

第四章 组合逻辑电路

4.1 组合逻辑电路分析

4.2 组合逻辑电路设计

4.4 译码器

4.5 多路选择器(数据选择器)

4.6 数值比较器

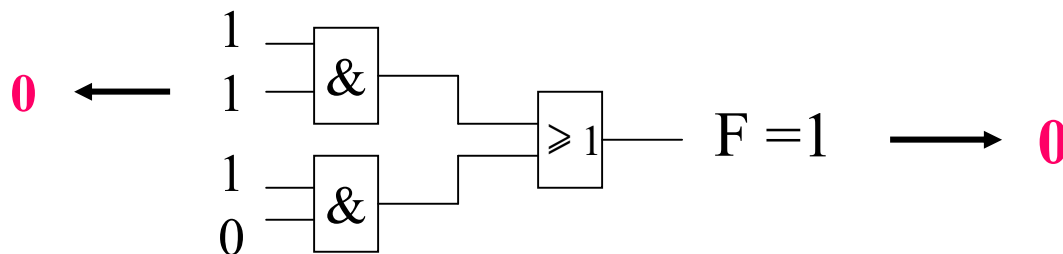
4.7 加法器

第四章 组合逻辑电路

➤ 逻辑电路 { 组合电路
时序电路

➤ 组合逻辑电路:

- 由门电路组成
- 不存在反馈电路 (无记忆功能)
- 任何时刻的输出只与该时刻的输入有关



§ 4.1 组合逻辑电路分析

➤分析:

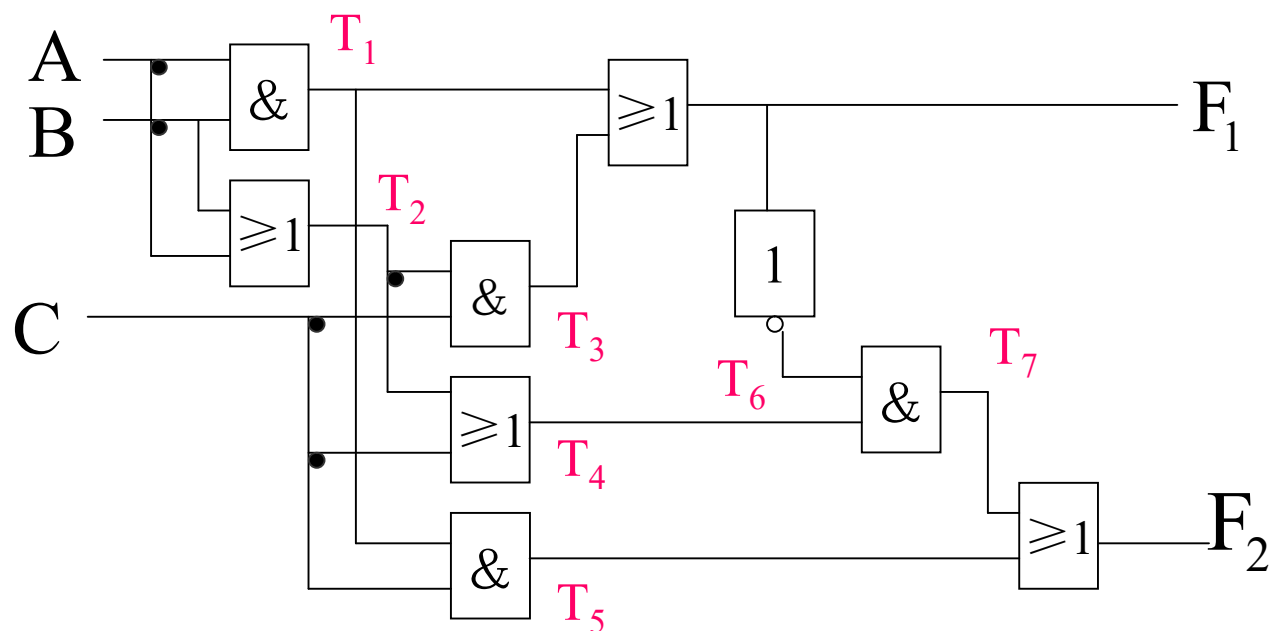
给定电路, 确定输出 F , 并讨论其功能

(已知电路求功能)

➤步骤:

- ① 根据输入逐级写出输出内容
- ② 化简逻辑功能
- ③ 列出真值表
- ④ 讨论功能

例：分析下列电路的逻辑功能



方法：

1. 写出每个门的输出表达式
2. 写出 F 并化简

$$T_1 = AB, \quad T_2 = A + B, \quad T_3 = (A + B)C,$$

$$T_4 = A + B + C, \quad T_5 = ABC,$$

$$F_1 = T_1 + T_3 = AB + (A + B)C = AB + AC + BC,$$

$$T_6 = \overline{F_1}$$

$$\begin{aligned} T_7 &= T_6 \cdot T_4 = \overline{AB + AC + BC}(A + B + C) \\ &= \overline{A} \overline{B} C + \overline{A} B \overline{C} + A \overline{B} \overline{C} \end{aligned}$$

$$F_2 = T_7 + T_5 = \overline{A} \overline{B} C + \overline{A} B \overline{C} + A \overline{B} \overline{C} + ABC$$

3. 列真值表

$$F_1 = AB + BC + AC$$
$$= \sum (3, 5, 6, 7)$$

$$F_2 = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

4. 分析

$$F_1 = AB + BC + AC$$

$$F_2 = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC)$$
$$= \bar{A}(B \oplus C) + A(\overline{B \oplus C})$$
$$= A \oplus B \oplus C$$

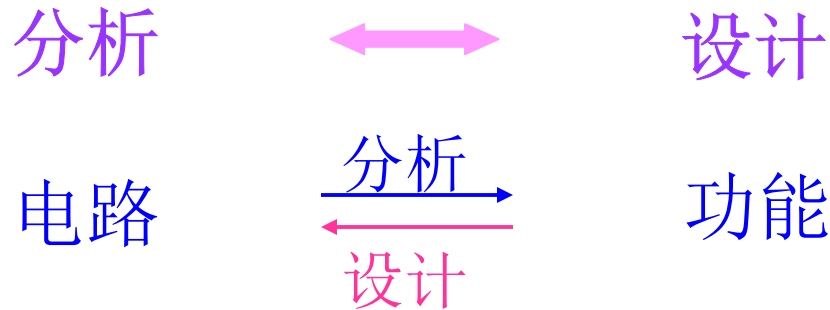
真值表

A	B	C	F_1	F_2
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

三变量表决电路

奇偶校验

§ 4.2 组合逻辑电路设计



主要步骤:

- 确定输入、输出以及它们之间的关系
- 列出真值表
- 化简
- 画出逻辑电路图

例 1: 设计三变量表决电路

三人选组长，同意为**1**，不同意为**0**；两票以上同意为当选（为**1**），未当选为**0**。

三人：A, B, C $\left\{ \begin{array}{ll} \text{1} & \text{同意} \\ \text{0} & \text{不同意} \end{array} \right.$

当选：F $\left\{ \begin{array}{ll} \text{1} & \text{当选} \\ \text{0} & \text{未当选} \end{array} \right.$

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

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

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

$$F = AB + AC + BC$$

(电路)

关键：分析题意

例 2

三人裁判举重比赛，一个主裁判，两个副裁判，认为杠铃举上时，按自己前面的电键（为1）否则不按（为0）；裁判结果用红，绿灯表示，灯亮为1，红绿灯都亮表示“完全举起”，只亮红灯为“需研究录像决定”，其余为未举起。

1. 三个裁判均按下电键，红绿灯全亮；
2. 一主裁一副裁按下电键，红绿灯全亮；
3. 两副裁或一主裁按电键，只红灯亮；
4. 其他情况红绿灯全灭。

试用与非门设计满足上述要求的控制电路

输入

A 主裁判 { 1 按下按钮
B } 副裁判 { 0 未按按钮
C }

输出

R, G { 1 亮灯
0 灭灯

真值表

A	B	C	R	G
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

化简

R

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

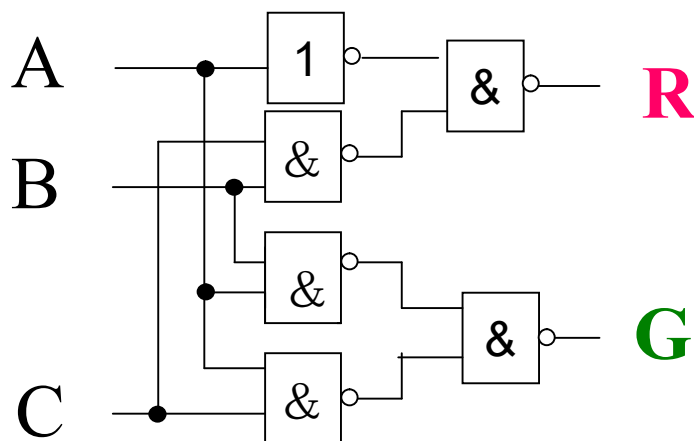
$$\mathbf{R} = \mathbf{A} + \mathbf{BC} = \overline{\overline{A}} \cdot \overline{\overline{BC}}$$

G

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

$$\mathbf{G} = \mathbf{AB} + \mathbf{AC} = \overline{\overline{AB}} \cdot \overline{\overline{AC}}$$

