

新编电气与电子信息类本科规划教材

TMS320C55x DSP

原理及应用(第二版)

汪春梅 孙洪波 编著 邹俊忠 审校



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

新编电气与电子信息类本科规划教材

TMS320C55x DSP 原理及应用

(第二版)

汪春梅 孙洪波 编著

邹俊忠 审校

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以 TMS320C55x DSP 为重点,全面介绍了 TMS320C5000 DSP 系统设计与开发。首先详细介绍了数字信号处理器的硬件结构、汇编指令、存储空间结构和寻址方式;其次介绍了 C55x 处理器中 C 语言与汇编语言的混合编程方法;然后从应用的角度,结合片内集成外设的结构和功能,给出了片内集成外设驱动程序开发方法及部分片内外设的调试方法;最后结合 DSP 软/硬件的设计,给出了详细的设计方法和丰富的应用实例。本书还对 DSP 的集成开发环境 CCS 的使用方法进行了详尽的描述。

本书旨在从应用的角度使读者了解 TMS320C55x DSP 的体系结构和基本原理,熟悉 DSP 芯片的开发工具和使用方法,掌握 DSP 系统设计和软/硬件开发。

本书内容丰富、新颖、实用性强,适合从事数字信号处理的科技人员和高校师生阅读。

(附二 策)

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。

基础 赵书怀 孙春玉

图书在版编目(CIP)数据

孙审 忠 孙雅

TMS320C55x DSP 原理及应用/汪春梅,孙洪波编著. —2 版. —北京:电子工业出版社,2008.7

新编电气与电子信息类本科规划教材

ISBN 978-7-121-07027-3

I. T… II. ①汪… ②孙 III. 数字信号—信息处理系统—系统设计—高等学校—教材 IV. TN911.72

中国版本图书馆 CIP 数据核字(2008)第 097715 号

策划编辑:陈晓莉

责任编辑:陈晓莉

印 刷:北京东光印刷厂

装 订:三河市皇庄路通装订厂

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:787×1 092 1/16 印张:24 字数:618 千字

印 次:2008 年 7 月第 1 次印刷

印 数:4 000 册 定价:38.00 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。

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

服务热线:(010) 88258888。

前 言

历经三十年的发展,数字信号处理器(DSP)的应用范围已经遍及军用电子、消费电子、工业控制等重要领域,各种崭新的应用层出不穷,这些应用对DSP的处理能力、功耗、体积以及开发的方便程度都提出了较高要求。而第三代DSP则以其强大的数字信号处理能力、超低功耗和适合手持设备的超小型封装等特点,较好地满足了新一代电子产品的要求,以CCS为代表的集成开发环境为应用者提供了方便、快捷的DSP开发手段。

C55x系列处理器在兼容C54x系列处理器指令集的基础上,将处理速度提高到200MHz~300MHz,而对DSP内核又进行了重大改进,将强大的处理能力和超低功耗完美结合,成为新一代数字信号处理器的典型代表。正是因为C55x系列处理器具有这些特点,所以特别适合嵌入式数字信号处理方面的应用。我们从2003年起将所使用的平台由C54x系列处理器转换为C55x系列处理器,先后在软件无线电、超声波探测等多个项目中应用C55x处理器,取得了较好的效果。作者在2004年编写了《TMS320C5000系列DSP系统设计与开发实例》(第一版),对应用心得和成果作了阶段性的总结,经过几年的积累,在汲取读者反馈意见的基础上,我们投入到了《TMS320C55x DSP原理及应用》(第二版)的编写当中。

为了适应初学者的需要,《TMS320C55x DSP原理及应用》(第二版)在内容上注意了由浅入深、图文并茂,全面系统地展开论述,在每章之后附上习题,方便读者理解和复习本章的内容;本书还对C55x芯片支持库、数字信号处理库和图像、视频处理库进行了介绍,并在附录中对数字信号处理库和图像、视频处理库的函数进行了详细说明;本书增加了大量的软硬件应用实例,并将篇幅扩展为两章。在书中介绍了大量实用DSP系统,能够满足DSP人员之所需,书中所列出的大量典型的应用实例,可在实际开发中直接引用,相信能够给开发者带来一些有益的帮助。

