

DSP应用丛书

TMS320C6000系列 DSPs原理与应用

(第2版)

李方慧 王 飞 何佩琨 编著
高梅国 审校



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

www.phei.com.cn

DSP 应用丛书

TMS320C6000 系列 DSPs 原理与应用(第2版)

李方慧 王 飞 何佩琨 编著
高梅国 审校

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

以高速数字信号处理器(DSPs)为基础的实时数字信号处理技术近年来发展迅速,并获得了广泛的应用。TMS320C6000 是得州仪器公司(Texas Instrument)推出的定点、浮点系列 DSPs,其中定点产品峰值处理能力达到 4800MIPS,浮点产品峰值处理能力达到 1350MFLOPS,是目前国际上性能最高的 DSPs 之一。

本书全面介绍了 C6000 系列的芯片结构、软件开发、硬件设计等内容。在第 1 版的基础上对内容作了大量的修改调整,修订了有关的技术资料,增加了对 C621x/C671x/C64x 等新芯片以及 Code Composer Studio 集成开发环境的介绍,并对 C6000 系统设计中的主要问题和技术进行了总结。

本书的读者对象是从事信号处理的科研和工程技术人员,以及通信、信息与信号处理专业的研究生、高年级本科生。

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

TMS320C6000 系列 DSPs 原理与应用/李方慧等编著. —2 版. —北京:电子工业出版社,2003.1

(DSP 应用丛书)

ISBN 7-5053-8153-9

I. T… II. ①李… ②王… ③何… III. 数字信号—信号处理—数字通信系统, TMS320C6000 IV. TN911.72

中国版本图书馆 CIP 数据核字(2002)第 087747 号

责任编辑:段 颖 特约编辑:印晓芬

印 刷:北京民族印刷厂

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:35.5 字数:859 千字

版 次:2003 年 1 月第 2 版 2003 年 1 月第 1 次印刷

印 数:4 000 册 定价:45.00 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

序

信息社会的发展,在很大程度上取决于信息与信号处理技术的先进性。数字信号处理技术的出现改变了信息与信号处理技术的整个面貌;而数字信号处理器作为数字信号处理的核心技术,其应用已经深入到涉及信号处理的航空、航天、雷达、声纳、通信、家用电器等各个领域,成为电子系统的核心。

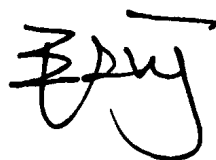
以雷达为例,数字信号处理器在性能上的每次突破都会给雷达领域带来巨大的震动,使得雷达信号处理的新算法在工程中得以实现,带来雷达整机性能的提高,甚至推动雷达新体制的实现,综合孔径雷达就是一个例子。

TI公司的TMS320C6000系列是目前数字信号处理器中性能最高的产品。最新数据表明,仅仅两年多的时间内,TMS320C6000系列数字信号处理器就已经在总价值为1.5亿美元的嵌入式产品设计中当选为数字信号处理器的解决方案,是推动数字信号处理技术发展的一个标志性产品。

本书的编写结合了作者的教学和科研实践经验,结构完整,既有对TI的TMS320C6000系列的完整介绍,也有对实时数字信号处理基本原理和数字信号处理器原理的介绍,兼顾了广大在校生的实际需求。

本书编排合理,内容切合信息产业发展的需要,对教学、生产和科研都有现实指导意义,是一本值得推荐的专著。

中国工程院院士
北京理工大学教授
2000年4月



755 91/02

前 言

TMS320C6000 系列 DSP 芯片是目前最先进、性能价格比最优的 DSP 芯片之一。本书第 1 版写于 TMS320C6000 系列第一代产品刚引进到我国不久。从本书第 1 版出版至今的两年多时间内, TMS320C6000 系列又取得了长足的发展。为适应 C6000 系列 DSP 芯片的开发需要, 我们重新编写了《TMS320C6000 系列 DSPs 原理与应用(第 2 版)》。与第 1 版比较, 第 2 版做了如下改动:

