电子计算机
程序入门

山西人民出版社

电子计算机程序入门

范逵曦

电子计算机程序入门

范逢曦

*

山西人民出版社出版　（太原并州路七号）
山西省新华书店发行　山西省七二五厂印刷

*

开本：787×1092 1/32 印张：5 1/4 字数：115 千字

1979年10月第1版　1980年3月第1次印刷
印数：1—3,500册

*

书号：15088·98　定价：0.56 元

目　录

第一章　电子计算机中数的表示
§1　数制 …… 1
§2　数的定点表示和浮点表示 …… 8
§3　负数表示法 …… 11
第二章　电子计算机的基本原理
§1　逻辑代数及简单电路 …… 17
§2　双稳态触发器 …… 25
§3　基本逻辑部件 …… 31
§4　电子计算机的组成 …… 38
§5　程序是怎样同计算机联系起来的 …… 51
第三章　程序设计的基本方法
§1　模型机的基本指令 …… 58
§2　简单程序编制 …… 63
§3　分支程序设计 …… 65
§4　循环程序设计 …… 68
§5　主程序与子程序 …… 76
第四章　DJS—100系列机的指令系统
§1　算术和逻辑型指令 …… 83
§2　访问内存型指令 …… 91
§3　输入输出型指令 …… 97
第五章　程序例解

第六章　BASIC语言
§1　关于语言的一些概念 …… 125
§2　BASIC语言的特点和基本量的表示法 …… 127
§3　BASIC语言的基本语句和语法 …… 131
§4　程序举例 …… 145
§5　附录 …… 149
题外篇　当代电子计算机发展的趋向
后　记

第一章　电子计算机中数的表示

§1.数制

1＋1＝2，这个结论对吗?有人会说，当然是绝对正确。这种说法是很片面的。事实上，并不是绝对正确，只可以说这个结论在十进制的数制系统中是正确的，因而说它正确，只具有相对的意义。

我们平常接触到的数制系统很多，主要是十进制，还有六十进制（如钟表），十六进制（如旧秤），八进制以及二进制等等。

十进制数制系统，有0～9十个数字，满足“逢十进一”的位置法则。所谓“逢十进一”的位置法则，就是对一个十进制数，从最低位起，第一位上的一个单位是“一”，称个位，第二位上的一个单位是“一”的10倍，称十位，第三位上的一个单位是“一”的10^2倍，称百位，第四位上的一个单位是“一”的10^3倍，称千位……。这样，任何一个十进制的数，均可写成10的幂的和的形式。

例如

$$3765 = 3000 + 700 + 60 + 5$$
$$= 3\times10^3 + 7\times10^2 + 6\times10^1 + 5\times10^0$$

二进制数制系统，只有两个数字：0，1。满足“逢二进一”的位置法则。所谓“逢二进一”的位置法则，就是对一个二进制数，从最低位起，第一位上的一个单位是“一”，

第二位上的一个单位是“-”的2倍，第三位上的一个单位是“—”的2^2倍……。具体地讲，二进制数制系统，是按如下算式作加法的：

$$\begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array},\quad \begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array},\quad \begin{array}{r} 10 \\ +\ 1 \\ \hline 11 \end{array},\quad \begin{array}{r} 11 \\ +\ 1 \\ \hline 100 \end{array},\quad \begin{array}{r} 100 \\ +\ 1 \\ \hline 101 \end{array},\quad \begin{array}{r} 101 \\ +\ 1 \\ \hline 110 \end{array},$$

$$\begin{array}{r} 110 \\ +\ 1 \\ \hline 111 \end{array},\quad \begin{array}{r} 111 \\ +\ 1 \\ \hline 1000 \end{array},\quad \cdots\cdots$$

从上可知，找到了十进制数制与二进制数制的对应关系：

十进制	二进制
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000

对于二进制的数，也可以写成幂的形式。例如1001可以写成为：

$$1001 = 1\times10^3 + 0\times10^2 + 0\times10^1 + 1\times10^0$$

这里，10不表示“拾”，也不能读作“拾”，它是二进制的10，即相当十进制的2。

既然一个数可以用二进制表示，也可以用十进制表示，那么二进制表示与十进制表示有什么关系呢？二者如何换算呢？我们分两方面回答。

第一，把二进制数化成十进制数。

为区别起见，用$(\quad)_2$表示二进制数，用$(\quad)_{10}$表示十进制数。先举二例说明。

$$\begin{aligned}(111)_2 &= 1\times(10)_2^2 + 1\times(10)_2^1 + 1\times(10)_2^0\\ &= 1\times(2)_{10}^2 + 1\times(2)_{10}^1 + 1\times(2)_{10}^0\\ &= (4)_{10} + (2)_{10} + (1)_{10}\\ &= (7)_{10}\end{aligned}$$

