固定不变的，属于有权码。

5421BCD 码从高位到低位的权值分别是 5、4、2、1。余 3 码是在 8421 码的基础上，把每个代码都加 0011 码而形成的。它的主要优点是执行十进制数相加时，能正确地产生进位信号，而且还给减法运算带来了方便。

格雷码的特点是，相邻两个代码之间仅有 1 位不同，其余各位均相同。计数电路按格雷码计数时，每次状态更新仅有 1 位代码变化，减少了出错的可能性。

在数字系统中，为了防止代码在传送过程中产生错误，还有其他一些编码方法，如奇偶校验码、汉明码等。国际上还有一些专门处理字母、数字和字符的二进制代码，如 ISO 码、ASCII 码等，读者可参阅有关书籍。

例 1.6　将一个三位十进制数 473 用 8421BCD 码表示。

解　将十进制数 473 每一位用 8421BCD 码表示即可。相应竖式为

$$\begin{array}{ccc} 4 & 7 & 3 \\ \downarrow & \downarrow & \downarrow \\ 0100 & 0111 & 0011 \end{array}$$

所以，$(473)_D=(010001110011)_{8421BCD}$。

例 1.7　将（100001010001）$_{8421BCD}$ 转换成十进制数。

解　相应竖式为

$$\begin{array}{ccc} 1000 & 0101 & 0001 \\ \downarrow & \downarrow & \downarrow \\ 8 & 5 & 1 \end{array}$$

所以，$(100001010001)_{8421BCD}=(851)_D$。

注意：BCD 码与数制的区别，例如：

$$(150)_D=(000101010000)_{8421BCD}=(10010110)_B=(226)_O=(96)_H$$

三、基本逻辑及其逻辑门电路

数字电路是具有逻辑功能的逻辑电路，其逻辑功能体现在输入、输出之间的因果关系。而输入、输出的存在或表现形式，有且仅有两个相互对立的状态，而且它必定是两个状态中的一个。例如：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”两种状态，等等。这两种相互对立的逻辑状态通常用“0”和“1”来表示。

