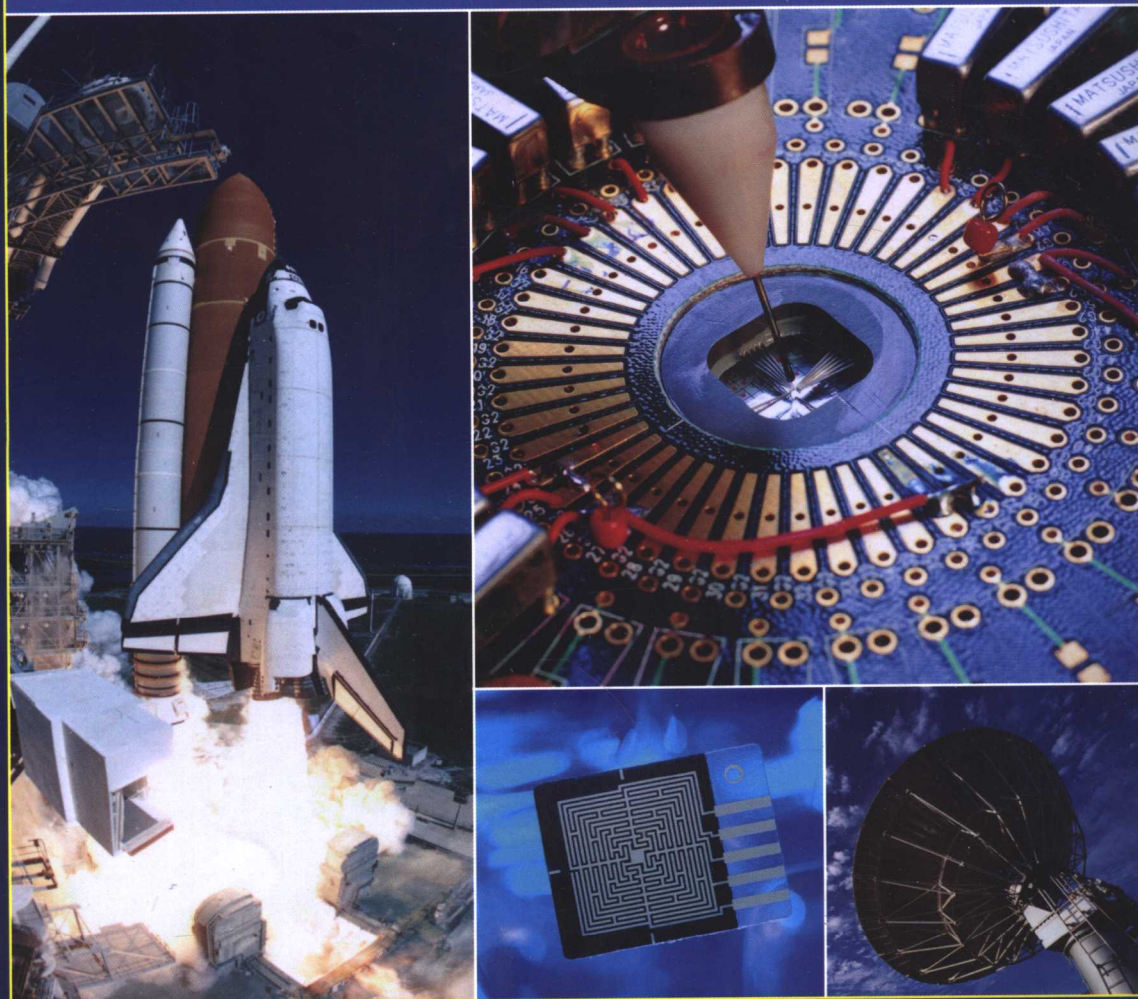


高性能工业控制DSP

——ADSP-2199x原理及应用

王晓明 庄喜润 孙维涛 崔建 编著

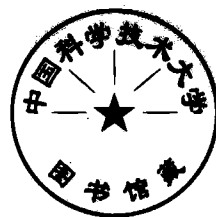


北京航空航天大学出版社



高性能工业控制 DSP—— ADSP-2199x 原理及应用

王晓明 庄喜润 编著
孙维涛 崔建



北京航空航天大学出版社

内 容 简 介

ADSP-2199x 是 ADI 公司最新推出的用于工业控制的 DSP,它是这个领域里目前世界上速度最快的 16 位定点 DSP。能够想像,每 6.25 ns 就可以进行一次乘加(减)运算,那么什么样先进的控制算法不能在这块芯片上实现呢?本书详细、系统地介绍了这种 DSP 的结构、内核功能、独特的类似于高级语言的汇编指令集、语法以及外围各模块的功能和使用方法。

