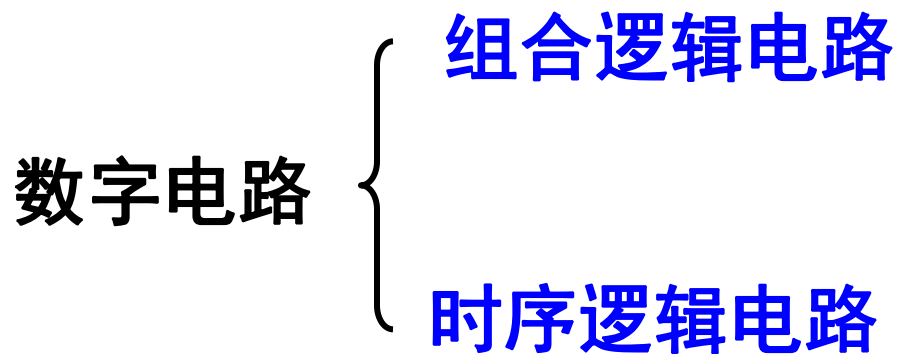


# 第3章 组合逻辑电路

---



组合逻辑电路：任一时刻的输出取决于当时的输入。

时序逻辑电路：任一时刻的输出不仅取决于当时的输入，还取决于过去的输入序列，或者过去的电路状态

**本章介绍组合逻辑电路。**

---

# 教学基本要求

---

- 1) 掌握SSI组合电路的分析方法与双轨输入条件下的设计方法。
  - 2) 了解MSI组合电路编码器、译码器、数据选择器、数据比较器、加法器的功能。
  - 3) 掌握用MSI组合电路数据选择器、数据比较器、加法器实现组合逻辑设计的方法。
  - 4) 了解组合电路中的竞争冒险现象，了解消除冒险的方法。
-

## 3.1 SSI构成的组合电路的分析和设计

---

**分析：** 确定电路实现的逻辑功能。

**设计：** 根据逻辑功能确定实现电路。

# 一、组合电路的分析

---

1. 分析目的：确定电路实现的逻辑功能

2. 分析步骤：

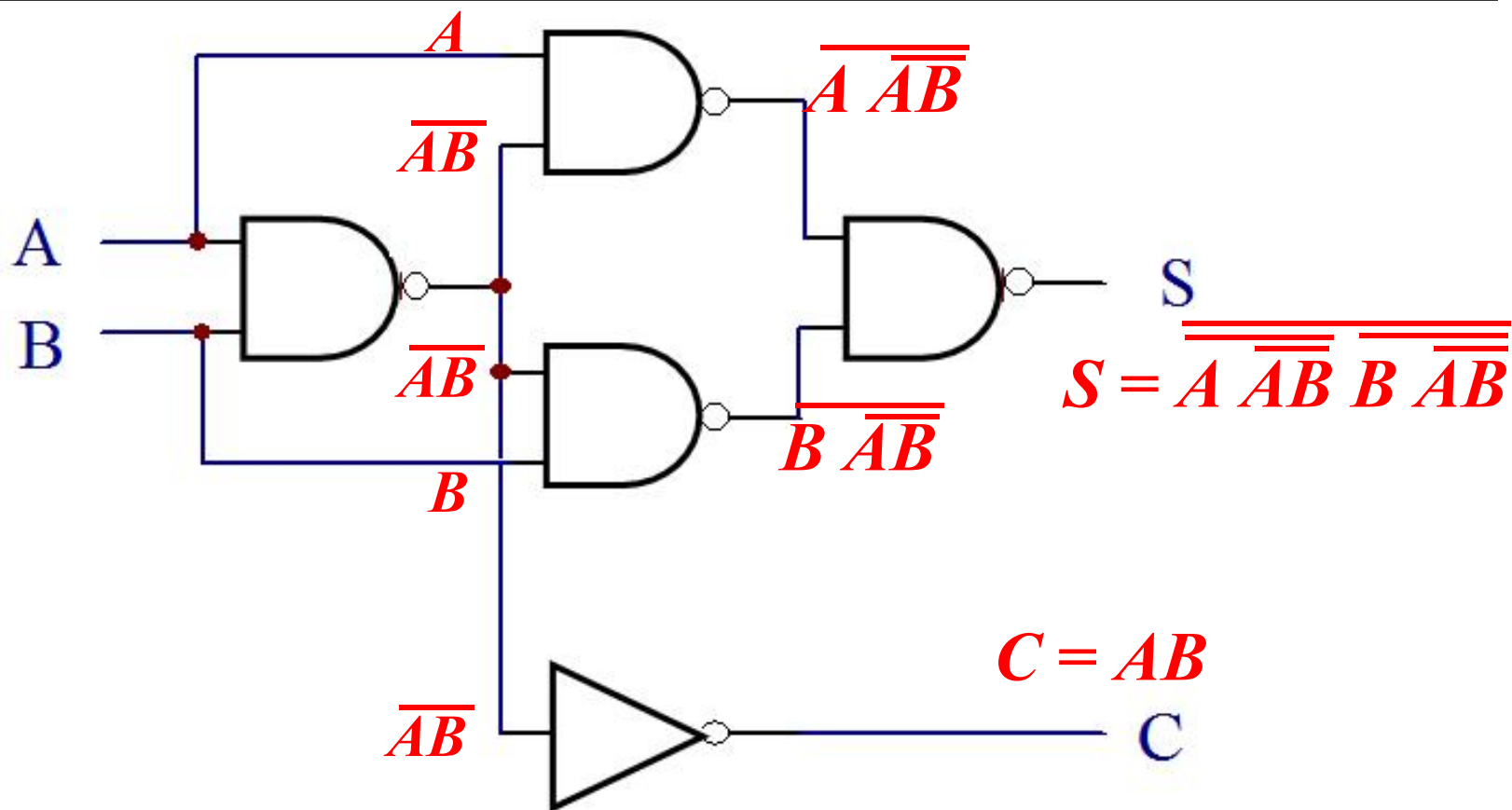
(1) 从输入端开始，逐级推导出函数表达式；

(2) 对表达式适当化简，列出真值表；

(3) 根据真值表，确定逻辑功能。

---

### 例3.1.1 分析如图所示的逻辑电路的逻辑功能。



解：(1) 写出逻辑表达式

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

$$C = \overline{\overline{AB}} = AB$$

表 4.1.1

(2) 列真值表

(3) 确定逻辑功能

$A$ 、 $B$  为一位二进制数，  
 $S$  为本位和，  
 $C$  为本位向高位的进位。

输入		输出	
$A$	$B$	$C$	$S$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

## 结论：

---

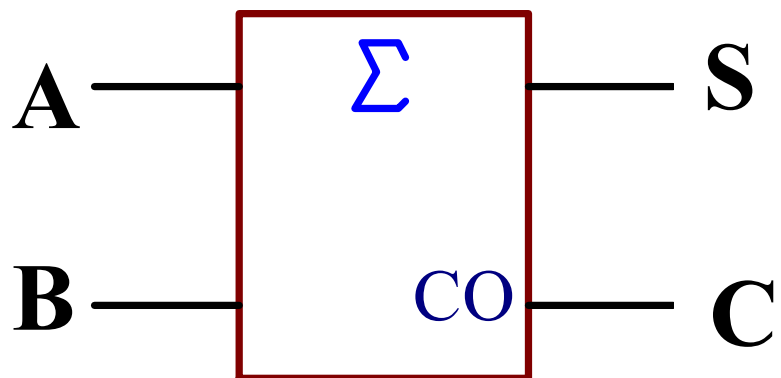
此电路完成一位二进制数的半加运算，是一个一位半加器。

## 半加器与全加器：

- 半加器仅考虑向高位产生的进位，  
不考虑来自低位的进位。
  - 全加器不仅考虑向高位产生的进位，  
而且考虑来自低位的进位。
-

半加器的逻辑符号如下图所示：

---



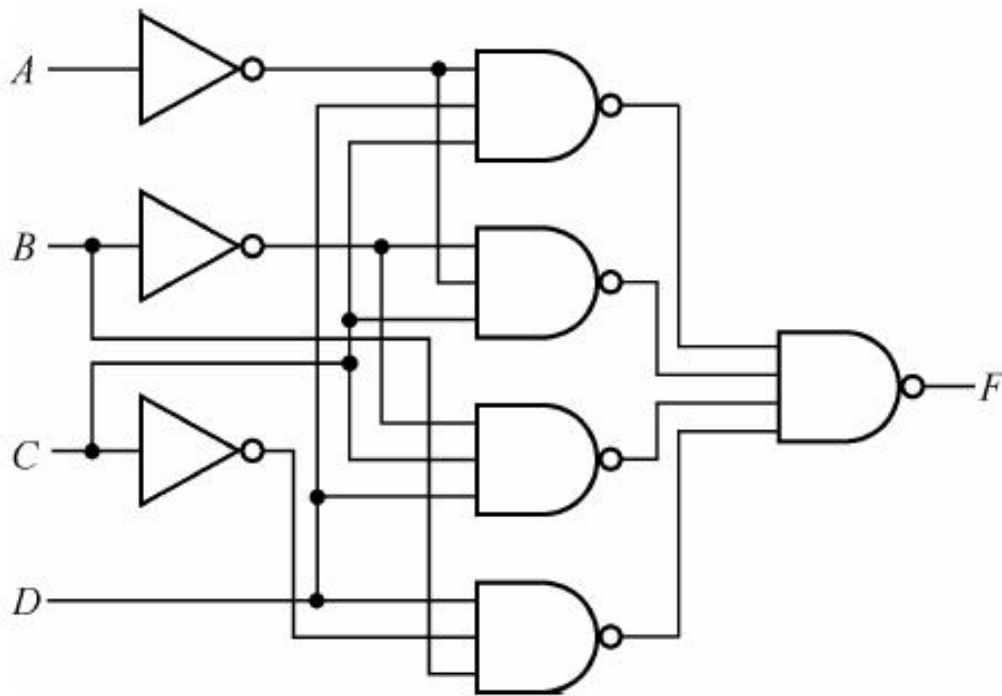
$$S = A \oplus B$$

$$C = AB$$

---



### 例3.1.2 分析如图所示的逻辑电路的逻辑功能。



解：(1) 写出逻辑表达式

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

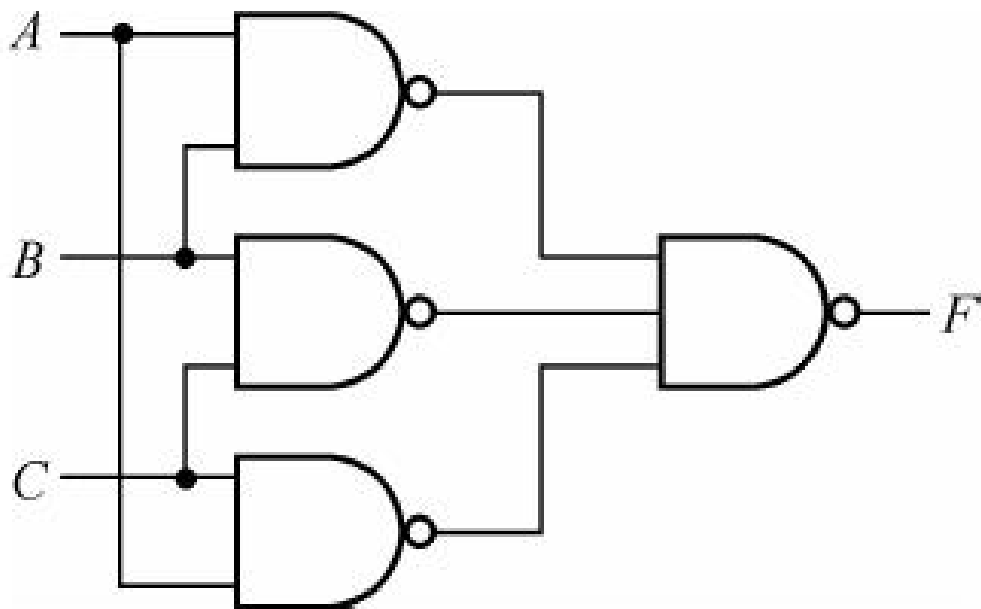
## (2) 列真值表

输入				输出
D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1

输入				输出
D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	F
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

(3) 确定逻辑功能：**得到4位二进制数中的素数。**

### 3.1.3 分析如图所示电路的逻辑功能。



$$F = \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{AC}} = AB + BC + AC$$

$$F = \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{AC}} = AB + BC + AC$$

A	B	C	F
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

**结论：**

输入A、B、C三变量中，只要有二个或二个以上变量为1，则输出为1。

**此电路为三人表决电路。**

## 3.1.2 组合逻辑电路的设计

---

**设计目的：确定满足一定逻辑功能的电路**

**双轨输入与单轨输入方式：**

**双轨输入：**信号源能提供原、反两种信号。

**单轨输入：**信号源只能提供原变量而无反变量。

---

## 双轨输入方式下的设计步骤：

---

- (1) 列真值表；
- (2) 写最简表达式；
- (3) 画逻辑电路

## 例 3.1.4 试设计一个1位全加器电路。

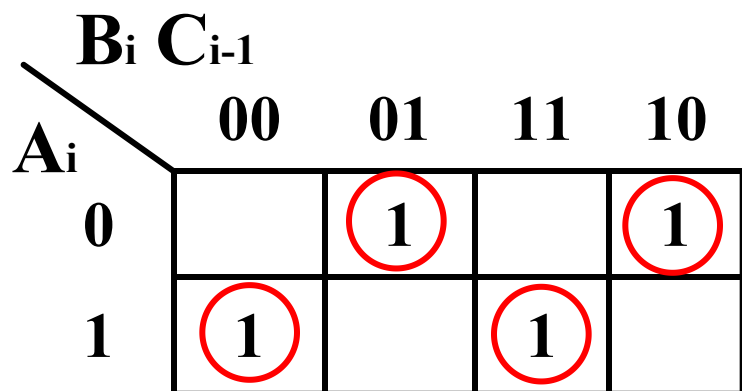
解：(1) 列真值表

表 4.1.3

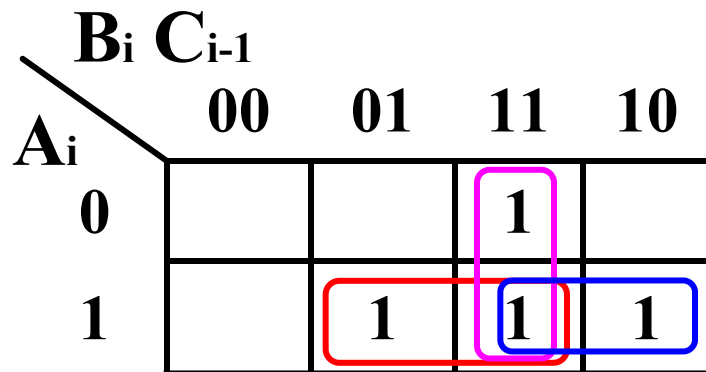
输 入			输 出	
$A_i$	$B_i$	$C_{i-1}$	$C_i$	$S_i$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0

输 入			输 出	
$A_i$	$B_i$	$C_{i-1}$	$C_i$	$S_i$
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

(2) 写最简表达式;



(a)  $S_i$  的卡诺图



(b)  $C_i$  的卡诺图

图 4.1.3

$$S_i = A_i \bar{B}_i \bar{C}_{i-1} + \bar{A}_i \bar{B}_i C_{i-1} + A_i B_i C_{i-1} + \bar{A}_i B_i \bar{C}_{i-1}$$

$$C_i = A_i B_i + B_i C_{i-1} + A_i C_{i-1}$$



变换 $S_i$ 、 $C_i$ ，可得：

---

$$\begin{aligned} S_i &= A_i \overline{B_i} \overline{C_{i-1}} + A_i \overline{B_i} C_{i-1} + A_i B_i \overline{C_{i-1}} + A_i B_i C_{i-1} \\ &= A_i (\overline{B_i} \overline{C_{i-1}} + \overline{B_i} C_{i-1}) + \overline{A_i} (\overline{B_i} C_{i-1} + B_i \overline{C_{i-1}}) \\ &= A_i \overline{B_i \oplus C_{i-1}} + \overline{A_i} (B_i \oplus C_{i-1}) \\ &= A_i \oplus B_i \oplus C_{i-1} \end{aligned}$$

$$\begin{aligned} C_i &= A_i B_i + B_i C_{i-1} + A_i C_{i-1} \\ &= A_i B_i + (A_i + \overline{A_i}) B_i C_{i-1} + (B_i + \overline{B_i}) A_i C_{i-1} \\ &= \underline{A_i B_i} + \underline{A_i B_i C_{i-1}} + \underline{\overline{A_i} B_i C_{i-1}} + \underline{A_i B_i C_{i-1}} + \underline{A_i \overline{B_i} C_{i-1}} \\ &= A_i B_i + C_{i-1} (A_i \oplus B_i) \end{aligned}$$

---

(3) 画逻辑电路，如下图 (a) 所示。

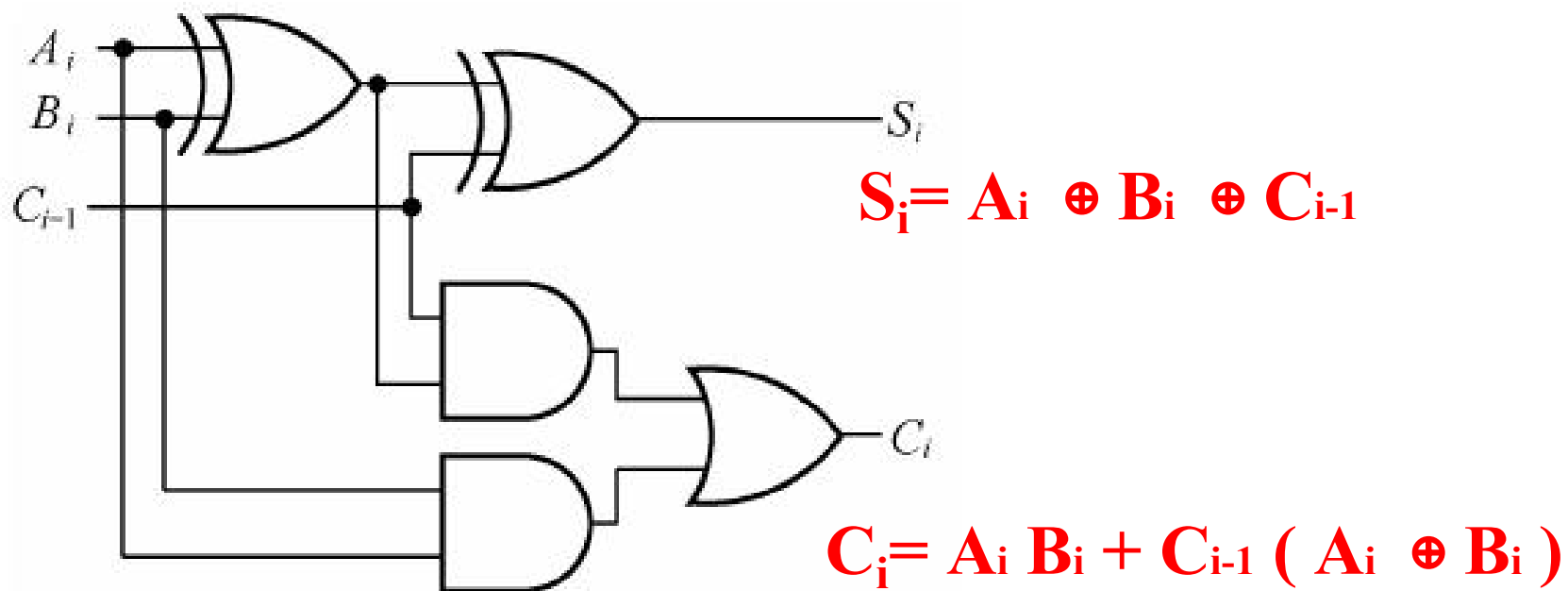


图 3.1.15 (a) 全加器电路

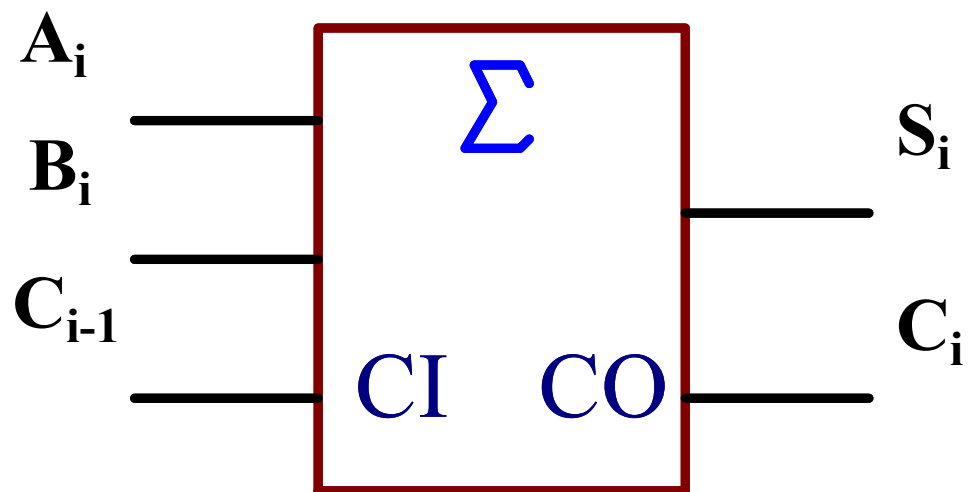


图 3.1.15 (b) 全加器逻辑符号

### 例 3.1.5 试设计一个1位二进制数比较单元。

解：(1) 列真值表

输 入		输 出		
A	B	$F_{A<B}$	$F_{A=B}$	$F_{A>B}$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

(2) 写最简表达式；

$$F_{A>B} = \overline{A}B$$

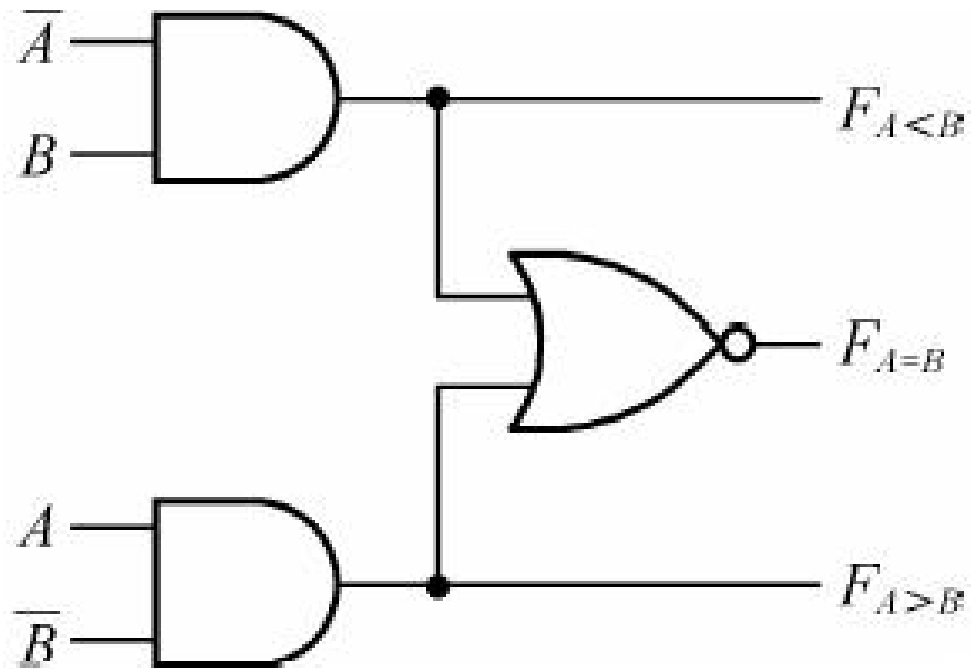
$$F_{A<B} = \overline{A}\overline{B}$$

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

$$= \overline{F_{A<B} + F_{A>B}}$$

### (3) 画逻辑电路

---



---

如何用门电路来实现所设计出的函数，是值得探讨的问题。如果考虑实现成本，则实现函数必须使电路最简单，也就是门电路的个数和连线都最少。如果从产品的设计速度来说，就要以设计过程、方法简单为主要目标。

**重点掌握用最少与非门实现电路。**

---

补充：用最少的与非门实现函数： $F = AB + \overline{BC}$

---

解：由于函数已是最简与或式，直接将F两次取反，并使用一次摩根定律。

$$F = \overline{\overline{AB + \overline{BC}}} = \overline{\overline{AB}} \cdot \overline{\overline{\overline{BC}}}$$

