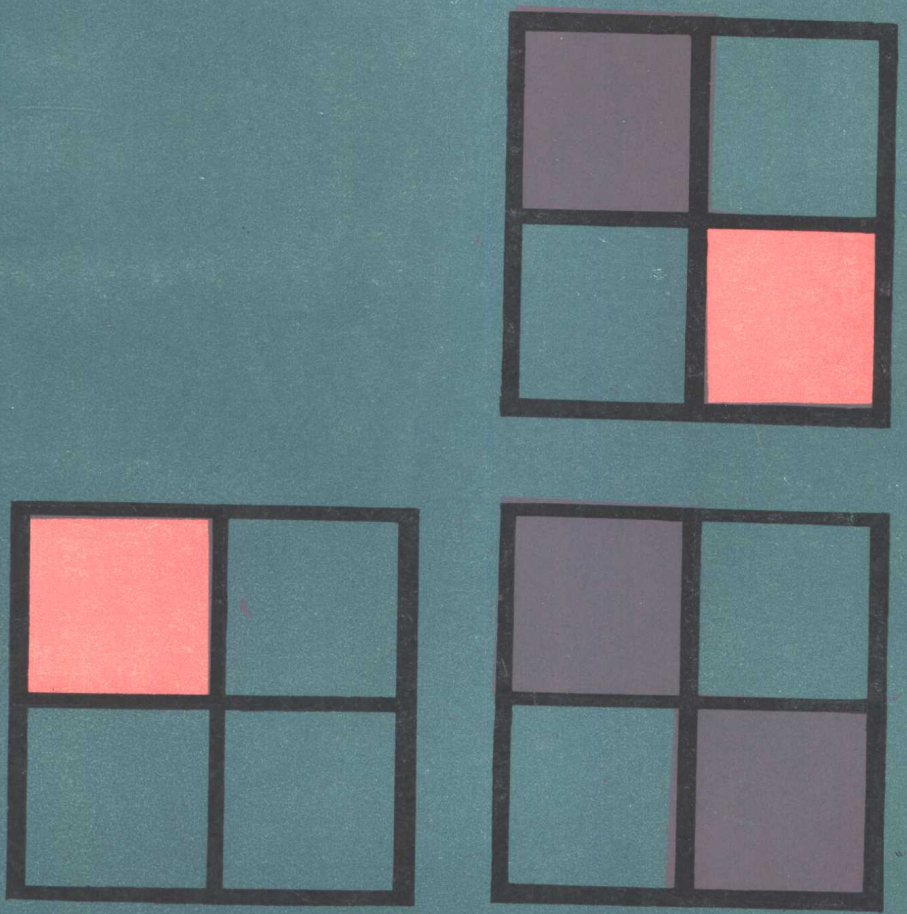
师范专科学校试用教材

逻辑代数与电子计算机简介

《逻辑代数与电子计算机简介》编写组



师范专科学校试用教材

逻辑代数与电子计算机简介

《逻辑代数与电子计算机简介》编写组

高等教育出版社

内 容 提 要

本书是为师范专科学校、教育学院和教师进修学院数学专业开设与计算机有关课程而编写的一本试用教材。主要内容有：集合代数，逻辑代数，电子计算机简介和BASIC语言。本书可作为师专教材或教学参考书，也可供中学教师以及其他人员阅读参考。

责任编辑：纪涌

师范专科学校试用教材

逻辑代数与电子计算机简介

《逻辑代数与电子计算机简介》编写组

*

高等教育出版社出版

新华书店北京发行所发行

人民教育出版社印刷厂印装

*

开本 850×1168 1/32　印张 8.75 字数 210,000

1985 年 5 月第 1 版　1985 年 5 月第 1 次印刷

印数 00,001— 42,600

书号 13010·01049　　定价 1.75 元

前　言

本书是根据一九八二年十月教育部在昆明召开的师专数学专业教学大纲审订会审订的《逻辑代数与电子计算机简介教学大纲》而编写的，并根据当前中学教学的需要，在大纲的基础上增加了BASIC语言的部分内容．它可以作为师范专科学校，教育学院和教师进修学院数学专业的基础课教材或教学参考书，也可供中学教师作为学习有关内容的参考读物．

