

# 80X86

# 微型计算机原理 及应用

● 易仲芳 主编



電子工業出版社

# 80X86 微型计算机 原理及应用

易仲芳 主编

电子工业出版社

180120

(京)新登字 055 号

### 内 容 提 要

本书是依据国家教委工科计算机基础课程指导委员会制定的“高等学校工科非计算机专业《微机原理及应用》课程基本要求”编写的。

鉴于微型计算机的高速发展,本书以 Intel 80X86 微处理器为背景,全面介绍了计算机基础、微机计算机组成、工作原理、CPU 功能结构、寻址方式和指令系统、汇编语言程序设计、存储器及其与 CPU 连接、输入输出与中断系统、接口技术、80X86 为 CPU 的计算机系统、微机应用系统开发技术和 PC DOS 操作系统。

本书可作为高等学校工科非计算机类专业的教材,也可作为大专类相应专业教材或参考书,同时也可作为从事计算机工作的工程技术人员自学或参考用书。

80X86 微型计算机原理及应用

主 编:易仲芳

责任编辑:杨逢仪

\*

电子工业出版社出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行 各地新华书店经销

北京市顺新印刷厂印刷

\*

开本:787×1092 毫米 1/16 印张:26.25 字数:634 千字

1995 年 4 月第一版 1995 年 4 月第一次印刷

印数:5000 册 定价:24.00 元

ISBN7-5053-2889-1/G·242

# 前 言

本书是依据国家教委工科计算机基础课程指导委员会制定的“高等学校工科非计算机专业《微型计算机原理及应用》课程教学基本要求”编写的。

微处理器及微型计算机由于价格低廉,性能不断完善,使得它们的应用领域越来越广。不仅应用到科学计算、辅助设计、信息处理、过程控制、程控交换、仪器和仪表,也进入了民用产品和家用电器,乃至千家万户。与此同时,需要掌握和使用计算机,特别是微型计算机的人也越来越多。因此,各高等学校所开设的“微型计算机原理及应用”已被列为重点基础课程之一。

由于历史原因,“微型计算机原理”课程多以8位微处理器为中心来组织安排教学,常见书籍亦是如此,通过近十年来计算机技术和超大规模集成电路发展,16位、32位微处理器取得了长足进展。市场竞争的结果,保留了两大系列: Intel 8086/80286/80386/80486 (80X86) 及 Motorola 的 68000/68020/68030/68040 (680X0)。而以 Intel 80X86 为 CPU 的微型计算机系统,因其先进的系统结构设计(采用 Over Drive 方案,可方便简捷地升级换代)、丰富的系统软件和实用软件,而成为当今世界上最有代表的主流机型,因而它拥有的用户,在计算机世界占有绝对优势。本书基于这个出发点,结合 80X86 微处理器来讲述微型计算机组成、工作原理、CPU 功能结构、寻址方式和指令系统、存储器及其与 CPU 连接、汇编语言程序设计、接口技术,最后讲解 80X86 微型计算机硬件系统、相应的总线系统及 PC DOS 操作系统,以求达到从理论和实践上,掌握微型计算机的基本组成、工作原理、接口电路及硬件的连接、基本操作和汇编语言编程技术,最终达到开发、应用各类微型计算机的目的。

本书是工科非计算机类“微型计算机原理及应用”的本科生教材。也可作为大专类相应专业的教材或参考书。书中内容是笔者参阅了当前国内外有关微型计算机的大量资料和多年从事教学、科研工作总结综合而成。选材上注重基础、实用、先进和系统性;叙述上,力求做到由浅入深、文字通俗易懂,并在各章后附有适当思考题与习题,以利于教学和学生自学;教材组织上,按先易后难,循序渐进的原则。本书参考学时 70~80 学时。考虑到工科非计算机专业差别较大,即使是同一类专业名称,各专业、各学校教学风格也不尽相同,为此,在使用本教材时应根据具体情况适当增删部分内容。最后必须指出,“微型计算机原理及应用”是一门实践性很强的课程,应特别重视实践性环节和上机能力的培养。

本书共九章,由易仲芳主编,并编写第一、二、五章,绪论和附录,第三、四章由吴宁编写,第六章由邢岩编写,第七、八章由毛建国编写,第九章由杨贤芳编写。全书由易仲芳统稿。

本书在出版过程中,得到电子工业部教材办公室领导具体指导和帮助;也得到南京航空航天大学教材科同志的鼎力帮助;奚杭生、陈鸿茂、徐伯良同志审阅了大部分和部分初稿,提出了许多宝贵意见,在此表示衷心感谢。

由于编著者水平有限，加上时间仓促，漏误之处难免，恭请同行专家和读者批评指正。

编著者

1993. 5.