1. 对近两年来新推出的 C621x、C671x 系列以及 C64xx 系列 DSP 芯片做了全面的介绍。
2. 在软件开发方面, 以阐述 C6000 的 CCS (Code Composer Studio) 集成开发环境为主线。应用于 C6000 的 CCS 是 C6000 系列 DSPs 的主流开发工具, 目前已推出 2.1 版本。
3. 根据 TI 公司文献资料的变动, 对本书第 1 版原有内容做了大幅度的修改调整。以“TMS320C6000 CPU and Instruction Set Reference Guide”为例, 从本书第 1 版编写至今, 该文献已改了 6 版, 编号从 SPRU189A 一直修正到 SPRU189F。又如 C6000 的 C/C++ 编译器, 原文献对独立的指针变量建议采用 `const` 关键字, 而新版则要求用 `restrict` 关键字, 还特别提醒用户不要用 `const` 关键字。对于 TI 文献中的这些重大变化, 本书相应地予以反映。
4. 在内容编排上重新作了调整。如第 2 章增加了对指令集的介绍, 而把芯片内硬件资源的内容全部放到第 5 章中。第 4 章在介绍 C 程序优化前, 增加了对 C 语言程序编写和 C 编译器使用的描述。

与第 1 版一样, 本书由北京理工大学雷达技术研究所组织编写, 第 2 版由李方慧博士负责, 王飞博士、何佩琨教授参与。第 1 章、第 5 章及第 6 章第 1~3 节由李方慧博士编写; 第 3 章、第 6 章第 4 节由王飞博士编写; 第 2 章、第 4 章由何佩琨教授编写。全书由高梅国教授审校。博士研究生沈光、吴琼之、姚迪参与了资料整理工作。

本书的编写受到北京理工大学雷达技术研究所 DSP 教学和科研工作方面的支持, 反映了雷达技术研究所 C6000 芯片开发方面的技术水平。从这个意义上讲, 本书也是雷达技术研究所集体努力的结晶之一。

本书第 1 版、第 2 版的出版都得到美国得州仪器(中国)公司与北京理工大学建立的 DSPs 联合实验室的支持。从 1996 年 DSPs 联合实验室建立以来, TI 公司对推动北京理工大学 DSP 技术的开发、教学做了许多有益的工作。这也是促使本书能够编写出版的条件之一。在此, 我们对美国得州仪器(中国)公司及其大学计划表示感谢。

本书的出版得到电子工业出版社领导的大力支持, 责任编辑段颖付出了巨大努力, 我们在此表示深深的谢意。

由于编者的学术水平有限, 编写时间紧迫, 本书在取材、编写中可能还存在一些问题, 敬请读者批评指正。

编著者
2002 年 9 月

目 录

第 1 章 实时数字信号处理与数字信号处理器	(1)
1.1 实时数字信号处理	(1)
1.1.1 信号处理领域的巨大变革	(1)
1.1.2 实时信号处理的概念	(1)
1.1.3 DSPs 在实时数字信号处理中的应用	(2)
1.2 实时 DSP 系统的构成	(2)
1.2.1 实时 DSP 系统的构成	(2)
1.2.2 DSP 子系统	(2)
1.2.3 以通用 DSPs 为核心的实时 DSP 子系统	(3)
1.3 DSPs 芯片	(4)
1.3.1 DSPs 芯片概述	(4)
1.3.2 DSPs 芯片的特点	(5)
1.3.3 性能指标	(7)
1.4 TMS320C6000 系列 DSPs	(8)
1.4.1 TMS320C6000 简介	(8)
1.4.2 TMS320C6000 的结构特点	(9)
1.4.3 TMS320C6000 的应用	(11)
第 2 章 TMS320C6000 系列 CPU 结构与指令集	(12)
2.1 C6000 系列 CPU 结构	(12)
2.1.1 中央处理器 (CPU) 结构	(12)
2.1.2 CPU 数据通路与控制	(13)
2.2 TMS320C62xx/C64xx/C67xx 公共指令集	(21)
2.2.1 TMS320C6000 公共指令集概述	(22)
2.2.2 寻址方式及 Load/Store 类指令	(25)
2.2.3 算术运算类指令	(29)
2.2.4 逻辑及位域操作指令	(33)
2.2.5 搬移(MV、MOV)类指令	(34)
2.2.6 程序转移类指令	(35)
2.2.7 资源对公共指令集的限制	(36)
2.3 TMS320C67x 浮点运算指令	(38)
2.3.1 IEEE 标准的浮点数表示法	(39)
2.3.2 C67x 的浮点运算控制寄存器	(40)
2.3.3 C67x 特有的运算指令	(43)
2.3.4 C67x 指令的延迟间隙, 资源对 C67x 指令的约束	(44)