无论多复杂的逻辑电路都是由基本的逻辑关系组成，其基本的逻辑关系有“与"或"非”，也称为三种基本的逻辑运算。

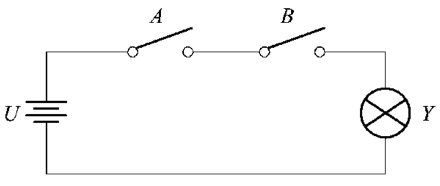
1.“与”逻辑和“与”门电路

当决定某一事件的全部条件都具备时，该事件才会发生，这样的因果关系称为“与”逻辑关系，简称“与”逻辑。在逻辑代数中，对应的运算为“与”运算。

能实现“与”逻辑的电路就称为“与”门电路。如图 1.3（a）所示，电路中只有当开

关 A、B 都闭合时，灯 Y 才亮；只要有一个开关断开，灯就不亮。由图 1.3（b）所示的状态图看出，串联的开关闭合和灯亮就是“与”逻辑关系。

设开关接通为“1”，断开为“0”，灯亮为“1”，灯灭为“0”，可列出图 1.3（c）所示的真值表。



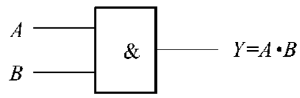
（a）电路图

开关 A	开关 B	灯 Y
不闭合	不闭合	不亮
不闭合	闭合	不亮
闭合	不闭合	不亮
闭合	闭合	亮

（b）状态表

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

（c）逻辑真值表

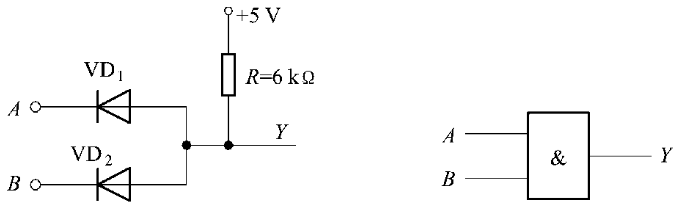


（d）逻辑符号

图 1.3 “与”逻辑和“与”门电路

除了图 1.3（a）满足“与”逻辑外，还有多种电路能实现“与”逻辑。如图 1.4 所示的由二极管构成的电路，在门电路中，我们通常把大于 2 V 的电平规定为高电平，低于 0.8 V 的电平规定为低电平，那么由图 1.4 我们可以分析出：只有当 A、B 两个输入端都为高电平时，输出才为高电平，而 A、B 只要有一个或两个为低电平，输出就为低电平，因此，电路也满足“与”逻辑。我们把凡是满足“与”逻辑的电

路用统一的符号表示，如图 1.4（b）所示。



（a）DTL“与”门电路　　（b）“与”门逻辑符号

图 1.4　由二极管构成的“与”门电路

满足“与”逻辑的函数表达式为：

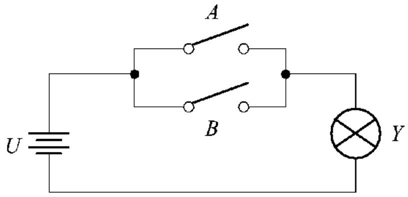
$$Y = A \cdot B$$

2.“或”逻辑和“或”门电路

当决定某一事件的所有条件中，只要有一个具备，该事件就会发生，这样的因果关系叫做“或”逻辑关系。对应的逻辑运算为“或”运算。

能实现“或”逻辑的电路就称为“或”门电路。如图 1.5（a）所示，电路中只要开关 A、B 有一个或两个闭合时，灯 Y 就亮；而当 A、B 都断开时，灯就不亮。由图 1.5（b）所示的状态图看出，并联的开关与灯亮是“或”逻辑关系。

图 1.6 所示为由二极管构成的“或”门电路，只要 A、B 两个输入端中有一个或两个为高电平，输出就为高电平；只有 A、B 两个都为低电平时输出才为低电平，即满足“或”逻辑。凡是满足“或”逻辑的电路，都用如图 1.6（b）所示的符号表示。



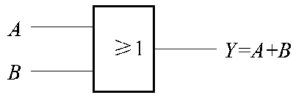
（a）电路图

开关 A	开关 B	灯 Y
不闭合	不闭合	不亮
不闭合	闭合	亮
闭合	不闭合	亮
闭合	闭合	亮

（b）状态表

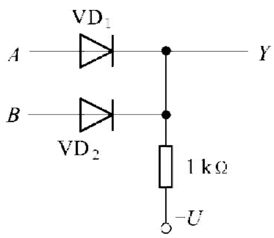
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

