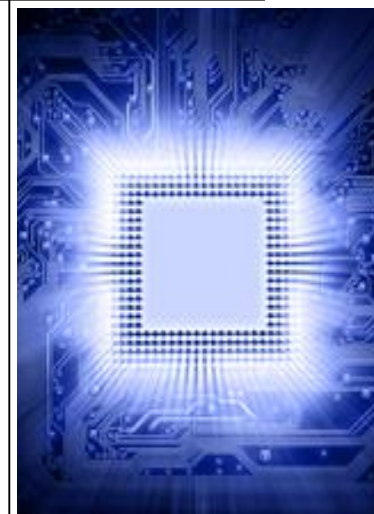


# 第7章 时序逻辑设计原理

(part-2)

武港山

南京大学人工智能学院





# 内容简介



- 双稳态电路(Bi-stable Device)
- 锁存器与触发器
- 时钟同步状态机分析
- 时钟同步状态机设计
  - 用状态表设计状态机
  - 用状态图设计状态机
  - 用转移表设计状态机
- 反馈时序电路分析
- 反馈时序电路设计
- Verilog设计时序电路

分析工具  
分析方法  
例题讲解



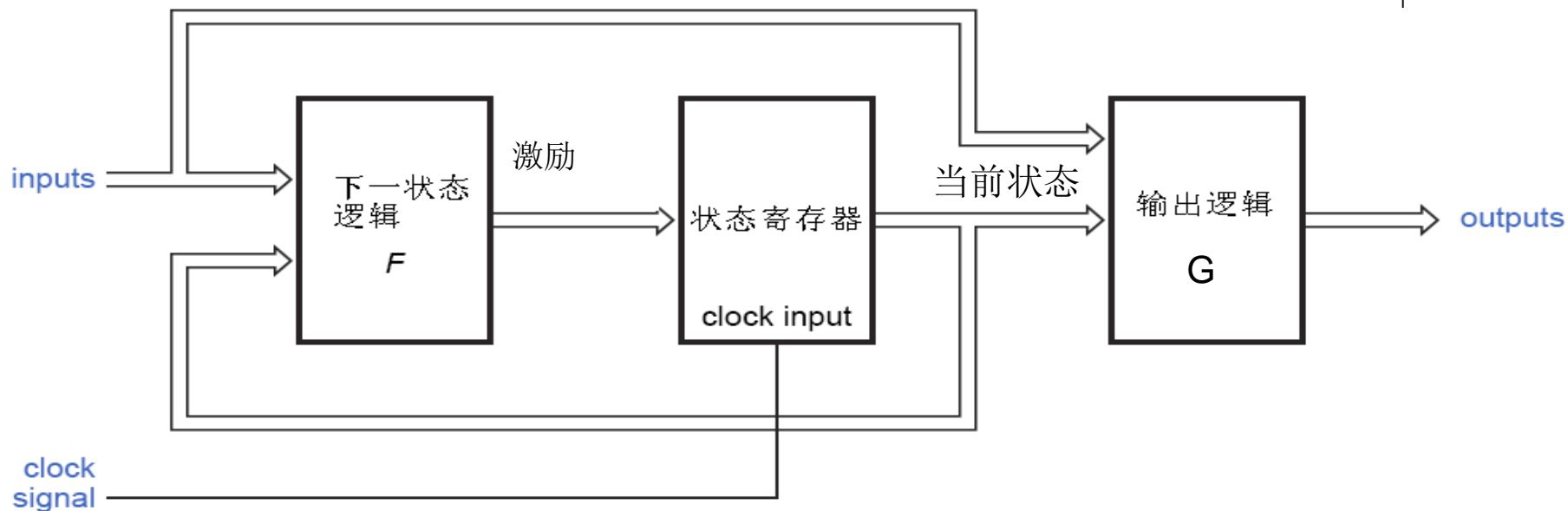
# 时钟控制时序逻辑电路分析



- **Sequential-Circuit: 组合逻辑+记忆电路**
  - 同步时序电路 (Synchronous Sequential Logic)
  - 异步时序电路 (Asynchronous Sequential Logic)
- **Synchronous Sequential Logic**
  - 统一时钟驱动，触发时刻更新记忆电路状态
  - 状态转移图描述
  - 设计方法、工具相对规范成熟
- **Asynchronous Sequential Logic**
  - 状态变化时刻不一致
  - 状态转移图描述
  - 异步电路多用在接口电路设计方面



# 同步时序电路的结构



- 状态寄存器：存储状态机当前状态（ $n$ 个触发器， $2^n$ 种不同状态）。
- 下一状态逻辑： $F$ （当前状态，输入）
- 输出逻辑： $G$ （当前状态，输入）
- $F$ 、 $G$ ：组合电路



