

第八章 一般时序电路的设计

8.1 一般时序电路设计的步骤

8.2 建立原始状态图和状态表

8.3 状态化简

8.4 状态分配

8.5 Verilog HDL建模和仿真

8.1 一般时序电路设计的主要步骤

- 1) 根据文字叙述的逻辑功能或时间波形图建立原始状态图和状态表;
- 2) 状态化简, 消去多余状态, 得到最简状态表, 又称最小化状态表;
- 3) 状态分配, 对用文字或符号表示的状态进行编码;
- 4) 本课程要求用**Verilog HDL**建模;
- 5) 在**EDA**平台进行仿真验证 (不做考核要求)

8.2 建立原始状态图和状态表

充分考虑输入输出的各种状态，“宁多勿漏”。有两种常用的方法：

1) 直接状态指定法。

根据文字描述的设计要求，假设一个初态，从这个初态出发，每加入一个输入，就确定其次态，该次态可能是现态本身、另一个已有状态、需新增状态。重复上述过程，直到每一个现态向次态的转换都已确定且不再产生新的状态。

2) 信号序列法。

根据逻辑要求，先列出输入和输出的信号序列，并用箭头标出各信号间的后续关系，再画出状态图。

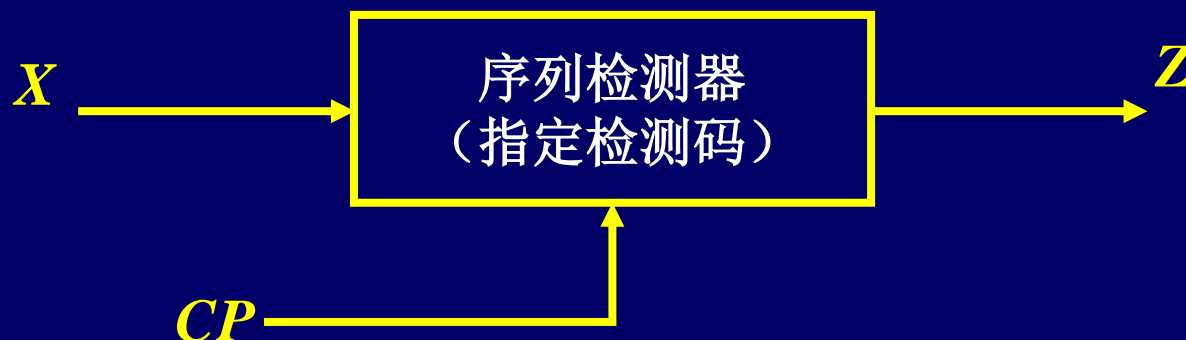
常见时序电路设计的举例：

- 序列检测器
- B码识别器和双向可逆分配器
- 串行加法器
- 非法码检测、串行代码转换

8.2.1 序列检测器

功能描述：

序列检测器有固定的检测码，接受一组串行随机信号（ X ），在时钟 CP 的控制下，每一个 CP 接受一位 X 的数据，当输入数据与检测码相同时，检测电路输出有效信号。



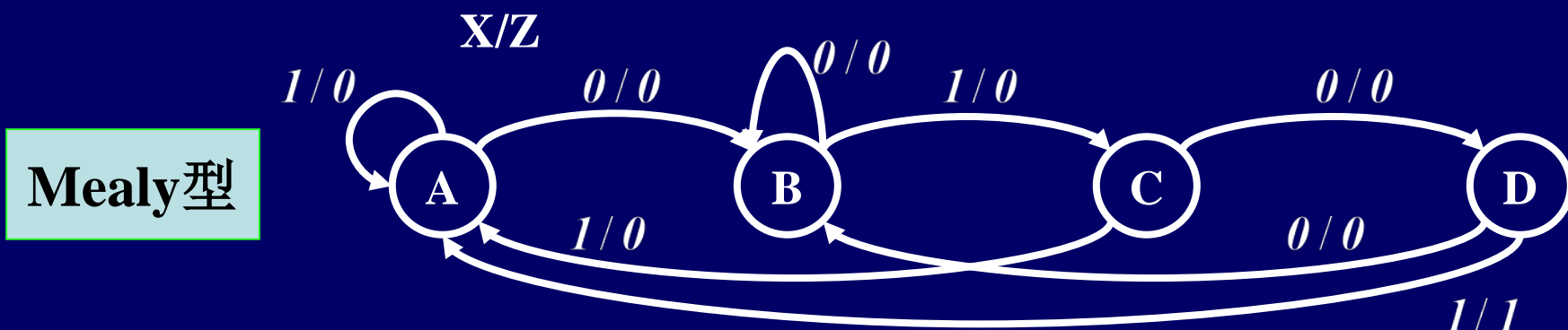
电路较简单，可以采用直接状态指定法。

要 点

- 1) 检测码的位数=状态数;
- 2) 因为只有一位外部输入, 所以每个状态有两个出口;
- 3) 相同的输入序列, 可以产生不同的输出序列, 分为可重或不可重;
- 4) 先画出主序列, 再画出辅助序列。

例1：画出0101序列检测器的原始状态图和状态表。

解：四位检测码应有四个状态数，分别为A、B、C、D。先画主序列，再画辅助序列：



输入X 现态	0	1
A	B/0	A/0
B	B/0	C/0
C	D/0	A/0
D	B/0	A/1

次态/输出Z

验证：

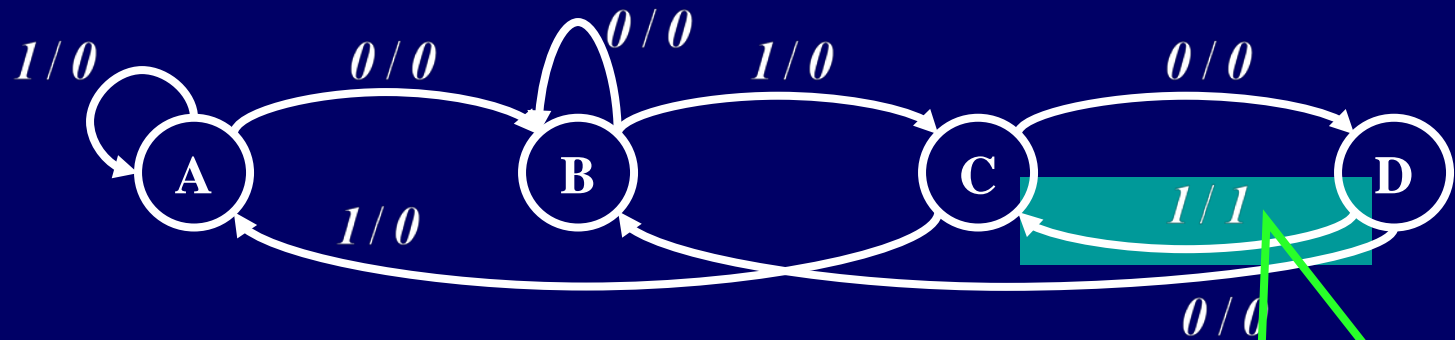
当输入X=010101010101.....时，

输出为Z=000100010001

不可重!!!

分析：若输入 $X=01010101101\dots$ 时，
输出为 $Z=00010101000$

可重!!!



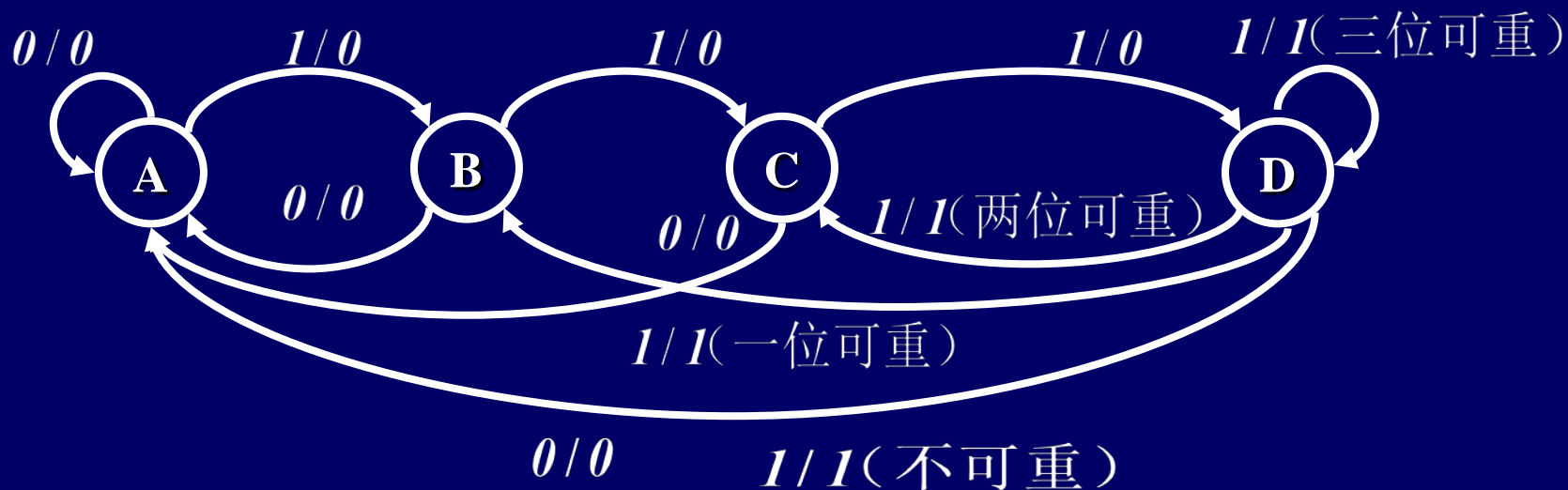
输入X \ 现态	0	1
A	B/0	A/0
B	B/0	C/0
C	D/0	A/0
D	B/0	C/1

