微电脑学习与应用丛书·翁瑞琪 主编

微电脑应用基础

侯紫达 李俊旺 曹焕林 阴仲朋 编著

WEIDIANNAO XUEXI YU YINGYONG CONGSHU

国防工业出版社
GFGYCBS

微电脑应用基础

侯紫达　李俊旺
曹焕林　阴仲朋　编著

国防工业出版社
·北京·

图书在版编目(CIP)数据

微电脑应用基础/翁瑞琪主编．—北京：国防工业出版社，1995.8
（微电脑学习与应用丛书）
ISBN 7-118-01428-1

Ⅰ.微… Ⅱ.翁… Ⅲ.微型计算机-计算机应用-基本知识 Ⅳ.TP36

中国版本图书馆 CIP 数据核字(95)第 01859 号

国防工业出版社出版发行
（北京市海淀区紫竹院南路 23 号）
（邮政编码 100044）
北京怀柔新华印刷厂印刷
新华书店经售
*
开本 787×1092 1/16 印张 20½ 471 千字
1995 年 8 月第 1 版 1995 年 8 月北京第 1 次印刷
印数：1—7000 册 定价：19.50 元

丛书总序

当今人类正步入信息化时代，人们所面临的信息处理量之大、信息处理复杂程度之高，以致单靠人脑去处置已不能胜任。因此，人类迫切要求辅助脑力劳动的工具。电子计算机以其在信息处理上的独特优点而充当了人类脑力劳动的辅助工具，或者说，它起到了人类脑力活动在体外延长之作用。因此，人们美称电子计算机为电脑。

微型计算机（微电脑）的出现使计算机的应用得到极大普及。至今，微电脑已成为人类活动中不可缺少的有力助手。学习和应用微电脑是当今必然的趋势。为适应当今时代普及微电脑应用的需要，决定编写《微电脑学习与应用丛书》。

本丛书以教会如何使用微电脑为目的，帮助读者步入微电脑应用世界。

本丛书可供具有高中以上文化程度的微电脑使用者和广大爱好者自学，可用作微电脑应用短训班的培训教材，也可供大专院校广大学生学习计算机应用时参考。

本丛书由天津大学技术经济与系统工程系翁瑞琪教授主编，参加本丛书编写的是长期从事计算机教育和计算机科研工作的教学经验丰富、实践能力强的教师和专家。

本丛书的编辑出版得到国防工业出版社的大力支持，在此表示衷心的感谢。

期望本丛书的出版能为我国计算机应用范围的扩大和应用水平的提高起到促进作用。

热诚欢迎有关专家和广大读者对本丛书的编辑出版提出建设性的建议和改进意见。

翁瑞琪

前　言

本书是翁瑞琪教授主编的《微电脑学习与应用丛书》之一。

要掌握计算机的应用，首先要学好计算机及其应用方面的基础。因此，计算机应用基础是高等学校任一专业学生在计算机学习方面的必修基础课。

《微电脑应用基础》可用作计算机应用基础课程的学习教材，也可供微型机用户自学或用作相应的计算机培训班的培训教材。

本书分十三章，分为基础篇、数据库篇和应用篇三个部分。

基础篇共3章（包括第一章到第三章）。分别讲述计算机基础知识、磁盘操作系统及常用命令、文字编辑方面的内容。

数据库篇共5章（包括第四章到第八章）。以当前使用普遍的关系数据库管理系统FoxBASE＋为典型进行介绍，讲述了FoxBASE＋基础、数据库的基本操作、FoxBASE＋函数、FoxBASE＋的程序设计、FoxBASE＋的输入和输出方面的内容。

应用篇共5章（包括第九章到第十三章）。分别讲述了PCTOOLS、文件压缩实用程序、NDD、ADM等实用软件，并对计算机病毒及其防治作了简介。

参加本书编写的有侯紫达、李俊旺、曹焕林和阴仲朋。第一章和第三章由曹焕林执笔。第二章由李俊旺执笔。第四章到第八章和附录由侯紫达执笔。第九章到第十三章由阴仲朋执笔。全书由翁瑞琪审定和统一定稿。

由于编者水平所限，错误在所难免，欢迎批评指正。

内容简介

《微电脑学习与应用》丛书由天津大学翁瑞琪教授主编。本丛书以教会如何使用微电脑为目的，帮助读者步入微电脑应用世界。

本书是该丛书之一，侧重讲解微电脑应用的基本知识。全书共十三章。第一至三章介绍计算机基本知识、磁盘操作系统及常用命令、文字编辑等内容。第四至八章介绍 FoxBASE＋的基础、基本操作、函数、程序设计及输入输出方面的内容。第九至十三章介绍了 PCTOOLS、文件压缩实用程序、NDD、ADM 及计算机病毒等。

