



高等学校
电子信息类 规划教材

(第四版)

微型计算机原理

姚燕南 薛钧义 主编

西安电子科技大学出版社

<http://www.xduph.com>



高等学校
电子信息类 规划教材

微型计算机原理

(第四版)

姚燕南 薛钧义 编
欧 文 茹 锋

西安电子科技大学出版社

2000

内 容 简 介

本书以 80386/80486 为样板机,介绍微型计算机的数据类型,微处理器结构及微型计算机工作原理,半导体存储器与存储器管理技术,寻址方式、指令系统及汇编语言程序设计,中断、异常及输入输出接口技术。最后,对数学协同处理器、微机系统及其操作系统作了简介。本书的特点是注意了由浅入深、循序渐进,并从应用角度出发,软硬相结合地讲述了基本原理及使用方法,每章都有习题和思考题。

本书系高等院校电子信息类控制专业的重点规划教材,也适用于相近专业,并可作为从事微机开发应用方面的技术人员的参考用书。

图书在版编目(CIP)数据

微型计算机原理/姚燕南等编. —4版.

—西安:西安电子科技大学出版社,2000.2

ISBN 7-5606-0773-X

I. 微… I. 姚… II. 微型计算机-基本知识 N. TP36

中国版本图书馆 CIP 数据核字(1999)第 31967 号

责任编辑 夏大平

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)8227828 邮 编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 陕西画报社印刷厂

版 次 2000年2月第4版 2000年8月第2次印刷

开 本 787毫米×1092毫米 1/16 印张 28.5

字 数 674千字

印 数 6 001~12 000 册

定 价 29.00 元

ISBN 7-5606-0773-X/TP·0397

*** 如有印装问题可调换 ***

本书封面贴有西安电子科技大学出版社的激光防伪标志,无标志者不得销售。

出版说明

为做好全国电子信息类专业“九五”教材的规划和出版工作，根据国家教委《关于“九五”期间普通高等教育教材建设与改革的意见》和《普通高等教育“九五”国家级重点教材立项、管理办法》，我们组织各有关高等学校、中等专业学校、出版社，各专业教学指导委员会，在总结前四轮规划教材编审、出版工作的基础上，根据当代电子信息科学技术的发展和面向 21 世纪教学内容和课程体系改革的要求，编制了《1996—2000 年全国电子信息类专业教材编审出版规划》。

本轮规划教材是由个人申报，经各学校、出版社推荐，由各专业教学指导委员会评选，并由我们与各专指委、出版社协商后审核确定的。本轮规划教材的编制，注意了将教学改革力度较大、有创新精神、有特色风格的教材和质量较高、教学适用性较好、需要修订的教材以及教学急需、尚无正式教材的选题优先列于规划。在重点规划本科、专科和中专教材的同时，选择了一批对学科发展具有重要意义，反映学科前沿的选修课、研究生课教材列入规划，以适应高层次专门人才培养的需要。

限于我们的水平和经验，这批教材的编审、出版工作还可能存在不少缺点和不足，希望使用教材的学校、教师、学生和其他广大读者积极提出批评和建议，以不断提高教材的编写、出版质量，共同为电子信息类专业教材建设服务。

电子工业部教材办公室

12/566

前 言

本教材系按原电子工业部的《1996—2000 年全国电子信息类专业教材编审出版规划》，由自动控制专业教学指导委员会编审、推荐出版。本教材由西安交通大学姚燕南、薛钧义担任主编，主审谢剑英，责任编委陈怀琛。

本教材的参考学时数 90 学时，主要内容是：以 80386/80486 为样板机，主要介绍了微型计算机的数据类型，微处理器结构及微型计算机工作原理，半导体存储器及存储器管理技术，寻址方式、指令系统与汇编语言程序设计(包括 DOS 系统功能调用的应用)，中断、异常及输入输出接口技术，最后对数学协同处理器、微机系统及其操作系统作了简介。在输入输出接口技术一章中，除对通用接口 8255A、8253/8254、8251 及 8237A 进行介绍外，还对多功能 I/O 接口电路 82380 作了介绍。在微机系统简介中对最新流行总线、外设及系统(包括多媒体微机系统)作了介绍。操作系统一节中对 DOS 和 Windows 操作系统都作了简介。本教材在编写过程中注意了由浅入深、循序渐进，对基本概念讲述清楚，并有大量实例，具有软硬结合、图文并茂、内容丰富及取材较新的特点，不仅适合作教材，而且可作为工程技术人员的自学用书。

