“电气工程及其自动化”专业继续教育（函授）专科系列教材

数字电子技术

Shuzidianzijishu

王义军 主编

为继续教育（函授）量身定做

中国电力出版社
www.cepp.com.cn

“电气工程及其自动化”专业继续教育（函授）专科系列教材

数字电子技术

主　编：王义军
副主编：施正一　袁　燕

中国电力出版社
www.cepp.com.cn

内 容 提 要

全书共分为九章，主要内容包括：逻辑代数基础、门电路、组合逻辑电路、时序逻辑电路基础、集成时序逻辑电路、脉冲波形产生与整形电路、数模和模数转换电路、半导体存储器以及可编程逻辑器件与EDA技术。在各章的内容编排上，力求便于学生自学，各章节内容以基础知识和掌握逻辑关系为主，集成电路以外特性为主，减少对内部结构的讲解；增加了例题和课后习题。

本书简明扼要，深入浅出，主要面向高校工科电气工程自动化专业的专科成人函授学员，也可供从事电子技术工作的工程技术人员学习、参考和大专院校师生选用。

图书在版编目（CIP）数据

数字电子技术/王义军主编．—北京：中国电力出版社，2005

（电气工程及其自动化专业继续教育（函授）专科系列教材）

ISBN 7-5083-3196-6

Ⅰ．数…　Ⅱ．王…　Ⅲ．数字电路-电子技术-函授大学-教材　Ⅳ．TN79

中国版本图书馆CIP数据核字（2005）第087829号

中国电力出版社出版、发行

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

汇鑫印务有限公司印刷

新华书店北京发行所发行　各地新华书店经售

*

2006年1月第一版　　2006年1月北京第一次印刷

787毫米×1092毫米　16开本　16.5印张　374千字

印数0001—3000册　　定价**25.00**元

“电气工程及其自动化”专业继续教育（函授）系列教材

参 编 学 校

华中科技大学	电气与电子工程学院
武汉大学	电气工程学院
华北电力大学	电气工程学院
东北电力学院（大学）	电气工程学院
三峡大学	电气工程学院
上海电力学院	电力与自动化工程学院
长沙理工大学	电气与信息工程学院
武汉电力职业技术学院	

编者按语

根据《中国教育改革与发展纲要》中“要大力发展成人高等教育”的精神，由华中科技大学电气与电子工程学院和武汉大学电气工程学院牵头，组织华北电力大学电气工程学院、东北电力学院（大学）电气工程学院、三峡大学电气工程学院、上海电力学院电力与自动化工程学院、长沙理工大学电气与信息工程学院、武汉电力职业技术学院等单位，成立了“电气工程及其自动化”专业继续教育（函授）教学工作协作组，于2003年11月在武汉，就国家在新形势下对人才的需求及“电气工程及其自动化”专业继续教育（函授）的现状、特点和人才供需状况，对“电气工程及其自动化”专业继续教育（函授）的教学计划、课程体系和使用教材现状进行了充分地研讨，制定了“电气工程及其自动化”专业继续教育（函授）专科和专升本两个层次的指导性教学计划。在此基础上研究了本专业的教材建设问题，大家一致认为函授教材要遵循自学和面授相结合、理论和实践相结合的原则，体现市场经济和科技发展对继续教育知识更新和理念更新的要求。针对目前“电气工程及其自动化”专业继续教育（函授）专科和专升本两个层次尚缺乏系统性教材的现状，决定组织各学院有经验的教授和专家编写这两个层次的教材。我们希望这两套系列教材能为规范本专业的教学内容和提高本专业的教学质量起到积极的推动作用。

“电气工程及其自动化”专业继续教育（函授）的教材建设，现在只是开头，需不断改进和完善。因此，在使用过程中敬请读者随时提出宝贵的意见和建议，以便今后修订或增补。

“电气工程及其自动化”专业继续教育（函授）教学工作协作组
“电气工程及其自动化”专业继续教育（函授）系列教材编委会

2005年3月

前 言

为了适应现代电子技术迅速发展的需要，能够较好地面向21世纪，面向数字化和专用集成电路的新时代，在保证基本概念、基本原理和基本方法的前提下，本教材压缩了集成电路电气特性的讨论和内部电路工作原理的分析，而突出综合能力的培养、训练以及集成电路逻辑特性和工作特点的介绍。

