

第3章 数据在计算机中的表示

3.1 数据与数制

计算机中使用的数据一般可以分为两大类：数值数据和字符数据。数值数据常用于表示数的大小与正负；字符数据则用于表示非数值的信息，例如：英文、汉字、图形和语音等数据。数据在计算机中是以电子器件的物理状态（如：开、关状态）来表示的，因此，各种数据在计算机中都是用二进制编码的形式来表示。

3.1.1 数据及其分类

1. 数据

在计算机系统中，各种字母、数字符号的组合、语音、图形、图像等统称为数据，数据经过加工后就成为信息。

在计算机科学中，数据是指所有能输入到计算机并被计算机程序处理的符号的总称，是用于输入电子计算机进行处理，具有一定意义的数字、字母、符号和模拟量等的通称。数据种类很多。

2. 数据分类

（1）按性质分

数据按性质可分为：①定位的，如各种坐标数据；②定性的，如表示事物属性的数据（居民地、河流、道路等）；③定量的，反映事物数量特征的数据，如长度、面积、体积等几何量或重量、速度等物理量；④定时的，反映事物时间特性的数据，如年、月、日、时、分、秒等。

（2）按表现形式分

数据按表现形式可分为：①数字数据，如各种统计或量测数据；②模拟数据，由连续函数组成，是指在某个区间连续变化的物理量，又可以分为图形数据（如点、线、面）、符号数据、文字数据和图像数据等，如声音的大小和温度的变化等。

（3）按记录方式分

数据按记录方式可分为：地图、表格、影像、磁带、纸带。按数字化方式分为矢量数据、格网数据等。

3.1.2 数制

数制是学习数据在计算机中表示的基础，不掌握数制的运算很难理解各种数据在计算机中的表示。

数制也称计数制，是指用一组固定的符号和统一的规则来表示数值的方法。按进位的方法进行计数，称为进位计数制。在日常生活和计算机中采用的是进位计数制。在日常生活中，人们最常用的是十进位计数制，即按照“逢十进一”的原则进行计数的。

在进位计数制中有数码、数位（位置）、基数和位权等用语。数码是在一个计数制中用来表示数值的符号；数位是指数码在一个数中所处的位置；基数是指在某种进位计数制中，每个数位上的数码所代表数值大小的基准数；位权是基数位置的幂次，表示数码在不同位置上所代表的数值。

例如，一个十进制数中每个数位上可以使用的数码为 0、1、2、3、4、5、6、7、8、9 十个数码，基数为十。基数为十的进位计数制按“逢十进一”的原则进行计数。数码所处的位置不同，代表数值的大小也不一样。

例如，在十进位计数制中，小数点左边第一位为个位数，其位权为 10^0 ，第二位为十位数，其位权为 10^1 ，第三位是百位数，其位权为 10^2 ，……；小数点右边第一位是十分位数，其位权为 10^{-1} ，第二位是百分位数，其位权为 10^{-2} ，第三位是千分位数，其位权为 10^{-3} ，……。十进数的 7675 从右面起的第一位 5 是个位，第二位 7 是十位，第三位 6 是百位，第四位是 7 千位，……。“个、十、百、千……”在数学上叫做“位权”或“权”。每一位上的数码与该位“位权”的乘积表示了该位数值的大小。

基数、位权和进位原则是进位计数制中的三个要素。

在微机中，常用的是十进制、二进制、八进制和十六进制，它们对应的关系如表 3-1 所示。

表 3-1 十进制、二进制、八进制、十六进制的对应关系

十进制	二进制	八进制	十六进制
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	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

1. 十进制数

十进位计数制简称十进制。十进制数具有下列特点：

- (1) 基数：10。
- (2) 数码：0、1、2、3、4、5、6、7、8、9。

(3) 每一个数码根据它在这个数中所处的位置（数位），按“逢十进一”来决定其实际数值，即各数位的位权是以 10 为底的幂次。

例如 $(356.456)_{10}$ ，以小数点为界，从小数点往左依次为个位、十位、百位，从小数点往右依次为十分位、百分位、千分位。因此，小数点左边第一位 6 代表数值 6，即 6×10^0 ，第二位 5 代表数值 50，即 5×10^1 ；第三位 3 代表数值 300，即 3×10^2 ；小数点右边第一位 4 代表数值 0.4，即 4×10^{-1} ；第二位 5 代表数值 0.05，即 5×10^{-2} ；第三位 6 代表数值 0.006，即 6×10^{-3} 。因而该数可表示为如下形式：

$$(356.456)_{10} = 3 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2} + 6 \times 10^{-3}$$

由上述分析可归纳出，任意一个十进制数 D ，可表示成如下形式：

$$(D)_{10} = D_{n-1} \times 10^{n-1} + D_{n-2} \times 10^{n-2} + \cdots + D_1 \times 10^1 + D_0 \times 10^0 + D_{-1} \times 10^{-1} + D_{-2} \times 10^{-2} + \cdots + D_{-m} \times 10^{-m}$$

式中 D_i ($i: n-1, n-2, \dots, m-1$) 为数位上的数码，其取值范围为 $0 \sim 9$ ； n 为整数位个数， m 为小数位个数，10 为基数， $10^{n-1}, 10^{n-2}, \dots, 10^1, 10^0, 10^{-1}, \dots, 10^{-m}$ 是十进制数的位权。因为人们习惯使用十进制数，所以在计算机中，一般用十进制数作为数据的输入和输出。

2. 二进制数

二进位计数制简称二进制。二进制数具有下列特点：

- (1) 基数：2。
- (2) 数码：0、1。
- (3) 每个数码根据它在这个数中的数位，按“逢二进一”来决定其实际数值。

例如： $(11011.01)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = (27.625)_{10}$

任意一个二进制数 B ，可以表示成如下形式：

$$(B)_2 = B_{n-1} \times 2^{n-1} + B_{n-2} \times 2^{n-2} + \cdots + B_1 \times 2^1 + B_0 \times 2^0 + B_{-1} \times 2^{-1} + \cdots + B_{-m} \times 2^{-m}$$

式中 B_i 为数位上的数码，其取值范围为 $0 \sim 1$ ； n 为整数位个数， m 为小数位个数。2 为基数。 $2^{n-1}, 2^{n-2}, \dots, 2^1, 2^0, 2^{-1}, \dots, 2^{-m}$ 是二进制数的位权。

计算机中数的存储和运算都使用二进制数。

3. 八进制数

八进位计数制简称八进制。八进制数具有下列特点：

- (1) 基数：8。
- (2) 数码：0、1、2、3、4、5、6、7。
- (3) 每个数码根据它在这个数中的数位，按“逢八进一”来决定其实际的数值。

例如： $(432.24)_8 = 4 \times 8^2 + 3 \times 8^1 + 2 \times 8^0 + 2 \times 8^{-1} + 4 \times 8^{-2} = (282.3125)_{10}$

任意一个八进制数 Q ，可以表示成如下形式：

$$(Q)_8 = Q_{n-1} \times 8^{n-1} + Q_{n-2} \times 8^{n-2} + \cdots + Q_1 \times 8^1 + Q_0 \times 8^0 + Q_{-1} \times 8^{-1} + \cdots + Q_{-m+1} \times 8^{-m+1} + Q_{-m} \times 8^{-m}$$

式中 Q_i 为数位上的数码，其取值范围为 $0 \sim 7$ ； n 为整数位个数， m 为小数位个数。8 为基数， $8^{n-1}, 8^{n-2}, \dots, 8^1, 8^0, 8^{-1}, \dots, 8^{-m}$ 是八进制数的位权。

八进制数是计算机中常用的一种计数方法，它可以弥补二进制数书写位数过长的不足。

4. 十六进制数

十六进位计数制简称为十六进制。十六进制数具有下列特点：

- (1) 基数：16。
- (2) 数码：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。由于数字只有 $0 \sim 9$ 十个，而十六进制要使用十六个数字，而每位上只能有一个数码，所以用 A~F 六个英文字母分别表示数字 10~15。

(3) 每个数码根据它在这个数中的数位, 按“逢十六进一”来决定其实际的数值。

例如: $(A3B.48)_{16} = A \times 16^2 + 3 \times 16^1 + B \times 16^0 + 4 \times 16^{-1} + 8 \times 16^{-2} = (2619.28125)_{10}$

任意一个十六进制数 H , 可表示成如下形式:

$$(H)_{16} = H_{n-1} \times 16^{n-1} + H_{n-2} \times 16^{n-2} + \cdots + H_1 \times 16^1 + H_0 \times 16^0 + H_{-1} \times 16^{-1} + \cdots + H_{-m} \times 16^{-m}$$

其中 H_i 为数位上的数码, 其取值范围为 $0 \sim F$; n 为整数位个数, m 为小数位个数。16 为基数, $16^{n-1}, 16^{n-2}, \dots, 16^1, 16^0, 16^{-1}, \dots, 16^{-m}$ 为十六进制数的位权。

十六进制数是计算机中常用的一种计数方法, 它可以弥补二进制数书写位数过长的不足。

总结以上四种计数制, 可将它们的特点概括为:

(1) 每一种计数制都有一个固定的基数 R (R 为大于 1 的整数), 它的每一数位可取 R 个不同的数码, 分别为 $0 \sim R-1$;

(2) 每一种计数制都有自己的位权, 并且遵循“逢 R 进一”的原则。

对于任一个 R 进制计数制数 S , 可表示为:

$$\begin{aligned} (S)_R &= \pm (S_{n-1}R^{n-1} + S_{n-2}R^{n-2} + \cdots + S_1R^1 + S_0R^0 + S_{-1}R^{-1} + \cdots + S_{-m}R^{-m}) \\ &= \pm \sum_{i=n-1}^{-m} S_i R^i \end{aligned}$$

式中 S_i 表示各数位上的数码, 其取值范围为 $0 \sim R-1$, R 为计数制的基数, i 为数位的编号 (整数位取 $n-1 \sim 0$, 小数位取 $-1 \sim -m$)。

另外, 在表示十六进制数时, 在数码后加 H , 如 $(A3B.48)_{16}$, 可写成 $A3B.48H$ 。

3.1.3 不同进制数之间的转换

不同进位计数制之间的转换, 实质是基数间的转换。一般转换的原则是: 如果两个有理数相等, 则两数的整数部分和小数部分一定分别相等。因此, 各数制之间进行转换时, 通常对整数部分和小数部分分别进行转换。

1. 非十进制数转换成十进制数

非十进制数转换成十进制数的方法是, 把各个非十进制数按位权展开求和即可。即把二进制数 (或八进制数, 或十六进制数) 写成 2 (或 8 或 16) 的各次幂之和的形式, 然后计算其结果。

【例 3-1】 把下列二进制数转换成十进制数。

$$\begin{aligned} (1110101)_2 &= 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 64 + 32 + 16 + 0 + 4 + 0 + 1 = (117)_{10} \\ (10101.101)_2 &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 16 + 0 + 4 + 0 + 1 + 0.5 + 0 + 0.125 = (21.625)_{10} \end{aligned}$$

