SHUZI DIANZI JISHU

数字电子技术

主编 林春方

“十一五”高职高专教材

安徽大学出版社

SHUZI DIANZI JISHU

数字电子技术

主编 林春方

“十一五”高职高专教材

安徽大学出版社

图书在版编目(CIP)数据

数字电子技术/林春方主编.一合肥:安徽大学出版社,2006.5

"十一五"高职高专教材

ISBN 7—81110—123—8

Ⅰ.数… Ⅱ.林… Ⅲ.数字电路—电子技术—高等学校:技术学校—教材 Ⅳ.TN79

中国版本图书馆CIP数据核字(2006)第039527号

内容简介

本书紧密结合高职高专教育特点,主动适应社会实际需要,突出应用性、针对性,强调学生实践能力的培养。在内容上,以应用为目的,以"必需"和"够用"为度,以讲清概念、强化应用为重点,深入浅出地阐述了数字集成电路的基本工作原理和逻辑功能,突出了中、大规模集成电路的应用。

本书主要内容包括:数字电路基础、逻辑门电路、组合逻辑电路、集成触发器、时序逻辑电路、脉冲信号的产生及波形变换、数模转换器和模数转换器以及大规模集成电路介绍、数字电路实训等,并介绍了EWB仿真软件在数字电路中的应用。

本书可作为高等职业院校、高等专科院校、成人高校、民办高校及本科院校举办的二级职业技术学院电子信息类、通信类及相关专业的教学用书,也适用于五年制高职、中职相关专业,并可作为社会从业人士的业务参考书及培训用书。

"十一五"高职高专教材

数 字 电 子 技 术

林春方 **主编**

出版发行	安徽大学出版社 (合肥市肥西路3号 邮编230039)	**印 刷**	合肥中德印刷培训中心印刷厂
联系电话	编辑室 0551-5108498 发行部 0551-5107784	**开 本**	787×1092 1/16
		印 张	12.25
		字 数	298千
责任编辑	朱丽琴	**版 次**	2006年5月第1版
封面设计	张 莉	**印 次**	2006年5月第1次印刷
E—mail	zlqemail@tom.com		

ISBN7 81110-123-8 / T·92 **定价** 18.50元

如有影响阅读的印装质量问题,请与出版社发行部联系调换

前言

随着数字电子技术的发展，新器件、新知识、新工艺在数字电子技术方面得到广泛的应用，结合职业教育的特点，教材编写力求更新教学观念和内容，在保证基本概念、基本原理和基本分析及设计方法的前提下，简化集成电路的内容、结构和工作原理的讲述，减少小规模集成电路的内容，尽可能多地介绍新型大规模集成电路及其应用。

本书紧密结合高职高专教育特点，主动适应社会实际需要，突出应用性、针对性，加强实践能力的培养。在内容上，以能力培养为主线，以应用为目的，突出思路与方法的阐述，以“必需”和“够用”为度，以讲清概念、强化应用为重点，深入浅出地阐述了数字集成电路的基本工作原理和逻辑功能，突出了中、大规模集成电路的应用。

本书主要内容包括：数字电路基础、逻辑门电路、组合逻辑电路、集成触发器、时序逻辑电路、脉冲信号的产生及波形变换、数模转换器和模数转换器以及大规模集成电路介绍、数字电路实训等，并介绍了 EWB 仿真软件在数字电路中的应用。

本书由安徽电子信息职业技术学院林春方老师担任主编，并编写第 7 章；淮南联合大学朱玲老师编写第 1 章；宿州职业技术学院韩军老师编写第 2 章和第 6 章；阜阳职业技术学院李华彬老师编写第 3 章；芜湖职业技术学院王苹老师编写第 4 章；万博科技职业学院李美莲老师编写第 5 章；宣城职业技术学院谢金忠老师编写第 8 章；安徽电子信息职业技术学院方庆山老师编写第 9 章。安徽大学吴小培教授审阅全部书稿，并提出了许多宝贵的意见和建议。在编写过程中，还得到了安徽大学出版社以及安徽电子信息职业技术学院的领导和老师们的大力支持，在此一并表示衷心的谢意。

