

面向 **21** 世纪

高等学校信息工程类专业系列教材

现代 DSP 技术

Modern DSP Techniques

潘松 黄继业 王国栋 编著

2

西安电子科技大学出版社
<http://www.xduph.com>

面向 21 世纪高等学校信息工程类专业系列教材

现代 DSP 技术

Modern DSP Techniques

潘 松 黄继业 王国栋 编著

西安电子科技大学出版社

2003

内 容 简 介

本书详尽介绍了目前在电子信息和通信领域被广泛应用的数字信号处理硬件实现的全新解决方案,即基于 EDA 与 SOPC 的现代 DSP 开发技术,以及与之相关的开发工具的使用方法、设计理论和一些典型的设计实例。全书内容包括基于 MATLAB/SimuLink 的 DSP 和通信系统模块的设计技术;Quartus II 的基本使用方法、详细的设计流程向导、多种优化设计方法、逻辑锁定技术、嵌入式逻辑分析仪 SignalTap II 的使用方法、Quartus II/DSP Builder 及第三方 EDA 工具 Synplify、Leonardo Spectrum 和 Modelsim 的优化设计接口技术。全书深入、系统地介绍了基于 MATLAB 环境下 DSP Builder/SignalCompiler 对现代 DSP 系统开发的基本方法,以及基于这些工具的 DSP IP 核的使用方法和使用规则。

本书内容新颖、实用,为 DSP 领域的读者展示了有别于传统 TI DSP 处理器的、全新的 DSP 系统实现技术,为软件无线电领域的读者提供了一项不可或缺的系统设计解决方案,也为电子信息领域的读者向 EDA 技术的更高层次迈进提供了有用的工具。

本书可作为电子类各专业高年级本科生、研究生的教材,或作为相关领域工程技术人员的参考书,也可作为现代电子系统设计、电子设计竞赛、DSP 应用系统以及通信电子系统高层次开发的参考书,或用作 SOC/SOPC 技术实验教材。

★ 本书配有电子教案,需要者可与出版社联系,免费索取。

图书在版编目(CIP)数据

现代 DSP 技术=Modern DSP Techniques / 潘松等编著. —西安:西安电子科技大学出版社, 2003.8

(面向 21 世纪高等学校信息工程类专业系列教材)

ISBN 7-5606-1281-4

I. 现… II. 潘… III. 数字信号—信号处理—高等学校—教材 IV. TN911.72

中国版本图书馆 CIP 数据核字(2003)第 058673 号

策 划 马乐惠

责任编辑 李惠萍

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)8227828 8201141 邮 编 710071

http://www.xduph.com

E-mail: xdupfxb001@163.com xdupfxb001@sina.com

经 销 新华书店

印刷单位 陕西光大印务有限责任公司

版 次 2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 19.375

字 数 456 千字

印 数 1~4 000 册

定 价 22.00 元

ISBN 7-5606-1281-4 / TN·0234 (课)

XDUP 1552001 - 1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

序

第三次全国教育工作会议以来,我国高等教育得到空前规模的发展。经过高校布局和结构的调整,各个学校的新专业均有所增加,招生规模也迅速扩大。为了适应社会对“大专业、宽口径”人才的需求,各学校对专业进行了调整和合并,拓宽专业面,相应的教学计划、大纲也都有了较大的变化。特别是进入 21 世纪以来,信息产业发展迅速,技术更新加快。面对这样的发展形势,原有的计算机、信息工程两个专业的传统教材已很难适应高等教育的需要,作为教学改革的重要组成部分,教材的更新和建设迫在眉睫。为此,西安电子科技大学出版社聘请南京邮电学院、西安邮电学院、重庆邮电学院、吉林大学、杭州电子工业学院、桂林电子工业学院、北京信息工程学院、深圳大学、解放军电子工程学院等 10 余所国内电子信息类专业知名院校长期在教学科研第一线工作的专家教授,组成了高等学校计算机、信息工程类专业系列教材编审专家委员会,并且面向全国进行系列教材编写招标。该委员会依据教育部有关文件及规定对这两大类专业教学计划和课程大纲,对目前本科教育的发展变化和相应系列教材应具有的特色和定位以及如何适应各类院校的教学需求等进行了反复研究、充分讨论,并对投标教材进行了认真评审,筛选并确定了高等学校计算机、信息工程类专业系列教材的作者及审稿人。这套教材预计在 2004 年春季全部出齐。

审定并组织出版这套教材的基本指导思想是力求精品、力求创新、好中选优、以质取胜。教材内容要反映 21 世纪信息科学技术的发展,体现专业课内容更新快的要求;编写上要具有一定的弹性和可调性,以适合多数学校使用;体系上要有所创新,突出工程技术型人才培养的特点,面向国民经济对工程技术人才的需求,强调培养学生较系统地掌握本学科专业必需的基础知识和基本理论,有较强的本专业的基本技能、方法和相关知识,培养学生具有从事实际工程的研发能力。在作者的遴选上,强调作者应在教学、科研第一线长期工作,有较高的学术水平和丰富的教材编写经验;教材在体系和篇幅上符合各学校的教学计划要求。