【例 3-2】 把下列八进制数转换成十进制数。

$$\begin{aligned} (305)_8 &= 3 \times 8^2 + 0 \times 8^1 + 5 \times 8^0 = 192 + 5 = (197)_{10} \\ (456.124)_8 &= 4 \times 8^2 + 5 \times 8^1 + 6 \times 8^0 + 1 \times 8^{-1} + 2 \times 8^{-2} + 4 \times 8^{-3} \\ &= 256 + 40 + 6 + 0.125 + 0.03125 + 0.0078125 \\ &= (302.1640625)_{10} \end{aligned}$$

【例 3-3】 把下列十六进制数转换成十进制数。

$$\begin{aligned} (2A4F)_{16} &= 2 \times 16^3 + A \times 16^2 + 4 \times 16^1 + F \times 16^0 \\ &= 8192 + 2560 + 64 + 15 = (10831)_{10} \end{aligned}$$

$$(3B2F.A8)_{16} = 3 \times 16^3 + B \times 16^2 + 2 \times 16^1 + F \times 16^0 + A \times 16^{-1} + 8 \times 16^{-2}$$

$$= 12288 + 2816 + 32 + 15 + 0.625 + 0.03125 = (15151.65625)_{10}$$

2. 十进制数转换成非十进制数

(1) 十进制数转换成二进制数

把十进制数转换为二进制数的方法是：

整数转换用“除2取余法”；小数转换用“乘2取整法”。

所谓除2取余法，就是将已知十进制数反复除以2，若每次相除之后余数为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数的低位，最后一次余数是二进制数的高位。从低位到高位逐次进行，直到商为0。最后一次除法所得的余数为 K_{n-1} ，则 K_{n-1} 、 K_{n-2} 、……、 K_1 、 K_0 即为所求的二进制数。

例如：将 $(100)_{10}$ 转换成二进制整数，其全过程可表示如下：

		余数	
2	100	0	逆 排 列
2	50	0	
2	25	1	
2	12	0	
2	6	0	
2	3	1	
2	1	1	
	0		

$$(100)_{10} = (K_6 K_5 K_4 K_3 K_2 K_1 K_0)_2 = (1100100)_2$$

十进制纯小数转换成二进制纯小数，采用乘2取整法，就是将已知十进制纯小数反复乘以2，每次乘2之后，所得新的整数部分为1，相应位为1，如果整数部分为0，则相应位为0。从高位向低位逐次进行，直到乘2取整后的小数部分为0或满足精度要求，若保留 m 位小数，对 $m+1$ 位小数进行0舍1入。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 $K_{-1}K_{-2} \cdots K_{-m}$ 。

例如：将 $(0.625)_{10}$ 转换成纯二进制小数，转换过程如下：

	整数	0.625	
正 排 列	1	0.250	整数部分=1, K_{-1}
	0	0.500	整数部分=0, K_{-2}
	1	0.000	整数部分=1, K_{-3}

$$(0.625)_{10} = (0.K_{-1}K_{-2}K_{-3})_2 = (0.101)_2$$

再如：将 $(0.6531)_{10}$ 转换成纯二进制小数，转换过程如下：

		0.6531	
	整数	$\times 2$	
	1	1.3062	 整数部分=1, K_1
		$\times 2$	
	0	0.6124	 整数部分=0, K_2
		$\times 2$	
正	1	1.2248	 整数部分=1, K_3
排		$\times 2$	
列	0	0.4496	 整数部分=0, K_4
		$\times 2$	
	0	0.8992	 整数部分=0, K_5
		$\times 2$	
	1	1.7984	 整数部分=1, K_6
		$\times 2$	
	1	1.5968	 整数部分=1, K_7
		$\times 2$	
	1	1.1936	 整数部分=1, K_8
		$\times 2$	
	0	0.3872	 整数部分=0, K_9

如只取八位小数能满足精度要求, 则得

$$\begin{aligned}
 (0.6531)_{10} &= (0.K_1K_2\dots K_m)_2 \\
 &\approx (0.K_1K_2K_3K_4K_5K_6K_7K_8) \\
 &= (0.10100111)_2
 \end{aligned}$$

可见, 十进制纯小数不一定能转换成完全等值的二进制纯小数。遇到这种情况时, 根据精度要求, 取近似值。

则 $(100.6531)_{10} \approx (1100100.10100111)_2$

(2) 十进制数转换成八进制数

十进制数转换成八进制数的方法是:

整数部分转换采用“除 8 取余法”;

小数部分转换采用“乘 8 取整法”。

(3) 十进制数转换成十六进制数

将十进制数转换成十六进制数的方法是:

整数部分转换采用“除 16 取余法”;

小数部分转换采用“乘 16 取整法”。

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

对于二进制整数, 只要自右到左将每四位二进制数分为一组, 不足四位时, 在左面添 0, 补足四位; 对于二进制小数, 只要自左到右将每四位二进制数分为一组, 不足四位时, 在右面添 0, 补足四位, 然后将每组用相应的十六进制数代替, 即可完成转换。

例如：把 $(1011000001101.0100101)_2$ 转换成十六进制数。

$$\begin{array}{ccccccc} (0001 & 0110 & 0000 & 1101 & .0100 & 1010)_2 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ (1 & 6 & 0 & D & . & 4 & A)_{16} \end{array}$$

$$\text{则 } (1011000001101.0100101)_2 = (160D.4A)_{16}$$

(2) 十六进制数转换成二进制数

将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 $(18A3.5E)_{16}$ 转换成二进制数。

$$\begin{array}{ccccccc} (1 & 8 & A & 3 & . & 5 & E)_{16} \\ \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ (0001 & 1000 & 1010 & 0011 & . & 0101 & 1110)_2 \end{array}$$

$$\text{则 } (18A3.5E)_{16} = (1100010100011.0101111)_2$$

4. 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

对于二进制整数，只要自右到左将每三位二进制数分为一组，不足三位时，在左面添0，补足三位；对于二进制小数，只要自左到右将每三位二进制数分为一组，不足三位时，在右面添0，补足三位，然后将每组用相应的八进制数代替，即可完成转换。

例如：把 $(101110111.0100101)_2$ 转换成八进制数。

$$\begin{array}{ccccccc} (101 & 110 & 111 & . & 010 & 010 & 100)_2 \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\ (5 & 6 & 7 & . & 2 & 2 & 4)_8 \end{array}$$

$$\text{则 } (101110111.0100101)_2 = (567.224)_8$$

(2) 八进制数转换成二进制数

将八进制数转换成二进制数，只要将每一位八进制数用三位相应的二进制数表示即可完成转换。

例如：将 $(1703.53)_8$ 转换成二进制数。

$$\begin{array}{ccccccc} (1 & 7 & 0 & 3 & . & 5 & 3)_8 \\ \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ (001 & 111 & 000 & 011 & . & 101 & 011)_2 \end{array}$$

$$\text{则 } (1703.53)_8 = (1111000011.101011)_2$$

3.2 数值在计算机中的表示

3.2.1 数值编码

1. 机器数

在日常生活中我们用到的数有正有负，在计算机中，使用的是二进制数，只有0和1两

种值。那么在计算机中如何表示正负数呢？把+、-号也用0和1表示（0表示正号，1表示负号）放在存储器的最高位，数值的绝对值放在除符号位以外的部分，这种数值在计算机中的二进制表示形式，称为机器数，如图3-1所示。



图3-1 在计算机中用一个字节表示一个数示意图

机器数也有不同的表示法，常用的有3种：原码、补码和反码。

2. 真值

机器数所对应的原来的数值称为该机器数的真值。如机器数10011100的真值为-28（-0011100）。

在数的表示中，机器数与真值的区别是：真值带符号如-0011100（-28），真值在机器中的表示形式为机器数10011100。

例如：真值数为+0111001(+57)，其对应的机器数为00111001，其中最高位为0，表示该数为正数。

3. 原码、反码和补码

机器数中，数值和符号全部数字化。计算机在进行数值运算时，采用把各种符号位和数值位一起编码的方法。常见的有原码、补码和反码表示法。

（1）原码

原码表示法是机器数的一种简单的表示法。其符号位用0表示正号，用1表示负号，数值一般用二进制形式表示。

设有一数为X，则原码表示可记作 $[x]_{\text{原}}$ 。

例如， $X_1=+0001010$

$X_2=-0001010$

其原码记作：

$[X_1]_{\text{原}}=[+0001010]_{\text{原}}=00001010$

$[X_2]_{\text{原}}=[-1001010]_{\text{原}}=10001010$

原码表示数值的范围与二进制位数有关。当用8位二进制原码来表示整数时，其表示范围：

最大值为01111111，其真值为 $(127)_{10}$ 或为 $(+1111111)_2$ ；

最小值为11111111，其真值为 $(-127)_{10}$ 或为 $(-1111111)_2$ 。

在原码表示法中，对0有两种表示形式：

$[+0]_{\text{原}}=00000000$

$[-0]_{\text{原}}=10000000$

（2）反码

机器数的反码可由原码得到。如果机器数是正数，则该机器数的反码与原码一样；如果机器数是负数，则该机器数的反码是对它的原码（符号位除外）各位取反而得到的。

设有一数X，则X的反码表示记作 $[X]_{\text{反}}$ 。

例如: $X_1=+0000110$ (+6)

$X_2=-1000110$ (-6)

那么 $[X_1]_{\text{原}}=00000110$

$[X_1]_{\text{反}}=[X_1]_{\text{原}}=00000110$

$[X_2]_{\text{原}}=10000110$

$[X_2]_{\text{反}}=11111001$

当用 8 位二进制反码来表示整数时, 其表示范围为: $-127 \sim +127$ 。

在反码表示法中, 对 0 有两种表示形式:

$[+0]_{\text{原}}=00000000$

$[-0]_{\text{反}}=11111111$

(3) 补码

机器数的补码可由反码得到。如果机器数是正数, 则该机器数的补码与反码相同, 当然也与原码一样; 如果机器数是负数, 则该机器数的补码是在其反的末位上加 1。

设有一数 X , 则 X 的码表示记作 $[X]_{\text{补}}$ 。

例如:

$[+6]_{\text{补}}=[+6]_{\text{反}}=[+6]_{\text{原}}=00000110$

$[-6]_{\text{原}}=10000110$

$[-6]_{\text{反}}=11111001$

$[-6]_{\text{补}}=11111010$

$$\begin{array}{r} 11111001 \\ + \quad \quad 1 \\ \hline 11111010 \end{array}$$

在补码表示法中, 对 0 的表示只有一种, 即:

$[+0]_{\text{补}}=00000000$

$[-0]_{\text{补}}=00000000$

当用 8 位二进制补码来表示整数时, 其表示范围为:

最大为 01111111 , 其真值为 $(127)_{10}$ 或为 $(+1111111)_2$;

最小为 10000000 , 其真值为 $(-128)_{10}$ 或为 $(-10000000)_2$ 。

补码在微型机中是一种重要的编码形式, 请注意:

1) 采用补码后, 可以方便地将减法运算转化成加法运算, 运算过程得到简化。正数的补码即是它所表示的数值的真值, 而负数的补码的数值部份却不是它所表示的数的真值。采用补码进行运算, 所得结果仍为补码。

