第三讲 逻辑关系描述

- 逻辑函数
- 真值表与波形图
- 标准范式表示

3.1 逻辑函数

◆ **逻辑函数**是反映输入变量和输出变量之间逻辑关系的表达式。

- 将一组取值范围在{0，1} 之中的输入变量唯一映射到在同样取值范围中的输出变量。

设$X_1, X_2, \cdots, X_n$是n个变量，每个变量取值0 或者取值1。

$F(X_1, X_2, \cdots, X_n)$ 是$X_1, X_2, \cdots, X_n$的一个逻辑函数，其取值由$X_1, X_2, \cdots, X_n$的取值决定——所有可能的情况

$X_1,X_2,\cdots,X_n$	$F(X_1,X_2,\cdots,X_n)$
0 0 ⋯ 0	0
0 0 ⋯ 1	0
⋯	⋯
0 1 ⋯ 0	1
0 1 ⋯ 1	0
⋯	⋯
1 0 ⋯ 0	1
⋯	⋯
1 1 ⋯ 1	0

3.1 逻辑函数

- 每一组输入组合都有一个确定的输出值
- 每个逻辑函数都有一组确定的输出
- 两个输入变量（4组输入组合）的函数F(x,y)可能的16组输出值如下：

X	Y	$F_i(X, Y)$															
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

三个输入变量有多少种可能的逻辑函数f(x,y,z)?

有2^8=256种

3.2 真值表与波形图

- **真值表 true table**：用二维表的形式列出逻辑函数**所有的输入组合**和对应的**输出值**。
- 标题栏左侧为输入组合，右侧对应输出。
- 输入组合通常按照**递增**顺序排列，输出写在相邻的列中。
- n个输入变量逻辑函数的真值表有2^n行。
- 当n较大时，真值表将变得十分巨大而失去使用价值。

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

输出Y的逻辑表达式是？

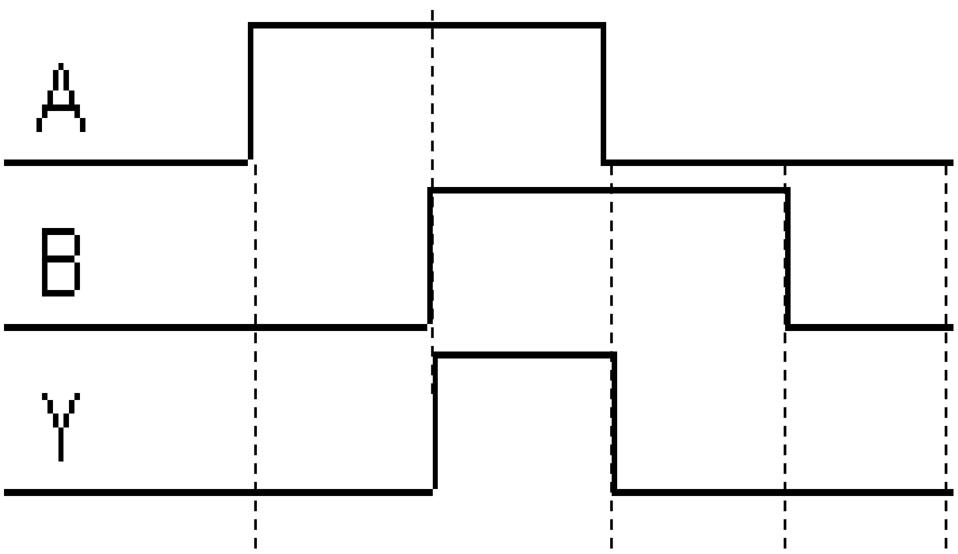
$$Y = A \cdot B + B \cdot C + A \cdot C$$

3.2 真值表与波形图

◆ **波形图**描绘了逻辑函数输出变量对于输入变量的变化所产生的响应。在理想状态下，忽略时间延迟。

- 横轴表示时间
- 纵向用**横线的高低**来表示逻辑值大小

◆ 完整的波形图至少需要列出所有的输入组合和所对应的输出值



$Y=A \cdot B$

3.3 逻辑函数的标准表示

- **乘积项**：包含1个或1个以上逻辑变量的与项。例如，X、$X\cdot Y$和$\overline{X}\cdot\overline{Y}\cdot Z$都是乘积项。

- **求和项**：包含1个或1个以上逻辑变量的或项。例如，X、$X+Y$和$\overline{X}+\overline{Y}+Z$都是求和项。

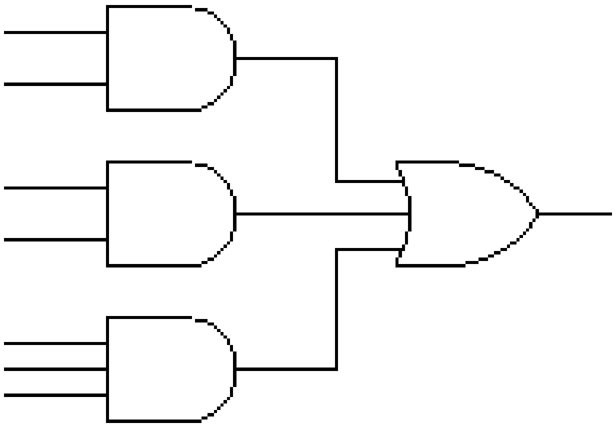
- **"与-或"表达式**或**积之和表达式**(Sum of Product，SOP)：多个乘积项的或运算。例如：$X\cdot Y+\overline{X}\cdot\overline{Y}\cdot Z$。

