

高等院校本科教材

供电子、信息控制、生物医学工程类专业用

XINBIAN WEIXING JISUANJI YUANLI YU JIEKOUJISHU

# 新编微型计算机原理 与接口技术

● 主 编 / 钱宗才



Xinbian weixing jisuanji  
yuanli yu jiekoujishu



第四军医大学出版社

# 新编微型计算机原理与接口技术

主 编 钱宗才

副主编 林永超 周智明

编 者 钱宗才 卢虹冰 钱芊宏 周智明 李志宏

第四军医大学出版社

## 内容简介

本书以 80486 微处理器为模型, 讨论了 Intel 系列 32 位微处理器的逻辑和编程结构、存储器管理以及在高档微机中使用的新技术; 介绍了 80x86 汇编语言完整的指令系统和程序设计方法, 并对汇编语言与高级语言的接口作了十分详尽的叙述; 在输入/输出接口技术中, 对 Intel 系列的通用接口芯片 8259A、8255A、8251、8253 / 8254、8237A 等可编程芯片进行了介绍并附有应用举例。本书在知识的结构上具有系统性和完整性, 内容由浅入深, 可作为大专院校电子、控制和生物医学工程本科生的教材, 对于学习和从事微机系统设计和应用的科技人员也是一本好的自学教材或参考书。

## 图书在版编目 (CIP) 数据

新编微型计算机原理与接口技术/钱宗才主编. —西安: 第四军医大学出版社, 2003.8

ISBN 7-81086-063-1

I. 新… II. 钱… III. 微型计算机—理论; 微型计算机—接口设备 IV. TP36

中国版本图书馆 CIP 数据核字 (2003) 第 065798 号

第四军医大学出版社出版发行

(西安市长乐西路 17 号 邮政编码: 710032)

电话: 029-3376765 (发行部) 029-3376763 (总编室)

传真: 029-3376764 E-mail: fmmup03@fmmu.edu.cn

西安力顺彩印有限责任公司印刷

\*

开本: 787×1092 1/16 印张: 28.25 字数: 680 千字

2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

印数: 1~3000 册 定价: 42.00 元

ISBN 7-81086-063-1/TP·2

## 前 言

《新编微型计算机原理和接口技术》是为电子信息类专业、工业自动化专业、生物医学工程专业等专业门类的本科生学习微型计算机原理和应用课程而编写的教材。在此之前,该课程的学习是以 8086/8088 微处理器为代表的 16 位微处理器和对应的微机系统为主。十多年来,微处理器从 16 位发展到 32 位,从 80386、80486 发展到今天的奔腾系列,其结构更加合理,功能不断增强;同时,与计算机相关的各个技术领域,如集成电路技术、存储器技术、总线技术以及系统软件技术都经历了飞速发展。上述变化迫切地要求对教材进行一次大的更新。

微型计算机原理与应用课程突出的教学特点是以应用为目标。学习和掌握微处理器的逻辑结构和工作原理,与学习和掌握面向处理器的汇编语言程序设计具有同等重要的地位。计算机技术是硬件和软件紧密结合的、专业性十分强的技术门类,软件依托硬件才能发挥其效能,硬件通过软件才能如虎添翼。所以,原理的学习以及语言的掌握与运用都是为应用服务的。课程的学习既然是以应用为目标,理论学习则必须与实验室的实习密切而有机地结合。理论学习只是对所学知识的理解,而应用是理论学习的延伸和拓展,必须通过实验或实习,才能使这种理解得以加深、升华,并且上升到一个理性的层次,才能事半功倍。

知识的学习过程是一个由浅入深、循序渐进的过程,微型计算机原理和应用课程的学习也不能例外。由于 80486 CPU 与奔腾 CPU 系列芯片之间保持了很好的一致性,并且起到了承上启下的作用,所以本教材以 80486 为原型,介绍 Intel 公司的 32 位 CPU 的编程结构、工作原理、系统组成和接口技术。同时又由于 32 位微处理器是在 16 位微处理器技术的基础上发展起来的,后者与前者之间具有很好的衔接,所以 16 位 CPU 结构原理的学习仍然是不可逾越的。学习 16 位 CPU 结构原理并不是浪费时间或多此一举,相反,它是学习和掌握 32 位微处理器工作原理的入门和基础。

一本好的教科书,在内容的安排上不仅应从初学者学习知识的角度出发,为其提供一条好的学习思路,还要有利于教师的教学组织。教学的内容和安排绝对不是内容的简单组合,而是要严格遵循教学规律,循序渐进,突出重点,理论学习和实验课程要紧密配合、环环紧扣,内容要前后照应。这是教科书和一般的书籍、技术资料之间的重要区别。本教材在内容的组织与安排上经过了反复推敲和精心设计,结合作者自己十多年的教学实践和经验,尽量地为教和学提供一个好的思路,既有利教师教学内容的组织,又顺应了教学的客观规律,使教学双方相得益彰。

本教材由钱宗才担任主编,林永超、周智明任副主编,由钱宗才编写第六、七、八、九章,卢虹冰编写第一、四、五章,钱芊宏编写第二、三章并负责全书的校对工作,周智明、李志宏编写第十章并承担附录的编撰工作。