2) 与原码、反码不同, 数值 0 的补码只有一个, 即 $[0]_{\text{补}}=00000000$ 。

$[+0]_{\text{原}}=00000000$

$[-0]_{\text{反}}=11111111$

11111111

$$\begin{array}{r} + \quad \quad 1 \\ \hline \boxed{1}0000000 \end{array}$$

$[-0]_{\text{补}}=00000000$ 。

3) 若字长为8位, 则补码所表示的范围为-128~+127; 进行补码运算时, 应注意所得结果不应超过补码所能表示数的范围。

设有一数X, 则X的补码表示记作 $[X]_{\text{补}}$ 。

例如, $X_1=+1110110$

$X_2=-1101010$

那么, $[X_1]_{\text{原}}=01110110$

$[X_1]_{\text{补}}=01110110$

即 $[X_1]_{\text{原}}=[X_1]_{\text{补}}=01110110$

$[X_2]_{\text{原}}=11101010$

$[X_2]_{\text{反}}=10010101$

$[X_2]_{\text{补}}=10010101+1=10010110$

【例 3-4】 已知 $[X]_{\text{原}}=10111010$, 求 $[X]_{\text{补}}$ 。

分析如下:

由 $[X]_{\text{原}}$ 求 $[X]_{\text{补}}$ 的原则是: 若机器数为正数, 则 $[X]_{\text{原}}=[X]_{\text{补}}$; 若机器数为负数, 则该机器数的补码可对它的原码(符号位除外)所有位求反, 再在末位加1而得到。现给定的机器数为负数, 故有 $[X]_{\text{补}}=[X]_{\text{反}}+1$, 即

$$\begin{array}{r} [X]_{\text{原}}=10111010 \\ [X]_{\text{反}}=11000101 \\ \quad 11000101 \\ + \quad \quad 1 \\ \hline \quad 11000110 \\ [X]_{\text{补}}=11000110 \end{array}$$

【例 3-5】 已知 $[X]_{\text{补}}=11110110$, 求 $[X]_{\text{原}}$ 。

分析如下:

对于机器数为正数, 则 $[X]_{\text{原}}=[X]_{\text{补}}$;

对于机器数为负数, 则有 $[X]_{\text{原}}=[[X]_{\text{补}}]_{\text{补}}$;

现给定的为负数, 故有:

$$\begin{array}{r} [X]_{\text{补}}=11110110 \\ [[X]_{\text{补}}]_{\text{反}}=10001001 \\ \quad 10001001 \\ + \quad \quad 1 \\ \hline \quad 10001010 \\ [[X]_{\text{补}}]_{\text{补}}=10001010=[X]_{\text{原}} \end{array}$$

用补码表示法可以使减法运算成为加法运算。例如 4-9 的运算如下:

$$\begin{array}{r} 4 \text{ 的补码形式为:} \quad \quad 00000100 \\ -9 \text{ 的补码形式为:} \quad + \quad 11110111 \\ \hline \quad \quad \quad 11111011 \end{array}$$

结果为 11111011, 是-5 的补码形式。

又如: $(-9) + (-4)$ 的运算如下:

$$\begin{array}{r} -9 \text{ 的补码形式为: } \quad 11110111 \\ -4 \text{ 的补码形式为: } \quad + \quad 11111100 \\ \hline \boxed{1}1110011 \end{array}$$

最高位的 1 自动丢失, 运算结果为 11110011, 是-13 的补码形式。

由上可见, 利用补码可以方便地进行运算, 在数的有效范围内, 符号位如同数值一样参加运算, 也允许产生最高位的进位 (补自动丢失), 所以使用较广泛。

3.2.2 数值在计算机中的表示

在计算机中, 一般用若干个二进制位表示一个数或一条指令, 把它们作为一个整体来处理、存储和传送。这种作为一个整体来处理的二进制位串, 称为计算机字。表示数据的字称为数据字, 表示指令的字称为指令字。

计算机是以字为单位进行处理、存储和传送的, 所以运算器中的加法器、累加器以及其他一些寄存器, 都选择与字长相同位数。字长一定, 则计算机数据字所能表示的数的范围也就确定了。例如, 使用 16 位字长的计算机, 它可以表示无符号整数的最大值是 $(65535)_{10} = (1111111111111111)_2$ 。运算时, 若数值超出机器数所能表示的范围, 就会停止运算和处理, 这种现象称为溢出。

1. 定点数

计算机中运算的数, 有整数, 也有小数, 如何表示整数和小数呢? 那就是规定小数点的位置固定不变, 这时的机器数称为定点数。

(1) 定点整数

定点整数就是将小数点的位置固定在存储器的末端, 这时机器数所表示的数就是纯整数。假设机器字长为 16 位, 符号位占 1 位, 数值部分占 15 位, 于是机器数: 0111111111111111 所表示的整数为十进制的 +32767, 如图 3-2 所示。

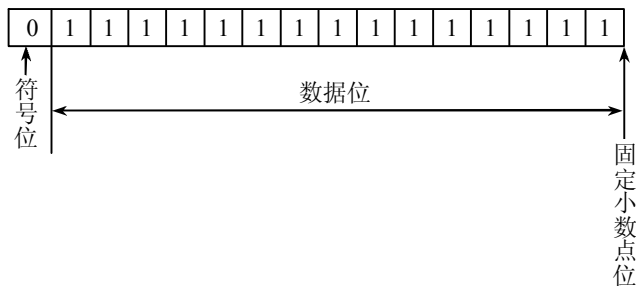


图 3-2 定点整数在计算机中的表示示意图

16 位定点整数补码表示数的范围:

最大为 0111111111111111, 其真值为 $(2^{15})_{10} = (32767)_{10}$;

最小为 1000000000000000, 其真值为 $(-32768)_{10}$ 。

(2) 定点小数

定点小数就是将小数点的位置固定在符号位的右端, 数值部分的左端, 这时机器数所表

示的数就是纯小数。假设机器字长为 16 位，符号位占 1 位，数值部分占 15 位，于是机器数 0111111111111111（即 0.111111111111111_2 ）所表示的纯小数为十进制的 $+(1-2^{-15})$ ，如图 3-3 所示。

16 位定点小数补码表示数的范围：

最大为 0.111111111111111_2 ，其真值为 $(1-2^{-15})_{10}$ ；

最小为 1.000000000000000_2 ，其真值为 $(-1)_{10}$ 。

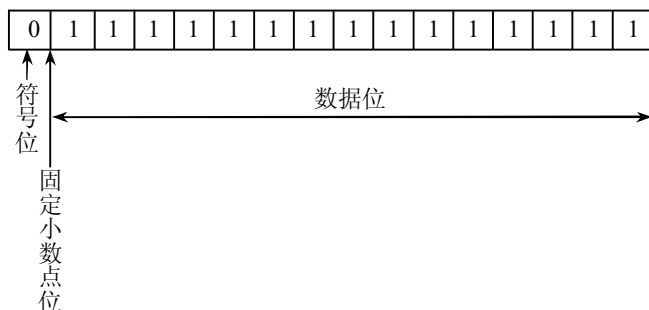


图 3-3 定点小数在计算机中的表示示意图

定点表示法所能表示的数值范围很有限，为了扩大定点数的表示范围，可以通过编程技术，采用多个字节来表示一个定点数，例如，采用 4 个字节或 8 个字节等。

2. 浮点数

在计算机中处理的数，除整数外，大多处理的是实数，即既带整数又带小数的数。在计算机中，为了能表示实数，采用浮点表示法。在同样字长的情况下，浮点表示法能表示的数的范围扩大了，浮点表示法就是小数点在数中的位置是浮动的，浮点表示法由两部分组成，即尾数和阶码。

例如， 0.3125×10^3 ，则 0.3125 为尾数，3 是阶码。

在浮点表示法中，小数点的位置是浮动的，阶码可以取不同的数值。

例如，十进制数 -3125.8125 可表示为： $-3.1258125 \times 10^{+3}$ 、 -3125.8125×10^0 、 $-3125812.5 \times 10^{-3}$ 等多种形式。

为了便于计算机中小数点的表示，规定将浮点数写成规格化的形式，即尾数的绝对值大于等于 0.1 并且小于 1，从而唯一地规定了小数点的位置。

十进制数 -3125.8125 以规格化形式表示为： -0.31258125×10^4 。

对于任意二进制数在计算机中的浮点表示法包括两个部分：一部分是阶码（表示指数，记作 E）；另一部分是尾数（表示有效数字，记作 M）。设任意一数 N 可以表示为：

$$N = M2^E$$

其中 2 为基数，E 为阶码，M 为尾数。

假设机器字长为 16 位，阶码部分用一个字节表示，尾数部分用一个字节表示，则机器数 0000011001000001 所表示的数为十进制的 $+(2^{-1}+2^{-7})2^6$ ，即 +32.5，如图 3-4 所示。

其中：

阶码 $E = (00000110)_2 = (6)_{10}$

尾数 $M = (0.1000001)_2 = (0.5078125)_{10}$

数值 $N = M2^E = 0.5078125 \times 2^6 = (32.5)_{10}$

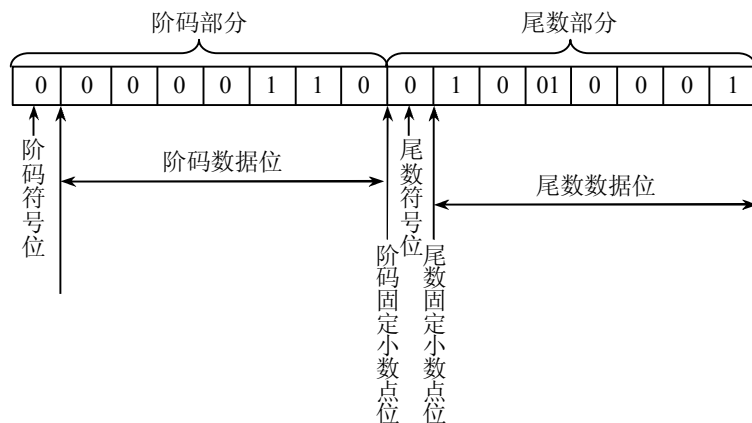


图 3-4 浮点数在计算机中的表示示意图

由尾数部分隐含的小数点位置可知，尾数总是 ≤ 1 的数字，它给出该浮点数的有效数字。尾数部分的符号位确定该浮点数的正负。阶码给出的总是整数，它确定小数点浮动的位数，若阶符为正，则相当于小数点向右移动；若阶符为负，则相当于小数点向左移动。

浮点数表示法对尾数有如下规定：

$$1/2 \leq M < 1$$

即要求尾数中第 1 位数不为零，这样的浮点数称为规格化的数。

当浮点数的尾数为零或者阶码为最小值时，机器通常规定，把该数看作零，称为“机器零”。浮点数表示和运算中，当一个数的阶码大于机器所能表示的最大阶码时，产生“上溢”。上溢时机器一般不再继续运算而转入“溢出”处理。当一个数的阶码小于机器所能表示的最小阶码、最小尾数或绝对值最小尾数时，“下溢”时一般当作机器零来处理。

若机器字长为 32 位，阶码部分用一个字节表示，尾数部分用三个字节表示，则机器能表示的数的范围为 $-1.7 \times 10^{38} \sim +1.7 \times 10^{38}$ 。

