

- 学习DSP芯片的软、硬件设计方法
- 掌握DSP芯片的最新开发手段

张雄伟 编著

DSP

芯片的原理与开发应用



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.co.cn>

DSP 芯片的原理与开发应用

张雄伟 编著

電子工業出版社
Publishing House of Electronics Industry

内 容 简 介

可编程 DSP 芯片是一种应用非常广泛的微处理器。本书比较全面系统地介绍了 DSP 芯片的基本原理、开发和应用。首先介绍了目前广泛使用的 DSP 芯片的基本结构和特征,定点和浮点 DSP 处理中的一些关键问题。然后,以目前应用最广的 TI DSP 芯片为例,介绍了定点和浮点 DSP 芯片的软硬件设计方法、DSP 芯片的 C 和汇编语言的开发方法以及 DSP 芯片的开发工具及使用。通过两个应用系统介绍了定点和浮点 DSP 芯片的开发过程,并介绍了数字滤波器和 FFT 等常用数字信号处理算法的 DSP 实现。

本书的特点是:例子丰富、内容新颖和实用性强。

本书旨在使读者在了解 DSP 芯片基本原理的基础上,掌握 DSP 芯片的系统设计和软硬件开发方法。可供通信和电子等领域从事 DSP 芯片开发应用的广大科技人员和教师阅读参考,也可作为相关专业研究生和高年级本科生的教材。

书 名: DSP 芯片的原理与开发应用

编 著 者: 张雄伟

责任编辑: 张 欣

排版制作: 电子工业出版社计算机排版室

印 刷 者: 北京科技大学印刷厂

出版发行: 电子工业出版社出版、发行 URL: <http://www.phei.co.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张: 15.375 字数: 394 千字

版 次: 1997 年 9 月第 1 版 1997 年 9 月第 1 次印刷

书 号: ISBN 7-5053-4269-X
TN·1090

定 价: 22.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·翻印必究

前 言

DSP 芯片,也称数字信号处理器,是一种具有特殊结构的微处理器。DSP 芯片的内部采用程序和数据分开的哈佛结构,具有专门的硬件乘法器,广泛采用流水线操作,提供特殊的 DSP 指令,可以用来快速实现各种数字信号处理算法。

自 80 年代初 DSP 芯片诞生以来,在短短的十多年时间里它得到了飞速的发展。随着 DSP 芯片性能价格比和开发手段的不断提高,DSP 芯片已经在通信与电子系统、信号与信息处理、自动控制、雷达、军事、航空航天、医疗、家用电器等许多领域得到广泛的应用。

DSP 芯片可分为通用型和专用型两大类。通用型 DSP 芯片是一种软件可编程的 DSP 芯片,可适用于各种 DSP 应用。专用型 DSP 芯片则将 DSP 处理的算法集成到 DSP 芯片内部,一般用于某些专用的场合。本书主要讨论通用型 DSP 芯片。

目前,DSP 芯片的主要供应商有美国的德州仪器公司(TI)、AD 公司、AT&T 公司和 Motorola 公司等。其中,TI 公司占世界 DSP 芯片市场的 50%左右,在国内也被广泛的采用。因此,本书在开发应用部分主要以 TI 公司的 DSP 芯片为例进行介绍。

本书共十三章。第一章概述了 DSP 芯片的发展、分类、选择和应用。第二章介绍 DSP 芯片的基本结构和主要特征。第三章介绍了定点 DSP 处理中的定标和浮点到定点的转换方法。第四章介绍了浮点 DSP 处理中的浮点数格式和快速浮点运算的方法等内容。

第五章和第六章分别介绍了 TMS320C2X/C5X 定点 DSP 芯片和 TMS320C3X/C4X 浮点 DSP 芯片的软硬件设计方法,包括存储器设计、模数转换接口设计、编程技巧和有关软件应用等内容。第七章介绍了广泛采用的公共目标文件格式(COFF)和编程方法。第八章介绍了 DSP 芯片的开发工具,包括代码生成工具和代码调试工具,着重介绍了 C 编译器和 C/汇编源码调试器等工具的使用方法。第九章介绍了用 C 语言开发 DSP 芯片的方法。第十章介绍了采用 C 和汇编混合编程开发 DSP 芯片的方法。

第十一章以 TMS320C25 和 TMS320C31 为例介绍了两个应用系统的开发方法及开发过程。第十二章介绍了数字滤波器的 DSP 实现,分别以 TMS320C2X 和 TMS320C3X 为例介绍了 FIR、IIR 和自适应滤波器的定点 DSP 和浮点 DSP 实现方法。第十三章介绍了 FFT 的定点和浮点 DSP 实现。

