

DSP 控制器原理及应用

宁改娣 杨拴科 编著

科学出版社

北 京

内 容 简 介

DSP 的应用范围已日益广泛,许多用 MCU 的领域也积极地更新换代为 DSP。DSP 控制器是一种适合于控制系统的 DSP 芯片。本教材以 TMS320C24x 为模型介绍其结构、指令系统及应用;书中介绍了 DSP 的开发流程和开发环境,提供实验内容,带领读者进入一个完整的学习 DSP 的环境,同时给出了一些通过实验调试成功的应用实例。此外还简单介绍了 MATLAB 软件,并给出用 MATLAB 实现 DSP 算法仿真的实例。

本书简单明了、易学易用,应用举例全部通过调试和验证。本书可作为学习 DSP 控制器的本科生和研究生教材,也适用于高等院校相关专业师生和科技人员阅读参考。

图书在版编目(CIP)数据

DSP 控制器原理及应用/宁改娣,杨拴科编著. 北京:科学出版社,2002

ISBN 7-03-010909-0

I. 数… II. ①宁…②杨… III. 数字信号-信号处理-微处理器
IV. TN911.72

中国版本图书馆 CIP 数据核字(2002)第 081451 号

责任编辑:赵卫江/责任校对:包志虹
责任印制:吕春珉/封面设计:飞天创意

出版

北京东黄城根北街 16 号
邮政编码:100717

<http://www.sciencep.com>

印刷

科学出版社发行 各地新华书店经销

*

2002 年 11 月第 一 版 开本:787×1092 1/16
2006 年 8 月第三次印刷 印张:17 3/4
印数:6 001—7 000 字数:400 000

定价:26.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

前 言

DSP 一方面是指数字信号处理技术,另一方面是指数字信号处理器。数字信号处理器即 DSP 芯片的诞生,使得数字信号处理技术的理论研究成果广泛应用到实际系统中,成为数字信号处理技术和应用之间的桥梁,并进一步促进数字信号处理技术的深入发展和应用领域的拓宽。本教材主要介绍数字信号处理器。

在微电子技术发展的带动下,DSP 芯片功能日益强大,性能价格比不断提高,开发环境日臻完善,应用领域不断扩大。在步入数字化时代的进程中扮演着举足轻重的角色。

虽然早在 20 世纪 80 年代中期,DSP 芯片在国内已有许多应用,但直到最近几年,各大专院校才有相应的课程设置,而且许多院校开设的课程都是以 TI 的 TMS320C5000 系列芯片为模型机进行介绍,教材较少。介绍 TMS320C2000 系列及其应用的教材更是凤毛麟角。本教材就是为了方便教学和应用 DSP 控制器的科技人员参考为目的而编写的。

全书共分 8 章,主要包括 3 个部分。

第一部分为基础部分,包括第 1~3 章和第 5~6 章。主要介绍 DSP 芯片的发展、特点,TI 公司 TMS320 DSP 各系列特点以及数字基础;以 TMS320C24X DSP 控制器为模型介绍了其指令系统和硬件结构,在对硬件深入介绍的同时还给出部分模块的扩展方法。本部分还详细介绍了 DSP 指令系统的寻址方式,对各种指令只给出分类列表,而对每条指令的理解和使用,我们给大家提供了一个学习的途径和方法。

第二部分为应用开发部分,包括第 4 章和第 8 章。主要介绍 DSP 开发流程、开发工具的使用及实验。在第一部分基础上,介绍 DSP 的开发环境(软件 CCS 和 PCI 仿真卡),使得开发者能够很快熟悉开发环境,缩短开发周期。根据以往的教学经验和学生的反馈意见,对于学习微处理器芯片及其应用类课程,实验是非常重要的一个环节。因此,在第 8 章中提供了大量的实验内容。部分实验给出了通过调试的源程序。

第三部分(第 7 章)为算法仿真,DSP 的主要特点是适用于数字信号处理。在具体实现 DSP 算法以前,一般首先要对其进行模拟仿真以判断其正确性、复杂性和可靠性。以前 DSP 算法一般用 C 语言或其他高级语言来模拟实现,但模拟过程繁琐、调试不便。目前这种模拟可以用 MATLAB 语言快速方便地实现,仿真结果可以转化为 C 语言。另外 MATLAB 还可以与 TI 的 DSP 仿真环境通讯,但这部分有待大家一起探索。由此可见,借助 MATLAB 将会非常有利于数字信号处理的实现。因此,本书对 MATLAB 仿真软件做简单介绍,并给出通过调试的算法仿真程序。

本书由宁改娣和杨拴科编写,第一、三章由杨拴科编写;其余部分由宁改娣编写。在编写过程中,参阅了不少国内外参考书及资料,学习和吸取了不少经验。在此向这些作者致以谢意!在书稿的录入过程中,刘涛、杜倩宁、何永威、雍亮、廉海涛等同志参与了少部分工作,在此表示感谢!另外,书中一些实例是由选修 DSP 课程的部分研究生提供,陶坤宇工程师也对本书提出了许多建议,在此一并表示感谢!本教材能得以出版还要特别感谢樊捷和李洁的支持。本书由申忠如教授主审,并提出了许多宝贵意见,在此表示衷心的感谢。

感谢。

我校 DSP 课程的开设以及本书的出版也得到电气工程学院各位领导及电工电子教学中心领导和同事们的大力支持和关心,在此表示诚挚的谢意。

由于时间仓促,加之作者的水平和掌握的资料有限,书中错误和不当在所难免,恳请读者批评指正。

作 者

2002 年 6 月于西安交通大学

目 录

第一章 概述	(1)
1.1 引言	(1)
1.2 DSP 芯片	(2)
1.2.1 DSP 芯片概述	(2)
1.2.2 DSP 芯片的结构特征	(6)
1.3 TMS320 系列 DSP	(7)
1.3.1 TMS320 系列概况	(7)
1.3.2 TMS320 系列 DSP 命名	(10)
1.4 数字运算基础	(10)
1.4.1 数的定标	(11)
1.4.2 DSP 定点算术运算	(12)
第二章 TMS320C2××的 CPU 结构和存储器配置	(14)
2.1 TMS320C2××系列 DSP 结构	(14)
2.1.1 概述	(14)
2.1.2 'C24×控制器结构	(16)
2.2 'C2××的总线结构	(18)
2.3 'C2××系列 CPU 结构	(19)
2.3.1 中央算术逻辑单元(CALU)和累加器(ACC)	(20)
2.3.2 定标移位器	(20)
2.3.3 乘法器	(21)
2.3.4 辅助寄存器算术单元(ARAU)和辅助寄存器	(21)
2.3.5 状态寄存器 ST0 和 ST1	(22)
2.4 'C2××存储器和 I/O 空间	(23)
2.4.1 存储器概述	(23)
2.4.2 DSP 片内存储器类型	(24)
2.4.3 程序存储器	(25)
2.4.4 局部数据存储器	(28)
2.4.5 全局存储器及扩展	(30)
2.4.6 I/O 空间	(32)
2.5 程序控制	(33)
2.5.1 程序地址的产生	(33)
2.5.2 堆栈和微堆栈	(35)

第三章 寻址方式与指令系统	(37)
3.1 寻址方式	(37)
3.1.1 立即寻址方式	(37)
3.1.2 直接寻址方式	(38)
3.1.3 间接寻址方式	(39)
3.2 'C2××的指令集	(40)
3.3 'C2××的伪指令	(47)
3.4 宏指令	(53)
第四章 DSP 开发环境	(54)
4.1 JTAG 标准接口	(54)
4.2 开发工具安装	(56)
4.3 TMS320F240EVM 板使用说明	(59)
4.4 软件开发流程	(66)
4.5 软件 CCS 的使用	(77)
4.5.1 建立新工程	(77)
4.5.2 程序调试	(81)
第五章 TMS320C2 ××系统功能和其他功能模块	(85)
5.1 系统接口模块	(85)
5.2 系统配置	(85)
5.3 时钟模块及低功耗方式	(88)
5.3.1 时钟结构及时钟产生电路	(88)
5.3.2 各种时钟信号及时钟模块编程	(89)
5.3.3 低功耗(省电)模式	(92)
5.4 DSP 复位	(94)
5.4.1 引起 DSP 复位的原因	(94)
5.4.2 复位源识别	(95)
5.5 等待状态发生器	(95)
5.6 中断系统	(97)
5.6.1 'C2××中断类型及结构	(97)
5.6.2 'C2××中断流程	(102)
5.6.3 中断编程	(107)
5.7 数字输入/输出(I/O)	(110)
5.7.1 F/C240 数字输入/输出模块及寄存器	(110)
5.7.2 F/C240 共享引脚配置	(113)
5.8 看门狗(WD)和实时中断(RTI)	(114)
5.8.1 看门狗(WD)和实时中断(RTI)结构	(115)