# 目 录

|                          |        |
|--------------------------|--------|
| 绪论                       | ( 1 )  |
| 第一章 计算机基础                | ( 5 )  |
| § 1.1 计算机中数的表示方法及运算      | ( 5 )  |
| 1.1.1 机器数和真值             | ( 5 )  |
| 1.1.2 负数的表示方法——原码、反码和补码  | ( 6 )  |
| 1.1.3 补码的运算              | ( 9 )  |
| 1.1.4 计算机中数的表示范围         | ( 11 ) |
| 1.1.5 定点数与浮点数            | ( 11 ) |
| § 1.2 计算机中常用的码制          | ( 13 ) |
| 1.2.1 BCD 码及其十进制调整       | ( 13 ) |
| 1.2.2 ASCII 码及 EBCDIC 码  | ( 15 ) |
| § 1.3 微型计算机基本工作原理        | ( 16 ) |
| 1.3.1 微型计算机硬件组成          | ( 16 ) |
| 1.3.2 微型计算机软件系统          | ( 23 ) |
| 1.3.3 微型计算机中指令执行的基本过程    | ( 24 ) |
| § 1.4 评估计算机主要技术指标        | ( 28 ) |
| 1.4.1 字长                 | ( 28 ) |
| 1.4.2 内存储器容量             | ( 28 ) |
| 1.4.3 指令执行时间             | ( 29 ) |
| 1.4.4 外部设备配备多少           | ( 29 ) |
| 1.4.5 软件配备情况             | ( 29 ) |
| 思考题与习题                   | ( 29 ) |
| 第二章 80X86 微处理器           | ( 31 ) |
| § 2.1 8086/8088 CPU 内部结构 | ( 31 ) |
| 2.1.1 CPU 结构与特点          | ( 31 ) |
| 2.1.2 寄存器配置              | ( 35 ) |
| § 2.2 8086/8088 引脚及功能    | ( 38 ) |
| 2.2.1 8088 CPU 引脚功能      | ( 39 ) |
| 2.2.2 8086 CPU 引脚功能      | ( 43 ) |
| § 2.3 80286 微处理器         | ( 44 ) |

|                                  |        |
|----------------------------------|--------|
| 2.3.1 从 8086 到 80286 .....       | ( 44 ) |
| 2.3.2 80286 引脚及编程结构 .....        | ( 46 ) |
| 2.3.3 80286 保护(虚地址)模式 .....      | ( 49 ) |
| § 2.4 80386 微处理器 .....           | ( 52 ) |
| 2.4.1 80386 寄存器组织 .....          | ( 53 ) |
| 2.4.2 80386 对存储器的管理 .....        | ( 54 ) |
| 2.4.3 80386 引脚及编程结构 .....        | ( 58 ) |
| § 2.5 80486 微处理器 .....           | ( 59 ) |
| 2.5.1 从内部结构看提高性能两大措施 .....       | ( 59 ) |
| 2.5.2 提高速度的其它措施 .....            | ( 60 ) |
| § 2.6 8086/8088 系统组织 .....       | ( 60 ) |
| 2.6.1 8086/8088 支持芯片 .....       | ( 60 ) |
| 2.6.2 单 CPU 模式(最小模式)系统 .....     | ( 64 ) |
| 2.6.3 多 CPU 模式(最大模式)系统 .....     | ( 66 ) |
| 思考题与习题 .....                     | ( 69 ) |
| 第三章 80X86 指令系统 .....             | ( 71 ) |
| § 3.1 8086/8088 指令系统 .....       | ( 71 ) |
| 3.1.1 8086/8088 指令格式 .....       | ( 71 ) |
| 3.1.2 8086/8088 指令系统的寻址方式 .....  | ( 73 ) |
| 3.1.3 8086/8088 指令系统 .....       | ( 78 ) |
| § 3.2 80X86 指令系统 .....           | (110)  |
| 3.2.1 80286 增强与增加的指令 .....       | (110)  |
| 3.2.2 80386/80486 增强与增加的指令 ..... | (112)  |
| 思考题与习题 .....                     | (115)  |
| 第四章 汇编语言程序设计 .....               | (118)  |
| § 4.1 汇编语言的语句 .....              | (119)  |
| 4.1.1 语句的种类和格式 .....             | (120)  |
| 4.1.2 伪指令语句 .....                | (123)  |
| 4.1.3 条件汇编语句 .....               | (134)  |
| 4.1.4 宏指令语句 .....                | (134)  |
| § 4.2 汇编语言程序结构 .....             | (137)  |
| 4.2.1 汇编语言源程序的结构 .....           | (137)  |
| 4.2.2 汇编语言和操作系统 PC-DOS 的接口 ..... | (138)  |
| § 4.3 汇编语言程序设计的基本方法 .....        | (139)  |
| 4.3.1 概述 .....                   | (139)  |
| 4.3.2 程序的基本结构形式 .....            | (141)  |

