

# 微机 及其应用 基本教程



邱小平 陈 斌 龙小波 编著



中山大学出版社

## 内容简介

本书是一本通俗易懂的电脑操作应用教材,是依据国家考试中心制定的计算机考试一级和二级考试大纲要求进行编写的,能适用于各种文化水平的初学者使用。

全书共分七章,第一、第二章是计算机的基础知识,第三至第五章是电脑操作的基本技能学习和训练,第六章是电脑编程的入门引导,第七章是目前常用的电脑应用软件工具的介绍和计算机病毒防治知识。学习了本书后,能应用电脑进行文章的输入和各类表格的制作及计算;能掌握基本的电脑数据库知识并能应用 dBASE III 结合工作和生活中各种需要而编写实用的管理程序;能具有维护电脑的基本技术,应用各种软件工具对电脑资源进行管理。

本书可作为文、理工、医类的大学、大专和中专学生学习计算机知识的教材,也可用作电脑操作员和初级程序员的培训教材,同时也是广大在职工作人员学习电脑的得力助手。

## 前 言

随着微型计算机的普及和推广,社会上对电脑应用人员的需要越来越迫切。笔者从事计算机教学多年,深感到极需要一本通俗易懂的电脑操作应用教材,能作为众多初学者的入门学习引导,以益于电脑应用的推广。为此我们编写了这本书,希望能受到欢迎。

全书共分七章,第一、第二章是计算机的基础,第三、第四至第五章是电脑操作的基本技能学习和训练,第六章是由基本操作向电脑编程应用的入门引导,第七章是目前常用的电脑应用软件工具的介绍和计算机病毒防治知识。学习了本书后,能应用电脑进行文章的输入和各类表格的制作及计算;能掌握基本的电脑数据库知识并能应用 dBASE Ⅲ 结合工作、生活中各种需要而编写实用的管理程序;能具有维护电脑的基本技术,应用各种软件工具对电脑资源进行管理。

本书是依据国家考试中心制定的计算机考试一级和二级考试大纲要求进行编写的,力求能通俗易懂,使其能适用于各种文化水平的初学者使用。本书可作为文、理工、医类的大学、大专和中专学生学习计算机知识的教材,也可用作电脑操作员和初级程序员的培训教材,也是广大在职工作人员学习电脑的得力助手。

本书的编写,其中第一、第二、第六章和第七章的第三节(NORTON)由邝小平编写,第三、第四章和第七章的第二节(QAPLUS)由龙小波编写,第五章和第七章的第一、第四节由陈斌编写,由邝小平对全书进行修改整理。本书于1993年编出后作为中山大学计算机科学系各种培训班和大专班的教材使用,得到各位教师和学生的好评。在编写和修改过程中特别得到了计算机科学系李宏新教授和中大出版社吴湘辉教授的许多具体指导,在此深表感谢。

在本书编写的过程中,参考了众多有关计算机方面的资料和各类教材,受益不浅,在此不再一一列举,仅以此拙作献出,为推广电脑的应用添上一砖一瓦。因我们的水平不高,书中难免有错漏之处,恳请各位读者提出宝贵的意见。