智者千虑,难免一失。由于编者水平的限制和时间的仓促,书中出现的错误与不当之处,敬请读者批评指正。

编 者

2003 年 6 月于西安

# 目 录

|                          |        |
|--------------------------|--------|
| 第一章 微型计算机概述              | ( 1 )  |
| 第一节 概述                   | ( 1 )  |
| 第二节 微处理器的发展概况            | ( 2 )  |
| 第三节 微处理器与微型计算机的硬件组成和工作原理 | ( 3 )  |
| 一、微处理器                   | ( 4 )  |
| 二、总线                     | ( 4 )  |
| 三、存储器                    | ( 6 )  |
| 四、输入/输出 (I/O) 设备及其接口     | ( 6 )  |
| 五、微型计算机的工作原理概述           | ( 7 )  |
| 第四节 微型计算机的应用概述           | ( 9 )  |
| 一、科学计算                   | ( 9 )  |
| 二、过程控制                   | ( 9 )  |
| 三、信息管理和办公自动化             | ( 10 ) |
| 四、计算机辅助设计                | ( 10 ) |
| 五、人工智能和计算机仿真             | ( 10 ) |
| 六、计算机辅助教学                | ( 11 ) |
| 七、计算机在医学中的应用             | ( 11 ) |
| 第二章 数制和码制                | ( 12 ) |
| 第一节 数和数制及其转换             | ( 12 ) |
| 一、各种进位数制的表示方法            | ( 12 ) |
| 二、进位数制间的相互转换             | ( 13 ) |
| 第二节 二进制数的运算规则            | ( 17 ) |
| 一、二进制数的算术运算              | ( 17 ) |
| 二、二进制数的逻辑运算              | ( 17 ) |
| 第三节 有符号数的表示及其运算          | ( 18 ) |
| 一、符号二进制数的表示方法            | ( 18 ) |
| 二、符号二进制数运算中的溢出问题         | ( 21 ) |
| 第四节 实数的表示方法              | ( 22 ) |
| 一、定点方法                   | ( 23 ) |
| 二、浮点方法                   | ( 23 ) |
| 第五节 BCD 编码               | ( 25 ) |
| 第六节 ASCII 字符编码           | ( 26 ) |
| 第三章 8086 微处理器结构及系统组成     | ( 28 ) |
| 第一节 8086 微处理器结构及引脚功能     | ( 28 ) |
| 一、8086 微处理器的基本结构         | ( 28 ) |
| 二、8086 的寄存器组织            | ( 31 ) |

|                               |         |
|-------------------------------|---------|
| 三、8086 CPU 的引脚及功能 .....       | ( 33 )  |
| 第二节 8086 系统的存储器组织 .....       | ( 36 )  |
| 一、数据的存储格式和 8086 的存储结构 .....   | ( 36 )  |
| 二、存储器的分段和物理地址的形成 .....        | ( 38 )  |
| 第三节 8086 的 I/O 组织 .....       | ( 39 )  |
| 第四节 8086 最小方式和最大方式系统的组成 ..... | ( 40 )  |
| 一、8086 最小方式的系统组成 .....        | ( 40 )  |
| 二、8086 最大方式的系统组成 .....        | ( 42 )  |
| 第五节 8086 系统总线的时序 .....        | ( 45 )  |
| 一、最小方式下系统总线的时序 .....          | ( 45 )  |
| 二、最大方式下系统总线的时序 .....          | ( 49 )  |
| 第六节 Intel 80x86 微处理器简介 .....  | ( 52 )  |
| 一、80x86 CPU 的基本结构 .....       | ( 52 )  |
| 二、80x86 CPU 的工作方式 .....       | ( 54 )  |
| 第四章 8086 的寻址方式和指令系统 .....     | ( 56 )  |
| 第一节 8086 的寻址方式 .....          | ( 56 )  |
| 一、操作数的寻址方式 .....              | ( 57 )  |
| 二、程序转移地址的寻址方式 .....           | ( 59 )  |
| 第二节 8086 的指令编码格式 .....        | ( 60 )  |
| 第三节 8086 的指令系统 .....          | ( 62 )  |
| 一、数据传送类指令 .....               | ( 63 )  |
| 二、算术运算类指令 .....               | ( 69 )  |
| 三、逻辑运算类指令 .....               | ( 79 )  |
| 四、移位和循环移位指令 .....             | ( 81 )  |
| 五、字符串操作指令 .....               | ( 82 )  |
| 六、控制转移指令 .....                | ( 87 )  |
| 七、处理器控制指令 .....               | ( 99 )  |
| 第五章 汇编语言程序设计的基本方法 .....       | ( 101 ) |
| 第一节 汇编语言的基本概念 .....           | ( 101 ) |
| 一、机器语言、汇编语言和高级语言 .....        | ( 101 ) |
| 二、汇编语言的语句组成及种类 .....          | ( 102 ) |
| 三、常数、标号、变量和表达式 .....          | ( 104 ) |
| 第二节 汇编语言的程序设计 .....           | ( 109 ) |
| 一、汇编语言程序设计的步骤 .....           | ( 109 ) |
| 二、伪指令 .....                   | ( 110 ) |
| 三、汇编语言的程序结构 .....             | ( 118 ) |
| 四、DOS 功能调用 .....              | ( 145 ) |
| 五、BIOS 功能调用 .....             | ( 151 ) |
| 六、串操作 .....                   | ( 156 ) |
| 七、宏指令 .....                   | ( 160 ) |