3. BCD 码

在计算机中，一般用户和计算机之间的输入和输出数都是十进制，由于计算机内用的都是二进制数，这就需要进行十进制和二进制之间的相互转换，这项工作由计算机本身自动完成。在计算机中采用了输入/输出转换的二至十进制编码，即 BCD 码。在二至十进制的转换中，用 4 位二进制数表示 1 位十进制数的编码方法。最常用的是 8421BCD 码。“8421”的含义是指 4 位二进制数从左到右每位对应的权是 8、4、2、1。BCD 码和十进制之间的对应关系如表 3-2 所示。

表 3-2 BCD 码和十进制数对照表

十进制数	0	1	2	3	4	5	6	7	8	9
BCD 码	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

例如：十进制数 785.5 用 BCD 码表示的二进制数为：(0111 1000 0101.0101)BCD。

3.3 字符在计算机中的表示

计算机中使用的数据有数值型数据和非数值型数据两大类。数值数据用于表示数量意义；

非数值数据又称为符号数据,包括字母和符号等。非数值数据用于记录事物信息。计算机除处理数值信息外,大量处理的是字符信息。例如,将用高级语言编写的程序输入到计算机时,人与计算机通信时所用的语言就不再是一种纯数字语言而是字符语言。由于计算机中只能存储二进制数,这就需要对字符进行编码,建立字符数据与二进制串之间的对应关系,以便于计算机识别、存储和处理。

3.3.1 英文字符在计算机中的表示

1. ASCII 字符

计算机中用得最多的符号数据是字符,它是用户和计算机之间的桥梁。用户使用计算机的输入设备,按下键盘上的字符键向计算机内输入命令和数据,计算机把处理后的结果也以字符的形式输出到屏幕或打印机等输出设备上。对于字符的编码方案有很多种,但使用最广泛的是 ASCII 码(American Standard Code for Information Interchange, ASCII)。ASCII 码是美国国家信息交换标准字符码,后来被采纳为一种国际通用的信息交换标准代码。ASCII 码就是用 7 位二进制编码表示一个英文字符,从 0000000 到 1111111 共有 128 种编码,可用来表示 128 个不同的字符。ASCII 码由 0~9 这 10 个数字符号,52 个大、小写英文字母,32 个符号及 34 个计算机通用控制符组成,共有 128 个元素,如表 3-3 所示。ASCII 码表的查表方式是:先查列(高三位即下表的 d6d5d4),后查行(低四位 d3d2d1d0),然后按从左到右的书写顺序完成,如 B 的 ASCII 码为 1000010。在 ASCII 码进行存放时,由于它的编码是 7 位,因 1 个字节(8 位)是计算机中常用单位,故仍以 1 字节来存放 1 个 ASCII 字符,每个字节中多余的最高位取 0。

表 3-3 ASCII 字符编码表

d6d5d4 d3d2d1d0	000	001	010	011	100	101	110	111
0000	NUL	DEL	SP	0	@	P	`	~
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	EXT	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

由表 3-4 可知, ASCII 码字符可分为两大类:

(1) 打印字符: 即从键盘输入并显示的 95 个字符, 如大、小写英文字母各 26 个 (小写字母的 ASCII 值比大写字母的 ASCII 值大 32), 数字 0~9 这 10 个数字字符的高 3 位编码 (d6d5d4) 为 011, 低 4 位为 0000~1001。当去掉高 3 位时, 低 4 位正好是二进制形式的 0~9。

(2) 不可打印字符: 共 33 个, 其编码值为 $0\sim 31(0000000\sim 0011111)_2$ 和 $(1111111)_2$, 不对应任何可印刷字符。不可打印字符通常为控制符, 用于计算机通信中的通信控制或对设备的功能控制。如编码值为 127 (1111111), 是删除控制 DEL 码, 它用于删除光标之后的字符。ASCII 码字符的码值可用 7 位二进制代码或 2 位十六进制来表示。例如字母 D 的 ASCII 码值为 $(1000100)_2$ 或 84H, 数字 4 的码值为 $(0110100)_2$ 或 34H 等。

计算机内部存储与操作常以字节为单位, 即 8 个二进制位为单位。因此一个英文字符在计算机内实际是用 8 位表示。在正常情况下, 最高位 d7 为 0。在需要奇偶校验时, 这一位可用于存放奇偶校验的值, 此时称为校验位。

2. 扩展 ASCII 码字符

计算机内部存储与操作常以字节为单位, 而 ASCII 码字符是用 7 位二进制数表示一个字符, 在一个字节中, 高位全为 0, 这就造成一个二进制位的浪费。因此有的单位或部门就对 ASCII 码字符进行了扩展, 用 8 位二进制数表示一个英文字符, 这就是扩展 ASCII 码字符。

EBCDIC 就是典型的扩展 ASCII 码字符。EBCDIC (Extended Binary Coded Decimal Interchange Code) 为国际商用机器公司 (IBM) 于 1963~1964 年间推出的字符编码表, 根据早期打孔机式的二进位十进数 (BCD, Binary Coded Decimal) 排列而成。

在了解了数值与英文字符在计算机内的表示后, 读者可能会产生一个问题: 二者在计算机内都是二进制数, 如何区分数值和英文字符呢?

例如, 内存中存放一个二进制数: 01000010, 它究竟表示数值 66, 还是表示字母 “B”? 而对一个孤立的字节, 确实无法区分, 但存放和使用这个数据的软件, 会以其他方式保存有关的类型信息, 指明这个信息是何种类型。以下汉字信息也是如此。

3.3.2 汉字字符在计算机中的表示

汉字是汉语书写的最基本单元, 其使用最晚始于商代, 历经甲骨文、大篆、小篆、隶书、楷书 (草书、行书) 诸般书体变化。秦始皇统一中国, 李斯整理小篆, “书同文” 的历史从此开始。尽管汉语方言发音差异很大, 但是书写系统的统一减少了方言差异造成的交流障碍。

三千余年来, 汉字的书写方式变化不大, 使得后人得以阅读古文而不生窒碍。但近代西方文明进入东亚之后, 整个汉字文化圈的各个国家纷纷掀起了学习西方的思潮, 其中, 放弃使用汉字是这场运动的一个重要方面。这些运动的立论以为: 跟西方拼音文字相比, 汉字是繁琐笨拙的。许多使用汉字的国家即进行了不同程度的汉字简化, 甚至还有完全拼音化的尝试。日文假名的拉丁转写方案以及汉语多种拼音方案的出现都是基于这种思想。中国大陆将汉字笔划参考行书草书加以省简, 于 1956 年 1 月 28 日审订通过《简化字总表》, 在中国及新加坡使用至今。中国香港、台湾地区则一直使用繁体中文。

目前在使用汉语的地区, 大都使用两种规范汉字, 分别是繁体中文 (繁体字) 和简体中文 (简体字)。

英文文字是拼音文字, 所有文字均由 26 个字母拼组而成, 所以使用一个字节表示一个字

符足够了。但汉字是象形文字，汉字的计算机处理技术比英文字符复杂得多，一般用两个字节表示一个汉字。

1. 汉字输入码

汉字输入码也叫外码，是为了通过键盘字符把汉字输入到计算机中而设计的一种编码。英文输入时，想输入什么字符便按什么键，输入码和内码是一致的。而汉字输入则不同，可能要按几个键才能输入一个汉字。汉字和键盘字符组合的对应方式称为汉字输入编码方案。外码是针对不同汉字输入法而言的，通过键盘按某种输入法进行汉字输入时，人与计算机进行信息交换所用的编码称为“汉字外码”。对于同一汉字而言，输入法不同，其外码也是不同的。例如，对于汉字“啊”，在区位码输入法中的外码是 1601，在拼音输入中的外码是 a，而在五笔字型输入法中的外码是 kbsk。汉字的输入码一般采用英文小写字母编码，所以输入汉字时，键盘应处在小写状态。汉字的输入码种类繁多，可分以下 4 种类型，音码、形码、音形码、形音码。

衡量一个输入码的优劣标准如下：

- 编码短，可以减少击键次数；
- 重码少，可以实现盲打；
- 好学好记，便于学习和掌握。

但到目前为止，还没有一种全部符合上述要求的汉字输入方法。

(1) 音码

音码输入码主要是以汉语拼音为基础进行拼音输入的编码方案，如全拼、简拼、智能 ABC、搜狗输入法等。优点是简单易学，甚至不需要再学习，与人们习惯一致。缺点是汉字同音多，即重码字多，输入汉字时要选字，则输入速度慢。搜狗输入法可以以词组为单位，很好地弥补了重码字多、输入速度慢的不足。

(2) 形码

形码输入码主要是根据汉语的特点，按汉字固有的形状，把汉字先拆分成部首，然后组合进行拼形输入的编码方案，如首尾码输入法、五笔字型输入法、郑码输入法等。

优点是不考虑字的读音，见字识码，重码字少，输入速度快，强化训练后可以实现盲打。缺点是拆字法没有统一的国家标准，拆字难，编码规则繁，记忆量大，需要学习和记忆。五笔字型输入法使用广泛，适合专业录入人员，可实现盲打，但必须记住字根、掌握汉字的间架结构和书写顺序，学会拆字和形成编码。

(3) 音形码

音形码输入码集拼音与拼形于一体，以拼音为主，兼顾拼形的编码方案，如音形码、自然码、钱码等。优点是它兼容了拼音、拼形输入的优点，大大减少了重码汉字，提高了输入速度。缺点是既要考虑字音，又要考虑字形，编码比较麻烦。

(4) 形音码

形音码输入码集拼形与拼音于一体，以拼形为主，兼顾拼音形的编码方案，它兼容了拼形、拼音输入的优点，几乎没有重码汉字。典型代表是形音码输入法，兼容了五笔字输入法和拼音输入法，并且对两种输入法进行适当调整的一种编码。

(5) 数字码

数字码也称流水码，它是用 0~9 十个数字字符进行编码的编码方案，如电报码、区位码

等。优点是无重码，不仅能对汉字编码，还能对各种字母、数字符号进行编码。缺点是人为规定的编码，属于无理码，只能作为专业人员使用。

(6) 基于模式识别的智能化输入法

模式识别与智能系统是在信号处理、人工智能、控制论、计算机技术等学科基础上发展起来的新型学科。该学科以各种传感器为信息源，以信息处理与模式识别的理论技术为核心，以数学方法与计算机为主要工具，探索对各种媒体信息进行处理、分类、理解，并在此基础上构造具有某些智能特性的系统或装置的方法、途径与实现，以提高系统性能。基于模式识别的智能化汉字输入法就是在此基础上产生的，如语音识别输入、手写输入法、扫描输入法等。

现在汉字输入法还在进一步研究中，将来有一天，汉字输入不再是人们使用计算机的瓶颈，人们坐在电脑前，畅游在计算机世界里，尽情享受计算机世界带给我们的快乐与便利。

不管哪种汉字输入法，都是操作者向计算机输入汉字的手段，而在计算机内部都是以汉字机内码表示和存储的。

2. 汉字国标码

汉字国标码也叫汉字交换码，主要是用作汉字信息交换的。1980年，为了使每个汉字有一个全国统一的代码，我国颁布了汉字编码的国家标准：GB2312—80《信息交换用汉字编码字符集（基本集）》，这就是国标码，简称GB码。这个字符集是我国中文信息处理技术的发展基础，也是目前国内所有汉字系统的统一标准。国标码是一个四位十六进制数，区位码是一个四位的十进制数，每个国标码或区位码都对应着一个唯一的汉字或符号，但因为十六进制数我们很少用到，所以大家常用的是区位码，它的前两位叫做区码，后两位叫做位码。

