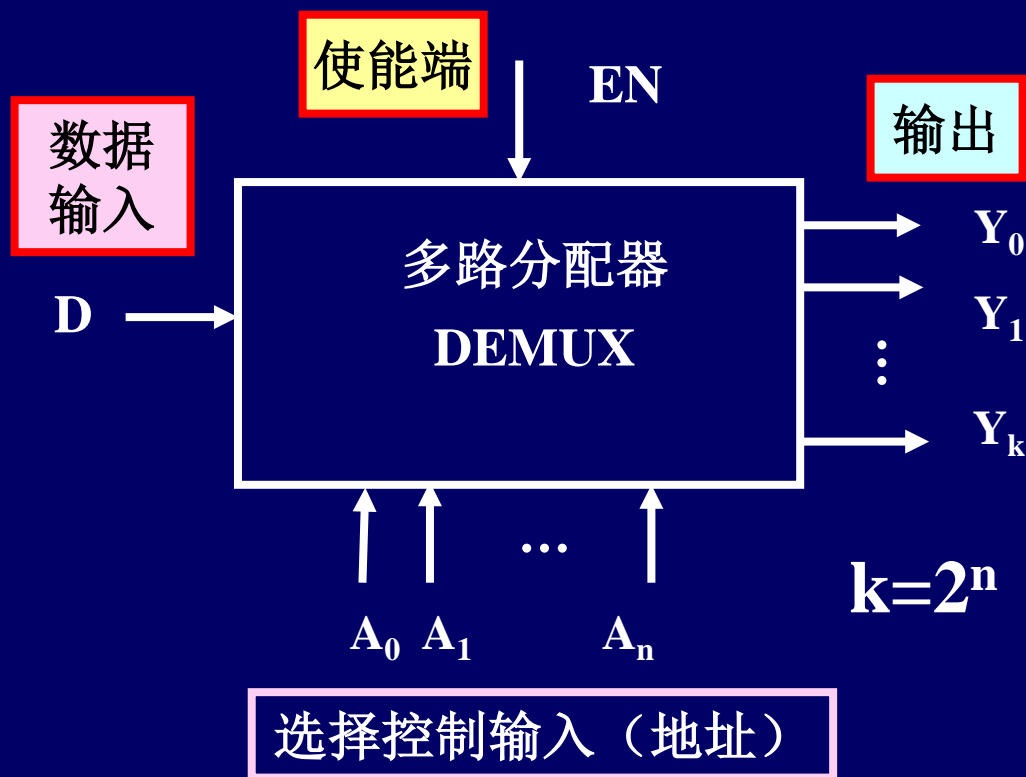


### 3 多路分配器

#### (1) 多路分配器 (Demultiplexer)工作原理

又称数据分配器，常用**DEMUX**表示。单输入，多输出。

多路分配器的功能是根据地址译码的指向，将输入数据**D**的逻辑值分配到相应的输出线上去。



当EN有效时

$$Y_0 = D \cdot m_0$$

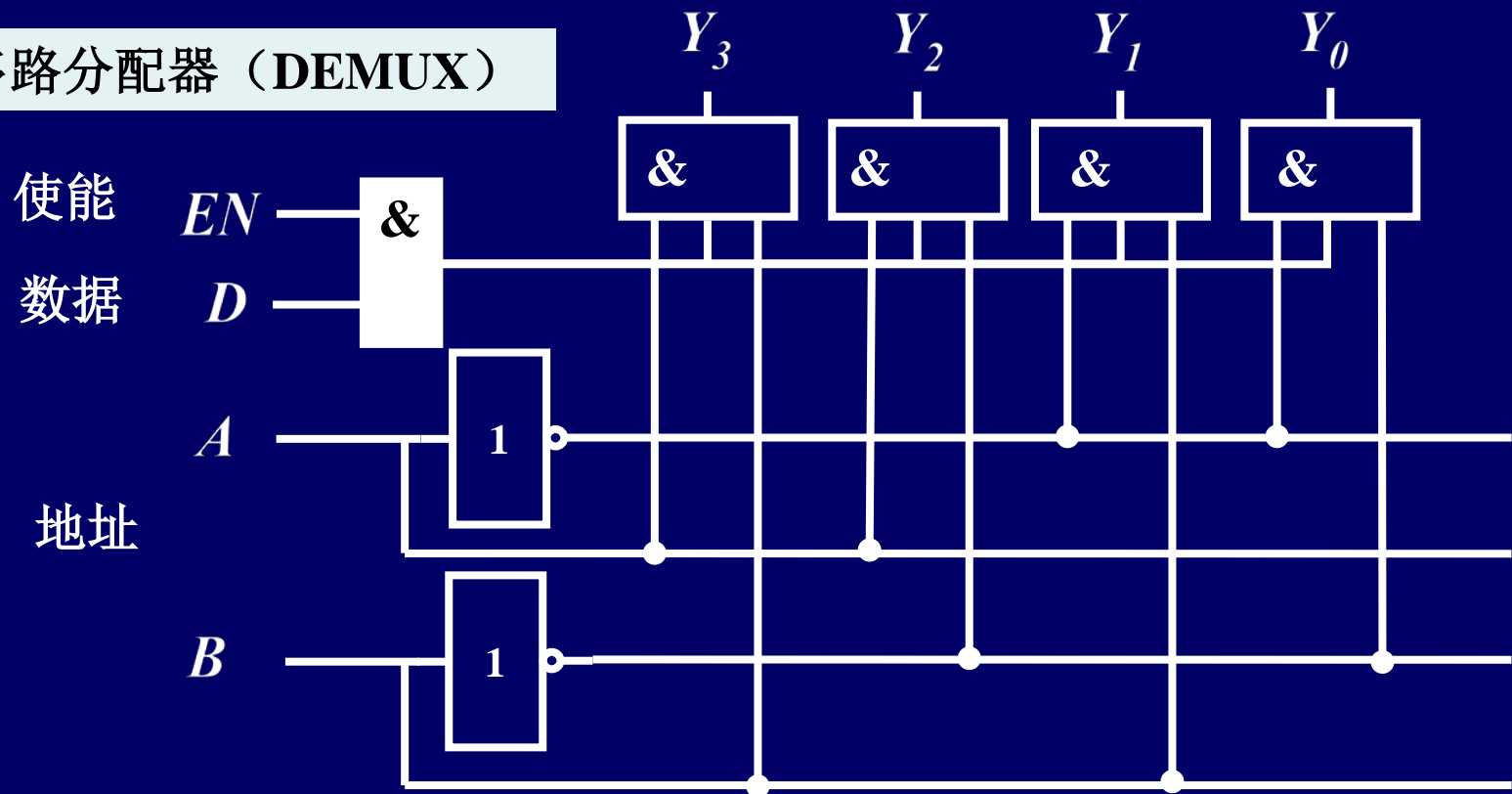
$$Y_1 = D \cdot m_1$$

$\vdots$

$$Y_k = D \cdot m_k$$

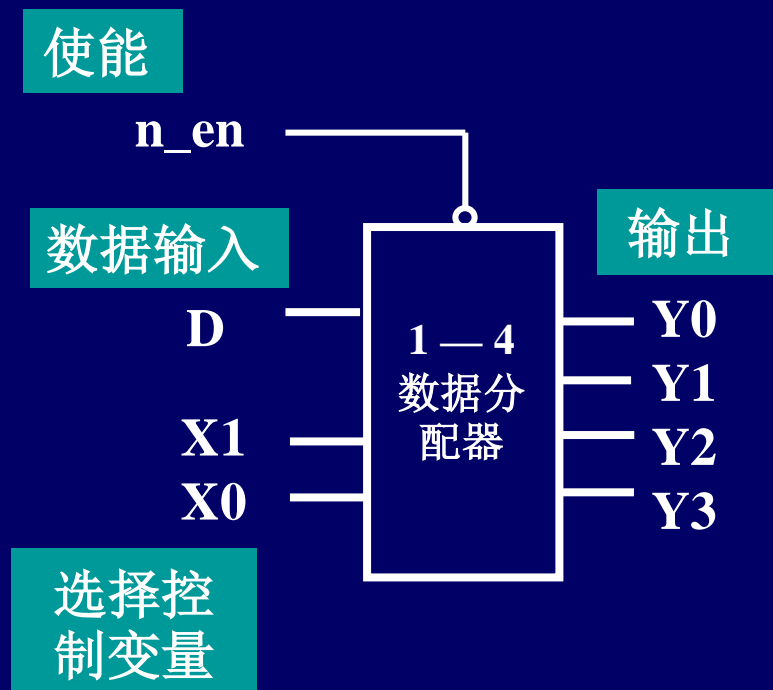
$m_k$  是  $A_n \dots A_1 A_0$  的最小项

# 1—4 多路分配器 (DEMUX)



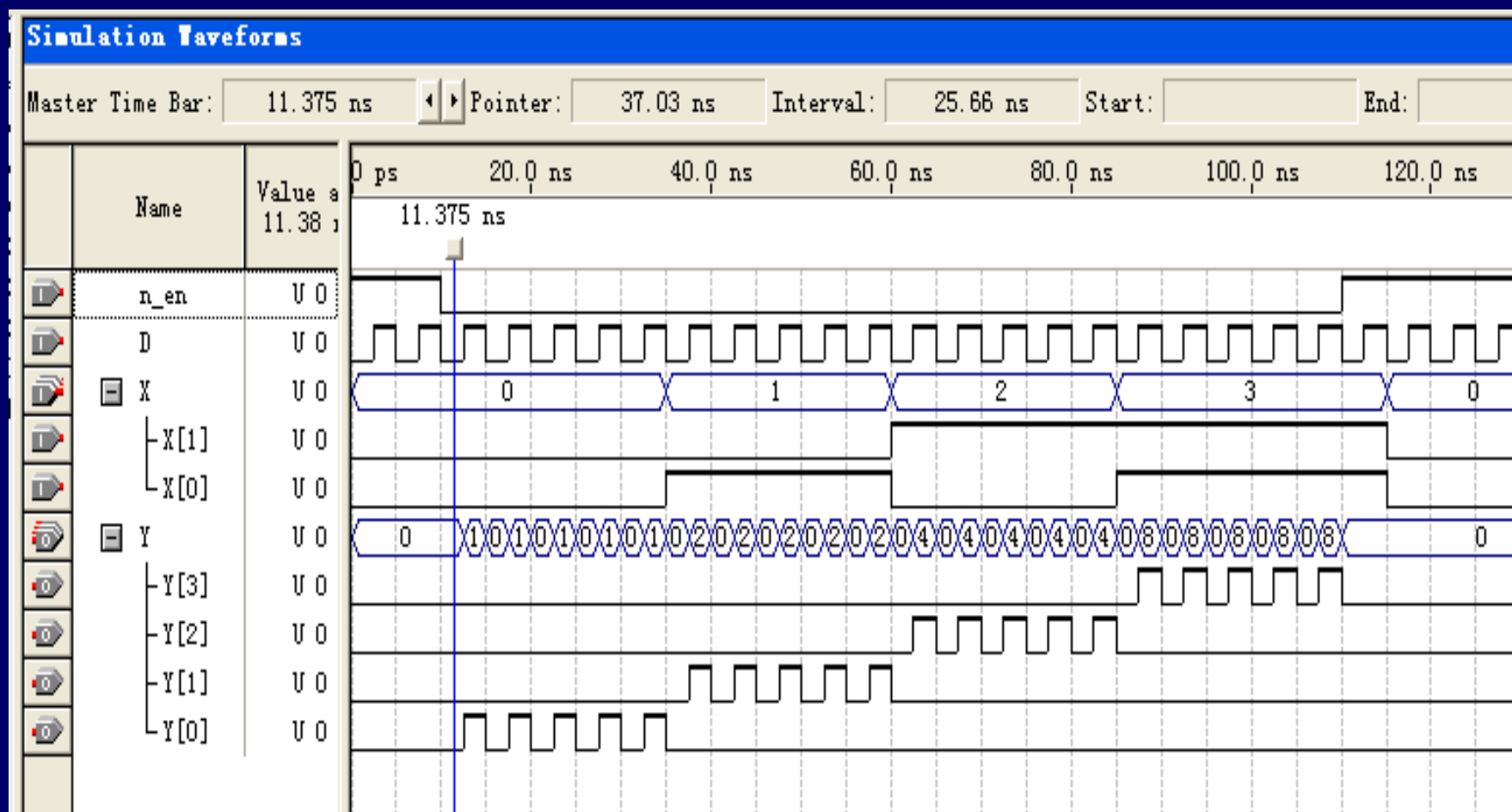
$A$	$B$	$EN$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
$\times$	$\times$	$0$	$0$	$0$	$0$	$0$
$0$	$0$	$1$	$0$	$0$	$0$	$D$
$0$	$1$	$1$	$0$	$0$	$D$	$0$
$1$	$0$	$1$	$0$	$D$	$0$	$0$
$1$	$1$	$1$	$D$	$0$	$0$	$0$

## (2) 1—4 数据分配器的Verilog HDL模型

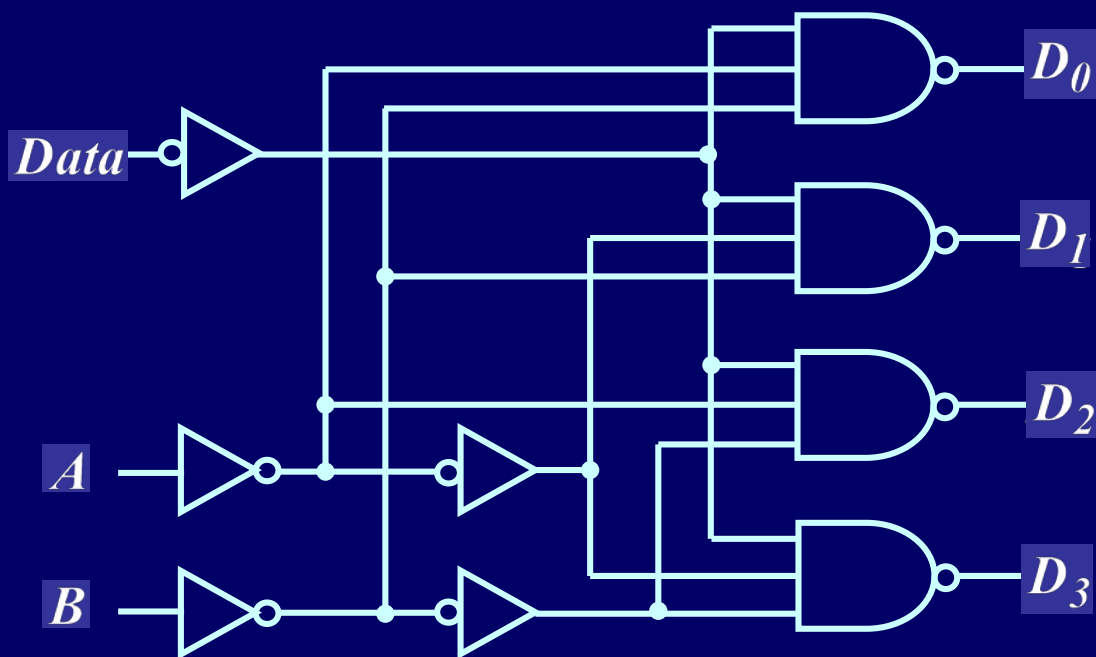


```
module demux1_4 (n_en, D, X, Y);  
