

第13章 组合逻辑电路的分析与设计

13.1 组合逻辑电路的分析

13.2 组合逻辑电路的设计

13.3 加法器

13.4 编码器

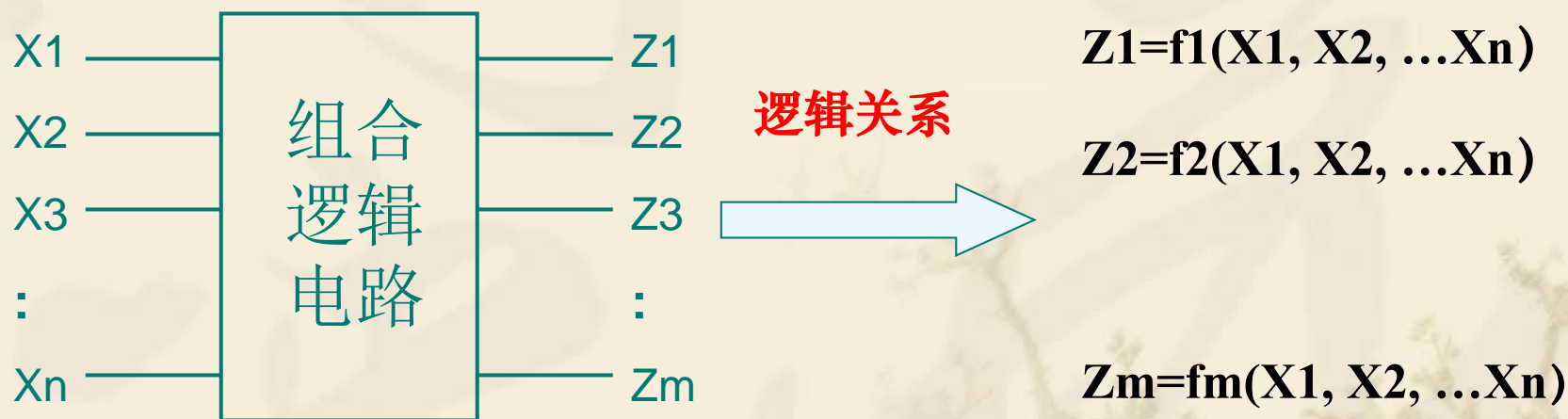
13.5 译码器和数据分配器

13.6 数据选择器

13.7 数据分配器

13.8 用MSI实现组合逻辑函数

定义：任意时刻的输出状态只决定于该时刻的输入状态，而与从前的状态（输入与输出）无关。用方框图表示如下：



电路结构特点： 由门电路组合而成，电路中没有记忆单元，没有反馈通路。

组合电路分类： 按逻辑功能特点，可分为：加法器、比较器、编码器、译码器、数据选择器、数据分配器、只读存储器等等
按所使用的基本开关元件，可分为CMOS、TTL等类型
按集成度不同，可分为SSI、MSI、LSI、VLSI

逻辑功能表示法： 真值表、卡诺图、逻辑表达式、时间图

关于组合电路，主要研究两个问题：

- (1) 若组合电路已经给定，一般要求对其进行分析
- (2) 若组合电路只给出功能要求，一般要对其进行设计

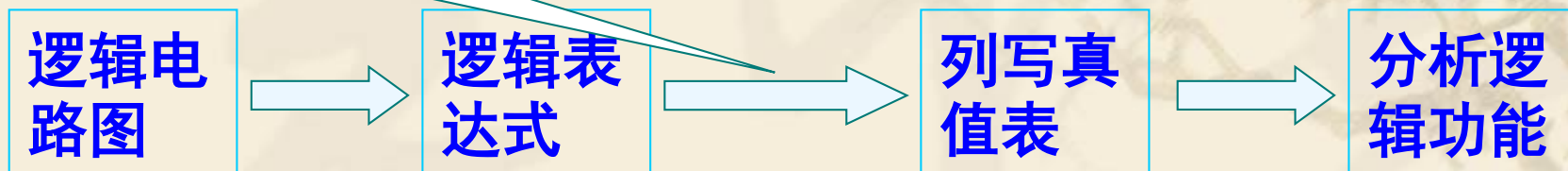
13.1 组合逻辑电路的分析

分析目的： 说明给定电路的逻辑功能

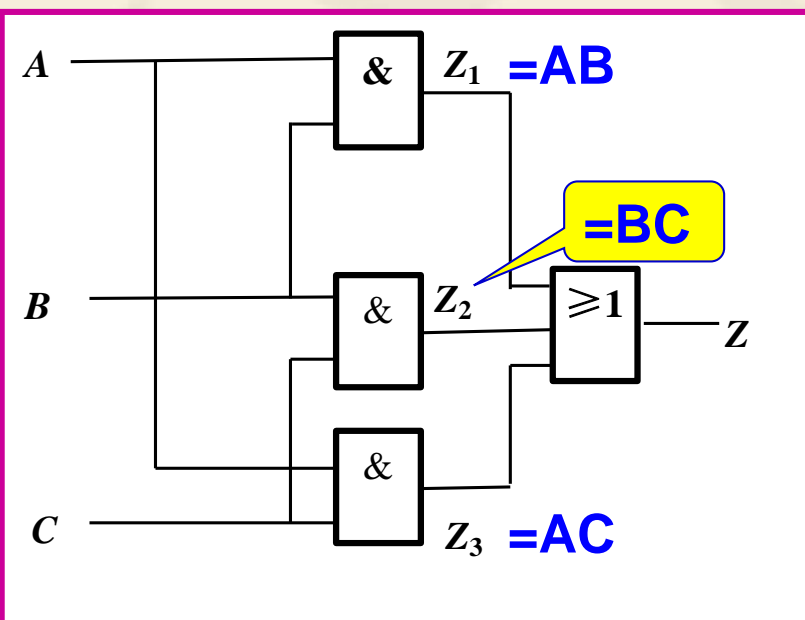
分析步骤：

- (1) 由逻辑电路图写出逻辑表达式；
- (2) 化简和变换各逻辑表达式；
- (3) 列出真值表；
- (4) 据真值表和表达式对逻辑电路进行分析，最后确定其逻辑功能。

化为最简
与或



例1 已知电路如图所示，分析说明该电路的逻辑功能。



解：1.根据逻辑电路图写出逻辑表达式：

$$Z = Z_1 + Z_2 + Z_3 = AB + BC + AC$$

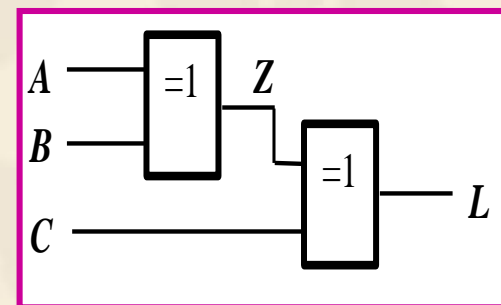
2.列出真值表

输入						输出
A	B	C	Z1	Z2	Z3	Z
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	1
1	0	0	0	0	0	0
1	0	1	0	0	1	1
1	1	0	1	0	0	1
1	1	1	1	1	1	1

3.分析逻辑功能

由真值表可知：ABC三个变量中，有两个或两个以上为“1”时，输出为“1”；有两个或两个以上为“0”时，输出为“0”；所以，这是一个：**多数表决电路**

例2 已知电路如图所示，试分析该电路的逻辑功能。



解：1.根据逻辑电路图写出逻辑表达式

$$Z = A \oplus B$$

$$L = Z \oplus C$$

$$= (A \oplus B) \oplus C$$

$$= A \oplus B \oplus C$$

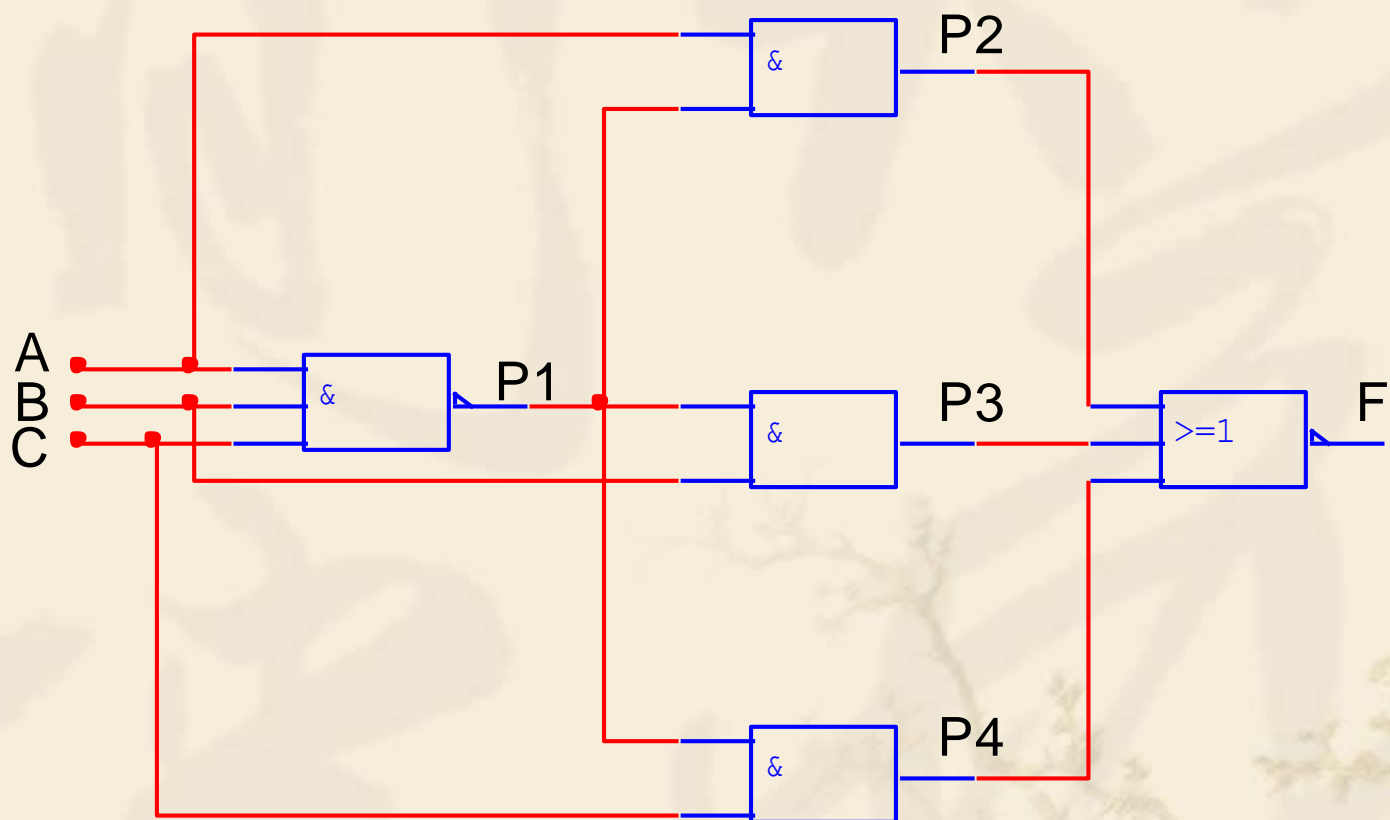
2. 列出真值表

3. 确定逻辑功能：

奇校验电路

A	B	C	$Z = A \oplus B$	$L = (A \oplus B \oplus C)$
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

例3 已知电路如图所示，试分析该电路的逻辑功能。



解： 1.根据逻辑电路图写出逻辑表达式

$$P_1 = \overline{ABC}$$

$$P_2 = A \cdot P_1 = A \cdot \overline{ABC}$$

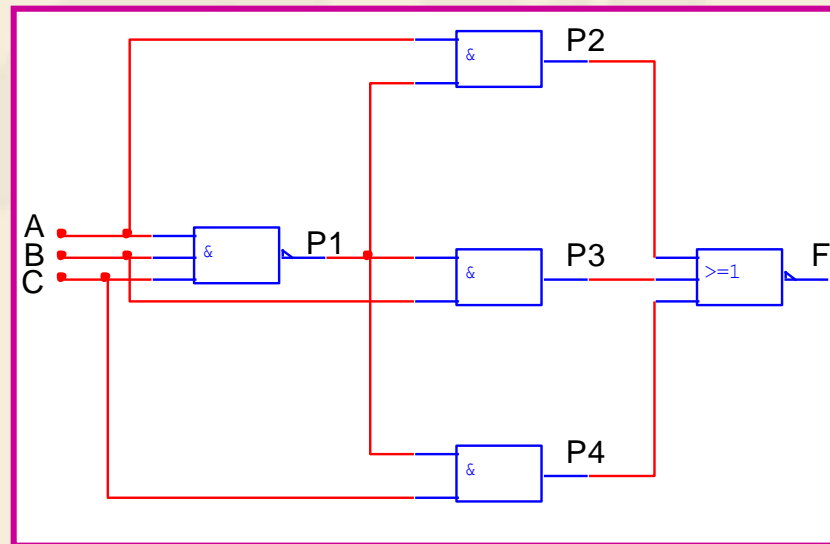
$$P_3 = B \cdot P_1 = B \cdot \overline{ABC}$$

$$P_4 = C \cdot P_1 = C \cdot \overline{ABC}$$

$$F = \overline{P_2 + P_3 + P_4} = \overline{A \cdot \overline{ABC} + B \cdot \overline{ABC} + C \cdot \overline{ABC}}$$

化简得：

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



2. 列出真值表

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

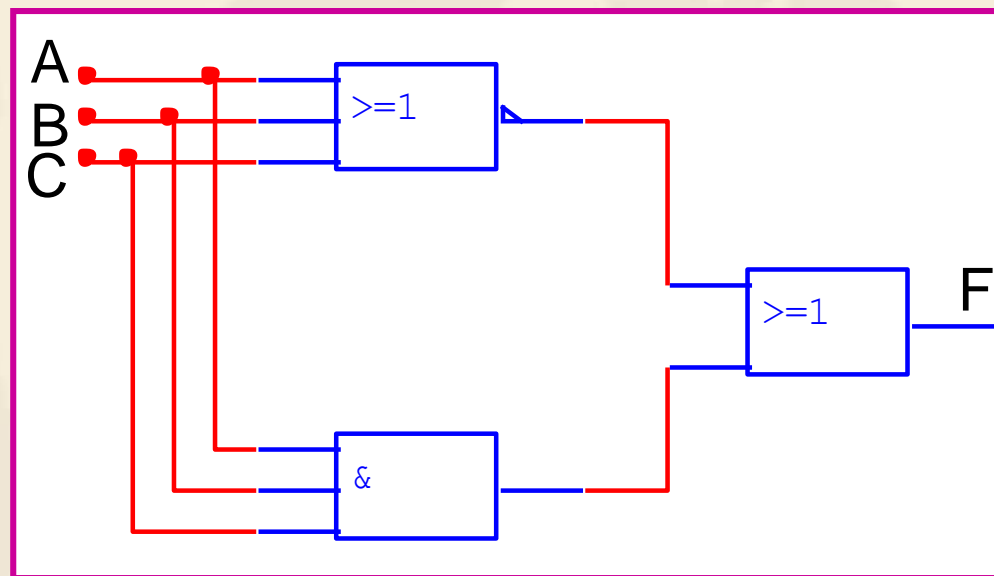
3. 确定逻辑功能:

由真值表可知, 当A、B、C取相同值时, F 为1, 否则 F 为0。所以该电路是一个"一致性电路"。

A	B	C	F
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

另外，上面的电路其实可以做进一步的简化

$$\begin{aligned} F &= \overline{\overline{ABC}(A+B+C)} \\ &= ABC + \overline{A+B+C} \end{aligned}$$



思考：不一致电路应是怎样的？

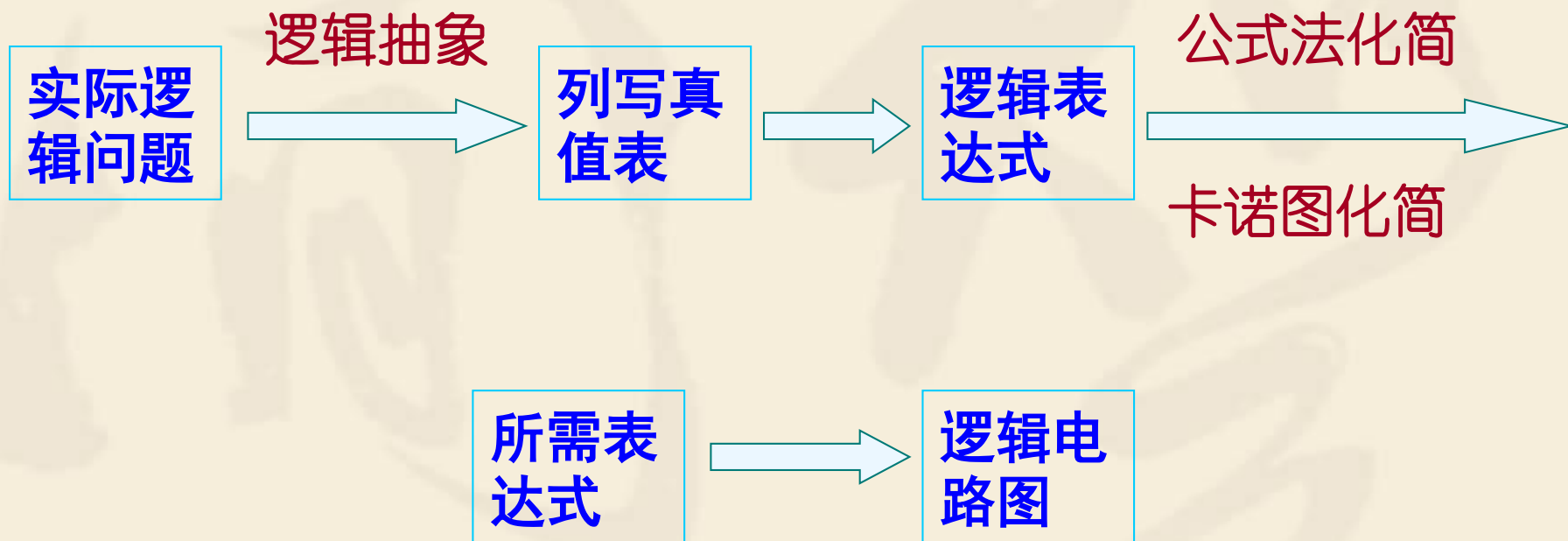
13.2 组合逻辑电路的设计

组合逻辑电路设计是其分析的逆过程

设计目的： 据给定逻辑功能及实现所用器件的要求，找出实现方案，并画出逻辑电路图

设计步骤：

- (1) 由实际问题进行逻辑抽象，列出真值表；
- (2) 由真值表列出逻辑表达式；
- (3) 简化和变换逻辑表达式；
- (4) 画出逻辑电路图逻辑功能。



若无要求，则化为最简逻辑式

若已要求逻辑门类型及个数，则视其要求而定

原则：最简（要求所用器件的种类和数量都尽可能少，且器件之间的连线也最少）。

例1：试用与非门和反相器设计一个优先排队电路。火车有特快、直快和慢车。它们进出站的优先次序是：特快、直快、慢车，同一时刻只能有一列车进出。

解：1) 由题意进行逻辑抽象。

输入变量及逻辑状态定义：

有无特快正准备进出站定义为变量A。A=1表示有特快正准备进出站，反之A=0

同理，将有无直快和慢车正准备进出站分别定义为变量B、C

输出变量及逻辑状态定义：

某时刻特快能否进出站定义为变量LA。LA=1表示特快能够进出站，反之LA=0

同理，将直快和慢车能否进出站分别定义为变量LB、LC

当特快A=1时，无论直快B、慢车C为何值， $L_A=1$ ， $L_B=L_C=0$ ；

当直快B=1，且A=0时，无论C为何值， $L_B=1$ ， $L_A=L_C=0$ ；

当慢车C=1，且A=B=0时， $L_C=1$ ， $L_A=L_B=0$ 。

经过逻辑抽象，可列真值表：

A	B	C	L_A	L_B	L_C
0	0	0	0	0	0
1	×	×	1	0	0
0	1	×	0	1	0
0	0	1	0	0	1

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

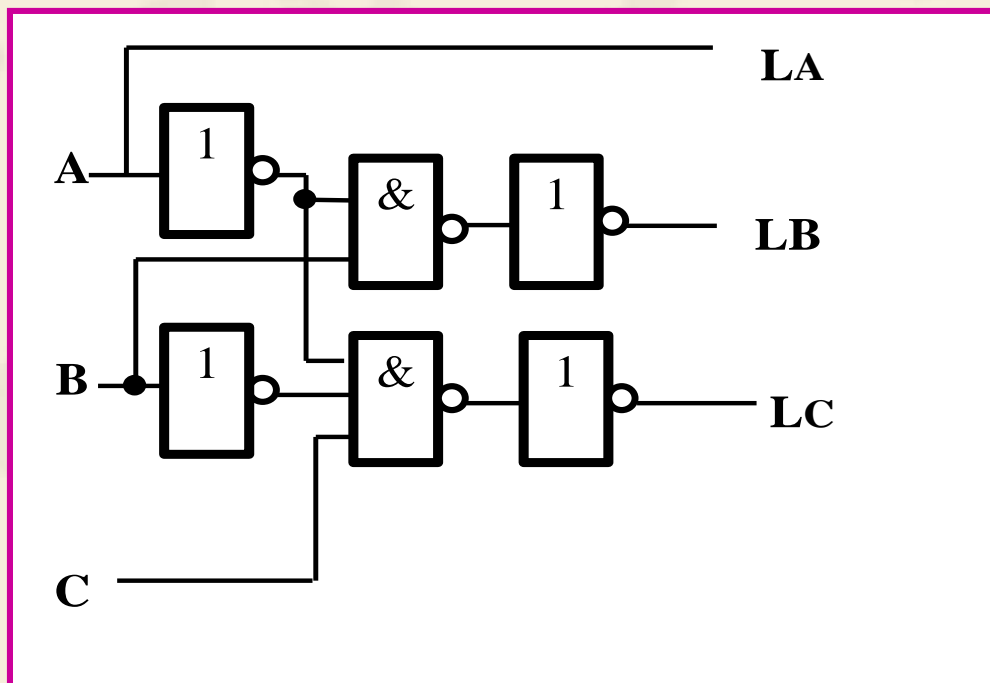
2) 写出逻辑表达式。

$$L_A = A, \quad L_B = \bar{A}B, \quad L_C = \bar{A}\bar{B}C$$

3) 根据题意，变换成与非形式

$$L_A = A, \quad L_B = \overline{\overline{A}B}, \quad L_C = \overline{\overline{A}BC}$$