在国标GB2312—80中有6763个常用汉字。国标GB2312规定，所有的国标汉字与符号组成一个 94×94 的矩阵。在此方阵中，每一行称为一个“区”，每一列称为一个“位”，因此，这个方阵实际上组成了一个有94个区（区号分别为01~94）、每个区内有94个位（位号分别为01~94）的汉字字符集。一个汉字所在的区号和位号简单地组合在一起就构成了该汉字的“区位码”。在汉字的区位码中，高两位为区号，低两位为位号。由此可见，区位码与汉字或符号之间是一一对应的。区号对应第一字节，位号对应第二字节。01~09区为符号、数字区，16~87区为汉字区，10~15区、88~94区是有待进一步标准化的空白区。GB2312将收录的汉字分成两级：第一级是常用汉字计3755个，置于16~55区，按汉语拼音字母/笔形顺序排列；第二级汉字是次常用汉字计3008个，置于56~87区，按部首/笔画顺序排列。

汉字区位码并不等于国标码，国标码是由区位码稍作转换得到的，其转换方法为：将十进制区码和位码各加32（或将十进制区码和位码转换为十六进制的区码和位码，再将区码和位码各加十六进制的20H）就构成了国标码。如“啊”字的区位码为1601，则“啊”字的国标码为4833（3021H），它是经过下面的转换得到的： $16+32=48$ ， $01+32=33$ （或 $(16)_{10}=(10H)_{16}$ ， $(01)_{10}=(01H)_{16}$ ， $10H+20H=30H$ ， $01H+20H=21H$ ）。

3. 汉字机内码

汉字机内码，简称“内码”，指计算机内部存储、处理加工和传输汉字时所用的二进制编码。输入码被接受后就由汉字操作系统的“输入码转换模块”转换为机内码，与所采用的键盘输入法无关。机内码是汉字最基本的编码，不管是什么汉字系统和汉字输入方法，输入的汉字外码到机器内部都要转换成机内码，才能被存储和进行各种处理。

国标码是汉字信息交换的标准编码，但因其前后字节的最高位为0，与ASCII码发生冲突，

如“啊”字，国标码为 30H 和 21H，而英文字符“0”和“!”的 ASCII 码也为 30H 和 21H，现假如内存中有两个字节为 30H 和 21H，这到底是一个汉字，还是两个英文字符“0”和“!”呢？于是就出现了二义性，显然，国标码是不可能在计算机内部直接采用的，于是，汉字的机内码采用变形国标码，其变换方法为：将国标码的每个字节都加上十进制 128(十六进制 80H)，即将两个字节的最高位由 0 改 1，其余 7 位不变，如：由上面我们知道，“啊”字的国标码为 3021H，前字节为 $(00110000)_2$ ，后字节为 $(00100001)_2$ ，高位改 1 为 $(10110000)_2$ ，后字节为 $(10100001)_2$ ，即为 $(1011000010100001)_2$ ，即为 B0A1H。因此，汉字“啊”的机内码就是 B0A1H。

请大家自己计算一下“中”字的机内码。

显然，汉字机内码的每个字节都大于 128，这就解决了与西文字符的 ASCII 码冲突的问题。如上所述，汉字输入码、区位码、国标码与机内码都是汉字的编码形式，它们之间有着千丝万缕的联系，但其间的区别也是不容忽视的。

4. 汉字字型码

汉字字型码又称汉字字模，用于汉字在显示屏或打印机输出。汉字字型码通常有两种表示方式：点阵和矢量表示方法。

用点阵表示字型时，汉字字型码指的是这个汉字字型点阵的代码。根据输出汉字的要求不同，点阵的多少也不同。简易型汉字为 16×16 点阵，提高型汉字为 24×24 点阵、 32×32 点阵、 48×48 点阵等，如图 3-5 所示。

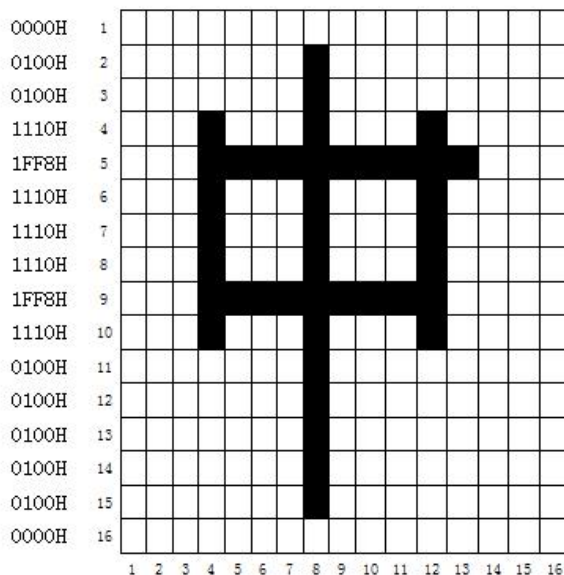


图 3-5 16×16 点阵汉字字形码示意图

汉字字形点阵规模愈大，字型愈清晰美观，所占存储空间也愈大。以 16×16 点阵为例，存储一个汉字要占 $16 \times 16 / 8 = 32$ 个字节。

把所有汉字字形所构成的集合叫汉字库，汉字库分为点阵字库和矢量字库。点阵字模占的存储空间较大，而且不同的点阵字模要有不同的汉字字库，所以汉字字库一般在外存存放，当显示输出汉字时才检索字库，输出相应的字模得到相应的字形。

矢量表示方式存储的是描述汉字字型的轮廓特征, 当要输出汉字时, 通过计算机的计算, 由汉字字型描述生成所需大小和形状的汉字点阵。矢量化字型描述与最终文字显示的大小、分辨率无关, 因此可以产生高质量的汉字输出。Windows 中使用的 TrueType 技术就是汉字的矢量表示方式。

True Type (简称 TT) 是由美国 Apple 公司和 Microsoft 公司联合提出的一种新型数字化字形描述技术。TT 是一种彩色数字函数描述字体轮廓外形的一套内容丰富的指令集合, 这些指令中包括字型构造、颜色填充、数字描述函数、流程条件控制、栅格处理器 (TT 处理器) 控制、附加提示信息控制等指令。

TT 采用几何学中的二次 B 样条曲线及直线来描述字体的外形轮廓, 二次 B 样条曲线具有一阶连续性和正切连续性。抛物线可由二次 B 样条曲线来精确表示, 更为复杂的字体外形可用 B 样条曲线的数学特性以数条相接的二次 B 样条曲线及直线来表示。描述 TT 字体的文件 (内含 TT 字体描述信息、指令集、各种标记表格等) 可能通用于 Mac 和 PC 平台。在 Mac 平台上, 它以 “Sfnt” 资源的形式存放, 在 Windows 平台上以 TTF 文件出现。为保证 TT 的跨平台兼容性, 字体文件的数据格式采用 Motorola 式数据结构 (高位在前, 低位在后) 存放。所有 Intel 平台的 TT 解释器在执行之前, 只要进行适当的预处理即可。Windows 的 TT 解释器已包含在其 GDI (图形设备接口) 中, 所以任何 Windows 支持的输出设备, 都能用 TT 字体输出。

TT 技术具有以下优势:

(1) 真正的所见即所得效果。由于 TT 支持几乎所有的输出设备, 因而对于目标输出设备而言, 无论系统的屏幕、激光打印机或激光照排机, 所有在操作系统中安装了 TT 字体均能在输出设备上以指定的分辨率输出, 所以多数排版类应用程序可以根据当前目标输出设备的分辨率等参数, 来对页面进行精确的布局。

(2) 支持字体嵌入技术, 保证文件的跨系统传递性。TT 嵌入技术解决了跨系统间的文件和字体的一致性问题。在应用程序中, 存盘的文件可将文件中使用的所有 TT 字体采用嵌入方式一并存入文件。使整个文件及其所使用的字体可方便地传递到其他计算机的同一系统中使用。字体嵌入技术保证了接收该文件的计算机即使未安装所传送文件使用的字体, 也可通过装载随文件一同嵌入的 TT 字体来对文件进行保持原格式, 使用原字体进行打印和修改。

(3) 操作系统平台的兼容性。Mac 和 Windows 平台均提供系统级的 TT 支持。所以在不同操作系统平台间的同名应用程序文件有跨平台兼容性。如 Mac 机上的 PageMaker 可以使用在如果已安装了文件中所用的所有 TT 字体, 则该文件在 Mac 上产生的最终输出效果将与在 Windows 下的输出保持高度一致。

(4) ABC 字宽值。在 TT 字体中的每个字符都有其各自的字宽值, TT 所用的字宽描述方法比传统的 PS 的 Type 1 更为完善和科学。它采用 ABC 字宽表, 即将传统上的一个字符的字宽值划分为三部分: A 宽度为在放置字符轮廓前的空白间距, B 宽度为字符轮廓本身的间距, C 为字符轮廓右方的空白间距。A+B+C 即相当于传统的字宽值, 同时 A 或 C 间距可以分别为负值, 以产生特殊的排字效果。采用 ABC 字宽值可避免传统上使用整字字宽值而引起的累积舍入误差。同时可以避免由此产生的行尾的最后一个字符移至下一行, 或行左首字符无法对齐等输出时不可预料的后果。排版应用程序在计算一行的累积字宽时所产生的舍入误差, 可以分散到整行中每一字符的 A 和 C 间距中, 保证了断行的高度准确性和用户定义的左右边界尺寸

的精确性。

在 Windows 98 以上系统中, 系统使用得最多的就是*.TTF (True Type) 轮廓字库文件, 它既能显示也能打印, 并且支持无极变倍, 在任何情况下都不会出现锯齿问题。而*.FOT 则是与*.TTF 文件对应的字体资源文件, 它是TTF 字体文件的资源指针, 指明了系统所使用的 TTF 文件的具体位置, 而不用必须指定到 FONTS 文件夹中。*.FNT (矢量字库) 和*.FON (显示字库) 的应用范围都比较广泛。另外, 那些使用过老版本的 WPS 的用户可能对*.PS 文件还有一定的印象, *.PS 实际上是 DOS 下轮廓字库的一种形式, 其性能与*.TTF 基本类似, 采用某些特殊方法之后, 我们甚至还可以实现在 Windows 中直接使用这些*.PS 字库 (*.PS1、*.PS2 都是 PS 字库)。

点阵字形与矢量字形的区别是: 点阵字形编码、存储方式简单, 显示和打印汉字时不需要转换, 直接在字库中取字形输出即可, 但字形放大后产生的效果差, 而且同一种字体不同的点阵需要不同的汉字库。矢量字形正好与前者相反。

5. 汉字地址码

每个汉字字形码在汉字字库中的相对位移地址称为汉字地址码。需要向输出设备输出汉字时, 必须通过地址码, 才能在汉字字库中找到所需要的汉字字形码位置, 取出所需要的字形, 最终在输出设备上形成可见的汉字字形。为了能快速地找到所需要的汉字字形码, 地址码和机内码要有简明的对应转换关系。