|                                                  |       |
|--------------------------------------------------|-------|
| 4.3.3 分支程序设计 .....                               | (142) |
| 4.3.4 循环程序设计 .....                               | (147) |
| 4.3.5 子程序设计 .....                                | (151) |
| 4.3.6 IBM PC-DOS 系统功能调用 .....                    | (157) |
| § 4.4 程序设计举例 .....                               | (162) |
| 4.4.1 十进制算术运算 .....                              | (162) |
| 4.4.2 代码转换 .....                                 | (168) |
| 4.4.3 表处理和应用 .....                               | (171) |
| 思考题与习题 .....                                     | (178) |
| 第五章 80X86 CPU 时序 .....                           | (181) |
| § 5.1 概述 .....                                   | (181) |
| § 5.2 基本 BIU 总线周期 .....                          | (181) |
| § 5.3 8086/8088 在单 CPU 系统时读存储器和读 I/O 端口的时序 ..... | (182) |
| 5.3.1 8086 存储器读时序 .....                          | (182) |
| 5.3.2 8086 存储器写时序 .....                          | (183) |
| 5.3.3 8088 访问存储器时序 .....                         | (184) |
| § 5.4 8086/8088 多 CPU 系统读/写存储器和 I/O 端口时序 .....   | (185) |
| 5.4.1 8086 多 CPU 系统读存储器或 I/O 端口时序 .....          | (185) |
| 5.4.2 8086 多 CPU 系统写存储器或 I/O 端口时序 .....          | (185) |
| 5.4.3 8088 多 CPU 系统读/写总线周期时序 .....               | (186) |
| § 5.5 8086/8088 等待(WAIT)状态 .....                 | (186) |
| § 5.6 总线空闲周期 .....                               | (187) |
| § 5.7 80X86 时序介绍 .....                           | (188) |
| 思考题与习题 .....                                     | (189) |
| 第六章 半导体存储器 .....                                 | (190) |
| § 6.1 概述 .....                                   | (190) |
| 6.1.1 半导体存储器的性能指标 .....                          | (190) |
| 6.1.2 分类 .....                                   | (190) |
| § 6.2 随机存取存储器 RAM .....                          | (191) |
| 6.2.1 静态 MOS RAM .....                           | (191) |
| 6.2.2 动态 MOS RAM .....                           | (194) |
| 6.2.3 IRAM .....                                 | (197) |
| § 6.3 只读存储器 ROM .....                            | (197) |
| 6.3.1 掩模 ROM、PROM 介绍 .....                       | (197) |
| 6.3.2 可擦除的 PROM .....                            | (198) |
| § 6.4 存储器连接与扩充 .....                             | (201) |



|                               |       |
|-------------------------------|-------|
| 6.4.1 存储器芯片选择 .....           | (201) |
| 6.4.2 存储器容量扩充 .....           | (203) |
| § 6.5 8086/8088 与存储器连接 .....  | (205) |
| 思考题与习题 .....                  | (209) |
| 第七章 输入/输出和中断 .....            | (211) |
| § 7.1 概述 .....                | (211) |
| § 7.2 输入/输出的传送方式 .....        | (212) |
| 7.2.1 程序控制的输入和输出 .....        | (213) |
| 7.2.2 中断控制的输入和输出 .....        | (216) |
| 7.2.3 直接数据通道传送(DMA) .....     | (216) |
| § 7.3 中断技术 .....              | (218) |
| 7.3.1 中断的概念及中断处理过程 .....      | (218) |
| 7.3.2 中断优先权 .....             | (221) |
| § 7.4 80 X 86 中断系统 .....      | (224) |
| 7.4.1 80 X 86 中断结构 .....      | (224) |
| 7.4.2 内部中断——软件中断 .....        | (225) |
| 7.4.3 外部中断——硬件中断 .....        | (227) |
| 7.4.4 各类中断的优先权 .....          | (230) |
| § 7.5 8259 A 可编程中断控制器 .....   | (230) |
| 7.5.1 8259A 的功能、结构和工作原理 ..... | (230) |
| 7.5.2 8259A 的编程 .....         | (234) |
| 思考题与习题 .....                  | (248) |
| 第八章 接口技术 .....                | (250) |
| § 8.1 概述 .....                | (250) |
| § 8.2 计数器和定时器电路 .....         | (251) |
| 8.2.1 概述 .....                | (251) |
| 8.2.2 8253 可编程定时器/计数器 .....   | (252) |
| § 8.3 并行接口芯片 .....            | (263) |
| 8.3.1 并行通讯概述 .....            | (263) |
| 8.3.2 8255A 可编程并行接口 .....     | (264) |
| 8.3.3 键盘和显示器接口 .....          | (275) |
| § 8.4 串行通讯和串行接口芯片 .....       | (281) |
| 8.4.1 串行通讯的基本概念 .....         | (282) |
| 8.4.2 可编程异步通讯接口 8250 .....    | (289) |
| § 8.5 模拟通道接口 .....            | (300) |
| 8.5.1 概述 .....                | (300) |