4) 画出逻辑电路图



例2：某工厂有A、B、C三台设备，其中A和B的功率相等，C的功率是A的两倍。这些设备由X和Y两台发电机供电，发电机X的最大输出功率等于A的功率，发电机Y的最大输出功率是X的三倍。要求设计一个逻辑电路，能够根据各台设备的运转和停止状态，以最节约能源的方式启、停发电机。

解：1) 逻辑抽象。

设A、B、C为输入变量，分别表示A、B、C三台设备的运停状态。设备运转为1状态，设备停止为0状态，

X、Y为输出变量，分别表示X、Y两台发电机的启停。1表示发电机启动，0表示发电机停机

2)列真值表

3) 卡诺图化简,并写出X、 Y的逻辑表达式;

卡诺图 X

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

卡诺图 Y

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

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

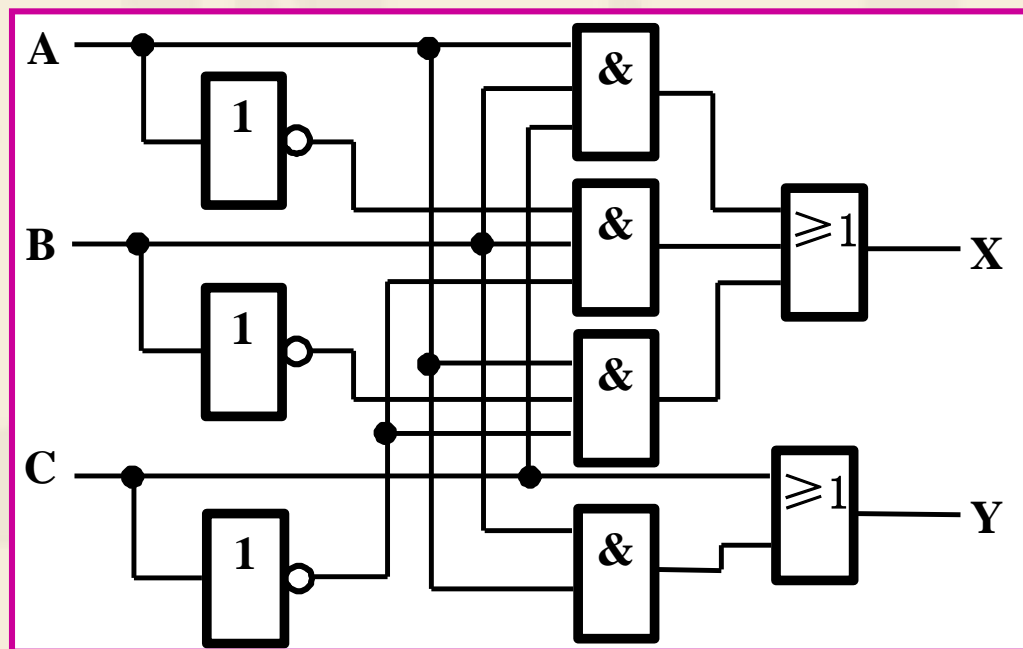
$$Y = AB + C$$

输 入			输 出	
A	B	C	X	Y
0	0	0	0	0
0	0	1	0	1
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

4) 画出逻辑图。

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

$$Y = AB + C$$



给定逻辑门后列写对应逻辑函数的方法：

要求用与非门实现



写原函数最简与或式

$$F_1 = AB + CD = \overline{\overline{AB} \cdot \overline{CD}}$$

要求用或非门实现



写原函数最简或与式

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

要求用与或非门实现



写反函数最简与或式

$$\overline{F_3} = AB + C \quad \text{---} \rightarrow \quad F_3 = \overline{AB + C}$$

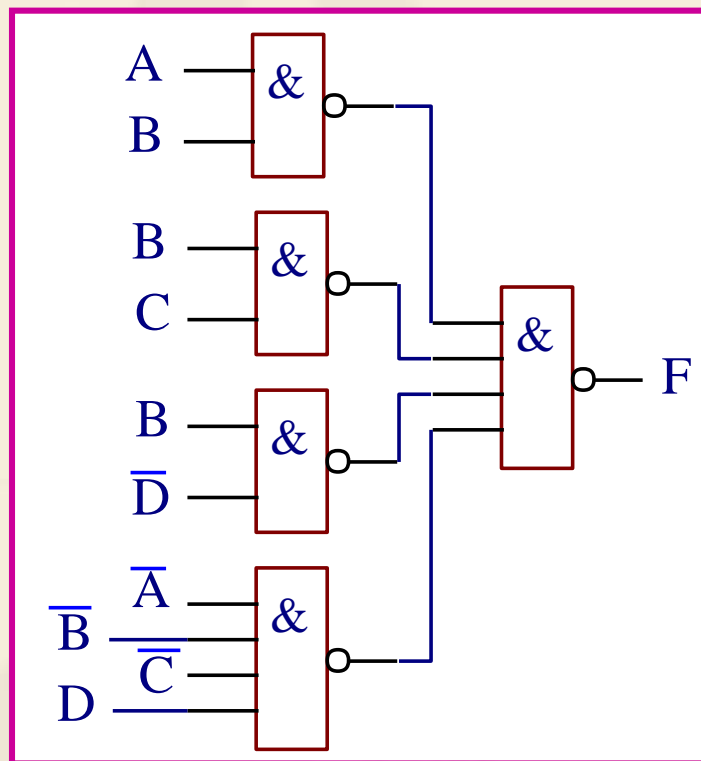
例3：用与非门实现函数 $F = AB + BC + BD + \overline{A}\overline{B}\overline{C}D$

(本题单变量的反变量可直接于逻辑图给出，不用经过反相器实现)

解：由于函数已是最简与或式，直接将F两次取反，得

$$F = \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{BD} \cdot \overline{\overline{A}\overline{B}\overline{C}D}}$$

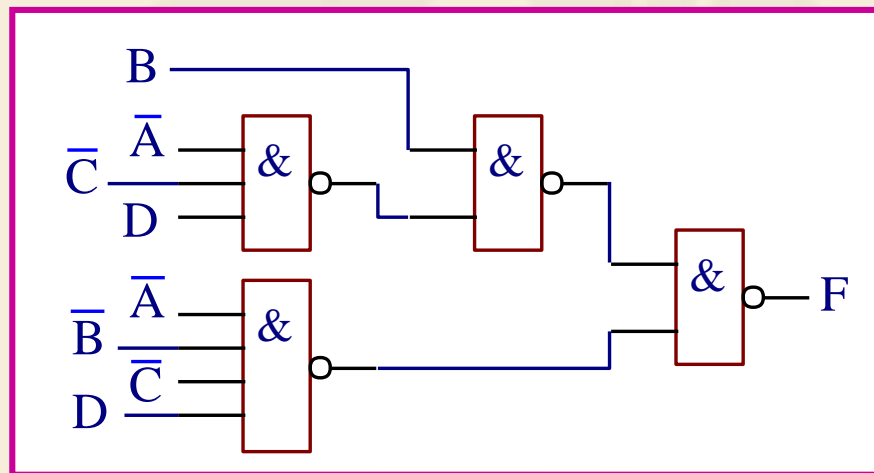
画逻辑电路图，如右图所示

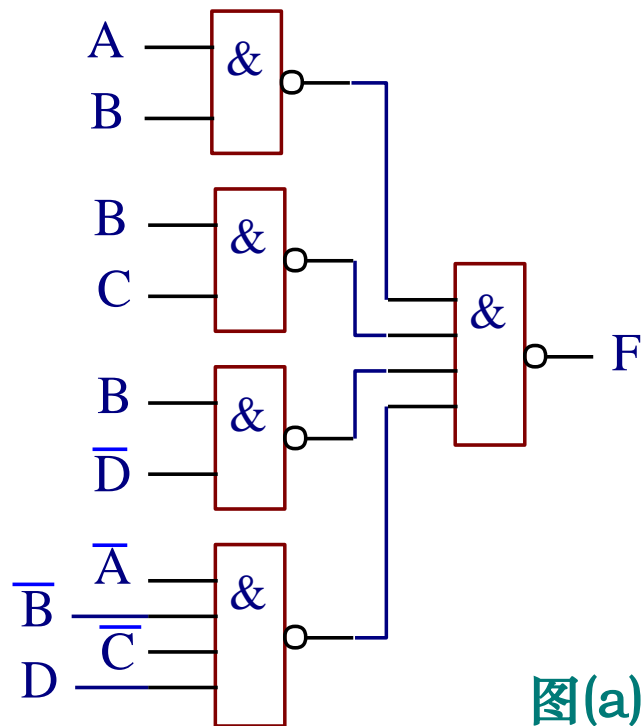


对函数还可做如下变换：

$$\begin{aligned} F &= AB + BC + B\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} \\ &= B(A + C + \bar{D}) + \bar{A}\bar{B}\bar{C}\bar{D} \\ &= \overline{\overline{B}\overline{A}\overline{C}\overline{D}} + \bar{A}\bar{B}\bar{C}\bar{D} \\ &= \overline{\overline{\overline{B}\overline{A}\overline{C}\overline{D}} \cdot \overline{\bar{A}\bar{B}\bar{C}\bar{D}}} \end{aligned}$$

相应的逻辑电路图，如右图所示

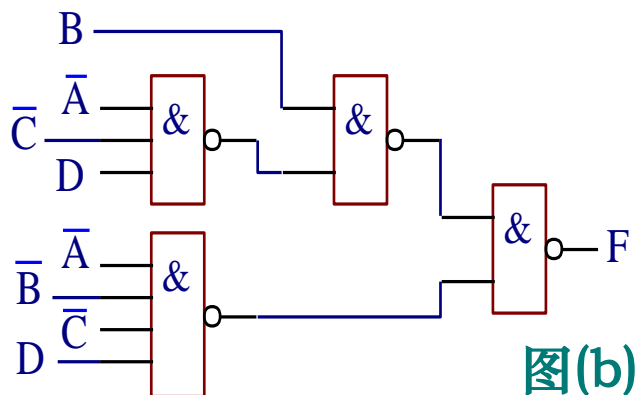




图(a)

图(a) 为二级5与非门，图(b) 为三级4与非门，显然图(b)中的门电路较少，此为其优点，但是其级数较多，这会使得电路工作速度变慢。

“门电路的数量最少”和“级数最少”通常相互矛盾。通常，题目不特别指明，即按“级数最少”解题；特别指明侧重前者，则按“门电路的数量最少”解题。



图(b)

13.3 加法器

13.3.1 半加器 (Half Adder)

不考虑低位来的进位，
只考虑两个加数 A_i 、 B_i 相加，
其和为 S_i ，向高位的进位为
 C_i 。计算过程如下：

$\begin{array}{r} 0 \\ +) 0 \\ \hline 0 \end{array}$	$\begin{array}{r} 0 \\ +) 1 \\ \hline 1 \end{array}$
$\begin{array}{r} 1 \\ +) 0 \\ \hline 1 \end{array}$	$\begin{array}{r} 1 \\ +) 1 \\ \hline 1 \end{array}$
$\begin{array}{r} 1 \\ +) 0 \\ \hline 1 \end{array}$	$\begin{array}{r} 1 \\ +) 1 \\ \hline 1 \end{array}$
$\begin{array}{r} 1 \\ +) 0 \\ \hline 1 \end{array}$	$\begin{array}{r} 1 \\ +) 1 \\ \hline 1 \end{array}$

真值表：

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

逻辑表达式：

$$S = A\bar{B} + \bar{A}B = A \oplus B$$

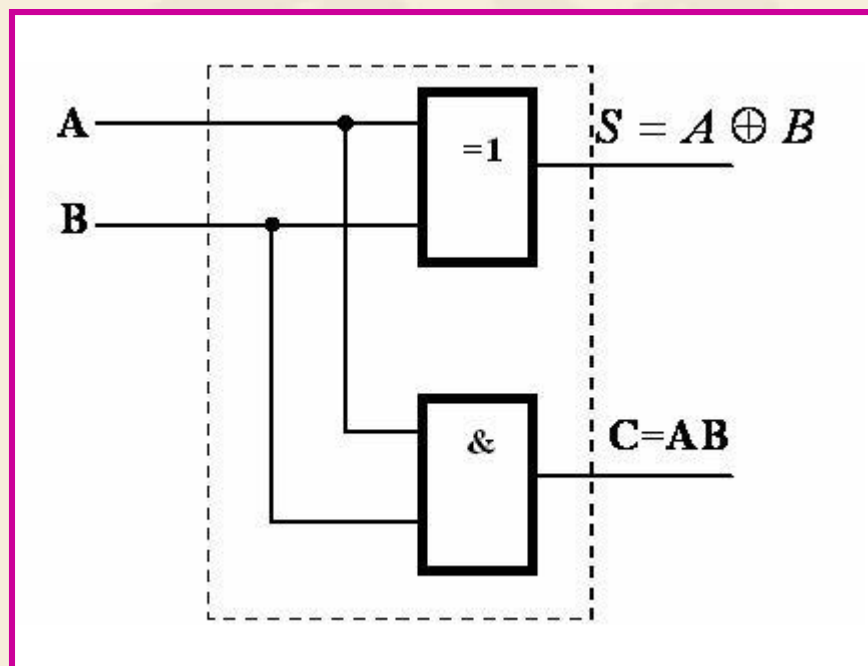
$$C = AB$$

进位 C_i

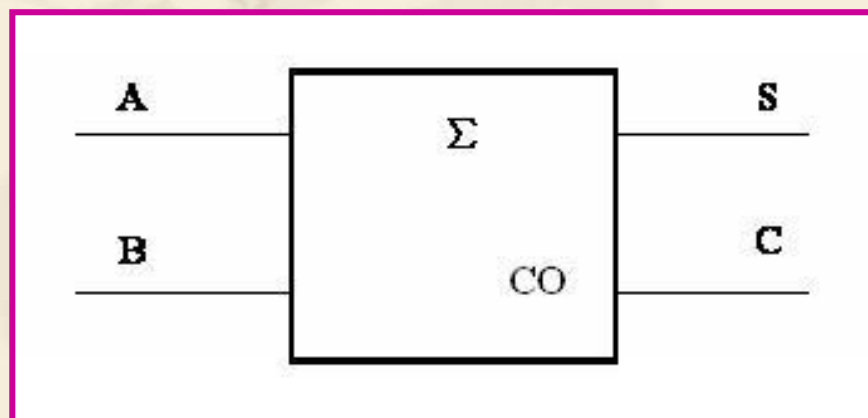
逻辑图

$$S = A\bar{B} + \bar{A}B = A \oplus B$$

$$C = AB$$



逻辑符号



13.3.2 全加器 (Full Adder)

全加器能进行加数、被加数和低位来的进位信号相加，并根据求和结果给出该位的进位信号。

真值表：

C_{i-1}	A_i	B_i	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

卡诺图：

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

		$A_i B_i$			
		00	01	11	10
C_i	0	0	0	1	0
	1	0	1	1	1

卡诺图化简：

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

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

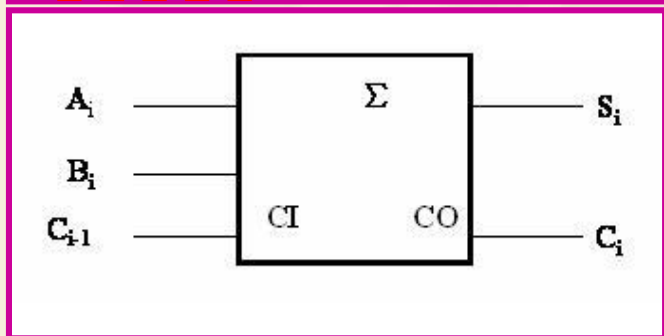
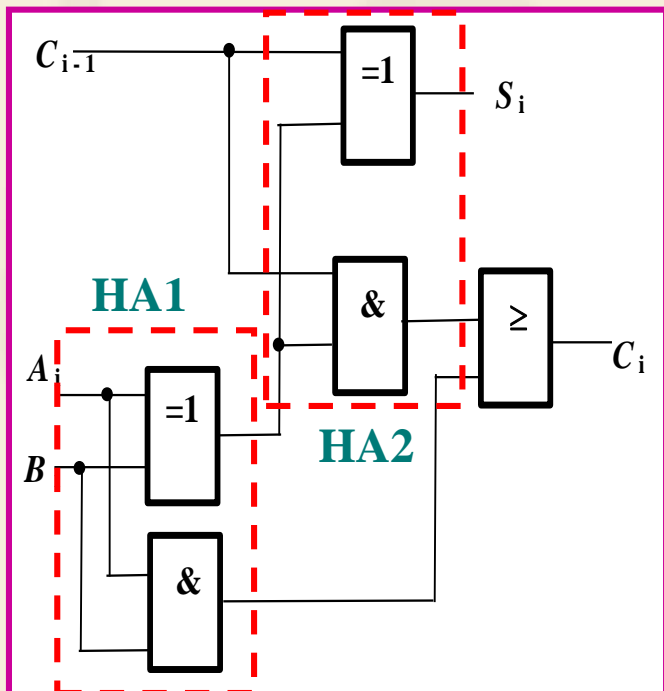
$$\begin{aligned}
 S_i &= \overline{C_{i-1}} \overline{A_i} B_i + \overline{C_{i-1}} A_i \overline{B_i} + C_{i-1} \overline{A_i} \overline{B_i} + C_{i-1} A_i B_i \\
 &= A_i \oplus B_i \oplus C_{i-1}
 \end{aligned}$$

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

逻辑图:

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

$$C_i = C_{i-1}(A_i \oplus B_i) + A_i B_i$$



该图表明:

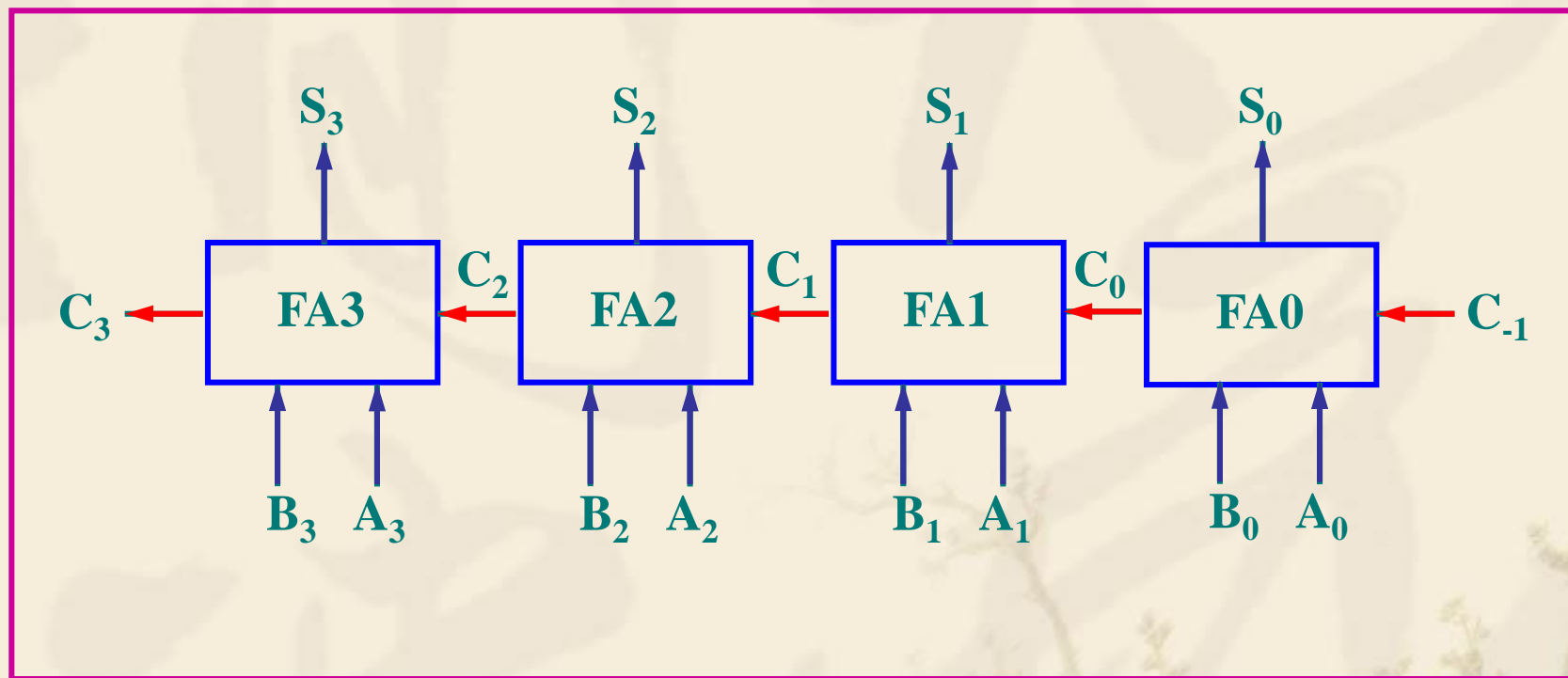
1、用两个半加器和一个或门可以实现全加器：先求两个加数的半加和，再与低位的进位作第二次半加，所得结果即全加器的和。