编者

1995年3月于广州

# 目 录

|                             |           |
|-----------------------------|-----------|
| <b>第一章 计算机基础知识 .....</b>    | <b>1</b>  |
| 1.1 计算机的运算基础——二进制 .....     | 1         |
| 1.1.1 进制数 .....             | 1         |
| 1.1.2 二进制数和十进制数的转换 .....    | 2         |
| 1.1.3 八进制数和十六进制数 .....      | 3         |
| 1.1.4 比特和字节的概念 .....        | 6         |
| 1.2 机器中数的表示 .....           | 6         |
| 1.2.1 机器数和字长 .....          | 6         |
| 1.2.2 数的正、负符号的表示 .....      | 6         |
| 1.2.3 小数点的表示 .....          | 7         |
| 1.2.4 有符号数的表示 .....         | 8         |
| 1.3 字符的编码 .....             | 8         |
| 1.4 计算机的电路基础——逻辑电路 .....    | 9         |
| 1.4.1 逻辑乘和“与门” .....        | 10        |
| 1.4.2 逻辑加和“或门” .....        | 11        |
| 1.4.3 逻辑非和“非门” .....        | 11        |
| 习题 .....                    | 12        |
| <b>第二章 微型计算机的组成 .....</b>   | <b>14</b> |
| 2.1 IBM PC 及其兼容机的硬件组成 ..... | 14        |
| 2.1.1 主机箱 .....             | 14        |
| 2.1.2 显示器、键盘和鼠标器 .....      | 17        |
| 2.1.3 打印机 .....             | 18        |
| 2.2 磁盘操作系统 DOS .....        | 21        |
| 2.2.1 DOS 的组成 .....         | 21        |
| 2.2.2 DOS 的文件管理 .....       | 22        |
| 2.2.3 DOS 的启动 .....         | 24        |
| 2.2.4 DOS 的常用功能键和编辑键 .....  | 25        |
| 2.2.5 常用 DOS 命令 .....       | 27        |
| 2.2.6 DOS 的内存管理 .....       | 43        |
| 2.2.7 系统配置文件和批处理文件 .....    | 46        |
| 习题 .....                    | 51        |
| <b>第三章 键盘操作 .....</b>       | <b>52</b> |

|                               |     |
|-------------------------------|-----|
| 3.1 键盘结构和功能                   | 52  |
| 3.2 键盘指法                      | 55  |
| 3.3 键盘指法练习软件 TT 的使用           | 57  |
| 习题和上机实习                       | 62  |
| <b>第四章 中文操作系统</b>             | 65  |
| 4.1 金山 DOS——Super-CCDOS V5.1  | 65  |
| 4.2 CC 版 2.13H 中文系统           | 69  |
| 4.3 王码中文系统——WM10S 5.0         | 75  |
| 4.4 CCDOS V4.0                | 77  |
| 4.5 中文输入方法                    | 79  |
| 4.5.1 区位码输入法                  | 79  |
| 4.5.2 拼音输入法                   | 80  |
| 4.5.3 五笔字型输入法                 | 86  |
| 习题                            | 98  |
| <b>第五章 文字编辑软件</b>             | 99  |
| 5.1 文字处理系统 WPS                | 99  |
| 5.1.1 WPS 软件的特点               | 99  |
| 5.1.2 WPS 的启动                 | 102 |
| 5.1.3 文本文件的编辑命令               | 104 |
| 5.1.4 打印控制和版面控制命令             | 118 |
| 5.1.5 版面编辑格式及制表               | 127 |
| 5.1.6 模拟显示与打印输出               | 131 |
| 5.1.7 其他编辑功能                  | 138 |
| WPS 命令速查表                     | 142 |
| 5.2 中文 WordStar               | 145 |
| 5.2.1 特点                      | 145 |
| 5.2.2 启动                      | 145 |
| 5.2.3 编辑文书文件                  | 149 |
| 5.2.4 打印控制                    | 151 |
| 5.2.5 盘满的处理                   | 152 |
| WS 命令一览表                      | 153 |
| 5.3 中、英文文表编辑软件 CCED           | 156 |
| 5.3.1 CCED 系统的启动              | 156 |
| 5.3.2 编辑命令                    | 157 |
| 习题与上机实习                       | 168 |
| <b>第六章 汉字 dBASE Ⅲ 数据库管理系统</b> | 170 |
| 6.1 基本概念                      | 170 |
| 6.1.1 数据库的概念                  | 170 |