2.4	TMS320C64x 扩展的定点运算指令	(46)
2.4.1	扩展的寻址方式及 Load/Store 类指令	(47)
2.4.2	扩展的算术运算指令	(48)
2.4.3	扩展的逻辑及位域操作指令	(53)
2.4.4	数据打包与解包类指令	(54)
2.4.5	扩展的控制转移类指令	(56)
2.4.6	资源对 C64x 指令的限制	(57)
2.5	流水线	(59)
2.5.1	TMS320C6000 流水线概述	(59)
2.5.2	TMS320C6000 指令的流水线操作	(64)
2.5.3	C6000 流水线运行的几个问题	(65)
2.6	中断	(72)
2.6.1	中断类型和中断信号	(72)
2.6.2	中断服务表 (IST)	(73)
2.6.3	中断控制寄存器	(77)
2.6.4	中断性能和编程考虑事项	(79)
第 3 章	TMS320C6000 的开发环境	(84)
3.1	软件开发流程和开发工具	(84)
3.1.1	开发流程和开发工具	(84)
3.1.2	C6000 程序基本结构	(91)
3.1.3	其他代码开发工具	(110)
3.2	集成开发环境 Code Composer Studio	(110)
3.2.1	CCS 概述	(110)
3.2.2	CCS 的设置	(111)
3.2.3	CCS 项目的创建	(112)
3.2.4	源程序的编辑	(114)
3.2.5	CCS 项目的构建 (Build)	(116)
3.2.6	程序调试的基本操作	(117)
3.2.7	多处理器调试	(121)
3.2.8	断点和探测点 (Break Points and Probe Points)	(122)
3.2.9	文件输入输出与探测点	(124)
3.2.10	图形窗口	(127)
3.2.11	存储器映射 (Memory Maps)	(129)
3.2.12	变量观察窗口 (Watch Window)	(131)
3.2.13	代码剖析 (Profile)	(132)
3.2.14	GEL 语言	(134)
3.3	实时操作系统 DSP/BIOS	(140)
3.3.1	实时操作系统基本概念	(140)
3.3.2	DSP/BIOS 概述	(141)

3.3.3	程序生成	(144)
3.3.4	监测 (Instrumentation)	(152)
3.3.5	任务调度	(163)
3.3.6	输入/输出和管道	(186)
3.3.7	其他 DSP/BIOS 模块	(191)
3.3.8	其他 C6000 实时操作系统	(192)
3.4	eXpressDSP™ 算法标准	(195)
第 4 章	TMS320C6000 的软件编程及代码优化	(197)
4.1	概述	(197)
4.2	TMS320C6000 的 C 语言编程及优化	(198)
4.2.1	TMS320C6000 C/C++ 编译器简介	(198)
4.2.2	编写及编译 C 语言程序	(199)
4.2.3	优化 C 语言程序	(203)
4.3	线性汇编语言简介	(216)
4.3.1	线性汇编语句的基本结构	(216)
4.3.2	线性汇编中的伪指令	(217)
4.3.3	汇编优化器选项	(219)
4.4	通过线性汇编优化汇编代码	(220)
4.4.1	写并行代码	(220)
4.4.2	使用字访问短型数据和使用双字访问字(数据打包处理)	(224)
4.4.3	软件流水	(227)
4.4.4	多周期循环的模编排	(236)
4.4.5	循环传递路径	(247)
4.4.6	循环中的 If-Then-Else 语句	(252)
4.4.7	循环展开	(257)
4.4.8	生命太长问题(Live-Too-Long)	(263)
4.4.9	消除冗余取	(268)
4.4.10	避免存储体 (Bank) 访问冲突	(275)
4.4.11	软件流水外环	(285)
4.4.12	同内环一起条件地执行外环	(289)
4.5	C64x 编程的几点考虑	(298)
4.5.1	C64x 的打包数据处理	(298)
4.5.2	C64x 线性汇编的几点考虑	(310)
第 5 章	C6000 系列 DSPs 的集成外设与硬件开发	(315)
5.1	综述	(315)
5.2	片内存储器	(317)
5.2.1	简介	(317)
5.2.2	C620x/C670x 的片内程序存储器	(318)
5.2.3	C620x/C670x 的片内数据存储器	(321)