本丛书可供具有高中以上文化程度的微电脑使用者和广大爱好者自学，又可作微电脑应用短训班的培训教材，也可供大专院校广大学生参考。

目　录

第一章　计算机基础知识

本章主要介绍电子计算机的发展和应用；数制的表示及转换；微型计算机的构成。

1.1　电子计算机的发展和应用

1.1.1　电子计算机的发展概况

1946 年由美国宾夕法尼亚大学的穆尔学院研制成的“埃尼阿克”(ENIAC)是世界上第一台电子数字计算机。

电子计算机大致经历了三代的演变，目前正全面向四代机过渡，多年前已开始研制第五代机。

第一代(1946 年～1957 年)是电子管计算机。这个时期的计算机采用电子管作逻辑元件，内存储器采用磁性元件，外存储器开始采用磁带。编程采用机器语言，以后采用汇编语言。这代机器的特点是体积大、耗电量大、运算速度低、可靠性差、内存储器容量小。

第二代(1958 年～1963 年)是晶体管计算机。在这个阶段计算机的速度有所提高，体积、功耗已大大减小，可靠性和内存储器容量也有较大提高。这代机器应用于各种事务数据处理，并开始用于工业控制。其特点是体积小、耗电少、可靠性和内存容量有较大的提高。

第三代(1971 年之后)是大规模集成电路计算机。计算机的逻辑元件采用大规模集成电路，具有图形功能的高清晰度的彩色显示器得到广泛应用。运算速度和可靠性又有了很大提高，功能更加完善。计算机开始向巨型机、微型机、网络和智能机多样化的方向发展。

1.1.2　电子计算机的特点

①运算速度快。目前第四代计算机的速度已达每秒几亿次，有的达数十亿次。

②精度高。计算机的每个字所包含的位数称为字长，它反映计算机的计算精度。微型机字长达 32 位，并且可双倍字长或多倍字长运算。

③有“记忆”和“逻辑判定”的功能。它能把数据、程序存入存储器然后进行处理、计算并把结果保存起来。这也是计算机区别于其他计算工具的主要特征。

④能自动进行控制。计算机是程序控制的自动化操作。

1.1.3　电子计算机的主要用途

①用于科学计算。电子计算机是发展尖端科学技术必不可少的重要工具，为物理、化学、力学、数学等基础学科的科学家们提供了重要的计算工具。

②用于数据处理和信息加工。对大量的数据进行综合分析，是指对在科学研究、生产实践、经济活动、甚至日常生活领域中所获得的大量信息，进行归纳、整理、分类、统计加工，绘出数据分布曲线或打印出报表。

③用于自动控制。特别是用于工业生产过程的自动控制。

④用于计算机辅助设计(CAD)和计算机辅助教学(CAI)等。

⑤用于人工智能方面的研究和应用。

1.2 数制和编码系统

1.2.1 数制

按进位的方法进行计数，称为进位计数制。在日常生活中，我们最熟悉的是十进制数，还有十二进制、十六进制、二十四进制、六十进制。

1. 十进制数制

①它有十个不同的数字符号(又称数码)，即 0、1、2、3、4、5、6、7、8、9。

②它是逢"十"进位的。同一数码处于不同的数位(称为"权")代表的意义是不同的。例如 555.55 这个数中，小数点左边的第一位"5"代表个位，其值为 5×10^0；左边的第二位"5"代表十位，其值为 5×10^1；左边的第三位代表百位，其值为 5×10^2；小数点右边的第一位"5"，其值为 5×10^{-1}；右边的第二位"5"，其值为 5×10^{-2}。因此这个数可写成：

$$555.55=5\times10^2+5\times10^1+5\times10^0+5\times10^{-1}+5\times10^{-2}$$

一般地说，任意一个十进制数 N 都可以表示为：

$$\begin{aligned}N=&\pm(K_{n-1}\times10^{n-1}+K_{n-2}\times10^{n-2}+\cdots+K_1\times10^1+K_0\times10^0\\&+K_{-1}\times10^{-1}+K_{-2}\times10^{-2}+\cdots+K_{-m}\times10^{-m})\\=&\pm\sum_{i=-m}^{n-1}(K_i\times10^i)\end{aligned}$$