|                              |       |
|------------------------------|-------|
| 8.5.2 D/A 转换器 .....          | (301) |
| 8.5.3 A/D 转换器 .....          | (312) |
| 思考题与习题 .....                 | (327) |
| 第九章 微型计算机系统 .....            | (330) |
| § 9.1 80X86 微机的硬件系统 .....    | (330) |
| 9.1.1 IBM PC/XT 微机硬件系统 ..... | (331) |
| 9.1.2 PC/AT 微机的硬件特点 .....    | (336) |
| 9.1.3 386/486 微机的硬件特点 .....  | (337) |
| 9.1.4 EISA 总线 .....          | (338) |
| § 9.2 PC DOS 操作系统 .....      | (341) |
| 9.2.1 PC DOS 结构及系统启动 .....   | (341) |
| 9.2.2 文件系统 .....             | (346) |
| 9.2.3 DOS 的内存分配 .....        | (348) |
| 9.2.4 DOS 的盘区分配 .....        | (349) |
| § 9.3 微机应用系统的开发 .....        | (350) |
| 9.3.1 四种方法比较 .....           | (351) |
| 9.3.2 BIOS 功能调用举例 .....      | (352) |
| 9.3.3 用四种方式实现异步通信 .....      | (354) |
| 9.3.4 用中断方法实现数据的串行传送 .....   | (360) |
| 9.3.5 调试程序 DEBUG .....       | (364) |
| 9.3.6 DEBUG 命令 .....         | (366) |
| 9.3.7 完整程序在微机上的执行 .....      | (374) |
| 思考题与习题 .....                 | (377) |
| 附录 .....                     | (379) |
| 一 8086/8088 指令系统 .....       | (379) |
| 二 BIOS 软件中断 .....            | (395) |
| 三 PC DOS 系统功能 .....          | (402) |
| 参考书目 .....                   | (408) |

## 绪 论

电子数字计算机的出现,是近代重大科学成就之一。自从 1946 年第一台电子计算机问世至今近 50 年历史中,它的发展已经历了四代,目前正向第五代过渡。现在,在个人计算机(Personal Computer)上操作的人们,很难想象到第一台电子计算机 ENIAC (electronic numerical integrator and computer)是怎样的一个庞然大物:它体积有 85 立方米,占地面积 150 平方米,重 30 吨,有 1800 多只电子管,耗电 140 千瓦,价值 40 多万美元,而其性能上,与现代一些通用微型机相比相差更远,它只能存储 750 条指令,每秒钟只能进行 360 次乘法运算,平均每运行 7 分钟就可能产生一次故障;而本书即将介绍的 16 位机可储存几百万条指令,每秒钟进行乘法运算可高达几百万次,但价格仅有 ENIAC 的几分之一,更可贵的是,平均无故障时间以年来计算,基于上述原因,它已从科学计算、企业管理、办公室自动化深入到工业控制和仪器仪表智能化领域中了。

近 50 年来,电子数字计算机的发展已经历了四代,现正向第五代过渡。

1946~1958 年为第一代,主要特征是计算机的逻辑元件均采用电子管故称电子管数字计算机。主存储器采用磁芯、磁鼓,外存储器已开始采用磁带。运算速度为每秒几千次到几万次。主要用于科学计算。编写程序采用机器语言,后期发展为汇编语言。

1958~1964 年为第二代,主要是用晶体管代替了电子管。主存储器仍为磁芯,外存储器开始使用磁盘。计算机软件有了较大发展,高级语言(FORTRAN、COBOL、ALGOL)和编译程序已普遍采用。计算机运算速度已提高到每秒几十万次。不仅用于科学计算和数据处理,而且还用于工业控制。

