



第9章 组合逻辑电路

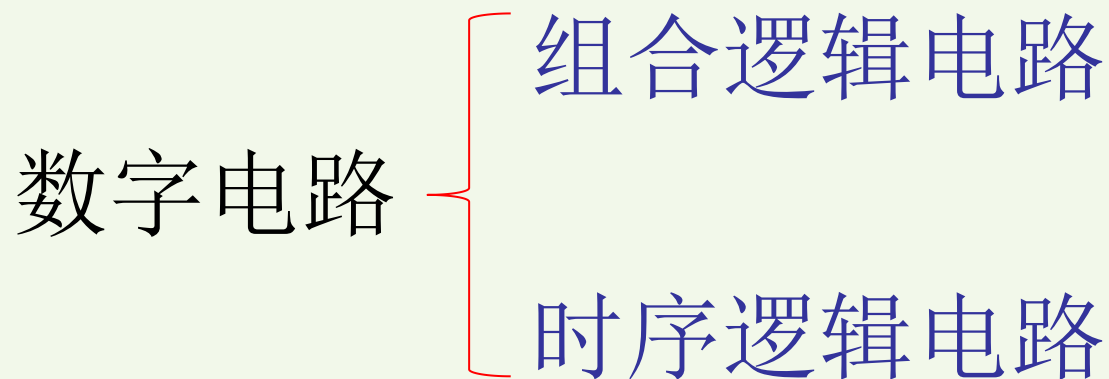
§9.1 组合逻辑电路的分析与设计方法

§9.2 常用组合逻辑电路介绍

§9.3 组合逻辑电路中的竞争与冒险



9.1.1 组合逻辑电路概述

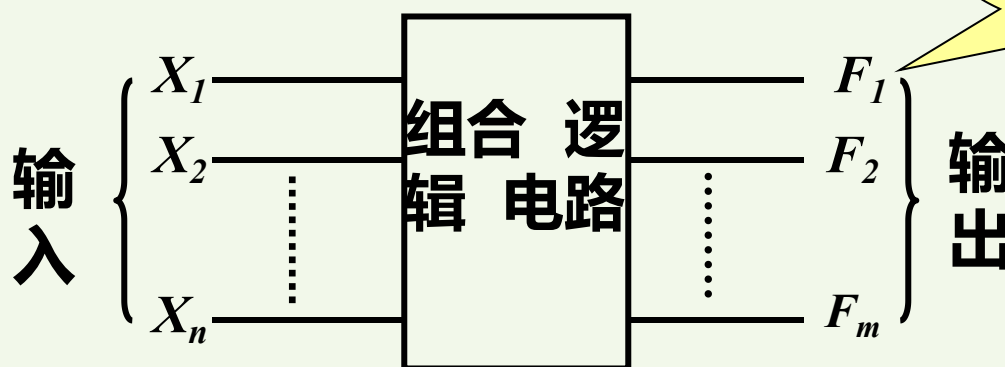


◆组合逻辑电路的结构特点：

(1) 输出与输入之间**无反馈**

(2) 不包括**记忆（存储）元件**

◆组合逻辑电路的框图：



◆组合逻辑电路的功能特点：

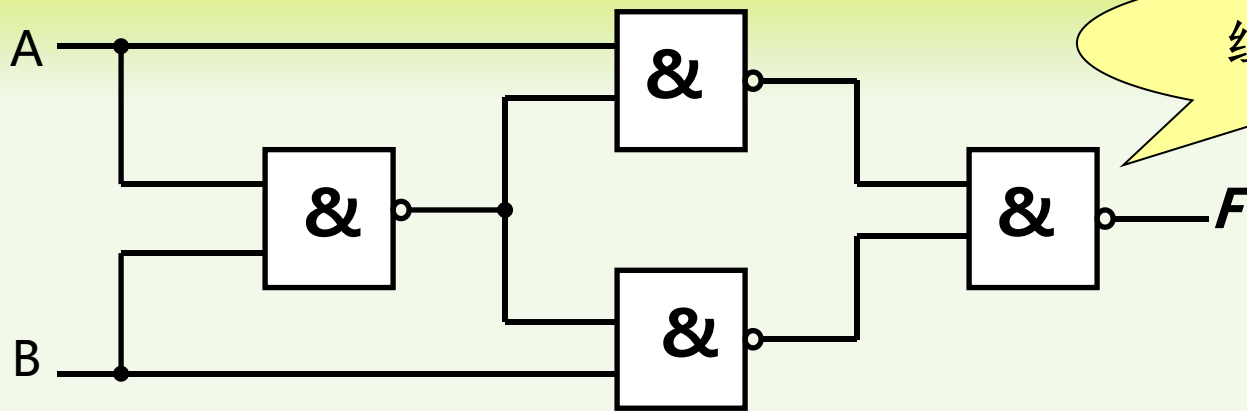


电路的输出只与电路**当前**的输入有关，即**无记忆性**。

◆组合逻辑电路输出函数的一般逻辑表达式：

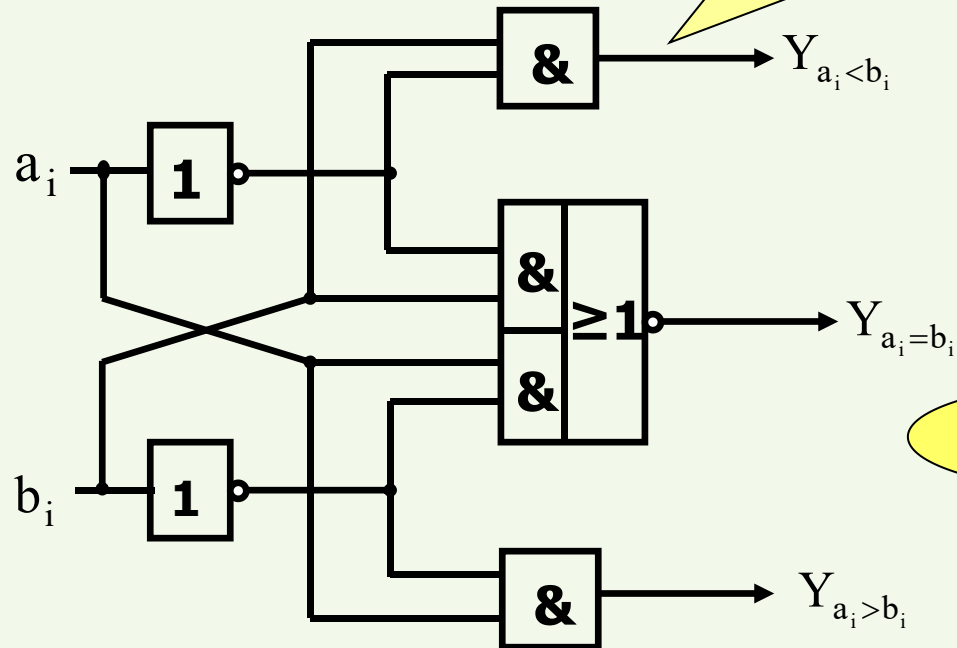
$$\begin{cases} F_1 = f_1(X_1, X_2, \dots, X_n) \\ F_2 = f_2(X_1, X_2, \dots, X_n) \\ \vdots \\ F_m = f_m(X_1, X_2, \dots, X_n) \end{cases}$$

各输出函数**仅由**
输入确定，彼此
相互独立



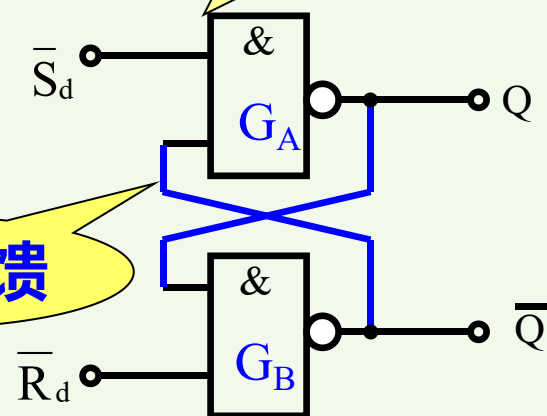
组合逻辑电路

组合逻辑电路



时序逻辑电路

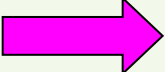
存在反馈





9.1.2 组合逻辑电路的分析

分析要求:

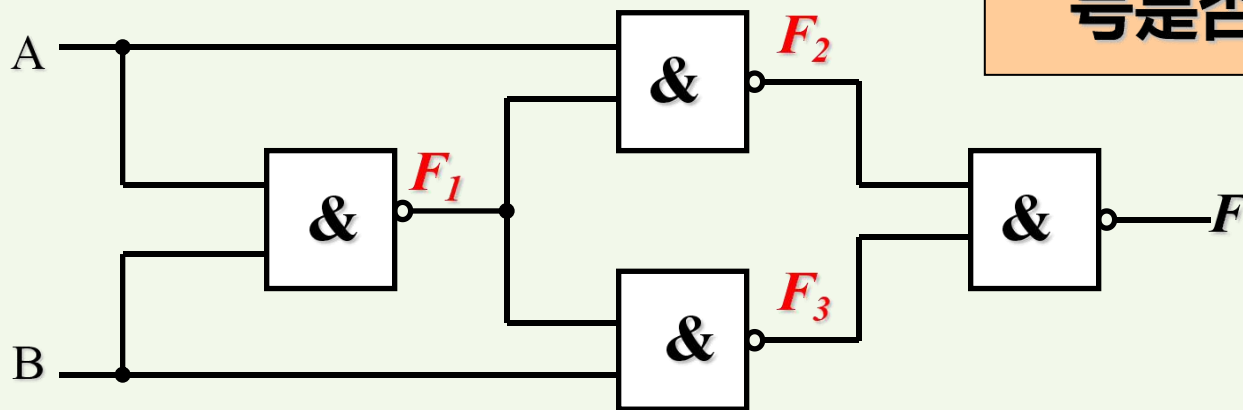
已知电路结构
(逻辑图)  **求**电路的功能
(输出输入逻辑关系)

分析步骤:

- 由逻辑图得出**逻辑函数表达式**, 并化简;
- **列真值表**;
- **确定逻辑功能**。

例1 分析图示电路的逻辑功能。

“异或”逻辑
可用来判断两信号是否一致。



解： ● 写出输出函数式

$$\because F_1 = \overline{AB}$$

$$F_2 = \overline{AF_1} = \overline{A\overline{AB}}$$

$$F_3 = \overline{BF_1} = \overline{B\overline{AB}}$$

$$\therefore F = \overline{F_2 F_3} = \overline{\overline{A\overline{AB}} \cdot \overline{B\overline{AB}}}$$

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

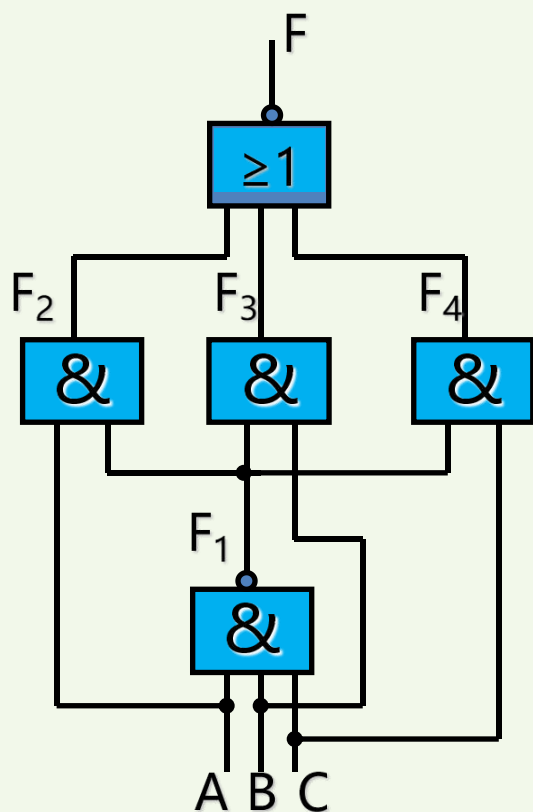
$$= A(\overline{A} + \overline{B}) + B(\overline{A} + \overline{B})$$

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

$$= A \oplus B$$



例2：分析图示电路的逻辑功能。



解： ● 写出输出函数式

$$F_1 = \overline{ABC} \quad F_2 = AF_1 \quad F_3 = BF_1 \quad F_4 = CF_1$$

$$F = \overline{F_2 + F_3 + F_4} = \overline{\overline{ABC}(A + B + C)}$$

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

● 列真值表

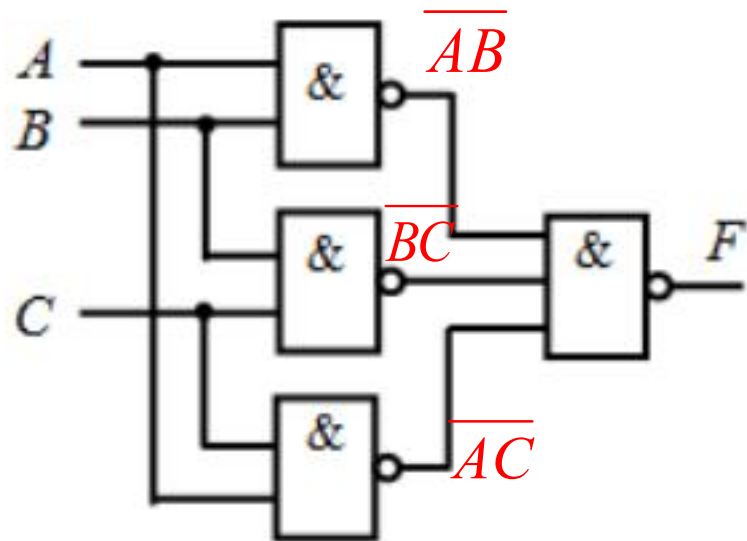
● 分析电路功能

<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

当*A*、*B*、*C*全为0
或1时，*F*为1，
否则*F*为0。

“判一致电路”

