数字电子技术基础

欧阳天鹤　王梅香
赵鸣谦　陈　晰　编著

科学出版社
1989

内容简介

本书是根据高等院校电子技术基础教学大纲(自动化类)的要求，按60学时授课时间编写的. 内容包括逻辑代数及逻辑函数简化，门电路，组合逻辑电路，时序电路，MOS集成电路，集成脉冲单元电路和数字电路应用中的若干实际问题.

本书在处理电子器件与电路的关系时，把重点放在基本电路的分析和应用上；在处理基本单元逻辑电路和中大规模集成电路关系时，则力求把基本逻辑单元电路讲透，同时举例加以分析，为进一步分析中大规模集成电路作好准备.

本书在文字叙述上力求通俗易懂，讲清物理概念，为教学和自学提供方便.

本书可作为工科院校自动控制专业及其他有关专业的"数字电子技术"课程的教材，也可作为有关工程技术人员自学"数字电子技术"的参考书.

数字电子技术基础

欧阳天鹤 王梅香 赵鸣谦 陈晰 编著

责任编辑 张建荣

科学出版社出版

北京东黄城根北街16号

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1989年11月第一版 开本：787×1092 1/16

1989年11月第一次印刷 印张：14

印数：0001—4,380 字数：317,000

ISBN 7-03-001292-5/TN·67

定价：6.20元

目　　录

第一章　逻辑代数基础及逻辑函数的简化

本章主要讨论逻辑代数的初步知识和逻辑函数的简化方法，逻辑代数(又称布尔代数)是逻辑设计的数学基础，是数字电路的构成依据。应用它可以完成对数字逻辑系统的分析和综合。

所谓“逻辑系统的分析”就是对已知线路(电路图或逻辑图)逻辑功能的分析。先写出若干逻辑函数式，然后再进行形式上的简化或变换，最后得到的逻辑函数式便能准确地表示出本系统的全部的逻辑功能。

所谓“逻辑系统的综合”就是建立描述逻辑功能的逻辑函数式，根据选用的器件，对逻辑函数进行简化和变化，设计出满足要求的逻辑线路。

这两方面的应用都要涉及到本章内容，其工作过程如图 1-1 所示。

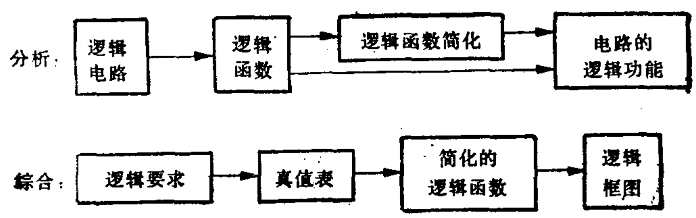

图 1-1　逻辑问题分类

1.1　计数制简介

在日常生活中人们十分习惯于使用十进制的计数制，可是在数字电路中，由于受到电子元件和线路限制，以及受到使用要求的限制，常常需要采用不同的计数体制进行计数。为此，首先介绍一下数字电路中常用的几种计数制。

1.1.1　几种常用的计数体制

1. 十进计数制

十进计数制是大家十分习惯的一种计数方法。它的特点是用十个数码 0，1，2，…，9 的不同组合来表示某一个数的，即十进计数体制中的任何一个数都可以用这十个数码按要求排列起来，其计数规律为“逢十进一”(或借一当十)。任何一个十进制数都可以用其幂的形式来表示。

例如一个十进制数 364，可以写成 $(364)_{10}$ 或用 $(364)_D$ 表示(右下角上的 10 或 D 是表示这个数制为十进制数，以示区别不同的计数制)。这三个数码在数中的位置不同而存在不同的含义。3 在百位，表示 300；6 在十位，表示 60；4 在个位，表示 4。所以 $(364)_{10}$ 又可写为：

$$(364)_{10} = 3 \times 10^2 + 6 \times 10^1 + 4 \times 10^0$$

其中的 10 称为底数(或为基数)。

又如 $(5555)_{10}$ 可写为:

$$(5555)_{10}=5\times10^3+5\times10^2+5\times10^1+5\times10^0$$

式中的数码 5 称为系数,式中的 $10^3, 10^2, 10^1, 10^0$ 称为"位权"。

由此可见,同一个数码 5 由于所处的位置 $(10^3, 10^2, 10^1, 10^0)$ 不同,它所表示的数值也不同。其数值为系数和位权的乘积。例如 10^3 上的 5,其值为 $5\times10^3=5000$;其余的均可类推。