综上所述, 在一个汉字处理系统中, 输入、内部处理和输出汉字, 对汉字的编码要求也不一样, 因此要进行一系列的汉字编码及转换。汉字信息处理中各编码及流程如图 3-6 所示。

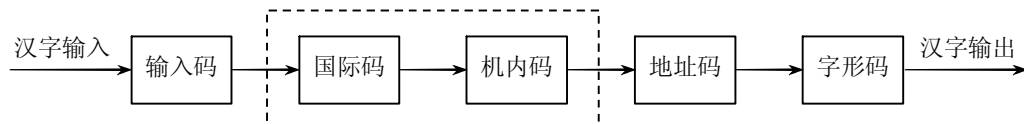


图 3-6 汉字信息处理系统模型

6. 其他汉字编码

除 GB 码外, 目前常用的还有: UCS 码、Unicode 码、GBK、Big-5 码等。

(1) UCS 码

为了统一表示世界各国、各地区的文字, 便于全球范围的信息交流, 1993 年, 国际标准化组织公布了名为“通用多八位编码字集”的国际标准 ISO/IEC 10646 (Universal Multiple-Octet Coded Character Set, UCS)。UCS 为世界上正在使用的各种文字, 规定了统一的编码方案, 为多文种的信息交换和信息处理创造了基本条件。UCS 是所有其他字符集标准的一个超集。它保证与其他字符集是双向兼容的, 也就是说, 如果将任何文本字符串翻译到 UCS 格式, 然后再翻译回原编码, 不会丢失任何信息。

UCS 整个字符集可以容纳 128 组, 每组有 256 个平面, 每个平面有 256 行, 每行有 256 个字位, 这样大的空间完全包容世界上的所有文字。全世界的字符在 UCS 中用 4 个字节 (组号、平面号、行号、和字位号) 唯一地表示。

第一个平面 (00 组中的 00 平面) 称为基本多文种平面 (BMP)。它包含字母文字、音节文字和表意文字等。它分为四个区: A、I、O、R。

A 区：19903 个字位，用于字母文字、音节文字和各种符号；

I 区：20992 个字位，用于中、日、韩（CJK）统一的表意文字；

O 区：16384 个字位，留于未来标准化用。

R 区：8190 个字位，作为基本多文种平面的限制使用区，它包括专用字符、兼容字符等。

例如，ASCII 码字符“A”，它的 ASCII 码为 41H，它在 UCS 中的编码为 00000041H。汉字“大”，它在 GB2312 中编码为 3473H，它在 UCS 中的编码为 00005927H。

UCS 的实际表现形式为 UTF-8/UTF-16/UTF-32 编码。

（2）Unicode 码

Unicode 是另一个国际编码标准，采用双字节表示世界上的主要文字，其字符集与 UCS 的 BMP 相同。内容包含符号 6811 个、汉字 20902 个（其中：大陆提出的汉字 17124 个，台湾提出的汉字 17258 个，并集后共提出的汉字为 20158 个，未提出字符共 744 个）。韩文拼音 11172 个，造字区 6400 个，保留 20249 个，共计 65534 个。

由于 UCS 中每个字符需要用 4 个字节表示，消耗的存储空间也就多。事实上全世界正在使用的各种文字，经常使用的只是其中的一部分，其数量不足 65536 (2^{16})，因此，UCS 字符集中最前面一部分，即 0 组的 0 号平面被规定为“基本多文种平面”（BMP），用于放置全世界的主要文字、音节文字和包括控制符在内的各种符号。“基本多文种平面”的字符，其 4 字节代码的前 2 字节都是 0，在绝大多数应用场合，可以用后 2 个字节作为其代码。这就构成了 Unicode 代码，以 2 字节统一表示世界上的主要文字，得到越来越广泛的应用。支持 Unicode 编码的相关电脑系统软件，如 UNIX、Windows 已有推出，但是由于 Unicode 的 ASCII 码是用双字节编码（即一般电脑系统中的单字节 ASCII 码前加 0x00），同时其汉字编码与各国的现有编码也不兼容，造成现有的软件和数据不能直接使用，所以目前完全使用 Unicode 软件系统的用户并不多，大多数只将它作为一个国际语言编码标准来使用。

（3）GBK 码

GBK 全称《汉字内码扩展规范》（GBK 即“国标”“扩展”汉语拼音的第一个字母，英文名称：Chinese Internal Code Specification），中华人民共和国全国信息技术标准化技术委员会于 1995 年 12 月 1 日制订，国家技术监督局标准化司、电子工业部科技与质量监督司于 1995 年 12 月 15 日联合以技监标函〔1995〕229 号文件的形式，将它确定为技术规范指导性文件。这一版的 GBK 规范为 1.0 版。

GBK 向下与 GB 2312 编码兼容，向上支持 ISO 10646.1 国际标准，是前者向后者过渡过程中的一个承上启下的产物。我国 1993 年发布的 GB 13000.1 国家标准等同于 UCS。

GBK 编码是在 GB2312—80 标准基础上的内码扩展规范，使用了双字节编码方案，其编码范围从 8140 至 FEFE（剔除 xx7F），共 23940 个码位，收录了 21003 个汉字，882 个符号，共计 21885 个字符。完全兼容 GB2312—80 标准，支持国际标准 ISO/IEC10646—1 和国家标准 GB13000—1 中的全部中日韩汉字，并包含了 Big-5 编码中的所有汉字。目前中文版的 Windows 等都支持 GBK 编码方案。

全拼、紫光拼音等输入法，能够录入如“镨、镨、炆、央、喆、嘉、姤、赆、龔、眈、堃、慙、靖、殲”等 GBK 简繁体汉字。

（4）Big-5 码

Big 是在 1984 年由台湾地区 13 家厂商与台湾地区财团法人信息工业策进会为五大中

文套装软件（宏碁、神通、佳佳、零壹、大众）所设计的中文内码，所以就称为 Big-5 中文内码，虽然五大套装软件并没有成功，但 Big-5 码却深远地影响了中文电脑内码，直至今日。“五大码”的英文名称“Big-5”后来被人按英文字序译回中文，以致现在有“五大码”和“大五码”两个中文名称。

Big-5 码，是通行于中国台湾、香港地区的一个繁体字编码方案。大五码是使用繁体中文社群中最常用的电脑汉字字符集标准，共收录 13060 个中文字。Big-5 虽普及于中国的台湾、香港与澳门等繁体中文通行区，但长期以来并非当地的国家标准，而只是业界标准。倚天中文系统、Windows 等主要系统的字符集都是以 Big-5 为基准，但厂商又各自增删，衍生出多种不同版本。

2003 年，Big-5 被收录到台湾官方标准的附录当中，取得了较正式的地位。这个最新版本被称为 Big-5—2003。

（5）关于汉字编码

为进行信息交换，各汉字使用地区都制订了一系列汉字字符集标准。

1) GB2312—80 字符集，收入 6763 个汉字，682 个符号，共计 7445 个字符。这个字符集是我国中文信息处理技术的发展基础，也是目前国内所有汉字系统的统一基础标准。

2) GB2313 字符集，收入汉字 6763 个，符号 715 个，总计 7478 个字符，这是大陆普遍使用的简体字符集。楷体-GB2313、仿宋-GB2313、华文行楷等市面上绝大多数字体支持显示这个字符集，亦是大多数输入法所采用的字符集。市面上绝大多数所谓的繁体字体，其实采用的是 GB-2313 字符集简体字的编码，用字体显示为繁体字，而不是直接用 GBK 字符集中繁体字的编码，错误百出。

3) BIG-5 字符集，收入 13060 个繁体汉字，808 个符号，总计 13868 个字符，目前普遍使用于中国的台湾、香港等地区。台湾教育部标准宋体、楷体等港台大多数字体支持这个字符集的显示。

4) GBK 字符集，又称大字符集（GB=GuóBiāo 国标，K=扩展），包含以上两种字符集汉字，收入 21003 个汉字，882 个符号，共计 21885 个字符，包括了中日韩（CJK）统一汉字 20902 个、扩展 A 集（CJK Ext-A）中的汉字 52 个。Windows 95\98 简体中文版就带有这个 GBK.txt 文件。宋体、隶书、黑体、幼圆、华文中宋、华文细黑、华文楷体、标楷体（DFKai-SB）、Arial Unicode MS、MingLiU、PMingLiU 等字体支持显示这个字符集。

Big-5（繁体中文）与 GB2313（简体中文）编码不相兼容，字符在不同的操作系统中会产生乱码。文本文字的简体与繁体（文字及编码）之间的转换，可用 BabelPad、TextPro 或 Convertz 之类的转码软件来解决。若是程序，如 Windows XP 操作系统，可用 Microsoft AppLocale Utility 1.0 解决；Windows 2000 的操作系统，可用：中文之星、四通利方、南极星、金山快译之类的转码软件解决。

5) GB18030 字符集，是我国继 GB2312—1980 和 GB13000.1—1993 之后最重要的汉字编码标准，是我国计算机系统必须遵循的基础性标准之一。GB18030 有两个版本：GB18030—2000 和 GB18030—2005。GB18030—2000 是 GBK 的取代版本，它的主要特点是在 GBK 基础上增加了 CJK 统一汉字扩充 A 的汉字。GB18030—2005 的主要特点是在 GB18030—2000 基础上增加了 CJK 统一汉字扩充 B 的汉字。它还去掉了单字节编码的欧元符号 80H。

GB18030 有 1611668 个码位，在 GB18030—2005 中定义了 76556 个字符。随着我国汉字

整理和编码研究工作的不断深入,以及国际标准 ISO/IEC 10646 的不断发展,GB18030 所收录的字符将在新版本中增加。

6) 方正超大字符集,包含 GB18030 字符集、CJK Ext-B 中的 36862 个汉字,共计 64395 个汉字。宋体-方正超大字符集支持这个字符集的显示。Microsoft Office XP 或 2003 简体中文版就自带这个字体。Windows 2000 的操作系统需安装超大字符集支持包“Surrogate 更新”。

7) ISO/IEC 10646 / Unicode 字符集,这是全球可以共享的编码字符集,两者相互兼容,涵盖了世界上主要语文的字符,其中包括简繁体汉字,计有:CJK 统一汉字 20902 个,CJK Ext-A 6582 个,Ext-B 42711 个,共计 70195 个汉字。SimSun-ExtB(宋体)、MingLiU-ExtB(细明体)能显示全部 Ext-B 汉字。至今尚无单独一款字体能够显示全部 70195 个汉字,但可用海峰五笔、新概念五笔、仓颉输入法世纪版、新版的微软新注音、仓颉输入法 6.0 版(单码功能)等输入法录入。Ext-C 还有 2 万多个汉字。详情请参阅香港中文大学网站、马来西亚仓颉之友网站、福建陈清钰个人网站。

8) 汉字构形数据库 2.3 版,内含楷书字形 60082 个、小篆 11100 个、楚系简帛文字 2627 个、金文 3459 个、甲骨文 177 个、异体字 12768 组。可以安装该程序,亦可以解压后使用其中的字体文件,对于整理某些古代文献十分有用。

