



普通高等教育“十一五”国家级规划教材

电子信息类精品教材

DSP芯片的原理 与开发应用

(第5版)

• 张雄伟 杨吉斌 吴其前 曹铁勇 贾冲 邹霞 等编著



中国工信出版集团



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

普通高等教育“十一五”
电子信息类精品教材

DSP 芯片的原理与开发应用

(第 5 版)

张雄伟 杨吉斌 吴其前
曹铁勇 贾冲 邹霞 李莉

编著

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

可编程 DSP 芯片是一种应用非常广泛的微处理器。本书按照“基础知识—开发环境—软件开发—硬件开发—综合实例”的顺序,由浅入深、全面系统地介绍了 DSP 芯片的基本原理、开发过程和应用方法。首先,介绍了 DSP 系统的设计、DSP 芯片的特点、DSP 芯片的软硬件开发环境以及定点和浮点 DSP 的运算基础;其次,在介绍 DSP 芯片存储资源管理的基础上,重点介绍了基于 C 语言和汇编语言的软件开发方法;接着,介绍了 DSP 系统的硬件设计和开发方法;最后,通过实例介绍了 DSP 综合系统的开发过程和开发方法。

本书的目的是使读者了解 DSP 芯片的基本原理和常用 DSP 芯片的应用,熟悉 DSP 芯片开发工具及使用方法,掌握 DSP 系统的软硬件设计和应用系统开发方法,具备从事 DSP 芯片软硬件设计和 DSP 系统开发的能力。为方便教学,本书提供 PPT 课件。

本书结构清晰、内容全面、举例丰富、实用性强,可作为电子信息类研究生和相关专业高年级大学生的教材,也可作为 DSP 芯片应用人员的培训教材,对于从事 DSP 芯片应用开发的科技人员和高校教师也具较高的参考价值。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

DSP 芯片的原理与开发应用 / 张雄伟等编著. —5 版. —北京:电子工业出版社,2016.8

电子信息类精品教材

ISBN 978-7-121-29493-8

I. ①D… II. ①张… III. ①数字信号处理—高等学校—教材②数字信号—微处理器—高等学校—教材 IV. ①TN911.72②TP332

中国版本图书馆 CIP 数据核字(2016)第 174816 号

责任编辑:竺南直

印 刷:三河市良远印务有限公司

装 订:三河市良远印务有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:24 字数:645 千字

版 次:1997 年 8 月第 1 版

2016 年 8 月第 5 版

印 次:2016 年 8 月第 1 次印刷

定 价:49.80 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888,88258888。

质量投诉请发邮件至 zltz@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: davidzhu@phei.com.cn。

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, 特别适合于进行快速数字信号处理运算。DSP 芯片具有专门的快速硬件乘法器, 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 20 世纪 80 年代初 DSP 芯片诞生以来, DSP 芯片在 30 多年时间里得到了飞速的发展, DSP 芯片的性能不断提高, 价格显著降低, 开发手段越来越完善。DSP 芯片已经在通信与信息系统、信号与信息处理、自动控制、航空航天、雷达、军事、医疗、家用电器等众多领域得到越来越广泛的应用。

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 算法集成到 DSP 芯片内部, 一般适用于某些专用的场合。

本书主要讨论通用型的可编程 DSP 芯片。

目前, 国内广泛应用的 DSP 芯片主要来自于美国的德州仪器公司 (TI) 和模拟器件公司 (ADI)。其中, TI 公司的 DSP 芯片占 DSP 芯片市场近 50%, 因此, 本书主要以 TI 公司的 DSP 芯片为例进行介绍。

全书共 12 章, 可分为五个部分。

第一部分是基础知识, 包括第 1、2 章。第 1 章概述了 DSP 系统的特点、设计和开发方法以及 DSP 芯片的特点、发展、分类、选择和应用, 简要介绍了 TI 公司和 ADI 公司的常用 DSP 芯片; 第 2 章介绍 DSP 芯片的基本结构、主要特征以及存储单元、集成外设和中断等。