本书是作者近十年来从事 DSP 芯片应用开发的总结。在本书的编写过程中,得到了成立新副教授和关存太副教授的大力支持和帮助,张化云、甘斌、徐旭、李旺、蒋冰心等同志为本书做出许多工作,在此一并向他们表示衷心的感谢。

由于作者水平所限,书中错误之处在所难免,恳请广大读者给予批评指正。

作 者

1997 年 5 月于解放军南京通信工程学院

目 录

第一章 概述	(1)
1.1 引言	(1)
1.2 DSP 系统	(2)
1.2.1 DSP 系统构成	(2)
1.2.2 DSP 系统的特点	(2)
1.2.3 DSP 系统的设计过程	(3)
1.3 可编程 DSP 芯片	(4)
1.3.1 什么是 DSP 芯片	(4)
1.3.2 DSP 芯片的发展	(4)
1.3.3 DSP 芯片的分类	(6)
1.3.4 DSP 芯片的选择	(7)
1.3.5 DSP 芯片的应用	(8)
第二章 DSP 芯片的基本结构和特征	(10)
2.1 引言	(10)
2.2 DSP 芯片的基本结构	(10)
2.2.1 哈佛结构	(10)
2.2.2 流水线	(10)
2.2.3 专用的硬件乘法器	(11)
2.2.4 特殊的 DSP 指令	(11)
2.2.5 快速的指令周期	(12)
2.3 TI 定点 DSP 芯片	(12)
2.3.1 TMS320C1X	(12)
2.3.2 TMS320C2X	(14)
2.3.3 TMS320C5X	(16)
2.3.4 TMS320C2XX	(19)
2.3.5 TMS320C54X	(20)
2.4 TI 浮点 DSP 芯片	(21)
2.4.1 TMS320C3X	(21)
2.4.2 TMS320C4X	(24)
2.5 多处理器 DSP 芯片 TMS320C8X	(28)
2.6 其它 DSP 芯片简介	(30)
2.6.1 AD 公司的 DSP 芯片	(30)
2.6.2 AT&T 公司 DSP 芯片	(31)

2.6.3	Motorola 公司 DSP 芯片	(32)
2.7	小结	(33)
第三章	DSP 芯片的定点运算	(34)
3.1	数的定标	(34)
3.2	高级语言:从浮点到定点	(35)
3.2.1	加法/减法运算的 C 语言定点模拟	(35)
3.2.2	乘法运算的 C 语言定点模拟	(37)
3.2.3	除法运算的 C 语言定点模拟	(38)
3.2.4	程序变量的 Q 值确定	(38)
3.2.5	浮点至定点变换的 C 程序举例	(39)
3.3	DSP 定点算术运算	(41)
3.3.1	定点乘法	(41)
3.3.2	定点加法	(42)
3.3.3	定点除法	(43)
3.4	非线性运算的定点快速实现	(45)
3.5	小结	(49)
第四章	DSP 芯片的浮点运算	(50)
4.1	引言	(50)
4.2	浮点数的格式	(50)
4.2.1	IEEE 浮点数格式	(50)
4.2.2	TMS320C3X 浮点数格式	(51)
4.2.3	IEEE 浮点格式与 TMS320 浮点格式的转换	(52)
4.3	基本的浮点运算	(54)
4.3.1	浮点乘法和加减法	(54)
4.3.2	浮点除法	(54)
4.4	非线性浮点运算的快速实现	(55)
4.5	小结	(57)
第五章	TMS320 定点 DSP 的软硬件设计	(58)
5.1	引言	(58)
5.2	TMS320C2X 硬件设计	(58)
5.2.1	复位和时钟电路	(58)
5.2.2	等待状态发生器	(59)
5.2.3	存储器接口	(60)
5.2.4	模数接口电路	(63)
5.2.5	通信接口电路设计举例	(64)
5.3	主从式系统的硬件设计	(67)
5.3.1	并行通信实现双机通信	(67)

