普通高等教育"十一五"规划教材

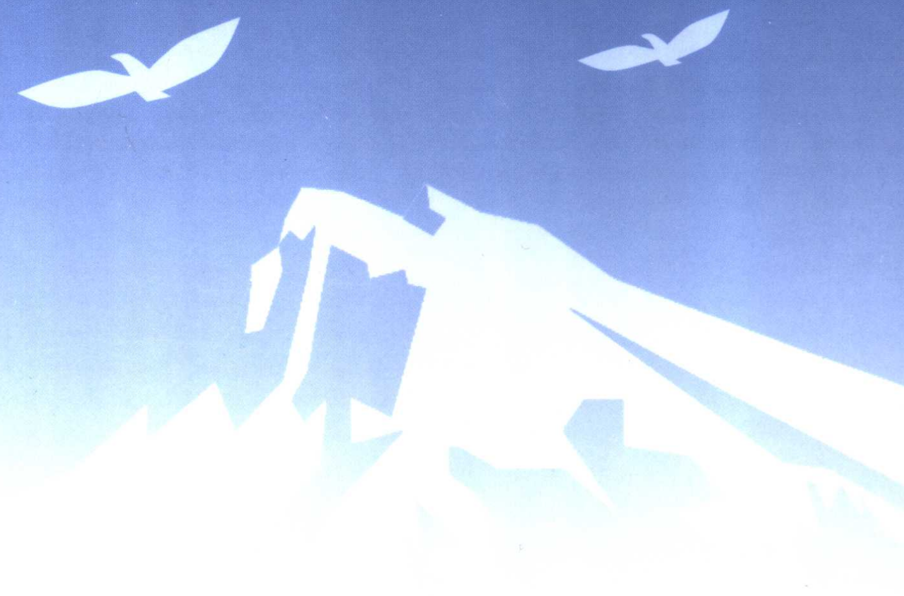
PUTONG GAODENG JIAOYU SHIYIWU GUIHUA JIAOCAI

SHUZI DIANZI JISHU

数字电子技术

朱传琴 主编

方舒燕 王亚盛 副主编



中国电力出版社

http://jc.cepp.com.cn

普通高等教育"十一五"规划教材
PUTONG GAODENG JIAOYU SHIYIWU GUIHUA JIAOCAI

SHUZI DIANZI JISHU

数字电子技术

主　编　朱传琴
副主编　方舒燕　王亚盛
编　写　杨露露　高安芹
主　审　杨霓清

中国电力出版社
http://jc.cepp.com.cn

内 容 提 要

本书为普通高等教育“十一五”规划教材。

全书共分为九章，内容包括数字电路基础、基本逻辑门电路、组合逻辑电路、集成触发器、时序逻辑电路、存储器与可编程逻辑器件、脉冲产生电路、模拟量与数字量的转换、传感器。每章后附有习题，便于读者自学。

本书可作为普通高等学校电气工程及其自动化、自动化、电子信息工程等专业的专业基础课教材，也可作为高职高专及函授教材，同时可作为相关工程技术人员的参考用书。

图书在版编目（CIP）数据

数字电子技术/朱传琴主编. —北京：中国电力出版社，2007

普通高等教育“十一五”规划教材

ISBN 978-7-5083-4880-3

Ⅰ. 数… Ⅱ. 朱… Ⅲ. 数字电路-电子技术-高等学校-教材 Ⅳ. TN79

中国版本图书馆 CIP 数据核字（2006）第 124428 号

中国电力出版社出版、发行

（北京三里河路 6 号 100044 http://jc.cepp.com.cn）

航远印刷有限公司印刷

各地新华书店经售

*

2007 年 1 月第一版 2007 年 1 月北京第一次印刷

787 毫米×1092 毫米 16 开本 14.125 印张 343 千字

印数 0001—3000 册 定价 **22.00** 元

前　言

为贯彻落实教育部《关于进一步加强高等学校本科教学工作的若干意见》和《教育部关于以就业为导向深化高等职业教育改革的若干意见》的精神，加强教材建设，确保教材质量，中国电力教育协会组织制订了普通高等教育“十一五”教材规划。该规划强调适应不同层次、不同类型院校需要，满足学科发展和人才培养的需求，坚持专业基础课教材与教学急需的专业教材并重、新编与修订相结合的原则。本书为新编教材。

本教材有以下几个特点。

（1）遵循以集成电路应用为主、分立电路分析为副的原则，以建立基本概念为目的对分立电路进行分析，原则上以“必须”和“够用”为尺度，删除了某些过时的内容，同时又保持了课程体系的完整性。

（2）增加了集成电路应用电路的内容。本教材增加了电子技术与计算机接口电路的内容，如各种显示驱动电路的应用、A/D转换器和D/A转换器与计算机接口等，为学生学习计算机技术打下良好的基础。

