

2. 逻辑代数与硬件描述语言基础

2.1 逻辑代数的基本定理和恒等式

2.2 逻辑函数表达式的形式

2.3 逻辑函数的代数化简法

2.4 逻辑函数的卡诺图化简法

2.5* 硬件描述语言Verilog HDL基础

教学基本要求

- 1、熟悉逻辑代数常用基本定律、恒等式和规则。
- 2、掌握逻辑代数的表示方法；
- 3、掌握逻辑代数的变换和卡诺图化简法；
- 4、熟悉硬件描述语言 **Verilog HDL**

2.1 逻辑代数的基本定理和规则

2.1.1 逻辑代数的基本定律和恒等式

2.1.2 逻辑代数的基本规则

2.1 逻辑代数的基本定理和规则

逻辑代数又称布尔代数。它是分析和设计现代数字逻辑电路不可缺少的数学工具。逻辑代数有一系列的定律、定理和规则，用于对表达式进行处理，以完成对逻辑电路的化简、变换、分析和设计。

逻辑关系指的是事件产生的条件和结果之间的因果关系。在数字电路中往往是将事情的条件作为输入信号，而结果用输出信号表示。条件和结果的两种对立状态分别用逻辑“1”和“0”表示。

2.1.1 逻辑代数的基本定律和恒等式

1、基本公式

$$0、1律: A + 0 = A \quad A + 1 = 1 \quad A \cdot 1 = A \quad A \cdot 0 = 0$$

$$互补律: A + \overline{A} = 1 \quad A \cdot \overline{A} = 0$$

$$交换律: A + B = B + A \quad A \cdot B = B \cdot A$$

$$结合律: A + B + C = (A + B) + C \quad A \cdot B \cdot C = (A \cdot B) \cdot C$$

$$分配律: A (B + C) = AB + AC \quad A + BC = (A + B)(A + C)$$

重叠律:

$$A + A = A$$

$$A \cdot A = A$$

反演律(摩根定理): $\overline{A + B} = \overline{A} \cdot \overline{B}$ $\overline{AB} = \overline{A} + \overline{B}$

吸收律

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

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

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

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

其它常用恒等式

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

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

2、基本公式的证明 (真值表证明法)

例 证明 $A + \bar{A} \cdot B = A + B$

列出等式、右边的函数值的真值表

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

例：试化简下列逻辑函数 $L=(A+B)(\bar{A}+B)$

$$L = A\bar{A} + AB + B\bar{A} + BB \text{ (分配律)}$$

$$= 0 + AB + B\bar{A} + B \quad (A \cdot \bar{A} = 0, A \cdot A = A)$$

$$= AB + B\bar{A} + B \quad (A + 0 = A)$$

$$= B(A + \bar{A} + 1) \quad [AB + AC = A(B + C)]$$

$$= B \cdot 1 = B \quad (A + 1 = 1, A \cdot 1 = A)$$

2.1.2 逻辑代数的基本规则

1. 代入规则：在包含变量A逻辑等式中，如果用另一个函数式代入式中所有A的位置，则等式仍然成立。这一规则称为代入规则。

例： $B(A + C) = BA + BC$,

用 $A + D$ 代替 A ，得

$$B[(A + D) + C] = B(A + D) + BC = BA + BD + BC$$

代入规则可以扩展所有基本公式或定律的应用范围

2. 反演规则:

对于任意一个逻辑表达式 L ，若将其中所有的与（ \cdot ）换成或（ $+$ ），或（ $+$ ）换成与（ \cdot ）；原变量换为反变量，反变量换为原变量；将1换成0，0换成1；则得到的结果就是原函数的反函数。

例2.1.1 试求 $L = \overline{A}\overline{B} + CD + 0$ 的非函数

解：按照反演规则，得

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

3. 对偶规则:

对于任何逻辑函数式，若将其中的与（ \cdot ）换成或（ $+$ ），或（ $+$ ）换成与（ \cdot ）；并将1换成0，0换成1；那么，所得的新的函数式就是 L 的对偶式，记作 L' 。

例: 逻辑函数 $L = (A + \bar{B})(A + C)$ 的对偶式为

$$L' = A\bar{B} + AC$$

当某个逻辑恒等式成立时，则该恒等式两侧的对偶式也相等。这就是对偶规则。利用对偶规则，可从已知公式中得到更多的运算公式，例如，吸收律

2.2 逻辑函数表达式的形式

2.2.1 逻辑函数表达式的形式

2.2.2 最小项与最小项表达式

2.2.3 最大项与最大项表达式

2.2 逻辑函数表达式的形式

2.2.1 逻辑函数表达式的基本形式

1、与-或表达式

若干与项进行或逻辑运算构成的表达式。由与运算符和或运算符连接起来。

$$L = A \cdot C + \bar{C} \cdot D$$

2、或-与表达式

若干或项进行与逻辑运算构成的表达式。由或运算符和与运算符连接起来。

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

通常表达式为混合形式 $L = A \cdot (B \cdot C + \bar{B} \cdot \bar{C}) + A \cdot (B \cdot \bar{C} + \bar{B} \cdot C)$

经过变换可转换为上述两种基本形式

2.2.2 最小项与最小项表达式

1. 最小项的定义和性质

n 个变量 X_1, X_2, \dots, X_n 的最小项是 n 个因子的乘积，每个变量都以它的原变量或非变量的形式在乘积项中出现，且仅出现一次。一般 n 个变量的最小项应有 2^n 个。

例如， A 、 B 、 C 三个逻辑变量的最小项有（ $2^3=$ ）8个，即

$$\overline{A}\overline{B}\overline{C}、\overline{A}\overline{B}C、\overline{A}B\overline{C}、\overline{A}BC、A\overline{B}\overline{C}、A\overline{B}C、AB\overline{C}、ABC$$

$\overline{A}B$ 、 $A\overline{B}C\overline{A}$ 、 $A(B+C)$ 等则不是最小项。

2、最小项的性质 三个变量的所有最小项的真值表

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

- 对于任意一个最小项，只有一组变量取值使得它的值为1；
- 任意两个最小项的乘积为0；
- 全体最小项之和为1。

3、最小项的编号

三个变量的所有最小项的真值表

			m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
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

最小项的表示：通常用 m_i 表示最小项， m 表示最小项，下标 i 为最小项号。

最小项的性质及编号

- 在变量的任意取值组合下，仅有一个最小项的值为1,其余的全部为0，即等于1的机会“最小”；任意两个不同的最小项的积为0;全部最小项的和为1
- 为了书写方便，常用 m_i 表示 n 变量的最小项, $i \in (0,1,2,\dots,2^n-1)$ ，若将使 m_i 的值为1的变量取值当成一个二进制数,这个二进制数所对应的十进制数，即为 i 的取值

2. 最小项表达式

由若干最小项相或构成的表达式，也称为标准与-或式。

- 为“与或”逻辑表达式；
- 在“与或”式中的每个乘积项都是最小项。

例1 将 $L(A, B, C) = AB + \bar{A}C$ 化成最小项表达式

$$\begin{aligned} L(A, B, C) &= AB(C + \bar{C}) + \bar{A}(B + \bar{B})C \\ &= ABC + AB\bar{C} + \bar{A}BC + A\bar{B}C \\ &= m_7 + m_6 + m_3 + m_5 \\ &= \sum m(7, 6, 3, 5) \end{aligned}$$

例2 将 $L(A, B, C) = \overline{(AB + \bar{A}\bar{B} + \bar{C})\bar{A}\bar{B}}$ 化成最小项表达式

a. 去掉非号
$$\begin{aligned} L(A, B, C) &= \overline{(AB + \bar{A}\bar{B} + \bar{C})} + AB \\ &= (\overline{AB} \cdot \overline{\bar{A}\bar{B}} \cdot \overline{\bar{C}}) + AB \\ &= (\bar{A} + \bar{B})(A + B)C + AB \end{aligned}$$

b. 去括号

$$\begin{aligned} &= \bar{A}BC + A\bar{B}C + AB \\ &= \bar{A}BC + A\bar{B}C + AB(C + \bar{C}) \\ &= \bar{A}BC + A\bar{B}C + ABC + AB\bar{C} \\ &= m_3 + m_5 + m_7 + m_6 = \sum m(3, 5, 6, 7) \end{aligned}$$

