ALGOL60
程序设计

（七〇九机）

电子计算机专业
陈　坚

上海工业大学

目　　录

第一章　关于算法语言的初步知识

第一节　“709”电子计算机简介

709机是一台通用浮点计算机，用国产TTL集成电路元件制成，运算速度每秒十二万五千次，而且配置有ALGOL60编译系统——用机器语言编写成一个加工程序，它能够把用算法语言编写的源程序编排并翻译成用机器语言表示的目标程序。而且它同一般的计算机一样，主要有运算器、存贮器，控制器，外部设备和控制台五个部分，下面分别介绍这五个部分：

（一）运算器：

运算器有三个寄存器，记为“S寄存器”、“L寄存器”、“J寄存器”。S、L、J三个寄存器除了存放参加运算的代码外，还有各自的作用。例如：S寄存器是专门用来存放运算的结果的，同时，与外部设备打交道也都是经过它的；L寄存器在运算中起了累加器的作用；J寄存器是专门同“内存贮器”打交道的，凡是要从存贮器取出一个数，总是先经过J寄存器随后送到S寄存器，同样，要存入一个数，也是先由S寄存器送至J寄存器，再送到存贮器。在计算过程中，S寄存器的内容不断变化，如果在S寄存器里得到的中间结果暂时不用，就必须送到存贮器里保存起来。运算器除了三个寄存器外，还有一组加法器，是具体实现四则运算的（+、－，×，÷，都由加法器实现）。

（二）控制器：

控制器从存贮器里取出一条一条指令与数，根据程序把各种命令传送给计算机的其它部分，通过它去指挥各部件什么时候做什么事情，另外控制器部分还有一台电传打印机，这是进行人机“会话”的设备，根据规定的各种操作规则使用电传打印机，可以使机器启动工作，停止运算，临时输入输出根据以及改变单元内容或运算步骤等等。

(三)存贮器:

又叫内存贮器，是用来存贮代码的。它是一个由成千上万粒磁蕊所构成的“磁蕊体”，而每粒磁蕊的体积比芝麻还小，存贮器划分成32768个单元，每单元字长48位，单元编号为00000～77777。每一个单元只能存放一条代码。00000～00002单元分别对应于控制台单元S_0～S_2，00003单元对应于S寄存器。00004是备用单元00005单元是计算机专用单元77700～77777是变址单元。

(四)外部设备:

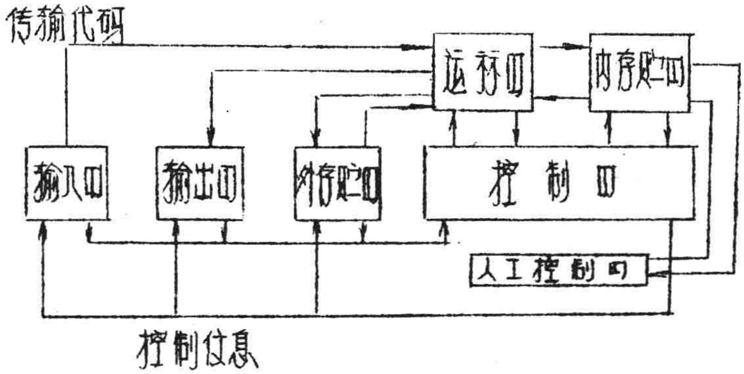
包括输入器、输出器和外存贮器，输入器是二台踏电输入机。初始数据和程序翻成代码，象电报一样，经过穿孔机将孔穿在纸带上，光照在纸带上，在有孔的地方透过去，射到光敏二级管上就变成了电信号。通过这种方式把代码输入到存贮器里，每秒钟可输入1180行。这个光电输入机还能自动检查输入信息的正确性。这二台光电输入机对八单位和五单位两种穿孔纸带都能使用，其输入方式可为八进制，十进制、十六进制、指令型、符号六等五种。输出器是二台宽行打印机，每排可印出80个字符，打印数据的速度为10排/秒。外存贮器是存贮器的一种，主要运用于存放大量暂时不参加运算的代码。当需要时，外存贮器的代码可成批地同存贮器交换，以解决内存贮器容量不够用的矛盾。外存贮器有二台立式磁鼓和二台磁带机，每台磁鼓的容量为一万四千多个单元，外存贮器存取信息的速度比内存贮器慢，应尽量不用外存贮器存取信息。