如果超出了输入法所支持的字符集,就不能录入计算机。如果没有相应字体的支持,则显示为黑框、方框或空白。如果操作系统或应用软件不支持该字符集,则显示为问号(一个或两个)。在网页上亦存在同样的情况。

3.4 图形和图像在计算机中的表示

3.4.1 图形在计算机中的表示

1. 图形

在计算机中图形和图像是两个不同的概念。图形一般是指通过绘图软件绘制的由直线、圆、圆弧、矩形、任意曲线等几何图形组成的画面。

2. 图形在计算机中的表示

图形是一种矢量图。矢量图是用数学的方式来描述一幅图形。矢量图形的描述包括形状、色彩、位置等。矢量图形本身就用数字化形式来表述。图形的最大优点在于可以分别控制处理图中的各个部分,如在屏幕上移动、旋转、放大、缩小而不失真,不同的物体还可以在屏幕上重叠并保持各自的特性,必要时仍可分开。但对于一幅复杂的彩色照片,很难用数学来描述,因此也难以用矢量图来表示。

3.4.2 图像在计算机中的表示

1. 图像

图像是由扫描仪、照像机、摄像机等输入的画面。形一般是指通过绘图软件绘制的由直线、圆、圆弧、矩形、任意曲线等几何图形组成的画面。

2. 静态图像在计算机中的表示

自然景物成像后可以复制成照片、录像带进行保存,这样的图像称为模拟图像。它们是

连续的，不能直接用计算机进行处理，还需要进一步转化处理。

静态图像在计算机中是用位图来表示的，位图也叫点阵图像，是由若干个排列成行、列的点构成，这些点称为像素（Pixels），用以描述图像中各点的明暗强度与颜色。每个像素点的亮度和颜色信息用若干数据位来表示，这些数据位的个数称为图像的颜色深度或灰度级。

图像的数字化就是将模拟图像转化成位图的过程，通常包括采样和量化。采样就是将连续的模拟图像变换成离散点的操作过程。在这个过程中，采样间隔是个重要参数，它决定了在一定的面积内取多少个像素点。在同等面积内，像素点数越多，分辨率就越高，图像质量越好。

采样后，图像成为离散的像素点，但其灰度仍是连续的。还需要将像素的灰度转换成离散的整数值，即量化。若一幅数字图像的量化灰度级为 8 bit，就可以有 $2^8=256$ 个灰度级差。颜色深度与显示的颜色数目之间的关系如表 3-4 所示。

表 3-4 颜色深度与显示的颜色数目之间的关系

颜色深度（bit）	颜色总数	图像名称
1	2 (2^1)	单色图像（黑白二色）
4	16 (2^4)	索引 16 色图像
8	256 (2^8)	索引 256 色图像
16	65536 (2^{16})	HI-Color 图像（实际只显示 32768 种颜色）
24	16777216 (2^{24})	True Color 图像（真彩色）

图像的分辨率越高、颜色深度越深，则数字化后的图像效果越逼真，图像数据量也越大。但是，经采样、量化形成的图像丢掉了部分数据，与模拟图像必然有一定的差距。但这个差距通常控制得相当小，以至人的肉眼难以分辨，人们可以将数字化图像等同于模拟图像。

3. 动画和视频

动画和视频是图形和图像的动态形式。动态图像是由一系列的静态画面按一定的顺序排列组成，并配以同步的声音。每一个静态画面称为一“帧”，当每秒以 24 帧的速度播放时，由于视觉的暂留现象产生动态效果。

动态的图像有动画和视频两种方式。动画的每一幅画面是通过工具软件（如 Flash 等）对图像素材进行编辑制作而成；而视频影像是对信号源（如电视机、摄像机等）进行采样和量化后制作而成。动画是用人工合成的方法对真实世界的一种模拟，而视频影像则是对真实世界的记录。

4. 图像压缩技术

随着数字化时代的发展，需要存储、传输和处理的信息的数量成指数级地增加。图像作为数字信息的重要组成部分，是信息交流的重要载体，也是蕴含信息量最大的媒体。因此图像压缩作为图像处理的一个重要组成部分，一直是人们研究的一个热点。

（1）图像压缩的基本原理

图像数据之所以能被压缩，就是因为数据中存在着冗余。图像数据的冗余主要表现为：图像中相邻像素间的相关性引起的空间冗余；图像序列中不同帧之间存在相关性引起的时间冗余；不同彩色平面或频谱带的相关性引起的频谱冗余。数据压缩的目的就是通过去除这些数据冗余来减少表示数据所需的比特数。由于图像数据量的庞大，在存储、传输、处理时非常困难，

因此图像数据的压缩就显得非常重要。

信息时代带来了“信息爆炸”，使数据量大增，因此，无论传输或存储都需要对数据进行有效的压缩。在遥感技术中，各种航天探测器采用压缩编码技术，将获取的巨大信息送回地面。

图像压缩是数据压缩技术在数字图像上的应用，它的目的是减少图像数据中的冗余信息从而用更加高效的格式存储和传输数据。

(2) 图像压缩基本方法

图像压缩可以是有损数据压缩也可以是无损数据压缩。对于如绘制的技术图、图表或者漫画优先使用无损压缩，这是因为有损压缩方法，尤其是在低的位速条件下将会带来压缩失真。如医疗图像或者用于存档的扫描图像等这些有价值的内容的压缩也尽量选择无损压缩方法。有损方法非常适合于自然的图像，例如一些应用中图像的微小损失是可以接受的（有时是无法感知的），这样就可以大幅度地减小位速。

(3) 图像压缩的标准

经典的视频压缩算法已渐形成一系列的国际标准体系，如 H.26x 系列建议、H.320 系列建议、JPEG 以及 MPEG 系列等标准。

(4) 衡量数据压缩技术的重要指标

- 压缩比，即压缩前后所需的信息存储之比要大。
- 恢复效果，即要尽可能恢复到原始数据。
- 压缩、解压速度，尤其解压速度更为重要，因为解压是实时的。
- 实现压缩的软、硬件开销要小。

3.4.3 常用的图像文件格式

在图形图像数据处理中，可用于图形图像文件存储的格式很多，现分类列出常用的文件格式。

1. 静态图像格式

(1) JPEG 格式

JPEG 是由联合照片专家组（Joint Photographic Experts Group）开发的。它既是一种文件格式，又是一种压缩技术。JPEG 作为一种很灵活的格式，具有调节图像质量的功能，允许用不同的压缩比例对这种文件压缩。作为先进的压缩技术，它用有损压缩方式去除冗余的图像和彩色数据，在获取极高的压缩率的同时能展现十分丰富生动的图像。JPEG 格式是现在使用最为广泛的格式之一，也是目前最优秀的数字化摄影图像的存储方式，JPEG 还是万维网中图像处理时使用的主要文件格式之一。

(2) PSD 格式

这种图像文件格式是 Photoshop 7.0 默认的格式。其文件格式后缀为 PSD 和 PDD。它是由 Adobe 公司专门开发的适用于 Photoshop、ImageReady 的图像格式。它支持从 BMP 到 CMYK 的所有色彩模型类型，是最能体现 Photoshop 7.0 功能与特征的优化格式。这种格式已经得到了广泛的应用。它支持 Photoshop 7.0 特有的一些图像信息，包括能够存储图层、蒙板通道和其他的图像信息，而且这种格式的文件是进行了压缩的，其压缩比例和 JPEG 差不多，并且压缩后不失真，不会影响到图像的质量。

(3) BMP 格式

BMP (Bitmap) 是 Windows 操作系统中的标准图像文件格式。这种格式的特点是包含的图像信息较丰富, 几乎不进行压缩, 但文件占用了较大的存储空间。BMP 格式支持 RGB、索引颜色、灰度和位图颜色模式 A。基本上绝大多数图像处理软件都支持此格式。

在 Windows 系统中系统所用的大部分图像都是以该格式保存的, 如墙纸图像、屏幕保护图像等。BMP 格式图像的文件可以使用 RLE (Run Length Encoding, 运行长度编码) 的无损压缩方案进行压缩。

(4) GIF 格式

GIF (Graphics Interchange Format) 是 CompuServe 公司开发的图像文件格式, 采用了压缩存储技术。GIF 格式同时支持线图、灰度和索引图像, 但最多支持 256 种色彩的图像。GIF 格式的特点是压缩比高, 磁盘空间占用较少、下载速度快。GIF 文件内部分成许多存储块, 用来存储多幅图像或者是决定图像表现行为的控制块, 用以实现动画和交互式应用。

2. 动态图像格式

数字化视频的数据量巨大, 通常采用特定的压缩算法对数据进行压缩, 根据压缩算法的不同, 保存数字化视频的常用格式如下:

(1) MPEG/MPG

MPEG (Moving Picture Experts Group) 是 1988 年成立的一个专家组。1991 年制定了一个 MPEG-1 国际标准, 采用的算法简称为 MPEG 算法, 是目前最常见的视频压缩方式, 可对包括声音在内的运动图像进行压缩。最大压缩可达约 1:200, 用该算法压缩的数据称为 MPEG 数据, 由该数据产生的文件称 MPEG 文件, 常以 MPG 为后缀, 部分采用 MPEG 格式压缩的视频文件也以 DAT 为扩展名。

(2) AVI

AVI (Audio Video Interleaved) 是一种音频视像交叉记录的数字视频文件格式。1992 年初, Microsoft 公司推出了 AVI 技术, 是对视频文件采用的一种有损压缩方式。在 AVI 文件中, 运动图像和伴音数据是以交替的方式进行存储的。这种组织数据的方式使得读取视频数据流时能更有效地从存储媒介得到连续的信息。

(3) MOV

MOV 是 Apple (苹果) 公司创立的一种视频格式, 它是图像及视频处理软件 QuickTime 所支持的格式, 在很长的一段时间里, 它都只是在苹果公司的 Mac 机上存在, 随着个人多媒体电脑近几年的飞速普及, Apple 公司不失时机地推出了 QuickTime 的 Windows 版本。QuickTime 能够通过 Internet 提供实时的数字化信息流、工作流与文件回放功能。

(4) ASF

ASF (Advanced Streaming Format) 是微软公司推出的高级流媒体格式, 也是一个在 Internet 上实时传播多媒体的技术标准。由于它使用了 MPEG-4 的压缩算法, 所以压缩率和图像的质量都很不错。它应用的主要部件是 NetShow 服务器和 NetShow 播放器。将媒体信息编译成 ASF 流后, 发送到 NetShow 服务器, 再由 NetShow 服务器发送给网络上的所有 NetShow 播放器, 从而实现单路广播或多路广播。

(5) RM

RM 格式是 Real Networks 公司开发的一种新型流式视频文件格式, 又称 Real Media, 是

目前 Internet 上最流行的跨平台的客户/服务器结构多媒体应用标准,它采用音频/视频流和同步回放技术,实现了网上全带宽的多媒体回放。RealPlayer 就是在网上收听收看这些实时音频、视频和动画的最佳工具。只要用户的线路允许,使用 RealPlayer 可以不必下载完音频/视频内容就能实现网络在线播放,更容易上网查找和收听、收看各种广播、电视。