最小项表达式

- 全部由最小项相加构成的“与—或”表达式，即最小项表达式。又称标准与—或表达式，标准积之和表达式。
- 一般形式是

$$Y = \sum_{i=0}^{2^n-1} a_i \cdot m_i$$

最小项和最小项表达式

- 对于任何一个逻辑函数，都可以写出它的标准与—或表达式。
- 对于逻辑函数的任何非标准与—或表达式，可用吸收律公式

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

- 将乘积项中所缺少的变量补齐，从而变换成标准与—或表达式

2.2.2 最大项与最大项表达式

1. 最大项的定义和性质

n 个变量 X_1, X_2, \dots, X_n 的最大项是 n 个因子或相，每个变量都以它的原变量或非变量的形式在或项中出现，且仅出现一次。一般 n 个变量的最大项应有 2^n 个。

例如， A 、 B 、 C 三个逻辑变量的最大项有（ $2^3=$ ）8个，即

$$\begin{aligned} &(\bar{A} + \bar{B} + \bar{C}), (\bar{A} + \bar{B} + C), (\bar{A} + B + \bar{C}), (\bar{A} + B + C), \\ &(A + \bar{B} + \bar{C}), (A + \bar{B} + C), (A + B + \bar{C}), (A + B + C) \end{aligned}$$

1. 最大项的定义和性质

最大项的表示：通常用 M_i 表示最大项， M 表示最大项，下标 i 为最大项号。

最大项的性质：

- 对于任意一个最大项，只有一组变量取值使得它的值为0；
- 任意两个最大项之和为1；
- 全体最大项之积为0。

2. 最小项和最大项的关系

两者之间为互补关系： $m_i = \overline{M_i}$ ，或者 $M_i = \overline{m_i}$

例：逻辑电路的真值表如右，写出最小项和最大项表达式。

最小项表达式：

将 $L=1$ 的各个最小项相加

$$L(A, B, C) = m_3 + m_5 + m_6$$

$$= \sum m(3, 5, 6)$$

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

A	B	C	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	$1 \rightarrow m_3$
1	0	0	0
1	0	1	$1 \rightarrow m_5$
1	1	0	$1 \rightarrow m_6$
1	1	1	0

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

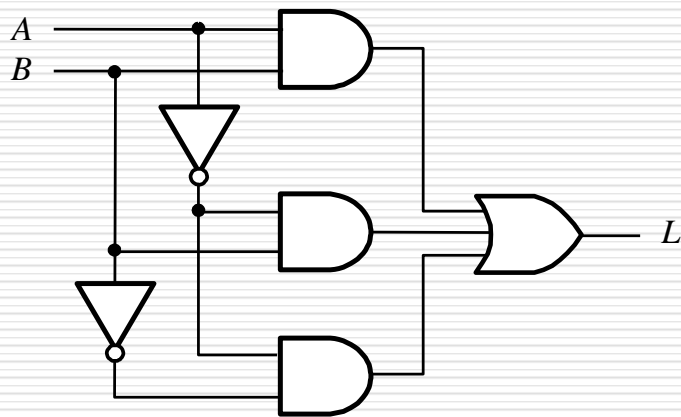
2.3 逻辑函数的代数化简法

2.3.1 逻辑函数的最简形式

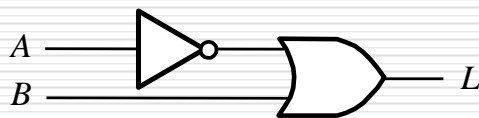
2.3.2 逻辑函数的代数化简法

2.3 逻辑函数的代数法化简

化简的目的：降低电路实现的成本，以较少的门实现电路。



(a) 标准积之和的电路



(b) 成本最低的电路

图 (a) 和图 (b) 的电路逻辑功能相同，但图 (b) 电路简单可靠性高，成本低。

2.3.1 逻辑函数的最简形式

逻辑函数有不同形式，如与-或表达式、与非-与非表达式、或-与表达式、或非-或非表达式以及与-或-非表达式等。

将其中包含的与项数最少，且每个与项中变量数最少的与-或表达式称为最简与-或表达式。

$$L = AC + \bar{C}D$$

$$= \overline{\overline{A} \overline{C}} \cdot \overline{\overline{C} \overline{D}}$$

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

$$= \overline{\overline{(A + \bar{C})} + \overline{(C + D)}}$$

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

“与-或” 表达式

“与非-与非” 表达式

“或-与” 表达式

“或非-或非” 表达式

“与-或-非” 表达式

2.3.2 逻辑函数的代数化简法

1、逻辑函数的化简

化简的主要方法：

1. 公式法（代数法）
2. 图解法（卡诺图法）

代数化简法：

运用逻辑代数的基本定律和恒等式进行化简的方法。

并项法： $A + \bar{A} = 1$

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

吸收法: $A + AB = A$

$$L = \underline{\bar{A}B} + \underline{\bar{A}BCD}(E + F) = \bar{A}B$$

消去法: $A + \bar{A}B = A + B$

$$L = AB + \underline{\bar{A}C} + \underline{\bar{B}C} = AB + (\bar{A} + \bar{B})C \quad \boxed{\bar{A} + \bar{B} = \overline{AB}}$$

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

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

配项法: $A + \bar{A} = 1$

$$L = AB + \bar{A}\bar{C} + \underline{BC} = AB + \bar{A}\bar{C} + \underline{(A + \bar{A})BC}$$

$$= \underline{AB} + \underline{\bar{A}\bar{C}} + \underline{ABC} + \underline{\bar{A}BC}$$

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

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

2、逻辑函数形式的变化

通常在一片集成电路芯片中只有一种门电路，为了减少门电路的种类，需要对逻辑函数表达式进行变换。

例：已知 $L = AB\bar{D} + \bar{A}\bar{B}\bar{D} + ABD + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD$

(1) 求最简的与-或式，并画出相应的逻辑图；

(2) 画出仅用与非门实现的电路。

解： $L = AB(\bar{D} + D) + \bar{A}\bar{B}\bar{D} + \bar{A}\bar{B}D(\bar{C} + C)$

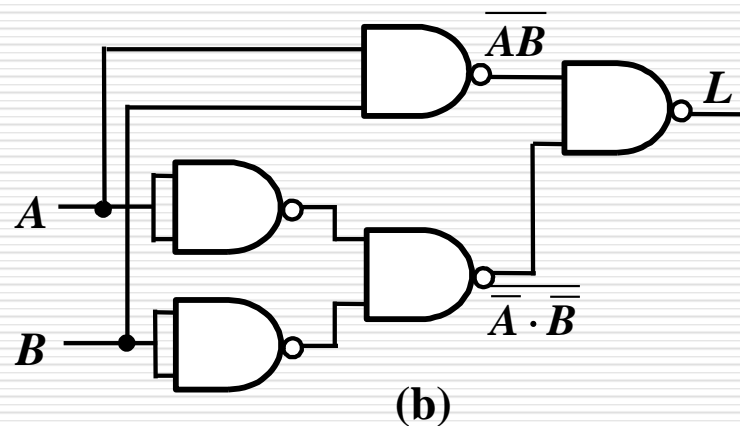
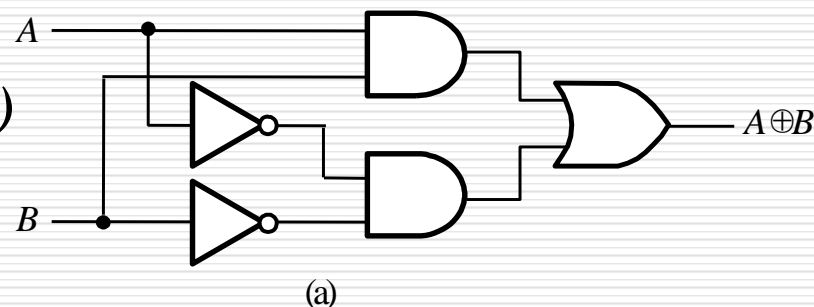
$$= AB + \bar{A}\bar{B}\bar{D} + \bar{A}\bar{B}D$$

$$= AB + \bar{A}\bar{B}(D + \bar{D})$$

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

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

$$= \overline{\overline{AB}} \cdot \overline{\overline{\bar{A}\bar{B}}}$$



2.4 逻辑函数的卡诺图化简法

2.4.1 用卡诺图表示逻辑函数

2.4.2 用卡诺图化简逻辑函数

代数法化简在使用中遇到的困难：

- 1.逻辑代数与普通代数的公式易混淆，化简过程要求对所有公式熟练掌握；
- 2.代数法化简无一套完善的方法可循，它依赖于人的经验和灵活性；
- 3.用这种化简方法技巧强，较难掌握。特别是对代数化简后得到的逻辑表达式是否是最简式判断有一定困难。
卡诺图法可以比较简便地得到最简的逻辑表达式。

