第2章　逻辑函数及其简化

逻 辑 代 数 概 述

2.1　逻　辑　代　数

2.2　逻辑函数的简化

1849年英国数学家乔治 ·布尔首先提出描述客观事物逻辑关系的数学方法一布尔代数。1938年克劳德 ·香农将布尔代数应用到继电器开关电路的设计，故称为开关代数。随着数字技术的发展，布尔代数成为数字逻辑电路分析和设计的基础，又称为逻辑代数，在二值逻辑电路中得到广泛的应用。

2.1 逻辑代数

逻辑 —— 事物（条件、事件）之间的一种因果关系

2.1.1 基本逻辑

只有当决定事件发生的条件全部具备时，事件才会发生。

这就是与逻辑。常用“ ·”或∧、∩、&、and表示

只要当决定事件发生的条件中有一个或多个具备时，事件就会发生。

这就是或逻辑。常用“＋”或∨、∪、or表示

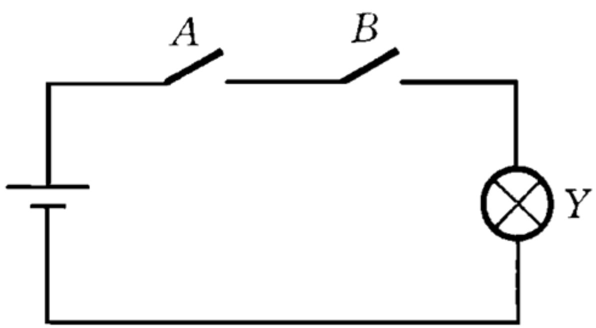
两个事件互为条件：

事件一发生时，事件二不会发生；事件一不发生时，事件二才会发生

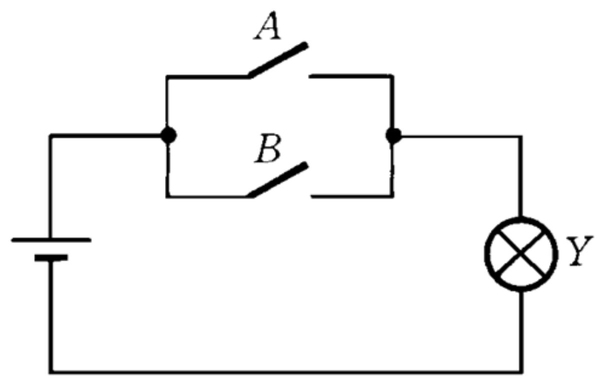
事件二发生时，事件一不会发生；事件二不发生时，事件一才会发生

这就是非逻辑。常用“-”或 no 表示

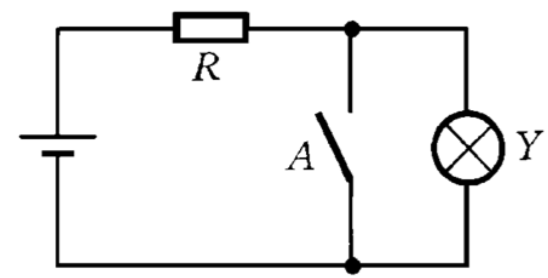
与（AND） 或（OR） 非（NOT）



(a)



(b)



(c)

以A=1表示开关A合上，A=0表示开关A断开；

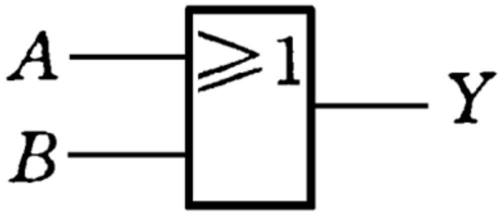
以Y=1表示灯亮，Y=0表示灯不亮；

三种电路的因果关系不同：

2.1.2 基本逻辑运算

逻辑加，又称为“或运算” Y＝A＋B 其意义在于：A或B中只要有一个为1，则函数值 Y 就为1。（有一出一）

逻辑加的逻辑符号是



或

运算规则是

0＋0＝0 A＋0＝A

0＋1＝1 A＋1＝1

1＋0＝1 A＋A＝A

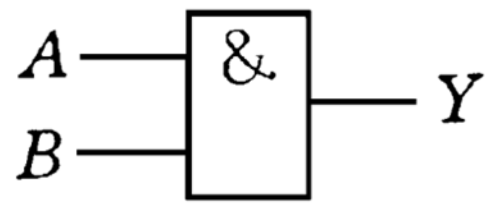
1＋1＝1

2.1.2 基本逻辑运算

逻辑乘，又称为“与运算” $Y=A\cdot B=AB$

其意义在于：只有A和B都为1时，函数值 Y 才为1。（全一出一）

逻辑乘的逻辑符号是



与

运算规则是

$0\cdot 0=0$　　$A\cdot 0=0$

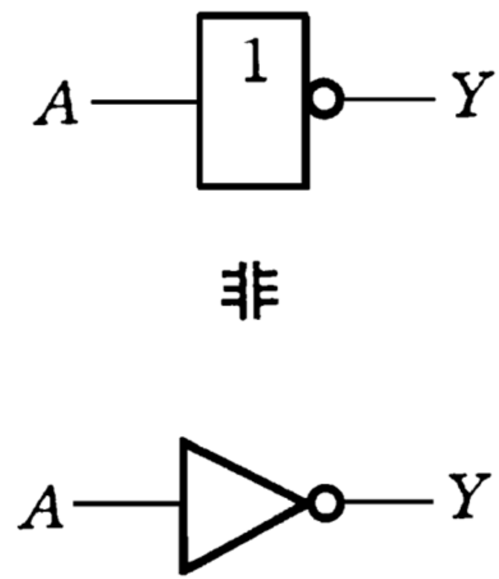
$0\cdot 1=0$　　$A\cdot 1=A$

$1\cdot 0=0$　　$A\cdot A=A$

$1\cdot 1=1$

逻辑非，又称为“非运算” $Y=\overline{A}$ 其意义在于：函数值 Y 等于输入变量的反。

逻辑非的逻辑符号是



运算规则是

$$\overline{0}=1 \qquad \overline{\overline{A}}=A$$

$$\overline{1}=0 \qquad A+\overline{A}=1$$

$$A\cdot\overline{A}=0$$

复合逻辑运算

与非逻辑运算 $Y=\overline{A\cdot B}$

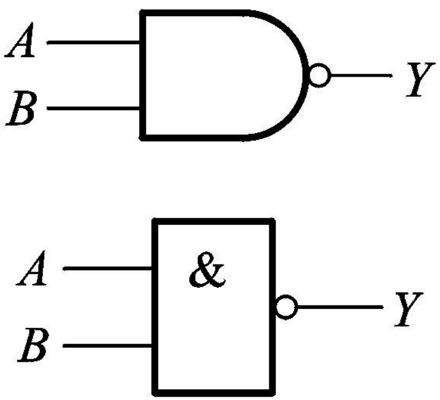
或非逻辑运算 $Y=\overline{A+B}$

与或非逻辑运算 $Y=\overline{AB+CD}$

异或逻辑运算 $Y=\overline{A}B+A\overline{B}=A\oplus B$

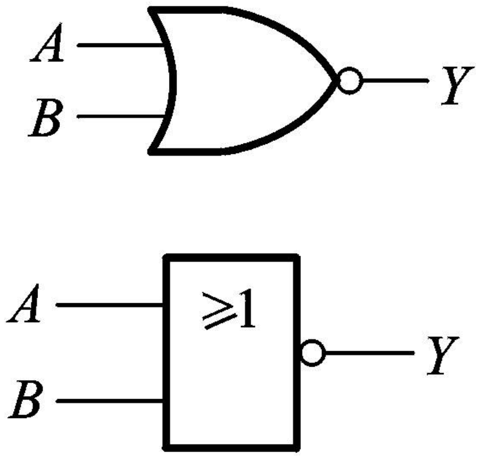
同或逻辑运算 $Y=\overline{A}\,\overline{B}+AB=A\odot B$