次态/输出Z

不可重：检测到有效序后，
返回初态；
可重：检测到有效序列后，
转到有关状态。

例2：画出1111序列检测器的原始状态图。

解：有不可重、一位可重、两位可重和三位可重四种情况，用一张图表示。



8.2.2 串行加法器

例：试画串行加法器原始状态图。

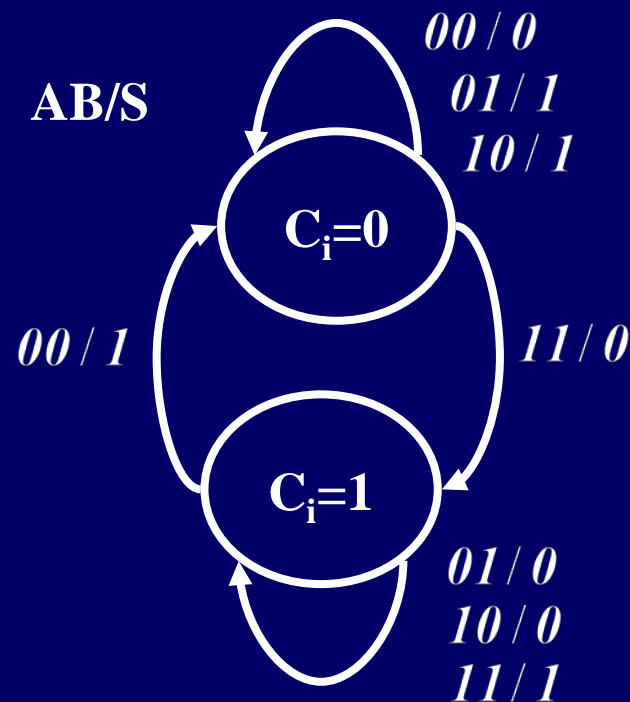
解：设两个二进制数为：
 $A = a_n a_{n-1} \cdots a_1 a_0$
 $B = b_n b_{n-1} \cdots b_1 b_0$

其运算的和为： $S = s_n s_{n-1} \cdots s_1 s_0$

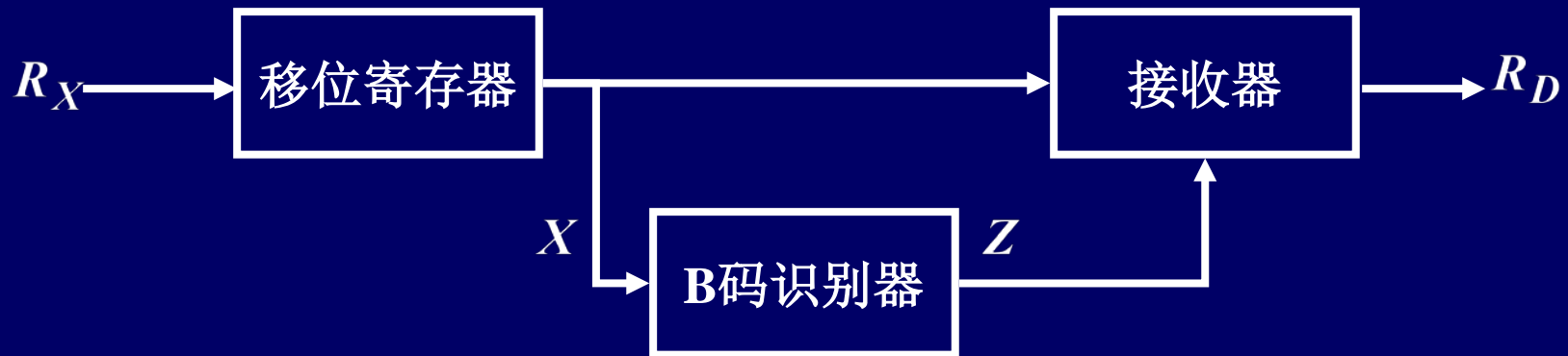
由低位向高位相加。低位相加的进位位需要保留参加高一位的运算，需要有一个触发器记录有无进位的两种状态。

$\begin{array}{c} AB \\ \hline C_i \end{array}$	00	01	11	10
0	0/0	0/1	1/0	0/1
1	0/1	1/0	1/1	1/0

C_i 次态/和S



8.2.3 B码识别器和双向可逆分配器

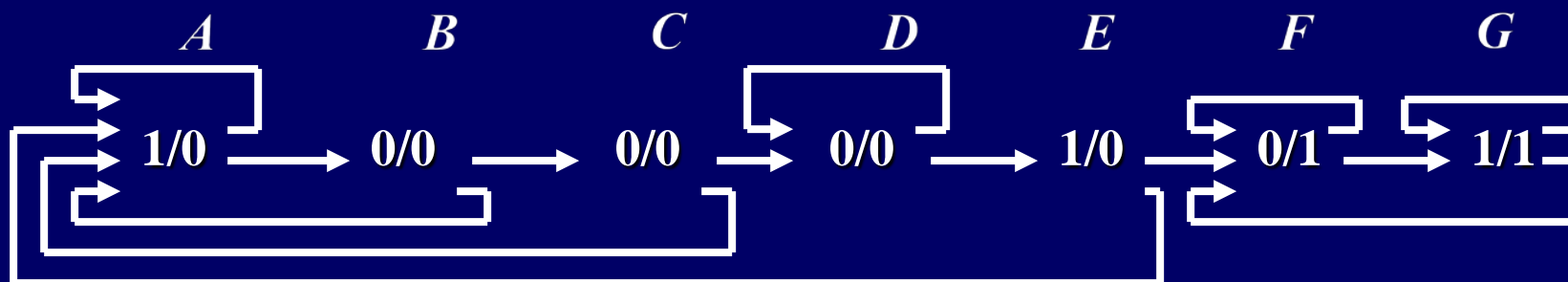


在数字通讯中，常常用到B码识别器。当输入的信号和检测码一致时，B码识别器输出为有效，数字电路打开接收器接受数据。

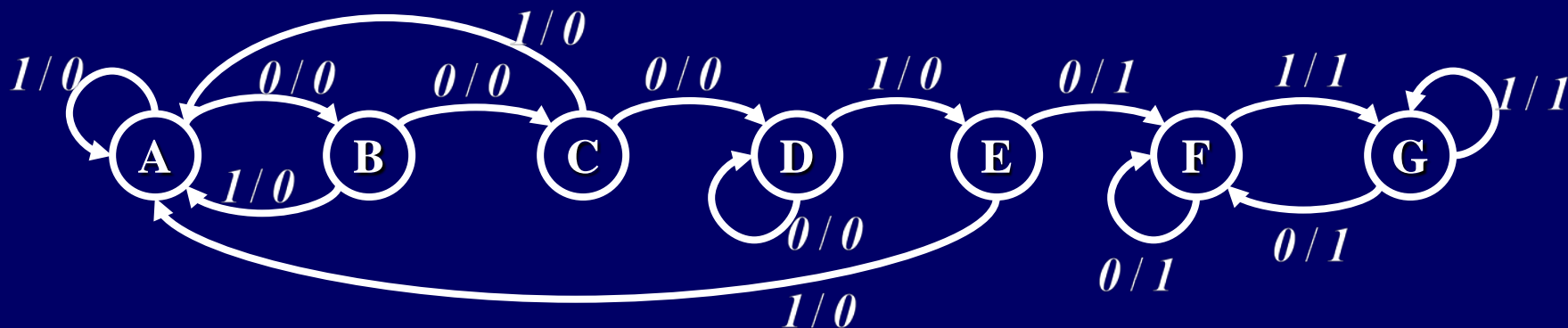
（在计算机网络（以太网）的同步通讯时，用01111110（7E）来表示一帧信息的开始和结束，实现帧同步的操作。）