|                         |                                |            |
|-------------------------|--------------------------------|------------|
| 6.1.2                   | 汉字 dBASE III 数据库的结构 .....      | 170        |
| 6.1.3                   | 数据的类型 .....                    | 171        |
| 6.1.4                   | 系统的组成和工作方式 .....               | 172        |
| 6.2                     | 汉字 dBASE III 基本操作命令 .....      | 173        |
| 6.2.1                   | 系统的启动和退出 .....                 | 174        |
| 6.2.2                   | 数据库的建立及库文件的打开和关闭 .....         | 174        |
| 6.2.3                   | 库文件结构的显示和修改 .....              | 176        |
| 6.2.4                   | 数据的输入, 查看和记录定位 .....           | 177        |
| 6.2.5                   | 数据的表示形式 .....                  | 181        |
| 6.2.6                   | 数据的修改和记录的插入、删除 .....           | 189        |
| 6.2.7                   | 数据的排序和索引 .....                 | 192        |
| 6.2.8                   | 数据的统计 .....                    | 195        |
| 6.3                     | 数据库之间的操作 .....                 | 197        |
| 6.3.1                   | 工作区的选择 .....                   | 197        |
| 6.3.2                   | 使用另一库文件的数据更新现用库文件的数据 .....     | 199        |
| 6.3.3                   | 两个库文件的记录关联定位 .....             | 199        |
| 6.3.4                   | 两个库文件联接而产生一个新的库文件 .....        | 200        |
| 6.3.5                   | 用其他库文件向现用库文件追加记录 .....         | 201        |
| 6.4                     | 命令文件的设计 .....                  | 202        |
| 6.4.1                   | 命令文件的建立和修改 .....               | 202        |
| 6.4.2                   | 输入、输出语句 .....                  | 203        |
| 6.4.3                   | 命令文件的基本结构 .....                | 205        |
| 6.4.4                   | 命令文件中使用的其他命令 .....             | 216        |
| 6.5                     | 输入输出格式的设计 .....                | 219        |
| 6.5.1                   | 库文件数据输入的屏幕格式设计 .....           | 219        |
| 6.5.2                   | 程序模式下的报表打印输出 .....             | 222        |
| 习题与上机实习 .....           |                                | 225        |
| <b>第七章 应用软件介绍 .....</b> |                                | <b>228</b> |
| 7.1                     | 微机工具箱 PC TOOLS .....           | 228        |
| 7.1.1                   | 启动 .....                       | 228        |
| 7.1.2                   | 路径选择 .....                     | 229        |
| 7.1.3                   | 文件服务 .....                     | 230        |
| 7.1.4                   | 磁盘服务 .....                     | 233        |
| 7.2                     | 检测软件 QAPLUS 的使用 .....          | 240        |
| 7.2.1                   | 简介 .....                       | 240        |
| 7.2.2                   | 使用 .....                       | 240        |
| 7.3                     | 实用程序 NORTOR 的介绍 .....          | 246        |
| 7.3.1                   | Nortor Disk Doctor (NDD) ..... | 247        |
| 7.3.2                   | Disk Tools (DISKOOL) .....     | 257        |
| 7.4                     | 计算机病毒的防治 .....                 | 262        |

|       |                  |     |
|-------|------------------|-----|
| 7.4.1 | 计算机病毒的概念 .....   | 262 |
| 7.4.2 | 计算机病毒造成的危害 ..... | 262 |
| 7.4.3 | 制造计算机病毒的动机 ..... | 263 |
| 7.4.4 | 计算机病毒的分类 .....   | 264 |
| 7.4.5 | 计算机病毒的原理 .....   | 265 |
| 7.4.6 | 计算机病毒的预防 .....   | 266 |
| 7.4.7 | 计算机病毒的诊治 .....   | 267 |
| 7.4.8 | 消毒软件的使用 .....    | 269 |

## 第一章 计算机基础知识

## 1.1 计算机的运算基础——二进制

### 1.1.1 二进制数

电子计算机是能对数据进行高速处理的机器,这些数据信息是用数字、字符、符号和表达式来表示的。由于计算机是用一些具有两种稳定状态的开关电子元件所组成,所以计算机里所处理的数据就必须用与这种元件的两种状态相对应的二进制的表示。

## 二、二进制数是一种怎样的数制？

首先我们讨论一下我们日常所习惯的十进制数。在十进制中的数是用 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 这十个数字组成的, 它遵循着逢十进一的规则, 即每一位的数当其数值满十时, 便向前一位进一位。

例如：

$$\begin{array}{r} 3.5621 \\ + 8.2432 \\ \hline 11.8053 \end{array}$$

这是一个小学生都懂得的加法运算。

在十进制数中,我们称小数点左边第一位为个位,其位权为  $10^0$ ;

小数点左边第二位为十位,其位权为  $10^1$ ;

小数点左边第三位为百位,其位权为  $10^2$ ;

• • • • •

小数点右边第一位为十分之一位,其位权为  $10^{-1}$ ;

小数点右边第二位为百分之一位,其位权为  $10^{-2}$ ;

