第1章 计算机中的数据和编码

1.1 计算机中的数制

1.1.1 进位计数制

人们在长期的社会生产活动和日常生活的过程中，形成了各种数制。按照进位的方法进行计数的数制称为进位计数制，简称进位制。人们习惯使用的数制是十进制，在计算机中采用的数制是二进制，为便于计算机信息的书写，常常采用十六进制。为了避免数制的混淆，可在数字的后面加填区分符，区分符可以用字母表示。二进制数的区分符用字母 B 表示，十进制数的区分符用字母 D 表示或不用区分符，十六进制数的区分符用字母 H 表示。例如，二进制数 1011.11B，十进制数 123.45D 或 123.45，十六进制数 3BA.4H。

1. 基数

数制是以表示数值所用符号的个数来命名的，表明计数制允许选用的基本数码的个数称为基数，用 R 表示。例如，最常用的十进制数，每个数位上允许选用 0, 1, 2, …, 9，共 10 个不同数码，它的基数 R=10；二进制数，每个数位上允许选用 0 和 1，它的基数 R=2；十六进制数，每个数位上允许选用 0, 1, 2, 3, … 9, A, … F，共 16 个不同数码，它的基数 R=16。表 1.1 是计算机中的数制对照表。

表 1.1 计算机中的数制对照表

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

2．权

在进位计数制中，一个数码处在数的不同位置时，它所代表的数值是不同的。每一个数位赋予的数值称为位权，简称权。权的大小是以基数为底，数位的序号为指数的整数次幂，用 i 表示数位的序号，用 R 表示数位的权。例如，342.54 各数位的权分别为 10^2, 10^1, 10^0, 10^{-1} 和 10^{-2}；1011.01B 各数位的权分别为 2^3, 2^2, 2^1, 2^0, 2^{-1} 和 2^{-2}；34A.7H 各数位的权分别为 16^2, 16^1, 16^0 和 16^{-1}。

3．进位计数制的按权展开式

进位计数制中，每个数位的数值等于该位数码与该位的权之乘积，用 K_i 表示第 i 位的系数，则该位的数值为 K_iR^i。任意进位制的数都可以写成按权展开的多项式和的形式，其一般表达为：

$$N=\sum_{i=n-1}^{-m}K_iR^i$$
$$=K_{n-1}R^{n-1}+K_{n-2}R^{n-2}+\cdots+K_0R^0+K_{-1}R^{-1}+\cdots+K_{-m}R^{-m}$$

（n 是进位制整数部分的位数，m 是进位制小数部分的位数）

例如：

$$345.75=\sum_{i=2}^{-2}K_i10^i$$
$$=3\times10^2+4\times10^1+5\times10^0+7\times10^{-1}+5\times10^{-2}$$
$$1101.11=\sum_{i=3}^{-2}K_i2^i$$
$$=1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}$$
$$=13.75$$
$$4F8.BH=\sum_{i=2}^{-1}K_i16^i$$
$$=4\times16^2+15\times16^1+8\times16^0+11\times16^{-1}$$
$$=1272.6875$$

1.1.2 进位计数制的相互转换

1．二进制数转换成十进制数

二进制数转换成十进制数，因转换算法不同分为整数转换和小数转换两种方法。

（1）整数转换法

写出二进制整数的按权展开式如下：

$$N=K_{n-1}\times2^{n-1}+K_{n-2}\times2^{n-2}+\cdots+K_0\times2^0$$

把上式改写成下式：

$$N=(((K_{n-1}\times2+K_{n-2})\times2+K_{n-3})\times2+\cdots+K_1)\times2+K_0$$

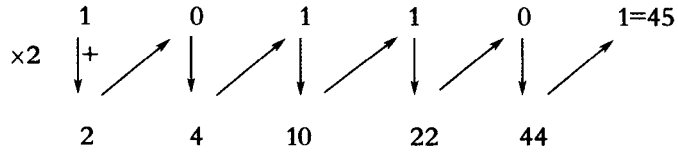
从上述表达式，得出转换方法如下：

从最高位开始乘以 2，加上次高位，再乘以 2，加上第三高位，……依此方法一直加到

最低位为止。二进制整数转换成十进制整数的方法称为乘 2 叠加法。

【例 1.1】 把二进制数 101101 转换成十进制数。

转换过程用线图表示：



转换结果是：101101B = 45

（2）小数转换法：写出二进制小数的按权展开式如下：

$$N = K_{-1}\times 2^{-1}+K_{-2}\times 2^{-2}+\cdots+K_{-m}\times 2^{-m}$$

把上式改写成下式：

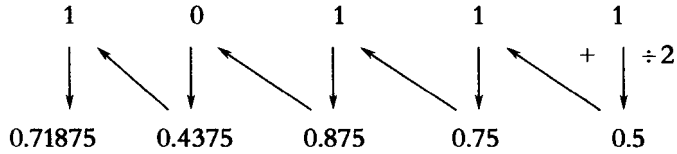
$$N = 2^{-1}\left(K_{-1}+2^{-1}\left(K_{-2}+\cdots+2^{-1}\left(K_{-m+1}+2^{-1}K_{-m}\right)\right)\right)$$

