

DSP应用丛书

Blackfin系列 DSP原理与系统设计

陈 峰 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

DSP 应用丛书

Blackfin 系列 DSP 原理与系统设计

陈 峰 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

Blackfin DSP 是 ADI 与 Intel 联合开发的体现高性能体系结构的首款第四代定点 DSP 产品。在其系列产品中, 最高时钟频率为 600 MHz, 峰值处理能力为 1.2 GMIPS。本书共分 8 章, 主要介绍了 Blackfin 系列芯片的结构、特点、接口功能、软件编程等内容, 并以视频应用为实例介绍了一些使用方法, 同时给出了几个用于 MPEG-4 算法中的典型函数的优化实例。

本书可供从事信号处理的科研和工程技术人员阅读参考, 也可作为信息与信号处理、通信等相关专业的研究生、高年级本科生的参考教材。

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

Blackfin 系列 DSP 原理与系统设计/陈峰编著. —北京: 电子工业出版社, 2004.1

(DSP 应用丛书)

ISBN 7-5053-9286-7

I. B… II. 陈… III. 数字信号—信号处理—数字通信系统, Blackfin IV. TN911.72

中国版本图书馆 CIP 数据核字 (2003) 第 098422 号

责任编辑: 沈艳波

印 刷: 北京京科印刷有限公司

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 18 字数: 460 千字

印 次: 2004 年 1 月第 1 次印刷

印 数: 5 000 册 定价: 30.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077。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。

序

Blackfin 处理器是新一代嵌入式多媒体处理器。它在一个芯片和一个开发平台上融合了 DSP 信号处理、RISC 的控制处理和视频处理功能,具有高度的并行处理和强大的 DMA 能力,具有专用的视频指令和硬件支持,高效的程序效率,动态的功耗管理和方便的开发工具和使用方法,领先的内存管理,并支持包括 uLinux 在内的嵌入式操作系统。Blackfin 处理器及其延伸产品适合在对功耗、性能、成本和运算能力等综合要求比较高的音频、视频和通信领域中应用。例如,各种制式、各种 PROFILE 和 LEVEL 的视频信号处理可以在一个统一的 Blackfin 平台上实现,易于技术升级,能有效地帮助开发人员提高技术的复用率及降低系统的材料成本和开发成本。

进入了 21 世纪,数字化浪潮席卷而来,高速互联网、宽带接入、无线通信和计算机技术及应用的迅猛发展使视、音频技术在各项工作和生活领域中得到广泛应用,不仅提高了人类的工作效率,改善了生活水平,还提供了高效的传输平台以及接入和处理手段。这些应用包括会议电视、视频点播、数码监控、数码相机、数码摄像机、机顶盒、DVD 播放机、个人数字助理、手机和远程医疗监控装置等。显然,基于传统架构的传统 DSP 和 RISC 控制器及其相关产品已不能满足视频应用的广泛性、灵活性及其标准的多样性和实时应用对多媒体处理器的要求。基于这种预见,美国 Intel 公司和 ADI 公司进行了战略合作,发挥各自在 RISC 和 DSP 上的技术优势,合作开发了全新的 MSA 架构技术,在此基础上,ADI 公司推出了 Blackfin 系列处理器。

在过去的一年中,美国 ADI 公司通过与大学和热衷于 DSP 推广的有关机构合作,举办了许多研讨会、网上培训和实验室培训,以帮助广大工程技术人员尽快学习和掌握在项目开发中使用 Blackfin 处理器。全国涌现出了一大批 Blackfin 处理器的初学者队伍,掌握 Blackfin 处理器的开发技术对于从事高技术开发工作的广大工程技术人员来说具有重要意义。这本专著的出版,对于广大的 Blackfin 处理器应用工程师和初学者来说可谓雪中送炭。本书全面介绍了 Blackfin 处理器的内部架构,硬件开发和主要外设,Blackfin 指令集,Visual DSP++开发系统,软件编程,操作系统内核,并结合视频开发的经验,介绍了基于 Blackfin 处理器的 MPEG-4 开发经验和代码实例。它是一本理想的大学教学参考书。谨借此机会对清华大学的教师在 Blackfin 处理器平台上取得的卓越科研和产业化成果并结合使用 Blackfin 处理器的经验成功地出版本书表示祝贺和感谢。

