

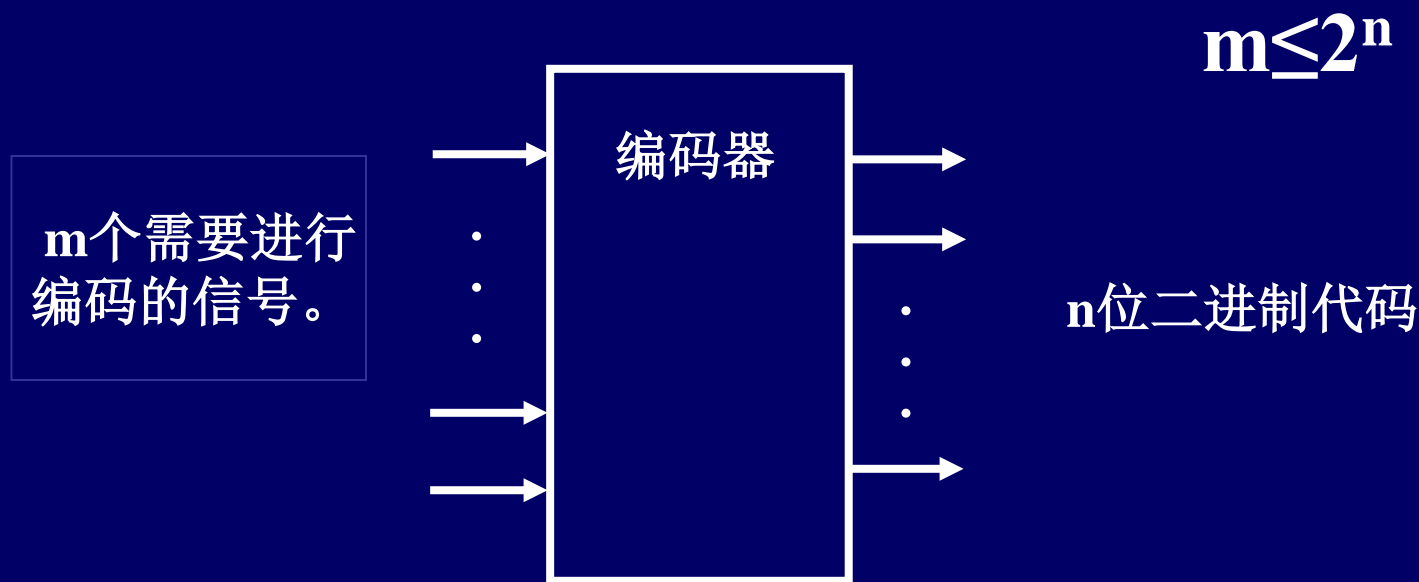
4.5 常用组合逻辑电路的设计及其应用

- (1) 电路设计模型
- (2) 对应的中规模 (MSI) 逻辑器件
- (3) 硬件描述语言 Verilog HDL 建模

1 编码器 (Encoder)

在数字系统中，采用机器状态对文字、符号、运算符、数字或状态信号进行编码，形成与其对应的二进制代码。

编码器就是完成编码过程的逻辑电路。



设计模型

一. 二进制编码器

用 n 位二进制代码对 $m=2^n$ 个信号进行编码的电路。

以3位二进制编码器为例，其输入是8个需要进行编码的信号 $I_0 \sim I_7$ ，输出是3位二进制代码 $Y_2 Y_1 Y_0$ 。

在任何时刻，编码器只能对一个输入信号进行编码，即不允许多个输入信号同时出现（有效），所以 $I_0 \sim I_7$ 是互相排斥的变量，可列出简化真值表。

输入	输出		
	Y_2	Y_1	Y_0
I_0	0	0	0
I_1	0	0	1
I_2	0	1	0
I_3	0	1	1
I_4	1	0	0
I_5	1	0	1
I_6	1	1	0
I_7	1	1	1

$$Y_2 = I_4 + I_5 + I_6 + I_7$$

$$Y_1 = I_2 + I_3 + I_6 + I_7$$

$$Y_0 = I_1 + I_3 + I_5 + I_7$$

二. 二-十进制编码器

将代表十进制数的10个输入信号0~9分别编成对应的BCD码。

输入	输出			
	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

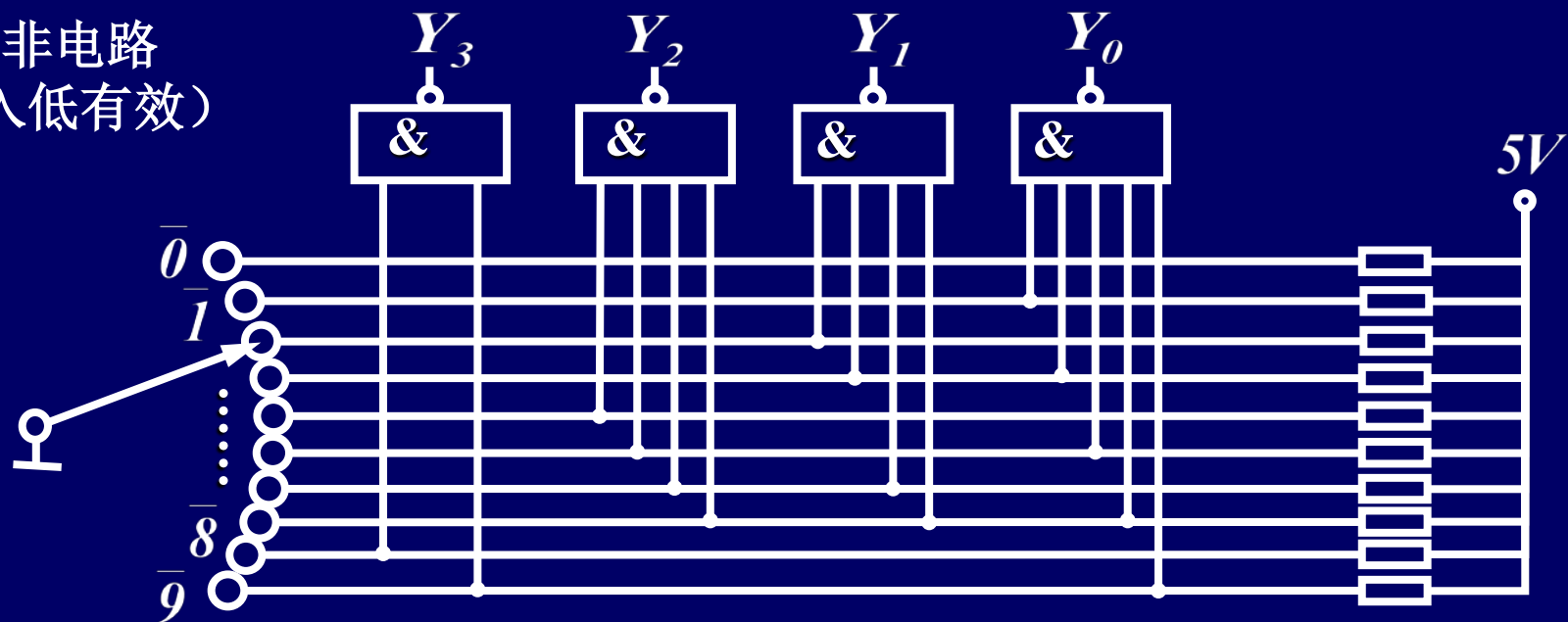
$$Y_3 = 8 + 9 = \overline{\overline{8} \bullet \overline{9}}$$

$$Y_2 = 4 + 5 + 6 + 7 = \overline{\overline{4} \bullet \overline{5} \bullet \overline{6} \bullet \overline{7}}$$

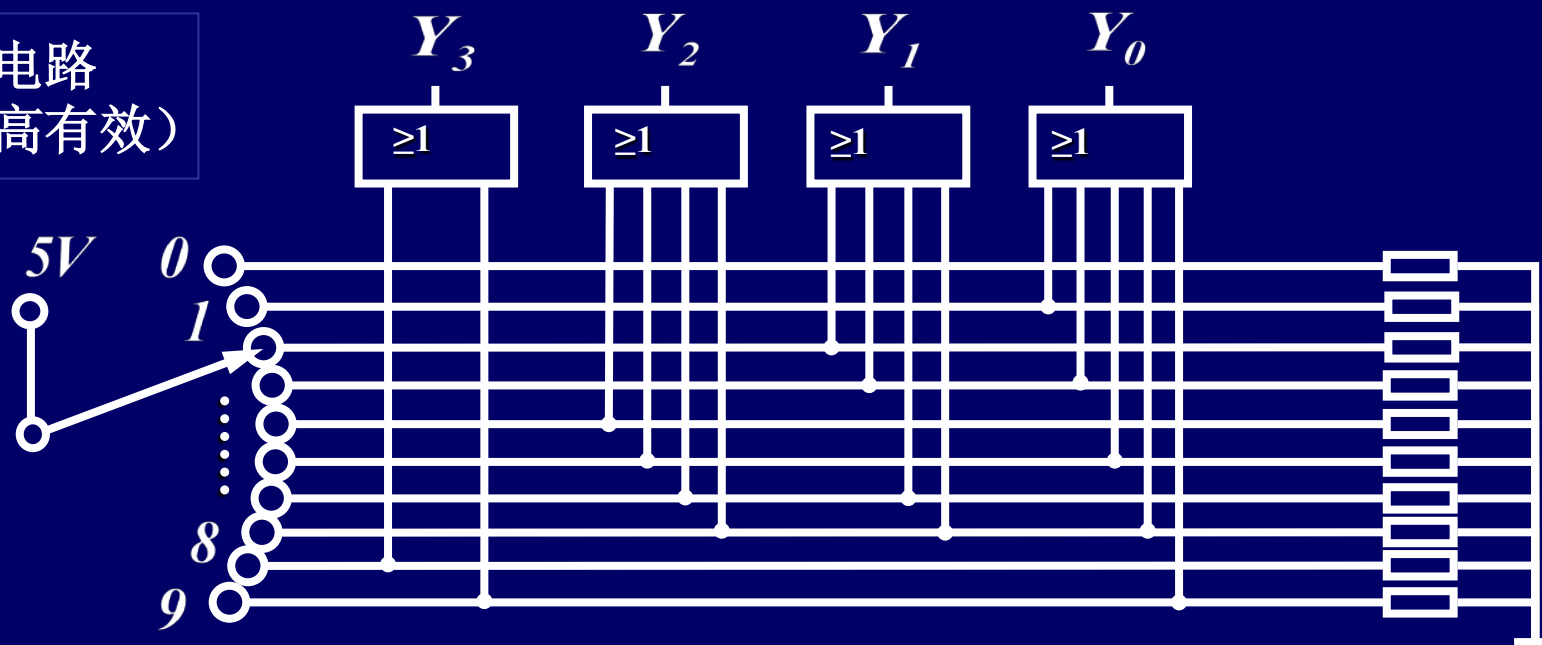
$$Y_1 = 2 + 3 + 6 + 7 = \overline{\overline{2} \bullet \overline{3} \bullet \overline{6} \bullet \overline{7}}$$

$$Y_0 = 1 + 3 + 5 + 7 + 9 = \overline{\overline{1} \bullet \overline{3} \bullet \overline{5} \bullet \overline{7} \bullet \overline{9}}$$