由于编者水平有限，疏漏之处在所难免，恳请读者批评指正。

编者

2006.5

目 录

第1章 数字电路基础

□学习目标

1. 了解数字电路的基本特点。
2. 掌握常用数制及其相互转换。
3. 掌握逻辑代数的基本规则及逻辑函数的化简方法。

1.1 数字电路概述

1.1.1 数字信号和数字电路

在自然界中有各种各样的物理量，它们的性质各不相同，就其变化的规律而言，可以分为两大类：数字量和模拟量。

数字量的变化在时间上和幅值上都是离散的，也就是说，数字量不是随时间连续变化的，它们的变化只是发生在一系列离散的时刻上，其数值大小和数值的变化都是某一个最小数量单位（分辨率）的整数倍，例如：流水线上产品的个数。表示数字量的信号称为数字信号，传输、处理数字信号的电路称为数字电路。

模拟量的变化在时间上和幅值上都是连续的，可以在一定范围内任意取值。例如：工业控制系统中常见的几大参数：温度、压力、流量、速度等。表示模拟量的信号称为模拟信号，传输、处理模拟信号的电路称为模拟电路。

1.1.2 数字电路的特点

数字电路处理的数字信号一般用二进制表示，每位二进制只有"0"和"1"两个基本数字代码，可用"0"和"1"来表示两个相对的状态，如脉冲的有、无或电平的高、低。例如：若用"1"表示高电平，则"0"就表示低电平；若用"0"表示高电平，则"1"就表示低电平，"0"和"1"在这里只表示两种对立的状态，没有数值上的大小，这是数字电子技术中首先要建立的重要概念：状态。具体用器件来实现就是利用二极管、三极管等元器件的开关特性，如二极管、三极管完全导通表示一种状态，完全截止表示相对立的另一种状态。数字电路在电路结构、研究内容、分析方法等方面都和模拟电路不同，它具有如下特点：

一是，数字电路中的晶体管通常工作在饱和或截止状态，即开关状态，对应于两种对立的状态，用"0"和"1"来表示。在实际工作时，对元件的参数要求不太严格，只要能工作于饱和或截止状态，从而能确切地区分出高、低电平即可。高、低电平都有一个允许的变化范围，只有当外来干扰相当强烈时，超出了允许高、低电平的范围，才有可能改变元件

的工作状态，所以数字电路的抗干扰能力较强。

二是，无论多么复杂的数字电路都是由几种最基本的单元电路组成的，这些最基本单元电路，其结构比较简单，对元器件的精度要求不高。因此，数字电路便于集成化和系列化生产。

三是，在数字电路中，人们重点研究的是电路的输入和输出状态之间的逻辑关系，从而进一步分析电路的逻辑功能。数字电路的研究内容可以分为两大类：一类是对现有的电路分析其逻辑功能，叫做逻辑分析；另一类是根据实际需求，通过分析，设计出满足要求的逻辑电路，称为逻辑设计。

1.1.3 数字电路的分类

按照不同的分类方法，数字电路有不同的类别，常见的有以下几种分类方式：

1. 按电路组成结构分有分立元件和集成电路两类。实际应用中以集成电路为主，根据集成度不同，集成电路又可分为小规模（SSI）、中规模（MSI）、大规模（LSI）和超大规模（VLSI）集成电路。

2. 按电路所用基本器件的种类可以把数字电路分为双极型和单极型电路。双极型电路如 TTL、ECL 和 HTL 等；单极型电路如 NMOS、PMOS 和 CMOS 等。

3. 按电路的结构和逻辑功能可分为组合逻辑电路和时序逻辑电路。组合逻辑电路的输出状态只取决于当前各输入状态的组合，与先前的状态无关，即无记忆性；时序逻辑电路的输出不仅和当前的输入状态有关，而且还与系统的前一状态有关，即具有记忆性。