5.8.2 看门狗(WD)和实时中断(RTI)寄存器	(116)
5.8.3 看门狗(WD)和实时中断(RTI)编程	(119)
第六章 TMS320C24×片内外设	(121)
6.1 模拟/数字转换(ADC)	(121)
6.1.1 双10位 A/D 转换原理	(121)
6.1.2 双10位 A/D 转换器编程	(123)
6.2 串行通信接口(SCI)	(127)
6.2.1 SCI 模块结构概述	(127)
6.2.2 SCI 多处理器通信	(130)
6.2.3 SCI 接收和发送时序及中断	(133)
6.2.4 SCI 编程	(135)
6.3 串行外设接口(SPI)	(144)
6.3.1 SPI 模块结构概述	(144)
6.3.2 SPI 编程	(151)
6.4 事件管理模块(EV).....	(157)
6.4.1 事件管理模块(EV)概述.....	(157)
6.4.2 通用定时器	(164)
6.4.3 比较单元	(175)
6.4.4 与全比较单元相关的 PWM 电路	(180)
6.4.5 捕获单元	(183)
6.4.6 正交编码脉冲电路.....	(187)
第七章 信号处理算法的 MATLAB 仿真	(190)
7.1 MATLAB 简介	(190)
7.1.1 MATLAB 的工作环境及编程方法	(190)
7.1.2 MATLAB 主要语法、运算符及命令简介	(192)
7.2 MATLAB 实现 DSP 算法仿真	(193)
7.2.1 信号的谱分析(FFT)	(194)
7.2.2 IIR 滤波器设计	(195)
7.2.3 随机信号的线性谱估计	(199)
第八章 DSP 应用(实验)	(206)
8.1 实验一:熟悉实验软、硬件环境	(206)
8.2 实验二:中断编程方法(以捕获单元为例)	(206)
8.3 实验三:数字振荡器的实现	(215)
8.4 实验四:伪随机序列发生器	(225)
8.5 实验五:FFT 算法的实现	(230)
8.6 实验六:双音多频电话拨号音频解调/发生器	(240)

8.7 实验七:PWM 波发生器	(242)
8.8 实验八:自选设计实验	(246)
附录	(247)
附录 A TMS 320F/C240 寄存器汇总	(247)
附录 B 英文缩写词汇表	(261)
附录 C F2×× Flash 烧写程序使用说明	(262)
附录 D COFF 文件到 EPROM 文件格式的转换	(265)
附录 E DSP 网页导航	(266)
附录 F 典型指令介绍	(269)
参考文献	(276)

第一章 概 述

1.1 引 言

我们通常所说的 DSP 有两个含义:其一是 Digital Signal Processing 的简称,是指数字信号处理技术,它不仅涉及许多学科,还广泛应用于多种领域。特别在 20 世纪 60 年代,随着计算机和信息技术的迅猛发展,进一步推动了数字信号处理技术的理论 and 应用领域的发展;DSP 的第二个含义是 Digital Signal Processor 的简称,即数字信号处理器(也称为 DSP 芯片),它不仅具有可编程性,而且其实时运行速度远远超过通用微处理器。它是一种适合于数字信号处理的高性能微处理器。数字信号处理器已成为数字信号处理技术和实际应用之间的桥梁,并进一步促进了数字信号处理技术的发展,也极大地拓展了数字信号处理技术的应用领域。

在微电子技术发展的带动下,DSP 芯片的功能日益强大,性能价格比不断提高,开发环境日臻完善,应用领域不断扩大。在步入数字化时代的进程中,数字信号处理器扮演着举足轻重的角色。

本教材中的 DSP 主要是指数字信号处理器。书中详细介绍了 DSP 芯片的结构、指令系统及应用。当然,要充分利用好数字信号处理器,必须要掌握相关的数字信号处理技术。在介绍 DSP 芯片之前,先简单回顾一下数字信号处理的一般流程。

数字信号处理系统一般的是利用计算机或专用处理设备对信号进行滤波、采集、变换、存储和处理,得到需要的信号形式。采用 DSP 芯片的信号处理系统的一般框图如图 1.1 所示。



图 1.1 DSP 信号处理系统框图

由图 1.1 可见,DSP 模块的输入和输出数据都是数字信号,为了利用 DSP 的快速实时的数字处理能力以及数字信号处理的要求,在数字信号处理之前必须加预处理模块;同样,DSP 的数字输出信号也必须转换为系统需要的信号形式。当然在有些系统中并不一定每部分都需要。比如,如果系统输入信号本身就是数字信号,显然不需要低通滤波和 ADC 模块,有些 DSP 芯片包括了 ADC 甚至 DAC。

数字信号处理模块是系统的核心,贯穿本教材的始末。下面先对框图中数字信号处理前后的各模块略做说明。

低通滤波——是将连续信号 $x(t)$ 中的一些次要成分滤除,比如,滤去幅度较小的高频成分及一些杂散信号,以满足采样定理等数字信号预处理要求。

ADC——一般系统中待处理的信号往往是模拟信号,那么在数字信号处理前,首先需

要将模拟信号经过模—数转换器(ADC)转换为数字信号。对模拟信号的采样必须满足采样定理,即采样频率必须大于或等于模拟信号最大频率分量的 2 倍,这样才能由采样信号无失真地恢复原模拟信号。在此前提下,通过按等间隔 T 对模拟信号的采样,得到一串采样点上的样本数据,这一串数据可看做时域离散信号 $x(n)$,用 m 位的 ADC,将各 $x(n)$ 转换为 m 位二进制数据,即形成数字信号。

DAC 与平滑滤波——数字信号经过处理后,要经过 DAC 转换为模拟信号,DAC 输出是一个零阶保持器输出,即输出是台阶形的。所以一般在 DAC 之后加一平滑低通滤波器,滤除多余的高频分量,对产生时间域模拟信号波形起平滑作用。

以上这些环节包括数字信号处理的一些算法,请读者参阅有关信号处理技术的专著,本教材仅介绍数字信号处理器——即 DSP 芯片。只有掌握 DSP 技术和熟悉 DSP 芯片的功能及应用技巧,才能设计、实现一个高效的信号处理系统。

1.2 DSP 芯片

1.2.1 DSP 芯片概述

1. DSP 的发展和分类

在数字信号处理技术发展的初始阶段,人们只能在通用的计算机上进行算法的研究和系统的模拟与仿真。随着数字信号处理技术和集成电路技术的发展,以及数字系统的显著优越性,导致了 DSP 芯片的产生和迅速发展,DSP 芯片的出现才使实时数字信号处理成为现实。

第一片 DSP 器件是 1978 年 AMI 公司宣布的 S2811。

1979 年 Intel 公司推出的 Intel 2920 是第一块脱离了通用型微处理器结构的 DSP 芯片,成为 DSP 芯片的一个主要里程碑。

1980 年前后,日本 NEC 公司推出的 μ PD7720 是第一个具有硬件乘法器的商用 DSP 芯片。第一个采用 CMOS 工艺生产浮点 DSP 芯片的是日本 Hitachi 公司,它于 1982 年推出了浮点 DSP 芯片。1983 年,日本 Fujitsu 公司推出的 MB8764,其指令周期为 120ns,且具有双内部总线,从而使处理器的数据吞吐量发生了一个大的飞跃。而第一片高性能的浮点 DSP 芯片应是 AT&T 公司于 1984 年推出的 DSP32。

1982 年前后,美国德州仪器公司(Texas Instrument,简称 TI)成功推出其第一代 DSP 芯片 TMS32010 及其系列产品 TMS32011、TMS320C10/C14/C15/C16/C17 等,之后相继推出了第二代 DSP 芯片 TMS32020、TMS320C25/C26/C28,第三代 DSP 芯片 TMS32C30/C31/C32,第四代 DSP 芯片 TMS32C40/C44,第五代 DSP 芯片 TMS32C50/C51/C52/C53 以及集多个 DSP 于一体的高性能 DSP 芯片 TMS32C80/C82,第六代更为高性能的 TMS320C64 $\times$ /C67 $\times$ 和高性能的 DSP 控制器 C28 $\times$ 等。

TI 在其 TMS320 系列芯片上设置了符合 IEEE1149 标准的 JTAG(Joint Test Action Group)标准测试接口及相应的控制器,通过 JTAG 和专用的仿真器支持 DSP 的仿真和程序的装入(下载),方便了 DSP 应用系统的开发。

Motorola 公司 1986 年推出 MC56001 定点 DSP 芯片,1990 年推出了与 IEEE 浮点格

式兼容的 MC96002 浮点 DSP 芯片。Motorola 的 DSP 芯片上设置了一个 OnCE(On-Chip Emulation)功能模块,用特定的电路和引脚使用户可以检查片内的寄存器、存储器及外设,用单步、断点和跟踪等方式控制和调试程序。目前在 DSP 市场仍有一定影响。

美国模拟器件公司(Analog Devices,简称 AD)也相继推出了一系列具有自己特点的 DSP 芯片,在 DSP 市场上也占有一定份额。

还有许多其他厂家生产 DSP,市场占有率排名前 4 位的公司有 TI、Agere(原 Lucent,中文名为朗讯)、Motorola 和 ADI。它们的市场份额分别是:TI 为 43.5%,Agere 为 16.1%,Motorola 为 12.0%,ADI 占 8.2%。2001 年 DSP 市场总营收为 42.6 亿美元(邵虞,2002)。

我国 DSP 技术起步早,基本上与国外同步发展。我国已有上百所大学从事 DSP 的教学和科研,在信号处理理论和算法上与国外处于同一水平。但 DSP 芯片几乎完全依赖进口,其中 TI 公司产品占 80%以上(田梓,2002)。TI 国内技术支持公司主要有 TI 中国办事处,北京闻亭(WINTECH)公司,北京合众达公司,武汉力源公司等。目前所用的 DSP 开发工具基本上都是这几家公司的产品。

如上所述,DSP 芯片型号多种多样,分类也有多种方法,但主要有以下两种:

按 DSP 芯片处理的数据格式来分,可以分为定点 DSP 芯片和浮点 DSP 芯片,不同的浮点 DSP 芯片所采用的浮点格式不完全一样,有的 DSP 芯片采用自定义的浮点格式,有的 DSP 芯片则采用 IEEE 的标准浮点格式。

