

人工电话交换机



数据选择器



无符号数补码：求反+1

第20讲 选择与基于事件的模拟

一、选择

二、基于事件的模拟

重点：简化的（紧凑）真值表

练习：41、47

选择

- 内容

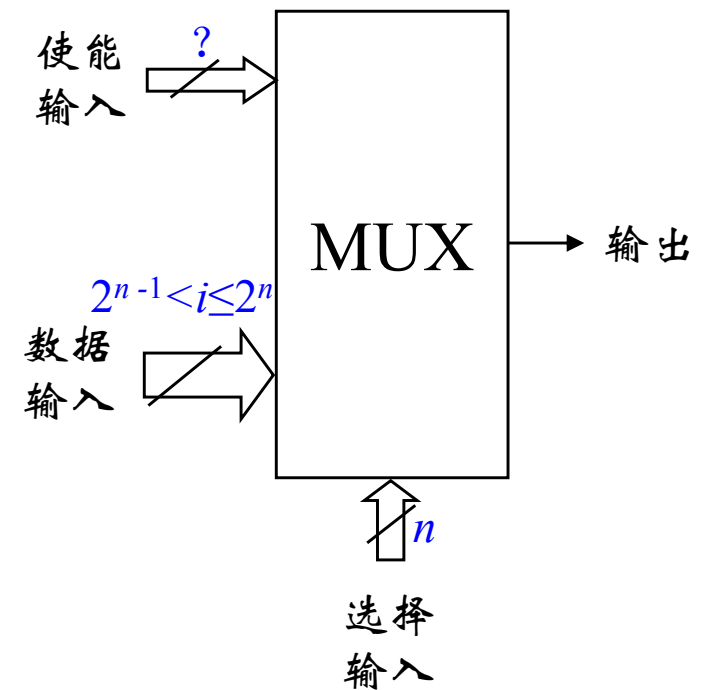
选择与选择器

- 目标

写出选择函数表达式，熟练运用选择器构建组合电路。

3.7 多路复用 -1位宽 (1/3)

- **选择**：从多个信号或数据中选择一个进行传输是数字系统中一项重要的操作
- 多路复用器或多路开关通常有
 - 1个或多个使能输入
 - 1组（共 n 个）**选择输入**
用来选择哪一个对象
 - 1组（共 i 个， $2^{n-1} < i \leq 2^n$ ）**数据输入** 作为被选择的对象
 - 1个输出，输出被选中的那个对象



3.7 多路复用 -1位宽 (2/3)

● 2-1 多路开关

- $i=2=2^n$, $n=1$;
- $S=0$ 时选 I_0 ; 否则选 I_1 。
- 输出表达式

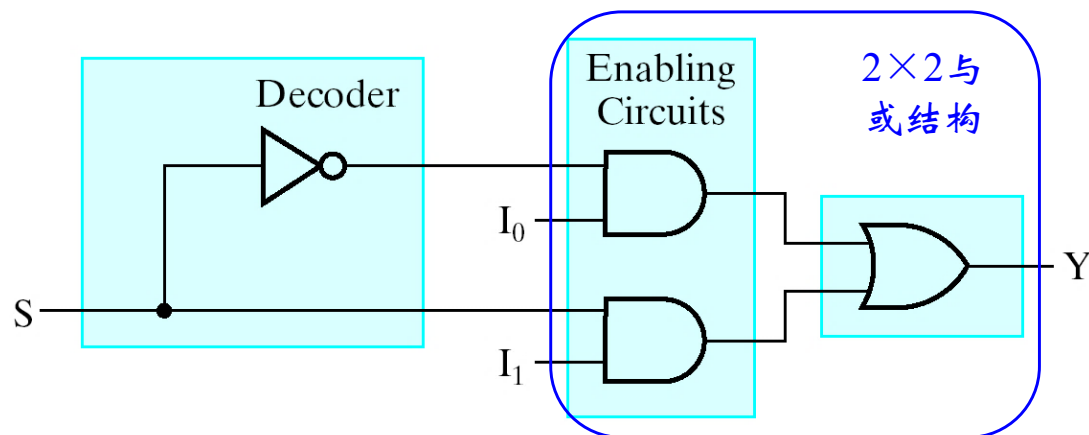
$$Y = \bar{S}I_0 + SI_1$$

| S | I ₀ | I ₁ | Y |
|---|----------------|----------------|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