5.2.4	C621x/C671x/C64x 的片内 2 级存储器	(326)
5.3	外部存储器接口的设计	(337)
5.3.1	概述	(337)
5.3.2	接口信号与控制寄存器	(339)
5.3.3	同步接口设计	(347)
5.3.4	异步接口设计	(373)
5.3.5	PDT 传输接口	(377)
5.3.6	HOLD 接口	(379)
5.3.7	EMIF 访问的仲裁	(379)
5.4	直接存储器访问 (DMA)	(380)
5.4.1	概述	(380)
5.4.2	DMA 的初始化和启动	(381)
5.4.3	DMA 的传输控制	(384)
5.4.4	单一通道的分裂操作	(388)
5.4.5	资源仲裁和优先级设置	(389)
5.4.6	DMA 通道的状态	(389)
5.5	扩展的直接存储器访问 (EDMA)	(390)
5.5.1	概述	(390)
5.5.2	EDMA 的控制机制	(392)
5.5.3	EDMA 的传输操作	(397)
5.5.4	优先级	(407)
5.5.5	快速 DMA (QDMA)	(408)
5.6	多通道缓冲串口 (McBSP)	(411)
5.6.1	概述	(411)
5.6.2	信号接口和控制寄存器	(412)
5.6.3	数据的收发	(416)
5.6.4	μ -律/A-律硬件压扩	(425)
5.6.5	多通道传输接口	(428)
5.6.6	SPI 协议的接口	(433)
5.6.7	通用 I/O 口	(436)
5.7	主机口 (HPI)	(436)
5.7.1	概述	(436)
5.7.2	有关信号与控制寄存器	(437)
5.7.3	主机口的存取操作	(440)
5.7.4	HPI 的加载操作	(443)
5.8	C6000 的扩展总线	(444)
5.8.1	概述	(444)
5.8.2	信号接口与控制寄存器	(445)
5.8.3	扩展总线上的 I/O 口工作方式	(447)

5.8.4	扩展总线的主机口工作方式	(450)
5.8.5	扩展总线的仲裁	(455)
5.8.6	通过扩展总线引导 DSP 芯片	(456)
5.9	PCI 接口	(456)
5.9.1	概述	(456)
5.9.2	接口的结构与控制寄存器	(458)
5.9.3	从模式数据传输	(463)
5.9.4	主模式数据传输	(464)
5.9.5	中断与状态报告	(466)
5.9.6	PCI 的复位和加载	(469)
5.9.7	EEPROM 接口	(469)
5.9.8	电源管理 (C62x/C67x)	(473)
5.10	UTOPIA 接口	(474)
5.10.1	概述	(475)
5.10.2	接口信号与控制寄存器	(475)
5.10.3	从模式 ATM 控制器	(478)
5.10.4	UTOPIA 接口的访问控制	(482)
5.10.5	UTOPIA 时钟和时钟检测	(483)
5.10.6	UTOPIA 接口的其他控制	(483)
5.11	芯片的设置	(485)
5.11.1	概述	(485)
5.11.2	芯片的设置管脚	(486)
5.11.3	芯片的设置选项	(487)
5.11.4	芯片的引导过程	(496)
5.12	其他片内集成外设	(497)
5.12.1	定时器	(497)
5.12.2	中断控制	(500)
5.12.3	power-down 逻辑	(505)
5.12.4	GPIO	(507)
第 6 章	C6000 系列 DSPs 的应用开发	(511)
6.1	概述	(511)
6.2	C6000 系列 DSPs 的板级设计	(511)
6.2.1	电源系统	(511)
6.2.2	JTAG 控制	(516)
6.2.3	调试 (Debugging) 考虑点	(517)
6.2.4	BGA 封装	(519)
6.2.5	高速数字电路设计	(520)
6.3	DSK 套件和 EVM 板	(528)
6.3.1	DSK 套件	(528)

6.3.2 EVM 评估板	(530)
6.4 C6000 DSP 应用系统实例	(535)
6.4.1 视频信号处理系统	(535)
6.4.2 双 C6201 的并行处理系统	(535)
6.4.3 基于 PCI 总线的双处理器视频信号处理系统	(537)
6.4.4 基于 CompactPCI 总线的多处理器并行系统	(538)
附录 A TMS3200C6000 的指令集及对应的 C/C++ Compiler Intrinsics	(541)
附录 B 与存储器操作有关的 Intrinsics 及对应的汇编指令	(552)
参考文献	(553)

第 1 章 实时数字信号处理与数字信号处理器

1.1 实时数字信号处理

1.1.1 信号处理领域的巨大变革

自从 20 世纪 70 年代末第一片数字信号处理器芯片 (Digital Signal Processors, DSPs) 问世以来, DSPs 就以数字器件特有的稳定性、可重复性、可大规模集成, 特别是可编程性高和易于实现自适应处理等特点, 给数字信号处理 (Digital Signal Processing, DSP) 的发展带来了巨大机遇, 并使信号处理手段更灵活, 功能更复杂, 其应用领域也拓展到国民经济生活的各个方面。近年来, 随着半导体制造工艺的发展和计算机体系结构等方面的改进, DSPs 芯片的功能越来越强大, 使信号处理系统的研究重点又重新回到软件算法上, 而不再像过去那样过多地考虑硬件可实现性。而且随着 DSPs 运算能力的不断提高, 能够实时处理的信号带宽也大大增加, 数字信号处理的研究重点也由最初的非实时应用转向高速实时应用。