    input n_en, D;  
    input [1:0] X;  
    output [3:0] Y;  
    reg [3:0] Y;  
    always @ (n_en or D or X)  
        if ( ~n_en )  
            case ( X )  
                2'b00 : Y[0]=D;  
                2'b01 : Y[1]=D;  
                2'b10 : Y[2]=D;  
                2'b11 : Y[3]=D;  
            endcase  
        else Y=4'b0000;  
endmodule
```

## 1—4数据分配器的仿真验证



•译码器用作数据分配器（Demultiplexer）。



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



$B$	$A$	$D_3$	$D_2$	$D_1$	$D_0$
$0$	$0$	$1$	$1$	$1$	$Data$
$0$	$1$	$1$	$1$	$Data$	$1$
$1$	$0$	$1$	$Data$	$1$	$1$
$1$	$1$	$Data$	$1$	$1$	$1$



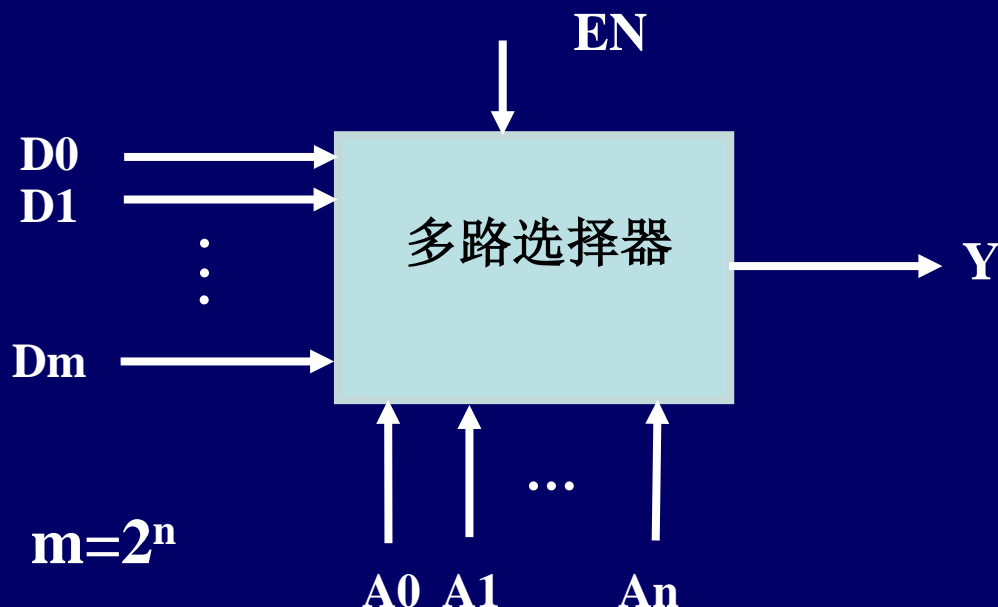
$B$	$A$	$/Y_3$	$/Y_2$	$/Y_1$	$/Y_0$
$0$	$0$	$1$	$1$	$1$	$/G$
$0$	$1$	$1$	$1$	$/G$	$1$
$1$	$0$	$1$	$/G$	$1$	$1$
$1$	$1$	$/G$	$1$	$1$	$1$

