半导体集成电路

天津科学技术出版社

内 容 简 介

本书是电子类技工学校教材，全书由三部分组成。第一部分（1～4章）介绍双极型数字集成电路；第二部分（5～8章）介绍MOS数字集成电路；第三部分（9～12章）介绍模拟集成电路基础、典型电路分析和版图。

本书在介绍各种集成电路时，以电路基本工作原理为主，注重基本概念，不引用过多的数学计算，

本书亦可作为有关工厂工人考工定级用书及职业中学、职工业余教育的教材。

本书前四章由林霞同志编写，后八章由姚金生同志编写。全书由姚金生同志修改定稿。清华大学王尔乾副教授担任了本书的主审工作。

全国电子类技工学校试用教材

半导体集成电路

姚金生　林霞　合编

责任编辑：吴孝钧

*

天津科学技术出版社出版

天津市赤峰道124号

天津新华印刷一厂印刷

新华书店天津发行所发行

*

开本 787×1092毫米 1/16　印张 16.5　字数 403,000

一九八五年二月第一版

一九八五年二月第一次印刷

印数：1－16,300

书号：15212·143　定价：2.10元

前　　言

为了适应技工学校电子类专业的教学需要，不断提高技工学校的培训质量，加速实现我国的四个现代化，国家劳动总局、第四机械工业部委托北京、天津、上海三市和四川、广东两省的劳动局、电子工业主管部门，组织编写了技工学校电子类三个专业（无线电技术、半导体器件、电子计算机）的部分技术基础课和专业课十二种教材。计有：电工基础、电子电路基础、电子测量与仪器、无线电接收设备、电视机原理调试与维修、无线电整机装配工艺基础、半导体器件制造工艺、半导体工艺化学、晶体管原理、制图与钳工知识、半导体集成电路、电子计算机原理。

这套教材对于二年制（招收高中毕业生）和三年制（招收初中毕业生）的技工学校均适用。这些专业的普通课教材没有另行编写，建议采用国家劳动总局和第一机械工业部委托上海市劳动局、上海市第一机电工业局一九七九年组织编写的全国技工学校机械类通用教材中的普通课教材。我们在组织这套教材的编写时，注意到了这两套教材在内容上的衔接。

根据技工学校的培养目标和教学计划的要求，这套教材在强调加强生产实习教学的同时，注意了加强基本理论知识和对新技术、新工艺的吸收。由于技工学校在教学范围内还有许多问题需要探讨，加之这套教材还没有通过教学实践的检验，故先作为试用教材出版发行。

因为时间仓促，编写经验不足，这套教材难免存在一些问题，恳切希望广大读者批评指正，以便作进一步修改。

国家劳动总局培训司
第四机械工业部教育局
一九八一年十二月

目　　录

第一章 逻辑代数

逻辑代数又称开关代数或布尔代数。最近几十年来，由于电话与数字系统，尤其是计算机的迅速发展，它已成为分析与设计开关电路的数学工具。

§1-1 二进制数的基本概念

一、二进制数的表示法

在日常工作和生活中，我们最常用、最熟悉的是十进制数，它有十个数码：0、1、2、3、4、5、6、7、8、9。它的进位制是“逢十进一”。

当用一组十进制数码表示数时，数码处于不同的位置，它所代表的数值是不同的。比如一个十进制整数7654，它的最右边第一位是个位（即10^0位），表示数值4；从右至左第二位为十位（即10^1位），表示数值50；从右至左第三位为百位（即10^2位），表示数值600，左边第一位为千位（即10^3位），表示数值7000。

十进制数7654可以表示为：

$$7654 = 7\times10^3 + 6\times10^2 + 5\times10^1 + 4\times10^0$$

7 6 5 4

——个位，表示 4

——十位，表示50

——百位，表示600

——千位，表示7000

和十进制数一样，当用一组二进制数码表示数时，处于不同位置的数码所代表的数值也是不同的。比如一个二进制数1001，它最右边一位为2^0位，从右至左第二位为2^1位，从右至左第三位为2^2位，最左边一位为2^3位。

