DIANZI

JISUANJI

RUMEN

电子计算机入门

前　言

现代科学技术，以原子能的利用、电子计算机技术和空间科学技术的发展为主要标志。它正在经历着一场伟大的革命，必将引起一系列新兴工业的诞生。

从1946年第一台电子数字计算机问世以来的三十多年，计算机的发展是异常迅速的。它经历了电子管、晶体管、集成电路各阶段，目前又跨入了大规模集成电路的时代。计算机的应用已愈来愈广泛。从科学计算到数据处理，从生产过程控制到军事指挥，从国民经济各部门到社会生活的各个领域，计算机都在发挥着愈来愈大的作用。用计算机控制作成的各种“机械手”、“机器人”正在取代人们的体力劳动，同时，计算机还被用来代替人们部分脑力劳动，成为人类思维和逻辑推理的工具。目前计算机正向巨型、微型、网络、智能模拟方面发展。计算机的科学技术水平、生产规模和应用程度，已经成为衡量一个国家现代化水平的显著标志。

我国第一台电子数字计算机诞生于1958年。在毛主席革命路线指引下，我国又相继在1967年试制成功晶体管大型通用数字计算机，1972年试制成功集成电路通用数字计算机。十几年时间，我国计算机的发展经历了三代，其速度是相当快的。但是由于林彪和“四人帮”的干扰、破坏，我国计算机工业和世

界先进水平之间已经缩小了的差距又拉大了。目前我国广大计算机科学工作者和工人群众，在新时期总任务的鼓舞下，正在以“攻关”的革命精神，向着计算机发展的新阶段进军。在我国社会主义建设事业中，计算机的普及应用也是指日可待了。

为了满足广大工农兵群众和革命干部学习现代科学技术的要求，我们编写了《电子计算机入门》一书。内容以基本的计算机结构为主，兼顾了软件，扩充了计算机发展中出现的新技术，诸如半导体存贮器、微程序控制器等，最后列举了计算机的应用，以引导各行各业的读者考虑在自己的工作领域中大胆采用计算机。本书的写法着重逻辑原理叙述，力求做到使基本概念通俗易懂，整机工作原理清晰，对那些初学计算机而电路知识不多的读者来说，本书可以起到一定的启蒙作用。

由于编者水平有限，书中难免有错误和不妥之处，欢迎批评指正。

编　　者

1978年9月于长沙

目　录

第一章　计算机中的数

§1.1　二　进　制

我们日常碰到的数是由十个符号（0、1、2、3、4、5、6、7、8、9）组成的，用这十个符号可以表示任意大小的数，如5、13、250、1027等等。对于这种数，我们知道数的位置不同(个位、十位、百位、千位等等)，它所代表的数值就不同。

例如，1027表示壹仟零贰拾柒，它可以写为：

$$1027 = \mathbf{1000} \times 1 + \mathbf{100} \times 0 + \mathbf{10} \times 2 + \mathbf{1} \times 7$$

这里**1000**、**100**、**10**、**1**为各位数的“权”。这种计数制叫做“十进制”，即逢十进一。

然而，在电子计算机中，却只用两个符号0与1。那么只用这两个数字0与1能表示2、5、8以及更多位的数吗？可以！在电子计算机中，虽然只有0与1这两个数字，但它可以表示任何大小的一个数。这是因为计算机中使用了一般人们不熟悉，但最简单的二进制计数法。

二进制就是逢二进一的计数制。

零写作“0”；一写作“1”；一加一等于二，逢二进一写作“10”；再加一是三，写作“11”；再加一是四，写作“100”；再加一是五，写作“101”；再加一是六，写作“110”；再加一是

七，写作“111”；再加一是八，写作“1000”；再加一是九，写作“1001”。这样0、1、2、3、4、5、6、7、8、9这十个数都表示出来了。再大的数也能用0与1表示。

表1.1 给出了零到拾伍的十进制数与其二进制数的对应关系。

表1.1

数	十进制	二 进 制	数	十进制	二 进 制
零	0	0 0 0 0	捌	8	1 0 0 0
壹	1	0 0 0 1	玖	9	1 0 0 1
贰	2	0 0 1 0	拾	10	1 0 1 0
叁	3	0 0 1 1	拾壹	11	1 0 1 1
肆	4	0 1 0 0	拾贰	12	1 1 0 0
伍	5	0 1 0 1	拾叁	13	1 1 0 1
陆	6	0 1 1 0	拾肆	14	1 1 1 0
柒	7	0 1 1 1	拾伍	15	1 1 1 1