复合逻辑运算



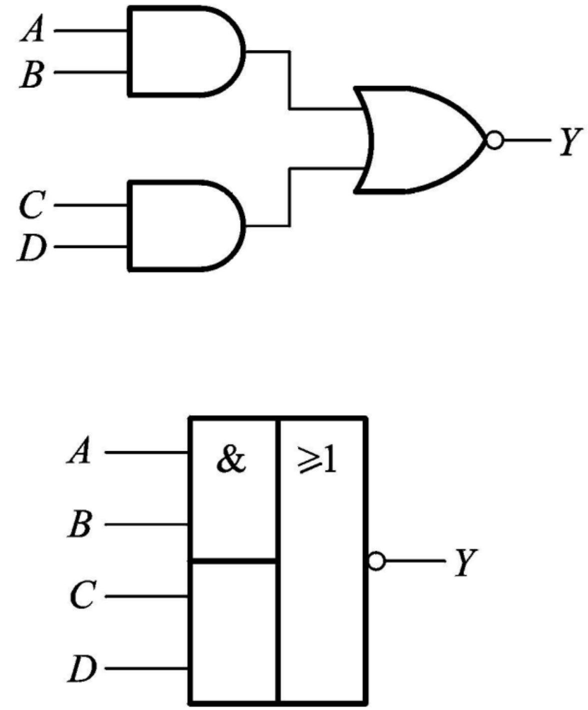
与非

$Y=(A \cdot B)'$



或非

$Y=(A+B)'$



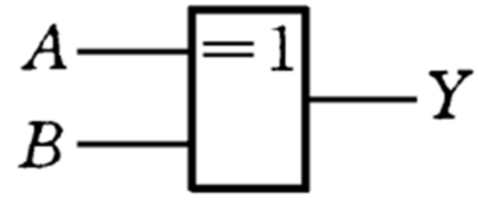
与或非

$Y=(A \cdot B+C \cdot D)'$

异或运算规则

- $Y = A \oplus B$
- 不同出一

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



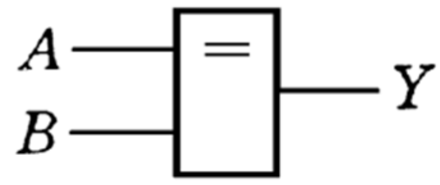
异或

$$Y = A \oplus B$$

同或运算规则

- $Y = A \odot B$
- 相同出一

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



同或

$Y = A \odot B$

异或与同或逻辑的关系

$$A \oplus B = \overline{A \odot B}$$

$$A \oplus B = \overline{A} \oplus \overline{B}$$

$$A \odot B = \overline{A \oplus B}$$

$$A \odot B = \overline{A} \odot \overline{B}$$

$$A \odot B = \overline{A} \oplus B = A \oplus \overline{B}$$

$$A \oplus B = \overline{A} \odot B = A \odot \overline{B}$$

2.1.3 逻辑函数和真值表

逻辑函数的定义：按某种逻辑关系，用有限个与、或、非逻辑运算关系将逻辑变量x_0, x_1, … x_n结合起来，得到的表达式：

$Y=f$（ x_0, x_1, … x_n ）称为逻辑函数

当逻辑变量x_0, x_1, … x_n 的取值唯一确定后, Y的取值也就唯一确定。

逻辑变量和逻辑函数的取值只有0和1，逻辑运算只有三种基本逻辑运算或其复合。一般可通过先列出其真值表，然后再建立该逻辑函数。

一个逻辑函数可以有多种表示方法。

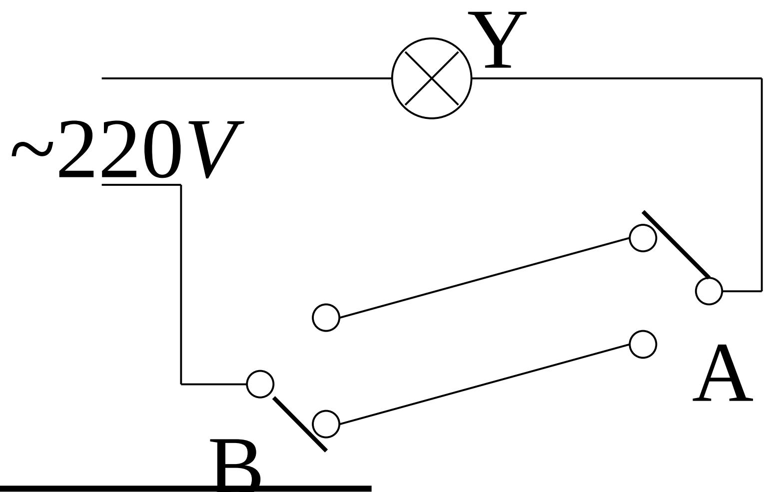
2.1.3 逻辑函数和真值表

真值表是逻辑函数的一种表示方法。真值表是一种表格；其左边(输入栏)列举了所有输入变量的全部取值组合，右边(输出栏)列出了每一种输入取值组合下对应的函数值。因此，它唯一、正确、完整地描述了一个确定的逻辑函数。通过对这个逻辑问题的详尽的分析和正确的描述可以方便地列出其真值表。

真值表

输入变量 $A\ \ B\ \ C\cdots\cdots$	输出 $Y_1\ \ Y_2\ \cdots\cdots$
遍历所有可能的输入变量的取值组合	输出对应的取值

例1：楼道灯控制电路的真值表



开关A	开关B	灯
向上	向上	亮
向上	向下	灭
向下	向上	灭
向下	向下	亮

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

例2：提案三人表决电路的真值表

甲	乙	丙	提案
否	否	否	否决
否	否	是	否决
否	是	否	否决
否	是	是	通过
是	否	否	否决
是	否	是	通过
是	是	否	通过
是	是	是	通过

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

表达式是逻辑函数的又一种表示方法。根据真值表可以方便地写出逻辑函数表达式。其方法是：

选择真值表中所有函数值Y＝1所对应的输入变量取值组合，用逻辑乘表示它们（逻辑乘是这样确定的：当输入变量的取值为1时，用原变量表示，当输入变量的取值为0时，用反变量表示)，再将全部逻辑乘相加便可得到逻辑函数的与－或表达式，也称为“积之和”表达式。

逻辑函数表达式还有另一种形式，获得的方法是：

选择真值表中所有函数值Y＝0所对应的输入变量取值组合，用逻辑加表示它们（逻辑加是这样来确定的：当输入变量的取值为0时，用原变量表示，当输入变量的取值为1时，用反变量表示），再将全部逻辑加相乘便可得到逻辑函数的或－与表达式，也称为“和之积”表达式

例1. 楼道灯控制电路逻辑函数的建立

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

$0\ 0\ 1 \rightarrow \overline{A}\,\overline{B}$

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

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

$1\ 1\ 1 \rightarrow AB$

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

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

例2.提案表决电路逻辑函数的建立

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

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

$A+B+C$

$A+B+\bar{C}$

$A+\bar{B}+C$

$\bar{A}BC$

$\bar{A}+B+C$

$A\bar{B}C$

$AB\bar{C}$

ABC

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

2.1.4 逻辑函数相等

具有相同输入变量的两个逻辑函数Y和G，如果在任意输入取值下其函数值都相等, 则逻辑函数Y和G相等。换句话说，若需要证明两个逻辑函数相等，只需看它们是否具有完全相同的真值表即可。

两个不同的逻辑电路，对应有不同形式的逻辑函数表达式，但是，只要这两个逻辑函数相等，则这两个逻辑电路具有完全相同的逻辑功能。

逻辑代数中的公式

01律

$$A + 0 = A$$
$$A + 1 = 1$$
$$A + \overline{A} = 1$$
$$A \oplus 0 = A$$
$$A \oplus 1 = \overline{A}$$
$$A \oplus \overline{A} = 1$$

01律

$$A \cdot 1 = A$$
$$A \cdot 0 = 0$$
$$A \cdot \overline{A} = 0$$
$$A \odot 1 = A$$
$$A \odot 0 = \overline{A}$$
$$A \odot \overline{A} = 0$$

逻辑代数中的公式

交换律

$$A + B = B + A$$

$$A \oplus B = B \oplus A$$

结合律

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

$$A \oplus B \oplus C = (A \oplus B) \oplus C$$

分配率

$$A(B + C) = AB + AC$$

$$A(B \oplus C) = AB \oplus AC$$

交换律

普通代数不适用

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

$$A \odot B = B \odot A$$

结合律

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

$$A \odot B \odot C = (A \odot B) \odot C$$

分配率

$$\boxed{A + BC = (A + B)(A + C)}$$

$$A + B \odot C = (A + B) \odot (A + C)$$

分配律的证明

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

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

逻辑代数中的公式

重叠率	重叠率
$A+A=A$	$A\cdot A=A$
$A\oplus A=0$	$A\odot A=1$
反演率	反演率
$\overline{A+B}=\overline{A}\cdot\overline{B}$	$\overline{A\cdot B}=\overline{A}+\overline{B}$
$\overline{A\oplus B}=A\odot B$	$\overline{A\odot B}=A\oplus B$

逻辑代数中的公式

调换率

若$A \oplus B = C$

则$A \oplus C = B$

$B \oplus C = A$

并推导出

$A + B = A \oplus B \oplus AB$

$A + B = A \odot B \odot AB$

调换率

若$A \odot B = C$

则$A \odot C = B$

$B \odot C = A$

并推导出

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

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

2.1.5 三个规则

代入规则　用逻辑函数Y代替任意含有变量A的等式中的全部变量A，则等式仍然成立

例如　$A(B+E) = AB+AE$

用$(C+D)$ 代替E　$A[B+(C+D)]=AB+A(C+D)$

则　原等式左边$=A[B+(C+D)]=AB+AC+AD$

原等式右边$=AB+A(C+D)=AB+AC+AD$

等式 $A[B+(C+D)]=AB+A(C+D)$ 仍然成立

反演规则 (德·摩根定理)将一个逻辑函数表达式Y中所有的“ ·”换成“＋”，所有的“＋”换成“ ·”，所有的常量0 换成常量1，所有的常量1 换成常量0，所有的原变量 换成反变量，所有的反变量换成 原变量，这样得到的一个新的函数式 $\overline{F}$,称为原函数Y的反(补)函数；这个变换过程可归纳为：变号、变常量、变变量，保留原有的运算和书写顺序不变，凡是不属于单个变量的非运算符号保留不变。利用反演规则可以方便地求逻辑函数的反函数。

反演规则 例

$$F_1 = AB C + \overline{B}C\overline{D} + C(\overline{B} + D)$$

$$\overline{F}_1 = (\overline{A} + \overline{B} + \overline{C})(B + \overline{C} + D)(\overline{C} + B\overline{D})$$

