DJS—21电子数字计算机

实用程序设计

中国人民解放军成字130部队编

兰州化学工业公司科技处翻印

毛主席语录

我们能够学会我们原来不懂的东西。我们不但善于破坏一个旧世界，我们还将善于建设一个新世界。

读书是学习，使用也是学习，而且是更重要的学习。

目　录

第一章　通用电子数字计算机

§1.1　基本结构和工作原理

现代的通用电子数字计算机是采用电子管、晶体管、磁元件、电阻、电容以及其它无线电技术零件所制成的一个复杂的、能高速度完成运算的电子自动装置。它的重要组成部份一般有存储器、运算器、控制器和输入输出器（如图1.1）。

存储器是存储代码的装置。我们把数、指令或一组数字编码统称为代码。

存储器好比一个旅馆，有成千上万个房间，每个房间有固定的编号。我们所谓的存储单元相当于单个房间，单元地址相当于房间编号，而代码就相当于房间中来往的旅客。存储单元的数量（简称存储量）可有数百、数千或者更大。

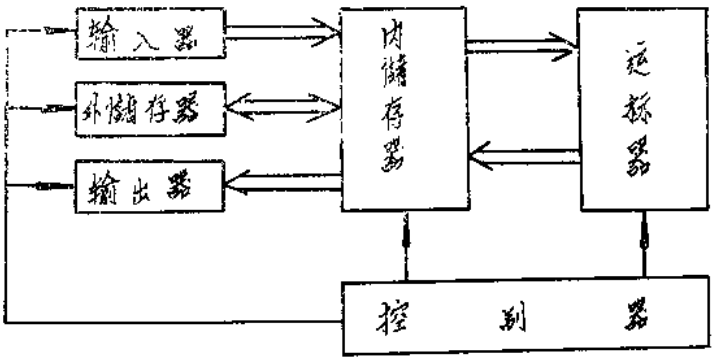


图 1.1

存储器具有接受、保存和发送代码的功能。存于存储器中的代码一般是不会改变的，从任何单元中读出代码后，该代码仍旧存留在单元中，但当送入新的代码时，就会把单元中原存的代码冲掉而代之以新的代码。

在较多的计算机中，存储器分内存储器和外存储器。内存储器与运算器有直接的联系，可以发送代码到运算器参与运算，并能从运算器接受运算结果。外存储器与运算器一般没有直接的联系，但有较大的存储量，并且可以和内存储器成批地进行代码的交换。对于具有大量数据或指令的计算问题，在内存储器不敷使用时，常可利用外存储器来存放计算中暂时不用的数或指令。

运算器是对送入运算器中的代码进行各种运算的装置，它由电子管或晶体管等元件所组成的计算线路构成。

控制器是用来实现机器各部分间的联系，它根据计算程序指挥机器的各部分进行工作，保证了计算过程的自动进行。控制器是由电子管或晶体管所组成的电子门线路构成。

输入器是用来输入原始数据和计算程序，可用穿孔卡片、穿孔纸带或磁带进行输入。**输出器**是用来输出工作结果（数或程序），可用印刷、照相、卡片穿孔、纸带穿孔

或记入磁带等方式进行输出。

输入器、输出器以及外存储器一般统称为外部设备，这些设备因与机械动作有关，因而工作速度较低。机器的其它部件因用电子线路构成，所以可以达到很高的速度，全机各部分用传送代码的路线和控制路线进行互相联系。

在机器上解题时，必须把解题的算法化为机器所能完成的基本运算，并使用机器所能识别的一串数字编码把所要进行的运算表达出来。这些指示机器完成一定操作的一串数字编码就称为指令。指令应指明：进行什么性质的运算，参与运算的数从哪里选取以及运算结果送到哪里去。完成解题所需的一系列指令称为计算程序。

机器完成解题的过程，总的说起来大致是：首先将程序和所需数据送入存储器，然后根据程序的安排，控制器控制机器自动地执行程序所指定的一系列运算，输出所要的运算结果，直到运算完毕自动停机。

机器完成一条运算指令的工作步骤，大致如下：

1、指令由内存储器送入控制器中存放指令的部件。

2、控制器对指令进行分析，发出从哪里取数和做哪种运算的信号。

3、运算器接受信号后，对数进行运算。

4、控制器发出信号，控制机器把运算结果送到指令所指定的地方。

5、机器准备转入下一条指令，并准备执行这条指令。

一架机器除包含运算指令外，为了保证机器的自动工作还需要有其它类型的指令，这些指令的工作步骤与上述可能不同。

根据不同机器的结构，指令列的执行顺序有二种方式：一种是在正被执行的指令中指出下一指令的地址（强制执行式）；另一种，在通常情况下是按指令的存放顺序执行，但当遇到某些特殊指令时，也可变更指令的执行顺序（自然执行式）。