5.3.2	DMA 实现双机通信	(68)
5.4	TMS320C50 的 BOOT 设计	(70)
5.5	TMS320C2X/C5X 的软件应用	(72)
5.5.1	TMS320C2X 软件编程技巧	(72)
5.5.2	提高 TMS320C5X 软件效率	(73)
5.5.3	软件应用	(75)
5.6	小结	(79)
第六章	TMS320 浮点 DSP 芯片的软硬件设计	(80)
6.1	引言	(80)
6.2	TMS320C3X 的硬件设计	(80)
6.2.1	存储器接口设计	(80)
6.2.2	中断电路的设计	(83)
6.2.3	模数转换接口电路	(84)
6.3	TMS320C3X 主从式系统硬件设计	(93)
6.4	TMS320C31 程序引导功能的实现	(95)
6.5	TMS320C3X 的软件应用	(98)
6.5.1	编程技巧	(98)
6.5.2	TMS320C3X 软件应用	(100)
6.6	小结	(102)
第七章	COFF - 公共目标文件格式	(103)
7.1	引言	(103)
7.2	COFF 目标文件格式	(103)
7.2.1	块(section)	(103)
7.2.2	汇编器对块的处理	(104)
7.2.3	链接器对块的处理	(105)
7.2.4	程序重定位	(107)
7.2.5	COFF 文件中的符号	(107)
7.3	COFF 格式编程举例	(108)
7.4	小结	(109)
第八章	DSP 芯片的开发工具及应用	(110)
8.1	引言	(110)
8.2	代码生成工具	(110)
8.2.1	代码生成工具程序	(110)
8.2.2	宏汇编器和链接器	(112)
8.2.3	C 编译器	(113)
8.2.4	文档管理器和库生成器	(114)
8.2.5	代码格式转换器	(115)

8.2.6 编译汇编链接应用举例	(116)
8.3 代码调试工具	(117)
8.3.1 C/汇编源码调试器	(117)
8.3.2 初学者工具 DSK	(122)
8.3.3 软件模拟器	(123)
8.3.4 评价模块(EVM)	(125)
8.3.5 软件开发系统(SWDS)	(125)
8.3.6 仿真器(XDS)	(125)
8.4 小结	(127)
第九章 用 C 语言开发 DSP 芯片	(128)
9.1 引言	(128)
9.2 优化 ANSI C 编译器	(128)
9.3 TMS320C3X/C4X 的 C 语言开发	(132)
9.3.1 存储器模式	(132)
9.3.2 寄存器规则	(134)
9.3.3 函数结构与调用规则	(137)
9.3.4 中断处理	(139)
9.3.5 系统初始化	(140)
9.3.6 TMS320C3X C 程序开发举例	(143)
9.4 TMS320C2X/C5X 的 C 语言开发	(145)
9.4.1 存储器模式	(145)
9.4.2 寄存器规则	(146)
9.4.3 函数调用规则	(147)
9.4.4 中断函数	(149)
9.4.5 表达式分析	(150)
9.4.6 TMS320C2X/C5X C 语言程序开发举例	(150)
9.5 小结	(153)
第十章 DSP 芯片的 C 语言和汇编语言混合编程	(154)
10.1 引言	(154)
10.2 C 和汇编的混合编程方法	(154)
10.2.1 独立的 C 和汇编模块接口	(154)
10.2.2 从 C 程序中访问汇编程序变量	(156)
10.2.3 在汇编程序中访问 C 程序变量	(157)
10.2.4 在 C 程序中直接嵌入汇编语句	(158)
10.2.5 修改编译器的输出	(160)
10.3 TMS320C3X/C4X 混合编程举例	(161)
10.4 TMS320C2X/C5X 混合编程举例	(162)
10.5 小结	(163)

第十一章 DSP 芯片应用开发举例	(164)
11.1 引言	(164)
11.2 一个基于 TMS320C25 DSP 应用系统的开发	(164)
11.2.1 系统简介	(164)
11.2.2 系统硬件设计	(165)
11.2.3 系统软件设计	(168)
11.2.4 硬件调试	(168)
11.2.5 软件调试	(171)
11.2.6 独立系统实现	(172)
11.3 一个基于 TMS320C31 DSP 系统的开发	(174)
11.3.1 系统简介	(174)
11.3.2 系统构成	(174)
11.3.3 系统软硬件设计	(175)
11.3.4 软硬件调试	(177)
11.3.5 独立系统实现	(179)
11.4 小结	(180)
第十二章 数字滤波器的 DSP 实现	(181)
12.1 引言	(181)
12.2 FIR 滤波器的 DSP 实现	(181)
12.2.1 FIR 滤波器的基本原理和设计方法	(181)
12.2.2 FIR 滤波器的定点 DSP 实现	(184)
12.2.3 FIR 滤波器的浮点 DSP 实现	(185)
12.3 IIR 滤波器的 DSP 实现	(187)
12.3.1 IIR 滤波器的基本原理和设计方法	(187)
12.3.2 IIR 滤波器的定点 DSP 实现	(190)
12.3.3 IIR 滤波器的浮点 DSP 实现	(192)
12.4 自适应滤波器的 DSP 实现	(195)
12.4.1 自适应滤波器的基本原理	(195)
12.4.2 自适应滤波器的定点 DSP 实现	(196)
12.4.3 自适应滤波器的浮点 DSP 实现	(199)
12.5 小结	(200)
第十三章 FFT 的 DSP 实现	(201)
13.1 引言	(201)
13.2 FFT 的基本原理	(201)
13.2.1 DFT	(201)
13.2.2 FFT 算法的导出	(202)
13.3 FFT 算法的高级语言实现	(206)