例1：设计检测码为01000的B码识别器（要求低位先进入识别器）的原始状态图。

解：采用信号序列法确定状态数：



共需要七个状态构造原始状态图：



请同学补充原始状态表

双向可逆分配器常见于指示和指针类设备

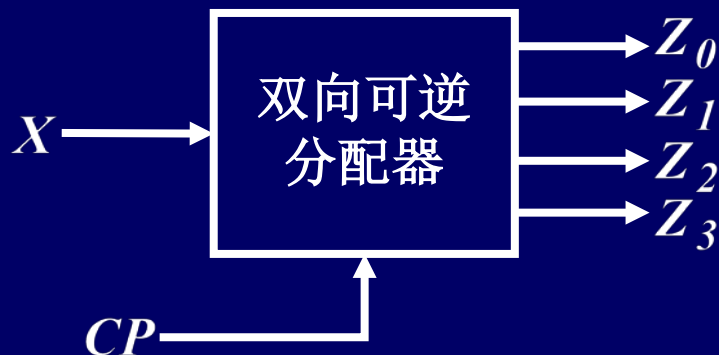
输入为 X ，是一个串行的数字系列；

输出有 n 个， Z_{n-1} ， Z_{n-2} ， \dots ， Z_1 ， Z_0 。

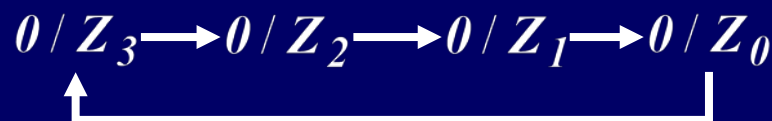
当 $X=0$ 时，在 CP 作用下，将 X 依次分配给 Z_{n-1} ， Z_{n-2} ， \dots ， Z_1 ， Z_0 ；

当 $X=1$ 时，在 CP 的作用下，将 X 依次分配给 Z_0 ， Z_1 ， \dots ， Z_{n-2} ， Z_{n-1} 。

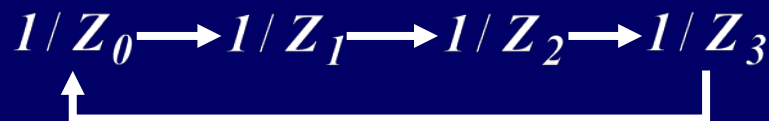
例2：某双向可逆分配器如下，做其原始状态图。



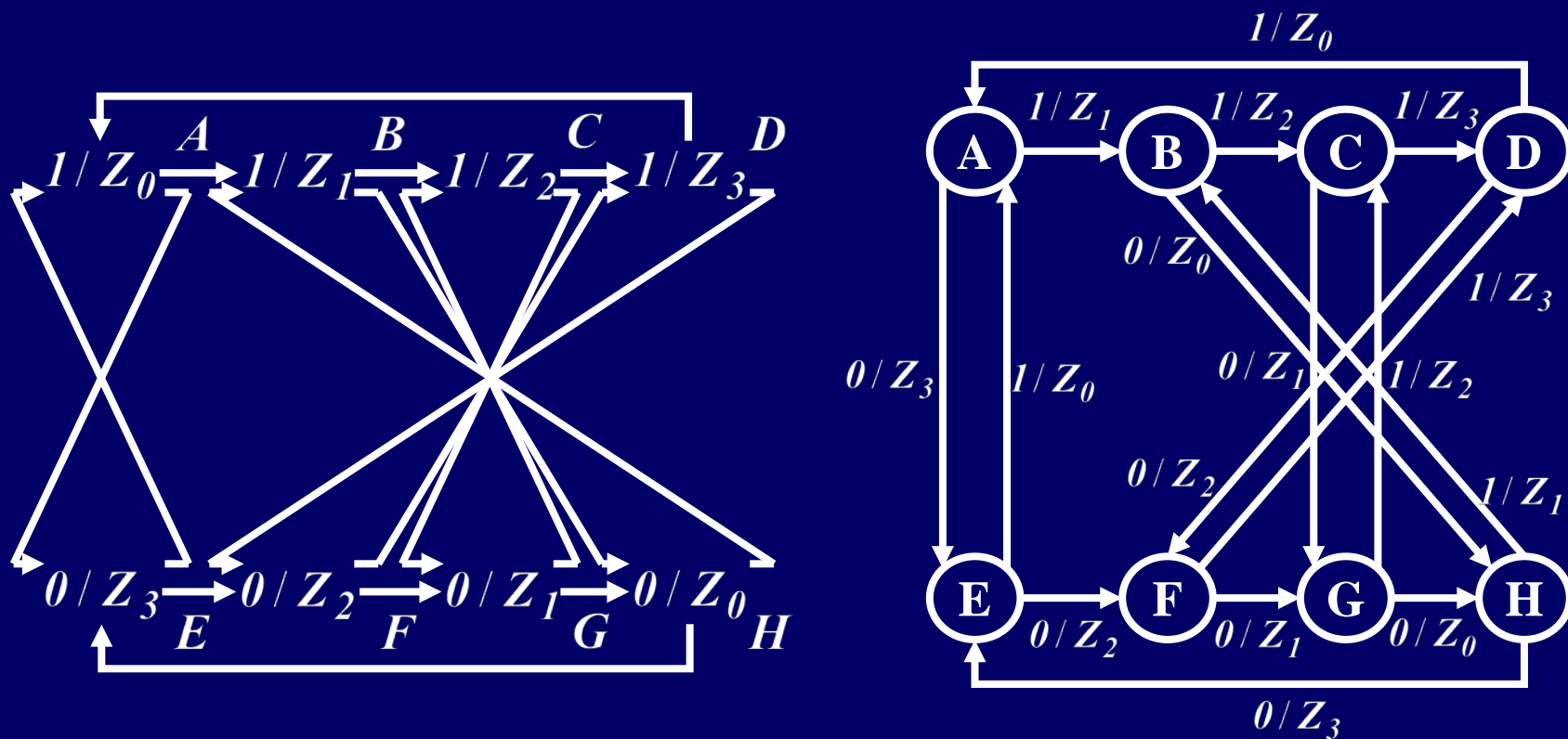
解：当 $X=0$ 时，信号序列 X/Z 为：



当 $X=1$ 时，信号序列 X/Z 为：



因 x 是随机变化的，所以两种工作方式相互转换。将每个信号的两种可能转向都要考虑进去，可以得到如下信号序列：



8.2.4 非法码检测、串行代码转换

这类电路的原始状态图通常画为树状，由信号序列决定状态。

形成状态树时注意：

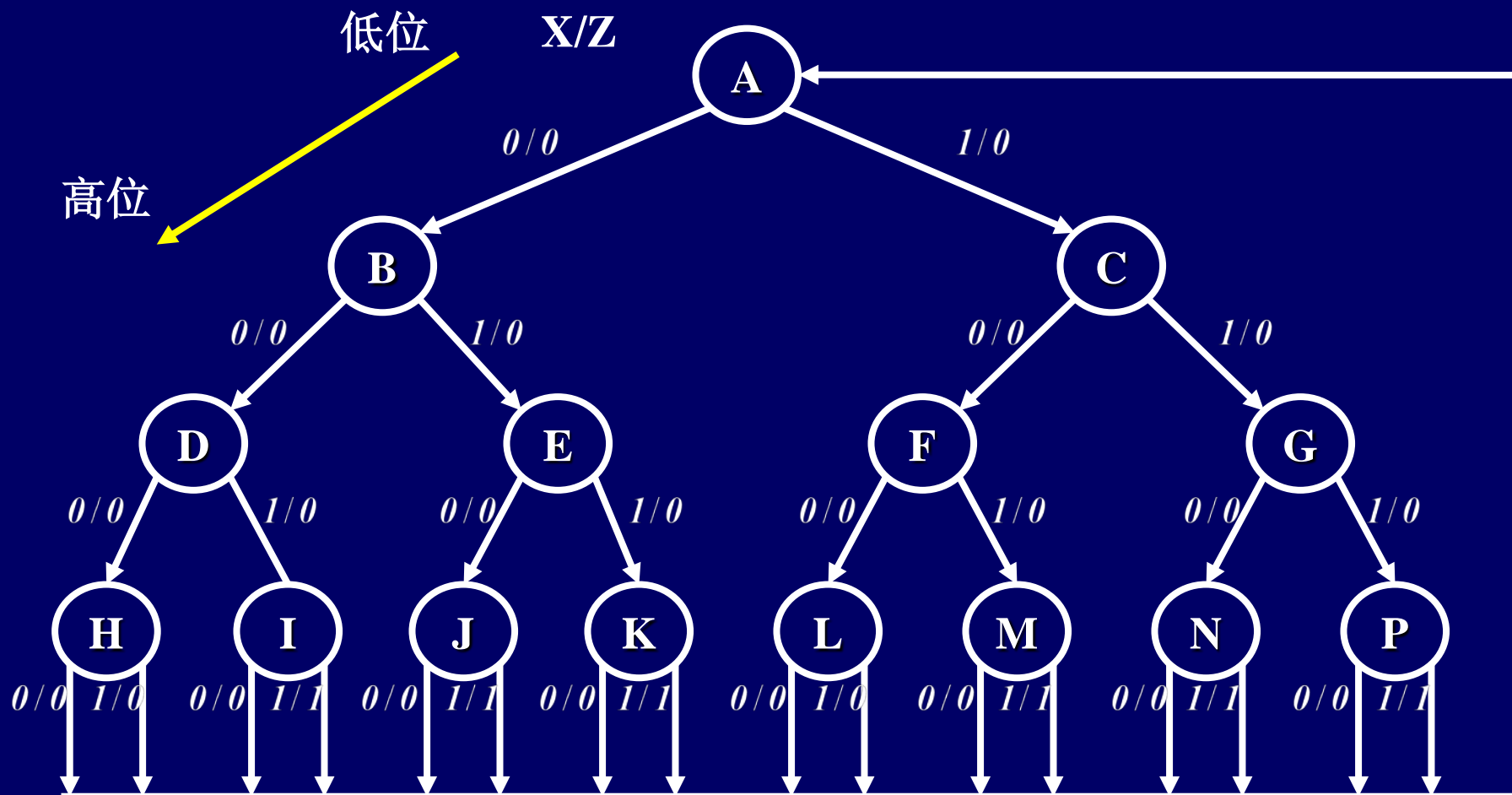
- 1) 按被检测码的位数将串行输入分段，每检测完一段都要返回初始态。
- 2) 从串行输入的高位还是低位开始检测。