任意一个十进制整数,可用下列通式表示:

$$\begin{aligned}(N)_{10}&=(a_{n-1}a_{n-2}\cdots a_1a_0)_{10}\\&=(a_{n-1}\times10^{n-1}+a_{n-2}\times10^{n-2}+\cdots+a_1\\&\quad\times10^1+a_0\times10^0)_{10}\\&=\sum_{i=0}^{n-1}a_i\times10^i\end{aligned}$$

式中 a_i 为系数,其值可以为 $0, 1\cdots, 9$, n 代表十进制数的位数。

带小数的十进制数亦可按上述方法进行讨论,这里不再赘述。

2. 二进计数制

由于二进计数制具有运算简单,在数字电路中易于实现的优点,所以在数字系统中是一种最常用的、也是非常重要的一种计数制。和十进制相比,二进制的特点是用两个数码 0 和 1 的不同组合来表示一个二进制数。其计数规律为"逢二进一"(或借一当二)。任何一个二进制数都可以用其幂的形式表示。

例如一个二进制数 1101 可写成 $(1101)_2$ 或用 $(1101)_B$ 表示(右下角上的 2 或 B 是表示这个数为二进制数,以区别不同的计数制)。这个二进制数可用其幂的形式表示:

$$(1101)_2=1\times2^3+1\times2^2+0\times2^1+1\times2^0$$

其中的 2 称为二进制的底数(或基数)。

任意一个二进制整数 N 可展开成以下形式:

$$\begin{aligned}(N)_2&=(a_{n-1}a_{n-2}\cdots a_1a_0)_2\\&=a_{n-1}\times2^{n-1}+a_{n-2}\times2^{n-2}+\cdots+a_1\times2^1+a_0\times2^0\\&=\sum_{i=0}^{n-1}a_i\times2^i\end{aligned}$$

式中, a_i 为系数,其值为 0 或 1;

n 为二进制数的位数;

2^i 为二进制数各位的"位权"。

用四位二进制数按其计数规律,对照十进制数,可有以下对应关系。见表 1-1。

由此可见,这里四位二进制数码的权分别为 8,4, 2, 1。其相应的码称为 8421码。

例如,8421 码的 1001 对应的十进制数为 9,这两个不同数制的对应关系可由下式得到。

$$\begin{aligned}(1001)_2&=1\times2^3+0\times2^2+0\times2^1+1\times2^0\\&=(9)_{10}\end{aligned}$$

表 1-1　二-十进制对照表

二进制数	十进制数	二进制数	十进制数
0 0 0 0	0	1 0 0 0	8
0 0 0 1	1	1 0 0 1	9
0 0 1 0	2	1 0 1 0	10
0 0 1 1	3	1 0 1 1	11
0 1 0 0	4	1 1 0 0	12
0 1 0 1	5	1 1 0 1	13
0 1 1 0	6	1 1 1 0	14
0 1 1 1	7	1 1 1 1	15

利用这个关系式可以将任意一个二进制数转换为十进制数。

除了十进制和二进制数以外，在数字电路和计算机中还经常要用到八进制数和十六进制数，其计数特点读者可以自行总结得到。

总之，任意一个数字，都可以用不同的计数制去表示，尽管表现形式上可以各不相同，但数值的大小是不变的。这四种常用的计数制的特点可从表 1-2 中进一步去认识和理解。

表 1-2　十进制、二进制、八进制和十六进制之间的对照关系

十进制数	二进制数	八进制数	十六进制数
0 0	0 0 0 0	0 0	0
0 1	0 0 0 1	0 1	1
0 2	0 0 1 0	0 2	2
0 3	0 0 1 1	0 3	3
0 4	0 1 0 0	0 4	4
0 5	0 1 0 1	0 5	5
0 6	0 1 1 0	0 6	6
0 7	0 1 1 1	0 7	7
0 8	1 0 0 0	1 0	8
0 9	1 0 0 1	1 1	9
1 0	1 0 1 0	1 2	*A*
1 1	1 0 1 1	1 3	*B*
1 2	1 1 0 0	1 4	*C*
1 3	1 1 0 1	1 5	*D*
1 4	1 1 1 0	1 6	*E*
1 5	1 1 1 1	1 7	*F*

