中学教师进修教材

计算机基础

上

陕西教育学院数学系编

1985.2

前　言

作为新技术革命主力的电子计算机正在走向社会的各个角落，因而计算机教育也正在由专业教育变成普通教育。为了贯彻教育部1984年初颁布的“中学BASIC语言教学大纲”，急需为中学培训一批计算机课程的教师，同时考虑到广大中学教师进修的需要，我们编写了这本讲义。

讲义分为两部分，前三章为第一部分，主要介绍数的进位制，逻辑代数的应用和电子计算机硬件简介；后七章为第二部分，重点介绍基本BASIC语言程序设计的基本方法和技巧。第一部分约需18学时，第二部分约需54学时（包括上机实习）。

由于编者水平有限，本讲义在选材的广度、深度及内容诸方面，肯定存在不当与错误之处，诚请读者批评指正。

编　者

1984年7月

上册目录

第一章　计算机中数的表示

计算机中数的表示采取什么形式，将直接影响计算机的性能和结构。在这一章，重点介绍使用计算机所涉及的几种不同进位制及其相互转换，同时提出数的定点和浮点表示，最后给出了数在计算机中的原码、补码和反码三种表示方法。

§1　进位计数制

我们最常用最熟悉的十进制数是由0，1，2，……9十个记数符号（又称数码），根据“逢十进一”，“退一当十”的进退位方式排列成的有规则的序列，其中每个数码按其所在位置对应一个10的若干次幂为位值（或权）。例如，十进制数1984.91可表为

$$1984.91$$
$$= 1\times10^{3}+9\times10^{2}+8\times10^{1}+4\times10^{0}+9\times10^{-1}+1\times10^{-2}$$

在一种进位制中，全部可使用的数码总数称为它的基数（或底数），十进制的基数为10。

如果只用两个数码“0”和“1”，用“逢二进一，退一当二”的进退位原则，就得到二进制的记数法。二进制的每个数码的位值为2的某次幂，而基数为2。例如，二进制数101.01可表为