本书第1章结合德州仪器公司的DSP产品对DSP技术的发展概况进行了介绍,读者可以根据本章内容和需求选取适合的DSP芯片;第2章重点介绍了TMS320VC55x处理器的硬件结构,并以TMS320VC5509为例介绍了C55x处理器的主要特性和功能,最后介绍了DSP的存储空间结构;第3章在介绍数字信号处理和DSP系统的基础上,详细介绍了C55x处理器的汇编指令和寻址方式;第4章介绍了C55x处理器的程序基本结构,C语言编程以及优化,C与汇编语言的混合编程,通用目标文件格式,最后对C55x处理器的数字信号处理库和图像、视频处理库进行了介绍;第5章介绍了TMS320C55x系列片内集成外设的结构、功能,以及芯片支持库的使用和驱动程序的开发,并给出了部分片内外设的测试过程;第6章介绍了DSP的集成开发环境CCS的原理和使用;第7、8章则详细介绍了DSP软、硬件的设计方法和应用实例。本书还附有光盘,内容包括教学课件和软件实例。

本书由汪春梅策划,汪春梅编写了其中第2、3、6、8章,第1、4、5、7章由孙洪波编写,全书由邹俊忠审校。在编写过程中,叶宏、倪继锋、贾红涛、朱品昌、王学刚等给予了支持和帮助,北京瑞泰创新科技有限公司的李心阳工程师提供了部分技术资料,电子工业出版社给予了极大的鼓励和支持。作者在此一并致谢。

本书既可以作为大中专院校的选修课教材,也可作为电子工程技术人员的参考书籍。

希望此书对读者有所裨益,也希望有助于 DSP 技术的推广。限于作者的水平,书中的错误在所难免,恳请读者不吝赐教!

汪春梅

2008 年 6 月于上海

2008年6月于上海

目 录

第 1 章 数字信号处理和 DSP 系统	1
1.1 实时数字信号处理技术的发展	1
1.2 数字信号处理器的特点	4
1.2.1 存储器结构	4
1.2.2 流水线	5
1.2.3 硬件乘法累加单元	5
1.2.4 零开销循环	5
1.2.5 特殊的寻址方式	6
1.2.6 高效的特殊指令	6
1.2.7 丰富的片内外设	6
1.3 德州仪器公司的 DSP 产品	7
1.3.1 C24x 系列和 C28x 系列	7
1.3.2 C62x 系列和 C64x 系列	12
1.3.3 C67x 系列和 C33	15
1.3.4 C54x 系列和 C55x 系列	16
1.4 DSP 芯片的选择	19
1.4.1 运算速度	19
1.4.2 算法格式和数据宽度	19
1.4.3 存储器	20
1.4.4 功耗	20
1.4.5 开发工具	20
1.5 DSP 应用系统设计流程	21
思考与练习题	22
第 2 章 TMS320C55x 的硬件结构	23
2.1 TMS320C55x DSP 的基本结构	23
2.1.1 C55x 的 CPU 体系结构	24
2.1.2 指令缓冲单元 (I)	25
2.1.3 程序流程单元 (P)	25
2.1.4 地址流程单元 (A)	26
2.1.5 数据计算单元 (D)	27
2.1.6 指令流水线	28
2.2 TMS320VC5509A 的主要特性	30

目 录

2.2.1 VC5509A 的主要特性	30
2.2.2 VC5509A 的引脚功能	31
2.3 TMS320C55x 存储空间结构	35
2.3.1 存储器映射	35
2.3.2 程序空间	36
2.3.3 数据空间	37
2.3.4 I/O 空间	38
思考与练习题	39
第 3 章 TMS320C55x 的指令系统	40
3.1 寻址方式	40
3.1.1 绝对寻址模式	40
3.1.2 直接寻址模式	41
3.1.3 间接寻址模式	43
3.2 TMS320C55x 的指令系统	47
3.2.1 C55x 指令的并行执行	47
3.2.2 TMS320C55x DSP 的汇编指令	50
思考与练习题	92
第 4 章 C55x 处理器的软件设计	93
4.1 C55x 处理器程序基本结构	93
4.1.1 自我调度程序的基本结构	94
4.1.2 应用嵌入式操作系统	95
4.2 C 语言程序开发及优化	98
4.2.1 C 语言中的数据类型	98
4.2.2 对 I/O 空间进行寻址	98
4.2.3 interrupt 关键字	100
4.2.4 onchip 关键字	101
4.2.5 C 语言的优化	101
4.3 C 语言与汇编语言的混合编程	102
4.3.1 在 C 语言中直接嵌套汇编语句	102
4.3.2 C 语言调用汇编模块的接口	103
4.4 通用目标文件格式	108
4.4.1 C/C++ 和汇编语言中段的分配	109
4.4.2 寄存器模式设置	110
4.4.3 链接命令文件	111
4.5 C55x 处理器的数字信号处理库和图像、视频处理库	113
4.5.1 C55x 的数字信号处理库	113
4.5.2 C55x 的图像、视频处理库	115
思考与练习题	116

