

# 第二章 逻辑代数基础

## 内容提要

本章介绍分析数字逻辑功能的数学方法。首先介绍逻辑代数的基本运算、常用公式和基本定理，然后介绍逻辑代数及其表示方法、逻辑函数的化简。重点掌握卡诺图化简逻辑函数，为后续课程打下基础。

## 本章的内容

2.1 概述

2.2 逻辑代数中的三种基本运算

2.3 逻辑代数的基本公式和常用公式

2.4 逻辑代数的基本定理

2.5 逻辑函数及其表示方法

2.6 逻辑函数的化简方法

2.7 具有无关项的逻辑函数及其化简

## 2.1 概述

### 2.1.1 二值逻辑和逻辑运算

在数字电路中，1位二进制数码“0”和“1”不仅可以表示数量的大小，也可以表示事物的两种不同的逻辑状态，如电平的高低、开关的闭合和断开、电机的起动和停止、电灯的亮和灭等。**这种只有两种对立逻辑状态的逻辑关系，称为二值逻辑。**

当二进制数码“0”和“1”表示二值逻辑，并按某种因果关系进行运算时，称为**逻辑运算**，最基本的三种逻辑运算为“与”、“或”、“非”，它与算术运算的本质区别是“0”和“1”没有数量的意义。故在逻辑运算中  $1+1=1$ （或运算）

## 2.1.2 数字电路的特点及描述工具

数字电路是一种开关电路，输入、输出量是高、低电平，可以用二值变量（取值只能为0，1）来表示。输入量和输出量之间的关系是一种逻辑上的因果关系。仿效普通函数的概念，数字电路可以用逻辑函数的数学工具来描述。

逻辑代数是布尔代数在数字电路中二值逻辑的应用，它首先是由英国数学家乔治·布尔（**George Boole**）提出的，用在逻辑运算上。后来用在数字电路中，就被称为开关代数或逻辑代数，它是逻辑函数的基础。

## 注意：

1. 逻辑代数和普通数学代数的运算相似，如有交换律、结合律、分配律，而且逻辑代数中也用字母表示变量，叫逻辑变量。
2. 逻辑代数和普通数学代数有本质区别，普通数学代数中的变量取值可以是正数、负数、有理数和无理数，是进行十进制（0~9）数值运算。而逻辑代数中变量的取值只有两个：“0”和“1”。并且“0”和“1”没有数值意义，它只是表示事物的两种逻辑状态。

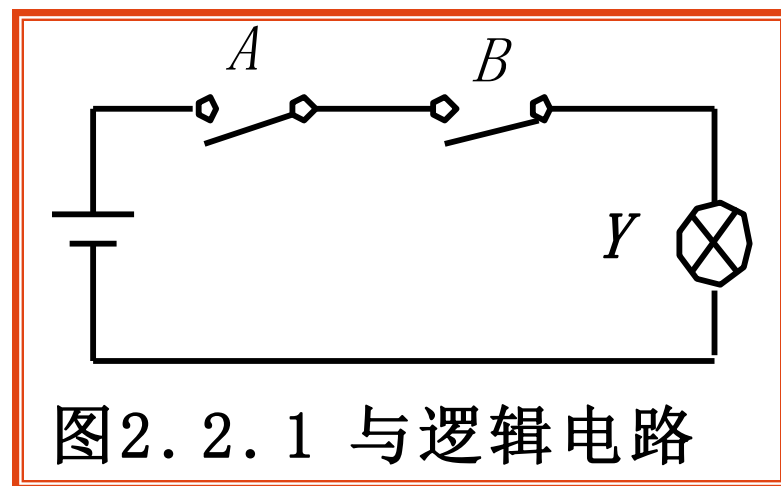
## 2.2 逻辑代数中的三种基本运算

在二值逻辑函数中，最基本的逻辑运算有与（AND）、或（OR）、非（NOT）三种逻辑运算。

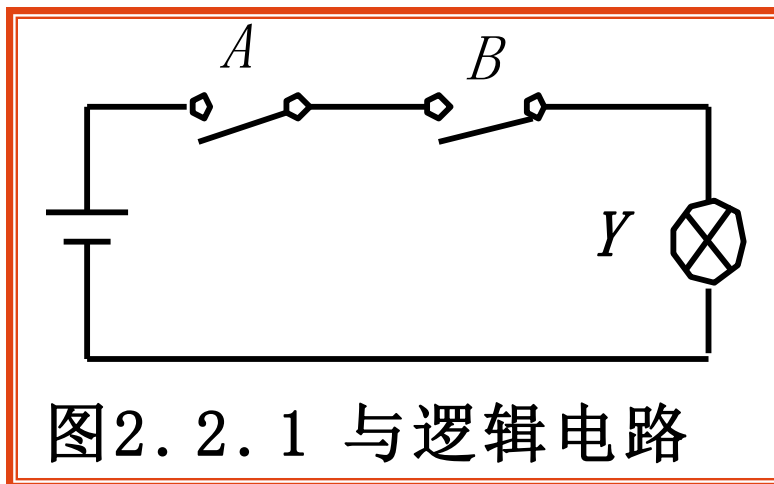
### 2.2.1 与运算

与运算也叫逻辑乘或逻辑与，即当所有的条件都满足时，事件才会发生，即“缺一不可”。

如图2.2.1所示电路，两个串联的开关控制一盏灯就是与逻辑事例，只有开关A、B同时闭合时灯才会亮。



设开关闭合用“1”表示，  
断开用“0”表示；灯亮用“1”  
表示，灯灭用“0”表示（逻辑  
赋值），则可得到表2.2.1所  
示的输入输出的逻辑关系，  
称为真值表



从表中可知，其逻辑规律服  
从“有0出0，全1才出1”

这种与逻辑可以写成下面的表  
达式：

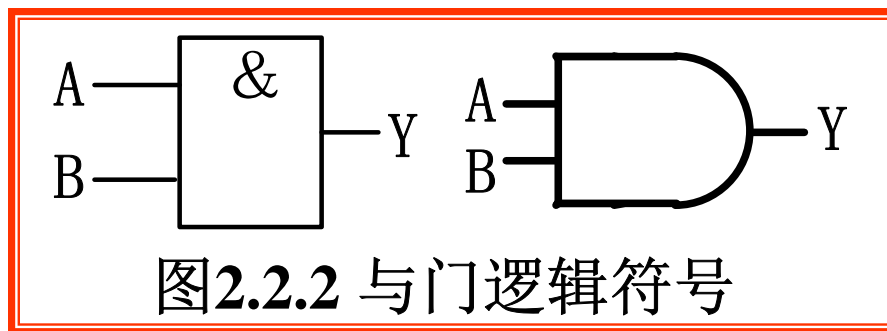
$$Y = A \bullet B$$

称为与逻辑式，这种运算称为与  
运算

表2.2.1 与逻辑真值表

输入		输出
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

也可以用图2.2.2表示与逻辑，称为逻辑门或逻辑符号，实现与逻辑运算的门电路称为与门。



若有 $n$ 个逻辑变量做与运算，其逻辑式可表示为

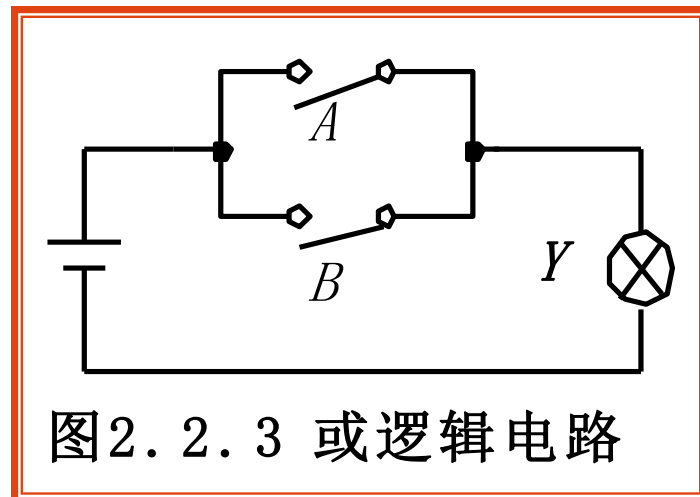
$$Y = A_1 A_2 \Lambda A_n$$

## 2.2.2 或运算

或运算也叫逻辑加或逻辑或，即当其中一个条件满足时，事件就会发生，即“有一即可”



如图2.2.3所示电路，两个并联的开关控制一盏灯就是或逻辑事例，只要开关A、B有一个闭合时灯就会亮。



用与前面相同的逻辑赋值同样也可得到其真值表如表2.2.2所示，其逻辑规律服从“有1出1，全0才出0”

其逻辑式为

$$Y = A + B$$

上式说明：当逻辑变量A、B有一个为1时，逻辑函数输出Y就为1。只有A、B全为0，Y才为0。

表2.2.2 或逻辑真值表

输入		输出
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

其逻辑门符号如图  
2.2.4所示，实现或逻辑  
运算的门电路称为或门。

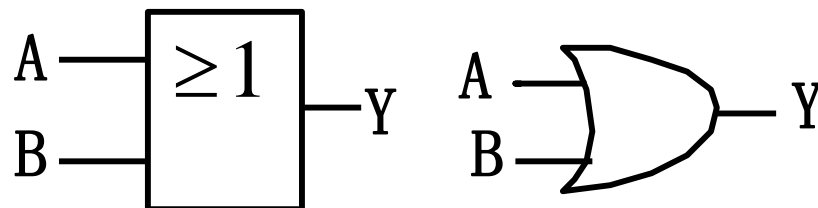


图2.2.4 或门逻辑符号

若有 $n$ 个逻辑变量做或运算，其逻辑式可表示为

$$Y = A_1 + A_2 + \Lambda + A_n$$

### 3. 非逻辑运算

条件具备时，事件不发生；条件不具备时，事件发生，这种因果关系叫做逻辑非，也称逻辑求反

如图2.2.5所示电路，一个开关控制一盏灯就是非逻辑事例，当开关A闭合时灯就会不亮。

用与前面相同的逻辑赋值同样也可得到其真值表如表2.2.3所示

非逻辑运算也叫逻辑非或非运算、反相运算，即输出变量是输入变量的相反状态。其逻辑式为

$$Y = A'$$

注：上式也可写成

$$Y = \bar{A} \text{ 或 } Y = \sim A \text{ 等}$$

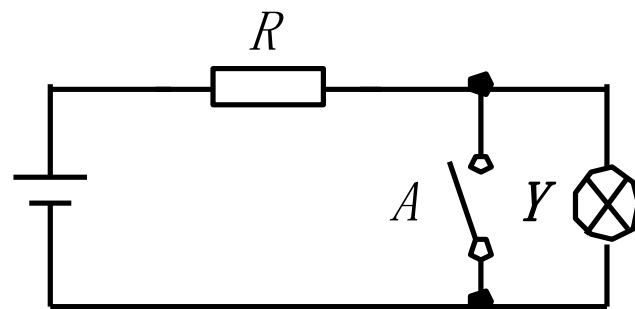


图2.2.5 非逻辑电路

表2.2.3 非逻辑真值表

A	Y
0	1
1	0

其逻辑门符号如图2.2.6所示，实现非逻辑运算的门电路称为非门

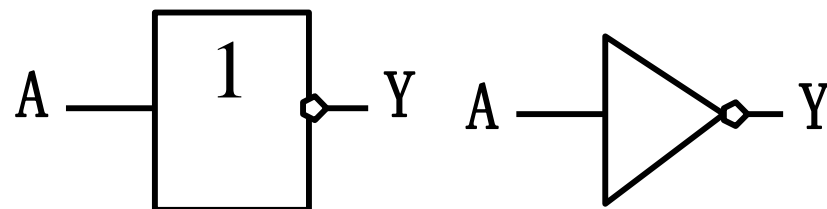


图2.2.6 非门逻辑符号

以上为最基本的三种逻辑运算，除此之外，还有下面的由基本逻辑运算组合出来的逻辑运算

#### 4. 与非（NAND）逻辑运算

与非运算是先与运算后非运算的组合。以二变量为例，布尔代数表达式为：

$$Y = (AB)'$$

其真值表如表2.2.4所示

表2.2.4 与非逻辑真值表

输入		输出
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

其逻辑规律服从“有0出1，全1才出0”