相信这套精心策划、精心编审、精心出版的系列教材会成为精品教材,得到各院校的认可,对于新世纪高等学校教学改革和教材建设起到积极的推动作用。

系列教材编委会
2002 年 8 月

高等学校计算机、信息工程类专业

系列教材编审专家委员会

主任：杨震（南京邮电学院副院长、教授）

副主任：张德民（重庆邮电学院通信与信息工程学院院长、教授）

韩俊刚（西安邮电学院计算机系主任、教授）

李荣才（西安电子科技大学出版社总编辑、教授）

计算机组

组长：韩俊刚（兼）

成员：（按姓氏笔画排列）

王小民（深圳大学信息工程学院计算机系主任、副教授）

王小华（杭州电子工业学院计算机分院副院长、副教授）

孙力娟（南京邮电学院计算机系副主任、副教授）

李秉智（重庆邮电学院计算机学院院长、教授）

孟庆昌（北京信息工程学院教授）

周娅（桂林电子工业学院计算机系副主任、副教授）

张长海（吉林大学计算机科学与技术学院副院长、教授）

信息工程组

组长：张德民（兼）

成员：（按姓氏笔画排列）

方强（西安邮电学院电信系主任、教授）

王晖（深圳大学信息工程学院电子工程系主任、副教授）

胡建萍（杭州电子工业学院电子信息分院副院长、副教授）

徐祎（解放军电子工程学院电子技术教研室主任、副教授）

唐宁（桂林电子工业学院通信与信息工程系副主任、副教授）

章坚武（杭州电子工业学院通信工程分院副院长、教授）

康健（吉林大学通信工程学院副院长、教授）

蒋国平（南京邮电学院电子工程系副主任、副教授）

总策划：梁家新

策划：马乐惠 云立实 马武装 马晓娟

电子教案：马武装



前言

一般而言,数字信号处理(DSP)是指应用数字的方法(非模拟电子技术)处理各种类型信息的基本理论和基本算法;而 DSP 技术,或者说 DSP(系统实现)开发技术主要是指将 DSP 基本理论和算法付诸实现的途径和方法。这是两个根本不同的概念,然而却常常被混淆,以致于一谈起 DSP,难免让人不由自主地联想到类似于 TI 的 DSP 器件,误认为 DSP 技术等同于 DSP 处理器的应用,认为 DSP 的实现方式只能是 DSP 处理器。这不能不让人想到是否是由于某种隐含的,但却是长时间的商业暗示所致。

由于 DSP 应用系统实现的复杂性和某些硬件性能的局限性,一般地, DSP 技术的发展总是滞后于基本理论的研究。特别是面对现代电子产品品种和性能的更高要求,这种矛盾已显得更加突出。这主要表现在随着 DSP 应用领域的拓宽和各类性能指标的不断提升,采用 DSP 处理器(如 TI 的 TMS320C 系列)的解决方案日益面临着不断增加的巨大挑战,而自身的技术瓶颈(如运行速度、吞吐量、总线结构的可变性、系统结构的可重配置性、硬件可升级性等等)致使这种解决方案在 DSP 的许多新的应用领域中的道路越走越窄。例如,软件无线电的概念在 10 年前就已提出,并付诸研究和实现,仅其频域中数字信号处理的等效速度也需达 50 GIPS。然而直到目前为止,最快的 DSP 处理器,如 TI 的 C6X 系列也没能超过 5 GIPS,至于现在最常用的 TMS320C54X 系列处理器,也只有 0.1 GIPS(ALTERA 基于 FPGA 的 DSP 等效速度大于 70 GIPS)。如果采取多片并用,那么其体积、功耗、可靠性、灵活性、结构可重构性乃至产权的自主性等等不利因素暂且不说,就是系统功能的可行性都未必不是一个问题。不言而喻,在许多应用领域,以处理器(DSP Processor)为核心的 DSP 系统具有很大的局限性。

然而,值得注意的是,长期以来国内绝大部分有关 DSP 的著述,无一不是在介绍 DSP 基本理论和算法后,都会毫无例外地仅仅给出了 DSP 处理器的相关内容,却忽视了在给定的实用领域中,书中介绍的处理器能否胜任某种 DSP 算法。

这种倾向的结果难免会给人这样的印象: DSP 处理器是 DSP 系统的惟一选择; DSP 技术等于 DSP 处理器应用; DSP 的开发应用、教学和实验也只需围绕 DSP 处理器来进行就可以了。

事实上,在 DSP 领域,除了普通的 DSP 处理器以外,利用 FPGA 来实现 DSP 系统已十分普遍。而且在许多诸如实时图像处理、联合战术无线电通信系统、3G 移动通信基站、实时工控系统、卫星导航设备等领域中, FPGA 的 DSP 解决方案已成为非此莫属的选择了!

现代 DSP 技术是相对于传统 DSP 技术而言的,后者主要是指目前已广泛使用的 DSP 处理器的解决方案,包括一系列软硬件技术与开发技术;前者则是基于 SOPC(a System On a Programmable Chip)技术、EDA 技术与 FPGA 实现方式的 DSP 技术,是现代电子技术发展的产物,它有效地克服了传统 DSP 技术中的诸多技术瓶颈,在许多方面显示了突出的优势,如高速与实时性,高可靠性,自主知识产权化,系统的重配置与硬件可重构性,单片 DSP 系统的可实现性以及开发技术的标准化和高效率。



本书对现代 DSP 技术作了完整的描述。第 1 章介绍了目前常用的 DSP 设计流程, 其中重点介绍了现代 DSP 系统的设计流程, 并从多侧面对现代 DSP 技术与传统 DSP 技术作了比较。由于 Quartus II 是 DSP 系统实现、SOPC 设计和一般 FPGA 与 CPLD 开发的综合性大型 EDA 开发工具, 因此在第 2 章、第 4 章和第 5 章中结合 DSP 实现, 作了详尽的介绍。其中包括循序渐进式的实例向导、多种途径的优化技术应用, 以及针对基于 FPGA 的 DSP 系统开发的第三方 EDA 工具软件(Synplify、Modelsim、LeonardoSpectrum)与 Altera 的 Quartus II 和 DSP Builder 间的接口方法和综合优化设计技术的应用; 另外在第 5 章中还介绍了 Quartus II 的逻辑锁定技术 LogicLock, 这对于单片大规模 DSP 系统的设计是十分有益和必需的。第 3 章重点介绍了目前被广泛用于许多学科中进行科学计算和工程计算的功能强大的数学分析工具(MathWorks 公司业界杰出的 MATLAB/Simulink), 以及在其环境下, 作为现代 DSP 开发的核心工具 DSP Builder 的使用方法, 包括 DSP 系统设计模型的建立、仿真测试、VHDL 转换、逻辑综合、结构优化、编译适配, 直至硬件实现等详细设计步骤。在该章中还就现代 DSP 技术中最具特色的 IP 核的应用, 以及嵌入式逻辑分析仪 SignalTap II 的使用流程作了介绍。而在第 4 章中则对 DSP 的优化设计作了进一步的叙述。第 6 章至第 10 章主要介绍了利用以上工具完成一些典型 DSP 模块设计的实例。第 11 章介绍了应用 DSP Builder 完成 DSP 模型设计的设计规则。第 12 章至第 14 章对重要的 DSP 设计库及其功能模块作了介绍。