第 5 章 TMS320C55x 的片内集成外设开发及测试	118
5.1 C55x 片内外设与芯片支持库简介	118
5.2 时钟发生器	119
5.2.1 时钟模式寄存器	119
5.2.2 工作模式	120
5.2.3 CLKOUT 输出	120
5.2.4 使用方法	121
5.2.5 使用方法及实例	121
5.2.6 时钟发生器的调试	122
5.3 通用定时器	122
5.3.1 结构框图	122
5.3.2 工作原理	123
5.3.3 使用方法	124
5.3.4 通用定时器的应用	125
5.3.5 通用定时器的调试	126
5.4 外部存储器接口	126
5.4.1 功能与作用	126
5.4.2 外部寄存器接口硬件连接与配置	128
5.4.3 外部寄存器接口的软件设置	135
5.5 主机接口 (EHPI)	137
5.5.1 EHPI 接口的非复用连接方式	138
5.5.2 EHPI 接口的复用连接方式	139
5.5.3 EHPI 口的寄存器	139
5.6 多通道缓冲串口 McBSP	140
5.6.1 概述	140
5.6.2 组成框图	141
5.6.3 采样率发生器	142
5.6.4 多通道选择	143
5.6.5 异常处理	144
5.6.6 McBSP 寄存器	145
5.6.7 多通道缓冲串口的应用	149
5.6.8 McBSP 串口的测试	152
5.7 通用输入/输出端口 GPIO	153
5.7.1 GPIO 口概述	153
5.7.2 上电模式设定	154
5.7.3 驱动程序开发	154
5.7.4 通用输入/输出 GPIO 的测试	155
5.8 DMA 控制器	155
5.8.1 概述	155
5.8.2 通道和端口	156

5.8.3	HPI 的配置	156
5.8.4	DMA 传输配置	157
5.8.5	DMA 控制器的寄存器	158
5.8.6	使用方法及实例	162
5.9	I ² C 总线	163
5.9.1	I ² C 总线简介	163
5.9.2	I ² C 寄存器	165
5.9.3	I ² C 模块的使用	166
5.10	多媒体卡控制器	167
5.10.1	多媒体卡控制器简介	167
5.10.2	MMC/SD 模式	168
5.10.3	SPI 模式	171
5.10.4	MMC 控制器的应用	173
5.11	通用串行总线 (USB)	174
5.11.1	通用串行总线 (USB) 简介	174
5.11.2	USB 的 DMA 控制器	174
5.11.3	USB 模块的寄存器	177
5.11.4	USB 模块的应用	182
5.12	模/数转换器 (ADC)	183
5.12.1	模/数转换器结构和时序	183
5.12.2	模/数转换器的寄存器	184
5.12.3	使用方法及实例	186
5.13	实时时钟 (RTC)	187
5.13.1	实时时钟基本结构	187
5.13.2	实时时钟的内部寄存器	188
5.13.3	实时时钟的应用	191
5.14	看门狗定时器 (Watchdog)	192
5.14.1	工作方式	192
5.14.2	寄存器说明	193
5.14.3	看门狗定时器的应用	195
5.15	异步串口 (UART)	195
5.15.1	异步串口基本结构	195
5.15.2	异步串口寄存器	200
5.15.3	异步串口的应用	206
	思考与练习题	207
第 6 章 DSP 集成开发环境 CCS		208
6.1	CCS 功能简介	208
6.2	CCS 安装与设置	209
6.2.1	CCS 软件安装	209