1.1.2　不同计数制之间的相互转换

既然相同数值的一个数可以用不同进制的计数方法去表示，它们之间就必然存在着一定的关系，因此，它们之间可以进行转换。

1. 任意进制数转换成十进制数

十进制数是人们最熟悉的计数方式，但为了计算和认读方便，有时需要用不同进制的

数来表示一个数，这样就需将其转换为十进制数．常用方法是将不同进制的数，按其数制幂的形式展开计算即可得到．

例 1.1.1

$$(372)_8 = 3 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 = (250)_{10}$$

例 1.1.2

$$(11010)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (26)_{10}$$

2. 十进制数转换成任意进制数

在数字电路中，有时必须将十进制转换成其它进制的数进行应用．实际上这正是上述过程的逆过程． 现以十进制数转换成二进制数为例来说明这个过程． 设十进制数为 $(9)_{10}$，将它转换成二进制数，可写成：

$$(9)_{10} = a_3 \times 2^3 + a_2 \times 2^1 + a_1 \times 2^1 + a_0 \times 2^0 \tag{1-1}$$

只要能求出 a_0, a_1, a_2, a_3，就可以实现转换．

为了求得 a_0, a_1, a_2, a_3，先来看式 (1-1) 右边． 因为 a_0, a_1, a_2 和 a_3 的值只可能为 0 或 1，并且 $2^0 = 1$，因此将式 (1-1) 两边连续地除以底数 2 时，等式右边必然依次得到余数 a_0, a_1, a_2，当商为 0 时，余数为 a_3． 同时等式左端相当有余数 1001． 等式两端同除以一个数，所得余数相等．因此可得 $a_0 = 1$, $a_1 = 0$, $a_2 = 0$, $a_3 = 1$． 其过程见下列短除法计算式．

$$\begin{array}{r|l l}
 & & \text{余数} \\
2 & 9 & \cdots 1 \\ \hline
2 & 4 & \cdots 0 \\ \hline
2 & 2 & \cdots 0 \uparrow \\ \hline
2 & 1 & \cdots 1 \\ \hline
 & 0 &
\end{array}
\quad \Longrightarrow \quad
\begin{array}{r|l r}
 & & \text{余数} \\
2 & a_3 \times 2^3 + a_2 \times 2^2 + a_1 \times 2^1 + a_0 \times 2^0 & \cdots a_0 \\ \hline
2 & a_3 \times 2^2 + a_2 \times 2^1 + a_1 \times 2^0 & \cdots\cdots a_1 \\ \hline
2 & a_3 \times 2^1 + a_2 \times 2^0 & \cdots\cdots a_2 \uparrow \\ \hline
2 & a_3 \times 2^0 & \cdots\cdots a_3 \\ \hline
 & 0 &
\end{array}$$

同理，可将一个十进制数用 8 或 16 连续去除，即可将它转换成八进制或十六进制数．

注意：应用上述方法时，一定要除到商等于 0 为止，所得余数应从下(高位)向上(低位)排列，不可随便颠倒．

例 1.1.3 将 $(25)_{10}$ 转换为二进制数．

解：

$$\begin{array}{r|r l l}
 & & \text{余数} & \\
2 & 25 & \cdots\cdots 1 & \uparrow a_0 \\ \hline
2 & 12 & \cdots\cdots 0 & \\ \hline
2 & 6 & \cdots\cdots 0 & \\ \hline
2 & 3 & \cdots\cdots 1 & \\ \hline
2 & 1 & \cdots\cdots 1 & a_4 \\ \hline
 & 0 & &
\end{array}$$

故

$$(25)_{10} = (11001)_2$$

例 1.1.4 将 $(443)_{10}$ 转换成八进制数。

解：

$$
\begin{array}{r|l l}
 & & \text{余数} \\
8 & 443 & \cdots\cdots\cdots\cdots\cdots\cdots 3 \\
8 & 55 & \cdots\cdots\cdots\cdots\cdots\cdots 7 \\
8 & 6 & \cdots\cdots\cdots\cdots\cdots\cdots 6 \\
 & 0 &
\end{array}
$$

