数字电路

第3章 逻辑代数基础

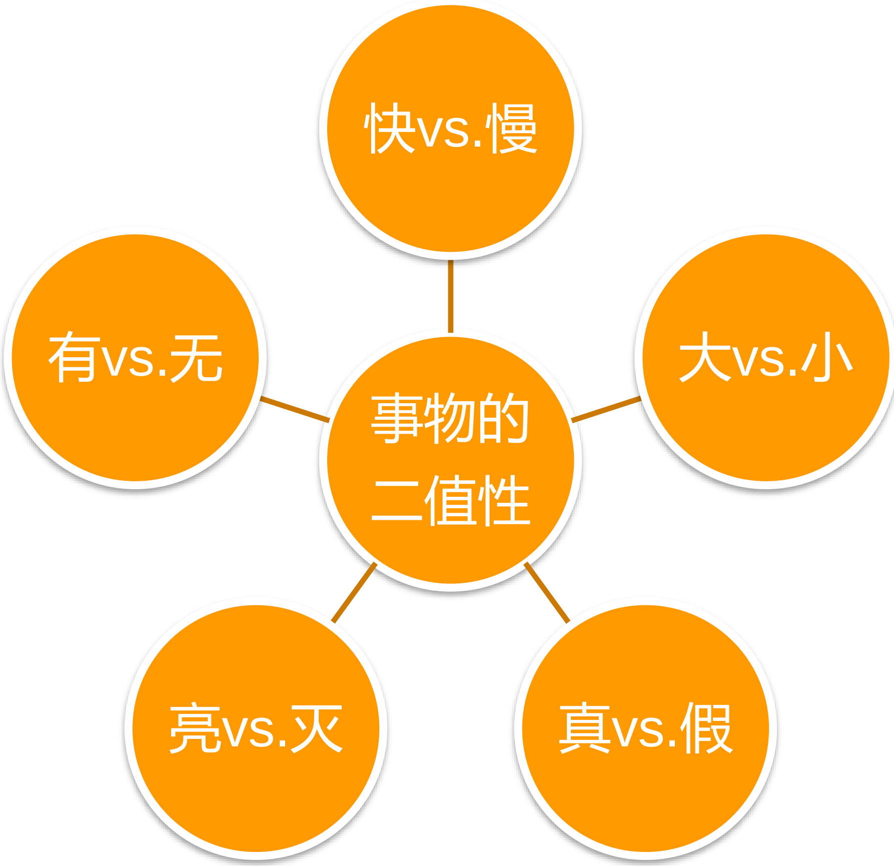
杨 旭

北京理工大学

pyro_yangxu@bit.edu.cn

本章内容

逻辑代数概述



逻辑代数

- 研究事物发展变化的因果关系的数学分支
- 又称布尔代数、开关代数

逻辑变量与逻辑函数

逻辑

事物的因果之间所应遵循的规律，最基本的逻辑关系包括“与”逻辑、“或”逻辑、“非”逻辑。

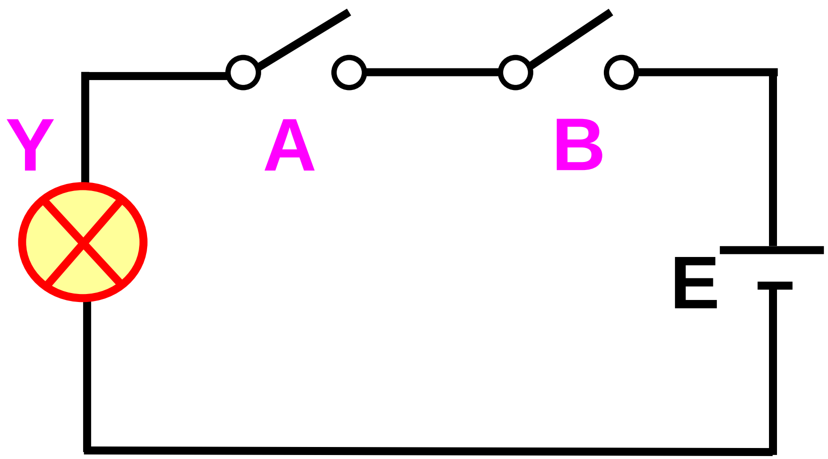
逻辑变量

逻辑代数中的二值变量，逻辑变量的取值只有“0”和“1”两个值，代表成对出现的逻辑概念，如“真”和“假”、“开”和“关”，“有”和“无”等。

“与”逻辑关系

“与”逻辑关系： 当决定事件的各个条件全部具备之后，事件才会发生。

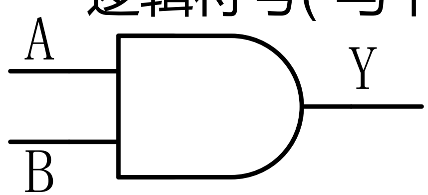
引入逻辑变量A、B、Y分别代表两个开关和灯泡的状态。
A(B)=1:开关闭合；A(B)=0:开关断开；Y=1:灯泡亮；Y=0:灯泡灭



逻辑表达式（逻辑乘法）

$$Y=A\wedge B= A\&B = A\cap B = A\cdot B = AB$$

逻辑符号(“与”门)

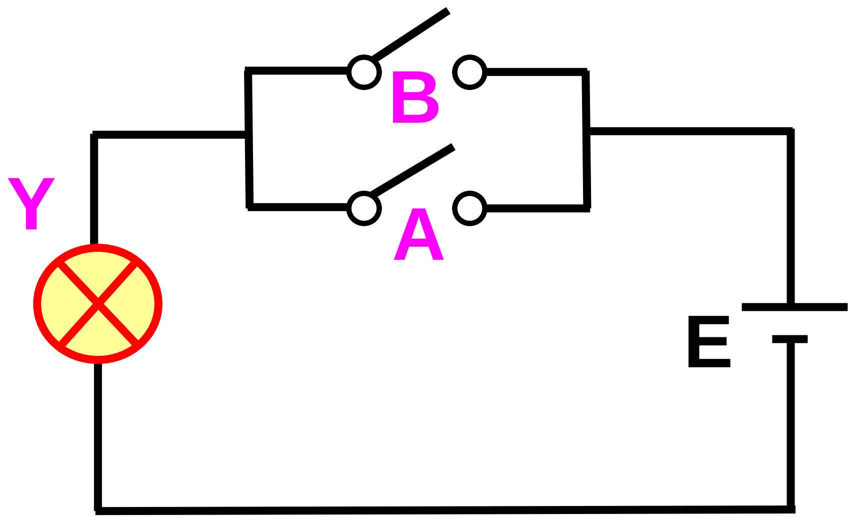


“与”逻辑真值表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

“或”逻辑关系

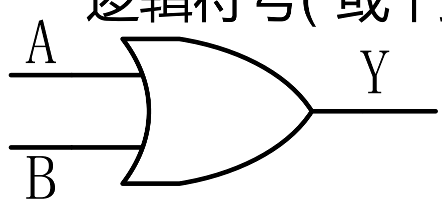
“或”逻辑关系： 当决定事件的各个条件中有一个或一个以上具备之后，事件就会发生。



逻辑表达式（逻辑加法）

$$Y=A \cup B = A \vee B = A|B = A+B$$

逻辑符号(“或”门)

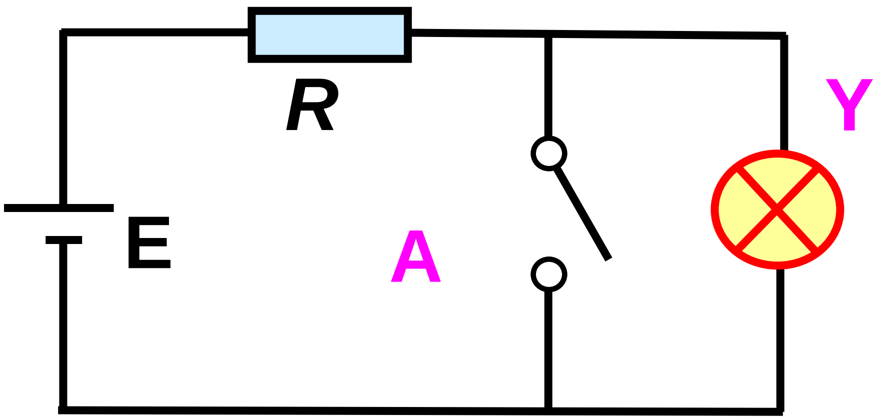


“或”逻辑真值表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

“非”逻辑关系

“非”逻辑关系： 决定事件的条件只有一个，当条件具备时，事件不会发生，条件不存在时，事件发生。

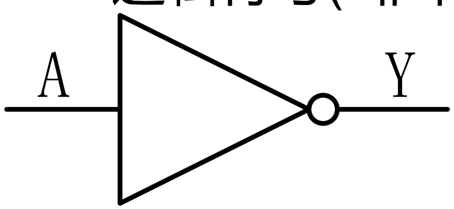


逻辑表达式（逻辑取反）

$$Y = \overline{A}$$

$\overline{A}$ 读作“A反”、“A非”或“A补”

逻辑符号(“非”门)

