

## 第 2 章 计算机体系结构与组织

### 学习目标

- 了解数据的机器编码、计算机系统的组成、存储系统的组织结构、基本的输入/输出系统、计算机系统的分类、多核技术、高性能计算机、并行计算机、网络计算机、分布式系统。
- 掌握数值数据的表示和计算、非数值数据的编码、数字逻辑与数字系统。

学习计算机知识应从一些基本的概念入手，包括数值数据在计算机中的表示与运算、非数值数据的表示、数据的机器编码、数字逻辑与数字系统等与计算机相关的基础知识。

### 2.1 数值数据

计算机只识别二进制编码的指令和数据，其他的如数字、字符、声音、图形、图像等信息都必须转换成二进制的形式，计算机才能进行识别和处理。二进制只有两个状态(即 0 和 1)，这正好与物理器件的两种状态相对应，如电压信号的高与低、门电路的导通与截止等；而十进制电路则需要用 10 种状态来描述，这将使得电路十分复杂，处理起来也非常困难。因此，采用二进制将使得计算机在物理实现上变得简单，且具有可靠性高、处理简单、抗干扰能力强等优点。下面对计算机中的各种数据表示进行详细介绍。

#### 2.1.1 数的表示及数制转换

十进制数是日常生活中常用的，一直伴随着人们的生活，它也是中国的一大发明。在商代时，中国已采用了十进制。从现已发现的商代陶文和甲骨文中，可以看到当时已能够用一、二、三、四、五、六、七、八、九、十、百、千、万 13 个数字，记十万以内的任何自然数。这些记数文字的 shape，在后世虽有所变化并演变成为现在的写法，但记数方法却从没有中断，一直被沿袭的同时日趋完善。十进制的记数法是古代世界中最先进、最科学的记数法，对世界科学和文化的发展起着不可估量的作用。正如李约瑟 (Joseph Terence Montgomery Needham, 1900—1995) 所说的：“如果没有这种十进制，就不可能出现我们现在这个统一化的世界了。”



Joseph Terence Montgomery  
Needham

除了十进制外，其他进制，如十二进制(时钟)等也在生活中广泛使用。而在计算机内所有的数据都是以二进制代码的形式存储、处理和传送，但是在输入/输出或书写时，为了用户的方便，也经常用到八进制和十六进制。



在十进制系统中,进位原则是“逢十进一”。由此可知,在二进制系统中,其进位原则是“逢二进一”;在八进制系统中,其进位原则是“逢八进一”;在十六进制系统中,其进位原则是“逢十六进一”。为了弄清进制概念及其关系,有必要掌握各种进位制数的表示方法以及不同进位制数的相互转换方法。

### 1. 数制的相关基本概念

在进位计数的数字系统中,如果只用  $R$  个基本符号(如  $0,1,2,\dots,R-1$ )来表示数值,则称其为“基  $R$  数制”。在各种进制中,基和位权这两个基本概念对数制的理解和多种数制之间的转换起着至关重要的作用。

(1) 基。称  $R$  为该数制的“基数”,简称“基”或“底”。例如,十进制数制的基  $R=10$ 。

(2) 位权。数值中每一固定位置对应的单位称为“位权”,简称“权”。它以数制的基为底,以整数为指数组成。例如,十进制数制的位权为  $1,10^2,10^1,10^0,10^{-1},10^{-2},L$ 。

对十进制数,可知  $R=10$ ,它的基本符号有 10 个,分别为  $0,1,2,\dots,9$ 。对二进制数,则取  $R=2$ ,其基本符号为 0 和 1。

进位计数的编码符合“逢  $R$  进位”的规则。各位的权是以  $R$  为底的幂,一个数可按权展开成多项式。例如,十进制数 523.47 可按权展开如下:

$$523.47=5\times 10^2+2\times 10^1+3\times 10^0+4\times 10^{-1}+7\times 10^{-2}$$

因此,可将任意数制的数  $N$  表示为以下通式:

$$\begin{aligned}(N)_R &= D_m \cdot R^m + D_{m-1} \cdot R^{m-1} + L + D_2 \cdot R^2 + D_1 \cdot R^1 + D_0 \cdot R^0 + D_{-1} \cdot R^{-1} + D_{-2} \cdot R^{-2} + L + D_{-k} \cdot R^{-k} \\ &= \sum_{i=-k}^m D_i R^i\end{aligned}$$

式中,  $(N)_R$  表示  $R$  进制的数  $N$ ,该数共有  $m+k+1$  位,且  $m$  和  $k$  为正整数;  $D_i$  可以是  $R$  进制的基本符号中的任意一个;  $R^i$  为该进制的权,  $R$  为基数或“底”。

在计算机中常用的数制有十进制、二进制、八进制、十六进制数制,它们的基、位权及基本符号总结如表 2.1 所示。

表 2.1 各种进制的基、位权及基本符号

| 进制名称 | 基 $R$ | 位 权                                                       | 基本符号                        |
|------|-------|-----------------------------------------------------------|-----------------------------|
| 十进制  | 10    | $\dots,10^3,10^2,10^1,10^0,10^{-1},10^{-2},10^{-3},\dots$ | $0,1,2,\dots,9$             |
| 二进制  | 2     | $\dots,2^3,2^2,2^1,2^0,2^{-1},2^{-2},2^{-3},\dots$        | $0,1$                       |
| 八进制  | 8     | $\dots,8^3,8^2,8^1,8^0,8^{-1},8^{-2},8^{-3},\dots$        | $0,1,2,\dots,7$             |
| 十六进制 | 16    | $\dots,16^3,16^2,16^1,16^0,16^{-1},16^{-2},16^{-3},\dots$ | $0,1,2,\dots,9,A,B,C,D,E,F$ |

### 2. 数制的表示

数制的表示方法有很多种,常用的有以下几种。

#### 1) 下标法

下标法是指用小括号将所表示的数据括起来,然后在括号外的右下角写上数制的基  $R$ 。

**例 2.1**  $(1001.01)_2$ 、 $(751)_8$ 、 $(560)_{10}$ 、 $(63AC)_{16}$  分别表示一个二进制数、八进制数、十进制数和十六进制数。



## 2) 字母法

字母法是指在所表示的数的末尾写上相应数制字母。对应的进制与字母如表 2.2 所示。

表 2.2 进制与字母

| 进制   | 十进制 | 二进制 | 八进制 | 十六进制 |
|------|-----|-----|-----|------|
| 所用字母 | D   | B   | Q   | H    |

由于生活中常用的数制为十进制，因此，有时对于十进制数，可以省略其后的字母 D。

**例 2.2** 1001.01B、751Q、560D 或 560、63ACH 分别表示一个二进制数、八进制数、十进制数和十六进制数。

## 3. 数制间的基本关系

十进制、二进制、八进制和十六进制数制之间的基本关系如表 2.3 所示。

表 2.3 数制间的基本关系

| 十进制 | 二进制  | 八进制 | 十六进制 |
|-----|------|-----|------|
| 0   | 0000 | 00  | 0    |
| 1   | 0001 | 01  | 1    |
| 2   | 0010 | 02  | 2    |
| 3   | 0011 | 03  | 3    |
| 4   | 0100 | 04  | 4    |
| 5   | 0101 | 05  | 5    |
| 6   | 0110 | 06  | 6    |
| 7   | 0111 | 07  | 7    |
| 8   | 1000 | 10  | 8    |
| 9   | 1001 | 11  | 9    |
| 10  | 1010 | 12  | A    |
| 11  | 1011 | 13  | B    |
| 12  | 1100 | 14  | C    |
| 13  | 1101 | 15  | D    |
| 14  | 1110 | 16  | E    |
| 15  | 1111 | 17  | F    |

## 4. 数制之间的转换

二进制、八进制、十六进制和十进制数制之间的相互转换有一定的规律，只要掌握了它们之间的基本规律，就很容易进行这些数制之间的相互转换。

### 1) 其他进制数转换为十进制数

其他进制转换为十进制的具体转换方法：相应位置的数码乘以对应位的权，再将所有的乘积进行累加，即得对应的十进制数。



$$\begin{aligned}
 \text{例 2.3} \quad (1101.101)_2 &= (1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3})_{10} \\
 &= (8 + 4 + 0 + 1 + 0.5 + 0 + 0.125)_{10} \\
 &= (13.625)_{10}
 \end{aligned}$$

$$\begin{aligned}
 \text{例 2.4} \quad (9B.F)_{16} &= (9 \times 16^1 + 11 \times 16^0 + 15 \times 16^{-1})_{10} \\
 &= (144 + 11 + 0.9375)_{10} \\
 &= (155.9375)_{10}
 \end{aligned}$$

## 2) 二进制数、八进制数、十六进制数之间的转换

八进制数和十六进制数是从二进制数演变而来的。其具体转换方法：由 3 位二进制数组成 1 位八进制数，由 4 位二进制数组成 1 位十六进制数。对于同时有整数和小数部分的数，则以小数点为界，对小数点前后的数分别进行分组处理，不足的位数用 0 补足，对整数部分将 0 补在数的左边，对于小数部分则将 0 补在数的右边。

$$\text{例 2.5} \quad (1 \underline{101.100 \ 1})_2 = (\underline{001 \ 101.100 \ 100})_2 = (15.44)_8$$

$$\text{例 2.6} \quad (1 \underline{0011.1011})_2 = (\underline{0001 \ 0011.1011})_2 = (13.B)_{16}$$

$$\text{例 2.7} \quad (2B.F2)_{16} = (\underline{0010 \ 1011.1111 \ 0010})_2 = (101011.1111001)_2$$

## 3) 十进制数转换为十六进制数

(1) 整数的转换方法：除基取余法。即用十进制整数除以要转换的进制数的基，取余数，从低位向高位逐次进行，然后对商继续这一操作，直到商为零为止。

**例 2.8** 现以  $(368)_{10}$  为例，利用除  $R$  取余法将其转换为二进制整数的过程描述如下：

$$\begin{array}{rcl}
 2) \underline{184} & \dots\dots\dots & x_0=0 \\
 2) \underline{92} & \dots\dots\dots & x_1=0 \\
 2) \underline{46} & \dots\dots\dots & x_2=0 \\
 2) \underline{23} & \dots\dots\dots & x_3=0 \\
 2) \underline{11} & \dots\dots\dots & x_4=1 \\
 2) \underline{5} & \dots\dots\dots & x_5=1 \\
 2) \underline{2} & \dots\dots\dots & x_6=1 \\
 2) \underline{1} & \dots\dots\dots & x_7=0 \\
 0 & \dots\dots\dots & x_8=1
 \end{array}$$

$$\begin{aligned}
 \text{即} (368)_{10} &= (x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0)_2 \\
 &= (101110000)_2
 \end{aligned}$$

将除  $R$  取余法推广，可以得到将十进制整数转换为其他进制数的方法，只需要将  $R$  转换为所转换的进制数的基即可。

(2) 小数的转换方法：乘基取整法。即将十进制小数乘以要转换的进制数的基，取整数，然后对小数点后的数继续这一操作，直到小数点后为零或达到所要求的精度为止。

**例 2.9** 以  $(0.5625)_{10}$  为例，说明其转换过程。

$$\begin{array}{rcl}
 0.5625 & & \\
 \times 2 & & \\
 1.1250 & \dots\dots\dots & x_{-1}=1 \\
 0.1250 & & \\
 \times 2 & &
 \end{array}$$



$$\begin{array}{r}
 0.2500 \quad \dots\dots\dots x_{-2}=0 \\
 0.2500 \\
 \times \quad 2 \\
 \hline
 0.5000 \quad \dots\dots\dots x_{-3}=0 \\
 0.5000 \\
 \times \quad 2 \\
 \hline
 1.000 \quad \dots\dots\dots x_{-4}=1
 \end{array}$$

$$\begin{aligned}
 \text{即}(0.5625)_{10} &= (0.x_{-1}x_{-2}x_{-3}x_{-4})_2 \\
 &= (0.1001)_2
 \end{aligned}$$

(3) 十进制混合小数的转换：将十进制数的整数和小数分别按上述方法进行转换，然后再组合到一起。十进制混合小数由整数和纯小数组成。将十进制数的整数部分按除 2 取余法转换为  $R$  进制的整数部分，将十进制的纯小数部分按乘 2 取整法转换为  $R$  进制的小数部分，然后再将  $R$  进制的整数部分和小数部分组合起来构成  $R$  进制混合小数。

**例 2.10** 十进制混合小数为  $(368.5625)_{10}$  的整数部分是 368，其纯小数部分是 0.5625。将 368 用除 2 取余法转换获得二进制数为  $(101110000)_2$ 。将 0.5625 用乘 2 取整法获得二进制数为  $(0.1001)_2$ ，再将  $(101110000)_2$  和  $(0.1001)_2$  相加，可以得到  $(368.5625)_{10} = (101110000.1001)_2$ 。