注意这些省略了的“ · ”

注意添加括号

$$F_2 = \overline{\overline{AB} + C} + (\overline{B}\,\overline{D} + C)\overline{A\overline{B} + \overline{D}}$$

$$\overline{F}_2 = \overline{\overline{\overline{A} + \overline{B}} \cdot \overline{C}} \cdot [(B + D) \cdot \overline{C}] + \overline{(\overline{A} + B) \cdot D}$$

对偶规则

将一个逻辑函数表达式Y中所有的“·”换成“+”，所有的“+”换成“·”，所有的常量0换成常量1，所有的常量1换成常量0，这样得到的一个新的函数式Y^*,称为原函数Y的对偶函数；这个变换过程可归纳为：变号、变常量、保留原有的运算书写顺序不变

对偶规则 例

$$F_1 = ABC + \bar{B}C\bar{D} + C(\bar{B} + D)$$

$$F_1^* = (A + B + C)(\bar{B} + C + \bar{D})(C + \bar{B}D)$$

注意这些省略了的“·”

$$F_2 = \overline{\overline{AB} + C} + (\bar{B}\,\bar{D} + C)\overline{A\bar{B} + \bar{D}}$$

$$F_2^* = \overline{\overline{A + B} \cdot C} \cdot [(\bar{B} + \bar{D}) \cdot C] + \overline{(A + \bar{B}) \cdot \bar{D}}$$

对偶规则

从上例中可以看到，对偶规则将一个逻辑函数的与－或(或－与)表达式Y，变成了或－与(与－或)表达式Y^*(需注意大多数逻辑函数Y和Y^*并不相等，若Y和Y^*相等，则称其为自对偶函数)。

可以证明，若 Y＝G，则$Y^*=G^*$

进一步可得 $(Y^*)^*=(G^*)^*$

由此，前述的逻辑代数的公式中左、右两组之间实际上呈互为对偶关系，当记忆这些公式时，仅需记忆一半即可，这为逻辑运算提供了许多方便。

2.1.6 常用公式

1. 吸收律

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

1. 吸收律

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

2. $A+AB=A$

2. $A(A+B)=A$

3. $A+\overline{A}B=A+B$

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

4. 添加项定理 $AB+\overline{A}C+BC=AB+\overline{A}C$

4. 添加项定理

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

添加项定理推广

$$AB+\overline{A}C+BCDE\cdots\cdots=AB+\overline{A}C$$

添加项定理的证明

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

等式左边 $=AB+\overline{A}C+(A+\overline{A})BC$

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

$=AB+\overline{A}C$

等式右边 $=AB+\overline{A}C$

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

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

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

证毕

5. 交叉互换律 $AB+\bar{A}C=(A+C)(\bar{A}+B)$

5. 交叉互换律 $(A+B)(\bar{A}+C)=AC+\bar{A}B$

6. $\overline{A\bar{B}+\bar{A}B}=AB+\bar{A}\,\bar{B}$

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

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

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

- $A + AB = A$
- $A + A'B = A + B$
- $AB + AB' = A$
- $A(A + B) = A$
- $AB + A'C + BC = AB + A'C$

  $AB + A'C + BCD = AB + A'C$
- $A(AB)' = AB'$; $A'(AB)' = A'$

2.1.7 逻辑函数的标准形式

最小项

逻辑函数的多个变量相乘构成的代数项，称为乘积项(积项)

最小项是乘积项，该乘积项包含了逻辑函数的全部变量，而且每个变量因子仅仅以原变量或反变量的形式在一个乘积项中唯一出现一次

n 变量逻辑函数共有2^n个最小项

三变量最小项

A	B	C	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$	$\bar{A}B\bar{C}$	$\bar{A}BC$	$A\bar{B}\bar{C}$	$A\bar{B}C$	$AB\bar{C}$	ABC
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

最小项的性质

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

四变量最小项

A B C D	m_i	对应最小项	A B C D	m_i	对应最小项
0 0 0 0	m_0	$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$	1 0 0 0	m_8	$A\,\overline{B}\,\overline{C}\,\overline{D}$
0 0 0 1	m_1	$\overline{A}\,\overline{B}\,\overline{C}\,D$	1 0 0 1	m_9	$A\,\overline{B}\,\overline{C}\,D$
0 0 1 0	m_2	$\overline{A}\,\overline{B}\,C\,\overline{D}$	1 0 1 0	m_{10}	$A\,\overline{B}\,C\,\overline{D}$
0 0 1 1	m_3	$\overline{A}\,\overline{B}\,C\,D$	1 0 1 1	m_{11}	$A\,\overline{B}\,C\,D$
0 1 0 0	m_4	$\overline{A}\,B\,\overline{C}\,\overline{D}$	1 1 0 0	m_{12}	$A\,B\,\overline{C}\,\overline{D}$
0 1 0 1	m_5	$\overline{A}\,B\,\overline{C}\,D$	1 1 0 1	m_{13}	$A\,B\,\overline{C}\,D$
0 1 1 0	m_6	$\overline{A}\,B\,C\,\overline{D}$	1 1 1 0	m_{14}	$A\,B\,C\,\overline{D}$
0 1 1 1	m_7	$\overline{A}\,B\,C\,D$	1 1 1 1	m_{15}	$A\,B\,C\,D$

最小项表达式

全部由最小项相加构成的“与一或”表达式，即最小项表达式。又称标准与一或表达式，标准积之和表达式。最小项表达式

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

可以写成 $Y(A,B,C) = m_3 + m_5 + m_6 + m_7$

进一步写成 $Y(A,B,C) = \sum m(3,5,6,7)$

一般形式是

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

最小项和最小项表达式

对于任何一个逻辑函数，都可以写出它的标准与一或表达式。

对于逻辑函数的任何非标准与一或表达式，可用吸收律公式

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

将乘积项中所缺少的变量补齐，从而变换成标准与一或表达式

最大项和最大项表达式

逻辑函数的多个变量相加（或）构成的代数项，称为相加项(和项)

最大项是相加项，该相加项包含了逻辑函数的全部变量，而且每个变量因子仅仅以原变量或反变量的形式在一个相加项中唯一出现一次

n 变量逻辑函数共有2^n个最大项

最大项的性质

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

四变量最大项

A B C D	M_i	对应最大项	A B C D	M_i	对应最大项
0 0 0 0	M_0	$A+B+C+D$	1 0 0 0	M_8	$\overline{A}+B+C+D$
0 0 0 1	M_1	$A+B+C+\overline{D}$	1 0 0 1	M_9	$\overline{A}+B+C+\overline{D}$
0 0 1 0	M_2	$A+B+\overline{C}+D$	1 0 1 0	M_{10}	$\overline{A}+B+\overline{C}+D$
0 0 1 1	M_3	$A+\overline{B}+C+D$	1 0 1 1	M_{11}	$\overline{A}+B+\overline{C}+\overline{D}$
0 1 0 0	M_4	$\overline{A}+\overline{B}+C+D$	1 1 0 0	M_{12}	$\overline{A}+\overline{B}+C+D$
0 1 0 1	M_5	$A+\overline{B}+C+\overline{D}$	1 1 0 1	M_{13}	$\overline{A}+\overline{B}+C+\overline{D}$
0 1 1 0	M_6	$A+\overline{B}+\overline{C}+D$	1 1 1 0	M_{14}	$\overline{A}+\overline{B}+\overline{C}+D$
0 1 1 1	M_7	$A+\overline{B}+\overline{C}+\overline{D}$	1 1 1 1	M_{15}	$\overline{A}+\overline{B}+\overline{C}+\overline{D}$

最大项表达式

全部由最大项相乘构成的“或－与”表达式，即最大项表达式。又称标准或－与表达式，标准和之积表达式。最大项表达式

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

可以写成 $Y(A,B,C)=M_0\cdot M_1\cdot M_2\cdot M_4$

进一步写成 $Y(A,B,C)=\Pi_M(0,1,2,4)$

一般形式是

$$Y=\prod_{i=0}^{2^n-1}(a_i+M_i)$$

最小项和最大项的关系

同一逻辑函数的下标i相同的最小项和最大项互补，即 $M_i = \overline{m_i}$ $\overline{M_i} = m_i$

最小项表达式和最大项表达式的关系

具有完全相同下标编号i，变量数相同的最小项表达式和最大项表达式互补

$$Y(a,b,c) = \Sigma m(0,1,2,4) = \overline{G(a,b,c)} = \overline{\Pi_M(0,1,2,4)}$$

则有 $Y(a,b,c) = \Sigma m(0,1,2,4) = \Pi_M(3,5,6,7)$

逻辑函数的异或（同或）表达式

$$
\begin{aligned}
Y &= \sum_{i=0}^{2^n-1} a_i \cdot m_i \\
&= a_0 m_0 + \sum_{i=1}^{2^n-1} a_i \cdot m_i \\
&= a_0 m_0 \oplus \sum_{i=1}^{2^n-1} a_i \cdot m_i \oplus (a_0 m_0 \cdot \sum_{i=1}^{2^n-1} a_i \cdot m_i) \\
&= a_0 m_0 \oplus \sum_{i=1}^{2^n-1} a_i \cdot m_i \oplus 0 \\
&= a_0 m_0 \oplus (a_1 m_1 + \sum_{i=2}^{2^n-1} a_i \cdot m_i) \\
&= a_0 m_0 \oplus a_1 m_1 \oplus \cdots \oplus a_{2^n-1} m_{2^n-1}
\end{aligned}
$$