紧凑式
真值表

| S | Y |
|---|----------------|
| 0 | I ₀ |
| 1 | I ₁ |

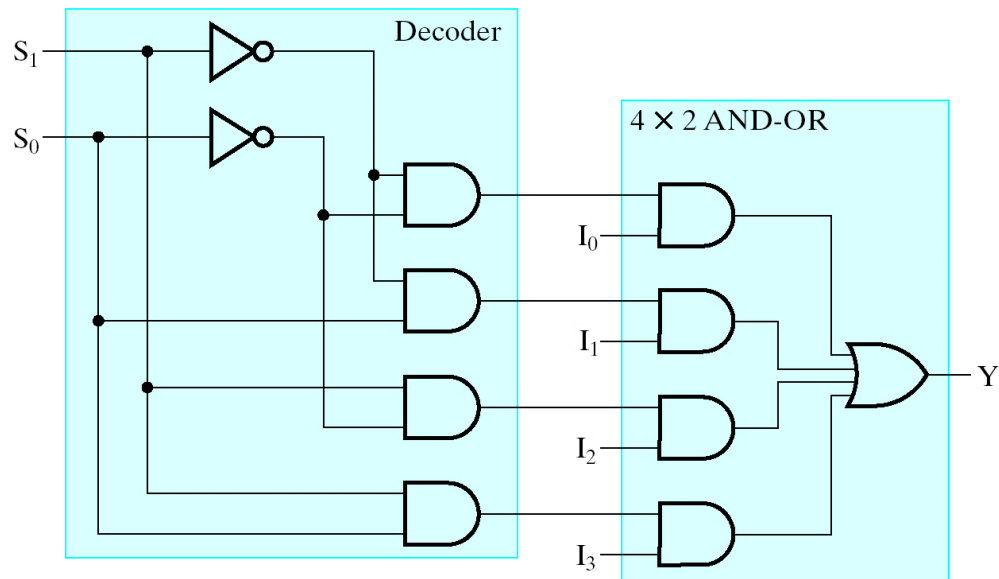
● 紧凑式真值表



3.7 多路复用 -1位宽 (3/3)

● 4-1 多路开关

| S_1 | S_0 | Y |
|-------|-------|-------|
| 0 | 0 | I_0 |
| 0 | 1 | I_1 |
| 1 | 0 | I_2 |
| 1 | 1 | I_3 |



● 1个 2^n-1 多路开关可分解为：

● 1个 $n-2^n$ 译码器和

● 1个 $2^n \times 2$ 与或结构。

3.7 多路复用 - 多位宽 (1/1)

