數字电路的邏輯设计

樸初题

冯昭逢编著

湖南科学技术出版社

数字电路的逻辑设计

冯昭逢编著

湖南科学技术出版社
一九八〇年·长沙

数字电路的逻辑设计

冯昭逢　编著

责任编辑：陈清山

装帧设计：姜铁山

*

湖南科学技术出版社出版

（长沙市展览馆路14号）

湖南省新华书店发行　湖南省新华印刷二厂印刷

*

1980年9月第1版第1次印刷

字数：342,000　印张：15.75　印数：1——4,200

统一书号：15204·41　定价：1.50元

内 容 提 要

本书简明地论述了数字电路的逻辑设计问题。它以生动有趣的实例介绍了布尔代数的基本内容和应用、组合逻辑电路、同步和异步时序电路的一般设计方法、数字传输差错的检测等内容，并对常用的各类数字电路——二进制运算电路、各种计数器、译码器、移位寄存器等的设计方法分别举例加以讨论。它是从事计算机逻辑设计、数字控制、数字仪表、数字自动化各领域教学、科研、生产人员的一本有益的参考书。

前　言

数字电路的逻辑设计是电子数字技术的重要基础之一。在我国现实的条件下，要推广电子数字计算机的应用，应该了解数字电路的逻辑设计知识。本书用通俗易懂的语言介绍了数字电路逻辑设计中的一些基本内容。

第一章是本书内容的基础，讨论了布尔代数的基本内容及其应用，并举了一些浅显且有趣的例子以帮助读者理解。第二章介绍了数字电路设计中最有效的工具——卡诺图，它是本书后续各章的主要设计手段。第三章介绍的二进制运算电路，是为了让不熟悉电子数字计算机的读者了解实现数字运算的基本方法，这些方法可以用于简单的数字控制等方面。第四章为时序电路的基本单元——触发器的逻辑功能。本章不再介绍分立元件触发器的原理和性能，而是着重讨论触发器的各种逻辑功能及其转换。第五章和第六章为同步、异步时序电路的一般设计方法，并通过实例设计的介绍，以达到使读者易于掌握的目的。第七章和第八章为应用最广泛的各类计数器、译码器和移位寄存器等设计方法的讨论，它是第二章、第五章和第六章介绍的一般方法的典型应用。第九章中介绍的数字传输中差错的检测是数字电路设计中必须考虑的问题。

本书主要是为希望应用计算机而又不了解计算机的各行各业的教学、科研、技术人员介绍逻辑设计的方法。本书可以作为计算机、通讯、雷达、导航、自动控制、数字测量等各专业师生的教学参考书，但更适合从事计算机应用的工程技术人员作为参考书使用。

涂克文、郭德纨和孙毅君同志审阅了全书，王博亮同志作了细致的校对。邵晶同志对本书文字进行了认真的修饰。作者在此一并表示感谢。

中国人民解放军国防科技大学

冯昭逵

1979年6月

目　　录

第一章　逻辑问题的引入和布尔代数

一、三种基本逻辑

1.0和1的逻辑含义

电子数字计算机能够运算各种复杂的数学问题，能够控制各种生产过程，能够处理许多烦琐的情报资料。它之所以能代替人的部分思惟能力，关键在于它具有逻辑判断能力。

任何矛盾现象，不管它多么复杂，总可以分解为一系列对立着的简单矛盾。人们判断事物，总是从简单的判据出发，逐步推理而深入的。简单的判据总是二值的，即不是“是”就是“非”。例如，我们判断“学校开学没有？”结果只能是“开学了”或者“还没开学”。判断“电灯亮没亮？”结果只能是“亮了”或者“没亮”。