例3：分析图示电路的逻辑功能。



解：●写出输出函数式

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

●分析电路功能

●列真值表

<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>
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

当*A*、*B*、*C*中为1的个数≥2时，输出为1
或1时，*F*为1，
“多数表决电路”

9.1.3 组合逻辑电路的设计

设计要求:

已知逻辑功能 \longrightarrow 求逻辑电路图

设计步骤:

- 根据逻辑功能, 进行**逻辑抽象**, 即确定逻辑变量 (输入和输出), 并对变量赋值;
- 根据功能列出**真值表**;
- 写出**逻辑函数表达式**, 化简;
- 画**逻辑电路图**。



例1：设计一个监视交通信号灯工作状态的逻辑电路。

电路由红、黄、绿三盏灯组成。正常工作时，任何时刻必有一盏而且只允许有一盏灯点亮；其它点亮状态时电路故障，要求发出故障信号。（要求用“与非”门实现）

解： 逻辑抽象

输入变量： A 、 B 、 C
(分别表示红、黄、绿三盏灯)

输出变量： F (表示报警与否)

输入 {
“1”：灯亮
“0”：灯不亮

输出 {
“1”：报警
“0”：不报警

列真值表

A	B	C	F
0	0	0	1
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

写出逻辑函数式

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

或： $F(A, B, C) = \sum_m(0, 3, 5, 6, 7)$

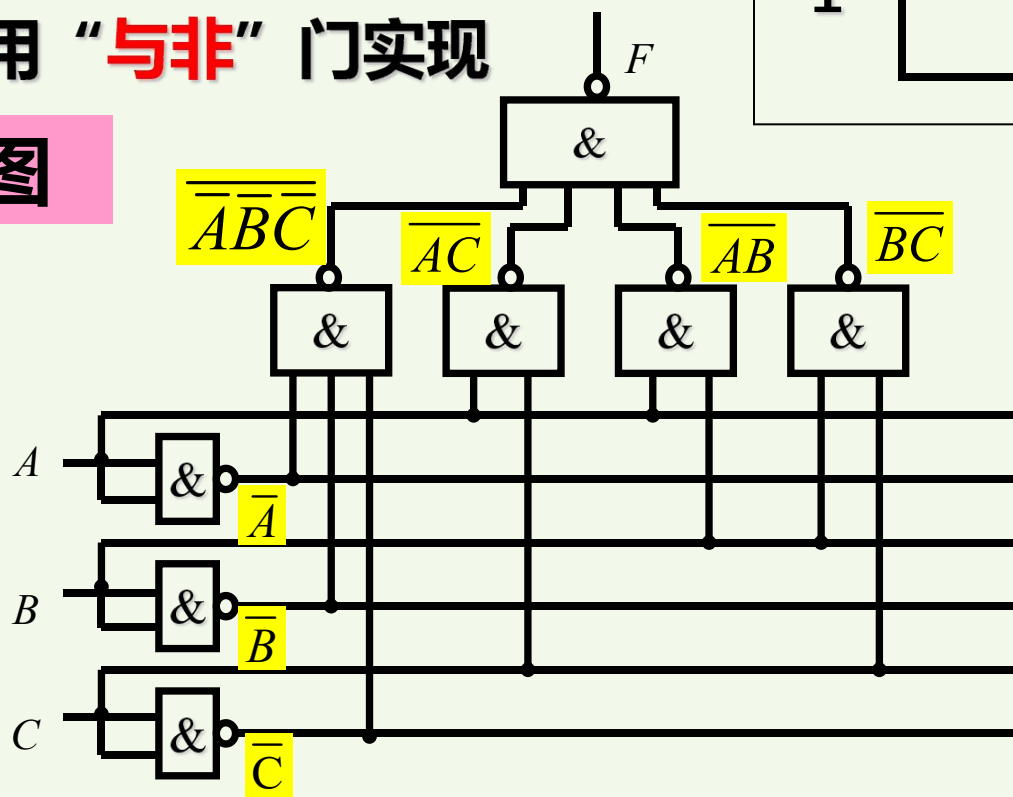
化简

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

$$= \overline{\overline{A}\overline{B}\overline{C}} \cdot \overline{AC} \cdot \overline{AB} \cdot \overline{BC}$$

要求用 “与非” 门实现

逻辑图



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



例2：举重比赛有 ABC 三个裁判及一个主裁判 D 。当主裁判认为合格时算为二票，而 ABC 裁判认为合格时分别算为一票。用“与非”门设计多数通过 (≥ 3 票) 的表决电路。

解：(1) 逻辑抽象

输入变量为 A 、 B 、 C 、 D ，为1表示合格，为0表示不合格，
输出变量为 F ，多数通过时 $F=1$ ，否则 $F=0$ ，

(2) 列真值表

(3) 写出逻辑函数表达式，并化简

$$F(A, B, C, D) = \sum_m (3, 5, 7, 9, 11, 13, 14, 15)$$

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



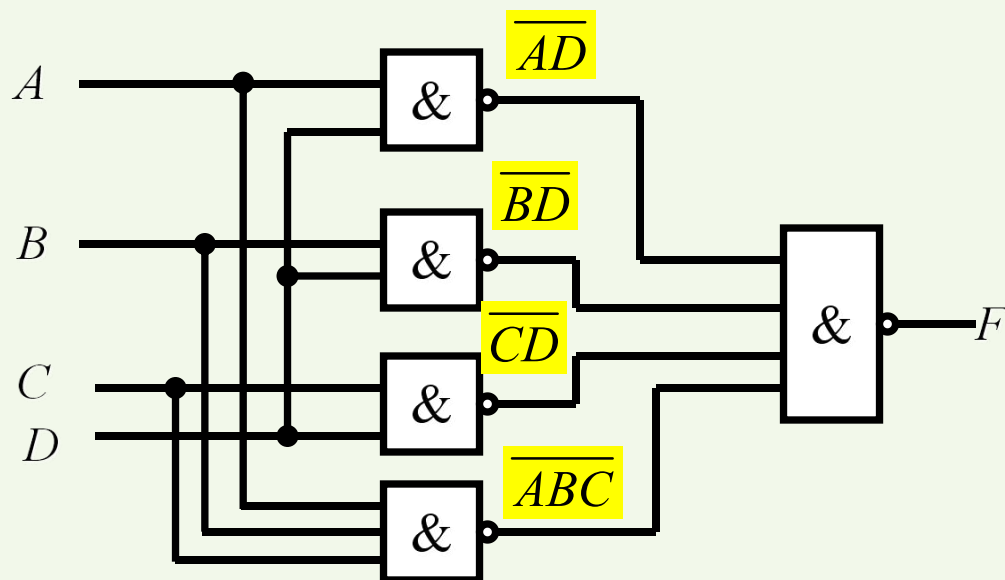
$$F(A, B, C, D) = \sum_m (3, 5, 7, 9, 11, 13, 14, 15)$$

$$F = AD + BD + CD + ABC$$

(4) 用“与非”门实现

$$\begin{aligned} F &= \overline{\overline{AD + BD + CD + ABC}} \\ &= \overline{\overline{AD} \overline{BD} \overline{CD} \overline{ABC}} \end{aligned}$$


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





9.1 组合逻辑电路的分析与设计小结

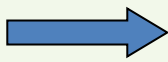
分析

已知电路**结构**  求电路的**功能**
(逻辑图) (输出输入逻辑关系)

分析步骤:

- 由逻辑图得出逻辑函数表达式，并化简
- 列真值表
- 确定逻辑功能

设计

已知逻辑**功能**  求逻辑**电路图**

设计步骤:

- 进行逻辑抽象，即确定逻辑变量（输入和输出）
- 列出真值表
- 写出逻辑函数表达式并化简
- 画逻辑电路图



课堂练习
第一节1, 2, 4, 5,
7, 8





9.2 常用组合逻辑电路介绍

9.2.1 编码器

9.2.2 译码器

9.2.3 数据选择器

9.2.4 加法器

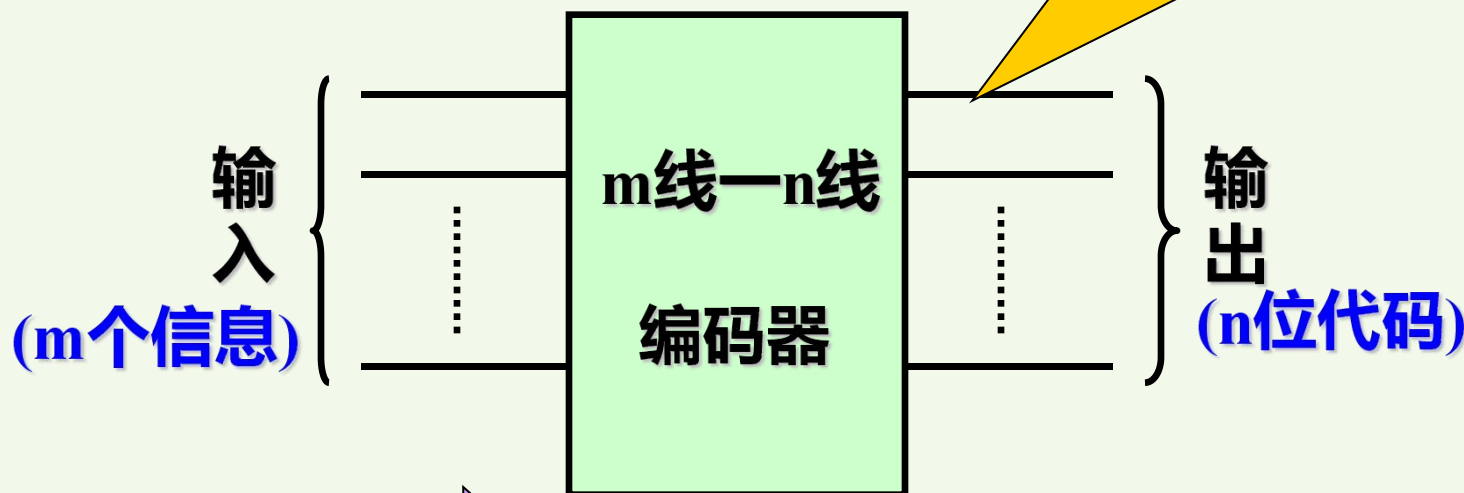


9.2.1 编码器

◆编码器的功能

能够实现用二进制代码表示各种符号、数字和信息这一编码过程的电路。

◆编码器的结构框图



m与n的关系

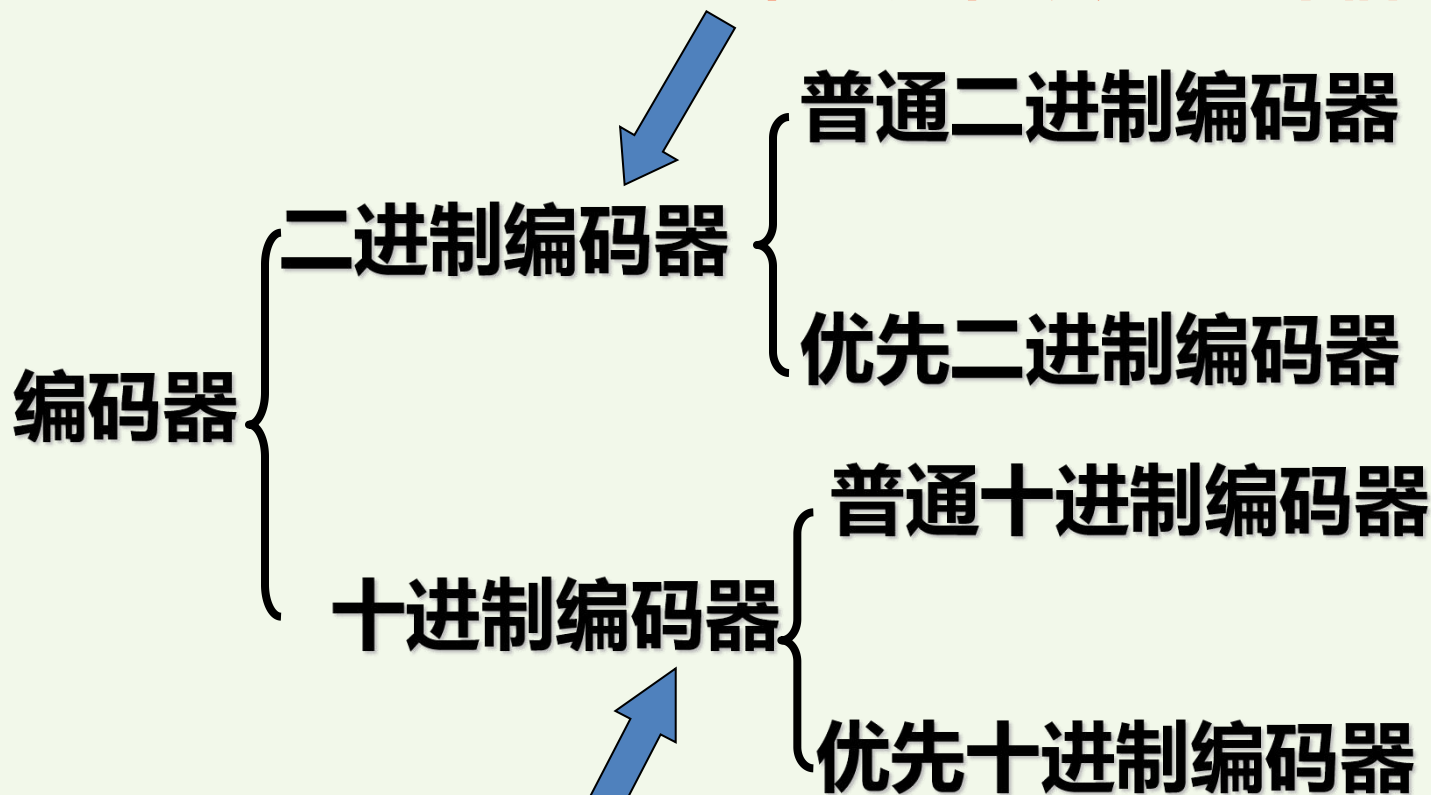
$$m \leq 2^n$$

如8线-3线编码器
16线-4线编码器



◆ 编码器的分类

由n位二进制数表示 2^n 个信号的编码电路



将十个输入信号编成对应的8421BCD码的电路



➤ 普通编码器 ---- 互斥编码器

编码器任何时刻，只允许一个输入信号有效，不允许两个或两个以上的输入信号同时有效。

逻辑符号



当输入的信号编码中出现多个1时应该如何处理呢？

编码表

I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

表示 I_0 信号出现



➤ 优先编码器

当有两个或两个以上的信号同时输入编码电路，电路只对其中一个优先级别高的信号进行编码。



编码表

I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	×	0	1
0	1	×	×	1	0
1	×	×	×	1	1