- **"或-与"表达式**或**和之积表达式**(Product of Sum，POS)：多个求和项的与运算。例如：$(X+Y)\cdot(\overline{X}+\overline{Y}+Z)$。

3.3 逻辑函数的标准表示

◆ **与-或表达式：积之和表达式SOP**

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



与项

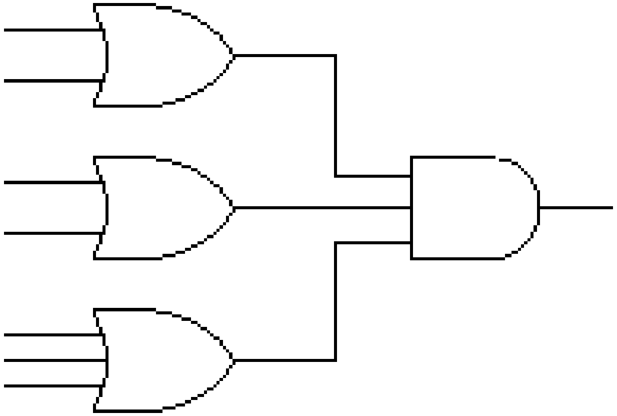
乘积项

◆ **或-与表达式：和之积表达式POS**

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

求和项

或项



3.3 逻辑函数的标准表示

◆ **标准乘积项（标准求和项）：每个逻辑变量出现且仅出现一次的乘积项（求和项）。 n 个变量的标准项共有2^n个。**

$A \cdot B$	11	$A + B$	00	$A \cdot B \cdot C$
$A \cdot \overline{B}$	10	$A + \overline{B}$	01	$A \cdot B \cdot \overline{C}$
$\overline{A} \cdot B$	01	$\overline{A} + B$	10	$A \cdot \overline{B} \cdot C$
$\overline{A} \cdot \overline{B}$	00	$\overline{A} + \overline{B}$	11	$A \cdot \overline{B} \cdot \overline{C}$
				$\overline{A} \cdot B \cdot C$
				$\overline{A} \cdot B \cdot \overline{C}$
				$\overline{A} \cdot \overline{B} \cdot C$
				$\overline{A} \cdot \overline{B} \cdot \overline{C}$

当每个逻辑变量取值定下来之后，**只有一个乘积项的输出值为1**

当每个逻辑变量取值定下来之后，**只有一个求和项的输出值为0**

3.3 逻辑函数的标准表示

- 标准乘积项（标准求和项）：每个逻辑变量出现且仅出现一次的乘积项（求和项）。 n 个变量的标准项共有2^n个。

- 标准乘积项也称为最小项，每个最小项对应真值表中一个输入组合，赋值该输入组合后，最小项的运算结果为1。
- 若某输入组合对应的二进制数值为 i，则用m_i表示该最小项，i称为该最小项的编号。
  - 如：$\overline{A}\bullet\overline{B}\bullet C$ 只有当输入001时，结果为1，最小项编号为m_1

- 标准求和项也称为最大项，每个最大项对应真值表中一个输入组合，赋值该输入组合后，最大项的运算结果为0。
- 若某输入组合对应的二进制数值为 i，则用M_i表示该最大项，i称为该最大项的编号。
  - 如：$\overline{A}+\overline{B}+C$ 只有输入110时，结果为0，最大项编号为M_6

3.3 逻辑函数的标准表示

真值表

序号	$A\ B\ C$	Y	最小项	最大项
0	000	1	$\overline{A}\bullet\overline{B}\bullet\overline{C}$	$A+B+C$
1	001	0	$\overline{A}\bullet\overline{B}\bullet C$	$A+B+\overline{C}$
2	010	0	$\overline{A}\bullet B\bullet\overline{C}$	$A+\overline{B}+C$
3	011	0	$\overline{A}\bullet B\bullet C$	$A+\overline{B}+\overline{C}$
4	100	0	$A\bullet\overline{B}\bullet\overline{C}$	$\overline{A}+B+C$
5	101	0	$A\bullet\overline{B}\bullet C$	$\overline{A}+B+\overline{C}$
6	110	1	$A\bullet B\bullet\overline{C}$	$\overline{A}+\overline{B}+C$
7	111	1	$A\bullet B\bullet C$	$\overline{A}+\overline{B}+\overline{C}$

标准和/最小项列表：使得函数输出为1的最小项之和。

$$Y=\overline{A}\bullet\overline{B}\bullet\overline{C}+A\bullet B\bullet\overline{C}+A\bullet B\bullet C$$
$$=\sum m\,(0,6,7)$$

标准积/最大项列表：使得函数输出为0的最大项之积。

$$Y=(A+B+\overline{C})\bullet(A+\overline{B}+C)\bullet(A+\overline{B}+\overline{C})\bullet(\overline{A}+B+C)\bullet(\overline{A}+B+\overline{C})$$
$$=\prod M(1,2,3,4,5)$$