第二部分是开发环境, 对应第 3 章。主要介绍 DSP 芯片的软硬件开发环境, 比较详细地介绍了 TI 公司的集成开发环境——CCS 的基本原理和使用方法。

第三部分是软件开发, 包括第 4、5、6、7、8 章。第 4 章介绍 DSP 的数值运算基础, 包括定点 DSP 中的定标、定点和浮点 DSP 的有关问题; 第 5 章介绍 DSP 芯片的存储资源管理, 重点介绍了广泛采用的公共目标文件格式 (COFF) 和编程方法; 第 6 章和第 7 章分别介绍基于 C 和汇编语言的 DSP 软件开发; 第 8 章通过实例介绍了 DSP 算法软件开发的一般过程。

第四部分是硬件开发, 包括第 9、10、11 章。第 9 章介绍了 DSP 系统的硬件设计过程和设计方法; 第 10 章介绍了基于 CSL 的外设程序和典型控制程序的开发方法; 第 11 章介绍了 DSP 脱机系统的设计和开发实例。

第五部分是综合实例, 对应第 12 章。通过实例系统地介绍了 DSP 系统的综合开发过程和开发方法。

为方便教学, 本书配套有 PPT 课件和部分实例代码, 可登录华信教育资源网 (<http://www.hxedu.com.cn>) 免费注册索取。

本书由张雄伟主编。张雄伟编著了第 1、2、4、5 章及附录 A、B、G、H; 杨吉斌编著了第 10、11、12 章及附录 D、E; 吴其前编著了第 6、7 章及附录 C; 曹铁勇编著了第 3 章及附录 F; 贾冲编著了第 9 章; 邹霞编著了第 8 章; 吴海佳绘制了部分插图, 李莉参与了部分章节的编写。全书由张雄伟、吴其前、杨吉斌进行校对并统稿。

本书第 1 版 (1997 年)、第 2 版 (2000 年)、第 3 版 (2003 年)、第 4 版 (2009 年) 深得

广大读者的厚爱，受到了广泛的欢迎，广大 DSP 开发人员将本书作为 DSP 应用开发的参考书，很多高校将本书作为 DSP 相关课程的教材。2001 年，本书第 2 版获得全国优秀畅销书奖；2006 年，本书被列入普通高等教育“十一五”国家级规划教材；2007 年，本书作者被评为电子工业出版社优秀作者。对于广大读者对本书的热情支持，作者表示深深的谢意。

在重新出版本书时，根据 DSP 芯片的发展以及教学和培训需要，我们对本书的结构和内容都做了较大程度的增删和修改，希望读者在使用本书的过程中提出宝贵的意见和建议，以便在今后的修订中参考。

