《数字逻辑》

（第二讲）

厦门大学信息学院软件工程系 曾文华

2024年9月2日

课程内容

- 全书共9章：

第1章　基本知识

第2章　逻辑代数基础

第3章　集成门电路与触发器

第4章　组合逻辑电路

第5章　同步时序逻辑电路

第6章　异步时序逻辑电路

第7章　中规模通用集成电路及其应用

第8章　可编程逻辑器件

第9章　综合应用举例

第2章　逻辑代数基础

- 2.1　逻辑代数的基本概念
- 2.2　逻辑代数的基本定理和规则
- 2.3　逻辑函数表达式的形式与变换
- 2.4　逻辑函数化简

2.1　逻辑代数的基本概念

- 逻辑代数是从哲学领域中的逻辑学发展而来的。
- 1847年，英国数学家G.Boole提出了“布尔代数”的概念。
- 1938年，C.E.Shannon提出了“开关代数”的概念，现在则称为“逻辑代数”。
- 逻辑代数L是一个封闭的代数系统，它由一个逻辑变量集K，常量0和1，以及“或”（+）、“与”（·）、“非”（-）3种基本运算所构成，记为：L = {K, +, ·, -, 0, 1}
- 逻辑代数满足下列公理（教材p19-p20）：
  - 公理1：交换律
  - 公理2：结合律
  - 公理3：分配律
  - 公理4：0-1律
  - 公理5：互补律

• 2.1.1 逻辑变量及基本逻辑运算

– 逻辑代数是一种二值代数系统，任何逻辑变量的取值只有两种可能性：**0或1**。

– 在数字系统中，可以用开关的接通和断开、电压的高和低等，表示逻辑值1和0。

– 逻辑代数中定义了“或”、“与”、“非”等3种基本运算。

– 1、或运算

- **F = A+B** 或者 **F = A∨B**
- 表2.1：或运算表。
- 图2.1：并联开关电路（灯F与开关A、开关B之间的关系为或逻辑）。

表 2.1 或运算表

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

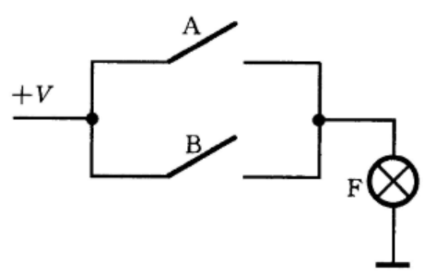


图 2.1 并联开关电路

开关“开”表示0，
开关“关”表示1

灯“亮”表示1，灯“不亮”表示0

– 2、与运算

- F = A·B　　或者　F = A∧B
- 表2.2：与运算表。
- 图2.2：串联开关电路（灯F与开关A、开关B之间的关系为与逻辑）。

表 2.2　与运算表

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

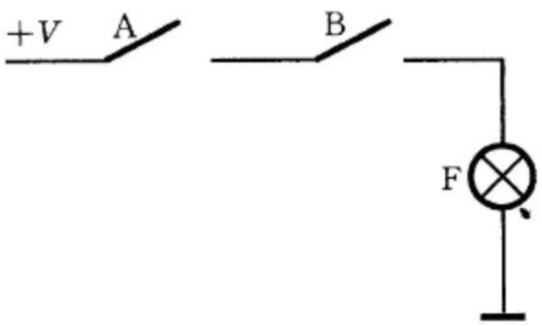


图 2.2　串联开关电路

开关“开”表示0，
开关“关”表示1

灯“亮”表示1，灯“不亮”表示0

– 3、非运算

- **F = /A** 或者 **F = ¬ A**
- 表**2.3**：非运算表。
- 图**2.3**：开关与灯并联电路（灯**F**与开关**A**之间的关系为非逻辑）。

表 2.3 非运算表

A	F
0	1
1	0

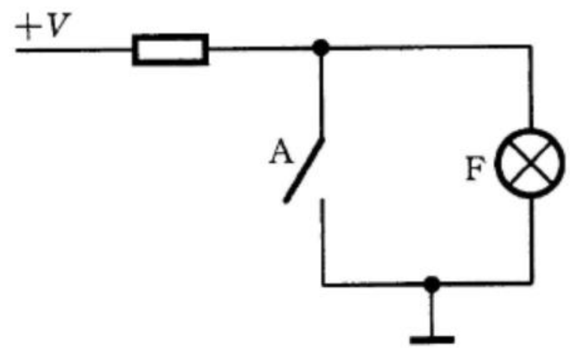


图 2.3 开关与灯并联电路

开关“开”表示**0**，
开关“关”表示**1**

灯“亮”表示**1**，灯“不亮”表示**0**

- **2.1.2 逻辑函数及逻辑函数间的相等**
  - 1、逻辑函数的定义
    - 逻辑函数的特点：
      ① 逻辑变量（$A_1,A_2,\ldots,A_n$）和逻辑函数（F）的取值只有0和1两种可能；
      ② 逻辑函数和逻辑变量之间的关系是由或、与、非3种基本运算决定的。
    - 逻辑函数：$F = f(A_1,A_2,\ldots,A_n)$
    - 图2.4：广义的逻辑电路。

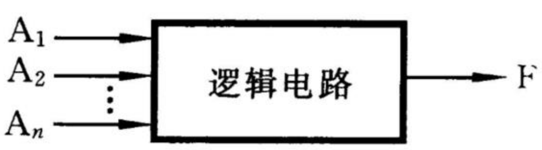


图 2.4　广义的逻辑电路

- **2、逻辑函数的相等**

- 设有两个逻辑函数：
  - $F_1 = f_1(A_1,A_2,\ldots,A_n)$
  - $F_2 = f_2(A_1,A_2,\ldots,A_n)$

- 若对应于逻辑变量$A_1,A_2,\ldots,A_n$的任何一组取值，F_1和F_2的值都相同，则称函数F_1和F_2相等。

- 判断两个逻辑函数是否相等的两种方法：
  - 第一种方法：列出输入变量所有可能的取值组合，并按逻辑运算法则计算出各种输入取值下两个函数的相应值，然后进行比较；
  - 第二种方法：用逻辑代数的公理、定理和规则进行证明。

• 2.1.3 逻辑函数的表示法

– 逻辑函数常用的表示方法有（3种）：

– 1、逻辑表达式

• 例如：F = f(A,B) = /A·B + A·/B

– 2、真值表

• 真值表是一种由逻辑变量的所有可能取值组合及其对应的逻辑函数值所构成的表格。

• 表2.4：函数F = A·/B + /A·C的真值表。

表 2.4　函数 $F=A\bar{B}+\bar{A}C$ 的真值表

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

– 3、卡诺图

- 卡诺图是一种用图形描述函数的方法（具体见2.4.2小节）。
- 图2.8：函数F = A·B + C·D + /A·/B·C的卡诺图。

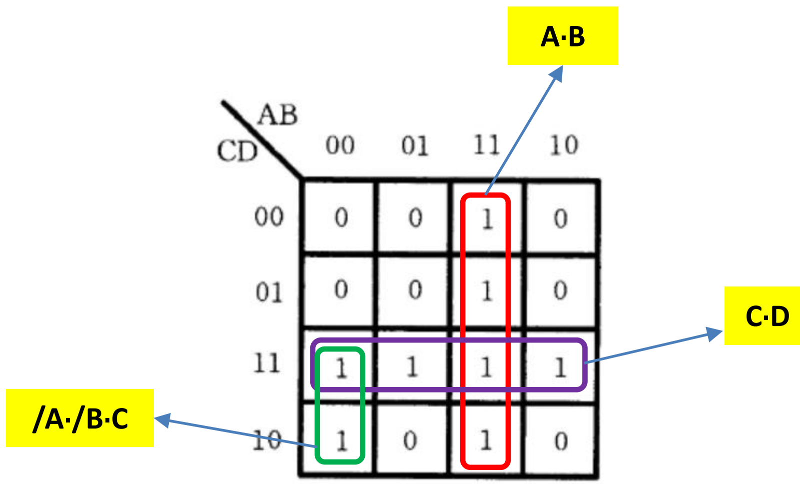


图 2.8　$F(A,B,C,D)=AB+CD+\bar{A}\bar{B}C$ 的卡诺图

2.2 逻辑代数的基本定理和规则

- **2.2.1 基本定理**
  - 定理1：0-1律
  - 定理2：重叠律
  - 定理3：吸收律
  - 定理4：消除律
  - 定理5：对合律
  - 定理6：互补律（也称摩根定理） /(A+B) = /A · /B　　/(A·B) = /A + /B
  - 定理7：并项律
  - 定理8：包含律
  - 具体见教材p24-p25

• 2.2.2 重要规则

– 1、代入规则

- 代入规则：任何一个含有变量A的逻辑等式，如果将所有出现A的位置都代之以同一个逻辑函数F，则等式仍然成立。

- 例如，给定逻辑等式：A·(B+C) = A·B+A·C，若等式中的C都用(C+D)代替，则该等式仍然成立，即：A·[B+(C+D)] = A·B + A·(C+D)

– 2、反演规则

- 反演规则：如果将逻辑函数F表达式中所有的“·”变成“+”，“+”变成“·”，“0”变成“1”，“1”变成“0”，原变量（A）变成反变量（/A），反变量（/A）变成原变量（A），并保持原函数中的运算顺序不变，则所得到的新的函数为原函数F的反函数/F。

- 例如，F = /A·B+C·/D，则：/F = (A+/B)·(/C+D)

如何由F，写出/F？

- 例如，F = /A+/B·(C+/D·E)，则：/F = A·[B+/C·(D+/E)]

– 3、对偶规则

- 对偶规则：如果将逻辑函数F表达式中所有的“·”变成“+”，“+”变成“·”，“0”变成“1”，“1”变成“0”，并保持原函数中的运算顺序不变，则所得到的新的逻辑表达式称为函数F的对偶式，记作F’。

- 例如：
  - F_1=/A·B+/B·(C+0)，则：F_1’=(/A+B)·(/B+C·1)
  - 令：F_2=(/A+B)·(/B+C·1)=F_1’，则：F_2’=/A·B+/B·(C+0)=F_1
  - F_3=A·B+/A·C+C·(D+E)，则：F_3’=(A+B)·(/A+C)·(C+D·E)
  - 令：F_4=(A+B)·(/A+C)·(C+D·E)=F_3’，则：F_4’=A·B+/A·C+C·(D+E)=F_3

如何由F，写出F’？

- 可见，如果F的对偶式是F’，则F’的对偶式就是F，即F和F’是互为对偶式，即：F=(F’)’。

- 自对偶函数：有些逻辑函数表达式的对偶式就是原函数表达式本身，即F'=F，称函数F为自对偶函数。例如，函数F=(A+/C)·/B+A·(/B+/C)，经过推导，F'也等于(A+/C)·/B+A·(/B+/C)，即函数F是一自对偶函数。

- 若两个逻辑函数表达式F(A,B,C,...)和G(A,B,C,...)相等，则其对偶式F’(A,B,C,...)和G’(A,B,C,...)也相等。例如，F(A,B,C)=A·B+/A·C+/B·C，G(A,B,C)=A·B+C；F’(A,B,C)=(A+B)·(/A+C)·(/B+C)，G’(A,B,C)=(A+B)·C；因为：A·B+/A·C+/B·C = A·B+C，即F(A,B,C) = G(A,B,C)；则有：F’(A,B,C) = G’(A,B,C)，即(A+B)·(/A+C)·(/B+C) = (A+B)·C。

• 2.2.3 复合逻辑

– 基本的逻辑关系：与、或、非。

– 由3种基本的逻辑关系可以得到与非、或非、与或非、异或等复合逻辑关系。

– 1、与非逻辑及或非逻辑

• （1）与非逻辑（与非门）

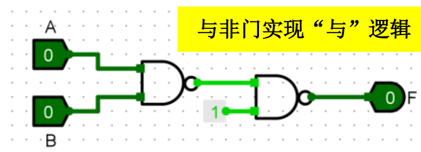
– F = /(A·B)

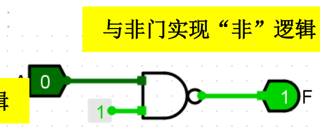
– 利用与非门可以实现3种基本的逻辑：

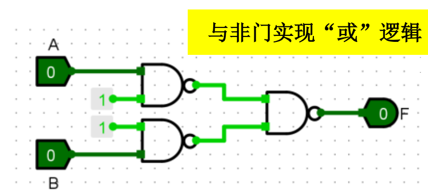
» 与：F = A·B = /(/(A·B)·1))

» 或：F = A+B = /(/(A·1)·/(B·1))

» 非：F = /A = /(A·1)







• （2）或非逻辑（或非门）

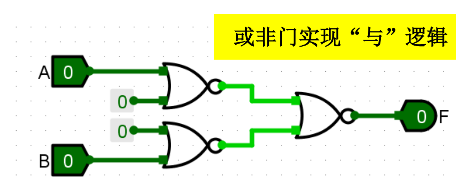
– F = /(A+B)

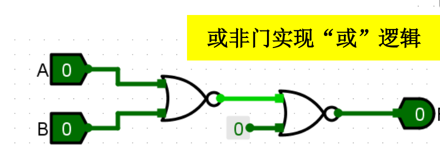
– 利用或非门可以实现3种基本的逻辑：

» 与：F = A·B = /(/(A+0) + /(B+0))

» 或：F = A+B = /(/(A+B)+0)

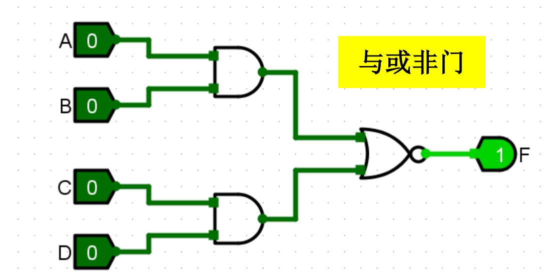
» 非：F = /A = /(A+0)



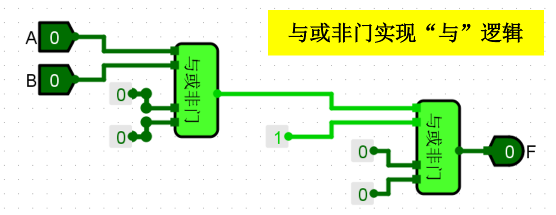


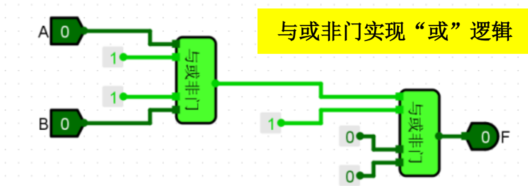
– 2、与或非逻辑（与或非门）

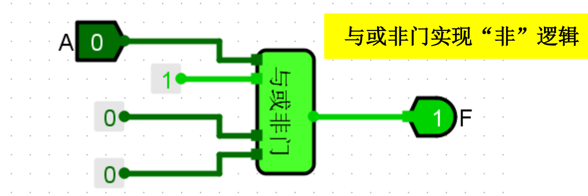
- F = /(A·B + C·D)



- 利用与或非门可以实现3种基本的逻辑：
  - » 与：F = A·B = /((/(A·B + 0·0))·1 + 0·0)
  - » 或：F = A+B = /((/(A·1 + B·1))·1 + 0·0)
  - » 非：F = /A = /(A·1 + 0·0)