与非电路
(输入低有效)



或电路
(输入高有效)



三. 8—3普通编码器的Verilog HDL建模

```
1 module code8_3(keyin,codeout,enout);
2     input  [8:1] keyin;
3     output [2:0] codeout;
4     output      enout;
5     reg      [2:0] codeout;
6     reg      enout;
7     always @ (keyin)
8         case(keyin)
9             8'b10000000 : {enout,codeout}=4'b1_000;
10            8'b01000000 : {enout,codeout}=4'b1_001;
11            8'b00100000 : {enout,codeout}=4'b1_010;
12            8'b00010000 : {enout,codeout}=4'b1_011;
13            8'b00001000 : {enout,codeout}=4'b1_100;
14            8'b00000100 : {enout,codeout}=4'b1_101;
15            8'b00000010 : {enout,codeout}=4'b1_110;
16            8'b00000001 : {enout,codeout}=4'b1_111;
17            default     : {enout,codeout}=4'b0_000;
18        endcase
19 endmodule
20
```

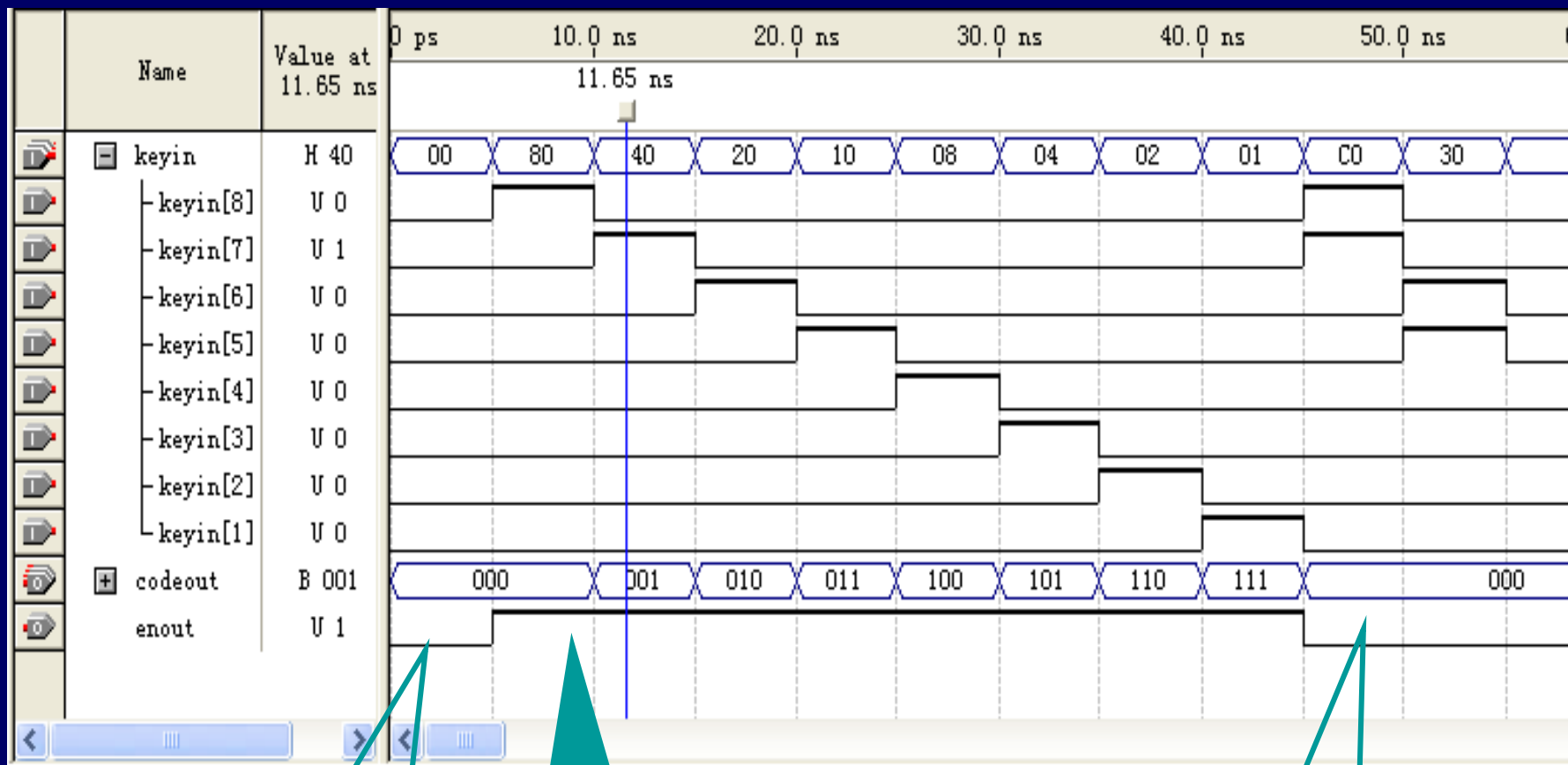
8个待编码信息输入

3位编码输出

编码有效标志

无输入信息

功能仿真



无输入
编码无效

有互斥输入
编码有效

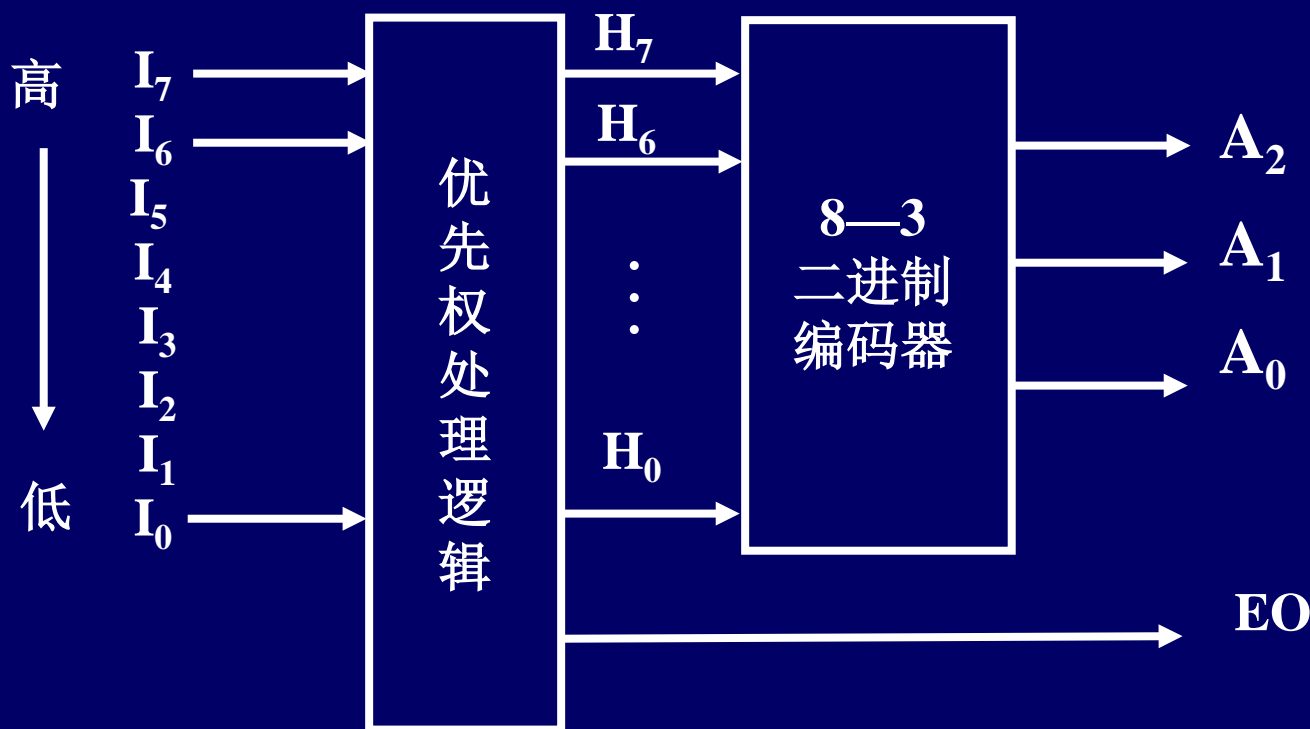
同时输入
编码无效

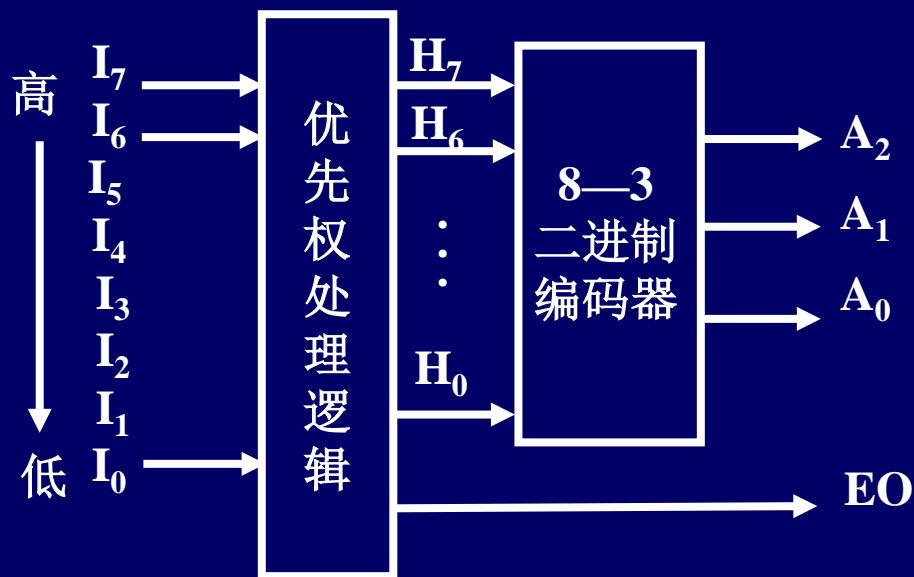
四. 优先权编码器

允许多个输入信号同时有效，但只对其中优先级别最高的信号进行编码。

具有指定输入端优先权顺序的编码器，称为优先权编码器。

以8—3优先权编码器为例进行设计说明





I_i 均为高有效, H_i 和 I_i 的关系是: 当 I_i 具有较高优先权且为1时, H_i 才为1。

$$H_7 = I_7$$

$$H_6 = I_6 \overline{I_7}$$

$$H_5 = I_5 \overline{I_7} \overline{I_6}$$

⋮

$$H_0 = I_0 \overline{I_7} \overline{I_6} \overline{I_5} \cdots \overline{I_1}$$

$$A_2 = H_4 + H_5 + H_6 + H_7$$

$$A_1 = H_2 + H_3 + H_6 + H_7$$

$$A_0 = H_1 + H_3 + H_5 + H_7$$

为判断是否出现有效输入请求, 增加一个使能输出EO, 以便于级联:

$$EO = \overline{I_0} \overline{I_1} \overline{I_2} \cdots \overline{I_7}$$

输入信号均无效时, EO才有效。

优先权编码器功能表

输入（低有效）									输出（低有效）				
$/EI$	$/I_7$	$/I_6$	$/I_5$	$/I_4$	$/I_3$	$/I_2$	$/I_1$	$/I_0$	$/Y_2$	$/Y_1$	$/Y_0$	$/CS$	$/EO$
1	d	d	d	d	d	d	d	d	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	d	d	d	d	d	d	d	0	0	0	0	1
0	1	0	d	d	d	d	d	d	0	0	1	0	1
0	1	1	0	d	d	d	d	d	0	1	0	0	1
0	1	1	1	0	d	d	d	d	0	1	1	0	1
0	1	1	1	1	0	d	d	d	1	0	0	0	1
0	1	1	1	1	1	0	d	d	1	0	1	0	1
0	1	1	1	1	1	1	0	d	1	1	0	0	1
0	1	1	1	1	1	1	1	0	1	1	1	0	1