按 DSP 芯片的用途来分,可分为通用型 DSP 芯片和专用型的 DSP 芯片。通用型 DSP 芯片适合普通的 DSP 应用,如 TI 公司的一系列 DSP 芯片。专用型 DSP 芯片是为特定的 DSP 运算而设计,更适合特殊的运算,如数字滤波、卷积和 FFT 等。

2. 当前 DSP 芯片发展的主要特点

自 1980 年以来,DSP 芯片得到了突飞猛进的发展,DSP 芯片的应用越来越广泛。从运算速度来看,MAC(一次乘法和一次加法)时间已经从 80 年代初的 400ns(如 TMS32010)减小到 10ns 以下(如 TMS32C54 $\times$),处理能力提高了几十倍。DSP 芯片内部关键的硬件乘法器占模片区(Die area)从 1980 年的 40%左右下降到 5%以下,片内 RAM 增加一个数量级以上。从制造工艺来看,1980 年采用 NMOS 工艺,而现在则普遍采用亚微米 CMOS 工艺。DSP 芯片的引脚数量从 1980 年的最多 64 个增加到现在的 200 个以上,引脚数量的增加,意味着结构灵活性的增加。

随着 DSP 的时钟频率和处理速度越来越高,功能越来越强,芯片的功耗(CMOS 芯片的功耗主要取决于动态功耗)也急速加大。尽管各生产厂家几乎无一例外地采用 CMOS 工艺等技术来降低功耗,但对于电池供电的便携式设备(比如,笔记本电脑、移动通信设备和 PDA 即 Personal Digital Assistant 等)中的 DSP,迫切要求在提高性能的同时,进一步降低工作电压,减小功耗。为此,各 DSP 生产厂家陆续推出低电压(3.3V/2.7V/1.8V 等)DSP 芯片,多数的 DSP 芯片还设置了多种低功耗工作方式。

未来 10 年,全球 DSP 产品将向着高性能、低功耗、加强融合和拓展多种应用的趋势发展,DSP 芯片将越来越多地渗透到各种电子产品当中,成为各种电子产品尤其是通信类电子产品的技术核心,将会越来越受到业界的青睐。据 TI 预测,到 2010 年,DSP 芯片

的集成度将会增加 11 倍,在单个芯片内将能集成 5 亿只晶体管。目前 DSP 的生产工艺已开始从 0.35mm 转向 0.25mm、0.18mm、0.10mm,预计到 2005 年,TI 生产 DSP 芯片的工艺将达到 0.075mm 的水平,届时,将能够在一块仅有拇指大小的单个芯片上集成 8 个 TMS320DSP 内核。ADI 公司副总裁 Ben Naskar 指出:“面对新世纪的网络产品、消费类电子产品以及无线通信等领域不断涌现的新应用,DSP 产品在不断地提高性能和增加功能的同时,正在不断地降低功耗和减小体积,以便适应市场的需求。”(琼月,2002)

随着 DSP 应用的日益广泛,DSP 已成为许多应用系统设计中不可缺少的组成部分,其结果使 DSP 厂商的投资集中于 DSP 体系结构的完善和支持软件的升级。比如,TI 为 TMS320 系列提供了 eXpressDSP 实时软件技术的支持,它包括 Code Composer Studio V2.0(即 CCS2.0)集成开发环境、DSP/BIOS 实时软件内核、TMS320 算法标准以及业界最大的第三方网络提供的可重用的软件模块。

由于汇编语言是面向机器的,具体地说是面向芯片,即不同厂商的 DSP 有不同的汇编语言指令系统,使用汇编语言编写 DSP 应用软件是一件繁琐与困难的事。而且随着 DSP 寻址空间越来越大,减小了对程序目标代码容量的限制。因此,各公司陆续推出了高级语言编译器,主要是 C 语言编译器,它可以将 C 语言程序编译并优化处理成相应的 DSP 汇编程序或目标程序。对于 TI 的高性能 TMS320C6000 系列及 TMS320C2000 的新成员 C28 $\times$,用 C 语言编程效率非常高,代码优化可达 85%左右,缩短了软件开发周期,程序可移植性好。

为了缩短 DSP 应用系统开发周期,各 DSP 生产厂家为应用软件的开发准备了一些常用数字信号处理函数库与软件工具包,以及各种接口程序等,这些经过优化的子程序为用户提供了很大的方便,使得程序设计更加简单快捷。

对 DSP 芯片的发展,可以总结为四个字:多、快、好、省。

多——型号越来越多,集成的片内外设越来越多。

快——DSP 频率越来越高,速度越来越快。

好——性能价格比越来越高。

省——功耗相对越来越低。

3. DSP 的应用

随着 DSP 的高速发展,性能价格比的不断提高,使 DSP 成为当今和未来技术发展的新热点,使用范围日益扩大,几乎遍及电子技术的所有领域。DSP 的典型应用主要有如下几个方面:

- 数字信号处理,如滤波、FFT、相关、卷积、模式匹配、窗函数和波形产生等;
- 通信,如调制解调、扩频通信、纠错编码、传真、语音信箱、噪音对消和可视电话等;
- 语音处理,如语音编码、语音合成、识别、增强、语音存储及语音邮件等;
- 图形/图像处理,如三维图形变换处理、模式识别、图像压缩与传输、图像增强、动画、机器人视觉和电子地图等;
- 仪器仪表,如频谱分析、函数/波形发生、数据采集;
- 军事,如保密通信、全球定位、雷达与声纳信号处理、搜索与跟踪导航与制导等。

另外,DSP 在医疗和消费电子等许多领域都得到广泛应用,并且会随着 DSP 性价比

的不断提高和开发工具的进一步完善扩展更多的应用领域。

4. DSP 芯片的选择

设计 DSP 应用系统时,其中首要且非常重要的一个环节就是选择 DSP 芯片。只有选定了 DSP 芯片才能进一步设计外围电路。DSP 芯片的选择应根据实际的应用系统需要而确定。一般来说,选择 DSP 芯片时考虑如下诸多因素:

- DSP 芯片的运算速度。运算速度是 DSP 芯片的一个最重要的性能指标,也是选择 DSP 芯片时所需要考虑的一个主要因素。DSP 芯片的运算速度主要由指令周期和 MIPS(即每秒执行百万条指令)来衡量;

- DSP 芯片的硬件资源及性价比;
- DSP 芯片的开发工具是否易学易用;
- 其他的因素,如封装的形式、质量标准、生命周期等;
- 应用系统对功耗的要求。

下面对功耗和运算速度两方面做较详细的介绍。

如果应用系统对功耗要求很高时,要考虑以下几个方面:

- 选择低功耗 DSP 器件;
- 合理设计软件降低功耗。TI 的 TMS320 系列 DSP 有几种降功耗模式,使 IDLE 指令进入低功耗模式;

- 合适的 DSP 运行速度。TMS320 系列的 DSP 一般采用 CMOS 工艺,CMOS 电路的静态功耗极小,而 CMOS 电路的动态功耗的大小与该电路改变逻辑状态的频率和速度密切相关,当时钟频率增加时,电流也相应地增加。TMS320 系列应用系统的功耗与 DSP 的工作频率几乎成正比。在不需要 DSP 的全部运算能力时,可以适当的降低 TMS320 的系统时钟频率使 DSP 适速运行以降低系统功耗;

- 正确处理外围电路。应尽可能地选用低功耗的外围器件,例如系统的显示部分应选用 LCD(液晶显示器)等。复杂的外围电路尽量采用单片的 CPLD 来完成。对 DSP 芯片中未使用的输入引脚应接地或接电源电压,若将这些引脚悬空,在引脚上很容易积累电荷,产生较大的感应电动势,使输入引脚电位处于 0 与 1 间的过渡区域,内部门电路处于动态过程(即反相器上、下两个场效应管都处于导通状态),使系统功耗大大增加。

DSP 的运算速度的快慢是选择 DSP 芯片首先要考虑的问题。DSP 应用系统的运算量是确定 DSP 芯片的基础。DSP 对数据的处理一般有两种方法,下面介绍按照运算量来选择 DSP 芯片的简单方法。

(1) 按样点处理

按样点处理就是 DSP 算法对每一个输入样点循环处理一次。例如;一个采用 LMS 算法的 256 抽头的自适应 FIR 滤波器,假定每个抽头的计算需要 3 个 MAC 周期,则 256 抽头计算需要 $256 \times 3 = 768$ 个 MAC 周期。如果采样频率为 8kHz,即样点之间的间隔为 $125\mu\text{s}$,若 DSP 芯片的 MAC 周期为 200ns,则 768 个周期需要 $153.6\mu\text{s}$ 的时间,显然无法实时处理,需要选用速度更快的芯片。

(2) 按帧处理

有些数字信号处理算法不是对每个输入样点进行处理,而是在一定的时间间隔(记为

$\Delta\tau_{\text{ns}}$)对多个数据(通常也称为帧)循环处理一次。所以选择 DSP 芯片应该比较一帧内 DSP 芯片的处理能力和 DSP 算法的运算量。假设所选 DSP 芯片的指令周期为 $T(\text{ns})$, 则该 DSP 芯片在一帧内最多能运行 $\Delta\tau/T$ 条指令。