1.1.2 实时信号处理的概念

信号处理的实质是对信号进行变换, 目的是获取信号中包含的有用信息, 并用更直观的方式进行表达。数字信号处理就是用数字的方法对信号进行变换来获取有用信息, 如离散傅里叶变换 (DFT) 就是最常用的 DSP 算法。这里介绍的实时信号处理就是实时数字信号处理。

实时指的是系统必须在有限的时间内对外部输入信号完成指定的处理, 即信号处理的速度必须大于等于输入信号更新的速度 (常见信号的典型数据率如图 1-1 所示), 而且从信号输入到处理后输出的延迟必须足够小, 如一个制导系统的输出延迟就要求在几毫秒以内。

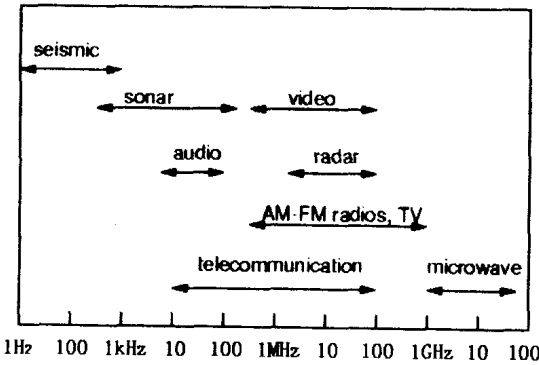


图 1-1 常见信号的典型数据率

从图 1-1 可以看到,不同类型的信号所要求的实时信号处理速度相差很大。如对一个音频信号,假设用 40 kHz 时钟采样,样本数据字长 16 bit,则该信号的输入数据率就是 80 kB/s,它对实时处理速度的要求是大于等于 80 kB/s。对于一个每帧数据字长 $512 \times 512 \times 16 \text{ bit}$ 、传输速率为 30 帧/秒的图像信号,其输入数据率是 15 MB/s,因而它对实时处理速度的要求是大于等于 15 MB/s。由此可知,对实时信号处理速度的要求与原始模拟信号带宽以及数据格式(字长、维数)等因素是密切相关的。

目前单片 DSPs 的处理能力已达到每秒 48 亿条指令(4800MIPS)和每秒 10 亿次浮点操作(1GFLPOS)的水平,使实时信号处理的应用空间越来越广阔。

1.1.3 DSPs 在实时数字信号处理中的应用

自从 20 世纪 80 年代初 DSPs 投入市场以来,实时 DSP 技术在国民经济和社会生活的各个方面得到了广泛的应用。特别是随着信息技术的发展和互联网的普及,机顶盒(Top Set Box)、网络电话(Internet Phone)以及个人数字助理(Personal Digital Assistant)等信息家电(Information Apparatus)的发展如雨后春笋。实时 DSP 技术有了更广阔的消费品市场,其发展又有了一次空前的机遇。

信号处理的复杂性和通信协议的发展更新快决定了以可编程的 DSPs 芯片为核心组成的应用系统具有以下优点:

- ① 能够快速制造原理样机和进行验证,加快产品上市时间。
- ② 高度可编程性使产品能够迅速应用新算法、新标准或新协议。
- ③ 可以通过软件更新,快速地进行产品升级。

1.2 实时 DSP 系统的构成

1.2.1 实时 DSP 系统的构成

实时信号处理系统所要处理的信号多为自然信号,因此首先需要通过传感器将自然信号转换为电信号。另外要对自然界的信号进行数字处理,就必须通过 A/D 子系统将其转换为数字形式。DSP 子系统对数字信号处理完成后,有时还需要通过 D/A 子系统把处理后的数字信号重新转换为模拟信号。图 1-2 是一个完整的实时 DSP 系统框图,其中, DSP 子系统是整个系统的核心。下面主要介绍 DSP 子系统的实现和构成。



图 1-2 实时 DSP 系统

1.2.2 DSP 子系统

在当前技术条件下, DSP 子系统一般有 6 种实现方法:

- ① 在通用计算机上用软件实现。
- ② 在通用计算机系统中加入专用的加速处理模块。

③ 利用通用单片机（用于数字控制等不太复杂的数字信号处理，如 Intel 的 MCS51 系列）。

④ 利用通用可编程 DSPs 芯片。

⑤ 利用专用 DSPs 芯片。

⑥ 利用基于通用 DSPs 内核的 ASIC（在用量较大的通信、硬盘控制等领域，一些 DSPs 厂商提供了一种基于通用 DSPs 核的 ASIC 设计和生产服务，即用户可以在通用 DSPs 的 CPU 基础上选用所需要的外设接口和存储器等资源，并在片内固化所需软件）。

上述方法中，第 1 种方法的缺点是速度相对较慢，不适合于实时 DSP，一般只用于 DSP 算法的模拟；第 2 种不适合于嵌入式应用，而且和第 5 种方法一样具有专用性较强的特点，应用受到很大限制；第 3 种不适合于以乘加运算为主的运算密集型 DSP 算法；第 4 种通用可编程 DSPs 芯片，由于其可编程性和强大的处理能力，在实时 DSP 领域居于主导地位。另外在批量应用中，基于通用 DSPs 核的 ASIC 由于其较好的系统性价比在近几年得到了广泛应用。

1.2.3 以通用 DSPs 为核心的实时 DSP 子系统

基于 DSPs 的 DSP 子系统一般由控制处理器（其控制功能可由通用微处理器或 DSPs 实现）、DSPs、数据传输网、存储器和输入/输出接口构成。其结构框图如图 1-3 所示。

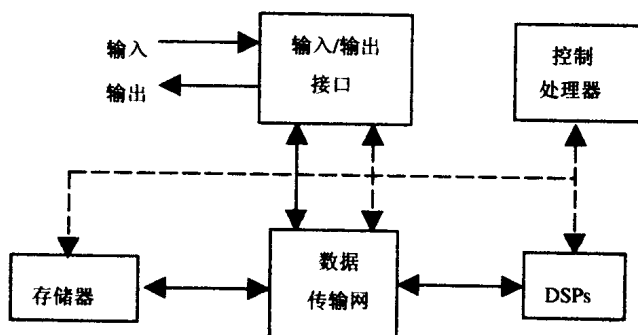


图 1-3 基于 DSPs 的 DSP 子系统结构

- **控制处理器** 完成系统控制功能，包括主机命令解释、数据传输控制和数据输入/输出等控制功能。控制处理器可以使 DSPs 专注于高速实时 DSP 算法的实现。根据具体的系统不同，控制处理器可以用通用微处理器或 DSPs 独立实现，也可以放在图 1-3 中的 DSPs 内实现。

- **DSPs** 完成实时信号处理算法。

- **数据传输网** 实现各个模块之间的互连，以传输数据。对于模块之间有大量数据传输的高速 DSP 系统设计，数据传输网的设计是一个关键环节，它很可能成为系统的瓶颈。

- **存储器** 支持数据存储。它的主要参数指标有存储器容量、存储器字长、访问速度、对特殊寻址方式的支持和存储器管理控制能力。

- **输入/输出接口** 用于输入待处理的数据或输出处理结果。输入/输出接口的主要参数有接口带宽、缓冲存储能力、数据字长、接口规程和接口所支持的输入输出个数。

上述 DSP 子系统, 对具体应用来说其复杂程度会有很大的差异。其硬件平台可能是以单个 DSPs 为核心的一个板卡, 也可能是围绕多个 DSPs 组成的一个分布式 DSP 系统。同时由于所选用的 DSPs 速度不同, 以及软件 DSP 算法等方面的差异, 其设计和实现难度也千差万别。

1.3 DSPs 芯片

1.3.1 DSPs 芯片概述

1.3.1.1 DSPs 芯片的出现和发展

在 DSPs 出现之前, 实时信号处理一般是在通用处理器 (8086 和 80286 等) 中完成的。随着集成电路制造工艺的不断提高, 20 世纪 70 年代末出现了专门的可编程数字信号处理器, 简称 DSPs。第一代 DSPs 以 AMD2900、NEC7720 和 TMS32010 为代表, 其中 TI 公司的 TMS32010 第一次使用了哈佛总线结构和硬件乘法器。