画逻辑电路图。（略）

---

**补充：用最少的与非门实现以下函数：**

---

$$F(A,B,C,D)=\sum m(4,5,6,7,8,9,10,11,12,13)$$



**举例:** 要求设计一个逻辑电路, 能够判断一位十进制数是奇数还是偶数, 当十进制数为奇数时, 电路输出为1, 当十进制数为偶数时, 电路输出为0。

		<b>CD</b>			
		00	01	11	10
<b>AB</b>	00	0	1	1	0
	01	0	1	1	0
	11	$\Phi$	$\Phi$	$\Phi$	$\Phi$
	10	0	1	$\Phi$	$\Phi$

$$F = D$$

<b>ABCD</b>	<b>L</b>
0000	0
0001	1
0010	0
0011	1
0100	0
0101	1
0110	0
0111	1
1000	0
1001	1
1010	×
1011	×
1100	×
1101	×
1110	×
1111	×

## 3.2 中规模集成组合逻辑电路 (MSI)

---

着重掌握其逻辑功能和使用方法。

- 编码器
  - 译码器
  - 数据选择器
  - 数值比较器
  - 全加器
  - 奇偶校验器
-

# 一、编码器

---

---

**编码：**在数字技术中，通常用二进制数码0和1构成的一组有序组合（称为代码）来表示各种对象（如十进制数、字符等）。这一指定过程，称为**编码**。

---

---

# 1. 二进制编码器

---

$2^n$ 个互不相同的状态  $\rightarrow$   $2^n$ 个代码 (共需 $n$ 位码元)

---

# 3位二进制编码器功能表

输 入								输 出		
I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	I <sub>7</sub>	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$A = I_4 + I_5 + I_6 + I_7$$

$$B = I_2 + I_3 + I_6 + I_7$$

$$C = I_1 + I_3 + I_5 + I_7$$

某一个输入端为高电平时，编码器将其编码为该线十进制下标的二进制代码。

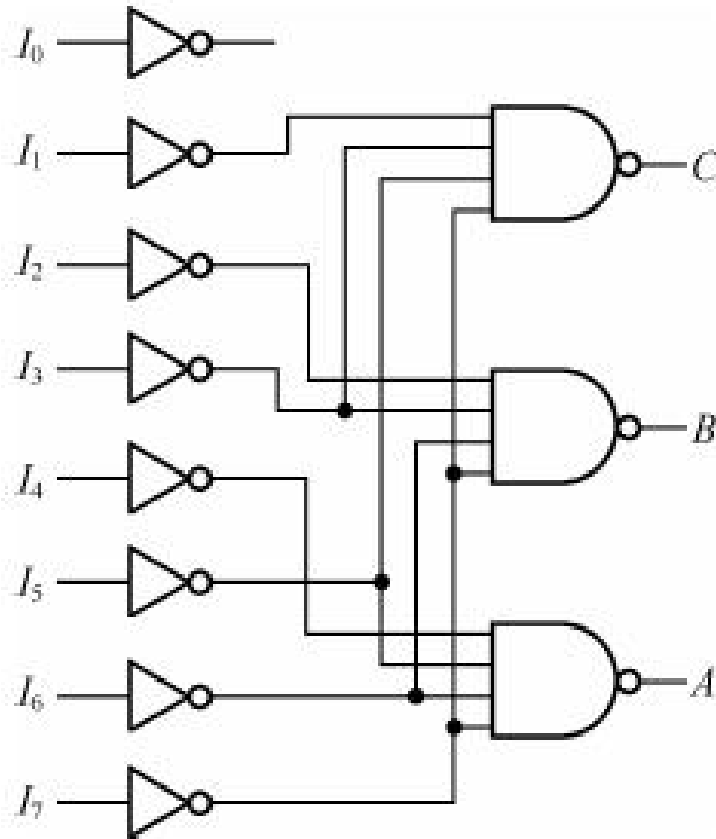
$$A = I_4 + I_5 + I_6 + I_7 = \overline{\overline{I_4 \cdot I_5 \cdot I_6 \cdot I_7}}$$

---

~~$$B = I_2 + I_3 + I_6 + I_7 = \overline{\overline{I_2 \cdot I_3 \cdot I_6 \cdot I_7}}$$~~

---

$$C = I_1 + I_3 + I_5 + I_7 = \overline{\overline{I_1 \cdot I_3 \cdot I_5 \cdot I_7}}$$



## 存在的问题：

---

要求输入信号互斥，  
即在任一时刻仅有一根输入线提出编码请求。  
否则，编码器产生逻辑错误。

---

---

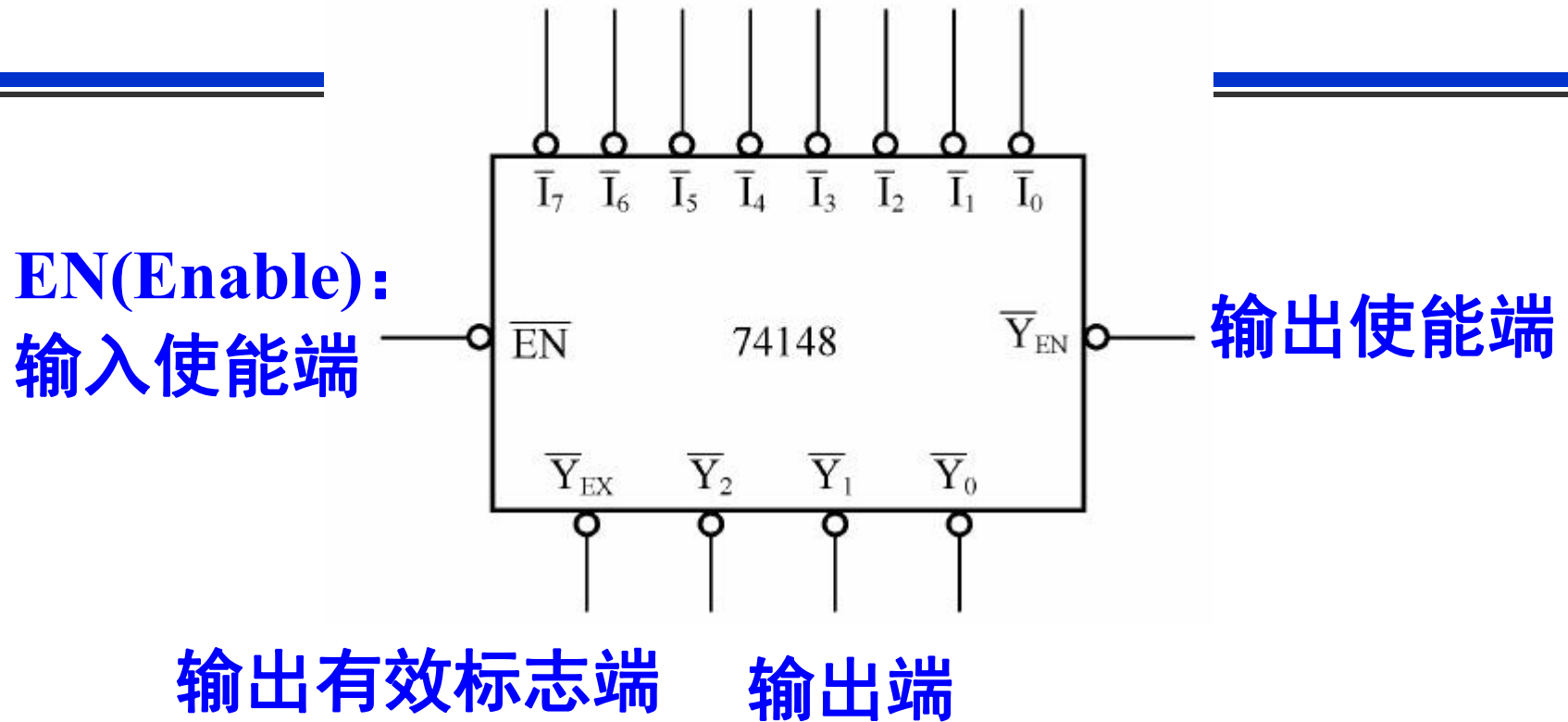
## 优先编码器：

不要求输入信号互斥，可对其中优先权最高的信号编码。

---



## 输入端



## 74148简化逻辑符号

74148各输入端、输出端都是低电平有效。

# 表4.2.2 8—3线优先编码器74148功能表

使能输入	输 入	输 出	输出标志	使能输出
$\overline{EN}$	$\overline{I}_7 \ \overline{I}_6 \ \overline{I}_5 \ \overline{I}_4 \ \overline{I}_3 \ \overline{I}_2 \ \overline{I}_1 \ \overline{I}_0$	$\overline{Y}_2 \ \overline{Y}_1 \ \overline{Y}_0$	$\overline{Y}_{EX}$	$\overline{Y}_{EN}$
1	∅ ∅ ∅ ∅ ∅ ∅ ∅ ∅	1 1 1	1	1
0	1 1 1 1 1 1 1 1	1 1 1	1	0
0	0 ∅ ∅ ∅ ∅ ∅ ∅ ∅	0 0 0	0	1
0	1 0 ∅ ∅ ∅ ∅ ∅ ∅	0 0 1	0	1
0	1 1 0 ∅ ∅ ∅ ∅ ∅	0 1 0	0	1
0	1 1 1 0 ∅ ∅ ∅ ∅	0 1 1	0	1
0	1 1 1 1 0 ∅ ∅ ∅	1 0 0	0	1
0	1 1 1 1 1 0 ∅ ∅	1 0 1	0	1
0	1 1 1 1 1 1 0 ∅	1 1 0	0	1
0	1 1 1 1 1 1 1 0	1 1 1	0	1

① 产生输入端十进制下标的自然二进制码的反码

② 输入端低电平（即逻辑“0”）有效

8个输入线中按顺序， $\overline{I}_7$  优先权最高， $\overline{I}_0$  优先权最低。

**输出使能端和输入使能端**配合使用，可以实现电路扩展。

74148的级联（不讲）

## 2. 二—十进制优先编码器74147

---

功能：

将  $\overline{I_1} - \overline{I_9}$  9个输入信号分别编成9个BCD码。

---

# 表4.2.3 二—十进制优先编码器74147功能表

输 入									输 出			
$\bar{I}_1$	$\bar{I}_2$	$\bar{I}_3$	$\bar{I}_4$	$\bar{I}_5$	$\bar{I}_6$	$\bar{I}_7$	$\bar{I}_8$	$\bar{I}_9$	$\bar{Y}_3$	$\bar{Y}_2$	$\bar{Y}_1$	$\bar{Y}_0$
0	1	1	1	1	1	1	1	1	1	1	1	0
∅	0	1	1	1	1	1	1	1	1	1	0	1
∅	∅	0	1	1	1	1	1	1	1	1	0	0
∅	∅	∅	0	1	1	1	1	1	1	1	0	1
∅	∅	∅	∅	0	1	1	1	1	1	1	0	0
∅	∅	∅	∅	∅	0	1	1	1	1	1	0	0
∅	∅	∅	∅	∅	∅	0	1	1	1	1	0	0
∅	∅	∅	∅	∅	∅	∅	0	1	1	1	1	0
∅	∅	∅	∅	∅	∅	∅	∅	0	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1

① 产生输入端十进制下标的8421BCD码的反码

② 输入端低电平（即逻辑“0”）有效

## 4.2.2 译码器

---

**译码：**编码的逆过程，将二进制代码或BCD码译成控制电平。

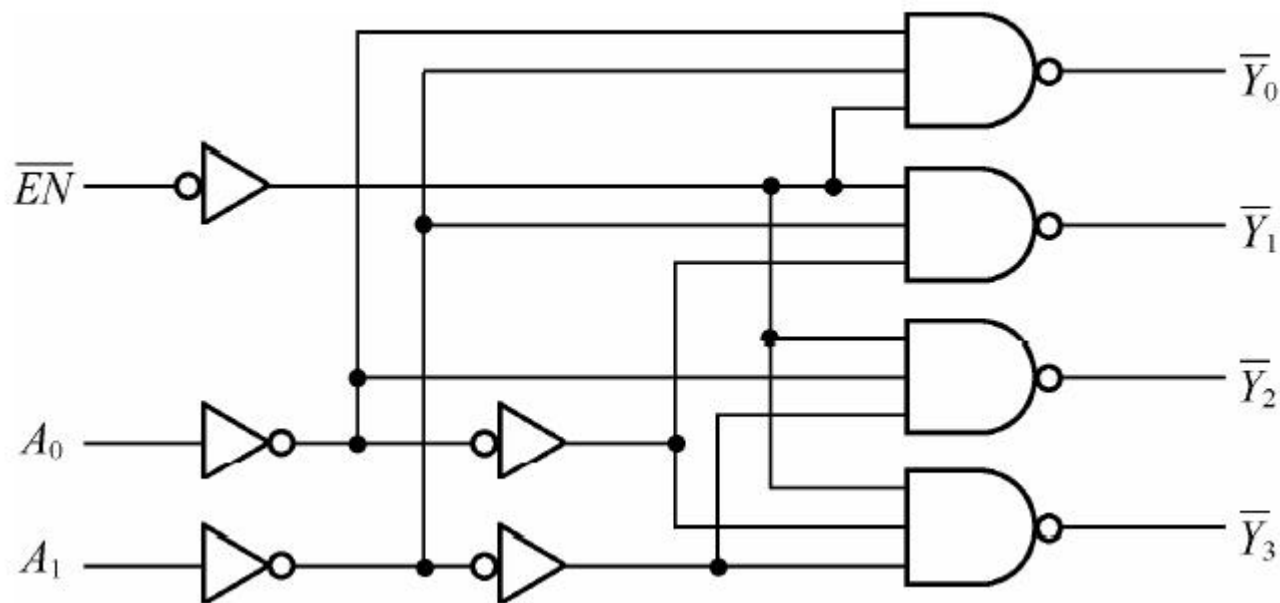
### 1. 二进制译码器（变量译码器）

$2^n$ 个  $n$ 位二进制代码  $\longrightarrow$   $2^n$ 个互不相同的状态

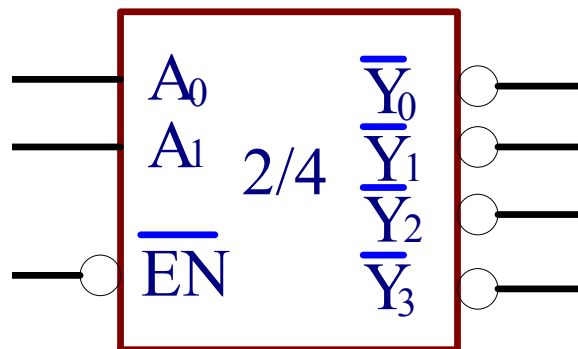
常用的有：

- 2-4线译码器
  - 3-8线译码器
  - 4-16线译码器
-

# (1) 2-4 线译码器



(a) 逻辑图



(b) 简化符号

表 4.2.4 2-4 线译码器的功能表

使能 输入	输 入		输 出			
$\overline{EN}$	$A_1$	$A_0$	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$
1	0	0	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

$\overline{EN}=0$  时，器件工作，算出输入的一组二进制代码对应的十进制数，以此数作为下标的输出端输出0。



## 2—4线译码器的逻辑表达式

$\overline{EN}=1, \overline{Y}_i=1 (i=0,1,2,3)$  ; 器件不工作

$\overline{EN}=0, \overline{Y}_i=\overline{m}_i (i=0,1,2,3)$  ; 器件工作

使能输入	输 入		输 出			
$\overline{EN}$	$A_1$	$A_0$	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$
1	$\emptyset$	$\emptyset$	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

# 双2-4线译码器74139：

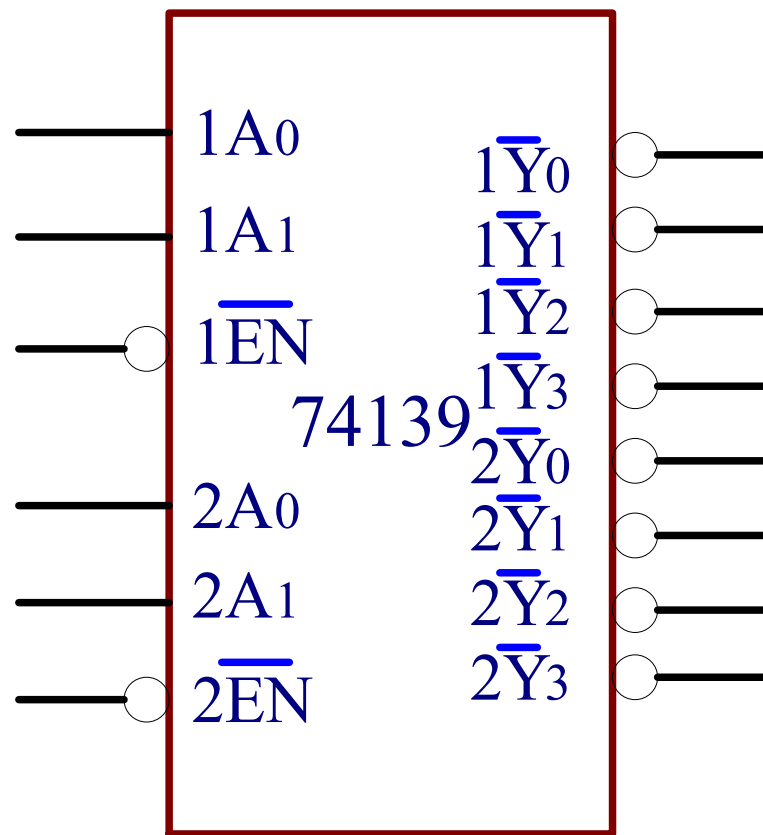


图 4.2.6 74139的简化逻辑符号

## (2) 3—8 线译码器

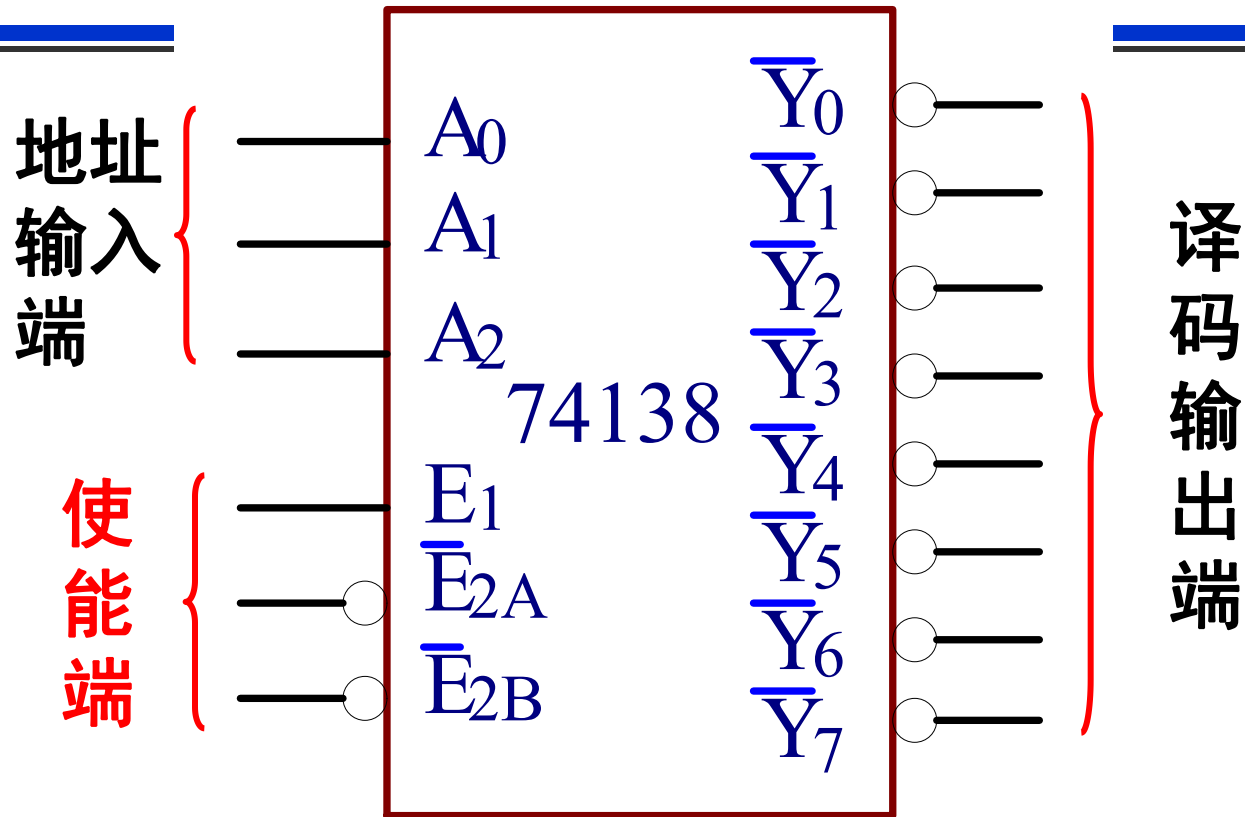


图 4.2.7 (a) 简化符号

$E_1=1$ 和 $\bar{E}_{2A}+\bar{E}_{2B}=0$ 同时满足，器件才正常工作。



## 3—8线译码器74138的逻辑表达式：

---

$$\left\{ \begin{array}{l} E_1=1 \text{ 和 } \bar{E}_{2A} + \bar{E}_{2B} = 0 \text{ 不同时满足,} \\ \bar{Y}_i = 1 \quad (i = 0, 1, \dots, 7) \quad ; \text{ 器件不工作} \\ \\ E_1=1 \text{ 和 } \bar{E}_{2A} + \bar{E}_{2B} = 0 \text{ 同时满足,} \\ \bar{Y}_i = \bar{m}_i \quad (i = 0, 1, \dots, 7) \quad ; \text{ 器件工作} \end{array} \right.$$

---

### (3) 使能端的使用：

利用使能端不仅可以控制译码器是否工作，且可以作为**逻辑扩展和功能扩展**之用。

• 两个2-4线译码器扩展为3-8线译码器：  
(74139的扩展)

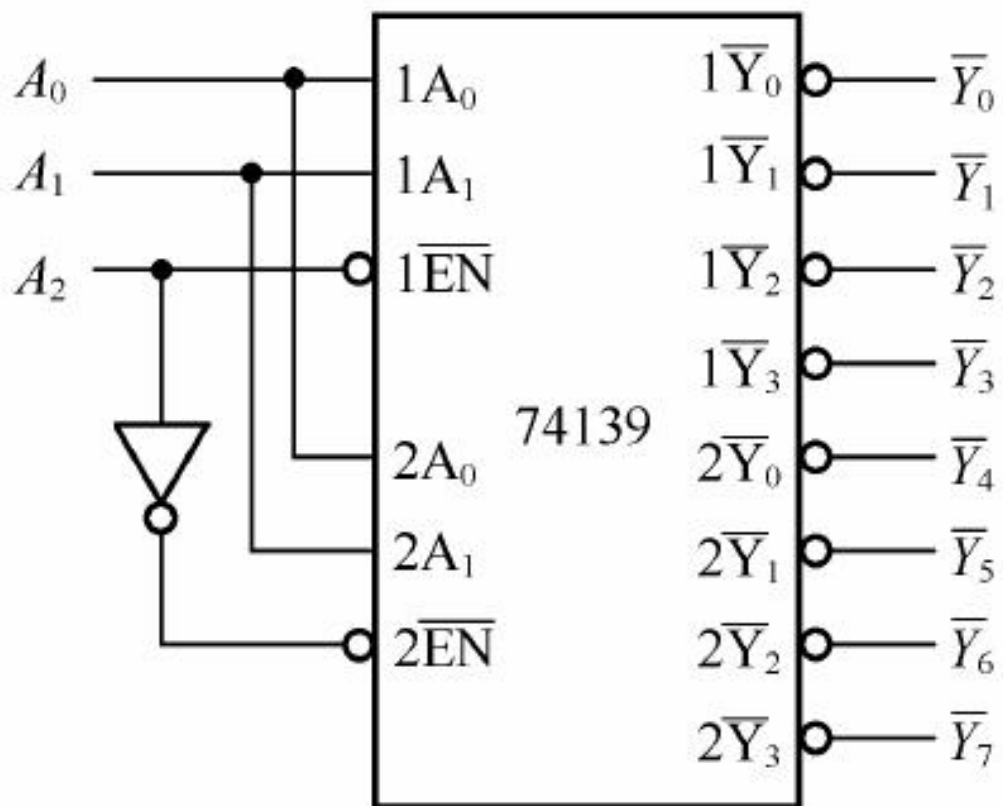


图 4.2.8 (a) 74139的扩展

# 两片芯片的工作状态：

---

$A_2$	$A_1$	$A_0$	$\overline{1EN}$	$\overline{2EN}$	I	II
0	$\varphi$	$\varphi$	0	1	工作	禁止
1	$\varphi$	$\varphi$	1	0	禁止	工作

$$\overline{1EN} = A_2$$

$$\overline{2EN} = \overline{A_2}$$

---



表 4.2.6 2—4 线译码器扩展为 3—8 线译码器的真值表

输 入	使能输入	输 出							
$A_2 A_1 A_0$	$1\overline{EN} \ 2\overline{EN}$	$1\overline{Y}_0$	$1\overline{Y}_1$	$1\overline{Y}_2$	$1\overline{Y}_3$	$2\overline{Y}_0$	$2\overline{Y}_1$	$2\overline{Y}_2$	$2\overline{Y}_3$
0 0 0	0 1	0	1	1	1	1	1	1	1
0 0 1	0 1	1	0	1	1	1	1	1	1
0 1 0	0 1	1	1	0	1	1	1	1	1
0 1 1	0 1	1	1	1	0	1	1	1	1
1 0 0	1 0	1	1	1	1	0	1	1	1
1 0 1	1 0	1	1	1	1	1	0	1	1
1 1 0	1 0	1	1	1	1	1	1	0	1
1 1 1	1 0	1	1	1	1	1	1	1	0
$A_2 A_1 A_0$		$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$