二进制整数1001表示的量是：

$$1001 = 1\times2^0 + 0\times2^1 + 0\times2^2 + 1\times2^3 = 9$$

它表示十进制数 9 。二进制数与十进制数的比较如表1-1所示。

表 1-1

二 进 制	十 进 制
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9

下面介绍二进制小数的表示法。

我们知道，十进制数小数点右起第一位是10^{-1}位，第二位是10^{-2}位，第三位是10^{-3}位。例如，十进制小数0.765表示的量是：

$$0.765 = 7\times10^{-1} + 6\times10^{-2} + 5\times10^{-3}$$

0.765

- 5 —— 千分位，表示$\frac{5}{1000}$
- 6 —— 百分位，表示$\frac{6}{100}$
- 7 —— 十分位，表示$\frac{7}{10}$

和十进制小数相似，在二进制小数中，小数点右起第一位是2^{-1}位，第二位是2^{-2}位，第三位是2^{-3}位。例如，二进制小数0.1011表示的量是：

$$\begin{aligned}0.1011 &= 1\times2^{-1}+0\times2^{-2}+1\times2^{-3}+1\times2^{-4}\\ &=0.5+0.125+0.0625\\ &=0.6875\end{aligned}$$

二进制小数与十进制小数对照表如表1-2所示。

二、二进制的特点

为什么电子计算机一般都采用二进制呢？

首先，二进制只取0和1两个数码，因此它的每一位数都可以用具有两个稳定状态的元件来表示。比如，可以用继电器的闭合表示“0”，继电器的断开表示“1”；晶体管的截止表示“1”，晶体管的导通表示“0”；可以用高电位表示“1”，低电位表示“0”等，如图1-1所示。一般说来，制造具有两个稳定状态的元件比制造多个稳定状态的元件容易得多。而一个十进制数却要用具有十个稳定状态的元件来表示，所以用二进制要比采用十进制方便得多。

表 1-2

二 进 制	十 进 制
0.1	0.5
0.01	0.25
0.001	0.125
0.0001	0.0625

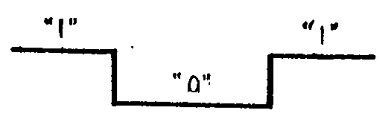


图 1-1 逻辑“0”、“1”表示法

其次，二进制数运算简单。对十进制数作算术运算，必须熟记加法口诀（如3 + 4 = 7，3 + 5 = 8 …）和乘法九九表等口诀。对于二进制数，因为它只有两个数码，一位二进制数的加法和乘法只有以下四种情况：

二进制的加法组合为：

$$0+0=0$$
$$1+0=0+1=1$$
$$1+1=10$$

二进制的乘法组合为：

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

因此，二进制实现加法和乘法比十进制实现加法和乘法要简单。

三、十进制数与二进制数之间的相互转换

1.十进制整数转换成二进制整数

将十进制整数转换成二进制整数，采用“除二取余”法。所谓“除二取余”法，就是将十进制整数不断用 2 去除，将每除一次得到的余数按低位到高位的次序排列起来，就是所要求的二进制整数。

例如，把十进制数725转换成二进制数

2 | 725	余数 = 1	（最低位）
2 | 362	余数 = 0	（次低位）
2 | 181	余数 = 1	
2 | 90	余数 = 0	
2 | 45	余数 = 1	
2 | 22	余数 = 0	
2 | 11	余数 = 1	
2 | 5	余数 = 1	
2 | 2	余数 = 0	（次高位）
2 | 1	余数 = 1	（最高位）
0		

较换的结果为：

$$(725)_{十进制} = (1011010101)_{二进制}$$

2.十进制小数转换成二进制小数

将十进制小数转换成二进制小数，采用“乘 2 取整法”。所谓“乘 2 取整法”，是将十进制小数不断用 2 去乘，将每乘一次得到的整数部分按高位到低位的次序排列起来，就是所要求的二进制小数。

例如，把十进制小数0.6875转换成二进制小数：

$$\begin{array}{r} 0.6875 \\ \times \quad\quad 2 \\ \hline 1.3750 \end{array} \quad 整数部分 = 1$$

$$\begin{array}{r} 0.3750 \\ \times \quad\quad 2 \\ \hline 0.7500 \end{array} \quad 整数部分 = 0$$

$$\begin{array}{r} \times \quad\quad 2 \\ \hline 1.5000 \end{array} \quad 整数部分 = 1$$

$$\begin{array}{r} 0.5000 \\ \times \quad\quad 2 \\ \hline 1.0000 \end{array} \quad 整数部分 = 1$$