值得注意的是, 在基于 FPGA 的 DSP 开发工具中, 无论是 Altera 的 DSP Builder 还是 Xilinx 的 System Generator, 将 MATLAB/Simulink 的 DSP 模型转换成的硬件描述语言都只限于 VHDL 语言。因此, 建议在本教程的前期课程中, 应适当安排 VHDL 的教学内容, 包括 VHDL 的 RTL 级描述与仿真。

由于现代 DSP 技术是 SOC/SOPC 技术的一个部分, 是 EDA 技术的延续, 因而其设计工具、设计语言和设计流程具有一般性和通用性。读者通过对本书内容的了解, 不仅能认识一项全新的 DSP 开发技术和 EDA 实用技术, 而且还能在开发实践中, 或在教学和实验中, 进一步体会现代电子技术中令人振奋的诸多优势, 如: 将远离硬件的系统级描述与高效的硬件实现有机结合的自顶向下的设计流程; 允许用户能随意把“测试仪表”连同 DSP 系统一起送入目标芯片内部去采集内部信号或卸去的嵌入式逻辑分析仪应用技术; 能随时改变硬件结构的重定制功能等等。显然, 本书除了可用作现代 DSP 技术的学习资料外, 也可作为高校电子信息、通信技术、工业自动化和计算机类 EDA 教学实验课的教材, 或作为 SOC/SOPC 技术实验教材。

本书的出版得到了美国 Altera 公司中国地区经理赵典锋先生和钟屹先生的大力支持和帮助, 作者在此表示衷心的感谢!

由于是初涉 DSP 领域, 书中难免会有失误, 恳请 DSP 界的专家学者不吝赐教, 多多给予批评指正, 作者不胜感激!