§1.2 数的表示

1. 计数系统

在日常生活中我们习惯采用十进位计数制，在这个计数系统中，有十个不同的数字：0,1,2,…,9,任何数B均用一串数字来表示：

$$B = b_n b_{n-1} \cdots b_1 b_0 . b_{-1} b_{-2} \cdots b_{-m},$$

其中b_i（$i = n, n-1, \cdots, -m$）为十个数字中的一个，且根据其所在位置的不同，具有不同的含意，b_0表示个位，b_1表示十位，…，b_{-1}表示十分之一位，b_{-2}表示百分之一位，…，即数B可表为：

$$\begin{aligned} B &= b_n \cdot (10)^n + b_{n-1} \cdot (10)^{n-1} + \cdots + b_1 \cdot 10 + b_0 + b_{-1} \cdot (10)^{-1} \\ &\quad + b_{-2} \cdot (10)^{-2} + \cdots + b_{-m} \cdot (10)^{-m} \\ &= \sum_{i=-m}^{n} b_i \cdot (10)^i , \end{aligned}$$

括弧中的10即为计数制的基数。

十进位制并不是唯一的计数制，即计数制的基数不一定是10，而可以是其它的任意正整数R，用R作基数的进位制（简称R进位制），对于任意数B可表为：

$$B=\sum_{i=-m}^{n} b_i R^i \quad (b_i \text{为}0,\ 1,\cdots,R-1\text{中的一个数字})$$

并亦可写成：

$B=b_nb_{n-1}\cdots b_1b_0.b_{-1}b_{-2}\cdots b_{-m}$，

b_0表示个位，b_1表示R位，b_2表示R^2位，…，b_{-1}表示R分之一位，b_{-2}表示R^2分之一位，…。

为区分数的不同进位制表示，我们用符号B_R表示数B的R进位制表示。

2.二、四、八、十六进位制；二—十进位制

最简单的进位制是二进位制，这时基数R＝2，并且对任何数只采用二个数字0和1来表示。例如数41_{10}的二进位制表示为：

$101001_2=1\cdot2^5+0\cdot2^4+1\cdot2^3+0\cdot2^2+0\cdot2^1+1\cdot2^0=41_{10}$。

目前有较多的机器是采用二进制计数的，这是由于用二进制有如下一些优点：

（1）可以用任何只具有二个不同稳定状态的元件来记数的每一位，将一种稳定状态表示数字0；另一种表示数字1。制造具有两个稳定状态的元件要比制造多稳定状态的元件容易得多。

（2）二进制的算术运算比较简单，例如：

加法　$0+0=0$，　$1+0=0+1=1$，　$1+1=10$；

乘法　$0\times0=0$，　$1\times0=0\times1=0$，　$1\times1=1$。

运算的简便，可以使机器的结构比较简单。

（3）可以节省存储设备。在说明原因之前，我们先引入“数字——位”的概念。所谓“数字——位”即是为表示在一定范围内的任意数所需的元件数目和元件稳定状态数目的乘积。例如用十进制表示0—999的数需要三位，每位需十个数字，这时的“数字—位”为$3\times10=30$。用二进制表示上面的数需要10位，每位需二个数字，“数字——位”为$10\times2=20$。

我们来作一般性的研究。设要表示从0到N的数，进位制的基为R，需要的位数为n，欲使“数字——位”$X=Rn$取最小值的R。显然我们有：

$$N=R^n-1,\qquad R^n=N+1=\bar{N},$$

$$\ln\bar{N}=n\cdot\ln R,\qquad X=R\frac{\ln\bar{N}}{\ln R},$$

为了求X的最小值，使微商$\frac{dX}{dR}$为零得：$\frac{dX}{dR}=\frac{\ln\bar{N}}{\ln R}-\frac{\ln\bar{N}}{\ln^2 R}=0$。

