

第1章 概 论

自 1982 年 TI 公司推出第一枚可编程 DSP 芯片以来, DSP 技术以其独特的优势取得了突飞猛进的发展, 该技术已在航空航天、遥测遥感、生物医学、自动控制、振动工程、通信雷达、水文科学等许多领域有着十分广泛的应用。随着微电子技术以及 DSP 算法的深入发展, 数字信号处理技术将深入到人们生活的各个方面。

1.1 数字信号处理器技术

信号是信息的载体, 信号处理本质上就是对信息的变换和提取, 将有用信息从各种噪声、干扰的环境中提取出来, 并变换为一种便于为人或机器所使用的形式。广义的数字信号处理技术是指理论上对数字信号处理的研究, 而数字信号处理器是指实现数字信号处理理论的一类专门芯片。DSP 已经逐渐成为数字信号处理器的代名词。

1.1.1 DSP 的发展历程

数字信号处理技术的发展大致经历了以下三个阶段:

- 信号解析与数字化研究阶段;
- 数字信号处理器产品普及阶段;
- 现代数字信号处理器快速发展阶段。

在数字信号处理器出现之前, 数字信号处理只能依靠微处理器单元 (MPU) 来完成, 但 MPU 较低的处理速度无法满足用户高速实时的要求。直到 1965 年 COLLEY-TUKEY 提出快速傅里叶变换算法, 才使开发人员进入到 DSP 的理论和算法基础的研究上。但此时的 DSP 仅仅被局限在教科书上, 研制出来的 DSP 系统都是由分立元件组成的, 其应用领域仅局限于军事、航空航天部门。

随着大规模集成电路技术的发展, 1982 年世界上诞生了首枚 DSP 芯片。这种 DSP 器件采用微米工艺 NMOS 技术制作, 虽功耗和尺寸稍大, 但运算速度却比 MPU 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。DSP 芯片的问世是个里程碑, 它标志着 DSP 应用系统由大型系统向小型化迈进了一大步。到了 20 世纪 80 年代中期, 随着 CMOS 技术的进步与发展, 第二代基于 CMOS 工艺的 DSP 芯片应运而生, 其存储容量和运算速度都得到了成倍的提高, 成为图像处理、语音处理、快速数据传输、生物医学系统等领域技术发展的基础。

到了 20 世纪 80 年代后期, 第三代 DSP 芯片问世以后, 它的运算速度进一步提高, 其应用范围逐步扩大到通信、计算机领域。进入 90 年代, DSP 发展最快, 相继出现了第四代和第五代 DSP 器件。现在的 DSP 属于第五代产品, 它与第四代相比, 系统集成度更高, 已将 DSP 芯核及外围器件综合集成在单一芯片上。这种集成度极高的 DSP 芯片不仅在通信、计算机领域大显身手, 而且逐渐渗透到人们的日常消费领域。

经过 20 多年的发展, DSP 产品的应用已深入到人们的学习、工作和生活的各个方面,

并逐渐成为电子产品更新换代的决定因素。目前，随着人们对 DSP 需求的不断增加，前景将十分乐观。

同时，数字信号处理技术在理论和算法上也取得了突破性进展，为其他新兴学科的发展奠定了理论基础。它本身也形成了比较完善的理论体系，包括信号采集，离散信号与离散系统分析，信号估计，信号建模，信号处理算法等内容。

1.1.2 DSP 的特点

DSP 之所以得到越来越广泛的应用，这与其自身的优势是分不开的。传统的微处理器不适用于进行数字信号处理所需的高等数学运算，其运算需要较长的计算时间，不能满足现代信号实时处理的要求。而数字信号处理器有以下明显的优势：

- 数字信号处理器具有较大的动态范围，易于实现机器和设备之间的匹配；
- 用于数据处理时具有较高的信噪比 (S/N)，在处理过程中不会产生噪声；
- 数字数据具有高度的灵活性，易于处理、缓存、组合，可以时分多用、并行处理，具有极好的重现性、可靠性和预见性；
- 执行速度较快，整体性能高，可实现真正的实时处理；
- 在现代数字信号处理器的硬件结构上，对噪声、非平衡干扰和多径干扰，都有相应的方法进行信号处理；
- 内部都采用了哈佛总线结构，程序和数据具有独立的存储空间，而且有各自独立的程序总线 and 数据总线，与程序和数据共用存储空间的冯·诺依曼结构相比，大大提高了数据处理速度，提高了数据的吞吐率，非常适合于实时信号处理；
- 采用专门的硬件乘法器，打破了传统乘法运算由于速度不高而产生的对数字信号处理的瓶颈限制，能在同一时钟周期内进行相乘和累加运算，大大加快了数字信号处理算法的运算速度；
- 采用流水操作结构，能减少指令执行时间，而且程序执行时间可预测；
- 有双地址单元，支持多种寻址方式；
- 具有有效的程序控制机制并有硬件逻辑支持零开销循环；
- 对输入、输出数据流没有限制；
- 具有特殊的 DSP 指令系统，支持汇编和高级语言编程。

1.1.3 DSP 结构

在 DSP 应用设计中，设计者必须对 DSP 的结构有深刻的了解，只有这样，设计的系统才能可靠地运行，程序代码的效率才可能大大提高。图 1.1 所示是 ADI 系列 DSP 的最基本的结构框图。

由图 1.1 可以看出，DSP 的基本结构主要包括以下四个部分。

1. 计算单元

由于 DSP 在 1 个时钟周期里必须完成乘/累加、加、减、与/或、移位等操作，所以对计算单元硬件结构的优化是 DSP 处理器的核心。也正是因为这一点，才将 DSP 和通用微处理器区分开来，通用微处理器需要多个周期才能完成这些操作。