13.3.1 FFT 算法的 Fortran 语言实现	(206)
13.3.2 FFT 算法的 C 语言实现	(207)
13.4 FFT 的定点 DSP 实现	(209)
13.5 FFT 的浮点 DSP 实现	(215)
13.6 小结	(223)
结束语	(224)
参考文献	(225)
附录 A TI 格式文件转化为二进制文件	(227)
附录 B 8 位 μ 律/16 位线性互换的 C 语言子程序	(231)
附录 C μ 律到线性变换表	(233)
附录 D 缩写词的英文对照	(234)

第一章 概 述

1.1 引 言

数字信号处理(Digital Signal Processing,简称 DSP)是一门涉及许多学科且广泛应用于许多领域的新兴学科。本世纪 60 年代以来,随着计算机和信息学科的飞速发展,数字信号处理技术应运而生并得到迅速的发展。在过去的十几年里,数字信号处理已经成为通信等领域里的一项极为重要的技术。

数字信号处理是利用计算机或专用处理设备,以数字形式对信号进行采集、变换、滤波、估值、增强、压缩、识别等处理,以得到符合人们需要的信号形式。

数字信号处理是围绕着数字信号处理的理论、实现和应用等几个方面发展起来的。数字信号处理在理论上的发展推动了数字信号处理应用的发展,反过来,数字信号处理的应用又促进了数字信号处理理论的提高,而数字信号处理的实现则是理论和应用之间的桥梁。

数字信号处理是以众多的学科为理论基础的,它所涉及的范围极其广泛。例如,在数学领域,微积分、概率统计、随机过程、数值分析等都是数字信号处理的基本工具,同时,与网络理论、信号与系统、控制论、通信理论、故障诊断等也密切相关。近来新兴的一些学科,如人工智能、模式识别、神经网络等,都是与数字信号处理密不可分的。可以说,数字信号处理是把许多经典的理论体系作为自己的理论基础,同时又使自己成为一系列新兴学科的理论基础。

数字信号处理的实现方法一般有以下几种:(1)在通用的计算机(如 PC 机)上用软件(如 Fortran、C 语言)实现。(2)在通用计算机系统中加上专用的加速处理机。(3)用通用的单片机(如 MCS-51 系列等)实现,这种方法可用于一些不太复杂的数字信号处理,如数字控制等。(4)用通用的可编程 DSP 芯片实现。与单片机相比,DSP 芯片具有更加适合于数字信号处理的软件和硬件资源,可用于复杂的数字信号处理算法。(5)用专用的 DSP 芯片实现。在一些特殊的场合,要求的信号处理速度极高,用通用 DSP 芯片很难实现,例如专用于 FFT、数字滤波、卷积、相关等算法的 DSP 芯片,这种芯片将相应的信号处理算法在芯片内部用硬件实现,无需进行编程。

在上述几种方法中,第一种方法的缺点是速度较慢,一般可用于 DSP 算法的模拟。第二种和第五种方法专用性强,应用受到很大的限制。第二种方法不适合于复杂的 DSP 算法。只有第三种方法才使数字信号处理的应用打开了新的局面。

虽然数字信号处理的理论发展迅速,但在 80 年代以前,由于实现方法的限制,数字信号处理的理论还得不到广泛的应用。直到 70 年代末 80 年代初世界上第一个单片可编程 DSP 芯片的诞生,才将理论研究结果广泛应用到低成本的实际系统中,并且推动了新的理论和应用领域的发展。可以毫不夸张地说,DSP 芯片的诞生及发展对近十几年来通信、计算机、控制等领域的发展起到十分重要的作用。

1.2 DSP 系统

1.2.1 DSP 系统构成

图 1.1 示出了一个典型的 DSP 系统。图中的输入信号可以有各种各样的形式,例如,它可以是麦克风输出的语音信号或是电话线来的已调数据信号,也可以是编码后在数字链路上传输或存储在计算机里的摄像机图象信号等。