2.2.3 用卡诺图表示逻辑函数

1、卡诺图的引出

卡诺图

是一种变形**真值表**。将输入变量分成行列两组，变量取值也分成两组，变量取值按**循环码**规律排列；各变量取值组合对应的逻辑函数值标注在行列两组变量交叉覆盖产生的方格中，它同样具有真值表的**唯一性，完整性和正确性**，但却可以方便地用于逻辑运算和逻辑函数化简。

2.2.3 用卡诺图表示逻辑函数

1、卡诺图的引出

卡诺图：将n变量的全部最小项都用小方块表示，并使具有逻辑相邻的最小项在几何位置上也相邻地排列起来，这样，所得到的图形叫n变量的卡诺图。

逻辑相邻的最小项：如果两个最小项只有一个变量互为反变量，那么，就称这两个最小项在逻辑上相邻。

如最小项 $m_6=ABC\bar{C}$ 、与 $m_7=ABC$ 在逻辑上相邻

m_6	m_7
-------	-------

A	B	C	D	m_i
0	0	0	0	$\bar{A}\bar{B}\bar{C}\bar{D}$
0	0	0	1	$\bar{A}\bar{B}\bar{C}D$
0	0	1	0	$\bar{A}\bar{B}C\bar{D}$
0	0	1	1	$\bar{A}\bar{B}CD$
0	1	0	0	$\bar{A}B\bar{C}\bar{D}$
0	1	0	1	$\bar{A}B\bar{C}D$
0	1	1	0	$\bar{A}BC\bar{D}$
0	1	1	1	$\bar{A}BCD$
1	0	0	0	$A\bar{B}\bar{C}\bar{D}$
1	0	0	1	$A\bar{B}\bar{C}D$
1	0	1	0	$A\bar{B}C\bar{D}$
1	0	1	1	$A\bar{B}CD$
1	1	0	0	$AB\bar{C}\bar{D}$
1	1	0	1	$AB\bar{C}D$
1	1	1	0	$ABC\bar{D}$
1	1	1	1	$ABCD$

真值表变形产生的卡诺图

		A		1	
		0		1	
C	D	B		0	
		0	1	1	0
0	0	m_0	m_4	m_{12}	m_8
	1	m_1	m_5	m_{13}	m_9
1	1	m_3	m_7	m_{15}	m_{11}
	0	m_2	m_6	m_{14}	m_{10}

卡诺图的一般形式

4变量卡诺图

AB		00	01	11	10
CD	00	m_0	m_4	m_{12}	m_8
	01	m_1	m_5	m_{13}	m_9
	11	m_3	m_7	m_{15}	m_{11}
	10	m_2	m_6	m_{14}	m_{10}

2变量卡诺图

A		0	1
B	0	m_0	m_2
	1	m_1	m_3

3变量卡诺图

AB		00	01	11	10
C	0	m_0	m_2	m_6	m_4
	1	m_1	m_3	m_7	m_5

卡诺图一般形式的简化

2变量卡诺图

		A	
		0	1
B	0	0	2
	1	1	3

4变量卡诺图

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

3变量卡诺图

		AB			
		00	01	11	10
C	0	0	2	6	4
	1	1	3	7	5

用卡诺图表示逻辑函数

$$F(A, B) = \Sigma m(0, 3) = \Pi_M(1, 2)$$

$$H(A, B, C, D) = \Sigma m(0, 1, 5, 7, 8, 10, 13, 15) \\ = \Pi_M(2, 3, 4, 6, 9, 11, 12, 14)$$

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

A			
B		0	1
0	0	1 ⁰	0 ²
	1	0 ¹	1 ³

$$G(A, B, C) = \Sigma m(3, 5, 6, 7) \\ = \Pi_M(0, 1, 2, 4)$$

AB					
C		00	01	11	10
0	00	0 ⁰	0 ²	1 ⁶	0 ⁴
	01	0 ¹	1 ³	1 ⁷	1 ⁵

逻辑相邻的概念

如果两个最小项中只有一个变量因子不相同，则称这两个最小项**逻辑相邻**。具有逻辑相邻性的两个最小项可以合并为一个乘积项，这个乘积项由它们的相同部分组成。逻辑相邻的数学基础是吸收律

$$ABC + A\bar{B}C = AC$$

任意一个 n 变量最小项有 n 个相邻的最小项

如果两个乘积项中只有一个变量因子不相同，则称这两个乘积项**逻辑相邻**。具有逻辑相邻性的两个乘积项可以消去这个不相同的变量因子合并为一个变量因子更少的乘积项

用卡诺图合并最小项

可以直观地凭借最小项在卡诺图中的几何位置来确定最小项的逻辑相邻性。再简单地将几何位置相邻的两个最小项方格（简称“1”格）用一个“圈”圈起来，便可产生一个合并项。根据逻辑相邻的概念，可以进一步将“圈”扩大，把4个、8个…… 2^i 个“1”格圈起来得到更为简单的合并项。