（c）逻辑真值表

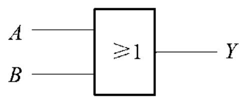


（d）逻辑符号

图 1.5 “或”逻辑和“或”门电路



（a）DTL“或”门电路



（b）“或”门逻辑符号

图 1.6　由二极管构成的“或”门电路

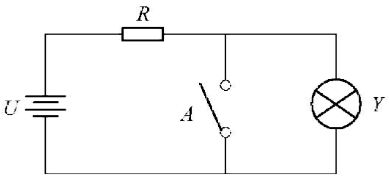
满足“或”逻辑的函数表达式为：

$$Y = A + B$$

3.“非”逻辑和“非”门电路

当某一条件具备了，事情不会发生；而此条件不具备时，事情反而发生。这种逻辑关系称为“非”逻辑关系。对应的运算为求“反”运算或“非”运算。

满足“非”逻辑的电路称为“非”门电路，如图 1.7 和图 1.8 所示。



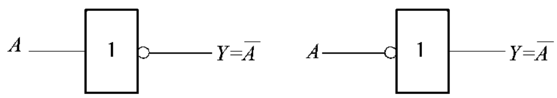
（a）电路图

开关 A	灯 Y
不闭合	亮
闭合	不亮

（b）状态表

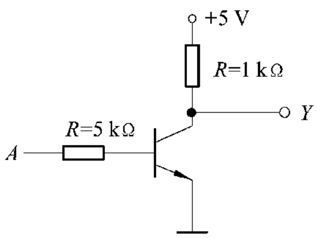
A	Y
0	1
1	0

（c）逻辑真值表

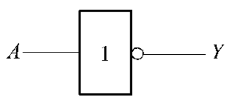


（d）逻辑符号

图 1.7 “非”逻辑和“非”门电路



（a）DTL“非”门电路



（b）“非”门逻辑符号

图 1.8 由三极管构成的“非”门电路

满足“非”逻辑的函数表达式为：

$$Y = \overline{A}$$

四、其他常用的组合逻辑门电路

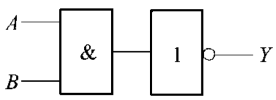
1.“与非”逻辑及“与非”门电路

由“与”门电路和“非”门电路组合在一起，就构成“与非”门电路。

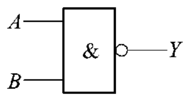
图 1.9 所示为由“与”门和“非”门构成的“与非”门电路及其逻辑符号、真值表。

“与非”逻辑的函数表达式为：

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



（a）逻辑电路



（b）逻辑符号

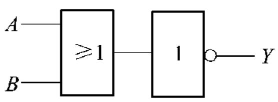
A	*B*	*Y*
0	0	1
0	1	1
1	0	1
1	1	0

（c）真值表

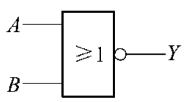
图 1.9 “与非”逻辑及“与非”门电路

2.“或非”逻辑及“或非”门电路

由“或”门电路和“非”门电路组合在一起，就构成“或非”门电路。图 1.10 所示为由“或”门和“非”门构成的“或非”门电路及其逻辑符号、真值表。



（a）逻辑电路



（b）逻辑符号

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

（c）真值表

图 1.10 “或非”逻辑及“或非”门电路

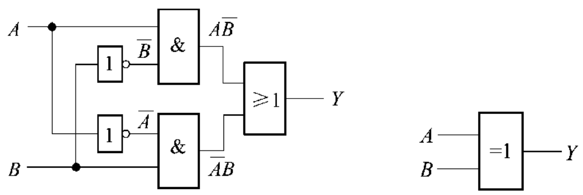
“或非”逻辑的函数表达式为：

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

3.“异或”逻辑及“异或”门电路

当两个条件中只要有任一条件且只有一个条件满足时，事情就会发生，这种逻辑关系就是“异或”逻辑。“异或”是一种二变量逻辑运算，当两个变量取值相同时，逻辑函数值为 0；当两个变量取值不同时，逻辑函数值为 1。

由“非”门、“与”门和“或”门组合的“异或”门电路及其逻辑符号、真值表如图 1.11 所示。



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

（a）逻辑电路　（b）逻辑符号　（c）真值表

图 1.11 “异或”逻辑及“异或”门电路

“异或”逻辑的函数表达式为：

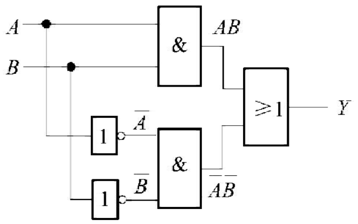
$$Y=A\overline{B}+\overline{A}B=A\oplus B$$

4.“同或”逻辑及“同或”门电路

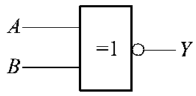
“同或”与“异或”的结果相反，当两个条件同时满足或同时不满足时，事情就会发生。“同或”是一种二变量逻辑运算，当两个变量取值相同时，逻辑函数值为“1”；当两个变量取值不同时，逻辑函数值为“0”。