1.2 数制和编码

1.2.1 几种常用数制

在日常生活和工作中人们都习惯用十进制数，而在数字电路中常采用二进制。这里先对十进制进行讨论。

1. 十进制

十进制数有以下两个特点：

(1) 十进制数共有 0、1、2、3、4、5、6、7、8、9 十个不同的数码，通常把这些数码的个数称为基数，所以十进制数的基数为 10；

(2)低位向相邻高位的进位规则是"逢十进一"。

相同的数码处于不同的位置，具有不同的值，每个数位的数值是由该位数码的值乘以处在这位上的一个固定常数。不同数位上的固定常数称为位权。就整数而言，右起第 1 位（个位）的权为 10^0，第 2 位（十位）的权为 10^1，依次类推。任意一个十进制数都可以表示为权的展开式，设该十进制数包含 n 位整数和 m 位小数，则

$$
\begin{aligned}
(D)_{10} = & a_{n-1}\times 10^{n-1} + a_{n-2}\times 10^{n-2} + \cdots + a_1\times 10^1 + a_0\times 10^0 + \\
& a_{-1}\times 10^{-1} + a_{-2}\times 10^{-2} + \cdots + a_{-m}\times 10^{-m} = \\
& \sum_{i=-m}^{n-1} a_i \times 10^i
\end{aligned}
\tag{1-1}
$$

其中 a_i 为第 i 位的系数，可以是 0～9 基本数码中的任何一个。

例如：

$$473.56=4\times10^2+7\times10^1+3\times10^0+5\times10^{-1}+6\times10^{-2}$$

若用 N 代替式(1-1)中的 10，类推可以得到 N 进制数展开式的一般形式：

$$(D)_N=\sum a_i\times N^i \tag{1-2}$$

N 称为进制数的基数；a_i 为第 i 位的系数；N^i 称为第 i 位的权。

2. 二进制

二进制数的基数为 2，每一位仅有 0 和 1 两个数码；低位和相邻高位之间的进位规则是"逢二进一"。按式(1-2)，任何一个二进制数都可以展开为

$$(D)_2=\sum a_i\times 2^i \tag{1-3}$$

例如：

$$(101.01)_2=1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}$$

二进制数每个数位上只有 0 和 1 两个数码，算术运算规则相对于十进制数来说要简单得多，相应的运算控制电路也简单。

二进制加法规则

$$0+0=0;1+0=0+1=1;1+1=10(逢二向高位进一)$$

二进制减法规则

$$0-0=0;1-0=1;1-1=0;10-1=1(向高位借 1)$$

二进制乘法规则

$$0\times0=0;0\times1=1\times0=0;1\times1=1$$

二进制除法规则

$$0\div1=0;1\div1=1$$

二进制数虽然有其自身的优点，但也有缺点。用二进制表示一个数时，往往位数过多，不便于读写与记忆，人们使用起来很不方便。所以，为了便于读写，在数字系统中，特别是在计算机系统常采用八进制和十六进制。

3. 八进制

八进制数的特点：

(1)八进制数每一位可以有 0、1、2、3、4、5、6、7 八个不同的数码，基数为 8；

(2)低位向相邻高位的进位规则是"逢八进一"。

任何一个八进制数都可以按式(1-2)展开为：

$$(D)_8=\sum a_i\times 8^i \tag{1-4}$$

例如：

$$(65.37)_8=6\times8^1+5\times8^0+3\times8^{-1}+7\times8^{-2}$$

4. 十六进制数

十六进制数每一位可以有 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 十六个不同的数码，其中符号 A～F 分别相当于十进制中的 10～15；低位向相邻高位的进位规则是"逢十六进一"，按式(1-2)，任意一个十六进制数可以展开为：

$$(D)_{16} = \sum a_i \times 16^i \tag{1-5}$$

例如：

$$(5EA.16)_{16} = 5\times16^2 + E\times16^1 + A\times16^0 + 1\times16^{-1} + 6\times16^{-2} = 5\times16^2 + 14\times16^1 + 10\times16^0 + 1\times16^{-1} + 6\times16^{-2}$$