由于开发工具的问题, 最初的 DSPs 开发非常困难, 要设计并实现一个基于 DSPs 的系统是一个专业性很强的工作。美国 TI 公司给 DSPs 引入了许多通用计算机微处理器的特点, 并为其产品开发了汇编语言和 C 语言代码产生工具以及各种软硬件调试工具, 使得 DSPs 的开发难度大大降低, 并且在 20 世纪 80 年代末和 90 年代初进入了快速发展的时期。现在 TI 公司的 DSPs 包括了定点、浮点和多处理器 3 个类型的产品, 每个类型又有不同性能和价格的具体系列可以供用户选择。

1.3.1.2 DSPs 芯片的分类

1. DSPs 芯片的分类

按照所支持的数据类型不同, DSPs 分为定点产品和浮点产品 2 大类。

定点 DSPs 进行算术操作时, 使用的是小数点位置固定的有符号数或无符号数。浮点 DSPs 进行算术操作时, 使用的是带有指数的小数, 小数点的位置随着具体数据的不同进行浮动。

定点器件在硬件结构上比浮点器件简单, 具有价格低和速度快的特点, 因而应用得最多; 而浮点器件的优点是精度高, 不需要进行定标和考虑有限字长效应, 但是其成本和功耗相对较高, 速度较慢, 适合于对数据动态范围和精度要求高的特殊应用。

除了定点和浮点的划分外, 各个 DSPs 厂家还根据 DSPs 的 CPU 结构和性能, 把自己的产品划分了不同系列。如 TI 公司的定点系列 DSPs 有 C20x、C24x、C5x、C54x 和 C62xx; 浮点系列 DSPs 有 C3x、C4x 和 C67xx。不同系列 DSPs 的 CPU 结构不同, 性能和价格也有很大的差异。

同一系列 DSPs 产品中, 各个不同型号的 DSPs 在 CPU 结构上基本相同, 不同之处只在于 DSPs 片内存储器和外设接口的配置不同。如 TI 公司的 C5x 系列中的 C50 和 C52, 它们的 CPU 结构完全相同, 不同之处在于 C50 的片内 ROM 为 2K 字, 片内 RAM 为 10K 字, 具有标准串口和 TDM 串口各 1 个; 而 C52 的片内 ROM 为 4K 字, 片内 RAM 为 1K

字, 只有 1 个标准串口。由于同一系列不同型号产品具有丰富多样的外设接口和存储器配置, 因此可以在不同的应用场合最大程度地减少 DSPs 的外围器件, 缩小电路板面积, 从而提高系统性价比。

2. DSPs 的代码兼容问题

所谓代码兼容, 是指为某种 DSPs 产品开发的代码可以在不加修改或只作很小修改的情况下在其他 DSPs 上执行, 称为完全兼容或部分兼容。不同厂商在生产自己的 DSPs 产品时, 所采用的代码兼容策略是不一样的。但是基本上每一个 DSPs 厂家同一系列 DSPs 产品中不同型号之间是代码兼容的。有的 DSPs 厂家还尽量保持不同系列 DSPs 产品之间的兼容性, 如 TI 公司的定点系列 DSPs 的 C1x、C2x 和 C5x 之间是向下兼容的, C1x 和 C2x 的代码可以不加修改地在 C5x 上运行。另外, TI 的 C6000 的定点系列 C62xx 的代码可以不加修改地在浮点系列 C67xx 上运行。

除了代码兼容之外, 还有所谓管脚兼容。如 TI 的 C62xx 与 C67xx 的某些片种之间就保持了管脚兼容, 使系统设计更加灵活。

1.3.2 DSPs 芯片的特点

下面介绍 DSPs 与数字信号处理功能相适应的特点。

1.3.2.1 功能特点

数字信号处理任务通常需要完成大量的实时计算, 如在 DSP 中常用的 FIR 滤波和 FFT 算法。数字信号处理中的数据操作具有高度重复的特点, 特别是乘加操作 $Y=A*B+C$ 在滤波、卷积和 FFT 等常见 DSP 算法中用得最多。DSPs 在很大程度上就是针对上述运算特点设计的。与通用微处理器相比, DSPs 在寻址和计算能力等方面作了扩充和增强。在相同的时钟频率和芯片集成度下, DSPs 完成 FFT 算法的速度比通用微处理器要快 2 到 3 个数量级 (如对于 1024 点的 FFT 算法, 时钟相同, 集成度相仿的 IBM PC/AT-386 和 TMS320C30, 运算时间分别为 0.3 s 和 1.5 ms, 速度相差 200 倍)。