由“非”门、“与”门和“或”门组合而成的“同或”门及其逻辑符号、真值表如图

1.12 所示。



（a）逻辑电路



（b）逻辑符号

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

（c）真值表

图 1.12 “同或”逻辑及“同或”门电路

“同或”逻辑的函数表达式为：

$$Y=\overline{A}\,\overline{B}+AB=A\otimes B=\overline{A\oplus B}$$

“同或”逻辑一般用“异或”逻辑门来表示，因为现在厂家很少专门生产“同或”门。

五、逻辑函数的表示

在逻辑电路中，如果输入变量 A、B、C、…的取值确定后，输出变量 Y 的值也唯一确定，Y 是 A、B、C、…的逻辑函数。逻辑函数的一般表达式可以写为：

$$Y=F(A,B,C,\dots)$$

逻辑函数 Y 和逻辑变量 A、B、C、…都只有逻辑“0”或逻辑“1”两种取值。

逻辑函数的表示方式主要有：真值表、函数表达式、卡诺图、逻辑图、波形图等。

1. 真值表

把变量的各种取值组合与对应的函数值以表格形式一一列举出来，这种表格就叫真值表。每个输入变量有“0”和“1”两个取值，n 个变量就有 2^n 个不同取值组合，将它们按顺序（一般按二进制数递增规律）排列起来，同时在相应位置写上函数值，即可得到逻辑函数的真值表。如前面的图 1.3（c）、图 1.5（c）、图 1.7（c）、图 1.9（c）、图 1.10（c）、图 1.11（c）、图 1.12（c）所示就为“与”门、“或”门、“非”门、“与非”门、“或非”门、“异或”门、“同或”门的真值表。

真值表的特点是直观明了地反映逻辑事件的逻辑功能关系，并且通过真值表能将实际逻辑问题抽象成为数学表达式形式，所以在数字电路设计过程中，第一步就是列出真值表；在分析数字电路逻辑功能时，最后也是列出真值表。

例如，在本学习项目的火灾报警控制电路的设计中，我们将感烟、感温和感红外线三种类型的火灾探测器作为控制电路的输入信号，并进行逻辑赋值，设三种探测器探测到烟（A）、温（B）、红外线（C）信号时赋值为 1，否则为 0，而把报警信号作为输出信号，并设有报警信号（Y）赋值为 1，否则为 0。则根据逻辑要求（为了防止误报警，只有当其中有两种或两种以上类型的探测器发出火灾检测信号时，报警系统产生报警控制信号）就可得到此逻辑事件的真值表，如表 1.2 所示。

表 1.2　火灾报警控制电路的真值表

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

2. 卡诺图

卡诺图是图形化的真值表。如果把各种输入变量取值组合下的输出函数值填入一种特殊的方格图中，即可得到逻辑函数的卡诺图。对卡诺图的介绍读者可参见本学习项目中的"逻辑函数的化简"部分。

3. 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，叫做逻辑函数表达式，例如，$Y = A + B$，$Y = A\overline{B} + \overline{A}B = A \oplus B$ 等。

逻辑函数表达式的优点是书写方便，便于用逻辑代数中的公式和定理变换、运算，有利于化简和作图；缺点是在逻辑函数比较复杂时，难以直接从输入变量取值看出函数值。

由真值表可以写出逻辑函数的表达式。在真值表中，变量取值为 1 的，以原变量表示，如 A、B、C、Y；变量取值为 0 的，以反变量表示，如 $\overline{A}$、$\overline{B}$、$\overline{C}$、$\overline{Y}$。将真值表中使函数值为 1 的输入变量以原变量或反变量的形式构成一个乘积项（"与"运算），如表 1.1 所示的真值表中，当 A 取 0、B 和 C 都取 1 时，函数值 Y 为 1，则构成的乘积项为 $\overline{A}BC$；将所有使函数值为 1 的乘积项相加("或"运算)，即得到逻辑函数标准的"与或"逻辑表达式。例如，从上面火灾报警控制电路的真值表中，可得到火灾报警电路的标准"与或"逻辑表达式为：