总之，各种进制之间的相互转换关系如图 2.1 所示。

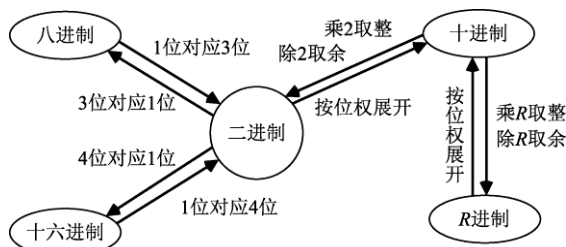


图 2.1 各种进制之间的相互转换关系

## 5. 计算机使用二进制的优点

在计算机中采用二进制数制，而没有使用十进制，是因为二进制有以下几个方面的优点。

(1) 二进制数的状态简单，容易实现。前面已经讲过，二进制中只有 0 和 1 两个状态，很容易用物理器件实现。例如，在计算机中常用电位的“高”和“低”、脉冲的“有”和“无”、晶体管的“导通”和“截止”来表示“1”和“0”。同时，在磁盘中，也用铁氧体磁芯沿不同方向的磁化等来表示“1”和“0”两种状态。在光盘中，用激光是否在光盘上烧制小的凹坑来表示“1”和“0”。显然，这种方式比用其他进制在物理上的实现要简单得多。

(2) 逻辑操作简单。二进制的“1”和“0”两个数码正好与逻辑命题中的两个值“是”和“否”，或者“真”和“假”相对应，而布尔代数中，正好与二进制的操作相对应。因此，用二进制转换为布尔代数来进行研究，为计算机实现逻辑运算和程序中的逻辑判断提供了便利条件，同时也为计算机的逻辑线路设计提供了方便。

(3) 二进制的运算规则简单。在做任何进位制的四则运算时，都需要记住两个整数的



求和及乘积的规则。对于  $R$  进制，需要记住  $R \times (R+1)/2$  个和与积的运算规则，如对于十进制数制需要记住  $10 \times (10+1)/2=55$  个和与积的运算规则(加法表和九九表)。但如采用二进制，由于  $R=2$ ，所以需要记住的运算规则数为  $2 \times (2+1)/2=3$ 。同时，0 和 1 这两个数进行相加或相乘的运算极其简单。

## 2.1.2 数的原码、反码和补码

一个数值数据要在计算机中进行表示，也应该与实际使用中的要求相同。数值数据在计算机中的表示必须明确指明符号表示方法和小数点的位置表示方法。

在计算机中，数值数据的符号表示方法简单，计算机中使用二进制 0 和 1，正好与正号“+”和负号“-”相对应。因此，在计算机中，表示一个数值数据的符号的方法是：占用一位二进制数位，用“0”表示正号，用“1”表示负号。为了区别符号和数值，二进制数值数据在计算机中有原码、反码和补码 3 种表示方法。

### 1. 真值与机器数

机器数是指数在计算机中的表示形式，为了表示普通的数与机器数的对应关系，将普通的数称为机器数的真值。因此，在计算机中只有机器数，不存在数的真值。

例如，两个数  $N_1$  和  $N_2$  的真值分别为：

$$N_1 = +1101010$$

$$N_2 = -1011100$$

则所对应的机器数分别为：

$$N_1: 01101010$$

$$N_2: 11011100$$

### 2. 原码

原码是一种简单的机器数表示法，其符号位用 0 表示正号，用 1 表示负号，数值部分按二进制书写。

其表示方法：对于最左边的符号位，若为正数，则原码符号位为 0；若为负数，则符号位为 1，其余数值位不变，写到符号右边。

整数的原码用公式定义如下：

$$[X]_{\text{原}} = \begin{cases} X & , 0 \leq X < 2^n \\ 2^n - X = 2^n + |X| & , -2^n < X < 0 \end{cases}$$

小数的原码用公式定义如下：

$$[X]_{\text{原}} = \begin{cases} X & , 0 \leq X < 2^n \\ 1 - X = 1 + |X| & , -1^n < X < 0 \end{cases}$$

即  $[X]_{\text{原}} = \text{符号位} + |X|$

|        |                |                           |
|--------|----------------|---------------------------|
| 例 2.11 | $X = +1001,$   | $[X]_{\text{原}} = 01001$  |
|        | $X = -1001,$   | $[X]_{\text{原}} = 11001$  |
|        | $X = +0.1001,$ | $[X]_{\text{原}} = 0.1001$ |



$$X = -0.1001, \quad [X]_{\text{原}} = 1.1001$$

原码的特点: 数的原码与真值之间的关系比较简单, 且与真值的转换也方便。在做乘除法运算时, 可将符号位和数值位分开处理, 运算结果的符号可用参加操作的两个操作数符号进行异或运算求得, 运算结果的数值可由操作数原码的数值部分按乘除规则运算获得。因此, 原码尤其适合于乘除法运算。它的最大缺点是在机器中进行加减法运算时比较复杂。

### 3. 反码

反码就是把二进制数按位求反。如果  $X_i = 1$ , 则反码为  $\overline{X_i} = 0$ ; 如果  $X_i = 0$ , 则反码  $\overline{X_i} = 1$ 。在  $X_i$  上面加一横表示反码的意思。

表示方法: 对于正数, 符号位为 0, 后面的数值位不变; 若为负数, 符号位为 1, 数值位按位求反。

整数的反码用公式定义如下:

$$[X]_{\text{反}} = \begin{cases} X, & 0 \leq X \leq 2^{n-1} - 1 \\ 2^{n-1} - 1 + X & -(2^{n-1} - 1) < X < 0 \end{cases}$$

小数的反码用公式定义如下:

$$[X]_{\text{反}} = \begin{cases} X, & 0 \leq X < 1 \\ 2 - 2^{-(n-1)} + X & -1 < X < 0 \end{cases}$$

即  $[X]_{\text{反}} = (2 - 2^{-n}) \times \text{符号位} + X \bmod (2 - 2^{-n})$ , 其中,  $n$  为小数点后有效位数。

|        |                |                           |
|--------|----------------|---------------------------|
| 例 2.12 | $X = +1001,$   | $[X]_{\text{反}} = 01001$  |
|        | $X = -1001,$   | $[X]_{\text{反}} = 10110$  |
|        | $X = +0.1001,$ | $[X]_{\text{反}} = 0.1001$ |
|        | $X = -0.1001,$ | $[X]_{\text{反}} = 1.0110$ |

反码的特点: 反码进行加减运算时, 若最高位有进位, 则要在最低位加 1, 此时要多进行一次加法运算, 增加了复杂性, 又影响了速度, 因此很少采用。

### 4. 补码

补码是一种很好的机器数表示法, 补码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, 简化了判断过程, 提高了计算机的运算速度, 并相应地节省了设备开销。因此, 补码是应用最广泛的一种机器数表示方法。

其表示方法: 对于正数, 符号位为 0, 后面的数值位不变; 若为负数, 符号位为 1, 数值位按位求反, 然后在最末位加 1。

整数的补码用公式定义如下:

$$[X]_{\text{补}} = \begin{cases} X, & 0 \leq X < 2^n \\ 2^{n+1} + X = 2^{n+1} - |X|, & -1 \leq X < 0 \end{cases}$$

小数的补码用公式定义如下:

$$[X]_{\text{补}} = \begin{cases} X, & 0 \leq X < 1 \\ 2 + X = 2 - |X|, & -1 \leq X < 0 \end{cases}$$



即  $[X]_{\text{补}} = 2 \times \text{符号位} + X \pmod{2}$

**例 2.13**  $X = +1001$ ,  $[X]_{\text{补}} = 01001$   
 $X = -1001$ ,  $[X]_{\text{补}} = 10111$   
 $X = +0.1001$ ,  $[X]_{\text{补}} = 0.1001$   
 $X = -0.1001$ ,  $[X]_{\text{补}} = 1.0111$

补码的特点：与原码相比，补码在正数轴方向上表示数的范围与原码相同，在负数轴方向上表示数的范围比原码增大了一个单位。

### 5. 三种码制的比较

数值数据的原码、反码和补码的表示方法有许多共同点和相异点。

三种码制的共同点如下。

(1) 三种码制主要是解决数值数据的符号在机器中的表示问题。正数的原码、补码和反码都等于真值；而对于负数，表示方法各有不同。

(2) 三种码制中的最高位都表示符号位，其中真值为正数时，符号位用“0”来表示；真值为负数时，符号位用“1”来表示。

三种码制的相异点如下。

(1) 原码的符号位和数值位运算必须分开进行，运算完后再组合到一起，计算上不方便；补码和反码的符号位可以和数值位一起参与运算，方便了计算机的处理。

(2) 原码和反码对于数值零各自都有两种表示方法： $[+0]_{\text{原}} = 00000000\text{B}$ ， $[-0]_{\text{原}} = 10000000\text{B}$  以及  $[+0]_{\text{反}} = 00000000\text{B}$ ， $[-0]_{\text{反}} = 11111111\text{B}$ 。而补码则只有唯一的表示法： $[0]_{\text{补}} = 00000000\text{B}$ ，这使得补码的运算更方便。

(3) 当需要将较短字长的代码向高位扩展为较长字长的代码，或代码向右移位时，原码、补码和反码采用的处理方法不同。原码的方法是符号位固定在最高位，扩展位数值位右移后空出的位填 0；而对于补码和反码的方法，符号位则是固定在最高位，扩展位或数值位右移后空出的位填写“与符号位相同的代码”。

(4) 原码和反码能表示的正数和负数的范围相对于零是对称的。例如，对于整数的表示，都是  $\pm(2^{n-1}-1)$ ；然而补码的负数表示范围比正数表示范围要宽，能够多表示一个最小负数。例如，对于整数的表示，其值等于  $-2^{n-1}$ 。

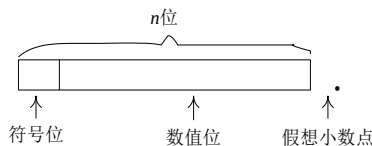
### 2.1.3 定点数和浮点数

对于数值数据的小数点表示方法，在计算机中分为定点数和浮点数两种表示形式。

#### 1. 定点数

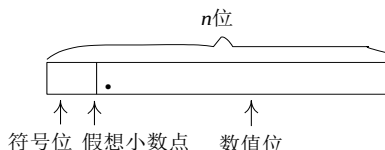
定点数是指计算机在运算过程中，数据中小数点的位置固定不变。小数点的位置是由计算机设计者在机器的结构中指定一个不变的位置，而不一定都必须具有小数点的指示装置。定点数一般有小数和整数两种表示形式。

定点整数是指所表示的数都为整数，而小数点则固定在数值位最低位之后，其格式为



即定点整数用  $n$  位二进制表示一个数，一般选择最左边一位作为符号位。若该位为 1，则表示该数为负数；若该位为 0，则表示该数为正数。由于小数点在最低位之后，最低位的位权值为  $2^0$ ，因此它所表示的数都为整数。

定点小数是指所表示的数都为小数，它的小数点固定在符号位与最左边数值位之间，其格式为



定点小数的小数点位置在数值位的最左端，即在符号位之后。由于小数点右边各位的位权值分别为： $2^{-1}, 2^{-2}, \dots$ ，因此它所表示的数只能是小数。

如果计算机采用定点整数表示，则要求参加运算的数都是大于或等于 1 的数，即都为整数。如果参加运算的数有小数部分，则在送入计算机运算之前，要乘以一个大于 1 的比例因子，将其放大后变为整数。如果计算机采用定点小数表示，则参与运算的数都是小于 1 的数。如果参加运算的数有整数部分，在送入计算机运算之前，要乘以一个小于 1 的比例因子，将其缩小后变为小数。

无论是定点小数还是定点整数，由于小数点始终固定在一个确定的位置，所以计算机在运算时不必对位，可以直接进行加减运算。因此，对于参与运算的数值数据本身就是定点数形式时，计算简单，而且方便。虽然定点数表示容易实现，但需要对参加运算的数进行比例因子的计算，这就增加了额外的计算量。

## 2. 浮点数

在科学计算和数据处理中，经常需要处理和计算非常大或者非常小的数值。定点表示法表示的数值有范围大小的限制，不能够精确地完成这种数值的表示。为了表示更大取值范围的数，可以采用如下所示的表示方法：