● ● ●      ● ● ●

如此类推,便可以写出成千上万的数字。

现在来看看二进制数。它们是用 0 和 1 这两个数字组成的，并遵循着逢二进一的规则。

例如：

$$\begin{array}{r} 1011 \\ + 1001 \\ \hline 10100 \end{array}$$

进一位   进一位   进一位

当某个位的数值满为 2 时,便向前一位进 1。

二进制数也有小数位,例:1101.111。

在小数点左边第一位,其位权为  $2^0$ ;

在小数点左边第二位,其位权为  $2^1$ ;

在小数点左边第三位,其位权为  $2^2$ ;

... ..

在小数点右边第一位,其位权为  $2^{-1}$ ;

在小数点右边第二位,其位权为  $2^{-2}$ ;

... ..

同理,依此类推,也可用二进制数写出成千上万的数字。在计算机里便是用这种二进制数来表示数据的。

### 1.1.2 二进制数和十进制数的转换

从两种数制的位权关系,我们不难看出它们之间进行转换的方法。

二进制数如何换算成十进制数? 请看下例:

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

二进制数 1011 是等于十进制数的 11。

$$\begin{aligned} \text{又例 } (111.101)_2 &= 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= (7.625)_{10} \end{aligned}$$

二进制数 111.101 是等于十进制数的 7.625。

显然,二进制数转化为十进制数,只要将其各位按位权写出展开式,用十进制的加法进行相加所得的和便是其对应的十进制数。

十进制数又如何转换成二进制数呢? 例如十进制数 12,我们按二进制的位权将其写成展开式:

$$(12)_{10} = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = (1100)_2$$

便得出十进制数的 12 是等于二进制数的 1100。这个展开式的各位是如何求出的? 它是用 2 不断去除 12,每次所得的余数 0 或 1 便是二进制相应位的数字,直除到商为 0 时为止。

|   |    |       |                          |
|---|----|-------|--------------------------|
| 2 | 12 | ————— | 余数为 0 ( $0 \times 2^0$ ) |
| 2 | 6  | ————— | 余数为 0 ( $0 \times 2^1$ ) |
| 2 | 3  | ————— | 余数为 1 ( $1 \times 2^2$ ) |
| 2 | 1  | ————— | 余数为 1 ( $1 \times 2^3$ ) |
|   | 0  |       |                          |

上述这种方法叫做除 2 取余法,是适用于将十进制数的整数部分转换成二进制数的方法。

要将十进制数小数部分转换成二进制数,使用的是乘 2 取整法,它是用 2 不断去乘

十进制数的小数部分,把每次所得部分积的整数(0 或 1)作为二进制数相应位的数值。  
例如:

$$\begin{array}{rcl}
 0.356 & & \\
 \times 2 & & \\
 \hline
 0.712 & \text{————— 整数部分为 } 0 (=0 \times 2^{-1}) & \\
 \times 2 & & \\
 \hline
 1.424 & \text{————— 整数部分为 } 1 (=1 \times 2^{-2}) & \\
 \times 2 & & \\
 \hline
 0.848 & \text{————— 整数部分为 } 0 (=0 \times 2^{-3}) & \\
 \times 2 & & \\
 \hline
 1.696 & \text{————— 整数部分为 } 1 (=1 \times 2^{-4}) & \\
 \times 2 & & \\
 \hline
 1.392 & \text{————— 整数部分为 } 1 (=1 \times 2^{-5}) & \\
 \dots & & 
 \end{array}$$

$$\text{得出 } (0.356)_{10} = 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5} + \dots$$

需要注意的是,这种乘法计算不需要无限进行下去,只要能满足精度要求便可截止。

$$\text{例如: } (0.356)_{10} \approx (0.0101101)_2$$

### 1.1.3 八进制数和十六进制数

在计算机中为了表示数的方便,也常用八进制和十六进制的表示方法。

#### 1. 八进制和十六进制的数表示方法

八进制数是用 0 ~ 7 这八个数字组成的,其遵循逢八进一的规则,在计算机中是用三位二进制数表示一位八进制数,例如:

$$(7)_8 = (111)_2$$

$$(13)_8 = (001011)_2$$

八进制各位的位权基数是 8,即:

在小数点左边第一位,其位权为  $8^0$ ;

在小数点左边第二位,其位权为  $8^1$ ;

...

