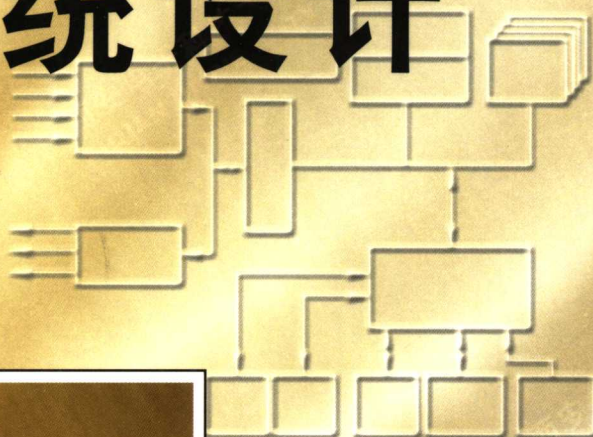


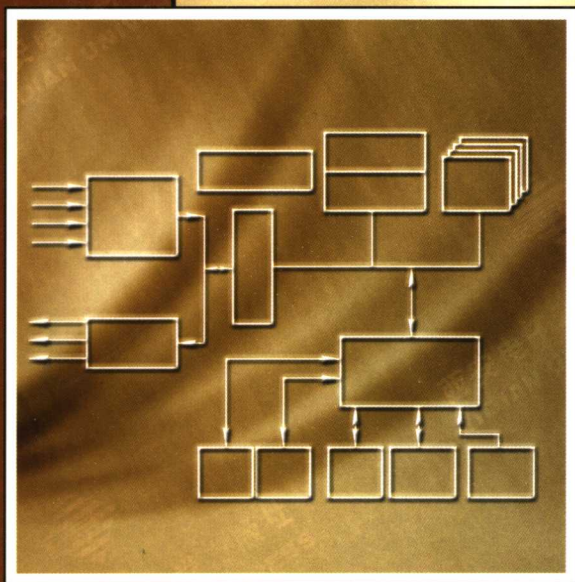


研究生系列教材

实时信号处理 系统设计



苏涛 何学辉 吕林夏 编著



西安电子科技大学出版社
<http://www.xduph.com>

内 容 简 介

本书介绍了以数字信号处理技术为核心的实时信号处理系统的设计方法,内容包括高速数字电路设计、数/模混合电路的硬件设计、DSP和FPGA等处理器的软/硬件设计以及系统设计所应考虑方方面面。

本书可作为通信、雷达和电子工程相关专业的研究生和高年级本科生的教材,也可作为相关领域的科研和工程人员系统掌握实时信号处理系统设计方法的参考书。

图书在版编目(CIP)数据

实时信号处理系统设计/苏涛等编著. —西安:西安电子科技大学出版社,2006.5

ISBN 7-5606-1657-7

I. 实… II. 苏… III. 数字信号—信号处理—实时操作系统—系统设计
IV. TN911.72

中国版本图书馆 CIP 数据核字(2006)第 025427 号

责任编辑 云立实 阎 彬

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)88242885 88201467 邮 编 710071

<http://www.xduph.com>

E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2006年5月第1版 2006年5月第1次印刷

开 本 787毫米×1092毫米 1/16 印 张 18.75

字 数 444千字

印 数 1~4000册

定 价 25.00元

ISBN 7-5606-1657-7/TP·0400

XDUP 1949001 - 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。

前 言

在实时数字信号处理应用需求的推动下,在信号处理仿真技术、电子设计自动化技术和超大规模集成电路技术飞速发展的今天,实时信号处理系统的性能 and 设计方法都发生了巨大的变化。以处理器为核心的各种规模的信号处理系统在通信、电力、雷达、导航、尖端武器等领域得到了越来越广泛的应用,在研制周期不断缩短的同时,其性能和复杂度也在不断提高。设计者在有更多的实现方法可供选择的同时,也面临着更大的挑战和竞争。结合当今先进的信号处理算法、算法仿真技术、硬件技术、软件技术、系统构建和检验方法,设计者可以在较短的时间内,以较小的代价制作完成一个具有较高性能的设备。

本书前6章的内容从信号处理系统的概述、设计流程、算法仿真开始,讲述了实时信号处理系统的概念及发展,基本组成及常用算法,介绍了 Matlab 工具在具体工程应用中的拓展用途,描述了如何根据具体应用选择系统结构和关键组件,特别是各类处理器的方法,并且详细分析对比了 DSP 和 FPGA 这两大处理器技术。第7、8章围绕硬件电路的设计,对数据存储和通信、模/数混合设计、高速数字电路设计、电源设计等进行了全面详细的论述。第9章对程序仿真技术、可编程处理器的软件设计和软、硬件协同设计方法进行了介绍。第10章介绍了保证系统安全性和可靠性的方法。本书将所涉及方法的简单原理和具体实例结合起来,以适合不同层面的应用需要。书中提供的代表性和实用性很强的硬件设计实例,大都直接取自于作者的科研实践。

本书第1~3、5~7章由苏涛编写,第8、9章由何学辉编写,第4、10章由吕林夏编写。全书由苏涛统稿。在编写过程中,得到了康丽艳、牛亚莉、江玲、孙进卿等研究生的帮助,这里对他们的辛勤工作深表谢意。