# 同步时序电路的类型



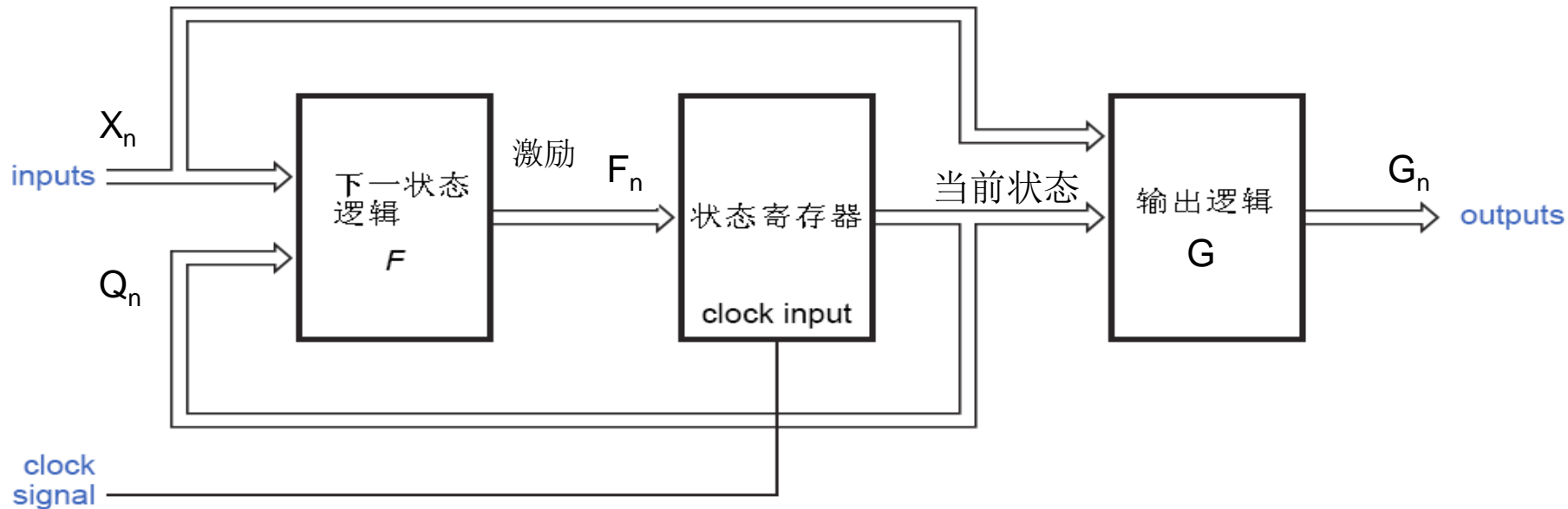
- 按输出方程的不同，分为：
  - Mealy型电路(*Mealy State Machine*)  
某时刻**输出**是该电路当前时刻输入和当前状态的函数
  - Moore型电路(*Moore State Machine*)  
某时刻**输出**仅是该电路当前状态的函数，与当前时刻的输入无关，如某些计数器



# Mealy型时序电路



- $Q_n$ 和 $X_n$ 作为记忆电路组合逻辑输入，产生激励信号 $F_n$   
激励方程： $F_n = F(X_n, Q_n)$
- 记忆电路新状态 $Q_{n+1}$ 取决于上一状态 $Q_n$ 和控制信号 $F_n$   
特征方程： $Q_{n+1} = Q(F_n, Q_n)$
- $Q_n$ 作为输出驱动，输出方程： $G_n = G(X_n, Q_n)$

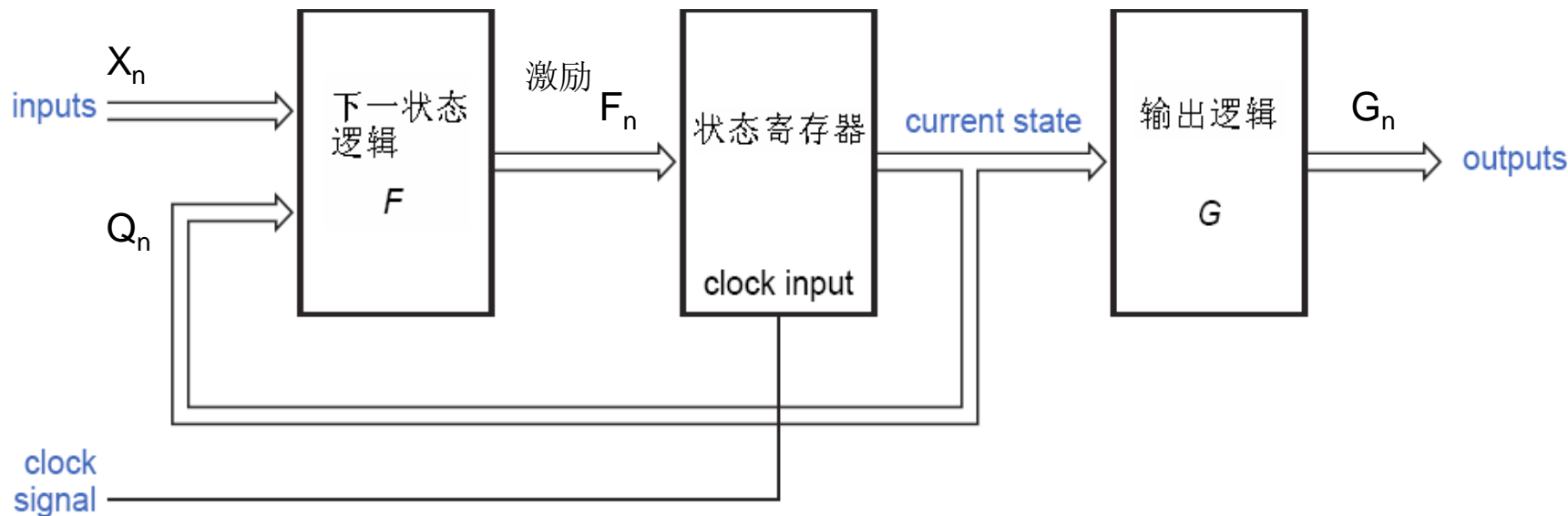




# Moore型时序电路



- $Q_n$  作为记忆电路组合逻辑输入，产生激励信号  $F_n$   
激励方程：  $F_n = F(X_n, Q_n)$
- 记忆电路新状态  $Q_{n+1}$  取决于上一状态  $Q_n$  和控制信号  $F_n$   
特征方程：  $Q_{n+1} = Q(F_n, Q_n)$
- $Q_n$  作为输出驱动，输出方程：  $G_n = G(Q_n)$



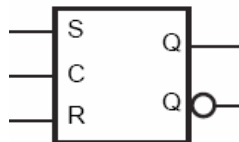


## 7.3.3 特征方程



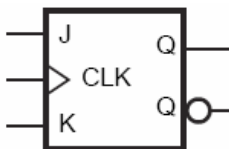
- 锁存器或触发器的功能特性采用特征方程进行形式描述
- $Q^*$ : 表示 $Q$ 的下一状态