本书在原贵阳师院和万县师专马自甦、吕传汉、余柏年编写的《逻辑代数与电子计算机简介》讲义以及上海师院邵存蓓编写的《BASIC语言》讲义的基础上写而成．全书共分两篇，第一篇《逻辑代数》由重庆计委计算中心（原万县师专）马自甦（第一，二章）和湖南常德师专黄根民（第三，四章）执笔，第二篇《BASIC 语言》由贵阳师院余柏年（第一、五章）和上海师院邵存蓓（第二、三、四章）执笔．在互相校阅、讨论的基础上，由邵存蓓统稿．

在本书的编写过程中，得到了陕西师大张德荣、芜湖师专刘为铨、烟台师院陈书典以及洛阳师专张鹏祥等老师的大力支持与帮助．他们认真地审阅了全稿，提出了十分宝贵的修改意见，使本书质量有了明显的提高．尤其是张德荣老师为本书的编写付出了辛勤的劳动，在此一并表示谢意．

另外，编者还要感谢贵阳师院的吕传汉老师，他在本书初稿的形成过程中做了很多工作．

《BASIC 语言》中的上机实习是一个不可缺少的教学环节，采用本书作为教材的学校可根据具体设备条件自行安排．

由于编者水平有限，实践经验不足，加上编写时间仓促，缺点、错漏之处在所难免，恳请读者批评指正。

编　　者

一九八四年六月

目　录

第一篇 逻辑代数

第一章 二 进 制

人类从对数的认识到掌握数的进位制，经历了一个漫长的历史过程．可以说，进位制记数法是人类最早最重大的数学发明，是件了不起的数学成就．当今电子计算机的广泛应用，更促进了人们对数的进位制及其运算的研究．由于电子计算机中一般采用的是二进制记数法，因此，在学习逻辑代数与电子计算机的基本知识之前，有必要首先介绍二进制．

众所周知，算盘是一种十进制的运算工具，它的每一个位上有七颗算珠，这七颗算珠可以组成十六种不同的状态．我们选定其中十种状态分别代表 0，1，2，…，9 这十个数码．这十种状态的区别是很鲜明的，并且很容易从一种状态变成另一种状态．

电子计算机如果采用十进制进行计算，电子元件就必须能鲜明地反映出十种不同的状态，并且能够从一种状态容易地转变成另一种状态．这给计算机的设计和制造带来很大的困难，这是很不经济的．

然而，如果要电子元件鲜明地反映两种状态，譬如，通电与截断，高电压与低电压，有脉冲与无脉冲，正向磁化与负向磁化等，则比较容易实现，而且，这两种物理状态能够灵敏地从一种转变成另一种．如果假设用某个电子元件的正向磁化状态代表“1”，负向磁化状态代表“0”，那么这样的电子元件就可以用来表示二进制的

一个数码．将这样的四个电子元件排列起来，就可以表示任何一个二进制四位数．

由于只要求电子元件反映两种状态，这就使计算机的设计和制造变得容易．所以，电子计算机不采用习惯的十进制，而采用二进制．

§1 数的进位制

为了讲清进位制，先从大家熟悉的十进制说起．

在生产和日常生活中，人们最常用的计数法是十进制计数法，它来源于人类最初用两手十指进行计数的实践．有时也用到非十进制的计数法，例如：一打铅笔是 12 支，是 12 进位的；1 小时是 60 分，是60进位的等等．

一个十进制数是用十个不同的数字符号 0, 1, 2, …, 9 中的某一些符号按照一定法则排列再冠以正负号来表示的．这十个数字符号我们称之为数码．一个数码处于不同的位置（数位）就有不同的位置值（或权），因而所表示的数值也就不同，这就是所谓位置原则．例如十进制数 109.09 的小数点前的“9”表示 9 个 1，小数点后的“9”表示 9 个 1%．作为特殊的数码“0”，它处于任何位置时都表示同一个数“0”．

数 109.09 实际上是下面写法的缩写：

$$1\times 10^{2}+0\times 10^{1}+9\times 10^{0}+0\times 10^{-1}+9\times 10^{-2}=109.09.$$

一般地，任一个十进制数 s（不妨设为正数）都可表为：

$$\begin{aligned}s&=k_n(10)^{n}+k_{n-1}(10)^{n-1}+\cdots+k_1(10)^{1}+k_0(10)^{0}\\&\quad+k_{-1}(10)^{-1}+\cdots+k_{-m}(10)^{-m}\\&=\sum_{i=n}^{-m}k_i(10)^{i}.\end{aligned}$$

其中 k_i 是 0, 1, 2, …, 9 十个数码之一，m、n 为正整数，10 称为基

数．所谓基数，就是该进位制中所有可能用到的数码的个数．实际上，基数可以为任意大于 1 的正整数 p．如前所述的 12 进位制的基数为 12，60 进位制的基数为60，2 进位制的基数为 2 等等．