通常 DSP 中的乘/累加器(MAC)、算术逻辑单元 (ALU) 和桶形移位器(移位器)，主

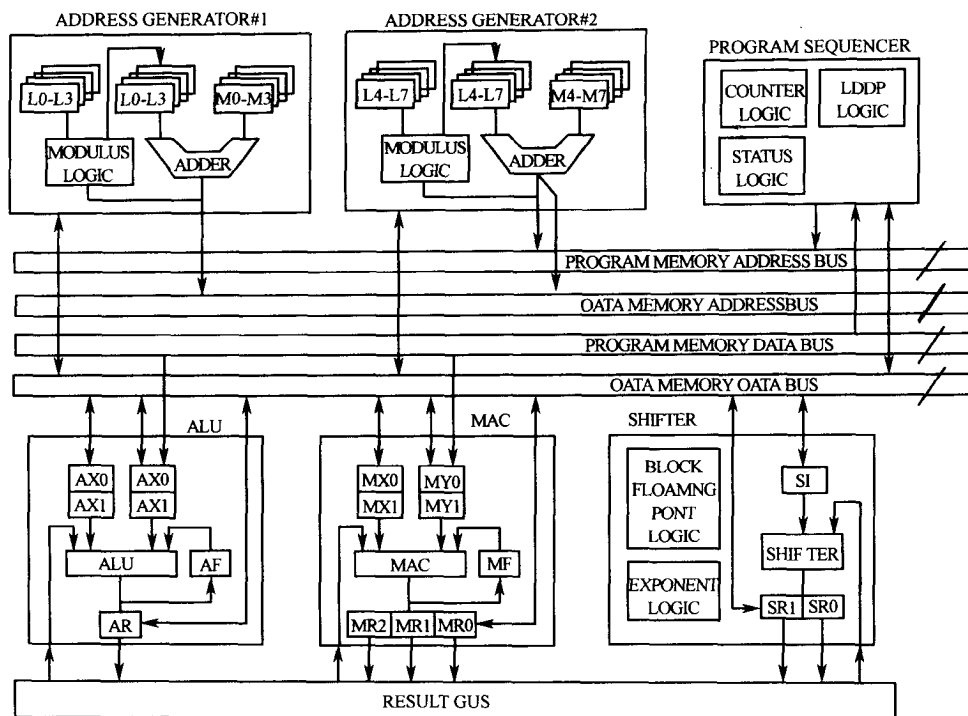


图 1.1 典型 ADSP 结构框图

要实现数值运算。MAC 实现乘/累加操作，多数 DSP 算法（例如 FIR、IIR 滤波和快速傅里叶变换）中都要用到该操作。ALU 实现加、减和逻辑操作。位操作和字操作主要在移位器中进行。图 1.1 也说明了 MAC、ALU 和移位器的并行结构，以及它们的数据输入和输出通道。

从编程的角度来看，使用计算单元相互分离的 DSP 结构使得程序开发具有很大的灵活性和较高的效率。数据流不会冲突，能够实现单周期的计算操作。DSP 的结构也为 MAC 操作提供了较高的动态范围，使 MAC 能处理 2 倍输入数据宽度的乘法结果，而累加器输出时也不会溢出（对一个 16 位的 DSP，这一特性等效于 16 位的数据输入和 40 位的 MAC 累加输出）。

此外，计算单元的其他特性也大大方便了实时系统的编程。计算操作会引起状态位的变化，对这些状态进行测试，可以确定计算操作是否产生进位、溢出、饱和等，从而使 DSP 可以快速地处理基于计算操作的程序流，实现不同条件下的连续操作。当然，由于计算单元需要不断地将数据反馈给输入，这将对 DSP 存储器和内部总线结构的设计产生很大影响。

2. 存储器部分

DSP 存储器和总线结构是根据处理速度的要求进行设计的。数据和指令必须在每一个指令周期里流入计算和程序控制器单元，而没有延迟和瓶颈的限制。因此，设计的关键就是如何提高数据的吞吐率。

要提高数据吞吐率，我们先来看看 DSP 存储器和通用微处理器存储器的设计之间的区别。以前大多数的微处理器都采用冯·诺依曼结构，都使用单一的存储器空间，既存放数据又存放指令，并用 1 条地址总线和 1 条数据总线。当在每个周期需要总线访问时，必须判断是访问指令还是数据，这会大大限制数据的吞吐率。而在 DSP 中采用哈佛结构，存储器通

常分为程序存储器和数据存储器，两个存储器都有独立的总线结构，这样就将指令和数据访问分离开来，DSP 能够在每个周期同时取指令和数据，从而使数据吞吐率加倍。此外，还可以使用其他优化措施（如指令缓存、结果反馈和上下文切换）来提高 DSP 的数据吞吐率。

另外，DSP 存储器结构的优化还包括重复的存储器访问。在大多数 DSP 算法中，例如在数字滤波器的使用中，需要从存储器中反复读取数据。DSP 通过两个地址产生器支持不同类型的缓冲寻址方式，并通过硬件逻辑以最高的效率执行地址修改和比较操作。然而，多数通用数字信号处理器都是通过软件方式实现这种操作，这就限制了处理器对实时信号处理的能力。

3. 程序控制器

多数 DSP 通常都是采用递归的算法，在递归软件设计中，DSP 程序控制器可以实现零开销的程序循环，这也是 DSP 区别通用微处理器的一个主要特性。通常，微处理器要靠软件维持程序循环，在程序的末尾处放置条件指令，决定地址指针是返回循环的起始地址还是跳转到其他地址。然而 DSP 通过硬件方式执行条件测试和指令跳转，而且通过指令流水结构实现指令预测，大大提高了程序的执行效率。

4. 输入输出（I/O）单元

DSP 需要很高的数据吞吐率，因此数据的输入和输出也是很关键的。现代 DSP 系统中都设计有专用的 I/O 处理器，主要可以完成自举载入、串行口通信、DMA 操作、存储器映射 I/O 等操作任务。

1.1.4 高速、高性能信号处理

早期的信号处理主要是采用模拟的处理方法，包括运算放大电路、声表面波器件（SAW），以及电荷耦合器件（CCD）等。模拟处理最大的弊端就是不灵活、不稳定，参数修改困难，而且对周围环境变化敏感。数字信号处理可以通过软件修改处理参数，因此具有很大的灵活性。而且数字电路采用了数值逻辑，只要环境温度、电路噪声的变化不造成电路逻辑的翻转，数字电路的工作都不会受到影响，因此具有很好的稳定性。正因为如此，数字信号处理已经成为信号处理技术的主流，但是处理量随处理精度、信息量的增加而成倍增长，因此研究高速的数字信号处理系统就显得异常重要。

高速数字信号处理器的特点首先是高速度，其处理速度可以达到每秒数百兆次的数量级。其次是大电流，高速信号处理芯片的电流经常在 1 A 以上。第三是低电压，这是为了在大电流下减小系统的功耗，系统的工作电压从标准的 5 V 降至 3.3 V、3 V、2.5 V、1.8 V，甚至 0.9 V。第四是高度集成，芯片的集成度在数十到数百万门的量级。第五就是为了提高运行速度而采用了多种并行的体系结构。

鉴于以上特点，在实现高速实时信号处理系统时，首先要采用先进的设计软件来保证系统设计的正确性，比如用电子设计自动化（EDA）软件进行优化设计；其次，采用专用集成电路（ASIC）技术减小体积，提高集成度，而在样机阶段，则通常采用可擦除可编程逻辑器件（EPLD）或现场可编程门阵列（FPGA）来减小风险；第三，要研究高速度、低电平器件的特点和使用方式；第四，要研究并行体系结构的设计和选择问题。