本书可作为在电动机控制、机器人控制、工业过程控制、设备控制、工业检测和智能系统等领域从事开发与研究的工程技术人员、高校教师、研究生和本科生的自学用书,也可供对工业控制 DSP 感兴趣的读者参考。

图书在版编目(CIP)数据

高性能工业控制 DSP: ADSP-2199x 原理及应用 / 王晓
明等编著. — 北京: 北京航空航天大学, 2005. 10
ISBN 7-81077-722-X

I. 高… II. 王… III. 数字信号—信号处理
IV. TN911.72

中国版本图书馆 CIP 数据核字(2005)第 089815 号

高性能工业控制 DSP——

ADSP-2199x 原理及应用

王晓明 庄喜润 孙维涛 崔建 编著
责任编辑 王慕冰 朱胜军

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail: bhpres@263.net

北京市松源印刷有限公司印装 各地书店经销

*

开本: 787×1092 1/16 印张: 23.25 字数: 595 千字

2005 年 10 月第 1 版 2005 年 10 月第 1 次印刷 印数: 5 000 册

ISBN 7-81077-722-X 定价: 39.00 元(含光盘 1 张)

前 言

如今,飞速发展的现代工业控制技术正在向智能化、网络化和集成化方向发展,要求控制器和处理器应该具有更高的处理速度、更快的实时性、更多的接口功能、更强的通信能力、更好的电磁兼容性、更低成本、支持低功耗和多任务系统。ADSP-2199x 系列数字信号处理器(Digital Signal Processor, 缩写为 DSP)的高性能能够满足现代工业控制技术的要求。

ADSP-2199x 系列 DSP 采用改进的哈佛结构和 6 级指令流水线,指令执行速度最高可达 160 MIPS,再加上 DMA 后台数据传送功能,堪称目前世界上速度最快的用于工业控制的 16 位定点 DSP。

它的指令系统和汇编语法结构与高级语言极为相似,因此容易学习,可读性好。当然,它也支持 C 语言。

内部有 12K~52K(可选)程序存储器、数据存储器 and 引导 ROM。支持并口和 SPI 同步串口,实现外部扩展。也可通过内部的 SPORT 模块或 CAN 模块与外部器件进行通信,实现网络功能和多主机功能。

集成了 PWM 模块、辅助 PWM 模块和编码器接口,可使 DSP 轻松地控制各种电动机。

高精度(14 位)和快速的 A/D 转换器提供了与外部模拟信号的接口。8 个通道完成一次转换最快所需时间是 725 ns。

2.5 V 内核工作电压、3.3 V I/O 工作电压以及 4 种低功耗模式可实现节能目的。看门狗和电源管理模块可保证安全性。

全书共有 11 章和 3 个附录。第 1 章概括地介绍了这种 DSP 的主要功能;第 2 章给出了计算单元的结构和使用方法;第 3 章描述了引脚功能、复位、引导、时钟设置、低功耗、I/O 标志和看门狗;第 4 章是存储器、数据地址发生器和外部接口;第 5 章讲述了程序控制器、程序结构与中断;第 6 章是指令系统;第 7 章展示了 DMA 的独特功能与用法;第 8 章介绍了定时器、PWM 模块和编码器接口;第 9 章是 A/D 转换器;第 10 章是关于 SPORT、SPI、CAN 串行接口;第 11 章给出了应用程序例子。附录 A 是开发工具 VisualDSP++ 3.5 的使用介绍;附录 B 是 ADSP-2199x DSP 封装图;附录 C 是光盘文件。

在本书的编写过程中,一直得到了 ADI 公司李川先生的热心帮助和支持,在此表示衷心的感谢。

本书大部分内容是作者本人在德国工作期间完成的,借此机会,感谢德国 Clausthal 大学能源技术研究所(IEE)的 Hans-Peter Beck 教授、Heiko Stichweh 博士、Ernst-August Wehrmann 博士、Dirk Turschner 博士、Elke Mendt 女士。他们在我工作期间给了很多帮助和支持,并使我在德国度过了非常愉快、令人永生难忘的日子。

除了庄喜润、孙维涛、崔建参与了本书的编写工作外,研究生杨秀艳、陈研、董玉林也参与了本书的资料收集、整理、图形绘制、文字输入和校对工作;另外,对本书的编写提供帮助和支持的还有:佟绍成、马新、谭文东、李卫民、常国威、鲁宝春、王天利、陈勇、卫绍元、王艳秋、李国义、李铁军、齐世武、李成英,在此一并表示谢意。

