

WEIYUYUANLI

# 微型计算机原理

主编 / 付燕宁 胡利民 刘 岩

● 吉林科学技术出版社

# 微型计算机原理

主 编 付燕宁 胡利民 刘 岩  
副主编 宗 林 郑 平 闫永信

吉林科学技术出版社

**【吉】新登字 03 号**

微型计算机原理

付燕宁 胡利民 刘 岩 主编

责任编辑：王维义

封面设计：杨峻嵩

出版 吉林科学技术出版社  
发行

787×1092 毫米 16 开本  
印刷 吉林省实验中学印刷厂

17 印张 396,000 字  
印数：1—2000 册

1995 年 12 月第 1 版  
1995 年 12 月第 1 次印刷

ISBN 7—5384—1427—4/TP·32

定价：18.00

# 前 言

微型计算机一经产生，显示出无比强大的生命力，发展十分迅猛。平均每隔 3~5 年，微处理器及其微型计算机就更新换代，目前由 Intel 公司推出的 Pentium，是 Intel 系列微处理器芯片中性能最高的产品。

Intel 微处理器芯片是当今世界 CPU 的主流芯片，本书以 Intel8086 及其所组成的微型计算机为对象，阐述其结构及原理，其原因如下：

①Intel8086 与其前期的微处理器相比，内部结构做了重大改进；并为以后微处理器升档奠定了基础。

②以 Intel8086 所组成的微型计算机系统，它的许多设计思想，芯片的连接，信号关系等都成为以后的微型计算机设计的主要参考因素。

③后期研制的微处理器在设计上都保持与 Intel8086 的兼容性。8086、80286、80386、80486、Pentium (80586) 是向上兼容的，8086 以后的微处理器包含了 8086 的全部指令系统和全部功能，由它们所组成的微型计算机系统能够运行 8086 微型计算机系统上的所有软件。

因此，学习微型计算机必须从 8086 开始，分析微型计算机也必须从 8086 着手。

本书以微型计算机硬件结构为出发点，首先给出了必要的基础知识，从假想模型机入手，说明微型计算机的基本概念和基本结构，并通过实例说明其基本工作原理，以期读者获得对微型计算机的整体认识。随后，主要以 8086/8088 微型计算机为对象，展开对组成微型计算机各个部件以及它们之间关系的讨论。最后，为了跟踪微处理器的发展动态，简要地介绍了 Intel 系列微处理器。

本书是编者在总结了微型计算机原理教学经验基础上，综合了大量资料编写而成。深入浅出，简明易懂，具有较好的系统性和先进性。适合于计算机应用专业微型计算机原理用教材，亦可作为从事计算机软、硬件开发工作的工程技术人员参考。

本书由付燕宁、胡利民、刘岩主编，宗林、郑平、闫永信副主编，杜莉萍、韩学文、王晓昕、万殿一也参加了本书的编写工作，最后由付燕宁总纂。中国农业银行长春管理干部学院辛国平教授主持了本书的审阅工作，赵维平、夏沫两位老师参加了本书的审阅，并提出许多宝贵意见，在此谨表谢意。

由于编者水平有限，书中难免会出现不当或错误之处，恳请广大读者批评指正。

编者

1995 年 12 月

# 目 录

## 第一章 计算机基础

|                        |      |
|------------------------|------|
| 第一节 微型计算机发展概况及特点 ..... | (1)  |
| 一、微型计算机发展概况 .....      | (1)  |
| 二、微型计算机特点 .....        | (2)  |
| 第二节 计算机中数的表示方法 .....   | (3)  |
| 一、数制 .....             | (3)  |
| 二、数制转换 .....           | (5)  |
| 三、码制 .....             | (6)  |
| 四、定点和浮点 .....          | (10) |
| 第三节 计算机中常用编码 .....     | (12) |
| 一、十进制数的编码——BCD 码 ..... | (12) |
| 二、字符的编码——ASCII 码 ..... | (13) |

## 第二章 计算机基本结构及工作过程

|                     |      |
|---------------------|------|
| 第一节 计算机的基本组成 .....  | (16) |
| 一、计算机组成 .....       | (16) |
| 二、指令及指令系统 .....     | (17) |
| 第二节 模型计算机结构 .....   | (18) |
| 一、中央处理器 .....       | (18) |
| 二、存储器 .....         | (21) |
| 三、外部总线 .....        | (22) |
| 四、计算机结构特点 .....     | (23) |
| 第三节 计算机基本工作过程 ..... | (23) |
| 一、程序的编制及输入 .....    | (23) |
| 二、程序的执行过程 .....     | (24) |