采用Verilog HDL建模

优先权编码器 74LS148

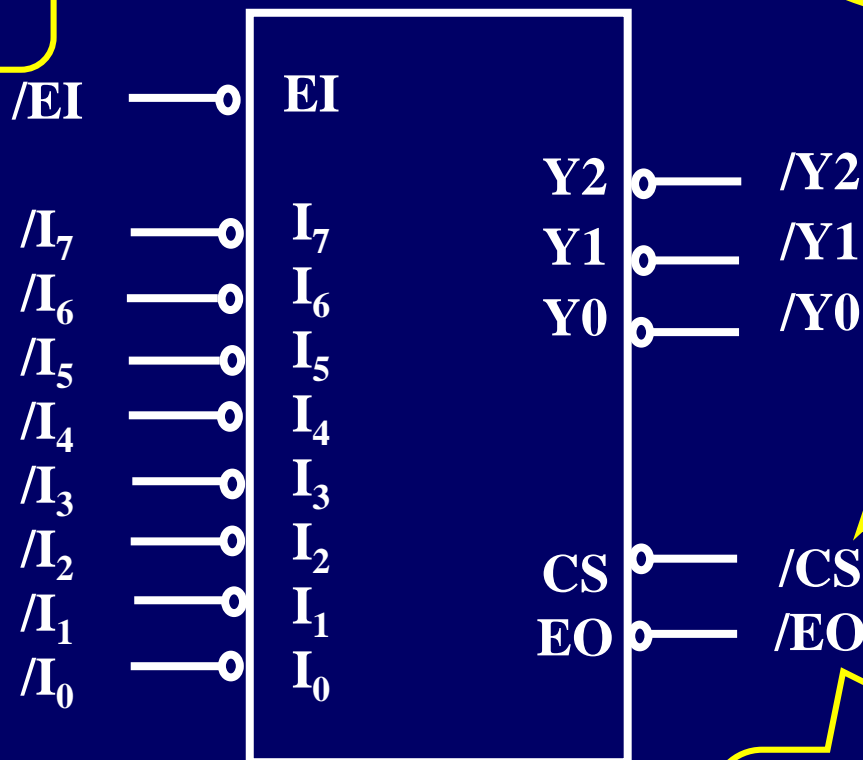
使能输入端，
低有效。

低有效（反码）
编码输出。

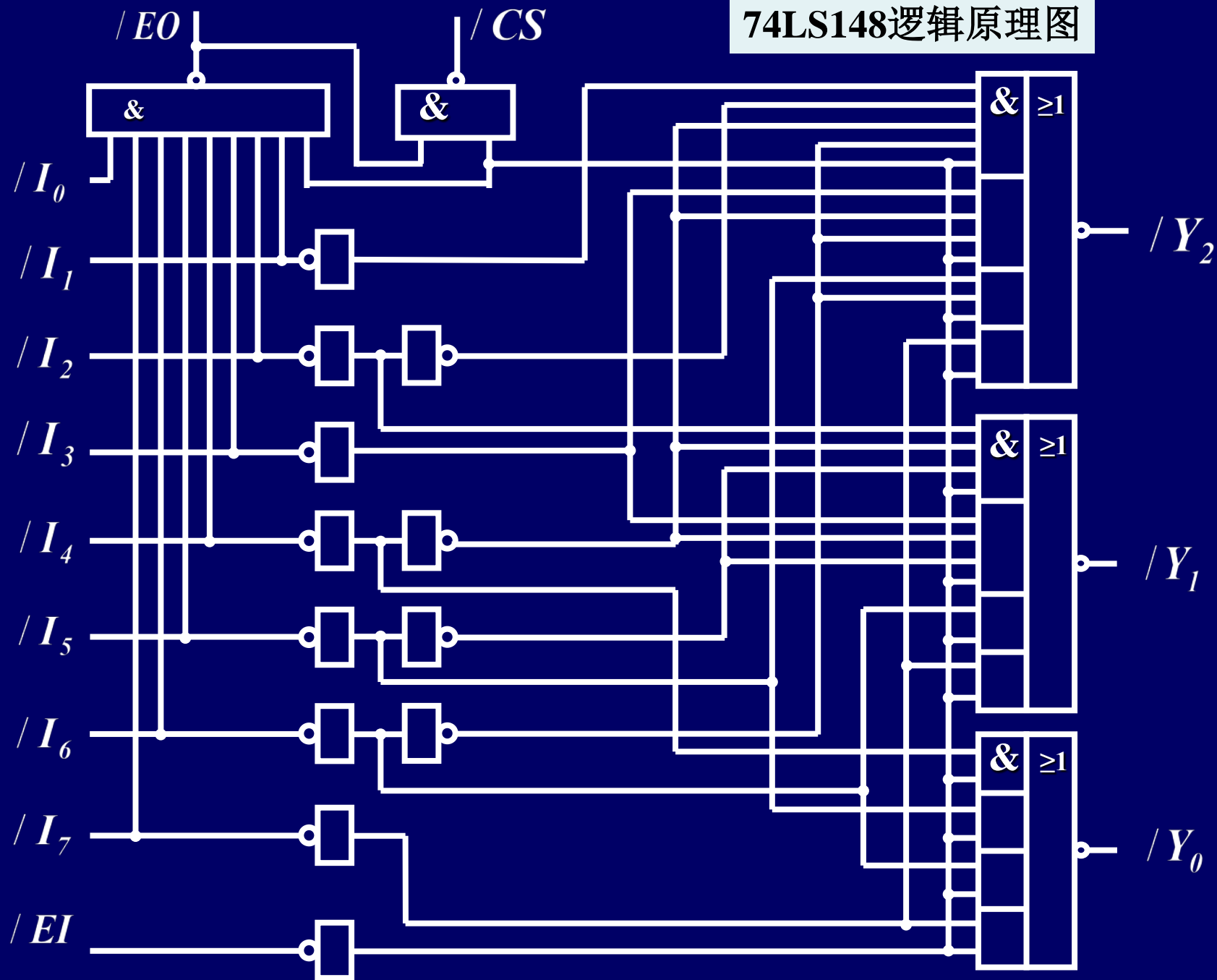
低有效输出
0: 表示编码器有信号输入；
1: 表示编码器没有信号输入；

低有效使能输出端：
当/EI有效，但无信号输入时，/EO有效。可用于级联时启动低片工作。

信号输入
高
↓
低有效



74LS148逻辑原理图



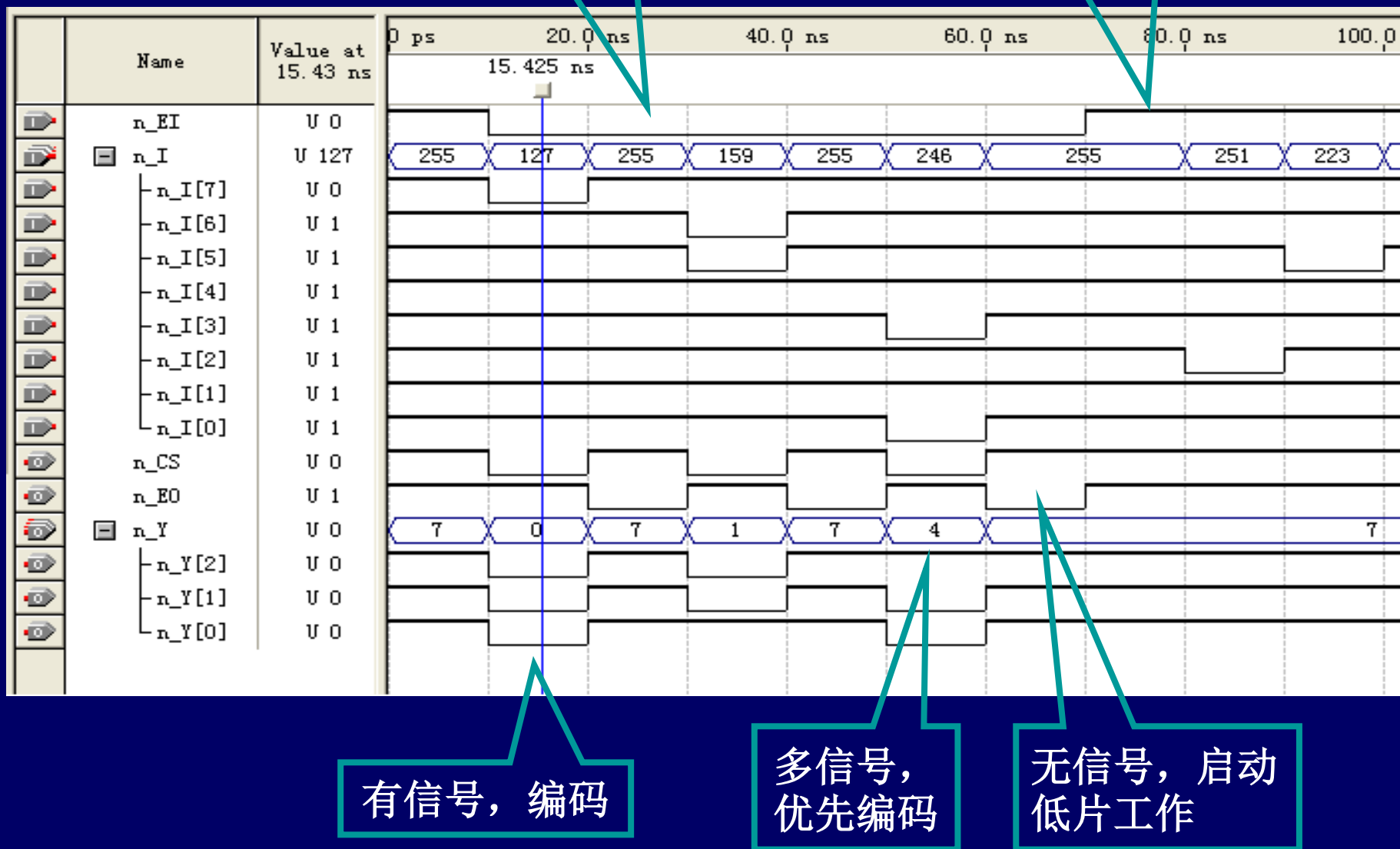
Verilog HDL建模

根据if_else语句的隐含优先级特性

abc youxian_148.v*

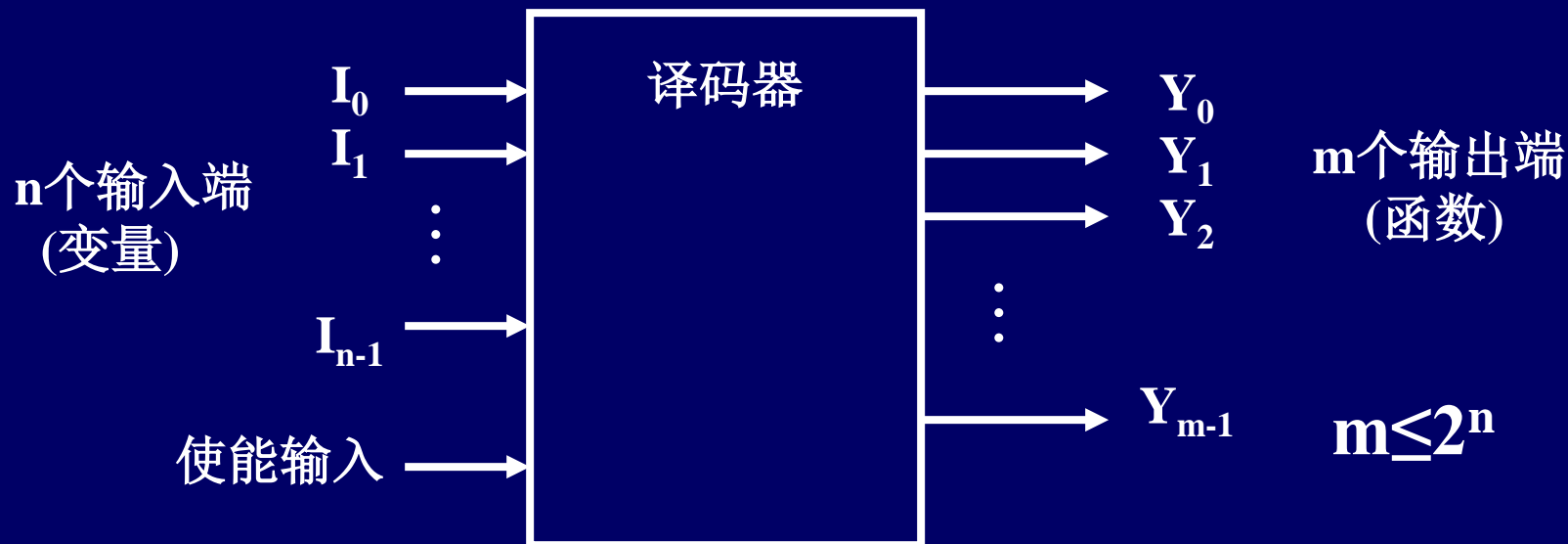
```
1 module youxian_148(n_EI,n_I,n_Y,n_CS,n_EO);
2     input          n_EI;
3     input  [7:0]   n_I;
4     output          n_CS, n_EO;
5     output  [2:0]   n_Y;
6     reg            n_CS, n_EO;
7     reg  [2:0]     n_Y;
8     always @ (n_EI or n_I)
9         if (n_EI==0)
10             if (n_I[7]==0)          {n_CS,n_EO,n_Y}=5'b0_1_000;
11             else if (n_I[6]==0)     {n_CS,n_EO,n_Y}=5'b0_1_001;
12             else if (n_I[5]==0)     {n_CS,n_EO,n_Y}=5'b0_1_010;
