

高等院校非计算机专业教材

微机应用基础

(第三版)

主 编

戴宗基 黄龙森

郭永青 雷于生



北京大学医学出版社

高等院校非计算机专业教材

微 机 应 用 基 础

(第三版)

主 编

戴宗基 黄龙森

郭永青 雷于生

编 委

(以姓氏笔画为序)

万奕	同济医科大学
邓洪志	同济医科大学
周宇菲	北京医科大学
赵铤	北京医科大学
胡加立	北京医科大学
胡彬	同济医科大学
郑凤	北京医科大学
高俊	同济医科大学
秦立轩	同济医科大学
郭永青	北京医科大学
黄龙森	同济医科大学
温厚津	北京医科大学
雷于生	同济医科大学
阙向红	同济医科大学
蓝顺碧	同济医科大学
魏民	北京医科大学
戴宗基	北京医科大学

北京大学医学出版社

WEI JI YING YONG JI CHU

图书在版编目 (CIP) 数据

微机应用基础/戴宗基主编. —3版. —北京: 北京大学医学出版社, 2000.3
ISBN 7-81071-029-X

I. 微... II. 戴... III. 微型计算机-基本知识
IV. TP36

中国版本图书馆 CIP 数据核字 (2000) 第 02721 号

微机应用基础

主 编: 戴宗基 黄龙森 郭永青 雷于生

出版发行: 北京大学医学出版社 (电话: 010-82802230)

地 址: (100083) 北京市海淀区学院路 38 号 北京大学医学部院内

网 址: <http://www.pumpress.com.cn>

E - mail: booksale@bjmu.edu.cn

印 刷: 莱芜市圣龙印务书刊有限责任公司

经 销: 新华书店

责任编辑: 暴海燕 责任校对: 齐 欣 责任印制: 郭桂兰

开 本: 787mm × 1092mm 1/16 印张: 28 字数: 707 千字

版 次: 2000 年 3 月第 3 版 2005 年 9 月第 10 次印刷 印数: 51001 - 54000 册

书 号: ISBN 7-81071-029-X/R · 029

定 价: 35.60 元

版权所有, 违者必究

(凡属质量问题请与本社发行部联系退换)

前言

计算机技术、网络技术、多媒体技术等飞速发展，不断地为信息处理技术的教学提供了日新月异的新内容。为了适应这一形势的发展，我们深感本书第二版的内容应当更新。于是我们编写了第三版。

本书第三版根据国家教委颁发的非计算机专业教学的基本要求和有关的教学大纲，对第二版进行了全面的改写。在基础知识中，我们增加了多媒体技术的介绍；对磁盘操作系统（DOS）不再独立成章，只对其最基本的常用命令在 Windows 98 中予以介绍；第二版中的 Windows 3.1 改为 Windows 98；Word、Excel 和 PowerPoint 均采用 Office 2000 版本；将原 WPS 一章改为 WPS 2000；原 FoxBase 改为 Visual FoxPro 6.0；删去了第二版中 Lotus 1-2-3；对于计算机网络专设了两章，全面介绍了计算机网络的基础知识和 Internet 及其应用。对于附录，充实了计算机常用英语，并增加了常见的各种文件的扩展名。这些内容不仅适合于高等医学院校，也适合于其他高等院校非计算机专业。

根据上述教学内容，我们建议在教学过程中，采用大屏幕投影仪讲课，用电子教案边讲边演示；同时增加上机操作，培养独立学习的能力。

本书共有十五章及二个附录。征得原著者和原参编单位的同意，参编人员及编写的分工均做了调整。其中戴宗基编写第一章的第一至第四节，温厚津编写第一章的第六、七节和第三节的第三部分，胡加立编写第一章的第五节和第六章，周宇菲编写第一章的第八节、第五章及附录一、二，郭永青编写第二章（其中第八节由赵铁编写），郑凤编写第三章，魏民编写第四章，邓洪志编写第七章和第八章的第一节，胡彬编写第八章的第二、三、六、七各节，黄龙森编写第八章的第四、五节和第九章的第三节，雷于生编写第九章的第一节和第十二章，秦立轩编写第九章第一节中关系数据模型部分及第二节和第三节的函数部分及第十章的第五节，高俊编写第十一章，万奕编写第十五章和第十一章第一节第二部分，阙向红编写第十三章和第十四章，蓝顺碧编写第十章的第一至第四节。第一至第六章由戴宗基、郭永青主审，第七至十五章由黄龙森、雷于生主审。全书由戴宗基统稿，郑凤进行编排。在编委会开会期间赵铁做了大量的工作。