例1：试画出检验串行输入8421码的非法码的检测电路的原始状态图。

解：设检测出非法码时输出Z为1，正常8421码时输出Z为0。电路由低位开始检测。

设检测电路的初态为A，随着X的第一位输入，电路由A状态转换为B（X=0）或者C（X=1）状态；随着X的第二位输入，电路由B、C状态进入D、E、F或G状态；当X的第三位输入时，状态又可以变化为H、I、J、K、L、M、N或P状态。在这阶段，输出Z一直处于0状态。当X输入第四位时，在不同的现态下，若检测出不符合8421码的非法码，输出Z为1，若符合8421码，输出Z为0，电路返回初始态。

8421码非法码检测电路原始状态图:



串行8421非码检测 原始状态表

X 现态	0	1
A	B/0	C/0
B	D/0	E/0
C	F/0	G/0
D	H/0	I/0
E	J/0	K/0
F	L/0	M/0
G	N/0	P/0
H	A/0	A/0
I	A/0	A/1
J	A/0	A/1
K	A/0	A/1
L	A/0	A/0
M	A/0	A/1
N	A/0	A/1
P	A/0	A/1

次态/输出Z

例2：试画出串行余3码-8421码的转换电路的原始状态图。

解：余3码为串行输入，8421码为串行输出，对照表如下：

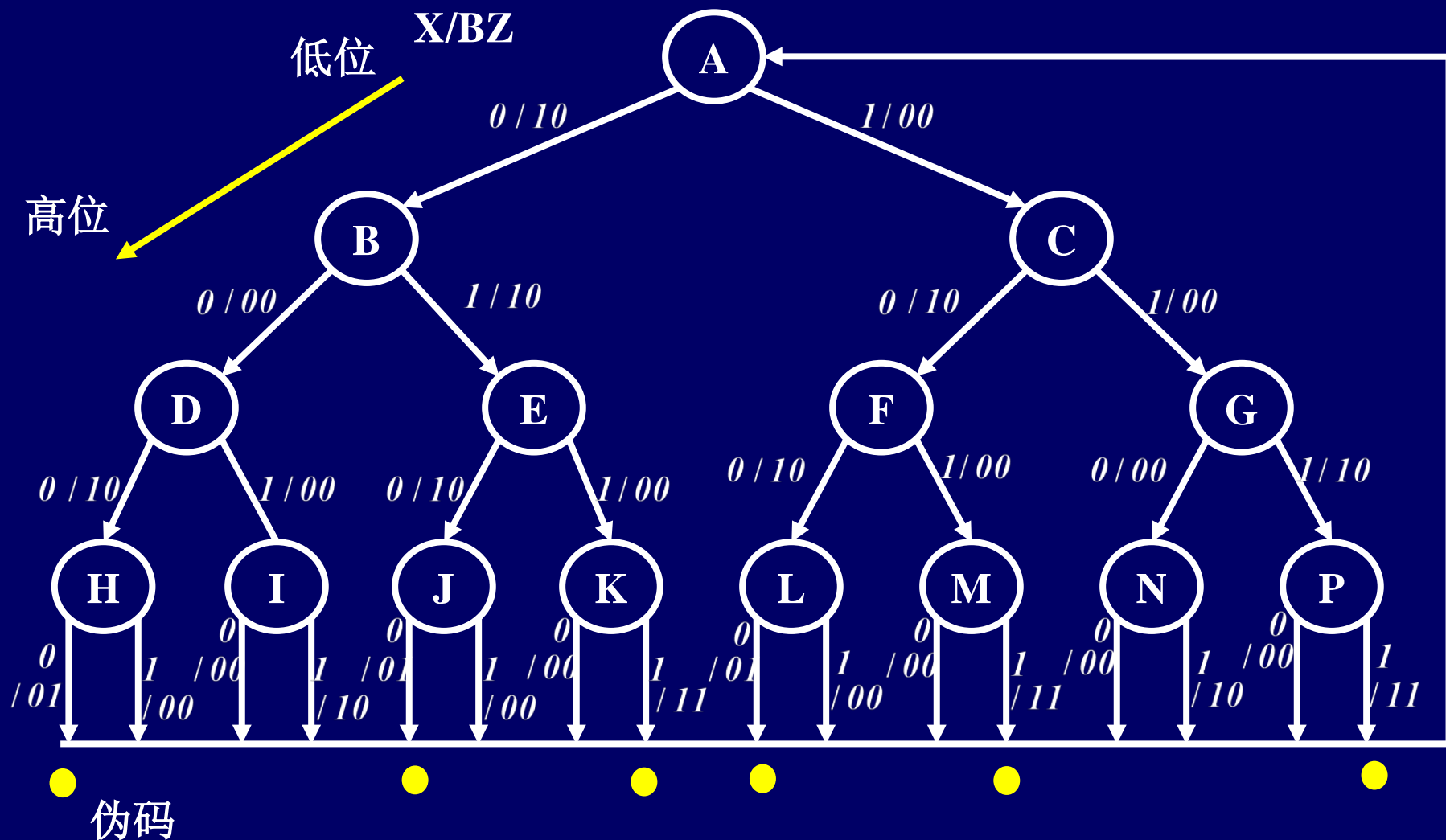
余3码	8421码
0 0 1 1	0 0 0 0
0 1 0 0	0 0 0 1
0 1 0 1	0 0 1 0
0 1 1 0	0 0 1 1
0 1 1 1	0 1 0 0
1 0 0 0	0 1 0 1
1 0 0 1	0 1 1 0
1 0 1 0	0 1 1 1
1 0 1 1	1 0 0 0
1 1 0 0	1 0 0 1

转换电路从低位开始转换，输入X为余3码，输出B为8421码，还需要有一位输出Z，用于检测非余3码，Z=1时，输入出现非余3码。输入输出的信号序列为：X/BZ。

电路初始态为A。

分析转换规律

串行余3码-8421码代码转换电路原始状态图：



请同学自己完成原始状态表

作业19:

8.1

8.2

8.4

8.3 状态化简

通过原始状态图就可以得到一张原始状态表。
本节提出的问题是：这张状态表中的状态数是不是最少？这直接关系到电路的繁简和优化。

当采用硬件描述语言建模时，关系到PLD器件中逻辑资源的有效占用。

为求得最简状态表，需要我们将等价的状态从原始状态表中解析出来，进行化简后形成一张最简状态表（最小状态表）。



所谓状态化简，就是采用某种化简技术从原始状态表中消去多余状态，得到一个既能正确描述给定的逻辑功能，又能使所包含的状态数目达到最少的状态表——最小状态表。

最常用的化简方法——隐含表法

8.3.1 完全给定同步时序电路状态表的化简

完全给定同步时序电路是指其状态表中的所有次态及输出都是确定的。

完全给定同步时序电路状态表的化简，是利用状态之间的等效关系进行的。

有关概念

状态等效

假设状态 S_A 和 S_B 是完全给定同步时序电路状态表中的两个状态，如果对于所有可能的输入序列，分别从 S_A 和 S_B 出发，所得到的输出响应序列完全相同，则两个状态是等效（等价）的，称 S_A 和 S_B 为等效对，记作： (S_A, S_B) 。

所有可能的输入序列，指输入序列的长度和结构是任意的。

从整体上讲，原始状态表已经反映了各状态在任意输入序列下的输出。

等效状态可以合并为一个状态，这种合并不会改变电路的外部特性。

等效状态的三个特点：

- 对称性：若 (S_A, S_B) ，则 (S_B, S_A) 。
- 自反性：对任何状态， (S_A, S_A) 。
- 传递性：若 (S_A, S_B) 且 (S_B, S_C) ，则 (S_A, S_C) 。