● 多重 (多位宽) 多路开关

● 每次选择1个向量

● 用1个译码器

● 并行使用多个与或结构

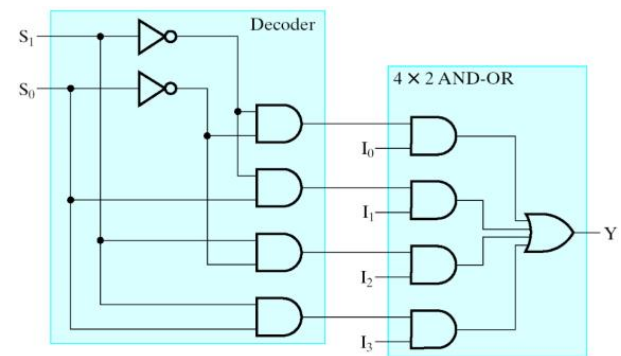
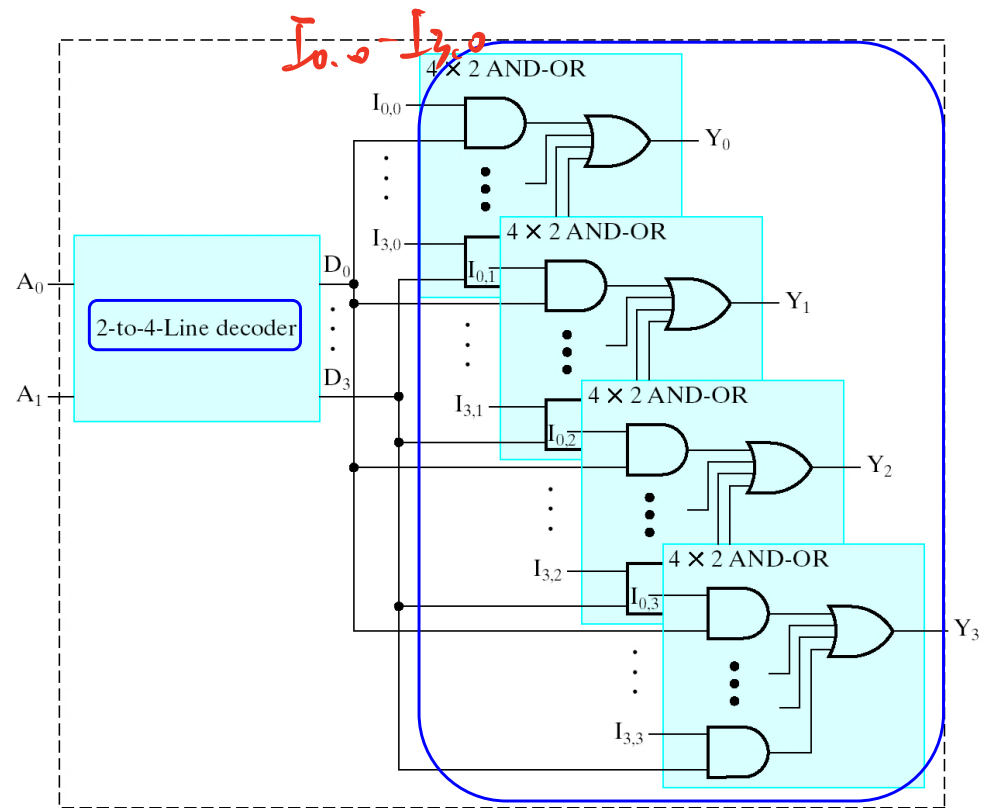
例如 4重4-1多路开关

● 输入向量信号:

$(I_{0,0}, I_{0,1}, I_{0,2}, I_{0,3}), \dots, (I_{3,0}, I_{3,1}, I_{3,2}, I_{3,3})$

● 选择信号: A_1, A_0

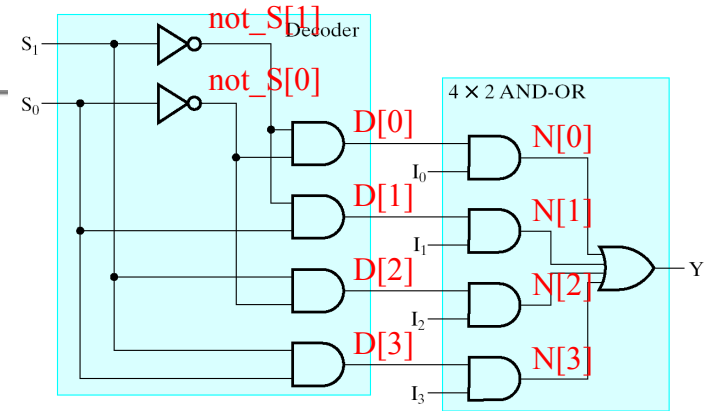
● 输出向量信号: (Y_0, Y_1, Y_2, Y_3)