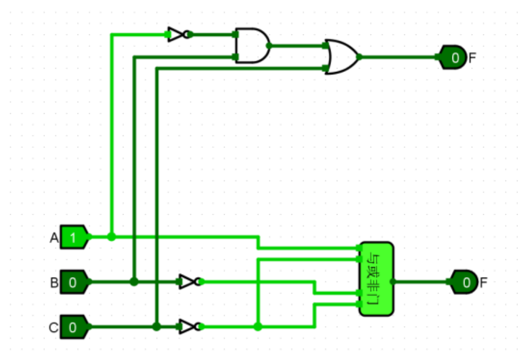






- 但是通常不利用与或非门来实现基本的逻辑功能（与、或、非），原因是这样会增加电路的复杂度。

- 有一些逻辑公式可以利用与或非门来实现，例如：F = /A·B+C = /(A·/C + /B·/C)



F = /A·B+C

与或非门实现：F = /(A·/C + /B·/C)

– 3、异或逻辑及同或逻辑

- （1）异或逻辑（异或门）
  - $F = A \oplus B = \overline{A} \cdot B + A \cdot \overline{B}$
  - 在进行异或运算的多个变量中，若有奇数个变量的值为1，则运算结果为1；若有偶数个变量的值为1，则运算结果为0。例如：$1\oplus1\oplus1\oplus1\oplus1=1$；$1\oplus1\oplus1\oplus1=0$。

- （2）同或逻辑（同或门、异或非门）
  - $F = A \odot B = \overline{A} \cdot \overline{B} + A \cdot B$
  - 同或逻辑与异或逻辑的关系既互为相反，又互为对偶，即有：
    - » $\overline{(A\oplus B)} = A \odot B$
    - » $(A\oplus B)' = A \odot B$

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

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

  - 在进行同或运算的多个变量中，若有奇数个变量的值为1，则运算结果为0；若有偶数个变量的值为1，则运算结果为1。例如：$1\odot1\odot1\odot1\odot1=0$；$1\odot1\odot1\odot1=1$。
  - 同或逻辑（同或门）也称为异或非逻辑（异或非门）。

2.3 逻辑函数表达式的形式与变换

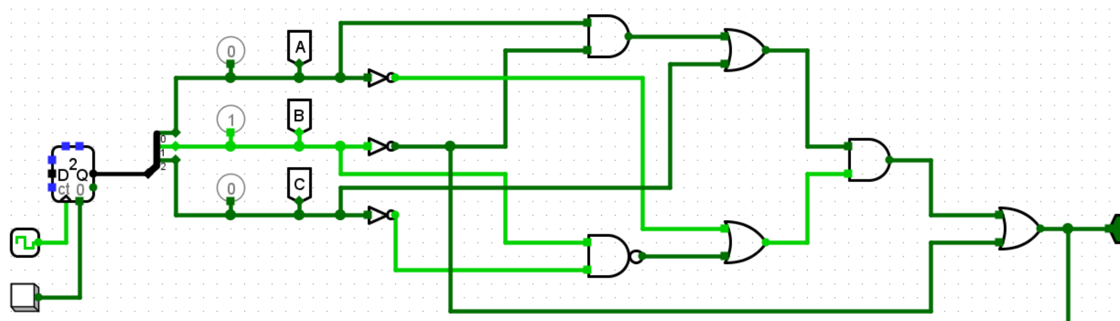
• 2.3.1 逻辑函数表达式的基本形式

- 逻辑函数表达式有“与-或”表达式和“或-与”表达式两种基本形式。
- **1、“与-或”表达式**
  - “与-或”表达式也称为“积之和”表达式。例如：F(A,B,C) = /A·B + A·/B·C + /C。
- **2、“或-与”表达式**
  - “或-与”表达式也称为“和之积”表达式。例如：F(A,B,C,D) = (/A + B) · (B + /C) · (A + /B + C) · /D。

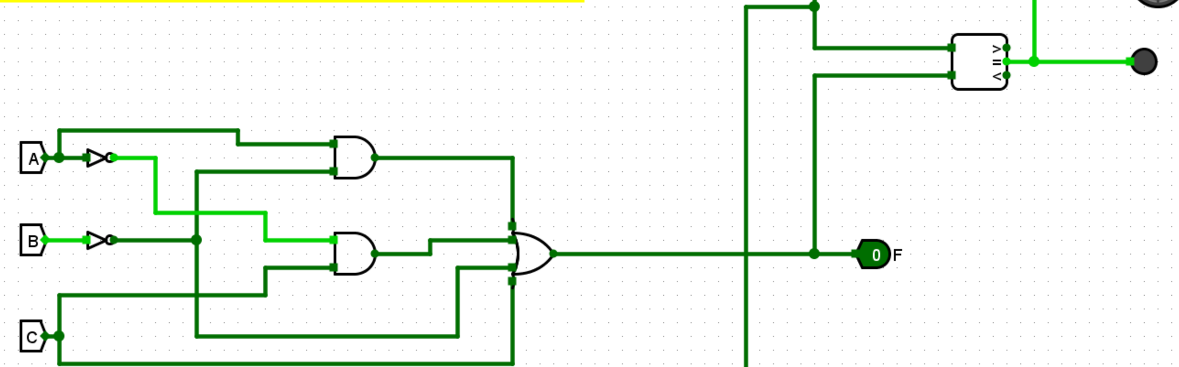
结论

- 任何形式的逻辑函数表达式都可以用“与-或”表达式和“或-与”表达式表示。例如： F(A,B,C) = (A·/B+C)·(/A+/(B·/C))+/B，可以表示为：F(A,B,C) = A·/B + /A·C + /B + C，也可以表示为：F(A,B,C) = (A+/B+C) · (/A+/B+C) · (/B+C)。

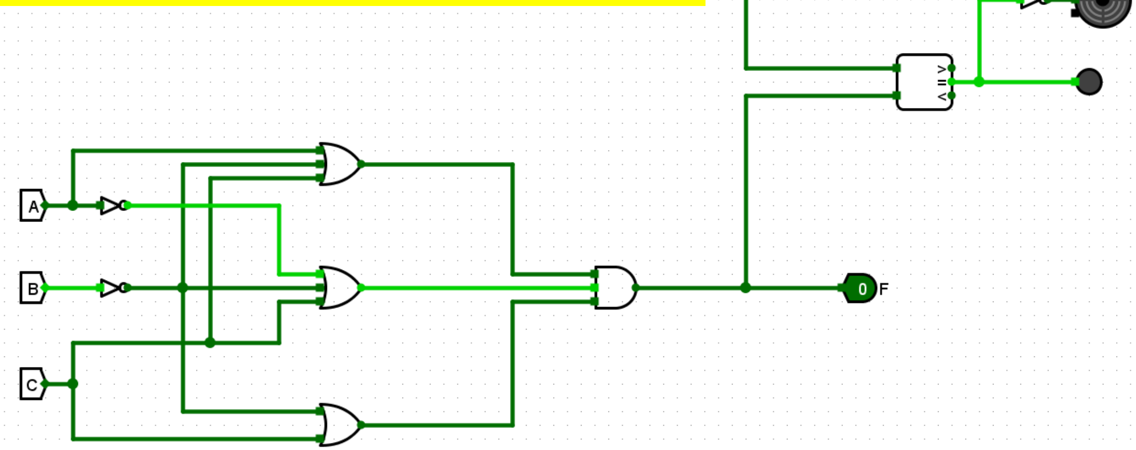
在Logisim上实现F(A,B,C)，并判断3种形式的F是否相等？



F(A,B,C) = (A·/B+C)·(/A+/(B·/C))+/B



“与-或”表达式（积之和）： F(A,B,C) = A·/B + /A·C + /B + C



“或-与”表达式（和之积）： F(A,B,C) = (A+/B+C) · (/A+/B+C) · (/B+C)

• 2.3.2 逻辑函数表达式的标准形式

– 1、最小项和最大项

• （1）最小项的定义和性质

- 1个变量（A）的最小项有2个，分别是：/A， A
- 2个变量（A、B）的最小项有4个，分别是：/A·/B， /A·B， A·/B， A·B
- 3个变量（A、B、C）的最小项有8个，分别是：/A·/B·/C， /A·/B·C， /A·B·/C， /A·B·C， A·/B·/C， A·/B·C， A·B·/C， A·B·C
- 通常用m_i表示最小项，例如3个变量的8个最小项分别为：m_0=/A·/B·/C，m_1=/A·/B·C，m_2=/A·B·/C，m_3=/A·B·C，m_4=A·/B·/C，m_5=A·/B·C，m_6=A·B·/C，m_7=A·B·C
- 最小项也称为标准与项。最小项具有以下的4个性质：
  - » 性质1：任意一个最小项，其相应变量有且仅有一种取值，使这个最小项的值为1；并且，最小项不同，使其值为1的变量取值也不同。
    - • 例如3个变量的最小项m_1=/A·/B·C，只有A=0、B=0、C=1时，m_1=1；而对于最小项m_4=A·/B·/C，使m_4=1的变量取值只为：A=1、B=0、C=0。

» 性质2：相同变量构成的两个不同最小项相与为0。

- 例如：
  - 3个变量中相同变量A构成的两个不同最小项分别是：/A·/B·/C和A·/B·/C，/A·B·/C和A·B·/C，/A·/B·C和A·/B·C，/A·B·C和A·B·C；
  - 其相与值为：(/A·/B·/C)·(A·/B·/C)=0，(/A·B·/C)·(A·B·/C)=0，(/A·/B·C)·(A·/B·C)=0，(/A·B·C)·(A·B·C)=0。

» 性质3：n个变量的全部最小项相或为1。

- 例如：
  - 1个变量的2个最小项相或：/A + A=1
  - 2个变量的4个最小项相或：/A·/B + /A·B + A·/B + A·B =1
  - 3个变量的8个最小项相或：/A·/B·/C + /A·/B·C + /A·B·/C + /A·B·C + A·/B·/C + A·/B·C + A·B·/C + A·B·C =1

» 性质4：n个变量构成的最小项，每个最小项有n个相邻最小项。

相邻最小项

- 相邻最小项是指除1个变量互为相反外，其余部分均相同的最小项。例如，/A·/B·/C 和 A·/B·/C 是相邻最小项（仅1个变量A互为相反），而 A·/B·/C 和 /A·B·/C 则不是相邻最小项（有2个变量A、B互为相反）。
- 对于1个变量构成的2最小项（/A、A）：
  - /A有1个相邻的最小项（A）
  - A有1个相邻的最小项（/A）
- 对于2个变量构成的最小项（/A·/B、/A·B、A·/B、A·B）：
  - /A·/B有2个相邻的最小项（A·/B和/A·B）
  - /A·B有2个相邻的最小项（A·B、/A·/B）
  - A·/B有2个相邻的最小项（/A·/B，A·B）
  - A·B有2个相邻的最小项（/A·B、A·/B）
- 对于3个变量构成的最小项（/A·/B·/C，/A·/B·C，/A·B·/C，/A·B·C，A·/B·/C，A·/B·C，A·B·/C，A·B·C）：
  - /A·/B·/C有3个相邻的最小项（A·/B·/C、/A·B·/C、/A·/B·C）
  - /A·/B·C有3个相邻的最小项（A·/B·C、/A·B·C、/A·/B·/C）
  - /A·B·/C有3个相邻的最小项（A·B·/C、/A·/B·/C、/A·B·C）
  - /A·B·C有3个相邻的最小项（A·B·C、/A·/B·C、/A·B·/C）
  - A·/B·/C有3个相邻的最小项（/A·/B·/C、A·B·/C、A·/B·C）
  - A·/B·C有3个相邻的最小项（/A·/B·C、A·B·C、A·/B·/C）
  - A·B·/C有3个相邻的最小项（/A·B·/C、A·/B·/C、A·B·C）
  - A·B·C有3个相邻的最小项（/A·B·C、A·/B·C、A·B·/C）

- （2）最大项的定义和性质
  - 1个变量（A）的最大项有2个，分别是：A，/A
  - 2个变量（A、B）的最大项有4个，分别是：A+B，A+/B，/A+B，/A+/B
  - 3个变量（A、B、C）的最大项有8个，分别是：A+B+C，A+B+/C，A+/B+C，A+/B+/C，/A+B+C，/A+B+/C，/A+/B+C，/A+/B+/C
  - 通常用M_i表示最大项，例如3个变量的8个最大项分别为：M_0=A+B+C，M_1=A+B+/C，M_2=A+/B+C，M_3=A+/B+/C，M_4=/A+B+C，M_5=/A+B+/C，M_6=/A+/B+C，M_7=/A+/B+/C
  - 最大项也称为标准或项。最大项具有以下的4个性质：
    - » 性质1：任意一个最大项，其相应变量有且仅有一种取值，使这个最大项的值为0；并且，最大项不同，使其值为0的变量取值也不同。
      - 例如3个变量的最大项M_1=A+B+/C，只有A=0、B=0、C=1时，M_1=0；而对于最大项M_4=/A+B+C，使M_4=0的变量取值只为：A=1、B=0、C=0。
    - » 性质2：相同变量构成的两个不同最大项相或为1。
      - 例如：
        - 3个变量中相同变量A构成的两个不同最大项分别是：/A+/B+/C和A+/B+/C，/A+B+/C和A+B+/C，/A+/B+C和A+/B+C，/A+B+C和A+B+C；
        - 其相或值为：(/A+/B+/C)+(A+/B+/C)=1，(/A+B+/C)+(A+B+/C)=1，(/A+/B+C)+(A+/B+C)=1，(/A+B+C)+(A+B+C)=1。

» 性质3：n个变量的全部最大项相与为0。

- 例如：
  - 1个变量的2个最大项相与：A · /A = 0
  - 2个变量的4个最大项相与：(A+B) · (A+/B) · (/A+B) · (/A+/B) = 0
  - 3个变量的8个最大项相与：(A+B+C) · (A+B+/C) · (A+/B+C) · (A+/B+/C) · (/A+B+C) · (/A+B+/C) · (/A+/B+C) · (/A+/B+/C) = 0

» 性质4：n个变量构成的最大项，每个最大项有n个相邻最大项。

相邻最大项

- 相邻最大项是指除1个变量互为相反外，其余部分均相同的最大项。例如，/A+/B+/C 和 A+/B+/C 是相邻最大项（仅1个变量A互为相反），而 A+/B+/C 和 /A+B+/C 则不是相邻最大项（有2个变量A、B互为相反）。

- 对于1个变量构成的2最大项（A、/A）：
  - A有1个相邻的最大项（/A）
  - /A有1个相邻的最大项（A）

- 对于2个变量构成的最大项（A+B、A+/B、/A+B、/A+/B）：
  - A+B有2个相邻的最大项（/A+B、A+/B）
  - A+/B有2个相邻的最大项（/A+/B，A+B）
  - /A+B有2个相邻的最大项（A+B、/A+/B）
  - /A+/B有2个相邻的最大项（A+/B和/A+B）

- 对于3个变量构成的最大项（A+B+C，A+B+/C，A+/B+C，A+/B+/C，/A+B+C，/A+B+/C，/A+/B+C，/A+/B+/C）：
  - A+B+C有3个相邻的最大项（/A+B+C、A+/B+C、A+B+/C）
  - A+B+/C有3个相邻的最大项（/A+B+/C、A+/B+/C、A+B+C）
  - A+/B+C有3个相邻的最大项（/A+/B+C、A+B+C、A+/B+/C）
  - A+/B+/C有3个相邻的最大项（/A+/B+/C、A+B+/C、A+/B+C）
  - /A+B+C有3个相邻的最大项（A+B+C、/A+/B+C、/A+B+/C）
  - /A+B+/C有3个相邻的最大项（A+B+/C、/A+/B+/C、/A+B+C）
  - /A+/B+C有3个相邻的最大项（A+/B+C、/A+B+C、/A+/B+/C）
  - /A+/B+/C有3个相邻的最大项（A+/B+/C、/A+B+/C、/A+/B+C）