最后,还要感谢北京航空航天大学出版社对本书的出版所做的贡献。

由于作者水平有限,加之时间紧迫,书中难免有错误和不完善之处,敬请读者批评指正。

联系方式:

motor-nc@126.com

motor-nc@sohu.com

王晓明
于辽宁工学院
2005年8月

目 录

第1章 概 述

1.1 ADSP-2199x 的特点	1
1.2 ADSP-2199x 的结构	4
1.2.1 DSP 的内核结构	4
1.2.2 DSP 的外围结构	5
1.2.3 存储器结构	8
1.2.4 其他功能	10
1.3 开发工具	11

第2章 计算单元

2.1 概 述	12
2.2 使用的数据格式和数据类型	13
2.2.1 数据格式	14
2.2.2 数据类型	14
2.2.3 格式总结	16
2.3 计算单元的模式设置	17
2.4 算术逻辑单元 ALU	19
2.4.1 ALU 的操作	20
2.4.2 ALU 状态标志	20
2.4.3 ALU 数据结构框图	21
2.4.4 ALU 除法	22
2.5 乘法器 MAC	24
2.5.1 乘法器的操作	24
2.5.2 乘法器的清零、舍入及饱和操作	26
2.5.3 乘法器的状态标志	26
2.5.4 乘法器的溢出和饱和	27
2.6 移位器	27
2.6.1 移位器的操作	27
2.6.2 块指数的提取	31
2.6.3 立即移位	32
2.6.4 反规格化	33
2.6.5 规格化	34

2.6.6 移位器状态标志	37
---------------	----

第3章 系统信号

3.1 引脚功能	38
3.1.1 引脚功能	38
3.1.2 引脚复位状态	41
3.2 DSP 的复位	45
3.2.1 硬复位	45
3.2.2 软件复位	46
3.3 引导	46
3.3.1 引导模式	46
3.3.2 引导流格式	48
3.4 DSP 的时钟管理	49
3.4.1 锁相环 PLL	50
3.4.2 时钟控制模块	51
3.5 低功耗	55
3.6 总线请求和响应信号	56
3.7 看门狗	58
3.8 FIO 模块	59
3.8.1 FIO 寄存器的设置	59
3.8.2 FIO 模块的操作	62

第4章 存储器、数据地址发生器和外部接口

4.1 存储器	65
4.1.1 内部存储器地址和数据总线	65
4.1.2 外部存储器地址和数据总线	67
4.1.3 内部数据总线交换	67
4.1.4 ADSP-2199x 存储器组织结构	68
4.1.5 写缓存 FIFO	71
4.2 数据地址发生器	71
4.2.1 DAG 模式的设置	72
4.2.2 DAG 操作	75
4.2.3 DAG 寄存器的延迟	80
4.3 外部接口	81
4.3.1 外部存储器接口寄存器	81
4.3.2 外部接口模式的设置	85
4.3.3 外部接口模式的使用	87
4.3.4 与外部存储器接口	90
4.3.5 编程举例	96

第 5 章 程序控制器

5.1 程序控制器功能	99
5.2 指令流水线	101
5.3 指令缓存	102
5.3.1 缓存的使用	103
5.3.2 指令缓存的优化	104
5.4 分支结构	105
5.5 循 环	108
5.6 中 断	109
5.7 堆 栈	112
5.8 外设中断	115
5.8.1 ADSP-2199x 外设中断源	115
5.8.2 一般性操作	117

第 6 章 ADSP-219x 指令集

6.1 ALU 指令	120
6.1.1 相关内容	120
6.1.2 ALU 指令的格式与功能	121
6.2 MAC 指令	129
6.2.1 相关内容	129
6.2.2 MAC 指令的格式与功能	130
6.3 移位器指令	134
6.3.1 移位器指令选项	134
6.3.2 移位器指令的格式和功能	134
6.4 多功能指令	139
6.4.1 相关内容	139
6.4.2 多功能指令的格式和功能	140
6.5 数据移动指令	144
6.5.1 相关内容	144
6.5.2 数据移动指令的格式与功能	146
6.6 程序流指令	154
6.6.1 相关内容	154
6.6.2 程序流指令的格式与功能	156
6.7 ADSP-219x 伪指令	165
6.8 ADSP-219x 指令和伪指令检索表	175

第 7 章 DMA

7.1 I/O 处理器的结构和 DMA 分类	180
------------------------------	-----

7.1.1 I/O 处理器的结构	180
7.1.2 基于描述符的 DMA 传送	181
7.1.3 基于自动缓冲的 DMA 传送	182
7.1.4 DMA 中断	183
7.2 存储器与存储器间 DMA 数据传送寄存器	183
7.3 DMA 的设置	186
7.3.1 存储器到存储器 DMA 设置	187
7.3.2 SPORT 串口 DMA 设置	187
7.3.3 SPI 串口 DMA 设置	188
7.4 DMA 的使用	188
7.4.1 存储器 DMA 的使用	188
7.4.2 SPORT 串口 DMA 的使用	188
7.4.3 SPI 串口 DMA 的使用	190
7.4.4 DMA 引导	193
7.5 存储器到存储器 DMA 数据传送程序示例	193