使用本教材时应注意先修数字电子技术课程。其教学方法应注重理论与实践相结合，要多上机、多实践，这样才能有好的效果。

本教材由姚燕南编写第 1、2、3、5、7 章，薛钧义编写第 4、8 章，欧文编写第 9 章，姚燕南及欧文共同编写第 10 章，茹锋编写第 6 章。谢剑英及陈怀琛两位教授为本书提出了许多宝贵意见，谢剑英教授还为本书的审阅付出了辛勤的劳动。在此对陈怀琛及谢剑英两位教授表示诚挚的感谢。

由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

编 者

1998 年 12 月

第三版前言

本教材系由“自动控制”教材编委会评选审定，并推荐再版。

为了适应国民经济高速发展的需要，除对原稿基础部分在文字上作了进一步压缩和修订外，已将原来以 Z80 为背景机改为以 8086/8088 16 位机为背景机，讲述微型机原理、编程及接口技术，并尽量在全书中反映当前最新技术，力求做到既深入浅出，又具有一定的先进性。

本课程参考学时数为 100 学时，其主要内容为：(1) 计算机概述(第一、二、四章)，介绍计算机的数制和码制，并从教学模型机入门，讲解计算机的整机工作原理。以 8085 及 8086/8088 为背景机介绍微型计算机的结构和组成。(2) 8086/8088 微处理机的指令系统、寻址方式、时序波形及汇编语言程序设计(第五、六、七章)。(3) 各种存储器和 I/O 接口电路以及中断处理和输入输出接口技术(第三、八、九章)。(4) 微型机系统及 IBM-PC 机；微机操作系统及 PC-DOS 简介(第十章)。使用本教材时应注重实践和应用。

本教材由姚燕南同志编写第一、二、三、四、八、十章；由薛钧义同志编写第五、六、九章；由薛钧义及姚燕南共同编写第七章。

本书在编写过程中得到胡保生教授及谢剑英教授的大力支持和指导，在此表示诚挚的感谢。由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

编者

于西安交通大学 1994.1.18

引 论

1. 微处理器及微型计算机

微处理器或微处理机亦称中央处理单元(Central Processing Unit), 简称 CPU。它是将计算机的控制逻辑和运算单元集成在一个芯片上实现的。通常, 微处理器中不包含内存储器及输入输出接口电路。内存储器是独立于 CPU 之外的芯片或芯片组; 输入输出接口电路也常独立地做在一个芯片上。由于输入输出设备的多样性, 使得接口电路各有特色。有时, 为了满足特殊外部设备的要求, 用户还必须自己设计专用的接口电路。

微型计算机由微处理器、内存储器、输入输出设备及其接口电路组成。若将微处理器、内存储器及输入输出接口电路集成在一个芯片上便称为单片微型计算机。微型计算机再加上软件便构成一个微型计算机系统。

微处理器内大多数操作中的数据位数(如 8 位、16 位、32 位等)是它的重要特征, 根据这个位数分别称其为 8 位 CPU、16 位 CPU、32 位 CPU 等。同时, 也将相应的微型计算机称为 8 位机、16 位机和 32 位机。

2. 微处理器发展史

第一代微处理器的典型产品是 Intel 公司 1971 年生产的 4004(4 位 CPU)及 8008(8 位 CPU)。采用 PMOS 工艺, 集成度约为 2 000 只晶体管/片。平均指令执行时间约达 $10\ \mu\text{s}$ ~ $20\ \mu\text{s}$ 。采用机器语言编程, 其主要用途是取代传统的复杂逻辑电路, 如智能打字机控制电路、交通灯控制电路及微波炉控制电路等。

第二代微处理器的典型产品是 1974 年 Intel 公司生产的 8080 CPU、Zilog 公司的 Z80 CPU、Motorola 公司的 6800 CPU 以及 Intel 公司的 8085 CPU(1976 年)。它们均为 8 位微处理器, 具有 16 位地址总线, 采用 NMOS 工艺, 集成度约为 9 000 只晶体管/片, 指令的平均执行时间为 $1\ \mu\text{s}$ ~ $2\ \mu\text{s}$ 。由第二代微处理器构成的微机系统(如 Apple - II 等)已配有单用户操作系统(如 CP/M 单用户操作系统), 并可采用汇编语言及 BASIC、FORTRAN 等高级语言。8 位微型机主要用于控制。