式中：m、n 均为正数；n 为小数点左边的位数，m 为小数点右边的位数；K_i 表示第 i 位的数码，可以是 0～9 十个数字符号中的任一个，由具体的 N 数来确定；而"10"为十进制数的基数，10^{n-1}、…、10^1、10^0、10^{-1}、…、10^{-m}称为十进制数的"权"。

2. 二进制数制

①它只有两个不同的数码，即 0 和 1。

②它是逢"二"进位的。

$$\begin{aligned}(1111.11)_2&=1\times2^3+1\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}\\&=8+4+2+1+0.5+0.25\\&=(15.75)_{10}\end{aligned}$$

一般地说，任意一个二进制数 N，都可以表示为：

$$\begin{aligned}N=&\pm(K_{n-1}\times2^{n-1}+K_{n-2}\times2^{n-2}+\cdots+K_1\times2^1+K_0\times2^0\\&+K_{-1}\times2^{-1}+K_{-2}\times2^{-2}+\cdots+K_{-m}\times2^{-m})\\=&\pm\sum_{i=-m}^{n-1}(K_i\times2^i)\end{aligned}$$

式中：K_i 只取"1"或"0"，由具体的 N 确定。m、n 为正整数，"2"是二进制的基数。

对于任意进位计数制，基数可用正整数 R 来表示，这时，数 N 可表示为：

$$N=\pm\sum_{i=-m}^{n-1}K_iR^i$$

式中 m、n 均为正整数；K_i 则是 0 至(R－1)中的任何一个；R 是基数，采用“逢 R 进一”的原则进行计数。

计算机中常用的几种进位计数制表示数的方法列于表 1－1。

表 1－1　常用计数制表示数的方法

十　进　制	二　进　制	八　进　制	十　六　进　制
0	00000	00	00
1	00001	01	01
2	00010	02	02
3	00011	03	03
4	00100	04	04
5	00101	05	05
6	00110	06	06
7	00111	07	07
8	01000	10	08
9	01001	11	09
10	01010	12	0A
11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10

1.2.2　不同进位计数制之间的转换

1. 十进制数与二进制数之间的转换

(1)十进制整数转换成二进制整数

十进制整数转换成二进制整数，通常采用除 2 取余法。所谓除 2 取余法，就是将已知十进制数反复除以 2，相除之后余数为 1，则对应于二进制数的相应位为 1；余数为 0，则相应位为 0。第一次除法得到的余数是二进制的最低位，最后一次余数是二进制数的最高位。从低位到高位逐次进行，直到商为 0 为止。如最后一次除法所得的余数为 K_{n-1}，则排列的顺序是 $K_{n-1}K_{n-2}\cdots K_1K_0$，即为所求之二进制数。

例如，将$(725)_{10}$转换成二进制数，其全过程可表示如下：

```
2 | 7 2 5
2 | 3 6 2   …余数为 1，K1＝1
2 | 1 8 1   …余数为 0，K2＝0
2 |   9 0   …余数为 1，K3＝1
2 |   4 5   …余数为 0，K4＝0
2 |   2 2   …余数为 1，K5＝1
2 |   1 1   …余数为 0，K6＝0
2 |     5   …余数为 1，K7＝1
2 |     2   …余数为 1，K8＝1
2 |     1   …余数为 0，K9＝0
  |     0   …余数为 1，K10＝1
```

所以　$(725)_{10}=(K_{10}K_9\cdots K_2K_1)=(1011010101)_2$

(2)十进制纯小数转换成二进制纯小数

十进制纯小数转换成二进制纯小数，通常采用乘2取整法。所谓乘2取整法，就是将已知十进制纯小数反复乘以2，每次乘以2之后，所得新数的整数部分为1，则相应位为1；整数部分为0，则相应位为0。从高位向低位逐次进行，直到满足精度要求或乘2之后的小数部分是0为止。最后一次乘2所得的整数部分为K_{-m}。转换后，所得的纯二进制小数为：

$0.K_{-1}K_{-2}\cdots K_{-m}$

例如，将$(0.6871)_{10}$转换成纯二进制小数，转换过程如下所示：

$$
\begin{array}{rl}
0.6871 & \\
\times)\quad 2 & \\
\hline
1.3752 & \cdots\text{整数部分为 } 1,K_{-1}=1 \\
\times)\quad 2 & \\
\hline
0.7504 & \cdots\text{整数部分为 } 0,K_{-2}=0 \\
\times)\quad 2 & \\
\hline
1.5008 & \cdots\text{整数部分为 } 1,K_{-3}=1 \\
\times)\quad 2 & \\
\hline
1.0016 & \cdots\text{整数部分为 } 1,K_{-4}=1 \\
\times)\quad 2 & \\
\hline
0.0032 & \cdots\text{整数部分为 } 0,K_{-5}=0
\end{array}
$$