## 第三章 Intel 8086 微处理器

|                              |      |
|------------------------------|------|
| 第一节 8086 CPU 内部寄存器结构 .....   | (29) |
| 第二节 8086 CPU 内部结构和工作原理 ..... | (32) |
| 一、8086 CPU 的内部结构 .....       | (32) |
| 二、8086 CPU 工作原理 .....        | (33) |
| 第三节 8086 CPU 的存储器组织 .....    | (33) |
| 一、8086 的存储空间 .....           | (33) |
| 二、20 位存储器物理地址的形成 .....       | (34) |
| 第四节 8086 CPU 外部引脚及功能 .....   | (36) |
| 一、地址和数据引脚 .....              | (37) |
| 二、控制和状态引脚 .....              | (38) |
| 三、电源和定时引脚 .....              | (41) |

|                              |      |
|------------------------------|------|
| 第五节 8086 CPU 外部系统总线 .....    | (41) |
| 一、微型计算机的系统总线 .....           | (41) |
| 二、8086 的总线周期和系统总线 .....      | (42) |
| <b>第四章 支持 8086 CPU 的外围芯片</b> |      |
| 第一节 8284 时钟发生器驱动器 .....      | (48) |
| 一、时钟信号 CLK .....             | (48) |
| 二、复位信号 RESET .....           | (49) |
| 三、准备就绪信号 READY .....         | (50) |
| 第二节 等待状态发生器 .....            | (52) |
| 第三节 8288 总线控制器 .....         | (53) |
| 一、状态译码和控制逻辑 .....            | (54) |
| 二、命令输出 .....                 | (55) |
| 三、控制输出 .....                 | (55) |
| 第四节 8086 CPU 的两种工作方式 .....   | (56) |
| 一、8086 的最小方式 .....           | (57) |
| 二、8086 的最大方式 .....           | (59) |
| <b>第五章 指令编码及寻址方式</b>         |      |
| 第一节 指令及其表示方法 .....           | (61) |
| 一、指令和指令系统 .....              | (61) |
| 二、指令的分类 .....                | (61) |
| 三、指令的表示方法 .....              | (61) |
| 第二节 8086 CPU 的寻址方式 .....     | (63) |
| 一、立即寻址 .....                 | (63) |
| 二、直接寻址 .....                 | (64) |
| 三、寄存器寻址 .....                | (64) |
| 四、寄存器间接寻址 .....              | (64) |
| 五、变址寻址 .....                 | (64) |
| 六、基址加变址的寻址 .....             | (65) |
| 第三节 8086 机器指令编码 .....        | (66) |
| 第四节 8086 指令的执行时间 .....       | (69) |
| <b>第六章 8086 CPU 的指令系统</b>    |      |
| 第一节 数据传送指令 .....             | (71) |
| 一、通用数据传送指令 .....             | (72) |
| 二、输入/输出数据传送指令 .....          | (75) |
| 三、目的地址传送指令 .....             | (77) |
| 四、标志位传送指令 .....              | (78) |
| 第二节 算术运算指令 .....             | (79) |
| 一、加法运算指令 .....               | (80) |
| 二、减法运算指令 .....               | (83) |
| 三、乘法运算指令 .....               | (87) |

|                               |       |
|-------------------------------|-------|
| 四、除法运算指令 .....                | (88)  |
| 第三节 逻辑运算和移位指令 .....           | (90)  |
| 一、逻辑运算指令 .....                | (91)  |
| 二、移位指令 .....                  | (94)  |
| 第四节 字符串操作指令 .....             | (97)  |
| 一、重复操作前缀指令 .....              | (98)  |
| 二、字符串传送指令 .....               | (98)  |
| 三、字符串比较指令 .....               | (99)  |
| 四、字符串搜索指令 .....               | (100) |
| 五、取字符串指令 .....                | (100) |
| 六、存字符串指令 .....                | (101) |
| 第五节 控制转移指令 .....              | (101) |
| 一、子程序调用指令 .....               | (102) |
| 二、无条件转移和条件转移指令 .....          | (104) |
| 三、循环控制指令 .....                | (108) |
| 四、中断指令 .....                  | (109) |
| 第六节 微处理器控制指令 .....            | (109) |
| 一、标志位操作指令 .....               | (110) |
| 二、交权指令和等待指令 .....             | (111) |
| 三、总线封锁前缀指令 .....              | (112) |
| 四、空操作指令 .....                 | (112) |
| 五、暂停指令 .....                  | (112) |
| 第七章 8086 CPU 的总线操作和时序         |       |
| 第一节 8086 的最小方式下总线操作时序分析 ..... | (114) |
| 一、存储器的读、写周期 .....             | (114) |
| 二、I/O 读、写周期与空转周期 .....        | (116) |
| 三、中断的响应周期 .....               | (116) |
| 四、系统复位 .....                  | (117) |
| 第二节 8086 的最大方式下总线操作时序分析 ..... | (118) |
| 一、存储器的读、写周期 .....             | (118) |
| 二、I/O 读、写周期 .....             | (120) |
| 第三节 8086 指令时序分析 .....         | (120) |
| 第八章 半导体存储器                    |       |
| 第一节 半导体存储器的分类 .....           | (126) |
| 一、随机存取存储器 RAM .....           | (126) |
| 二、只读存储器 ROM .....             | (127) |
| 第二节 随机存取存储器 RAM .....         | (128) |
| 一、基本存储电路 .....                | (128) |
| 二、RAM 的结构 .....               | (130) |
| 第三节 RAM 与 CPU 的连接 .....       | (134) |

