1、EMUD2/EMUC2 和 MUD3/EMUC3。

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、接收响应，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 引脚功能均与 PGD 和 PGC 引脚功能复用。
2. 如果选取了 EMUD1/EMUC1、EMUD2/EMUC2 或 EMUD3/EMUC3 作为调试 I/O 引脚对，那么需要 7 引脚的接口，这是因为，EMUDx/EMUCx 引脚功能 ($x = 1, 2$ 或 3) 不与 PGD 和 PGC 引脚功能复用。

表 20-7：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	BGST	LVDEN	LVDL<3:0>			EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	(注 1)	
OSCCON	0742	TUN3	TUN2	COSC<1:0>		TUN1	TUN0	NOSC<1:0>		POST<1:0>		LOCK	—	CF	—	LPOSCEN	OSWEN	(注 2)
PMD1	0770	T5MD	T4MD	T3MD	T2MD	T1MD	—	—	DCIMD	I2CMD	U2MD	U1MD	SPI2MD	SPI1MD	C2MD	C1MD	ADCMD	0000 0000 0000 0000
PMD2	0772	IC8MD	IC7MD	IC6MD	IC5MD	IC4MD	IC3MD	IC2MD	IC1MD	OC8MD	OC7MD	OC6MD	OC5MD	OC4MD	OC3MD	OC2MD	OC1MD	0000 0000 0000 0000

注 1：复位状态取决于复位类型。

2：复位状态取决于配置位。

表 20-8：器件配置寄存器映射

寄存器名称	地址	Bit 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
FOSC	F80000	—	FCKSM<1:0>			—	—	—	FOS<1:0>			—	—	—	—	FPR<3:0>		
FWDT	F80002	—	FWDTEN		—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
FBORPOR	F80004	—	MCLREN		—	—	—	—	—	—	—	BOREN	—	—	—	—	—	FPWRT<1:0>
FBS	F80006	—	—	—	RBS1	RBS0	—	—	—	EBS	—	—	—	—	BSS<2:0>			BWRP
FSS	F80008	—	—	—	RBS1	RBS0	—	—	ESS1	ESS0	—	—	—	—	SSS<2:0>			SWRP
FGS	F8000A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GSS1	GSS0

注：有关寄存器各位的说明，请参见《dsPIC30F 系列参考手册》(DS70046E_CN)。

dsPIC30F5011/5013

注：

21.0 指令集汇总

注：本数据手册总结了 dsPIC30F 系列器件的功能，但是不应把本手册当作无所不包的参考手册来使用。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 5 个基本类别：

- 针对字或字节的操作
- 针对位的操作
- 立即数操作
- DSP 操作
- 控制操作

表 21-1 给出了在说明指令时使用的通用符号。

表 21-2 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
- 运算结果的目标寄存器，通常是寄存器 Wd，带或不带地址修改量

不过，针对字或字节的文件寄存器指令只有两个操作数：

- 文件寄存器由 f 值指定
- 目标寄存器可以是文件寄存器 f 或 W0 寄存器（用 WREG 表示）

大多数位操作类指令（包括简单的循环 / 移位指令）有两个操作数：

- W 寄存器（带或不带地址修改量）或文件寄存器（由 Ws 或 f 的值指定）
- W 寄存器或文件寄存器中的位（由一个立即数指定，或者由寄存器 Wb 的内容间接指定）

涉及数据传送的立即数指令，可以使用下列操作数：

- 要被装入到 W 寄存器或文件寄存器中的立即数（由 k 值指定）
- 要装入立即数的 W 寄存器或文件寄存器（由 Wb 或 f 指定）

然而，涉及算术或逻辑操作的立即数指令，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
- 操作结果的目标寄存器（仅在与第一个源操作数不同时）通常是寄存器 Wd（带或不带地址修改量）

MAC 类 DSP 指令可使用下列操作数：

- 要使用的累加器（A 或 B）（必需的操作数）
- 要用作两个操作数的 W 寄存器
- X 和 Y 地址空间预取操作
- X 和 Y 地址空间预取目标寄存器
- 累加器回写目标寄存器

与乘法无关的其他 DSP 指令使用的操作数可能包括：

- 要使用的累加器（必需）
- 源操作数或目标操作数（分别由 Wso 或 Wdo 指定），带或不带地址修改量
- 移位位数，由 W 寄存器 Wn 或立即数指定

控制指令可以使用下列操作数：

- 程序存储地址
- 表读和表写指令的模式

除了某些双字指令外，所有指令都是单字指令；双字指令之所以是双字长的（48位），是因为要用48位来提供所需信息。在第二个字中，8个MSb全为0。如果指令自身把第二个字当作一条指令来执行的话，它将作为一条NOP指令来执行。

大多数单字长指令都在一个指令周期内执行，除非条件测试为真或者指令运行结果改变了程序计数器。对于上述两种特殊情况，指令执行需要两个指令周期，在第二个指令周期中执行一条NOP指令。值得注意的例外是BRA（无条件/计算转移）、间接CALL/GOTO、所有的表读和表写以及RETURN/RETFIE指令，它们是单字长指

令，但执行需要两个或三个周期。某些与跳过后续指令有关的指令，如果要执行跳过的话，可能需要两个或三个周期，这取决于被跳过的指令是单字还是双字指令。此外，双字传送需要两个周期。双字指令执行需要两个指令周期。

注： 有关指令集的更多详细信息，请参见《dsPIC30F/33F 程序员参考手册》(DS70157B_CN)。

表 21-1：操作码说明中使用的符号

字段	说明
#text	表示由text定义的立即数
(text)	表示text的内容
[text]	表示由text寻址的存储单元
{ }	可选字段或操作
<n:m>	寄存器位域
.b	字节模式选择
.d	双字模式选择
.S	影子寄存器选择
.w	字模式选择（默认）
ACC	两个累加器{A, B}之一
AWB	累加器回写目标地址寄存器 ∈ {W13, [W13]+=2}
bit4	4位位选择字段（用于字寻址指令）∈ {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 ∈ {0x0000...0xFFFF}
lit1	1位无符号立即数 ∈ {0,1}
lit4	4位无符号立即数 ∈ {0...15}
lit5	5位无符号立即数 ∈ {0...31}
lit8	8位无符号立即数 ∈ {0...255}
lit10	10位无符号立即数，对于字节模式， ∈ {0...255}；对于字模式， ∈ {0:1023}
lit14	14位无符号立即数 ∈ {0...16384}
lit16	16位无符号立即数 ∈ {0...65535}
lit23	23位无符号立即数 ∈ {0...8388608}；LSB必须为0
None	字段无需内容，可为空
OA, OB, SA, SB	DSP状态位：ACCA溢出、ACCB溢出、ACCA饱和和ACCB饱和
PC	程序计数器
Slit10	10位有符号立即数 ∈ {-512...511}
Slit16	16位有符号立即数 ∈ {-32768...32767}
Slit6	6位有符号立即数 ∈ {-16...16}

表 21-1：操作码说明中使用的符号（续）

字段	说明
Wb	基本 W 寄存器 $\in \{W0..W15\}$
Wd	目标 W 寄存器 $\in \{Wd, [Wd], [Wd++], [Wd-], [++Wd], [-Wd]\}$
Wdo	目标 W 寄存器 $\in \{Wnd, [Wnd], [Wnd++], [Wnd-], [++Wnd], [-Wnd], [Wnd+Wb]\}$
Wm,Wn	被除数和除数工作寄存器对（直接寻址）
Wm*Wm	用于平方指令的被乘数和乘数工作寄存器对 $\in \{W4*W4, W5*W5, W6*W6, W7*W7\}$
Wm*Wn	用于 DSP 指令的被乘数和乘数工作寄存器对 $\in \{W4*W5, W4*W6, W4*W7, W5*W6, W5*W7, W6*W7\}$
Wn	16 个工作寄存器之一 $\in \{W0..W15\}$
Wnd	16 个目标工作寄存器之一 $\in \{W0..W15\}$
Wns	16 个源工作寄存器之一 $\in \{W0..W15\}$
WREG	W0（文件寄存器指令中使用的工作寄存器）
Ws	源 W 寄存器 $\in \{Ws, [Ws], [Ws++], [Ws-], [++Ws], [-Ws]\}$
Wso	源 W 寄存器 $\in \{Wns, [Wns], [Wns++], [Wns-], [++Wns], [-Wns], [Wns+Wb]\}$
Wx	用于 DSP 指令的 X 数据空间预取地址寄存器 $\in \{[W8]+=6, [W8]+=4, [W8]+=2, [W8], [W8]=-6, [W8]=-4, [W8]=-2, [W9]+=6, [W9]+=4, [W9]+=2, [W9], [W9]=-6, [W9]=-4, [W9]=-2, [W9+W12], 无\}$
Wxd	用于 DSP 指令的 X 数据空间预取目标寄存器 $\in \{W4..W7\}$
Wy	用于 DSP 指令的 Y 数据空间预取地址寄存器 $\in \{[W10]+=6, [W10]+=4, [W10]+=2, [W10], [W10]=-6, [W10]=-4, [W10]=-2, [W11]+=6, [W11]+=4, [W11]+=2, [W11], [W11]=-6, [W11]=-4, [W11]=-2, [W11+W12], 无\}$
Wyd	用于 DSP 指令的 Y 数据空间预取目标寄存器 $\in \{W4..W7\}$

dsPIC30F5011/5013

表 21-2: 指令集概述

基本指令编号	汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
1	ADD	ADD Acc	累加器相加	1	1	OA,OB,SA,SB
		ADD f	$f = f + WREG$	1	1	C,DC,N,OV,Z
		ADD f,WREG	$WREG = f + WREG$	1	1	C,DC,N,OV,Z
		ADD #lit10,Wn	$Wd = lit10 + Wd$	1	1	C,DC,N,OV,Z
		ADD Wb,Ws,Wd	$Wd = Wb + Ws$	1	1	C,DC,N,OV,Z
		ADD Wb,#lit5,Wd	$Wd = Wb + lit5$	1	1	C,DC,N,OV,Z
		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	如果有进位则转移	1	1 (2)	无
		BRA GE,Expr	如果大于或等于则转移	1	1 (2)	无
		BRA GEU,Expr	如果无符号大于或等于则转移	1	1 (2)	无
		BRA GT,Expr	如果大于则转移	1	1 (2)	无
		BRA GTU,Expr	如果无符号大于则转移	1	1 (2)	无
		BRA LE,Expr	如果小于或等于则转移	1	1 (2)	无
		BRA LEU,Expr	如果无符号小于或等于则转移	1	1 (2)	无
		BRA LT,Expr	如果小于则转移	1	1 (2)	无
		BRA LTU,Expr	如果无符号小于则转移	1	1 (2)	无
		BRA N,Expr	如果为负则转移	1	1 (2)	无
		BRA NC,Expr	如果没有进位则转移	1	1 (2)	无
		BRA NN,Expr	如果不为负则转移	1	1 (2)	无
		BRA NOV,Expr	如果未溢出则转移	1	1 (2)	无
		BRA NZ,Expr	如果不为零则转移	1	1 (2)	无
		BRA 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

表 21-2：指令集概述（续）

基本 指令 编号	汇编 助记符	汇编语法	说明	字数	周期数	影响的 状态标志
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
11	BTSS	BTSS f,#bit4	对 f 中的指定位进行测试，如果为 1 则跳过	1	1 (2 或 3)	无
		BTSS Ws,#bit4	对 Ws 中的指定位进行测试，如果为 1 则跳过	1	1 (2 或 3)	无
12	BTST	BTST f,#bit4	对 f 中的指定位进行测试	1	1	Z
		BTST.C Ws,#bit4	对 Ws 中的指定位进行测试，并将其值存储到 C	1	1	C
		BTST.Z Ws,#bit4	对 Ws 中的指定位进行测试，并将其反码存储到 Z	1	1	Z
		BTST.C Ws,Wb	对 Ws<Wb> 位进行测试，并将其值存储到 C	1	1	C
		BTST.Z Ws,Wb	对 Ws<Wb> 位进行测试，并将其反码存储到 Z	1	1	Z
13	BTSTS	BTSTS f,#bit4	对 f 中的指定位进行测试，并将 f 中的该位置 1	1	1	Z
		BTSTS.C Ws,#bit4	对 Ws 中的指定位进行测试，并将其值存储到 C，然后将 Ws 中的该位置 1	1	1	C
		BTSTS.Z Ws,#bit4	对 Ws 中的指定位进行测试，并将其反码存储到 Z，然后将 Ws 中的该位置 1	1	1	Z
14	CALL	CALL lit23	调用子程序	2	2	无
		CALL Wn	间接调用子程序	1	2	无
15	CLR	CLR f	f = 0x0000	1	1	无
		CLR WREG	WREG = 0x0000	1	1	无
		CLR Ws	Ws = 0x0000	1	1	无
		CLR Acc,Wx,Wxd,Wy,Wyd,AWB	将累加器清零	1	1	OA,OB,SA,SB
16	CLRWDT	CLRWDT	将看门狗定时器清零	1	1	WDTO,Sleep
17	COM	COM f	f = 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 - C)	1	1	C,DC,N,OV,Z
21	CPSEQ	CPSEQ Wb, Wn	比较 Wb 和 Wn，如果相等则跳过	1	1 (2 或 3)	无
22	CPSGT	CPSGT Wb, Wn	比较 Wb 和 Wn，如果大于则跳过	1	1 (2 或 3)	无
23	CPSLT	CPSLT Wb, Wn	比较 Wb 和 Wn，如果小于则跳过	1	1 (2 或 3)	无
24	CPSNE	CPSNE Wb, Wn	比较 Wb 和 Wn，如果不相等则跳过	1	1 (2 或 3)	无
25	DAW	DAW Wn	Wn = 十进制调整 Wn	1	1	C
26	DEC	DEC f	f = f - 1	1	1	C,DC,N,OV,Z
		DEC f,WREG	WREG = f - 1	1	1	C,DC,N,OV,Z
		DEC Ws,Wd	Wd = Ws - 1	1	1	C,DC,N,OV,Z
27	DEC2	DEC2 f	f = f - 2	1	1	C,DC,N,OV,Z
		DEC2 f,WREG	WREG = f - 2	1	1	C,DC,N,OV,Z
		DEC2 Ws,Wd	Wd = Ws - 2	1	1	C,DC,N,OV,Z
28	DISI	DISI #lit14	在 k 个指令周期内禁止中断	1	1	无

表 21-2：指令集概述（续）

基本 指令 编号	汇编 助记符	汇编语法	说明	字数	周期数	影响的 状态标志
29	DIV	DIV.S Wm,Wn	有符号 16/16 位整数除法	1	18	N,Z,C,OV
		DIV.SD Wm,Wn	有符号 32/16 位整数除法	1	18	N,Z,C,OV
		DIV.U Wm,Wn	无符号 16/16 位整数除法	1	18	N,Z,C,OV
		DIV.UD Wm,Wn	无符号 32/16 位整数除法	1	18	N,Z,C,OV
30	DIVF	DIVF Wm,Wn	有符号 16/16 位小数除法	1	18	N,Z,C,OV
31	DO	DO #lit14,Expr	执行 Do 循环代码到 PC+Expr, 执行次数为 lit14+1 次	2	2	无
		DO Wn,Expr	执行 DO 循环代码到 PC+Expr, 执行次数为 (Wn)+1 次	2	2	无
32	ED	ED Wm*Wm,Acc,Wx,Wy,Wxd	欧几里德距离（无累加）	1	1	OA,OB,OAB, SA,SB,SAB
33	EDAC	EDAC Wm*Wm,Acc,Wx,Wy,Wxd	欧几里德距离	1	1	OA,OB,OAB, SA,SB,SAB
34	EXCH	EXCH Wns,Wnd	交换 Wns 和 Wnd 的内容	1	1	无
35	FBCL	FBCL Ws,Wnd	从左边 (MSb) 查找第一个位变化	1	1	C
36	FF1L	FF1L Ws,Wnd	从左边 (MSb) 查找第一个 1	1	1	C
37	FF1R	FF1R Ws,Wnd	从右边 (LSb) 查找第一个 1	1	1	C
38	GOTO	GOTO Expr	转移到地址	2	2	无
		GOTO Wn	间接转移到地址	1	2	无
39	INC	INC f	f = f + 1	1	1	C,DC,N,OV,Z
		INC f,WREG	WREG = f + 1	1	1	C,DC,N,OV,Z
		INC Ws,Wd	Wd = Ws + 1	1	1	C,DC,N,OV,Z
40	INC2	INC2 f	f = f + 2	1	1	C,DC,N,OV,Z
		INC2 f,WREG	WREG = f + 2	1	1	C,DC,N,OV,Z
		INC2 Ws,Wd	Wd = Ws + 2	1	1	C,DC,N,OV,Z
41	IOR	IOR f	f = f .IOR.WREG	1	1	N,Z
		IOR f,WREG	WREG = f .IOR.WREG	1	1	N,Z
		IOR #lit10,Wn	Wd = lit10 .IOR.Wd	1	1	N,Z
		IOR Wb,Ws,Wd	Wd = Wb .IOR.Ws	1	1	N,Z
		IOR Wb,#lit5,Wd	Wd = Wb .IOR.lit5	1	1	N,Z
42	LAC	LAC Wso,#Slit4,Acc	装载累加器	1	1	OA,OB,OAB, SA,SB,SAB
43	LINK	LINK #lit14	分配堆栈帧	1	1	无
44	LSR	LSR f	f = 逻辑右移 f	1	1	C,N,OV,Z
		LSR f,WREG	WREG = 逻辑右移 f	1	1	C,N,OV,Z
		LSR Ws,Wd	Wd = 逻辑右移 Ws	1	1	C,N,OV,Z
		LSR Wb,Wns,Wnd	Wnd = 将 Wb 逻辑右移 Wns 位	1	1	N,Z
		LSR Wb,#lit5,Wnd	Wnd = 将 Wb 逻辑右移 lit5 位	1	1	N,Z
45	MAC	MAC Wm*Wn,Acc,Wx,Wxd,Wy,Wyd,AWB	相乘并累加	1	1	OA,OB,OAB, SA,SB,SAB
		MAC Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	平方并累加	1	1	OA,OB,OAB, SA,SB,SAB
46	MOV	MOV f,Wn	将 f 中的内容送入 Wn	1	1	无
		MOV f	将 f 中的内容送入目标寄存器	1	1	N,Z
		MOV f,WREG	将 f 中的内容送入 WREG	1	1	N,Z
		MOV #lit16,Wn	将 16 位立即数送入 Wn	1	1	无
		MOV.b #lit8,Wn	将 8 位立即数送入 Wn	1	1	无
		MOV Wn,f	将 Wn 中的内容送入 f	1	1	无
		MOV Wso,Wdo	将 Ws 中的内容送入 Wd	1	1	无
		MOV WREG,f	将 WREG 中的内容送入 f	1	1	N,Z
		MOV.D Wns,Wd	将 W(ns):W(ns+1) 中的双字内容送入 Wd	1	2	无
		MOV.D Ws,Wnd	将 Ws 中的双字内容送入 W(nd+1):W(nd)	1	2	无
47	MOVSAC	MOVSAC Acc,Wx,Wxd,Wy,Wyd,AWB	预取操作数并保存累加器	1	1	无