3.7 多路复用 - Verilog模型 (1/3)

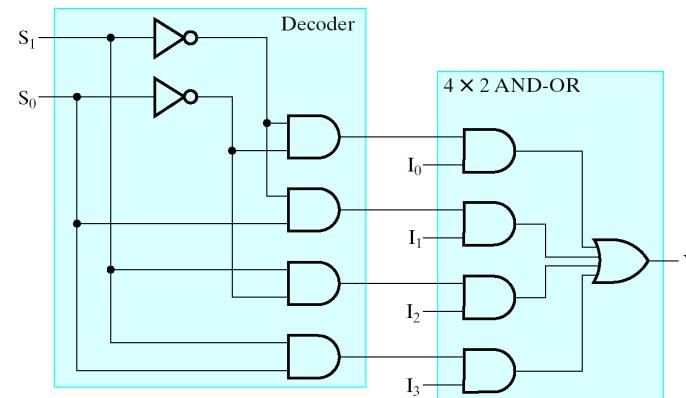
● 4-1 多路开关Verilog结构模型

```
1 //4-to-1 Multiplexer:structural verilog Description
2 //(See Figure 3-25 for logic diagram)
3 module multiplexer_4_to_1_st_v(S,I,Y);
4 input[1:0] S;
5 input[3:0] I;
6 output Y;
7
8 wire [1:0] not_S;
9 wire [0:3] D,N;
10
11 //not
12 not gn0(not_S[0],S[0]);
13 not gn1(not_S[1],S[1]);
14 //and
15 and g0(D[0],not_S[1],not_S[0]);
16 and g1(D[1],not_S[1],S[0]);
17 and g2(D[2],S[1],not_S[0]);
18 and g3(D[3],S[1],S[0]);
19 and g4(N[0],D[0],I[0]);
20 and g5(N[1],D[1],I[1]);
21 and g6(N[2],D[2],I[2]);
22 and g7(N[3],D[3],I[3]);
23 //or
24 or g8(Y,N[0],N[1],N[2],N[3]);
25 endmodule
```



3.7 多路复用 - Verilog模型 (2/3)

● 采用布尔表达式的4-1多路开关
Verilog数据流描述



```
// 4-to-1-Line Multiplexer: Dataflow Verilog Description
```

```
// (See Figure 3-25 for logic diagram)
```

```
module multiplexer_4_to_1_df_v(S, I, Y);
```

```
    input [1:0] S;
```

```
    input [3:0] I;
```

```
    output Y;
```

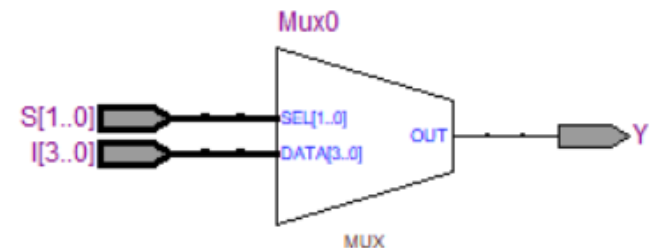
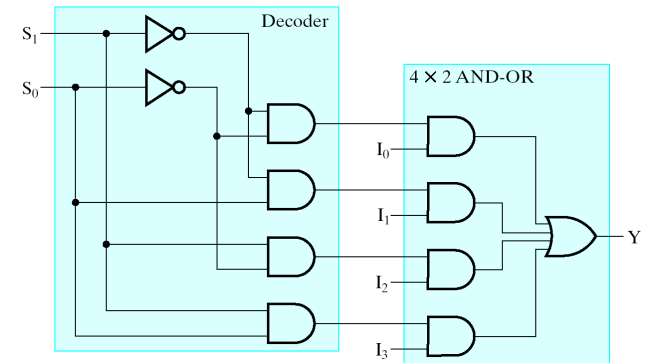
```
    assign Y = (~ S[1] & ~ S[0] & I[0]) | (~ S[1] & S[0] & I[1])  
              | (S[1] & ~ S[0] & I[2]) | (S[1] & S[0] & I[3]);
```

```
endmodule
```

