第3章

逻辑代数与逻辑函数

逻辑代数是数字电路分析和设计的主要数学工具。本章介绍逻辑函数、逻辑代数的基本运算、逻辑函数化简及逻辑函数门电路的实现等内容。

3.1 逻辑代数的基本运算

数字电路研究的是数字电路的输入与输出之间的因果关系，即逻辑关系。逻辑关系一般由逻辑函数来描述。逻辑函数是由逻辑变量A、B、C…和基本逻辑运算符号·（与）、+（或）、—（非）及括号、等号等构成的表达式来表示，如：

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

式中：A、B、C称为原变量，$\overline{A}$、$\overline{B}$、$\overline{C}$称为对应的反变量，F称为逻辑函数（$\overline{F}$称为逻辑反函数）。

3.1.1 基本运算公式

与（乘）	或（加）	非
$A \cdot 0 = 0$	$A + 0 = A$	
$A \cdot 1 = A$	$A + 1 = 1$	
$A \cdot A = A$	$A + A = A$	$\overline{\overline{A}} = A$
$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$	

3.1.2 基本运算定律

交换律	$A \cdot B = B \cdot A$	$A + B = B + A$
结合律	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$
分配律	$A \cdot (B + C) = AB + AC$	$A + B \cdot C = (A + B)(A + C)$
吸收律	$A(A + B) = A$	$A + AB = A$
	$A + \overline{A}B = A + B$	$(A + B)(A + C) = A + BC$
	$AB + \overline{A}C + BC = AB + \overline{A}C$	
反演律	$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$

以上这些定律可以用基本公式或真值表进行证明。

例 3.1.1 利用基本公式证明 $AB+\overline{A}C+BC=AB+\overline{A}C$。

证：左边 $=AB+\overline{A}C+(A+\overline{A})BC$

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

$=AB(1+C)+\overline{A}C(1+B)$

$=AB+\overline{A}C=$ 右边

例 3.1.2 利用真值表证明反演律(也称摩根定理)。

证：可将变量 A、B 的各种取值分别代入等式两边，其真值表如表 3.1.1 所示。从真值表可看出，等式两边的逻辑值完全对应相等，所以定理成立。

表 3.1.1 证明摩根定理的真值表

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

上面所列出的运算定理反映了逻辑关系，而不是数量之间的关系，因而在逻辑运算时不能简单套用初等代数的运算规则。例如，在逻辑运算中不能套用初等代数的移项规则，这是由于逻辑代数中没有减法和除法的缘故。

3.1.3 基本运算规则

1. 运算顺序

在逻辑代数中，运算优先顺序为：先算括号，再是非运算，然后是与运算，最后是或运算。

2. 代入规则

在逻辑等式中，如果将等式两边出现某一变量的位置都代之以一个逻辑函数，则等式仍然成立。这就是代入规则。

例如，已知$\overline{A\cdot B}=\overline{A}+\overline{B}$。若用 $Z=A\cdot C$ 代替等式中的 A，根据代入规则，等式仍然成立，即

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

摩根定律可以扩展对任意多个变量都成立。由此可见，代入规则可以扩展所有基本定律的应用范围。

3. 反演规则

已知函数 F 求其反函数$\overline{F}$时，只要将 F 式中的 1 换成 0，0 换成 1，· 换成+，+换成 ·，原变量换成反原变量，反变量换成原变量，所得到的表达式就是$\overline{F}$表达式。这就是反演规则。利用反演规则能较容易地求出一个函数的反函数。

例如，求 $F=\overline{A}\,\overline{B}+CD+0$ 和 $L=A+\overline{B\,\overline{C}+\overline{D+\overline{E}}}$ 的反函数。根据反演规则可求得：

$$\overline{F}=(A+B)\cdot(\overline{C}+\overline{D})\cdot 1=(A+B)(\overline{C}+\overline{D})$$