表 21-2：指令集概述（续）

基本 指令 编号	汇编 助记符	汇编语法	说明	字数	周期数	影响的 状态标志
48	MPY	MPY Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	Wm 与 Wn 相乘，结果存入累加器	1	1	OA,OB,OAB, SA,SB,SAB
		MPY Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Wm 平方，结果存入累加器	1	1	OA,OB,OAB, SA,SB,SAB
49	MPY.N	MPY.N Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	Wm 与 Wn 相乘并取反，结果存入累加器	1	1	无
50	MSC	MSC Wm*Wm,Acc,Wx,Wxd,Wy,Wyd, AWB	相乘再从累加器中减去	1	1	OA,OB,OAB, SA,SB,SAB
51	MUL	MUL.SS Wb,Ws,Wnd	{Wnd+1, Wnd} = signed(Wb) * signed(Ws)	1	1	无
		MUL.SU Wb,Ws,Wnd	{Wnd+1, Wnd} = signed(Wb) * unsigned(Ws)	1	1	无
		MUL.US Wb,Ws,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * signed(Ws)	1	1	无
		MUL.UU Wb,Ws,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * unsigned(Ws)	1	1	无
		MUL.SU Wb,#lit5,Wnd	{Wnd+1, Wnd} = signed(Wb) * unsigned(lit5)	1	1	无
		MUL.UU Wb,#lit5,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * unsigned(lit5)	1	1	无
		MUL f	W3:W2 = f * WREG	1	1	无
52	NEG	NEG Acc	将累加器内容求补	1	1	OA,OB,OAB, SA,SB,SAB
		NEG f	f = $\bar{f} + 1$	1	1	C,DC,N,OV,Z
		NEG f,WREG	WREG = $\bar{f} + 1$	1	1	C,DC,N,OV,Z
		NEG Ws,Wd	Wd = $\bar{Ws} + 1$	1	1	C,DC,N,OV,Z
53	NOP	NOP	空操作	1	1	无
		NOPR	空操作	1	1	无
54	POP	POP f	将栈顶 (TOS) 的内容弹出到 f	1	1	无
		POP Wdo	将栈顶 (TOS) 的内容弹出到 Wdo	1	1	无
		POP.D Wnd	将栈顶 (TOS) 的内容弹出到 W(nd):W(nd+1)	1	2	无
		POP.S	将影子寄存器的内容弹出到主寄存器	1	1	全部
55	PUSH	PUSH f	将 f 的内容压入栈顶 (TOS)	1	1	无
		PUSH Wso	将 Wso 的内容压入栈顶 (TOS)	1	1	无
		PUSH.D Wns	将 W(ns):W(ns+1) 的双字内容压入栈顶 (TOS)	1	2	无
		PUSH.S	将主寄存器中的双字内容压入影子寄存器	1	1	无
56	PWRSAV	PWRSAV #lit1	进入休眠或空闲模式	1	1	WDTO,Sleep
57	RCALL	RCALL Expr	相对调用	1	2	无
		RCALL Wn	计算调用	1	2	无
58	REPEAT	REPEAT #lit14	将下一条指令重复执行 lit14+1 次	1	1	无
		REPEAT Wn	将下一条指令重复执行 (Wn)+1 次	1	1	无
59	RESET	RESET	软件器件复位	1	1	无
60	RETFIE	RETFIE	从中断返回	1	3 (2)	无
61	RETLW	RETLW #lit10,Wn	返回并将立即数存入 Wn	1	3 (2)	无
62	RETURN	RETURN	从子程序返回	1	3 (2)	无
63	RLC	RLC f	f = 对 f 执行带进位的循环左移	1	1	C,N,Z
		RLC f,WREG	WREG = 对 f 执行带进位的循环左移	1	1	C,N,Z
		RLC Ws,Wd	Wd = 对 Ws 执行带进位的循环左移	1	1	C,N,Z
64	RLNC	RLNC f	f = 循环左移 f (不带进位)	1	1	N,Z
		RLNC f,WREG	WREG = 循环左移 f (不带进位)	1	1	N,Z
		RLNC Ws,Wd	Wd = 循环左移 Ws (不带进位)	1	1	N,Z
65	RRC	RRC f	f = 对 f 执行带进位的循环右移	1	1	C,N,Z
		RRC f,WREG	WREG = 对 f 执行带进位的循环右移	1	1	C,N,Z
		RRC Ws,Wd	Wd = 对 Ws 执行带进位的循环右移	1	1	C,N,Z
66	RRNC	RRNC f	f = 循环右移 f (不带进位)	1	1	N,Z
		RRNC f,WREG	WREG = 循环右移 f (不带进位)	1	1	N,Z
		RRNC Ws,Wd	Wd = 循环右移 Ws (不带进位)	1	1	N,Z

表 21-2: 指令集概述 (续)

基本指令编号	汇编助记符	汇编语法	说明	字数	周期数	影响的状态标志
67	SAC	SAC Acc,#Slit4,Wdo	保存累加器内容	1	1	无
		SAC.R Acc,#Slit4,Wdo	保存舍入后的累加器内容	1	1	无
68	SE	SE Ws,Wnd	Wnd = 符号扩展后的 Ws	1	1	C,N,Z
69	SETM	SETM f	f = 0xFFFF	1	1	无
		SETM WREG	WREG = 0xFFFF	1	1	无
		SETM Ws	Ws = 0xFFFF	1	1	无
70	SFTAC	SFTAC Acc,Wn	对累加器算术移位 (Wn) 次	1	1	OA,OB,OAB, SA,SB,SAB
		SFTAC Acc,#Slit6	对累加器算术移位 Slit6 次	1	1	OA,OB,OAB, SA,SB,SAB
71	SL	SL f	f = 左移 f	1	1	C,N,OV,Z
		SL f,WREG	WREG = 左移 f	1	1	C,N,OV,Z
		SL Ws,Wd	Wd = 左移 Ws	1	1	C,N,OV,Z
		SL Wb,Wns,Wnd	Wnd = 将 Wb 左移 Wns 位	1	1	N,Z
		SL Wb,#lit5,Wnd	Wnd = 将 Wb 左移 lit5 位	1	1	N,Z
72	SUB	SUB Acc	从累加器减去	1	1	OA,OB,OAB, SA,SB,SAB
		SUB f	f = f - WREG	1	1	C,DC,N,OV,Z
		SUB f,WREG	WREG = f - WREG	1	1	C,DC,N,OV,Z
		SUB #lit10,Wn	Wn = Wn - lit10	1	1	C,DC,N,OV,Z
		SUB Wb,Ws,Wd	Wd = Wb - Ws	1	1	C,DC,N,OV,Z
		SUB Wb,#lit5,Wd	Wd = Wb - lit5	1	1	C,DC,N,OV,Z
73	SUBB	SUBB f	f = f - WREG - (C)	1	1	C,DC,N,OV,Z
		SUBB f,WREG	WREG = f - WREG - (C)	1	1	C,DC,N,OV,Z
		SUBB #lit10,Wn	Wn = Wn - lit10 - (C)	1	1	C,DC,N,OV,Z
		SUBB Wb,Ws,Wd	Wd = Wb - Ws - (C)	1	1	C,DC,N,OV,Z
		SUBB Wb,#lit5,Wd	Wd = Wb - lit5 - (C)	1	1	C,DC,N,OV,Z
74	SUBR	SUBR f	f = WREG - f	1	1	C,DC,N,OV,Z
		SUBR f,WREG	WREG = WREG - f	1	1	C,DC,N,OV,Z
		SUBR Wb,Ws,Wd	Wd = Ws - Wb	1	1	C,DC,N,OV,Z
		SUBR Wb,#lit5,Wd	Wd = lit5 - Wb	1	1	C,DC,N,OV,Z
75	SUBBR	SUBBR f	f = WREG - f - (C)	1	1	C,DC,N,OV,Z
		SUBBR f,WREG	WREG = WREG - f - (C)	1	1	C,DC,N,OV,Z
		SUBBR Wb,Ws,Wd	Wd = Ws - Wb - (C)	1	1	C,DC,N,OV,Z
		SUBBR Wb,#lit5,Wd	Wd = lit5 - Wb - (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	释放堆栈帧	1	1	无
82	XOR	XOR f	f = f .XOR.WREG	1	1	N,Z
		XOR f,WREG	WREG = f .XOR.WREG	1	1	N,Z
		XOR #lit10,Wn	Wd = lit10 .XOR.Wd	1	1	N,Z
		XOR Wb,Ws,Wd	Wd = Wb .XOR.Ws	1	1	N,Z
		XOR Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N,Z
83	ZE	ZE Ws,Wnd	Wnd = 零扩展后的 Ws	1	1	C,Z,N

22.0 开发支持

一系列硬件及软件开发工具对 PIC[®] 单片机提供支持：

- 集成开发环境
 - MPLAB[®] IDE 软件
- 汇编器 / 编译器 / 链接器
 - MPASM[™] 汇编器
 - MPLAB C18 和 MPLAB C30 C 编译器
 - MPLINK[™] 目标链接器 /
MPLIB[™] 目标库管理器
 - MPLAB ASM30 汇编器 / 链接器 / 库
- 模拟器
 - MPLAB SIM 软件模拟器
- 仿真器
 - MPLAB ICE 2000 在线仿真器
 - MPLAB REAL ICE[™] 在线仿真器
- 在线调试器
 - MPLAB ICD 2
- 器件编程器
 - PICSTART[®] Plus 开发编程器
 - MPLAB PM3 器件编程器
 - PICkit[™] 2 开发编程器
- 低成本演示和开发板及评估工具包

22.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16 位单片机市场提供了前所未有的易于使用的软件开发平台。MPLAB IDE 是基于 Windows[®] 操作系统的应用软件，包括：

- 一个包含所有调试工具的图形界面
 - 模拟器
 - 编程器（单独销售）
 - 仿真器（单独销售）
 - 在线调试器（单独销售）
- 具有彩色上下文代码显示的全功能编辑器
- 多项目管理器
- 内容可直接编辑的可定制式数据窗口
- 高级源代码调试
- 可视化器件初始化程序，便于进行寄存器的初始化
- 鼠标停留在变量上进行查看的功能
- 通过拖放把变量从源代码窗口拉到观察窗口
- 丰富的在线帮助
- 集成了可选的第三方工具，如 HI-TECH 软件 C 编译器和 IAR C 编译器

MPLAB IDE 可以让您：

- 编辑源文件（汇编语言或 C 语言）
- 点击一次即可完成汇编（或编译）并将代码下载到 PIC MCU 仿真器和模拟器工具中（自动更新所有项目信息）
- 可使用如下各项进行调试：
 - 源文件（汇编语言或 C 语言）
 - 混合汇编语言和 C 语言
 - 机器码

MPLAB IDE 在单个开发范例中支持使用多种调试工具，包括从成本效益高的模拟器到低成本的在线调试器，再到全功能的仿真器。这样缩短了用户升级到更加灵活而功能更强大的工具时的学习时间。

22.2 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于所有的 PIC MCU。

MPASM 汇编器可生成用于 MPLINK 目标链接器的可重定位目标文件、Intel® 标准 HEX 文件、详细描述存储器使用状况和符号参考的 MAP 文件、包含源代码行及生成机器码的绝对 LST 文件以及用于调试的 COFF 文件。

MPASM 汇编器具有如下特征：

- 集成在 MPLAB IDE 项目中
- 用户定义的宏可简化汇编代码
- 对多用途源文件进行条件汇编
- 允许完全控制汇编过程的指令

22.3 MPLAB C18 和 MPLAB C30 C 编译器

MPLAB C18 和 MPLAB C30 代码开发系统是完全的 ANSI C 编译器，分别适用于 Microchip 的 PIC18 和 PIC24 系列单片机及 dsPIC30F 和 dsPIC33 系列数字信号控制器。这些编译器可提供其他编译器并不具备的强大集成功能和出众的代码优化能力，且使用方便。

为便于源代码调试，编译器提供了针对 MPLAB IDE 调试器的优化符号信息。

22.4 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器包含了由 MPASM 汇编器、MPLAB C18 C 编译器产生的可重定位目标。通过使用链接器脚本中的指令，它还可链接预编译库中的可重定位目标。

MPLIB 目标库管理器管理预编译代码库文件的创建和修改。当从源文件调用库中的一段子程序时，只有包含此子程序的模块被链接到应用中。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

22.5 MPLAB ASM30 汇编器、链接器和库管理器

MPLAB ASM30 汇编器为 dsPIC30F 器件提供转换自符号汇编语言的可重定位机器码。MPLAB C30 C 编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特征：

- 支持整个 dsPIC30F 指令集
- 支持定点数据和浮点数据
- 命令行界面
- 丰富的指令集
- 灵活的宏语言
- MPLAB IDE 兼容性

22.6 MPLAB SIM 软件模拟器

MPLAB SIM 软件模拟器在指令级对 PIC MCU 和 dsPIC® DSC 进行模拟，使得用户可以在 PC 主机的环境下进行代码开发。对于任何给定的指令，用户均可对数据区进行检查或修改，并通过各种触发机制来产生激励。可以将各寄存器的情况记录在文件中，以便进行进一步地运行时分析。跟踪缓冲器和逻辑分析器的显示使模拟器还能记录和跟踪程序的执行、I/O 的动作、大部分的外设及内部寄存器的状况。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C18 和 MPLAB C30 C 编译器以及 MPASM 和 MAPLAB ASM30 汇编器的符号调试。该软件模拟器可用于在硬件实验室环境外灵活地开发和调试代码，是一款完美且经济的软件开发工具。

22.7 MPLAB ICE 2000 高性能在线仿真器

MPLAB ICE 2000 在线仿真器旨在为产品开发工程师提供一整套用于 PIC 单片机的设计工具。MPLAB ICE 2000 在线仿真器的软件控制由 MPLAB 集成开发环境平台提供，它允许在单一环境下进行编辑、编译、下载以及源代码调试。

MPLAB ICE 2000 是全功能仿真器系统，它具有增强的跟踪、触发和数据监控功能。处理器模块可插拔，使系统可轻松进行重新配置以适应各种不同处理器的仿真需要。MPLAB ICE 2000 在线仿真器的架构允许对其进行扩展以支持新的 PIC 单片机。

MPLAB ICE 2000 在线仿真器系统设计为一款实时仿真系统，该仿真系统具备通常只有昂贵的开发工具中才有的高级功能。选择 PC 平台和 Microsoft® Windows® 32 位操作系统可使这些功能在一个简单而统一的应用中得到很好的利用。

22.8 MPLAB REAL ICE 在线仿真器系统

MPLAB REAL ICE 在线仿真器系统是 Microchip 针对其闪存 DSC® 和 MCU 器件而推出的新一代高速仿真器。结合 MPLAB 集成开发环境（IDE）所具有的易于使用且功能强大的图形用户界面，该仿真器可对 PIC® 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。

22.9 MPLAB ICD 2 在线调试器

Microchip 的在线调试器 MPLAB ICD 2 是一款功能强大而成本低廉的运行时开发工具，通过 RS-232 或高速 USB 接口与 PC 主机相连。该工具基于闪存 PIC MCU，可用于开发本系列及其他 PIC MCU 和 dsPIC DSC。MPLAB ICD 2 使用了闪存器件中内建的在线调试功能。该功能结合 Microchip 的在线串行编程（In-Circuit Serial Programming™，ICSP™）协议，可在 MPLAB 集成开发环境的图形用户界面上提供成本效益很高的在线闪存调试。这使设计人员可通过设置断点、单步运行以及对变量、CPU 状态以及外设寄存器进行监视的方法实现源代码的开发和调试。其全速运行特性可对硬件和应用进行实时测试。MPLAB ICD 2 还可用作某些 PIC 器件的开发编程器。

22.10 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款通用的、符合 CE 规范的器件编程器，其可编程电压设置在 VDDMIN 和 VDDMAX 之间时可靠性最高。它有一个用来显示菜单和错误信息的大 LCD 显示器（128 x 64），以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP™ 电缆。在单机模式下，MPLAB PM3 器件编程器不必与 PC 相连即可对 PIC 器件进行读取、验证和编程。在该模式下它还可设置代码保护。MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。MPLAB PM3 具备高速通信能力以及优化算法，可对存储器很大的器件进行快速编程，它还采用 SD/MMC 卡用作文件存储及数据安全应用。

22.11 PICSTART Plus 开发编程器

PICSTART Plus 开发编程器是一款易于使用而成本低廉的原型编程器。它通过 COM (RS-232) 端口与 PC 相连。MPLAB 集成开发环境软件使得该编程器的使用简便、高效。PICSTART Plus 开发编程器支持采用 DIP 封装的大部分 PIC 器件，其引脚数最多可达 40 个。引脚数更多的器件，如 PIC16C92X 和 PIC17C76X，可通过连接一个转接插槽来获得支持。PICSTART Plus 开发编程器符合 CE 规范。

22.12 PICkit 2 开发编程器

PICkit™ 2 开发编程器是一个低成本编程器；对于某些选定闪存器件，它也是一个调试器，通过其易于使用的接口可对众多 Microchip 的低档、中档和 PIC18F 系列闪存单片机进行编程。PICkit 2 入门工具包中包含一个有实验布线区的开发板、十二堂系列课程、软件和 HI-TECH 的 PICC™ Lite C 编译器，有助于用户快速掌握 PIC® 单片机的使用。这一工具包为使用 Microchip 功能强大的中档闪存系列单片机进行编程、评估和应用开发，提供了所需的一切。

22.13 演示、开发和评估板

有许多演示、开发和评估板可用于各种 PIC MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于测试和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、RS-232 接口、LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 和 dsPICDEM™ 演示 / 开发板系列电路外，Microchip 还有一系列评估工具包和演示软件，适用于模拟滤波器设计、KEELOQ® 数据安全产品 IC、CAN、IrDA®、PowerSmart® 电池管理、SEEVAL® 评估系统、Σ-Δ 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)。

23.0 电气特性

本章将对 dsPIC30F 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 dsPIC30F 系列器件的绝对最大额定值。器件长时间工作在最大额定值条件下可能会影响其可靠性。我们不建议使器件在或超过本规范指定的最大额定值条件下运行。

绝对最大额定值^(†)

环境温度.....	-40°C 至 +125°C
储存温度.....	-65°C 至 +150°C
任一引脚相对于 Vss 的电压（除 VDD 和 MCLR 外） ⁽¹⁾	-0.3V 至 (VDD + 0.3V)
VDD 引脚相对于 Vss 的电压.....	-0.3V 至 +5.5V
MCLR 引脚相对于 Vss.....	0V 至 +13.25V
流出 VSS 引脚的最大电流.....	300 mA
流入 VDD 引脚的最大电流 ⁽²⁾	250 mA
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 引脚的最大输出灌电流	25 mA
任一 I/O 引脚的最大输出拉电流	25 mA
所有端口的最大灌电流	200 mA
所有端口的最大拉电流 ⁽²⁾	200 mA

注 1: MCLR/VPP 引脚上的尖峰电压低于 Vss，导致感应电流超过 80 mA 时，可能引起锁死。因此，当把一个“低”电平加到 MCLR/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 Vss。

2: 允许的最大电流由器件最大功耗决定。有关 P_{DMAX} 的信息，请参见表 23-2。

[†]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。

注：指定了所有外设的电气特性。有关特定器件的具体外设，请参见相应系列的对照表。

23.1 直流特性

表 23-1: 工作 MIPS 与电压

V _{DD} 范围	温度范围	最大 MIPS		
		dsPIC30F501X-30I	dsPIC30F501X-20I	dsPIC30F501X-20E
4.75-5.5V	-40°C 至 85°C	30	20	—
4.75-5.5V	-40°C 至 125°C	—	—	20
3.0-3.6V	-40°C 至 85°C	15	10	—
3.0-3.6V	-40°C 至 125°C	—	—	10
2.5-3.0V	-40°C 至 85°C	7.5	7.5	—

dsPIC30F5011/5013

表 23-2: 热工作条件

额定值	符号	最小值	典型值	最大值	单位
dsPIC30F501x-30I 工作结温范围 工作环境温度范围	T _J T _A	-40 -40	— —	+125 +85	°C °C
dsPIC30F501x-20I 工作结温范围 工作环境温度范围	T _J T _A	-40 -40	— —	+150 +85	°C °C
dsPIC30F501x-20E 工作结温范围 工作环境温度范围	T _J T _A	-40 -40	— —	+150 +125	°C °C
功耗: 内部芯片功耗: $P_{INT} = V_{DD} \times (I_{DD} - \sum I_{OH})$	P _D	$P_{INT} + P_{I/O}$			W
I/O 引脚功耗: $P_{I/O} = \sum (\{V_{DD} - V_{OH}\} \times I_{OH}) + \sum (V_{OL} \times I_{OL})$					
最大允许功耗	P _{DMAX}	$(T_J - T_A) / \theta_{JA}$			W

表 23-3: 热封装特性

特性	符号	典型值	最大值	单位	注
封装热阻, 64 引脚 TQFP (10x10x1mm)	θ _{JA}	39	—	C/W	1
封装热阻, 80 引脚 TQFP (12x12x1mm)	θ _{JA}	39	—	C/W	1

注 1: 通过封装模拟获得结点与环境的热阻值 θ_{JA}。

表 23-4: 直流温度和电压规范

直流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明) 工作温度				
参数 编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
工作电压⁽²⁾							
DC10	V _{DD}	供电电压	2.5	—	5.5	V	工业级温度
DC11	V _{DD}	供电电压	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}	V _{DD} 启动电压 (确保内部上电复位信号)	—	V _{SS}	—	V	
DC17	V _{VDD}	V _{DD} 上升率 (确保内部上电复位信号)	0.05	—	—	V/ms	0-5V/0.1 s 0-3V/60 ms

注 1: 除非另外声明, 否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。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 23-5: 直流特性: 工作电流 (IDD)