第 8 章 PWM

8.1 定时器	197
8.1.1 定时器的结构	197
8.1.2 定时器的寄存器	198
8.1.3 定时器的操作	198
8.1.4 定时器的工作模式	200
8.1.5 程序举例	204
8.2 辅助 PWM 单元	209
8.2.1 辅助 PWM 的操作	209
8.2.2 辅助 PWM 的操作模式	211
8.2.3 辅助 PWM 的时序特性	212
8.3 PWM 模块	213
8.3.1 PWM 模块结构	213
8.3.2 PWM 模块寄存器	215
8.3.3 PWM 模块功能设置	215
8.3.4 电动机控制的特殊模式	226
8.4 编码器接口	230
8.4.1 编码器的结构	230
8.4.2 编码器寄存器	232
8.4.3 编码器接口的操作	233

第 9 章 A/D 转换器

9.1 A/D 转换器的内部结构和参考电压	245
-----------------------------	-----

9.1.1 内部结构	245
9.1.2 参考电压	246
9.2 A/D 转换器寄存器	248
9.3 A/D 转换器的操作	252
9.3.1 A/D 转换器时钟与数据格式	252
9.3.2 A/D 转换器启动方式和 A/D 转换器计数器	253
9.3.3 A/D 转换器转换模式	254

第 10 章 串行口 SPORT、SPI 和 CAN 模块

10.1 SPORT 同步串行口	258
10.1.1 SPORT 口结构	259
10.1.2 发送和接收配置寄存器	261
10.1.3 发送和接收缓冲寄存器	267
10.1.4 串行时钟和帧同步信号的频率	267
10.1.5 状态寄存器	269
10.1.6 多通道操作	269
10.1.7 SPORT 模式组合时序示例	274
10.1.8 SPORT 口的 DMA 数据传送	277
10.2 串行外设接口 SPI	281
10.2.1 SPI 接口的结构	282
10.2.2 SPI 寄存器	283
10.2.3 SPI 的操作	290
10.2.4 SPI 错误信息的处理	293
10.3 CAN 模块	295
10.3.1 CAN 模块寄存器	295
10.3.2 CAN 配置寄存器	301
10.3.3 CAN 的数据存储	304
10.3.4 邮箱控制逻辑	307
10.3.5 邮箱接收逻辑	308
10.3.6 邮箱发送逻辑	311
10.3.7 邮箱中断	315
10.3.8 全局中断	316
10.3.9 通用计数器单元	320
10.3.10 错误寄存器	322

第 11 章 应用举例

11.1 I/O 口及定时器应用举例	323
11.2 ADSP-2199x 与 PC 机的串行通信举例	328
11.3 ADSP-21992 CAN 控制器的使用方法	332

附录 A 开发工具 VisualDSP++ 3.5 使用介绍

A.1 VisualDSP++ 3.5 概述	336
A.2 集成开发环境的使用	336
A.2.1 创建一个新的工程文件	337
A.2.2 设置工程选项	338
A.2.3 编辑和添加工程源文件	343
A.2.4 编译链接 Debug 版的工程,输出可执行文件.dxe	343
A.2.5 建立 Debug Session 和加载可执行文件	343
A.2.6 运行和调试 Debug 程序	344
A.2.7 编译链接 Release 版本的工程	345
A.3 程序性能分析操作	345
A.3.1 Trace	345
A.3.2 Profile	346
A.4 调 试	348
A.4.1 设置断点	348
A.4.2 设置 Watch Point	348
A.4.3 仿真硬件环境	349
A.4.4 寄存器窗口操作	351
A.4.5 存储器窗口操作	351
A.4.6 其他窗口操作	353
A.4.7 画图 Plot	354

附录 B ADSP-2199x DSP 封装图**附录 C 光盘文件****参考文献**

第 1 章 概 述

ADSP-2199x 系列 DSP 是 ADI 公司最新推出的工业控制用 DSP。它可应用于电动机控制、机器人控制、工业过程控制、设备控制、工业检测、智能系统、便携式仪器、智能传感器、电源控制等领域。

在电动机控制方面,与以前推出的 ADMC 系列 DSP 相比,其速度由最快 26 MHz 提高到 160 MHz,是目前世界上速度最快的工业控制用 16 位定点 DSP。

ADI 公司的 DSP 采用类似高级语言的汇编指令集,例如循环指令,可免去传统的计数器赋值→计数器减 1→条件检测→跳转循环这样的汇编结构;而多功能指令可在一个指令周期里完成多项任务,因此更容易学习和理解。与其他汇编语言相比,可读性要好得多。