可见十进制表示的数码与二进制表示的数码是不同的。

那么十进制数与二进制数怎样互相换算呢？

由表1.1抽出十进制的1、2、4、8组成表1.2。

表1.2

十 进 制	二 进 制
1	0 0 0 1
2	0 0 1 0
4	0 1 0 0
8	1 0 0 0

由表1.2可以推想出表1.3。

表1.3

十 进 制	二 进 制
1	0 0 0 0 0 0 0 0 0 1
2	0 0 0 0 0 0 0 0 1 0
4	0 0 0 0 0 0 0 1 0 0
8	0 0 0 0 0 0 1 0 0 0
16	0 0 0 0 0 1 0 0 0 0
32	0 0 0 0 1 0 0 0 0 0
64	0 0 0 1 0 0 0 0 0 0
128	0 0 1 0 0 0 0 0 0 0
256	0 1 0 0 0 0 0 0 0 0
512	1 0 0 0 0 0 0 0 0 0

可以看出二进制各位的“权”从低位算起分别为1、2、4、8、16、32、64、128、256、512等等，其他数都可以由上述“权”相加而得。

例如　二进制数1 0 1 0 1 0 1 0 1 0

$$
\begin{aligned}
1\,0\,1\,0\,1\,0\,1\,0\,1\,0 = {} & 1\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\
& 0\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\
& 0\,0\,0\,0\,1\,0\,0\,0\,0\,0 \\
& 0\,0\,0\,0\,0\,0\,1\,0\,0\,0 \\
+) \quad & 0\,0\,0\,0\,0\,0\,0\,0\,1\,0
\end{aligned}
$$

由表1.3找出它们对应的十进制，并相加得：

$512+128+32+8+2=682$

可知二进制数1 0 1 0 1 0 1 0 1 0 可变为十进制数682。

再例如　十进制数320

因为　　320 = 256 + 64

分别找出256及64对应的二进制数，并相加得：

$$\begin{array}{r} 0\,1\,0\,0\,0\,0\,0\,0\,0\,0 \\ +)\ 0\,0\,0\,1\,0\,0\,0\,0\,0\,0 \\ \hline 0\,1\,0\,1\,0\,0\,0\,0\,0\,0 \end{array}$$

所以，十进制数320可变为二进制数 0101000000。

在电子计算机中除了用二进制外，还用一种“**二——十进制**”，它是用二进制数码来表示十进制的数。在二——十进制中，十进制中的每一位用四位二进制数码来表示。

例如　十进制185

其中，1等值于二进制1，写成四位0001

8等值于二进制1000，写成四位1000

5等值于二进制0101，写成四位0101

所以，185写成二——十进制时为：

0001 1000 0101

将十进制数换成二——十进制数，必须记住表1.4所示的十个基本数。

表1.4

十进制	二进制	十进制	二进制
0	0 0 0 0	5	0 1 0 1
1	0 0 0 1	6	0 1 1 0
2	0 0 1 0	7	0 1 1 1
3	0 0 1 1	8	1 0 0 0
4	0 1 0 0	9	1 0 0 1

不论十进制有多少位，每一位数都用表 1.4 对应的二进制表示。

例如：十进制数28153

二——十进制为：

0010	1000	0001	0101	0011
(2)	(8)	(1)	(5)	(3)

二——十进制是电子计算机外部设备中常用的计数制，它是电子计算机用的二进制与人们日常用的十进制之间的一种过渡性的计数制。

电子计算机内部使用二进制，但是在计算机外面，二进制读起来、写起来不方便，因此计算机使用人员和维护人员常用八进制。

八进制是逢八进一的计数制。8 是 2 的 3 次方($8=2^3$)，所以二进制和八进制之间关系很简单。把二进制从个位起每三位做一组读成(或写成)八进制的一位，这样三位、三位分下去，最后不足三位也当作八进制读(或写)。这样就把二进制换算成八进制了。