本书在编写过程中，得到北京医科大学和同济医科大学领导和北京医科大学出版社的支持，才能顺利出版。在此表示诚挚的感谢。

由于水平有限，本书存在不少错误和不足，敬请读者批评指正。

编 者

1999 年 12 月

目 录

第一章 微型计算机概述

1.1 电子计算机的发展与应用	1
1.2 计算机中采用的计数制	3
1.3 计算机中的数据信息的表示	5
1.4 电子计算机的基本结构	12
1.5 计算机软件	15
1.6 微型计算机的硬件系统	19
1.7 多媒体技术和多媒体计算机	25
1.8 计算机病毒与安全	26

第二章 Windows 98 应用基础

2.1 Windows 98 的运行环境	33
2.2 Windows 98 启动	33
2.3 Windows 98 退出和重新启动计算机	34
2.4 Windows 98 的基本操作	35
2.5 Windows 98 的桌面简介	36
2.6 文件系统	48
2.7 磁盘的管理	55
2.8 中文输入法	58
2.9 Windows 98 的应用程序和附件	62
2.10 数据交换	67
2.11 MS-DOS 方式	68
2.12 安装和运行 DOS 应用程序	72
2.13 系统设置	73
2.14 Windows 98 的多媒体	83
2.15 Windows 98 网络使用	85

第三章 Word

3.1 Word 2000 简介	90
3.2 文档的创建与保存	93
3.3 文档的基本操作	96
3.4 文档的排版和打印	99
3.5 样式和模板	113
3.6 视图	115
3.7 表格	117
3.8 图形	120

3.9	插入符号和公式	128
3.10	域和宏的概念	130
3.11	网络功能	132
第四章	Excel 的使用	
4.1	Excel 2000 简介	136
4.2	工作表的建立	139
4.3	工作表的编辑和格式化	152
4.4	图表	156
4.5	打印	163
4.6	数据管理和分析	166
4.7	Word 表格文件与 Excel 文件之间的转换	172
4.8	宏	173
4.9	统计分析	176
第五章	PowerPoint	
5.1	PowerPoint 的基本操作	182
5.2	演示文稿的处理技巧	188
5.3	演示文稿的放映和打印	191
第六章	WPS 2000 文字处理系统	
6.1	WPS 2000 简介	202
6.2	文字处理	203
6.3	图像处理	206
6.4	绘制图形	208
6.5	表格	209
6.6	公式与图文符号库的使用	213
6.7	多媒体文档的制作与演示	215
6.8	WPS 2000 的实用工具	217
6.9	打印	218
第七章	计算机网络基础	
7.1	计算机网络的组成与分类	219
7.2	局域网传输介质与互连设备	221
7.3	网络拓扑结构	224
7.4	数据通信原理	225
7.5	网络体系结构和协议	228
7.6	网络交换技术	231
7.7	局域网操作系统	232
第八章	Internet 及其应用	
8.1	Internet 概述	237
8.2	万维网 (World Wide Web)	243
8.3	浏览器	246

8.4	Web 的搜索技术	258
8.5	电子邮件	263
8.6	文件传输	268
8.7	远程登录 Telnet 与 BBS	271
第九章	Visual FoxPro 数据库系统概论	
9.1	数据库的基本概念	276
9.2	Visual FoxPro 概述	282
9.3	VFP 6.0 程序设计语言语法基础	286
第十章	数据库的建立与基本操作	
10.1	数据表的创建	298
10.2	数据库的创建	304
10.3	数据库的浏览	308
10.4	表的修改、复制与统计	311
10.5	多个表的使用	323
第十一章	排序、索引和查询	
11.1	排序、索引与搜索	334
11.2	查询	341
11.3	视图	355
第十二章	程序设计基础	
12.1	程序的创建、编辑和运行	364
12.2	程序设计的基本结构	365
12.3	过程与自定义函数	374
第十三章	面向对象的程序设计	
13.1	类	377
13.2	对象	380
第十四章	表单设计	
14.1	创建表单	386
14.2	表单的修改、保存与执行	387
14.3	数据环境	387
14.4	添加控件	389
14.5	定制表单	393
14.6	定制对象	395
第十五章	菜单和报表设计	
15.1	菜单系统设计	406
15.2	报表文件设计	413
附录		
附录 1	计算机常用英语	428
附录 2	常见扩展名	434

第一章 微型计算机概述

电子计算机的出现是 20 世纪最重要的科学技术成就之一,是人类开始进入信息时代的重要标志。它已广泛应用于人类社会的各个领域,深入到科研、生产、文化、教育、卫生、通讯直到家庭生活之中,对人类社会的发展产生了极其深刻的影响。计算机科学已成为人类的第二文化,成为科学技术进步的基石。掌握计算机基础知识和应用技能,不仅是专业的需要,也是参与各种社会、经济、政治、科技等活动的需要。