直流特性		标准工作条件: 2.5V 至 5.5V (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⁾						
DC30a	7.3	11	mA	25°C	3.3V	FRC (-2 MIPS)
DC30b	7.5	11.2	mA	85°C		
DC30c	7.6	11.4	mA	125°C		
DC30e	12.9	19.2	mA	25°C	5V	LPRC (-512 Hz)
DC30f	12.8	19.1	mA	85°C		
DC30g	12.8	19.1	mA	125°C		
DC31a	1.9	2.8	mA	25°C	3.3V	4 MIPS
DC31b	2.0	3	mA	85°C		
DC31c	2.0	3	mA	125°C		
DC31e	4.1	6.1	mA	25°C	5V	10 MIPS
DC31f	4.0	6	mA	85°C		
DC31g	3.8	5.7	mA	125°C		
DC23a	13.5	20	mA	25°C	3.3V	20 MIPS
DC23b	14	21	mA	85°C		
DC23c	15	22.5	mA	125°C		
DC23e	23	34.5	mA	25°C	5V	30 MIPS
DC23f	23.5	35	mA	85°C		
DC23g	24	36	mA	125°C		
DC24a	32	48	mA	25°C	3.3V	4 MIPS
DC24b	32.5	49	mA	85°C		
DC24c	33	49.5	mA	125°C		
DC24e	53.5	80	mA	25°C	5V	10 MIPS
DC24f	54	81	mA	85°C		
DC24g	54	81	mA	125°C		
DC27a	60	90	mA	25°C	3.3V	20 MIPS
DC27b	61	91.5	mA	85°C		
DC27d	101	152	mA	25°C		
DC27e	100	150	mA	85°C	5V	30 MIPS
DC27f	100	150	mA	125°C		
DC29a	145	217	mA	25°C		
DC29b	144	216	mA	85°C	5V	4 MIPS

注 1: 除非另外声明, 否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

2: 供电电流主要受工作电压和频率的影响。其他因素如 I/O 引脚负载和开关速率、振荡器类型、内部代码执行模式以及温度也对电流消耗有影响。所有 IDD 测量的测试条件为: OSC1 使用满幅的外部方波进行驱动。所有 I/O 引脚配置为输入且被拉到 VDD。MCLR = VDD, WDT, FSCM, LVD 和 BOR 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。

dsPIC30F5011/5013

表 23-6: 直流特性: 空闲电流 (I_{IDLE})

直流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明) 工作温度				
参数编号	典型值 ^(1,2)	最大值	单位	条件			
空闲电流 (I_{IDLE}): 内核不工作、时钟工作时的基本电流⁽²⁾							
DC50a	4.8	7.2	mA	25°C	3.3V	FRC (~2MIPS)	
DC50b	4.9	7.3	mA	85°C			
DC50c	5.0	7.5	mA	125°C	5V		
DC50e	8.9	13.3	mA	25°C			
DC50f	8.8	13.2	mA	85°C			
DC50g	8.8	13.2	mA	125°C			
DC51a	1.6	2.4	mA	25°C	3.3V	LPRC (~512 kHz)	
DC51b	1.62	2.43	mA	85°C			
DC51c	1.62	2.43	mA	125°C	5V		
DC51e	3.65	5.47	mA	25°C			
DC51f	3.4	5.1	mA	85°C			
DC51g	3.3	4.95	mA	125°C			
DC43a	8.5	12.75	mA	25°C	3.3V	4 MIPS EC 模式, 4X PLL	
DC43b	8.7	13	mA	85°C			
DC43c	9.6	14.4	mA	125°C			
DC43e	15.2	22.8	mA	25°C	5V		
DC43f	15.2	22.8	mA	85°C			
DC43g	15.2	22.8	mA	125°C			
DC44a	19.9	29.8	mA	25°C	3.3V	10 MIPS EC 模式, 4X PLL	
DC44b	20.2	30.3	mA	85°C			
DC44c	20.5	30.7	mA	125°C			
DC44e	33.4	50	mA	25°C	5V		
DC44f	33.7	50.5	mA	85°C			
DC44g	34	51	mA	125°C			
DC47a	37.4	56	mA	25°C	3.3V	20 MIPS EC 模式, 8X PLL	
DC47b	38	57	mA	85°C			
DC47d	62.3	93.4	mA	25°C	5V		
DC47e	62.9	94.3	mA	85°C			
DC47f	63.5	95.2	mA	125°C			
DC49a	90.8	136	mA	25°C	5V	30 MIPS EC 模式, 16X PLL	
DC49b	91	137	mA	85°C			

注 1: 除非另外声明, 否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

2: 基本 I_{IDLE} 电流的测量是在内核不工作、时钟工作而所有外设模块关闭的条件下进行的。

表 23-7： 直流特性：掉电电流 (IPD)

直流特性		标准工作条件: 2.5V 至 5.5V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)					
参数编号	典型值 ⁽¹⁾	最大值	单位	条件			
掉电电流 (IPD) ⁽²⁾							
DC60a	5	25	μA	25°C	3.3V	基本掉电电流 ⁽³⁾	
DC60b	8	40	μA	85°C			
DC60c	14	70	μA	125°C			
DC60e	8	40	μA	25°C	5V	看门狗定时器电流: ΔI _{WDT} ⁽³⁾	
DC60f	12	55	μA	85°C			
DC60g	20	100	μA	125°C			
DC61a	7.8	12	μA	25°C	3.3V		
DC61b	7.9	12	μA	85°C			
DC61c	8.4	13	μA	125°C			
DC61e	15.4	23.1	μA	25°C	5V	Timer1, 带 32 kHz 晶振: ΔIT _{I32} ⁽³⁾	
DC61f	14.7	22	μA	85°C			
DC61g	14.1	21.1	μA	125°C			
DC62a	3.8	6	μA	25°C	3.3V	Timer1, 带 32 kHz 晶振: ΔIT _{I32} ⁽³⁾	
DC62b	—	—	μA	85°C			
DC62c	—	—	μA	125°C			
DC62e	5.5	10	μA	25°C	5V	BOR 工作: ΔI _{BOR} ⁽³⁾	
DC62f	—	—	μA	85°C			
DC62g	—	—	μA	125°C			
DC63a	31.5	47.2	μA	25°C	3.3V	BOR 工作: ΔI _{BOR} ⁽³⁾	
DC63b	34.4	51.5	μA	85°C			
DC63c	36.5	55	μA	125°C			
DC63e	36.5	54.7	μA	25°C	5V	低电压检测: ΔI _{LVD} ⁽³⁾	
DC63f	39.1	58.7	μA	85°C			
DC63g	40.5	61	μA	125°C			
DC66a	19.6	29.4	μA	25°C	3.3V	低电压检测: ΔI _{LVD} ⁽³⁾	
DC66b	21.5	32.3	μA	85°C			
DC66c	23	34.5	μA	125°C			
DC66e	24	36	μA	25°C	5V		
DC66f	25.5	38.3	μA	85°C			
DC66g	26.2	39	μA	125°C			

注 1: 除非另外声明，否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

2: 基本 IPD 是在所有外设和时钟都关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。LVD、BOR 和 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。

dsPIC30F5011/5013

表 23-8： 直流特性：I/O 引脚输入规范

直流特性			标准工作条件：2.5V 至 5.5V (除非另外声明)				
参数编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
DI10 DI15 DI16 DI17 DI18 DI19	VIL	输入低电压 ⁽²⁾ I/O 引脚： 带施密特触发缓冲器	Vss	—	0.2 VDD	V	
		MCLR	Vss	—	0.2 VDD	V	
		OSC1 (在 XT、HS 和 LP 模式下)	Vss	—	0.2 VDD	V	
		OSC1 (在 RC 模式下) ⁽³⁾	Vss	—	0.3 VDD	V	
		SDA 和 SCL	Vss	—	0.3 VDD	V	禁止 SM 总线
		SDA 和 SCL	Vss	—	0.2 VDD	V	使能 SM 总线
DI20 DI25 DI26 DI27 DI28 DI29	VIH	输入高电压 ⁽²⁾ I/O 引脚： 带施密特触发缓冲器	0.8 VDD	—	VDD	V	
		MCLR	0.8 VDD	—	VDD	V	
		OSC1 (在 XT、HS 和 LP 模式下)	0.7 VDD	—	VDD	V	
		OSC1 (在 RC 模式下) ⁽³⁾	0.9 VDD	—	VDD	V	
		SDA 和 SCL	0.7 VDD	—	VDD	V	禁止 SM 总线
		SDA 和 SCL	0.8 VDD	—	VDD	V	使能 SM 总线
DI30	ICNPU	CNxx 上拉电流 ⁽²⁾	50	250	400	μA	VDD = 5V, VPIN = VSS
DI50 DI51 DI55 DI56	IIL	输入泄漏电流 ⁽²⁾⁽⁴⁾⁽⁵⁾ I/O 端口	—	0.01	±1	μA	VSS ≤ VPIN ≤ VDD, 引脚处于高阻态
		模拟输入引脚	—	0.50	—	μA	VSS ≤ VPIN ≤ VDD, 引脚处于高阻态
		MCLR	—	0.05	±5	μA	VSS ≤ VPIN ≤ VDD
		OSC1	—	0.05	±5	μA	VSS ≤ VPIN ≤ VDD, XT、HS 和 LP 振荡器模式

注 1：除非另外声明，否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。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：在 RC 振荡器配置中，OSC1/CLK1 引脚为施密特触发器输入。不建议在 RC 模式下使用外部时钟驱动 dsPIC30F 器件。

4：MCLR 引脚上的泄漏电流主要取决于所施加电压。规定电压为正常工作条件下的电压。在不同的输入电压下可能测得更高的泄漏电流。

5：负电流定义为引脚的拉电流。

表 23-9: 直流特性: I/O 引脚输出规范

直流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明)				
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 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)
DO10	VOL	输出低电压 ⁽²⁾ I/O 端口	—	—	0.6	V	IOL = 8.5 mA, VDD = 5V
					TBD	V	IOL = 2.0 mA, VDD = 3V
DO16		OSC2/CLKOUT (RC 或 EC 振荡器模式)	—	—	0.6	V	IOL = 1.6 mA, VDD = 5V
					TBD	V	IOL = 2.0 mA, VDD = 3V
DO20	VOH	输出高电压 ⁽²⁾ I/O 端口	VDD - 0.7	—	—	V	IOH = -3.0 mA, VDD = 5V
					TBD	V	IOH = -2.0 mA, VDD = 3V
DO26		OSC2/CLKOUT (RC 或 EC 振荡器模式)	VDD - 0.7	—	—	V	IOH = -1.3 mA, VDD = 5V
					TBD	V	IOH = -2.0 mA, VDD = 3V
DO50	Cosc 2	输出引脚上的容性负载规范 ⁽²⁾ OSC2/SOSC2 引脚	—	—	15	pF	当外部时钟用于驱动 OSC1 时 处于 XTL、XT、HS 和 LP 模 式下。
					—	pF	
DO56	CIO	所有 I/O 引脚和 OSC2	—	—	50	pF	RC 或 EC 振荡器模式
DO58	CB	SCL 和 SDA	—	—	400	pF	在 I ² C 模式下

注 1: 除非另外声明, 否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。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。

图 23-1: 低电压检测特性

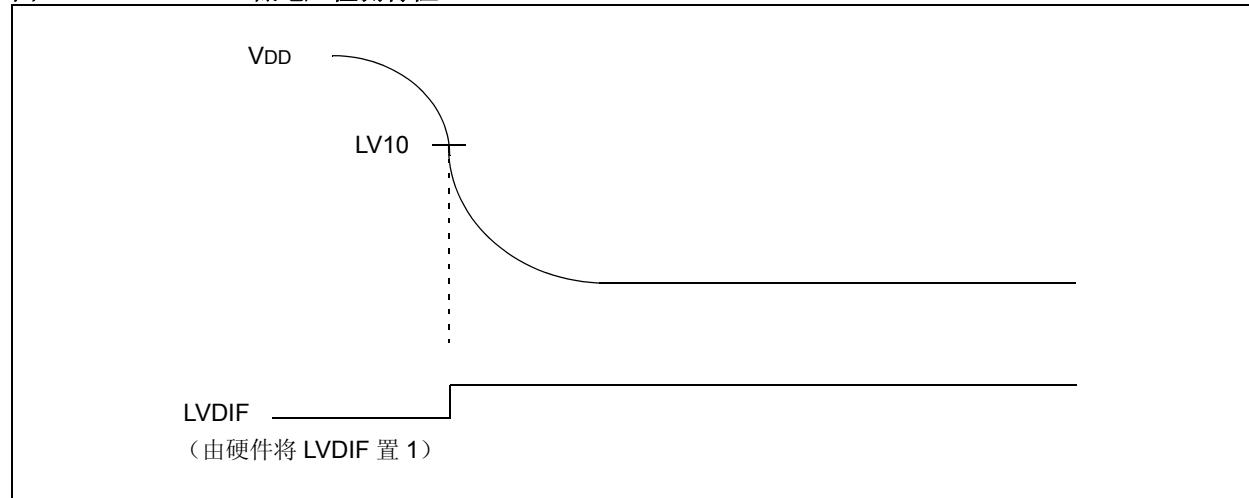


表 23-10: 电气特性: LVDL

直流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明) 工作温度				
参数编号	符号	特性 ⁽¹⁾	最小值	典型值	最大值	单位	条件
LV10	VPLVD	当 VDD 从高电平翻转至低电平时的 LVDL 电压	LVDL = 0000 ⁽²⁾	—	—	—	V
			LVDL = 0001 ⁽²⁾	—	—	—	V
			LVDL = 0010 ⁽²⁾	—	—	—	V
			LVDL = 0011 ⁽²⁾	—	—	—	V
			LVDL = 0100	2.50	—	2.65	V
			LVDL = 0101	2.70	—	2.86	V
			LVDL = 0110	2.80	—	2.97	V
			LVDL = 0111	3.00	—	3.18	V
			LVDL = 1000	3.30	—	3.50	V
			LVDL = 1001	3.50	—	3.71	V
			LVDL = 1010	3.60	—	3.82	V
			LVDL = 1011	3.80	—	4.03	V
			LVDL = 1100	4.00	—	4.24	V
			LVDL = 1101	4.20	—	4.45	V
			LVDL = 1110	4.50	—	4.77	V
LV15	VLVDIN	外部 LVD 输入引脚门限电压	LVDL = 1111	—	—	—	V

注 1: 这些参数为特性值, 但生产时未经测试。

2: 这些值不在可用工作电压范围内。

图 23-2: 欠压复位特性

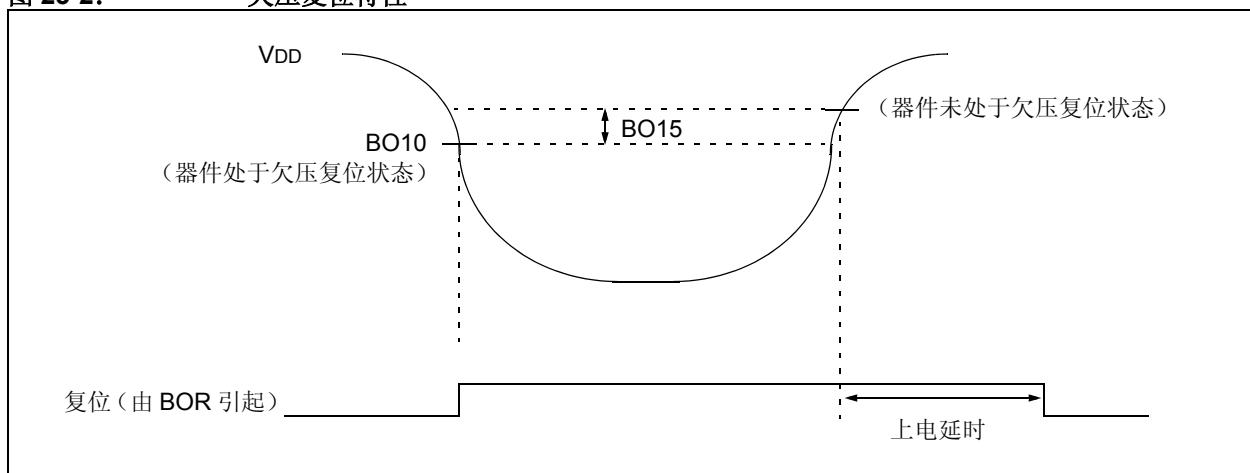


表 23-11：电气特性：BOR

直流特性		标准工作条件：2.5V 至 5.5V (除非另外声明)						
参数编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件	
BO10	VBOR	当 VDD 从高电平翻转至低电平时的 BOR 电压 ⁽²⁾	BORV = 11 ⁽³⁾	—	—	—	V	不在工作电压范围内
			BORV = 10	2.60	—	2.71	V	
			BORV = 01	4.10	—	4.40	V	
			BORV = 00	4.58	—	4.73	V	
BO15	VBHYS			—	5	—	mV	

注 1：除非另外声明，否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。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。

3：11 值不在可用工作电压范围内。

表 23-12：直流特性：程序和 EEPROM

直流特性		标准工作条件：2.5V 至 5.5V (除非另外声明)						
参数编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件	
D120	ED	数据 EEPROM 存储器 ⁽²⁾	100K	1M	—	E/W	-40°C ≤ TA ≤ +85°C	
D121	VDRW	字节耐擦写能力 读 / 写操作时的 VDD	VMIN	—	5.5	V	使用 EECON 读 / 写 VMIN = 最小工作电压	
D122	TDEW	擦 / 写周期时间	—	2	—	ms		
D123	TRETD	特性保持时间	40	100	—	年	假设没有违反其他规范	
D124	IDEW	编程时的 IDD	—	10	30	mA	行擦除	
D130	EP	闪存程序存储器 ⁽²⁾	10K	100K	—	E/W	-40°C ≤ TA ≤ +85°C	
D131	VPR	单元耐擦写能力 读操作时的 VDD	VMIN	—	5.5	V	VMIN = 最小工作电压	
D132	VEB	批量擦除时的 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		
D135	TRETD	特性保持时间	40	100	—	年	假设没有违反其他规范	
D136	TEB	ICSP 块擦除时间	—	4	—	ms		
D137	IPEW	编程时的 IDD	—	10	30	mA	行擦除	
D138	IEB	编程时的 IDD	—	10	30	mA	批量擦除	

注 1：除非另外声明，否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。

2：这些参数为特性值，但生产时未经测试。

dsPIC30F5011/5013

23.2 交流特性和时序参数

本节包含的信息说明了 dsPIC30F 系列器件的交流特性和时序参数。

表 23-13： 温度和电压规范——交流

交流特性	标准工作条件: 2.5V 至 5.5V (除非另外声明)
	工作温度 $-40^{\circ}\text{C} \leq \text{TA} \leq +85^{\circ}\text{C}$ (工业级) $-40^{\circ}\text{C} \leq \text{TA} \leq +125^{\circ}\text{C}$ (扩展级)
	工作电压 VDD 范围如直流规范第 23.0 节中所介绍。