由于 DSP 芯片的技术发展十分迅速，加上作者水平所限，书中错误之处在所难免，恳请广大读者给予批评指正。

本书是解放军理工大学智能语音处理课题组长期从事“DSP 芯片原理与应用”教学和科研工作的总结。本书的出版得到了解放军理工大学的专项资助。

编著者

2016 年 5 月于解放军理工大学，南京

目 录

第 1 章 概述	1
1.1 引言	1
1.2 DSP 系统	2
1.2.1 DSP 系统的基本构成	2
1.2.2 DSP 系统的特点	2
1.2.3 DSP 系统的设计与开发	3
1.2.4 DSP 系统的开发工具	4
1.2.5 实时 DSP 系统	5
1.3 DSP 芯片概述	6
1.3.1 DSP 芯片的定义	6
1.3.2 DSP 芯片的特点	6
1.3.3 DSP 芯片的发展	7
1.3.4 DSP 芯片的分类	8
1.3.5 DSP 芯片的选择	9
1.3.6 DSP 系统的运算量	12
1.3.7 DSP 芯片的应用	13
1.4 常用的 DSP 芯片	14
1.4.1 TI 公司 DSP 芯片	14
1.4.2 ADI 公司 DSP 芯片	19
1.5 内容组织与常用术语	20
1.5.1 内容组织结构	20
1.5.2 常用术语	21
本章小结	21
习题与思考题	22
第 2 章 DSP 芯片的基本结构和特征	23
2.1 引言	23
2.2 DSP 芯片的基本结构	23
2.2.1 概述	23
2.2.2 总线结构	23
2.2.3 流水线	25
2.3 中央处理单元 (CPU)	25
2.3.1 算术逻辑单元 (ALU)	26
2.3.2 累加器	27
2.3.3 桶形移位寄存器	27
2.3.4 乘累加单元	28
2.3.5 寻址单元	29

2.4	存储单元	30
2.4.1	片内存储器	30
2.4.2	存储器映射寄存器	31
2.4.3	Cache	31
2.4.4	外部扩展存储器	32
2.5	集成外设与接口	32
2.5.1	通用类外设	33
2.5.2	通信类外设	34
2.5.3	存储类接口	35
2.5.4	DMA 控制器	35
2.5.5	专用外设与接口	36
2.6	中断	36
2.6.1	中断源	37
2.6.2	中断优先级	37
2.6.3	中断处理过程	39
2.6.4	中断矢量表	39
	本章小结	40
	习题与思考题	40
第 3 章	DSP 芯片的开发环境	41
3.1	引言	41
3.2	软件开发流程	41
3.3	软件开发环境	43
3.3.1	基本概念	44
3.3.2	CCS 开发环境	46
3.3.3	软件开发功能	51
3.3.4	DSP/BIOS	62
3.4	硬件开发流程	67
3.5	硬件开发环境	67
3.5.1	DSP 仿真器	68
3.5.2	仿真器接口	68
3.5.3	TI 公司仿真器	70
	本章小结	70
	习题与思考题	71
第 4 章	DSP 的数值运算基础	72
4.1	引言	72
4.2	定点的基本概念	72
4.2.1	数的定标	72
4.2.2	数的转换	73
4.2.3	溢出保护	75
4.2.4	符号扩展	76

4.2.5	舍入与截尾	76
4.3	定点运算实现的基本原理	77
4.3.1	加法/减法运算的 C 语言定点模拟	77
4.3.2	乘法运算的 C 语言定点模拟	79
4.3.3	除法运算的 C 语言定点模拟	80
4.3.4	程序变量的 Q 值确定	80
4.3.5	浮点至定点变换的 C 程序举例	81
4.4	DSP 定点算术运算实现的基本原理	83
4.4.1	定点乘法	83
4.4.2	定点加法	84
4.4.3	定点除法	85
4.5	非线性运算定点实现方法	87
4.5.1	级数展开法	87
4.5.2	查表法	87
4.5.3	混合法	89
4.6	浮点数的表示格式	91
4.6.1	IEEE 浮点数格式	91
4.6.2	TMS320C3x 浮点数格式	92
4.7	基本的浮点运算	94
4.7.1	浮点乘法和加减法	94
4.7.2	浮点除法	94
4.8	非线性浮点运算的快速实现	95
	本章小结	97
	习题与思考题	97
第 5 章	DSP 芯片的存储资源管理	98
5.1	引言	98
5.2	TMS320C54x 的存储区组织	98
5.2.1	程序空间	98
5.2.2	数据空间	99
5.2.3	I/O 空间	99
5.2.4	存储器映射寄存器	100
5.2.5	TMS320VC5416 的存储资源	100
5.3	TMS320C55x 的存储区组织	101
5.3.1	存储空间组织	102
5.3.2	TMS320VC5509A 的存储资源	102
5.3.3	程序空间	103
5.3.4	数据空间	104
5.3.5	I/O 空间	105
5.4	程序结构与 COFF 目标文件格式	105
5.4.1	块 (section)	106