(五)控制台

控制台由板键开关和氖灯所组成，而控制台上的氖灯向我们显示出机器内部的工作状况。如：控制台上S、L、J三排氖灯，分别显示出S，L，J三个寄存器的状态。灯亮表示“1”，不亮表示“0”。下面“ZJ”旁的15位氖灯显示“指令计数器”的状态，表示机器将要执行那条指令的地址号码。控制台面板上三排板键开关，对应于内存00000～00002号单元，每排有48个板键对应于每个单元的48位代码，板键向上表示“0”，板键向下表示“1”。

此外，还有按钮开关。“总清”按钮一按就把控制面板上指示的寄存器全部清“0”。但应注意“总清”不是“内存总清”。“启动”按钮是命令计算机开始工作的，从哪里开始做，由当时“ZJ”（指令计数器）的状态决定。若是“总清”后再按“启动”就从控制台上S_0的左指令开始工作。启动后是连续执行指令，还是做完一条指令后就停机，由“单拍”或“单指”板键开关的状态决定。“单拍”——计算机启动后执行一个节拍就停机，一般是工作人员用于调 。“单指”——计算机启动后执行一条指令后停机。寄存器显示结果：若“单指”“单拍”都不用就表示连续，连续就表示计算机启动后连续执行一系列指令。

机器各部分之间的联系见图（1－1）



图（1－1）

第二节 电子数字计算机上数的表示形式

电子数字计算机加工、运算的对象是表示为浮点形式的二进制数。这种形式的数同我们日常接触的十进制数不一样，因此，要用计算机来解算生产实际问题，就要了解数在计算机里的各种表示形

式，以及它们之间的转换方法。

一　十进制，二进制，八进制

我们日常生活中，用得最多的数是表示为十进制形式的数，叫做十进制数。这种数的每一位数字都是0、1、2、3、4、5、6、7、8、9十个数字中的一个，而且，数中不同位置上的数字具具有不同的含义。

例如，十进制数2.375表示为

$2.375_{+}=2\times10^{0}+3\times10^{-1}+7\times10^{-2}+5\times10^{-3}$。其中，自左至右的十进位数字2、3、7、5分别是$10^{0}$、$10^{-1}$、$10^{-2}$、$10^{-3}$。数位上的数字。每一位上的数字满了十，就向高一位上进一，所以，十进制就是“逢十进一”的计数制。

除了十进制数之外，我们还经常碰到许多非十进制数，譬如计算时间的时候，60秒钟作为1分钟，60分钟作为1小时，这就是六十进制。在使用电子数字计算机时，我们将要更多地碰到二进制数和八进制数。

二进制数就是数的每一位数字都由0、1两个数字中的一个所构成，而且，数中不同位置上的数字具有不同含义。

例如，二进制数10.011表示为

$10.011=1\times2^{1}+0\times2^{0}+0\times2^{-1}+1\times2^{-2}+1\times2^{-3}=2.375_{+}$。其中，自左至右的二进位数字1、0、0、1、1分别是2^{1}、2^{0}、2^{-1}、2^{-2}、2^{-3}数位上的数字。每一位上的数字满了二，就向高一位上进一，所以，二进制就是“逢二进一”的计数制。

从本例看出，二进制中的10.011就等于十进制中的2.375。

由于二进制数的每一位数字只有0或1，形式很简单，运算也方便，所以，大多数计算机都采用二进制作为数的表示形式。当然，这种二进制形式的数使用时也有不便之外，譬如，它的数位很长，书写起来很累赘，而且，它同习惯使用的十进制数关系也不明显等。