3.7 多路复用 - Verilog模型 (3/3)

● 采用二进制判定的4-1多路开关
Verilog条件行为描述

```
1  module multi_4_to_1(S,I,Y);  
2      input[1:0] S;  
3      input[3:0] I;  
4      output Y;  
5      reg Y;  
6      always @(S or I)  
7      begin  
8          case (S)  
9              2'b00: Y=I[0];  
10             2'b01: Y=I[1];  
11             2'b10: Y=I[2];  
12             2'b11: Y=I[3];  
13             default: Y=1'bx;  
14         endcase  
15     end  
16 endmodule
```



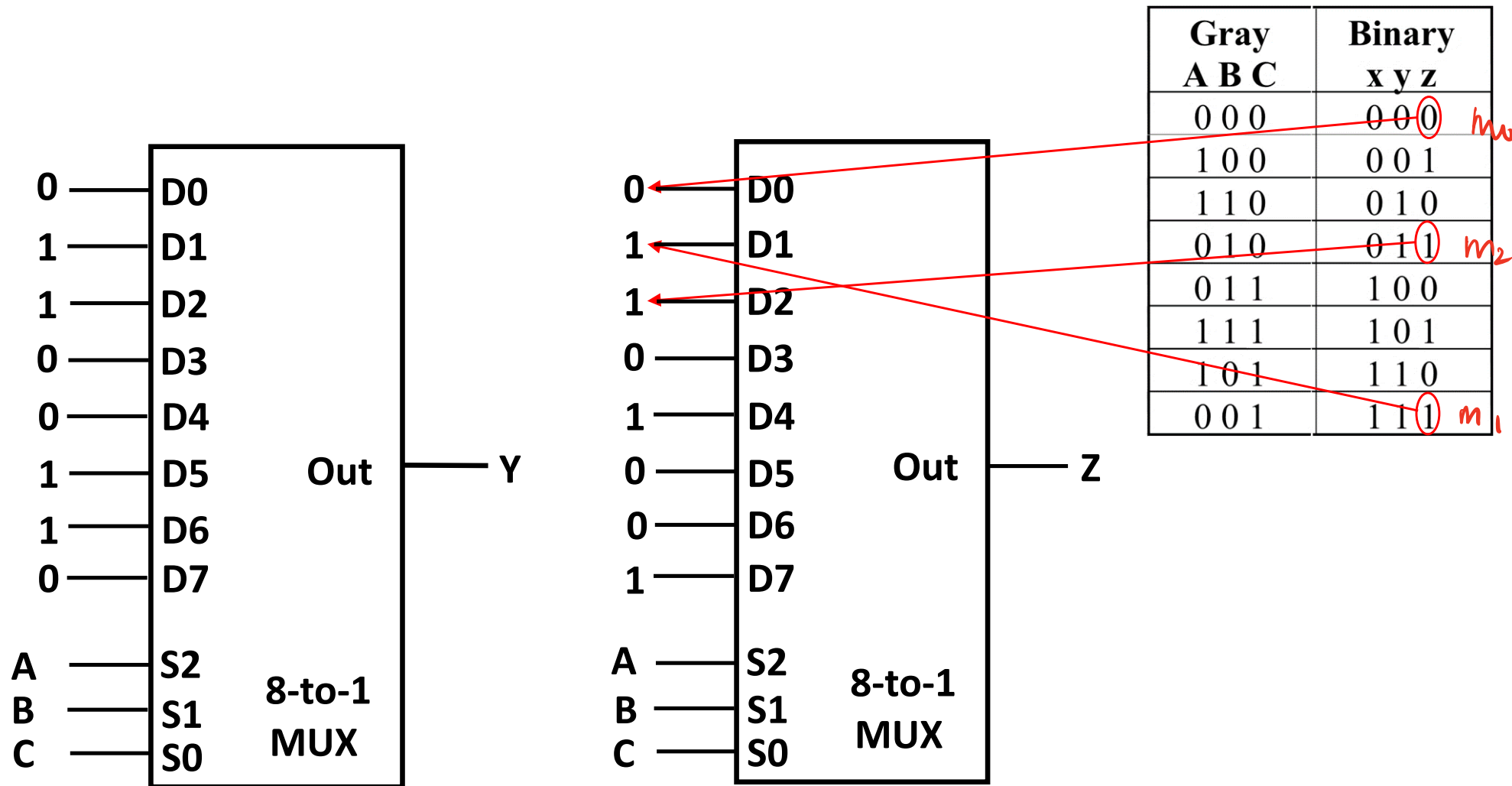
3.7 多路复用 -应用方法1 (1/2)