图 23-3： 器件时序规范的负载条件

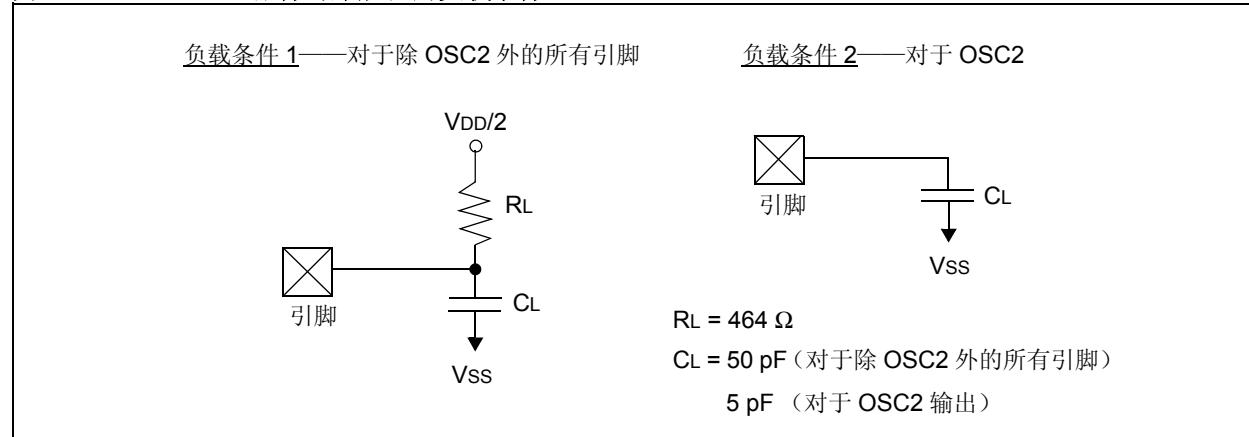


图 23-4： 外部时钟时序

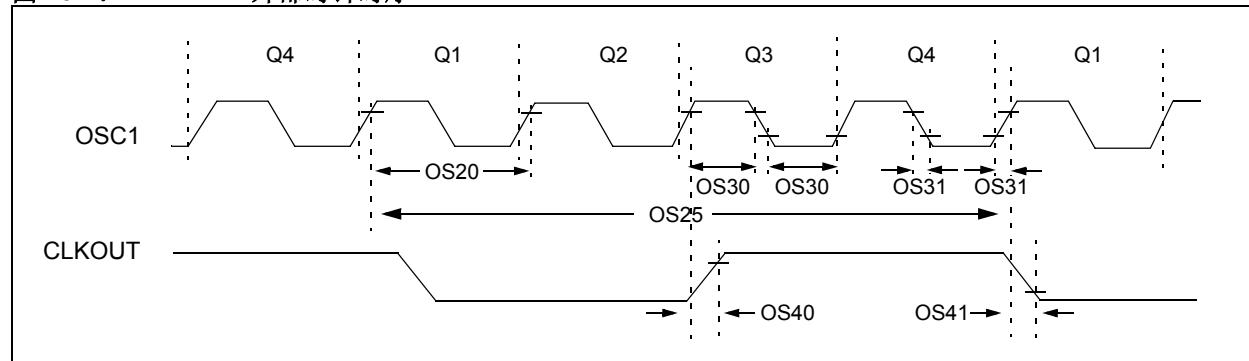


表 23-14: 外部时钟时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明)				
参数编号	符号	特性	最小值	典型值 ⁽¹⁾	最大值	单位	条件
OS10	Fosc	外部 CLKIN 频率 ⁽²⁾ (外部时钟仅允许运行于 EC 模式)	DC 4 4 4	— — — —	40 10 10 7.5	MHz MHz MHz MHz	EC 带 4x PLL 的 EC 带 8x PLL 的 EC 带 16x PLL 的 EC
		振荡器频率 ⁽²⁾	DC 0.4 4 4 4 4 10	— — — — — — —	4 4 10 10 10 7.5 25	MHz MHz MHz MHz MHz MHz MHz	RC XTL XT 带 4x PLL 的 XT 带 8x PLL 的 XT 带 16x PLL 的 XT HS LP
		Tosc = 1/Fosc	—	—	—	—	Fosc 值见参数 OS10
OS20	Tcy	指令周期时间 ⁽²⁾⁽³⁾	33	—	DC	ns	见表 23-17
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	CLKOUT 上升时间 ⁽²⁾⁽⁴⁾	—	—	—	ns	见参数 D031
OS41	TckF	CLKOUT 下降时间 ⁽²⁾⁽⁴⁾	—	—	—	ns	见参数 D032

- 注 1: 除非另外声明, 否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。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: 指令周期时间 (Tcy) 等于输入振荡器时基周期的 4 倍。所有规定值均为基于针对特定振荡器类型, 器件在标准工作条件下执行代码时的特性数据。超出这些规定的限定值, 可能导致振荡器运行不稳定和 / 或导致电流消耗超出预期值。所有器件在测试 “最小” 值时, 都在 OSC1/CLKI 引脚连接了外部时钟。当使用了外部时钟输入时, 所有器件的 “最大” 周期时间限制为 “DC” (无时钟)。
- 4: 测量在 RC 或 ERC 模式下进行。在 OSC2 引脚上测量 CLKOUT 信号。CLKOUT 在 Q1-Q2 周期 (1/2 Tcy 中为低电平, 在 Q3-Q4 周期 (1/2 Tcy) 中为高电平。

dsPIC30F5011/5013

表 23-15: PLL 时钟时序规范 ($V_{DD} = 2.5$ 至 5.5 V)

交流特性			标准工作条件: 2.5V 至 5.5V (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	FPLL1	PLL 输入频率范围 ⁽²⁾	4	—	10	MHz	带 PLL 的 EC、XT 和 FRC 模式
OS51	FSYS	片上 PLL 输出 ⁽²⁾	16	—	120	MHz	带 PLL 的 EC、XT 和 FRC 模式
OS52	TLOC	PLL 起振时间 (锁定时间)	—	20	50	μs	

注 1: 这些参数为特性值, 但生产时未经测试。

2: 除非另外声明, 否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

表 23-16: PLL 抖动

交流特性			标准工作条件: 2.5V 至 5.5V (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 ⁽¹⁾	最大值	单位	条件	
OS61	x4 PLL	—	0.251	0.413	%	-40C ≤ TA ≤ +85°C	$V_{DD} = 3.0$ 至 $3.6V$
		—	0.251	0.413	%	-40C ≤ TA ≤ +125°C	$V_{DD} = 3.0$ 至 $3.6V$
		—	0.256	0.47	%	-40C ≤ TA ≤ +85°C	$V_{DD} = 4.5$ 至 $5.5V$
		—	0.256	0.47	%	-40C ≤ TA ≤ +125°C	$V_{DD} = 4.5$ 至 $5.5V$
	x8 PLL	—	0.355	0.584	%	-40C ≤ TA ≤ +85°C	$V_{DD} = 3.0$ 至 $3.6V$
		—	0.355	0.584	%	-40C ≤ TA ≤ +125°C	$V_{DD} = 3.0$ 至 $3.6V$
		—	0.362	0.664	%	-40C ≤ TA ≤ +85°C	$V_{DD} = 4.5$ 至 $5.5V$
		—	0.362	0.664	%	-40C ≤ TA ≤ +125°C	$V_{DD} = 4.5$ 至 $5.5V$
	x16 PLL	—	0.67	0.92	%	-40C ≤ TA ≤ +85°C	$V_{DD} = 3.0$ 至 $3.6V$
		—	0.632	0.956	%	-40C ≤ TA ≤ +85°C	$V_{DD} = 4.5$ 至 $5.5V$
		—	0.632	0.956	%	-40C ≤ TA ≤ +125°C	$V_{DD} = 4.5$ 至 $5.5V$

注 1: 这些参数为特性值, 但生产时未经测试。

表 23-17： 内部时钟时序示例

时钟 振荡器模式	Fosc (MHz) (1)	Tcy (μs) (2)	MIPS ⁽³⁾ 无 PLL	MIPS ⁽³⁾ PLL x4	MIPS ⁽³⁾ PLL x8	MIPS ⁽³⁾ PLL x16
EC	0.200	20.0	0.05	—	—	—
	4	1.0	1.0	4.0	8.0	16.0
	10	0.4	2.5	10.0	20.0	—
	25	0.16	6.25	—	—	—
XT	4	1.0	1.0	4.0	8.0	16.0
	10	0.4	2.5	10.0	20.0	—

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: $T_{CY} = 1 / \text{MIPS}$ 。

3: 指令执行频率: $\text{MIPS} = (\text{Fosc} * \text{PLLx}) / 4$ (因为每个指令周期有 4 个 Q 时钟)。

表 23-18： 交流特性: 内部 FRC 抖动、精度和漂移⁽¹⁾

交流特性		标准工作条件: 2.5V 至 5.5V (除非另外声明)					
参数 编号	特性	最小值	典型值	最大值	单位	条件	
FRC 频率 = 7.37 MHz⁽²⁾ 时的内部 FRC 抖动							
OS62	FRC	—	± 0.04	± 0.16	%	$-40C \leq TA \leq +85C$	VDD = 3.0-3.6V
		—	± 0.07	± 0.23	%	$-40C \leq TA \leq +125C$	VDD = 4.5-5.5V
	带 4x PLL 的 FRC	—	± 0.31	± 0.62	%	$-40C \leq TA \leq +85C$	VDD = 3.0-3.6V
		—	± 0.34	± 0.77	%	$-40C \leq TA \leq +125C$	VDD = 4.5-5.5V
	带 8x PLL 的 FRC	—	± 0.44	± 0.87	%	$-40C \leq TA \leq +85C$	VDD = 3.0-3.6V
		—	± 0.48	± 1.08	%	$-40C \leq TA \leq +125C$	VDD = 4.5-5.5V
FRC 频率 = 7.37 MHz⁽²⁾ 时的内部 FRC 精度							
OS63	FRC	—	—	± 1.50	%	$-40C \leq TA \leq +125C$	VDD = 3.0-5.5V
FRC 频率 = 7.37 MHz⁽²⁾ 时的内部 FRC 漂移							
OS64		-0.7	—	0.5	%	$-40C \leq TA \leq +85C$	VDD = 3.0-3.6V
		-0.7	—	0.7	%	$-40C \leq TA \leq +125C$	VDD = 3.0-3.6V
		-0.7	—	0.5	%	$-40C \leq TA \leq +85C$	VDD = 4.5-5.5V
		-0.7	—	0.7	%	$-40C \leq TA \leq +125C$	VDD = 4.5-5.5V

注 1: FRC 总的偏差可通过将抖动、精度和漂移百分比的绝对值相加来计算。

2: 在 7.372 MHz $\pm 2\%$ 、25°C 和 5V 条件下进行频率校准。TUN 位 (OSCCON<3:0>) 可用来补偿温度漂移。

dsPIC30F5011/5013

表 23-19: 内部 LPRC 精度

交流特性		标准工作条件: 2.5V 至 5.5V (除非另外声明)				
参数 编号	特性	最小值	典型值	最大值	单位	条件
频率 = 512 kHz ⁽¹⁾ 时的 LPRC						
OS65		-20	—	+40	%	—

注 1: LPRC 频率将随 VDD 的变化而变化。

图 23-5: CLKOUT 和 I/O 时序特性

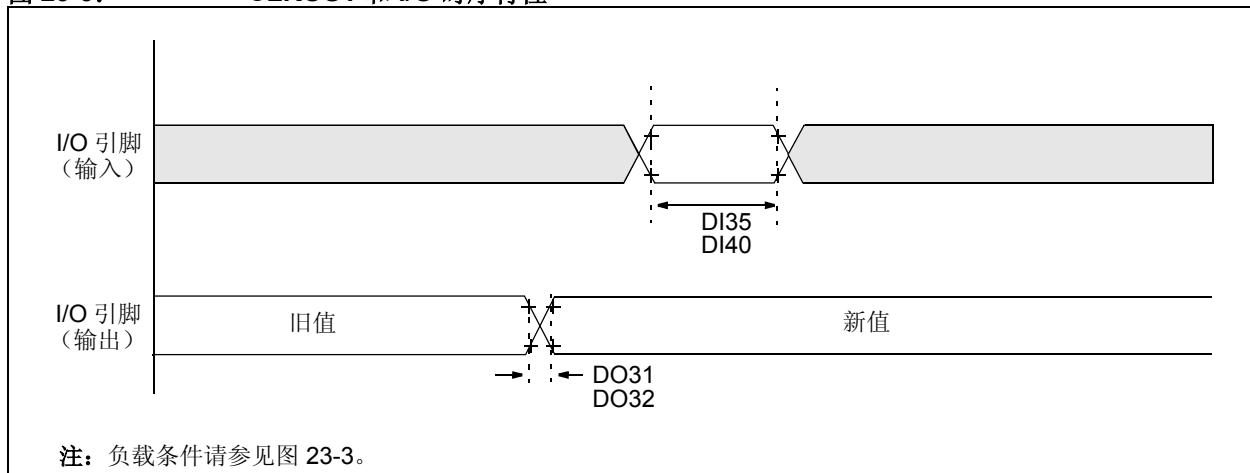


表 23-20: CLKOUT 和 I/O 时序要求

交流特性		标准工作条件: 2.5V 至 5.5V (除非另外声明)					
参数 编号	符号	特性 ⁽¹⁾⁽²⁾⁽³⁾	最小值	典型值 ⁽⁴⁾	最大值	单位	条件
DO31	T _{IO} R	端口输出上升时间	—	7	20	ns	—
DO32	T _{IO} F	端口输出下降时间	—	7	20	ns	—
DI35	T _{INP}	INTx 引脚高电平或低电平时间 (输出)	20	—	—	ns	—
DI40	T _{RB} P	CNx 高电平或低电平时间 (输入)	2 T _{CY}	—	—	ns	—

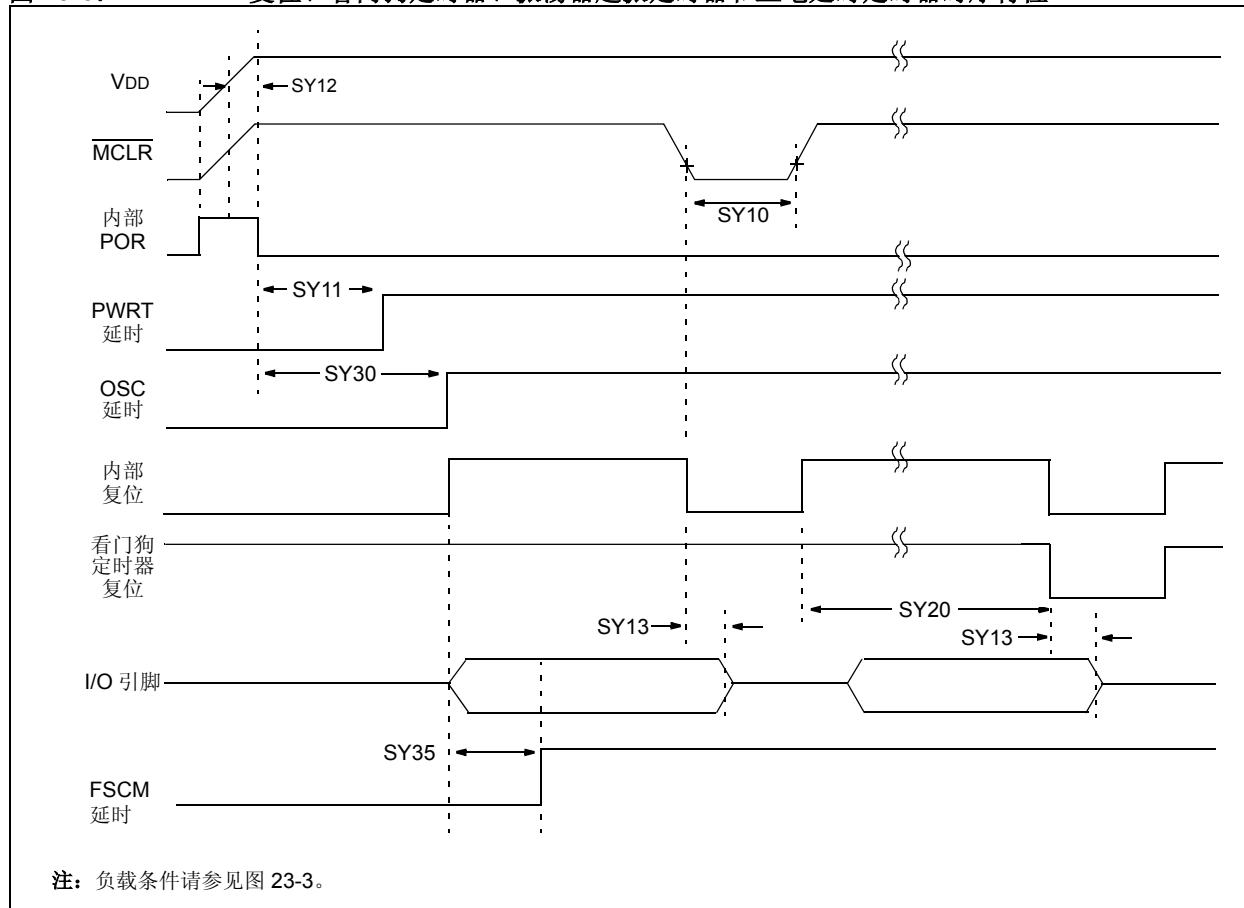
注 1: 这些参数是与任何内部时钟边沿无关的异步事件。

2: 测量是在 RC 模式和 EC 模式下进行的, 其中 CLKOUT 输出为 4 × T_{Osc}。

3: 这些参数为特性值, 但生产时未经测试。

4: 除非另外声明, 否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。

图 23-6：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



dsPIC30F5011/5013

表 23-21：复位、看门狗定时器、振荡器起振定时器、上电延时定时器和欠压复位时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明)				
参数 编号	符号	特性 ⁽¹⁾	最小值	典型值 ⁽²⁾	最大值	单位	条件
SY10	TmcL	MCLR 脉冲宽度 (低电平)	2	—	—	μs	-40°C 至 +85°C
SY11	TPWRT	上电延时定时器周期	3 12 50	4 16 64	6 22 90	ms	-40°C 至 +85°C 用户可编程
SY12	TPOR	上电复位延时	3	10	30	μs	-40°C 至 +85°C
SY13	TIOZ	自 MCLR 低电平或看门狗定时器复位起 I/O 处于高阻态的时间	—	0.8	1.0	μs	
SY20	TWDT1	看门狗定时器超时周期 (无预分频器)	1.6	2.2	3.0	ms	VDD = 5V, -40°C 至 +85°C
	TWDT2		1.7	2.2	3.1	ms	VDD = 3V, -40°C 至 +85°C
SY25	TBOR	欠压复位脉冲宽度 ⁽³⁾	100	—	—	μs	VDD ≤ VBOR (D034)
SY30	TOST	振荡器起振定时器周期	—	1024 TOSC	—	—	TOSC = OSC1 周期
SY35	TFSCM	故障保护时钟监视器延时	—	500	900	μs	-40°C 至 +85°C

注 1: 这些参数为特性值, 但生产时未经测试。

2: 除非另外声明, 否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。

3: 有关 BOR 的信息, 请参见图 23-2 和表 23-11。

图 23-7: 带隙启动时间特性

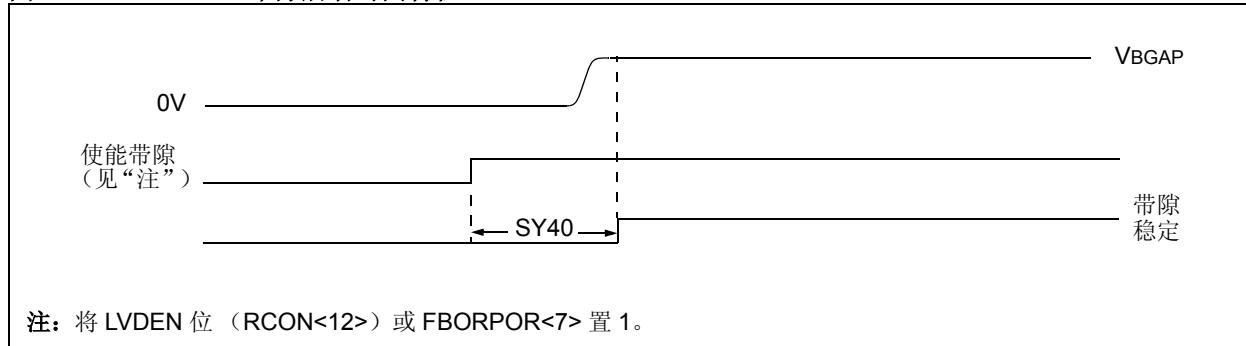


表 23-22: 带隙启动时间要求

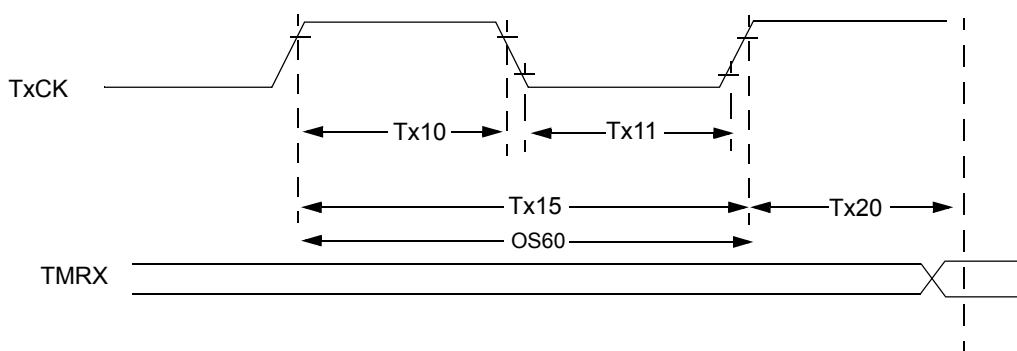
交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明)				
参数 编号	符号	特性 ⁽¹⁾	最小值	典型值 ⁽²⁾	最大 值	单位	条件
SY40	TBGAP	带隙启动时间	—	40	65	μs	定义为带隙功能使能和带隙参考电压稳定之间的时间。RCON<13> 状态位

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” 栏中的数据均为 5V 和 25°C 条件下的值。

dsPIC30F5011/5013

图 23-8: A 类、B 类和 C 类定时器外部时钟时序特性



注: 负载条件请参见图 23-3。

表 23-23: A 类定时器 (TIMER1) 外部时钟时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明)				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
TA10	TTxH	TxCK 高电平时间	同步, 无预分频器	0.5 TCY + 20	—	—	ns
			同步, 带预分频器	10	—	—	ns
			异步	10	—	—	ns
TA11	TTXL	TxCK 低电平时间	同步, 无预分频器	0.5 TCY + 20	—	—	ns
			同步, 带预分频器	10	—	—	ns
			异步	10	—	—	ns
TA15	TTxP	TxCK 输入周期	同步, 无预分频器	TCY + 10	—	—	ns
			同步, 带预分频器	取如下二者 中较大值: 20 ns 或 (TCY + 40)/N	—	—	—
			异步	20	—	—	ns
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	TCKEXTM RL	从外部 TxCK 时钟边沿到定时器递增之间的延时	0.5 TCY	—	1.5 TCY	—	