电路



§ 4.4 译码器

译码器是一个将N位二进制输入代码转换为M-线输出的逻辑电路，因此每条输入线只对应一组输入组合输出有效 ($M \leq 2^N$)。

N位二进制输入 $\xrightarrow{\text{转换}}$ $\leq 2^N$ 输出

分类

译码器

二进制译码器

码制变换译码器

显示译码器

4.1.1 二进制译码器

输入：二进制码, n 位

输出：输入的组合, 2^n

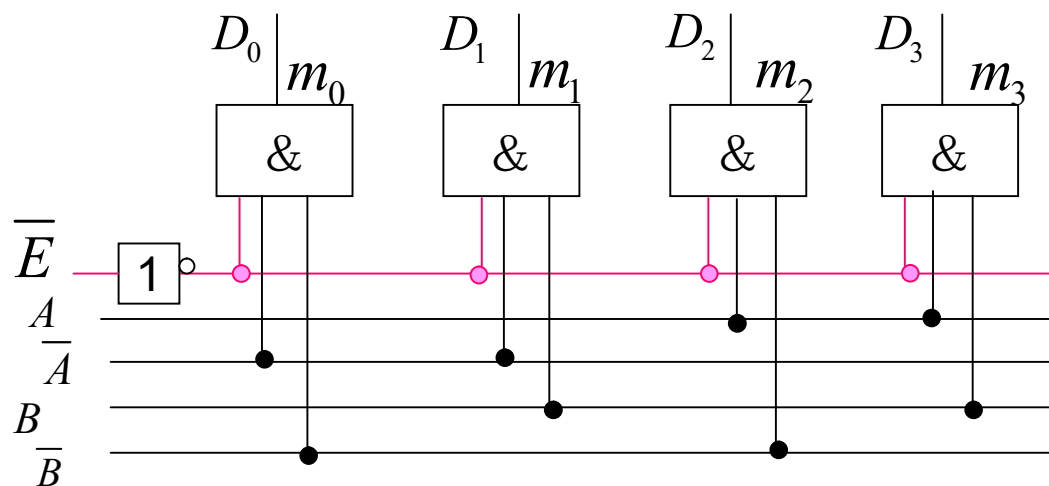
1. 2线 - 4线译码器

1). 高电平有效 2-4 译码器

输入		输出			
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

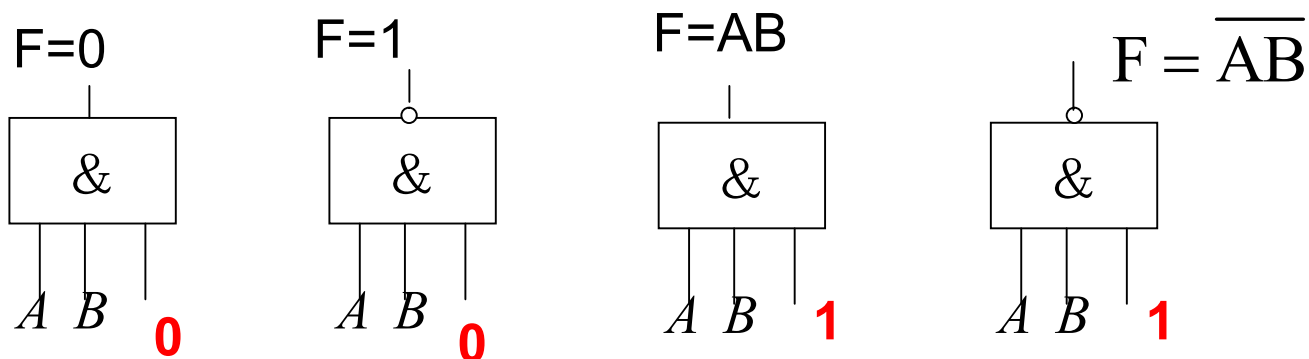
输入数码是二进制数几，第几号输出就是唯一的高电平，其余输出皆为低电平。

电路

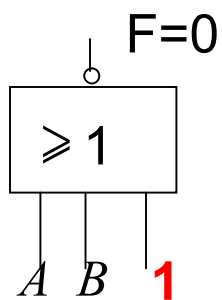
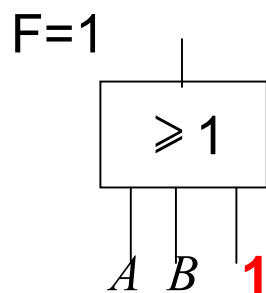


\overline{E} : 启用 $\left\{ \begin{array}{l} \overline{E}=0, \text{译码器工作} \\ \overline{E}=1, \text{译码器锁定} \end{array} \right.$

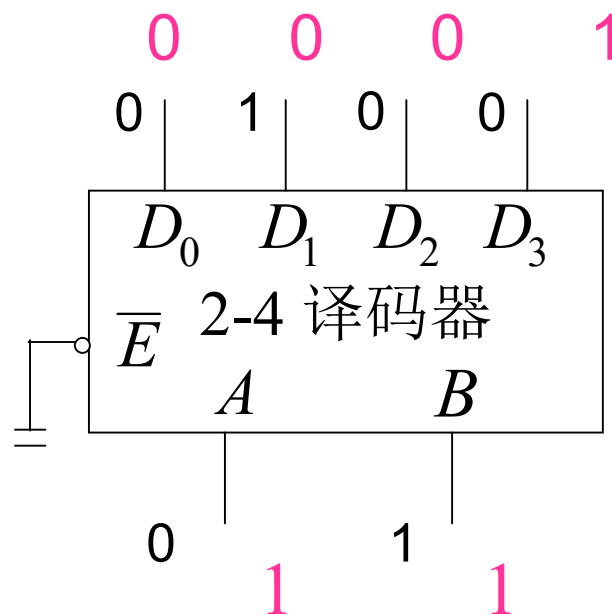
如果输入0到与门或与非门，门电路锁定



如果是输入1到或门或者或非门，门电路锁定



2-4 高电平有效译码器符号



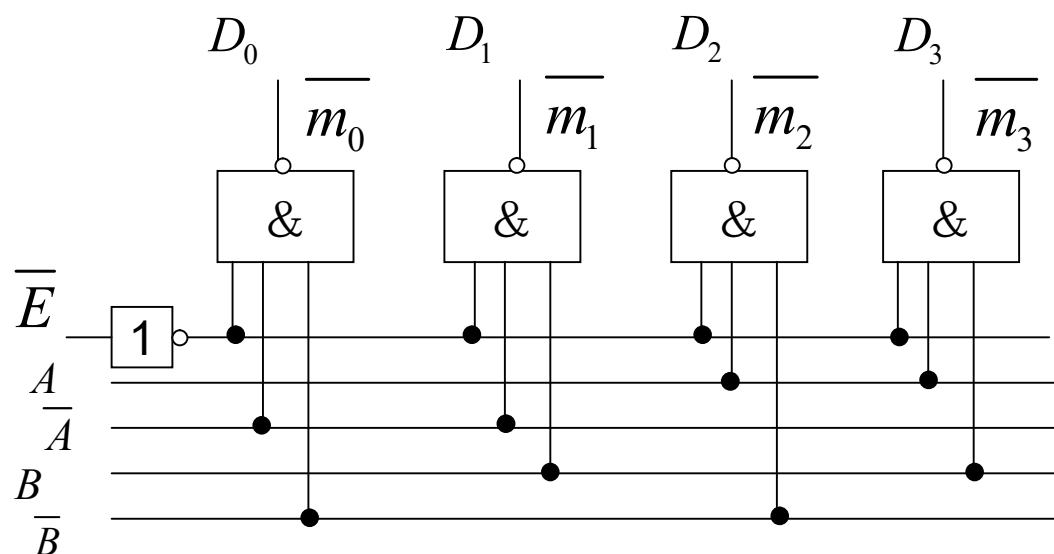
2). 低电平有效 2-4 译码器

输入		输出			
A	B	D_0	D_1	D_2	D_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

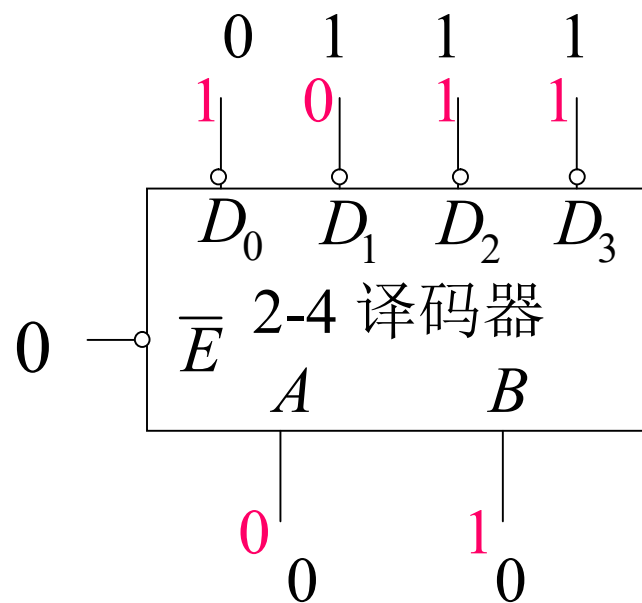
输入数码是几，第几号输出就是唯一的低电平0，其余输出均是高电平1



电路图



符号

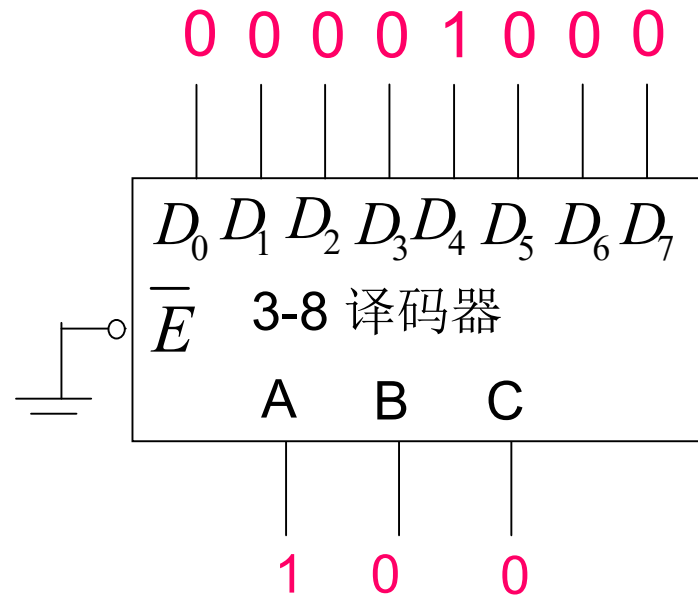


如果 $\overline{E} = 1$
 $D_0 D_1 D_2 D_3 = ?$

2. 3线-8线译码器

高电平有效3-8译码器

符号



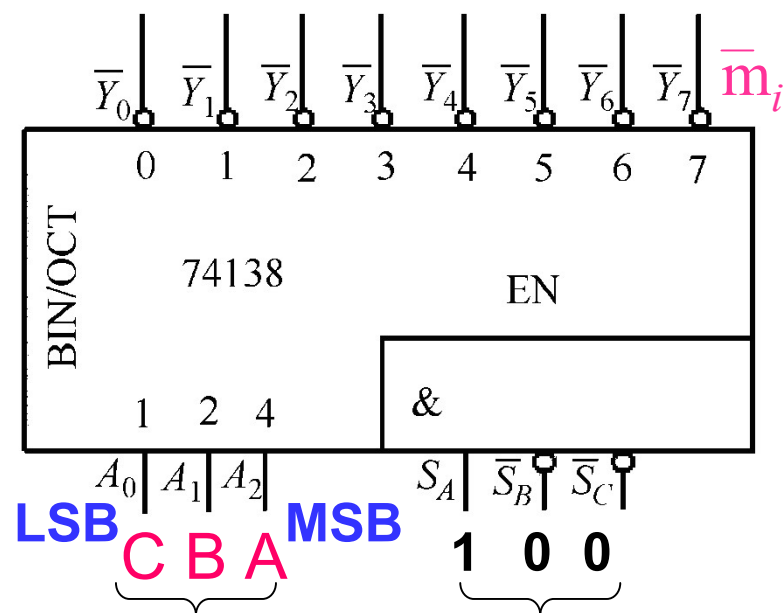
低电平有效 3-8 译码器：集成芯片74138

3线 数据输入

8线 输出

3个 使能端

$\begin{cases} S_A & \text{高电平有效} \\ \bar{S}_B & \\ \bar{S}_C & \end{cases}$ 低电平有效