## 4 多路选择器

多路选择器（Multiplexers）又称数据选择器、多路开关，常记为MUX。

它是一种多路输入、单路输出的组合逻辑电路。记为  $n/1$  或  $n-1$

逻辑功能：当使能端 EN 有效时，在选择控制变量的控制下，从多路输入数据里选中一路送到输出端。



当使能有效时

$$Y = \sum_{i=0}^{2^n-1} D_i m_i$$

$n$ 个选择控制变量的每种取值组合对应选中 $m=2^n$ 路输入数据中的一路送到输出端。

# 一. 常用多路选择器的设计

## 1. 8 选 1 多路选择器

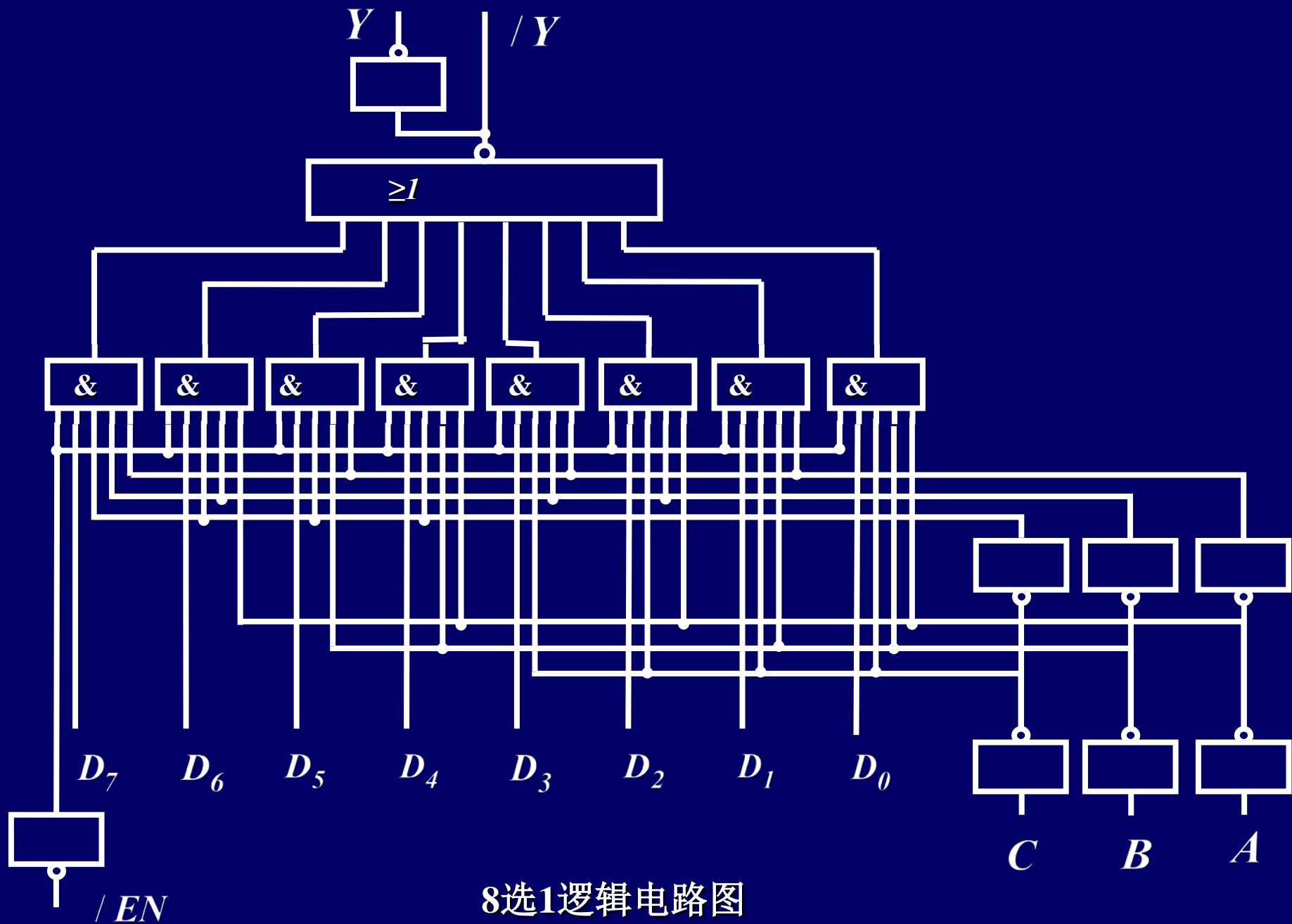
功能表

输入				输出	
$\overline{EN}$	$C$	$B$	$A$	$Y$	$\overline{Y}$
1	$d$	$d$	$d$	0	1
0	0	0	0	$D_0$	$\overline{D_0}$
0	0	0	1	$D_1$	$\overline{D_1}$
0	0	1	0	$D_2$	$\overline{D_2}$
0	0	1	1	$D_3$	$\overline{D_3}$
0	1	0	0	$D_4$	$\overline{D_4}$
0	1	0	1	$D_5$	$\overline{D_5}$
0	1	1	0	$D_6$	$\overline{D_6}$
0	1	1	1	$D_7$	$\overline{D_7}$

使能有效时  
Y的输出逻辑表达式?

使能有效时  
Y的卡诺图?

<div>BA</div> <div>C</div>		00	01	11	10
<div>0</div>		<b>D<sub>0</sub></b>	<b>D<sub>1</sub></b>	<b>D<sub>3</sub></b>	<b>D<sub>2</sub></b>
		<b>D<sub>4</sub></b>	<b>D<sub>5</sub></b>	<b>D<sub>7</sub></b>	<b>D<sub>6</sub></b>
1					