用卡诺图合并最小项

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

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

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

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

用卡诺图合并最小项

AB		00	01	11	10
CD	00	1	1		
	01				1
	11		1	1	1
	10	1			1

AB		00	01	11	10
CD	00				1
	01	1	1		1
	11	1	1		1
	10				1

用卡诺图合并最小项

		AB			
		00	01	11	10
CD	00	1			1
	01		1	1	
	11		1	1	
	10	1			1

		AB			
		00	01	11	10
CD	00		1	1	
	01	1			1
	11	1			1
	10		1	1	

用卡诺图合并最小项

		AB			
CD		00	01	11	10
00	1				1
01	1	1	1	1	1
11	1	1	1	1	1
10	1				1

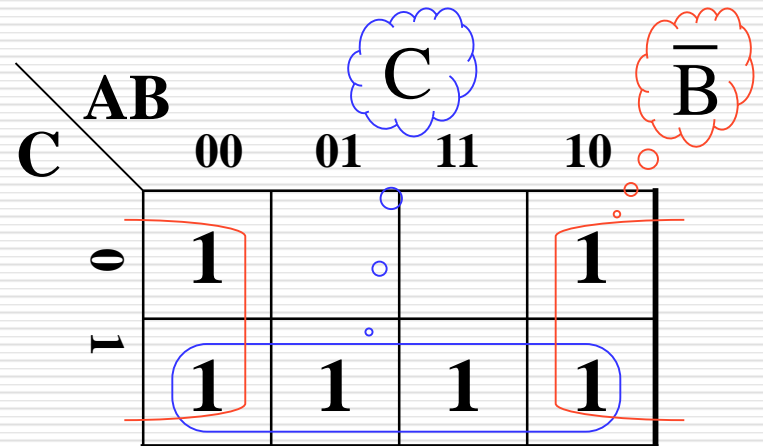
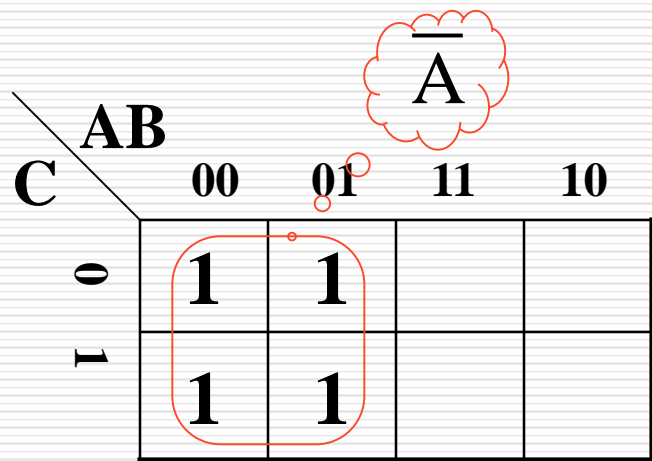
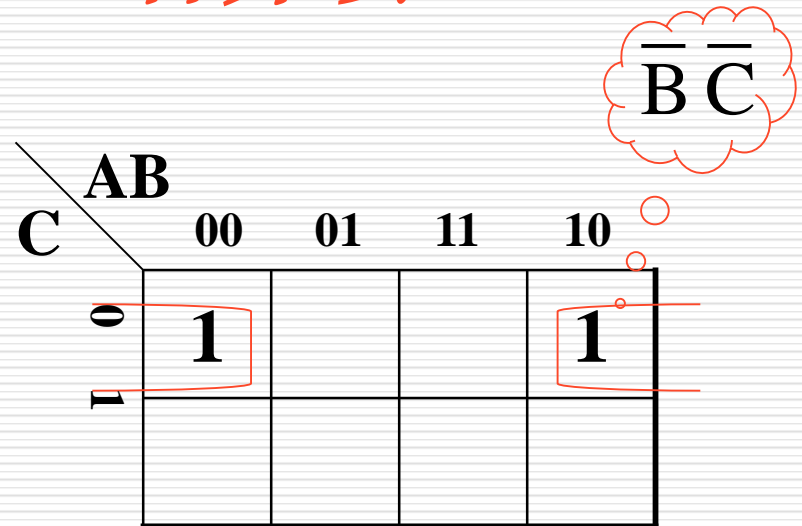
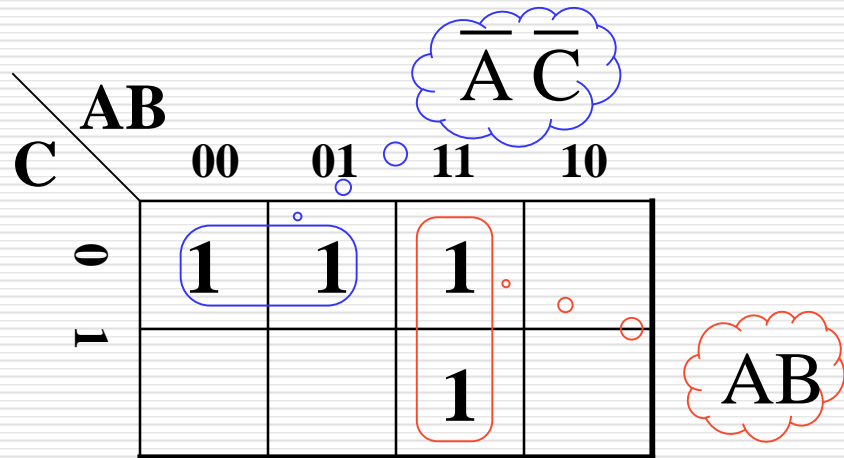
		AB		
CD		00	01	11
00	¹⁰ 1	1	1	1
01			1	1
11			1	1
10	1	1	1	1

用卡诺图合并最小项产生合并项

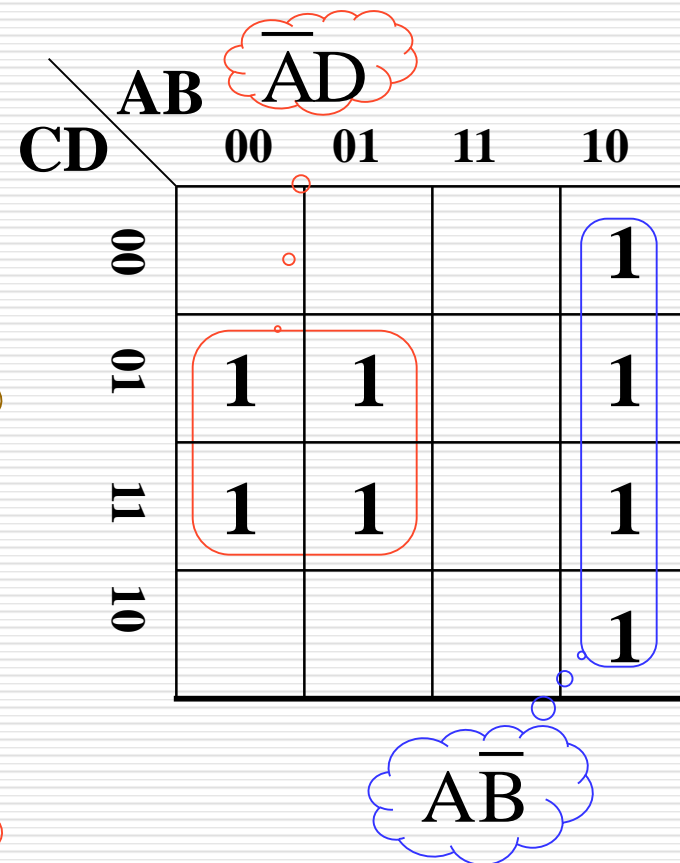
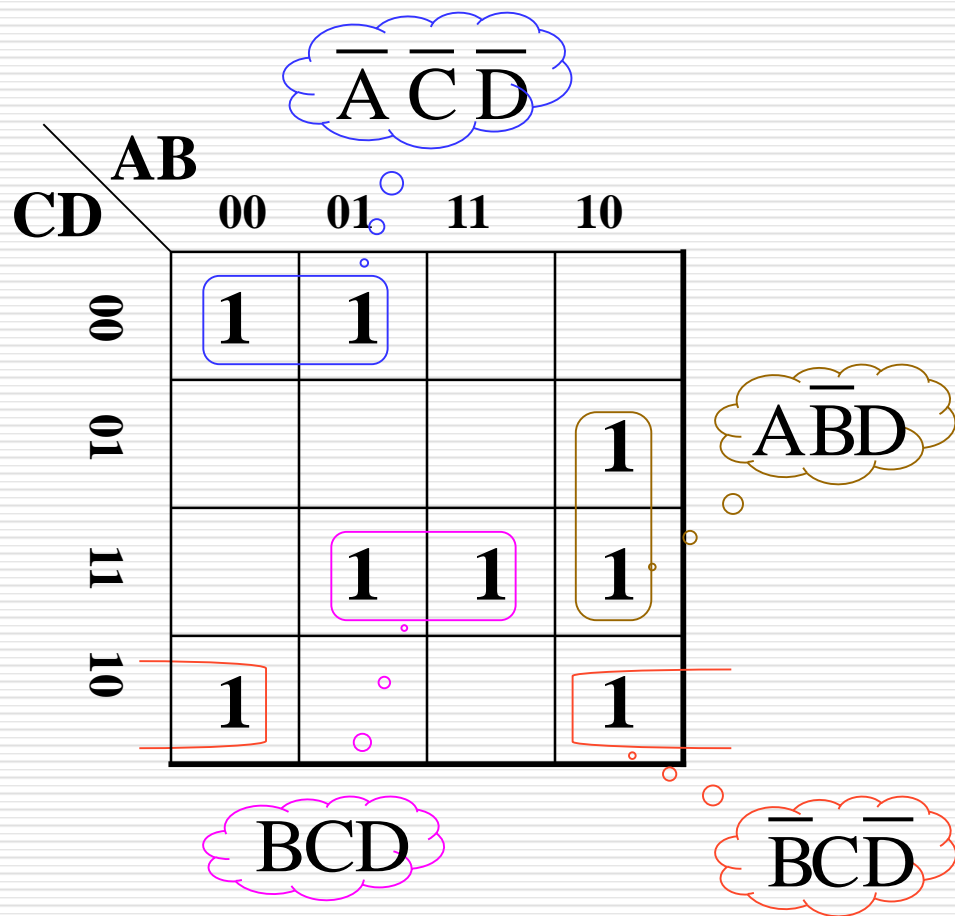
合并项由圈所覆盖的范围内没有发生变化的变量组成。