 博士

美国 ADI 公司中国 DSP 业务发展经理

前 言

信号处理是近年来十分活跃的领域，而以高速数字信号处理器（DSP）为基础的处理技术正在迅速发展。DSP 作为信号处理的基本硬件工具，在雷达、多媒体、通信、控制、航天和医疗等领域发挥着重要的作用。所以选择并掌握一种 DSP 芯片具有很大的现实意义。

作为国际上 DSP 芯片的主要供应商，美国模拟器件公司（简称 ADI）一直致力于发展高性价比的产品。其业务包括混合信号 DSP、通用 DSP，以及可靠数据服务、ADSL 调制解调器、GSM 手机、因特网接入、语音处理和电机控制等应用中的嵌入式 DSP。其成熟的 ADSP21XX 系列 DSP 在实际应用中获得了广泛的应用，得到用户的充分肯定。

Blackfin 系列 DSP 是 ADI 公司最新的基于微信号体系结构的 DSP，它具有两个 MAC，集成了大量的外围设备和存储器接口，每秒运算速度最高达到 1200 MMAC（兆次乘法加法运算），适用于各种视频、音频、通信领域。本书将集中介绍 Blackfin 系列 DSP 的结构、外设、开发环境和指令并给出视频压缩算法 MPEG-4 的应用实例。

本书共 8 章。第 1 章介绍了 DSP 芯片的基本特点和 Blackfin 系列 DSP 的结构和基本性能指标。第 2 章详细介绍了该芯片的内部结构，包括内核结构、算术逻辑单元和乘法器，存储器的结构，片内总线和内部时钟，程序控制器和指令流水线，中断的过程、服务和中断控制。第 3 章介绍了 DSP 的外设，包括外部总线，DMA 的使用，定时器，各种串口（共 3 种，同步外设接口（SPI）、串行接口（SPORT）、统一异步收发器（UART））的应用，USB 接口，PCI 接口，并口 PPI（支持视频输入和输出），启动设置，系统时钟，锁相环，动态电源管理，工作模式设定，可编程端口以及高频设计。

第 4 章介绍了 Blackfin DSP 的指令集，其中包括视频专用指令。第 5 章介绍了 DSP 的开发环境 Visual DSP++，包括 C/C++ 编译器，汇编和预处理，链接器和加载器的使用方法。第 6 章介绍了软件编程中用到的编译器、处理器和加载器的具体命令及设置，以及 C 语言和汇编语言的混合编程。第 7 章介绍了专用操作系统内核 VDK 的使用。

第 8 章介绍了 Blackfin DSP 在视频编码上的应用，给出了几个用于 MPEG-4 算法中的典型函数的优化实例。

在本书的编写过程中，黄晓伟、王宁、刘新、杜友田在收集资料和文稿校阅等方面做了大量工作，同时还得到了 ADI 公司及其第三方北京诺德世讯公司的大力支持和帮助，其中李川、袁潮、路鹤松和张增波为本书提供了翔实的数据和代码程序，在此一并向他们表示衷心的感谢。

由于作者水平有限，书中错误之处在所难免，恳请广大读者给予批评指正。

编著者

2003 年 8 月

目 录

第 1 章 数字信号处理和 DSP	(1)
1.1 数字信号处理的发展	(1)
1.2 DSP 在信号处理中的应用	(2)
1.3 DSP 芯片的特点	(3)
1.4 Blackfin 系列 DSP	(4)
1.4.1 Blackfin DSP 结构综述	(5)
1.4.2 Blackfin ADSP-BF535	(6)
1.4.3 Blackfin ADSP-BF531	(8)
1.4.4 Blackfin ADSP-BF532	(9)
1.4.5 Blackfin ADSP-BF533	(10)
1.4.6 开发工具	(10)
第 2 章 Blackfin DSP 内部结构	(12)
2.1 Blackfin DSP 内核结构	(12)
2.1.1 Blackfin DSP 的内核结构概述	(12)
2.1.2 寄存器组	(12)
2.1.3 算术逻辑单元 (ALU)	(14)
2.1.4 乘法-累加器 (乘法器)	(15)
2.1.5 桶形移位器	(17)
2.2 存储器	(17)
2.2.1 内存结构	(17)
2.2.2 片内存储器	(19)
2.3 片内总线	(30)
2.3.1 内部接口	(30)
2.3.2 Blackfin DSP 内部时钟	(31)
2.3.3 内核概述	(31)
2.3.4 系统概述	(32)
2.3.5 系统接口	(33)
2.4 程序控制器	(36)
2.4.1 概述	(36)
2.4.2 相关的寄存器	(36)
2.4.3 指令流水线	(38)
2.4.4 程序流的控制	(39)
2.5 中断	(40)
2.5.1 概述	(40)
2.5.2 系统中断过程	(41)