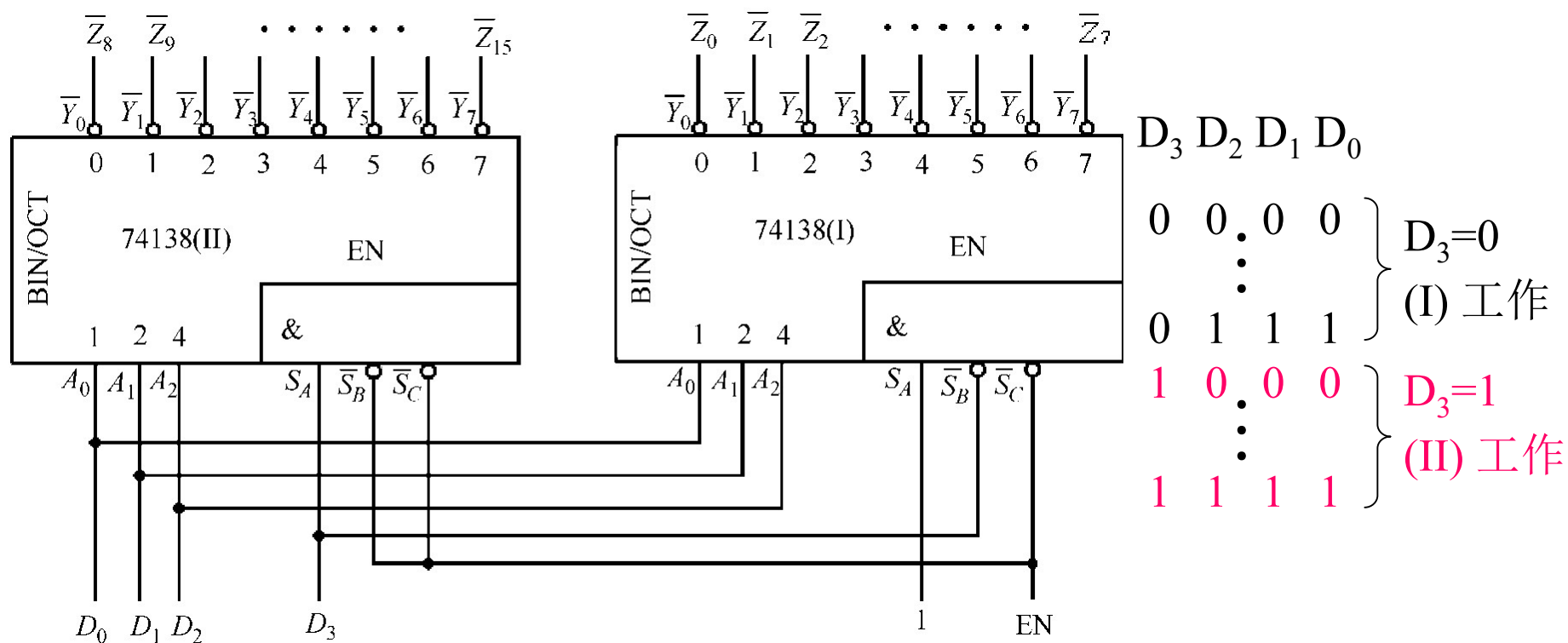


数据输入 使能输入

74138: MSI（中规模集成电路）

例: 使用74138芯片把3-8 译码器扩展为4-16 译码器

使用使能输入



4-16 译码器:

D_3 : $S_A(II)$ 连接 $\bar{S}_B(I)$ 作为 4-16 译码器的MSB 最高位
 $\bar{S}_B, \bar{S}_C(II)$ 和 $\bar{S}_C(I)$ 作为4-16译码器的总使能端

3. 使用译码器实现逻辑功能

例: 使用译码器和逻辑门实现以下功能函数

$$F_1(A, B, C) = \overline{A}\overline{B} + B\overline{C} + \overline{A} \cdot \overline{C}$$

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

标准式

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

F ₂ AB					
C		00	01	11	10
	0		0		
	1		0	0	

$$F_1(A, B, C) = \sum (0, 2, 4, 5, 6) = \prod (1, 3, 7)$$

$$F_2(A, B, C) = \sum (0, 1, 4, 5, 6) = \prod (2, 3, 7)$$

方法1:

译码器 + 或门

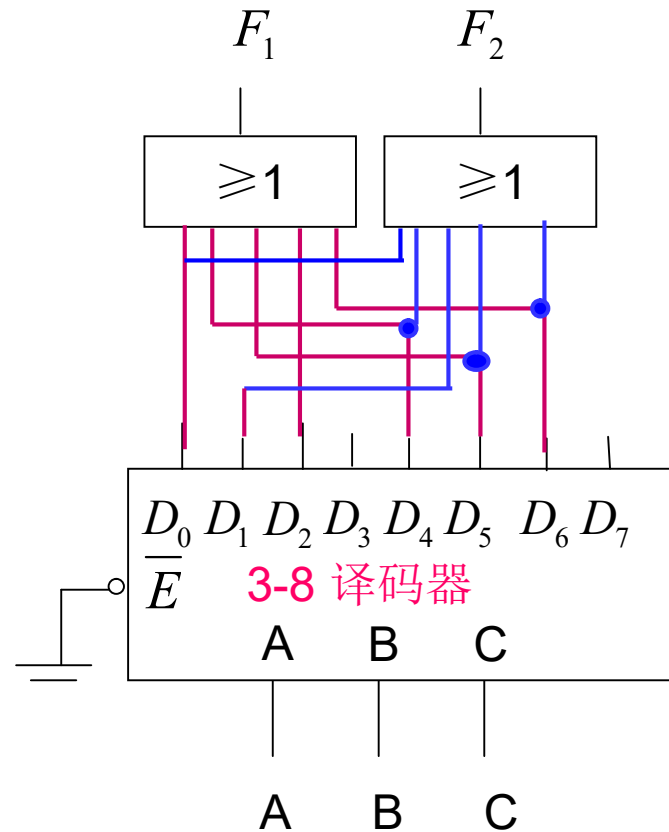
使用高电平有效译码器

输出: 最小项

标准与或式

$$F_1(A, B, C) = \sum (0, 2, 4, 5, 6)$$

$$F_2(A, B, C) = \sum (0, 1, 4, 5, 6)$$



方法 2:

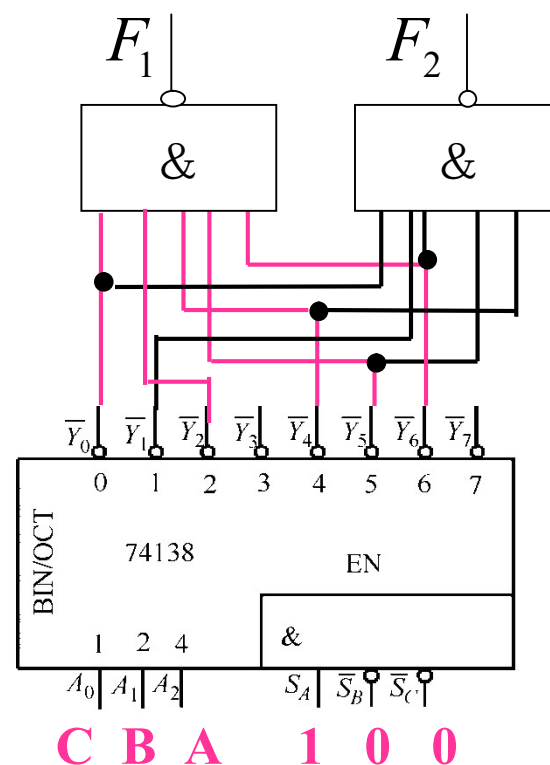
译码器+ 与非门

使用低电平有效译码器(74138)

与非门 → 最小项

$$\begin{aligned} F_1(A,B,C) &= m_0 + m_2 + m_4 + m_5 + m_6 \\ &= \overline{\overline{m_0 + m_2 + m_4 + m_5 + m_6}} \\ &= \overline{\bar{m}_0 \cdot \bar{m}_2 \cdot \bar{m}_4 \cdot \bar{m}_5 \cdot \bar{m}_6} \end{aligned}$$

标准与或式 → 与非门

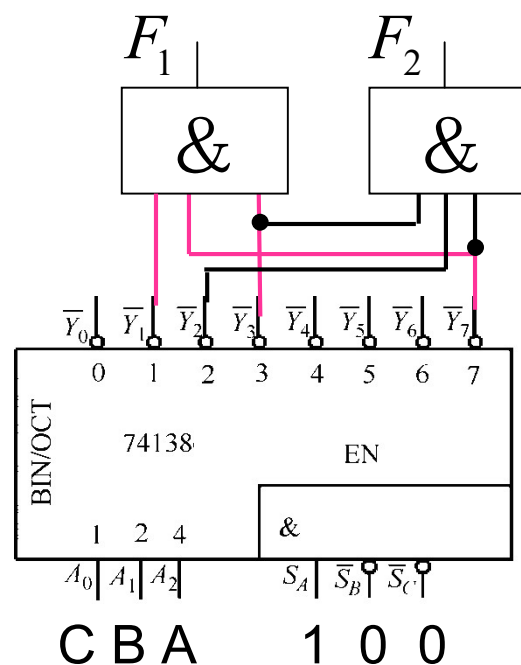


方法 3: 译码器 + 与门

低电平有效译码器(74138)

$$\begin{aligned}F_1(A,B,C) &= \Pi (1,3,7) \\&= M_1 \cdot M_3 \cdot M_7 \\&= \bar{m}_1 \cdot \bar{m}_3 \cdot \bar{m}_7\end{aligned}$$

$$\begin{aligned}F_2(A,B,C) &= \Pi (2,3,7) \\&= M_2 \cdot M_3 \cdot M_7 \\&= \bar{m}_2 \cdot \bar{m}_3 \cdot \bar{m}_7\end{aligned}$$



标准与或式: 低电平有效译码器+ 与门

总结:

使用一个译码器实现一组逻辑函数

高电平有效译码器 + 或门 (最小项)

低电平有效译码器 + 与门 (与非门)



4.4.2 BCD-十进制译码器

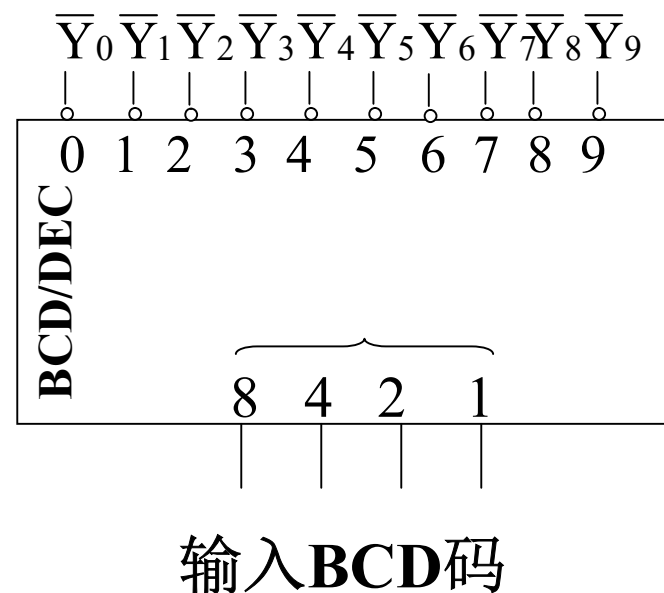
功能: 将**BCD** 码转换为为十进制码。

4-10 译码器集成芯片 7442

注意:

输出: 低电平有效

输入: { 有效输入 0000-1001
 无效输入 1010-1111



真值表

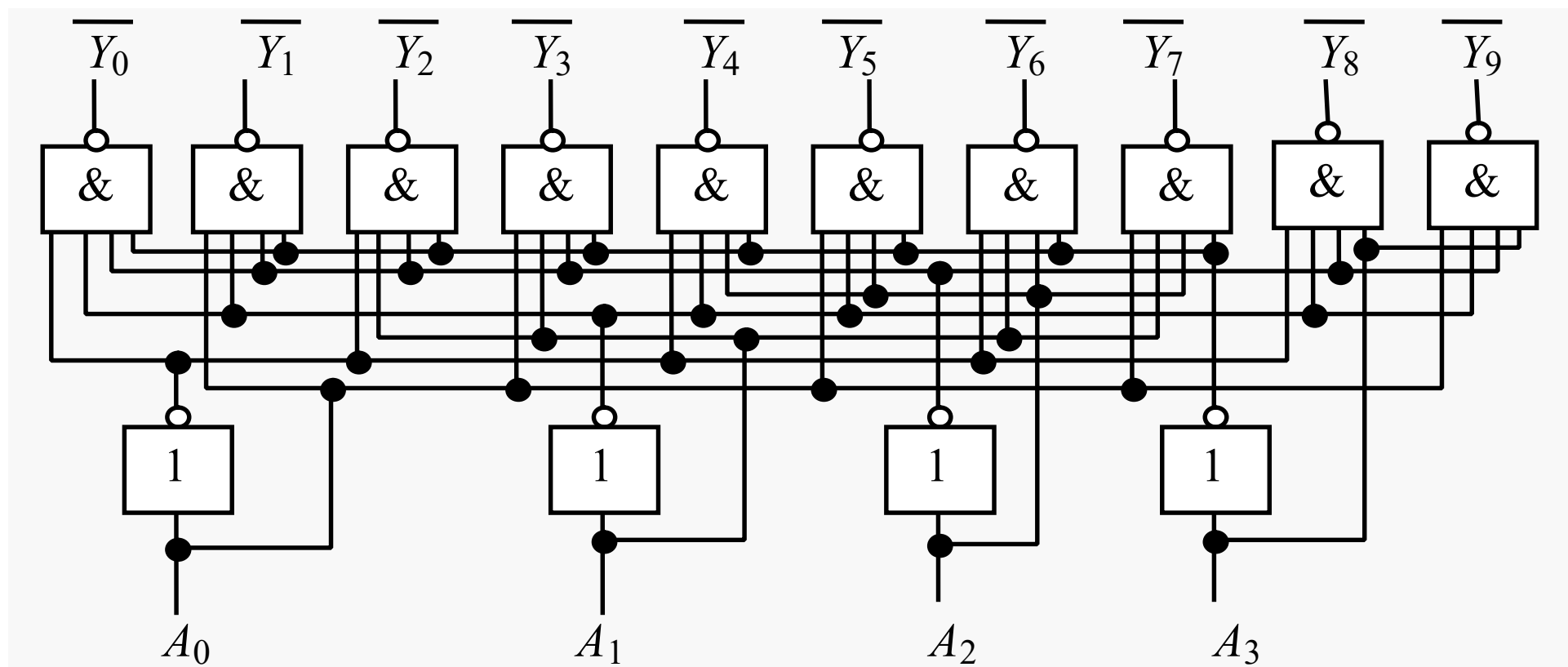
[illegible]

逻辑表达式

采用完全译码方案

$$\begin{array}{lll} \overline{Y_0} = \overline{\overline{A_3} \overline{A_2} \overline{A_1} \overline{A_0}} & \overline{Y_1} = \overline{\overline{A_3} \overline{A_2} \overline{A_1} A_0} & \overline{Y_2} = \overline{\overline{A_3} \overline{A_2} A_1 \overline{A_0}} \\ \overline{Y_3} = \overline{\overline{A_3} \overline{A_2} A_1 A_0} & \overline{Y_4} = \overline{\overline{A_3} A_2 \overline{A_1} \overline{A_0}} & \overline{Y_5} = \overline{\overline{A_3} A_2 \overline{A_1} A_0} \\ \overline{Y_6} = \overline{\overline{A_3} A_2 A_1 \overline{A_0}} & \overline{Y_7} = \overline{\overline{A_3} A_2 A_1 A_0} & \overline{Y_8} = \overline{A_3 \overline{A_2} \overline{A_1} \overline{A_0}} \\ \overline{Y_9} = \overline{A_3 \overline{A_2} \overline{A_1} A_0} & & \end{array}$$

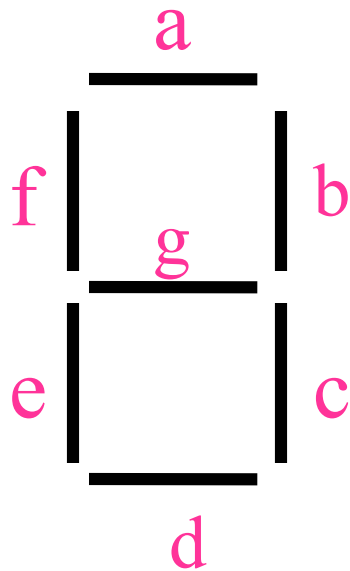
逻辑图



4.4.3 显示译码器(/驱动器)

1. 七段字符显示器

7 段数码管显示器是常见的显示器

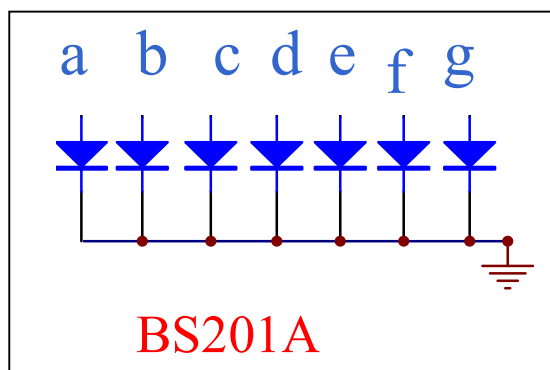


数码管由7段发光管构成

LED: 发光二极管

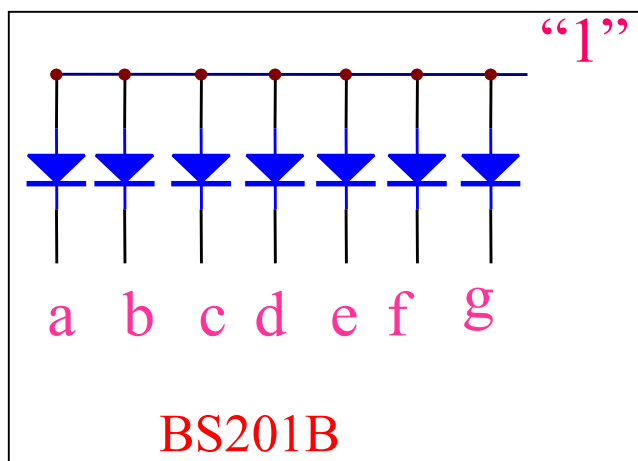
LCD: 液晶显示器

连接： 连接方式不同分成共阴极和共阳极两种



共阴极

二极管 → 逻辑高 → 亮



共阳极

二极管 → 逻辑低 → 亮

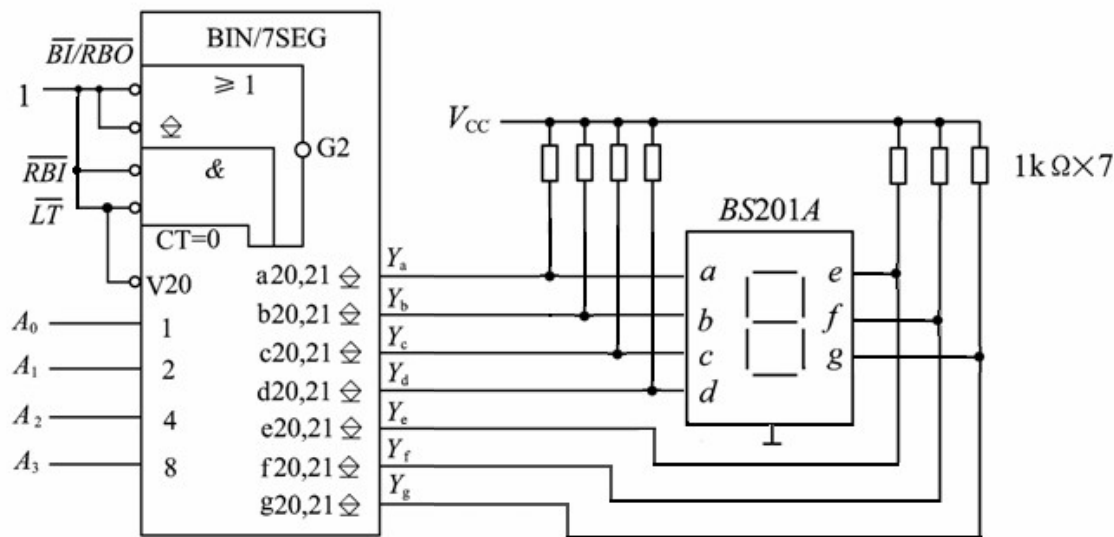
2. 显示译码器

要显示**0-9**十个数字，需要用译码器来驱动

BCD码七段显示译码器7448

输入 4 条线 4位二进制/ 8421BCD码

输出 7 条线 \longrightarrow 驱动七段字符显示器



输出高有效，
驱动共阴极管

不一定只有一个输出端高（或低）有效



辅助端功能

为了增强器件的功能，在 74LS48 中还设置了一些辅助端。这些辅助端的功能如下：

(1) 试灯输入端 \overline{LT} ：低电平有效。当 $\overline{LT} = 0$ 时，数码管的七段应全亮，与输入的译码信号无关。本输入端用于测试数码管的好坏。

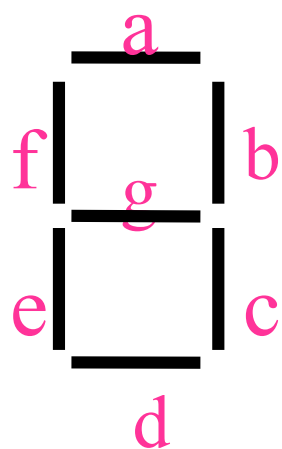
(2) 动态灭零输入端 \overline{RBI} ：低电平有效。当 $\overline{LT} = 1$ 、 $\overline{RBI} = 0$ 、且译码输入全为 0 时，该位输出不显示，即 0 字被熄灭；当译码输入不全为 0 时，该位正常显示。本输入端用于消隐无效的 0。如数据 0034.50 可显示为 34.5。

(3) 灭灯输入/动态灭零输出端 $\overline{BI} / \overline{RBO}$ ：这是一个特殊的端钮，有时用作输入，有时用作输出。

当 $\overline{BI} / \overline{RBO}$ 作为输入使用，且 $\overline{BI} / \overline{RBO} = 0$ 时，数码管七段全灭，与译码输入无关。