故$\ln R=1$，即$R=e$（$e\approx2.72$），与e接近的整数为2及3。

三进制的“数字——位”最小，但二进制的其它一些优点是三进制所不及的，故二进制还是较三进制用得普遍些。

固然采用二进制具有一系列的优点，但也有它不便之处，这主要是由于人们习惯于

十进制，因此增加了将原始数据由十进制转换成二进制；计算结果又需从二进制转换成十进制的手续。另外，用二进制表示数的数字较长，书写观看均有不便。

为克服书写不便的缺点，在记写时根据$2^2=4$，　$2^3=8$，　$2^4=16$

的特点，把二进制数以二位、三位、四位为一小节，相应就化为四进位、八进位、十六进位制的数。以这几种进位制来表示数，显然数位是缩短了，且数的本身仍保有二进制的特点。

二进位制　R = 2，　bi = 0或1。

四进位制　R = 4，　bi = 0，1，2，3。每位四进制数以两位二进制数表示时为：

0	1	2	3
↓	↓	↓	↓
00	01	10	11

八进位制　R = 8，　bi = 0，1，2，…，7。每位数以三位二进制数表示时为：

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

十六进位制　R = 16，　bi = 0，1，2，…，$\bar{0}$，$\bar{1}$，　$\bar{2}$，　$\bar{3}$，　$\bar{4}$，　$\bar{5}$。其中数字符号

$\bar{0}$，　$\bar{1}$，　$\bar{2}$，　$\bar{3}$，　$\bar{4}$，　$\bar{5}$表示相当于十进制数的10，11，12，13，14，15。

这时每位数需用四位二进制数来表示：

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

$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$
↓	↓	↓	↓	↓	↓
1010	1011	1100	1101	1110	1111

依照上述二进制数与四、八、十六进制数间的对应关系，要把二进制数化为四、八、十六进制数是十分简便的。例如：

$$11111000_2=3320_4,\quad 11111000_2=370_8,\quad 11111000_2=\bar{5}8_{16}。$$

反过来，知道了四、八、十六进制数后，要求二进制数，只要把每四、八、十六进制数以两位、三位、四位二进制数展开即得。如：

$$123_4=11011_2,\quad 257_8=10101111_2,\quad 13\bar{4}_{16}=100111110_2。$$

二—十进位制　用四位二进位数表示一位十进制数的计数制称为二—十进位制。如十个十进制数字采用下述表示时：

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

数157_{10}的二—十进制表示为：　0001　0101　0111（$_{2-10}$）。

二—十进位制用于某些按十进制计数的机器以及用于在二进制的机器上作为二进制和十进制转换中的中间进位制。在解题时，数制的转换通常是利用程序或特殊设备来实现的，在进行转换时将十进制的原始数据表成二—十进制输入机器，而二进制的计算结果又在机器上转换成二—十进制表示的十进制数，然后再行输出。

3.数的定点表示和浮点表示；规格化的数

例如十进制的432.56这样一个数有下面几种写法：

10×43.256　　$10^3\times 0.43256$　　$10^4\times 0.043256$

是完全相等的。

在机器中，数一般是以上述后面两种形式表示的，即表成$\pm 10^p\cdot d$（$0\leq d<1$）。对于R进制的数，其通式可表达如下：

$$B=\pm R^p\sum_{i=1}^{n} b_iR^{-i}=\pm R^p\times 0.b_1b_2\cdots b_n,$$

$$\left(0\leq\sum_{i=1}^{n} b_iR^{-i}<1\right),$$

其中整指数P决定小数点的位置，称为数的阶，它可正可负亦可为零。$b_1b_2\cdots bn$称为尾数，其中bi（i=1，2，…，n）为0,1,…，R-1中的某个数字，n是尾数的位数。例如十进制的

$$\pm 432.56=\pm 10^3(4\cdot 10^{-1}+3\cdot 10^{-2}+2\cdot 10^{-3}+5\cdot 10^{-4}+6\cdot 10^{-5})$$
$$=\pm 10^3\cdot 0.43256,$$
$$\pm 0.0045=\pm 10^{-2}(4\cdot 10^{-1}+5\cdot 10^{-2})=\pm 10^{-2}\cdot 0.45。$$

阶P等于常数的计算机，称为**定点**计算机。在这种机器中，数被表示成一串数字

$$b_0b_1b_2\cdots b_{n-1}b_n,$$

其中b_0表示数的符号，通常正号用“0”表示，负号用“1”表示。$b_1b_2\cdots bn$是数的尾数。这时小数点的位置是固定的。一般的**定**点机取阶p=0，即小数点固定在b_1之前。这时机器所能表示的数在+1和−1之间。这种机器的结构比较简单，但编制程序比较费事。

