第 1 章

逻辑事件及其表示方法

内容提要

数字电路广泛应用于各个领域。数字电路又称逻辑电路,是研究输入、输出之间逻辑关系的学科。本章通过实训,首先初步认识逻辑控制电路,然后通过理论叙述,介绍数字电路中基本的逻辑关系及其表示方法、数字电路中常见的数制,最后介绍逻辑代数的基本知识和逻辑函数的化简。

实训 1　信号灯的逻辑控制

1. 实训目的

(1) 了解逻辑控制的概念。

(2) 掌握表示逻辑控制的基本方法。

2. 实训设备和器件

发光二极管、限流电阻、继电器(两个)、直流电源、导线若干。

3. 实训电路图

图 1.1 为实训电路图。这是一个楼房照明灯的控制电路。设 A、B 分别代表上、下楼层的两个开关,发光二极管 F 代表照明灯。在楼上按下开关 A,可以将照明灯打开,在楼下闭合开关 B,又可以将灯关掉;反过来,也可以在楼下开灯,楼上关灯。

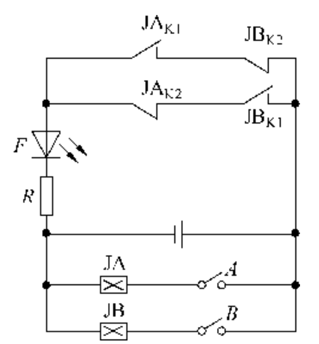


图 1.1　照明灯的逻辑控制电路

4. 实训步骤与要求

(1) 连接电路

按图 1.1 连接好电路,注意 JA、JB 两个继电器的开关不要接错。

(2) 试验开关和发光二极管的逻辑关系

接通电源,分别将开关 A、B 按表 1.1 的要求接通或者断开,观察发光二极管 F 的亮灭情况,并填入表 1.1 中。

5. 实训总结与分析

表 1.1

开关 A	开关 B	发光二极管 F
断开	断开	
断开	闭合	
闭合	断开	
闭合	闭合	

通过上述实训,可做如下总结:

(1) 图 1.1 中,JA 和 JB 分别代表继电器的两个线圈,JA_{K1}、JB_{K1} 代表继电器的常开触点,JA_{K2}、JB_{K2} 代表继电器的常闭触点。在实训图所示的状态下(开关 A、B 均断开),由于没有通路给发光二极管供电,所以发光二极管灭;开关 A 闭合,继电器线圈 JA 通电,其常开触点 JA_{K1} 闭合,常闭触点 JA_{K2} 断开,JB_{K1}、JB_{K2} 则维持原来状态,此时图 1.1 最上面的一条电路连通,通过电源给发光二极管供电,发光二极管亮。同样道理,如果只闭合开关 B,也会给发光二极管构成通路使之点亮;当开关 A、B 均闭合时,由于没有通路,所以发光二极管灭,读者可自行分析。

(2) 发光二极管 F 的状态,我们称为输出,是由开关 A、B 来决定的,开关 A、B 称为输入。输出和输入是一种逻辑控制电路,而且输入量和输出量都只分别对应两种状态。

(3) 从试验结果可以看出,当 A、B 同时闭合,或者同时断开,即处于相同状态时,二极管灭;相反,当 A、B 处于不同状态时,发光二极管点亮。如果定义开关闭合和灯亮为逻辑“1”,定义开关断开和发光二极管不亮为逻辑“0”,则 A、B、F 都可用两种逻辑状态“1”、“0”来描述。注意此时的“1”、“0”不代表任何数量的大小。表格的左边是两个输入状态的所有取值的组合,表格的右边是对应的输出状态。这样我们可以将实验步骤 4 得到的表 1.1 重新写为表 1.2,这种表征逻辑事件输入和输出之间全部可能状态的表格称为逻辑事件的真值表。

表 1.2

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

1.1 逻辑事件的数制转换

1.1.1 逻辑事件与逻辑控制

通过实训 1,我们初步认识了一个逻辑事件的控制电路。所谓逻辑,简单地说,就是表示事物的因果关系,即输入、输出之间变化的因果关系。而逻辑事件是这样的一类事物,它们具有如下共性:其存在或表现形式有且仅有两个相互对立的状态,而且它必定出现在这两个状态中的一个。例如:实训中的开关只有“闭合”和“断开”两种状态,而且开关的状态必为二者之一;发光二极管只有“亮”、“灭”两种对立状态。再例如:生物的活与死;射击导弹的击中目标与未击中目标;竞选的成功与失败;外星人的存在与不存在……上述事件都是逻辑事件,又可以叫做逻辑量。