SR锁存器



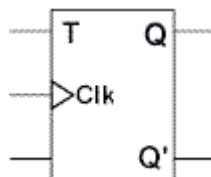
$$Q^* = S + R' \cdot Q$$

JK触发器



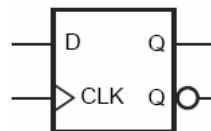
$$Q^* = J \cdot Q' + K' \cdot Q$$

T触发器



$$Q^* = Q'$$

D触发器



$$Q^* = D$$





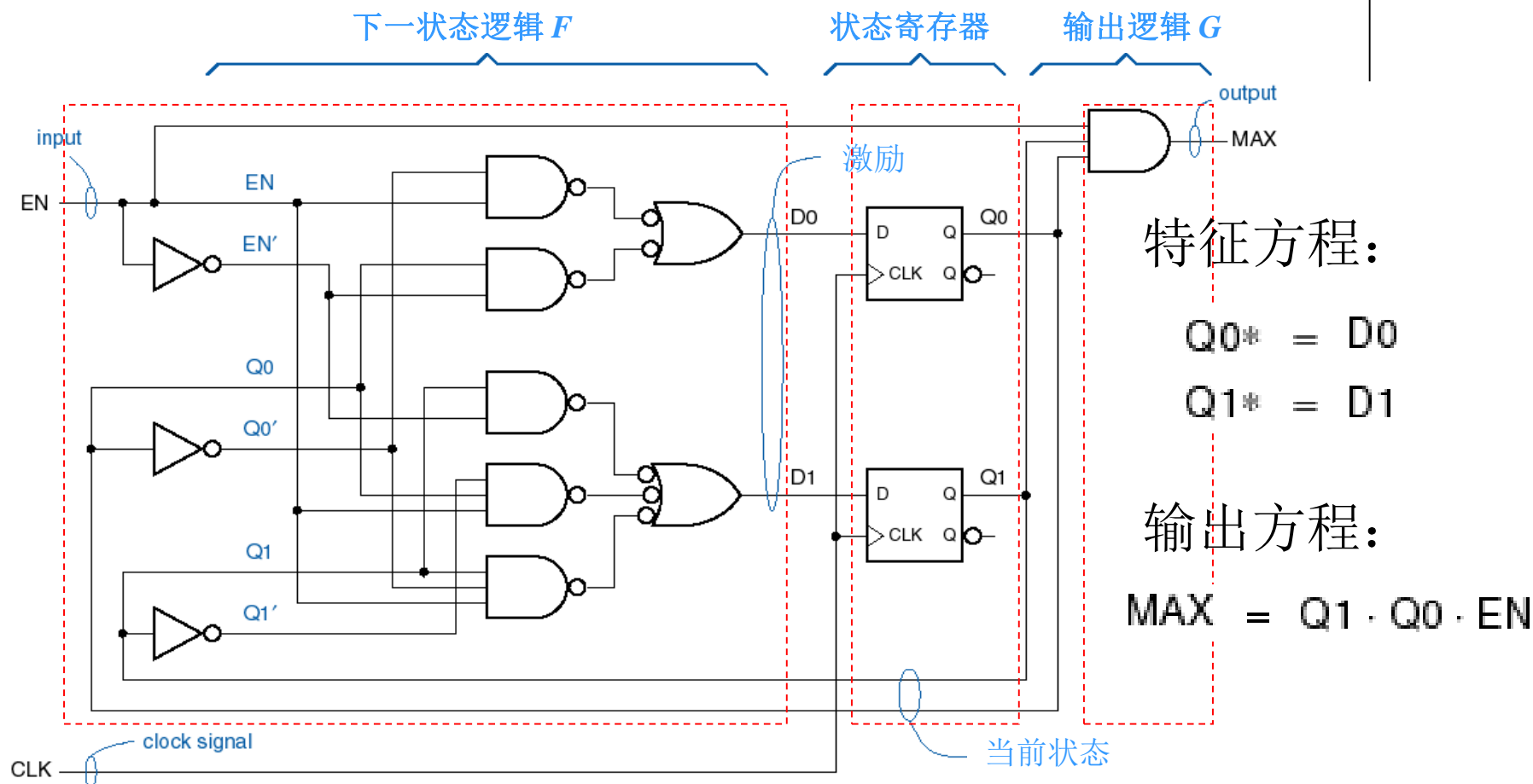
## 7.3.4 状态机分析的步骤



1. 确定下一状态函数**F**和输出函数**G**。
2. 用**F**和**G**构造**状态/输出表**(state/output table), 列出所有当前状态和输入的组合, 指定电路的下一状态和输出。
3. 画出**状态图** (state diagram) 。
4. 描述电路的功能。



# 时序电路举例—Mealy型电路





# 转移方程和状态表



转移方程：将激励方程带入特征方程

$$Q0^* = Q0 \cdot EN' + Q0' \cdot EN$$

$$Q1^* = Q1 \cdot EN' + Q1' \cdot Q0 \cdot EN + Q1 \cdot Q0' \cdot EN$$

输出方程：  $MAX = Q1 \cdot Q0 \cdot EN$

| <b><i>Q1 Q0</i></b>   | <b><i>EN</i></b> |                 |
|-----------------------|------------------|-----------------|
|                       | <b><i>0</i></b>  | <b><i>1</i></b> |
| 00                    | 00               | 01              |
| 01                    | 01               | 10              |
| 10                    | 10               | 11              |
| 11                    | 11               | 00              |
| <b><i>Q1* Q0*</i></b> |                  |                 |

转移表

状态符号化



| <b><i>S</i></b>  | <b><i>EN</i></b> |                 |
|------------------|------------------|-----------------|
|                  | <b><i>0</i></b>  | <b><i>1</i></b> |
| A                | A                | B               |
| B                | B                | C               |
| C                | C                | D               |
| D                | D                | A               |
| <b><i>S*</i></b> |                  |                 |