$$Y = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

4. 逻辑图

由逻辑符号表示逻辑函数的图形叫做逻辑电路图，简称逻辑图。

例如，逻辑表达式 $Y=\overline{A}BC+A\overline{B}C+AB\overline{C}+ABC$ 的逻辑图如图 1.13 所示。

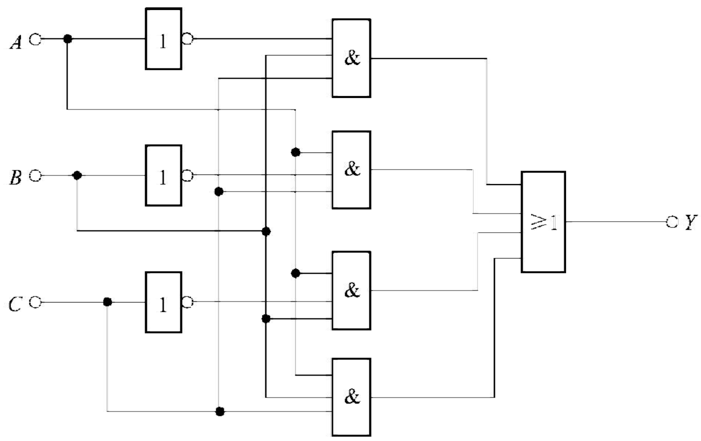


图 1.13　$Y=\overline{A}BC+A\overline{B}C+AB\overline{C}+ABC$ 的逻辑图

逻辑图的特点是逻辑图中的逻辑符号都有与之对应的实际电路器件存在，所以它比较接近工程实际。通过逻辑图，能将许多繁杂实际电路的逻辑功能层次分明地表示出来。但逻辑图不能用公式和定理进行运算和变换，所表示的逻辑关系没有真值表和卡诺图直观。

由逻辑图中层次分明的逻辑关系也很容易写出其逻辑表达式。例如，图 1.14 所示的逻辑图的逻辑表达式为：$Y=AB+\overline{A}C+BC$。

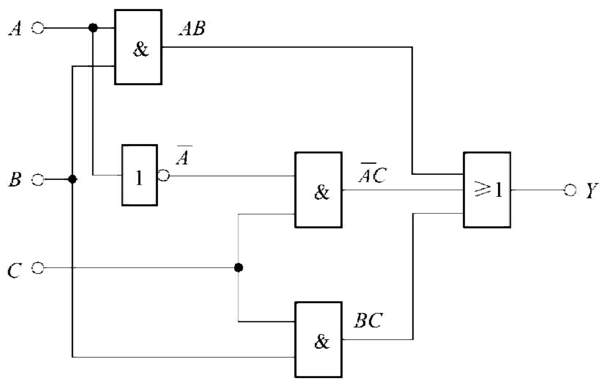


图 1.14　$Y = AB + \overline{A}C + BC$ 的逻辑图

5. 波形图

在给出输入变量取值随时间变化的波形后，根据逻辑函数中变量之间的逻辑关系、真值表或者卡诺图中变量取值和函数值的对应关系，都可对应画出输出变量（函数）随时间变化的波形。这种反映输入和输出变量对应取值随时间按照一定规律变化的图形，就叫波形图，也称为时序图。例如，$Y = A\overline{B} + \overline{A}B$，$A$、$B$ 的波形如图 1.15 所示，则可根据“异或”逻辑对应画出输出变量 Y 的波形。

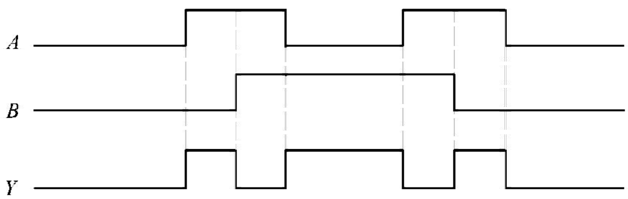


图 1.15　$Y = A\overline{B} + \overline{A}B$ 的波形图

六、逻辑代数的运算

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、…表示。变量的取值只有两种取值：“真”和“假”，一般“1”表示“真”，“0”表示“假”。表达式 $Y = F(A, B, C, \ldots)$ 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. 逻辑代数的基本公式

