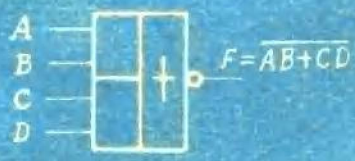
逻辑代数与电子计算机

汪世銘　任　毅　編



安徽省数学学会
芜湖教师进修学院

前　言

随着电子计算机和自动控制技术的日益发展和广泛应用，关于它的数学基础——逻辑代数，已经逐渐成为每一个参加四个现代化建设的普通劳动者必备的知识。因此，教育部决定将这部分内容有计划地逐步纳入中学数学课程中去。与此相适应，全国师范院校数学系（科）普遍增设了这方面的课程，然而缺乏合适的教材。为此，根据教育部去年十一月在天津召开的师专教学工作座谈会精神，参考我省及一些兄弟省市师专关于本课程的教学大纲，在我们教学实践的基础上编写了这本讲义。考虑到各校开设该课程的时间和学时数不尽相同，并且兼顾中学数学教师的自学进修，本讲义在内容安排上基本自成体系。全书可供65—70学时授课使用。如果授课安排52—56学时，可以删节带有*号的章节，这将不会影响前后衔接。

全书共七章。第一章简要地介绍了数的不同进位制的概念以及各种进位制之间的转换方法和理论证明。第二、三、四章分别阐述了集合代数、逻辑代数和开关代数的基本概念和理论。由于三者就公理化结构看，都属于同一种代数系统，在概念和理论上是互相平行的，所以，为了尽量避免重复，在讲述时各有侧重。集合代数部分着重阐明公理系统的建立以及从公理出发推证集合恒等式的方法；逻辑代数部分着重叙述理论的完备性及逻辑函数的代数化简法；开关代数部分着重介绍另一种实际工作者广泛采用的卡诺图化简法及其在逻辑设计中的应用。第五章，在前述各种实际背景的基础上

引入抽象代数结构的概念，进而证明集合代数、逻辑代数和开关代数的同构性。于是，可从理论上认识到三者为什么如此类似的实质。第六章阐述了一般布尔代数的概念和性质，这对于逻辑代数理论的完整性和居高临下地认识逻辑代数的抽象本质都是必要的。第七章简略地介绍了电子计算机的发展情况，基本工作原理和硬件、软件等基本概念。考虑到目前多数师范专科学校尚缺乏上机实习的条件，对计算机的构造和操作方法未作介绍。然而，有了本书对电子计算机概貌的认识，读者完全可以自行阅读计算机的“程序设计”、“算法语言”等书，学会使用计算机。

在编写这本讲义的过程中，安徽大学数学系和芜湖教师进修学院的领导给予我们很大的鼓励和支持。安徽大学数学系郑祖庥副教授、芜湖师专葛尚范副教授对编写工作做了指导并认真地审阅和订正了全部手稿。安徽大学数学系计算机教研室张道胜、黄涛同志、巢湖师专程正权同志、芜湖师专刘为铨同志、蚌埠教师进修学院张振龙同志以及我省各兄弟院校的主讲该课程的同志们提出了宝贵的意见。徐家来同志提供了不少参考资料，程世琼、程立志同志协助选编了部分习题。在出版发行的过程中，杨守昌同志和安大印刷厂给予很多帮助。在此一并表示衷心的感谢。

由于编者水平所限，书中难免有不少错误和缺点，敬请读者批评指正。

编　　者

一九八二年二月

目　录

第一章　进位制

当前，电子计算机的应用是如此广泛，以至于有人称之为电子计算机时代。而要了解计算机就首先要了解它所采用的进位制。事实上，在计算机的设计和使用中涉及了多种进位制，如二进制、八进制、十进制、十六进制、二十进制等等，但最基本的还是二进制。另一方面，由于人们的长期习惯，在生产实践和科学实验中提出的数据通常都用十进制表示的，这就要求我们弄清楚不同的进位制之间的转换关系。本章所要阐述的内容主要是这两个方面。