故

$$(443)_{10} = (673)_8$$

例 1.1.5 将 $(3901)_{10}$ 转换成十六进制数。

解：

$$
\begin{array}{r|l l}
 & & \text{余数} \\
16 & 3901 & \cdots\cdots\cdots\cdots\cdots\cdots B \\
16 & 243 & \cdots\cdots\cdots\cdots\cdots\cdots 3 \\
16 & 15 & \cdots\cdots\cdots\cdots\cdots\cdots F \\
 & 0 &
\end{array}
$$

故

$$(3901)_{10} = (F3B)_{16}$$

1.2 逻 辑 代 数

任何数字系统，它的电路往往是十分复杂的，可仔细地分析一下这些电路却都是由几种基本的简单的逻辑门构成。因此，必须从研究这些简单逻辑门的逻辑特性开始，写出逻辑代数式，并从中找出它们的运算规律。

1.2.1 三种基本逻辑关系

逻辑问题是研究一事物的原因（或条件）与结果之间的关系。条件与结果均包含互相对立的两个方面。如某一事件的真和假，是和非，开关的通与断，电路中电流的有和无，电压的高和低等等。它们都反映了事物的对立的两个方面。

最基本的逻辑关系有三种："与"逻辑，"或"逻辑和"非"逻辑。 下面分别讨论上述的三种逻辑。

1. "与"逻辑

与某事件有关的全部条件具备时，这事件才会发生.这种因果关系称为"与"逻辑关系.

例如图 1-2 所示的开关电路，只有当开关（条件）A_1 与 B_1 同时都闭合时，电路中才会有电流（结果）I_1 流过。因此，开关 A_1 与 B_1 的闭合和有电流 I_1 之间为"与"逻辑关系。

2. "或"逻辑

与某事件有关的条件中，只要具备一个以上的条件，这事件就会发生。这种因果关系

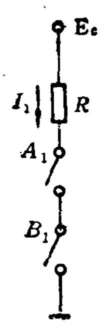

图 1-2 “与”逻辑图

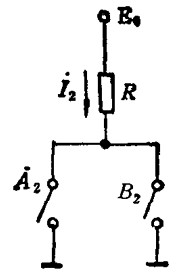

图 1-3 “或”逻辑电路图

称为“或”逻辑关系。

例如图 1-3 所示的开关电路，只要开关 A_2 或 B_2 中有一个闭合，电路中就会有电流 I_2，因此，开关 A_2 或 B_2 的闭合和有电流 I_2 之间为“或”逻辑关系。

3.“非”逻辑

某事件的条件与结果是相反的关系。这种因果关系称为“非”逻辑。

例如图 1-4 所示的开关电路，当开关 A_3 闭合时，电路中有电流 I_3，这时，电压输出 $V=0$V；反之，当开关 A_3 打开时，电路中无电流 I_3，输出电压 $V=E_c$V。若把电流 I_3 作为条件，把电压 V 作为结果，则可知：I_3 与 V 刚好互为相反状态，因此，电流 I_3 与电压 V 之间是“非”逻辑关系。