1.1 电子计算机的发展与应用

电子计算机是一种能自动按程序执行各种操作的电子设备,也可以说它是一种能自动完成信息处理的电子设备。前者是从电子计算机的物理特征来定义的,后者是从电子计算机的功能角度来定义的。自动按程序执行各种操作,就是指自动完成信息的输入、存取、处理及输出的全过程。在当今信息化时代,计算机是信息自动化处理的最基本、最有效的工具。

1.1.1 电子计算机发展简史

计算机出现的时间不长,但人类创造计算工具、发展计算技术则有着悠久的历史。我国是最古老的计算工具一算筹、珠算盘的发明地。19 世纪,资本主义生产力的发展,第二次产业革命的实现,促进了计算技术的发展。19 世纪 50 年代,英国数学家乔治·布尔(George Boole)创立了逻辑代数,用二进制进行运算,是当前电子计算机的数学基础。1936 年英国科学家图灵(Alan Mathison Turing)首次提出逻辑机的通用模型—即“图灵机”,建立了算法理论,为电子计算机的出现提供了重要的理论根据,被称为计算机之父。1946 年 2 月世界第一台电子计算机在美国宾州大学诞生,它的全名是电子数值积分计算机(The Electronic Numerical Integrator and Calculator),缩写为 ENIAC。ENIAC 使用了 18000 个电子管,重约 30 吨,每秒钟能做 5 000 次加法运算,是机械计算机的 1 000 倍。但仍不完善,美籍匈牙利数学家冯·诺依曼(Jhon Von Neumann)于同年六月提出了顺序存储程序通用电子计算机的方案,这个方案的最重大的改进是存储程序的概念,从而奠定了计算机结构的基本框架。时至今日,计算机技术有了突飞猛进的发展,但其结构是诺依曼的结构。

1951 年 6 月 14 日第一台计算机作为商品交付使用,从此计算机从实验室走向社会,标志着人类进入计算机时代。

自 ENIAC 诞生以来,人们根据计算机硬件技术的发展,将其划分为四代:

第一代计算机是指 1946 年至 1958 年间的计算机。其主要特征是以电子管作为逻辑开关元件,尽管后期使用了磁鼓、磁芯做存储元件,但存储器的容量很小。软件上没有操作系统,只认机器语言编写的程序。

第二代计算机是指 1959 年至 1964 年间的计算机。其主要特点采用晶体管取代电子管做逻辑开关元件,因而,这一时期也称为晶体管计算机时代。除了使用磁芯做存储元件外,出现了磁盘、磁带用做辅助存储器。软件上先后产生了汇编语言和 FORTRAN、COBOL、ALGOL 等高级语言,提出了操作系统的概念。

第三代计算机是指 1965 年至 1971 年间的计算机。其主要特点是用集成电路取代晶体管,因而这一时期便称为集成电路计算机时代。这一时代的晚期,已使用了大规模集成电路。虽然仍用磁芯存储器做主存储器,但已开始使用半导体存储器,使存储器容量有很大的提高。操作系统已有很快的发展,分时系统也已使用,结构化、模块化的程序设计方法已出现。

第四代计算机是指 1972 年至今这一较长时期。其特点是采用超大规模集成电路,淘汰了磁芯存储器,采用半导体存储器,存储容量急速扩大。作为辅助存储器的软、硬盘容量有了大幅度提高,出现了光盘。这一时期中,微型计算机的出现有着特别重要的意义。它使计算机技术迅速普及,渗透到人类社会的每一领域,使过去由一台大型计算机带若干终端的集中化使用模式向可以人手一机、独立使用的分散化模式转变,导致了计算机应用的社会化和家庭化。同时,网络技术的发展,计算机技术与通讯技术的结合,个人电脑纷纷入网,实现了国际电脑互联网,做到了资源共享,特别是因特网(Internet)的出现,缩短了世界各地的距离,使大家紧密地联系在一起,不出家门,不仅能知天下事,还能办成大事,改变着人们的生活方式、工作方式,改变着世界科技和经济的面貌。另一方面,自 80 年代发展起来的多媒体技术,日臻成熟,已深入到寻常百姓家中。目前,计算机技术正在继续向微型化、网络化、巨型化、智能化的方向迅速发展。

从 80 年代开始,人们正在探索新一代计算机的研究。对新一代的计算机有各种各样的设想和方案:有的研究非诺依曼结构的计算机,有的研究有高度智能的、模拟人的大脑功能的计算机等等。可以相信,不久的将来新一代的计算机就会出现在我们的面前。