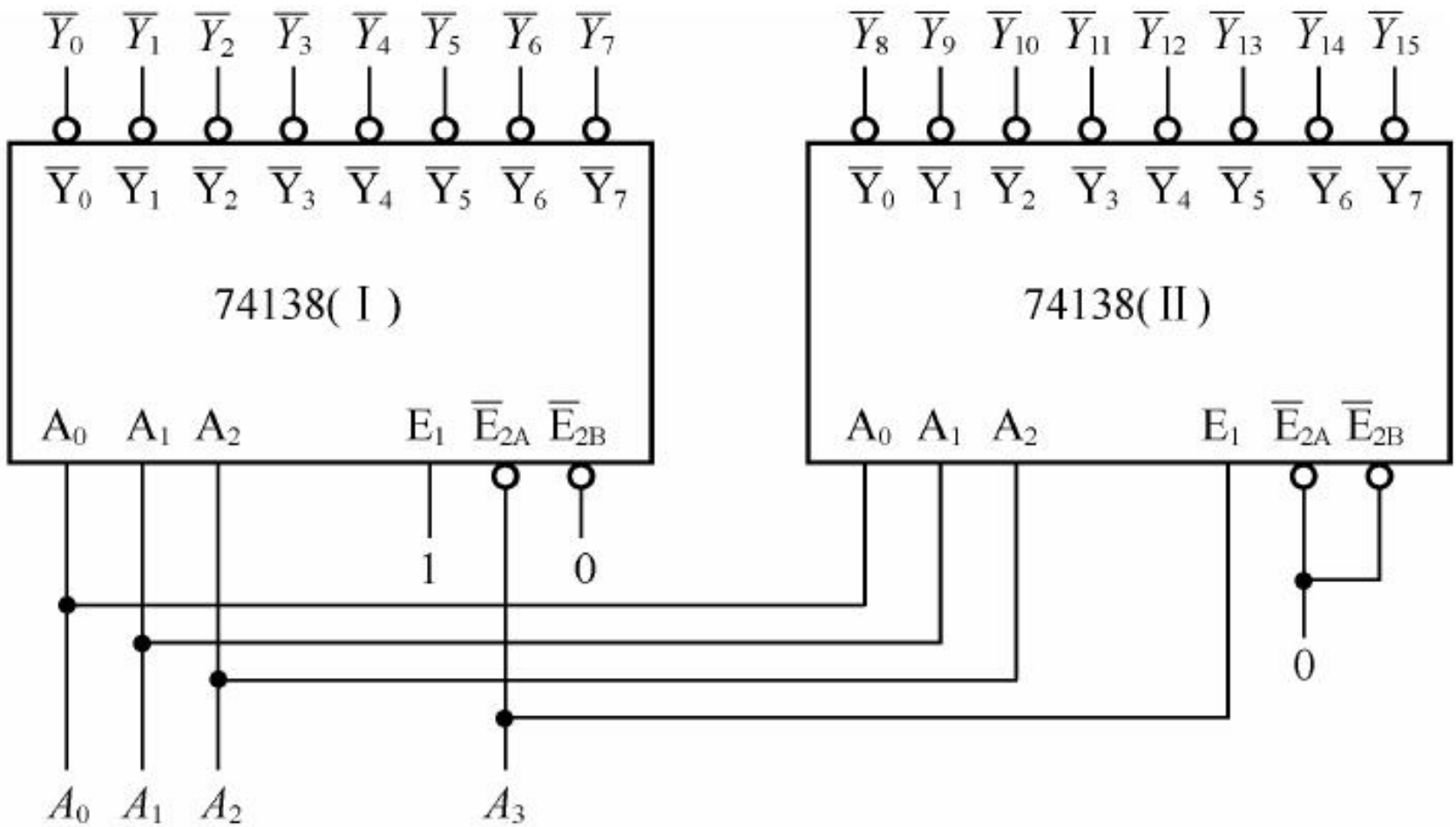


图 3.2.9 (b) 两片3-8线译码器扩展为4-16线译码器

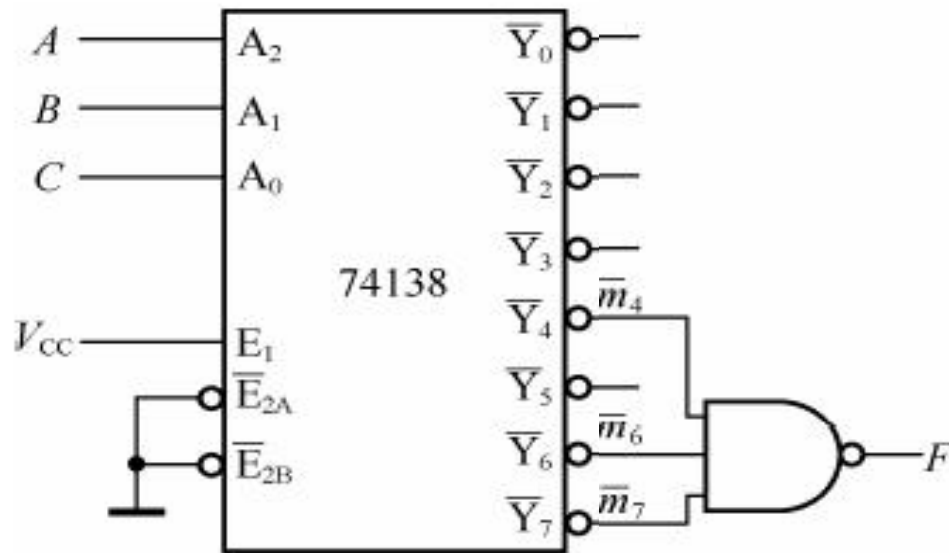
## (4) 用译码器设计组合逻辑电路

原理：译码器每个输出端分别与某一个最小项  
非（低电平译码）相对应。

例1 用74138实现函数  $F = \overline{AB} + AC$  。

$$\begin{aligned}
 \text{解： } F(A,B,C) &= \overline{AB} + AC = m_4 + m_6 + m_7 \\
 &= \overline{\overline{m_4 + m_6 + m_7}} = \overline{m_4 \cdot m_6 \cdot m_7} \\
 &= \overline{Y_4 \cdot Y_6 \cdot Y_7}
 \end{aligned}$$

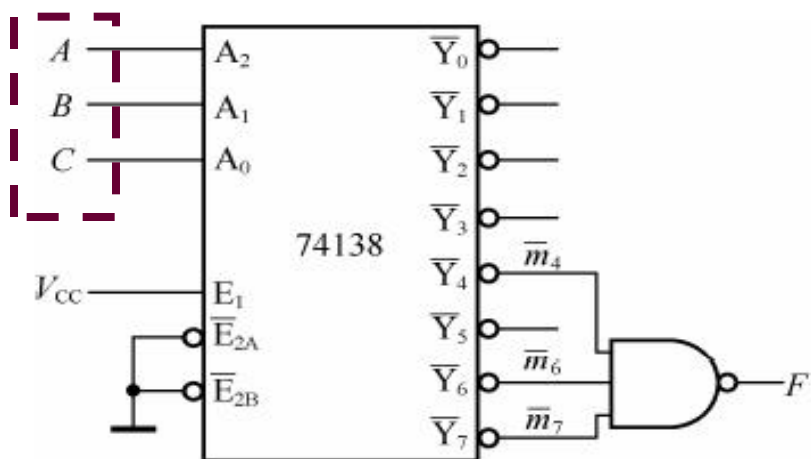
$$\begin{aligned}
 \text{解: } F(A,B,C) &= \overline{AB} + \overline{AC} = \overline{m_4} + \overline{m_6} + \overline{m_7} \\
 &= \overline{m_4 + m_6 + m_7} = \overline{m_4} \cdot \overline{m_6} \cdot \overline{m_7} \\
 &= \overline{Y_4} \cdot \overline{Y_6} \cdot \overline{Y_7}
 \end{aligned}$$



(a)

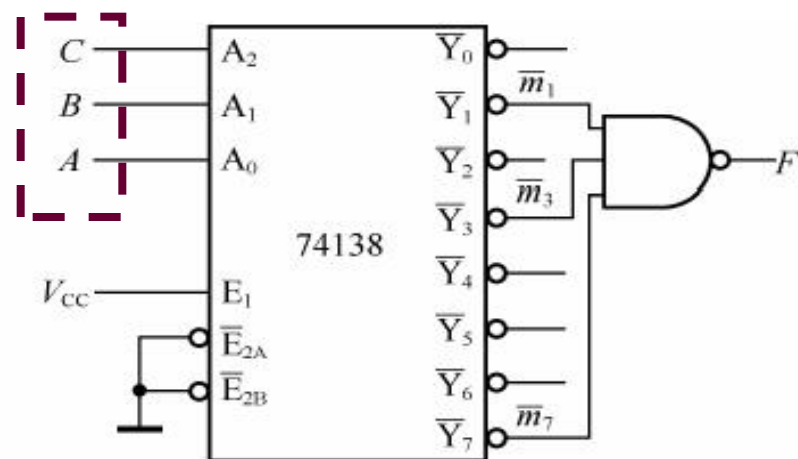
图 3.2.11 (a)

$$\begin{aligned}
 \text{另, } F(C,B,A) &= AB + \bar{A}C = m_1 + m_3 + m_7 \\
 &= \overline{\overline{m_1 + m_3 + m_7}} \\
 &= \overline{Y_1 \cdot Y_3 \cdot Y_7}
 \end{aligned}$$



(a)

图 3.2.11 (a)



(b)

图 3.2.11 (b)

例2 用74138设计一个多输出组合网络，它的输入为A、B、C三个变量，输出为下面三个函数。

$$\begin{cases} F_1 = \overline{A}C + BC \\ F_2 = A + C \\ F_3 = A + B + C \end{cases}$$

$$\begin{aligned} \text{解: } F_1(A,B,C) &= \overline{A}C + BC = m_1 + m_5 + m_7 \\ &= \underline{m_1} \cdot \underline{m_5} \cdot \underline{m_7} \end{aligned}$$

$$F_2(A,B,C) = A + C = m_0 + m_2 + m_4 + m_5 + m_6 + m_7$$

$$= m_0 + m_2 + m_4 + m_5 + m_6 + m_7$$

$$= m_1 + m_3 = \overline{m_1} \cdot m_3 = \overline{Y_1} \cdot \overline{Y_3}$$

$$F_3(A,B,C) = A+B+C = \overline{A} \overline{B} \overline{C} = m_0 = Y_0$$

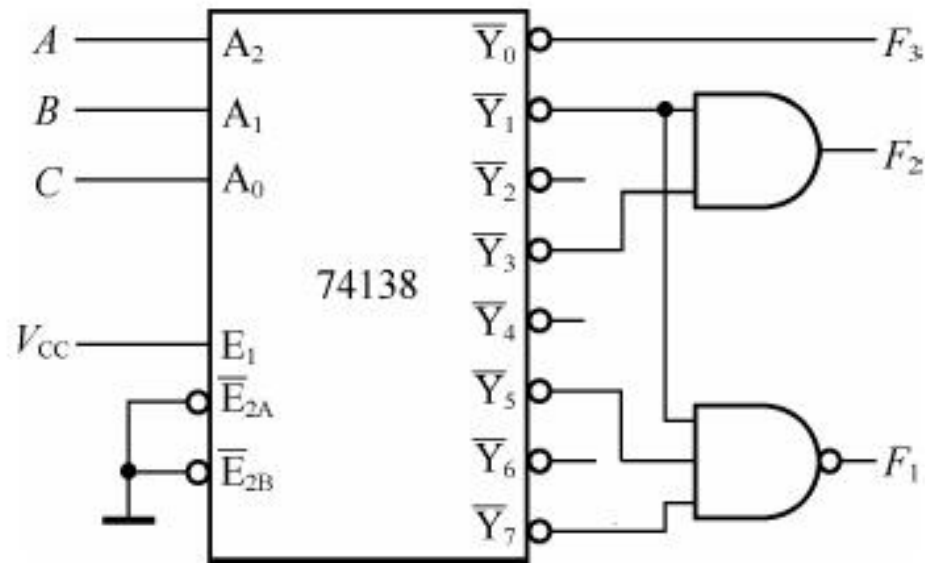


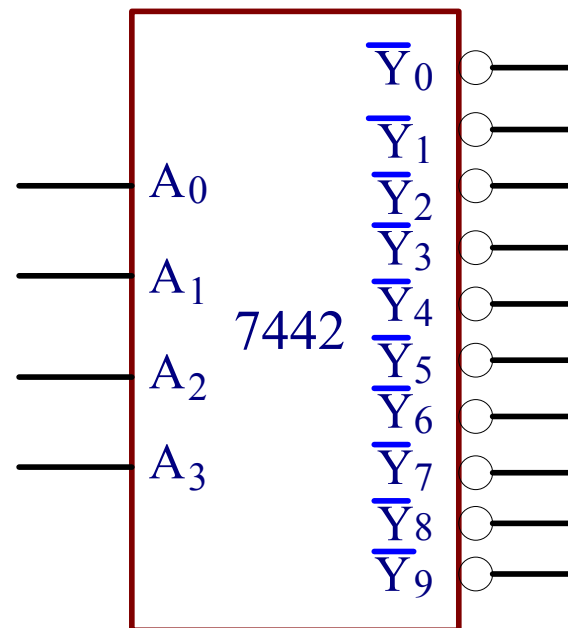
图 3.2.12

## 2.二—十进制译码器

8421BCD码译码器**7442**

余3BCD码译码器**7443**

余3格雷BCD码译码器**7444**



$A_3 \sim A_0$ ：地址输入端；

$\overline{Y}_9 \sim \overline{Y}_0$ ：输出端。

图 3.2.12 (c)



表 3.2.7 二—十进制译码器 7442的功能表

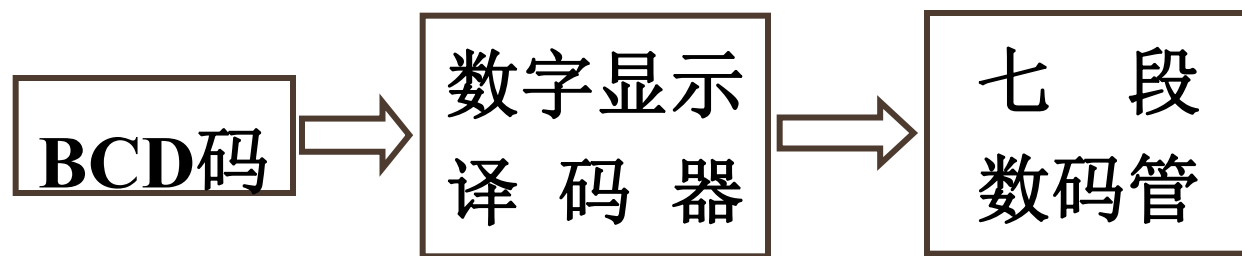
输 入				输 出									
$A_3$	$A_2$	$A_1$	$A_0$	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$	$\overline{Y}_8$	$\overline{Y}_9$
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1



## 续表

输 入				输 出									
$A_3$	$A_2$	$A_1$	$A_0$	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$	$\overline{Y_4}$	$\overline{Y_5}$	$\overline{Y_6}$	$\overline{Y_7}$	$\overline{Y_8}$	$\overline{Y_9}$
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

### 3. 显示译码器



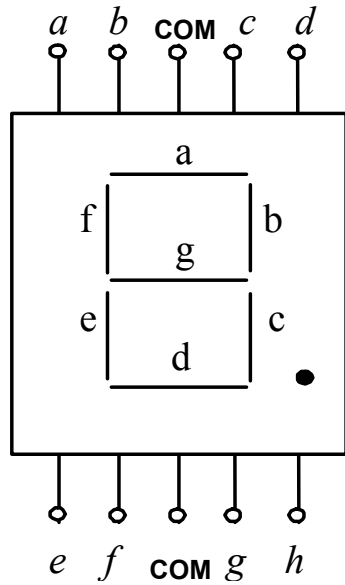
#### (1) 七段数码管

半导体  
数码管

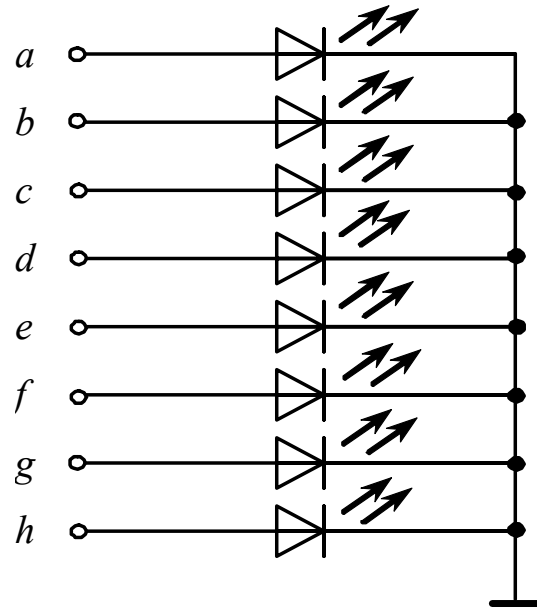
共阳极

共阴极

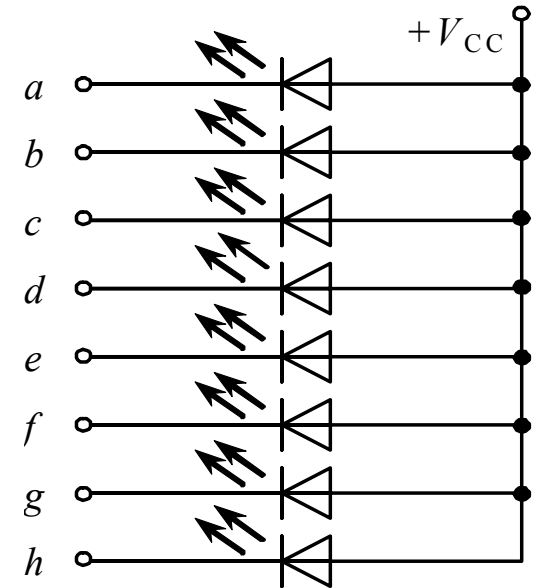
液 晶  
数码管



(a) 外形图



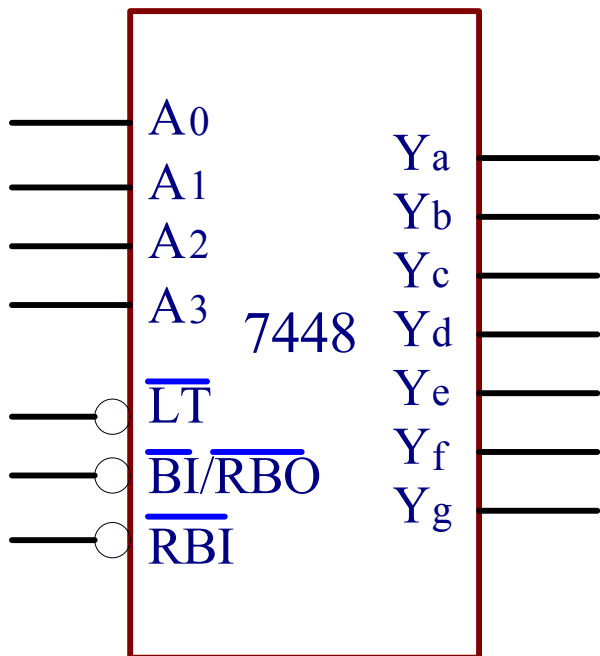
(b) 共阴极



(c) 共阳极

图 3.2.14 C—391E

## (2) 数字显示译码器7448



简化符号

$A_3 \sim A_0$ : 8421BCD

输入端

$Y_a \sim Y_g$ : 七段输出端

$\overline{LT}$ : 灯光测试输入端

$\overline{BI} / \overline{RBO}$ : 熄灭输入/  
串行灭零输

出端

$\overline{RBI}$ : 串行灭零输入端

## 辅助控制端功能：

### ① 熄灭输入 $\overline{BI}$ 。

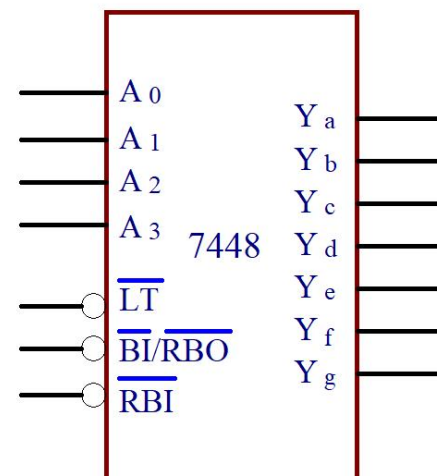
如果  $\overline{BI}=0$  时，数码管七段全灭。

### ② 灯光测试输入端 $\overline{LT}$ ：

当  $\overline{BI}=1, \overline{LT}=0$  时，数码管七段全亮，用于测试数码管的好坏。

### ③ 灭零输入端 $\overline{RBI}$ 、灭零输出端 $\overline{RBO}$ ：

当  $\overline{LT}=1, \overline{RBI}=0$  时，如果译码输入为零，则零字被熄灭，且  $\overline{RBO}=0$ 。当输入不为 0 时，该位正常显示，且  $\overline{RBO}=1$ 。用于消隐无效的 0。



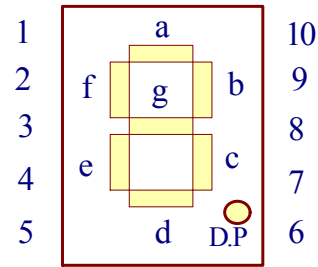


表 3.2.7 7448功能表

十进制 数 或 功 能	输 入						$\overline{\text{BI}} / \overline{\text{RBO}}$	输 出							说明
	$\overline{\text{LT}}$	$\overline{\text{RBI}}$	$A_3$	$A_2$	$A_1$	$A_0$		$Y_a$	$Y_b$	$Y_c$	$Y_d$	$Y_e$	$Y_f$	$Y_g$	
0	1	1	0	0	0	0	1	1	1	1	1	1	0	译  码  显	
1	1	0	0	0	0	1	1	0	1	1	0	0	0		
2	1	0	0	0	1	0	1	1	1	0	1	1	0		
3	1	0	0	0	1	1	1	1	1	1	0	0	1		
4	1	0	0	1	0	0	1	0	1	1	0	0	1		
5	1	0	0	1	0	1	1	1	0	1	1	0	1		
6	1	0	0	1	1	0	1	0	0	1	1	1	1		
7	1	0	0	1	1	1	1	1	1	1	0	0	0		
8	1	0	1	0	0	0	1	1	1	1	1	1	1		
9	1	0	1	0	0	1	1	1	1	1	0	0	1		