13             else if (n_I[4]==0)     {n_CS,n_EO,n_Y}=5'b0_1_011;
14             else if (n_I[3]==0)     {n_CS,n_EO,n_Y}=5'b0_1_100;
15             else if (n_I[2]==0)     {n_CS,n_EO,n_Y}=5'b0_1_101;
16             else if (n_I[1]==0)     {n_CS,n_EO,n_Y}=5'b0_1_110;
17             else if (n_I[0]==0)     {n_CS,n_EO,n_Y}=5'b0_1_111;
18             else                    {n_CS,n_EO,n_Y}=5'b1_0_111; //?
19         else                        {n_CS,n_EO,n_Y}=5'b1_1_111; //?
20 endmodule
21
```

功能仿真波形



2 译码器 (Decoder)

译码器是应用最广泛的一种多输入、多输出的逻辑器件。



常见译码器种类:

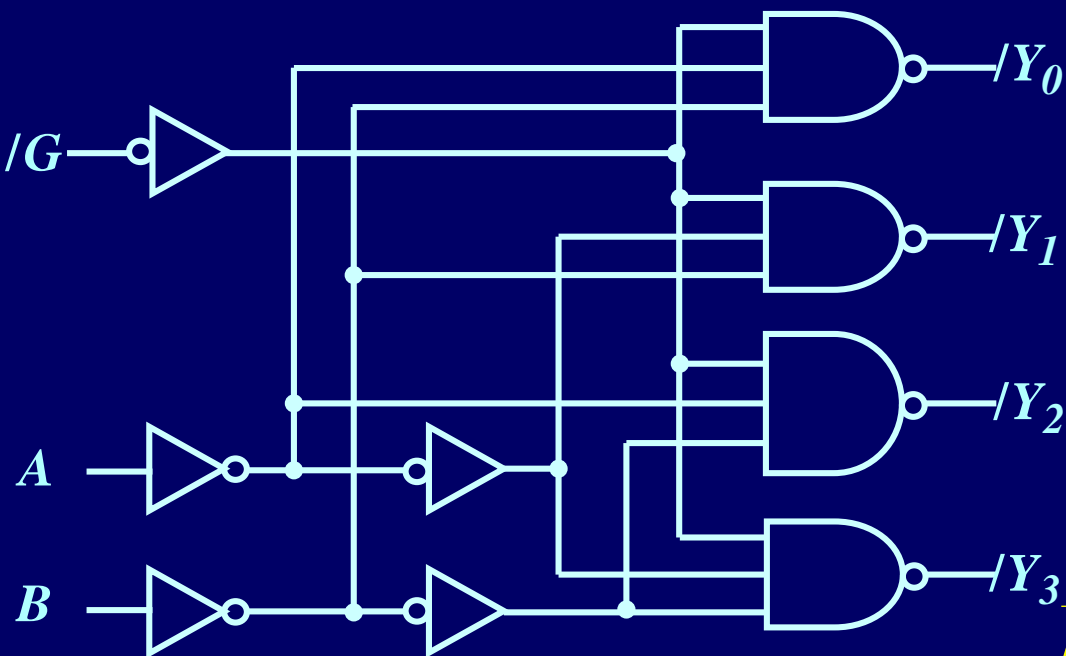
二进制译码器

BCD译码器

BCD—七段显示译码器

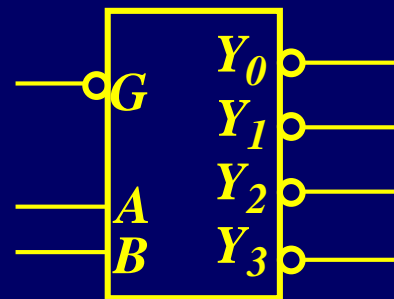
一. 二进制译码器 ($n=2^n$ 译码器) 原理

原理图



注意这里B是高位，A是低位。

逻辑符号



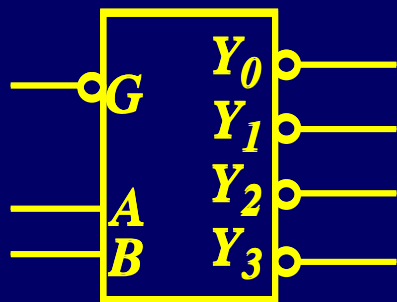
2-4译码器

真值表

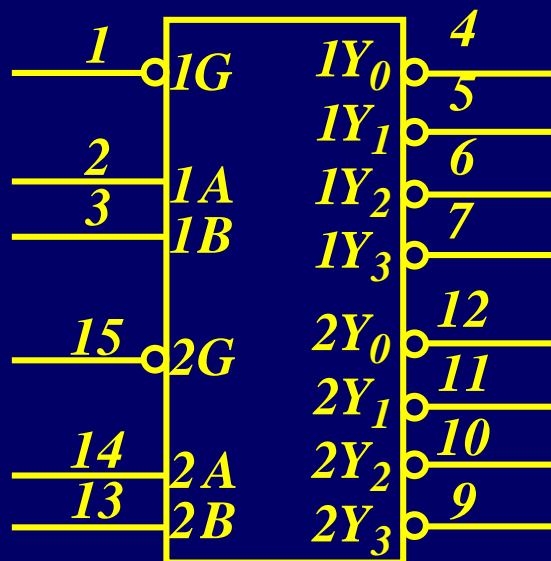
$/G$	B	A	$/Y_3$	$/Y_2$	$/Y_1$	$/Y_0$
1	d	d	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1