1964~1975 年为第三代。计算机逻辑元件采用中小规模集成电路构成。计算机运算速度进一步提高,成本则相继降低。在软件方面,操作系统日趋完善和功能进一步强化,极大提高了计算机工作效率。这一期间出现了小型机系列,人们开始考虑大、中和小型机配套使用,出现了计算机网络和数据库,微程序技术也得到了广泛的应用。

一般人们认为第四代计算机的起点是在 70 年代中期,一直沿伸至 80 年代。这时的计算机由大规模集成电路构成。大规模集成电路是在一块几平方毫米的半导体芯片上,集成成千上万乃至百万个逻辑门,因而计算机体积更小,耗电更少,运算速度提高到每秒数百万次,其可靠性也大大提高。第四代计算机有四个发展方向,人们常称为四个“Super”。一是速度高,处理能力极强的巨型计算机,即 Super computer。巨型机的运算速度可达每秒几亿次,甚至达几十亿次。第二个 Super 是小巨型机,即 Mini super computer。它是并行计算技术和多微处理器系统相结合的产物。90 年代以后的小巨型机可能包含成千上万个微处理器,其性能接近巨型机,但价格要低得多。第三个 Super 是超小型机,即 Super mini computer。目前的超小型机是在 70 年代后期的 16 位小型机基础上发展起来的 32 位小型机,其性能大大超过了传统的小型机,同时又保持了小型机的价格便宜等特点。最后一种 Super 是采用超大规模集成技术(VLSI)的超级微型机,即 Super micro computer。它们都采用超大规模集成的 32

位微处理器芯片,性能价格比都很高。继第四代计算机之后,自 80 年代中期,又开始了新一代计算机系统的研究,即所谓第五代计算机的研究。它是基于对人大脑神经网络功能的模拟研究,开发神经元计算(neuro computing)系统,以便直接仿效人脑对信息的加工处理。

特别要强调说明的是,70 年代初期,由于微电子技术的发展,产生了超大规模集成电路(VLSI),从而导致以微处理器为核心微型计算机的诞生。微型计算机(Micro computer)与其它计算机的区别在于它的中央处理器 CPU(Central Processing Unit)是采用大规模(或超大规模)技术集成在一块硅片上;而其它的大、中和小型机的 CPU 是由相当多的集成电路组成。为了和其它计算机的 CPU 区别,称微型计算机的 CPU 芯片为微处理器(Microprocessor),或简称 MPU(Microprocessing Unit)。微型计算机的发展是以微处理器的发展为表征的。微处理器自 1971 年诞生以来,在短短十几年里发展成了四代产品,几乎是每 2~3 年就要更换一代。

1971~1973 年为第一代,主要产品是 INTEL 4004~8008。基本特点是采用 PMOS 工艺,集成度低(1200~2000 晶体管/片),系统结构与指令都较简单,且速度慢(基本指令执行时间为 10~20 $\mu$ s)。主要应用于家用电器和简单控制场合。

1974~1977 年为第二代,主要产品 INTEL 8080,Zilog-80,Motorola 的 MC 6800。这一代显著特点是采用了 NMOS 工艺,集成度提高约 4 倍(5000~9000 晶体管/片),速度提高了 10~15 倍(基本指令执行时间为 1~2 $\mu$ s)。系统结构与指令都比较完善。

1978~1981 年为第三代,主要产品 INTEL 8086/8088,Z 8000 和 MC 68000,已从典型 8 位上升至 16 位或准 16 位。这一代主要特点是采用 HMOS 工艺,其集成度(达 20000~70000 晶体管/片)和速度(基本指令执行时间约 0.5 $\mu$ s)都比第二代提高了一个数量级。其体系结构与指令都更为完善与丰富,采用了多级中断,多重寻址,段式寄存器等结构。

1982 年以后为第四代,它主要是 32 位微处理器。主要产品是 INTEL 80386/80486 和 MC 68030/68040。这一代微处理器大多采用了 HMOS 或 CMOS 工艺,其集成度每片芯片高达 100 万只晶体管,基本指令执行时间一般在 25mips(每秒百万条指令)。

当前 32 位微处理器大致可分为两种类型。一种是现有 16 位微处理器的扩充,即外部总线为 16 位,而内部寄存器、ALU 总线宽度等可扩充为 32 位,实际上是准 32 位。NS 16032 等即属此类。另一类是外部数据总线也为 32 位的微处理器,如 80386,这才是真正的 32 位微处理器。微处理器的出现,经过十多年市场激烈竞争,目前保留了两大系列即 INTEL 8086/80386/80486/80586/80686/80786 简称 80X86 直到 2000 年 Micro 2000 和 Motorola 的 68000/68020/68040/68050 简称 680X0。