仿照十进制数的写法，任一 p 进制数 s 可以表为：

$$\begin{aligned} s &= k_n p^n + k_{n-1} p^{n-1} + \cdots + k_1 p^1 + k_0 p^0 + k_{-1} p^{-1} \\ &\quad + \cdots + k_{-m} p^{-m} \\ &= \sum_{i=n}^{-m} k_i p^i, \end{aligned}$$

其中 k_i 可以是 0, 1, 2, …, $(p-1)$ 中的任一个数码，它和一个固定的 p^i 相对应．p^i 称为各个数位上的位置值（或权）．上式也称为数 s 按权的展开式．

在 p 进制中，相邻两个数位的权相差 p 倍．对 p 进制小数而言，小数点左移一位原数缩小 p 倍，即乘了 $\frac{1}{p}$；右移一位则扩大 p 倍，即乘了 p．用 p 进制计数时，按“逢 p 进一，退一当 p”的进退位原则进行．

p 进制数各位的名称规定为：数码 k_i 所在的数位叫做 p 的 i 次幂位．例如二进制数 101.11 的各个数位的叫法是：

1	0	1	.	1	1
2^2 位		2^1 位		2^0 位	2^{-1} 位	2^{-2} 位

§2 二进制数

在 p 进制数的按权的展开式中，令 $p=2$ 即得二进制数：

$$\begin{aligned} s &= k_n 2^n + k_{n-1} 2^{n-1} + \cdots + k_1 2^1 + k_0 2^0 + k_{-1} 2^{-1} + \cdots \\ &\quad + k_{-m} 2^{-m}, \end{aligned}$$

其中 k_i 只取数码“0”和“1”，它由 s 的值决定，m、n 为正整数．

在二进制数中，相邻两个数位的权相差 2 倍．用二进制计数

时，按“逢二进一，退一当二”的原则进行.

通常将按权的展开式写成如下缩写形式：

$$s=k_nk_{n-1}\cdots k_1k_0.k_{-1}k_{-2}\cdots k_{-m}.$$

要用电子计算机进行计算，就必须将十进制数转变成二进制数. 另一方面，电子计算机输出的结果也必须还原成十进制数，否则我们难以看懂. 因此，我们来研究二进制数与十进制数相互间的转换. 为了区分两种进制的数，必要时在数的右下角加标注. 通常对十进制的标注省略不写.

任给一个二进制数，很容易把它转换成等值的十进制数. 方法是将二进制数按权展开，然后按十进制计算其值. 例如：

$$\begin{aligned}11011.1011_{(2)}&=1\times2^4+1\times2^3+0\times2^2+1\times2^1+1\times2^0+1\times2^{-1}\\&\quad+0\times2^{-2}+1\times2^{-3}+1\times2^{-4}\\&=16+8+2+1+\frac{1}{2}+\frac{1}{8}+\frac{1}{16}\\&=27.6875.\end{aligned}$$

上式表明，二进制数 11011.1011 等于十进制数 27.6875.

将一个十进制数化成二进制数的方法在下一节讨论.

我们看到，二进制数位数太多，有时使用不方便，为此，引入八进制数作为二进制数与十进制数的中间过渡.

在一般 p 进制中，取 $p=8$ 就得到八进制.

任给一个八进制数如 625.1，按权展开并计算其值，就可将八进制数转换成十进制数：

$$\begin{aligned}625.1_{(8)}&=6\times8^2+2\times8^1+5\times8^0+1\times8^{-1}\\&=384+16+5+0.125\\&=405.125.\end{aligned}$$

即八进制数 625.1 等于十进制数 405.125.

下面讨论二进制数与八进制数的相互转换关系.

由重复排列的知识知，所有可能不同的三位二进制数（包括首位为0的情形）共有八个，将这八个三位的二进制数由小到大顺次列出，然后转换成十进制数，可得表1。

表1

二进制数	000	001	010	011	100	101	110	111
十进制数	0	1	2	3	4	5	6	7

以上八个十进制数恰好是八进制中的八个数码，表1表示出三位二进制数与一位八进制数的对应关系．按照这个对应关系，我们可以实现八进制数与二进制数的相互转换．

譬如，将八进制数625.1转换成二进制数，我们先作如下变化：

$$625.1_{(8)}=6\times8^2+2\times8^1+5\times8^0+1\times8^{-1}$$
$$=(2^2+2)\times2^3+2\times2^3+(2^2+2^0)\times2^0$$
$$+1\times2^{-3}$$
$$=2^8+2^7+2^4+2^2+2^0+2^{-3}$$
$$=110010101.001_{(2)}.$$

