数字逻辑电路

北京无线电学校

王惠民　主编

国防工业出版社

数字逻辑电路

北 京 无 线 电 学 校

王惠民 主编

国 防 工 业 出 版 社

内容简介

本书是在数字逻辑电路基本设计理论和方法的基础上，选用常用TTL数字集成电路，作为逻辑设计和分析的举例，并介绍使用标准TTL中速SSI和MSI电路实现电路设计和应用中的问题。

本书系中等专业学校工科电子类数字电子计算机专业统编试用教材，也可以供从事这方面工作的同志们参考。

数字逻辑电路

北京无线电学校

王惠民 主编

国防工业出版社出版

北京市书刊出版业营业许可证出字第074号

西北电讯工程学院印刷厂印刷

内部发行

●

787×1092 1/16 印张 13$\frac{1}{8}$ 331千字

1983年3月第一版 1983年3月第一次印刷 印数1-5,000册

统一书号：N15034(教-81) 定价 1.08 元

前　言

本书系中等专业学校工科电子类数字电子计算机专业（硬件）统编教材。

根据原四机部79年6月无锡教材会议推荐的教学参考大纲，以及同年在北京举行的“数字逻辑电路”编写提纲讨论会上通过的编写题纲编写。

本课程为数字电子计算机专业（硬件）的一门专业基础课。其前导课程为：“晶体管脉冲与数字电路”；后继课程为：“数字电子计算机原理”。根据编写题纲，本教材由三大部分组成：逻辑电路基础（第一、二章）；组合电路（第三、四、五章）和同步时序电路（第六、七、八章）等，总学时数为120学时左右。在保持以上教材内容理论系统性的前提下，考虑中专的教学特点，侧重介绍逻辑电路中最基本的、实用性较强的设计理论和实现方法，以及TTL中小规模集成电路的逻辑功能分析和典型应用。以使学生适应目前国内计算机硬件结构由TTL小规模电路向中大规模集成电路过渡的发展趋势。

为了加强中专教学重视理论和实践相结合的教学特点，在使用本教材时应结合“逻辑电路训练器”（国内已有多种产品）进行实验教学，以培养学生运用TTL集成电路进行逻辑电路设计及调试的能力。

本书由北京无线电学校王惠民同志负责编写，湖北电子技术学校喻光明同志负责统审。在本书编审的过程中，南京无线电校、天津无线电机械学校、贵州无线电校、常州无线电校、738厂技校以及有关厂、所同志参加了对本书初稿的讨论。

北京无线电学校楼星、哈长富同志对教材初稿本的试用及统审修订工作，都给予了热情的帮助。王丽平同志为本书绘制了全部插图，对于以上兄弟学校及许多同志的支持，在此仅致谢意。

由于编者学识水平和教学经验所限，书中错误和不妥之处一定很多，希望得到读者的批评和指正。

编　者

目　　录

第一章　逻辑代数

逻辑代数又称布尔代数，其基本原理是英国数学家乔治·布尔（G·Boole）于1847年提出的。布尔提出了用符号来表达语言和思维的逻辑性，并推导出一些基本定律。可是，直到1938年，美国麻省理工学院的香农（Shannon）才将布尔代数用于设计电话继电器开关电路，从而使布尔代数逐步进入了数字电子学的技术领域。

目前，逻辑代数已成为分析和设计数字逻辑电路的一种数学工具。逻辑代数特别适用于设计由集成门和触发器所构成的各种数字逻辑电路。所以，掌握逻辑电路的基本概念和运算方法，将是学习数字逻辑电路的一个十分重要的方面。

从逻辑代数的数学体系上看，它又可分为集合代数、命题代数和开关代数三个组成部分。本章不是从纯数学的角度来讨论逻辑代数，而是以逻辑代数在数字逻辑电路中的应用为重点，将命题代数和开关代数的一些内容加以综合，简要地阐述逻辑代数的基本概念和定律，以及如何用逻辑代数形式描述逻辑关系和进行运算的方法。

1.1 逻辑变量

普通代数中用英文字母表示变量，变量代表的是不同的十进制数。

逻辑代数中也采用英文字母表示变量。不过，这个变量却是指“表达语言和思维的逻辑性”。所以，我们称逻辑代数中的变量为逻辑变量。而逻辑变量又可归结为命题的真假取值。

1.1.1 命题

人的思维是对客观事物的认识活动。虽然所认识的事物在内容上是千变万化的，但人们用以认识客观事物的方法即思维的逻辑形式却是相同的，它往往表现为判断，把判断写成语言文字的形式便是命题。