|                              |       |
|------------------------------|-------|
| 第四节    只读存储器 ROM .....       | (140) |
| 一、掩膜式固定只读存储器 ROM .....       | (140) |
| 二、可擦除可编程的只读存储器 EPROM .....   | (141) |
| 第五节    8086 的存储器 .....       | (143) |
| 一、8086 存储器的结构 .....          | (143) |
| 二、外部存储器寻址 .....              | (145) |
| 三、存储器的段 .....                | (148) |
| 四、实际地址的产生 .....              | (149) |
| 五、堆栈和堆栈操作 .....              | (152) |
| 六、系统专用和保留的存储单元 .....         | (154) |
| 第九章    接口与中断                 |       |
| 第一节    输入输出接口 .....          | (155) |
| 一、引入接口部件的原因及作用 .....         | (155) |
| 二、接口电路的基本结构 .....            | (156) |
| 三、输入/输出寻址方式 .....            | (157) |
| 四、输入/输出指令 .....              | (159) |
| 第二节    CPU 与外设间的数据传送方式 ..... | (161) |
| 一、无条件传送 .....                | (161) |
| 二、程序查询传送 .....               | (162) |
| 三、程序中断传送 .....               | (164) |
| 四、直接存储器存取传送 .....            | (164) |
| 第三节    中断及中断处理 .....         | (167) |
| 一、基本概念 .....                 | (167) |
| 二、中断一般处理过程 .....             | (168) |
| 第四节    8086 的中断系统 .....      | (171) |
| 一、8086 的中断源 .....            | (171) |
| 二、中断向量表 .....                | (171) |
| 三、外部中断 .....                 | (172) |
| 四、内部中断 .....                 | (175) |
| 五、8086 中断处理顺序 .....          | (179) |
| 六、多重中断和多重处理 .....            | (182) |
| 第十章    中断控制器与 DMA 控制器        |       |
| 第一节    可编程中断控制器——8259A ..... | (185) |
| 一、内部逻辑结构 .....               | (185) |
| 二、外部引脚 .....                 | (187) |
| 三、8259A 的编程 .....            | (188) |
| 四、8259A 的工作方式 .....          | (194) |
| 第二节    DMA 控制器——8237A .....  | (199) |
| 一、主要功能 .....                 | (199) |
| 二、8237 结构 .....              | (199) |
| 三、8237 工作周期 .....            | (201) |

|                                 |       |
|---------------------------------|-------|
| 四、8237 引脚 .....                 | (201) |
| 五、8237 工作模式 .....               | (203) |
| 六、8237 寄存器组和编程 .....            | (205) |
| <b>第十一章 串行与并行接口</b>             |       |
| 第一节 串行通信 .....                  | (213) |
| 一、串行通信概念及其优点 .....              | (213) |
| 二、串行通信的基本方式 .....               | (213) |
| 三、串行通信的基本技术 .....               | (215) |
| 四、串行通信的校验方法 .....               | (216) |
| 第二节 可编程串行通信接口——8251A .....      | (216) |
| 一、基本功能 .....                    | (216) |
| 二、内部结构 .....                    | (217) |
| 三、外部引脚 .....                    | (219) |
| 四、8251A 的编程 .....               | (221) |
| 五、8251A 应用举例 .....              | (224) |
| 第三节 可编程并行通信接口——8255A .....      | (226) |
| 一、内部结构 .....                    | (226) |
| 二、外部引脚 .....                    | (227) |
| 三、8255A 的控制字 .....              | (228) |
| 四、8255A 的工作方式 .....             | (231) |
| 五、8255A 的应用举例 .....             | (235) |
| <b>第十二章 Intel 系列微处理器简介</b>      |       |
| 第一节 Intel80286 微处理器 .....       | (238) |
| 一、80286 微处理器的性能特点 .....         | (238) |
| 二、80286 微处理器的内部结构 .....         | (238) |
| 三、80286 的指令系统及寻址方式 .....        | (241) |
| 四、80286 的外部引脚 .....             | (242) |
| 第二节 Intel80386 微处理器 .....       | (244) |
| 一、80386 的主要性能 .....             | (244) |
| 二、80386 的功能结构 .....             | (244) |
| 三、寻址方式与存储器组织 .....              | (249) |
| 四、80386 总线与 80286 总线的主要区别 ..... | (253) |
| 第三节 Intel80486 微处理器 .....       | (254) |
| 一、80486 微处理器的性能特点 .....         | (254) |
| 二、80486 微处理器的内部结构 .....         | (254) |
| 三、80486 新增加的指令 .....            | (255) |
| 四、80486 微处理器外部引脚 .....          | (256) |
| 第四节 新一代微处理器 Pentium .....       | (259) |
| 一、Pentium 的主要性能 .....           | (259) |
| 二、Pentium 的内部结构 .....           | (260) |
| 三、P5 总线与 80486 总线的主要区别 .....    | (263) |