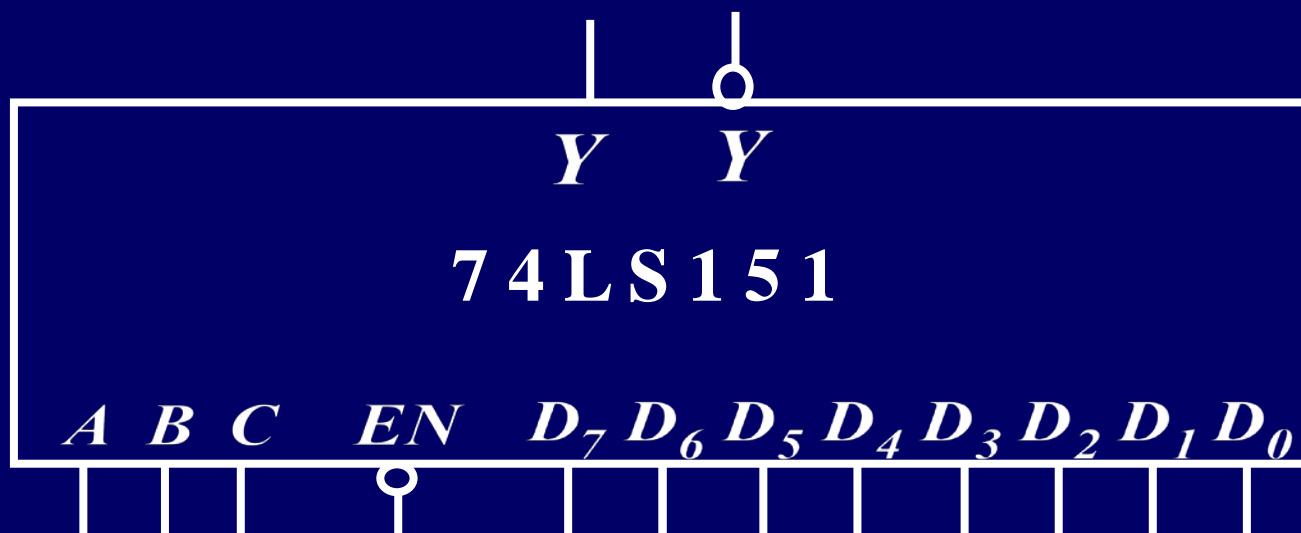


8选1逻辑电路图



电路封装，逻辑符号

互补输出



选择控制变量输入端，C为高位。

低有效使能输入端

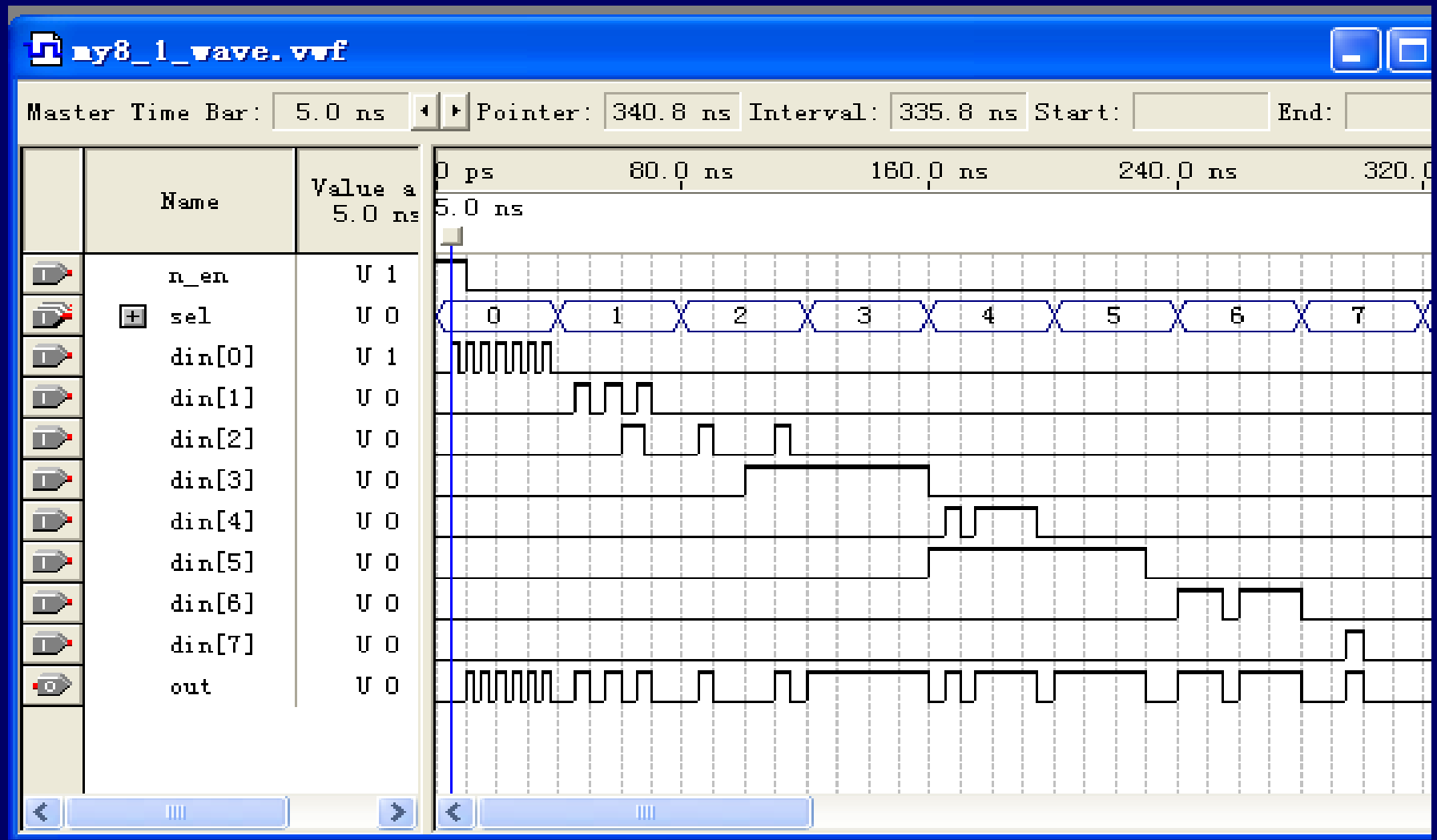
8个数据输入端，其下标对应选择控制变量C、B、A状态组合的十进制值。

## 8选1的Verilog HDL模型

abc my8\_1\_wave.v\*

```
1 module my8_1_wave(din,sel,n_en,out);
2     input [7:0] din;
3     input [2:0] sel;
4     input      n_en;
5     output     out;
6     reg        out;
7     always @ (din or sel or n_en)
8         begin
9             if (!n_en)
10                case (sel)
11                    3'b000: out=din[0];
12                    3'b001: out=din[1];
13                    3'b010: out=din[2];
14                    3'b011: out=din[3];
15                    3'b100: out=din[4];
16                    3'b101: out=din[5];
17                    3'b110: out=din[6];
18                    3'b111: out=din[7];
19                    default : out=1'b0;
20                endcase
21            else out=1'b0;
22        end
23 endmodule
24
```

# 8选1的功能仿真



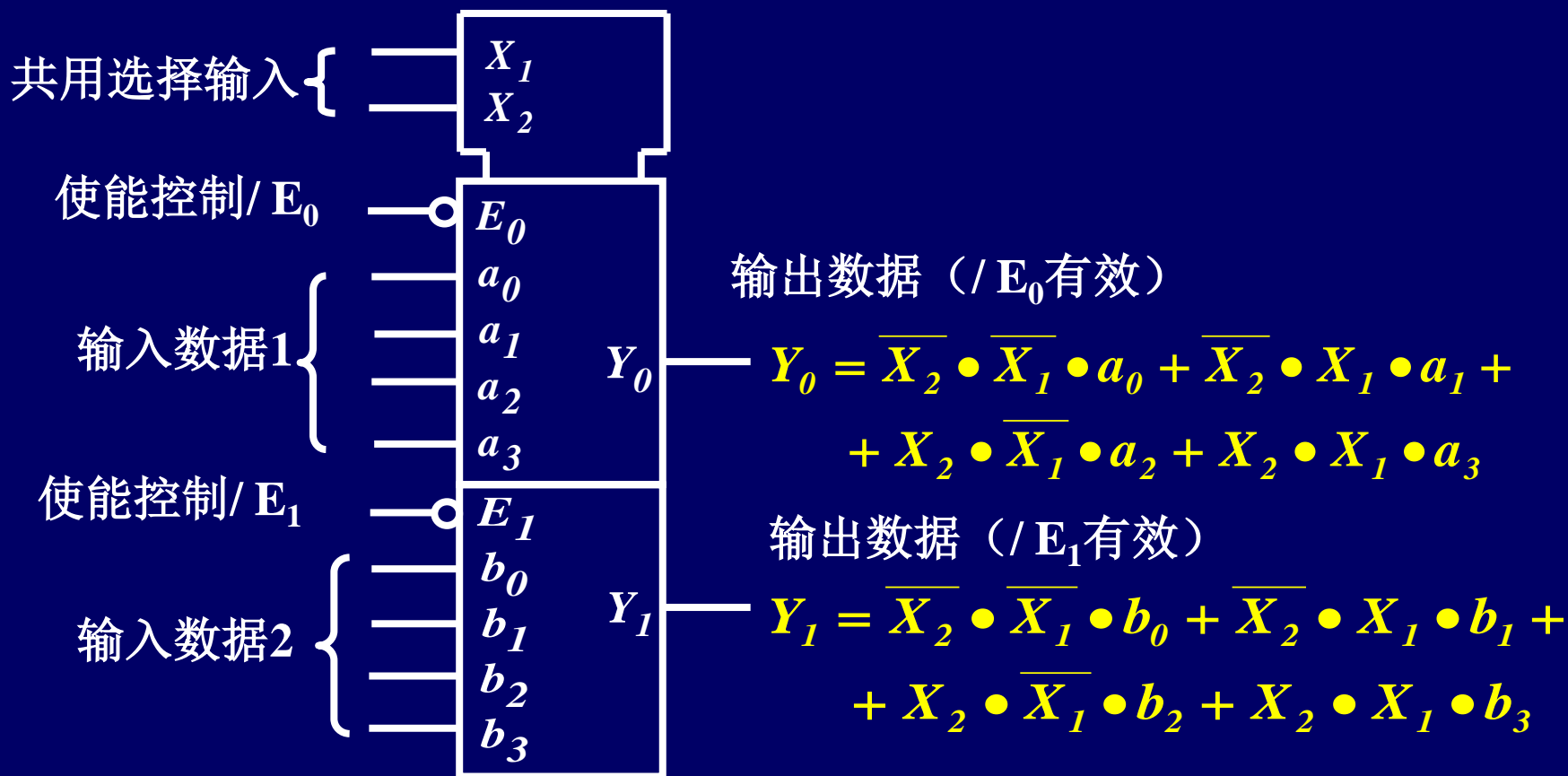
## 2. 分析 74LS153的逻辑功能，建立Veriloh HDL模型

74LS153功能表

输入			输出
$/E_0$	$X_2$	$X_1$	$Y_0$
1	d	d	0
0	0	0	$a_0$
0	0	1	$a_1$
0	1	0	$a_2$
0	1	1	$a_3$