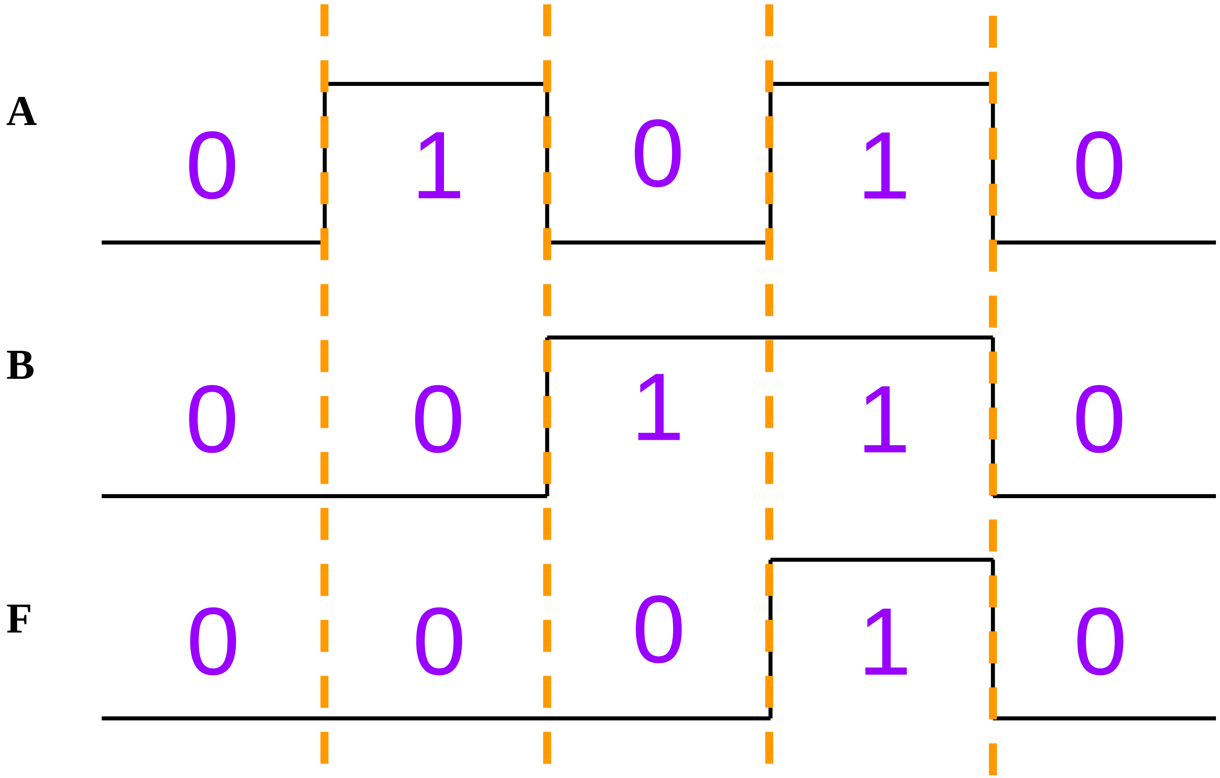


“非”逻辑真值表

A	Y
0	1
1	0

波形图—逻辑运算的另一种表示法

采用波形图也可以对逻辑运算进行描述



“与”运算的波形图

逻辑运算的五种表示方法

注意

- 对于“与”运算

$$0 \bullet 0 = 0,\ \ 0 \bullet 1 = 0,\ \ 1 \bullet 0 = 0,\ \ 1 \bullet 1 = 1$$

  - 与普通乘法的规律相同，但是含义却不同

- 对于“或”运算

$$0 + 0 = 0,\ \ 0 + 1 = 1,\ \ 1 + 0 = 1,\ \ 1 + 1 = 1$$

  - 前三条与普通加法的规律相同，但是最后一条却不同
  - 说明逻辑运算不是数值运算，逻辑运算是因果关系的逻辑判断

- 对于“非运算”

$$\overline{0} = 1,\ \ \overline{1} = 0$$

  - 这是算术里没有的

逻辑运算的优先次序

- **单变量上的“非”运算优先级最高。**

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

- **“与”运算(逻辑乘)要优先于“或”运算(逻辑加)**

$$A+B\bullet C$$

- **括弧“（）”内的运算优先于括弧外的运算**

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

- **多变量上的“非”运算相当于加括弧**

$$\overline{A+B}\bullet C \Longleftrightarrow \overline{(A+B)}\bullet C$$

逻辑函数

逻辑函数：把“与”、“或”、“非”三种基本逻辑运算组合成逻辑表达式，并将该逻辑表达式的运算结果赋予另外一个逻辑变量。

如：

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

逻辑因变量

逻辑自变量

记为：

$$F = f(A, B, C, D)$$

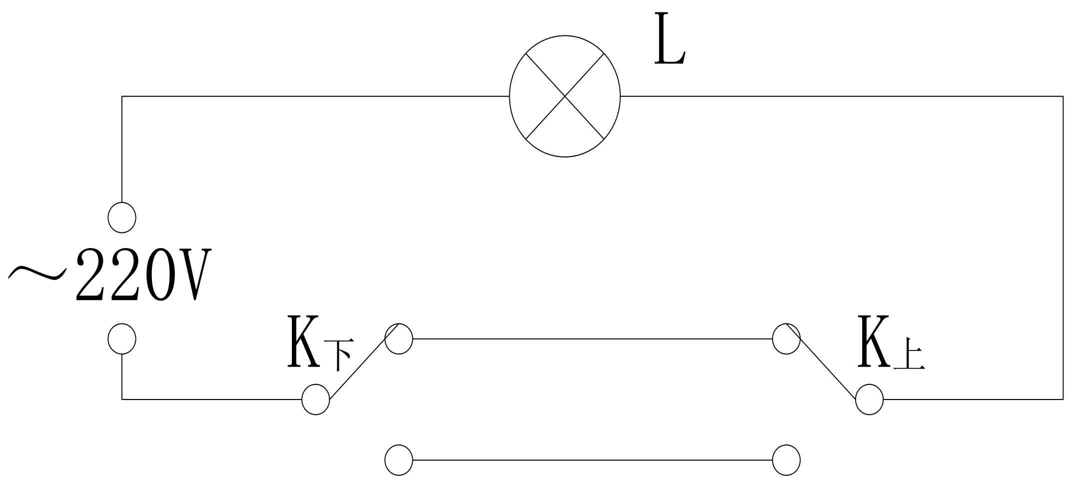
两个逻辑函数的相等

- 若两个逻辑函数F和G的输入变量相同，而且对于任意的一组变量取值都有相同的函数值，则这两个函数相等，记做：F=G。

- 任何形式的两个逻辑函数，只要它们的真值表相同，则彼此相等

逻辑函数

任何一个逻辑行为都可用一个逻辑函数来描述



$$F = A \bullet B + \overline{A} \bullet \overline{B}$$

F：灯L的状态
- F=1：灯亮
- F=0：灯灭

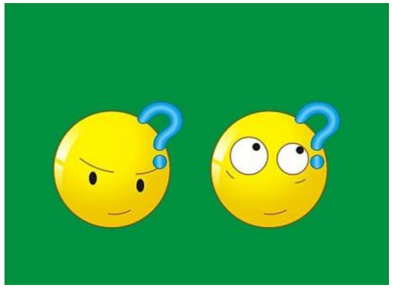
A：$K_{下}$的状态
- A=1：$K_{下}$向上扳
- A=0：$K_{下}$向下扳

B：$K_{上}$的状态
- B=1：$K_{上}$向上扳
- B=0：$K_{上}$向下扳

逻辑自变量也称为**输入逻辑变量**
逻辑因变量也称为**输出逻辑变量**

逻辑函数与真值表之间的关系

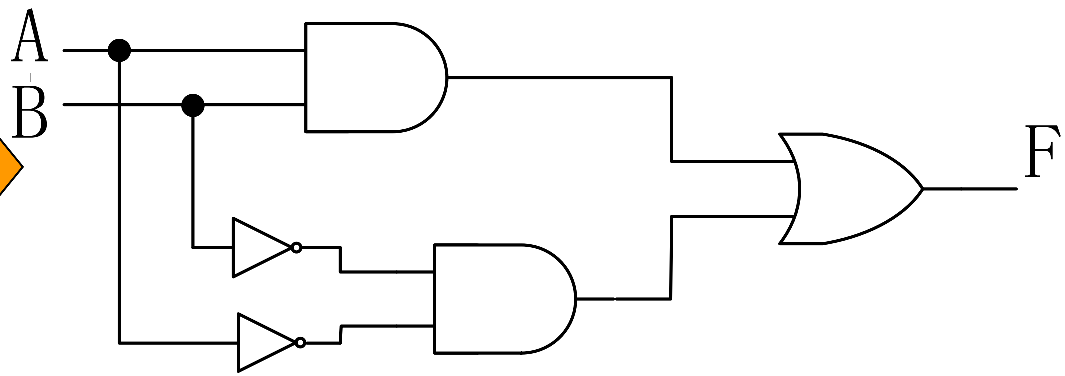
- 任何一个逻辑操作的过程，都可以用一个具有若干逻辑变量的逻辑函数来描述，也可以用一张真值表来描述
- 即，逻辑函数和真值表各自都能完全地描述一个逻辑操作的过程
- 逻辑函数与真值表之间的对应关系：
  - 一个逻辑函数对应了一张真值表
  - 一张真值表也对应了一个（或若干个）逻辑函数

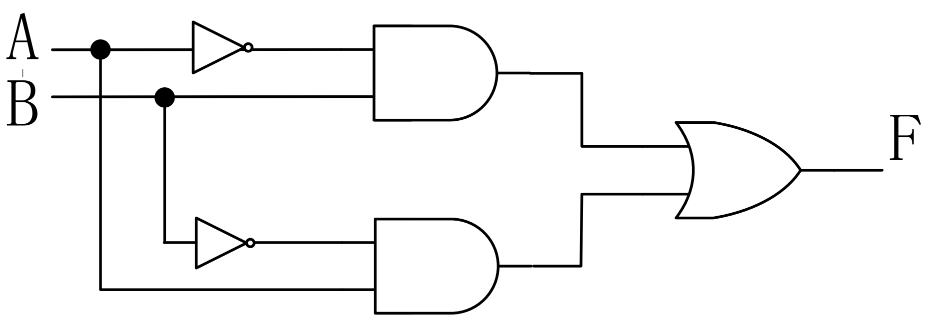


逻辑函数与逻辑电路的关系

逻辑函数 ⇄ 逻辑电路（实现 / 描述）

$$F = A \bullet B + \overline{A} \bullet \overline{B}$$





$$F = A \bullet \overline{B} + \overline{A} \bullet B$$

逻辑代数公理

逻辑代数公理—逻辑常数“0”和“1”的基本运算规则。

与	$0 \cdot 0 = 0$ $0 \cdot 1 = 1 \cdot 0 = 0$ $1 \cdot 1 = 1$	或	$0+0 = 0$ $0+1 = 1+0 = 1$ $1+1 = 1$
非	$\overline{0} = 1$		$\overline{1} = 0$

逻辑代数基本定律

名称	公式		类别	
0-1律	1. $A\cdot 0 = 0$	1'. $A+1 = 1$ *	常量和变量间的等式	
自等律	2. $A\cdot 1 = A$	2'. $A+0 = A$		
互补律	3. $A\cdot\overline{A} = 0$ *	3'. $A+\overline{A} = 1$ *		
交换律	4. $A\cdot B = B\cdot A$	4'. $A+B = B+A$	类似普通代数	变量间的等式
结合律	5. $(A\cdot B)\cdot C = A\cdot(B\cdot C)$	5'. $(A+B)+C = A+(B+C)$		
分配律	6. $A\cdot(B+C) = A\cdot B + A\cdot C$	6'. $A+B\cdot C = (A+B)(A+C)$ *		
重叠律	7. $A\cdot A = A$ *	7'. $A+A = A$ *	逻辑代数特有	
反演律(德·摩根定理)	8. $\overline{A\cdot B} = \overline{A}+\overline{B}$ *	8'. $\overline{A+B} = \overline{A}\cdot\overline{B}$ *		
还原律	9. $\overline{\overline{A}} = A$ *			