本书中把逻辑代数基础作为第一章，以便更好地从逻辑分析和综合角度，组织、讲解课程的主要内容；为开拓学员视野，在第九章增加了可编程逻辑器件相关内容的介绍。

本书由东北电力学院（大学）王义军副教授作为主编，上海电力学院施正一副教授和武汉电力高等专科学校的袁燕老师作为副主编，王义军老师编写本书的第一、第四、第六、第七、第八、第九章，施正一老师编写本书的第二、第三章，袁燕老师编写本书的第五章，王义军修改和统编了全书。

在本书的编写过程中还得到了东北电力学院（大学）任先文教授、李辉老师的友情帮助，任先文教授还参与了本书的大纲和各章节内容的讨论，提出很多宝贵意见，在此一并致谢。

由于我们水平有限，加之时间仓促，书中一定存在错误和不妥之处，敬请各方面的读者予以批评指正，以便今后不断改进。

编　者

2005年4月

目录

第 1 章 逻辑代数基础

逻辑代数是分析和设计数字逻辑电路的基本数学工具，它的基本和常用运算也是数字电路要实现的重要操作。本章主要讲述逻辑代数的基本概念、公式和定理，逻辑函数的化简方法，几种常用逻辑函数的表示方法及其相互间的转换，同时对数字在计算机内的表示法也做了必要的介绍。

1.1 概 述

1.1.1 数字信号和模拟信号

电子电路所处理的电信号可以分为两大类：一类是在时间和数值上均连续的信号，称为模拟信号，例如电压、电流、温度等，把工作在模拟信号下的电子电路叫做模拟电路；另一类是在时间和数值上都分散的信号，称为数字信号，把工作在数字信号下的电子电路叫做数字电路。

1.1.2 数制和码制

一、数制

用数字量表示物理量的大小时仅用一位数码往往不够用，常用进位计数的方法组成多位数码使用。我们把多位数码中每一位的构成方法以及从低位到高位的进位规则称为数制。

二进制数是数字电路中应用最广泛的一种数值表示方法。为了更容易的理解有关概念，首先介绍熟悉的十进制数表示方法，进一步介绍二进制以及多位二进制数构成的八进制和十六进制数的计数规律。

1. 十进制

十进制是日常生活和工作中应用最广泛的一种计数进位制。在这种计数进位制中，每一位有 0～9 十个数码，所以计数的基数是十。低位数和相邻高位数的关系是“逢十进一”，故称为十进制。

【例 1－1】

$$193.74 = 1\times10^2 + 9\times10^1 + 3\times10^0 + 7\times10^{-1} + 4\times10^{-2}$$

所以任意一个十进制数 D 均可展开为

$$D = \sum_{i=-m}^{n-1} k_i \times 10^i \tag{1-1}$$

式（1-1）中，k_i 是第 i 位的系数，它可以是 0~9 中的任何一个数字。i 的取值范围从 $-m$ 到 $n-1$，其中 m 是小数部分的位数，n 是整数部分的位数。若用 r 取代式（1-1）中的 10，即可得到任意进制数展开式的普遍形式：

$$D = \sum_{i=-m}^{n-1} k_i \times r^i \tag{1-2}$$

式中 i 的取值与式（1-1）的规定相同；r 称为计数的基数，k_i 为第 i 位的系数，r^i 称为第 i 位的权。

2. 二进制

在数字电路中应用最广泛的是二进制。在二进制数中，每一位仅有 **0** 和 **1** 两个可能的取值，所以计数的基数 $r=2$。低位向高位的进位关系是“逢二进一”。

根据式（1-2），任何一个二进制数均可展开为：

$$D = \sum_{i=-m}^{n-1} k_i \times 2^i \tag{1-3}$$

并可利用此式计算出它所表示的十进制数值。

【例 1-2】

$$(\mathbf{1101.11})_2 = \mathbf{1} \times 2^3 + \mathbf{1} \times 2^2 + \mathbf{0} \times 2^1 + \mathbf{1} \times 2^0 + \mathbf{1} \times 2^{-1} + \mathbf{1} \times 2^{-2} = (13.75)_{10}$$