实时信号处理系统设计是一个快速发展的领域,本书的内容涉及面广,再加上作者的水平 and 掌握的资料有限,书中存在不当与错误之处在所难免,敬请读者提出宝贵意见。

作 者

2005.12

目 录

第1章 实时信号处理概述	1	3.2.2 带通信号采样定理	38
1.1 信号处理的概念	1	3.2.3 完全正交信号的采样率	39
1.2 DSP 技术	2	3.3 数字正交采样	41
1.3 实时信号处理	4	3.4 数据格式	42
1.4 高速信号处理系统	6	3.4.1 浮点数据格式	42
1.5 处理器的发展现状	6	3.4.2 定点数据格式	43
1.6 实时信号处理的性质	7	3.5 数字滤波器设计	45
思考题	10	3.6 谱分析和 FFT	47
第2章 实时信号处理系统的设计		3.7 卷积和相关的快速算法	48
流程	11	3.7.1 卷积运算量	49
2.1 实时信号处理系统的设计框图	11	3.7.2 频域法最佳分段长度的选择	50
2.2 算法仿真	12	3.8 自适应信号处理	52
2.3 DSP 软件设计方法的变革	15	3.8.1 闭环自适应系统	53
2.4 用 Matlab 仿真 DSP 软件设计	18	3.8.2 开环自适应系统	54
2.4.1 用 Matlab 仿真浮点 DSP	18	3.9 矩阵运算	56
2.4.2 用 Matlab 精确仿真定点 DSP		3.10 常用运算	56
运算	20	3.11 专用算法	58
2.4.3 用 Matlab 功能仿真定点 DSP		思考题	59
运算	25	第4章 信号处理前端和后端设计	60
2.4.4 常用的 Matlab 函数	25	4.1 模拟前端	60
2.4.5 Matlab 辅助设计应用实例	30	4.1.1 小信号放大	61
2.5 定点 DSP 模块库简介	31	4.1.2 滤波	61
2.6 Matlab 下的 DSP 集成设计环境	32	4.1.3 增益控制	62
2.6.1 Matlab - DSP 集成设计环境下的		4.1.4 抗混叠滤波	63
工具包	33	4.2 模拟/数字变换	63
2.6.2 Matlab Link for CCS Development		4.2.1 采样速度	63
Tools 简介	34	4.2.2 量化位数	65
2.6.3 CCS Link 面向 TI DSP 的系统级		4.2.3 量化位数和采样速度的转换	65
设计方法	34	4.2.4 其他因素	66
2.6.4 Matlab - DSP 集成设计实例	35	4.2.5 ADC 的性能指标	67
思考题	36	4.2.6 设计高品质的 ADC 电路	73
第3章 基本原理和算法	37	4.2.7 ADC 的种类和选型	76
3.1 实时信号处理系统的基本结构	37	4.2.8 ADC 的数字接口	82
3.2 采样定理	37	4.3 数字/模拟变换	85
3.2.1 Nyquist 采样定理	37	4.4 信号产生	86
		4.4.1 模拟式振荡信号发生器	87

4.4.2 数字式信号发生器	87	7.2 运算和 I/O 的并行	145
4.5 数字、模拟频率	93	7.3 数据通信方式	146
4.6 数字信号预处理	93	7.3.1 数据通信分类	146
思考题	95	7.3.2 总线的性能	148
第 5 章 处理器的实现方法	96	7.3.3 总线标准的发展	148
5.1 多种数字信号处理器的比较	96	7.3.4 总线的比较与演变	153
5.1.1 通用处理器	97	7.4 各种类型的存储器	154
5.1.2 通用处理器的选择	99	7.4.1 存储器种类	155
5.1.3 硬件处理器	99	7.4.2 等待模式	157
5.1.4 软/硬件处理器性能的比较	100	7.4.3 软等待访问模式	157
5.1.5 软/硬件处理器开发手段的 比较	103	7.4.4 硬等待访问模式	158
5.1.6 其他实现方法	106	7.4.5 软等待和硬等待	159
5.2 DSP 处理系统的组成和设计	106	7.5 译码和片选	160
5.3 DSP 软件设计	108	7.6 DSP 与存储器接口	161
5.4 FPGA 的设计应用	108	7.6.1 DSP 与 SRAM 接口	161
5.5 处理器技术的发展趋势	122	7.6.2 DSP 与 EPROM 接口	164
思考题	122	7.6.3 DSP 与 Flash 接口	164
第 6 章 多处理器系统设计	124	7.6.4 DSP 与快速 Flash 接口	170
6.1 并行处理的必要性	124	7.6.5 DSP 与多端口存储器的接口	172
6.1.1 大型并行系统的结构	124	7.6.6 DSP 直接与 SDRAM 接口	174
6.1.2 大规模并行处理器的发展	127	7.6.7 DSP 扩展 SDRAM 接口	180
6.1.3 实时信号处理的需求	129	7.6.8 DSP 与同步突发 SRAM(SBSRAM) 接口	181
6.2 实时并行处理机的构成	129	7.7 扩展存储接口方式	184
6.2.1 并行处理机拓扑结构的分类 和选择	129	7.7.1 地址/数据复用总线	184
6.2.2 并行处理机的互连网络	132	7.7.2 将 DSP 接口扩展为异步串口	188
6.3 并行处理机的性能指标	134	7.7.3 USB 通信接口	189
6.3.1 并行处理机的基本性能指标	134	7.8 DSP 与 PCI/CPCI 总线的连接	190
6.3.2 影响性能的因素	136	思考题	191
6.4 实时并行信号处理机的设计与实现	138	第 8 章 硬件设计	193
6.4.1 实时性要求	139	8.1 电源设计	193
6.4.2 通用性和易维护性	139	8.1.1 电源需求	193
6.4.3 系统设计	140	8.1.2 直流稳定电源的种类及选用	194
思考题	141	8.1.3 线性电源	195
第 7 章 数据存储和通信	142	8.1.4 开关电源	198
7.1 运算和 I/O 的平衡	142	8.1.5 其他类型的变换器	199
7.1.1 运算和 I/O	142	8.1.6 电源用电容器的选择	200
7.1.2 数据采样率和数据量	143	8.1.7 电源的设计要求	203
7.1.3 处理单元个数	143	8.1.8 电源的实现方法	204
7.1.4 解决 I/O 瓶颈的软件途径	144	8.1.9 开关型变换器的 EMC 抑制	206
7.1.5 解决 I/O 瓶颈的硬件途径	144	8.1.10 开关电源 PCB 和整体布局的 设计要点	209
		8.1.11 电源安全设计	210