在现实生活中的一些实际关系,会使某些逻辑量的取值互相依赖,或互为因果。例如实训中开关的通断决定了发光二极管的亮、灭,反过来也可以从发光二极管的状态推出开关的相应状态,这样的关系称为逻辑控制。

在实际应用中,会遇到各种复杂的逻辑控制电路,但它们都是由基本的逻辑关系组成的。在数字电路中,有一些基本的逻辑控制电路,反映了这些基本的逻辑关系(又称逻辑

运算）。这些基本的逻辑运算是构成各种复杂数字逻辑电路的基础。下面分别讨论几种基本的逻辑关系。

1.1.2　逻辑事件中常用的数制

在上述讨论的各类逻辑事件中，有且仅有两个相互对立的状态是其最重要的特点，这两个对立的状态在数字逻辑电路中往往用“0”、“1”来表示。只有 0、1 两种状态的数字，我们称之为二进制数，二进制数是数字逻辑电路、计算机技术的基础。

在实际应用中，常用的数制有十进制、二进制、八进制和十六进制。数制有三个要素：基、权、进制。

基：数码的个数。例如，十进制数的基为 10。

权：数码所在位置表示数值的大小。例如，十进制数每一位的权位为 10^n。

进制：逢基进一。例如，十进制数是逢十进一。

日常生活中，十进制数最为常用，以 1234 这个数为例，按位展开后是：

$$1234=1\times10^3+2\times10^2+3\times10^1+4\times10^0$$

其中，10^3、10^2、10^1、10^0 是十进制数各位的权值。下面我们对其他几种进制加以简单介绍。

1． 二进制数（**Binary**）

在数字电路中，应用最广泛的是二进制数。二进制数的每一位仅有 0 和 1 两个数码，在计数时低位和相邻高位之间的进位关系是“逢二进一”，借位关系是“借一当二”。基数是 2，它的位权是由基数 2 的幂决定的。在表示时，二进制数的后面加上字母 B，以和十进制数区别，因此，一个二进制数可以展开表示为：

$$1101\text{B}=1\times2^3+1\times2^2+0\times2^1+1\times2^0$$

可以看出，二进制数每一位的权值是……2^3、2^2、2^1、2^0。

在数字电路和计算机内部，常常采用二进制，因为二进制只有 0 和 1，采用开关的通、断和脉冲的高、低电平都很容易用它来表示。此外，二进制数运算简单，便于用后续的电子线路实现。

2． 十六进制数（**Hex**）

十六进制数是人们研究和学习二进制数的一种工具，它是随着计算机的发展而广泛应用的。十六进制的基数为 16，它有 0～9，A，B，C，D，E，F 十六个数码，计数规则是“逢十六进一”，“借一当十六”，它的位权是由基数 16 的幂决定的。在表示时，十六进制数的后面加上字母 H，因此，一个十六进制数可以展开表示为：

$$\begin{aligned}3\text{CB0H}&=3\times16^3+\text{C}\times16^2+\text{B}\times16^1+0\times16^0\\&=3\times16^3+12\times16^2+11\times16^1+0\times16^0\end{aligned}$$

可以看出，十六进制数每一位的权值是……16^3、16^2、16^1、16^0。

3． 八进制数（**Octal**）

八进制的基数为 8，它有 0～7 八个数码，计数规则是“逢八进一”，“借一当八”，它的位权是由基数 8 的幂决定的。在表示时，十六进制数的后面加上大写字母 O，因此，一个八进制数可以展开表示为：