§1.1　如何选择計算机中的进位制

1.1.1　进位制的概念

“逢十进一，退一当十”是大家都习惯的。流行的说法是，人有十个手指头，屈指计数，自然逢十进一，因此十进制好象是古往今来，天经地义。其实不然，进位制历来就不是单一的。六十秒为一分钟，六十分钟为一小时，却是“逢六十进一”，又如二十四小时为一天，七天为一星期，这些都说明并非所有场合都是“逢十进一”。从历史上看，我国长期用十进制计数，而巴比伦人则用六十进制，还有个别印第安部落使用八进制。实际上，即使对一个民族来说，在它的整个发展过程中进位制也是有所变化，而且多种并用的。

总之，进位制的产生和存在，既取决于历史原因，也取决于使用上的方便。所以，我们应当把视野放宽，以自然的态度来接受不同的进位制，特别是计算机中使用的二进制。

让我们进一步考虑数的表示方式，从中引出记数符号和位值制的概念。先看几个例子：

巴比伦人以Y 表示 1，≺表示10，其他各数则由这两种记号叠排在一起，总共用两个符号，如4876这个数写成

$$\mathrm{Y}\ \prec\prec\mathrm{Y}\quad\prec\begin{matrix}\mathrm{Y}&\mathrm{Y}&\mathrm{Y}\\ \mathrm{Y}&\mathrm{Y}&\mathrm{Y}\end{matrix}$$

(1)　(21)　　(16)

$(4876 = 1\times60^2+21\times60+16)$

罗马人的记数法，则是用分别代表“一、五、十、五十、百、五百、千”的符号“I、V、X、L、C、D、M”，以大数为基础结合右加左减的法则来表示各个数。例如，

Ⅸ表示10－1＝9，XL表示50－10＝40，

XXX表示10＋10＋10＝30，

Ⅵ表示 5 ＋ 1 ＝ 6

凡此等等，总共用七个符号。这里“千、百、十”是用不同的符号来代表的，各个符号所表示的数值，实质上可与符号在数字中所处的位置无关，是一种无位值记数法。这种罗马记数法在13世纪以前流行于欧洲各国，用它表示很大的数字，必须写很长的一串符号，作乘除运算时更不方便。因此，从人类的认识过程来看，合理地表示数字并不是容易的事。

我国古代用算筹计数，算筹的九种不同摆法作为九个不同的基本记数符号，有纵横两种：

一 二 三 四 五 六 七 八 九

纵式：𝍠 𝍡 𝍢 𝍣 𝍤 𝍥 𝍦 𝍧 𝍨

横式：𝍩 𝍪 𝍫 𝍬 𝍭 𝍮 𝍯 𝍰 𝍱

在记数时，个位、百位、万位……采用纵式，十位、千位……采用横式。如1982写成：一𝍧𝍰𝍡，同时，如遇到零就空一位。这种算筹记数制，实际上是一种十进制的有位值计数法。据史书记载，这种筹算法在我国从春秋战国一直用到公元十五世纪左右。这种十进制算筹记数法与巴比伦的六十进制记数法有一个特点，即符号放在不同的位置表示不同的位值。十进制分别是个位、十位、百位……，而六十进制则为个位、六十位、六十平方位……。这种记法叫做位值制。记数法有没有位值制是一个飞跃，古埃及人是不知道用位值制的，欧洲迟到十三世纪前后还用无位值制的罗马记数法。而我国却是最早确立位值制的国家。

综上所述，要表示数和它的运算，应该有三个因素，即记数符号（又叫数码），进位制和位值制。对下面讨论的进位制总假定这三点都具备。

1.1.2 r进制

对十进制，我们采用阿拉伯数码0、1、2、……、9，并应用位值的概念来表示数。如

$1982 = 1 \times 10^3 + 9 \times 10^2 + 8 \times 10^1 + 2 \times 10^0$

$0.1416 = 1 \times 10^{-1} + 4 \times 10^{-2} + 1 \times 10^{-3} + 6 \times 10^{-4}$

$332.157 = 3 \times 10^2 + 3 \times 10^1 + 2 \times 10^0 + 1 \times 10^{-1} + 5 \times 10^{-2} + 7 \times 10^{-3}$

用这种方式，任何一个十进制数S（不妨假设它是正的）都可以表示为：

$$S = x_n x_{n-1} \cdots x_1 x_0 . x_{-1} x_{-2} + x_{-(m-1)} x_{-m}$$
$$= x_n(10)^n + x_{n-1}(10)^{n-1} + \cdots + x_1(10)^1 \cdots x_0(10)^0 + x_{-1}(10)^{-1} + x_{-2}(10)^{-2} + \cdots + x_{-m}(10)^{-m}$$
$$= \sum_{i=n}^{-m} x_i(10)^i$$