ADSP-2199x 系列 DSP 目前的产品型号、工作速度、存储器容量、工作电压和封装形式见表 1-1。

表 1-1 ADSP-2199x 系列 DSP

型 号	速度/ MHz	存储器/字			核/I/O 工作电压/V	引脚数/封装
		DM RAM	PM RAM	PM ROM		
ADSP-21990BST	160	4K	4K	4K	2.5/3.3	176-LQFP
ADSP-21990BBC	150	4K	4K	4K	2.5/3.3	196-MBGA
ADSP-21991BST	160	8K	32K	4K	2.5/3.3	176-LQFP
ADSP-21991BBC	150	8K	32K	4K	2.5/3.3	196-MBGA
ADSP-21992BST	160	16K	32K	4K	2.5/3.3	176-LQFP
ADSP-21992BBC	150	16K	32K	4K	2.5/3.3	196-MBGA
ADSP-21992YST	100	16K	32K	4K	2.5/3.3	176-LQFP
ADSP-21992YBC	150	16K	32K	4K	2.5/3.3	196-MBGA

注: B=工业级温度范围(-40~+85℃);

Y=特级温度范围(-40~+125℃);

MBGA 封装=Mini Ball Grid Array(15 mm×15 mm);

LQFP 封装=Low-profile Quad Flat Pack(24 mm×24 mm)。

1.1 ADSP-2199x 的特点

ADSP-2199x 系列 DSP 是采用 ADSP-219x 内核的混合信号处理器。它适于各种高性能的电动机控制和工业信号处理应用。高性能的 DSP 内核和嵌入式混合信号外围的集成,以及与诸如 CAN 和 SPI 通信接口的结合,为这些应用提供了可靠的保证。

ADSP-2199x 可提供 160 MIPS 的指令执行速度,集成了常见的串行接口、1 个 DMA 控制

器、3 个可编程定时器、通用可编程 I/O 引脚、片内程序和数据存储器、一整套高度集成的用于快速电动机控制和信号处理的嵌入式控制外围模块。

使用 ADSP-219x 高性能的内核与高性能的总线 PM、DM 和 DMA 相配合,保证了每个周期都能执行一条计算指令。总线和指令缓存确保到内核的数据流快速无阻、高速执行。

片内 3 条总线见图 1-1。它们是程序总线 PM、数据总线 DM 和 DMA 总线。

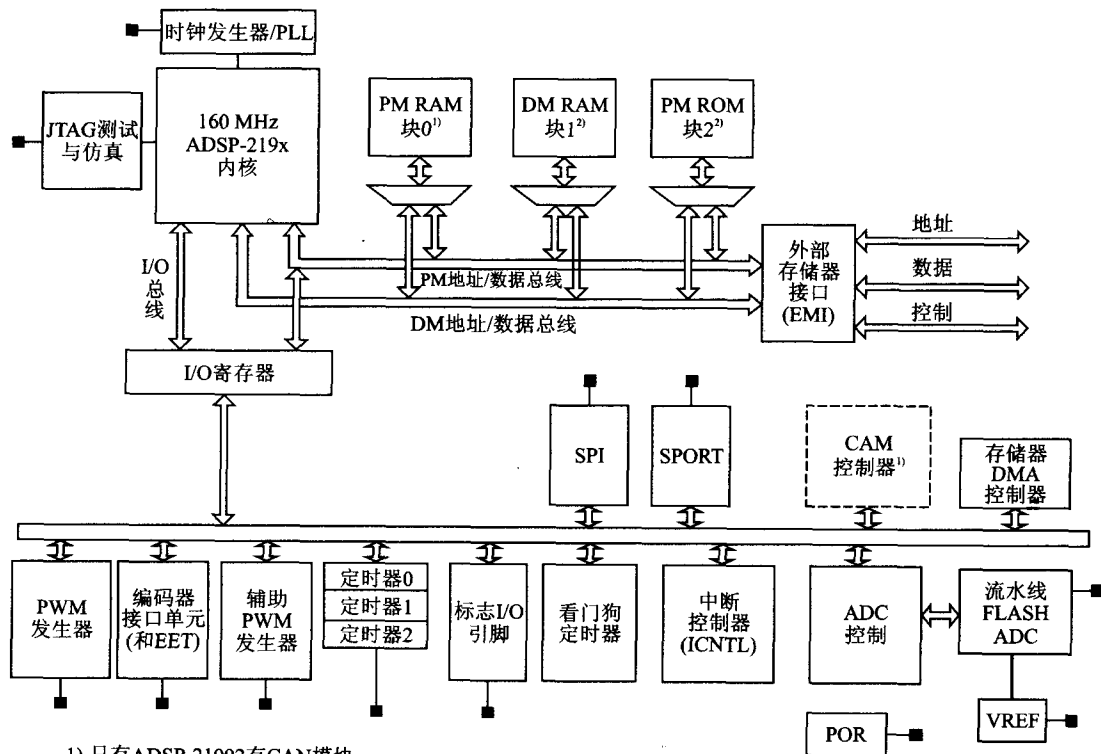


图 1-1 ADSP-2199x DSP 功能框图

程序总线 PM 既可访问指令,也可访问程序中的数据。在一个周期内,这些总线可使控制器同时访问两个操作数(一个来自于 PM,另一个来自于 DM)和访问一次指令(来自于指令缓存)。