在小数点右边第一位,其位权为  $8^{-1}$ ;

在小数点右边第二位,其位权为  $8^{-2}$ ;

...

同理,十六进制数是遵循逢十六进一的规则,组成十六进制数的数字有 16 个,除了从 0~9 这十个数字外,分别用 A,B,C,D,E,F 这几个字母表示 10~15 的数字。在计算机中是用四位二进制数组成一位十六进制数。例如:

$$(5)_{16} = (0101)_2 \quad (11)_{16} = (00010001)_2 \quad (7B)_{16} = (01111011)_2$$

十六进制各位的位权基数是 16,即:

在小数点左边第一位,其位权为  $16^0$ ;

... ..

在小数点右边第一位,其位权为  $16^{-1}$ ;

... ..

## 2. 数制之间的转换

### (1) 二进制数和八进制数、十六进制数之间的转换

二进制数转换成八进制数的方法:

将二进制数的整数部分和小数部分分别以三位为一组,每组求出对应的八进制数字。例如:

二进制数 001 101 010 . 110 110 ; 010 011 101 . 001 110

八进制数 1 5 2 . 6 6 ; 2 3 5 . 1 6

八进制数转换为二进制数的方法是上述的逆方法,即把每位八进制数用三位二进制数展开,便可变成二进制数。例如:

八进制数 7 6 5 . 4 2 ; 1 0 3 . 4

二进制数 111 110 101 . 100 010 ; 001 000 011 . 100

同理二进制数和十六进制数之间的转换是使用同样的方法,只是用四位二进制数组成一位的十六进制数,例如:

二进制数 0001 0110 1011 . 0011 0100 ; 0001 1101 0111 . 1000 1000

十六进制数 1 6 B . 3 4 ; 1 D 7 . 8 8

### (2) 十进制数转换为八进制数或十六进制数

其与前面介绍的十进制数转换为二进制数的方法是一样的,同样要求对整数部分和小数部分分别进行转换。整数部分是用位权的基数除以十进制数,取其整数部分;小数部分是用位权的基数乘以十进制数,取其整数部分。对于八进制,位权基数为 8;而十六进制的位权基数为 16。

例 1 将十进制数 96.35 转换为八进制数

整数部分的转换

$$\begin{array}{r|l} 8 & 96 \\ \hline 8 & 12 \\ \hline 8 & 1 \end{array} \quad \begin{array}{l} \text{—————余数为 0 } (0 \times 8^0) \\ \text{—————余数为 4 } (4 \times 8^1) \\ \text{—————余数为 1 } (1 \times 8^2) \end{array}$$

小数部分的转换 (取小数点后的 3 位)

0.35

× 8

2.80 ————— 整数部分为 2 ( $=2 \times 8^{-1}$ )

× 8

6.40 ————— 整数部分为 6 ( $=6 \times 8^{-2}$ )

× 8

3.20 ————— 整数部分为 3 ( $=3 \times 8^{-3}$ )

...

得出:  $(96.35)_{10} \approx (140.263)_8$

例 2 将十进制数 96.35 转换成十六进制数

整数部分的转换

|    |    |       |                           |
|----|----|-------|---------------------------|
| 16 | 96 | ————— | 余数为 0 ( $0 \times 16^0$ ) |
| 16 | 6  | ————— | 余数为 6 ( $6 \times 16^1$ ) |
|    | 0  |       |                           |

小数部分的转换 (取小数点后的 3 位)

0.35

× 16

5.60 ————— 整数部分为 5 ( $=5 \times 16^{-1}$ )

× 16

9.60 ————— 整数部分为 9 ( $=9 \times 16^{-2}$ )

× 16

9.60 ————— 整数部分为 9 ( $=9 \times 16^{-3}$ )

...