续表



图 3.2.16 0~15 十六个字符显示

十进制 数 或 功 能	输 入				$\overline{\text{BI}} / \overline{\text{RBO}}$	输 出							说明 译	
	$\overline{\text{LT}}$	$\overline{\text{RBI}}$	$A_3$	$A_2$		$A_1$	$A_0$	$Y_a$	$Y_b$	$Y_c$	$Y_d$	$Y_e$		$Y_f$
10	1	$\emptyset$	1	0	1	0	0	0	1	1	0	1	码 显	
11	1	$\emptyset$	1	0	1	1	0	0	1	1	0	0		1
12	1	$\emptyset$	1	1	0	0	0	1	0	0	0	1		1
13	1	$\emptyset$	1	1	0	1	1	0	0	1	0	1		1
14	1	$\emptyset$	1	1	1	0	0	0	1	1	1	1		1
15	1	$\emptyset$	1	1	1	1	0	0	0	0	0	0		0
$\overline{\text{BI}}=0$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	0	0	0	0	0	0	0	熄灭
$\overline{\text{LT}}=0$	0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	1	1	1	1	1	1	测试
$\overline{\text{RBI}}=0$	1	0	0	0	0	0	0	0	0	0	0	0	0	灭零



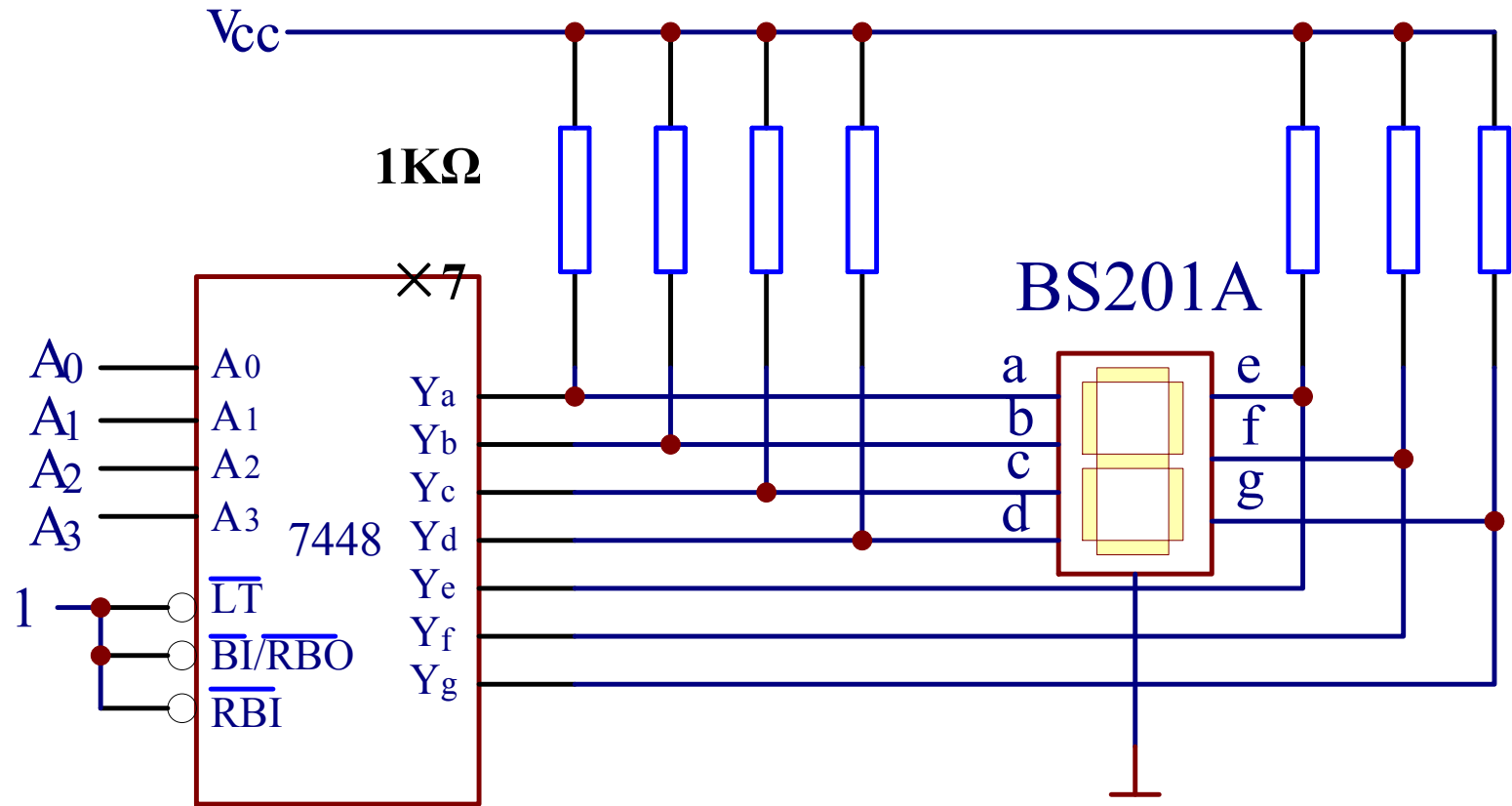


图 3.2.18 用7448驱动BS201A的连接方法

$\overline{\text{RBI}}=0$ 时，灭零，且 $\overline{\text{RBO}}=0$ 。当输入不为0时，正常显示，且 $\overline{\text{RBO}}=1$ 。

灭零输入端 $\overline{\text{RBI}}$ 和灭零输出端 $\overline{\text{RBO}}$ 配合使用，实现多位十进制

数码显示系统的**整数前和小数后的**灭零控制。

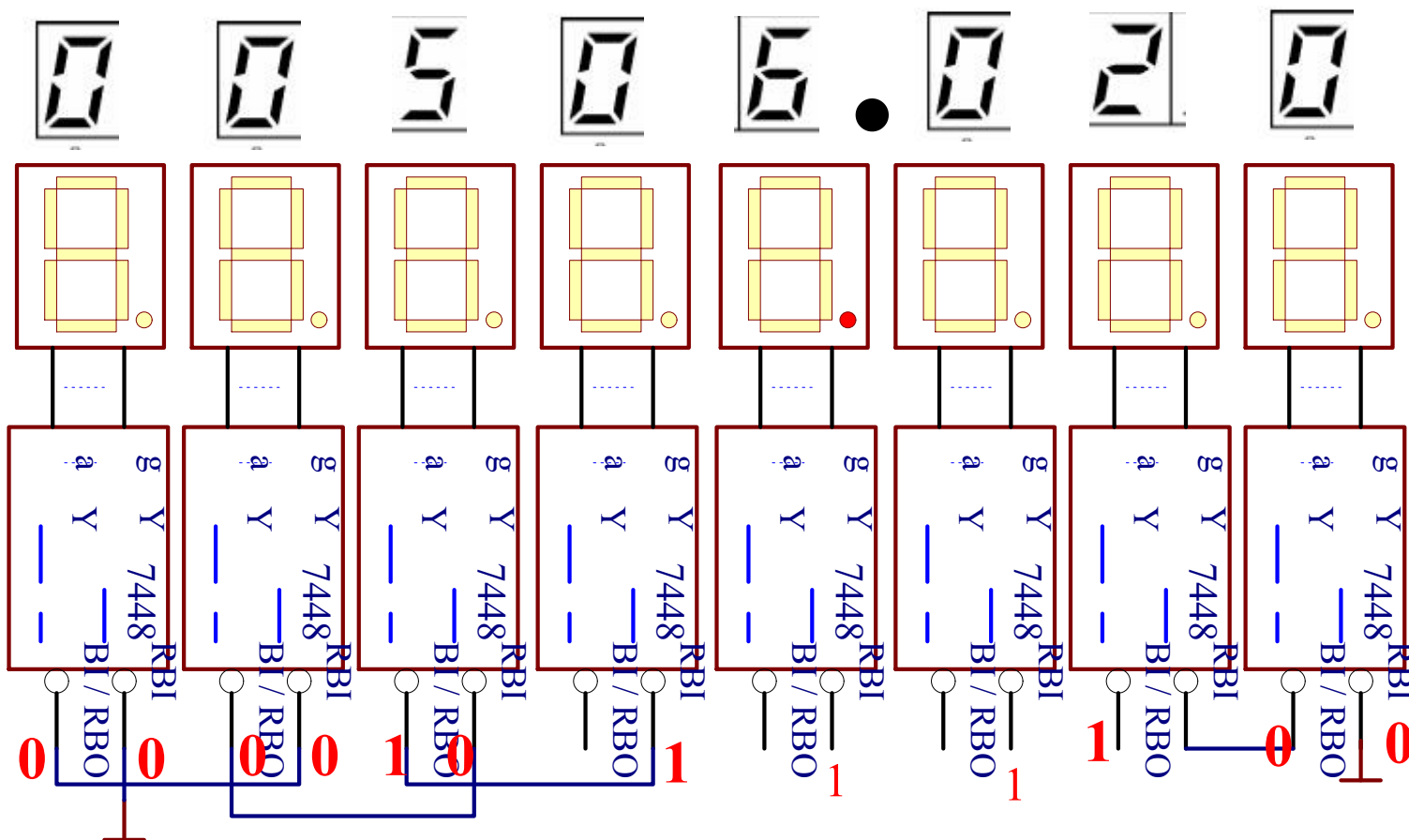


图 3.2.19 有灭零控制的8位数码显示系统

### 3.2.3 数据选择器 (MUX)

数据选择器又称**多路选择器**(Multiplexer, 简称MUX)。每次在地址输入的控制下, 从多路输入数据中选择一路输出。

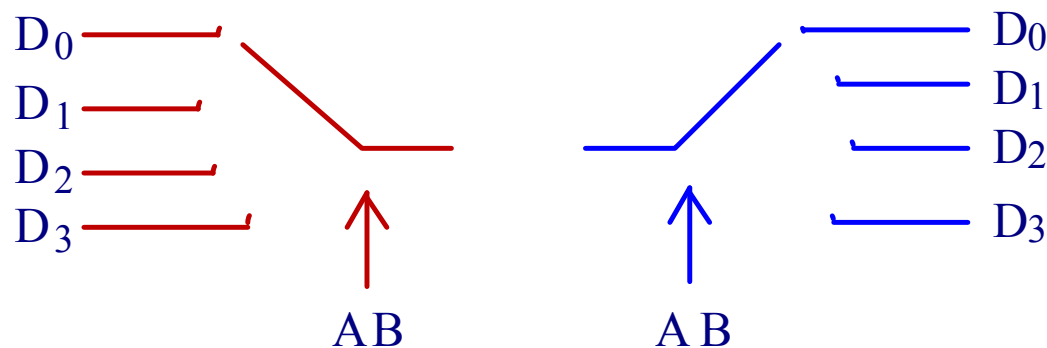
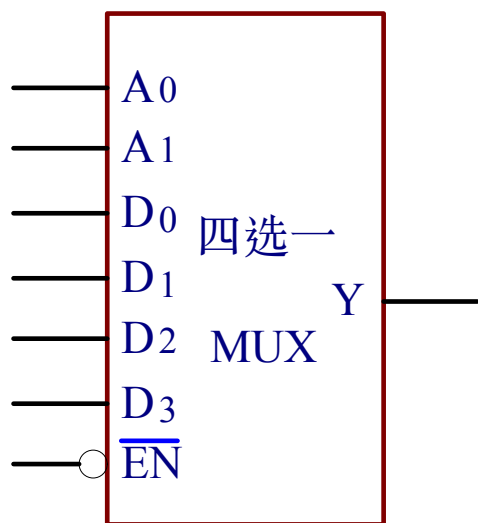


图 3.2.20 数据选择器和数据分配器示意图

# 1. 四选一数据选择器



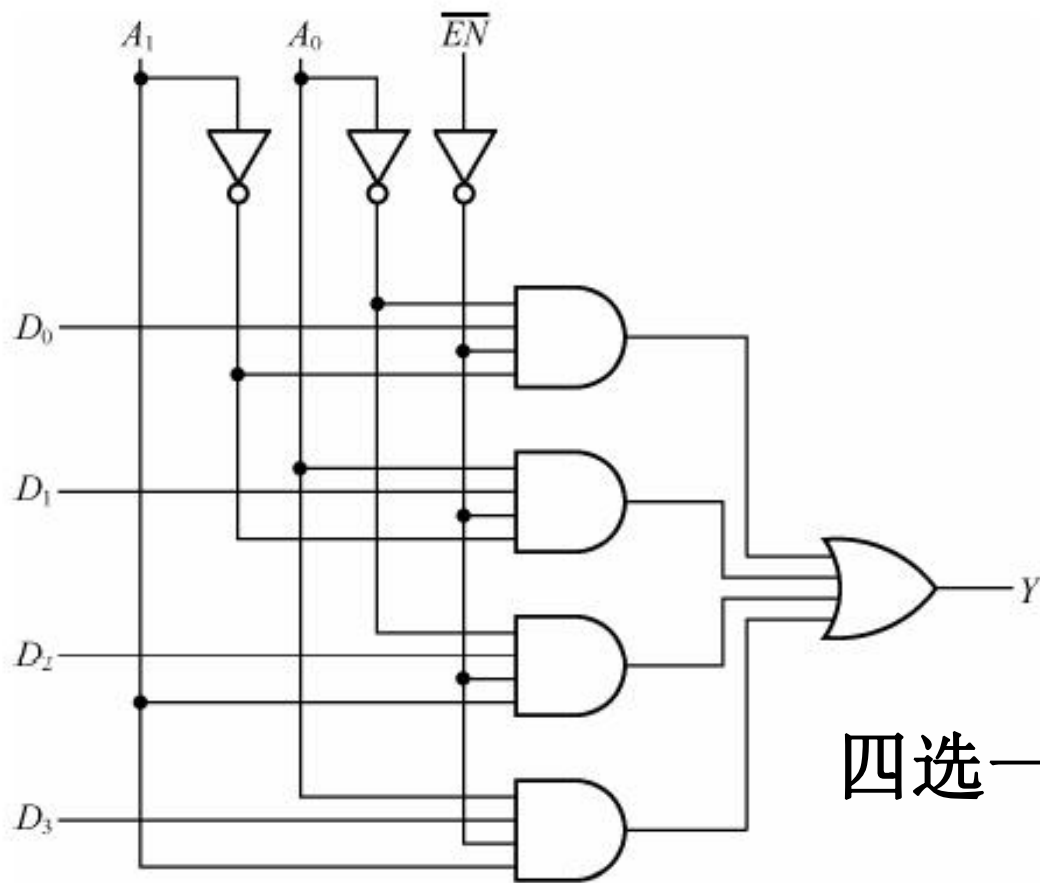
$A_1$ 、 $A_0$ ：地址输入端；

$D_3 \sim D_0$ ：数据输入端；

$\overline{EN}$ ：使能端；

$Y$ ：输出端；

图 3.2.21 (b) 简化符号



四选一MUX逻辑图

$$\left\{ \begin{array}{l} \overline{EN} = 1, Y = 0; \\ \overline{EN} = 0, Y = \overline{A_1}\overline{A_0}D_0 + \overline{A_1}A_0D_1 + A_1\overline{A_0}D_2 + A_1A_0D_3 \\ = \sum_{i=0}^3 m_i D_i \end{array} \right.$$

$$\overline{EN} = 0, Y = \overline{A_1}\overline{A_0}D_0 + \overline{A_1}A_0D_1 + A_1\overline{A_0}D_2 + A_1A_0D_3$$

$$= \sum_{i=0}^3 m_i D_i$$

$$\begin{cases} \overline{EN} = 1, Y = 0; \\ \overline{EN} = 0, Y = \overline{A_1}\overline{A_0}D_0 + \overline{A_1}A_0D_1 + A_1\overline{A_0}D_2 + A_1A_0D_3 \\ = \sum_{i=0}^3 m_i D_i \end{cases}$$

表 3.2.9 四选一MUX的功能表

使能 输入	输 入		输 出
$\overline{EN}$	$A_1$	$A_0$	$Y$
1	0	0	0
0	0	0	$D_0$
0	0	1	$D_1$
0	1	0	$D_2$
0	1	1	$D_3$

四选一MUX的卡诺图

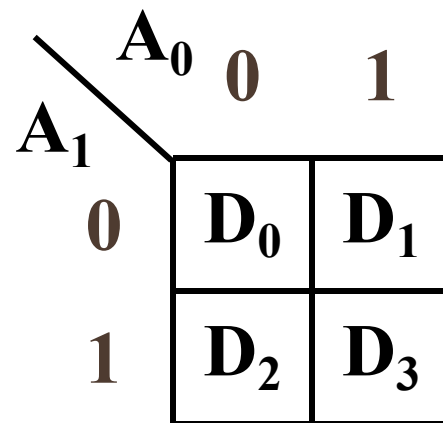


图3.2.26( b )

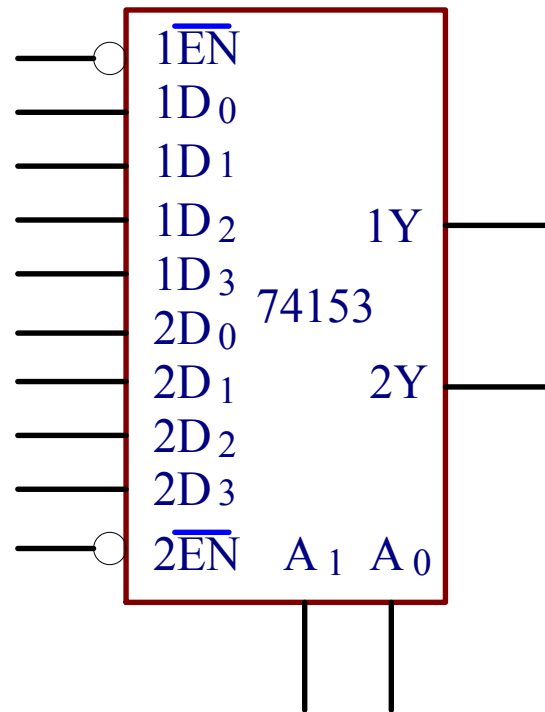
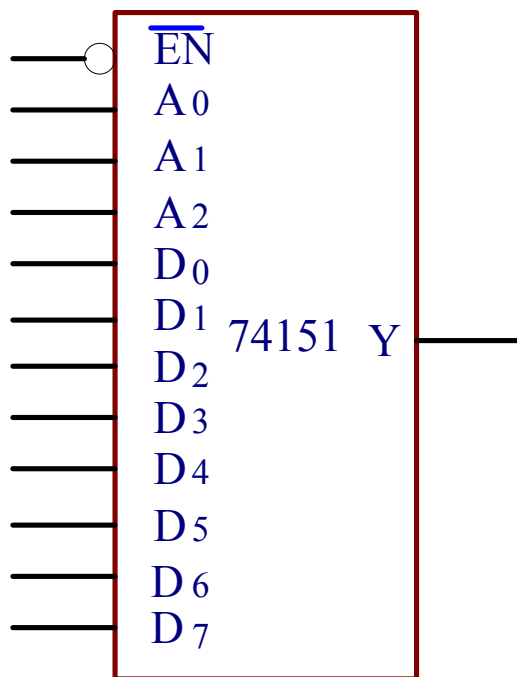


图 3.2.22 74153的简化逻辑符号

## 2. 八选一数据选择器



$A_2 \sim A_0$ : 地址输入端;

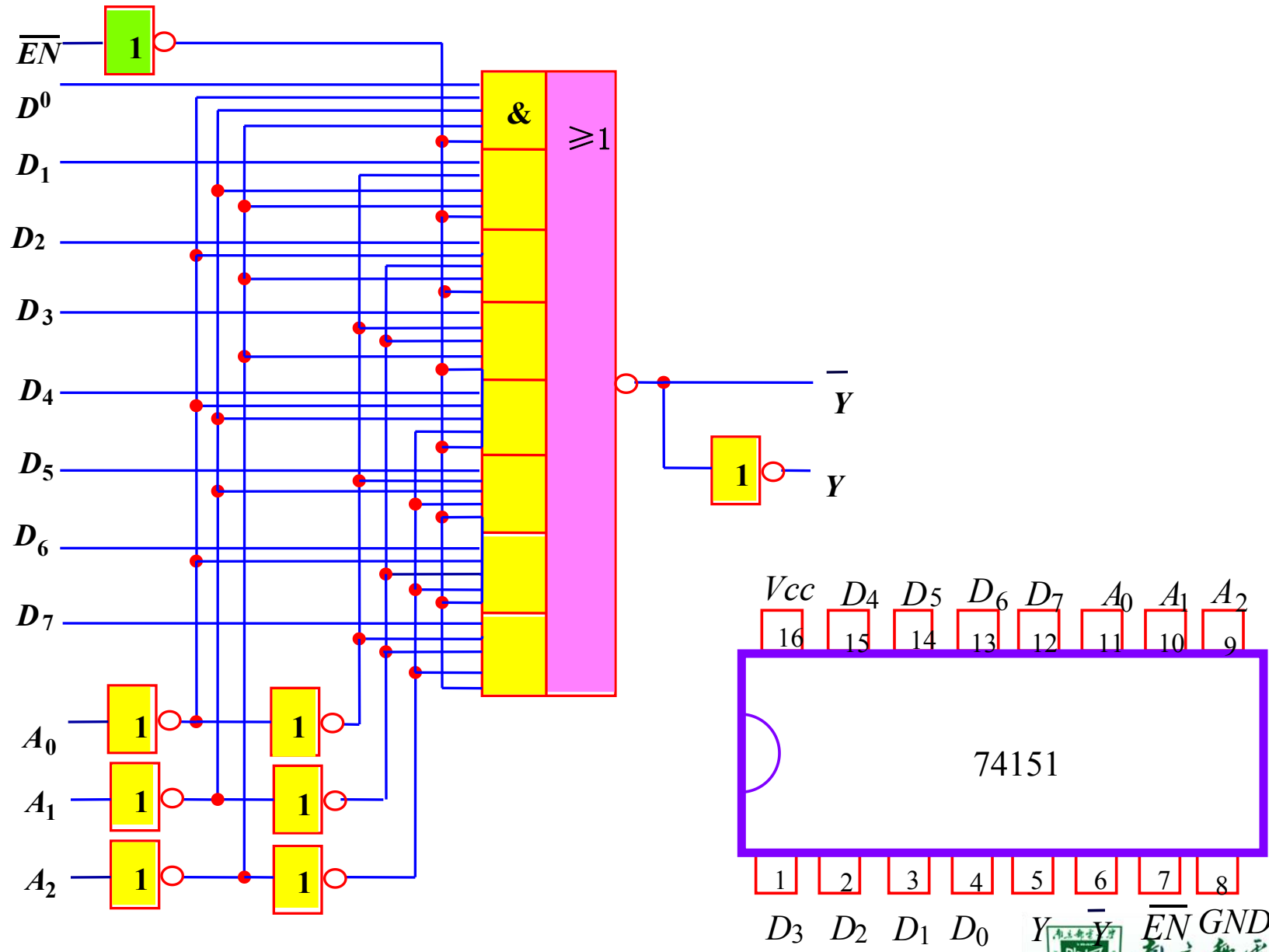
$D_7 \sim D_0$ : 数据输入端;

$\overline{EN}$ : 使能端;

Y: 输出端;

图 3.2.23 简化符号





## 八选一MUX的功能表

使能 输入	输 入			输 出
$\overline{EN}$	$A_2$	$A_1$	$A_0$	$Y$
1	0	0	0	0
0	0	0	0	$D_0$
0	0	0	1	$D_1$
0	0	1	0	$D_2$
0	0	1	1	$D_3$

使能 输入	输 入			输 出
$\overline{EN}$	$A_2$	$A_1$	$A_0$	$Y$
0	1	0	0	$D_4$
0	1	0	1	$D_5$
0	1	1	0	$D_6$
0	1	1	1	$D_7$

## 八选一MUX的卡诺图

		$A_1A_0$			
		00	01	11	10
$A_2$	0	$D_0$	$D_1$	$D_3$	$D_2$
	1	$D_4$	$D_5$	$D_7$	$D_6$

## 八选一MUX的逻辑表达式

$$\begin{cases} \overline{EN} = 1, Y = 0; \\ \overline{EN} = 0, Y = \overline{A_2}\overline{A_1}\overline{A_0}D_0 + \overline{A_2}\overline{A_1}A_0D_1 + \overline{A_2}A_1\overline{A_0}D_2 + \\ \overline{A_2}A_1A_0D_3 + A_2\overline{A_1}\overline{A_0}D_0 + A_2\overline{A_1}A_0D_1 + A_2A_1\overline{A_0}D_2 + \\ A_2A_1A_0D_3 \\ = \sum_{i=0}^7 m_i D_i \end{cases}$$

图3.2.24 (a)