当 $\overline{BI} / \overline{RBO}$ 作为输出使用时，当 $\overline{LT} = 1$ 且 $\overline{RBI} = 0$ ，译码输入全为 0 时， $\overline{BI} / \overline{RBO} = 0$ ；其它情况下 $\overline{BI} / \overline{RBO} = 1$ 。本端钮主要用于显示多位数字时，多个译码器之间的连接。

显示译码器内部电路设计



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 1	0 1 1 0 0 0 0
0 0 1 0	1 1 0 1 1 0 1
0 0 1 1	1 1 1 1 0 0 1
0 1 0 0	0 1 1 0 0 1 1
0 1 0 1	1 0 1 1 0 1 1
0 1 1 0	0 0 1 1 1 1 1
0 1 1 1	1 1 1 0 0 0 0
1 0 0 0	1 1 1 1 1 1 1
1 0 0 1	1 1 1 0 0 1 1

输出

0

:

2

3

4

5

6

7

8

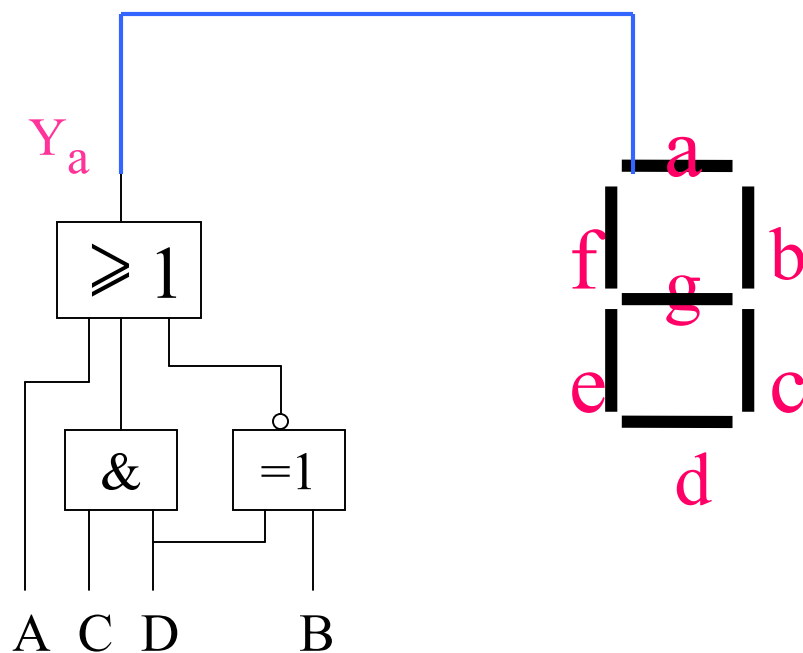
9

分别做7个卡诺图

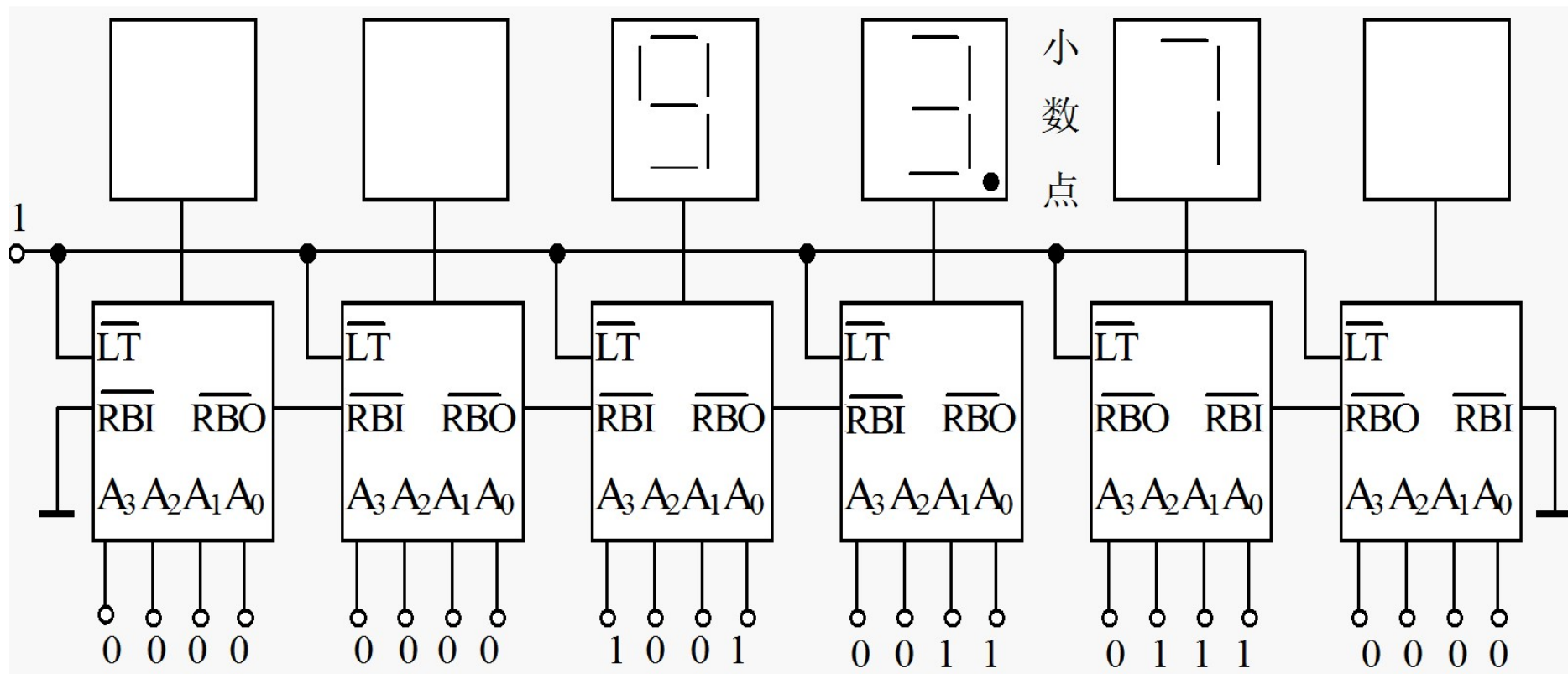
$$Y_a = A + \overline{B} \cdot \overline{D} + BD + CD$$

$$= A + CD + \overline{B \oplus D}$$

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



数码显示电路的动态灭零



整数部分：高位的 $\overline{BI} / \overline{RBO}$ 与低位的 \overline{RBI} 相连

小数部分：低位的 $\overline{BI} / \overline{RBO}$ 与高位的 \overline{RBI} 相连

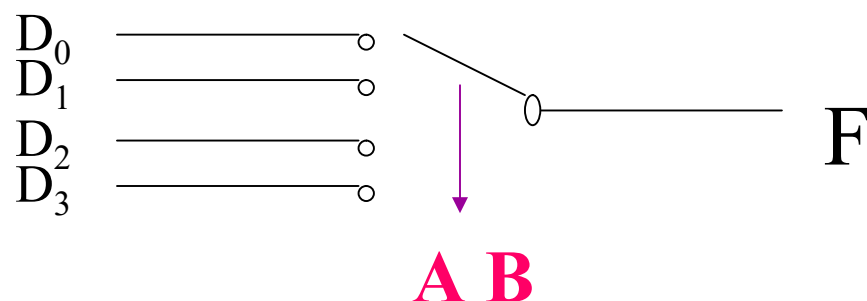
§ 4.5 多路（数据）选择器

多路选择器功能:

将多个输入数据中的一个送到唯一的输出端

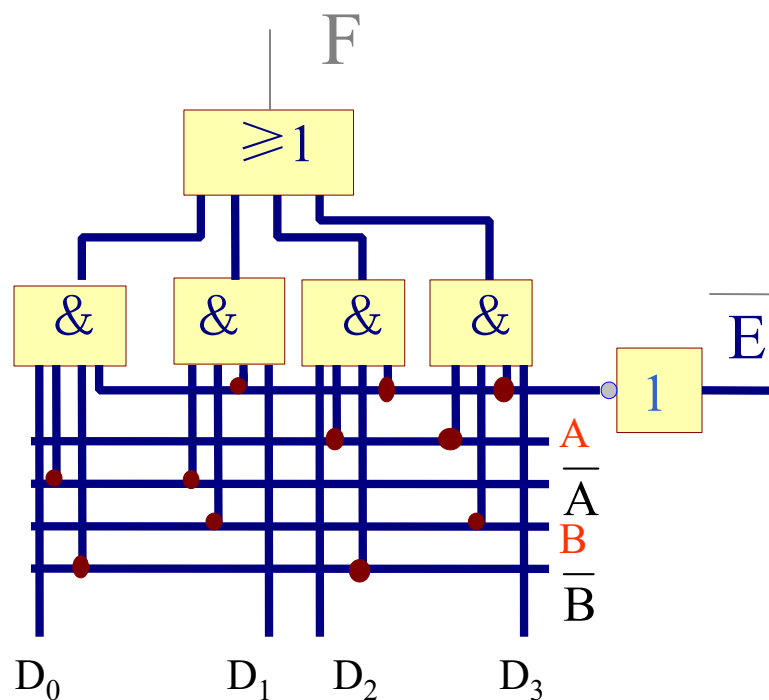
1. 四选一多路选择器

相当于4个数据 D_0, D_1, D_2, D_3 中选一个，由开关A B控制



A B: 控制输入(地址输入)

n 位地址线可以控制 2^n 个数据输入



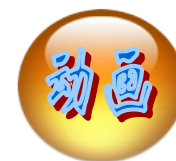
\overline{E}	A	B	F
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	ϕ	ϕ	0

A B任取一值时，只有一个与门输出1(D)，其他为0，或之后为F。

\overline{E} 为使能端。在 $\overline{E} = 0$ 的条件下，

控制码是几，就把第几号数据送到唯一的输出端

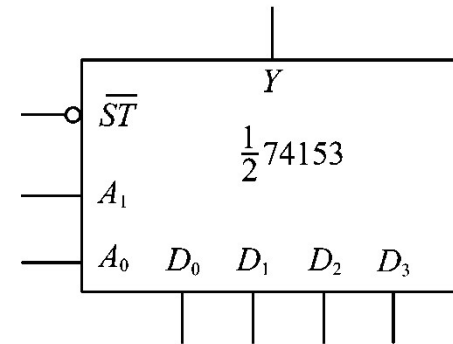
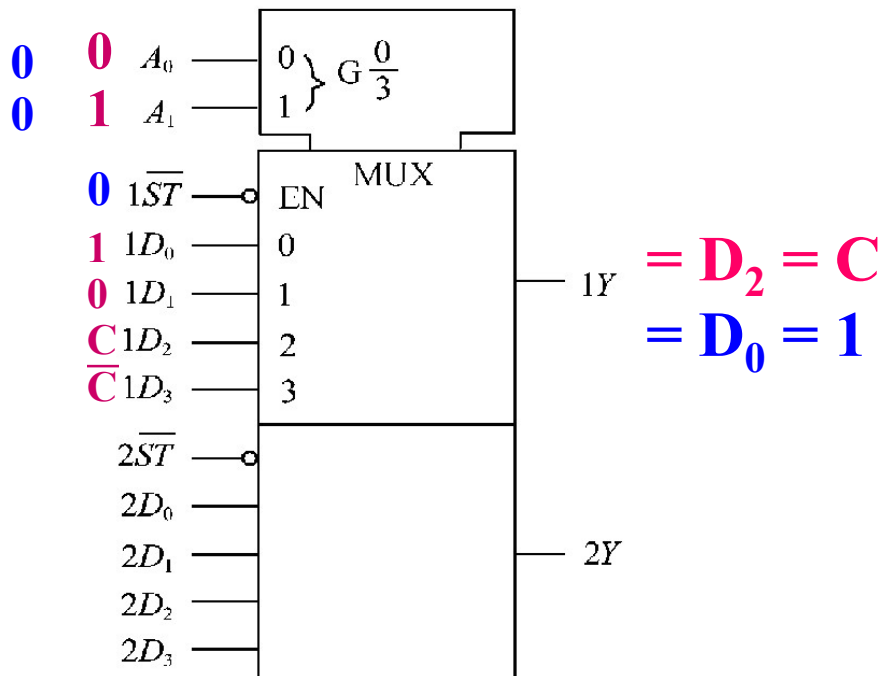
译码器 + 数据线 + 或门