第三代微处理器的典型产品是 1978 年 Intel 公司生产的 8086 CPU、Zilog 公司的 Z8000 CPU 和 Motorola 公司的 68000 CPU。它们均为 16 位微处理器, 具有 20 位地址总线。8088 CPU 是 Intel 公司 1979 年推出的 8 位副产品, 或称准 16 位机(内部数据总线 16 位, 外部数据总线 8 位)。1981 年 IBM 公司采用 8088 CPU 研制出风靡全球的 PC(Personal Computer)机(时钟频率为 4 MHz), 使 8086/8088 CPU 成为世界上最主要的微处理器结构。

1982 年 Intel 公司又生产了 80186/80188 及 80286 CPU, 它们也属于 16 位微处理器, 其中 80186/80188 的功能与 8086/8088 CPU 完全一样, 只是将 8086/8088 微处理器及其

系统中所需要支持芯片都集成在一个芯片上。80286 CPU 却是在 8086 CPU 基础上增加了许多功能,它具有 24 位地址总线,并具有多任务系统所必需的任务转换功能、存储器管理功能以及各种保护功能,即能以多任务方式运行 8086 应用程序。

8086/8088、80186/80188、80286 以及以后的 80386、80486 CPU 等形成了 80x86 系列微处理器。80x86 系列微处理器的每一代产品都配有相应的浮点运算协处理器,又称数学协处理器,如 8087、80187、80287、80387 等。

16 位微处理器组成的微型计算机系统已能替换部分小型机的功能,采用了多种高级编程语言,并普及了 CP/M 及 MS-DOS 磁盘操作系统,其中高档 16 位微处理器(如 80286)组成的微机系统(如 IBM PC/AT),时钟频率为 6 MHz~20 MHz,采用了多用户操作系统。

第四代微处理器的典型产品是 1983 年 NS 公司推出的 32032(NMOS 工艺)、1984 年 Motorola 公司推出的 68020(HCMOS)、1985 年、1990 年 Intel 公司先后推出的 80386、80486(CHMOS 工艺),它们均为 32 位微处理器,具有 32 位地址总线,其中 80386 又分 80386 SX 和 80386 DX 两种。80386 SX 内部数据总线 32 位,外部数据总线 16 位,配用 80287 协处理器;80386 DX 内部及外部数据总线均为 32 位,配用 80387 协处理器。80386 CPU 具有完善的段页式存储器管理机制,时钟频率为 12.5 MHz~33 MHz。

80486 CPU 在 16 mm×11 mm 芯片上集成了约有 120 万只晶体管,它相当于将 80386、80387 及 8 KB 高速缓冲存储器集成在一块芯片上(若芯片上没有数学协处理器称为 80486 SX),时钟频率可达 50 MHz~66 MHz,性能也较 80386 有较大提高。

32 位微处理器采用多用户多任务操作系统,如 UNIX、Windows 等,具有丰富的多种高级编程语言,主要用于个人计算机和 workstation。

第五代微处理器的典型产品是 1993 年 Intel 的 Pentium(奔腾, Intel 586),IBM、Apple 和 Motorola 三公司合作生产的 Power PC。Pentium 微处理器数据总线 64 位,地址总线 36 位,具有两条超标量流水线,两个并行执行单元及双高速缓冲存储器,工作频率为 50 MHz、66 MHz、133 MHz 和 166 MHz。Power PC 是一种精简指令集计算机 RISC(Reduced Instruction Set Computer),也是一种优异的 64 位数据总线微处理器。特点是指令格式规整,功能简单,从而使指令译码电路也变得简单,译码速度更快。另外,指令系统设计较为紧凑,只设置了一些使用频率高的常用指令,因而指令条数少,指挥指令执行的控制逻辑电路也简单,指令执行速度快。Power PC 中也采用了先进的超标量流水线技术及双高速缓存。与精简指令集相对应的是复杂指令计算机 CISC(Complex Instruction Set Computer),Pentium 微处理器便是 CISC。

第六代微处理器的典型产品是 Pentium Pro、MMX Pentium 和 Pentium II。

Pentium Pro 微处理器属于多芯片模式或称为双腔 PGA(Pin-Grid-Array),它将 CPU 芯片(550 万个晶体管)和二次高速缓冲存储器(1 550 万个晶体管)集成在一个 387 脚的陶瓷封装内。Pentium Pro 微处理器较 Pentium 更为先进,它使用三条超标量流水线,有 5 个并行执行单元,8 KB 一次程序高速缓冲存储器和 8 KB 一次数据高速缓冲存储器(简称缓存),并采用错序执行、动态转移预测等技术。Pentium Pro 微处理器的主振频率为 200 MHz~250 MHz,每秒能作 2.5 亿次浮点运算。此外,在多媒体应用方面 Pentium Pro 也显示出更优越的特性。