例20.1 用8-1多路开关设计一个3位格雷码-二进制码转换器，真值表如右所示。

- $x = C$
- y 和 z 分别用一个8-1多路开关实现
 - 将 y 和 z 作为输出
 - A 、 B 、 C 作为选择输入
 - 将 y 和 z 在表中的取值0或1作为数据输入

| Gray A B C | Binary x y z |
|---------------|-----------------|
| 0 0 0 | 0 0 0 |
| 1 0 0 | 0 0 1 |
| 1 1 0 | 0 1 0 |
| 0 1 0 | 0 1 1 |
| 0 1 1 | 1 0 0 |
| 1 1 1 | 1 0 1 |
| 1 0 1 | 1 1 0 |
| 0 0 1 | 1 1 1 |

3.7 多路复用 - 应用方法1 (2/2)



● 数据输入为固定值的多路开关实际上就是ROM。

3.7 多路复用 -应用方法2 (1/3)

例20.2 用4-1多路开关设计一个3位格雷码-二进制编码的转换器，真值如右所示。

- $x = C$;
- y 和 z 分别由一个4-1多路开关来和一个反相器实现。
 - 将 y 、 z 作为输出;
 - 将 A 、 B 作为选择输入;
 - 将0、1、 C 和 \bar{C} 作为数据输入。

| Gray A B C | Binary x y z |
|---------------|-----------------|
| 0 0 0 | 0 0 0 |
| 1 0 0 | 0 0 1 |
| 1 1 0 | 0 1 0 |
| 0 1 0 | 0 1 1 |
| 0 1 1 | 1 0 0 |
| 1 1 1 | 1 0 1 |
| 1 0 1 | 1 1 0 |
| 0 0 1 | 1 1 1 |

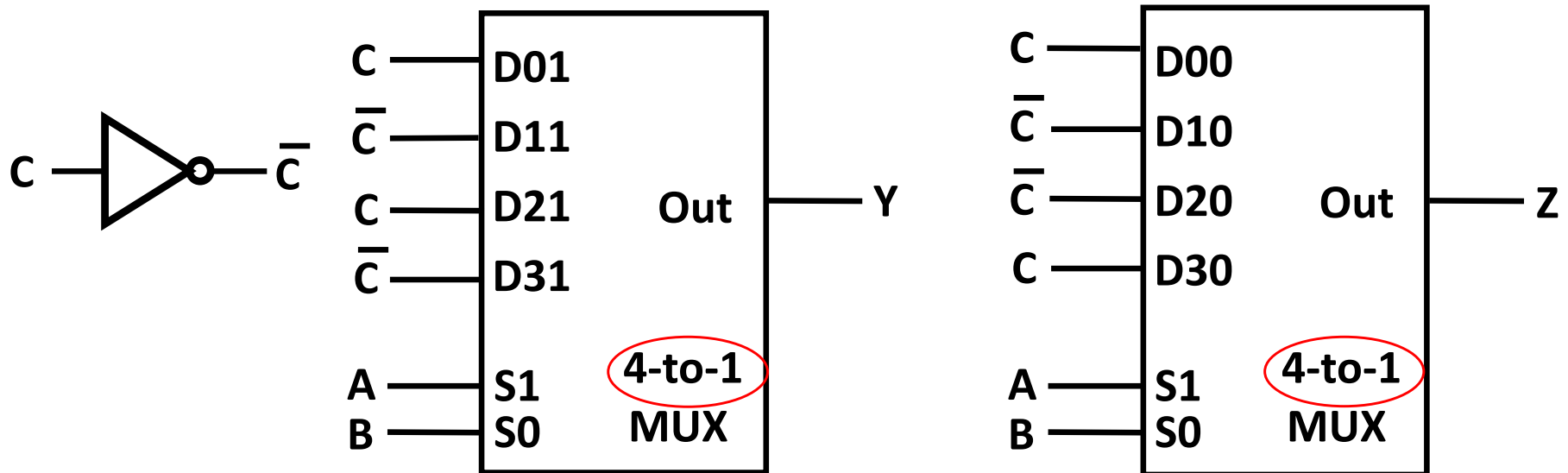
3.7 多路复用 -应用方法2 (2/3)