- 结论：**
- (1) 该编码器输入信号的有效电平为高电平
 - (2) 电路的优先级别从高到低分别为 I_3 、 I_2 、 I_1 、 I_0

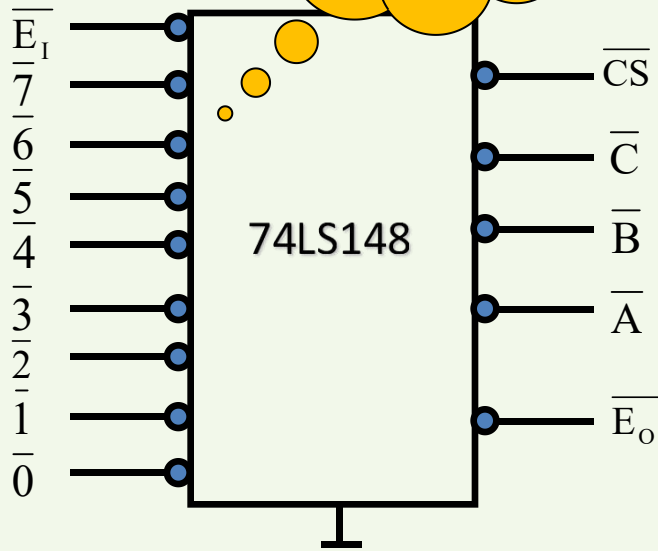
• 优先二进制编码器74LS148

$\overline{0} \sim \overline{7}$ 是信号输入端
当输入为0时，表明该
信号出现；
 $\overline{7}$ 输入优先权最高

代码输出端
反码输出

逻辑符号

低电平
有效

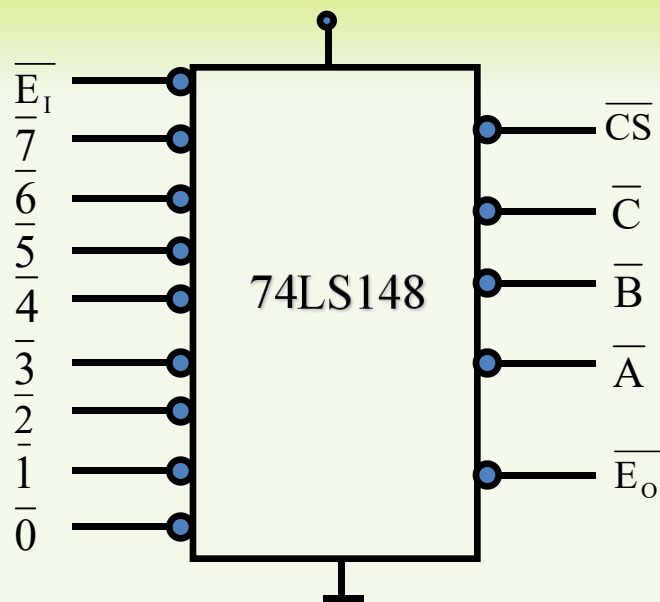


功能表

$\overline{E_1}$	$\overline{7}$	$\overline{6}$	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$	$\overline{0}$	\overline{C}	\overline{B}	\overline{A}	\overline{CS}	$\overline{E_0}$
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	1	0
0	0	X	X	X	X	X	X	X	0	0	0	0	1
0	1	0	X	X	X	X	X	X	0	0	1	0	1
0	1	1	0	X	X	X	X	X	0	1	0	0	1
0	1	1	1	0	X	X	X	X	0	1	1	0	1
0	1	1	1	1	0	X	X	X	1	0	0	0	1
0	1	1	1	1	1	0	X	X	1	0	1	0	1
0	1	1	1	1	1	1	0	X	1	1	0	0	1
0	1	1	1	1	1	1	1	0	1	1	1	0	1

• 优先二进制编码器74LS148

功能表



输入输出端说明:

$\overline{E_I}$: 使能输入端,

$\overline{E_I} = 0$ 时芯片工作

$\overline{E_O}$: 使能输出端,

$\overline{E_O} = 0$ 时为有效, 使下一级编码器有效

\overline{CS} : 优先标志输出端,

$\overline{CS} = 0$ 时表明芯片处于工作状态

$\overline{E_I}$	7	6	5	4	3	2	1	0	\overline{C}	\overline{B}	\overline{A}	\overline{CS}	$\overline{E_O}$
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

9.2.2 译码器



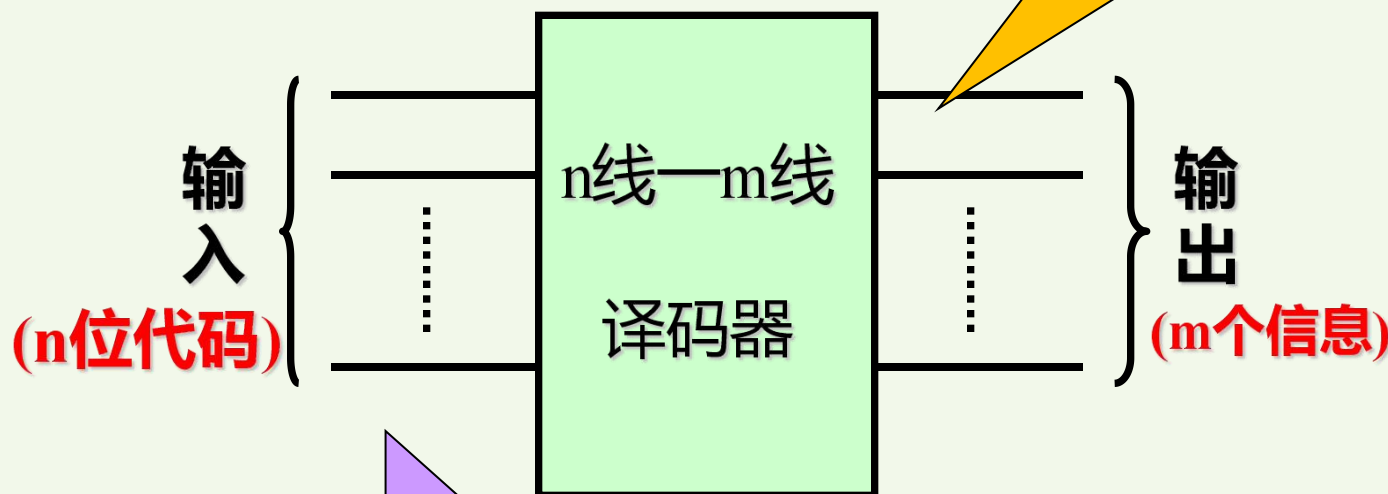
◆ 译码 ---将代码的特定含义翻译出来的过程。

◆ 译码器的功能

--实现译码功能的电路，

一般译码器的输入端数远小于输出端数

◆ 译码器的结构框图



m与n的关系

$$m \leq 2^n$$



◆译码器的分类

二进制译码器

---用来表示输入状态的全部组合
 n 位输入, 2^n 位输出

十进制译码器

---将8421BCD码翻译成10个对应的十进制数码
的电路

显示译码器

---将代码还原成相应数字、文字、符号并
显示出来的电路

译码器

◆ 二进制译码器

以74LS139译码器说明输出

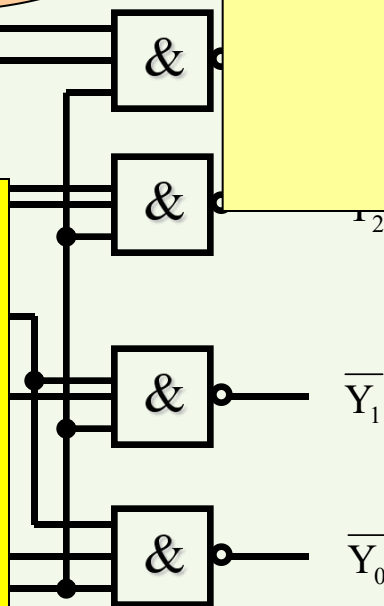
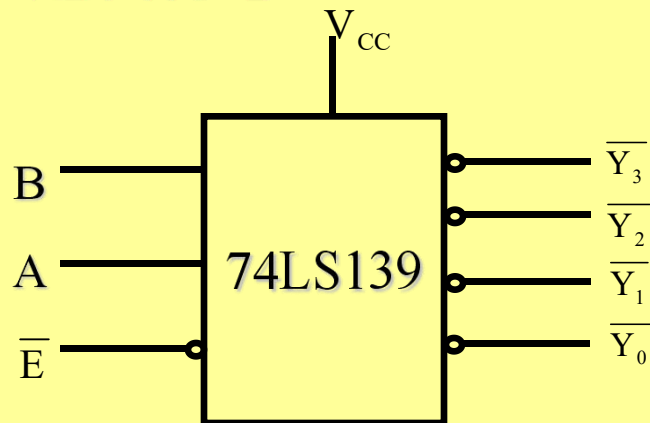
使能端低
电平有效

输出端低
电平有效

真值表

\overline{E}	B	A	\overline{Y}_3	\overline{Y}_2	\overline{Y}_1	\overline{Y}_0
1	×	×	1	1	1	1
0	0	0	1	1	1	0
	0	1	1	1	0	1
	1	0	1	0	1	1
	1	1	0	1	1	1

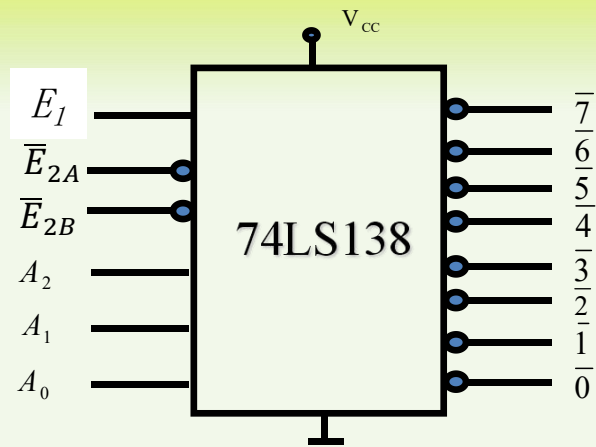
逻辑符号



$\overline{E} = 0$:

$$\left\{ \begin{array}{l} \overline{Y}_3 = \overline{BA} = \overline{m_3} \\ \overline{Y}_2 = \overline{BA} = \overline{m_2} \\ \overline{Y}_1 = \overline{BA} = \overline{m_1} \\ \overline{Y}_0 = \overline{BA} = \overline{m_0} \end{array} \right.$$

◆二进制译码器74LS138 (3线-8线译码器)



当 $E_1 = 1, \bar{E}_{2A} + \bar{E}_{2B} = 0$
时, 各输出表达式为

74LS138功能表

输 入					输 出							
E_1	$\bar{E}_{2A} + \bar{E}_{2B}$	A_2	A_1	A_0	$\bar{7}$	$\bar{6}$	$\bar{5}$	$\bar{4}$	$\bar{3}$	$\bar{2}$	$\bar{1}$	$\bar{0}$
0	×	×	×	×	1	1	1	1	1	1	1	1
×	1	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	0	1	1	1	1	1
1	0	1	1	0	1	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1

$$\bar{Y}_0 = \bar{A}_2 \bar{A}_1 \bar{A}_0$$

$$\bar{Y}_1 = \bar{A}_2 \bar{A}_1 A_0$$

$$\bar{Y}_2 = \bar{A}_2 A_1 \bar{A}_0$$

$$\bar{Y}_3 = \bar{A}_2 A_1 A_0$$

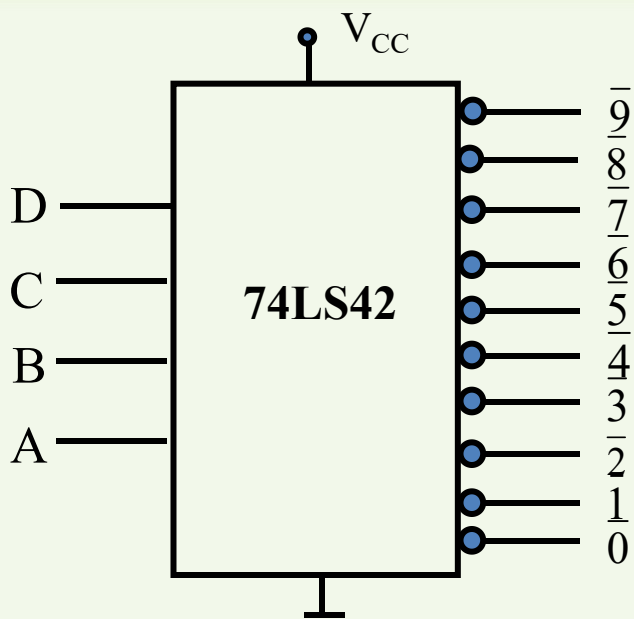
$$\bar{Y}_4 = A_2 \bar{A}_1 \bar{A}_0$$

$$\bar{Y}_5 = A_2 \bar{A}_1 A_0$$

$$\bar{Y}_6 = A_2 A_1 \bar{A}_0$$

$$\bar{Y}_7 = A_2 A_1 A_0$$

◆ BCD/十进制译码器



$$\left\{ \begin{array}{l} \overline{Y}_0 = \overline{\overline{DCBA}}, \overline{Y}_1 = \overline{\overline{DCBA}} \\ \overline{Y}_2 = \overline{\overline{DCBA}}, \overline{Y}_3 = \overline{\overline{DCBA}} \\ \overline{Y}_4 = \overline{\overline{DCBA}}, \overline{Y}_5 = \overline{\overline{DCBA}} \\ \overline{Y}_6 = \overline{\overline{DCBA}}, \overline{Y}_7 = \overline{\overline{DCBA}} \\ \overline{Y}_8 = \overline{\overline{DCBA}}, \overline{Y}_9 = \overline{\overline{DCBA}} \end{array} \right.$$