一个圈内应当、也只能圈入 2^i 个方格，则这个合并项与最小项相比消去了 i 个变量

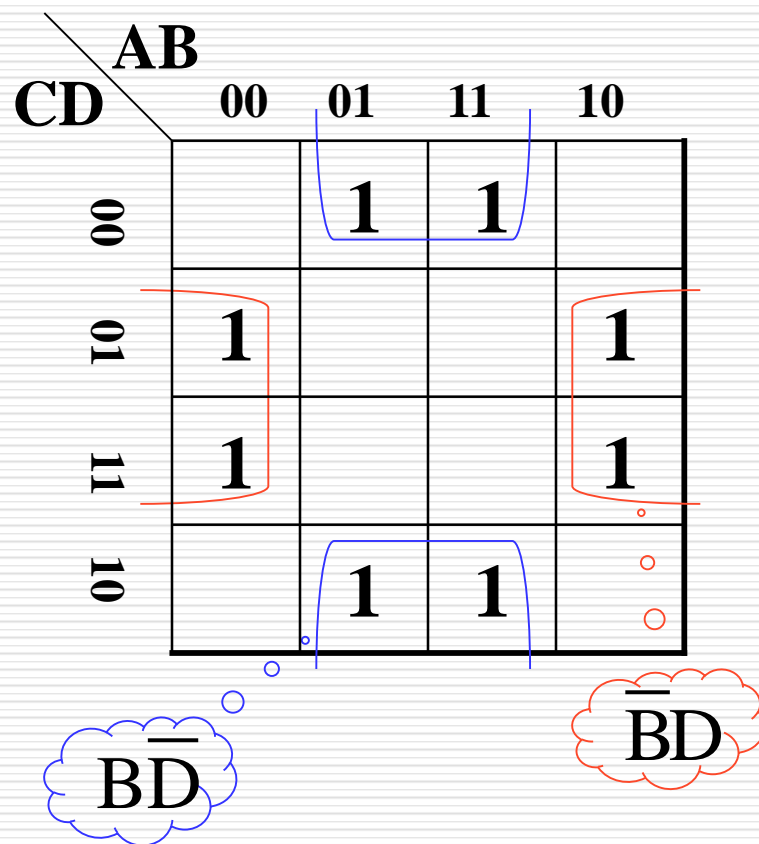
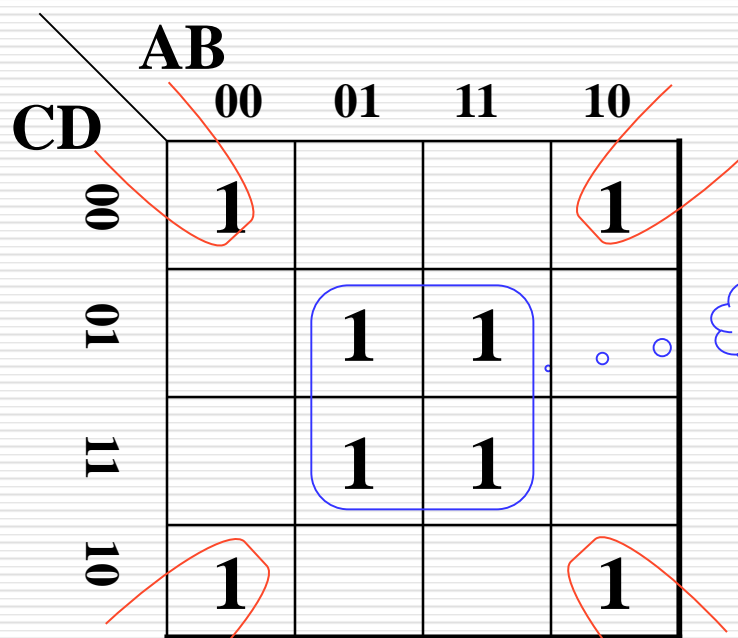
用卡诺图合并最小项产生合并项



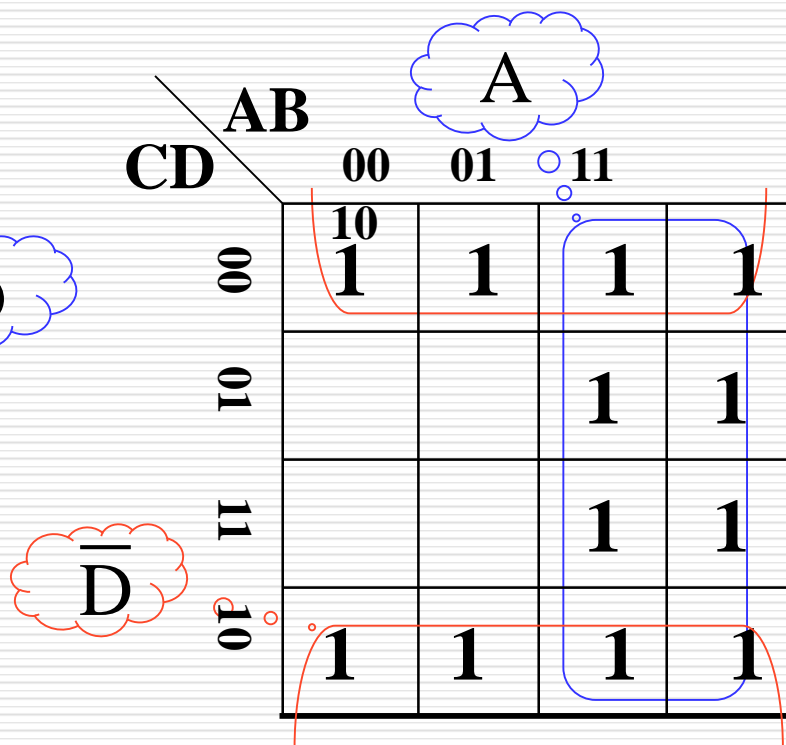
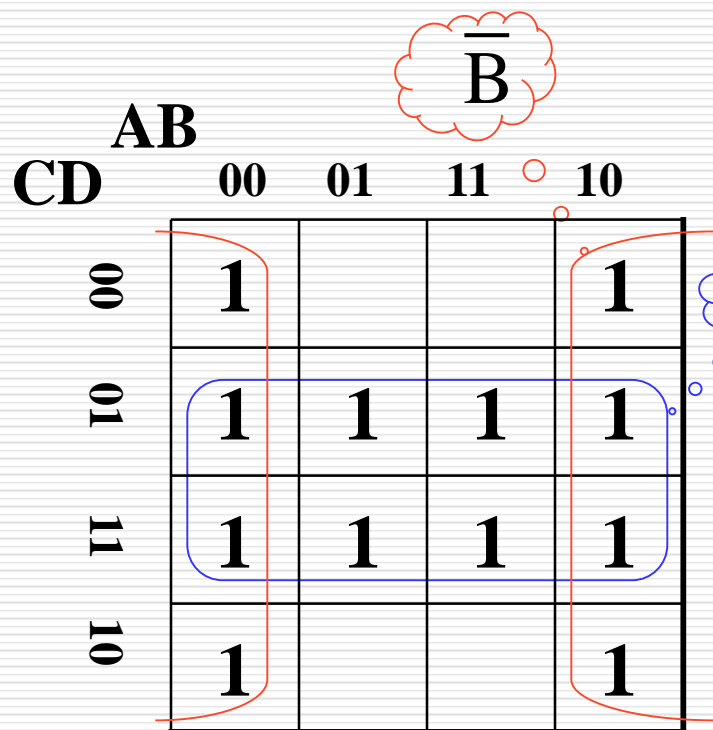
用卡诺图合并最小项产生合并项