1.2.2 DSP 芯片的结构特征

为了实现快速的数字信号处理,DSP 芯片一般都采用特殊的软硬件结构。下面以 TI 公司的 TMS320 系列为例介绍 DSP 芯片的基本结构。

TMS320 系列 DSP 主要采取了哈佛结构、流水线技术、硬件乘法器和特殊 DSP 指令等。以下对这些特点分别介绍。

1. 哈佛结构

哈佛结构是一种并行体系结构,主要特点是将程序和数据存储在不同的存储器空间,对程序和数据独立编址,独立访问。而且在 DSP 内部设置了数据和程序两套总线,使得取指令和执行能完全重叠运行,提高数据吞吐量。为了进一步提高速度和灵活性,TMS320 系列产品中,在哈佛结构上作了改进,一是允许程序存储在高速缓存(Cache)中,提高指令读取速度;二是允许数据存放在程序存储器中,并被算术运算指令直接使用,增强芯片的灵活性。另外 DSP 中的双口 RAM(DARAM)及独立读写总线使数据存取速度提高。

2. 流水线技术

DSP 芯片广泛采用流水线技术,增强了处理器的处理能力。TMS320 系列流水线深度为 2~6 级不等,也就是说,处理器在一个时钟周期可并行处理 2~6 条指令,每条指令处于流水线的不同阶段。图 1.2 为三级流水线操作的例子。在三级流水线操作中,取指令、指令译码和执行可以独立地处理,这样 DSP 可以同时处理多条指令,只是每条指令处于不同的处理阶段。例如,在取第 N 条指令时,前一条指令(即第 $N-1$ 条)处于译码阶段,而第 $N-2$ 条指令则在执行阶段。

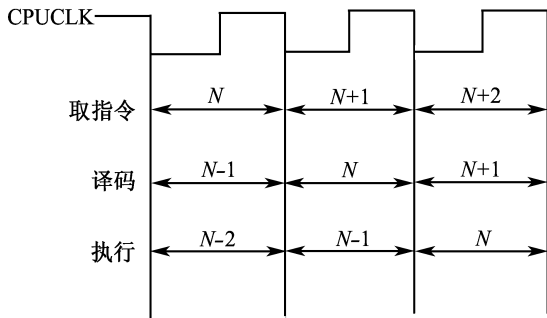


图 1.2 三级流水线

3. 硬件乘法器

在数字信号处理的许多算法中(如 FFT 和 FIR 等),需要做大量的乘法和加法。显然,乘法速度越快,数据处理能力就越强。在通用的微处理器中,有些根本没有乘法指令,有乘法指令的处理器,其乘法指令的执行时间也较长。相比而言,DSP 芯片一般都有一个硬件乘法器,在 TMS320 系列中,一次乘累加最少可在一个时钟周期完成。

4. 特殊 DSP 指令

DSP 芯片的另外一个特点就是采用了特殊的寻址方式和指令。比如,TMS320 系列的位反转寻址方式,LTD、MPY、RPTK 等特殊指令。采用这些适合于数字信号处理的寻址方式和指令,进一步减小了数字信号处理的时间。

另外,由于 DSP 的时钟频率提高,执行周期的缩短,加上以上一些 DSP 结构特征使得 DSP 实现实时数字信号处理成为可能。

1.3 TMS320 系列 DSP

1.3.1 TMS320 系列概况

TMS320 系列包括了定点、浮点和多处理器数字信号处理芯片。其体系结构适合于实时数字信号处理。TMS320 主要 DSP 系列产品如图 1.3 所示,其中,TMS320C1×(简称为'C1×,以下类似)、'C2×、'C2××、'C5×、'C54×、'C64×和'C62×系列为定点 DSP;'C3×、'C4×和'C67×系列为浮点 DSP;'C8×系列是多处理器 DSP。'C24×是'C2××定点数字信号处理器中的一种,它将实时处理能力与控制器的外设功能集于一身,为控制系统数字处理提供了一个理想的解决方案,所以称为 DSP 控制器。

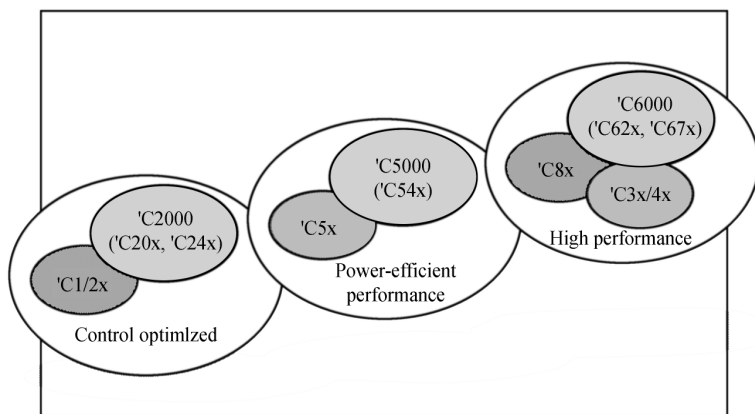


图 1.3 TMS320 系列产品

TI 的 DSP 产品可分为三种不同指令集的三大系列:TMS320C2000、TMS320C5000 和 TMS320C6000。各系列特点如下。

TMS320C2000——作为优化控制的最佳 DSP, 该系列提供了业界成本最低, 涉及最广的数字化控制解决方案, 自然成为家电、空调(HVAC)系统、厂房自动化系统、电机控制和电力电子的首选控制器件。该系列是由'C5×简化和改进产生, 因此两者汇编指令兼容。C2000 系列的新成员——'C28×是 TI 推出的又一种高性能控制器, 可达 400MIPS (Million Instructions Per Second, 即每秒百万条指令, 简称 MIPS)。'C24×是目前使用较多的 16 位定点 DSP 控制器, 速度为 20MIPS。用于电机拖动、工业自动化、电力系统、信息家电等。常用芯片售价约为 4~18 美元。

TMS320C5000——是最节能的 DSP。16 位定点, 速度 40~200MIPS。主要用途是有线和无线通信、IP、便携式信息系统、寻呼机、助听器等。价格 5~75 美元。在将电压降至 0.9V, 功耗降至 0.05 mW/MIPS 的同时, 仍保持高性能的器件。如此低的功耗, 使得 C5000 系列 DSP 最适合于个人便携产品, 诸如数码音乐播放器、通信系统、3G 蜂窝电话和数码相机等。其成员主要有'C54×和'C55×, 两者软件兼容。

TMS320C6000——是高性能的、最快的 DSP, 产品包括不同的性能级别, 最高的可达 1.1GHz。这样杰出的性能, 使其在诸如有线/无线宽带网络、组合 Modem、GPS 导航、基站数字波束形成、医学图像处理、语音识别、3-D 图形、ADSL Modem、网络系统、中心局交换机、数字音频广播设备等应用领域游刃有余。其成员主要有'C64×和'C62×和'C67×。

TMS320 系列 DSP 具有以下特点:

- 灵活的指令系统;
- 灵活的操作性能;
- 高速的运算能力;
- 改进的并行结构;
- 很高的性能价格比;
- 符合 IEEE1149 的标准的 JTAG 标准测试接口。

TMS320 系列中的同一系列产品(比如,'C2××系列中的各 DSP 芯片)具有相同的

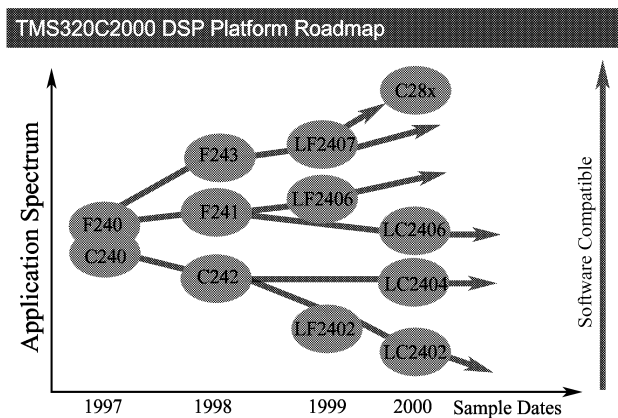


图 1.4 TMS320C2000 系列产品

CPU 结构,只是片内存储器和(或)片内的外设配置不同,派生的器件集成了新的片内存储器和(或)外设,以满足不同应用场所的不同需要。将存储器和外设集成到 DSP 芯片内,降低了系统成本,节省了电路板空间,提高了系统可靠性。

本教材主要介绍 TMS320C2000 中的'C24×系列,'C2××的主要成员如图 1.4 所示。表 1.1 给出了'C2××系列芯片的资源配置。表 1.2 给出了'C24×系列各芯片的更详细的资源配置及性价对比情况。

采用 DSP 进行系统设计时,首先要对 DSP 选型,TI 的网站(www.ti.com)提供了 TMS320 各系列成员的有关特点,可以帮助选型。另外,TI 的第三方公司(TI DSP Third Party)也会提供设计咨询和系统方案等。

表 1.1 TMS320C2××系列芯片的资源配置

TMS320 C2××	指令周期 (ns)	片内 ROM (字)	片内 RAM (字)	片内 Flash (字)	同步 串行口	异步 串行口
C203	25/35/50		544		1	1
C204	25/35/50	4K	544		1	1
C205	25/35/50		4.5K		1	1
F206	25/35/50		4.5K	32K	1	1
F207	25/35/50		4.5K	32K	2	1
C209	35/50	4K	4.5K			
C240	50	16K	544		1	1
F240	50		544	16K	1	1
C241	50	8K	544		1	1
F241	50		544	8K	1	1
C242	50	4K	544		1	1
F243	50		544	8K	1	1