注: Timer1 属于 A 类定时器。

表 23-24: B 类定时器 (TIMER2 和 TIMER4) 外部时钟时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
TB10	TtxH	TxCK 高电平时间	同步, 无预分频器	0.5 TCY + 20	—	—	ns
			同步, 带预分频器	10	—	—	ns
TB11	TtxL	TxCK 低电平时间	同步, 无预分频器	0.5 TCY + 20	—	—	ns
			同步, 带预分频器	10	—	—	ns
TB15	TtxP	TxCK 输入周期	同步, 无预分频器	TCY + 10	—	—	ns
			同步, 带预分频器	取如下二者 中较大值: 20 ns 或 (TCY + 40)/N			
TB20	TCKEXTMRL	从外部 TxCK 时钟边沿到定时器递增之间的延时	0.5 TCY	—	1.5 TCY	—	—

注: Timer2 和 Timer4 属于 B 类定时器。

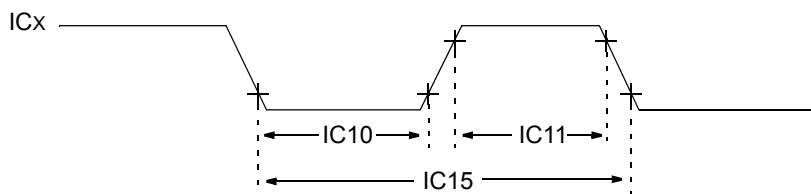
表 23-25: C 类定时器 (TIMER3 和 TIMER5) 外部时钟时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
TC10	TtxH	TxCK 高电平时间	同步	0.5 TCY + 20	—	—	ns
TC11	TtxL	TxCK 低电平时间	同步	0.5 TCY + 20	—	—	ns
TC15	TtxP	TxCK 输入周期	同步, 无预分频器	TCY + 10	—	—	ns
			同步, 带预分频器	取如下二者 中较大值: 20 ns 或 (TCY + 40)/N			
TC20	TCKEXTMRL	从外部 TxCK 时钟边沿到定时器递增之间的延时	0.5 TCY	—	1.5 TCY	—	—

注: Timer3 和 Timer5 属于 C 类定时器。

dsPIC30F5011/5013

图 23-9: 输入捕捉 (CAPx) 时序特性



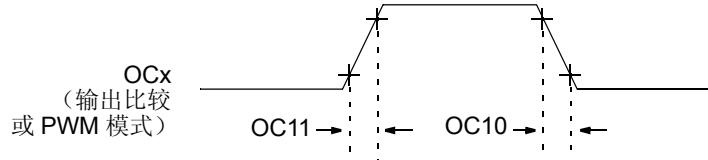
注: 负载条件请参见图 23-3。

表 23-26: 输入捕捉时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (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 或 16)

注 1: 这些参数为特性值, 但生产时未经测试。

图 23-10: 输出比较模块 (OCx) 时序特性



注：负载条件请参见图 23-3。

表 23-27: 输出比较模块时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (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”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

dsPIC30F5011/5013

图 23-11: 输出比较 /PWM 模块时序特性

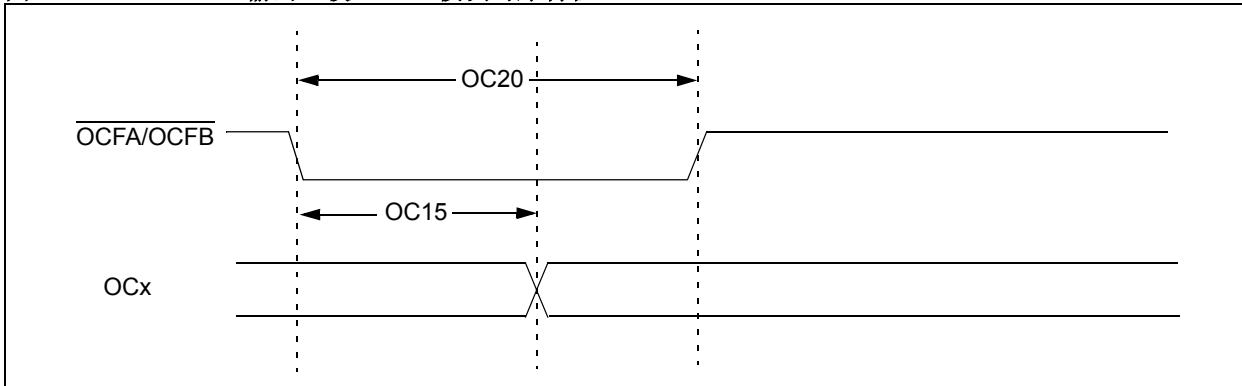


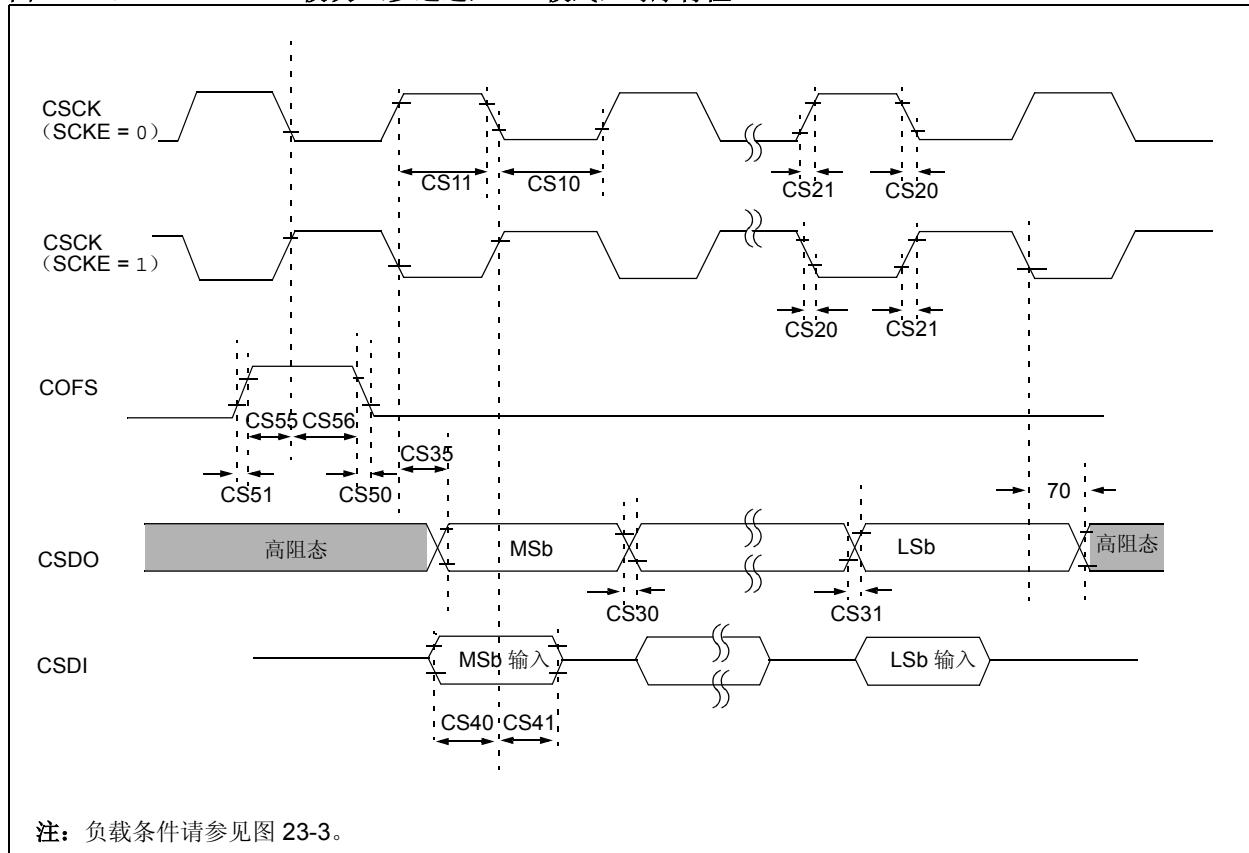
表 23-28: 简单输出比较 /PWM 模式时序要求

交流特性				标准工作条件: 2.5V 至 5.5V (除非另外声明)			
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	故障输入到 PWM I/O 发生变化的时间	—	—	50	ns	—
OC20	TFLT	故障输入脉冲宽度	50	—	—	ns	—

注 1: 这些参数为特性值, 但生产时未经测试。

2: 除非另外声明, 否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

图 23-12: DCI 模块 (多通道, I²S 模式) 时序特性



dsPIC30F5011/5013

表 23-29: DCI 模块 (多通道, I²S 模式) 时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明)				
参数编号	符号	特性 ⁽¹⁾	最小值	典型值 ⁽²⁾	最大值	单位	条件
CS10	TcsCKL	CSCK 输入低电平时间 (CSCK 引脚为输入)	T _{CY} / 2 + 20	—	—	ns	—
		CSCK 输出低电平时间 ⁽³⁾ (CSCK 引脚为输出)	30	—	—	ns	—
CS11	TcsCKH	CSCK 输入高电平时间 (CSCK 引脚为输入)	T _{CY} / 2 + 20	—	—	ns	—
		CSCK 输出高电平时间 ⁽³⁾ (CSCK 引脚为输出)	30	—	—	ns	—
CS20	TcsCKF	CSCK 输出下降时间 ⁽⁴⁾ (CSCK 引脚为输出)	—	10	25	ns	—
CS21	TcsCKR	CSCK 输出上升时间 ⁽⁴⁾ (CSCK 引脚为输出)	—	10	25	ns	—
CS30	TcsDOF	CSDO 数据输出下降时间 ⁽⁴⁾	—	10	25	ns	—
CS31	TcsDOR	CSDO 数据输出上升时间 ⁽⁴⁾	—	10	25	ns	—
CS35	TDV	时钟边沿到 CSDO 数据有效的时间	—	—	10	ns	—
CS36	TDIV	时钟边沿到 CSDO 三态的时间	10	—	20	ns	—
CS40	TCSDI	CSDI 数据输入到 CSCK 边沿的建立时间 (CSCK 引脚为输入或输出)	20	—	—	ns	—
CS41	THCSDI	CSDI 数据输入到 CSCK 边沿的保持时间 (CSCK 引脚为输入或输出)	20	—	—	ns	—
CS50	TcoFSF	COFS 下降时间 (COFS 引脚为输出)	—	10	25	ns	注 1
CS51	TcoFSR	COFS 上升时间 (COFS 引脚为输出)	—	10	25	ns	注 1
CS55	TscoFS	COFS 数据输入到 CSCK 边沿的建立时间 (COFS 引脚为输入)	20	—	—	ns	—
CS56	THCOFS	COFS 数据输入到 CSCK 边沿的保持时间 (COFS 引脚为输入)	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” 栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

3: CSCK 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 DCI 引脚上的负载均为 50 pF。

图 23-13: DCI 模块 (AC-LINK 模式) 时序特性

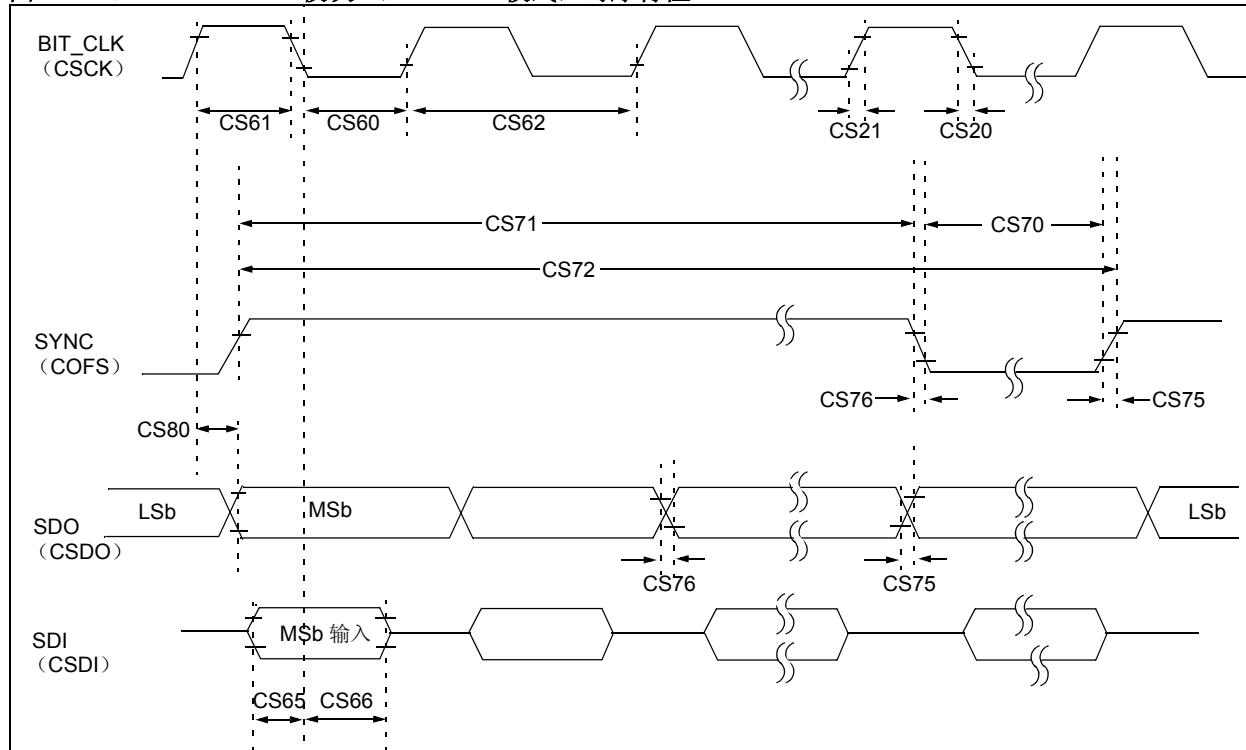


表 23-30: DCI 模块 (AC-LINK 模式) 时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明)				
参数编号	符号	特性 ⁽¹⁾⁽²⁾	最小值	典型值 ⁽³⁾	最大值	单位	条件
CS60	TBCLKL	BIT_CLK 低电平时间	36	40.7	45	ns	—
CS61	TBCLKH	BIT_CLK 高电平时间	36	40.7	45	ns	—
CS62	TBCLK	BIT_CLK 周期	—	81.4	—	ns	位时钟为输入
CS65	TsACL	到 BIT_CLK 下降沿的输入建立时间	—	—	10	ns	—
CS66	THACL	自 BIT_CLK 下降沿起的输入保持时间	—	—	10	ns	—
CS70	TSYNCLO	同步数据输出低电平时间	—	19.5	—	μs	注 1
CS71	TSYNCHI	同步数据输出高电平时间	—	1.3	—	μs	注 1
CS72	TSYNC	同步数据输出周期	—	20.8	—	μs	注 1
CS75	TRACL	上升时间, 同步, SDATA_OUT	—	10	25	ns	CLOAD = 50 pF, VDD = 5V
CS76	TFACL	下降时间, 同步, SDATA_OUT	—	10	25	ns	CLOAD = 50 pF, VDD = 5V
CS77	TRACL	上升时间, 同步, SDATA_OUT	—	10	25	ns	CLOAD = 50 pF, VDD = 3V
CS78	TFACL	下降时间, 同步, SDATA_OUT	—	10	25	ns	CLOAD = 50 pF, VDD = 3V
CS80	ToVDACL	自 BIT_CLK 上升沿起的输出有效延时	—	—	15	ns	—

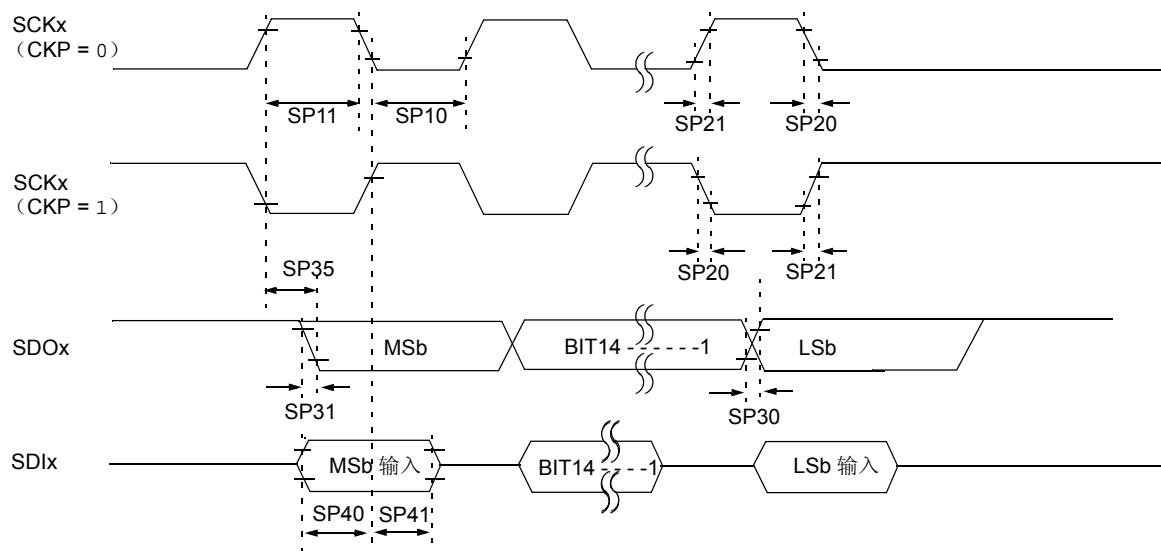
注 1: 这些参数为特性值, 但生产时未经测试。

2: 这些值假设 BIT_CLK 频率为 12.288 MHz。

3: 除非另外声明, 否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

dsPIC30F5011/5013

图 23-14: SPI 模块主模式 ($CKE = 0$) 时序特性



注: 负载条件请参见图 23-3。

表 23-31: SPI 主模式 ($CKE = 0$) 时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明) 工作温度 $-40^{\circ}\text{C} \leq \text{TA} \leq +85^{\circ}\text{C}$ (工业级) $-40^{\circ}\text{C} \leq \text{TA} \leq +125^{\circ}\text{C}$ (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
SP10	TscL	SCKx 输出低电平时间 ⁽³⁾	$\text{T}_{CY}/2$	—	—	ns	—
SP11	TscH	SCKx 输出高电平时间 ⁽³⁾	$\text{T}_{CY}/2$	—	—	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	在 SCKx 边沿之后 SDOx 数据输出有效的时间	—	—	30	ns	—
SP40	TdiV2sCH, TdiV2sCL	SDIx 数据输入到 SCKx 边沿的建立时间	20	—	—	ns	—
SP41	Tsch2diL, TscL2diL	SDIx 数据输入到 SCKx 边沿的保持时间	20	—	—	ns	—

注 1: 这些参数为特性值, 但生产时未经测试。

2: 除非另外声明, 否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。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 引脚上的负载均为 50 pF。