1.3.2.2 结构特点

DSPs 的结构特点在很大程度上体现了 DSP 算法的需求。下面介绍 DSPs 在结构上的主要特点。

1. 算术单元

(1) 硬件乘法器

由于 DSPs 的功能特点, 乘法操作是 DSPs 的一个主要任务。而在通用微处理器内通过微程序实现的乘法操作往往需要 100 多个时钟周期, 非常费时, 因此在 DSPs 内都设有硬件乘法器来完成乘法操作, 以提高乘法速度。硬件乘法器是 DSPs 区别于通用微处理器的一个重要标志。

(2) 多功能单元

为进一步提高速度, 可以在 CPU 内设置多个并行操作的功能单元 (ALU、乘法器和地址产生器等)。如 C6000 的 CPU 内部有 8 个功能单元, 包括 2 个乘法器和 6 个 ALU。这 8 个功能单元最多可以在 1 个周期内同时执行 8 条 32 位指令。由于多功能单元的并行

操作使 DSPs 在相同时间内能够完成更多的操作，因而提高了程序的执行速度。

针对乘加运算，多数 DSPs 的乘法器和 ALU 都支持在 1 个周期内同时完成 1 次乘法和 1 次加法操作。另外很多定点 DSPs 还支持在不增加操作时间的前提下对操作数或操作结果的任意位移。

另外，DSP 的算法特点和数据流特点还可以使现代 DSPs 采用指令比较整齐划一的精简指令集（RISC），有利于 DSPs 结构的简化和成本的降低。

2. 总线结构

通用微处理器是为计算机设计的。基于成本上的考虑，传统的微处理器通常采用冯·诺曼总线结构：统一的程序和数据空间，共享的程序和数据总线。由于总线的限制，微处理器执行指令时，取指和存取操作数共享内部总线，因而程序指令只能串行执行。

对于面向数据密集型算法的 DSPs 而言，冯·诺曼总线结构使系统性能受到很大限制，因此 DSPs 采用了程序总线 and 数据总线分离的哈佛总线结构，这样 DSPs 就能够同时取指和取操作数了。而且很多 DSPs 甚至有 2 套或 2 套以上的内部数据总线，这种总线结构称为修正的哈佛结构。对于乘法或加法等运算，1 条指令要从存储器中取 2 个操作数，如果采用多套数据总线就可以同时取得 2 个操作数，因此提高了程序效率。

C6000 系列 DSPs 则采用了新的 VILW（甚长指令字）结构，片内提供 8 个独立的运算单元、256 位的程序总线、2 套 32 位数据总线和 1 套 32 位的 DMA 专用总线。灵活的总线结构大大缓解了数据瓶颈对系统性能的限制。

3. 专用寻址单元

DSPs 面向的是数据密集型应用，随着频繁的数据访问，数据地址的计算时间也线性增长。如果不在地址计算上作特殊考虑，有时计算地址的时间比实际的算术操作时间还长。例如，8086 做一次加法需要 3 个时钟周期，但是计算一次地址却需要 5~12 个时钟周期。因此，DSPs 通常都有支持地址计算的算术单元——地址产生器。地址产生器与 ALU 并行工作，因此地址的计算不再额外占用 CPU 时间。由于有些算法通常需要一次从存储器中取 2 个操作数，所示 DSPs 内的地址产生器一般也有 2 个。

DSPs 的地址产生器一般都支持间接寻址，而且有些 DSPs 还能够支持位反转寻址（用于 FFT 算法）和循环寻址，如 C6000 就支持循环寻址。

4. 片内存储器

由于 DSPs 面向的是数据密集型应用，因此存储器访问速度对处理器的性能影响很大。现代微处理器内部一般都集成有高速缓存器（cache），但是片内一般不设存储程序的 ROM 和存储数据的 RAM。这是因为通用微处理器的程序一般都很大，片内存储器不会给处理器性能带来明显改善。而 DSP 算法的特点是需要大量的简单计算，相应地其程序就比较短小，存放在 DSPs 片内就可以减少指令的传输时间，并有效缓解芯片外部总线接口的压力。除了片内程序存储器外，DSPs 内一般还集成有数据 RAM，用于存放参数和数据。片内数据存储器不存在外部存储器的总线竞争问题和访问速度不匹配问题，因此访问速度快，可以缓解 DSPs 的数据瓶颈，充分利用 DSPs 强大的处理能力。C6000 系列 DSPs 内部集成有 1~8 Mbit 的程序 RAM 和数据 RAM，对有些片种，这些存储器还可以配置为程序 Cache 或数据 Cache 来使用。