真值表的表示是唯一的，因此标准项列表的表示也是唯一的。

3.3 逻辑函数的标准表示

- 函数的最小项列表和最大项列表**等价**且可**相互转换**。
- n 个变量的逻辑函数其最小项列表编号集合与最大项列表编号集合之并集为n位编号全集{0，1，…，2^{n-1}}，且这两个集合之交集为0，它们为**互补**关系。
- 两个列表之间可方便转换，只需对相应的编号集合**求补**即可。
  - $f(A,B,C)=\Sigma\, m\,(0,1,2,3) = \Pi\, M(4,5,6,7)$
  - $f(X,Y) = \Sigma\, m\,(1) = \Pi\, M(0,2,3)$
  - $f(A,B,C) = \Sigma\, m\,(0,6,7) = \Pi\, M(1,2,3,4,5)$

标准表示并非最简表示

第四讲 逻辑函数的化简与变换

- 代数法化简
- 卡诺图化简
- 等效逻辑符号表示
- 逻辑函数变换

4 逻辑函数的化简与变换

化简的目的

- 减少输入变量的数目
    - 最小项和最大项的数目随变量数呈指数级增长
- 减少门的数目
    - 可以使用更小的集成电路器件
- 减少电路的规模
    -

1. 通常不考虑输入反相器的成本。
2. 通常从真值表或标准范式开始化简。

4.1 代数法化简

- 利用布尔代数的公理、定理、定律等，消去逻辑表达式式中多余的乘积项或多余的因子，进行化简。
  - 利用互补律T5：$\overline{X}+X=1$，可消去一个变量。
  - 利用吸收律T9：$X+X\cdot Y=X$ 和 $X+\overline{X}\cdot Y=X+Y$，可消去乘积项中一个因子。
  - 利用组合律T10：$X\cdot Y+X\cdot\overline{Y}=X$ 和 $(X+Y)\cdot(X+\overline{Y})=X$，可消去一个变量。
  - 利用一致律T11，可消去冗余的乘积项。
- 如果表达式的层级超过了两级，则先转换为两级。与或，或与
- 如有整体取反运算，则先转换为单变量取反运算。

4.1 代数法化简

- 化简：F(w,x,y,z)=$w\cdot x+w\cdot x\cdot y+\bar{w}\cdot y\cdot z+\bar{w}\cdot\bar{y}\cdot z+\bar{w}\cdot x\cdot y\cdot\bar{z}$。

$=w\bullet x+w(y\bullet z\overline{+}y\bullet z+x\overline{\bullet y}\bullet z)$ (根据T8和T9)

$=w\bullet x+\overline{w}(z+x\bullet y\overline{\bullet z})$ (根据T10)

$=w\bullet x+\overline{w}(z+x\bullet y)$ (根据T9)

$=w\bullet x+\overline{w}\bullet z+\overline{w}\bullet x\bullet y$ (根据T8)

$=(w+\overline{w}\bullet y)\bullet x+\overline{w}\bullet z$ (根据T8)

$=(w+y)\bullet x+\overline{w}\bullet z$ (根据T9)

$=w\bullet x+x\bullet y+\overline{w}\bullet z$ (根据T8)

- 和原表达式相比，化简后减少了2个与门、11个输入端。

原来第一级有5个与门，15个输入端；第二级是5个与门的输出作为输入端，接一个或门
化简后第一级有3个与门，6个输入端；第二级是3个与门的输出作为输入端，接一个或门

4.1 代数法化简

- **化简** $f(A,B,C,D) = \sum m(1,3,5,7)$

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

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

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

- **优点**
  - **不受变量数目的限制**
  - **化简比较直观**
- **缺点**
  - **没有一定的规律和步骤，技巧性很强**
  - **难以判断化简结果是否最简**
- **判别逻辑表达式是否为最小化**
  - **乘积项（与项）最少**
  - **每个乘积项中因子（逻辑变量）最少**

4.2 卡诺图化简

- **卡诺图（Karnaugh map）：真值表的图形化表示，把能化简的最小项通过相邻项合并的可视化方式标识出来。**
  - n个变量的卡诺图是一个含有2^n个单元的矩阵图
  - 每一行和每一列的编号对应逻辑变量的输入组合，0 表示反变量，1 表示原变量
  - 编号按照格雷码的顺序排列，即相邻编号只有1位不同
  - 根据格雷码的规则，空间位置上（上下、左右或首尾）相邻的小方格具有逻辑相邻性

$A \backslash B$	0	1
0	$\overline{A}\,\overline{B}$	$\overline{A}B$
1	$A\overline{B}$	AB

$A \backslash B$	0	1
0	m_0	m_1
1	m_2	m_3

每个单元对应一个最小项

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

4.2 卡诺图化简

◆ **行、列可互换**

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

XY \ Z	0	1
00	0	1
01	2	3
11	6	7
10	4	5

Y \ X	0	1
0	m_0	m_2
1	m_1	m_3

(a) 二变量

Z \ XY	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

(b) 三变量

YZ \ WX	00	01	11	10
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}

(c) 四变量

4.2 卡诺图化简

- 每个单元标注对应最小项在**真值表**中的输出值，若为1，称之“**1单元**”
- 若两个“1单元”**相邻**，则表示两个最小项仅1个变量相反。根据T10，这两个最小项可合并为**一个乘积项**，并消去**那个**具有相反取值的变量