转换结果为：

$$(0.6875)_{十进制} = (0.1011)_{二进制}$$

如果一个数既有整数部分，又有小数部分，则可将整数部分与小数部分分别进行转换。例如：

$$(2.25)_{十进制} = (10.01)_{二进制}$$

3.二进制数转换成十进制数

二进制数转换成十进制数是十分方便的，只需将二进制数写成前面所讲的“展开式”，

就能得到所要求的十进制数。

例如，要将二进制数（101011.1001）转换成十进制数：

$$
\begin{aligned}
(101011.1001)_{\text{二进制}} &= 1\times2^5+0\times2^4+1\times2^3+0\times2^2+1\times2^1+1\times2^0\\
&\quad +1\times2^{-1}+0\times2^{-2}+0\times2^{-3}+1\times2^{-4}\\
&=32+8+2+1+0.5+0.0625\\
&=(43.5625)_{\text{十进制}}
\end{aligned}
$$

§1-2 十进制数的二进制编码表示法

十进制数除了可按前一节所讲过的方法转换成二进制数外，还可以用另一种方法来表示，那就是十进制数的二进制编码表示法。

这种表示法是利用四个二进制数码来表示十进制数的每一个数码。这样，它既具有二进制数的形式，又具有十进制数的特点。二进制数码表示法有许多种形式，在这里只介绍最常用的“8421”码，它也称“BCD”码。

“8421”码用四位二进制数来表示一个十进制码。这四位二进制码自左至右其值和二进制数是一样的，即分别是8、4、2、1。“8421”码的名称就是由此而来的。例如，“8421”码1001就代表

$$1\times2^3+0\times2^2+0\times2^1+1\times2^0=9。$$

表1-3是一位十进制数的“8421”码的表示法。

表 1-3

十进制数	8 4 2 1码
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

用“8421”码来表示十进制数的优点是既简单又方便。例如，十进制数427的“8421”码就是

$$\underbrace{0100}_{4}\quad\underbrace{0010}_{2}\quad\underbrace{0111}_{7}$$

“8421”码1001 10000100就是984。

在“8421”码中，代码1010～1111是不允许出现的，因为它们不能和一个十进制的单个数字符号相对应。

§1-3 逻辑代数的基本原理

一、基本逻辑运算

逻辑代数和普通代数一样，也可以用字母A、B、C、$D\cdots X$、Y、Z来表示变量，但是逻辑代数的变量的取值只有“0”或“1”两种。此外，在逻辑代数中，基本运算比较少，只有三种基本运算：与、或、非。以下分别介绍这三种基本运算：

1.或运算

它又称“逻辑加”运算。在图1-2的开关电路中，有两个并联开关A和B，它们控制一只电灯F。不难看出，当开关A接通或开关B接通（也包括A和B都接通），电灯F亮；开

关A和开关B都不接通，灯不亮，或者说，灯F亮的条件是开关A接通或开关B接通。

开关只有通和断两种状态，电灯也只有亮和不亮两种状态。如果用“0”表示开关的“断开”状态，用“1”表示开关的“接通”状态；用“0”表示电灯不亮，用“1”表示电灯亮；此外，用符号“+”表示或关系（要注意，“括”号读成或，它不是算术运算中的“加”），那么，上述关于电灯亮和不亮的条件可用以下四个式子表示：

$$\left.\begin{aligned}0+0&=0\\1+0&=1\\0+1&=1\\1+1&=1\end{aligned}\right\}\qquad(1\text{-}1)$$

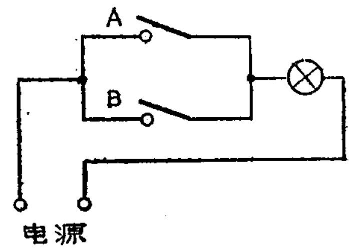


图 1-2 并联开关电路

式中“+”号左边的数字代表开关A的状态，“+”号右边的数字代表开关B的状态，等号右边代表电灯F的状态。电灯F亮和不亮的条件还可用表1-4(a)中F和A、B的关系来表示，习惯上称它为真值表。应该说，表1-4(a)和式(1-1)所代表的内容是完全一致的，只是表达形式不同而已。我们把由式（1-1）或由表1-4(a)所表达的运算称为或运算。将描述A、B和F关系的式（1-1）合并写成如下形式：

$$A+B=F \qquad (1\text{-}2)$$

并把它称为两输入的或运算的一般表达式。在式（1-2）中，当A、B取一组值后，F就有另一个确定的值与之相对应，所以F是A、B的或逻辑函数。我们把F称为输出逻辑变量，A、B称为输入逻辑变量。

