**解：** 从输入端 $A$、$B$ 开始逐个写出每个图形符号输出端的逻辑式，得到 $Y=((A+B)'+(A'+B')')'$。将该式变换后得到

$$Y=((A+B)'+(A'+B')')'=(A+B)(A'+B')$$
$$=AB'+A'B=A\oplus B$$

可见，输出 $Y$ 和 $A$、$B$ 间是**异或**逻辑关系。

3. 波形图与真值表的相互转换

在从已知的逻辑函数波形图求对应的真值表时，首先需要从波形图上找出每个时间段里输入变量与函数输出的取值，然后将这些输入、输出取值对应列表，就得到了所求的真值表。

在将真值表转换为波形图时，只需将真值表中所有的输入变量与对应的输出变量取值依次排列画成以时间为横轴的波形，就得到了所求的波形图，如我们前面已经做过的那样。

**【例 2.5.5】** 已知逻辑函数 $Y$ 的波形图如图 2.5.6 所示，试求该逻辑函数的真值表。

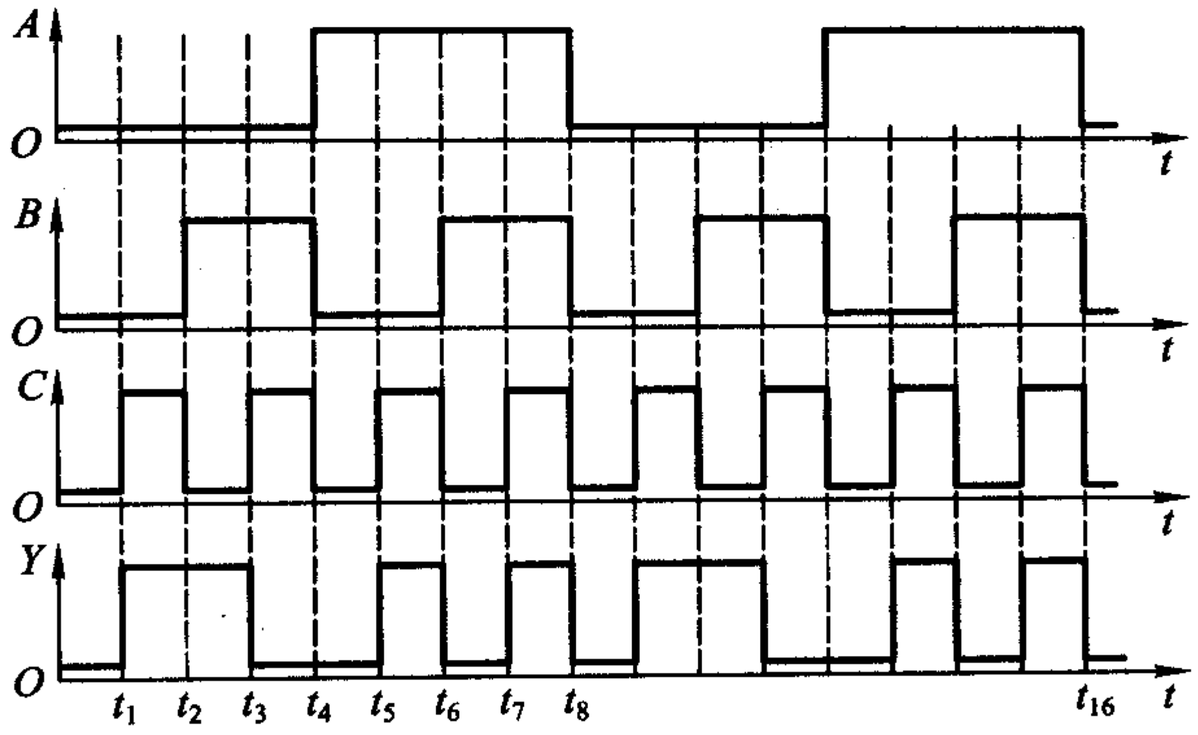


图 2.5.6　例 2.5.5 的波形图

**解：** 从 $Y$ 的波形图上可以看出，在 $0\sim t_8$ 时间区间里输入变量 $A$、$B$、$C$ 所有可能的取值组合均已出现了，而且 $t_8\sim t_{16}$ 区间的波形只不过是 $0\sim t_8$ 区间波形的重复。因此，只要将 $0\sim t_8$ 区间每个时间段里 $A$、$B$、$C$ 与 $Y$ 的取值对应列表，即可得表 2.5.4 所示的真值表。