2、两个半加器的进位作逻辑加，即全加器的进位。

全加器的符号

13.3.3 串行进位加法器

构成：用多个1位全加器组成



- 低位的进位输出送至邻近高位的进位输入。属于并行相加，串行进位
- 缺点：运算速度慢。

13.3.4 超前进位加法器

$$C_i = C_{i-1} \underbrace{(A_i \oplus B_i)}_{P_i} + \underbrace{A_i B_i}_{G_i}$$



$$C_i = G_i + P_i C_{i-1}$$

$$S_i = P_i \oplus C_{i-1}$$

$$C_i = G_i + P_i C_{i-1}$$

进位信号的产生:

$$C_0 = G_0 + P_0 C_{-1}$$

$$C_1 = G_1 + P_1 C_0 = G_1 + P_1 G_0 + P_1 P_0 C_{-1}$$

$$C_2 = G_2 + P_2 C_1 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{-1}$$

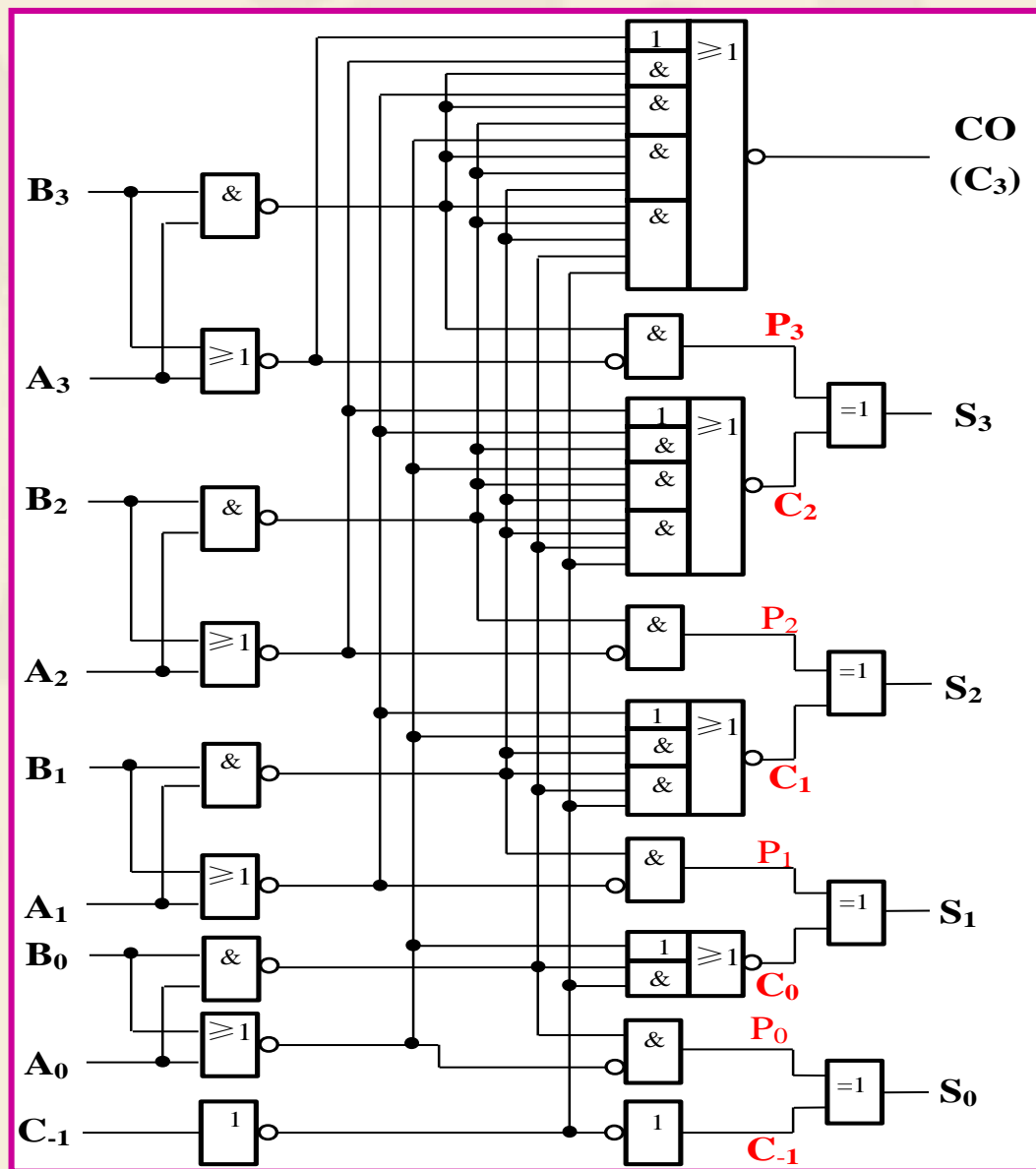
$$C_3 = G_3 + P_3 C_2 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{-1}$$

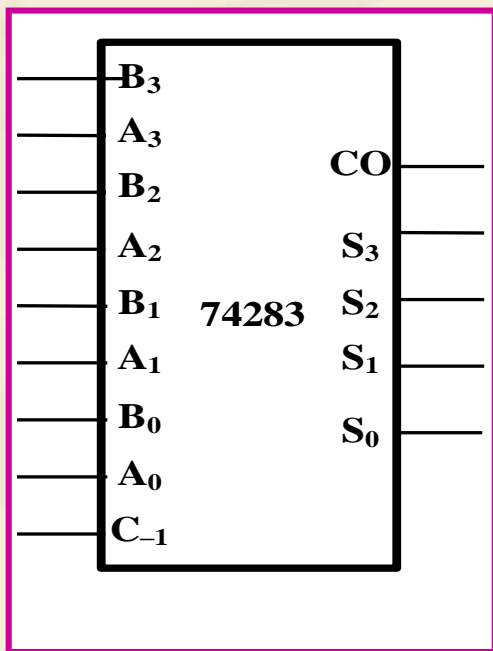
集成4位加法器 74LS283

逻辑图：

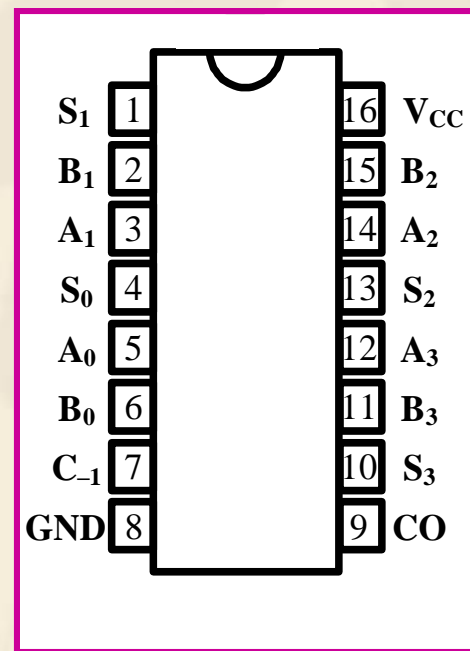
$$S_i = P_i \oplus C_{i-1}$$

$$C_i = G_i + P_i C_{i-1}$$





74LS283逻辑框图



74LS283引脚图

13.4 编码器

13.4.1 编码器的定义及功能

❖ **编码**：将若干个二进制码“0”和“1”，按某种规律编排组合成为代码，并赋予特定的含义，用以表示二进制以外其他的数或字母、符号、控制符，这就是编码。

❖ **编码器**：具有编码功能的逻辑电路。

❖ **编码器的特点**：能将某一组输入信息变换为二进制的代码输出。即对应输入的每一个状态，输出一个编码。(输入多→输出少)

❖ **编码器分为**：普通编码器和优先编码器。

❖ **普通编码器**：同一时刻只允许一个输入信号有效

如BCD编码器：将10个输入编成4位8421BCD码输出。

如4/2线编码器：将输入的4个状态编成2位二进制数码输出；

如8/3线编码器：将输入的8个状态编成3位二进制数码输出；

一般而言，N个不同的信号，至少需要n位二进制数编码。

N和n之间满足下列关系：

$$2^n \geq N$$

为此目的而设计的编码电路称为二进制编码器，详称为n位二进制编码器

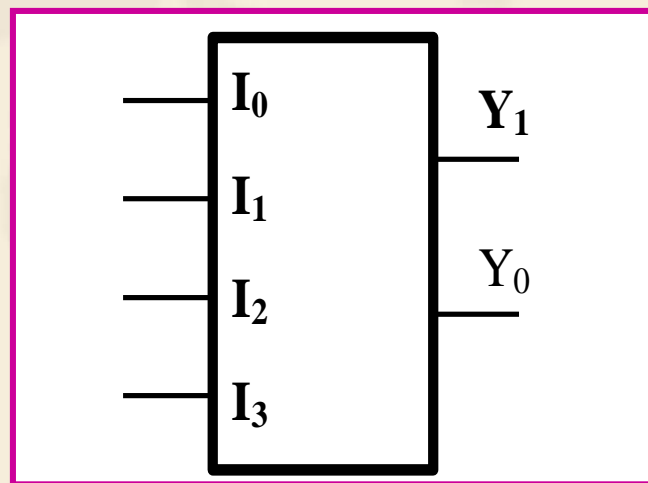
❖ **优先编码器**：允许两个以上的输入信号有效，但当同时输入几个有效信号时，优先编码器能按设定的优先级别，只对其中优先权最高的一个进行编码。

1. 普通4/2线编码器

4/2线编码器功能表

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

四输入



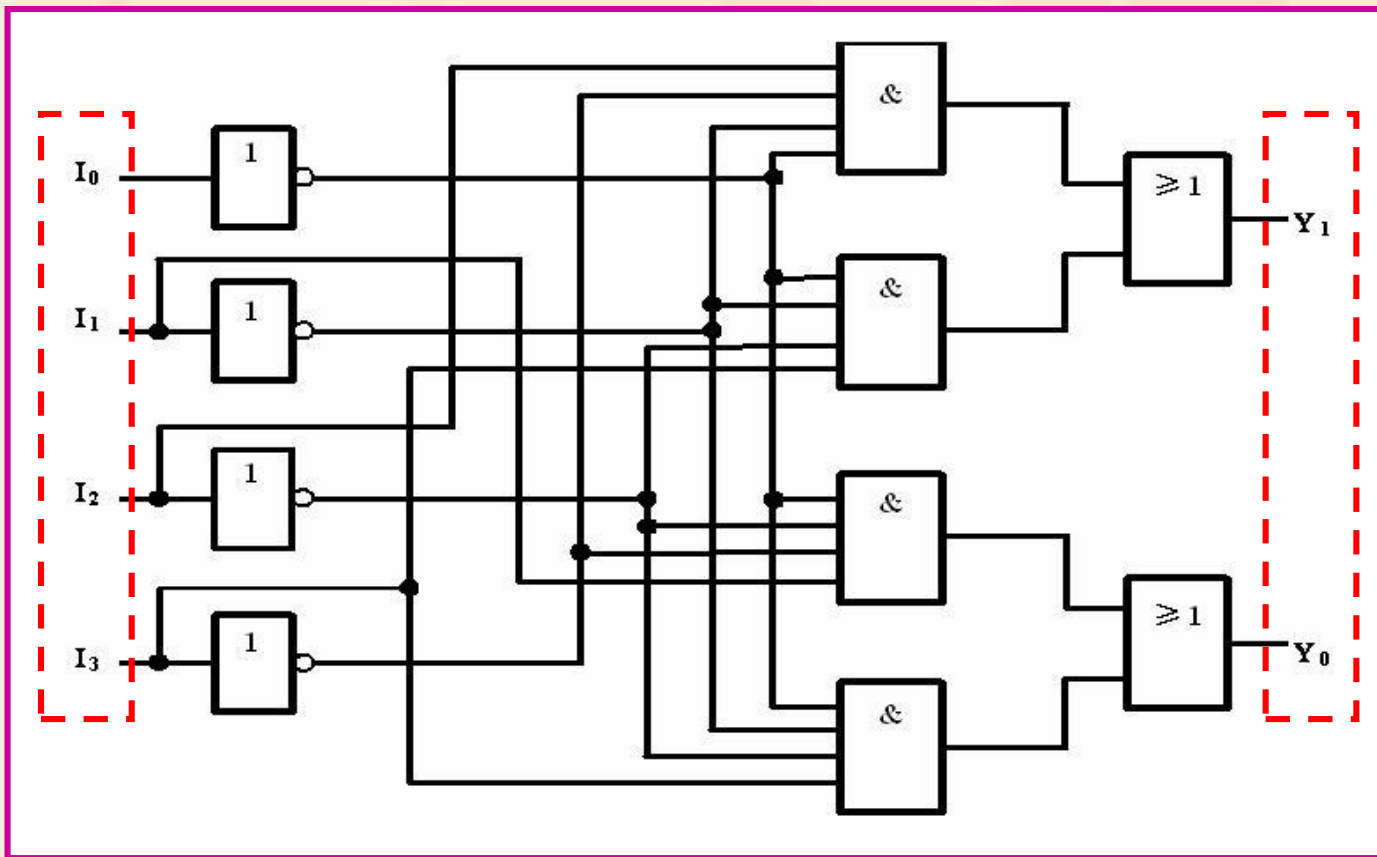
二进制码输出

编码器的输入为高电平有效。

$$Y_1 = \overline{I_0} \overline{I_1} I_2 \overline{I_3} + \overline{I_0} \overline{I_1} \overline{I_2} I_3$$

$$Y_0 = \overline{I_0} I_1 \overline{I_2} \overline{I_3} + \overline{I_0} \overline{I_1} \overline{I_2} I_3$$

逻辑图：



$$Y_1 = \overline{I_0} \overline{I_1} I_2 \overline{I_3} + \overline{I_0} \overline{I_1} I_2 I_3$$

$$Y_0 = \overline{I_0} I_1 \overline{I_2} \overline{I_3} + \overline{I_0} I_1 \overline{I_2} I_3$$

当所有的输入都为0时，电路的输出 $Y_1 Y_0 = ?$ 00

4/2线编码器功能表

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

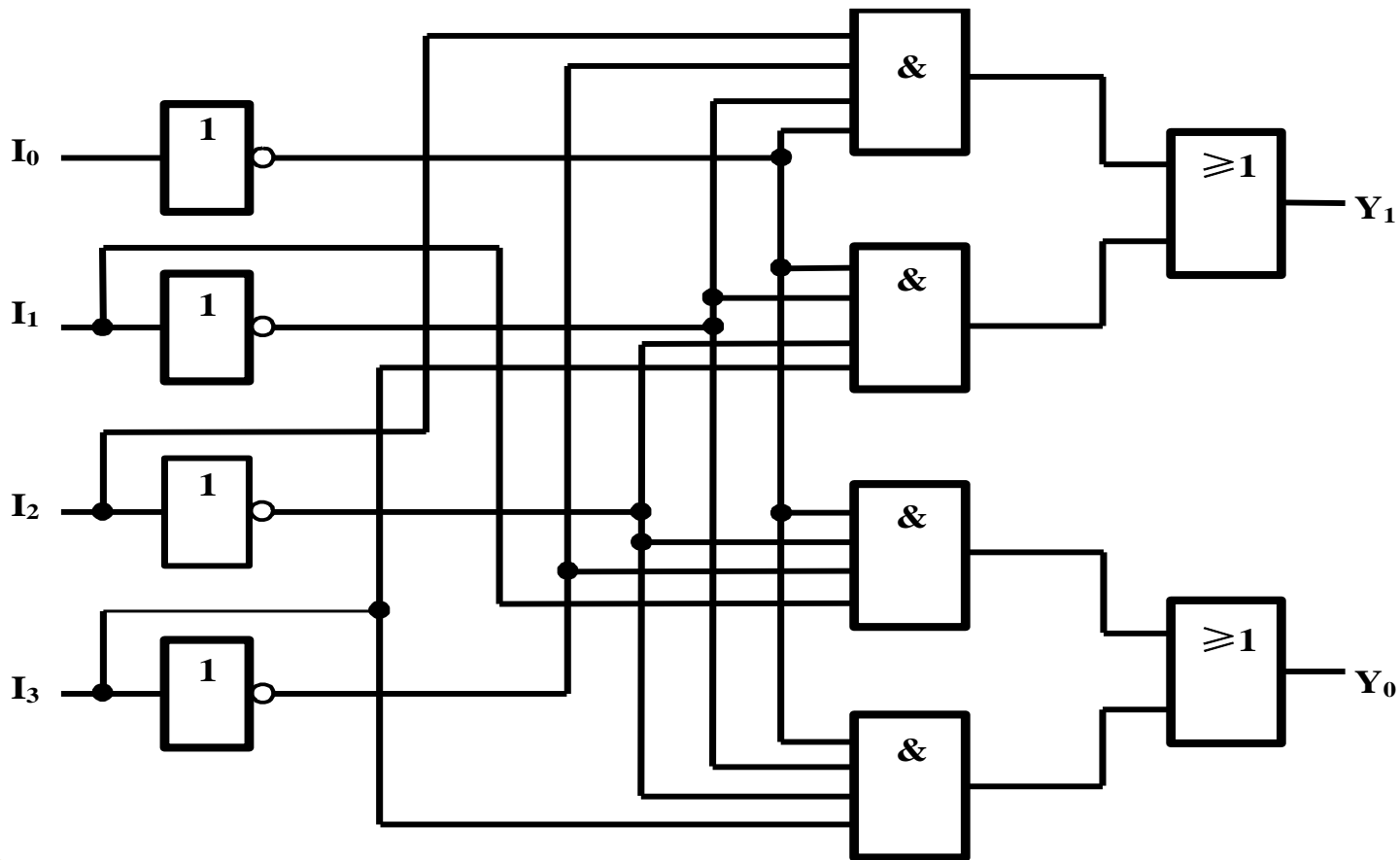
$I_0I_1I_2I_3=0000$ 时，电路的输出 $Y_1Y_0 = 00$

$I_0I_1I_2I_3=1000$ 时，电路的输出 $Y_1Y_0 = 00$

$$Y_1 = \overline{I_0}\overline{I_1}I_2\overline{I_3} + \overline{I_0}\overline{I_1}I_2I_3$$

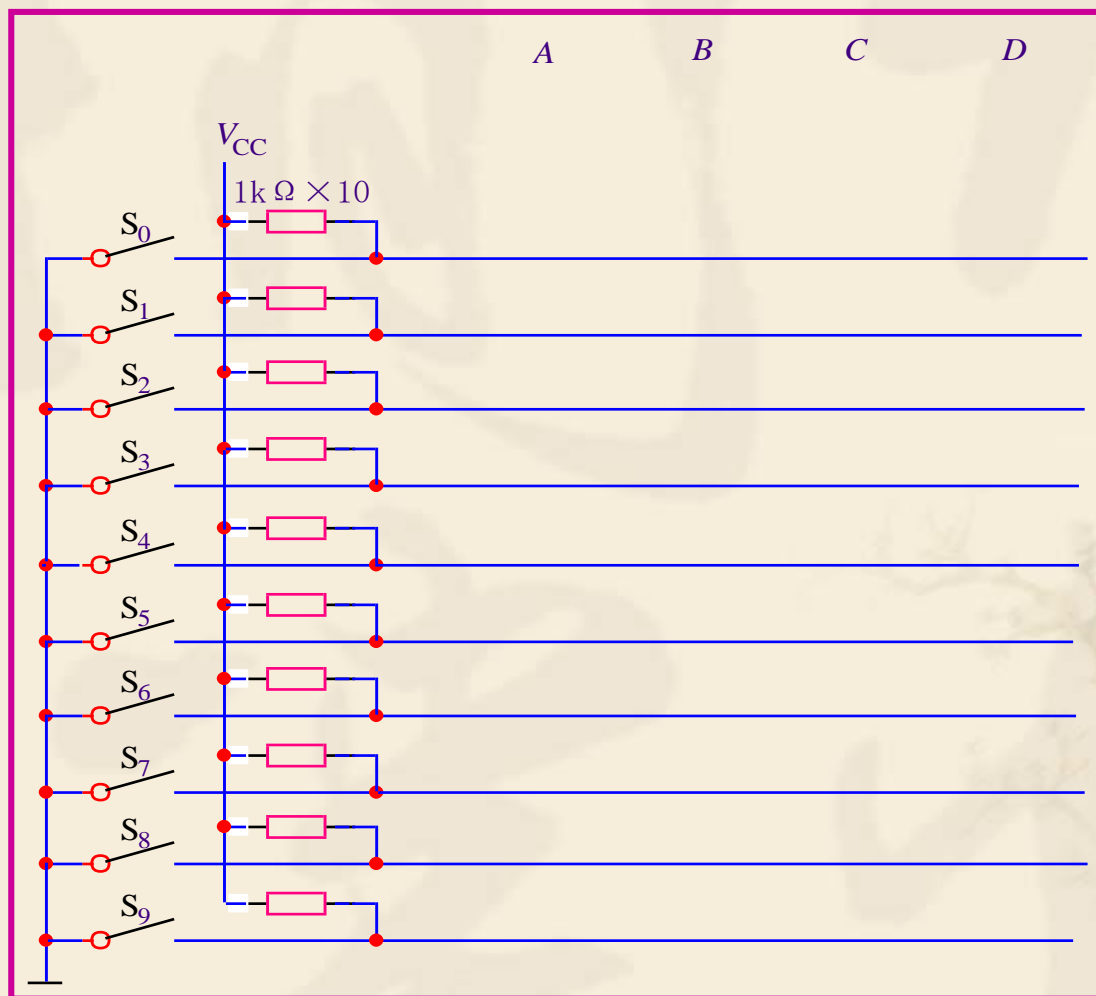
$$Y_0 = \overline{I_0}I_1\overline{I_2}\overline{I_3} + \overline{I_0}\overline{I_1}\overline{I_2}I_3$$