表 1.2 TMS320C24×系列芯片的资源配置

器件名称	MIPS	频率 (MHz)	RAM (字)	ROM (字)	Flash (字)	引导 装载	外部存储 器接口	PWM 通道数	10 位 A/D 个数	A/D 转换 时间	CAN 模块	定时 器数	串行 口	价格 (US \$)
SM320F240	20	40	544		16K	Flash	YES	12	2	6.1μs		3	2	104.72
SMJ320F240	20	40	544		16K	Flash	YES	12	2	6.1μs		3	2	123.20
TMS320C242	20	20	544	4K				8	8	900ns		2	1	Call
TMS320F240	20	20	544		16K	Flash	YES	12	16	6.1μs		3	2	16.21
TMS320F241	20	20	544		8K	Flash		8	8	900ns	YES	2	2	12.37
TMS320F243	20	20	544		8K	Flash	YES	8	8	900ns	YES	2	2	13.99
TMS320LC2401A	40	40	1K	8K				7	5	500ns		2	1	Call
TMS320LC2402A	40	40	544	6K				8	8	425ns		2	1	Call
TMS320LC2404A	40	40	1.5K	16K				16	16	375ns		4	2	5.84
TMS320LC2406A	40	40	2.5K	32K				16	16	375ns	YES	4	2	6.32

续表

器件名称	MIPS	频率 (MHz)	RAM (字)	ROM (字)	Flash (字)	引导 装载	外部存储 器接口	PWM 通道数	10 位 A/D 个数	A/D 转换 时间	CAN 模块	定时 器数	串行 口	价格 (US \$)
TMS320LF2401A	40	40	1K		8K	ROM		7	5	500ns		2	1	7.24
TMS320LF2402	30	30	1K		8K	ROM		8	8	500ns		2	1	8.64
TMS320LF2402A	40	40	1K		8K	ROM		8	8	500ns		2	1	8.21
TMS320LF2403A	40	40	1K		16K	ROM		8	8	500ns	YES	2	2	9.38
TMS320LF2406	30	30	2.5K		32K	ROM		16	16	500ns	YES	4	2	10.56
TMS320LF2406A	40	40	2.5K		32K	ROM		16	16	500ns	YES	4	2	10.03
TMS320LF2407	30	30	2.5K		32K	ROM	YES	16	16	500ns	YES	4	2	11.12
TMS320LF2407A	40	40	2.5K		32K	ROM	YES	16	16	500ns	YES	4	2	10.56

1.3.2 TMS320 系列 DSP 命名

TMS320 系列产品命名方法如图 1.5 所示。

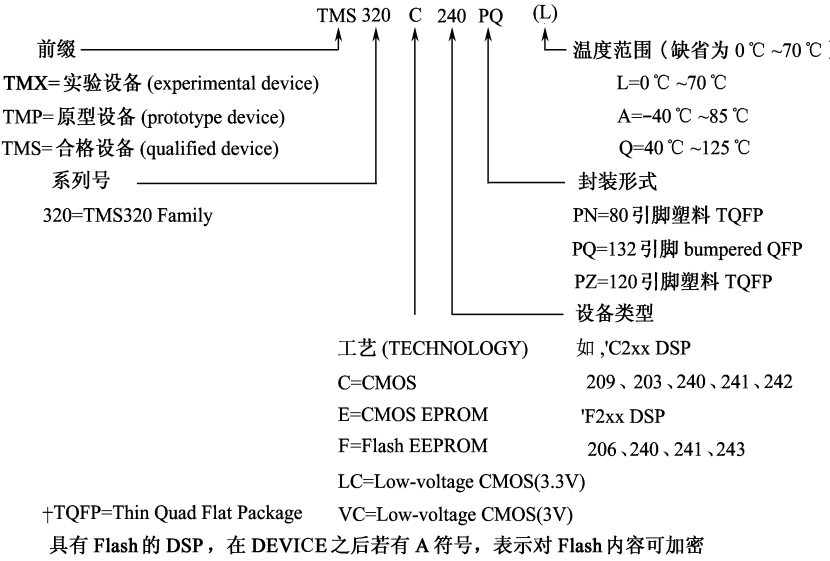


图 1.5 TMS320 系列产品命名方法

1.4 数字运算基础

以上提到 DSP 芯片有定点 DSP 和浮点 DSP，教材中介绍的 TMS2000 系列的 DSP 控制器为定点 DSP，即采用定点进行数值运算。因此，在详细介绍 DSP 内容以前，先介绍一

些定点数据运算基础。

定点进行数值运算操作数一般采用整型数来表示。一个整型数的最大取值范围取决于 DSP 的字长。显然字长越长,所能表示的数据范围越大,精度越高。TMS320C/F240 字长为 16。一般用一个字长表示一个数。

DSP 芯片和所有微处理器一样,它也是以 2 的补码形式表示带符号数。最高为符号位,0 表示数值为正,1 表示负数,其余位为数据位。

例如,带符号数 $11111111111111001\text{B} = \text{FFF9H} = -7$

$0110111011110000\text{B} = 6\text{EF0H} = 28400$

对处理器 DSP 而言,参与数值运算的数都看作 16 位的整型数。但在许多情况下,处理的数据不一定为整数,这种情况下要由程序员来确定一个数的小数点位置处于 16 位的哪一位。这就是数的定标。

1.4.1 数的定标

数的定标是设定小数点在不同位置,来表示不同大小和不同精度的小数。有 Q 表示法和 S 表示法两种定标方法。如果处理器字长为 16 位,将 16 位二进制数由最低有效位 (LSB)到最高有效位的位置依此排为 0~15,那么 Q 表示法的 Q_i 表示小数点在数据第 i 位之后,该数值 i 一般用 Q 表示。比如 Q_0 ,表示小数点在最低位之后,即 $Q=0$ 。Q 表示法的 Q_0 实际上表示一个整数。S 表示法的 $S_{i,j}$ 表示有 i 位整数, j 位小数,即小数点在第 j 位之后,且 $i+j=15$ 。表 1.3 给出两种表示法及各自所表示的十进制数范围。

表 1.3 Q 表示法和 S 表示法及其十进制数值范围

Q 表示法	S 表示法	十进制数值范围
Q_{15}	$S_{0,15}$	$-1 \leq x \leq 0.9999695$
Q_{14}	$S_{1,14}$	$-2 \leq x \leq 1.9999390$
Q_{13}	$S_{2,13}$	$-4 \leq x \leq 3.9998779$
Q_{12}	$S_{3,12}$	$-8 \leq x \leq 7.9997559$
Q_{11}	$S_{4,11}$	$-16 \leq x \leq 15.9995117$
Q_{10}	$S_{5,10}$	$-32 \leq x \leq 31.9990234$
Q_9	$S_{6,9}$	$-64 \leq x \leq 63.9980469$
Q_8	$S_{7,8}$	$-128 \leq x \leq 127.9960938$
Q_7	$S_{8,7}$	$-256 \leq x \leq 255.9921875$
Q_6	$S_{9,6}$	$-512 \leq x \leq 511.9804375$
Q_5	$S_{10,5}$	$-1024 \leq x \leq 1023.96875$
Q_4	$S_{11,4}$	$-2048 \leq x \leq 2047.9375$
Q_3	$S_{12,3}$	$-4096 \leq x \leq 4095.875$
Q_2	$S_{13,2}$	$-8192 \leq x \leq 8191.75$
Q_1	$S_{14,1}$	$-16384 \leq x \leq 16383.5$
Q_0	$S_{15,0}$	$-32768 \leq x \leq 32767$

由表 1.3 可见,同一个 16 位数,若小数点设定不同,其表示的数据大小截然不同,精度也不同。Q 越大,表示的数值范围越小(绝对值小于 2^{15-Q}),但精度越高(小数精度为

$1/2^Q$)。相反 Q 越小,表示的数值范围越大,但精度越低。因此,DSP 的字长一定,数值范围与精度是一对矛盾,在实际中必须考虑这一点。

定点数和浮点数有以下转换关系:

浮点数(x)转换为定点数(x_q): $x_q = \text{int}[x \times 2^Q]$

定点数(x_q)转换为浮点数(x): $x = \text{float}[x_q \times 2^{-Q}]$

例如,浮点数 $x=0.4$,定标 $Q=15$,则对应定点数为 $x_q = \text{int}[0.4 \times 2^{15}] = 13107$ 。反之,一个 $Q15$ 表示的定点数 13107 对应的浮点数为: $13107 \times 2^{-15} \approx 0.3999939$ 。

在实际应用中, Q 数值由以下方法确定。

设系统中变量表示的数据最大绝对值为 $|\max|$,而且 $|\max|$ 小于或等于 32767,由下式:

$$2^{n-1} \leq |\max| \leq 2^n$$

可得 $Q=15-n$ 。

例如,某变量取值范围为 -1 到 1 。那么变量的 $|\max|=1$, $n=0$,则 $Q=15$ 。

1.4.2 DSP 定点算术运算

在 DSP 运算中经常用到 Q 表示法。由以上介绍可知,16 位二进制数的 Q 表示法,高位为符号位,依此为 $15-Q$ 位的整数位和 Q 位的小数位。其表示的数据绝对值不超过 2^{15-Q} ,精度为 $1/2^Q$ 。

根据数据的范围和精度要求,不同应用 Q 值一般不同。但通常情况更多的是将数表示为用 $Q15$ 格式表示的纯小数和 $Q0$ 格式表示的整数,这样一来,有利于乘法等运算,即小数乘以小数结果仍为小数,整数乘以整数结果仍为整数。

1. 两个定点数的算术运算

(1) 两个定点数的加/减法

定点加/减法必须保证两个操作数的定标值一样。如果两个数据的 Q 值不同,在保证数据准确性的前提下调整 Q 值使数据精度最高,即尽量将 Q 值小的数调整为与另一个数的 Q 一样大。另外,做加/减运算时,必须注意运算结果可能超出 16 位的表示范围。