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

运用反演规则时必须注意两点：

(1) 保持原来的运算优先顺序。

(2) 对于反变量以外的非号应保留不变。

4. 对偶规则

将逻辑函数 F 中所有的 1 换成 0，0 换成 1，· 换成＋，＋换成 ·，变量保持不变，得到新函数 F'。F' 称为 F 的对偶式。例如：

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

变换时仍需注意保持原式中先与后或的顺序。

如果某个逻辑恒等式成立时，则其对偶式也成立，这就是对偶规则。

3.2 逻辑函数的变换和化简

3.2.1 逻辑函数变换和化简的意义

利用基本逻辑运算可以将同一个逻辑函数变换为不同的表达式，例如，$F=\overline{A}B+AC$ 可写为：

$$F=\overline{A}B+AC+BC$$

或

$$F=\overline{\overline{\overline{A}B}\;\overline{AC}}$$

这样描述同一个逻辑函数有以上 3 个不同的表达式，若用逻辑门实现这 3 个表达式有 3 种不同的门电路，如图 3.2.1 所示。

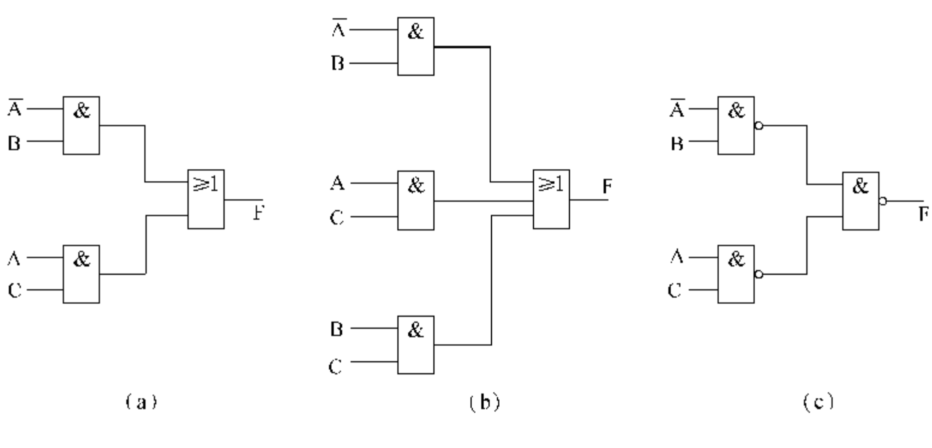


图 3.2.1　同一个函数三种不同的电路

由图 3.2.1 可知，表达式复杂，实现的电路就复杂，表达式运算的种类多，实现电路所需门电路的种类就多。实际应用中，逻辑函数与逻辑电路之间存在着一一对应的关系，将较烦琐的逻辑表达式变换为与之等效的简化表达式意味着可以用较少的逻辑元件和较少的输入端来实现相同的逻辑功能，电路越简单，可靠性越高，而成本越低。所以需要进行逻辑函数的变化和化简。

3.2.2 逻辑函数的代数法变换

一个逻辑函数通常有以下 5 种类型的表达式：

类型 1：与或表达式

$$F = A + BC$$

类型 2：或与表达式

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

类型 3：与非一与非表达式(简称与非表达式)

$$F = \overline{\overline{A} \cdot \overline{BC}}$$

类型 4：或非一或非表达式(简称或非表达式)

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

类型 5：与或非表达式

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

以上 5 个表达式是同一函数不同形式的表达式，利用逻辑代数的基本运算可以将逻辑函数变换为不同形式的表达式。例如，运用摩根定律可以将与或表达式变换为与非表达式，即

$$F = A + BC = \overline{\overline{A} \cdot \overline{BC}}$$

从而可以用与非门电路来实现；运用摩根定律还可以将或与表达式变换为或非表达式，即

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

从而可以用或非门电路来实现；运用摩根定律还可以将或非表达式变换为与或非表达式，即

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

例 3.2.1 已知函数 Y＝AB＋BC，试写出与非表达式和或非表达式。

解 根据逻辑代数运算：

$$Y = AB + BC = \overline{\overline{AB} \cdot \overline{BC}}$$

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

3.2.3 逻辑函数的代数法化简

由于与或表达式易于从真值表直接写出，因此，我们以与或表达式为例，讨论逻辑函数代数法化简的方法。

最简的与或表达式有以下两个特点：

(1) 与项(即乘积项)的个数最少。

(2) 每个与项中变量的个数最少。

代数法化简逻辑函数是运用逻辑代数的基本定律和基本公式进行化简，常用下列方法。

1. 吸收法

利用吸收律公式 $A+AB=A$，消去多余项。例如：

$$
\begin{aligned}
F &= A\bar{C} + A\bar{C}D(E+F) \\
&\quad\text{(应用公式 } A + AB = A\text{)} \\
&= A\bar{C}
\end{aligned}
$$

2. 并项法

利用公式 $A+\bar{A}=1$，将两项合并成一项，并消去一个变量。如：

$$
\begin{aligned}
F &= AB\bar{C} + A\overline{B\bar{C}} \\
&\quad\text{(应用分配律)} \\
&= A(B\bar{C} + \overline{B\bar{C}}) \\
&\quad\text{(应用 } A + \bar{A} = 1\text{)} \\
&= A \cdot 1 \\
&= A
\end{aligned}
$$