(T10) $X \cdot Y + X \cdot \overline{Y} = X$

- 相邻单元数越多可消去的变量数越多
- 相邻2^i个“**1单元**” 可以合并成一个乘积项，并消去**i**个不同的变量
- 使用一个方框来标注可以合并的“1单元”，这个方框称为**卡诺圈**

A \ BC	00	01	11	10
0		1		
1		1		

$\overline{B} \cdot C$

A取值相反

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

$\overline{B} \cdot \overline{D}$

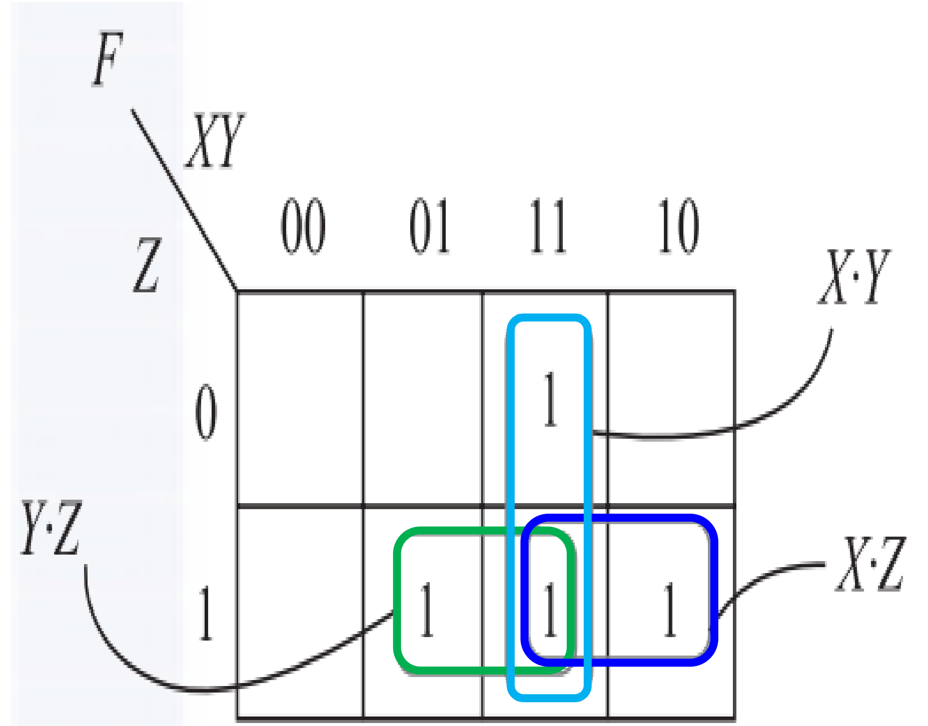
A和C取值相反

AB \ CDE	000	001	011	010	110	111	101	100
00			1			1		
01			1			1		
11			1			1		
10			1			1		

A、B、C取值相反 $D \cdot E$

4.2 卡诺图化简

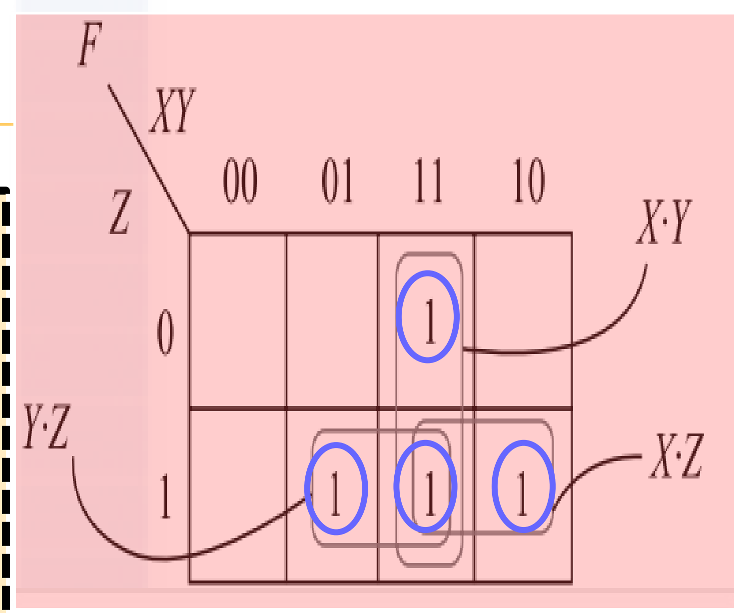
序号	X Y Z	F (X, Y, Z)
0	0 0 0	0
1	0 0 1	0
2	0 1 0	0
3	0 1 1	1
4	1 0 0	0
5	1 0 1	1
6	1 1 0	1
7	1 1 1	1



$$F\ (X,\ Y,\ X) = Y\cdot Z + X\cdot Z + X\cdot Y$$

简化后的表示也并非一定是：最简表示

4.2 卡诺图化简



◆ **蕴涵项**是一个乘积项，**覆盖**了逻辑函数的1个或多个最小项。

如，$F(X,Y,Z)=\sum m(3,5,6,7)$ 的蕴涵项有：$\overline{X}\cdot Y\cdot Z$、$X\cdot Y\cdot Z$和$Y\cdot Z$等，$Y\cdot Z$等价于$\overline{X}\cdot Y\cdot Z+X\cdot Y\cdot Z$，因而$Y\cdot Z$**覆盖了**最小项$\overline{X}\cdot Y\cdot Z$和$X\cdot Y\cdot Z$。