这种二值的简单判据我们称其为逻辑命题。

例如：用A来表示“学校开学没有”这个命题。A＝1表示“开学了”，A＝0表示“还没开学”。当然，也可以相反。

用B来表示“电灯亮没亮”这个命题。B＝1表示“灯亮了”，B＝0表示“灯没亮”。

由此可见，0和1已经不是我们过去所熟悉的数字了，而是有了新的逻辑含义，它们分别表示一个逻辑命题的两种相反的结果。若1表示“肯定”的意思，则0就表示“否定”的意思。二值判据，在计算机中或其他控制设备中是很容易实现的。可以用电平的高低，信号的有无，晶体管的通断等等来表示逻辑命题的两种相反的结果“0”和“1”。

人们在进行逻辑推理时，存在着三种最基本的逻辑判据。这三种最基本的逻辑就是“与”逻辑、“或”逻辑和“否”逻辑。

2.“与”逻辑

例如：甲、乙两人好下棋，走到一起必下棋。“下棋”就是一个事件。人们判断“下棋”这件事发生没有，就要看甲、乙两人到没到？

甲与乙都来了，“下棋”这事件就成立。

若甲到了，而乙没来；或乙来了，甲却没到，“下棋”此事就不可能实现。只有甲与乙都到场，下棋才能实现。甲、乙“到没到”和“下棋”这事件之间的关系就是“与”逻辑关系。

若甲、乙来了用“1”表示，没来用“0”表示。“下棋”这事件若用丙表示，丙＝1

表示“下棋”成立，丙＝0表示没有下棋。

则“与”逻辑表达式

$$丙=甲\times 乙$$

称为“逻辑乘”。对于上例中只有两个逻辑变量甲、乙的逻辑乘来说，显然只有四种情况。

即：甲乙都没来；

甲来乙没来；

乙来甲没来；

甲乙都来了。

甲	乙	丙＝甲×乙
0	0	0
0	1	0
1	0	0
1	1	1

图1－1　二变量逻辑乘真值表

可以列表如图1－1所示。此表称为逻辑乘的真值表。

用串联开关能更好地说明“与”逻辑的概念　若三个乒乓开关A、B、C和一个灯泡串联，如图1－2所示，显然，只有A与B与C三个开关全部闭合时，灯泡才亮。A、B、C中只要有一个断开，灯泡就不亮。

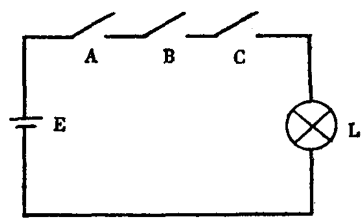

图1－2　“与”逻辑电路

开关A、B、C的“断开、闭合”和灯泡“亮、不亮”之间的逻辑关系就是“与”逻辑关系。“与”在此例中就是开关A、B、C“闭合”这三个条件都具备的意思。

若规定A、B、C为“1”表示开关闭合，为“0”表示开关断开；用L表示“灯泡亮否”，L＝1表示“灯亮”，L＝0表示“灯不亮”。则其逻辑表达式为

$$L=A\times B\times C$$

这里逻辑变量有三个，A、B、C取值情况的组合有八种。这八种组合情况列在图1－3所示的三个逻辑变量逻辑乘的真值表中。对于有n个逻辑变量的逻辑乘，其将有2^n种组合。即真值表将有2^n行。

A	B	C	L＝A×B×C
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

图1－3　三变量逻辑乘真值表

通过以上二例的介绍，我们可以用文字来叙述“与”逻辑的一般意义。

“一事件x成立与否，与命题A、B、C相联系。如果，命题A、B、C都成立，则x亦成立。如果，A、B、C之中，只要有一个或一个以上不成立，则x亦不成立。事件x和命题A、B、C之间为“与”逻辑关系。”