任何复杂的逻辑关系都是由这三种逻辑关系组合而成。具有应用价值的“与”、“或”、“非”电路将在第二章中作详细讨论。

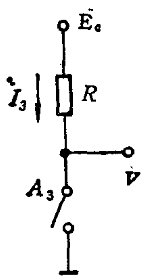

图 1-4 “非”逻辑电路图

1.2.2 逻辑关系的代数表示法

前面用文字描述了三种基本逻辑关系，这种描述在使用中很不方便，如果能用逻辑代数式将它们表现出来，它将会给逻辑电路的分析和设计带来极大的方便。

为了用逻辑代数式描述逻辑电路的逻辑关系，和普通代数一样，用字母 $A, B, I, L, \cdots$ 表示逻辑变量。由于逻辑变量只表示事物对立的逻辑状态，即开关的通与断，电流的有与无，电压的高与低等等。因此，逻辑变量只能取两个逻辑值，常取数码 1 和 0 表示。它不代表电流和电压的具体数值，这和普通代数的变量取值是不相同的。在一个具体的逻辑电路中，若令逻辑变量 $A, B, I, L, \cdots$ 取值为 1，代表开关接通，有电流状态或高电平；则 $A, B, I, L, \cdots$ 取值为 0，就表示开关的断开、无电流状态或低电平。在逻辑代数式中，还用 $\overline{A}, \overline{B}, \overline{I}, \overline{L}, \cdots$ 分别代表与 $A, B, I, L, \cdots$ 取值相反的变量。如 $A=0$ 时，$\overline{A}=1$；$A=1$ 时，$\overline{A}=0$。变量上的一横“—”叫做非号，“$\overline{A}$”读作“A 非”。有时称 $A, B, I, L, \cdots$ 为原变量，则 $\overline{A}, \overline{B}, \overline{I}, \overline{L}, \cdots$ 称为反变量。

在上述规定下，以图 1-2，1-3，1-4 为例，设各变量取值为“1”，代表开关接通，有电流或高电平，这时可用式 (1-2)，(1-3)，(1-4) 分别描述各电路的逻辑关系。

"与"逻辑 $$I_1 = A_1 \cdot B_1 \tag{1-2}$$

"或"逻辑 $$I_2 = A_2 + B_2 \tag{1-3}$$

"非"逻辑 $$V = \bar{I}_3 = \bar{A}_3 \tag{1-4}$$

显然，上述各逻辑代数式的运算规律是应当完全符合相应电路的工作情况的。所以将相应变量取值为 0 或 1 代入上述三式中，得到下列各等式。

$$0 \cdot 0 = 0 \quad 0 + 0 = 0 \quad \bar{0} = 1$$
$$0 \cdot 1 = 0 \quad 0 + 1 = 1 \quad \bar{1} = 0$$
$$1 \cdot 0 = 0 \quad 1 + 0 = 1$$
$$1 \cdot 1 = 1 \quad 1 + 1 = 1$$

它们的物理意义可用开关的通断情况来说明。它们的正确性也是容易理解的，这里无需赘述。注意： $1 + 1 = 1$，这是逻辑代数式与普通代数式的一个重要区别。

1.2.3 逻辑代数的基本公式和规则

根据三种基本逻辑运算的规律，可以推导出逻辑运算的一些基本公式或定律，熟悉并应用这些公式或定律，使分析设计逻辑电路更为方便。这些公式或定律有些与初等代数相似，有些则完全不同。这些公式或定律可利用代入不同的变量取值，以证明它们是正确的(下面就不再一一证明了)。

1. 变量与常量关系的公式

$$A + 0 = A \qquad A \cdot 1 = A$$
$$A + 1 = 1 \qquad A \cdot 0 = 0$$
$$A + \bar{A} = 1 \qquad A \cdot \bar{A} = 0$$

2. 交换律，结合律，分配律

交换律：

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

结合律：

$$\begin{aligned} A + B + C &= (A + B) + C \\ &= A + (B + C) \end{aligned}$$
$$\begin{aligned} A \cdot B \cdot C &= (A \cdot B) \cdot C \\ &= A \cdot (B \cdot C) \end{aligned}$$

分配律：

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

3. 逻辑代数中的一些特殊规律

重叠律(同一律)：

$$A + A = A$$

$$A \cdot A = A$$

反演律(摩根定律):

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

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

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

4. 逻辑代数中的三个重要规则

下面介绍的三个重要规则,可以作为一种工具更简捷地处理一些问题,也可以更好地根据已知的等式,推导出更多的等式.

(1) 代入规则

在任何一个逻辑等式中,如果将等式两边所出现的某一变量 A 的地方都代之以一个函数 F,则等式仍然成立,这个规则就叫做代入规则.

由于任何一个逻辑函数和逻辑变量一样,只有两种可能取值"0"或"1",所以代入规则是正确的.

代入规则在推导公式中用处很大,因为将已知等式中某一变量用任意一函数代替后,就得到了新的等式,所以扩大了等式的应用范围.

