

# 数字电路

## 第5章 组合逻辑电路



杨旭

北京理工大学

`pyro_yangxu@bit.edu.cn`

# 本章内容

---

- ❑ 5.1 概述
- ❑ 5.2 组合逻辑电路的分析
- ❑ 5.3 组合逻辑电路的设计
- ❑ 5.4 组合逻辑电路中的竞争和冒险
- ❑ 5.5 组合电路的系统应用
- ❑ 5.6 用Multisim2001设计和分析组合电路



# 重点

---

组合逻辑电路分析

组合逻辑电路的设计



# 组合逻辑电路和时序逻辑电路

---

## 组合逻辑电路

由逻辑门组成

内部无反馈环节

不具有“记忆”功能

输出仅由当前时刻的输入决定

## 时序逻辑电路

有反馈环节，具有“记忆”功能，输出和之前的状态有关。



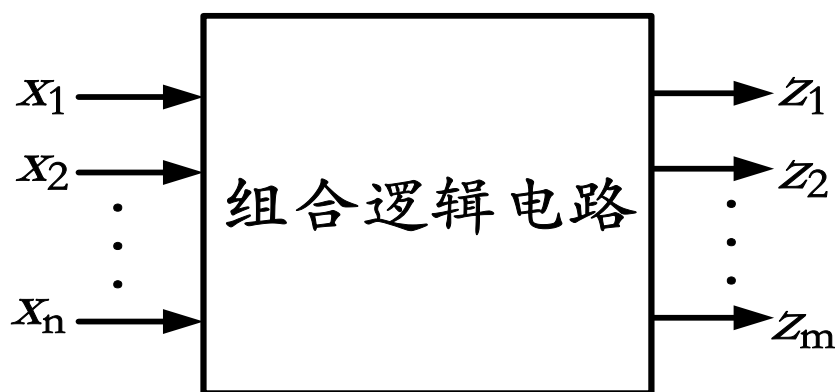
# 4.1 概述

组合电路是由逻辑门组成的多输入、多输出（或单输出）的逻辑电路。

如：一个组合电路

$n$ 个输入 ( $x_1 \sim x_n$ )

$m$ 个输出 ( $z_1 \sim z_m$ )



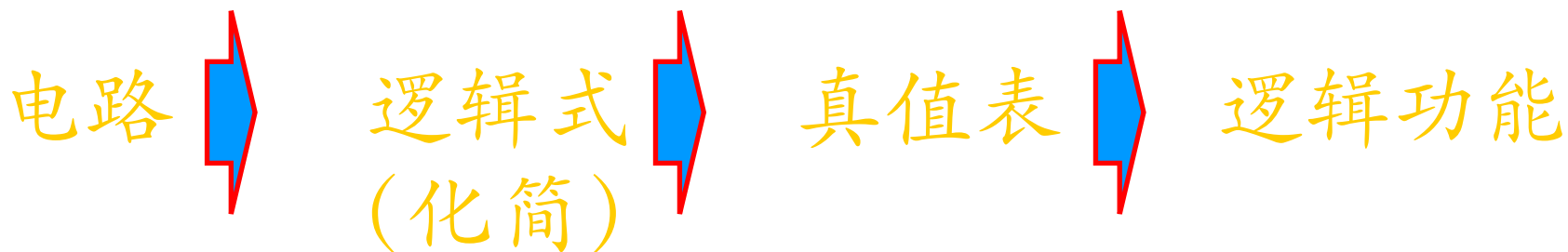
$$\begin{cases} z_1 = f_1(x_1, x_2, \dots, x_n) \\ z_2 = f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ z_m = f_m(x_1, x_2, \dots, x_n) \end{cases}$$



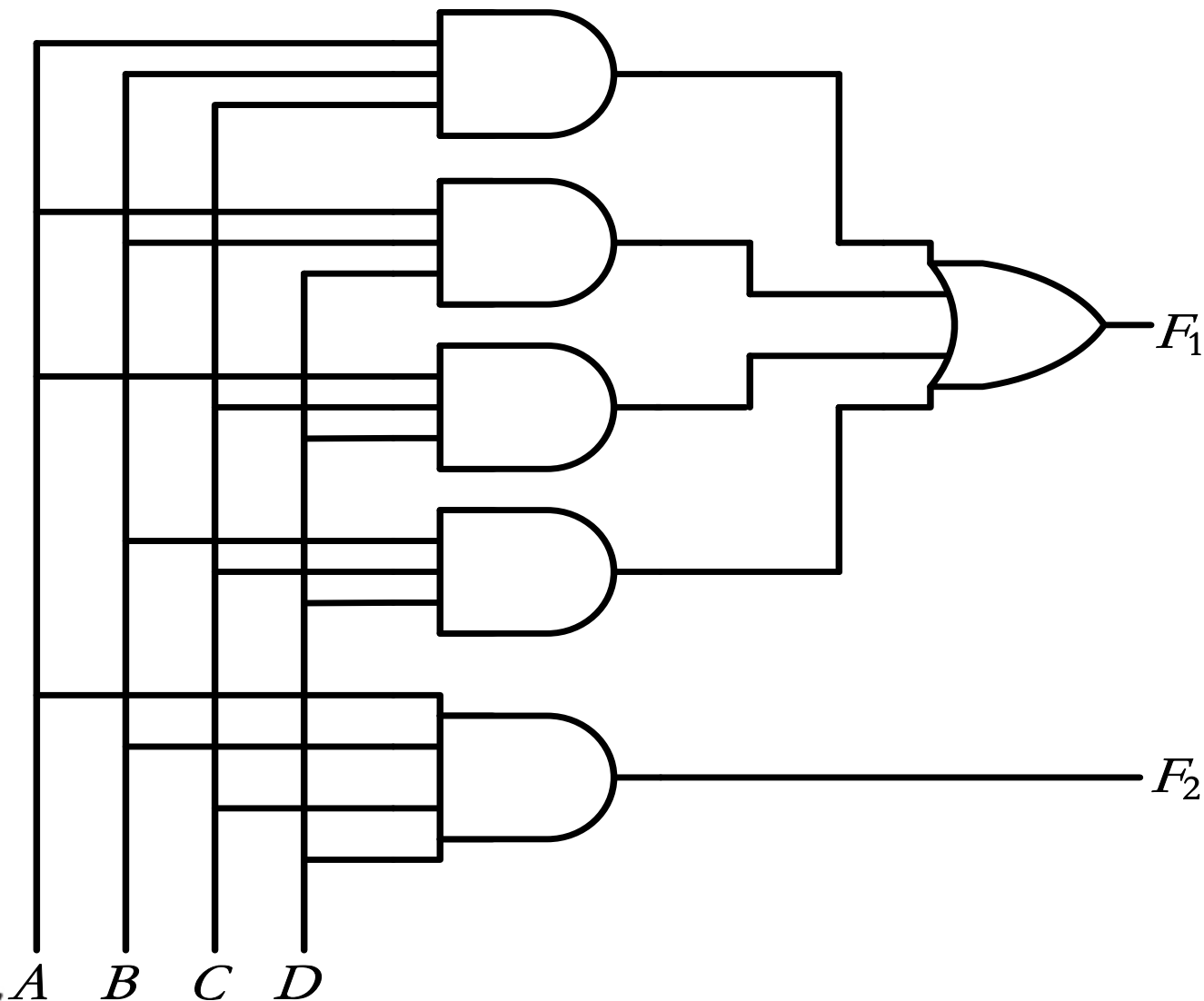
## 4.2 组合逻辑电路的分析

根据已知逻辑电路，利用逻辑表达式、真值表等，  
分析其逻辑功能。

### 组合电路分析的一般步骤



## 例：分析某四输入、两输出的逻辑电路



第一步：写表达式

第二步：列真值表

第三步：分析功能

输 入				输 出	
$A$	$B$	$C$	$D$	$F_1$	$F_2$
0	1	1	1	1	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	1	1
其 他				0	0

$ABCD$ 中：

$$F_1 = \overset{\text{多数为1时, } F_1=1;}{ABC} + \overset{\text{全部为1时, } F_2=1。}{ABD} + ACD + BCD$$

表决电路：

$$F_2 = \overset{\text{多数通过和一致通过}}{ABCD}$$

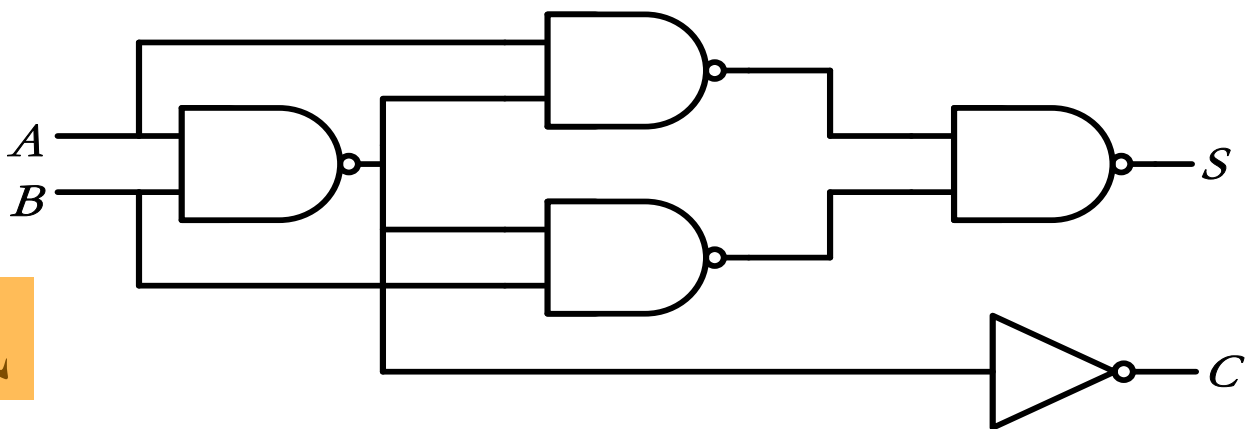




# 常用组合电路及其分析

## 1 加法器

由5个逻辑门组成的2输入、2输出逻辑电路如图



写逻辑表达式

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

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



## 列真值表

$$S = A \oplus B$$

$$C = AB$$

## 分析功能

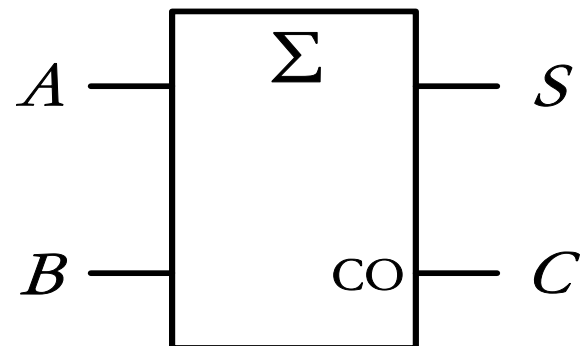
输入		输出	
<i>A</i>	<i>B</i>	<i>S</i>	<i>C</i>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

两个1位二进制数相加，其当*A*和*B*相加时，因没有低位进位，只考虑本位和*S*和进位*C*。

这种加法运算称为“**半加**”运算，实现半加运算的电路称为“**半加器**”。

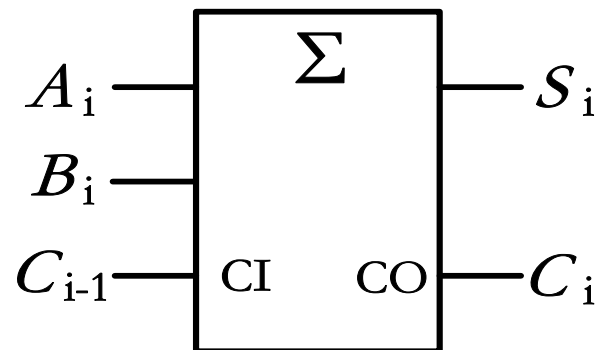


半加器逻辑符号如图



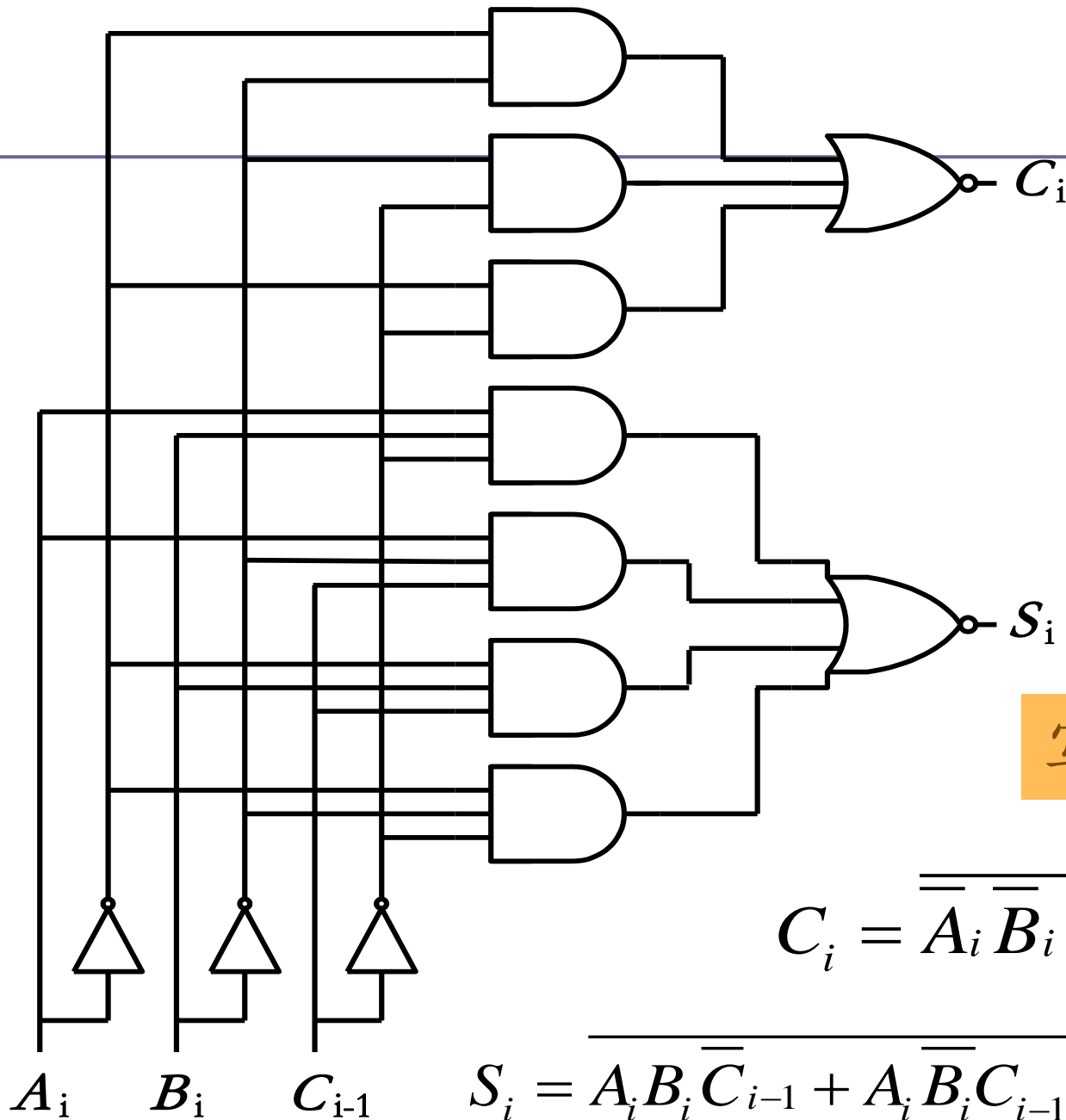
两个二进制数相加时，  
还需要考虑低位的进位，  
称为“**全加**”运算。

实现全加运算的电路称为“**全加器**”



这是一个由12个门组成的3输入、2输出组合逻辑电路。

写逻辑表达式



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

$$S_i = \overline{A_i B_i \overline{C_{i-1}}} + \overline{A_i \overline{B_i} C_{i-1}} + \overline{\overline{A_i} B_i C_{i-1}} + \overline{\overline{A_i} \overline{B_i} \overline{C_{i-1}}}$$



列真值表

分析功能

符合全加运算的规律，所以该电路为全加器。

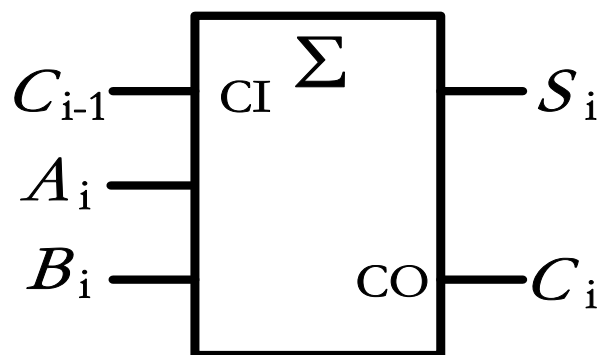
输入			输出	
$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

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

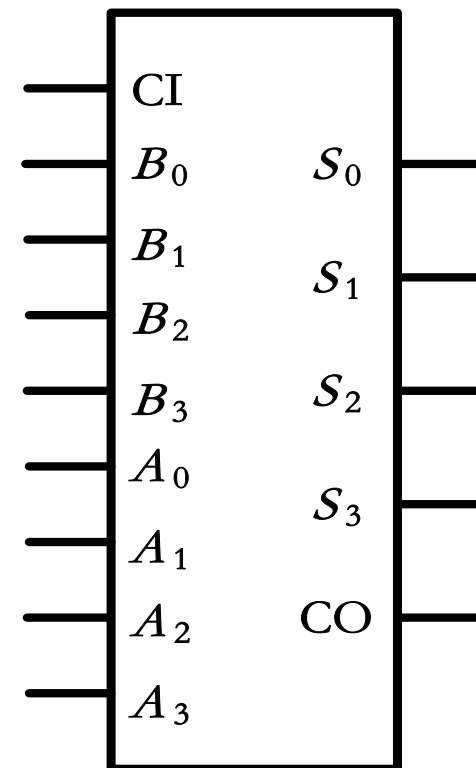
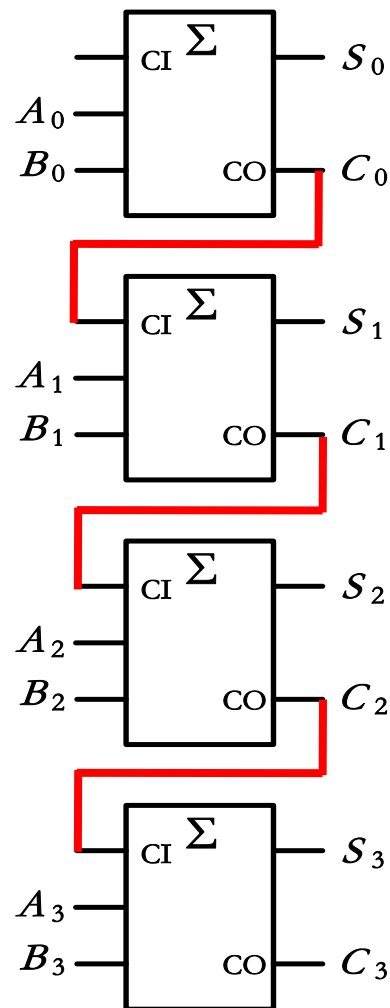
$$S_i = A_i B_i \overline{C_{i-1}} + A_i \overline{B_i} C_{i-1} + \overline{A_i} B_i C_{i-1} + \overline{A_i} \overline{B_i} \overline{C_{i-1}}$$



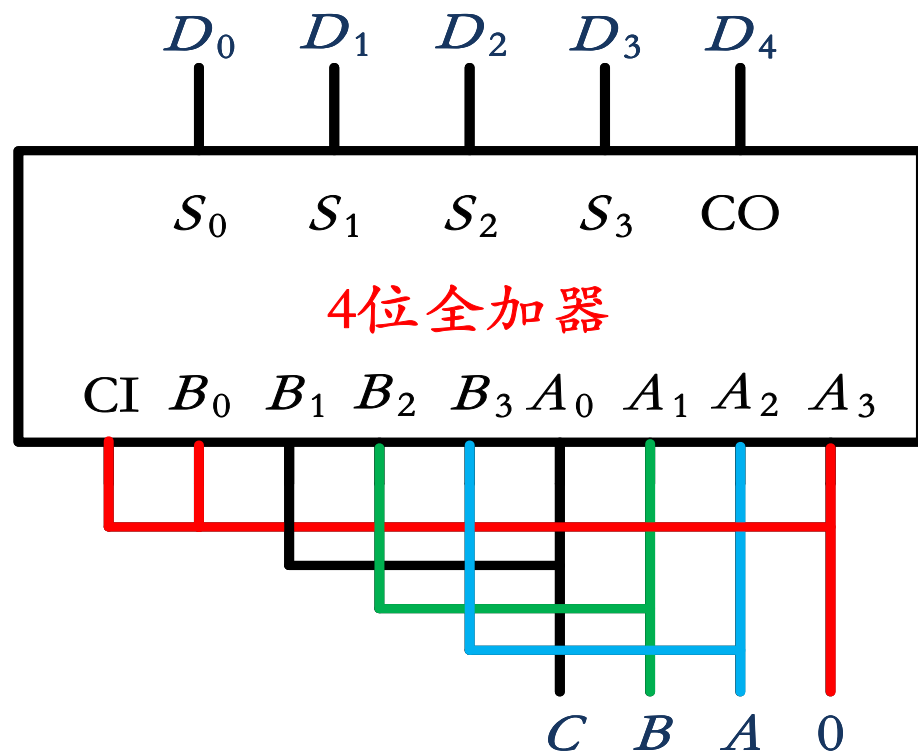
# 全加器的应用—组成多位全加器



组成四位全加器



# 4位全加器应用电路



$$\begin{array}{r}
 0 \ A \ B \ C \\
 + \ A \ B \ C \ 0 \\
 \hline
 D_4 \ D_3 \ D_2 \ D_1 \ D_0
 \end{array}$$

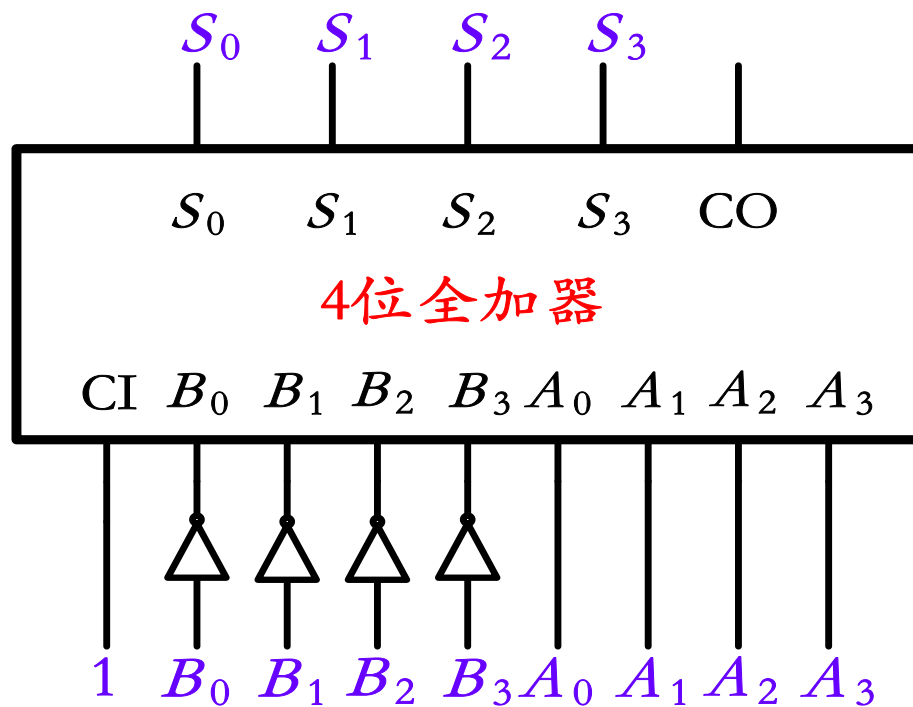
输入			输出				
A	B	C	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	1
0	1	0	0	0	1	1	0
0	1	1	0	1	0	0	1
1	0	0	0	1	1	0	0
1	0	1	0	1	1	1	1
1	1	0	1	0	0	1	0
1	1	1	1	0	1	0	1

当输入为000~111时，输出为相应输入

值的3倍

“×3” 电路



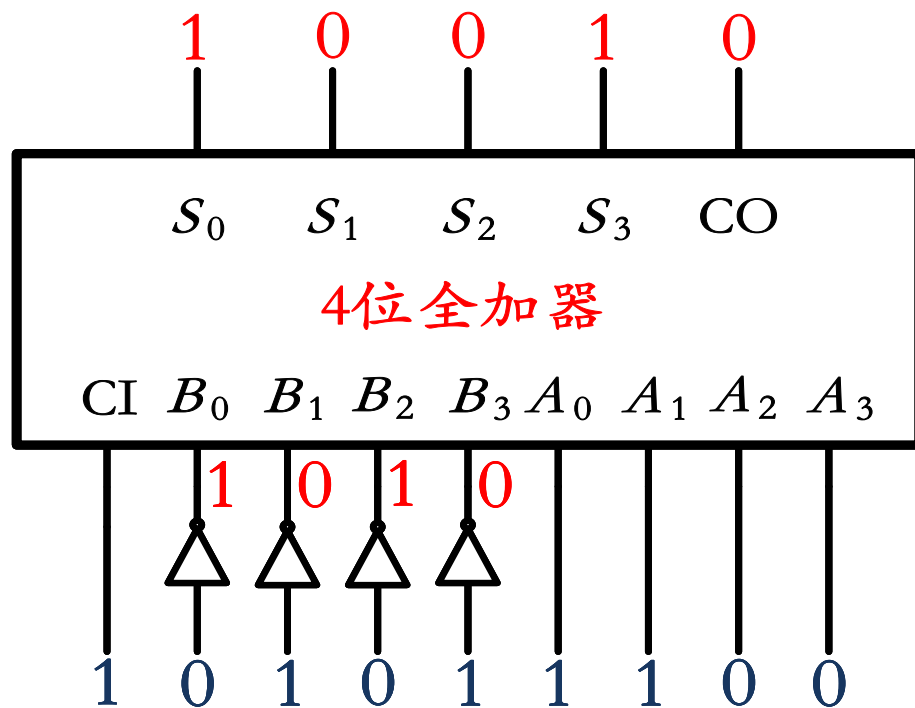


$A$ 加 $B$ 的补码 (反码+1)  
 相当于进行 $A$ 减 $B$ 的运算。

$$\begin{aligned}
 S_3 S_2 S_1 S_0 &= A_3 A_2 A_1 A_0 + \overline{B_3} \overline{B_2} \overline{B_1} \overline{B_0} + 1 \\
 &= A_3 A_2 A_1 A_0 - B_3 B_2 B_1 B_0
 \end{aligned}$$