（**1**）变量和常量的逻辑加

$$A+0=A,\quad A+1=1$$

（2）变量和常量的逻辑乘

$$A\cdot 0=0,\quad A\cdot 1=A$$

（3）变量和反变量的逻辑加和逻辑乘

$$A+\overline{A}=1,\quad A\cdot\overline{A}=0$$

证明：当 $A=0$ 时

$$0+0=0,\quad 0+1=1$$

$$0\cdot 0=0,\quad 0\cdot 1=0$$

当 $A=1$ 时

$$1+1=1,\quad 1+0=1$$

$$1\cdot 1=1,\quad 1\cdot 0=0$$

2. 逻辑代数的基本定律

（1）交换律

$$A+B=B+A,\quad A\cdot B=B\cdot A$$

（2）结合律

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

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

（3）重叠律

$$A+A=A,\quad A\cdot A=A$$

（4）分配律

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

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

（5）吸收律

$$A + AB = A \text{，} \quad A \cdot (A + B) = A$$
$$A + \overline{A}B = A + B \text{，} \quad A(\overline{A} + B) = AB$$

（6）非非律

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

（7）反演律

$$\overline{A + B} = \overline{A} \cdot \overline{B} \text{，} \quad \overline{A \cdot B} = \overline{A} + \overline{B}$$

证明：

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

（8）还原律

$$AB + \overline{A}B = B \text{，} \quad (A + B)(A + \overline{B}) = A$$

（9）冗余律

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

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

七、逻辑函数的化简

在数字电路中，相同的逻辑功能可以用不同的逻辑函数表达式表示，而逻辑

函数表达式越简单，与之对应的逻辑电路也越简单，使用的器件就越少，越经济，可靠性越高。因此，必须对逻辑函数式化简。

逻辑函数表达式可以有多种形式，而且可以相互转换，例如：

$$Y = AC + \overline{A}B \qquad \text{(“与或”表达式)}$$

$$Y = AC + \overline{A}B = (A + B)(\overline{A} + C) \qquad \text{(“或与”表达式)}$$

$$Y = AC + \overline{A}B = \overline{\overline{AC} \cdot \overline{\overline{A}B}} \qquad \text{(“与非与非”表达式)}$$

$$Y = AC + \overline{A}B = \overline{\overline{A + B} + \overline{\overline{A} + C}} \qquad \text{(“或非或非”表达式)}$$

$$Y = AC + \overline{A}B = \overline{A\overline{C} + \overline{A}\overline{B}} \qquad \text{(“与或非”表达式)}$$

其中，“与或”式是最基本的函数表达式。

判断“与或”表达式是否最简的条件是：

①“与”项最少，即表达式中“+”号最少。

② 每个“与”项中的变量数最少，即表达式中“·”号最少。

化简逻辑函数的方法最常用的有公式法和卡诺图法。

1. 公式化简法

逻辑函数的公式化简法，就是利用逻辑代数的基本公式、基本定理和常用公式，将复杂的逻辑函数进行化简的方法。常用的有并项法、吸收法、消去法和配项法。

（**1**）并项法

运用公式 $A + \overline{A} = 1$，将两项合并为一项，消去一个变量。例如：

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

（2）吸收法

运用吸收律 $A+AB=A$，消去多余的“与”项。例如：

$$Y=A\overline{B}+A\overline{B}(C+DE)=A\overline{B}\left[1+(C+DE)\right]=A\overline{B}$$

（3）消去法

运用吸收法 $Y=A+\overline{A}B=A+B$ 消去多余的逻辑变量。例如：

$$Y=\overline{A}+AB+\overline{B}E=\overline{A}+B+\overline{B}E=\overline{A}+B+E$$

（4）配项法

先通过乘以 $A+\overline{A}$ 或加上 $A\overline{A}$ 增加必要的乘积项，再进行化简。例如：

$$
\begin{aligned}
Y &= AB+\overline{A}C+BCD=AB+\overline{A}C+BCD(A+\overline{A}) \\
&= AB+\overline{A}C+ABCD+\overline{A}CBD \\
&= AB(1+CD)+\overline{A}C(1+BD) \\
&= AB+\overline{A}C
\end{aligned}
$$