通常，一个高速实时数字信号处理系统的研制可能包括以下问题：

- 高速实时数据采集（ADC）；
- 高速实时数据存储（MEM）；

- 高速实时周边器件（中小规模器件）；
- 高速实时电路集成（EPLD/FPGA/ASIC）；
- 高速实时信号生成（DAC/DDS）；
- 高速实时 DSP 与并行体系结构；
- 高速实时总线技术（VME/VXI/PCI）；
- 高速实时系统设计（EDA）等。

目前，高速实时数字信号处理技术已经取得了飞速的发展，单片 DSP 芯片的速度已经达到每秒 16 亿条指令（1600 MIPS）。高速实时 DSP 芯片采用了各种并行处理技术，包括片内并行和片间并行等方式来提高处理速度。其中，主要的并行 DSP 芯片包括美国 TI 公司的 TMS320C8x 和 TMS320C6x，以及美国 ADI 公司的 ADSP-2106x 及 ADSPTS-101S 等。

TMS320C8x 采用片内并行的多指令多数据 MIMD 体系结构，在每一片 TMS320C80 内部，都有 1 个浮点 RISC 类型主处理器 DSP 和 4 个定点并行处理器 DSP。每个 DSP 都配有 10 KB 片内存储器（RAM）。TMS320C8x 是一个紧耦合的多指令多数据流（MIMD）的单片多处理器系统。这一系统的运行速度等效于每秒 20 亿次 RISC 类型的操作。在这个系统中，一个显著的特点是采用交叉开关（Crossbar）代替了传统的总线互连，避免了因为总线仲裁对总线传输数据速度的限制。

TMS320C6x 采用片内并行的超长指令字（VLIW）体系结构，单指令字长为 32 位，8 个指令组成一个指令包，总字长为 $8 \times 32 \text{ b} = 256 \text{ b}$ 。芯片内部设置了专门的指令分配模块，可以将每个 256 位的指令包同时分配到 8 个处理单元，并由 8 个处理单元同时运行。当芯片内部 8 个处理单元同时运行时，其最大处理能力可以达到 1600 MIPS。

在 VLIW 的体系结构中，多个功能单元并行工作，所有的功能单元共享公用大型寄存器堆。VLIW 处理机的另一个特点是在指令获取、指令分配、指令执行、数据存储等阶段需要进行多级流水，而且不同指令执行的流水延迟时间也不相同，因此各种指令的安排要尽量不破坏指令流水的执行，否则处理机运行的效率也会大大降低。

ADSP-2106x 是美国 ADI 公司推出的可并行扩展的超级哈佛指令计算机（SHARC）。芯片内部有 4 套独立的总线，可完成双向数据存取、指令存取、非指令性 I/O。这种芯片的主要型号包括 ADSP-21060, ADSP-21061, ADSP-21062 等，其主要特点是在一个 ADSP-21020 的浮点 DSP 核心基础上集成了片内大容量双口 RAM 和并行处理接口，因此是一个可并行扩展的 SHARC 结构。ADSP-2106 支持多 DSP 系统，可以方便地构成各种体系结构的多 DSP 系统。ADSP-2106x 对多 DSP 处理的支持包括 6 个链路（Link Port），片内大容量共享存储器和 DMA 传输，统一分配的片内存储器地址，与宿主机的接口，分布式总线仲裁逻辑等。因此，采用多片 ADSP-2106x 可方便地构成无缝连接的多片并行处理系统，包括共享存储器并行处理系统、数据流并行处理系统、集束并行处理系统、单指令多数据流（SIMD）并行处理系统等。

ADSP-2106x 并行处理的典型应用是 AD14060，它是把 4 片 ADSP-2106x 以集束多处理模式，采用 MCM 工艺集成于一个芯片模块之中，因此这种芯片模块的处理能力是单片 ADSP-2106x 的 4 倍，即峰值处理能力为 480 MFLOPS，持续处理能力为 320 MFLOPS。

1.1.5 DSP 发展趋势

从市场趋势看，根据美国 Forward Concepts 调查公司针对 1999 年全球 DSP 市场所作的

调查显示,美国 TI (Texas Instruments) 公司和美国 ADI (Analog Devices Inc.) 公司在市场占有率方面,以双强局面赢得全球排名之冠。成长最快的是 ADI 公司,1999 年市场占有率较前一年增加 43%,主要是由于亚洲市场中 GSM Chip-Set 的带动。TI 公司则年增长 28%,尤其在移动电话芯片市场约占全球份额的 60%。Lucent 和 Motorola 则在移动电话基站方面加快发展。其他厂商方面,如日本 NEC 与日立、美国 Zilog 等都在成长中。今后美国的 LSI-Logic、德国的 Infineon Technologies 数家厂商也将参与到 DSP 市场的竞争行列。

DSP 的发展日新月异,主要有以下几个发展方向。

1. 系统级集成 DSP

缩小 DSP 芯片尺寸始终是 DSP 的技术发展方向。各 DSP 厂商纷纷采用新工艺,改进 DSP 芯核,并将几个 DSP 芯核、MPU 芯核、专用处理单元、外围电路单元、存储单元系统集成在一个芯片上,成为 DSP 系统级集成电路。

2. 可编程 DSP

采用可编程 DSP,设计简单,易于修改,性价比高。

3. 追求更高的运算速度

目前一般的 DSP 运算速度为 100 MIPS,即每秒钟可运算 1 亿条指令。TI 的 TM320C6X 芯片由于采用 VLIW 结构设计,其指令速度已高达 2000 MIPS,即每秒可运算 20 亿条指令。按照 CMOS 的发展趋势,DSP 的指令速度再提高 100 倍(达到 1 600 GIPS)是完全有可能的。

4. 定点 DSP 是主流

从理论上讲,虽然浮点 DSP 的动态范围比定点 DSP 大,且更适合于各种类型的 DSP 应用场合,但定点运算的 DSP 器件的成本较低,对存储器的要求也较低,而且耗电较省。因此,定点运算的可编程 DSP 器件仍是市场上的主流产品。据统计,目前销售的 DSP 器件中的 80% 以上属于 16 位定点可编程 DSP 器件,预计今后的比重将逐渐增大。