从上述表达式，得出转换方法如下：

从最低位开始，除以 2，加上次低位，再除以 2，加上第三低位，……依此方法一直到小数点后第一位除以 2 为止。二进制小数转换成十进制小数的方法称为除 2 叠加法。

【例 1.2】 把二进制数 0.10111 转换成十进制小数。

转换过程用线图表示：



转换结果是：0.10111B = 0.71875

2. 十进制数转换成二进制数

十进制数转换成二进制数，因转换算法不同也分为整数转换和小数转换两种方法。

（1）整数转换法

下面举例说明转换方法。

【例 1.3】 把十进制数 205 转换成二进制整数。

十进制数 205 转换成二进制数的数位系数是：

$$205 = (K_{n-1}K_{n-2}\cdots K_1K_0)\,B$$

将二进制整数写成按权展开式：

$$205 = K_{n-1}\times 2^{n-1}+K_{n-2}\times 2^{n-2}+\cdots+K_1\times 2^1+K_0\times 2^0$$

等式右边二进制数提取公因子 2:

$$205 = 2\left(K_{n-1}\times 2^{n-2}+K_{n-2}\times 2^{n-3}+\cdots+K_1\times 2^0\right)+K_0$$

等式两边同除以 2 得到：

$$102+1/2 = K_{n-1}\times 2^{n-2}+K_{n-2}\times 2^{n-3}+\cdots+K_1\times 2^0+K_0/2$$

上述等式两边的 1/2 和 $K_0/2$ 相等，即 K_0=1。

等式右边二进制数，再次提取公因子 2:

$$102 = 2（K_{n-1} \times 2^{n-3} + K_{n-2} \times 2^{n-4} + \cdots + K_2 \times 2^0）+ K_1$$

等式两边再同除以 2 得到：

$$51 = K_{n-1} \times 2^{n-3} + K_{n-2} \times 2^{n-4} + \cdots + K_2 \times 2^0 + K_1 / 2$$

上式左边除以 2 后余数为 0，即 $K_1=0$。如此进行下去直到等式左边为 0 停止，每次除以 2 的余数，即是二进制数各数位上的系数。

根据上例，得出转换方法如下：

把十进制数的整数部分连续除以 2，依次取得余数，直到商为 0 停止，依次得出的余数序列即是二进制数从低位到高位各数位上的系数。十进制整数转换为二进制整数的方法称为除 2 取余法。

上例用竖式表示如下：

十进制整数／2	二进制数位系数 = 余数
205 / 2 = 102	$K_0 = 1$
102 / 2 = 51	$K_1 = 0$
51 / 2 = 25	$K_2 = 1$
25 / 2 = 12	$K_3 = 1$
12 / 2 = 6	$K_4 = 0$
6 / 2 = 3	$K_5 = 0$
3 / 2 = 1	$K_6 = 1$
1 / 2 = 0	$K_7 = 1$

转换结果是：205 = 11001101B。

上例用线图表示如下：

									÷2
0	← 1	← 3	← 6	← 12	← 25	← 51	← 102	← 205	
	↓	↓	↓	↓	↓	↓	↓	↓	
余数	1	1	0	0	1	1	0	1	

（2）小数转换法

下面举例说明转换方法。

【例 1.4】 把十进制小数 0.8125 转换成二进制小数。

十进制小数 0.8125 转换成二进制小数的数位系数是：

$$0.8125 = （0.K_{-1} K_{-2} \cdots K_{-m}）\ \mathrm{B}$$

将二进制小数写成按权展开式：

$$0.8125 = K_{-1} \times 2^{-1} + K_{-2} \times 2^{-2} \cdots + K_{-m} \times 2^{-m}$$

等式两边乘以 2 得到：

$$1.6250 = K_{-1} \times 2^0 + K_{-2} \times 2^{-1} \cdots + K_{-m} \times 2^{-m+1}$$

等式两边整数部分对应相等，$K_{-1}=1$，小数部分也对应相等：

$$0.6250 = K_{-2} \times 2^{-1} + K_{-2} \times 2^{-1} \cdots + K_{-m} \times 2^{-m+1}$$