作者的电子邮箱是: span88@mail.hz.zj.cn

作者
2003 年 6 月

目 录

第 1 章 概述 1

1.1 DSP 实现方案及设计流程 1

1.1.1 常用 DSP 应用器件及其性能特点 2

1.1.2 DSP 处理器结构与性能的发展 4

1.1.3 FPGA 的结构与性能的发展 4

1.1.4 基于 DSP 处理器的 DSP 设计流程 6

1.1.5 基于 FPGA 的 DSP 设计流程 7

1.2 现代 DSP 设计流程概述 9

1.3 两类 DSP 解决方案的比较 12

习题 18

第 2 章 Quartus II 设计向导 19

2.1 频率计的 VHDL 设计 19

2.1.1 创建工程和编辑设计文件 19

2.1.2 创建工程 23

2.1.3 编译前设置 25

2.1.4 编译及了解编译结果 27

2.1.5 仿真 28

2.1.6 引脚锁定和下载 32

2.1.7 Quartus II 在 Windows 2000 上的 安装设置 36

2.2 使用宏功能块设计频率计 36

2.3 嵌入式系统块 ESB 的应用 I 40

2.3.1 应用普通方法设计正弦信号 发生器 40

2.3.2 应用 ESB 优化正弦信号 发生器的设计 42

2.4 嵌入式系统块 ESB 的应用 II 44

2.4.1 定制 ROM 数据文件 45

2.4.2 定制 ROM 46

2.4.3 编译和结果观察 48

2.5 时序驱动式优化编译方法 49

2.5.1 需求频率值设置 49

2.5.2 优化时钟时序设置 50

2.5.3 编译 51

2.6 频率计结构原理 52

2.7 Quartus II 优化特性 53

习题 55

第 3 章 DSP Builder 设计入门 57

3.1 DSP Builder 及其设计流程 57

3.2 DSP Builder 设计向导 59

3.2.1 建立一个新的模型(Model) 60

3.2.2 Simulink 模型仿真 69

3.2.3 SignalCompiler 的使用 73

3.2.4 使用 ModelSim 验证生成的 VHDL 代码 81

3.2.5 在 Quartus II 中指定器件管脚、 进行编译、下载 84

3.2.6 硬件测试 85

3.3 调幅电路模型设计示例 85

3.3.1 建立调幅电路模型 85

3.3.2 模型仿真和实现 91

3.3.3 在 MATLAB/Simulink 中使用 MegaCore 核 93

3.4 使用 SignalTap II 嵌入式逻辑分析仪 95

3.4.1 安装 SignalTap II 95

3.4.2 设置触发条件 98

3.4.3 设置总线的数据格式 99

3.4.4 启动 SignalTap II 分析操作 99

3.4.5 信号节点的资源利用情况 100

3.4.6 一般触发条件 101

3.4.7 Node 模块 101

习题 102

第 4 章 DSP Builder 设计进阶 103

4.1 层次化设计 103

4.1.1 DSP Builder 的子系统 103



4.1.2 在 DSP Builder 中使用外部的 VHDL 代码	109	5.3 应用逻辑锁定技术	144
4.2 用 ModelSim 进行 RTL 级 VHDL 仿真	111	5.3.1 底层模块设计及其 VQM 文件保存	145
4.3 使用 Synplify 进行综合	116	5.3.2 确定逻辑锁定区域及其特性	147
4.3.1 Synplify 与 DSP Builder 的 接口(自动流程).....	116	5.3.3 将设计实体移至锁定区域	151
4.3.2 Synplify 和 DSP Builder 的 接口(手动流程).....	117	5.3.4 编译优化锁定后的 filter 模块	154
4.3.3 Synplify 与 Quartus II 的接口	122	5.4 顶层设计优化	156
4.4 使用 LeonardoSpectrum 进行综合	122	5.4.1 记录锁定信息	156
4.4.1 LeonardoSpectrum 与 DSP Builder 的接口(自动流程).....	122	5.4.2 逻辑锁定信息的输出	157
4.4.2 LeonardoSpectrum 和 DSP Builder 的接口(手动流程).....	123	5.4.3 逻辑锁定信息的输入	158
4.4.3 LeonardoSpectrum 与 Quartus II 的接口	126	5.4.4 编译和结果观察	160
4.5 Quartus II 与 DSP Builder 的接口	127	习题	162
4.5.1 使用 Quartus II 进行综合、适配 (手动流程).....	127	第 6 章 FIR 数字滤波器设计	163
4.5.2 使用 Quartus II 进行时序仿真	129	6.1 FIR 数字滤波器原理	163
4.5.3 在 Quartus II 中建立 元件(Symbol)	129	6.2 使用 DSP Builder 设计 FIR 数字滤波器	164
习题	130	6.2.1 3 阶常系数 FIR 滤波器的设计	164
第 5 章 LogicLock 优化技术	131	6.2.2 4 阶 FIR 滤波器节的设计	167
5.1 LogicLock 技术的基本内容	131	6.2.3 16 阶 FIR 滤波器模型设计	169
5.1.1 LogicLock 技术解决系统 设计优化	131	6.2.4 使用 MATLAB 的滤波器 设计工具	171
5.1.2 LogicLock 的基本内容	133	6.2.5 16 阶 FIR 滤波器的硬件实现	180
5.1.3 锁定区域的基本方式	133	6.3 使用 FIR IP Core 设计 FIR 滤波器	180
5.1.4 层次化逻辑锁定区域	134	6.3.1 FIR 滤波器核与 DSP Builder 集成	181
5.1.5 LogicLock 技术的不同应用流程	135	6.3.2 FIR 滤波器核的使用	181
5.1.6 系统性能强化策略	137	习题	186
5.1.7 锁定区域的移植与再利用	138	第 7 章 IIR 数字滤波器设计	187
5.2 未用 LogicLock 的数字滤波器设计	138	7.1 IIR 滤波器原理	187
5.2.1 数字滤波器结构及其 VHDL 描述	138	7.2 使用 DSP Builder 设计 IIR 滤波器	189
5.2.2 滤波器设计和结果观察	142	7.2.1 4 阶直接 II 型 IIR 滤波器设计	189
		7.2.2 4 阶级联型 IIR 滤波器设计	192
		7.3 在 Quartus II 中使用 IIR 滤波器 IP 核 ..	197
		7.3.1 配置 Quartus II 以便使用 IIR 滤波器核	198
		7.3.2 使用 IIR 滤波器核	199
		习题	203



第 8 章 FFT 设计	204	11.2 频率设计规则	231
8.1 FFT 的原理	204	11.2.1 单时钟设计规则	231
8.1.1 快速傅立叶变换 FFT	204	11.2.2 多时钟设计	235
8.1.2 快速傅立叶反变换 IFFT	205	11.2.3 使用 PLL 的高级特性	236
8.2 FFT 与蝶形运算	206	11.3 DSP Builder 设计的取名规则	236
8.3 使用 DSP Builder 设计 FFT	206	11.4 定点数据下标说明	237
8.3.1 8 点 DIT FFT 模型的建立	206	11.5 在 SBF 中二进制小数点的位置	238
8.3.2 8 点 DIT FFT 模型的实现	209	11.6 GoTo 和 From 模块的支持特性	238
8.4 在 DSP Builder 中使用 FFT IP Core	210	11.7 MegaCore 功能块支持特性	239
习题	211	11.8 层次化设计	240
第 9 章 DDS 设计	213	11.9 黑盒子化	241
9.1 DDS 的基本原理	213	11.10 将 DSP Builder 设计方式用 在外部 RTL 设计	242
9.2 DDS 的模块设计	216	第 12 章 AltLab 库	243
9.2.1 建立 DDS 模型	216	12.1 SignalCompiler 模块	243
9.2.2 DDS 模型的使用	218	12.1.1 综合域编译流程	243
9.3 FSK 调制器设计	219	12.1.2 数据位宽的传递	245
9.3.1 FSK 调制器原理	219	12.1.3 Tapped Delay Line	246
9.3.2 FSK 模型	219	12.1.4 时钟设置	248
习题	220	12.1.5 DSP Builder 报告文件	250
第 10 章 编码与译码	222	12.2 Subsystem Builder 模块	250
10.1 伪随机序列	222	第 13 章 算术库	252
10.1.1 m 序列	222	13.1 比较器模块	252
10.1.2 m 序列发生器模型	223	13.2 计数器模块	252
10.2 帧同步检出	224	13.3 差分模块	253
10.2.1 巴克码	224	13.4 除法模块	253
10.2.2 巴克码的检出模型	225	13.5 增益模块	254
10.3 RS 码	226	13.6 递增递减模块	255
10.3.1 RS 码简介	226	13.7 乘法累加模块	256
10.3.2 使用 IP Core 设计 RS 编码器	227	13.8 乘加模块	258
10.3.3 使用 IP Core 设计 RS 译码器	228	13.9 并行加减法器模块	259
10.4 Viterbi 译码	229	13.10 乘积模块	260
10.4.1 卷积码的 Viterbi 译码	229	13.11 其它算术模块	261
10.4.2 用 IP Core 设计 Viterbi 译码器	229	13.11.1 SOP TAP 模块	261
习题	229	13.11.2 流水线加法器模块	261
第 11 章 DSP Builder 设计规则	231	13.11.3 积分模块	262
11.1 位宽设计规则	231		