表2.2.4 与非逻辑真值表

输入		输出
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

实现与非运算用与非门电路来实现，如图2.2.7所示

## 5. 或非（NOR）运算

或非运算是先或运算后非运算的组合。以二变量A、B为例，布尔代数表达式为：

$$Y = (A + B)'$$

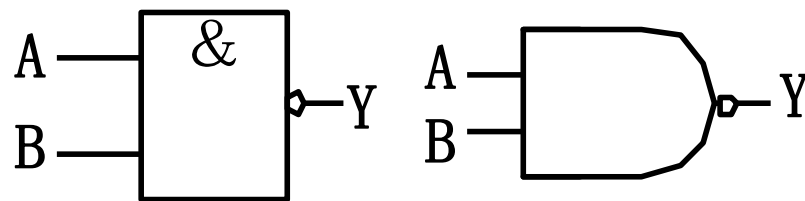


图2.2.7 与非门逻辑符号

其真值表如表2.2.5所示

或非逻辑规律服从**有“1”出“0”**  
**全“0”出“1”**

或非运算用或非门电路来实现，  
如图2.2.8所示

表2.2.5 或非逻辑真值表

输入		输出
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

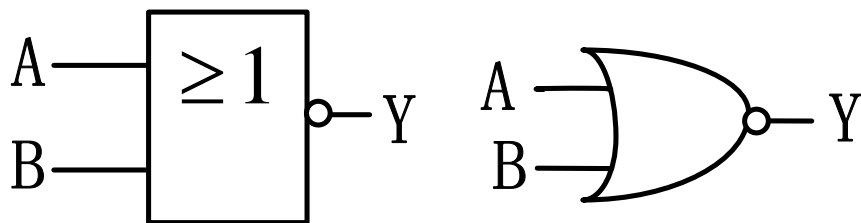


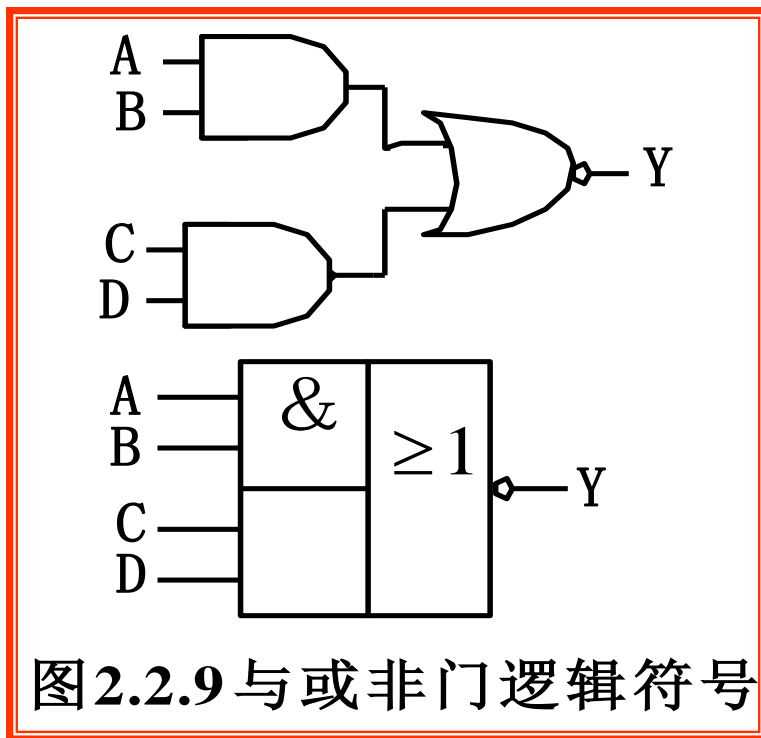
图2.2.8 或非门逻辑符号

## 6.与或非运算

与或非运算是“先与后或非再非”三种运算的组合。  
以四变量为例，逻辑表达式为：

$$Y = (AB + CD)'$$

上式说明：当输入变量A、B同时为1或C、D同时为1时，输出Y才等于0。与或非运算是先或运算后非运算的组合。在工程应用中，与或非运算由与或非门电路来实现，其真值表见书P22表2.2.6所示，逻辑符号如图2.2.9所示



## 7. 异或运算

其布尔表达式（逻辑函数式）为

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

符号“ $\oplus$ ”表示异或运算，即两个输入逻辑变量取值不同时 $Y=1$ ，即不同为“1”相同为“0”，异或运算用异或门电路来实现

其真值表如表2.2.6所示

其门电路的逻辑符号如图2.2.10所示



图2.2.10 异或门逻辑符号

表2.2.6 异或逻辑真值表

输入		输出
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



# ☺异或运算的性质

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： $A(B \oplus C) = AB \oplus AC$

4. 
$$\begin{aligned} A \oplus \bar{A} &= 1 & A \oplus A &= 0 \\ A \oplus 1 &= \bar{A} & A \oplus 0 &= A \end{aligned}$$

推论：当  $n$  个变量做异或运算时，若有偶数个变量取“1”时，则函数为“0”；若奇数个变量取1时，则函数为1.

## 8. 同或运算：

其布尔表达式为 
$$Y = A \odot B = (A \oplus B)' = AB + A'B'$$

符号“ $\odot$ ”表示同或运算，即两个输入变量值相同时  $Y=1$ ，即相同为“1”不同为“0”。同或运算用同或门电路来实现，它等价于异或门输出加非门，

其真值表如表2.2.7所示

其门电路的逻辑符号如图2.2.11所示

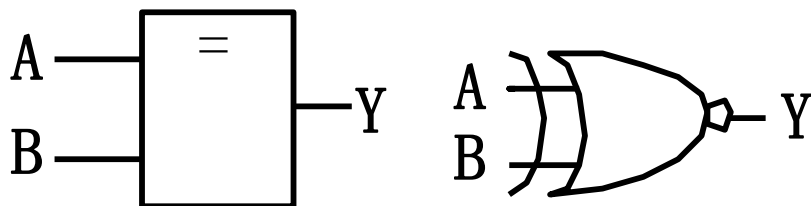


图2.2.11 同或门逻辑符号

表2.2.7 同或逻辑真值表

输入		输出
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

## 2.3 逻辑代数的基本公式和常用公式

### 2.3.1 基本公式 (pp24)

表2.3.1为逻辑代数的基本公式，也叫布尔恒等式

表2.3.1 逻辑代数的基本公式

序号	公 式	序号	公 式
1	$0 \cdot A = 0$	10	$1' = 0 \quad 0' = 1$
2	$1 \cdot A = A$	11	$1 + A = 1$
3	$A \cdot A = A$	12	$0 + A = A$
4	$A \cdot A' = 0$	13	$A + A = A$
5	$A \cdot B = B \cdot A$	14	$A + A' = 1$
6	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	15	$A + B = B + A$
7	$A \cdot (B + C) = A \cdot B + A \cdot C$	16	$A + (B + C) = (A + B) + C$
⑧	$(A \cdot B)' = A' + B'$	⑪	$A + B \cdot C = (A + B) \cdot (A + C)$
9	$(A')' = A$	⑫	$(A + B)' = A' \cdot B'$

# 说明：由表中可以看出

## 1.关于变量与常数关系的定理

$$A \cdot 0 = 0$$

$$A + 0 = A$$

$$A \cdot 1 = A$$

$$A + 1 = 1$$

## 2. 交换律、结合律、分配律

a. 交换律： $AB = BA$        $A + B = B + A$

b. 结合律： $A(BC) = (AB)C$

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

c. 分配律： $A(B + C) = AB + AC$



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

p25

### 3.逻辑函数独有的基本定理

a. 互补律： $A \bullet A' = 0 \quad A + A' = 1$

b. 重叠律： $A \cdot A = A \quad A + A = A$

c. 非非律： $(A')' = A$

d. 吸收律： $A + A B = A \quad A (A+B) = A$

$$A + A' B = A + B$$

e. 摩根定律： $(AB)' = A' + B' \quad (A + B)' = A' \cdot B'$

注：以上定律均可由真值表验证

## 2.3.2 若干常用公式

表2.3.2为常用的一些公式

表2.3.2 常用公式

序号	公 式
21	$A + A \cdot B = A$
22	$A + A' B = A + B$
23	$A \cdot B + A \cdot B' = A$
24	$A(A + B) = A$
25	$A \cdot B + A' \cdot C + BC = A \cdot B + A' \cdot C$ $A \cdot B + A' \cdot C + BCD = A \cdot B + A' \cdot C$
26	$A \cdot (A \cdot B)' = A \cdot B' \quad A' \cdot (A \cdot B)' = A'$

## 说明：

1.  $A + AB = A$ ：在两个乘积项相加时，如果其中一项包含另一项，则这一项是多余的，可以删掉；
2.  $A + A'B = A + B$ ：在两个乘积项相加时，如果其中一项含有另一项的取反因子，则此取反因子多余的，可从该项中删除；
3.  $AB + A B' = A$ ：在两个乘积项相加时，如果它们其中的一个因子相同，而另一个因子取反，则两项合并，保留相同因子；
4.  $A (A + B) = A$ ：在当一项和包含这一项的和项相乘时，其和项可以消掉

**5.  $AB + A' C + BC = AB + A' C$  :** 在三个乘积项相加时, 如果前两项中的一个因子互为反, 那么剩余的因子组成的另一项则是多余的, 可以删掉; 公式  $AB + A' C + BCD = AB + A' C$  的原理和上述相同

**6.  $A (A B)' = A B'$  :** 如果某项和包含这一项的乘积项取反相乘时, 则这一项可以删掉;

**7.  $A' (A B)' = A'$  :** 当某个项取反和包含这一项的乘积项取反相乘时, 则只保留这个取反项

以上的公式比较常用, 应该能熟用, 为以后逻辑函数的化简打好基础



## 2.4 逻辑代数的基本定理

### 2.4.1 代入定理

**内容**：任何一个含有变量  $A$  的等式，如果将所有出现  $A$  的位置都用同一个逻辑函数  $G$  来替换，则等式仍然成立。

利用代入定理可以证明一些公式，也可以将前面的两变量常用公式推广成多变量的公式

## 例2.4.1 试用代入规则证明摩根定律适用多变量的情况

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

证明：设  $G = BC$                   代入公式左右的  $B$  中

可得

$$\begin{aligned} \text{左} &= (AG)' = (ABC)' = A' + G' = A' + (BC)' \\ &= A' + B' + C' \end{aligned}$$

$$\text{右} = A' + G' = A' + (BC)' = A' + B' + C'$$

故：

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

同理设  $G = B + C$  代入式子左右的  $B$

可得

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

## 2. 反演定理

**内容：**若已知逻辑函数 $Y$ 的逻辑式，则只要将 $Y$ 式中所有的“.”换为“+”，“+”换为“.”，常量“0”换成“1”，“1”换成“0”，所有原变量（不带非号）变成反变量，所有反变量换成原变量，得到的新函数即为原函数 $Y$ 的反函数（补函数） $Y'$ 。利用摩根定律，可以求一个逻辑函数的反函数。

**注意：**1. 变换中必须保持先与后或的顺序；  
2. 对跨越两个或两个以上变量的“非号”要保留不变；

例2.4.2 已知 $Y = A ( B + C ) + C D$  , 求 $Y'$

解：由反演定理

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

或直接求反，由摩根定律

$$\begin{aligned} Y' &= [A(B + C) + CD]' = [A(B + C)]' \cdot (CD)' \\ &= [A' + (B + C)'] \cdot (C' + D') \\ &= (A' + B'C') \cdot (C' + D') \\ &= A'C' + A'D' + B'C'C' + B'C'D' \\ &= A'C' + A'D' + B'C' \end{aligned}$$