– 表2.5：2变量最小项、最大项真值表

表 2.5　2 变量最小项、最大项真值表

变量		最小项				最大项			
A	B	$\bar{A}\bar{B}$	$\bar{A}B$	$A\bar{B}$	AB	$A+B$	$A+\bar{B}$	$\bar{A}+B$	$\bar{A}+\bar{B}$
		m_0	m_1	m_2	m_3	M_0	M_1	M_2	M_3
0	0	1	0	0	0	0	1	1	1
0	1	0	1	0	0	1	0	1	1
1	0	0	0	1	0	1	1	0	1
1	1	0	0	0	1	1	1	1	0

- （3）最小项和最大项的关系
  - 相同变量构成的最小项m_i和最大项M_i之间存在互补关系。即：$/m_i = M_i$，或者，$m_i = /M_i$。
  - 例如：
    - » 1变量构成的最小项：m_0=/A，M_0=/m_0=A；m_1=A，M_1=/m_1=/A
    - » 2变量构成的最小项：m_0=/A·/B，M_0=/m_0=A+B；m_1=/A·B，M_1=/m_1=A+/B；m_2=A·/B，M_2=/m_2=/A+B；m_3=A·B；M_3=/m_3=/A+/B
    - » 3变量构成的最小项：m_0=/A·/B·/C，M_0=/m_0=A+B+C；m_1=/A·/B·C，M_1=/m_1=A+B+/C；m_2=/A·B·/C，M_2=/m_2=A+/B+C；m_3=/A·B·C，M_3=/m_3=A+/B+/C；m_4=A·/B·/C，M_4=/m_4=/A+B+C；m_5=A·/B·C，M_5=/m_5=/A+B+/C；m_6=A·B·/C，M_6=/m_6=/A+/B+C；m_7=A·B·C，M_7=/m_7=/A+/B+/C

– 2、逻辑函数表达式的标准形式

- 逻辑函数表达式的标准形式有标准“与-或”表达式和标准“或-与”表达式两种类型：

- （1）标准“与-或”表达式

  – 由若干最小项相或构成的逻辑表达式称为标准“与-或”表达式，也称为最小项表达式。

  – 例如：F(A,B,C) = /A·/B·C + /A·B·/C + A·/B·/C + A·B·C

  – 也可以表示为：$F(A,B,C) = m_1+m_2+m_4+m_7 = \sum m(1,2,4,7)$

- （2）标准“或-与”表达式

  – 由若干最大项相与构成的逻辑表达式称为标准“或-与”表达式，也称为最大项表达式。

  – 例如：F(A,B,C) = (A+B+C) · (/A+B+/C) · (/A+/B+/C)

  – 也可以表示为：$F(A,B,C) = M_0 \cdot M_5 \cdot M_7 = \prod M(0,5,7)$

• 2.3.3 逻辑函数表达式的转换

– 将一个任意逻辑函数表达式转换成标准表达式有两种常用方法：

– 1、代数转换法　代数转换法实现起来比较困难

- 用代数转换法将一个任意逻辑函数转换为标准“与-或”表达式，一般分为两步：
  - 第一步：将函数表达式变换成一般“与-或”表达式。
    - » 例如：F(A,B,C) = /((A·/B+B·/C)·/(A·B)) = /A·/B + /A·C + B·C + A·B
  - 第二步：反复使用公式X=X·(/Y+Y)，将表达式中所有非最小项的与项扩展成最小项。
    - » 例如：F(A,B,C) = /A·/B + /A·C + B·C + A·B = /A·/B·(/C+C) + /A·C·(/B+B) + B·C·(/A+A) + A·B·(/C+C) = /A·/B·/C + /A·/B·C + /A·B·C + A·B·/C + A·B·C = $m_0+m_1+m_3+m_6+m_7$ = $\sum$m(0,1,3,6,7)

- 用代数转换法将一个任意逻辑函数转换为标准“或-与”表达式，同样分为两步：
  - 第一步：将函数表达式变换成一般“或-与”表达式。
    - » 例如：F(A,B,C) = /(A·B+/A·C) + /B·C = (/A+/B) · (A+/B+/C) · (/A+/B+C)
  - 第二步：反复使用公式X=(X+/Y)·(X+Y)，将表达式中所有非最大项的或项扩展成最大项。

    (X+/Y)·(X+Y) = X·X+X·Y+/Y·X+/Y·Y = X+X·Y+X·/Y+0 = X·(1+Y+/Y) = X

    - » 例如：F(A,B,C) = (/A+/B) · (A+/B+/C) · (/A+/B+C) = (/A+/B +/C) · (/A+/B +C) · (A+/B+/C) · (/A+/B+C) = (A+/B+/C) · (/A+/B +C) · (/A+/B +/C) = $M_3 \cdot M_6 \cdot M_7$ = $\prod$M(3,6,7)

- 如何将函数表达式变换成一般“或-与”表达式？
  - 反复利用公式：**$X+Y\cdot Z = (X+Y)\cdot(X+Z)$**

    $(X+Y)\cdot(X+Z) = X\cdot X+X\cdot Z+Y\cdot X+Y\cdot Z = X+X\cdot Z+Y\cdot X+Y\cdot Z = X\cdot(1+Z+Y)+Y\cdot Z = X+Y\cdot Z$
  - 例如：
    - **F(A,B,C) = /(A·B+/A·C) + /B·C**

      = /(A·B) · /(/A·C) + /B·C

      = (/A+/B) · (A+/C) + /B·C

      = ((/A+/B) · (A+/C) + /B) · ((/A+/B) · (A+/C) + C)

      = (/B + (/A+/B) · (A+/C)) · (C + (/A+/B) · (A+/C))

      = (/B + (/A+/B)) · (/B + (A+/C)) · (C + (/A+/B)) · (C + (A+/C))

      = (/A+/B) · (A+/B+/C) · (/A+/B+C)

– 2、真值表转换法

真值表转换法实现起来比较容易

- 例如：将函数表达式F(A,B,C)=A·/B+B·/C表示成最小项表达式。

- 首先列出F的真值表（表2.6），然后根据真值表直接写出F的最小项表达式 F(A,B,C)= ∑ m(2,4,5,6)

表 2.6　函数 $F(A,B,C)=A\bar{B}+B\bar{C}$ 的真值表

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$F(A,B,C)=\sum m(2,4,5,6)$

- 例如：将函数表达式F(A,B,C)=/A·C+A·/B·/C表示成最大项表达式。

- 首先列出F的真值表（表2.7），然后根据真值表直接写出F的最大项表达式 F(A,B,C)= ∏ M(0,2,5,6,7)

表 2.7　函数 $F(A,B,C)=\bar{A}C+A\bar{B}\,\bar{C}$ 的真值表

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

$F(A,B,C)=\prod M(0,2,5,6,7)$

2.4 逻辑函数化简

2.4.1 代数化简法
2.4.2 卡诺图化简法
2.4.3 列表化简法

- **2.4.1 代数化简法**

代数化简法实现起来比较困难

  - 1、“与-或”表达式的化简
    - 最简“与-或”表达式应满足两个条件：
      ① 表达式中的与项个数最少；
      ② 在满足条件①的前提下，每个与项中的变量个数最少。
    - 化简“与-或”表达式的几种常用方法：
      - （1）并项法
        » 利用定理：**A·B+A·/B = A**，将两个与项合并为一个与项。
      - （2）吸收法
        » 利用定理：**A+A·B = A**，消去多余的项。
      - （3）消去法
        » 利用定理：**A+/A·B = A+B**，消去多余的变量。
      - （4）配项法
        » 利用公理：**A·1 = A** 及 **A+/A = 1**，先从函数式中适当选择某些与项，并配上其所缺的一个合适的变量，然后再利用并项、吸收和消去等方法进行简化。

根据公式：**X+Y·Z = (X+Y)·(X+Z)**

有：**A+/A·B = (A+/A)·(A+B) = A+B**

- 例2.5： F=A·D+A·/D+A·B+/A·C+B·D+/B·E+D·E
- 解：
  - F=A·D+A·/D+A·B+/A·C+B·D+/B·E+D·E   A·D+A·/D=A

    =A+A·B+/A·C+B·D+/B·E+D·E   A+A·B=A

    =A+/A·C+B·D+/B·E+D·E   A+/A·C=A+C

    =A+C+B·D+/B·E+D·E

    教材上写成：F=A+C+B·D+/B·E，有误

- 例2.6： F=/A·/C· (/B+B·D)+A·/C·D
- 解：
  - F=/A·/C· (/B+B·D)+A·/C·D   /B+B·D=/B+D

    =/A·/C· (/B+D)+A·/C·D

    =/A·/B·/C+/A·/C·D+A·/C·D   /A·/C·D+A·/C·D=/C·D

    =/A·/B·/C+/C·D

- 例2.7： F=A·B+A·/C+/B·C+B·/C+/B·D+B·/D+A·D·E
- 解：
  - F=A·B+A·/C+/B·C+B·/C+/B·D+B·/D+A·D·E   B+/C=/(/B·C)

    =A·/(/B·C)+/B·C+B·/C+/B·D+B·/D+A·D·E   A/(/B·C)+/B·C=A+/B·C

    =A+/B·C+B·/C+/B·D+B·/D+A·D·E   A+A·D·E=A

    =A+/B·C+B·/C+/B·D+B·/D

    =A+/B·C·(/D+D)+B·/C+/B·D+B·/D·(C+/C)

    =A+/B·C·/D+/B·C·D+B·/C+/B·D+B·C·/D+B·/C·/D

    =A+(/B·C·/D+B·C·/D)+(/B·D·C+/B·D)+(B·/C+B·/C·/D)

    =A+C·/D+/B·D+B·/C

– 2、“或-与”表达式的化简

- 最简“或-与”表达式应满足两个条件：
  ① 表达式中的或项个数最少；
  ② 在满足条件①的前提下，每个或项中的变量个数最少。

- “或-与”表达式的化简方法和“与-或”表达式的化简方法类似。

- 例2.8： F=(A+B)·(A+/B)·(B+C)·(B+C+D)

X·(X+Y)=X·X+X·Y=X+X·Y=X·(1+Y)=X

- 解：
  – F=(A+B)·(A+/B)·(B+C)·(B+C+D)
    =(A+B)·(A+/B)·(B+C)
    =A·(B+C)

(B+C)·(B+C+D)=B+C

(A+B)·(A+/B)=A

(X+Y)·(X+/Y)=X·X+X·/Y+Y·X+Y·/Y=X+X·/Y+X·Y+0=X· (1+/Y)+X·Y=X+X·Y=X· (1+Y)=X

- 例2.9：两次对偶法。F=(A+/B)·(/A+B)·(B+C)·(/A+C)
- 解：
  – 先求F的对偶式F’，并进行化简（“与-或”表达式的化简）：
    » F’=A·/B+/A·B+B·C+/A·C
      =A·/B+/A·B+(B+/A)·C
      =A·/B+/A·B+/(A·/B)·C
      =A·/B+/A·B+C

B+/A=/(A·/B)

A·/B+/(A·/B)·C=A·/B+C

  – 再对F’求对偶，得到F：
    » F = (F’)’ = (A+/B)·(/A+B)·C

• 2.4.2 卡诺图化简法

– 卡诺图化简法也称为图形化简法，该方法简单、直观、容易掌握。

– 1、卡诺图的构成

- 图2.5：2～5变量卡诺图（注：1变量太简单，通常不需要卡诺图）
  - AB（CD、DE）取值为00、01、11、10，不是00、01、10、11

注意

  - ABC取值为000、001、011、010、100、101、111、110，不是000、001、010、011、100、101、110、111

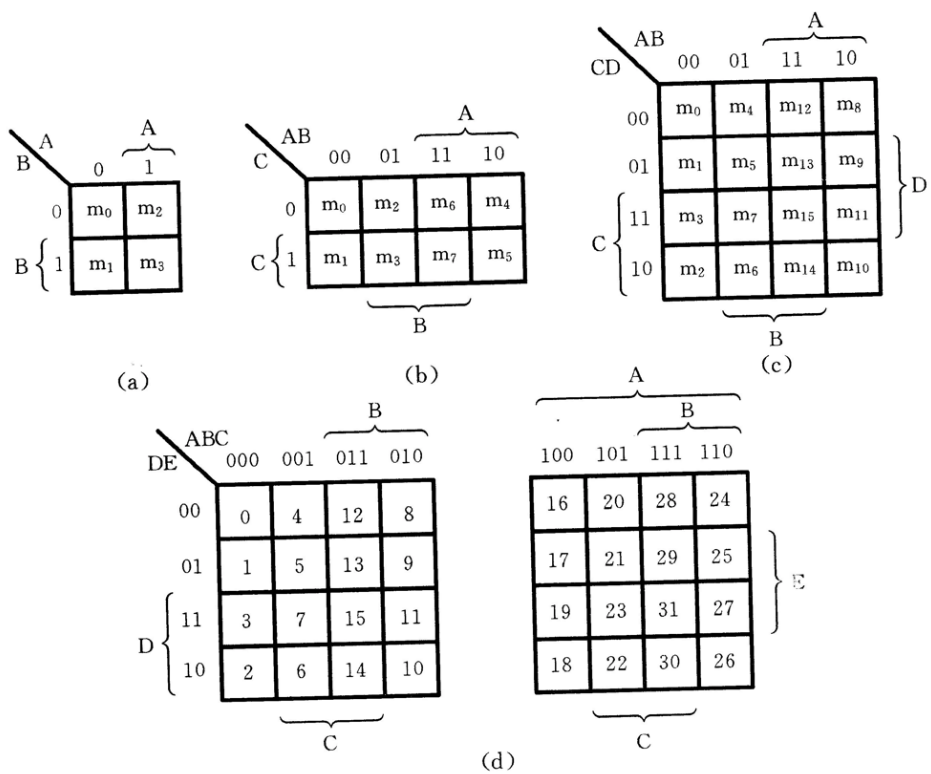
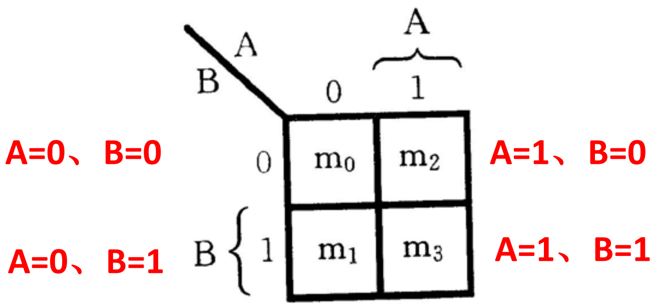
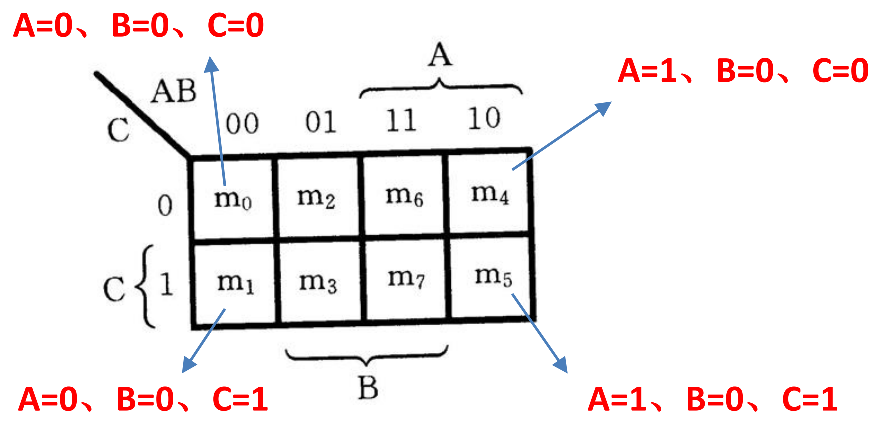
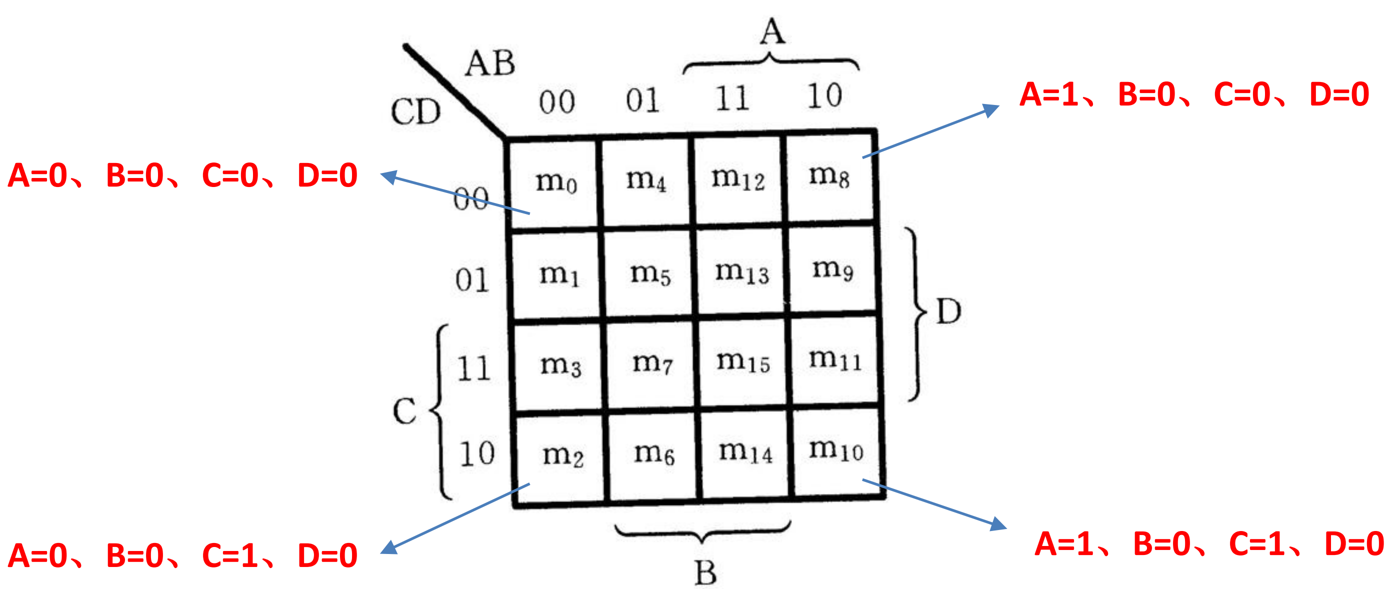