逻辑函数的异或（同或）表达式

同理，可以得到同或表达式

$$Y=(a_0+M_0)\odot(a_1+M_1)\odot\cdots\cdots\odot(a_{2^n-1}+M_{2^n-1})$$

可以将一个逻辑函数的任意表达式变换成标准与一和表达式后，再变换成异或表达式。

这里约定主要采用最小项表达式来描述逻辑函数

2.2 逻辑函数的简化

逻辑函数简化的意义在于：简化逻辑电路，减少元、器件数量，降低设备成本，提高设备可靠性

简化的目标是获得“最简”与一或表达式

“最简”的涵义是：与一或表达式中乘积项的个数最少，每个乘积项中包含的变量数最少

2.2.1 公式化简法（代数法）

化简工具: 逻辑代数的全部定律(理)、规则、公式，

化简方法

利用公式 $AB+A\overline{B}=A$ 的并项法

利用公式 $A+AB=A$ 的吸收法

利用公式 $A+\overline{A}B=A+B$ 的消去法

利用公式 $AB+\overline{A}C+BC=AB+\overline{A}C$ 的配项法

公式化简法的特点

公式化简法带有明显的试探性。试探的成功率和化简过程的简繁，取决于对公式定理的理解和熟悉程度。实际的化简，往往并不是可以简单套用某个公式来解决的，需要在认真观察、分析之后灵活地运用公式，最后求得最简表达式。许多情况下，化简方法不是唯一的，甚至结果也不是唯一的，不要强求套用某个固定的模式，要充分利用最熟悉的公式，从一点突破，最终解决问题。

公式化简法例

$$
\begin{aligned}
F_1 &= ABC + A\bar{B}\,\bar{C} + AB\bar{C} + A\bar{B}C \\
&= A(BC + \bar{B}\,\bar{C}) + A(B\bar{C} + \bar{B}C) \\
&= A\overline{(\cancel{B\bar{C} + \bar{B}C})} + A(\cancel{B\bar{C} + \bar{B}C}) \\
&= A
\end{aligned}
$$

$$
\begin{aligned}
F_2 &= AC + A\cancel{\bar{B}C}D + \cancel{ABC} + \bar{C}D + ABD \\
&= AC + \bar{C}D + \cancel{ABD} \\
&= AC + \bar{C}D
\end{aligned}
$$

公式化简法例

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

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

公式化简法例

$$F_4 = (A+B+C)(A+B+\overline{C})\cdot$$
$$\cdot(A+\overline{B}+C)(A+\overline{B}+\overline{C})(\overline{A}+B+C)$$
$$= (A+B)(A+\overline{B})(B+C)$$
$$= AB+AC$$

$$F_4^* = ABC+AB\overline{C}+A\overline{B}C+A\overline{B}\,\overline{C}+\overline{A}BC$$
$$= AB+A\overline{B}+BC$$
$$= A+BC$$

$$F_4 = (F_4^*)^* = A(B+C) = AB+AC$$

公式化简法例

$$F_5=\bar A\,\bar C+\bar A\,\overset{\textcircled{2}}{\cancel{\overline{\overline{B+D}+\bar A}}}+\bar A\,\bar B+BC+$$

$$+\underset{\textcircled{1}}{\cancel{\bar A\,\bar C\,\bar D}}+\overline{\underset{\textcircled{3}}{\cancel{\overline{AB\overline{B+D}}}}\;\overline{\bar C D}+BC}$$

$$=\bar A(\overset{\textcircled{4}}{\cancel{\bar B+\bar C}})+BC+\bar C D\underset{\textcircled{4}}{\cancel{\overline{BC}}}$$

$$=\bar A+BC+\bar C D$$

$$F_6=(A+B)(A+\bar B)(\bar A+B)(\bar A D+C)+$$

$$+\overline{\bar A+\cancel{\bar B+C}}\,(\cancel{\bar B CD}+CD)$$

$$=ABC$$

公式化简法例

$$F_7 = A(\cancel{A+B})(\overline{A}+C)(\underline{B+D})(\overline{A}+\cancel{C+E+F})\cdot$$

$$\cdot(\underline{\overline{B}+F})(\cancel{D+E+F})$$

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

$$= AC(BF+\overline{B}D)$$

$$= ABCF+A\overline{B}CD$$

$$F_8 = \overline{A}BD+BC+AC+C\overline{D}$$

$$= \underline{\overline{A}BD}+\cancel{BCD}+\cancel{BC\overline{D}}+\underline{AC}+\underline{C\overline{D}}$$

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

2.2.2 图解法（卡诺图法）

卡诺图

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

2.2.2 图解法（卡诺图法）

A	B	C	D	m_i
0	**0**	**0**	**0**	$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$
0	**0**	**0**	**1**	$\overline{A}\,\overline{B}\,\overline{C}\,D$
0	**0**	**1**	**0**	$\overline{A}\,\overline{B}\,C\,\overline{D}$
0	**0**	**1**	**1**	$\overline{A}\,\overline{B}\,C\,D$
0	**1**	**0**	**0**	$\overline{A}\,B\,\overline{C}\,\overline{D}$
0	**1**	**0**	**1**	$\overline{A}\,B\,\overline{C}\,D$
0	**1**	**1**	**0**	$\overline{A}\,B\,C\,\overline{D}$
0	**1**	**1**	**1**	$\overline{A}\,B\,C\,D$
1	**0**	**0**	**0**	$A\,\overline{B}\,\overline{C}\,\overline{D}$
1	**0**	**0**	**1**	$A\,\overline{B}\,\overline{C}\,D$
1	**0**	**1**	**0**	$A\,\overline{B}\,C\,\overline{D}$
1	**0**	**1**	**1**	$A\,\overline{B}\,C\,D$
1	**1**	**0**	**0**	$A\,B\,\overline{C}\,\overline{D}$
1	**1**	**0**	**1**	$A\,B\,\overline{C}\,D$
1	**1**	**1**	**0**	$A\,B\,C\,\overline{D}$
1	**1**	**1**	**1**	$A\,B\,C\,D$

真值表变形产生的卡诺图

<table>
<tr><td colspan="2" rowspan="2">A
B
C D</td><td colspan="2">0</td><td colspan="2">1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td rowspan="2">0</td><td>0</td><td>m_0</td><td>m_4</td><td>m_{12}</td><td>m_8</td></tr>
<tr><td>1</td><td>m_1</td><td>m_5</td><td>m_{13}</td><td>m_9</td></tr>
<tr><td rowspan="2">1</td><td>1</td><td>m_3</td><td>m_7</td><td>m_{15}</td><td>m_{11}</td></tr>
<tr><td>0</td><td>m_2</td><td>m_6</td><td>m_{14}</td><td>m_{10}</td></tr>
</table>

卡诺图的一般形式

2变量卡诺图

B \ A	0	1
0	m_0	m_2
1	m_1	m_3

4变量卡诺图

CD \ AB	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}

3变量卡诺图

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

卡诺图一般形式的简化

2变量卡诺图

B \ A	0	1
0	0	2
1	1	3

4变量卡诺图

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

3变量卡诺图

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

用卡诺图表示逻辑函数

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

B \ A	0	1
0	1^{0}	0^{2}
1	0^{1}	1^{3}

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

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

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

C \ AB	00	01	11	10
0	0^{0}	0^{2}	1^{6}	0^{4}
1	0^{1}	1^{3}	1^{7}	1^{5}

逻辑相邻的概念

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

任意一个n变量最小项有n个相邻的最小项

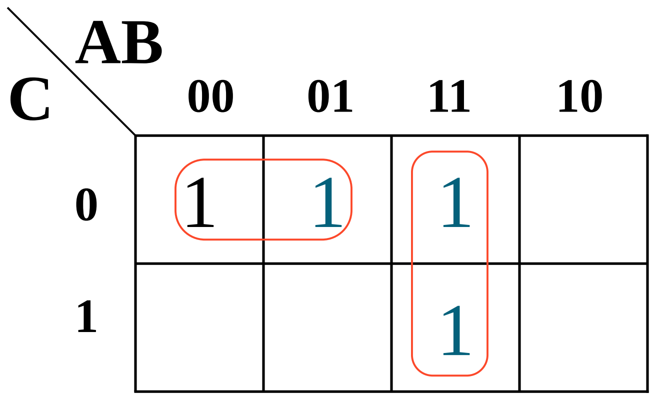
如果两个乘积项中只有一个变量因子不相同，则称这两个乘积项逻辑相邻。具有逻辑相邻性的两个乘积项可以消去这个不相同的变量因子合并为一个变量因子更少的乘积项