双四选一多路选择器74153

(一芯片上有 2 个 4选1 多路选择器)

符号

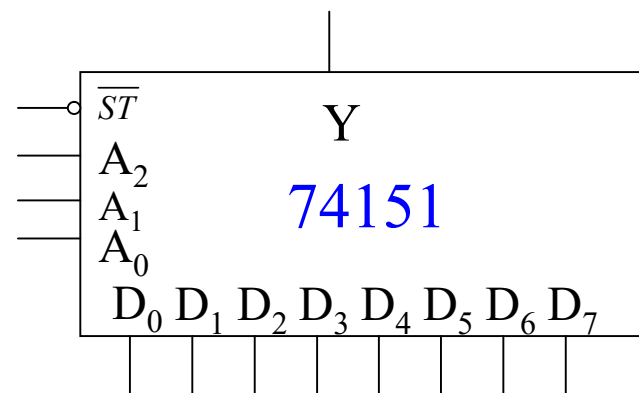
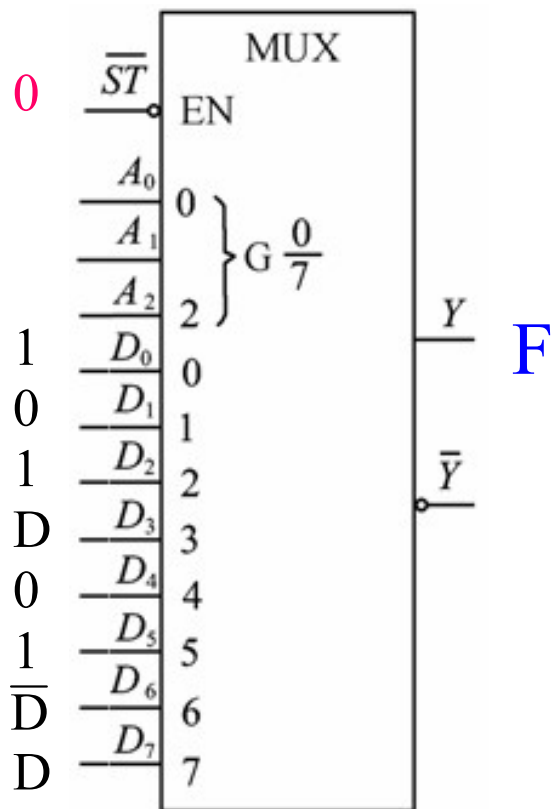


\overline{ST} 片选控制 低电平有效

$A_1 A_0$: 地址线 (控制输入)

2. 八选一多路选择器 74151 (MSI)

3个地址线: $A_2 A_1 A_0$; 8 个数据输入: $D_0 - D_7$



地址输入

A_2	A_1	A_0	F
1	0	0	0
0	1	1	D
0	0	0	1
1	1	0	\overline{D}

3. 使用多路选择器实现逻辑函数

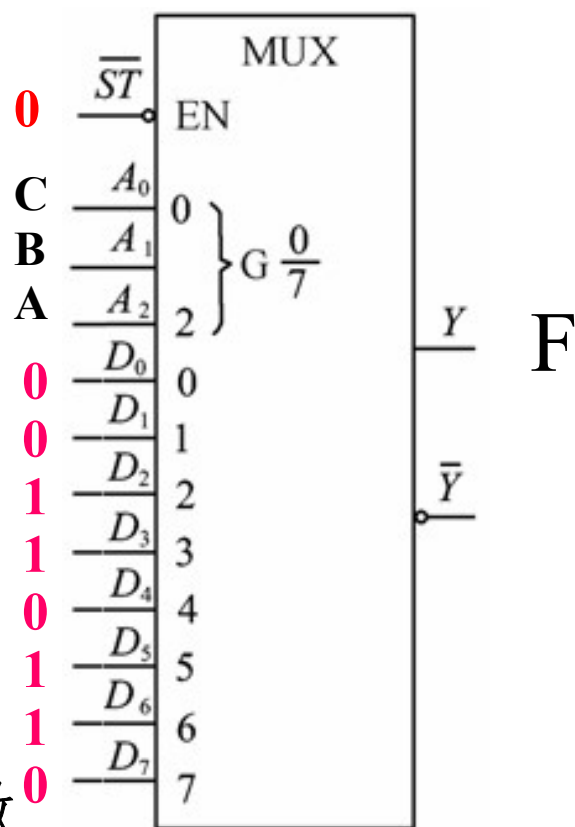
例1：使用多路选择器实现

$$F(A,B,C) = \bar{A}BC + B\bar{C} + A\bar{B}C = \sum (2, 3, 5, 6)$$

解： 3 变量

选择74151 (八选一多路选择器)

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



一个多路选择器只能实现一个逻辑函数

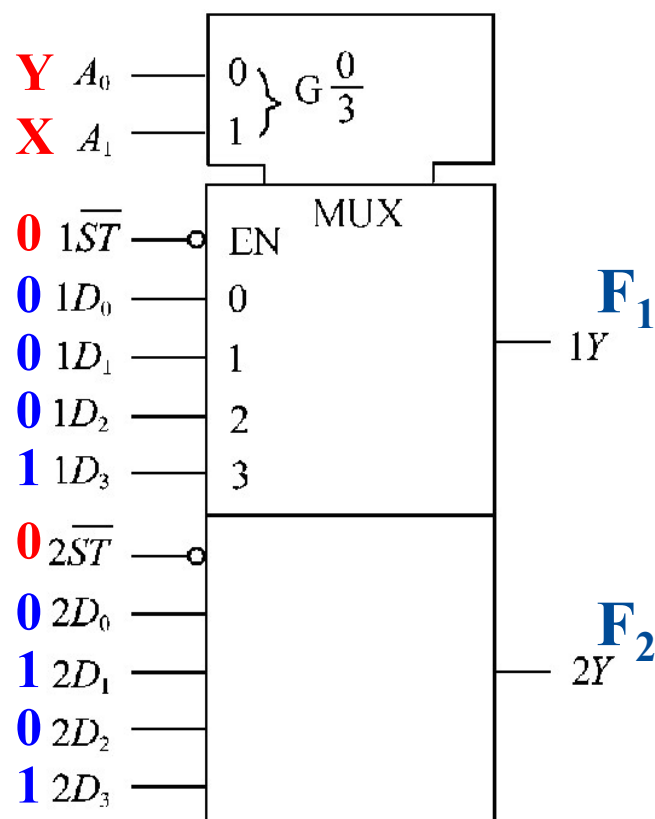
例 2: 使用双四选一多路选择器74153实现以下逻辑函数

$$F_1(X,Y) = X(\bar{X} + Y) = \textcolor{red}{XY} = m_3$$

$$F_2(X,Y) = \prod(0,2) = \textcolor{red}{m_1 + m_3}$$

解:

标准形式



例3: 使用一片74151 实现

$$F(A, B, C, D) = ABCD + \overline{A}\overline{B}C\overline{D} + \overline{A}BCD + \overline{A}\overline{B}C + ABC\overline{D}$$

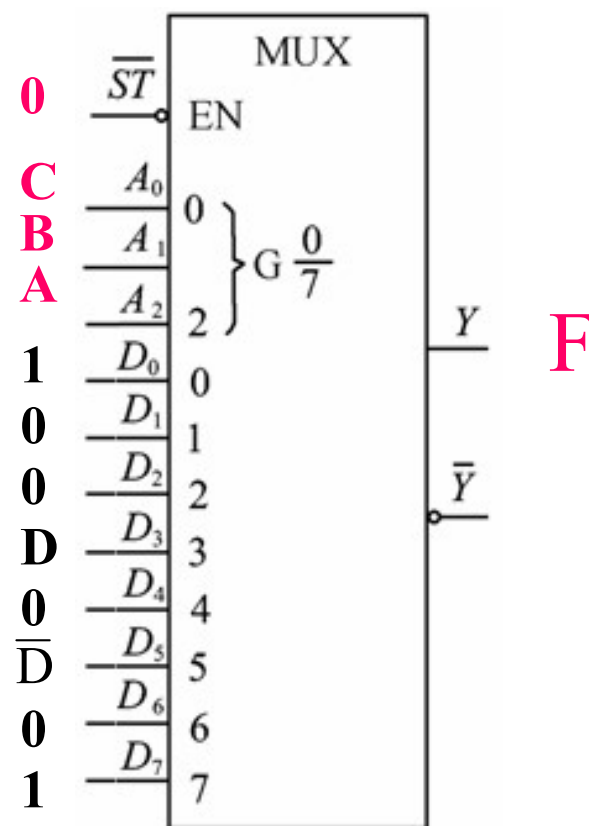
解:

74151 3 变量

D → VEM

F AB					
C		00	01	11	10
	0	1	0	0	0
	1	0	D	D + \overline{D}	\overline{D}

↓
1



例 4. 使用一片四选一选择器实现

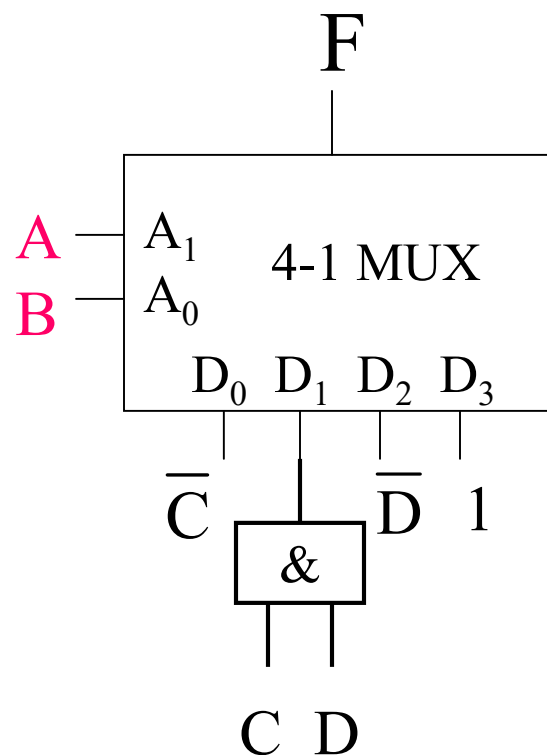
$$F(A,B,C,D) = \overline{A}BCD + AB + \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot \overline{D}$$