5.4.2	汇编器对块的处理	106
5.4.3	链接器对块的处理	108
5.4.4	程序重定位	108
5.4.5	COFF 文件中的符号	109
5.4.6	COFF 文件格式编程示例	109
5.5	存储区分配与 CMD 文件	111
5.5.1	文件链接方法	111
5.5.2	链接命令文件	112
5.5.3	TMS320VC5509A 的 CMD 文件	112
	本章小结	115
	习题与思考题	115
第 6 章	基于 C 语言的 DSP 芯片开发	116
6.1	引言	116
6.2	TMS320C55x 的 C 语言	116
6.2.1	变量和常数	118
6.2.2	函数	121
6.2.3	预处理	123
6.2.4	asm 语句	125
6.3	C 语言程序代码的优化	125
6.3.1	C 语言程序代码编译分析	125
6.3.2	C 语言程序的优化方法	131
	本章小结	141
	习题与思考题	141
第 7 章	基于 TMS320C55x 的汇编语言开发	142
7.1	汇编语言源程序格式	142
7.2	汇编源程序中常见符号和伪指令	144
7.3	汇编指令系统	147
7.4	寻址方式	150
7.4.1	绝对寻址方式	150
7.4.2	直接寻址方式	151
7.4.3	间接寻址方式	153
7.5	汇编代码的优化	162
7.6	汇编语言和 C 的混合编程方法	162
7.6.1	独立的 C 和汇编模块接口	163
7.6.2	从 C 程序中访问汇编程序变量	164
7.6.3	在汇编程序中访问 C 程序变量	165
7.6.4	在 C 程序中直接嵌入汇编语句	166
7.6.5	TMS320C55x 混合编程举例	166
	本章小结	167
	习题与思考题	167

第 8 章	DSP 算法软件开发实例	168
8.1	引言	168
8.2	基于 FFT 的单音检测算法原理	168
8.2.1	单音检测算法概述	168
8.2.2	DFT 的基本原理	169
8.2.3	FFT 算法的导出	169
8.3	基于 MATLAB 的 DSP 算法仿真	174
8.3.1	MATLAB 简介	174
8.3.2	单音检测算法的 MATLAB 仿真	175
8.4	单音检测算法的浮点 C 语言实现	177
8.5	单音检测算法的定点 C 语言实现	181
8.5.1	FFT 运算溢出及避免方法	182
8.5.2	单音检测算法的定点 C 语言实现	183
8.6	单音检测算法的定点 DSP 芯片实现	188
	本章小结	190
	习题与思考题	190
第 9 章	DSP 系统的硬件设计	191
9.1	引言	191
9.2	DSP 系统硬件设计的基本步骤	191
9.3	最小 DSP 系统的硬件设计	192
9.3.1	电源电路	192
9.3.2	复位电路	195
9.3.3	时钟电路	196
9.3.4	JTAG 电路	197
9.3.5	引脚的电平转换与处理	198
9.3.6	硬件系统的可靠性设计	200
9.4	外部存储器接口设计	203
9.4.1	TMS320C54x 存储器接口设计	204
9.4.2	TMS320C55x 存储器接口设计	208
9.5	模数接口电路的设计	209
9.5.1	TLV320AIC23B 接口的设计	209
9.5.2	MC145483 接口的设计	212
9.6	通信接口的设计	213
9.6.1	通信接口	213
9.6.2	同步通信接口电路设计	214
9.6.3	异步通信接口电路设计	215
9.7	主从式系统的设计	215
9.7.1	共享存储器实现双机通信	216
9.7.2	利用 HPI 实现双机通信	217
	本章小结	218