状态表

| <b><i>S</i></b>       | <b><i>EN</i></b> |                 |
|-----------------------|------------------|-----------------|
|                       | <b><i>0</i></b>  | <b><i>1</i></b> |
| A                     | A, 0             | B, 0            |
| B                     | B, 0             | C, 0            |
| C                     | C, 0             | D, 0            |
| D                     | D, 0             | A, 1            |
| <b><i>S*, MAX</i></b> |                  |                 |

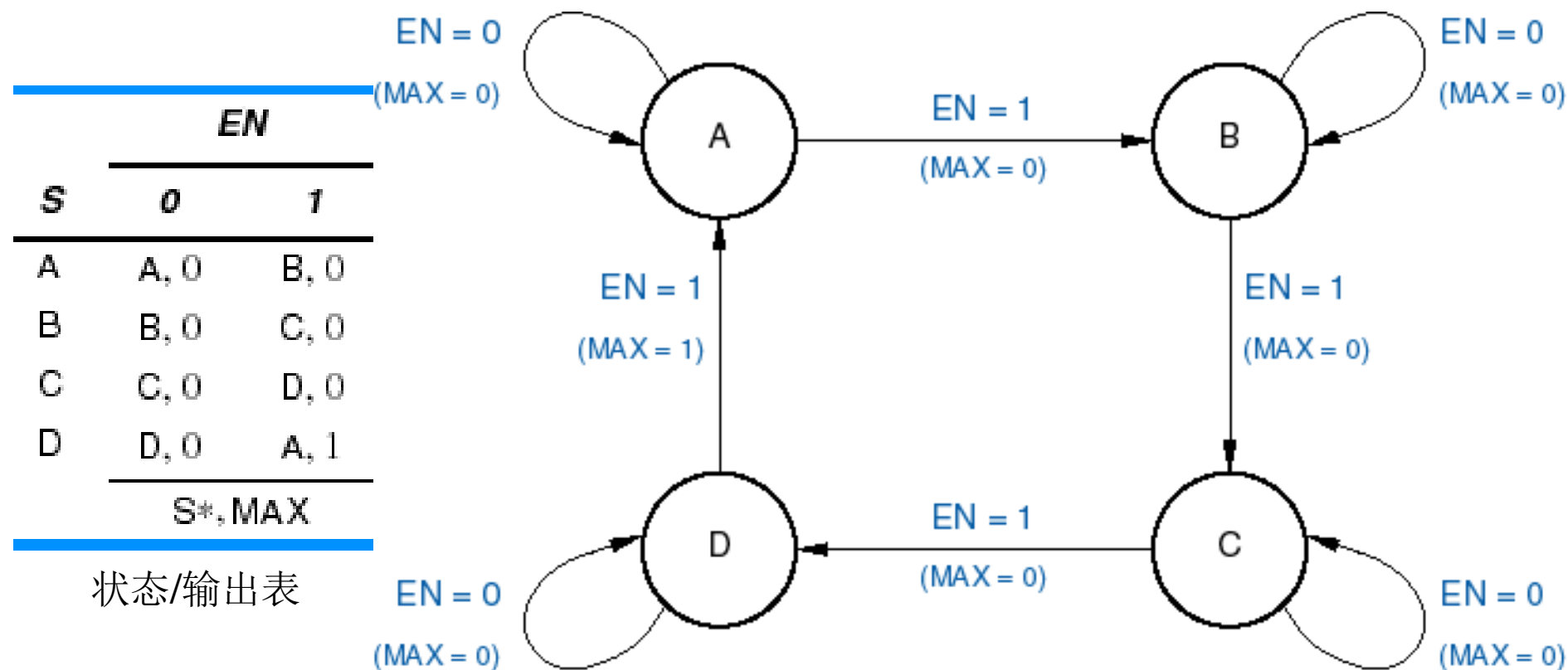
状态/输出表