例如在 $A \cdot (B + E) = A \cdot B + A \cdot E$ 式中,将所有出现"E"的地方都代之以 $f = C + D$,则等式仍然成立.

$$\begin{aligned} \text{原式左边} &= A \cdot [B + (C + D)] \\ &= A \cdot B + A \cdot (C + D) \\ &= A \cdot B + A \cdot C + A \cdot D \\ \text{原式右边} &= A \cdot B + A \cdot (C + D) \\ &= A \cdot B + A \cdot C + A \cdot D \end{aligned}$$

(2) 反演规则

对于任意一个函数表达式 L,如果将 L 中所有的"·"换成"+","+"换成"·",所有的原变量换成反变量(如 A 换成 $\overline{A}$),反变量换成原变量(如 $\overline{B}$ 换成 B),所有的常量"0"换为"1","1"换为"0",那么所得到的函数就是 $\overline{L}$,这就是反演规则.

实际上,反演规则不过是反演律的推广,反演规则的实际意义在于:可以比较容易地求出一个函数的反函数.

例如求 $L = \overline{A} \cdot \overline{B} + C \cdot D$ 的反函数. 利用反演规则求得:

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

若运用反演律和其它基本公式也可以求出 $\overline{L}$. 方法如下:

$$\begin{aligned} L &= \overline{A} \cdot \overline{B} + C \cdot D \\ \overline{L} &= \overline{\overline{A} \cdot \overline{B} + C \cdot D} \\ &= \overline{\overline{A} \cdot \overline{B}} \cdot \overline{C \cdot D} \\ &= (\overline{\overline{A}} + \overline{\overline{B}}) \cdot (\overline{C} + \overline{D}) \\ &= (A + B) \cdot (\overline{C} + \overline{D}) \end{aligned}$$

运用两种不同的方法均可得到 $\overline{L}$. 显然,应用反演规则容易一些.

注意：在运用反演规则时，要特别注意运算符号的优先顺序，例如 $L = \overline{A} \cdot \overline{B} + C \cdot D$，按运算符号优先的规定，应先做 $A \cdot B$ 和 $C \cdot D$，然后再进行两者相加的运算. 所以应写成：

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

而不能写成：

$$\overline{L} = A + B \cdot \overline{C} + \overline{D}$$

(3) 对偶规则

对于任何一个逻辑函数 L，如果把 L 中的“+”换为“·”，“·”换为“+”；常量“1”换为“0”，“0”换为“1”，那么就可以得到一个新的表达式，记作 L'，L' 称为 L 的对偶式. 例如

$$L = A \cdot (B + \overline{C})$$
$$L' = A + B \cdot \overline{C}$$

或者

$$L = A + B\overline{C}$$
$$L' = A \cdot (B + \overline{C})$$

由上例可以看出，如果 L 的对偶式是 L'，那么 L' 的对偶式就是 L，也就是说 L 和 L' 是互为对偶的.

一般来说，函数的原表达式和它的对偶式在形式上是不同的，有少数表达式的对偶式在形式上是相同的，表达式的对偶式就是它自身. 例如

$$L = A \quad L' = A$$
$$L = \overline{A} \quad L' = \overline{A}$$

对偶规则在实际运用上是很有价值的. 首先有了对偶规则，使要证明的公式减少一半. 其次，当证明了某两个表达式相等，则它们的对偶式也相等.

例如，已知 $A(B + C + D) = AB + AC + AD$，则等式两边表达式的对偶式亦相等. 即

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

又如，已知 $A + \overline{A}B = A + B$ 成立，则它们的对偶式亦成立. 即

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

反演规则和对偶规则是不同的两个规则，它们的物理意义也是截然不同的. 下面以图 1-5 所示的“与”逻辑为例说明这两个规则.

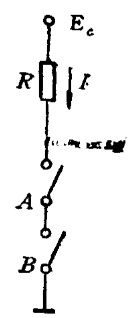

图 1-5 “与”逻辑电路

这个电路可以从两个方面讨论：

1) 当开关 A 和 B 同时接通时，电路中为有电流状态，$I \neq 0$，从有电流的情况考虑，说明电流与开关 A，B 之间是“与”逻辑关系.

2) 当开关 A 或 B 只要有一个断开时，电路中为无电流状态，$I = 0$，从无电流的情况考虑，说明电流与开关 A，B 之间的关系是“或”逻辑关系.