在逻辑运算中，“与”逻辑称之为“逻辑乘”。在书写时，逻辑乘中的乘号可以略去不写。即

$$L=A\times B\times C=A\cdot B\cdot C$$

由图 1－1 和图 1－3 的逻辑乘真值表，我们可以看出，逻辑乘运算和四则运算中的乘法运算在形式上是类似的。而在本质上，逻辑乘运算和乘法运算是完全不同的。

3.“或”逻辑

例如：实验室门的钥匙只有甲、乙两人有。人们判断“实验室的门能否打开？”这样一个事件是否成立时，就要看甲、乙来了没有？

只要甲或乙来了，实验室的门就一定能打开。

只有甲和乙都没来，实验室的门才打不开。甲、乙“来没来”与“实验室的门开否”之间的关系就是“或”逻辑关系。

若甲、乙来了用“1”表示，没来用“0”表示。“门开否”这个事件用丙表示，且丙＝1

甲	乙	丙＝甲＋乙
0	0	0
0	1	1
1	0	1
1	1	1

图 1－4　二变量逻辑加真值表

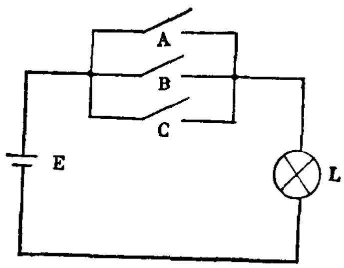

图 1－5　“或”逻辑电路

表示门打开，丙＝0 表示门没打开。

则“或”逻辑表达式为

$$丙＝甲＋乙$$

称为“逻辑加”。对于两个变量的逻辑加来说，亦只有四种情况。

即：甲、乙都没来；

甲来乙没来；

乙来甲没来；

甲乙都来了。

同样可以列出图 1－4 所示的逻辑加的真值表。

用并联开关能更好地说明“或”逻辑的概念。三个乒乓开关A、B、C并联控制一个灯泡，如图 1－5 所示。显然，只要其中有一个开关闭合，灯泡就亮。即A或B或C中只要有一个或一个以上闭合，“灯泡亮”这个事件就发生。

若规定A、B、C为“1”表示开关闭合，为“0”表示开关断开；用L表示“灯泡亮否”，L＝1 表示灯亮，L＝0 表示灯不亮。则“或”逻辑表达式为

$$L = A + B + C$$

这是一个三变量的逻辑加，A、B、C取值的组合有八种情况。图 1－6 为三变量逻辑加的真值表。

通过“或”逻辑二例的介绍，我们亦可用文字来叙述“或”逻辑的一般意义。

“一事件 x 成立与否，与命题A、B、C相联系。如果命题A或B或C中至少有一个成立，则 x 成立。只有命题A、B、C都不成立，x 才不成立。则事件 x 和命题A、B、C之间为“或”逻辑关系。”

A	B	C	L = A + B + C
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

图 1-6　三变量逻辑加真值表

在逻辑运算中，“或”逻辑称之为逻辑加。由图 1-4 和图 1-6 逻辑加真值表可知，逻辑加运算和四则运算中的加法运算在形式上还有不同之处。逻辑加中 $1+1=1$，在四则运算中是不成立的。当然，在本质上这两种运算意义是完全不同的。

4.“否”逻辑

例如：某地区电力紧张。居民供电时，工厂就停电。工厂供电时，居民就停电。人们判断“工厂供电与否？”只要看看家中有电没有？家中有电，工厂就一定停电，家中电灯不亮，说明工厂正在供电。

若以 $A=1$ 表示居民供电，$A=0$ 表示居民停电。$B=1$ 表示工厂供电，$B=0$ 表示工厂停电。则“否”逻辑表达式为

$$B=\overline{A}$$

称之为“逻辑非”。

逻辑非的真值表如图 1-7 所示。其含意是显然的。

同样可以用文字来叙述“否”逻辑的一般意义。

“事件 x 成立与否，与命题A相联系。若A成立，则 x 不成立。A不成立，则 x 成立。事件 x 与命题A之间即为“否”逻辑关系”。

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

图 1-7　逻辑非真值表

在逻辑运算中，“否”逻辑称之为“逻辑非”。“逻辑非”运算是逻辑运算中特有的运算形式，在四则运算中是找不到与之相对应的运算的。

于是，我们就得到$x = A \times B \times C$；$x = A + B + C$；$x = \overline{A}$这样三种“形式语言”来表示的三种基本逻辑运算。运用下面介绍的布尔代数，就可以实现用逻辑运算来代替人的部分的逻辑思维。这就是本章将讨论的主题。