2.5.3	中断控制寄存器	(43)
2.5.4	事件控制器的寄存器	(46)
2.5.5	中断的全局禁止和使能	(47)
2.5.6	事件向量表	(47)
2.5.7	硬件错误中断	(51)
2.5.8	中断服务	(52)
2.5.9	中断的嵌套	(53)
第 3 章	Blackfin 硬件开发和主要外设	(57)
3.1	外部总线	(57)
3.1.1	概述	(57)
3.1.2	外部总线框图	(57)
3.1.3	内部存储器接口	(57)
3.1.4	EBIU 仲裁	(58)
3.1.5	外部存储器接口	(58)
3.1.6	EBIU 编程模板	(59)
3.1.7	出错检测	(59)
3.1.8	异步存储器接口	(60)
3.1.9	SDRAM 控制器 (SDC)	(60)
3.2	DMA	(61)
3.2.1	概述	(61)
3.2.2	基于描述符的 DMA 传送	(62)
3.2.3	基于自动缓冲的 DMA	(64)
3.2.4	外设 DMA 寄存器列表	(65)
3.2.5	存储器 DMA (MemDMA)	(66)
3.2.6	DMA 中止条件	(66)
3.2.7	DMA 总线错误条件	(66)
3.2.8	BF531/2/3 的 DMA 特点	(67)
3.3	TIMER	(69)
3.3.1	概述	(69)
3.3.2	通用定时器	(69)
3.3.3	定时器模式	(70)
3.3.4	内核定时器	(71)
3.3.5	看门狗定时器	(71)
3.4	SPI	(72)
3.4.1	概述	(72)
3.4.2	接口信号	(73)
3.4.3	SPI 中断	(74)
3.4.4	SPI 寄存器	(74)
3.4.5	SPI 传输格式	(75)
3.4.6	SPI 通用操作	(76)

3.4.7	错误信号和标志	(78)
3.4.8	SPI 传送的开始和结束	(78)
3.5	SPORT	(79)
3.5.1	概述	(79)
3.5.2	SPORT 的常见操作	(81)
3.5.3	SPORT 的禁止	(81)
3.5.4	SPORT 的模式设置	(82)
3.5.5	时钟和帧同步频率	(82)
3.5.6	时钟信号选择	(82)
3.5.7	帧同步选择	(83)
3.5.8	多通道操作	(85)
3.5.9	数据在 SPORT 和存储器间的移动	(85)
3.5.10	对标准协议的支持	(86)
3.6	UART	(86)
3.6.1	概述	(86)
3.6.2	串行通信	(86)
3.6.3	非 DMA 模式	(87)
3.6.4	DMA 模式	(87)
3.6.5	混合模式	(88)
3.6.6	IrDA 模式	(88)
3.7	USB	(88)
3.7.1	概述	(88)
3.7.2	USB 结构	(90)
3.7.3	功能和模式	(91)
3.7.4	相关寄存器	(93)
3.7.5	中断	(94)
3.7.6	USB 编程模式	(97)
3.8	PCI (外围部件互连)	(104)
3.8.1	概述	(104)
3.8.2	PCI 总线标准	(105)
3.8.3	PCI 设备功能	(105)
3.8.4	PCI 主机功能	(105)
3.8.5	处理器核对 PCI 空间的访问	(105)
3.8.6	外部 PCI 要求	(105)
3.8.7	设备模式操作	(105)
3.8.8	主机模式操作	(109)
3.8.9	输出配置	(110)
3.8.10	复位和相应控制	(110)
3.8.11	中断和相应控制	(111)
3.8.12	PCI 编程模式	(111)