8.2 时钟电路	211	思考题	246
8.2.1 时钟电路的选择原则	211	第9章 软件设计	247
8.2.2 时钟电路电源和地的设计	211	9.1 软件设计的范畴	247
8.2.3 系统时钟与局部时钟	212	9.2 算法仿真	248
8.2.4 时钟的影响	212	9.3 程序设计的一般框架	249
8.3 电源监控电路和复位电路	213	9.4 DSP 的软件开发工具	250
8.4 驱动和隔离以及电平转换	214	9.4.1 代码生成及调试工具	250
8.4.1 驱动和隔离	214	9.4.2 软件仿真器	251
8.4.2 电平转换	215	9.4.3 硬件仿真器	251
8.5 测试和自检	218	9.4.4 DSP 程序设计语言	251
8.5.1 信号检测	218	9.5 DSP 的程序设计和优化	252
8.5.2 自检功能	220	9.5.1 DSP 的程序设计内容	252
8.6 高速电路设计的特殊性	220	9.5.2 优化的方法	254
8.7 传输线效应	222	9.5.3 基于算法的优化措施	255
8.7.1 串行端接	223	9.5.4 基于高级语言的优化措施	255
8.7.2 并行端接	223	9.5.5 基于硬件特点的优化措施	256
8.8 信号完整性	224	9.5.6 基于代码的优化措施	257
8.8.1 信号完整性问题的起因、种类 及表现	224	9.5.7 优化的代价	258
8.8.2 信号完整性分析模型	226	9.6 操作系统在 DSP 程序设计中的 运用	259
8.8.3 信号完整性分析	226	9.7 DSP 系统的测试	262
8.8.4 信号完整性问题的解决方法	227	9.8 软件和硬件的关系	262
8.9 电磁兼容性	228	9.9 软、硬件协同设计	264
8.9.1 电磁干扰的要素及其种类	228	9.9.1 系统描述	264
8.9.2 电路内干扰	231	9.9.2 系统设计	265
8.9.3 外界干扰	232	9.9.3 系统评价	265
8.10 模/数混合电路	232	9.9.4 综合实现	266
8.10.1 模拟地和数字地	233	思考题	266
8.10.2 采样时钟	235	第10章 系统设计的考虑	267
8.10.3 模/数系统的布局	236	10.1 折衷设计	267
8.10.4 模/数混合电路的设计要点	237	10.2 效率和成本以及兼容性	268
8.11 PCB 的设计要点	238	10.3 功耗和散热	269
8.11.1 电源设计	239	10.3.1 降低功耗	270
8.11.2 地线设计	239	10.3.2 散热	271
8.11.3 重要信号线的设计	240	10.4 抗干扰设计	274
8.11.4 时钟电路干扰及其抑制	240	10.4.1 硬件抗干扰设计	274
8.11.5 高速电路设计系统中的非理想 因素	241	10.4.2 软件抗干扰设计	275
8.11.6 电磁兼容性设计	242	10.5 可靠性设计	276
8.11.7 PCB 的抗干扰措施	243	10.5.1 可靠性的相关定义	276
8.11.8 器件建库和 BGA 设计	244	10.5.2 元器件的选用	278
8.11.9 电路设计工具	245	10.5.3 元器件的降额使用	280
8.11.10 电路调试	246	10.5.4 可靠性计算	280
		10.5.5 简化设计	285

10.5.6 低功耗设计	285	10.6 冗余设计	287
10.5.7 保护电路设计	286	10.7 加密	287
10.5.8 灵敏度分析	286	10.8 实时信号处理系统的优化设计	289
10.5.9 均衡设计	286	10.9 进度估计	289
10.5.10 整机热设计	286	思考题	290
10.5.11 元器件装配工艺对可靠性的 影响	286	参考文献	291

第1章 实时信号处理概述

1.1 信号处理的概念

信号是信息的载体和表现形式。所谓信号处理,就是对信息的变换和提取,是指将信息从各种噪声、干扰的环境中提取出来,并变换为一种便于被人或机器所使用的形式。

由于信号以多种形式存在于自然界中,因此,信号处理也可按多种形式进行。例如,光信号经过透镜、偏振器可以得到处理。但我们通常所说的信号处理,几乎无一例外地以电信号的形式进行,其他形式的信号可以通过传感器转化为电信号。电信号通常可以分为两大类:一类是时间和幅度都连续取值的模拟信号;另一类是时间和幅度只能取某些离散值的数字信号。所以,信号处理也分为模拟信号处理和数字信号处理两大类。

实现信号处理的设备称为信号处理系统。对模拟信号进行处理的设备称为模拟系统;而对数字信号进行处理的设备称为数字系统。由于自然界中的绝大多数信号属于模拟信号,因此,最早的信号处理技术大都采用模拟信号处理,所用的设备当然也就是模拟系统,如使用模拟计算机和晶体管及 R 、 L 、 C 等基本模拟元器件所组成的系统。早期的数字处理系统速度很慢,又很昂贵。而近 30 年来,随着大规模集成电路的普及和数字计算机的广泛使用,数字信号处理技术有了很大的发展,并得到了越来越多的应用。

数字系统与模拟系统相比,具有如下显著的优点:

(1) 高稳定性与高精度。与模拟系统相比,数字系统的特性不易受到使用条件的影响,其稳定性、可靠性和计算精度都很高。以加法器为例,由于模拟电路加法器采用了运放、电阻、电容,因而电源电压波动,电阻、电容误差,负载阻抗不匹配,温度漂移等各种因素都会使加法结果不精确或不稳定。同样的电路,其电气特性也无法一致,如至少存在 1% 的误差,调整校正很难,即使校准了,当温度变化后,还需重新调整。而数字系统则不然,若数据采用 16 位定点格式,则其精度即可达到 $1/65536 = 1.5 \times 10^{-5}$;若采用 32 位浮点格式,则其精度即可达到 1.7×10^{-38} 。

(2) 高度的灵活性。采用数字系统,可以通过在线重构(如用 FPGA)或软件编程(如用 DSP/计算机),在相同的硬件上完成不同的处理功能。

(3) 便于大规模集成。随着微电子技术的发展,数字电路的集成度越来越高,数字系统的体积更小,重量更轻,可靠性更高,且便于批量生产。

(4) 数字系统可以完成模拟系统无法完成的各种复杂处理功能,使许多复杂的信号处理方法,如自适应算法、小波变换等得以完成。

(5) 便于存储和精确回放与验证。通常, 模拟信号用磁带存储, 数字信号用磁盘存储。前者会因为磨损、失效等原因而使信号失真, 后者则始终是无失真的。不过在目前, 存储同样质量的电视信号, 磁带体积小于磁盘。为降低存储量, 在模拟电视信号数字化的过程中采用了较低的采样率和较少的位数, 丢失了一定的信息量, 因此, 目前数字化后的电视信号质量不如原模拟信号的质量高。

(6) 便于系统联网, 信息共享。

1.2 DSP 技术

数字系统中的数字信号处理又简称为 DSP。这是广义的 DSP 概念, 它包括各种数字信号处理的算法, 诸如 DFT 及 FFT 变换、FIR 数字滤波等, 还有数字信号处理得以实现的硬件设备, 即各种数字信号处理系统。而狭义的 DSP 概念则专门指 DSP 处理器芯片, 如 TI 公司的 TMS 系列芯片、ADI 公司的 SHARC 系列芯片等。本书中的 DSP 概念是指 DSP 处理器芯片, 它与 CPU、FPGA 并列作为数字信号处理的关键硬件核心。

数字信号处理的历史可以追溯到 20 世纪初, 那时人们提出的数值计算、Z 变换和离散傅立叶变换(DFT), 就是当代数字信号处理技术的理论基础。但当时因为条件所限, 所以实际应用中一直沿用模拟处理方法。1965 年, 库利(J. W Cooley)和图基(J. W Tukey)提出了计算 DFT 的快速算法(FFT), 使 DFT 得到了实际的应用, 从而大大推动了数字信号处理的发展。同时, 数字滤波的基本理论也在逐渐形成, 如凯泽(Kaiser)提出了双线性变换滤波器设计方法。因此, 到 20 世纪 60 年代中期就形成了较为完整的数字信号处理理论。与此同时, 数字元件工艺及数字计算机也在不断发展, 这为数字信号处理技术的发展奠定了物质基础。

20 世纪 70 年代, 大规模集成电路(LSI)技术得到了发展, 这为数字信号处理系统的硬件实现提供了条件。最初, 人们用多个 IC 芯片来实现数字信号处理中的一个基本运算, 如加法、乘法、迟延等。1980 年前后陆续出现了将许多个加法器、乘法器和迟延电路等集中在一起, 按信号流程排列的 LSI 芯片。世界上第一个单片 DSP 芯片应当是 1978 年 AMI 公司发布的 S2811。1979 年, 美国 Intel 公司发布的商用可编程器件 2920 是 DSP 芯片的一个重要里程碑。1980 年, 日本 NEC 公司推出的 μ PD7720 是第一个具有硬件乘法器的商用 DSP 芯片, 从而被认为是第一块成熟的单片 DSP 器件。

随着超大规模集成电路(VLSI)技术的发展, 1982 年, 美国德州仪器公司(Texas Instruments, 简称 TI)推出了世界上第一代 DSP 芯片 TMS320C10 及其系列产品, 它标志着在实时数字信号处理应用领域有了重大突破。虽然这种 DSP 器件的功耗和尺寸稍大, 但运算速度却比当时的计算机核心 CPU(中央处理器)快了几十倍, 尤其在语音合成和编码/解码器中得到了广泛应用。作为最大的 DSP 厂商, TI 公司持续改进 DSP, 相继推出了第二代 DSP 芯片 TMS320C20、TMS320C25/C26/C28(统称 C2X, 与后来的 TMS320C5X 兼容, 而与目前常用的 TMS320C5000 不兼容)及第三代 DSP 芯片(32 位浮点型的 TMS320C30/C31/C32, 统称 C3X)。到了 20 世纪 90 年代, DSP 的更新速度越来越快, TI 公司相继推出了第四代 DSP 芯片 TMS320C40/C44 及第五代 DSP 芯片 TMS320C54X/

C54XX/C55XX(统称 C5000), 并推出了第二代 DSP 芯片的兼容改进型 TMS320C5X 和 TMS320C2XX(统称 C2000), 还有集多片 DSP 芯片于一体的高性能 DSP 芯片 TMS320C8X 以及目前速度最快的第六代 DSP 芯片 TMS320C62XX/C67XX 等。

日本的 Hitachi 公司在 1982 年推出第一个基于 CMOS 工艺的浮点 DSP 芯片。1983 年, 日本的 Fujitsu 公司推出了 MB8764, 其指令周期为 120 ns, 且具有双内部总线, 从而使 DSP 处理吞吐量发生了一个大的飞跃。而第一个高性能浮点 DSP 芯片应是 AT&T 公司于 1984 年推出的 DSP32。这些 DSP 都比后来大为流行的浮点 DSP——TI 公司的 TMS320C30 出现得早。