74LS42功能表

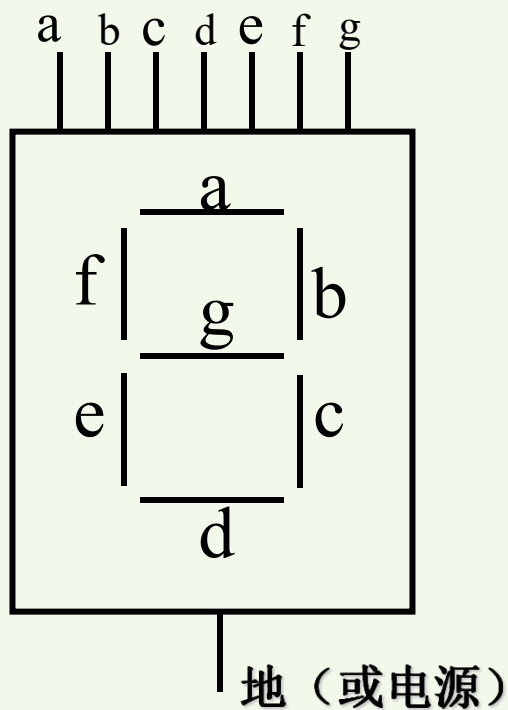
[illegible]

◆ 数字显示译码器

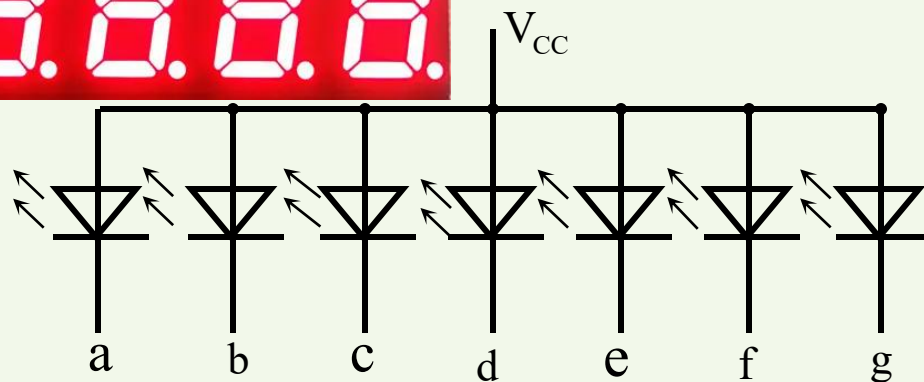
显示译码器可直接用来驱动显示器件，以显示代码所表示的数字、字符等信息。



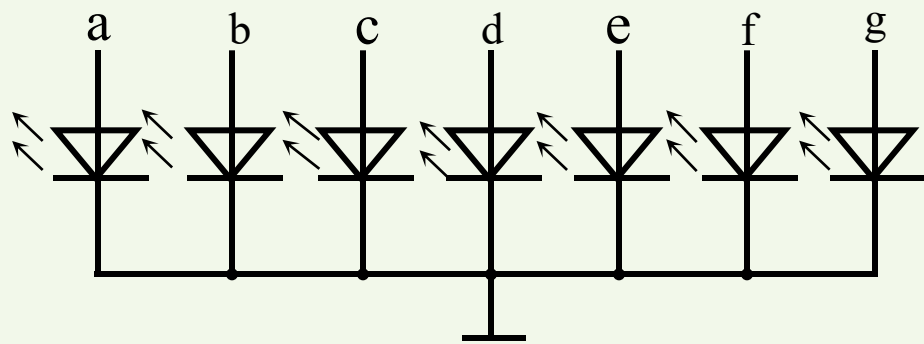
● LED数码管显示器件



数码管符号



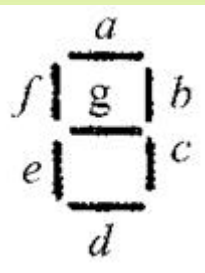
共阳极LED结构



共阴极LED结构

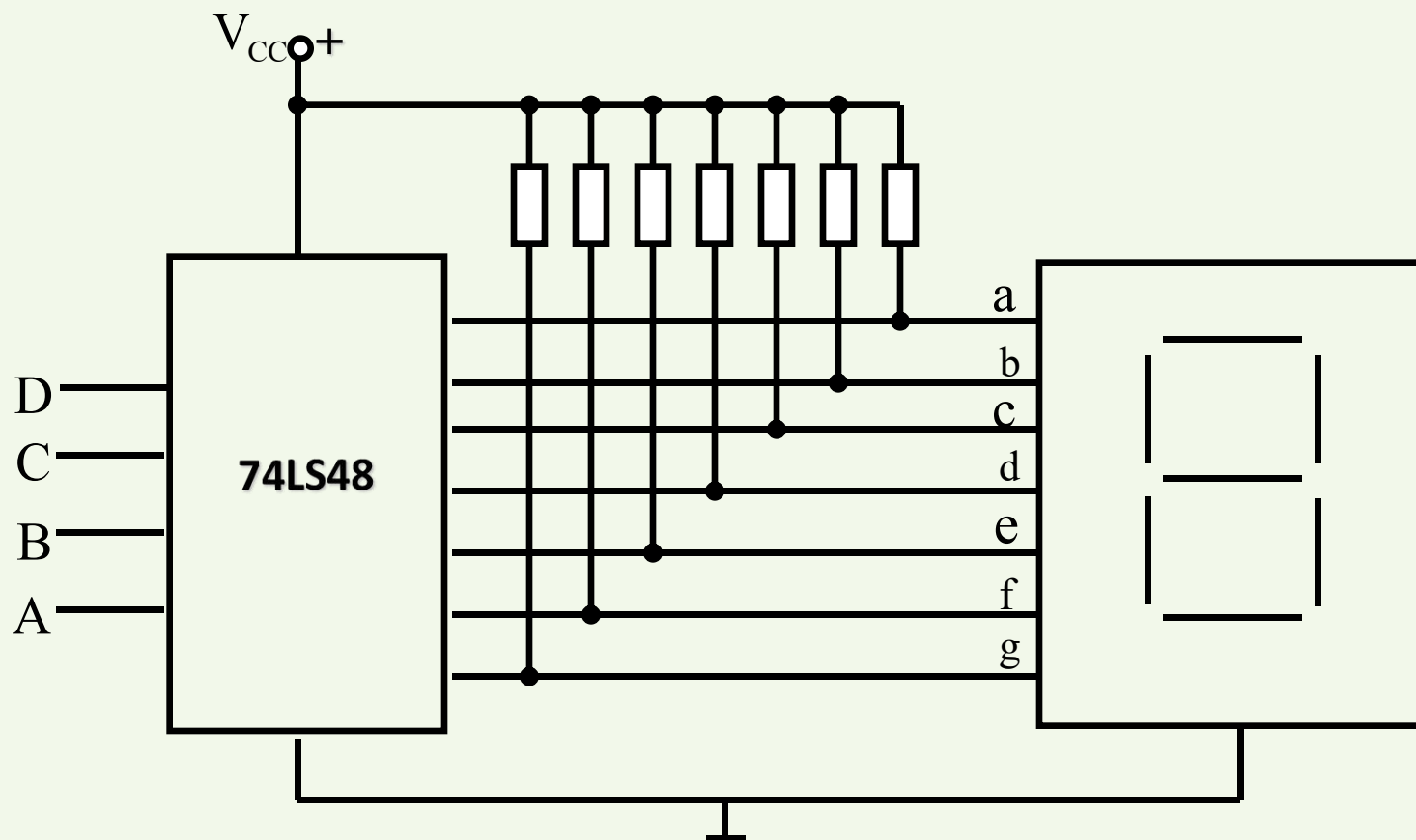


74LS48 (共阴极) 功能表



输 入				输 出							
D	C	B	A	a	b	c	d	e	f	g	显示字符
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	0	1	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	0	1	1	9

• 集成译码显示器74LS48 (共阴极)

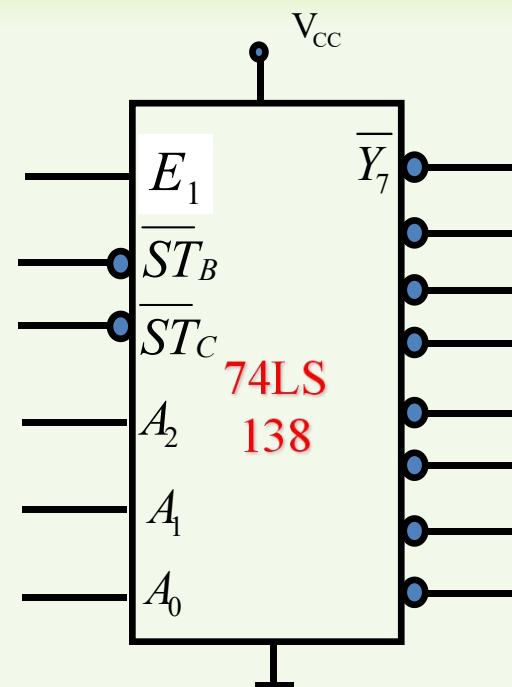


数码管的显示驱动

译码器的应用

◆实现逻辑函数

由于 n 变量的二进制译码器可以提供 2^n 个最小项的输出，而任何逻辑函数均可写为最小项之和的形式，所以利用二进制译码器和一些必要的逻辑门可以实现任意组合逻辑函数。

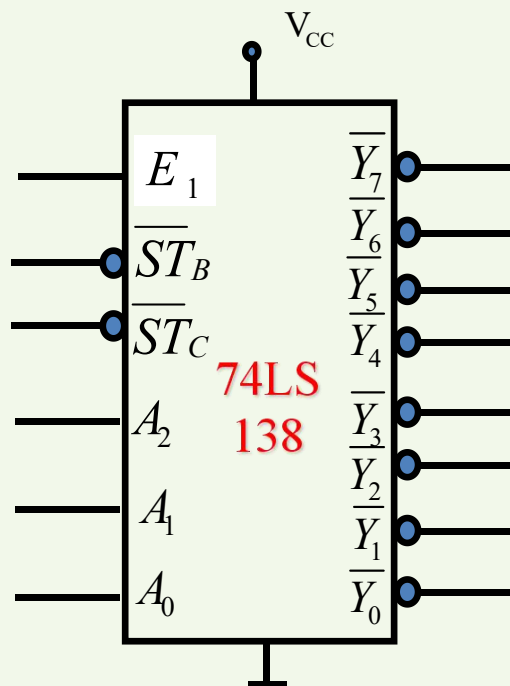


例1：试用74LS138和与非门实现逻辑函数

$$F(C, B, A) = \sum_m (1, 4, 6, 7)$$



解： 已知74LS138逻辑关系为：



当 $E_1 = 1$, $\overline{ST}_B + \overline{ST}_C = 0$ 时,

各输出表达式为：

$$\overline{Y}_0 = \overline{A_2} \overline{A_1} \overline{A_0} = \overline{m_0}$$