二、基本逻辑门

在电子计算机和其他数字装置中，这三种基本逻辑运算是依靠所谓的“逻辑门”电路来实现的。如前所述，在逻辑运算中，是用“0”和“1”来表示逻辑命题的两种相反的结果。而在逻辑门中，则是用电位的高低来表示“0”和“1”的。我们先假设：用高电位表示“1”，用低电位表示“0”。

1.“与”门电路

与门就是一个能够实现逻辑乘运算的具有多端输入而单端输出的逻辑电路。图1—8就是一个三输入的与门。

输入讯号A、B、C都是只有两种状态，它们不是处于高电位$+V_1$，就是处于低电位0V（地）。换句话说，A、B、C不是“1”就是“0”，都是二值变量。显然，输出x是输入变量A、B、C的二值函数。很容易看出，只有三个输入A、B、C都是高电位$+V_1$时，输出x才可能是高电位$+V_1$（忽略二极管正向通导的压降）。因为任一输入为0V时，输出x必为0V，使高电位输入端的二极管反偏截止。

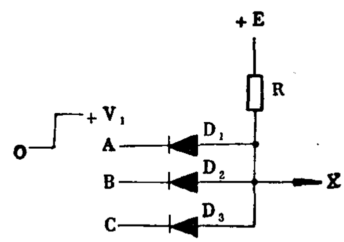

图1-8　与门电路

即，A、B、C都是“1”时，输出x才可能是“1”。A、B、C中只要有一个为“0”，x即为“0”。与图1-3的三变量逻辑乘真值表对照，可见此逻辑门电路的输出x和输入A、B、C之间是符合“与”逻辑关系的。故此与门电路可以完成逻辑乘的逻辑运算。

$$x = A \cdot B \cdot C$$

通常用图1-9所示的逻辑符号来表示与门电路。

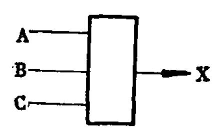

图1-9　与门逻辑符号

在与门电路中，二极管D_1、D_2、D_3是作为开关运用的。实际上，在二极管通导时，二极管两端有正向压降。若采用硅二极管，这个压降有0.6V左右。在二极管反向偏置时，有一个反向漏流。为分析简单起见，我们均不考虑顺向压降和反向漏流，即把二极管看作是一个理想开关。正向时，相当于短路，反向时，相当于开路。图1-10中给出了输入A、B、C为各种不同波形情况下输出x的波形。从波形图中，可以明显看出，只有当A、B、C都是高电位时，输出x才是高电位。

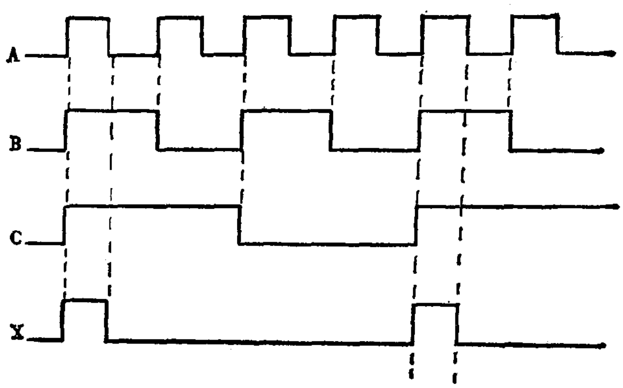

图 1-10　与门输入输出波形

2."或"门电路

或门是一个能够实现逻辑加运算的具有多端输入而单端输出的逻辑电路。图1-11为一个三输入的或门电路。输入讯号A、B、C也都是不是高电位$+V_1$，就是0V（地）的二值变量。由或门电路很容易看出，只要输入讯号A、B、C中，至少有一个输入为高电位$+V_1$时，输出x必为高电位。只有当输入讯号全都为低电位时，输出x才为低电位。例如：A输入为高电位$+V_1$，而B、C输入为0V，显然，输出x为$+V_1$，D_2、D_3将反偏截止。换言之，A、B、C中只要有一个为"1"，输出就为"1"，只有A、B、C全部为"0"时，输出x方为"0"。根据图1-6逻辑加真值表，此或门电路输出x和输入A、B、C之间是符合"或"逻辑关系。即或门可以完成逻辑加的逻辑关系。