3. 消去因子法

利用公式 $A+\bar{A}B=A+B$，消去多余的因子，如：

$$
\begin{aligned}
F &= AB + \bar{A}C + \bar{B}C \\
&\quad\text{(应用分配律)} \\
&= AB + (\bar{A} + \bar{B})C \\
&\quad\text{(应用反演律)} \\
&= AB + \overline{AB}C \\
&\quad\text{(应用吸收律)} \\
&= AB + C
\end{aligned}
$$

4. 配项消项

先利用公式 $A+\bar{A}=1$ 增加必要的乘积项，再用并项或吸收的办法使项数减少，如：

$$
\begin{aligned}
F &= AB + \bar{A}\bar{C} + B\bar{C} \\
&\quad\text{(应用 } A + \bar{A} = 1 \text{ 和 } A \cdot 1 = A\text{)} \\
&= AB + \bar{A}\bar{C} + (\bar{A} + A)B\bar{C} \\
&\quad\text{(应用分配律)} \\
&= AB + \bar{A}\bar{C} + AB\bar{C} + \bar{A}B\bar{C} \\
&\quad\text{(应用交换律)} \\
&= (AB + AB\bar{C}) + (\bar{A}\bar{C} + \bar{A}\bar{C}B) \\
&\quad\text{(应用吸收律)} \\
&= AB + \bar{A}\bar{C}
\end{aligned}
$$

3.3 逻辑函数的卡诺图法化简法与变换

利用代数化简逻辑函数不但要求熟练掌握逻辑代数的基本公式,而且需要一些技巧,特别是较难掌握获得代数化简后的最简逻辑表达式的方法。下面介绍的卡诺图化简法能直接获得最简与或表达式,并且易于掌握。

3.3.1 最小项

n 个变量 $X_1, X_2, \cdots, X_n$ 的最小项由 n 个变量的乘积项构成,每个变量都以它的原变量或反变量的形式在乘积项中出现,且仅出现一次,共有 2^n 个不同的乘积项。

举例来说,设 A、B、C 是 3 个逻辑变量,由这 3 个原变量和反变量的不同组合可以构成 8 个乘积项,这些乘积项中各变量只仅出现一次,这 8 项即为最小项,每个最小项有 3 个变量。变量与最小项真值表如表 3.3.1 所示。

表 3.3.1 变量与最小项真值表

A B C	$\overline{A}\,\overline{B}\,\overline{C}$	$\overline{A}\,\overline{B}C$	$\overline{A}B\overline{C}$	$\overline{A}BC$	$A\overline{B}\,\overline{C}$	$A\overline{B}C$	$AB\overline{C}$	ABC
0 0 0	1	0	0	0	0	0	0	0
0 0 1	0	1	0	0	0	0	0	0
0 1 0	0	0	1	0	0	0	0	0
0 1 1	0	0	0	1	0	0	0	0
1 0 0	0	0	0	0	1	0	0	0
1 0 1	0	0	0	0	0	1	0	0
1 1 0	0	0	0	0	0	0	1	0
1 1 1	0	0	0	0	0	0	0	1

观察表 3.3.1 可得到最小项的性质。

性质 1:对于任意一个最小项,只有一组变量的取值使其值为 1,而在变量取其他各组值时这个最小项的取值都是 0。例如,$\overline{A}B\overline{C}$最小项为 1 的对应于变量组的取值是 010,除此之外,其他变量组取值都使$\overline{A}B\overline{C}=0$。

性质 2:对于变量的任一组取值,任意两个最小项之积为 0。

性质 3:对于变量的一组取值,全部最小项之和为 1。

最小项可以用变量表示,也可用符号 m_i 表示。下标 i 是该最小项值为 1 时对应的变量组取值的十进制等效值,如最小项$\overline{A}B\overline{C}$记作 m_2,下标 2 对应最小项$\overline{A}B\overline{C}=1$ 时变量组取值 010,而 010 相当于十进制的 2,因而下标 $i=2$。显然,最小项各变量与变量组取值(即 m_i 中的下标 i)的关系是:变量组取值中的 1 对应最小项中的原变量;变量组取值中的 0 对应最小项中的反变量。由此可见表 3.3.1 中从左到右的 8 个最小项的表示符号分别为 $m_0, m_1, m_2, \cdots, m_7$。