若规定 A，B，I 的取值为 1 时表示开关接通，即有电流，则电路用“与”逻辑描述.

$$I = A \cdot B$$

在这种规定下，$\bar{A}$，$\bar{B}$ 和 $\bar{I}$ 为 1 值时表示开关断开，即无电流，则电路应当用“或”逻辑描述。

$$\bar{I} = \bar{A} + \bar{B}$$

若又重新规定：A，B 和 I 的取值为 1 时表示开关断开，即无电流，则电路用“或”逻辑式描述。

$$I = A + B$$

在这种新规定下，$\bar{A}$，$\bar{B}$ 和 $\bar{I}$ 为 1 值时表示开关接通，即有电流，则电路应当用“与”逻辑关系描述。

$$\bar{I} = \bar{A} \cdot \bar{B}$$

可见，对于同一个开关电路，由于变量取值的规定不同，可以得到四种不同的逻辑代数式，见表 1-3。

表 1-3　四种不同的逻辑表达式

变量取值为 1 的含义	逻　辑　代　数　式
表示开关接通，有电流	$I = A \cdot B \xleftrightarrow{\text{反演}} \bar{I} = \bar{A} + \bar{B}$ 对$\uparrow$　　$\uparrow$对
表示开关断开，无电流	偶$\downarrow$　　$\downarrow$偶 $I = A + B \xleftrightarrow{\text{反演}} \bar{I} = \bar{A} \cdot \bar{B}$

从表 1-3 中可见：描述同一电路的原式和反演式中变量取值为 1 的含义是相同的。描述同一电路的原式和对偶式中变量取值为 1 的含义是相反的。这一点在以后讨论正负逻辑制时是非常有用的。

1.2.4　逻辑代数中若干常用公式

运用前面讨论过的基本公式和三个规则，可以得到更多的公式。下面介绍一些常用的基本公式，这些公式(包括它们的对偶式)是普通代数中没有的，它们在逻辑函数的代数法化简中经常用到。

公式 1　$A \cdot B + A \cdot \bar{B} = A$

证明：

$$A \cdot B + A \cdot \bar{B} = A(B + \bar{B}) = A$$

说明在两个乘积项中包含 B，$\bar{B}$，而其它因子都相同时，则可采用公式 1，将这两项合并，消去变量 B 和 $\bar{B}$。

公式 2　$A + A \cdot B = A$

证明：

$$A + AB = A(1 + B) = A$$

说明在一个“与-或”表达式中，如果一个乘积项是另一个乘积项的因子，则另一个乘积项是多余的。

公式 3　$A + \bar{A}B = A + B$

证明：

$$
\begin{aligned}
A+\bar{A}B &= A(B+\bar{B})+\bar{A}B \\
&= AB+A\bar{B}+\bar{A}B+AB \\
&= A(B+\bar{B})+B(A+\bar{A}) \\
&= A+B
\end{aligned}
$$

说明在一个“与-或”表达式中,如果一个乘积项 (A) 的反是另一个乘积项 (AB) 中的因子,则这个乘积项的反 ($\bar{A}$) 是多余的.

公式 4 $AB+\bar{A}C+BC=AB+\bar{A}C$

证明:

$$
\begin{aligned}
\text{左式} &= AB+\bar{A}C+BC(A+\bar{A}) \\
&= AB+\bar{A}C+ABC+\bar{A}BC \\
&= AB(1+C)+\bar{A}C(1+B)=AB+\bar{A}C
\end{aligned}
$$

由公式 4 可得以下推论.

推论 $AB+\bar{A}C+BCD=AB+\bar{A}C$

证明:

$$
\begin{aligned}
\text{左式} &= AB+\bar{A}C+BC+BCD \\
&= AB+\bar{A}C+BC(1+D) \\
&= AB+\bar{A}C
\end{aligned}
$$

公式 4 和推论说明在一个“与-或”表达式中，如果两个乘积项里，一个包含了原变量(A),而另一项包含了反变量 ($\bar{A}$)，且这两项的其余因子都是第三个乘积项的因子，则第三个乘积项是多余的.