外部总线为处理器提供了与外部存储器、外部 I/O、外部引导存储器数据传送与控制的接口。外部接口起到了总线仲裁、控制信号共享、扩展存储器和 I/O 的作用。

图 1-2 给出了处理器内核的详细结构。

- 计算单元: 包括乘法器、ALU、移位器和数据寄存器阵列。
- 程序顺序控制器: 包括指令缓存、定时器和数据地址发生器(DAG1 和 DAG2)。
- 2 块 SRAM。
- 与片外存储器、外围和主机的接口。
- 1 个 SPORT 串行接口、1 个 SPI 串行接口和 1 个 CAN 总线模块。
- 混合信号和嵌入式控制外围: A/D 转换器、编码器接口模块、PWM 模块。
- JTAG 接口。

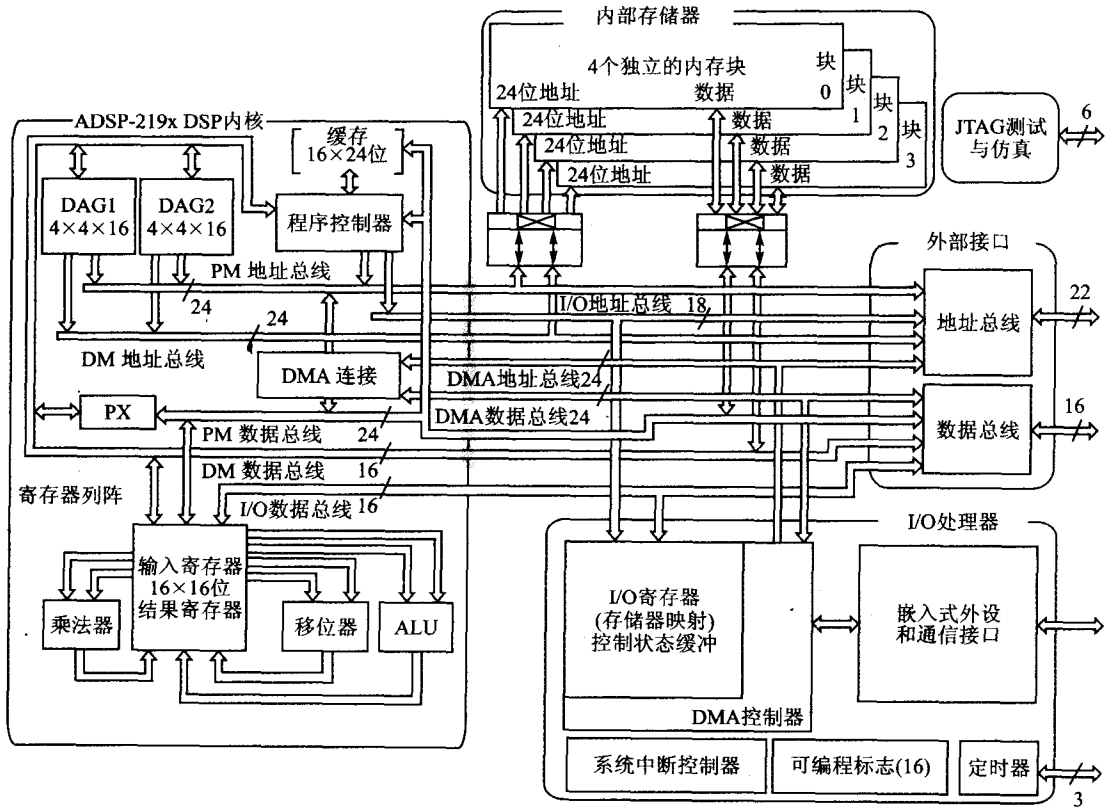


图 1-2 DSP 内核结构

ADSP-2199x 从以下 5 个方面满足需求：

- 快速灵活的算法计算单元。所有计算指令执行均使用 1 个周期,从而保证快速运算和一系列算法的实现。
- 往返计算单元的数据流不受限制。ADSP-2199x 采用改进的哈佛结构与数据存储器文件相结合;在一个周期内 DSP 可以从存储器中读 2 个数据或向存储器写 1 个数据;完成 1 次计算;向寄存器列阵写 3 个值。
- 可扩展计算精度和动态范围。有 40 位可扩展精度,可控制 16 位整数和小数格式(二进制补码和无符号数),通过计算单元中的结果寄存器扩展精度,从而限制了中间计算的舍入误差。
- 支持循环缓存的 2 个数据地址发生器。2 个数据地址发生器可提供直接和间接(预修改和过修改)寻址。支持模数(Modulus)和位反转操作,存储器页只限制数据缓存的分布。
- 有效的程序顺序控制。DSP 支持快速建立和退出循环的方法。循环是可嵌套(硬件支持 8 级嵌套)和可中断的。处理器支持延迟和非延迟分支。