以 80X86 CPU 为例,今后的 PC 系列将向更微型化、速度更快、价廉、多图形和功能更完善方向发展。人们预计,今天见到的 486 微机,3~5 年后只能发挥余热了。那时的“PC 机”,除保持现有 486 全部功能外,还会把电话、收音机、TV、FAX/Modem、多媒体集成进去。与此相应的一些技术,如 RISC、总线(除已有的 EISA、VESA 和 PCI 外)、接口技术(串行口采用远红外通信外)、打印机及显示技术将会按人们需求更加先进和完善。

“微电脑”或“微型计算机”这个术语,已为人们所熟悉,实际上这几个术语是好几个概念的统称。为了下面不致混淆,特作如下解释:

微处理器(Microprocessor)也称微处理机,它本身不是计算机,但它是微型计算机的核

心部件(或中央处理器 CPU)。微处理器包括三个基本部分,即算术逻辑部件 ALU(Arithmetical Unit),寄存器组(Registers)和控制部件(Control Unit)。一个微处理器可以由一片或几片 LSI, VLSI 器件组成。

微型计算机(Micro computer,简称 MC)就是由微处理器、存储器、输入、输出接口(I/O 口)和其它支持逻辑所组成。它们彼此通过系统总线(地址总线、数据总线和控制总线)连接起来,以完成某些特定运算与实时控制。

微型计算机系统(Micro Computer System,简称 MCS),是在上述微型计算机的基础上,配置系统软件与应用软件、外部设备、系统总线接口和电源等,构成一个完整的计算机系统。

单片微型计算机(Single Chip Micro Computer 简称单片机),它是把 CPU、内存储器和某些 I/O 接口集成在一块超大规模芯片上的微型计算机。它主要用于工业控制和智能仪器仪表,故又称单片微控制器。

单板微型计算机(Single board micro computer,简称单板机),它是把 CPU、存储器和 I/O 接口电路等大规模或超大规模集成电路芯片及必要的外围支持电路芯片组装在一块印刷电路板上,并辅以简单键盘和 LED 显示器构成的微型计算机。由于简单和价廉,80 年代我国把它主要用于教学实验和工业控制。

今天计算机及微型计算机应用的发展速度、深度及其广度,都远远超过了历史上任何一种技术手段和装备,不论是国防、科学研究、政治经济,以及教育文化乃至家庭生活、文化娱乐等方面无不涉及。所用之处,不仅引起该领域巨大变革,而且应用中碰到的问题,反过来又推动计算机技术向前发展。总的来说,国外计算机应用大致可分为三个阶段。科学计算、工业控制和数据处理以及管理和事务处理等方面。我国的计算机应用情况也大致与国外相仿。目前总体应用水平,接近国外 70 年代到 80 年代初的水平。归纳起来有以下几个方面:

### 一、企业管理

现代企业管理包括的内容很多,如财务项目、信贷业务管理、购销、编制生产计划、销售预测、劳动力及人事档案、成本分析与核算等方面。国外在这方面的工作,要占微机应用的 80%,我国则起步较晚,正在发展之中。

### 二、科学计算和数据处理

在工程设计、科学研究和国防事务中,遇到大量的数学计算问题,采用计算机可得到快速而较理想的解答。

用计算机进行数据处理、图像分析等已在医学诊断、材料探伤、地质勘探、地震预测、气象研究等方面得到广泛应用。

### 三、工业控制

在工农业生产、国防、交通等方面,利用计算机进行监视、测量和控制,如对生产过程的压力、流量、温度等进行监测控制,机床控制,铁路交通的信息管理控制,矿井中环境条件检测控制以确保人身安全与生产效率的提高。

#### **四、通讯技术**

计算机与通讯技术结合产生了计算机网络,计算机网络是由多台计算机组成,各计算机之间采用通信线路互连的网络。另外,计算机在程控电话交换中应用,它也是通信技术与计算技术结合的产物。

#### **五、仪器、仪表**

由于在仪器、仪表中应用微处理器或微型计算机,使仪器的重量、体积、质量和功能等方面都有很大改进,特别是单片机的发展与应用,可使仪表体积更小。

#### **六、计算机辅助设计**

为了提高产品设计质量、缩短设计周期以及提高自动化水平,借助计算机帮助进行设计称为计算机辅助设计CAD(Computer Aided Design)。当今计算机辅助设计已扩展到测试和制造等过程中,实现了计算机辅助测试(CAT)和计算机辅助制造(CAM)。有的把设计制造融为一体而成为计算机集成制造(CIMS)。