如果图1-2中有三个开关A、B、C并联，那么描述电灯和开关关系的表达式应是一个三输入变量的或逻辑函数表达式：

$$F=A+B+C$$

三输入或运算真值表如表1-4(b)所示。

表 1-4(a)

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

表 1-4(b)

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

或逻辑功能可用图1-2所示的开关电路来实现，也可用电子电路来实现。实现或运算的电子线路称为或门。图1-3是几种常用的或门符号，其输入、输出之间的关系为：

$$F=A+B+C$$

2.与运算

它又称“逻辑乘”运算。如果用两只串联开关A、B控制一只电灯F，如图1-4所示，那么，开关A、B接通或断开和电灯F亮灭之间的关系是：只要A、B中有一个断开（也包括A、B都断开），F就灭；只有当A与B都接通，F才亮。如果我们仍用“0”表示开关断

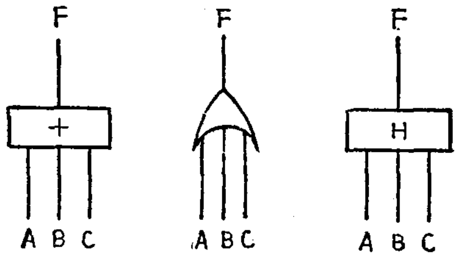


图 1-3 或门符号

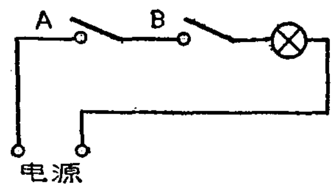


图 1-4 串联开关电路

开和灯不亮；用"1"表示开关接通和灯亮；用"·"表示与，则上述关于电灯亮和不亮的条件可以用以下四个式子表示：

$$\left.\begin{array}{l}0\cdot 0=0\\1\cdot 0=0\\0\cdot 1=0\\1\cdot 1=1\end{array}\right\}\qquad(1-3)$$

（要注意，"·"是逻辑代数中的与，不是算术运算中的乘）如果把上述四个式子用表1-5(a)来描述，那么，它就是与运算的真值表。描述A、B和F关系的式（1-3）可用下式来概括：

表 1-5(a)

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

表 1-5(b)

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

$$A\cdot B=F\qquad(1-4)$$

并把它称为两输入与逻辑函数表达式

如果图1-4中有三个开关串联，那么应当用三输入与表达式来描述：

$$F=A\cdot B\cdot C$$

表1-5(b)为三输入与运算真值表。

用电子线路实现与运算的电路称为与门。图1-5是几种常用的与门图形符号，其输入、输出之间的关系为：

$$F=A\cdot B\cdot C$$

3.非运算

它又称反相运算。如果用一只"单刀双掷"开关控制两个灯A和B，如图1-6所示，当开关上合，则灯A亮，B灭；开关下合，则灯B亮，A灭，即A的状态和B的状态是"相反"的。

若用"0"表示灯灭，用"1"表示灯亮，则A、B状态的关系可用表1-6表示：

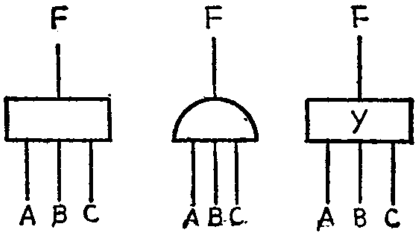


图 1-5 与门图形符号

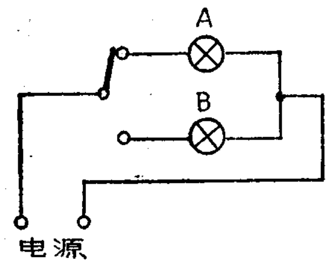


图 1-6 单刀双掷开关电路

表 1-6

A	B
0	1
1	0

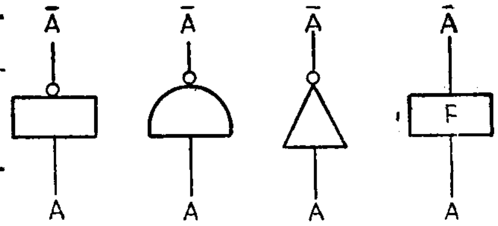


图 1-7 非门符号

我们将这个关系用下式表示：

$$\left.\begin{aligned} A &= \bar{B} \\ B &= \bar{A} \end{aligned}\right\} \tag{1-5}$$