**表 2.5.4　例 2.5.5 的真值表**

| $A$ | $B$ | $C$ | $Y$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |

续表

| A | B | C | Y |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

### 2.5.3 逻辑函数的两种标准形式

在讲述逻辑函数的标准形式之前，先介绍一下最小项和最大项的概念，然后再介绍逻辑函数的“最小项之和”及“最大项之积”这两种标准形式。

**一、最小项和最大项**

1. 最小项

在 $n$ 变量逻辑函数中，若 $m$ 为包含 $n$ 个因子的乘积项，而且这 $n$ 个变量均以原变量或反变量的形式在 $m$ 中出现一次，则称 $m$ 为该组变量的最小项。

例如，$A$、$B$、$C$ 三个变量的最小项有 $A'B'C'$、$A'B'C$、$A'BC'$、$A'BC$、$AB'C'$、$AB'C$、$ABC'$、$ABC$ 共 8 个（即 $2^3$ 个）。$n$ 变量的最小项应有 $2^n$ 个。

输入变量的每一组取值都使一个对应的最小项的值等于 1。例如，在三变量 $A$、$B$、$C$ 的最小项中，当 $A=1$、$B=0$、$C=1$ 时，$AB'C=1$。如果把 $AB'C$ 的取值 101 看作一个二进制数，那么它所表示的十进制数就是 5。为了今后使用的方便，将 $AB'C$ 这个最小项记作 $m_5$。按照这一约定，就得到了三变量最小项的编号表，如表 2.5.5 所示。

**表 2.5.5 三变量最小项的编号表**

| 最小项 | 使最小项为 1 的变量取值 | | | 对应的十进制数 | 编 号 |
|---|---|---|---|---|---|
| | A | B | C | | |
| $A'\ B'\ C'$ | 0 | 0 | 0 | 0 | $m_0$ |
| $A'\ B'\ C$ | 0 | 0 | 1 | 1 | $m_1$ |
| $A'\ B\ C'$ | 0 | 1 | 0 | 2 | $m_2$ |
| $A'\ B\ C$ | 0 | 1 | 1 | 3 | $m_3$ |
| $A\ B'\ C'$ | 1 | 0 | 0 | 4 | $m_4$ |
| $A\ B'\ C$ | 1 | 0 | 1 | 5 | $m_5$ |
| $A\ B\ C'$ | 1 | 1 | 0 | 6 | $m_6$ |
| $A\ B\ C$ | 1 | 1 | 1 | 7 | $m_7$ |

根据同样的道理，我们将 $A$、$B$、$C$、$D$ 这 4 个变量的 16 个最小项记作 $m_0 \sim m_{15}$。

从最小项的定义出发可以证明它具有如下的重要性质：

① 在输入变量的任何取值下必有一个最小项，而且仅有一个最小项的值为 **1**。

② 全体最小项之和为 **1**。

③ 任意两个最小项的乘积为 **0**。

④ 具有相邻性的两个最小项之和可以合并成一项并消去一对因子。

若两个最小项只有一个因子不同，则称这两个最小项具有相邻性。例如，$A'BC'$和 $ABC'$两个最小项仅第一个因子不同，所以它们具有相邻性。这两个最小项相加时定能合并成一项并将一对不同的因子消去

$$A'BC' + ABC' = (A' + A)BC' = BC'$$

* 2. 最大项

在 $n$ 变量逻辑函数中，若 $M$ 为 $n$ 个变量之和，而且这 $n$ 个变量均以原变量或反变量的形式在 $M$ 中出现一次，则称 $M$ 为该组变量的最大项。

例如，三变量 $A$、$B$、$C$ 的最大项有 $(A' + B' + C')$、$(A' + B' + C)$、$(A' + B + C')$、$(A' + B + C)$、$(A + B' + C')$、$(A + B' + C)$、$(A + B + C')$、$(A + B + C)$ 共 8 个（即 $2^3$ 个）。对于 $n$ 个变量则有 $2^n$ 个最大项。可见，$n$ 变量的最大项数目和最小项数目是相等的。