（3）本教材增加了传感器及其应用一章内容，使参加电子设计竞赛的学生正确地将被测物理量与电路联系起来。另外书中还配有大量的例题，以便于学生正确理解课程内容和自学。

全书共分九章，其中第一、四、五章由山东电力高等专科学校朱传琴老师编写，第六、七章由郑州电力高等专科学校方舒燕老师编写，第八、九章由威海职业学院王亚盛老师编写，第二章由山东电力高等专科学校高安芹老师编写，第三章由山东电力高等专科学校杨露露老师编写。山东大学杨霓清教授担任本书的主审，杨霓清教授在百忙中对书稿进行了非常认真的审查，并提出了许多宝贵意见和建议，在此表示衷心的感谢。

本书参考了许多著作与资料，对书后所列参考书籍的各位编者，在此表示深深的感谢。

由于编者水平有限，加上时间有些仓促，难免会有错误和不足之处，恳请读者批评、指正。

联系电话：0531-82999441
E-mail：　zhuchuanqin@sina.com

编　者

目　　录

第一章 数字电路基础

[本章教学基本要求]

掌握：数制与码制的基本概念，与、或、非基本逻辑函数，与非、或非、异或、同或等常用复合逻辑函数，逻辑函数的表达式、真值表、逻辑图、卡诺图等表示形式及相互转换，逻辑代数基本定律和逻辑函数的化简方法。

熟悉：无关项的概念及具有无关项逻辑函数的化简方法。

第一节 概 述

一、数字信号和数字电路

在电子电路中，信号分为两类。一类是模拟信号，其特点是在时间上和数值上都是连续变化的。例如：由温度传感器得到的与温度成正比的电压信号等。传输和处理模拟信号的电路称为模拟电路。另一类是数字信号，这类信号在时间上和数值上都是离散的，如生产中记录产品个数的计数信号等。传输和处理数字信号的电路称为数字电路。

数字电路在现代电子技术中占有十分重要的地位。由于数字电路具有抗干扰能力强、可靠性和准确性高、集成度高、电路设计灵活等优点，因此，它在通信、医学、电视、电子测量、自动控制、电子计算机等领域都得到了广泛的应用。

二、数字电路特点

数字信号是采用 0 和 1 来表示电平的高、低，脉冲的有、无，开关的闭合与断开等。所以，数字电路就是和 0、1 打交道，在电路上是以高电平实现 1，低电平实现 0。与模拟电路相比具有以下特点：

（1）数字电路在稳态时，三极管工作在开关状态，即饱和与截止状态，分别用 0 和 1 来表示。

（2）数字电路研究的主要问题是：输出与输入的逻辑关系，即电路的逻辑功能。所以数字电路也称作逻辑电路。

（3）分析数字电路的主要工具是逻辑代数，描述电路的方式是逻辑表达式、逻辑图、真值表、卡诺图和波形图等。

第二节 数制与二—十进制编码

一、数制

数制就是按某种进位规则进行计数的计数体制。在我们日常生活中广泛使用十进制，在数字电路中使用的却是二进制。因此，我们既要熟悉十进制、二进制，还要熟悉二者之间的相互转换。

1. 十进制

十进制是以 10 为基数的计数体制，由 0、1、2、…、9 十个数码组成，相邻位的关系是逢十进一、借一当十。在十进制数中，位于不同数位上的数码所代表的数值是不同的。如十进制数 333，同样的数码 3，在百位上代表 300，在十位上代表 30，在个位上代表 3。我们把 10^k 称作十进制数的“位权”，$k=0$、1、2、…，个位的“位权”是 10^0，十位的“位权”是 10^1，百位的“位权”是 10^2…，依次类推。数码所处的数位越高，其“位权”越大。

十进制数有十个数码，在电路中要有十个不同的状态来实现。这显然是不现实的，因此在电路中一般都采用二进制。

2. 二进制

二进制是以 2 为基数，每一位二进制数是由 0 和 1 两个数码组成，相邻位的关系是逢二进一、借一当二。其“位权”为 2^k，$k=0$、1、2、…。

3. 八进制

八进制是以 8 为基数，每一位是由 0～7 八个数码组成，低位到高位的进位关系是逢八进一、借一当八。

4. 十六进制

十六进制是以 16 为基数，每一位是由 0～9、A、B、C、D、E、F 十六个数码组成，低位到高位的进位关系是逢十六进一、借一当十六。

人们熟悉十进制，而电路只能识别和实现二进制数，为了便于人机交互，就要熟悉二者之间的相互转换。

5. 二进制与十进制相互转换

(1) 二进制数转换为十进制数

二进制数转换为十进制数，可以采用按“位权”相加的方法，即把一个二进制数中为 1 的系数相应的“位权”相加即可。

【例 1-1】 $(101101)_2=1\times2^5+1\times2^3+1\times2^2+1\times2^0=32+8+4+1=(45)_{10}$