与其他公司相比, Motorola 公司在推出 DSP 芯片方面相对较晚。1986 年, 该公司推出了定点处理器 MC56001, 后来衍生出了定点处理器 MC56000 系列; 1990 年, 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)后来居上, 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 ADSP21XX 系列, 如 ADSP2103/2105、ADSP2111/2115、ADSP2161/2162/2164、ADSP2171, 目前常用的是 ADSP218X/219X。AD 推出的另一种定点 DSP 是与 ADSP21XX 不兼容、但速度极快的 Blackfin。浮点 DSP 芯片有 ADSP21000/21020、ADSP21060/21062、ADSP21160/21161 等, 这些浮点 DSP 指令兼容, 统称为 SHARC。目前最快的浮点 DSP 是 AD 公司的 Tiger SHARC 系列, 其典型代表是 TS101、TS201。

随着竞争的加剧, DSP 厂商发生了分化。目前, 最先进的处理器技术全部由美国公司主导。TI 公司作为最大的 DSP 供应者, 提供了各种型号的 DSP, 其次是 AD 公司。Motorola 公司和 Lucent(前身是 AT&T)公司则退出了通用 DSP 市场, 转而提供通信专用的 DSP。

与 CPU 和单片机等处理器不同, DSP 更适合于高性能、重复性、数值运算密集型的实时处理, 并提供了很高的数据输入/输出能力。就 FFT、相关运算来说, DSP 不仅处理速度快, 而且可以无间断地以流水方式完成数据的实时输入/输出。DSP 的结构相对单一, 普遍采用汇编语言编程, 其任务完成时间的可预测性要比结构和指令复杂(超标量指令)、依赖于编译系统的普通微处理器强得多。以一个 FIR 滤波器的实现为例, 每输入一个数据, 对应每阶滤波器系数, 任何一种处理器都需要的操作是: 一次乘, 一次加, 一次取指, 两次取数, 有时还需要专门的数据移动操作。DSP 可以单周期完成乘加并行操作以及 3~4 次的数据存取操作, 而普通微处理器则需要至少 4 个指令周期来完成乘法, 上述所有操作需要 7 个指令周期。因此, 在相同的指令速度和片内指令缓存条件下, DSP 是普通微处理器运算速度的 7 倍左右。

自 1980 年以来, DSP 芯片得到了突飞猛进的发展, DSP 芯片的应用也越来越广泛, 并逐渐成为电子产品更新换代的决定因素。从运算速度来看, MAC(一次乘法和一次加法)时间已经从 20 世纪 80 年代初的 400 ns(如 TMS320C10)减少到 10 ns 以下(如 TMS320C54XX、TMS320C62XX/67XX 等), 处理能力提高了几十倍。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 数量增加了一个数量级。DSP 芯片的引脚数量从 1980 年的最多 64 个增加到现在的 600 个以上, 引脚数量的增加, 意味着结构灵活性的增加, 更便于外部存储器的扩展和处理器间的通信。

近年来发展起来的新型集成电路技术和相应的开发工具,如 FPGA、CPLD 等技术,使开发者可以随时构成任意的专用集成电路,这为信号的实时处理提供了有利条件,使得数字信号处理技术发展得更快速了。

1.3 实时信号处理

目前,数字信号处理已经成为信号处理的主要手段。从应用角度来看,一般将数字信号处理分为两大类,一类是非实时信号处理,另一类是实时信号处理。

非实时信号处理是指对已采集好的信号或者计算机产生的仿真信号进行处理,不要求现场完成并得到结果。例如,在地下资源勘探过程中,研究人员可以将采集的信号带回实验室,分门别类地进行反复的分析处理,直至得到正确满意的结果。

实时信号处理则要求在限定的时间内将采集的数据在现场处理完成并得到一定的结果,即信号处理的时间要小于或者等于下一批数据的输入时间,有时甚至要求在特定的时间、地点来完成信号处理。实时又可以分为软实时和硬实时。软实时要求响应时间越短越好,而硬实时要求响应时间必须小于设定值。如安装在导弹、鱼雷上的制导系统,需要在极短的时间内完成发、收信号,目标跟踪、识别、引爆等多个步骤。又如雷达跟踪系统,需要对采集到的信号进行现场处理,并报告出目标的方位、速度等信息,以调整跟踪系统的控制信息。安装在汽车上的防撞雷达,对响应速度和准确度也都有很高的要求。随处可见的手机通信,是在一个小巧、低功耗的移动设备内,由处理器将接收到的信号进行实时处理,以便促成通信双方的低延迟通话过程。利用便携式仪器进行的道路地基、堤坝漏洞缝隙探测也强调实地操作,即时得到结果。还有诸如 GPS 导航设备等也都有实时性要求。

目前,实时信号处理系统大多数是脱离了 PC 机的嵌入式系统,常见的是由 DSP、单片机及 FPGA/CPLD 组成的系统等。PC 机虽然功能全面、便于使用,但受限于实时信号处理的特定要求而不便采用。实时信号处理系统带有专用性,一般具有延迟小、体积/功耗小、成本低等特点。所以大部分的实时信号处理都用专用设备,也有少数的用 PC 机来构成处理系统。

一个完备、复杂的实时信号处理系统可以由以下几部分组成:

- (1) 实时信号采集,简称 ADC。
- (2) 实时数据存储,采用 SRAM、SDRAM、Flash 和磁盘阵列。
- (3) 实时控制,以 CPLD/FPGA/ASIC 为主。
- (4) 实时信号产生,以 DAC/DDS 为主。
- (5) 实时运算,以 DSP/FPGA/ASIC 为主。
- (6) 实时通信,通过 VME/PCI 总线、LVDS 接口进行。
- (7) 电源。
- (8) 电磁兼容与屏蔽系统。