例如　二进制数　11111011101100

将其三位作一组：

	11	111	011	101	100
	↓	↓	↓	↓	↓
写成八进制为	3	7	3	5	4

故二进制数　11111011101100 相当于八进制数“37354”。

将八进制换成为二进制办法也是一样的。

由于二进制数与八进制数的换算如此简便，因此在电子计算机中用的二进制数码，如果人们需要读(写)，可以读(写)成八进制，这样作非常方便。

§1.2 定点与浮点

电子计算机中参加运算的数有正数、负数、整数、小数等等。

那么计算机中数是怎样表示的呢？

计算机中的数，一般有两种表示方法："定点法"与"浮点法"。采用定点法的叫做定点计算机，采用浮点法的叫做浮点计算机。

定点法：

在数学里处理正数与负数问题时，将数的绝对值前边加一符号，"+"表示正数，"－"表示负数，如＋25，－36等等。在定点机中，为了区分正、负数，每个数的前面也用符号位来表示。

一般是：用"0"表示正数

用"1"表示负数

在定点机中，一个数的最左边的一位是符号位，小数点固定在符号位之后，符号位右边的第一位就是小数点后面的第一位。

例如从定点机中读出的两个数是：

0101101 及 1101101

我们应该想到它们分别表示：

$$+0.101101 \quad 及 \quad -0.101101$$

如果计算机能表示六位数，带符号位为七位，则它所能表示的绝对值最大数为 0.111111；所能表示的绝对值最小数为 0.000001。

因此，六位数所能表示的数的绝对值的范围为：

$$0.000001 \sim 0.111111$$

显然，在定点机中参加运算的数绝对值必须小于 1。

自然有人会问：一个算题中，一定要求初始数据、中间结果、最后结果绝对值都不得大于 1，这怎么能作得到呢？在算题中，若这些数据绝对值大于 1 怎么办呢？

办法是这样的：在定点机中，在算题之前，要选择合适的比例尺，相当于将算题中的初始数据、中间结果、最后结果都缩小若干倍。

如果由于考虑不周，经过某种运算所得结果数值绝对值大于 1，最高位的数就被丢掉了，因而结果也就不正确了。发生这种情况叫做“**过满**”或“**溢出**”。为了防止“过满”现象，在算题之前必须考虑好比例尺，使得初始数据，中间结果、最后结果不超过计算机所能表示的数的最大数值。

浮点法：

浮点法表示数是指小数点的位置不是固定的，而是浮动的。

例如，十进制数29.5，它可以变换为下列各种形式：

29.5	29.5
2.95×10^{1}	295×10^{-1}
0.295×10^{2}	2950×10^{-2}
等等	等等

小数点每向前移一位，表示小了10倍，再乘以10，等于没有变化。	小数点每向后移一位，表示大了10倍，再除以10，等于没有变化。

同样地，二进制数1010.11，它可以变为下列各种形式：

$101.011\times\mathbf{10}^{1}$ 或 $10101.1\times\mathbf{10}^{-1}$

$10.1011\times\mathbf{10}^{10}$ $101011\times\mathbf{10}^{-10}$

$1.01011\times\mathbf{10}^{11}$ 等等

注意上述表示式中全部是用二进制表示的。二进制 $\mathbf{10}^{1}$、$\mathbf{10}^{10}$，$\mathbf{10}^{11}$等等与十进制对应关系如表1.5所示。

一般来说，二进制数N可以写为：

$$N = M\cdot\mathbf{10}^{P}$$

注意这里N，M，P均为二进制表示。10也为二进制，相

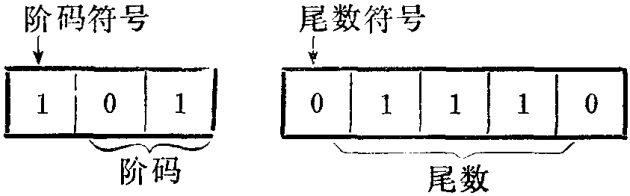
表1.5

二 进 制	十 进 制
$\mathbf{10}^{1}$	2^{1}
$\mathbf{10}^{10}$	2^{2}
$\mathbf{10}^{11}$	2^{3}
$\mathbf{10}^{-1}$	2^{-1}
$\mathbf{10}^{-10}$	2^{-2}
$\mathbf{10}^{-11}$	2^{-3}