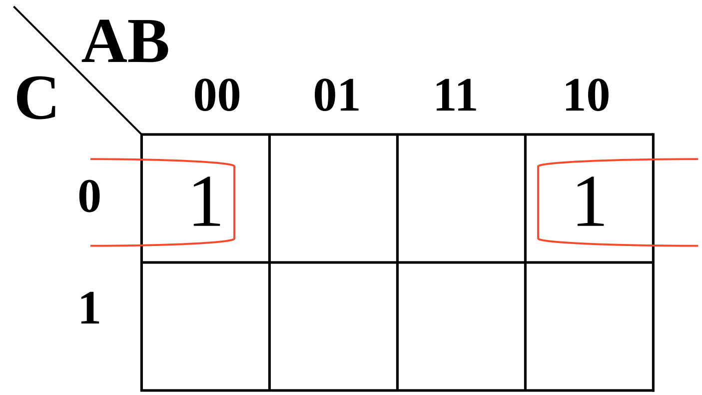
用卡诺图合并最小项

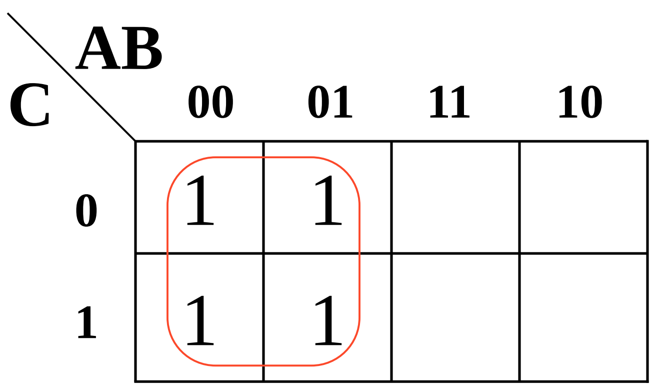
可以直观地凭借最小项在卡诺图中的几何位置来确定最小项的逻辑相邻性。再简单地将几何位置相邻的两个最小项方格（简称“1”格）用一个“圈”圈起来，便可产生一个合并项

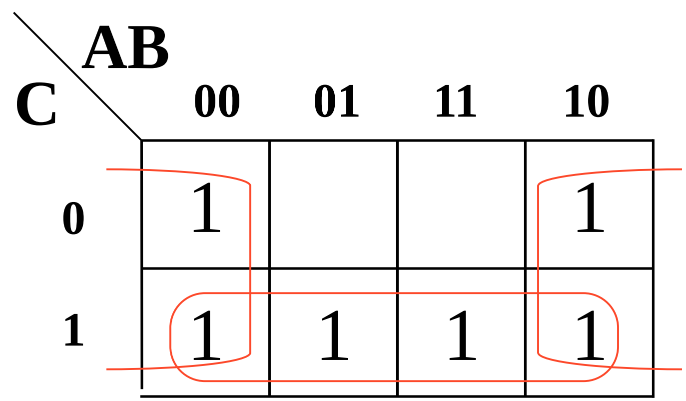
根据逻辑相邻的概念，可以进一步将“圈”扩大，把4个、8个 ……2^i 个“1”格圈起来得到更为简单的合并项

用卡诺图合并最小项


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


AB
C
00 01 11 10
0 1 1
1


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


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

用卡诺图合并最小项

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

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

用卡诺图合并最小项

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

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

用卡诺图合并最小项

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

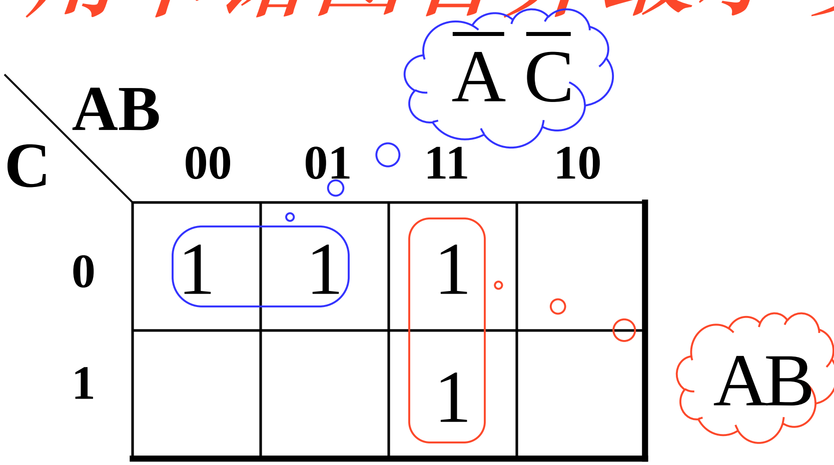
CD \ AB	00	01	11	10
00	1	1	1	1
01			1	1
11			1	1
10	1	1	1	1

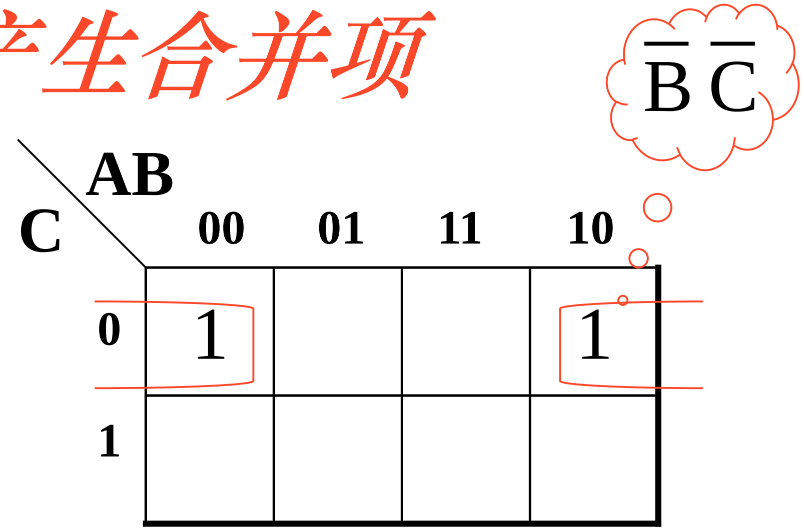
用卡诺图合并最小项产生合并项

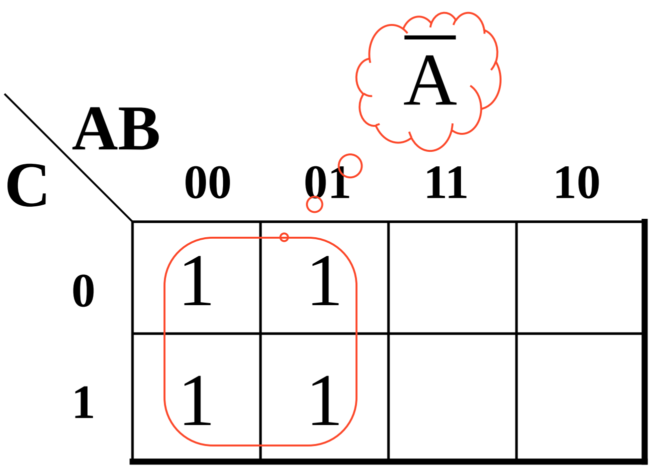
合并项由圈所覆盖的范围内没有发生变化的变量组成。

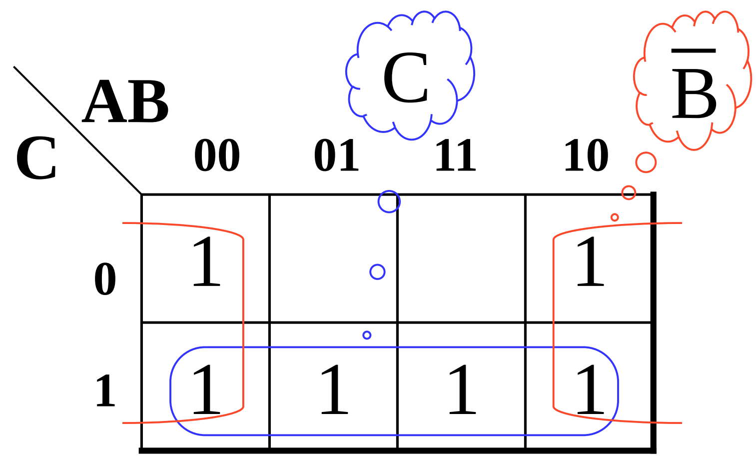
一个圈内应当、也只能圈入 2^i 个方格，则这个合并项与最小项相比消去了 i 个变量