6.2.2	CCS 设置	209
6.3	CCS 的基本操作	212
6.3.1	CCS 的窗口和工具条	213
6.3.2	文件的编辑	217
6.3.3	反汇编窗口的使用	220
6.3.4	存储器窗口的使用	221
6.3.5	寄存器窗口的使用	222
6.3.6	观察窗口的使用	223
6.4	创建工程项目	225
6.4.1	工程项目的建立、打开和关闭	225
6.4.2	向工程项目添加各类文件	226
6.4.3	工程项目的构建	227
6.5	利用 CCS 调试程序	227
6.5.1	装载并运行程序	228
6.5.2	断点的使用	230
6.5.3	探针点的使用	233
6.5.4	图形工具的使用	234
6.6	程序分析工具的使用	241
6.7	GEL 工具的使用	243
6.8	CCS 开发 DSP 软件使用举例	247
	思考与练习题	251
第 7 章	TMS320C55x 硬件设计实例	252
7.1	DSP 最小系统设计	252
7.1.1	C55x 的电源设计	252
7.1.2	复位电路设计	253
7.1.3	时钟电路设计	254
7.1.4	JTAG 接口电路设计	254
7.1.5	程序加载部分	254
7.2	A/D 与 D/A 设计	262
7.2.1	串行多路 A/D 设计	262
7.2.2	高速并行 A/D 设计	266
7.2.3	并行 D/A 设计	269
7.3	C55x 在语音系统中的应用	270
7.4	手写系统的实现	274
7.5	C55x 在软件无线电中的应用	276
	思考与练习题	283

第 8 章 TMS320C55x 软件设计实例	284
8.1 卷积算法	284
8.1.1 卷积算法	284
8.1.2 卷积算法的 MATLAB 实现	284
8.1.3 卷积算法的 DSP 实现	285
8.2 有限冲激响应滤波器 (FIR) 的实现	290
8.2.1 有限冲激响应滤波器的特点和结构	290
8.2.2 有限冲激响应滤波器的 MATLAB 设计	290
8.2.3 有限冲激响应滤波器 DSP 实现	292
8.3 无限冲激响应滤波器 (IIR) 的实现	293
8.3.1 无限冲激响应滤波器的结构	293
8.3.2 无限冲激响应滤波器的 MATLAB 设计	293
8.3.3 无限冲激响应滤波器的 DSP 实现	294
8.4 快速傅里叶变换 (FFT)	298
8.4.1 快速傅里叶变换 (FFT) 算法	298
8.4.2 快速傅里叶变换 (FFT) 的 DSP 实现	299
8.5 语音信号编解码 (G.711)	303
8.5.1 语音信号编解码原理	303
8.5.2 语音信号编解码的 DSP 实现	305
8.6 数字图像的锐化	307
8.7 Viterbi 译码	309
思考与练习题	324
附录 A 寄存器	325
附录 B VC5510 中断	334
附录 C TMS320C55x 的状态寄存器	336
附录 D TMS320C55x 的汇编指令集	339
附录 E TMS320C55xDSP 库函数	352
附录 F VC5509 系统部分原理图	358
习题答案	363
参考文献	374

第 1 章 数字信号处理和 DSP 系统

当德州仪器 (TI) 公司于 1982 年推出第一款商用数字信号处理器时, 谁也不会想到它竟能给世界带来如此大的变化。从移动通信到消费电子领域, 从汽车电子到医疗仪器, 从自动控制到军用电子系统中都可以发现它的身影。刚诞生的第一代数字信号处理器仅包含了 55000 个晶体管, 4KB 内存, 指令处理能力只有 5MIPS (每秒百万条), 经过 20 余年的发展, 单核数字信号处理器的处理能力已经达到 9600MIPS 的惊人速度, 寻址能力高达 1280MB。数字信号处理器给世界带来了巨大的变化, 未来可能的应用包括无人驾驶汽车、精确化的楼宇照明控制、自动识别并报警的安防系统等, 有无数令人兴奋的应用在等待我们去开拓。那么就让我们进入这个充满变化、充满挑战, 而又拥有无限精彩的 DSP (数字信号处理器) 世界吧。

1.1 实时数字信号处理技术的发展

20 世纪 60 年代以来, 随着信息技术的不断进步, 数字信号处理技术应运而生并得到迅速发展。80 年代以前, 由于方法的限制, 数字信号处理技术处于理论研究阶段, 还得不到广泛的应用。在此阶段, 人们利用通用计算机进行数字滤波、频谱分析等算法的研究, 以及数字信号处理系统的模拟和仿真。而将数字信号处理技术推向高峰的则是实时数字信号处理技术的高速发展。