带*的公式在普通代数里没有

反演定律的证明

最简单的证明方法：真值表

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

反演定律推广（推广到多个变量的情况）

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

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

用代数方法证明基本定律

证明公式6’： A+B·C = (A+B)(A+C)

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

$$=\overline{\overline{A}\bullet\overline{B\bullet C}}=\overline{\overline{A}\bullet(\overline{B}+\overline{C})}$$ (狄·摩根定理)

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

$$=\overline{\overline{A}\bullet\overline{B}}\bullet\overline{\overline{A}\bullet\overline{C}}=(\overline{\overline{A}}+\overline{\overline{B}})(\overline{\overline{A}}+\overline{\overline{C}})$$ (狄·摩根定理)

$$=(A+B)(A+C)$$ (还原律)

特别注意

逻辑代数中不会出现指数和系数

$$A\bullet A \neq A^2 ;\quad A+A \neq 2A$$

逻辑代数中没有减法和除法

$$\cancel{A(A+B)=A \;\rightarrow\; A+B=1}$$ ？？？

$$\cancel{A+\bar{A}B=A+B \;\rightarrow\; \bar{A}B=B}$$ ？？？

逻辑代数运算的三个重要规则

规则一

代入规则：任何一个逻辑等式，将等式两边出现的同一个逻辑变量都代之以同样的逻辑函数，逻辑等式仍然成立。

利用代入规则证明多变量狄·摩根定理

狄·摩根定理： (1) $\overline{A \bullet B} = \overline{A} + \overline{B}$ (2) $\overline{A + B} = \overline{A} \bullet \overline{B}$

令 **B=C•D，** 代入（1）式：

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

再令 **B=C+D，** 代入（2）式：

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

逻辑代数运算的三个重要规则

反函数：若两个逻辑函数F和G的输入变量相同，而且F和G对于任意的一组输入变量取值都有相反的函数值，则称这两个函数互反（互补），记做 $F = \overline{G}$

规则二

反演规则：对于任意逻辑函数F，做以下三种变化后可以得到F的反函数F。

1）把原表达式中所有的“•”运算符换成“+”运算符，同时把所有的“+”运算符换成“•”运算符。

2）把原表达式中所有的逻辑常量“0”换成逻辑常量“1”，而把所有的逻辑常量“1”换成逻辑常量“0”。

3）把原表达式中所有的原变量换成反变量，再把所有的反变量换成原变量。

反演规则的证明

- 若 $F = A \bullet \overline{B} + C \bullet \overline{D}$
- 则

$$\overline{F} = \overline{A \bullet \overline{B} + C \bullet \overline{D}} = \overline{A \bullet \overline{B}} \bullet \overline{C \bullet \overline{D}} = \left(\overline{A} + B\right) \bullet \left(\overline{C} + D\right)$$

逻辑代数运算的三个重要规则

绝不能打乱原式的运算顺序

$$F = \overline{A} \bullet \overline{B} + \overline{C} + 0 \Rightarrow \overline{F} = (A + B) \bullet \overline{C} \bullet 1$$

$$\boxed{\overline{F} \neq A + B \bullet \overline{C} \bullet 1}$$

不属于单变量上的非号应保留不变

$$F = A + \overline{B + \overline{C} + \overline{D + \overline{E}}} \quad \Rightarrow \quad \overline{F} = \overline{A} \bullet \overline{\overline{B} \bullet C \bullet \overline{\overline{D} \bullet E}}$$

逻辑代数运算的三个重要规则

对偶式：对于任意逻辑函数F，做以下三种变化后即得到F的对偶式（对偶函数）F′。

1）把原表达式中所有的“•”运算符换成“+”运算符，同时把所有的“+”运算符换成“•”运算符。

2）把原表达式中所有的逻辑常量“0”换成逻辑常量“1”，而把所有的逻辑常量“1”换成逻辑常量“0”。

3）原表达式中所有的原变量和反变量均保持不变。

$$F = A \bullet \overline{B} + C \bullet \overline{D} \quad \Rightarrow \quad F' = (A + \overline{B}) \bullet (C + \overline{D})$$

$$F = A \quad \Rightarrow \quad F' = A$$

逻辑代数运算的三个重要规则

规则三

对偶规则：如果两个函数相等，则他们的对偶式（对偶函数）也相等。

若F=G，则F' = G'。

一般情况下：$\overline{F} \neq F'$

逻辑代数基本定律表格中，右边一栏的公式分别是左边一栏公式的对偶式

逻辑代数基本定理

（定理一）合并定理

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

证明:

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

$$\begin{aligned}(A+B)(A+\overline{B}) &= AA + AB + A\overline{B} + B\overline{B} \\ &= AA + (AB + A\overline{B}) + 0 \\ &= A + A \\ &= A\end{aligned}$$

逻辑代数基本定理

（定理二）吸收定理

$$A+AB=A \qquad A(A+B)=A$$

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

证明:

$$A+AB=A(1+B)=A\bullet 1=A$$

$$A(A+B)=AA+AB=A+AB=A$$

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

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

逻辑代数基本定理

（定理三）添加项定理

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

$$AB + \overline{A}C + BCD = AB + \overline{A}C \qquad (A+B)(\overline{A}+C)(B+C+D) = (A+B)(\overline{A}+C)$$

证明:

$$\begin{aligned} AB + \overline{A}C + BC &= AB + \overline{A}C + (A+\overline{A})BC \\ &= (AB + ABC) + (\overline{A}C + \overline{A}BC) \\ &= AB(1+C) + \overline{A}C(1+B) \\ &= AB + \overline{A}C \end{aligned}$$

逻辑代数基本定理

（定理四）*叫什么好呢？？*

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

证明：

$$\begin{aligned}
\overline{AB+\bar{A}C} &= \overline{AB}\bullet\overline{\bar{A}C} \\
&= (\bar{A}+\bar{B})\bullet(A+\bar{C}) \\
&= A\bar{A}+A\bar{B}+\bar{A}\bar{C}+\bar{B}\bar{C} \\
&= A\bar{B}+\bar{A}\bar{C}+\bar{B}\bar{C} \\
&= A\bar{B}+\bar{A}\bar{C}
\end{aligned}$$

复合运算和复合逻辑门

复合运算

将三种基本运算（与、或、非）按某种形式进行的简单的组合构成的新的逻辑运算

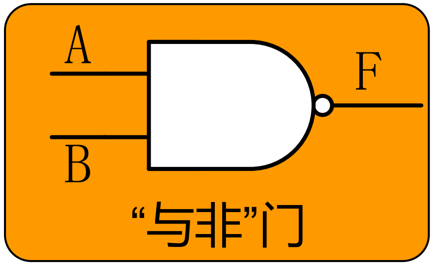
复合逻辑门

用于实现复合逻辑运算的逻辑门电路，简称复合门

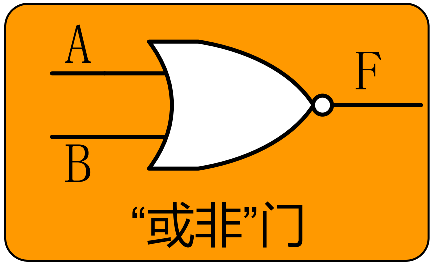
复合运算

复合运算1： “与非”、 “或非”、 “与或非”

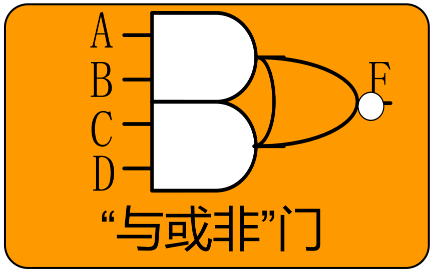
与非： $F=\overline{A\bullet B}$



或非： $F=\overline{A+B}$



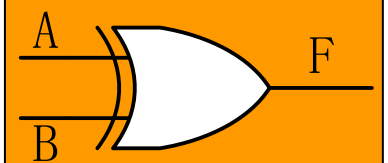
与或非： $F=\overline{A\bullet B+C\bullet D}$



复合运算

复合运算2： “异或”

定义： $F = A \oplus B = A\overline{B} + \overline{A}B$



“异或”门

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

扩展到多变量： $F = A \oplus B \oplus C$

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

异或运算满足如下基本规律：

1）交换律 $A \oplus B = B \oplus A$

2）结合律 $A \oplus B \oplus C = A \oplus (B \oplus C)$

3）分配律 $A \bullet (B \oplus C) = (A \bullet B) \oplus (A \bullet C)$

自行证明

复合运算

“异或”运算的特性

特性1：多变量“异或”运算的结果取决于变量中取值为“1”的变量的个数，而与取值为“0”的变量的个数无关。若取值为“1”的变量的个数是奇数，则“异或”的结果为“1”；若取值为“1”的变量的个数为偶数，则“异或”的结果为“0”。

由特性1可得：“异或”函数中任意一个输入变量取反，将导致运算结果取反

特性2：异或运算具有因果互换关系。即，等式两边的逻辑变量可以互相交换位置而仍然保持等式的成立。

若$A\oplus B=C$成立，则$\mathrm{A}\oplus\mathrm{C}=\mathrm{B}$成立；$\mathrm{B}\oplus\mathrm{C}=\mathrm{A}$成立。

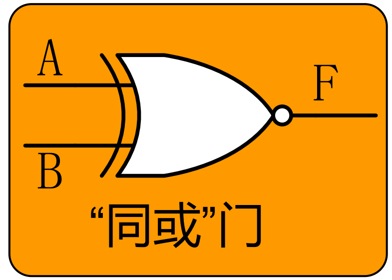
推广到多个变量：

若$A\oplus B\oplus \mathrm{C}\oplus D=F$成立，则$\mathrm{A}\oplus F\oplus\mathrm{C}\oplus\mathrm{D}=\mathrm{B}$成立

复合运算

复合运算3：“同或”

定义： $F = A \odot B = AB + \overline{A}\,\overline{B}$



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

扩展到多变量：

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

同或运算满足如下基本规律：

1）交换律 $A \odot B = B \odot A$

2）结合律 $A \odot B \odot C = A \odot (B \odot C)$

3）分配律 $A + (B \odot C) = (A + B) \odot (A + C)$

自行证明

复合运算

“同或”运算的特性

特性1：多变量“同或”运算的结果取决于变量中取值为“0”的变量的个数，而与取值为“1”的变量的个数无关。若取值为“0”的变量的个数是偶数，则“同或”的结果为“1”；若取值为“0”的变量的个数为奇数，则“同或”的结果为“0”。