等效类

若干彼此等价的状态构成的集合。

由 (S_A, S_B) 和 (S_B, S_C) ，可以推出 (S_A, S_C) ，进而可知 S_A 、 S_B 、 S_C 属于同一等价类，记作：

$$(S_A, S_B), (S_B, S_C) \longrightarrow \{S_A, S_B, S_C\}$$

在等效关系中，等效对是狭义的概念，针对两个状态而言。等效类是广义的概念，针对若干个状态而言，甚至一个状态可称为等效类。

最大等效类

不是任何其它等效类子集的等效类称为最大等效类。

完全给定同步时序电路原始状态表的化简过程，就是寻找最大等效类，将每个最大等价类中的所有状态合并为一个新状态，从而得到最小状态表的过程。

化简后的状态数等于最大等效类的个数。

判断原始状态表中两个状态是否等效（等价）的标准：

如果两个状态，对每一位可能的输入都满足下列两个条件，则这两个状态等效。

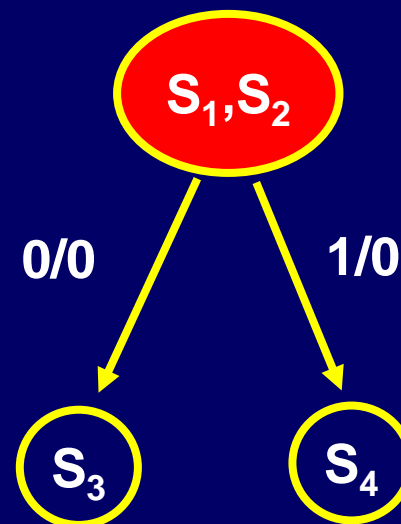
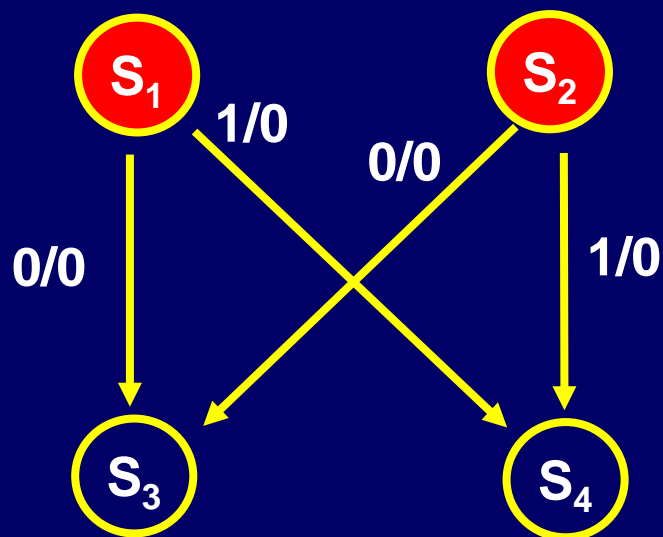
第一，它们的输出完全相同。

第二，它们的次态属于下列情况之一：

- 1) 次态相同
- 2) 次态交错或者次态维持
- 3) 后继状态等效
- 4) 次态循环

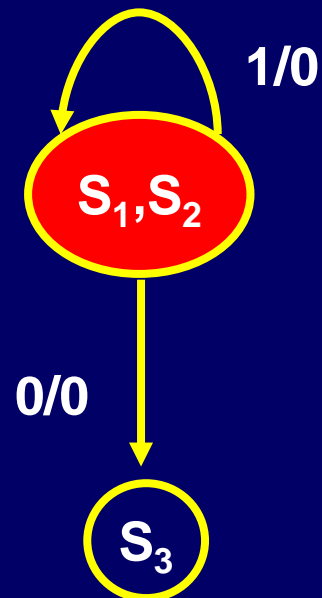
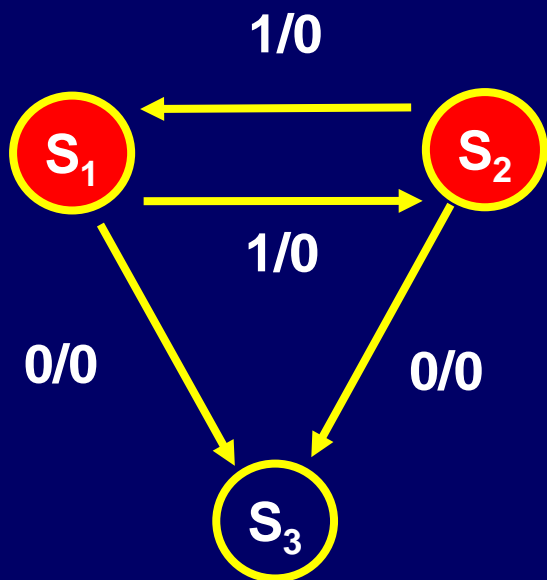
在原始状态图上判别状态的等效

输入/输出



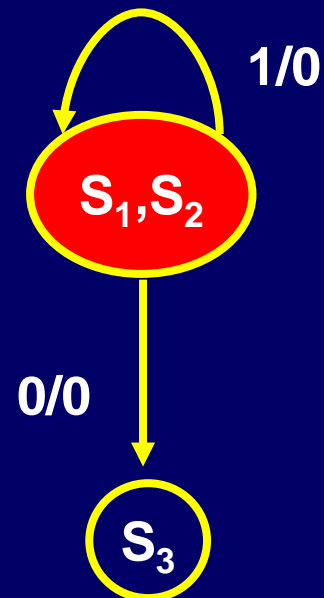
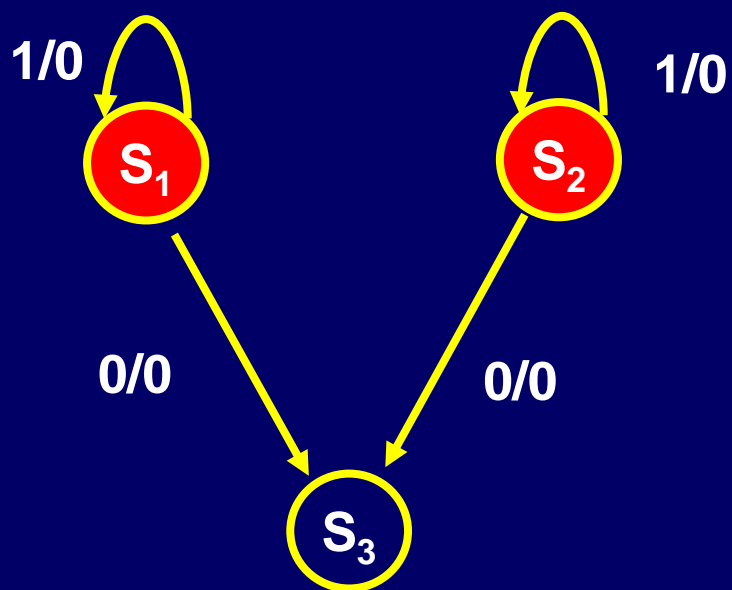
次态相同

输入/输出

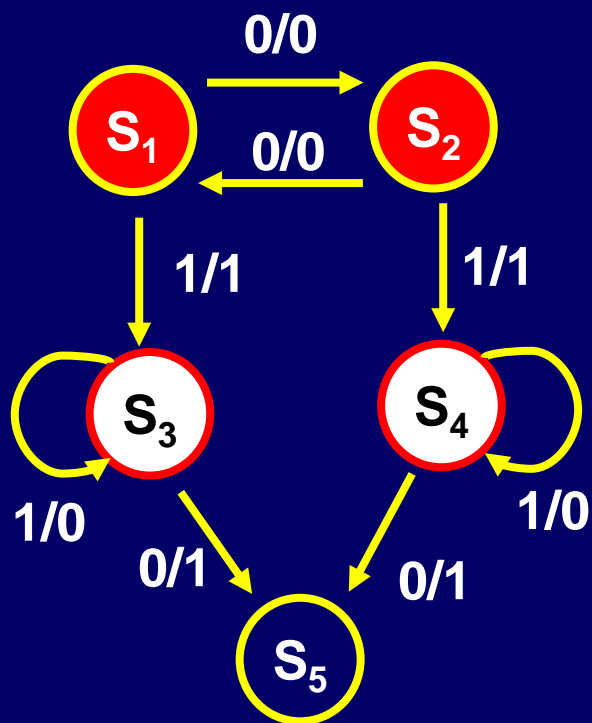


次态交错

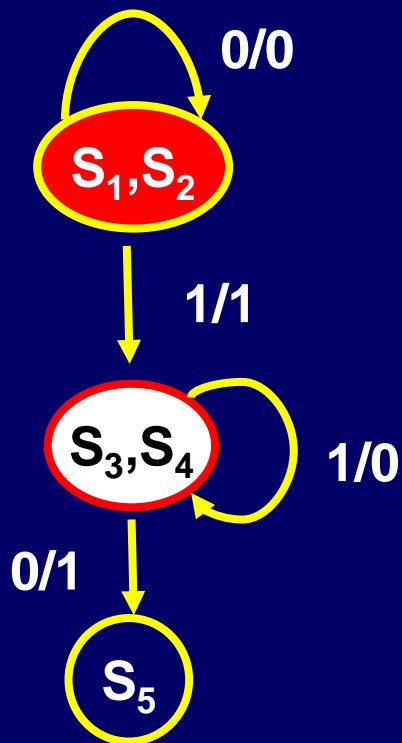
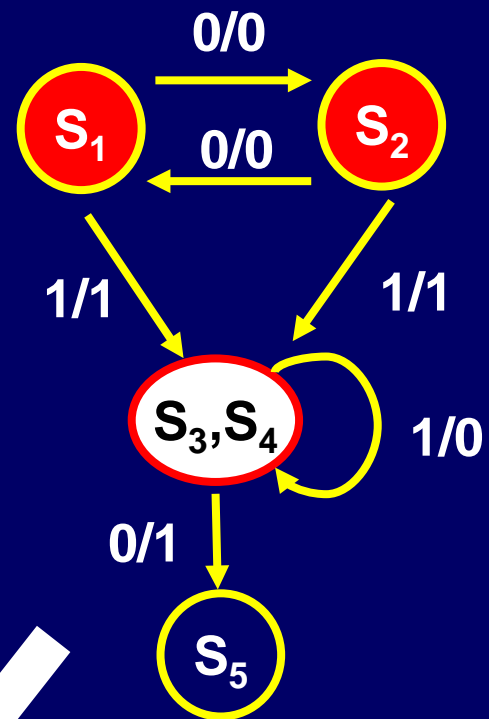
输入/输出