即

$$\begin{array}{r}
 \phantom{+} 0011 \\
 + 0101 \\
 \hline
 1000 \\
 + \phantom{000} 1 \\
 \hline
 1001 \\
 \\
 \phantom{+} 0011 \\
 - 1010 \\
 \hline
 1001
 \end{array}$$



---

练习：利用4位全加器和所需门电路，设计一个受M控制的电路。

当 $M=0$ 时，实现两个四位数的加法运算，

当 $M=1$ 时，实现两个四位数的减法运算。



四位全加器的输入与输出之间关系为

$$S_3S_2S_1S_0 = A_3A_2A_1A_0 + B_3B_2B_1B_0 + CI$$

设两个四位二进制数分别为 $C_3C_2C_1C_0$ 和

$D_3D_2D_1D_0$ ，输出为 $S_3S_2S_1S_0$



$$S_3 S_2 S_1 S_0 = A_3 A_2 A_1 A_0 + B_3 B_2 B_1 B_0 + \textcolor{red}{CI}$$


---

M=0 时

$$B_3 B_2 B_1 B_0 = \overline{M} (D_3 D_2 D_1 D_0)$$

$$S_3 S_2 S_1 S_0 = C_3 C_2 C_1 C_0 + D_3 D_2 D_1 D_0 + 0$$

M=1 时

$$B_3 B_2 B_1 B_0 = M (\overline{D_3} \overline{D_2} \overline{D_1} \overline{D_0})$$

$$S_3 S_2 S_1 S_0 = C_3 C_2 C_1 C_0 + \overline{D_3} \overline{D_2} \overline{D_1} \overline{D_0} + 1$$

$$\textcolor{red}{CI} = M$$

$$A_3 A_2 A_1 A_0 = C_3 C_2 C_1 C_0$$

$$B_3 B_2 B_1 B_0 = M \oplus (D_3 D_2 D_1 D_0)$$

$$A_3 = C_3$$

$$A_2 = C_2$$

$$A_1 = C_1$$

$$A_0 = C_0$$

$$CI = M$$

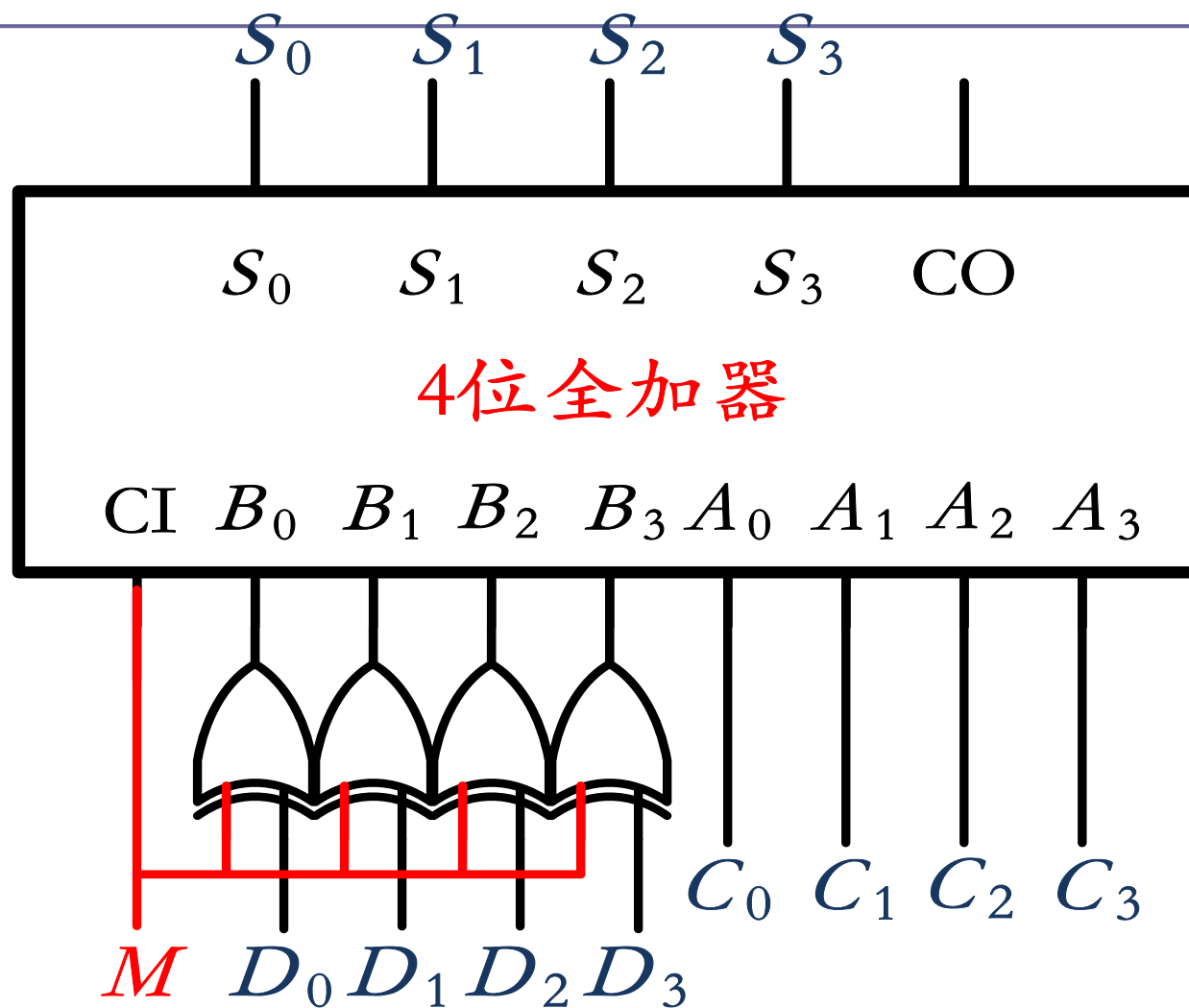
$$B_3 = M \oplus D_3$$

$$B_2 = M \oplus D_2$$

$$B_1 = M \oplus D_1$$

$$B_0 = M \oplus D_0$$





$$A_3 = C_3$$

$$A_2 = C_2$$

$$A_1 = C_1$$

$$A_0 = C_0$$

$$CI = M$$

$$B_3 = M \oplus D_3$$

$$B_2 = M \oplus D_2$$

$$B_1 = M \oplus D_1$$

$$B_0 = M \oplus D_0$$



# Data Selector Multiplexer

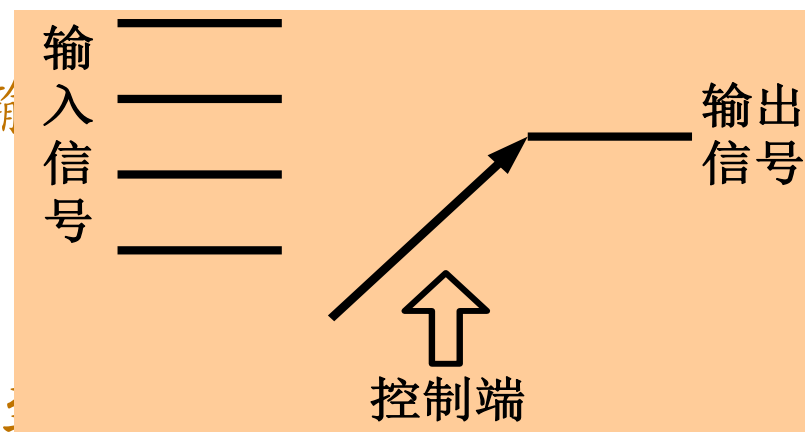
## 2 数据选择器

数据选择器又称为多路开关、多路转换器。

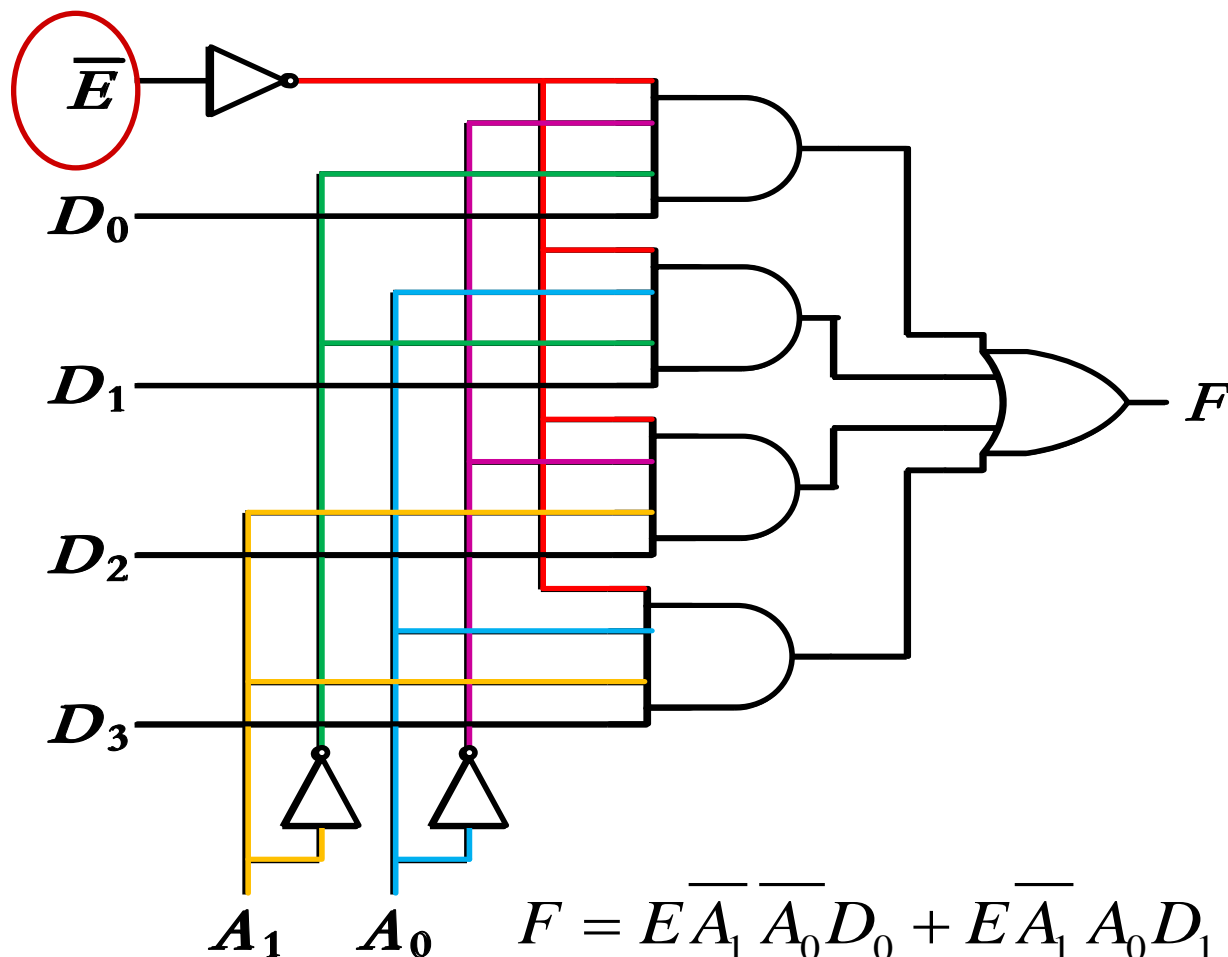
### 功能

在控制信号作用下，从多个输入信号到输出。

如从4路信号 $D_0 \sim D_3$ 中选一个到输出。  
数据选择器。



# 原理分析



写逻辑表达式

$$\overline{E} \overline{A_1} \overline{A_0} D_0$$

$$\overline{E} \overline{A_1} A_0 D_1$$

$$E A_1 \overline{A_0} D_2$$

$$E A_1 A_0 D_3$$

$$F = \overline{E} \overline{A_1} \overline{A_0} D_0 + \overline{E} \overline{A_1} A_0 D_1 + E A_1 \overline{A_0} D_2 + E A_1 A_0 D_3$$

$$= E \left( \overline{A_1} \overline{A_0} D_0 + \overline{A_1} A_0 D_1 + A_1 \overline{A_0} D_2 + A_1 A_0 D_3 \right)$$



$$F = E \left( \overline{A_1} \overline{A_0} D_0 + \overline{A_1} A_0 D_1 + A_1 \overline{A_0} D_2 + A_1 A_0 D_3 \right)$$

$\overline{E}$  为使能端，表示低电平有效。

列真值表

分析逻辑功能

$\overline{E}$  为选通端、低电平有效。

控制端  $A_1 A_0$  为 00、01、10、11 时，分别选中  $D_0$ 、 $D_1$ 、 $D_2$ 、 $D_3$  到输出  $F$ 。

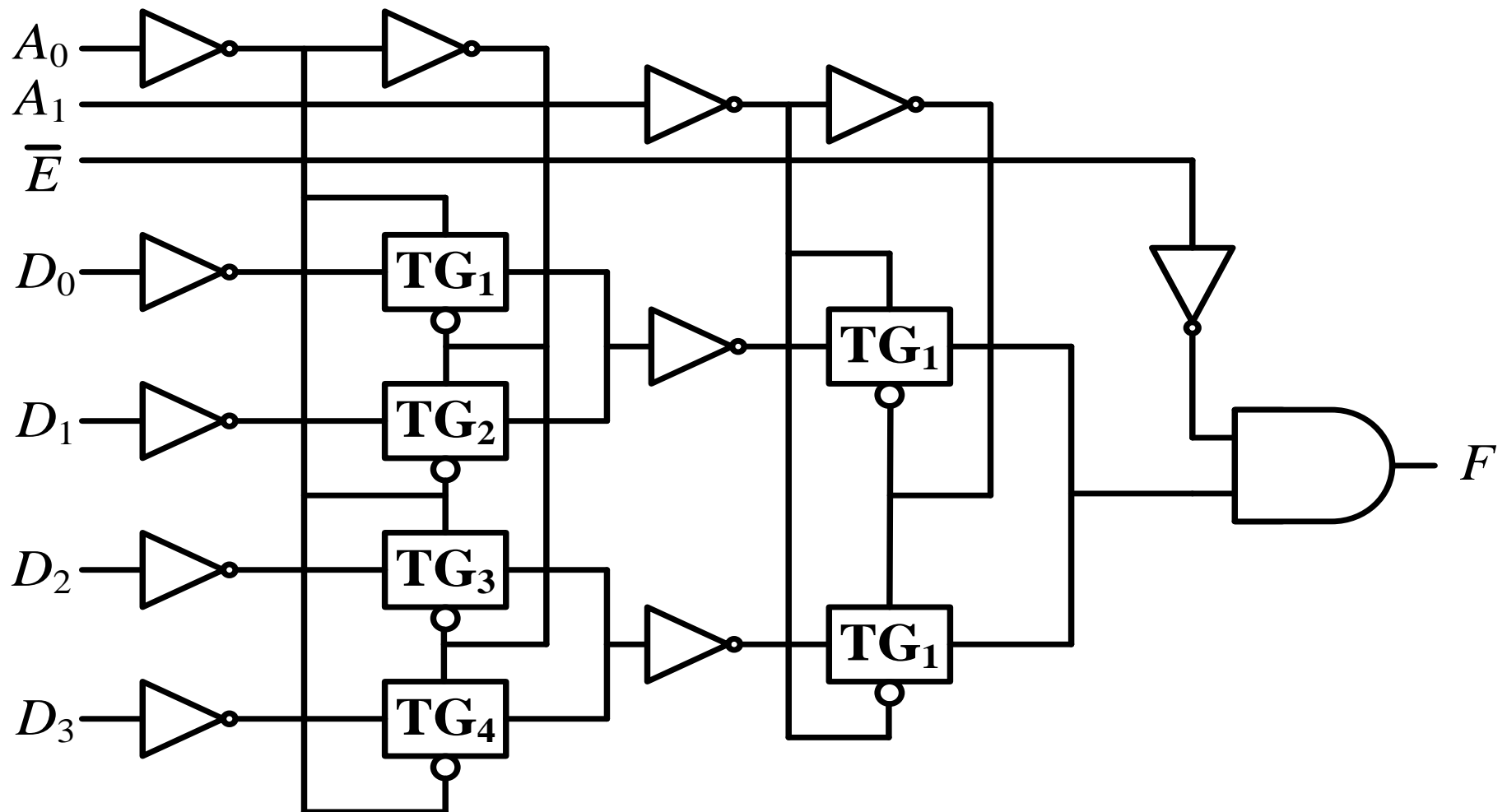
4选1数据选择器。

输入			输出
$\overline{E}$	$A_1$	$A_0$	$F$
1	任意		0
0	0	0	$D_0$
0	0	1	$D_1$
0	1	0	$D_2$
0	1	1	$D_3$

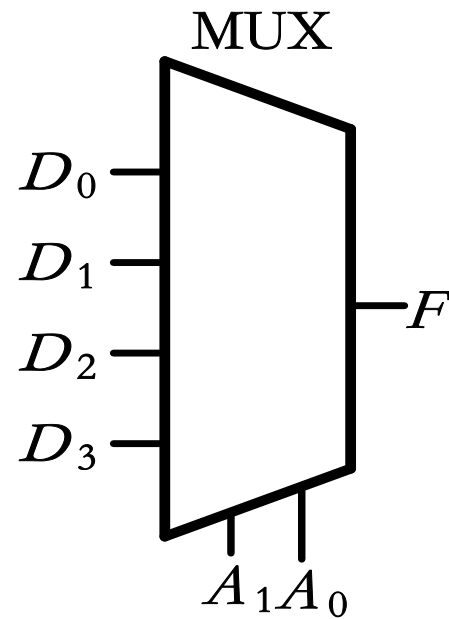
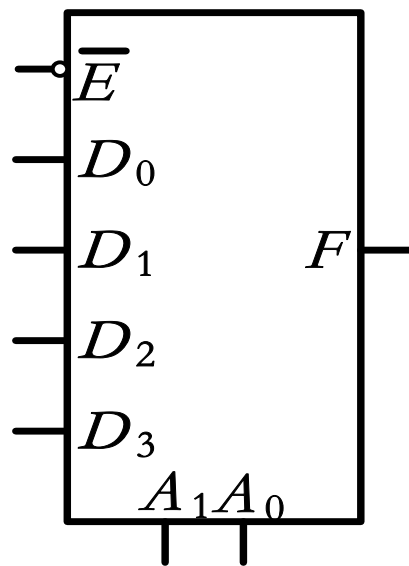


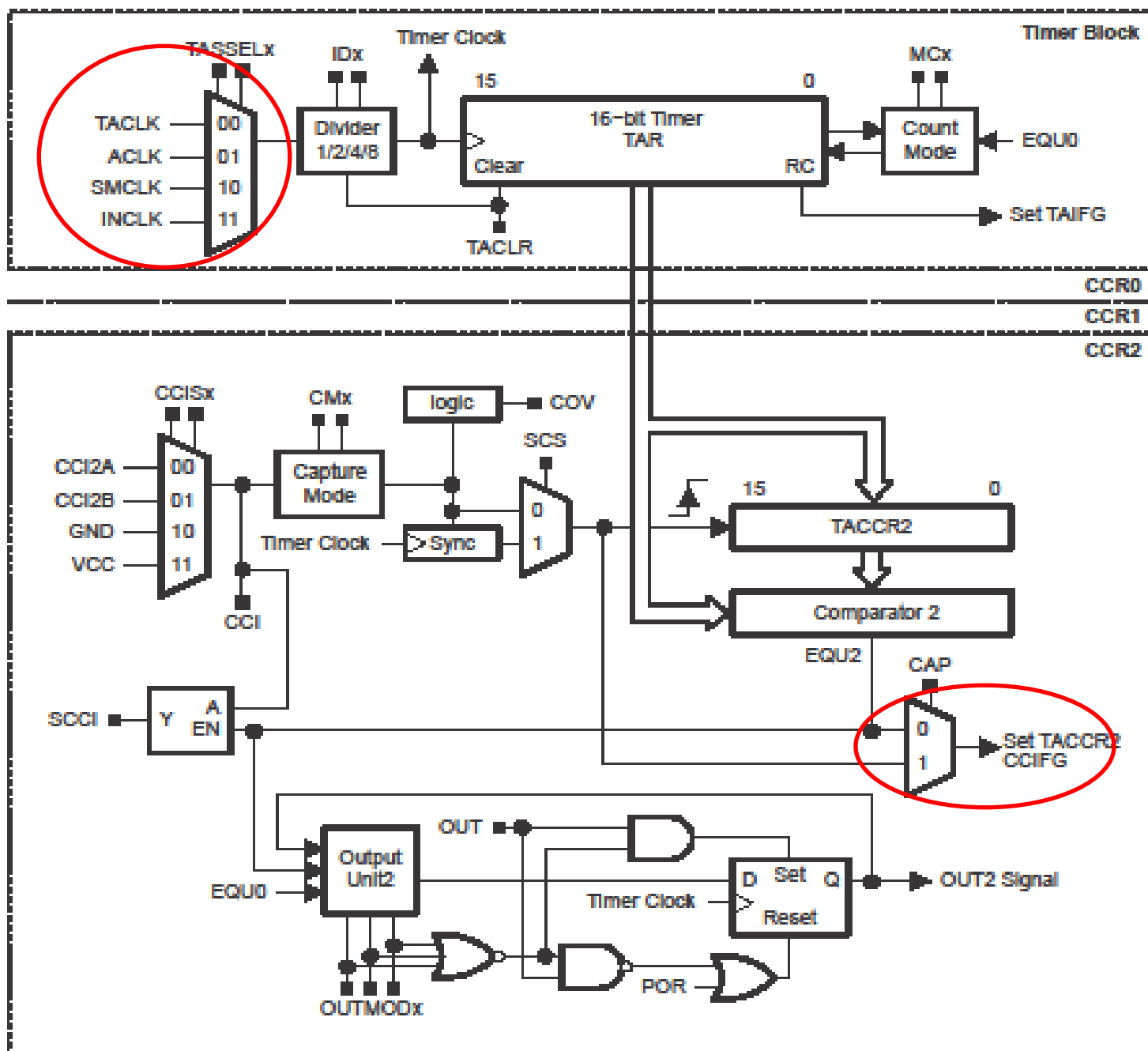


## 由传输门构成的4选1数据选择器



# 数据选择器逻辑符号





# 选择器的自扩展

---

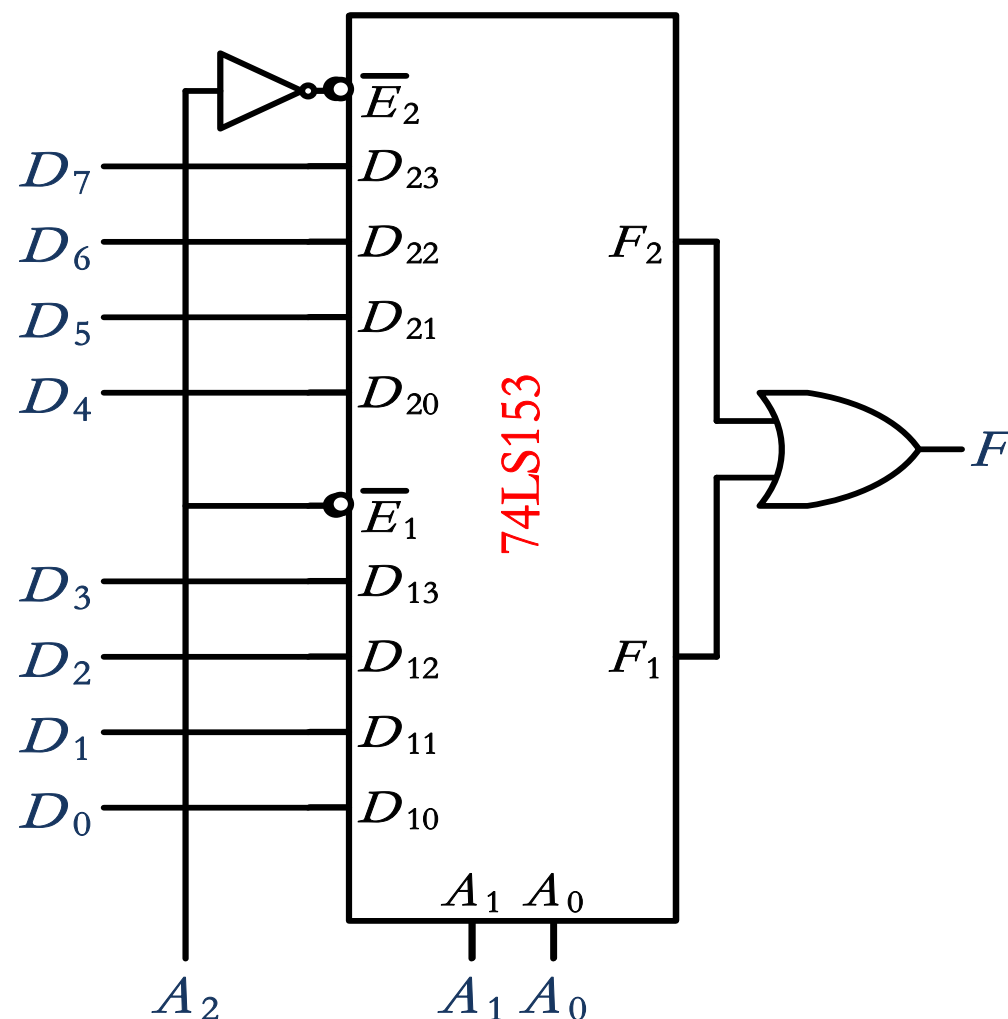
数据选择器的自扩展就是用多片某类选择器构成更大选择范围的选择器。

如利用两个4选1数据选择器实现从8个输入信号中进行选择的要求（构成8选1数据选择器）。



74153内部有两个独立的4选1数据选择器。

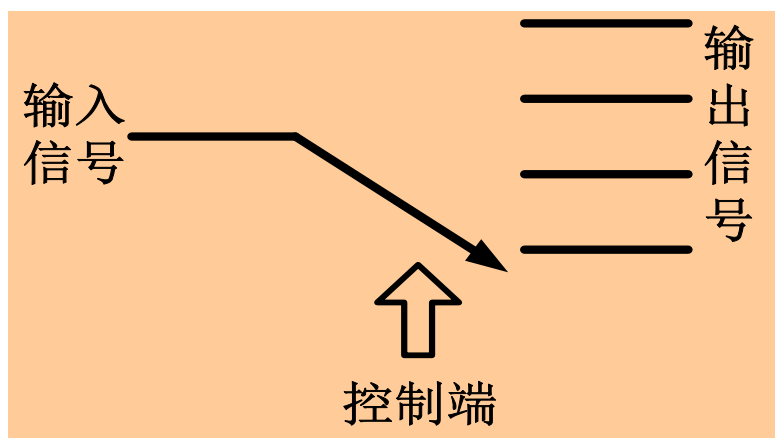
利用扩展端 $A_2$ 控制两个选通端，在 $A_2$ 为0、1时各有一个选择器工作，实现8选1选择器的功能。