阶p是可变的计算机称为**浮点**计算机，这时对于每一个数，除给出尾数外，还需给出数的阶，一般数B在存储单元中的形式为：

$$p_0p_1p_2\cdots p_rb_0b_1b_2\cdots bn,$$

p_0，b_0各表示数的阶和数本身的符号。$p_1p_2p_3\cdots p_r$表示数的阶，$b_1b_2\cdots bn$是数的尾数，而小数点在b_1之前。

浮点计算机表示数的范围较大，程序设计较定点计算机省事，但机器结构要比定点机复杂得多。浮点机器的四则运算按下列规则进行。

加法：　$X=x_1+x_2$　数先对阶（使二数的阶相等），然后尾数相加。

减法：　$X=x_1-x_2$　规则同上。

乘法：　$X=x_1\times x_2$　两数的阶相加，尾数相乘。

除法：　$X=x_1:x_2$　两数的阶相减，尾数相除。

浮点机器中，数的尾数的最高位$b_1\neq 0$时，称为已规格化的数，例如$10^3\times 0.43256$。否则称为未规格化的数，如$10^4\times 0.043256$。

采用二进位制计数时，已规格化的数表为：

$B=\pm 2^{p}\cdot d$，其中$1/2\leq d<1$。

4.数的原码、补码和反码表示

在计算机中，数和阶都要表成一组数字$X=x_0x_1x_2\cdots x_n$，其中x_0表示符号，而$x_1\cdots x_n$是描绘X的一串数字。

一般说来，数有三种表示，即原码、补码及反码，分别记作$(x)_{原}$、$(x)_{补}$、$(x)_{反}$。

我们先假定X是二进制小数，即小数点在x_1之前。

对于正数$X=0.x_1x_2\cdots x_n$，我们规定：

$$X=(X)_{原}=(X)_{补}=(X)_{反}=0.x_1x_2\cdots x_n。$$

下面我们描述X为零或负数时的情况。

原码 数本身为绝对值，其正负号由符号位表示。

若 $X=-0.x_1x_2\cdots x_n=-|X|$，

则 $(X)_{原}=1.x_1x_2\cdots x_n=1+|X|=1-X$。

例 $X=-0.1011$，则$(X)_{原}=1.1011$。

用这种表示法，零有两种可能的形式，分别称为正零及负零。

$(+0)_{原}=0.00\cdots0$，$(-0)_{原}=1.00\cdots0$。

X用原码表示时，其变化范围为$[-(1-2^{-n}),(1-2^{-n})]$。

补码 负数$X=-0.x_1x_2\cdots x_n$的补码表示，具有形状

$$(X)_{补}=1.x_1'x_2'\cdots x_n',$$

式中 $0.x_1'x_2'\cdots x_n'=1-|X|=1+X$，故当$X<0$时

$$(X)_{补}=2-|X|=2+X。$$

例 $X=-0.1011$，则$(X)_{补}=1.0101$，

用这种表示法，零只有一种形式，即

$$(0)_{补}=0.00\cdots0,$$

X用补码表示时，其变化范围为$[-1,1-2^{-n}]$。

综合正数、零及负数的补码表示，可统一写成$(X)_{补}\equiv X\pmod 2$，

由此得出

$$(x+y)_{补}\equiv x+y\equiv(x)_{补}+(y)_{补}\pmod 2。$$

反码 负数$X=-0.x_1x_2\cdots x_n$的反码表示，具有形状

$$(X)_{反}=1.\bar{X}_1\bar{X}_2\cdots\bar{X}_n,$$

式中若 $X_k=1$，则$\bar{X}_k=0$；若$X_k=0$，则$\bar{X}_k=1$ $(k=1,2,\cdots,n)$

故 $(X)_{反}+|X|=1.11\cdots1=2-2^{-n}$。

所以当$X<0$时 $(X)_{反}=2-2^{-n}-|X|=X+(2-2^{-n})$。

例：$X=-0.1011$，则$(X)_{反}=1.0100$。

容易看出，当$X<0$时，$(X)_{补}$与$(X)_{反}$仅在最低位相差1，在$(X)_{反}$的最低位加上1时，即为$(X)_{补}$。

用这种表示法，零也有两种形式：

$(+0)_{反}=0.00\cdots0$，　$(-0)_{反}=1.11\cdots1=2-2^{-n}$，

X用反码表示时，其变化范围为〔$-(1-2^{-n})$，　$(1-2^{-n})$〕

综合正数、零及负数的反码表示，可统一写为：

$(X)_{反}\equiv X\ [\mathrm{mod}(2-2^{-n})]$，