在前面的火灾报警控制电路中，我们从真值表得到其逻辑表达式，利用公式法进行化简：

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

在化简逻辑函数时，要灵活运用上述方法，才能将逻辑函数化为最简。

例 **1.8** 用公式法化简 $Y = AB + A\overline{C} + \overline{B}C + \overline{C}B + \overline{B}D + \overline{D}B + ADE(\overline{F} + G)$。

解 根据摩根定律有

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

可得

$$Y = A\overline{\overline{B}C} + \overline{B}C + \overline{C}B + \overline{B}D + \overline{D}B + ADE(\overline{F} + G)$$

根据公式 $A + \overline{A}B = A + B$ ，得

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

即 $$Y = A + \overline{B}C + \overline{C}B + \overline{B}D + \overline{D}B + ADE(\overline{F} + G)$$

根据公式 $A + AB = A$ ，得

$$A + ADE(\overline{F} + G) = A$$

即 $$Y = A + \overline{B}C + \overline{C}B + \overline{B}D + \overline{D}B$$

利用配项法再进行化简，可得

$$\begin{aligned}
Y &= A + \overline{B}C + \overline{C}B + \overline{B}D + \overline{D}B \\
&= A + \overline{B}C(D + \overline{D}) + \overline{C}B + \overline{B}D + \overline{D}B(C + \overline{C}) \\
&= A + \overline{B}CD + \overline{B}C\overline{D} + \overline{C}B + \overline{B}D + \overline{D}BC + \overline{D}B\overline{C} \\
&= A + (\overline{B}CD + \overline{B}D) + (\overline{B}C\overline{D} + \overline{D}BC) + (\overline{C}B + \overline{D}B\overline{C}) \\
&= A + \overline{B}D + \overline{D}C + \overline{C}B
\end{aligned}$$

2. 卡诺图化简法

卡诺图就是将逻辑函数变量的最小项按一定规则排列出来，构成的正方形或矩形的方格图。图中分成若干个小方格，每个小方格填入一个最小项，按一定的规则把小方格中所有的最小项进行合并处理，就可以得到化简的逻辑函数表达

式，这就是卡诺图化简。

（1）最小项及最小项表达式

设 A、B、C 是三个逻辑变量，若由这三个逻辑变量按以下规则构成乘积项：

① 每个乘积项都含三个变量，且每个变量都是它的一个因子。

② 每个变量都以反变量（$\overline{A}$、$\overline{B}$、$\overline{C}$）或以原变量（A、B、C）作为因子出现一次，且仅出现一次。

具备以上条件的乘积项共 8 个，我们称这 8 个乘积项为三变量 A、B、C 的最小项。

推广：由于一个变量仅有原变量和反变量两种形式，因此 N 个变量共有 2^N 个最小项。

最小项的定义：对于 N 个变量，如果 P 是一个含有 N 个因子的乘积项，而且每一个变量都以原变量或者反变量的形式作为一个因子在 P 中出现且仅出现一次，那么就称 P 是这 N 个变量的一个最小项。

两个变量 A、B 的最小项为 $\overline{A}\overline{B}, \overline{A}B, A\overline{B}, AB$；三个变量 A、B、C 的最小项为 $\overline{A}\overline{B}\overline{C}, \overline{A}\overline{B}C, \overline{A}B\overline{C}, \overline{A}BC, A\overline{B}\overline{C}, A\overline{B}C, AB\overline{C}, ABC$，依此类推四个变量、五个变量…的最小项。

最小项也可用“m_i”表示，下标“i”即最小项的编号。编号方法：把最小项所对应的那一组变量取值组合当成二进制数，与其相应的十进制数就是该最小项的编号。例如，最小项 $\overline{A}\overline{B}C = m_1, A\overline{B}C = m_5$。