# 状态图



- 状态图：以图形方式表示状态/输出表中的信息
  - 节点：对应着状态
  - 有向弧/箭头：表示一个转移





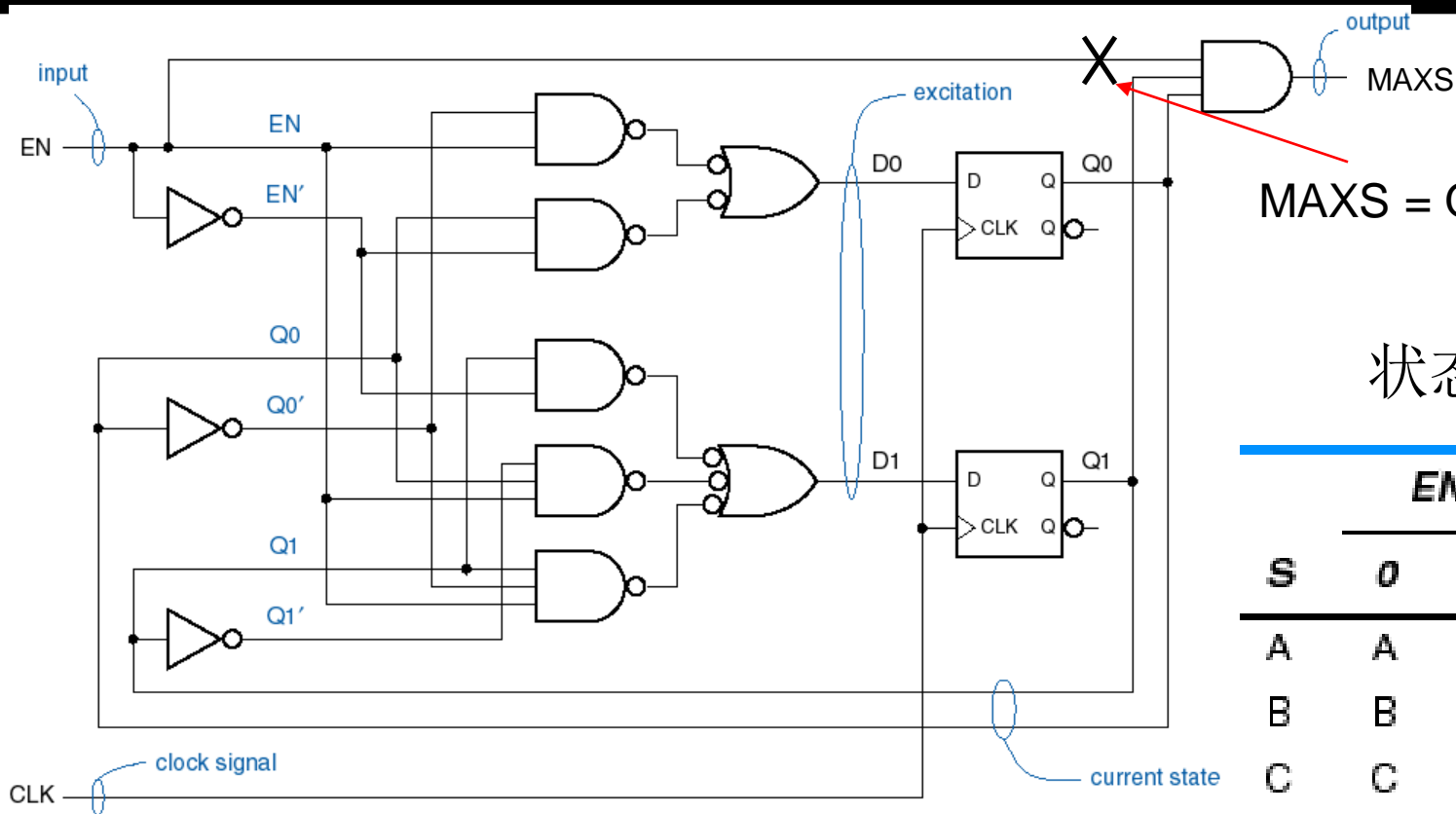
# 状态图



- State Diagram(状态图)中，离开某个特定状态的所有转移条件，满足
  - Mutually Exclusive(**互斥性**)，每种输入组合都有唯一的下一状态。否则相同输入组合对应不同的下一状态(二义性)
  - All Inclusion(**完备性**)，所有输入组合都有确定的下一状态(有时需要根据设计需求，在不违背题意的前提下作出合理安排)
  - 称为完全确定的时序电路



# 改为moore型电路



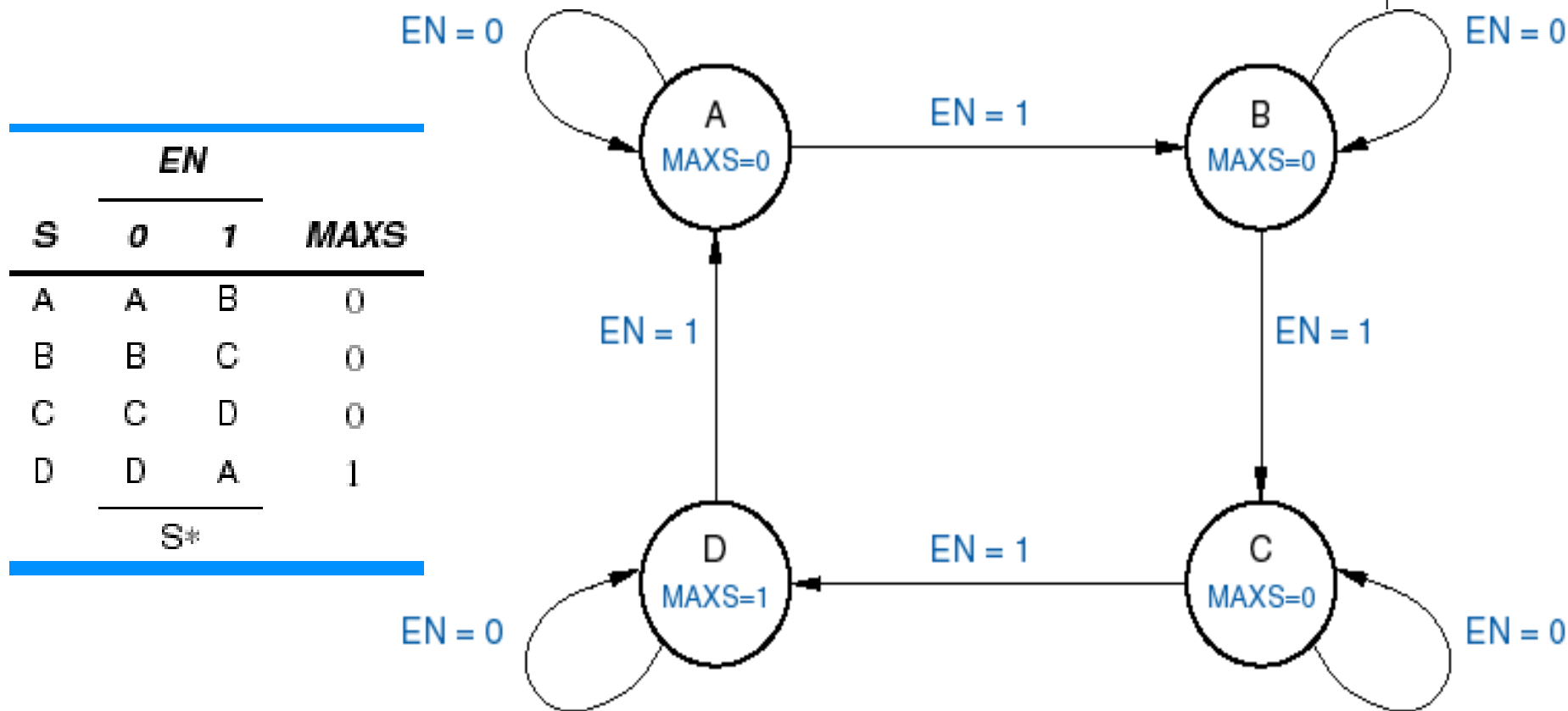
状态表

| S  | EN |   | MAXS |
|----|----|---|------|
|    | 0  | 1 |      |
| A  | A  | B | 0    |
| B  | B  | C | 0    |
| C  | C  | D | 0    |
| D  | D  | A | 1    |
| S* |    |   |      |