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

式中， $N$  为浮点数； $M$  和  $E$  为带符号的定点数， $E$  为阶码； $R$  称为“阶的基数(或底)”。

尾数  $M$  可用原码、反码和补码三种码制中的任意一种来表示，可以是整数，也可以是小数。目前大多数计算机都把  $M$  规定为纯小数，阶码  $E$  规定为纯整数。 $M$  中的小数点可以随  $E$  值的变化而左右浮动。所以把这种表示法叫作数的浮点表示法，而这样的数又称为浮点数。

**例 2.14** 十进制数 0.00012345 可以写成：

$$\begin{aligned} F &= 0.00012345 \\ &= 0.00000012345 \times 10^{+3} \end{aligned}$$



$$= 0.12345 \times 10^{-3}$$

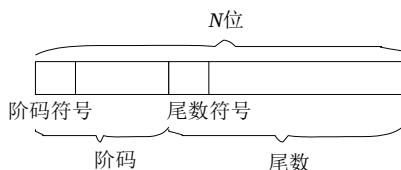
虽然它们的数字部分都是 0.00012345，但因幂次不同，所表示的数值差异也很大。按同样的道理，二进制数也可以写为以下形式：

$$011011 = 0.11011 \times 2^{+101}$$

$$0110.11 = 0.11011 \times 2^{+011}$$

$$0.0011011 = 0.11011 \times 2^{-010}$$

机器中浮点数的表示格式分为阶码和尾数两部分，其中阶码一般用定点整数表示，尾数用定点小数表示。在计算机中的表示格式如下：



其中：阶码占  $m$  位(阶码符号占 1 位)，尾数占  $n$  位(尾数符号占 1 位)，阶码和尾数共占  $N=m+n$  位。

**例 2.15** 要表示真值  $x=(38)_{10}$ ，对某一机器，用  $N=12$  位二进制代码表示一个浮点数，阶码为  $m=4$  位，尾数为  $n=8$  位，则其浮点数的表示如下。

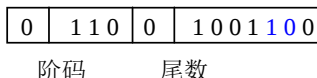
$$\begin{aligned} \text{若 } x &= (38)_{10} = (100110)_2 \\ &= (0.1001100) \times 2^{+110} \end{aligned}$$

基数  $R$  为 2，其基数都为 2。

阶码：+110

尾数：+0.1001100

因此，在机器中如以原码表示阶码和尾数，则有  $x$  的浮点表示为



考虑到表示方便，在阶码与尾数之间用空格分开，即  $x$  的浮点表示形式为

0110 0.1001100

### 2.1.4 十进制数的编码

计算机内部只处理二进制数据，但是人们都习惯使用十进制数据。因此，在输入时要将十进制数转换成二进制数，在输出时再将二进制数转换成十进制数。常用的十进制数的编码方法有 BCD(Binary Coded Decimal)码、余 3 码、格雷码等。下面简要介绍 BCD 码。

BCD 码是二-十进制编码，具有二进制的形式，又具有十进制数的特点。BCD 码是用 4 位二进制数的组合代表十进制数的 0、1、2、3、4、5、6、7、8、9 这 10 个数符。4 位二进制数有 16 种组合，原则上可任选其中的 10 种作为代码，分别代表十进制制中的 10 个数符。最常用的 BCD 码称为 8421BCD 码，8、4、2、1 分别是 4 位二进制数的位权，每位都是固定位权的，因此又称为有权码。



## 1. 8421BCD 码与十进制数的转换

十进制数转换成 8421BCD 码的方法很简单,将每 1 位十进制数用 4 位二进制数码表示即可。

**例 2.16** 将十进制数 578.43 转换为 8421BCD 码。

578.43=(0101 0111 1000.0100 0011)BCD。

8421BCD 码转换成十进制数就是将每 4 位二进制数用 1 位十进制数表示。

**例 2.17** 将 8421BCD 码 1001 1001 0111.0011 0110 转换为十进制数。

(1001 1001 0111.0011 0110)BCD=997.36。

## 2. 8421BCD 码的格式

8421BCD 码分为压缩的 8421BCD 码(也叫组合的 BCD 码)和非压缩 8421BCD 码(也叫非组合的 BCD 码、分离的 BCD 码)。

(1) 压缩 8421BCD 码。每一位十进制数用 4 位二进制数来表示,即一个字节表示 2 位十进制数。

例如,压缩 BCD 码(01100101)BCD,表示为十进制数为 65。

(2) 非压缩 8421BCD 码。每一位十进制数用 8 位二进制数来表示,即一个字节表示 1 位十进制数,且只用每个字节的低 4 位来表示 0~9,高 4 位为不确定的数码。

例如,十进制数 65,用非压缩 BCD 码表示为 (---- 0110 ---- 0101)BCD,其中“-”为不确定的数码。

## 2.2 非数值数据

在计算机中,除了可以对数值数据进行处理外,还能处理非数值数据,非数值数据有字符、声音、图形、图像等数据信息。由于计算机只处理二进制编码形式的数据,因此非数值数据都必须转换为二进制表示的形式才能提供给计算机进行处理。

### 2.2.1 文字信息的编码

文字信息处理是指对语言文字符号系统进行转换、传输、存储、加工、复制等处理的一种技术,是计算机信息技术的重要应用领域。

在计算机中进行文字处理的主要工作有以下几种。

(1) 语言文字信息的输入(获取)。主要包括由键盘实现的字符代码输入、由语音识别方法实现的语音输入、由图像识别方法实现的书面文字输入等。

(2) 语言文字信息的传输(通信)。通过数据传输设备(由调制解调器、编译/译码器和传输线等组成)传送数字信息。

(3) 语言文字信息的加工。主要利用计算机作为信息加工设备,使用软件和接口设备进行对信息的加工。其操作主要包括筛选、编排、分析、存储、翻译、还原等。

(4) 语言文字信息的输出。即文字资料的显示或复制。

文字信息根据不同国家使用的文字不同进行分类。在我国常用的主要是英文字符和中文字符两种形式。



### 1. 英文字符的编码

在计算机中，对非数值的文字和其他符号进行处理时，要对文字和符号进行数字化处理，即采用二进制编码来表示文字和符号。字符编码就是规定用二进制数表示文字和符号的方法。目前，通用的西文字符编码方法主要为 ASCII 码(American Standard Code for Information Interchange, 美国国家信息交换标准代码)，已被国际标准化组织(International Organization for Standardization, ISO)批准为国际标准，称为 ISO 646 标准。它适用于所有的拉丁文字字母，已在全世界通用。我国相应的国家标准是 GB(Guóbìāo) 1988(称为信息处理交换用的 7 位编码字符集标准)。ASCII 码分为 7 位和 8 位两种版本。常用的是 7 位二进制编码形式的 ASCII 编码表，如表 2.4 所示。

表 2.4 7 位 ASCII 码表

| <div><div>D<sub>6</sub> D<sub>5</sub> D<sub>4</sub></div><div>D<sub>3</sub> D<sub>2</sub> D<sub>1</sub> D<sub>0</sub></div></div> | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|-----------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| 0000                                                                                                                              | NUL | DLE | SP  | 0   | @   | P   | '   | P   |
| 0001                                                                                                                              | SOH | DC1 | !   | 1   | A   | Q   | a   | Q   |
| 0010                                                                                                                              | STX | DC2 | "   | 2   | B   | R   | b   | R   |
| 0011                                                                                                                              | ETX | DC3 | #   | 3   | C   | S   | c   | S   |
| 0100                                                                                                                              | EOT | DC4 | \$  | 4   | D   | T   | d   | T   |
| 0101                                                                                                                              | ENQ | NAK | %   | 5   | E   | U   | e   | U   |
| 0110                                                                                                                              | ACK | SYN | &   | 6   | F   | V   | f   | V   |
| 0111                                                                                                                              | BEL | ETB | ,   | 7   | G   | W   | g   | W   |
| 1000                                                                                                                              | BS  | CAN | (   | 8   | H   | X   | h   | X   |
| 1001                                                                                                                              | HT  | EM  | )   | 9   | I   | Y   | i   | Y   |
| 1010                                                                                                                              | LF  | SUB | *   | :   | J   | Z   | j   | Z   |
| 1011                                                                                                                              | VT  | ESC | +   | ;   | K   | [   | k   | {   |
| 1100                                                                                                                              | FF  | FS  | ,   | <   | L   | \   | l   |     |
| 1101                                                                                                                              | CR  | GS  | -   | =   | M   | ]   | m   | }   |
| 1110                                                                                                                              | SO  | RS  | .   | >   | N   | ↑   | n   | ~   |
| 1111                                                                                                                              | SI  | US  | /   | ?   | O   | ←   | o   | DEL |

7 位 ASCII 码的每个字符都由 7 个二进制位 D<sub>6</sub>, D<sub>5</sub>, D<sub>4</sub>, D<sub>3</sub>, D<sub>2</sub>, D<sub>1</sub>, D<sub>0</sub> 表示，总共可以组成 128 种编码。因此，最多可表示 128 种字符，其中包括 10 个数字、26 个小写字母、26 个大写字母以及各种运算符号和标点符号等。

ASCII 码表中第 000~001 列和最右下角字符，共 33 个字符，其编码值为 0~31 和 127，是不可显示或打印的。它们是控制码，用来控制计算机某些外围设备的工作特性和某些计算机软件的运行情况。例如，CR 称为回车字符，编码为 00001101，是使显示器光标或打印机换行的控制字符。

表 2.4 中第 010~111 列(共 6 列)中共有 94 个可打印或显示的字符，称为图形字符。图形字符有确定的结构形状，可在打印机和显示器等输出设备上输出。同时，这些字符均可





在计算机键盘上找到相应的键,按键后就可以将相应字符的二进制编码输入计算机内。

随着计算机应用的发展和深入,7位的字符集已显得不够用。为此,国际标准化组织又制定了ISO 2022标准《7位字符集的代码扩充技术》。它在保持与ISO 646兼容的基础上,规定了扩充ASCII字符集8位代码的方法。当最高位D<sub>7</sub>置0时,称为基本ASCII码(编码同7位ASCII码)。当最高位置1时,形成扩充ASCII码,它表示数值的范围为128~255,可表示128个字符。

### 2. 汉字字符的编码

为适应计算机处理汉字信息的需要,我国于1981年发布了《信息处理交换用汉字编码字符集——基本集》,即国家标准GB2312。它是根据GB2311的代码扩充方法制定的汉字交换编码标准。它以94个可显示的ASCII码字符为基集,由两个字节构成,每个字节为7位二进制码。GB2312图形字符构成一个二维平面,它分成94行、94列,行号称为区号,列号称为位号。每一个汉字或符号在二维平面的码表中都各自唯一对应一个区号和位号。区号在左,位号在右,区号和位号合在一起构成4位十进制数码,叫作该字的区位码。该标准规定了进行一般汉字信息处理交换用的6763个汉字和682个非汉字图形字符的代码,合计7445个。

为了与国际标准一致,汉字又用国标码来表示。每个汉字的区号和位号上必须加上32(十进制数)之后再转换成十六进制数形式,叫作汉字的国标码。国标码用两个字节的十六进制数表示,如“中华人民共和国”的国标码分别是“5650H, 3B2AH, 484BH, 4371H, 3932H, 3A4DH, 397AH”。

### 2.2.2 声音的编码

随着多媒体技术的出现,音频数据在计算机中的处理与存储成为现实。声音等非字符信息也是通过数值化的方法在计算机里进行表示的。复杂的声波由许多具有不同振幅和频率的正弦波组成,这些连续的模拟量不能由计算机直接处理,必须将其数字化后才能被计算机所识别和处理。计算机获取声音信息的过程即是声音信号数字化的处理过程。声音被计算机处理主要经过音频信号的采样、量化和编码等过程,如图2.2所示。

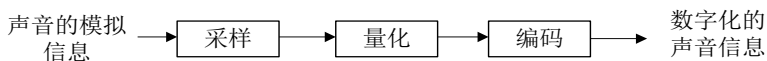


图 2.2 声音信息的数字化过程

用数字方式记录声音,首先是对声波进行采样。采样以一定的频率进行,即采样频率,以Hz为单位。如果提高采样频率,单位时间内所得到的振幅值就增多,即采样频率越高对原声音曲线的模拟就越精确,但所需的存储空间也就越大。在计算机中,存储声音的文件格式有很多,常用的声音文件扩展名有wav、au、voc、mp3等。

### 2.2.3 图形数据的编码

随着信息技术的发展,越来越多的图形信息要求用计算机来存储和处理。在计算机中,有两种不同的图形编码方式,即位图编码方法和矢量编码方法。不同的编码方式会影响图像的质量、存储图像的空间大小、图像传送的时间和修改图像的难易程度。