图 23-15: SPI 模块主模式 (CKE = 1) 时序特性

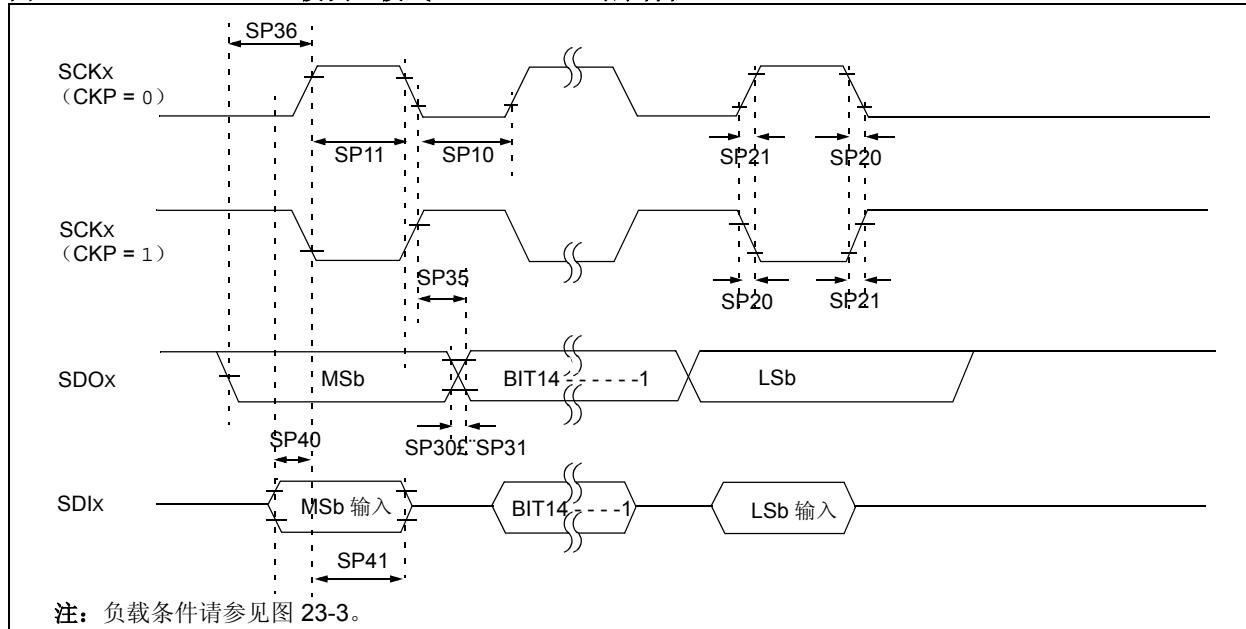


表 23-32: SPI 模块主模式 (CKE = 1) 时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明)				
参数编号	符号	特性 ⁽¹⁾	最小值	典型值 ⁽²⁾	最大值	单位	条件
SP10	TscL	SCKx 输出低电平时间 ⁽³⁾	TCY / 2	—	—	ns	—
SP11	TscH	SCKx 输出高电平时间 ⁽³⁾	TCY / 2	—	—	ns	—
SP20	TscF	SCKx 输出下降时间 ⁽⁴⁾	—	—	—	ns	见参数 DO32
SP21	TscR	SCKx 输出上升时间 ⁽⁴⁾	—	—	—	ns	见参数 DO31
SP30	TdoF	SDOx 数据输出下降时间 ⁽⁴⁾	—	—	—	ns	见参数 DO32
SP31	TdoR	SDOx 数据输出上升时间 ⁽⁴⁾	—	—	—	ns	见参数 DO31
SP35	TscH2doV, TscL2doV	在 SCKx 边沿之后 SDOx 数据输出有效的时间	—	—	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” 栏中的数据均为 5V 和 25°C 条件下的值。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, 在主模式下产生的时钟不能违反该规定。

4: 假定所有 SPI 引脚上的负载均为 50 pF。

dsPIC30F5011/5013

图 23-16: SPI 模块从模式 ($CKE = 0$) 时序特性

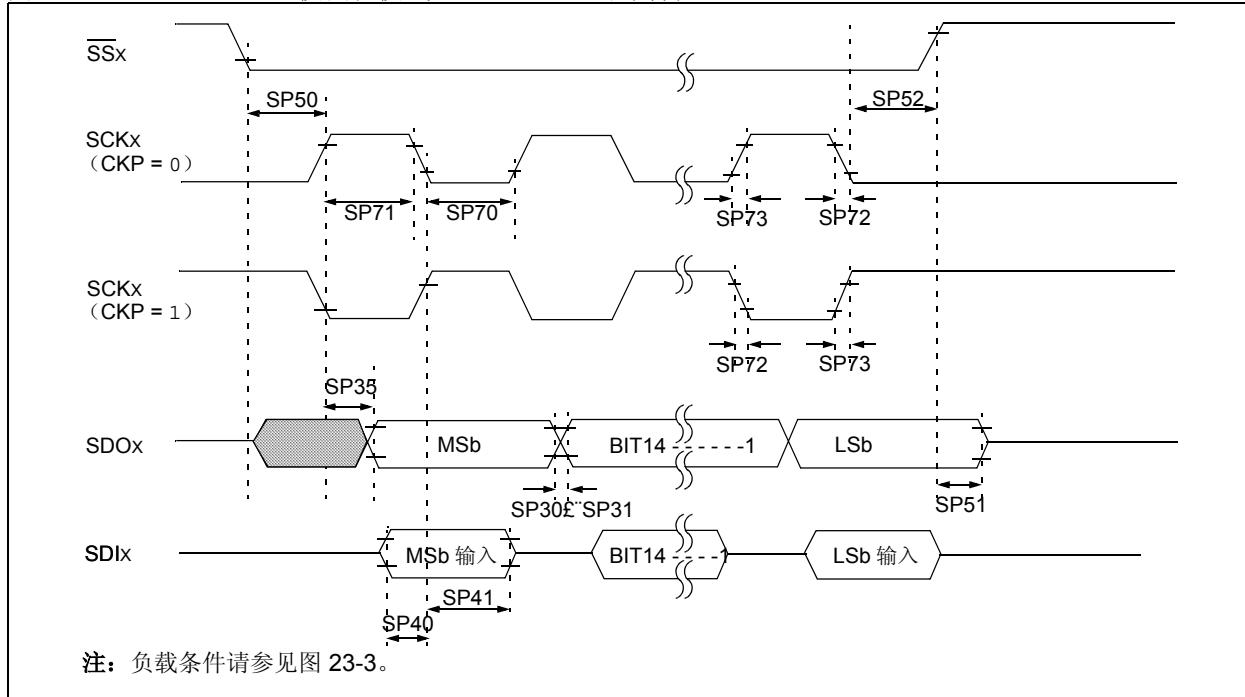


表 23-33: SPI 模块从模式 ($CKE = 0$) 时序要求

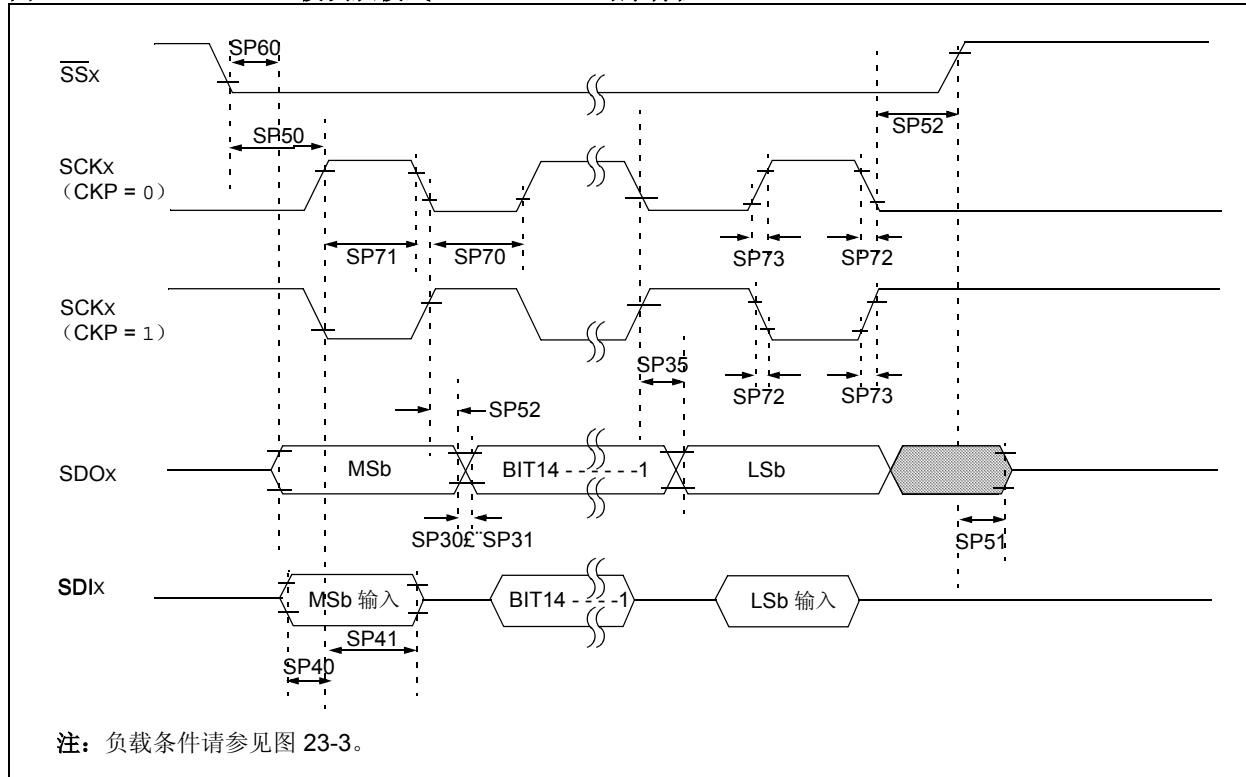
交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明)				
参数编号	符号	特性 ⁽¹⁾	最小值	典型值 ⁽²⁾	最大值	单位	条件
SP70	TscL	SCKx 输入低电平时间	30	—	—	ns	—
SP71	TscH	SCKx 输入高电平时间	30	—	—	ns	—
SP72	TscF	SCKx 输入下降时间 ⁽³⁾	—	10	25	ns	—
SP73	TscR	SCKx 输入上升时间 ⁽³⁾	—	10	25	ns	—
SP30	TdoF	SDOx 数据输出下降时间 ⁽³⁾	—	—	—	ns	见参数 DO32
SP31	TdoR	SDOx 数据输出上升时间 ⁽³⁾	—	—	—	ns	见参数 DO31
SP35	TscH2doV, TscL2doV	在 SCKx 边沿之后 SDOx 数据输出有效的时间	—	—	30	ns	—
SP40	TdiV2sch, TdiV2scl	SDIx 数据输入到 SCKx 边沿的建立时间	20	—	—	ns	—
SP41	TscH2dil, TscL2dil	SDIx 数据输入到 SCKx 边沿的保持时间	20	—	—	ns	—
SP50	TssL2sch, TssL2scl	SSx↓ 到 SCKx↑ 或 SCKx↓ 输入的时间	120	—	—	ns	—
SP51	TssH2doZ	SSx↑ 到 SDOx 输出高阻态的时间 ⁽³⁾	10	—	50	ns	—
SP52	TscH2ssH TscL2ssH	SCK 边沿后 SSx 有效的时间	1.5 Tcy +40	—	—	ns	—

注 1: 这些参数为特性值, 但生产时未经测试。

2: 除非另外声明, 否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考, 未经测试。

3: 假定所有 SPI 引脚上的负载均为 50 pF。

图 23-17: SPI 模块从模式 (CKE = 1) 时序特性



dsPIC30F5011/5013

表 23-34: SPI 模块从模式 (CKE = 1) 时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (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	在 SCKx 边沿之后 SDOx 数据输出有效的时间	—	—	30	ns	—
SP40	TdiV2sch, TdiV2scl	SDIx 数据输入到 SCKx 边沿的建立时间	20	—	—	ns	—
SP41	TscH2diL, TscL2diL	SDIx 数据输入到 SCKx 边沿的保持时间	20	—	—	ns	—
SP50	TssL2scH, TssL2scL	SSx↓ 到 SCKx↓ 或 SCKx↑ 输入的时间	120	—	—	ns	—
SP51	TssH2doZ	SS↑ 到 SDOx 输出高阻态的时间 ⁽⁴⁾	10	—	50	ns	—
SP52	TscH2ssH TscL2ssH	SCKx 边沿后 SSx↑ 有效的时间	1.5 Tcy + 40	—	—	ns	—
SP60	TssL2doV	在 SSx 边沿之后 SDOx 数据输出有效的时间	—	—	50	ns	—

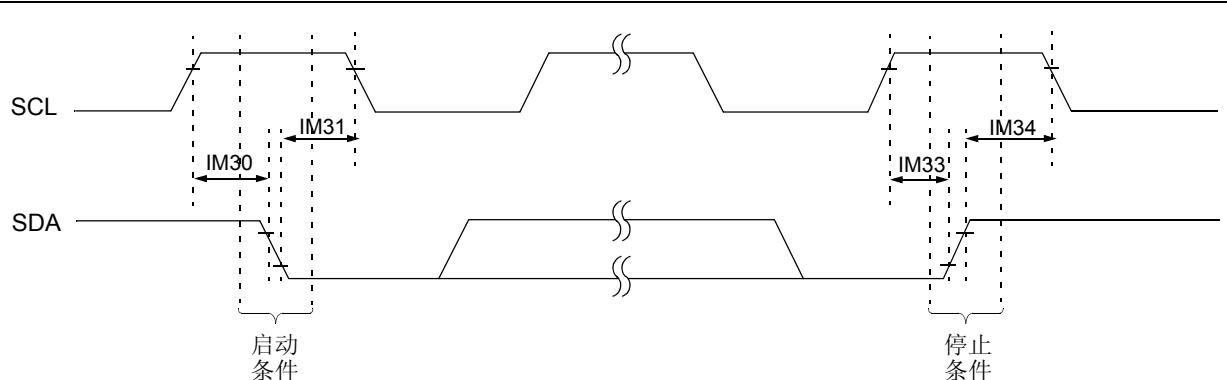
注 1: 这些参数为特性值, 但生产时未经测试。

2: 除非另外声明, 否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。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。

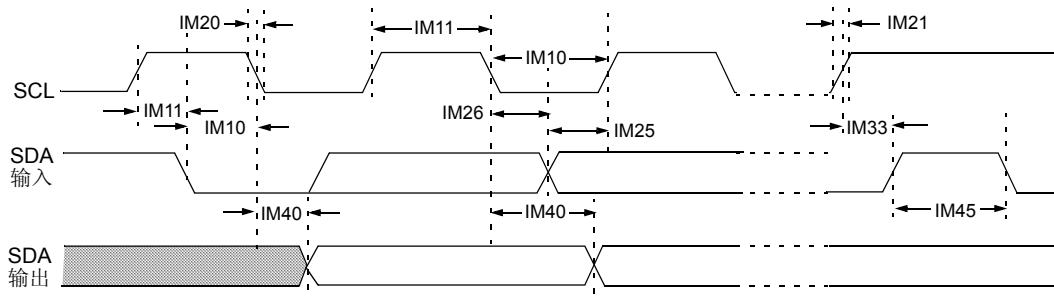
4: 假定所有 SPI 引脚上的负载均为 50 pF。

图 23-18:

I²CTM 总线启动 / 停止位时序特性（主模式）

注：负载条件请参见图 23-3。

图 23-19:

I²CTM 总线数据时序特性（主模式）

注：负载条件请参见图 23-3。

dsPIC30F5011/5013

表 23-35: I²CTM 总线数据时序要求 (主模式)

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)			
参数编号	符号	特性	最小值 ⁽¹⁾	最大值	单位	条件
IM10	TLO:SCL	时钟低电平时间	100 kHz 模式	TCY / 2 (BRG + 1)	—	s
			400 kHz 模式	TCY / 2 (BRG + 1)	—	s
			1 MHz 模式 ⁽²⁾	TCY / 2 (BRG + 1)	—	s
IM11	THI:SCL	时钟高电平时间	100 kHz 模式	TCY / 2 (BRG + 1)	—	s
			400 kHz 模式	TCY / 2 (BRG + 1)	—	s
			1 MHz 模式 ⁽²⁾	TCY / 2 (BRG + 1)	—	s
IM20	TF:SCL	SDA 和 SCL 下降时间	100 kHz 模式	—	300	ns
			400 kHz 模式	20 + 0.1 CB	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
			400 kHz 模式	20 + 0.1 CB	300	ns
			1 MHz 模式 ⁽²⁾	—	300	ns
IM25	TSU:DAT	数据输入 建立时间	100 kHz 模式	250	—	ns
			400 kHz 模式	100	—	ns
			1 MHz 模式 ⁽²⁾	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	s
			1 MHz 模式 ⁽²⁾	TBD	—	ns
IM30	TSU:STA	启动条件 建立时间	100 kHz 模式	TCY / 2 (BRG + 1)	—	s
			400 kHz 模式	TCY / 2 (BRG + 1)	—	s
			1 MHz 模式 ⁽²⁾	TCY / 2 (BRG + 1)	—	s
IM31	THD:STA	启动条件 保持时间	100 kHz 模式	TCY / 2 (BRG + 1)	—	s
			400 kHz 模式	TCY / 2 (BRG + 1)	—	s
			1 MHz 模式 ⁽²⁾	TCY / 2 (BRG + 1)	—	s
IM33	TSU:STO	停止条件 建立时间	100 kHz 模式	TCY / 2 (BRG + 1)	—	s
			400 kHz 模式	TCY / 2 (BRG + 1)	—	s
			1 MHz 模式 ⁽²⁾	TCY / 2 (BRG + 1)	—	s
IM34	THD:STO	停止条件 保持时间	100 kHz 模式	TCY / 2 (BRG + 1)	—	ns
			400 kHz 模式	TCY / 2 (BRG + 1)	—	ns
			1 MHz 模式 ⁽²⁾	TCY / 2 (BRG + 1)	—	ns
IM40	TAA:SCL	自时钟边沿到输出有效的时间	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	—	s
			400 kHz 模式	1.3	—	s
			1 MHz 模式 ⁽²⁾	TBD	—	s
IM50	CB	总线容性负载	—	400	pF	

注 1: BRG 为 I²C 波特率发生器的值。请参见《dsPIC30F 系列参考手册》(DS70046E_CN) 中的第 21 节“**I²C 模块**”。

2: 所有 I²C 引脚的最大引脚电容为 10 pF (仅对于 1 MHz 模式)。

图 23-20: I²CTM 总线启动 / 停止位时序特性（从模式）

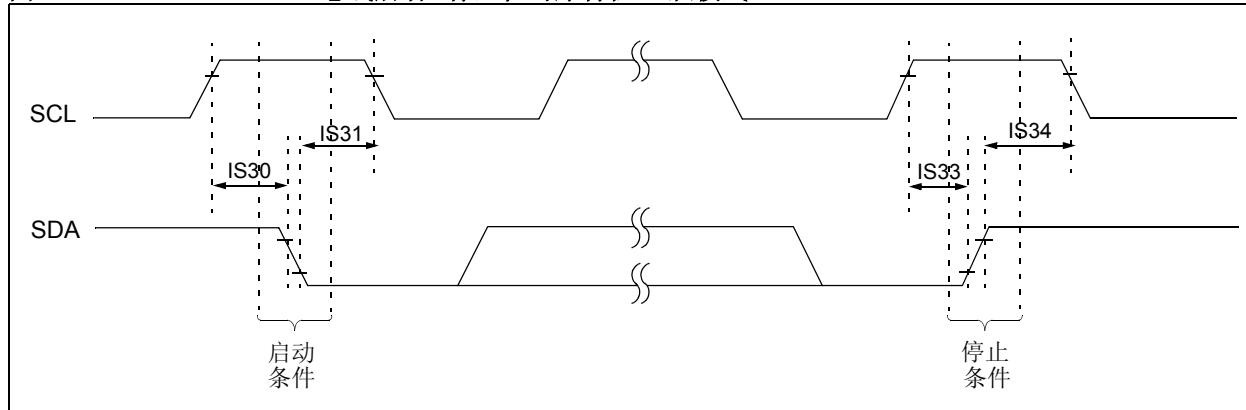
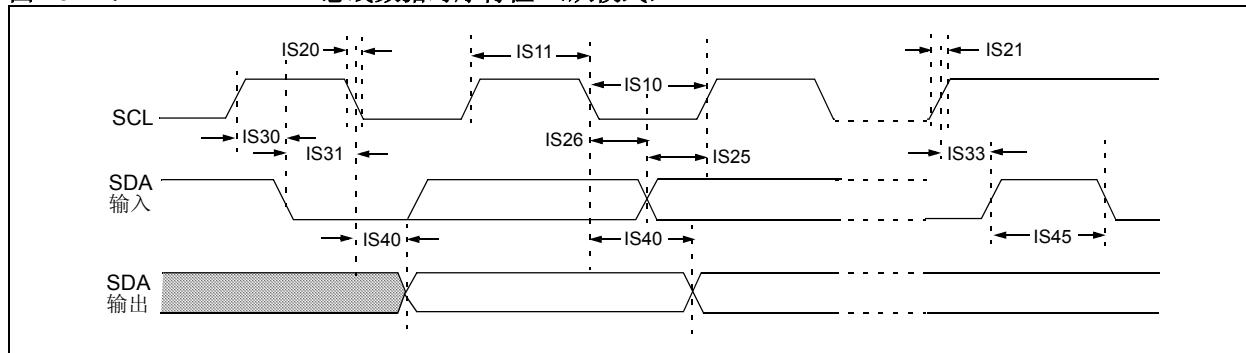


图 23-21: I²CTM 总线数据时序特性（从模式）



dsPIC30F5011/5013

表 23-36: I²CTM 总线数据时序要求 (从模式)