得出:  $(96.35)_{10} \approx (60.599)_{16}$

十进制数,二进制数,八进制和十六进制数基本数字之间的关系及表示如下表:

| 十进制 | 二进制  | 八进制 | 十六进制 | 十进制 | 二进制   | 八进制 | 十六进制 |
|-----|------|-----|------|-----|-------|-----|------|
| 0   | 0000 | 0   | 0    | 9   | 1001  | 11  | 9    |
| 1   | 0001 | 1   | 1    | 10  | 1010  | 12  | A    |
| 2   | 0010 | 2   | 2    | 11  | 1011  | 13  | B    |
| 3   | 0011 | 3   | 3    | 12  | 1100  | 14  | C    |
| 4   | 0100 | 4   | 4    | 13  | 1101  | 15  | D    |
| 5   | 0101 | 5   | 5    | 14  | 1110  | 16  | E    |
| 6   | 0110 | 6   | 6    | 15  | 1111  | 17  | F    |
| 7   | 0111 | 7   | 7    | 16  | 10000 | 20  | 10   |
| 8   | 1000 | 10  | 8    | 17  | 10001 | 21  | 11   |

#### 1.1.4 比特和字节的概念

我们把二进制数的一位称为 Bit,其可作为量度信息的单位,二进制数的一位包含的信息称为一比特(Bit)。

另外,把 8 位二进制数作为一个位组,称为字节(Byte)。字节通常用来作为计算机存储器的单位。例如表示主机的内存为 640KB,此处表示单位的 B 便是 Byte 即字节,而 K 是等于  $2^{10}$  即 1024,一般我们读作 640K 字节。

### 1.2 机器中数的表示

现在我们已经知道,在计算机里是采用二进制数的,在机器中表示的数,表示的字母和符号,甚至控制计算机进行操作的命令,全部都是采用二进制编码来表示的。

#### 1.2.1 机器数和字长

在计算机里能表示和保存二进制信息的基本部件叫寄存器,每一位寄存器都有两种稳定状态,用高、低电平分别表示“1”和“0”。寄存器有多少位,就能同时存放多少位的二进制代码,也即可以一次处理信息的字长。字长是衡量计算机技术性能的指标之一,它表示机器处理数据的速度和精确度的能力。目前的微型计算机有 8 位字长的(例如中华学习机),有 16 位字长的(例如 IBM PC 机和 286 机),并发展到 32 位字长的(例如 386 微机和 486 微机)。

数在机器中的表示形式称为机器数,机器数所表示数的范围受到字长的限制。8 位字长的机器数,能表示二进制数的最大值是  $(11111111)_2 = (255)_{10}$ ,是一个无符号的整数。当数值超出此范围就不能正确表示,便会产生错误,这种情况称为“溢出”。

#### 1.2.2 数的正、负符号的表示

我们知道,数是有正数和负数之分,在数学上表示正数和负数,是在数字前面加上

一个“+”号或“-”号,例如+127和-127。在计算机里要表示这个“正”和“负”的符号,仍然要转化为二进制数字来表示,用“0”表示正,用“1”表示负,并占据字长的第一位来作为符号位。因为在机器中,数的字长是固定的,所以要表示带有符号的数时其范围便缩小了。例如8位字长的机器数,表示有符号数的范围是:最大正整数 $(01111111)_2 = (+127)_{10}$ 到最小负整数 $(10000000)_2 = (-128)_{10}$ 。

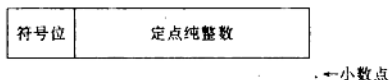
### 1.2.3 在计算机里小数点位置的表示

在计算机的数中,小数点不占据字长的一位空间,只是表示其所在的位置。

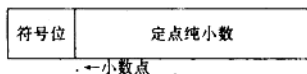
表示小数点位置的方法有两种,一种是小数点的位置固定不变,这种数称为定点数,另一种是小数点的位置可以变动,这种数称为浮点数。

#### (1) 定点数

若小数点定位在字长最低位的右边,便为定点纯整数:



若小数点定位在符号位和第一位数字之间,则为定点纯小数:



#### (2) 浮点数

小数点的位置用阶码来表示,随着阶码数值的不同,小数点的位置有所不同。如何认识阶码?