综上所述,由于计算机和微型计算机的应用,对社会进步产生巨大的影响。有人把微型计算机的出现称之为第二次工业革命,近来,日本理光公司人工神经计算机的出现,必将推动社会进步到一个更高的层次。

为了早日把我国经济搞上去,必须学习好微型计算机技术,开发更多微机应用领域,尽早赶上先进发达的国家。

# 第一章 计算机基础

## § 1.1 计算机中数的表示方法及运算

电子数字计算机是一个自动化信息处理机,其中心任务就是对输入的数据进行高速运算处理,计算机中的数据、指令和参加运算的数都是二进制数,即用“0”和“1”两个码表示。十六进制数,只是为了将二进制数表示得更简便时才用到。本节讨论数的表示,特别是负数在机器中的表示和运算。

### 1.1.1 机器数和真值

在计算机中,只能表示0和1两种数码,为了表示正数和负数,专门选择一位符号位来表示数的符号,通常选择最高位作为符号位,当该位为“0”时,表示正号,为“1”时,表示负号,即最高位为符号位:符号位为0,表示正数的符号“+”;符号位为1,表示负数的符号“-”。

这就是说,数的符号在计算机中也数码化了。我们把一个数在机器(计算机)中的表示形式称为机器数,而把原来实际数本身值叫真值(机器数真值)。真值可以用二进制数表示,也可用十进制数表示,但根据习惯,常用十进制数表示,如图1-1所示。

机器数有如下特点:

1. 前已述及,机器数的正负号已经数值化了,这是它的第一个特点。

2. 机器数所能表示数的范围受到机器(计算机)字长的限制。那么什么是计算机字长呢?我们先来解释几个计算机中常用术语。

(1)位(Bit):位是计算机所能表示的最小数据单位,它只能有二种状态“0”和“1”。

(2)字(Word):计算机中,一次可以处理或运算的一组二进制数,是计算中信息的基本单元。

(3)字长(Word Length):指字的二进制数的位数。8位微处理器的字长为8位,32位微处理器的字长为32位等。

(4)字节(Byte):一个8位二进制数称为一个字节。在8位微处理器中,每个字由一个字节构成;而在16位微处理器中,每个字由2个字节构成,如图1-2所示。

3. 小数点不能直接标出,需要按一定方式约定小数点的位置。

| 机器数   |                                                                                                         | 真值                           |
|-------|---------------------------------------------------------------------------------------------------------|------------------------------|
| $N_1$ | <div><div>D<sub>7</sub></div><div>0 1 1 0 1 0 1 1</div><div>D<sub>0</sub></div><div>符号位 数值位</div></div> | $N_1 = +1101011$<br>$= +107$ |
| $N_2$ | <div><div>1 1 1 0 1 0 1 1</div><div>符号位 数值位</div></div>                                                 | $N_2 = -1101011$<br>$= -107$ |

图 1-1 机器数真值

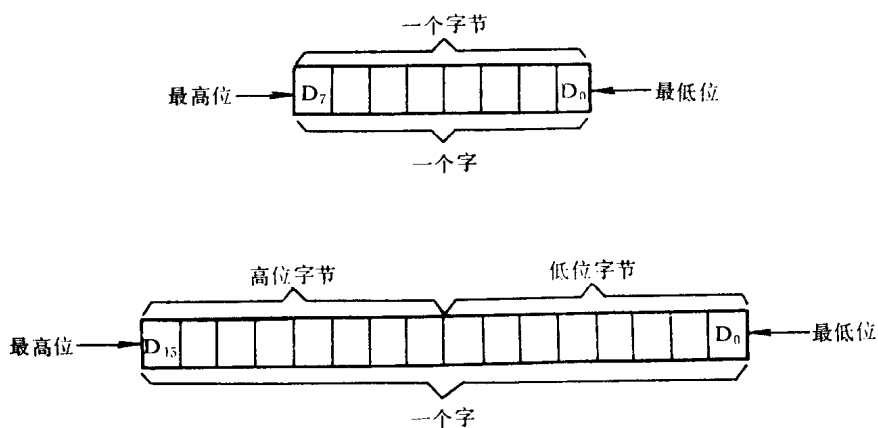


图 1-2 字和字节

### 1.1.2 负数的表示方法——原码、反码和补码

既然一个数的数值和符号全都是数码,那么当对这种机器数进行运算时,符号位怎样处理?能不能也同数值位一道参加运算呢?为妥善解决这些问题,引出了机器数  $x$  的三种不同的编码形式,即原码、反码和补码。分别记作:  $[x]_{\text{原}}$ 、 $[x]_{\text{反}}$  和  $[x]_{\text{补}}$ 。