输入变量的每一组取值都使一个对应的最大项的值为 **0**。例如，在三变量 $A$、$B$、$C$ 的最大项中，当 $A = 1$、$B = 0$、$C = 1$ 时，$(A' + B + C') = 0$。若将使最大项为 **0** 的 $ABC$ 取值视为一个二进制数，并以其对应的十进制数给最大项编号，则 $(A' + B + C')$ 可记作 $M_5$。由此得到的三变量最大项编号表，如表 2.5.6 所示。

**表 2.5.6　三变量最大项的编号表**

| 最大项 | 使最大项为 0 的变量取值 | | | 对应的十进制数 | 编　号 |
|---|---|---|---|---|---|
| | $A$ | $B$ | $C$ | | |
| $A + B + C$ | 0 | 0 | 0 | 0 | $M_0$ |
| $A + B + C'$ | 0 | 0 | 1 | 1 | $M_1$ |
| $A + B' + C$ | 0 | 1 | 0 | 2 | $M_2$ |
| $A + B' + C'$ | 0 | 1 | 1 | 3 | $M_3$ |
| $A' + B + C$ | 1 | 0 | 0 | 4 | $M_4$ |
| $A' + B + C'$ | 1 | 0 | 1 | 5 | $M_5$ |
| $A' + B' + C$ | 1 | 1 | 0 | 6 | $M_6$ |
| $A' + B' + C'$ | 1 | 1 | 1 | 7 | $M_7$ |

根据最大项的定义同样也可以得到它的主要性质,这就是:

① 在输入变量的任何取值下必有一个最大项,而且只有一个最大项的值为 0。

② 全体最大项之积为 0。

③ 任意两个最大项之和为 1。

④ 只有一个变量不同的两个最大项的乘积等于各相同变量之和。

如果将表 2.5.5 和表 2.5.6 加以对比则可发现,最大项和最小项之间存在如下关系

$$M_i = m_i' \tag{2.5.2}$$

例如,$m_0 = A'B'C'$,则 $m_0' = (A'B'C')' = A + B + C = M_0$

**二、逻辑函数的最小项之和形式**

首先将给定的逻辑函数式化为若干乘积项之和的形式(亦称“积之和”形式),然后再利用基本公式 $A + A' = 1$ 将每个乘积项中缺少的因子补全,这样就可以将**与或**的形式化为最小项之和的标准形式。这种标准形式在逻辑函数的化简以及计算机辅助分析和设计中得到了广泛的应用。

例如,给定逻辑函数为

$$Y = ABC' + BC$$

则可化为

$$Y = ABC' + (A + A')BC = ABC' + ABC + A'BC = m_3 + m_6 + m_7$$

或写作

$$Y(A,B,C) = \sum m(3,6,7)$$

**【例 2.5.6】** 将逻辑函数 $Y = AB'C'D + A'CD + AC$ 展开为最小项之和的形式。

**解:**
$$\begin{aligned} Y &= AB'C'D + A'(B + B')CD + A(B + B')C \\ &= AB'C'D + A'BCD + A'B'CD + ABC(D + D') + AB'C(D + D') \\ &= AB'C'D + A'BCD + A'B'CD + ABCD + ABCD' + AB'CD + AB'CD' \end{aligned}$$

或写作

$$Y(A,B,C,D) = \sum m(3,7,9,10,11,14,15)$$

*** 三、逻辑函数的最大项之积形式**

利用逻辑代数的基本公式和定理,首先我们一定能把任何一个逻辑函数式化成若干多项式相乘的**或与**形式(也称“和之积”形式)。然后再利用基本公式 $AA' = 0$ 将每个多项式中缺少的变量补齐,就可以将函数式的**或与**形式化成最大项之积的形式了。

**【例 2.5.7】** 将逻辑函数 $Y = A'B + AC$ 化为最大项之积的形式。

**解：** 首先可以利用基本公式 $A+BC=(A+B)(A+C)$ 将 $Y$ 化成**或与**形式

$$\begin{aligned} Y &= A'B+AC \\ &= (A'B+A)(A'B+C) \\ &= (A+B)(A'+C)(B+C) \end{aligned}$$