Pentium Pro 微处理器的性能虽优越,但全速同步二级高速缓存与 CPU 封装在一起,使其价格居高不下(全速同步高速缓存需要价格昂贵的高速静态 RAM 支持),不能成为第六代微处理器主流产品。面对这种情况,Intel 公司一面加紧开发研制新产品,一面对 Pentium 进行改进。在提高 Pentium 芯片主振频率的同时,加入了多媒体扩展技术 MMX(Multimedia Extension),并于 1997 年 1 月推出 MMX Pentium(多能奔腾)。多能奔腾微处理器出台后迅速占领市场,成为第六代微处理器典型产品之一。

Pentium II 微处理器是 1997 年 5 月推出的另一种第六代微处理器,它是在对 Pentium Pro 进行改进并增加多媒体扩展功能后推出的,是 Pentium Pro 与 MMX Pentium 的结合。无论在运行速度上,还是多媒体处理方面均呈现出高性能,且价格较低,得到很好普及。

第五代及第六代微处理器的性能和速度都已经与中小型计算机抗衡,能采用 DOS、UNIX、Windows、Windows 95 及 Windows NT、Windows 98 等多种操作系统,使用多种高级编程语言,特别是 Visual Basic、Visual C++、Visual FoxPro 等语言的使用,更使它们放出夺目的光彩。

目 录

引论

第 1 章 微型计算机中的数据类型	1
1.1 常用数据类型	1
1.2 数学协处理器的数据格式	14
习题和思考题	17
第 2 章 微处理器结构及微型计算机工作原理	18
2.1 微型计算机的组成及工作原理	18
2.2 8086/8088 及 80286 微处理器的功能结构	20
2.3 80386 微处理器的功能结构	25
2.4 80486 微处理器的功能结构	36
2.5 Pentium 级微处理器简介	39
习题和思考题	43
第 3 章 半导体存储器	45
3.1 概述	45
3.2 随机存取存储器 RAM	47
3.3 只读存储器 ROM	58
3.4 高速缓冲存储器 Cache	65
习题和思考题	70
第 4 章 80x86 寻址方式与指令系统	71
4.1 计算机指令格式	71
4.2 指令长度与字长的关系	73
4.3 80x86 指令(编码)格式及寻址方式	73
4.4 8086/8088 指令系统	82
4.5 80x86 的寻址方式及新增的指令	113
习题和思考题	122
第 5 章 存储器管理	124
5.1 实方式存储器管理	124
5.2 保护方式存储器管理	126
5.3 保护及任务切换	139
5.4 虚拟 8086 方式	156
习题和思考题	157
第 6 章 汇编语言程序设计	158
6.1 汇编语言语法	158
6.2 汇编语言程序设计	187

6.3 汇编程序及上机过程	216
6.4 DOS 及 BIOS 功能调用	224
6.5 任务切换与混合语言编程	233
习题和思考题	238
第 7 章 中断和异常	240
7.1 概述	240
7.2 中断	242
7.3 异常	243
7.4 中断及异常的暂时屏蔽	248
7.5 中断及异常的优先级	249
7.6 实方式下的中断	249
7.7 保护方式下的中断和异常	253
7.8 中断优先级管理器 8259A PIC	257
习题和思考题	269
第 8 章 输入/输出方法及常用的接口电路	271
8.1 概述	271
8.2 I/O 端口的编址及基本输入/输出方法	273
8.3 8255A 并行接口电路	280
8.4 可编程计数/定时器 8253/8254	292
8.5 串行通信及 8251 串行接口电路	308
8.6 DMA 控制器 8237A	329
8.7 多功能 I/O 接口电路 82380	350
习题和思考题	357
第 9 章 数学协处理器	359
9.1 概述	359
9.2 协处理器的逻辑结构	361
9.3 协处理器指令系统及应用	369
习题和思考题	384
第 10 章 微型计算机系统及其操作系统简介	385
10.1 微型计算机系统的组成	385
10.2 微型计算机系统的总线	386
10.3 微型计算机系统的常用外部设备	394
10.4 微型计算机系统举例	398
10.5 多媒体计算机简介	402
10.6 微型计算机操作系统简介	411
习题和思考题	416
附录 A 8086/8088 指令系统	417
附录 B 80x86 指令系统参考资料	436
参考文献	443

第 1 章

微型计算机中的数据类型

1.1 常用数据类型

为了解决一个实际问题在微型机上编程，首先要熟悉微型机中的数据类型。80x86 系列微机中，常用的数据类型包括：带符号整数、无符号整数、BCD 数(包括压缩的和非压缩的二一十进制码)、字符串、位、浮点数。