3.3.2　逻辑函数的最小项表达式

由前面的讨论得知，对于某种逻辑关系，用真值表来表示是唯一的，而用前面讨论的逻辑表达式来表示可以有多个表达式。如果用最小项之和组成的表达式来表示，也是唯一的。由最小项之和组成的表达式称为逻辑函数标准与或表达式，也称为最小项表达式。

1. 由真值表求最小项表达式

根据给定的真值表，利用最小项性质1，可以直接写出最小项表达式。例如，对于表3.3.2所示的3变量真值表，左边每一组输入变量的取值代表一个最小项，右边每一个输出变量的值代表一个最小项的逻辑取值，由真值表可知，输出变量F为1的条件是：

$A=0,B=1,C=1$，即最小项$\overline{A}BC=m_3=1$；

$A=1,B=0,C=1$，即最小项 $A\overline{B}C=m_5=1$；

$A=1,B=1,C=0$，即最小项 $AB\overline{C}=m_6=1$；

$A=1,B=1,C=1$，即最小项 $ABC=m_7=1$。

4个条件中满足一个F就是1，所以F的表达式可以写成最小项之和的形式，即

$$F=\overline{A}BC+A\overline{B}C+AB\overline{C}+ABC=m_3+m_5+m_6+m_7$$

或写成

$$\begin{aligned}F(A,B,C)&=\sum m(3,5,6,7)\\&=\sum(3,5,6,7)\end{aligned}$$

表3.3.2　逻辑函数真值表

A	B	C	F
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. 由一般表达式转换为最小项表达式

任何一个逻辑函数表达式都可以转换为最小项表达式。例如，将$F(A,B,C)=AB+\overline{A}C$转换为最小项表达式时，可利用逻辑运算关系$(A+\overline{A})=1$，将逻辑函数中的每一项转换成包含所有变量的项，即

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

$$= m_7 + m_6 + m_3 + m_1$$
$$= \sum(1,3,6,7)$$

又如，要将 $F(A,B,C) = AB\overline{C} + A\overline{B\overline{C}}$ 转换为最小项表达式，虽然逻辑函数中的每一项都包含有3个变量，但 $A\overline{B\overline{C}}$ 项不是只由原变量和反变量组成的3个变量乘积项，因此逻辑函数表达式不是最小项表达式，化成最小项表达式的具体步骤如下：

(1) 利用摩根定理去掉非号，即

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

(2) 利用分配律除去括号，即

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

(3) 用公式 $(B+\overline{B})=1$、$(C+\overline{C})=1$，将逻辑函数中的每一项转换成包含所有变量的项，即

$$\begin{aligned} F(A,B,C) &= AB\overline{C} + A\overline{B}(C+\overline{C}) + AC(B+\overline{B}) \\ &= AB\overline{C} + A\overline{B}C + A\overline{B}\,\overline{C} + ABC \\ &= m_6 + m_5 + m_4 + m_7 \\ &= \sum(4,5,6,7) \end{aligned}$$

3.3.3 卡诺图

卡诺图是真值表的图形表示。图3.3.1分别表示了二变量、三变量、四变量和五变量的卡诺图。

A \ B	0	1
0	0	1
1	2	3

(a) 二变量

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

(b) 三变量

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

(c) 四变量

AB \ CDE	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

(d) 五变量

图3.3.1 卡诺图

有关卡诺图的说明：

(1) 卡诺图中的每一个方格代表一个最小项，方格内的数字表示相应最小项的下标，最小项的逻辑取值填入相应方格。

(2) 卡诺图方格外为输入变量及其相应逻辑取值，变量取值的排序不能改变。

(3) 卡诺图中相邻的两个方格称为逻辑相邻项，相邻项中只有一个变量互为反变量，而其余变量完全相同。图3.3.1(b)中4、5相邻方格对应的最小项分别为 $A\overline{B}\,\overline{C}$、$A\overline{B}C$。除相邻的两个方格是相邻项外，卡诺图左右两侧、上下两侧相对的方格也是相邻项。图3.3.1(c)中方格4、6为相邻项，方格1、9也为相邻项。

3.3.4 逻辑函数的卡诺图表示

由逻辑函数的真值表和表达式可以直接画出逻辑函数的卡诺图。

1. 由逻辑函数真值表直接画出的卡诺图

真值表的每一行对应一个最小项及相应的逻辑取值，因此，也对应卡诺图中的一个方格及该方格的逻辑值，将表 3.3.2 真值表的取值填入卡诺图对应方格中，即可画出卡诺图，如图 3.3.2 所示。