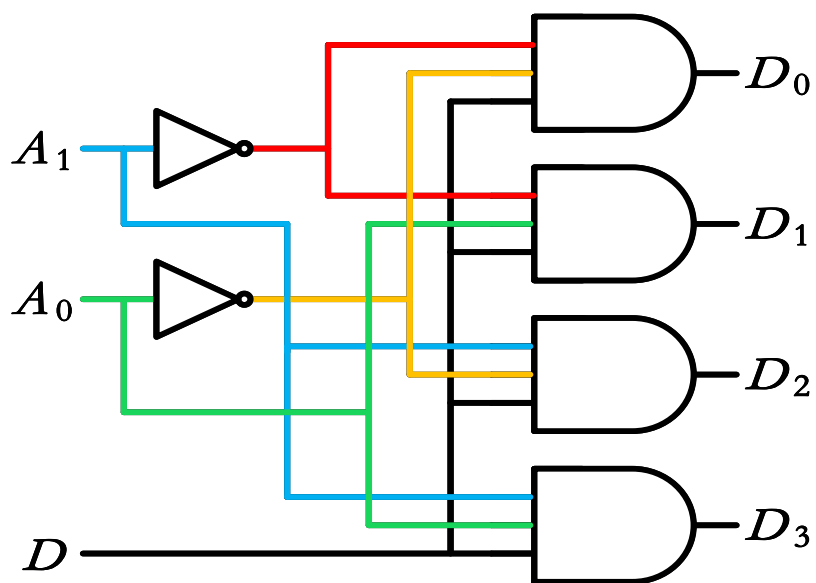
### 3 多路分配器

#### 功能

与数据选择器的功能相反，多路分配器可以在通道选择端的作用下，将一个数据分别送到多个输出端。



# 原理分析



$$D_0 = \bar{A}_1 \bar{A}_0 D$$

$$D_1 = \bar{A}_1 A_0 D$$

$$D_2 = A_1 \bar{A}_0 D$$

$$D_3 = A_1 A_0 D$$

写逻辑表达式

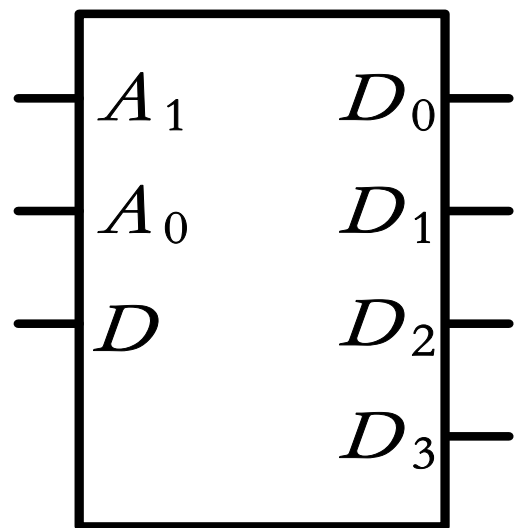
列真值表

$A_1$	$A_0$	输出
0	0	$D_0=D$
0	1	$D_1=D$
1	0	$D_2=D$
1	1	$D_3=D$



## 分析逻辑功能

当 $A_1A_0$ 为不同组合时，输入数据 $D$ 可以有选择地被分配到 $D_0 \sim D_3$ 四路输出中，实现了数据的多路分配





## 4 编码器

---

### 编码的概念

用数码信号表示特定对象的过程称为编码。

例如运动员号码、身份证号码、汉字编码等。

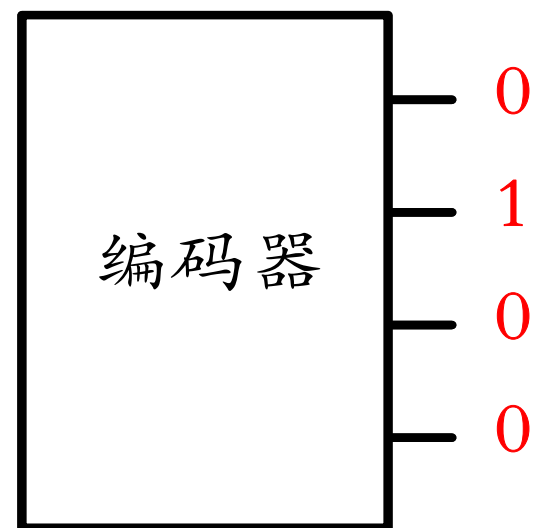
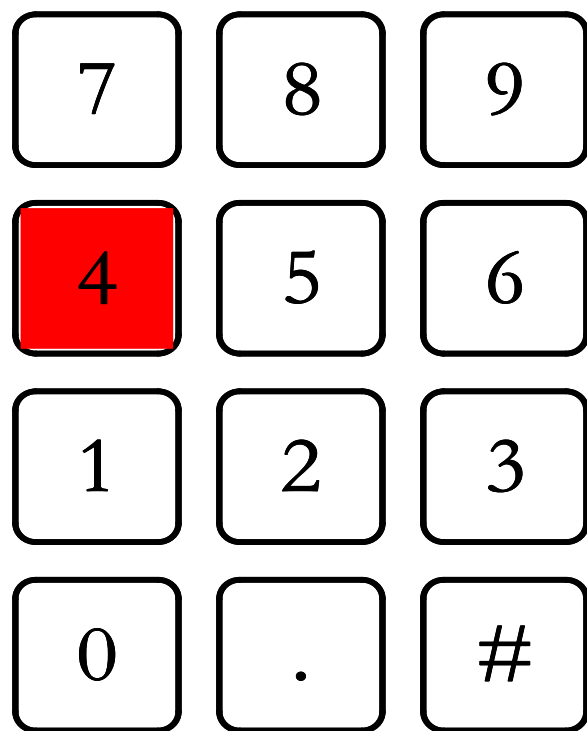


## 二进制编码

用多位二进制数形成一组二进制代码，如果将代码赋予特定的含义，就称为二进制编码。

如计算机、计数器的键盘和按键，可将数字、符号转换为相应的二进制代码，是典型的编码器。





## 8线-3线普通编码器

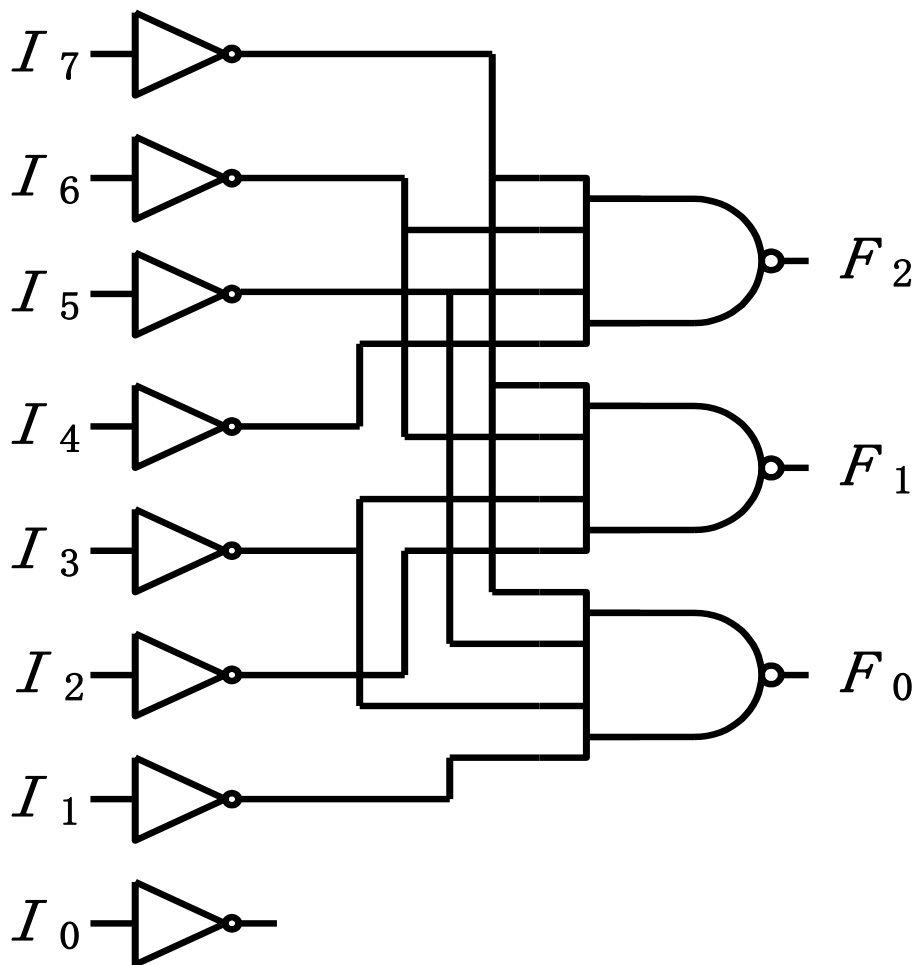
输入8个高、低电平信号，输出3位二进制数。

如5有效（0或1），即对5编码，输出0101（原码）

或1010（反码）



## 写逻辑表达式



$$F_2 = \overline{\overline{I_4} \overline{I_5} \overline{I_6} \overline{I_7}}$$

$$F_1 = \overline{\overline{I_2} \overline{I_3} \overline{I_6} \overline{I_7}}$$

$$F_0 = \overline{\overline{I_1} \overline{I_3} \overline{I_5} \overline{I_7}}$$



## 列真值表

## 分析逻辑功能

$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$	$F_2$	$F_1$	$F_0$
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

输入1有效  
输出对应的  
二进制代码  
实现编码功能

问题

同一时刻只  
允许输入一个1



# 总结

---

## 组合逻辑电路的一般分析方法

- 1、写表达式;
- 2、列真值表;
- 3、说明功能。

## 常用组合逻辑电路

加法器、数据选择器

多路分配器、编码器



# 4.3 组合逻辑电路的设计

---

## 概述

设计组合电路时，由于所设计的电路功能、复杂程度不同，所需的逻辑门电路从几个、几十个到数百个甚至更多。

应该根据实际要求，选择不同规模的集成电路。





实际问题

用小规模集成电路 (SSI) 实现  
SSI — 各种逻辑门

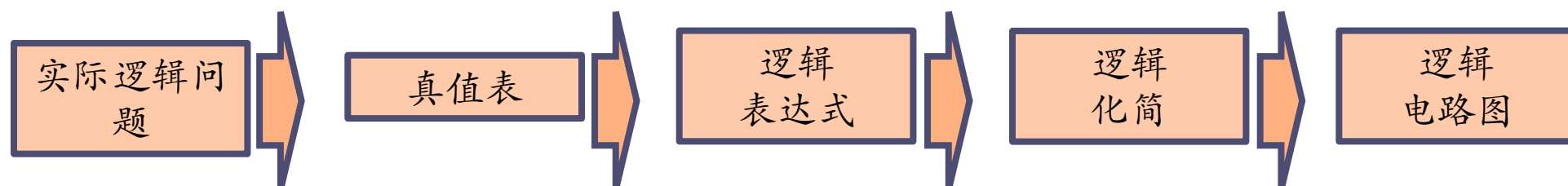
用中规模集成电路 (MSI) 实现  
MSI — 译码器、选择器等

用大规模集成电路 (LSI) 实现  
LSI — 存储器、可编程器件等



# 用小规模集成电路设计组合电路

## 设计的一般步骤



# 例：设计三人表决电路

## 第一步

实际问题逻辑化

输入 $A$ 、 $B$ 、 $C$ 同意为1、  
不同意为0；表决结果 $F$ 通过  
为1、否则为0。

## 第二步

根据要求列真值表

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



## 第三步

根据真值表，列出逻辑表达式

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

如果选用与门和或门

$$F = AB + AC + BC$$

如果选用与非门

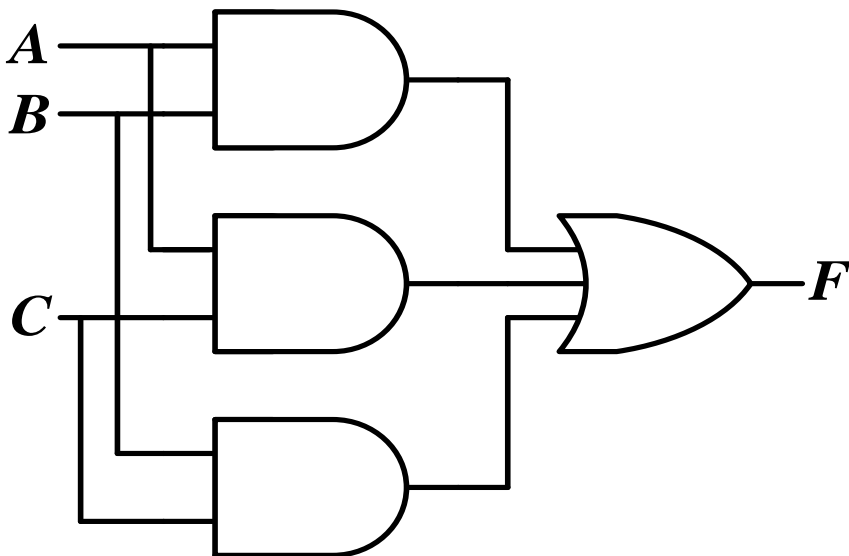
$$F = \overline{\overline{AB} \overline{AC} \overline{BC}}$$



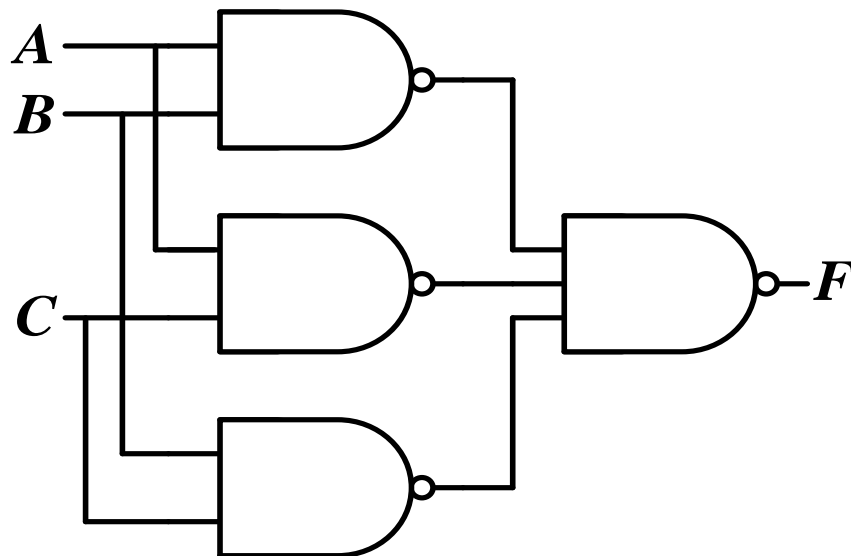
## 第四步

根据逻辑式画逻辑图

$$F = AB + AC + BC$$



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



# 常用组合电路及设计

---

## 1 优先编码器的设计及应用

功能：

允许多个输入同时有效，按规定的优先级别进行编码。



---

例：设计一个10线-4线优先编码器，输入 $I_0 \sim I_9$ （低电平输入有效）、输出反码、优先级自高向低为：

$$I_9 I_8 \cdot \cdot \cdot I_1 I_0$$



# 第一步：按要求写出真值表

输入									输出			
$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$	$I_8$	$I_9$	$F_3$	$F_2$	$F_1$	$F_0$
1	1	1	1	1	1	1	1	1	1	1	1	1
x	x	x	x	x	x	x	x	0	0	1	1	0
x	x	x	x	x	x	x	0	1	0	1	1	1
x	x	x	x	x	x	0	1	1	1	0	0	0
x	x	x	x	x	0	1	1	1	1	0	0	1
x	x	x	x	0	1	1	1	1	1	0	1	0
x	x	x	0	1	1	1	1	1	1	0	1	1
x	x	0	1	1	1	1	1	1	1	1	0	0
x	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0





## 第二步：写出逻辑表达式

$$F_3 = \overline{\overline{I_9} + \overline{I_8}I_9}$$

$$F_2 = \overline{\overline{I_7}I_8I_9 + \overline{I_6}I_7I_8I_9 + \overline{I_5}I_6I_7I_8I_9 + \overline{I_4}I_5I_6I_7I_8I_9}$$

$$F_1 = \overline{\overline{I_7}I_8I_9 + \overline{I_6}I_7I_8I_9 + \overline{I_3}I_4I_5I_6I_7I_8I_9 + \overline{I_2}I_3I_4I_5I_6I_7I_8I_9}$$

$$F_0 = \overline{\overline{I_9} + \overline{I_7}I_8I_9 + \overline{I_5}I_6I_7I_8I_9 + \overline{I_3}I_4I_5I_6I_7I_8I_9 + \overline{I_1}I_2I_3I_4I_5I_6I_7I_8I_9}$$



### 第三步：化简表达式

$$F_3 = \overline{\overline{I_9} + \overline{I_8}}$$

$$F_2 = \overline{\overline{I_7 I_8 I_9} + \overline{I_6 I_8 I_9} + \overline{I_5 I_8 I_9} + \overline{I_4 I_8 I_9}}$$

$$F_1 = \overline{\overline{I_7 I_8 I_9} + \overline{I_6 I_8 I_9} + \overline{I_3 I_4 I_5 I_8 I_9} + \overline{I_2 I_4 I_5 I_8 I_9}}$$

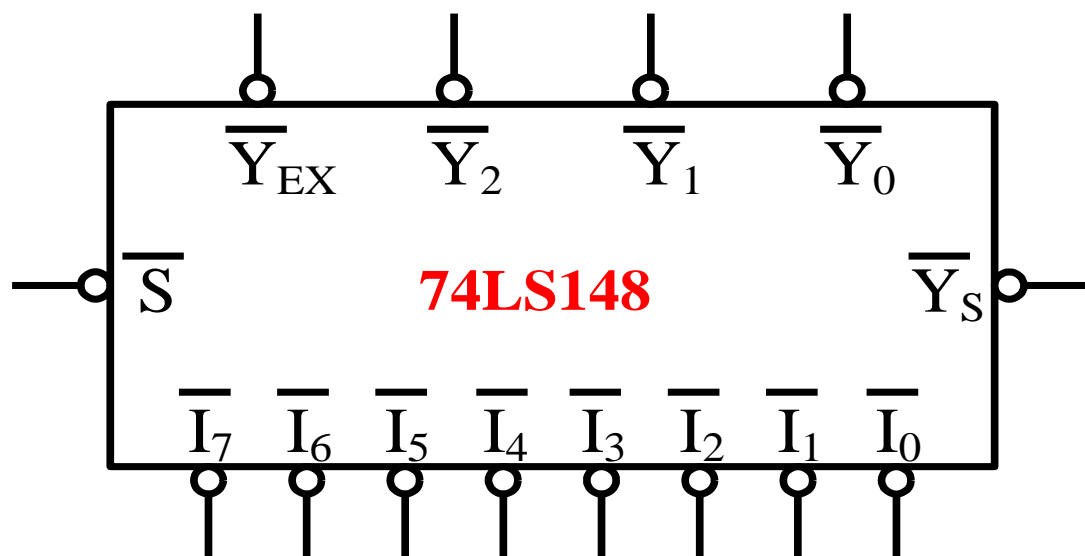
$$F_0 = \overline{\overline{I_9} + \overline{I_7 I_8} + \overline{I_5 I_6 I_8} + \overline{I_3 I_4 I_6 I_8} + \overline{I_1 I_2 I_4 I_6 I_8}}$$

### 第四步：画出逻辑图



## 例：典型优先编码器（148）及扩展应用

74LS148为8线-3线优先编码器，输入低电平有效、输出反码。

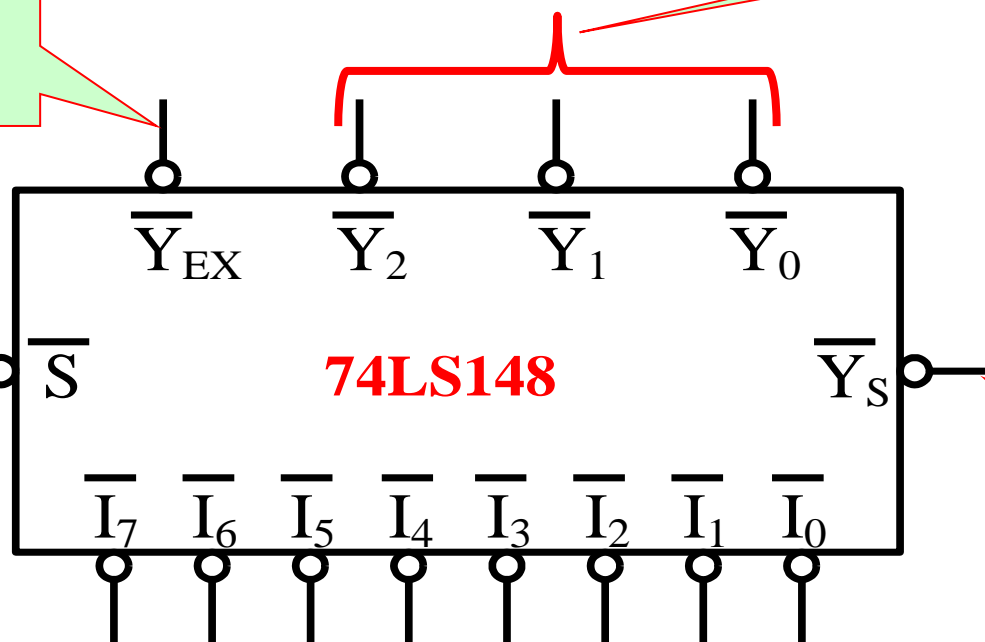


输 入									输 出				
$\overline{S}$	$\overline{I_7}$	$\overline{I_6}$	$\overline{I_5}$	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	$\overline{I_0}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$	$\overline{Y_S}$	$\overline{Y_{EX}}$
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	x	x	x	x	x	x	x	0	0	0	1	0
0	1	0	x	x	x	x	x	x	0	0	1	1	0
0	1	1	0	x	x	x	x	x	0	1	0	1	0
0	1	1	1	0	x	x	x	x	0	1	1	1	0
0	1	1	1	1	0	x	x	x	1	0	0	1	0
0	1	1	1	1	1	0	x	x	1	0	1	1	0
0	1	1	1	1	1	1	0	x	1	1	0	1	0
0	1	1	1	1	1	1	1	0	1	1	1	1	0



输出反码

有编码输入时，为  
0



选通端  
0有效

无编码输入时，为  
0

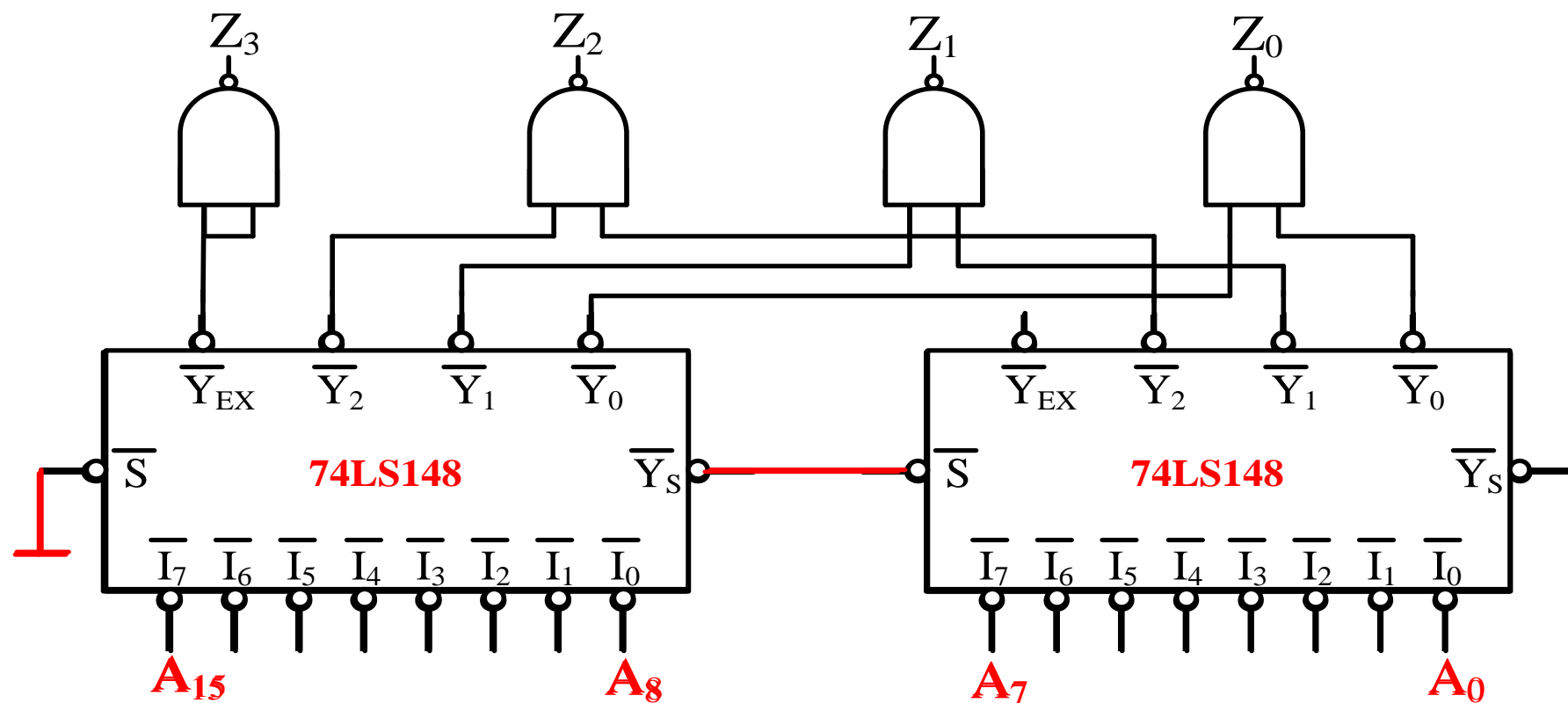
优先级



# 优先编码器的扩展

用两片148组成为16线-4线优先编码器

输出为原码



## 2 译码器的设计及应用

---

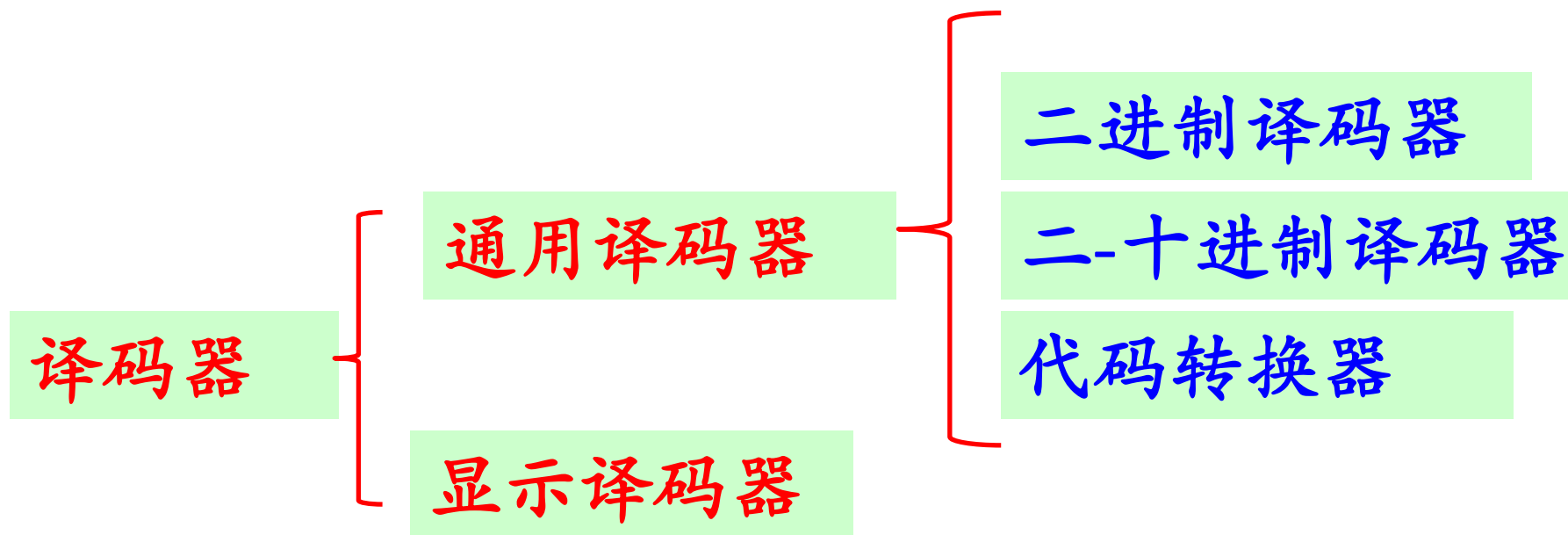
译码是编码的逆过程，即将代码“翻译”为特定的对象，将一组二进制代码“翻译”为一组高低电平信号。