用卡诺图合并最小项产生合并项

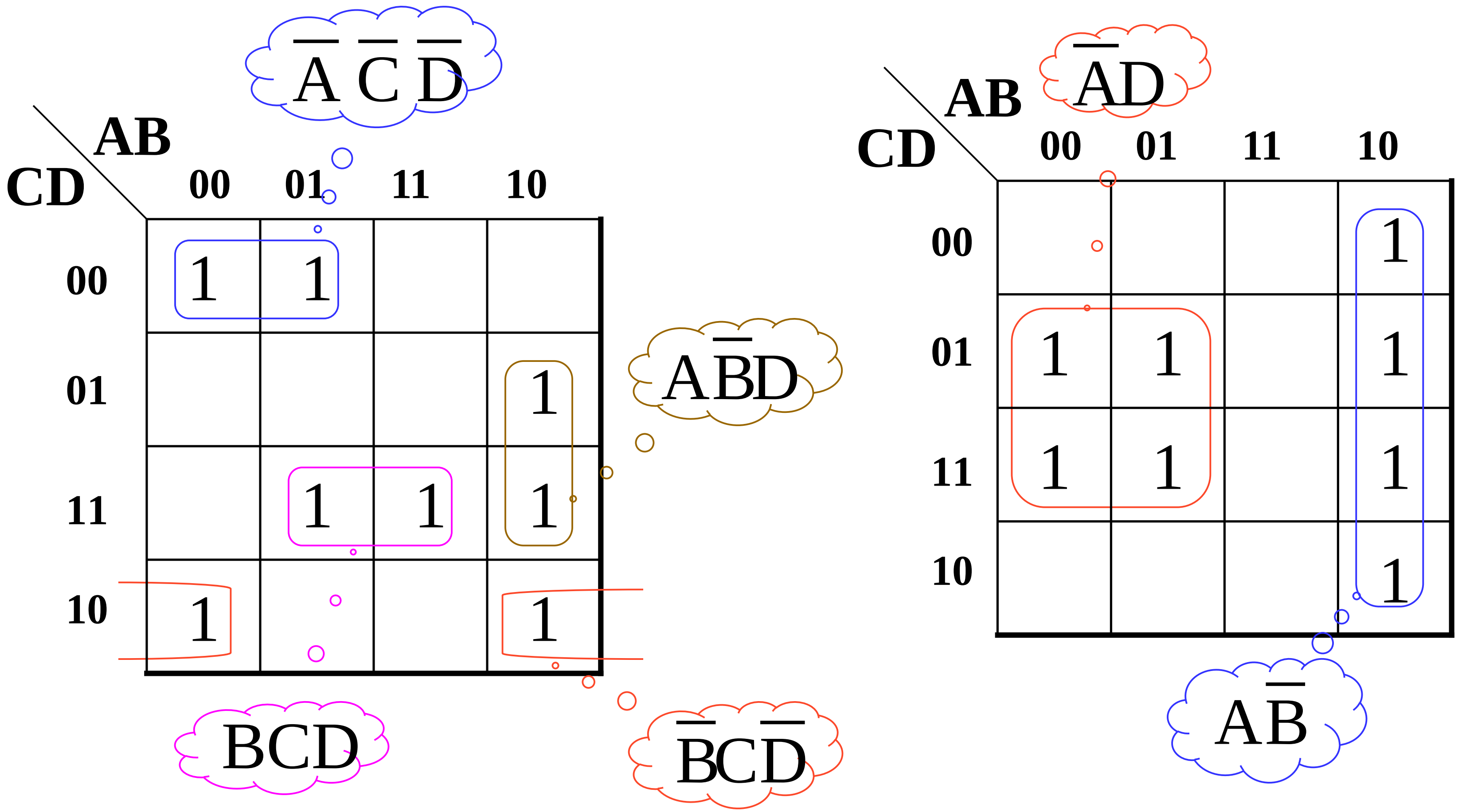




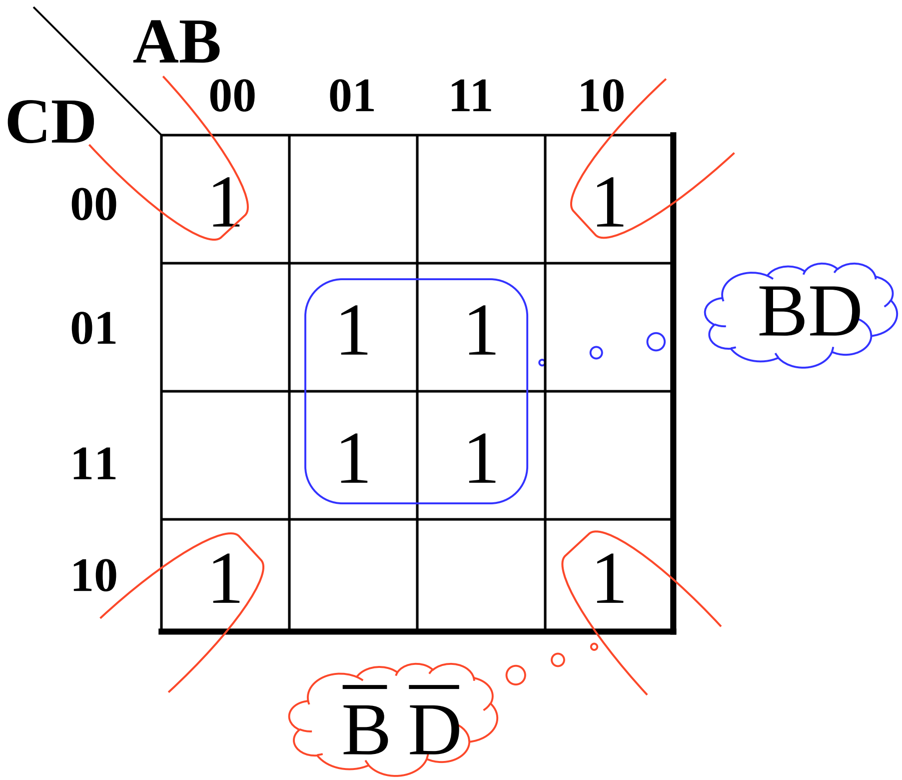


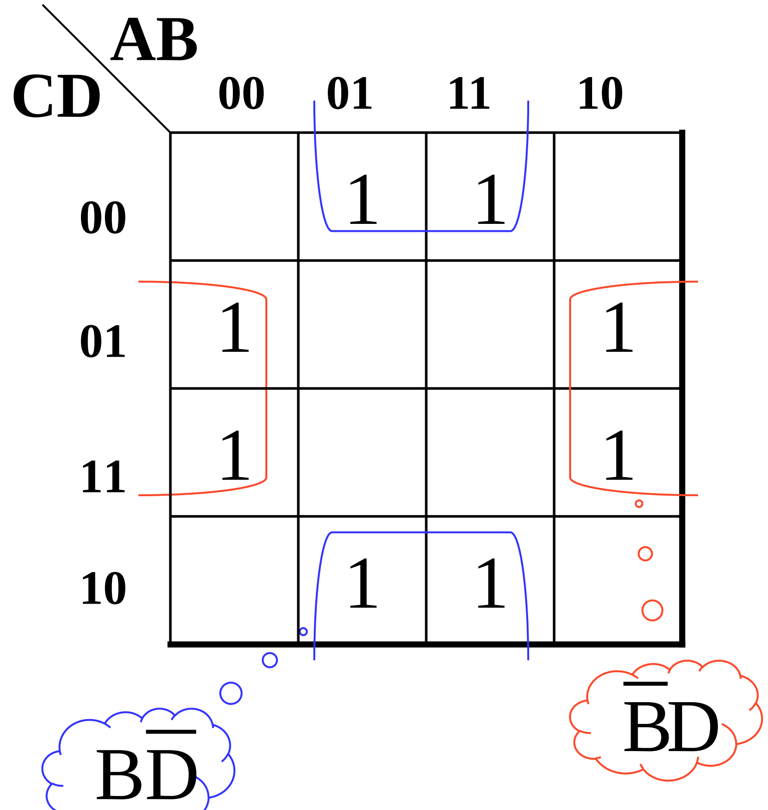


用卡诺图合并最小项产生合并项

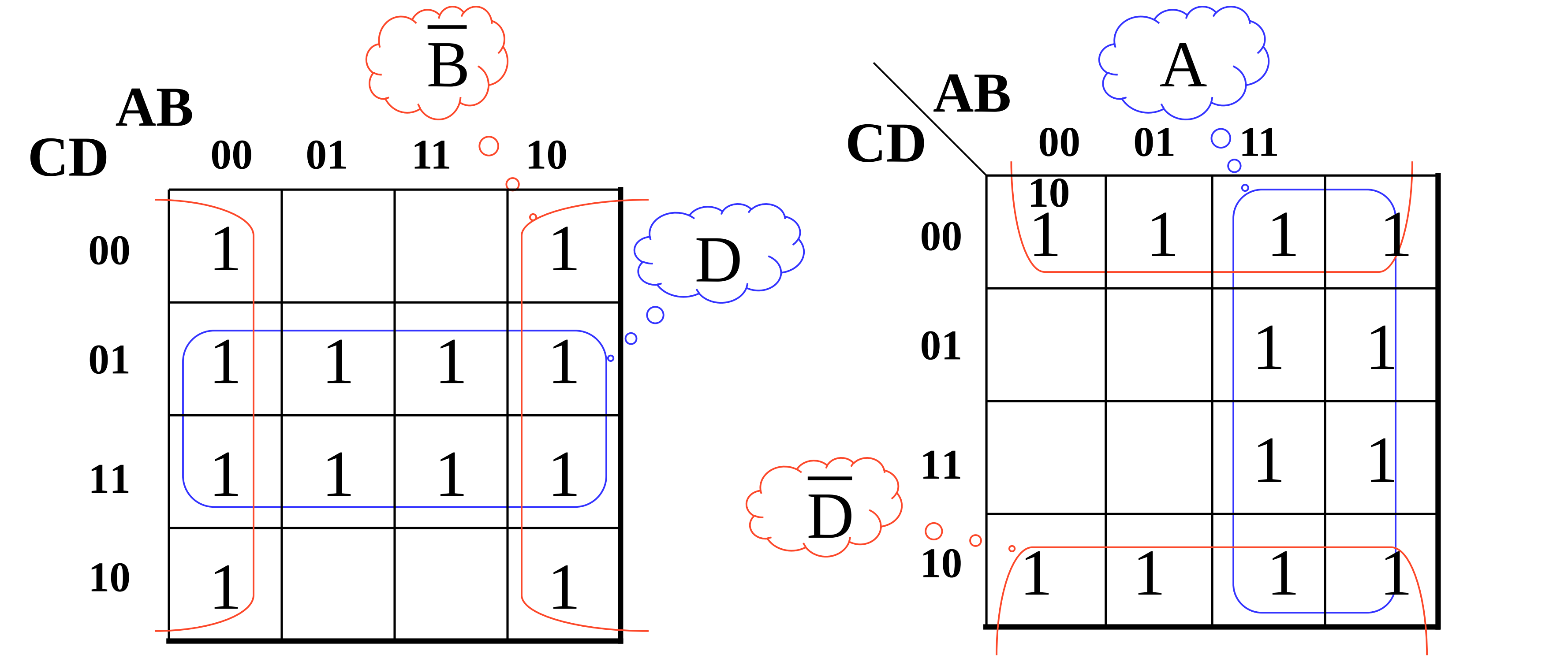


用卡诺图合并最小项产生合并项





用卡诺图合并最小项产生合并项



当一个圈覆盖了卡诺图的全部1格时产生的合并项为1

用卡诺图合并最小项产生的合并项

主要项(素项、本原蕴含项)：当合并圈不能再扩大所产生的合并项

必要项(实质素项、实质本原蕴含项)：凡主要项圈中至少有一个“特定”的1格没有被其它主要项所覆盖

多余项：一个主要项圈中如果不包含有“特定”的1格，即该圈内全部1格都被其它圈所覆盖，这个主要项就是多余项

用卡诺图化简逻辑函数

用卡诺图化简逻辑函数就是寻找全部必要项的过程，具体地说就是：

1. 作图：用卡诺图表示一个确定的逻辑函数

2. 圈1：圈出全部1格，合并产生全部必要项

3. 将全部必要项相加即可得到最简与—或表达式

关于圈圈的几点说明

1. 从没有相邻项的孤立1格圈起，先圈只有一种圈法的主要项，再由少到多逐渐扩大圈的覆盖面，圈要尽可能大，但必须保证圈内1格数是 2^i 个，任意1格允许被重复多次圈入；

2. 注意不要漏掉边缘和四角的相邻格；

3. 同一卡诺图可以有多种圈法，应选择一种合适的圈法，直到所有1格无遗漏地至少被圈一次，且圈的总数尽可能少；

4. 也可以圈“0格”化简，得到最简或一与表达式；但是，决不允许在同一卡诺图内既圈“0格”同时又圈“1格”

*卡诺图化简逻辑函数*例1

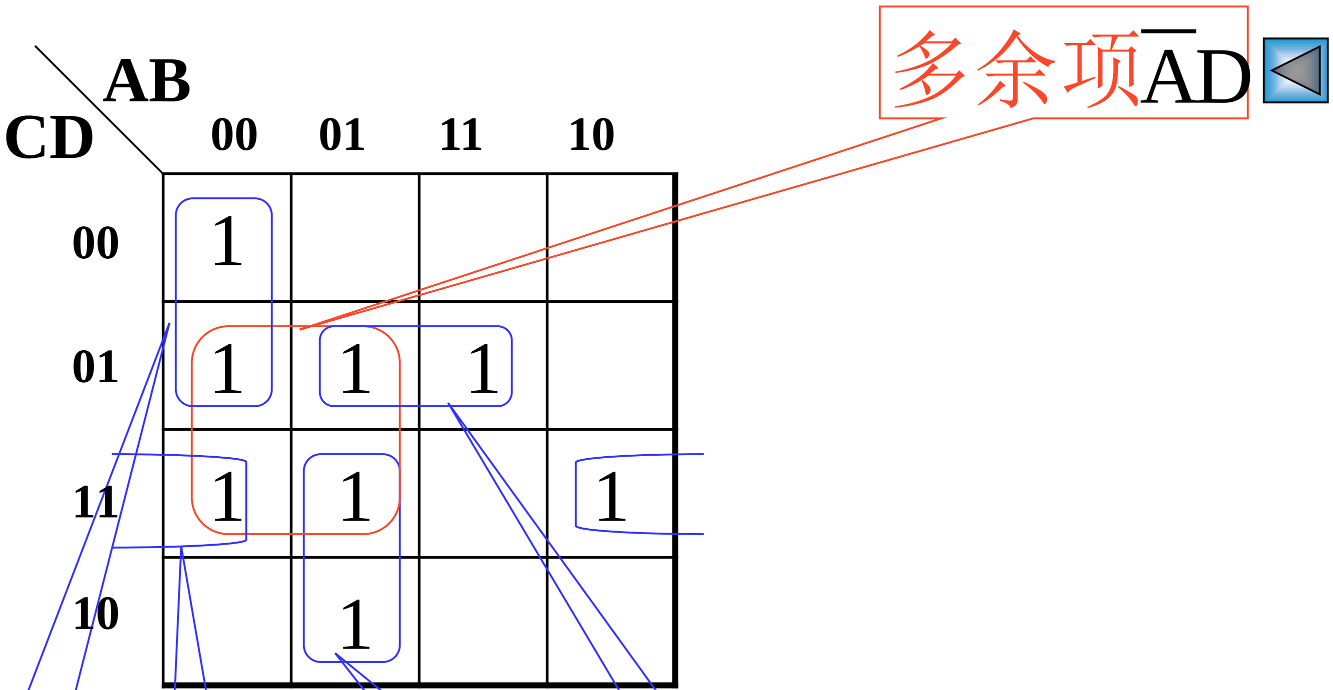
$$F_1(A,B,C,D)=\Sigma m(0,4,5,8,10,12,15)$$