1.1.1 数据在内存储器中的存储方式

微型机中的内存储器(简称内存)用来存储参加运算的操作数、运算的中间结果和最后结果。数据在内存中常以字节(8 位二进制数称为一个字节)Byte 为单位进行存储，所以，可以说内存储器是有唯一地址的字节的有序阵列。一个字节占用内存的一个地址，称为一个存储单元。存储单元的地址从 0 开始，直到 CPU 所能支持的最高地址。80386 有 32 根地址总线，共有 2^{32} ($4\text{ G} \approx 4 \times 10^9$) 个存储单元。

当二进制数的位数超过 8 位，且为 8 位的倍数时，就需用多个相邻字节来存放。通常称两个相邻字节组成的 16 位二进制数为一个字(Word)，称 4 个相邻字节组成的 32 位二进制数为一个双字(DWord 或 Double Word)。80x86 系列微机中，多字节数据的存储采取高位字节在地址号高的存储单元中，低位字节在地址号低的存储单元中的规则，见图 1.1。

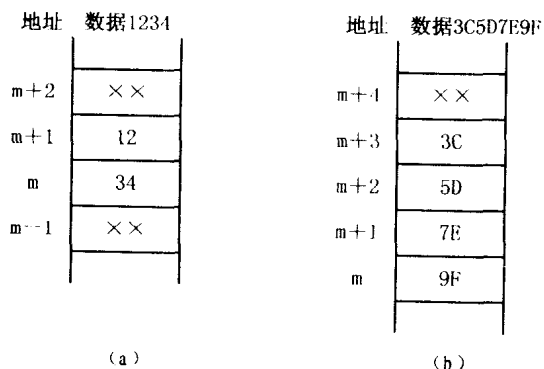


图 1.1 多字节的存储
(a) 字的存储；(b) 双字的存储

对于一个字数据来说，若用来存储它的相邻两个存储单元的最低地址号为 m ，则该字中低位字节数据便存放于地址号为 m 的存储单元中；高位字节数据存储在地地址号为 $m+1$ 的存储单元中，且称该字的地址为 m 。对于一个双字数据来说，若用来存储它的相邻 4 个存储单元的最低地址号为 m ，则该双字的最低字节存放于地址号为 m 的存储单元中；最高字节存放在地址号为 $m+3$ 的存储单元中，且称该双字的地址为 m 。

1.1.2 数的进位制表示约定

微型机中对各种进位制数表示时，用后缀字母加以区别。二进制数后缀为 B(Binary)，如 1000B(十进制值为 8)；十六进制数后缀为 H(Hexadecimal)，如 1000H(十进制值为 4096)；十进制数后缀为 D(Decimal)或不加后缀。表 1.1 列出二进制数、十六进制数及十进制数对照表。

表 1.1 十六进制数、二进制数、十进制数对照表

十六进制	二进制	十进制	十六进制	二进制	十进制
0	0	0	A	1010	10
1	1	1	B	1011	11
2	10	2	C	1100	12
3	11	3	D	1101	13
4	100	4	E	1110	14
5	101	5	F	1111	15
6	110	6	10	10000	16
7	111	7	11	10001	17
8	1000	8	⋮	⋮	⋮
9	1001	9			

1.1.3 整数

80x86 系列微处理器中，参加运算的整数操作数可为 8 位长的字节、16 位长的字；80386/80486 CPU 中，参加运算的整数操作数还可为 32 位长的双字。整数分带符号数和无符号数两种。

1. 无符号数

所谓无符号数是指字节、字、双字整数操作数中，对应的 8 位、16 位、32 位二进制数全部用来表示数值本身，没有用来表示符号位的位，因而为正整数。它们在内存中的存放格式见图 1.2，其中位 0 为最低有效位 LSB，位 7、位 15、位 31 分别为字节、字、双字的最高有效位。