由此推出：

$$(x+y)_{反}\equiv x+y\equiv(x)_{反}+(y)_{反}\ [\mathrm{mod}(2-2^{-n})]。$$

对于X非小数的情形，亦可作类似的讨论，设X的小数点固定在第k位之前，即

$$X=X_0X_1\cdots X_{k-1}X_k\cdots X_n，$$

此时，只需在上面的叙述中，将mod2和mod $(2-2^{-n})$ 用$\mathrm{mod}2^k$和$\mathrm{mod}(2^k-2^{k-1-n})$代替即可。特别当$k=n+1$时，也就是小数点固定在所有的数字之后，此时X为整数。当$X<0$，有

$$(X)_{原}=2^n+|X|，$$
$$(X)_{补}=2^{n+1}-|X|，$$
$$(X)_{反}=(2^{n+1}-1)-|X|。$$

在十进制中，亦可作类似的讨论，例如：

$$X_{10}=-0.123，$$

则

$$(X)_{原}=1.123，$$
$$(X)_{补}=1.877\ (因0.877=1-0.123)，$$
$$(X)_{反}=1.876，$$

其中8，7，6分别是1，2，3对9的补数。

数用原码表示时，作乘除法方便，因乘除时符号与数分别运算，但作加减法不方便，因机器须事先检查两数的符号，才能决定作加法还是作减法。

数用补码或反码表示时，作加减法方便，但作乘除法不方便。

§1.3 指令和指令系统

1. 指　令

当我们用计算机解题时，需要将选定的算法编成程序，程序由一系列的指令组成，每一个指令能使机器执行确定的操作，为了把指令输入机器，须将指令表成数码形式，每条指令包含一个操作码和若干个地址码。操作码指示所规定的操作，地址码指示操作对象及操作结果的存放位置或提供其它信息。

指令的形式一般可表达如下：　$\theta A_1A_2\cdots A_k$，

其中θ为操作码，$A_1A_2\cdots A_k$为地址码。如果一架机器的指令含有K个地址，则称为K地址计算机。常见的有以下几种：

三地址计算机，其指令形式是：　$\theta A_1A_2A_3$

通常A_1，A_2是两个运算对象所在的单元地址，A_3是存放运算结果的单元地址，三地址的指令完全符合于算术运算的逻辑构造，每个算术运算通常有三个数参加，两个作为

参与运算的数，第三个作为运算结果的数。

二地址计算机。指令中两个地址的用法可有多种形式，例如可在两个地址中指明存放运算对象的单元地址，且其中的一个地址又是存放结果的单元地址。或者，在两个地址中，一个指明存运算对象的单元地址，一个指明存结果的单元地址。

一地址计算机的指令形式为：　　θ　A

地址A和各种操作码θ的使用可有各种各样的组合形式。例如一些操作码θ用地址A所存的数进行运算，另一些则向地址A送入结果等等。

在原则上，每个k地址运算都可分解为k个一地址运算。例如对三地址运算可分解为

$\theta_1 A_1$—把A_1中的数送入运算器寄存起来；

$\theta_2 A_2$—把运算器中寄存的数和地址A_2中的数进行操作码为θ_2的运算；

$\theta_3 A_3$—将结果送入单元A_3。

但不能由此得出结论，说一地址计算机的程序比三地址的程序长三倍，因为一般一地址计算机的运算器都有一个或几个寄存器，在连续运算时可用它来保存中间结果，所以程序只比三地址计算机程序稍长一点。

四地址计算机，指令形式为：

$$\theta A_1 A_2 A_3 A_4$$

一般是强制执行式的计算机。$\theta A_1 A_2 A_3$的作用与三地址指令类似，A_4是指明下一次所要执行的指令的地址。

这种强制执行式的指令也有用三地址或二地址的。此时除用一个地址说明下一次执行的指令地址外，其它部分分别与二地址或一地址指令相类似。

由于指令的操作码和地址码都是用数码来表示的，因此在存储单元中的指令也具有数的形式，可以参与运算，改变指令的形式。

2. 指 令 系 统

机器所能执行的基本操作的总和称为机器的指令系统。各种机器的指令系统很不一致，但为保证机器自动完成计算工作，对于通用的计算机，它们都具有以下三类指令：

（1） 算术运算指令，如加、减、乘、除等。

（2） 逻辑运算指令，如逻辑乘法、逻辑加法等。

（3） 控制指令，如改变指令执行顺序、改变机器的工作状态，停机等。

显然，机器的指令系统愈完备，把解题算法写成计算程序就愈容易实现，但如果系统中包含的操作过多，则会使机器的结构复杂化，因此在设计机器的指令系统时，总是兼顾到便利程序设计和简化机器结构这二方面的。