二. 常用 MSI 译码器

双2—4译码器 74LS139:将两个独立的2-4译码器封装在一个集成电路的芯片中。



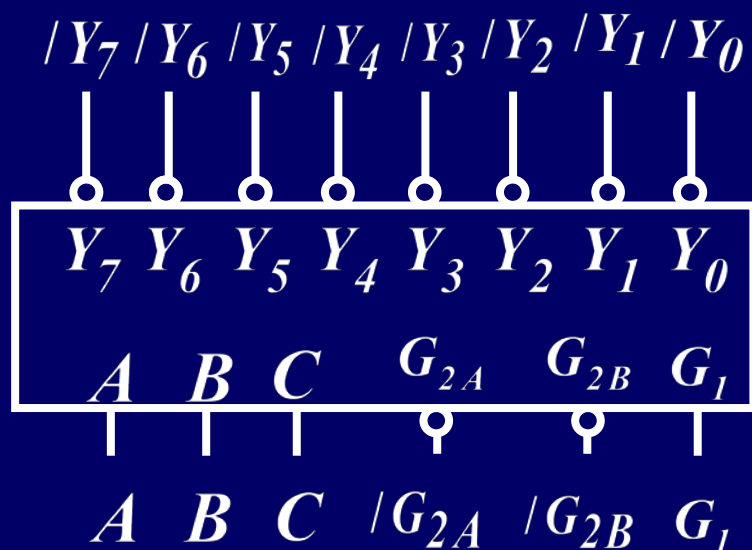
2-4译码器



74LS139

逻辑符号外边的数字是集成电路的管脚编号。8脚是地线（GND），16脚是电源线（V_{CC}）。

3-8译码器74LS138



74LS138逻辑图

1.符号框内的内部逻辑状态表达式

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

m_i 为输入变量 C、B、A 的最小项

2.内部信号与外部信号的关系

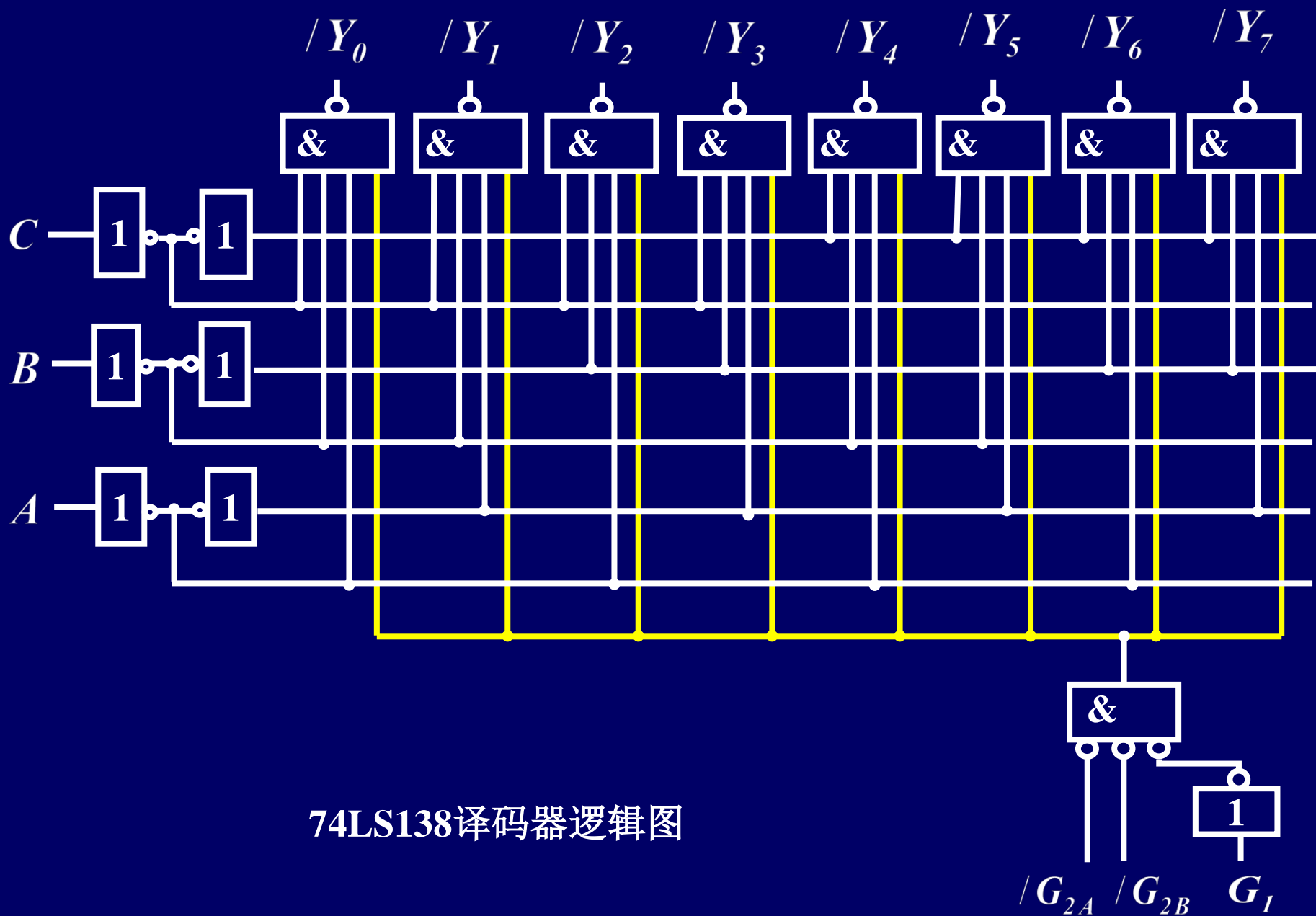
$$G_1 = G_1, G_{2A} = \overline{/G_{2A}},$$

$$G_{2B} = \overline{/G_{2B}}, Y_i = \overline{/Y_i}$$

3.外部逻辑状态表达式

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

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



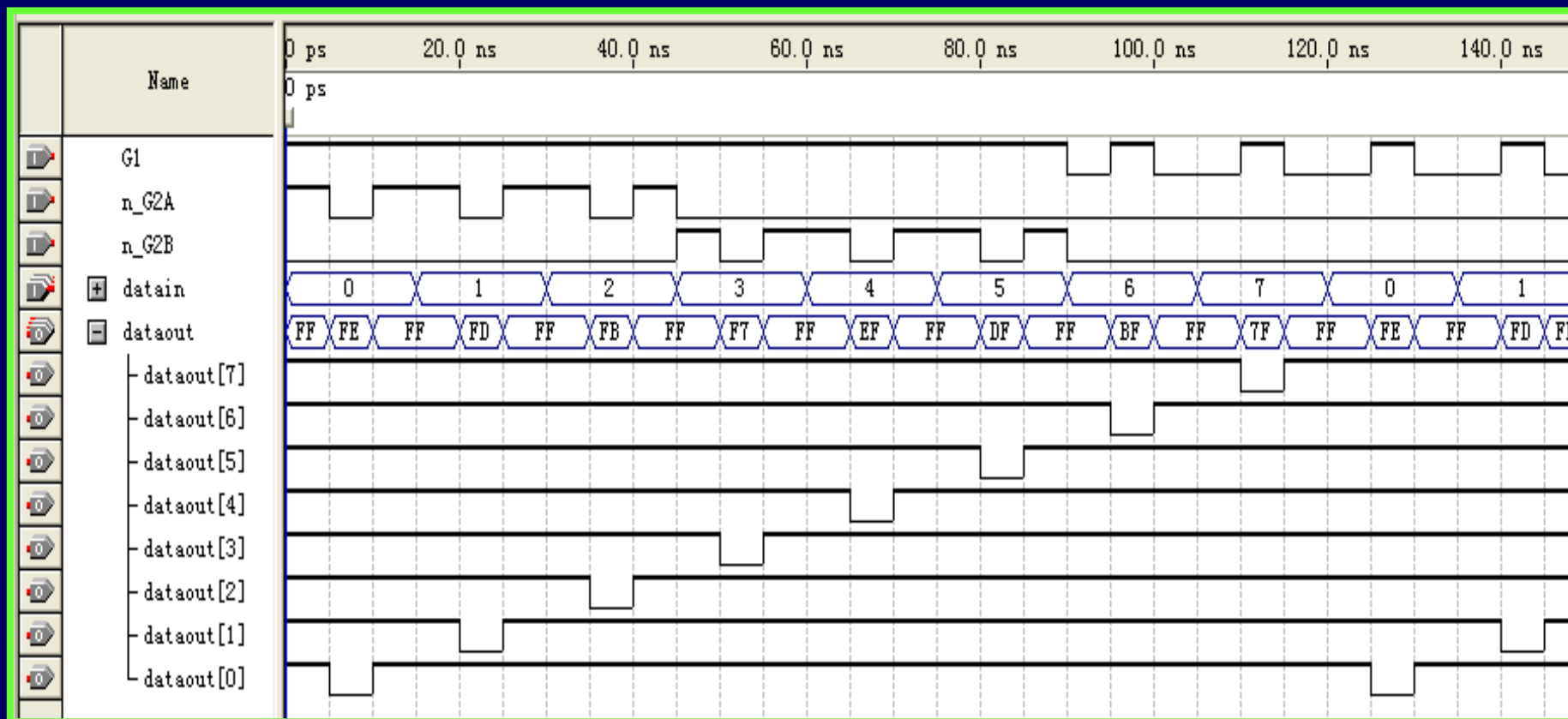
74LS138真值表

G_1	$/G_{2A}$	$/G_{2B}$	C	B	A	$/Y_7$	$/Y_6$	$/Y_5$	$/Y_4$	$/Y_3$	$/Y_2$	$/Y_1$	$/Y_0$
0	d	d	d	d	d	1	1	1	1	1	1	1	1
d	1	d	d	d	d	1	1	1	1	1	1	1	1
d	d	1	d	d	d	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1

当 $G_1=1$, $/G_{2A}+/G_{2B}=0$ 时: $/Y_i=M_i=\overline{m_i}$

应用Verilog HDL描述3-8译码器:

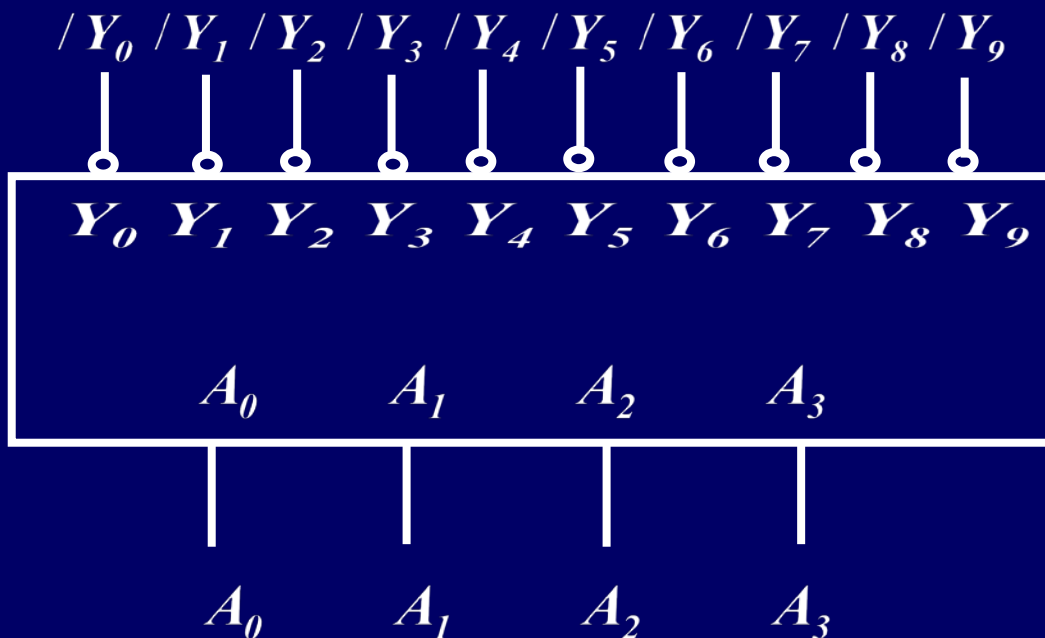
```
module decode_3_8_v(G1,n_G2A,n_G2B,datain,dataout);  
  input G1,n_G2A,n_G2B;  
  input[2:0] datain;  
  output[7:0] dataout;  
  assign dataout[0]=G1 && !n_G2A && !n_G2B?datain!='b000:1;  
  assign dataout[1]=G1 && !n_G2A && !n_G2B?datain!='b001:1;  
  assign dataout[2]=G1 && !n_G2A && !n_G2B?datain!='b010:1;  
  assign dataout[3]=G1 && !n_G2A && !n_G2B?datain!='b011:1;  
  assign dataout[4]=G1 && !n_G2A && !n_G2B?datain!='b100:1;  
  assign dataout[5]=G1 && !n_G2A && !n_G2B?datain!='b101:1;  
  assign dataout[6]=G1 && !n_G2A && !n_G2B?datain!='b110:1;  
  assign dataout[7]=G1 && !n_G2A && !n_G2B?datain!='b111:1;  
endmodule
```