习题与思考题	218
第 10 章 DSP 芯片外设驱动程序的开发	219
10.1 引言	219
10.2 基于 CSL 的外设程序设计	220
10.2.1 CSL 简介	220
10.2.2 CSL 的特点	222
10.2.3 CSL 程序开发基本步骤	224
10.3 中断控制程序的开发	233
10.3.1 中断初始化	234
10.3.2 中断服务程序的开发	237
10.3.3 中断控制程序实例	238
10.4 典型接口功能的控制程序开发	241
10.4.1 外部存储的控制程序	241
10.4.2 串行 A/D 和 D/A 功能的控制程序	244
10.4.3 外部主机通信功能的控制程序	249
10.5 DMA 的控制程序开发	250
10.5.1 DSP 芯片的 DMA 数据传输模式	251
10.5.2 DSP 芯片的 DMA 传输地址	252
10.5.3 DSP 芯片的 DMA 事件和中断	252
10.5.4 DMA 控制程序举例	254
本章小结	257
习题与思考题	257
第 11 章 DSP 脱机系统的开发	258
11.1 联机系统和脱机系统	258
11.2 DSP 芯片的 BOOT	260
11.2.1 BOOT 简介	260
11.2.2 BOOT 过程	261
11.2.3 BOOT 表	266
11.3 DSP 脱机系统实现	267
11.3.1 硬件设计	267
11.3.2 软件代码准备与烧录	268
11.4 DSP 脱机系统软件开发实例	271
11.4.1 并口 FLASH 脱机系统的软件开发	271
11.4.2 串口 FLASH 脱机系统的软件开发	273
本章小结	276
习题与思考题	277
第 12 章 DSP 系统的开发实例	278
12.1 引言	278
12.2 基于 TMS320VC5509A 的 DSP 应用系统实例	278

12.2.1	系统简介	278
12.2.2	系统架构	279
12.3	系统的硬件开发	280
12.3.1	器件选型	280
12.3.2	接口设计	282
12.3.3	电路设计	283
12.3.4	电路调试	286
12.4	系统的软件开发	287
12.4.1	算法仿真程序编写和测试	287
12.4.2	接口控制程序编写和调试	289
12.5	系统的软硬件集成	297
	本章小结	307
	习题与思考题	308
附录 A	缩略词的中英文对照	309
附录 B	TMS320VC5509A PGE LQFP 引脚图及定义	313
附录 C	TMS320C55x 指令集	315
附录 D	CSL 库函数	332
附录 E	代码实例	346
附录 F	CCS V5.5 的安装	365
附录 G	8 位 μ 律 PCM/16 位线性转换的 C 语言程序	369
附录 H	μ 律 PCM 到线性变换表	371
	参考文献	372

第1章 概 述

1.1 引 言

数字信号处理 (DSP) 是一个涉及多门学科且在众多领域得到广泛应用的学科专业领域。

20 世纪 60 年代以来, 随着电子信息技术的飞速发展, DSP 技术应运而生, 并得到迅速的发展。目前, DSP 技术已经在通信、自动控制、航空航天、军事、仪器仪表、智能终端、家用电器等众多领域得到越来越广泛的应用, DSP 已经来到了我们每个人的身边。

DSP 是指利用计算机、微处理器或专用处理设备, 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, 以得到符合应用需要的信号形式。

DSP 是围绕着 DSP 的理论、技术、算法、实现、应用等几个方面发展起来的。DSP 在理论和技术上的进步推动了 DSP 应用的发展; 反过来, 越来越广泛的 DSP 应用又促进了 DSP 理论和技术的发展; DSP 的算法及实现则是理论、技术和工程应用之间的桥梁。

DSP 以众多学科的理论为基础, 它所涉及的范围极其广泛。例如, 数学领域的微积分、概率统计、随机过程、数值分析等都是 DSP 的基本工具, 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。一些新兴的学科, 如机器学习、人工智能、模式识别、神经网络等, 都与 DSP 密不可分。因此, DSP 将许多经典的理论体系作为自身的理论基础, 同时又使自身成为一系列新兴学科的理论基础。