由特性1可得：“同或”函数中任意一个输入变量取反，将导致运算结果取反

特性2：“同或”运算具有因果互换关系。即，等式两边的逻辑变量可以互相交换位置而仍然保持等式的成立。

若$A\odot B=C$成立，则$\mathrm{A}\odot\mathrm{C}=\mathrm{B}$成立；$\mathrm{B}\odot\mathrm{C}=\mathrm{A}$成立。

推广到多个变量：

若$A\odot B\odot \mathrm{C}\odot D=F$成立，则$\mathrm{A}\odot F\odot\mathrm{C}\odot\mathrm{D}=\mathrm{B}$成立

逻辑运算符号的完备性

由“与”、“或”、“非”三种基本逻辑运算可以组成任意逻辑函数

“与”、“或”、“非”是一组逻辑功能完备的逻辑运算符

“与非”、“或非”、“与或非”运算各自都是功能完备的逻辑运算符

“与非”、“或非”、“与或非”中的任何一种运算都能单独实现“与”、“或”、“非”这三种基本逻辑运算，都可单独组成任何一个逻辑函数

例：用“或非”门实现“与”、“或”、“非”运算

“与” $F = A \bullet B = \overline{\overline{A \bullet B}} = \overline{\overline{A} + \overline{B}} = \overline{\overline{A+0} + \overline{B+0}} = \overline{\overline{A+A} + \overline{B+B}}$

“或” $F = A + B = \overline{\overline{A+B}} = \overline{\overline{A+B}+0} = \overline{\overline{A+B}+\overline{A+B}}$

“非” $F = \overline{A} = \overline{A+A} = \overline{A+0}$

逻辑函数的标准形式

一个逻辑函数可以有多种表达式形式

- “与或”表达式
- “或与”表达式
- “与非-与非”表达式
- “或非-或非”表达式
- “与或非”表达式

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

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

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

$$= (\bar{A}+B)(A+C)$$ “或与”表达式

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

$$= \overline{\overline{AB} \bullet \overline{\bar{A}C}}$$ “与非-与非”表达式

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

$$= \overline{\overline{\bar{A}+B} + \overline{A+C}}$$ “或非-或非”表达式

$$= \overline{A\bar{B} + \bar{A}\bar{C}}$$ “与或非”表达式

有没有相对唯一的表达式?

最大项和最小项

最小项：n个变量的最小项是n个变量的积，其中每一个变量都以原变量或反变量的形式出现一次，且仅出现一次。

最大项：n个变量的最大项是n个变量的和，其中每一个变量都以原变量或反变量的形式出现一次，且仅出现一次。

n个变量则有2^n个最小项和2^n个最大项

三变量最小项真值表

No.	变量值 A B C	m_0 $\bar{A}\bar{B}\bar{C}$	m_1 $\bar{A}\bar{B}C$	m_2 $\bar{A}B\bar{C}$	m_3 $\bar{A}BC$	m_4 $A\bar{B}\bar{C}$	m_5 $A\bar{B}C$	m_6 $AB\bar{C}$	m_7 ABC
0	0 0 0	1	0	0	0	0	0	0	0
1	0 0 1	0	1	0	0	0	0	0	0
2	0 1 0	0	0	1	0	0	0	0	0
3	0 1 1	0	0	0	1	0	0	0	0
4	1 0 0	0	0	0	0	1	0	0	0
5	1 0 1	0	0	0	0	0	1	0	0
6	1 1 0	0	0	0	0	0	0	1	0
7	1 1 1	0	0	0	0	0	0	0	1

最小项的性质

一 对于任意一个最小项，只有一组变量的取值使它的值为1，而当变量取其它各组值时该最小项的值都是0。

二 任意两个不同的最小项的乘积（相“与”）恒为0。

三 全体最小项之和（相“或”）恒为1。

三变量最大项真值表

No.	变量值 A B C	M_7 $\bar{A}+\bar{B}+\bar{C}$	M_6 $\bar{A}+\bar{B}+C$	M_5 $\bar{A}+B+\bar{C}$	M_4 $\bar{A}+B+C$	M_3 $A+\bar{B}+\bar{C}$	M_2 $A+\bar{B}+C$	M_1 $A+B+\bar{C}$	M_0 $A+B+C$
0	0 0 0	1	1	1	1	1	1	1	0
1	0 0 1	1	1	1	1	1	1	0	1
2	0 1 0	1	1	1	1	1	0	1	1
3	0 1 1	1	1	1	1	0	1	1	1
4	1 0 0	1	1	1	0	1	1	1	1
5	1 0 1	1	1	0	1	1	1	1	1
6	1 1 0	1	0	1	1	1	1	1	1
7	1 1 1	0	1	1	1	1	1	1	1

最大项的性质

一

对于任意一个最大项，只有一组变量的取值使它的值为0，而当变量取其它各组值时该最大项的值都是1。

二

任意两个不同的最大项的和（相“或”）恒为1。

三

全体最大项之积（相“与”）恒为0。

最大项和最小项的关系

变量数相同，编号相同的最小项和最大项之间存在互补关系。

$$\overline{m_i^n} = M_i^n \quad \text{或者} \quad \overline{M_i^n} = m_i^n$$

逻辑函数的两种标准形式

标准表达式1：最小项之和式

由若干个最小项相“加”（相“或”）而构成，也叫标准“与或”式。

例如：

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

可以简写为：

$$F(A,B,C)=m_3+m_5+m_6=\sum m(3,5,6)=\sum(3,5,6)$$

例题：把 $F(A,B,C)=AB+\overline{A}C$ 展开为最小项之和式

解：

$$\begin{aligned}F(A,B,C)&=AB+\overline{A}C\\&=AB(C+\overline{C})+\overline{A}C(B+\overline{B})\\&=ABC+AB\overline{C}+\overline{A}BC+\overline{A}\,\overline{B}C\\&=m_7+m_6+m_3+m_1\\&=\sum m(1,3,6,7)\end{aligned}$$

任何一个逻辑函数表达式都可以被展开成唯一的最小项之和式

逻辑函数的两种标准形式

标准表达式2：最大项之积式

由若干个最大项相“乘”（相“与”）而构成，也叫标准“或与”式。

例如：

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

可以简写为：

$$G(A,B,C)=M_0\bullet M_3\bullet M_5\bullet M_6=\prod M(0,3,5,6)=\prod(0,3,5,6)$$

例题：把 $F(A,B,C)=(A+B)(\overline{A}+C)$ 展开为最大项之积式

解：
$$\begin{aligned}F(A,B,C)&=(A+B)(\overline{A}+C)\\&=(A+B+C\overline{C})(\overline{A}+C+B\overline{B})\\&=(A+B+C)(A+B+\overline{C})(\overline{A}+B+C)(\overline{A}+\overline{B}+C)\\&=M_0\bullet M_1\bullet M_4\bullet M_6\\&=\prod M(0,1,4,6)\end{aligned}$$

任何一个逻辑函数表达式都可以被展开成唯一的最大项之积式

逻辑函数的两种标准形式

由真值表写逻辑表达式

最小项之和式（找取值为1的最小项）

$$F(A,B,C)=\sum m(3,5,6,7)$$

最大项之积式（找取值为0的最大项）

$$G(A,B,C)=\prod M(0,1,2,4)$$

三人表决逻辑真值表

No.	A B C	F
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

两种标准表达式之间的关系

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

两种标准表达式之间的关系

两种表达式所含的编号是互相补充的，即最大项之积式中的最大项编号正好是最小项之和式中未包含的号码，反之亦然。

$$若F(A,B,C,D)=\sum m(0,2,3,5,7,9,12)$$

$$则F(A,B,C,D)=\prod M(1,4,6,8,10,11,13,14,15)$$

反函数的标准表达式

给定 F，可直接写出 $\overline{F}$ 的两种标准表达式

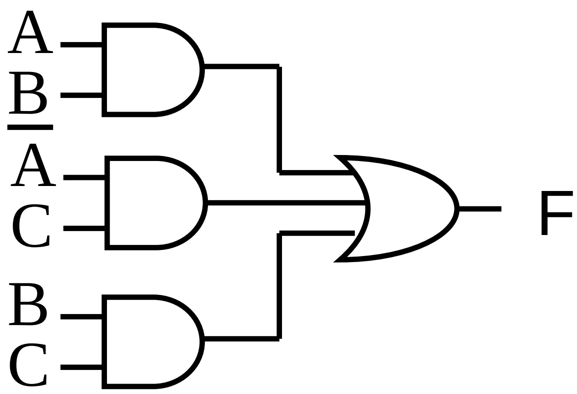
例 $F=\sum(1,3,6,7)$

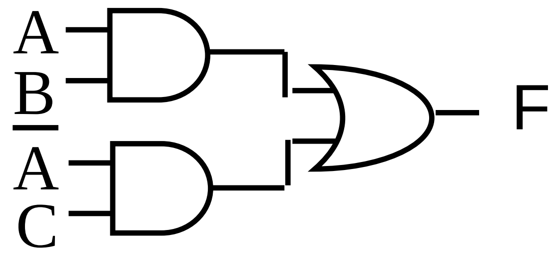
因为$F+\overline{F}=1$，所以$\overline{F}=\sum(0,2,4,5)$

$\overline{F}=\overline{\sum(1,3,6,7)}=\prod(1,3,6,7)$

逻辑函数的化简

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





为什么要化简逻辑函数?