当于十进制数的2。

上式中，称M为**尾数**，P为**阶码**。

假设计算机用六位表示数，四位表示尾数，二位表示阶码，对于0.0111而言，其浮点形式为0.1110×10^{-1}。在计算机中，该数将如下表示：



为了提高运算精度以及在算术运算中可以得到全部有效数字结果，在浮点计算机中通常采用“**规格化数**”表示法。

所谓规格化，就是说如果尾数第一位上的数字为1时，此数就是已规格化的数。反之，如果尾数第一位上为0时，此数就是未规格化的数。

例如 0.1110×10^{10} 为已规格化的数。

0.0111×10^{11} 为未规格化的数。

从表示数的范围来看，定点法表示数的范围是较小的，而浮点法表示数的范围较大。例如不算符号位在内，定点六位二进制，它能表示的数的绝对值的范围为0.000001到0.111111，相当于十进制数的$+\frac{1}{64}$到$+\frac{63}{64}$，而四位表示尾数，二位表示阶码的六位浮点形式，它能表示的数的范围，其绝对值为$+0.0001\times10^{-11}$到$+0.1111\times10^{+11}$，相当于十进制数$+\frac{1}{128}$

到 $+7\frac{1}{2}$。实际上计算机位数都在二进制16位以上，那么定点法与浮点法表示数的范围相差就更大了。

定点法使机器结构简单，所以小型机或专用机用的较多。浮点法数的表示范围大，使用方便，但结构复杂，一般是大型计算机采用它。

§1.3 原码、补码、反码

电子计算机是对二进制数进行运算的。二进制运算是相当简单的。下面介绍二进制加、减、乘、除是怎样运算的。

加法：二进制一位加法规则是：

$$\begin{array}{cccc} 1 & 0 & 0 & 1 \\ +\ 0 & +\ 1 & +\ 0 & +\ 1 \\ \hline 1 & 1 & 0 & 1\ 0 \end{array}$$

↑——向高位进位

即 $1+0=1$

$0+1=1$

$0+0=0$

$1+1=0$（向高位有进位）

利用此规则很容易实现加法运算。

例如 $5+6=?$

十进制

$$\begin{array}{c} 5 \\ +\ 6 \\ \hline 1\ 1 \end{array}$$

二进制

$$\begin{array}{c} 0\ 1\ 0\ 1 \\ +\ 0\ 1\ 1\ 0 \\ \hline 1\ 0\ 1\ 1 \end{array}$$

减法：二进制一位减法规则是：

$$\begin{array}{cccc} 1 & 0 & 0 & 1 \\ -\,0 & -\,0 & -\,1 & -\,1 \\ \hline 1 & 0 & 1 & 0 \end{array}$$

向高位借1

即 $1-0=1$

$0-0=0$

$0-1=1$（向高位借1）

$1-1=0$

利用此规则可以实现二进制数的减法运算。

例如 $11-6=?$

十进制　　二进制

$$\begin{array}{r} 11 \\ -\ 6 \\ \hline 5 \end{array} \qquad \begin{array}{r} 1\ 0\ 1\ 1 \\ -\ 0\ 1\ 1\ 0 \\ \hline 0\ 1\ 0\ 1 \end{array}$$

乘法：二进制一位乘法规则是：

$$\begin{array}{cccc} 1 & 0 & 0 & 1 \\ \times\,0 & \times\,0 & \times\,1 & \times\,1 \\ \hline 0 & 0 & 0 & 1 \end{array}$$

即 $1\times0=0$

$0\times0=0$

$0\times1=0$

$1\times1=1$

可见乘法规则特别简单，除了1×1=1外，其他情况均为0。

例如　13×5=？

十进制　　二进制

```
  13          1101
×  5        ×  101
  65          1101
            0000
          +)1101
           1000001
```

从上例可以看出：当乘数为1时，相当于把被乘数照抄一遍，只是它的最后一位与相应的乘数位对齐，当乘数为0时，实际上没有任何作用，所有的乘数位算完之后，把各部分乘积加起来便可得到结果。实际上二进制的乘法是由加法和移位操作来实现的。

除法：除法是乘法的逆运算。

例如　65÷5=？

十进制　　二进制

```
  13          1101
5)65    101)1000001
  5           101
  15          110
  15          101
   0            101
               101
                 0
```

由上例可以看出：二进制除法实质上是由减法和移位两种