# 第一章 计算机基础

## 第一节 微型计算机发展概况及特点

电子计算机 (Electronic computer) 是一种能够自动、高速、精确地进行信息处理的现代化电子设备。自它产生以来, 发展十分迅猛, 显示出强大的生命力, 现已深入到社会生活的各个领域。

### 一、微型计算机发展概况

现代科学技术发展的需要是计算机产生的主要原因, 电子技术和自动控制技术的迅速发展, 又为电子计算机的产生提供了物质技术条件。

20 世纪 40 年代中期, 为解决复杂、繁重的导弹设计问题, 美国宾夕法尼亚大学的 J·w·manchley 和 J·P·Eckert 研制成功世界第一台电子计算机——ENIAC (Electronic Numerical Integrator And calculator)。它的字长 12 位, 使用 18800 只电子管, 1500 个继电器, 耗电 150KW, 占地 150m<sup>2</sup>, 重达 30t, 时钟频率为 100KHZ, 每秒仅能完成 5000 次加法运算, 尽管其功耗大、体积大、速度慢, 但是它为电子计算机的发展奠定了技术基础。此后, 电子计算机发展迅速, 已经历了电子管、晶体管、中小规模集成电路、大规模集成电路四代、以超大规模集成电路和人工智能技术为主要特征的第五代计算机正在蓬勃开展。

步入 70 年代后, 计算机发展呈现出四种势态, 即微型化、巨型化、网络化、智能化。1971 年, 世界上第一台微处理器和微型计算机在美国问世, 从而开创了计算机发展的崭新时代——微型计算机时代。微型计算机技术的发展, 是与大规模集成电路技术的发展分不开的。从理论上讲, 微型计算机无论在系统结构或基本工作原理与其它各类计算机并无本质上的区别, 所不同的是它广泛地采用了集成度相当高的部件和器件。尤其它的核心部件——微处理器, 采用了大规模或超大规模集成电路芯片。微型计算机每次在技术上的突破都直接反映了 LSI 技术所取得的新进展。为此, 人们在划分微型计算机技术发展阶段时, 往往以微型计算机的集中体现者——微处理器的新进展为主要标志。微处理器作为微型计算机的集中体现者, 它的性能决定了微型计算机的性能, 从而也就决定了微型计算机“代”的划分。

自微处理器产生以来, 发展很快, 不仅从 8 位、16 位、32 位发展到 64 位, 而且从 CISC (复杂指令系统计算机) 发展到 RISC (精简指令系统计算机), 最近又出现了 CISC 和 RISC 相结合的微处理器。下面主要以当今微处理器主流芯片——Intel 芯片为例, 说明微型计算机发展情况。

第一代 (1971 年~1973) 是低档四位微处理器 Intel4004。采用 PMOS 工艺, 一个芯片上集成了 1200 个晶体管; 可进行串行十进制运算; 时钟频率为 0.5MHZ, 基本指令执行时间为 10.8μs。虽然第一代微处理器在结构、性能上还很很不成熟, 但它标志着计算机进入了

一个崭新发展阶段——微型计算机时代。

第二代(1974年~1977年)是八位微处理器。1974年Intel公司推出8位微处理器Intel 8080,随后九个月Motorola公布了它的8位微处理器MC6800。它们都采用NMOS工艺,每片集成了5000个晶体管;时钟频率为1~2.5MHz,基本指令执行时间为2 $\mu$ s。1976年Intel公司推出Intel 8085,Zilog公司推出Z-80,Motorola又推出MC6809。芯片集成度达9000管/片左右,基本指令执行时间为1~1.6 $\mu$ s。与初期产品相比,速度更快,集成度更高。

第三代(1978年~1984年)是十六位微处理器。在1978年,Intel公司推出了Intel 8086微处理器。不久,Zilog公司的Z8000和Motorola公司的MC68000相继问世,它们成为微处理器第三代代表产品。8086采用HMOS工艺,在一个芯片上集成了29000个晶体管,不仅提高了集成度,而且对内部结构也进行了重大改进。增加了寄存器组和预取指令队列,因而提高了执行速度,基本指令执行速度达0.3 $\mu$ s。总之,各项性能指标均比第二代提高一个数量级,形成了与小型计算机相竞争的局面。1982年,Intel公司推出新一代十六位微处理器80826,提供对虚拟存储器的支持和对地址空间的保护。仍采用HMOS工艺,集成度达13.4万个晶体管;时钟频率为6MHz,基本指令执行时间为0.25 $\mu$ s。

第四代(1985年~1992年)是三十二位微处理器。这种微处理器对指令采用流水线方式,实现了完全32位操作,增加页式存储器管理,加强了支持图形的数据类型及操作,扩充了工作模式,寻址空间4GB。