例2.4.3 若  $Y = [(A'B)' + C + D]' + C$  , 求反函数

解：由反演定理

$$\begin{aligned} Y' &= [(A + B')' \cdot C' \cdot D']' \cdot C' \\ &= ([ (A + B')' ]' + (C')' + (D')') \cdot C' \\ &= (A + B' + C + D) \cdot C' \\ &= AC' + B'C' + CC' + C'D = AC' + B'C' + C'D \end{aligned}$$

或直接求反得

$$\begin{aligned} Y' &= \{ [(A'B)' + C + D]' + C \}' = \{ [(A'B)' + C + D]' \}' \cdot C' \\ &= [(A'B)' + C + D] \cdot C' = (A + B' + C + D) \cdot C' \\ &= AC' + B'C' + CC' + C'D = AC' + B'C' + C'D \end{aligned}$$

### 3.对偶定理

**对偶式：**设Y是一个逻辑函数，如果将Y中所有的“+”换成与“.”， “.”换成与“+”， “1”换成与“0”， “0”换成与“1”，而变量保持不变，则所得的新的逻辑式 $Y^D$ 称为Y的对偶式。

如：

$$Y = A(B + C') \quad Y^D = A + BC'$$

$$Y = (A + B')(A + C \cdot 1) \quad Y^D = A \cdot B' + A \cdot (C + 0)$$

$$Y = [A' + (BC')']' \quad Y^D = [A' \cdot (B + C')']'$$

**对偶规则**：如果两个函数Y和G相等，则其对偶式 $Y^D$ 和 $G^D$ 也必然相等，Vice versa。利用对偶式可以证明一些常用公式

**例2.4.4** 试利用对偶规则证明分配律  
 $A + BC = (A+B)(A+C)$  式子成立

证明：设  $Y = A + BC$ ， $G = (A+B)(A+C)$ ，则它们的对偶式为

$$Y^D = A \cdot (B + C) = AB + AC$$

$$G^D = AB + AC$$

由于

$$Y^D = G^D$$

故  $Y = G$ ，即  $A + BC = (A+B)(A+C)$

## 2.5 逻辑函数及其表示方法

### 2.5.1 逻辑函数

在数字电路中，输入为二值逻辑变量，输出也是二值变量，则表示输入输出的逻辑函数关系，即

$$Y = F(A_1, A_2 \wedge A_n)$$

则F称为***n*变量的逻辑函数**

其中： $A_1, A_2 \dots A_n$ 称为***n*个输入逻辑变量**，取值只能是“0”或是“1”，Y为**输出逻辑变量**，取值只能是“0”或是“1”

如  $Y = A + B' C$ ，表示输出等于变量B取反和变量C的与，再和变量A相或。



## 2.5.2逻辑函数的几种表示方法

逻辑函数的表示方法很多，比较常用的如下：

### 一、逻辑真值表

逻辑真值表就是采用一种表格来表示逻辑函数的运算关系，其中输入部分列出输入逻辑变量的所有可能取值得组合，输出部分根据逻辑函数得到相应的输出逻辑变量值。

如表2.5.1表示的异或逻辑关系的函数，即

$$Y = A'B + AB'$$

表2.5.1

输入		输出
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

## 二、逻辑函数式

按一定逻辑规律写成的函数形式，也是逻辑代数式。与普通函数数不同的是，逻辑函数式中的输入输出变量都是二值的逻辑变量。

如异或关系的逻辑函数可写成

$$Y = A'B + AB'$$

## 三、逻辑图法

采用规定的图形符号，来构成逻辑函数运算关系的网络图形

图2.5.1表示的是异或关系的逻辑图

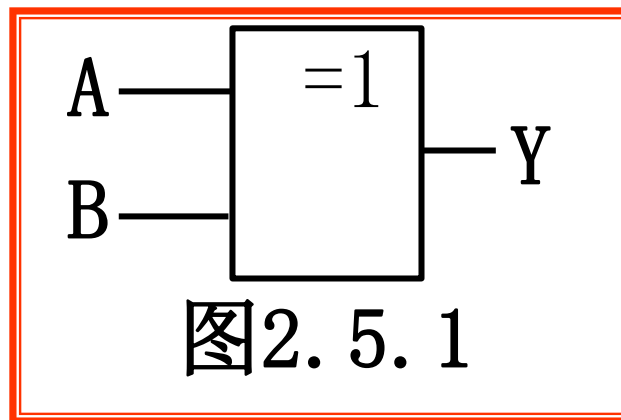


图2.5.1

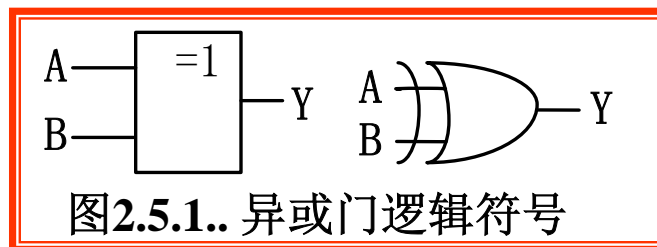


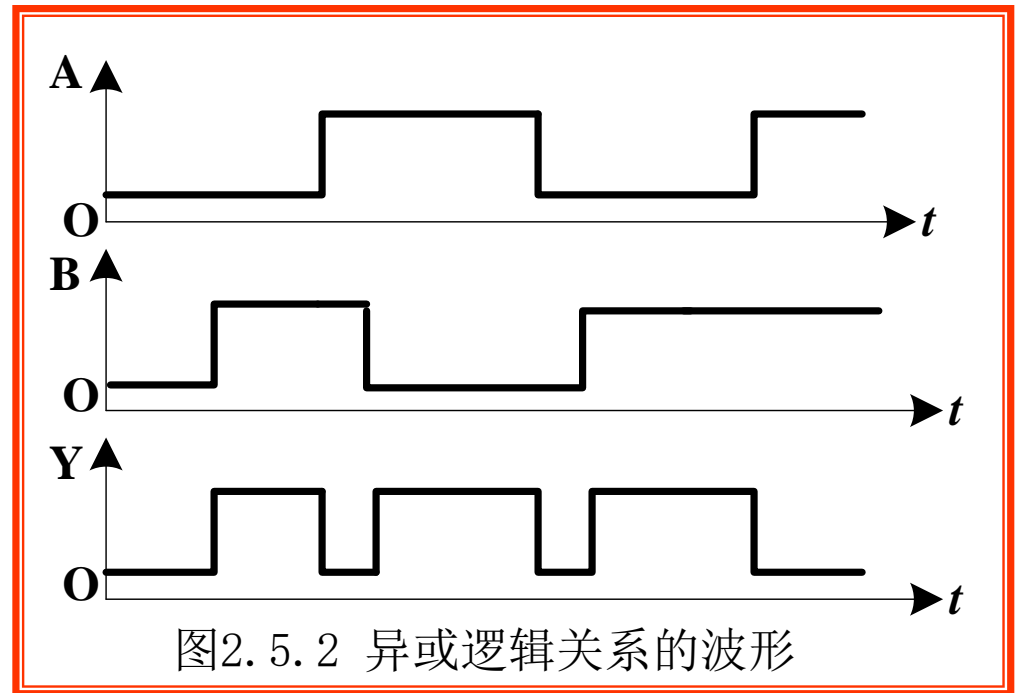
图2.5.1.. 异或门逻辑符号

## 四 波形图法:

一种表示输入输出变量动态变化的图形，反映了函数值随时间变化的规律，也称时序图(timing diagram)。

如图2.5.2表示异或逻辑关系的波形。

除上面介绍的四种逻辑函数表示方法外，还有卡诺图法、点阵图法及硬件描述语言等。在后面的课程中将重点介绍卡诺图法。



## 五、各种表示方法间的相互转换

在设计数字电路时，有时需要进行各种表示逻辑函数方法的转换。

### 1. 真值表与逻辑函数式的相互转换

#### (1) 由真值表写逻辑函数式

通过下面的例子得出  
由真值表写出逻辑函数的  
方法

例2.5.1 某逻辑函数的真值表如表2.5.2所示，写出逻辑函数式

表2.5.2

输入			输出	输出
A	B	C	$Y_1$	$Y_2$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

解：逻辑式为

$$\begin{aligned} Y_1 &= A'B'C + A'BC' + AB'C' + ABC \\ &= (A'B' + AB)C + (A'B + AB')C' \\ &= (A \oplus B)' \cdot C + (A \oplus B) \cdot C' \\ &= (A \oplus B) \oplus C \end{aligned}$$

$$\begin{aligned} Y_2 &= A'BC + AB'C + ABC' + ABC \\ &= (A'B + AB')C + AB \\ &= (A \oplus B)C + AB \end{aligned}$$

表2.5.2

输入			输出	输出
A	B	C	Y <sub>1</sub>	Y <sub>2</sub>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

总结：

①找出真值表中使逻辑函数为“1”的输入变量的组合；

②对应每个输出为“1”变量组合关系为与的关系，即乘积项，其中如图输入变量取值为“1”的写成原变量，输入变量取值为“0”的写成反变量，如 $A'B'C$

③将这些乘积项相加，即得到输出的逻辑式

表2.5.2

输入			输出	输出
A	B	C	$Y_1$	$Y_2$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