语言的单位是句子，而句子又有命令句、陈述句、疑问句等。其中只有陈述句具有判断功能。如果判断是客观现实的正确反映，它是真实可靠的；否则便是错误的、假的。所以说，陈述句就是命题。

由此可知，逻辑代数中的字母表示逻辑变量，而每一个逻辑变量又是一个命题。换言之，一个字母表示一个命题。

命题有二个重要特性：一是命题有具体内容；二是命题有真和假之分。

如下列陈述句（命题）：

地球绕太阳转。　　（真命题）
雪不是黑的。　　（真命题）
熊猫生活在北极。　　（假命题）
51能被4整除。　　（假命题）

有些命题，有时可能是真的；有时又可能是假的，应根据说话时的事实情况来决定该命题的真假。

如：今天下雨了。

我在教室里。

以上六个命题都有实际内容，称为实际命题。由于逻辑代数研究的是命题之间的一般运算规律问题，因而不必事事涉及具体的实际命题，可以用字母代表的抽象命题来替代实际命题。

1.1.2 逻辑变量的取值

逻辑代数中用字母表示抽象的命题。如用 A 表示一个命题，A 的内 容 可 以是任意的陈述句，其真假值只有二种情况：真和假。

为了便于书写和演算，逻辑代数规定用“0”和“1”两种符号表示命题的假和真。（当然也可以作相反的规定，本书采用“0”表示假命题；“1”表示真命题。）如为真命题时，记作 $A=1$；A 为假命题 时，记作$A=0$。如果有：$A=0$；$B=1$；$C=0$，则 表示有三个抽象命题，对命题的内容并未说明，但它们的真假值是明确的，即 A为假命题；B 为 真命题；C 为假命题。

这里需要强调指出的是：逻辑代数中的“0”和“1”仅是逻辑变量的二种取值，它们没有数量的概念和性质。和普通代数的 0 和 1 有着本质上的区别，切勿混淆。

1.2 逻 辑 运 算

在上节中，列举了几个判断真假的句子，它们都仅包含一个判断内容，称为基本命题。可是，在实际问题中，为了描述较复杂事物间的逻辑关系，则需要用二个或二个以上的基本命题进行复合，于是就出现了复合命题。

在综合研究了大量的复合命题之后，可以发现当基本命题构成复合命题时，仅有三种基本关系。逻辑代数中称为：逻辑“或”；逻辑“与”；逻辑“非”。用字母和符号表示为：$F=A+B$；$F=A\cdot B$；$F=\overline{A}$。采用代数式表示之复合命题，称为逻辑运算表达式。

1.2.1 逻辑“或”运算

当命题 A 和 B 有一个为真命题时，（包括两个都为真）由 A 和 B 构成的复合命题才为真命题，称为逻辑“或”运算。“或”运算的算符有“V”和“+”等。我们采用“+”号表示“或”运算。用逻辑表达式描述“或”逻辑时可记 作：$F=A+B$。“+”可以读作“加”。“或”逻辑也可以用二个以上的基本命题进 行复合。如 $F=A+B+C+\cdots\cdots$

下面给出几个实际复合命题，用以说明“或”运算的含义和性质。

例 1—1 某图书室 借 阅 处由小王和小李负责，若要判断“能否借书？”时，就要看小王和小李来了没有？

如果用 A 表示小王；B 表示小 李。“1”表 示 来 了，“0”表示没来。用 $F=1$ 表示借书；$F=0$ 表示不借书。那么，这个判断在什么情况下是真；什么情况下是 假 呢？因为具体情况不同，其真假值如表 1—1 所示。

所以，该复合命题为：“小王或小李来了，才能借书。” 即$F=A+B$

例 1-2 如图 1-1 所示的并联开关 A 和 B；判断电灯是否发光，可根据开关的通断情况来决定。

设开关接通为“1”；断开为“0”。电灯F亮的复合命题为“1”，不亮为“0”。则判断此电路中灯F亮不亮的复合命题为：“当开关A或开关B接通时，电灯F亮”。

即 $F=A+B$ 在实际判断时，其真假值如表1-2所示。

表1—1 例1—1的真假值

A	B	F	说　　明
0	0	0	小王没来；小李没来，不借书。
0	1	1	小王没来；小李来了，能借书，
1	0	1	小王来了；小李没来，能借书。
1	1	1	小王来了；小李来了，能借书。

表1-2　例1-2的真假值

A	B	F	说　　明
0	0	0	开关A，B断开，电F灯不亮。
0	1	1	开关A断开，B接通，电灯F亮。
1	0	1	开关A接通，B断开，电灯F亮。
1	1	1	开关A接通，B接通，电灯F亮。

以上二例中，虽然A、B、F所表示的基本命题内容不同，但其抽象的复合命题都是一样的。即为：$F=A+B$。其真假值的取值组合，均为表1-3所示的形式。这种反映逻辑变量各种取值组合的表格，称全值表或真值表。

真值表是一种非常直观地描述逻辑问题运算的表达方式。利用真值表可以分析各种逻辑

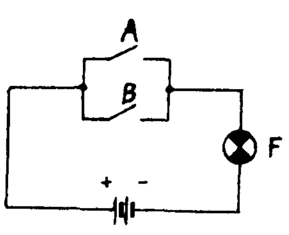

图1-1　并联开关电路

表1-3 “或”运算真值表

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

运算的性质，以及推导和验证逻辑代数基本定律和公式。利用真值表输入逻辑变量（指基本命题）组合和输出逻辑变量(指复合命题)的关系，是分析和设计逻辑电路的基本方法之一。

根据表1-3“或”运算真值表，可以归纳出“或”运算的一些基本性质：

1. 0，1律

0，1律揭示了单变量和常量之间的关系，它们有：

$$A+0=A \qquad (1.1a)$$

由 $0+0=0$；$1+0=1$ 而得出。

$$A+1=1 \tag{1.2a}$$

由 $0+1=1$；$1+1=1$ 而得出。

2. 重迭律

$$A+A=A \tag{1.3a}$$

3. 交换律

$$A+B=B+A \tag{1.4a}$$

4. 结合律

$$A+(B+C)=(A+B)\ C \tag{1.5a}$$

以上各定律均可由真值表加以证明。

“或”运算除重迭律外，其余均与普遍代数的加法运算性质在形式上相同。所以，“或”运算又称逻辑和，逻辑加等。

需要指出的是：“十”号在本书中有多种含义。

如：$1+1=2$　　普通加法

　　$1+1=10$　　二进制加法

　　$1+1=1$　　逻辑加法

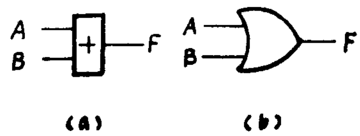

图 1-2 “或”运算逻辑图形符号

本章所讨论的为逻辑加。在以后各章中的“十”运算，可根据具体情况加以区别。

为了便于将逻辑代数表达式和构成实际逻辑电路的器件联系起来，逻辑运算之间的关系，还可以采用图形符号来描述。我国于 1977 年颁布的部标 SJ223—77 规定使用矩形符号。国际上则较广泛采用 IEEE—ASA 图形符号。

“或”运算的逻辑图形符号，如图 1-2 所示。其中（a）为部标符号；（b）为 IEEE—ASA 符号。以下将 IEEE—ASA 符号称国际通用符号。

1.2.2 逻辑“与”运算

当且仅当命题 A 与 A 皆为真时，才为真的复合命题，称为逻辑“与”运算。“与”运算的算符有：“∧”；“×”；“·”等。我们采用“·”表示“与”运算。用逻辑代数式描述“与”运算时，可记作 $F=A\cdot B$；或 $F=AB$，“·”读作“与”或“乘”均可。

“与”逻辑也可以用二个以上的基本命题进行复合。如：$F=A\cdot B\cdot C\cdot \cdots\cdots$

“与”逻辑真值表如表 1-4，逻辑图形符号如图 1-3 所示。其中（a）为部标；（b）为国际通用符号。

表1-4　与运算真值表

A	B	$F=A\cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

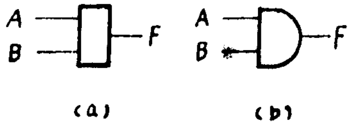

图 1-3 “与”运算逻辑图形符号

根据表 1-4“与”运算真值表，可得出“与”运算的一些基本性质：

1. 0，1 律

$$A \cdot 1 = A \tag{1.1b}$$

$$A \cdot 0 = 0 \tag{1.2b}$$

2. 重迭律

$$A \cdot A = A \tag{1.3b}$$

3. 交换律

$$A \cdot B = B \cdot A \tag{1.4b}$$

4. 结合律

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C \tag{1.5b}$$

大家知道，在普通代数运算中 $A \cdot A = A^2$，而逻辑代数中 $A \cdot A = A$。除此之外，“与”运算和普通代数乘法规律在形式上相同，所以“与”运算又称为逻辑乘，或称为逻辑积。

1.2.3 逻辑“非”运算

与命题A真假值相反的命题，称为逻辑“非”运算。因为“非”运算仍为基本命题，所以它不是一个复合命题，其作用就是对原命题进行否定，所以“非”运算又称逻辑反；逻辑补；逻辑否定等。“非”运算的算符有：“′”“*”“—”，我们采用“—”。用逻辑表达式描述“非”运算时，记作 $F = \overline{A}$，读作“非”A或A“非”均可。

“非”运算反映二个互相矛盾的命题的判断问题。如有这样两个命题，A：小赵昨天去学校了。F：小赵昨天没有去学校。命题F成立与否，和命题A有关。若A为假，则F为真。A为真，则F为假。

例 1-3 如图 1-4 所示的电灯开关电路，假设开关A断开为“1”，灯F亮为“1”A接通为“0”，灯F不亮为“0”，则F和A的关系为“非“运算。

“非”运算的真值表和图形符号分别如表 1-5 及图 1-5 所示。(a) 为部标；(b) 为国际通用符号。

表1-5 “非”运算真值表

A	$F = \overline{A}$
0	1
1	0

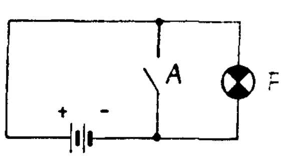

图 1-4 “非”逻辑开关电路

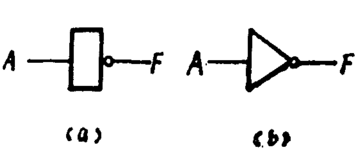

图 1-5 “非”运算图形符号

采用“非”运算之后，逻辑代数中的字母不仅可以表示命题，同时还可以表示命题的真假值，从而为描写各种实际命题提供了方便。

如图 1-1 所示的并联电路，在引入“非”运算之后，可以用逻辑代数表达式，写出亮灯的逻辑判断 (F = 1) 或复合命题。

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

也可以写出灯不亮的复合命题。(F = 0)

$$\overline{F} = \overline{A}\,\overline{B}$$

不难看出，上二式即为表 1-2 的逻辑代数式的一般描述形式。

根据表 1-5“非”逻辑真值表，“非”运算性质有：

1． 互反律

$$A + \overline{A} = 1 \tag{1.6a}$$

$$A \cdot \overline{A} = 0 \tag{1.6b}$$

2． 双重否定律

$$\overline{\overline{A}} = A \tag{1.7}$$

逻辑“或”；逻辑“与”；逻辑“非”是逻辑代数的三种基本运算。任何一个实际命题，都可以用三种逻辑运算的综合来加以描述。所以一般说：逻辑代数就是以命题为对象，包含三种逻辑运算的一种代数。

1.3 逻辑函数

1.3.1 逻辑函数表达式

用逻辑“或”、“与”、“非”三种运算描述较为复杂的命题，且用逻辑代数式的形式来表示时，称逻辑函数表达式，简称逻辑表达式。其普遍形式为：

$$F = f(A,\ B,\ C,\ \cdots)$$

由于 A、B、C、…表示逻辑变量，则 F 就是由A、B、C、…等逻辑变量构成的逻辑函数。

例 1-4“上班时我乘坐公共汽车或电车，并在车中读英语”。这个复杂的命题是由三个基本命题构成的。

如用字母 A 来表示：我乘坐公共汽车。

B：我乘坐电车。

C：我在车中读英语。

用逻辑表达式描述时，则为：

$$F = (A + B)C$$

例 1-5 如图1-6 所示楼梯开关电路，A表示楼上的开关；B表示楼下的开关。F 为楼梯照明电灯。上楼时先扳下开关 B，灯F 亮。上楼后再扳上开关A，灯F灭。反之，下楼时，先扳下开关 A，灯F亮。后扳上开关 B，灯F灭。

对于象例 1-5 这样较复杂的命题，直接写出逻辑表达式，比较困难。应先列出其真值表，再根据真值表中亮灯的条件写出逻辑表达式。

首先，假设开关 A，开关 B 扳向下时，为 A = 1；B = 1。开关扳向上时，为A = 0；B = 0。电灯亮时 F = 1；不亮时 F = 0。根据命题要求，列出例 1-5 的真值表，如表 1-6 所示。

从表 1-6 中可知，F = 1 的条件是：开关 A 和 B 同时扳向上，记作 $\overline{A} \cdot \overline{B}$；或者为开关 A 和 B 同时扳向下，记作 $A \cdot B$。

据此，可以写出亮灯的逻辑表达式：

表 1-6　例 1-5 真值表

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

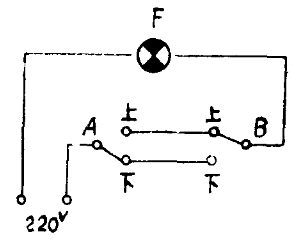

图 1-6　楼梯灯开关电路

$$F=\overline{A}\,\overline{B}+AB \qquad (“与”号可以略去)$$

当然，也可以写出灯不亮的逻辑表达式：

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

由此可知，逻辑函数用逻辑表达式或真值表来表示之后，对逻辑问题的分析提供了方便。

1.3.2　逻辑函数基本运算公式

一个复杂的实际命题，都可表现为二种或三种基本逻辑运算的综合。任何实际问题又可抽象为逻辑函数表达式，对逻辑表达式进行逻辑运算，就相当于代替了人的部分思维作用。

例 1-6　某科研项目取得了大量的数据资料，现在要从中选出有关的数据。依据的条件是：

① 把能被 4 整除，尾数是 0，各数字之和大于 13 的数据选出；（如 5680）

② 把不能被 4 整除，尾数是 0，各数字之和小于 13 的数据选出；（如 2330）

③ 把能被 4 整除，尾数是 0，各数字之和小于 13 的数选出；（如 2240）

④ 把不能被 4 整除，尾数是 0，各数字之和大于 13 的数选出；（如 23730）

⑤ 把能被 4 整除，尾数不是 0，各数字之和大于 13 的数据选出。（如 88）

根据题意，约定：

A 表示“能被 4 整除”；

$\overline{A}$ 表示“不能被 4 整除”；

B 表示“尾数是 0”；

$\overline{B}$ 表示“尾数不是 0”；

C 表示“各数字之和大于 13”；

$\overline{C}$ 表示“各数字之和小于 13”。

于是，五个复合命题可以用五个逻辑表达式表示为：

$$F_{①}=ABC; \qquad F_{②}=\overline{A}B\overline{C}; \qquad F_{③}=AB\overline{C};$$

$$F_{④} = \overline{A}BC; \qquad F_{⑤} = A\overline{B}C。$$

因为挑选条件中说明符合一个条件时即可选出。例 1-6 的逻辑表达式为：

$$\begin{aligned} F &= F_{①} + F_{②} + F_{③} + F_{④} + F_{⑤} \\ &= ABC + \overline{A}B\overline{C} + AB\overline{C} + \overline{A}BC + A\overline{B}C \end{aligned}$$

从五个命题中可以看出，命题中有些条件是重复的，肯定可以将其归纳的更简单一些。对于逻辑表达式来说，它可以通过逻辑运算，实现这个归纳过程。其运算如下：（有关运算方法，1.5节述叙）

$$\begin{aligned} F &= ABC + \overline{A}B\overline{C} + AB\overline{C} + \overline{A}BC + A\overline{B}C \\ &= ABC + AB\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C + ABC \\ &= AB(C + \overline{C}) + \overline{A}B(\overline{C} + C) + AC(\overline{B} + B) \\ &= AB + \overline{A}B + AC \\ &= B(A + \overline{A}) + AC \\ &= B + AC \end{aligned}$$

运算的最简结果为 $F = B + AC$，可以解释为：将“尾数是 0 或能被 4整除，同时各数之和大于 13”的数选出来。从而归纳出选出数据的二个条件。

逻辑运算的结果，使复杂的问题得以化简。所以，逻辑运算又称逻辑化简。

由此可知，对于实际问题的逻辑分析，以及开关电路的描述和设计，都可归结为逻辑化简。于是，逻辑代数就归纳了一些演算公式和规则。

1． 分配律

（1）乘对加的分配律

$$A(B + C) = AB + AC \tag{1.8a}$$

（2）加对乘的分配律

$$A + BC = (A + B)(A + C) \tag{1.8b}$$

这个性质是逻辑代数所特有的，现证明如下：

$$\begin{aligned} (A + B)(A + C) &= (A + B)A + (A + B)C && \text{(用 1.8a)} \\ &= AA + AB + AC + BC && \text{(用1.8a)} \\ &= A + AB + AC + BC && \text{(用1.3b)} \\ &= A \cdot 1 + AB + AC + BC && \text{(用1.1b)} \\ &= A(1 + B + C) + BC && \text{(用1.8a)} \\ &= A \cdot 1 + BC && \text{(用1.1b)} \\ &= A + BC \end{aligned}$$

2． 吸收律

$$A + AB = A \tag{1.9a}$$

证明：

$$\begin{aligned} A + AB &= A \cdot 1 + AB && \text{(用1.1b)} \\ &= A(1 + B) && \text{(用1.8a)} \\ &= A \cdot 1 && \text{(用1,1b)} \\ &= A \end{aligned}$$

3． 剔除律

$$A+\overline{A}B=A+B \qquad (1.10a)$$

证明：

$$A+\overline{A}B=(A+\overline{A})(A+B) \qquad (用1.8b)$$

$$=1\cdot(A+B) \qquad (用1.6a)$$

$$=A+B$$

4. 对合律

$$AB+A\overline{B}=A \qquad (1.11a)$$

证明：

$$AB+A\overline{B}=A(B+\overline{B}) \qquad (用1.8a)$$

$$=A$$

5. 冗余律

$$AB+\overline{A}C+BC=AB+\overline{A}C \qquad (1.12a)$$

证明：

$$AB+\overline{A}C+BC=AB+\overline{A}C+BC(A+\overline{A}) \qquad (用1.6a)$$

$$=AB+\overline{A}C+ABC+\overline{A}BC \qquad (用1.8a)$$

$$=AB(1+C)+\overline{A}C(1+B) \qquad (用1.8a)$$

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

吸收律、剔除律、对合律、冗余律均为常用的化简公式。也就是说，在出现等式左边的式子时，就可以用右边的较简单的式子来代替它。

6. 反演律

反演律又称狄·摩根（De·Morgan）定理，它使用“非”运算揭示出“或”运算和“与”运算之间的关系，它是逻辑代数中一个十分重要的定律。

$$\overline{A+B}=\overline{A}\cdot\overline{B} \qquad (1.13a)$$

$$\overline{A\cdot B}=\overline{A}+\overline{B} \qquad (1.13b)$$

可利用真值表加以证明，如表1-7所示。

表1-7(a) $\overline{A+B}=\overline{A}\cdot\overline{B}$

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

表1-7(b) $\overline{A\cdot B}=\overline{A}+\overline{B}$

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

反演律揭示了“与”运算和“或”运算的内在联系。“与”之“非”得“或”；“或”

之“非”得“与”。因此，逻辑代数的三种基本运算，可以归并为“与”、“非”二种（或者为“或”、“非”二种）。甚至进一步归并为一种“与非”逻辑运算（或者为“或非”逻辑运算）。

如：用“与非”运算实现“与”运算；

$$F=\overline{\overline{AB}}=AB$$

用“与非”运算实现“或”运算；

$$F=\overline{\overline{A}\cdot\overline{B}}=A+E$$

用“与”运算实现“非”运算；

$$F=\overline{A\cdot A}=\overline{A}$$

或者：用“或非”运算实现“与”运算；

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

用“或非”运算实现“或”运算；

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

用“或非”运算实现“非”运算；

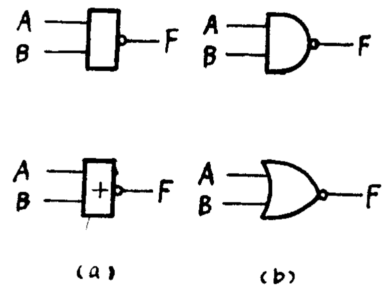

图 1-7 “与非”“或非”逻辑图形符号

$$F=\overline{A+A}=A$$

为此，在实用中我们都喜欢用“与非”逻辑（或者“或非”逻辑）来描述逻辑函数。其逻辑符号和真值表如图 1-7 和表 1-7 所示。图1-7中(a)为部标符号，(b)为国际通用符号。

1.3.3 逻辑函数的三个运算规则

以上公式均以二个逻辑变量运算为依据，如果变量为任意的，还可以根据下述三个规则将公式加以扩充。

1. 代入规则

在一个等式中，如果将某个变量的位置，都用同一函数式来替代，则新等式仍然成立。

例 1-7　$A+AB=A$

以 $A=(C+D)E$ 代入上式得：

$$(C+D)E+(C+D)EB=(C+D)E$$

证明：

$$\begin{aligned}(C+D)E+(C+D)EB&=(C+D)(E+EB)\\&=(C+D)E(1+B)\\&=(C+D)E\end{aligned}$$

例 1-8　$\overline{A+B}=\overline{A}\cdot\overline{B}$

以 $A=A+C$ 代入上式得：

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

由此不难求出 n 变量的狄·摩根定理：

$$\overline{A_1+A_2+A_3+\cdots+A_n}=\overline{A_1}\cdot\overline{A_2}\cdot\overline{A_3}\cdot\cdots\cdot\overline{A_n}$$

$$\overline{A_1\cdot A_2\cdot A_3\cdot\cdots\cdot A_n}=\overline{A_1}+\overline{A_2}+\overline{A_3}+\cdots+\overline{A_n}$$

在使用公式时，应把上式中的变量A看成是一个函数，就可以利用公式对函数式求反，增加运算的灵活性。

例 1-9　求$F = A\overline{B}C + \overline{A}B + A\overline{B}\,\overline{C}$的反函数

$$\overline{F} = \overline{ABC + AB + ABC}$$

$$= \overline{ABC} \cdot \overline{AB} \cdot \overline{ABC}$$

$$= (\overline{A} + B + \overline{C})(A + \overline{B})(\overline{A} + B + C)$$

需要指出的是，在逻辑运算中，如函数式中没有括号，其运算次序为：先做逻辑“非”运算，再做逻辑“与”运算，最后做逻辑“或”运算。如果有括号，则先运算括号内，然后再外，无论是括号内的运算，还是括号外的运算，均应按“非—与—或”的次序进行。

如需对逻辑表达式取反时，不得改变原来函数式的运算顺序。所以，取反后要加括号，才能保证原函数式的运算顺序不变。如将例 1-9 式写作：

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

$$= \overline{A} + B + A\overline{C} + \overline{A}\,\overline{B} + B + C$$

由于没加括号，改变了运算顺序，因而结果也是错误的。

2.　反演规则

反演规则是反演律和代入规则的扩充应用，利用反演规则，可以直接写出逻辑函数式的反演结果。

反演规则：对函数式 F 取反时，只需将 F 式中所有“·”号，变成“+”号；所有“+”号，变成“·”号。原变量变为反变量；反变量变为原变量。“0”变“1”；“1”变“0”。且不得改变原函数的运算顺序。所得式子就是$\overline{F}$。

例 1-10　求$F = \overline{A} \cdot B \cdot \overline{C} + C \cdot \overline{D}(E \cdot G + \overline{E} \cdot \overline{G})$的反函数。

利用反演规则，可直接写出：

$$\overline{F} = (A + \overline{B} + C)[\overline{C} + D + (\overline{E} + \overline{G})(E + G)]$$

当然，我们也可以用反演律，并通过基本公式运算求出同样结果，只是麻烦了些。

3.　对偶规则

（1）对偶式：在了解对偶规则之前，先介绍对偶式。

如果将逻辑函数表达式 F 中，所有的“+”换为“·”，“·”换为“+”；“1”换为“0”，“0”换为“1”，且不改变原逻辑函数的运算顺序，所得到的新逻辑函数表达式，记作 F′，称为 F 的对偶式。

例 1-11　$F = A \cdot (\overline{B} + C)$；　　　　$F' = A + \overline{B}C$

$F = A \cdot \overline{B} + \overline{A(C + 0)}$；　　　　$F' = (A + \overline{B})(\overline{A + C \cdot 1})$

显而易见，F 和 F′ 是互为对偶的。

（2）对偶规则

如果两个逻辑函数表达式 F 和 G 相等，则它们的对偶式 F′ 和 G′ 也相等。

关于对偶规则的证明，可以由公式（1.8a）和（1.8b）的对偶关系中看出。