然后在第一个括号内加入一项 $CC'$，在第二个括号内加入 $BB'$，在第三个括号内加入 $AA'$，于是得到

$$\begin{aligned} Y &= (A+B+CC')(A'+BB'+C)(AA'+B+C) \\ &= (A+B+C)(A+B+C')(A'+B+C)(A'+B'+C) \end{aligned}$$

或写作

$$Y(A,B,C,D)=\prod M(0,1,4,6)$$

### 2.5.4 逻辑函数形式的变换

我们在上一小节中已经讲过，可以通过运算将给定的**与或**形式逻辑函数式变换为最小项之和的形式或最大项之积的形式。

此外，在用电子器件组成实际的逻辑电路时，由于选用不同逻辑功能类型的器件，还必须将逻辑函数式变换成相应的形式。

例如，我们想用门电路实现如下的逻辑函数

$$Y=AC+BC' \tag{2.5.3}$$

按照上式的形式，需要用两个具有**与**运算功能的**与**门电路和一个具有**或**运算功能的**或**门电路，才能产生函数 $Y$。

如果受到器件供货的限制，只能全部用**与非**门实现这个电路，这时就需要将式(2.5.3)的**与或**形式变换成全部由**与非**运算组成的**与非－与非**形式。为此，可用摩根定理将式(2.5.3)变换为

$$\begin{aligned} Y &= ((AC+BC')')' \\ &= ((AC)'(BC')')' \end{aligned} \tag{2.5.4}$$

如果要求用具有**与或非**功能的门电路实现式(2.5.3)的逻辑函数，则需要将式(2.5.3)化为**与或非**形式的运算式。根据逻辑代数的基本公式 $A+A'=1$ 和代入定理可知，任何一个逻辑函数 $Y$ 都遵守公式 $Y+Y'=1$。又因为全部最小项之和恒等于1，所以不包含在 $Y$ 中的那些最小项之和就是 $Y'$。将这些最小项之和再求反，也得到 $Y$，而且是**与或非**形式的逻辑函数式。

**【例 2.5.8】** 将逻辑函数 $Y=AC+BC'$ 化为**与或非**形式。

**解：** 首先将 $Y$ 展开为最小项之和的形式，得到

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

或写作

$$Y(A,B,C) = \sum m(2,5,6,7)$$

将 $Y$ 式中不包含的最小项相加，即得

$$Y'(A,B,C) = \sum m(0,1,3,4)$$

将上式求反，就得到了 $Y$ 的**与或非**式

$$\begin{aligned} Y &= (Y')' = (m_0 + m_1 + m_3 + m_4)' = (A'B'C' + AB'C' + A'B'C + A'BC)' \\ &= (B'C' + A'C)' \end{aligned} \tag{2.5.5}$$

如果要求全部用**或非**门电路实现逻辑函数，则应将逻辑函数式化成全部由**或非**运算组成的形式，即**或非－或非**形式。这时可以先将逻辑函数式化为**与或非**的形式，然后再利用反演定理将其中的每个乘积项化为**或非**形式，这样就得到了**或非－或非**式。例如，我们已经得到了式(2.5.5)的**与或非**式，则可按上述方法将它变换为**或非－或非**形式

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

## 复习思考题

R2.5.1 逻辑函数的表示方法有哪几种？你能把由任何一种表示方法给出的逻辑函数转换为由其他任何一种表示方法表示的逻辑函数吗？

R2.5.2 在逻辑函数的真值表和波形图中，任意改变各组输入和输出取值的排列顺序对函数有无影响？

# 2.6 逻辑函数的化简方法

## 2.6.1 公式化简法

在进行逻辑运算时常常会看到，同一个逻辑函数可以写成不同的逻辑式，而这些逻辑式的繁简程度又相差甚远。逻辑式越是简单，它所表示的逻辑关系越明显，同时也有利于用最少的电子器件实现这个逻辑函数。因此，经常需要通过化简的手段找出逻辑函数的最简形式。