2. 由逻辑函数表达式画出的卡诺图

若函数表达式是最小项表达式，例如，$F(A,B,C,D)=\sum m(0,1,3,5,10,11,12,15)$，可根据图 3.3.1 所示的四变量卡诺图的形式，将上述逻辑函数最小项表达式中的各项，在卡诺图对应方格内填入 1，即在四变量卡诺图中，将与最小项 $m_0, m_1, m_3, m_5, m_{10}, m_{11}, m_{12}, m_{15}$ 对应的格内填入 1，其余的方格内均填入 0，最后得出如图 3.3.3 所示的 F 函数的卡诺图。

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

图 3.3.2 真值表的卡诺图

AB \ CD	00	01	11	10
00	1	1	1	0
01	0	1	0	0
11	1	0	1	0
10	0	0	1	1

图 3.3.3 函数 F 的卡诺图

若函数表达式是非最小项表达式，可先转换成最小项表达式，再画出其卡诺图。例如，$G(A,B,C)=AB+BC+AC$，有：

$$\begin{aligned} G(A,B,C) &= AB+BC+AC \\ &= AB(C+\overline{C})+BC(A+\overline{A})+AC(B+\overline{B}) \\ &= ABC+AB\overline{C}+A\overline{B}C+\overline{A}BC \\ &= m_7+m_6+m_5+m_3 \end{aligned}$$

在对应的卡诺图 3.3.1(b)中最小项下标为 3,5,6,7 的方格内填入 1，其余的方格内均填入 0，便得到如图 3.3.4 所示的卡诺图。

也可由非最小项表达式的函数表达式直接画出卡诺图。例如，$L(A,B,C)=A+BC$。与项 A 对应卡诺图 A=1 一行的 4 个方格，而与项 BC 对应卡诺图 BC=11 的一列两个方格，在这些方格中填 1，其余方格中填 0，即可得到函数 L 的卡诺图，如图 3.3.5 所示。

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

图 3.3.4 函数 G 的卡诺图

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

图 3.3.5 函数 L 的卡诺图

3.3.5 逻辑函数的卡诺图化简

1. 化简依据

卡诺图中任何两个为1的相邻方格的最小项可以合并为一个与项，并且消去一个变量，如 $F(A,B,C)=\sum m(6,7)=AB\overline{C}+ABC=AB$，被消去的变量在这个最小项中以互为反变量的形式出现，如此处的C变量；4个为1的相邻方格的最小项可以合并成一个与项，并消去2个变量，如 $F(A,B,C)=\sum m(4,5,6,7)=A\overline{B}\,\overline{C}+A\overline{B}C+AB\overline{C}+ABC=A\overline{B}(\overline{C}+C)+AB(\overline{C}+C)=A(\overline{B}+B)=A$，由于B、C和$\overline{B}$、$\overline{C}$均出现在最小项中，故被消去；8个为1的相邻最小项，可以合并成一个与项，并消去3个变量，例如：

$$
\begin{aligned}
F(A,B,C,D) &= \sum m(8,9,10,11,12,13,14,15)\\
&= A\overline{B}\,\overline{C}\,\overline{D}+A\overline{B}\,\overline{C}D+A\overline{B}C\overline{D}+A\overline{B}CD+AB\overline{C}\,\overline{D}\\
&\quad +AB\overline{C}D+ABC\overline{D}+ABCD\\
&= A\overline{B}\,\overline{C}(\overline{D}+D)+A\overline{B}C(\overline{D}+D)+AB\overline{C}(\overline{D}+D)+ABC(\overline{D}+D)\\
&= A\overline{B}(\overline{C}+C)+AB(\overline{C}+C)\\
&= A(\overline{B}+B)\\
&= A
\end{aligned}
$$

由于$\overline{B}$、B、$\overline{C}$、C、$\overline{D}$、D均出现在最小项中，故被消去。

由此可见：卡诺图中 2^k 个为1的相邻最小项，可以合并成一个与项，并消去 k 个变量，$k=0,1,2,\cdots$。

2. 化简原则

(1) 将为1的相邻方格用线包围起来，包围圈内的方格越多越好，但应满足 2^k 个，包围圈的数目越少越好。

(2) 每个为1的方格可重复使用，每个包围圈内至少含有一个新的为1的方格，每个为1的方格都要圈起来。

(3) 将所有包围圈内的最小项合并成对应与项，然后相加，最后得到的逻辑函数就是最简与或表达式。

例 3.3.1 试用卡诺图化简函数 $F(A,B,C)=\overline{A}BC+A\overline{B}\,\overline{C}+ABC+AB\overline{C}$。