$$\begin{aligned}(1101)_2 &= 1\times(10)_2^3 + 1\times(10)_2^2 + 0\times(10)_2^1\\ &\quad + 1\times(10)_2^0\\ &= 1\times(2)_{10}^3 + 1\times(2)_{10}^2 + 0\times(2)_{10}^1\\ &\quad + 1\times(2)_{10}^0\\ &= (8)_{10} + (4)_{10} + (0)_{10} + (1)_{10}\\ &= (13)_{10}\end{aligned}$$

习惯了就不必这么罗嗦，也不必写$(10)_2$和$(2)_{10}$一类的符号，可直接写成2的幂的和的形式。

例如

$$(1101)_2 = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$$
$$= 8 + 4 + 1$$
$$= (13)_{10}$$

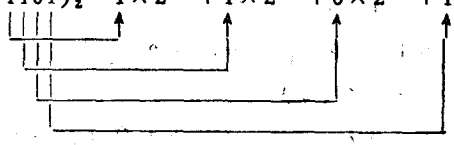
从而推论出二进制整数化成十进制整数的法则为：将二进制整数从最低位向左逐位数去，依次为十进制的1、2、4、8、16、32、……的倍数；它所对应的十进制整数，就是以每位二进制的数（不是0，就是1），乘对应的那个位的倍数的累加和。

例如：

$$(1101)_2 = 1\times8 + 1\times4 + 0\times2 + 1\times1 = (13)_{10}$$
$$(10111)_2 = 1\times16 + 0\times8 + 1\times4 + 1\times2 + 1\times1$$
$$= (23)_{10}$$
$$(11111)_2 = 1\times16 + 1\times8 + 1\times4 + 1\times2 + 1\times1$$
$$= (31)_{10}$$

对于二进制小数化成十进制小数的方法，可用负幂的展开式表达出来。例如：

$$(0.1101)_2 = 1\times2^{-1} + 1\times2^{-2} + 0\times2^{-3} + 1\times2^{-4}$$



$$\therefore\quad (0.1101)_2 = \frac{1}{2} + \frac{1}{4} + \frac{1}{16}$$
$$= \frac{13}{16}$$
$$= (0.8125)_{10}$$

同样地，推论出二进制小数化成十进制小数的法则为：将二进制小数从小数点后的第一位逐位向右数去，依次为十进制的2^{-1}、2^{-2}、2^{-3}、2^{-4}、2^{-5}、……的倍数；其所对应的十进制数，就是以每位二进制数（不是0，就是1），乘对应的那个位的倍数的累加和。

例如：

$$(0.1001)_2 = 1\times 2^{-1} + 0\times 2^{-2} + 0\times 2^{-3} + 1\times 2^{-4}$$

$$= \frac{1}{2} + \frac{1}{16}$$

$$= \frac{9}{16}$$

$$= (0.5625)_{10}$$

第二，把十进制数化成二进制数。这是二进制数化成十进制数的逆运算。也可以从整数化法和小数化法分别叙述。

将十进制整数化成二进制整数的方法，就是将此数以2除，显然，能整除余数为0，不能整除余数为1，记下余数；接着再以上面求得的商以2除之，记下余数；照这样每次用上次求得的商以2除之，记下余数，一直作下去，直至商小于2为止，将此商作为最后余数。我们以最后余数为最高位，最初余数为最低位，依次将所有余数连接起来，就是所要求的二进制整数。现通过具体例子给出一个求法格式。

例：将十进制整数115换算成二进制整数。

具体格式为：

$\therefore (115)_{10} = (1110011)_2$

类似的情形，将十进制小数化成二进制小数的方法，就是将十进制小数乘2，得到“小数点前位值”（为0或为1）和乘积，记下“小数点前位值”；接着再以得到的乘积（使其小数点前位值取0）乘2，又得到新的小数点前位值和乘积，记下小数点前位值；照这样每次用新的乘积（使其小数点前位值取0）乘2，记下乘积和小数点前位值，一直继续下去，有的数可使最后乘积为0，有的数则“最后”乘积不易或不能为0。我们以第一次得到的“小数点前位值”为所得二进制数的小数点后的最高位，依次把各次的小数点前位值连接起来，即为所要求的二进制数。