因此，$Y\cdot Z$是蕴涵项。

◆ **质蕴涵项**(prime implicant)：不能被逻辑函数的其它蕴涵项所**覆盖**的蕴涵项。

如，$F(X,Y,Z)=\sum m(3,5,6,7)$ 的质蕴涵项有：$Y\cdot Z$、$X\cdot Z$、$X\cdot Y$，而其他蕴含项都可以被另外的蕴含项所覆盖，因此，不是质蕴含项

◆ **实质蕴涵项**(essential prime implicant)：覆盖的最小项中**至少有一个最小项是**没有被其他质蕴涵项所覆盖的质蕴涵项。

如，$F(X,Y,Z)=\sum m(3,5,6,7)$ 的实质蕴涵项有：$Y\cdot Z$、$X\cdot Z$、$X\cdot Y$。

$Y\cdot Z$覆盖的$\overline{X}\cdot Y\cdot Z$、$X\cdot Z$覆盖的$X\cdot\overline{Y}\cdot Z$、$X\cdot Y$覆盖的$X\cdot Y\cdot\overline{Z}$都没有被其他质蕴含项所覆盖。

◆ 质蕴涵项覆盖的最小项越多**（即卡诺圈越大）**越可能是实质蕴涵项。

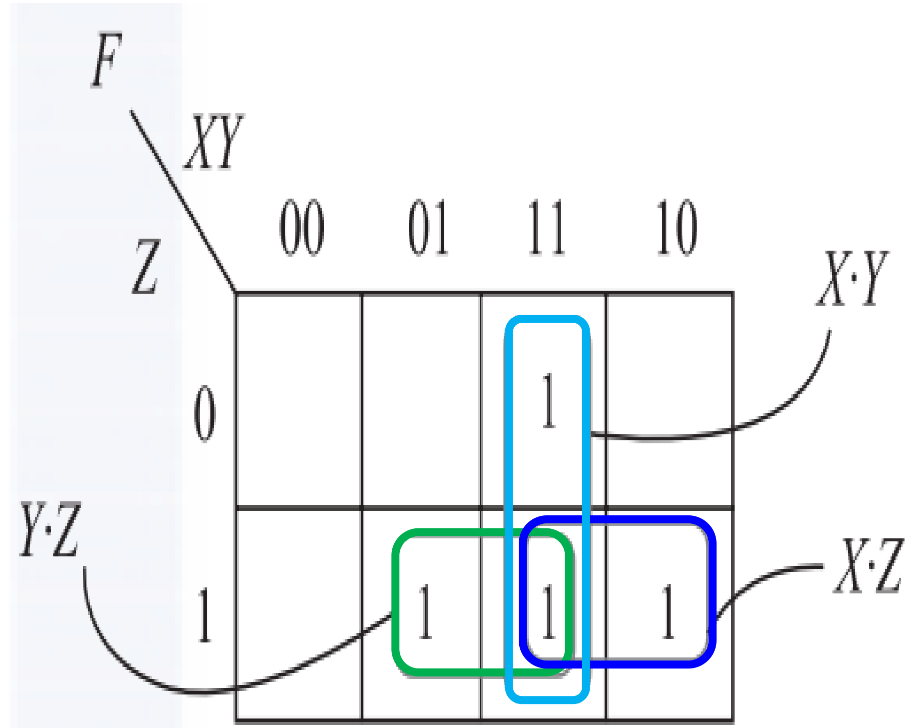
4.2 卡诺图化简

- 若逻辑函数的所有最小项都被一组质蕴涵项所覆盖，则该组质蕴涵项称为**函数的一个覆盖**(Cover)，它一定包含**所有的实质蕴涵项**。

如，{Y·Z，X·Z，X·Y}是F(X, Y, Z)=∑m(3, 5, 6, 7)的一个覆盖，包含所有实质蕴含项

- **最小覆盖**：包含**质蕴涵项数最少**，并且质蕴涵项中的**变量总数也是最少**。
- 逻辑函数化简问题就转化为寻找函数的**最小覆盖**问题。

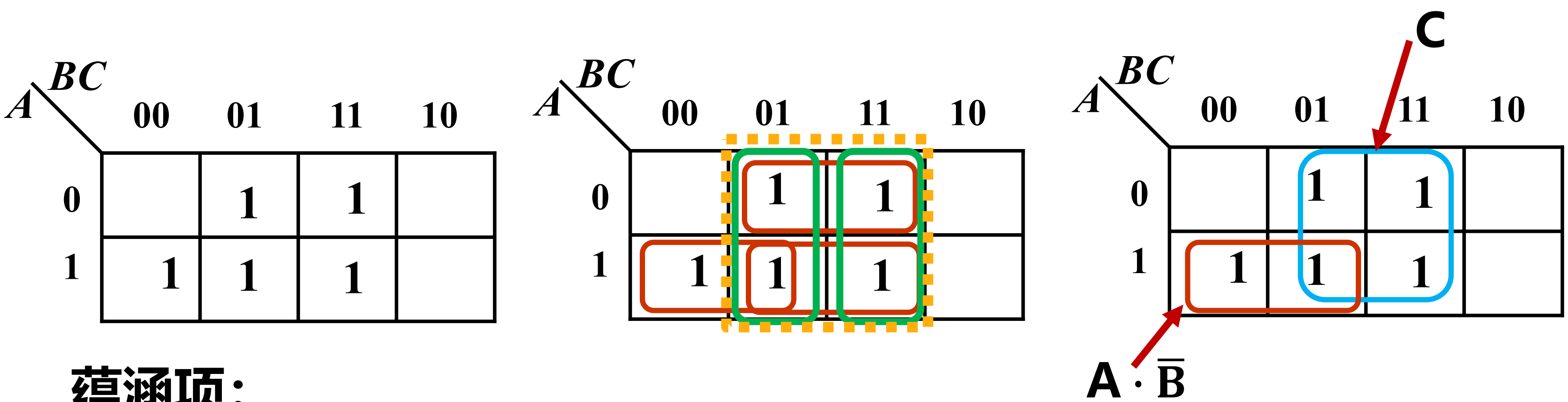
$F(X, Y, X) = Y \cdot Z + X \cdot Z + X \cdot Y$