1.2.5 逻辑函数的代数法化简

例如某一逻辑功能函数为

$$
\begin{aligned}
L(A,B,C,D) &= ABCD+\bar{A}BCD+\bar{A}\bar{B}CD+AB\bar{C}\bar{D} \\
&\quad +\bar{A}B\bar{C}D+\bar{A}\bar{B}\bar{C}D+ABC\bar{D}+\bar{A}BC\bar{D} \\
&\quad +\bar{A}\bar{B}C\bar{D}+AB\bar{C}D+\bar{A}B\bar{C}\bar{D}+\bar{A}\bar{B}\bar{C}\bar{D}
\end{aligned}
$$

要实现这一功能至少要用 12 个与门和 12 个输入端的或门(这种或门实际上没有),才可完成上述功能,可是这个逻辑函数在经过化简之后得到:

$$L(A,B,C,D)=\bar{A}+B$$

即实现上述同样功能只要一个两输入端的或门就行了. 由此可见，要设计好一个数字逻辑电路除了根据功能要求得到电路的逻辑函数外,我们还必须对它进行化简,一般来说逻辑函数式越简单越好,这样设计出来的逻辑线路内部连线少,所需要的元件也少，不仅经济效益高,而且还可以提高电路的可靠性.因此如何简化逻辑函数式,是十分重要的.下面就介绍几种用代数法化简逻辑函数的方法.

1. 并项法

利用 $A+\bar{A}=1$ 的公式将两项合并为一项,合并时消去一个变量. 例如

$$\bar{A}\bar{B}C+\bar{A}\bar{B}\bar{C}=\bar{A}\bar{B}(C+\bar{C})=\bar{A}\bar{B}$$

$$AB\bar{C}+A\overline{B\bar{C}}=A$$

2. 吸收法

利用 $A + AB = A$ 和 $AB + \overline{A}C + BC = AB + \overline{A}C$ 这两个公式消去多余项．例如

$$
\begin{aligned}
&A\overline{B} + A\overline{B}CD(E + F) \\
&\quad = A\overline{B}[1 + CD(E + F)] \\
&\quad = A\overline{B}
\end{aligned}
$$

$$
\begin{aligned}
F &= AB + \overline{A}\overline{C} + B\overline{C} \\
&= AB + \overline{A}\overline{C}
\end{aligned}
$$

3. 消去法

利用 $A + \overline{A}B = A + B$ 消去多余的因子．例如

$$
\begin{aligned}
&AB + \overline{A}C + \overline{B}C \\
&\quad = AB + (\overline{A} + \overline{B})C \\
&\quad = AB + \overline{AB}C \qquad \text{（反演律）} \\
&\quad = AB + C
\end{aligned}
$$

4. 配项法

利用 $A = A(B + \overline{B})$ 将它作为配项用，然后消去更多的项．例如

$$
\begin{aligned}
L &= AB + \overline{A}\overline{C} + B\overline{C} \\
&= AB + \overline{A}\overline{C} + B\overline{C}(A + \overline{A}) \\
&= AB + \overline{A}\overline{C} + AB\overline{C} + \overline{A}B\overline{C} \\
&= AB(1 + \overline{C}) + \overline{A}\overline{C}(1 + B) \\
&= AB + \overline{A}\overline{C}
\end{aligned}
$$

在实际化简逻辑函数过程中，往往不是单独应用上述四种方法中的一种，而更多地是四种方法的综合应用．请看下面的例子．

例 1.2.1 化简 $L = AD + A\overline{D} + AB + \overline{A}C + BD + A\overline{B}EF + \overline{B}EF$

解:

(1) 利用 $A + \overline{A} = 1$ 把 AD 和 $A\overline{D}$ 合并，得:

$$L = A + AB + \overline{A}C + BD + A\overline{B}EF + \overline{B}EF$$

(2) 利用 $A + AB = A$ 把包含 A 因子的乘积项都消去，得:

$$L = A + \overline{A}C + BD + \overline{B}EF$$

(3) 利用 $A + \overline{A}B = A + B$ 可消去 $\overline{A}C$ 中的 $\overline{A}$，得:

$$L = A + C + BD + \overline{B}EF$$

例 1.2.2 化简 $L = AB + A\overline{C} + \overline{B}C + \overline{C}B + \overline{B}D + \overline{D}B + ADE(F + G)$

解:

(1) 据反演律得:

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