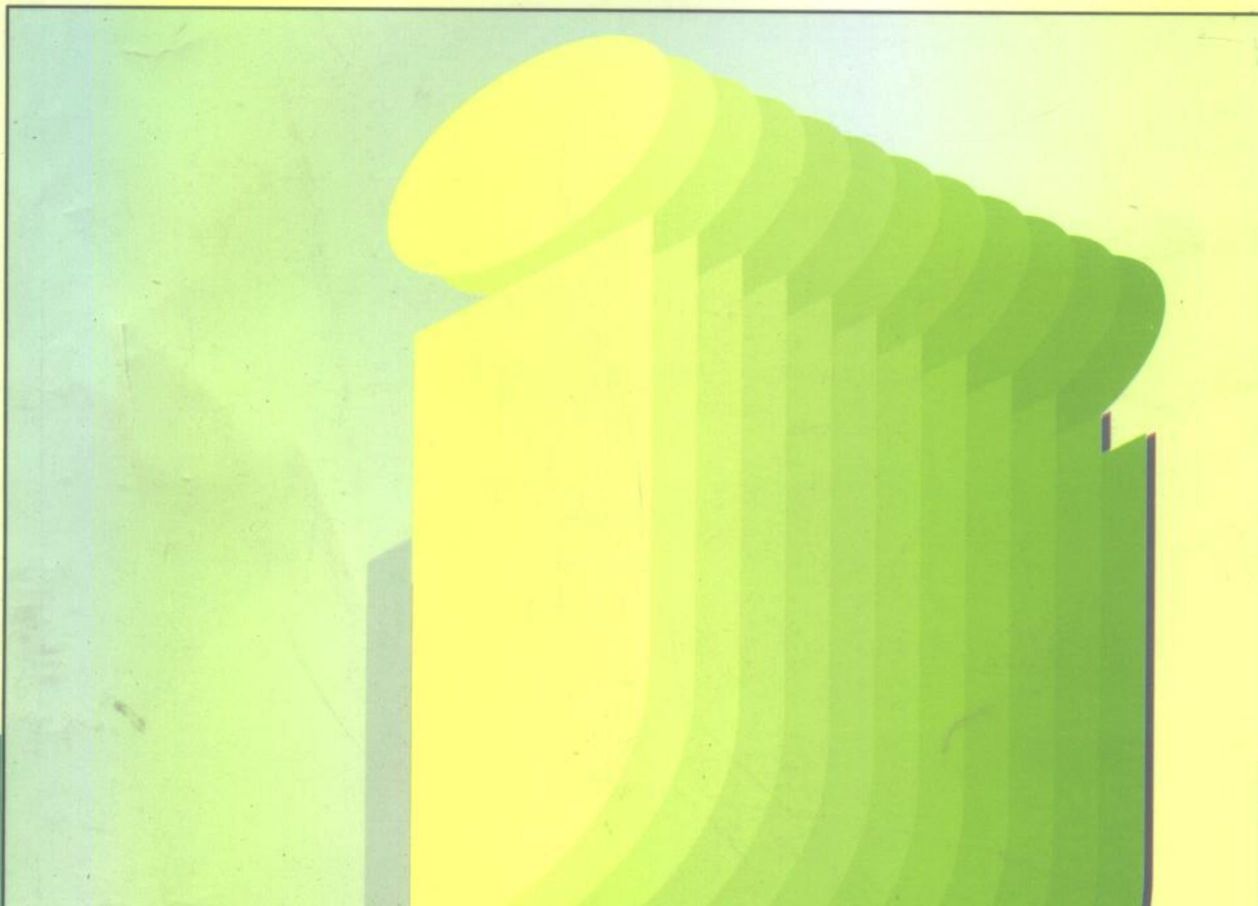
机电一体化系列教材

数字电子技术

梁淼　陈在壎　编



机械工业出版社

机电一体化系列教材

数字电子技术

梁　森　陈在壗　编

机械工业出版社

IA16115

本书主要内容有数字电路基础、逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、脉冲信号的产生和整形、模拟量和数字量的转换、数字电路综合应用。本书既重视外部逻辑功能的分析，又重视基本单元电路工作机理和负载能力的研究，通过对典型电路的分析，提高分析和设计数字系统的能力。

图书在版编目（CIP）数据

数字电子技术/梁森，陈在壛编．—北京：机械工业出版社，1998.2

机电一体化系列教材

ISBN 7-111-06018-0

Ⅰ．数…　Ⅱ．①梁…　②陈…　Ⅲ．数字电子-教材　Ⅳ．TN711.5

中国版本图书馆 CIP 数据核字（97）第 25535 号

出 版 人：马九荣（北京市百万庄大街 22 号　邮政编码 100037）

责任编辑：孙本绪　版式设计：王　颖　责任校对：魏俊云

封面设计：姚　毅　责任印制：王国光

机械工业出版社京丰印刷厂印刷·新华书店北京发行所发行

1998 年 7 月第 1 版第 1 次印刷

787mm×1092mm 1/16·13 印张·310 千字

0 001—3 000 册

定价：21.00 元

前　　言

在现代工业飞速发展的今天，利用电子技术生产的家用电器和现代化通信设备如电视、录像机、传真机、卫星通信设备等，琳琅满目。在工业控制方面，采用电子技术制作的传感器、测量仪表、控制器和驱动装置使系统更加灵敏、精确，从而有效地提高了自动控制系统的质量。采用大规模、超大规模集成电路工艺生产的微型计算机，正以前所未有的速度在各个领域中得到了广泛的应用。为了适应科学技术的发展和教学改革的形势，在电工学教学中，我们深感需要一套把电工技术、电子技术、机械工程、计算机和自动控制等高新技术有机结合起来的机电一体化系列教材。《数字电子技术》就是在这一思想的指导下，根据国家教委批准的《高等工业学校数字电子技术基础课程教学基本要求》，并结合多年来的教学实践编写的。

在编写时，力求内容紧密结合教学要求，突出基本概念、基本原理和基本分析方法。本教材除了包括数字电路的基础知识外，还以中小规模集成电路为主来组织内容。在淡化数字功能部件内部结构的前提下，既重视外部逻辑功能的分析，又重视基本单元电路工作机理和负载能力的研究。面对诸多集成电路产品，不求面面俱到，而是通过对典型电路的分析，提高学生分析和设计简单数字系统的能力。根据数字电子技术的发展，加强CMOS电路和中、大规模集成电路的内容，并且适当地拓宽了数字电路的内容，以适应机电一体化的需要。

为了帮助读者理解书中的主要内容，各章中均附有大量与内容配合的习题与思考题，还有部分提高和拓宽知识面的习题。

本书第一、二、三章由梁淼编写，第四、五、六、七、八章由陈在编墉写。梁淼任主编，负责全书的修改与统稿。本书由北京理工大学赵金声教授主审，赵金声教授认真审阅了全部书稿，对全书各章中存在的问题提出了很好的意见和建议，编者谨此致以由衷的感谢。

在编写本书时，得到了北京理工大学电工教研室老师们的关心和支持，在此对他们表示衷心的感谢。

限于编者水平，书中缺点和错误在所难免，尚盼读者批评指正。

编　者

1997年6月于北京

目　　录

第 1 章　数字电路基础

1.1　概述

在电子技术中传递、加工和处理的信号，通常可以分为两大类：一类信号是模拟信号，所谓模拟信号，是指该信号无论从时间上还是从大小上看其变化都是连续的，处理模拟信号的电子电路叫做模拟电路；另一类信号是数字信号，所谓数字信号，是指该信号无论从时间上还是从大小上看其变化都是离散的，或者说是不连续的，处理数字信号的电子电路叫做数字电路。

数字电路是电子计算机和各种数字测量、数字控制技术的基础，是电子技术的重要组成部分。由于数字电路的信号及工作方式与模拟电路不同，因而数字电路的组成、工作特点及分析方法将与模拟电路有很大的差别。与模拟电路相比，数字电路具有以下一些特点：

a. 数字电路的基本工作信号是二进制的数字信号，只有 0 和 1 两个基本数字，反映在电路上就是低电平和高电平两种状态。因此，凡是具有两个稳定状态的元件，其状态都可以用来表示二进制的两个数码，故其基本单元电路简单，对实现电路的集成化十分有利。

b. 由于数字电路传递、加工和处理的是二进制信号，因此在稳态时，数字电路中的半导体器件都工作在开、关状态。而研究数字电路时关心的仅是输入、输出之间的逻辑关系。

c. 数字电路不仅能进行数值运算，而且能进行逻辑判断和逻辑运算，这在控制系统中是不可缺少的，因此，也常把数字电路称为“数字逻辑电路”。

d. 数字电路工作可靠，精度较高，并且具有较强的抗干扰能力。数字信号便于长期储存，可使大量可贵的信息资源得以妥善保存，保密性好，使用方便，通用性强。

由于数字电路具有上述特点，其发展十分迅速。但是，数字电路也有一定的局限性。与此同时，模拟电路也有其优于数字电路的一些特点。因此，实际的电子系统往往是数字电路和模拟电路相结合。

1.2　数制和编码

数制就是选定某种进位计数制来表示某个数的值。同一个数可以采用不同的进位计数制来计量，日常生活中，人们习惯于使用十进位计数制，而在数字电路中常采用二进位计数制和十六进位计数制。另外，在数字系统中，常需将有特定意义的数字或字符用若干位二进制数来表示，这种用多位二进制数表示具有一定意义的数字或字符的过程称为编码。本节将介绍几种常用的数制和编码。

1.2.1　常用计数制

一种进位计数制包含着两个基本因素：

a. 基数　它是计数制中每一位数所用到的数码的个数，一般地说，基数为 N 的计数制

中，就包含0、1、…、N_{n-1}个数码，进位规律是“逢N进一。”

b. 位权　在一个进位计数制表示的数中，当一个数码处在不同的数位时所代表的数值就不同。某一个数位的数值是由这一位数码的值乘上处在这位的一个固定常数，这个固定常数称为位权值，简称位权。不同的数位有不同的位权。例如，十进制数536可表示为

$$536=5\times10^2+3\times10^1+6\times10^0$$

其中　10^2、10^1、10^0分别为百位、十位、个位的“位权”，也就是相应数位的1所代表的实际数值。由此可见，位数越高，位权值越重。相邻高位权值是相邻低位权值的10倍。

1. 十进制（Decimal）

十进制是用0、1、2、3、4、5、6、7、8、9十个数码的不同组合来表示一个数，并且自左向右由高位到低位排列。

十进制数的特点是：

a. 十进制数的基数是10。

b. 十进制数的位权是10的幂，即10^i。也就是说任何一个十进制数都可以用其幂的形式来表示，例如：

$$125.68=1\times10^2+2\times10^1+5\times10^0+6\times10^{-1}+8\times10^{-2}$$

c. 低位数和相邻的高位数之间的进位关系是“逢十进一”。

显然，任意一个十进制数M可以表示为

$$(M)_{10}=\sum_{i=-\infty}^{\infty}K_i\times10^i$$

其中　K_i为基数10的i次幂的系数，它可以是0～9中的任一个数码。i可为$-\infty$到$+\infty$之间的任意整数。

2. 二进制（Binary）

二进制数的数值表示符号只有0和1，它同十进制数一样，自左到右由高位到低位排列。

二进制数的特点是：

a. 二进制数的基数是2。

b. 二进制数的位权是2的幂，相邻高位是相邻低位权值的2倍。

c. 低位和相邻高位之间的进位关系是“逢二进一”，借位规则是“借1当2”。

同十进制数一样，每个数码处在不同的数位代表不同数值。例如，二进制数1101.101所代表的数值是

$$(1101.101)_2=1\times2^3+1\times2^2+0\times2^1+1\times2^0+$$
$$1\times2^{-1}+0\times2^{-2}+1\times2^{-3}=(13.625)_{10}$$

显然，任意一个二进制数M可以表示为

$$(M)_2=\sum_{i=-\infty}^{\infty}K_i\times2^i$$

其中　K_i为基数2的i次幂的系数，它可以是1，也可以是0，i为$-\infty$到$+\infty$之间的任意整数。

3. 十六进制（Hexadecimal）

由于二进制数比十进制数位数多，不便于书写和记忆，因此在计算机应用中，经常用十六进制数来表示。

十六进制数有 0、1、2、3、4、5、6、7、8、9、A (10)、B (11)、C (12)、D (13)、E (14)、F (15) 十六个数字符号，它也是由左向右从高位到低位排列。

十六进制数的特点是：

a. 十六进制数的基数是 16。

b. 十六进制数的位权是 16 的幂，相邻高位是相邻低位权值的 16 倍。

c. 低位和相邻高位之间的进位关系是“逢十六进一”，借位规则是“借 1 当 16”。

每一个数字处在不同数位代表不同的数值，例如十六进制数 5AE 可表示成

$$(5AE)_{16}=5\times16^2+10\times16^1+14\times16^0=(1454)_{10}$$

任意十六进制数可表示为

$$(M)_{16}=\sum_{i=-\infty}^{\infty}K_i\times16^i$$

其中 K_i 为基数 16 的 i 次幂的系数，它可以是 0～F 十六个数码中的任一个。i 可为 $-\infty$ 到 $+\infty$ 之间的任意整数。

书写计算机程序时，有时也使用八进制。在八进制数中，有 0～7 八个数码，计数的基数为 8，低位和相邻高位间的关系是“逢八进一”，各位数的权是 8 的幂。

1.2.2 不同数制之间的相互转换

1. 二进制和其它进制转换成十进制

由二进制、八进制和十六进制数的一般表达式可知，只要将它们按权展开，求各位数值之和，就可得到相应的十进制数。这种方法通常称为“按权相加”法。

例 1 将二进制数 11010.101 转换成十进制数。

解 用按权相加法，可得

$$(11010.101)_2=1\times2^4+1\times2^3+0\times2^2+1\times2^1+0\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}=$$

$$16+8+0+2+0+0.5+0+0.125=(26.625)_{10}$$

例 2 将八进制数 403 转换成十进制数。

解 同样用按权相加法，可得

$$(403)_8=4\times8^2+0\times8^1+3\times8^0=256+0+3=(259)_{10}$$

例 3 将十六进制数 7A.58 转换成十进制数。

解 用按权相加法

$$(7A.58)_{16}=7\times16^1+10\times16^0+5\times16^{-1}+8\times16^{-2}=$$

$$112+10+0.3125+0.03125=(122.34375)_{10}$$

2. 十进制转换成二进制

一个具有整数部分和小数部分的十进制数转换为二进制数时，应当分别将其整数部分和小数部分转换为二进制数，然后用小数点将两部分连接起来。

将十进制数的整数部分转换成二进制数的基本方法是“除 2 取余”法，即用 2 不断去除

十进制数，直至商为零时止，所得余数由下向上的顺序读取，即为所求的等值二进制数。

例 4 将十进制数 13 转换成二进制数。

解 按“除 2 取余”法，可用竖式除法表示出这个转换过程

$$
\begin{array}{r|l}
2 & \underline{13} \\
2 & \underline{6} \quad \cdots \quad 余 1 \quad \cdots \quad K_0 \\
2 & \underline{3} \quad \cdots \quad 余 0 \quad \cdots \quad K_1 \\
2 & \underline{1} \quad \cdots \quad 余 1 \quad \cdots \quad K_2 \\
  & 0 \quad \cdots \quad 余 1 \quad \cdots \quad K_3
\end{array}
\quad \uparrow \text{读取顺序}
$$

所以 $(13)_{10}=(K_3K_2K_1K_0)_2=(1101)_2$

任何十进制数的纯小数部分可用“乘 2 取整”的方法，求出相应的二进制数。

十进制纯小数转换为二进制小数的步骤如下：

a. 将给定的十进制纯小数乘以 2，其积的整数部分便是等值二进制纯小数的最高位。

b. 将上一步乘积的小数部分再乘以 2，所得乘积的整数部分便是次高位。

c. 重复步骤 b，直至乘积的小数部分为 0，或者满足误差要求进行“四舍五入”为止，各次乘积的整数部分依次为 K_{-1}、K_{-2}、…、K_{-n}，则 0、$K_{-1}K_{-2}\cdots K_{-n}$即为十进制小数的二进制表达式。

例 5 将纯小数 $(0.913)_{10}$转换成误差 ε 不大于 2^{-5}的二进制纯小数。

解 用“乘 2 取整”法，可求出相应的二进制纯小数：

$$
\begin{array}{ll}
 & \text{取整} \\
0.913\times2=1.826\cdots1 & \text{即 } K_{-1} \\
0.826\times2=1.652\cdots1 & \text{即 } K_{-2} \\
0.652\times2=1.304\cdots1 & \text{即 } K_{-3} \\
0.304\times2=0.608\cdots0 & \text{即 } K_{-4}
\end{array}
$$

由于最后余的小数 $0.608>0.5$ 则根据“四舍五入”的原则，可得 $K_{-5}=1$。因此

$$(0.913)_{10}=(0.11101)_2$$

且其误差 $\varepsilon<2^{-5}$，

例 6 将十进制数 13.913 转换成二进制数。

解 可由

$$(13)_{10}=(1101)_2$$

$$(0.913)_{10}=(0.11101)_2$$

得

$$(13.913)_{10}=(1101.11101)_2$$

3. 八进制、十六进制和二进制的相互转换

a. 八进制和二进制的相互转换　由于八进制基数为 8，而 $8=2^3$，因此，三位二进制数就相当于一位八进制数。若要将多位二进制数转换成八进制数，则其整数部分可由小数点起向左推，三位成一组，直至分到最高位为止，若此时不足三位，可在高位补零，其小数部分可由小数点起向右推，三位成一组，直至分到最低位，若此时不足三位，可在低位补零。完成上述分组后，将每组的三位二进制数转换成一位八进制数，就完成了二进制数到八进制数的

转换。

例 7 将二进制数 $(10011101.01)_2$ 转换成八进制数。

解 将 $(10011101.01)_2$ 化成分组形式：$(010\ 011\ 101.010)_2$，然后将各组三位二进制数转换为八进制数，得

$$(235.2)_8$$

八进制转换成二进制也是很容易的。由于每位八进制数相应于三位二进制数，因此，任意八进制数均可由各位变成三位二进制数而得到相应的二进制形式。

例 8 将八进制数 $(6403.1)_8$ 转换成二进制数。

解 可分别将 6、4、0、3、1 化成相应的二进制数，按位的高低依次排列，就可得到相应的二进制数：

$$(6403.1)_8 = (110\ 100\ 000\ 011.001)_2$$

b. 十六进制和二进制的相互转换　由于十六进制基数为 16，而 $16=2^4$，因此，四位二进制数就相当于一位十六进制数。按照八进制和二进制之间的转换步骤，只要将二进制数按四位分组，即可实现它们之间的转换。

例 9 将二进制数 $(111100.101101)_2$ 转换成十六进制数。

解 首先将 $(111100.101101)_2$ 化成分组形式：

$$(0011\ 1100.1011\ 0100)_2$$

然后将各组四位二进制数转换为十六进制数，得

$$(3C.B4)_{16}$$

例 10 将十六进制数 $(AF6.65)_{16}$ 转换成二进制数。

解 可分别将 A、F、6、6、5 化为相应的二进制数，按位的高低依次排列，就得到相应的二进制数：

$$(AF6.65)_{16} = (1010\ 1111\ 0110.0110\ 0101)_2$$

c. 八进制和十六进制的相互转换　八进制和十六进制之间的相互转换，可以把二进制作为桥梁，即先由八进制转换为二进制，再由二进制转换为十六进制。若是由十六进制转换为八进制也是如此。

另外，任意进制与十进制之间的转换原理及方法，同二进制与十进制之间的转换原理及方法相类似，不再重复。

而任意两种进制之间的转换，也可以把十进制作为桥梁，即先由一种进制转换为十进制，再由十进制转换为另一种进制。

1.2.3 编码

数字系统处理的信息，一类是数值，另一类是文字和符号，它们都可以用多位二进制数来表示，这种多位二进制数叫做代码，给每个代码赋与一定的含义叫编码。

1. 二—十进制（BCD）码

二—十进制码（Binary Coded Decimal Codes）又称 BCD 码。它是用四位二进制数组成一组代码，来表示一位十进制数 0～9。因为四位二进制数共有 $2^4=16$ 种不同的组合可作为代码，而十进制数的 0～9 十个数字符号只需用其中的十种组合来表示，因而从这十六种组合中选用哪十种组合的编码方案有很多种，其中最常用的是“恒权”二—十进制编码方法。

在这种编码中，如果在四位二进制代码的十六种状态中取出前面十种状态，表示 0～9 十

个数码，后面六种状态去掉，如表 1-1 所示。二进制代码各位的 1 所代表的十进制数值从高位到低位依次为 $2^3=8$、$2^2=4$、$2^1=2$、$2^0=1$，称之为“权”或“位权”。若将四位二进制代码的每个数码乘以各位的“权”，而后相加，即得出该二进制代码所表示的一位十进制数。例如，二进制码 0111 所表示的十进制数为 $8\times0+4\times1+2\times1+1\times1=7$，由于这个四位二进制数码每一位的权是固定的，故称为 8421BCD 恒权代码。恒权代码还有 5421 码和 2421 码等，见表 1-1 所示。

表 1-1　常用的二—十进制编码

编码种类 / 十进制数	8421 码	5421 码	5211 码	2421 码
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0		0 0 1 0
3	0 0 1 1	0 0 1 1		0 0 1 1
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 0 0
5	0 1 0 1		0 1 0 1	
6	0 1 1 0			
7	0 1 1 1		0 1 1 1	
8	1 0 0 0	1 0 0 0	1 0 0 0	
9	1 0 0 1	1 0 0 1	1 0 0 1	
10		1 0 1 0		
11		1 0 1 1		1 0 1 1
12		1 1 0 0	1 1 0 0	1 1 0 0
13			1 1 0 1	1 1 0 1
14				1 1 1 0
15			1 1 1 1	1 1 1 1

注：空位表示无效的组合。

2. 格雷码

格雷码是一种常见的无权码，即每位二进制数都没有固定的权值。格雷码的特点是相邻两个代码之间仅有一位不同，其余各位均相同。在典型的几位格雷码中，0 和最大数 (2^{n-1}) 之间也只有一位不同，故它是一种循环码。格雷码的这个特点使它在代码形成与传输时引起的误差较小。

格雷码是一种无权码，不能直接进行计算。但它很容易转换成二进制码，表 1-2 给出了格雷码与二进制码的关系。

表 1-2 格雷码与二进制码的关系对照表

十进制数	二进制码	格雷码	十进制数	二进制码	格雷码
	$B_3\,B_2\,B_1\,B_0$	$R_3\,R_2\,R_1\,R_0$		$B_3\,B_2\,B_1\,B_0$	$R_3\,R_2\,R_1\,R_0$
0	0 0 0 0	0 0 0 0	8	1 0 0 0	1 1 0 0
1	0 0 0 1	0 0 0 1	9	1 0 0 1	1 1 0 1
2	0 0 1 0	0 0 1 1	10	1 0 1 0	1 1 1 1
3	0 0 1 1	0 0 1 0	11	1 0 1 1	1 1 1 0
4	0 1 0 0	0 1 1 0	12	1 1 0 0	1 0 1 0
5	0 1 0 1	0 1 1 1	13	1 1 0 1	1 0 1 1
6	0 1 1 0	0 1 0 1	14	1 1 1 0	1 0 0 1
7	0 1 1 1	0 1 0 0	15	1 1 1 1	1 0 0 0

1.3 逻辑代数

逻辑代数是英国数学家乔治·布尔（George Boole）于 1847 年在他的著作《逻辑的数学分析》及《思维规律》中，首先阐述了逻辑代数的基本性质和概念，并进行了系统的论述，所以又称为布尔代数。因为它所研究的是两值变量的运算规律，所以又叫两值代数。我们知道在普通代数中，其变量取值可以从$-\infty$到$+\infty$，而逻辑代数中变量的取值只能是 0 和 1。这两个值不是数量上的概念，而是表示完全对立的两个方面：1 表示条件具备或事件发生；0 表示条件不具备或事件不发生。也就是说，逻辑运算不是数值运算，而是因果关系（条件和结果的关系）的逻辑判断。在逻辑电路中，常用 0 或 1 表示电位的低或高，脉冲的无或有。

逻辑代数是分析和设计数字系统的数学工具。与普通代数一样，逻辑代数中也是用字符表示自变量和因变量，由自变量的取值而确定的因变量叫逻辑函数。本节主要讨论逻辑代数的基本运算、基本公式以及逻辑函数的化简方法。

1.3.1 3 种基本逻辑运算

逻辑代数中只有 3 种基本运算：逻辑乘（即与逻辑关系）、逻辑加（即或逻辑关系）和逻辑非（即“非”逻辑关系）。

1. 逻辑乘（与逻辑关系）

当决定某一事件的所有条件都具备之后，该事件才会而且一定会发生，这样的因果关系称为与逻辑关系，或称为逻辑乘关系。

图 1-1a 是用来说明与逻辑关系基本概念的电路。图中要发生的事件是灯亮，开关 A、B 的闭合是事件发生的条件。显然，只有开关 A、B 同时闭合，灯 Y 才会亮。此时 Y 和 A、B 之间便存在与逻辑关系。在逻辑电路中常用图 1-1b 的图形符号表示与逻辑。

为了详细描述逻辑关系，常把“条件”和“结果”的各种可能性列成表格对应表示出来。如果假设开关接通和灯亮均用 1 表示；开关断和灯不亮均用 0 表示，则可得到表 1-3。这种用逻辑变量的真正取值反映逻辑关系的表格称为逻辑真值表，简称真值表。真值表的左栏列出条件变量可能出现的全部组合，右栏填人逻辑判断的相应结果。

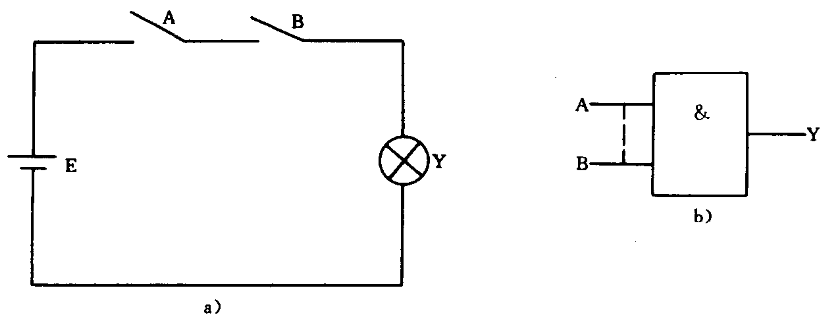


图 1-1 与逻辑关系举例

a) 串联开关电路 b) 与逻辑图形符号

设 A 和 B 为两个逻辑变量，若对 A、B 进行逻辑乘，则其逻辑表达式为

$$Y=A \cdot B$$

表 1-3 与逻辑真值表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

逻辑乘符号“·”可以省略，即上式亦可写为 Y=AB。逻辑乘可用口决“全 1 则 1，有 0 则 0”来协助记忆。

逻辑乘的运算规则如下：

$$0 \cdot 0=0 \qquad 1 \cdot 0=0$$
$$0 \cdot 1=0 \qquad 1 \cdot 1=1$$

在电子技术中，与逻辑关系的例子俯拾皆是，但是把事物抽象为逻辑问题时，必须注意两点：a）只有能明确地用“是”或“否”作出回答的事物才能定义为逻辑变量；b）定义逻辑变量时，必须明确该变量何时取值为 1。

如果用“1”表示条件具备或事件发生（高电平），用“0”表示条件不具备或事件不发生（低电平）称为正逻辑。若用“0”表示高电平，“1”表示低电平则称为负逻辑。对于一个数字电路或系统，既可采用正逻辑，也可采用负逻辑，通常使用正逻辑，在本书中，除特别声明外，都采用正逻辑。

2. 逻辑加（或逻辑关系）

当决定一事件的各个条件中，只要具备一个条件，事件就会发生，这样的关系称为或逻辑关系，或称逻辑加。

在图 1-2a 电路中，如果 Y 和 A、B 的含义同上，当 A、B 中任一闭合，灯 Y 便能亮。此

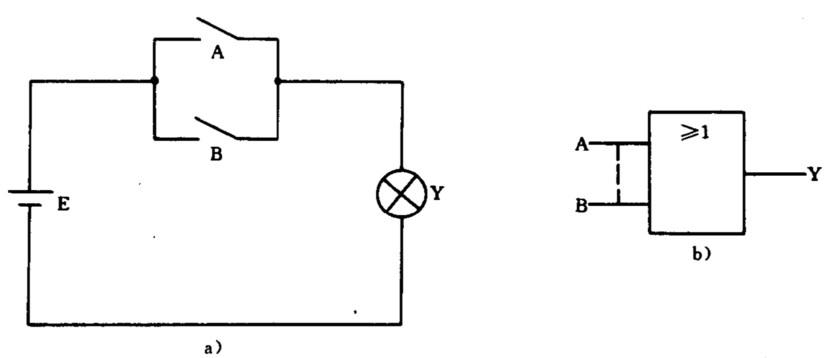


图 1-2 或逻辑关系举例

a) 并联开关电路 b) 或逻辑图形符号

时 Y 和 A、B 之间便存在或逻辑关系。图 1-2b 是或逻辑的图形符号。

按照前述假设，用二值逻辑变量不难列出或逻辑关系的真值表，如表 1-4 所示。

对逻辑变量 A、B 进行逻辑加的逻辑表达式是

$$Y=A+B$$

式中符号“+”表示或。逻辑加可用口诀：“全 0 则 0，有 1 则 1”来协助记忆。

表 1-4　或逻辑真值表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

逻辑加的运算规则如下：

$$0+0=0 \qquad 1+0=1$$
$$0+1=1 \qquad 1+1=1$$

应当指出，在逻辑代数中，逻辑乘的运算顺序优先于逻辑加。例如，在逻辑表达式 $Y=A+B\cdot C$ 中，$B\cdot C$ 先进行。这一点与普通代数类似。

3. 逻辑非（非逻辑关系）

当某一条件满足时，结果却不发生；而这一条件不满足时，结果才会发生。这种因果关系称为逻辑非，也叫非逻辑关系。例如图 1-3a，开关 A 闭合则灯 Y 不亮，A 断开则 Y 亮，便是一例。图 1-3b 是非逻辑的图形符号。

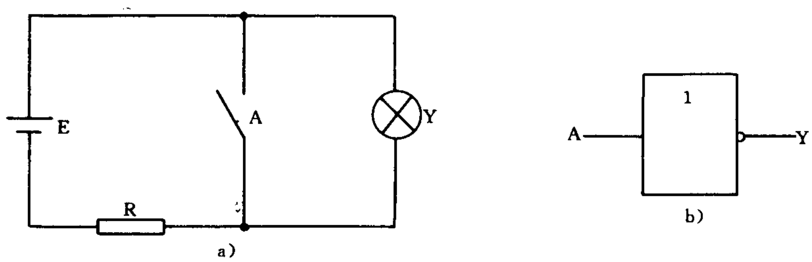


图 1-3　非逻辑关系举例

a）单开关电路　b）非逻辑图形符号

非逻辑的真值表如表 1-5 所示，并可用如下逻辑表达式表示：

$$Y=\overline{A}$$

通常，A 称为原变量，$\overline{A}$ 称为反变量，读作“A 非”或“A 反”。

逻辑非的运算规则如下：

$$\overline{0}=1 \qquad \overline{1}=0$$

在逻辑代数中，逻辑非称为非运算，也称为求反运算。

与、或、非是三种最基本的逻辑关系，任何其它的复杂逻辑关系都可以以它们为基础来表示。表 1-6 是几种常见的复合逻辑关系。

表 1-5　非逻辑真值表

A	Y
0	1
1	0

1.3.2　逻辑代数的基本定律和运算规则

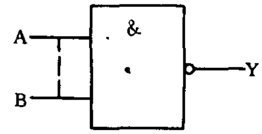
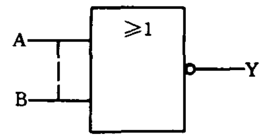
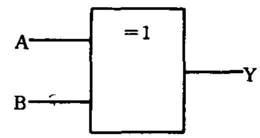
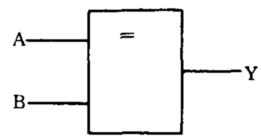
1. 基本定律

根据逻辑变量的特点和与、或、非三种基本逻辑运算关系，在逻辑代数中有以下一些基本定律。

（1）代数定律　逻辑代数中，有的定律与普通代数中的定律相似。

①交换律：　　$A+B=B+A \qquad A\cdot B=B\cdot A$

表 1-6 几种复合逻辑关系

逻辑关系	含义	逻辑表达式	记忆口诀	图形符号
与非	条件 A、B 等都具备，则事件 Y 不发生	$Y=\overline{AB}$	全 1 则 0 有 0 则 1	
或非	条件 A、B 中任一具备，则事件 Y 不发生	$Y=\overline{A+B}$	全 0 则 1 有 1 则 0	
异或	条件 A、B 中只要有一个具备，另一个不具备，则事件 Y 发生	$Y=A\overline{B}+\overline{A}B$ $=A\oplus B$		
同或	条件 A、B 同时具备，或者同时不具备时，则事件 Y 发生	$Y=AB+\overline{A}\,\overline{B}$ $=A\odot B$		

②结合律：　$(A+B)+C=A+(B+C)$　　$(A\cdot B)\cdot C=A\cdot(B\cdot C)$

③分配律：$A\cdot(B+C)=A\cdot B+A\cdot C$　　$A+B\cdot C=(A+B)(A+C)$

(2) 特有定律

①0 1 律：　　$0+A=A$　　$1\cdot A=A$

　　　　　　$1+A=1$　　$0\cdot A=0$

0 1 律反映了变量与常量的关系。

②重叠律：　　$A+A=A$　　$A\cdot A=A$

③互补律：　　$A+\overline{A}=1$　　$A\cdot\overline{A}=0$

④否定律：　　$\overline{\overline{A}}=A$

⑤反演律：　　$\overline{A+B}=\overline{A}\cdot\overline{B}$　　$\overline{A\cdot B}=\overline{A}+\overline{B}$

反演律又称为德·摩根 (De Morgan) 定理，在化简较复杂的逻辑关系时十分有用，并且还可推广到两个以上变量的情况，即

$$\overline{A+B+C+\cdots}=\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\cdots$$

$$\overline{A\cdot B\cdot C\cdot\cdots}=\overline{A}+\overline{B}+\overline{C}+\cdots$$

⑥吸收律：　$A+AB=A$　　$A\cdot(A+B)=A$

$A+\overline{A}B=A+B$　　$A\cdot(\overline{A}+B)=AB$

逻辑代数区别于普通代数的一个重要特点是：逻辑表达式中的某些项（或某些因子）可能被其它项所吸收。吸收律可由基本公式证明如下：

$$A+AB=A \qquad A\cdot(A+B)=A$$

证明：

$$A+AB=A(1+B)=A\cdot 1=A$$

$$A\cdot(A+B)=A+AB=A$$

它说明，在与或表达式中，若一项是另一个乘积项的因子，则该乘积项是可以被吸收的，这种情况常被称作原变量的吸收。

$$A+\overline{A}B=A+B \qquad A\cdot(\overline{A}+B)=AB$$

证明：

$$A+\overline{A}B=(A+\overline{A})\cdot(A+B)=1\cdot(A+B)=A+B$$

$$A\cdot(\overline{A}+B)=A\overline{A}+AB=AB$$

这说明一个与或表达式中，如果一项的反是另一个乘积项的因子，则该因子是可以被吸收的，这种情况又被称作反变量的吸收。

吸收律还可以推演出一些其它公式，它们在逻辑函数的化简中经常用到，如：

$$AB+\overline{A}C+BC=AB+\overline{A}C$$

$$AB+\overline{A}C+BCD=AB+\overline{A}C$$

⑦对合律：　$AB+A\overline{B}=A$　　$(A+B)\cdot(A+\overline{B})=A$

对上述各定律均可列逻辑真值表证明其正确性，即分别列出等式两边逻辑表达式的真值表，若两张真值表完全一致，则表明两个逻辑表达式相等，定律得证。

例如，要证明反演律，首先列出表1-7所示真值表。从表中看到，$\overline{A+B}$与$\overline{A}\cdot\overline{B}$在变量A、B的所有4种取值组合下结果完全一致，因而$\overline{A+B}=\overline{A}\cdot\overline{B}$得证。类似地，也证明了$\overline{A\cdot B}=\overline{A}+\overline{B}$。

表1-7　反演律的真值表

A　B	$\overline{A+B}$	$\overline{A}\cdot\overline{B}$	$\overline{A\cdot B}$	$\overline{A}+\overline{B}$
0　0	1	1	1	1
0　1	0	0	1	1
1　0	0	0	1	1
1　1	0	0	0	0

需要注意的是，学习逻辑代数应当严格地从逻辑变量的二值性和三种基本逻辑运算出发，理解和处理问题。例如，在普通代数中，把等式两边相同的项消去，等式仍成立，但在逻辑代数中则不然。请看下例：可以证明

$$\overline{A}B+A\overline{B}+AB=A+B+AB$$

但是

$$\overline{A}B+A\overline{B}\neq A+B$$

也就是说，不能从等式两边同时消去AB，以得到一个新的等式，这是因为逻辑代数中只有与、或、非三种运算，而不存在什么“减法”、“除法”等运算。

另外，逻辑代数的运算顺序和书写方式有下列规定：

（1）逻辑代数的运算顺序和普通代数一样，应该先算括号里的内容，然后算逻辑乘，最后算逻辑加。