最小覆盖对应的逻辑表达式称为：最简逻辑表达式

4.2 卡诺图化简

- **确定逻辑函数F(A,B,C)=Σm(1,3,4,5,7)的最小覆盖的方法**



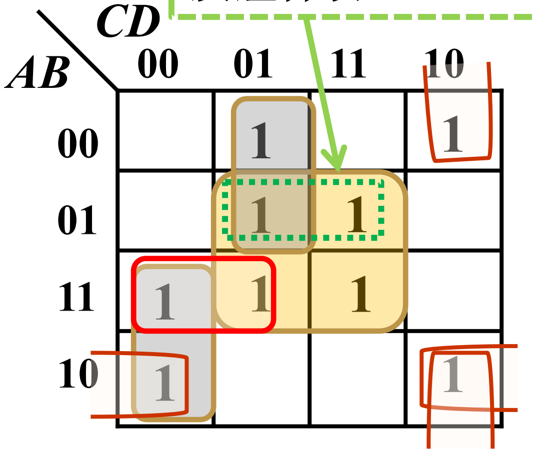
蕴涵项：

最小项： $\{\bar{A}\cdot\bar{B}\cdot C,\ \bar{A}\cdot B\cdot C,\ A\cdot\bar{B}\cdot\bar{C},\ A\cdot\bar{B}\cdot C,\ A\cdot B\cdot C\}$

4.2 卡诺图化简

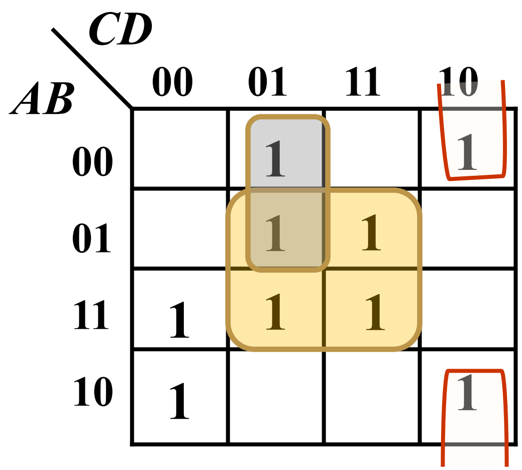
◆ $F(A,B,C,D) = \sum m(1,2,5,7,8,10,12,13,15)$

这是蕴含项，但不是质蕴含项



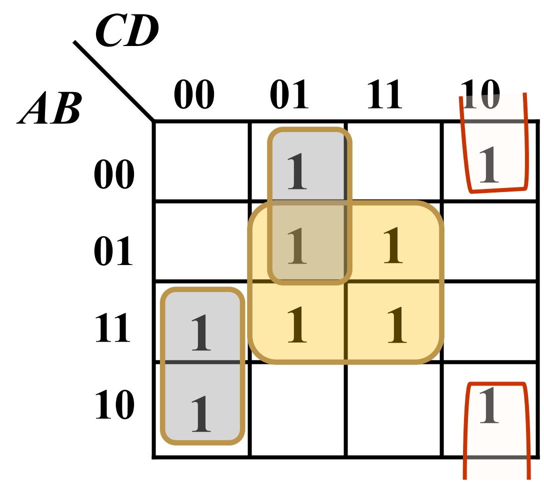
找出所有质蕴涵项

$B\cdot D$, $\bar{A}\cdot\bar{C}\cdot D$, $\bar{B}\cdot C\cdot\bar{D}$, $A\cdot B\cdot\bar{C}$, $A\cdot\bar{B}\cdot\bar{D}$, $A\cdot\bar{C}\cdot\bar{D}$