为了弥补这些缺点，所以，我们引进数的八进制形式。因为，用八进制形式表示一个数时，它所占的位数同用十进制表示时相差不大，而且，数的八进制形式同二进制形式有简明的换算关系。

八进制数就是数的每一个数字都是0、1、2、3、4、5、6、7八个数字中的一个，而且，数中不同位置上的数字具有不同的含义。

例如，八进制数2.3表示为

$$2.3_{(八)}=2\times8^{0}+3\times8^{-1}=2.375_{(十)}\ 。$$

其中，2是8^{0}数位上的数字，3是8^{-1}数位上的数字。每一位上的数字满了八，就向高一位上进一，所以，八进制就是“逢八进一”的计数制。

从本例看出，八进制表示中的2.3，等于十进制表示中的2.375，也等于二进制表示中的10.011。

二　数制的转换

（一）数的十进制形式同八进制形式的互化

任何一个数X，在十进制里的表示形式与八进制里的表示形式是不一样的，但它们代表的数值大小是相同的，所以这两种数的表示形式之间有一定的转换关系。

先讲数的十进制形式转换成八进制形式的方法。

1．整数的化法“除八取余”法则。

〔例〕　把十进制整数$75_{(十)}$转换成八进制形式的整数。

假设转换后的八进制整数是$K_nK_{n-1}\cdots K_{0(八)}$，则应有

$$75_{(十)}=K_n\times8^{n}+K_{n-1}\times8^{n-1}+\cdots+K_0\times8^{0}。$$

我们从这个式子中把K_n、K_{n-1}、…、K_0确定出来，为此，两端除以8，得

$$9+3\times8^{-1}=K_n\times8^{n-1}+\cdots+K_0\times8^{-1}\ .$$

比较等式两端得

$$K_0=3\ ,$$

$$9=K_n\times8^{n-1}+K_{n-1}\times8^{n-2}+\cdots+K_1\times8^{0}。$$

从第二式两端除以8，得

$$1+1\times8^{-1}=K_n\times8^{n-2}+K_{n-1}\times8^{n-3}+\cdots+K_1\times8^{-1}.$$

再比较等式两端得

$$K_1=1,$$

$$1=K_n\times8^{n-2}+K_{n-1}\times8^{n-3}+\cdots+K_2\times8^{0}.$$

又从第二式比较两端得

$$K_2=1$$

从而，

$$75_{(十)}=(K_2K_1K_0)_{(八)}=113_{(八)}$$

2．小数的化法——“乘八取整”法则。

〔例2〕 把十进制小数$0.825_{(十)}$转换成八进制形式的小数。

假设转换后的八进制小数是$0.K_1K_2\cdots K_n\cdots_{(八)}$，则应有

$$0.825_{(十)}=K_1\times8^{-1}+K_2\times8^{-2}+\cdots+K_n\times8^{-n}+\cdots.$$

我们从这个式子中把K_1、K_2…、K_n、…确定出来，为此，两端乘8，得

$$6+0.6=K_1+K_2\times8^{-1}+\cdots+K_n\times8^{-n+1}+\cdots.$$

比较等式两端得

$$K_1=6,$$

$$0.6=K_2\times8^{-1}+K_3\times8^{-2}+\cdots+K_n\times8^{-n+1}+\cdots.$$

第二式两端乘8，得

$$4+0.8=K_2+K_3\times8^{-1}+\cdots+K_n\times8^{-n+2}+\cdots.$$

第二式两端乘8，得

$$6+0.4=K_3+K_4\times8^{-1}+\cdots+K_n\times8^{-n+3}+\cdots.$$

比较等式两端得

$$K_2=4,$$

$$0.8=K_3\ 8^{-1}+K_4\times8^{-2}+\cdots+K_n\times8^{-n+2}+\cdots$$

第二式两端乘8，得

$$6+0.4=K_3+K_4\times8^{-1}+\cdots+K_n\times8^{-n+3}+\cdots$$

比较两式得

$K_3 = 6$,

$0.4 = K_4 \times 8^{-1} + K_5 \times 8^{-2} + \cdots + K_n \times 8^{-n+3} + \cdots\cdots$。