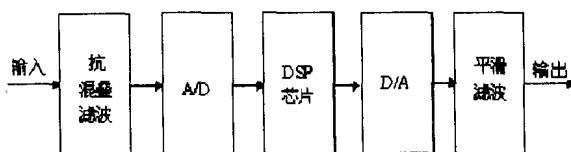


图 1.1 典型的 DSP 系统

输入信号首先进行带限滤波和抽样,然后进行模数(A/D)变换将信号变换成数字比特流。根据奈奎斯特抽样定理,为保持信息的不丢失,抽样频率至少必须是输入带限信号最高频率的 2 倍。

DSP 芯片的输入是 A/D 变换后得到的以抽样形式表示的数字信号,DSP 芯片对输入的数字信号进行某种形式的处理,如进行一系列的乘累加操作(MAC)。数字处理是 DSP 的关键,这与其它系统,如电话交换系统有很大的不同,在交换系统中,处理器的作用是进行路由选择,它并不对输入数据进行修改。因此虽然两者都是实时系统,但两者的实时约束条件却有很大的不同。最后,经过处理后的数字样值再经 D/A 变换转换为模拟样值,之后再进行内插和平滑滤波就可得到连续的模拟波形。

必须指出的是,上面给出的 DSP 系统模型是一个典型模型,但并不是所有的 DSP 系统都必须具有模型中的所有部件。如语音识别系统在输出端并不是连续的波形而是识别结果,如数字、文字等。有些输入信号本身就是数字信号(如 CD),因此就不必进行模数变换了。

1.2.2 DSP 系统的特点

数字信号处理系统是以数字信号处理为基础,因此具有数字处理的全部优点:

(1)接口方便。DSP 系统与其它以现代数字技术为基础的系统或设备都是相互兼容的,与这样的系统接口以实现某种功能要比模拟系统与这些系统接口要容易的多。

(2)编程方便。DSP 系统中的可编程 DSP 芯片可使设计人员在开发过程中灵活方便地对软件进行修改和升级。

(2)稳定性好。DSP 系统以数字处理为基础,受环境温度以及噪声的影响较小,可靠性高。

(3)精度高。16 位数字系统可以达到的精度。

(4)可重复性好。模拟系统的性能受元器件参数性能变化比较大,而数字系统基本不受影响,因此数字系统便于测试、调试和大规模生产。

(5)集成方便。DSP 系统中的数字部件有高度的规范性,便于大规模集成。

当然,数字信号处理也存在一定的缺点。例如,对于简单的信号处理任务,如与模拟交换线的电话接口,若采用 DSP 则使成本增加。DSP 系统中的高速时钟可能带来高频干扰和电磁

泄漏等问题,而且 DSP 系统消耗的功率也较大。此外,DSP 技术更新的速度快,数学知识要求多,开发和调试工具还不尽完善。

虽然 DSP 系统存在着一些缺点,但其突出的优点已经使之在通信、语音、图象、雷达、生物医学、工业控制、仪器仪表等许多领域得到越来越多的应用。

1.2.3 DSP 系统的设计过程

总的说来,DSP 系统的设计还没有非常好的正规设计方法。图 1.1 是 DSP 系统设计的一般过程。

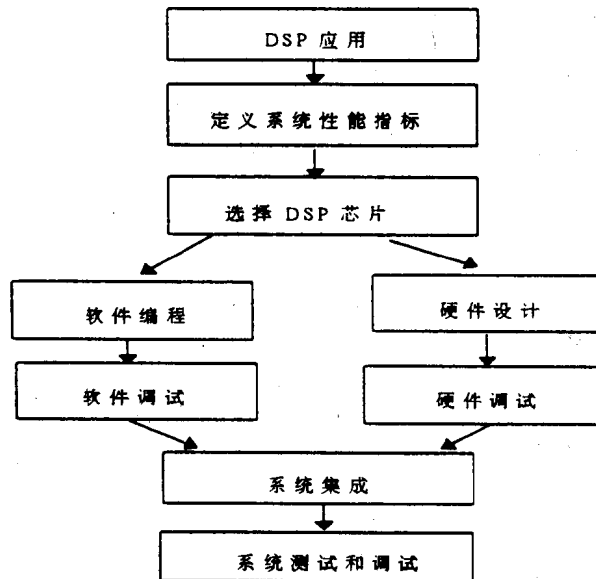


图 1.2 DSP 系统的设计流程

在设计 DSP 系统之前,首先必须根据应用系统的目标确定系统的性能指标、信号处理的要求,通常可用数据流程图、数学运算序列、正式的符号或自然语言来描述。

第二步是根据系统的要求进行高级语言的模拟。一般来说,为了实现系统的最终目标,需要对输入的信号进行适当的处理,而处理方法的不同会导致不同的系统性能,要得到最佳的系统性能必须在这一步确定最佳的处理方法,即数字信号处理的算法(Algorithm),因此这一步也称算法模拟阶段。例如,语音压缩编码算法就是要在确定的压缩比条件下,获得最佳的合成语音。算法模拟所用的输入数据是实际信号经采集而获得的,通常以计算机文件的形式存储为数据文件。如语音压缩编码算法模拟时所用的语音信号就是实际采集而获得并存储为计算机文件形式的语音数据文件。有些算法模拟时所用的输入数据并不一定要是实际采集的信号数据,只要能够验证算法的可行性,输入假设的数据也是可以的。