图 2.5　2～5 变量卡诺图

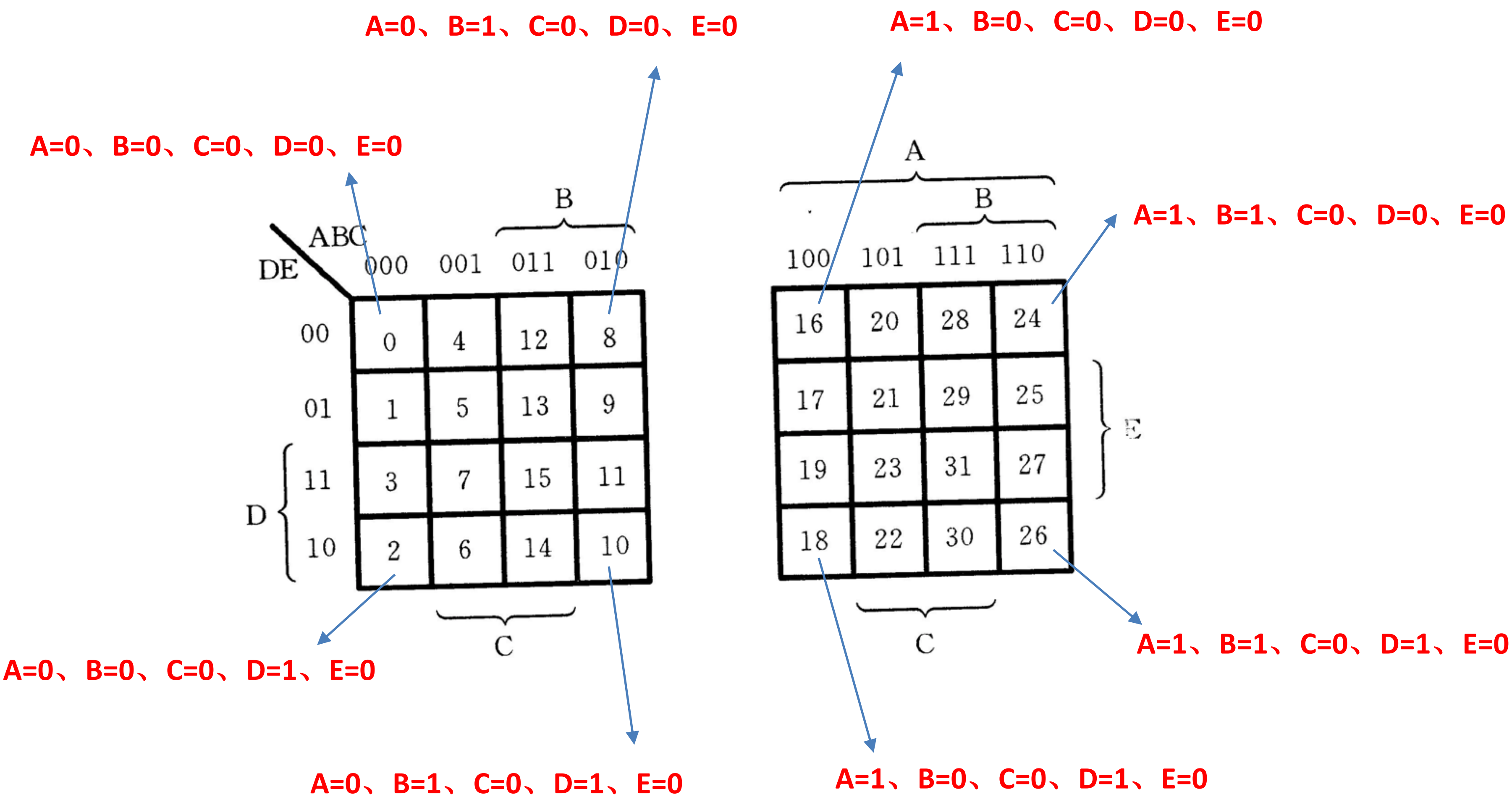
- 最小项的编号（2变量、3变量、4变量）：



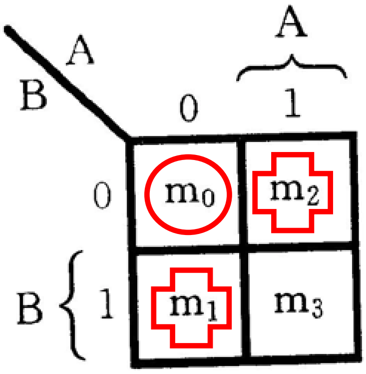




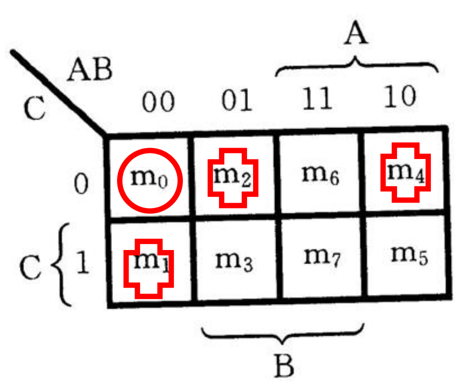
- 最小项的编号（5变量）：


A=0、B=1、C=0、D=0、E=0
A=1、B=0、C=0、D=0、E=0
A=0、B=0、C=0、D=0、E=0
A=1、B=1、C=0、D=0、E=0
A
B
B
ABC
DE
000 001 011 010
100 101 111 110
00
01
11
10
D
E
C
C
0 4 12 8 16 20 28 24
1 5 13 9 17 21 29 25
3 7 15 11 19 23 31 27
2 6 14 10 18 22 30 26
A=0、B=0、C=0、D=1、E=0
A=1、B=1、C=0、D=1、E=0
A=0、B=1、C=0、D=1、E=0
A=1、B=0、C=0、D=1、E=0

- 相邻最小项（2变量）：
  - 对于2个变量构成的最小项（/A·/B、/A·B、A·/B、A·B）：
    - /A·/B（m_0）有2个相邻的最小项（/A·B和A·/B，即m_1和m_2）
    - /A·B（m_1）有2个相邻的最小项（/A·/B、A·B，即m_0和m_3）
    - A·/B（m_2）有2个相邻的最小项（/A·/B，A·B，即m_0和m_3）
    - A·B（m_3）有2个相邻的最小项（/A·B、A·/B，即m_1和m_2）



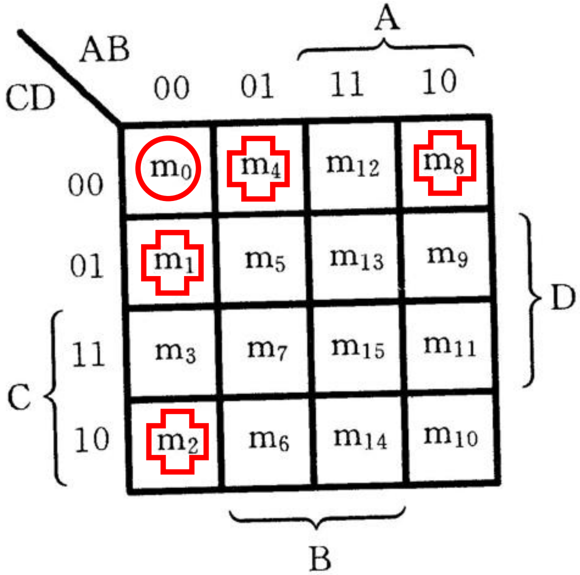
- 相邻最小项（3变量）：
  - 对于3个变量构成的最小项（/A·/B·/C，/A·/B·C，/A·B·/C，/A·B·C，A·/B·/C，A·/B·C，A·B·/C，A·B·C）：
    - /A·/B·/C（m_0）有3个相邻的最小项（/A·/B·C、/A·B·/C、A·/B·/C，即m_1、m_2和m_4）
    - /A·/B·C（m_1）有3个相邻的最小项（/A·/B·/C、/A·B·C、A·/B·C，即m_0、m_3和m_5）
    - /A·B·/C（m_2）有3个相邻的最小项（/A·/B·/C、/A·B·C、A·B·/C，即m_0、m_3和m_6）
    - /A·B·C（m_3）有3个相邻的最小项（/A·/B·C、/A·B·/C、A·B·C，即m_1、m_2和m_7）
    - A·/B·/C（m_4）有3个相邻的最小项（/A·/B·/C、A·/B·C、A·B·/C，即m_0、m_5和m_6）
    - A·/B·C（m_5）有3个相邻的最小项（/A·/B·C、A·/B·/C、A·B·C，即m_1、m_4和m_7）
    - A·B·/C（m_6）有3个相邻的最小项（/A·B·/C、A·/B·/C、A·B·C，即m_2、m_4和m_7）
    - A·B·C（m_7）有3个相邻的最小项（/A·B·C、A·/B·C、A·B·/C，即m_3、m_5和m_6）



m_4也称为相对位置的相邻最小项

- 相邻最小项（4变量）：
  - 对于4个变量构成的最小项（/A·/B·/C·/D、/A·/B·/C·D、/A·/B·C·/D、/A·/B·C·D、/A·B·/C·/D、/A·B·/C·D、/A·B·C·/D、/A·B·C·D、A·/B·/C·/D、A·/B·/C·D、A·/B·C·/D、A·/B·C·D、A·B·/C·/D、A·B·/C·D、A·B·C·/D、A·B·C·D）：
    - /A·/B·/C·/D（m_0）有4个相邻的最小项（/A·/B·/C·D、/A·/B·C·/D、/A·B·/C·/D、A·/B·/C·/D，即m_1、m_2、m_4和m_8）
    - 其余的15个最小项的4个相邻最小项请同学们自行填写！

m_2、m_8也称为相对位置的相邻最小项



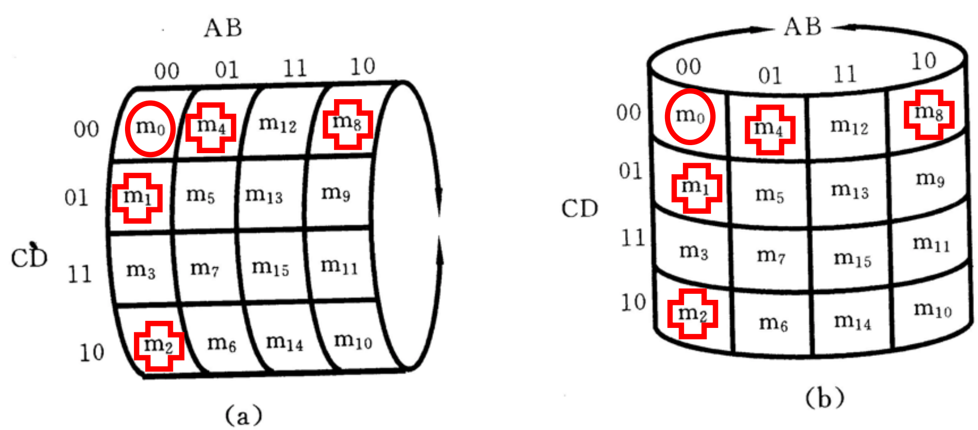


图 2.6 4 变量卡诺图卷成筒状的示意图

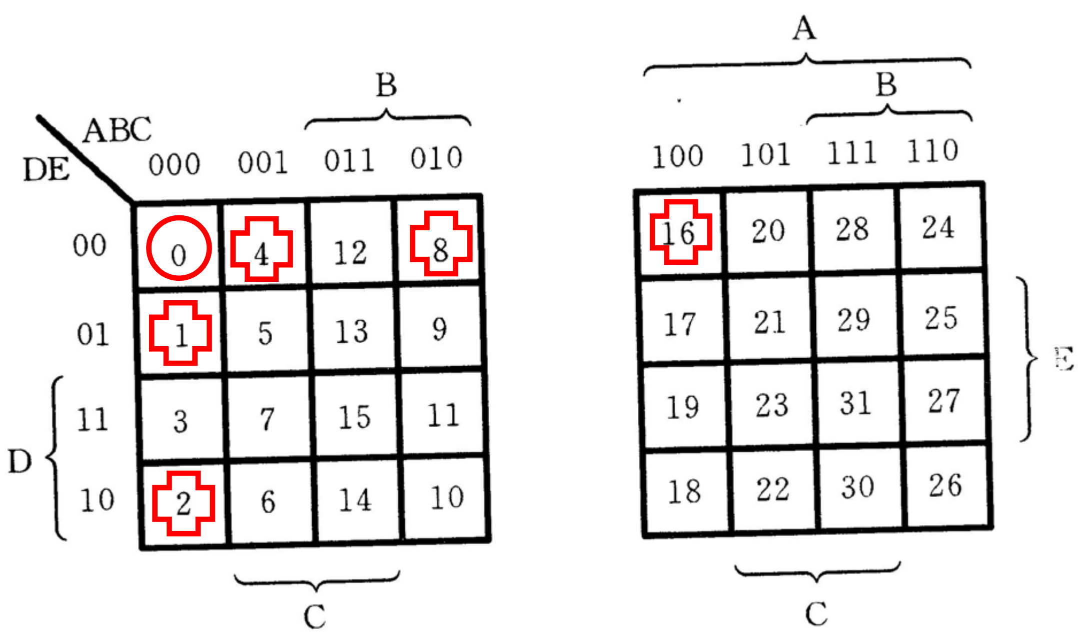
m_2、m_8也称为相对位置的相邻最小项

- 相邻最小项（5变量）：

  – 对于5个变量构成的最小项（/A·/B·/C·/D·/E、/A·/B·/C·/D·E、/A·/B·/C·D·/E、/A·/B·/C·D·E、/A·/B·C·/D·/E、/A·/B·C·/D·E、/A·/B·C·D·/E、/A·/B·C·D·E、/A·B·/C·/D·/E、/A·B·/C·/D·E、/A·B·/C·D·/E、/A·B·/C·D·E、/A·B·C·/D·/E、/A·B·C·/D·E、/A·B·C·D·/E、/A·B·C·D·E、A·/B·/C·/D·/E、A·/B·/C·/D·E、A·/B·/C·D·/E、A·/B·/C·D·E、A·/B·C·/D·/E、A·/B·C·/D·E、A·/B·C·D·/E、A·/B·C·D·E、A·B·/C·/D·/E、A·B·/C·/D·E、A·B·/C·D·/E、A·B·/C·D·E、A·B·C·/D·/E、A·B·C·/D·E、A·B·C·D·/E、A·B·C·D·E）：