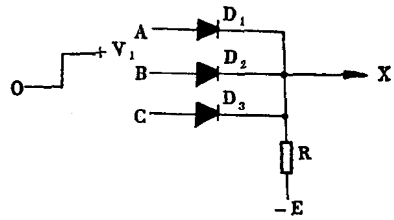

图 1-11　或门电路

$$x = A + B + C$$

通常用图1-12所示的逻辑符号来表示或门电路。

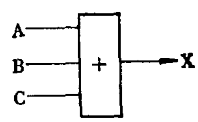

图 1-12　或门逻辑符号

图1-13给出了输入A、B、C为各种不同波形时输出x的波形。从波形图中，可以看出只有输入讯号都

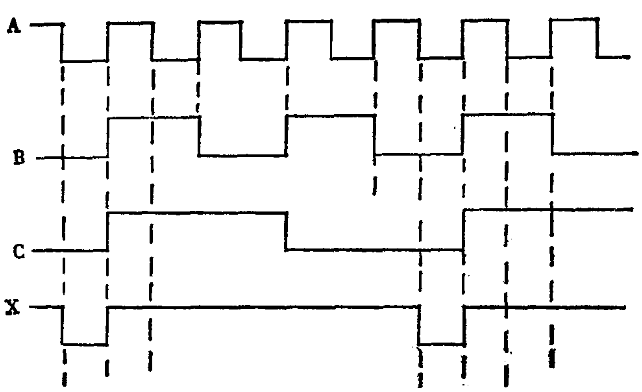

图 1-13　或门输入输出波形

处于低电位时，输出 x 方为低电位。

3．“非”门电路

非门是一个能够完成逻辑非运算的具有单端输入和单端输出的逻辑电路。图 1 -14为非门电路。当输入A为高电平，即“1”时，晶体管T饱和通导，输出 x 为低电位，即为“0”。输入A为低电平，即“0”时，晶体管T截止，输出 x 为高电位，即为“1”。可见，这个电路实现了逻辑非的逻辑运算。

$$x = \overline{A}$$

图 1 -14中给出了输入和输出的波形。这种非门电路又称为倒相器，其名称来由可由输入和输出波形比较而知。

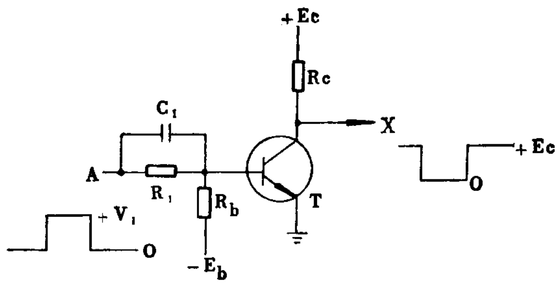

图 1 -14　非门电路

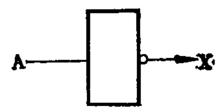

图 1 -15　非门逻辑符号

通常用图 1 -15所示逻辑符号表示非门电路。

本书将不去讨论与门、或门、非门的负载能力、时延、设计计算等问题，读者可以从其他书籍中了解这些基本知识。

4.正逻辑与负逻辑

门电路中，输入讯号和输出都是二值变量，它们都是用电位的高低来表示“0”和“1”的。前面我们曾假设用高电位来表示“1”，而用低电位来表示“0”。在这种假设条件下，逻辑关系称为“正逻辑”。自然也可以假设用高电位表示“0”，而用低电位来表示“1”。这时的逻辑关系称为“负”逻辑。