$$101\cdot01$$

$$=1\times2^{2}+0\times2^{1}+1\times2^{0}+0\times2^{-1}+1\times2^{-2}$$

一般地，若R≥2的整数，设R进制数S记作$(S)_R$，则$(S)_R$可表示为：

$$(S)_R=(a_n a_{n-1}\cdots a_1 a_0 a_{-1} a_{-2}\cdots a_{-(m-1)} a_{-m})_R$$

$$=a_nR^n+a_{n-1}R^{n-1}+\cdots+a_1R^1+a_0R^0+a_{-1}R^{-1}$$

$$=a_{-2}R^{-2}+\cdots+a_{-m}R^{-m}$$

$$=\sum_{i=n}^{-m}a_iR^i$$

公式$(S)_R=\sum_{i=n}^{-m}a_iR^i$叫做R进制数S的按权展开式。

由此式可将R进制数S的特点归结如下：

(1) 只有R个数码，每个a_i只能在0，1，2，…R－1这R个数码中间取值；

(2) 每一位数码a_i所表示的数值是a_iR^i，R^i即数码a_i所在数位的位权或位值（简称权），a_i是R^i的系数，a_i所在数位称为R的i次幂位。

(3) “逢R进一，退一当R”，相邻两位中高位的一个单位等于低位的R个单位。

当基数R＞10时要加选数码。如R＝12，要加0，1（或t，e）表示10和11两个数码。

例1，在什么进位制中，16324是125的平方？

解：设在K进制中，16324是125的平方

那么　　$(16324)_K=(125)_K^2$

即　　$K^4+6K^3+3K^2+2K+4=(K^2+2K+5)^2$

亦即　　$2K^2-11K^2-18K-21=0$

上式加上并减去$3K^2+3K$，得

$$(K-7)\quad(2K^2+3K+3)=0$$

由此　　$K=7$

例2，证明：在基数是a的进位制中，基数的前面一个数的两倍和基数的前面一个数的平方用同样的数字写出，但取相反的顺序。

证：∵基数是a

∴基数前面一个数是$a-1$

从而$2(a-1)=2a-2=a+(a-2)=(1\quad a-2)_a$

$(a-1)^2=a^2-2a+1=(a-2)a+1=(a-2\quad 1)_a$

故结论正确。

例3　在某些进位制中，任何数能被已知数d整除的特征是其数字和能被d整除。求出所有这些进位制。

解：设$N=(a_na_{n-1}\cdots a_1a_0)_R$，

其中R是进位制的基数，a_i（$i=n,n-1,\cdots,1,0$）是数的各位数字。

依题意有

$$N-(a_n+a_{n-1}+\cdots+a_1+a_0)=m\cdot d$$

如果　$N=R$，那么 $a_n+a_{n-1}+\cdots+a_1+a_0=1$

从而有　　$R-1=m\ d$

由此　　$R=m\ d+1$

§2　二进制与电子计算机

在计算机中究竟采用何种进位制，取决于该进位制在机器设计制造上是否容易实现，是否计算简便，是否节省设备。

一、三进制最省设备

对于R进制，每一位数需要R种不同状态的元件来表示，则表示m位R进制数所需不同物理状态的个数为

$$X=m\cdot R \qquad (1)$$

设m位R进制所能表示的最大数是N，显然有

$$N+1=R^m \qquad (2)$$

我们的问题是，R取何值时，X取最小值。

由(1)　　$$m=\frac{X}{R} \qquad (3)$$

(2) 两边取对数，得 $L_n(1+N)=mL_nR$　　(4)

(3) 代入(4)得　　$$L_n(1+N)=\frac{X}{R}L_nR$$

从而 $$X=\frac{R\ \ L_n(1+N)}{L_n\ R}$$

令 $\frac{dx}{dR}=0$，即有

$$\frac{L_n(N+1)}{L_n{}^2R}\ (L_nR-1)=0$$

$\because R\geq 2$ 且 $N+1\neq 1$，$\therefore \frac{L_n(N+1)}{L_n{}^2R}\neq 0$

因此 $L_nR-1=0$

所以，当 $R=e$ 时，X 取最小值。

实际上R应为整数，而与e靠近的整数有2和3，且因为

$$X(2)-X(3)=\frac{2L_n(N+1)}{L_n2}-\frac{3L_n(N+1)}{L_n3}$$

$$=L_n(N+1)\frac{L_n9-L_n8}{L_n2\cdot L_n3}>0$$

故三进制最省设备。

二、二进制的特点

目前，大多数计算机都采用二进制，这是因为：

第一，易于实现，制造具有两个稳定物理状态的元件，比制造

具有三个或十个稳定物理状态的元件要简单得多。由于采用二进制，在计算机中，数的存贮和传送，也可以简单而可靠的方式进行，如电压的高低，脉冲的有无等。

第二，运算简单。一般地说，进行R制的四则运算需要记住 $\frac{R(R+1)}{2}$ 个和与积。当R＝2时，$\frac{R(R+1)}{2}=3$。也就是说二进制的和与积数表很简单，容易记忆

加法表：

0＋0＝0

1＋0＝0＋1＝1

1＋1＝10

乘积表：

0×0＝0

1×0＝0×1＝0

1×1＝1

下面举几个例子，看看二进制正整数的运算规则：

$$\begin{array}{r} 111 \\ +\ 011 \\ \hline 1010 \end{array} \qquad \begin{array}{r} 1101 \\ -\ 110 \\ \hline 111 \end{array}$$

```
    10111
  ×  1010
  -------
    00000
   10111
  00000
 10111
 -------
 11100110
```

```
          11001 (商)
        ----------
111 ) 10110001
      111
      ---
       1000
        111
       ----
          1001
           111
          ----
            10 (余数)
```