    - /A·/B·/C·/D·/E（m_0）有5个相邻的最小项（/A·/B·/C·/D·E、/A·/B·/C·D·/E、/A·/B·C·/D·/E、/A·B·/C·/D·/E、A·/B·/C·/D·/E，即m_1、m_2、m_4、m_8和m_{16}）

    - 其余的31个最小项的5个相邻最小项请同学们自行填写！



m_2、m_8也称为相对位置的相邻最小项

m_{16}也称为相重位置的相邻最小项

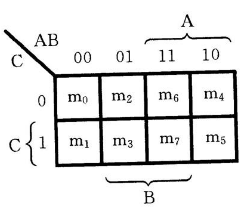
- 卡诺图的特点：
  ① n个变量的卡诺图由2^n个小方格组成，每个小方格代表一个最小项。
  ② 卡诺图上处在相邻、相对、相重位置的小方格所代表的最小项为相邻最小项。

– 2、逻辑函数在卡诺图上的表示

- 逻辑函数为标准“与-或”表达式：例如，3变量函数**F(A,B,C)=∑m(1,2,3,7)**，对应的卡诺图如图**2.7**所示（只需将每个最小项对应的位置填**1**，其余位置填**0**）。

- 逻辑函数为一般“与-或”表达式：例如，4变量函数**F(A,B,C,D) = A·B+C·D+/A·/B·C**，对应的卡诺图如图**2.8**所示。先填写**A·B**对应的**4**个**1**（红色框），再填写**C·D**对应的**4**个**1**（绿色框），最后填写**/A·/B·C**对应的**2**个**1**（紫色框）。注意：会存在重叠的部分。

- 逻辑函数为其他形式时，可将其变换为上述形式后，再填写卡诺图。



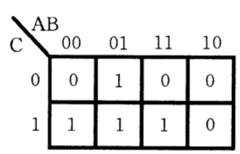


图 2.7 $F(A,B,C)=\sum m(1,2,3,7)$ 的卡诺图

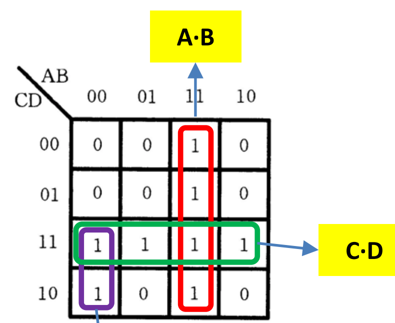


图 2.8 $F(A,B,C,D)=AB+CD+\bar{A}\bar{B}C$ 的卡诺图

– 3、卡诺图上最小项的合并规律

- （1）当卡诺图上的2个小方格几何相邻，或处于某行（列）两端时，可以合并，从而消去1个变量。例如：
  - 图2.9(a)：/A·B+A·B = B，合并后消去A
  - 图2.9(b)：/A·/B + /A·B + A·/B = /A·/B + /A·B + /A·/B + A·/B = /A + /B
  - 图2.9(c)：/A·/B·/C + /A·/B·C + A·/B·/C = /A·/B·/C + /A·/B·C + /A·/B·/C + A·/B·/C = /A·/B + /B·/C

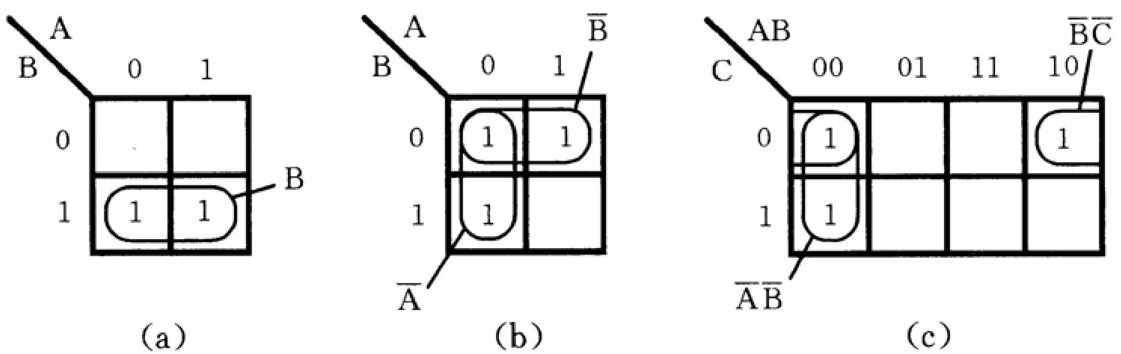


图 2.9　2、3 变量卡诺图上 2 个相邻最小项合并的典型情况

- （2）当卡诺图上的4个小方格组成1个大方格，或组成1行（列），或处于相邻两行（列）的两端，或处于四角时，可以合并，从而消去2个变量。例如：
  - 图2.10(a)：/A·/B·/C+/A·/B·C+A·/B·/C+A·/B·C=/A·/B+A·/B = /B，合并后消去A和C
  - 图2.10(b)：合并为A
  - 图2.11(a)：合并为B·D+/B·/D
  - 图2.11(b)：合并为/B·/C+A·/D
  - 图2.11(c)：合并为/A·B+C·D

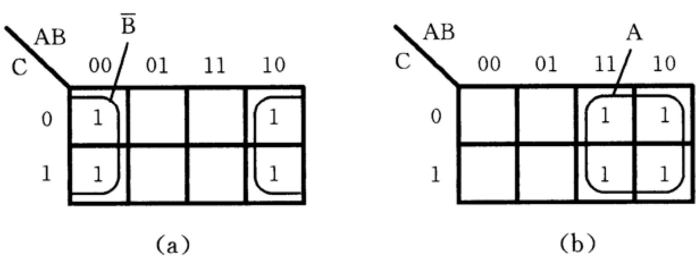


图 2.10　3 变量卡诺图上 4 个相邻最小项合并的典型情况

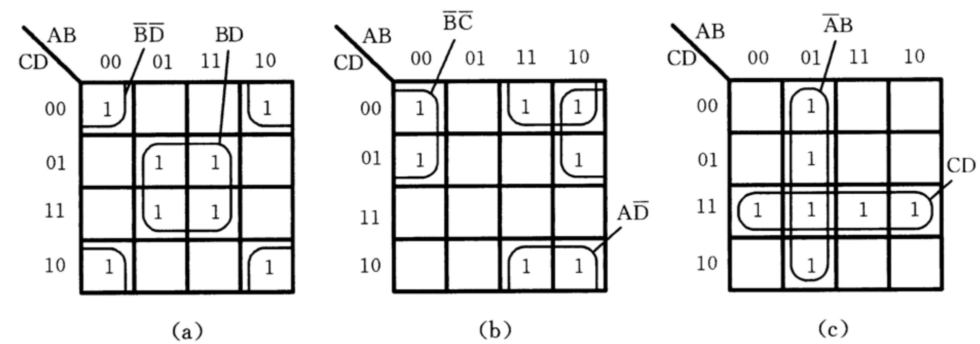


图 2.11　4 变量卡诺图上 4 个相邻最小项合并的典型情况

- （3）当卡诺图上的8个小方格组成1个大方格，或处于两个边行（列）时，可以合并，从而消去3个变量。例如：
  - 图2.12(a)：合并为1，即F(A,B,C)=1，消去A、B和C
  - 图2.12(b)：合并为/B+D

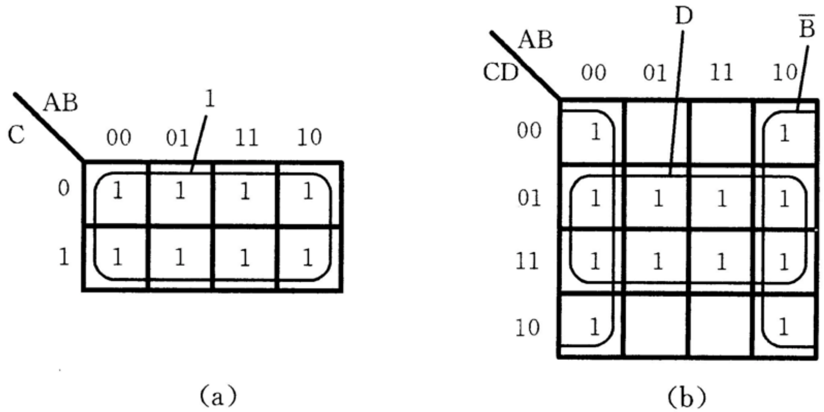


图 2.12　3、4 变量卡诺图上 8 个相邻最小项合并的典型情况

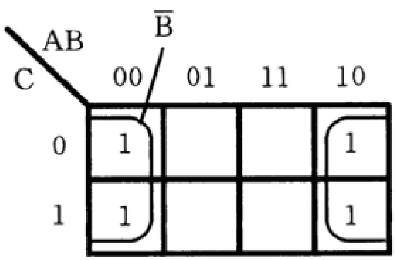
- **n个变量卡诺图中最小项的合并规律：**

  ① 卡诺圈（通常把用来包围那些能由一个简单与项代替的若干最小项的圈称为卡诺圈）中小方格的个数必须为2^m个（1个、2个、4个、8个、…），m<=n。

  ② 卡诺圈中的2^m个小方格含有m个不同变量，(n-m)个相同变量。

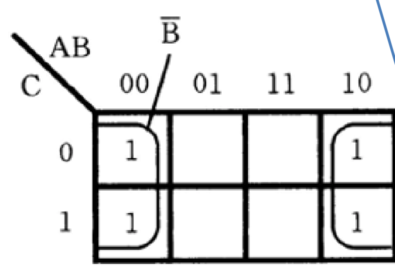
  ③ 卡诺圈中的2^m个小方格对应的最小项可用(n-m)个变量的与项表示，该与项由这些最小项中的相同部分构成。

  ④ 当m=0时，卡诺圈中包含1个最小项（2^0=1）；当m=n时，卡诺圈包围了整个卡诺图，此时F(A,B,C,...)=1。



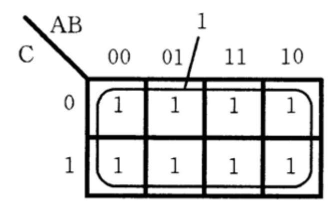
/A·/B·/C+/A·/B·C+A·/B·/C+A·/B·C=/B

n=3，m=2，4个（2^m=4）小方格含有m=2个不同变量（A、C），n-m=1个相同变量（/B）



/A·/B·/C+/A·/B·C+A·/B·/C+A·/B·C=/B

n=3，m=2，4个（2^m=4）小方格可用n-m=1个变量的与项表示（/B），该与项由这些最小项中的相同部分（/B）构成



– 4、卡诺图化简逻辑函数的步骤

- （1）几个术语
  - 蕴涵项：在函数的“与-或”表达式中，每个与项被称为该函数的蕴涵项（Implement）。例如，F(A,B,C,D) = /A·C·D + /B·C·/D + /A·B·C·/D + A·/B·/D + A·B·/C·/D，其中：/A·C·D、/B·C·/D、/A·B·C·/D、A·/B·/D、A·B·/C·/D称为该函数的蕴涵项。
  - 质蕴涵项：若函数的一个蕴涵项不是该函数中其他蕴涵项的子集，则此蕴涵项被称为质蕴涵项（Prime Implement），简称质项。例如，F(A,B,C,D)=/A·C+A·/C·/D+A·/B·/D，其中：/A·C、A·/C·/D、A·/B·/D也称为质蕴涵项。
  - 必要质蕴涵项：若函数的一个质蕴涵项包含有不被函数的其他任何质蕴涵项所包含的最小项，则此质蕴涵项被称为必要质蕴涵项（Essential Prime Implement），简称必要质项。

- （2）求逻辑函数的最简“与-或”表达式
  - 用卡诺图求逻辑函数最简“与-或”表达式的一般步骤：
    - » 第一步：画出函数的卡诺图。
    - » 第二步：在卡诺图上圈出函数的全部质蕴涵项。
    - » 第三步：从全部质蕴涵项中找出所有必要质蕴涵项。
    - » 第四步：若函数的所有必要质蕴涵项，尚不能覆盖卡诺图上的所有1方格，则从剩余质蕴涵项中找出最简的所需质蕴涵项，使它和必要质蕴涵项一起构成函数的最小覆盖（即最简的质蕴涵项集）。

– 例2.10：化简F(A,B,C,D) = $\sum m(0,3,5,6,7,10,11,13,15)$

蕴涵项：9个

– 解：

- 第一步：画出函数F的卡诺图，如图2.13(a)。

- 第二步：在卡诺图上圈出函数的全部质蕴涵项（卡诺圈），如图2.13(b)；先画2个大的卡诺圈（包含4个小方格），再画2个小卡诺圈（包含2个小方格），剩余1个左上角的最小卡诺圈（包含1个小方格）；共有5个卡诺圈（5个质蕴涵项）。

- 第三步：从全部质蕴涵项中找出所有必要质蕴涵项，如图2.13(c)；图中标有“*”的最小项（小方格）称为必要最小项；包含必要最小项的质蕴涵项称为必要质蕴涵项；共有5个必要质蕴涵项。

- 因为5个必要质蕴涵项覆盖了卡诺图上的所有1方格，因此： F(A,B,C,D) = /A·/B·/C·/D + /A·B·C + A·/B·C + B·D + C·D。F(A,B,C,D)由9项简化为5项。