例如，有两个逻辑函数

$$Y = ABC + B'C + ACD \tag{2.6.1}$$

$$Y = AC + B'C \tag{2.6.2}$$

将它们的真值表分别列出后即可见到，它们是同一个逻辑函数。显然，下式比上式简单得多。

在**与或**逻辑函数式中，若其中包含的乘积项已经最少，而且每个乘积项里的因子也不能再减少时，则称此逻辑函数式为**最简形式**。对**与或**逻辑式最简形式的定义对其他形式的逻辑式同样也适用，即函数式中相加的乘积项不能再减少，而且每项中相乘的因子不能再减少时，则函数式为最简形式。

化简逻辑函数的目的就是要消去多余的乘积项和每个乘积项中多余的因子，以得到逻辑函数式的最简形式。常用的化简方法有公式化简法、卡诺图化简法以及适用于编制计算机辅助分析程序的 Q－M 法等。

公式化简法的原理就是反复使用逻辑代数的基本公式和常用公式消去函数式中多余的乘积项和多余的因子，以求得函数式的最简形式。

公式化简法没有固定的步骤。现将经常使用的方法归纳如下。

### 一、并项法

利用表 2.3.3 中的公式 $AB + AB' = A$ 可以将两项合并为一项，并消去 $B$ 和 $B'$这一对因子。而且，根据代入定理可知，$A$ 和 $B$ 均可以是任何复杂的逻辑式。

**【例 2.6.1】** 试用并项法化简下列逻辑函数

$$Y_1 = A(B'CD)' + AB'CD$$

$$Y_2 = AB' + ACD + A'B' + A'CD$$

$$Y_3 = A'BC' + AC' + B'C'$$

$$Y_4 = BC'D + BCD' + BC'D' + BCD$$

**解：**

$$Y_1 = A((B'CD)' + B'CD) = A$$

$$Y_2 = A(B' + CD) + A'(B' + CD) = B' + CD$$

$$Y_3 = A'BC' + (A + B')C' = (A'B)C' + (A'B)'C' = C'$$

$$Y_4 = B(C'D + CD') + B(C'D' + CD)$$

$$= B(C \oplus D) + B(C \oplus D)' = B$$

### 二、吸收法

利用表 2.3.3 中的公式 $A + AB = A$ 可将 $AB$ 项消去。$A$ 和 $B$ 同样也可以是任何一个复杂的逻辑式。

**【例 2.6.2】** 试用吸收法化简下列逻辑函数

$$Y_1 = ((A'B)' + C)ABD + AD$$

$$Y_2 = AB + ABC' + ABD + AB(C' + D')$$

$$Y_3 = A + (A'(BC)')'(A' + (B'C' + D)') + BC$$

**解：**

$$Y_1 = ((A'B)' + C)B \cdot AD + AD = AD$$

$$Y_2 = AB + AB(C' + D + (C' + D')) = AB$$

$$Y_3 = (A + BC) + (A + BC)(A' + (B'C' + D)') = A + BC$$

### 三、消项法

利用表2.3.3中的公式 $AB + A'C + BC = AB + A'C$ 及 $AB + A'C + BCD = AB + A'C$ 将 $BC$ 或 $BCD$ 项消去。其中 $A$、$B$、$C$、$D$ 均可以是任何复杂的逻辑式。

**【例 2.6.3】** 用消项法化简下列逻辑函数

$$Y_1 = AC + AB' + (B + C)'$$

$$Y_2 = AB'CD' + (AB')'E + A'CD'E$$

$$Y_3 = A'B'C + ABC + A'BD' + AB'D' + A'BCD' + BCD'E'$$

**解：**

$$Y_1 = AC + AB' + B'C' = AC + B'C'$$

$$Y_2 = (AB')CD' + (AB')'E + (CD')(E)A'$$
$$= AB'CD' + (AB')'E$$

$$Y_3 = (A'B' + AB)C + (A'B + AB')D' + BCD'(A' + E')$$
$$= (A \oplus B)'C + (A \oplus B)D' + CD'(B(A' + E'))$$
$$= (A \oplus B)'C + (A \oplus B)D'$$