$$\overline{Y}_1 = \overline{A_2} \overline{A_1} A_0 = \overline{m_1}$$

$$\overline{Y}_2 = \overline{A_2} A_1 \overline{A_0} = \overline{m_2}$$

$$\overline{Y}_3 = \overline{A_2} A_1 A_0 = \overline{m_3}$$

$$\overline{Y}_4 = A_2 \overline{A_1} \overline{A_0} = \overline{m_4}$$

$$\overline{Y}_5 = A_2 \overline{A_1} A_0 = \overline{m_5}$$

$$\overline{Y}_6 = A_2 A_1 \overline{A_0} = \overline{m_6}$$

$$\overline{Y}_7 = A_2 A_1 A_0 = \overline{m_7}$$



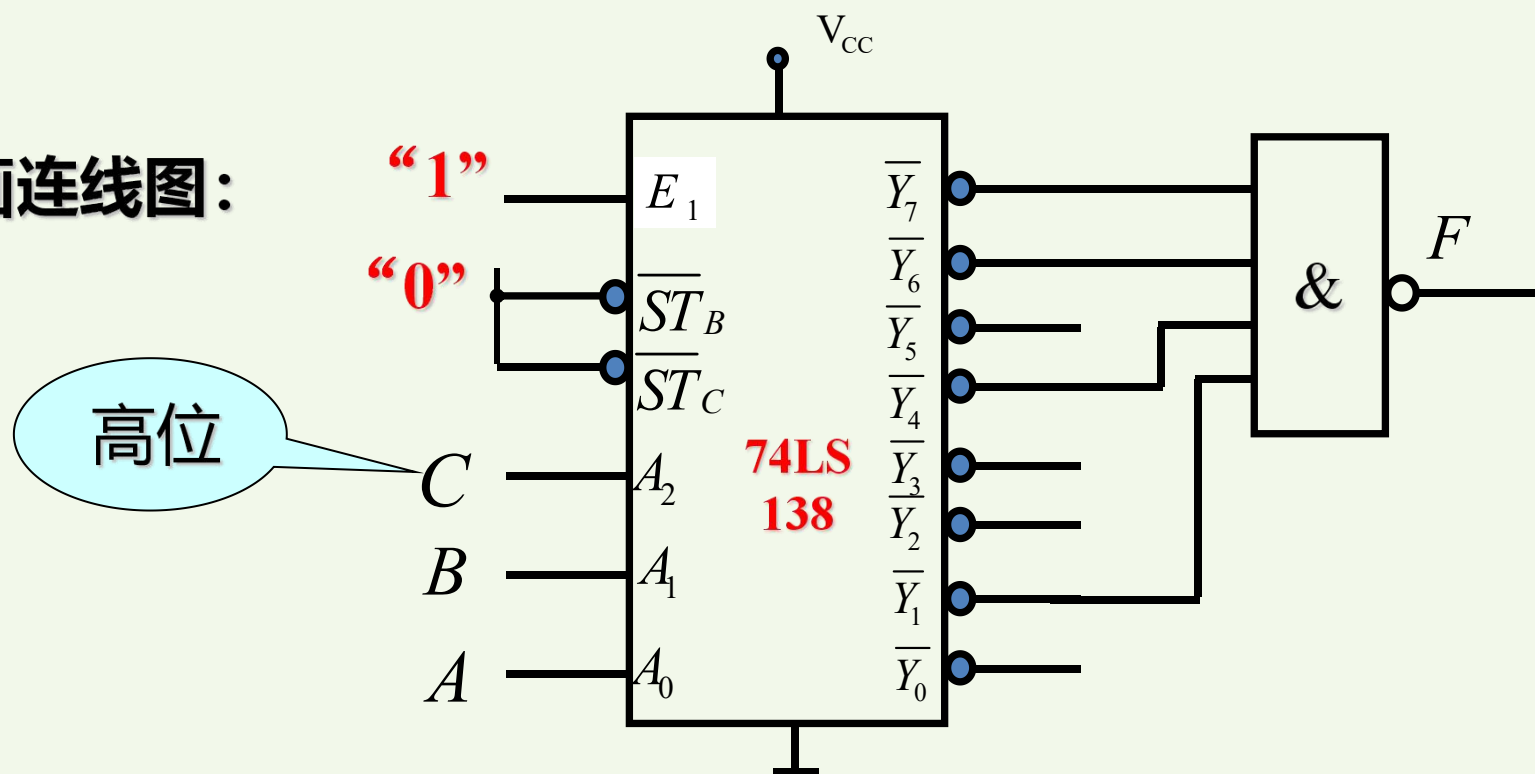
$$F(C, B, A) = \sum_m (1, 4, 6, 7)$$

$$= m_1 + m_4 + m_6 + m_7 = \overline{\overline{m_1 + m_4 + m_6 + m_7}} = \overline{m_1 \cdot m_4 \cdot m_6 \cdot m_7}$$

令 $A_2=C, A_1=B, A_0=A$

则 $F = \overline{\overline{Y}_1 \cdot \overline{Y}_4 \cdot \overline{Y}_6 \cdot \overline{Y}_7}$

画连线图：





◆ 利用译码器设计电路的步骤

❖ 选择集成二进制译码器

译码器输入**二进制代码位数**=**函数变量数**

❖ 写出函数的标准“与或”式

若用**低电平有效**芯片实现→“**与非-与非**”式

❖ 确认译码器和门电路输入信号的表达式

译码器输入：函数变量（注意**排列顺序**）

门电路输入：逻辑函数包括的最小项
所对应的译码器输出

❖ 画连线图

例2:



设X、Z均为三位二进制数，X为输入，Z为输出。当 $2 \leq X \leq 5$ 时， $Z = X + 2$ ；当 $X < 2$ 时， $Z = 1$ ；当 $X > 5$ 时 $Z = 0$ 。试用一片3线 - 8线译码器74LS138构成实现上述要求的逻辑电路。

解：(1) 由题意可得真值表如下：

X_2	X_1	X_0	Z_2	Z_1	Z_0
0	0	0	0	0	1
0	0	1	0	0	1
0	1	0	1	0	0
0	1	1	1	0	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	0	0

(2) 由真值表可得：

$$Z_2(X_2, X_1, X_0) = m_2 + m_3 + m_4 + m_5$$

$$Z_1(X_2, X_1, X_0) = m_4 + m_5$$

$$Z_0(X_2, X_1, X_0) = m_0 + m_1 + m_3 + m_5$$

(3) 用译码器实现

$$\begin{aligned} Z_2 &= m_2 + m_3 + m_4 + m_5 \\ &= \overline{\overline{m_2 + m_3 + m_4 + m_5}} \\ &= \overline{\overline{m_2} \cdot \overline{m_3} \cdot \overline{m_4} \cdot \overline{m_5}} \end{aligned}$$

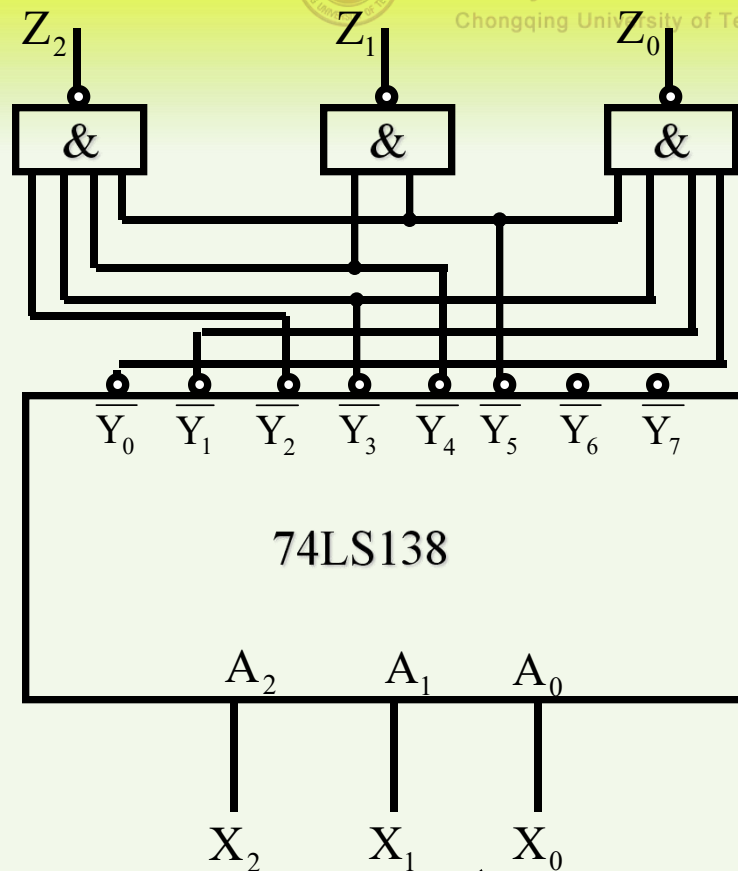
令： $A_2 = X_2, A_1 = X_1, A_0 = X_0$,

则： $Z_2 = \overline{\overline{Y_2} \cdot \overline{Y_3} \cdot \overline{Y_4} \cdot \overline{Y_5}}$

同理可得：

$$Z_1 = \overline{\overline{Y_4} \cdot \overline{Y_5}}$$

$$Z_0 = \overline{\overline{Y_0} \cdot \overline{Y_1} \cdot \overline{Y_3} \cdot \overline{Y_5}}$$



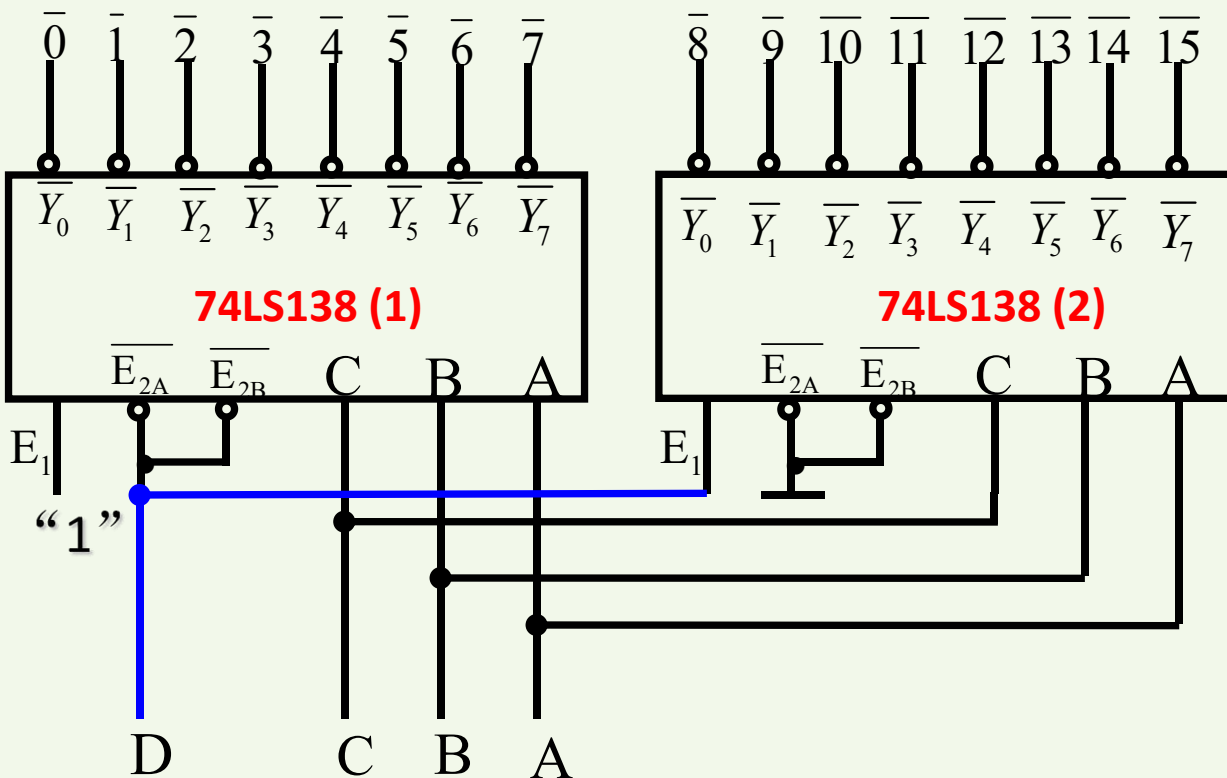
利用译码器设计组合
逻辑电路比较适合于
多输出逻辑函数。



◆扩展应用

例：将两片74LS138扩展为4线-16线译码器。

分析如下：



当**D=0**时，芯片**(1)**工作，对**0000~0111**代码译码， $\overline{0} \sim \overline{7}$ 依次输出“0”。

当**D=1**时，芯片**(2)**工作，对**1000~1111**代码译码， $\overline{8} \sim \overline{15}$ 依次输出“0”。



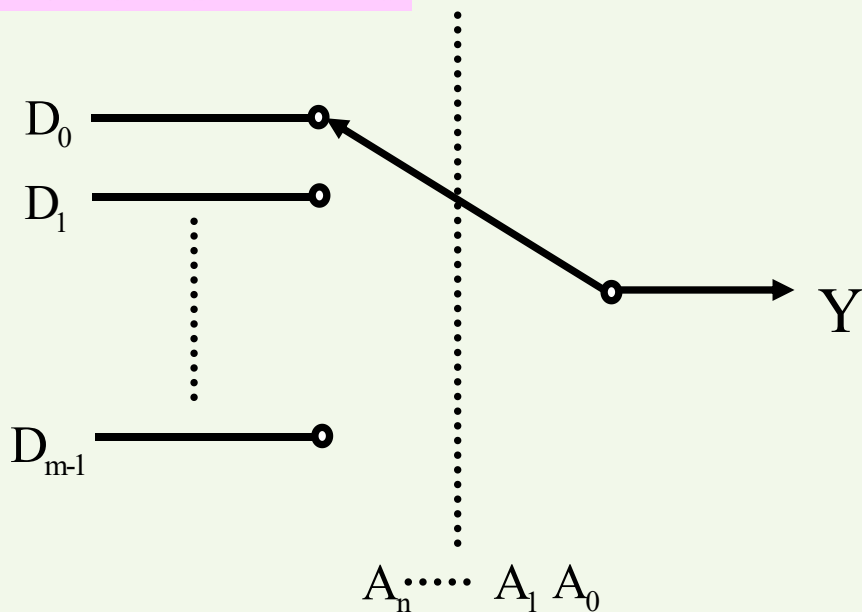
9.2.3 数据选择器

◆数据选择器的功能:

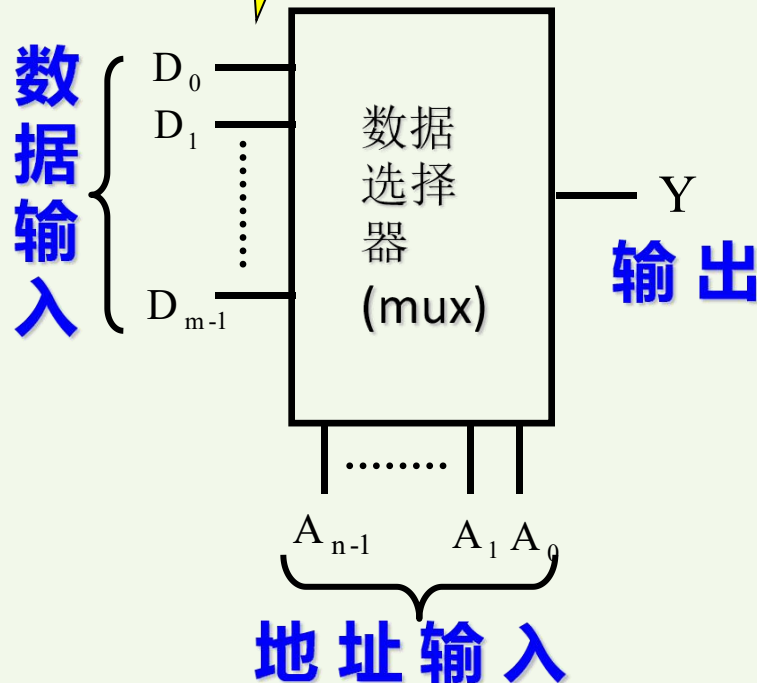
----从多路输入数字信号中, 根据地址码, 选出一路信号送到输出。
又称多路转换器或多路开关。

m 与 n 的关系如何?