|                      |       |
|----------------------|-------|
| 第三节 汇编语言和高级语言的程序接口   | (170) |
| 一、调用方式               | (171) |
| 二、嵌入汇编代码的方式          | (181) |
| 第四节 汇编语言的汇编、连接与调试    | (182) |
| 一、程序的编辑、汇编与连接        | (182) |
| 二、程序的调试              | (182) |
| 第六章 存储器组织            | (190) |
| 第一节 存储器概述            | (190) |
| 第二节 随机存储器 RAM        | (192) |
| 一、静态 RAM             | (192) |
| 二、动态 RAM             | (194) |
| 三、新型 RAM 器件简介        | (198) |
| 第三节 只读存储器 ROM        | (198) |
| 一、掩模 ROM             | (198) |
| 二、可编程 ROM (PROM)     | (199) |
| 三、可擦可编程只读存储器 (EPROM) | (200) |
| 四、只读存储器芯片举例          | (201) |
| 第四节 CPU 与存储器的连接      | (202) |
| 一、连接时要考虑的问题          | (202) |
| 二、存储器容量的扩充           | (204) |
| 第五节 高速缓冲存储器 (Cache)  | (207) |
| 一、概述                 | (207) |
| 二、片内高速缓存             | (208) |
| 三、外部高速缓存             | (211) |
| 第六节 外部存储器简介          | (212) |
| 一、硬盘存储器              | (212) |
| 二、光盘                 | (214) |
| 第七章 80x86 微处理器及多任务机制 | (218) |
| 第一节 概述               | (218) |
| 第二节 80486 的寄存器结构     | (219) |
| 一、基本体系结构寄存器          | (219) |
| 二、系统级寄存器             | (221) |
| 第三节 80486 的外部引脚功能    | (223) |
| 一、地址总线               | (223) |
| 二、数据总线               | (223) |
| 二、控制总线               | (224) |
| 第四节 80486 的存储器管理     | (227) |
| 一、80x86 的存储器组织和地址空间  | (227) |
| 二、存储器的分段管理           | (228) |
| 三、存储器的分页管理           | (235) |

|                            |       |
|----------------------------|-------|
| 第五节 多任务及保护                 | (238) |
| 一、多任务及其转换                  | (238) |
| 二、保护                       | (241) |
| 三、保护方式下的控制转移               | (246) |
| 第六节 80x86 的寻址方式及指令系统       | (251) |
| 一、80x86 的数据类型和全地址          | (251) |
| 二、80x86 指令的寻址方式            | (252) |
| 三、80286 增强与增加的指令           | (254) |
| 四、80386 / 80486 增强与增加的指令   | (259) |
| 第八章 输入/输出接口基础              | (264) |
| 第一节 I/O 寻址方式和指令            | (264) |
| 一、端口地址和编址方法                | (264) |
| 二、输入/输出指令                  | (265) |
| 第二节 输入/输出方式                | (267) |
| 一、输入/输出的同步                 | (267) |
| 二、数据输入/输出的基本方式             | (267) |
| 第三节 I/O 端口地址的分配及译码         | (270) |
| 一、端口地址的分配                  | (270) |
| 二、端口地址的译码                  | (271) |
| 第四节 PC 机的系统级总线及资源          | (273) |
| 一、PC / XT 总线               | (273) |
| 二、AT 总线 (ISA 总线)           | (276) |
| 三、PC 系统的接口资源               | (279) |
| 第五节 80x86 微机中的中断系统         | (281) |
| 一、中断的概念和分类                 | (281) |
| 二、实模式下的中断和异常处理             | (283) |
| 三、保护模式下的中断和异常处理            | (286) |
| 四、可编程中断控制器 8259A           | (288) |
| 五、中断的应用及举例                 | (299) |
| 第九章 输入/输出接口技术              | (302) |
| 第一节 并行接口电路                 | (302) |
| 一、并行接口概述                   | (302) |
| 二、可编程并行接口芯片 8255A          | (303) |
| 三、8255A 的应用举例              | (310) |
| 第二节 可编程定时器/计数器 8253 / 8254 | (314) |
| 一、8253 的组成及功能              | (314) |
| 二、8253 的工作方式及编程            | (316) |
| 三、8253 的应用                 | (323) |
| 四、8254 可编程定时器/计数器          | (325) |
| 第三节 串行通信和 8251 串行接口        | (326) |

|                           |       |
|---------------------------|-------|
| 一、基本概念                    | (326) |
| 二、可编程串行通信接口 8251A         | (328) |
| 三、8251A 的编程               | (333) |
| 四、RS-232C 标准              | (336) |
| 五、8251A 的应用               | (338) |
| 第四节 DMA 传送和 DMA 控制器 8237A | (341) |
| 一、概述                      | (341) |
| 二、DMA 控制器 8237A           | (342) |
| 三、8237A 的应用与编程            | (351) |
| 第五节 多功能接口芯片 82380 简介      | (355) |
| 一、82380 的内部功能结构           | (355) |
| 二、82380 的主要功能             | (356) |
| 第十章 常用外设的输入/输出接口技术        | (358) |
| 第一节 概述                    | (358) |
| 第二节 并行打印机及接口              | (359) |
| 一、打印机接口的信号和时序             | (359) |
| 二、打印机接口的逻辑组成及编程           | (361) |
| 第三节 鼠标器及其编程               | (364) |
| 一、鼠标器工作原理及接口              | (364) |
| 二、鼠标器的编程                  | (365) |
| 第四节 显示系统及其编程              | (370) |
| 一、显示器系统的组成                | (370) |
| 二、VGA 显示模式与编程接口           | (372) |
| 三、标准 VGA BIOS 功能调用        | (377) |
| 四、VESA 扩展 VGA BIOS 功能调用   | (384) |
| 附录                        |       |
| 附录 I ASCII 编码表            | (389) |
| 附录 II DOS 功能调用表 (INT 21H) | (390) |
| 附录 III 汇编语言的使用及存储模型       | (402) |
| 一、汇编语言的使用                 | (402) |
| 二、汇编语言的存储模型               | (403) |
| 附录 IV 通用汇编语言的指示符 (即伪操作符)  | (405) |
| 附录 V DEBUG 命令一览表          | (407) |
| 附表 V-1 DEBUG 命令的格式及功能说明   | (407) |
| 附表 V-2 DEBUG 命令的使用举例      | (408) |
| 附录 VI 80x86 指令系统说明        | (410) |
| 参考文献                      | (441) |