## (2) 由逻辑函数式写出真值表

将输入变量所有取值组合，代入逻辑函数式，得出输出的值，并以表的形式表示出来。

**例2.5.2 写出逻辑函数 $Y = AB' + C'$ 的真值表**

解：其真值表如表2.5.3  
示

表2.5.3

输入			输出
A	B	C	Y
0	0	0	1
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

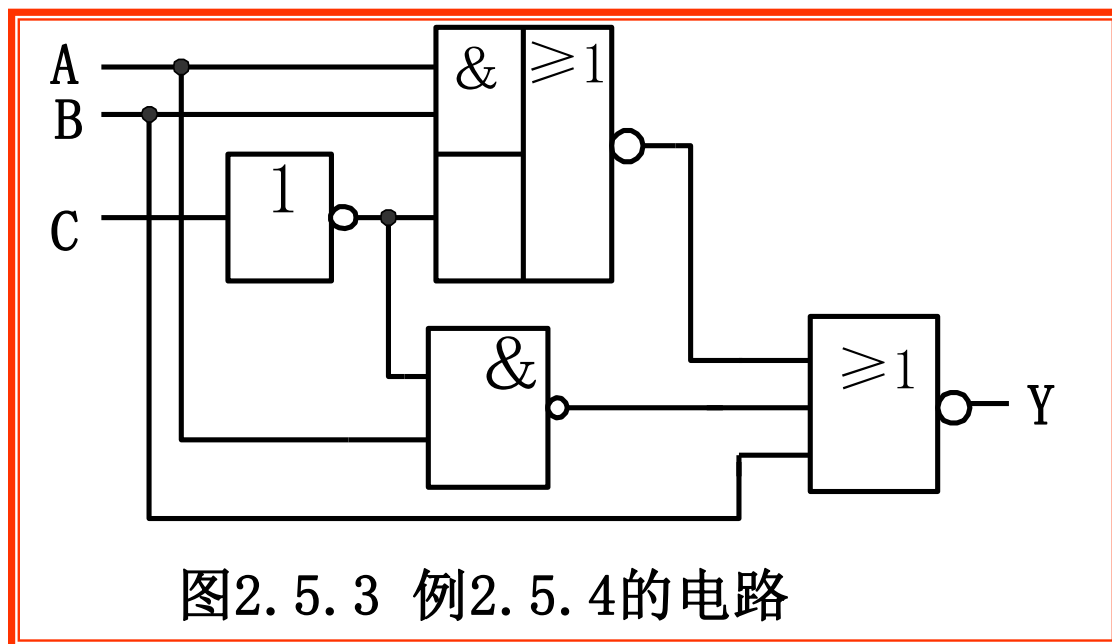
## 2.逻辑函数式与逻辑图的相互转换

### (1) 由逻辑函数式画出逻辑图

用逻辑符号代替逻辑函数中的逻辑关系，即可得到所求的逻辑图

**例2.5.4** 画出逻辑函数 $Y = [ ( AB + C' )' + ( AC' )' + B ]'$ 的逻辑电路

解：其实现电路如图2.5.3所示





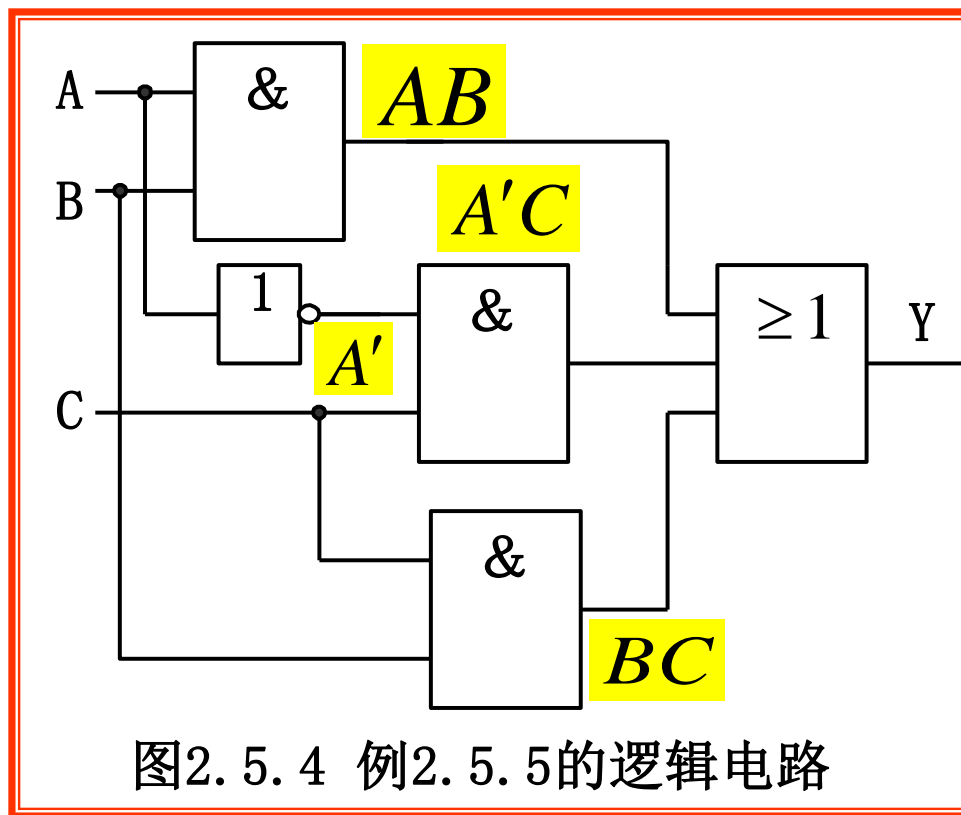
## (2) 由逻辑图写出逻辑函数式

已知逻辑图，根据逻辑门的输入输出关系，写出整个逻辑图的输入输出关系，得出输出的逻辑函数式

例2.5.5 已知逻辑电路如图2.5.4，试写出输出端的逻辑函数式。

解：输出的逻辑式为

$$Y = AB + A'C + BC$$



**例2.5.6** 设计一个逻辑电路，当三个输入A、B、C至少有两个为低电平时，该电路输出为高，试写出该要求的真值表和逻辑表达式，画出实现的逻辑图

解：由逻辑要求写出真值表，表2.5.6  
如表2.5.6所示

输入			输出
A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

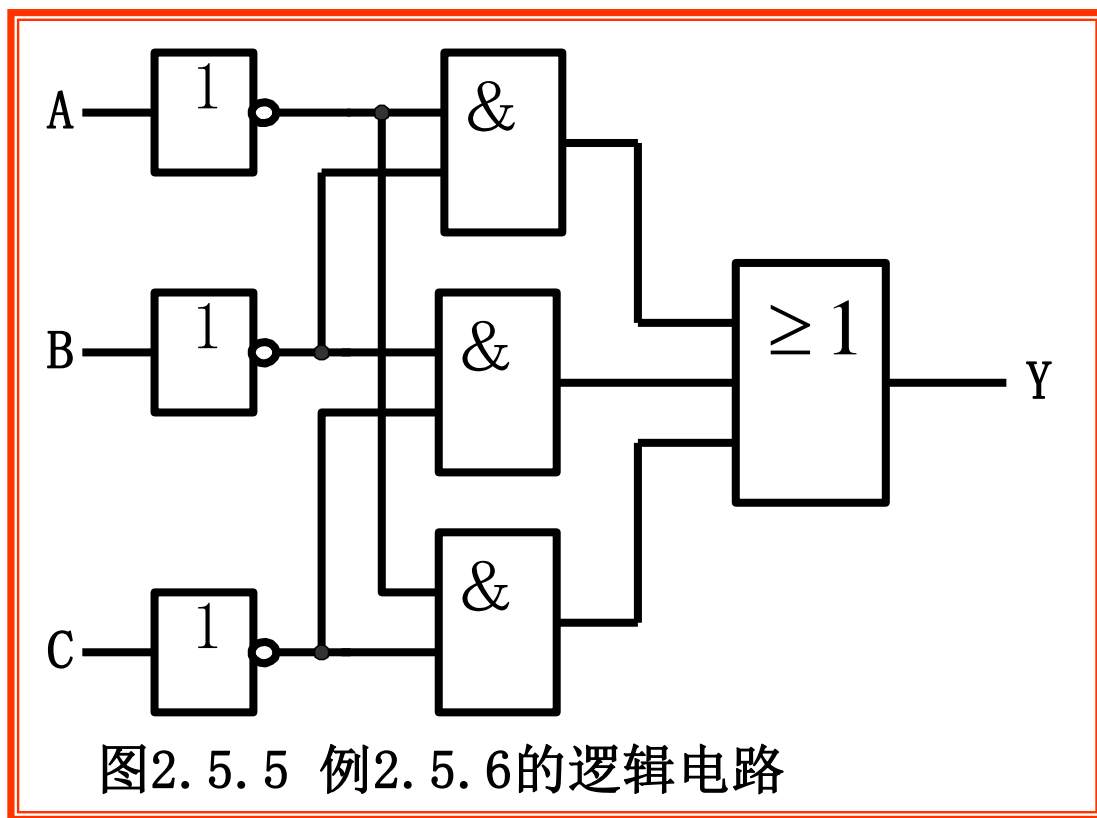
由真值表写出逻辑式为

$$\begin{aligned} Y &= A'B'C' + A'B'C + A'BC' + AB'C' \\ &= A'B'(C' + C) + A'BC' + AB'C' \\ &= A'B' + A'BC' + AB'C' \\ &= A'(B' + BC') + AB'C' \\ &= A'(B' + C') + AB'C' \\ &= A'B' + A'C' + AB'C' \\ &= A'B' + C'(A' + AB') \\ &= A'B' + C'(A' + B') \\ &= A'B' + A'C' + B'C' \end{aligned}$$

表2.5.6

输入			输出
A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

其实现的逻辑图如图2.5.5所示  $A'B' + A'C' + B'C'$



# 作业

题2.3      题2.7      题2.8

### 3.波形图与真值表的相互转换

#### (1) 由波形图得到真值表

根据所给的波形，列出各输入变量组合所对应的输出值

例2.5.7 已知逻辑函数 $Y$ 的输出波形如图2.5.6所示，试分析其逻辑功能。

解：由所给的波形写出输入输出的真值表，如表2.5.7所示

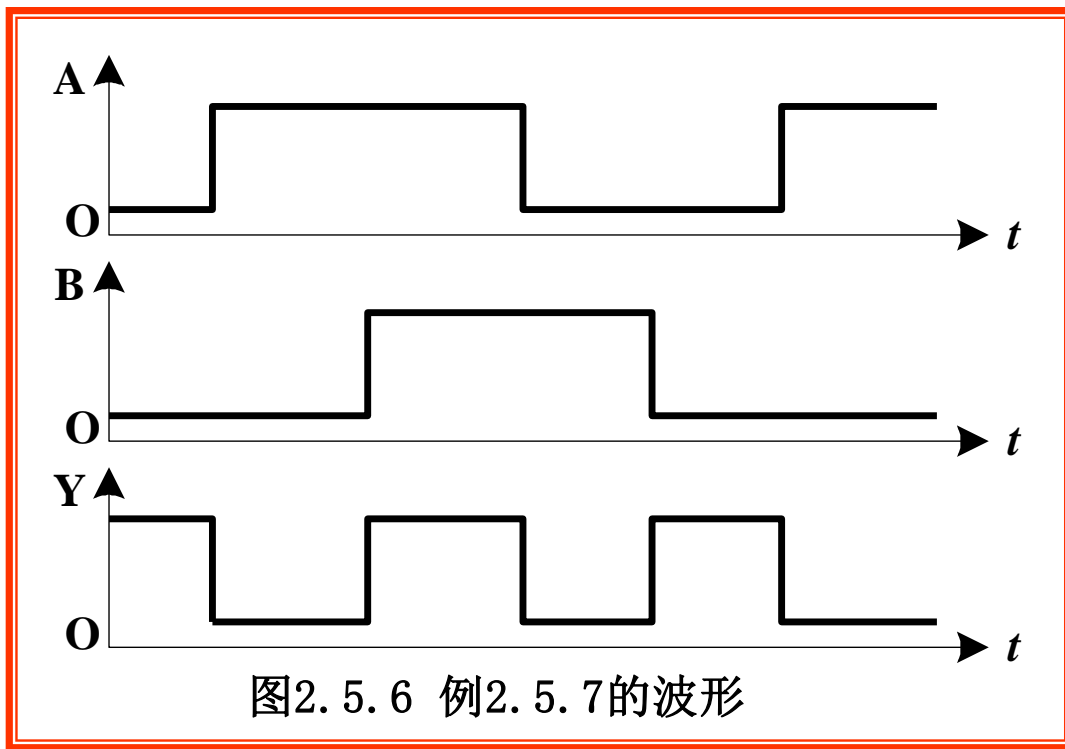
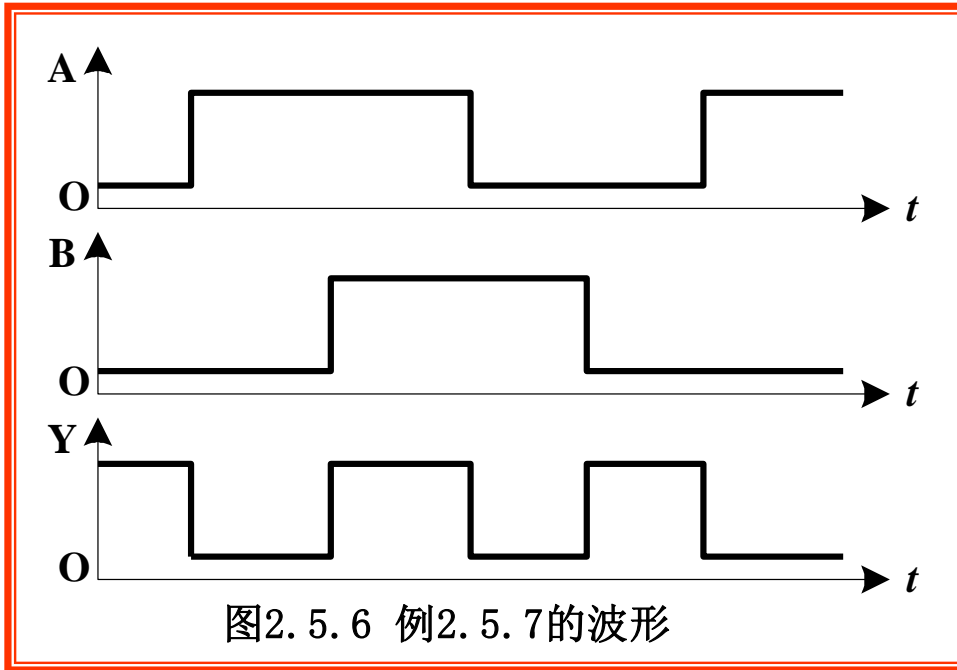