### 四、消因子法

利用表2.3.3中的公式 $A + A'B = A + B$ 可将 $A'B$ 中的 $A'$ 消去。$A$、$B$ 均可以是任何复杂的逻辑式。

**【例 2.6.4】** 试利用消因子法化简下列逻辑函数

$$Y_1 = B' + ABC$$

$$Y_2 = AB' + B + A'B$$

$$Y_3 = AC + A'D + C'D$$

**解：**

$$Y_1 = B' + ABC = B' + AC$$

$$Y_2 = AB' + B + A'B = A + B + A'B = A + B$$

$$Y_3 = AC + A'D + C'D = AC + (A' + C')D = AC + (AC)'D$$
$$= AC + D$$

### 五、配项法

① 根据基本公式中的 $A + A = A$ 可以在逻辑函数式中重复写入某一项，有时能获得更加简单的化简结果。

**【例 2.6.5】** 试化简逻辑函数 $Y = A'BC' + A'BC + ABC$。

**解：** 若在式中重复写入 $A'BC$，则可得到

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

② 根据基本公式中的 $A + A' = 1$ 可以在函数式中的某一项上乘以 $(A + A')$，然后拆成两项分别与其他项合并，有时能得到更加简单的化简结果。

**【例 2.6.6】** 试化简逻辑函数 $Y = AB' + A'B + BC' + B'C$。

**解：** 利用配项法可将 $Y$ 写成

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

在化简复杂的逻辑函数时，往往需要灵活、交替地综合运用上述方法，才能得到最后的化简结果。

**【例 2.6.7】** 化简逻辑函数

$$Y = AC + B'C + BD' + CD' + A(B + C') + A'BCD' + AB'DE$$

**解：**

$$Y = AC + B'C + BD' + \underbrace{CD'} + A(B + C') + \underbrace{A'BCD'} + AB'DE$$

↓（根据 $A + AB = A$，消去 $A'BCD'$）

$$= AC + \underbrace{B'C} + BD' + CD' + A\ \underbrace{(B'C)'} + AB'DE$$

↓（根据 $A + A'B = A + B$，消去 $A\ (B'C)'$ 中的 $(B'C)'$ 因子）

$$= \underbrace{AC} + B'C + BD' + CD' + \underbrace{A + AB'DE}$$

↓（根据 $A + AB = A$，消去 $AC$ 和 $AB'DE$）

$$= A + \underbrace{B'C} + \underbrace{BD'} + \underbrace{CD'}$$

↓（根据 $AB + A'C + BC = AB + A'C$，消去 $CD'$）

$$= A + B'C + BD'$$

## 2.6.2　卡诺图化简法

### 一、逻辑函数的卡诺图表示法

将 $n$ 变量的全部最小项各用一个小方块表示，并使具有逻辑相邻性的最小项在几何位置上也相邻地排列起来，所得到的图形称为 $n$ 变量最小项的卡诺图。因为这种表示方法是由美国工程师卡诺（M. Karnaugh）首先提出的，所以将这种

图形称为卡诺图(Karnaugh Map)。

图 2.6.1 中画出了二到五变量最小项的卡诺图。图形两侧标注的 0 和 1 表示使对应小方格内的最小项为 1 的变量取值。同时,这些 0 和 1 组成的二进制数所对应的十进制数大小也就是对应的最小项的编号。