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

$$F_1=\boxed{ABCD}+\boxed{\overline{A}B\overline{C}}+\boxed{A\overline{B}\,\overline{D}}+\boxed{\overline{C}\,\overline{D}}$$

卡诺图化简逻辑函数例2

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



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

*卡诺图化简逻辑函数*例3

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

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

$$F_3=\overline{A}\,\overline{C}\,\overline{D}+B\overline{C}D+ACD+\overline{B}C\overline{D}$$

$$F_3=\overline{A}\,\overline{B}\,\overline{D}+\overline{A}B\overline{C}+ABD+A\overline{B}C$$

*卡诺图化简逻辑函数*例4

$$F_4(A,B,C) = A\overline{B} + B\overline{C} + \overline{B}C + \overline{A}B$$
$$= \Sigma m(1,2,3,4,5,6)$$

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

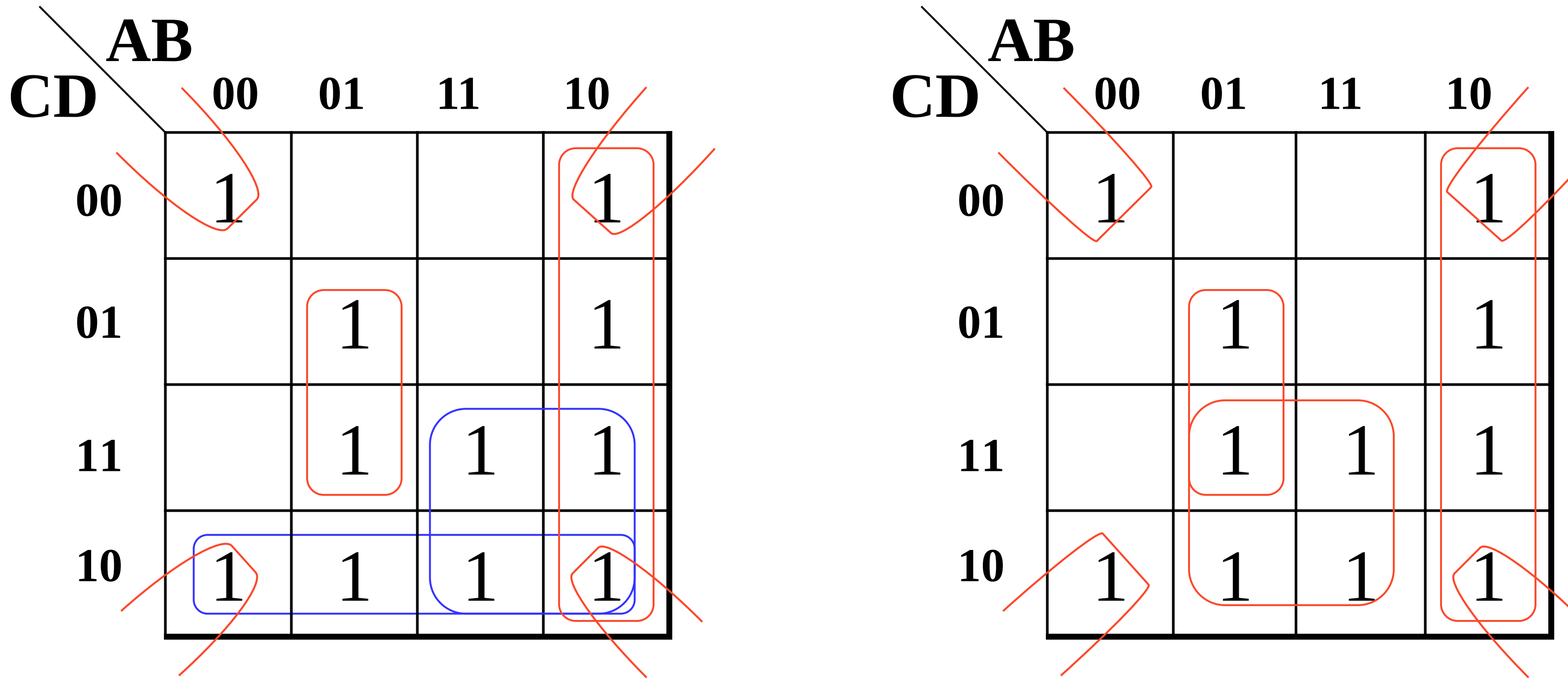
$$F_4 = \overline{A}C + B\overline{C} + A\overline{B}$$