附 录　人工换算数制的方法

在二进制机器上解题时，虽然数制的转换可由机器来完成，但在实际工作中亦常有需要由人工换算少量的数据，因此熟悉一种转换数制的方法是很必要的。

前面我们已介绍过二进制数与四、八、十六进制数之间的换算方法，在机器上的二进位数，人们常用八进制或十六进制数来记录。因此解决了十进制与八进制，或十进制

与十六进制之间的换算，也就解决了十进制数与二进制数间的转换。由于本书以后各章的介绍是以121机为背景的，而121机器的二进制数是用十六进制数来记录的，因此下面我们就只讨论十进制数与十六进制数之间的换算。至于其它不同进位制间的换算，读者可用类似的法则得到。

$X_{10}\rightarrow X_{16}$　　若x_{10}为整数，因

$$x_{16}=x_{10}=b_r\,16^r+b_{r-1}16^{r-1}+\cdots+b_1 16^1+b_0 16^0$$
$$=(b_r\,16^{r-1}+b_{r-1}16^{r-2}+\cdots+b_1)\,16+b_0,$$

故x_{16}的个位数b_0为x_{10}除以16后的整商余数，再继续用16除每次所得的商，我们就将得到16^1位数b_1，16^2位数b_2等等。我们可写出换算规则如下：

以16除x_{10}，设商为q_1，余数为$b_0=x_{16}$的个位数；

以16除q_1，设商为q_2，余数为$b_1=x_{16}$的16^1位数；

以16除q_2，设商为q_3，余数为$b_2=x_{16}$的16^2位数；

……。

当商为零时，余数b_r即为最高位的十六进位数。

例1．$1024_{10}=(\ ?\ )_{16}$

16|1024| 0　　以16除1024，商为64，余数为0；
16|6 4| 0　　以16除 64，商为 4，余数为0；
16| 4 | 4　　以16除 4，商为 0，余数为4。
0

故　　$1024_{10}=400_{16}$。

例2．$2047_{10}=(\ ?\ )_{16}$

16|2047 |15　　以16除2047，商为127，余数为$\overline{5}$；
16|127|15　　以16除127，商为 7，余数为$\overline{5}$；
16|7|7　　以16除 7，商为 0，余数为7。
0

故　　$2047_{10}=7\overline{55}_{16}$

若x_{10}为小数，因

$$x_{16}=x_{10}=b_1 16^{-1}+b_2 16^{-2}+\cdots+b_r\,16^{-r}$$
$$16x_{10}=b_1+(b_2+b_3 16^{-1}+\cdots+b_r\,16^{-r+2})\,16^{-1},$$

故b_1为$16x_{10}$的整数部分，即b_1为x_{16}的第一位小数，再继续用16乘各次剩下的小数部分，就将继续求得x_{16}的第二位小数，第三位小数等等。因此得一般规则为：

以16乘x_{10}，设$c_1=16x_{10}$的小数部分，$b_1=16x_{10}$的整数部分$=x_{16}$的第一位数；

以16乘c_1，设$c_2=16c_1$的小数部分，$b_2=16c_1$的整数部分$=x_{16}$的第二位数；

以16乘c_2，设$c_3=16c_2$的小数部分，$b_3=16c_2$的整数部分$=x_{16}$的第三位小数；

……。

直到求得所需的位数即可停止。

例3．$0.634_{10}=(\ ?\ )_{16}$（取三位有效数字）

0.634×16以16乘0.634，得小数部分为0.144，得整数部分为$\overline{0}$；

10.	144	以16乘0.144，得小数部分为0.304，得整数部分为2；
2.	304	以16乘0.304，得小数部分为0.864，得整数部分为4；
4.	864	以16乘0.864，得小数部分为0.824，得整数部分为$\bar{3}$。
13.	824	