显然，无法区分上面两种情况，
本电路存在问题。



2. 键控8421BCD码编码器---10/4线编码器

设计一个键控8421BCD码编码器



解：（1）列出真值表：

输 入										输 出				
S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	A	B	C	D	GS
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	1	0	0	1	1
1	1	1	1	1	1	1	1	0	1	1	0	0	0	1
1	1	1	1	1	1	1	0	1	1	0	1	1	1	1
1	1	1	1	1	1	0	1	1	1	0	1	1	0	1
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	0	0	1
1	1	1	0	1	1	1	1	1	1	0	0	1	1	1
1	1	0	1	1	1	1	1	1	1	0	0	1	0	1
1	0	1	1	1	1	1	1	1	1	0	0	0	1	1
0	1	1	1	1	1	1	1	1	1	0	0	0	0	1

该编码器输入低电平有效

(2) 由真值表写出各输出的逻辑表达式为：

$$A = \overline{S_8} + \overline{S_9} = \overline{S_8 S_9} \quad B = \overline{S_4} + \overline{S_5} + \overline{S_6} + \overline{S_7} = \overline{S_4 S_5 S_6 S_7}$$

$$C = \overline{S_2} + \overline{S_3} + \overline{S_6} + \overline{S_7} = \overline{S_2 S_3 S_6 S_7}$$

$$D = \overline{S_1} + \overline{S_3} + \overline{S_5} + \overline{S_7} + \overline{S_9} = \overline{S_1 S_3 S_5 S_7 S_9}$$

重新整理得：

$$A = \overline{S_8 S_9}$$

$$B = \overline{S_4 S_5 S_6 S_7}$$

$$C = \overline{S_2 S_3 S_6 S_7}$$

$$D = \overline{S_1 S_3 S_5 S_7 S_9}$$

输 入										输 出				
S ₀	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇	S ₈	S ₉	A	B	C	D	GS
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	1	0	0	1	1
1	1	1	1	1	1	1	1	0	1	1	0	0	0	1
1	1	1	1	1	1	1	0	1	1	0	1	1	1	1
1	1	1	1	1	1	0	1	1	1	0	1	1	0	1
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	0	0	1
1	1	1	0	1	1	1	1	1	1	0	0	1	1	1
1	1	0	1	1	1	1	1	1	1	0	0	1	0	1
1	0	1	1	1	1	1	1	1	1	0	0	0	1	1
0	1	1	1	1	1	1	1	1	1	0	0	0	0	1

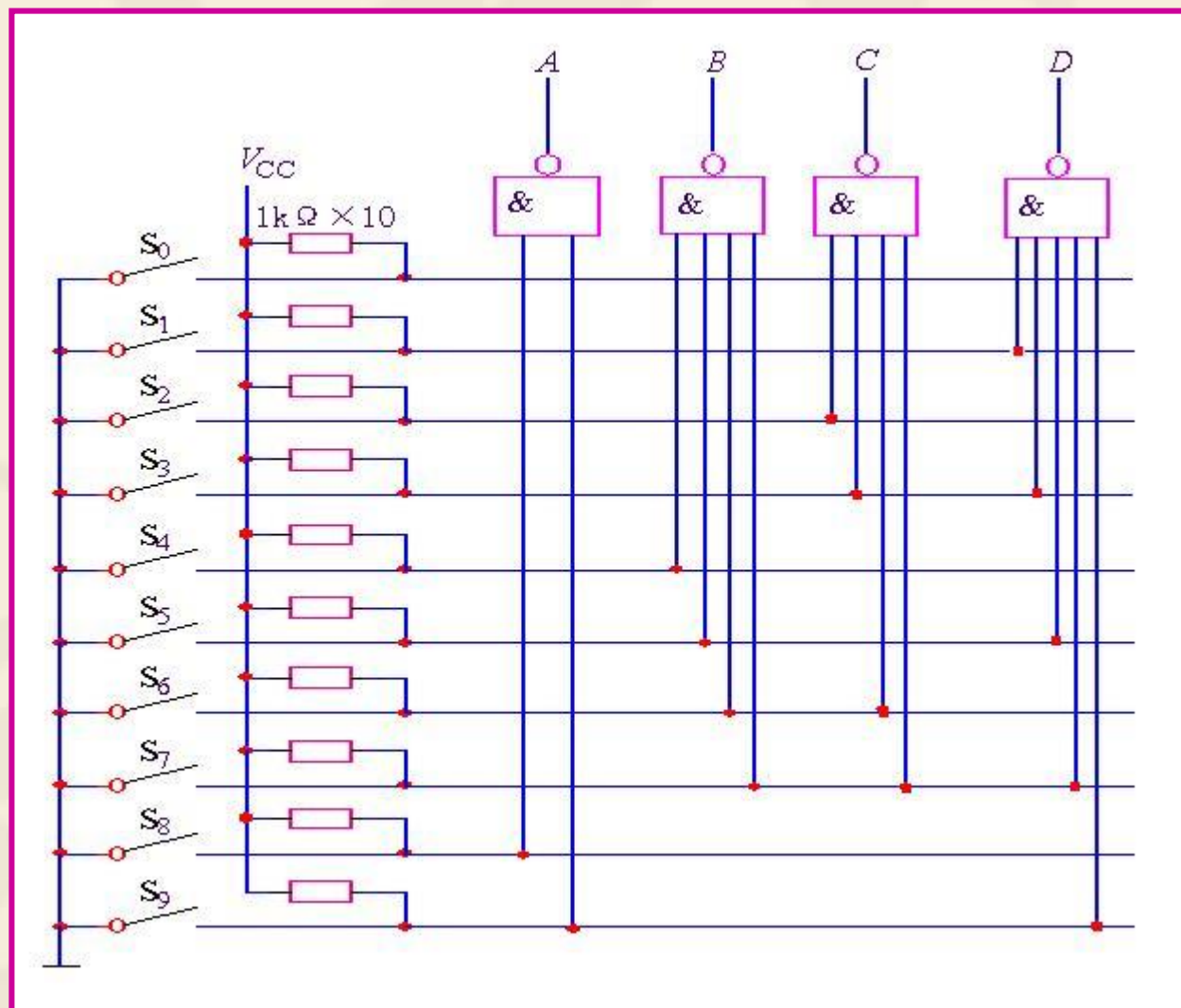
$$A = \overline{S_8 S_9}$$

$$B = \overline{S_4 S_5 S_6 S_7}$$

$$C = \overline{S_2 S_3 S_6 S_7}$$

$$D = \overline{S_1 S_3 S_5 S_7 S_9}$$

(3) 由表达式画出
逻辑图：



(4) 增加控制使能标志GS：

(1)当按下 $S_0 \sim S_9$

任意一个键时，

GS=1，表示有

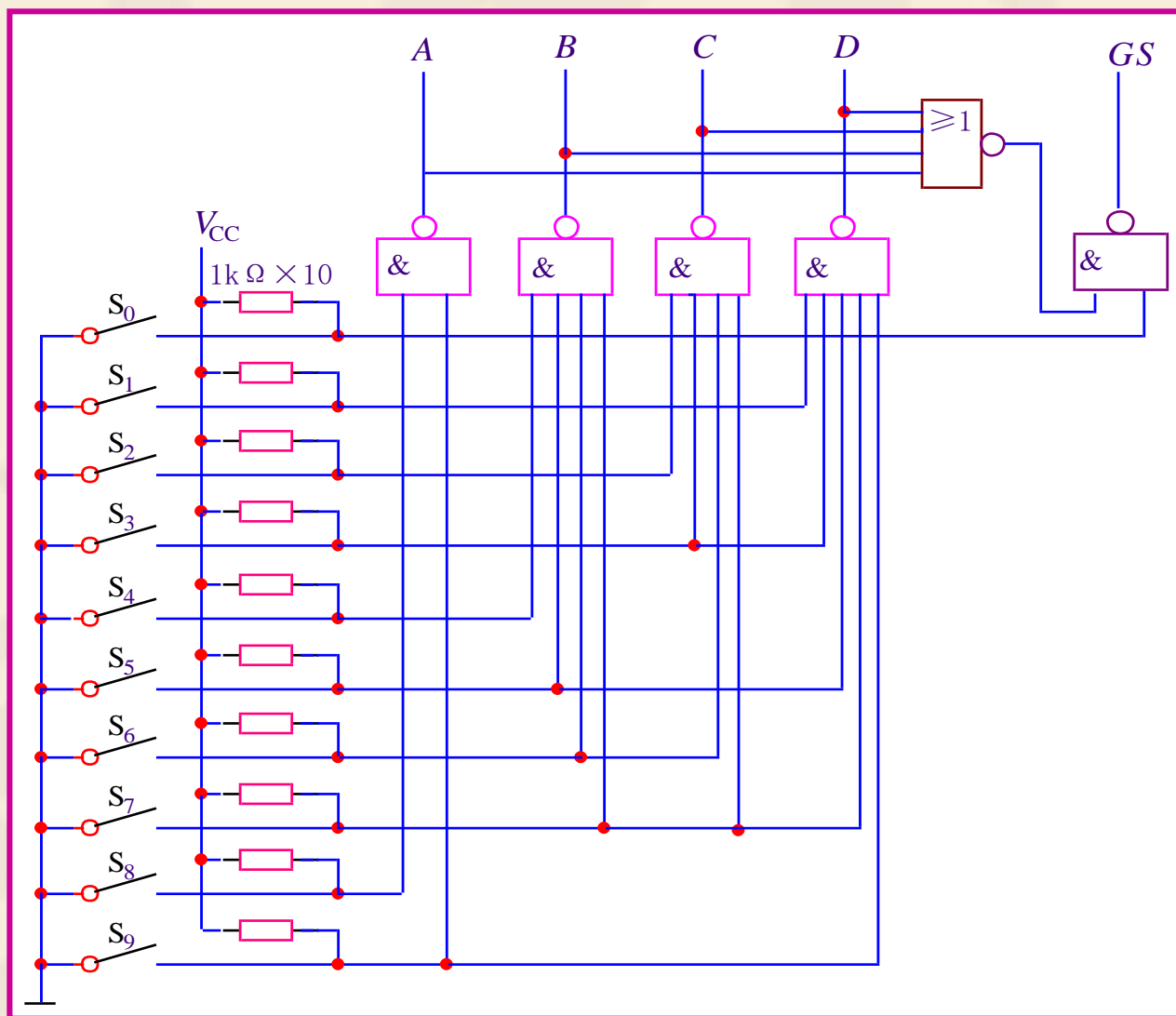
信号输入；

(2)当 $S_0 \sim S_9$ 均没

按下时，GS=0，

表示没有信号

输入。



13.5 译码器和数据分配器

13.5.1 译码器的定义及功能

❖ **译码**: 是编码的逆过程。将具有特定含义的不同二进制码识别出来, 并转换成控制信号。

❖ **译码器**: 具有译码功能的逻辑电路称为译码器。

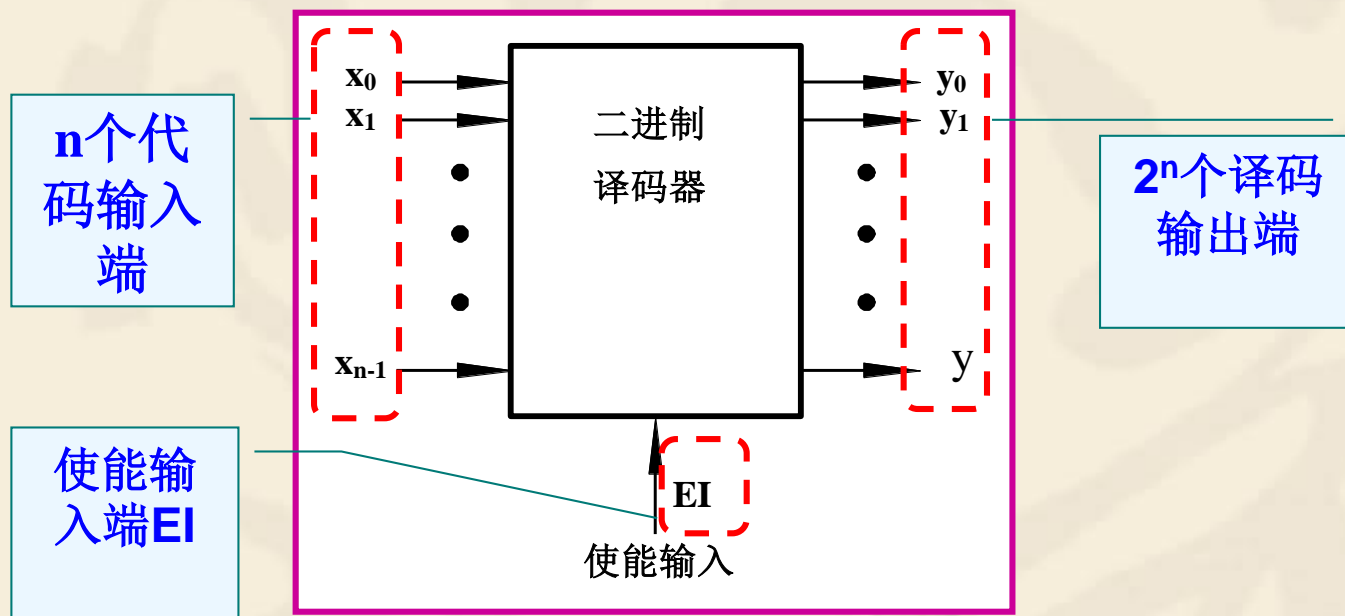
❖ **译码器分类**: 具有译码器按功能的不同可分为:

变量译码器 —— 表示输入变量状态, 2/4, 3/8, 4/16

码制变换译码器 —— BCD/十, 余3/十, 格雷码/十

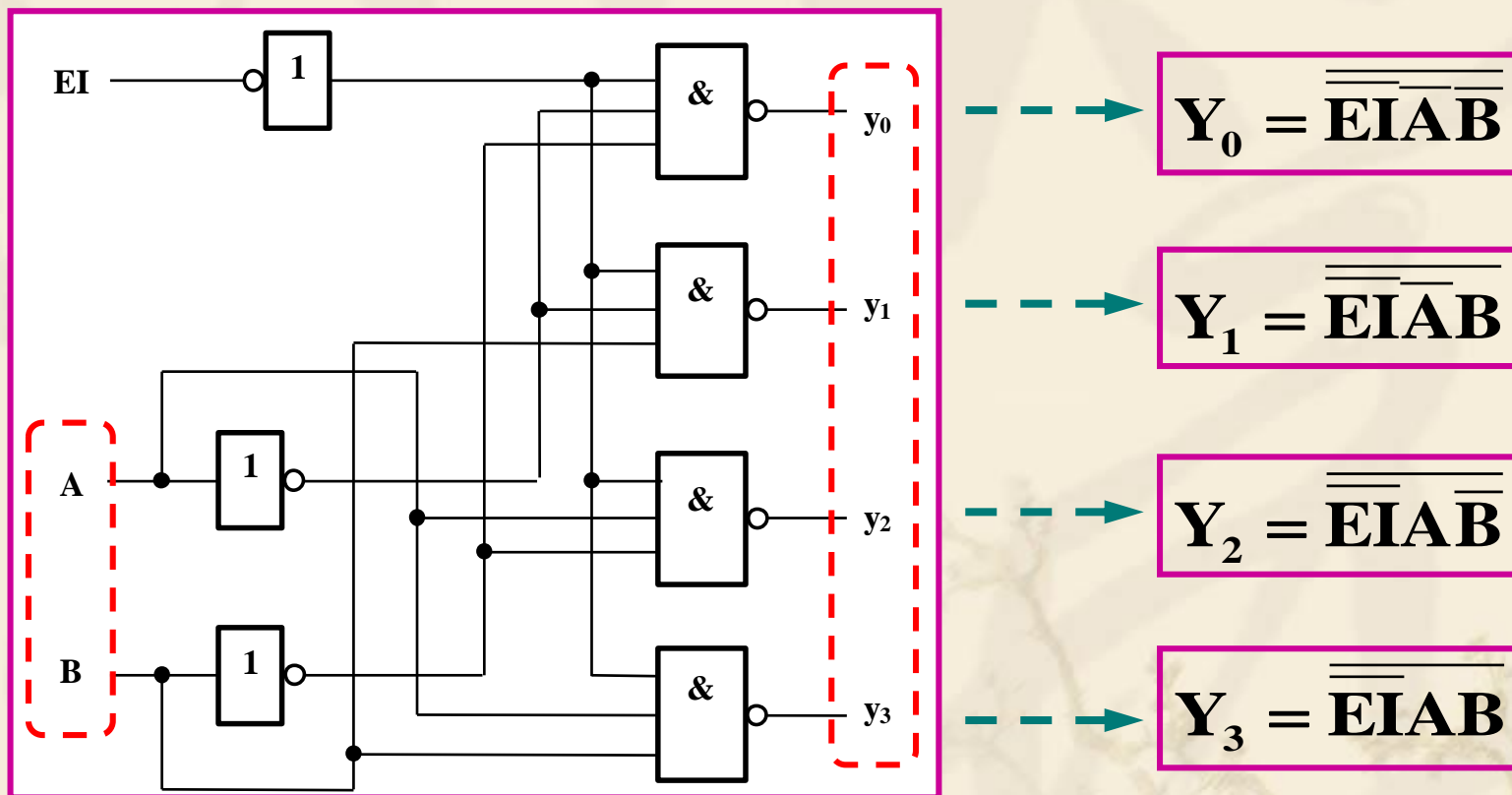
显示译码器 —— 驱动显示器件

1. 二进制译码器的一般原理



当使能端EI为有效电平时，对应每一组输入代码，只有其中一个输出端为有效电平(输入少 → 输出多)

2. 由门电路构成的2/ 4线译码器



列写功能表如下：

$$Y_0 = \overline{\overline{EIAB}}$$

$$Y_1 = \overline{\overline{EIAB}}$$

$$Y_2 = \overline{\overline{EIAB}}$$

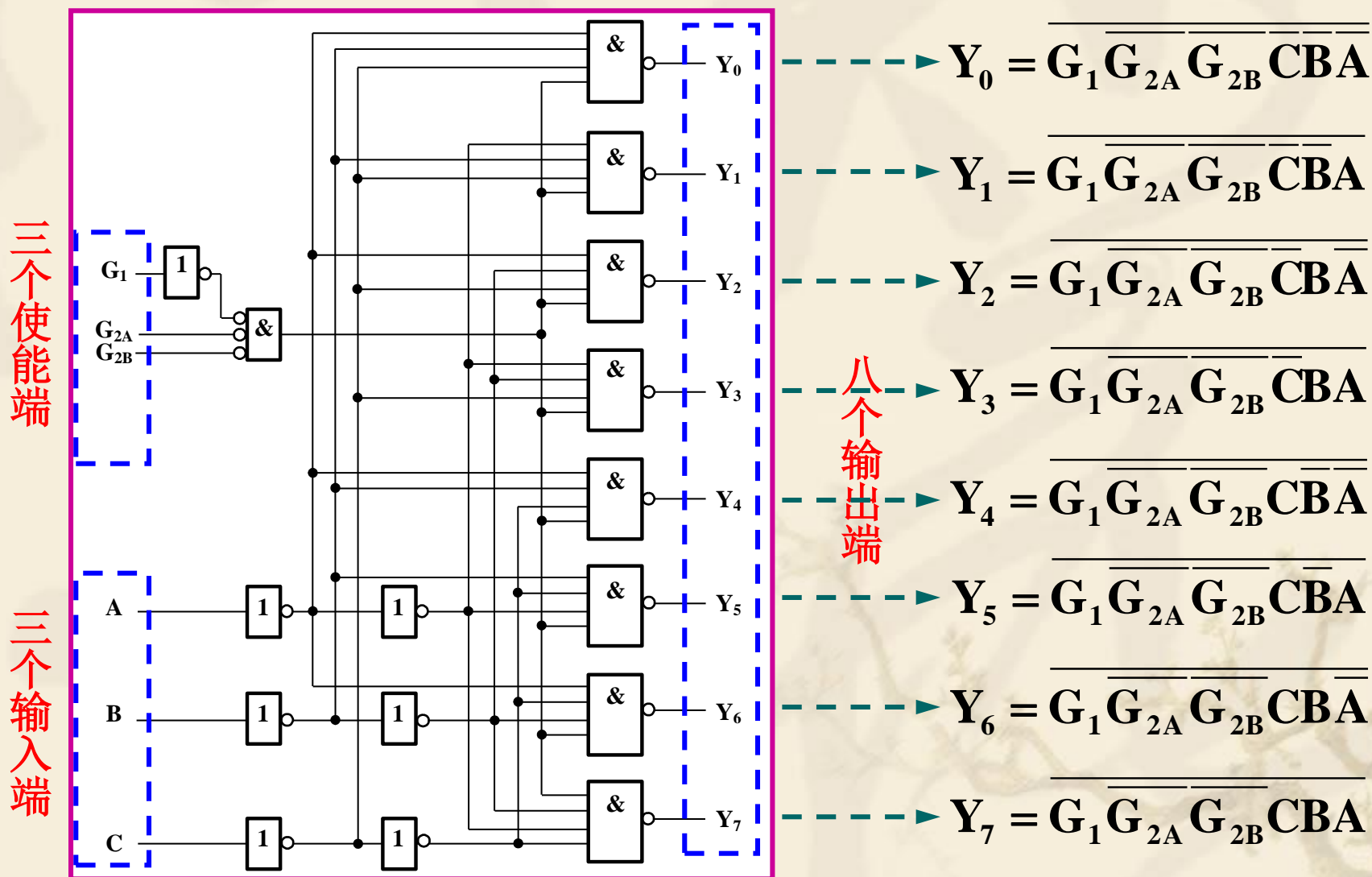
$$Y_3 = \overline{\overline{EIAB}}$$

输 入			输出			
EI	A	B	Y ₀	Y ₁	Y ₂	Y ₃
H	×	×	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