## 第一章 微型计算机概述

学习本教材的读者,应该具有计算机应用基础、电路、模拟电子技术和数字电子技术的基础知识,并且已经掌握 1 至 2 门计算机编程语言,具有一定的硬件基础和软件编程能力。同时,在计算机应用不断拓展、知识不断更新的今天,在微型计算机工作原理和应用的层面上进行深入一步的学习是十分必要的。

### 第一节 概 述

本书的前半部分主要围绕微处理器的结构,即对微处理器的内部、外部结构和寄存器组织方式进行解析,有利于学生对微机工作原理深入了解;本书后半部分结合对汇编语言和程序设计方法、对微处理器可编程的外围接口芯片编程的学习,加上一定学时的上机实习与实验,可以逐步地掌握微型计算机软、硬件应用与设计的基本技能,为以后在实际工作中进行微型计算机的应用与开发打下一个扎实的基础。

完整的微型计算机系统应该包括硬件(Hardware)和软件(Software)两部分。所谓硬件,是指微型计算机的物理实体。软件主要是指程序,不仅包括操作系统程序、调试程序、连接程序、文字编辑程序和高级语言的编程环境等,还包括程序员为解决特定问题而编制的应用程序。

在微型计算机的硬件组成中,微处理器是计算机的核心部件,也叫中央处理单元(Central Processing Unit,简称 CPU)。它是由计算机的控制单元和运算单元集成在一个半导体芯片上实现的。内存储器是独立于 CPU 之外的芯片或芯片组。输入/输出的控制接口电路也常常做到一个芯片上,通常叫做外围接口芯片。微处理器、内部存储器、接口电路和输入输出外部设备共同组成了微型计算机的硬件部分。

计算机软件分为两类:即系统软件和应用软件。系统软件是用来调度、监控和管理计算机中所有硬件和软件资源并提供友好的人机界面,为应用程序的开发、调试、运行提供一个良好的环境。系统软件主要包括操作系统,程序设计语言的开发工具(如 FORTRAN、Visual Foxpro、C++、JAVA、Visual C++、Visual Basic 等)以及诊断程序、调式程序和编辑程序等。应用程序通常是指使用计算机系统的用户为解决自己特定问题而开发或购买的程序。计算机功能的强弱和好坏,不仅取决于计算机硬件的性能,也取决于软件的功能和性能。

操作系统(Operating System,简称 OS)是最基本的系统软件,在所有软件中有着特殊的地位。操作系统最贴近计算机的硬件,它是管理、监控和维护计算机硬件和软件资源的软件,所有其它的软件,如高级语言的集成开发环境、各种工具软件 and 应用程序,都必须以操作系统作为基础。操作系统要面对用户,要接纳用户的命令并进行适当的处理,因此它还必须为用户提供一个良好的人机界面。操作系统担负着内部存储器的管理、设备的输入输出管

理、处理器的管理、目录和文件的管理以及作业管理等许多纷繁复杂的任务。在微机上常用的操作系统有微软公司 (Microsoft Corporation) 的 MS DOS (Microsoft Disk Operating System) 磁盘操作系统和视窗操作系统 (包括 Windows 3.x、Windows 95 /98、Windows Me、Windows 2000 以及 Windows XP 等各种不同的版本)。

## 第二节 微处理器的发展概况

计算机技术在最近 20 多年间的迅速发展以及它给各行业带来的变化,是有目共睹的事实。计算机技术的不断发展和成熟体现在计算机的体系结构的日趋复杂和完备,硬件和软件性能的不断改善与增强。其中,微处理器的诞生、发展和不断地更新换代起着决定性的作用。本节就微处理器的发展概况做简要的回顾。

世界上第一个微处理器芯片是 Intel 公司 1971 年制造的 Intel 4004。事实上,这只是一个数据宽度为 4 位的单片可编程控制器芯片,只能寻址 4096 个存储单元。4004 的指令系统只有 45 条指令,指令执行速度为每秒 50 000 条指令。

1972 - 1973 年间, Intel 公司将 4004 的数据位扩展到 8 位,地址位也进行了扩展,先后推出了 8008 和 8080 芯片。8080 是技术上很成功的一个 8 位微处理器产品,它可以寻址 64K 字节的存储空间,指令系统也得到了扩展,且指令周期缩短到 2 微秒 ( $\mu\text{s}$ ),比起它的早期产品,运行速度提高了一个数量级。

1978 年, Intel 公司推出了被称为第三代微处理器的 8086 和 8088 芯片。这两个版本的芯片都是 16 位微处理器,指令周期进一步缩短到了 400 纳秒 (ns),相当于 2.5 MIPS (Million Instructions Per Second)。8086 和 8088 都采用了超大规模集成电路技术 (Very Large Scale Integration, 简称 VLSI),集成度比 8 位微处理器提高了一个数量级;它们的地址总线为 20 位,可以寻址 1M 字节存储器空间,数据总线也从 8 位扩展到 16 位,时钟频率达到 5 ~ 10 MHz,指令系统得到进一步扩展,比起 8 位微处理器,它们的性能和速度得到了大幅度的提升。