1.1.2 电子计算机的应用

电子计算机运算速度快且越来越快;运算精度高;存储信息的能力强且越来越强,因而有很强的记忆能力;不仅可作算术运算,也能作逻辑运算与逻辑判断;不仅能做数值计算,也能做各种非数值计算性质的处理(如文字处理、声音、图形、图像处理等);能在程序的控制下自动地进行工作,只要不断电,就能长期连续的不停工作等等。由于电子计算机有这一系列的工作特点,使其具有越来越广泛的应用领域。计算机的应用领域包括科学计算、信息处理、自动控制、人工智能、计算机辅助系统等方面。联网以后,电子计算机又成为信息的传播媒介,在信息高速公路、电子商务等方面形成了新的应用领域。

在医药卫生领域,电子计算机已进入了一切部门,可以说无处不在,形成了众多的新的边缘交叉学科,特别是医药信息学(Medical Informatics)。现在,国际上有医药信息学会(International Medical Informatics Association; IMIA)。我国也于 1981 年成立了中国医药信息学会(China Medical Informatics Association; CMIA),是 IMIA 的成员。在医药卫生领域,电子计算机的主要应用有以下几方面:医学数据处理,特别是各类的医学数据的统计分析;医学情报检索,当校园网、Internet 开通以后,医学情报的获取将会更快捷、更广泛;医学信号处理、图像处理及各种智能化仪器;医学专家系统;远程诊断;远

程教育；医院信息系统（Hospital Information System, HIS）；计算机辅助教学（Computer Assisted Instruction, CAI）在医学教育中更有特殊意义。

1.2 计算机中采用的计数制

计算机内各种信息都是用二进制代码来表示的。但是在应用中，还采用十进制和十六进制。这里对这些计数制做一简单介绍。

1.2.1 十进制数制（Decimal Number System）

十进制是我们日常所用的计数制。它的计数规律是：

使用 0, 1, 2, ..., 9 个数码；

计数规则是逢十进一；

它的位权是 10 的整数幂，即：

... $10^4 10^3 10^2 10^1 10^0 \cdot 10^{-1} 10^{-2} 10^{-3}$...

整个数字量的值是每个数码乘以位权后得的数值之和。

用十进制表示数字量时，其值的大小不仅与所用的数码有关，还与该数码在数字量中的位置有关。如 4 4 4，同是数码 4，由于在数字中位置不同，所表示的值不同。最左边的一位表示 4 百，第二位表示 4 拾，最右边的一位表示 4 个。这是因为它们所在位置的权重（Weight）即位权不同。位权是数码 1 在该位置所具有的值。故整个数字量的值为四百四十四。

1.2.2 二进制数制（Binary Number System）

与十进制相仿，它也有类似的规律：

使用 0、1 两个数码：其计数规则为逢二进一；

它的权是 2 的整数幂，即：

... $2^4 2^3 2^2 2^1 2^0 \cdot 2^{-1} 2^{-2} 2^{-3}$...

整个用二进制表示的数字量的值是每个数码乘以位权后得的数值之和。其和是十进制表示的值。

如： $(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (11)_{10}$

$(111.01)_2 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (7.25)_{10}$

二个二进制数相加时，遵守逢二进一的规则，如：

$(1011)_2 + (110)_2 = (10001)_2$

显然，与十进制相比，二进制数书写长，不好读，不好记。那么，计算机内部工作为什么采用二进制呢？原因有三：

1. 最基本的原因是，自然界中，电子元器件里，最容易找到表示两种稳定状态的物理元件，如电压的高低，脉冲的有无，开关的开闭等等，都可以用 0 或 1 来表示。然而要找到表示三个以上的稳定状态的物理元件是较难的。目前可以做到以表示二种状态的电子元件为基础，通过一定的组合来表示 0、1、... 9 等状态，但其结构十分复杂。

2. 二进制运算规则简单、方便。以加法为例，二进制加法公式只有 4 条：

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=10$$

然而十进制的加法公式有 100 条:

$$\begin{array}{ccccccc} 0+0 & 0+1 & 0+2 & \cdots & 0+9 \\ 1+0 & 1+1 & 1+2 & \cdots & 1+9 \\ \cdots & \cdots & \cdots & & \\ 9+0 & 9+1 & 9+2 & \cdots & 9+9 \end{array}$$

3. 数字量与逻辑量共存, 可以实现逻辑运算。

逻辑是一门推理和论证的科学, 一般用于处理命题的真伪。命题正确为真, 反之为假, 论断是唯一的, 不存在二义性。在计算机中, 可用一字符(或一串字符)来表示某一命题, 这字符(或字符串)称为逻辑变量。逻辑变量的特点是它的值要么为真(True), 要么为假(False)。计算机内, 逻辑真常用“1”来表示, 逻辑假常用“0”来表示。“真”(True, 1)与“假”(False, 0)也称为逻辑常量。显然, 逻辑量(包括逻辑变量与逻辑常量)的值是不能比大小的, 但逻辑量可以进行运算。最基本的逻辑运算有逻辑与、逻辑或、逻辑非三种, 并在此基础上可以完成多种复杂的逻辑运算。