表2.5.7

输入		输出
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1



由真值表可知，当输入变量A、B取值相同时，输出Y = 1；A、B取值不同时，输出Y = 0。故输出和输入是同或关系。其逻辑函数式为

$$Y = A'B' + AB$$

## 2.5.3 逻辑函数的两种标准型

一种输入输出的逻辑关系可以有多种等效的表达式表示，但可以化为标准形式。其标准型有两种：

**标准与或式和标准或与式**

### 一、最小项和最大项

#### 1.最小项

a. 定义: 在 $n$ 变量的逻辑函数中，设有 $n$ 个变量 $A_1 \sim A_n$ ，而 $m$ 是由所有这 $n$ 个变量组成的乘积项（与项）。若 $m$ 中包含的每一个变量都以 $A_i$ 或 $A'_i$ 的形式出现一次且仅一次，则称 $m$ 是 $n$ 变量的最小项。

注： $n$ 个变量构成的最小项有 $2^n$ 个，通常用 $m_i$ 表示第 $i$ 个最小项，变量按 $A_1 \sim A_n$ 排列，以原变量出现时对应的值为“1”，以反变量出现时对应的值取“0”，按二进制排列时，其十进制数即为 $i$ 。



表2.5.10、表2.5.11、表2.5.12分别为二变量、三变量和四变量的最小项

表2.5.10 二变量			
十进制数	A	B	$m_i$
0	0	0	$A'B'(m_0)$
1	0	1	$A'B(m_1)$
2	1	0	$AB'(m_2)$
3	1	1	$AB(m_3)$

表2.5.11 三变量				
十进制数	A	B	C	$m_i$
0	0	0	0	$A'B'C'(m_0)$
1	0	0	1	$A'B'C(m_1)$
2	0	1	0	$A'BC'(m_2)$
3	0	1	1	$A'BC(m_3)$
4	1	0	0	$AB'C'(m_4)$
5	1	0	1	$AB'C(m_5)$
6	1	1	0	$ABC'(m_6)$
7	1	1	1	$ABC(m_7)$

表2.5.12 四变量

A	B	C	D	$m_i$	A	B	C	D	$m_i$
0	0	0	0	$A'B'C'D'(m_0)$	1	0	0	0	$AB'C'D'(m_8)$
0	0	0	1	$A'B'C'D(m_1)$	1	0	0	1	$AB'C'D(m_9)$
0	0	1	0	$A'B'CD'(m_2)$	1	0	1	0	$AB'CD'(m_{10})$
0	0	1	1	$A'B'CD(m_3)$	1	0	1	1	$AB'CD(m_{11})$
0	1	0	0	$A'BC'D'(m_4)$	1	1	0	0	$ABC'D'(m_{12})$
0	1	0	1	$A'BC'D(m_5)$	1	1	0	1	$ABC'D(m_{13})$
0	1	1	0	$A'BCD'(m_6)$	1	1	1	0	$ABCD'(m_{14})$
0	1	1	1	$A'BCD(m_7)$	1	1	1	1	$ABCD(m_{15})$

## b. 最小项的性质(p35)

①在输入变量的任何取值下必有一个最小项，而且仅有一个最小项的取值为1

②n变量组成的全体最小项之逻辑和为“1”。即

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

表2. 5. 10 二变量			
十进制数	A	B	$m_i$
0	0	0	$A'B'(m_0)$
1	0	1	$A'B(m_1)$
2	1	0	$AB'(m_2)$
3	1	1	$AB(m_3)$

## 2.最大项

a. 定义：在 $n$ 变量的逻辑函数中，设有 $n$ 个变量 $A_1 \sim A_n$ ，而 $M$ 是由所有这 $n$ 个变量组成的和项（或项）。若 $M$ 中包含的每一个变量都以 $A_i$ 或 $A'_i$ 的形式出现一次且仅一次，则 $M$ 是 $n$ 变量的最大项。

注： $n$ 个变量构成的最大项也有 $2^n$ 个，通常用 $M_i$ 表示第 $i$ 个最大项，变量按 $A_1 \sim A_n$ 排列，以原变量出现时对应的值为“0”，以反变量出现时对应的值取“1”，按二进制排列时，其十进制数即为 $i$ 。

表2.5.13、表2.5.14分别为二变量、三变量的最大项，  
四变量最大项课下自己写出

表2.5.13 二变量			
十进制数	A	B	$M_i$
0	0	0	$A + B(M_0)$
1	0	1	$A + B'(M_1)$
2	1	0	$A' + B(M_2)$
3	1	1	$A' + B'(M_3)$

表2.5.14 三变量				
十进制数	A	B	C	$M_i$
0	0	0	0	$A + B + C(M_0)$
1	0	0	1	$A + B + C'(M_1)$
2	0	1	0	$A + B' + C(M_2)$
3	0	1	1	$A + B' + C'(M_3)$
4	1	0	0	$A' + B + C(M_4)$
5	1	0	1	$A' + B + C'(M_5)$
6	1	1	0	$A' + B' + C(M_6)$
7	1	1	1	$A' + B' + C'(M_7)$

## b. 最大项的性质(p37)

①对于任一个最大项，仅有一组变量取值使它的值为“0”，而其它取值均使它为“1”。或者说在输入变量的任何取值必有一个最大项也仅有一个最大项的值为“0”。

②n变量组成的全体最大项之逻辑积为“0”。即

$$\prod_{i=0}^{2^n-1} M_i = 0$$

表2.5.13 二变量

十进制数	A	B	$M_i$
0	0	0	$A + B(M_0)$
1	0	1	$A + B'(M_1)$
2	1	0	$A' + B(M_2)$
3	1	1	$A' + B'(M_3)$

## 二、逻辑函数的标准与或式型 - 最小项之和标准型

如

$$Y(A, B) = m_0 + m_3 = A'B' + AB$$

$$\begin{aligned} Y(A, B, C) &= m_0 + m_1 + m_3 + m_5 + m_6 \\ &= A'B'C' + A'B'C + A'BC + AB'C + ABC' \end{aligned}$$

$$\begin{aligned} Y(A, B, C, D) &= m_0 + m_1 + m_3 + m_7 + m_{10} + m_{13} \\ &= A'B'C'D' + A'B'C'D + A'B'CD + A'BCD + AB'CD' + ABC'D \end{aligned}$$

与或型特点：1. 式子为乘积和的形式；

2. 不一定包含所有的最小项，但每一项必须为最小项

## 标准与或式的写法：

在 $n$ 变量的逻辑函数中，若某一乘积项由于缺少一个变量不是最小项，则在这项中添加此变量与这个变量的反变量之和这一项，使之称为最小项，即利用公式 $A + A' = 1$

例2.5.10 将逻辑函数 $Y = A + B'C$ 写成标准与或式

解：

$$\begin{aligned} Y &= A + B'C = A(B' + B)(C' + C) + (A' + A)B'C \\ &= AB'C' + AB'C + ABC' + ABC + A'B'C + AB'C \\ &= m_4 + m_5 + m_6 + m_7 + m_1 \\ &= \sum m(1, 4, 5, 6, 7) \end{aligned}$$

注意：变量的排列顺序。



### 三、 逻辑函数的标准或与式型 - 最大项之积标准型

如  $Y(A, B) = M_1 \cdot M_3 = (A + B')(A' + B')$

$$\begin{aligned} F(A, B, C) &= M_0 \cdot M_2 \cdot M_4 \cdot M_5 \cdot M_7 \\ &= (A + B + C)(A + B' + C)(A' + B + C)(A' + B + C')(A' + B' + C') \end{aligned}$$

或与型特点：1. 式子为和积的形式；

2. 逻辑函数不一定包含所有的最大项，但每一项必须为最大项

## 标准或与式的写法：

在 $n$ 变量的逻辑函数中，若某一和项由于缺少一个变量不是最大项，则在这项中加上此变量与这个变量的反变量之积这一项，即利用公式 $AA' = 0$ ，然后利用公式 $A + BC = (A + B)(A + C)$ 使之称为最大项

例2.5.11 将逻辑函数 $Y = AC + B'C$ 写成标准或与式解：

$$\begin{aligned} Y &= AC + B'C = (AC + B')(AC + C) = (A + B')(B' + C)C \\ &= (A + B' + CC')(B' + C + AA')(C + AA') \\ &= (A + B' + C)(A + B' + C')(B' + C + A)(B' + C + A')(C + A)(C + A') \\ &= M_2 M_3 M_6 (A + C + BB')(A' + C + BB') \\ &= M_2 M_3 M_6 (A + C + B)(A + C + B')(A' + C + B)(A' + C + B') \\ &= M_2 M_3 M_6 M_0 M_4 = \prod M(0, 2, 3, 4, 6) \end{aligned}$$

#### 四、最小项与最大项的关系

设有三变量A、B、C的最小项，如 $m_5 = AB'C$ ，对其求反得

$$m_5 = A\bar{B}C$$

$$\bar{m}_5 = \overline{A\bar{B}C} = \bar{A} + B + \bar{C} = M_5$$

由此可知对于 $n$ 变量中任意一对最小项 $m_i$ 和最大项 $M_i$ ，都是互补的，即

$$\bar{m}_i = M_i \quad \text{或} \quad \bar{M}_i = m_i$$

## 五、标准与或式和或与式之间的关系

若某函数写成最小项之和的形式为

$$Y = \sum m_i$$

则此函数的反函数必为

$$Y' = \sum m_k (k \neq i)$$

如表2.5.15中

$$Y = m_3 + m_6 + m_7 = \sum m_i (3,6,7)$$

$$\begin{aligned} Y' &= m_0 + m_1 + m_2 + m_4 + m_5 \\ &= \sum m_k (0,1,2,4,5) \end{aligned}$$

表2.5.15 逻辑函数Y的真值表

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	0
1	1	0	1
1	1	1	1

$$Y' = m_0 + m_1 + m_2 + m_4 + m_5$$

$$= \sum m_k (0,1,2,4,5)$$

上式或写成

$$Y = (m_0 + m_1 + m_2 + m_4 + m_5)'$$

$$= m'_0 \cdot m'_1 \cdot m'_2 \cdot m'_4 \cdot m'_5$$

利用反演定理可得

$$Y = \overline{\sum m_k (k \neq i)} = \prod_{k \neq i} \bar{m}_k = \prod_{k \neq i} M_k$$