3.8.13	系统 MMR 控制和状态寄存器	(111)
3.8.14	配置空间控制和状态寄存器	(114)
3.8.15	PCI I/O 接口	(114)
3.8.16	反射波开关	(114)
3.8.17	上电顺序	(114)
3.8.18	PCI 时钟要求	(115)
3.9	PPI	(115)
3.9.1	PPI 存储-映射寄存器	(115)
3.9.2	ITU-656 简介	(116)
3.9.3	PPI 的工作模式	(117)
3.9.4	DMA 操作	(119)
3.10	BOOT	(119)
3.10.1	引导模式设定	(119)
3.10.2	引导过程	(120)
3.11	PLL	(121)
3.11.1	时钟系统概述	(121)
3.11.2	锁相环与时钟控制	(121)
3.11.3	PLL 时钟倍频比	(121)
3.11.4	PLL 存储器映像寄存器 (MMR)	(123)
3.12	动态电源管理	(125)
3.12.1	动态电源管理控制器	(125)
3.12.2	运行方式	(125)
3.12.3	运行方式的切换	(126)
3.12.4	外设的定时	(128)
3.12.5	动态电源电压控制	(129)
3.13	工作模式设定	(130)
3.13.1	概述	(130)
3.13.2	用户模式	(131)
3.13.3	监控模式	(133)
3.13.4	仿真模式	(134)
3.13.5	空闲状态	(134)
3.13.6	复位状态	(134)
3.13.7	系统复位和上电设置	(135)
3.14	可编程端口	(137)
3.14.1	概述	(137)
3.14.2	可编程标志存储器映射寄存器 (MMR)	(138)
3.15	硬件设计	(142)
3.15.1	概述	(142)
3.15.2	管脚处理	(142)
3.15.3	DSP 复位	(144)

3.15.4	DSP 引导	(144)
3.15.5	DSP 时钟管理	(145)
3.15.6	中断配置和服务	(146)
3.15.7	信号量	(146)
3.15.8	PCI 仲裁	(147)
3.15.9	USB 设备连接	(147)
3.15.10	外部存储器设计	(147)
3.15.11	高频设计	(149)
第 4 章	Blackfin 指令集	(152)
4.1	Blackfin 指令说明	(152)
4.1.1	语约定	(152)
4.1.2	记约定	(152)
4.1.3	术语表	(153)
4.2	程序流程控制指令	(156)
4.2.1	跳转指令	(156)
4.2.2	条件跳转指令	(156)
4.2.3	调用和返回指令	(157)
4.2.4	循环指令	(157)
4.3	加载/存储 (LOAD/STORE) 指令	(158)
4.3.1	加载 (LOAD) 指令	(158)
4.3.2	存储 (STORE) 指令	(159)
4.4	寄存器数据转移 (MOVE) 指令	(160)
4.4.1	寄存器数据转移指令	(160)
4.4.2	扩展半个字到一个字的数据转移	(161)
4.5	栈控制指令	(161)
4.5.1	进栈指令	(161)
4.5.2	出栈指令	(162)
4.5.3	连接和撤销连接指令	(162)
4.6	逻辑运算指令	(163)
4.6.1	逻辑与指令	(163)
4.6.2	逻辑非指令	(163)
4.6.3	逻辑或指令	(163)
4.6.4	逻辑异或指令	(163)
4.6.5	位方式异或指令	(164)
4.7	移位指令	(166)
4.7.1	算术移位	(166)
4.7.2	逻辑移位	(167)
4.7.3	加后移位	(167)
4.7.4	移位后加	(168)
4.7.5	循环移位	(168)

4.8	算术运算指令	(169)
4.8.1	绝对值指令	(169)
4.8.2	加法指令	(170)
4.8.3	减法指令	(171)
4.8.4	乘法指令	(172)
4.8.5	除法指令	(173)
4.8.6	最大值最小值指令	(174)
4.8.7	取补指令	(174)
4.8.8	舍入指令	(174)
4.8.9	指数检测指令	(175)
4.8.10	饱和指令	(175)
4.8.11	符号位指令	(176)
4.9	位操作指令	(176)
4.9.1	位清除指令	(176)
4.9.2	位设置指令	(176)
4.9.3	位翻转指令	(176)
4.9.4	位测试指令	(177)
4.9.5	存放指令	(177)
4.9.6	位提取指令	(178)
4.9.7	位复用指令	(180)
4.9.8	ONES 指令 (统计寄存器中 1 的个数)	(180)
4.10	控制代码位 (CC) 操作指令	(181)
4.10.1	比较数据寄存器、指针、累加器指令	(181)
4.10.2	取补 CC 指令	(181)
4.10.3	传送 CC 指令	(181)
4.11	外部事件操作指令	(182)
4.11.1	空闲指令	(182)
4.11.2	内核同步指令	(182)
4.11.3	系统同步指令	(182)
4.11.4	强制仿真指令	(183)
4.11.5	中断禁止指令	(183)
4.11.6	中断允许指令	(183)
4.11.7	空指令	(183)
4.12	高速缓存 (CACHE) 控制指令	(184)
4.12.1	数据缓存预取指令	(184)
4.12.2	数据缓存刷新指令	(184)
4.12.3	数据缓存行刷新和无效指令	(184)
4.12.4	指令缓存无效指令	(185)
4.13	视频像素操作指令	(185)
4.13.1	ALIGN8, ALIGN16, ALIGN24	(185)