图像的存储方式最直接的就是点阵方式，点阵即点的阵列，阵列中的点称为像素。图像中的像素越多，能表示的细节(如物体)也就越多；每个像素的表示范围越大，能表示的细节(如颜色、灰度)也就越多。非字符信息有众多的表示方式，比如图像，对于不同的应用环境，就要使用不同的表示方法，以取得最佳结果。目前常用的图像存储方式有很多，如 GIF(Graphics Interchange Format)和 JPEG(Joint Photographic Experts Group)等格式，这些格式的优点是压缩率高，因此传输速率高。

## 2.3 数据的机器编码

计算机内部的信息分为两大类，即控制信息和数据信息。控制信息是指指挥计算机如何操作的指令；数据信息是计算机加工的对象。

指令是规定计算机执行某种操作的指示和命令，又称为机器指令。计算机的工作基本上体现为执行指令，一台计算机的所有指令的集合构成该计算机的指令系统。从设计计算机的观点看，机器指令系统提出了对 CPU(Central Processing Unit)的功能要求。CPU 任务的大部分实现都涉及机器指令系统的实现，所以指令系统与计算机的硬件结构紧密相关。从用户角度看，选取机器语言(实际上是汇编语言)的用户，必须要熟悉所用机器的指令系统，熟悉机器所直接支持的寄存器和存储器结构、数据结构以及算术逻辑运算单元(Arithmetic and Logic Unit, ALU)功能。因此，指令系统的设计是计算机系统设计的一个最有影响的方面，是计算机设计人员和编程人员能看到的同一机器的分界面。

### 2.3.1 数据的机器级编码

计算机是通过执行指令序列来解决问题的。因此，每种计算机都有一组指令集供给用户使用。计算机中的指令由操作码字段和操作数字段两部分组成。操作码字段指明计算机所要执行的操作，而操作数字段则指明在指令执行操作的过程中所需要的操作数。例如，乘法指令除需要指定做乘法操作外，还需要提供被乘数与乘数。

计算机只能识别二进制代码，所以机器指令是由二进制代码组成的。机器指令由操作码和地址码组成，操作码规定 CPU 执行什么操作；地址码指明源操作数从哪里取，结果送往什么地方以及下一条指令从哪里取。因此，机器指令的基本格式表示为

|     |     |
|-----|-----|
| 操作码 | 地址码 |
|-----|-----|

由于机器指令与 CPU 紧密相关，所以不同种类的 CPU 所对应的机器指令也就不同，而且它们的指令系统往往相差很大。但对同一系列的 CPU 来说，为了使各型号之间具有良好的兼容性，应做到新一代 CPU 的指令系统必须包括先前同系列 CPU 的指令系统。只有这样，先前开发出来的各类程序在新一代 CPU 上才能正常运行。

机器语言用来直接描述机器指令、使用机器指令的规则等，是 CPU 能直接识别的唯一语言。用机器语言编写的程序不易读、出错率高、难以维护，也不能直观地反映用计算机解决问题的基本思路。



## 2.3.2 数据的汇编级编码

为便于使用和记忆,用汇编语言来代替机器指令进行程序编写。汇编语言是一种符号语言,它用助记符来表示操作码,用符号或符号地址来表示操作数或操作数地址,它与机器指令是一一对应的。但计算机只能识别机器指令。因此,用汇编指令编写的程序必须首先转换为机器指令才能在计算机上执行。

汇编语言是一种介于机器语言和高级语言之间的计算机编程语言,它既不像机器语言那样直接使用计算机所认识 and 理解的二进制代码来构成,也不像高级语言那样独立于机器之外直接面向用户。

用汇编语言编写的程序叫汇编语言程序,汇编语言的源代码是用很像英文缩写的助记符编写而成的,因此,较适合英语为第一语言的国家使用。将汇编语言翻译成机器语言需要用到汇编程序。

用汇编语言编写的源程序能直接使用计算机的硬件,因此能编制出高质量的程序,用汇编语言编写的程序要比用与它等效的高级语言程序生成的目的代码精简得多,占用的内存存储器空间少,执行的速度快。用汇编语言编写程序,最主要的缺点是所编写的程序与所要解决的问题的数学模型之间的关系不直观,使得编制程序的难度加大,而且编写的程序可读性也差,导致出错的可能性增加,因此程序设计和调试的时间较长。同时,汇编语言程序在不同机器间的可移植性也较差。

## 2.4 数字逻辑与数字系统

在计算机中,所有的数据表示与运算都是用二进制数进行的,处理二进制数的基本电路是逻辑门。早期的逻辑门是由分立元件构成的,随着集成电路技术的发展,如今逻辑门均已集成化而成为集成逻辑门。

### 2.4.1 基本逻辑关系及逻辑门

逻辑门是构成数字电路的基本单元,每一种逻辑门的输入和输出之间都有一定的逻辑关系。所有的逻辑关系都可以由“与”“或”“非”三种基本的逻辑关系来表示,而实现这些基本逻辑关系的电路就是逻辑门。最基本的逻辑门是“与”门、“或”门和“非”门。

#### 1. “与”逻辑关系及“与”门

“与”逻辑关系又称为逻辑乘,运算符号是“ $\cdot$ ”“ $\wedge$ ”“ $\cap$ ”或“AND”。“与”逻辑关系可以用串联开关电路来说明,如图2.3所示,灯亮的条件是开关A和B同时连通,否则灯不会亮。

“与”运算、“与”逻辑关系或说“与”门的运算规则可表示为:只有当所有的输入都为1时,输出才为1。换言之,当输入中有一个不为1时,输出就为0。

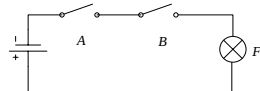


图 2.3 串联开关电路



“与”逻辑关系用逻辑函数表示为

$$F=A B \tag{2.1}$$

读作“ $F$  等于  $A$  与  $B$ ”，符号“ $\cdot$ ”读为“与”(注意：不能读为“乘”)。式(2.1)可省略为  $F=AB$ 。

“与”运算的真值表如表 2.5 所示。

表 2.5 “与”运算的真值表

| $A$ | $B$ | $F=A \cdot B$ |
|-----|-----|---------------|
| 0   | 0   | 0             |
| 0   | 1   | 0             |
| 1   | 0   | 0             |
| 1   | 1   | 1             |

在表 2.5 中，设开关接通为“1”，断开为“0”；灯亮为“1”，灯灭为“0”。在数字电路中，实现“与”运算的电路称为“与”门，“与”门的逻辑符号如图 2.4 所示。

2. “或”逻辑关系及“或”门

“或”逻辑关系又称为逻辑加，运算符号是“+”“ $\vee$ ”“ $\cup$ ”或“OR”。“或”逻辑关系可以用并联开关电路来说明，如图 2.5 所示，灯亮的条件是开关  $A$  和  $B$  只要有一个连通即可，只有当开关  $A$  和  $B$  都不接通时，灯才会不亮。



图 2.4 “与”门的逻辑符号

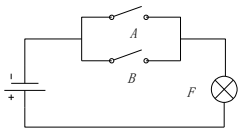


图 2.5 并联开关电路

“或”运算、“或”逻辑关系或说“或”门的运算规则可表示为：只有当所有的输入都为 0 时，输出才为 0。换言之，当输入中有一个不为 0 时，输出就为 1。

“或”逻辑关系用逻辑函数表示为

$$F=A+B$$

读作“ $F$  等于  $A$  或  $B$ ”，符号“+”为“或”(注意：不能读为“加”)。“或”运算的真值表如表 2.6 所示。

表 2.6 “或”运算的真值表

| $A$ | $B$ | $F=A+B$ |
|-----|-----|---------|
| 0   | 0   | 0       |
| 0   | 1   | 1       |
| 1   | 0   | 1       |
| 1   | 1   | 1       |





在数字电路中，实现“或”运算的电路称为“或”门，“或”门的逻辑符号如图 2.6 所示。



图 2.6 “或”门的逻辑符号

### 3. “非”逻辑关系及“非”门

“非”逻辑关系又称为反相运算，又叫逻辑否定。“非”运算、“非”逻辑关系或说“非”门的运算规则可表示为：当输入为 1 时，输出为 0；当输入为 0 时，输出为 1。“非”逻辑关系用逻辑函数表示为

$$F = \bar{A}$$

读作“ $F$  等于  $A$  的非”。“非”运算的真值表如表 2.7 所示。

表 2.7 “非”运算的真值表

| $A$ | $F = \bar{A}$ |
|-----|---------------|
| 0   | 1             |
| 1   | 0             |

在数字电路中，实现“非”运算的电路称为“非”门，“非”门的逻辑符号如图 2.7 所示。

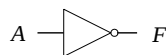


图 2.7 “非”门的逻辑符号

### 4. “异或”逻辑关系及“异或”门

“异或”运算、“异或”逻辑关系或说“异或”门的运算规则可表示为：当输入的两个信号相异时，输出为 1，否则输出为 0。“异或”逻辑关系用逻辑函数表示为

$$F = \bar{A}B + A\bar{B} = A \oplus B$$

读作“ $F$  等于  $A$  异或  $B$ ”，符号“ $\oplus$ ”读为“异或”。“异或”运算的真值表如表 2.8 所示。

表 2.8 “异或”运算的真值表

| $A$ | $B$ | $F = A \oplus B$ |
|-----|-----|------------------|
| 0   | 0   | 0                |
| 0   | 1   | 1                |
| 1   | 0   | 1                |
| 1   | 1   | 0                |

在数字电路中，实现“异或”运算的电路称为“异或”门，“异或”门的逻辑符号如图 2.8 所示。



图 2.8 “异或”门的逻辑符号

## 2.4.2 逻辑代数与逻辑函数

逻辑代数又称为布尔代数，是 19 世纪由英国数学家布尔创立的。1938 年，香农将布尔代数直接应用于开关电路，因此布尔代数也称为开关代数。现在布尔代数广泛地应用于数字电路的分析与设计，成为数字逻辑的主要数学工具。



赋予逻辑属性值真或假的变量称为全通逻辑变量；描述逻辑变量关系的函数称为逻辑函数；实现逻辑函数的电路称为逻辑电路；用逻辑电路做成计算机系统中常用的部件，称为逻辑部件。

### 1. 逻辑变量与函数

逻辑代数与普通代数一样，用字母  $A$ 、 $B$ 、 $C$ 、 $\dots$  表示变量，称为逻辑变量。但是逻辑变量的取值只有两种状态：0 或 1。这两种状态可以表示电路中开关的接通与断开、脉冲的有或无、电位的高或低。因此，与普通代数不同，逻辑变量的取值 0 或 1 并没有数量的概念。

逻辑函数的概念与普通代数相似，区别只是逻辑变量和函数的取值只有 0 和 1 两种情况。因此，逻辑函数相对于普通代数来说更简单。逻辑函数是由逻辑变量  $A$ 、 $B$ 、 $C$ 、 $\dots$  和算子 “ $\cdot$ ” “ $+$ ” “ $-$ ” 及括号、等号等构成的一个表达式，例如：

$$F = A + B$$

$$G = A \cdot B$$

### 2. 基本逻辑运算

逻辑代数中最基本的逻辑运算有“与”运算、“或”运算、“非”运算，可以组成其他各种复杂的逻辑运算，并且可以由逻辑电路来实现。因此，在研究计算机的硬件组成时，将数字电路转换为数字逻辑来加以研究，可使问题变得更加方便和简单。

### 3. 逻辑代数的定理及常用公式

逻辑代数和其他任何推理数学系统一样，有公理系统和基本定理两部分。

#### 1) 公理系统

为了证明定理和推导公式方便，任何代数系统都必须确定一套公理系统。

逻辑代数是一个封闭的代数系统，它由逻辑变量集、0 和 1 以及“与”“或”“非”运算组成。对于任意逻辑变量  $A$ 、 $B$ 、 $C$ ，逻辑代数系统应满足下列公理。

|      |       |                                           |                                           |
|------|-------|-------------------------------------------|-------------------------------------------|
| 公理 1 | 交换律   | $A + B = B + A$                           | $AB = BA$                                 |
| 公理 2 | 结合律   | $(A + B) + C = A + (B + C)$               | $(AB)C = A(BC)$                           |
| 公理 3 | 分配律   | $A + (B \cdot C) = (A + B) \cdot (A + C)$ | $A \cdot (B + C) = A \cdot B + A \cdot C$ |
| 公理 4 | 0-1 律 | $A + 0 = A$                               | $A \cdot 1 = A$                           |
|      |       | $A + 1 = 1$                               | $A \cdot 0 = 0$                           |
| 公理 5 | 互补律   | $A + \bar{A} = 1$                         | $\bar{A} \cdot A = 0$                     |