虽然 DSP 的理论发展迅速, 但在 20 世纪 80 年代以前, 由于工程实现方法的限制, DSP 的理论和技术的进步还得不到广泛的应用。直到 20 世纪 80 年代初世界上第一片可编程 DSP 芯片的诞生, 才将理论研究成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。

因此, 可以毫不夸张地说, DSP 芯片的诞生及发展对近 30 多年来通信、自动控制、计算机应用等领域的发展起到十分重要的作用。

DSP 的工程实现一般有以下几种方法:

- (1) 在通用的计算机 (如 PC) 上用软件 (如 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实现。
- (3) 基于通用的单片机 (如 STM32、MSP430 系列等) 实现。
- (4) 基于通用的可编程 DSP 芯片实现。与单片机相比, DSP 芯片具有更加适合于 DSP 的软件和硬件资源, 可用于复杂的 DSP 算法。
- (5) 基于通用的可编程逻辑器件 (如 FPGA 等) 实现。由于可编程逻辑器件具有较好的并行处理能力, 因此这种方法可以适应一些速度要求极高的场合。
- (6) 基于专用的 DSP 芯片实现。在一些特殊的应用场合, DSP 算法特殊, 且要求的信号处理速度很高, 采用通用 DSP 芯片难以实现, 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专用硬件或软件实现, 用户应用时无须再进行编程。

上述几种方法中：第 1 种方法的优点是实现比较方便，缺点是运行速度较慢，一般可用于 DSP 算法的模拟与仿真。第 2 种和第 6 种方法专用性强，优点是对特定应用处理能力强，缺点是应用受到比较大的限制，第 2 种方法也不便于 DSP 系统的独立运行。第 3 种方法的优点是成本和功耗较低，缺点是处理能力有限，适用于实现简单的 DSP 算法。第 5 种方法的优点是处理能力强，特别是其并行能力适合于多通道大数据吞吐量的信号处理，缺点是成本高、功耗大，对开发人员的编程能力要求较高。第 4 种方法具有通用性、高效性等特点，为 DSP 的工程应用奠定了实现的基础，也为 DSP 的推广应用提供了良好的工具和平台。

学习掌握 DSP 芯片的开发应用对于实现各种 DSP 的应用系统具有十分重要的实际意义。本书将介绍如何使用可编程 DSP 芯片来实现 DSP 应用系统。

1.2 DSP 系统

1.2.1 DSP 系统的基本构成

图 1-1 所示为一个典型的 DSP 系统。图中，输入信号可以有各种各样的形式，例如，可以是麦克风采集的语音信号、摄像机采集的视频信号等。

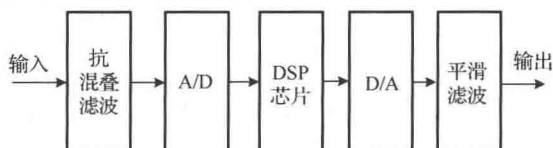


图 1-1 典型的 DSP 系统

一般地，输入信号首先进行带限滤波和采样，然后进行模/数（A/D: Analog to Digital）转换将模拟信号转换成数字比特流。根据奈奎斯特采样定理，对低通模拟信号，为保持信息的不丢失，采样频率至少必须是输入信号最高频率的 2 倍。

DSP 芯片的输入是 A/D 转换后得到的以样值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。数字处理是 DSP 的关键，这与其他系统（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。最后，经过处理后的数字样值再经数/模（Digital to Analog, D/A）转换将数字信号转换为模拟样值，并进行内插和平滑滤波后，就可得到连续的模拟波形。

必须指出的是，图 1-1 给出的只是一个典型的 DSP 系统，但并不是所有的 DSP 系统都必须包括图 1-1 中的所有部分。例如，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，因此就不必进行 A/D 转换了。

1.2.2 DSP 系统的特点

DSP 系统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 系统与其他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。

(3) 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, 系统的可靠性高。

(4) 数值精度高。16 位的数字系统可以达到 10^{-5} 的数值精度。

(5) 可重复性好。模拟系统的性能受元器件参数性能变化的影响比较大, 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。