其代表产品有Intel公司的386DX和386SX,Motorola公司的MC68030等。1986年,Intel公司推出的386DX采用CHMOS工艺;集成度达27.5万个晶体管;时钟频率为33MHz,基本指令和执行速度为0.15 $\mu$ s。386SX介于286与386DX之间的产品,其内部结构为32位,外部总线则保持与286一致,即16位。1986年,Intel研制成功80486;同期,Motorola公司推出MC68040,与80486相抗衡。80486吸收了RISC技术的核心思想,降低了每条指令执行时间,使大多数指令能在一个机器周期内完成;还采用了Burst Bus总线技术,解决了I/O瓶颈。其集成度可达90万个晶体管。1993年,Intel公司公布了新一代微处理器Pentium,从Intel产品的系列发展上看,Pentium应叫做80586,但美国商标法不保护用数字串命名的产品名,出于专利考虑,给它取名为Pentium,简称P5。

“奔腾”不仅在名称上和Intel以往产品不同,而且在技术上也有重大突破。奔腾在结构上的特点主要有:①吸收RISC长处,使用两条并行整数流水线可同时执行两条指令;②指令和数据高速暂存分开、使用64位数据总线;③在指令系统中追加了分支预测,缩短因程序分支而造成的等待;④使浮点小数运算高度流水线化。

奔腾是目前Intel系列中性能最高的产品,时钟频率分为60/66MHz两种,集成度为310万个晶体管,在一个时钟周期可以执行两条指令,运算速度超过100MIPS;即1亿次/秒。

Intel现已研制成功奔腾下一代产品P6,预计不久即可投入批量生产。

## 二、微型计算机特点

集成电路技术的发展推动了微型计算机的产生。由于微型计算机采用了集成度相当高的元、器件,体积大为缩小。但微型计算机在系统结构上与其它各类计算机并无本质区别,

而且在系统逻辑结构, 工艺制造水平等方面还有新发展。所以, 微型计算机除具备电子计算机的特点外, 还有一些它自身的特点。

### 1. 体积小、重量轻

随着大规模集成电路技术的迅速发展, 组成微型计算机的各种芯片的集成度越来越高, 微型计算机将进一步微型化。八十年代中期, 微型计算机迅速地从台式机向便携式机方向发展。世界各大计算机公司竞相推出各种体积更小、重量更轻的新机种, 如: 笔记本式、笔式等。笔记本式便携机体积适中, 大小一般为  $29.7 \times 22.6 \times 4.80 \text{cm}^3$ , 重量在  $2.3 \sim 2.63 \text{kg}$  之间, 最近, 深圳“英达利”信息通讯产品有限公司与香港生产力发展局共同研制出当今世界上最小的计算机(PC机)。该机只有衣袋般大小, 装上电池也不过重 280 克。

### 2. 产量大、价格低。

由于集成电路和计算机制造工艺的不断成熟, 自动生产线的采用, 微型计算机已形成批量生产; 加之世界各大计算机厂家的激烈竞争, 微型计算机将不断降价。在国外, 8086 微处理器由最初的 360 美元锐减到现在的 8 美元。据世界上最大的软件公司 Microsoft 总裁比尔·盖茨预言: 95 年奔腾机带 16M 内存、400M 硬盘和 CD-ROM、这样的多媒体 PC 机将降价到 2000 美元。

### 3. 功耗低、可靠性高

微型计算机与同等水平的小型机比, 功耗低  $60 \sim 150\%$ , 使各种元、器件受热影响减少; 另外, 由于采用了大规模集成电路技术, 许多功能组装在一个芯片上, 器件大为减少, 外部焊接点和导线也因此减少, 因此, 故障率下降, 可靠性提高。

### 4. 灵活性、适应性强

由于微型计算机内部采用总线结构, 可以根据需要, 灵活地组成各种不同规模的系统; 同时, 由于构成微型计算机的基本部件大多已系列化、标准化, 增强了微型计算机的可兼容性。

## 第二节 计算机中数的表示方法

### 一、数制

在日常生活中, 人们常采用十进制数, 而计算机所处理的是二进制数, 广泛采用十六进制来表示。下面分别简单地介绍各种数制及其转换关系。

数制是通过它的基数来划分的, 基数就是用来表示数值大小的数字的个数; 同一数字处于不同的数位(称为权), 所表示的大小是不同的, 它的值是该位置所处的权乘以该数字。

通常, 任何一种进制的数均可以用各位上的数字乘以相应的权。

$$N = \pm (a_{n-1} \cdot R^{n-1} + a_{n-2} \cdot R^{n-2} + \cdots + a_1 R^1 + a_0 R^0 + \cdots + a_{-m} \cdot R^{-m})$$

其中,  $m$  为小数部分的位数,  $n$  为整数部分位数;  $a_i$  为数字,  $R$  为基数, 表示进位制。

#### 1. 二进制数

一个二进制数, 具有以下两个特点:

①具有两个不同的数字符号, 即 0 和 1。

## ②逢二进一。

任何一个二进制数可以表示为

$$(B)_2 = \pm (B_{n-1} \times 2^{n-1} + B_{n-2} \times 2^{n-2} + \cdots + B_1 \times 2^1 + B_0 \times 2^0 + B_{-1} \times 2^{-1} + B_{-2} \times 2^{-2} + \cdots + B_{-m} \times 2^{-m}) = \pm \sum_{i=-m}^{n-1} (B_i \times 2^i)$$