2. 带符号整数

带符号整数编码的表示法有 4 种：原码、反码、补码及移码，它们同样具有字节、字及双字 4 种不同长度的整数类型。

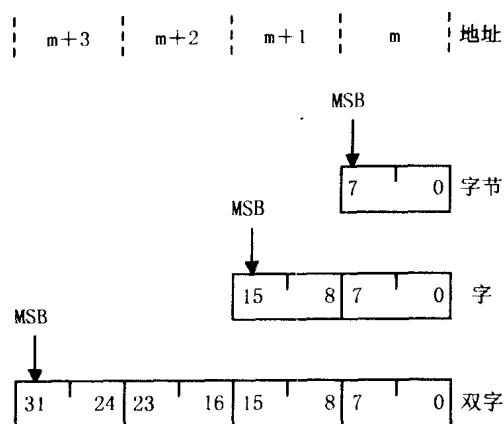


图 1.2 无符号整数

1) 原码

对一个二进制数而言,若用最高位表示数的符号(常以 0 表示正数,1 表示负数),其余各位表示数值本身,则称为该二进制数的原码表示法。例如,设 $X = +1011100$, $Y = -1011100$, 则 $[X]_{\text{原}} = 01011100$, $[Y]_{\text{原}} = 11011100$ 。 $[X]_{\text{原}}$ 和 $[Y]_{\text{原}}$ 分别为 X 和 Y 的原码,是符号数值化了的数,可在计算机中使用,称为机器数。原来的带正负号的数 X 和 Y 称为相应机器数的真值。原码 $[X]_{\text{原}}$ 和真值 X 之间的关系如下:

(1) 正数的原码表示

设 $X = +X_{n-2}X_{n-3}\cdots X_1X_0$ (即 $n-1$ 位二进制正数), 则

$$[X]_{\text{原}} = 0X_{n-2}X_{n-3}\cdots X_1X_0 \quad (n \text{ 位二进制数, 其中最高位为符号位})$$

(2) 负数的原码表示

设 $X = -X_{n-2}X_{n-3}\cdots X_1X_0$ (即 $n-1$ 位二进制负数), 则

$$\begin{aligned} [X]_{\text{原}} &= 1X_{n-2}X_{n-3}\cdots X_1X_0 = 2^{n-1} + X_{n-2}X_{n-3}\cdots X_1X_0 \\ &= 2^{n-1} - (-X_{n-2}X_{n-3}\cdots X_1X_0) = 2^{n-1} - X \end{aligned}$$

它也是一个 n 位二进制数, 其中最高位为符号位。

(3) 零的原码表示

在二进制数原码表示中有正零和负零之分, 即

$$\left. \begin{aligned} [+0]_{\text{原}} &= 000\cdots 00 \\ [-0]_{\text{原}} &= 100\cdots 00 \end{aligned} \right\} \quad (n \text{ 位二进制数, 最高位为符号位})$$

综上所述, 原码和真值的关系可归纳为如下数学定义式:

$$[X]_{\text{原}} = \begin{cases} X & X \geq +0 \\ 2^{n-1} - X & X \leq -0 \end{cases} \quad (n \text{ 位二进制数, 最高位为符号位})$$

2) 补码和反码

一个二进制数, 若以 2^n 为模, 它的补码叫做 2 补码, 以后把 2 补码简称补码, 即 $[X]_{\text{补}} = 2^n + X$ 。同理, 一个十进制数, 若以 10^n 为模, 它的补码叫 10 补码。而一个二进制数, 若以 $2^n - 1$ 为模, 它的补码叫做 1 补码。 $2^n - 1$ 实际上就是 n 个 1, 即 $111\cdots 11$, 这是 1 补码的由来。1 补码也称反码, 即 $[X]_{\text{反}} = 2^n - 1 + X$ 。

根据 $[X]_{\text{补}}$ 及 $[X]_{\text{反}}$ 的定义式得如下关系:

当 X 为正数时, $[X]_{\text{反}} = [X]_{\text{补}} = [X]_{\text{原}} = X$;

当 X 为负数时, $[X]_{\text{反}}$ 是将 $[X]_{\text{原}}$ 的符号位保持不变, 其余各位变反; $[X]_{\text{补}}$ 则是将 $[X]_{\text{原}}$ 的符号位保持不变, 其余各位变反加 1。

例如, 若 $X_1 = +1010011\text{B}$, $X_2 = -1010011\text{B}$, 则

$$[X_1]_{\text{补}} = [X_1]_{\text{反}} = [X_1]_{\text{原}} = 01010011\text{B};$$

$$[X_2]_{\text{原}} = 11010011\text{B}, [X_2]_{\text{反}} = 10101100\text{B}, [X_2]_{\text{补}} = 10101101\text{B}$$

当 X 为 $+0$ 时, 则

$$[+0]_{\text{补}} = 2^n + 00 \cdots 00 = 0$$

$$[+0]_{\text{反}} = [+0]_{\text{原}} = 00 \cdots 00 = 0$$

当 X 为 -0 时, 则

$$[-0]_{\text{补}} = 2^n - 00 \cdots 00 = 0$$

$$[-0]_{\text{反}} = 111 \cdots 11$$

3) 移码

移码是在数的真值上加一个偏移量形成的, 它定义如下:

$$[X]_{\text{移}} = 2^{n-1} - 1 + X \quad 2^{n-1} \geq X > -2^{n-1}$$

其中 X 表示二进制数真值, n 表示包括符号位和数值部分在内的二进制数位数, $2^{n-1} - 1$ 为偏移量。例如, 若 $X = +10010\text{B} = +18$, $Y = -10010\text{B} = -18$, 则 $[X]_{\text{移}} = 110001\text{B}$, $[Y]_{\text{移}} = 001101\text{B}$ 。