实时数字信号处理对数字信号处理系统的处理能力提出了严格的要求, 所有运算、处理都必须小于系统可接受的最大时延。以视频会议为例, 从发送端图像、声音信号的采集、压缩, 通过信道传输, 到接收端完成数据接收, 图像、声音信号的解压、还原, 其中任何一个处理环节都应满足最大时延要求, 否则将出现图像、声音信号的间断, 从而影响视频会议的正常进行, 如果每个数据包都包含了 20ms 的音、视频信号, 可以很容易得出整个系统的整体延迟必须小于 500ms, 而每个数据包的处理时间必须小于 20ms 才能满足系统实时处理要求。

典型实时数字信号处理系统的基本部件包括: 抗混叠滤波器 (Anti-aliasing filter)、模/数转换器 (Analog-to-Digital Converter, ADC)、数字信号处理、数/模转换器 (Digital-to-Analog Converter, DAC) 和抗镜像滤波器 (Anti-image filter), 如图 1-1 所示。其中, 抗混叠滤波器将输入的模拟信号中高于奈奎斯特频率的频率成分滤掉; ADC 将模拟信号转换成 DSP 可以处理的并行或串行的数字比特流; 数字信号处理部分完成数字信号处理算法; 经过处理的数字信号经 DAC 转换成为模拟样值之后, 再由抗镜像滤波器完成模拟波形的重建。

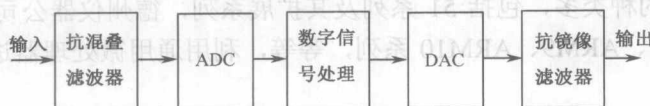


图 1-1 典型实时数字信号处理系统框图

和其他数字系统一样,实时数字信号处理系统具有许多模拟系统不具备的优点,如灵活,可编程,支持时分复用,易于模块化设计,可重复使用,可靠性高,抗环境干扰能力强,易于维护等。当前实用的数字信号处理系统主要有以下几种,它们各具优缺点,这就需要使用者根据具体情况做出相应选择。

1. 利用 X86 处理器完成实时数字信号处理

随着 CPU 技术的不断进步, X86 处理器的处理能力不断发展,基于 X86 处理器的处理系统已经不仅局限于以往的模拟和仿真,也能满足部分数字信号的实时处理要求,而各种便携式或工业标准的推出,如 PC104、PC104 Plus 结构,以及 CPCI 总线标准的应用,这些都改善了 X86 系统的抗恶劣环境的性能,扩展了 X86 系统的应用范围。利用 X86 系统进行实时数字信号处理有下列优点:

(1) 处理器选择范围较宽: X86 处理器涵盖了从 386 到奔腾系列,处理速度从 100MHz 到几 GHz,而为了满足工控等各种应用, X86 厂商也推出了多款低功耗处理器,其功耗远远小于商用处理器。

(2) 主板及外设资源丰富:无论是普通结构,还是基于 PC104 和 PC104 Plus 结构,以及 CPCI 总线标准,都有多种主板及扩展子板可供选择,节省了用户的大量硬件开发时间。

(3) 有多种操作系统可供选择:这些操作系统包括 Windows、Linux、Win CE 等,而针对特殊应用,还可根据需要对操作系统进行剪裁,以适应实时数字信号处理要求。

(4) 开发、调试较为方便: X86 的开发、调试工具十分成熟,使用者不需要很深的硬件基础,只要能够熟练使用 VC、C-Build 等开发工具即可进行开发。

但使用 X86 进行实时信号处理的缺点也是十分明显的。

(1) 数字信号处理能力不强: X86 系列处理器没有为数字信号处理提供专用乘法器等资源,寻址方式也没有为数字信号处理进行优化,实时信号处理对中断的响应延迟时间要求十分严格,通用操作系统并不能满足这一要求;

(2) 硬件组成较为复杂:即使是采用最小系统, X86 数字信号处理系统也要包括主板(包括 CPU、总线控制、内存等)、非易失存储器(硬盘或电子硬盘、SD 卡或 CF 卡)和信号输入/输出部分(这部分通常为 AD 扩展卡和 DA 扩展卡),如果再包括显示、键盘等设备,系统将更为复杂;

(3) 系统体积、重量较大,功耗较高:即使采用紧凑的 PC104 结构,其尺寸也达到 96mm×90mm,而采用各种降低功耗的措施, X86 主板的峰值功耗仍不小于 5W,高功耗则对供电提出较高要求,则需要便携系统提供容量较大的电池,进一步增大了系统的重量;