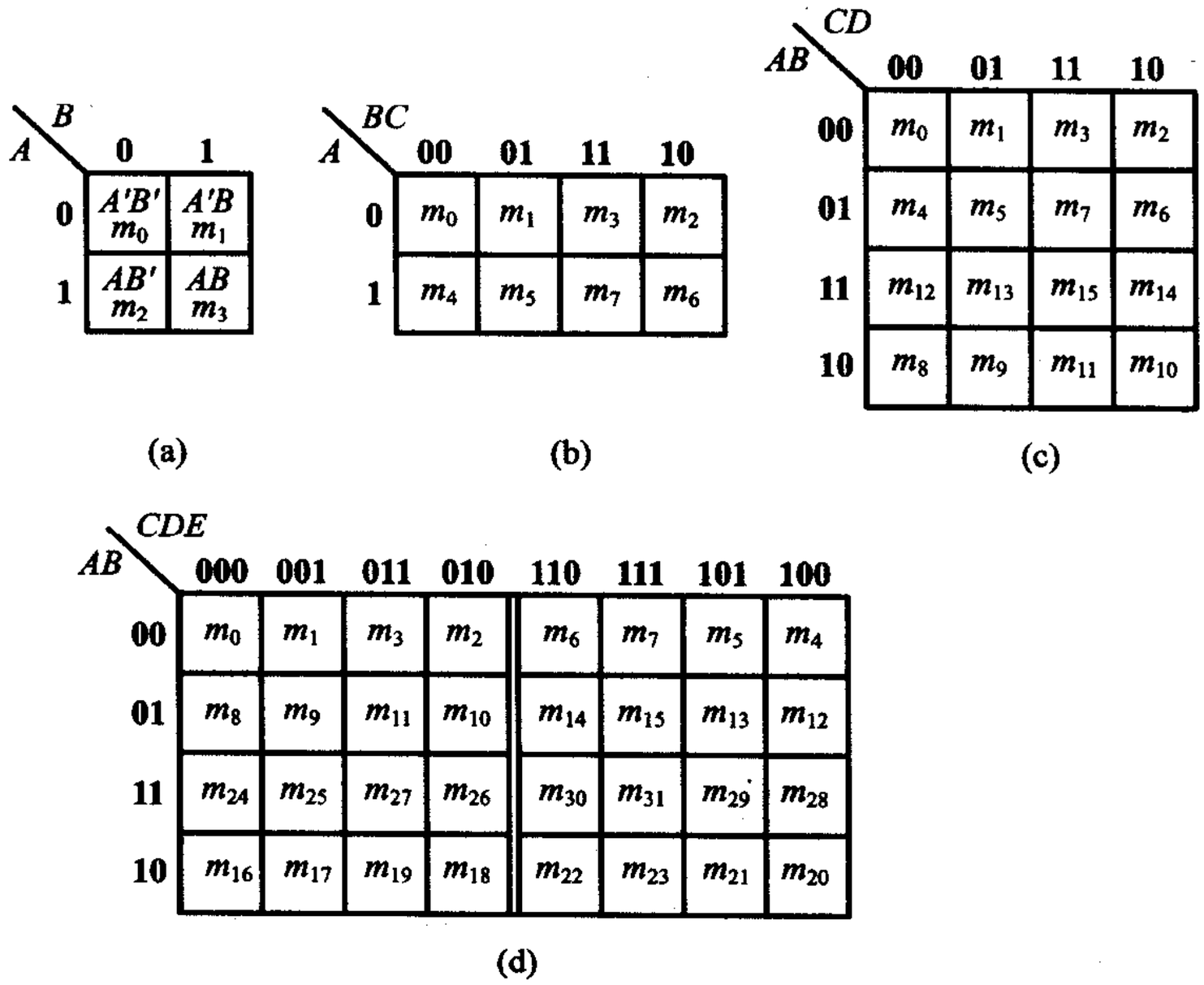


图 2.6.1 二到五变量最小项的卡诺图

(a) 两变量($A$、$B$)最小项的卡诺图 (b) 三变量($A$、$B$、$C$)最小项的卡诺图

(c) 四变量($A$、$B$、$C$、$D$)最小项的卡诺图 (d) 五变量($A$、$B$、$C$、$D$、$E$)最小项的卡诺图

为了保证图中几何位置相邻的最小项在逻辑上也具有相邻性,这些数码不能按自然二进制数从小到大地顺序排列,而必须按图中的方式排列,以确保相邻的两个最小项仅有一个变量是不同的。

从图 2.6.1 所示的卡诺图上还可以看到,处在任何一行或一列两端的最小项也仅有一个变量不同,所以它们也具有逻辑相邻性。因此,从几何位置上应当将卡诺图看成是上下、左右闭合的图形。

在变量数大于、等于五以后,仅仅用几何图形在两维空间的相邻性来表示逻辑相邻性已经不够了。例如,在图 2.6.1(d)所示的五变量最小项的卡诺图中,除了几何位置相邻的最小项具有逻辑相邻性以外,以图中双竖线为轴左右对称位置上的两个最小项也具有逻辑相邻性。

既然任何一个逻辑函数都能表示为若干最小项之和的形式,那么自然也就