【例 1-2】 $(11001011)_2=1\times2^7+1\times2^6+1\times2^3+1\times2^1+1\times2^0=128+64+8+2+1$
$=(203)_{10}$

(2) 十进制数转换为二进制数

十进制数转换为二进制数可采用除 2 取余法，该方法比较繁杂，读者可参阅其他电子技术书籍。本教材介绍一种分解“位权”的方法，即把一个十进制数分解成二进制数“位权”相加的形式，对应二进制数的“位权”，在相应的位上填 1，缺少的“位权”在相应位上填 0。只要熟记二进制数从低位到高位的“位权”分别为 2^0、2^1、2^2、2^3、2^4、2^5、2^6、2^7、2^8、2^9、2^{10}等，即分别为 1、2、4、8、16、32、64、128、256、512、1024、…。可直接写出转换后的二进制数为

【例 1-3】 $(15)_{10}=8+4+2+1=(2^3+2^2+2^1+2^0)=(1111)_2$

【例 1-4】 $(56)_{10}=32+16+8=(2^5+2^4+2^3)=(111000)_2$

【例 1-5】 $(87)_{10}=64+16+4+2+1=(2^6+2^4+2^2+2^1+2^0)=(1010111)_2$

【例 1-6】 $(135)_{10}=128+4+2+1=(2^7+2^2+2^1+2^0)=(10000111)_2$

在例 1-4 中，缺少二进制数“位权”4、2、1，所以在相应位上填 0。在例 1-5 中，缺少二进制数“位权”32 和 8，在相应位上也填上 0。在例 1-6 中，缺少二进制数“位权”

64、32、16 和 8，在相应位上也填上 0。

二、二—十进制编码

十进制数的 0、1、2、…、9 十个数码，电路是不能识别的。必须先用二进制数码将其表示出来，再用电路实现并进行传输和处理。这种用四位二进制数码表示一位十进制数码的方法称为二—十进制编码，即 BCD 码（Binary Decimal Code）。

四位二进制数码有十六个不同的状态，只需从中选出十个状态，就可以分别表示一位十进制的十个数码。选取方法有多种，对应有多种 BCD 码，其中最常用的是 8421BCD 码和循环码（格雷码）。8421BCD 码编码方式是，将十进制数码的 0、1、2、…、9 十个数码，分别与自然二进制数码一一对应。而循环码则是每相邻的两组码中只有一个数码不同，这样使得状态变化比较平缓。此外还有余 3 码、5421 码、2421 码等。几种常用的 BCD 码如表 1-1 所示。

表 1-1　　8421BCD 编　码

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

由表 1-1 的第二列可看出，四位二进制数码，每位对应的“位权”由高到低分别是 8、4、2、1，因此，称这种码为 8421BCD 码。

8421BCD 码在形式上与四位二进制数相同，但二者是不一样的。

如：$(1001)_{BCD}=(9)_{10}$　　　　$(1001)_2=(9)_{10}$

$(10010101)_{BCD}=(95)_{10}$　　　　$(10010101)_2=(149)_{10}$

由表 1-1 第三列看出，相邻的两个数码的四位二进制码中只有一位二进制码是不同的，而其他三位都是相同的。如：数码 8 和数码 9 对应的四位二进制码分别是 1100 和 1000，只有第三位数码不同，其他三位都相同，再如：数码 9 和数码 0 的四位二进制码分别是 1000 和 0000，同样也只有第四位不同，这就是循环码的由来。

由表 1-1 第四列可知，余 3 码是由四位二进制码十六个组合中的前 3 个组合及后 3 个组合去掉而得到的。所以称作余 3 码。

另外国际上还规定了一些专门用于大小写字母、运算符号等的二进制码，如 ISO 码、ASCII 码等，读者可查阅有关书籍和资料，这里不再进行介绍。

第三节 逻 辑 函 数

逻辑代数也称布尔代数，它是分析和设计逻辑电路的数学工具。逻辑代数和普通代数相类似，也分常量、变量和函数。在逻辑代数中称为逻辑常量、逻辑变量和逻辑函数。下面分别进行介绍。

一、逻辑函数基本概念

1. 逻辑常量

所谓常量就是取值一定、不再变化的量。在普通代数中有无穷多个常量。而在逻辑代数中，只有 0、1 两个逻辑常量。这里的 0 和 1 不表示数值大小，只表示两个对立的逻辑状态。如开关的闭合与断开，电平的高和低，灯亮与灯灭等。我们称它为逻辑 0 和逻辑 1。

2. 逻辑变量

逻辑变量也称逻辑自变量，用 A、B、C、D 等 26 个字母表示，其取值可以自主变化，可以且只能取 0 或取 1，这里的 0 和 1 同样不表示数值的大小，而是表示两种对立的逻辑状态。例如，一个开关可以断开也可以闭合，所以，开关可视为一个逻辑变量，用字母 A 表示。当它闭合时，其值取为 1；当它断开时，其值取为 0。