(4) 抗环境影响能力较弱:便携系统往往要工作于自然环境当中,温度、湿度、振动、电磁干扰等都会给系统正常工作带来影响,而为了克服这些影响, X86 系统所需付出的代价将是十分巨大的。

2. 利用通用微处理器完成实时数字信号处理

通用微处理器的种类多,包括 51 系列及其扩展系列,德州仪器公司的 MSP430 系列, ARM 公司的 ARM7、ARM9、ARM10 系列,等等,利用通用微处理器进行信号处理的优点如下。

(1) 可选范围广:通用微处理器种类多,使用者可从速度、片内存储器容量、片内外设资源等各种角度进行选择,许多处理器还为执行数字信号处理专门提供了乘法器等资源。

(2) 硬件组成简单: 只需要非易失存储器, A/D、D/A 即可组成最小系统, 这类处理器一般都包括各种串行、并行接口, 可以方便地与各种 A/D、D/A 转换器进行连接。

(3) 系统功耗低, 适应环境能力强。

利用通用微处理器进行信号处理的缺点如下。

(1) 信号处理的效率较低: 以一个两个数值乘法为例, 处理器需要先用两条指令从存储器当中取值到寄存器中, 用一条指令完成两个寄存器的值相乘, 再用一条指令将结果存到存储器中, 这样, 完成一次乘法就花费了 4 条指令, 使信号处理的效率难以提高。

(2) 内部 DMA 通道较少: 数字信号处理需要对大量的数据进行搬移, 如果这些数据搬移全部通过 CPU 进行, 将极大的浪费 CPU 资源, 但通用处理器往往 DMA 通道数量较少, 甚至没有 DMA 通道, 这也将影响信号处理的效率。

针对这些缺点, 当前的发展趋势是在通用处理器中内嵌硬件数字信号处理单元, 如很多视频处理器产品都是在 ARM9 处理器中嵌入 H.264、MPEG4 等硬件视频处理模块, 从而取得了较好的处理效果; 而另一条路径是在单片中集成 ARM 处理器和 DSP 处理器, 类似的产品如德州仪器的 OMAP 处理器及最新的达芬奇视频处理器, 它们就是在一个芯片中集成了一个 ARM9 处理器和一个 C55x 处理器或一个 C64x 处理器。

3. 利用可编程逻辑阵列 (FPGA) 进行实时数字信号处理

随着微电子技术的快速发展, FPGA 的制作工艺已经进入到 45nm 时期, 这意味在一片集成电路当中可以集成更多的晶体管, 芯片运行更快, 功耗更低。其主要优点如下。

(1) 适合高速信号处理: FPGA 采用硬件实现数字信号处理, 更加适合实现高速数字信号处理, 对于采样率大于 100MHz 的信号, 采用专用芯片或 FPGA 是适当的选择。

(2) 具有专用数字信号处理结构: 纵观当前最先进的 FPGA, 如 ALTERA 公司的 Stratix II、III 系列、Cyclone II、III 系列, Xilinx 公司的 Virtex-4、Virtex-5 系列都为数字信号处理提供了专用的数字信号处理单元, 这些单元由专用的乘法累加器组成, 所提供的乘法累加器不仅减少了逻辑资源的使用, 其结构也更加适合实现数字滤波器、FFT 等数字信号处理算法。

使用 FPGA 的缺点如下。

(1) 开发需要较深的硬件基础: 无论用 VHDL 还是 Verilog HDL 语言实现数字信号处理功能都需要较多的数字电路知识, 硬件实现的思想与软件编程有着很大区别, 从软件算法转移到 FPGA 硬件实现存在着很多需要克服的困难。

(2) 调试困难: 对 FPGA 进行调试与软件调试存在很大区别, 输出的信号需要通过示波器、逻辑分析仪进行分析, 或者利用 JTAG 端口记录波形文件, 而很多处理的中间信号量甚至无法引出进行观察, 因此 FPGA 的更多工作是通过软件仿真来进行验证的, 这就需要编写全面的测试文件, FPGA 的软件测试工作是十分艰巨的。

4. 利用数字信号处理器实时实现数字信号处理

数字信号处理器 (Digital Signal Processor, DSP) 是一种专门为实时、快速实现各种数字信号处理算法而设计的具有特殊结构的微处理器。20 世纪 80 年代初, 世界上第一片可编程 DSP 芯片的诞生为数字信号处理理论的实际应用开辟了道路; 随着低成本数字信号处理器的不断推出, 更加促进了这一进程。90 年代以后, DSP 芯片的发展突飞猛进。其功能日益强大, 性能价格比不断上升, 开发手段不断改进。DSP 芯片已成为集成电路中发展最快