### 3. 数据选择器的扩展

例：试用一片双4选1数选器74LS153组成一个8选1数据选择器。

解：

$A_2$	$A_1$ $A_0$	$Y$
0	00 ~ 11	$D_0 \sim D_3$
1	00 ~ 11	$D_4 \sim D_7$

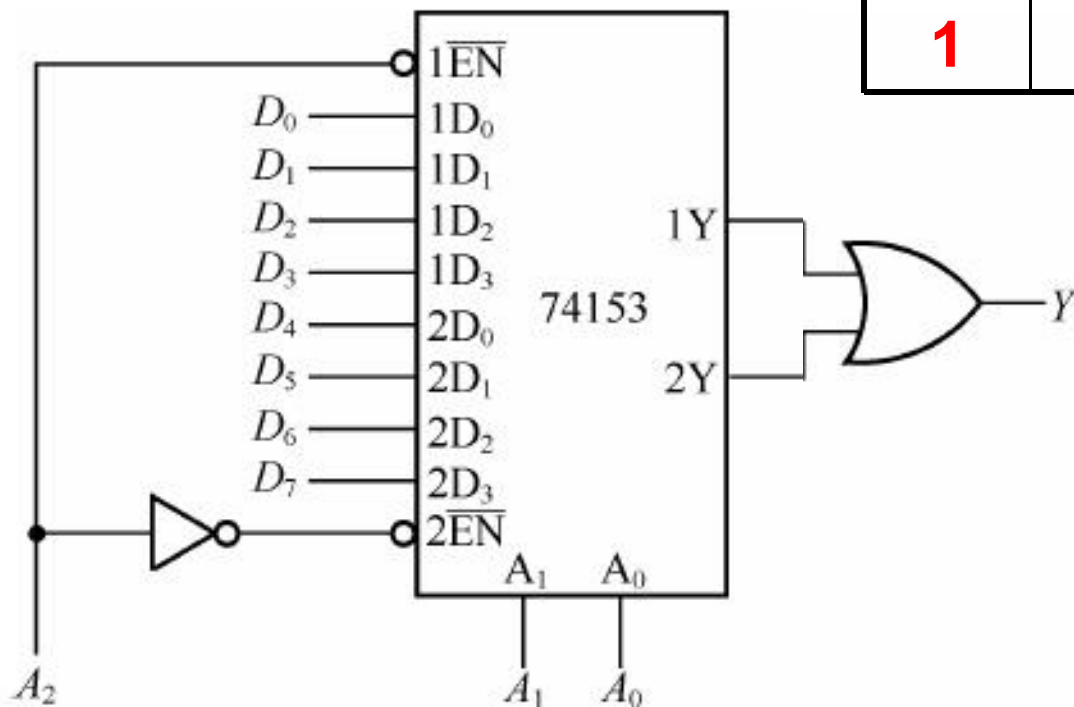


图 3.2.24 四选一扩展为八选一 MUX

例：试将8选1数选器74LS151扩展成一个32选1数据选择器。

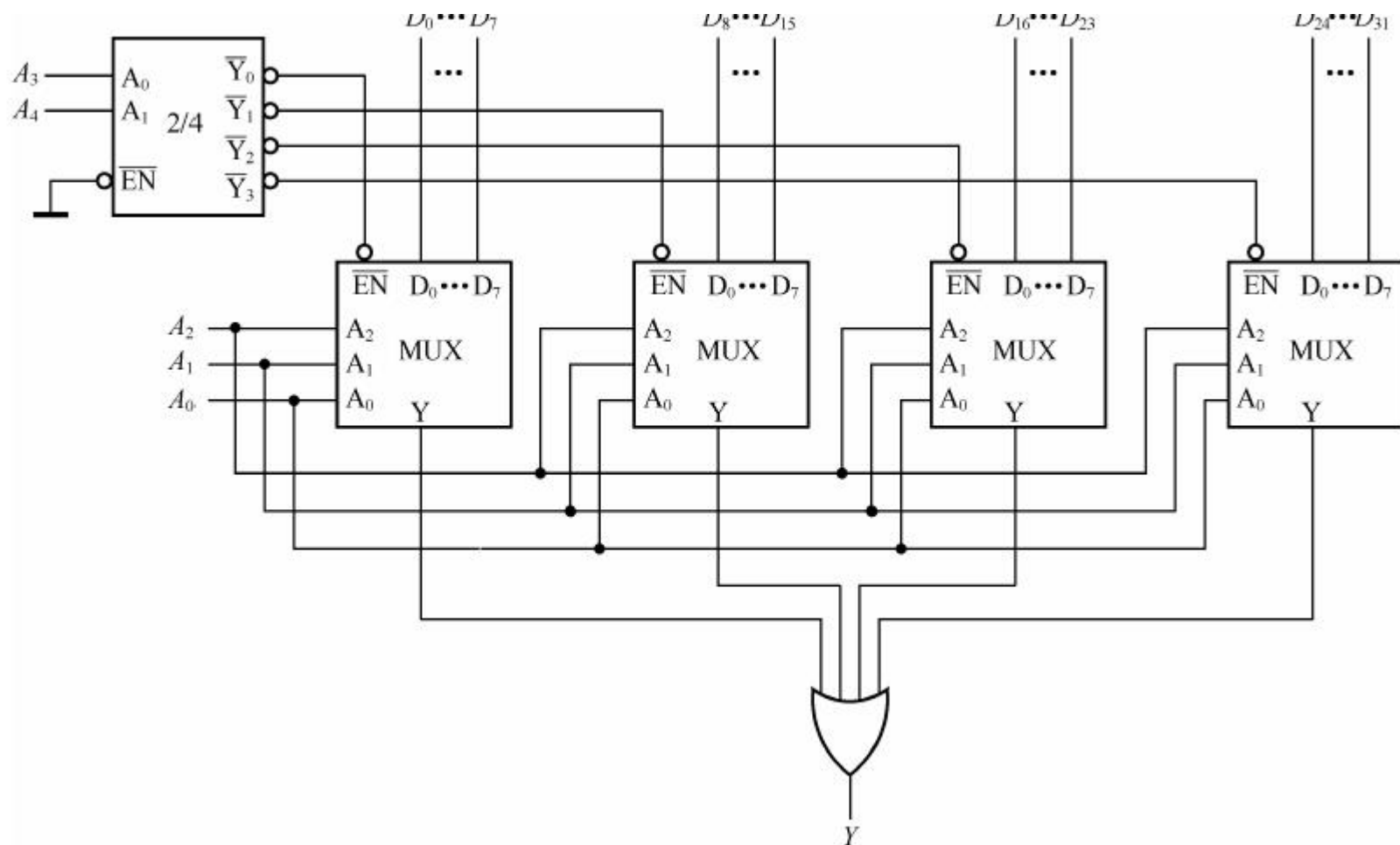


图 3.2.25 (a) 八选一扩展为三十二选一MUX

# 方法二

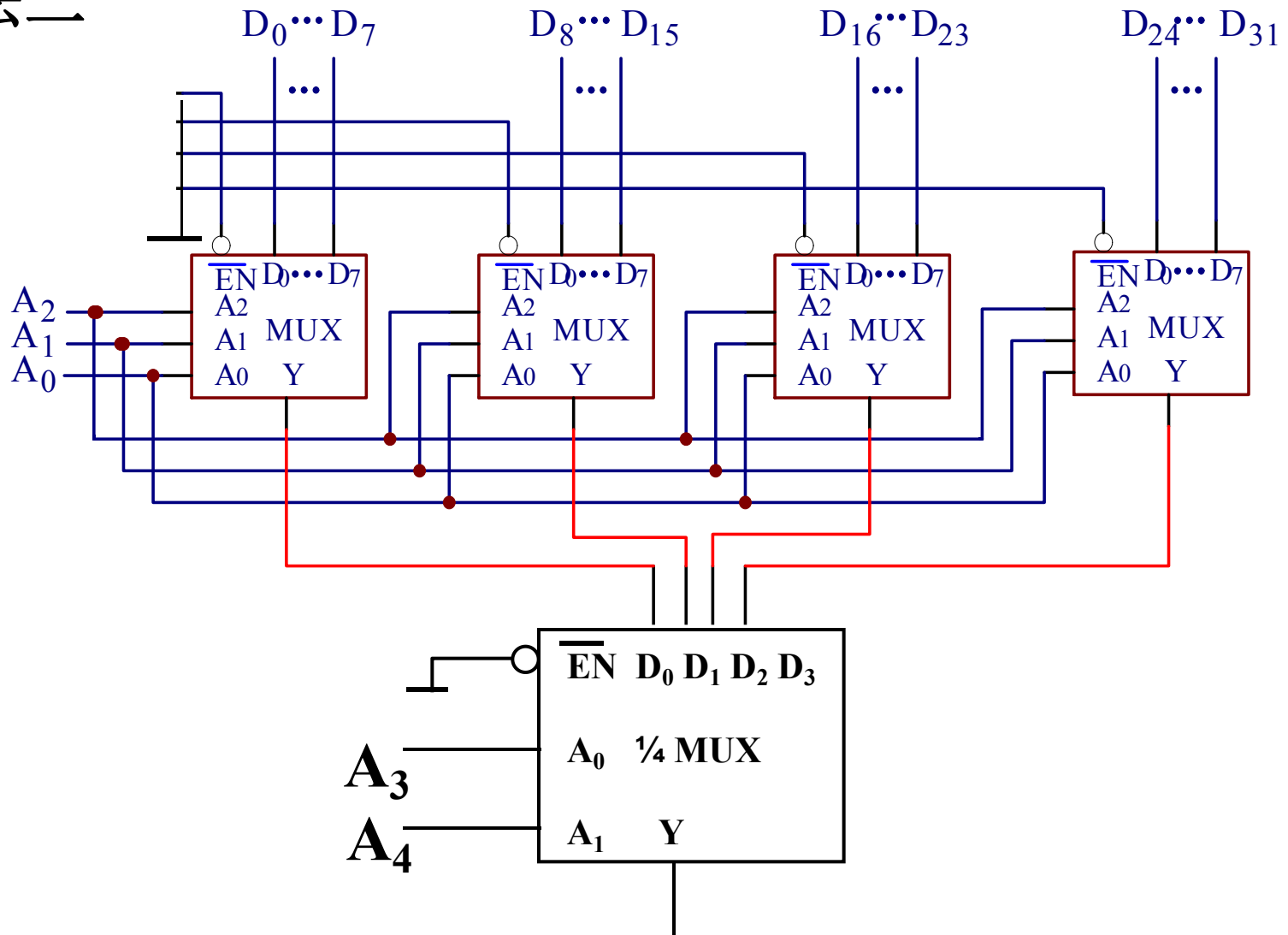


图 3.2.25 (b) 八选一扩展为三十二选一MUX

## 4.用数据选择器设计组合逻辑电路

例：试用1/2个74LS153实现如下逻辑函数的组合逻辑电路。

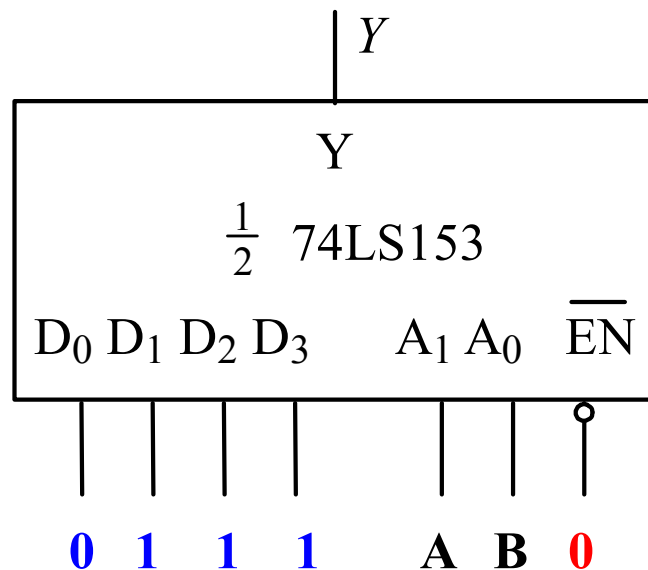
$$Y(A,B) = \overline{A}B + A$$

解：逻辑函数变形为最小项之和形式

$$\begin{aligned} Y &= \overline{A}B + A = \overline{A}B + \overline{A}B + AB \\ &= m_1 + m_2 + m_3 \\ &= D_0m_0 + D_1m_1 + D_2m_2 + D_3m_3 \end{aligned}$$

比较可得：

$$D_0=0, D_1=1, D_2=1, D_3=1$$



例2：用74151设计函数  $F = AB + A\bar{C}$ 。

		$A_1A_0$			
		00	01	11	10
$A_2$	0	$D_0$	$D_1$	$D_3$	$D_2$
	1	$D_4$	$D_5$	$D_7$	$D_6$

Y

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

F

比较两卡诺图，若A、B、C分别接 $A_2$ 、 $A_1$ 、 $A_0$ ，再令 $D_0=D_1=D_2=D_3=D_5=0$ ， $D_4=D_6=D_7=1$ ，则 $Y=F$ ，相应的电路图如下所示：



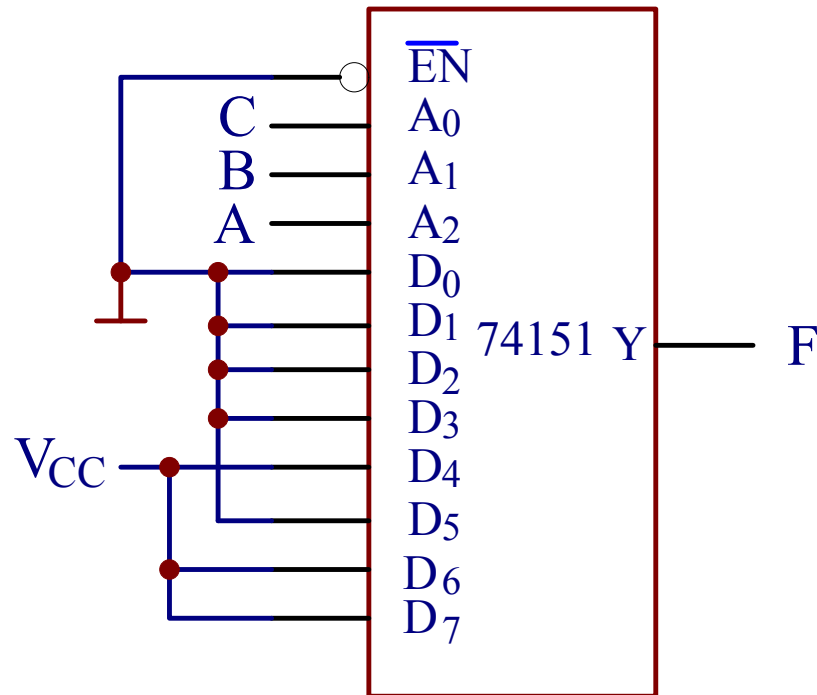


图 3.2.28 ( a )

若C、B、A分别接 $A_2$ 、 $A_1$ 、 $A_0$ ，则Y、F的卡诺图分别如下，

		$A_1A_0$			
		00	01	11	10
$A_2$	0	$D_0$	$D_1$	$D_3$	$D_2$
	1	$D_4$	$D_5$	$D_7$	$D_6$

Y

		$BA$			
		00	01	11	10
C	0	0	1	1	0
	1	0	0	1	0

F

再令 $D_0=D_2=D_4=D_5=D_6=0$ ， $D_1=D_3=D_7=1$ ，则 $Y=F$ ，相应的电路图如下所示：

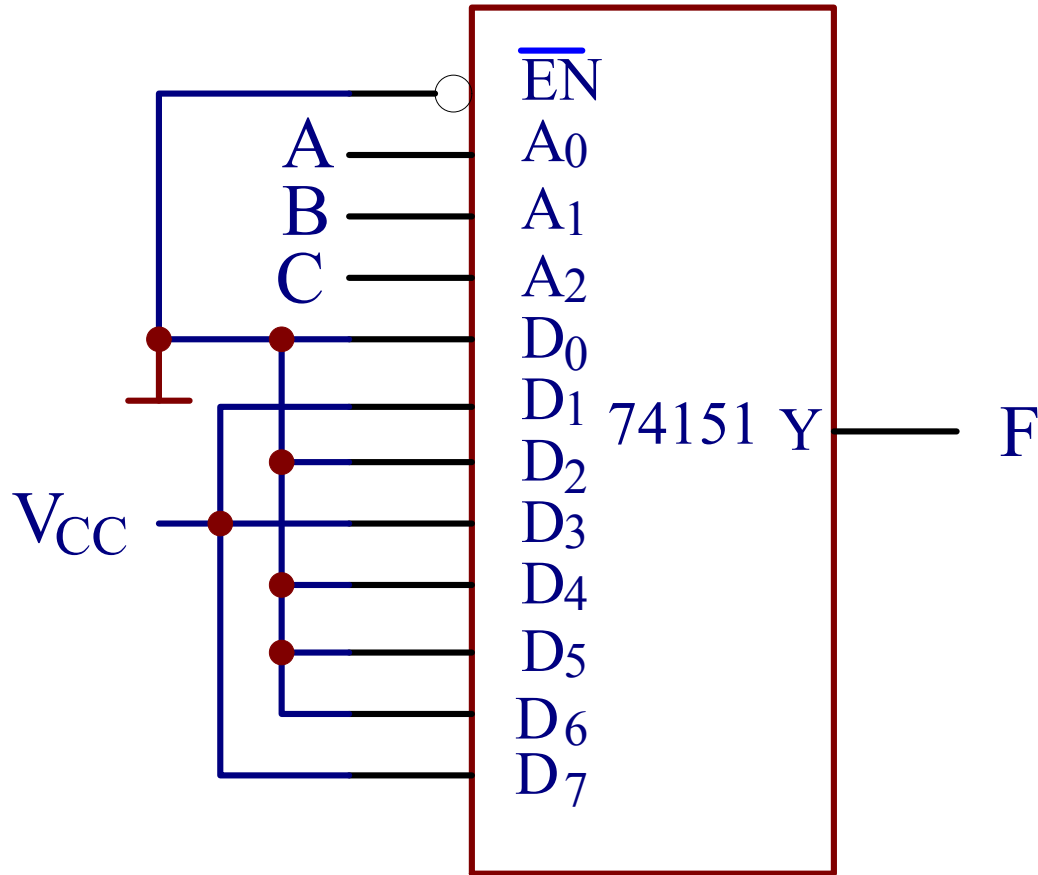


图 3.2.28 ( b )

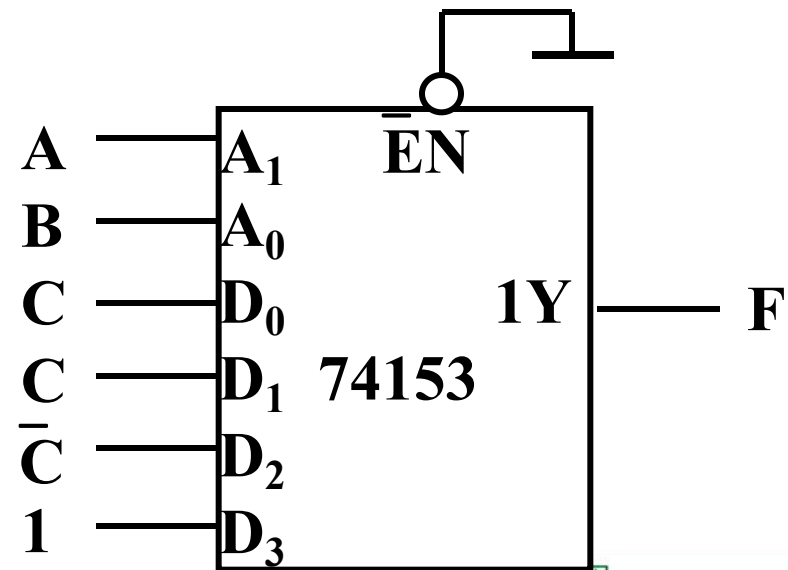
若要实现的组合逻辑函数 $F$ 有 $m$ 个变量，提供的数据选择器的地址端为 $n$ ，且 $n < m$ ，例如函数 $F(ABC)$ 用四选一数据选择器实现。则需对函数 $F(ABC)$ 降维。

降维方法：1、代数法；2、卡诺图法

## (1) 代数法

例1:  $F(ABC) = A\bar{C} + \bar{A}C + ABC$ , 用1/2个  
74153实现。

$$\begin{aligned} \text{解: } F(ABC) &= A\bar{C} + \bar{A}C + ABC \\ &= \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + AB1 \end{aligned}$$



## (2) 卡诺图法

例2：用一片74153设计一个一位全加器。

解：一位全加器的卡诺图分别如下

		$B_i C_{i-1}$			
		00	01	11	10
$A_i$	0	0	1	0	1
	1	1	0	1	0

$S_i$

		$B_i C_{i-1}$			
		00	01	11	10
$A_i$	0	0	0	1	0
	1	0	1	1	1

$C_i$

		$A_0$	
		0	1
$A_1$	0	$D_0$	$D_1$
	1	$D_2$	$D_3$

$Y$

		$B_i C_{i-1}$			
		00	01	11	10
$A_i$	0	0	1	0	1
	1	1	0	1	0

$S_i$

降1维



		$B_i$	
		0	1
$A_i$	0	$C_{i-1}$	$\bar{C}_{i-1}$
	1	$\bar{C}_{i-1}$	$C_{i-1}$

$S_i$

		$B_i C_{i-1}$			
		00	01	11	10
$A_i$	0	0	0	1	0
	1	0	1	1	1

$C_i$

降1维



		$B_i$	
		0	1
$A_i$	0	0	$C_{i-1}$
	1	$C_{i-1}$	1

$C_i$

全0填0；全1填1；

既有0也有1，看1所在小格对应被降变量取值



	$B_i$	0	1
$A_i$	0	$C_{i-1}$	$\overline{C}_{i-1}$
	1	$\overline{C}_{i-1}$	$C_{i-1}$

$S_i$

	$B_i$	0	1
$A_i$	0	0	$C_{i-1}$
	1	$C_{i-1}$	1

$C_i$

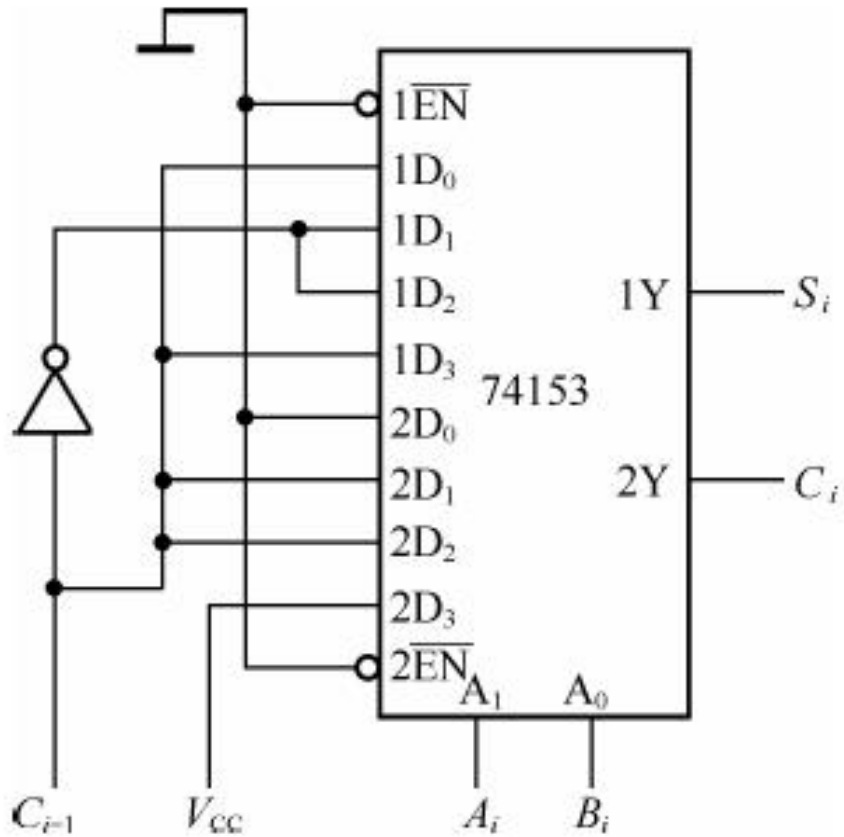


图 3.2.30



用数据选择器设计组合逻辑电路的步骤：

(1) 降维；(可选)

(2) 比较；

(3) 画逻辑图。

例3.2.1：分别用一片74151和 $\frac{1}{2}$  74153 实现函数

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}BC + AB\bar{D} + A\bar{B}D + AC\bar{D}。$$