## \*\*2.5.4 逻辑函数形式的变换

除了上述标准与或式和标准或与式的外，还需要将逻辑函数变换成其它形式。假如给出的是一般与或式，要用与非门实现，就需要将其变成与非-与非式。

### 一、与或式化为与非-与非式 - 利用反演定理

**例2.5.10** 将下式 $Y=AC+BC'$ 用与非门实现，并画出逻辑图。

解：用二次求反，将第一级非号用摩根定理拆开，第二级保持不变。

$$Y = ((AC + BC')')' = ((AC)' \cdot (BC')')'$$

$$Y = ((AC + BC')')' = ((AC)' \cdot (BC')')$$

如果本身有反变量输入，则用二级与非门就可实现该函数，其逻辑电路如图2.5.10所示。

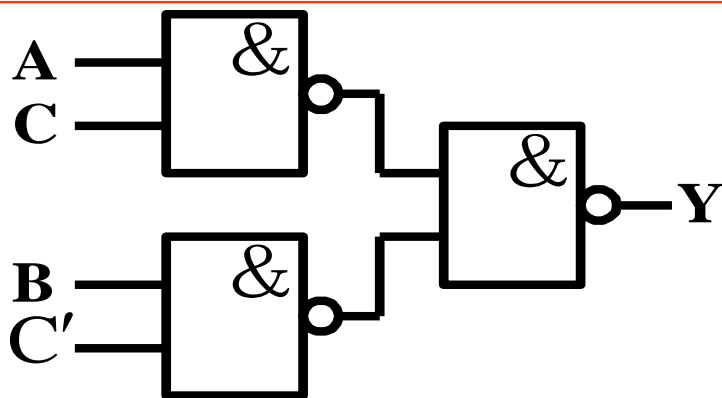


图2.5.10 输入有反变量输入

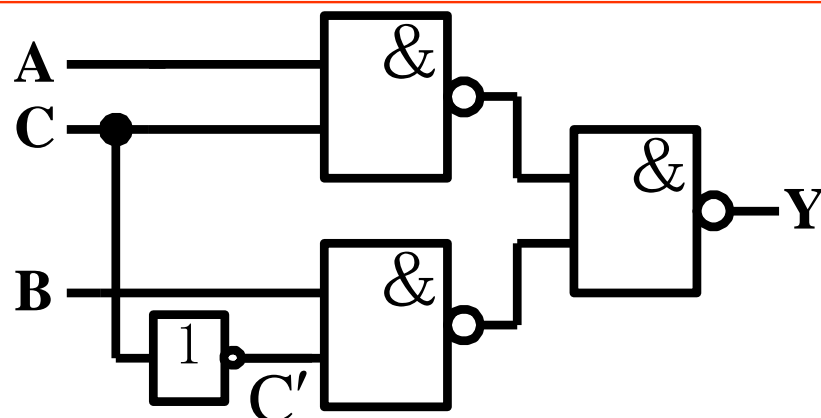


图2.5.11 输入只有原变量输入

如果只有原变量输入，另外要用与非门实现反相 $C'$ ，其逻辑电路如图2.5.11所示

## 二、将与非式化为与或非式

例2.5.11将 $Y=AC+BC'$ 用与或非门实现，画出逻辑图。

解：先用反演定理求函数 $Y$ 的反函数 $Y'$ ，并整理成与或式，再将左边的反号移到等式右边，即两边同时求反。

多余项

$$Y' = (AC + BC')' = (A' + C')(B' + C) \\ = A'B' + A'C + B'C'$$

$$Y' = B'C' + A'C$$

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

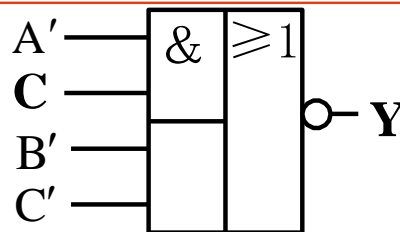


图2.5.12 与或非实现  
逻辑函数

这就可用与或非门实现。  
其电路如图2.5.12所示



### 三、将与或式化为或非 - 或非式

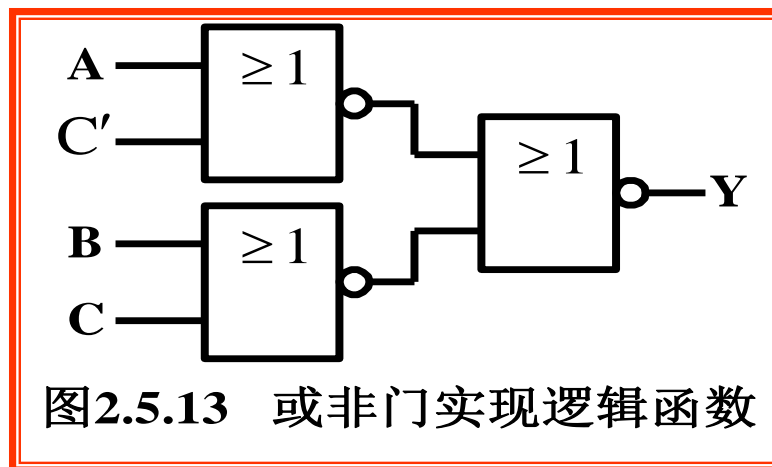
例2.5.11 将下式 $Y=AC+BC'$ 用或非门实现。

解：先将函数 $Y$ 化为与或非形式，再用反演定理求 $Y'$ ，并用摩根定理展开，再求 $Y$ ，就可得到或非 - 或非式。

$$Y = AC + BC'$$
$$= (B'C' + A'C)'$$

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

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



或者先写成最大项之积形式，再两次取反，利用反演定理得到或非式

$$\begin{aligned} Y &= AC + BC' = A(B + B')C + (A + A')BC' \\ &= ABC + AB'C + ABC' + A'BC' \\ &= \sum m(2,5,6,7) = \prod M(0,1,3,4) \\ &= (A + B + C)(A + B + C')(A + B' + C')(A' + B + C) \\ &= (A + C')(B + C) \end{aligned}$$

$$\begin{aligned} Y &= (Y')' = [((A + C')(B + C))']' \\ &= [(A + C')' + (B + C)']' \end{aligned}$$

# 作业

题2.10 (1) (6)

题2.11 (4)

写成

$$Y = \sum m_i$$

$$Y = \prod M_k$$

## 2.6 逻辑函数的化简方法

一个逻辑函数有多种不同形式的逻辑表达式，虽然描述的逻辑功能相同，但电路实现的复杂性和成本是不同的。逻辑表达式越简单，实现的电路越简单可靠，且低成本。因此在设计电路时必须将逻辑函数进行简化。

**注：**随着集成电路的发展，集成芯片的种类越来越多。逻辑函数是否“最简”已无太大意义。但作为设计思路，特别对于中小规模集成电路，逻辑函数的简化是不能忽视的

逻辑函数的简化方法很多，主要有逻辑代数简化法（**公式法**）和**卡诺图法**

## 2.6.1 公式化简法

公式法化简就是利用逻辑代数的一些定理、公式和运算规则，将逻辑函数进行简化。实现电路的器件不同，最终要得到的逻辑函数的形式不同，其最简的定义也不同。

对于要小规模集成门电路实现的电路，常用的门为与非门、或非门、与或非门等。由上一节可知，其最终都可以由与或式、或与式转换而成。故最常用的是最简与或式和最简或与式。

最简与或式：最简的与或式所含乘积项最少，且每个乘积项中的因子也最少。

最简或与式：最简的或与式所含和项最少，且每个和项中的相加的项也最少。

## 2.6.1 公式化简法

下面讨论公式法常用的化简方法。

### 1. 与或式的简化

(1) 与或式：就是先与后或式（乘积和），最简的与或式是所含与项最少，且每个与项的逻辑变量最少，则这个与或式是最简的。

$$Y_1 = ABC + B'C + ACD$$

$$Y_2 = AC + B'C$$

上式 $Y_1$ 和 $Y_2$ 实现同样的逻辑功能，但 $Y_1$ 中不仅所含变量多，而且乘积项也多了一项，要用3个与门（不含非门）和一个或门实现，而 $Y_2$ 的变量有3个，两个乘积项，用2个与门、1个或门实现即可，这样即节省元件，也减少布线和功耗。

## 2.6.1 公式化简法

### (2) 与或式的简化方法

- a. **合并项法**：利用  $AB + A'B = B$  消去一个变量；
- b. **消因子法**：利用  $A + A'B = A + B$  消去多余变量；
- c. **配项法**：利用  $A + A' = 1$  增加一些项，再进行简化

**吸收法，消项法。。。**

说明：一般化简需要各种方法综合起来。化简需要技巧和经验，需多练习。另外最后的结果是否为最简，难以判断。

## 2.6.1 公式化简法

例2.6.1 将下式化为最简与或式

$$Y = AB'C + A'BC + ABC' + ABC$$

解法一：配项法

$$Y = AB'C + A'BC + ABC' + ABC$$

配项ABC

$$Y = (AB'C + ABC) + (A'BC + ABC) + (ABC' + ABC)$$

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

$$= AC + BC + AB$$



## 2.6.1 公式化简法

解法二：用吸收法和消因子法

$$Y = AB'C + A'BC + ABC' + ABC$$

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

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

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

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

$$= AC + BC + AB$$

二种方法结果一致，但过程繁简不同。尽量选择最佳方法，使化简过程简单

## 2.6.1 公式化简法

例2.6.2 试将下面的逻辑函数简化为最简与或式

$$Y = AB' + A'B + BC' + B'C$$

解  $Y = AB' + A'B + BC' + B'C$

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

$$= AB'C + AB'C' + A'B + ABC' + A'BC' + B'C$$

$$= B'C(A + 1) + A'B(1 + C') + A'C'(B + B')$$

$$= B'C + A'B + AC'$$

与p39例2.6.6对比

注：从原式看，很难看出是不是最简，而且用代数法简化逻辑函数，不仅要熟悉逻辑代数公式，而且要灵活运用，而且不能保证最后结果最简。

## 2.6.1 公式化简法

例2.6.3 试将下面逻辑函数简化成最简与或式

$$Y = AC + B'C + BD' + CD' + A(B + C') + A'BCD' + AB'DE$$

解：

反演定理

$$Y = AC + B'C + BD' + CD' + A(B + C') + A'BCD' + AB'DE$$

$$= AC + B'C + BD' + CD'(1 + A'B) + A(B'C) + AB'DE$$

$$= AC + A + B'C + BD' + CD' + AB'DE$$

$$= A(C + 1 + B'DE) + B'C + BD'$$

$$= A + B'C + BD'$$

多余项

## 2.6.2 卡诺图化简法

公式法简化逻辑函数不直观，且要熟练掌握逻辑代数的公式以及简化技巧，而卡诺图法能克服公式法的不足，可以直观地给出简化的结果。

### 一.卡诺图

a. 定义：将逻辑函数的真值表图形化，把真值表中的变量分成两组分别排列在行和列的方格中，就构成二维图表，即为卡诺图，它是由卡诺（**Karnaugh**）和范奇（**Veich**）提出的。

b. 卡诺图的构成：将最小项按相邻性排列成矩阵，就构成卡诺图,实质是**将逻辑函数的最小项之和以图形的方式表示出来**。最小项的**相邻性**就是它们中变量只有一个是不同的。

## 2.6.2 卡诺图化简法

下面表2.6.1 是二变量的卡诺图

表2. 5. 10 二变量			
十进制数	A	B	$m_i$
0	0	0	$A'B'(m_0)$
1	0	1	$A'B(m_1)$
2	1	0	$AB'(m_2)$
3	1	1	$AB(m_3)$

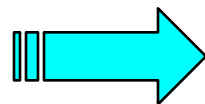


表2.6.1 二变量的卡诺图		
A \ B	0	1
0	$m_0$	$m_1$
1	$m_2$	$m_3$

## 2.6.2 卡诺图化简法

表2.6.2为三变量的卡诺图

表2.5.11 三变量				
十进制数	A	B	C	$m_i$
0	0	0	0	$A'B'C'(m_0)$
1	0	0	1	$A'B'C(m_1)$
2	0	1	0	$A'BC'(m_2)$
3	0	1	1	$A'BC(m_3)$
4	1	0	0	$AB'C'(m_4)$
5	1	0	1	$AB'C(m_5)$
6	1	1	0	$ABC'(m_6)$
7	1	1	1	$ABC(m_7)$

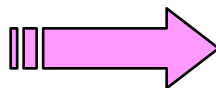


表2.6.2 三变量的卡诺图					
BC A		00	01	11	10
		$m_0$	$m_1$	$m_3$	$m_2$
0		$m_0$	$m_1$	$m_3$	$m_2$
1		$m_4$	$m_5$	$m_7$	$m_6$