$$3245O=3\times8^3+2\times8^2+4\times8^1+5\times8^0$$

1.1.3 常用的数制间的相互转换

逻辑电路和计算机中普遍采用二进制数操作，但人们习惯使用十进制，这样往往会遇到不同数制之间的转换。图1.2表示了十进制、二进制和十六进制数之间的转换方法。

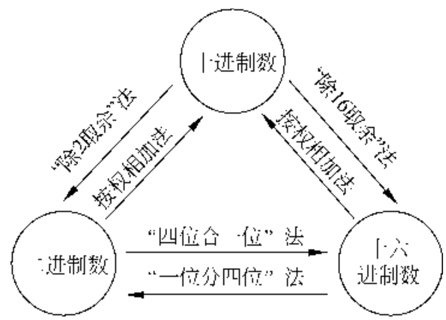


图1.2 三种数制间数的转换方法示意图

1. 二进制数和十进制数的转换

(1) 二进制数转换成十进制数

二进制数转换为十进制数方法很简单，只需要把欲转换的二进制数按位展开相加即可，例如：

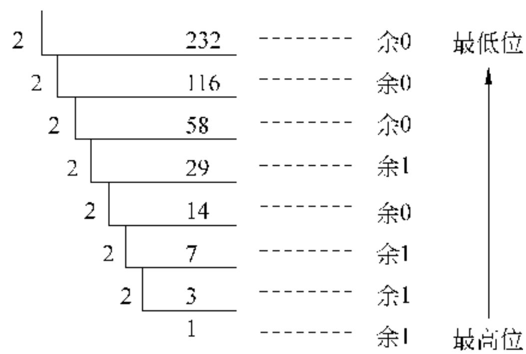
$$\begin{aligned}10101101B&=1\times2^7+0\times2^6+1\times2^5+0\times2^4\\&\quad+1\times2^3+1\times2^2+0\times2^1+1\times2^0\\&=173\end{aligned}$$

(2) 十进制数转换成二进制数

常用的方法是"除2取余"法，该法则是用2连续去除要转换的十进制数，直到商小于2为止，然后把各次余数按最后得到的为最高位、最早得到的为最低位，依次排列起来所得到的数便是所求的二进制数，现举例加以说明。

例 求出十进制数232的二进制数。

解 把232连续除以2，直到商数小于2，相应竖式为：



把所得余数按箭头方向从高到低排列起来便可得到：

$$232=11101000B$$

2. 十六进制数和十进制数的转换

(1) 十六进制数转换成十进制数

十六进制数转换成十进制数方法和二进制数转换为十进制数相似，即把欲转换的十六进制数按权展开相加即可，例如：

$$3CB0H=3\times16^3+12\times16^2+11\times16^1+0\times16^0=15536$$

(2) 十进制数转换成十六进制数

与十进制数转换为二进制数类似，用的方法是"除16取余法"，即用16连续去除要转换的十进制数，直到商小于16为止，然后把各次余数按最后得到的为最高位、最早得到的

为最低位，依次排列起来所得到的数便是所求的十六进制数。

例　求 3256 所对应的十六进制数。

解　把 3256 连续除以 16，直到最后，相应竖式为：

16	3256	--------	余8	写成8	最低位
16	203	--------	余11	写成B	↑
	12	--------	余12	写成C	最高位

把所得余数按箭头方向从高到低排列起来便可得到：

$$3256 = \mathrm{CB8H}$$

3. 二进制数和十六进制数的转换

二进制数和十六进制数之间的转换十分方便，这也是人们为什么采用十六进制数来表示二进制数的原因。

(1) 二进制转换为十六进制

采用“四位合一位”的方法，即将二进制数从最低位开始，依次向高位划分，每 4 位一组，当不足 4 位时在高位以 0 补上，然后每组用一个十六进制数表示，按顺序相连即可。

例　把 10111010110 转换为十六进制数。

解　将二进制数 4 个一组划分，然后写为十六进制数即可。

不足 4 位，高位补 0

$$\downarrow$$
$$\underbrace{0101}_{5}\,\underbrace{1101}_{D}\,\underbrace{0110}_{6}$$

所以，相应的十六进制数为 5D6H。

(2) 十六进制转换为二进制

采用“一位分四位”的方法，与上述方法刚好相反。

例　将十六进制数 4B2CH 转换为一个二进制数，则为：

$$\begin{array}{cccc} 4 & B & 2 & C \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 0100 & 1011 & 0010 & 1100 \end{array}$$

所以，4B2CH＝0100101100101100B。

表 1.3 为十进制、二进制、十六进制常用数之间的对应关系。

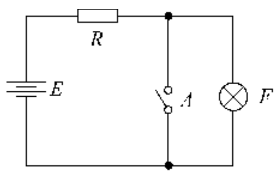
表 **1.3**　常用数转换表

十进制数	二进制数	十六进制数	十进制数	二进制数	十六进制数
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2		1010	A
3	0011	3		1011	B
4	0100	4		1100	C
5	0101	5		1101	D
6	0110	6		1110	E
7	0111	7		1111	F