在完成第二步之后,接下来就可以设计实时 DSP 系统,实时 DSP 系统的设计包括硬件设计和软件设计两个方面。硬件设计首先要根据系统运算量的大小、对运算精度的要求、系统成本限制以及体积功耗等要求选择一合适的 DSP 芯片。然后设计 DSP 芯片的外围电路及其它电路。软件设计和编程主要根据系统要求和所选的 DSP 芯片编写相应的 DSP 汇编程序,若系统运算量不大且有高级语言编译器支持,也可用高级语言(如 C 语言)编程。当然,由于现有的高级语言编译器的效率还比不上手工编写汇编语言的效率,因此在实际应用系统中常常采用高级语言和汇编语言的混合编程方法,即在算法运算量大的地方,用手工编写的方法编写汇

编语言,而运算量不大的地方则采用高级语言。采用这种方法,既可缩短软件开发的周期,提高程序的可读性和可移植性,又能满足系统实时运算的要求。

DSP 硬件和软件设计完成后,就需要进行硬件和软件的调试。软件的调试一般借助于 DSP 开发工具,如软件模拟器、DSP 开发系统或仿真器等。调试 DSP 算法时一般采用对实时结果与模拟结果进行的方法,如果实时程序和模拟程序的输入相同,则两者的输出应该一致。应用系统的其它软件可以根据实际情况进行调试。硬件调试一般采用硬件仿真器进行调试,如果没有相应的硬件仿真器,且硬件系统不是十分复杂,也可以借助于一般的工具进行调试。

系统的软件和硬件分别调试完成后,就可以将软件脱离开发系统而直接应用系统上运行。当然,DSP 系统的开发,特别是软件开发是一个需要反复进行的过程,虽然通过算法模拟基本上可以知道实时系统的性能,但实际上模拟环境不可能做到与实时系统环境完全一致,而且将模拟算法移植到实时系统时必须考虑算法是否能够实时运行的问题。如果算法运算量太大不能在硬件上实时运行,则必须重新修改或简化算法。

1.3 可编程 DSP 芯片

1.3.1 什么是 DSP 芯片

DSP 芯片,也称数字信号处理器,是一种特别适合于进行数字信号处理运算的微处理器,其主要应用是实时快速实现各种数字信号处理算法。根据数字信号处理的要求,DSP 芯片一般具有如下一些主要特点:

- (1)在一个指令周期内可完成一次乘法和一次加法。
- (2)程序和数据空间分开,可以同时访问指令和数据。
- (3)片内具有快速 RAM,通常可通过独立的数据总线在两块中同时访问。
- (4)具有低开销或无开销循环及跳转的硬件支持。
- (5)快速的中断处理和硬件 I/O 支持。
- (6)具有在单周期内操作的多个硬件地址产生器。
- (7)可以并行执行多个操作。
- (8)支持流水线操作,使取指、译码和执行等操作可以重叠执行。

当然,与通用微处理器相比,DSP 芯片的其它通用功能相对较弱些。

1.3.2 DSP 芯片的发展

世界上第一个单片 DSP 芯片是 1978 年 AMI 公司宣布的 S2811,1979 年美国 Intel 公司宣布的商用可编程器件 2920 是 DSP 芯片的一个主要里程碑。这两种芯片内部都没有现代 DSP 芯片所必须有的单周期乘法器。1980 年,日本 NEC 公司推出的 μ PD7720 是第一个具有乘法器的商用 DSP 芯片。

在这之后,最成功的 DSP 芯片当数美国德克萨斯仪器公司(Texas Instruments,简称 TI)的一系列产品,TI 公司在 1982 年成功推出其第一代 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 TMS320C30/C31/C32,第四代 DSP 芯片 TMS320C40/C44,第五代 DSP 芯片 TMS320C50/C51/C52/C53 以及集多个 DSP 芯片于一体的高性能 DSP 芯片 TMS320C80/

C82 等。如今, TI 的一系列 DSP 产品已经成为当今世界上最有影响的 DSP 芯片。TI 公司 已经成为世界上最大的 DSP 芯片供应商, 其 DSP 市场份额占 50% 左右。