功能示意图

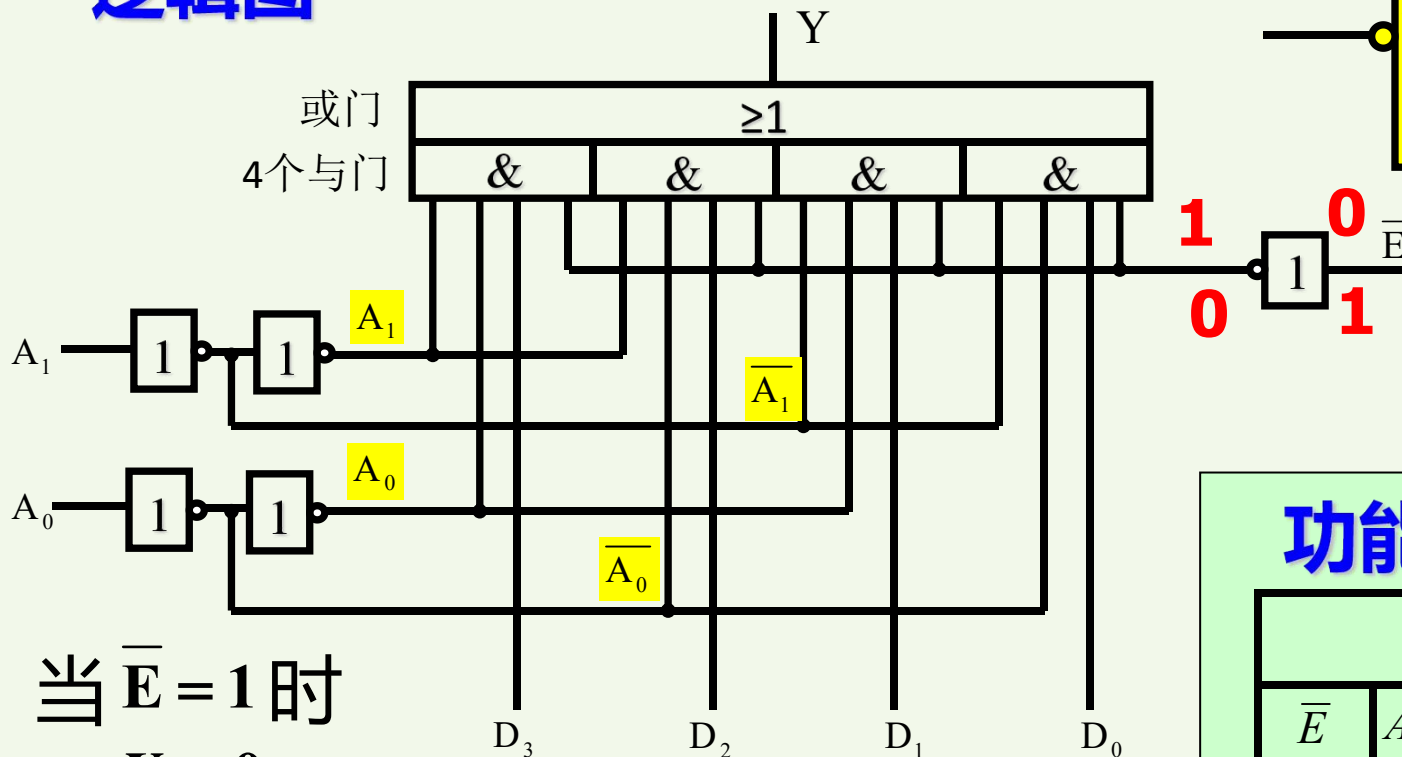


结构示意图



◆4选1数据选择器

逻辑图

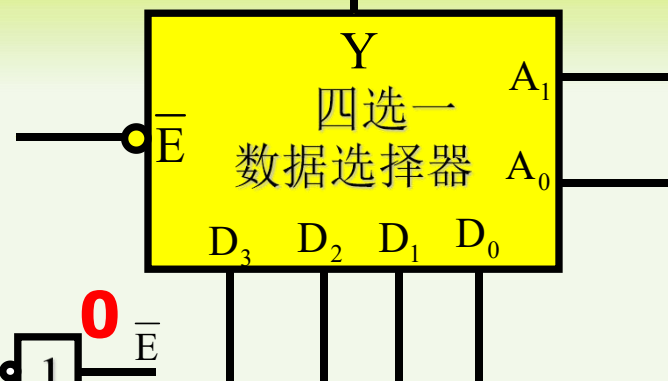


当 $\bar{E} = 1$ 时

$Y = 0$

当 $\bar{E} = 0$ 时

$$Y = A_1 A_0 D_3 + A_1 \bar{A}_0 D_2 + \bar{A}_1 A_0 D_1 + \bar{A}_1 \bar{A}_0 D_0$$
$$= m_3 D_3 + m_2 D_2 + m_1 D_1 + m_0 D_0 = \sum_{i=0}^3 m_i D_i$$



功能表

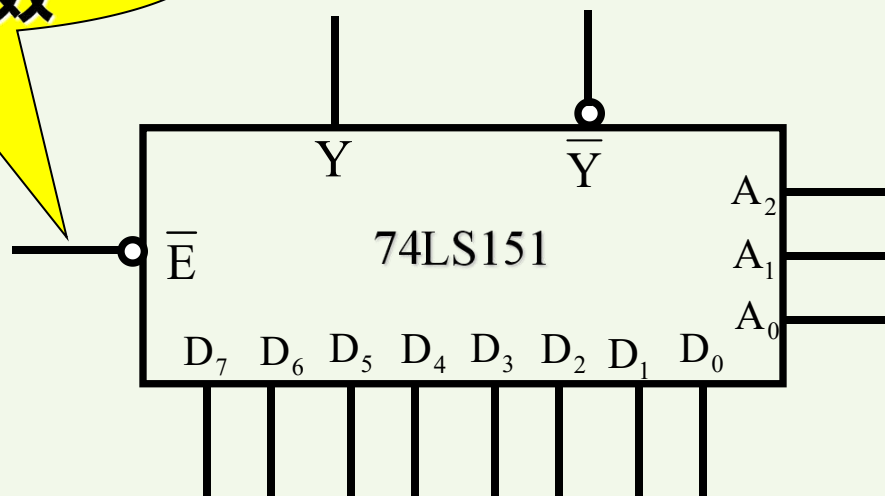
输 入			输出	
\overline{E}	A_1	A_0	$D_3 \sim D_0$	Y
1	×	×	$D_3 \sim D_0$	0
	0	0	$D_3 \sim D_0$	D_0
	0	1	$D_3 \sim D_0$	D_1
	1	0	$D_3 \sim D_0$	D_2
	1	1	$D_3 \sim D_0$	D_3



◆ 8选1数据选择器74LS151

使能控制端，
低电平有效

逻辑符号



逻辑关系

当 $\overline{E} = 0$ 时

$$Y = \sum_{i=0}^7 m_i D_i$$

$$= m_7 D_7 + m_6 D_6 + m_5 D_5 + m_4 D_4 + m_3 D_3 + m_2 D_2 + m_1 D_1 + m_0 D_0$$

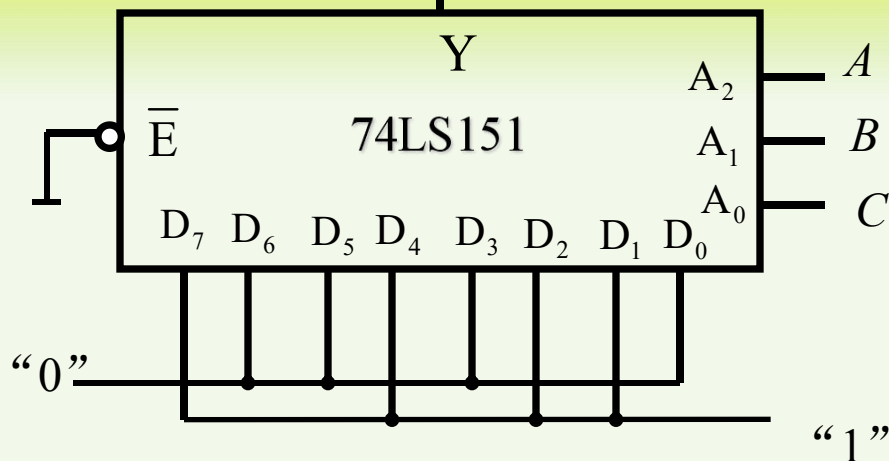
$$= A_2 A_1 A_0 D_7 + A_2 A_1 \overline{A_0} D_6 + A_2 \overline{A_1} A_0 D_5 + A_2 \overline{A_1} \overline{A_0} D_4$$

$$+ \overline{A_2} A_1 A_0 D_3 + \overline{A_2} A_1 \overline{A_0} D_2 + \overline{A_2} \overline{A_1} A_0 D_1 + \overline{A_2} \overline{A_1} \overline{A_0} D_0$$



例：在如图所示电路中，
74LS151为8选1数据选择器，
试分析电路的功能。

解：由8选1数据选择器的逻辑
功能得：



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

由电路结构，得：

$$\begin{cases} A_2 = A & A_1 = B & A_0 = C \\ D_0 = D_3 = D_5 = D_6 = 0 \\ D_1 = D_2 = D_4 = D_7 = 1 \end{cases}$$

判断输入有
无奇数个1

$$F = ABC \cdot 1 + ABC \cdot 0 + A\bar{B}C \cdot 0 + A\bar{B}\bar{C} \cdot 1 + A\bar{B}C \cdot 0 + A\bar{B}\bar{C} \cdot 1 + A\bar{B}C \cdot 1 + A\bar{B}\bar{C} \cdot 0 \\ = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC = A \oplus B \oplus C$$

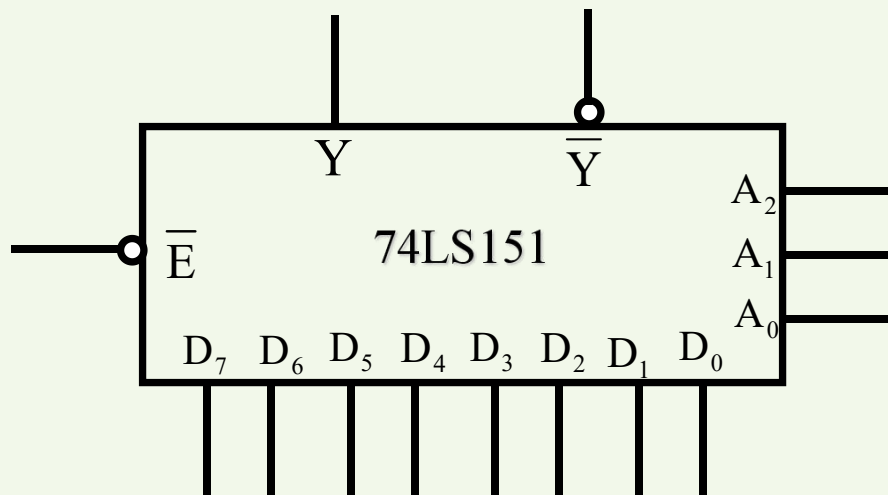


◆ 利用数据选择器实现逻辑函数

- 提供了地址变量的全部最小项
- 具有标准“与或”表达式的形式

例1：用八选一数据选择器74LS151实现函数

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





解： 由8选1数据选择器的逻辑功能得：

$$Y = \sum_{i=0}^7 m_i D_i = \bar{A}_2 \bar{A}_1 \bar{A}_0 D_0 + \bar{A}_2 \bar{A}_1 A_0 D_1 + \bar{A}_2 A_1 \bar{A}_0 D_2 + \bar{A}_2 A_1 A_0 D_3 \\ + A_2 \bar{A}_1 \bar{A}_0 D_4 + A_2 \bar{A}_1 A_0 D_5 + A_2 A_1 \bar{A}_0 D_6 + A_2 A_1 A_0 D_7$$

$$F(A, B, C) = \bar{A}\bar{B} + AC = \bar{A}\bar{B}(C + \bar{C}) + A(B + \bar{B})C \\ = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}C + ABC \\ = \bar{A}\bar{B}\bar{C} \bullet 0 + \bar{A}\bar{B}\bar{C} \bullet 0 + \bar{A}\bar{B}\bar{C} \bullet 0 + \bar{A}\bar{B}\bar{C} \bullet 0 \\ + \bar{A}\bar{B}\bar{C} \bullet 1 + \bar{A}\bar{B}\bar{C} \bullet 1 + A\bar{B}\bar{C} \bullet 0 + ABC \bullet 1$$