例如：

$$1101.10 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2}$$

目前，在计算机中采用二进制数的原因有二。

①数在计算机中是以器件的物理状态来表示的，一个具有两个不同稳定状态并能相互转换的器件就可以表示一位二进制数，因此，采用二进制数简单而且可靠。

②二进制的乘法、除法运算可以通过部分积相加转化为加法运算，而加法运算装置简单、易实现。

二进制的主要缺点是数位太长，不便阅读和书写，人们也不习惯。为此常采用八进制和十六进制作为二进制的缩写形式；为适应人们习惯，输入和输出采用十进制数，由计算机自己完成二进制与十进制之间的相互转换。

## 2. 十六进制数

一个十六进制数的特点为：

①具有十六个数字符号，采用 0~9 和 A~F。这 16 个数字符号与十进数和二进制数之间的关系如表 1-1 所示。

表 1-1 十进制、二进制和十六进制之间关系对照表

| 十进制 | 十六进制 | 二进制  | 十进制 | 十六进制 | 二进制   |
|-----|------|------|-----|------|-------|
| 0   | 0    | 0000 | 9   | 9    | 1001  |
| 1   | 1    | 0001 | 10  | A    | 1010  |
| 2   | 2    | 0010 | 11  | B    | 1011  |
| 3   | 3    | 0011 | 12  | C    | 1100  |
| 4   | 4    | 0100 | 13  | D    | 1101  |
| 5   | 5    | 0101 | 14  | E    | 1110  |
| 6   | 6    | 0110 | 15  | F    | 1111  |
| 7   | 7    | 0111 | 16  | 10   | 10000 |
| 8   | 8    | 1000 |     |      |       |

## ②逢 16 进位。

任何一个十六进制数可表示为：

$$(D)_{16} = \pm (D_{n-1} \times 16^{n-1} + D_{n-2} \times 16^{n-2} + \cdots + D_1 \times 16^1 + D_0 \times 16^0 + D_{-1} \times 16^{-1} + D_{-2} \times 16^{-2} + \cdots + D_{-m} \times 16^{-m}) = \pm \sum_{i=-m}^{n-1} (D_i \times 16^i)$$

其中， $D_i$  的值在范围 1~9 和 A~F 中， $m$ ， $n$  分别是小数和整数部分的位数。

例如：

$$(3AB \cdot 4A)_{16} = 3 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 + 4 \times 16^{-1} + 10 \times 16^{-2}$$

在计算机中，数是用二进制表示的，但在书写时普遍采用十六进制表示。原因如下：

①二进制和十六进制之间存在着一种特殊关系,即  $2^4=16$ , 于是一位十六进制数可以用四位二进制数表示,两者之间的转换十分简捷而又方便。

②当前大部分微型计算机的字长是八位、十六位或三十二位的,都是四的整数倍,书写时容易用十六进制来表示。

### 3. 八进制数

一个八进制数,具有两个特点:

①具有八个不同数字符号,  $0\sim 7$ 。

②逢八进一。

任何一个八进制数也可表示为:

$$(Q)_8 = \pm (Q_{n-1} \times 8^{n-1} + Q_{n-2} \times 8^{n-2} + \cdots + Q_2 \times 8^1 + Q_0 \times 8^0 + Q_{-1} \times 8^{-1} + Q_{-2} \times 8^{-2} + \cdots + Q_{-m} \times 8^{-m}) = \pm \sum_{i=-m}^{n-1} (Q_i \times 8^i)$$

其中,  $Q_i$  的值在  $0\sim 7$  范围内,  $m, n$  仍然表示小数和整数部分的位数。

例如:

$$(1053.24)_8 = 1 \times 8^3 + 0 \times 8^2 + 5 \times 8^1 + 3 \times 8^0 + 2 \times 8^{-1} + 4 \times 8^{-2}$$

由此可以看出,该数值就是各位上的数字乘以相应的权。

## 二、数制转换

### 1. 二~十进制之间转换

二进制数转换成十进制数很简单,将二进制数按权展开并相加即得到相应的十进制数。

八进制数或十六进制数转换成十进制数也同此理。例如:

$$(110.1)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} = 6.5$$

由上例可知,二转十就是分别求出二进制数每一个数位为 1 所表示的十进制数,然后将所有转换的十进制数相加,即按“权”相加。

十进制数转换成二进制数采用基数乘法;即对整数部分除 2 取余,对小数部分乘 2 取整。其道理如下。

十进制数  $N$  用二进制整数表示时可写成:

$$(N)_{10} = (a_{n-1}a_{n-2}\cdots a_1a_0)_2 \\ = a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \cdots + a_1 \times 2^1 + a_0 \times 2^0$$

从上式可知,只要找出系数  $a_{n-1}, a_{n-2}, \cdots, a_1, a_0$ , 就得到相应的二进制数,这些系数只能是 0 或 1。等号左边除  $a_0$  外,其余各项均能被 2 整除,因此用 2 去除十进制数  $N$ , 余数必为  $a_0$ ; 同理,用 2 除  $N/2$ , 得到的余数即为  $a_1$ , 照此做下去,直至所取精度为止,就可得到  $a_{n-1}, a_{n-2}, \cdots, a_1, a_0$  的值。