3.5 声音在计算机中的表示

3.5.1 声音在计算机中的表示

声音是通过空气的震动发出,通常用模拟波的方式表示它。振幅反映声音的音量,频率反映了音调。音频是连续变化的模拟信号,而计算机只能处理数字信号,要使计算机能处理音频信号,必须把模拟音频信号转换成用“0”“1”表示的数字信号,这就是音频的数字化,将模拟信号数字化,会涉及到采样、量化及编码等多种技术。音频信号的数字化过程如图 3-7 所示。

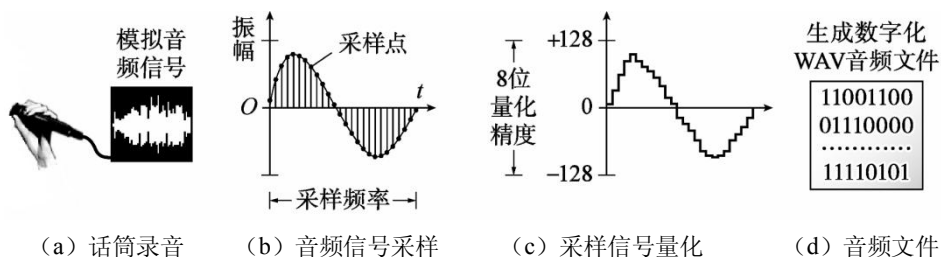


图 3-7 音频信号的数字化过程

每隔一个时间间隔在模拟声音波形上取一个幅度值,称为采样,采样频率是指一秒钟内采样的次数。采样频率越高,声音的保真度越好。根据采样定理,采样频率应不低于声音信号最高频率的两倍。人耳能听到声音的频率范围大约为 20Hz~20 kHz,所以要得到高保真效果,采样频率应大于 40 kHz。

量化是用数字化的方法反映采样后的声波信号幅度值的大小。量化时采用的二进制数的位数称为量化位数,也叫采样精度或采样位数。它决定了模拟信号数字化以后声音的动态范围。量化位数越多,音效也就越好。采样时的量化是通过声卡中 A/D 模数转换器来实现的。量化位数有一般有 8 位、16 位等。表 3-5 是常见声音信号的采样频率和量化精度。

表 3-5 常见声音信号的采样频率和量化精度

信号类型	采样频率/kHz	量化精度/位
电话	8	8
调幅广播	11.025	8
调频广播	22.05	16
数字激光唱盘	44.1	16

经过量化后得到的数字信息，还必须按一定格式转换成计算机可以识别的二进制形式，才能在计算机中保存。用二进制形式表示量化值的过程称为编码。

3.5.2 声音文件格式

数字音频信息在计算机中是以文件的形式保存，存储声音信息的文件可以有多种格式，如 WAV、MIDI、MP3、WMA 等。

1. WAV 格式

WAV 格式的文件又称波形文件，是用不同的采样率对声音的模拟波形进行采样得到的一系列离散的采样点，以不同的量化位数（16 位、32 位或 64 位）把这些采样点的值转换成二进制数得到的。WAV 是数字音频技术中最常用的格式，它还原的音质较好，被称为“无损的音乐”，但所需存储空间较大，是目前 PC 机上广为流行的声音文件格式，几乎所有的音频编辑软件都能够读取 WAV 格式。

2. MIDI 格式

MIDI 是 Musical Instrument Digital Interface（乐器数字接口）的缩写。它是由世界上主要电子乐器制造厂商建立起来的一个通信标准，并于 1988 年正式提交给 MIDI 制造商协会，成为数字音乐的一个国际标准。MIDI 标准使不同厂家生产的电子合成乐器可以互相发送和接收音乐数据。MIDI 文件记录的是一系列指令而不是数字化后的波形数据，所以它占用存储空间比 Wav 文件要小很多。一个 MIDI 文件每存 1 分钟的音乐只用大约 5~10KB。

3. MPEG 格式

MPEG 标准主要有以下五个，MPEG-1、MPEG-2、MPEG-4、MPEG-7 及 MPEG-21 等。该专家组建于 1988 年，专门负责为 CD 建立视频和音频标准，而成员都是视频、音频及系统领域的技术专家。及后，他们成功将声音和影像的记录脱离了传统的模拟方式，建立了 ISO/IEC 1172 压缩编码标准，并制定出 MPEG 格式，令视听传播方面进入了数码化时代。因此，大家现时泛指的 MPEG-X 版本，就是由 ISO（International Organization for Standardization）所制定而发布的视频、音频、数据的压缩标准。MPEG 标准的视频压缩编码技术主要利用了具有运动补偿的帧间压缩编码技术以减小时间冗余度，利用 DCT 技术以减小图像的空间冗余度，利用熵编码则在信息表示方面减小了统计冗余度。这几种技术的综合运用，大大增强了压缩性能。

MPEG 3 是对 MPEG Layer 3 的简称，是当前使用最广泛的数字化声音格式。其技术采用 MPEG Layer 3 标准对 WAVE 音频文件进行压缩而成，特点是能以较小的比特率、较大的压缩率达到近乎 CD 音质。MPEG 音频文件的压缩是一种有损压缩，其压缩率可达 10:1~12:1，由于其文件尺寸小、音质好，所以网上音乐大量使用 MP3 格式。

4. WMA 格式

WMA（Windows Media Audio）是微软自己开发的 Windows Media Audio 技术。它支持流式播放。WMA 格式的可保护性极强，甚至可以限定播放机器、播放时间及播放次数，具有相当的版权保护能力，它比起 MP3 压缩技术，WMA 无论从技术性能（支持音频流）还是压缩率（比 MP3 高一倍）都超过了 MP3 格式。用它来制作接近 CD 品质的音频文件，其体积仅相当于 MP3 的 1/3。

本章小结

在计算机科学中,数据是指所有能输入到计算机并被计算机程序处理的符号的总称,是用于输入电子计算机进行处理,具有一定意义的数字、字母、符号和模拟量等的通称。数据的种类很多,无论什么数据在计算机中都是用二进制代码表示的。

把数值的正、负号也用 0、1 表示放在存储器的最高位形成的数叫机器数,该机器数所对应原来的数值叫做该机器数的真值。机器数常用的有原码、反码、补码表示法。

本章较详细地介绍了英文字符、汉字、图形、图像和声音在计算机中的表示。

习题 3

一、选择题

- 十进制数 1024 转换成二进制数是 ()。

A. 1000000000	B. 10000000000
C. 1111111110	D. 11111111110
- 十进制数 127 对应的八进制数是 ()。

A. 117	B. 771	C. 87	A. 177
--------	--------	-------	--------
- 十进制数 234 对应的八进制数是 ()。

A. 270	B. 462	C. 352	D. 264
--------	--------	--------	--------
- 十进制数 37 转换为无符号二进制整数是 ()。

A. 100101	B. 101101	C. 101001	D. 110100
-----------	-----------	-----------	-----------
- 十六进制数 1021 转换成十进制数是 ()。

A. 4096	B. 1024	C. 4129	D. 8192
---------	---------	---------	---------
- 十六进制数 2B9 可表示成 ()。

A. 2B90	B. 2B9E	C. 2B9F	D. 2B9H
---------	---------	---------	---------
- 十六进制数 365 对应的八进制数为 ()。

A. 3022	B. 1702	C. 1545	D. 3072
---------	---------	---------	---------
- 十六进制数 5C 对应的十进制数为 ()。

A. 92	B. 93	C. 75	D. 90
-------	-------	-------	-------
- 十六进制数 FFFH 转换为二进制数是 ()。

A. 11111111111	B. 101010101010
C. 10101010101	D. 100010001000
- 十六进制数值 BD 等值的十进制数是 ()。

A. 156	B. 177	C. 196	D. 189
--------	--------	--------	--------
- 下列一组数中最小的数是 ()。

A. $(11001001)_2$	B. $(205)_{10}$	C. $(312)_8$	D. C8H
-------------------	-----------------	--------------	--------
- 下列一组数中最小的数是 ()。

- A. $(11011001)_2$ B. $(36)_{10}$ C. $(37)_8$ D. 3AH
13. 原码 01010111 的反码是 ()。
- A. 10101000 B. 10000001 C. 10000000 D. 01010111
14. 字符 5 的 ASCII 表示为 ()。
- A. 1100101 B. 10100011 C. 1000101 D. 110101
15. 二进制数真值+1110111 的补码是 ()。
- A. 11000111 B. 01110111 C. 11010111 D. 00101010
16. 二进制数真值-1010111 的补码是 ()。
- A. 00101001 B. 11000010 C. 11100101 D. 10101001
17. 字符的 ASCII 编码在机器中的表示方法准确地描述是 ()。
- A. 8 位二进制代码, 最右 1 位为 1
B. 8 位二进制代码, 最左 1 位为 0
C. 8 位二进制代码, 最右 1 位为 0
D. 8 位二进制代码, 最左 1 位为 1
18. 字符的标准 ASCII 编码是用 () 位二进制表示的。
- A. 8 B. 7 C. 16 D. 15
19. 字母 A 的 ASCII 表示为 ()。
- A. 11000001 B. 110001 C. 10000001 D. 1000001
20. 在数字视频信息获取与处理过程中, 下述顺序 () 是正确的。
- A. A/D 变换、采样、压缩、存储、解压缩、D/A 变换
B. 采样、压缩、A/D 变换、存储、解压缩、D/A 变换
C. 采样、A/D 变换、压缩、存储、解压缩、D/A 变换
D. 采样、D/A 变换、压缩、存储、解压缩、A/D 变换
21. 下列采样的波形声音质量最好的是 ()。
- A. 单声道、8 位量化、44.1kHz 采样频率
B. 双声道、8 位量化、22.05kHz 采样频率
C. 双声道、16 位量化、44.1kHz 采样频率
D. 单声道、16 位量化、22.05kHz 采样频率
22. 以下属于多媒体动态图像文件格式的是 ()。
- A. AVI B. MP3 C. JPG D. MID

二、简答题

1. 一个字节是由几位二进制数组成的? 字和字长的含义是什么?
2. 什么是机器数和真值?
3. 数据压缩的基本方法有几种?
4. 构成位图图像的最基本单位是什么?
5. 请举例说明图像采样的方法。
6. 试举出几种常见的图像文件格式, 并简述其特点。
7. 试举出几种常见的声音文件格式, 并简述其特点。

三、计算题

1. 进行下列数的数制转换：

- (1) 将 598.76 用 BCD 码表示。
- (2) 将 $(255)_{10}$ 转换为二、八、十六进制数。
- (3) 将 $(127.365)_{10}$ 转换为二、八、十六进制数。
- (4) 将 $(1100111.01110)_2$ 转换为十、八、十六进制数。
- (5) 将 $(3ABC.6D)_{16}$ 转换为二、八、十进制数。
- (6) 将 $(1110000111)_2$ 转换为八、十、十六进制数。
- (7) 将 $(10100101)_2$ 转换为八、十、十六进制数。
- (8) 将 $(4CA2)_{16}$ 转换为二、八、十进制数。

2. 请查出“A”“C”和“9”的 ASCII 码值。

3. 256MB 等于多少字节？

4. 已知 $X=40$, $Y=-50$, 求 $[X+Y]_{\text{补}}$ 及 $X+Y$ 。

5. 已知汉字“张”的区位码是：5337（十进制），求它的国标码和机内码。