实时信号处理系统最主要的核心就是执行各种算法的运算部分。特别是对运算量极大的应用来说,采用合适的处理器和互连结构是至关重要的。

通常设计一个数字信号处理系统都需要经过非实时信号分析和实时信号处理系统的设

计这两个步骤,即:

第一步:理论分析和仿真计算,即所谓的前期预先研究。

理论分析和仿真计算又包括预先研究、分析、系统设计前的仿真和事后分析。设计人员常常要用到80%的时间进行算法搜集与研究,然后将其整理成为一个完整的处理流程,之后要在计算机上进行仿真验证。当前,最常用的仿真开发环境是Matlab,当然,还有Fortran、C语言等开发工具。通常的分析研究只需在PC机上就可以实现,但是对于复杂的、大型的工程,还会用到巨型机、多台连网计算机以及互联网。这种类型的信号处理侧重于分析、验证、测试和仿真,它在整个信号处理系统设计过程中占有极其重要的地位。

第二步:工程实现,包括系统设计、硬件设计和软件编程。

工程实现就要求设计者将搜集到的算法用硬件系统加上相应的软件来实现。现在的硬件系统设计一般以FPGA/CPLD和处理器来实现。FPGA/CPLD主要完成外围电路的地址译码、时序控制等,有时也会做一些如编码、FIR滤波之类的运算。而处理器则用软件编程实现所选择的算法。

这两步在方法上的区别在于:前者注重分析算法的性能,属非实时信号处理;而后者则属实时信号处理,注重系统设计的可行性以及整个工程的成本、硬件系统的体积,还有整个系统实际运行起来的可靠性。

在上述两步过程中,有两个容易混淆的概念:模拟与仿真。仿真和模拟都是从英文Simulation翻译而来。按照我们的习惯说法,用于理论分析的Simulation称为仿真,如Matlab仿真。在硬件设计中的Simulation有时也称为仿真,如在FPGA设计过程中,利用计算机模仿FPGA的逻辑和时序功能的过程也叫做仿真。而模拟一词用于近似的硬件设备中,即把“用简单硬件模仿实际硬件”的设备称为模拟器(Simulator),例如信号源模拟器(也可称为模拟信号源,但会和下文的模拟信号混淆)、飞行模拟器等。

然而,在DSP处理芯片的设计过程中,人们习惯于把前期用计算机软件验证程序功能的过程叫做Simulation,译为模拟,它以脱离DSP硬件电路板、利用PC机的资源模仿DSP及其外围电路的工作方式,营造出了一个模拟环境,并在此环境下调试DSP代码,此环境称为软件模拟器(Simulator)。把后期用实际硬件在线调试程序功能的过程叫Emulation,译为仿真,它将PC机和DSP电路板(称目标板)用专用的DSP仿真器(Emulator)及电缆连接起来,从PC机上实时监控、控制DSP的运行。这种手段称为DSP的实时仿真(Emulating)。因此,仿真和模拟的概念到底哪一个更接近于实际硬件,并不统一,甚至很混乱。

本书避免使用“模拟”一词,将Matlab中对算法的分析称做算法仿真,把FPGA设计中借助于计算机的验证也称做仿真。将DSP设计中借助于计算机的纯软件验证过程用软仿真(Simulation)表示。把与硬件相关的仿真过程,例如硬件在线调试DSP的过程称为硬仿真(Emulation)。

在DSP设计过程中,软仿真一般用于DSP代码的前期调试,只是验证代码的功能和性能。它使用方便,修改“硬件配置”时简便灵活,无需添加设备。但软仿真的缺点是速度太慢,例如一段图像压缩代码在真正的DSP上实时运行的时间是1s,而在2.0GHz奔腾IV机型上的软仿真器下需要1.5h,其速度相差5400倍。因此,用软仿真器验证运算复杂度高、运算量大的代码很不合适。此外,用软仿真器无法验证DSP在实际运行过程中的输

入/输出操作。而硬仿真对实际的 DSP 硬件目标板进行监控时,几乎可以真实地得到 DSP 实际运行过程中的状态信息。

1.4 高速信号处理系统

理论分析和仿真在整个信号处理系统设计过程中都占有极其重要的地位。这些分析和仿真工作从头到尾都是编程、计算、测试,除了计算机外,与实际设备不打交道。随着人类在科技领域研究的问题越来越深入、算法越来越复杂,需要的计算速度也越来越高。特别是在科学计算、地质分析、气象预测、仿真模拟、图像处理等领域,需要做几十亿到几万亿次每秒运算。如此庞大的运算量,已远远超出传统的 PC 机的处理能力。另外,在一些仿真实验过程中,要仿真的系统越来越大,仿真结果越来越细致逼真。例如,在核武器仿真实验中,采用千亿次每秒运算速度的处理机进行仿真,耗时将达一个月,这是研究人员无法承受的。可见,对于非实时信号处理来说,高速信号处理仍是一个问题。

伴随着理论分析的深入和算法的复杂化,在实时信号处理领域,也有各种各样的新方法不断出现,虽然新方法使信号处理性能提高了,但其复杂度和运算量也大大增加。因此,高速实时信号处理系统就应运而生。

高速实时信号处理系统常应用在如下领域:

- (1) 通信和语音信号:几千万到几亿次运算每秒,大的如基站,小的如手机、MP3。
- (2) 视频信号和图像:几亿到几十亿次运算每秒,如网络图像传输应用。
- (3) 工程应用:导航,探测,识别。
- (4) 尖端技术:航空,航天,武器控制。
- (5) 雷达、声纳信号:几十至几百亿次运算每秒。