3. 逻辑函数

与普通代数类似，逻辑函数也称因变量，当决定事物的自变量取值一定时，被决定事物的结果也惟一地确定。这一结果称为逻辑函数或逻辑因变量。当变量取值一定时，函数值也惟一地确定了。逻辑函数值也只有 0 和 1 两个。0 和 1 也是表示事物的两个对立状态，而不表示数值大小。

二、三种基本逻辑函数及其表示方法

在逻辑代数中，逻辑问题是千变万化的，但最基本的逻辑函数有三种，分别是与、或、非三种逻辑函数。

1. 与逻辑函数

在图 1-1 所示的照明电路中，电灯 F 亮与不亮决定于开关 A、B 接通与否。显然，只有当开关 A 和 B 同时闭合时，电灯 F 才能亮。开关 A、B 闭合（条件）和灯亮（结果）之间的这种因果关系，称为与逻辑关系。

若灯亮用 1 表示，不亮用 0 表示，开关闭合用 1 表示，断开用 0 表示，可以列出以真值 0 和 1 表示的与逻辑关系表格，我们称其为真值表，如表 1-2 所示。

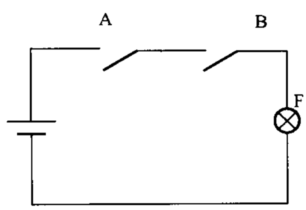
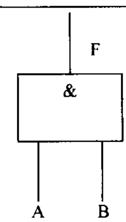


图 1-1 与逻辑电路

表 1-2 与逻辑函数的几种表示方式

真值表		表达式	逻辑符号	逻辑规律
A B	F			
0 0	0	F=AB		有 0 出 0
0 1	0			全 1 出 1
1 0	0			
1 1	1			

真值表是研究数字电路的重要工具。表的左边是变量的所有取值组合，每个变量可以取 0 和 1 两个值，n 个变量则有 2^n 个取值组合。表的右边是在每一组变量取值下，按着与的逻辑关系所确定的函数值。与逻辑函数表达式为：$F=A\cdot B$ 或 $F=AB$，读作 F 等于 A 与 B。式中 A、B 为逻辑变量，取值可以自主变化，但取值只能是 0 或 1。F 是变量 A、B 的函数，也称因变量。当 A、B 的值取定以后，F 的值也惟一地确定了。

与逻辑函数可以推广到 n 个变量 A_1，A_2，A_3，…，A_n　$F=A_1A_2A_3\cdots A_n$。

由与逻辑真值表可以得到与逻辑规律是：有 0 出 0，全 1 出 1。

与逻辑关系在数字电路中用与逻辑门电路来实现，逻辑变量对应逻辑门的输入，逻辑函数对应逻辑门的输出。与的逻辑符号如图 1-2 所示。

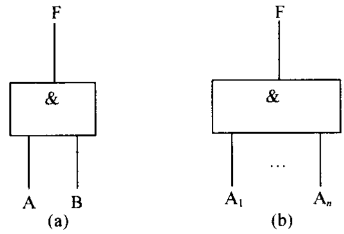


图 1-2　与逻辑符号

(a) 两变量与逻辑符合；(b) 多变量与逻辑符合

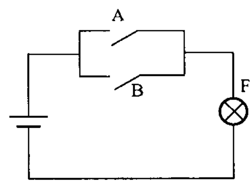


图 1-3　或逻辑电路

2. 或逻辑函数

在图 1-3 中，开关 A 和 B 并联后控制电灯 F。

当开关 A 闭合或者 B 闭合，或者 A 和 B 都闭合时，电灯 F 都会亮。只有当开关 A 和 B 都断开时，灯才不亮。开关 A、B 闭合（条件）与灯亮（结果）之间的这种逻辑关系，称为或逻辑关系。用表达式 $F=A+B$ 表示，读作 F 等于 A 或 B。若灯亮用 1 表示，灯不亮用 0 表示，开关闭合用 1 表示，断开用 0 表示，列出或逻辑函数真值表如表 1-3 所示。或逻辑在电路中用或门电路来实现。或逻辑函数的逻辑符号如图 1-4 所示。或逻辑规律如表 1-3 所示。

或逻辑也可以推广到 n 个变量 A_1，A_2，…，A_n

$$F=A_1+A_2+\cdots+A_n$$

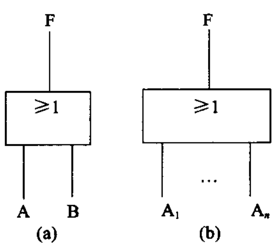
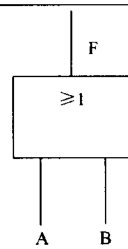