5. 与可编程器件结合

Forward Concepts 公司分析师 Will Strauss 表示,许多新应用需要比传统 DSP 处理器更加强化的数字信号处理能力。而据《可编程逻辑新闻和观点》的主编 Murray Dismar 介绍,大约 10% 的 PLD/FPGA 业务与 DSP 业务相关。对 PLD 和 FPGA 供应商来说,DSP 还是一项刚刚起步的新应用。在过去几年中,主要供应商在蜂窝基站的应用领域实现了长足的发展,因为设计者往往会借助 PLD 和 FPGA 来满足他们日益提高的信号处理需求。与常规 DSP 器件相比,FPGA 器件配合传统的 DSP 器件可以处理更多信道,可在基站中用来实现高速处理功能。

DSP 在无线电通信方面的应用也极其广泛,无论是基站还是手机都离不开 DSP 芯片,新型无线通信系统、设备的升级换代都基于 DSP 芯片性能的提高和发展,无线 Modem、多媒体高级终端等都依赖于 DSP 芯片,基于 DSP 的软件无线电成为当前无线电通信中的研究热点。

不论 DSP 市场如何风云变幻,有一点是毫无疑问的,DSP 技术将会突飞猛进地发展,DSP 产品将会在很多方面改变我们的生活。

1.2 数字信号处理器应用设计

随着 DSP 芯片性能的不断提高和芯片价格的不断降低, DSP 的应用范围将更为广泛。

1.2.1 DSP 应用领域

DSP 的应用越来越广泛,几乎遍及整个电子领域,目前主要应用市场为 3C (Communication, Computer, Consumer) 领域,几乎占整个市场需求的 90%。常见的典型应用有以下几种。

1. 通用数字信号处理

通用数字信号处理包括数字滤波、自适应滤波、卷积与相关、FFT 变换等。

2. 通信领域的应用

在通信领域的应用主要包括音频邮件、视频会议、ISDN、GSM、高速调制/解调器和传真等。数字蜂窝电话是 DSP 最为重要的应用领域。由于采用 DSP 技术,蜂窝电话的更新换代变得更为容易,只须在统一的硬件平台基础上通过软件的不不断升级就可生产出各式各样的新产品,实现真正意义上的软件无线电。在高速调制解调器领域, DSP 的应用成效显著。在 DSP 技术的推动下,56 kb/s 调制解调器才刚刚投入市场,高速率的 ADSL 调制解调器又问世了,其传输速率已高达 6.3 Mb/s,且具有接收动态图像的能力,这使得更多用户可以快速登录到因特网。

3. 图像处理

图像处理主要包括模式识别、实时图像压缩等。在图像处理领域,已经开发出专用的用于 JPEG 标准静态图像数据处理的 DSP 和用于动态图像数据处理的 DSP,广泛应用于三维图像变换、图像增强等方面。

4. 仪器仪表

主要应用于数据采集、暂态分析、频谱分析,以及波形产生等。

5. 医学电子

主要包括计算机 X 射线扫描设备、超声波检测仪、心电图仪等。

6. 雷达和声呐

主要应用于雷达实时信号处理和声呐实时信号处理。

7. 计算机与工作站

可编程多媒体 DSP 是 PC 领域的主流产品。以 XDSL Modem 为代表的高速通信技术与 MPEG 图像技术相结合,使得高品质的音频和视频形式的计算机数据进行实时交换成为可能。

8. 音频与视频处理

由于 DSP 的广泛应用,促进了数字电视、HDTV、DVD/CD、媒体播放器、数字收音机 (DAB)、声音与音乐处理等产品的开发与普及。

9. 高速控制

高速控制主要包括激光打印控制、硬盘驱动控制、模拟机器人、电机控制、伺服控制等方面。

对于一个完整的 DSP 系统，也主要有两种应用模式：一种是用同步控制单元实现多路 DSP 处理，例如美国 ORBAN 公司生产的 FM8200 数字音频处理器，其控制单元采用多片 DSP56001F27；另一种工作模式就是利用 DSP 高速的信号处理芯片进行各种实时处理，例如立体声信源编码、解码，视频图像压缩编码、解码，包括运动预测和 DCT 变换算法。在 Ureka-147 DAB 广播、DVB 系统中，DSP 芯片可实现信源编解码，如数字频带的分析、心理模型和量化编码；实现信道编解码（COFDM），如卷积编码、帧交织、维特比译码、校正、校验等。

1.2.2 典型 DSP 系统的构成

前面介绍了处理器的结构，那么如何利用处理器构成应用系统呢？图 1.2 给出了典型的 DSP 数据采集和控制系统的组成框图。

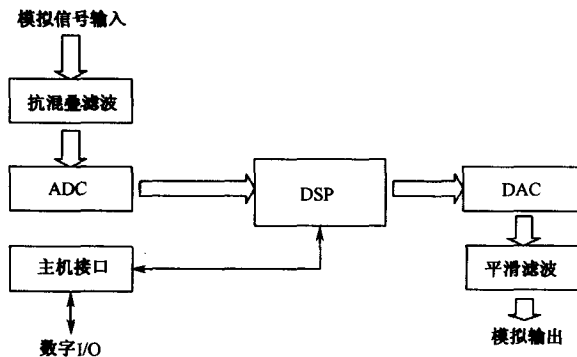


图 1.2 DSP 典型系统组成框图

下面简要说明图 1.2 中各组成部分的功能。

(1) 模拟输入：模拟输入信号可以是电压信号、电流信号，形式可以是连续信号或脉冲信号。模拟输入信号经过抗混叠滤波器后，得到近似的带限信号，然后输入到 ADC，转换成数字比特流。为了保证信息不丢失，根据抽样定理，抽样频率必须是输入带限信号最小频率的 2 倍。而且，为了提高信噪比，A/D 转换器的采样精度要尽可能高。数据采样后，转换器中断 DSP 处理器，让 DSP 接收采样数据。ADC 与 DSP 接口究竟采样并行接口还是串行接口的数据，则由数据量、处理速度、要求设计复杂度、开销、布线空间、功耗以及成本等因素决定。

(2) 数字信号处理：DSP 主要实现对输入采样数据的处理。DSP 系统也可以由多个 DSP 构成，这取决于对信号处理的要求。处理算法也是多种多样，主要根据应用要求而定。输入数据经过算法软件处理后，就可以将处理结果送到 DAC。由于信号处理是可编程的，程序可以不断调整、修正，这样，数字信号处理器在处理数据和提高系统性能方面就具有相当大的灵活性。

(3) 模拟输出：DAC 转换器在下一个采样时钟周期内把 DSP 输出数据转换成需要的模拟量输出。转换器的输出经过低通和平滑滤波后，就可以重建模拟输入信号。

(4) 主机接口：主机接口主要实现对 DSP 与外界通信、发送和接收数据的控制，使得 DSP 系统具有良好的人机交互性。

以上给出的是相对比较完备的 DSP 系统，但并不是所有的 DSP 应用系统都要求有以上功能模块，设计者应根据需要选择 DSP 的外围组件。

1.2.3 DSP 系统设计与应用设计问题

在利用 DSP 进行系统设计时，首先要明确设计任务，编写设计任务说明书，将系统要达到的功能描述准确，然后转化为可量化的技术指标，结合所采用的 DSP 的性能进行设计。图 1.3 说明了 DSP 系统设计的大致步骤。