第一个采用 CMOS 工艺生产浮点 DSP 芯片的是日本的 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, 从而使处理的吞吐量发生了一个大的飞跃。而第一个高性能的浮点 DSP 芯片应是 AT&T 公司于 1984 年推出的 DSP32。

与其它公司相比, Motorola 公司在推出 DSP 芯片方面相对较晚。1986 年, 它推出了定点处理器 MC56001。1990 年, 它推出了与 IEEE 浮点格式兼容的浮点 DSP 芯片 MC96002。

美国模拟器件公司 (Analog Devices, 简称 AD) 在 DSP 芯片市场上也占有一定的份额, 相继推出了一系列具有自己特点的 DSP 芯片, 其定点 DSP 芯片有 ADSP2101/2103/2105、ASDP2111/2115、ADSP2161/2162/2164 以及 ADSP2171/2181, 浮点 DSP 芯片有 ADSP21000/21020、ADSP21060/21062 等。

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)降低到 40ns 以下(如 TMS320C40), 处理能力提高了 10 多倍。DSP 芯片内部关键的乘法器部件从 1980 年的占模片区(die area)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 4 μ m 的 N 沟道 MOS(NMOS)工艺, 而现在则普遍采用亚微米(Micron) 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, 如外部存储器的扩展和处理器间的通信等。此外, DSP 芯片的发展, 使 DSP 系统的成本、体积、重量和功耗都有很大程度的下降。表 1.1 是 TI 公司 DSP 芯片 1992 年与 1982 年的比较表。表 1.2 则是世界上主要 DSP 芯片供应商的代表芯片的一些数据。

表 1.1 TI DSP 10 年发展比较表(典型值)

年份	1982 年	1992 年
制造工艺	4 μ m NMOS	0.8 μ m CMOS
MIPS	5MIPS	40MIPS
MHZ	20MHz	80MHz
内部 RAM	144 字	1K 字
内部 ROM	1.5K	4K 字
价格	\$ 150.00	\$ 15.00
功耗	250mW/MIPS	12.5mW/MIPS
集成晶体管数	50K	500K

表 1.2 单片可编程 DSP 芯片

公司	DSP 芯片	推出时间	MAC 周期(ns)	定点位数	浮点位数
AMI	S2811	1978	300	12/16	
NEC	μ PD7720	1980	250	16/32	32
	μ PD77230	1985	150		

续表

公司	DSP 芯片	推出时间	MAC 周期(ns)	定点位数	浮点位数
TI	TMS32010	1982	390	16/32	32/40 40
	TMS32020	1987	200	16/32	
	TMS320C25	1989	100	16/32	
	TMS320C30	1989	60	24/32	
	TMS320C40	1992	40	32	
	TMS320C50	1990	35	16/32	
Motorola	MC56001	1986	75	24	32/44
	MC96002	1990	50	32/64	
	MC56002	1991	50	24/48	
AT&T	DSP32C	1988	80	16 或 24	32/40
	DSP16A	1988	25	16/36	32/40
	DSP3210	1992	60	24	
AD	ADSP2101	1990	60	16	32/40
	ADSP21020	1991	40	32	

1.3.3 DSP 芯片的分类

DSP 芯片可以按照下列三种方式进行分类。

1. 按基础特性分

这是根据 DSP 芯片的工作时钟和指令类型来分类的。如果在某时钟频率范围内的任何时钟频率上, DSP 芯片都能正常工作, 除计算速度有变化外, 没有性能的下降, 这类 DSP 芯片一般称之为静态 DSP 芯片。例如, 日本 OKI 电气公司的 DSP 芯片属于这一类。这种 DSP 芯片在存取速度较慢的存储器时不必再插入等待状态。

如果有两种或两种以上的 DSP 芯片, 它们的指令集和相应的机器代码机管脚结构相互兼容, 则这类 DSP 芯片称为一致性 DSP 芯片。例如, 美国 TI 公司的 TMS320C1X 就属于这一类。

2. 按数据格式分

这是根据 DSP 芯片工作的数据格式来分类的。数据以定点格式工作的 DSP 芯片称为定点 DSP 芯片, 如 TI 公司的 TMS320C1X/C2X/C5X 系列、AD 公司的 ADSP21XX 系列、AT&T 公司的 DSP16/16A、Motorola 公司的 MC56000 等。以浮点格式工作的称为浮点 DSP 芯片, 如 TI 公司的 TMS320C3X/C4X/C8X、AD 公司的 ADSP21XXX 系列、AT&T 公司的 DSP32/32C、Motorola 公司的 MC96002 等。

