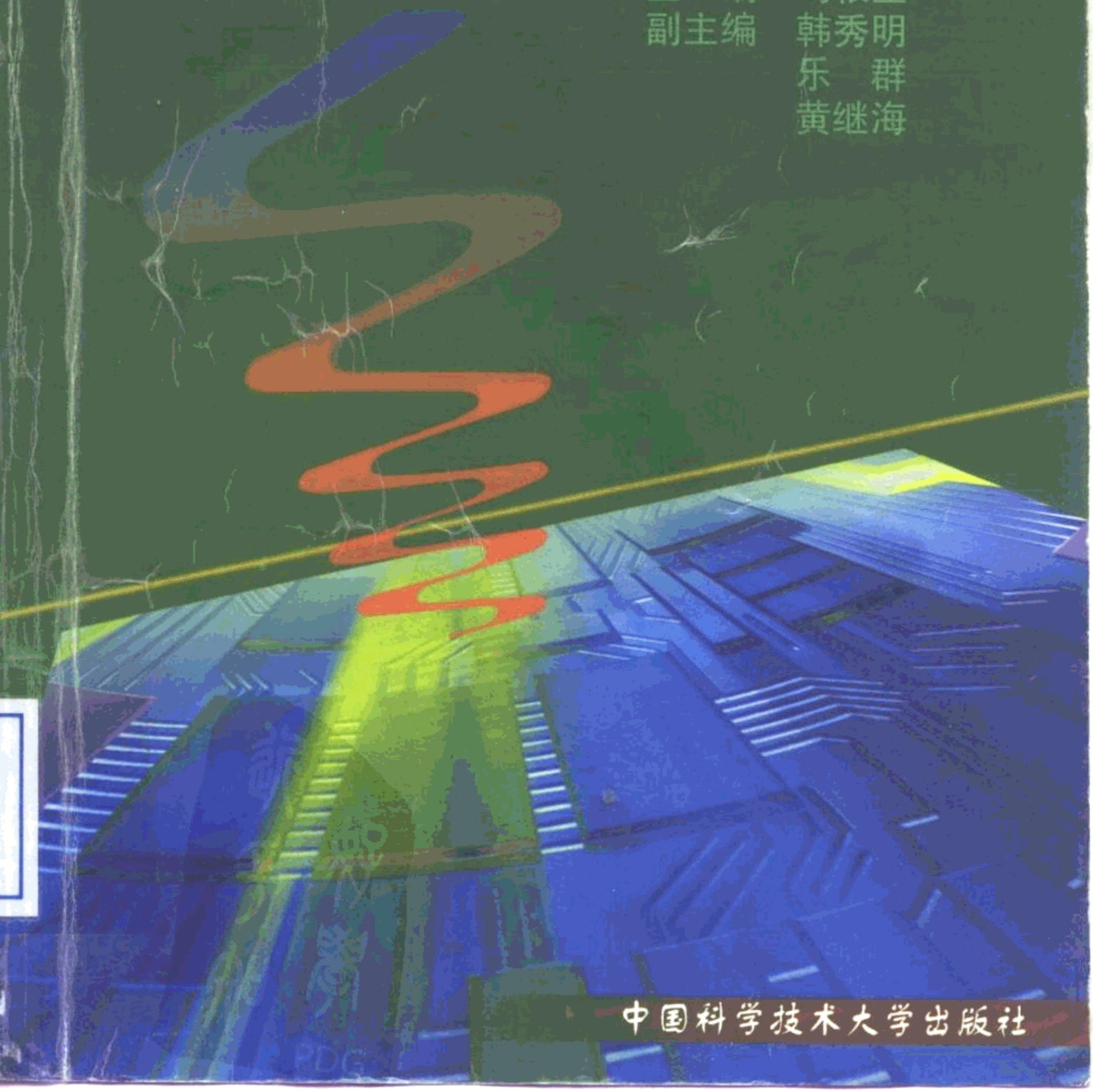
数字电子技术

主　编　冯根生
副主编　韩秀明
乐　群
黄继海



中国科学技术大学出版社

前　言

《电工理论》,《模拟电子技术》,《数字电子技术》是非电专业的重要技术基础课。为了适应我国社会主义现代化建设和高等教育事业的发展,反应现代文化科技的先进水平,贯彻落实军队新时期教育方针,并根据国家教委制定的高等工科院校非电类电工学课程教学的基本要求编写了本套教材。

《数字电子技术》教材的编写过程中,在内容的选取上,我们在保证基本理论完整性的原则下,删去或精减一些分立元件内容,将中小规模逻辑门和触发器作为数字电路的基础器件,并在此基础上重点介绍中规模数字电路的各种逻辑功能部件,并注意介绍门阵列、存储器、A/D与D/A转换器等新技术内容。对问题的阐述上力求循序渐进,通俗易懂,条理清楚,使教材显示出鲜明性和生动性,做到既便于教学又便于能力的培养。

本书由蚌埠坦克学院、合肥炮兵学院发起,武汉二炮指挥学院、郑州高炮学院、徐州空军后勤学院、南京炮兵学院等六所院校联合编写。并于一九九五年十二月在蚌埠坦克学院召开了第一次编委会,经过充分协商,确定了编写教材的指导思想和原则,制定了纲目。成稿后经集体初审,专家审阅,最后在广泛听取各方面的意见的基础上,由主编冯根生同志统审并定稿。

本书由郑州高炮学院何斌副教授担任主审,他仔细审阅了初稿并提出许多宝贵意见。在编写出版的过程中,还得到各兄弟院校的领导,所在教研室的同志的关心、支持与帮助。在此谨表谢意。

由于编者水平有限,难免有不妥和错误之处,敬请读者批评指正。

编者

1999年2月

目　　录

0 绪　论

数字电路是用来处理数字信号的电路。在数字电子系统中,信息用二元数0、1来表示。一个0或一个1通常称为1比特,有的也将一个0或一个1的持续时间称为一拍。对于数字信息的0和1,可以用开关的闭合和断开来表示,也可以用电位的低和高来表示,也可以用脉冲信号的有和无来表示。

数字信号不同于模拟信号,一方面,它们的变化在时间上是不连续的,总是发生在一个瞬间;另一方面,它们表示数值的大小和增减变化,也都是采用数字形式。例如,可以用1101表示数值13,1100表示数值12等。

数字电路的输出和输入信号都是数字信号。通常将数字电路的输出和输入之间的关系称为逻辑关系。通常把完成一些基本逻辑关系的数字电路用逻辑符号表示,由一系列逻辑符号及它们之间的联系所构成的电路图就叫做逻辑图或称为逻辑电路。逻辑电路只反映数字电路或设备的逻辑功能,而不反映电路或设备在电气上的参数、性能等。所以,对数字电路的分析和设计需要考虑两方面的问题,一方面是电气性能方面,另一方面是这类电路的逻辑功能。

由于数字信号简单,只需要用两个不同状态来分别表示0和1即可,在电子线路中常用晶体三极管的饱和和截止两个截然不同的状态来表示。

因此数字电路的基本单元比较简单,而且对元件的要求也不严格,只要能区分0和1就够了。这样就能在一块硅片上把众多的基本单元制作在一起,产生数字集成电路。因而,对数字电路电气性能方面的分析,重点讨论基本数字集成电路(如集成逻辑门、集成触发器)的电气性能。对于较复杂的数字电路,侧重在逻辑功能的分析和设计。所以本书中,有时把数字电路叫逻辑电路。

对于逻辑电路的分析和设计,采用了一套完全不同于模拟电路的分析和设计方法。由于逻辑电路的输入和输出信号只有两种取值0和1,它可以用"逻辑代数"这一数学工具来加以描述。常用真值表、卡诺图、特征方程和状态转移图等方法来分析和设计逻辑电路。

目前,数字电路的应用已极为广泛。在数字通信系统中,在图像及电视信号处理中,都可以用若干个0和1编制成各种代码,分别代表不同的信息含义;在自动控制中,可以利用数字电路的逻辑功能,设计出各种各样的数字控制装置;在测量仪表中,可以利用数字电路对测量信号进行处理,并将测试结果用十进制数码显示出来;尤其是在数字电子计算机中,可以利用数字电路实现各种功能的数字信息处理。数字电子计算机已日益渗透到国民经济和人民生活的一切领域之中,并已带来了许多方面根本性的变革。

必须指出,数字电路只能对数字信号进行处理,它的输入和输出均为数字信号,而大量的物理量几乎都是模拟信号。因此,必须将模拟信号转换成数字信号,才可送给数字电路进行处理,而且还要把数字结果再转换成模拟信号。完成将模拟信号转换成相应数字信号的电路称为模数转换电路;完成将数字信号转换成相应模拟信号的电路称为数模转换电路。

必须说明的是,随着中、大规模集成电路的飞速发展,成本不断降低,通用中、大规模功

能块已大量使用。与此同时,逻辑设计方法也在不断地发展。此外,数字电路的概念也在发生变化,例如,在单片计算机中已将元器件制造技术、电路设计技术、系统构成技术等融为一体,元器件、电路、系统的概念已经趋于模糊了。数字电路和设备随着新技术的发展也在不断变化,类型层出不穷,所以数字技术是一门发展很快的学科。本书仅仅介绍了数字电路的基础。

数字电路是一门实践性很强的技术基础课,除要掌握基本原理、基本方法以外,更重要的是灵活运用,因此,在学习中要完成一定数量的习题和配有一定的实验。只有这样,才能掌握本课程基本内容,掌握分析问题的基本方法,培养灵活地解决实际问题的能力。

1 数字逻辑基础

1.1 数制与码制

1.1.1 数制

数制是指进位计数制。一般有二进制、八进制、十进制和十六进制等。人们最常用的是十进制数，而在数字系统中多采用二进制数，有时也采用八进制数和十六进制数。

1.十进制数

十进制数是由 0、1、2、3、4、5、6、7、8、9 十个不同的数码来表示的。任何一个数都可以用上面十个数码按一定规律排列起来表示。十进制的特点是："逢十进一"。

例如把十进制数 2996 展开表示为：

$$2996 = 2 \times 10^3 + 9 \times 10^2 + 9 \times 10^1 + 6 \times 10^0$$

其中十进制计数制的 10 称为基数，10^0、10^1、10^2、…、10^n 称为十进制数的"权"，即个位数的权是 10^0，十位数的权是 10^1，百位数的权是 10^2，…。一个 n 位的十制数 N_D 可表示为：

$$N_D = a_{n-1} \times 10^{n-1} + a_{n-2} \times 10^{n-2} + \cdots + a_1 \times 10^1 + a_0 \times 10^0$$

式中：$a_{n-1}, a_{n-2}, \cdots, a_1, a_0$ 是各数位相应的系数，下标 D 表示十进制数，通常十进制数下标可以省略。

上述十进制数表示法，也可以扩展到小数，不过这时小数点以右的各位数码要乘以基数的负幂次，例如，数 3.142 表示为：

$$3.142 = 3 \times 10^0 + 1 \times 10^{-1} + 4 \times 10^{-2} + 2 \times 10^{-3}$$

一般来说，任意十进制数可表示为：

$$N_D = \sum_{i=-\infty}^{\infty} a_i \times 10^i$$

式中 a_i 为基数"10"的第 i 次幂的系数。

2.二进制数

二进制数有"0"和"1"两个数码。它的每一位数都可以用任何具有两个不同稳定状态的元件来表示。如三极管的饱和与截止，继电器触点的闭合与断开，灯泡的亮与不亮等。只要规定其中一种状态表示"1"，另一种状态表示"0"就可以表示二进制数。二进制计数特点是："逢二进一"，即 1 + 1 = 10(读为壹零)，本位为零，并向高位进一。二进制数的基数是 2，每位数的"权"分别为 2^0、2^1、2^2、…、2^n。一个 n 位的二进制数 N_B 可表示为：

$$N_B = b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \cdots + b_1 \times 2^1 + b_0 \times 2^0$$

3.八进制数与十六进制数

八进制有八个数码，它们分别是 0、1、2、3、4、5、6、7。其计数特点是"逢八进一"。八进制数的基数为 8，它的"权"相应为 8^0、8^1、8^2、…。

十六进制数有十六个数码，它们分别是 0、1、2、3、4、5、6、7、8、9、A(10)、B(11)、C(12)、D(13)、E(14)、F(15)。其计数特点是："逢十六进一"，基数为 16，它的"权"相应为 16^0、16^1、16^2…。

例如十六进制数 4AB6 可表示为：

$$N_H = 4AB6 = 4 \times 16^3 + A \times 16^2 + B \times 16^1 + 6 \times 16^0$$

式中 A、B 对应于十进制数中的 10,11，下标 H 表示十六进制数，若下标为 O 表示八进制数。

4.数制之间的转换

(1)十进制数转换成二进制数

要把某一个十进制数转换成二进制数，通常采用"除 2 取余"的方法来解决，即将 N_D 除 2 取余数得 b_0，其商再连续除 2，余数为 b_1，如此下去直至商为零。每次所求的余数分别为 b_0、b_1、b_2、…、b_{n-2}、b_{n-1}，而余数 b_{n-1}、b_{n-2}、…、b_1、b_0 组成的数就是对应于十进制数 N_D 的二进制数。

现以 56 为例，将十进制数转换成二进制数的过程表示如下：

$$\begin{array}{r|l l}
2 & \underline{56} & \text{余 } 0 \cdots\cdots b_0 \\
2 & \underline{28} & \text{余 } 0 \cdots\cdots b_1 \\
2 & \underline{14} & \text{余 } 0 \cdots\cdots b_2 \\
2 & \underline{7} & \text{余 } 1 \cdots\cdots b_3 \\
2 & \underline{3} & \text{余 } 1 \cdots\cdots b_4 \\
2 & \underline{1} & \text{余 } 1 \cdots\cdots b_5 \\
 & 0 &
\end{array}$$

得： $b_5b_4b_3b_2b_1b_0 = 111000$

即： $56 = (111000)_B$

对于小数可写成：

$$N_D = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \cdots + b_{-(n-1)} \times 2^{-(n-1)} + b_{-n} \times 2^{-n}$$

将上式两边分别乘以 2，得：

$$2 \times N_D = b_{-1} \times 2^0 + b_{-2} \times 2^{-1} + \cdots + b_{-(n-1)} \times 2^{-(n-2)} + b_{-n} \times 2^{-(n-1)}$$

由此可见，将十进制小数乘以 2，取其个位数为 b_{-1}。不难推知，将十进制小数每次除去上次所得积中之个位连续乘以 2，直到满足误差要求进行"四舍五入"为止，就可完成由十进制小数转换成二进制小数。例如，可按如下步骤将(0.706)D 转换成误差 ε 不大于 2^{-10}的二进制小数：

$$0.706 \times 2 = 1.412 \cdots 1 \cdots b_{-1}$$
$$0.412 \times 2 = 0.824 \cdots 0 \cdots b_{-2}$$
$$0.824 \times 2 = 1.648 \cdots 1 \cdots b_{-3}$$

$$0.648 \times 2 = 1.296 \cdots 1 \cdots b_{-4}$$
$$0.296 \times 2 = 0.592 \cdots 0 \cdots b_{-5}$$
$$0.592 \times 2 = 1.184 \cdots 1 \cdots b_{-6}$$
$$0.184 \times 2 = 0.368 \cdots 0 \cdots b_{-7}$$
$$0.368 \times 2 = 0.736 \cdots 0 \cdots b_{-8}$$
$$0.736 \times 2 = 1.472 \cdots 1 \cdots b_{-9}$$

由于最后的小数小于 0.5，根据"四舍五入"的原则，b_{-10} 因为 0。所以，$(0.706) = (0.101101001)_B$，其误差 $\varepsilon < 2^{-10}$。

(2)二进制数转换成十进制数

要把一个二进制数转换为十进制数，可按转换数制的"权"与系数的乘积之和求得，即将该二进制数按权位展开，相加而得。如将$(11011)_B$转换成十进制，有：

$$(11011)_B = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 16 + 8 + 2 + 1 = 27$$

(3)十进制数与八、十六进制数的转换

一个十进制数要转换成八进制数或十六进制数，其方法与十进制数转换为二进制数相类似，即采用"求商取余"法，一个十进制数连续除以 8 或 16，直到商为零，得到的一组余数就是要转换的结果。

例 1.1　把125 转换为八制数

解

$$\begin{array}{r|l}
8 & 125 \quad 余5 \\ \hline
8 & 15 \quad 余7 \\ \hline
8 & 1 \quad 余1 \\ \hline
 & 0
\end{array}$$

得：　$125 = (175)_O$

例 1.2　把578 转换为一个十六进制数

解

$$\begin{array}{r|l}
16 & 578 \quad 余2 \\ \hline
16 & 36 \quad 余4 \\ \hline
16 & 2 \quad 余2 \\ \hline
 & 0
\end{array}$$

得：　$578 = (242)_H$

一个八进制数或十六进制数转换为一个十进制数其方法同样可采用按权展开相加，只是要注意十六进制数中 A、B、C、D、E、F 相当于十进制数 10、11、12、13、14、15 即可。

例 1.3　把十六进制数6C5 转换为十进制数

解 $(6C5)_H = 6\times16^2 + 12\times16^1 + 5\times16^0 = 6\times256 + 12\times16 + 5 = 1373$

(4)二、八、十六进制数之间的转换

二、八、十六进制数之间的转换是很方便的。其转换方法如下：

将二进制数从右向左，每三位为一组，(不足三位补0)由每组对应的八进制数码所构成的数，便是它所对应的八进制数。如：

二进制数：100 100 111 011

八进制数： 4 4 7 3

即$(100100111011)_B = (4473)_O$

表 1.1.1 二进制数与八进制数之间的对应关系

二进制	八进制	二进制	八进制
000	0	100	4
001	1	101	5
010	2	110	6
011	3	111	7

将二进制数从右向左，每四位为一组(不足四位补0)，于是由每组对应的十六进制数码所构成的数，便是它所对应的十六进制数。如：

二进制数： 1001 0011 1011

十六进制数： 9 3 B

即$(100100111011)_B = (93B)_H$

表 1.1.2 二进制数与十六进制数对应关系

二进制	十六进制	二进制	十六进制
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

把一个八进制数或十六进制数转换为一个二进制数只要把每位数用三位二进制数或四位二进制数表示即可。

例 1.4 把$(527)_O$八进制数转换为二进制数

解 5用101表示

2用010表示

7用111表示

故$(527)_O = (101010111)_B$

例 1.5 把$(8BC)_H$ 十六进制数转换为二进制数

解 因8→1000

B→1011

C→1100

故 $(8BC)_H=(100010111100)_B$

1.1.2 码制

在数字系统中,其信息可分为两类,一类是数值,另一类是文字符号。为了表示文字符号信息,往往也采用一定位数的二进制数码来表示,这个特定的二进制码称为代码。建立这种代码与十进制数值、字母、符号一一对应关系称为编码。

1.*有权码*

在编码中,用四位二进制数 $b_3b_2b_1b_0$ 来表示十进制数中的 0~9 十个数码,如果把二进制数码的值赋以一定的位权,这样的数码称为有权码。其中,最常用的是 8421BCD 码,即十个数码(0~9)与自然二进制数一一对应。b_0 位的权为 $2^0=1$,b_1 位的权为 $2^1=2$,b_2 位的权为 $2^2=4$,b_3 位的权为 $2^3=8$。8421BCD 码是由四位二进制数的 0000(0)到 1111(15)十六种组合中的前十种组合,即取 0000(0)~1001(9),其余六种组合是无效的。

如果把 b_3 位的权定为 2 或 5,b_2 位的权定为 4,b_1 位的权定为 2,b_0 位的权为 1,则产生 2421BCD 码或 5421BCD 码。这类有权码对应关系见表 1.1.3。各种有权码,只要把每位二进制码与它的权相乘并求和,就是该有权码所代表的十进制数值。

表 1.1.3 几种有权 BCD 码

十进制数	8421BCD 码	2421BCD 码	5421BCD 码
0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	1000
6	0110	0110	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100

2.*无权码*

无权码与有权码不同,无权码中各数位并不具有固定的“权”。常用的无权码有余 3 码和格雷码。

余 3 码是由 8421BCD 码加 3(0011)而得来的,即用 0011~1100 十个数码来表示十进制数 0~9,从编码中可以看出,每位十进制数是不能通过权位关系表示的。

格雷码在码盘装置中常被采用。这种码的特点是:相邻的两个码组之间仅有一位不同。

因此，在数字电路中，数码按次序变化时电路不会产生瞬时突跳误差，减小出错的可能性。

1.2 逻辑代数基础

逻辑代数又称布尔代数，是由十九世纪英国数学家布尔（Boole）所提出的，是研究逻辑电路的数学工具，它为分析和设计逻辑电路提供了理论基础。人们把根据一定的逻辑关系作出因果判断的数字电路称作“逻辑电路”。

1.2.1 逻辑函数

逻辑代数是按一定逻辑规律进行运算的，用字母表示变量，但这个逻辑变量只取两个值“0”和“1”，而没有其它值，其中“0”与“1”并不表示数量上的大小，而只是表示两种对应的逻辑状态。

如果 A 是逻辑变量，F 是 A 的函数，即：

$$F = f(A)$$

由于只有一个变量 A，它是单变量函数。

对于二变量逻辑函数 F = f(A、B)，变量 A、B 有四种可能的组合，当每一种 A、B 取值确定后，就能确定 F 值，对于多变量逻辑函数可依次类推。

1.2.2 基本逻辑运算

在逻辑代数中，有“与”、“或”、“非”三种基本逻辑运算。运算作为一种函数关系处理时，它可以用语句描述，或用逻辑表达式描述，还可用表格及图形来描述。把描述逻辑关系的表格称为真值表，它是逻辑变量所有可能取值的全部结果。用规定的图形符号来表示逻辑运算称为逻辑符号。

1.与运算

当决定事件的条件（A、B…）全部具备之后，该事件（F）才会发生，这种因果关系称为“与”逻辑，也称逻辑乘。写为：

$$F = A \cdot B$$

简写为 F = AB，乘号“·”常被省略。

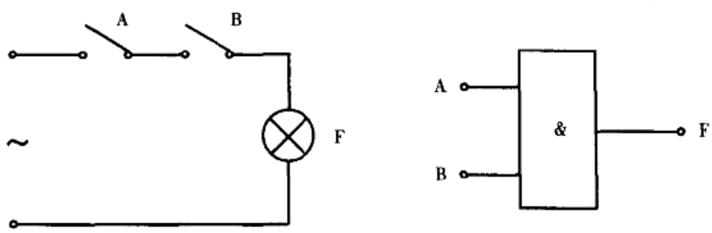


图 1.2.1 与逻辑

在图 1.2.1(a)照明电路中，开关 A 与 B 属串联关系，只有当 A 与 B 都闭合时，灯泡 F 才亮。开关 A、B 中只要有一个断开，灯泡 F 就不亮。由此可知，只有当 A 闭合，B 闭合两个条件全部具备时，才能产生灯亮的结果。因此，构成“与”逻辑关系。其逻辑符号如图 1.2.1

(b)所示。

开关 A、B 只有两种状态,即开关闭合与断开,用“1”表示闭合,则用“0”表示断开。灯泡 F 也只有两种状态,灯泡亮与灭,用“1”表示灯泡亮,则用“0”表示灯泡灭。

灯 F 与开关 A、B 用逻辑函数关系表示为:

$$F = A \cdot B = AB$$

通常把逻辑电路的输入、输出关系列成表格,称为逻辑关系表(又叫做真值表)。它能完整地表达逻辑电路的输入、输出关系。表 1.2.1 就是两输入端与门的关系表。

2.或运算

决定某事件的各个条件(A、B…)中,只要具备某一个条件,该事件(F)就会发生。这种因果关系称为或逻辑关系,即为或运算,也称逻辑或。写为:

$$F = A + B$$

在图 1.2.2(a)所示的照明电路中,开关 A 与 B 属并联关系,只要开关 A 或 B 有一个闭合,灯泡 F 就亮。由此可得,灯 F 与开关 A、B 属于或逻辑关系。其逻辑符号如图 1.2.2(b)所示,逻辑真值表如表 1.2.2 所示。

表 1.2.1 与逻辑真值表

A	B	F = AB
0	0	0
0	1	0
1	0	0
1	1	1

表 1.2.2 或逻辑真值表

A	B	F = A + B
0	0	0
0	1	1
1	0	1
1	1	1

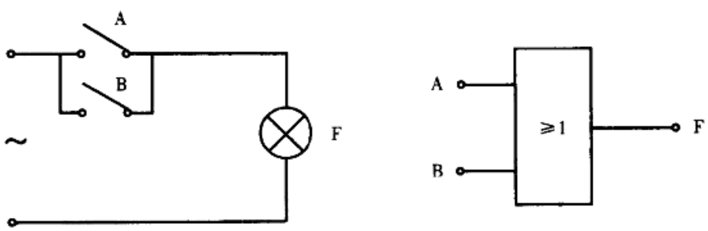


图 1.2.2 或逻辑

3.非运算

结果和条件处于相反状态的因果关系称为“非”逻辑。如图 1.2.3(a)中开关 A 与灯泡 F 相并联,当 A 断开时灯亮,当 A 闭合时灯反而熄灭,可见灯亮这一结果和开关 A 闭合这个条件构成非逻辑。其逻辑表达式写为:$F = \overline{A}$,逻辑真值表如表 1.2.3 所示。非运算逻辑符号如图 1.2.3(b)。

表 1.2.3 非运算真值表

A	F
0	1
1	0

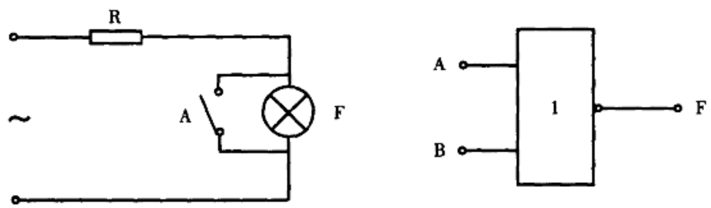


图 1.2.3　非逻辑

1.2.3　逻辑代数的运算法则

1. 基本定律

(1)逻辑加：$0+0=0$　$A+0=A$

$0+1=1$　$A+1=1$

$1+0=1$　$A+A=A$

$1+1=1$

(2)逻辑乘：$0\cdot 0=0$　$A\cdot 0=0$

$0\cdot 1=0$　$A\cdot 1=A$

$1\cdot 0=0$　$A\cdot A=A$

$1\cdot 1=1$

(3)逻辑非：$\bar{0}=1$　$\bar{\bar{A}}=A$

$\bar{1}=0$　$A+\bar{A}=1$

$A\cdot\bar{A}=0$

2. 交换律

$$AB=BA$$
$$A+B=B+A$$

3. 结合律

$$ABC=(AB)C=A(BC)$$
$$A+B+C=A+(B+C)=(A+B)+C$$

4. 分配律

$$A(B+C)=AB+AC$$
$$A+BC=(A+B)(A+C)$$

5. 吸收律

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

6. 反演律(摩根定律)

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

例 1.6　用真值表证明：$\overline{AB}=\bar{A}+\bar{B}$

证明　列写各变量真值表：

A	B	$\bar{A}$	$\bar{B}$	$\overline{AB}$	$\bar{A}+\bar{B}$
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	0	0

由真值表可见，A、B 输入端在任一逻辑状态时，等号两边的逻辑值$\overline{AB}$与 $\bar{A}+\bar{B}$ 完全相等，故$\overline{A\cdot B}=\bar{A}+\bar{B}$ 成立。

例 1.7 试证明：$ABC+AB\bar{C}+A\bar{B}C+\bar{A}BC=AB+AC+BC$

证明 左边 $=ABC+AB\bar{C}+A\bar{B}C+\bar{A}BC$

$$=(ABC+AB\bar{C})+(ABC+A\bar{B}C)+(ABC+\bar{A}BC)$$
$$=AB(C+\bar{C})+AC(B+\bar{B})+BC(A+\bar{A})$$
$$=AB+AC+BC$$
$$=\text{右边}$$

故等式成立。

最后值得指出的是：

(1)逻辑变量 A、B、C 等只能取值为“0”或“1”。

(2)这些定律只反映逻辑关系，而不是数量之间的关系。在运算中，不能简单地套用普通代数的运算规则，例如，移项规则就不能使用。

(3)基本定律的正确性，可以通过与、或、非三种基本逻辑功能加以证明，其它定律都可以用真值表来得到证实。

1.3 逻辑函数的化简

一个逻辑函数可以有多种不同的逻辑表达式，如：与一或表达式、或一与表达式、与非一与非表达式及与一或一非表达式等。例如：

$F=AB+\bar{B}C$ 与一或表达式

$=(A+\bar{B})(B+C)$ 或一与表达式

$=\overline{\overline{AB}\cdot\overline{\bar{B}C}}$ 与非一与非表达式

$=\overline{\overline{(A+\bar{B})}+\overline{(B+C)}}$ 或非一或非表达式

$=\overline{\bar{A}B+\bar{B}\bar{C}}$ 与一或一非表达式

其中与一或表达式是比较常见的，同时与一或表达式可以比较容易地同其它形式的表达式相互转换。

逻辑函数的化简，一般是指要求化为最简的与一或表达式，即在满足乘积项最少的条件下，要求每个乘积项中变量的个数也最少。用化简后的逻辑表达式构成的逻辑电路可节省器件，降低成本，提高逻辑电路工作的可靠性。

化简逻辑函数的方法，一般有代数法和卡诺图法，下面分别加以介绍。