等式两边再同乘以 2 得到：

$$1.2500 = K_{-2}\times 2^{0}+K_{-3}\times 2^{-1}\cdots+K_{-m}\times 2^{-m+2}$$

等式两边整数部分对应相等，$K_{-2}=1$，…，继续把上式两边小数乘以 2，直至十进制小数部分为 0 停止。

根据上例，得出转换方法如下：

把十进制小数部分连续乘以 2，依次取得整数，直到乘积小数部分为 0 停止，依次得出乘积的整数序列即是二进制小数从高位到低位各数位上的系数。十进制小数转换成二进制小数的方法称为乘 2 取整法。

上例用竖式表示如下：

十进制小数 ×2	二进制小数的数位系数 = 十进制整数部分
$0.8125\times 2 = 1.625$	$K_{-1} = 1$
$0.625\times 2 = 1.25$	$K_{-2} = 1$
$0.25\times 2 = 0.5$	$K_{-3} = 0$
$0.5\times 2 = 1.0$	$K_{-4} = 1$

转换结果是：$0.8125 = 0.1101B$。

上例用线图表示如下：

$$\begin{array}{lccccc} & & \times 2 & & & \\ 0.8125 & \rightarrow & 0.625 & \rightarrow & 0.25 \rightarrow & 0.5 \rightarrow 0 \\ & & \downarrow & & \downarrow \quad \downarrow & \downarrow \\ \text{积的整数部分} & & 1 & & 0 \quad 1 & \end{array}$$

3. 二进制数转换成十六进制数

二进制数转换成十六进制数的转换方法是：从小数点开始，整数部分向左，小数部分向右，每四位二进制数为一组用一位十六进制数表示，不足四位的用 0 补足。

【例 1.5】 把二进制数 11110110101.10101 转换成十六进制数。

二进制数　（0）111　1011　0101. 1010　1（000）

十六进制数　7　B　5.　A　8

转换结果是：11110110101.10101B = 7B5.A8H。

4. 十六进制数转换成二进制数

十六进制数转换成二进制数的转换方法是：每一位十六进制数用相应的四位二进制数代替，多余的 0 舍去。

【例 1.6】 把十六进制数 9F.8 转换成二进制数。

十六进制数　9　F　.　8

二进制数　1001　1111 .　1000

舍去

转换结果是：9F.8H = 10011111.1B

1.1.3 进位计数制的计量单位

二进制信息的基本单位是位（bit），由 8位二进制信息组成一个字节（Byte）。表示位和字节的英文符号分别为 b和 B。

因为 $10^3 = 1000$，$2^{10} = 1024$，所以 3 个十进制位近似于 10 个二进制位。

在国际单位制中，十进制是以 3 个十进位分挡的，即：

千（kilo）$= 10^3 = 1K = 1000$；
兆（mega）$= 10^6 = 1M = 10^3K = 1000K$；
吉（giga）$= 10^9 = 1G = 10^3M = 1000M$；
太（tera）$= 10^{12} = 1T = 10^3G = 1000G$。

在国际单位制中，二进制是以 10个二进位分挡的，即：

千（kilo）$= 2^{10} = 1K = 1024$；
兆（mega）$= 2^{20} = 1M = 2^{10}K = 1024K$；
吉（giga）$= 2^{30} = 1G = 2^{10}M = 1024M$；
太（tera）$= 2^{40} = 1T = 2^{10}G = 1024G$。

1.2 计算机中数的表示

1.2.1 机器数和真值

数在计算机中的表示形式称为机器数，而把这个数的本身称为真值。机器数有以下两个基本特点。

1. 数的符号数值化

数分为正数和负数，分别用正号“＋”和负号“－”表示。在计算机中，数的符号只能用 0和 1 表示，以 0表示正号，以 1表示负号，这样有符号数的符号就被数值化了。在计算机中通常把符号放在最高位，该位称为符号位。一个机器数是由符号位和数值位两部分组成的。例如，真值是＋1001B，对应的机器数为 01001B；真值是－1001B，对应的机器数为 11001B。

2. 数的位数固定

计算机内一次能表示二进制数的位数叫做计算机的字长，一台计算机的字长是固定的。字长为 8 位叫做一个字节，计算机字长一般都是字节的整数倍，如字长 8 位、16 位、32 位、64 位及 128 位。