解：这是一个四变量函数，对其一次降维后可用74151实现，两次降维后可用 $\frac{1}{2}$  74153实现。

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

降维D  
→

		C	
		0	1
AB	00	1	
	01		1
	11	$\bar{D}$	$\bar{D}$
	10	D	1

↓ 降维C

		C
		$\bar{C}$
AB	00	$\bar{C}$
	01	C
	11	$\bar{D}$
	10	C+D

( a )

图 3.2.28

	<b>C</b>	
	<b>0</b>	<b>1</b>
<b>AB</b>		
<b>00</b>	<b>1</b>	
<b>01</b>		<b>1</b>
<b>11</b>	$\bar{D}$	$\bar{D}$
<b>10</b>	<b>D</b>	<b>1</b>

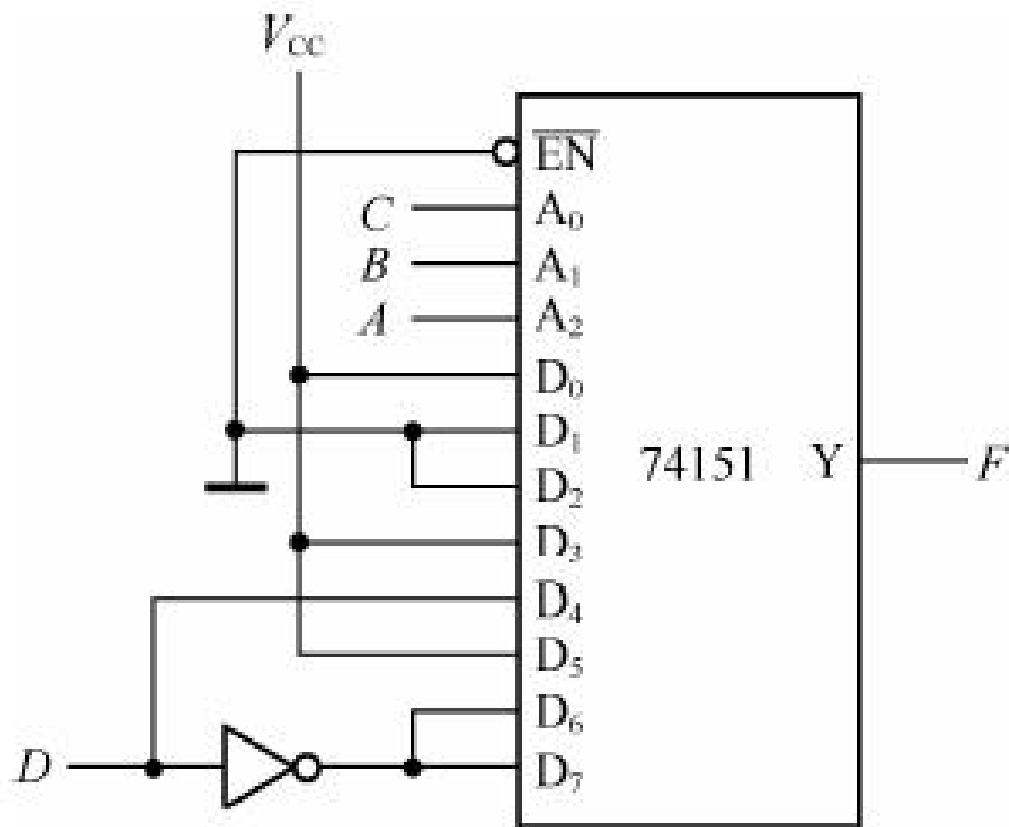


图 3.2.32

<b>AB</b>	
<b>00</b>	$\bar{C}$
<b>01</b>	$C$
<b>11</b>	$\bar{D}$
<b>10</b>	$C+D$

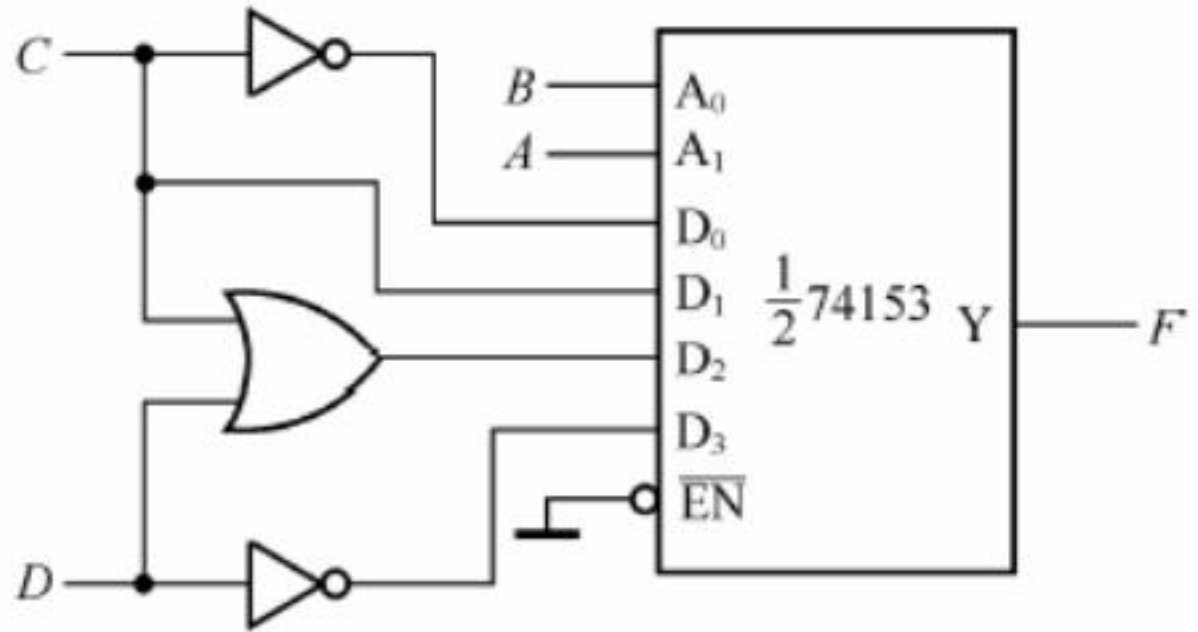
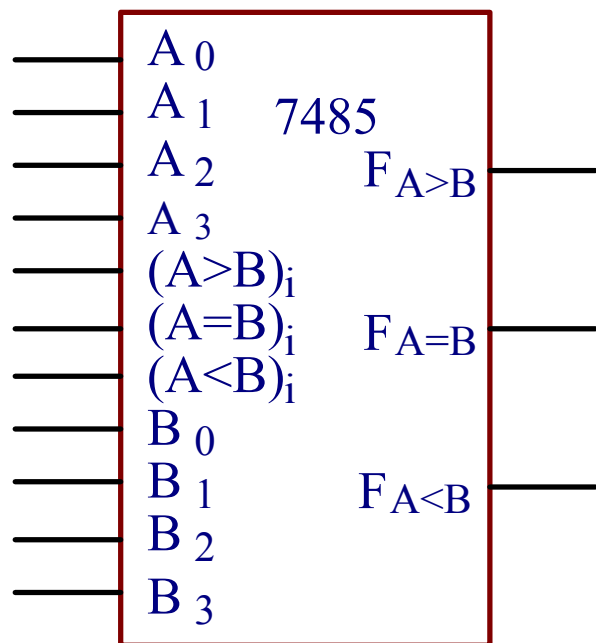


图 3.2.32 (c)

## 3.2.4 数据比较器

### 1. 四位并行数据比较器7485



$A_3 \sim A_0$ 、 $B_3 \sim B_0$  :

数码输入端;

$(A>B)_i$   $(A=B)_i$   $(A<B)_i$  :

级联输入端;

$F_{A>B}$ 、 $F_{A=B}$ 、 $F_{A<B}$  : 比较结果输出端;

图 3.2.33 (b) 简化符号

表 3.2.10 4位数值比较器7485功能表

输 入				输 出					
$A_3 B_3$	$A_2 B_2$	$A_1 B_1$	$A_0 B_0$	$(A>B)_i$	$(A<B)_i$	$(A=B)_i$	$F_{A>B}$	$F_{A<B}$	$F_{A=B}$
$A_3 > B_3$	$\emptyset \emptyset$	$\emptyset \emptyset$	$\emptyset \emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	0	0
$A_3 < B_3$	$\emptyset \emptyset$	$\emptyset \emptyset$	$\emptyset \emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	0	1	0
$A_3 = B_3$	$A_2 > B_2$	$\emptyset \emptyset$	$\emptyset \emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	0	0
$A_3 = B_3$	$A_2 < B_2$	$\emptyset \emptyset$	$\emptyset \emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	$\emptyset \emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	$\emptyset \emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	$\emptyset$	$\emptyset$	$\emptyset$	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	$\emptyset$	$\emptyset$	$\emptyset$	0	1	0

## 续表

输 入				输 出					
$A_3 B_3$	$A_2 B_2$	$A_1 B_1$	$A_0 B_0$	$(A > B)_i$	$(A < B)_i$	$(A = B)_i$	$F_{A > B}$	$F_{A < B}$	$F_{A = B}$
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	0	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	0	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	0	1	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	1	0	0	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	1	1	0	0	1

## 2. 数据比较器的扩展

### (1) 串联方式

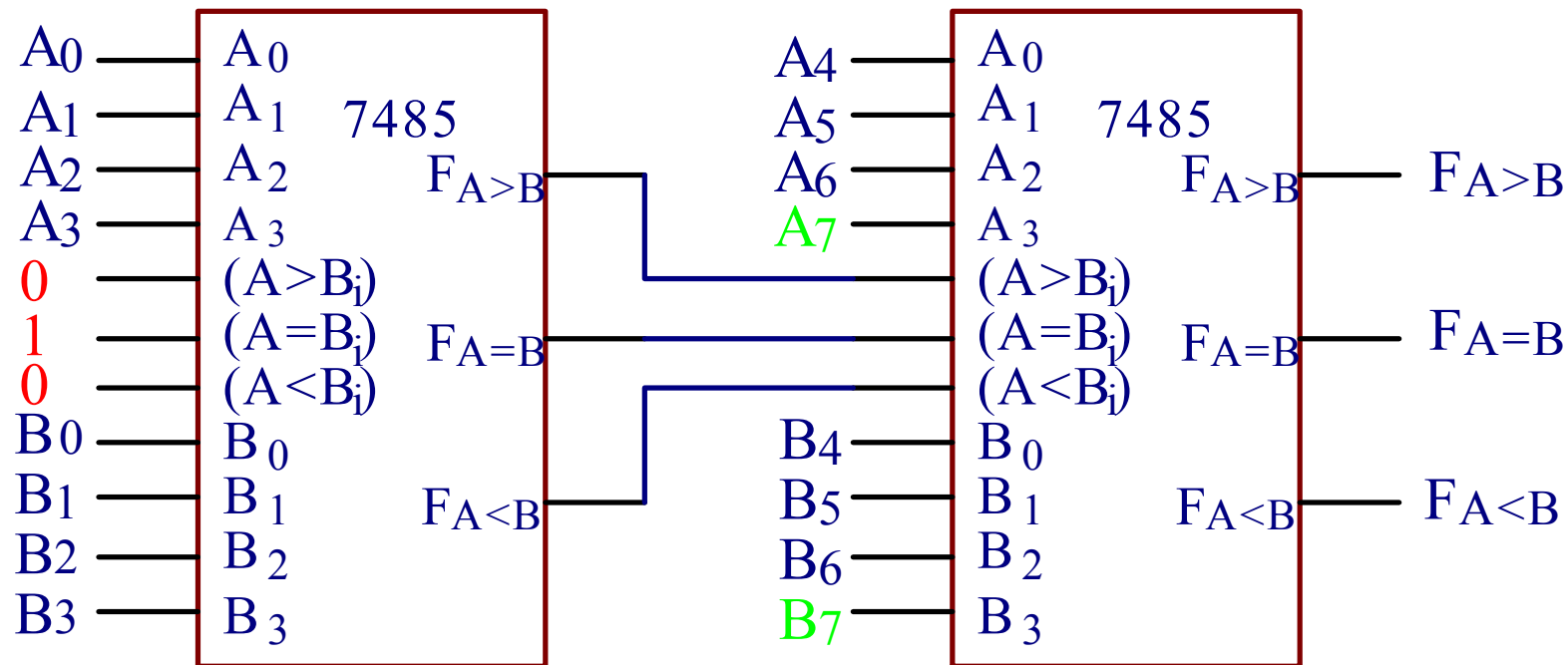


图 3.2.34



## (2) 并联方式

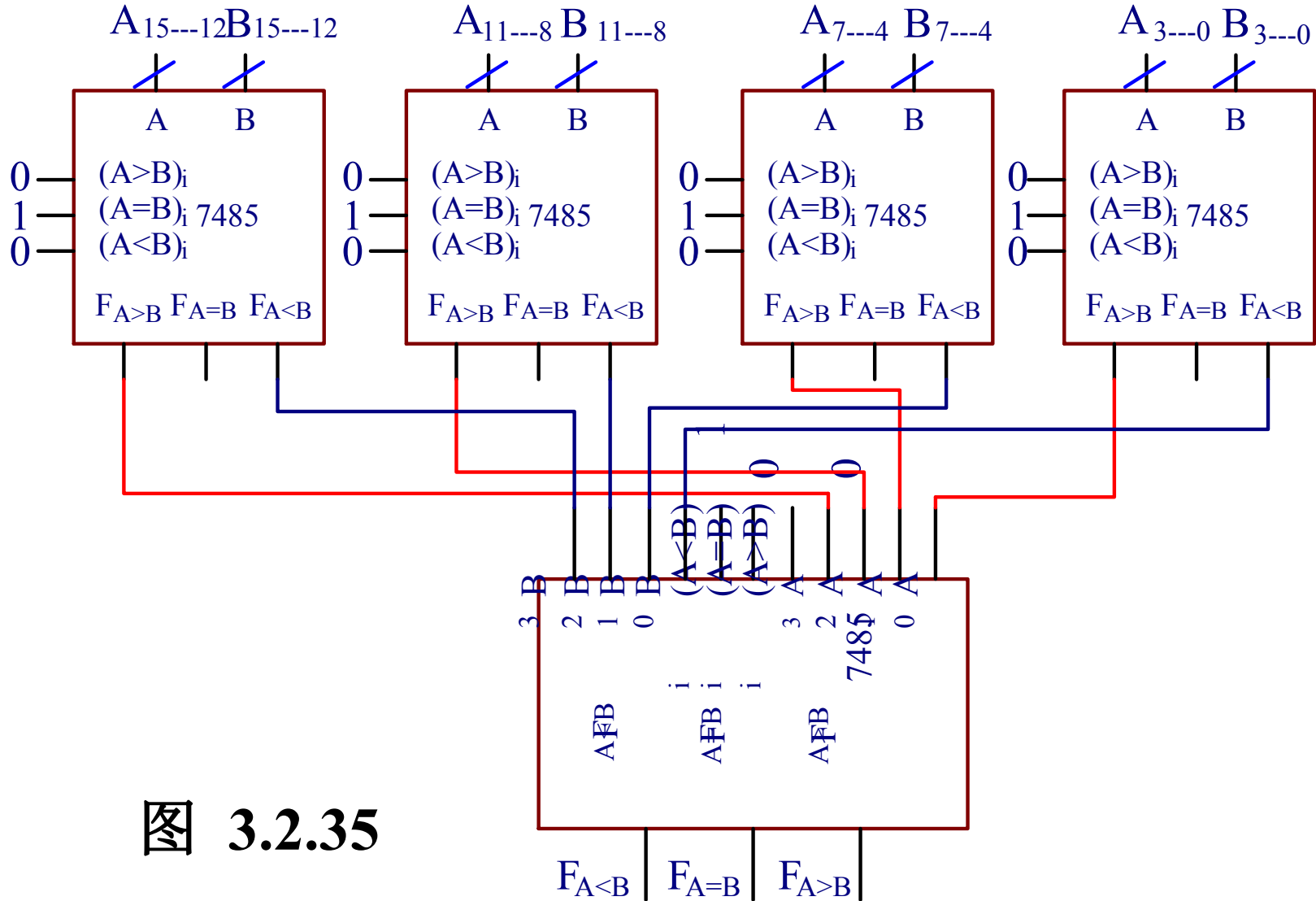


图 3.2.35

例：用一片7485实现两个5位二进制数 $A_4A_3A_2A_1A_0$ 和 $B_4B_3B_2B_1B_0$ 的比较。

解：关键在最低位的比较

$A_0$	$B_0$	$(A>B)_i$	$(A<B)_i$	$(A=B)_i$
0	0	$\Phi$	$\Phi$	1
0	1	0	1	0
1	0	1	0	0
1	1	$\Phi$	$\Phi$	1

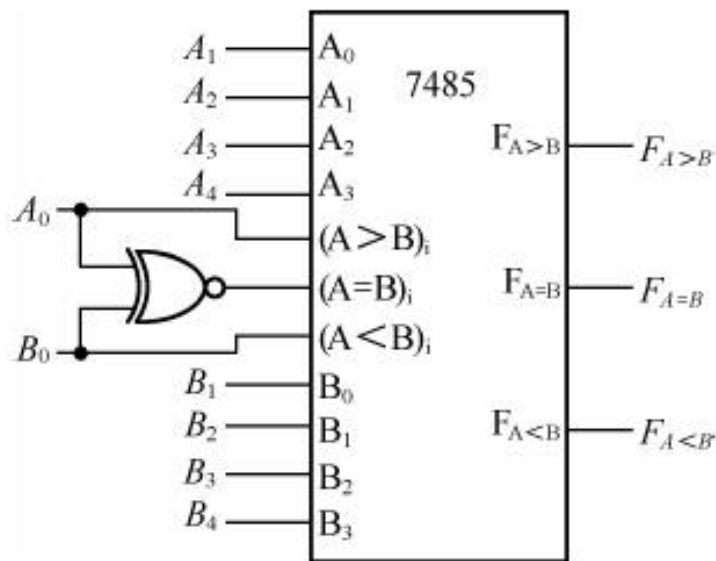
输入			输出		
$(A>B)_i$ $(A<B)_i$			$F_{A>B}$	$F_{A<B}$	
$(A=B)_i$			$F_{A=B}$		
1	0	0	1	0	0
0	1	0	0	1	0
0	0	1	0	0	1
0	0	0	1	1	0
0	1	1	0	0	1
1	0	1	0	0	1
1	1	0	0	0	0
1	1	1	0	0	1

$A_0$	$B_0$	$(A>B)_i$	$(A<B)_i$	$(A=B)_i$
0	0	$\Phi$	$\Phi$	1
0	1	0	1	0
1	0	1	0	0
1	1	$\Phi$	$\Phi$	1

$(A>B)_i = A_0$

$(A<B)_i = B_0$

$(A=B)_i = A_0 \odot B_0$



两个5位二进制数 $A_4A_3A_2A_1A_0$ 和 $B_4B_3B_2B_1B_0$ 的比较

## 3.2.5 全加器

### 1. 四位串行进位全加器

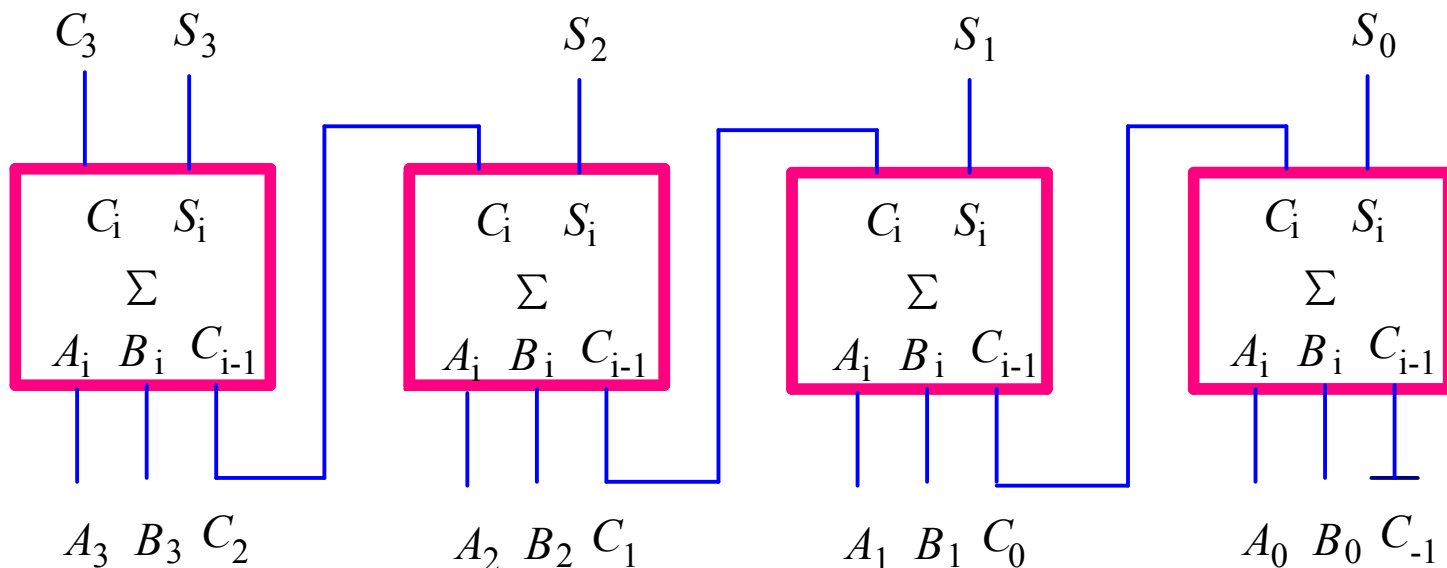


图 3.2.35

## 2. 四位超前进位全加器

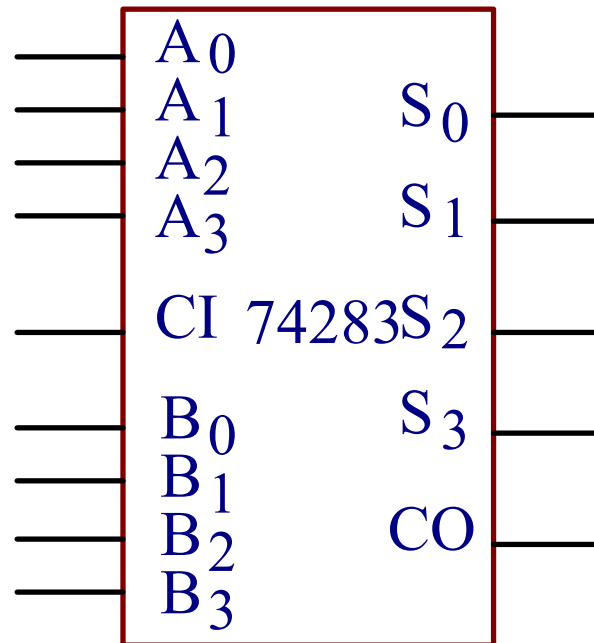


图 3.2.38 (b) 简化符号

由于  $S_i = A_i \oplus B_i \oplus C_{i-1}$

$$C_i = A_i B_i + A_i C_{i-1} + B_i C_{i-1}$$

则  $S_0 = A_0 \oplus B_0 \oplus CI = f_0 ( A_0, B_0, CI )$

$$C_0 = A_0 B_0 + A_0 CI + B_0 CI = g_0 ( A_0, B_0, CI )$$

$$S_1 = A_1 \oplus B_1 \oplus C_0 = f_0 ( A_1, B_1, C_0 )$$

$$= f_0 ( A_1, B_1, g_0 ( A_0, B_0, CI ) )$$

$$= f_1 ( A_1, B_1, A_0, B_0, CI )$$

$$C_1 = A_1B_1 + A_1C_0 + B_1C_0 = g_0 ( A_1, B_1, C_0 )$$

$$= g_1 ( A_1, B_1, A_0, B_0, CI )$$

⋮

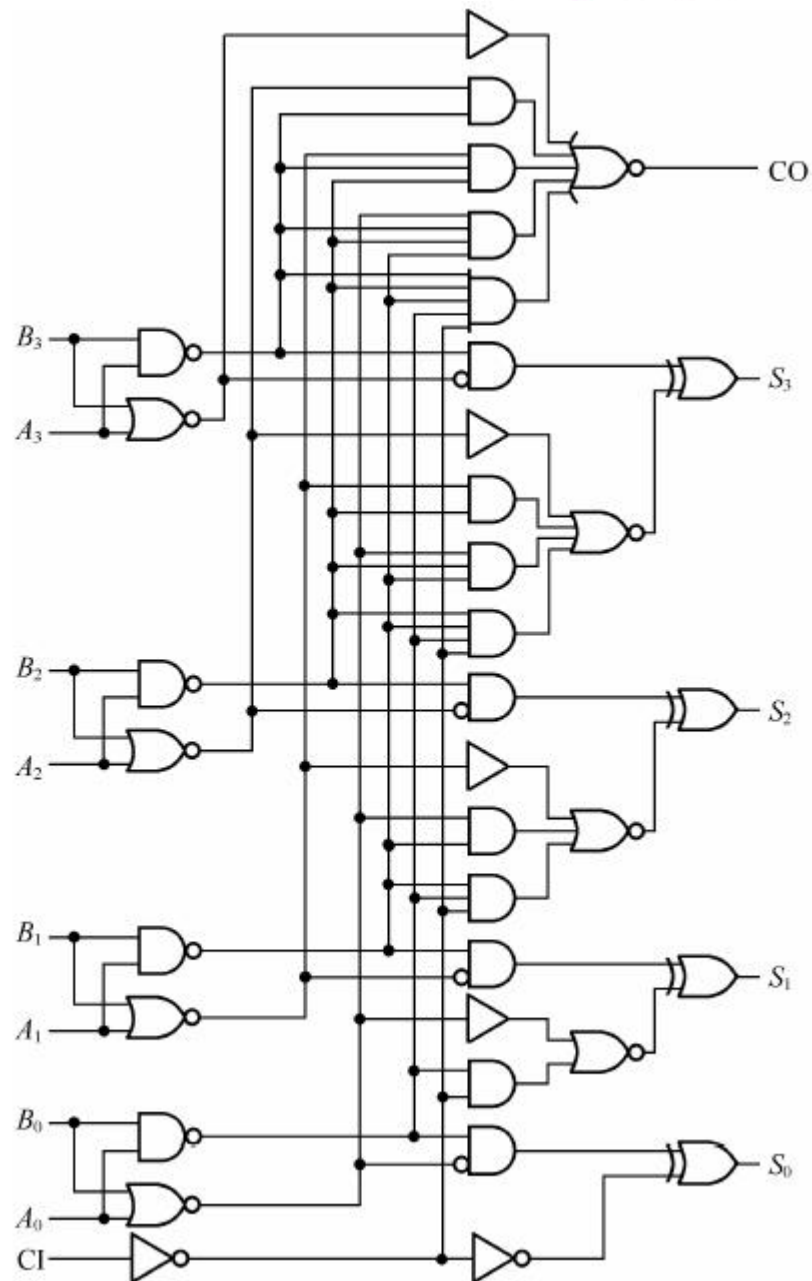
$$S_3 = f_3 ( A_3, A_2, A_1, A_0, B_3, B_2, B_1, B_0, CI )$$

$$C_3 = g_3 ( A_3, A_2, A_1, A_0, B_3, B_2, B_1, B_0, CI )$$

在加法运算前，根据进位 $CO_i$ 是 $A_{i-1}, A_{i-2}, \dots, A_0$ 及 $B_{i-1}, B_{i-2}, \dots, B_0$ 的函数关系得到每个位的进位 $CI_i$ ，这样一次就可以完成整个加法运算。