1.2 基本逻辑事件的表示方法

1. 非

图1.3是一个简单的非逻辑电路。分析电路可以知道，只有开关 A 断开的时候，灯泡 F 才亮。它们之间的关系，可以用图1.3(b)所示的状态图来表示。开关 A 对应于断开和闭合两种状态，灯泡 F 对应于亮和灭两种可能，这两种对立的逻辑状态我们可以用"0"和"1"来表示，但是它们并不代表数量的大小，只是表示了两种对立的可能。假设开关断开和灯泡不亮用"0"表示，开关闭合和灯泡亮用"1"表示，又可以得到图1.3(c)，该图称为真值表。从真值表可以看出，逻辑非的含义为：当条件不具备时，事件才发生。



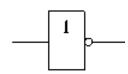
(a)

A	F
断开	亮
闭合	灭

(b)

A	F
0	1
1	0

(c)



(d)

图1.3 非逻辑电路、真值表和符号

在逻辑电路中，把能实现非运算的基本单元叫做非门，其逻辑符号如图1.3(d)所示。

对逻辑变量 A 进行逻辑非运算的表达式为：

$$F = \overline{A}$$

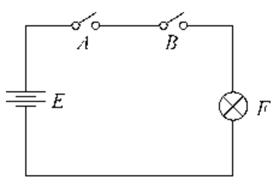
其中 $\overline{A}$ 读做 A 非或 A 反。注意在这个表达式中，变量(A、F)的含义与普通代数有本质的区别：无论输入量(A)还是输出量(F)都只有两种取值0、1，没有任何第三种值。

2. 与、与非

图1.4(a)是两个开关 A、B 和灯泡 F 及电源组成的串联电路，这是一个简单的与逻辑电路。分析电路可知，只有当开关 A 和 B 都闭合时，灯泡 F 才会亮；A 和 B 只要有一个断开或者全都断开，则灯泡灭，它们之间的关系可以用图1.4(b)表示。其真值表如图1.4(c)所示。与的含义是：只有当决定一事件的所有条件都全部具备时，这个事件才会发生。逻辑与也叫逻辑乘。

在逻辑电路中，把能实现与运算的基本单元叫做与门，其逻辑符号如图1.4(d)所示。

逻辑函数 F 与逻辑变量 A、B 的与运算表达式为：



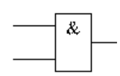
(a)

A	B	F
断开	断开	灭
断开	闭合	灭
闭合	断开	灭
闭合	闭合	亮

(b)

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

(c)



(d)

图1.4 与逻辑电路、真值表和符号

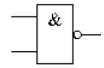
$$F = A \cdot B$$

式中“$\cdot$”为逻辑与运算符，也可以省略。

表达式 $F=\overline{AB}$ 称作逻辑变量 A、B 的与非，其真值表和逻辑符号如图 1.5 所示。

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

(a)



(b)

图 1.5　$F=\overline{AB}$的真值表和逻辑符号

3. 或、或非

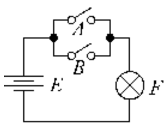
图 1.6 就是一个简单的或逻辑电路。若逻辑变量 A、B、F 和前述的定义相同，通过分析电路显然可知：A、B 中只要有一个为 1，则 $F=1$，即 $A=1$、$B=0$，$A=0$、$B=1$ 或 $A=1$、$B=1$ 时都有 $F=1$；只有 A、B 全为 0 时，F 才为 0。其真值表如图 1.6(b)所示。因此，或的含义是：在决定一事件的各条件中，只要有一个或一个以上的条件具备时，这个事件就发生。逻辑或也叫逻辑加。

在逻辑电路中，把能实现或运算的基本单元叫做或门，其逻辑符号如图 1.6(c)所示。逻辑函数 F 与逻辑变量 A、B 的或运算表达式为：

$$F = A + B$$

式中“+”为逻辑或运算符。

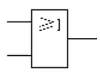
表达式 $F=\overline{A+B}$ 称作逻辑变量 A、B 的或非，其真值表和逻辑符号如图 1.7 所示。



(a)

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

(b)

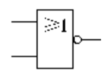


(c)

图 1.6　或逻辑电路、真值表和逻辑符号

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

(a)



(b)

图 1.7　$F=\overline{A+B}$的真值表和逻辑符号

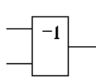
4. 同或和异或

实训中所遇到的逻辑关系，称为异或。逻辑表达式 $F=\overline{A}B+A\overline{B}$ 表示 A 和 B 的异或运算，其真值表和逻辑符号如图 1.8 所示，这个真值表和实训中的表 1.2 是完全相同的。从真值表可以看出，异或运算的含义是：当输入变量相同时，输出为 0；当输入变量不同时，输出为 1。$F=\overline{A}B+A\overline{B}$ 又可表示为 $F=A\oplus B$，符号“$\oplus$”读做异或。