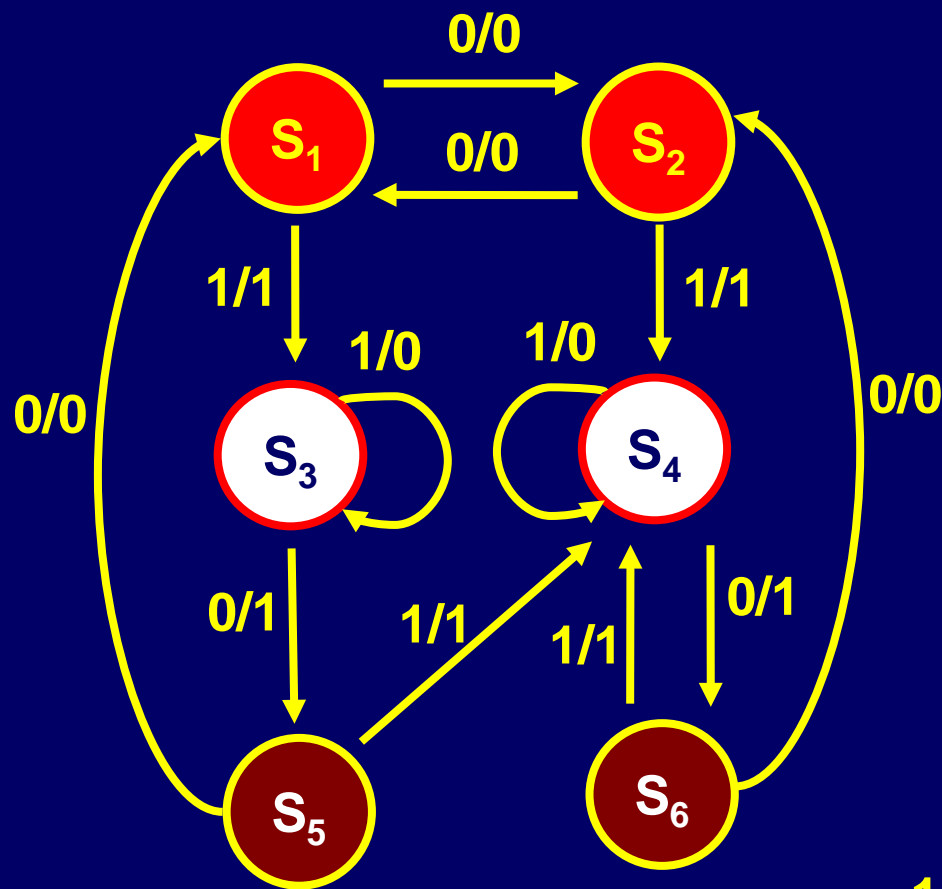


次态维持

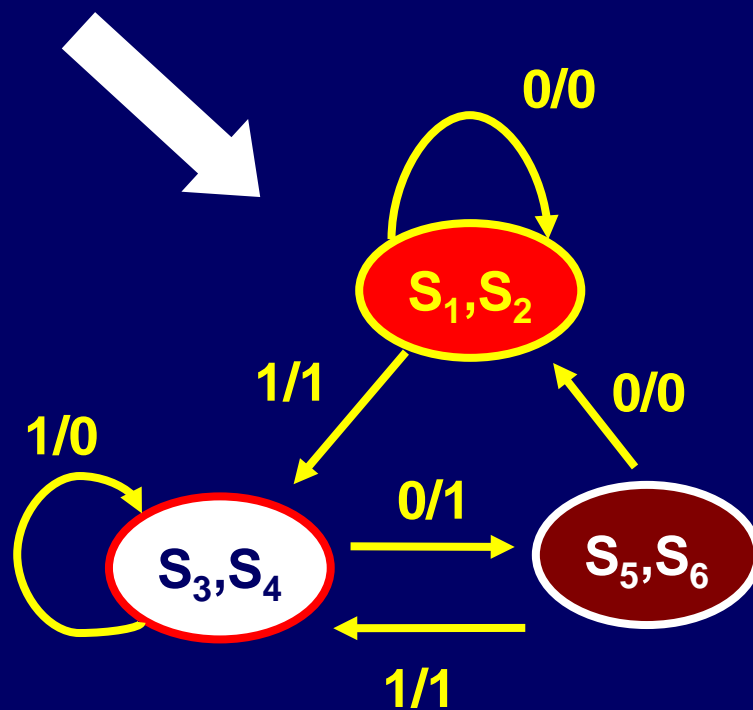


后继状态等效





次态循环



在原始状态表中判断状态的等效

$X \backslash Y$	0	1
A	B / 0	E / 1
B	A / 0	E / 1
C	A / 0	D / 1
D	E / 1	B / 0
E	E / 1	B / 0

$Y_{(t+1)} / Z$

输出不相等，则不等效。例如：C和D...

输出相等时：

- 1) 次态相等，等效。如状态D和E等效；
- 2) 次态交错，等效。如状态A和B等效；
- 3) 后继状态等效，等效。此例中B和C是否等效，要看E和D是不是等效，因为E和D等效，所以B和C等效。

根据等效的传递性可知，A和B等效，B和C等效，则A和C等效

等效对：(A, B) (B, C) (A, C) (D, E)



等效类： {A, B, C} {D, E}

最大等效类： {A, B, C} {D, E}

将所有最大等效类重新命名，令：

$$S_1 = \{ A, B, C \}$$

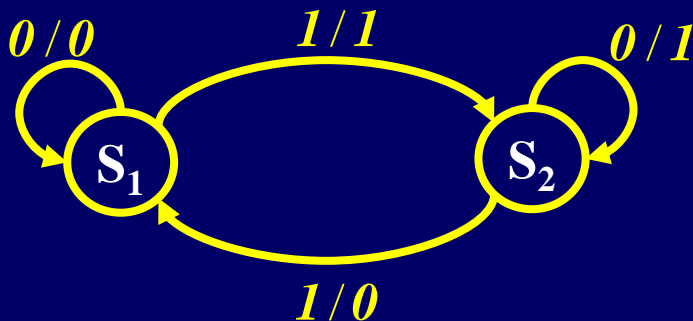
$$S_2 = \{ D, E \}$$

则可得化简后的状态表
(最小化状态表)：

$X \backslash Y$	0	1
A	B / 0	E / 1
B	A / 0	E / 1
C	A / 0	D / 1
D	E / 1	B / 0
E	E / 1	B / 0

$Y_{(t+1)} / Z$

化简后的状态图：



$X \backslash S$	0	1
S ₁	S ₁ / 0	S ₂ / 1
S ₂	S ₂ / 1	S ₁ / 0

$S_{(t+1)} / Z$

利用隐含表进行完全给定同步时序电路状态表的化简

一般步骤:

1) 作隐含表

2) 寻找等效对

3) 求出最大等效类

注意: a)各最大等效类之间不应出现相同状态

b)原始状态表中的每一个状态必须属于某一个最大等效类

4) 作出最小化状态表

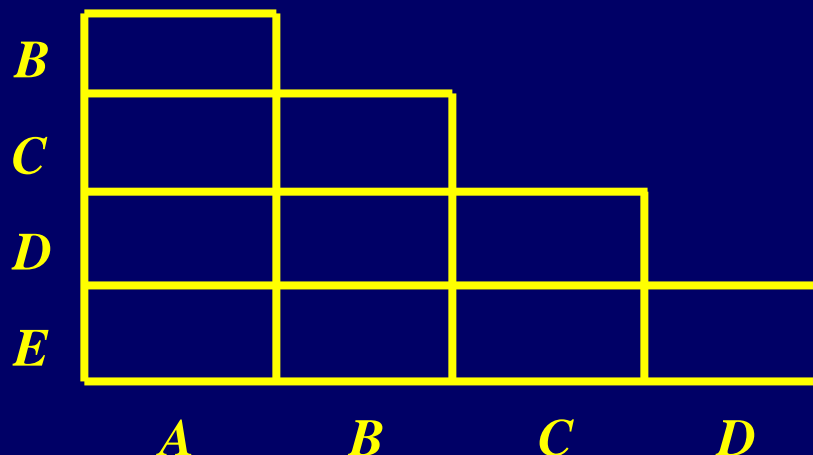
隐含表

隐含表是用来标注原始状态表中所有的状态对之间，按照等效的判定条件进行“状态对”比较的一种表格。

隐含表是一个直角三角形阶梯表，两直角边的网格数相同，它等于原始状态表中的状态数减 1，用状态名进行顺序标注。纵坐标从上到下标注且“缺头”（缺少第一个状态）；横坐标从左到右标注且“少尾”（缺少最后一个状态）。横纵坐标交汇的每个方格代表一个状态对。