令 $A_2 = A, A_1 = B, A_0 = C$

比较两式可得：

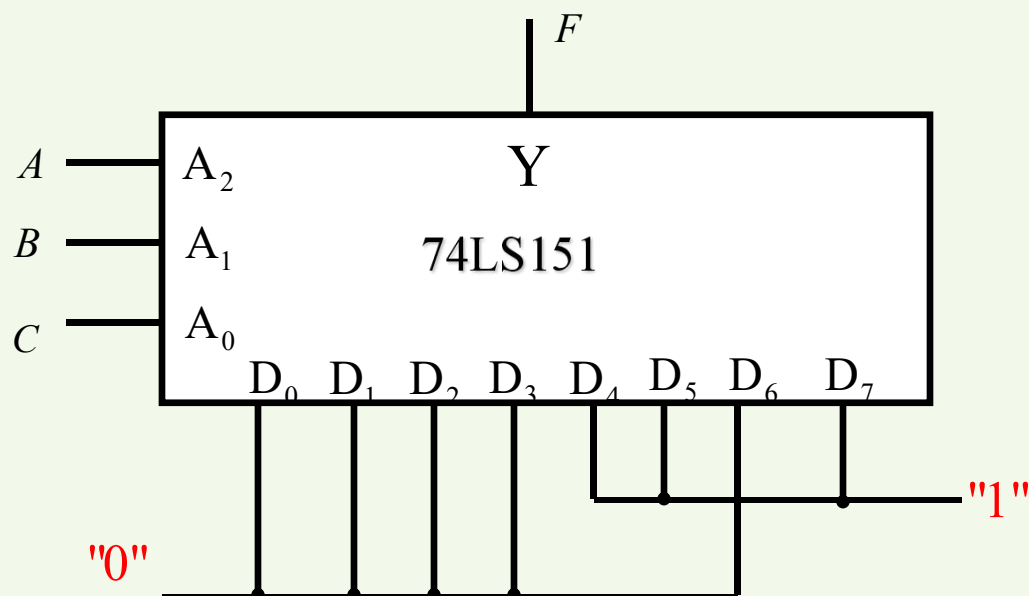
$$D_0 = 0 \quad \text{同理可得：} \quad D_1 = D_2 = D_3 = D_6 = 0$$

$$D_4 = D_5 = D_7 = 1$$



令 $\left\{ \begin{array}{l} A_2 = A, A_1 = B, A_0 = C \\ D_0 = D_1 = D_2 = D_3 = D_6 = 0 \quad D_4 = D_5 = D_7 = 1 \end{array} \right.$

则 $Y = F$





例2：用八选一数据选择器实现函数

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

解1： 八选一数据选择器输出为：

$$Y = \overline{A}_2\overline{A}_1\overline{A}_0D_0 + \overline{A}_2\overline{A}_1A_0D_1 + \overline{A}_2A_1\overline{A}_0D_2 + \overline{A}_2A_1A_0D_3 \\ + A_2\overline{A}_1\overline{A}_0D_4 + A_2\overline{A}_1A_0D_5 + A_2A_1\overline{A}_0D_6 + A_2A_1A_0D_7$$

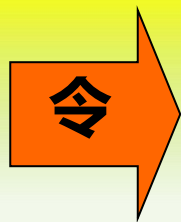
令 $A_2 = A, A_1 = B, A_0 = C$

则 $F = \underline{\overline{A}\overline{B}\overline{C}D} + \underline{\overline{A}\overline{B}CD} + \underline{\overline{A}B\overline{C}D} + \underline{A\overline{B}\overline{C}\overline{D}} + \underline{AB\overline{C}\overline{D}} + \underline{ABC}$

$$= m_0 \cdot D + m_1 \cdot D + m_2 \cdot D + m_4 \cdot \overline{D} + m_6 \cdot \overline{D} + m_7 \cdot 1 \quad + m_3 \cdot 0 + m_5 \cdot 0$$

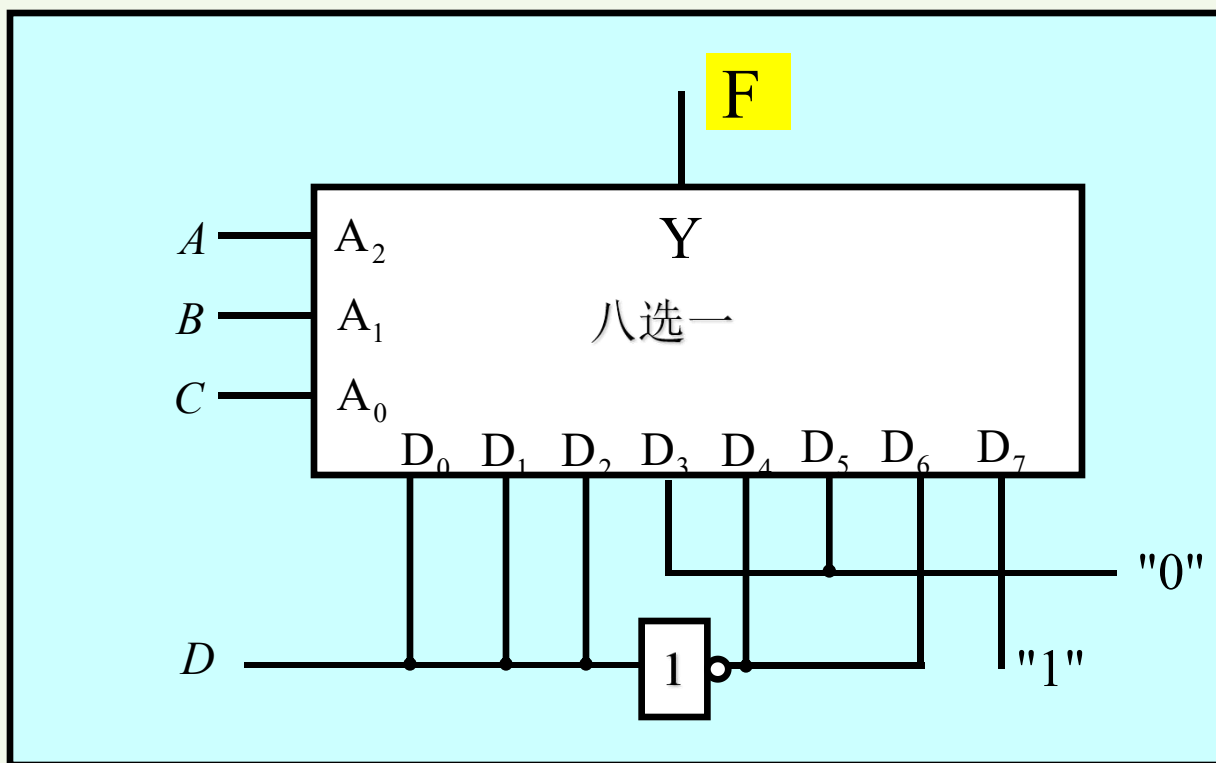
得：

$$\begin{cases} D_0 = D_1 = D_2 = D & D_4 = D_6 = \overline{D} \\ D_7 = 1 & D_3 = D_5 = 0 \end{cases}$$



$$\left\{ \begin{array}{l} A_2 = A, A_1 = B, A_0 = C \\ D_0 = D_1 = D_2 = D \quad D_4 = D_6 = \bar{D} \quad D_7 = 1 \quad D_3 = D_5 = 0 \end{array} \right.$$

则 $Y = F$





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

解2: 画出函数的卡诺图

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

若令 $A_2=C$, $A_1=B$,
 $A_0=A$, 则函数实现
有何不同?

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

令

$$A_2 = A \quad A_1 = B \quad A_0 = C$$

则

$$F = m_0 D + m_1 D + m_2 D + m_3 \cdot 0 + m_4 \overline{D} + m_5 \cdot 0 + m_6 \overline{D} + m_7$$

得

$$D_0 = D_1 = D_2 = D \quad D_4 = D_6 = \overline{D} \quad D_7 = 1 \quad D_3 = D_5 = 0$$



➤ 利用数选设计电路的步骤

- ❖ 确定所用数据选择器的输出表达式

$$Y = \sum m_i D_i$$

- ❖ 写出函数的标准“与或”式

- ❖ 将所设计电路的表达式与数选输出表达式比较

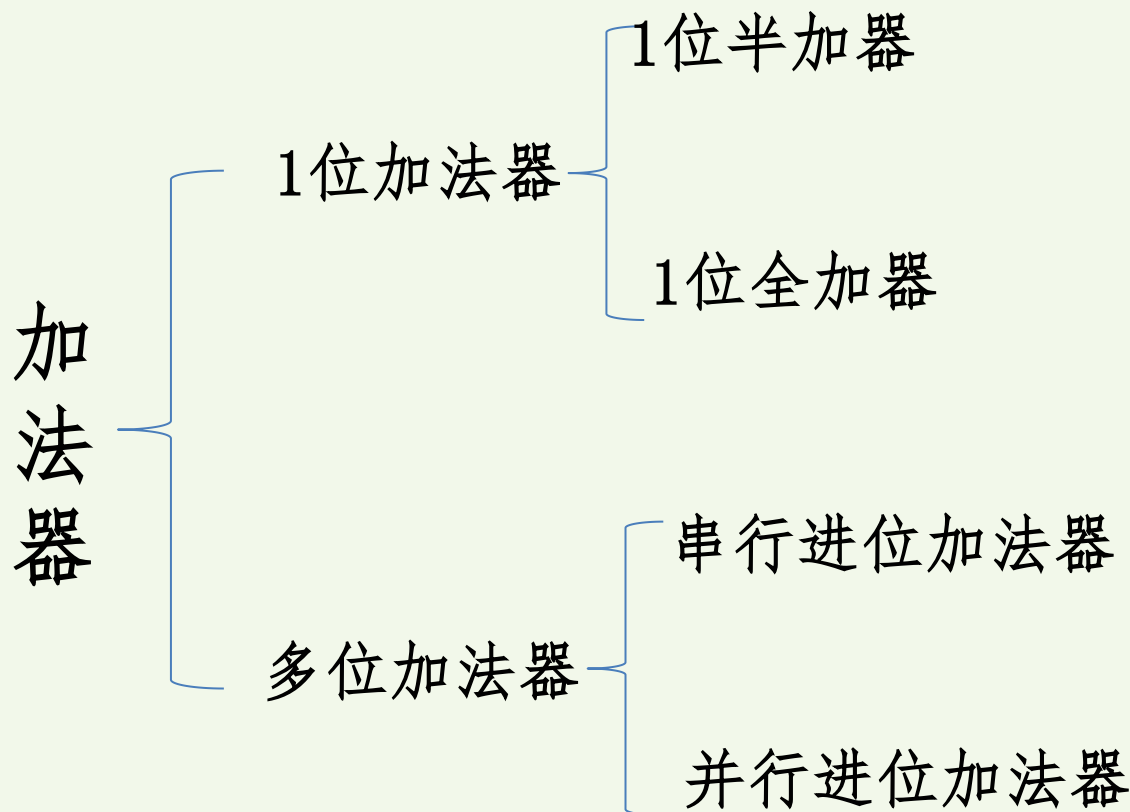
{ 若数选的地址数=输入变量数，则 $D=0$ 或 $D=1$
若数选的地址数<输入变量数，
则利用代数法或卡诺图降维法确定 D