8421 BCD码输入的4—10线译码器 74LS42

将十进制数字符号的二进制编码（BCD码）翻译成对应的十个输出信号的电路，称为二-十进制译码器。4个输入，10个输出。

74LS42逻辑符号



输入				输出									
A ₃	A ₂	A ₁	A ₀	/Y ₉	/Y ₈	/Y ₇	/Y ₆	/Y ₅	/Y ₄	/Y ₃	/Y ₂	/Y ₁	/Y ₀
0	0	0	0	1	1	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	1	1	0	1	1	1
0	1	0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	1	1	1	1	1	0	1	1	1	1	1
0	1	1	0	1	1	1	0	1	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1	1	1	1
1	0	0	0	1	0	1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	1	1	1	1	1	1	1
1010				1	1	1	1	1	1	1	1	1	1
伪 码									
									
	1111			1	1	1	1	1	1	1	1	1	1

由74LS42的功能表可看出，当输入出现1010~1111六种伪码时，输出均为无效状态“1”。即该译码器拒绝伪码。

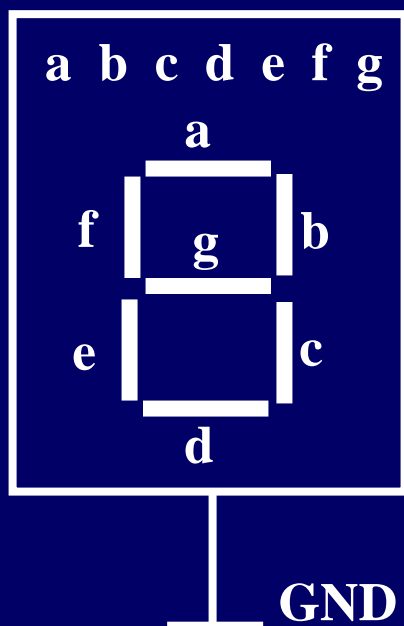
// 8421码4-10线译码器的Verilog HDL描述

```
module decoder3_8 (en, in, out) ;  
    input  [3:0] in ;  
    input   en;      // ?  
    output [9:0] out ; // ?  
    reg    [9:0] out ;  
    always @ (en or in)  
        if ( en )      // ?  
            case ( in )  
                4'b0000 : out=10'b1111111110;  
                4'b0001 : out=10'b1111111101;  
                .....  
                4'b1001 : out=10'b0111111111;  
                default : out=10'b1111111111;  
            endcase  
        else out=10'b1111111111;  
endmodule
```

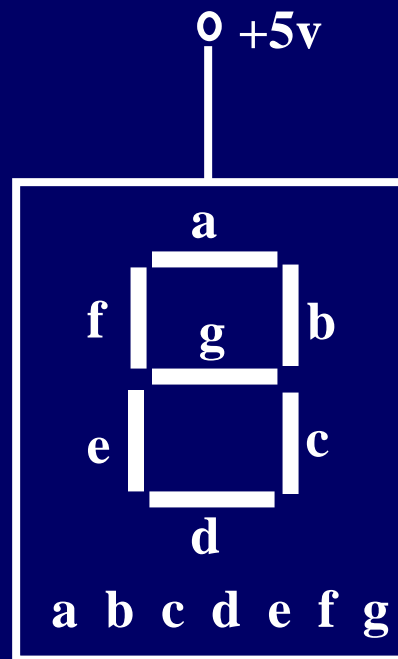
余3码4 —10线
译码器的描述？

BCD—七段显示译码器

七段LED显示器，用七个发光二极管做成a、b、c、d、e、f、g共七个笔划段，并分为共阴极与共阳极两种。

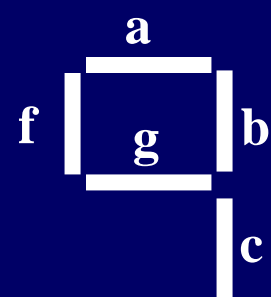
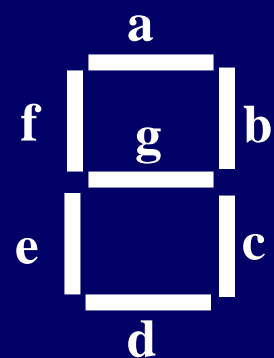
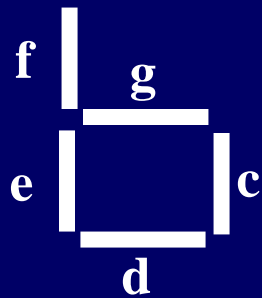
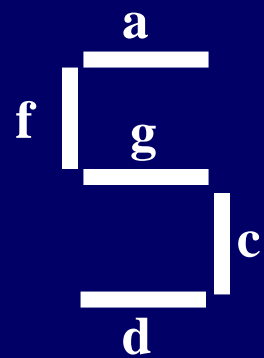
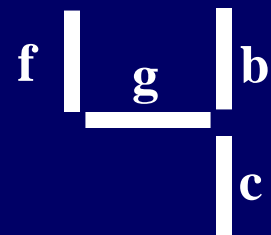
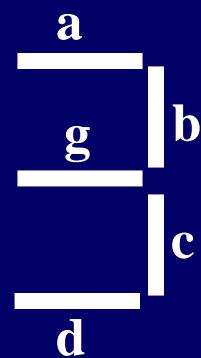
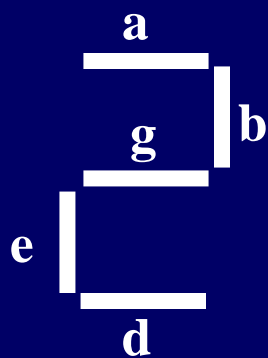
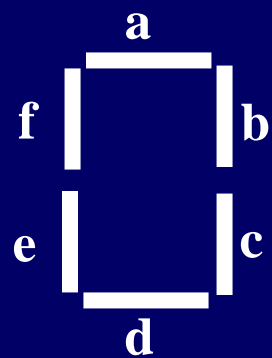
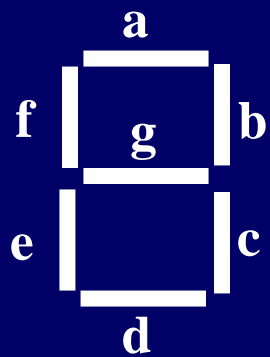


共阴极



共阳极

通过“点亮”某些笔划段，可使其显示出不同数字或字符。



//8421BCD码—七段显示译码器的Verilog HDL描述

```
module decode4-7 ( decodeout , indec ) ;
```

```
    output [6:0] decodeout ;           //输出到共阴七段数码管
```

```
    input [3:0] indec ;               //8421码
```

```
    reg [6:0] decodeout ;
```

```
    always @ ( indec )
```

```
        begin    case (indec)
```

```
            4'd0 : decodeout = 7'b1111110 ;
```

```
            4'd1 : decodeout = 7'b0110000 ;
```

```
            4'd2 : decodeout = 7'b1101101 ;
```

```
            4'd3 : decodeout = 7'b1111001 ;
```

```
            4'd4 : decodeout = 7'b0110011 ;
```

```
            4'd5 : decodeout = 7'b1011011 ;
```

```
            4'd6 : decodeout = 7'b1011111 ;
```

```
            4'd7 : decodeout = 7'b1110000 ;
```

```
            4'd8 : decodeout = 7'b1111111 ;
```

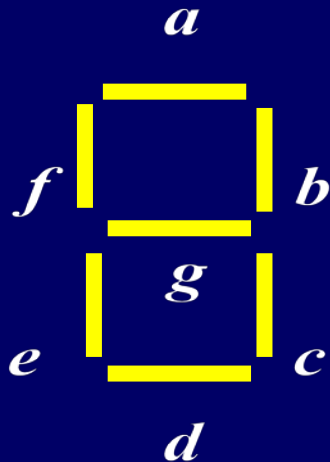
```
            4'd9 : decodeout = 7'b1110011 ;
```

```
            default : decodeout = 7'bx ;
```

```
        endcase
```

```
    end
```