输入			输出
$/E_1$	$X_2$	$X_1$	$Y_1$
1	d	d	0
0	0	0	$b_0$
0	0	1	$b_1$
0	1	0	$b_2$
0	1	1	$b_3$

## 74LS153逻辑符号



功能：具有共用选择输入端的双4选1

## 74LS153的Verilog HDL描述

```
module my153(x2, x1, n_e0, n_e1, a, b, y0, y1) ;  
    input  x2, x1, n_e0, n_e1 ;  
    input  [3:0] a,b;  
    output y0, y1;  
    reg    y0, y1;  
    always @ (x2 or x1 or n_e0 or n_e1 or a or b)  
    begin  
        if (! n_e0)  
            case ({x2,x1})  
                2'b00 : y0=a[0];  
                2'b01 : y0=a[1];  
                2'b10 : y0=a[2];  
                2'b11 : y0=a[3];  
            endcase  
        else y0=1'b0;
```

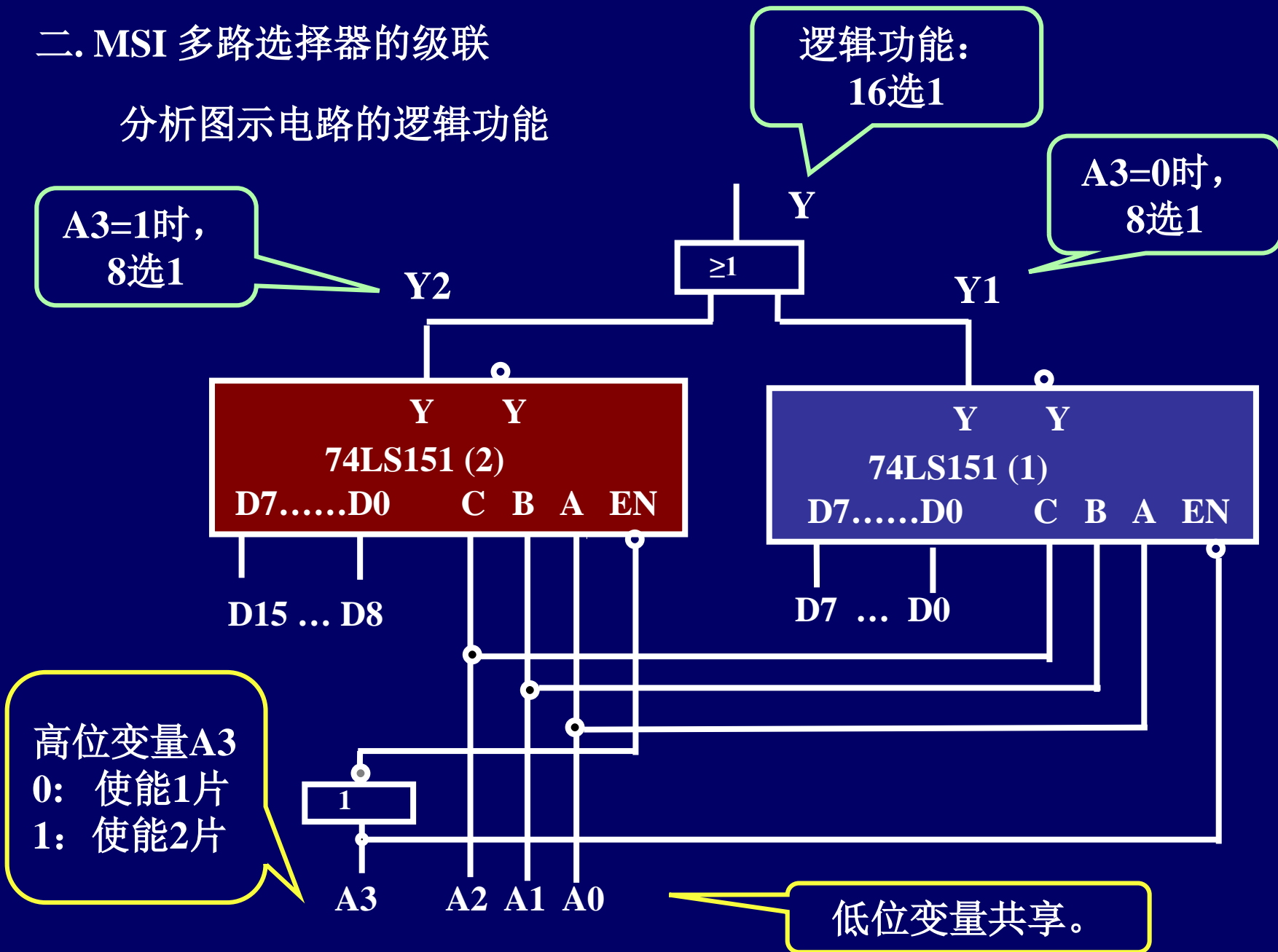
后续

续前

```
    if (n_e1==0)
        case ({x2,x1})
            2'b00 : y1=b[0];
            2'b01 : y1=b[1];
            2'b10 : y1=b[2];
            2'b11 : y1=b[3];
        endcase
    else y1=1'b0;
end
endmodule
```