这些领域对实时信号处理的要求是实时性强、迟延小、可靠性高、体积小、功耗小、成本低等。

DSP 等高速实时信号处理器和相关设备的出现,满足了上述要求。随着实时信号处理设备性价比的进一步提高,其应用已从单一尖端和军用领域发展到了民用领域的许多方面。

1.5 处理器的发展现状

高速信号处理的客观基础是超大规模集成电路(VLSI)设计手段和制作工艺的飞速提高。目前,VLSI 的制造工艺可达 $0.065\ \mu\text{m}$,集成度 1000 万门以上,速度达 3 GHz。而单个处理器的处理能力以 10 年 100 倍的速度增长,出现了每秒运算几十亿次的运算单元。这些物质条件的发展大大促进了信号处理的发展。

自 1978 年 Intel 公司推出的 8086,到现在的 Pentium IV 处理器(这并不是当前最快的处理器,而是最普及的高速处理器),20 多年的时间里,CPU 的速度是以超线性加速度提高的。在提高的这 27 000 倍里,540 倍来自于时钟频率的提高,另外 50 倍来自于处理器结构的改进和变革。例如,8086 执行一条指令需要 15 个时钟周期,而 Pentium 处理器一个

时钟周期即可执行3条指令。

结构的改进方面最突出的就是并行技术的应用。因为处理器性能受限于VLSI器件的开关速度、芯片内部的连线延迟以及其理论特征尺寸等,所以芯片内部的高速处理必须由多个部件并行完成。处理器或运算单元内部结构的设计一般用到两种并行的手段:流水线和片内多部件并发工作。以运算单元为例,取数、运算、存结果是由多个部件以流水方式完成的,而运算可以由乘法器、加法器并发完成。处理机系统设计也利用了类似的流水和并发手段。

近年来,计算机体系结构也在不断地更新,由原来的单机处理向多机处理发展,并以分布式并行处理为主要发展方向。计算机软件技术也从低层次设计向高层次设计发展,从独立的专用开发发展到分工合作、协同开发、系统集成。这些变化已扩展到实时信号处理领域。

最新的通用并行计算机集成了几百到几千个处理单元,可达到几万亿次每秒的运算能力(峰值速度)。本书第6章列出了目前世界上的许多顶级巨型计算机。与这些巨型机相对应,各种实时信号处理设备也已研制成功,如:

(1) 语音处理器:单片亿次运算每秒。

(2) 图像处理专用器:单片几十亿次运算每秒。

(3) 无线通信基站及软件无线电:几十亿次运算每秒。

(4) 空时二维信号处理(STAP)中的大规模并行处理(MPP):可达到几十亿次浮点运算每秒的速度。

(5) 脉动阵/波前阵列:应用ASIC技术,包含几十至上千个处理单元,每个处理单元速度可达1~100亿次运算每秒。

1.6 实时信号处理的性质

衡量实时信号处理性能的两个基本指标分别是数据吞吐能力(数据率)和处理延迟。

反映吞吐能力的量叫做吞吐率,即:处理一段数据的有效数据点数除以处理该段数据所用的时间。当处理设备给定时,所用的时间与运算量成正比,可见吞吐率反比于单位时间内的运算量。对批处理来说,平均吞吐能力为

$$S_d = \frac{\text{输入批数据长度}}{\text{处理该批数据的运算量(时间)}}$$

数据的输入速率一般情况下等于数据的采样率,输出速率等于输入速率。但在有些情况下,输出速率会低于输入速率,即输出的数据量小于输入的数据量。例如,在正交插值过程中,输入数据率为 f_s ,由于正交插值在低通滤波后会进行抽取,从而降低了数据率,使得输出数据率低于输入数据率。如果对于单纯的FIR滤波来说,数据率则是恒定的。

当输入、输出速率不相等时,吞吐能力取决于输入速率。

数据的输入、输出速率不相等的例子是抽取滤波器和内插滤波器。过采样下,输入速率大于输出速率,是为了便于A/D前的模拟滤波(初步滤波,防止采样产生混叠)。A/D转换后,再使用数字低通滤波,然后抽取,减轻了后续DSP的压力。抽取提高了数据利用率。

内插滤波是指在输入速率小于输出速率的情况下,为了平滑 D/A 输出台阶,便于 D/A 后的模拟滤波,在数字域提高采样率而采用的方法,具体方法是先将数字序列以某个间隔补零,再进行滤波。

实时信号处理系统中还有一个比较重要的概念就是延迟。延迟是指吞吐率给定后,从输入到输出之间的延迟时间。如 N 阶 FIR 滤波器的群延迟就是 $N/2$ 。群延迟是一个平均的概念,用下式来换算成时间

$$\text{群延迟} = \frac{N/2}{S_d} \quad (1.1)$$

延迟越小,实时性就越好。下面提到的批处理形式需要等一批数据都到来后,才能开始运算,其延迟显然比流水处理方法的延迟大。

1. 按处理形式分类

按处理形式可将实时信号处理分为流水处理和批处理两种。

流水处理:即逐点处理,如 FIR、IIR、DFT、DBF 等。其示意图如图 1.1 所示。流水处理的优点是结构简单,延迟小;但平均到每个数据的运算量较大,较复杂的算法难以用流水形式实现。

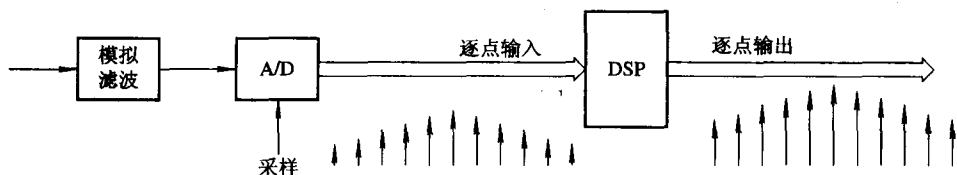


图 1.1 流水处理示意图

批处理(块处理):即分批处理,数据到齐后才开始处理,数据同时参加运算,结果也同时得到,如 FFT 及其他相干积累算法。其示意图如图 1.2 所示。这种处理方法可大大减小运算量,缩短运算时间,适合于较复杂的算法;但相对流水处理来说其延迟要大一些,且要求较大的存储量,系统实现较为复杂。