- 重新安排表格，使得数据输入按计数顺序排列，然后用基本函数表示各个输出。

| Gray A B C | Binary x y z | Rudimentary Functions of C for y | Rudimentary Functions of C for z |
|---------------|-----------------|--|--|
| 0 0 0 | 0 0 0 | $y = C$ | $z = C$ |
| 0 0 1 | 1 1 1 | | |
| 0 1 0 | 0 1 1 | $y = \bar{C}$ | $z = \bar{C}$ |
| 0 1 1 | 1 0 0 | | |
| 1 0 0 | 0 0 1 | $y = C$ | $z = \bar{C}$ |
| 1 0 1 | 1 1 0 | | |
| 1 1 0 | 0 1 0 | $y = \bar{C}$ | $z = C$ |
| 1 1 1 | 1 0 1 | | |

3.7 多路复用 - 应用方法2 (3/3)

- 将数据输入连接到基本函数，给多路开关的输出赋值。



- 该方法与例3.6相比成本降低约1/2，但其结构已经不是ROM了。

基于事件的模拟

- 内容

基于事件的组合电路模拟

- 目标

用人工方法**进行**基于事件的模拟。

基于事件的组合电路模拟 (1/2)

- 如何确认设计没有错误是一个NP难问题

- 功能模拟

- 要多少输入组合?

- 形式化验证 (数学证明)

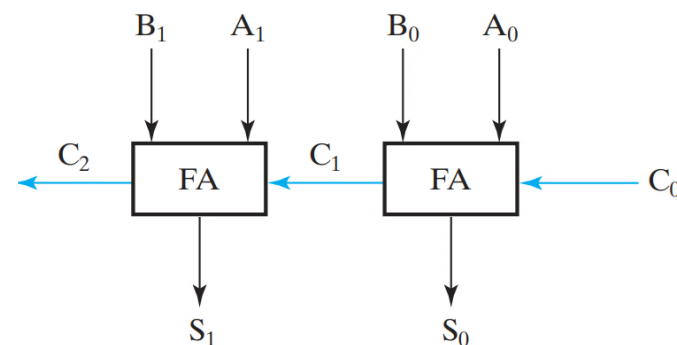
- 太复杂

- C-testable Circuits (常数可测的电路)