80286 是 16 位微处理器的改进版本,除了可寻址 16M 字节存储空间外,其它方面几乎和 8086 完全一样。

80386 是 Intel 公司在 1986 年推出的第一个 32 位的新一代微处理器,它具有 32 位的地址总线和 32 位的数据总线。由于具有 32 位地址总线,80386 理论上可以寻址 4GB (1GB = 1024MB) 字节的存储空间。80386 的外部数据总线宽度从 16 位增加到了 32 位,并且在处理器内部可以直接处理数据宽度为 32 位的操作数 (如长整数和实数数据类型),使得微处理器对数据的传送速度和处理能力有了很大的提高。不仅如此,80386 处理器内部增加了存储管理部件,它允许操作系统通过存储管理部件的硬件结构逻辑对计算机的存储器资源进行分配和管理,这样既减少了软件的开销,又提高了微处理器的工作效率。

80386 的指令系统保持了与 8086、8088 和 80286 微处理器的兼容,又增加了访问 32 位寄存器和存储器管理的指令,所以原来使用的 16 位软件仍然可以在 80386 的微机上运行。

Intel 公司在 1989 年推出的 80486 是又一种新的 32 位微处理器,它把 80386 微处理器和 80387 的数学浮点协处理器两者做在一个硅片上,还增加了 8K 字节的高速缓冲存储器。80486 对原来的内部结构作了修改,结构更加合理,使得几乎有半数的指令可在一个时钟周期内执行完。一种叫做 80486DX 的 80486 的改进版本,可以在双倍时钟频率下工作,叫做倍

时钟型,它使得微处理器的运行速度增加了一倍。甚至还有三倍时钟型的 80486 DX4 微处理器芯片,它的运行速度甚至超过了早期的 Pentium (奔腾处理器) 芯片。

1993 年 8 月, Intel 公司推出了第五代的微处理器 Pentium。与 80486 微处理器相比较, Pentium 处理器采用了新的体系结构。例如,它具有 2 个独立的超标量技术 (Superscaler technology) 的整数处理器,允许微处理器可在一个时钟周期内执行两条整数指令;采用跳转预取技术,加快了循环程序的运行速度;它还将内置的高速缓冲存储器 (Cache) 增加到 16K,并组织成为两个独立的 8K 缓存,分别作为指令 Cache 和数据 Cache,大大提高了 Pentium 微处理器的性能。与 486 处理器一样, Pentium 微处理器仍采用内置的浮点协处理器,但在硬件上进行了新的设计,从而使得其浮点执行速度提高了大约 5 倍。

Intel 公司在近年内先后发布了 Pentium Pro、MMX Pentium、Pentium II、Pentium III 等一系列微处理器。

Pentium Pro 是 Pentium 微处理器的改进版本,除了 16K 的一级 Cache (也叫 L1 Cache) 外,另外还有一个独立于 CPU 的 256K 二级高速缓存 (即 L2 Cache),通过 64 位专用总线与 CPU 相连。Pentium Pro 比 Pentium 微处理器更先进,主振频率为 200 ~ 250MHz。MMX Pentium 也叫多能奔腾处理器,是在 Pentium 处理器的基础上进行 MMX 多媒体功能扩展后推出的一种改进版本。MMX 多媒体功能扩展技术是在微处理器的内部加入 57 条 MMX 扩展指令,用以提升音频和视频的处理能力和速度。MMX Pentium 微处理器在运行多媒体软件时,速度可以提高大约 50%。Pentium II 微处理器简称 P II,结构与 Pentium Pro 基本上相同,但一级高速缓存增加了一倍,指令 Cache 和数据 Cache 各为 16K,从而减少了访问二级 Cache 的频度,提高了效率。P II 处理器内部也加入了 57 条 MMX 扩展指令,从而提升了音频和视频的处理速度。Pentium III 微处理器简称 P III,是更为先进的新一代微处理器,采用 0.25 微米的制造工艺,具有新的 SECC2 插口。与 Pentium II 微处理器相比,它的二级高速缓存增加到了 512K, CPU 的主频的起始频率为 450 ~ 500MHz,总线频率达到 100MHz。P III 微处理器还增加了 70 条指令的 SSE 指令集,其中 8 条指令用于对内存连续数据流进行优化及处理,50 条指令是单指令多数据 (SIMD) 的浮点运算指令,12 条指令是经过算法改进后的新的多媒体指令,从而使图形、图像处理的速度和质量都有了新的提高。

Pentium 系列的微处理器和 80386、80486 相比较,不仅时钟频率大幅度提升,工艺和集成度更加先进,而且采用了许多先进技术,如超标量流水线技术、二级高级缓存技术、更合理的浮点部件设计、动态分支预测技术、MMX 多媒体功能扩展和 SSE 指令集扩展技术等。Pentium 系列微处理器的外部数据总线宽度为 64 位,总线的数据传送速度和能力更高更强;地址总线为 36 位,可寻址的内存空间更加开阔。但 Pentium 系列微处理器的寄存器结构和内部总线的宽度为 32 位,与 80386、80486 相同,因此它们仍然属于 32 位微处理器的范畴。

### 第三节 微处理器与微型计算机的硬件组成和工作原理

关于计算机的组成,我们传统地把它分为五个部分:运算器、控制器、存储器、输入设备和输出设备 (即 I/O 设备) 及其接口电路,这种划分能够清楚地体现出计算机各个部分的功能和逻辑关系,图 1-1 给出了微型计算机组成的框图。