能实现译码功能的电路称为译码器。

译码器也是一种多输入多输出的组合逻辑电路。



# 译码器的种类





## 二进制译码器

---

将 $n$ 位二进制代码，译为特定含义的 $2^n$ 个输出信号，称为二进制译码器。

常用的有2线-4线译码器、3线-8线译码器和4线-16线译码器等。



**例：** 设计3线-8线译码器，输入原码、输出高电平有效。

---

## 分析

该电路为3输入、8输出的组合电路。

当输入为000~111时，8个输出依次为高电平。

例如若 $ABC$ 为110，则 $F_7 \sim F_0$ 为01000000



## 第一步：按照要求写真值表

<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i> <sub>7</sub>	<i>F</i> <sub>6</sub>	<i>F</i> <sub>5</sub>	<i>F</i> <sub>4</sub>	<i>F</i> <sub>3</sub>	<i>F</i> <sub>2</sub>	<i>F</i> <sub>1</sub>	<i>F</i> <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0



## 第二步：写逻辑式

$$F_0 = \overline{A}\overline{B}\overline{C} = m_0$$

$$F_1 = \overline{A}\overline{B}C = m_1$$

$$F_2 = \overline{A}B\overline{C} = m_2$$

$$F_3 = \overline{A}BC = m_3$$

$$F_4 = A\overline{B}\overline{C} = m_4$$

$$F_5 = A\overline{B}C = m_5$$

$$F_6 = AB\overline{C} = m_6$$

$$F_7 = ABC = m_7$$

<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i> <sub>7</sub>	<i>F</i> <sub>6</sub>	<i>F</i> <sub>5</sub>	<i>F</i> <sub>4</sub>	<i>F</i> <sub>3</sub>	<i>F</i> <sub>2</sub>	<i>F</i> <sub>1</sub>	<i>F</i> <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0



### 第三步：画逻辑图

$$F_0 = \overline{\overline{A}}\overline{\overline{B}}\overline{\overline{C}} = m_0$$

$$F_1 = \overline{\overline{A}}\overline{\overline{B}}C = m_1$$

$$F_2 = \overline{\overline{A}}B\overline{\overline{C}} = m_2$$

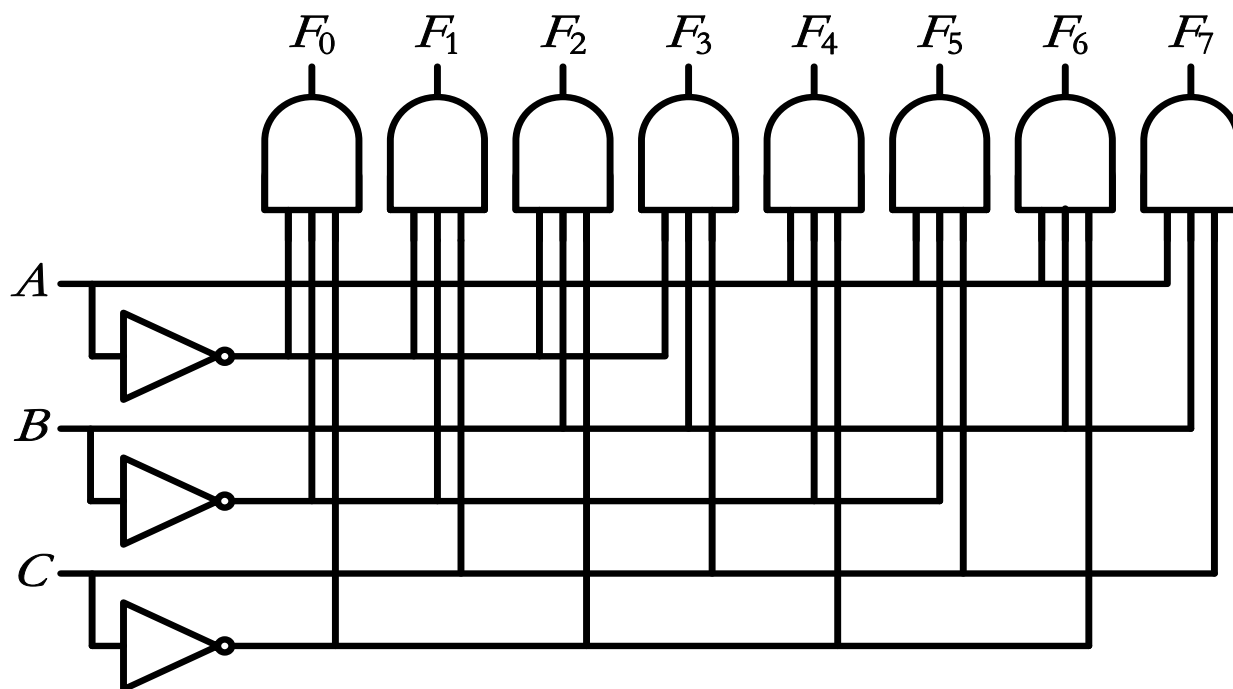
$$F_3 = \overline{\overline{A}}BC = m_3$$

$$F_4 = A\overline{\overline{B}}\overline{\overline{C}} = m_4$$

$$F_5 = A\overline{\overline{B}}C = m_5$$

$$F_6 = AB\overline{\overline{C}} = m_6$$

$$F_7 = ABC = m_7$$

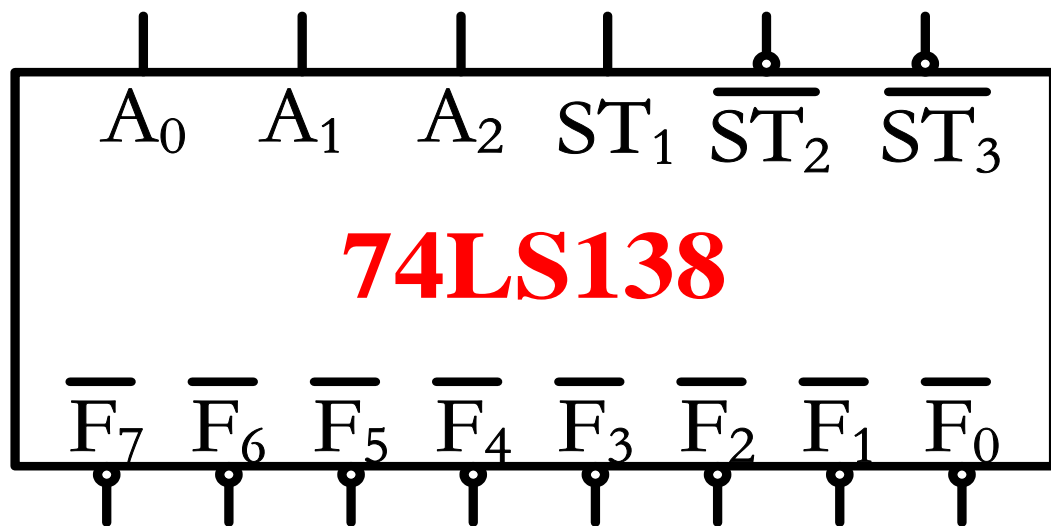


每个输出都是输入变量的最小项，因此又称为最小项译码器。



## 二进制译码器的典型产品 — 74LS138

74LS138为3线-8线译码器，输入原码、输出低电平有效。



选通端

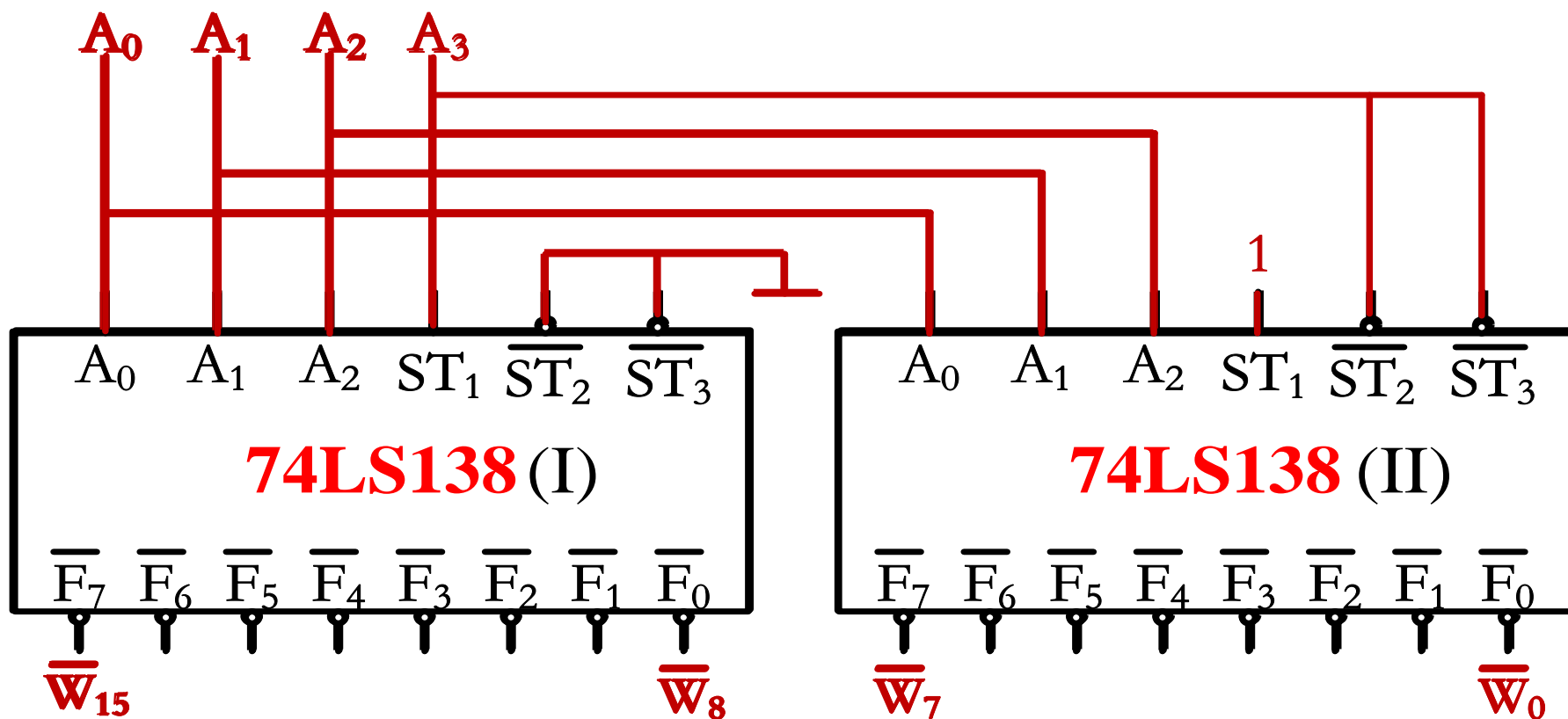
输入端

输出端

$ST_1$	$\overline{ST_2}$	$\overline{ST_3}$	$A_2$	$A_1$	$A_0$	$\overline{F_0}$	$\overline{F_1}$	$\overline{F_2}$	$\overline{F_3}$	$\overline{F_4}$	$\overline{F_5}$	$\overline{F_6}$	$\overline{F_7}$
1	0	0	0	0	0	0	1	1	1	1	1	1	1
			0	0	1	1	0	1	1	1	1	1	1
			0	1	0	1	1	0	1	1	1	1	1
			0	1	1	1	1	1	0	1	1	1	1
			1	0	0	1	1	1	1	0	1	1	1
			1	0	1	1	1	1	1	1	0	1	1
			1	1	0	1	1	1	1	1	1	0	1
			1	1	1	1	1	1	1	1	1	1	0
其他			x	x	x	1	1	1	1	1	1	1	



## 74LS138的扩展——组成4线-16线译码器





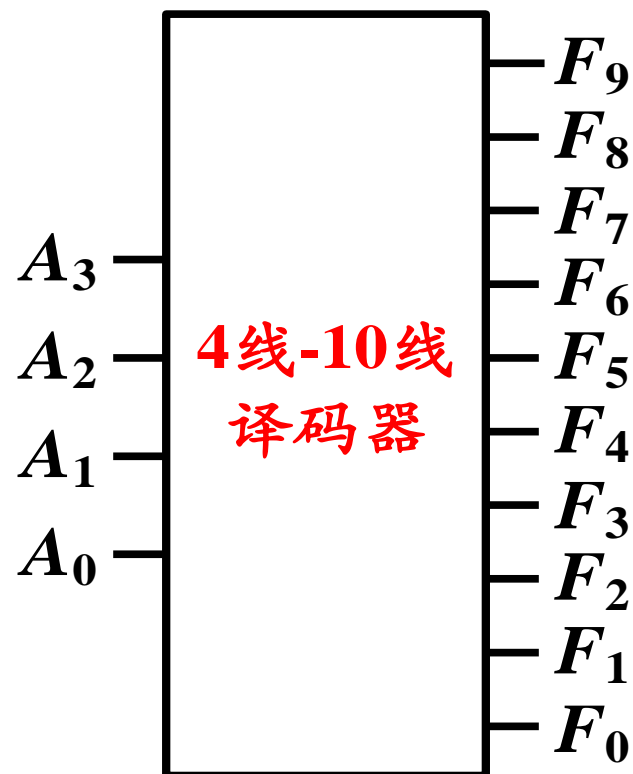
## 二 - 十进制译码器

功能：

输入4位二进制代码，输出10路高低电平信号。

74LS42

输入（0000~1001）为原码、输出（ $F_9 \sim F_0$ ）为高电平有效。



# 显示译码器

---

功能:

将 4 位二进制代码，译为数码显示器所需的信号。

如七段数码显示器，则译为 7 个显示信号，通过数码管显示相应的数字。



二进制代码

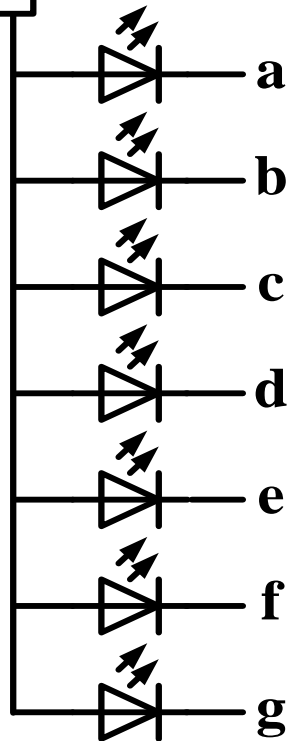
显示  
译码器

显示代码

Vcc

R

共阳极



f

e

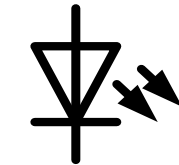
g

d

b

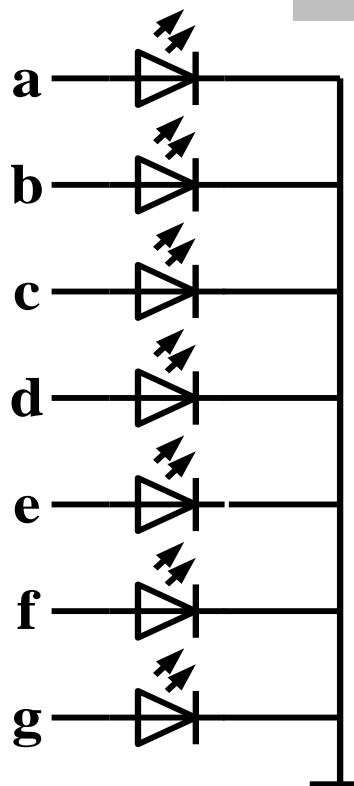
c

阳极



阴极

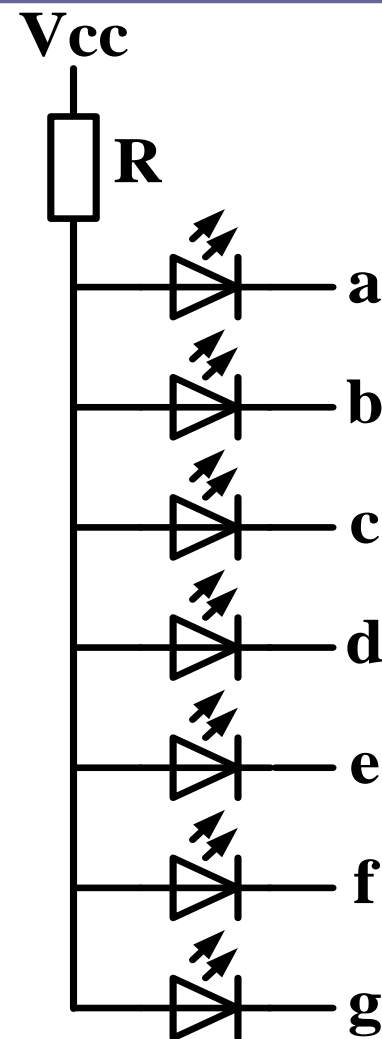
共阴极



# 共阳极数码管

如果七个发光二极管的阳极接在一起并接电源，称为共阳极接法。

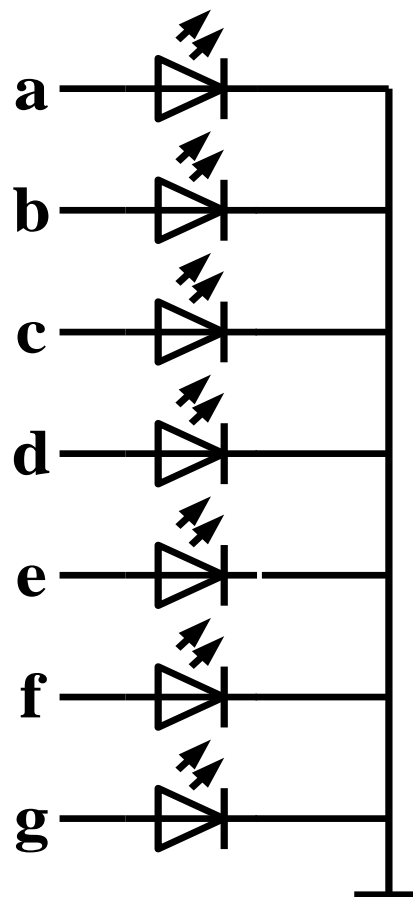
显示代码 ( $a \sim g$ ) 为低电平时，相应的发光二极管导通并发光。



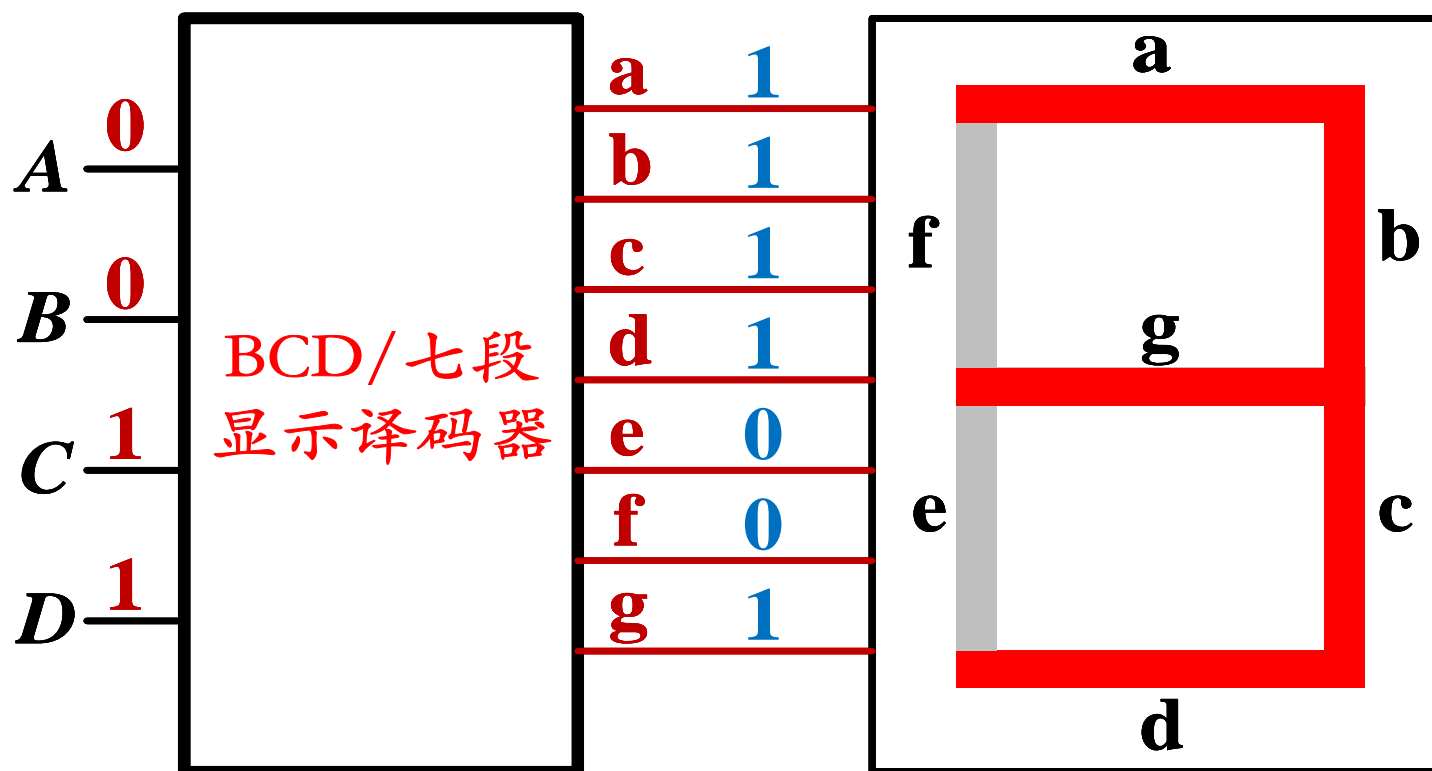
# 共阴极数码管

如果七个发光二极管的阴极接在一起并接地，称为共阴极接法。

显示代码 ( $a \sim g$ ) 为高电平时，相应的发光二极管导通并发光。



## 以共阴极数码管为例



# 七段显示译码器的设计

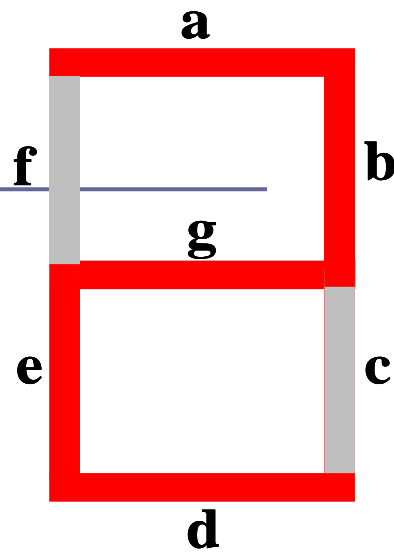
按照显示的要求（数字、字母等）及数码管的结构（共阳极或共阴极），根据组合电路的设计方法进行设计。

例：设计一个七段显示译码器，将 0000、0001、0010 · · · · · 1001（8421BCD码），用共阴极接法的半导体七段显示器依次显示为 0、1 · · · · · 9



# 第一步 列真值表

输入				输出							显示数字
A	B	C	D	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	1	0	1	9
其他				无关项							