例：将十进制小数$(0.8125)_{10}$换算成二进制小数。

依照上面的概括，具体以如下格式换算：

$$
\begin{array}{l}
\quad 0.8125 \\
\times \quad \backslash 2 \quad (\text{乘}2) \\
\hline
\quad \boxed{1}.6250 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \text{小数点前位值为}1 \\
\\
\quad 0.6250 \quad (\text{取上次乘积，使“小数点前位值”为}0) \\
\times \quad \backslash 2 \quad (\text{乘}2) \\
\hline
\quad \boxed{1}.2500 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \text{小数点前位值为}1 \\
\\
\quad 0.2500 \quad (\text{取上次乘积，使“小数点前位值”为}0) \\
\times \quad \backslash 2 \quad (\text{乘}2) \\
\hline
\quad \boxed{0}.5000 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \text{小数点前位值为}0 \\
\\
\quad 0.5000 \quad (\text{取上次乘积，使“小数点前位值”为}0) \\
\times \quad 2 \quad (\text{乘}2) \\
\hline
\quad \boxed{1}.0000 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \text{小数点前位值为}1
\end{array}
$$

$\therefore (0.8125)_{10} = (0.1101)_2$

在换算时，乘 2 的次数越多，化得的二进制位数越多。如果要求二进制位数固定时，不管最末次乘2是否得0，满足二进制位数则停止。若最末次乘2后为0（象上例），表示此十进制数与换算后的二进制数完全相等；若最末次乘 2 后不为0，则表示只是换算成近似于它的二进制数。

对于既有整数部分、又有小数部分的十进制数，化为二进制数时，就只按整数部分化整数部分，小数部分化小数部分，然后以小数点为界连接起来即可。比如，前面二例是将十进制整数和小数换算成了二进制整数和小数，显然可得：

$(115.8125)_{10} = (1110011.1101)_2$

有了前述十进制、二进制数制系统的概念，其它进制的数

制系统是很好理解的。比如八进制数制系统，它的规则是“逢八进一”，因而在八进制数制系统中只有0～7等八个数字。实际上八进制和二进制是紧密联系的。根据“逢八进一”的位置法则，三位二进制的数可以构成一位八进制的数。所以求出二进制的数后，从而转换成八进制的数将是很方便的。就以（115.8125）$_{10}$转换成二进制数为（1110011.1101）$_2$又要将此二进制转换成八进制来说，其办法是：以小数点为界，向左每三位划分为八进制数一位（整数部分），向右同样每三位划分为八进制数一位（小数部分）。于是得

$$(115.8125)_{10}=(1\underline{110}\,\underline{011}.\underline{110}\,1)_2=(163.64)_8$$

当然，十进制数直接换算成八进制数时，可以仿照十进制数化二进制数时的办法，整数部分用“8”除记余数，小数部分用“8”乘记小数点前位值。对照上例，读者可自行换算。

§2. 数的定点表示和浮点表示

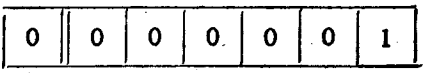
电子计算机中，是以二进制形式表示数的。那么，用几位二进制数来表示一个数字呢？这就是电子计算机中“字长”的概念。所谓“字长”，就是一个数在计算机中，以几位二进制数来表示。比如有的机器字长为16位，就是以16个二进制数位表示一个数。为了反映这个数的正负，在最高位前面加一位表示符号（也有在最低位后面加一位表示符号的）。通常正号（+）用“0”表示，负号（-）用“1”表示。为了叙述和书写方便，假定机器字长为6位，则一个数表示为如下形式：

符号位　　数　字　部　分

对于$\frac{1}{64}$这个数字，可以先换算成二进制，即

$\frac{1}{64}\approx(0.0157)_{10}=(0.000001)_2$，

然后按上面六位（连符号位七位）表示为：



符号位　　数　　字

就一定字长的计算机来说，按小数点的表示方法可以分为两种：定点表示法和浮点表示法。定点表示法，就是小数点的位置是固定的。原则上讲，其小数点的位置固定在那一位无关紧要，但习惯上都把数化成绝对值小于1的数来表示，即小数点在最高位前面，最前面是符号。当然一个大于1的数要缩小一定倍数，才能使之绝对值小于1，在使用计算机时要注意到缩小倍数的问题（通常在程序设计中，把缩小或扩大的倍数，叫做比例因子）。象上面$\frac{1}{64}$所表示的那种形式就是定点表示法。