式中，右下标 2 和 10 分别表示括号里的数是二进制和十进制。也可以用字母 B 表示二进制数，字母 D 表示十进制数。

3. 十六进制

十六进制的每一位有十六个不同的数码，由 0~9 和用大写字母 A、B、C、D、E、F（表示十进制的 10、11、12、13、14、15 六个数字）组成。因此，任意一个十六进制的数均可展开为

$$D = \sum_{i=-m}^{n-1} k_i \times 16^i \tag{1-4}$$

并可利用式（1-4）计算出它所表示的十进制数值。

【例 1-3】

$$(\mathrm{BA.7F})_{16} = 11 \times 16^1 + 10 \times 16^0 + 7 \times 16^{-1} + 15 \times 16^{-2} = (186.4960937)_{10}$$

式中，下标 16 表示括号里的数是十六进制，也可用字母 H 表示。

4. 八进制

八进制的每一位有八个不同的数码，由 0~7 组成。因此，任意一个八进制的数均可展开为

$$D = \sum_{i=-m}^{n-1} k_i \times 8^i \tag{1-5}$$

并由式（1-5）计算出它所表示的十进制数值。

【例 1-4】

$$(43.25)_8 = 4 \times 8^1 + 3 \times 8^0 + 2 \times 8^{-1} + 5 \times 8^{-2} = (35.328125)_{10}$$

式中，下标 8 表示括号里的数是八进制，也用字母 O 表示。

二、数制转换

1. 二～十转换

只要将二进制数按式（1-3）展开，然后把所有各项的数值按十进制数相加即可。

【例 1-5】

$$
\begin{aligned}
(11011.01)_2 &= \mathbf{1}\times 2^4+\mathbf{1}\times 2^3+\mathbf{0}\times 2^2+\mathbf{1}\times 2^1+\mathbf{1}\times 2^0+\mathbf{0}\times 2^{-1}+\mathbf{1}\times 2^{-2} \\
&= (27.25)_{10}
\end{aligned}
$$

2. 十～二转换

（1）整数部分的转换：假定十进制整数为 $(S)_{10}$，等值的二进制数为 $(k_{n-1},\cdots,k_0)_2$，则依式（1-3）知：

$$
\begin{aligned}
(S)_{10} &= k_{n-1}2^{n-1}+k_{n-2}2^{n-2}+\cdots+k_1 2^1+k_0 2^0 \\
&= 2(k_{n-1}2^{n-2}+k_{n-2}2^{n-3}+\cdots+k_1)+k_0 \qquad (1-6)
\end{aligned}
$$

式（1-6）表明，若将 $(S)_{10}$除以 2，则得到的商为 $k_{n-1}2^{n-2}+k_{n-2}2^{n-3}+\cdots+k_1$ 而余数为 k_0。根据等式两端对应项系数相等的原则可以得到十进制数 S 除以 2 的余数就是 k_0，同理由其商再除以 2 可得余数就是 k_1。依此类推，反复将每次得到的商再除以 2，直到除尽为止，就可求得二进制数的每一位的系数。

【例 1-6】

将 $(57)_{10}$化为二进制数可如下进行：

2	57	余数 = **1** = k_0
2	28	余数 = **0** = k_1
2	14	余数 = **0** = k_2
2	7	余数 = **1** = k_3
2	3	余数 = **1** = k_4
2	1	余数 = **1** = k_5
	0	

$$(57)_{10}=(111001)_2$$

（2）小数部分的转换：假设 $(S)_{10}$是一个十进制的纯小数，对应二进制小数为 $(0,k_{-1},k_{-2},\cdots,k_{-m})_2$ 则据式（1-3）可知

$$(S)_{10}=k_{-1}2^{-1}+k_{-2}2^{-2}+\cdots+k_{-m}2^{-m}$$

将上式两边同乘以 2 得到

$$2\times(S)_{10}=(k_{-1}+k_{-2}2^{-1}+k_3 2^{-2}+\cdots+k_{-m}2^{-m+1}) \qquad (1-7)$$

根据等式两端对应项系数相等的原则，式（1-7）说明，将小数 $(S)_{10}$乘以 2 所得乘积的整数部分的系数就是 k_{-1}，同理将得到的结果再次乘以 2 就会得到二进制小数的其他位的系数。