- ❖ 画连线图



9.2.4 加法器

两个二进制数之间的算术运算无论是加、减、乘、除，在数字计算机中都是化为若干步**加法**运算和**移位**进行的。所以加法器是算术运算器的基本单元。

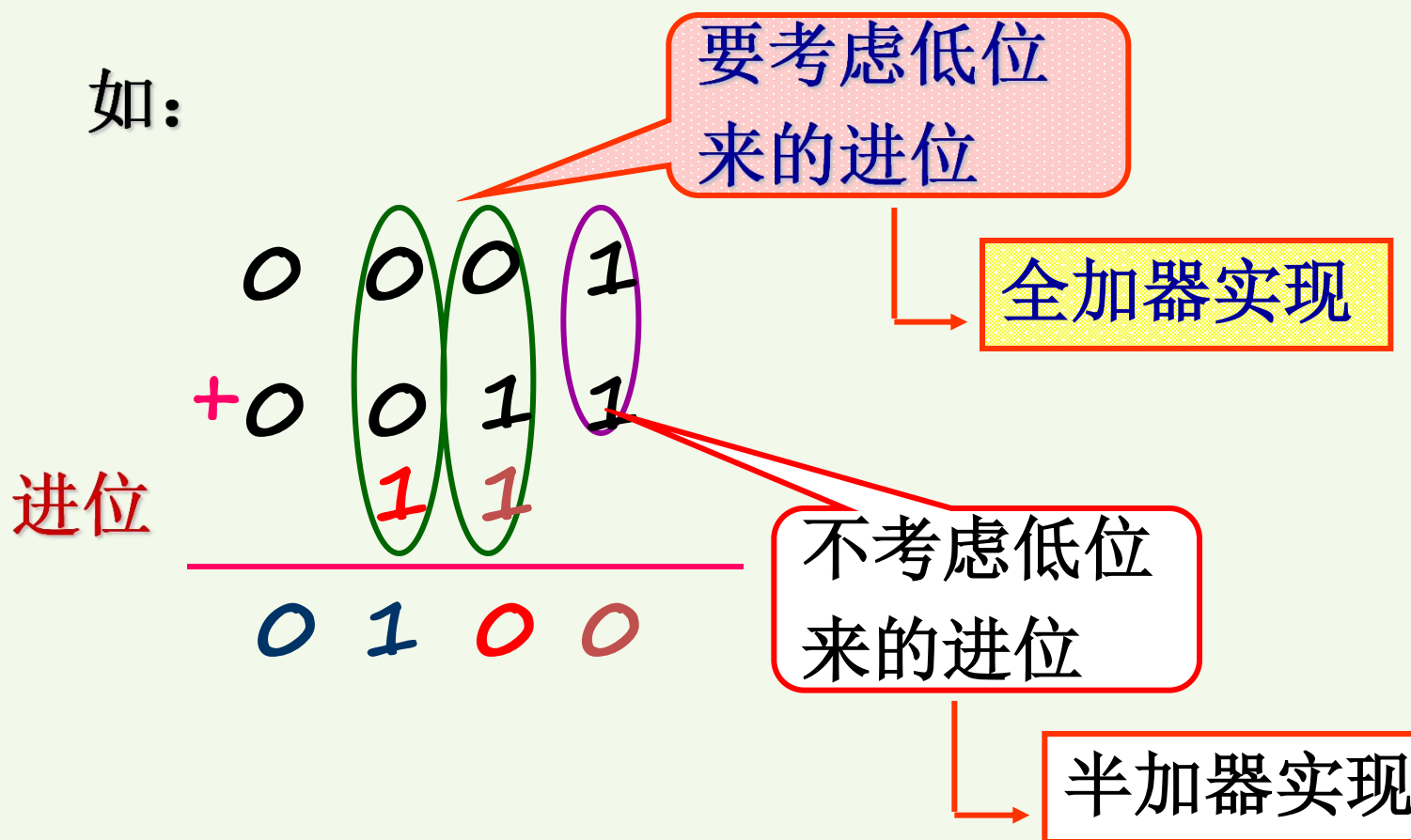




◆ 加法器的功能

--实现两个二进制数之间的相加运算。

如：





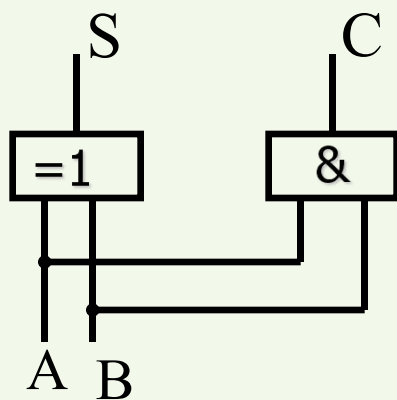
◆一位半加器 ---不考虑低位进位的一位加法器



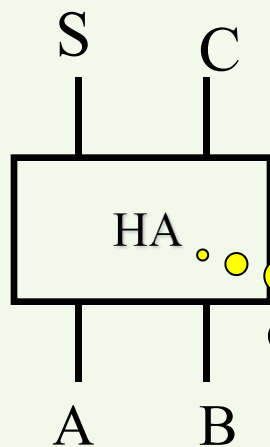
真值表

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

逻辑图



符号



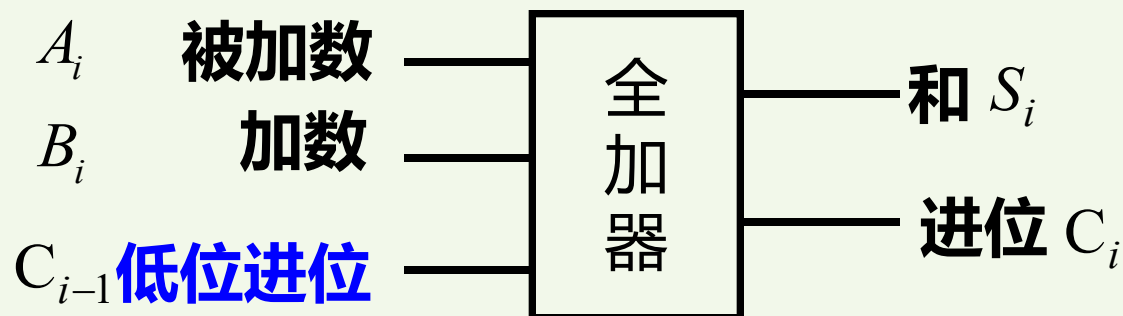
表达式

$$\begin{cases} S = \bar{A}B + A\bar{B} = A \oplus B \\ C = AB \end{cases}$$

Half Adder



◆一位全加器： ---考虑低位进位的一位加法器



真值表:

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

表达式:

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

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

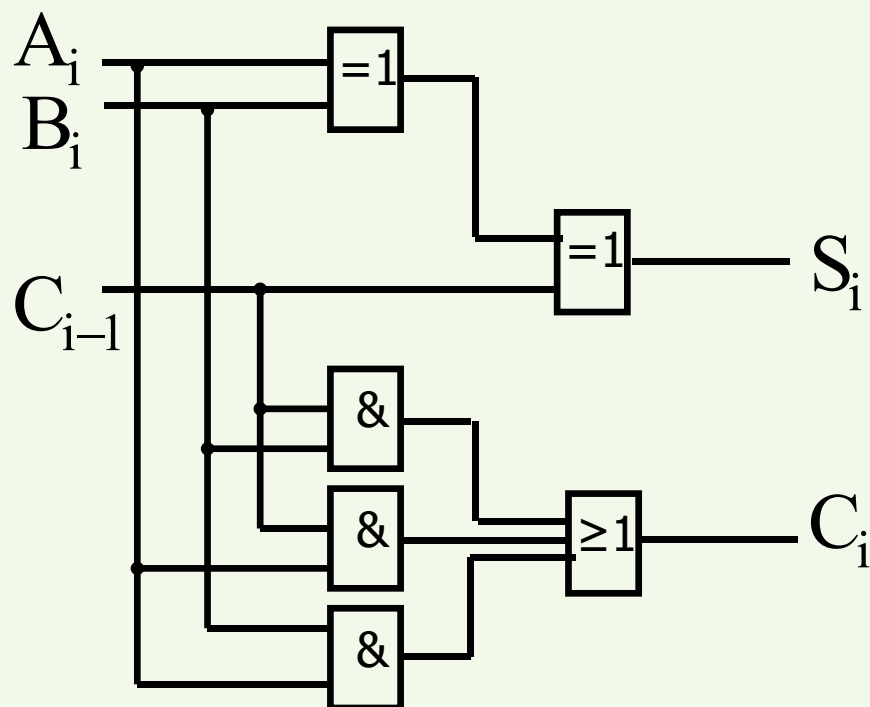
$$\begin{aligned} C_i &= \overline{A_i}B_iC_{i-1} + A_i\overline{B_i}C_{i-1} + A_iB_i\overline{C_{i-1}} + A_iB_iC_{i-1} \\ &= A_iB_i + B_iC_{i-1} + A_iC_{i-1} \end{aligned}$$

表达式:

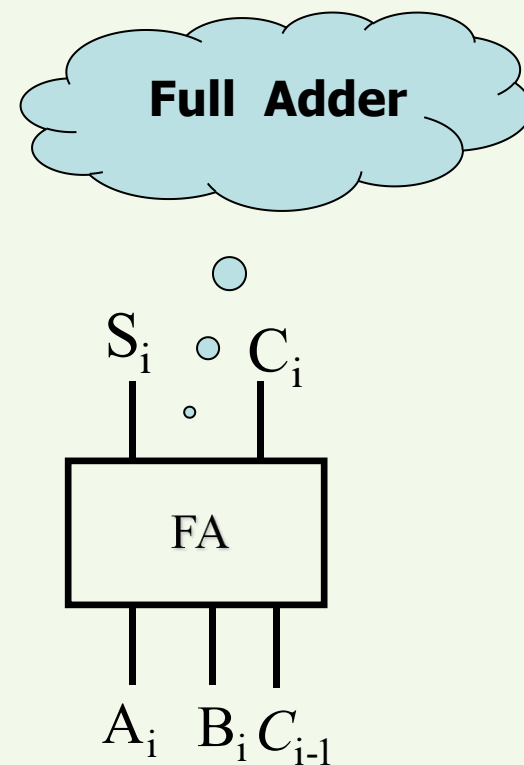
$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

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

逻辑图



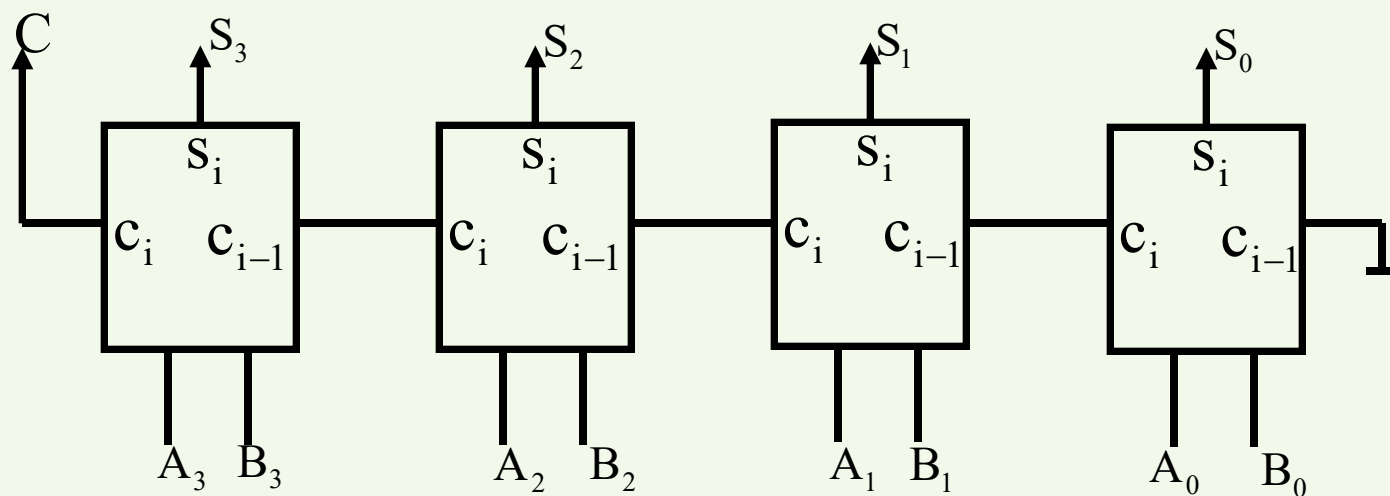
符号





◆多位加法器 { 串行进位加法器 并行进位加法器

例：四位串行进位加法器



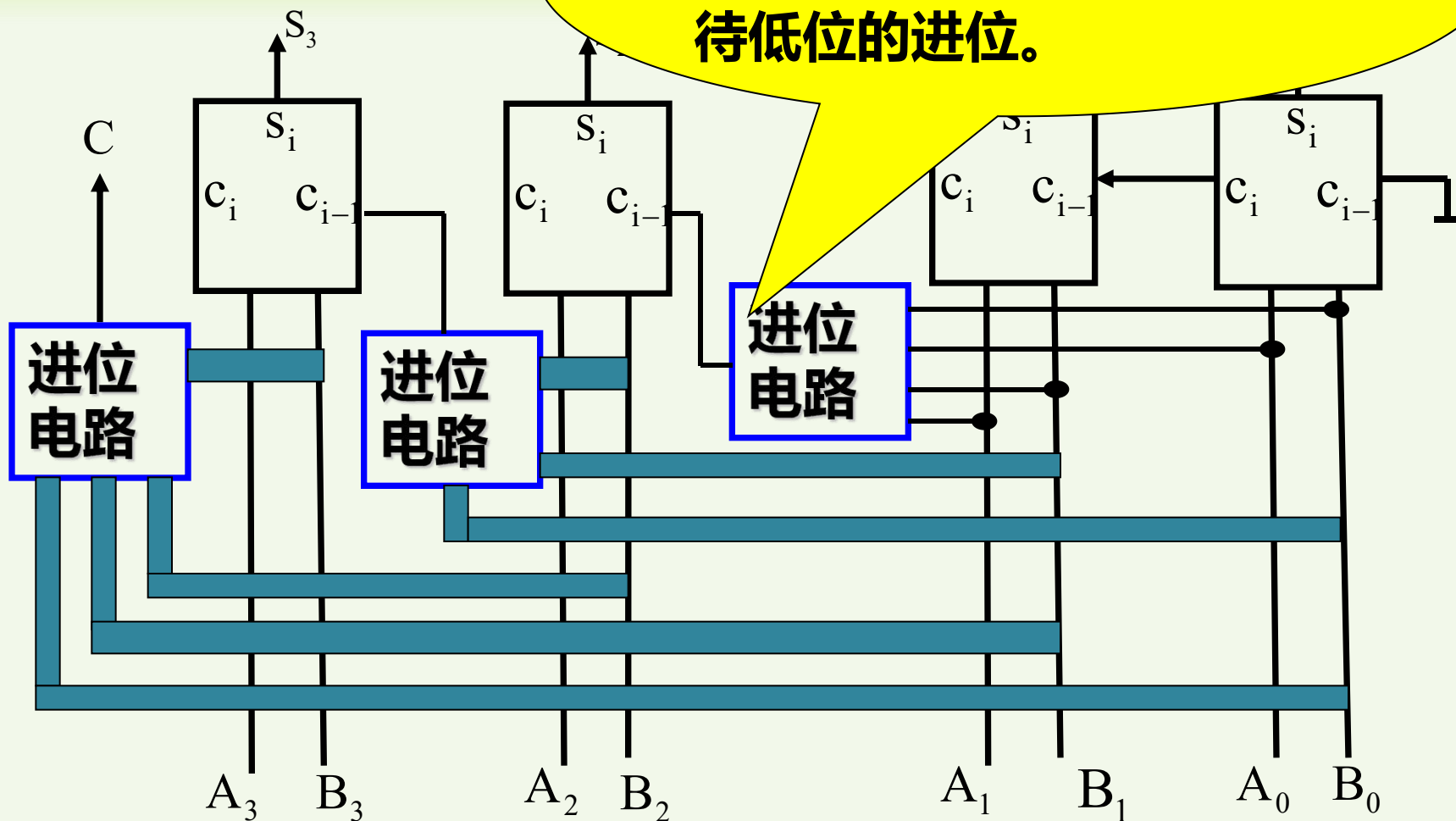
特点

结构简单

各位全加器间的进位需串行传递，速度较慢。

例：四位并行进位加法

直接由输入数据产生各位所需进位信号，不需要等待低位的进位。



特点 { 运算速度快
电路结构复杂



课程小结

1、组合逻辑电路的分析与设计

- 组合逻辑电路的特点
- 组合逻辑电路的分析
- 组合逻辑电路的设计

➤ 2、常用集成组合逻辑电路

- 编码器
- 译码器
- 数据选择器
- 加法器



课堂练习

第二节1, 2, 4, 5