1.2.3 二进制与十进制的转换

在介绍二进制时, 已经指出整个用二进制表示的一个数的值是每个数码乘以权重后得数值之和, 其和是十进制表示的值。这就是二进制数转换为十进制数的方法。

把一个十进制数转换为二进制数, 要分为整数与小数两部分进行。先介绍整数部分的转换方法—除 2 取余法。例如有十进制数 11, 转换为二进制的方法为:

2	11	...	余 1	2^0	低位
2	5	...	余 1	2^1	
2	2	...	余 0	2^2	
2	1	...	余 1	2^3	高位
	0				

$$\text{即 } (11)_{10} = (1011)_2$$

把十进制小数部分转换为二进制的方法是乘 2 取整法。

例如 将 $(0.375)_{10}$ 化为二进制数

解:	0.375		
	×) 2	...	0
	0.750	2^{-1}	高位
	×) 2	...	1
	1.500	2^{-2}	
	×) 2	...	1
	1.000	2^{-3}	低位

$$\text{即 } (0.375)_{10} = (0.011)_2$$

1.2.4 十六进制 (Hexadecimal Number System)

与十进制相仿, 也有以下的规律:

使用 0, 1, 2, ..., A, B, C, D, E, F 十六个数码;

其计数规则为逢 16 进一;

它的位权是 16 的整数幂, 即

$$\dots 16^4 16^3 16^2 16^1 16^0.16^{-1} 16^{-2} 16^{-3} \dots$$

整个用十六进制表示的数字量的值是每个数码乘以位权后得的数值之和，其和是十进制表示的值。

例如： $(1A)_{16} = 1 \times 16^1 + 10 \times 16^0 = (26)_{10}$

$$(ABF)_{16} = 10 \times 16^2 + 11 \times 16^1 + 15 \times 16^0 = (2751)_{10}$$

十进制数转换为十六进制数时，其整数部分用除 16 取余法，小数部分用乘 16 取整法。但需将 10~15 各数相应换为 A~F。

十六进制数与其它进制数间的转换，通过二进制较方便。我们先看表 1.1。

由表可知，四位二进制数相应于一位十六进制数。因此，二进制数整数部分转换为十六进制数时，只需从二进制数小数点开始往左，每四位为一组，与一位十六进制数相对应，最后若不足四位，可在其左端用 0 补齐。二进制数的小数部分转换为十六进制数时则以小数点开始往右，每四位为一组与一位十六进制数相对应，最后若不足四位，在其右端用 0 补齐。

表 1.1

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

例如： $(1011\ 0011\ 1111\ 1001)_2 = (B3F9)_{16}$

$$(10\ 1111\ 0000\ 1100)_2 = (2F0C)_{16}$$

$$(1FFF)_{16} = (0001\ 1111\ 1111\ 1111)_2$$

在有的机器中，十六进制数用数后加 H 表示，如 1FFFH 就是 $(1FFF)_{16}$ 。

1.3 计算机中的数据信息的表示

1.3.1 计算机中的基本运算

前面已经介绍过，在计算机中采用二进制数进行运算。数学理论可以证明，任何数学运算最终可以用加法和移位这两种基本操作来完成。对于加法，根据二进制的原则，逢二进一，很容易处理。对于减法，可以用加一个负数来表示，依然是作加法运算。对于乘除法，则用加法和移位操作来实现。

1.3.1.1 二进制的乘法和除法

二进制的乘除法规则与十进制的乘除法规则是一致的。

例如乘法： 1110×101 和除法： $11100001/101$

$$\begin{array}{r}
 \times \quad 1110 \\
 \quad 101 \\
 \hline
 1110 \\
 0000 \\
 1110 \\
 \hline
 1000110
 \end{array}$$

$$\begin{array}{r}
 101101 \\
 101 \overline{) 11100001} \\
 \underline{101} \\
 1000 \\
 \underline{101} \\
 110 \\
 \underline{101} \\
 101 \\
 \underline{101} \\
 0
 \end{array}$$

可以看出，乘法可以通过将被乘数左移位然后用加法相加来实现。除法可以通过除数右移和减除数两种运算来实现，在计算机中，再将减法运算转化为加法运算。于是，除法运算能通过右移位与加法两种操作来实现。