- Moore型电路输出仅依赖于当前状态



# Moore电路的状态图



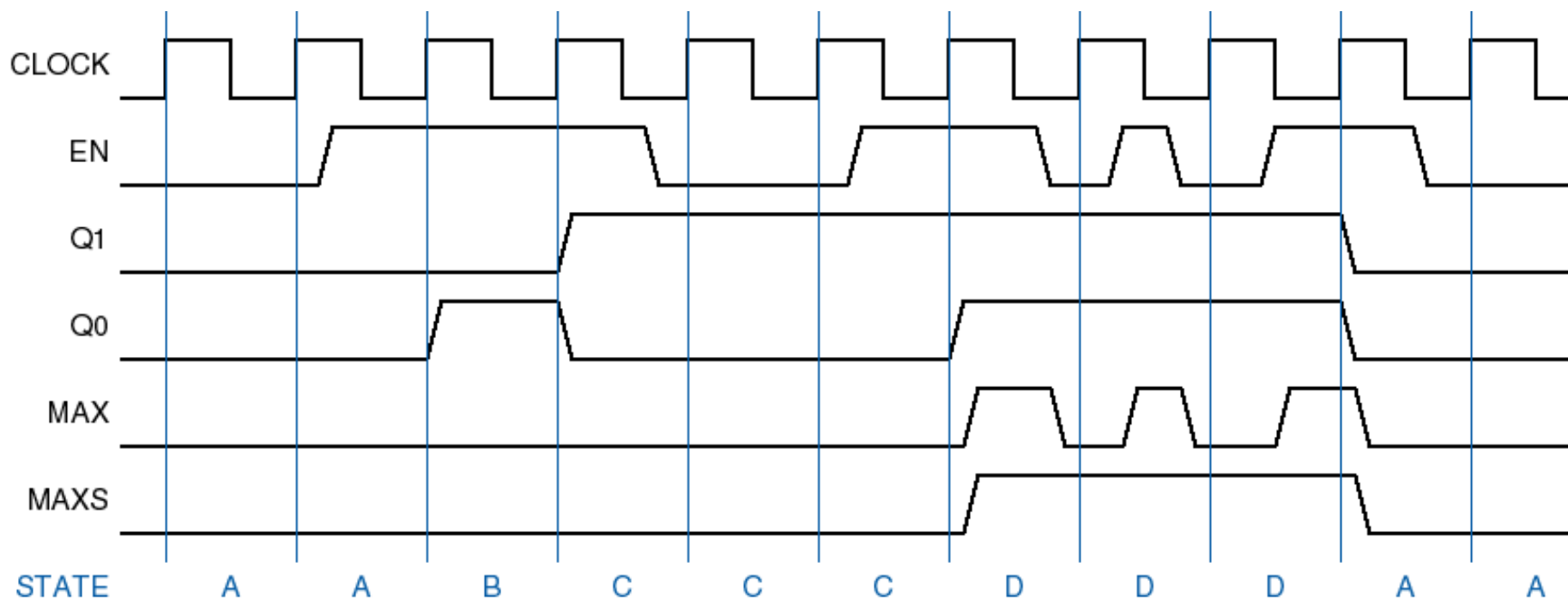


# 时序图Timing diagram



- 利用转移、状态和输出表，可构造定时图
- 定时图：表示出状态机在任何期望的起始状态和输入序列的作用下所产生的行为。

扫描触发器的功能体现在哪里？







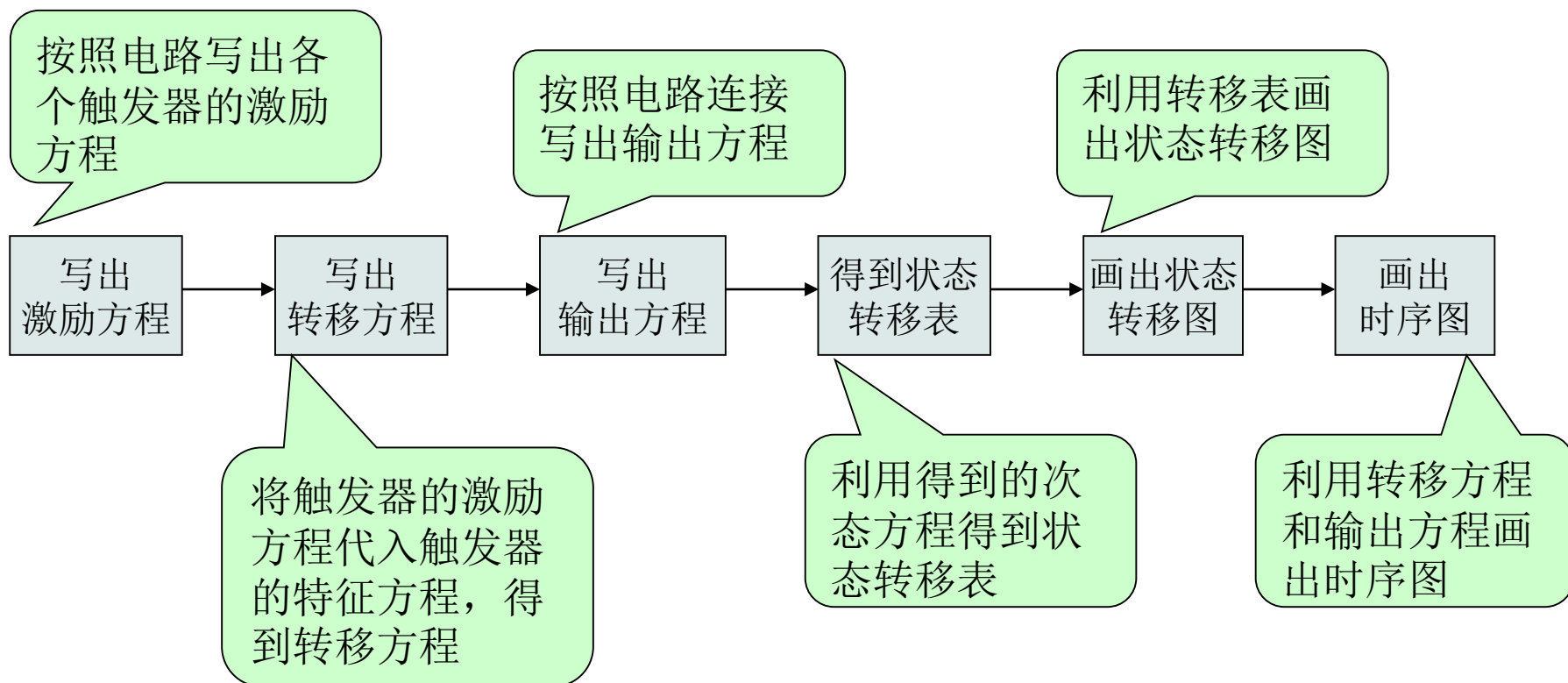
# 总结：同步时序电路的分析步骤



1. 写出各触发器的**激励方程**。
2. 把得到的激励方程代入到触发器的特征方程，得到**转移方程/次态方程**。
3. 确定**输出方程**。
4. 根据转移方程构造转移表，在转移表中对每一种状态/输入组合添加输出值，构成**状态/输出表**。
5. 利用状态名得到时序电路的**状态图**。
6. 画出**时序图**。