选择所有的实质蕴涵项

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



在剩余质蕴含项中选择最小覆盖子集

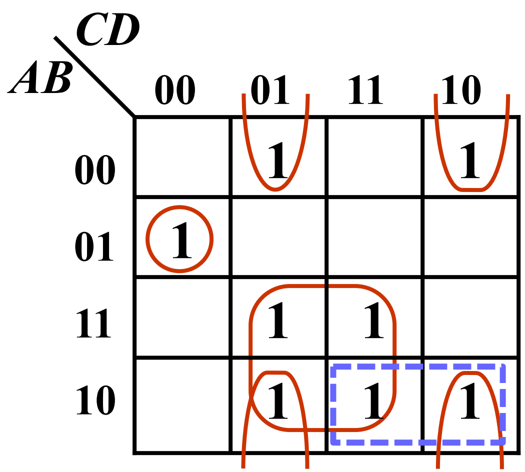
$A\cdot\bar{C}\cdot\bar{D}$

合并实质蕴含项和最小覆盖子集，得到最简逻辑表达式：

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

4.2 卡诺图化简

$$Y(A,B,C,D)=\sum m(1,\quad 2,\quad 4,\quad 9,\quad 10,\quad 11,\quad 13,\quad 15)$$



这是质蕴含项，但
不是实质蕴含项

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

4.2 卡诺图化简

- 利用对偶性原理，卡诺图也可以用来化简**和之积表达式**，只需要将真值表中输出值为0的最大项对应的单元标注为0，然后合并相邻的“0 单元”，得到求和的质蕴涵项。

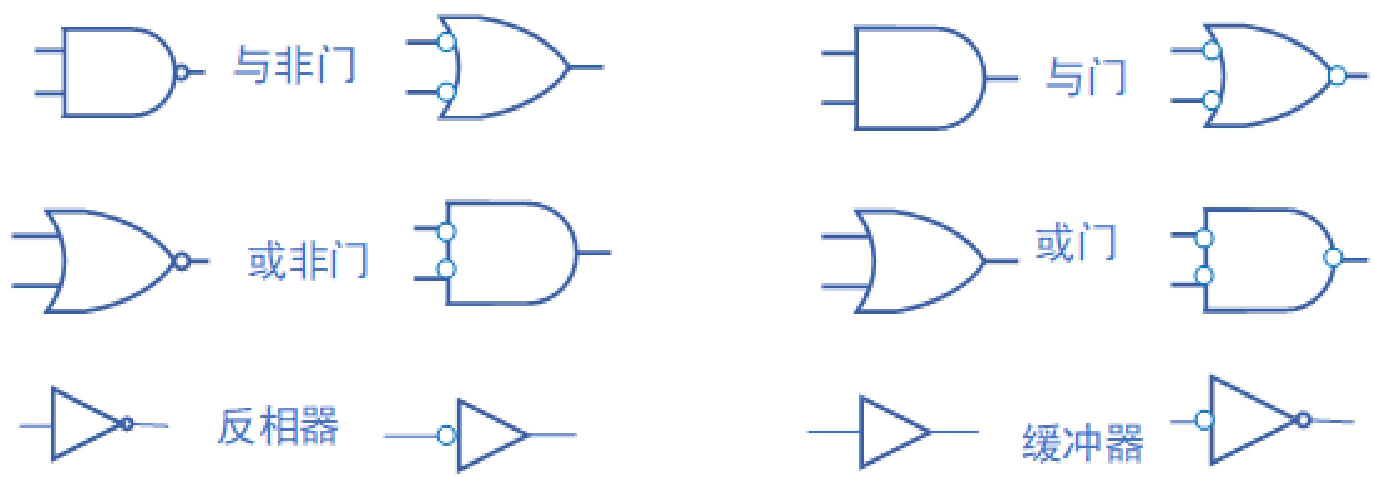
- 卡诺图化简优点
  - 方便、直观、容易掌握

- 卡诺图化简缺点
  - 受到变量数量的约束，当变量数大于6时，卡诺图绘制以及相邻关系的识别将变得非常复杂，从而导致难以直观化简。

在现代数字系统设计中，大多采用编程实现**逻辑函数的自动化简**。

4.3 逻辑函数变换——实际电路实现方式

- **等效逻辑符号**：功能相同，符号不同。
  - 反相输出门利用一次德•摩根定理，转换为非反相输出门。
  - 非反相输出门两次取反，利用德•摩根定理转换下层的取反运算，可得到反相输出门。