图 1-4　或逻辑符号

(a) 两变量或逻辑符号；

(b) 多变量或逻辑符号

表 1-3　或逻辑函数的几种表示方式

真值表			表达式	逻辑符号	逻辑规律
A	B	F			
0	0	0	F=A+B		有 1 出 1 全 0 出 0
0	1	1			
1	0	1			
1	1	1			

3. 非逻辑函数

在图 1-5 所示电路中，NC 为继电器动断触点。线圈 A 不通电时，NC 是闭合的，电灯亮。当线圈通电时，动断触点断开，电灯不亮。我们把这种逻辑关系称为非逻辑关系。若灯亮用 1 表示，灯不亮用 0 表示，线圈通电用 1 表示，不通电用 0 表示，列出非逻辑的真值表

如表 1-4 所示。非的逻辑表达式写为：$F=\overline{A}$，读作 F 等于 A 非或 A 反。逻辑非的逻辑规律为：进 0 出 1，进 1 出 0。在数字电路中，非逻辑关系用非门电路来实现，非逻辑符号如图 1-6 所示，输出端的圆圈表示求非。

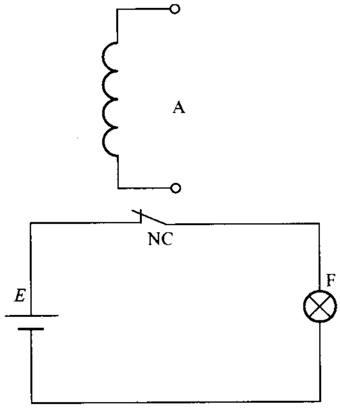


图 1-5　非逻辑电路

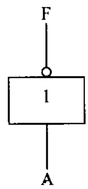
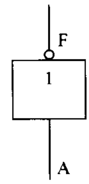


图 1-6　非逻辑符号

表 1-4　非逻辑函数的几种表示方式

真值表		表达式	逻辑符号	逻辑规律
A	F			
0 1	1 0	$F=\overline{A}$		进 1 出 0 进 0 出 1

三、常用复合逻辑函数

实际的逻辑关系往往要比单纯的与、或、非三种逻辑关系复杂得多。但不管有多复杂，总能用与、或、非的组合来实现。含有两种或两种以上的逻辑运算，称为复合逻辑运算，也称复合逻辑函数。最常用的复合逻辑函数有与非、或非、异或和同或等逻辑函数。目前，人们已经做出了实现这些复合逻辑函数的相应门电路，分别为与非门电路、或非门电路、异或门电路和同或门电路等。下面分别介绍它们的逻辑功能和逻辑符号。

1. 与非逻辑函数

与非逻辑函数是“与”和“非”组合成的复合逻辑函数，其运算顺序是先与后非。由与的逻辑规律和非的逻辑规律，可得到与非的逻辑规律是：有 0 出 1，全 1 出 0。与非的表达式是：$F=\overline{AB}$。真值表及逻辑符号见表 1-5。

表 1-5　与非逻辑函数的几种表示方式

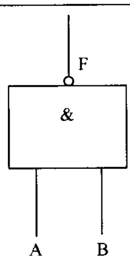
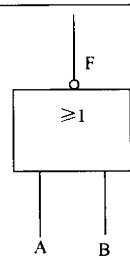
真值表		表达式	逻辑符号	逻辑规律
A　B	F			
0　0 0　1 1　0 1　1	1 1 1 0	$F=\overline{AB}$		有 0 出 1 全 1 出 0

表 1-6　或非逻辑函数的几种表示方式

真值表		表达式	逻辑符号	逻辑规律
A　B	F			
0　0 0　1 1　0 1　1	1 0 0 0	$F=\overline{A+B}$		有 1 出 0 全 0 出 1

2. 或非逻辑函数

或非逻辑函数是或逻辑和非逻辑组合而成的复合逻辑函数。其运算顺序是，先或后非。其真值表、表达式、逻辑符号和逻辑规律见表 1-6。

3. 异或逻辑函数

异或逻辑函数的定义是：两个输入变量取值不同（相异）时，函数值是 1，两个输入变

量取值相同时，函数值是0。异或逻辑函数在数字电路中是十分有用的复合逻辑函数，它可以使复杂的逻辑关系简单化。异或函数的真值表、表达式、逻辑符号及逻辑规律如表1-7所示。

表1-7 异或逻辑函数的几种表示方式

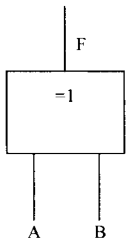
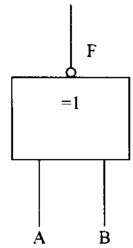
真值表		表达式	逻辑符号	逻辑规律
A B	F	$F=A\oplus B$ $=\overline{A}B+A\overline{B}$	 	相异出1 相同出0
0 0	0			
0 1	1			
1 0	1			
1 1	0			