4.13.2	禁止校正异常指令	(186)
4.13.3	双 16 位累加器加法提取指令	(186)
4.13.4	BYTEOP1P (4 个 8 位数求平均值, 结果仍是 Byte)	(186)
4.13.5	BYTEOP2P (4 个 8 位数求平均值, 结果是半字)	(187)
4.13.6	BYTEOP3P (双 16 位 Add / Clip)	(188)
4.13.7	BYTEOP16P (4 个 8 位数相加)	(189)
4.13.8	BYTEOP16M (4 个 8 位数相减)	(190)
4.13.9	SAA (4 个 8 位数相减取绝对值累加)	(190)
4.13.10	BYTEPACK (4 个 8 位数打包)	(190)
4.13.11	BYTEUNPACK (4 个 8 位数解包)	(191)
第 5 章	Visual DSP++	(192)
5.1	Visual DSP++ 总体介绍	(192)
5.1.1	Visual DSP 简单介绍	(192)
5.1.2	开发环境	(195)
5.2	C/C++ 编译器	(206)
5.2.1	在命令行运行编译器	(206)
5.2.2	在 Visual DSP++ 中指定编译器选项	(207)
5.2.3	优化控制	(208)
5.3	汇编与预处理	(209)
5.3.1	汇编器	(209)
5.3.2	预处理器	(211)
5.4	链接器	(211)
5.4.1	链接过程概述	(211)
5.4.2	链接文件	(212)
5.4.3	链接器参数选择	(213)
5.4.4	专家链接器	(215)
5.5	加载器	(219)
5.5.1	从命令行运行加载器	(219)
5.5.2	加载器命令行参数开关	(220)
5.5.3	从 Visual DSP++ 配置加载器	(222)
第 6 章	软件编程	(224)
6.1	汇编编译器	(224)
6.1.1	汇编概要	(224)
6.1.2	支持 C 结构的汇编器	(224)
6.1.3	汇编语法	(225)
6.2	C 编译器	(231)
6.2.1	C/C++ 编译器简介	(231)
6.2.2	数据类型存储长度	(231)
6.2.3	C/C++ 编译语言扩展	(232)
6.3	预处理命令	(236)

6.3.1	头文件·····	(236)
6.3.2	宏的编写·····	(236)
6.3.3	条件汇编和编译·····	(237)
6.4	C 代码优化·····	(241)
6.5	C 与汇编混合编程·····	(242)
6.5.1	C/C++和汇编混合命名规则·····	(242)
6.5.2	C 运行环境下寄存器的使用·····	(243)
6.5.3	参数传递和局部堆栈存储分配——LINK 和 UNLINK 指令·····	(243)
6.5.4	在 C/C++程序中调用汇编子程序·····	(244)
6.5.5	在汇编程序中调用 C/C++函数·····	(245)
第 7 章	Visual DSP++内核 ·····	(247)
7.1	操作系统内核·····	(247)
7.1.1	内核的作用·····	(247)
7.1.2	调度·····	(247)
7.2	VDK 的组成·····	(248)
7.2.1	线程·····	(248)
7.2.2	调度·····	(249)
7.2.3	中断服务例行程序·····	(250)
7.3	VDK 程序的开发·····	(251)
第 8 章	开发实例——MPEG-4 编码 ·····	(256)
8.1	MPEG-4 编码介绍·····	(256)
8.1.1	MPEG-4 视频压缩方法简介·····	(256)
8.1.2	DCT 和 IDCT·····	(256)
8.1.3	量化·····	(257)
8.1.4	运动估计·····	(257)
8.1.5	熵编码·····	(257)
8.2	代码实例·····	(258)
8.2.1	PADDING 代码的编写和优化·····	(258)
8.2.2	DCT 变换代码的编写和优化·····	(264)
8.2.3	运动矢量预测·····	(265)
8.2.4	SAD 最小搜索·····	(271)
参考文献	·····	(273)

第 1 章 数字信号处理和 DSP

1.1 数字信号处理的发展

半个世纪以来,在计算机和信息技术的带动下,数字信号处理(Digital Signal Processing)技术得到迅速发展。它已经在数据处理、工业控制和数据通信等许多领域得到广泛的应用。数字信号处理的主要功能包括: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。