例如, $x=0.4$, $y=0.2$,计算 $x+y$ 。

根据 Q 法的表示范围,采用 $Q15$ 表示两个数据可以得到最高精度的运算结果。

所以 x 、 y 的 $Q15$ 定点表示分别为: $x_q=13107$, $y_q=6553$ 。

$x_q + y_q = 13107 + 6553 = 19660$ 。

将运算结果转换为浮点数为: $19660 \times 2^{-15} \approx 0.5999756$ 。

当进行加/减法运算的数据范围无法确定时,一定注意对溢出的判断和处理。

(2) 定点数的乘法

两个 16 位定点数的乘法分以下几种情况。

1) 小数乘小数(即数用 $Q15$ 表示)

$Q15 \times Q15 = Q30$,即 32 位的乘积结果有 2 位符号位,利用 DSP 的移位操作很容易得到 $Q31$,取结果的高字得到 $Q15$ 的乘积结果。

例如, $0.5 \times 0.5 = 0.25$

操作数用 Q15 表示为: $0.5 \times 2^{15} = 16384 = 0100\ 0000\ 0000\ 0000\text{B}$

$$\begin{array}{r} 0.100\ 0000\ 0000\ 0000 \qquad ;Q15 \\ \times \quad 0.100\ 0000\ 0000\ 0000 \qquad ;Q15 \\ \hline 00.01\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000 \qquad ;Q30 \end{array}$$

将运算结果左移一位得到乘积结果的 Q31 表示为 20000000H, 浮点数为 0.25

2) 整数乘整数(即数用 Q0 表示法)

$$Q0 \times Q0 = Q0$$

$$\begin{array}{r} \text{例如, } 13 \times 5 = 65 \\ 0000\ 0000\ 0000\ 1101 \qquad ;Q0 \\ \times \quad 0000\ 0000\ 0000\ 0101 \qquad ;Q0 \\ \hline 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0100\ 0001 \qquad ;Q0 \end{array}$$

如果乘积运算结果不超出 16 位二进制数范围, 则得到的乘积低位字为 16 位的 Q0 表示法。

两个 16 位定点数相乘, 乘积结果最多为 32 位。

3) 混合表示法(操作数采用 Q0 与 Q15 之间的表示法)

在对精度和数据范围要求都比较高的情况下, 就必须采取介于 Q15 与 Q0 之间的 Q 表示法, 即混合表示。如果这时的两个操作数分别用 Q_m 和 Q_n 表示, 则乘积结果一般为 Q_{m+n} , 整数位为 $32 - (m + n)$, 如果数据的动态范围预先可以知道, 一般可以提高数的精度。

例如, 参与运算的两个数 x 和 y 的数值范围分别不超过 ± 1 和 ± 4 , 那么为了保证数据范围的同时使精度尽量高, 应该采用 Q13 表示法。比如, $x = 0.4$, $y = 3.2$, 计算 $x + y$ 。

因为, x 、 y 的 Q13 定点表示分别为: $x_q = 3276$, $y_q = 26214$ 。

$$x_q + y_q = 3276 + 26214 = 29490。$$

运算结果转换为浮点数为: $29490 \times 2^{-13} \approx 3.59985$ 。

DSP 没有专门的除法指令, 利用条件减法指令编写子程序实现。

2. 非线性运算

在 DSP 应用中, 经常会遇到非线性运算, 如三角函数运算和指数运算等, 这些运算一般有: 调用 DSP 编译软件的库函数 math.h、查表法和混合法。

在 C 语言编程中, 调用库函数非常简单方便, 但运算量大, 在目前的 DSP 控制器应用中较少使用。

查表法是快速实现非线性运算最常用的方法, 而且精度要求越高, 表格相应就越大, 需要的存储器容量也越大, 一般要根据系统存储器情况和数据精度要求确定表格大小。查表法比较适合于周期性函数。在第八章的实验三中就采用了查表法。

混合法是在存储器容量和精度要求不能同时满足时采用的一种方法, 这种方法比较灵活, 最常用的方法是在查表的基础上结合少量的运算(折线近似), 插入一些中间点来提高表格两点之间的精度, 这种方法在第八章的实验三中也体现。

第二章 TMS320C2××的 CPU 结构和存储器配置

TMS320C2××是 16 位的定点 DSP,运算速度高达 40MIPS,新成员'C28×的运算速度可达 400MIPS。优良的性能价格比使其应用领域不断拓展,甚至扩展到传统的微控制器应用领域。

2.1 TMS320C2××系列 DSP 结构

2.1.1 概述

'C2××使用改进哈佛结构,将程序存储器和数据存储器的总线分开,以便最大限度

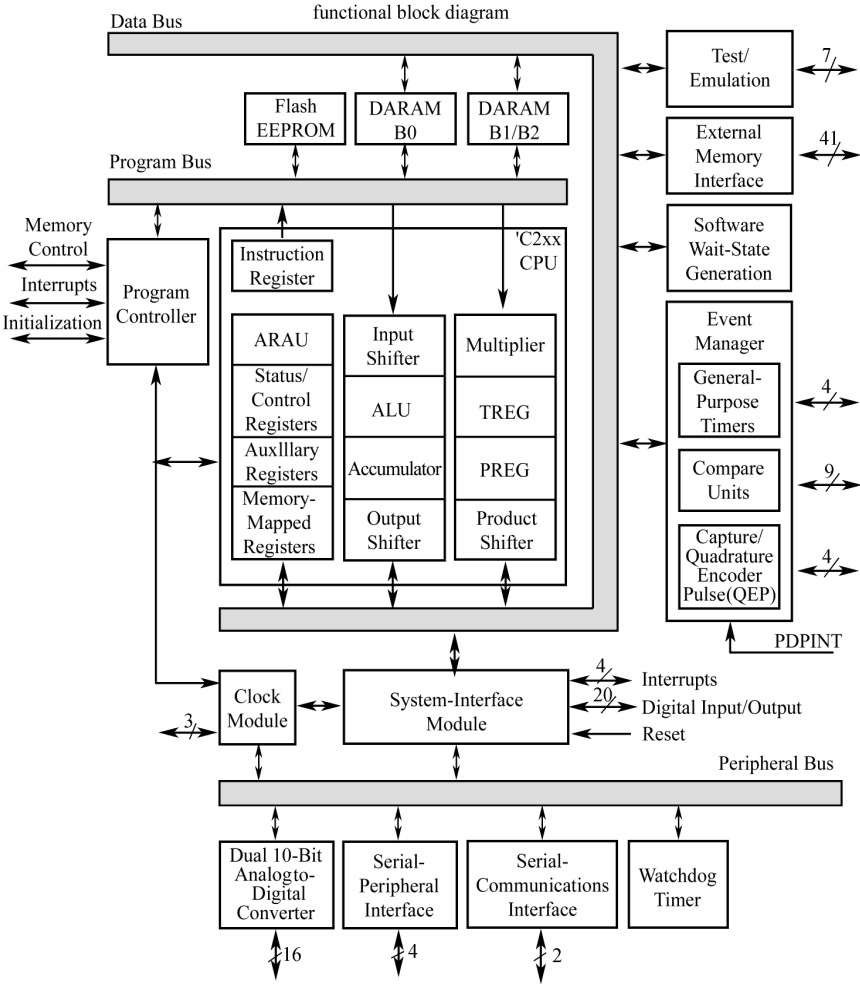


图 2.1 'C2××系列的功能框图 1(图中外设为'×240 系列)

地提高处理能力。图 2.1 和图 2.2 分别是 TMS320C2××的两种结构框图。图 2.3 为引脚分布图。图 2.1 体现各模块及总线间连接关系。图 2.2 给出各功能模块与引脚的清晰关系。

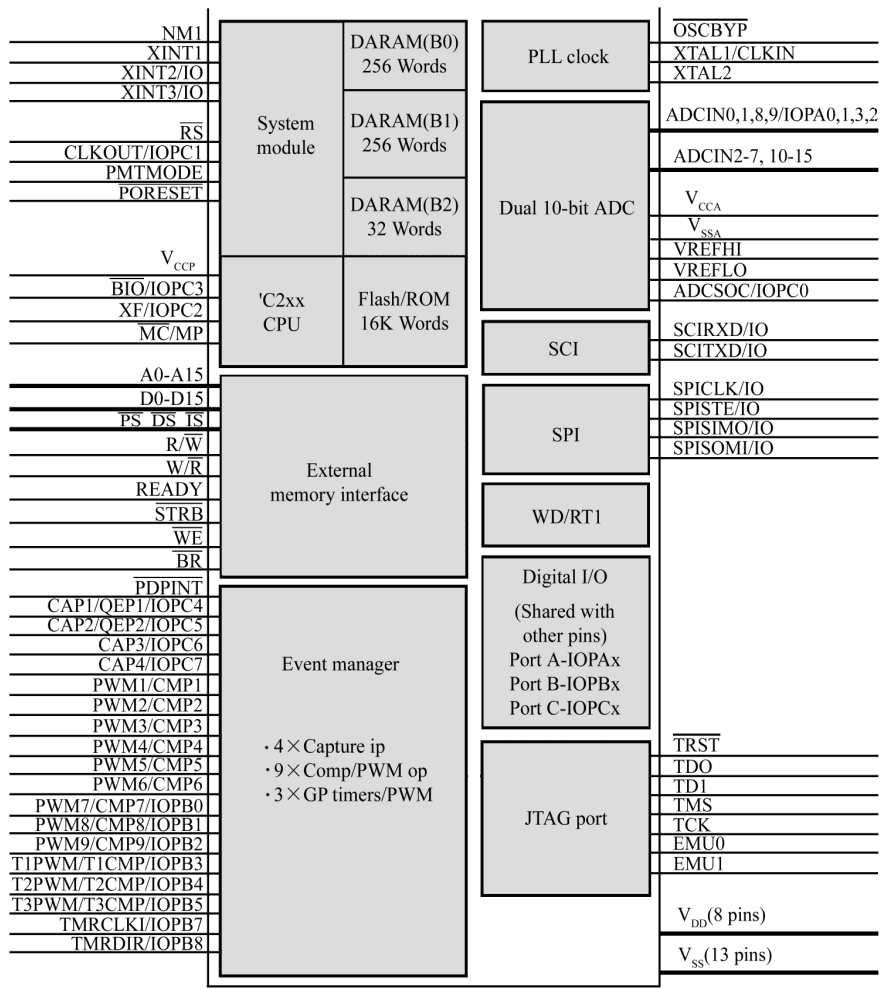


图 2.2 'C2××系列的功能框图 2(图中外设为'×240 系列)