【例 1-7】

将 $(0.8125)_{10}$化为二进制小数时：

$$
\begin{array}{r}
0.8125 \\
\times \quad\quad 2 \\
\hline
\boxed{1}.6250 \\
\times \quad\quad 2 \\
\hline
\boxed{1}.2500 \\
\times \quad\quad 2 \\
\hline
\boxed{0}.5000 \\
\times \quad\quad 2 \\
\hline
\boxed{1}.0000
\end{array}
\quad
\begin{array}{l}
\\
\\
\\
整数部分 = \mathbf{1} = k_{-1} \\
\\
\\
整数部分 = \mathbf{1} = k_{-2} \\
\\
\\
整数部分 = \mathbf{0} = k_{-3} \\
\\
\\
整数部分 = \mathbf{1} = k_{-4}
\end{array}
$$

$$故 \ (0.8125)_{10} = (\mathbf{0.1101})_2$$

应该指出，因为十进制数乘以 2 不一定都能乘尽，所以并不是所有的十进制小数都能转化成有限位数的二进制。通常情况下这种转换需要给出转换精度，转换满足精度就可以。

3. 二进制～十六进制转换

4 位二进制一共有 16 种组合，将这十六种组合按照每位的权展开就得到二进制到十六进制的对应表如表 1－1 所示。

表 1－1　二进制～十六进制对应表

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 4 位二进制数看作一个整体，以小数点为出发点向两侧每 4 位二进制数分为一组，整数部分不足 4 位的高位用**0**补充，小数部分不足 4 位的低位用**0**补充，每四位二进制数以等值的十六进制数表示，即可得到相应的十六进制数。

【例 1－8】　将 $(\mathbf{1011110.1011001})_2$ 化为十六进制数时可得

$$(\underline{\mathbf{0}}\mathbf{101}\quad \mathbf{1110.1011}\quad \mathbf{001}\,\underline{\mathbf{0}})_2 = (5E.B2)_{16}$$

式中用下划线表示的 0 是由于数位不足而补充的。

4. 十六进制～二进制转换

把每 1 位十六进制数对应 4 位二进制数，就可实现十六进制向二进制的转换。

如将 $(8FA.C6)_{16}$化二进制数时，得

$$(8 \quad\quad F \quad\quad A. \quad\quad C \quad\quad 6)_{16} =$$

$$(\mathbf{1000} \quad \mathbf{1111} \quad \mathbf{1010.} \quad \mathbf{1100} \quad \mathbf{0110})_2$$

5. 十六进制数～十进制数的转换

十六进制化为十进制：根据式（1－4）将各位按权展开后相加求得。

6. 十进制～十六进制的转换

方法与十进制到二进制的转换方法类似：整数部分，除 16 取余；小数部分，乘 16 取整。另外，也可先将十进制数转换为二进制数，再把二进制数转换成十六进制数。

三、码制

用文字、代码、数字等来表示不同的事物的过程叫做编码，在编制代码时要遵循一定的规则，这些规则就叫做码制。

例如在用4位二进制数码表示1位十进制数的0~9这十个状态时，通常称为二~十进制代码，简称BCD（Binary Code decimal Code）代码。

8421码是BCD代码中最常用的一种。它的4位二值代码的**1**从高位到低位所代表数值分别是8、4、2、1，把每一位的1代表的十进制数加起来，就得到所代表十进制数码。表1-2给出8421 BCD码和对应的十进制数的关系。

表1-2　8421BCD码编码表

十进制数字	8421BCD码				十进制数字	8421 BCD码			
	B3	B2	B1	B0		B3	B2	B1	B0
0	**0**	**0**	**0**	**0**	5	**0**	**1**	**0**	**1**
1	**0**	**0**	**0**	**1**	6	**0**	**1**	**1**	**0**
2	**0**	**0**	**1**	**0**	7	**0**	**1**	**1**	**1**
3	**0**	**0**	**1**	**1**	8	**1**	**0**	**0**	**0**
4	**0**	**1**	**0**	**0**	9	**1**	**0**	**0**	**1**

1.1.3　算术运算

1位二进制数码的**0**和**1**可以表示两种不同的逻辑状态称为二值逻辑。二进制数码的算术运算在加法时遵循“逢二进一”，在减法时遵循“借一当二”的规律。

例如，两个二进制数**1001**和**0101**的算术运算：

加法运算　　　　　　　　减法运算

$$\begin{array}{r} 1001 \\ +0101 \\ \hline 1110 \end{array} \qquad\qquad \begin{array}{r} 1001 \\ -0101 \\ \hline 0100 \end{array}$$