解：

四选一多路选择器 2 变量

2 变量 **C D** → VEM

		A	
		0	1
B	0	\overline{C}	\overline{D}
	1	CD	1



§ 4.6 数值比较器

基本功能:

比较两个二进制数的大小

4.6.1 一位数值比较器

输入: A, B

输出:

比较结果:

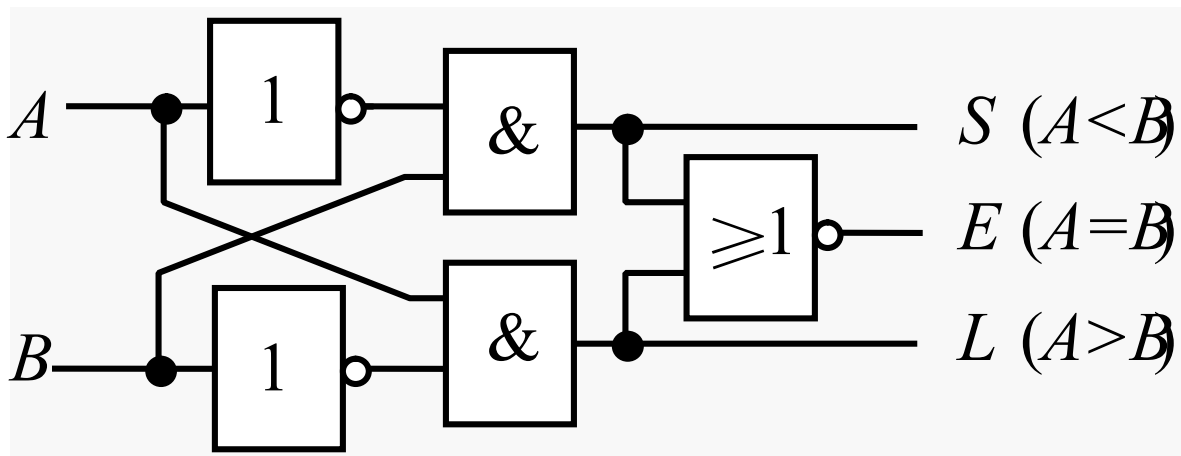
$$\left\{ \begin{array}{ll} L (A > B) & \text{大于} \\ S (A < B) & \text{小于} \\ E (A = B) & \text{等于} \end{array} \right\} \begin{array}{l} \text{高电平有效} \\ \text{是 } 1 \\ \text{否 } 0 \end{array}$$

真值表

A	B	L	S	E
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

$$\therefore \begin{cases} L = \overline{A}B \\ S = \overline{A}\overline{B} \\ E = AB + \overline{A} \cdot \overline{B} \\ \quad = A \odot B \end{cases}$$

电路



4.6.2 四位数值比较器

$$\begin{array}{l} \text{8 输入} \quad \left\{ \begin{array}{l} A: A_3 A_2 A_1 A_0 \\ B: B_3 B_2 B_1 B_0 \end{array} \right. \quad \text{3 输出} \quad \left\{ \begin{array}{l} L (A > B) \\ S (A < B) \\ E (A = B) \end{array} \right. \end{array}$$

从最高位开始，当找到一个不相等的位，两个数的关系就可以得出。

MSI四位数值比较器7485包括：

$$\begin{array}{l} \text{3个级联输入端} \quad \left\{ \begin{array}{l} \mathbf{l} (A > B) \\ \mathbf{s} (A < B) \\ \mathbf{e} (A = B) \end{array} \right. \quad \text{低位比较的结果} \end{array}$$

真值表：

比较输入				级联输入			输出		
$A_3 B_3$	$A_2 B_2$	$A_1 B_1$	$A_0 B_0$	$l(A' > B')$	$s(A' < B')$	$e(A' = B')$	$L(A > B)$	$S(A < B)$	$E(A = B)$
$A_3 > B_3$	X	X	X	X	X	X	1	0	0
$A_3 < B_3$	X	X	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	1	0	0
E_3	$A_2 < B_2$	X	X	X	X	X	0	1	0
E_3	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	1	0	0
E_3	E_2	$A_1 < B_1$	X	X	X	X	0	1	0
E_3	E_2	$A_1 = B_1$	$A_0 > B_0$	X	X	X	1	0	0
E_3	E_2	E_1	$A_0 < B_0$	X	X	X	0	1	0
E_3	E_2	E_1	$A_0 = B_0$	1	0	0	1	0	0
E_3	E_2	E_1	E_0	0	1	0	0	1	0
E_3	E_2	E_1	E_0	0	0	1	0	0	1

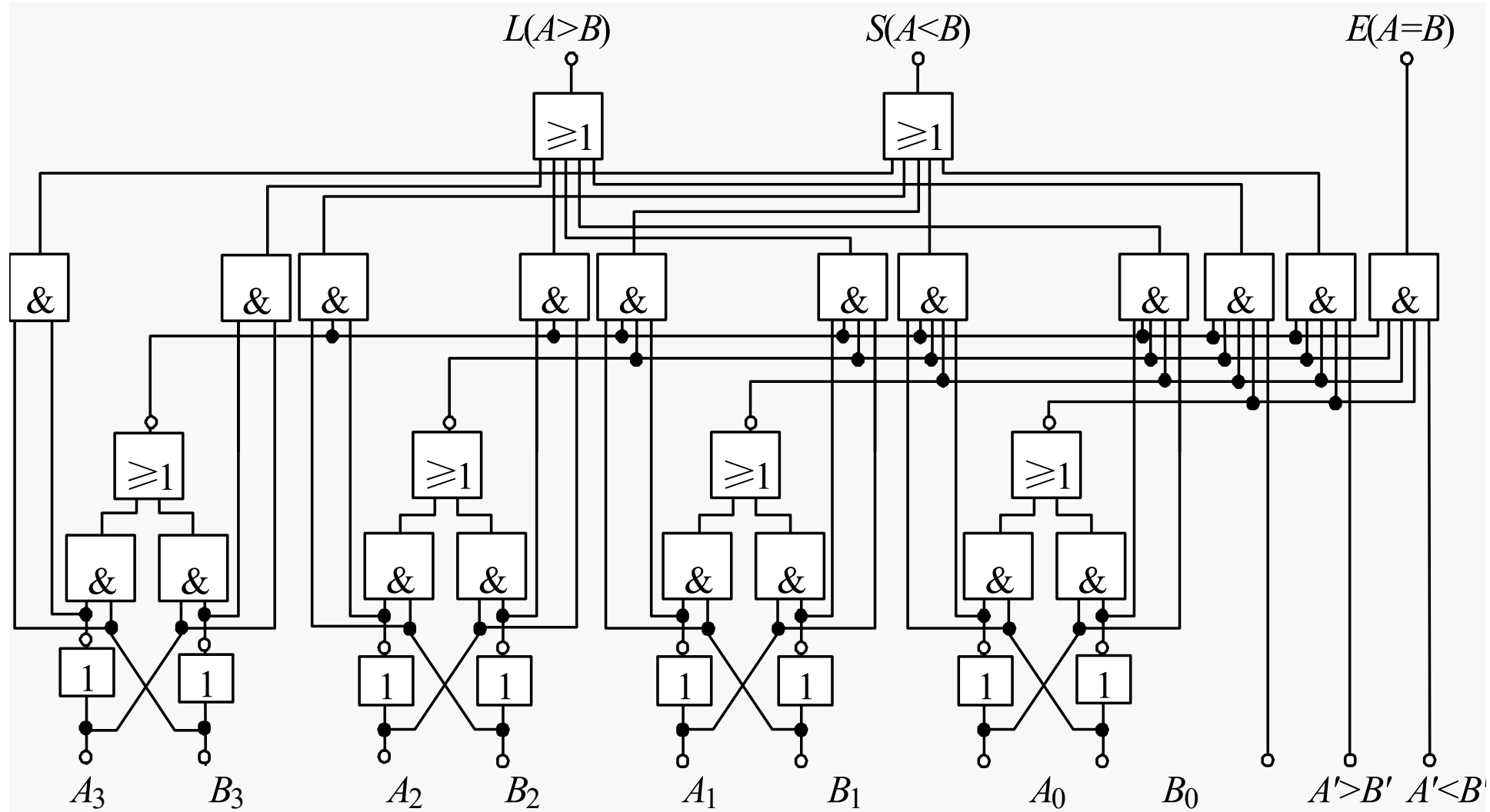
真值表中的输入变量包括 A_3 与 B_3 、 A_2 与 B_2 、 A_1 与 B_1 、 A_0 与 B_0 以及另外两个低位数 A' 与 B' 的比较结果。设置低位数比较结果输入端，是为了能与其它数值比较器连接，以便组成更多位数的数值比较器；3个输出信号 $L(A > B)$ 、 $S(A < B)$ 、和 $E(A = B)$ 分别表示本级的比较结果。

设 $l = (A' > B')$ ， $s = (A' < B')$ ， $e = (A' = B')$ ，
 $L_3 = A_3 \bar{B}_3 = (A_3 > B_3)$ ， $S_3 = \bar{A}_3 B_3 = (A_3 < B_3)$ ，
 $E_3 = \overline{\bar{A}_3 B_3 + A_3 \bar{B}_3} = (A_3 = B_3)$ ， 余类推。由真值表可得：

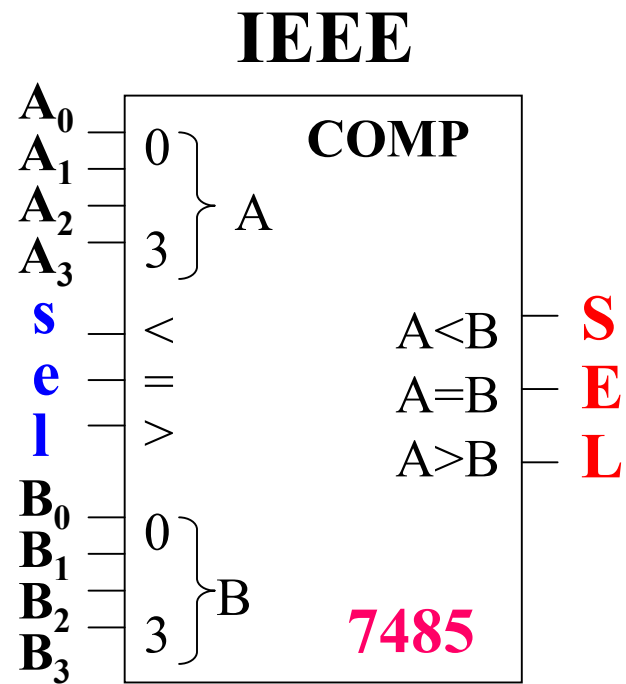
输出：

$$\begin{cases} E = E_3 E_2 E_1 E_0 e \\ L = L_3 + E_3 L_2 + E_3 E_2 L_1 + E_3 E_2 E_1 L_0 + E_3 E_2 E_1 E_0 l \\ S = S_3 + E_3 S_2 + E_3 E_2 S_1 + E_3 E_2 E_1 S_0 + E_3 E_2 E_1 E_0 s \end{cases}$$