根据移码定义, 可得如下结论: 移码在数轴上表示的范围恰好相当于它的真值在数轴上的表示范围向正方向移动 $2^{n-1} - 1$ 个单位 (见图 1.3), 故有移码 (即偏移二进制码) 之称。

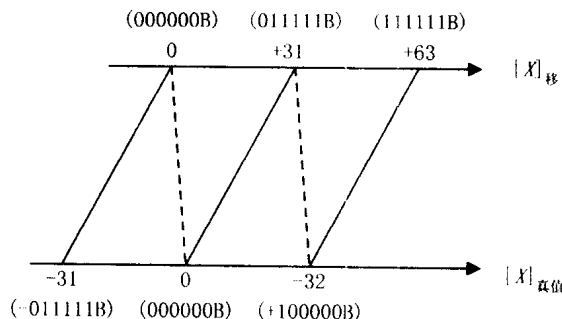


图 1.3 移码在数轴上的表示

3. 带符号数的运算

上已述及, 计算机中带符号数的表示法有 4 种: 原码、反码、补码及移码。

用原码法表示数时, 仅仅是将真值的符号用一位二进制数来表示, 因而原码数的运算完全类同于正负数的笔算。比如两个正数相减, 如果被减数的绝对值小于减数的绝对值, 笔算时, 就必须把两者颠倒过来, 再作减法, 并把求得的差加上负号。用计算机实现时, 过程是先比较两个数绝对值的大小, 然后决定是颠倒过来相减还是直接相减, 最后在结果前面加上正确的正负号。由于原码数的运算类同于笔算, 因此处理过程非常繁琐, 要求计算机的结构也极为复杂。但是, 原码法表示数时, 最大优点是直观。浮点数的有效数字常用

原码表示，二进制乘除法运算时，也多采用原码表示法。

若计算机中的带符号二进制数以补码形式表示，则当两个二进制数进行补码加减法运算时，一方面可以使符号位数一起参加运算；另一方面还可以将二数相减变为减数变补与被减数相加来实现，即

$$[X \pm Y]_{\text{补}} = [X]_{\text{补}} + [\pm Y]_{\text{补}} \quad |X|, |Y|, |X+Y| < 2^{n-1}$$

其中 $[-Y]_{\text{补}} = [Y]_{\text{变补}}$ 。它是将 $[Y]_{\text{补}}$ 包括符号位在内全部变反加 1 求得的(或是先由 $[Y]_{\text{补}}$ 求出 $[-Y]_{\text{原}}$ ，再求 $[-Y]_{\text{补}} = [Y]_{\text{变补}}$)。由于补码进行减法运算时可用加法来代替，且符号位也可以和数一起参加运算，使计算机的运算速度大大提高，同时也简化了计算机的硬件结构。因而，计算机在进行加减法运算时，都采用补码法表示数。然而，采用补码法表示数时，最大的缺点是其表示方法与人们习惯的表示法不一致。因而，从它的表示形式来判断一个数真值的大小很易出错。例如，若 $X = +10010\text{B} = +18$ ， $Y = -10010\text{B} = -18$ ，则 $[X]_{\text{补}} = 010010\text{B}$ ， $[Y]_{\text{补}} = 101110\text{B}$ ，故很容易按人们的习惯表示法判断出 $101110\text{B} > 010010\text{B}$ (即 $-18 > +18$)而出错。

反码法表示二进制数时，由于进行运算很不方便，已很少用在算术运算中，仅在计算机进行求反逻辑运算时用。

移码法表示数时，克服了补码表示法与人们习惯表示法不一致的缺点。它使最小的负数变为 0，最大的正数变为最大数(11...11)，见表 1.2。因而，采用移码法表示数时可很方便地判断出其真值大小。移码表示法常用作模/数(A/D)转换器和数/模(D/A)转换器的双极性编码，也可用在浮点数的阶码中。