在数字电路和数字电子计算机中，二进制数的正、负号分别用**0**和**1**表示。以最高位作为符号位，其他位不变的数码称为原码。

$(\mathbf{0}\quad \mathbf{1011001})_2 = (+89)_{10}$

正数符号位　原有数值

$(\mathbf{1}\quad \mathbf{1011001})_2 = (-89)_{10}$

负数符号位　原有数值

补码：最高位为符号位，正数的补码和它的原码相同；负数的补码可通过将原码的数值位逐位求反，然后加上1得到。

【例1-9】　计算 $(\mathbf{1001})_2-(\mathbf{0101})_2$

解答　根据二进制的运算规则可知

$$\begin{array}{r} 1001 \\ -0101 \\ \hline 0100 \end{array}$$

采用补码运算时，首先求出 $(+\mathbf{1001})_2$ 和 $(-\mathbf{0101})_2$ 的补码，它们是

$$(+1001)_{补} = \underset{\text{符号位}}{\underset{\downarrow}{\mathbf{0}}}\quad 1001$$

$$(-0101)_{补} = \underset{\substack{\downarrow \\ \text{符号位}}}{1}\quad 1011$$

然后将两个补码相加并舍去进位

$$\begin{array}{r} 01001 \\ -\ \ 11011 \\ \hline \underset{\substack{\downarrow \\ \text{进位舍去}}}{1}\ \ 00100 \end{array}$$

舍去进位后与减运算的结果相同。同时应该指出，在计算机内所有的四则运算都是利用加法来实现的。

1.2 逻辑代数中的基本运算

1849 年，英国数学家乔治·布尔（Geogle Boole）首先提出了描述客观事物逻辑关系的数学方法，称为逻辑代数，又称为开关代数或布尔代数。逻辑代数中用大写字母表示的变量称为逻辑变量。逻辑变量在二值逻辑中只有 **0** 和 **1** 两种取值，这里 **0** 和 **1** 表示逻辑状态。

在研究事件的因果变化关系时，决定事件变化的因素称为逻辑自变量，而与之对应事件的结果称为逻辑结果，以某种形式表示的逻辑自变量与逻辑结果之间的函数关系称为逻辑函数。

1.2.1 逻辑代数中的三种基本运算

在逻辑代数中，基本运算有三种，即逻辑**与**、逻辑**或**、逻辑**非**。与之相对应，在逻辑代数中，基本的逻辑运算也有三种：**与**运算、**或**运算、**非**运算。为了理解**与**、**或**、**非**三种基本逻辑运算的含义，下面以图形为例进行说明。

图 1－1（a）表明只有决定事物结果（灯亮）的全部条件（开关闭合）同时具备时，结果才发生。这种因果关系称为**与**逻辑，或者称为逻辑相乘。

图 1－1（b）表明在决定事物结果的诸条件中只要有一个满足，结果就发生。这种因果关系称为逻辑**或**，也称为逻辑相加。

图 1－1（c）表明只要条件具备了结果便不会发生；而条件不具备时，结果一定发生，这种因果关系称为逻辑**非**，也称为逻辑求反。

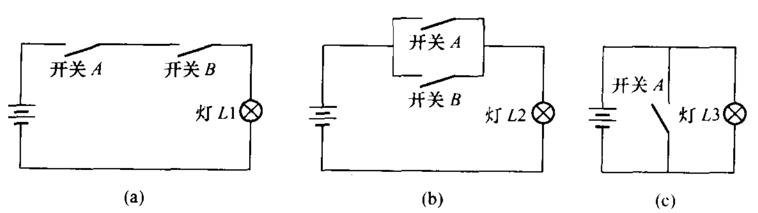


图 1－1　基本逻辑关系电路举例

若以 A、B 表示开关状态，1 为闭合，0 为断开；以 L 表示指示灯，1 为亮，0 为不亮，图 1-2 列出了由0、1表示的**与**、**或**、**非**逻辑关系的图表，这些表称为逻辑函数的真值表，其描述了电路输入变量的所有取值与输出变量取值的对应关系。

A	B	$L1$
0	0	0
0	1	0
1	0	0
1	1	1

(a)

A	B	$L2$
0	0	0
0	1	1
1	0	1
1	1	1

(b)