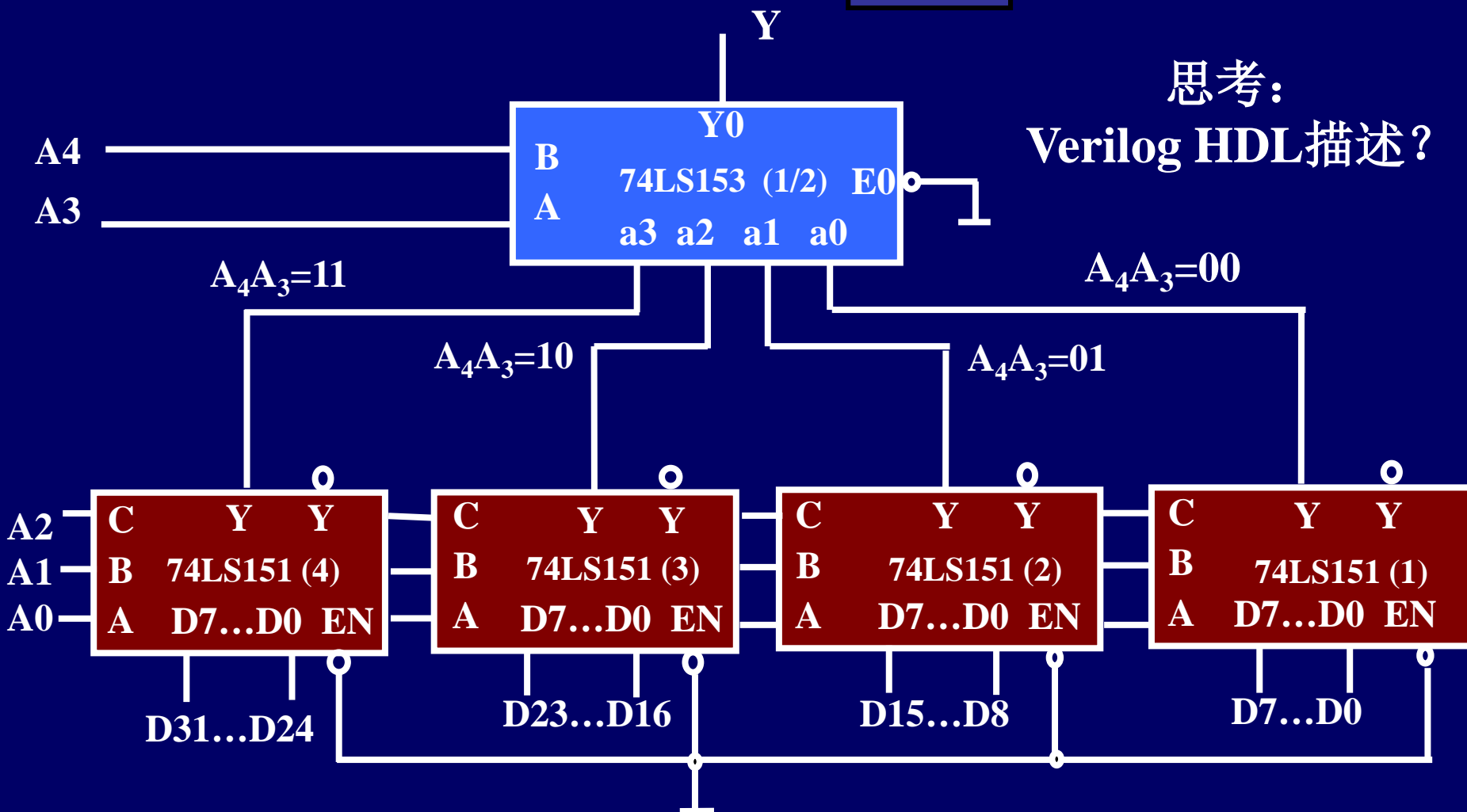
## 二. MSI 多路选择器的级联

分析图示电路的逻辑功能





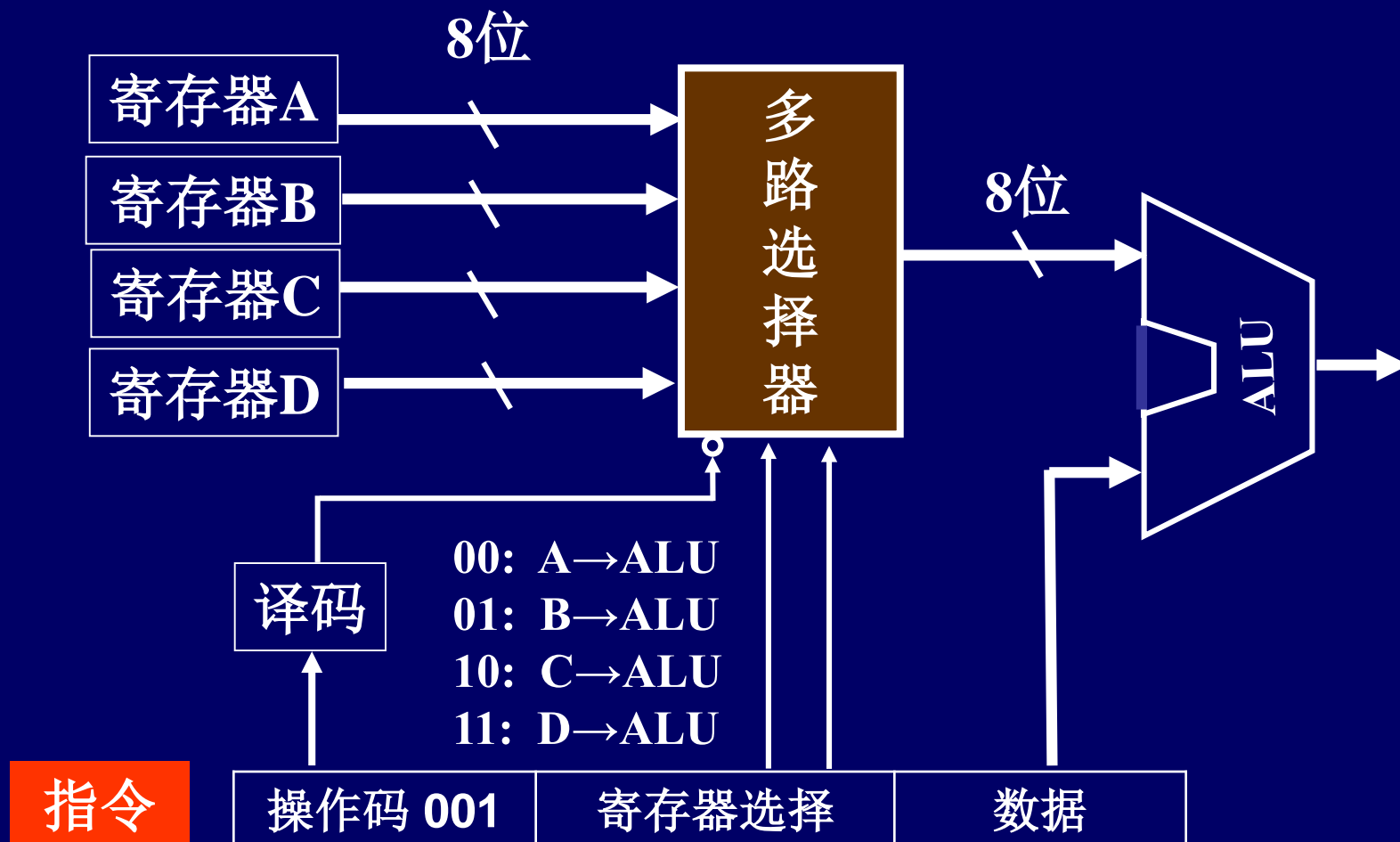
## 32选1



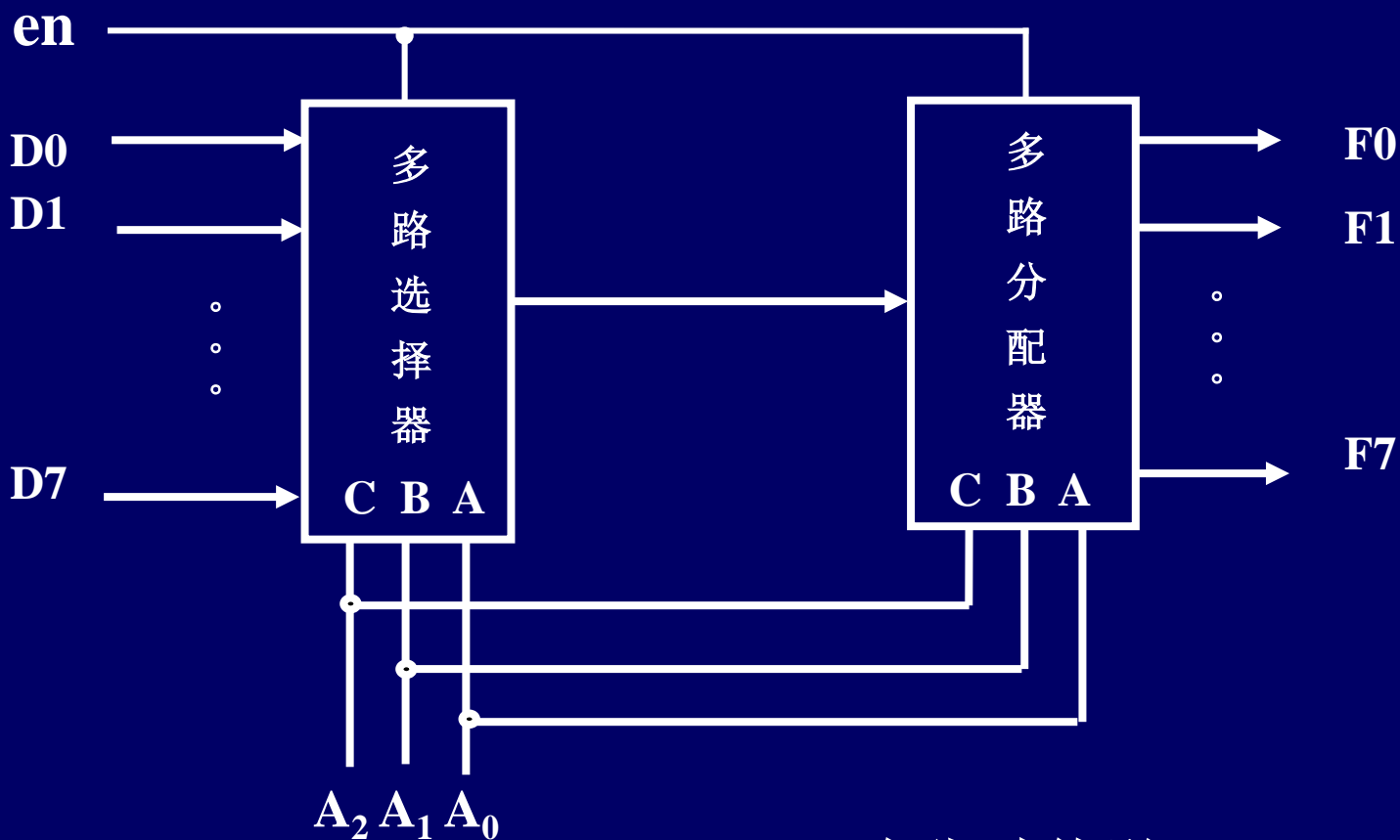
## 思考： Verilog HDL描述？

### 三. 多路选择器的应用

#### 1. 数据选择传送



## 2. 多路选择器与多路分配器联用，实现多通道数据分时传送。



8 路分时传送

## 8 路分时传送Verilog HDL模型

```
module time_tran (en, sel, D, F) ;  
    input          en ;  
    input  [2:0]   sel ;  
    input  [7:0]   D ;  
    output [7:0]   F ;  
    always @ (en or sel or D)  
        if (en==1)  
            case (sel)  
                3'b000 : F[0]=D[0];  
                3'b001 : F[1]=D[1];  
                .....  
                3'b111 : F[7]=D[7];  
            endcase  
        else F=8'b0;  
endmodule
```

### 3. 逻辑函数发生器

#### 多路选择器的特点

一般表达式: 
$$Y = \sum_{i=0}^{2^n-1} D_i m_i$$

- (1)具有标准“与或”表达式的形式
- (2)提供了选择变量的全部最小项
- (3)一般情况下,  $D_i$ 可当成一个变量处理
- (4)受使能端控制, 低有效。