节省门电路

减少输入端数量

提高电路稳定性

逻辑函数的化简

五种常用最简形式

一
- “与或”表达式：$F=AB+AD$

二
- “或与”表达式：$F=(A+B)(C+D)$

三
- “与非-与非”表达式：$F=\overline{\overline{AB}\ \overline{CD}}$

四
- “或非-或非”表达式：$F=\overline{\overline{A+B}+\overline{C+D}}$

五
- “与或非”表达式：$F=\overline{AB+CD}$

逻辑函数的化简

本节主要介绍“与或”表达式的化简

- 任何一个逻辑函数表达式都能展开成一个“与或”表达式
- 从一个最简“与或”表达式，很容易得到“与非-与非”、“与或非”等其他形式的表达式
- 只要掌握了“与或”表达式的化简方法，利用对偶式，不难化简“或与”表达式

最简“与或”表达式应该满足如下两个条件

- 表达式中乘积项的个数应该是最少的
- 每一个乘积项中所含的变量个数最少

逻辑函数的化简

代数化简法

卡诺图化简法

自学内容

系统化简法（Q-M法）

自学，也就是说不考喽

逻辑函数的代数化简法

代数化简法

反复运用基本公式和常用公式消去多余项和多余因子，以便求得最简表达式。

- 并项法
- 消项法
- 消元法
- 配项法

逻辑函数的代数化简法

并项法： **利用合并定理** $AB+A\bar{B}=A$ **和互补律** $A+\bar{A}=1$ **将两项合并成一项，同时消去一个“因子”（变量）。**

例

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

逻辑函数的代数化简法

消项法： **利用吸收定理** $A+AB=A$ **和添加项定理** $AB+\overline{A}C+BC=AB+\overline{A}C$ **消去多余的项。**

例

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

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

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

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

逻辑函数的代数化简法

消元法： 利用吸收定理 $A + \bar{A}B = A + B$ 消去多余的项。

例

$$
\begin{aligned}
F &= C\bar{D} + \bar{C}E + DE + \bar{A}B\bar{E} \\
&= C\bar{D} + (\bar{C} + D)E + \bar{A}B\bar{E} \\
&= C\bar{D} + \overline{C\bar{D}}E + \bar{A}B\bar{E} \\
&= C\bar{D} + E + \bar{A}B\bar{E} \\
&= C\bar{D} + E + \bar{A}B
\end{aligned}
$$

逻辑函数的代数化简法

配项法： **利用互补律**$A+\overline{A}=1$**，重叠率** $A+A=A$ **和添加项定理** $AB+\overline{A}C+BC=AB+\overline{A}C$ **等，在逻辑表达式中先添项目，再消项。**

例

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

逻辑函数的代数化简法

化简$Y = ABC + ABD + \overline{A}B\overline{C} + CD + B\overline{D}$

$$Y = ABC + ABD + \overline{A}B\overline{C} + CD + B\overline{D} = ABC + \overline{A}B\overline{C} + CD + B(AD + \overline{D})$$

$$= ABC + \overline{A}B\overline{C} + CD + B(A + \overline{D}) = ABC + \overline{A}B\overline{C} + CD + BA + B\overline{D}$$

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

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

$$= BA + B(\overline{C} + \overline{D}) + CD = BA + B\overline{CD} + CD$$

$$= BA + B + CD = B + CD$$

“或与”表达式的化简

化简标准

- “或项”（“和”项）最少
- 每个“或项”所含的变量个数最少

化简方法

- 利用逻辑代数基本定律和基本定理
- 利用对偶式，先将“或与”表达式转换成“与或”表达式，化简后再转换成“或与”表达式

$$
\begin{array}{ccc}
\text{F(“或与”式)} & \xrightarrow{\text{求对偶式}} & \text{F’（“与或”式）} \\
 & & \big\downarrow \text{化简} \\
\text{F(最简“或与”式)} & \xleftarrow{\text{求对偶式}} & \text{F’（最简“与或”式）}
\end{array}
$$

"或与"表达式的化简

例：求以下函数的最简"或与"式

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

1) 求F' $F'=AC+\overline{B}C+B\overline{D}+C\overline{D}+AB+A\overline{C}+\overline{A}BC\overline{D}+A\overline{B}DE$

2) 化简F'

$$\begin{aligned}
F' &= AC+\overline{B}C+B\overline{D}+C\overline{D}+AB+A\overline{C}+\overline{A}BC\overline{D}+A\overline{B}DE \\
&= (AC+A\overline{C}+AB+A\overline{B}DE)+\overline{B}C+B\overline{D}+(C\overline{D}+\overline{A}BC\overline{D}) \\
&= A(C+\overline{C}+B+\overline{B}DE)+\overline{B}C+B\overline{D}+C\overline{D} \\
&= A(1+B+\overline{B}DE)+\overline{B}C+B\overline{D}+C\overline{D} \\
&= A+\overline{B}C+B\overline{D}
\end{aligned}$$

3) 求(F')',即F $F=(F')'=A(\overline{B}+C)(B+\overline{D})$

最简“与非-与非”表达式

化简标准

- 表达式的“非”号最少（不包含单个变量上的“非”号）
- 每个“非”号下的变量个数最少

化简方法

- 用“求反加非”和反演律将已化简的“与或”式变换为最简“与非-与非”表达式

最简“与非-与非”表达式

例：用最少的“与非”门实现 $F=\bar{A}\bar{B}+\bar{A}BD+A\bar{B}\bar{D}$

解：先求F的最简“与或”式：$F=\bar{A}\bar{B}+\bar{A}BD+A\bar{B}\bar{D}$

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

用“取反加非”的方式将F变成最简“与非-与非”式

$$F=\overline{\overline{F}}=\overline{\overline{\bar{A}D+\bar{B}\bar{D}}}=\overline{\overline{\bar{A}D}\bullet\overline{\bar{B}\bar{D}}}$$

最简“或非-或非”表达式

化简标准

- 表达式的“非”号最少（不包含单个变量上的“非”号）
- 每个“非”号下的变量个数最少

化简方法

- 用“求反加非”和反演律将已化简的“或与”式变换为最简“或非-或非”表达式

最简“或非-或非”表达式

例：用最少的“或非”门实现 $F=\bar{A}\bar{B}+\bar{A}BD+A\bar{B}\bar{D}$

解：先求F的最简“或与”式

$$F=(\bar{A}+\bar{D})(\bar{B}+D) \qquad \text{（过程略）}$$

用“取反加非”的方式将F变成最简“或非-或非”式

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

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

最简“与或非”表达式

化简标准

- “与”项的个数最少
- 每个“与”项所含的变量个数最少

化简方法

- 对 $\overline{F}$ 的最简“与或”式“求反”，就得到F的最简“与或非”式

最简“与或非”表达式

例：求 $F = \overline{A}\,\overline{B} + \overline{A}BD + A\overline{B}\,\overline{D}$ 的最简“与或非”式

解：先求 $\overline{F}$ 的最简“与或”式

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

所以F的最简“与或非”式为

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

常用变换方法

- F的最简“与或”式 $\xrightarrow{\text{求反加非}}$ F的最简“与非-与非”式
- $\overline{F}$ 的最简“与或”式 $\xrightarrow{\text{反演}}$ F的最简“或与”式
- F的最简“或与”式 $\xrightarrow{\text{求反加非}}$ F的最简“或非-或非”式
- $\overline{F}$ 的最简“或与”式 $\xrightarrow{\text{反演}}$ F的最简“与或”式
- $\overline{F}$ 的最简“与或”式 $\xrightarrow{\text{加一非}}$ F的最简“与或非”式

逻辑函数的卡诺图化简法

逻辑相邻最小项（“逻辑相邻项”或“相邻项”）

任意两个变量个数相同的最小项，如果组成它们的各个变量（原变量或反变量）中，只有一个变量互补（互反）而其余变量均相同（同为原变量或反变量）时，就称这两个最小项是逻辑相邻的最小项，简称“逻辑相邻项”或“相邻项”。

$AB \qquad A\bar{B}$ √

$AB\bar{C}D \qquad A\bar{B}\bar{C}D$ √

$A\bar{B}\bar{C} \qquad A\bar{B}C$ √

$A\bar{B}\bar{C}D \qquad \bar{A}\bar{B}CD$ ✗

两个逻辑相邻的最小项相“或”，结果将产生一个“与”项并同时消掉一个变量。如：

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

卡诺图的构成

A \ B	0	1
0	$\bar{A}\bar{B}$ 0	$\bar{A}B$ 1
1	$A\bar{B}$ 2	AB 3

二变量卡诺图

A \ BC	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$ 0	$\bar{A}\bar{B}C$ 1	$\bar{A}BC$ 3	$\bar{A}B\bar{C}$ 2
1	$A\bar{B}\bar{C}$ 4	$A\bar{B}C$ 5	ABC 7	$AB\bar{C}$ 6

三变量卡诺图

最小项编号

AB \ CD	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$ 0	$\bar{A}\bar{B}\bar{C}D$ 1	$\bar{A}\bar{B}CD$ 3	$\bar{A}\bar{B}C\bar{D}$ 2
01	$\bar{A}B\bar{C}\bar{D}$ 4	$\bar{A}B\bar{C}D$ 5	$\bar{A}BCD$ 7	$\bar{A}BC\bar{D}$ 6
11	$AB\bar{C}\bar{D}$ 12	$AB\bar{C}D$ 13	$ABCD$ 15	$ABC\bar{D}$ 14
10	$A\bar{B}\bar{C}\bar{D}$ 8	$A\bar{B}\bar{C}D$ 9	$A\bar{B}CD$ 11	$A\bar{B}C\bar{D}$ 10

四变量卡诺图

卡诺图的构成

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

A=0

BC \ DE	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

A=1

五变量卡诺图

卡诺图的特点

- n变量的卡诺图有2^n个小方格，每个小方格表示一个最小项。
- 变量按行、列分为两组，每组变量的取值按格雷码排列。
- 卡诺图任一行或一列的两端上的小方格所代表的最小项在逻辑上是相邻的。在几何空间上应该把卡诺图看成上下、左右循环连接的图形，如同一个封闭球面。
- 对于变量个数大于四个的情形，仅用二维几何空间的位置相邻性已经不能完全地表示最小项的逻辑相邻性
- 卡诺图不止有一种形式。

逻辑函数的卡诺图表示

对于给定的逻辑函数，如何用卡诺图表示?