表 1.2 带符号整数格式

十进制值	2 的补码	2 的反码	移码(偏移量=127)	带符号的值
128	无此值	无此值	11111111B	无此值
127	01111111B	01111111B	11111110B	01111111B
126	01111110B	01111110B	11111101B	01111110B
⋮	⋮	⋮	⋮	⋮
2	00000010B	00000010B	10000001B	00000010B
1	00000001B	00000001B	10000000B	00000001B
0	00000000B	00000000B	01111111B	00000000B
-0	无此值	11111111B	无此值	10000000B
-1	11111111B	11111110B	01111110B	10000001B
-2	11111110B	11111101B	01111101B	10000010B
⋮	⋮	⋮	⋮	⋮
-126	10000010B	10000001B	00000001B	11111110B
-127	10000001B	10000000B	00000000B	11111111B
-128	10000000B	无此值	无此值	无此值

4. 无符号数的运算

对两个无符号数进行补码加减法运算，无疑下边公式是正确的。

$$[X]_{\text{补}} + [\pm Y]_{\text{补}} = [X \pm Y]_{\text{补}} \quad (X, Y \text{ 及 } (X \pm Y) \text{ 都小于 } 2^n)$$

这就是说，两个无符号数进行加法，只要和的绝对值不超过整个字长，就不溢出，则和也一定为正数的补码形式，它等于和的原码；两个无符号数相减，可用减数变补与被减数相加来求得。减数变补，意即将整个减数变反加 1，或者说用模 2^n 减去减数，即

$$[Y]_{\text{变补}} = 2^n - Y$$

这样, $X + [-Y]_{\text{补}} = X + [Y]_{\text{变补}} = X + 2^n - Y = 2^n + (X - Y) = X - Y$, 由此得出如下两点结论:

(1) 若 $X \geq Y$, 则二者原码相减时无借位, 差值为正, $X + [Y]_{\text{变补}}$ 的和必大于 2^n , 最高位有进位, 得到的和为正数 $[X - Y]$ 的补码。它等于 $[X - Y]_{\text{原}}$ 。

(2) 若 $X < Y$, 则二者原码相减时有借位, 差值为负, 但 $X + [Y]_{\text{变补}}$ 的和必小于 2^n , 最高位无进位, 得到的和为负数 $[X - Y]$ 的补码。

在计算机中, 对两个无符号数进行补码减法运算时, 要判断结果是正数还是负数, 必须看减数变补与被减数相加时有无进位。有进位, 表示二数原码相减无借位, 结果为正; 无进位, 表示二数原码相减有借位, 结果为负。

例 1.1 用补码进行下列运算: I. $129 - 79$; II. $79 - 129$ 。

解 I. $X = 129, Y = 79$, 则

$$X = 10000001\text{B}$$

$$Y = 01001111\text{B}$$

$$[-Y]_{\text{补}} = [Y]_{\text{变补}} = 10110001\text{B}$$

$$\begin{array}{r} 10000001\text{B} \quad X \\ + 10110001\text{B} \quad [-Y]_{\text{补}} \\ \hline 100110010\text{B} \quad [X - Y]_{\text{补}} \end{array}$$

由于减数变补与被减数相加时, 最高位有进位, 表示原码相减时, 最高位无借位, 结果为正, 故

$$[X - Y]_{\text{原}} = [X - Y]_{\text{补}} = 00110010\text{B} = 50$$

II. $X = 79, Y = 129$, 则

$$X = 01001111\text{B}$$

$$Y = 10000001\text{B}$$

$$[-Y]_{\text{补}} = [Y]_{\text{变补}} = 01111111\text{B}$$

$$\begin{array}{r} 01001111\text{B} \quad X \\ + 01111111\text{B} \quad [-Y]_{\text{补}} \\ \hline 011001110\text{B} \quad [X - Y]_{\text{补}} \end{array}$$

由于减数变补与被减数相加时, 最高位无进位, 表示原码相减时, 最高位有借位, 结果为负, 故对 $[X - Y]_{\text{补}}$ 再求补得

$$[X - Y]_{\text{原}} = 100110010\text{B} = -50$$

综上所述, 不管参加运算的两个 n 位二进制数是带符号的补码形式, 还是无符号的数, 对计算机来说, 处理方法都是一样的。作加法时, 直接将二数相加即可; 作减法时, 是用减数变补与被减数相加来实现。只不过两种情况下, 结果的正负判别方法不同而已。若参加运算的两数为带符号的补码形式, 则结果的正负以最高位(符号位)来判别, 最高位为 0 是正数, 最高位为 1 是负数(以运算结果不溢出为条件); 若参加运算的两数为无符号的数, 则加法结果必为正数(以不溢出为条件), 没有符号位而减法结果的正负都应以最高位有无