'C2××器件主要包括三个主要的功能单元:'2××DSP 的 CPU(或称为 DSP 内核)、内部存储器和外围设备。所有'C2××系列产品都具有相同的 CPU。本章仅对 CPU 和内部存储器部分做详细介绍,其他部分的分析放在介绍完指令系统以后进行。

除了上述三个功能单元,'C2××DSP 还有几个系统级部件。这些部件包括外部存储器控制接口、器件复位、中断、数字输入/输出(I/O)、时钟产生、低功耗操作等。为了更好地理解这些模块,有些内容安排在第三、四两章之后,便于配合软件,深入了解各功能模块。

的老化、部件间相容和漂移等因素。

高速的 CPU 使数字控制系统能够实时处理数据。'C24×DSP 的指令集包括信号处理指令和通用控制指令,同时还提供了扩展的开发支持,当用户从其他通用 TMS320 定点 DSP 升级时,该指令集能够保护以往的开发软件。'C24×源码和目标码均兼容于'C2××成员,同时源码还向下兼容'C2×,向上兼容'C5×系列。

'C24×的体系结构也适用于处理控制信号,如 16 位字长、用于存放临时数据的 32 位寄存器和完全独立于 CPU 的数字定标硬件移位器。上述特点的结合使量化和截尾误差最小,同时'C24×还增强了用于辅助功能的处理能力。这些辅助功能包括用于取消系统中机械共振的陷波滤波器,或者用于减小系统中状态传感器的估计技术。

另外,'C24×DSP 控制器强大的外设功能,使得 TI 可以为应用提供不同性价比的方案。

DSP 控制器的外设还正在继续丰富和调整,以适应嵌入式控制系统应用的需要。

TMS320C240 是'C24×系列的第一代产品。具有用于高速信号处理和数字控制功能所必需的结构,同时还具有单片电机控制所需的外设功能。'C24×采用亚微米 CMOS 制造工艺,降低功耗,此外,还具有进一步降低功耗的工作模式。'F240 与'C240 结构基本相同,一般用'×240 代表 TMS320C240 和 TMS320F240,除非另有注明,这两个 DSP 芯片的指令运算能力为 20MIPS。表 2.1 是两个器件特点比较。从表中可见,两者之间的惟一区别在于程序存储器的类型不同,'C240 有 16K 字的 ROM 而'F240 有 16K 字的闪速存储器。

表 2.1 TMS320×240 控制器比较

'×240	片内存储器(字)				电源电压 (V)	周期 (ns)	封装及引脚
	数据 RAM	数据/程序 RAM	ROM	闪速 E ² PROM			
'C240	288	256	16K	0	5	50	PQ 132P
'F240	288	256	0	16K	5	50	PQ 132P

'×240 采用 132 引脚塑料四方扁平封装(Plastic Quad Flat Package),如图 2.3 所示。与 TTL 电平兼容。

'×240 具有改进的哈佛结构,可以通过外部总线分别访问数据、程序、输入/输出三个独立的空间,具体技术特点如下:

- (1) 高性能静态 CMOS 技术,4 种用于减少功耗的省电方式
- (2) 包括 TMS320C2××内核 CPU
 - 与'C2××、'C2×和'C5×源码兼容;
 - 50ns 的指令周期,且大多数指令为单周期指令。
- (3) 存储器
 - 片内有 544 字(544×16 位)双口 RAM;
 - 16K 字(16K×16 位)的片内程序 ROM('C240)/闪速存储器 Flash('F240);
 - 共 224K 字的存储器寻址空间(64K 字的程序空间,64K 字的数据空间,64K 字的 I/O 空间和 32K 字的全局空间)。

- (4) 事件管理器 (Event-manager module, 简称 EV)
 - 12 个比较/脉宽调制(PWM)通道(其中 9 个相互独立);
 - 3 个 16 位通用定时器,有 6 种工作模式,包括连续递加/减计数;
 - 3 个 16 位全比较单元,有死区功能 Deadband;
 - 3 个 16 位简单比较单元;
 - 4 个捕获单元,其中 2 个具有直接连接正交编码器脉冲的能力。
 - (5) 双 10 位模数转换器(ADC)
 - (6) 28 个独立可编程多路复用 I/O 引脚
 - (7) 基于锁相环(PLL)的时钟模块
 - (8) 带实时中断(RTI)的看门狗(WD)定时器模块
 - (9) 异步串行通信接口 SCI
 - (10) 同步串行外设接口 SPI
 - (11) 基于扫描的仿真 JTAG
 - (12) 可用的开发工具
 - TI ANSI C 编译器,汇编/连接器和 C 源代码调试器 CCS(Code Composer Studio);
 - 全范围的仿真产品:自仿真(SDS510)Scan-Based Self-Emulation (XDS510E);
 - 有 JTAG 仿真的仿真模块;
 - 支持第三方数字电机控制和模糊逻辑开发。
- 本教材主要以 F240 为模型介绍 C2000 系列 DSP。

2.2 'C2××的总线结构

'C2××的内部将程序、数据读和写总线分开,构成 6 组 16bit 总线,如图 2.4 所示。由图可见,'C2××的片内总线有:

- PAB(Program Address Bus)——程序地址总线。它提供了访问程序存储区的地址。
- DRAB(Data-Read Address Bus)——数据读地址总线。它提供了从数据存储器读取数据的地址。
- DWAB(Data-Write Address Bus)——数据写地址总线。它提供了写数据存储器的地址。
- PRDB(Program Read Bus)——程序读总线。它载有从程序存储器读取的指令代码以及表格等信息等,并送到 CPU。
- DRDB(Data Read Bus)——数据读总线。它将数据从数据存储器载送到中心算术逻辑单元(CALU)和辅助寄存器算术单元(ARAU)。
- DWEB(Data Write Bus)——数据写总线。它将数据载送至程序存储器和数据存储器。

独立的数据读地址总线(DRAB)和数据写地址总线(DWAB)使得 CPU 可以在同一个机器周期内读和写操作。独立的程序和数据地址总线允许同时访问程序指令和数据。例如,数据在作乘法时,前面的乘积可以加给 ACC,与此同时,产生一个新的地址。这种

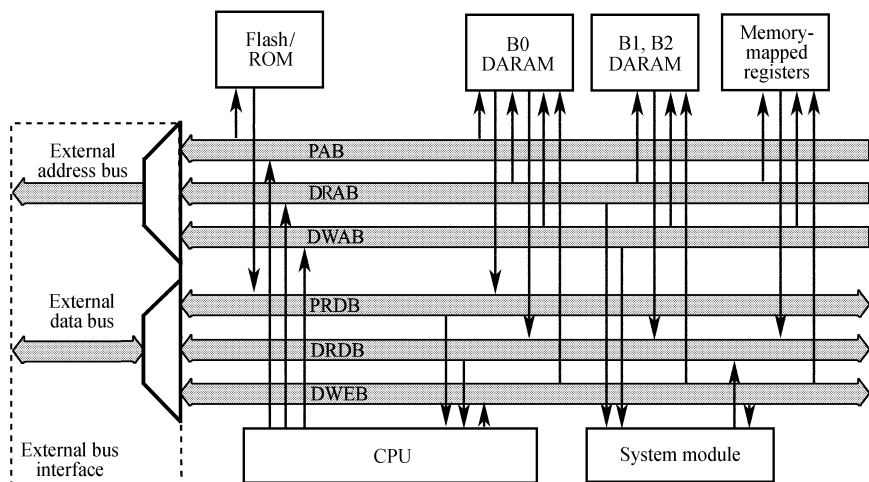


图 2.4 C2××总线结构

并行机制可以在单个机器周期内完成一组算术、逻辑、位操作运算。此外,C2××还具有管理中断、循环操作、函数/子程序调用的控制机制。

'C2××系列中的各芯片都具有同样的 CPU 和总线结构。但不同的芯片具有不同的片内存储器划分和片内外设。

2.3 'C2××系列 CPU 结构

'C2××的各芯片具有同样的 CPU,图 2.5 为 CPU 的基本结构框图。'C2××的 CPU 主要包括以下一些功能模块:

- 一个 32bit 的中央算术逻辑单元(CALU);
- 一个 32bit 的累加器(ACC);
- CALU 的输入和输出数据定标移位器;
- 一个 16×16bit 的乘法器;
- 一个乘积定标移位器。

另外,CPU 还包括一个辅助寄存器运算单元(ARAU),图中没有画出。该单元执行与中央算术逻辑单元无关的算术运算。其中包括 8 个辅助寄存器(AR0~AR7)和一个辅助寄存器算术单元(ARAU),这部分提供了灵活而强大的间接寻址能力。此外,CPU 还有两个状态寄存器 ST0 和 ST1,它们包含有决定处理器方式、地址指针值以及指示处理器不同条件和算术逻辑运算结果的位。因此,CPU 共用 11 个寄存器:ACC、ST0、ST1、AR0~AR7。