如此一直下去，可以求出 K_1，K_2，K_3，…。从而，

$0.825_{(十)} = (0.K_1 K_2 K_3 \cdots)_{(八)} = 0.6463146314\cdots_{(八)}$

3．一般十进制数的化法。

先用上面两种方法，把一般十进制数的整数和小数分别转换成相应的八进制形式，再综合两部分的结果，就得到一般十进制数的八进制形式。

〔例 3〕　把十进制数 $75.825_{(十)}$ 转换成八进制形式。

先把十进制整数 $75_{(十)}$ 按“除八取余”法则转换成八进制形式的整数 $113_{(八)}$：

$$\begin{array}{r|ll} 8 & 75 & \\ \hline 8 & \boxed{9} & 3 \\ \hline & 1 & 1 \end{array}$$

再把十进制小数 $0.825_{(十)}$ 按“乘八取整”法则转换成八进制形式的小数 $0.6463146314\cdots_{(八)}$：

$$\begin{array}{rr} & 0.825 \\ \times & 8 \\ \hline & \boxed{6}.600 \\ \times & 8 \\ \hline & \boxed{4}.800 \\ \times & 8 \\ \hline & \boxed{6}.400 \\ & \vdots \end{array}$$

最后，把两部分结合并起来，即得

$75.825_{(十)} = 113.6463146314\cdots_{(八)}$

如果要把一个八进制形式的数转换成十进制形式，则可以根据八进制数中各位数字的含义直接得出。

〔例 4〕　把八进制数 $15.6_{(八)}$ 化为十进制形式，只要按公式

直接计算即可：

$$15.6_{(八)}=1\times8^{1}+5\times8^{0}+6\times8^{-1}=13.75_{(十)}$$

（二）数的八进制形式同二进制形式的互化

由于 $8=2^3$，所以，八进制数的一位可以用二进制的三位来表示：

八进制	0	1	2	3	4	5	6	7
二进制	000	001	010	011	100	101	110	111

由此，数的八进制形式化为二进制形式时，可以采用“一位拉三位”的方法。

〔例5〕 八进制数 $113.6463_{(八)}$ 化为二进制形式时有

$$113.6463_{(八)}=001001011.110100110011_{(二)}$$

反过来，数的二进制形式化为八进制形式时，应以小数点作为基准，往左、往右都是用“三位并一位”。

〔例6〕 二进制数 $1110.011111_{(二)}$ 化为八进制形式时有

$$1110.011111_{(二)}=16.37_{(八)}$$

三、数的浮点形式

上面提到的数的各种表示形式，有一个共同的特点，就是它们的小数点位置都是固定的。例如，$75.825_{(十)}$ 中，小数点固定在数字5与8之间。再如，$110.011_{(二)}$ 中，小数点固定在左起第三、第四位数字中间；$75.32_{(八)}$ 中，小数点固定在左起第二、第三位数字中间等等。这种小数点具有固定位置的数叫做定点数。

另外，对于任何一个数来说，除了可以表示为定点形式外，还可以表示成浮点的形式，即表示成小数点是浮动的形式。

〔例7〕 十进制数 $75.825_{(十)}$ 可以表示为

$$75.825_{(十)}=+10^{+2}\,0.75825_{(十)}$$
$$=+10^{+3}\cdot 0.075825_{(十)}=\cdots$$

〔例8〕 二进制数$110.011_{(二)}$可以表示为

$$110.011_{(二)}=+2^{+11}\cdot 0.110011_{(二)}$$
$$=+2^{+100}\cdot 0.0110011_{(二)}=\cdots。$$