在十六进制中，我们采用七舍八入的办法，于是得

$$0.634_{10}=0.\bar{0}25_{16}。$$

如果x_{10}是混合数，既包含小数又包含整数的数，可将整数和小数分别换算，然后把它们加起来即得。

例4. $1024.634_{10}=400_{16}+0.\bar{0}25_{16}=400.\bar{0}25_{16}$

$X_{16}\rightarrow X_{10}$　　当x_{16}为整数，因

$$x_{10}=x_{16}=b_n16^n+b_{n-1}16^{n-1}+\cdots+b_116^1+b_016^0$$

$$=(\cdots((b_n16+b_{n-1})16+b_{n-2})16+\cdots+b_1)16+b_0。$$

所以作　　$b_n16+b_{n-1}=q_1$，

$q_116+b_{n-2}=q_2$，

$\cdots\quad\cdots$

$q_{n-2}16+b_1=q_{n-1}$，

$q_{n-1}16+b_0=x_{10}$

就是所要求的结果。

实际换算时可采用下列格式：

	b_n	b_{n-1}	b_{n-2}	$\cdots$	b_1	b_0	16
+)		$16b_n$	$16q_1$		$16q_{n-2}$	$16q_{n-1}$	
	b_n	q_1	q_2		q_{n-1}	x_{10}	

例5. $23\bar{5}_{16}=(\ ?\ )_{10}$

	2	3	15	16
+)		32	560	
	2	35	575	

故　　$23\bar{5}_{16}=575_{10}$

当x_{16}为小数，因

$$x_{10}=x_{16}=b_116^{-1}+b_216^{-2}+\cdots+b_{n-1}16^{-(n-1)}+b_n16^{-n}$$

$$=16^{-n}(b_116^{n-1}+b_216^{n-2}+\cdots+b_{n-1}16+b_n)，$$

括号中的数为整数，可用前述整数$x_{16}\rightarrow x_{10}$的方法求出结果，然后除以16^n即得所要求的x_{10}。

例6. $0.\bar{0}2_{16}=(\ ?\ )_{10}$

$$0.\bar{0}2_{16}=16^{-2}(\bar{0}2)_{16}=(16^{-2}\times 162)_{10}=\left(\frac{162}{256}\right)_{10}=0.6328125_{10}。$$

第二章　DJS—21型电子数字计算机

§2.1　一般介绍

DJS—21型电子数字计算机（以下简称121机）是一台中型、晶计管、单地址、通用电子数字计算机。平均运算速度每秒三万条指令。可供国防、工程设计、科学研究以及高等院校等部门，用于解决各种科学研究和大量的工程计算问题。

121电子数字计算机采用二进制数制。并行操作。能作定、浮点，单字长及双倍字长的运算。具有自动变址和中断处理等功能。

121电子数字计算机的内存储器采用磁芯存储器。其容量为8192单元，可扩充至16384单元。存贮周期六微秒。外存储器具有立式磁鼓两台。每台容量32×512个单元。宽行磁带两台，根据需要可以扩充到四台。

121电子数字计算机输入采用RG—2型五单位光电输入机和RG—3型五—八单位光电输入机各一台。输出采用CJ—1型快速打印机。打印速度每秒15行。此外还备有国产六单位电传机（或55型五单位电传机）一台，作为慢速输入输出用。在实现程序自动化时亦可用来打印符号。

此外，121电子数字计算机在指令的按排上为实时控制作了适当的考虑。配上相应的外围设备后可作为实时控制用。

§2.2　数和指令的表示

1. 数的表示

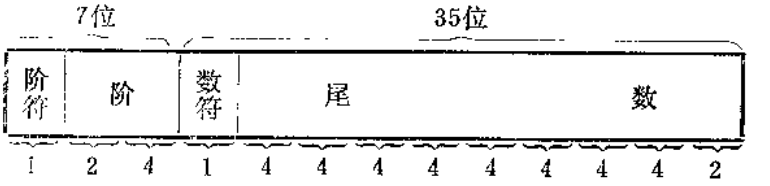
121机的数是采用二进制定点和浮点两种形式表示。

浮点数，共42位，其中阶码7位（包括一位阶符号位），尾数35位（包括一位数符号位）。

表示范围　　$2^{-64} \leq |X| < 2^{+64}$　　（二进制）

$10^{-19} < |X| < 10^{+19}$　　（十进制）

书写形式



定点数，共35位。其中数符一位

表示范围　　$2^{-34} \leq |X| < 1$　　（二进制）