各种不同的数制各有其优缺点，应用的场合各不相同。可以把其推广为任意进制 N，相应的内容请读者自己思考。

1.2.2 数制之间的相互转换

同一个数可以用二进制或十进制等不同形式来表达，它们之间就必然存在着一定的关系，可以相互转换。

1. 非十进制转换为十进制

将非十进制数写成按权展开的形式，相加的结果就是与之对应的十进制数。

【例 1-1】 $(11011.11)_2 = 1\times2^4 + 1\times2^3 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2} = 27.75$

【例 1-2】 $(147.2)_8 = 1\times8^2 + 4\times8^1 + 7\times8^0 + 2\times8^{-1} = 103.25$

【例 1-3】 $(4E7)_{16} = 4\times16^2 + 14\times16^1 + 7\times16^0 = 1255$

2. 十进制数转换为非十进制

(1)十进制转换为二进制

十进制整数转换为二进制整数一般采用"除 2 取余"法。"除 2 取余"法则是用二进制的基数 2 连续去除待转换的十进制数，直到商为 0，每次除得的余数作为要转换数的系数，先得到的余数为转换的低位，后得到的为高位。

十进制小数转换成二进制小数通常采用"乘 2 取整"法。"乘 2 取整"法则是用二进制基数 2 连续去乘待转换的十进制小数，直到所得的积小数部分为 0 或满足所给的精度为止，然后把每次得到的整数按先得到的为高位，最后得到的为最低位，依次按顺序排列起来所对应的便是所要求的二进制小数。

【例 1-4】 将十进制数 57.125 转换二进制数。

第一步：整数部分转换

		余数
2	57	1
2	28	0
2	14	0
2	7	1
2	3	1
	1	1

结果：$(57)_D = (111001)_B$

第二步：小数部分转换

	整数
0.125 × 2	
0.25	0
0.25 × 2	
0.5	0
0.5 × 2	
1	1

结果：$(0.125)_D = (0.001)_B$

第三步：合并

$$(57.125)_D = (111001.001)_B$$

值得注意的是，在十进制小数转换成二进制小数时，有可能会出现无限位。在这种情况下，一般可以根据精度要求，取有限位即可。

(2)十进制转换为十六进制

把十进制转换为二进制的规则加以推广，可得到十进制转换为任意进制的一般规律：将十进制数的整数部分和小数部分分开，分别用"除基取余"法和"乘基取整"法化成

相应进制的整数部分和小数部分，然后合在一起。这里所说的“基”是指待转换的任意进制的基数，例如：转换成二进制，相应的基数就是 2；转换成八进制，相应的基数就是 8；转换成 16 进制，相应的基数就是 16。

【例 1-5】 把十进制 1002.45 转换十六进制数。

解：将 1002.45 分为整数和小数两部分转换：

第一步：整数部分转换

		余数
16	1002	10
16	62	14
	3	3

结果：$(1002)_D = (3EA)_H$

第二步：小数部分转换

	整数
0.45 × 16 = 7.2	7
0.2 × 16 = 3.2	3
0.2 × 16 = 3.2	3

结果：$(0.45)_D = (0.733)_H$

第三步：合并

$(1002.45)_D = (3EA.733\cdots)_H$

3. 二进制与八进制、十六进制之间的相互转换

(1)二进制与八进制之间的相互转换

三位二进制数有八种不同组合，可以表示 8 种不同状态。一位八进制数有 8 个数码，也可以表示 8 种不同的状态，因此，可以把一位八进制数看作是三位二进制数的缩写。

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

【例 1-6】 将$(1011011.00101011)_B$转换成八进制数。

解：

$$(1011011.00101011)_B =$$
$$(1\quad 011\quad 011.001\quad 010\quad 110)_B =$$
$$(1\quad 3\quad 3.1\quad 2\quad 6)_O$$