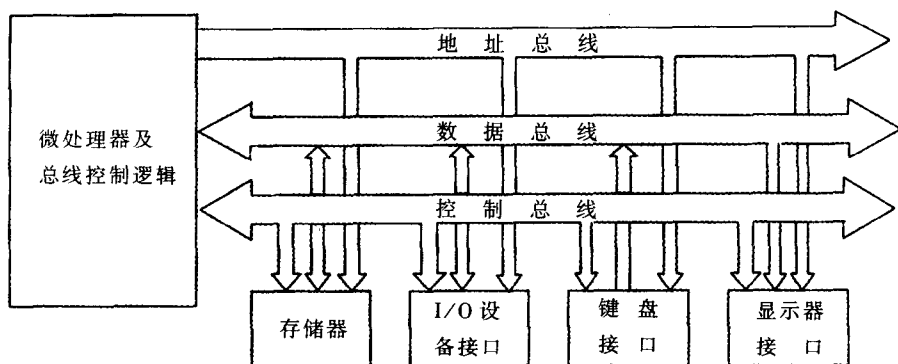


图 1-1 微型计算机组成框图

## 一、微处理器

微处理器是微型计算机的核心部件，是将计算机中的运算单元和控制单元集成在一个大规模集成电路的芯片中，在微机系统中起着控制和运算中心的作用，因此也叫做中央处理单元。CPU 通过被称为系统总线的一组连线，和存储器、输入/输出接口部件相连接，也就是说，所有的存储器、输入输出设备接口和微处理器都毫无例外地挂接在系统总线上。

微处理器的基本功能之一是完成程序中各种指令所界定的基本操作。它必须具有以下基本功能：

- 对指令进行预取、寄存、译码并执行；
- 执行加、减、乘、除的算术操作和与、或、取反和求补的逻辑操作；
- 可暂存少量数据；
- 执行对存储器的读/写操作；
- 执行与输入/输出接口之间的数据传送操作；
- 提供系统所需要的各种控制信号和定时信号；
- 能够对外部的各种请求信号给予响应（如中断请求、总线请求等）；

从程序流向控制的角度考虑，微处理器还应该能够：

- 执行无条件转移的操作；
- 处理关系和逻辑表达式并执行条件转移；
- 处理循环操作和执行子程序等。

微处理器执行算术逻辑运算和进行信息的传输操作与总线的的数据宽度有关。早期的 8086 和 80286 微处理器只能直接处理 8 位数和 16 位数，而不能直接处理 32 位数据。只有 80386、80486 和 Pentium 微处理器具有直接处理 32 位数据的能力。而 80486 以上的微处理器具有浮点处理以及可高速缓冲的能力。从 80486 开始到目前的 Pentium 处理器，数学协处理器成为内置结构，用于进行复杂的浮点数学运算；而且其浮点操作与运算逻辑的操作以并行的方式执行。而 386 以前版本的微处理器，数学协处理器还只是作为一般附加的可选部件而非必需的部件。

## 二、总线

采用总线结构是微型计算机系统的基本特点之一，总线是微型计算机沟通与外部部件或

输入/输出设备之间联系的必要通道。采用总线结构的优点在于,简化了各个模块之间的连线结构,使系统便于扩充,使系统结构实现模块化。

根据总线的传送方向,我们可以将总线分为单向总线和双向总线。单向总线允许挂接在总线上的某些主控部件或设备向另一些部件或设备有选择性地传送信息;而双向总线则允许挂在总线上的并且获得总线使用权的某一部件或设备与总线上的另一部件或设备之间发送/接收信息。

根据总线上传送信息种类的不同,人们常常把系统总线分为三种类型,即数据总线(Data Bus)、地址总线(Address Bus)和控制总线(Control Bus)。

- 地址总线:传送访问存储器或 I/O 设备的地址信息;
- 数据总线:传送指令代码信息或数据信息;
- 控制总线:接受来自微处理器外部的请求、发出来自微处理器的各类控制和响应信号,传送状态信息。

### 1. 地址总线

地址总线用来选中存储器的一个单元或 I/O 设备的一个端口寄存器,它是由 CPU 发出、送往存储器或 I/O 设备端口的单向总线。

地址总线的宽度与微型计算机对存储器的寻址空间有关,如 8086/8088 有 20 位地址线,可以寻址从 00000H~0FFFFFFH 的 1M 字节的存储空间;80286 有 24 位地址线,因而可以寻址 000000H~0FFFFFFFH 的 16 M 字节的存储空间;80386 和 80486 有 32 位地址线,可以访问的地址间从原理上为 00000000H~0FFFFFFFFFH 的 4 GB;Pentium 微处理器的微机一样也寻址 4GB 的存储器地址空间。地址总线与 I/O 设备的端口寻址也有关系,如所有的 80x86 处理器均使用地址总线的低 16 位对 I/O 端口寄存器寻址,因此,包括 8086~80386、80486 和 Pentium 微处理器在内,它们的 I/O 地址空间都应 64K 字节。

### 2. 数据总线

数据总线用于在微处理器与存储器或 I/O 接口之间传送信息(包括数据信息或代码信息),数据总线是双向的,而且具有三态控制功能。

数据总线的宽度决定了 CPU 和存储器或 I/O 设备接口之间完成一次传送的最大数据宽度即比特数。一般说来,各种不同版本的 80x86 微处理器的数据总线的宽度是不相同的。例如 8086 和 80286 CPU 的数据总线是 16 位,与它的内部寄存器的宽度相同,因此,8086 是 16 位的微处理器。80386 和 80486 的数据总线宽度为 32 位,而 Pentium 处理器的寄存器结构和内部总线的宽度为 32 位,但是它的外部数据总线宽度却是 64 位。数据总线宽度不断增宽的这种趋势,主要是针对微处理器与外部设备之间的信息交换的瓶颈问题。比如,传送一个单精度的浮点数(32 位),8088(其数据总线为 16 位宽度)需要 4 次传送才能完成,而 386 以上的微处理器只需传送 1 次即可。在日常要处理的数据中,包括长整数和浮点数的数据类型在内,绝大多数是 32 位的。