电路图

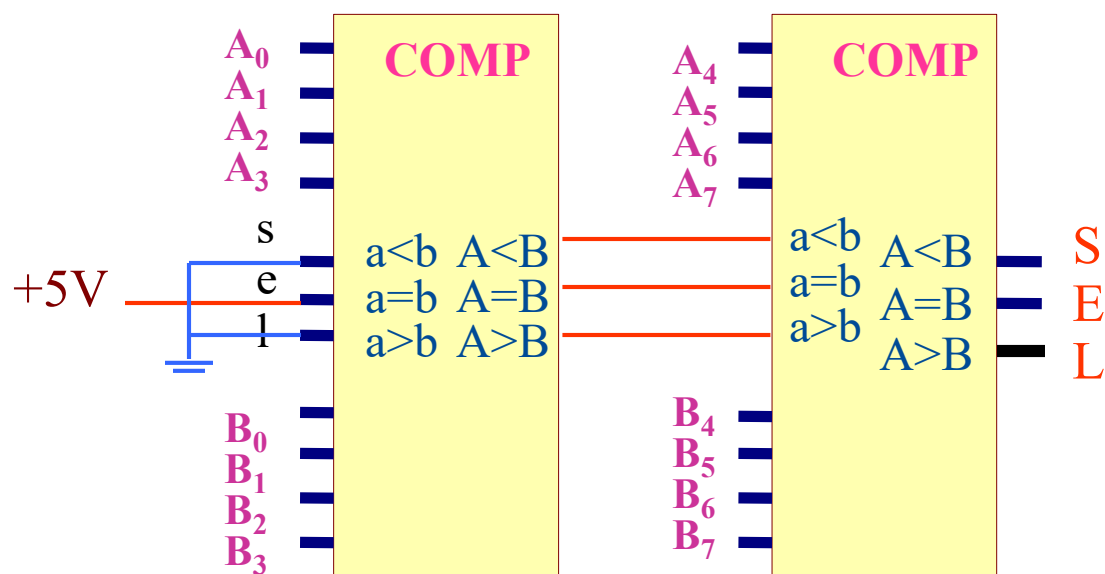


7485 符号:



4.6.3 比较器级联扩展

2片7485 连成一个8位数值比较器



先用高位片，若高位片比出结果 ($A > B$ 或 $A < B$)，则与级联输入状态无关；若高位片相等 ($A = B$)，再看级联输入，即看低位比较结果 $l s e$ ，若低位仍相等，则 $A = B$ 。

例：比较

$$\begin{cases} A = 11010111 \\ B = 10110010 \end{cases}$$

输出(L,E,S)=(1,0,0)

$$\begin{cases} A = 01000111 \\ B = 01001010 \end{cases}$$

输出(L,E,S)=(0,0,1)

§ 4.7 加法器

加法器不仅在计算机中很重要，而且在很多数字系统中发挥作用。

4.7.1 半加器

功能：实现两个一位二进制数字相加

2输入：A, B

2 输出：S (和) C_o (进位)

$$\begin{array}{r} A \\ + B \\ \hline C_o S \end{array}$$

A	B	S	C_o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

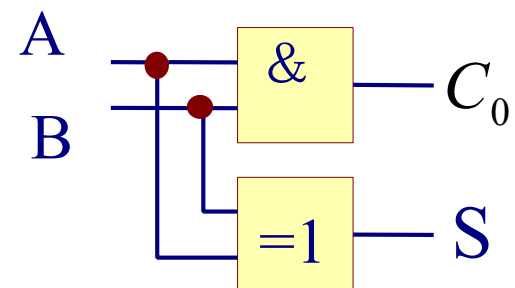
$$S = A \oplus B$$

$$C_o = AB$$

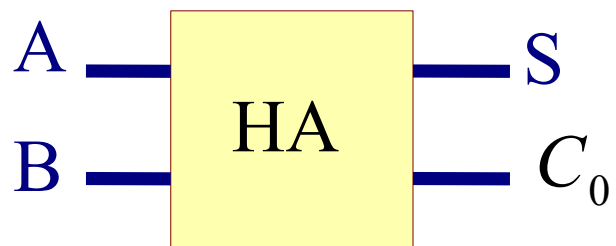
$$S = A \oplus B$$

$$C_0 = AB$$

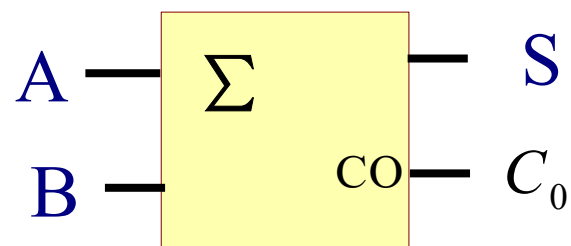
电路



符号:



IEEE



4.7.2 全加器

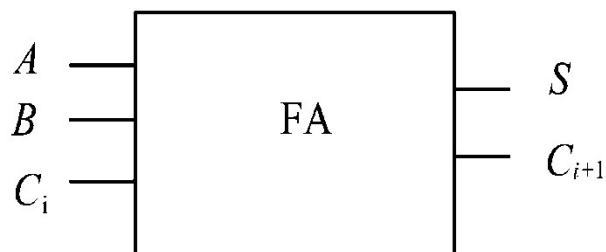
3 输入: A, B, C_i (来自低位的进位)

2 输出: S, C_{i+1} (向高位的进位)

$$S = A \oplus B \oplus C_i$$

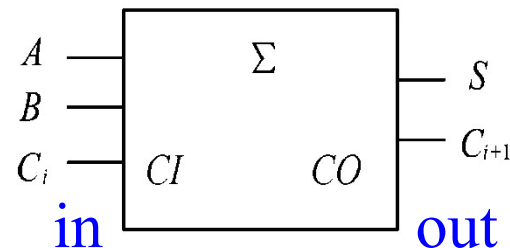
$$C_{i+1} = AB + AC_i + BC_i$$

符号



A	B	C_i	S	C_{i+1}
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

IEEE

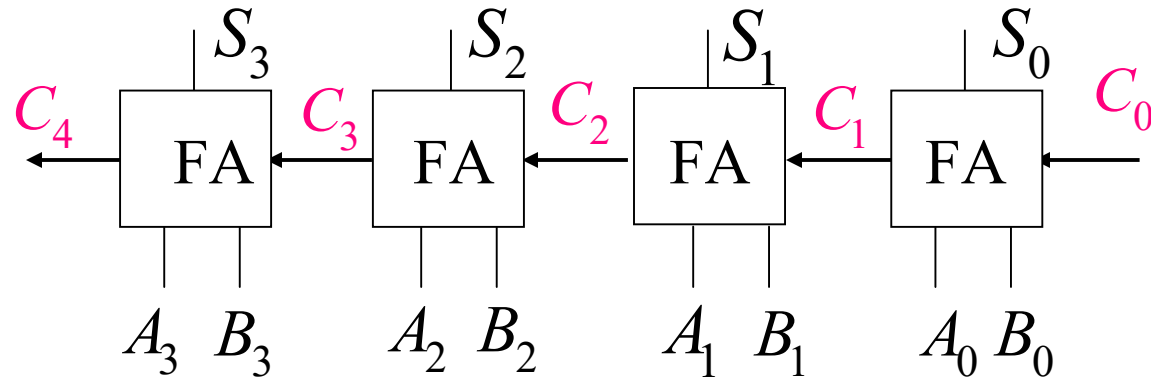


4.7.3 并行加法器

多位二进制数相加时，每位一个全加器，加法器并行

并行加法器中的进位：{ 串行(脉冲)进位
超前进位

串行(脉冲)进位：



并行输入，串行进位：结构简单，速度慢

为了加快串行进位加法效率，超前进位加法器出现了。

超前进位

分析

全加器输出：

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

定义：

$$\begin{cases} G_i = A_iB_i & \text{产生变量} \\ P_i = A_i \oplus B_i & \text{传输变量} \end{cases}$$

上式写成
$$\begin{cases} S_i = P_i \oplus C_i \\ C_{i+1} = G_i + P_iC_i \end{cases}$$

进位:

$$C_1 = G_0 + P_0 C_0$$

$$\begin{cases} S_i = P_i \oplus C_i \\ C_{i+1} = G_i + P_i C_i \end{cases}$$

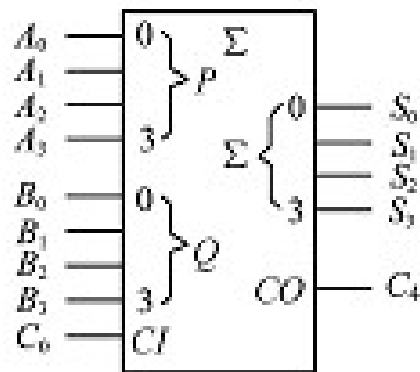
$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

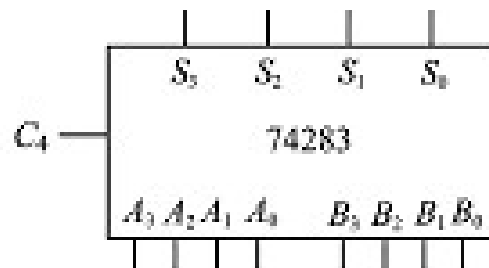
$\because C_0 = 0$, C_i 只与 G, P 有关, 即只与 A, B 有关, 可以并行产生

相当于四个全加器同时计算, 而不是第一个做完, 得到 C_1 后, 第二个再做。提高速度。

74283 加法器: 超前进位



国际标准符号



惯用符号

$$\begin{array}{r} A_3 A_2 A_1 A_0 \\ + B_3 B_2 B_1 B_0 \\ \hline C_4 \quad S_3 S_2 S_1 S_0 \end{array}$$

小 结

- 组合逻辑电路的分析与设计
- 译码器工作原理、符号、真值表、功能
 - 实现逻辑函数
- 数据选择器工作原理、符号、真值表、功能
 - 实现逻辑函数
- 数值比较器工作原理、符号、功能、级联
- 加法运算电路工作原理、符号、功能

作业：

4 .2 4 .5 4 .6 4.10

4.14 4.21 4.23

4.26 4 .28 4.30

作业：

一、某工厂有三个车间，每个车间各需要**10KW**电力。三个车间由两台发电机组供电，一台是**10KW**，另一台是**20KW**。三个车间经常不同时工作，有时只一个车间工作，也可能两个车间或三个车间同时工作。为了节省能源，又保证电力供应，试用与非门和异或门设计一个逻辑电路，自动完成配电任务。请列出真值表，写出函数表达式，画出逻辑电路图。

作业：

二、设计一个能判断 A ， B 两个数大小的比较电路。

A 、 B 都是二进制数，

$A=A_1A_0$ ， $B=B_1B_0$ ，当

$A \geq B$ 时电路输出为1。试

用8选1数据选择器构成电

路，列出真值表，写出函

数表达式，画出逻辑电路

图。