前面讨论的或门、与门、非门都是在正逻辑条件下讨论的。因此，都必须冠以“正与门”、“正或门”以与负逻辑区别。对于非门来说，在正、负逻辑下，非门是一样的。

这里有一个很有趣的现象：只要我们认真地分析一下“正与门”和“正或门”的输入输出波形，就可以看出，正与门就是负或门，正或门就是负与门。从逻辑乘和逻辑加的真值表更可以直接看出这种互易关系。设图 1 - 3 和图 1 - 6 都是正逻辑下的“与”、“或”逻辑关系。显然，只要把所有的“1”换成“0”，把所有的“0”换成“1”，正逻辑的逻辑乘真值表就变成了负逻辑的逻辑加真值表，正逻辑的逻辑加真值表就变成了负逻辑的逻辑乘真值表。这种互易关系有时是很有用的。以后，本书中除特殊说明者外，均是指正逻辑电路。

三、布尔代数基础

布尔代数这一名称是由乔治·布尔而来。乔治·布尔在一八四七年发表的一篇关于逻辑的数理分析论文中首先提出了布尔代数。但是，直到90年以后，在一九三八年，才由C.E.香农发现布尔代数特别适用于电话和计算机的开关线路。因此，布尔代数的创始人并没有想到布尔代数今天对推动数字计算机的发展是起着多大的作用。布尔代数还有许多其他的名称。例如，符号逻辑；命题演算；逻辑代数；开关代数；…。它的主要功绩是在复杂的开关网络的设计中，用逻辑运算的方法替代了人的部分逻辑思维。

1.布尔代数的基本运算法则

如前所述，基本逻辑运算有三种：逻辑加、逻辑乘、逻辑非。对逻辑加来说：

$$x = A + B$$

其运算法则：

$$0+0=0; \quad 0+1=1;$$
$$1+0=1; \quad 1+1=1。$$

与普通代数中的加法相比较，从形式上来说，布尔代数的加法运算法则中，只有 $1+1=1$ 这一项是和普通加法不相同的。但实质上，它们之间内容是根本不同的。普通代数中的加法的含意是数值的相加。其“0”表示“没有任何东西”，而“1”表示“单独的一个”，都是数值的大小，故 $1+1=2$。而在布尔代数中，加法是执行“或”逻辑的工具，而“0”和“1”仅仅只是表示某个逻辑命题的两种相反的结果。对于“或”逻辑来说，两个相同条件的同时存在和其中一个条件的存在是等价的。例如：发生火灾，有两个人去报警和一个人去报警的结果是一样的。因此，$1+1=1$。

由此，我们还可以得出以下一些推论：

显然有

$$A+0=A$$
$$A+1=1$$
$$A+A=A$$

对于逻辑乘来说，

$$x = A \cdot B$$

其运算法则

$$0\cdot 0=0; \quad 1\cdot 0=0;$$
$$0\cdot 1=0; \quad 1\cdot 1=1$$

从形式上来说，逻辑乘和普通代数中的乘法没有什么不同。因此，更应该强调它们二者之间在实质上是毫无相同之处的。在普通代数中乘法是数值的运算。例如：假设，一个人只有一支笔。$1\times 1=1$ 的含意是一个人共有一支笔。而在布尔代数中，逻辑乘是执行“与”逻辑的工具。$1\cdot 1=1$ 的含意是两个逻辑命题同时肯定的条件下，事件方能成立。

同样，不难得出以下推论：

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A \cdot A = A$$

在布尔代数中是不存在普通代数中的减法、除法和乘方的。

对于逻辑非来说，

$$x = \overline{A}$$

其运算法则更为简单

$$\overline{0} = 1$$

$$\overline{1} = 0$$

而在普通代数中是没有一个可以与逻辑非运算相对应的什么运算的。“$\overline{0}$”我们称之为“非0” 对于二值元素来说，不是“0”必是“1”。“$\overline{1}$”则称之为“非1”，自然就是“0”。若一变量A是0，则$\overline{A}$必是1，反之亦然。因而不难得出以下推论：