输 入			输出			
EI	A	B	Y ₀	Y ₁	Y ₂	Y ₃
1	×	×	H	H	H	H
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

13.5.2 集成译码器

——以74138-3/8线译码器为例



$$Y_0 = \overline{G_1} \overline{G_{2A}} \overline{G_{2B}} \overline{CBA}$$

$$Y_1 = \overline{G_1} \overline{G_{2A}} \overline{G_{2B}} CBA$$

$$Y_2 = \overline{G_1} \overline{G_{2A}} \overline{G_{2B}} C\overline{B}\overline{A}$$

$$Y_3 = \overline{G_1} \overline{G_{2A}} \overline{G_{2B}} CBA$$

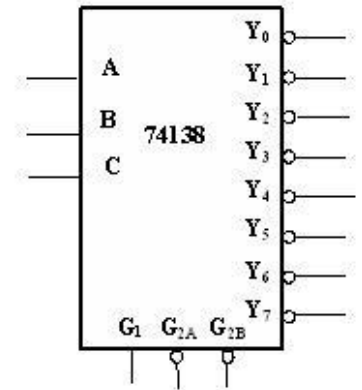
$$Y_4 = \overline{G_1} \overline{G_{2A}} \overline{G_{2B}} C\overline{B}\overline{A}$$

$$Y_5 = \overline{G_1} \overline{G_{2A}} \overline{G_{2B}} CBA$$

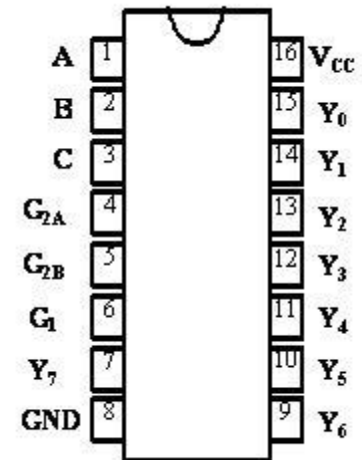
$$Y_6 = \overline{G_1} \overline{G_{2A}} \overline{G_{2B}} C\overline{B}\overline{A}$$

$$Y_7 = \overline{G_1} \overline{G_{2A}} \overline{G_{2B}} CBA$$

$$Y_i = \overline{G_1} \overline{G_{2A}} \overline{G_{2B}} m_i$$



74138逻辑符号



74138芯片引脚

根据功能表可得知74138芯片的用法:习惯上,常用 $A_2A_1A_0$ 表示最高、次高、最低位,

74138集成译码器功能表													
输 入			输 出										
G_1	G_{2A}	G_{2B}	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
×	H	×	×	×	×	H	H	H	H	H	H	H	H
×	X	H	×	×	×	H	H	H	H	H	H	H	H
L	×	×	×	×	×	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

故此可用 $A_2A_1A_0$ 分别表示CBA

此处使能端控制使芯片处于非正常工作状态

$Y_0 = \overline{A_2} \overline{A_1} \overline{A_0} = \overline{m_0}$
 $Y_1 = \overline{A_2} \overline{A_1} A_0 = \overline{m_1}$
 $Y_2 = \overline{A_2} A_1 \overline{A_0} = \overline{m_2}$
 $Y_3 = \overline{A_2} A_1 A_0 = \overline{m_3}$
 $Y_4 = A_2 \overline{A_1} \overline{A_0} = \overline{m_4}$
 $Y_5 = A_2 \overline{A_1} A_0 = \overline{m_5}$
 $Y_6 = A_2 A_1 \overline{A_0} = \overline{m_6}$
 $Y_7 = A_2 A_1 A_0 = \overline{m_7}$

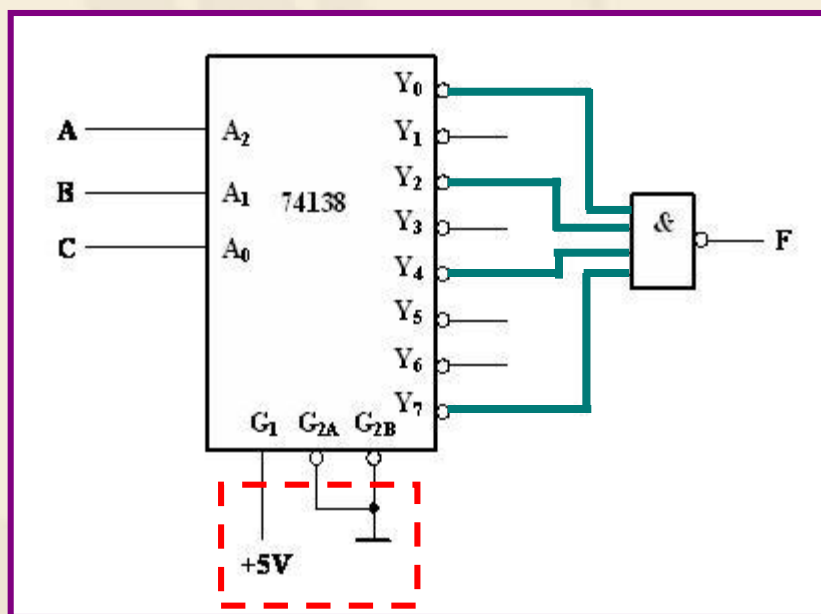
3/8线译码器能产生三变量函数的全部最小项。

基于这一点用该器件能够方便地实现三变量逻辑函数。

例1：用3/8线译码器实现逻辑函数 $F = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC$

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

$$= \overline{Y_0 \cdot Y_2 \cdot Y_4 \cdot Y_7}$$



$$Y_0 = \overline{A_2}\overline{A_1}\overline{A_0}$$

$$Y_2 = \overline{A_2}A_1\overline{A_0}$$

$$Y_4 = A_2\overline{A_1}\overline{A_0}$$

$$Y_7 = A_2A_1A_0$$

74138工作条件：
 $G_1=1$ ， $G_{2A}=G_{2B}=0$

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

用译码器实现逻辑函数时可不用化简，直接由最小项表达。

例2: 用74138实现函数 $F = AB + A\bar{C}$

解: $F(A,B,C) = AB + A\bar{C} = m_4 + m_6 + m_7$

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

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

$$= \overline{Y_4 \cdot Y_6 \cdot Y_7}$$

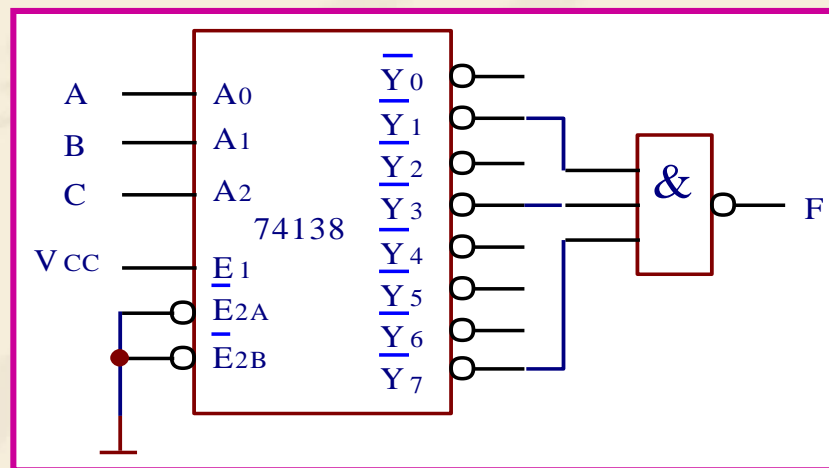
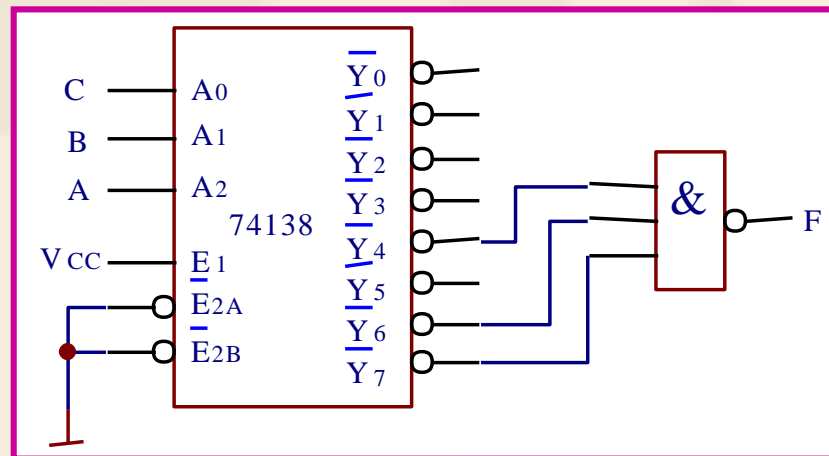
另:

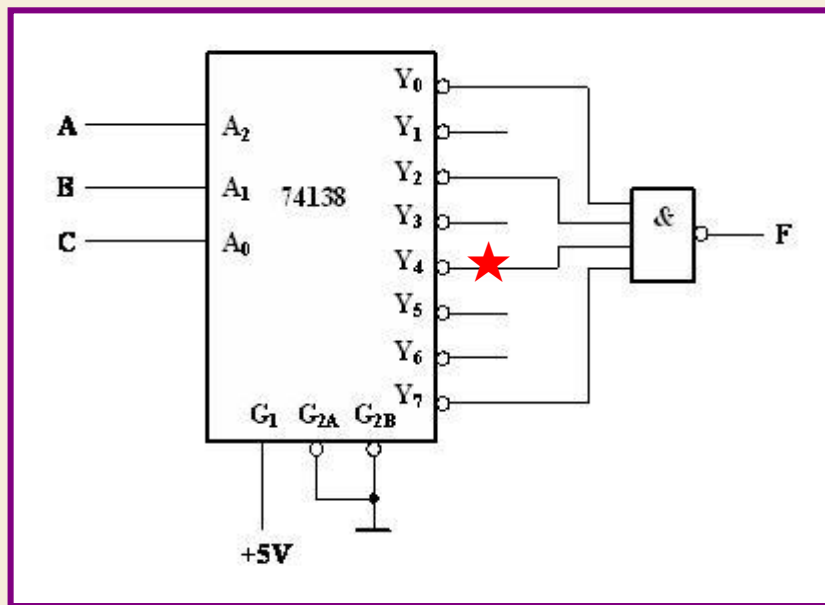
$F(C,B,A) = AB + A\bar{C} = m_1 + m_3 + m_7$

$$= \overline{\overline{m_1 + m_3 + m_7}}$$

$$= \overline{\overline{m_1} \cdot \overline{m_3} \cdot \overline{m_7}}$$

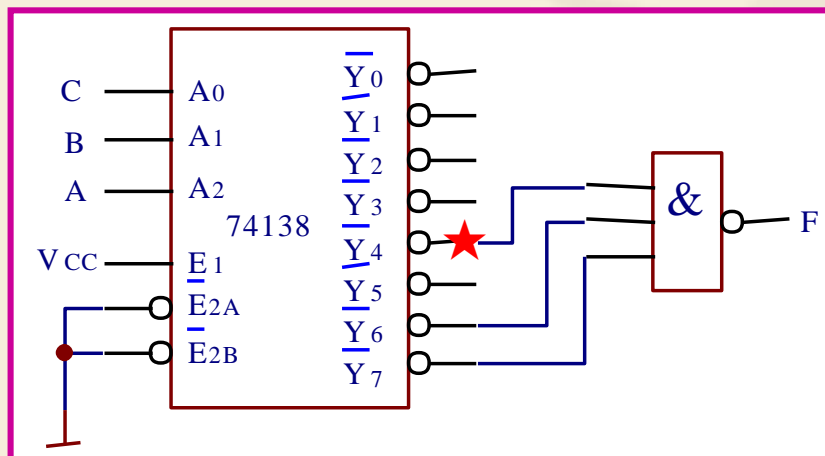
$$= \overline{Y_1 \cdot Y_3 \cdot Y_7}$$





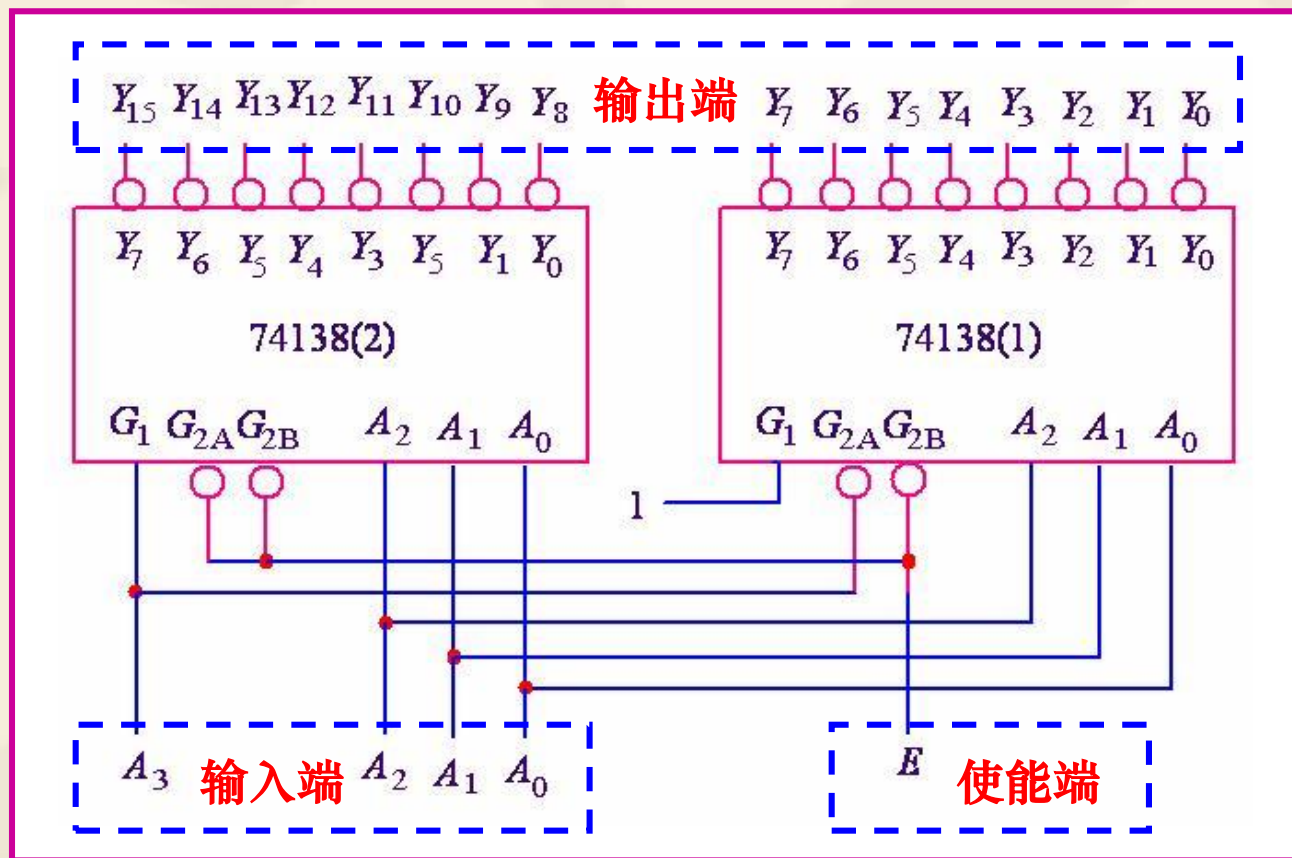
$$Y_4 = m_4 = \overline{A_2} \overline{A_1} \overline{A_0}$$

★处相同都为 $\overline{A_2} \overline{A_1} \overline{A_0}$



$$Y_4 = m_4 = A_2 \overline{A_1} \overline{A_0}$$

例3：用两片74138扩展为4线—16线译码器

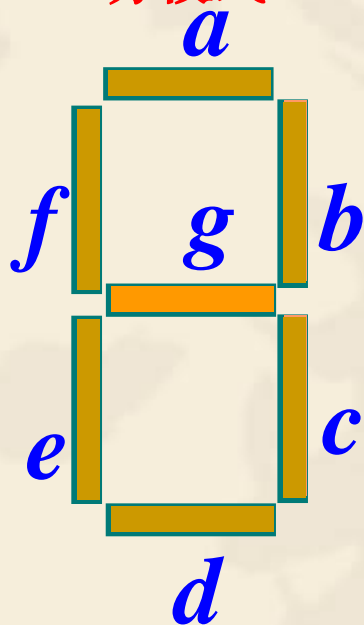


连接好三个部分：输出、输入、使能

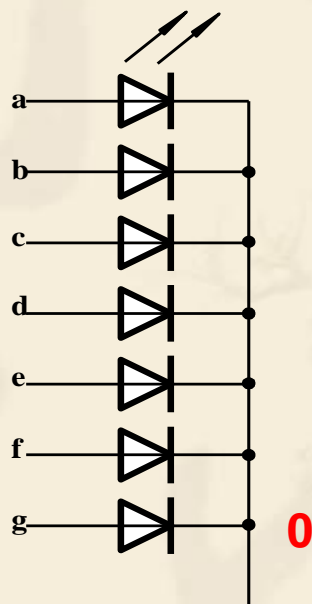
13.5.3 显示器和译码器

显示器主要分为两类：

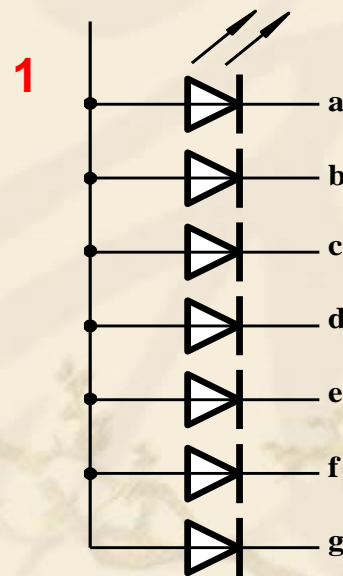
- (1) 按发光物质不同，分为“发光二极管显示器”、“荧光数字显示器”、“液晶显示器”和“气体放电显示器”四种。
- (2) 按字形显示方式不同，分为“字型重叠式”、“点阵式”和“分段式”三种。



七段显示器



共阴七段显示器



共阳七段显示器

a b c d e f g

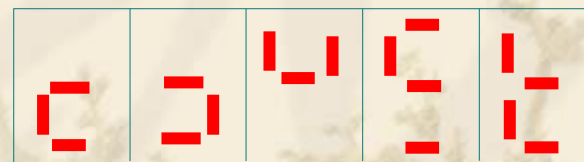
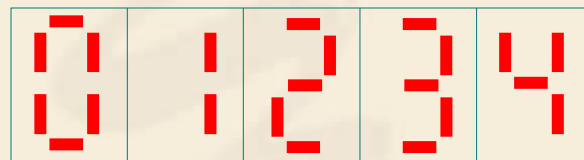
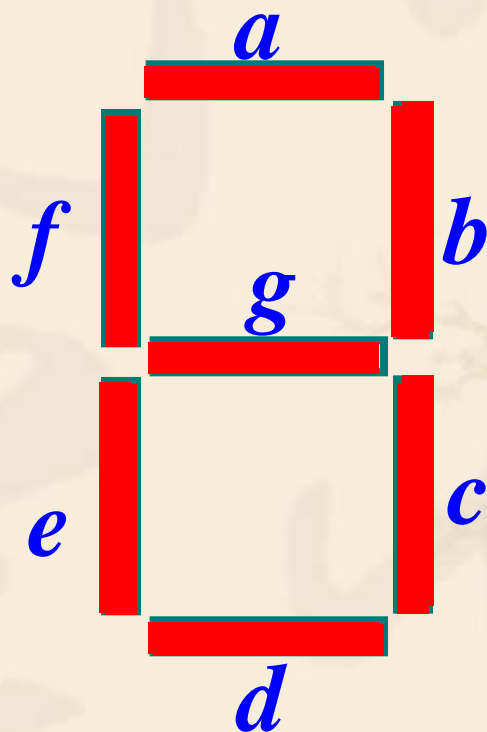
1 1 1 1 1 1 0