第 14 章 其它 DSP 设计库	263	14.3.2 IF 语句模块	276
14.1 总线控制库	263	14.3.3 逻辑位操作符模块	277
14.1.1 AltBus 模块	263	14.3.4 逻辑总线操作符模块	278
14.1.2 输入/输出常数模块	266	14.3.5 LUT 模块	279
14.1.3 小数点确定模块	267	14.3.6 n-to-1 多路选择器模块	279
14.1.4 BusBuild 模块	267	14.4 状态机函数库	280
14.1.5 Bus Concatenation 模块	268	14.4.1 FIFO 控制状态机设计示例	280
14.1.6 Bus Conversion 模块	269	14.4.2 状态机设计流程	282
14.1.7 Extract Bit 模块	270	14.5 Storage 库	288
14.2 复数信号库	270	14.5.1 延迟模块	288
14.2.1 Butterfly 算子模块	270	14.5.2 Down Sampling 和 Up Sampling 模块	289
14.2.2 复数加减模块	271	14.5.3 双口 RAM 模块	290
14.2.3 复数乘积模块	272	14.5.4 并行到串行转换模块与串行到 并行转换模块	291
14.2.4 复数共轭模块	272	14.5.5 数据排列方式模块	293
14.2.5 复数与 Real-Imag 间的连接 模块	273	14.5.6 ROM EAB 模块	293
14.2.6 复数多路选择器模块	274	14.5.7 Shift Taps 模块	294
14.2.7 复数延迟模块	274	14.5.8 PLL 模块	295
14.2.8 复数常数模块	274	附录	297
14.3 Gates 库	275	参考文献	300
14.3.1 Case 语句模块	275		



第1章 概述

不断发展的数字信号处理(DSP, Digital Signal Processing)技术迅速地扩展到了其应用领域,如 3 G 移动通信、网络会议、多媒体系统、雷达卫星系统、医学仪器、实时图像识别与处理、联合战术无线电系统、智能基站,以及民用电器等。所有这一切在功能实现、性能指标与成本方面都在不断增加其要求。

在过去很长一段时间,DSP 处理器(如 TI 的 TMS320 系列)是 DSP 应用系统核心器件的惟一选择。尽管 DSP 处理器具有通过软件设计能适用于实现不同功能的灵活性,但面对当今迅速变化的 DSP 应用市场,特别是面对现代通信技术的发展,DSP 处理器早已显得力不从心。例如其硬件结构的不可变性导致了其总线的不可改变性,而固定的数据总线宽度,已成为 DSP 处理器一个难以突破的瓶颈。DSP 处理器的这种固定的硬件结构特别不适合于当前许多要求能进行结构特性随时变更的应用场合,即所谓面向用户型的 DSP 系统,或者说是用户可定制型,或可重配置型的 DSP 应用系统(Customized DSP 或 Reconfigurable DSP 等),如软件无线电、医用设备、导航、工业控制等方面。至于在满足速度要求方面,由于采用了顺序执行的 CPU 架构,DSP 处理器则更加不堪重负。

面向 DSP 的各类专用 ASIC 芯片虽然可以解决并行性和速度的问题,但是高昂的开发设计费用、耗时的设计周期及不灵活的纯硬件结构,使得 DSP 的 ASIC 解决方案日益失去其实用性。

现代大容量、高速度的 FPGA 的出现,克服了上述方案的诸多不足。在这些 FPGA 中,一般都内嵌有可配置的高速 RAM、PLL、LVDS、LVTTTL 以及硬件乘法累加器等 DSP 模块。用 FPGA 来实现数字信号处理可以很好地解决并行性和速度问题,而且其灵活的可配置特性,使得 FPGA 构成的 DSP 系统非常易于修改、易于测试及硬件升级。

在利用 FPGA 进行 DSP 系统的开发应用上,已有了全新的设计工具和设计流程。DSP Builder 就是 Altera 公司推出的一个面向 DSP 开发的系统级工具。它是作为 MATLAB 的一个 Simulink 工具箱(ToolBox)出现的。MATLAB 是功能强大的数学分析工具,广泛应用于科学计算和工程计算,可以进行复杂的数字信号处理系统的建模、参数估计、性能分析。Simulink 是 MATLAB 的一个组成部分,用于图形化建模仿真。DSP Builder 作为 Simulink 中的一个工具箱,使得用 FPGA 设计 DSP 系统完全可以通过 Simulink 的图形化界面进行,只要简单地进行 DSP Builder 工具箱中的模块调用即可。值得注意的是,DSP Builder 中的 DSP 基本模块是以算法级的描述出现的,易于用户从系统或者算法级进行理解,甚至不需要了解 FPGA 本身和硬件描述语言。

1.1 DSP 实现方案及设计流程

为了满足 DSP 技术领域中的各种需求以及顺应 DSP 市场的发展,DSP 应用系统的实现



方式和目标器件的品种类型、结构特点乃至开发技术本身都经历着不断的改善和变革。

1.1.1 常用 DSP 应用器件及其性能特点

如前所述, DSP 作为数字信号的算法的实现方案有多种, 对于不同的应用领域、适用范围和指标要求, 可以选用不同的解决方案和 DSP 系统的实现器件。目前, 为了完成 DSP 的开发与应用, 可选的目标器件有如下 4 类:

- DSP Processor DSP 处理器;
- ASICs(Application-Specific Integrated Circuits) 专用集成电路;
- ASSPs(Application-Specific Standard Proucts) 专用标准电路模块;
- FPGA。

DSP Processor 主要是指目前最常用的基于 CPU 架构的器件, 通过软件指令的方式完成 DSP 算法。早期的 DSP 处理器(如大多数 DSP 实验室中采用的 TMS320VC5402)中只有一个乘法器, 而现在的处理器(如 C6000 系列)中, 有的已达 8 个乘法器, 工作性能有了很大的提高。DSP Processor 在硬件结构上的不断改进, 并没有摆脱传统 CPU 的工作模式。因而, 尽管拥有多个硬件乘加器, 使用了环形叠代的方法进行乘法操作, 且许多 DSP 处理器还拥有使用多乘法器的并行指令, 用于加速算术运算, 然而由于其顺序的工作方式、较低的数据处理速率, 以及缺乏实时工作的性能, 使其至今仍只适合于低端的数字信号处理。