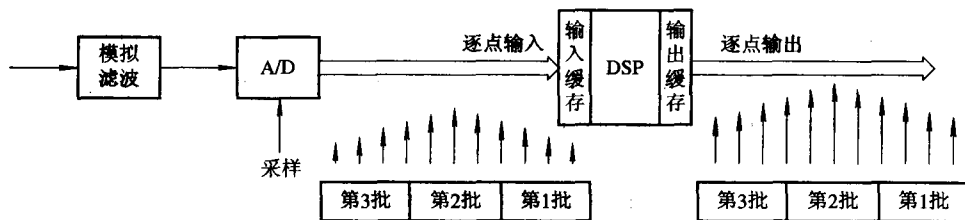


图 1.2 批处理示意图

实际中,究竟选择哪种处理形式主要是由处理要求和算法特点决定的,其次也受性价比因素的影响。吞吐率、延迟等指标反映了处理性能。当两种形式都能满足要求时,就应选择性价比高的一种。

频率检测就是一个实例。

在频率检测中,可以选择用 FFT 变换,即批处理方式,也可以选择 DFT 变换,即流水处理方式。表 1.1 分别列出了输入 64 点和 1024 点时所需的运算量。

表 1.1 FFT 和 DFT 运算量对比

处理形式 运算次数	FFT: $N\log(N)$	DFT: N^2 , 全检测	DFT: 检测 8 点
$N=64$	384	4096	512
$N=1024$	10 000	1 000 000	8192

由表 1.1 可见, 对于慢变信号且检测的频率点很多时, 批处理方式的 FFT 算法成本低; 而对于快变信号且检测的频率点较少时, 流水 DFT 处理方式性能好, 成本低。具体的应用如程控电话拨号中的 DTMF 检测。每个拨号用一个高音和一个低音表示, 高音有 4 个固定频率, 低音也有 4 个固定频率, 组合起来有 16 种。每个拨号音就是两个单频信号。这时用 DFT 算法就比 FFT 算法合适。DFT 算法可改进为 DFT 递推算法, 运算量更少, 从而实时性更好, 能更及时地检测出信号频率的变化。而普通的 FFT 算法运算量大, 不能快速检测信号频率的变化, 改为 FFT 流水算法或者滑动/重叠的 FFT 算法后, 运算量更大。

2. 按实现方法分类

从实现方法上来看, 实时信号处理又可划分为硬件实现和软件实现。

(1) 硬件实现: 如应用 FPGA/CPLD、ASIC 等进行设计。以硬件实现的系统固定、简单, 且吞吐率高。

(2) 软件实现: 如用 PC 机、DSP、单片机等的软件程序来实现。虽然该方法也离不开硬件设计, 但它主要是以软件设计为主。这种方法灵活性好, 但吞吐率比硬件实现的要低。

在通用的 PC 机上, 就是用软件实现信号处理的。Mathworks 公司的 Matlab 软件可以说是这方面成功的范例。当前, 还不断有国内外研究机构或公司推出不同的数字信号处理的软件包。基于 PC 机的实现方法简单、经济(可重复使用), 但速度太慢, 设备体积庞大, 目前只能用于演示和验证。

在单片机上用软件实现信号处理也是一种可行的方法。目前, 单片机的技术非常成熟, 但它偏重于控制而不善于运算。利用单片机的硬件环境加上信号处理软件, 可直接应用于工程实际, 如数控机床、医疗仪器设备等。

软件实现的最佳选择是用 DSP, 而硬件实现的最佳选择是用 FPGA。

3. 按实现途径分类

按实现途径可将实时信号处理分为两种。一种是完全自行开发, 即从底层软/硬件到整个系统都由科研人员自主设计。这种方法一般来说成本低, 但开发周期长、难度大。另一种是集成, 即购买某些现成的板卡和相应的操作系统、函数库、底层软件, 开发人员只需在现成的板卡上做高层软件开发并进行集成即可, 不需要进行底层设计。这种方法开发周期短、难度低, 但设备的体积大、成本高。

由于实时信号处理设备的用量远不如计算机和消费电子行业的软/硬件用量那样大, 因而造成购买的实时信号处理设备的软/硬件成本比自行开发的成本高得多, 而在计算机和消费电子行业中恰恰相反, 买来的产品价格往往比自行开发的成本低很多。

4. 按规模分类

从民用到军用, 从简单系统到复杂系统, 实时信号处理的运算量差别很大, 这使得其

规模也有很大差别。按规模可将实时信号处理分为简单系统和复杂系统。简单系统由一块电路板构成,用一片处理器作为核心,只需少量的软件或者硬件设计,这类系统的设计难度相对较低。而复杂系统必须由多片处理器、多块电路板构成,这样的多片并行系统的设计难度和调试难度远大于单处理器系统。

5. 按应用分类

按应用领域可将实时信号处理分为民用和军用。不同的要求导致设计原则大不相同。用于消费产品的实时信号处理系统在研制和生产时,设计人员特别关心其成本,追求最大的性价比,性能往往让位于成本、体积和功耗。为此,大都采用专门设计的方式,用 ASIC、专用 DSP 来降低设备的成本、体积和功耗。但在航空、航天、国防领域,设备的用量很小,追求的是性能第一,体积和功耗第二,而成本退居第三。为此,要经常采用最先进、最可靠的技术手段来实现系统的功能。

思 考 题

- 1.1 信号处理和实时信号处理的区别体现在哪些方面?
- 1.2 实时信号处理系统的设计分为哪几步?