### 3. 控制总线

控制总线用来传输来自外部设备的状态和请求信号,以及微处理器对外部逻辑的控制和应答信号。

在大多数微机中都有  $\overline{\text{MRDC}}$  (存储器读控制信号)、 $\overline{\text{MWTC}}$  (存储器写控制信号)、 $\overline{\text{IORC}}$  (I/O 读控制信号)、 $\overline{\text{IOWC}}$  (I/O 写控制信号)和中断请求、中断响应信号,这些信号线都

属于控制总线。例如,当微处理器要读取存储单元的内容时,先通过地址总线发出地址信号去选中目的存储器单元,接下来微处理器向选中的目的存储单元发送存储器读控制信号  $\overline{\text{MRDC}}$  (低电平有效),最后读出的数据通过数据总线送往微处理器。

### 三、存储器

存储器是微型计算机硬件结构中的重要组成部分,用于存储程序代码和数据信息。有了存储器,计算机才具有存储和记忆的功能。存储器可分为外部存储器和内部存储器。外部存储器主要是指硬盘驱动器、软盘驱动器和 CD ROM 等,可以提供大容量、永久性的存储功能,是非易失性存储器。对于外部存储器,CPU 不能直接地使用指令进行读/写操作。内部存储器采用半导体存储器件,即采用大规模或超大规模集成电路技术做成的半导体存储器芯片。

内部存储器简称内存,又称主存。主要用于存放驻留于内存的程序和数据,以及正在执行中的程序和数据,容量相对较小,但存取速度快。内部存储器按存取方式的不同又可以分为两种类型:随机存取存储器(Random Access Memory,简称 RAM)和只读存储器(Read Only Memory,简称 ROM)。

#### 1. 随机存取存储器 RAM

随机存取存储器指在机器运行期间可以进行读操作和写操作的存储器。用户可以随时通过写操作指令,把有关的程序和数据写入到 RAM 中,在需要的时候可以进行读出或写入。当机器断电后,RAM 中的程序和数据将自然丢失,因此,RAM 是易失性存储器。在当前的 PC 机中广泛使用一种称为 SIMM (Single Inline Memory Module) RAM,它是将多个存储芯片做在一条窄电路板上形成的存储器插条,因此占据空间较小且存储容量较大。DIMM RAM 则是 SIMM 的改进型。这两种存储条的引线规格不同,因此,在选择存储器插条时应根据插槽的形式来选择它们。

#### 2. 只读存储器 ROM

只读存储器是指在机器运行期间只能读出信息、不能写入信息的存储器。ROM 中的信息不会消失,即使断电也不会丢失,通电后即可使用,是非易失性的存储器件。只读存储器一般用来存放固定不变的程序或数据,例如基本的输入输出程序(BIOS)、设备的启动程序和计算机的引导装入程序等。当对计算机进行加电时,计算机便立即进入并执行 ROM 中的引导装入程序,把存放在硬盘或者软盘上的程序和数据装入到内存中,并启动该程序的运行。只读存储器有 3 种类型:掩模型只读存储器(掩模 ROM)、可编程只读存储器(PROM)和可擦可编程只读存储器(EPROM)。可擦可编程只读存储器 EPROM 是使用十分灵活和方便的只读存储器,当信息被一次写入后,如果需要,可以用特定的方法进行擦除操作,根据需要再写入新的内容。因此,EPROM 在实际中得到了广泛的应用。

根据制造工艺的不同,半导体存储器又可分为双极型 TTL 逻辑、射极耦合逻辑 ECL、NMOS、CMOS、HMOS 等几种不同的电路形式。

### 四、输入输出(I/O)设备及其接口

输入和输出(I/O)设备是微型计算机系统的重要组成部分。

#### 1. 输入设备

输入设备的任务是将原始数据、现场信息转换成计算机所能识别的形式输入到计算机

中。如温度、湿度、位移、压力、速度和流量等,这些物理量先要通过传感器转换为电压或电流的形式,再经过放大和模拟/数字(A/D)转换过程转变成数字量,才能输入到计算机进行存储和适当的处理。常用的输入设备有键盘、鼠标器、扫描仪、数字化仪和A/D转换器等。

## 2. 输出设备

输出设备的任务是接受计算机输出的各类数据和信息,以人们所能接受和理解的形式表示出来。微型机中常用的输出设备包括显示器、打印机、绘图仪、扫描仪和D/A转换器等。

有的设备,如磁盘机、磁带机和可读写光盘,既是输入设备,又是输出设备。

## 3. I/O 接口电路

由于输入/输出设备的种类很多,有电子式、电磁式和机械式,其原理、结构各异,而且一般多为慢速设备。它们与主机之间在信号的形式、速度、工作电压、逻辑电平和时序等方面都存在着差异,不能达到匹配和兼容,因此微型计算机与I/O设备之间无法直接地进行信息的交换。一般地说,I/O接口电路(I/O Interface)是指在主机的系统总线与外设之间的一种具有特殊功能和作用的适配电路(I/O Adapter),其作用是实现主机与外设之间在信号形式、速度、工作电压、逻辑电平和时序等方面的匹配,从而实现主机与外设之间信息的顺利传输和变换。这种I/O接口电路通常做成板卡的形式,插在系统总线(或输入/输出总线)的插槽上,我们将它们叫做I/O接口板。例如,显示适配器(即显示卡)是主机与显示器之间的I/O适配电路。I/O接口电路的种类很多,如打印机接口卡、显示卡、网络适配器(即网卡)、图像采集卡、音频适配器(即声卡)等。