例如:将 41 变换为二进制数

$$1 \leftarrow 0 \leftarrow 1 \leftarrow 0 \leftarrow 0 \leftarrow 1 \leftarrow \text{取余}$$

$$0 \leftarrow 1 \leftarrow 2 \leftarrow 5 \leftarrow 10 \leftarrow 20 \leftarrow 41 \leftarrow \text{被除数, 除数为 2}$$

即  $(41)_{10} = (101001)_2$  因此,可以将十进制整数转换成二进制数,总结为这样一句话:“除 2 逆取余”。

若将十进制小数 0.875 展开成二进制小数，其形式为：

$$(0.8125)_{10} = (0 \cdot a_{-1}a_{-2}\cdots a_{-m}) \\ = (a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + \cdots + a_{-m} \times 2^{-m})$$

由此多项式可以看出，若等号两边同时乘以 2，则可以得到  $a_{-1}$  的值。即：

$$1.6250 = a_{-1} + a_{-2} \times 2^{-1} + a_{-3} \times 2^{-2} + \cdots + a_{-m} \times 2^{-m+1}$$

由此可知， $a_{-1}=1$ 。等号两边再乘以 2，可得到  $a_{-2}$ ，照此进行下去，直至小数部分为 0 或所取精度为止。

例如：将十进制小数 0.8125 转换成二进制数

$$\begin{array}{ccccccc} 1 & \leftarrow & 0 & \leftarrow & 1 & \leftarrow & 1 & \leftarrow \text{取整} \\ 0 \leftarrow 0.5 & \leftarrow & 0.25 & \leftarrow & 0.625 & \leftarrow & 0.8125 & \leftarrow \text{被乘数, 除数为 2} \end{array}$$

$$\text{即: } (0.8125)_{10} = (0.1101)_2$$

将十进制小数转换成二进制的方法可归纳成一句话：“乘 2 顺取整”。

## 2. 二~十六进制之间的转换

由于十六进制基数 16 与二进制基数之间存在关系：

$$16^1 = 2^4$$

因此一位十六进制数相当于四位二进制数，它们之间关系见表 1-1。

二进制转换成十六进制的方法是：以二进制数的小数点为中心，向两侧每四位划分为一组，最高位或最低位不足四位以零补足，然后把每四位二进制数用相应的十六进制数代替，即可转换成十六进制数。

例如：(1111101010101)<sub>2</sub> 可转换为

$$\begin{array}{cccc} \overline{0011} & \overline{1110} & \overline{1010} & \overline{1000} \\ | & | & | & | \\ 3 & E & A & 8 \end{array}$$

$$\text{因此, } (1111101010101)_2 = (3E.A8)_{16}$$

十六进制转换成二进制的方法是：不论是十六进制整数或小数，只要将每一位十六进制数用四位二进制数表示展开即可。

例如：(3E.A8)<sub>16</sub> 可转换为

$$\begin{array}{cccc} \overline{3} & \overline{E} & \overline{A} & \overline{8} \\ | & | & | & | \\ 0011 & 1110 & 1010 & 1000 \end{array}$$

$$\text{因为, } (3E.A8)_{16} = (00111110.10101000)_2$$

## 三、码制

上面提到的二进制数，没有涉及数的符号。要完整地表示一个数，不仅要指出其数值，还要指出数的正或负，也即数总是由数符和数值两部分组成的。例如：

$$\begin{array}{c} +1011101 \\ \uparrow \quad \uparrow \\ \text{数符} \quad \text{数值} \end{array}$$

而在计算机中数的正、负也必须以二进制数来表示，以“0”表示正号，为“1”表示负号。

通常，将最高位作为符号位。我们将数码化数符的二进制数称为机器数，其值称为机器数的真值。

在计算机中，带符号二进制数常采用三种表示方法，即原码表示法、反码表示法和补码表示法。由于反码在运算中对计算机结构有特别要求，不常采用，引出反码是为了能方便求得补码。所以，这里着重介绍原码的补码。

### 1. 原码

在带符号二进制数原码表示中，符号位为 0 表示正数，符号位为 1 表示负数，其余各位为数值本身。

对于正数  $x = +1010100$ ,  $[x]_{\text{原}} = \underset{\substack{\uparrow \\ \text{数符}}}{0} \underset{\substack{\uparrow \\ \text{数值}}}{1010100}$

对于负数  $x = -1010100$ ,  $[x]_{\text{原}} = \underset{\substack{\uparrow \\ \text{数符}}}{1} \underset{\substack{\uparrow \\ \text{数值}}}{1010100}$

对于零

数“0”是一个特殊的数，它无正，负之分；即可以认为它是+0，也可以认为它是一0，这样，零的原码表示就有两种：

$[+0]_{\text{原}} = \underset{\substack{\uparrow \\ \text{数符}}}{0} \underset{\substack{\uparrow \\ \text{数值}}}{0000000}$  或  $[-0]_{\text{原}} = \underset{\substack{\uparrow \\ \text{数符}}}{1} \underset{\substack{\uparrow \\ \text{数值}}}{0000000}$