(6) 集成方便。DSP 系统中的数字部件有高度的规范性, 便于大规模系统集成。

当然, DSP 也存在一定的缺点。DSP 基于数字处理, 首先需要对模拟信号进行采样、量化和编码, 这就必然会引入量化噪声, 如何保证数值的精度是系统设计必须考虑的问题; 对于简单的信号处理任务, 如与模拟交换线的控制接口, 若采用 DSP 则可能会使成本和系统功耗增加; 此外, DSP 系统中的高速时钟可能会带来高频干扰和电磁泄漏等问题。

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



1.2.3 DSP 系统的设计与开发

图 1-2 是 DSP 系统设计与开发的一般流程。

1. 定义系统性能指标

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。

2. 采用高级语言进行性能模拟

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恰当的处理, 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 DSP 的算法, 因此, 这一步也称算法模拟或仿真阶段。

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。

3. 设计 DSP 系统

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功能、运算量、运算精度、成本以及体积、功耗等诸多要求选择合适的 DSP 芯片; 然后, 围绕 DSP 芯片, 设计其外围电路及其他电路。

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 程序, 可用高级语言 (如 C 语言) 或芯片的汇编语言编程。一般地, 高级语言并不是完全针对 DSP 应用设

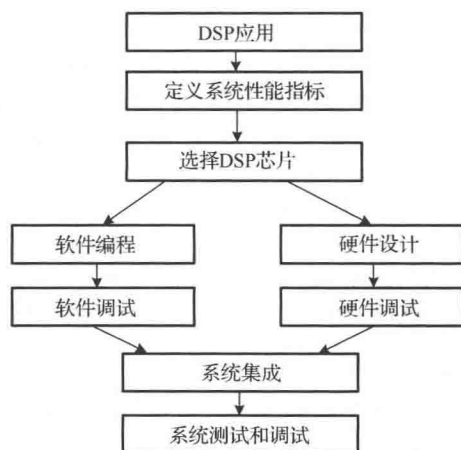


图 1-2 DSP 系统设计与开发的一般流程

计的，基于芯片结构设计的汇编语言实现的代码效率更高，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，采用汇编语言，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。

4. 借助开发工具进行软硬件调试

软件的调试需要借助于 DSP 开发工具，如软件模拟器、DSP 开发系统或仿真器等。调试 DSP 算法时一般采用比较 DSP 实现结果与模拟仿真结果的方法，如果 DSP 程序和模拟程序的输入相同，则两者的输出应该一致。

应用系统的其他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。

5. 系统集成与独立系统运行

DSP 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在设计系统上运行。当然，DSP 系统的开发，特别是软件开发是一个需要反复优化进行的过程，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芯片上实时运行，则必须重新修改或优化算法。

 1.2.4 DSP 系统的开发工具

根据图 1-2 的设计与开发流程，要开发一个完整的 DSP 系统，需要借助于诸多软硬件开发工具，表 1-1 列出了可能需要的开发工具。当然，有些工具也不一定是必备的，如逻辑分析仪；有些工具则是可选的，如算法模拟时可以用 C 语言，也可以用 MATLAB 语言，或者先进行 MATLAB 模拟，再进行 C 语言模拟，还可以用其他程序语言。在采用美国得克萨斯仪器公司（Texas Instruments, TI）的 DSP 芯片（以下简称 TI-DSP 芯片）进行系统开发时，一般需采用 CCS（Code Composer Studio）工具软件，这是一个集成开发环境，包括了编辑、编译、汇编、链接、软件模拟、调试等几乎所有需要的软件。此外，如果 DSP 系统中还有其他微处理器，当然还必须有相应的开发工具支持。