根据同样的原理，把八进制数转换成相应的二进制数是上面转换过程的逆过程。

【例 1-7】 将八进制数 75.23 转换成相应的二进制数。

解：

八进制数：	7	5.2	3
二进制数：	111	101.010	011

所以

$$(75.23)_O = (111101.010011)_B$$

(2)二进制与十六进制之间的相互转换

四位二进制数有 16 种不同的组态。1 位十六进制数有 16 个不同的数码，因此，也可以把一位十六进制数看作是四位二进制数的缩写形式，即四位二进制数可以表示一位十六进制数，一位十六进制数也可以用四位二进制数来表示，它们之间的转换与二进制和

八进制之间的转换相类似。

【例 1-8】 将十六进制数 A057.62 转换为二进制数。

解：

十六进制	A	0	5	7.6	2
二进制	1010	0000	0101	0111.0110	0010

所以

$$(A057.62)_H = (1010000001010111.0110001)_B$$

【例 1-9】 将二进制数 1011110 转换成十六进制数。

解：

二进制	0101	1110
十六进制	5	E

所以

$$(1011110)_B = (5E)_H$$

1.2.3 BCD 编码

在数字电路和计算机内部处理的数据信息中，通常都是用二进制数表示的。而实际上需要处理的不仅有二进制数、十进制数等，而且还有各种文字符号。为了解决这一矛盾，人们通常用若干位二进制数按一定的组合方式来表示数和字符等信息，这就是编码。用四位二进制数表示十进制数，叫做二一十进制编码，简称 BCD(Binary Coded Decimal)编码。

BCD 编码的方式有很多种，常分为有权 BCD 码和无权 BCD 码。

1. 有权 BCD 码

四位二进制数共有 $2^4=16$ 种状态，根据取舍的不同，有权 BCD 码有多种形式，表 1-1 给出了有权 BCD 码的常见几种形式。有权 BCD 码是按位权值来命名的，其中 8421BCD 码是一种最简单、最常用的 BCD 码，它的位权值与二进制正好一致，常称为自然权码。任何一个十进制数要写成 BCD 码表示时，只要把它按位转换成相应的 BCD 码即可。

表 1-1 常见 BCD 码与十进制数的对应关系

十进制数	8421BCD 码	2421(A)码	2421(B)码	5421 码	余 3 码	格雷码 0
0	0000	0000	0000	0000	0011	0000
1	0001	0001	0001	0001	0100	0001
2	0010	0010	0010	0010	0101	0011
3	0011	0011	0011	0011	0110	0010
4	0100	0100	1010	0100	0111	0110
5	0101	0101	1011	1000	1000	0111
6	0110	0110	1100	1001	1001	0101
7	0111	0111	1101	1010	1010	0100
8	1000	1110	1110	1011	1011	1100
9	1001	1111	1111	1100	1100	1000
权	8421	2421	2421	5421	无权	无权

2. 无权BCD码

表1-1中的余3码和格雷码是两种常见的无权BCD码。无权码没有确定的位权值，但都有各自的特点。例如，余3码是在8421码的基础上加3后得到的，在数字系统中利用余3码进行加、减运算特别方便。格雷码的特点是相邻的两个码之间仅有一位不同。在实用中，当模拟量发生微小变化可能引起数字量发生变化时，格雷码仅改变一位，这样更可靠，减少出错。另外由于某些原因信息代码在形成、存储和传递过程中很可能会出现错误，为了提高可靠性，往往采用格雷码。

3. ASCII码

ASCII码是在计算机中普遍采用的美国标准信息交换码(American Standard Code for Information Interchange)。ASCII码由7位二进制数码构成，可以为128个字符编码。其中0～9的ASCII码是30～39H，26个大写英文字母A～Z的ASCII码是41H～5AH，小写英文字母a～z的ASCII码是61H～7AH，完整的ASCII码表可参考其他资料。