0 1 1 0 0 0 0

1 1 0 1 1 0 1

⋮

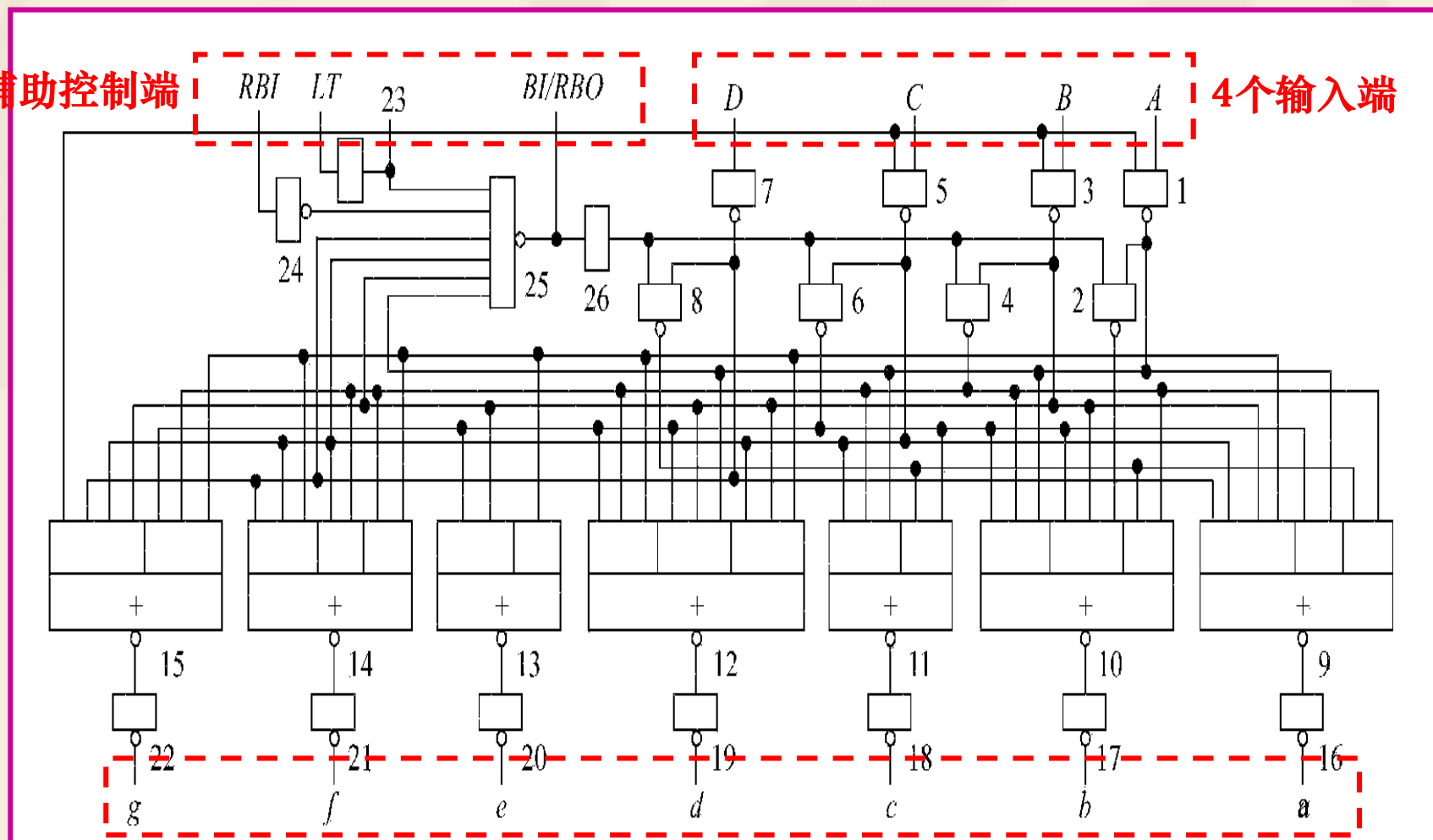
共阴极显示器



译码器7448（输出为高电平，驱动共阴显示器）

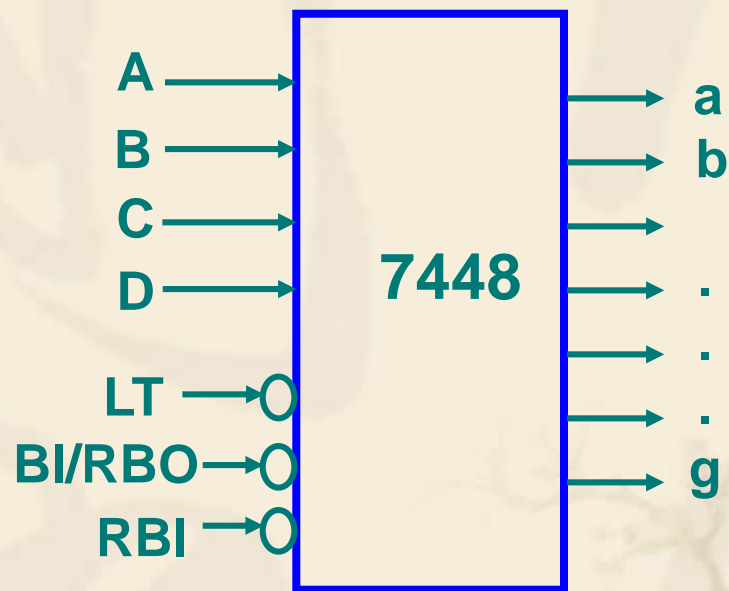
3个辅助控制端

4个输入端








7个输出端

译码器7448



7448功能框图

BCD-七段显示译码器7448的逻辑功能

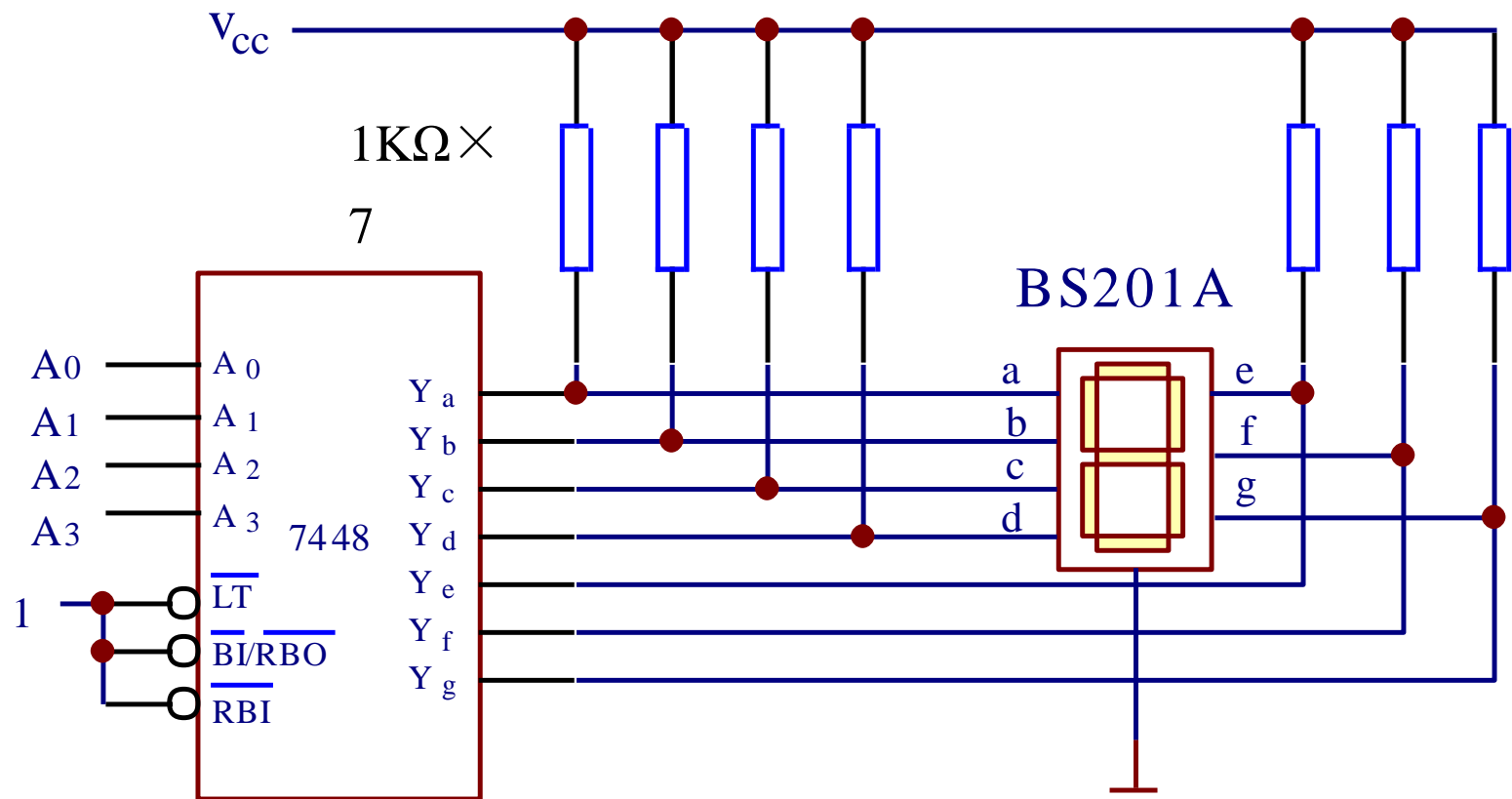
十进制 或功能	输 入							输出							字 形
	LT	RBI	D	C	B	A	BI/ RBO	a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	H	H	H	H	H	H	L	
1	H	×	L	L	L	H	H	L	H	H	L	L	L	L	
2	H	×	L	L	H	L	H	H	H	L	H	H	L	H	
3	H	×	L	L	H	H	H	H	H	H	H	L	L	H	
15	H	×	H	H	H	H	H	L	L	L	L	L	L	L	
消 隐	×	×	×	×	×	×	L	L	L	L	L	L	L	L	
脉冲消隐	H	L	L	L	L	L	L	L	L	L	L	L	L	L	
灯 测 试	L	×	×	×	×	×	H	H	H	H	H	H	H	H	

译码器7448

•BCD-七段显示译码器74148的逻辑功能

功能	输 入							输出							字形
	LT	RBI	D	C	B	A	BI/ RBO	a	b	c	d	e	f	g	
消 隐	×	×	×	×	×	×	L	L	L	L	L	L	L	L	
脉冲消隐	H	L	L	L	L	L	L	L	L	L	L	L	L	L	
灯 测 试	L	×	×	×	×	×	H	H	H	H	H	H	H	H	8

试灯输入LT： 当LT=0时（BI/RBO是输出端，且RBO=1），无论其他输入端是什么状态，所有段输出均为1，显示字形8。用于检查译码器和显示器的好坏。



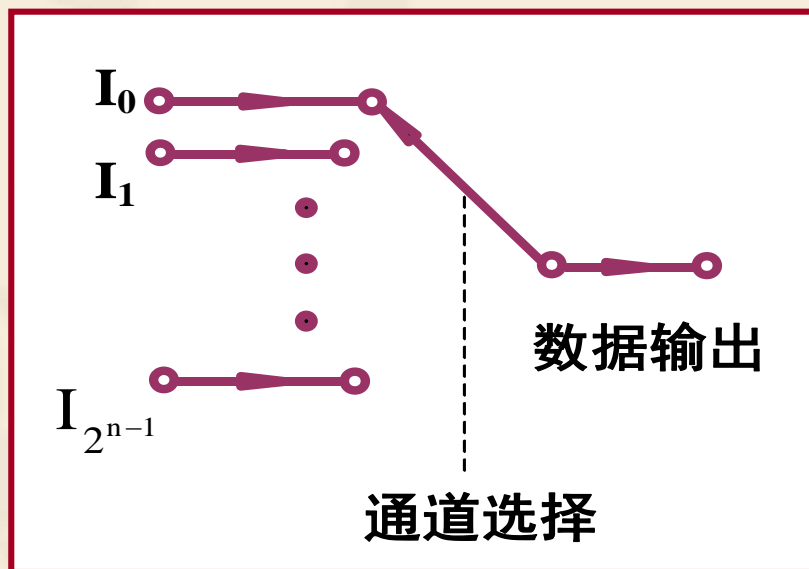
用7448驱动BS201A的连接方法

13.6 数据选择器

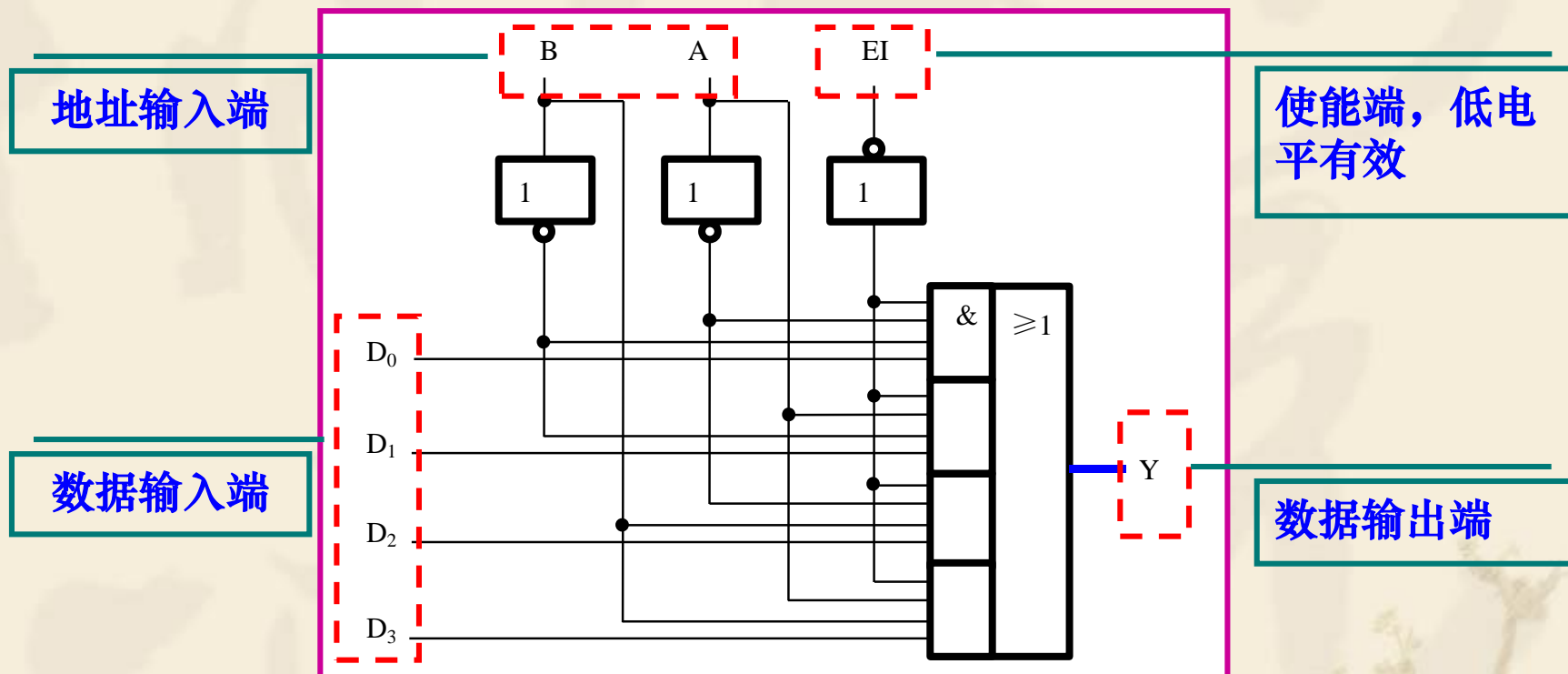
13.6.1 数据选择器的定义及功能

数据选择器：数据的定向。将多个数据源来的数据，经过选择后，分时送到唯一的通道去。

也相当于一个单刀多掷开关，又称“多路开关”。

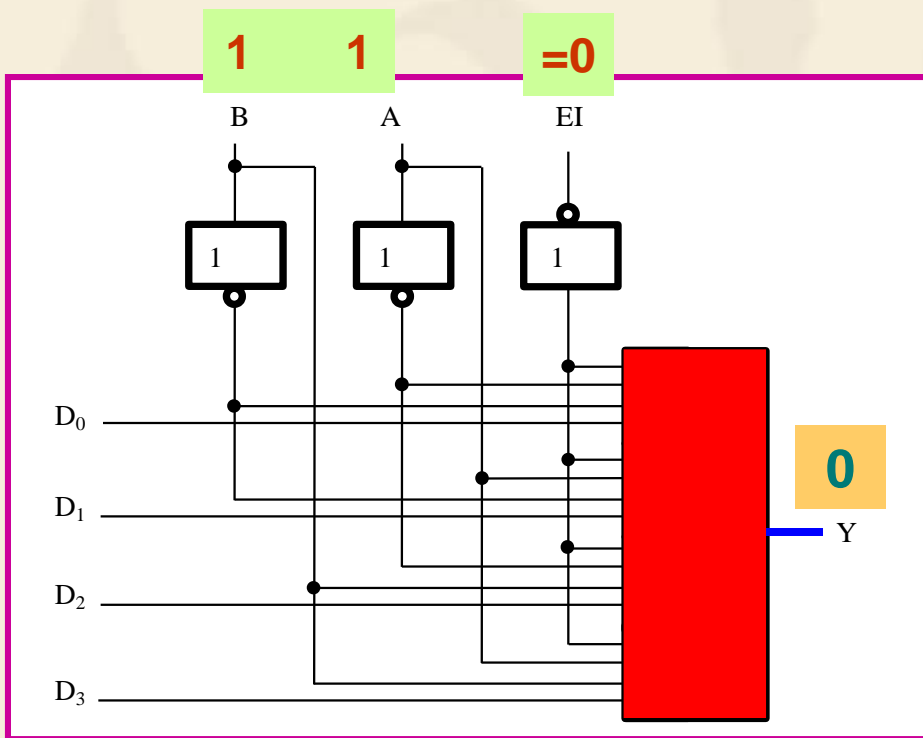


以4选1数据选择器为例（74LS153）



四选一数据选择器

74LS153的工作原理

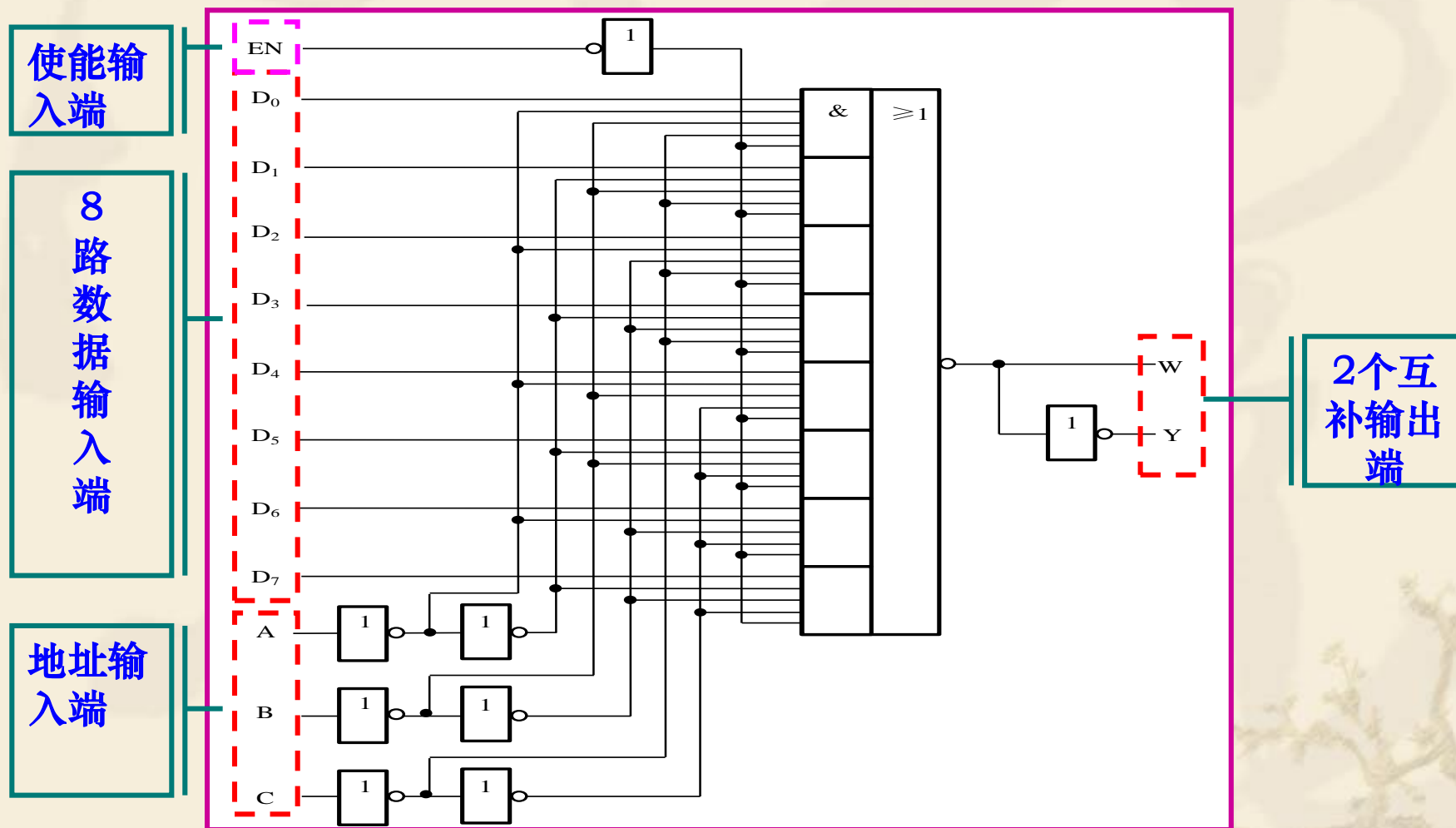


4选1 功能表

输 入			输出
使能	地址		
EI	B	A	Y
1	×	×	0
0	0	0	D ₀
0	0	1	D ₁
0	1	0	D ₂
0	1	1	D ₃

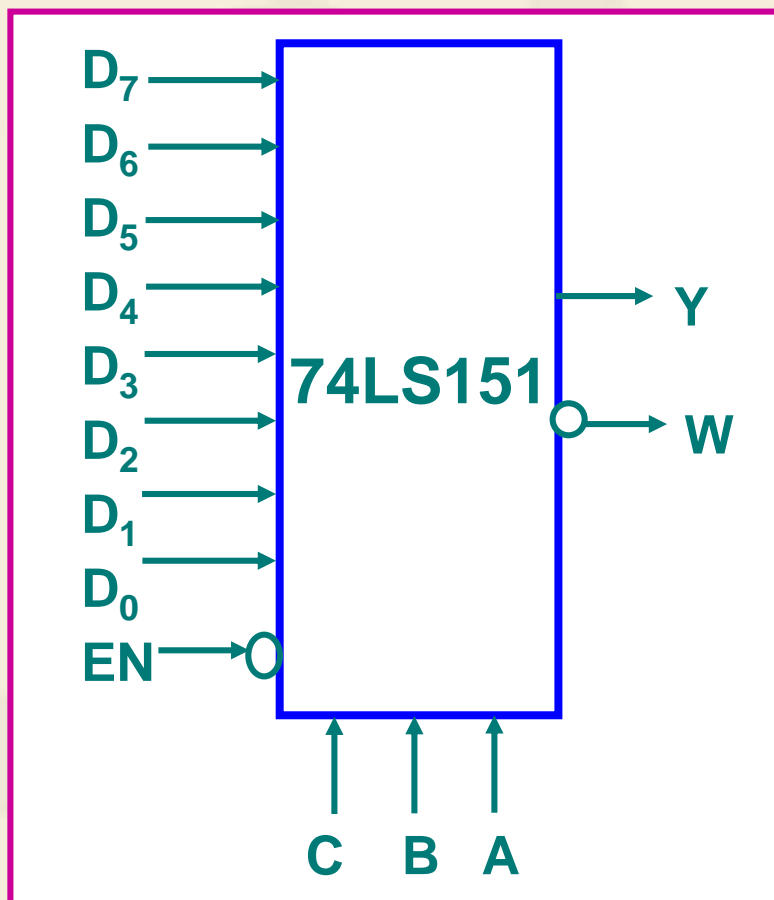
$$Y = D_0 \overline{B} \overline{A} \overline{E} \overline{I} + D_1 \overline{B} A \overline{E} \overline{I} + D_2 B \overline{A} \overline{E} \overline{I} + D_3 B A \overline{E} \overline{I}$$