A	$L3$
0	1
1	0

(c)

图 1-2　图 1-1 所示电路的真值表

(a) 与逻辑运算的真值表；(b) 或逻辑运算的真值表；(c) 非逻辑运算的真值表

在逻辑代数中：

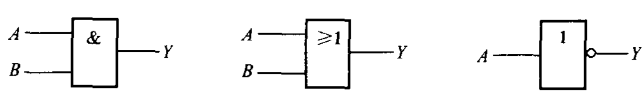
以“·”表示**与**运算，可写成　　$Y = A \cdot B$　　(1-8)

以“+”表示**或**运算，可写成　　$Y = A + B$　　(1-9)

在变量上方加“-”表示**非**运算，可写成　　$Y = \overline{A}$　　(1-10)

分别把实现**与**、**或**、**非**逻辑运算的单元电路称为**与**门、**或**门、**非**门（也叫反向器）。

与、**或**、**非**逻辑的国际符号如图 1-3 所示。



(a)　(b)　(c)

图 1-3　与门、或门、非门的国际标准符号

(a) 与门逻辑图；(b) 或门逻辑图；(c) 非门逻辑图

1.2.2　常用的几种复合逻辑关系

实际中还有复杂的逻辑运算如：**与非**、**或非**、**异或**等，其逻辑关系的真值表如图 1-4 所示，逻辑图如图 1-5 所示。

与非运算表示成　　$Y = \overline{A \cdot B}$　　(1-11)

或非运算表示成　　$Y = \overline{A + B}$　　(1-12)

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

(a)

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

(b)

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

(c)

图 1-4　与非、或非、异或逻辑的真值表

(a) 与非逻辑运算的真值表；(b) 或非逻辑运算的真值表；

(c) 异或逻辑运算的真值表

异或运算表示成 $$Y = A \oplus B \tag{1-13}$$

与非、或非、异或逻辑的国际符号如图 1-5 所示。

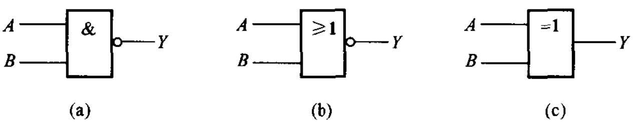


图 1-5　与非、或非、异或逻辑符号

(a) 与非门逻辑图；(b) 或非门逻辑图；(c) 异或门逻辑图

1.3　逻辑函数的定理和规则

1.3.1　逻辑代数基本公式和常用公式

表 1-3 给出了逻辑代数的基本公式，这些公式也叫布尔恒等式。

表 1-3　　**逻辑代数的基本公式**

常量之间的关系	$0 \cdot 0 = 0$	$1 + 1 = 1$
	$0 \cdot 1 = 0$	$1 + 0 = 1$
	$1 \cdot 0 = 0$	$0 + 1 = 1$
	$1 \cdot 1 = 1$	$0 + 0 = 0$
	$\overline{0} = 1$	$\overline{1} = 0$
变量与常量之间的关系	$A \cdot 1 = A$	$A + 0 = A$
	$A \cdot 0 = 0$	$A + 1 = 1$
	$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$

交换律	$A \cdot B = B \cdot A$	$A + B = B + A$
结合律	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A + B) + C = A + (B + C)$
分配律	$A \cdot (B + C) = A \cdot B + A \cdot C$	$A + (B \cdot C) = (A + B) \cdot (A + C)$
同一律	$A \cdot A = A$	$A + A = A$
德·摩根定理	$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$
还原律	$\overline{\overline{A}} = A$	

以上公式的证明可以用真值表的方法来进行，如果等式两端的真值表完全相同，那么这个等式就成立。

【例 1-10】　用真值表证明表 1-3 中式 $A + B \cdot C = (A + B) \cdot (A + C)$ 的正确性。

解答　这个分配律是逻辑代数特有的，列出等式两端式子的真值表。

将 A、B、C 所有可能的取值组合逐一代入上式的两边，算出相应的结果，即得到表 1-4 的真值表。可见，等式两边对应的真值表相同，故等式成立。

表 1-4　　**式 $A + B \cdot C$ 和式 $(A + B) \cdot (A + C)$ 的真值表**

A	B	C	$A + B \cdot C$	$(A + B) \cdot (A + C)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1