〔例9〕 八进制数$75.32_{(八)}$可以表示为

$$75.32_{(八)}=+8^{+2}\cdot 0.7532_{(八)}$$
$$=+8^{+3}\cdot 0.07532_{(八)}=\cdots。$$

一般地，任何一个数x都可以表示为

$$x=\pm N^{\pm p}\cdot q。$$

这里，N——数制的底数。十进制数的底数N等于10，二进制数的底数N等于2，八进制数的底数N等于。

p——称为“阶数”，它是N进制表示的正整数。

q——称为“尾数”，它是N进制表示的正小数，即

$$0\leqslant q<1。$$

若数的浮点形式中，有$\frac{1}{N}\leqslant q<1$，则称该浮点数为N进制浮点规格化数、例如，二进制浮点数$+2^{+101}\cdot 0.101001$是浮点规格化的数，而$+2^{+110}\cdot 0.0101001$是二进制非规格化的浮点数。

四、计算机单元中数的表示形式

电子数字计算机要求把参与运算的数表示为二进制浮点形式

$$x=\pm 2^{\pm p}\cdot q，$$

其中，阶数p是二进制正整数，尾数q是二进制正小数。在709机上限定

$$-1000000\leqslant\pm p<1000000，$$
$$0\leqslant q<1。$$

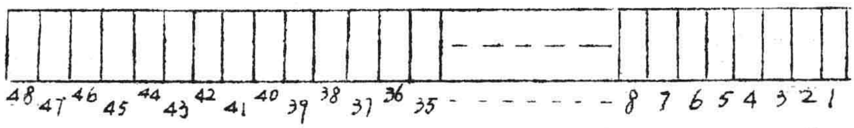
且若$q\leqslant\frac{1}{2}$，则称该数为二进制浮点规格化数。

数字计算机为什么要把数表示为二进制浮点形式呢？这是因为：

第一，二进制数在计算机单元里容易表示。

如前所述，计算机存放数据的单元是由一串互相联系着的小元件构成的。709机的每一个单元有48个小元件，即

709机单元



单元的每一个小元件都具有两种不同的稳定状态（如晶体管的饱和或截止）。如果规定一种状态代表1，另一种状态代表0，那么每一个小元件就可以表示一个二进位数字，因而，在确定正负号位和小数点位置后，一个单元中小元件任何一种稳定状态的组合，就就代表着一个有限位长度的二进制数。譬如，在一个具有48个小元件的单元中，若把第48位定为符号位（用0代表正，1代表负），小数点位置固定在第47、48位之间，这样，这个单元就能把任何一个具有47个二进位数字的定点数表示出来。

第二，浮点形式可以扩大数的表示范围。如果单元中表示的是定点数，则数的表示范围就会受到限制，例如上面的例子就只能把二进制的小数表示出来。为了扩大数在单元中的表示范围，可以采用小数点不固定的浮点形式。

在709计算机单元中，二进制数规定表示为如下的浮点形式；

$$x_{(二)} = \pm 2^{\pm p_1 p_2 p_3 p_4 p_5 p_6} 0.\ q_1 q_2 \cdots q_{39},\ q_{40}$$

其中，阶数 $p_1 p_2 p_3 p_4 p_5 p_6$ 是二进制正整数，尾数 $0.\ q_1 q_2 \cdots q_{39} q_{40}$ 是二进制正小数，且