1.3.1.2 二进制数的定点与浮点表示法

二进制数 101.1 和 10.11 都是由 1011 四个数码组成，其差别是小数点所在的位置不同，当然其值也不同。那么，在计算机中，小数点怎样表示的呢？通常有两种方法，即定点表示法和浮点表示法。

①定点表示法就是在计算机内，数的小数点的位置是固定的。通常都把小数点固定在最高数据位的左边，在小数点前再设一位符号位，即

$$\begin{array}{c}
 \text{符号位} \quad b_1 b_2 b_3 \cdots b_m \\
 \uparrow \\
 \text{想象的小数点位置}
 \end{array}$$

对于符号位通常用 0 表示正号，用 1 表示负号。小数点后面是 m 个二进制位表示的该小数的值。小数点不用明确表示出来，因为它总是定在符号位与最高数值位之间。这种表示法使计算机所能表示的数总小于 1 。然而实际问题中，数可能大于 1 。这就需要在解决问题之前，选择一个适当的比例因子，使全部参与运算的数的绝对值都小于 1 ，并保证运算的中间结果和最终结果的绝对值均小于 1 。而在输出结果时，再按相应的比例因子对该数扩大。

用定点表示法所能表示的小数数值范围很小，对用 $m+1$ 个二进制位表示的小数来说，其值范围为：

$$|N| \leq 1-2^{-m}$$

这里要想到还有符号位。比如 7 位二进制 ($0000000 \sim 0111111$) 表示的数值范围为：

$$|N| \leq 1-2^{-6}$$

对于定点整数的表示法，除符号位仍安排在最高位外，其小数点可以设想是定在数值最低位的右面，即

$$\begin{array}{c}
 \text{符号位} \quad b_n b_{n-1} \cdots b_1 b_0 \\
 \uparrow \\
 \text{设想小数点在此}
 \end{array}$$

此时, 这个用 $n+1$ 位二进制表示的数值范围为:

$$|N| \leq 2^n - 1$$

例如 对于16位二进制数(0000000000000000 ~ 0111111111111111)的正整数, 其值 N 的范围在0~32 767之间。这里 b_{n+1} 是符号位, 其值为0, 表示正数。

②浮点表示法: 是指小数点在数据中的位置不是固定的, 而是可以左右移动的。它通常以指数形式表示为:

$$N = M \cdot R^E$$

式中 M (Mantissa)称为浮点数的尾数, R (Radix)称为基数, E (Exponent)称为阶码。 R 在计算机中通常取值为2、8或16, 总之是一个常数。如 R 的值定为2, 它不需要在浮点数中明确表示出来。尾数 M 通常是以定点小数形式表示的, 它的位数决定了此浮点数的精度。阶码 E 通常是用整数形式表示的, 阶码的值确定了小数点在数据中的位置, 也就决定了此浮点数的表示范围。浮点数也要加上符号位。小数点则想象在尾数的最高位的左面, 即尾数被规定为纯小数形式。在 IEEE 标准中, 浮点数的表示形式为:



在 Intel 公司的 80486 微处理器中, 支持三种浮点数据类型, 即单精度、双精度和扩展精度实数。

单精度实数有 1 位尾数符号位, 8 位阶码, 23 位尾数, 共 32 位。相对于十进制数, 约有 7 位有效数字, 可表示数的范围约为 $\pm 1.7 \times 10^{\pm 38}$ 。

双精度实数有 1 位符号位, 11 位阶码, 52 位尾数, 共 64 位。相对于十进制数, 约有 16 位有效数字, 可表示数的范围约为 $10^{\pm 308}$ 。

扩展精度实数有 1 位符号位, 15 位阶码, 64 位尾数, 共 80 位。相对于十进制数, 约有 20 位有效数字, 可表示的范围为 $10^{\pm 4932}$ 。

1.3.1.3 二进制数的减法

二进制减法运算可以改变为加法运算, 如何实现呢? 在计算机里是这样进行的: 被减数为正时, 用原码表示, 减数用补码表示, 用被减数的原码与减数的补码相加, 就实现了减法运算。我们先介绍一下原码、反码与补码。

把一个十进制数用二进制数来表示时, 该数的二进制码就叫原码, 其最高位为符号位。如果对原码逐位取反 (即 0 变 1, 1 变 0) 但符号位不变, 这样的二进制码就叫反码。反码加 1 就得补码。我们规定, 对于正数, 它的反码与补码就是原码。比如 +5, 其原码、反码和补码的表示形式都是 00000101。这里采用的是 8 位二进制。对于负数, 这三种码的表示形式是不同的, 如 -5:

原码	10000101
反码	11111010
补码	11111011

如果用定点数运算 $7-5$, 则有

	0.0000111	(7 的原码)
+	1.1111011	(-5 的补码)
	10.0000010	
丢掉	1	

式中小数点是想象的，有了小数点就相当于把 7 和 5 的二进制码往右移了 7 位，即除了一个比例因子 2^7 在计算机内，符号位是最高位，它的进位是没有设备来保存的，所以被丢掉。相减的实际结果，表示是一个正数，再乘以比例因子 2^7 ，所得结果为二进制的 10，即十进制的 2。

如果是 5-7，其运算过程是：

$$\begin{array}{r}
 0.0000101 \quad (5 \text{ 的原码}) \\
 + 1.1111001 \quad (7 \text{ 的补码}) \\
 \hline
 1.1111110 \quad (-2 \text{ 的补码})
 \end{array}$$

结果没有进位，符号位为 1，表明运算结果是一负数。负数是以补码形式表示的，因此求其原码：补码减 1 再求反，即得 1.0000010，是十进制 -2 的原码。当然，这个过程已经把比例因子的乘除考虑进去了。

实际上，在机器中从补码求原码，也不用减法，仍用求反加 1（即再取一次补）来实现：

$$1.1111110 \xrightarrow{\text{取反}} 1.0000001 \xrightarrow{\text{末位加 1}} 1.0000010$$

综上所述，机器中二进制的减法运算器规律是：正数用原码表示，负数用补码表示；两数相加时符号位参与运算；相加结果，符号位有进位时，表示其结果是正数，进位被丢掉，结果是原码；若符号位无进位，表示运算结果是负数，此时需对结果再进行求补，便可得运算结果的原码。

1.3.1.4 逻辑运算

前面已经指出，在计算机中可以完成逻辑与、逻辑或和逻辑非三种基本逻辑运算，并在此基础上可以完成多种复杂的逻辑运算。

① 逻辑与（又称逻辑乘）：如果有两个逻辑变量 A 与 B，它们的“与”运算表达式：

$$A \wedge B \quad \text{或} \quad A \cdot B$$

式中“ $\wedge$ ”和“ $\cdot$ ”为运算符。这种运算的规则是：

$$0 \wedge 0 = 0 \quad 0 \wedge 1 = 0 \quad 1 \wedge 0 = 0 \quad 1 \wedge 1 = 1$$

即两个逻辑变量做与运算时，只要任何一个变量为 0，则结果为 0，当且仅当二个变量均为 1 时，其结果才为 1。概括地说就是任 0 则 0，全 1 则 1。

在计算机应用中，常用 AND 表示“与”运算符，其运算结果常表示为 T 或 F。

② 逻辑或（又称逻辑加）：其运算符为“ $\vee$ ”或“+”，表达式为

$$A \vee B \quad \text{或} \quad A + B$$

“或”运算的规则是：

$$0 \vee 0 = 0 \quad 1 \vee 0 = 1 \quad 0 \vee 1 = 1 \quad 1 \vee 1 = 1$$

即任 1 则 1，全 0 则 0。请注意，这里不存在进位关系。

在计算机应用中，常用 OR 表示“或”运算符，其运算结果常表示为 T 或 F。

③ 逻辑非（也称非运算）：它是对逻辑变量取值的否定，如果逻辑变量 A 的值 1，那么对 A 进行非运算的结果为 0，可用下述方法表示：

逻辑变量 A 上方的一横，就表示非运算。非运算符也可用 NOT 表示，运算的结果也常用 T 或 F 表示。

④复杂的逻辑运算：由上述三种基本逻辑运算，可以完成复杂的逻辑运算，各种复杂的逻辑运算，遵循布尔代数的法则与定理。例如：

$\overline{A \wedge B}$	“与非”运算，先做与运算再做非运算
$\overline{A \vee B}$	“或非”运算，先做或运算再做非运算
$(A \wedge \overline{B}) \vee (\overline{A} \wedge B)$	“异或”运算，可记做 $A \oplus B$ 。

1.3.2 符号数据的表示

符号数据通常是指字符、字符串、图形符号、汉字等等。如英文大小写字母、数字符号（0 到 9、一到十等）、各种专用符号（+、-、*、/、\$、…）。这种数据没有数值大小的含义，也称非数值数据,但它们却是使用计算机时不可少的重要组成部分。比如各种操作命令都离不开这些符号来表示。

计算机中的字符，一般是用八个二进制位的编码来表示。编码有多种，用得最普遍的是 ASCII 码，即美国标准信息交换代码（American Standard Code for Information Interchange, ASCII）。

表 1.2 ASCII 字符编码表

B ₆ b ₅ b ₄				000	001	010	011	100	101	110	111
b ₃	b ₂	b ₁	b ₀								
0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	"	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	'	7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	R	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	
1	1	0	1	CR	GS	-	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	↑	n	~
1	1	1	1	SI	US	/	?	O	—	o	DEL