数字信号处理亦称信号的数字处理。从信号的数字处理技术的研究史来看,可以归纳为以下几个阶段:

- (1) 信号解析手段的研究阶段;
- (2) 各种模拟信号的数字化阶段;
- (3) 信号数字处理技术本身的发展阶段;
- (4) 现代数字信号处理阶段。

前两个阶段发生在 17 世纪末 18 世纪初离散数学诞生到 20 世纪 60 年代之间。第三阶段是以 1965 年 Colley 和 Turkey 提出快速傅里叶变换算法(FFT)为标志的,这时的 DSP 技术主要用于图像处理、快速数据传输、生物医学系统等领域。第四阶段的时间划分并不明确,但其重要特点是新理论、新算法不断涌现,硬件处理水平得到极大提高,应用领域不断扩大,数字信号处理的高速实时应用成为研究的重点和热点。同时,数字信号处理的应用领域也在不断扩大。

从信号的数字处理技术的发展历程可以看出,数字信号处理技术是以众多学科为理论基础的一门新兴的交叉学科,所涉及的研究范围是相当广泛的。例如,在数学领域,微积分、概率统计、随机过程、数值分析、近世代数、复变函数、线性代数、泛函分析等都是数字信号处理的分析工具;网络理论、图论、信号与系统、控制论、通信理论等均是数字信号处理的理论基础。在学科发展上,数字信号处理又是现代控制理论(包括最优控制、人工智能、模式识别、神经网络、模糊控制)、现代通信理论、故障检测和诊断理论以及现代测量等学科或技术的理论基础。在算法实现上,数字信号处理技术和计算机学科及微电子技术密不可分。可以说,数字信号处理是在许多经典理论体系的基础上发展而来的,同时又已成为一系列新兴学科的理论基础,并与它们相互交叉、互相促进。

数字信号处理理论经过 30 年的发展已经形成了比较完善的理论体系。其主要内容包括^[1]:

- (1) 信号的采集(A/D 技术、采样定理、多采样率、量化噪声分析等);
- (2) 离散系统分析(系统的描述、系统的单位采样响应、转移函数及频率特性等);
- (3) 离散信号的分析(时域及频域分析、多种变换技术、信号特征的描述等);
- (4) 信号处理的快速算法(快速傅里叶变换、快速卷积与相关等);
- (5) 信号的估值(各种估值理论、相关函数与功率谱估计);
- (6) 信号的建模(最常用的是 AR、MA、ARMA、PRONY 等各种模型);
- (7) 滤波技术(各种数字滤波器的设计与实现);

(8) 信号处理中的其他特殊算法(如采样、插值、奇异值分解、反卷积、信号重建等)。

现在, 数字信号处理进入了一个新的发展时期: 在优化、自适应、高分辨率、多维多通道分析方面的理论和方法日趋系统化; 对系统的分析已不再限于理想模型, 而是考虑到各种实际因素, 研究其鲁棒性; 对性能的描述也不仅仅停留在定性的水平, 而是力求给出系统的统计性能评价。以前通常假设信号及其背景是高斯的、平稳的, 对信号的分析局限于它的二阶矩和傅氏谱, 对象系统也仅限于时不变的线性因果最小相位系统。随着数字信号处理技术的不断成熟, 人们开始研究非平稳、非高斯的信号与背景噪声, 研究时变、非因果、非最小相位、非线性性的系统——这些都是现代信号处理的热点问题。

数字信号处理相对于模拟信号处理有很大的优越性, 比如它的精度高、灵活性大、可靠性好、易于大规模集成、易于存储, 而且可以采用多种性能优良的数字信号处理方法和算法。总之, 随着基础理论的不完善、相关学科的不断发展和微电子技术计算机技术的不断进步, 可以预见, 21 世纪将是数字信号处理理论与算法的大发展时期。

1.2 DSP 在信号处理中的应用

数字信号处理系统的实现方法一般有以下儿种:

(1) 在通用计算机上用软件实现。软件可以由用户自己编写, 也可以是现成的软件包。这种方法的缺点是速度慢, 一般只能用于算法验证、仿真研究或信号的离线处理, 而不适用于实时系统。近年发展迅速的 MATLAB 软件, 几乎可以实现所有数字信号处理的仿真, 而且 MATLAB 下的部分仿真程序还可以转化为 C 语言, 再通过 DSP 的 C 编译器直接在 DSP 硬件上运行, 这对非实时系统或准实时系统来说是很有吸引力的。