$\begin{array}{c} X \\ \diagdown \\ Y \end{array}$	Z	
	0	1
A	B / 0	E / 1
B	A / 0	E / 1
C	A / 0	D / 1
D	E / 1	B / 0
E	E / 1	B / 0

$Y_{(t+1)} / Z$



例1：化简图示状态表。

$X \backslash Y$	0	1
A	C/1	B/0
B	C/1	E/0
C	B/1	E/0
D	D/1	B/1
E	D/1	B/1

$Y_{(t+1)} / Z$

B	BE			
C	BC BE	√		
D	X	X	X	
E	X	X	X	√
	A	B	C	D

1) 作隐含表

2) 求等效对

● 顺序比较

所有“状态对”逐一检查、比较。

等效：方格内画√；

不等效：方格内画x；

与其它状态对有关：方格内填写相关状态对。

●关联比较

若相关状态对都等效，则方格对应的状态对等效。不增加标志。

若相关状态对有一个不等效，则方格对应的状态对不等效。画 /。

<i>B</i>	BE			
<i>C</i>	BC BE	√		
<i>D</i>	X	X	X	
<i>E</i>	X	X	X	√
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>

等效对为： (B, C) , (D, E)

3) 求出最大等效类

利用等效状态的对称性、自反性、传递性，求出等效类。

$\{B, C\}$, $\{D, E\}$, $\{A\}$ 。

等效类 $\{B, C\}$, $\{D, E\}$, $\{A\}$ 均不包含在任何其他等效类中，所以 $\{A\}$, $\{B, C\}$, $\{D, E\}$ 是最大等价类。

4) 作最小化状态表

令 $S_1 = \{A\}$, $S_2 = \{B, C\}$, $S_3 = \{D, E\}$

$X \backslash S$	0	1
S_1	$S_2 / 1$	$S_2 / 0$
S_2	$S_2 / 1$	$S_3 / 0$
S_3	$S_3 / 1$	$S_2 / 1$

$S_{(t+1)} / Z$

$X \backslash Y$	0	1
A	$C / 1$	$B / 0$
B	$C / 1$	$E / 0$
C	$B / 1$	$E / 0$
D	$D / 1$	$B / 1$
E	$D / 1$	$B / 1$

$Y_{(t+1)} / Z$

例2：化简图示原始状态表

现态	次态/输出	
	输入X=0	输入X=1
A	C/0	B/1
B	F/0	A/1
C	F/0	G/0
D	D/1	E/0
E	C/0	E/1
F	C/0	G/0
G	C/1	D/0

因为CF等效，所以AB等效

B	CF					
C	X	X				
D	X	X	X			
E	BE	AE CF	X	X		
F	X	X	✓	X	X	
G	X	X	X	CD DE	X	X
	A	B	C	D	E	F

CF等效且AE, BE次态循环，所以AE等效，BE也等效。

请同学自己求出最大等效类、作出最小状态表

作业20:

8.5 (a)

8.14

8.4 状态编码（状态分配）

对最小化状态表中用字符表示的状态进行编码的过程，称为状态编码（状态分配）。

目的：构造卡诺图形式的二进制状态表。

用传统的 触发器+组合逻辑 实现。

用Verilog HDL建模。

例：对下列最小化状态表进行状态编码。

$Y \backslash X$	0	1
A	C/0	D/0
B	C/0	A/0
C	B/0	D/0
D	A/1	B/1

$Y_{(t+1)} / Z_{(t)}$

解：有4个状态，需要2位二进制，
即需2个触发器，设为 Q_2Q_1 。

1) 按二进制自然码编码

Y	$Q_2 Q_1$		$Q_2 Q_1 \backslash X$	0	1
A	0	0	→ 00	10/0	11/0
B	0	1	→ 01	10/0	00/0
C	1	0	↘ 11	00/1	01/1
D	1	1	↗ 10	01/0	11/0

$Q_{2(t+1)} Q_{1(t+1)} / Z_{(t)}$

2) 按格雷码编码

编码方式不同，
二进制状态表不同。

Y	$Q_2 Q_1$	
A	0	0
B	0	1
C	1	1
D	1	0

$Q_2 Q_1 \backslash X$		0	1
00	→	11 / 0	10 / 0
	→	11 / 0	00 / 0
01	→	01 / 0	10 / 0
	→	00 / 1	01 / 1

$Y \backslash X$		0	1
A	→	C / 0	D / 0
	→	C / 0	A / 0
B	→	B / 0	D / 0
	→	A / 1	B / 1

$Y_{(t+1)} / Z_{(t)}$

$Q_{2(t+1)} Q_{1(t+1)} / Z_{(t)}$



求状态方程和输出方程，
用D触发器或JK触发器
实现。

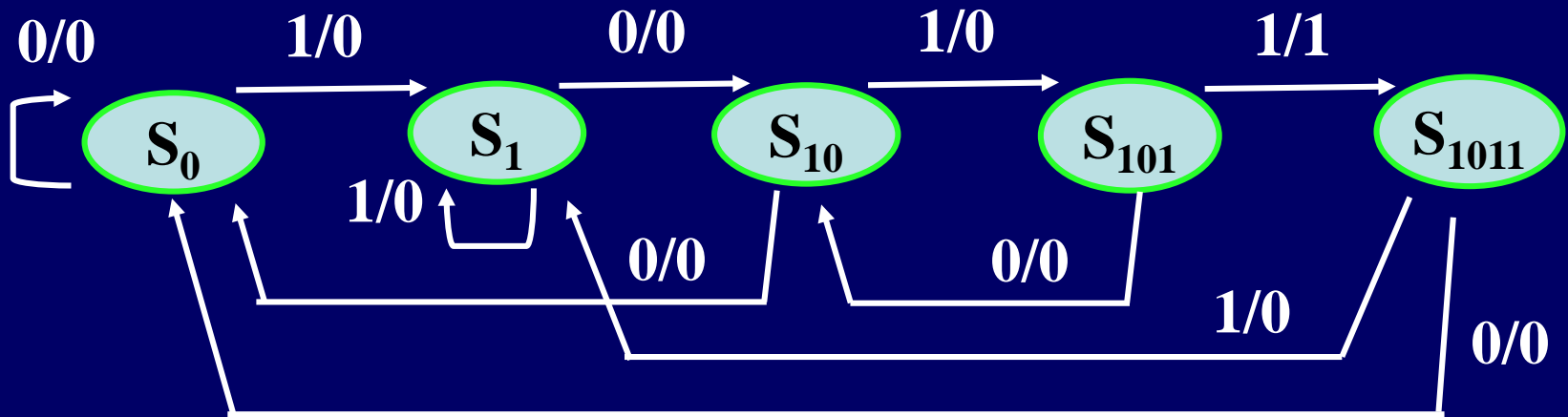
也可用Verilog HDL建模。

下面有关状态分配的介绍，不作为
教学要求，供学生自学时参考。



8.5 一般同步时序电路设计举例

例1：设计一个“1011”序列检测器。（不可重）



S \ X	0	1
S ₀	S ₀ / 0	S ₁ / 0
S ₁	S ₁₀ / 0	S ₁ / 0
S ₁₀	S ₀ / 0	S ₁₀₁ / 0
S ₁₀₁	S ₁₀ / 0	S ₁₀₁₁ / 1
S ₁₀₁₁	S ₀ / 0	S ₁ / 0

化简
→

Y \ X	0	1
Y ₀	Y ₀ / 0	Y ₁ / 0
Y ₁	Y ₂ / 0	Y ₁ / 0
Y ₂	Y ₀ / 0	Y ₃ / 0
Y ₃	Y ₂ / 0	Y ₀ / 1

$Y \backslash X$	0	1
Y_0	$Y_0 / 0$	$Y_1 / 0$
Y_1	$Y_2 / 0$	$Y_1 / 0$
Y_2	$Y_0 / 0$	$Y_3 / 0$
Y_3	$Y_2 / 0$	$Y_0 / 1$