## 4位超前进位加法器 74LS283的逻辑图

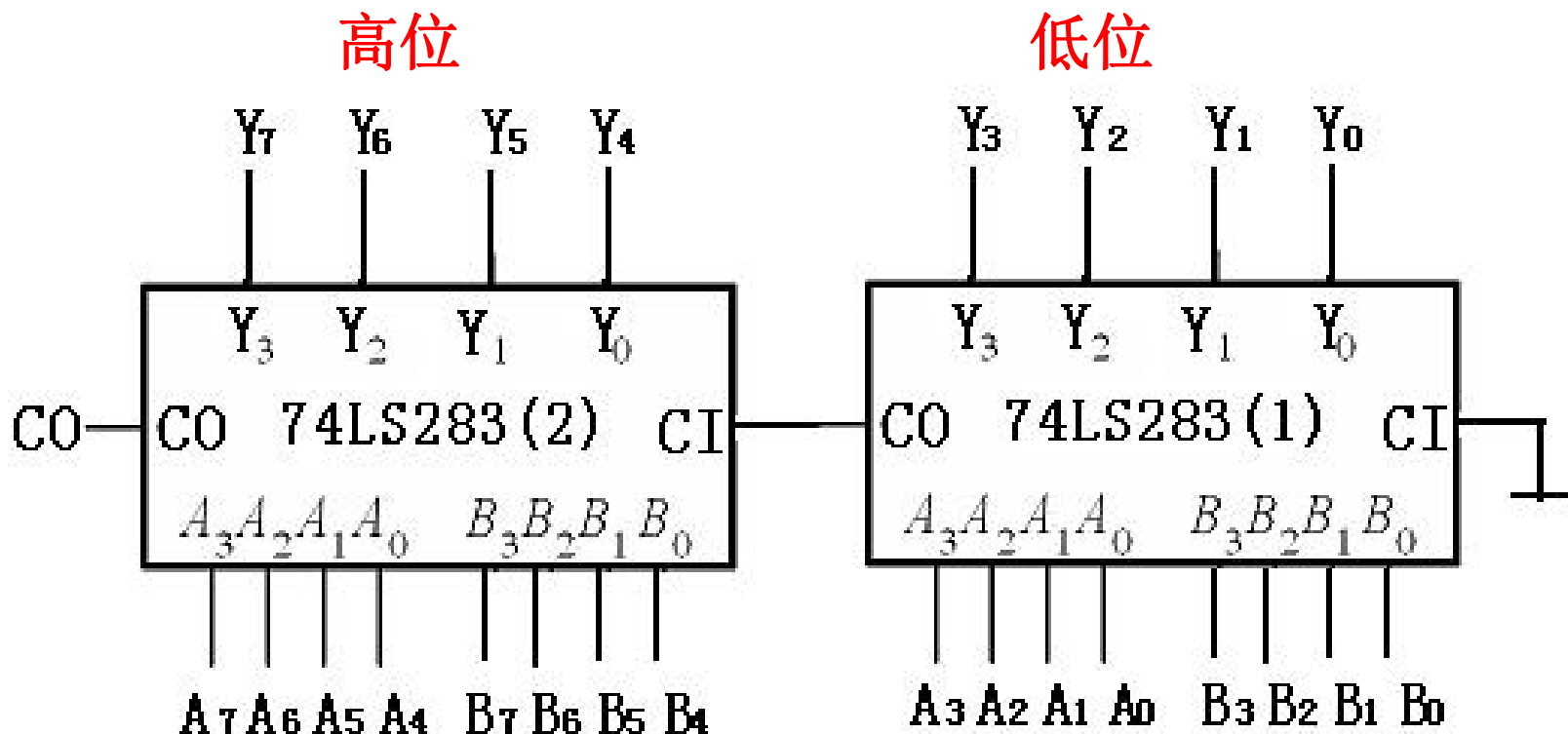
只需经过三级门电路  
的延迟时间，等价于1位  
全加器的时间延迟。





例：试用两片4位超前进位加法器74LS283构成一个8位加法器。

解：低位芯片的高位进位输出端接高位芯片的低位进位输入端。



### 3. 全加器的应用举例

例 3.2.2 已知BCD码  $(A_3A_2A_1A_0.a_3a_2a_1a_0)_{8421BCD}$ ，试用中规模器件和部分门电路设计一个电路将该数四舍五入。

解：分析：当小数部分大于4时，整数部分应加1，即

$$\begin{array}{r}
 A_3A_2A_1A_0 \\
 + \quad \quad \quad 1 \\
 \hline
 CO \ S_3 \ S_2 \ S_1 \ S_0
 \end{array}$$

当相加结果  $S_3 S_2 S_1 S_0$  为**1010**时，应进行修正。

选择芯片：7485 和74283。7485实现小数部分与4的比较，74283实现整数部分加1和修正（加6）功能。

需要设计一个“修正”电路F, F=1时表示需要修正, F=0时表示不需要修正。函数F输出1必须满足两个条件：

(1) $a_3a_2a_1a_0 > 4$ ; (2) $A_3A_2A_1A_0 = 1001$ ;

若 $F_1$ 表示条件1,  $F_2$ 表示条件2, 则 $F = F_1 \cdot F_2$ 。

## 函数F2真值表

输入				输出
A3	A2	A1	A0	F2
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	Φ
1	0	1	1	Φ
1	1	0	0	Φ
1	1	0	1	Φ
1	1	1	0	Φ
1	1	1	1	Φ

$$F_2 = A_3 A_0$$

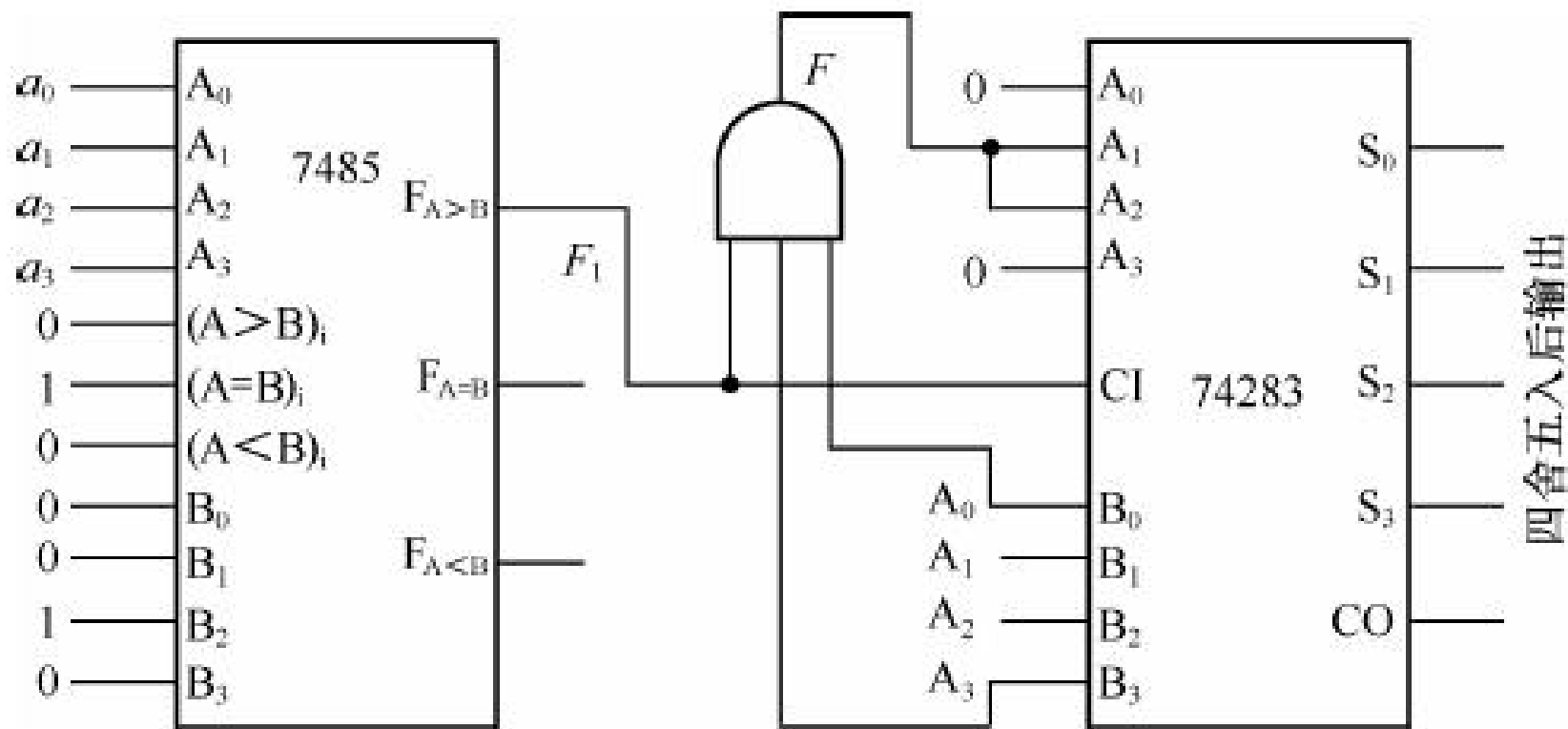


图 3.2.43 例 3.2.7 的电路图

## 3.3 竞争和冒险

### 3.3.1 竞争和冒险的概念

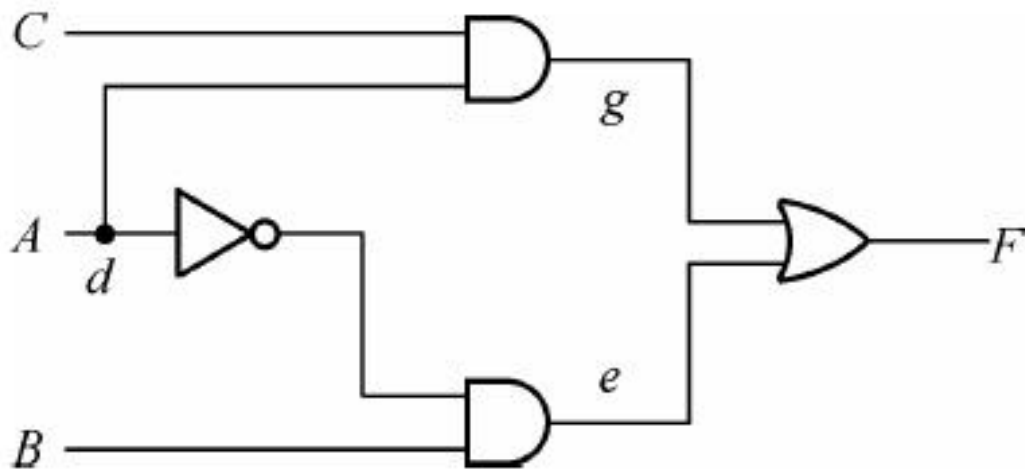
#### 1. 竞争

(1) 由于连线和集成门有一定的延迟时间，致使同一输入信号经过不同路径到达输出端有先有后

(1个或1个以上输入信号变化)；

(2) 多个输入信号同时变化，由于变化的快慢不同，致使多个输入信号到达输出端有先有后（2个或2个以上输入信号变化）。

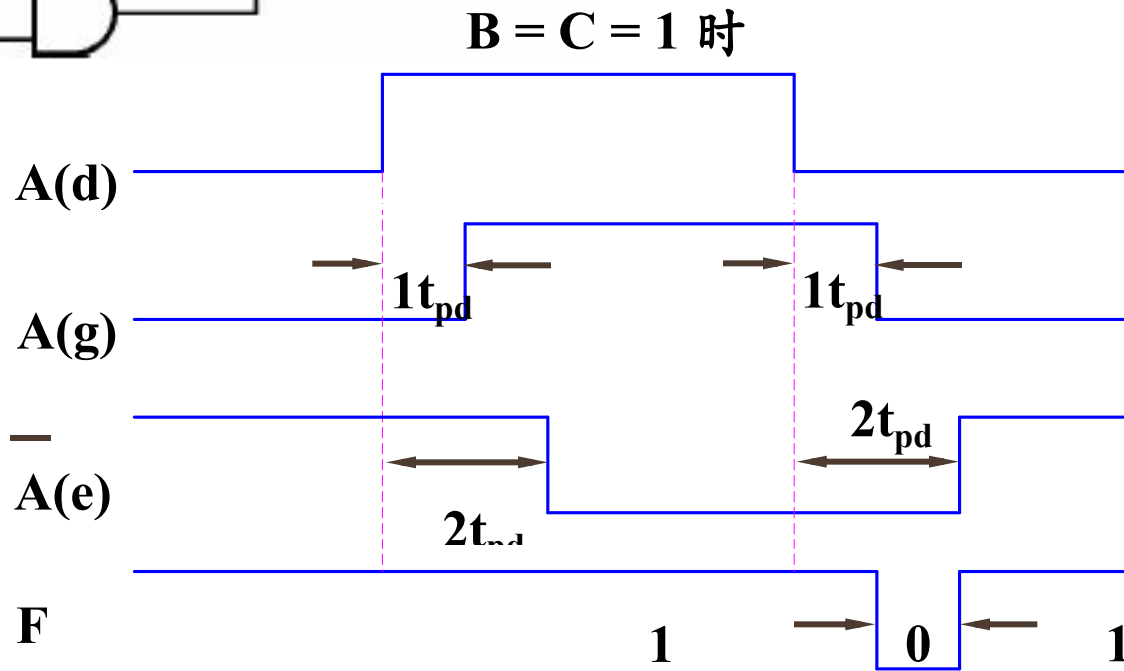
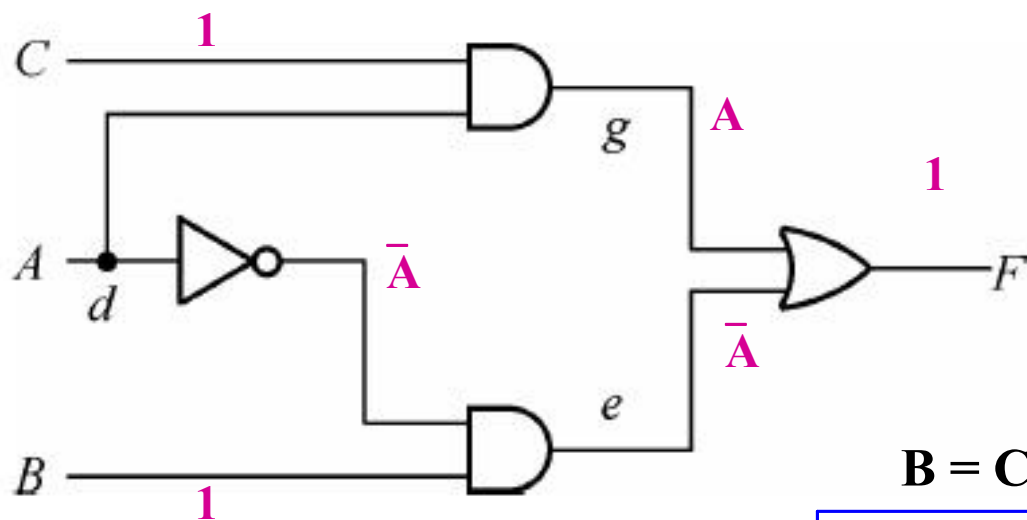
例1:  $F = AC + \bar{A}B$ 。



对于A信号变量而言，它在不同路径上传输，传输到达输出或门的输入端的时间不一样，而这种现象就称为变量的竞争。

**冒险：**由于竞争的存在，使数字电路输出出现

非预期信号的现象。



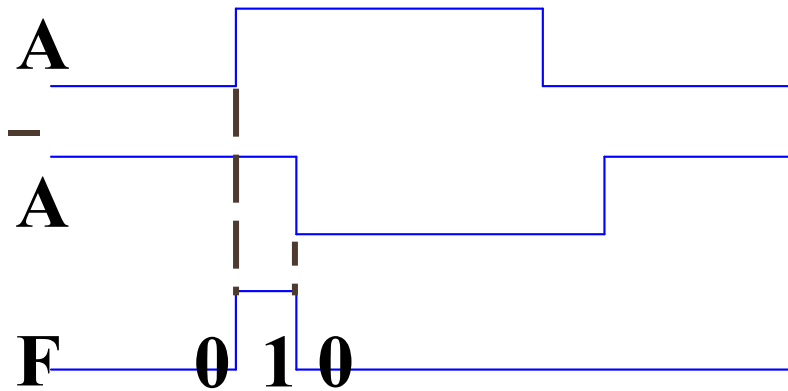
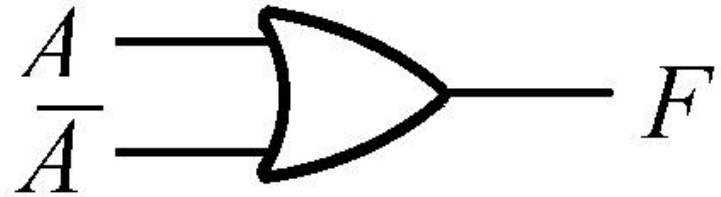
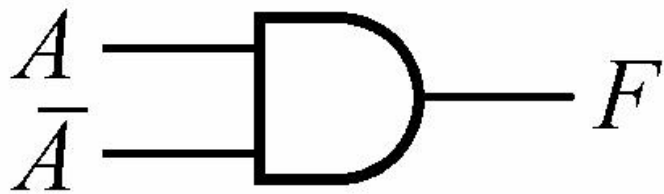
可看出冒险  
发生在输入信  
号动态变化的  
瞬间。

图 3.3.2 冒险的产生

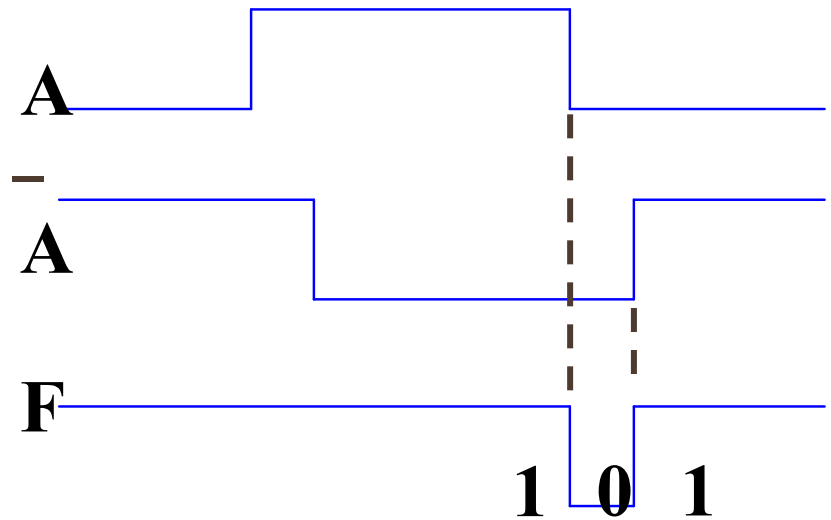


## 冒险的分类

### ① 按短暂尖峰极性



(a) 1型冒险



(b) 0型冒险

②按产生的原因分为逻辑冒险和功能冒险。

**逻辑冒险：**由于输入信号经过的路径不同而引起的冒险；

**功能冒险：**由于若干个信号同时变化，变化的快慢不同而引起的冒险。

## 3.3.2 冒险的判别方法

### 1. 逻辑冒险的判别

判断方法：代数法和卡诺图法

例1：用代数法判断是否存在逻辑冒险：

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

解：A是具有竞争条件的变量。

B	C	Y
0	0	0
0	1	$\bar{A}$
1	0	A
1	1	$A + \bar{A}$

∴当 $B = C = 1$ 时， $Y = A + \bar{A}$  —— 存在“0”型冒险

例2：用代数法判断电路是否存在逻辑冒险现象。

$$Y = AC + AB + \bar{A}C$$

解：A和C是具有竞争条件的变量。

B	C	Y
0	0	$\bar{A}$
0	1	A
1	0	$\bar{A}$
1	1	$A + \bar{A}$

A	B	Y
0	0	$\bar{C}$
0	1	1
1	0	C
1	1	C

∴当 $B = C = 1$ 时， $Y = A + \bar{A}$

变量A存在"0"型冒险

变量C不存在冒险现象。

例3：判断 $F = (A+C) \cdot (B + \bar{C})$ 是否存在逻辑冒险。

解：当 $A=B=0$ 时， $F=C\bar{C}$ ，

在C发生跳变时，可能出现“1”型冒险。

## 判断方法:

1. 观察表达式是否某个变量同时以原、反变量的形式存在。
2. 若有则将其余变量取固定值，是否能得到  $F=A+\bar{A}$  或  $F=A\bar{A}$  的形式，若有，则可能存在逻辑冒险。

**结论:** 与或表达式得到“0”型冒险；  
或与表达式得到“1”型冒险。

## 2) 卡诺图法

例：判断 $F=AC+B\bar{C}$ 是否存在逻辑冒险？

	<b>BC</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
<b>A</b>	<b>0</b>				1
<b>1</b>		1	1	1	

**结论：**取值在两个相切卡诺圈变化时，逻辑冒险可能发生。

**判断方法：**只要有两个存在部分相切，且相切部分没有被另外的卡诺圈包围，则可能存在冒险。  
冒险发生在其中一个卡诺圈中的最小项对应的取值变为另一卡诺圈中任一最小项对应取值的时刻。

例:  $F = B\bar{C} + \bar{A}\bar{B} + ACD$ 。试判断是否存在逻辑冒险。

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

在  $AC = 00$  时，或  
 在  $BCD = 011$  时，或  
 在  $ABD = 111$  时，  
 存在0型逻辑冒险。



## 2. 功能冒险的判别

例：判断  $F=AC+B\bar{C}$  是否存在功能冒险？

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

$111 \rightarrow \underline{110}$  (C先于A变化)  $\rightarrow 010$  对应的输出函数值为  $1 \rightarrow 1 \rightarrow 1$ 。