所以 $(0.6871)_{10}\approx(0.10110)_2$

可见，十进制纯小数不一定都能转换成完全等值的二进制纯小数。遇此情况时，根据精度要求，取近似值。

混合小数由整数和小数复合而成，只要按上述方法分别进行转换，然后将转换结果组合起来即可。

(3)二进制数转换为十进制数

将二进制数转换成十进制数，只要将二进制数用计数制通用形式表示出来，计算结果，便得到相应的十进制数。

例如：$(101011.1001)_2=1\times2^5+1\times2^3+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-4}$

$=(43.5625)_{10}$

2. 二进制数与八进制数之间的转换

(1)二进制数转换成八进制数

二进制整数转换为八进制数时，可以从最低位(小数点左第一位)开始，每三位分为一组，不够三位的补足三位，每三位二进制数用一位八进制数表示。

例如，把二进制数11111101.01001111转换成八进制数。

$(11111101.01001111)_2=(375.236)_8$

可以看出，对于混合小数，从小数点分界，整数部分与小数部分分别进行转换，便可写出相应的八进制数。

(2)八进制数转换成二进制数

八进制数转换成二进制数，只要将每位八进制数用相应的三位二进制数代替即可完

成转换。

例如，把八进制数$(745.513)_8$转换成二进制数。

$$\begin{array}{cccccccc} (& \overbrace{7} & \overbrace{4} & \overbrace{5} & \cdot & \overbrace{5} & \overbrace{1} & \overbrace{3} &)_8 \\ & \overbrace{111} & \overbrace{100} & \overbrace{101} & \cdot & \overbrace{101} & \overbrace{001} & \overbrace{011} & \end{array}$$

即　$(745.513)_8=(111100101.101001011)_2$

3. 二进制数与十六进制数之间的转换

(1)二进制数转换成十六进制数

仿照二进制数与八进制数的转换方法，很容易得到二进制数与十六进制数之间的转换方法。

对于二进制小数，从小数位置开始，整数部分向左，小数部分向右，每四位二进制数用一位十六进制数代替，不足四位时整数部分在左面补0，小数部分在右面补0，即可完成转换。

例如，把$(101101101.0100101)_2$转换成十六进制数，则

$$\begin{array}{cccccccc} & \overbrace{0001} & \overbrace{0110} & \overbrace{1101} & \cdot & \overbrace{0100} & \overbrace{1010} &)_2 \\ (& \overbrace{1} & \overbrace{6} & \overbrace{D} & . & \overbrace{4} & \overbrace{A} &)_{16} \end{array}$$

所以$(101101101.01001010)_2=(16D.4A)_{16}$

(2)十六进制数转换成二进制数

将一个十六进制数转换成二进制数，只要将每一位十六进制数用四位相应的二进制数表示即可完成转换。

例如，将$(1A74.3AB)_{16}$，转换成二进制数，则：

$$\begin{array}{cccccccc} (& \overbrace{1} & \overbrace{A} & \overbrace{7} & \overbrace{4} & \cdot & \overbrace{3} & \overbrace{A} & \overbrace{B} &)_{16} \\ & \overbrace{0001} & \overbrace{1010} & \overbrace{0111} & \overbrace{0100} & \cdot & \overbrace{0011} & \overbrace{1010} & \overbrace{1011} & \end{array}$$

所以$(1A74.3AB)_{16}=(1101001110100.001110101011)_2$

1.2.3　计算机中数的表示方法

在计算机中，一个数的小数点通常有两种表示方法：定点表示法和浮点表示法。采用定点表示法的计算机叫定点机；采用浮点表示法的计算机称浮点机。

所谓定点和浮点，是指计算机中数的小数点位置是固定的还是浮动的。

1. 定点表示法

在定点表示法中，小数点位置是固定不动的。通常把小数点固定在数值部分的最高位之前，或是把小数点固定在数值部分的最后面。前者将数表示为纯小数，后者将数表示成整数。因此，定点数在计算机中有两种表示方法：

纯小数表示方法 数符 | . 尾数

整数表示方法 数符 | 尾数 .