解 步骤1：画出卡诺图，如图3.3.6所示。

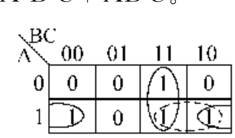


图 3.3.6 例 3.3.1 卡诺图

步骤2：圈出相邻为1的包围圈，共2个，每个含有2个方格，即圈出 m_3、m_7 和 m_4、m_6 2个包围圈。

步骤3：合并圈出的相邻项并相加，2个相邻最小项，可以消去1个变量，合并成一个与项，m_3 和 m_7 相邻最小项合并得一个与项BC，m_4 和 m_6 合并后的与项为 $A\overline{C}$，相加后得到最简表达式，即：

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

注意：画虚线的圈内，没有新的为1的方格，因此无效。

例 3.3.2　试用卡诺图化简函数 $F(A,B,C,D)=\sum m(0,1,2,4,5,6,8,9,12,13,14)$。

解　画出卡诺图,圈出相邻项,如图 3.3.7 所示。

卡诺图中,一组为 8 个 1 的相邻项,这 8 个最小项中只有变量 C 的取值均为 0,其他变量取值均成对为 0 和 1,故可以消去 3 个变量合并成一个与项$\overline{C}$,两组为 4 个 1 的相邻最小项,其中一组中的变量 B 和 C 的取值均成对为 0 和 1,故被消去,合并成$\overline{A}\,\overline{D}$,而另一组按同样的原则消去 A 和 C,合并成 $B\overline{D}$,化简后的函数为:

$$F=\overline{C}+\overline{A}\,\overline{D}+B\overline{D}$$

例 3.3.3　试用卡诺图化简函数 $F(A,B,C,D)=\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}+\overline{A}\,\overline{B}C\overline{D}+A\overline{B}\,\overline{C}\,\overline{D}+\overline{A}B\overline{C}D+A\overline{B}C\overline{D}$。

解　画出卡诺图,圈出相邻项,如图 3.3.8 所示。

卡诺图中,4 个角上的 1 可以圈在一起,形成与项$\overline{B}\,\overline{D}$,独立的 1 直接形成与项$\overline{A}B\overline{C}D$,化简后的函数为:

$$F=\overline{B}\,\overline{D}+\overline{A}B\overline{C}D$$

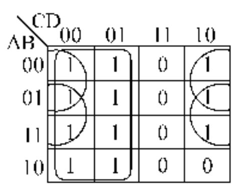


图 3.3.7　例 3.3.2 卡诺图

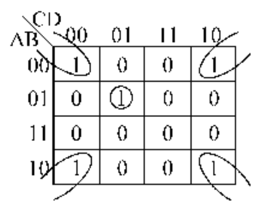


图 3.3.8　例 3.3.3 卡诺图

利用卡诺图表示逻辑函数式时,如果卡诺图中各小方格被 1 占去了大部分,虽然可用包围 1 的方法进行化简,但由于要重复利用 1 项,往往显得零乱而易出错。这时采用包围 0 的方法简化更为简单。求出反函数 F 后,再对反函数求非,其结果相同,下面举例说明。

例 3.3.4　化简逻辑函数 $F(A,B,C,D)=\sum m(0\sim3,5\sim11,13\sim15)$。

解　(1) 由 F 画出卡诺图,如图 3.3.9(a)所示。

(2) 用包围 1 的方法化简,如图 3.3.9(b)所示。得:

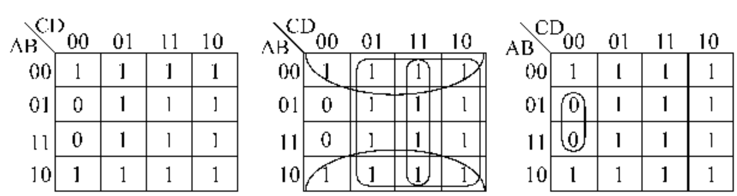
$$F=\overline{B}+C+D$$

(3) 用包围 0 的方法化简,如图 3.3.9(c)所示。得:

$$\overline{F}=B\overline{C}\,\overline{D}$$

$$F=\overline{\overline{F}}=\overline{B}+C+D$$

两种方法结果相同。



(a) 卡诺图　(b) 卡诺图圈 1　(c) 卡诺图圈 0

图 3.3.9　例 3.3.4 卡诺图