第一步

- 根据逻辑函数中变量的个数画出相应的卡诺图框

第二步

- 根据给定函数的表达式形式（各种表达形式或真值表）填写卡诺图框图（往卡诺图上小方格里填写“1”或“0”）

逻辑函数为最小项之和式

在卡诺图中，将表达式中所有的最小项所对应的小方格里都填写“1”，其余小方格里都填写“0”。

$$F=\overline{A}\,\overline{B}\,\overline{C}+\overline{A}B\overline{C}+A\overline{B}C+ABC=\sum m(0,2,5,7)$$

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

简化画法

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

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

逻辑函数为最大项之积式

将卡诺图上编号与表达式中最大项编号相同的小方格里都填写“0”，而其余的小方格里都填写“1”。

$$F=(A+B+\overline{C})(A+\overline{B}+\overline{C})(\overline{A}+B+C)(\overline{A}+\overline{B}+C)=\prod M(1,3,4,6)$$

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

三变量最小项m_7和最大项M_7的卡诺图

$$m_7 : ABC \qquad M_7 : \overline{A}+\overline{B}+\overline{C}$$

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

m_7的卡诺图

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

M_7的卡诺图

从图中“1”的个数看，m_7只有一个格为“1”，而M_7除一个格为“0”外，其余均为“1”，这就是最大项和最小项名称的由来。

上图证明了最大项和最小项的互补关系。

逻辑函数为真值表形式

若卡诺图上的某个小方格所代表的最小项，在真值表里所对应的函数F取值为“1”，则该小方格里填“1”；否则就填“0”。

序号	A	B	C	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

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

逻辑函数为一般“与或”式

将“与或”式中所有“与”项在卡诺图中所覆盖的区域内的所有小方格都填“1”（已经填过“1”的小格除外），其余都填“0”。

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

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

逻辑函数为一般“或与”式

将“或与”式中所有“或”项在卡诺图中所覆盖的区域内的所有小方格都填“0”（已经填过“0”的小格除外），其余都填“1”。

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

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

逻辑函数为其他形式的逻辑表达式

将这些表达式转化为“与或”式或者“或与”式，再填写卡诺图

例1： $F = \overline{A}B \bullet \overline{AC} + \overline{B}$

$$
\begin{aligned}
F &= \overline{A}B \bullet \overline{AC} + \overline{B} \\
&= \overline{A}B(\overline{A} + \overline{C}) + \overline{B} \\
&= \overline{A}B + \overline{A}B\overline{C} + \overline{B}
\end{aligned}
$$

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

逻辑函数为其他形式的逻辑表达式

将这些表达式转化为“与或”式或者“或与”式，再填写卡诺图

例2： $F=\overline{\overline{A}\,\overline{B}}\bullet\overline{AC}+\overline{C}$

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

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

卡诺图的性质

性质1：若F的卡诺图中所有的小格都填“1”，则F=1。

性质2：若F的卡诺图中所有的小格都填“0”，则F=0。

性质3：若将F的卡诺图中所有小格内的“0”都换成“1”、“1”都换成“0”，则得到$\overline{F}$的卡诺图。（卡诺图反演）

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

函数F的卡诺图

$\xrightarrow{求反}$

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

补函数$\overline{F}$的卡诺图

卡诺图的运算-“乘”

卡诺图的相“乘”（“与”）运算：若函数F为某两个函数F_1和F_2相“与”而构成，则F的卡诺图等于F_1的卡诺图和F_2的卡诺图的“与”。即：

若$F=F_1 \cdot F_2$，则$K(F)=K(F_1) \cdot K(F_2)$

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

F_1的卡诺图

•

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

F_2的卡诺图

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

$F_1 \cdot F_2$的卡诺图

卡诺图的运算-“加”

卡诺图的相“加”（“或”）运算：两个函数相“或”的卡诺图等于这两个函数各自的卡诺图相“或”。即：

若$F=F_1+F_2$，则$K(F)=K(F_1)+K(F_2)$

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

F_1的卡诺图

+

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

F_2的卡诺图

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

F_1+F_2的卡诺图

卡诺图的运算-“异或”

卡诺图的 “异或” 运算：两个函数相“异或”，其卡诺图等于这两个函数各自的卡诺图相“异或”。即：

若$F=F_1 \oplus F_2$，则$K(F)=K(F_1) \oplus K(F_2)$

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

F_1的卡诺图

$\oplus$

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

F_2的卡诺图

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

$F_1 \oplus F_2$的卡诺图

卡诺图的运算（例）

例：求函数 $F=(\bar{A}+B+C)\overline{AB+BC+AC}+[(\bar{A}+B)(\bar{A}+C)(B+C)\oplus\bar{A}\bar{C}]$ 的卡诺图。

解：利用卡诺图运算，先分解函数

设

$$F_1=\bar{A}+B+C$$

$$F_2=AB+BC+AC$$

$$F_3=(\bar{A}+B)(\bar{A}+C)(B+C)$$

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

则

$$F=F_1\bullet\overline{F_2}+[F_3\oplus F_4]$$

卡诺图的运算（例）

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

F_2的卡诺图

↓ 求反

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

$\overline{F_2}$的卡诺图

•

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

F_1的卡诺图

=

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

$F_1 \cdot \overline{F_2}$的卡诺图

卡诺图的运算（例）

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

F_3的卡诺图

$\oplus$

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

F_4的卡诺图

$=$

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

$F_3 \oplus F_4$的卡诺图

$+$

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

$F_1 \cdot \overline{F_2}$的卡诺图

$=$

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

F的卡诺图

最小项合并规律

把逻辑相邻的最小项合并（相“或”）在一起，结果将产生一个“与”项，并消去一个逻辑变量

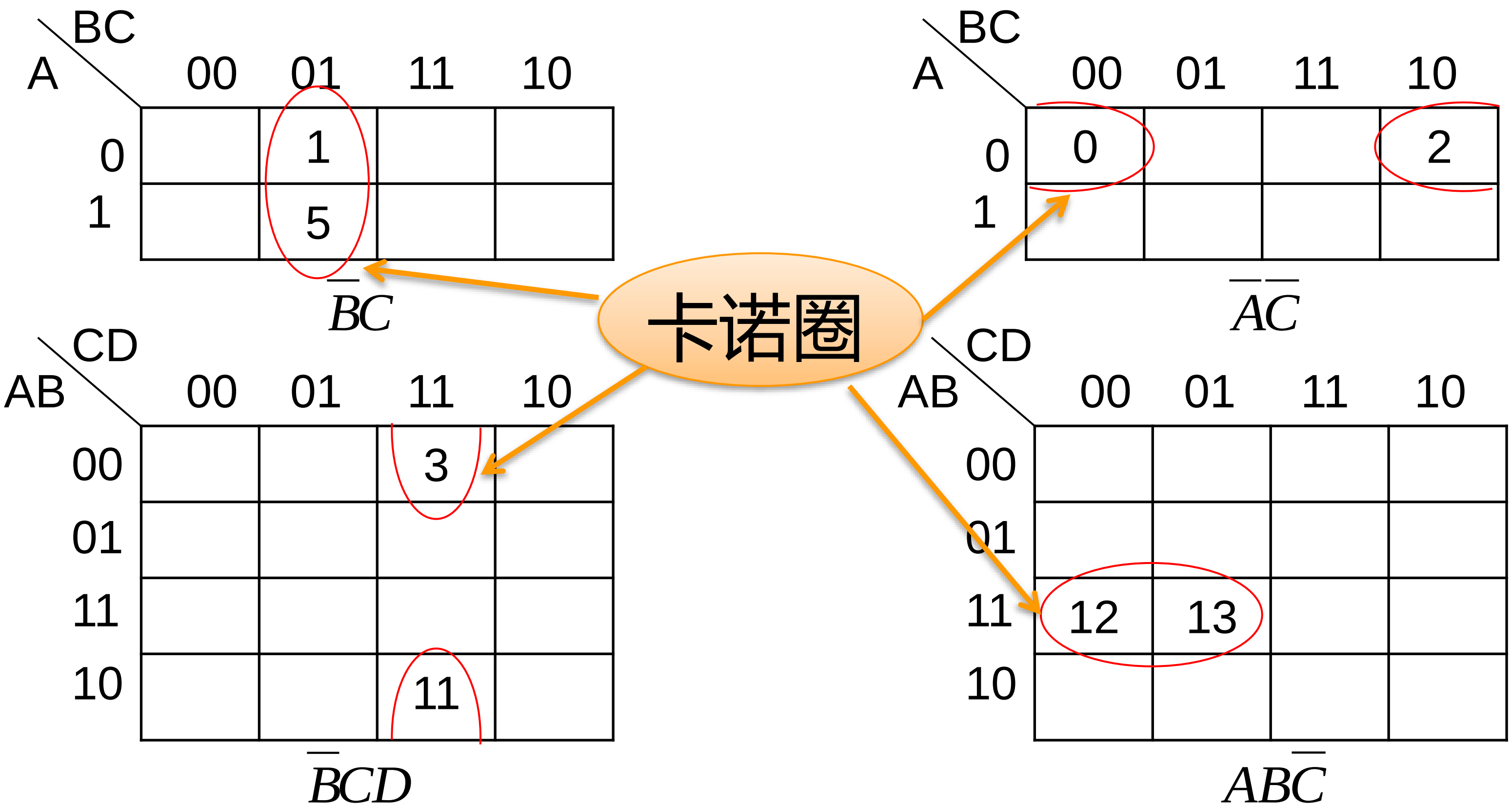
$$\sum m^2(2,3) = A\overline{B} + AB = A(\overline{B} + B) = A$$

$$\sum m^3(0,4) = \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}\,\overline{C} = (\overline{A} + A)\overline{B}\,\overline{C} = \overline{B}\,\overline{C}$$

$$\sum m^4(12,13) = AB\overline{C}\,\overline{D} + AB\overline{C}D = AB\overline{C}(\overline{D} + D) = AB\overline{C}$$

最小项合并规律

在卡诺图上将两个相邻的小格“圈”在一起，相当于把这两个小格所代表的“逻辑相邻”的最小项合并（相“或”）在一起，从而产生一个“与项”并消去一个互补的变量。



最小项合并规律

卡诺图上四个相邻的最小项合并（相“或”）在一起，从而产生一个“与项”并消去两个变量。

$$\sum m4(0,2,8,10) = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}C\overline{D} + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}C\overline{D}$$

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

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

AB \ CD	00	01	11	10
00	0	1	3	2
01				
11				
10				

$\overline{A}\,\overline{B}$

AB \ CD	00	01	11	10
00				
01	4			6
11	12			14
10				

$B\overline{D}$

AB \ CD	00	01	11	10
00	0			2
01				
11				
10	8			10

$\overline{B}\,\overline{D}$

最小项合并规律

以此类推卡诺图上8个相邻的最小项可以合并成一个“与项”并消去三个变量……

AB \ CD	00	01	11	10
00				
01	4	5	7	6
11	12	13	15	14
10				

B

AB \ CD	00	01	11	10
00	0			2
01	4			6
11	12			14
10	8			10

$\overline{D}$

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

$$\sum_{i=0}^{15} m_i^4 = 1$$

五变量最小项合并

BC \ DE	00	01	11	10
00	b 0	1	d 3	2
01	4	c 5	7	6
11	12	c 13	15	a 14
10	b 8	9	d 11	10

A=0

BC \ DE	00	01	11	10
00	16	17	d 19	18
01	20	c 21	23	22
11	28	c 29	31	a 30
10	24	25	d 27	26

A=1

- 在单一的四变量图中，相邻关系与四变量的卡诺图一样。
- 在两图之间，相应的小方格也同样相邻。

用卡诺图化简逻辑函数——求最简“与或”式

最简“与或”式化简：圈“1”合并

画卡诺框

填值

圈组合并（最小覆盖原则）

写“与或”式

最小覆盖原则

- 卡诺图上的每一个填“1”小格都要被卡诺圈所覆盖，也就是说，每一个“1”都至少被圈组合并一次。
- 在满足上一条件的情况下，卡诺图上卡诺圈的个数要尽量少。
- 每个卡诺圈所包含的填“1”小格的个数要尽量多，但必须是2^i（i=0,1,2,….,n）个；
- 每个卡诺圈都至少包含一个其他所有卡诺圈不包含的填“1”小格。即：每个卡诺圈都必须至少有一个独属于它自己的填“1”小格。

用卡诺图化简逻辑函数

例：用卡诺图化简逻辑函数

$$F(A,B,C,D)=\sum m(0,1,2,3,4,8,10,11,14,15)$$

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

$$L_1=\overline{A}\,\overline{B}$$

$$L_2=AC$$

$$L_3=\overline{B}\,\overline{D}$$

$$L_4=\overline{A}\,\overline{C}\,\overline{D}$$

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

圈组合并时注意事项

画卡诺圈时，要求“圈”的个数尽可能少

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

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

圈法不正确

VS.