交流特性		标准工作条件: 2.5V 至 5.5V (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	器件工作频率不得低于 1.5 MHz
			400 kHz 模式	1.3	—	μs	器件工作频率不得低于 10 MHz
			1 MHz 模式 (1)	0.5	—	μs	—
IS11	THI:SCL	时钟高电平时间	100 kHz 模式	4.0	—	μs	器件工作频率不得低于 1.5 MHz
			400 kHz 模式	0.6	—	μs	器件工作频率不得低于 10 MHz
			1 MHz 模式 (1)	0.5	—	μs	—
IS20	TF:SCL	SDA 和 SCL 下降时间	100 kHz 模式	—	300	ns	CB 值规定在 10 至 400 pF 之间
			400 kHz 模式	20 + 0.1 CB	300	ns	
			1 MHz 模式 (1)	—	100	ns	
IS21	TR:SCL	SDA 和 SCL 上升时间	100 kHz 模式	—	1000	ns	CB 值规定在 10 至 400 pF 之间
			400 kHz 模式	20 + 0.1 CB	300	ns	
			1 MHz 模式 (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	这个周期后产生第一个时钟脉冲
			400 kHz 模式	0.6	—	μs	
			1 MHz 模式 (1)	0.25	—	μs	
IS33	TSU:STO	停止条件 建立时间	100 kHz 模式	4.7	—	μs	—
			400 kHz 模式	0.6	—	μs	
			1 MHz 模式 (1)	0.6	—	μs	
IS34	THD:STO	停止条件 保持时间	100 kHz 模式	4000	—	ns	—
			400 kHz 模式	600	—	ns	
			1 MHz 模式 (1)	250	—	ns	
IS40	TAA:SCL	自时钟边沿到输出 有效的时间	100 kHz 模式	0	3500	ns	—
			400 kHz 模式	0	1000	ns	
			1 MHz 模式 (1)	0	350	ns	
IS45	TBF:SDA	总线空闲时间	100 kHz 模式	4.7	—	μs	在启动一个新的传输前总线必须保持空闲的时间
			400 kHz 模式	1.3	—	μs	
			1 MHz 模式 (1)	0.5	—	μs	
IS50	CB	总线容性负载		—	400	pF	—

注 1: 所有 I²CTM 引脚的最大引脚电容为 10 pF (仅对于 1 MHz 模式)。

图 23-22: CAN 模块 I/O 时序特性

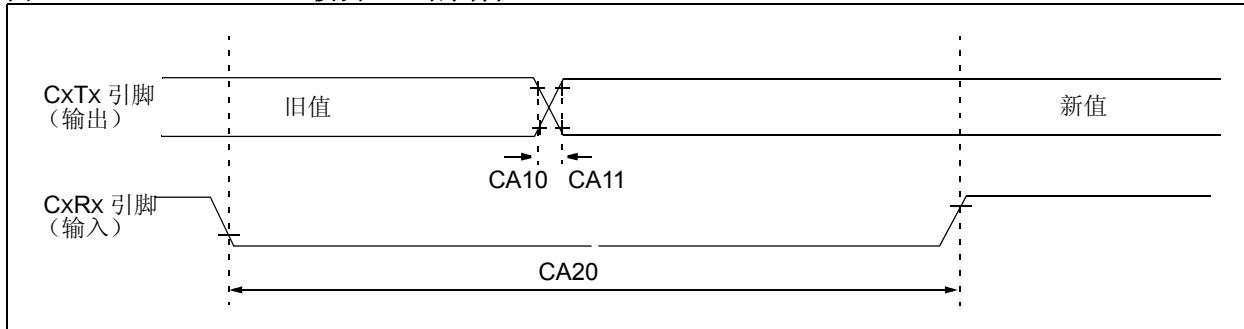


表 23-37: CAN 模块 I/O 时序要求

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明)				
参数编号	符号	特性 ⁽¹⁾	最小值	典型值 ⁽²⁾	最大值	单位	条件
CA10	TioF	端口输出下降时间	—	—	—	ns	见参数 D032
CA11	TioR	端口输出上升时间	—	—	—	ns	见参数 D031
CA20	Tcwf	触发 CAN 唤醒过滤器的脉冲宽度	500			ns	—

注 1: 这些参数为特性值，但生产时未经测试。

2: 除非另外声明，否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

dsPIC30F5011/5013

表 23-38: 12 位 A/D 模块规范

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明)				
参数 编号	符号	特性	最小值	典型值	最大值	单位	条件
器件电源							
AD01	AVDD	模块 VDD 电源	VDD - 0.3 和 2.7 中的 较大值	—	VDD + 0.3 和 5.5 中的 较小值	V	—
AD02	AVSS	模块 Vss 电源	Vss - 0.3	—	Vss + 0.3	V	—
参考输入							
AD05	VREFH	参考电压高电平	AVSS + 2.7	—	AVDD	V	—
AD06	VREFL	参考电压低电平	AVss	—	AVDD - 2.7	V	—
AD07	VREF	绝对参考电压	AVss - 0.3	—	AVDD + 0.3	V	—
AD08	IREF	电流消耗	—	150 .001	200 1	μA	A/D 工作 A/D 关闭
模拟输入							
AD10	VINH-VINL	满量程输入范围	VREFL	—	VREFH	V	见“注”
AD11	VIN	绝对输入电压	AVss - 0.3	—	AVDD + 0.3	V	—
AD12	—	泄漏电流	—	±0.001	±0.610	μA	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V 电源阻抗 = 2.5 kΩ
AD13	—	泄漏电流	—	±0.001	±0.610	μA	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V 信号源阻抗 = 2.5 kΩ
AD17	RIN	模拟信号源的推荐阻抗	—	—	2.5K	Ω	—
DC 精度							
AD20	Nr	分辨率	12 个数据位			OE ^a	—
AD21	INL	积分非线性	—	—	<±1	Lsb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD21A	INL	积分非线性	—	—	<±1	Lsb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22	DNL	微分非线性	—	—	<±1	Lsb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD22A	DNL	微分非线性	—	—	<±1	Lsb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23	GERR	增益误差	+1.25	+1.5	+3	Lsb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD23A	GERR	增益误差	+1.25	+1.5	+3	Lsb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD24	EOFF	失调误差	-2	-1.5	-1.25	Lsb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD24A	EOFF	失调误差	-2	-1.5	-1.25	Lsb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V

注 1: A/D 转换结果不会因输入电压的增加而减小，并且不会丢失编码。

表 23-38: 12 位 A/D 模块规范 (续)

交流特性			标准工作条件: 2.5V 至 5.5V (除非另外声明) 工作温度				
参数 编号	符号	特性	最小值	典型值	最大值	单位	条件
AD25	—	单调性 ⁽¹⁾	—	—	—	—	保证
动态性能							
AD30	THD	总谐波失真	—	-71	—	dB	—
AD31	SINAD	信号噪声和失真	—	68	—	dB	—
AD32	SFDR	无杂散动态范围	—	83	—	dB	—
AD33	FNYQ	输入信号带宽	—	—	100	kHz	—
AD34	ENOB	有效位数	10.95	11.1	—	位	—

注 1: A/D 转换结果不会因输入电压的增加而减小，并且不会丢失编码。

dsPIC30F5011/5013

图 23-23: 12 位 A/D 转换时序要求 (ASAM = 0, SSRC = 000)

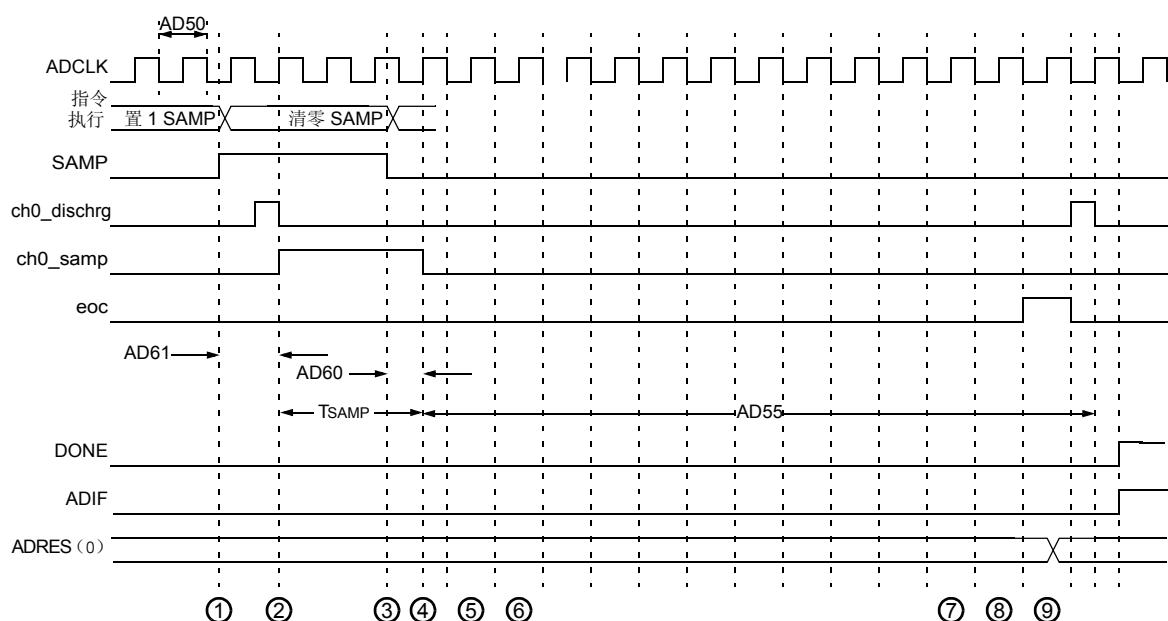


表 23-39: 12 位 A/D 转换时序要求

交流特性			标准工作条件: 2.7V 至 5.5V (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
时钟参数							
AD50	TAD	A/D 时钟周期		334	—	ns	VDD = 3-5.5V (注 1)
AD51	TRC	A/D 内部 RC 振荡器周期	1.2	1.5	1.8	μs	—
转换速率							
AD55	TCONV	转换时间	—	14 TAD		ns	—
AD56	FCNV	吞吐率	—	—	200	kspS	VDD = VREF = 5V
AD57	TsAMP	采样时间	—	1 TAD	—	ns	VDD = 3-5.5V 源阻抗 RS = 0-2.5 kΩ
时序参数							
AD60	TPCS	从触发采样到启动转换的时间	—	1 TAD	—	ns	—
AD61	TPSS	从采样位 (SAMP) 置 1 到采样启动的时间	0.5 TAD	—	1.5 TAD	ns	—
AD62	Tcss	转换结束到采样启动 (ASAM = 1) 的时间	—	0.5 TAD	—	ns	—
AD63	TDPU	从 A/D 关闭到 A/D 开始工作使模拟级稳定的时间	—	20	—	μs	—

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。

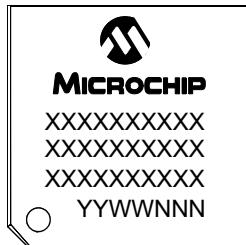
dsPIC30F5011/5013

注：

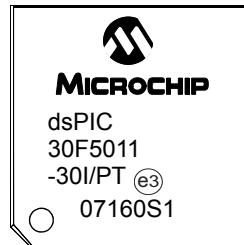
24.0 封装信息

24.1 封装标识信息

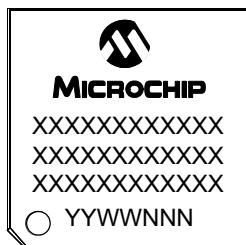
64 引脚 TQFP



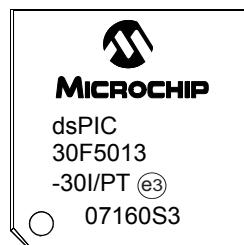
示例



80 引脚 TQFP



示例



图注:

XX...X 客户信息

Y 年份代码（日历年的最后一位数字）

YY 年份代码（日历年的最后两位数字）

WW 星期代码（1月1日的星期代码为“01”）

NNN 以字母数字排序的追踪代码

(e3) 雾锡（Matte Tin, Sn）的 JEDEC 无铅标志

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

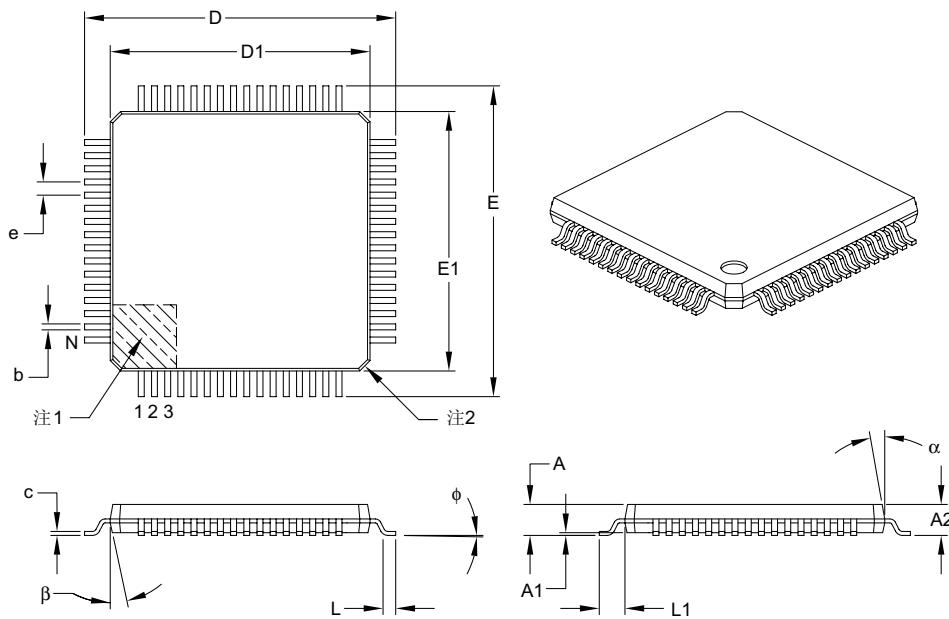
注:

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

dsPIC30F5011/5013

64引脚塑封薄型正方扁平封装 (PT) ——10x10x1 mm主体，2.00 mm引脚投影长度[TQFP]

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



单位		毫米		
尺寸范围		最小	正常	最大
引脚数		64		
引脚间距		0.50 BSC		
总高度		A	—	1.20
塑模封装厚度		A2	0.95	1.00
悬空间隙		A1	0.05	—
底脚长度		L	0.45	0.60
引脚投影长度		L1	1.00 REF	
底脚倾斜角		phi	0°	3.5°
总宽度		E	12.00 BSC	
总长度		D	12.00 BSC	
塑模封装宽度		E1	10.00 BSC	
塑模封装长度		D1	10.00 BSC	
引脚厚度		c	0.09	—
引脚宽度		b	0.17	0.22
塑模顶部锥度		alpha	11°	12°
塑模底部锥度		beta	11°	12°
			13°	

注：

1. 引脚1的可见定位功能可能不同，但必须在阴影区域内。
2. 切角处的斜面是可选的；大小可能有所不同。
3. 尺寸D1和E1不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过0.25毫米。
4. 尺寸和公差请参见ASME Y14.5M。

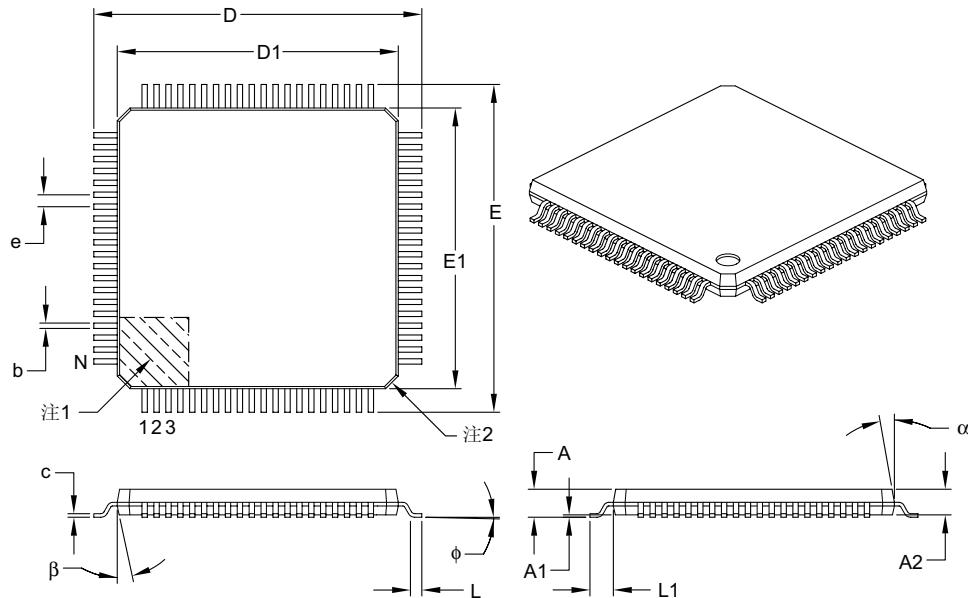
BSC：基本尺寸。显示的是没有公差的理论精确值。

REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology图号C04-085B

80引脚塑封薄型正方扁平封装（PT）——12x12x1 mm主体，2.00 mm引脚投影长度[TQFP]

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



尺寸范围		毫米		
	N	最小	正常	最大
引脚数	N	80		
引脚间距	e	0.50 BSC		
总高度	A	—	—	1.20
塑模封装厚度	A2	0.95	1.00	1.05
悬空间隙	A1	0.05	—	0.15
底脚长度	L	0.45	0.60	0.75
引脚投影长度	L1	1.00 REF		
底脚倾斜角	φ	0°	3.5°	7°
总宽度	E	14.00 BSC		
总长度	D	14.00 BSC		
塑模封装宽度	E1	12.00 BSC		
塑模封装厚度	D1	12.00 BSC		
引脚厚度	c	0.09	—	0.20
引脚宽度	b	0.17	0.22	0.27
塑模顶部锥度	α	11°	12°	13°
塑模底部锥度	β	11°	12°	13°

注：

1. 引脚1的可见定位功能可能不同，但必须在阴影区域内。
2. 切角处的斜面是可选的；大小可能有所不同。
3. 尺寸D1和E1不包括塑模毛边或突起。塑模每侧的毛边或突起不得超过0.25毫米。
4. 尺寸和公差请参见ASME Y14.5M。

BSC：基本尺寸。显示的是没有公差的理论精确值。

REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology图号C04-092B

dsPIC30F5011/5013

注：

附录 A: 版本历史

版本 F (2006 年 5 月)

本数据手册的早期版本中包含超前或初步信息，其中包含不完善的特性数据。

本版本进行了如下更新：

- 支持 I²C 从地址（见表 15-1）
- ADC 转换时钟选择允许采样速率为 200 kHz（见第 19.0 节“12 位模数转换器（ADC）模块”）
- 工作电流（I_{dd}）规范
(见表 23-5)
- BOR 电压限制
(见表 23-11)
- I/O 引脚输入规范
(见表 23-8)
- 看门狗定时器超时限制
(见表 23-21)

版本 G (2007 年 1 月)