(2) 在通用计算机系统中加上专用的加速处理机(比如单片机)实现。单片机的接口性能良好, 设计简单, 速度也比较快。但由于单片机普遍采用的是冯·诺依曼总线结构, 所以数据运算(尤其是关键的乘法运算)速度慢, 在运算量大的实时控制系统中很难有所作为。

(3) 用通用的单片机(如 MCS-51、96 系列等)实现, 这种方法只可用于一些不太复杂的数字信号处理场合, 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优点: 如普遍采用改进的哈佛总线结构, 内部有硬件乘法器、累加器, 使用流水线结构, 具有良好的并行特性, 并有专门设计的适于数字信号处理的指令系统等。DSP 芯片的这些特点为不允许延迟的实时应用领域, 如蜂窝电话、计算机驱动器等提供了一个理想的选择。可以说, DSP 芯片的问世及飞速发展, 为信号处理技术应用于实际工程提供了可能。

(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, 用一般的通用 DSP 芯片也很难实现。但有一些专用芯片将常用的信号处理算法(如 FFT、卷积、相关等)在芯片内部用硬件实现, 无须进行软件编程。这种方案的缺点是灵活性差、成本高, 而且开发工具尚不完善。

综上所述, 第 1 种方法的缺点是速度较慢, 一般可用于 DSP 算法的模拟; 第 2 种和第 5 种方法专用性强, 应用受到很大的限制, 第 2 种方法也不便于系统的独立运行; 第 3 种方法只适用于简单 DSP 算法的实现; 只有第 4 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 20 世纪 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,才使得将理论研究结果应用到低成本的实际系统中成为可能,并且由此推动了新的理论和新的应用领域的发展。可以毫不夸张地说,20 世纪 70 年代末 80 年代初 DSP 芯片的诞生及发展对近 20 年来通信、计算机、控制等领域尤其是信号处理技术的发展起到十分重要的作用。

DSP 芯片的高速发展,一方面得益于集成电路技术的发展,另一方面也得益于巨大的市场。目前,DSP 芯片的价格越来越低,性能价格比日益提高,具有巨大的应用潜力。DSP 芯片的应用主要有^[2]:

(1) 通用信号处理——如数字滤波、自适应滤波、快速傅里叶变换、相关运算、谱分析、卷积、模式匹配、加窗、波形产生等;

(2) 通信——如调制解调器、编/译码器、自适应均衡器、数据加密、数据压缩、回波抵消、噪声抑制、多路复用、传真、程控交换机、蜂窝移动电话、数字基站、数字留言机、扩频通信、纠错编码、可视电话等;

(3) 语音——如语音编码、语音合成、语音识别、语音增强、说话人辨认、说话人确认、语音邮件、语音存储等;

(4) 图形/图像——如二维和三维图像处理、图像压缩与传输、图像增强、动画、机器人视觉等;

(5) 军事——如保密通信、雷达信号处理、声呐处理、导航、导弹制导等;

(6) 仪器仪表——如频谱分析、函数发生、锁相环、地震信号处理等;

(7) 自动控制——如引擎控制、声控、自动驾驶、机器人控制、磁盘控制等;

(8) 医疗——如助听、超声设备、诊断工具、病人监护等;

(9) 计算机与工作站——阵列处理机、计算加速卡、图形加速卡、多媒体计算机等;

(10) 家用电器——如高保真音响、音乐合成、音调控制、玩具与游戏、数字电话/电视、汽车电子装置、住宅电子安全系统等。

随着 DSP 芯片性能价格比的不断提高,可以预见 DSP 芯片将会在更多的领域内得到更为广泛的应用。

1.3 DSP 芯片的特点

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,主要用于实时快速实现各种数字信号处理算法。为此,DSP 除了具备普通微处理器所强调的高速运算、控制功能外,还针对实时数字信号处理,在处理器结构、指令系统、指令流程上做了很大的改动,其结构特点如下^[3]:

(1) DSP 普遍采用了数据总线和程序总线分离的哈佛结构及改进的哈佛结构,比传统处理器的冯·诺依曼结构有更高的指令执行速度。哈佛结构的主要特点是将程序和数据存储在不同的存储空间中,即程序存储器和数据存储器是两个相互独立的存储器,每个存储器独立编址,独立访问。与两个存储器相对应的是系统中设置了程序总线 and 数据总线,从而使数据的吞吐率提高了一倍。由于程序和存储器在两个分开的空间中,因此取指和执行能完全重叠。尽管这样做的代价是系统复杂化,但半导体工业的进步已使这一结构变得容易实现。