在公理确定之后，就可以用它们证明定理、推导公式。在逻辑代数中，由于每个公理的函数表达式都是成对出现的，所以由公理推导出来的定理的函数表达式也是成对的。

#### 2) 基本定理

根据逻辑代数的公理，可以推导出逻辑代数的许多基本定理。

|      |                 |                 |                 |                 |
|------|-----------------|-----------------|-----------------|-----------------|
| 定理 1 | $0 + 0 = 0$     | $1 + 0 = 1$     | $0 + 1 = 1$     | $1 + 1 = 1$     |
|      | $0 \cdot 0 = 0$ | $1 \cdot 0 = 0$ | $0 \cdot 1 = 0$ | $1 \cdot 1 = 1$ |
| 推论   | $\bar{1} = 0$   | $\bar{0} = 1$   |                 |                 |
| 定理 2 | $A + A = A$     | $AA = A$        |                 |                 |



- 定理3  $A + AB = A$   $A(A + B) = A$   
 定理4  $A + \bar{A} \cdot B = A + B$   $A \cdot (\bar{A} + B) = A \cdot B$   
 定理5  $\bar{\bar{A}} = A$   
 定理6  $\overline{A + B} = \bar{A} \cdot \bar{B}$   $\overline{A \cdot B} = \bar{A} + \bar{B}$   
 定理7  $A \cdot B + A \cdot \bar{B} = A$   $(A + B) \cdot (A + \bar{B}) = A$   
 定理8  $A \cdot B + \bar{A} \cdot C + B \cdot C = A \cdot B + \bar{A} \cdot C$

### 3) 逻辑代数的重要规则

逻辑代数有三条重要规则, 即对偶规则、反演规则和代入规则, 这些规则在逻辑运算中十分有用。

(1) 对偶规则。如果将逻辑函数表达式  $F$  中所有的 “ $\cdot$ ” 变成 “ $+$ ”, “ $+$ ” 变成 “ $\cdot$ ”, “0” 变成 “1”, “1” 变成 “0”, 而逻辑变量保持不变, 则所得到的新逻辑函数表达式称为原函数  $F$  的对偶式, 记作  $F'$ 。例如:

$$F = \bar{A}C + B\bar{D} \quad F' = (\bar{A} + C) \cdot (B + \bar{D})$$

由上面的例子可以看出, 如果  $F$  的对偶式是  $F'$ , 那么  $F'$  的对偶式就是  $F$ 。换言之,  $F$  和  $F'$  互为对偶式。求某一函数表达式的对偶式时, 要保持原函数的运算顺序不变。

若两个逻辑函数表达式  $F$  和  $G$  相等, 则其对偶式  $F'$  和  $G'$  也相等, 这一规则称为对偶规则。利用对偶规则可以使定理、公式的证明减少一半。

(2) 反演规则。如果将逻辑函数表达式  $F$  中所有的 “ $\cdot$ ” 变成 “ $+$ ”, “ $+$ ” 变成 “ $\cdot$ ”, “0” 变成 “1”, “1” 变成 “0”, 原变量变成反变量, 反变量变成原变量, 则所得到的新逻辑函数表达式称为原函数  $F$  的反函数  $\bar{F}$ , 这一规则称为反演规则。

例如, 已知函数  $F = \bar{A}B + C\bar{D}$ , 根据反演规则可得到:

$$\bar{F} = (A + \bar{B}) \cdot (\bar{C} + D)$$

使用反演规则时, 应注意保持原函数式中运算符号的优先顺序不变, 并且两个以上变量的公用 “非” 号保持不变。

(3) 代入规则。代入规则是指任何一个含有变量  $A$  的逻辑表达式, 如果将所有出现  $A$  的位置都代之以同一个逻辑函数  $F$ , 则等式仍然成立, 这个规则称为代入规则。

代入规则的正确性是显然的, 因为任何逻辑代数都和逻辑变量一样, 只有 0 和 1 两种可能的取值。在使用代入规则时需将等式中所有出现同一变量的地方均以同一函数代替, 否则, 代入后的等式就不成立。

代入规则在推导公式时有重要意义, 利用这条规则可以将逻辑代数公理、定理中的变量用任意函数代替, 从而推导出更多等式, 这些等式可直接当作公式使用, 无须另加证明。

## 4. 逻辑表达式的化简

一个逻辑函数可以有多种不同的表达式, 实现这些表达式的逻辑线路也有许多种。在数字系统中, 实现某一逻辑功能的逻辑电路的复杂性与描述该功能的逻辑函数表达式的复杂性直接相关。一般来说, 逻辑函数表达式越简单, 其对应的逻辑电路也就越简单。对于某一个逻辑函数来说, 其函数表达式的形式可以不相同, 但它们所表示的逻辑功能却是相同的。然而, 通过逻辑问题概括出来的逻辑函数往往不是最简的, 为了减小复杂度、提高可靠性、降低成本, 必须对逻辑函数进行化简, 求得最简的逻辑函数表达式。通常, 把逻



辑函数化简成最简形式，称为逻辑函数的最小化。

通常，化简逻辑函数采用代数化简法和卡诺图化简法等方法。

代数化简法就是运用逻辑代数的公理、定理和规则对逻辑函数进行化简。这种方法没有固定的步骤可以遵循，主要取决于对逻辑代数的公理、定理和规则的熟练运用程度。尽管如此，仍然可以总结出一些适用于大多数情况的方法，以便对逻辑函数进行变换，得到最简的函数表达式，如合并项法、吸收法、配项法、消去法等。

$$\begin{aligned}\text{例 2.18} \quad AB + \overline{A}C + \overline{B}C &= AB + (\overline{A} + \overline{B})C \\ &= AB + \overline{ABC} \\ &= AB + C\end{aligned}$$

利用公理和定理来化简公式是一个技巧性较强的工作，只有熟练地掌握公式，才能较好地完成公式的化简运算。

卡诺图是一种由  $2^n$  个方格构成的图形，每个方格表示逻辑函数的最小项，所有的最小项巧妙地排列成一个方格阵列，能清楚地反映它们的相邻关系。由于任何布尔函数都可以表示成“最小项之和”的形式，因此，一个逻辑函数可由卡诺图中若干方格构成的区域来表示，并且这些方格与包含在函数中的各个最小项相对应。具体的卡诺图化简方法可参考数字逻辑等相关的书。

## 2.5 计算机系统的组成

自 1946 年诞生第一台计算机以来，经过多年的努力，计算机技术已有了很大发展。1981 年 8 月 IBM 公司推出了 IBM PC 微型计算机，这是一种面向个人用户的微型计算机，所以又称 PC(Personal Computer)。

### 2.5.1 图灵模型

#### 1. 图灵的基本思想

图灵的基本思想是用机器来模拟人们用纸笔进行数学运算的过程，他把这样的过程看作下列两种简单的动作：在纸上写上或擦除某个符号；把注意力从纸的一个位置移动到另一个位置。而在每个阶段，人要决定下一步的动作，依赖于此人当前所关注的纸上某个位置的符号和此人当前思维的状态。

这个机器的每一部分都是有限的，但它有一个潜在的无限长的纸带。因此，这种机器只是一个理想的设备。图灵认为这样的一台机器就能模拟人类所能进行的任何计算过程。

#### 2. 图灵机的变体

图灵机有很多变种，但可以证明这些变种的计算能力都是等价的，即它们识别同样的语言类。证明两个计算模型  $A$  和  $B$  的计算能力等价的基本思想是：用  $A$  和  $B$  相互模拟，若  $A$  可模拟  $B$ ，且  $B$  可模拟  $A$ ，显然它们的计算能力等价。这里不考虑计算的效率，只考虑计算理论上的“可行性”。



## 2.5.2 冯·诺依曼模型

现代计算机的基本结构是由冯·诺依曼提出的,迄今为止,所有已实用化的计算机都是按冯·诺依曼提出的结构体系和工作原理设计制造的,故又统称为“冯·诺依曼型计算机”。其特点如下。

- (1) 计算机完成的任务是由事先编好的程序完成的。
- (2) 计算机的程序被事先输入存储器中,程序运算的结果也被存放在存储器中。
- (3) 计算机能自动连续地完成程序。
- (4) 程序运行所需要的信息和结果可以通过输入/输出设备完成。
- (5) 计算机由运算器、控制器、存储器、输入和输出设备组成,其结构如图 2.9 所示。

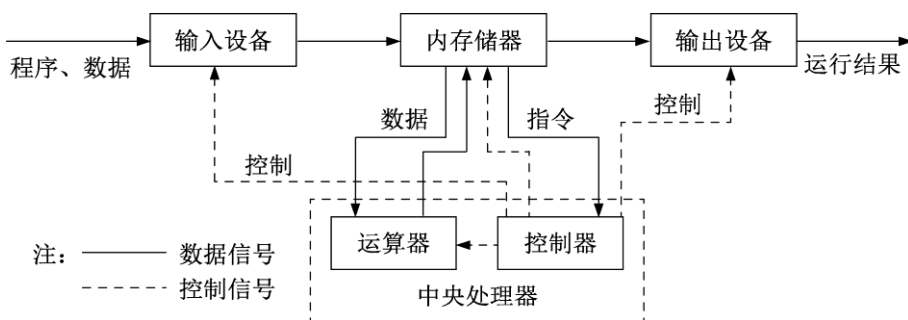


图 2.9 计算机的部件

① 运算器。运算器又称算术逻辑单元(Arithmetic Logic Unit, ALU),是计算机对数据进行加工处理的部件,它的主要功能是对二进制数进行加、减、乘、除等算术运算和与、或、非等基本逻辑运算,实现逻辑判断。运算器是在控制器的控制之下实现其功能的,运算结果由控制器发出的指令送到内存中。

② 控制器。控制器主要由指令寄存器、译码器、程序计数器和操作控制器等组成。控制器用来控制计算机各部件协调工作,并使整个处理过程有条不紊地进行。它的基本功能就是从内存中取出指令和执行指令,即控制器按照程序计数器给出的指令地址从内存中取出该指令进行译码,然后根据该指令功能向有关部件发出控制命令,并执行该指令。另外,控制器在工作过程中,还要接收各部件反馈回来的信息。

通常把运算器、控制器集成在一个大规模的集成电路板上,称为中央处理器,又称 CPU(Central Processing Unit)。成立于 1968 年的英特尔公司(Integrated Electronics Corporation)是全球最大的个人计算机零件和 CPU 制造商,1971 年,英特尔推出了全球第一个微处理器。微处理器所带来的计算机和互联网革命,改变了整个世界。AMD(Advanced Micro Devices, 美国超微半导体公司)专门为计算机、通信和消费电子行业设计和制造各种创新的微处理器(CPU、GPU、APU、主板芯片组、电视卡芯片等)、闪存和低功率处理器解决方案。

③ 存储器。存储器是计算机的记忆装置,用于存放原始数据、中间数据、最终结果和处理程序。为了对存储的信息进行管理,把存储器划分成存储单元,每个单元的编号称为该单元的地址。各种存储器基本上都是以 1 个字节作为一个存储单元。存储器内的信息是按地址存取的。若要访问存储器中的某个信息,就必须知道其地址。向存储器里存入信息

也称为“写入”，新写入的内容将覆盖原来的内容。从存储器里取出信息也称为“读出”，信息读出后并不破坏原来存储的内容，因此信息可以重复读出、多次利用。

通常把内存储器、运算器和控制器合称为计算机主机，也可以说主机是由 CPU 与内存储器组成的，而主机以外的装置称为外部设备。外部设备包括输入/输出设备、外存储器等。

④ 输入和输出设备。输入和输出设备简称 I/O(Input/Output)设备。用户通过输入设备将程序和数据输入计算机，输出设备将计算机处理的结果(如数字、字母、符号和图形)显示或打印出来。常用的输入设备有键盘、鼠标、扫描仪等；常用的输出设备有显示器、打印机和绘图仪等。

### 2.5.3 计算机系统的组成

计算机系统包括硬件(Hardware)系统和软件(Software)系统两大部分，其基本组成如图 2.10 所示。硬件也称硬设备，是指计算机系统各种看得见、摸得着、实实在在的装置，是计算机系统的物理基础。软件是指所有计算机的应用技术，是一些看不见、摸不着的程序与数据，但是可以感觉到它的存在，包括程序和与之相关的文档。硬件是软件建立和运行的基础，软件是计算机系统的灵魂。没有硬件对软件的物理支持，软件的功能则无从谈起。没有软件的计算机称为“裸机”，不能供用户直接使用。因此，计算机系统既包括硬件也包括软件，两者不可分割，只有两者有机地结合起来才能充分发挥计算机的功能。