1.2.2 机器数的表示方法

常用的机器数表示方法有 4 种：原码、反码、补码和移码。

1．原码表示法

原码表示法为：正数的符号位为 0，负数的符号位为 1，数值位是真值的绝对值。即：

$X=+X_1X_2\cdots X_n$，$[X]_{原}=0X_1X_2\cdots X_n$；

$X=-X_1X_2\cdots X_n$，$[X]_{原}=1X_1X_2\cdots X_n$。

【例 1.7】 写出真值 $X_1=+1001010$，$X_2=-1001010$ 的原码。

$[X_1]_{原}=01001010$，$[X_2]_{原}=11001010$。

【例 1.8】 写出 8 位原码表示的最大和最小整数。

$Max[X]_{原}=[01111111]_{原}=+1111111B=+127$；

$Min[X]_{原}=[11111111]_{原}=-1111111B=-127$。

用 8 位原码表示整数的范围是＋127～−127。

采用原码表示的数很直观，而且乘除法可直接进行，但用原码进行加减法运算的运算规则则比较复杂。

2．反码表示法

反码表示法为：正数的符号位为 0，数值位取真值；负数的符号位为 1，数值位取真值的相反码。即：

$X=+X_1X_2\cdots X_n$，$[X]_{反}=0X_1X_2\cdots X_n$；

$X=-X_1X_2\cdots X_n$，$[X]_{反}=1X_1X_2\cdots X_n$。

【例 1.9】 写出真值 $X_1=+1100111$，$X_2=-1100111$ 的反码。

$[X_1]_{反}=01100111$，$[X_2]_{反}=10011000$。

【例 1.10】 写出 8 位反码表示的最大和最小整数。

$Max[X]_{反}=[01111111]_{反}=+1111111B=+127$；

$Min[X]_{反}=[10000000]_{反}=-1111111B=-127$。

用 8 位反码表示整数的范围是＋127～−127。

3．补码表示法

（1）补码的概念

如果使用两位计算器，计算器超出两位数的范围则自动丢失，79−38 和 79＋62 的运算结果相同。在计算三角函数值时，$\alpha-30°$ 和 $\alpha+330°$ 的三角函数值相同。在数学上用同余式表示：

$79-38=79+(100-38)(\mathrm{mod}100)$

$\alpha-30°=\alpha+(360-30°)(\mathrm{mod}360°)$

这里的 100 和 360 在数学上叫做模，用 mod 或 M表示。模是指一个计量系统的测量范围，其大小以计量进位制的基数为底，数的位数为指数的幂。计算机是一种有限长的数字系统，它的模是 2^n，n 是计算机的字长。8 位字长计算机的模是 $2^8=256$。在模一定的条件下，负数总可以用一个正数等价，这个负数的等价量称为该数的补码。当模为 100 时，−38

的补码是 62；模 为 360 时，−30 的补码为 330；模 为 256 时，−1 的补码为 255。

（2）补码表示法

用补码表示计算机中的有符号数，正数的符号位为 0，数值位取真值；负数的符号位为 1，数值位取真值的相反码加 1。即：

当 $X=+X_1X_2\cdots X_n$ 时，$[X]_{补}=0X_1X_2\cdots X_n$；

当 $X=-X_1X_2\cdots X_n$ 时，$[X]_{补}=1\ (\ \overline{X_1}\ \overline{X_2}\cdots\overline{X_n}+1)$。

【例 1.11】 写出真值 $X_1=+1001110$，$X_2=-1001110$ 的补码。

$[X_1]_{补}=01001110$　　$[X_2]_{补}=10110010$

【例 1.12】 写出 8 位补码表示的最大和最小整数。

$Max[X]_{补}=[01111111]_{补}=+1111111B=+127$

$Min[X]_{补}=[10000000]_{补}=-10000000B=-128$

8 位补码表示整数的范围是 +127～−128。

【例 1.13】 写出 16 位补码表示的最大和最小整数。

$Max[X]_{补}=[0111111111111111]_{补}=+111111111111111B=+32767$；

$Min[X]_{补}=[1000000000000000]_{补}=-1000000000000000B=-32768$。

用 16 位补码表示整数范围是 +32767～−32768。

用补码表示法能使减法运算转化为加法运算，并且在进行加减运算时，能使符号位和数值位一起运算，从而简化运算规则。

4．移码表示法

移码也称作增码，就是在补码的基础上增加一个偏移量。根据多数高级程序语言软件包的实数标准格式，字长为 8 位的移码，其偏移量为 127（7FH）；字长为 11 位的移码，其偏移量为 1023（3FFH）。