编码



$Q_2 Q_1 \backslash X$	0	1
00	00 / 0	01 / 0
01	11 / 0	01 / 0
11	00 / 0	10 / 0
10	11 / 0	00 / 1

$$Q_{2(t+1)} = \overline{Q_2} Q_1 \overline{X} + Q_2 \overline{Q_1} \overline{X} + Q_2 Q_1 X$$

$$= \overline{Q_2} \cdot Q_1 \overline{X} + Q_2 \cdot (\overline{Q_1} \overline{X} + Q_1 X)$$

$$= \overline{Q_2} \cdot Q_1 \overline{X} + Q_2 \cdot \overline{(Q_1 \oplus X)}$$

$$Q_{2(t+1)} Q_{1(t+1)} / Z$$

状态方程

$$Q_{1(t+1)} = \overline{Q_2} Q_1 + \overline{Q_2} \overline{Q_1} X + Q_2 \overline{Q_1} \overline{X} \quad \text{用D触发器, 画图?}$$

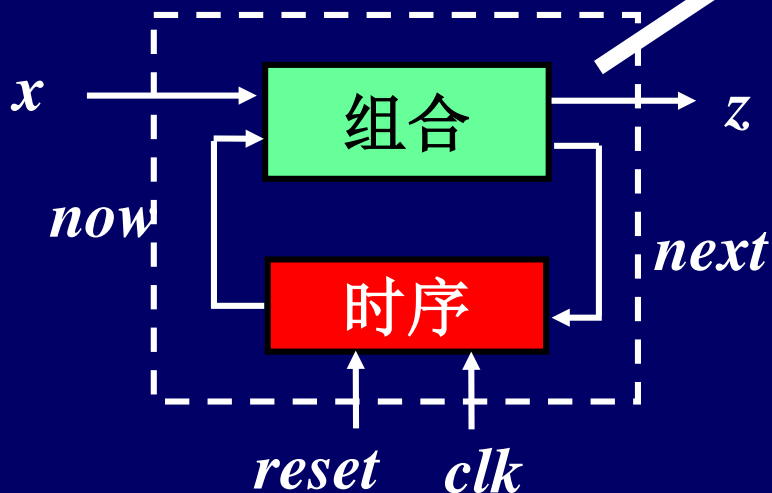
$$= (Q_2 \oplus X) \cdot \overline{Q_1} + \overline{Q_2} \cdot Q_1 \quad \text{用JK触发器, 画图?}$$

$$Z = Q_2 \overline{Q_1} X$$

采用Verilog HDL建模

$Q_2Q_1 \backslash X$	0	1
00	00 / 0	01 / 0
01	11 / 0	01 / 0
11	00 / 0	10 / 0
10	11 / 0	00 / 1

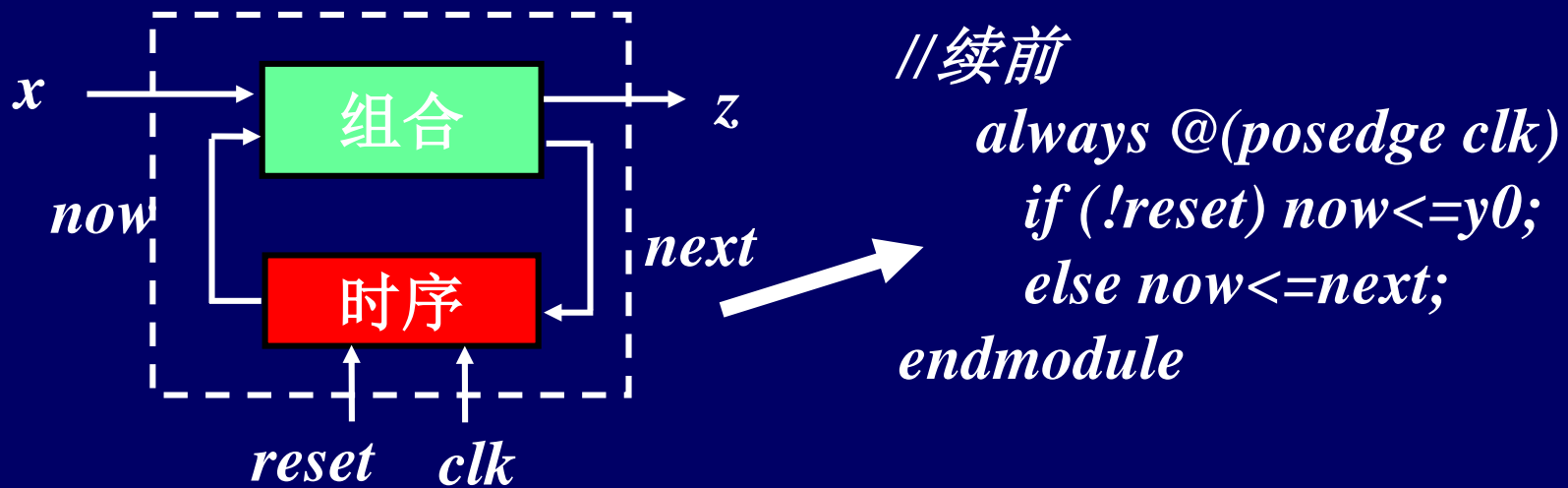
$Q_{2(t+1)}Q_{1(t+1)}/Z$



```

module t_1011(reset,clk,x,z);
    input  reset, clk, x;
    output z;
    parameter y0=2'b00, y1=2'b01,
              y2=2'b11, y3=2'b10;
    reg  z;
    reg [2:1] now, next;
    always @(x or now)
        case (now)
            y0 : if (x) begin next=y1;z=0;end
                  else begin next=y0;z=0;end
            y1 : if (!x) begin next=y2;z=0;end
                  else begin next=y1;z=0;end
            y2 : if (x) begin next=y3;z=0;end
                  else begin next=y0;z=0;end
            y3 : if (x) begin next=y0;z=1;end
                  else begin next=y2;z=0;end
            default : begin next=y0;z=0;end
        endcase
endcase
    
```

后续



设计模型要点:

- 1) 用parameter 进行状态编码
- 2) 定义现态、次态变量
- 3) 用一个always描述组合部分
- 4) 用另一个always描述时序部分

同步时序电路小结

一. 触发器逻辑符号（**SR**、**JK**、**D**、**T**），建立概念：

有效沿

状态方程（次态方程、特性方程）

功能表（特性表）

波形图

二. 同步时序电路的分析

1) 基于触发器设计的时序电路分析

分析方法（步骤）

通过课件例题及作业，掌握步骤，灵活应用。

难点：由状态方程→状态表→状态图



记忆：函数与卡诺图的关系

注意：Moore? Mealy?

如何正确画出时序图？

先画时钟，有效沿
输入序列

触发器状态的改变与保持
输出序列

2) 基于MSI时序器件的电路分析

74LS163计数器——单片应用： $M < 16$ 的计数器（跳越的概念及应用）
多片级联：较大模数计数器电路的分析
分频器电路的分析

以**74LS163**为核心的电路形式较多，例如**163+逻辑门**、**163+151**、**163+138**、..... 重点是：画出状态转换图，分析功能

74LS194移位寄存器——熟练掌握逻辑特性，看懂功能表。

以**74LS194**为核心的电路形式较多，例如**194+逻辑门**、**194+151**、**194+138**、..... 重点是：画出状态转换图，分析功能

分析方法——从启动开始，获得初态（现态），根据反馈逻辑计算产生次态的各项条件（各输入端的状态），根据功能表，确定在时钟有效沿作用下进入的下一个状态（次态），以它作为现态，重复上面的步骤，直到出现状态循环为止。

若电路较复杂，可划成功能块分别进行，某功能块的输出看作下一功能块的输入。例如：**163+138**模式、**194+138**模式。

三.典型同步时序电路的设计（Verilog HDL建模）

1) 根据逻辑命题，画出状态图，建模。

2) 根据给定功能表，建模。

3) 根据给定时序图，画状态图，建模。

自然二进制编码、格雷码计数器；

8421码、余三码、2421码、格雷BCD十进制计数器；

左循环一个“0” 左循环一个“1”

右循环一个“0” 右循环一个“1”

扭环形移位计数器（步进码）的设计，例如：四位、三位、模5、模7.....

四.一般同步时序电路的设计（Verilog HDL建模）

步骤：

原始状态图和原始状态表的建立

状态化简——隐含表法

得到最简状态表

状态编码：自然二进制码，格雷码

得到卡诺图形式状态表

Verilog HDL建模

建议：以序列检测器（四位，例如：**0001**）为例，进行完整设计训练。

基于Verilog HDL的同步时序电路设计模板

```
module 模块名（端口名列表）；  
    parameter //定义参数  
    input      //定义输入  
    output     //定义输出  
    reg        //定义always中的被赋值变量  
    //门级描述组合部分  
    assign     //描述组合部分  
    always @ (组合输入信号列表)  
        begin  
            //阻塞赋值（=）  
            //if_else、case、for行为语句  
        end  
    always @(边沿信号列表) // 时序部分描述  
        begin  
            //非阻塞赋值（<=）  
            //if_else、case、for行为语句  
        end  
endmodule
```