#### 一、原码

设  $x = x_1, x_2, \dots, x_{n-1}$ , 其中  $x_i$  为一位二进制数,  $i = 1, 2, \dots, (n-1)$ 。

$$[x]_{\text{原}} = \begin{cases} 0 & x_1 & x_2 & \dots & x_{n-1} & \text{当 } x \geq 0 \\ 1 & x_1 & x_2 & \dots & x_{n-1} & \text{当 } x \leq 0 \end{cases}$$

即一个数的原码,就是数值部分不变,仅仅用 0 和 1 分别来表示数的符号“+”和“-”的机器数。

例  $x_1 = 67 = +1000011$   $[x_1]_{\text{原}} = 01000011$   
 $x_2 = -67 = -1000011$   $[x_2]_{\text{原}} = 11000011$  在原码表示法中,根据定义,数 0 的原码有两种不同形式(设字长为 8 位):

$$[+0]_{\text{原}} = 00000000$$

$$[-0]_{\text{原}} = 10000000$$

原码表示简单易懂,而且与真值的转换方便。但原码表示的数不便于计算机运算,因为在两原码数运算时,首先要判断它们的符号,然后再决定用加法还是用减法。致使机器的结构相应地复杂化或增加机器的运算时间。为解决上述弊病,引入反码和补码表示法。

#### 二、反码

设  $x = x_1, x_2, \dots, x_{n-1}$ , 其中  $x_i$  为一位二进制数,  $i = 1, \dots, (n-1)$ 。

则



$$[x]_{\text{反}} = \begin{cases} 0 & x_1 & x_2 & \cdots & x_{n-1} & \text{当 } x \geq 0 \\ 1 & \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_{n-1} & \text{当 } x \leq 0 \end{cases}$$

$$\text{式中 } \bar{x}_i = \begin{cases} 0 & \text{当 } x_i = 1 \\ 1 & \text{当 } x_i = 0 \end{cases}$$

即对于正数,反码与原码相同,符号位为 0,其余位为数值位本身;对于负数,反码的符号位为 1,其余位为数值位按位取反(即将 1 变为 0,0 变为 1)。

$$\text{例 } x_1 = 83 = +1010011, \quad [x_1]_{\text{反}} = 01010011$$

$$x_2 = -83 = -1010011, \quad [x_2]_{\text{反}} = 10101100$$

在反码表示法中,根据定义,数 0 的反码有两种不同形式(设字长为 8 位):

$$[+0]_{\text{反}} = 00000000$$

$$[-0]_{\text{反}} = 11111111$$

### 三、补码

上面已经谈到,引入补码的概念,目的在于将加、减运算简化为单纯的相加运算。

#### 1. 同余的概念和补码

设有两个数  $a=17, b=27$ ,若用 10 去除  $a$  和  $b$ ,则它们的余数均为 7,我们称 17 和 27 在以 10 为模时是同余的,并记作:

$$17 = 27 \pmod{10}$$

或者说 17 和 27 在以 10 为模时是相等的。此处的模就是一个计量系统所能表示的最大量程(或一个计量单位称为模或模数)。

由同余概念,不难得出:

$$a + M = a \pmod{M}$$

$$a + 2M = a \pmod{M}$$

因此当  $a$  为负数时,如  $a = -4$ ,在以 10 为模时,有:

$$-4 + 10 = -4 \pmod{10}$$

$$\text{或 } 6 = -4 \pmod{10}$$

上式说明,在以 10 为模时,  $-4$  与  $+6$  是相等的。我们称  $+6$  为  $-4$  的补码,或者说  $+6$  与  $-4$  对模 10 来说互为补数。有了补码的概念,就可将减法转化为加法(加补码)来进行。如:

$$7 - 4 = 7 + 6 \pmod{10}$$

该式说明,在以 10 为模时,7 减 4 可以通过 7 加  $-4$  的补码 6 来进行,而所得的结果是相同的(只要在加补码 6 时,将所产生的进位舍弃即可,这正是以 10 为模的意思)。

现在把同余和补码的概念推广到二进制数,设计算机字长为  $n$  位,那么其模  $2^n$ ,即:

$$\underbrace{\overbrace{100 \cdots 0}^{n \text{ 个 } 0}}_{n+1 \text{ 位}}$$

这就是说,字长  $n$  位的机器,  $2^n$  在机器中仅能以  $n$  个 0 表示。或者说,  $2^n$  和 0 在机器中