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

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

圈法正确

圈组合并时注意事项

画卡诺圈时，要求被“圈”的小格尽可能多

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

$$F = \overline{A} + ABC$$

圈法不正确

VS.

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

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

圈法正确

圈组合并时注意事项

卡诺圈中所圈小格的个数必须符合2^i的形式，只有相邻的小格才能被圈组合并在一起

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

圈法正确

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

圈法不正确

圈组合并时注意事项

圈组合并顺序是“先多后少”,不要出现多余的卡诺圈

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

多余的卡诺圈

圈组合并时注意事项

例：用卡诺图法化简逻辑函数

$$F(A,B,C,D)=\sum m(0,3,4,5,6,7,9,12,14,15)$$

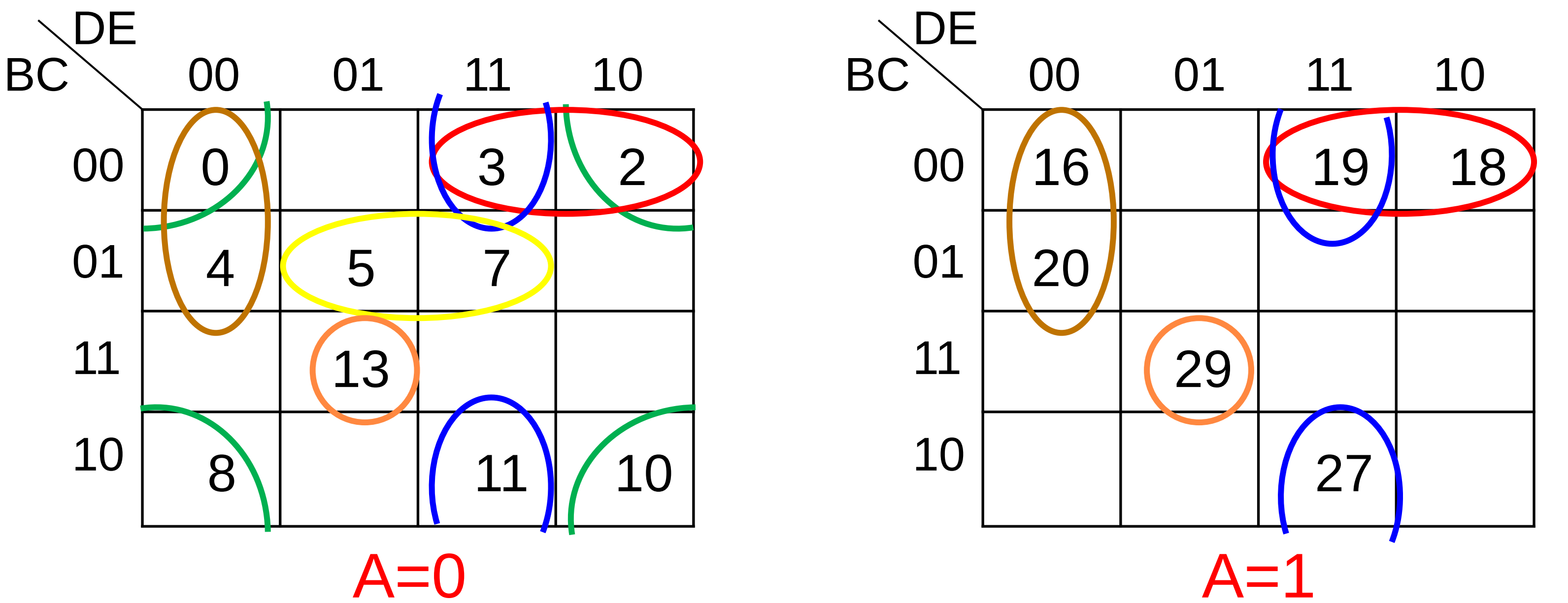
AB \ CD	00	01	11	10
00	1	0	1	0
01	1	1	1	1
11	1	0	1	1
10	0	1	0	0

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

五变量逻辑函数化简

例：用卡诺图法化简逻辑函数

$$F(A,B,C,D,E)=\sum m(0,2,3,4,5,7,8,10,11,13,16,18,19,20,27,29)$$



$$F(A,B,C,D,E)=\overline{A}\,\overline{C}\,\overline{E}+\overline{B}\,\overline{D}\,\overline{E}+\overline{B}\,\overline{C}D+\overline{C}DE+\overline{A}\,\overline{B}CE+BC\overline{D}E$$

用卡诺图化简逻辑函数——求最简“或与”式

方法一：利用函数F的反函数求最简“或与”式

圈“0”求$\overline{F}$的最简“与或”式。对$\overline{F}$进行反演运算，就得到F的最简“或与”式。

例：求$F(A,B,C,D)=\sum m(5,6,7,9,10,11,13,14,15)$ 的最简“或与”式

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

F的卡诺图

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

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

pyro_yangxu@bit.edu.cn

用卡诺图化简逻辑函数——求最简“或与”式

方法二：直接圈“0”写“或”项得到函数F的最简“或与”式

例：求$F(A,B,C,D)=(A+\overline{B}+C+D)(\overline{A}+B)(\overline{A}+\overline{C})\overline{C}$的最简“或与”式

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

F的卡诺图

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

多输出逻辑函数的卡诺图化简法

利用各函数间的公共项，达到整体化简的目的

例：化简如下两个逻辑函数

$$F_1 = \overline{A}BC + AB\overline{C} + ABC$$

$$F_2 = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC$$

（1）分别化简F1和F2

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

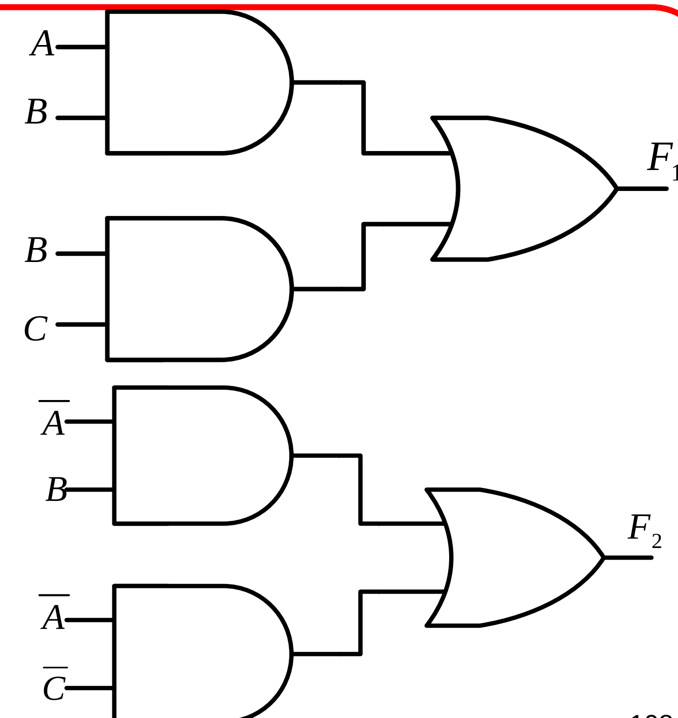
F_1的卡诺图

$$F_1 = AB + BC$$

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

F_2的卡诺图

$$F_2 = \overline{A}\,\overline{C} + \overline{A}B$$



多输出逻辑函数的卡诺图化简法

利用各函数间的公共项，达到整体化简的目的

例：化简如下两个逻辑函数

$$F_1 = \bar{A}BC + AB\bar{C} + ABC$$

$$F_2 = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC$$

（2）利用公共项整体化简

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

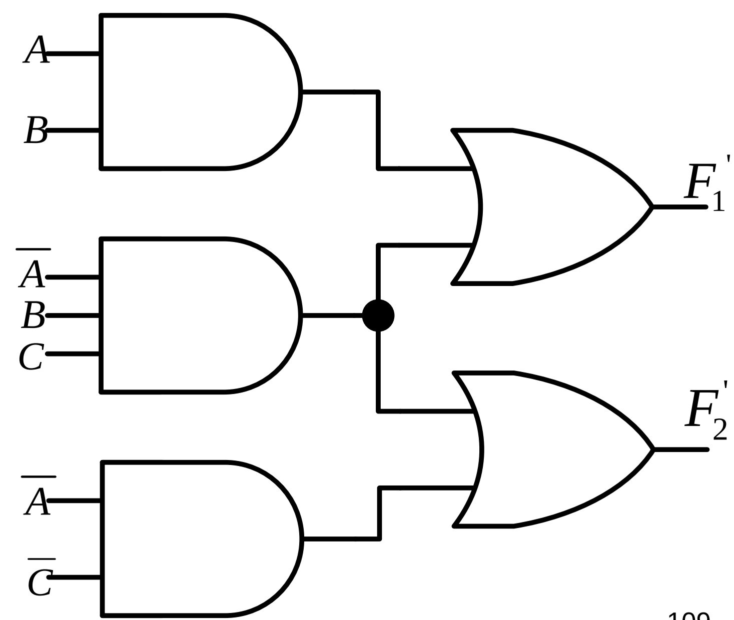
F_1的卡诺图

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

F_2的卡诺图

$$F_1' = AB + \bar{A}BC$$

$$F_2' = \bar{A}\bar{C} + \bar{A}BC$$



非完全描述逻辑函数

完全描述逻辑函数

- 对输入变量的每一组取值都有确定的函数值F（“1”或“0”）与之对应

非完全描述逻辑函数

- 逻辑函数不是被完全定义的，对于某些组变量的取值，函数有确定的取值与之对应；而对于另外其他组变量的取值，函数没有确定的取值与之对应。

非完全描述逻辑函数的表示

任意项（无关项）：使函数值为任意值（仅限于“0”和“1”）的变量取值所对应的最小项。用小写希腊字母φ或d_i表示；与无关项对应的函数值用“x”或“d”表示。

$$F(A,B,C)=\sum m(0,2,7)+\sum \varphi(3,4)$$

$$F(A,B,C)=\sum m(0,2,7)+\sum d(3,4)$$

$$\begin{cases} F(A,B,C)=\sum m(0,2,7) \\ \overline{A}BC+A\overline{B}\,\overline{C}=0 \end{cases}$$