表 1-1 DSP 系统的开发工具

开发步骤	开发内容	开 发 工 具	
		硬 件	软 件
1	算法模拟	计算机	C 语言、MATLAB 语言等
2	DSP 软件编程	计算机	编辑器（如 UltraEdit、CCS 等）
3	DSP 软件调试	计算机、DSP 仿真器等	DSP 代码生成工具（包括 C 编译器、汇编器、链接器等） DSP 软件模拟器 Simulator、CCS 等
4	DSP 硬件设计	计算机	电路设计软件（如 Protel、Altium 等）、其他相关软件（如 EDA 软件等）
5	DSP 硬件调试	计算机、DSP 仿真器、示波器、信号发生器、逻辑分析仪等	相关支持软件
6	系统集成	计算机、DSP 仿真器、编程器、示波器、信号发生器、逻辑分析仪等	相关支持软件



1.2.5 实时 DSP 系统

实时性能是衡量 DSP 系统的一项重要指标。根据 DSP 实现方式,可以将 DSP 系统分为实时 DSP 系统和非实时 DSP 系统。在介绍实时和非实时 DSP 系统之前,先来看一看 DSP 算法的概念。

1. DSP 算法

通常,我们将一个 DSP 系统所承担的特定数字信号处理方法称为 DSP 算法。程序员根据 DSP 系统所实现的目标设计 DSP 算法,例如,需要实现一个语音识别系统,那么程序员首先必须研究并设计一个语音识别算法,然后将该算法转换成 DSP 代码,交给 DSP 系统来实现。

2. 实时与非实时 DSP 系统

DSP 系统实现 DSP 算法通常具有周期重复性的特点,也就是说,对于连续的输入信号,DSP 系统需要重复完成相同的 DSP 算法,只不过对于不同的输入信号,得到不同的输出结果。

图 1-3 给出了 DSP 系统的一般处理过程。

(1) 实时 DSP 系统

如果 DSP 系统在下一个任务到来之前完成当前的 DSP 算法处理,表明该系统能够在连续两个任务的时间间隔内完成 DSP 算法,该系统就是一个实时 DSP 系统。实时 DSP 系统的一个特点是处理任务通常是周期性定时到来的,并且这个周期时长不随 DSP 处理时间的长短而改变。

(2) 非实时 DSP 系统

如果 DSP 系统处理方式是完成当前处理任务(不论处理当前任务需要多少时间)之后,再去取下一个任务,这个系统就是一个非实时 DSP 系统。

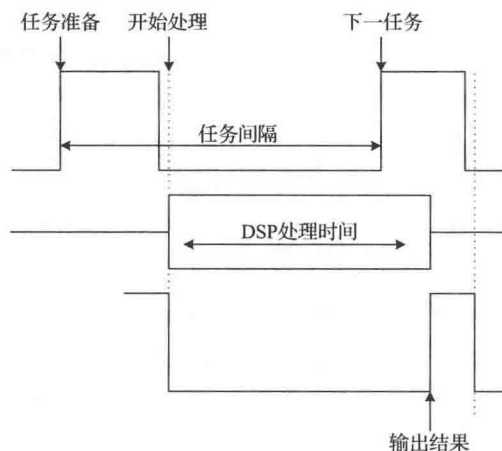


图 1-3 DSP 系统的一般处理过程

3. DSP 的两种处理方式

(1) 按样点处理

如果 DSP 算法对每一个输入样点循环一次,即 DSP 算法的对象是一个样点,这种处理方式就是按样点处理。例如,在数字滤波器中,通常需要对每一个输入样点计算一次。

(2) 按帧处理

有些数字信号处理算法不是每个输入样点循环一次,而是每隔一定的时间间隔(通常称为帧)循环一次,DSP 算法的对象是一帧信号而不是一个样点,这种处理方式就是按帧处理。例如,低速语音编码算法通常以 20ms 为一帧,每隔 20ms 语音编码算法循环一次,对于 8kHz 的采样,20ms 相当于 160 个样点。

图 1-4 所示是两种处理方式的示意图。