传统的 DSP 开发者通常都选用 DSP 处理器来构成 DSP 应用系统。DSP 处理器的优势主要是具有很好的通用性和灵活性, 有适用于各种 DSP 算法实现的通用硬件结构。图 1-1 所示的是一种比较典型的 DSP 系统电路图, 除了选用的是 DSP 处理器以及应用程序加载工作方式外, 与普通单片机应用系统十分相似, 只要将调试好的机器码放在程序 ROM 中, 就能使系统正常工作。因此通过这种硬件结构, 能对实现各种数据处理的程序进行有效的执

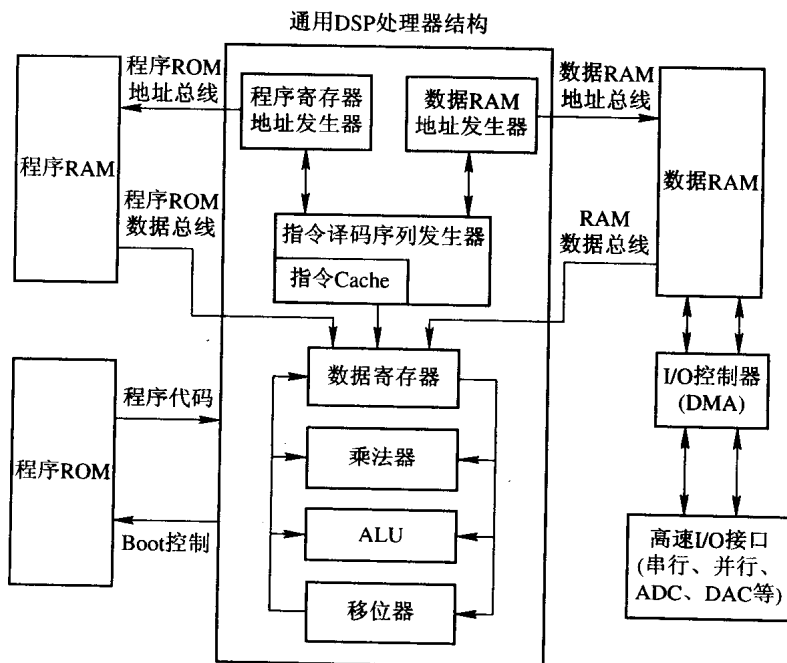


图 1-1 基于通用 DSP 处理器的 DSP 系统



行。由此可知,这种灵活性几乎是没有限制的,因为,只要能对应指令程序表达出 DSP 算法,其硬件结构都能接受并执行。

然而,各种算法可实现的灵活性与各种技术指标的可实现性,以及硬件结构的可变性完全是两回事。

ASSP 和 ASIC 是专门针对完成某种 DSP 算法的集成电路器件,因此在性能指标、工作速度、可靠性和应用成本上优于 DSP 处理器。如卷积相关器 IMSA100、FFT 处理器 A41102、复乘加器组 PDSP16116、求模/相角器 PDSP16330、下变频/信号解调器 HSP50214 等等。其优秀的工作性能主要源于特定的算法全部由 ASSP 和 ASIC 中的硬件电路完成。ASSP 是半定制集成电路,在许多 DSP 算法的实现方面(如 FIR, IIR 滤波器)都优于 DSP 处理器,但在功能重构,以及应用性修正方面缺乏灵活性;ASIC 虽然有一定的可定制性,但开发周期长,而且有一个最小定制量,因此应用风险和开发成本过高,正在逐渐失去其实用性。

但若直接使用 FPGA 完成 DSP 功能,则能在许多实用领域综合 DSP 处理器与 ASIC/ASSP 器件的优点,再加上 FPGA 本身的诸多优势,即能有效克服传统 DSP 系统的弱点。

由 FPGA 构成的 DSP 电路可以同样以并行或顺序方式工作。如图 1-2 所示,在并行工作方面, FPGA 与 ASIC/ASSP 相当,远优于 DSP 处理器。对 DSP 处理器需要大量运算指令完成的工作, FPGA 只需一个时钟周期的时间就能完成。而在顺序执行方面, FPGA 也比 DSP 处理器快,因为 FPGA 中可以使用各种状态机,或使用嵌入式微处理器来完成,并且,每一顺序工作的时钟周期中都能同时并行完成许多执行,而 DSP 处理器却不能。就灵活性而言, FPGA 的灵活性远胜于 ASIC/ASSP,也胜于 DSP 处理器。

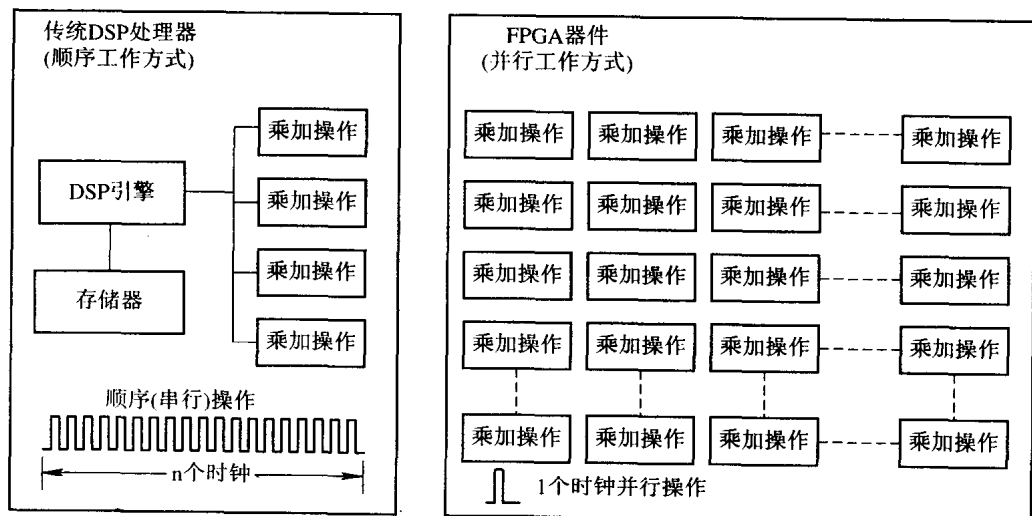


图 1-2 DSP 处理器顺序工作方式与 FPGA 的并行工作方式

DSP 处理器的灵活性主要表现在软件更改的容易性以及对各种算法的处理和复杂算法的实现上,而对于系统硬件本身的更改, DSP 是没有任何灵活性可言的。

现代大容量 FPGA 以其相关的开发技术在可重配置的 DSP 应用领域,以及 DSP 数据大吞吐量 and 数据的纯硬件处理方面,有独特的优势。FPGA 面对传统的 DSP 处理器诸多难以