【例 1.14】 写出 $X_1=+0000011B$，$X_2=-0000011B$ 的移码。

$[X_1]_{移}=[X_1]_{补}+偏移量=[00000011B]_{补}+01111111B=[10000010B]_{移}$；

$[X_2]_{移}=[X_2]_{补}+偏移量=[11111101B]_{补}+01111111B=[01111100B]_{移}$。

1.2.3 数的定点和浮点表示

任意一个二进制数都可以表示为纯整数或纯小数与一个 2 的整数次幂的乘积。即：

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

其中：S 称为数 N 的尾数，是数值的有效数字；E 称为数 N 的阶码（指数），指明小数点的位置；2 称为阶码的底。如：

$1011101B=2^{+6}\times 1.011101B$；

$101.1101B=2^{+2}\times 1.011101B$；

$0.01011101B=2^{-2}\times 1.011101B$。

1. 定点数表示法

当阶码为常数时，这种数的表示方法称为定点数表示法。定点数表示法的小数点位置有以下两种约定：

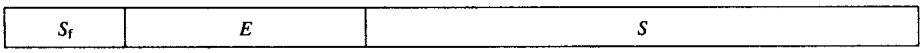
- 所有机器数的小数点位置隐含在数的最低位之后，把所有的数化为纯整数，这称为定点整数。
- 所有机器数的小数点位置隐含在符号位之后，把所有的数化为纯小数，这称为定点小数。

在计算机采用定点数表示时，对于既有整数又有小数的原始数据，需要将数据按比例因子缩小成定点小数或扩大成定点整数，参加运算后的结果，再按比例因子计算出实际值。定点数表示方法简单直观，但表示数的范围较小。

2. 浮点数表示法

当阶码取不同的数值时，这种数的表示方法称为浮点数表示法。

电气和电子工程师协会 IEEE-754 标准规定了浮点数的格式，在许多高级语言中被采用。浮点数在计算机中的表示形式如下：



其中：E 是阶码，常用移码表示：S_f 是尾数的符号位；S 是尾数，一般采用原码表示。浮点数表示法也有以下两种形式：

- 单精度浮点数（Single）：字长为 32 位实数，由 1 位符号、8 位阶码和 23 位尾数组成，以 4 个字节形式存储。
- 双精度浮点数（Double）：字长为 64 位实数，由 1 位符号、11 位阶码和 52 位尾数组成，以 8 个字节形式存储。

1.3 计算机中的编码

数字、字母、符号、汉字等信息统称为数据。能够进行算术运算得到明确数值概念的信息称为计算机数值数据，其余的信息称为非数值数据。送入计算机的数据，必须转换成二进制数据才能被计算机接受、存储及进行算术运算或逻辑运算，因此必须对数据进行编码。

1.3.1 数字编码

计算机的输入输出数据是十进制数，而计算机内部运算是用二进制数，因此十进制数必须用二进制数形式表达。用四位二进制数表示一位十进制数的编码，称为二进制编码的十进制数，简称 BCD 码。用四位二进制数表示一位十进制数的编码种类繁多，但最常用的是 8421BCD 码。

8421BCD 码的 4 个二进制位自左向右每位的权分别是 8, 4, 2, 1，用二进制数 0000～1001 十个编码分别表示十进制数的 0～9，而 1010, 1011, 1100, 1101, 1110, 1111 六个编码是 8421BCD 码的非法编码。8421BCD 码和十进制数的对应关系如表 1.2 所示。

表 1.2　8421BCD 编码表

0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
非法编码	1 0 1 0
	1 0 1 1
	1 1 0 0
	1 1 0 1
	1 1 1 0
	1 1 1 1

在一个字节内存放两位 BCD 数称为压缩的 BCD 数，如压缩的 BCD 数 10011001B 表示十进制数 99。在一个字节内只存放一位 BCD 数称为非压缩的 BCD 数，高半个字节为 0，低半个字节为 BCD 数，如非压缩的 BCD 数 00001001B 表示十进制数 9。在计算机中的扩展精度 BCD 数占 10 个字节（80 位），第 1～9 个字节是压缩的 BCD 数，也就是 18 位 BCD 数，第 10 个字节是符号位。

1.3.2　校验码

由于计算机结构、工艺及电气性能等方面因素的影响，常常使数据在存取、传输的过程中出现错误，为了及时发现并修正错误，计算机采用校验码对传输的数据进行校验和修正。校验码的种类很多，最普遍使用的检验码是奇偶校验码。

奇偶校验码是将每个信息的代码，扩展一个二进制位作为校验位。校验位的取值原则是：若是奇校验，在编码中含有“ 1 ”的个数连同校验位的取值共有奇数个；若是偶校验，在编码中含有“ 1 ”的个数连同校验位的取值共有偶数个。例如，是奇校验，信息编码是 10001000B，在信息中有两个“ 1”，所以校验位取值为“ 1”，使“ 1 ”的总数有奇数个，该信息的奇校验码为 110001000B。

在计算机中进行代码检验时，当检测信息编码中“ 1 ”的个数与预先设定的奇／偶校验不符时，表明被检测的信息有误。奇偶校验方法是有局限性的，只能校验奇数个二进制位的错误，但一位出错比两个或多位出错的概率高，所以它仍不失是一种有效的校验方法。