1.3 逻辑代数基础

逻辑代数是英国数学家乔治·布尔(George Boole)于19世纪中期提出的，因此又称布尔代数，它是研究逻辑电路的数学工具，为分析和设计逻辑电路提供了理论基础。

1.3.1 逻辑变量和基本逻辑运算

1. 逻辑变量

逻辑代数中的变量叫做逻辑变量，逻辑变量通常用字母A、B、C、…、X、Y、Z等来表示，逻辑变量的取值只有两个：1和0。在这里已完全没有数的大小概念，“0”和“1”只表示两种相对立的状态，如真和假、开和关、高和低、是和非等。

在逻辑电路中，电位的高和低可用逻辑1和逻辑0分别表示，这样就相应形成两种系统——正逻辑和负逻辑，通常规定如下：

正逻辑：高电平表示有信号，并用1表示；低电平表示无信号，用0表示。

负逻辑：低电平表示有信号，并用1表示；高电平表示无信号，用0表示。

对于数字电路或系统，可以采用正逻辑，也可采用负逻辑。需要注意的是，同一个电路采用正逻辑或负逻辑，实现的逻辑功能可能不相同。在本书中如不特殊说明，一律采用正逻辑。

2. 基本逻辑运算

基本的逻辑关系有三种：与、或、非。逻辑代数中与之相对应的基本逻辑运算也有三种：与运算、或运算和非运算。

(1)与运算

在图1-1所示的电路中，假设开关闭合用1表示，断开用0表示，灯亮用1表示，灯灭用0表示。很容易推出，该电路只有当开关A和B都闭合时，灯才会亮。对这个简单的实例加以总结可得出与逻辑关系定义：当决定一件事情的各个条件全部具备时，这件事才发生，这种逻辑关系称为与逻辑。用逻辑表达式描述，可以写作：$L=A\cdot B=AB$，小圆

点"·"表示 A、B 的与运算，也称作逻辑乘，通常被省略。

如果将变量 A、B 和 L 的取值对应关系用一张表格同时列出来(称为真值表)，可得到表 1-2 所示的与运算真值表。

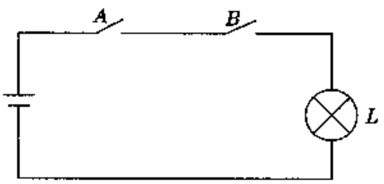

图 1-1　与逻辑关系

表 1-2　与运算真值表

A	B	$L=AB$
0	0	0
0	1	0
1	0	0
1	1	1

从表 1-2 中可以清楚地看出：只有当 A 和 B 同时为 1 时，L 才为 1，与运算规律如下：

$$0\cdot0=0;0\cdot1=0;1\cdot0=0;1\cdot1=1$$

(2)或运算

当决定一件事情的各个条件中，只要一个或一个以上具备时，这件事情就会发生，这样的逻辑关系称为或逻辑。或逻辑关系可用图 1-2 所示的电路模拟。

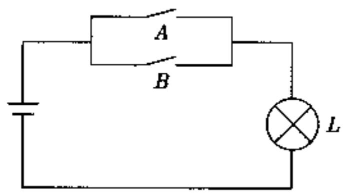

图 1-2　或逻辑关系

同样，如果开关闭合用 1 表示，断开用 0 表示，灯亮用 1 表示，灯灭用 0 表示，可用逻辑表达式表示为：$L=A+B$。或逻辑关系真值表如表 1-3 所示。

表 1-3　或运算真值表

A	B	$L=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

或运算规律如下：

$$0+0-0;0+1=1;1+0=1;1+1=1$$

(3)非运算

非逻辑关系可用如图 1-3 所示的电路模拟，如果用 1 表示开关闭合，0 表示开关断开；1 表示灯亮，0 表示灯灭，当开关 A 接通时，即为 1 时，灯灭为 0；当开关断开，即为 0 时，灯亮为 1。逻辑非运算是对逻辑变量 A 进行否定，其表达式可写作 $L=\overline{A}$。