在 ASCII 码中，是用八个二进制位的编码来表示一个字符或控制符，其最高位为 0，因此，实际上用的是七位二进制码，即 $b_7b_6b_5b_4b_3b_2b_1b_0$ ， $b_7=0$ 。它可以表示 128 个字符。表 1. 2 是 ASCII 码和 128 个字符间的对应关系表。上横栏为 ASCII 码的高四位（ $b_7b_6b_5b_4$ ，因 $b_7=0$ ，只写了三位），左竖栏为低四位（ $b_3b_2b_1b_0$ ）。这样分，可以用十六进制码来表示，如从表中查出数码 0 的 ASCII 码为 00110000，用 16 进制则为 (30)₁₆。当然也可用十进制表示，数码 0 的 ASCII 为 (48)₁₀。ASCII 编码中表示每个字符的二进制码的值也称 ASCII 码值，此码值用十进制或十六进制表示较二进制更方便更好记。要注意的是 ASCII 码仅表示字符，没有数值的意思。例如数码 1 的 ASCII 码值为 (49)₁₀ 或 (31)₁₆，它都不是数码 1 的值。

如果 ASCII 码的最高位为 1($b_7=1$)时, ASCII 码可表示 256 个字符。前 128 个字符与上述表中列举的字符与编码的对应关系完全一样,而后 128 个字符及其对应的编码关系,称为扩展的 ASCII 码。扩展的 ASCII 码往往被许多国家用来定义为本国的文字代码。

1.3.3 汉字编码

对英文及数字信息进行数字化编码是比较容易的,因为这些信息本身是有序的,对汉字信息进行数字化编码,远比西文要难得多。因为首先需要确定汉字的排列顺序。汉字的排序方法有许多种,例如可以按发音排列,也可以按偏旁部首及笔划多少排列等等。更突出的是汉字不是拼音文字,且数量很多,字形复杂,无论从读音、形状、定义等方面看,都缺乏有机的联系。此外,如何把计算机只能处理的二进制编码转换成各种各样的汉字,甚至打印成包括繁体字、简体字以及诸如宋体、楷体、隶书、行楷等不同类型的字来,就进一步增加了利用计算机进行信息处理的难度。不过,通过我国专业工作者的努力,提出了很多汉字编码方案,成功地解决了这一系列问题。其中,人们经常接触的有如下几种编码:国标码、国标区位码、机内码、输入码、汉字输出码。

1.3.3.1 国标码(交换码)与国标区位码

国标码是根据国家标准 GB2312《信息交换用汉字编码字符集—基本集》所编的汉字编码。该编码相当于国际通用的 ASCII 字符集,是用于汉字信息系统中或通信系统中进行汉字信息交换的代码。它把该代码表分成 1 到 94 个区,每个区又分成 94 位。任何汉字或符号均从它所在的区、位号来区别,因此,它是唯一的,每个字符对应一个编码。每个汉字、字符、表格符都用 2 个字节表示,每字节的最高位为 0,其余 7 位为代码。国标码又称汉字交换码,共有汉字及符号 7445 个,其中:汉字 6763 个,分为两级,其中一级字库从第 16 区到 55 区,包含了最常用字、高频字 3755 个。二级字库从 56 区到 87 区,包含了次常用汉字 3008 个。

国标区位码是国标码的一种变形,它与国标码的区别主要有两点:

(1) 每个汉字或符号的国标码以四个 16 进制数表示,例如汉字的“啊”的编码为 3021H,其中前两位表示区码,后两位表示位码。而汉字的国标区位码是以四个十进制数表示,例如“啊”字的对应码为(1601)₁₀。同样,前两位表示区码,后两位表示位码。

(2) 国标区位码中没有包括 ASCII 码字符集中的 32 个控制字符,而国标码则包括了这 32 个字符,而且将其排在了所有汉字与字符的前面。正是因为国标区位码与国标码有着这样明显的区别,因此,两者在计算机进行汉字信息处理时的作用也是完全不同的:国标码用于计算机之间的或计算机与终端之间的信息交换,属于信息交换码;而国标区位码可以作为汉字的输入编码,即可采用这种编码输入汉字。

国标区位码和国标码之间具有明显的对应关系,可由一种码经过换算得到另一种码。它们的换算规则是:用国标区位码的 16 进制数,加上未包括在内的 32 个 ASCII 码控制字符的序号数,即为所对应的国标码。例如:汉字“啊”的国标区位码为 1601,其 16 进制数为 1001H,按区码和位码分别加以序数 32 的 16 进制数“20H”,即为该字的国标码。其过程如下式所示: