数 字 逻 辑 教 程

张礼平　编著

华东理工大学出版社

内 容 提 要

本教材在阐述数字逻辑电路基本概念和原理的基础上，介绍数字系统和计算机逻辑设计的基本理论和方法，重点讨论分析方法和设计方法。教材共分10章，由逻辑代数、组合逻辑和时序逻辑的分析和设计、中、大规模器件原理及应用和专题——处理器和控制器逻辑设计四部分构成。

本教材可作为计算机类、电子类、自控类及相关专业的教材，也可作为有关专业工程技术人员的参考书。

图书在版编目(CIP)数据

数字逻辑教程/张礼平编著. —上海:华东理工大学出版社,2002.8

ISBN 7-5628-1293-4

Ⅰ.**数**... Ⅱ.**张**... Ⅲ.**数字逻辑-高等学校—教材** Ⅳ.TP302.2

中国版本图书馆 CIP **数据核字**(2002)**第** 051403 **号**

数字逻辑教程

张礼平 编著

出版	**华东理工大学出版社**	**开本**	787×1092 1/16
社址	**上海市梅陇路** 130 **号**	**印张**	17
邮编	200237 **电话**(021)64250306	**字数**	409 **千字**
网址	www.hdlgpress.com.cn	**版次**	2002 **年** 8 **月第** 1 **版**
经销	**新华书店上海发行所**	**印次**	2002 **年** 8 **月第** 1 **次**
印刷	**上海长阳印刷厂**	**印数**	1-3050 **册**

ISBN 7-5628-1293-4/O·63 **定价**:27.50 **元**

前　　言

进入21世纪后，为了适应计算机技术发展，培养面向21世纪、参与国内外市场竞争的计算机专业技术人才，教材需要推陈出新。数字逻辑是计算机科学与技术专业的技术基础，是数字电子计算机的基础理论之一。逻辑设计的理论和方法对从事数字系统和计算机研制、开发及应用的工程技术人员而言是非常重要的。

本书共10章：第1章和第2章介绍逻辑代数概论和集成逻辑门；第3章和第4章介绍组合逻辑的分析、设计方法及常用的中规模集成组合逻辑器件；第5章介绍常用的触发器器件；第6章至第8章介绍时序逻辑的分析、设计方法及常用的中规模集成时序逻辑部件；第9章介绍大规模可编程器件及应用；第10章介绍专题——处理器和控制器的逻辑设计。教材编写以培养创新能力为宗旨，力求基础理论和实际应用并重。

在本教材编写过程中，李静、金萍、朱法枝老师和研究生虞涛、杨辛宝、杨清在文字和图表方面做了大量的工作，在此深表谢意！本教材参考和引用了国内外同行的著作，在此表示由衷的感谢！教材的第1章和第2章由张立科老师编写。由于编者水平有限，教材中难免有不足之处和疏漏，恳请读者批评指正。

编者

2002年8月

目　　录

1 逻辑代数概论

本章要点 研究开关理论以及分析、设计数字逻辑电路的数学工具是逻辑代数。逻辑代数又称布尔代数或开关代数。逻辑函数的建立和逻辑函数的化简是数字逻辑电路分析和设计的基础。

本章主要讲述逻辑代数的公理、定理和规则、运算规律，逻辑函数的建立、表示和化简方法。

1.1 逻辑代数

1.1.1 逻辑代数的基本运算

逻辑代数是用代数的形式来研究逻辑问题的数学工具。其变量可以用字母 A,B,C,D 等表示，并称为逻辑变量。它所表示电路的开关接通还是断开，逻辑判断的结果是真还是假，事物存在还是不存在等等的两种可能性。逻辑变量只有两个取值，一般用 1 和 0 来表示，所以这种逻辑称为二值逻辑。二值逻辑中的“1”和“0”仅仅是一种符号的代表，既无数量的含义，又无大小、正负之分。逻辑变量之间的相互联系就反映为数学上的运算关系。

逻辑代数的基本运算有“与”运算、“或”运算及“非”运算三种。使用三种基本运算可描述各种复杂的逻辑关系。

1.1.1.1 “与”逻辑运算

“与”逻辑运算又称为逻辑乘。图 1-1 所示为两个串联开关 A，B 及一个电灯 F 所构成的电路。图中两个开关的状态组合有四种，四种不同的组合状态与灯亮、灯灭之间的关系，如表 1-1 所示。由表 1-1可见，当且仅当开关 A 与 B 同时闭合时，灯 F 才亮；否则灯灭。

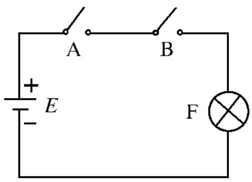


图 1-1 “与”逻辑示例

开关 A，B 同灯 F 之间的关系就是“与”逻辑关系。当决定某个事件的几个条件全部成立时，事件才会发生的因果关系称为“与”逻辑关系。

用“1”来表示开关“闭合”及灯“亮”，用“0”来表示开关“断开”及灯“灭”。表 1-1 所示“与”逻辑关系可变换成表 1-2 所示的真值表形式。所谓真值表是指把逻辑变量的所有可能的组合及其对应的结果构成二维表，此表称为真值表。

表 1-1 “与”电路状态表

开关 A 的状态	开关 B 的状态	灯 F 的状态
断 开	断 开	不 亮
断 开	闭 合	不 亮
闭 合	断 开	不 亮
闭 合	闭 合	亮

表 1-2 “与”电路真值表

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

上述的“与”逻辑关系还可表示成如下的逻辑函数表达式：$F=A\cdot B$。式中“·”为“与”逻辑运算符号，“·”可省略，A，B 是输入逻辑变量（自变量），F 是输出逻辑变量（因变量）。读作 F 等于 A“与”B。

由表 1－2 可知“与”逻辑运算（简称“与”运算）的运算规律是

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

实现“与”运算逻辑功能的电路被称为“与”门。图 1－2 是一个用二极管组成的“与”门电路。设输入的高电平为＋3 伏（用“1”表示），低电平为 0 伏（用“0”表示）；忽略二极管正向导通时的管压降，当输入 A，B 中有一个为低电平“0”时，则相应的二极管导通，输出也为低电平“0”；如果输入均为高电平“1”，则输出才是高电平“1”。图 1－3 所示为“与”门逻辑符号。

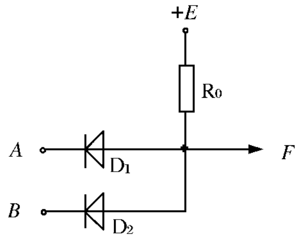


图 1－2　二极管“与”门

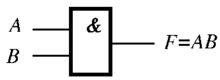


图 1－3　二输入与门逻辑符号

1.1.1.2　“或”逻辑运算

“或”逻辑运算简称“或”运算，又称为逻辑加。可用两个或两个以上开关的并联来描述。图 1－4 所示为两个开关 A，B 并联及一个电灯 F 所构成的电路。开关 A，B 与电灯亮、灭之间的关系，如表 1－3 所示。这里，开关 A，B 同灯 F 之间的关系就是“或”逻辑关系。这种逻辑关系可用表 1－4 所示真值表来描述。因此，“或”逻辑关系是指这样一种因果关系：在决定某个事件的几个条件中，只要有一个或几个条件或全部条件都得到满足，这个事件就会发生。

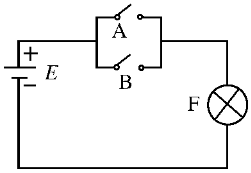


图 1－4　“或”逻辑示例

上述的“或”逻辑关系还可表示成逻辑函数表达式：$F=A+B$，式中“＋”为“或”运算符，读做 F 等于 A“或”B。

由真值表可知，“或”运算的运算规律是

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

表 1－3　“或”电路状态表

开关 A 的状态	开关 B 的状态	灯 F 的状态
断　开	断　开	不　亮
断　开	闭　合	亮
闭　合	断　开	亮
闭　合	闭　合	亮

表 1－4　“或”电路真值表

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

应该注意逻辑运算与算术运算的区别。算术运算有数值大小，各位都有权值；而逻辑运算只考虑各位本身，它没有数值的大小。例如，在二进制算术运算中，$1+1=10$；而在逻辑运算中 $1+1=1$。

实现“或”运算的电路被称为“或”门。图 1－5 是一个用二极管组成的“或”门电路。若输入端 A 或 B 中有一个为高电平“1”时，则相应的二极管就导通，输出 F 便为高电平 1；只有输入 A，B 都为低电平“0”时，输出才为低电平“0”。图 1－6 是“或”门逻辑符号。

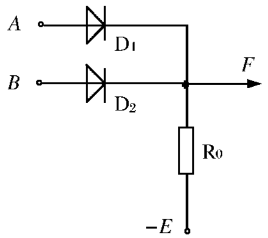


图 1－5　二极管或门

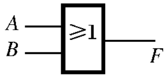


图 1－6　二输入或门逻辑符号

1.1.1.3　“非”逻辑运算

“非”逻辑运算简称“非”运算，又称逻辑“非”，或称逻辑否定。可用图 1－7 所示的电路来描述。当开关 A 断开时，灯 F 亮；当开关 A 闭合时，则灯被短路而灭。这里灯亮这件事与开关 A 闭合这个条件之间的逻辑关系就是“非”逻辑关系，真值表如表 1－5 所示。

“非”运算的逻辑函数表达式是：$F=\overline{A}$，读做 F 等于 A 非（或非 A）。称 A 为原变量，称 $\overline{A}$ 为反变量。

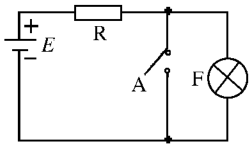


图 1－7　“非”逻辑示例

表 1－5　“非”逻辑真值表

A	F
0	1
1	0

由真值表可知，“非”运算规律是：$\overline{0}=1$　　$\overline{1}=0$

实现“非”运算的电路称为“非”门。图 1－8 所示的反相器就是一个“非”门。当输入 A 为高电平 1 时，晶体管 T 饱和，输出 F 为低电平 0；当输入 A 为低电平时，晶体管 T 截止，输出 F 为高电平 1。“非”门的逻辑符号，如图 1－9 所示。

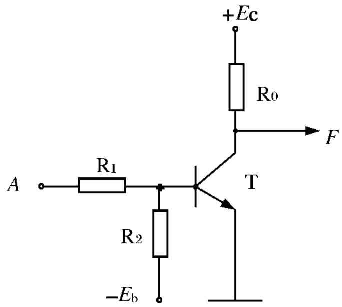


图 1－8　“非”门电路

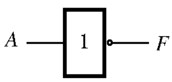


图 1－9　“非”门逻辑符号

1.1.1.4　逻辑函数

实际问题中，往往是用“与”、“或”、“非”这三种基本运算符号把有关的逻辑变量连接起来，实现特定的逻辑关系。例如楼梯灯控制电路如图 1－10 所示。楼上和楼下分别装有单刀双掷开关 A，B。上楼时先在楼下开灯，上去后再在楼上关掉；下楼时，先在楼上开灯，下来后再在楼下把灯关掉。当两个开关都扳上或扳下时，灯 F 就亮，否则灯 F 就灭。设“1”表示开关扳到上面，“0”表示扳到下面；灯 F 亮为“1”，不亮为“0”。则 F 同 A，B 之间的逻辑关

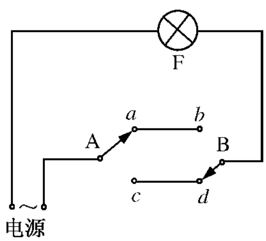


图 1－10　楼梯灯控制电路

系可表示成：$F=AB+\overline{A}\,\overline{B}$，当 A,B 都为 1，或者 A,B 都为 0 时，F 为 1，即灯亮。A,B 是输入逻辑变量（自变量），F 为输出逻辑变量（因变量）。当 A,B 取定一组值后，F 的值也就被确定了，因此 F 是 A,B 的逻辑函数。

一般，若输入逻辑变量 $A,B,C\cdots$ 的值确定以后，其输出变量 F 的值也就被唯一地确定了，则称 F 为 $A,B,C\cdots$ 的逻辑函数，记作逻辑函数表达式 $F=f(A,B,C\cdots)$。

逻辑函数表达式的书写及运算规则如下。

（1）先做括号内的逻辑运算。

（2）对某变量取“非”，可不加括号。如$\overline{A+B}$，$\overline{C+D}$等不必写成$(\overline{A+B})$，$(\overline{C+D})$。

（3）逻辑函数表达式中，如既有“与”运算，又有“或”运算，则按先“与”后“或”的规则省去括号。如$(A\cdot B)+(C\cdot D)$可写成 $AB+CD$，但$(A+B)\cdot(C+D)$则不能省去括号。

逻辑函数关系还可以用真值表来表示。例如上面讨论的逻辑函数 $F=AB+\overline{A}\,\overline{B}$，其输入变量 A 和 B，每个变量有 0，1 两个取值，所以输入状态的组合数目为 $2^2=4$，将这些输入状态分别代入 F 的表达式，求出相应的输出状态，即表 1-6 所示的真值表。

表 1-6　图 1-10 的真值表

A	B	$F=AB+\overline{A}\,\overline{B}$
0	0	1
0	1	0
1	0	0
1	1	1

逻辑函数关系除了用表达式和真值表描述外，还可用文字来表述。如上例，可表述为：当输入变量 A,B 取值相同时，则输出变量 F 的值为“1”。

如果逻辑函数 F_1 及 F_2 的真值表相同，则 F_1 等于 F_2。反之，如果 $F_1=F_2$，则其对应的真值表一定相同。

如果逻辑函数表达式 F_1 等于逻辑函数表达式 F_2，则两个逻辑函数表达式的逻辑功能相同。一个逻辑功能可用不同形式的逻辑函数表达式来表示。这时逻辑函数表达式的繁简程度是不相同的。逻辑函数表达式的繁简程度的差别，可能使实现这些逻辑函数表达式的逻辑电路的复杂程度不同。

为了使实现某种逻辑功能的逻辑电路尽可能简单，就要求出该逻辑函数表达式的最简形式，即要对逻辑函数进行简化。

1.1.2　逻辑代数的基本定律及规则

对逻辑函数表达式进行简化或作形式上的变换，需要掌握逻辑代数的基本定律及规则。为了讨论方便，先把前面已经得到的逻辑代数的基本逻辑运算规则归纳在表 1-7 中。

表 1-7　三种基本逻辑运算

逻辑运算	运算符号	运算规律
与运算	·（或者 $\wedge$）	$0\cdot0=0$　$0\cdot1=0$ $1\cdot0=0$　$1\cdot1=1$
或运算	+（或者 $\vee$）	$0+0=0$　$0+1=1$ $1+0=1$　$1+1=1$
非运算	—	$\overline{0}=1$　$\overline{1}=0$

1.1.2.1 基本定律

逻辑代数的基本定律列于表 1-8 中，这些定律可由表 1-7 所列的基本逻辑运算推导，也可通过真值表给予验证。

表 1-8 基本定律

定律名称	公式	
(1) 0—1 律	$A\cdot 0=0$	$A+1=1$
(2) 自等律	$A\cdot 1=A$	$A+0=A$
(3) 重叠律	$A\cdot A=A$	$A+A=A$
(4) 互补律	$A\cdot\overline{A}=0$	$A+\overline{A}=1$
(5) 交换律	$A\cdot B=B\cdot A$	$A+B=B+A$
(6) 结合律	$A\cdot(B\cdot C)=(A\cdot B)\cdot C$	$A+(B+C)=(A+B)+C$
(7) 分配律	$A\cdot(B+C)=AB+AC$	$A+(B\cdot C)=(A+B)\cdot(A+C)$
(8) 非非律	$\overline{(\overline{A})}=A$	
(9) 反演律	$\overline{AB}=\overline{A}+\overline{B}$	$\overline{A+B}=\overline{A}\cdot\overline{B}$
(10) 吸收律(一)	$A+AB=A$	$A\cdot(A+B)=A$
吸收律(二)	$A+\overline{A}B=A+B$	$A\cdot(\overline{A}+B)=AB$
吸收律(三)	$AB+\overline{A}C+BC=AB+\overline{A}C$	$(A+B)\cdot(\overline{A}+C)\cdot(B+C)=(A+B)\cdot(\overline{A}+C)$

反演律又称为狄·摩根(De Morgan)定律，是逻辑代数中经常使用的重要定律。用真值表证明反演律的正确性。其真值表列于表 1-9。可见，$\overline{A+B}=\overline{A}\cdot\overline{B}$，$\overline{AB}=\overline{A}+\overline{B}$。

下面用逻辑代数的公式来证明吸收律的三个等式成立。

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

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

(3) $$\begin{aligned} AB+\overline{A}C+BC &=AB+\overline{A}C+(A+\overline{A})BC \\ &=AB+ABC+\overline{A}C+\overline{A}CB \\ &=AB+\overline{A}C \end{aligned}$$

表 1-9 反演律真值表

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

1.1.2.2 逻辑代数的三个规则

逻辑代数中有三个重要规则，即代入规则、反演规则和对偶规则。

1) 代入规则

代入规则是：对逻辑等式中的任意变量 A，如果将所有出现变量 A 的位置都代以某个逻辑函数 F，则此等式仍成立。

[例 1-1] 已知 $A(B+C)=AB+AC$，若将函数 $F=C+D$ 代入等式中任一变量(假定设为 B)后，此等式仍成立。

证明:等式左边 $=A[(C+D)+C]=A(C+D)+AC=AC+AD$

等式右边 $=A(C+D)+AC=AC+AD$

等式左边 = 等式右边

[**例 1-2**]　已知 $\overline{AB}=\overline{A}+\overline{B}$,若用"$BC$"代替等式中的 B,则等式仍成立。

证明:等式左边 $=\overline{A(BC)}=\overline{ABC}$

等式右边 $=\overline{A}+\overline{BC}=\overline{A}+\overline{B}+\overline{C}$

等式左边 = 等式右边,$\overline{ABC}=\overline{A}+\overline{B}+\overline{C}$

可见,利用代入规则,可以将两个变量的狄·摩根定律推广到 n 个变量,就演变成了反演规则。

2）反演规则

反演规则是将逻辑函数表达式 F 中所有"·"变成"+",或"+"变成"·","1"变成"0","0"变成"1",并将原变量变成反变量,反变量变成原变量,并保持运算优先顺序不变,可求得该函数 F 的反函数 $\overline{F}$,这就是反演规则。

[**例 1-3**]　已知 $F=A[\overline{B}+(C\overline{D}+\overline{E}G)]$,求 $\overline{F}=$?

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

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

运用反演规则时,应注意其结果不改变原先函数的运算顺序。

3）对偶规则

对偶规则是将逻辑函数 F 中所有"·"换成"+","+"换成"·","0"换成"1","1"换成"0",而变量和运算优先顺序保持不变,就可得到一个新的逻辑函数,这个新函数称为函数 F 的对偶式,记作 F'。

[**例 1-4**]　已知 $W=A(B+C)$,求其对偶式 W'。

解:$W'=A+BC$

[**例 1-5**]　已知 $Z=A+BC$,求其对偶式 Z'。

解:$Z'=A\cdot(B+C)$

综上可见:逻辑函数两次求对偶后,得到该函数本身。

如果逻辑函数 F 和 G 相等,则其对偶式 F' 和 G' 也相等,即 $F'=G'$。

根据对偶规则来看表 1-8 中的基本定律,可以发现每一定律的左边公式和右边公式都是一对互为对偶式。所以只要左边公式成立,其右边公式也自然成立。

对偶规则对任何两个相等的逻辑函数都适用。掌握对偶规则可使要证明的公式减少一半,有利于简化函数式及函数式的变换。

[**例 1-6**]　试证明 $A\overline{B}+\overline{A}C+B\overline{C}=\overline{ABC}(A+B+C)$。

证明:$A\overline{B}+\overline{A}C+B\overline{C}=A\overline{B}+\overline{A}C+B\overline{C}+\overline{B}C+A\overline{C}+\overline{A}B$

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

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

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

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

一般情况下,$\overline{F}\neq F'$,仅在个别特殊情况下,才有 $\overline{F}=F'$。

1.1.3　常用逻辑函数及其转换

实际使用的逻辑门除“与”门、“或”门及“非”门外，还有“与非”、“或非”、“异或”、“同或”及“与或非”等集成逻辑门电路，它们可由“与”、“或”及“非”三种逻辑导出。

1.1.3.1　**常用逻辑函数**

1）**“与非”逻辑**

若逻辑变量 A,B 同输出逻辑变量 F 之间的关系如真值表 1－10 所示，则 F 同 A,B 之间便是“与非”逻辑关系，其逻辑函数表达式为 $F=\overline{AB}$。

实现“与非”逻辑的门电路称为“与非”门，一个二输入端“与非”门的逻辑符号如图1－11所示。

表 1－10　“与非”逻辑真值表

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

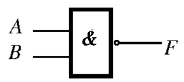


图 1－11　“与非”门逻辑符号

2)**“或非”逻辑**

若输入逻辑变量 A,B 同输出逻辑变量 F 之间的关系如真值表 1－11 所示，则 F 同 A,B 之间便是“或非” 逻辑关系。其逻辑函数表达式为 $F=\overline{A+B}$。

实现“或非”逻辑的门电路称为“或非”门。一个二输入端“或非”门的逻辑符号如图 1－12所示。

表 1－11　“或非”逻辑真值表

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

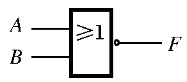


图 1－12　“或非”门逻辑符号

3）**“与或非”逻辑**

“与”、“或”及“非”三种运算结合在一起的逻辑称为“与或非”逻辑。如 $F=\overline{AB+CD}$就是一种“与或非”逻辑函数表达式。实现“与或非”逻辑的电路叫“与或非”门，其相应的逻辑符号如图 1－13 所示。

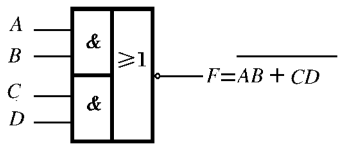


图 1－13　“与或非”门逻辑符号

4）**“异或”逻辑**

若两个输入逻辑变量相异时其输出为 1，相同时其输出为 0，则此输出与输入的逻辑关系便为“异或”逻辑关系，其真值表如表 1－12 所示。逻辑函数表达式为

$$F=A\oplus B=A\,\overline{B}+\overline{A}B=\overline{\overline{A}\cdot B+\overline{A}\cdot\overline{B}}=(A+B)(\overline{A}+\overline{B})$$

式中$\oplus$为“异或”运算符。

实现“异或”逻辑的电路叫“异或”门，其逻辑符号如图 1-14 所示。须注意，“异或”门只有两个输入端。

表 1-12　“异或”逻辑真值表

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

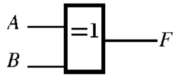


图 1-14　“异或”门逻辑符号

由真值表可知“异或”运算的规律为

$$0\oplus0=0\qquad 0\oplus1=1\oplus0=1\qquad 1\oplus1=0$$

“异或”逻辑的公式和性质。

(1) $A\oplus0=A$

(2) $A\oplus1=\overline{A}$

(3) “异或”运算中，等式一边或两边的变量位置可以互换，设 $A\oplus B=C$，则 $B\oplus A=C$ 或 $B\oplus C=A$，$C\oplus A=B$ 等等。

(4) $A\oplus(B\oplus C)=(A\oplus B)\oplus C$

(5) $A\cdot(B\oplus C)=(AB)\oplus(AC)$

(6) $A\oplus A=0$，$A\oplus\overline{A}=1$，据此可以推得

$$\underbrace{A\oplus A\oplus\cdots\oplus A}_{2N}=0,\ \underbrace{A\oplus A\oplus\cdots\oplus A}_{2N+1}=A\qquad \text{其中 } N \text{ 为自然数}$$

由此可知，“异或”运算的输出结果与变量值为 0 的个数无关；而变量值为 1 的个数为奇数，则输出为 1，如变量值为 1 的个数为偶数，则输出为 0。

例如：$1\oplus1\oplus1\oplus0\oplus0=1$，$1\oplus1\oplus1\oplus1\oplus0\oplus0=0$。

5）“同或”逻辑

若两个输入逻辑变量相同时，其输出为 1，相异时，输出为 0，则此输出与输入的逻辑关系便为“同或”逻辑关系，简称“同”逻辑，其真值表如表 1-13 所示。逻辑函数表达式为 $F=A\odot B=AB+\overline{A}\,\overline{B}=\overline{A\,\overline{B}+\overline{A}B}=(\overline{A}+B)\cdot(A+\overline{B})$，式中$\odot$为“同或”运算符。

实现“同或”逻辑的电路叫“同或”门，其逻辑符号如图 1-15 所示。“同或”门也只有两个输入端。

表 1-13　“同或”逻辑真值表

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

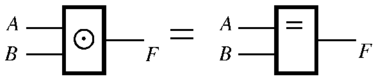


图 1-15　“同或”门逻辑符号

由真值表可知“同或”运算的规律为

$$0\odot0=1\qquad 0\odot1=1\odot0=0\qquad 1\odot1=1$$

“同或”逻辑的公式和性质。

(1) $A\odot0=\overline{A}$

（2）$A\odot 1=A$

（3）“同或”运算中，等式一边或两边的变量位置可以互换。

设：$A\odot B=C$，则 $B\odot A=C$，或 $B\odot C=A$，或 $C\odot A=B$ 等等。

（4）$A\odot(B\odot C)=(A\odot B)\odot C$

（5）$A+(B\odot C)=(A+B)\odot(A+C)$

（6）$A\odot A=1$，$A\odot\overline{A}=0$，据此可以推得

$\underbrace{A\odot A\odot\cdots\odot A}_{2N}=1$，$\underbrace{A\odot A\odot\cdots\odot A}_{2N+1}=A$　　其中 N 为自然数

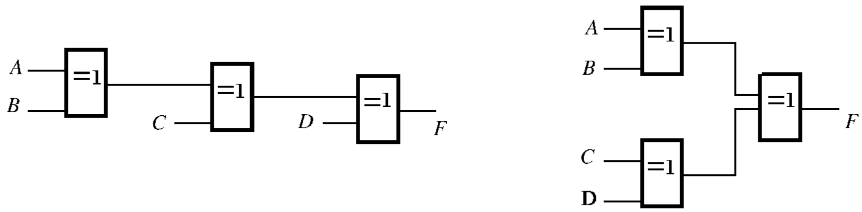
由此可知，同或运算的输出结果与变量值为 1 的个数无关；而变量值为 0 的个数为偶数，则输出为 1，如变量值为 0 的个数为奇数，则输出为 0。例如 $1\odot1\odot1\odot0\odot0=1$，$1\odot1\odot1\odot0\odot0\odot0=0$。

对“异”运算及“同”运算进行比较后可以发现

$$A\odot B=\overline{A\oplus B}=A\oplus\overline{B}=\overline{A}\oplus B$$

$$A\oplus B=\overline{A\odot B}=A\odot\overline{B}=\overline{A}\odot B$$

由于“异或”门及“同或”门都只有两个输入端，因此，当有多个变量需要进行“异或”（“同或”）运算时，须两两“异或”（“同或”）运算后再将其结果进行“异或”（“同或”）运算，直至最后结果。例如 $F=A\oplus B\oplus C\oplus D$，可有如图 1－16 所示的两种接法。



（a）　　（b）

图 1－16　$F=A\oplus B\oplus C\oplus D$ 的两种接法

1.1.3.2　逻辑函数表达式的变换

同一个逻辑函数可以有不同形式的逻辑函数表达式。

$$F=\overline{A}\,\overline{B}C+\overline{A}BC+AB\overline{C}+ABC \qquad \text{与-或表达式}$$

$$=AB+\overline{A}C \qquad \text{与-或表达式}$$

$$=(A+C)(\overline{A}+B) \qquad \text{或-与表达式}$$

$$=\overline{\overline{AB}\cdot\overline{\overline{A}C}} \qquad \text{与非-与非表达式}$$

$$=\overline{\overline{A+C}+\overline{\overline{A}+B}} \qquad \text{或非-或非表达式}$$

$$=\overline{A\cdot\overline{B}+\overline{A}\cdot\overline{C}} \qquad \text{与或非表达式}$$

其中与-或表达式是逻辑函数的最基本表示形式。运用逻辑代数基本定律，很容易被变换成其他形式表达式。

实现逻辑函数 F 有多种电路形式，使用“与非”门时，采用“与非-与非”式较方便；使用“或非”门时，采用“或非-或非”式较方便等等。但是，实际上由于任何一个逻辑函数都可以方便地表示成“与或”式，所以“与或”式是逻辑函数的最基本形式。

（1）由“与或”式变换成“或与”式。