## 第二步 写出逻辑表达式

(利用无关项化简)

$$a = A + C + BD + \overline{\overline{B}}\overline{\overline{D}} = \overline{\overline{\overline{A}}\overline{\overline{C}}\overline{\overline{B}}\overline{\overline{D}}\overline{\overline{B}}\overline{\overline{D}}}$$

$$b = \overline{B} + \overline{C}\overline{D} + CD = \overline{\overline{\overline{B}}\overline{\overline{C}}\overline{\overline{D}}\overline{\overline{C}}\overline{\overline{D}}}$$

$$c = \overline{C} + B + D = \overline{\overline{C}}\overline{\overline{B}}\overline{\overline{D}}$$

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

$$e = C\overline{D} + \overline{B}\overline{D} = \overline{\overline{\overline{C}}\overline{\overline{D}}\overline{\overline{B}}\overline{\overline{D}}}$$

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

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

## 第三步 画出逻辑电路图



## 典型的BCD-七段显示译码器（4线-7线译码器）

---

74LS47:

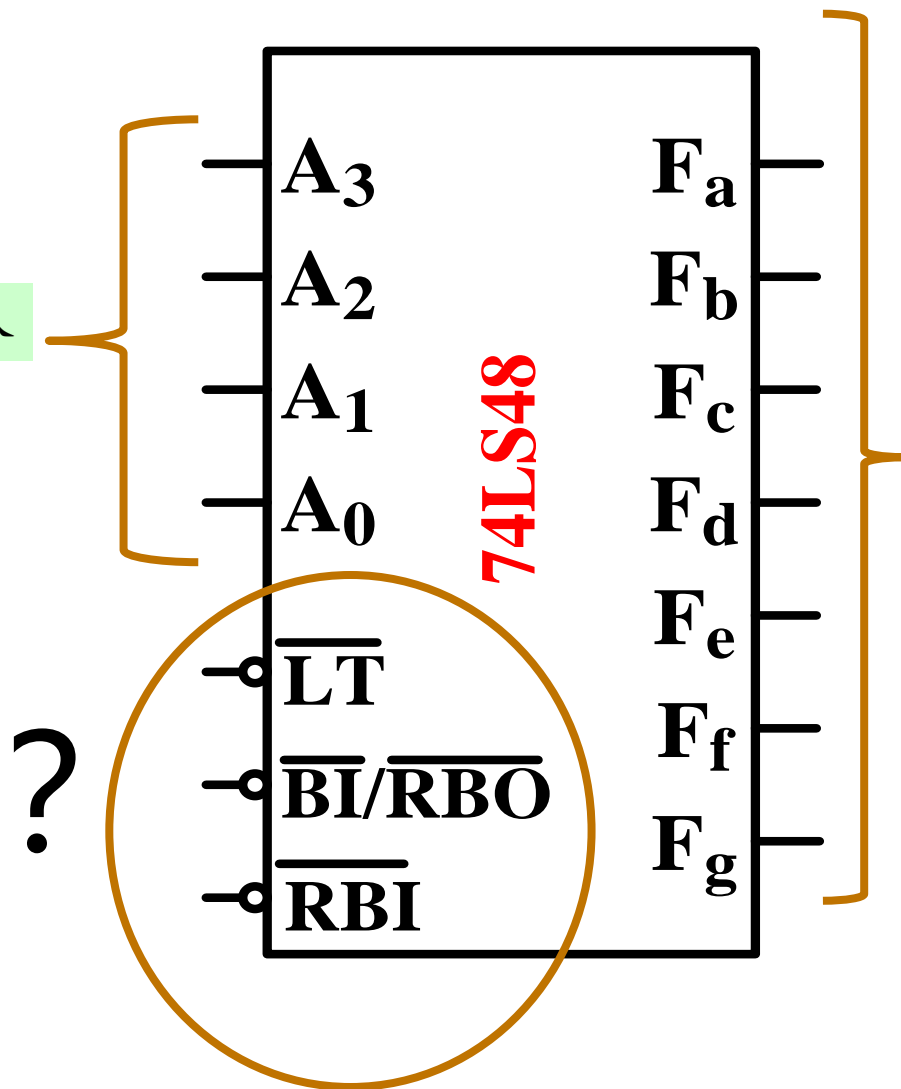
输出低电平有效，用于共阳极数码管

74LS48:

输出高电平有效，用于共阴极数码管



BCD码输入



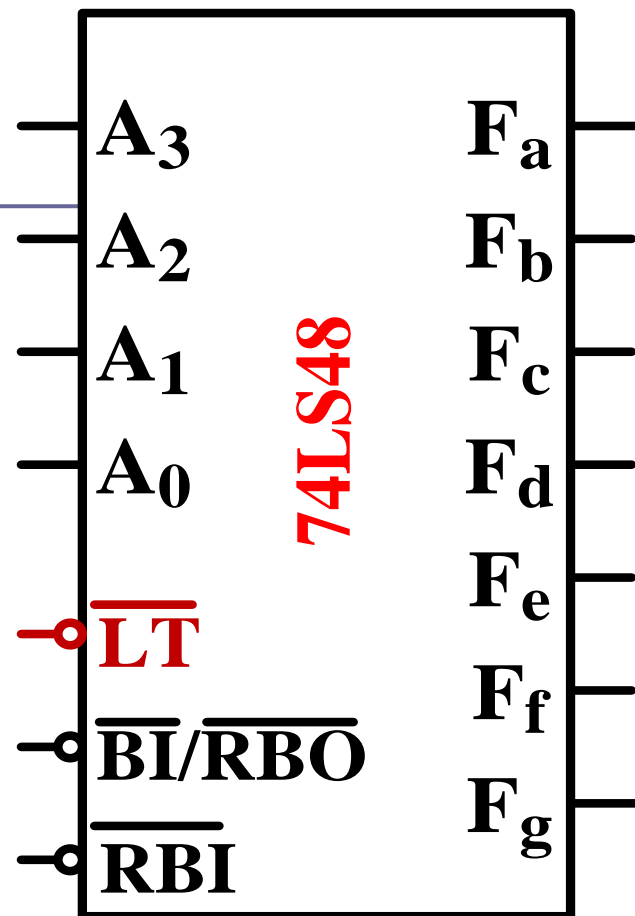
显示信号输出



# 灯测试输入

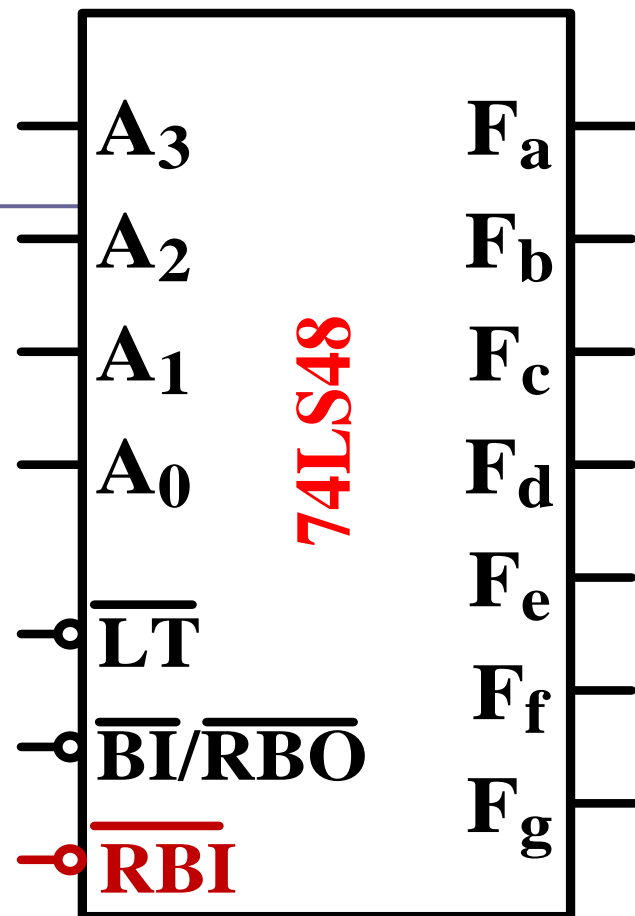
为低电平时，数码管应显示  
“8”。

正常使用，应接高电平。



# 灭零输入

为低电平且 $A_3A_2A_1A_0=0$ 时，  
数码管不显示（灭）。

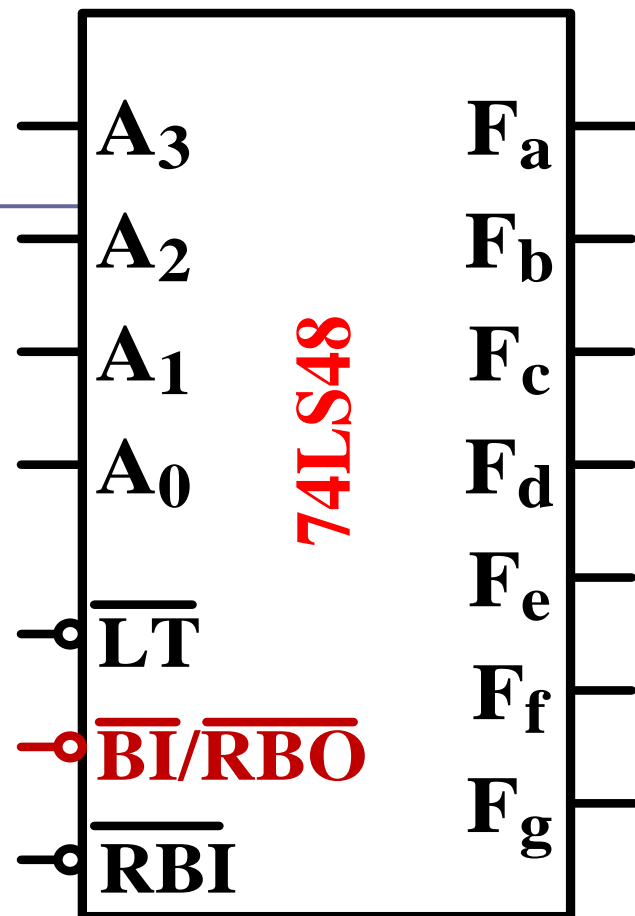


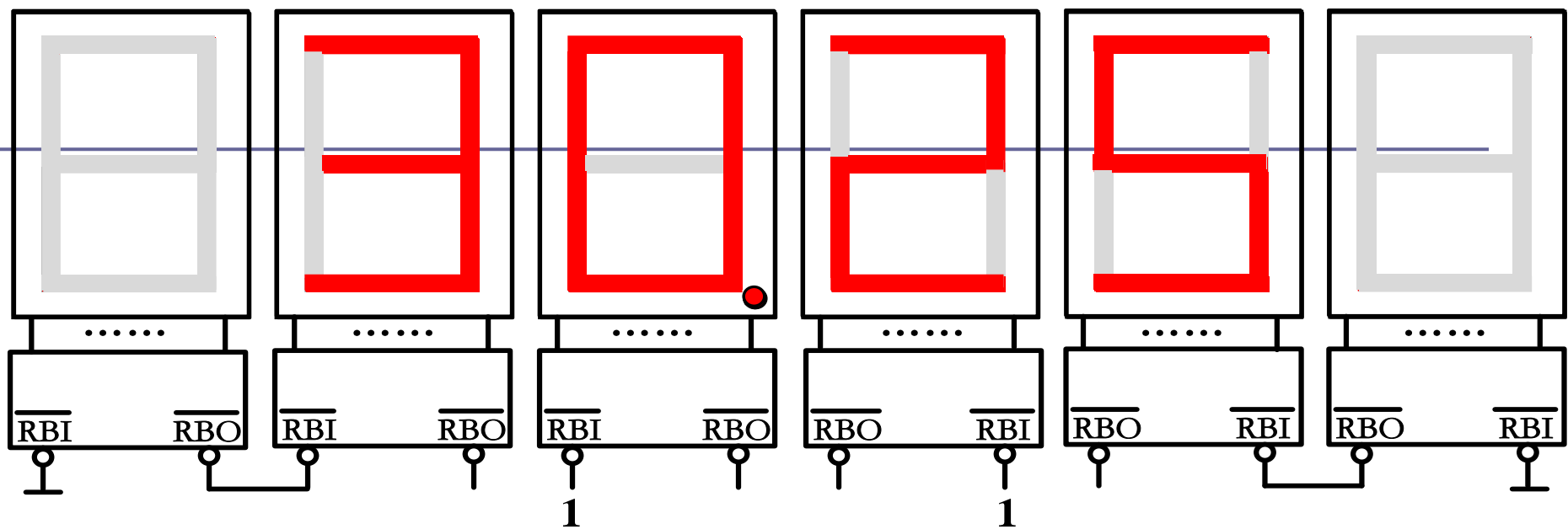
# 熄灭输入/灭零输出

双重功能的输入/输出端。

输入：外加低电平时所有灯  
熄灭。

输出：当  $A_3A_2A_1A_0=0$  且  $\overline{RBI}$   
= 0 时输出为 0。





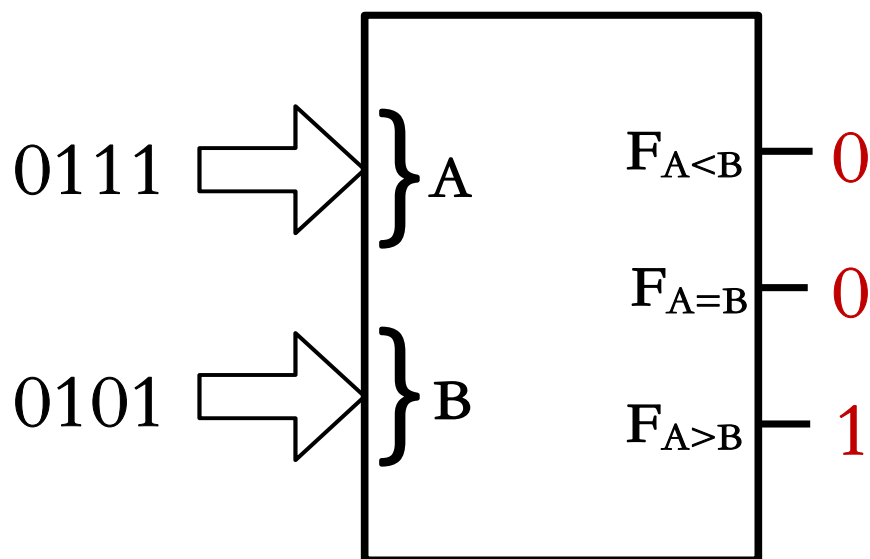
这是一个多位显示电路，整数部分的最高位和小数部分的最低位不显示0，如这两位为0则熄灭，同时整数部分的次高位和小数部分的次低位也不能显示0。但小数点前后两位应能显示0。



### 3 数值比较器的设计及应用

## 功能

比较两个相同位数的二进制数的大小，由 $F_{A=B}$ 、 $F_{A<B}$ 、 $F_{A>B}$ 三个输出表示比较的结果。





# 一位比较器的设计

$A$ 、 $B$  均为一位二进制数，输出为  $F_{A>B}$ 、 $F_{A=B}$ 、

$F_{A<B}$ 。

## 第一步 列真值表

$A$	$B$	$F_{A>B}$	$F_{A=B}$	$F_{A<B}$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0



## 第二步 写逻辑表达式

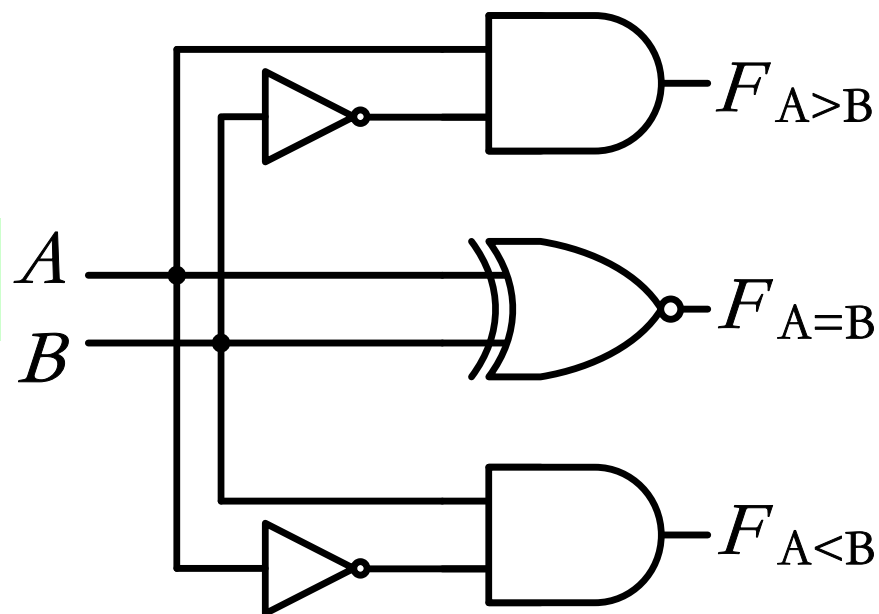
$$F_{A>B} = A\bar{B}$$

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

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

A	B	$F_{A>B}$	$F_{A=B}$	$F_{A<B}$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

## 第三步 画逻辑电路图



# 多位比较器的设计

$A$ 、 $B$ 是两个多位二进制数，应从最高位开始比较，依次比较，只有各位数相比全部相等时，两数才相等。

例

设计4位比较器， $A$ 、 $B$ 均为4位二进制数。



# 第一步 列真值表

输入				输出		
$A_3B_3$	$A_2B_2$	$A_1B_1$	$A_0B_0$	$F_{A>B}$	$F_{A=B}$	$F_{A<B}$
$A_3>B_3$	x	x	x	1	0	0
$A_3<B_3$	x	x	x	0	0	1
$A_3=B_3$	$A_2>B_2$	x	x	1	0	0
$A_3=B_3$	$A_2<B_2$	x	x	0	0	1
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	x	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	x	0	0	1
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	0	0	1
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	1	0



## 第二步 列逻辑表达式

$$F_{A<B} = \overline{A_3}B_3 + \overline{A_3 \oplus B_3} \overline{A_2}B_2 + \overline{A_3 \oplus B_3} \overline{A_2 \oplus B_2} \overline{A_1}B_1 \\ + \overline{A_3 \oplus B_3} \overline{A_2 \oplus B_2} \overline{A_1 \oplus B_1} \overline{A_0}B_0$$

$$F_{A=B} = \overline{A_3 \oplus B_3} \overline{A_2 \oplus B_2} \overline{A_1 \oplus B_1} \overline{A_0 \oplus B_0}$$

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

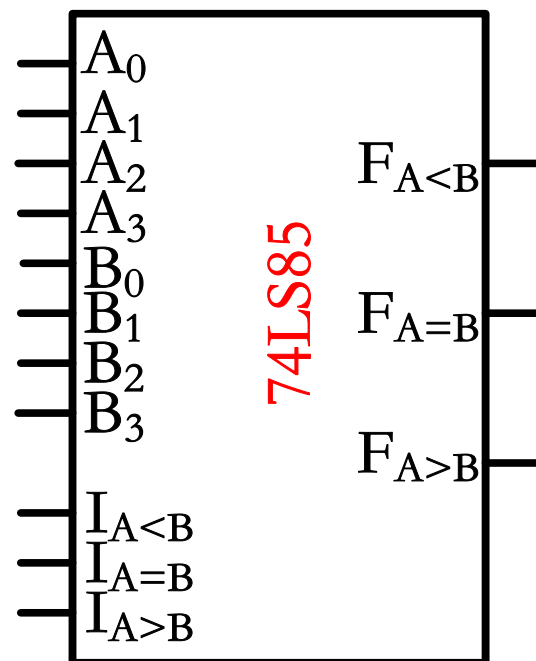
第三步 画逻辑电路图 (略)



# 集成化4位比较器（74LS85）及级联

特点：

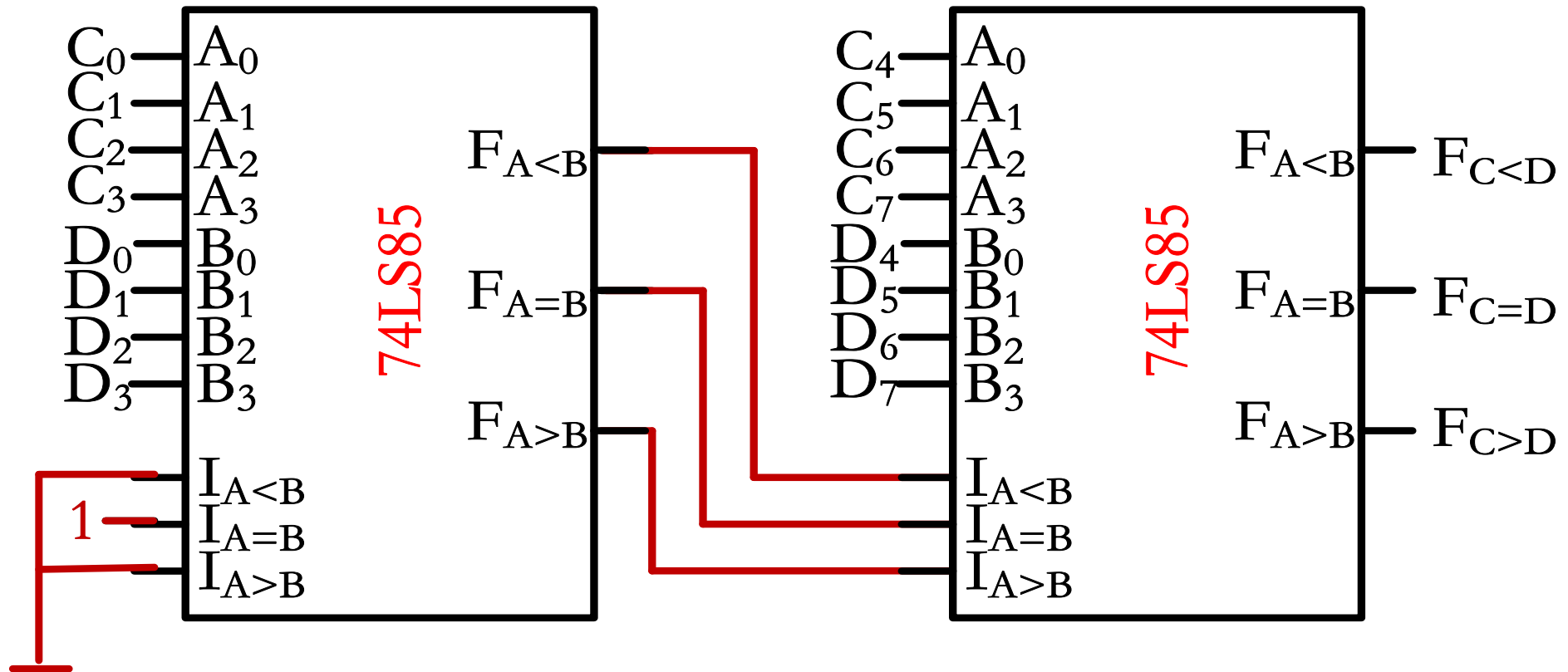
能进行两个4位二进制数比较，为扩展使用，增加级联输入端。



输入							输出		
$A_3B_3$	$A_2B_2$	$A_1B_1$	$A_0B_0$	$I_{A>B}$	$I_{A=B}$	$I_{A<B}$	$F_{A>B}$	$F_{A=B}$	$F_{A<B}$
$A_3>B_3$	x	x	x	x	x	x	1	0	0
$A_3<B_3$	x	x	x	x	x	x	0	0	1
$A_3=B_3$	$A_2>B_2$	x	x	x	x	x	1	0	0
$A_3=B_3$	$A_2<B_2$	x	x	x	x	x	0	0	1
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	x	x	x	x	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	x	x	x	x	0	0	1
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	x	x	x	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	x	x	x	0	0	1
				0	0	1	0	0	1
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	1	0	0	1	0
				1	0	0	1	0	0



## 用74LS85实现8位二进制数的比较（级联）





# 用中规模集成电路设计组合电路

---

采用中规模器件（一般指译码器、数据选择器和全加器等）设计组合电路，应对逻辑函数进行变换，得到与指定器件相一致的表达式。



# 用数据选择器、译码器设计的思路

实际问题

与或形式的表达式

数据选择器  
(4选1)

$$\begin{aligned} F &= \bar{A}_1 \bar{A}_0 D_0 + \bar{A}_1 A_0 D_1 + A_1 \bar{A}_0 D_2 + A_1 A_0 D_3 \\ &= m_0 D_0 + m_1 D_1 + m_2 D_2 + m_3 D_3 \end{aligned}$$

实现

译码器

$$\begin{aligned} F_0 &= \bar{A}\bar{B} = m_0 \\ F_1 &= \bar{A}B = m_1 \\ F_2 &= A\bar{B} = m_2 \\ F_3 &= AB = m_3 \end{aligned}$$



# 用数据选择器设计组合电路

---

## 用数据选择器实现逻辑函数

例：设计三人表决电路

解：

通过真值表，得到逻辑表达式：

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



## 方法一：选用8选1数据选择器

用3个控制端的选择器实现3变量的组合电路。

写出8选1数据选择器的逻辑式：

$$F = \bar{A}_2 \bar{A}_1 \bar{A}_0 D_0 + \bar{A}_2 \bar{A}_1 A_0 D_1 + \bar{A}_2 A_1 \bar{A}_0 D_2 + \bar{A}_2 A_1 A_0 D_3 \\ + A_2 \bar{A}_1 \bar{A}_0 D_4 + A_2 \bar{A}_1 A_0 D_5 + A_2 A_1 \bar{A}_0 D_6 + A_2 A_1 A_0 D_7$$



$$F = \bar{A}_2 \bar{A}_1 \bar{A}_0 D_0 + \bar{A}_2 \bar{A}_1 A_0 D_1 + \bar{A}_2 A_1 \bar{A}_0 D_2 + \bar{A}_2 A_1 A_0 D_3 + A_2 \bar{A}_1 \bar{A}_0 D_4 + A_2 \bar{A}_1 A_0 D_5 + A_2 A_1 \bar{A}_0 D_6 + A_2 A_1 A_0 D_7$$