- 叠代电路的可测性

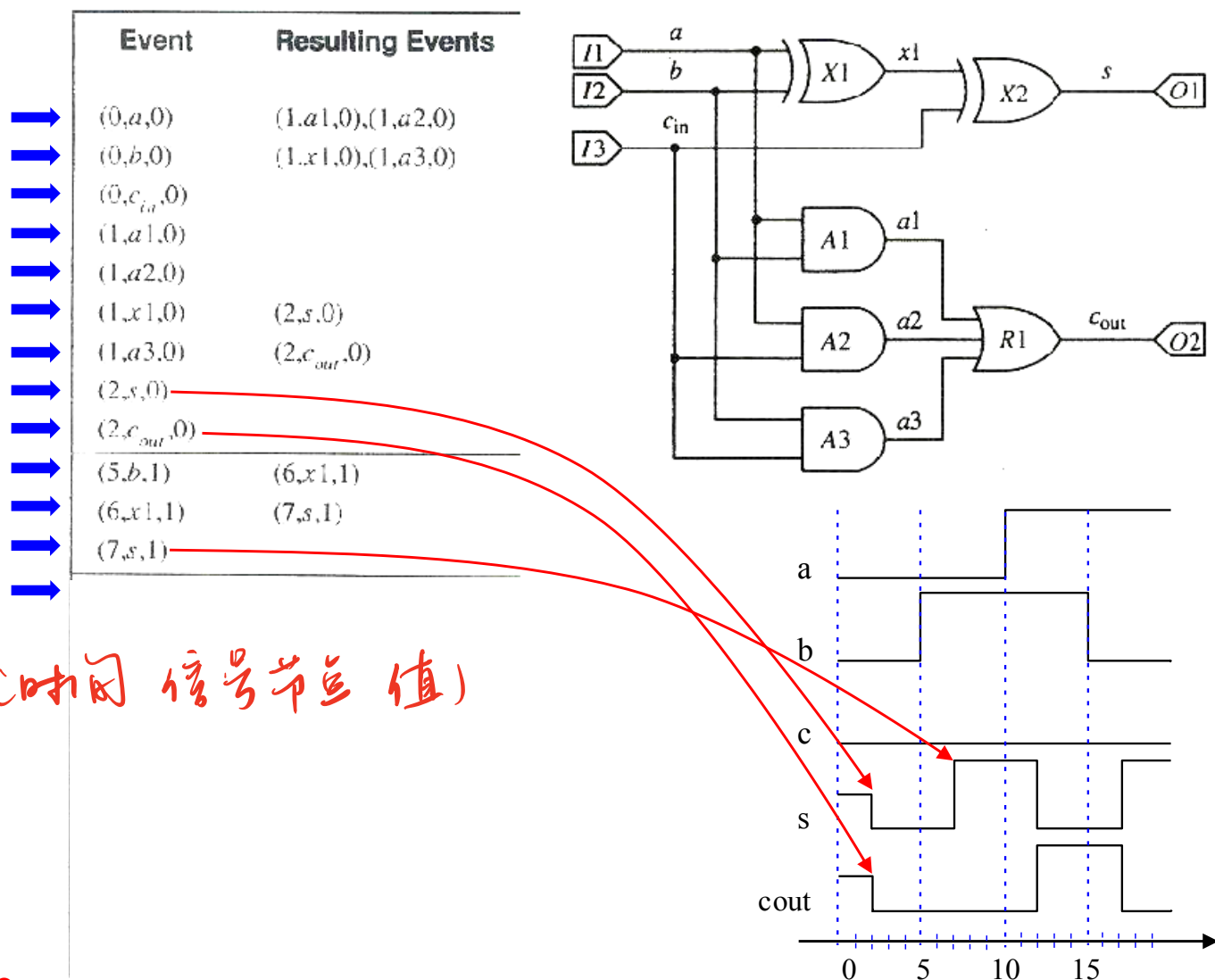
- $n=1$ 时, 8个输入组合

- $n=2$ 时, 多少个输入组合?



基于事件的组合电路模拟 (2/2)

● (基于事件的) 组合电路 模拟



(时间信号节点值)

小结

- 多路复用
 - 1位宽多路复用器
 - 多位宽多路复用器
 - 用多路开关实现组合电路
- 紧凑的真值表
- 基于事件的模拟
 - C-可测电路
 - 叠代电路的可测性
- 测验