逻辑表达式 $F=\overline{A}\overline{B}+AB$ 表示 A 和 B 的同或运算，其真值表和逻辑符号如图 1.9 所示。从真值表可以看出，同或运算的含义是：当输入变量相同时，输出为 1；当输入变量不同时，输出为 0。$F=\overline{A}\overline{B}+AB$ 又可表示为 $F=A\odot B$，符号“$\odot$”读做同或。

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

(a)

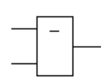


(b)

图 1.8　$F=\overline{A}B+A\overline{B}$ 的真值表和逻辑符号

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

(a)



(b)

图 1.9　$F=\overline{A}\overline{B}+AB$ 的真值表和逻辑符号

通过图 1.8 和图 1.9 的真值表也可以看出,异或和同或互为非运算,即

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

上面我们讨论了几种基本的逻辑运算,这些基本的逻辑关系也可以推广到多变量的情况,例如:

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

$$F = A + B + C \cdots$$

实际的逻辑问题往往非常复杂,但是它们可以通过基本逻辑关系的组合来实现,如:

$$F = \overline{A \cdot B \cdot C} \qquad \text{与非运算}$$

$$F = \overline{A + B + C} \qquad \text{或非运算}$$

$$F = \overline{AB + CD} \qquad \text{与或非运算}$$

$$F = \overline{A(B + C)} + DEF \qquad \text{复杂运算}$$

在复合逻辑运算中要特别注意运算的优先顺序,该优先顺序为:(a)圆括号;(b)非运算;(c)与运算;(d)或运算。

1.3 逻辑变量与逻辑函数

分析研究各种逻辑事件、逻辑电路,就必须借助逻辑代数这一数学工具。在逻辑代数中的变量称为逻辑变量,用字母 $A,B,C,\cdots$ 表示,例如前述的照明灯控制开关 A、B 等。逻辑变量只有两种取值:真和假,一般“1”表示真,“0”表示假。表达式 $F=AB$ 等称为逻辑函数。掌握逻辑函数的运算是研究数字电路的基础。

1.3.1 逻辑代数的基本运算

熟悉和掌握逻辑函数的运算法则,将为分析和设计数字电路提供很多方便。逻辑函数的运算法则包括公理、基本定律、基本规则和一些公式。

1. 公理和基本定律

逻辑代数的公理有:

(1) $\overline{1}=0$;$\overline{0}=1$

(2) $1 \cdot 1=1$;$0+0=0$

(3) $1 \cdot 0=0 \cdot 1=0$;$1+0=0+1=1$

(4) $0 \cdot 0=0$;$1+1=1$

(5) 如果 $A\neq 0$,则 $A=1$;如果 $A\neq 1$,则 $A=0$。

这些公理符合逻辑推理,其合理性不言而喻。

逻辑代数的基本定理有:

(1) 交换律:$A \cdot B=B \cdot A$;$A+B=B+A$

(2) 结合律:$A(BC)=(AB)C$;$A+(B+C)=(A+B)+C$

(3) 分配律:$A(B+C)=AB+AC$;$A+BC=(A+B)(A+C)$

(4) 01 律:$1 \cdot A=A$;$0+A=A$

$0 \cdot A=0$;$1+A=1$

(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) 反演律(德・摩根定律)：$\overline{A \cdot B}=\overline{A}+\overline{B}$；$\overline{A+B}=\overline{A} \cdot \overline{B}$

(8) 还原律：$\overline{\overline{A}}=A$

如果两个逻辑函数具有相同的真值表，则这两个逻辑函数相等。因此，证明以上定律的基本方法是用真值表法，即分别列出等式两边逻辑表达式的真值表，若两张真值表完全一致，就说明两个逻辑表达式相等。

【例 1.1】 证明德・摩根定律：$\overline{A \cdot B}=\overline{A}+\overline{B}$。

解　等式两边的真值表如表 1.4 所示：

表 1.4　证明 $\overline{A \cdot B}=\overline{A}+\overline{B}$ 的真值表

A	B	$\overline{A \cdot B}$	$\overline{A}+\overline{B}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