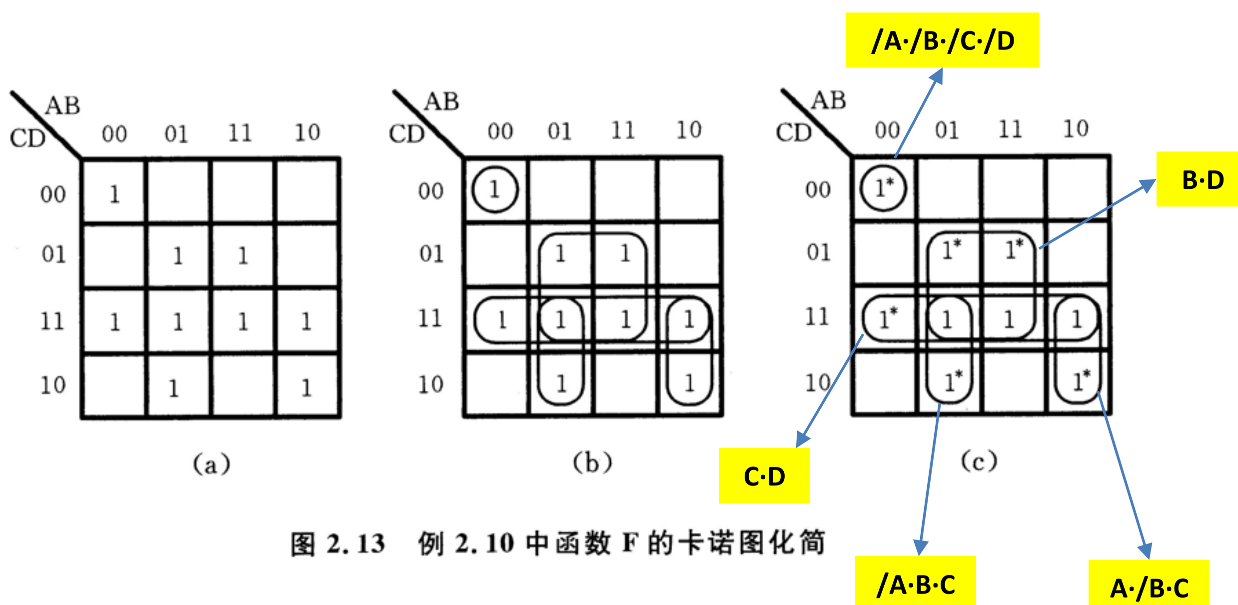


图 2.13 例 2.10 中函数 F 的卡诺图化简

– 例2.11：化简F(A,B,C,D) = /A·C·D + /B·C·/D + /A·B·C·/D + A·/B·/D + A·B·/C·/D

蕴涵项：5个

– 解：

- 第一步：画出函数F的卡诺图，如图2.14(a)。
- 第二步：在卡诺图上圈出函数的全部质蕴涵项（卡诺圈），如图2.14(b)；先画1个大的卡诺圈（包含4个小方格），再画3个小卡诺圈（包含2个小方格）；共有4个卡诺圈（4个质蕴涵项）。
- 第三步：找出所有必要质蕴涵项，如图2.14(c)；图中标有“*”的最小项（小方格）称为必要最小项；包含必要最小项的质蕴涵项称为必要质蕴涵项；共有2个必要质蕴涵项（红框），这2个必要质蕴涵项不能覆盖卡诺图上的所有1方格。
- 第四步：确定除必要质蕴涵项外的最简质蕴涵项（1个，有2种形式：A·/B·/D或/B·C·/D）；求出最简蕴涵项集（最小覆盖：2个必要质蕴涵项+1个最简质蕴涵项）。
- 因此：F(A,B,C,D) = /A·C+A·/C·/D+A·/B·/D；或者：F(A,B,C,D) = /A·C+A·/C·/D+/B·C·/D；可见函数的最简“与-或”表达式不是唯一的。F(A,B,C,D)由5项简化为3项。

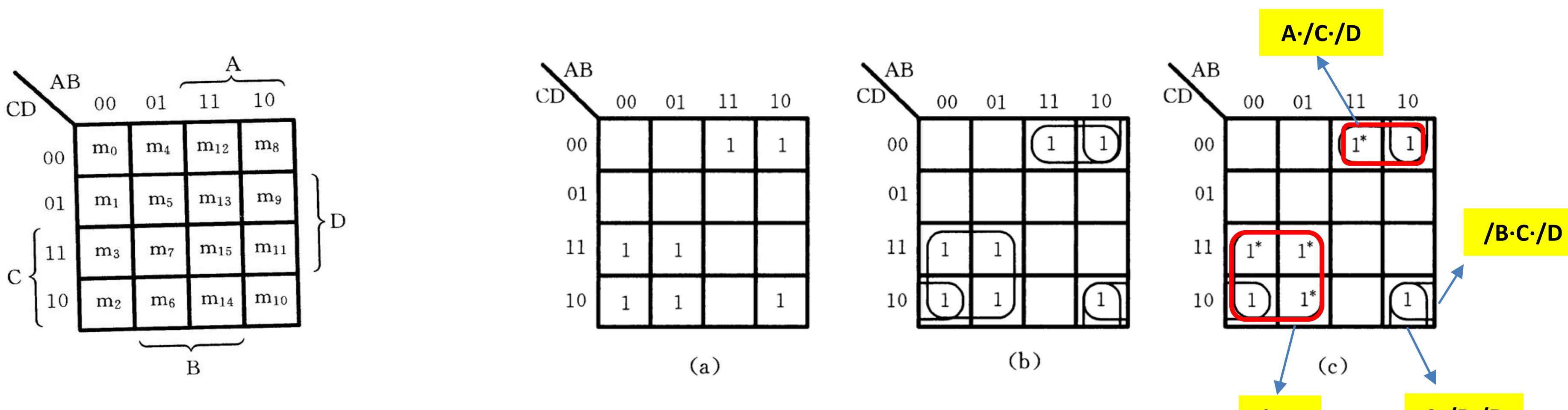
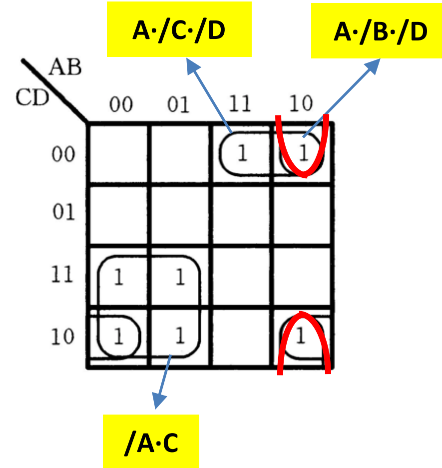
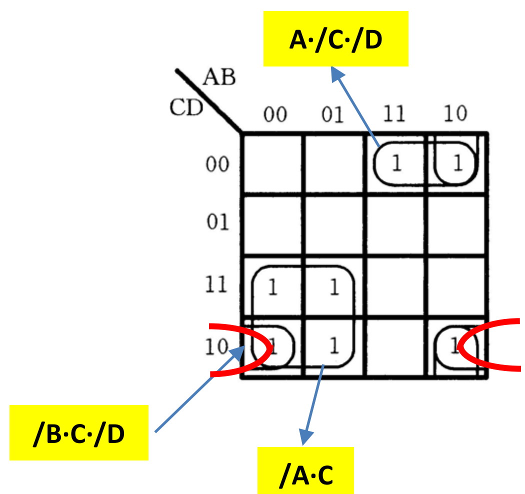


图 2.14 例 2.11 中函数 F 的卡诺图化简



F(A,B,C,D) = /A·C+A·/C·/D+A·/B·/D



F(A,B,C,D) = /A·C+A·/C·/D+/B·C·/D

– 卡诺图法化简的**总原则**：在覆盖函数中的所有最小项（标有1的小方格）的提取下，卡诺圈的个数达到最少，每个卡诺圈达到最大。

- （3）求逻辑函数最简“或-与”表达式

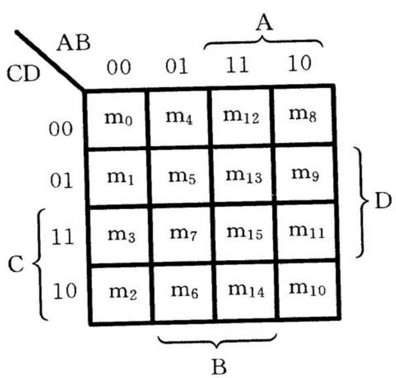
  – ①给定的逻辑函数为“与-或”表达式

    - 例2.12：用卡诺图化简逻辑函数F(A,B,C,D) = A·/C + A·D + /B·/C + /B·D的最简“或-与”表达式。
    - 解：
      – 第一步：画出卡诺图，如图2.15；先根据逻辑函数F的表达式标出1小方格，再标出0小方格。

      – 第二步：合并卡诺图中的0小方格（卡诺圈），如图2.15。

      – 第三步：根据卡诺圈，得到反函数/F的最简“与-或”表达式：/F(A,B,C,D) = /A·B + C·/D。

      – 第四步：对/F取反，即可得到F的最简“或-与”表达式：F(A,B,C,D) = (A+/B)·(/C+D)。



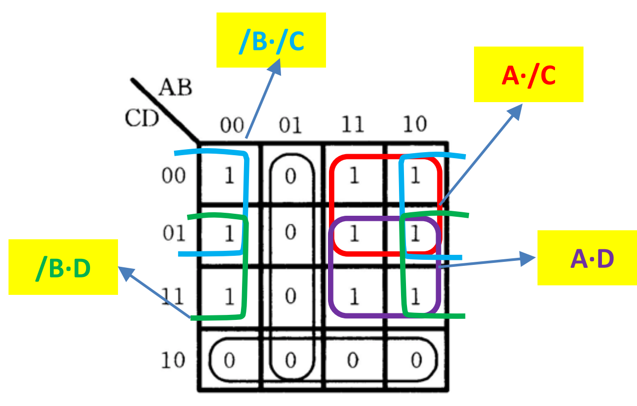


图 2.15 例 2.12 中函数 F 的卡诺图

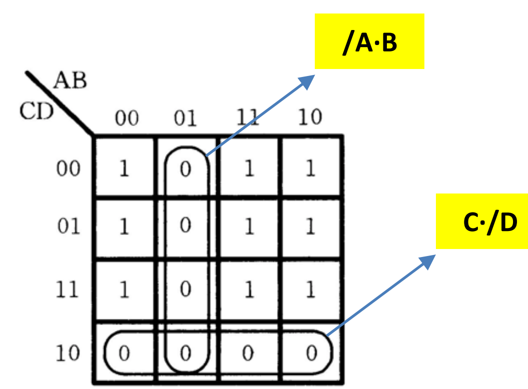
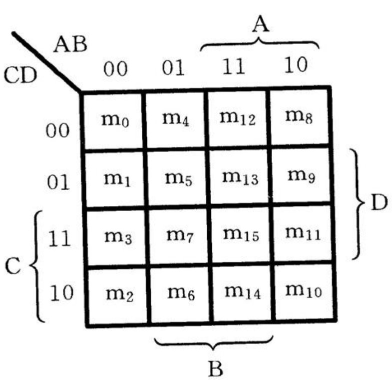


图 2.15 例 2.12 中函数 F 的卡诺图

– ②给定的逻辑函数为“或-与”表达式

- 例2.13：用卡诺图化简逻辑函数F(A,B,C,D) = (/A+D)·(B+/D)·(A+B)的最简“或-与”表达式。
- 解：
  - 第一步：根据反演规则，求F的反函数：/F(A,B,C,D) = A·/D + /B·D + /A·/B。
  - 第二步：画出/F的卡诺图，如图2.16。
  - 第三步：合并卡诺图中的1小方格（卡诺圈），如图2.16。
  - 第四步：根据卡诺圈，得到反函数/F的最简“与-或”表达式：/F(A,B,C,D) = /B + A·/D。
  - 第五步：对/F取反，即可得到F的最简“或-与”表达式：F(A,B,C,D) = B·(/A+D)。



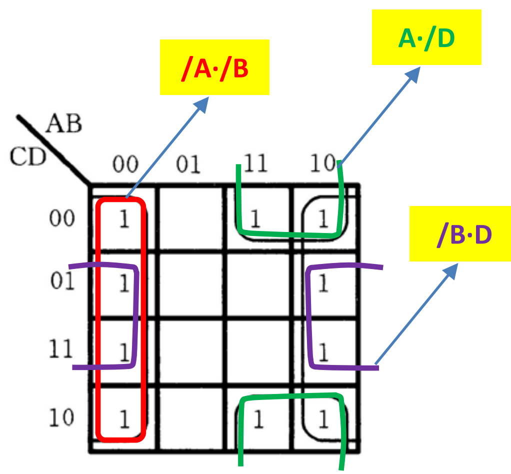


图 2.16　例 2.13 中反函数 $\bar{F}$ 的卡诺图

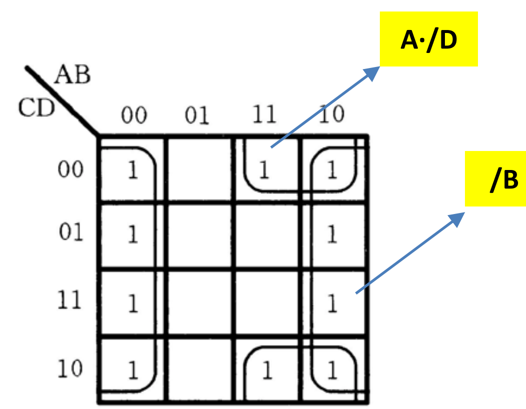
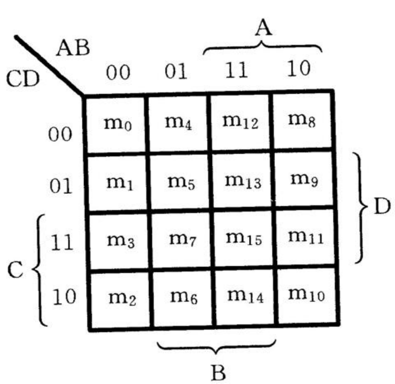


图 2.16　例 2.13 中反函数 $\bar{F}$ 的卡诺图

- 例2.14：用卡诺图化简逻辑函数$F(A,B,C,D) = \prod M(3,4,6,7,11,12,13,14,15)$的最简“与-或”表达式和最简“或-与”表达式。
- 解：
  - 第一步：画出F的卡诺图，如图2.17(a)；F为标准“或-与”表达式，对应的最大项位置填0（M_3、M_4、M_6、M_7、M_{11}、M_{12}、M_{13}、M_{14}、M_{15}），其他位置填1。
  - 第二步：合并卡诺图中的1小方格（卡诺圈），如图2.17(b)。
  - 第三步：根据卡诺圈，得函数F的最简“与-或”表达式：F(A,B,C,D) = /B·/D+/B·/C+/A·/C·D。
  - 第四步：合并卡诺图中的0小方格（卡诺圈），如图2.17(c)。
  - 第五步：根据卡诺圈，得到函数/F的最简“与-或”表达式：/F(A,B,C,D) = A·B+C·D+B·/D。
  - 第六步：对/F取反，即可得到函数F的最简“或-与”表达式：F(A,B,C,D) = (/A+/B)·(/C+/D)·(/B+D)
  。