1.3.3 字符编码

数字、字母、通用符号和控制符号等统称为字符，用来表示字符的二进制代码称为字符编码。字符编码有多种方式，国际上普遍采用的一种字符编码是美国国家信息交换代码（American Standard Code for Information Interchange），简称 ASCII 码，其字符集如表 1.3 所示。我国标准局颁发的 GB1988—80 信息交换七位编码字符集和 ASCII 码基本相同（只是货币符号的美元符号“$”改为人民币符号“￥”）。ASCII 码选择了 4 类共 128 个常用字符，说明如下：

- 数字 0～9：ASCII 码的 10 个数字字符和 BCD 码是两个不同概念。
- 字母：包括 26 个大写英文字母和小写英文字母。
- 通用符号：如 * . . . % . = 等。
- 控制符号：如 ESC、CR 等。

表 1.3 美国国家信息交换代码字符集

$b_3 b_2 b_1 b_0$ \ $b_6 b_5 b_4$		0	1	2	3	4	5	6	7
		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	S P	0	@	P	`	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	“	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	’	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	|
D	1101	CR	GS	—	=	M	]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	US	/	?	O	_	o	DEL

ASCII 码特殊控制功能符的解释如下

MUL	空	（空白）	DC1	设备控制 1	（机控 1）
SOH	标题开始	（序始）	DC2	设备控制 2	（机控 2）
STX	正常结束	（文始）	DC3	设备控制 3	（机控 3）
ETX	本文结束	（文终）	DC4	设备控制 4	（机控 4）

EOT	传输结束	（送毕）	NAK	否定	（否认）
ENQ	询问	（询问）	SYN	空转同步	（同步）
ACK	承认	（承认）	ETB	信息组传送结束	（组终）
BEL	报警符	（告警）	CAN	作废	（作废）
BS	退一格	（退格）	EM	纸尽	（载终）
HT	横向列表	（横表）	SUB	减	（取代）
LF	换行	（换行）	ESC	换码	（扩展）
VT	垂直列表	（纵表）	FS	文字分隔符	（卷隙）
FF	走纸控制	（换页）	GS	组分隔符	（群隙）
CR	回车	（回车）	RS	记录分隔符	（录隙）
SO	移位输出	（移出）	US	单元分隔符	（元隙）
SI	移位输入	（移入）	SP	空格	（间隔）
DLE	数据链换码	（转义）	DEL	作废	（抹掉）

注：（）内为我国国家信息代码对编码的解释；非（）的是美国国家信息代码的注释。

ASCII 码字符表是以二维表形式列出的，共 8 列 16 行。前两列（0 列、1 列）是控制字符，共 32 个。以后 6 列（2 列至 7 列）是图形字符，包括数字、字母和通用符号，共 96 个，除“空格”字符和“抹掉”字符外，其余 94 个都是可见图形字符。字符行是 3～0 位的 4 位二进制编码，字符列是 6～4 位的 3 位二进制编码，由行列编码（列在前，行在后）组成 7 位二进制编码。例如，字母“E”所在位置是 5 行 4 列，行编码为 0101，列编码为 100，字母“E”的 ASCII 码为 1000101B = 45H；数字“3”所在位置是 3 行 3 列，其 ASCII 码为 0110011B = 33H。因为计算机的字节长是 8 位，存放一个 ASCII 码后多余 1 位，其多余的最高位可以是零或是奇偶校验码。

1.3.4 汉字编码

汉字的历史源远流长，世界约四分之一的人口使用汉字，汉语被联合国列为法定的 6 种正式语言和工作语言之一。近年来我国为了推广计算机的应用和加强国际交流，颁发了一系列相关的中文信息处理标准。

1981 年，我国颁发了 GB2312—80 国家标准信息交换用汉字编码字符集——基本集，规定常用汉字总数为 6763 个，其中一级汉字有 3755 个，是常用汉字；二级汉字有 3008 个，是不常用汉字。除汉字外，有一般符号 202 个（包括间隔符号、标点符号、运算符号、单位符号、制表符号），序号 60 个，数字 22 个，拉丁字母 52 个，日语假名 169 个，希腊字母 48 个，俄文字母 66 个，汉语拼音符号和注音符号 63 个。汉字编码表是以二维表形式列出的（见附录 D），汉字编码表有 94 行 94 列，行号称为区号，列号称为位号。一般符号占 1 区到 10 区，一级汉字占 16 区到 55 区，二级汉字占 56 区到 87 区。汉字的编排顺序是：一级汉字以汉语拼音顺序排列；二级汉字以其偏旁部首顺序排列。