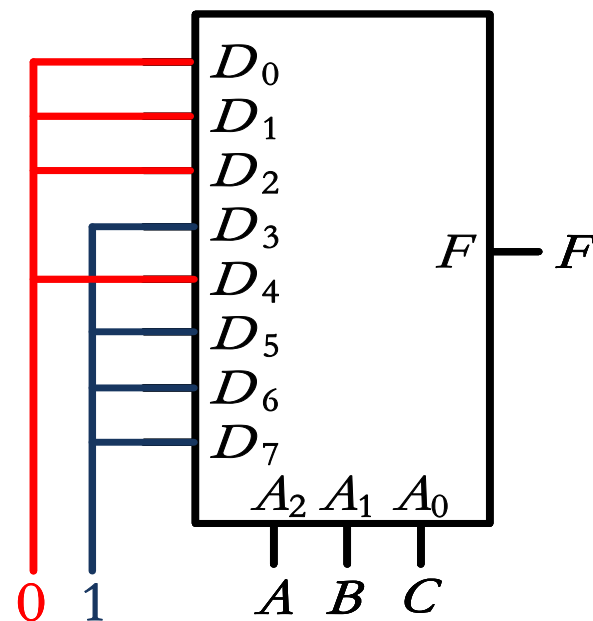
和要实现的逻辑表达式对比

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

令：  $A_2=A$ 、  $A_1=B$ 、  $A_0=C$

则：  $D_0=D_1=D_2=D_4=0$

$D_3=D_5=D_6=D_7=1$



即可实现所要求的逻辑功能。



## 方法二：选用4选1数据选择器

用2个控制端的选择器实现3变量的组合电路，需分离出多余的变量。

写出4选1数据选择器的逻辑式：

$$F = \overline{A_1} \overline{A_0} D_0 + \overline{A_1} A_0 D_1 + A_1 \overline{A_0} D_2 + A_1 A_0 D_3$$



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

$$= \bar{A}\bar{B}(0) + \bar{A}B(C) + A\bar{B}(C) + AB(1)$$

将要实现的逻辑表达式变换

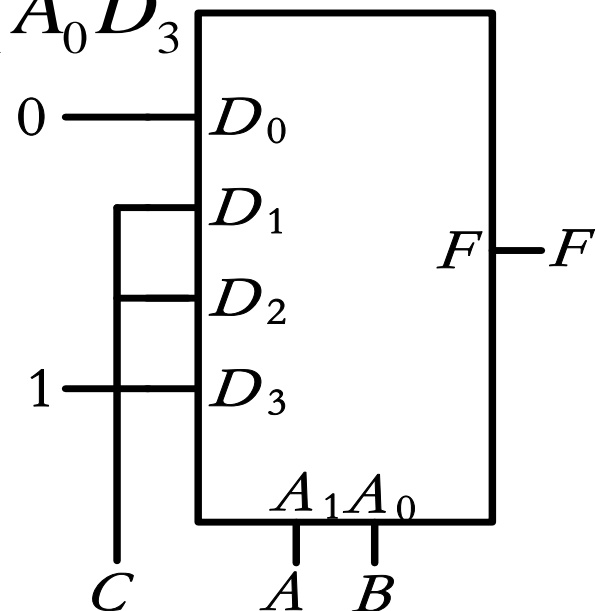
$$F = \bar{A}_1\bar{A}_0D_0 + \bar{A}_1A_0D_1 + A_1\bar{A}_0D_2 + A_1A_0D_3$$

令：  $A_1=A$ 、  $A_0=B$

$$D_0=0$$

$$D_1=D_2=C$$

$$D_3=1$$



即可实现所要求的逻辑功能。



# 用数据选择器设计逻辑函数小结

---

如果逻辑函数输入变量数与数据选择器控制端数量相同（如用8选1实现3变量函数），则输入变量与控制变量一一相接，数据输入端接高、低电平。





---

如果逻辑函数输入变量数多于数据选择器控制端数  
(如用4选1实现3变量函数)，则需分离多余的变量。

未被分离的输入变量与控制变量相接，被分离变量  
则与数据输入端相接。



---

一般情况下，一个 $n$ 变量的逻辑函数可用  $(2^n)$  选1 或  $(2^{n-1})$  选1数据选择器实现。

如果部分变量出现的频率更低的话，则通过一些门电路可实现更多变量的逻辑函数。



例：用8选1数据选择器实现5变量逻辑函数

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

用3个控制端的选择器实现5个变量的逻辑函数，需分离出两个变量。

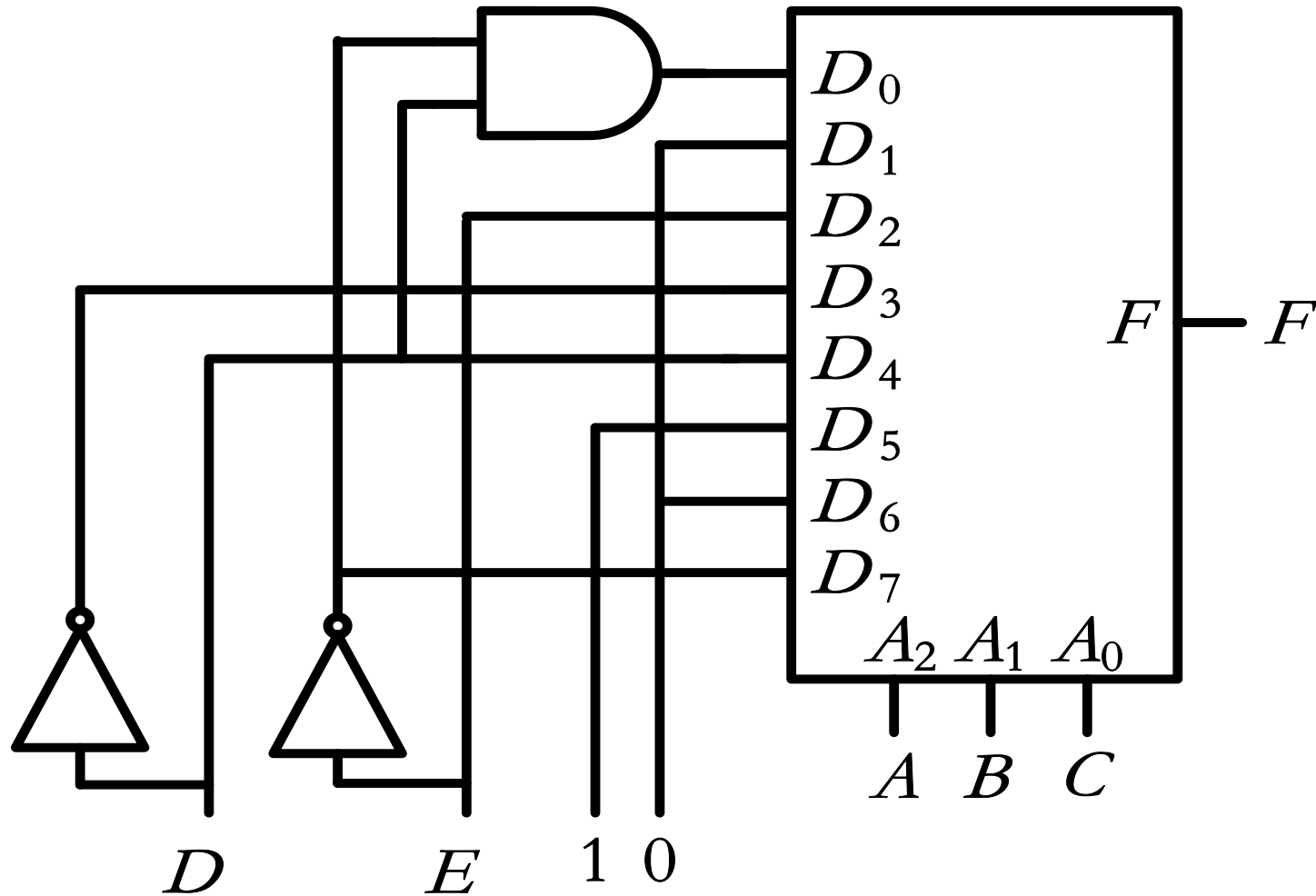
D、E出现较少，可将其分离，并经过附加的电路送到输入端。

$$F = \overline{A}\overline{B}\overline{C}(D\overline{E}) + \overline{A}\overline{B}\overline{C}(E) + \overline{A}\overline{B}C(\overline{D}) + A\overline{B}\overline{C}(D) + A\overline{B}C(1) + ABC(\overline{E})$$



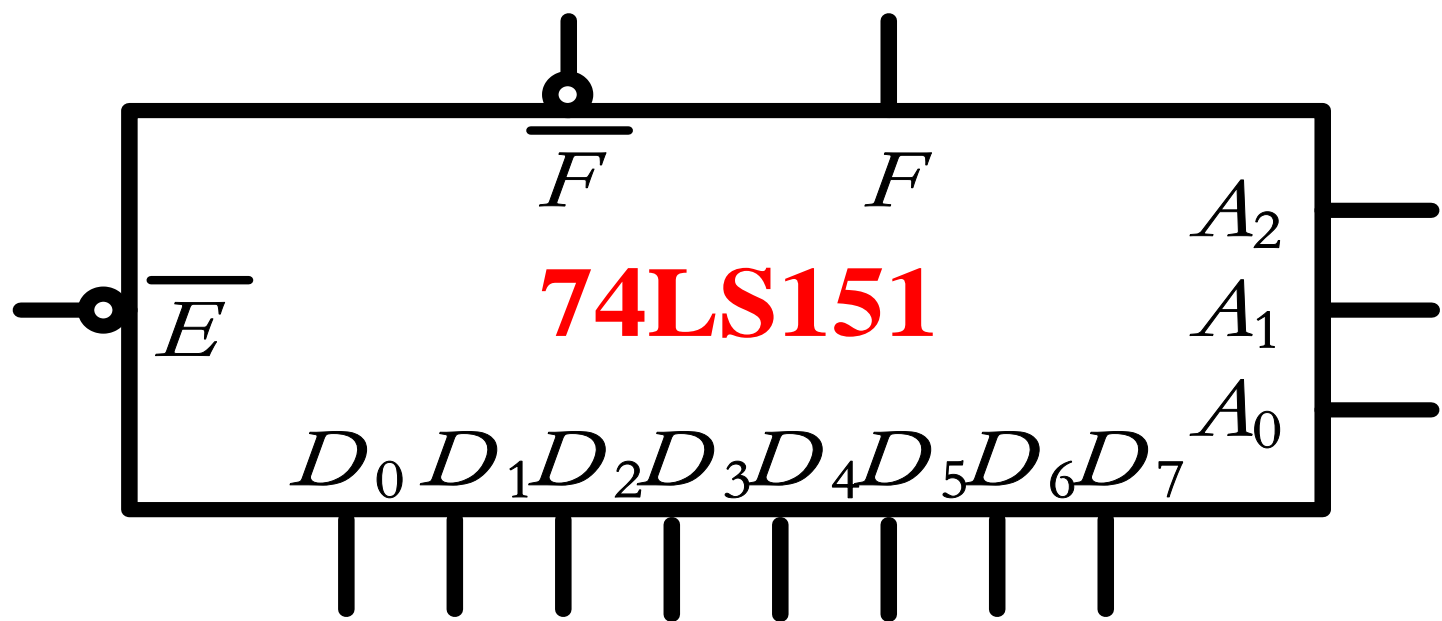
$$F = \overline{A}\overline{B}\overline{C}(D\overline{E}) + \overline{A}\overline{B}\overline{C}(E) + \overline{A}BC(\overline{D}) + A\overline{B}\overline{C}(D) + A\overline{B}C(1) + ABC(\overline{E})$$


---



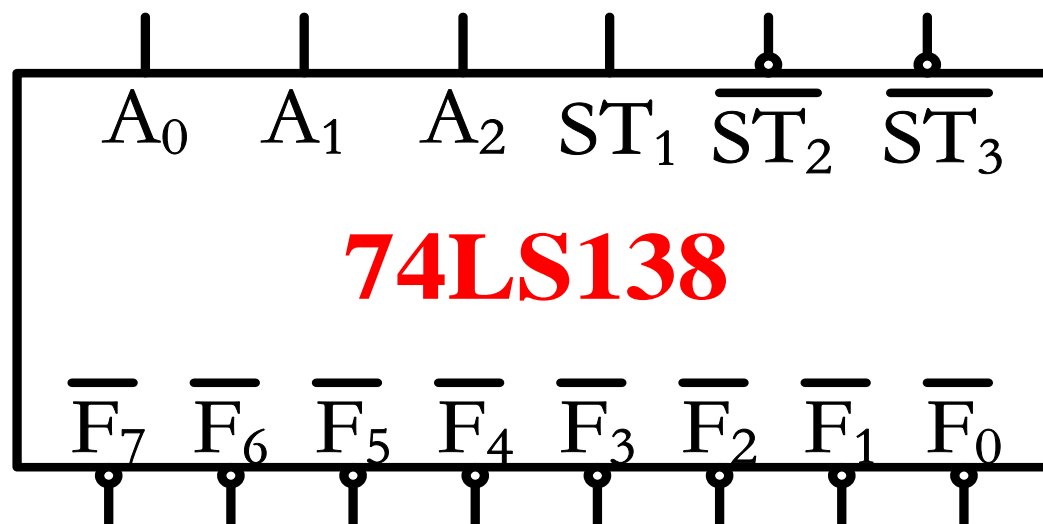
# 用数据选择器构成等值数码比较器

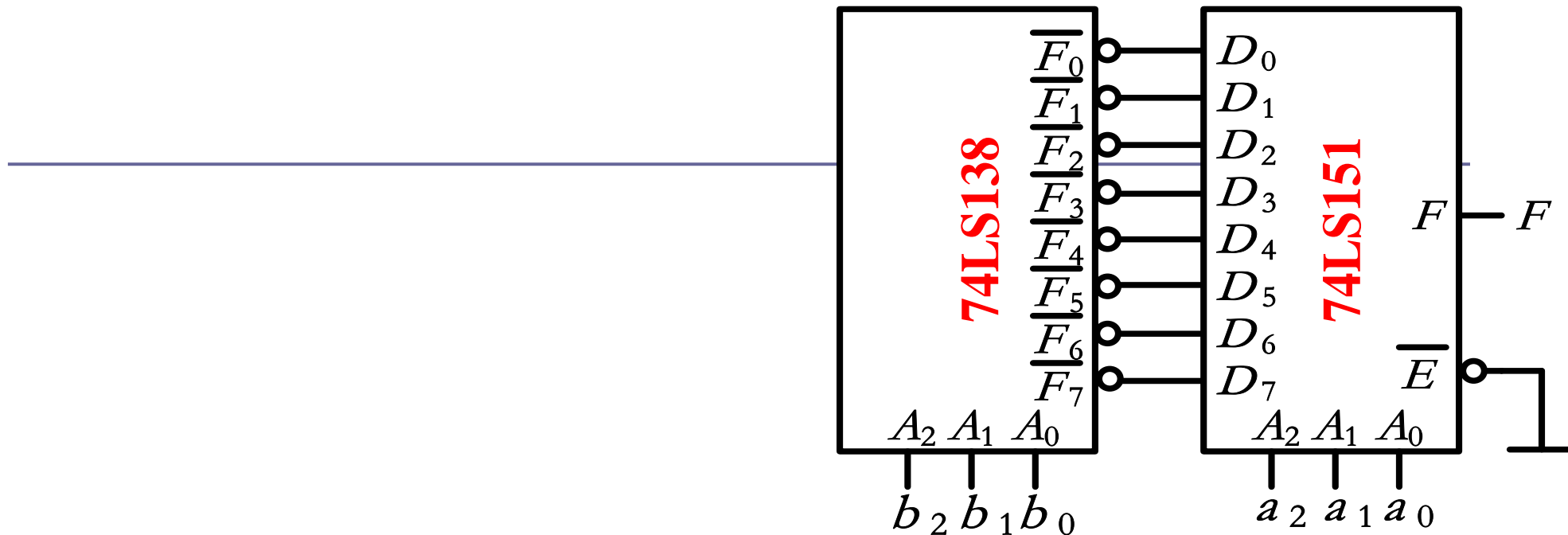
74LS151, 八选一数据选择器



# 复习— 74LS138

74LS138为3线-8线译码器，输入原码、输出低电平有效。





两输入  $A = a_2 a_1 a_0$ ,  $B = b_2 b_1 b_0$

当  $A$  与  $B$  相等时, 输出  $F=0$

当  $A$  与  $B$  不等时, 输出为  $F=1$ 。



# 用译码器设计组合电路

## 用译码器设计组合电路

对于最小项译码器来说，其输出是输入变量的所有最小项。

由于所有逻辑函数都可转化成其最小项的和的形式，因此任何逻辑函数都可采用译码器实现。

首先需要将表达式转换成最小项的和的形式。





---

3线-8线译码器可实现任何3变量的逻辑函数。

4线-16线译码器可实现任何4变量的逻辑函数



## 例：用译码器实现逻辑函数

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

$F$ 为3变量逻辑函数，因此选用 74LS138（3线-8线译码器）。

输入为原码、输出低电平有效。每个输出对应一个以输入为变量的最小项。其逻辑式为：

$$\overline{F}_0 = \overline{\overline{A}\overline{B}\overline{C}} = \overline{m_0}$$

$$\overline{F}_1 = \overline{\overline{A}\overline{B}C} = \overline{m_1}$$

$$\overline{F}_2 = \overline{\overline{A}B\overline{C}} = \overline{m_2}$$

$$\overline{F}_3 = \overline{\overline{A}BC} = \overline{m_3}$$

$$\overline{F}_4 = \overline{A\overline{B}\overline{C}} = \overline{m_4}$$

$$\overline{F}_5 = \overline{A\overline{B}C} = \overline{m_5}$$

$$\overline{F}_6 = \overline{AB\overline{C}} = \overline{m_6}$$

$$\overline{F}_7 = \overline{ABC} = \overline{m_7}$$



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

---

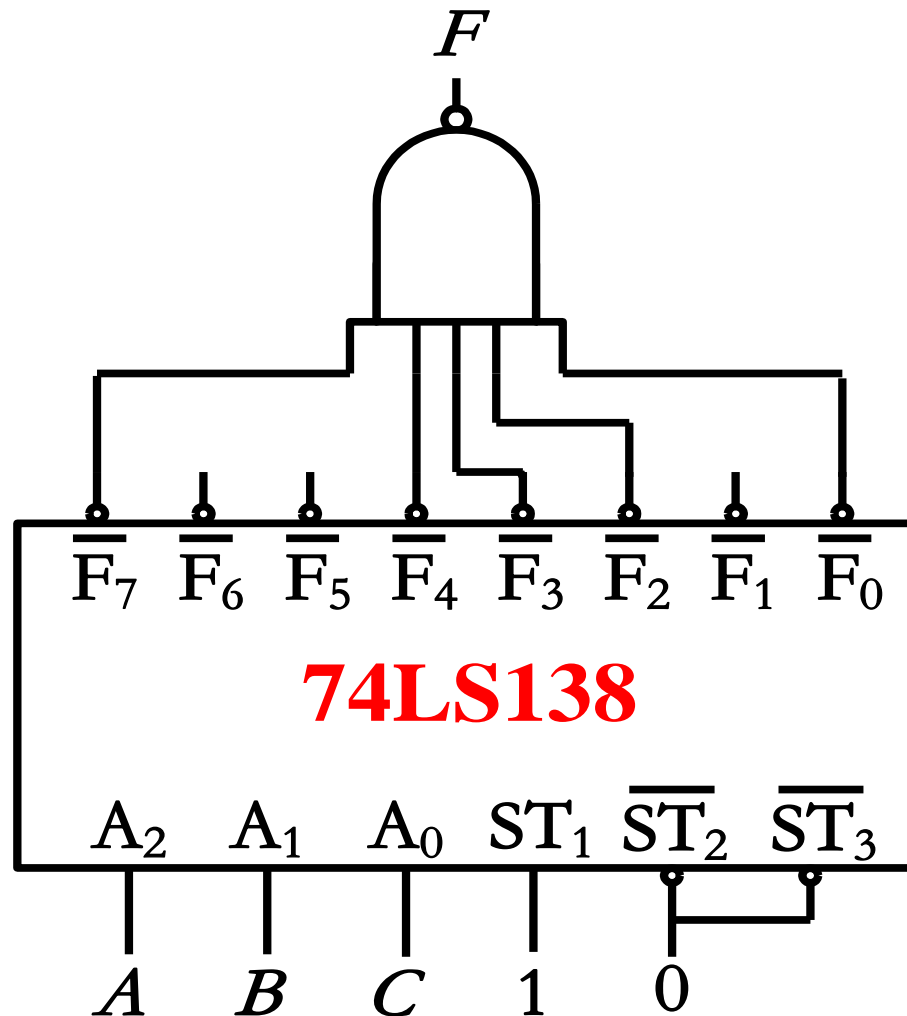
因为138的输出为低电平有效，因此将逻辑函数转换成最小项的反变量的形式。

$$\begin{aligned} F &= \sum m(0, 2, 3, 4, 7) = \overline{\overline{m_0 + m_2 + m_3 + m_4 + m_7}} \\ &= \overline{\overline{m_0} \overline{m_2} \overline{m_3} \overline{m_4} \overline{m_7}} = \overline{F_0 F_2 F_3 F_4 F_7} \end{aligned}$$

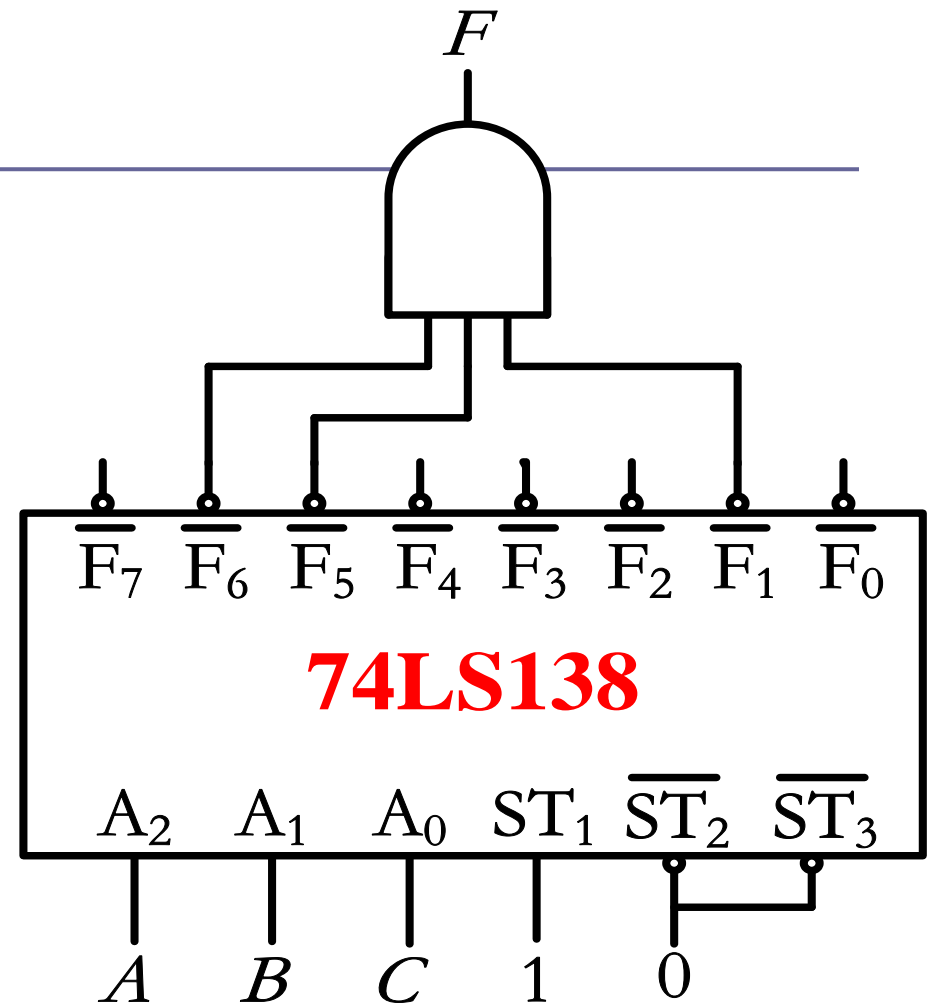
由表达式可见，各个输出 ( $m_i$ ) 用与非门连接便可实现逻辑函数  $F$ 。



$$F = \Sigma m(0, 2, 3, 4, 7) = \overline{\overline{F_0} \overline{F_2} \overline{F_3} \overline{F_4} \overline{F_7}}$$