介绍集成电路数据选择器74LS151

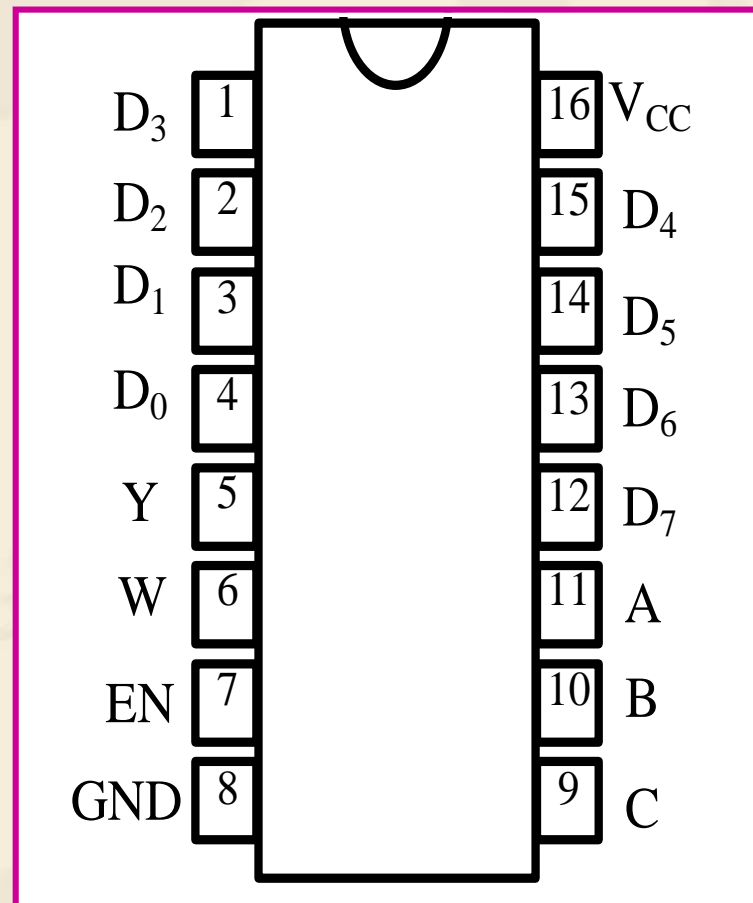


74LS151的逻辑图

集成电路数据选择器74LS151的逻辑符号及功能引脚



74LS151功能框图



74LS151引脚图

74LS151的功能表

输 入				输 出	
使 能	选 择			Y	W
EN	C	B	A		
H	X	X	X	L	H
L	L	L	L	D ₀	$\overline{D_0}$
L	L	L	H	D ₁	$\overline{D_1}$
L	L	H	L	D ₂	$\overline{D_2}$
L	L	H	H	D ₃	$\overline{D_3}$
L	H	L	L	D ₄	$\overline{D_4}$
L	H	L	H	D ₅	$\overline{D_5}$
L	H	H	L	D ₆	$\overline{D_6}$
L	H	H	H	D ₇	$\overline{D_7}$

- 当EN=1时，Y=0 。无效输出
- 当EN=0时，Y的表达式为：

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

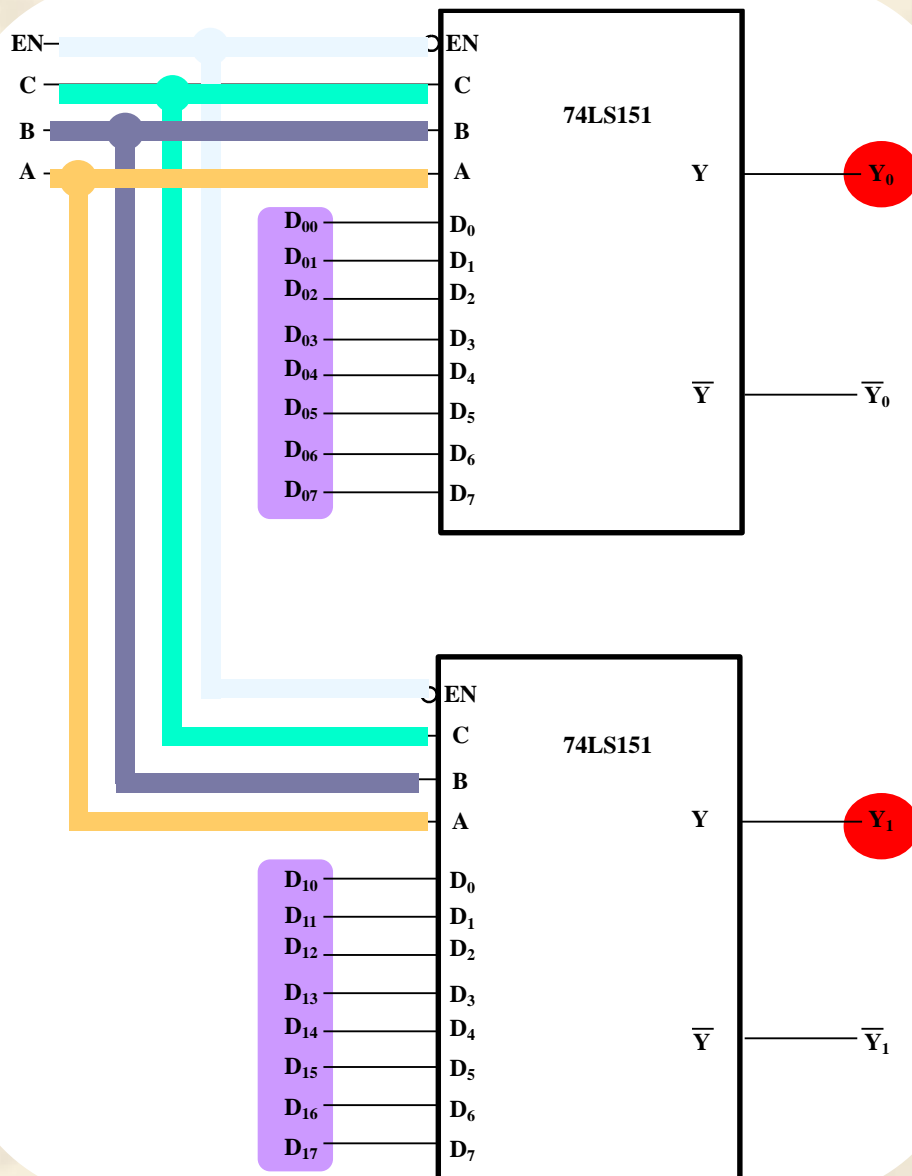
八选一MUX的逻辑表达式

$$\left\{ \begin{array}{l} \overline{\mathbf{EN}} = 1 \quad \mathbf{Y} = 0 \\ \overline{\mathbf{EN}} = 0 \quad \mathbf{Y} = \overline{\mathbf{A}_2}\overline{\mathbf{A}_1}\overline{\mathbf{A}_0}\mathbf{D}_0 + \overline{\mathbf{A}_2}\overline{\mathbf{A}_1}\mathbf{A}_0\mathbf{D}_1 + \overline{\mathbf{A}_2}\mathbf{A}_1\overline{\mathbf{A}_0}\mathbf{D}_2 + \overline{\mathbf{A}_2}\mathbf{A}_1\mathbf{A}_0\mathbf{D}_3 \\ \quad + \mathbf{A}_2\overline{\mathbf{A}_1}\overline{\mathbf{A}_0}\mathbf{D}_4 + \mathbf{A}_2\overline{\mathbf{A}_1}\mathbf{A}_0\mathbf{D}_5 + \mathbf{A}_2\mathbf{A}_1\overline{\mathbf{A}_0}\mathbf{D}_6 + \mathbf{A}_2\mathbf{A}_1\mathbf{A}_0\mathbf{D}_7 \\ \quad = \sum_{i=0}^7 \mathbf{m}_i \mathbf{D}_i \end{array} \right.$$

如需选择多位
数据时：

可由几个1位的
数选器并联而成

2位八选一数选
器的连接方法

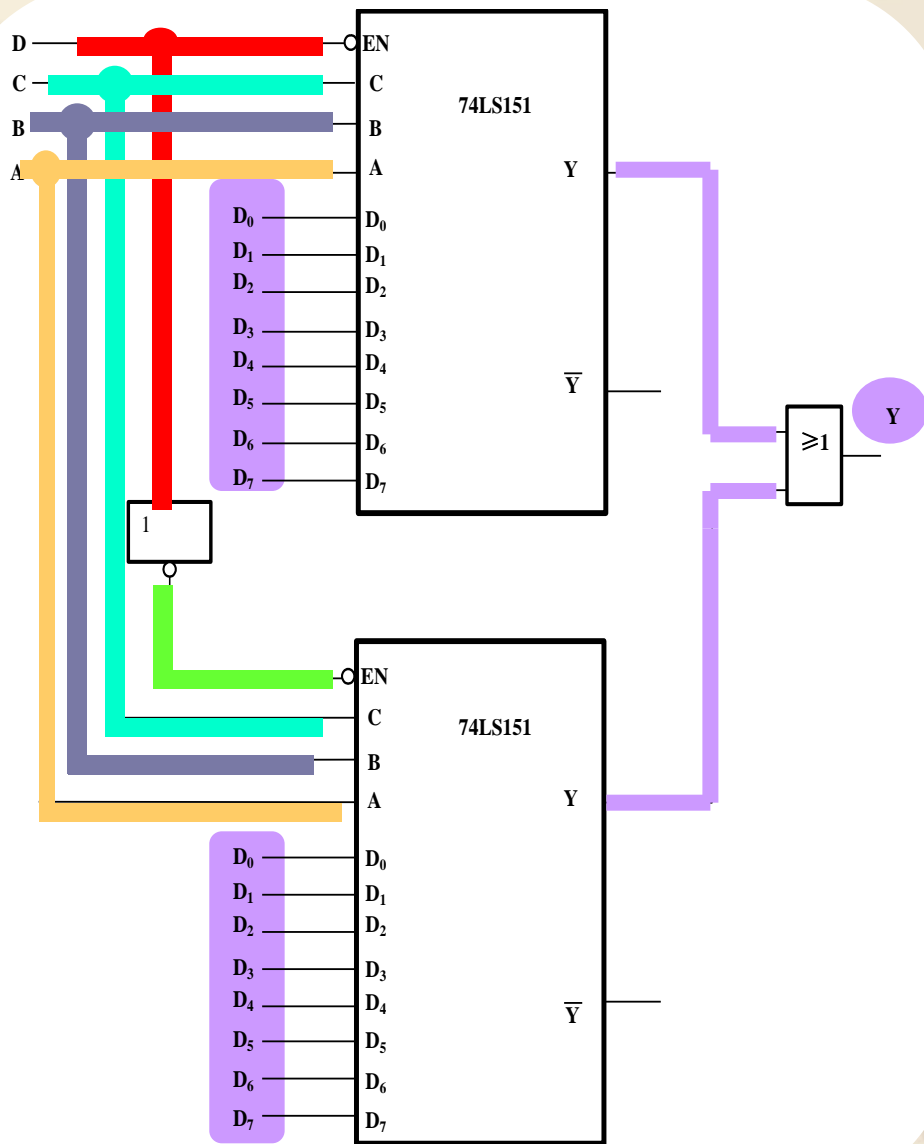


用两片8选1数选器构成16选1数选器

使能端作为地址输入端的最高位。

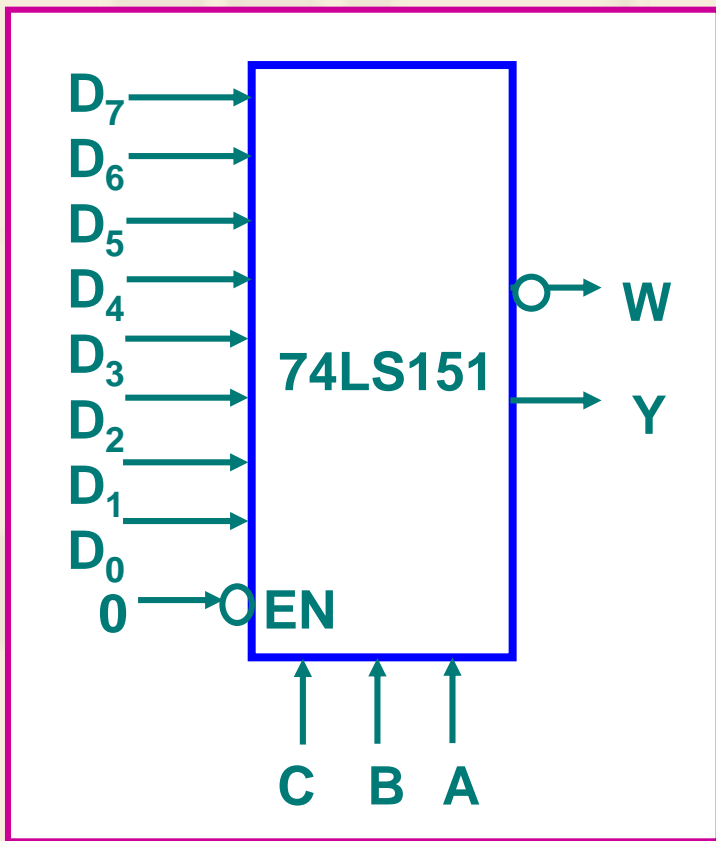
$D_0—D_{15}$ 为16路数据输入；Y为输出

16选1数据选择器



数据选择器74LS151的应用

- 1、完成多路数据的传送，如前所述。
- 2、逻辑函数产生器



从功能表和逻辑图上可知：

当使能端 $\overline{EN}=0$ 时，输出 Y 的表达式为：

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

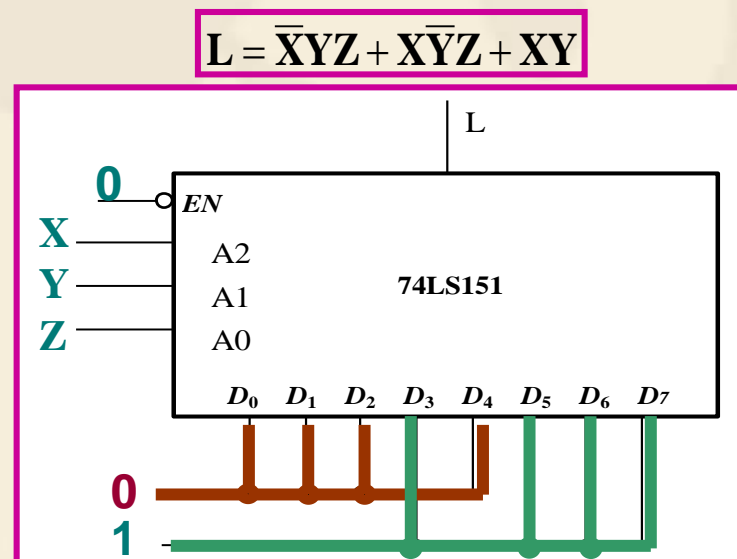
当 $D_i=1$ 时，对应的最小项在表达式中出现；
当 $D_i=0$ 时，不出现。

将地址信号 CBA 作为函数的输入变量，
数据输入 $D_0—D_7$ 作为控制信号（控制各最小项在输出函数中是否出现），则数选器就成为一个逻辑函数产生器。

例 试用8选1数据选择器74LS151产生逻辑函数 $L = \bar{X}YZ + X\bar{Y}Z + XY$

解:
$$L = \bar{X}YZ + X\bar{Y}Z + XYZ + XY\bar{Z}$$
$$= m_3D_3 + m_5D_5 + m_6D_6 + m_7D_7$$

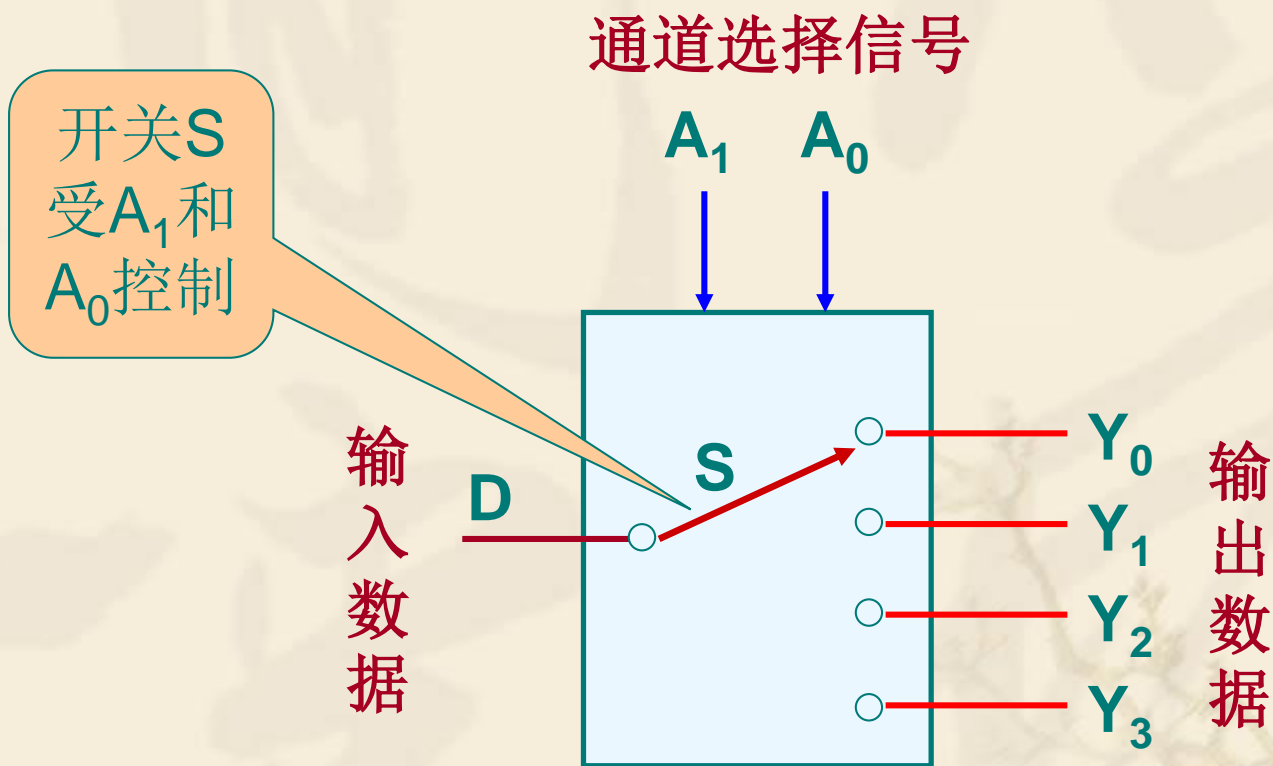
显然, 当 $D_3=D_5=D_6=D_7=1$,
 $D_0=D_1=D_2=D_4=0$ 时,
74151的输出即为逻辑函数L。