## 2.6.2 卡诺图化简法

表2.6.3为4变量的卡诺图

表2.6.3 四变量的卡诺图

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

## 2.6.2 卡诺图化简法

从上面卡诺图可以看出

任意两个相邻的最小项在图上是相邻的，并且图中最左列的最小项与左右列相应最小项也是相邻的（如 $m_0$ 和 $m_2$ ， $m_8$ 和 $m_{10}$ ）。位于最上面和最下面的相应最小项也是相邻的（ $m_0$ 和 $m_8$ ， $m_2$ 和 $m_{10}$ ），所以四变量的最小项有四个相邻最小项。可以证明 $n$ 变量的卡诺图中的最小项有 $n$ 个相邻最小项

表2.6.3 四变量的卡诺图

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



## 2.6.2 卡诺图化简法

$n$ 变量的卡诺图可有 $n - 1$ 变量的卡诺图采用折叠法构成,如五变量的卡诺图可由四变量的卡诺图折叠得到,如表2.6.4

表2.6.4 五变量的卡诺图									
CDE AB		C取0				C取1			
		000	001	011	010	110	111	101	100
00	$m_0$	$m_1$	$m_3$	$m_2$	$m_6$	$m_7$	$m_5$	$m_4$	
01	$m_8$	$m_9$	$m_{11}$	$m_{10}$	$m_{14}$	$m_{15}$	$m_{13}$	$m_{12}$	
11	$m_{24}$	$m_{25}$	$m_{27}$	$m_{26}$	$m_{30}$	$m_{31}$	$m_{29}$	$m_{28}$	
10	$m_{16}$	$m_{17}$	$m_{19}$	$m_{18}$	$m_{22}$	$m_{23}$	$m_{21}$	$m_{20}$	

格雷码？

## 2.6.2 卡诺图化简法

### 二. 逻辑函数的卡诺图表示法

如果画出逻辑函数的卡诺图，首先将逻辑函数化成标准与或型（**最小项和**），在相应的最小项位置填“1”，其方法如下

**a. 利用真值表：**将逻辑函数的真值表做出，将表中对应“1”项的最小项填到卡诺图中

例2.6.5 画出下面函数的卡诺图

$$Y = A'B + ABC$$

## 2.6.2 卡诺图化简法

$$Y = A'B + ABC$$

$$\begin{aligned} Y &= A'B + ABC \\ &= B(A' + AC) \\ &= B(A' + C) \\ &= A'B + BC \end{aligned}$$

解：其真值表如表2.6.5所示,其卡诺图如表2.6.6所示

表2.6.5

输入			输出
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

表2.6.6 Y的卡诺图

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

$$Y = BC + A'B$$

## 2.6.2 卡诺图化简法

### b.化为标准与或型

$$Y = \sum m_i$$

例2.6.6 画出下面逻辑函数的卡诺图

$$Y = A'BD + B'D' + A'B'D$$

解：

$$\begin{aligned} Y &= A'BD + B'D' + A'B'D \\ &= A'B(C + C')D + (A + A')B'(C + C')D' \\ &\quad + A'B'(C + C')D \\ &= A'BCD + A'BC'D + AB'CD' + AB'C'D' \\ &\quad + A'B'CD' + A'B'C'D' + A'B'CD + A'B'C'D \\ &= m_7 + m_5 + m_{10} + m_8 + m_2 + m_0 + m_3 + m_1 \\ &= \sum m(0, 1, 2, 3, 5, 7, 8, 10) \end{aligned}$$

## 2.6.2 卡诺图化简法

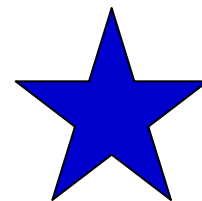
卡诺图如表2.6.6

$$Y = \sum m(0, 1, 2, 3, 5, 7, 8, 10)$$

表2.6.6 Y的卡诺图

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

## 2.6.2 卡诺图化简法



重点掌握

### c 观察法

采用观察法不需要前两种方法需要将逻辑函数转换成最小项，而是采用观察逻辑函数，将应为“1”的项填到卡诺图中

例2.6.7 用卡诺图表示下面的逻辑函数

$$Y = A'B'C'D + A'BD' + ACD + AB'$$

解：其卡诺图如表2.6.7所示

表2.6.7 Y的卡诺图

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

## 2.6.2 卡诺图化简法

### 三、利用卡诺图简化逻辑函数

#### ①卡诺图的性质

a. 卡诺图上任何2 (  $2^1$  )个标“1”的相邻最小项合并成一项，并消去1个取值不同的变量

例如表2.6.10中，有

$$\begin{aligned} m_2 + m_3 &= A'B'CD + A'B'CD' \\ &= A'B'C(D + D') \\ &= A'B'C \end{aligned}$$

消去变量D

表2.6.10 Y的卡诺图

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

## 2.6.2 卡诺图化简法

b. 卡诺图上任何4 (  $2^2$  )个标“1”的相邻最小项，可以合并成一项，并消去2个取值不同的变量

例如表2.6.11中，有

$$\begin{aligned} & m_5 + m_7 + m_{13} + m_{15} \\ &= \bar{A}\bar{B}\bar{C}D + \bar{A}BCD + AB\bar{C}D + ABCD \\ &= BD(\bar{A}\bar{C} + \bar{A}C + A\bar{C} + AC) \\ &= BD \end{aligned}$$

表2.6.11 Y的卡诺图

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

消去变量AC



## 2.6.2 卡诺图化简法

表2.6.11 Y的卡诺图

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

$$\begin{aligned} & m_0 + m_2 + m_8 + m_{10} \\ &= A'B'C'D' + A'B'CD' + AB'C'D' + AB'CD' \\ &= B'D'(A'C' + A'C + AC' + AC) = B'D' \end{aligned}$$

## 2.6.2 卡诺图化简法

c. 卡诺图上任何8 ( $2^3$ )个标“1”的相邻最小项，可以合并成一项，并消去3个取值不同的变量

消去变量ABC

例如表2.6.12中，有

$$\begin{aligned} & m_1 + m_3 + m_5 + m_7 + m_9 + m_{11} + m_{13} + m_{15} \\ &= A'B'C'D + A'B'CD + A'BC'D + A'BCD \\ &+ AB'C'D + AB'CD + ABC'D + ABCD \\ &= D(A'B'C' + A'B'C + A'BC' + A'BC \\ &+ AB'C' + AB'C + ABC' + ABC) \\ &= D \end{aligned}$$

表2.6.12 Y卡诺图

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

## 2.6.2 卡诺图化简法

或者下面的圈“1”法

表2.6.12 Y的卡诺图

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

$$Y = \sum m(0,1,2,3,8,9,10,11) \\ = B'$$

## 2.6.2 卡诺图化简法

### ②卡诺图简化逻辑函数为与或式的步骤

- a. 将逻辑函数化为最小项（可略去）；
- b. 画出表示该逻辑函数的卡诺图；
- c. 找出可以合并的最小项,即1的项（必须是 $2^n$ 个1），进行圈“1”，圈“1”的规则为：
  - \* 圈内的“1”必须是 $2^n$ 个；
  - \* “1”可以重复圈，但每圈一次必须包含没圈过的“1”；
  - \* 每个圈包含“1”的个数尽可能多，但必须相邻，必须为 $2^n$ 个；

## 2.6.2 卡诺图化简法

圈“1”的规则为

- \* 圈数尽可能的少；
- \* 要圈完卡诺图上所有的“1”。

d. 圈好“1”后写出每个圈的乘积项，然后相加，即为简化后的逻辑函数。

注：卡诺图化简不是唯一，不同的圈法得到的简化结果不同，但实现的逻辑功能相同的。

## 2.6.2 卡诺图化简法

例2.6.10 用卡诺图简化下面逻辑函数(p43)

$$Y = AC' + A'C + BC' + B'C$$

解：其卡诺图如表2.6.13所示  
圈法如图，则

$$Y = AB' + A'C + BC'$$

表2.6.13 Y的卡诺图

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

## 2.6.2 卡诺图化简法

或者圈法如表2.6.14所示，  
则

$$Y = B'C + A'B + AC'$$

与第一种圈法相比

$$Y = AB' + A'C + BC'$$

故卡诺图简化不是唯一的

表2.6.14 Y的卡诺图

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

表2.6.13 Y的卡诺图

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

## 2.6.2 卡诺图化简法

例2.6.11 用卡诺图简化下面逻辑函数(p44)

$$Y = ABC + ABD + AC'D + C'D' + AB'C + A'CD'$$

解：其卡诺图如表2.6.15所示  
则简化后的逻辑函数为

$$\begin{aligned} Y &= ABC + ABD + AC'D \\ &\quad + C'D' + AB'C + A'CD' \\ &= A + D' \end{aligned}$$

表2.6.15 Y的卡诺图

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



## 2.6.2 卡诺图化简法

注：以上是通过合并卡诺图中的“1”项来简化逻辑函数的，有时也通过合并“0”项先求F的反函数，再求反得Y

例如上面的例题,圈“0”情况  
如表2.6.15所示，可得

$$Y' = A'D$$

$$Y = (A'D)' = A + D'$$

表2.6.15 Y的卡诺图

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

## 2.6.2 卡诺图化简法

### 例2.6.12 用卡诺图简化下面逻辑函数

$$Y(A, B, C, D) = \sum m(0, 1, 2, 3, 4, 6, 8, 9, 10, 12, 14)$$

解:卡诺图如表2.6.16

可得

$$Y = D' + A'B' + B'C'$$

表2. 4. 16 Y的卡诺图

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

### ③ 利用卡诺图简化逻辑函数为或与式



了解

在卡诺图上圈“0”的最小项，其规则与化成与或式相同，但写最简或与式时，消去取值不同的变量，保留取值相同的变量。写相同变量时，取值为“0”写成原变量，取值为“1”写成反变量，每个圈写这些相同变量的和，不同的圈为乘积的关系。

$$Y = \overline{\sum m_k (k \neq i)} = \prod_{k \neq i} \overline{m_k} = \prod_{k \neq i} M_k$$

## 2.6.2 卡诺图化简法

例2.6.13 用卡诺图将下面逻辑函数简化成最简与或式和或与式

$$Y(A, B, C, D) = \sum m(1, 4, 5, 6, 7, 8, 10, 14)$$

解：其卡诺图如表2.6.17所示  
对于与或式，圈“1”，则

$$\begin{aligned} Y(A, B, C, D) \\ = A'B + A'C'D + ACD' + AB'D' \end{aligned}$$

注：Y的最简与或式不是唯一的

表2.6.17 Y的卡诺图

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

## 2.6.2 卡诺图化简法

对于或与式，圈“0”，则

$$Y(A, B, C, D) = (A' + D')(A + B + C')(A' + B' + C)(A + B + D)$$

由表2.6.17的卡诺图可得

$$Y' = AD + A'B'C + ABC' + A'B'D'$$

故

$$\begin{aligned} Y &= (AD + A'B'C + ABC' + A'B'D')' \\ &= (AD)' \cdot (A'B'C)' \cdot (ABC')' \cdot (A'B'D')' \\ &= (A' + D')(A + B + C') \\ &\quad (A' + B' + C)(A + B + D) \end{aligned}$$

表2.6.17 Y的卡诺图

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

## 2.6.2 卡诺图化简法

例2.6.14 试将下面逻辑函数化成最简与或式和或与式。

$$Y = \sum m(4,5,6,13,14,15)$$

解：卡诺图如表2.6.18所示  
圈“1”化成最简与或式，则  
可得

$$Y = A'BC' + ABD + BCD'$$

表2.6.18 Y的卡诺图

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

## 2.6.2 卡诺图化简法

圈“0”化成最简或与式为

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

表2.6.18 Y的卡诺图

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

$$Y = \sum m_k (k \neq i) = \prod_{k \neq i} \bar{m}_k = \prod_{k \neq i} M_k$$

## 2.6.2 卡诺图化简法

例2.6.15 试将下面逻辑函数化成最简与或式和或与式

$$Y = \prod M(2,3,4,6,11,12,14)$$

解：由于最大项对应输入函数取值为“0”，如  $M_6 = A + B' + C' + D$ ，当  $ABCD = 0110$  时， $M_6 = 0$ ，故在相应最大项的位置上填“0”即可得逻辑函数的卡诺图。则Y的卡诺图如表2.6.19所示