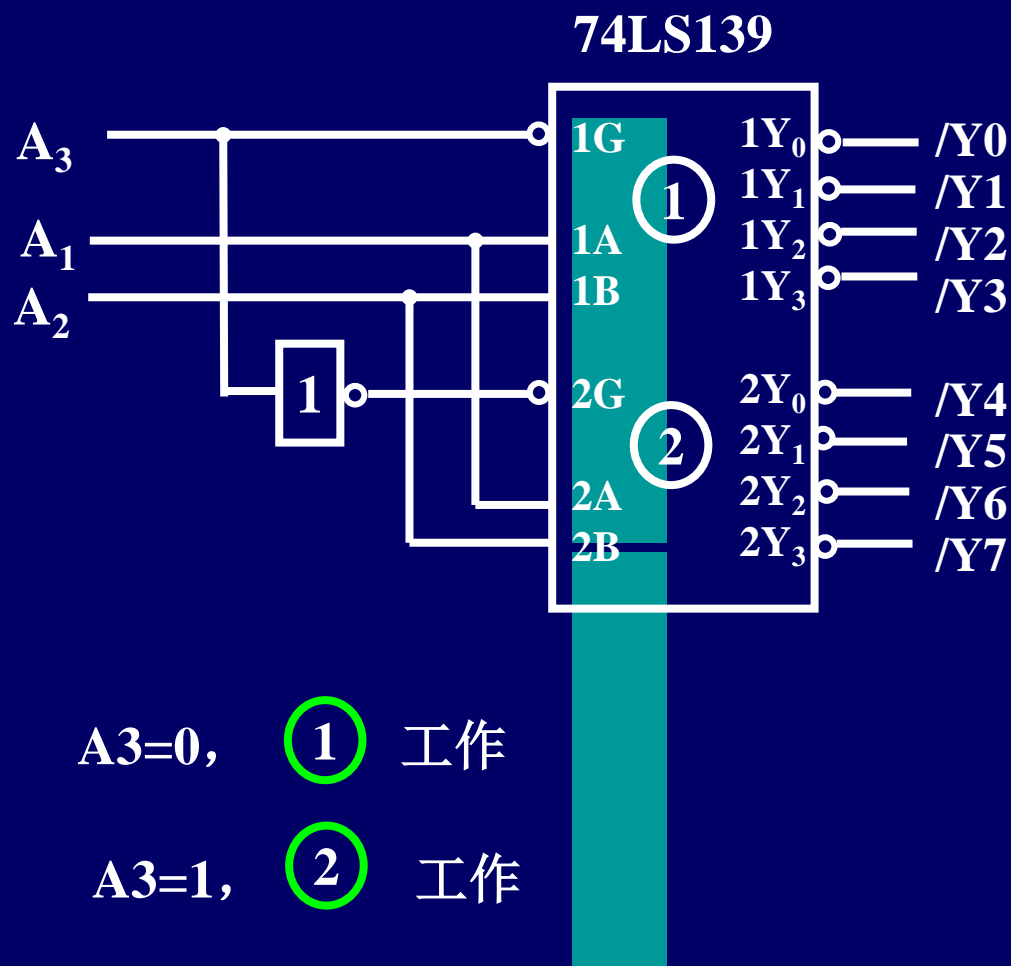
```
endmodule
```



共阳极？

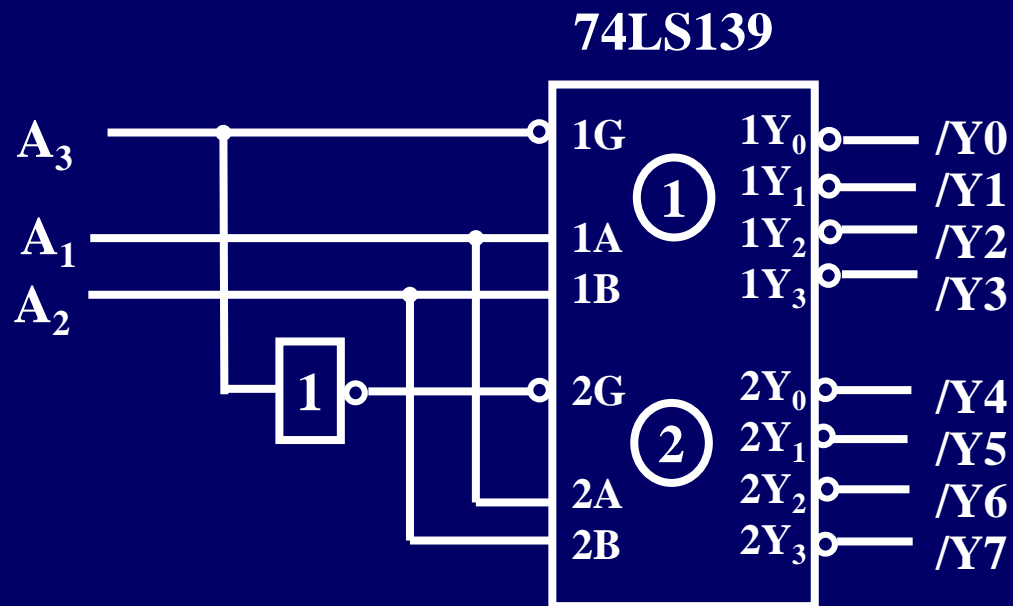
三. 二进制译码器的级联（扩展）

例1：分析图示电路



A3	A2	A1	/Y0	/Y1	/Y2	/Y3	/Y4	/Y5	/Y6	/Y7
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

74LS139中的两个 2-4 译码器级联扩展成 3-8 译码器。

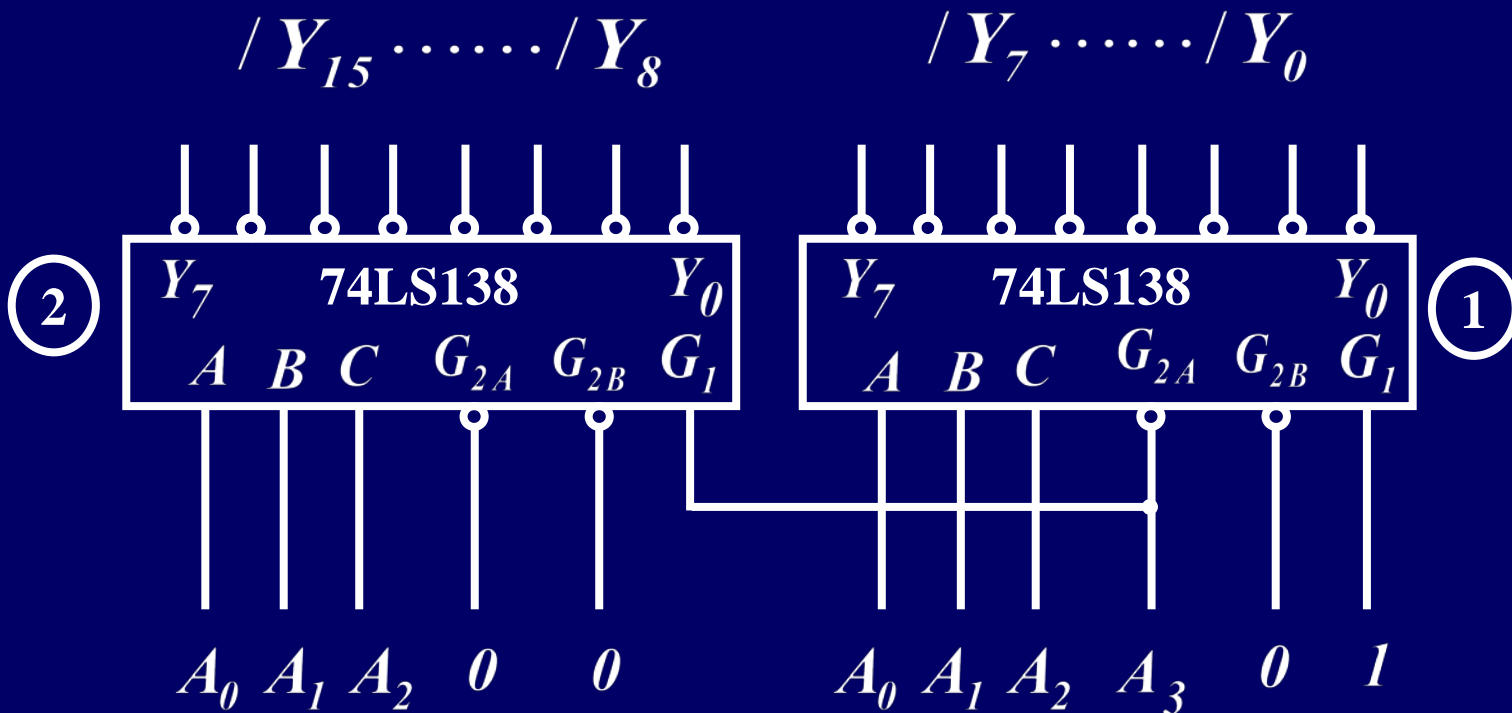


级联扩展原理:

低位变量共享，注意变量连接顺序；
利用高位变量（ A_3 ）和译码器的使能端进行扩展；
标定输出顺序。

例2：用两片74LS138译码器构成4-16译码器。

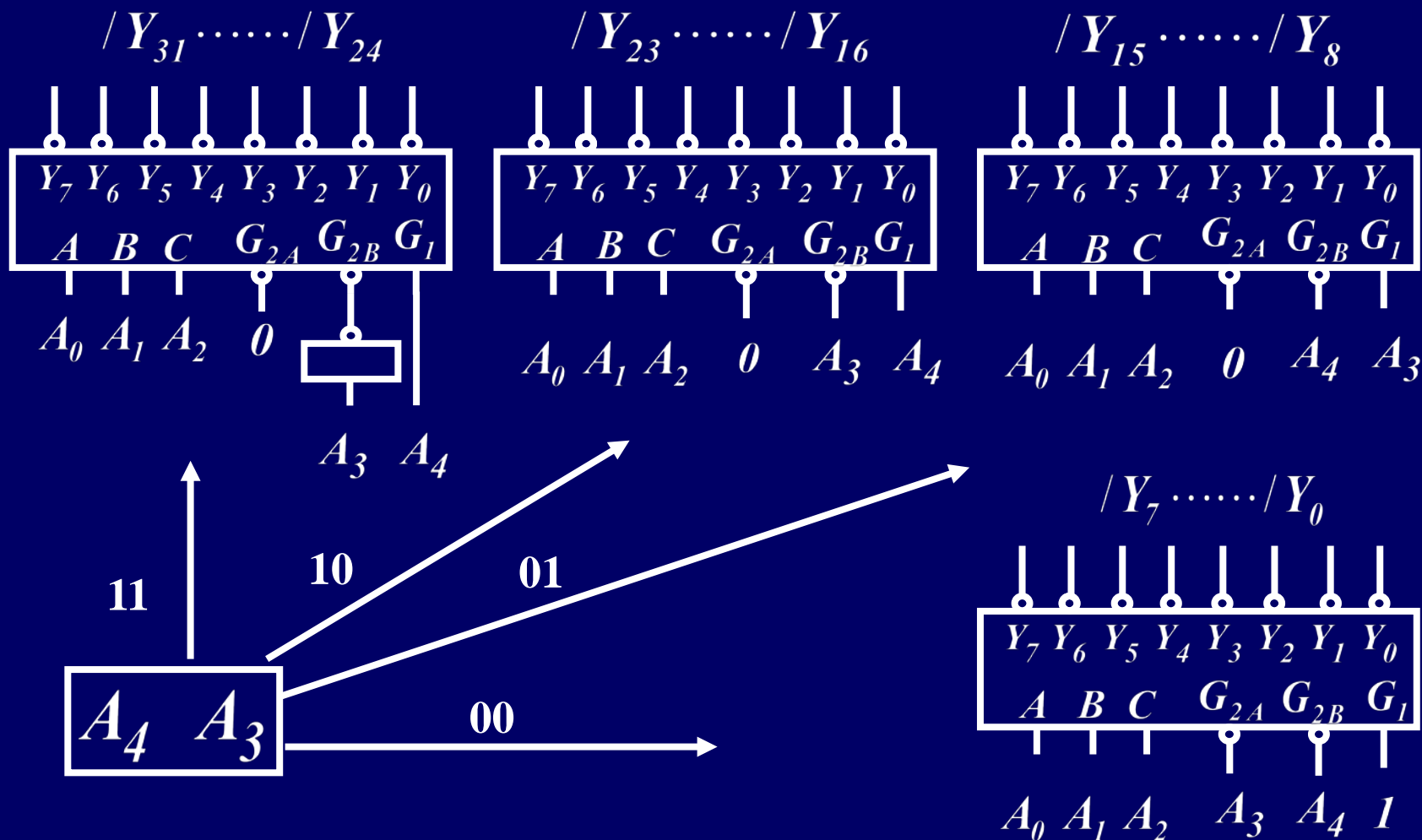
解：设输入变量为 $A_3A_2A_1A_0$ ， $A_2A_1A_0$ 共享，用高位变量 A_3 控制使能端，实现扩展。注意连接时的变量顺序。