任何一个逻辑函数都可以表示为最小项之和的形式，称为标准“与或”表达式。而且这种形式是唯一的，也就是说，一个逻辑函数只有一种最小项表达式。例如，我们前面得到的火灾控制电路的表达式 $Y = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$，就是标准的“与或”式。

也可将 $Y = AB + BC$ 展开成标准的“与或”表达式，即

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

或 $$Y(A,\ B,\ C)=m_3+m_6+m_7=\sum m(3,\ 6,\ 7)$$

（2）卡诺图及其画法

卡诺图是把最小项按照一定规则排列而构成的方框图。构成卡诺图的原则是：

① N 变量的卡诺图有 2^N 个小方块（最小项）。

② 最小项排列规则：几何相邻的必须逻辑相邻，即两相邻小方格所代表的最小项只有一个变量取值不同。要满足几何相邻，变量取值必须按循环码的顺序取值。

循环码与二进制数码之间有着对应的关系。例如，一个 n 位二进制码 $A=A_{n-1}A_{n-2}\ldots A_i\ldots A_1A_0$，其对应的循环码用 $B=B_{n-1}B_{n-2}\ldots B_i\ldots B_1B_0$ 表示，则对应位循环码 $B_i=A_i\oplus A_{i+1}$。又如，三位二进制码 110 的循环码为 101。循环码的特点是相邻两个码之间只有一位不同。

几何相邻的含义：

① 相邻——紧挨的（上下、左右）。

② 相对——任一行或一列的两头。

③ 相重——对折起来后位置相重。

满足上述其中之一的都称为几何相邻或逻辑相邻。

图 1.16 所示为两个变量的卡诺图，图 1.17、图 1.18 为三变量和四变量的卡诺图。

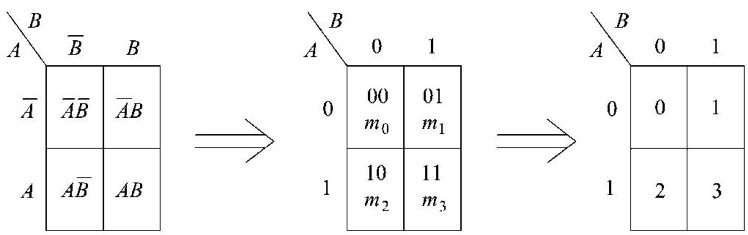


图 1.16 两个变量的卡诺图

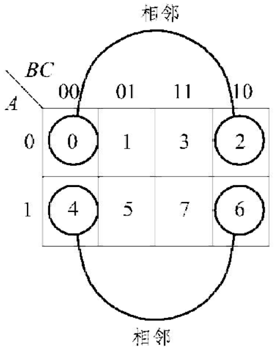


图 1.17　三变量的卡诺图

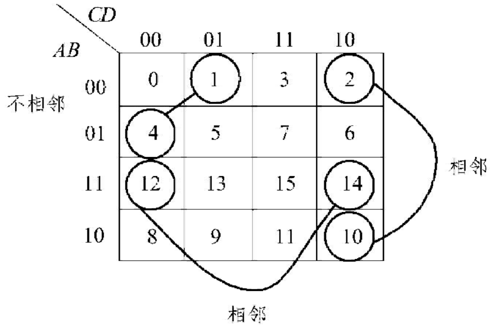


图 1.18　四变量的卡诺图

（3）用卡诺图表示逻辑函数

a. 从真值表画卡诺图

根据变量个数画出卡诺图，再按真值表填写每一个小方块的值（0 或 1）即可。需注意二者顺序不同。

例如，已知逻辑函数 Y 的真值表，确定 Y 对应的卡诺图，如图 1.19 所示。

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

⇒

A \ BC	00	01	11	10
0			1	
1		1	1	1

图 1.19　由真值表确定卡诺图

b. 从最小项表达式画卡诺图

把表达式中所有的最小项在对应的小方块中填入 1，其余的小方块中填入 0（0 一般不用写出）。

例如，函数 $Y(A、B、C、D) = \sum m(0，3，5，7，9，12，15)$ 的卡诺图如图