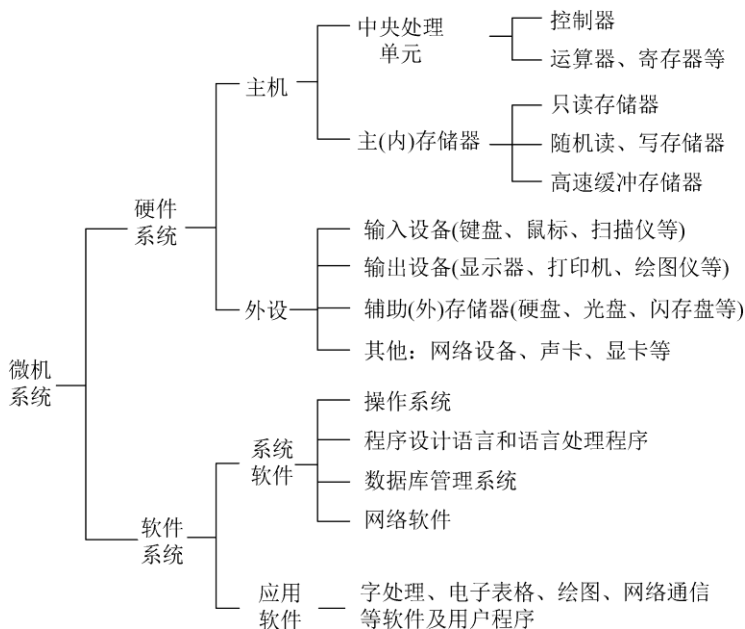


图 2.10 计算机系统的基本组成

### 2.5.4 计算机的硬件系统

早期的冯·诺依曼机在结构上是以控制器为中心，演变到现在，计算机已转向以存储器为中心，其最基本的结构图如图 2.11 所示，描述了五个部分的功能和关系。

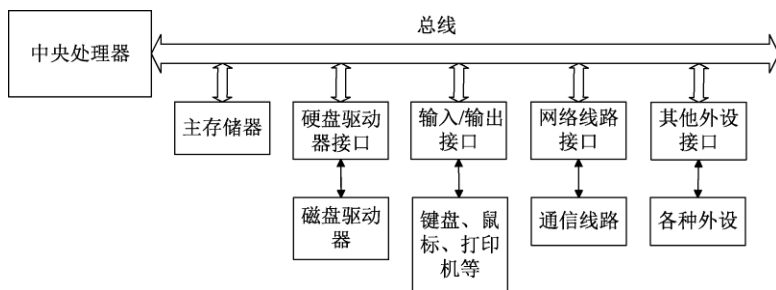


图 2.11 计算机的硬件结构

计算机的基本原理仍遵循冯·诺依曼提出的“存储程序”原则，是把复杂的计算、操作过程表示成由许多条基本指令组成的程序预先存入存储器中，需要时发出运行命令，计算机按程序规定的顺序逐条地执行指令，以完成所需的功能。

## 1. 主机

主机的外观如图 2.12 所示，主要配件有主机板、机箱、电源、显卡、硬盘驱动器、光盘驱动器等，其内部结构如图 2.13 所示。

(1) 机箱。机箱是计算机的外壳，用来安装电源、主板、硬盘驱动器、软盘驱动器等部件。

(2) 主板。主板(Mainboard)又称系统板(Systemboard)或微机母板(Motherboard)，是计算机中最大的一块电路板，它是连接 CPU 与其他部件的平台。主板包括 CPU、内存、基本 I/O 接口、中断控制器、DMA 控制器及连接其他部件的总线等，主板上都必须有 CPU 芯片、内存槽(Bank)、扩展槽(Slot)、各辅助电路及有关的跳线(Jumper)等。主板结构如图 2.14 所示。

(3) 微处理器。微处理器也称中央处理单元，安装在主板的 CPU 插槽中。微处理器是 PC 的核心部件，一般由高速的电子线路组成，集成在一块半导体芯片上，控制着整个计算机系统工作。目前 PC 系列计算机上使用的 CPU，主要是 Intel 公司和 AMD 公司生产的 CPU 系列芯片。微处理器如图 2.15 所示。



图 2.12 主机的外观



图 2.13 主机的内部结构



图 2.14 主板结构



图 2.15 微处理器



(4) 存储器。现代计算机系统都采用了多种类型的存储器，建立起合理的存储体系。典型的存储系统包括高速缓冲存储器、主存储器和辅助存储器等。

① 主存储器。主存储器又称内存储器，简称内存，用于存放当前正在使用的或随时要使用的数据，供 CPU 直接读取。计算机存取数据的时间快慢主要取决于内存，内存的大小和读写速度直接影响系统的速度及整机性能。由此不难看出，内存在计算机中的重要作用。目前，随着硬件制造水平的不断提高，计算机中所安装的内存容量越来越大，速度也越来越快。但内存的速度与 CPU 的速度相比仍有很大差距，内存仍是影响计算机速度的瓶颈之一。主存储器如图 2.16 所示。

内存容量以 MB 为单位，可以简写为 M。一般都是 2 的整次方倍，比如 64MB、128MB、256MB 等。一般而言，内存容量越大越有利于系统的运行。目前台式机中主流采用的内存容量为 4GB 或 8GB。系统对内存的识别以 Byte(字节)为单位，每个字节由 8 位二进制数组成，即 8bit(比特，也称“位”)。按照计算机的二进制方式，1Byte=8bit；1KB=1024Byte；1MB=1024KB；1GB=1024MB；1TB=1024GB；1PB=1024TB；1EB=1024PB；1ZB=1024EB；1YB=1024ZB。

系统中内存的数量等于插在主板内存插槽上所有内存条容量的总和，内存容量的上限一般由主板芯片组和内存插槽决定，不同的主板芯片组可以支持的容量是不同的。

② 辅助存储器。辅助存储器简称辅存，又称为外存储器，简称外存。它主要用于长期存储那些暂时不用的程序和数据。目前计算机上常用的辅助存储器有硬盘、光盘、软盘等。

硬盘是计算机中最主要的存储装置。与内存相比，硬盘的特点是存储容量大、价格较低，而且在断电的情况下可以长期保存信息，故又称为永久性存储器。普通硬盘如图 2.17 所示。

移动硬盘(Mobile Hard Disk)是以硬盘为存储介质，在计算机之间交换大容量数据，强调便携性的存储产品。近几年来，随着技术的推进和对存储速度的要求，固态硬盘(Solid State Disk, SSD)进入了人们的视野。固态硬盘是用固态电子存储芯片阵列而制成的硬盘，由控制单元和存储单元(FLASH 芯片、DRAM 芯片)组成。固态硬盘的接口规范和定义、功能及使用方法与普通硬盘完全相同。移动硬盘和固态硬盘如图 2.18 所示。



图 2.16 主存储器



图 2.17 普通硬盘



图 2.18 移动硬盘和固态硬盘

U 盘(USB flash disk, USB 闪存盘)是一种使用 USB 接口的无须物理驱动器的微型大容量移动存储产品，通过 USB 接口与电脑连接，实现即插即用。它以其存储容量大、价格适中、携带方便赢得人们的青睐，是当今移动存储的重要工具。U 盘如图 2.19 所示。

随着多媒体技术的发展，光盘以其容量大、寿命长、成本低的特点受到人们的欢迎，普及相当迅速。目前常用的光盘驱动器有只读型光盘、一次写入型光盘和可擦型光盘等三类光盘，也可分为 CD 和 DVD 等。光盘如图 2.20 所示。

软盘(见图 2.21)由于其方便小巧而流行了很多年，但由于其容量小、读取速度慢，而不能满足现在计算机对存储容量的要求，目前已经被淘汰，取而代之的是移动硬盘和 U 盘。



图 2.19 U 盘



图 2.20 光盘



图 2.21 软盘

(5) 显示卡。显示卡又简称为显卡，它是主机与显示器之间的接口电路。显卡直接插在主板的扩展槽上，但也有些主板上集成显示接口电路。显卡的主要功能是将主机输出的信号转换成显示器所能接受的形式。显示卡如图 2.22 所示。

(6) 声卡。声卡又称为音效卡，它安装在计算机主板的扩展槽上，有立体声输入/输出端口、麦克风插口、MIDI(Musical Instrument Digital Interface)接口等。目前，主板上大多集成了声卡，当对声音有特殊要求时，也可以安装声卡，以取得更好的声音效果。声卡如图 2.23 所示。

(7) 网卡。随着宽带的兴起，网卡已成为计算机的标准配置，目前常用的配置有 100Mbps 或 1000Mbps 网卡。网卡如图 2.24 所示。

(8) 电源。随着计算机部件和功能的不断增加，相关部件的功率越来越大，主机电源也因此越来越受到人们的重视。电源如图 2.25 所示。



图 2.22 显示卡



图 2.23 声卡



图 2.24 网卡



图 2.25 电源

## 2. 外部设备

外部设备主要包括以下几个方面。

(1) 显示器。显示器由监视器和显示适配器两部分组成，是计算机系统必不可少的输出设备，用于显示输出各种数据，将电信号转换成可以直接观察到的字符、图形或图像。液晶显示器和多屏显示器分别如图 2.26 和图 2.27 所示。



图 2.26 液晶显示器



图 2.27 多屏显示器

(2) 鼠标。鼠标是输入设备中最常用的一种，通常分为滚轮式鼠标、光电式鼠标和无线鼠标。鼠标如图 2.28 所示。

(3) 键盘。键盘是计算机中最常见的输入设备，用户的各种命令、程序和数据都可以通过键盘输入计算机。键盘如图 2.29 所示。

(4) 打印机。显示器只能将输出的结果显示在屏幕上供使用者观看，而打印机能将计算机所运行的结果或所需要的数据，按要求的方式在传统的纸张上输出，以便于交流。打印机如图 2.30 所示。



图 2.28 鼠标



图 2.29 键盘



图 2.30 打印机

近年来流行的 3D 打印机，是快速成形技术的一种机器，它是一种以数字模型文件为基础，运用粉末状金属或塑料等可黏合材料，通过逐层打印的方式来构造物体的技术。3D 打印机其打印的模型如图 2.31 所示。

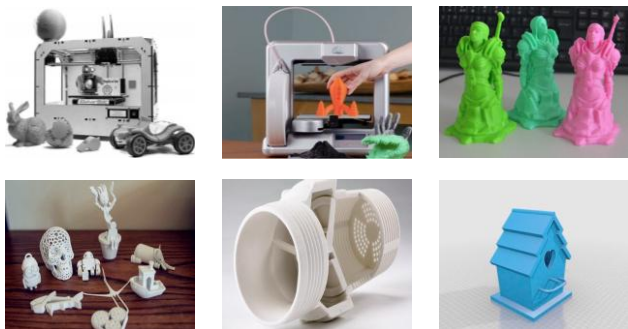


图 2.31 3D 打印机及其打印的模型

3D 打印技术最突出的优点是无须机械加工或任何模具，就能直接从计算机图形数据中生成任何形状的零件，从而极大地缩短产品的研制周期，提高生产率并降低生产成本。

(5) 扫描仪。扫描仪是计算机的重要输入设备。它可以将图片、文字等信息输入计算机中。扫描仪如图 2.32 所示。



图 2.32 扫描仪



图 2.33 光笔

(7) 摄像头。摄像头是网络时代重要的交流工具。它能实时地传播图像信息，进行实时视频等交流。常见的摄像头如图 2.34 所示。



图 2.34 摄像头



(8) 数码相机和数码摄像机。它们是信息时代重要的图像输入设备，分别如图 2.35 和图 2.36 所示。



图 2.35 数码相机



图 2.36 数码摄像机

### 2.5.5 计算机的软件系统

软件是指能在计算机上运行的各种程序，及各种相关的文档。通常将软件分为系统软件和应用软件两大类。

#### 1. 系统软件

可以把系统软件分成若干层，最内层是对硬件的扩充与完善，而外层则是对内层的再次扩充与完善。一般把靠近内层、为方便使用和管理计算机资源的软件，称为系统软件。系统软件通常是负责管理、控制和维护计算机的各种软硬件资源，并为用户提供一个友好的操作界面，以及服务于一般目的的上机环境。系统软件包括操作系统、计算机的监控管理程序、高级程序设计语言的编译和解释程序以及系统服务程序等。操作系统在系统软件中处于核心地位，其他的系统软件在操作系统的支持下工作；高级程序设计语言的编译和解释程序，将软件工程师编写的软件“翻译”成为计算机能够“理解”的机器语言；系统服务程序是为计算机系统的正常运行提供服务。