这就是非逻辑函数的表达式。这里，A、B上面的一横读作非。

实现非运算的电路称为“非”门。图1-7是非门的常用符号。

人们常把A称为原变量，称$\bar{A}$为A的反变量。

二、复合逻辑运算

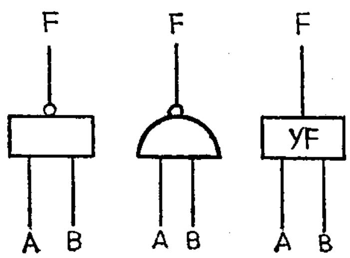
上面讲了三种基本逻辑运算——或、与、非。从这三种基本运算可以引伸出很多复合逻辑运算。

1.与非逻辑运算

与非逻辑是与逻辑和非逻辑的组合，与非逻辑的输入变量A、B和输出变量F之间有以下关系：

$$F = \overline{A \cdot B} \tag{1-6}$$

实现与非逻辑的门电路称为与非门，它由与门和非门串联而成，图1-8给出了它的符号和它的真值表。



“与非”逻辑运算真值表

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

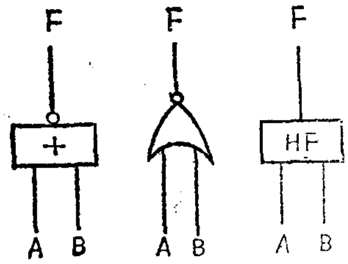
图 1-8 与非门的符号和真值表

2.或非逻辑运算

或非逻辑是或逻辑和非逻辑的组合。或非逻辑的输入变量A、B和输出变量F之间有以下关系：

$$F = \overline{A + B} \tag{1-7}$$

实现或非逻辑的门电路称为或非门，它是由或门和非门串联而成的。图1-9给出了它的符号



"或非"逻辑运算真值表

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

图 1-9 或非门符号和真值表

和真值表。

3. 与或非逻辑运算

与或非逻辑实现以下运算：

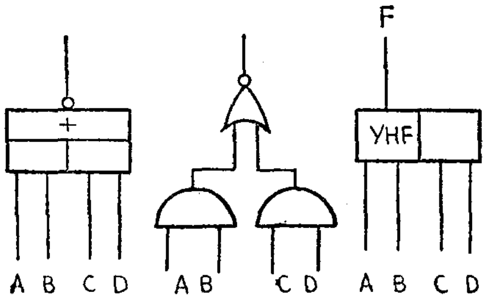
$$F=\overline{A\cdot B+C\cdot D} \tag{1-8}$$

它包含与、或及非三种运算。

实现与或非逻辑的与或非门由与门、或门和非门串联而成。图1-10给出了它的逻辑符号和真值表。

4. 异或非逻辑（又称同或逻辑）运算

异或非逻辑在日常生活中是常常会遇到的。例如，装在楼梯上的电灯，为了开灯和关灯的方便，常常在楼上、楼下各装一个"单刀双掷"开关，如图1-11所示。上楼时在楼下开灯，照亮楼梯，上楼后顺手用楼上的开关关掉电灯；下楼时在楼上开灯，照亮楼梯，下楼后



"与或非"
逻辑运算真值表

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

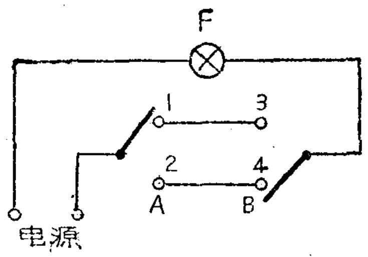


图 1-11 具有异或非关系的开关电路

图 1-10 与或非门符号和真值表

顺手用楼下的开关关掉电灯。在图1-11中，开关A表示楼上的开关，开关B表示楼下的开关，F表示电灯。当A与B都扳到上面的位置（即A处于位置1，同时B处于位置3），或者A与B都扳到下面的位置（即A处于位置2，同时B处于位置4）时，F就亮。当开关A、B