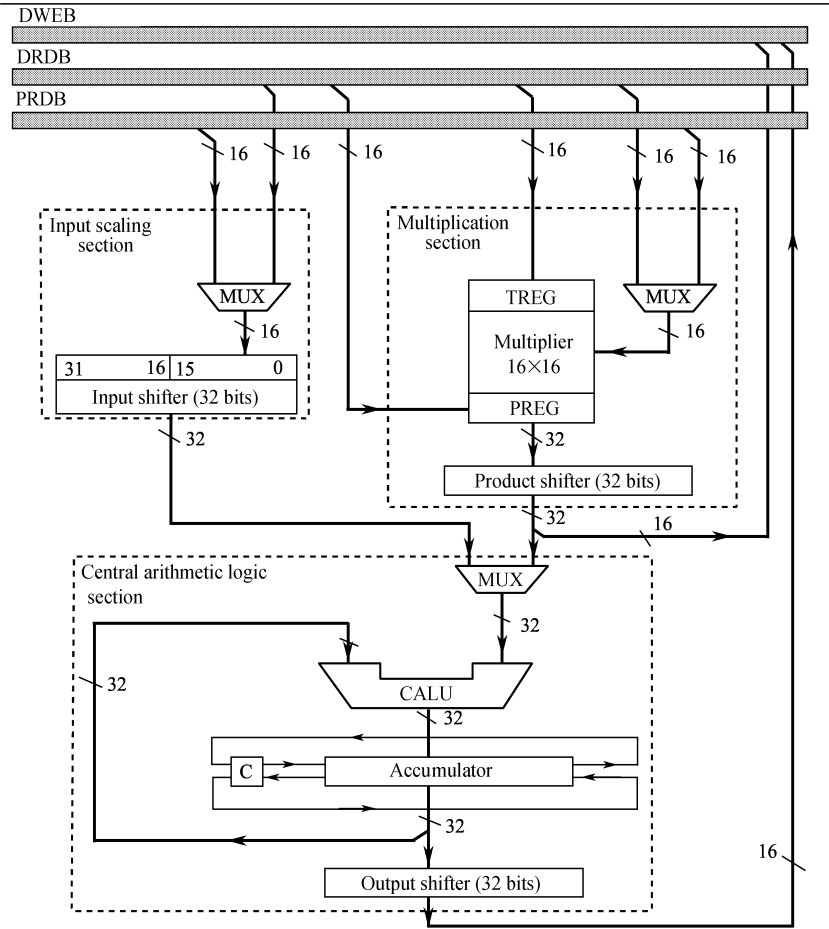


图 2.5 TMS320C2×系列 CPU 的基本结构框图

2.3.1 中央算术逻辑单元(CALU)和累加器(ACC)

C2××可以利用其 32bit 的 CALU 来进行二进制算术运算。CALU 使用从数据存储器或指令中直接取得的 16bit 字长的数,或者从乘法器取得的 32bit 的乘积。除了算术运算之外,CALU 还可以作布尔运算,其结构如图 2.6 所示。

累加器存储 CALU 的运算结果,它也可以向 CALU 提供第二个输入。ACC 的宽度为 32bit,分为高 16bit(bit 31 到 16)和低 16bit(bit 15 到 0)。汇编语言指令可以分别将高 16bit 和低 16bit 存入数据存储器。

2.3.2 定标移位器

- C2××有 3 个 32bit 的移位器,可以作移位、bit 提取、扩展算术运算和溢出防止运算。
- 输入数据定标移位器(输入移位器)。该移位器将输入的 0 到 16bit 的数据左移

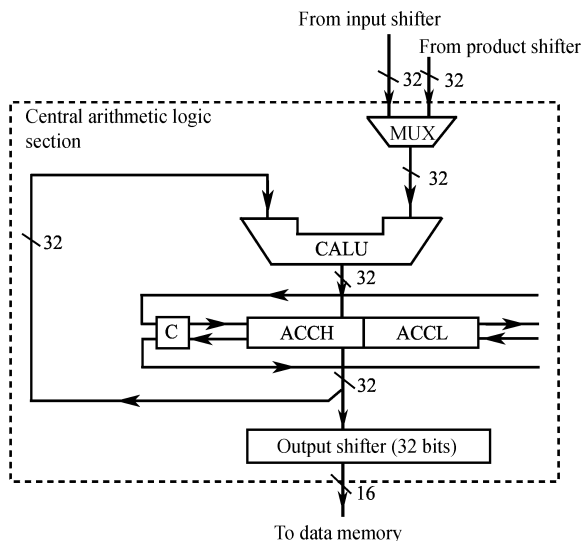


图 2.6 中央算术逻辑功能图

16bit, 作为 CALU 的 32bit 输入的数据

- 输出数据定标移位器(输出移位器)。该移位器将累加器里的结果输出到数据存储器之前左移 0 到 7bit。累加器里的数据保持不变。
- 乘积定标移位器(乘积移位器)。乘积寄存器(PREG)接受乘法器的输出。而乘积移位器在乘积寄存器的数据输出到 CALU 之前对其作移位。乘积移位器具有 4 种移位模式(不移位、左移 1bit、左移 4bit、右移 6bit)。这些模式对于乘/加运算、小数算术运算、以及有理小数乘积等运算都是很有用的。

2.3.3 乘法器

片内乘法器作 $16 \times 16\text{bit}$ 二进制乘法, 得到 32bit 的结果。在与乘法器连接时, 使用一个 16bit 的暂时寄存器(TREG)和 32bit 的乘积寄存器(PREG)。TREG 总是保存乘数中的一个, 而 PREG 则接受每次乘法的结果。

使用乘法器、暂时寄存器和乘积寄存器, C2××就能有效地进行基本的 DSP 运算, 诸如卷积、相关、滤波等。每次乘法的有效执行时间最短是一个指令周期。大大提高了 DSP 信号处理能力。

2.3.4 辅助寄存器算术单元(ARAU)和辅助寄存器

当一个指令使用间接寻址来访问数据存储器时, ARAU 就产生数据存储器的地址。ARAU 是由 8 个辅助寄存器(AR0 到 AR7)支持的, 其中每一个寄存器都可以从数据存储器或直接从指令字中装入一个 16bit 的值。每个辅助寄存器中的值也可以存入数据存储器。这些辅助寄存器由嵌入在状态寄存器 ST0 中的一个 3bit 的辅助寄存器指针(ARP)来查找(即由 ARP 指定使用哪一个 AR)。

2.3.5 状态寄存器 ST0 和 ST1

DSP CPU 的两个状态寄存器 ST0 和 ST1 可以被保存到数据存储器或从数据存储器加载,常用的指令有 LST(装载状态寄存器)和 SST(存储状态寄存器),另外 SETC 和 CLRC 可以设置和清除这两个寄存器的许多位,即对 ST0 和 ST1 进行位操作。ST0 和 ST1 可以用于子程序中保存和恢复机器的状态。两个状态寄存器的格式分别如图 2.7 和图 2.8 所示。

D ₁₅ ~D ₁₃	D ₁₂	D ₁₁	D ₁₀	D ₉	D ₈ ~D ₀
ARP	OV	OVM	1 ⁺	INTM	DP
R/W-x	R/W-0	R/W-x		R/W-1	R/W-x

(注:R=读访问;W=写访问;符号-之后的值为该位的复位值,x 说明复位对该位没影响
1⁺该保留位(reserved),读为 1,写无影响)

图 2.7 CPU 状态寄存器 ST0

其中:

ARP——辅助寄存器指针,用于选择作为间接寻址的辅助寄存器(ARn)。当装载 ARP 时,ARP 的旧值被复制到 ST1 中的 ARB 中,LST、MAR 及所有的间接寻址指令都可以修改 ARP,当 LST 指令装载 ARB 时,同时将 ARB 的内容装入 ARP。

OV——溢出标志位,用于表示算术运算单元 CALU 是否有溢出发生。当 CALU 溢出,OV 置 1 一旦溢出发生,OV 保持置位状态直到下列条件发生才能清除:复位、以 OV 或 NOV 为条件的条件转移指令或 LST 指令。

OVM——溢出方式位。决定如何处理 CALU 中的溢出,用 SETC 和 CLRC 置 1 和清 0 该位。

OVM=0 累加器中结果正常溢出;

OVM=1 当溢出时,累加器被置为最大正数或最小负数使误差最小。

INTM——中断允许位。使能或禁止所有的可屏蔽中断,INTM 不受 LST 指令影响。

INTM=0 允许所有可屏蔽中断;

INTM=1 禁止所有可屏蔽中断。

DP——数据页指针。当使用直接寻址时,DP 存放存储器的数据页(即存储器地址的高 9 位),DP 与指令代码的最低 7 位构成 16 位存储器地址。

D ₁₅ ~D ₁₃	D ₁₂	D ₁₁	D ₁₀	D ₉	D ₈	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁ ~D ₀
ARB	CNF	TC	SXM	C	1 ⁺	1 ⁺	1 ⁺	1 ⁺	XF	1 ⁺	1 ⁺	PM
R/W-x	R/W-0	R/W-x	R/W-1	R/W-1					R/W-1			R/W-00

(注:R=读访问;W=写访问;符号-之后的值为该位的复位值,x 说明复位对位没有影响
1⁺该保留位(reserved),读为 1,写无影响)

图 2.8 CPU 状态寄存器 ST1