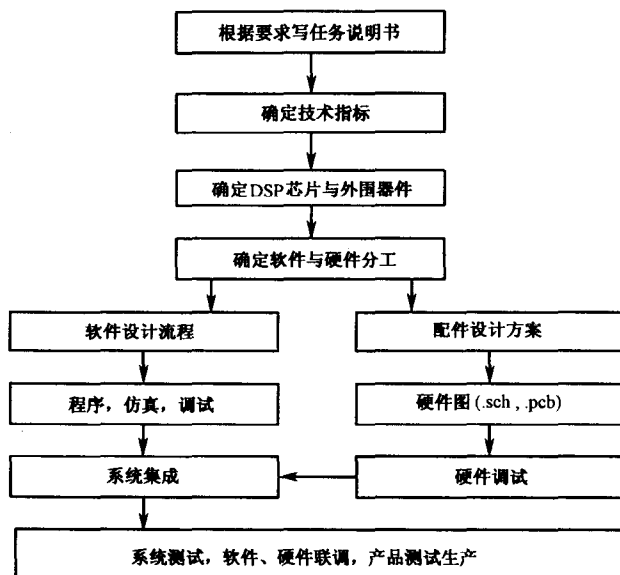


图 1.3 DSP 系统设计步骤

设计人员在确定技术指标后，根据要求选择 DSP 芯片的类型和外围器件是整个系统设计的关键，主要应考虑以下因素。

1. DSP 的速度

DSP 速度指标的选择通常是在 DSP 实时处理系统设计时需要考虑的因素。实际上，许多数字信号处理算法早在几十年前就已经研究出来，而得不到实际应用就是因为受到 CPU 处理速度的限制。即使在目前，DSP 的处理能力高达每秒 1G 字节左右的浮点处理能力，面对一些复杂的问题也仍显不足，若不采用多 DSP 并行处理的方式，还是会受到 DSP 处理速度的制约。处理器的处理速度通常用处理器的指令周期来衡量，即处理器每秒执行多少兆条指令（MIPS）。但是该参数也具有一定的片面性，有的 DSP 采用超长指令字（VLIW）结构，单个周期时间内可以实现多条指令。因此，也可以采用完成一个基本操作（而不是指令）所需的时间作为标准来比较处理器的性能。

此外，利用标准程序的执行速度来衡量 DSP 的性能也是一个好的指标，如完成标准 1024 点 FFT 所需的时间。在比较处理器时钟速率时，还应该考虑 DSP 内部是否具有时钟倍频器或锁相环，如果有，则 DSP 可以使用外部低频时钟产生片上所需的高频时钟信号，具有更

高的执行效率。

2. 数据格式

处理器的数据格式决定了处理器具有处理不同精度、动态范围和信噪比的信号的能力。

数据格式一般是指 DSP 处理器采用定点算法还是浮点算法。绝大多数的 DSP 处理器使用定点算法，数值为整数或 $-1.0 \sim 1.0$ 之间的小数；而采用浮点算法的处理器，数据则表示成尾数乘以指数的形式：尾数 $\times 2^{\text{指数}}$ 。

浮点算法是一种较复杂的常规算法，但利用浮点数据可以实现大的数据动态范围。浮点 DSP 在设计时可不用关心动态范围和精度一类的问题。浮点 DSP 比定点 DSP 更容易编程，但是成本和功耗高。编程和算法设计人员通过分析或仿真来确定所需要的动态范围和精度。如果要求易于开发，而且动态范围很宽、精度很高，可以考虑采用浮点 DSP。

也可以在采用定点 DSP 的条件下采用“块浮点”的办法，将具有相同指数，而尾数不同的一组数据作为数据块进行处理。“块浮点”处理通常用软件来实现。

3. 数据宽度

浮点 DSP 的字宽通常为 32 位，而定点 DSP 的字宽一般为 16 位，也有 24 位和 20 位的 DSP。由于字宽与 DSP 的外部尺寸、管脚数量以及需要的存储器的大小等有很大的关系，所以字宽的长短直接影响到器件的成本。在满足设计要求的条件下，要尽量选用小字宽的 DSP，以减少成本。同时还应该权衡字宽和开发复杂度之间的关系。

4. 动态范围

DSP 在雷达、声呐、自适应滤波、图像处理以及音频识别等商业应用中都要求有较宽的动态范围，能够将有用的信号从噪声中提取出来。动态范围用最大数和最小数的比值来表示。传统上，数据压缩和解压算法都是对已知带宽的信号进行操作。目前的算法已逐渐发展到去掉对算法调整和中间结果动态范围的限制。图 1.4 所示说明了定点 DSP 不同数据位数与数据动态范围的关系，图 1.5 所示是浮点 DSP 与定点 DSP 动态范围的比较。由此可以看出，浮点 DSP 比定点 DSP 有更大的动态范围。

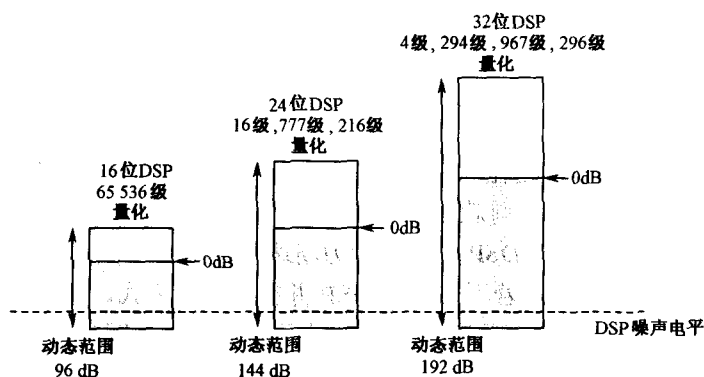


图 1.4 定点 DSP 数据位数与动态范围关系

5. 是否支持多处理器连接

在某些数据计算量很大的应用中，经常要求使用多个 DSP 处理器。在这种情况下，多处理器间的互连能力和性能（关于相互间通信流量、开销和时间延迟）成为重要的考虑因素。