其中，数符用来表示数的正负，正数用"0"表示、负数用"1"表示。通常符号位设在数的最高位之前。尾数是指某数的本身数值部分。

采用纯小数定点表示时，计算机所能表示的有效数值(指绝对值)总是小于1的，但实际的有效数值可能大于1，这就需要在计算机解题之前，选择适当的比例因子，使全部参加运算数的有效数值都缩小若干倍，化成小于1的数值，而计算的结果又必须用相应的比例增大若干倍使其恢复为真实值。而且对运算结果加以估计，应该小于1，否则，计算机将产生"溢出"。

例如，二进制数+0.1001111 和−0.1001111，在定点机中的表示形式为：

采用整数定点表示时，计算机只能表示整数，与上述表示法并无原则上的区别。

小数点“.”，在计算机中实际是不表示出来的，需要预先约定它的位置。

2. 浮点表示法

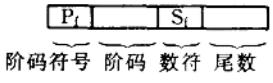
在浮点表示法中，小数点的位置不是固定的，而是浮动的。我们知道，任何一个二进制数 N 可以表示成下式：

$$N=2^P\times S$$

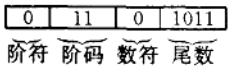
式中：S——数 N 的尾数，表示 N 的有效数值。用 S_f 表示尾数的符号，$S_f=0$ 表示正数；$S_f=1$ 表示负数。

P——数 N 的阶码，表示小数点的位置。用 P_f 表示阶码的符号位，$P_f=0$ 表示阶码为正数；$P_f=1$ 表示阶码为负数。

因此，浮点数分为阶码和尾数两个部分，在数的表示中都有各自的符号位，其形式为：



例如，$N=2^{+11}\times 1011$



可以看出，浮点数表示的范围比定点数表示的范围大。

3. 原码、补码和反码

主要介绍在计计算机中带符号数的各种表示法。

(1)真值与机器数

所谓真值就是数的本身，即用“+”、“−”号表示的数，称为真值。

例如：　$N_1=+1010100$

$N_2=-1001101$

机器数是将数的符号在机器中数码化了，正数的符号用 0 表示；负数的符号用 1 表示。

例如：　N_1:01010100

N_2:11001101

为了运算方便，在计算机中，机器数常用三种表示方法，即原码、补码和反码。下面分别进行讨论。

(2)原码表示法

原码表示法是最简单的一种机器数表示法，只要把真值的符号部分用 0 或 1 表示即

可。

例如：　　$N_1=+1001010$

　　$N_2=-1001010$

其原码记为：

$$[N_1]_{原}=[+1001010]_{原}=01001010$$

$$[N_2]_{原}=[-1001010]_{原}=11001010$$

由上述原码的表示形式，可将原码定义为：

$$[X]_{原}=\begin{cases}2^n+x & 当\ 0\leqslant x<2^{n-1}\\ 2^{n-1}-x & 当\ -2^{n-1}<x\leqslant 0\end{cases}$$

式中：x 为真值；

$[x]_{原}$为 x 的原码。

从原码定义可导出下列简单性质。

①当 $x>0$ 时，$[x]_{原}$与 x 的区别仅在于符号位用 0 表示；

②当 $x<0$ 时，$[x]_{原}$与 x 的区别仅在于符号位用 1 表示；

③当 $x<0$ 时，有$[+0]_{原}$和$[-0]_{原}$两种情况，即

$$[+0]_{原}=000\cdots0\quad(n个“0”)$$

$$[-0]_{原}=1\,00\cdots0\quad(n-1个“0”)$$

机器把这两种情况都当作 0 处理。

(3)补码表示法

补码表示法可以把负数转化为正数，使减法转换为加法，从而使正负数的加减运算转化为单纯的正数相加的运算，这可从后面数的运算中看到。

正数的补码表示与原码相同，即：

$$x=+0.x_1x_2\cdots x_n$$

$$[x]_{补}=0.x_1x_2\cdots x_n$$

而负数的补码表示是将它先变为反码，然后在最低位上加 1 形成，即：

$$x=-0.x_1x_2\cdots x_n$$

$$[x]_{补}=1.\overline{x}_1\overline{x}_2\cdots\overline{x}_n+0.\,00\cdots01\quad((n-1)个“0”)$$

例如：　$x=+0.1010$　　$x=-0.1101$

　　$[x]_{补}=0.1010$　　$[x]_{补}=1.0011$

若从补码求原码，其原则也是除符号外，尾数求反加 1。

补码的特点：

①正 0 的补码等于负 0 的补码：

$$[+0]_{补}=[-0]_{补}=000\cdots0$$

②8 位二进制补码所能表示的数值为：

$$+127\sim-128$$