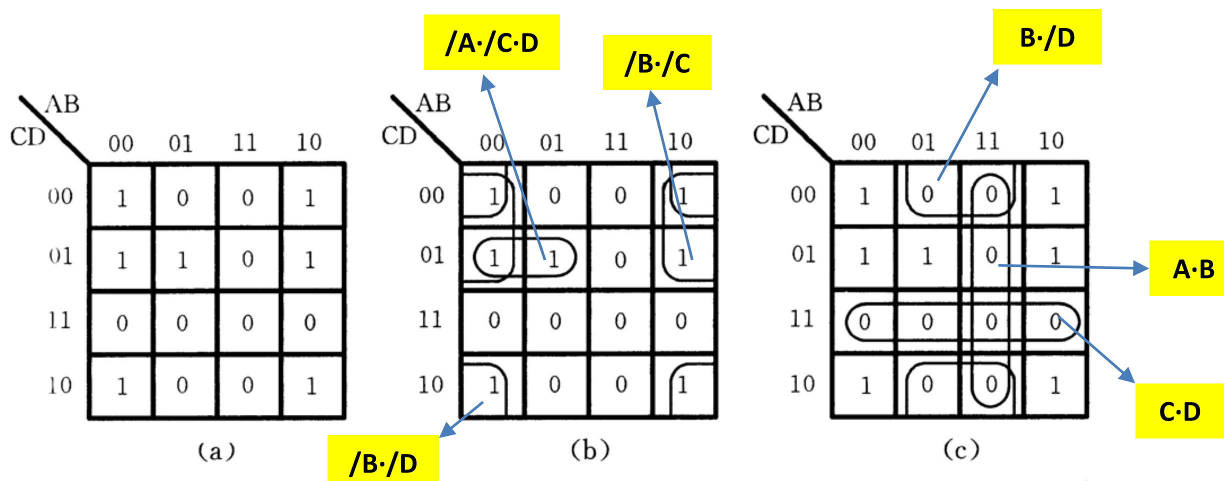


图 2.17　例 2.14 中函数 F 的卡诺图

• 2.4.3 列表化简法

– 如果逻辑函数的变量个数大于4，采用卡诺图化简法就非常困难，此时可以采用列表化简法。

– 列表化简法也称为奎恩-麦克拉斯基（Quine-McCluskey）法，是一种系统化简法，简称Q-M化简法。

– 列表化简法包括以下4步：

  - 第一步：将函数表示成“最小项之和”（标准“与-或”表达式）的形式，并用二进制码表示每一个最小项。
  - 第二步：找出函数的全部质蕴涵项。
  - 第三步：找出函数的必要质蕴涵项。
  - 第四步：找出函数的最小覆盖。

- 例2.15：用列表法化简逻辑函数$F(A,B,C,D)=\sum m(0,5,7,8,9,10,11,14,15)$ 。
- 解：
  - 第一步：用二进制码表示函数F的每一个最小项，如表2.8。

表 2.8　例 2.15 中函数 F 的部分真值表

项号	A	B	C	D	F	项号	A	B	C	D	F
0	0	0	0	0	1	10	1	0	1	0	1
5	0	1	0	1	1	11	1	0	1	1	1
7	0	1	1	1	1	14	1	1	1	0	1
8	1	0	0	0	1	15	1	1	1	1	1
9	1	0	0	1	1						

  - 第二步：找出函数F的全部质蕴涵项。
    - （1）将表2.8的9个最小项重新排列：4个变量全部为0的编为第0组，4个变量只有1个为1的编为第1组，4个变量有2个为1的编为第2组，4个变量有3个为1的编为第3组，4个变量全部为1的编为第4组；见表2.9的“(I)最小项”。

表 2.9　质蕴涵项产生表

(Ⅰ)最小项				(Ⅱ)$(n-1)$个变量的与项				(Ⅲ)$(n-2)$个变量的与项			
组号	m_i	ABCD	p_i	组号	$\sum m_i$	ABCD	p_i	组号	$\sum m_i$	ABCD	p_i
0	0	0000	√	0	0,8	—000	p_5	1	8,9,10,11	10——	p_2
1	8	1000	√	1	8,9	100—	√	2	10,11,14,15	1—1—	p_1
2	5	0101	√		8,10	10—0	√				
	9	1001	√	2	5,7	01—1	p_4				
	10	1010	√		9,11	10—1	√				
3	7	0111	√		10,11	101—	√				
	11	1011	√		10,14	1—10	√				
	14	1110	√	3	7,15	—111	p_3				
4	15	1111	√		11,15	1—11	√				
					14,15	111—	√				

» （2）比较“(I)最小项”中相邻组号中的两个最小项，如果只有1位是不同的，则进行合并，得到3个变量的与项（n-1=3）：

- 组0和组1是相邻组号，两个最小项m_0和m_8只有1位是不同的，将其合并为“—000”。
- 组1和组2是相邻组号，两个最小项m_8和m_5有3位是不同的，不合并；两个最小项m_8和m_9只有1位是不同的，将其合并为“100—”；两个最小项m_8和m_{10}只有1位是不同的，将其合并为“10—0”。
- 组2和组3是相邻组号，m_5和m_7、m_9和m_{11}、m_{10}和m_{11}、m_{10}和m_{14}都只有1位是不同的，将其分别合并为“01—1”、“10—1”、“101—”、“1—10”；m_5和m_{11}、m_5和m_{14}、m_9和m_7、m_9和m_{14}、m_{10}和m_7都有多位是不同的，不合并。
- 组3和组4是相邻组号，m_7和m_{15}、m_{11}和m_{15}、m_{14}和m_{15}都只有1位是不同的，将其分别合并为“—111”、“1—11”、“111—”。
- 将合并后的10个3变量的与项进行排列：3个都是0的编为第0组，2个0、1个1的编为第1组，1个0、2个1的编为第2组，3个都是1的编为第3组；见表2.9的“(II)(n-1)个变量的与项”；同时，对9个最小项进行标注，因为9个最小项都参与了合并，因此，表2.9中“(I)最小项”的p_i列全部打“√”。

表 2.9 质蕴涵项产生表

(Ⅰ)最小项				(Ⅱ)(n−1)个变量的与项				(Ⅲ)(n−2)个变量的与项			
组号	m_i	ABCD	p_i	组号	$\sum m_i$	ABCD	p_i	组号	$\sum m_i$	ABCD	p_i
0	0	0000	√	0	0,8	—000	p_5	1	8,9,10,11	10——	p_2
1	8	1000	√	1	8,9	100—	√	2	10,11,14,15	1—1—	p_1
					8,10	10—0	√				
2	5	0101	√	2	5,7	01—1	p_4				
	9	1001	√		9,11	10—1	√				
	10	1010	√		10,11	101—	√				
					10,14	1—10	√				
3	7	0111	√	3	7,15	—111	p_3				
	11	1011	√		11,15	1—11	√				
	14	1110	√		14,15	111—	√				
4	15	1111	√								

» （3）比较“(II)(n-1)个变量的与项”中相邻组号中的两个与项，如果只有1位是不同的，则进行合并，得到2个变量的与项（n-2=2）：

- 组0和组1是相邻组号，“—000”和“100—”、“—000”和“10—0”都有多位是不同的，不合并。

- 组1和组2是相邻组号，“100—”和“101—”、“10—0”和“10—1”都只有1位是不同的，将其合并为“10——”；其余，“100—”和“01—1”、“100—”和“10—1”、“100—”和“1—10”、“10—0”和“01—1”、“10—0”和“101—”、“10—0”和“1—10”都有多位是不同的，不合并。

- 组2和组3是相邻组号，“101—”和“111—”、“1—10”和“1—11”都只有1位是不同的，将其合并为“1—1—”；其余，“01—1”和“—111”、“01—1”和“1—11”、“01—1”和“111—”、“10—1”和“—111”、“10—1”和“1—11”、“10—1”和“111—”、“101—”和“—111”、“101—”和“1—11”、“1—10”和“—111”、“1—10”和“111—”都有多位是不同的，不合并。

- 将合并后的2个2变量的与项进行排列：有1个1、1个0的编为第1组，有2个1的编为第2组；见表2.9的“(III)(n-2)个变量的与项”；同时，对10个3变量的与项进行标注，因为7个3变量的与项参与了合并，因此，表2.9中“(II)(n-1)个变量的与项”的p_i列的7个项打“√”。

表 2.9 质蕴涵项产生表

(Ⅰ)最小项				(Ⅱ)(n－1)个变量的与项				(Ⅲ)(n－2)个变量的与项			
组号	m_i	ABCD	p_i	组号	Σm_i	ABCD	p_i	组号	Σm_i	ABCD	p_i
0	0	0000	√	0	0,8	—000	p_5	1	8,9,10,11	10——	p_2
1	8	1000	√	1	8,9	100—	√	2	10,11,14,15	1—1—	p_1
2	5	0101	√		8,10	10—0	√				
	9	1001	√	2	5,7	01—1	p_4				
	10	1010	√		9,11	10—1	√				
3	7	0111	√		10,11	101—	√				
	11	1011	√		10,14	1—10	√				
	14	1110	√	3	7,15	—111	p_3				
4	15	1111	√		11,15	1—11	√				
					14,15	111—	√				

» （4）比较“(III)(n-2)个变量的与项”中相邻组号中的两个与项，“10——”和“1—1—”有多位是不同的，无法合并。

» （5）表2.9中没有参与合并的“最小项”、“(n-1)个变量的与项”、“(n-2)个变量的与项”，即为全部质蕴涵项（即没有打“√”的），分别是：

- $p_1 = \sum m(10,11,14,15)$ = “1—1—” = A·C
- $p_2 = \sum m(8,9,10,11)$ = “10——” = A·/B
- $p_3 = \sum m(7,15)$ = “—111” = B·C·D
- $p_4 = \sum m(5,7)$ = “01—1” = /A·B·D
- $p_5 = \sum m(0,8)$ = “—000” = /B·/C·/D

表 2.9　质蕴涵项产生表

(Ⅰ)最小项				(Ⅱ)(n－1)个变量的与项				(Ⅲ)(n－2)个变量的与项			
组号	m_i	ABCD	p_i	组号	$\sum m_i$	ABCD	p_i	组号	$\sum m_i$	ABCD	p_i
0	0	0000	✓	0	0,8	—000	p_5	1	8,9,10,11	10——	p_2
1	8	1000	✓	1	8,9	100—	✓	2	10,11,14,15	1—1—	p_1
2	5	0101	✓		8,10	10—0	✓				
	9	1001	✓	2	5,7	01—1	p_4				
	10	1010	✓		9,11	10—1	✓				
3	7	0111	✓		10,11	101—	✓				
	11	1011	✓		10,14	1—10	✓				
	14	1110	✓	3	7,15	—111	p_3				
4	15	1111	✓		11,15	1—11	✓				
					14,15	111—	✓				

– 第三步：找出函数F的全部必要质蕴涵项。

» 建立必要质蕴涵项产生表，如表2.10。

» 表2.10中，第2行为F的9个最小项，$F(A,B,C,D) = \sum m(0,5,7,8,9,10,11,14,15)$

» 第3～7行中的“x”对应$p_1 \sim p_5$中包含最小项的情况：

- $p_1 = \sum m(10,11,14,15)$
- $p_2 = \sum m(8,9,10,11)$
- $p_3 = \sum m(7,15)$
- $p_4 = \sum m(5,7)$
- $p_5 = \sum m(0,8)$

» 依次检查表2.10中的第2～10列中的“x”，如果该列只有1个“x”，则标注为“x外加圆圈”，共有4个，其对应的行（p_1^*、p_2^*、p_4^*、p_5^*）即为必要质蕴涵项，如表2.10。

» 检查$p_1^* = \sum m(10,11,14,15)$、$p_2^* = \sum m(8,9,10,11)$、$p_4^* = \sum m(5,7)$、$p_5^* = \sum m(0,8)$覆盖最小项的情况，将覆盖的情况在表2.10的最后一行进行标注（打√），全部覆盖。

表 2.10　必要质蕴涵项产生表

p_i	m_i								
	0	5	7	8	9	10	11	14	15
p_1^*						×	×	⊗	×
p_2^*				×	⊗	×	×		
p_3			×						×
p_4^*		⊗	×						
p_5^*	⊗			×					
覆盖情况	√	√	√	√	√	√	√	√	√

– 第四步：找出函数F的最小覆盖。

» 由表2.10可知，必要质蕴涵项（p_1^*、p_2^*、p_4^*、p_5^*）覆盖了全部最小项；因此，函数F最终简化为：

- $F = p_1^* + p_2^* + p_4^* + p_5^* = A \cdot C + A \cdot /B + /A \cdot B \cdot D + /B \cdot /C \cdot /D$

- 如果上述步骤三中的必要质蕴涵项没有覆盖全部的最小项，则需要从中找出最小覆盖。

- 例2.16：已知函数$F(A,B,C,D) = \sum m(0,3,4,5,6,7,8,10,11)$的必要质蕴涵项产生表如表2.11所示，求出该函数的最小覆盖 。
- 解：
    - 根据表2.11可知，必要质蕴涵项为p_1^*，但是p_1^*只覆盖了最小项m_4、m_5、m_6、m_7，没有覆盖全部的最小项。
    - 从表2.11中去掉p_1^*行，以及其对应的m_4、m_5、m_6、m_7列，得到表2.12（所需质蕴涵项产生表）。
    - 分析表2.12，p_4行完全包含在p_3行中，p_7行完全包含在p_6行中，根据行消去规则，可以去掉p_4和p_7行，得到表2.13（消去多余行后的所需质蕴涵项产生表）。注意：表2.12中没有可以消去的列。

行消去规则：对于所需质蕴涵项产生表中的任意质蕴涵项p_i和p_j，若p_i行中的“x”号完成包含在pj行中，即$p_i \in p_j$，则可消去p_i行。这是因为选取了质蕴涵项p_j后，不仅可以覆盖质蕴涵项p_i所能覆盖的最小项，而且还可以覆盖其他最小项。

表 2.11 必要质蕴涵项产生表

p_i	m_i								
	0	3	4	5	6	7	8	10	11
p_1^*			×	⊗	⊗	×			
p_2								×	×
p_3		×							×
p_4		×				×			
p_5							×	×	
p_6	×						×		
p_7	×		×						
覆盖情况			✓	✓	✓	✓			

表 2.12 所需质蕴涵项产生表

p_i	m_i				
	0	3	8	10	11
p_2				×	×
p_3		×			×
p_4	—	×	—	—	—
p_5			×	×	
p_6	×		×		
p_7	-×-	—	—	—	—

表 2.13 消去多余行后的所需质蕴涵项产生表

p_i	m_i				
	0	3	8	10	11
p_2			|	×	×
p_3		×	|		×
p_5			×	×	|
p_6	×		×		|

- 进一步分析表2.13，m_0列完全包含在m_8列中，m_3列完全包含在m_{11}列中，根据列消去规则，可以去掉m_8列和m_{11}列，得到表2.14（消去多余行和多余列后的所需质蕴涵项产生表）。

- 由表2.14可见， m_0和m_3所对应的列，均只有一个“x”号，因此，p_3^{**}和p_6^{**}是必须选取的质蕴涵项，也称为二次质蕴涵项。

- 此时，还剩下最小项m_{10}未被覆盖，可以选取p_2或p_5作为所需质蕴涵项（注：p_2和p_5的复杂度是一样的）。

- 因此，函数F的最小覆盖为：$p_1 + p_3 + p_6 + p_2$，或者， $p_1 + p_3 + p_6 + p_5$

- 即，函数F可以化简为：$F= p_1 + p_2 + p_3 + p_6 = /A·B + A·/B·C + /B·C·D + /B·/C·/D$；或者，$F= p_1 + p_3 + p_5 + p_6 = /A·B + /B·C·D + A·/B·/D + /B·/C·/D$；具体见下页。

列消去规则：对于所需质蕴涵项产生表中的任意最小项m_i和m_j，若m_i列中的“x”号完成包含在m_j列中，即$m_i \in m_j$，则可消去m_j列。这是因为选取了覆盖m_i的质蕴涵项后，一定能覆盖m_j，反之则不一定。