为方便计，上式的求和记号从正整数n到负整数－m（下同），而x_i可以是0，1，2，3，4，5，6，7，8，9十个数码中的任何一个，它由S决定；m，n为正整数；10表示该记数制所使用的全部数码的个数叫做基数或底，基数10的各次幂恰是数的各个数位的位权（又叫位值或位数）：

$10^0 = 1$（个位）　$10^{-1} = \frac{1}{10} = 0.1$（十分位）

$10^1 = 10$（十位）　$10^{-2} = \frac{1}{100} = 0.01$（百分位）

$10^2 = 100$（百位）　$10^{-3} = \frac{1}{1000} = 0.001$(千分位)

$10^3 = 1000$（千位）　$10^{-4} = \frac{1}{10000} = 0.0001$（万分位）

……　　……

一般地，任何一个大于或等于2的整数r都可以作为进位数制的基数，与十进制类似，任何一个r进制数S都可以写成：

$$(S)_r = (x_n x_{n-1} \cdots x_1 x_0 . x_{-1} x_{-2} \cdots x_{-(m-1)} x_{-m})_r$$
$$= x_n r^n + x_{n-1} r^{n-1} + \cdots + x_1 r^1 + x_0 r^0 + x_{-1} r^{-1} + \cdots + x_{-m} r^{-m}$$
$$= \sum_{i=n}^{-m} x_i r^i$$

这里x_i取r个基本符号0，1，…，r－1中的任何一个，记号$(\ )_r$表示括号里的是r进制数，r＝10时，也可以不加标注；特别地，r＝2时，x_i仅能取数码0或1，任何一个二进制数S可以表示为：

$$(S)_2=(x_nx_{n-1}\cdots x_1x_0.x_{-1}x_{-2}\cdots x_{-m})_2$$

$$=\sum_{i=n}^{-m}x_i(2)^i$$

例如：$(101011)_2=1\times2^5+0\times2^4+1\times2^3+0\times2^2+$
$+1\times2^1+1\times2^0$

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

$(101.011)_2=1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+$
$+1\times2^{-2}+1\times2^{-3}$

注意：$(\underbrace{10\cdots0}_{n个0})_2=2^n$，即二进制数整数部分从右往左数第n＋1位的位权是$2^n$。

$(0.\underbrace{0\cdots0}_{m个0}1)_2=2^{-m}$，即二进制数小数部分从左往右数第m位（不包括小数点前整数部分的一个零）的位权是2^{-m}。

上述公式$(S)_r=\sum_{i=n}^{-m}x_ir^i$叫做r进制数S的按权展开式，这是一个十分重要而又常用的公式。由此公式容易知道r进制数S的特点可归结为：

（i）只有r个数码。因而各个x_i只能在0，1，2，…，r－1这r个数码中间取值；

（ii）每一位数码x_i表示的数值是x_ir^i，r^i即数码x_i所在数位的位权或位值，x_i叫做r^i的系数；

（iii）“逢r进一，退一当r”是r 进制的进退位法则，它表明r进制数中相邻两位高位的一个单位等于低位的r个单位。（此时基数为r。）

尚需指出两点：

第一，当基数r＞10时应当加选数码。例如r＝12，我们这里选用的数码是0，1，2，3，4，5，6，7，8，9，$\overline{0}$，$\overline{1}$，也有的书采用字母t代替$\overline{0}$，e代替$\overline{1}$。

第二，r进制数各位的名称。例如，$(1101.101)_2$的各个数位的叫法可以是：

1	1	0	1	•	1	0	1
2^3位	2^2位	2^1位	2^0位		2^{-1}位	2^{-2}位	2^{-3}位

一般地，$(S)_r=\sum_{i=n}^{m}x_ir^i$中数码$x_i$所在的数位叫做r的i次幂位。

1.1.3 二进制与电子计算机

电子计算机要进行大量的数据运算，选择什么样的记数法来表示数字对计算机的构造和性能都有很大的影响。大多数计算机之所以采用二进制，这是因为：

第一，易于实现。制造具有两个稳定物理状态的元件，如电子元件的开和关，比制造具有三个稳定物理状态或十个状态的元件要简单得多。

第二，节省设备。假定表示一种状态需要一个元件，那么采用十进制表示0到999这一千个数需要用三位，每位十种状态，总共需要30个元件；而采用二进制表示0到1023这一千零二十四个数需要十位，每位两种状态，总共只要20个元件。后面我们将证明，基数为3时最省设备。但是，目前