$$-1000000 \leq \pm p_1 p_2 p_3 p_4 p_5 p_6 < 1000000,$$

$$0.\ q_1 q_2 \cdots q_{39} q_{40} < 1,$$

这个数如果用八进制浮点形式表示，则为

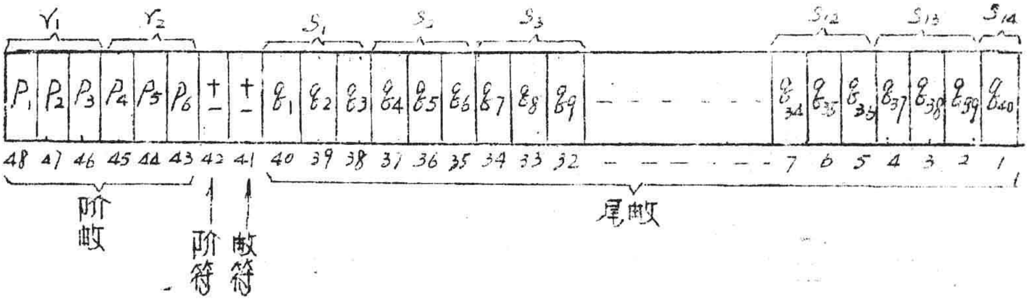
$x_{(八)} = \pm 2^{\pm r_1 r_2} \cdot 0.\ s_1 s_2 \cdots s_{13} s_{14}$ 是八进制正小数，

且

$$-100 \leq \pm R_1 R_2 < 100,\ 0 \leq 0.\ S_1 S_2 \cdots S_{13} S_{14} < 1,$$

二进制浮点数在计算机单元中的形式是

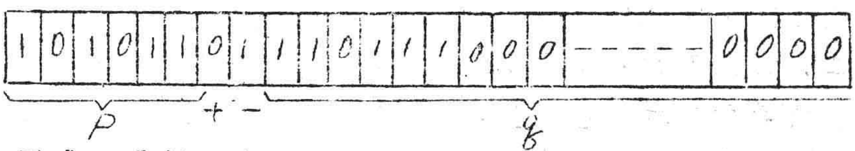
709机单元



其中，阶符和数符分别用0表示正，用1表示负，另外还规定，当阶符为负时，阶数要用$1000000-|P_1P_2P_3P_4P_5P_6|$来代替。

现在举几个例子。

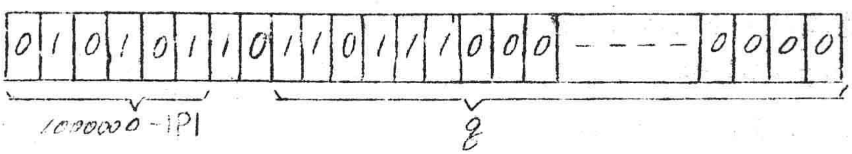
〔例10〕 二进制数 $X_{(二)}=-2^{+101011}$。0.11011100…0 在单元中表示为



写成八进制形式，即

$X_{(八)}=-2^{+53}$。0.67000000…0。

〔例11〕 二进制数 $X_{(二)}=+2^{-101011}$。0.11011100…0 在单元中表示为



写成八进制形式，即

$X_{(八)}=+2^{-53}$。0.67000…0。

由于计算机单元的字长是有限的，那么，它能表示二进制浮点规格化数的数位范围是多少呢？设有一个数为X，因为$X_{(二)}=\pm 2^{\pm P_1 P_2 \cdots P_6} 0.\ Q_1 Q_2 \cdots Q_{40}$。

先考虑最大709机的阶码是六位，尾数是40位，令它为全1即

$$P=\underbrace{+111111_{(2)}}_{6\text{位}}=63_{(10)}$$

$$Q=\underbrace{0.\ 1111\cdots\cdots 11}_{40\text{位}}=\underbrace{0.\ 1111\cdots\cdots 11}_{40\text{位}}+$$

$$\underbrace{0.\ 0000\cdots\cdots 001}_{40\text{位}}-\underbrace{0.\ 0000\cdots\cdots 001}_{40\text{位}}=1-2^{-40}$$