近年，我国颁发了“汉字内码扩展规范（GBK）”，该规范兼容 GB2312—80 国家标准汉字编码字符集，并包括了我国台湾地区使用的计算机汉字编码字符集（13070 个繁体字），是支持全世界各种文字所使用字符的编码标准（ISO 10646）。

1．区位码

汉字的区位编码是把汉字所在位置的区号和位号合起来（区号在前，位号在后）得到的四位数字的编码。例如，“啊”字的区位编码是 1601，“波”字的区位编码是 1808。

2．国标码

汉字的国标码是用二进制表示汉字的编码，行和列都是 7 位二进制编码。每个汉字的国标码是由两个字节的代码组成的，第一个字节的最高位是 0，低 7 位是行二进制编码；第二个字节的最高位也是 0，低 7 位是列二进制编码。汉字的区位码和国标码的转换规律如下：

国标码高字节 = 区码+20H

国标码低字节 = 位码+20H

例如，“啊”字的国标码为 3021H，“波”字的国标码为 3228H。

3．内码

汉字在计算机内部存储、运算的信息代码称为汉字的内码。内码也是由两个字节组成的，两个字节的最高位均为 1，两个字节的低 7 位为国标码。将汉字国标码的两个字节最高位均加 1，即是该汉字的内码。汉字的区位码、国标码和内码的转换规律如下：

内码高字节 = 国标码高字节+80H

内码低字节 = 国标码低字节+80H

例如；“啊”字的内码为 B0A1H，“波”字的内码为 B2A8H。

为了便于汉字和西文信息处理有较好的兼容性，每个字节的最高位作为区分 ASCII 码和汉字内部码的标识位。标识位为“0”，则是 ASCII 码，又称为半角字符编码；标识位为“1”，则是汉字内码，采用汉字内码的字符编码称为全角字符编码。