注意上式两端的特点，我们发现有如下规律：将八进制数的每一位换成对应的三位二进制数，按原来的顺序排列起来即得到这个八进制数所对应的二进制数．

将以上过程反推转去即可总结出将一个二进制数转换成八进制数的方法：

将二进制数从小数点开始，分别向左和向右依次将每三位分成一组，譬如，

二进制数10110110011.01100111分组后成为：

10　110　110　011.011　001　11

如果首末两组不足三位可以在前和在后添0：

010　110　110　011．011　001　110

最后把每组三位二进制数按对应关系写出八进制数，按此法则，上述二进制数转换成八进制数就是 2663.316.

可见，二进制数与八进制数的相互转换是很简单的．采用八进制书写二进制数，位数约减少到原来的三分之一．

下面讨论二进制数的算术运算．

1. 二进制的加法规则如下：

$0+0=0,\quad 0+1=1,\quad 1+0=1,\quad 1+1=10.$

此规则还可以写成更紧凑的格式——加法表：

表 2　加法表

+	0	1
0	0	1
1	1	10

此规则的特点是“逢二进一”．

例 1　求 $101+111=?$　　$101+11=?$

解　写成竖式：

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

2. 二进制的减法规则如下：

$0-0=0,\quad 1-0=1,\quad 1-1=0,\quad 10-1=1.$

此规则的特点是退一当二．

例 2　求 $101-10=?$　　$1011-110=?$

解　写成竖式：

$$\begin{array}{r} 1\,0\,1 \\ -\;\;1\,0 \\ \hline 1\,1 \end{array} \qquad \begin{array}{r} 1\,0\,1\,1 \\ -\;1\,1\,0 \\ \hline 1\,0\,1 \end{array}$$

由以上二进制的加法和减法的规则可以验证，二进制的加法适合交换律和结合律，而减法既不适合交换律又不适合结合律，这和十进制的情形是一样的.

3. 二进制的乘法规则如下：

×	0	1
0	0	0
1	0	1

例 3 求 $1101\times1010=?$

解 写成竖式：

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

从此例看出：当乘数为 1 时，把被乘数照写一遍即得部分积；当乘数为 0 时，部分积为0(实际计算时，这一行用不着写). 由此，二进制的乘法实际上是由加法和移位操作来实现的. 由乘法规则还可验证，十进制乘法的交换律、结合律、乘法对于加法的分配律，在二进制的乘法中也是成立的.

4. 二进制的除法.

二进制的除法运算与十进制一样，也是乘法的逆运算.

例 4 求 $110101\div101=?$

解 写成竖式：

$$\begin{array}{r} \quad\ 1\ 0\ 1\ 0\cdots\cdots\text{商数} \\ 1\ 0\ 1\ \overline{)\ 1\ 1\ 0\ 1\ 0\ 1} \\ \quad\ 1\ 0\ 1 \\ \hline \quad\quad 1\ 1\ 0 \\ \quad\quad 1\ 0\ 1 \\ \hline \quad\quad\quad 1\ 1\cdots\cdots\text{余数} \end{array}$$

由此例看出，二进制的除法实质上是由减法和移位操作来实现的．显然，除法也不适合交换律和结合律．

综上所述，由于电子计算机采用了二进制，使得计算机中只要有能实现加法运算的加法器和能实现移位操作的移位器就能完成全部算术运算，从而使计算机的设计和制造变得容易，这是计算机采用二进制的又一大优点．

§3 化十进制数为二进制数

为了叙述方便，我们结合具体例子来说明方法．

例1 将十进制数 405 化成二进制数．

解 设 $405=k_nk_{n-1}\cdots k_1k_{0(2)}$．

我们的目的是求出 k_j 的值．

将等式右端按权展开：

$$\begin{aligned}405&=k_n2^n+k_{n-1}2^{n-1}+\cdots+k_12^1+k_02^0\\&=2(k_n2^{n-1}+k_{n-1}2^{n-2}+\cdots+k_1)+k_0.\end{aligned}$$