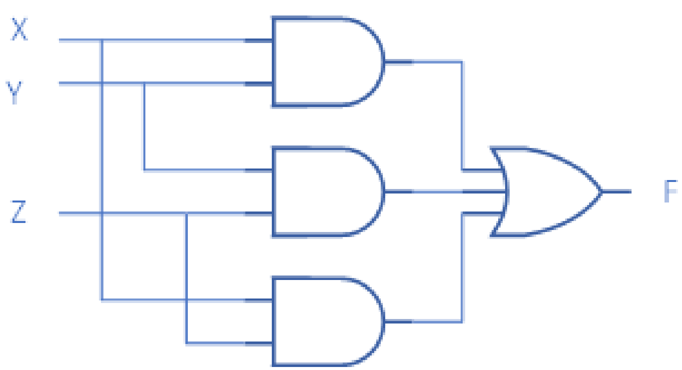


(a)反相门的等效符号　　(b)非反相门的等效符号

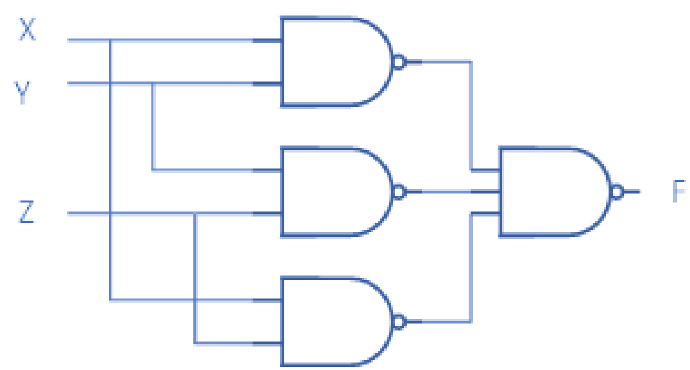
4.3 逻辑函数变换——实际电路实现方式

- 在数字电路中，**与非门**和**或非门**通常比**与门**和**或门**的执行**速度快**。
- 将“与-或”表达式转换为“与非-与非”表达式
  - 将“与-或”表达式整体**两次取反**，然后运用德摩根定律**转换下层**的取反运算，就可以得到“与非-与非”表达式
  - 使用**与非门**替代**与门**和**或门**来实现逻辑函数
  - 例如：F(X,Y,Z)=X•Y+Y•Z+X•Z，转换为：

$$F(X,Y,Z)=\overline{\overline{X\cdot Y+Y\cdot Z+X\cdot Z}}=\overline{\overline{X\cdot Y}\cdot\overline{Y\cdot Z}\cdot\overline{X\cdot Z}}$$



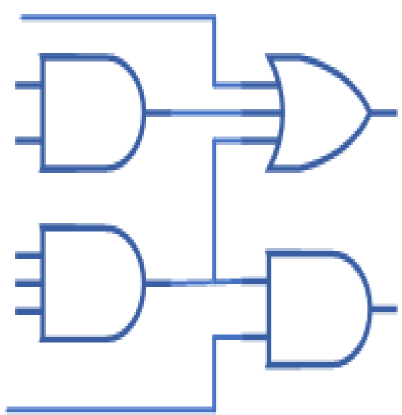
(a)与-或电路



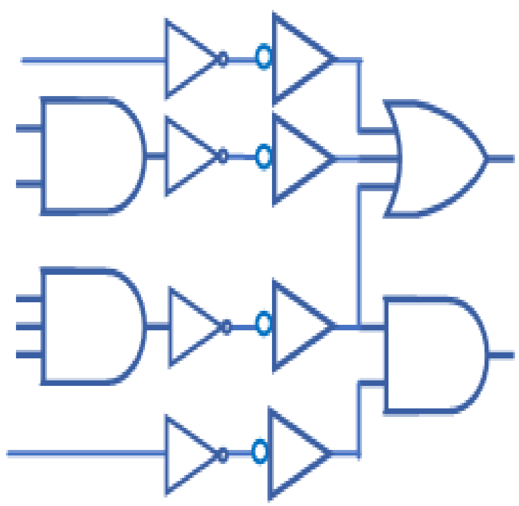
(b)与非-与非电路

4.3 逻辑函数变换

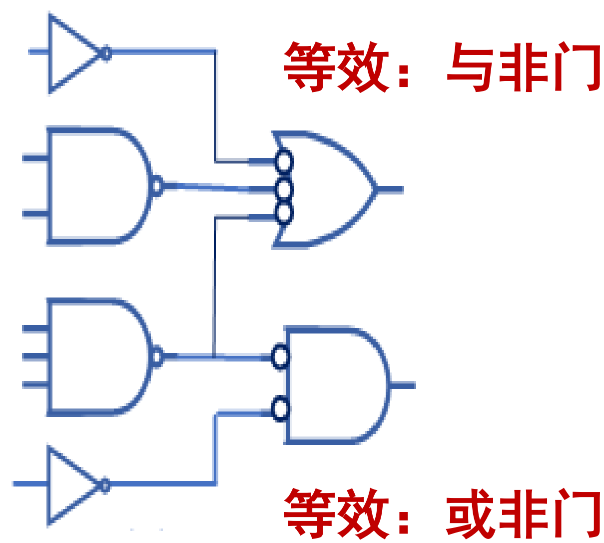
- 任何**积之和**表达式：都可使用“**与-或**”电路和“**与非-与非**”电路这两种方法实现。
- 任何**和之积**表达式：都可使用“**或-与**”电路和“**或非-或非**”电路这两种方法实现。
- 任何两级电路：在第一级的输出和第二级的输入之间加入**一对反相器**，来实现用反相门替代与门和或门。



(a)初始电路



(b)加入反相器对的电路



(c)使用反相输出端和反向输入端的电路

第2章总结

- 现实世界的模拟信号需转换为数字信号。数字系统中的所有信号都是二值的，用0和1表示
- **逻辑门**是最基础的数字电路，可通过**CMOS晶体管**实现
- 最基本的逻辑运算有与、或、非三种运算，对应的逻辑门分别为与门、或门和非门
- **布尔代数**是数字系统分析和设计的基础理论工具，对应的公理系统和定理可对逻辑表达式进行化简，实现逻辑函数间的相互转换
- 通常使用**真值表**、**波形图**以及**逻辑表达式**来描述逻辑变量间的关系
- 可使用**代数法**、**卡诺图**等来化简逻辑表达式
- 在实现数字系统时，为了提高速度、降低成本，通常利用**与非门**和**或非门**来构建电路。

作业：习题3、5、6（1）、7（5）、7（8）、8（1）、8（2）、8（4）、12、13（2）、13（5）
9月30日之前提交