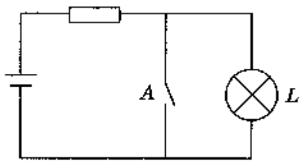

图 1-3 非逻辑关系

表 1-4 非逻辑真值表

A	L
0	1
1	0

非逻辑运算规律： $\overline{0}=1;\overline{1}=0$

1.3.2 基本公式和定理

1. 基本公式

自等律： $A+0=A$

$A\cdot 1=A$

吸收律： $A+1=1$

$A\cdot 0=0$

互补律： $A+\overline{A}=1$

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

交换律： $A+B=B+A$

$A\cdot B=B\cdot A$

重叠律： $A+A=A$

$A\cdot A=A$

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

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

还原律： $\overline{\overline{A}}=A$

结合律： $(A+B)+C=A+(B+C)=(A+C)+B$

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

分配律： $A\cdot(B+C)=AB+AC$

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

上述基本公式能用列真值表的方法加以证明，读者可以自己去做。

2. 三个重要规则

逻辑代数中有三个重要规则，分别是：代入规则、反演规则和对偶规则。应用这三个规则，可以由已知的基本公式推导出更多实用的公式。

(1)代入规则

在任何逻辑等式中，如果将所有出现某一逻辑变量用一个逻辑函数代入，则等式仍然成立。

例如：已知等式 $\overline{A\cdot B}=\overline{A}+\overline{B}$，若用 $Z=BC$ 代替等式中的 B，则

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

(2)反演规则

对于任意逻辑表达式 Z，如果将其中所有的"·"换成"+"、"+"换成"·"；再将原变量换为反变量、反变量换为原变量；将 1 换成 0，0 换成 1。那么得到的逻辑函数称为 Z 的反函数，用 $\overline{Z}$ 表示。利用反演规则可以很方便地求一个函数的反函数。

【例 1-10】 求函数 $F_1=\overline{A}B+A\overline{B}C+CD$ 的反函数。

解：根据反演规则可得

$$\overline{F_1}=(A+\overline{B})\cdot(\overline{A}+B+\overline{C})\cdot(\overline{C}+\overline{D})$$

【例 1-11】 求函数 $F_2=A+\overline{B+\overline{C}+\overline{D+\overline{E}}}$的反函数。

解：利用反演规则得

$$\overline{F_2}=\overline{A}\cdot\overline{\overline{B}\cdot C\cdot\overline{\overline{D}\cdot E}}$$

由上述两个例子可以看出，使用反演规则需要注意两点：

①必须遵守"先括号，然后乘，最后或"的运算顺序；

②不属于单个变量上的"反"号应保留不变。

(3)对偶规则

对于任意逻辑表达式 Z，如果将其中的"·"换成"+"、"+"换成"·"；1 换成 0、0 换成 1，得到的新表达式记为 Z'，称为 Z 的对偶式。

例如：$Z=(A+\overline{B})(A+C)$的对偶式为：$Z'=A\cdot\overline{B}+A\cdot C$，$L=A+B\overline{C}$的对偶式为：$L'=A\cdot(B+\overline{C})$。

变换时同样要遵照逻辑运算的优先顺序的规定。另外，当证明了两个逻辑表达式相等后，其相应的对偶式也必然相等。

3. 常用公式

利用前面介绍的基本公式和三个重要规则，可以得到更多的公式。下面只列出一些常用公式。

公式 1 $A\cdot B+A\cdot\overline{B}=A$

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

公式 2 $A+AB=A$

证明： $A+AB=A(1+B)=A$

公式 3 $A+\overline{A}B=A+B$

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

公式 4 $AB+\overline{A}C+BC=AB+\overline{A}C$

证明： $AB+\overline{A}C+BC=$
$AB+\overline{A}C+(\overline{A}+A)BC=$
$(AB+ABC)+(\overline{A}C+\overline{A}BC)=$
$AB+\overline{A}C$

同样可以推得： $AB+\overline{A}C+BCD=AB+\overline{A}C$