的位置不处于上述情况（即A扳到上面位置，B扳到下面位置，或A扳到下面位置，B扳到上面位置）时，电灯不亮。如果仍以$F=0$表示灯灭，$F=1$表示灯亮；“1”表示开关处于上面位置，“0”表示开关处于下面位置，那么，描述电灯F状态和A、B状态关系的表达式为：

$$F=A\cdot B+\overline{A}\cdot\overline{B} \tag{1-9}$$

它所表达的意思是：当A与B同时处于上面的位置（即“$A\cdot B$”项为“1”），或者（即“+”号）当A与B都不处于上面的位置（即$\overline{A}\cdot\overline{B}$为“1”项），则灯亮（即$F=1$）。而其余情况下灯灭。式（1-9）就是异或非的逻辑表达式。由图1-12给出了异或非逻辑的符号及其真值表。由真值表可看到，只有当A、B状态相同时，输出才为“1”，A、B状态相异时，输出为“0”。具有这种特点的逻辑又称同或逻辑。

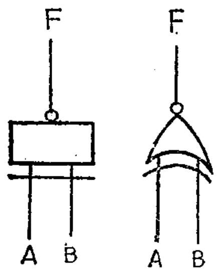
5. 异或逻辑运算

异或逻辑是异或非逻辑及非逻辑的组合，其逻辑表达式是：

$$F=\overline{A\cdot B+\overline{A}\cdot\overline{B}} \tag{1-10}$$

图1-13给出了它的符号及其真值表。由表可知，只有当A、B状态相异时，输出才为“1”。因此把具有这种特点的逻辑称为异或逻辑。

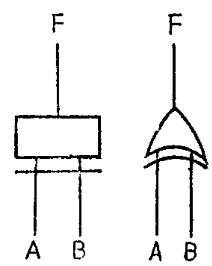
用基本逻辑还可以组成很多其它的复合逻辑。



异或非运算真值表

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

图 1-12 异或非门的符号和真值表



异或逻辑运算真值表

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

图 1-13 异或门图形符号和真值表

三、逻辑函数的表示方法与逻辑图

上面已讲到过，逻辑运算有两种表示方法，函数表示法及真值表表示法。

逻辑函数表达式是由输入逻辑变量的与、或及非逻辑运算组合而成的。例如，$F=A\cdot B+\overline{A}\cdot\overline{B}$就是这样的表达式。

真值表是由逻辑变量的各种可能取值和相应的输出函数值列成的表。表1-7就是逻辑函数式$F=A\cdot B+\overline{A}\cdot\overline{B}$的真值表。

逻辑函数表达式和真值表是逻辑函数的两种不同的表示方法，它们在本质上是相同的，两者可以互相转换。按照逻辑式子，对变量的各种可能取值进行运算，求出相应的函数值，把变量取值和函数值一一对应，列成表格，就得到真值表。上述逻辑函数式中有两个输入变量，它们的取值的可能组合有$2^2=4$个，即：$A=0$；$B=0$；$A=0$，$B=1$；$A=1$，$B=0$；$A=1$，$B=1$。按照逻辑式进行运算，当$A=0$、$B=0$时，得$F=1$；当$A=0$、$B=1$时，得$F=0$；当$A=1$、$B=0$时得$F=0$；当$A=1$、$B=1$时，得$F=1$。这样就得到表1-7所示的真值表。反之，由真值表也可以很容易地得到逻辑表达式，其方法为：把真值表中函数等于“1”的变量组合挑出来，变量值是“1”的写成原变量，是“0”的写成原变量的非值，把组合中的各个变量相与，然后把各个与项相或，就得到了相应的逻辑表达式。例如，由表1-7所示的真值表中只有第一栏和第四栏的函数值F等于“1”，我们把它们挑出

来。在第一栏中，输入变量A、B的取值都是“0”，因此，把它们写成$\overline{A}$、$\overline{B}$，再把它们相与，得$\overline{A}\cdot\overline{B}$。在第四栏中，$A$、$B$取值都为“1”，因此都把它们写成原变量，即$A$、$B$，再把它们相与得$A\cdot B$。最后，再把这两个与项$\overline{A}\cdot\overline{B}$、$A\cdot B$相或，就可以写出$F=\overline{A}\cdot\overline{B}+A\cdot B$。

表 1-7

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

表 1-8

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

下面再举一个由真值表变为逻辑表达式的例子。例如，把表1-8变为逻辑表达式。在这张表中只有第二栏、第三栏的函数值F为“1”。在第二栏中，A取值为“1”，B取值为“0”，所以把它们写成A和$\overline{B}$，再把它们相与，得$A\cdot\overline{B}$。在第三栏中，A取值为“0”，B取值为“1”，所以把它们写成$\overline{A}$和B，它们相与，得$\overline{A}\cdot B$，最后，把$A\cdot\overline{B}$和$\overline{A}\cdot B$相或，即得函数表达式：