计算机采用二进制，除了可以进行算术运算之外，还能进行多种逻辑运算。

第三，节省设备。前面我们证明了基数为3时最省设备，但是，目前制造具有三种稳定状态的元件不但难度大，而且可靠性差。

第四，采用二进制，就可以使用逻辑代数。这就为计算机的逻辑设计提供了便利的工具。

二进制的缺点是，表示较大的数时，读、写不方便；数据在计算机中输入输出，需要对数进行数制转换，占用不少的机器时间。

所以根据不同的需要，有些专用数据处理机采用十进制；编制手编程序时常采用八或十六进制。

§3　不同进位制数间的转换方法

十进制中位数不太多的数，写成二进制数时，是一个位数相当多的数。例如

$$1024=2^{10}=10000000000$$

十个零

因此，在为电子计算机编程序时，用二进制是很不方便的，为此，人们又常常采用八进制、十六进制。那样，就需要各种进位制之间的互相转换。

不同进位制之间的转换是根据如果两个有理数相等，则两数的整数部分和分数部分一定分别相等的原理进行的。

一、十进制与二进制数之间的转换

1、二进制数转换成十进制数

方法1　按权展开求和法，直接利用二进制数的按权展开式，在十进制中计算，便可将任何一个二进制数转换成十进制数。

例1　(1)　$(10110)_{二}=1\times2^4+1\times2^2+1\times2^1=22$

(2)　$(0\cdot1101)_{二}=2^{-1}+2^{-2}+2^{-4}=0\cdot8125$

(3)　$(111111\cdot011001)_{二}$

$$= 2^5+2^4+2^3+2^2+2^1+1+2^{-2}+2^{-3}+2^{-6} = 63\frac{25}{64}$$

方法2　2乘、除加整法。为了避免位数较多时求和的麻烦，可采用2乘、除加整法。

例2　求(1)　$(1011011)_{二}=(?)_{十}$

(2)　$(0.1001)_{二}=(?)_{十}$

因为　$(1011011)_{二}=2^5+0\times2^4+2^3+0\times2^2+2+1$

$$=2\{2^4+0\times2^3+2^2+0\times2^1+1\}+1$$

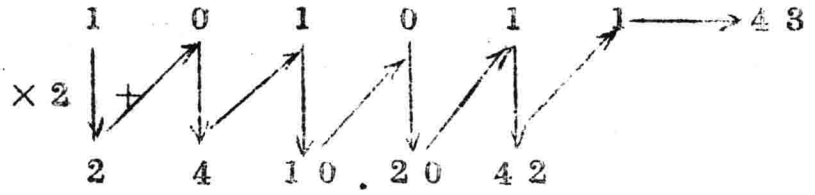
$$=2\{2(2^3+0\times2^2+2^1+0\times1)+1\}+1$$

$$=2\{2(2(2^2+0\times2^1+1)+0)+1\}+1$$

$$=2\{2(2(2\overline{2+0}+1)+0)+1\}+1$$

$$=2\{2(2(2\overline{2\times1+0}+1)+0)+1\}+1$$

从最里层的括号向外看，恰好是用2去乘原二进数$(1011011)_{二}$最高位的1加上其后一位的0，再乘2上再下一位的1，再乘2加上…，这个运算过程可以简写成：