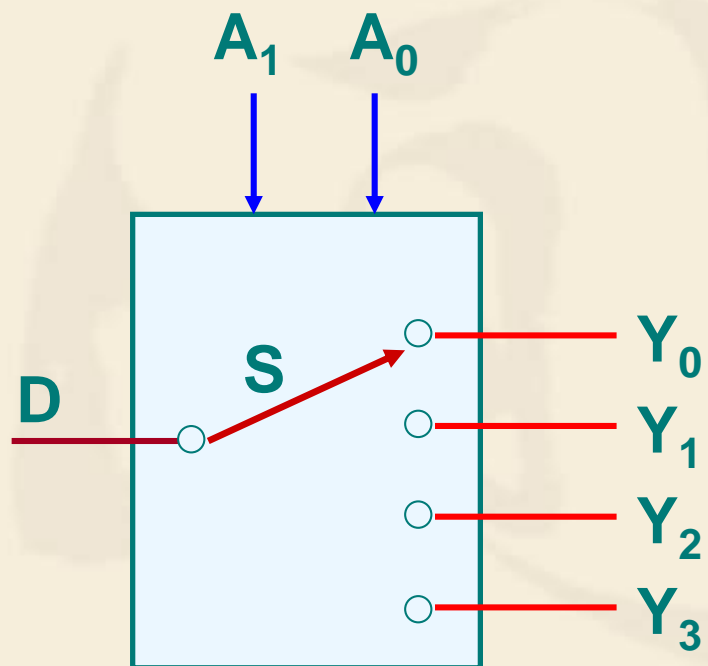
使用数据选择器来实现逻辑函数的好处是：逻辑函数无需化简

13.7 数据分配器

数据分配器：数据的分路。将一个数据源来的数据分时送到多个不同的通道上去。相当于一个单刀多掷开关。



数据分配器示意图

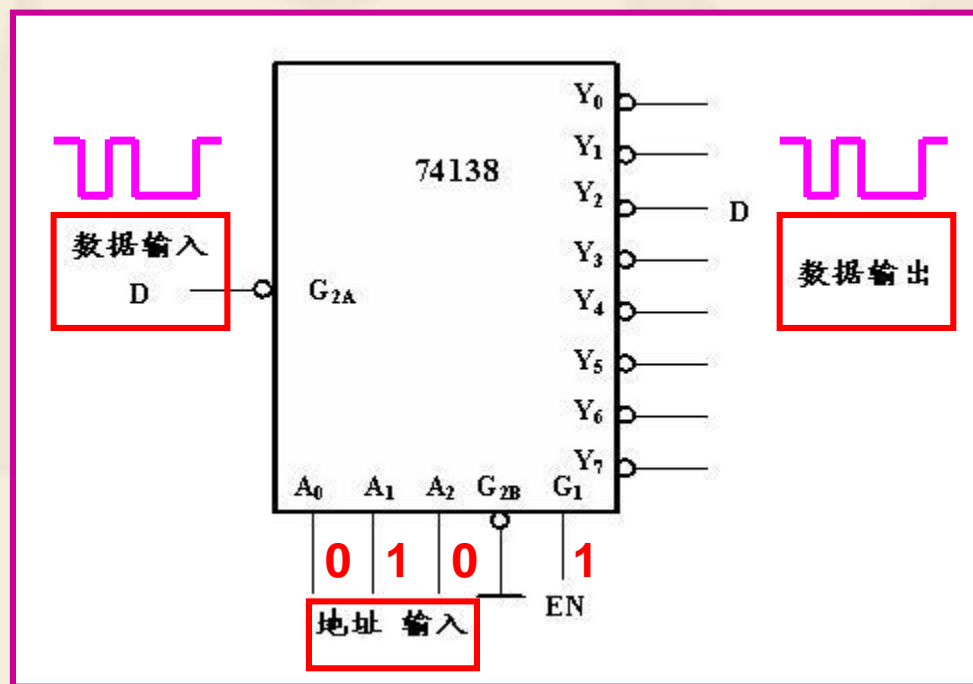


数据分配器的结构和译码器相同,只是译码器的地址信号在这里作为选择信号而已

故可用二—四线译码器作为1路—4路数据分配器

可用三—八线译码器作为1路—8路数据分配器

例:用3/8线译码器，将数据信号分配到8个不同的通道。



$$Y_2 = \overline{(G_1 \cdot \overline{G_{2A}} \cdot \overline{G_{2B}})} \cdot \overline{A_2} \cdot A_1 \cdot \overline{A_0} = G_{2A}$$

此时:

$$Y_0 = \overline{(G_1 \cdot \overline{G_{2A}} \cdot \overline{G_{2B}})} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0} = 1$$

$$Y_1 = \overline{(G_1 \cdot \overline{G_{2A}} \cdot \overline{G_{2B}})} \cdot \overline{A_2} \cdot \overline{A_1} \cdot A_0 = 1$$

⋮

$$Y_i = \overline{G_1} \cdot \overline{G_{2A}} \cdot \overline{G_{2B}} \cdot m_i$$

74138译码器作为数据分配器时的功能表

[illegible]

13.8 用MSI实现组合逻辑函数

用MSI设计组合电路的步骤：

- 1) 列真值表
- 2) 写逻辑函数表达式（SSI中一般要写成最简，此不必）
- 3) 将表达式变换成与所用MSI逻辑函数相似的形式
- 4) 根据对比结果，确定数据线，选择线的对应关系，画图

一、用二进制译码器实现逻辑函数

例：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。

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

解： $F_1(A, B, C) = AC + \overline{B}C = AC(B + \overline{B}) + \overline{B}C(A + \overline{A})$

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

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

$$= \overline{\overline{\overline{A}BC} \cdot \overline{A\overline{B}C} \cdot \overline{ABC}}$$

$$= \overline{m_1 \cdot m_5 \cdot m_7} = \overline{Y_1 \cdot Y_5 \cdot Y_7}$$

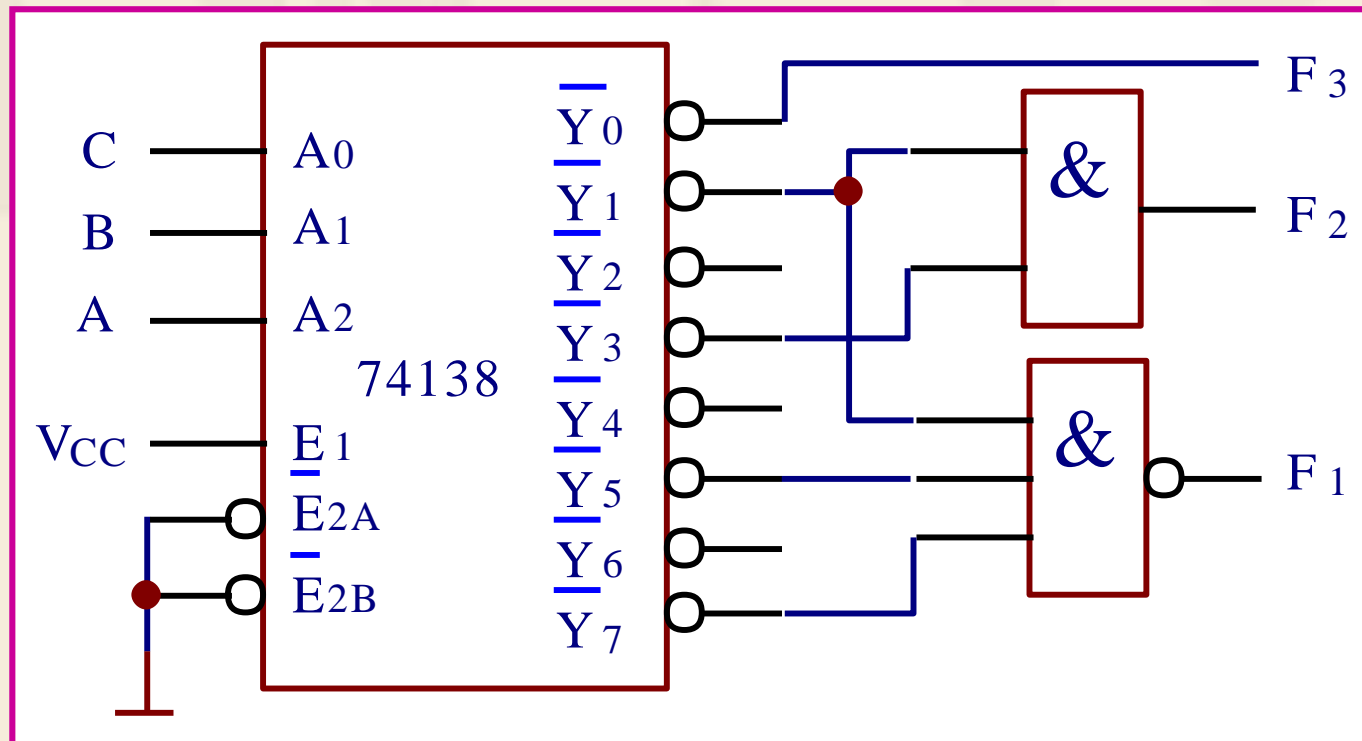
若将译码器的输入信号A₂、A₁、A₀分别对应于A、B、C,则有

$$\overline{Y_1} = \overline{\overline{A}BC} \quad \overline{Y_5} = \overline{A\overline{B}C} \quad \overline{Y_7} = \overline{ABC}$$

$$F_2(A, B, C) = A + \bar{C} = \overline{m_1 + m_3} = \overline{m_1} \cdot \overline{m_3} = \overline{Y_1} \cdot \overline{Y_3}$$

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

$$F_1(A, B, C) = \overline{\overline{Y_1} \cdot \overline{Y_5} \cdot \overline{Y_7}}$$



二、用数据选择器实现逻辑函数

例1： 试用8选1数据选择器74151实现逻辑函数： $L = AB + BC + AC$

法1： 8选1数据选择器标准“与或式”为

$$Y = \overline{A_2}\overline{A_1}\overline{A_0}D_0 + \overline{A_2}\overline{A_1}A_0D_1 + \overline{A_2}A_1\overline{A_0}D_2 + \overline{A_2}A_1A_0D_3 + A_2\overline{A_1}\overline{A_0}D_4 + A_2\overline{A_1}A_0D_5 + A_2A_1\overline{A_0}D_6 + A_2A_1A_0D_7$$

将待实现的逻辑函数转换成最小项表达式并令 $A=A_2, B=A_1, C=A_0$ ：

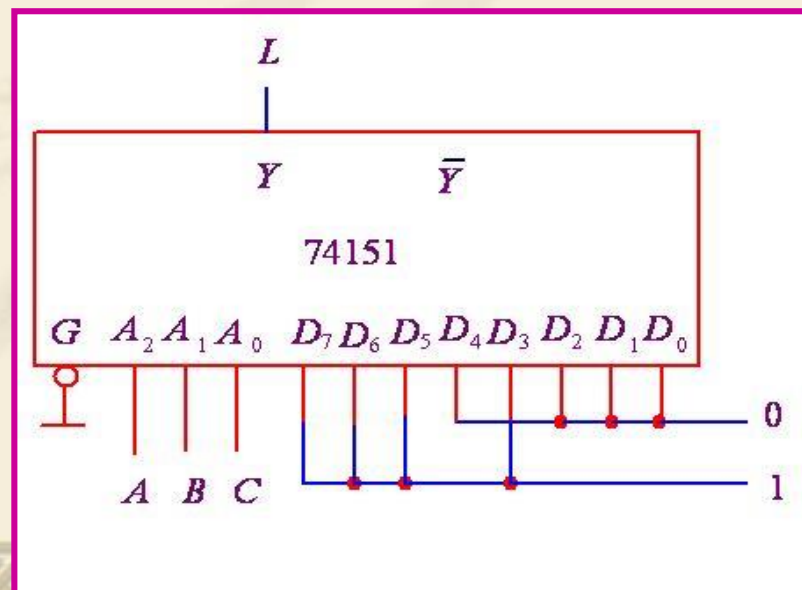
$$\begin{aligned} L &= \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC \\ &= \overline{A_2}A_1A_0 + A_2\overline{A_1}A_0 + A_2A_1\overline{A_0} + A_2A_1A_0 \end{aligned}$$

比较以上两式可得：

$$D_3 = D_5 = D_6 = D_7 = 1$$

$$D_0 = D_1 = D_2 = D_4 = 0$$

画出连线图如右图所示：



法2: 8选1数据选择器标准“与或式”为

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

令 $A=A_2, B=A_1, C=A_0$ 并将待实现的逻辑函数转换成最小项表达式:

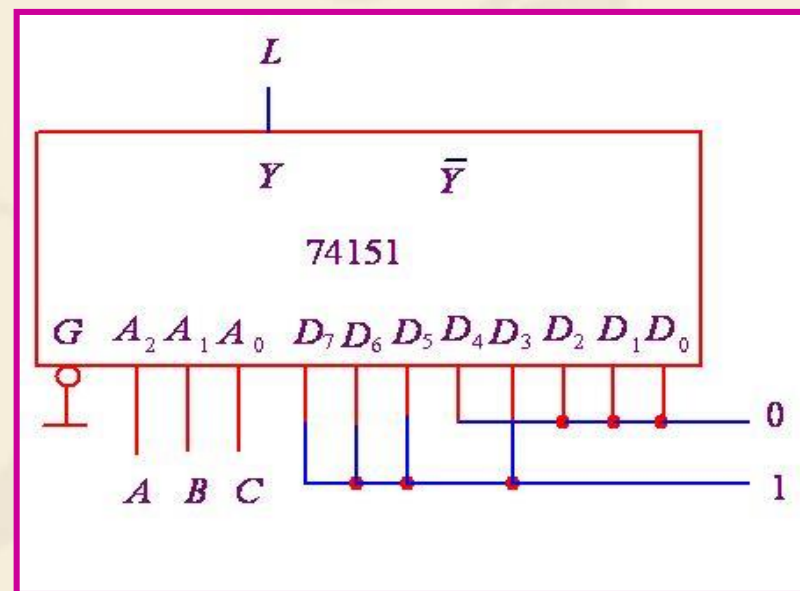
$$\begin{aligned} L &= \bar{A}BC + A\bar{B}C + ABC\bar{C} + ABC \\ &= m_3 + m_5 + m_6 + m_7 \end{aligned}$$

比较以上两式可得:

$$D_3 = D_5 = D_6 = D_7 = 1$$

$$D_0 = D_1 = D_2 = D_4 = 0$$

画出连线图如右图所示:

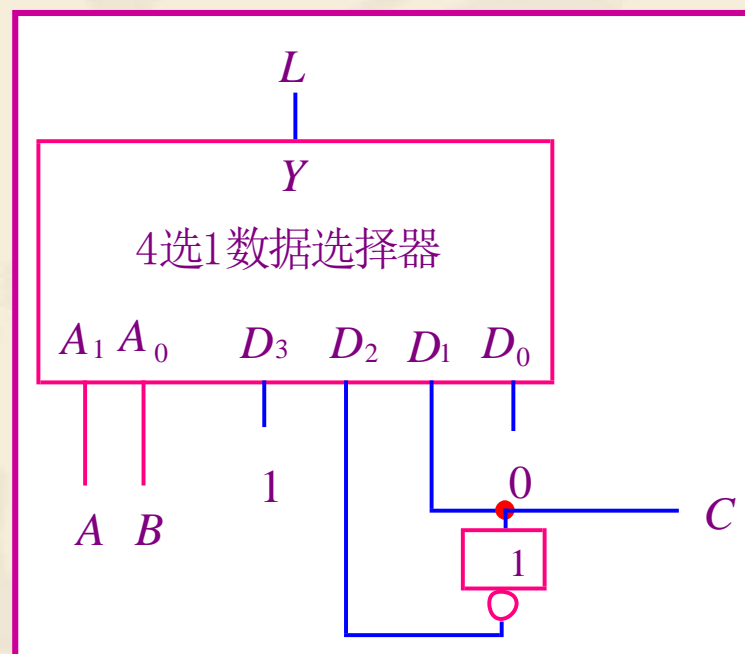


例2：试用4选1数据选择器实现逻辑函数 $L = AB + BC + A\bar{C}$

解：将 A 、 B 接到地址输入端， C 加到适当的数据输入端。

作出逻辑函数 L 的真值表，根据真值表画出连线图。

L 的真值表			
A	B	C	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



本章小结

1. 常用的中规模组合逻辑器件包括编码器、译码器、数据选择器、数值比较器、加法器等。
2. 上述组合逻辑器件除了具有其基本功能外，还可用来设计组合逻辑电路。应用中规模组合逻辑器件进行组合逻辑电路设计的一般原则是：使用MSI芯片的个数和品种型号最少，芯片之间的连线最少
3. 用MSI芯片设计组合逻辑电路最简单和最常用的方法是，用数据选择器设计多输入、单输出的逻辑函数；用二进制译码器设计多输入、多输出的逻辑函数。



以下为课件所用
素材

$$\begin{array}{r}
 0 \\
 +) 0 \\
 \hline
 0
 \end{array}$$

$$\begin{array}{r}
 1 \\
 +) 0 \\
 \hline
 1
 \end{array}$$

$$\begin{array}{r}
 0 \\
 +) 1 \\
 \hline
 1
 \end{array}$$

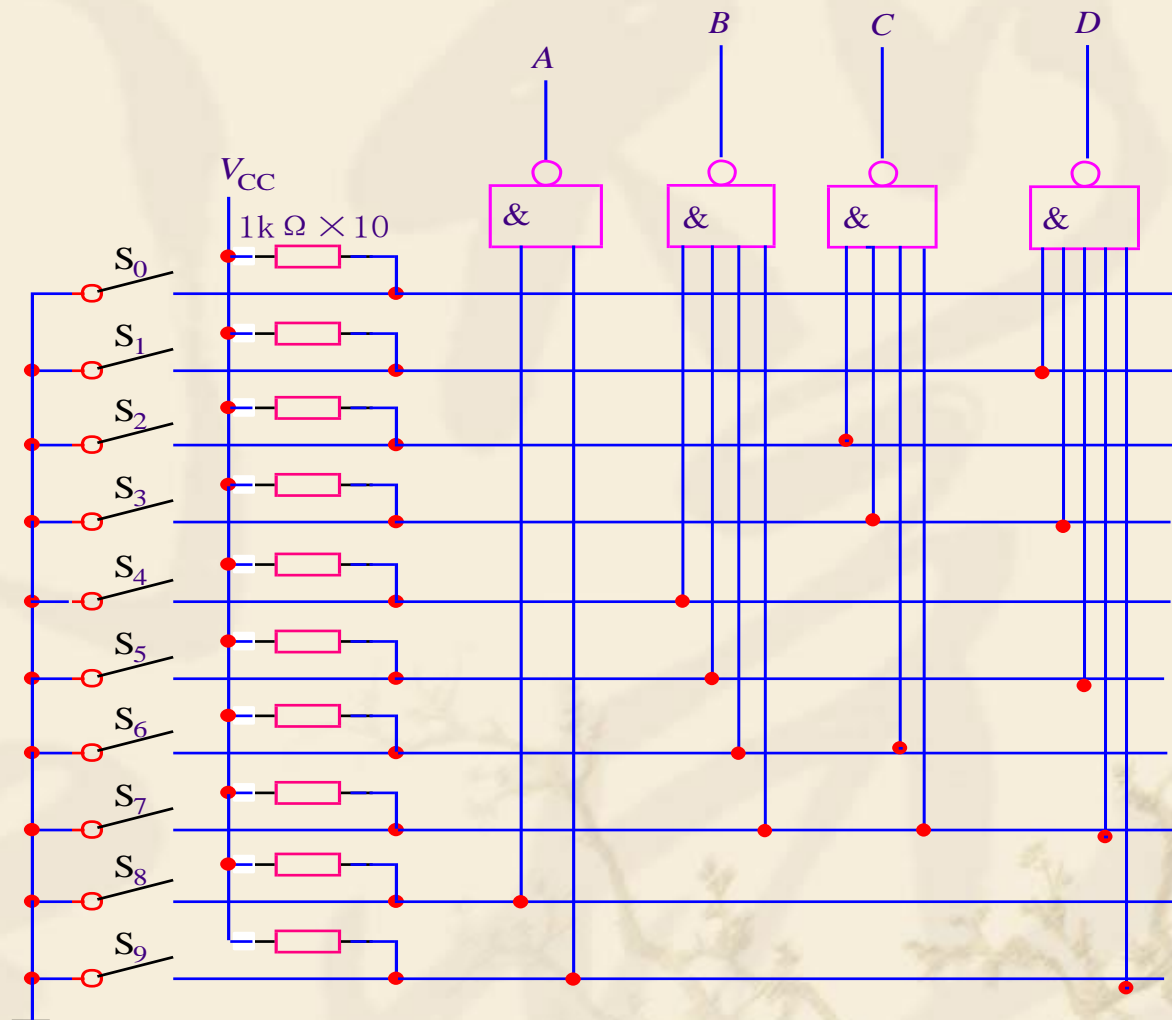
$$\begin{array}{r}
 1 \\
 +) 1 \\
 \hline
 1 \ 0
 \end{array}$$



C_{i-1}	A_i	B_i	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

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

		$A_i B_i$			
		00	01	11	10
C_i	0	0	0	1	0
	1	0	1	1	1



[illegible][illegible]