表1-8 同或逻辑函数的几种表示方式

真值表		表达式	逻辑符号	逻辑规律
A B	F	$F=A\odot B$ $=AB+\overline{A}\cdot\overline{B}$	 	相同出1 不同出0
0 0	1			
0 1	0			
1 0	0			
1 1	1			

4. 同或逻辑函数

同或逻辑函数的定义是：两个输入变量取值相同时，函数值是1，两个输入变量取值不同时，函数值是0。从定义可知，同或逻辑函数和异或逻辑函数是相反的，它们互为反函数。同或逻辑函数的真值表、表达式、逻辑符号和逻辑规律如表1-8所示。

说明：异或和同或逻辑函数都只有两个输入变量，而与、或、与非、或非都可以是多输入变量。

读者可自行证明：$A\oplus 0=A$

$A\oplus 1=\overline{A}$

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

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

$A\odot B\odot C=A\oplus B\oplus C$

第四节 逻辑函数的几种表示形式及其相互转换

对前面所讨论的与、或、非三种基本逻辑函数以及与非、或非、异或、同或四种常用的复合逻辑函数，因为都已经有相应的逻辑门电路，都有各自的逻辑符号，所以都可以用真值表、表达式、逻辑符号表示。而对一般逻辑函数，又是由以上七种逻辑函数中的某几种函数组合起来的，除了用真值表、表达式表示外，还可以用逻辑图表示，另外还可以用卡诺图来表示，这将在以后进一步讨论。

一、逻辑函数几种表示形式

1. 逻辑函数表达式

逻辑函数表达式，就是把逻辑函数（输出）和变量（输入）之间的逻辑关系，写成与、或、非、与非、或非、异或、同或等运算的组合式，即逻辑函数表达式。

例如：$F=\overline{AB}+A\overline{C}+\overline{B+C}$是由非、与非、或非、与以及或等组合起来的三变量逻辑函数表达式。

在逻辑运算中的运算顺序分别是：①单变量求非；②括号；③与；④或。

例如：在 F=（A+B）（A+C）中，第一级运算是加括号的 A 或 B 以及 A 或 C，第二级运算是两个或运算的结果进行与运算。有些特殊情况括号可以省略，如$\overline{(A+B)}C$可以写为$\overline{A+B}C$。

2. 逻辑函数真值表

用真值表表示逻辑函数，是一种直观的表格表示方法。表 1-9 是函数 $F=A\overline{B}+\overline{A}C+\overline{B}\cdot\overline{C}$ 的真值表。

表 1-9　$F=A\overline{B}+\overline{A}C+\overline{B}\cdot\overline{C}$ 的真值表

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

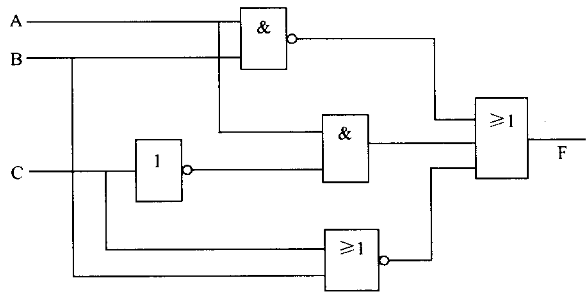


图 1-7　$F=\overline{AB}+A\overline{C}+\overline{B+C}$的逻辑图

3. 逻辑图

逻辑图就是用与、或、非、与非、或非等逻辑符号按照逻辑函数中运算顺序组合起来的图形，如图 1-7 所示就是逻辑函数 $F=\overline{AB}+A\,\overline{C}+\overline{B+C}$的逻辑图。

表达式、真值表、逻辑图是同一个逻辑函数的不同表示形式，在逻辑上是等价的，所以三者之间可以互换。

二、逻辑函数各种表示形式的相互转换

逻辑函数各种形式的转换如图 1-8 所示。

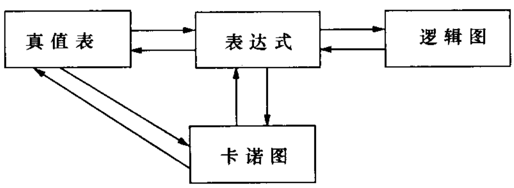


图 1-8　逻辑函数形式转换示意图

从框图图 1-8 可看出，只有表达式可以直接转换为真值表、卡诺图和逻辑图，而真值表和卡诺图不能直接转换为逻辑图，需要经表达式转换。下面介绍表达式、真值表和逻辑图的转换方法，有关卡诺图的转换在后面再予以讨论。

1. 已知表达式求真值表

设逻辑函数有 n 个变量，每个变量可取 0 或 1 两个值。n 个变量有 2^n 个取值组合。为了不重复、不遗漏，可以把 n 个变量当作 n 位二进制数，由全取 0 开始，按每次加 1 的递增顺序，直到全取 1 为止，列出 n 个变量所有取值组合，然后将每一组变量取值组合，代入函数表达式中，算出函数值，填入对应的行中，便可得到该逻辑函数的真值表。