则最简与或式为

$$Y = BD + B'C' + AB'D'$$

表2.6.19 Y的卡诺图

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



## 2.6.2 卡诺图化简法

圈“0”可得最简的或与式为

表2.6.19 Y的卡诺图

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

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

$$Y = \sum \overline{m_k (k \neq i)} = \prod_{k \neq i} \overline{m_k} = \prod_{k \neq i} M_k$$

**\*2.6.3 奎恩 - 麦克拉斯基化简法（Q - M法）（自学）**

## 2.7 具有无关项的逻辑函数及其化简

### 2.7.1 约束项、任意项和逻辑函数式中的无关项

#### 1. 定义：

**a. 约束项**：在逻辑函数中，输入变量的取值不是任意的，受到限制。对输入变量取值所加的限制称为约束，被约束的项叫做约束项。

例如有三个逻辑变量A、B、C分别表示电动机的正转、反转和停止。若A=1表示电动机正转，B=1表示电动机反转，C=1表示电动机停止。由于电动机不可能同时正转和反转，也不可能正转和停止，因此ABC的取值只能是100、010、001，而其它的状态如000、011、101、110、111是不能出现的状态，故ABC为具有约束的变量，恒为0。可写成

这些恒等于“0”的最小项称为约束项

$$A'B'C' + A'BC + AB'C + ABC' + ABC = 0$$

## 2.7.1 约束项、任意项和逻辑函数式中的无关项

b.任意项：输入变量的某些取值对电路的功能没影响，  
这些项称为任意项。

例如8421BCD码取值为0000 ~ 1001十个状态，而1010~1111这六个状态不可能出现，故对应的函数取“0”或取“1”对函数没有影响，这些项就是任意项。

c.无关项：将约束项和任意项统称为无关项。即把这些最小项是否写入卡诺图对逻辑函数无影响

## 2. 含有无关项的逻辑函数的表示方法

最小项的表达式为  $Y = \sum m + \sum d$  其中 $\sum d$ 为无关项

也可以写成

$$\begin{cases} Y = \sum m + \sum d \\ \text{约束条件: } \sum d = 0 \end{cases}$$

## 2.7.1 约束项、任意项和逻辑函数式中的无关项

### 3. 无关项在化简逻辑函数中的应用

步骤：

- ① 将给定的逻辑函数的卡诺图画出来；
- ② 将无关项中的最小项在卡诺图相应位置用“×”表示出来；
- ③ 化简时，根据需要无关项可以作为“1”也可作“0”处理，以得到相邻最小项矩形组合最大（包含“1”的个数最多）为原则。

**利用无关项可以使得函数进一步简化**

## 2.7.1 约束项、任意项和逻辑函数式中的无关项

例2.6.1 用卡诺图简化下列逻辑函数，并写成最简与或式和或与式

$$Y(A, B, C, D) = \sum (0, 1, 6, 9, 14, 15) + \sum d(2, 4, 7, 8, 10, 11, 12, 13)$$

解：Y的卡诺图如表2.6.1所示

则最简与或式为

$$Y = D' + A + B'C'$$

表2.6.1 Y的卡诺图

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

## 2.7.1 约束项、任意项和逻辑函数式中的无关项

还有另一种圈法，如图2.6.2所示

此种圈法圈数少，变量少，  
比上一种简单

简化后的逻辑函数为

$$Y = B'C' + BC$$

表2.6.2 Y的卡诺图

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

## 2.7.1 约束项、任意项和逻辑函数式中的无关项

写成或与式为

$$Y = (B' + C)(B + C')$$

表2.6.3 Y的卡诺图

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



## 2.7.1 约束项、任意项和逻辑函数式中的无关项

例1.4.13 试简化下列逻辑函数，写成最简与或式和或与式

$$\begin{cases} Y(A, B, C, D) = A'BC' + A'BCD' + AB'CD' + AB'CD \\ \text{约束条件: } A \odot B = 0 \end{cases}$$

解：约束条件为

$$A'B' + AB = 0$$

(即AB取值不能相同)

则Y的卡诺图如表2.6.4所示

最简与或式为

$$Y = AC + A'C' + CD'$$

表2.6.4 Y的卡诺图

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

## 2.7.1 约束项、任意项和逻辑函数式中的无关项

圈“0”

则最简或与式为

$$Y = (A' + C)(A + C' + D')$$

表2.6.4 Y的卡诺图

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

## \*2.7 卡诺图的其它应用

卡诺图除了简化逻辑函数，还可以有下面的一些应用

### 2.7.1.判明函数关系和进行函数的运算

#### 1 判明函数关系

利用卡诺图可以判明函数是否相等、互补。若两个函数的卡诺图相同，则这两个函数一定相等。即若函数 $Y$ 和 $G$ 的卡诺图相同，则 $Y = G$ 。若两个函数的卡诺图中“0”和“1”对调，则这两个函数为互补。

## 2.7.1.判明函数关系和进行函数的运算

例如

$$Y = AB + A'C + BC$$

$$G = AB + A'C$$

它们的卡诺图如表  
2.7.1所示，则 $Y = G$

表2.7.1 Y和G的卡诺图

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

## 2.7.1.判明函数关系和进行函数的运算

再例如  $Y = A'B + C$

$$G = AC' + B'C'$$

它们的卡诺图如表2.7.2和2.7.3所示

则  $Y = G'$

表2.7.2 Y的卡诺图

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

表2.7.3 G的卡诺图

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

## 2.7.1.判明函数关系和进行函数的运算

### 2.函数运算

若已知函数 $Y_1$ 和 $Y_2$ ，则可利用卡诺图做逻辑运算。

例2.7.1若 $Y_1 = A'B + AC'$ ， $Y_2 = A + BC$ 试利用卡诺图求 $Y_1 + Y_2$ 、 $Y_1 \cdot Y_2$ 及 $Y_1 \odot Y_2$

解：  $Y_1$ 和 $Y_2$ 的卡诺图如表2.7.4及2.7.5所示

表2.7.4  $Y_1$ 的卡诺图

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

表2.7.5  $Y_2$ 的卡诺图

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

## 2.7.1.判明函数关系和进行函数的运算

### 则两个函数的与为

表2.7.4  $Y_1$ 的卡诺图

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

表2.7.5  $Y_2$ 的卡诺图

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

•

表2.7.6  $Y$ 的卡诺图

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

$$Y = Y_1 \cdot Y_2$$

$$= AC' + A'BC$$

## 2.7.1.判明函数关系和进行函数的运算

则两个函数的或为

表2.7.4 Y<sub>1</sub>的卡诺图

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

+

表2.7.5 Y<sub>2</sub>的卡诺图

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

表2.7.7 Y的卡诺图

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

$$Y = Y_1 + Y_2 = A + B$$



## 2.7.1.判明函数关系和进行函数的运算

则两个函数的同或为

表2.7.4  $Y_1$ 的卡诺图

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



表2.7.5  $Y_2$ 的卡诺图

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

表2.7.8  $Y$ 的卡诺图

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

$$Y = Y_1 \odot Y_2$$

$$= AC' + B'C' + A'C$$

## 2.7.2 逻辑函数表达式类型的转换

逻辑函数表达式的形式有很多种，如与或式、或与式、与非式、与或非式等，不同的表达式可由不同的门电路来实现。一般的逻辑函数为与或式（乘积和），这样需要转换成其它的形式，利用卡诺图可以很方便的实现转换。

### 1. 与或式转换成或与式

已知逻辑函数的与或式，先画出逻辑函数的卡诺图，再圈“0”，便可得到最简的或与式。

## 2.7.2 逻辑函数表达式类型的转换

例2.7.2将下面逻辑函数化成最简或与式

$$Y = AB' + BC' + A'C$$

解：其卡诺图如表2.7.8所示  
则

$$\begin{aligned} Y &= AB' + BC' + A'C \\ &= (A + B + C)(A' + B' + C') \end{aligned}$$

表2.7.8 Y的卡诺图

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

## 2.将与或式转换成与或非式

已知逻辑函数式，先画出其卡诺图，然后圈“0”写出逻辑函数的补函数的与或式，再取反即可得到与或非式

## 2.7.2 逻辑函数表达式类型的转换

例2.7.3 将下面逻辑函数简化成最简与或非式

$$Y = ABD + AB'CD' + AC'$$

解：其卡诺图如表2.7.9所示  
圈“0”可得 $Y'$ 为

$$Y' = A' + BCD' + B'CD$$

取反即得与或非式，即

$$Y = (A' + B'CD + BCD')'$$

表2.7.9 Y的卡诺图

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

## 2.7.2 逻辑函数表达式类型的转换

### 3.将与或式转换成或非式

已知逻辑函数的与或式，先画出卡诺图，圈“0”，得到最简或与式，进行两次取反，利用摩根定理即可得到或非式

例1.5.4 将下面逻辑函数化成最简或非式

$$Y = A(B + C') + A'(B' + C) + B'C'D + BCD$$

解：

$$\begin{aligned} Y &= A(B + C') + A'(B' + C) + B'C'D + BCD \\ &= AB + AC' + A'B' + A'C + B'C'D + BCD \end{aligned}$$

## 2.7.2 逻辑函数表达式类型的转换

$$\begin{aligned} Y &= A(B + C') + A'(B' + C) + B'C'D + BCD \\ &= AB + AC' + A'B' + A'C + B'C'D + BCD \end{aligned}$$

其卡诺图如图2.7.10所示，则

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

$$\begin{aligned} Y &= (((A + B' + C)(A' + B + C'))')' \\ &= ((A + B' + C)' + (A' + B + C')')' \end{aligned}$$

表2.7.10 Y的卡诺图

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

## 2.7.2 逻辑函数表达式类型的转换

例1.5.5 将下面的逻辑函数简化成与非式、与或非式和或非式

$$Y(A, B, C, D) = \sum m(0, 1, 2, 5, 8, 9, 11, 15)$$

解：卡诺图如表2.7.11所示，  
则最简与或式为

$$Y = B'C' + ACD + A'C'D + A'B'D'$$

两次取反可得与非式为：

$$\begin{aligned} Y &= ((B'C' + ACD + A'C'D + A'B'D'))' \\ &= ((B'C')' \cdot (ACD)' \cdot (A'C'D)' \cdot (A'B'D'))' \end{aligned}$$

表2.7.11 Y的卡诺图

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

## 2.7.2 逻辑函数表达式类型的转换

表2.7.11圈“0”，可得Y的反函数的与或式为

表2.7.11 Y的卡诺图

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

表2.7.11 Y的卡诺图

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

$$Y' = ABC' + A'CD + ACD' + A'BD'$$

$$Y = (ABC' + A'CD + ACD' + A'BD')'$$



## 2.7.2 逻辑函数表达式类型的转换

$$Y = (ABC' + A'CD + ACD' + A'BD')'$$

或与式为

$$\begin{aligned} Y &= (ABC' + A'CD + ACD' + A'BD')' \\ &= (ABC')' \cdot (A'CD)' \cdot (ACD')' \cdot (A'BD')' \\ &= (A' + B' + C)(A + C' + D')(A' + C' + D)(A + B' + D) \end{aligned}$$

或非式为

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

# 作业

题2.13(5) ( 9 )

题2.14(a) ( c )

题2.16(3) ( 5 ) ( 7 )

题2.20 (3)

题2.21 (4)

题2.23 (3)