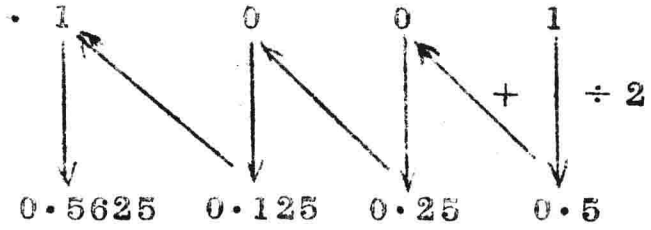


所以　$(1011011)_{二}=43$

又因为　$(0\cdot1001)_{二}=2^{-1}+0\times2^{-1}+0\times2^{-3}+2^{-4}$

$=2^{-1}\times1+2^{-1}(0+2^{-1}(0+1\times2^{-1}))$

且$1\times2^{-2}=1\div2$　故可得简写式



所以　$(0\cdot1001)_{二}=0\cdot5625$

应用这种方法可将整数部分和小数部分分别转换成十进制数，再将这两个十进制数加起来，就是所求得的结果。

方法3　二～八～十转换方法。二进制数位数较多时，还可先将其转换成八进制数，再把八进制数转换成十进制数。

例3　$(11101110\cdot0111101)_{二}=(?)_{十}$

$(011,101,110\cdot011,110,100)_{二}$

$=(356\cdot364)_{八}$

$=3\times8^{2}+5\times8+6+3\times8^{-1}+6\times8^{-2}+4\times8^{-3}$

$=238\frac{244}{512}=238\frac{61}{128}$

2．十进制数转换成二进制数

(1)　整数

关键在于将十进制整数写成2的非负整数次幂的和，例如：

$$(57)_{十}=32+16+8+1$$

$$=1\times2^5+1\times2^4+1\times2^3+0\times2^2+0\times2^1+1\times2^0$$

$$=(111001)_{二}$$

或 $$=5\times10+7=(2^2+2^0)(2^3+2^1)+(2^2+2+2^0)$$

$$=\cdots\cdots$$

再看下面的一系列除式：

$$57=2\times28+1$$

$$28=2\times14+0$$

$$14=2\times7+0$$

$$7=2\times3+1$$

$$3=2\times1+1$$

$$1=2\times0+1$$

这些除式的特点是：除数都是2；前一个除式的商是后一个除式的被除数；余数是0或1。

对这些除式从下往上依次读余数，恰好也是要求的二进制数 $(111001)_{二}$。这不是偶然的巧合，我们以四位的二进制整数为例说明如下：

假设有一个四位二进制整数 $a_3a_2a_1a_0$，不知道各个数位上的数码，但知道这个数的十进制表示是M。则

$$M=a_3\times2^3+a_2\times2^2+a_1\times2^1+a_0$$

从已知的M及2开始，作一系列除法：

$$M=2\times(a_3\times2^2+a_2\times2^1+a_1)+a_0$$

$$(a_3\times2^2+a_2\times2+a_1)=2\times(a_3\times2+a_3)+a_1$$

$$(a_3\times2+a_2)=2\times(a_3)+a_2$$

$$(a_3)=2\times0+a_3$$

把在已知十进制数M和2的条件下，依次求出a_0，a_1，a_2，a_3的这种方法，叫做2除取余法。

下面举例说明常用的“2除取余法”的计算格式：

2	37	
2	18……1（余数1就是2^0的系数）	
2	9……0（" 0 "2^1" ）	书写顺序
2	4……1（" 1 "2^2" ）	
2	2……0（" 0 "2^3" ）	
2	1……0（" 0 "2^4" ）	
	0……1（" 1 "2^5" ）	

因此，$(37)_十=(100101)_二$。解题时，括号内的说明可以不写，箭头方向表示从下向上读余数。

(2) 纯小数

关键在于将十进制纯小数写成2的负整数次幂的和，例如，

$$(0\cdot625)_十=0\cdot5+0\cdot125$$

$$=1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$$

$$=(0.101)_{二}。$$

再看下面的一系列乘法等式：

$$2\times0.625=1+0.25;$$

$$2\times0.25=0+0.5$$

$$2\times0.5=1+0$$

这些乘法的特点是：乘数都是2，被乘数是前一个乘式右边的纯小数部分，最后一个等式的右边，整数部分是1，纯小数部分是0。

对这些乘式从上向下读等式右边的整数部分，恰好也是要求二进纯小数$(0.101)_{二}$。这也不是偶然的巧合。现作一般的说明如下：

假设有一个十进制纯小数M，它的二进制表示是$0.b_1b_2b_3\cdots$，按二进制数的意义，有

$$M=b_1\times2^{-1}+b_2\times2^{-2}+b_3\times2^{-3}\times b_4\times2^{-4}+\cdots$$

第一次用2乘等式两边，得

$$2M=\underbrace{b_1}_{\text{整数}}+\underbrace{b_2\times2^{-1}+b_3\times2^{-2}+b_4\times2^{-3}+\cdots}_{\text{纯小数}}$$

第二次再用2乘等式两边，得

$$2(2M-b_1)=\underbrace{b_2}_{\text{整数}}+\underbrace{b_3\times2^{-1}+b_4\times2^{-2}+\cdots}_{\text{纯小数}}$$

第三次再用2乘等式两边，得

$$2\{2(2M-b_1)-b_2\}=\underbrace{b_3}_{\text{整数}}+\underbrace{b_4\times2^{-1}+\cdots}_{\text{纯小数}}$$

如果纯小数部分没有变为零，那么，这样的乘法可以一直做下去。

这样把从上到下的每一等式的右边的整数依次写出，就得所要求的二进数 $0.b_1b_2b_3\cdots$，这种方法叫做二乘取整法。

下面举例说明常用的计算格式：

0.	8125		
	×2		
1.	6250	（整数部分1是2^{-1}的系数）	书写顺序 ↓
	×2		
1.	2500	（ " 1是2^{-2} " ）	
	×2		
0.	5000	（ " 0是2^{-3} " ）	
	×2		
1.	0000	（ " 1是2^{-4} " ）	

箭头方向表示从上向下读出整数部分，因此，（0.8125）十