【例 1-7】　试列出 $F=A\overline{B}+AC+\overline{B}\cdot\overline{C}$ 的真值表。

解　由表达式可知，这是三变量的逻辑函数。变量取值组合从 000 开始，逐次加 1，直

到 111 共有 $2^3=8$ 项。算出每一项的函数值，得到 $F=A\overline{B}+AC+\overline{B}\cdot\overline{C}$ 的真值表如表 1-10 所示。

2. 已知真值表写表达式

已知真值表写表达式的方法是：先找出函数值为 1 的项，对每一项写出一个由输入变量为因子的乘积项。在这些乘积项中，当变量取 1 时，以其原变量为因子，变量取值为 0 时，以其反变量为因子，然后将这些乘积项相加即可得到相应的逻辑函数表达式。

表 1-10　$F=A\overline{B}+AC+\overline{B}\cdot\overline{C}$ 真值表

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

表 1-11　例 1-8 真值表

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

【例 1-8】 写出表 1-11 所示逻辑函数真值表对应的表达式。

解 由表 1-11 可见函数值为 1 的项共有三项，所以表达式有三个乘积项相加。第一项对应 A=0、B=1、C=0，在乘积项中以 $\overline{A}$、B、$\overline{C}$ 为因子写出乘积项 $\overline{A}B\overline{C}$、第二项的乘积项为 $AB\overline{C}$、第三项的乘积项为 ABC，得逻辑函数表达式 $F=\overline{A}B\overline{C}+AB\overline{C}+ABC$。

3. 已知表达式画逻辑图

由表达式画逻辑图的方法是，用逻辑符号代替表达式中的逻辑运算符号，并根据运算顺序，将这些逻辑符号一一连接起来，即可得到该逻辑函数的逻辑图。

【例 1-9】 画出 $F=AB+\overline{B}C+\overline{AC}$ 的逻辑图。

解 在逻辑表达式中，第一级是非运算，第二级顺序是与和与非运算，第三级运算顺序是或，由此可画出对应的逻辑图，如图 1-9 所示。

【例 1-10】 画出 $F=\overline{\overline{AB}\cdot\overline{AC}\cdot\overline{BC}}$ 的逻辑图。

解 第一级是与非运算，分别是 $\overline{AB}$、$\overline{AC}$、$\overline{BC}$，第二级还是与非运算。画出逻辑图，如图 1-10 所示。