也可通过与门连接



$$F = \sum_m (0, 2, 3, 4, 7) = \prod (1, 5, 6)$$

$$= M_1 M_5 M_6 = \overline{m_1} \overline{m_5} \overline{m_6}$$



# 用译码器实现一位全加器

输入			输出	
$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$
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

$$S_i = \sum m(1, 2, 4, 7)$$

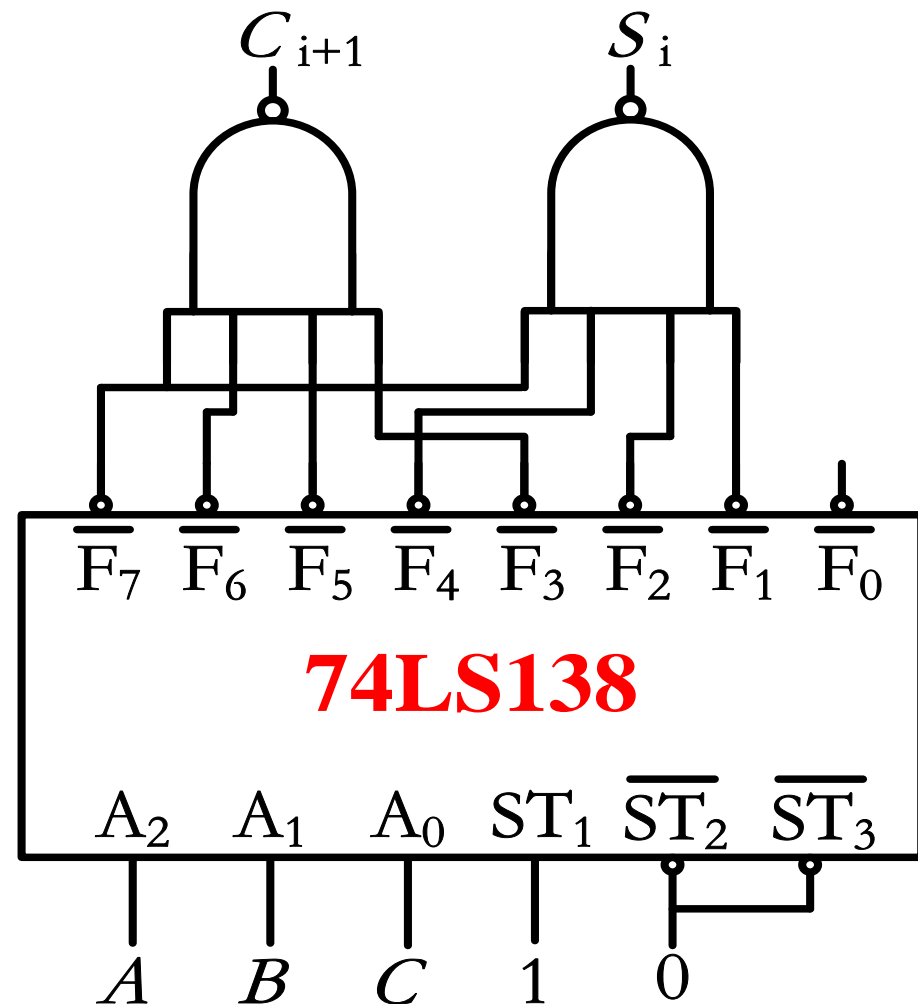
$$C_{i+1} = \sum m(3, 5, 6, 7)$$



## 选用一片74LS138及与非门实现

$$S_i = \sum m(1, 2, 4, 7)$$

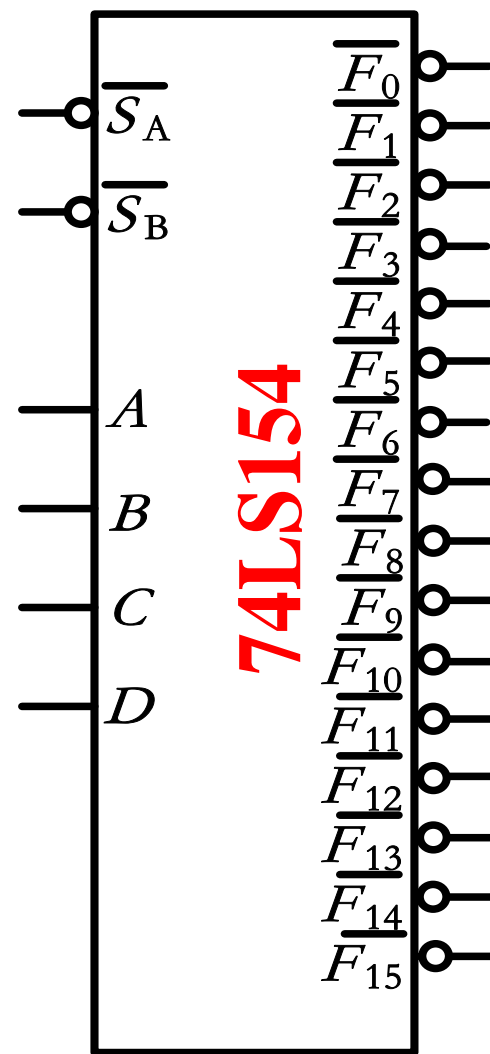
$$C_{i+1} = \sum m(3, 5, 6, 7)$$



# 用二进制译码器构成各种BCD译码器

任何一种BCD码通过4线-16线译码器进行译码。

74LS154为4线-16线译码器，输入原码、输出低电平有效。

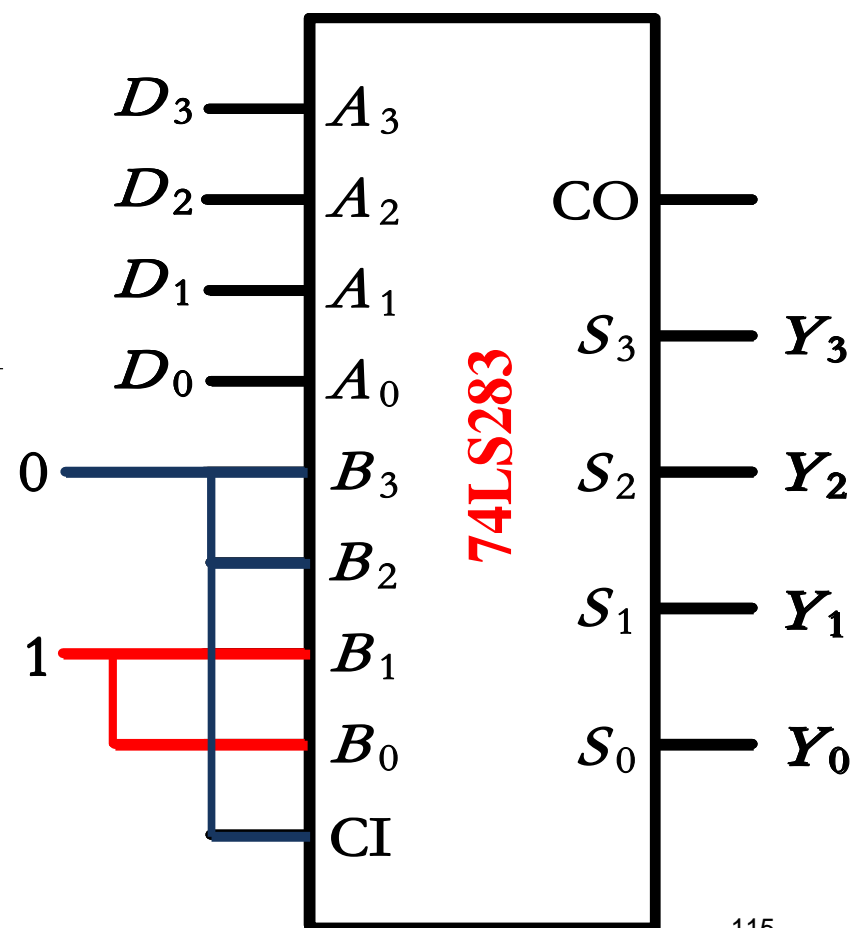




# 用全加器实现组合逻辑电路

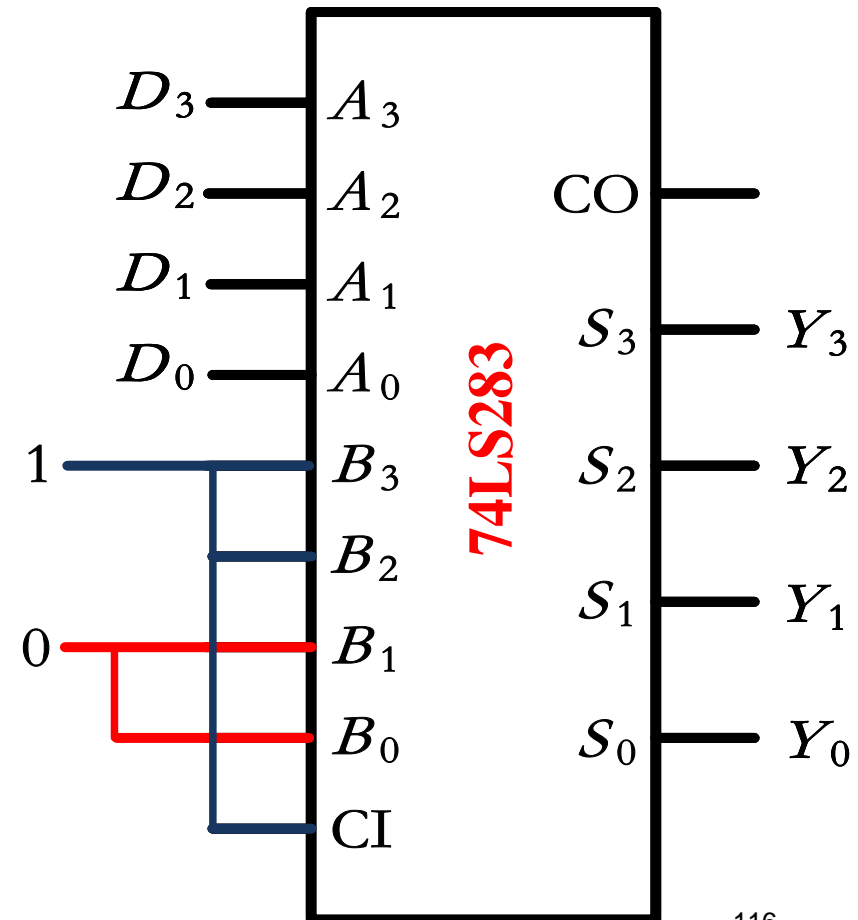
例：设计一个代码转换电路，将8421BCD码转换成余3码。

$$Y_3Y_2Y_1Y_0 = D_3D_2D_1D_0 + 0011$$



例：设计一个代码转换电路，将余3码转换成8421BCD码。

$$Y_3Y_2Y_1Y_0 = D_3D_2D_1D_0 + 1101$$



例：设计一个代码转换电路，将8421BCD码转换成5421BCD码。

8421BCD码				5421BCD码			
D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

$$Y_3 Y_2 Y_1 Y_0 = D_3 D_2 D_1 D_0 + 0000$$

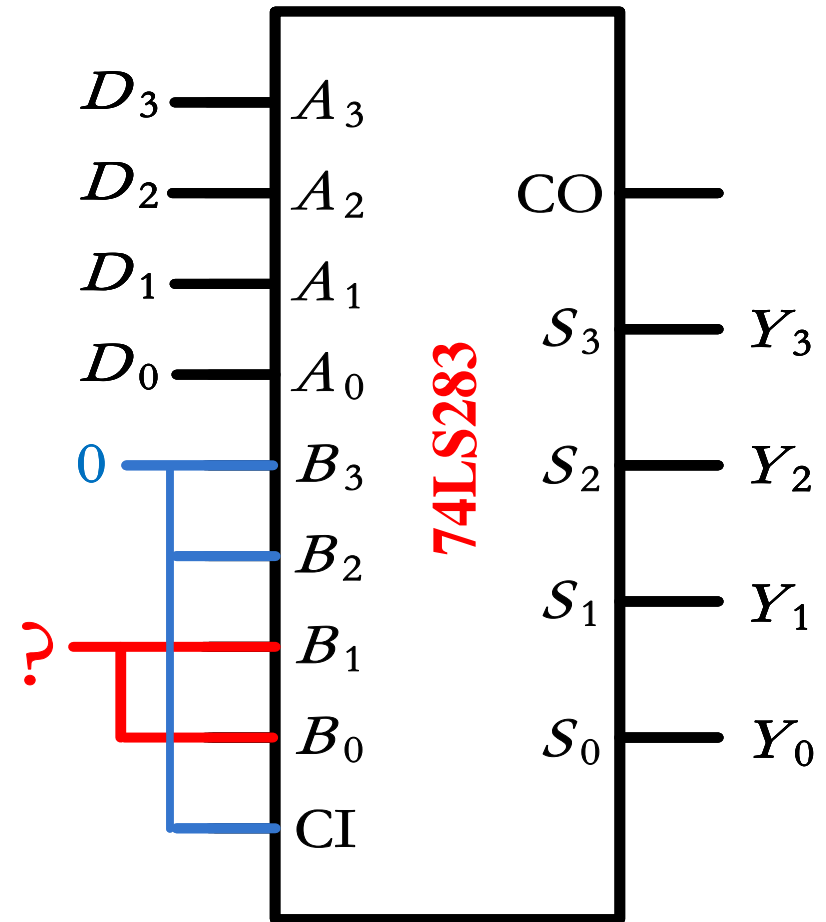
$$Y_3 Y_2 Y_1 Y_0 = D_3 D_2 D_1 D_0 + 0011$$



$$Y_3Y_2Y_1Y_0 = D_3D_2D_1D_0 + 0000$$

$$Y_3Y_2Y_1Y_0 = D_3D_2D_1D_0 + 0011$$

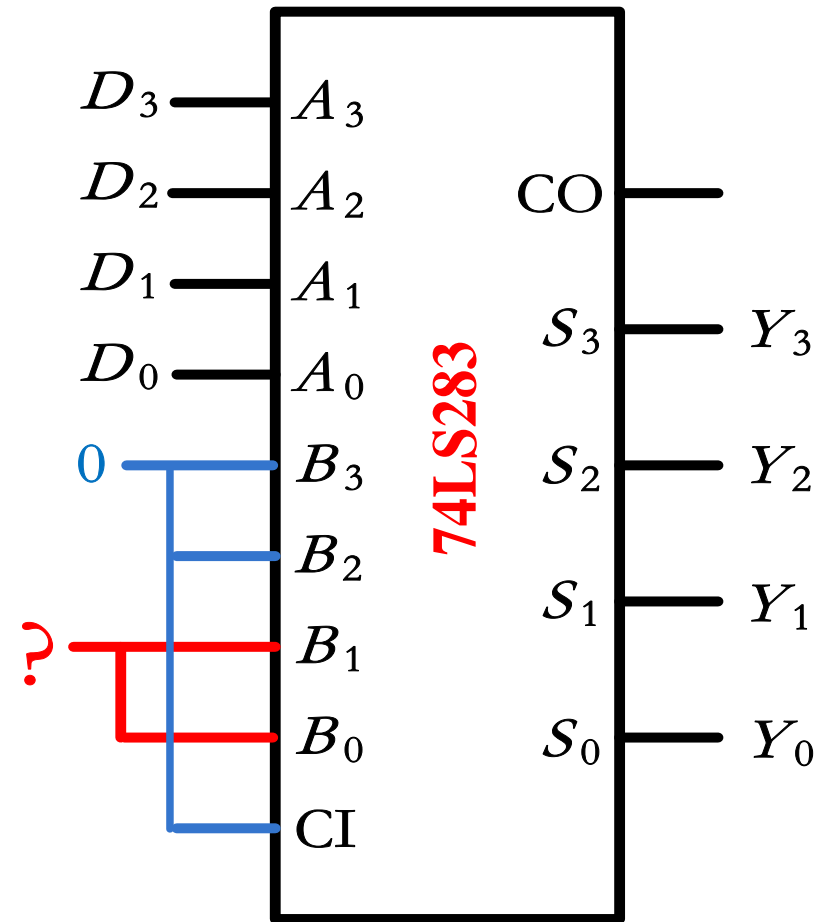
8421BCD码				5421BCD码			
$D_3$	$D_2$	$D_1$	$D_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0



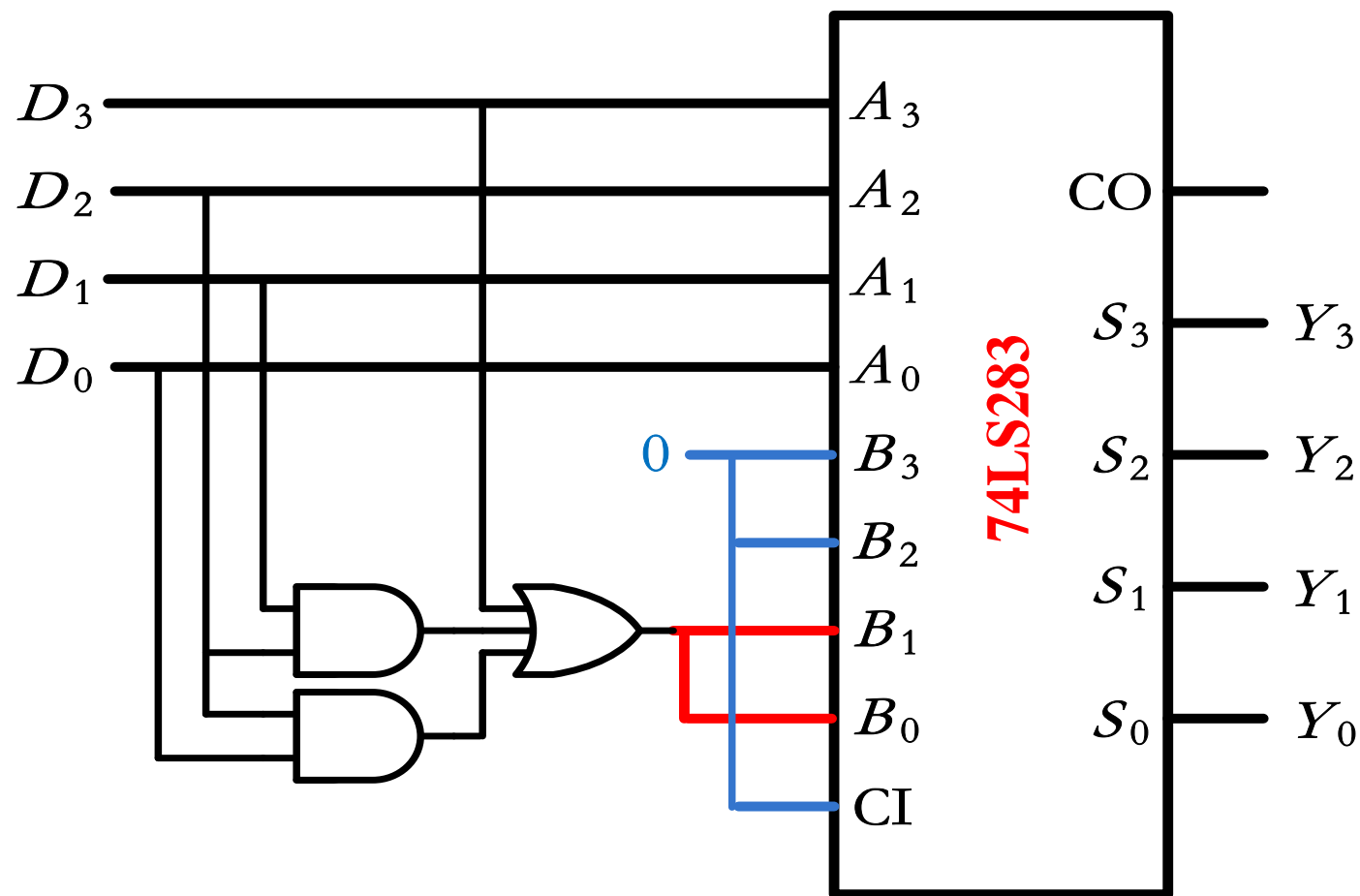
$$\overline{D_3}D_2\overline{D_1}D_0 + \overline{D_3}D_2D_1\overline{D_0} + \overline{D_3}D_2D_1D_0 + D_3\overline{D_2}\overline{D_1}\overline{D_0} + D_3\overline{D_2}\overline{D_1}D_0$$

$$D_3 + D_2D_1 + D_2D_0$$

8421BCD码					
D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	B <sub>1</sub>	B <sub>0</sub>
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	0	1	1	1



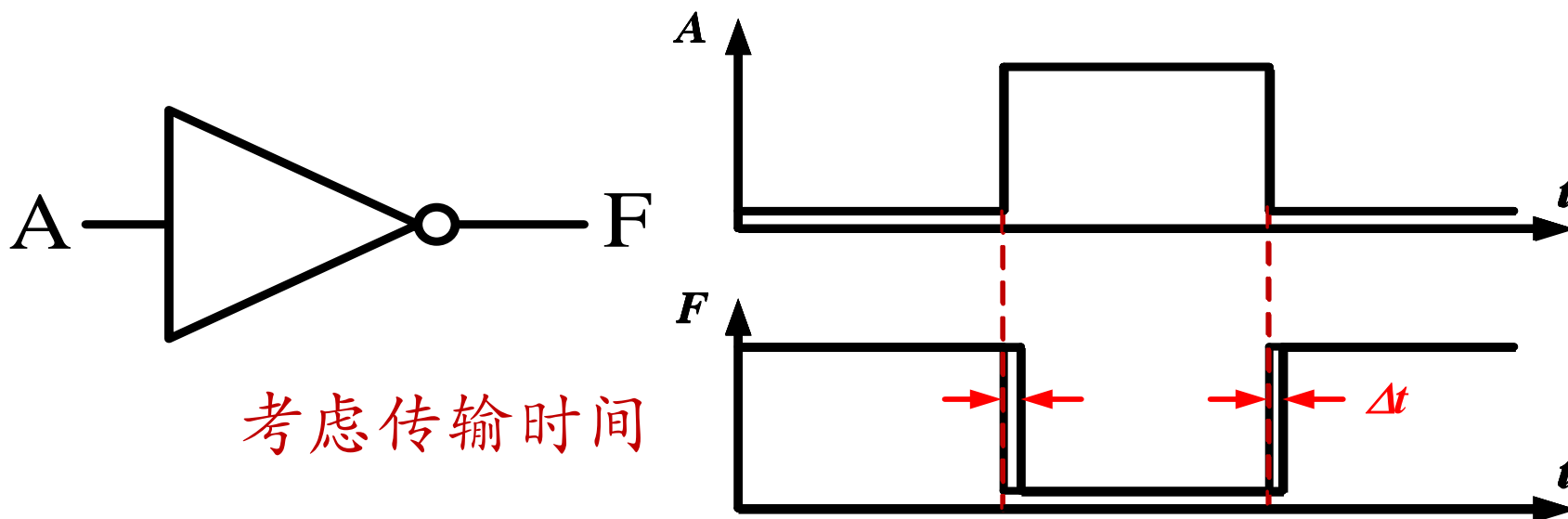
# 4位加法器实现8421BCD码转5421BCD码电路



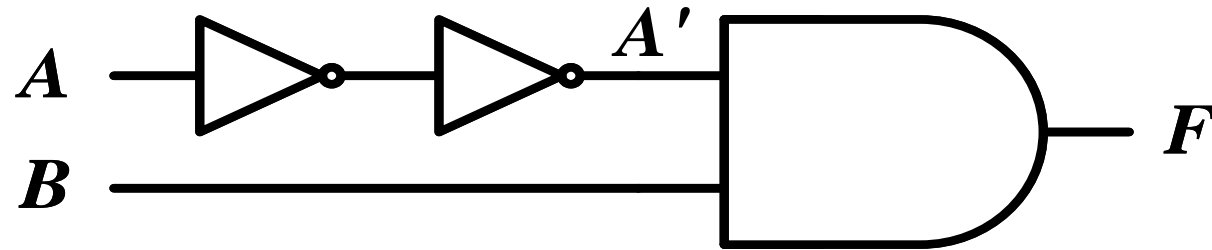
## 4.4 组合逻辑电路中的竞争和冒险

### 竞争和冒险产生的原因

信号通过门传输需要时间，即实际的逻辑门存在传输延迟时间。



# 竞争和冒险的产生

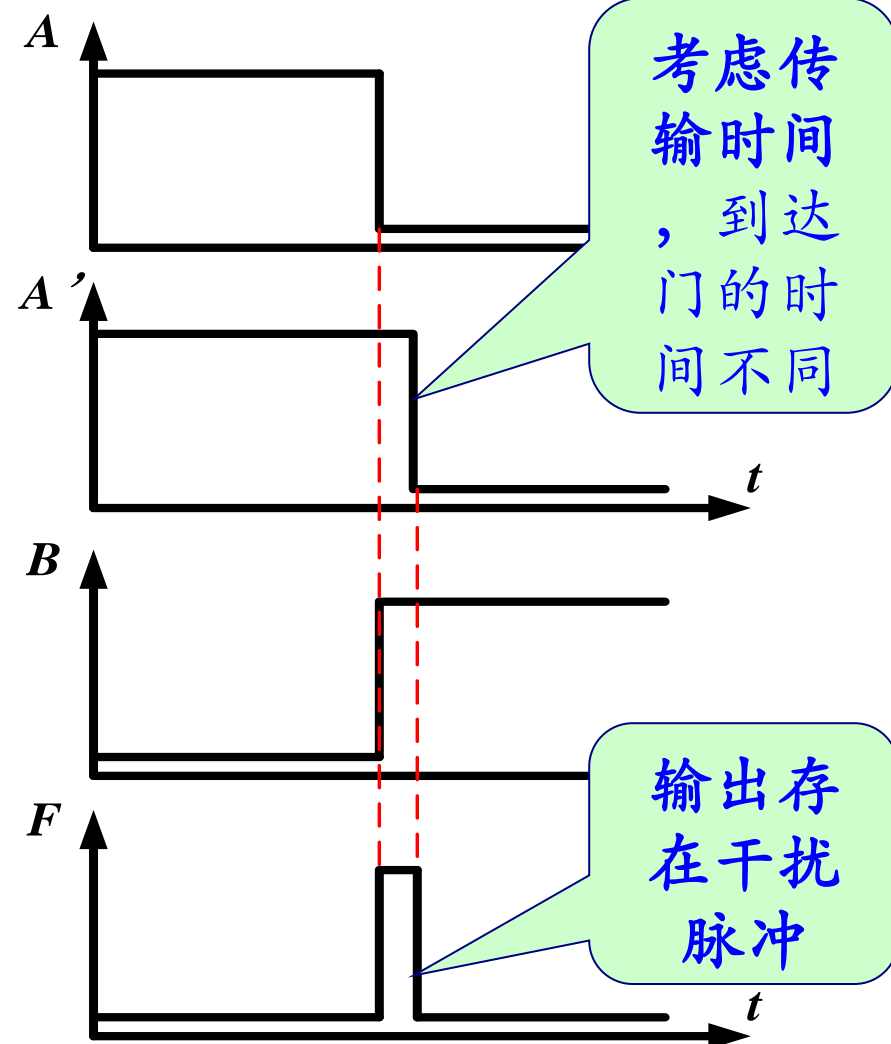
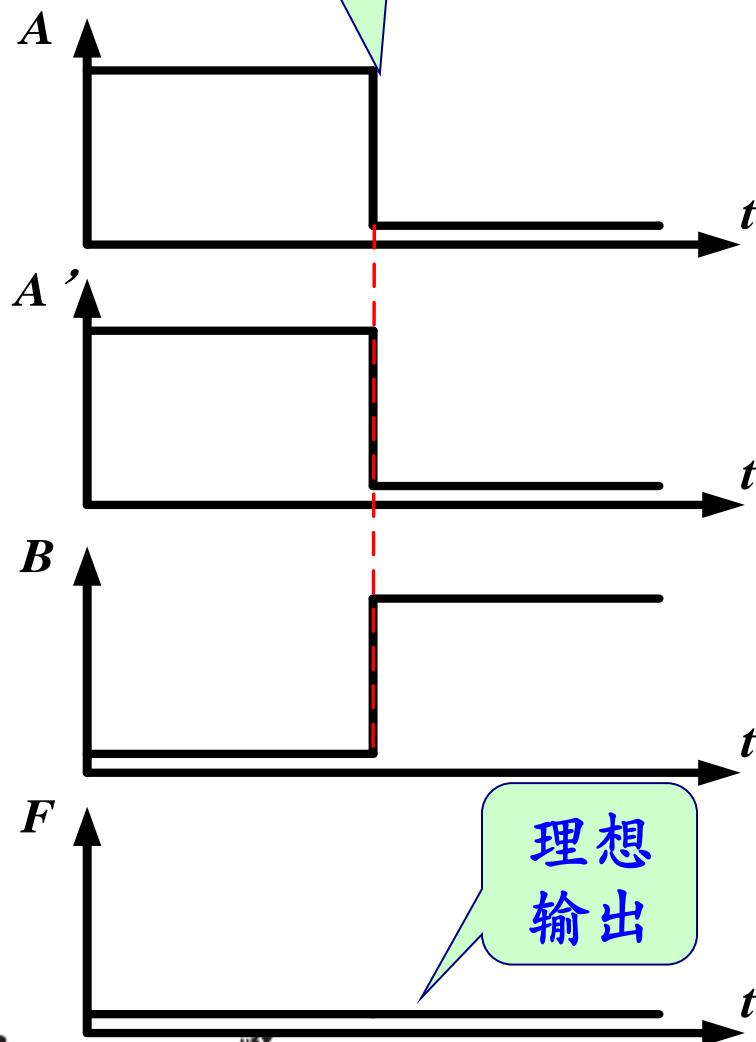
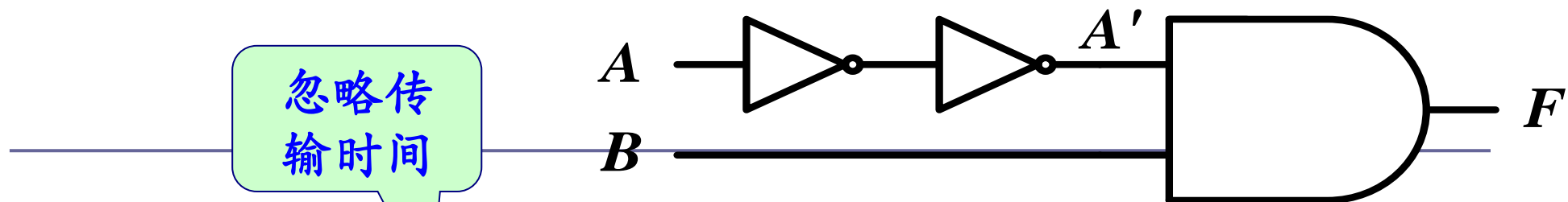