克服的技术瓶颈,已有了突破性的应用。在各种 DSP 应用场合, FPGA 具有全硬件的用户可定制性以及重配置性,即可根据需要随时通过改变 FPGA 中构成 DSP 系统的硬件结构来改变系统的功能、技术指标、通信方式、硬件加密算法、编解码方式等等。这种结构变化可以由 DSP 开发者在开发时完成,也能在 DSP 系统投入实用中随时“在系统”更改,这就是所谓的可重配置特性。所以,基于 FPGA 实现的 DSP 系统,具有完全的硬件结构可定制性,包括总线结构的可定制性,存储器的可定制性,硬件加速器模块的结构与数量的可定制性,以及大量的 MAC(乘加器)模块的可选性(系统设计中的硬件结构可变性称为可定制性,即 Customized;系统设计完成后,成为产品后仍能随开发者甚至用户的要求随时进行硬件结构重构的,称为可重配置性,即 Reconfigurable)。

1.1.2 DSP 处理器结构与性能的发展

DSP 处理器与普通处理器的基本差异是 DSP 处理器中有硬件乘加模块(MAC),专用的存储器以及适用于高速数据运行的总线结构。

MAC 的工作性能通常是 DSP 处理器性能的关键所在。DSP 应用系统主要是完成一些诸如 FFT、FIR、IIR、矩阵相乘、卷积等算法。大部分的算法涉及到乘加结构的数学计算(如 $y = a \times b + c \times d$),这称为乘法累加操作(MAC)。为了提高 DSP 的性能, DSP 处理器生产厂商在提高 MAC 模块硬件性能的同时增加它们的数量,以提高乘加的位宽与速度。例如, TI 的 TM320C6411 中有 8 个 MAC,能在一个时钟周期中完成 8 个乘法运算。

尽管在 DSP 处理器中增加更多的 MAC 单元,能进一步提高 DSP 的吞吐量,但对于某些一般数据处理量比较大的 DSP 算法,其通用性能却下降了,例如 Viterbi 编译码器和 FIR 滤波器就属于这种情况。为了解决这个问题, DSP 生产厂商又将一些专用的硬件加速器类的协处理器模块加入进处理器结构。如 Viterbi 协处理器、Turbo 协处理器和增强型滤波器协处理器等。由于这些处理器本身的非通用性,即主要是倾向于某种或某些算法的协处理器,从而导致了 DSP 处理器无法适用于多种类型的 DSP 算法和 DSP 应用。结果成本增加了,通用性下降了。因此,大部分的 DSP 应用场合并没有从这种增加了各种硬件加速器的 DSP 结构的变化中得到好处。不但如此,这种硬件加速器模块是一种固定的硬件结构,无法根据特定的设计需要来作任何更改,特别是面向当今通信领域中不断发生的各种技术标准和协议的变更,这种加速器模块容易很快过时,从而导致整个 DSP 应用系统的过时。

针对这种情况, DSP 处理器生产商又试图在 DSP 中增加一些可定制的指令,以便能对结构功能根据要求作一定的改变。然而这些修补并没有从根本上解决问题。

为了弥补速度及吞吐量的不足,实用的 DSP 系统使用多片 DSP 处理器进行并行工作已经是十分普遍的事,但这也同时带来了系统在灵活性、功耗、成本、体积、可靠性等技术性能上的进一步变差。

1.1.3 FPGA 的结构与性能的发展

FPGA 器件是由大量逻辑宏单元构成的。通过配置,可以使这些逻辑宏单元形成不同的硬件结构,从而构成不同的电子系统,完成不同的功能。正是 FPGA 的这种硬件重构的灵活性,使得设计者能够将硬件描述语言(如 VHDL 或 Verilog)描述的电路在 FPGA 中实现。这样以来,同一块 FPGA 能实现许多完全不同的电路结构和功能。如 DSL 布线器、数字调



制解调器、JPEG 编码器、数字通信系统，以及网络接口等等。

随着达数百万门高密度的 FPGA 的出现，FPGA 在原有的高密度的逻辑宏单元的基础上嵌入了许多面向 DSP 的专用硬核模块，结合大量可配置于 FPGA 硬件结构中的参数化的 DSP IP 软核，DSP 开发者能十分容易地将整个 DSP 应用系统实现在一片 FPGA 中，从而实现了所谓的可编程 SOC 系统，即 SOPC。

FPGA 中的面向 DSP 的嵌入式模块有可配置 RAM、DSP 乘加模块和嵌入式处理器等，使 FPGA 能很好地适用于 DSP 功能的实现。例如 Altera 的 Stratix 系列器件中含有丰富的 DSP 功能块，包括能完成诸如累加、加/减、并行加等通用算法的功能模块，有多达 224 个乘法器，在一个时钟周期中就能完成 224 次乘法操作。

乘法器的带宽决定了整个 DSP 的带宽，而 FPGA 的 DSP 带宽比 DSP 处理器要宽得多。例如，Stratix 器件的 DSP 模块能提供 70GMACs/每 DSP 模块的吞吐量，而当今主流的 DSP 处理器只能达到 4.8 GMACs。