$$X\leqslant 2^{+63}\cdot(1-2^{-40})$$

考虑最小，（考虑到规格化）

$$P=-111111_{(2)}=-63_{(10)}$$

$$Q=-\underbrace{0.\ 1000\cdots\cdots 000_{(2)}}_{40\text{位}}=-\frac{1}{2}$$

$$X\geqslant 2^{-63}\cdot(-\frac{1}{2})$$

所以709机能表示的二进制浮点规格化数的数值范围是

$$2^{-63}\cdot\frac{1}{2}\leqslant|X|\leqslant 2^{+63}\cdot(1-2^{-40})$$

这一范围在十进制记数制中，大约等于$10^{\pm 19}$。

对于绝对值小于$2^{-63}\cdot 1/2$的数，计算机将表示为机器零：$2^{-64}\cdot 0$。若在运算过程中，出现大于机器所能表示的最大的数，则计算机将产生溢出停机。

最后，我们用一个例子写出把一个十进制数化成二进制浮点规格化形式并置入单元的步骤。

〔例12〕 把 $-53.14_{(十)}$ 化成二进制浮点规格化形式后置入单元。

(a)把 $-53.14_{(十)}$ 化成八进制形式。

先用“除八取余”法则把 $53_{(十)}$ 化成八进制整数 $65_{(八)}$:

$$\begin{array}{r|l} 8 & 53 \\ \hline & \boxed{6}\ \ \boxed{5} \end{array}$$

再用“乘八取整”法则把 $0.14_{(十)}$ 化成八进制小数 $0.107\cdots_{(八)}$:

$$\begin{array}{r} 0.14 \\ \times\quad 8 \\ \hline \boxed{1}.12 \\ \times\quad 8 \\ \hline \boxed{0}.96 \\ \times\quad 8 \\ \hline \boxed{7}.68 \\ \times\quad 8 \\ \hline \vdots \end{array}$$

合并上面两项结果，得 $-53.14_{(十)}$ 相应的八进制形式

$-65.107\cdots_{(八)}$。

(b)用“一位拉三位”方法把八进制数 $-65.107\cdots_{(八)}$ 转换为二进制形式:

$-65.107\cdots_{(八)}=-110101.001000111\cdots_{(二)}$。

(c)把二进制数 $-110101.001000111\cdots_{(二)}$ 改写成浮点规格化形式，并置入单元。

$-110101.001000111\cdots_{(二)}$

$=-2^{+000110}\cdot 0.110101001000111\cdots$。

从而，在计算机单元中表示成

0	0	0	1	1	0	0	1	1	1	0	1	0	1	0	0	1	0	0	0	1	1	1	-----	
						+	−																	

五、数的二一十进制形式

如前所述，程序运算的对象是表示为浮点形式的二进制数，这个规定对用惯十进制数的人来说是十分不便的，为了避免直接同二进制浮点数打交道，所以引进数的二一十进制浮点形式。

数的二一十进制形式，就是十进制数在计算机单元中的表示形式，它把十进制数中的每一个数字都用二进制数字来表示。由于十进制数字同二进制数字的关系是

十进制	0	1	2	3	4	5	6	7	8	9
二进制	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

因此，任何一个十进制数，譬如$319_{(+)}$，就可以写出它的二一十进制形式：001100011001。再如，$0.408_{(+)}$的二一十进制形式是0.010000001000。

一般地，对任何一个十进制浮点规格化数

$$X_{(+)} = \pm 10^{\pm P} \cdot Q,$$

它可写为

$$X_{(+)} = \pm 10^{\pm P_1 P_2} 0.\ Q_1 Q_2 \cdots Q_{m-1} Q_m,$$

$$Q_1 Q_2 \pm \pm Q_1 Q_2 \cdots Q_m,$$

这里，规定取M位有效数字（709机单元中，M=10），并限定$P_1 P_2 < 19$，Q_1、Q_2、Q_i（I=1，2，…，M）都是十进制数字。

从而，相应的二一十进制浮点形式规定这样写出：P_1用两个二进制数位表示，P_2和Q_i（I=1，2，…，M）都用四个二进制数位表示，阶符和数符都用0表示正、1表示负。这个数在计算机单元中表示成

709机单元