1.2 ADSP-2199x 的结构

DSP 提供一个完整的片上系统,它集成了足够容量的、快速的、支持 DMA 总线的 SRAM 和 I/O 外围。图 1-1 给出了 ADSP-2199x 各结构模块的功能。

ADSP-2199x 系列 DSP 的结构包括 ADSP-219x 内核(3 个计算单元、2 个数据地址发生器、1 个程序顺序控制器)、1 个 A/D 转换器、1 个编码器接口、1 个 PWM 模块、1 个 CAN 模块(只有 ADSP-21992 有)、1 个 SPORT 串行口、1 个 SPI 兼容同步串行口、1 个 DMA 控制器、3 个可编程定时器、通用可编程 I/O 引脚、众多的中断源、片内程序存储器和数据存储器模块。

间接寻址操作增强了寻址的灵活性,采用 8 位立即数、或补码、或基址寄存器的预修改和过修改,可很容易地实现循环缓冲寻址。

ADSP-2199x 系列 DSP 片内存储器的容量各不相同,参见第 4 章存储器的相关内容。

低功耗功能满足了便携式设备电池供电系统的低功耗需求。

ADSP-2199x 系列灵活的结构和指令集支持乘法的并行操作。例如,在一个指令周期里可以做如下事情:

- 为下一条指令取指提供地址;
- 取下一条指令;
- 完成一个或两个数据传送;
- 更新一个或两个数据地址指针;
- 完成一次运算。

这些操作也可以发生在如下情况下:

- 通过 SPORT 口连续接收和发送数据;
- 通过 SPI 口连续接收或发送数据;
- 通过外部数据存储器接口连续访问外部存储器;
- 连续递减定时器;
- 连续操作嵌入式外围(ADC、PWM、EIU 等)。

1.2.1 DSP 的内核结构

ADSP-219x 的指令集提供了灵活的数据传送和多功能(1 个计算执行 1 次或 2 次数据传送)指令。单字节指令可在 1 个周期内完成。ADSP-219x 汇编语言使用了类似高级语言的格式,易于编程和阅读。

在图 1-2 中,给出了 ADSP-219x 的内核结构。它包含 3 个计算单元:ALU、乘法器/累加器、移位器。计算单元处理寄存器列阵中的 16 位数据,支持多精度计算。ALU 可进行算术和逻辑运算,也支持简单除法。乘法器在单周期内可完成单乘法运算、乘加运算或乘减运算。乘法器有两个 40 位累加器,它可防止数据溢出。移位器可进行逻辑和算术移位、规格化和反规格化、二进制提取指数操作。

寄存器用于存放输入数据和结果数据。在大多数操作中,计算单元的数据寄存器作为一个称之为“数据寄存器列阵”来使用,它允许任意一个输入或结果寄存器作为任一次计算的数据输入。对于反馈操作,计算单元允许任何单元的输出(结果)作为下一个周期中任何单元的

输入;但对于条件指令或多功能指令则有一些限制。

程序顺序控制器控制程序的流程。控制器支持条件跳转、子程序调用和中断。由于使用了内部的循环计数器和循环堆栈,ADSP-2199x 可执行高效率的循环指令,无须用外部的 JUMP 指令来维持循环。

2 个数据地址发生器 DAG 可同时进行 2 个寻址操作(对数据存储器寻址和对程序存储器寻址)。每一个 DAG 可维护和更新 4 个 16 位地址指针。无论何时地址指针用于访问数据(间接寻址),都可通过对 4 个指针的任一个进行预修改和过修改寻址。一个长度数据和一个基地址与每个指针结合,都可实现循环缓冲寻址。但这些循环缓冲区不能跨越存储器的 64K 字页边界。二级缓冲寄存器复制了 DAG 主寄存器的全部内容,以利于实现数据的快速切换。

内核通过使用下列内部总线来获得高效的数据传递:

- 程序存储器地址总线 PMA;
- 程序存储器数据总线 PMD;
- 数据存储器地址总线 DMA;
- 数据存储器数据总线 DMD;
- DMA 地址总线;
- DMA 数据总线。

所有内部地址总线都与一条外部地址总线接口,允许扩展片外存储器。同样,所有内部数据总线也都与一条外部数据总线接口。引导存储器和外部 I/O 存储器也共用这条外部总线。