不同浮点 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, 如 TMS320C3X; 而有的 DSP 芯片则采用 IEEE 的标准浮点格式, 如 Motorola 公司的 MC96002、FUJITSU 公司的 MB86232 和 ZORAN 公司的 ZR35325 等。

3. 按用途分

按照 DSP 的用途来分, 可分为通用型 DSP 芯片和专用型 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 芯片属于通用型 DSP 芯片。专用 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, 如 Motorola 公司的 DSP56200、Zoran 公司的 ZR34881、Inmos 公司的 IMSA100 等就属于专用型的 DSP 芯片。

本书主要讨论通用型的 DSP 芯片。

1.3.4 DSP 芯片的选择

设计 DSP 应用系统,选择 DSP 芯片是非常重要的一个环节。只有选定了 DSP 芯片才能进一步设计其外围电路及系统的其它电路。总的来说,DSP 芯片的选择应根据实际的应用系统需要而确定。不同的 DSP 应用系统由于应用的场合、应用目的等不尽相同,对 DSP 芯片的选择也是不同的。一般来说,选择 DSP 芯片时应考虑到如下诸多因素。

1. 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 芯片的运算速度可以用以下几种性能指标来衡量:

(1)指令周期:就是执行一条指令所需的时间,通常以 ns 为单位。如 TMS320C25 在主频为 40MHz 时的指令周期为 100ns。

(2)MAC 时间:即一次乘法加上一次加法的时间。大部分 DSP 芯片可在一个指令周期内完成一次乘法和加法操作,如 TMS320C25 的 MAC 时间就是 100ns。

(3)FFT 执行时间:即运行一个 N 点 FFT 程序所需的时间。由于 FFT 运算涉及的运算在数字信号处理中很有代表性,因此 FFT 运算时间常作为衡量 DSP 芯片运算能力的一个指标。

(4)MIPS:即每秒执行百万条指令。如 TMS320C25 在时钟为 40MHz 时的处理能力为 10 MIPS,即每秒执行一千万条指令。

(5)MOPS:即每秒执行百万次操作。如 TMS320C40 的运算能力为 275 MOPS。

(6)MFLOPS:即每秒执行百万次浮点操作。如 TMS320C30 在主频为 33.33MHz 时的处理能力为 33.33 MFLOPS。

(7)BOPS:即每秒执行十亿次操作。如 TMS320C80 的处理能力为 2 BOPS。

2. DSP 芯片的价格。DSP 芯片的价格也是选择 DSP 芯片所需考虑的一个重要因素。如果采用价格昂贵的 DSP 芯片,即使性能再高,其应用范围肯定会受到一定的限制。因此根据实际系统的应用情况,需确定一个价格适中的 DSP 芯片。当然,由于 DSP 芯片发展迅速,DSP 芯片的价格往往下降较快,因此在开发阶段选用某种价格稍贵的 DSP 芯片,等到系统开发完毕,其价格可能已经下降一倍甚至更多。

3. DSP 芯片的硬件资源。不同的 DSP 芯片所提供的硬件资源是不相同的,如片内 RAM、ROM 的数量,外部可扩展的程序和数据空间,总线接口、I/O 接口等。即使是同一系列的 DSP 芯片,如 TI 的 TMS320C5X 系列,不同的 DSP 芯片也可以适应不同的需要。

4. DSP 芯片的运算精度。一般的定点 DSP 芯片的字长为 16 位,如 TMS320 系列。但有的公司的定点芯片为 24 位,如 Motorola 公司的 MC56001 等。浮点芯片的字长一般为 32 位,累加器为 40 位。

5. DSP 芯片的开发工具。在 DSP 系统的开发过程中,开发工具是必不可少的。如果没有开发工具的支持,要想开发 DSP 系统几乎是不可能的。如果有功能强大的开发工具的支持,如 C 语言支持,则开发的时间就会大大缩短。所以,在选择 DSP 芯片的同时必须注意其开发工具的支持情况,包括软件和硬件的开发工具。

6. DSP 芯片的功耗。在某些 DSP 应用场合,功耗也是一个需要特别注意的问题。如便携式的 DSP 设备、手持设备、野外应用的 DSP 设备等对功耗有特殊的要求。

7. 其它。除了上述因素外,选择 DSP 芯片还应考虑到封装的形式、质量标准、供货情况、生命周期等。有的 DSP 芯片可能有 DIP、PGA、PLCC、PQFP 等多种封装形式。有些 DSP 系统可能最终要求的是工业级或军用级标准,在选择时就需要注意到所选的芯片是否有工业级或军