(2) DSP 大多采用流水线技术,即每条指令都由片内多个功能单元分别完成取指、译码、取数、执行等步骤,就像经过工厂里的生产流水线上的一道道工序一样。而同时,在流水线的

其他阶段又分别有其他的指令在顺序地执行着。采用指令流水线的执行方式,可以大大提高系统的执行效率,使得系统可以低延迟或“无延迟”地执行较复杂的指令。从而在不提高时钟频率的条件下减少了每条指令的执行时间,进一步增强了处理器的数据处理能力。但流水线的采用也增加了软件设计的难度,要求设计者在程序设计的过程中尽量考虑流水的需要。

(3) 针对滤波、相关、矩阵运算等特点, DSP 大都配有独立的乘法器和加法器,使得在同一时钟周期内可以完成相乘、累加两个运算。有的 DSP 还可以同时进行加、减、乘运算,大大加快了 FFT 的蝶形运算速度。相比之下,在一般的计算机上,算术逻辑单元 (ALU) 只能完成两个操作数的加、减及其他逻辑运算,而乘法(或除法)则由加法和移位来实现。因此,在这样的计算机的汇编语言中虽然有乘法指令,但在机器内部,实际上还是由加法和移位来实现的。因此它们实现乘法运算就比较慢。在数字信号处理运算中,无论是滤波器,还是 DFT、FFT 运算,一般的算法中都有大量的乘法运算存在。乘法运算的速度是数字信号处理实现中的一个瓶颈问题,因此 Colley 和 Tukey 将 DFT 的乘法运算量由 N^2 次降为 $N/2 \lg N$ 次,被公认为是数字信号处理发展史上的一个转折点,并以此作为数字信号处理这一学科的开端。同样,其他各种算法的改进也将降低算法中的乘法运算次数作为一项主要的目标。在算法(软件)上减少乘法运算次数的同时,在硬件上, DSP 的硬件乘法器使得乘法运算可以在一个指令周期内完成。

(4) 片内有多条总线可以同时进行取指令和多个数据存取操作,并且有辅助寄存器用于寻址,他们可以在当前访问前/后自动修改内容以指向下一个要访问的地址(自动变址),并且支持循环寻址和位反序寻址。

(5) 具有硬件接口逻辑和软件等待功能,使得 DSP 这种快速芯片能与 SDRAM、SRAM、FLASH、FIFO 等各种慢速存储器接口。

(6) 许多 DSP 带有多个 DMA 通道控制器,配合片内多总线结构,可以实现内存之间,内存和外设之间的整块数据传输,数据块传送速度大大提高。

(7) 配有中断处理器、定时控制器和实时时钟,可以很方便地构成一个小规模系统。

(8) 低功耗,一般为 $0.5 \sim 4 \text{ W}$,采用低功耗技术的 DSP 甚至只有 0.05 W ,可用电池供电,对嵌入式系统很合适。相比之下, Pentium、PowerPC 等普通微处理器的功耗则一般为 $20 \sim 50 \text{ W}$ 。

(9) 良好的多机并行运行特性。在一定技术条件下, DSP 芯片的单机处理能力是有限的,系统的数据处理容量还是经常会超出单个 DSP 的处理能力。随着 DSP 芯片的逐年广泛使用和 DSP 芯片价格的不断降低,多个 DSP 芯片的并行处理已经成为近年来一个研究热点,并逐渐在应用中崭露头角。

(10) 丰富的外设接口,许多 DSP 都在子系列中增加了一些特定的接口,如 USB 和 PCI 等。这使得 DSP 非常适合一些嵌入式的应用。

上面所述的各种结构特点,使得 DSP 芯片非常适合实时数字信号处理。也正是这种优越性,使 DSP 在迈入 21 世纪之际成为数字世界的关键所在。

1.4 Blackfin 系列 DSP

美国模拟器件公司 (ADI) 是全世界领先的高性能信号处理集成电路制造商,是全球主要的可编程 DSP 芯片供应商之一。在通用 DSP 市场上, ADI 公司占有约 40% 的市场份额。Blackfin DSP 是 16 位产品的一个大系列,是 ADI 与 Intel 联合开发的体现高性能体系结构的首款第四