此版本包含了对封装图的更新。

dsPIC30F5011/5013

注：

索引

数字

12 位模数转换器 (A/D) 模块 129

A

A/D 129

 ADCHS 寄存器 129

 ADCON1 寄存器 129

 ADCON2 寄存器 129

 ADCON3 寄存器 129

 ADCSSL 寄存器 129

 ADPCFG 寄存器 129

 CPU 空闲模式期间的工作 135

 CPU 休眠模式期间的工作 135

 采样触发的编程设定 131

 采样要求 134

 掉电模式 135

 复位的影响 135

 寄存器映射 137

 结果缓冲器 130

 连接注意事项 136

 配置模拟端口引脚 60, 136

 输出格式 135

 选择转换顺序 130

 中止转换 131

 转换操作 130

AC-Link 模式的工作 126

 16 位模式 126

 20 位模式 127

ADC

 选择转换时钟 131

ADC 转换速度 132

B

BOR 特性 175

BOR。请参见欠压复位。

备用向量表 39

编程操作 49

 擦除程序存储器的一行 49

 启动编程过程 50

 闪存程序存储器的算法 49

 装入写锁存器 50

编程模型 14

 框图 15

变更通知客户服务 219

表指令操作汇总 47

C

CAN 模块 107

 CAN1 寄存器映射 114

 CAN2 寄存器映射 116

 I/O 时序特性 201

 I/O 时序要求 201

 报文发送 111

 报文接收 110

 波特率设置 112

 概述 107

 工作模式 109

 帧类型 107

C 编译器

 MPLAB C18 164

 MPLAB C30 164

CLKOUT 和 I/O 时序

 特性 180

 要求 180

 CPU 架构概述 13

 CPU 空闲模式期间的输出比较工作 85

 操作码说明中使用的符号 156

 程序地址空间 23

表指令

 TBLRDH 25

 TBLRDL 25

 TBLWTH 25

 TBLWTL 25

 存储器映射 23

 构成 24

 使用表指令访问程序存储器中的数据 25

 使用程序空间可视性访问程序存储器中的数据 26

 数据表访问 (低位字) 25

 数据表访问 (最高字节) 26

 数据访问自, 地址生成 24

 数据空间窗口到程序空间的操作 27

 程序和 EEPROM 特性 175

 程序计数器 14

 除法支持 16

 指令 (表) 16

 从休眠和空闲中唤醒 39

 从休眠状态唤醒 139

 存储器构成 23

 内核寄存器映射 32

D

DCI 模块

 COFS 引脚 119

 CPU 空闲模式期间的工作 126

 CPU 休眠模式期间的工作 126

 CSCK 引脚 119

 CSDI 引脚 119

 CSDO 模式位 126

 CSDO 引脚 119

 I/O 引脚 119

 Slot 状态位 126

 采样时钟边沿控制位 124

 从器件帧同步的工作 122

 发送 / 接收移位寄存器 119

 发送时隙使能位 124

 发送状态位 125

 工作 121

 缓冲器长度控制 125

 缓冲器控制 119

 缓冲器数据对齐 119

 缓冲器与数据帧的对齐 125

 寄存器映射 128

 简介 119

 接收时隙使能位 124

 接收状态位 125

 使能 121

 时序特性

 AC-Link 模式 191

 多通道, I²S 模式 189

 时序要求

 AC-Link 模式 191

 多通道, I²S 模式 190

 数据对齐控制位 124

 数字环回模式 126

 同步数据传输 124

 位时钟发生器 123

 下溢模式控制位 126

 用于常见 Codec CSCK 频率的器件频率 (表) 123

 帧同步发生器 121

dsPIC30F5011/5013

帧同步模式控制位	121
帧同步时时隙使能位的工作	124
中断	126
主器件帧同步的工作	121
字长选择位	121
DSP 引擎	17
乘法器	19
代码保护	139
代码示例	
擦除程序存储器的一行	49
擦除数据 EEPROM 的一个字	54
擦除数据 EEPROM 的一块	54
读数据 EEPROM	53
启动编程过程	50
数据 EEPROM 块写入	56
数据 EEPROM 字写入	55
装入写锁存器	50
带隙启动时间	
时序特性	183
要求	183
低电压检测特性	173
低电压检测 (LVD)	150
地址发生器单元	41
电气特性	167
AC	176
DC	167
掉电电流 (IPD)	171
读者反馈	220
端口写 / 读示例	60
F	
防止意外写入 OSCCON	144
封装信息	207
标识	207
复位	139, 145
BOR, 可编程	147
POR	
FSCM 和 PWRT 禁止时的工作	147
具有长晶振起振时间	147
POR (上电复位)	145
欠压复位 (BOR)	139
上电复位 (POR)	139
上电延时定时器 (PWRT)	139
振荡器起振定时器 (OST)	139
复位的来源	
非法指令陷阱	37
看门狗定时器超时	37
欠压复位 (BOR)	37
未初始化的 W 寄存器陷阱	37
陷阱锁定	37
复位过程	37
复位的来源	37
复位时序特性	181
复位时序要求	182
负载条件	176
G	
公式	
ADC 转换时钟	131
COFG 周期	121
波特率	103
串行时钟速率	96
时钟发生的时间份额	113
位时钟频率	123
工作电流 (IDD)	169
工作模式	
初始化	109
环回	109
监听	109
监听所有报文	109
禁止	109
正常工作	109
H	
汇编器	
MPASM 汇编器	164
I	
I/O 端口	59
并行 (PIO)	59
I ² C 10 位从模式的工作	93
发送	94
接收	94
I ² C 7 位从模式的工作	93
发送	93
接收	93
I ² C 模块	91
CPU 休眠和空闲模式期间的工作	96
IPMI 支持	95
编程模型	91
地址	93
各种模式	91
功能说明	91
广播呼叫地址支持	95
寄存器	91
寄存器映射	97
软件控制的时钟延长 (STREN = 1)	94
数据总线时序要求	
从模式	200
主模式	198
斜率控制	95
引脚配置	91
中断	95
总线启动 / 停止位时序特性	
从模式	199
主模式	197
总线数据时序特性	
从模式	199
主模式	197
I ² C 主模式的工作	95
波特率发生器	96
多主器件通信、总线冲突和总线仲裁	96
发送	95
接收	95
时钟仲裁	96
I ² C 主模式支持	95
I ² S 模式的工作	127
数据对齐	127
帧和数据字长选择	127
J	
简单捕捉事件模式	79
Timer2 和 Timer3 选择模式	80
缓冲器操作	80
霍尔传感器模式	80
预分频器	79
简单 PWM 模式	84
输入引脚故障保护	84
周期	85
简单输出比较 /PWM 模式时序要求	188
简单输出比较匹配模式	84
交流特性	176

负载条件	176	应用	44
内部 LPRC 精度	180		
交流温度和电压规范	176		
禁止 UART	101		
K			
开发支持	163		
看门狗定时器			
时序特性	181	寄存器映射	51
时序要求	182	内部时钟时序示例	178
看门狗定时器 (WDT)	139, 150	内核架构	
工作	150	概述	13
使能和禁止	150		
勘误表	6		
可编程	139		
客户通知服务	219		
客户支持	219		
空闲电流 (IDLE)	170		
控制寄存器	48		
NVMDR	48	PICSTART 2 开发编程器	166
NVMDRU	48	PICSTART Plus 开发编程器	166
NVMCON	48	PLL 时钟时序规范	178
NVMKEY	48	PORTA	
快速现场保护	39	dsPIC30F5013 的寄存器映射	61
框图		PORTB	
12 位 A/D 功能	129	dsPIC30F5011/5013 的寄存器映射	61
16 位 Timer2	71	PORTC	
16 位 Timer3	71	dsPIC30F5011 的寄存器映射	61
16 位 Timer4	76	dsPIC30F5013 的寄存器映射	61
16 位 Timer5	76	PORTD	
16 位 Timer1 模块	65	dsPIC30F5011 的寄存器映射	62
32 位 Timer2/3	70	dsPIC30F5013 的寄存器映射	62
32 位 Timer4/5	75	PORTF	
CAN 缓冲器和协议引擎	108	dsPIC30F5011/5013 的寄存器映射	62
DCI 模块	120	POR. 请参见上电复位。	
dsPIC30F5011	8		
dsPIC30F5013	9		
DSP 引擎	18		
SPI	88		
SPI 主 / 从连接	88		
UART 发送器	99		
UART 接收器	100		
复位系统	145		
共用端口结构	60		
输出比较模式	83		
输入捕捉模式	79		
外部上电复位电路	147		
振荡器系统	141		
专用端口结构	59		
L			
LVLD 特性	174		
M			
Microchip 因特网网站	219		
MPLAB ASM30 汇编器、链接器和库管理器	164	RCON 寄存器的状态位、含义以及初始化条件，情形 1	148
MPLAB ICD 2 在线调试器	165	RCON 寄存器的状态位、含义以及初始化条件，情形 2	149
MPLAB ICE 2000 高性能通用在线仿真器	165	软件堆栈指针，帧指针	14
MPLAB PM3 器件编程器	165	CALL 堆栈帧	31
MPLAB REAL ICE 在线仿真器系统	165	软件模拟器 (MPLAB SIM)	164
MPLINK 目标链接器 /MPLIB 目标库管理器	164		
模数转换器。请参见 A/D。			
模寻址	42		
W 地址寄存器选择	43		
递增缓冲区操作示例	43		
起始地址和结束地址	43		
N			
NVM			
寄存器映射	51		
内部时钟时序示例	178		
内核架构			
概述	13		
P			
PICSTART 2 开发编程器	166		
PICSTART Plus 开发编程器	166		
PLL 时钟时序规范	178		
PORTA			
dsPIC30F5013 的寄存器映射	61		
PORTB			
dsPIC30F5011/5013 的寄存器映射	61		
PORTC			
dsPIC30F5011 的寄存器映射	61		
dsPIC30F5013 的寄存器映射	61		
PORTD			
dsPIC30F5011 的寄存器映射	62		
dsPIC30F5013 的寄存器映射	62		
PORTF			
dsPIC30F5011 的寄存器映射	62		
dsPIC30F5013 的寄存器映射	62		
PORTG			
dsPIC30F5011/5013 的寄存器映射	62		
POR. 请参见上电复位。			
Q			
器件概述	7		
器件 ID 存储单元	139		
器件配置			
寄存器映射	153		
器件配置寄存器			
FBORPOR	151		
FBS	151		
FGS	151		
FOSC	151		
FSS	151		
FWDT	151		
欠压复位			
时序要求	182		
特性	174		
R			
RCON 寄存器的状态位、含义以及初始化条件，情形 1	148		
RCON 寄存器的状态位、含义以及初始化条件，情形 2	149		
软件堆栈指针，帧指针	14		
CALL 堆栈帧	31		
软件模拟器 (MPLAB SIM)	164		
S			
SPI 模块	87		
CPU 空闲模式期间的工作	89		
CPU 休眠模式期间的工作	89		
SDOx 禁止	87		
SPI1 寄存器映射	90		
SPI2 寄存器映射	90		
从选择同步	89		
功能说明	87		
时序特性			
从模式 (CKE = 1)	194, 195		
主模式 (CKE = 0)	192		
主模式 (CKE = 1)	193		

时序要求	
从模式 (CKE = 0)	194
从模式 (CKE = 1)	196
主模式 (CKE = 0)	192
主模式 (CKE = 1)	193
帧 SPI 支持	87
字和字节通信	87
闪存程序存储器	47
上电延时定时器	
时序特性	181
时序要求	182
省电模式	
空闲	151
休眠	150
休眠和空闲	139
使能和设置 UART	
设置数据、奇偶性和停止位选择	101
使能 UART	101
时序规范	
PLL 抖动	178
PLL 时钟	178
时序特性	
A/D 转换	
低速 (ASAM = 0, SSRC = 000)	204
A 类、B 类和 C 类定时器外部时钟	184
CAN 模块 I/O	201
CLKOUT 和 I/O	180
DCI 模块	
AC-Link 模式	191
多通道, I ² S 模式	189
I ² C 总线启动 / 停止位	
从模式	199
主模式	197
I ² C 总线数据	
从模式	199
主模式	197
SPI 模块	
从模式 (CKE = 0)	194
从模式 (CKE = 1)	195
主模式 (CKE = 0)	192
主模式 (CKE = 1)	193
带隙启动时间	183
复位	181
看门狗定时器	181
上电延时定时器	181
输出比较 /PWM 模块	188
输出比较模块	187
输入捕捉 (CAPX)	186
外部时钟	176
振荡器起振定时器	181
时序图	
CAN 位	112
I ² S 接口帧同步	122
PWM 输出	85
上电过程中的延时时序 (MCLR 连接至 VDD)	146
上电过程中的延时时序 (MCLR 未连接至 VDD), 情形 1	146
上电过程中的延时时序 (MCLR 未连接至 VDD), 情形 2	146
帧同步, AC-Link 帧起始	122
帧同步, 多通道模式	122
时序图和规范	
直流特性表内部 RC 精度	178
时序图。见时序特性	
时序要求	
A/D 转换	
低速	205
A 类定时器外部时钟	184
B 类定时器外部时钟	185
CAN 模块 I/O	201
CLKOUT 和 I/O	180
C 类定时器外部时钟	185
DCI 模块	
AC-Link 模式	191
多通道, I ² S 模式	190
I ² C 总线数据 (从模式)	200
I ² C 总线数据 (主模式)	198
SPI 模块	
从模式 (CKE = 0)	194
从模式 (CKE = 1)	196
主模式 (CKE = 0)	192
主模式 (CKE = 1)	193
带隙启动时间	183
复位	182
简单输出比较 /PWM 模式	188
看门狗定时器	182
欠压复位	182
上电延时定时器	182
输出比较模块	187
输入捕捉	186
外部时钟	177
振荡器起振定时器	182
输出比较 /PWM 模块时序特性	188
输出比较模块	83
寄存器映射	86
时序特性	187
时序要求	187
输出比较中断	85
数据地址空间	27
MCU 和 DSP (MAC 类) 指令示例	29
Near 数据空间	31
存储器映射	27, 28
对齐	30
对齐 (图)	30
非法存储器访问的后果 (表)	30
空间	30
宽度	30
软件堆栈	31
数据 EEPROM 存储器	53
擦除	54
擦除, 块	54
擦除, 字	54
读	53
防止误写操作的保护措施	57
写	55
写校验	57
写, 块	56
写, 字	55
数据累加器和加法器 / 减法器	19
回写	20
舍入逻辑	20
数据空间写饱和	21
溢出和饱和	19
数据转换器接口 (DCI) 模块	119
输入捕捉模块	
寄存器映射	81
中断	80
输入捕捉时序要求	186
输入捕捉 (CAPX) 时序特性	186
输入电平变化通知模块	63

dsPIC30F5011 寄存器映射 (Bit 15-8)	63	自动波特率支持	104
dsPIC30F5011 寄存器映射 (Bit 7-0)	63	W	
dsPIC30F5013 寄存器映射 (Bit 15-8)	63	WWW 地址	219
dsPIC30F5013 寄存器映射 (Bit 7-0)	63	WWW, 在线支持	6
双输出比较匹配模式	84	外部时钟时序特性	
单脉冲模式	84	A 类、B 类和 C 类定时器	184
连续脉冲模式	84	外部时钟时序要求	177
T		A 类定时器	184
Timer2/3 模块	69	B 类定时器	185
16 位定时器模式	69	C 类定时器	185
32 位定时器模式	69	外部中断请求	39
32 位同步计数器模式	69	外设模块禁止 (PMD) 寄存器	152
ADC 事件触发信号	72	位反转寻址	44
定时器预分频器	72	示例	44
寄存器映射	73	实现	44
门控操作	72	修改量表	45
休眠模式下的操作	72	序列表 (16 项)	45
中断	72	温度和电压规范	
Timer4/5 模块	75	AC	176
寄存器映射	77	X	
Timer2 和 Timer3 选择模式	84	系统集成	139
Timer1 模块	65	寄存器映射	153
16 位定时器模式	65	陷阱向量	38
16 位同步计数器模式	65	休眠和空闲模式下的输入捕捉工作	80
16 位异步计数器模式	65	CPU 空闲模式	80
寄存器映射	68	CPU 休眠模式	80
门控操作	66	休眠模式下的输出比较工作	85
实时时钟	66	Y	
振荡器操作	67	异常过程	
中断	67	陷阱的来源	37
休眠模式下的操作	66	引脚说明	10
预分频器	66	因特网地址	219
中断	66	运行时自编程 (RTSP)	47
桶形移位寄存器	21	Z	
通用异步收发器 (UART) 模块	99	在线串行编程 (ICSP)	47, 139
U		振荡器	
UART 工作		工作模式 (表)	140
空闲模式	104	配置	142
休眠模式	104	LP 振荡器控制	142
UART 模块		初始时钟源选择	142
8 位或 9 位数据模式下的接收	102	低功耗 RC (LPRC)	143
8 位数据模式下的发送	101	故障保护时钟监视器	144
9 位数据模式下的发送	101	快速 RC (FRC)	143
CPU 休眠和空闲模式期间的工作	104	起振定时器 (OST)	142
UART1 寄存器映射	105	锁相环 (PLL)	143
UART2 寄存器映射	105	系统概述	139
波特率发生器	103	振荡器起振定时器	
地址检测模式	103	时序特性	181
发送缓冲器 (UxTXB)	101	时序要求	182
发送间隔字符	102	振荡器选择	139
发送数据	101	指令集	
发送中断	102	概述	158
概述	99	汇总	155
环回模式	103	指令寻址模式	
接收错误处理	102	MAC 指令	42
接收缓冲器溢出错误 (OERR 位)	102	MCU 指令	41
接收缓冲器 (UxRXB)	102	其他指令	42
接收间隔字符	103	文件寄存器指令	41
接收数据	102	支持的基本模式	41
接收中断	102	传送指令和累加器指令	42
空闲状态	103	直流特性	167
奇偶校验错误 (PERR)	103	BOR	175
使能和设置	101		
帧错误 (FERR)	103		

dsPIC30F5011/5013

I/O 引脚输出规范	173
LVDL	174
程序和 EEPROM	175
低电压检测	173
掉电电流 (IPD)	171
工作电流 (IDD)	169
空闲电流 (I _{IDLE})	170
欠压复位	174
温度和电压规范	168
中断	35
中断过程	39
中断堆栈帧	39
中断控制器	
寄存器映射	40
中断优先级	36
陷阱	37
状态寄存器	14
自动时钟延长	94
10 位寻址时 (STREN = 1)	94
7 位寻址时 (STREN = 1)	94
发送模式	94
接收模式	94

MICROCHIP 网站

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

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及存档软件
- **一般技术支持**——常见问题 (FAQ)、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排、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 的客户通知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)”服务并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。

dsPIC30F5011/5013

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： dsPIC30F5011/5013

文献编号：

DS70116G_CN

问题：

1. 本文档中哪些部分最有特色？

2. 本文档是否满足了您的软硬件开发要求？如何满足的？

3. 您认为本文档的组织结构便于理解吗？如果不便于理解，那么问题何在？

4. 您认为本文档应该添加哪些内容以改善其结构和主题？

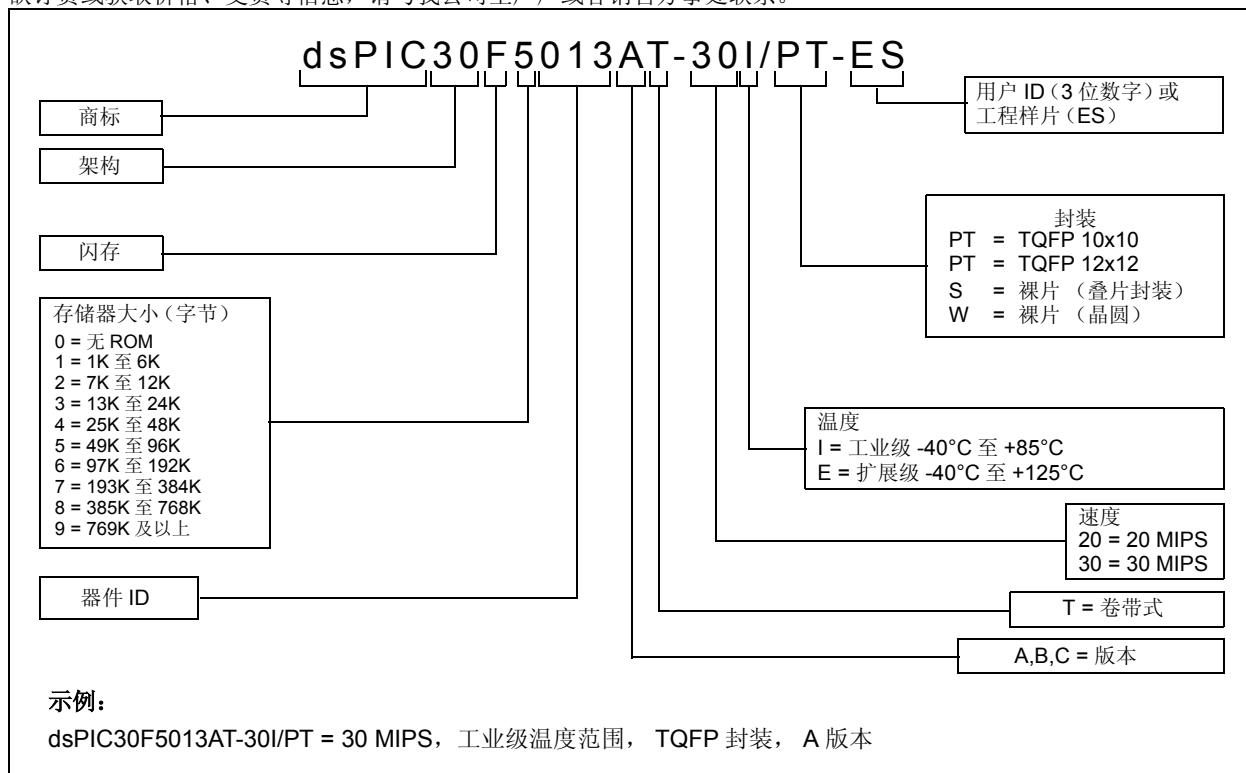
5. 您认为本文档中可以删减哪些内容，而又不会影响整体使用效果？

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

7. 您认为本文档还有哪些方面有待改进？

产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。





MICROCHIP

全球销售及服务网点

美洲

公司总部 Corporate Office

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://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: 60-4-646-8870
Fax: 60-4-646-5086

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