理想情况下， $F=AB$ 。

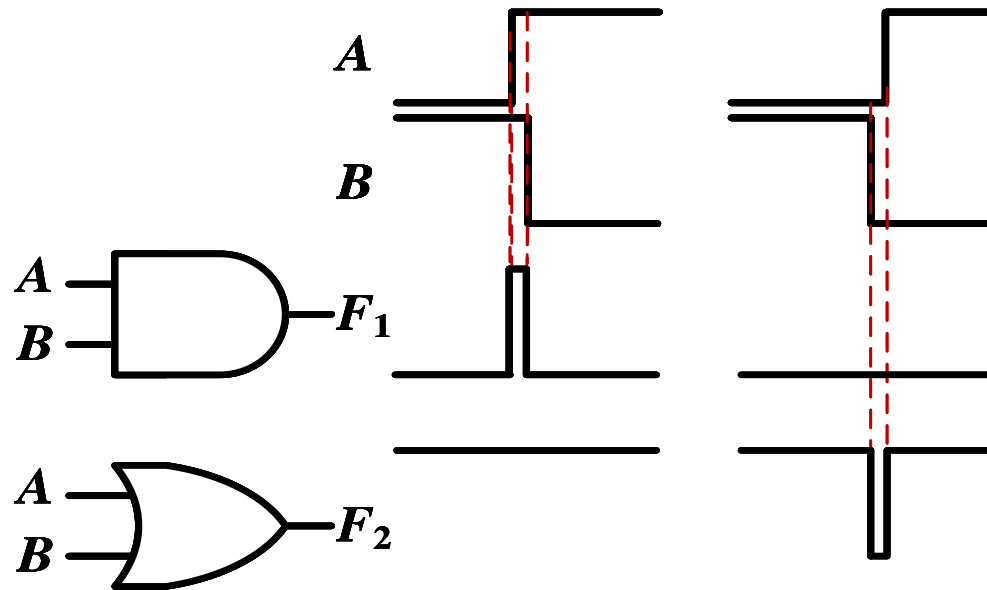
但考虑到 $A$ 、 $B$ 实际到达与门的时间不同，存在竞争，可能产生干扰脉冲，称为冒险。







## 与门和或门产生竞争、冒险的原因



输入有竞争现象时，输出不一定都产生冒险。

冒险分为逻辑冒险和功能冒险两种。



## 逻辑冒险及消除

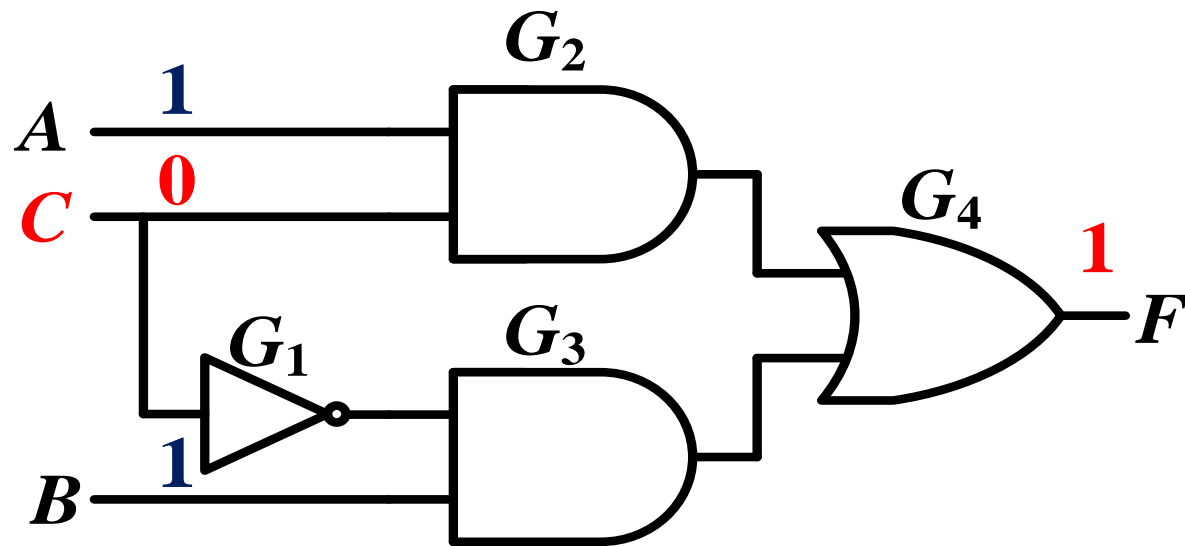
---

当多个输入信号中某一个发生变化时，由于此信号在电路中经过的途径不同，使到达电路某个门的多个输入信号之间产生时间差，即存在由所有的逻辑部件的延迟时间引起的竞争，称为“逻辑竞争”。

由此产生的冒险为“逻辑冒险”。

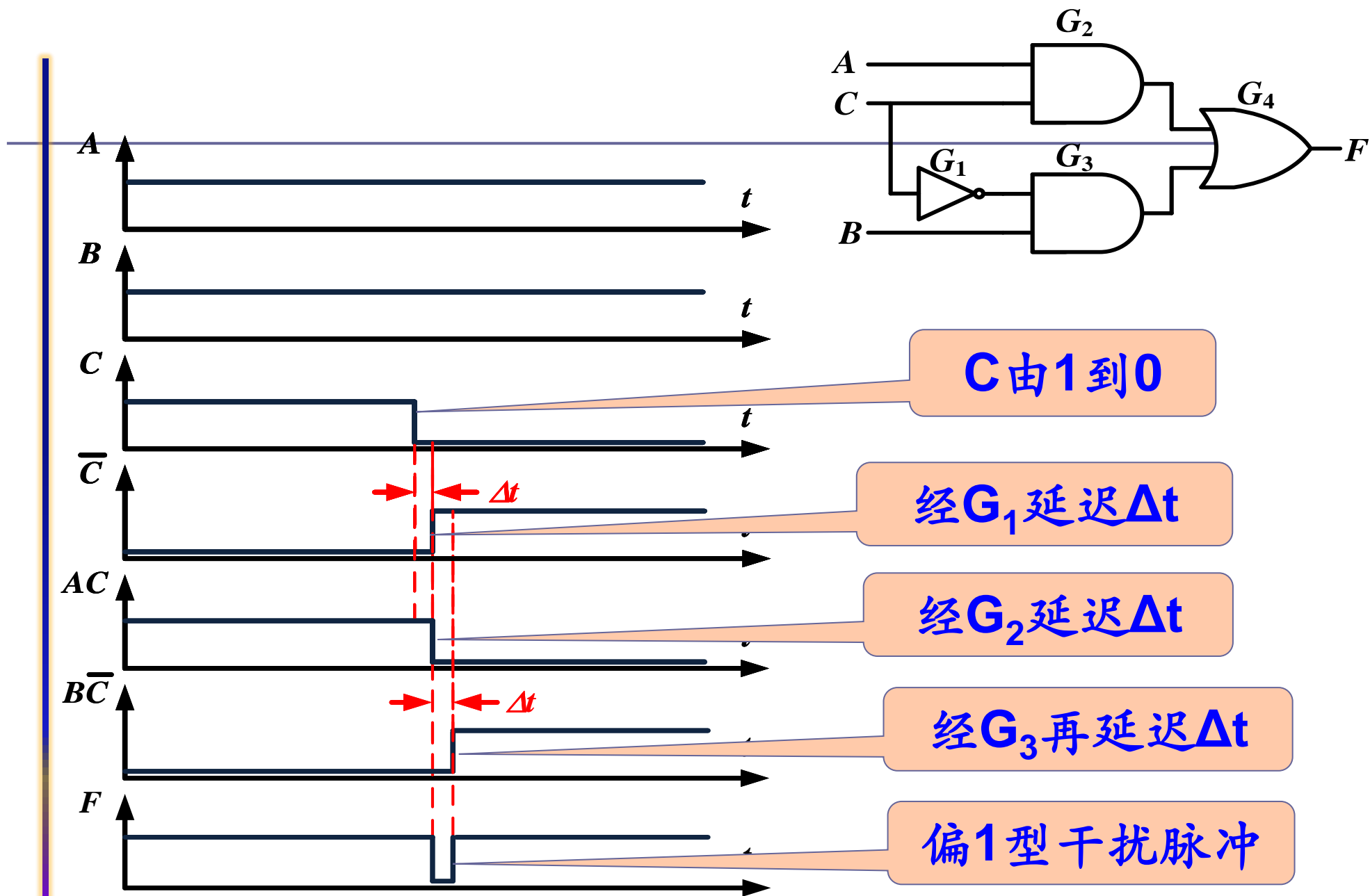


$$F = AC + B\bar{C}$$

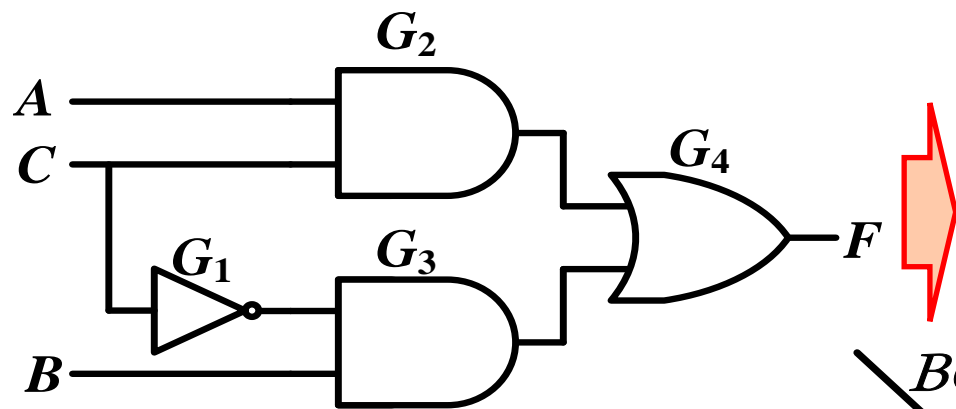


变量C发生变化时，经过门的数量不同，达到G4门的时间就不同，存在“逻辑竞争”。





# 逻辑冒险的消除



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

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

两圆圈相切时，可能产生逻辑冒险

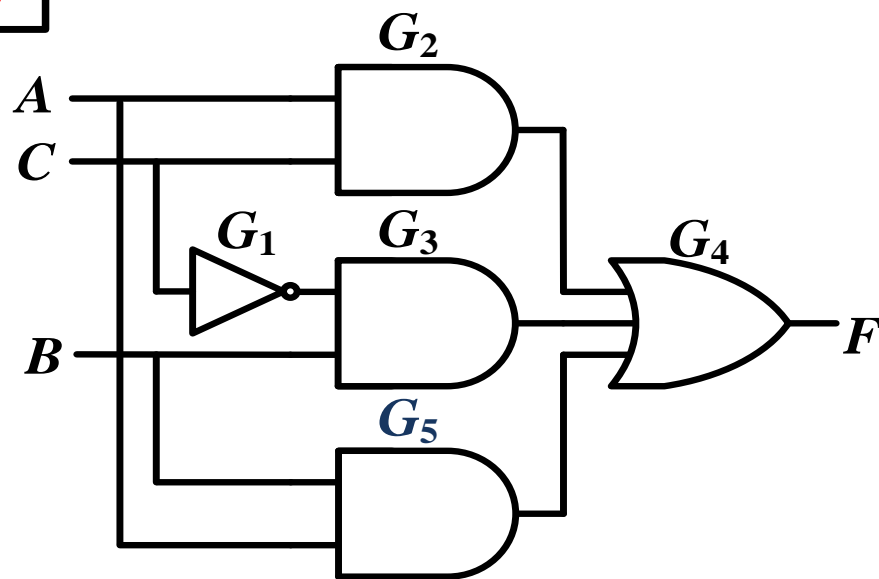


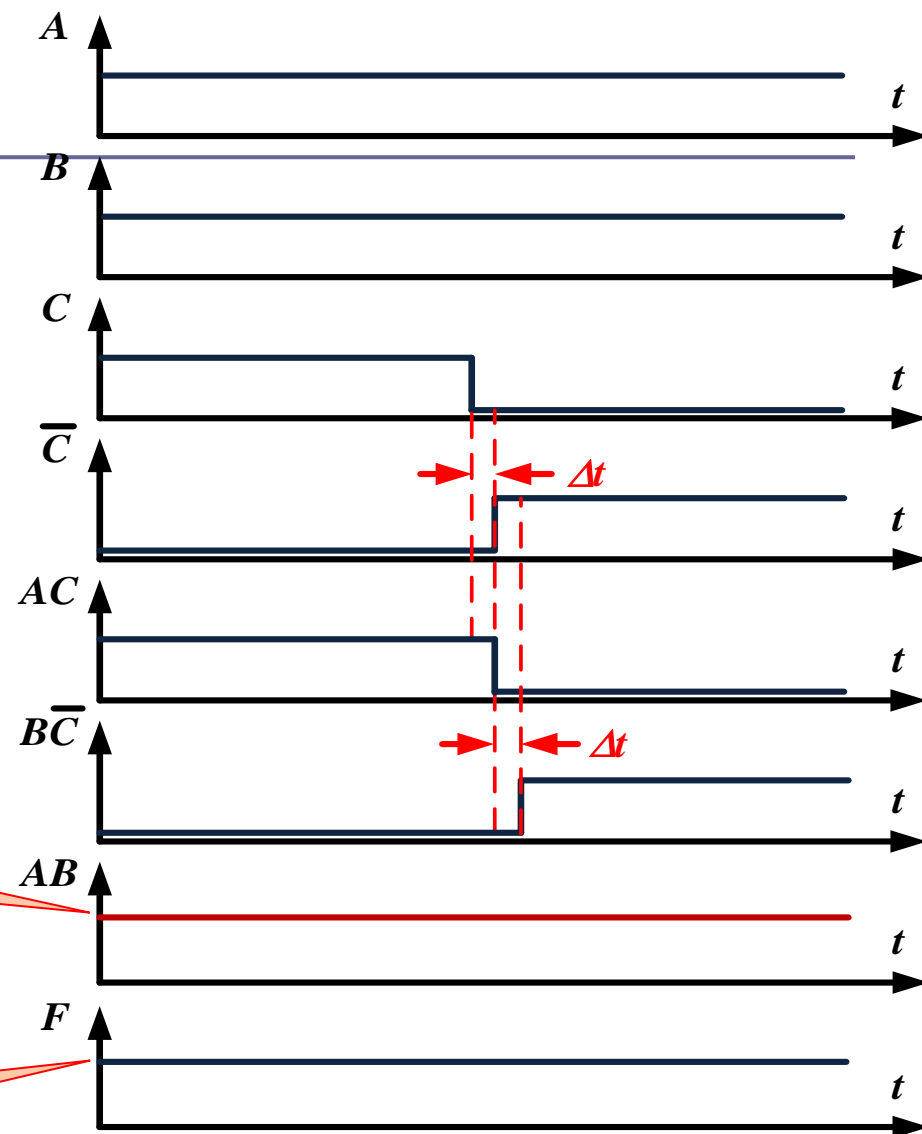
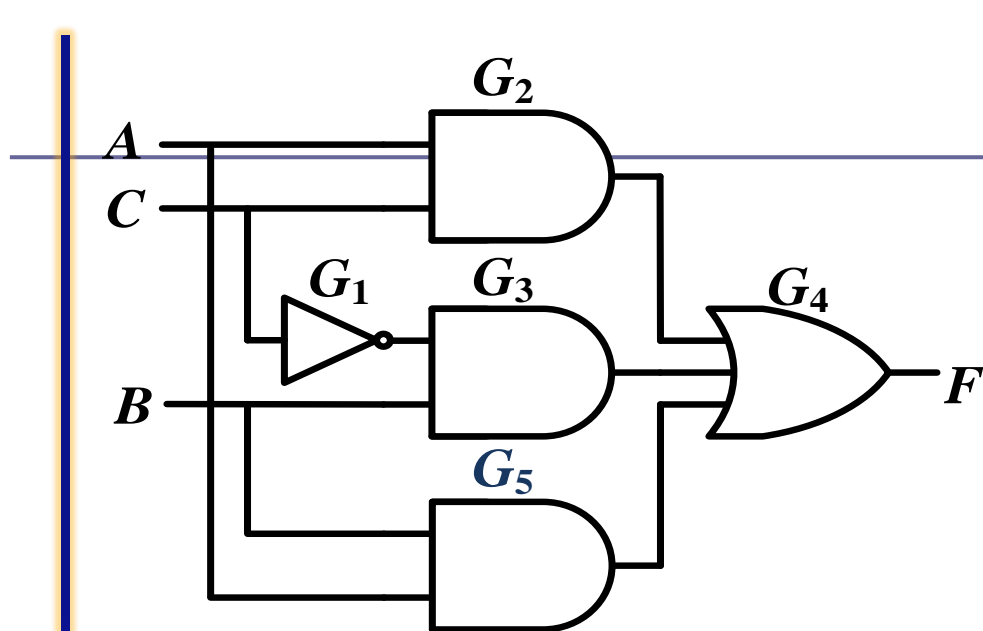
# 利用增加冗余项消除逻辑冒险

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

增加冗余项  $AB$

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





**AB保持高电平**

**F保持高电平**





# 功能冒险及消除

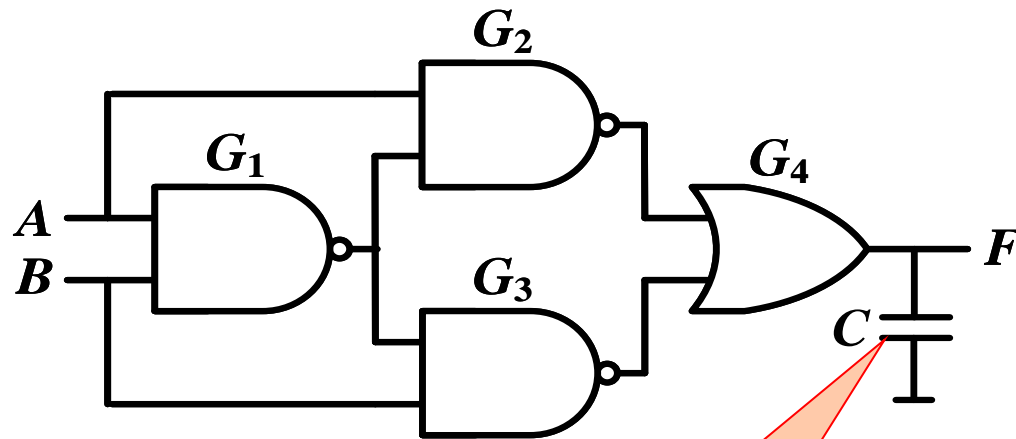
---

在组合电路的输入端，当有几个变量变化时，由于其变化的快慢不同，传递到某个门的输入端必然存在时间差，这种现象叫作“功能竞争”。

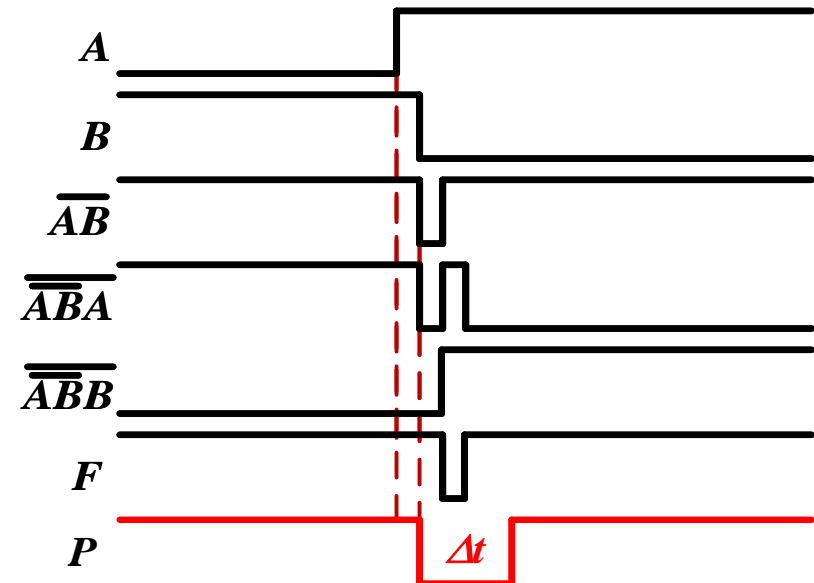
由此产生的冒险为“功能冒险”。



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



消除干扰措施之一：  
加滤波电容



消除干扰措施之二：  
引入禁止脉冲



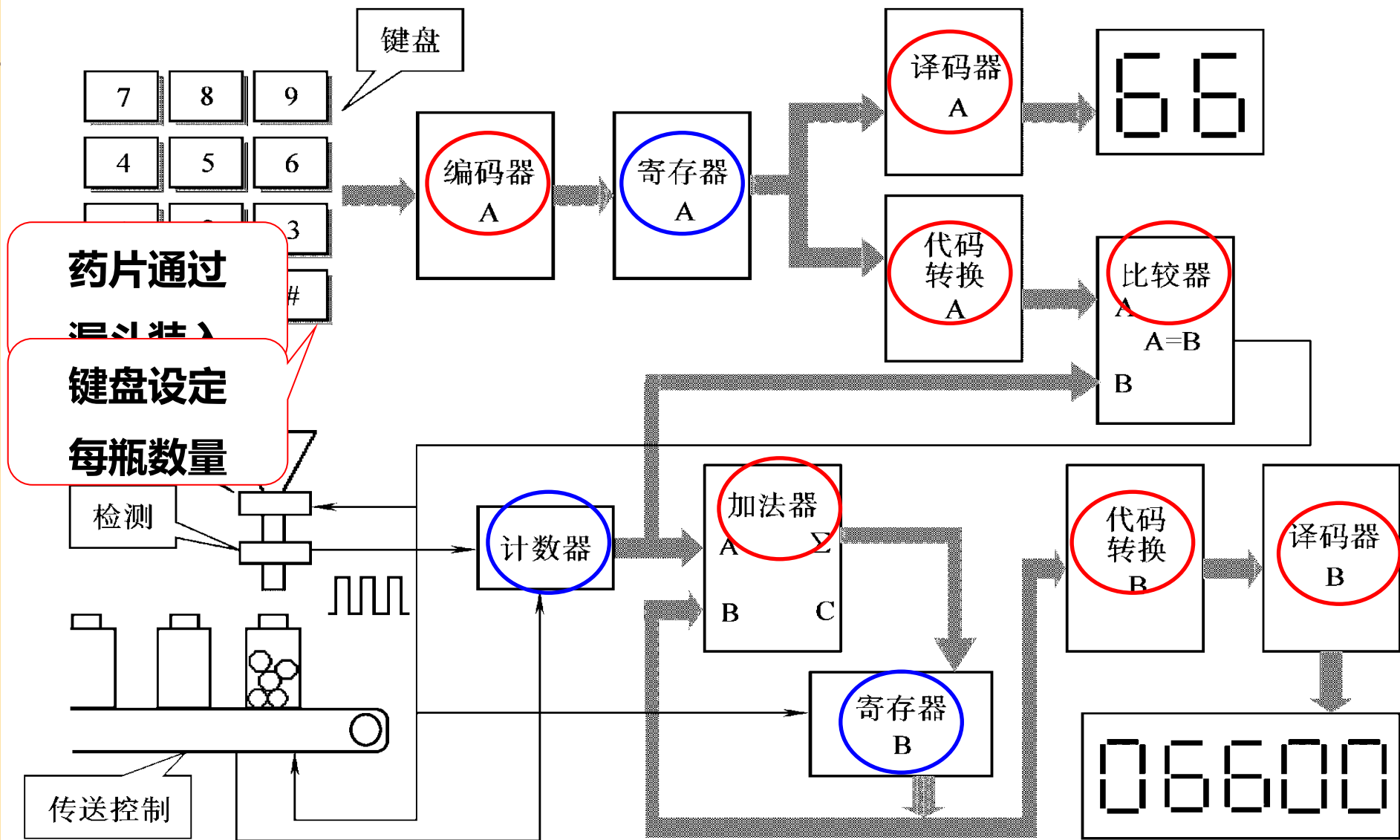
## 4.5 组合电路的系统应用

---

实例：

工厂的产品（药片）通过漏斗装入位于传送带上的药瓶之中，每瓶中药片的数量可以预先设定，要求通过数码管显示所设每瓶中药片数量及若干瓶中药片总的数量。





## 4.6 用Multisim设计和分析组合电路

---

课下练习



# 本章小结

---

## 组合逻辑电路的特点

## 组合逻辑电路的一般分析方法

- 1、写表达式;
- 2、列真值表;
- 3、说明功能。



# 组合逻辑电路的一般设计方法

- 1、列真值表；
- 2、写表达式、变换；
- 3、画逻辑图。



# 常用组合逻辑电路

- 1、编码器；
- 2、译码器；
- 3、数据选择器；
- 4、加法器；
- 5、数值比较器

## 中规模集成电路的应用

电路分析、自扩展和实现逻辑功能





# 组合逻辑电路中的竞争—冒险现象