在主机板上,微型计算机还提供了在微处理器与输入/输出总线之间的可编程外围接口电路的硬件资源。其中常用的接口电路有8255可编程并行接口电路、8259可编程中断控制电路、8253/8254可编程定时/计数器电路、8237直接存储器存取电路、8250可编程串行接口电路以及82380多功能的I/O接口电路等。

# 五、微型计算机的工作原理概述

上面我们介绍了微型计算机的几个重要组成,这些组成部分构成了微型计算机的硬件基础。但是,仅靠计算机的硬件,计算机是无法做任何有效的工作的。计算机只有在软件的配合和指挥下,才能进行诸如数学计算、数据传送、信息处理等有意义的工作。所谓软件,是指各种程序的总称。为了说明计算机的工作过程,先要明确几个重要概念。

## 1. 指令和程序

微型计算机的工作过程就是执行程序的过程。一个程序,无论它有多么复杂,都是由各种指令组成的。指令就是组成程序的基本单位,而程序则是为了执行特定的功能和任务,由微处理器的各种指令所组成的一定的指令序列。

前面,我们在微处理器一节中提到了微处理器的基本功能之一是完成程序中各种指令所规定的基本操作。例如,在微处理器中,算术逻辑单元ALU作为运算器的核心部件,在控制信号的驱动下,可以进行两个数的加、减、乘、除、和逻辑与、逻辑或的操作,还可以对一个数进行取反、求负和移位的操作。除了算术和逻辑运算之外,微处理器还可以执行寄存器与寄存器之间、寄存器和存储器之间的数据传送操作,对I/O接口电路的输入/输出的操

作,以及对处理器本身的控制操作等等。事实上,微处理器所能执行的每一种基本操作都对应着一条指令。所以,对微处理器而言,指令就是微处理器所能执行的每一种基本操作。理解这一点,对进一步理解微机的工作原理、学习汇编语言程序设计是十分必要的。

对微处理器来说,它的指令的种类及数量是一定的,指令的种类多少反映了微处理器本身操作和处理能力的强弱。对于一种微处理器而言,它的全部指令的集合构成了它的指令系统 (Instruction Set)。所以,指令系统是微处理器功能的具体体现。

对于一条指令来说,有两个问题需要明确。第一个问题是指令在机器中是以怎样的形式体现的。我们知道,数字计算机属于二进制系统,它只能认识二进制码,机器中所有的信息都是以二进制表示的,所以,微处理器指令系统中的每一条指令也是以二进制的形式来表示的,叫做机器指令。对于任何一条机器指令来讲,都具有与其它指令不同的二进制代码,我们称之为指令码。指令码通常是由操作码 (Operation Code) 和操作数 (Operand) 这两部分组成的,操作码部分用来表示指令所要执行的操作信息,说明微处理器执行的是什么操作;而操作数部分则用来描述指令的操作对象。例如,在执行一条加法指令时,操作数部分不仅要说明并指出加数 1、加数 2 分别来自何处,还要指出得到的和数送往何处。由于指令的不同,有的指令可以没有操作数,而有的指令可以有一个、两个或者三个操作数不等,这要视具体的指令而定。第二个问题是对于有操作数的指令来说,怎样去获取操作数,这就涉及到指令的寻址方式了,关于这部分内容,我们将在第四章作详细的讨论。

前面提到,所谓程序是指为执行特定的功能和任务,由微处理器的各种指令所组成的一定的指令序列。为了运行程序,必须先把程序的每一条机器指令逐条地准确无误地输入到计算机的存储器中,叫做手编程序。做这样的工作实在是麻烦而费功夫的事情,而且容易出错,可是,早期的程序员确实是这么做的。为了便于理解、记忆和书写,人们发明用字母表示的助记符 (一般为英文单词或缩写) 来代替操作码,用符号表示操作数。这种指令叫汇编语言指令。例如:MOV AX, 1234 指令,其中的 MOV 是操作码助记符,AX 和 1234 分别是两个操作数。显然,对于程序员来说,汇编语言指令要比机器指令更加直观和容易理解。

## 2. 计算机工作过程概述

计算机的工作过程就是执行程序的过程,如图 1-2 所示。

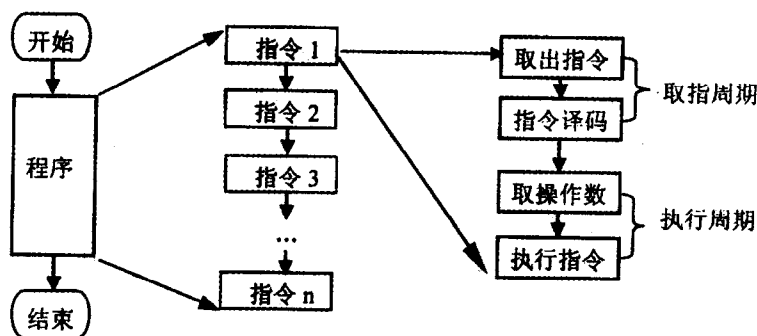


图 1-2 微型计算机程序执行过程示意图

在前面,我们说明了计算机中的内部存储器主要是用来存放驻留于内存的程序和数据,以及正在执行中的程序和数据。为了不至于引起混乱,程序的二进制代码和程序使用的数据信息是分开存放的。在指令和程序一节中我们已经明确了程序的概念,即为了执行特定任