程序存储器可存储指令和数据,允许 DSP 内核在单个周期内同时取 2 个操作数:一个来自于内部程序存储器;一个来自于内部数据存储器。DSP 的双存储器总线也允许 DSP 内核在一个周期中从内部数据存储器中取 1 个操作数,同时从内部程序存储器中取下一条指令。

1.2.2 DSP 的外围结构

图 1-1 给出了 DSP 芯片的外围结构,它包括外部存储器接口、JTAG 检测和仿真接口、通信接口、混合信号外设、定时器、I/O 标志寄存器和中断控制器。

ADSP-2199x 有一个外部存储器接口,它连接着 DSP 内核、DMA 控制器和 DMA 可用外设(包括 SPORT 串行口、SPI 串行口和 A/D 转换器)。这个外部接口有 8 位或 16 位数据总线以及 22 位地址总线。数据总线可选择使用 8 位或 16 位。DSP 对数据打包的支持使 DSP 可直接使用来自于外部存储器的 16 位或 24 位数据,而不用考虑外部数据总线的宽度是多少。

存储器 DMA 控制器允许 ADSP-2199x 在外部和内部存储器之间传送数据。片内外设也可使用这个接口实现与存储器间的 DMA 数据传送。

ADSP-2199x 最多能响应 17 个中断:3 个内部中断(堆栈、仿真内核和掉电中断)、2 个外部中断(仿真和复位中断)和 12 个用户自定义(外设)中断。用户可指定任一个外设中断作为这 12 个自定义中断之一,同时也定义了这 12 个中断的优先级。几个外设中断也可组合在一起使用一个中断请求。

3 个可编程的定时器能产生周期性的中断信号。每一个定时器都可独立地操作在以下 3 种模式之一:

- PWM 模式;
- 脉宽计数和捕捉模式;

- 外部事件看门狗模式。

每个定时器都有 1 个双向引脚和 4 个用于控制它的操作模式的寄存器,即 1 个配置寄存器、1 个计数器、1 个周期寄存器和 1 个脉宽寄存器。定时器全局状态和控制寄存器包含了 3 个定时器的状态。在模式状态寄存器中的一个特定位可全局地控制所有 3 个定时器的使能或无效;在每个定时器配置寄存器中的一个特定位只能控制相应的定时器的使能或无效,而不影响其他定时器。

以下简单介绍各外围模块。

1. DSP 串行接口 SPORT

ADSP-2199x 为多处理器串行通信提供了一个同步的串口 SPORT。SPORT 支持以下形式:

- SPORT 有独立的发送和接收引脚,支持双向操作。
- SPORT 有 8 级深的发送和接收缓冲接口;一个数据寄存器用于和 DSP 其他部分之间传送数据字;还有一个移位寄存器,将数据移入或移出数据寄存器。
- 每一个发送和接收引脚或者使用一个外部串行时钟(≤ 80 MHz),或者使用内部自己的时钟,频率范围为 1 144 Hz~80 MHz。
- SPORT 支持多种串行数据长度,从 3 位~16 位,可选择以高位先行或低位先行方式传送。
- 每一个发送和接收接口都可选择使用或不使用帧同步信号。
- 按照 ITU 推荐的 G. 711 标准,SPORT 支持按 A 法则或 μ 法则实现硬件压扩格式。
- 在 DMA 的帮助下,SPORT 能自动地接收和发送整个数据块,实现每个 DSP 周期传送一个数据字。
- 在完成一个数据字传送或使用 DMA 完成整个数据块的传送后,每一个发送或接收接口都能产生一个中断。
- SPORT 支持 H. 100 多通道标准。

2. 串行外围接口 SPI

ADSP-2199x 有一个独立的串行外围接口 SPI。它为各种 SPI 兼容外围提供一个 I/O 接口。SPI 接口有其自身控制寄存器和数据缓冲设置。使用一系列的可配置选项,SPI 提供了一个非粘性的硬件接口,用于与其他各种 SPI 兼容外围接口。

SPI 有 4 个引脚,它们是 2 个数据引脚、1 个片选引脚和 1 个时钟引脚。SPI 是全双工同步串行接口,支持主模式、从模式和多主环境。对于多从环境,ADSP-2199x 能使用可编程标志引脚 PF1~PF7,作为给 SPI 从设备发出的从选信号。

SPI 引脚的波特率、时钟相位或极性是可编程的。发送和接收通道都可分别配置为 DMA 传送。但 SPI 的 DMA 控制器是单方向的,只能在任何给定时间里进行单方向(发送或接收)DMA 操作。

在数据传送过程中,SPI 接口可同时在它的 2 条串行数据线上串行移入和移出数据。串行时钟使 2 条串行数据线上的数据移位和采样同步。

3. CAN 模块

在 ADSP-21992 上有一个 CAN 模块,设计符合 CAN V2.0B 标准。这个 CAN 模块是低