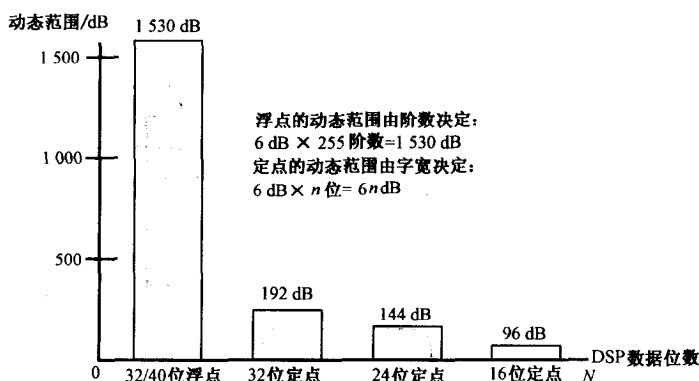


图 1.5 浮点 DSP 动态范围与定点动态范围比较

如 ADSP-2106X 系列提供了简化多处理器系统设计的 6 个链路口连接和总线连接，使得构成 ADSP-2106X 多处理器系统变得比较容易，可很方便地实现无缝连接。

6. 软件开发是否简便

在不同的应用中，对开发简便性的要求是不一样的。因此，选择 DSP 时还需要考虑软件开发工具（包括汇编、链接、仿真、调试、编译、代码库以及实时操作系统等部分）、硬件工具（开发板和仿真机）和高级工具（如基于框图的代码生成环境）等因素。

在选择好 DSP 器件后，就可以根据选用 DSP 芯片的性能和设计指标初步确定 A/D、D/A、存储器、电源、控制逻辑、总线接口等基本外围器件。

- 根据采样率以及数据精度确定 A/D 或 D/A 的型号，以及是否需要片上自带采样保持电路、多路器、基准电源、输入输出运放等，必须确保转换精度比滤波器的噪声电平要高。

- 选定 DSP 后，其对外部存储器系统的管理能力也就确定，片内存储器大小也就确定，此时需要根据性能指标确定存储器的工作频率、内存容量以及字长、接口方式、工作电压，确定是采用 SRAM、EPROM 还是 FLASH 等存储器件。

- 对于外围控制逻辑，首先要确定是用 PLD, CPLD，还是 FPGA。然后根据 DSP 的工作频率和对公司芯片的了解情况选择具体的型号。

- 电源管理和功耗也是整个系统设计中很重要的一部分，首先要根据各个选定器件的工作电压，确定电源应用模块。通常，对于便携式应用产品，还应考虑是否需要采用“休眠”或“空闲”模式，以降低整个系统的功耗。

以上这些在器件选择时需要考虑的因素有时也是相互影响的。此外，供货能力、性能价格比、技术支持、使用经验等都是需要充分考虑的因素。

完成应用系统的总体设计和芯片选择之后，就进入系统的软硬件设计阶段。图 1.6 和图 1.7 所示分别说明了系统软件和硬件的设计过程。

对于软件设计，ADI 公司为 ADSP SHARC 系列处理器提供了面向对象的软件开发平台——Visual DSP++2.0，该软件操作平台支持多处理器的汇编语言、C 语言和 C++ 高级程序设计语言，而且完全可视化，高度集成。ADSP SHARC 系列处理器指令系统比其他计算机指令系统简单，易于理解，而且助记符简单易记。同时，ADI 公司还针对各个不同类型的 ADSP 芯片提供了一些优化的常用数字信号处理算法、标准程序库等，这大大降低了软件设计的难度，使用户很容易编写出紧凑、高效的应用软件。

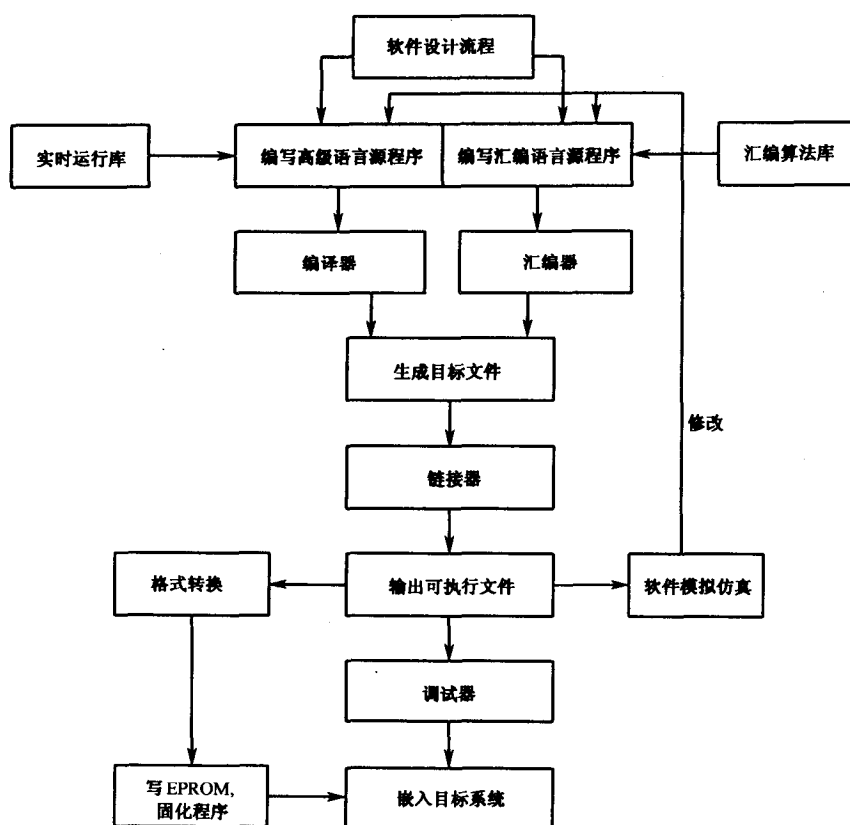


图 1.6 系统的软件设计过程

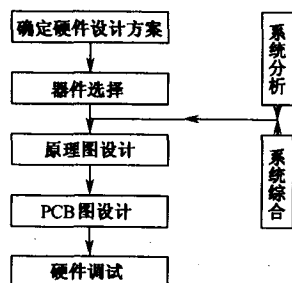


图 1.7 系统的硬件设计过程

如图 1.6 所示,在完成对源文件的编写后,送到汇编器或编译器进行编译,生成目标文件,然后将目标文件送入链接器进行链接,生成可执行文件。在生成可执行文件之后,就可以将可执行文件调入到调试器(包括软件仿真、系统仿真、评测模块等)进行调试,检查程序运行结果是否正确,否则要对程序不断进行修改、调试。当调试通过以后,就可以烧写 EPROM,固化程序,脱机运行程序,并检测运行结果是否正确,最后完成一系列软件测试。