用卡诺图合并最小项产生合并项



用卡诺图合并最小项产生合并项



当一个圈覆盖了卡诺图的全部1格时产生的合并项为1

用卡诺图合并最小项产生的合并项

- **主要项** (素项、本原蕴含项)：当合并圈不能再扩大所产生的合并项
- **必要项** (实质素项、实质本原蕴含项)：凡主要项圈中至少有一个“特定”的1格没有被其它主要项所覆盖
- **多余项**：一个主要项圈中如果不包含有“特定”的1格，即该圈内全部1格都被其它圈所覆盖，这个主要项就是多余项

用卡诺图化简逻辑函数

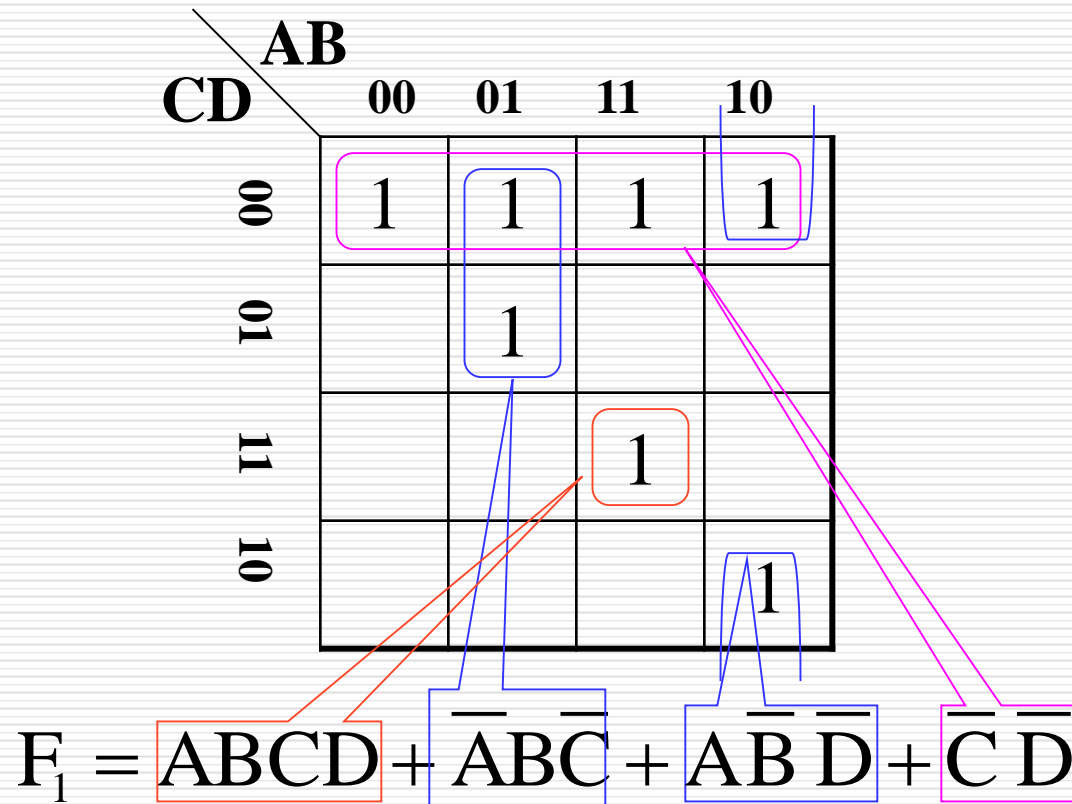
- 用卡诺图化简逻辑函数就是寻找全部必要项的过程，具体地说就是：
- 1. 作图；用卡诺图表示一个确定的逻辑函数
- 2. 圈1；圈出全部1格，合并产生全部必要项
- 3. 将全部必要项相加即可得到最简与—或表达式

关于圈圈的几点说明

- 1. 从没有相邻项的孤立1格圈起，先圈只有一种圈法的主要项，再由少到多逐渐扩大圈的覆盖面，圈要尽可能大，但**必须保证圈内1格数是 2^i 个**，任意1格允许被重复多次圈入；
- 2. 注意不要漏掉边缘和四角的相邻格；
- 3. 同一卡诺图可以有多种圈法，应选择一种合适的圈法，直到所有1格无遗漏地至少被圈一次，且圈的总数尽可能少；
- 4. 也可以圈“0格”化简，得到最简或与表达式；但是，**决不允许在同一卡诺图内既圈“0格”同时又圈“1格”**