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

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

*卡诺图化简逻辑函数*例5

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



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

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

*卡诺图化简逻辑函数*例6

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

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

圈1化简

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

圈0化简

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

*卡诺图化简逻辑函数*例7(非标准形式的化简）

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

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

$\overline{B}CD$ $\quad m_3+m_{11}$

$B\overline{C}$ $\quad m_4+m_5+m_{12}+m_{13}$

$\overline{A}\,\overline{C}$ $\quad m_0+m_1+m_4+m_5$

$A\overline{B}C$ $\quad m_{10}+m_{11}$

$C\overline{D}$ $\quad m_2+m_6+m_{14}+m_{10}$

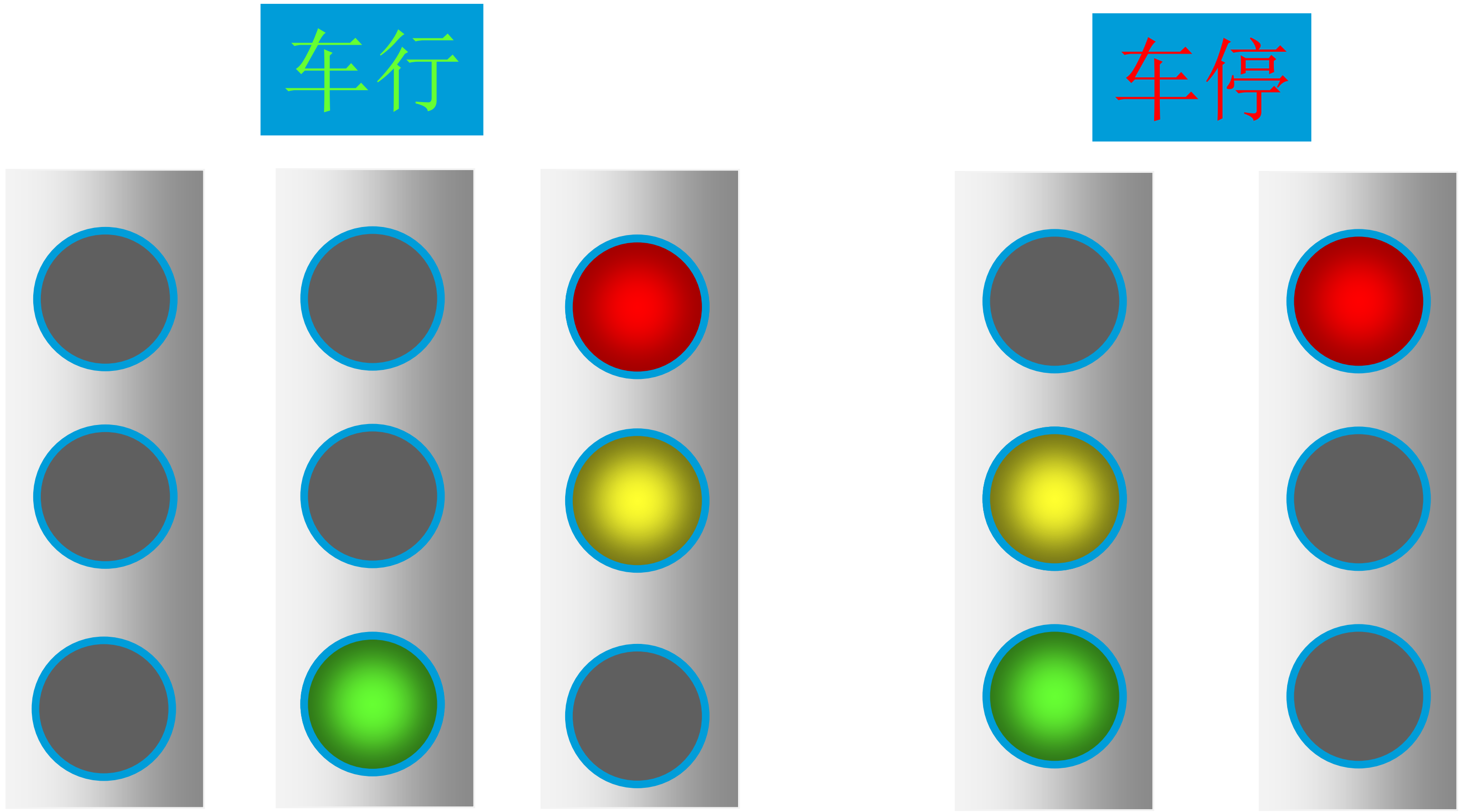
$$F_7=\overline{B}C+B\overline{C}+\overline{A}\,\overline{C}+C\overline{D}$$

任意项的使用

当逻辑函数变量的某些取值组合不会出现时（变量之间存在某种约束关系），或者函数在某些变量取值下的输出不确定，可能为0，也可能为1，这样的变量取值组合所对应的最小项称为任意项（约束项、随意项）。具有任意项的逻辑函数称为非完全描述逻辑函数。对于非完全描述逻辑函数，合理地利用任意项可以将逻辑函数进一步化简。

*非完全描述逻辑函数*例

交通指挥系统中，红（R）、黄（Y）、绿（G）灯状态示意图



非完全描述逻辑函数中任意项的表示

非完全描述逻辑函数的任意项在表达式中用 Σd (或 Πd) 表示，在卡诺图中用“×”表示。例如，交通指挥系统中，红、黄、绿灯与车行、车停的关系就是一个非完全描述逻辑函数

G \ RY	00	01	11	10
0	0	×	0	1
1	0	1	×	×

$$P(R,G,Y)=\Sigma m(3,4)+\Sigma d(2,5,7)$$

$$P(R,Y,G)=\Pi_M(0,1,6)\cdot\Pi_d(2,5,7)$$

任意项在卡诺图化简逻辑函数中的应用

G \ RY	00	01	11	10
0	0	×	0	1
1	0	1	×	×

$$P = R\overline{Y}\,\overline{G} + \overline{R}YG$$

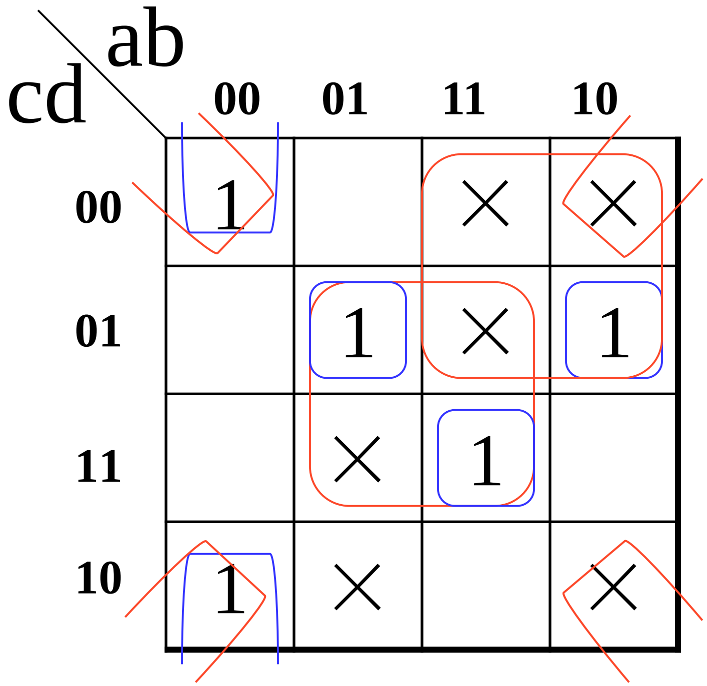
G \ RY	00	01	11	10
0	0	×	0	1
1	0	1	×	×

$$P = R\overline{Y} + YG$$

本例化简结果可见，不利用任意项时和利用任意项时，分别得到的两个不同结果是有明显差别的

任意项在卡诺图化简逻辑函数中的应用

$$F(a,b,c,d)=\Sigma m(0,2,5,9,15)+\Sigma d(6,7,8,10,12,13)$$



未利用任意项

$$F=\bar{a}\,\bar{b}\,\bar{d}+\bar{a}b\bar{c}d+abcd+a\bar{b}\bar{c}d$$

利用任意项

$$F=\bar{b}\,\bar{d}+bd+a\bar{c}$$

根据化简时需要，任意项可以作为1格、也可以作为0格参与化简，应以有利于获得最简表达式为前提

任意项在卡诺图化简逻辑函数中的应用

当输入8421BCD为奇数时电路的输出为1，否则为0，写出该电路的最简与或表达式

cd \ ab	00	01	11	10
00	0	0	×	0
01	1	1	×	1
11	1	1	×	×
10	0	0	×	×

未利用任意项

$$Y = \bar{a}d + \bar{b}\bar{c}d$$

利用任意项

$$Y = d$$

END