从表 1.4 可以看出，$\overline{A \cdot B}$ 与 $\overline{A}+\overline{B}$ 在变量 A、B 的四种取值组合下结果完全一样，因此等式成立。

2. 逻辑代数的三个基本规则

(1) 代入规则

在任何一个含有变量 A 的逻辑代数等式中，如果将所有出现 A 的地方都代之以一个逻辑函数，则等式仍然成立，这个规则称为代入规则。

例如，在等式 $B(A+C)=BA+BC$ 中，将所有出现 A 的位置都代以函数 $(A+D)$，则：

等式左边为　$B[(A+D)+C]=B(A+D)+BC=BA+BD+BC$

等式右边为　$B(A+D)+BC=BA+BD+BC$

显然，等式仍然成立。

(2) 反演规则

已知逻辑函数 F，欲求其反函数时，可以将 F 中所有的与“•”换成或“+”，所有的或“+”换成与“•”；“0”换成“1”，“1”换成“0”；原变量换成反变量，反变量换成原变量。经过这种变换后所得到的逻辑函数表达式就是反函数 $\overline{F}$。这个规则称为反演规则。

利用反演规则，可以比较容易地求出一个函数的反函数。但变换时要注意两点，一点是要保持原式中逻辑运算的优先顺序；另一点是，不是一个变量上的反号应保持不变，否则就要出错。

例如：$F=\overline{A}\overline{B}+CD$，则反函数为 $\overline{F}=(A+B) \cdot (\overline{C}+\overline{D})$，而不是 $\overline{F}=A+B \cdot \overline{C}+\overline{D}$。

又例如：$F=A+\overline{B+\overline{C}+\overline{D+\overline{E}}}$，则反函数为 $\overline{F}=\overline{A} \cdot \overline{\overline{B} \cdot C \cdot \overline{\overline{D} \cdot E}}$。

(3) 对偶规则

对于一个逻辑表达式 F，如果将 F 中的与“•”换成或“+”，或“+”换成与“•”；“1”换

成“0”,“0”换成“1”,那么就得到一个新的逻辑表达式,这个新的表达式称为 F 的对偶式 F'。变换时要注意变量保持不变、原表达式中的优先顺序保持不变。

例如:$F=A\cdot(B+C)$　　　则对偶式 $F'=A+B\cdot C$

$F=(A+0)\cdot(B\cdot 1)$　　则对偶式 $F'=A\cdot 1+(B+0)$

所谓对偶规则,是指当某个恒等式成立时,则其对偶式也成立。

如果两个逻辑表达式相等:$F=G$,那么它们的对偶式也相等:$F'=G'$。

3. 常用公式

利用上面的公理、定律、规则可以得到一些常用的公式,掌握这些常用公式,对逻辑函数的化简很有帮助。

(1) 吸收律

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

(2) 还原律

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

(3) 冗余律

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

证明
$$\begin{aligned} AB+\overline{A}C+BC &= AB+\overline{A}C+BC(A+\overline{A}) \\ &= AB+\overline{A}C+ABC+\overline{A}BC \\ &= (AB+ABC)+(\overline{A}C+\overline{A}BC) \\ &= AB+\overline{A}C \end{aligned}$$

推论　$AB+\overline{A}C+BCDE=AB+\overline{A}C$

上述其他公式的证明,读者可自行解决。

1.3.2 逻辑函数的表示方法

逻辑函数的表示方法主要有逻辑函数表达式、真值表、卡诺图、逻辑图等。

1. 逻辑函数表达式

用与、或、非等逻辑运算表示逻辑变量之间关系的代数式,叫逻辑函数表达式。例如,$F=A+B$,$G=A\cdot B+C+D$ 等。

2. 真值表

在前面的论述中,已经多次用到真值表。描述逻辑函数各个变量的取值组合和逻辑函数取值之间对应关系的表格,叫真值表。每一个输入变量有 0,1 两个取值,n 个变量就有 2^n 个不同的取值组合,如果将输入变量的全部取值组合和对应的输出函数值一一对应地列举出来,即可得到真值表。表 1.5 分别列出两个变量与、或、与非及异或运算的真值表。下面举例说明列真值表的方法。

【例 1.2】 列出函数 $F=\overline{AB}$ 的真值表。

解　该函数有两个输入变量,共有 4 种输入取值组合,分别将它们代入函数表达式,并进行求解,得到相应的输出函数值。将输入、输出值一一对应列出,即可得到表 1.6 所示的真值表。