无论是何种汉字输入方法，汉字的内码都是相同的。

习题 1

1．选择题

（1）十进制数 36 等值的二进制数是（）。

A. 00100100　　B. 00011110

C. 00110110　　D. 00110100

（2）十进制数 0.6875 转换为二进制数是（）。

A. 0.1101　　B. 0.0111

C. 0.1011　　D. 0.1100

（3）十进制数 36.875 转换为二进制数是（）。

A. 110100.011　　B. 100100.111

C. 100110.111　　D. 100101.101

(4) 二进制数 111010011 转换为十六进制数是（ ）。

A. 323H　　B. 1D3H

C. 133H　　D. 3D1H

(5) 二进制数 111010.11 转换为十六进制数是（ ）。

A. 3ACH　　B. 72.6H

C. 3A3H　　D. 3A.6H

(6) 原码 +127 的反码是（）。

A. +127　　B. −127

C. +0　　D. −0

(7) 十进制数−38 的 8 位二进制补码是（）。

A. 01011011　　B. 11011010

C. 11011011　　D. 10101010

(8) 8 位二进制补码是 10110101，其相应的十进制数是（）。

A. −54　　B. −68

C. −74　　D. −48

(9) 十进制数−35 的二进制码 11011101，机器数的表示法是（）。

A. 原码　　B 反码

C 补码　　D 移码

(10) 用二进制数表示 0～99999 十进制数，最少需要（）位。

A. 15 位　　B. 16 位

C. 17 位　　D. 18 位

(11) 下列数据中（）最小。

A. 11011001B　　B. 256

C. 0A7H　　D. 217

(12) 下列数据中（）最大。

A. 11110110B　　B. 245

C. 0F5H　　D. 240

(13) 用 8位二进制的补码形式表示带符号数的范围是（）。

A. −128～+128　　B. −127～+127

C. −127～+128　　D. −128～+127

(14) ASCII码 00110111 所表示的十进制数是（）。

A. 7　　B. 37

C. 55　　D. 67

(15) 在 GB2312—80 国家标准中，汉字的机内码 DBB5为（）。

A. 图形字符　　B. 一级汉字

C. 二级汉字　　D. 非法码

2. 完成下列不同进制数的转换

(1) 将下列十进制数转换成二进制数。

① 270；　　② 455；

③ 0.8125；　　④ 720.3125。

(2) 将下列二进制数转换成十进制数。

① 1001001；　　② 11001100；

③ 0.0101；　　④ 11011.1011。

(3) 将下列二进制数转换成十六进制数。

① 11100011；　　② 10001111；

③ 0.0011101；　　④ 110011011.01011。

(4) 将下列十六进制数转换成二进制数。

① A21；　　② 4B7；

③ 0.00A3；　　④ 2E8.0D5。

3. 完成下列机器数和真值的转换

(1) 用 8位二进制补码表示下列各数。

① 11001B；　　② −11001B；

③ 100000B；　　④ −100000B。

(2) 将下列各十进制数用 8位二进制补码表示。

① 65；　　② −75；

③ 120；　　④ −100。

(3) 将下列各十进制数用 16位二进制补码表示。

① 1000；　　② −12；

③ 800；　　④ −3212。

(4) 将下列用补码表示的二进制数转换为十进制数。

① 10000001B；　　② 01100110B；

③ 0111011101110111B；　　④ 1000000000000001B。

4. 完成下列各数值和计算机编码的转换

(1) 将下列二进制数表示为压缩的和非压缩的 BCD 数。

① 01100011B；　　② 01010000B；

③ 0000001100001111B；　　④ 0001111111111111B。

(2) 用十六进制数表示下列编码。

①换行；　　② 字母“Q”；

③ ASCII 码“7”；　　④ 空格；

⑤汉字“险”（国标码）；　　⑥ 汉字“保”(内码)。

(3) 一个十进制数 15，请分别用以下编码表示。

① 用 8位二进制补码表示；　　② 用压缩的 BCD 码表示；

③用非压缩的 BCD 码表示；　　④ 用 ASCII 码表示；

⑤用区位码表示；　　⑥ 用国标码表示；

⑦ 用内码表示；　　⑧ 用奇校验码表示。

5. 分析解答以下各题

(1) $(111)x = 273$，基数 $x = ?$

(2) 有两个二进制数，$x = 01101010$，$y = 10001100$，试比较它们的大小。

① x 和 y 两个数均为无符号数；

② x 和 y 两个数均为有符号的补码数。

(3) 一个用十六进制表示的两位整数， 如果改用十进制数表示，顺序正好颠倒，该数是多少？

第2章 微型计算机的基本结构

2.1 微型计算机系统

微型计算机系统是由硬件和软件两部分组成的。硬件是指实际的物理设备，包括计算机的主机及其外部设备；软件是指程序及其文档。在微型计算机系统中，硬件是物质基础，软件则能发挥计算机的效能，两者是相辅相成的，构成了统一体。

2.1.1 微型计算机的硬件

冯·诺伊曼（Von Neumann）“存储程序控制式”计算机系统结构是由运算器、控制器、存储器、输入设备和输出设备五部分组成的。运算器和控制器合称为中央处理单元（Central Processing Unit），简称 CPU。把中央处理单元的功能集成在一个芯片上的大规模或超大规模集成电路，称做微处理器，简称 MPU。

微型计算机由微处理器、存储器、输入／输出接口和总线组成，就体系结构而言，称为总线结构。微型计算机的硬件系统结构如图 2.1 所示。

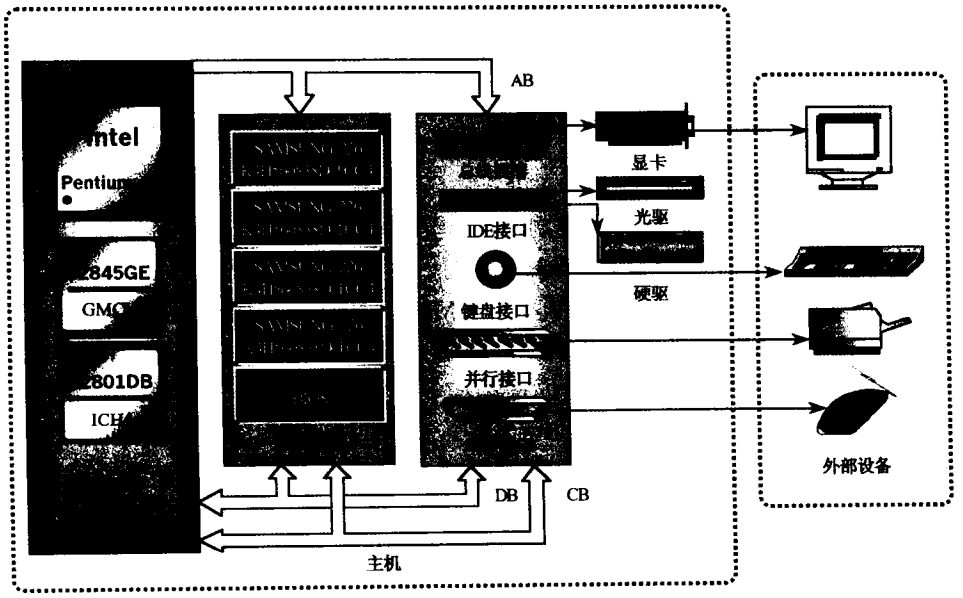


图 2.1 微型计算机系统结构