#### 2. 应用软件

应用软件是针对某个应用领域的具体问题而开发和研制的程序，它是由专业人员针对各种应用目的而开发的。应用软件必须在系统软件的支持下才能工作，它具有很强的实用性和专业性。正是由于应用软件的开发和使用，才使得计算机的应用日益渗透到社会的各行各业。应用软件可以由用户自己开发，也可在市场上购买。

### 2.5.6 计算机的特点

计算机的主要特点如下。

#### 1. 运算速度快、精度高

计算机的字长越长，其精度越高。一般计算机可以有十几位甚至几十位(二进制)有效数字，计算精度可由千分之几到百万分之几，是任何计算工具都望尘莫及的，如对于卫星轨道的计算、大型水坝的计算、气象预报等精度要求高、实时性强的工作，没有计算机进行数据处理，仅靠手工是无法实现的。

#### 2. 具有逻辑判断和记忆能力

计算机有准确的逻辑判断能力和超强的记忆能力，能够进行各种逻辑判断，并根据判



断的结果自动决定下一步应该执行的指令。

计算机已经远远不只是计算的工具，将计算机的计算能力、逻辑判断能力和记忆能力三者结合，使之可以模仿人的某些智能活动，是人类的脑力延伸，把计算机称为“电脑”也正是源于此。

### 3. 高度的自动化和灵活性

计算机采取存储程序的方式工作，即把编好的程序输入计算机，机器便可依次逐条执行，这就使计算机实现了高度的自动化和灵活性。每台计算机提供的基本功能是有限的，这是在设计和制造时就决定了的。但计算机可以在人们精心编写的程序下，用这些有限的功能，快速、自动地完成多种多样的基本功能序列，从而实现计算机的通用性，达到计算机应用的各种目的。

计算机的应用领域不断拓展，日益发挥着越来越重要的作用，成为“信息社会”科学技术和社会发展的重要推动力量。

## 2.5.7 计算机系统主要的技术指标

评价计算机的性能指标有很多。通常人们从计算机的字长、时钟周期和主频、运算速度、内存容量、数据输入/输出最高速率等技术指标来评价计算机系统的性能。

### 1. 字长

在计算机中，用若干二进制位表示一个数或一条指令，前者称为数据字，后者称为指令字。字长直接影响计算机的功能强弱、精度高低和速度快慢。计算机处理数据时，一次可以运算的数据长度称为一个“字”(Word)，字的长度称为字长。一个字可以是一个字节(Byte，简称B)，也可以是多个字节。常用的字长有8位(Bit)、16位、32位、64位等。如某一类计算机的字由8个字节组成，则字的长度为64位，相应的计算机称为64位机。

### 2. 时钟周期和主频

计算机的中央处理器对每条指令的执行是通过若干个微指令操作来完成的。这些微指令操作是按时钟周期的节拍来“动作”的。时钟周期的微秒数反映出计算机的运算速度。有时也用时钟周期的倒数——时钟频率(兆频)，即人们常说的主频来表示。一般说来，主频越高(时钟周期越短)，计算机的运算速度越快。但是，主频并不能全面准确地反映计算机的运算速度，而每秒执行百万条指令数(MIPS)指标则能较全面、准确地反映计算机的运算速度。例如，早年推出的IBM PC/XT微机的CPU主频只有4.77MHz，而Pentium 4 CPU的主频已超过1GHz。

### 3. 运算速度

计算机的运算速度是衡量计算机水平的一项主要指标，它取决于指令执行的时间。运算速度的计算方法多种多样，目前常用单位时间内执行多少条指令来表示，而计算机执行各种指令所需的时间不同。因此，常根据在一些典型题目计算中，各种指令执行的频度以及每种指令的执行时间来折算出计算机的等效速度。



### 4. 内存容量

存储器的容量反映计算机记忆信息的能力,它常以字节为单位表示。存储器的容量越大,则存储的信息越多,计算机的功能就越强。

计算机操作大多是与内存交换信息,但内存的存取速度相对于CPU的算术和逻辑运算速度要低1~2个数量级。因此,内存的读、写速度也是影响计算机运行速度的主要因素之一。

### 5. 数据输入/输出最高速率

主机与外部设备之间交换数据的速率也是影响计算机系统工作速度的重要因素。由于各种外部设备本身工作的速度不同,常用主机所能支持的数据输入/输出最大速率来表示。

## 2.5.8 计算机的分类

根据计算机工作原理和运算方式的不同,以及计算机中信息表示形式和处理方式的不同,计算机可分为数字式电子计算机(Digital Electronic Computer)、模拟式电子计算机(Analog Electronic Computer)和数字模拟混合电子计算机(Digital analog Hybrid Electronic Computer)。当今广泛应用的是数字式电子计算机。因此,常把数字式电子计算机(Digital Electronic Computer)简称为电子计算机或计算机。

根据计算机的用途,计算机可分为通用计算机(General Purpose Computer)和专用计算机(Special Purpose Computer)两大类。通用计算机能解决多种类型的问题,是具有较强的通用性的计算机,一般的数字式电子计算机多属此类;专用计算机是为解决某些特定问题而专门设计的计算机,如嵌入式系统。

根据计算机的总体规模(按照计算机的字长、运算速度、存储容量大小、功能强弱、配套设备多少、软件系统的丰富程度),计算机可分为巨型计算机(Supercomputer)、大/中型计算机(Mainframe)、小型计算机(Minicomputer)、微型计算机(Microcomputer)和网络计算机(Network Computer)五大类。

常见的微型计算机还可以分为台式机、笔记本和掌上型电脑等多种类型。

1966年,迈克尔·J.弗林(Michael J. Flynn, 1934—)根据指令流和数据流的数量对计算机系统结构进行分类。指令流是指机器执行的指令序列;数据流是由指令流调用的数据序列,包括输入数据和中间结果。按指令流和数据流的多倍性,弗林将计算机系统结构分成单指令流单数据流、单指令流多数据流、多指令流单数据流、多指令流多数据流四类。由于当前的计算机系统结构的主流发展方向是控制驱动方式下的并行处理,故这一分类法获得普遍认同。



Michael J. Flynn

## 2.5.9 计算机的基本工作方式

计算机的基本工作方式可概括为“IPOS循环”,即输入(Input)、处理(Processing)、输出(Output)和存储(Storage),它反映了计算机进行工作的基本步骤。

(1) 输入。接收由输入设备(如键盘、鼠标器、扫描仪等)提供的数据。

(2) 处理。对数值、逻辑、字符等各种类型的数据进行操作,按指定的方式进行转换。



- (3) 输出。将处理所产生的结果等数据由输出设备(如显示器、打印机、绘图仪等)输出。
- (4) 存储。计算机可以存储程序和数据供以后使用。

## 2.6 存储系统的结构

存储器是具有“记忆”功能的部件,在计算机系统中占据十分重要的地位。存储器的基本功能是存放以二进制形式表示的程序和数据。如何设计容量大、速度快且成本低的存储器,一直是计算机发展中的关键问题,目前还没有哪一种存储器功能完全满足计算机系统对存储器的需求。因此,计算机系统通常配备分层结构的存储器系统,以满足容量、速度、成本等方面的要求。

### 2.6.1 存储器的分类

现代计算机是以存储器为中心的计算机系统,当利用计算机完成某项任务时,首先把解决问题的程序和所需数据存入存储器中,在执行程序时再由存储器快速地提供给处理机。这种方式是现代计算机的重要特征之一,通常称为程序存储。

目前已出现了多种类型的存储器,根据存储器位于主机中的不同位置,可以将其分为内存储器和外存储器两种。内存储器主要用于存放当前执行的程序和数据,与外存储器相比,其存储容量较小,但工作速度较快。外存储器位于主机外部,主要用于存放当前不参加运行的程序与数据,在需要时,可与内存以批处理的方式交换信息,其特点是存储容量大,但速度较低。

#### 1. 按存储介质划分

按存储介质,可将存储器分为半导体存储器、磁存储器和光存储器。

(1) 半导体存储器。一般用大规模集成电路工艺制成一定容量的芯片,再由若干芯片组成大容量的存储器。半导体存储器又分为双极型半导体存储器、MOS 半导体存储器等。

(2) 磁存储器。是在金属或塑料基体的表面上,涂敷一层磁性材料作为记录介质,这层介质称为磁层。工作时,磁层随机体高速运动,用磁头在磁层上进行读、写操作。根据介质的不同开关,磁存储器有磁芯存储器、磁带存储器和磁盘存储器等,当前广泛使用的是磁盘存储器。在计算机系统中,磁盘存储器是基本配置。

(3) 光存储器。是使用激光技术来保存信息的存储器。常见的有激光存储器、集成光路存储器等。

#### 2. 按工作方式划分

按工作方式,可将存储器分为随机存取存储器、只读存储器、顺序存取存储器和直接存取存储器等。

(1) 随机存取存储器(Random Access Memory, RAM)。在随机存取存储器中,以任意次序读、写任意存储单元所用的时间都相同,并且在一个存或取周期内只能进行一次访问。随机存取存储器常用作主存或高速缓冲存储器。

(2) 只读存储器(Read Only Memory, ROM)。只读存储器常用来存放固定不变的系统程



序,因此又叫“固存”。早期的只读存储器,在工作时内容不能改变,即禁止写入,是一种非破坏性读出。后来又出现了可编程的只读存储器(PROM)、可擦编程只读存储器(EPROM)、电可擦可编程只读存储器(EEPROM)等。

(3) 顺序存取存储器(Sequential Access Memory, SAM)。在顺序存取存储器中,只能以某种预先确定的顺序来读、写存储单元,如磁带存储器就是顺序存取存储器。这种存储器的特点是存取时间长,速度慢。但它的存储容量可以做得较大,单位价格也较低。

(4) 直接存取存储器(Direct Access Memory, DAM)。直接存取存储器既不像随机存取存储器那样随机地选择存储地址进行存储,也不像磁带那样顺序地进行存储,而是介于两者之间。如果要存取所需的信息,就必须有两个逻辑操作:首先利用随机存取的方式找到地址 A 附近的范围,然后以顺序存取的方式在地址 A 附近找到地址 A,如磁盘存储器就是直接存取存储器。这种存储器的容量大,存取速度介于随机存取存储器与顺序存取存储器之间,多用作辅助存储器。

### 3. 按信息的可保存性划分

按信息的可保存性,可将存储器分为易失性存储器和非易失性存储器。断电后,存储的信息将消失在存储器称为易失性存储器,常见的有 RAM 存储器;断电后,存储的信息仍保存在存储器中,称为非易失性存储器,常见的有 ROM、磁盘存储器等。

### 4. 按在计算机中的作用划分

按在计算机中的作用,可将存储器分为主存储器、辅助存储器和高速缓冲存储器等。

## 2.6.2 存储系统的组织结构

随着计算机硬件系统和软件系统的不断发展、计算机应用领域的日益扩大,对存储器的要求也越来越高。然而,由于硬件技术发展的限制,存储器也成为影响计算机性能发展的瓶颈。

目前对计算机存储器既要求存储容量大、存取速度快,又要求成本低。这种要求本身是相互矛盾的,也是相互制约的,在同一存储器中难以同时满足。因此,设计存储器时,相应的约束有容量、速度和价格,而对容量似乎没有限制,不管容量多大,总要开发出应用程序来使用。为了获得最好的性能,存储器的速度必须能够跟上 CPU,即当 CPU 执行指令时,不想让它停下来等待指令或操作数。价格问题也必须考虑,对于实用的系统,存储器的价格必须相对于其他部件是合理的。

半导体存储器具有较快的存取速度,但存储容量却有限;磁盘和磁带的存储容量大,但存取速度慢。显然,摆在设计者面前的难题是存储器不仅需要大的容量,而且需要低的价格。解决的办法是采用存储器分层结构,而不只是依赖单一的存储部件或技术。为了发挥它们各自的优点,按照一定的体系结构有机地组合起来,可得到如图 2.37 所示的三级存储体系分层结构。

这种三级结构的存储器系统,是围绕读写速度尚可、存储容量适中的主存储器来组织和运行的,并由高速缓冲存储器缓解主存储器读写速度慢、不能满足 CPU 运行速度需要的矛盾;用虚拟存储器更大的存储空间来解决主存

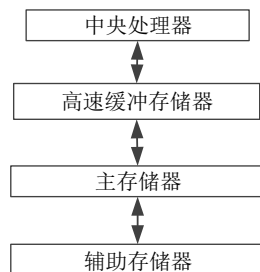


图 2.37 三级存储体系分层结构

容量小、存不下规模更大的程序与更多数据的难题，从而达到使整体存储器系统有更高的读写速度、更大的存储空间、相对较低的制造与运行成本的要求。

## 2.7 输入/输出系统

计算机同外部世界通信或交换数据的设备称为输入/输出(Input/Output, I/O)系统,它是计算机系统的重要组成部分,对计算机系统的运行性能有很大影响。I/O 系统的主要作用是连通计算机的各个功能部件和设备,并在它们之间实现数据交换。它与计算机的速度、处理能力、实用性、兼容性等各种性能都有着非常密切的关系。随着计算机应用的不断深入,I/O 系统占有越来越重要的地位。I/O 系统的硬件部分主要由计算机总线和 I/O 接口两部分组成,软件方面则需要有操作系统的支持。