等式两端同除以 2：

$$202+\frac{1}{2}=(k_n2^{n-1}+k_{n-1}2^{n-2}+\cdots+k_1)+\frac{k_0}{2}.$$

要等式成立，必须要等式两端整数和小数部分对应相等，于是 $k_0=1$，它正好是 $405\div2$ 的余数．

等式两端划去$\frac{1}{2}$后再同除以2：

$$101=(k_n2^{n-2}+k_{n-1}2^{n-3}+\cdots+k_2)+\frac{k_1}{2}.$$

比较等式两端得 $k_1=0$，它正好是 $202\div2$ 的余数．

将以上过程继续下去，直到商数等于 0 时为止．

整个计算过程和结果可用竖式表述如下：

$$
\begin{array}{r|l}
2 & 4\ 0\ 5 \\
\hline
2 & 2\ 0\ 2 \cdots\cdots 1 = k_0 \\
\hline
2 & 1\ 0\ 1 \cdots\cdots 0 = k_1 \\
\hline
2 & 5\ 0 \cdots\cdots 1 = k_2 \\
\hline
2 & 2\ 5 \cdots\cdots 0 = k_3 \\
\hline
2 & 1\ 2 \cdots\cdots 1 = k_4 \\
\hline
2 & 6 \cdots\cdots 0 = k_5 \\
\hline
2 & 3 \cdots\cdots 0 = k_6 \\
\hline
2 & 1 \cdots\cdots 1 = k_7 \\
\hline
 & 0 \cdots\cdots 1 = k_8
\end{array}
\quad \uparrow
$$

最后按从下到上的顺序将余数排列起来即得所求的二进制数，即

$$405 = 110010101_{(2)}.$$

通常称以上方法为“二除取余法”.

例 2 将十进制小数 0.625 化为二进制小数.

解 设 $0.625 = 0.k_{-1}k_{-2}\cdots k_{-m}\cdots\cdots_{(2)}$

$$= k_{-1}2^{-1} + k_{-2}2^{-2} + \cdots + k_{-m}2^{-m} + \cdots\cdots.$$

我们的目的是确定 $k_{-1}, k_{-2}, \cdots$的值.

将等式两端同乘以 2:

$1.25 = k_{-1} + (k_{-2}2^{-1} + \cdots + k_{-m}2^{-m+1} + \cdots\cdots)$。

等式两端整数部分和小数部分必须对应相等，于是 $k_{-1} = 1$，它正好是 0.625 乘以 2 的整数部分.

等式两端划去整数部分后在两端再乘以 2:

$0.5 = k_{-2} + (k_{-3}2^{-1} + \cdots + k_{-m}2^{-m+2} + \cdots\cdots)$.

比较等式两端得 $k_{-2} = 0$，它正好是 0.25 乘以 2 的整数部分.

继续以上过程，并列成竖式:

$$
\begin{array}{l|l}
0.6\ 2\ 5 & \\
\times\quad 2 & \\
\hline
1. & 2\ 5 \cdots\cdots 1 = k_{-1} \\
 & \times\ 2 \\
\hline
0. & 5 \cdots\cdots 0 = k_{-2} \\
 & \times\ 2 \\
\hline
1. & 0 \cdots\cdots 1 = k_{-3}
\end{array}
\quad \downarrow
$$

最后将每一步所得的整数部分按从上到下的顺序排列起来即得所求的二进制小数:

$$0.625=0.101_{(2)}.$$

此方法通常称为"二乘取整法".

在上面的例子中,经有限次乘法就出现小数部分为 0 的情形,从而使乘 2 取整过程终止,但确有乘积的小数部分永不为 0 的情形,例如将十进制小数 0.1 化成二进制小数就会出现无限循环的情形:

$$0.1=0.000110011001100\cdots\cdots_{(2)}.$$

此时可根据精度要求取有限位作为近似值:

$$0.1\approx 0.0001100110011_{(2)}.$$

例 3 将十进制数 405.625 化成二进制数.

解 将十进制数 405.625 的整数部分和小数部分分别按上述法则化成二进制的整数部分和小数部分,然后合在一起,即得所求的二进制数:

$$405.625=110010101.101_{(2)}.$$

读者不难将以上方法推广到任意 p 进制.

例 4 化十进制数 405.625 为八进制数,再化成二进制数.

解 整数部分用"八除取余"得

$$405=625_{(8)}.$$

小数部分用"八乘取整"得

$$0.625=0.5_{(8)}.$$

于是 $405.625=625.5_{(8)}$.

再利用八进制与二进制的转换关系最后得

$$405.625=625.5_{(8)}=110010101.101_{(2)}.$$

由此看出,用八进制作为二进制与十进制的中间过渡是很方便的.