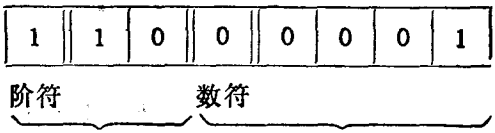
浮点表示法，就是将一个数字分二部分表示，一部分表示其有效数字，称为尾数；另一部分则是指数部分，表示该数的“阶”。还是以$\frac{1}{64}$为例，将上式改写为

$$0.000001=2^{-10}\times 0.0001$$

其中2^{-10}的指数“-10”是以二进制表示的。我们把指数“-10”就称为阶，把0.0001就称为尾数。这里阶是负的，

尾数是正的。如果字长仍以6位考虑，把这6位将分成二部分：阶占二位，尾数占四位，再加上两个符号位：一为阶符，一为数符，其浮点形式为

将$2^{-10}\times0.0001$写成计算机中所表示的浮点形式，则为：

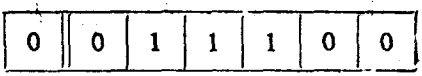


例 以字长6位，将0.0111表示为定点形式和浮点形式。

对0.0111表示为定点时，它为

0.011100

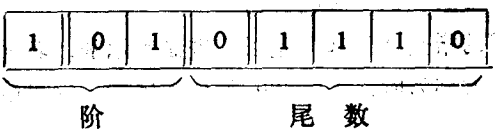
在机器内表示为：



如把0.0111作相应变换得：

$0.0111=0.1110\times2^{-1}$

在机器内表示成浮点形式为



由此可知，定点表示法和浮点表示法效果是相同的。后面将要讲到，字长的每一位以及符号位都是由一种“双稳态触发电路”实现其“寄存”功能的，浮点表示比定点表示除只多一符号位外，设备并无增加，但是浮点表示比定点表示的数的范围扩大了。现仍以字长为6位计算，定点表示的数的范围为：0.000001～0.111111（相当于十进制的$\frac{1}{64}$～$\frac{63}{64}$）；而浮点表示的数的范围为：0.0001×2^{-11}～$0.1111\times2^{+11}$（相当于十进制的$\frac{1}{128}$～7.5）。

一般地说，用定点表示法处理数字的计算机叫做定点计算机。用浮点表示法处理数字的计算机叫做浮点计算机。现在有不少计算机，既可作定点运算，又可作浮点运算，兼有定浮点两种功能。

§3.负数表示法

上面我们讲了符号表示方法，但举的例子都是正数。对于负数如何表示呢？在计算机中，负数通常有三种方法表示，即原码、补码和反码表示法。因为对正数来说，三种表示方法是相同的；而对负数则不相同。

对于原码表示法，可作这样定义：

$$[X]_{原}=\begin{cases}X & 1>X\geqslant 0\\ 1-X & 0\geqslant X>-1\end{cases}$$

从定义可知，正数的原码为其自己，而负数的原码为1－x、写成数学形式，即为：

若$X=0.X_1X_2X_3\cdots\cdots Xn$，它表示X是n位二进制用定

点表示的正数，根据定义则得

$$[X]_{原}=X=0.X_1X_2X_3\cdots\cdots Xn;\quad 且1>[X]_{原}\geqslant 0。$$

如若$X=-0.X_1X_2X_3\cdots\cdots Xn$，它表示X是n位二进制用定点表示的负数，根据定义则得，

$$[X]_{原}=1-X=1+|x|=1+0.X_1X_2X_3\cdots\cdots Xn$$

$$=1.X_1X_2X_3\cdots\cdots Xn;\quad 且2>[X]_{原}\geqslant 1。$$

所以，对于原码的结论是：当X为正数时，$[X]_{原}$就是它自己；当X为负数时，$[X]_{原}$只要将小数点左边位写“1”，表示符号为负，小数点右边数字部分不变。

例如：

$X=0.1011$，则其原码为

$[X]_{原}=0.1011$

↑

表示符号为正

若$X=-0.1011$，则其原码为

$[X]_{原}=1.1011$

↑

表示符号为负

补码表示法的定义为：

$$[X]_{补}=\begin{cases}X & 1>X\geqslant 0\\ 2+X & 0>X>-1\end{cases}$$

补码有类似于上述原码解释的方法。这里只举例对上述