先看一个十进制数的例子:

$$\begin{array}{ccccc}
 20 & \times & 10^1 & = & 200. \\
 \uparrow & & \uparrow & & \uparrow \\
 \text{尾数} & & \text{阶码阶码} & & \text{小数点} \\
 \downarrow & & \downarrow \downarrow & & \downarrow \\
 20 & \times & 10^{-2} & = & 0.2
 \end{array}$$

同样的尾数20当其阶码不同时小数点的位置也不同,表示的数值含义也就不一样。同理二进制数也可用这种方法来表示小数点的位置。例:

$$1011 \times 2^{-1} = 101.1$$

$$1011 \times 2^{-2} = 10.11$$

在计算机中浮点数表示的形式为：

|          |    |          |    |
|----------|----|----------|----|
| 阶码<br>符号 | 阶码 | 尾数<br>符号 | 尾数 |
|----------|----|----------|----|

### 1.2.4 在计算机里有符号数的表示方法

在计算机里的运算核心部件是一个加法器，实际上是把加、减、乘、除的各种运算，通过一定的方法转换成加法运算而计算出其结果。对于有符号的数，也要把符号位考虑到这种转换中，即要求将符号位和数值一起编码参加运算。这种编码的方法有三种：原码、补码、和反码。

#### (1) 原码

即一般表示有符号机器数的方法：符号位用 0 表示正，1 表示负，数值是原二进制的机器数。若用  $X$  表示机器数，原码符号记作  $[X]_{\text{原}}$

例 机器数  $X_1 = +1011001$  原码  $[X_1]_{\text{原}} = 01011001$   
 $X_2 = -1010011$   $[X_2]_{\text{原}} = 11010011$

#### (2) 补码 符号记作 $[X]_{\text{补}}$

当机器数为正数，其补码与原码相同；

当机器数为负数，其补码的符号位取“1”，其它数位的值取反值，再在末位加 1。

例  $X_1 = +1011001$   $X_2 = -1010011$   
 $[X_1]_{\text{补}} = 01011001$   $[X_2]_{\text{补}} = 10101101$   
 $= [X_1]_{\text{原}}$

#### (3) 反码 符号记作 $[X]_{\text{反}}$

当机器数为正数时，其反码与原码相同；

当机器数为负数时，其反码的符号位取“1”，其它数位的值取反值。

例  $X_1 = +1011001$   $X_2 = -1010011$   
 $[X_1]_{\text{反}} = 01011001$   $[X_2]_{\text{反}} = 10101100$

从上面的定义可以看出，负数的补码是等于其反码在末位加 1。

## 1.3 字符的编码

计算机是用二进制编码表示字母和符号的，目前最普遍使用的字符编码是 ASCII 码，即美国标准信息交换码 (American Standard Code for Information Interchange)，是由七位二进制数组成的编码，一般将其转换成十进制码或十六进制码表示，如下所示：

ASCII 码表

| 字符 | 十进制码 | 十六进制码 | 字符 | 十进制码 | 十六进制码 | 字符 | 十进制码 | 十六进制码 |
|----|------|-------|----|------|-------|----|------|-------|
| !  | 33   | 21    | )  | 41   | 29    | 1  | 49   | 31    |
| "  | 34   | 22    | *  | 42   | 2A    | 2  | 50   | 32    |
| #  | 35   | 23    | +  | 43   | 2B    | 3  | 51   | 33    |
| \$ | 36   | 24    | ,  | 44   | 2C    | 4  | 52   | 34    |
| %  | 37   | 25    | -  | 45   | 2D    | 5  | 53   | 35    |
| &  | 38   | 26    | .  | 46   | 2E    | 6  | 54   | 36    |
| '  | 39   | 27    | /  | 47   | 2F    | 7  | 55   | 37    |
| (  | 40   | 28    | 0  | 48   | 30    | 8  | 56   | 38    |
| 9  | 57   | 39    | A  | 65   | 41    | I  | 73   | 49    |
| :  | 58   | 3A    | B  | 66   | 42    | J  | 74   | 4A    |
| ;  | 59   | 3B    | C  | 67   | 43    | K  | 75   | 4B    |
| <  | 60   | 3C    | D  | 68   | 44    | L  | 76   | 4C    |
| =  | 61   | 3D    | E  | 69   | 45    | M  | 77   | 4D    |
| >  | 62   | 3E    | F  | 70   | 46    | N  | 78   | 4E    |
| ?  | 63   | 3F    | G  | 71   | 47    | O  | 79   | 4F    |
| @  | 64   | 40    | H  | 72   | 48    | P  | 80   | 50    |
| Q  | 81   | 51    | Y  | 89   | 59    | a  | 97   | 61    |
| R  | 82   | 52    | Z  | 90   | 5A    | b  | 98   | 62    |
| S  | 83   | 53    | [  | 91   | 5B    | c  | 99   | 63    |
| T  | 84   | 54    |    | 92   | 5C    | d  | 100  | 64    |
| U  | 85   | 55    | ]  | 93   | 5D    | e  | 101  | 65    |
| V  | 86   | 56    | ^  | 94   | 5E    | f  | 102  | 66    |
| W  | 87   | 57    | _  | 95   | 5F    | g  | 103  | 67    |
| X  | 88   | 58    | '  | 96   | 60    | h  | 104  | 68    |
| i  | 105  | 69    | q  | 113  | 71    | y  | 121  | 79    |
| j  | 106  | 6A    | r  | 114  | 72    | z  | 122  | 7A    |
| k  | 107  | 6B    | s  | 115  | 73    | {  | 123  | 7B    |
| l  | 108  | 6C    | t  | 116  | 74    |    | 124  | 7C    |
| m  | 109  | 6D    | u  | 117  | 75    | }  | 125  | 7D    |
| n  | 110  | 6E    | v  | 118  | 76    | ~  | 126  | 7E    |
| o  | 111  | 6F    | w  | 119  | 77    |    |      |       |
| p  | 112  | 70    | x  | 120  | 78    |    |      |       |