逻辑函数都可以转换为与或表达式形式

## 用多路选择器实现逻辑函数的基本方法

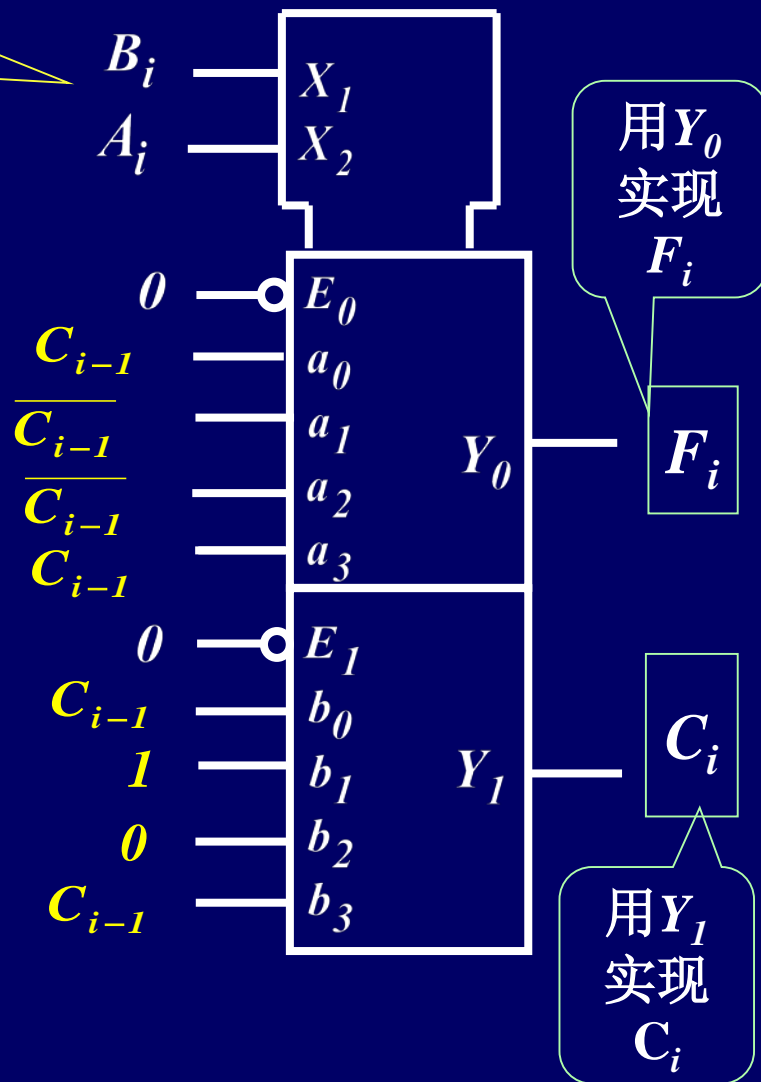
- (1) 函数的高位变量作为选择控制变量
- (2) 分析高位变量各种组合情况下，输出与其它变量的逻辑关系，确定各数据输入端的连接。  
(0, 1, 原变量、反变量、变量组合)
- (3) 画出电路图

# 例1：用双4选1多路选择器74LS153实现全减器功能。

确定 $A_i, B_i$ 作为选择控制变量接 $X_2, X_1$

全减器真值表：

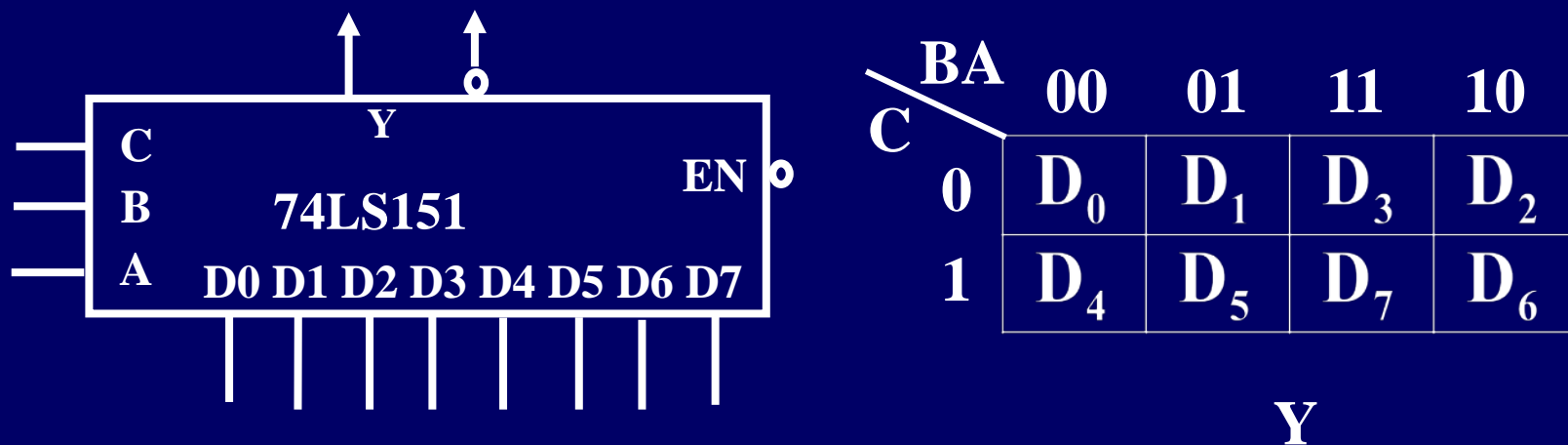
被减数	减数	低位借位	差	借位
$A_i$	$B_i$	$C_{i-1}$	$F_i$	$C_i$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