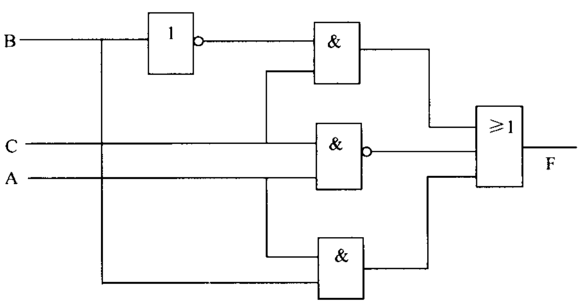


图 1-9　$F=AB+\overline{B}C+\overline{AC}$ 逻辑图

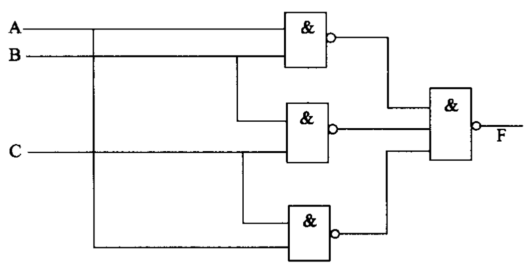


图 1-10　$F=\overline{\overline{AB}\cdot\overline{AC}\cdot\overline{BC}}$ 的逻辑图

4. 已知逻辑图写逻辑函数表达式

根据给定的逻辑图，由输入级开始，写出每个逻辑符号对应的逻辑表达式，便可以得到

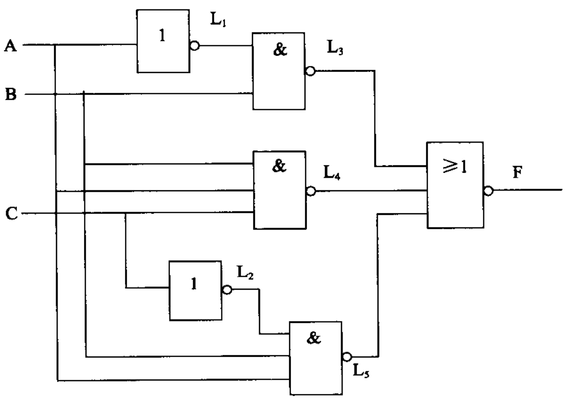


图 1-11　例 1-11 的逻辑图

该逻辑图的逻辑函数表达式。

【例 1-11】 已知逻辑图如图 1-11 所示，试写出逻辑函数表达式。

解 第一级 L_1 和 L_2 为非运算，第二级 L_3、L_4、L_5 为与非运算，最后一级为或非运算，从输入到输出每级的表达式分别为：

$$L_1=\overline{A},\ L_2=\overline{C},\ L_3=\overline{\overline{A}B}$$

$$L_4=\overline{ABC},\ L_5=\overline{AB\overline{C}}$$

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

第五节　逻辑代数基础

一、逻辑代数基本定律

1. 0-1 律

$1\cdot A=A$；　　　　$0+A=A$。

$0\cdot A=0$；　　　　$1+A=1$。

2. 交换律

$A\cdot B=B\cdot A$；　　　　$A+B=B+A$。

3. 结合律

$A\cdot(B\cdot C)=(A\cdot B)\cdot C$；　　　　$A+(B+C)=(A+B)+C$。

4. 分配律

$A\cdot(B+C)=A\cdot B+A\cdot C$；　　　　$A+(B\cdot C)=(A+B)\cdot(A+C)$。

5. 互补律

$A\cdot\overline{A}=0$；　　　　$A+\overline{A}=1$。

6. 重叠律

$A\cdot A=A$；　　　　$A+A=A$。

7. 还原律

$\overline{\overline{A}}=A$。

8. 反演律

$\overline{ABC}=\overline{A}+\overline{B}+\overline{C}$；　　　　$\overline{A+B+C}=\overline{A}\cdot\overline{B}\cdot\overline{C}$。

9. 吸收律

$A+AB=A$；　　　　$A(A+B)=A$。

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

在以上定律中，以分号为界分为两部分。分号后面的和分号前面的公式互为对偶式，只要证明分号前面的公式是正确的，根据后面要介绍的对偶规则，分号后面的公式也是正确的。

在以上公式中，有些和普通代数相同，有些不同，而且这些公式在化简和转换逻辑函数

时非常有用。要熟记这些公式。这些公式是否成立，可用下列方法证明。

二、基本定律的证明方法

1. 真值表证明法

对以上所列公式，最有效的证明方法，是检验等式左边逻辑函数的真值表，与等式右边逻辑函数的真值表是否相等。

【例 1-12】 证明反演律$\overline{ABC}=\overline{A}+\overline{B}+\overline{C}$的正确性。

解 由于是三变量真值表，有八种取值组合，如表 1-12 所示。右边两行分别是$\overline{ABC}$和$\overline{A}+\overline{B}+\overline{C}$的值。由真值表可以看出，等号两边的真值表是相同的，从而证明等式是成立的。

表 1-12　例 1-12 的真值表

A	B	C	$\overline{ABC}$	$\overline{A}+\overline{B}+\overline{C}$
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

2. 代数法证明

证明逻辑代数中的基本定律是否成立，还可以用代数法，就是用已经证明过了的公式证明未证明过的等式。

【例 1-13】 用代数法证明吸收律 $A+AB=A$ 的正确性（设其他公式都已用真值表法得到了证明）。

证明 $A+AB=A\ (1+B)$（用分配律和 0-1 律）

$=A\cdot 1$　　（用 0-1 律）

$=A$　　（用 0-1 律）

【例 1-14】 用代数法证明吸收律 $A+\overline{A}B=A+B$

证明 左边 $A+\overline{A}B$

$=A+AB+\overline{A}B$　　（用吸收律）

$=A+B\ (A+\overline{A})$　　（用分配律）

$=A+B$　　（用互补律）

等于右边

三、逻辑函数的三个基本规则

1. 代入规则

代入规则是：在等式中，用一个表达式代替等式中的某一变量，得到一个新的等式，新的等式也是成立的。该规则的应用，可以将前面的基本定律扩展出更多的公式。

例如，前面已证明 $A+\overline{A}B=A+B$ 是成立的，利用代入规则可知，$AB+\overline{AB}C=AB+C$ 以及 $\overline{A}+AB=\overline{A}+B$ 也都是成立的。

2. 对偶规则

对偶规则是：将原函数式 $F=f\ (A,\ B,\ C,\ \cdots)$ 中的“·”变成“+”，“+”变成“·”，0 变成 1，1 变成 0，所有变量保持不变，得到一个新的逻辑函数式 F'，则 F' 是原函数式 F 的对偶式，或者说 F 和 F' 互为对偶。

利用对偶规则，可使基本定律的公式证明减少一半。因为某一等式成立，其对偶式也成立。

运用对偶规则要注意以下几点。

（1）求对偶式时，原来的运算顺序保持不变，原来先运算的仍然要先运算。

如：$F=\overline{A}\overline{B}+A\ (C+D)$ 的对偶式：$F'=(A+\overline{B})\cdot(A+CD)$。