制造具有三个稳定状态的元件不但难度大，而且可靠性差，所以通常计算机还是采用二进制。

第三，运算简单。任何一种进位制的算术运算都必须以相应的加法表、乘法表为基础。在§1.2中，我们将会看到二进制的加法表、乘法表比十进制的简单得多。实际上，二进制的加法表、乘法表比任何其它进位制的都简单。

第四，还可用来进行多种逻辑运算。这是因为二进制的两个数码“1”和“0”恰可用来表示二值逻辑中命题的两个值。

应当指出，二进制虽有上述优点，但也不是完美无缺的，把一个较大的数1023表示成二进制数是1111111111，共十位，读写和观察都不方便。还有，实际数据都是十进制的，计算机又只能对二进制数进行运算，因此在把数据输入和输出机器的过程中，要进行两次数据转换，占用不少机器时间。所以，根据不同的需要，也有不少专用数据处理机是采用十进制的。在实际工作中，还有采用八进制或十六进制的。

1.1.4 在计算机中使用三进制最省设备的证明。

设m位R进制数所能表示的最大数是N。例如，四位二进制数所能表示的最大数是15，九位二进制数所能表示的最大数是（111111111）$_2$ = 511。那么，所需物理状态的个数 $x = m \cdot R$，具有

$$N + 1 = R^m$$

两边取对数

$$\ln(N + 1) = m\ln R = \frac{x}{R}\ln R$$

解得

$$x=\frac{R\ln(N+1)}{\ln R}$$

欲求x 的最小值，只须令$\frac{dx}{dR}=0$，

$$\frac{dx}{dR}=\frac{\ln(N+1)}{\ln R}+R\left(\frac{\ln(N+1)}{\ln R}\right)'$$

$$=\frac{\ln(N+1)}{\ln R}+R\,\frac{-\ln(N+1)}{(\ln R)^2}\cdot\frac{1}{R}$$

$$=\frac{\ln(N+1)}{\ln R}-\frac{\ln(N+1)}{\ln^2 R}=0$$

从而求得R = e时，x达最小值。但是，进位制的基数只能是整数，而最接近于e的整数是3，所以说采用三进制最省设备。例如，$(10000000)_3=3^7=2187$，使用 7 × 3 = 21 个元件在三进制中可以表示出从 0 到2186间共计二千一百八十七个数，而 $(100000000000)_2=2^{11}=2048$，使用 11 × 2 = 22 个元件在二进制中也只能表示从 0 到2047间共计二千零四十八个数。

习 题 一

1．分别写出二进制，五进制，八进制的前十二个整数（从 0 开始，由小往大写）。

2．在□中写入“1 或 0”使下列不等式成立：

（1）110□□>11000，（2）100□□>10010，

（3）10□□0>10100，（4）1□01□>11010。

3．比较下列各组二进制数的大小：

（1）101与110，(2)110与1000，（3）11001与11010，

（4）1001与1100。

4．当 P = 2，8 时，分别计算 (1011.011)$_P$ 所对应的十进制数。

5．写出下面各数的后一整数：

(1) $(222)_3$，(2) $(110111)_2$，(3) $(4344)_5$，

(4) $(57\overline{0}\,\overline{0}\,\overline{1})_{12}$。

6．用二进制数表示一个三位长的十进制数最少需要几位？最多需要几位？

7．十二位长的二进制整数的表示范围是多大？（提示：这里既没有规定最高位非零，又没有说明该数可否为负数。）

§1.2 二进制数的运算

二进制数的四则运算与十进制数相类似，但它只有 0，1 两个数码，实际运算起来比十进制数简单得多。

1.2.1 加法

加法表是：

+	0	1
0	0	1
1	1	10

特点：逢二进一。

例 1．　1011 + 111 + 101 + 11 = ?