表 2.13 消去多余行后的所需质蕴涵项产生表

p_i	m_i				
	0	3	8	10	11
p_2			|	×	×
p_3		×	|		×
p_5			×	×	|
p_6	×		×		|

表 2.14 消去多余行和多余列后的所需质蕴涵项产生表

p_i	m_i		
	0	3	10
p_2			×
p_3^{**}		⊗	
p_5			×
p_6^{**}	⊗		

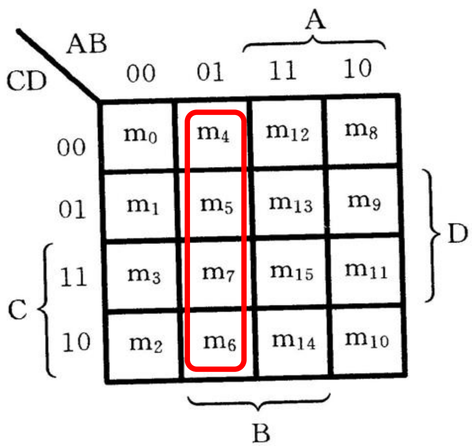
表 2.11 必要质蕴涵项产生表

p_i	m_i								
	0	3	4	5	6	7	8	10	11
p_1^*			×	⊗	⊗	×			
p_2								×	×
p_3		×							×
p_4		×				×			
p_5							×	×	
p_6	×						×		
p_7	×		×						
覆盖情况			✓	✓	✓	✓			

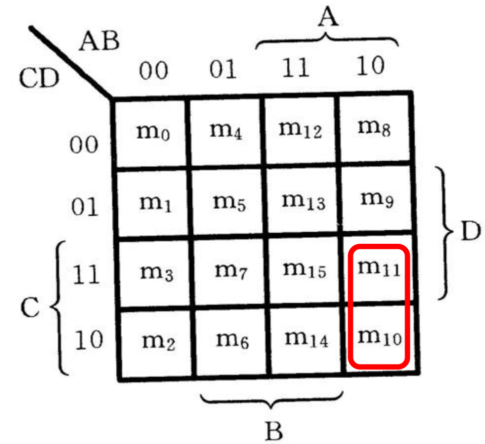
表 2.11　必要质蕴涵项产生表

p_i	m_i 0	3	4	5	6	7	8	10	11
$p_1{}^*$			×	⊗	⊗	×			
p_2								×	×
p_3		×							×
p_4		×				×			
p_5							×	×	
p_6	×						×		
p_7	×		×						
覆盖情况			✓	✓	✓	✓			

p_1 = /A·B

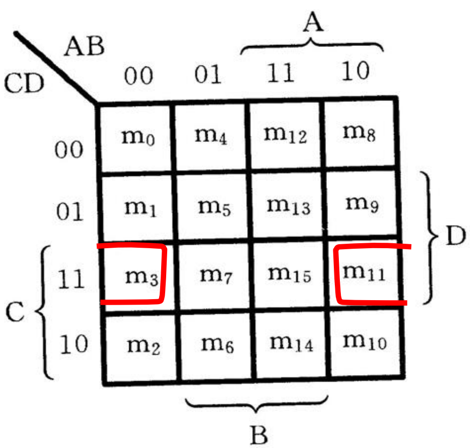


p_2 = A·/B·C

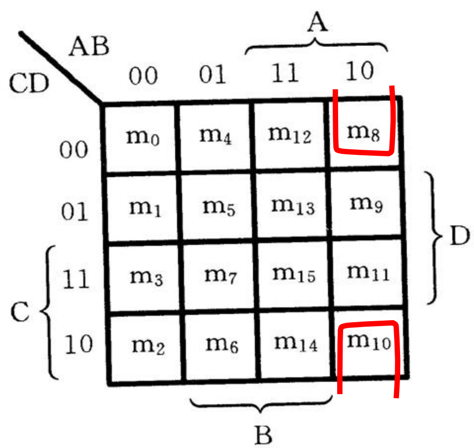


F= p_1 + p_2 + p_3 + p_6 = /A·B + A·/B·C + /B·C·D + /B·/C·/D

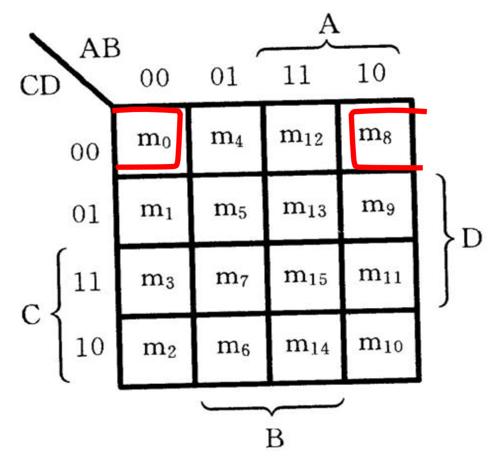
F= p_1 + p_3 + p_5 + p_6 = /A·B + /B·C·D + A·/B·/D + /B·/C·/D



p_3 = /B·C·D



p_5 = A·/B·/D



p_6 = /B·/C·/D

本章小结

- 逻辑代数也称布尔代数、开关代数。

- 逻辑代数的5个公理（交换律、结合律、分配律、0-1律、互补律）。

- 3种基本的逻辑运算：或、与、非。

- 逻辑函数的3种表示方法：
  ① 逻辑表达式：例如，F(A,B) = /A·B + A·/B

  ② 真值表

  ③ 卡诺图

表 2.4　函数 $F=A\bar{B}+\bar{A}C$ 的真值表

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

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

图 2.8　$F(A,B,C,D)=AB+CD+\bar{A}\bar{B}C$ 的卡诺图

- 逻辑代数的8个定理（0-1律、重叠律、吸收律、消除律、对合律、互补律、并项律、包含律）。

也称摩根定理

- 逻辑代数的重要规则：
  - 1、代入规则：任何一个含有变量A的逻辑等式，如果将所有出现A的位置都代之以同一个逻辑函数F，则等式仍然成立。
    - 例如，给定逻辑等式：A·(B+C) = A·B+A·C，若等式中的C都用(C+D)代替，则该等式仍然成立，即：A·[B+(C+D)] = A·B + A·(C+D)
  - 2、反演规则：如果将逻辑函数F表达式中所有的“·”变成“+”，“+”变成“·”，“0”变成“1”，“1”变成“0”，原变量（A）变成反变量（/A），反变量（/A）变成原变量（A），并保持原函数中的运算顺序不变，则所得到的新的函数为原函数F的反函数/F。
    - 例如，F = /A·B+C·/D，则：/F = (A+/B)·(/C+D)　　如何由F，写出/F？
  - 3、对偶规则：如果将逻辑函数F表达式中所有的“·”变成“+”，“+”变成“·”，“0”变成“1”，“1”变成“0”，并保持原函数中的运算顺序不变，则所得到的新的逻辑表达式称为函数F的对偶式，记作F’。
    - 例如：F=/A·B+/B·(C+0)，则：F’=(/A+B)·(/B+C·1)　　如何由F，写出F’？

- 复合逻辑：
  - 1、与非门：F = /(A·B)
  - 2、或非门：F = /(A+B)
  - 3、与或非门：F = /(A·B + C·D)
  - 4、异或门：$F = A \oplus B$ = /A·B + A·/B
  - 5、同或门（异或非门）：$F = A \odot B$ = /A·/B + A·B = /($A \oplus B$)

- 逻辑函数表达式的基本形式：
  - 1、“与-或”表达式：F(A,B,C) = /A·B + A·/B·C + /C
  - 2、“或-与”表达式：F(A,B,C,D) = (/A + B) · (B + /C) · (A + /B + C) · /D

- 最小项（m_i）：/A·/B·/C， /A·/B·C， /A·B·/C， /A·B·C， A·/B·/C， A·/B·C， A·B·/C， A·B·C
  - 最小项的4个性质。

- 最大项（M_i）：A+B+C，A+B+/C，A+/B+C，A+/B+/C，/A+B+C，/A+B+/C，/A+/B+C，/A+/B+/C
  - 最大项的4个性质。

- 最小项与最大项的关系：/m_i = M_i，或者，m_i = /M_i。

- 逻辑函数表达式的标准形式：
  - 1、标准“与-或”表达式：F(A,B,C) = /A·/B·C + /A·B·/C + A·/B·/C + A·B·C = m1+m2+m4+m7 =∑m(1,2,4,7)
  - 2、标准“或-与”表达式：F(A,B,C) = (A+B+C) · (/A+B+/C) · (/A+/B+/C) = M0·M5·M7 =∏M(0,5,7)

- 将一个任意逻辑函数表达式转换成标准表达式有两种常用方法：
  - 1、代数转换法（转换起来比较麻烦）
  - 2、真值表转换法（转换起来比较方便）

- # 逻辑函数化简：
  - **1、代数化简法（简化起来比较麻烦）**
    - 最简“与-或”表达式应满足两个条件：
      - ① 表达式中的与项个数最少；
      - ② 在满足条件①的前提下，每个与项中的变量个数最少。
    - 最简“或-与”表达式应满足两个条件：
      - ① 表达式中的或项个数最少；
      - ② 在满足条件①的前提下，每个或项中的变量个数最少。
  - **2、卡诺图化简法（简化起来比较方便）**
    - 第一步：画出函数的卡诺图。
    - 第二步：在卡诺图上圈出函数的全部质蕴涵项。
    - 第三步：从全部质蕴涵项中找出所有必要质蕴涵项。
    - 第四步：若函数的所有必要质蕴涵项，尚不能覆盖卡诺图上的所有1方格，则从剩余质蕴涵项中找出最简的所需质蕴涵项，使它和必要质蕴涵项一起构成函数的最小覆盖（即最简的质蕴涵项集）。

第一步：画出卡诺图
第二步：画出卡诺圈（这一步最重要）
第三步：根据卡诺圈，写出简化后的函数

  - **3、列表化简法（用于变量大于4的情况，简化起来非常麻烦）**
    - 第一步：将函数表示成“最小项之和”（标准“与-或”表达式）的形式，并用二进制码表示每一个最小项。
    - 第二步：找出函数的全部质蕴涵项。
    - 第三步：找出函数的必要质蕴涵项。
    - 第四步：找出函数的最小覆盖。

习题（P48–P49）

- 2.2（1）
- 2.3（1）
- 2.4（1）
- 2.5
- 2.6（1）
- 2.7（1）
- 2.8（1）
- 2.9（2）
- 2.10（1）

习题（P48–P49）

2.1 假定一个电路中，指示灯 F 和开关 A、B、C 的关系为

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

试画出相应电路图。

2.2 用逻辑代数的公理、定理和规则证明下列表达式：

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

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

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

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

2.3 用真值表验证下列表达式：

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

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

2.4 利用反演规则和对偶规则求下列函数的反函数和对偶函数：

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

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

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

(4) $F=A[\overline{B}+(C\overline{D}+\overline{E})G]$

2.5 判断下列逻辑命题正误，并说明理由：

(1) 如果 $X+Y$ 和 $X+Z$ 的逻辑值相同，那么，Y 和 Z 的逻辑值一定相同。

(2) 如果 XY 和 XZ 的逻辑值相同，那么，Y 和 Z 的逻辑值一定相同。

(3) 如果 $X+Y$ 和 $X+Z$ 的逻辑值相同，且 XY 和 XZ 的逻辑值相同，那么，Y 和 Z 的逻辑值一定相同。

(4) 如果 $X+Y$ 和 $X\cdot Y$ 的逻辑值相同，那么，X 和 Y 的逻辑值一定相同。

2.6 用代数化简法求下列逻辑函数的最简与-或表达式：

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

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

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

(4) $F=BC+D+\overline{D}(\overline{B}+\overline{C})(AC+B)$

2.7　将下列逻辑函数表示成“最小项之和”及“最大项之积”的简写形式：

(1) $F(A,B,C,D)=B\bar{C}\bar{D}+\bar{A}B+AB\bar{C}D+BC$

(2) $F(A,B,C,D)=\overline{\bar{A}\bar{B}+ABD}+B+CD$

2.8　用卡诺图化简法求出下列逻辑函数的最简与-或表达式和最简或-与表达式：

(1) $F(A,B,C,D)=\bar{A}\,\bar{B}+\bar{A}\,\bar{C}D+AC+B\bar{C}$

(2) $F(A,B,C,D)=BC+D+\bar{D}(\bar{B}+\bar{C})(AD+B)$

(3) $F(A,B,C,D)=\prod M(2,4,6,10,11,12,13,14,15)$

2.9　用卡诺图判断函数 F 和 G 之间的关系：

(1) $F(A,B,C,D)=\bar{B}\,\bar{D}+\bar{A}\,\bar{D}+\bar{C}\,\bar{D}+AC\,\bar{D}$

$G(A,B,C,D)=\bar{B}D+CD+\bar{A}\,\bar{C}D+ABD$

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

$G(A,B,C)=\overline{AB+BC+AC}(A+B+C)+ABC$

2.10　某函数的卡诺图如图 2.18 所示，请回答如下问题：

(1) 若 $b=\bar{a}$，则当 a 取何值时能得到最简的与-或表达式？

(2) 若 a、b 均任意，则 a 和 b 各取何值时能得到最简的与-或表达式？

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

图 2.18　卡诺图

2.11　用列表法化简逻辑函数：

$$F(A,B,C,D)=\sum m(0,2,3,5,7,8,10,11,13,15)$$

作业样例

- **1.1 冯·诺依曼结构计算机的基本思想是什么？按此思想设计的计算机硬件系统应由哪些部件组成？它们各有何作用？**

- 答：
  - 数学家冯·诺依曼提出了计算机制造的三个基本原则，即采用二进制逻辑、程序存储执行（存储程序和程序控制）以及计算机由五个部分组成（运算器、控制器、存储器、输入设备、输出设备），这套理论被称为冯·诺依曼体系结构。
  - 计算机硬件系统应由运算器、控制器、存储器、输入设备、输出设备等组成。
  - 运算器是一种用于信息加工处理的部件，它对数据进行算术运算和逻辑运算。运算器通常由算术逻辑单元（ALU，Arithmetic and Logic Unit）和一系列寄存器组成。通常将运算器一次运算能处理的二进制位数称为机器字长。现代计算机具有多个寄存器，称为寄存器组。
  - 控制器是整个计算机的指挥中心，它可使计算机各部件协调工作。计算机中有两股信息在流动，一股是控制流信息，另一股是数据流信息。控制流信息的发源地是控制器，控制器产生控制流信息的依据来自3个方面：指令寄存器、状态寄存器和时序电路。
  - 存储器的主要功能是存放程序和数据，目前计算机的主存储器都是半导体存储器。
  - 输入设备就是将信息输入计算机的外部设备，它将人们熟悉的信息形式转换成计算机能接收并识别的信息形式。
  - 输出设备就是将计算机运算结果转换成人们和其他设备能接收和识别的信息形式的设备，如字符、文字、图形、图像、声音等。

关于作业提交

- 作业必须按时提交（上传到学院的FTP服务器上），否则认为是迟交作业；如果期末仍然没有提交，则认为是未提交作业。
  - 作业完成情况成绩=第1次作业提交情况*第1次作业评分+第2次作业提交情况*第2次作业评分+……+第N次作业提交情况*第N次作业评分。
  - 作业评分：A（好）、B（中）、C（差）三挡。
  - 作业提交情况：按时提交（1.0）、迟交（0.5）、未提交（0.0）。
- 请采用电子版的格式（PPT文档）上传到FTP服务器上，文件名取“学号+姓名+第X次作业.pptx”。
  - 例如：1192022202406+刘济华+第2次作业.pptx
- 下次上课时（2024年9月23日）会随机抽取2位同学到讲台上汇报作业。
- 第2次作业提交的截止日期为：2024年9月22日晚上24点。

Thanks