#### 1.4 计算机的电路基础——逻辑电路

计算机的基本电路单元是具有两种稳定状态的逻辑电路,分析这种电路的数学工具是逻辑代数。逻辑代数中只有两种可取的值:“1”和“0”,这不是数字的1和0,而是表示逻辑值的“真”和“假”。对应逻辑代数的三种基本运算有三种基本的逻辑电路。

### 1.4.1 逻辑乘和“与”门

设有两个变量 A 和 B。

逻辑乘的表达式为： $A \times B = F$  或  $A \wedge B = F$ 。

它的含义是：只有当 A 和 B 都是 1 时，F 才为 1，否则 F 为 0。

图 1.4-1 为一个用两个串联的开关控制一盏灯的电路，其非常形象地说明了这种逻辑乘的关系。如图示，我们设开关闭合时表示的状态为 1，开关打开时的状态为 0；电灯亮时状态为 1，灭时状态为 0。

第一章 图示

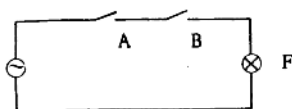


图 1.4-1

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

逻辑乘真值表

不难看出，只有当 A 和 B 两开关同时都闭合时，即都为 1 时，电灯 F 才会亮，这便是逻辑乘的含义。图右的表格，列出了 A、B、F 三者在逻辑乘中所有可能的运算关系，我们称这种表格为真值表。

在计算机中实现逻辑乘的电路称为“与”门电路。图 1.4-2 是一种简单的二极管“与”门逻辑电路。图中 A、B 是“与”门的输入端，F 为“与”门的输出端。分析此电路可知，当二极管  $D_a$  或  $D_b$  的负极 A 端或 B 端为低电平时，二极管便会导通，使到电路输出端 F 点的电平为低，即为 0 状态。只有  $D_a$  和  $D_b$  两二极管都不导通，即 A 和 B 端均为高电平时，门电路输出端 F 点的电平才能为高电平，即为 1。这便是逻辑乘的关系。作为“与”门电路，其输入端可以多于两个以上，但输出端只有一个，不管其输入端有几个，都必须全部为 1 即全为高电平时，门电路的输出端才可能为 1。图 1.4-3 是与“门”电路的常用符号图。

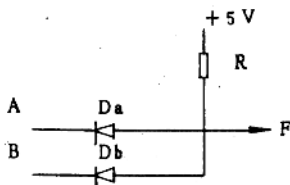


图 1.4-2 二极管与门电路

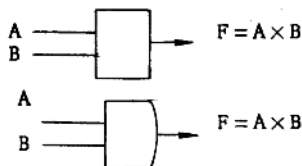


图 1.4-3 与门电路符号图