的电子产品之一。DSP 芯片迅速成为众多电子产品的核心器件，DSP 系统也被广泛地应用于当今技术革命的各个领域——通信电子、信号处理、自动控制、雷达、军事、航空航天、医疗、家用电器、电力电子，而且新的应用领域还在不断地被发现、拓展。可以说，基于 DSP 技术的开发应用正在成为数字时代应用技术领域的潮流。

1.2 数字信号处理器的特点

DSP 系统的应用领域极其广泛，目前其主要的应用领域如下。

- (1) 基本信号处理：数字滤波器、自适应滤波、FFT、相关运算、谱分析、卷积运算、模式匹配、窗函数、波形产生和变换等。
- (2) 通信：调制解调、自适应均衡、数据加密、数据压缩、回波抵消、多路复用、传真、扩频通信、纠错编码等。
- (3) 语音：语音编码、语音合成、语音识别、语音增强、说话人的辨认和确认、语音邮件、语音储存等。
- (4) 图形图像：二维和三维的图形处理和图像的压缩、传输与增强、机器人视觉等。
- (5) 军事：保密通信、雷达信号处理、声呐信号处理、导航、导弹制导等。
- (6) 仪器仪表：频谱分析、函数发生、锁相环、地震信号处理等。
- (7) 控制：引擎控制、声控、自动驾驶、机器人控制、磁盘控制等。
- (8) 医疗：助听、超声设备、诊断工具、患者监护等。
- (9) 家用电器：高保真音响、智能玩具与游戏、数字电话、数字电视等。

DSP 当前最大的应用领域是通信。以无线通信领域中的数字蜂窝电话为例，蜂窝电话中的 DSP 协调模拟基带芯片、电源处理芯片，数字基带处理芯片，RF 射频处理芯片合理而快速地工作，并兼有开发和测试的功能。使移动通信设备更加个性化、智能化。

军事领域是高性能 DSP 的天地。例如，雷达图像处理中使用 DSP 进行目标识别和实时飞行轨迹估计，要求浮点 DSP 每秒执行数十亿次浮点运算，而定点 DSP 的运算能力已经高达 9600MIPS。

嵌入 DSP 的家用电器已经融入了我们的生活之中。例如，在清晰数字电视中，就采用 DSP 实现了其中关键的 MPEG2 译码电路；又如，使用 DSP 技术的家庭音响可以产生比模拟音响更自然，更清晰和更丰富的音响效果；再如，配置了 DSP 处理器的洗衣机、冰箱不仅提高了系统的功能、效率和可靠性，减少了系统能耗和电磁干扰，而且更加容易操作和控制。

DSP 的应用领域也在不断地扩大。例如，DSP 是运行计算机图像学 (Computer Graphics, CG) 软件和提供虚拟现实 (Virtual Reality, VR) 系统三维图形处理能力最为关键的器件。DSP 使 CG、VR 传统分析方法得到了质的飞跃。可以预见，随着 DSP 芯片性价比的不断提高和新的实用 DSP 算法的不断出现，DSP 系统的应用在深度和广度上会有更大的发展。

1.2.1 存储器结构

众所周知，微处理器的存储器结构分为两大类：冯·诺伊曼结构和哈佛结构。由于成本的原因，GPP 广泛使用冯·诺伊曼存储器结构。典型冯·诺伊曼结构的特点是只有一个

存储器空间、一套地址总线和一套数据总线；指令、数据都存放在这个存储器空间中，统一分配地址，所以处理器必须分时访问程序和数据空间。通常，做一次乘法会发生 4 次存储器访问，用掉至少 4 个指令周期。

为了提高指令执行速度，DSP 采用了程序存储器空间和数据存储器空间分开的哈佛结构和多套地址、数据总线，其结构如图 1-2 所示。哈佛结构是并行体系结构，程序和数据存于不同的存储器空间，每个存储器空间独立编址、独立访问。因此，DSP 可以同时取指令（来自程序存储器）和取操作数（来自数据存储器）；而且，还允许在程序空间和数据空间之间相互传送数据。哈佛读/写结构使 DSP 很容易实现单周期乘法运算。