解：

$$\begin{array}{r} 1011 \\ 111 \\ 101 \\ +\quad 11 \\ \hline 11010 \end{array}$$

∴　1011 + 111 + 101 + 11 = 11010

对于初学者，验算常常是必须的，其办法是作相应十进制数的运算。如例 1 中，由1011，111，101，11的相应十进制数11，7，5，3 相加，得和26，恰为与11010 相应的十进制数，故知运算结果是正确的。

由加法表的对称性可知，与十进制数的加法一样，两个或多个二进制数相加，交换律和结合律也是成立的。

1.2.2 减法

减法是加法的逆运算，从加法表的和减去一个加数，便得另一个加数，不过应注意的是遇到不够减时要向相邻高位去借。二进制减法的借位特点是：退一当二。

例 2． 1011.101 − 10.11 = ?

以下为帮助初学者验算，在写每例的竖式解法时，把相应十进制数并列地写在该二进制数的右边。

解：
$$\begin{array}{rl} & 1011.101\cdots 11\frac{5}{8} \\ - & \underline{10.11\ \cdots 2\frac{3}{4}} \\ & 1000.111\cdots 8\frac{7}{8} \end{array}$$

∴ 1011.101 − 10.11 = 1000.111

电子计算机中为避免另做减法线路，常采用求补数的办法化减法为加法来做。

补数概念可看成来源于下列形式的十进制简便运算：

12 − 7 = 12 + 3 − 10 = 5，

127 − 74 = 127 + 26 − 100 = 53，

1025 − 787 = 1025 + 213 − 1000 = 238，

12 − 7.8 = 12 + 2.2 − 10 = 4.2，

0.63 − 0.47 = 0.63 + 0.53 − 1.00 = 0.16

这里，10 − 7 = 3，100 − 74 = 26，1000 − 787 = 213．10 −

7.8＝2.2，1.00－0.47＝0.53依次是7，74，787，7.8，0.47的补数。由这些例子不难概括出下列一般性结论：

（1）任何一个十进制整数或带小数x之补数是满足条件：$x+y=10^n$（n为自然数）的整数或带小数y。

（2）任何一个十进制纯小数x的补数是满足条件：$x+y=10^0=1$的纯小数y。

总之，任何一个十进制数x的补数y必满足条件：$x+y=10^n$（n为非负整数，恰等于x中非零整数部分的位数），则称y为x的补数。注意这里的n与y的非零整数部分的位数未必相同。

例3．求二进制数10100，11.01，0.101的补数。

解：$(10)_2^5-(10100)_2=(1100)_2$，$(10)_2^2-(11.01)_2=(0.11)_2$，$(10)_2^0-(0.101)_2=(0.011)_2$，故所求的补数依次为：

1100，0.11，0.011

由此例可总结出求二进制补数的法则：

（1）从右向左看：如果原数第一位数码是“1”，则补数的此位数码仍为“1”，以后补数的各位数码可由原数数码逢“1”变“0”，逢“0”变“1”得到。

（2）从右向左看：如原数第一位数码是“0”，则第一个出现的“1”数码连同其前所有“0”都和补数数码一样，而后补数的各位数码，仍由原数数码“0”变“1”，“1”变“0”得到。

例4．求下列各二进制数的补数：

100，111，10.01，110.1，0.001

解：所求的补数依次是：

100，1，1.11，1.1，0.111

例5． 10101－110＝？

解：

$$\begin{array}{rrr} 10101 & 10111 & 11111 \\ +\quad 10 & +\ 1000 & -10000 \\ \hline 10111 & 11111 & 1111 \end{array}$$

即 $10101-110=10101+10-1000$

$=10111+1000-10000=1111$

注意，本例连续两次采用了“加补”“减整”的方法。

与十进制数的减法一样，二进制数的减法既是不可交换的，又是不可结合的。

1.2.3 乘法

乘法表是：

×	0	1
0	0	0
1	0	1

这比十进制的乘法九九表简单得多。

例6． $1101.1\times1.01=?$

解：

$$\begin{array}{r} 1101.1\cdots13\frac{1}{2} \\ \times\quad 1.01\cdots1\frac{1}{4} \\ \hline 11011 \\ 00000 \\ +11011 \\ \hline 10000.111\cdots16\frac{7}{8} \end{array}$$

$\therefore\quad 1101.1\times1.01=10000.111$

从竖式演算可见，它实际上是用加法和移位这两种手续实现的。与十进制一样，乘法的交换律，结合律，乘法对加法的分配律，在二进制数的乘法中也是成立的。