对于系统硬件设计，也是整个应用系统很关键的环节，一旦硬件系统出现错误，修改就比较困难，这将给整个系统带来严重的影响。因此，在制作 PCB 电路板之前，一定要认真分析硬件系统设计方案。

在选定好硬件设计方案和器件后，就可以进行原理图设计，原理图设计的成功与否是 DSP 系统能否正常工作的最重要的因素之一，如果原理图出现错误，会导致 PCB 以及整个应用系统的错误。因此，在原理图设计时必须清楚了解器件的特性和使用方法，并在必要时进行一定的仿真。

完成原理图设计后，就可以制作 PCB 印制板图，布线时要清楚系统工作原理、布线工艺和系统结构设计方法，要注意一些常见的印制板布线对系统性能的影响因素，如布线干扰等。在完成系统的软硬件设计后就是系统集成。集成后的系统还应经过一系列的软、硬件联合调试，系统指标测试等过程。如果发现问题，可能还要对系统的软件、硬件进行必要的修改。修改的过程甚至要回到对硬件的原理图进行修改，对 PCB 图修改，重新进行硬件调试等过程。DSP 处理系统经过一定的反复后，会更加成熟，各项指标、性能才会逐渐达到预定的目标。实际上，其他的系统设计差不多都要经过同样的过程，这就是科学研究的规律。

第2章 ADSP SHARC 系列处理器原理

2.1 ADSP SHARC 系列处理器概述

ADSP SHARC 系列处理器是一种高性能的数字信号处理器, 主要型号包括 ADSP-21065L, ADSP-21061, ADSP-21062, ADSP-21060, ADSP-21160, 以及 2002 年推出的 Tiger SHARC TS-101 等, 这些处理器都以性能优良的浮点 DSP 核 ADSP-21000 为基础发展起来。芯片内部采用超级哈佛结构, 同时集成了大容量的双端口静态存储器 SRAM、程序/数据外部总线以及 I/O 外设, 从而形成了一个完整的数字信号处理系统。ADSP SHARC 系列处理器提供了多种外部控制信号, 只需很少的硬件逻辑, 就能构成一个功能强大、硬件结构简化的应用系统。多片处理器无需外部控制逻辑, 就能无缝连接成一个高效的紧耦合(通过总线结构)或者松耦合(通过链路结构)方式的并行多处理器系统。

2.1.1 ADSP SHARC 系列处理器发展历程

ADI 公司进入数字信号处理领域以来, 首先推出了 ADSP-21xx 系列定点 DSP, 主要产品有 ADSP-210x, ADSP-216x, ADSP-2111, ADSP-217x, ADSP-218x 和 ADSP-219x。随后又推出了高性能 BLACKfin 系列定点 DSP ADSP-21532 和 ADSP-21535 等。

实际应用中, 对处理速度和数据动态范围的要求越来越高。随着半导体技术的发展, 芯片制造工艺不断提高, DSP 芯片的性能也有极大的提高, 达到 100 MIPS。ADI 公司推出的 ADSP-210xx 系列每浮点 DSP 可以直接构成分布式并行系统和共享存储式并行系统, 数据动态范围宽, 满足实时信号处理的可重构性要求, 从最早的 ADSP-21000, 发展到今天高性能的 Tiger SHARC。随着 ADSP 芯片的不断推广, 其价格也不断下降, 受到越来越多的设计人员的认可。ADSP SHARC 系列处理器的发展历程如图 2.1 所示。

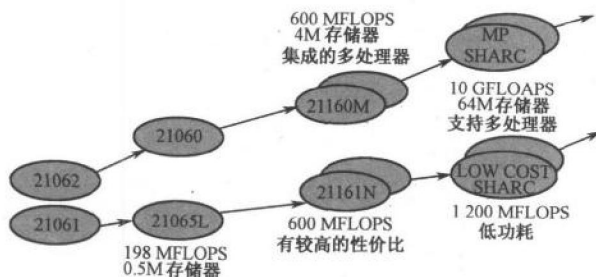


图 2.1 ADSP SHARC 系列处理器 SHARC 发展历程

2.1.2 ADSP SHARC 系列处理器与 TMS320 处理器性能比较

TI 公司和 ADI 公司是 DSP 芯片的两大主要供货商, 其芯片也代表了 DSP 领域的最高技术。TMS320 处理器主要基于 VLIW (超长指令字) 结构, 而 SHARC 系列 DSP 在结构设

计时更强调系统的平衡性。表 2.1、表 2.2 和表 2.3 给出了 ADI 与 TI 公司性能基本相当的部分 DSP 的性能比较，以给读者在选用器件时有一个明确的性能分析参考，以便在应用设计中选择性能更适合要求的处理器。

表 2.1 ADSP-21160 与 TMS320C6701 的比较

特 性	TMS320C6701(167M)	ADSP-21160 SHARC(100M)
是否支持 IEEE32 位浮点格式	支持	支持
是否支持 32 位定点	支持	支持
内部 SRAM 容量	1Mb	4Mb
是否有内部双端口存储器	无	有
串行口数量	2	2
是否支持多处理器连接	否	支持并行总线和链路连接
DMA 通道数量	2	14
是否支持零开销的 DMA	否	是
寄存器数量	32	128
累加器宽度	40 位	80 位
是否支持 64 位乘法	否	是
存储器带宽	128 位/周期	128 位/周期
是否支持软件循环	在紧凑循环中不允许中断	在紧凑循环中支持中断
汇编程序复杂度	高度复杂	代数汇编语言
支持环形缓冲数量	8	32
是否支持条件执行	需要额外的寄存器	支持简捷的条件逻辑
FIR 滤波指令大小	要 100 条指令	只需 25 条指令
32 位 1024 点 FFT 速度	120 μ s	90 μ s
功耗	不可测	2.5V, 2W

表 2.2 ADSP-21065L 与 TMS320C32 的比较

特 性	ADSP-21065L	TMS320C32
指令执行时间	15 ns	33 ns
峰值 MFLOPS	198	60
1024 点复数 FFT	0.27 ms	1.67 ms
除法 (Y/X)	6 周期, 90 ns	36 周期, 478 ns
数据寄存器数量	32	12
支持环形缓冲数量	32	1 (固定长度)
是否有备用寄存器	有	无
零开销的循环嵌套	6 级	1 级
中断相应时间	3 个周期, 45 ns	7 个周期, 231 ns
是否支持 32 位浮点	支持	支持
是否支持 IEEE 数据格式	支持	不支持
是否支持扩展经断 (40 位) 浮点	支持	不支持
定点乘法	32 $\times$ 32>=>64 位	24 $\times$ 24=>32 位