今天，高性能 GPP 采用了片内高速缓存（Cache）技术以加快其处理速度。在 DSP 中也引入了这一技术，TMS320VC5510 就为内核提供了指令高速缓存。采用这一技术的原因是指令可能存储在内部存储器或外部存储器，而当其存储在外部存储器时，CPU 可以用高速缓存保存最近执行的指令，从而提高了系统的处理效率。

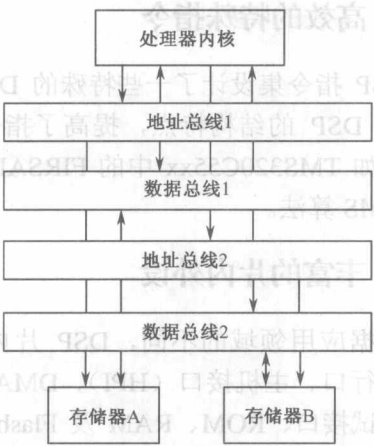


图 1-2 双总线存储器结构

1.2.2 流水线

流水线结构将指令的执行分解为取指、译码、取操作数和执行等几个阶段。在程序运行过程中，不同指令的不同阶段在时间上是重叠的，流水线结构提高了指令执行的整体速度，有助于保证数字信号处理的实时性。因此，所有 DSP 均采用一定级数的流水线，如 TMS320C54x DSP 采用 6 级流水线，而 TMS320C6xxx DSP 采用 8 级流水线。TMS320C55x DSP 的流水线则被分为指令流水线和执行流水线两部分，指令流水线完成访问地址产生、等待存储器回应、取指令包、预解码等工作；执行流水线完成译码、读取/修改寄存器、读操作数和输出结果等工作。

1.2.3 硬件乘法累加单元

由于 DSP 任务包含大量的乘法—累加操作，所以 DSP 处理器使用专门的硬件来实现单周期乘法，并使用累加器寄存器来处理多个乘积的累加；而且几乎所有 DSP 指令集都包含有 MAC 指令。而 GPP 通常使用微程序实现乘法。

1.2.4 零开销循环

DSP 算法的特点之一是主要的处理时间用在程序中的循环结构中，因此多数 DSP 都有专门支持循环结构的硬件。所谓“零开销”（zero overhead）是指循环计数、条件转移等循环机制由专门硬件控制，而处理器不用花费任何时间。通常 GPP 的循环控制是用软件来实现的。

1.2.5 特殊的寻址方式

除了立即数寻址、直接寻址、间接寻址等常见寻址方式之外，DSP 支持一些特殊的寻址方式。例如为了降低卷积、自相关算法和 FFT 算法的地址计算开销，多数 DSP 支持循环寻址和位倒序寻址。而 GPP 一般不支持这些寻址方式。

1.2.6 高效的特殊指令

DSP 指令集设计了一些特殊的 DSP 指令用于专门的数字信号处理操作。这些指令充分利用了 DSP 的结构特点，提高了指令执行的并行度，从而大大加快了完成这些操作的速度。例如 TMS320C55xx 中的 FIRSADD 指令和 LMS 指令，分别用于对称结构 FIR 滤波算法和 LMS 算法。

1.2.7 丰富的片内外设

根据应用领域的不同，DSP 片内集成了众多类型的硬件设备。例如，定时器、串行口、并行口、主机接口（HPI）、DMA 控制器、等待状态产生器、PLL 时钟产生器、JTEG 标准测试接口、ROM、RAM 及 Flash 等，如图 1-3 所示。这些片内外设提高了处理速度和数据吞吐能力，简化了接口设计，同时降低了系统功耗和节约了电路板空间。



图 1-3 DSP 片内外设实例

除了上述软、硬件区别之外，从程序开发的角度上，DSP 和 GPP 也有重要区别。例如，GPP 一般使用 C 语言或 C++ 语言等高级语言；但因为大多数高级语言并不适合于描述典型 DSP 算法，所以 DSP 应用程序一般要用汇编语言或 C 语言与汇编语言嵌套的方式编写。即使采用 C 源代码编译为汇编代码的方法，许多核心代码最后还要用汇编语言进行手工优化。此外，大多数 DSP 厂商都提供一些开发仿真工具，以帮助程序员完成其开发仿真工作。DSP 仿真工具能够精确到指令周期，这对于确保实时性和代码优化非常重要。而 GPP 厂商通常并不需要提供这样的工具。