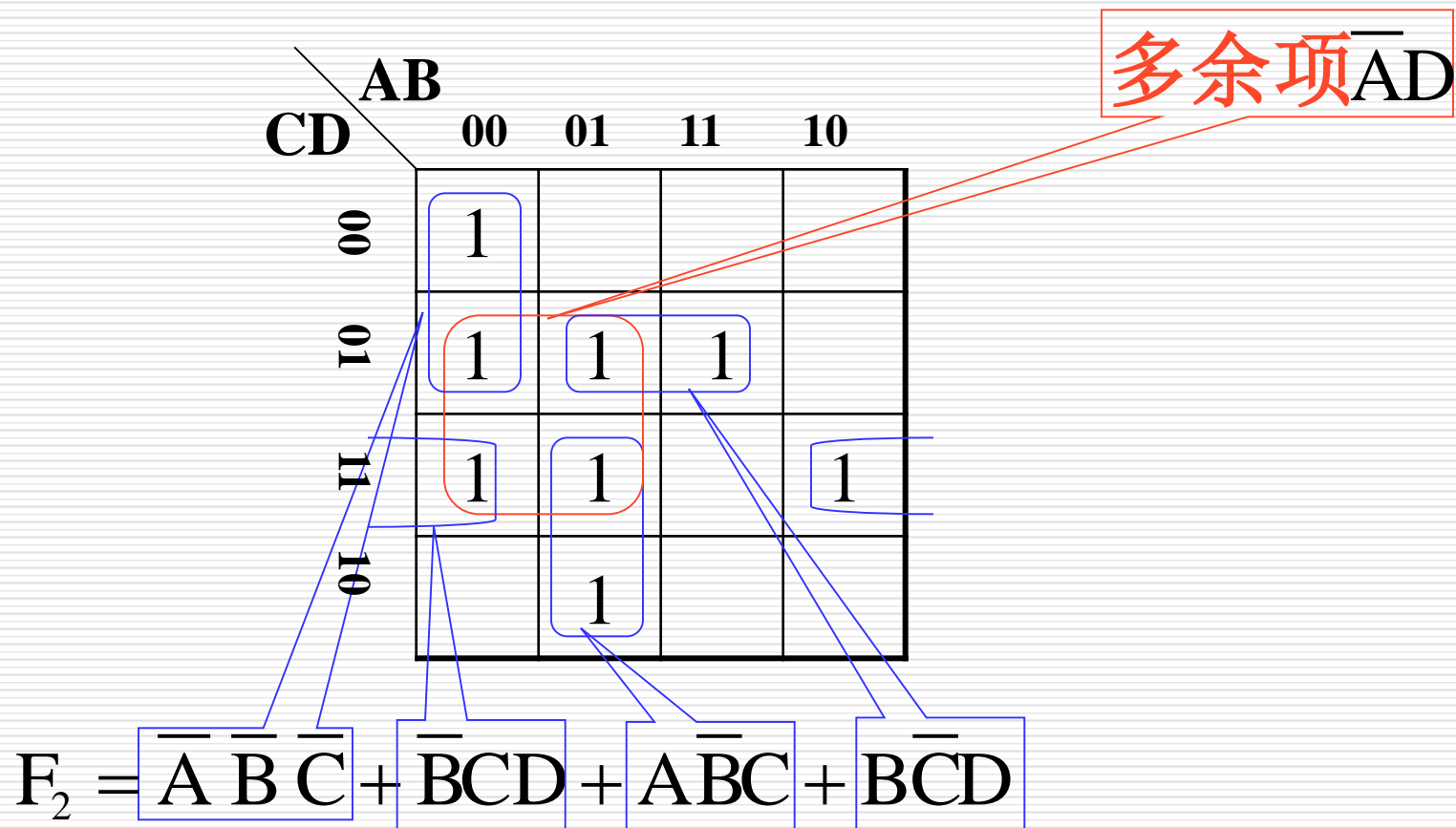
卡诺图化简逻辑函数例1

$$F_1(A, B, C, D) = \Sigma m(0, 4, 5, 8, 10, 12, 15)$$



卡诺图化简逻辑函数例2

$$F_2(A, B, C, D) = \sum m(0, 1, 3, 5, 6, 7, 11, 13)$$



卡诺图化简逻辑函数例3

$$F_3(A, B, C, D) = \Sigma m(0, 2, 4, 5, 10, 11, 13, 15)$$

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

$$F_3 = \bar{A} \bar{C} \bar{D} + B \bar{C} D + A C D + \bar{B} C \bar{D}$$

$$F_3 = \bar{A} \bar{B} \bar{D} + \bar{A} B \bar{C} + A B D + A \bar{B} C$$

卡诺图化简逻辑函数例4

$$F_4(A, B, C) = A\bar{B} + B\bar{C} + \bar{B}C + \bar{A}B$$

$$= \Sigma m(1, 2, 3, 4, 5, 6)$$

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

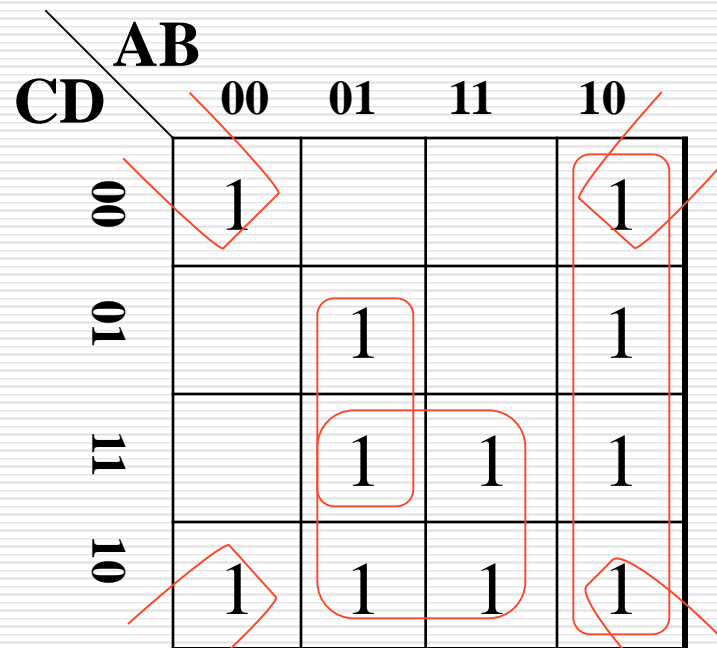
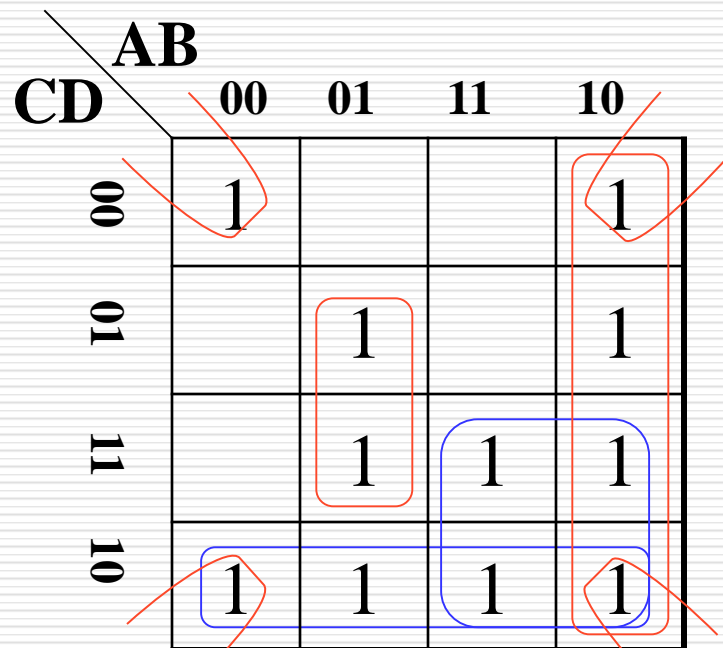
$$F_4 = \bar{A}C + B\bar{C} + A\bar{B}$$

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

$$F_4 = \bar{B}C + \bar{A}B + A\bar{C}$$

卡诺图化简逻辑函数例5

$$F_5(A, B, C, D) = \Sigma m(0, 2, 5, 6, 7, 8, 9, 10, 11, 14, 15)$$



左边圈法结果是 $F_5 = \bar{A}BD + A\bar{B} + \bar{B}\bar{D} + AC + \bar{C}\bar{D}$

显然右边圈法更好 $F_5 = \bar{A}BD + A\bar{B} + \bar{B}\bar{D} + BC$

卡诺图化简逻辑函数例6

$$F_6(A, B, C, D) = \Sigma m(0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14)$$

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

圈1化简

$$F_6 = \bar{B} + \bar{D} + A\bar{C} + \bar{A}C$$

圈0化简

$$\begin{aligned}
 F_6 &= (A + \bar{B} + C + \bar{D})(\bar{A} + \bar{B} + \bar{C} + \bar{D}) \\
 &= (\bar{B} + \bar{D}) + (A + C)(\bar{A} + \bar{C}) \\
 &= \bar{B} + \bar{D} + A\bar{C} + \bar{A}C
 \end{aligned}$$

卡诺图化简逻辑函数例7(非标准形式的化简)

$$F_7 = \bar{B}CD + B\bar{C} + \bar{A}\bar{C} + A\bar{B}C + C\bar{D}$$

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

$$\bar{B}CD$$

$$m_3 + m_{11}$$

$$B\bar{C}$$

$$m_4 + m_5 + m_{12} + m_{13}$$

$$\bar{A}\bar{C}$$

$$m_0 + m_1 + m_4 + m_5$$

$$A\bar{B}C$$

$$m_{10} + m_{11}$$

$$C\bar{D}$$

$$m_2 + m_6 + m_{14} + m_{10}$$

$$F_7 = \bar{B}C + B\bar{C} + \bar{A}\bar{C} + C\bar{D}$$

两变量卡诺图

		B	
		0	1
A	0	m_0	m_1
	1	m_2	m_3

三变量卡诺图

三变量卡诺图

		B			
		00	01	11	10
A	0	m_0	m_1	m_3	m_2
	1	m_4	m_5	m_7	m_6
		C			

四变量卡诺图

$AB \backslash CD$		C			
		00	01	11	10
A	00	m_0	m_1	m_3	m_2
	01	m_4	m_5	m_7	m_6
	11	m_{12}	m_{13}	m_{15}	m_{14}
	10	m_8	m_9	m_{11}	m_{10}
		D			
		B			