续表

特 性	ADSP-21065L	TMS320C32
定点累加器宽度	80 位	32 位
DMA 通道数量	6	1
是否支持无干预的 DMA	支持	不支持
片外到片内传输速度	264 MB/s	120 MB/s
片内到片外传输速度	264 MB/s	60 MB/s
串行口数量	2 个, 支持 TDM 模式	1 个, 不支持 TDM 模式
串行口传输速度	40 MB/s	15 MB/s
片内 SRAM 大小	16K×32 位字	0.5K×32 位字
片内 I/O 是否存在开销	没有, 双端口 RAM 消除了	总线冲突, 存在开销
存储区是否可配置 (程序与数据)	可配置	不可以配置
是否支持多处理器连接	并行总线支持 2 片	不可以
是否支持主机接口	支持并行接口	不支持

表 2.3 ADSP-21060 与 TMS320C40 的比较

特 性	ADSP-21060 (40 MHz)	TMS320C40(80MHz)
指令执行时间	25 ns	25 ns
峰值 MFLOPS	120	80
1024 点复数 FFT	0.46 ms	0.97 ms
除法 (Y/X)	150 ns	225 ns
数据寄存器数量	32	12
支持环形缓冲数量	32	1 (固定长度)
零开销的循环嵌套	6 级	1 级
是否支持 32 位浮点	支持	支持
是否支持 IEEE 数据格式	支持	不支持
是否支持扩展经断 (40 位) 浮点	支持	支持
定点乘法	32×32=>80 位	32×32=>32 位
DMA 通道数量	10	6
片外到片内传输速度	240 MB/s	320 MB/s
片内到片外传输速度	240 MB/s	160 MB/s
串行口数量	2 个, 传输速度 20 MB/s	无串行口
片内 SRAM 大小	128K×32 位字	2K×32 位字
存储区是否可配置 (程序与数据)	可配置	不可以配置

由以上三表的性能比较可以看出, 对于同一档次的处理器, SHARC 处理器明显比 TMS320 处理器具有更强的功能。特别由于 ADSP SHARC 系列处理器提供了片间无缝连接的能力, 使得在多 DSP 处理系统中, 外围控制逻辑也会变得十分简单, 更容易满足不同用

2.1.3 ADSP SHARC 系列处理器功能结构图

ADSP SHARC 系列处理器都以 ADSP-21000 处理器核为基础, 结构大体相同, 在程序代码设计上也具有高度的兼容性。图 2.2 和图 2.3 所示分别是 ADSP-2106x 与 ADSP-21160 的内部结构框图, 由图可知, ADSP SHARC 系列处理器具有以下基本特性。

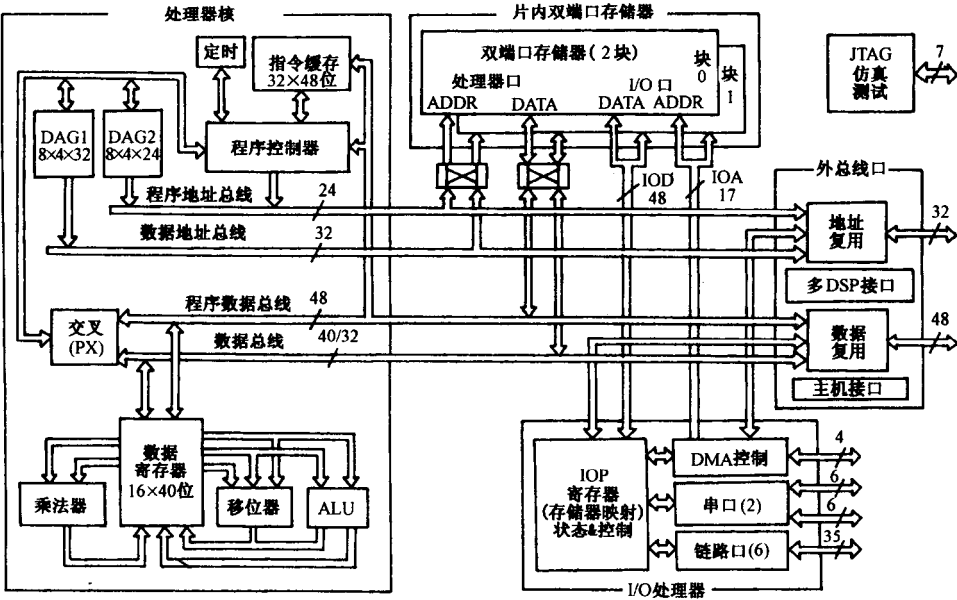


图 2.2 ADSP-2106x 内部结构框图

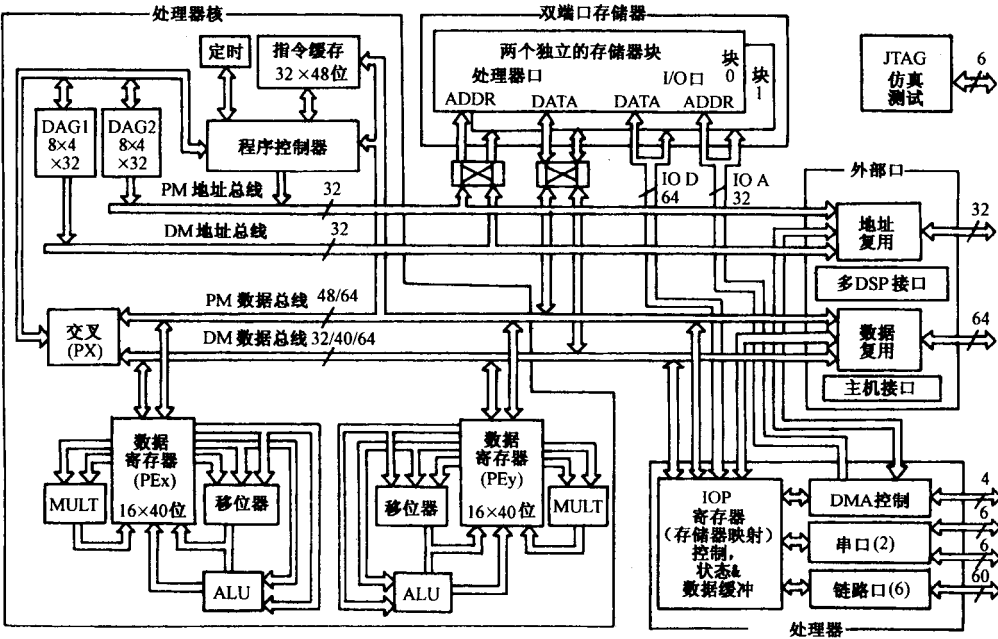


图 2.3 ADSP-21160 内部结构框图