例2：用一个8-1多路选择器（74LS151）实现下列逻辑函数的功能。

$$F(X_3, X_2, X_1, X_0) = \sum m(1, 2, 4, 9, 10, 11, 12, 14, 15)$$

分析：74LS151逻辑符号，输出Y的卡诺图



逻辑函数的特点：四个变量



解：1. 作函数的卡诺图

2. 选函数变量 $X_3$ 、 $X_2$ 、 $X_1$ 作为选择控制变量，接到74LS151的C,B,A端。作函数的降维卡诺图。

3. 与74LS151的卡诺图比较，确定  $D_0 \sim D_7$ 。

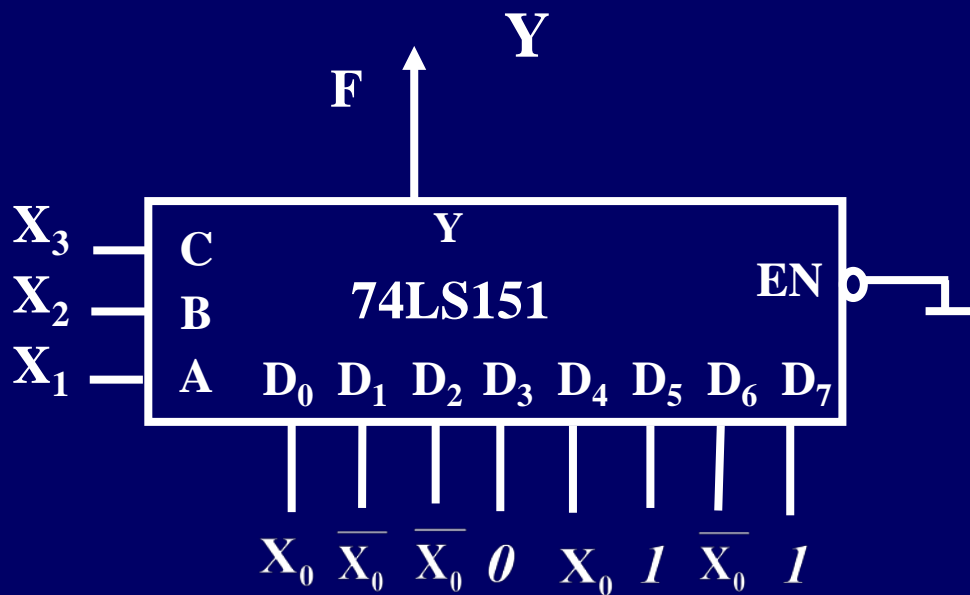
4. 画电路图

$X_1X_0$		00	01	11	10
$X_3X_2$	00	0	1	0	1
	01	1	0	0	0
	11	1	0	1	1
	10	0	1	1	1

$BA$		00	01	11	10
$C$	0	$D_0$	$D_1$	$D_3$	$D_2$
	1	$D_4$	$D_5$	$D_7$	$D_6$

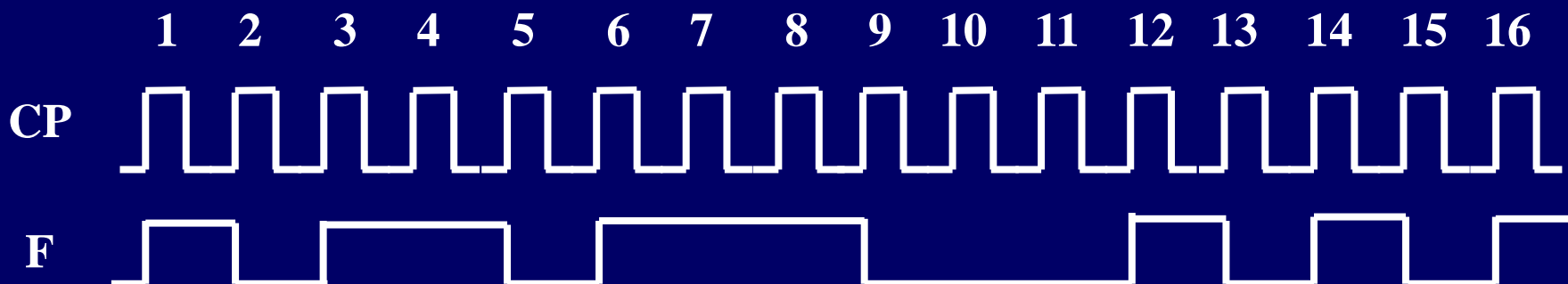
$X_2X_1$		00	01	11	10
$X_3$	0	$X_0$	$\overline{X_0}$	0	$\overline{X_0}$
	1	$X_0$	1	1	$\overline{X_0}$

$F$  (降维)



## 4. 不规则脉冲信号发生器

例：分析波形，用一个8选1实现 F 。



解：

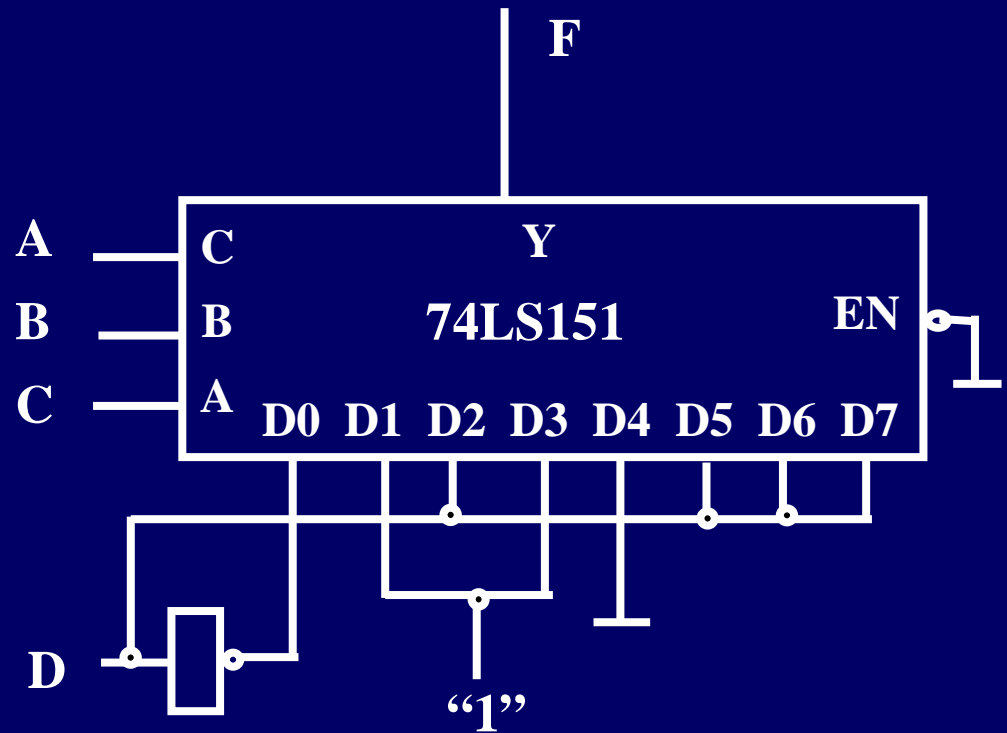
用变量A B C D为对应脉冲编码，即第一个脉冲对应0000，第二个脉冲对应0001，.....，第十六个脉冲对应1111

转换为F关于A,B,C,D的函数，列真值表。

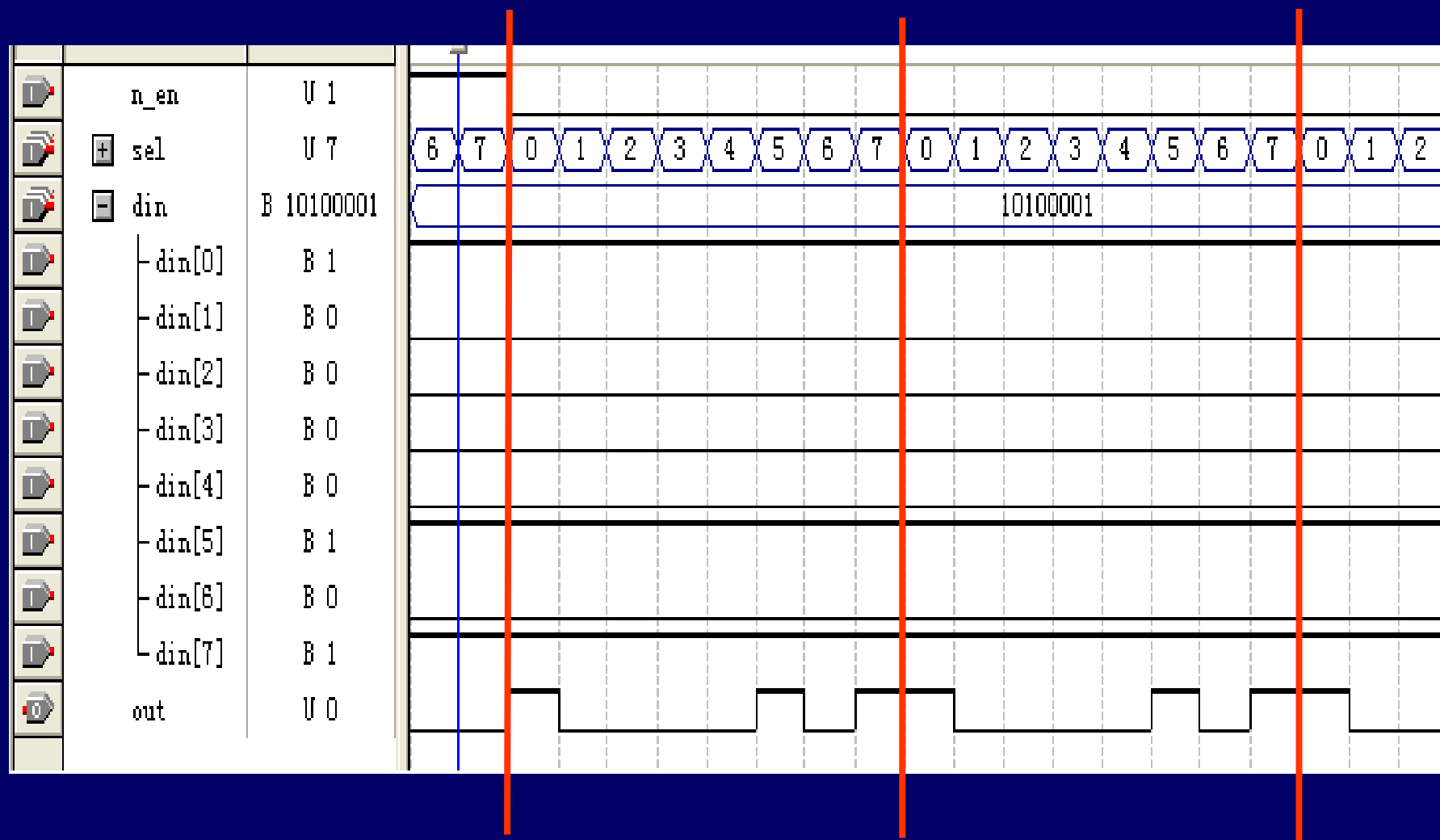
真值表

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>F</i>	
0	0	0	0	1	$\overline{D}$
0	0	0	1	0	
0	0	1	0	1	<b>1</b>
0	0	1	1	1	
0	1	0	0	0	<b>D</b>
0	1	0	1	1	
0	1	1	0	1	<b>1</b>
0	1	1	1	1	
1	0	0	0	0	<b>0</b>
1	0	0	1	0	
1	0	1	0	0	<b>D</b>
1	0	1	1	1	
1	1	0	0	0	<b>D</b>
1	1	0	1	1	
1	1	1	0	0	<b>D</b>
1	1	1	1	1	

将 *A B C* 作为多路选择器的选择控制变量，*D* 作为数据输入端的匹配变量，用 MUX 8-1 实现。



## 图示为输出10000101的不规则脉冲信号发生电路的仿真波形



请同学分析并建立Verilog HDL模型

**作业11:**

**4.18**

**4.19**

**4.25 (1)**