2、卡诺图的特点:各小方格对应于各变量不同的组合,而且上下左右在几何上相邻的方格内只有一个因子有差别,这个重要特点成为卡诺图化简逻辑函数的主要依据。

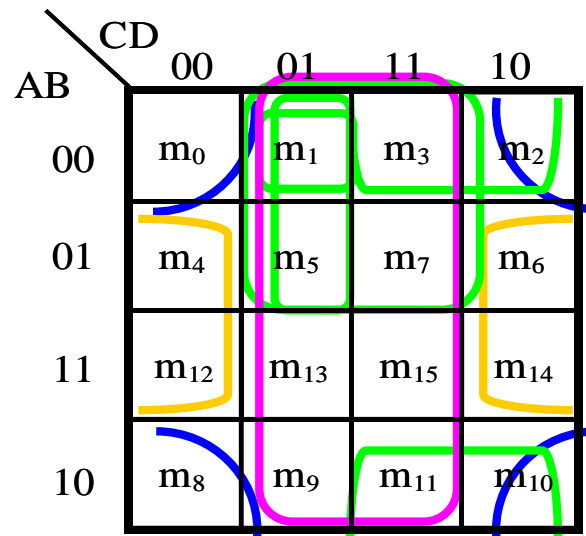
2、化简的步骤

用卡诺图化简逻辑函数的步骤如下：

- (1) 将逻辑函数写成最小项表达式
- (2) 按最小项表达式填卡诺图，凡式中包含的最小项，其对应方格填1，其余方格填0。
- (3) 合并最小项，即将相邻的1方格圈成一组(包围圈)，每一组含 2^n 个方格，对应每个包围圈写成一个新的乘积项。本书中包围圈用虚线框表示。
- (4) 将所有包围圈对应的乘积项相加。

画包围圈时应遵循的原则：

- (1) 包围圈内的方格数一定是 2^n 个，且包围圈必须呈矩形。
- (2) 循环相邻特性包括上下底相邻，左右边相邻和四角相邻。
- (3) 同一方格可以被不同的包围圈重复包围多次，但新增的包围圈中一定要有原有包围圈未曾包围的方格。
- (4) 一个包围圈的方格数要尽可能多，包围圈的数目要尽可能少。

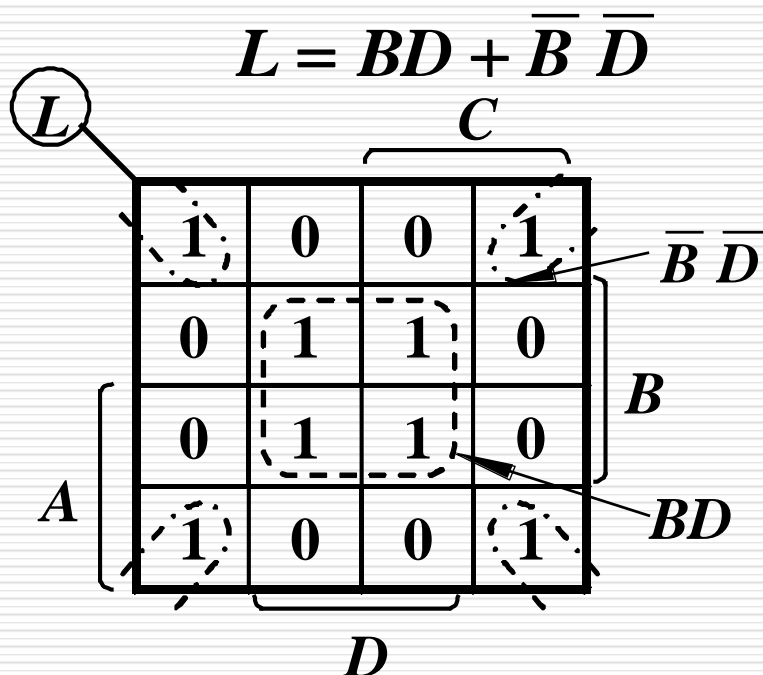


例 :用卡诺图法化简下列逻辑函数

$$L(A, B, C, D) = \sum m(0, 2, 5, 7, 8, 10, 13, 15)$$

解: (1) 由 L 画出卡诺图

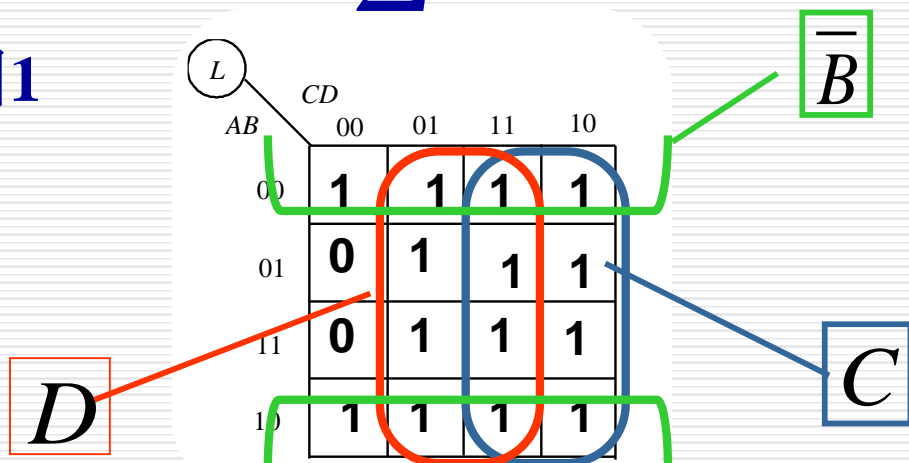
(2) 画包围圈合并最小项, 得最简与-或表达式



例： 用卡诺图化简

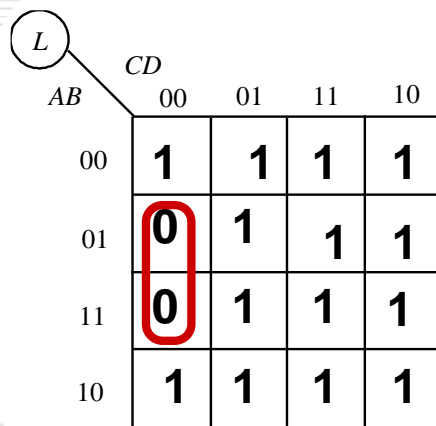
$$L(A, B, C, D) = \sum m(0 \sim 3, 5 \sim 7, 8 \sim 11, 13 \sim 15)$$

圈1



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

圈0



$$\overline{L} = B\overline{C}\overline{D}$$

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

3、具有无关项的化简

(1) 什么叫无关项：

在真值表内对应于变量的某些取值下，函数的值可以是任意的，或者这些变量的取值根本不会出现，这些变量取值所对应的最小项称为无关项或任意项。

在含有无关项逻辑函数的卡诺图化简中，它的值可以取0或取1，具体取什么值，可以根据使函数尽量得到简化而定。

例: 要求设计一个逻辑电路, 能够判断一位十进制数是奇数还是偶数, 当十进制数为奇数时, 电路输出为1, 当十进制数为偶数时, 电路输出为0。

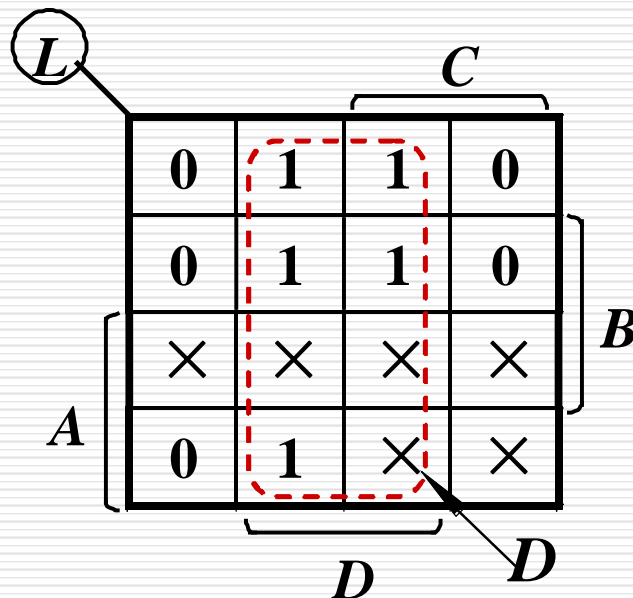
解:

(1) 列出真值表

(2) 画出卡诺图

(3) 卡诺图化简

$$L = D$$



ABCD	L
0000	0
0001	1
0010	0
0011	1
0100	0
0101	1
0110	0
0111	1
1000	0
1001	1
1010	x
1011	x
1100	x
1101	x
1110	x
1111	x