$111 \rightarrow \underline{011}$  (A先于C变化)  $\rightarrow 010$  对应的输出函数值为  $1 \rightarrow 0 \rightarrow 1$ ，出现了“0”型冒险。

## 判断方法:

1. 输入变量变化前后，函数值相同；
2. 有  $p$  ( $\geq 2$ ) 个变量同时变化；
3. 不变的  $(n-p)$  个输入变量组成的乘积项所对应的卡诺圈中，有“1”也有“0”。

则电路中存在功能冒险。

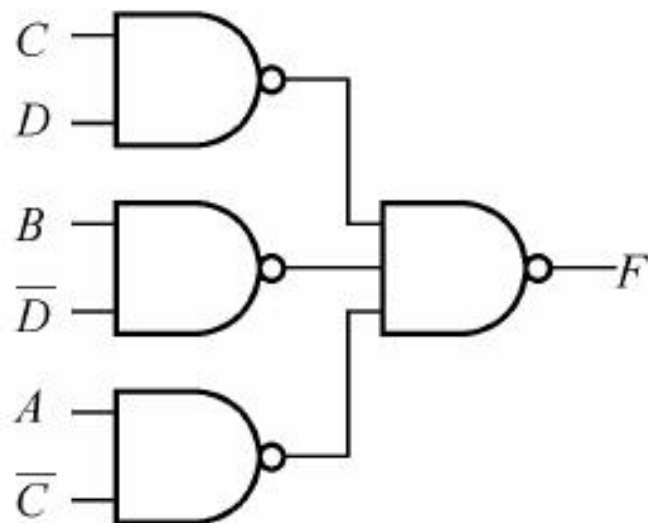
例6 已知  $F = A C + B \bar{C} + \bar{A} \bar{C}$ ，当ABC从101变为110时，是否会出现功能冒险。

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

1.  $F(1,0,1) = F(1,1,0)$ ;
2. B和C 2个变量同时变化;
3. A对应的卡诺圈中有“0”也有“1”。

所以，电路中存在功能冒险。

例7：分析下图所示组合电路，当输入信号ABCD从0100变化到1101、从0111变化到1110以及从1001变化到1011时，是否由冒险现象发生。



解：该组合逻辑电路的逻辑函数表达式为：

$$F = CD + B \bar{D} + A \bar{C}$$

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

$$F = CD + B \bar{D} + A \bar{C}$$

(1) 0100 → 1101

可能存在0型功能冒险

不存在逻辑冒险

(2) 0111 → 1110

不存在功能冒险,可能存在0型逻辑冒险

(3) 1001 → 1011

可能存在0型逻辑冒险,不存在功能冒险。

### 3.3.3 冒险的消除方法

#### 1) 增加多余项，可消除逻辑冒险

在卡诺圈内部取值变化，不会存在逻辑冒险，因此可以用增加多余卡诺圈的方法，消除逻辑冒险。

例8：  $F=AC+B\bar{C}+\bar{A}\bar{C}$ ，试用增加多余项的方法消除逻辑冒险。

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

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

例9  $F = B\bar{C} + \bar{A}\bar{B} + ACD$ 。试用增加多余项的方法消除逻辑冒险。

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

$$F = B\bar{C} + \bar{A}\bar{B} + ACD + \bar{A}\bar{C} + ABD + \bar{B}CD$$

## 2) 加滤波电容 (对输出波形边沿要求不高的情况下运用)

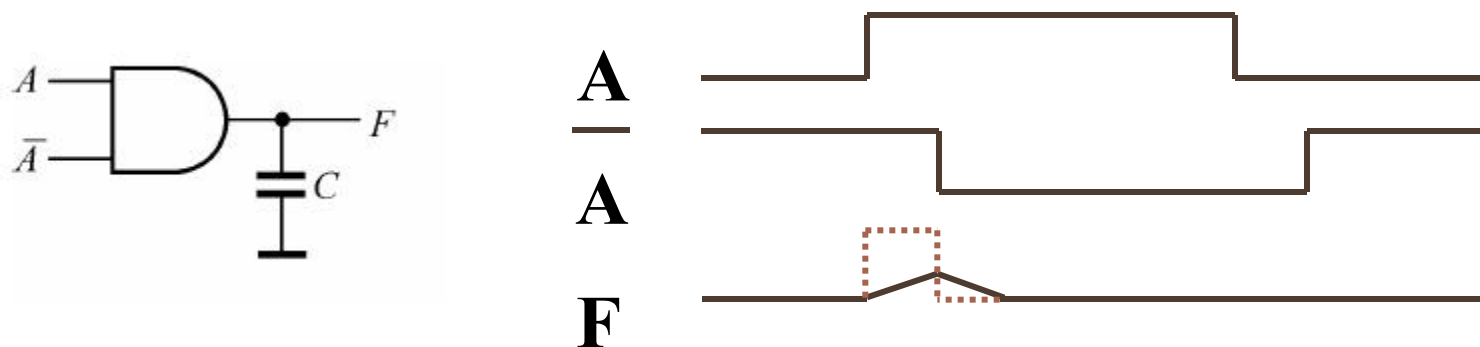
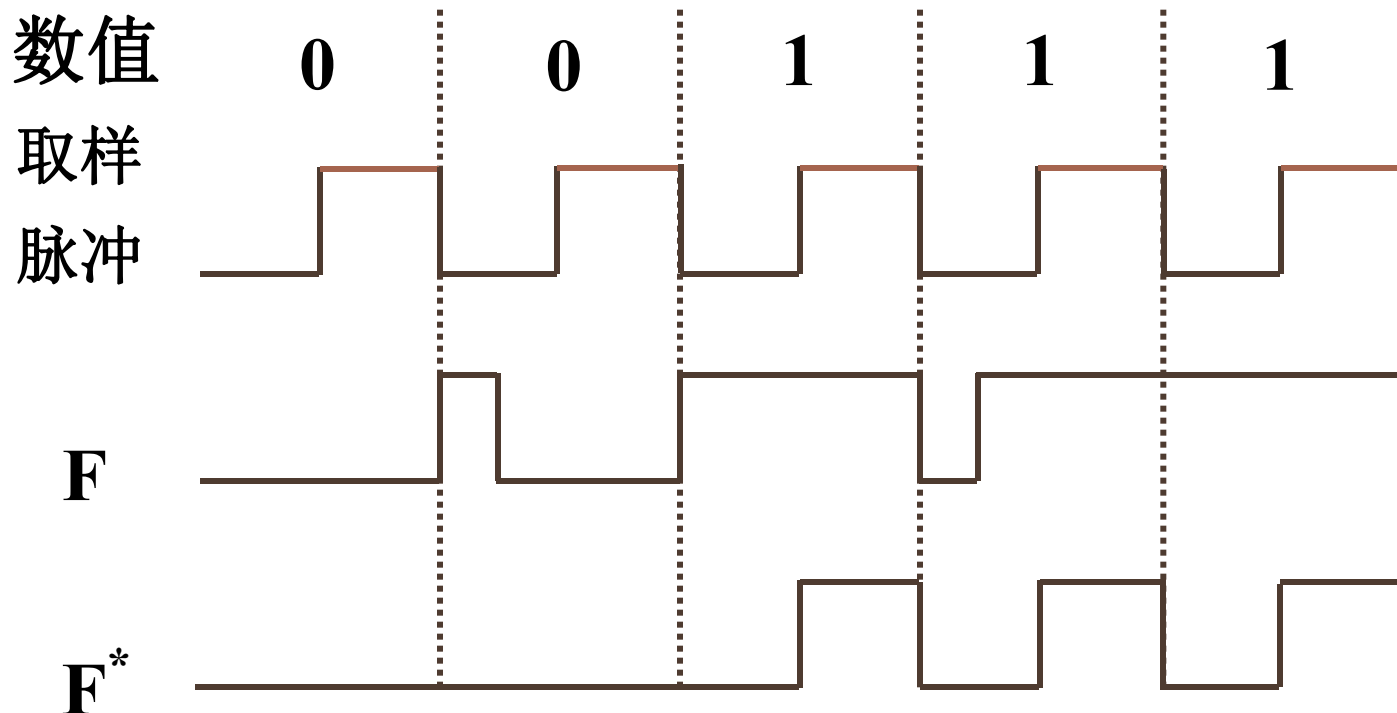


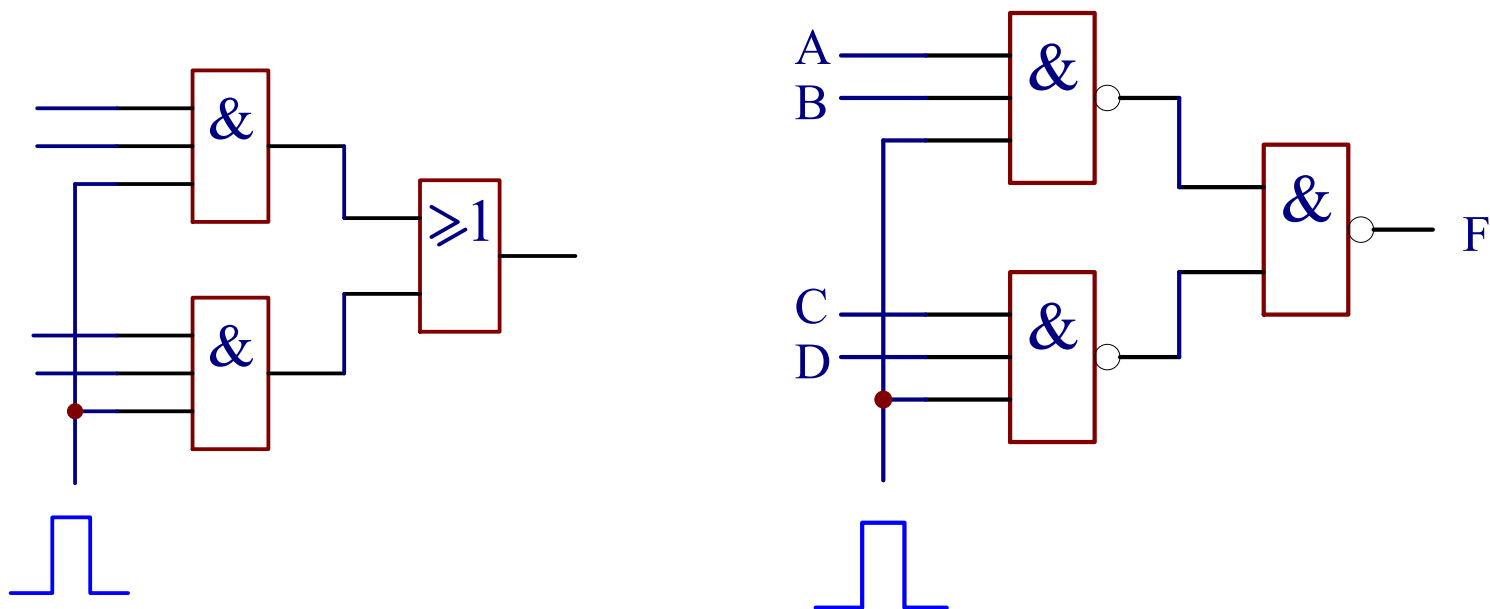
图 3.3.10 加电容消除冒险



## 3) 加取样脉冲

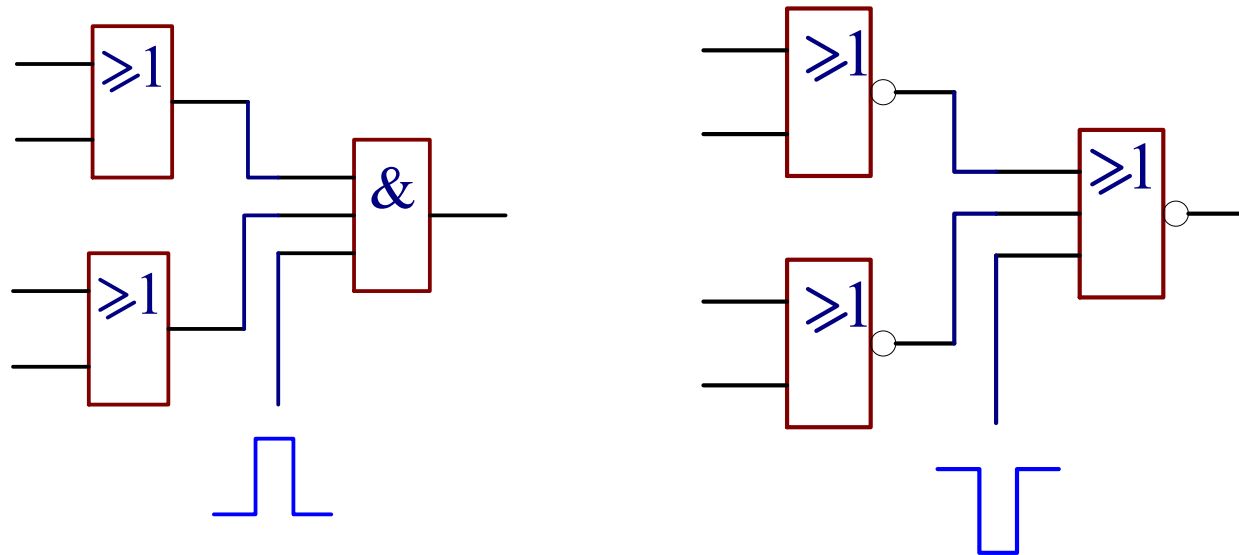


其中， $F$ 和 $F^*$  分别表示组合电路加取样脉冲之前、之后的输出。  $F^* = F \cdot CP$



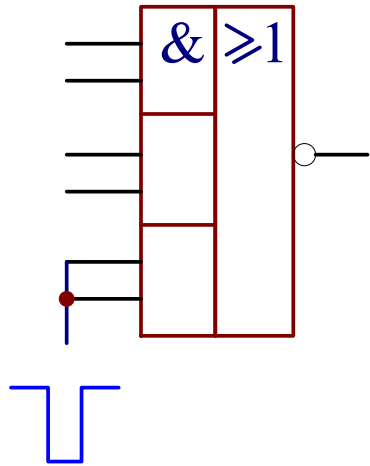
$$\begin{aligned}
 F^* &= F \cdot CP = (AB + CD)CP = AB \cdot CP + CD \cdot CP \\
 &= \overline{\overline{AB \cdot CP \cdot CD \cdot CP}}
 \end{aligned}$$

图 3.3.12 用取样法避免冒险



$$\begin{aligned}
 F^* &= F \cdot CP = (A + B) \cdot (C + D) \cdot CP \\
 &= \overline{\overline{A + B + C + D + CP}}
 \end{aligned}$$

图 3.3.12 用取样法避免冒险（续）



$$\begin{aligned}
 F^* &= F \cdot CP \\
 &= \overline{AB + CD} \cdot CP \\
 &= \overline{AB + CD + \overline{CP}}
 \end{aligned}$$

图 3.3.12 用取样法避免冒险（续）

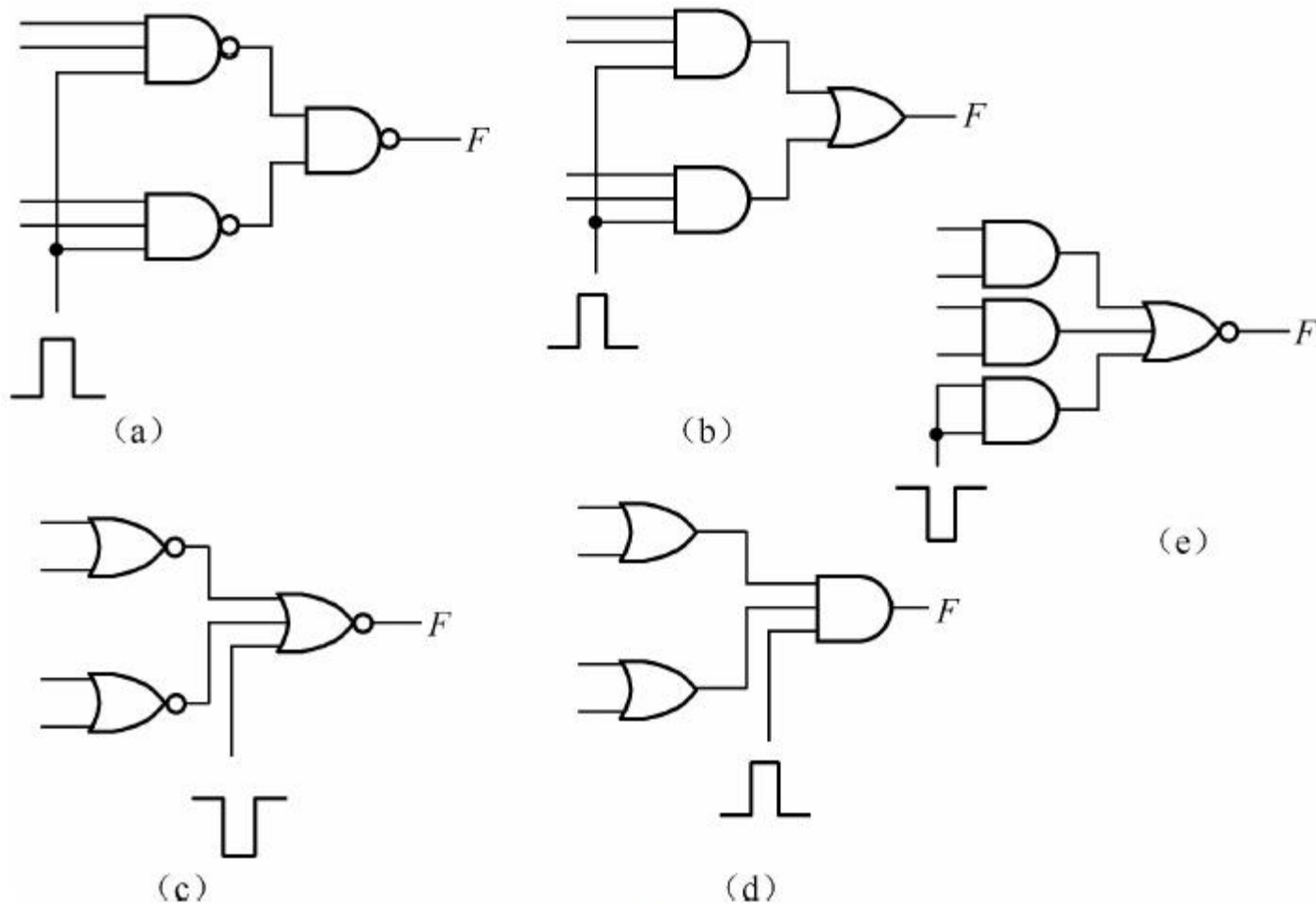
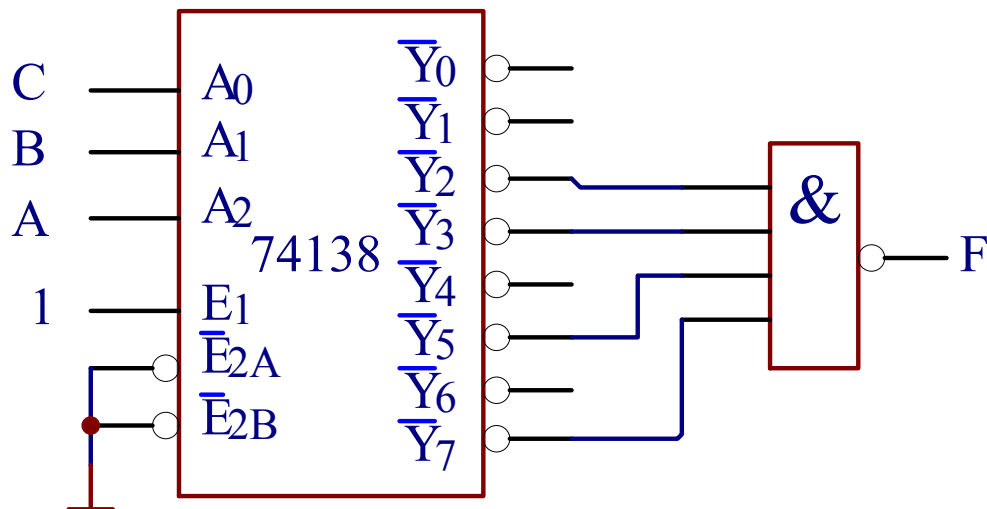


图 3.3.12 常用组合电路加取样脉冲避免冒险的方法

例10 已知  $F = \bar{A}B + AC$ ，用74138实现该函数，电路图如下所示。试分析电路是否存在逻辑冒险，若存在，加取样脉冲避免之。



解： 
$$F = \overline{\overline{m_2} \overline{m_3} \overline{m_5} \overline{m_7}} = m_2 + m_3 + m_5 + m_7$$

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

由图可知，存在相切的卡诺圈，该电路可能存在0型逻辑冒险。

取样脉冲加在74138的使能端上，如下图所示：

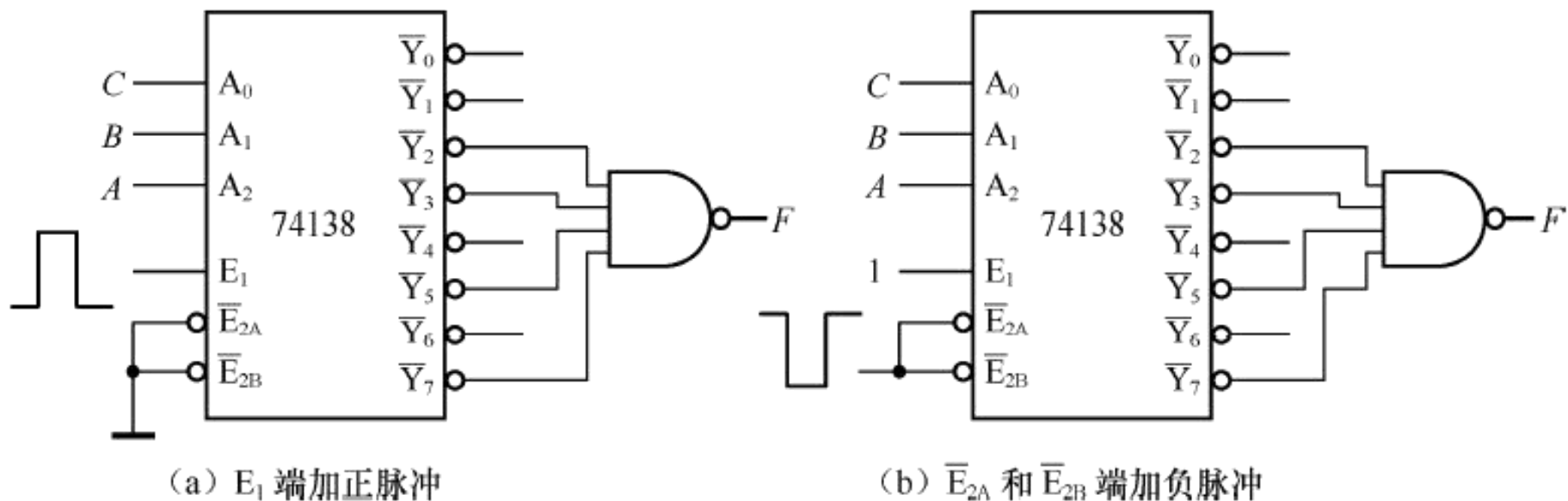


图 3.3.13 MSI 器件用使能端避免冒险

(a)

$$F^* = F \cdot CP = \overline{\overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_5} \cdot \overline{Y_6}} \cdot \left( E_1 \cdot \overline{\overline{E_{2A}} + \overline{E_{2B}}} \right) \cdot CP = \overline{\overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_5} \cdot \overline{Y_6}} \cdot \left( \overline{\overline{E_{2A}} + \overline{E_{2B}}} \right) \cdot CP$$

(b)

$$F^* = F \cdot CP = \overline{\overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_5} \cdot \overline{Y_6}} \cdot \left( E_1 \cdot \overline{\overline{E_{2A}} + \overline{E_{2B}}} \right) \cdot CP$$

$$= \overline{\overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_5} \cdot \overline{Y_6}} \cdot \overline{\overline{E_{2A}} + \overline{E_{2B}} + \overline{CP}} = \overline{\overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_5} \cdot \overline{Y_6}} \cdot E_1 \cdot \overline{\overline{CP}}$$