由于种类繁多、功能结构各异的 I/O 设备,在与主机的连接方法和信息交换方法上各不相同,因此 I/O 系统是计算机系统中最具多样性和复杂性的子系统。为了发挥计算机的性能,提高效率,以及得到高效可靠的信息传送,设计合理的 I/O 系统,配备先进的 I/O 技术及接口部件是非常必要的。

计算机与外部世界的联系是通过交换信息实现的。在硬件上,输入/输出系统由四部分组成,如图 2.38 所示。

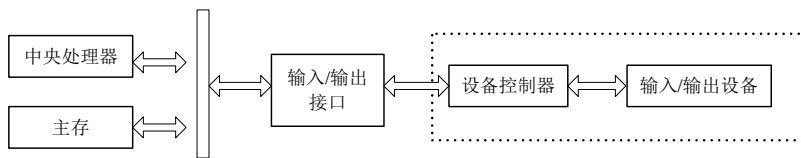


图 2.38 输入/输出系统的组成

(1) 外部设备。外部设备能够利用光、电、磁及机械传动等手段,把信息转换为二进制代码的形式。常用的基本外部设备有键盘、鼠标、显示器、打印机等。

(2) 设备控制器。设备控制器能将设备生成的各种形式的二进制代码转换成电信号,并根据输入信号的要求对设备进行控制。设备控制器是设备与计算机连接的界面,是外部设备的一个组成部分。

(3) 输入/输出接口。主机不能直接与外部设备交换数据,必须在主机与设备之间设置一个硬件,称为输入/输出接口。

(4) 中央处理器。中央处理器执行输入/输出指令,对输入/输出系统进行启动、检测及控制。

## 2.8 多核技术

多核处理器(Multicore Chips)是指在一枚处理器中集成两个或多个完整的计算引擎(内核)。多核技术的开发源于工程师们意识到,仅仅提高单核芯片的速度会产生过多热量且无法带来相应的性能改善,即使没有热量问题,其性价比也令人难以接受,速度稍快的处理器价格要高很多。

工程师们开发了多核芯片,使之满足“横向扩展”(而非“纵向扩充”)方法,从而提高了性能。该架构实现了“分治法”策略,通过划分任务,线程应用能够充分利用多个执



行内核，并可在特定的时间内执行更多的任务。通过在多个执行内核之间划分任务，多核处理器可在特定的时钟周期内执行更多任务。多核架构能够使目前的软件更出色地运行，并能促进未来的软件编写更趋完善。

## 2.9 高性能计算机

在过去的近 30 年中，计算机科学和技术的迅速发展，具体表现在市场上计算机的性能价格比得到迅速提高。以高性能计算机为基础的计算科学已经成为继理论科学和实验科学之后人类科学研究的第三大支柱，在一些新兴的学科，如新材料技术和生物技术领域，高性能计算机已成为科学研究的必备工具。同时，高性能计算机也越来越多地渗透到石油工业等一些传统的产业之中，从而帮助这些产业提高了生产效率、降低了生产成本。

### 1. 高性能计算机系统

高性能计算机的概念并无明确的定义，一般地，运算速度非常快的计算机就可以被认为是高性能计算机。严格地讲，高性能计算机是一个拥有最先进的硬件、软件、网络和算法的综合概念，“高性能”的标准是随着技术的发展而发展的。

随着高性能计算机性能的提高、价格的降低，高性能计算机已经从传统的科研和国家战略需要走向更广泛的行业应用，已经有越来越多的经济部门使用高性能计算机，如石油勘探、机械设计、金融分析、生物制药等。

### 2. 集群系统

集群是指共同为客户机提供网络资源的一组计算机系统，其中的每一台提供服务的计算机，称之为节点。当一个节点不可用或者不能处理客户的请求时，该请求将会转到另外的可用节点来处理。而这些对客户端来说，它根本不必关心这些要使用的资源的具体位置，集群系统会自动完成。

高性能集群计算机的特征是将集群内不同节点的资源集中使用，并行完成同一任务，实现负载均衡，同时提供良好的失效平滑接管。集群系统可以通过全冗余方式完全屏蔽单点失效，集群内任何节点失效后，其他节点通过内存同步技术可以在无延迟和不丢失数据的情况下接管失效节点。而且随着应用的发展，集群可以通过随时增加节点数来平滑地增加处理能力。

此外，高性能计算机还有超标量计算机、向量计算机等，它们都有各自的优点和缺点，在不同的领域发挥着各自的作用。

## 2.10 并行计算机

在近 10 年中，随着微处理器的发展，单处理机性能的增长达到了自晶体管计算机出现以来的最高速度。然而，单处理机的发展正在走向尽头。随着多核处理器芯片的流行，并行计算机在未来将会发挥更大的作用。

并行计算机是由多个处理器组成(在这些处理器之间还可以相互通信和协调)，并能够高速、高效率地进行复杂问题计算的计算机系统。并行计算机是相对于串行计算机而言的。



串行计算机是指只有单个处理器，顺序执行计算程序的计算机，也称为顺序计算机。并行计算作为计算机技术，是在 20 世纪 70 年代中期提出来的，至今已有 40 多年的发展历史。该技术的应用已经带来单处理机计算能力的巨大改进。

## 2.11 分布式系统

并行计算机系统结构、计算机网络和分布式操作系统是三种联系紧密并且相互渗透的系统。分布式系统是指呈现给用户的如同单机系统一样的独立计算机的集合，可将多处理机和多计算机以及高速局域网，看成分布式系统的硬件平台、分布式操作系统，包括进程和线程管理、分布式共享存储器管理、分布式文件系统等，属于分布式系统的软件范畴。

随着计算机技术和计算机网络技术的发展，构造在局域网甚至广域网之上，由多台到几十甚至上百台计算机组成的多计算机系统已成为当前分布式系统的主流。

支持这种松散耦合式多机系统的操作系统可分为两大类。一是网络操作系统(Net Operating System, NOS)，属于松耦合的软件。NOS 主要是提供资源共享和高效可靠的网络通信。二是分布式操作系统(Distributed Operating System, DOS)，属于紧耦合的软件。DOS 主要是支持大型计算问题的高速求解和大型数据库管理。

真正的分布式操作系统与网络操作系统有一个重要的不同点，即它强调整个多机系统在它的支持和管理下呈现在用户面前，如同一个传统的分时单机系统。对于任意一个以计算机网络(多是高速局域网)连接起来的松散耦合的多计算机系统，却要在用户面前呈现出像单机系统一样，则分布式操作系统的设计难度会很大。因此，要求分布式操作系统具有更深层次的系统资源共享能力，更加高速有效的不只是用户作业级的而是进程级的通信能力，以及统一的用户界面。

## 本章小结

本章介绍了计算机的相关基础知识，包括数值数据的表示与运算，非数值数据的表示方法、数据的机器级及汇编级表示、数字逻辑和数字系统、计算机系统的组成、存储系统、输入/输出系统结构、多核技术的研究、高性能、并行计算机的概念和分布式系统等。

通过本章的学习，读者应对计算机中数据的表示有基本的理解，对计算机的体系结构有基本的认识，对于多核技术、高性能和并行计算机等前沿领域有初步的了解，从而为进一步学习本书的后续各章打好基础。

## 习 题

### 一、选择题

- 就工作原理而论，提出存储程序控制原理的科学家是( )。  
A. 巴尔基            B. 牛顿            C. 希尔            D. 冯·诺依曼
- 计算机组成包括输入设备、输出设备、运算器、存储器和( )。  
A. 键盘            B. 显示器            C. CPU            D. 控制器



3. 中央处理器包括( )。
- A. 运算器和控制器  
B. 运算器和存储器  
C. 控制器和输入设备  
D. 输入设备和存储器
4. 硬盘属于计算机的( )。
- A. 主存储器  
B. 输入设备  
C. 输出设备  
D. 辅助存储器
5. 显示器属于计算机的( )。
- A. 主存储器  
B. 输入设备  
C. 输出设备  
D. 辅助存储器
6. 计算机要接入因特网, 需要( )。
- A. 键盘  
B. 鼠标  
C. 网卡  
D. 闪存盘
7. ( )不是英特尔公司发布的微处理器。
- A. 奔腾  
B. 酷睿  
C. 凌动  
D. 速龙
8. 两个数  $N_1$  和  $N_2$  的真值分别为:  $N_1 = +0101010$ ,  $N_2 = -01111100$ , 那么它们的机器数为( )。
- A.  $N_1 = 00101010$ ,  $N_2 = 001111100$   
B.  $N_1 = 00101010$ ,  $N_2 = 101111100$   
C.  $N_1 = 10101010$ ,  $N_2 = 001111100$   
D.  $N_1 = 10101010$ ,  $N_2 = 101111100$
9. 如果  $[X]_{\text{补}} = 11110011$ , 则  $[-X]_{\text{补}}$  是( )。
- A. 11110011  
B. 01110011  
C. 00001100  
D. 00001101
10. 存储器 ROM 的功能是( )。
- A. 可读可写数据  
B. 可写数据  
C. 只读数据  
D. 不可读写数据
11. 若十进制数据为 137.625, 则其二进制数为( )。
- A. 10001001.11  
B. 10001001.101  
C. 10001011.101  
D. 1011111.101
12. 存储器存储容量单位中, 1KB 表示( )。
- A. 1024 个字节  
B. 1024 位  
C. 1024 个字  
D. 1000 个字节
13. 十进制数 45D 的二进制数表示形式为( )。
- A. 101101H  
B. 110010B  
C. 101101B  
D. 110010Q
14. 编译程序的作用是( )。
- A. 把源程序译成目标程序  
B. 解释并执行程序  
C. 把目标程序译成源程序  
D. 对源程序进行编辑
15. 下列各数中最大的是( )。
- A. 110B  
B. 110O  
C. 110H  
D. 110D
16. 主存储器和 CPU 之间增加高速缓冲存储器的目的是( )。
- A. 解决 CPU 和主存之间的速度匹配问题  
B. 扩大主存储器的容量  
C. 扩大 CPU 中通用寄存器的数量  
D. 既扩大主存容量又扩大 CPU 通用寄存器数量
17. 串行接口是指( )。
- A. 主机和接口之间、接口和外设之间都采用串行传送  
B. 主机和接口之间串行传送, 接口和外设之间并行传送  
C. 主机和接口之间并行传送, 接口和外设之间串行传送  
D. 系统总线采用串行总线



18. 完整的计算机系统应包括( )。
- A. 运算器、存储器、控制器      B. 主机和实用程序  
C. 配套的硬件设备和软件设备      D. 外部设备和主机
19. 32 个汉字的机内码需要的字节数是( )。
- A. 16      B. 32      C. 64      D. 128
20. 数据总线、地址总线、控制总线 3 类划分根据是( )。
- A. 总线传送的内容      B. 总线所处的位置  
C. 总线传送的方向      D. 总线传送的方式
21. 多媒体是指计算机处理信息媒体的多样化,它是以( )方式进行的。
- A. 交互      B. 声音      C. 视频      D. 文本
22. 每次可传送一个字或一个字节的全部代码,并且是对一个字或字节各位同时进行处理的信息传递方式是( )。
- A. 串行方式      B. 并行方式      C. 查询      D. 中断
23. 若以每次传送一位的方式实现数据的传送,即每个字节是按位传送的,这种信息传递方式是( )。
- A. 串行方式      B. 并行方式      C. 查询      D. 中断

## 二、简答题

1. 简述计算机采用二进制的原因。
2. 什么是定点数? 它分为哪些种类?
3. 简述声音的编码过程。
4. 简述计算机有哪些特点。
5. 简述计算机软件系统的分类。
6. 列出常用的系统软件和应用软件。
7. 存储器的分类有哪些?
8. 存储器的功能是什么?
9. 存储器的主要指标是什么?
10. 简述多核的关键技术。
11. 什么是高性能计算机?
12. 什么是接口? 它的主要功能是什么?
13. 简述并行算法的基本内容。
14. 什么是网络计算机? 它有什么优点?

## 三、讨论题

1. 为什么计算机使用二进制,而不使用人们生活中的十进制来表示数据信息?
2. 网络计算机有许多优点,请结合其特点谈谈我国网络计算机的发展前景。