高位变量 A_3 作为使能信号，“0”时①工作；“1”时②工作。

例3：用四片74LS138译码器和最少的其它电路实现5-32译码。

解：设五个输入变量是 A_4 、 A_3 、 A_2 、 A_1 、 A_0 ，利用 A_4 、 A_3 的组合及多个使能端的有效特性进行扩展。注意变量顺序。

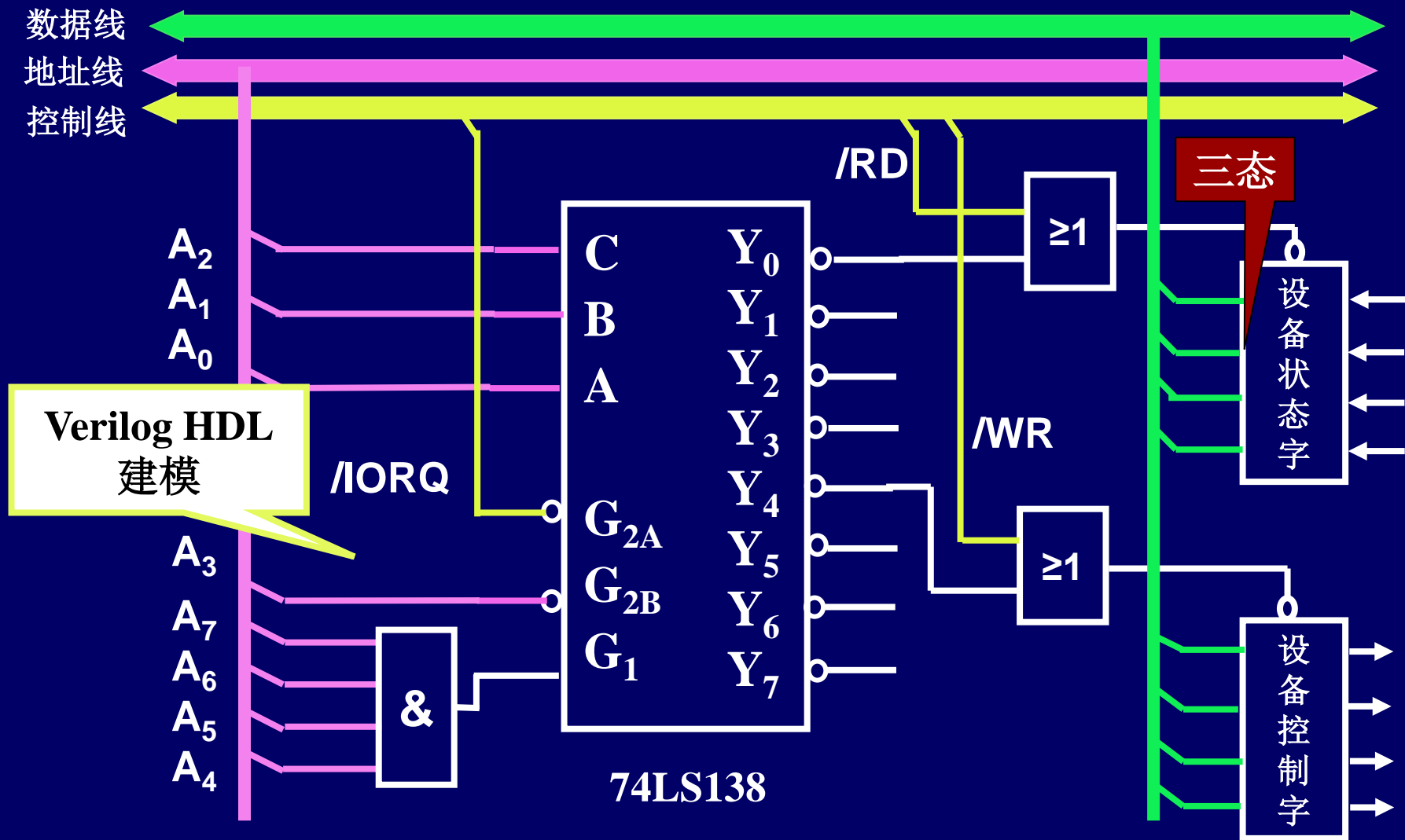


四. 二进制译码器的应用

数字系统中的三总线：数据、地址、控制

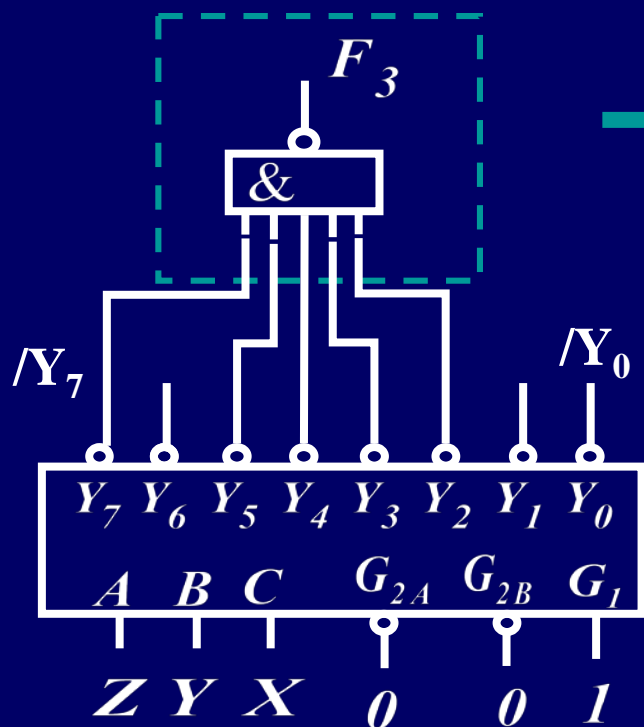
译码器在数字系统中常用于地址译码，以选通（选中）设备，在控制信号的作用下，完成读写操作。

例1：在某数字（微机）系统中，有如下电路，请分析其功能。



当/IORQ、/RD有效，且 A7...A0=11110000时，读状态字；
当/IORQ、/WR有效，且 A7...A0=11110100时，写控制字。

例2：写出图示电路（74LS138）中 F_3 的逻辑表达式。



$$F_3 = \overline{Y_7} \cdot \overline{Y_5} \cdot \overline{Y_4} \cdot \overline{Y_3} \cdot \overline{Y_2}$$

根据译码器 $\overline{Y_i} = m_i$

关于 x, y, z

$$\begin{aligned} F_3(x, y, z) &= \overline{m_7} \cdot \overline{m_5} \cdot \overline{m_4} \cdot \overline{m_3} \cdot \overline{m_2} \\ &= m_7 + m_5 + m_4 + m_3 + m_2 \end{aligned}$$

进一步：最简与或式、最简与非式、
最简或与式、最简或非式、
最简与或非式

结论：采用译码器、逻辑门可以实现逻辑函数（逻辑功能）。

采用二进制译码器实现组合逻辑函数

n — 2^n 译码器的输出，对应 n 个变量的全部最小项。所以，只要得到逻辑函数的最小项表达式，就可采用译码器和适当逻辑门实现之。

例3： 用一片74LS138译码器再加最少的与非门实现：

$$F(x,y,z)=\sum m(2,3,4,5,7)$$

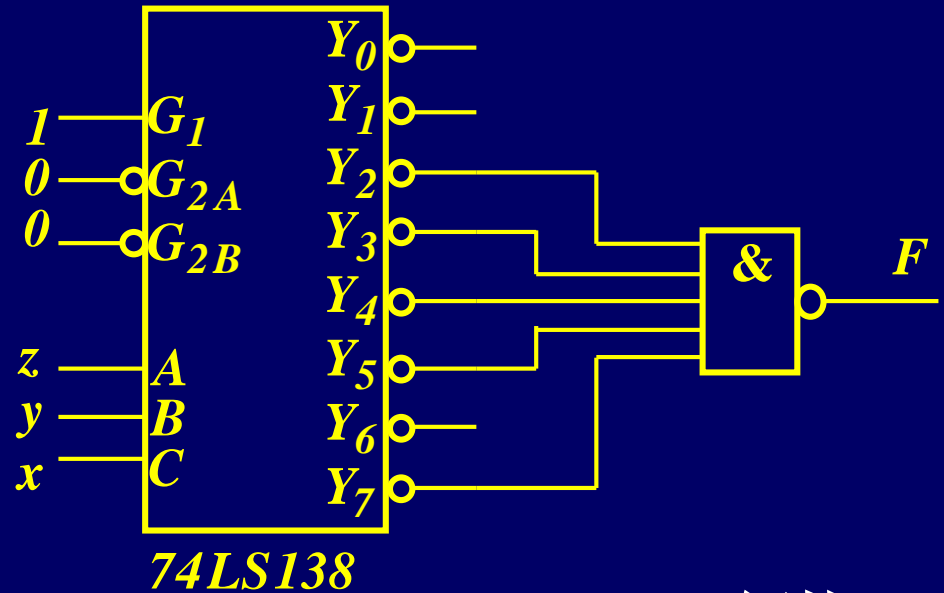
解： 由题可以写出

$$F(x,y,z)=Y_2+Y_3+Y_4+Y_5+Y_7=\overline{\overline{Y_2}\cdot\overline{Y_3}\cdot\overline{Y_4}\cdot\overline{Y_5}\cdot\overline{Y_7}}$$

当使能端有效时：

$$/Y_i=M_i=\overline{m_i}$$

注意这里
以 x 为高
位。



解毕。

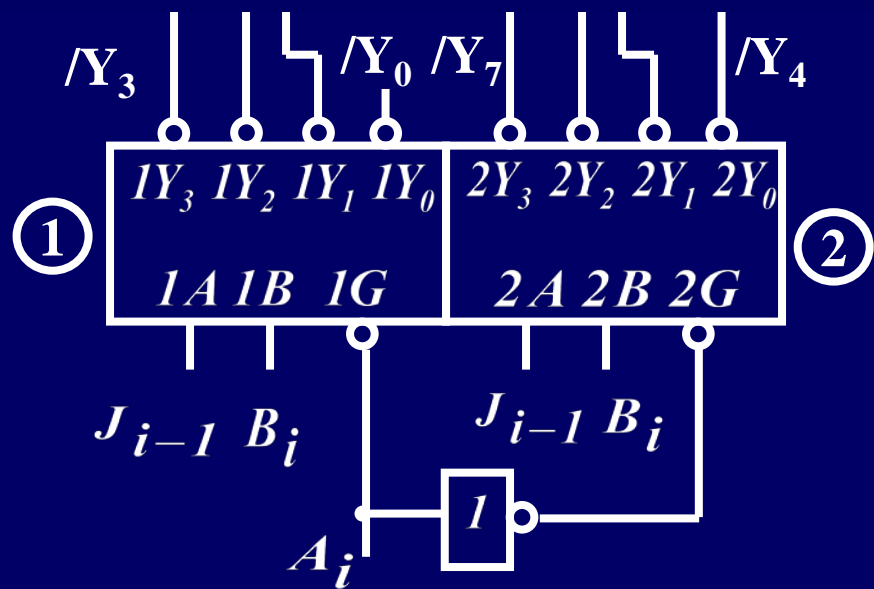
例3：用74LS139译码器和适当与非门实现全加器。

全加器真值表：

A_i	B_i	J_{i-1}	H_i	J_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

$$H_i = \overline{A_i} \overline{B_i} J_{i-1} + \overline{A_i} B_i \overline{J_{i-1}} + A_i \overline{B_i} \overline{J_{i-1}} + A_i B_i J_{i-1}$$

$$J_i = \overline{A_i} B_i J_{i-1} + A_i \overline{B_i} J_{i-1} + A_i B_i \overline{J_{i-1}} + A_i B_i J_{i-1}$$



作业10:

4.15

4.16

4.25 (2)