约束条件

序号	A	B	C	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	x
4	1	0	0	x
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

非完全描述逻辑函数的表示

"非完全描述逻辑函数"也可以用最大项之积表示，最大无关项用大写ϕ或D_i表示。

$$F(A,B,C)=\prod M(1,5,6)\bullet\prod \phi(3,4)$$

$$F(A,B,C)=\prod M(1,5,6)\bullet\prod D(3,4)$$

$$\begin{cases} F(A,B,C)=\prod(1,5,6) \\ (A+\overline{B}+\overline{C})(\overline{A}+B+C)=1 \end{cases}$$

序号	A B C	F
0	0 0 0	1
1	0 0 1	0
2	0 1 0	1
3	0 1 1	x
4	1 0 0	x
5	1 0 1	0
6	1 1 0	0
7	1 1 1	1

代数法化简非完全描述逻辑函数

根据需要加进或者舍弃某些“无关最小项”，加进“无关最小项”的原则是使“逻辑相邻”的最小项个数最大化，以使得原逻辑表达式得到进一步的简化。

例：化简“一位十进制数质数探测器”逻辑函数

$$F=\sum m(2,3,5,7)+d(10,11,12,13,14,15)$$

1）先不考虑“无关最小项”

$$\begin{aligned}F&=\sum m(2,3,5,7)\\&=\overline{b_3}\,\overline{b_2}\,b_1\overline{b_0}+\overline{b_3}\,\overline{b_2}\,b_1b_0+\overline{b_3}\,b_2\overline{b_1}\,b_0+\overline{b_3}\,b_2b_1b_0\\&=\overline{b_3}\,\overline{b_2}\,b_1(\overline{b_0}+b_0)+\overline{b_3}\,b_2b_0(\overline{b_1}+b_1)\\&=\overline{b_3}\,\overline{b_2}\,b_1+\overline{b_3}\,b_2b_0\end{aligned}$$

代数法化简非完全描述逻辑函数

2）再考虑将“无关最小项”的d_{10}、d_{11}、d_{13}、d_{15}加入上式

$$
\begin{aligned}
F &= \sum m(2,3,5,7) + \sum d(10,11,13,15) \\
&= \overline{b_3}\,\overline{b_2}b_1 + \overline{b_3}b_2b_0 + b_3\overline{b_2}b_1\overline{b_0} + b_3\overline{b_2}b_1b_0 + b_3b_2\overline{b_1}b_0 + b_3b_2b_1b_0 \\
&= \overline{b_2}b_1(\overline{b_3} + b_3\overline{b_0} + b_3b_0) + b_2b_0(\overline{b_3} + b_3\overline{b_1} + b_3b_1) \\
&= \overline{b_2}b_1 + b_2b_0
\end{aligned}
$$

所有加进“与或”表达式的“无关最小项”，相当于确认他们对应的逻辑函数为“1”；舍弃的“无关最小项”，相当于确认它们对应的逻辑函数值为“0”。

用卡诺图化简非完全描述逻辑函数

例：化简“一位十进制数质数探测器”逻辑函数

$$F=\sum m(2,3,5,7)+d(10,11,12,13,14,15)$$

$b_3b_2 \backslash b_1b_0$	00	01	11	10
00	0	0	1	1
01	0	1	1	0
11	x	x	x	x
10	0	0	x	x

F的卡诺图

$$F=b_2b_0+\overline{b_2}b_1$$

用卡诺图化简非完全描述逻辑函数

例：求“一位十进制数质数探测器”逻辑函数F的补函数$\overline{F}$的最简“与或”式。

b_3b_2 \ b_1b_0	00	01	11	10
00	0	0	1	1
01	0	1	1	0
11	x	x	x	x
10	0	0	x	x

F的卡诺图

$$\overline{F} = b_2\overline{b_0} + \overline{b_2}\,\overline{b_1}$$

利用约束项化简非完全描述逻辑函数时，所得到的F和$\overline{F}$的最简“与或”式一定是互补函数吗？

用卡诺图化简非完全描述逻辑函数

例：求逻辑函数$F=\prod M(3,6,7,10,12)\bullet\prod \phi(2,4,11,13)$ 的最简“与或”式

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

F的卡诺图

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

F的卡诺图

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

“非完全描述逻辑函数”可能有多种不同的最简形式

逻辑函数的描述

例：有三个人A、B、C对一项提案进行表决。如果有两个或两个以上的人同意，则提案通过；否则，提案通不过。试用逻辑函数表示提案通过的条件。

以逻辑变量A、B、C代表三个人。如果某人同意提案，则相应的变量取“1”；否则取“0”。用逻辑变量F代表提案是否通过，“1”代表通过；“0”代表不通过。

描述方法一：真值表

序号	A	B	C	F
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

逻辑函数的描述

描述方法二：卡诺图

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

描述方法五：时序图（波形图）

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

描述方法三：逻辑表达式

$F = \sum m(3,5,6,7)$ (最小项之和式)

$= \prod M(0,1,2,4)$ (最大项之积式)

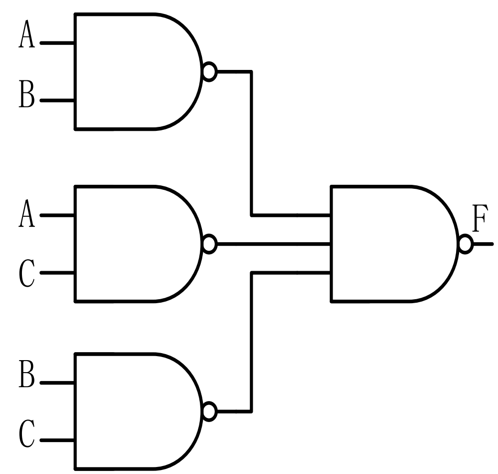
$= AB + AC + BC$ (最简“与或”式)

$= (A+B)(A+C)(B+C)$ (最简“或与”式)

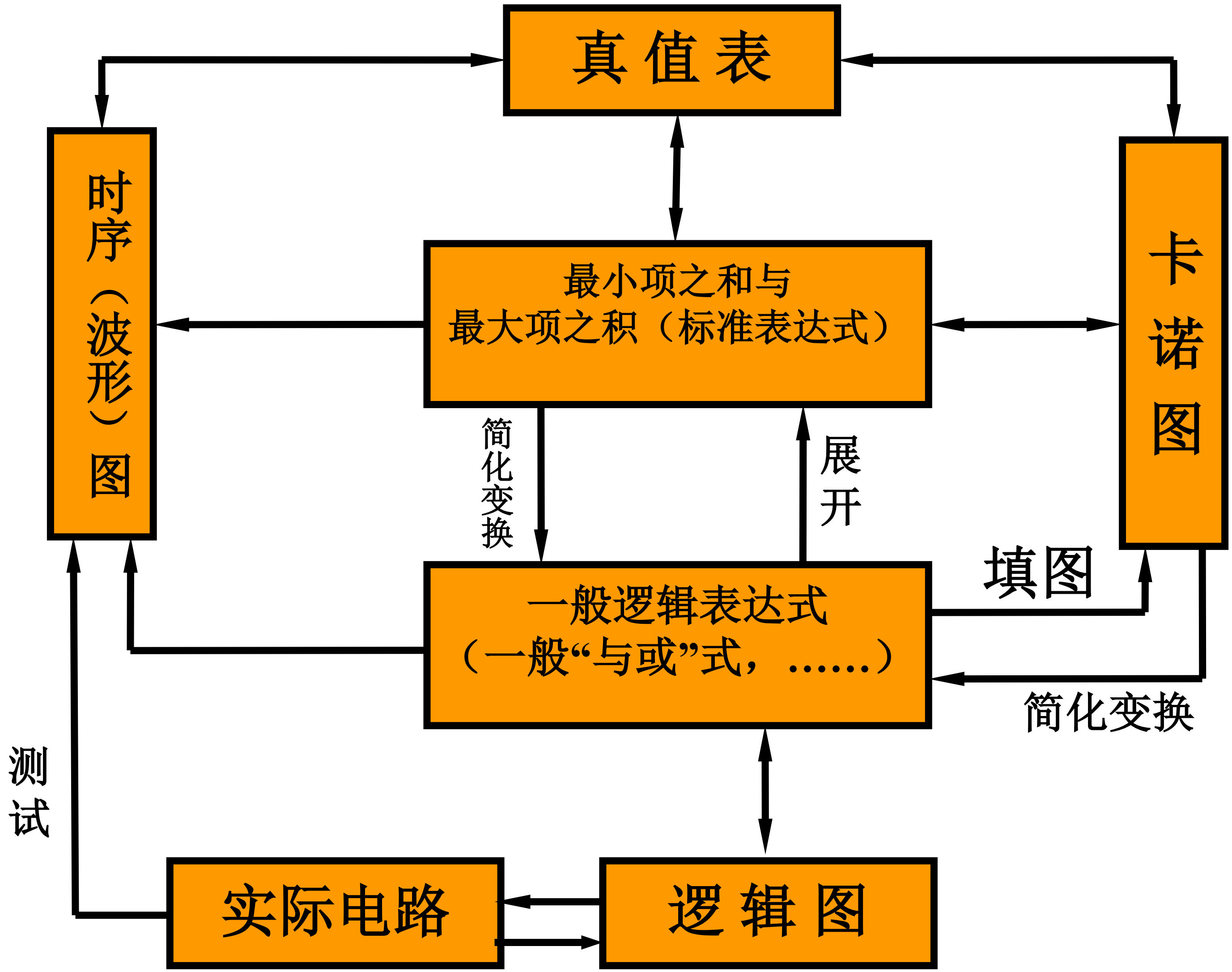
$= \overline{\overline{AB} \bullet \overline{AC} \bullet \overline{BC}}$ (最简“与非-与非”式)

……

描述方法四：逻辑图



逻辑函数描述方法之间的转换



第3章小结

- 逻辑函数的概念
- 基本运算：与、或、非
- 公理、定理、公式、性质
  - 三个基本定律
- 化简：代数法、卡诺图法
- 将函数化简为任意形式
- 非完全描述函数
- 各种描述方法之间的关系