计算机遇到这两种情况均当作 0 来处理。

综上所述,  $[x]_{\text{原}} = \begin{cases} x & \text{当 } x > 0 \text{ 或 } x = +0 \text{ 时} \\ 2^{n-1} - x & \text{当 } x < 0 \text{ 或 } x = -0 \text{ 时} (n \text{ 为字长}) \end{cases}$

对于加、减法：两数运算时，首选判别两数符号是否相同。若两数符号相同，则两数的数值相加，其结果的符号就是参加运算的数的符号；若两数符号不同，则以绝对值大的减去绝对值小的数，就是结果的数值，结果的数符就是两数中绝对值大的那个数的数符。

对于乘、除法：计算机做二进制原码乘、除法运算时和十进制数乘、除法一样，对数符的运算和对数值的求积（或商）运算是分别进行的。对数值的求积（或商）通常分解为一系列加（或商）法及移位操作来实现的，而对积（或商）的符号是按一定规则得到的，见表 1-2。

表 1-2 被乘（除）数、乘（除）数及积（商）数符之间关系

| 被乘数（被除数）符号 | 乘数（除数）符号 | 积（商）符号 |
|------------|----------|--------|
| 0          | 0        | 0      |
| 1          | 1        | 0      |
| 0          | 1        | 1      |
| 1          | 0        | 1      |

这种数符之间关系，在数字逻辑中通过半加（或异或）的逻辑关系，很易实现。最后将分别求得的积或商的数值及其符号拼在一起，得到一个完整结果。

### 2. 反码

在带符号二进制数反码表示中，正数的反码等于其原码，负数的反码是把该数的数值

位按位取反，符号位仍然为1。

对于正数， $[x]_{\text{反}} = [x]_{\text{原}}$

$$x = +1101001 \quad [x]_{\text{反}} = 01101001$$

对于负数，

$$x = -1101001 \quad [x]_{\text{反}} = 10010110$$

对于零，它有两种表示法  $[+0]_{\text{反}}$  或  $[-0]_{\text{反}}$ 。

$$[+0]_{\text{反}} = 00000000 \text{ 或 } [-0]_{\text{反}} = 11111111$$

因此，在反码表示中，“0”的表示不唯一。

### 3. 补码

原码表示法，对加、减法运算较为复杂，相应地其硬件结构也较复杂。若能将减法运算转化为加法运算将简化硬件结构的复杂程度。补码表示法可以将负数转化为正数，使减法转化为加法，这样就解决了原法表示法在加减法运算中硬件结构复杂的矛盾。

#### (1) 补码的概念

设有一台时钟，指针指在10时，而标准时间是6时，要拨准时钟有两种方法：

其一是倒拨四时，即  $10 - 4 = 6$

其二是顺拨八时，即  $10 + 8 = 12 + 6 = 6$   
└──自动丢失

两种方法得到相同的结果，说明  $(10 - 4)$  可以用  $(10 + 8)$  代替。自动丢失的数“12”称为模(mod)。模和某数  $x$  (例如  $-4$ ) 相加所得的数，如  $[12 + (-4) = 8]$  称为该数的补码。即：

$$[x]_{\text{补}} = \text{模} + x$$

所以， $10 - 4 = 10 + [-4]_{\text{补}} = 10 + (12 + (-4)) = 10 + 8 = 6 \bmod 12$

由此可见，有模的数，其减法可以化做加法。计算机中的数是有模的数，假设机器字长为  $n$  位，则它的模就是  $2^n$ 。这是一个  $n+1$  位的数  $\frac{n \text{ 位}}{1 \ 00 \cdots 0}$ ，由于机器字长为  $n$  位，因此大于或等于模的数，将会在机器字之外丢失一个模。

#### (2) 补码表示

正数的补码表示：设  $x = +1010100$

$$[x]_{\text{补}} = \text{模} + x = 2^8 + 1010100 = 01010100 \pmod{2^8}$$

由此可见，正数的补码等于原码。

负数的补码表示：设  $x = -1010100$ ，

$$[x]_{\text{补}} = \text{模} + x = 2^8 - 1010100 = 10101100 \pmod{2^8}$$

在计算机中，引出二进制数的补码表示，是为了化减法为加法，从而简化机器中的运算装置。但是在上述求负数的补码中，仍然用了减法，这样就没有起到简化运算，简化硬件运算装置的作用。而二进制补码可以不用减法，可通过另一种方法方便地求出。即首先求出该负数反码，然后在末位加1。这种求补码的方法硬件实现上就来得容易。

$$\text{设 } x = -1010100 \quad [x]_{\text{补}} = [x]_{\text{反}} + 1 = 10101100$$

零的补码表示：

$$\text{设 } x = +0000000, \text{ 则 } [x]_{\text{补}} = 2^8 + 0000000 = 00000000$$

$$\text{设 } x = -0000000, \text{ 则 } [x]_{\text{补}} = 2^8 - 0000000 = 00000000$$