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

前面已经讲过，可以用门电路的逻辑图符号来表示逻辑表达式，所得到的图叫做逻辑图。每一张逻辑图，其输出与输入之间都具有确定的逻辑函数关系。逻辑图的输入，就是函数的输入逻辑变量，逻辑图的输出就是函数的输出逻辑变量。反之，一张逻辑图也可以用相应的逻辑函数来表示。所以可以把逻辑图看作是表示逻辑函数的一种方法。下面我们通过几个例子来说明逻辑图和逻辑函数是如何转换的。

例 1　列出图1-14所示的逻辑图的逻辑函数。

这张逻辑图由两个与非门（门1、2）和一个或非门（门3）组成，我们由输入至输出逐级写出输出端的逻辑表达式：

与非门1的输出表达式　$F_1=\overline{A\cdot B}$

与非门2的输出表达式　$F_2=\overline{C\cdot D}$

最后，写出或非门3的输出表达式，即这张逻辑图的逻辑函数为：

$$F=\overline{F_1+F_2}=\overline{\overline{A\cdot B}+\overline{C\cdot D}}$$

例 2　列出图1-15所示的逻辑图的逻辑函数。

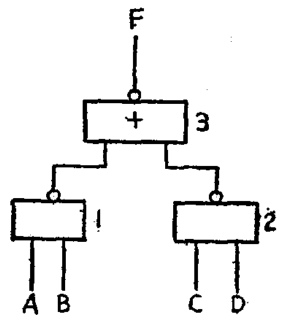


图 1-14　与或非逻辑函数的逻辑图

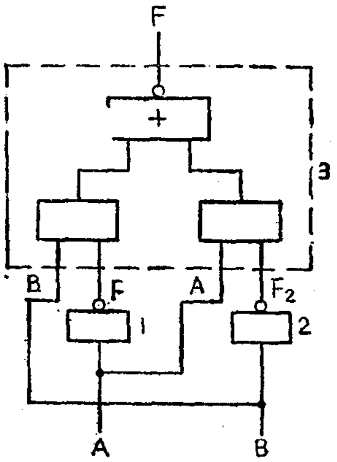


图 1-15　与或非逻辑函数的逻辑图

这张逻辑图由两个非门（门 1 、 2 ）和一个与或非门（门 3 ）组成。先写出门 1 、 2 的输出表达式：

$$F_1=\overline{A}$$
$$F_2=\overline{B}$$

然后再写与或非门 3 的输出表达式：

$$F=\overline{F_1\cdot B+A\cdot F_2}=\overline{\overline{A}\cdot B+A\cdot\overline{B}}$$

这就是这张逻辑图的输出函数。

下面我们用几个例子来说明如何从逻辑函数画出逻辑图。

例 1　画出$F=A\cdot B+\overline{A}\cdot\overline{B}$的逻辑图。

上式中（$A\cdot B$）是与运算，用与门实现；$\overline{A}$是A的非运算；$\overline{B}$是B的非运算，它们都用非门实现；（$\overline{A}\cdot\overline{B}$）是与运算，用与门实现；$A\cdot B+\overline{A}\cdot\overline{B}$表示对两个与门的输出进行或运算；用或门实现。这样就得到了实现逻辑函数$F=A\cdot B+\overline{A}\cdot\overline{B}$的逻辑图（图1-16）。

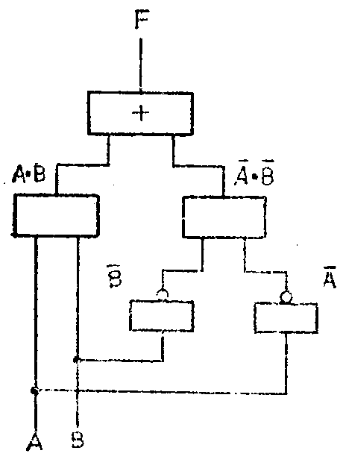


图 1-16　逻辑函数$F=A\cdot B+\overline{A}\cdot\overline{B}$的逻辑图

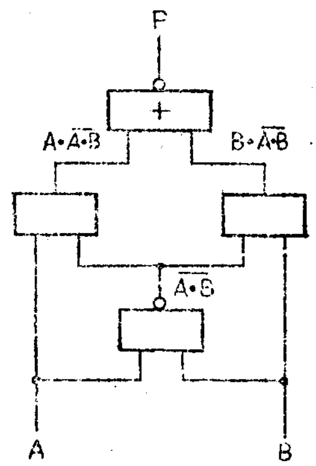


图1-17　逻辑函数$F=\overline{A\cdot\overline{A\cdot B}+B\cdot\overline{A\cdot B}}$的逻辑图

例 2　画出$F=\overline{A\cdot\overline{AB}+B\cdot\overline{A\cdot B}}$的逻辑图。

上式中，$\overline{A\cdot B}$用与非门实现，$\overline{A\cdot\overline{A\cdot B}+B\cdot\overline{A\cdot B}}$ 用与或非门实现，这样就可以得到如图1-17所示的逻辑图。