此外，绝大部分的 DSP 处理器应用系统是用外部存储器来解决大数据量的处理的。然而 FPGA 的嵌入式高速可配置存储器在大多数情况下都能满足相类似的数据处理要求。例如，Stratix 系列的 FPGA 可提供达 10 Mb/s 速率的嵌入式 RAM。

FPGA 中的嵌入式处理器进一步提高了 FPGA 的系统集成和灵活性，使之成为一个软件与硬件联合开发和灵活定制的结合体，可使设计者既能在嵌入式处理器中完成系统软件模块的开发和利用，也能利用 FPGA 的通用逻辑宏单元完成硬件功能模块的开发。Altera 的 FPGA 器件还为用户提供了嵌入式处理器软核与硬核的选择。

嵌入式处理器软核是由网表文件表达的硬件结构，当同其它设计一同配置于 FPGA 中后，就成为 FPGA 芯片中的一个硬处理器核。高效率的 SOPC 设计能很容易地将软核(如 Nios 嵌入式处理器)连同与该核相关的外围接口系统一同编程下载进同一片 FPGA 中。设计者能根据实际应用的需要定制 Nios 软核，使之满足不同的总线数量、总线宽度和总线功能要求，优化总线设计，排除传统 DSP 中许多常见的问题。还能定制处理器的指令，此时 Nios 的开发软件编译器将自动作相应的变化。这种结构可变(可定制)的 DSP 处理器能很好地根据实际的需求，在性能/功能与结构大小之间作最好的平衡，以适用于不同的 DSP 应用领域。

硬核处理器主要指在 Excalibur 系列 FPGA 中的 ARM 核。这种核已预先嵌入在 FPGA 中，含有完整的外围接口系统，如 SDRAM、存储器控制单元、UART 等。

现代的 FPGA 中含有十分灵活的、针对特定算法的加速器模块。与传统的 DSP 处理器中的加速器模块不同，FPGA 中实现的硬件加速器是可以针对不同应用的，这可以使设计者针对不同的 DSP 任务实现硬件功能。设计者针对具体任务在 FPGA 中实现硬件加速器模块的途径很多，主要有下述几种：

- 用硬件描述语言 HDL 完成；
- 基于通用逻辑宏单元 LCs 的 HDL 设计；
- 基于可配置的 DSP 硬核模块：存储器、乘法器、并行加法器、累加器等；
- 基于全参数可设置的 DSP 软 IP 核的应用(进入 FPGA 中后其将成为性能优良的硬件模块)；
- Nios 软核处理器；
- ARM 硬核处理器。



在基于 FPGA 的 DSP 开发中,面向 DSP 的 IP 核的应用是最方便的设计方案。由 Altera 与其 AMPP 合作伙伴(Altera Megafunction Partner Program)提供针对 DSP 设计的各种用于硬件加速的 IP 核,大致可分为以下 5 类:

- (1) 通用 IP 核(如 FIR, IIR, NCO, FFT 等);
- (2) 图像处理 IP 核(如 JPEG, DCT 等);
- (3) 调制 IP 核(如 QPSK, Equalizer 等);
- (4) 编码 IP 核(如 DES, Rijndael 等);
- (5) 纠错 IP 核(如 Viterbi, Turbo, CRC 等)。

以上的每一个核都可以利用 Quartus II 中的“MegaWizard Plug-In Manager”管理器进行参数设置以构成针对特定应用的硬件功能模块。这种通过软件设置能随意改变专用硬件模块功能的技术,极大地提高了 FPGA 在 DSP 设计方面的灵活性。IP 核的利用,可以使设计者将 IP 核加入到任何标准硬件描述语言中,完成特定的功能而不改变原来的设计程序;即使在设计中和设计完成后,都能根据实际需要改变嵌入的 IP 核的技术参数,而不改变整体设计综合得来的代码,从而改变 DSP 系统的技术指标和硬件功能。此外,IP 核本身基本不依赖于某种特定的 FPGA 硬件结构,即具有硬件通用性(这一点与 DSP 处理器及其库程序有很大不同),因此可以在更新、更廉价的 FPGA 中实现。

显然, FPGA 与软 IP 核的巨大灵活性能十分迅速地对即使是已经完成的设计系统,仍然能适应通信领域中不同的,诸如 Wireless802.11a, Wireless Broadband Working Group 802.16 及 HiperLAN12 等的新标准的要求,而绝对不必如使用传统 DSP 处理器那样,要等待很长时间才能有适用的器件出现,或是在开发前期必须针对某种通信的标准,而费时费力地确定某款合适的 DSP 处理器型号。

1.1.4 基于 DSP 处理器的 DSP 设计流程

图 1-3 是传统的 DSP 系统的典型开发流程。大致步骤如下:

(1) DSP 开发者首先使用诸如 MATLAB 这样的数学开发工具对 DSP 算法进行优化设计和仿真测试,或用 Simulink 进行 DSP 系统建模,以获得满足功能要求和适应硬件特点的算法模型。

尽管现在已有可将 Simulink 的 DSP 模型直接转换成 DSP 处理器的 C 程序,但由于器件通用性问题,特别由于 C 程序的运行效率问题,通过 MATLAB 途径获得的 C 程序几乎没有实用价值。

(2) 根据 DSP 目标系统的功能要求、技术指标、系统升级可行性、性能标准可能的变化以及成本限度等因素,具体确定 DSP 处理器(并行使用的)数量和型号。这时必须十分熟悉当前主流 DSP 器件的详细硬件特性与价格范围,同时还要与手头的 DSP 开发软硬件工具和功能块程序库结合起来考虑。由于目前 DSP 处理器的厂商、品种、系列、功能、适用面以及价格等因素差异颇大,准确无误地确定 DSP 处理器的品种不但十分棘手,同时也事关重大。因为如果当系统最终设计并调试完成后,若发现由于 DSP 处理器导致的某项技术指标不能达到,或在以后的某项技术规范的升级要求中无法实现,将出现整个硬件系统必须重新开发的严重局面。