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

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

$$\overline{\overline{A}} = A$$

布尔代数的基本运算法则就是这些。读者也许会提出疑问：这不是太简单了吗？以后我们会了解用这些简单的法则来描述各种复杂的逻辑思维过程并不是一件很容易的事情。就像一块砖是比较简单的，然而，一座用砖块砌起来的摩天大楼就不那么简单了。

布尔代数的基本逻辑运算及其一些推论可以用文氏图比较直观地来加以证明。图1—16就是一个单变量的文氏图。整个矩形表示1的所有可能的集合。矩形中圆内的所有点表示变量A，则圆外的所有点为$\overline{A}$。

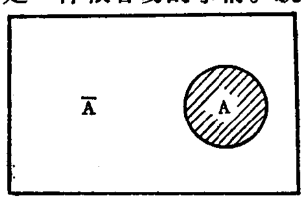

图1-16　单变量的文氏图

显然，由图1—16的文氏图，

即有：　　$A + \overline{A} = 1$

对于逻辑乘来说，文氏图如图1-17所示。图中左边圆内为变量A的集合，右边圆内为变量B的集合。两个圆的重叠处，画斜线的面积即表示A·B。

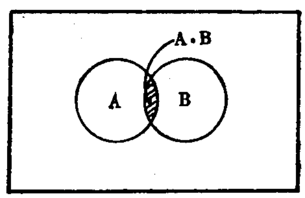

图1-17　A·B的文氏图

对于逻辑加来说，其文氏图如图1-18所示。两个圆联接的整个面积（画斜线的面积），即为A+B，当然两个圆也可不联接。两个圆的面积即为A+B

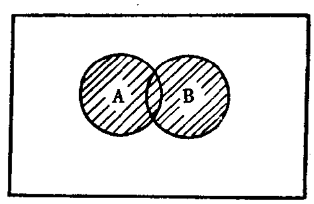

图1-18　A+B的文氏图

文氏图可以用来证明一些布尔代数式，它以直观明了的特点著称。但它仅限于用三个变量，超过三个变量

文氏图就失去了它的优点。

例如：推论 $A\cdot\overline{A}=0$

由图 1-16 单变量的文氏图可以直接得到证明。A 和 $\overline{A}$ 是永远不会有重叠的地方的。

2．布尔代数的基本定律

普通代数中的交换律、结合律、分配律也适用于布尔代数。必须再次强调的是，这种适用仅仅是在形式上相似，而实质内容是根本不同的。

① 交换律

加法的交换律：

$$A+B=B+A$$

乘法的交换律：

$$A\cdot B=B\cdot A$$

② 结合律

加法的结合律：

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

乘法的结合律：

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

以上二律的形式与普通代数完全相同，其结论是显然的。

③ 分配律

加法的分配律：

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

乘法的分配律：

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

乘法的分配律和普通代数乘法分配律形式相同，证明亦是显然的。但是加法的分配律在普通代数中就不存在了，此加法分配律若用于普通代数，肯定是错误的。

逻辑加法的分配律是需要证明一下的。布尔代数式的证明一般有两种方法：演绎法和归纳法。

演绎法：就是用已经得到证明的定律、公理、推论等来推理证明之。即用一般公认的理论去证明这个具体的特殊的公式。故演绎法是一种“由一般到特殊”的推理方法。

由加法分配律的右边：

$$\begin{aligned}(A+B)(A+C)&=A\cdot A+AB+AC+BC\\&=A+A(B+C)+BC\\&=A(1+B+C)+BC \qquad 1+B+C=1\\&=A+BC\end{aligned}$$

归纳法：对等式两边的式子，在变量所有各种组合情况下进行比较。如果两边完全相