四、逻辑代数的基本公式

逻辑代数是针对二进制数的代数学。这一节我们将着重讨论逻辑代数的基本运算公式。通过这些公式可以了解逻辑代数的基本规律。这些公式有一些和初等代数相似，有些则完全不同。

1．0　1律

公式 1　$A+0=A$　　公式 1′　$A\cdot 1=A$

公式 2　$A+1=1$　　公式2′　$A\cdot 0=0$

公式 3　$A+\overline{A}=1$　　公式3′　$A\cdot\overline{A}=0$

2．交换律、结合律、分配律

公式 4　$A+B=B+A$　　　　（交换律）

公式4′　$A\cdot B=B\cdot A$　　　　（交换律）

公式 5　$(A+B)+C=A+(B+C)$　（结合律）

公式5′　$(A\cdot B)\cdot C=A\cdot(B\cdot C)$　（结合律）

公式 6　$A\cdot(B+C)=A\cdot B+A\cdot C$　（分配律）

公式6′　$A+(B\cdot C)=(A+B)\cdot(A+C)$　（分配律）

3.逻辑代数的一些特殊规律

公式 7　$A+A=A$　（重叠律）

公式7′　$A\cdot A=A$　（重叠律）

公式 8　$\overline{A+B}=\overline{A}\cdot\overline{B}$　（反演律）

公式8′　$\overline{A\cdot B}=\overline{A}+\overline{B}$　（反演律）

公式 9　$\overline{\overline{A}}=A$　（对合律）

这些等式的正确性都可以用真值表加以证明，其方法如下：

对变量的各种可能的组合取值，如果等号两边式子的值相等则等式成立，反之，等式就不成立。这里作为例子将能证明公式6′和公式 8 。

证明公式6′　$A+B\cdot C=(A+B)\cdot(A+C)$的正确性。

列出等号两边的式子的真值表，见表1-9。

由表中可以看出，输入变量共有三个，它们的组合可能有八种，等号两边的式子$A+B\cdot C$和$(A+B)\cdot(A+B)$，在变量前三种情况下都等于“0”，在后五种情况下都等于“1”，它们的真值表是相同的，所以公式成立。

证明公式 8，$\overline{A+B}=\overline{A}+\overline{B}$。

表1-10是等号两边式子的真值表。由表中可以看出，等号两边的式子$\overline{A+B}$和$\overline{A}\cdot\overline{B}$的真值表是相同的，所以公式成立。

表 1-9

输入	等式左边		等式右边		
ABC	$B\cdot C$	$A+B\cdot C$	$A+B$	$A+C$	$(A+B)\cdot(A+C)$
0 0 0	0	0	0	0	0
0 0 1	0	0	0	1	0
0 1 0	0	0	1	0	0
0 1 1	1	1	1	1	1
1 0 0	0	1	1	1	1
1 0 1	0	1	1	1	1
1 1 0	0	1	1	1	1
1 1 1	1	1	1	1	1

表 1-10

输入	等式左边		等式右边		
A　B	$A+B$	$\overline{A+B}$	$\overline{A}$	$\overline{B}$	$\overline{A}\cdot\overline{B}$
0　0	0	1	1	1	1
0　1	1	0	1	0	0
1　0	1	0	0	1	0
1　1	1	0	0	0	0

五、逻辑代数的常用公式

运用基本公式和上述基本规则，可以得到更多的公式。下列公式在实践中是经常遇到的。

公式10　$A\cdot B+A\cdot\overline{B}=A$

证明：　$AB+A\overline{B}=A\cdot(B+\overline{B})=A\cdot 1=A$

公式11　$A+AB=A$

证明：　$A+AB=A\cdot(1+B)=A\cdot 1=A$

公式11说明，如果一个变量和一个与项相或，而与项中又包含这个变量，那么这个与项