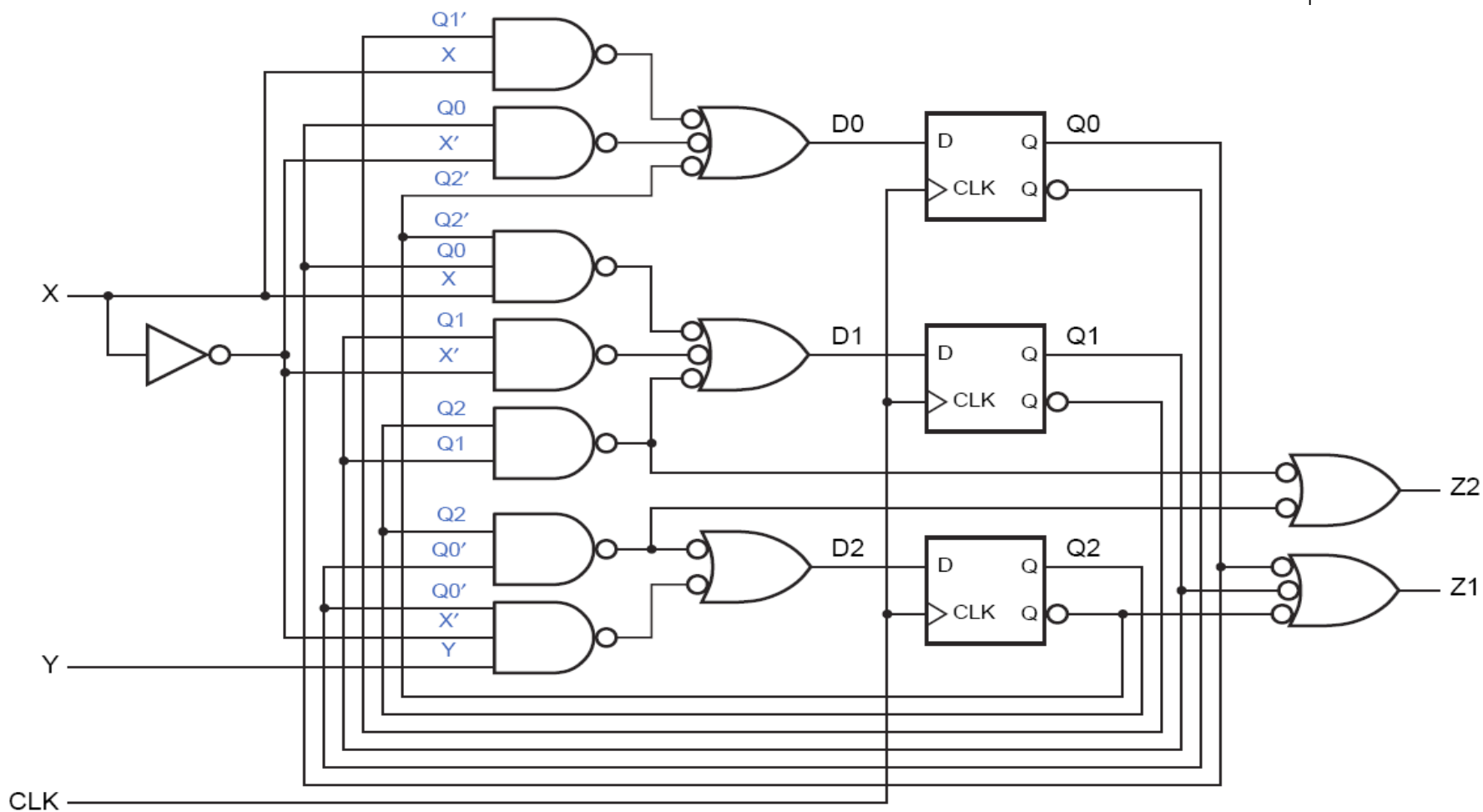


# 同步时序电路的分析方法



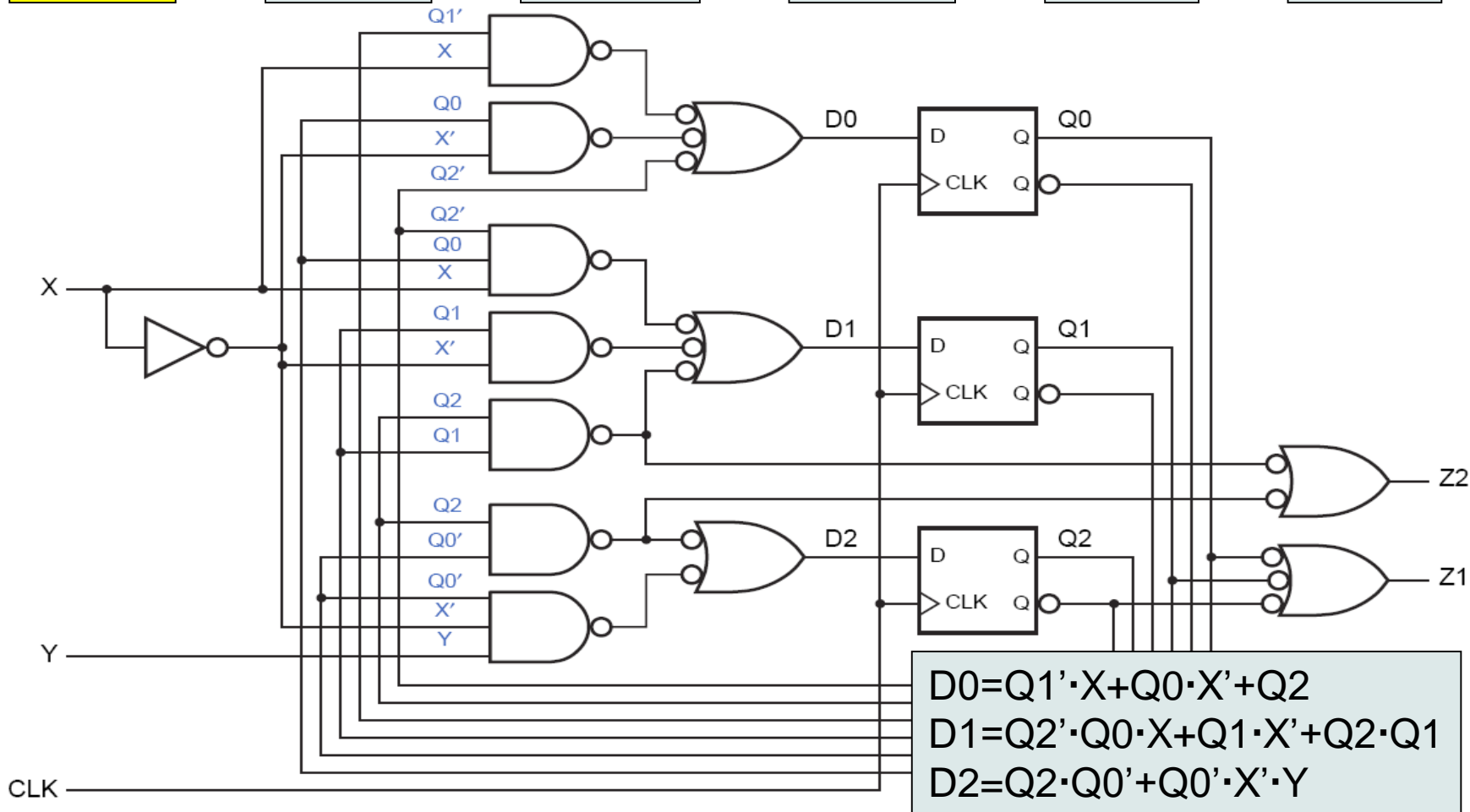
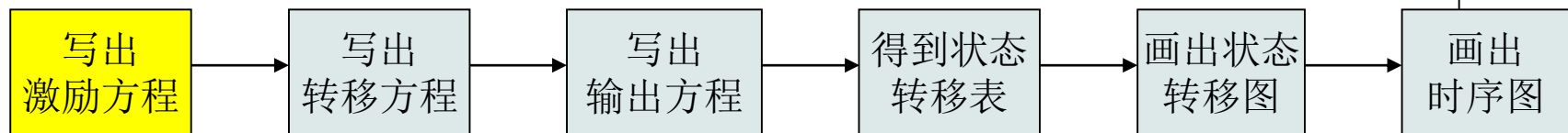


# 同步时序电路分析例1



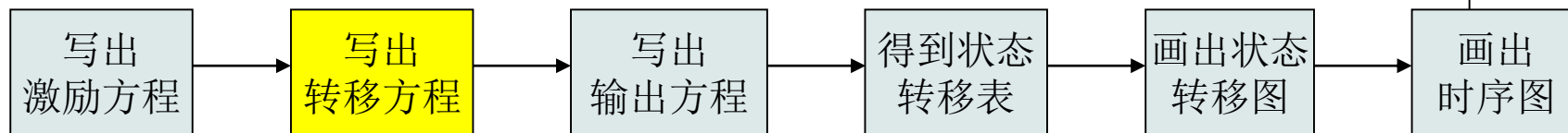


# 第一步：写出激励方程





## 第二步：得到转移方程



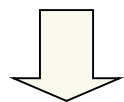
激励方程

$$D0=Q1' \cdot X + Q0 \cdot X' + Q2$$

$$D1=Q2' \cdot Q0 \cdot X + Q1 \cdot X' + Q2 \cdot Q1$$

$$D2=Q2 \cdot Q0' + Q0' \cdot X' \cdot Y$$

D触发器的特征方程  $Q^{n+1} = D$



转移方程

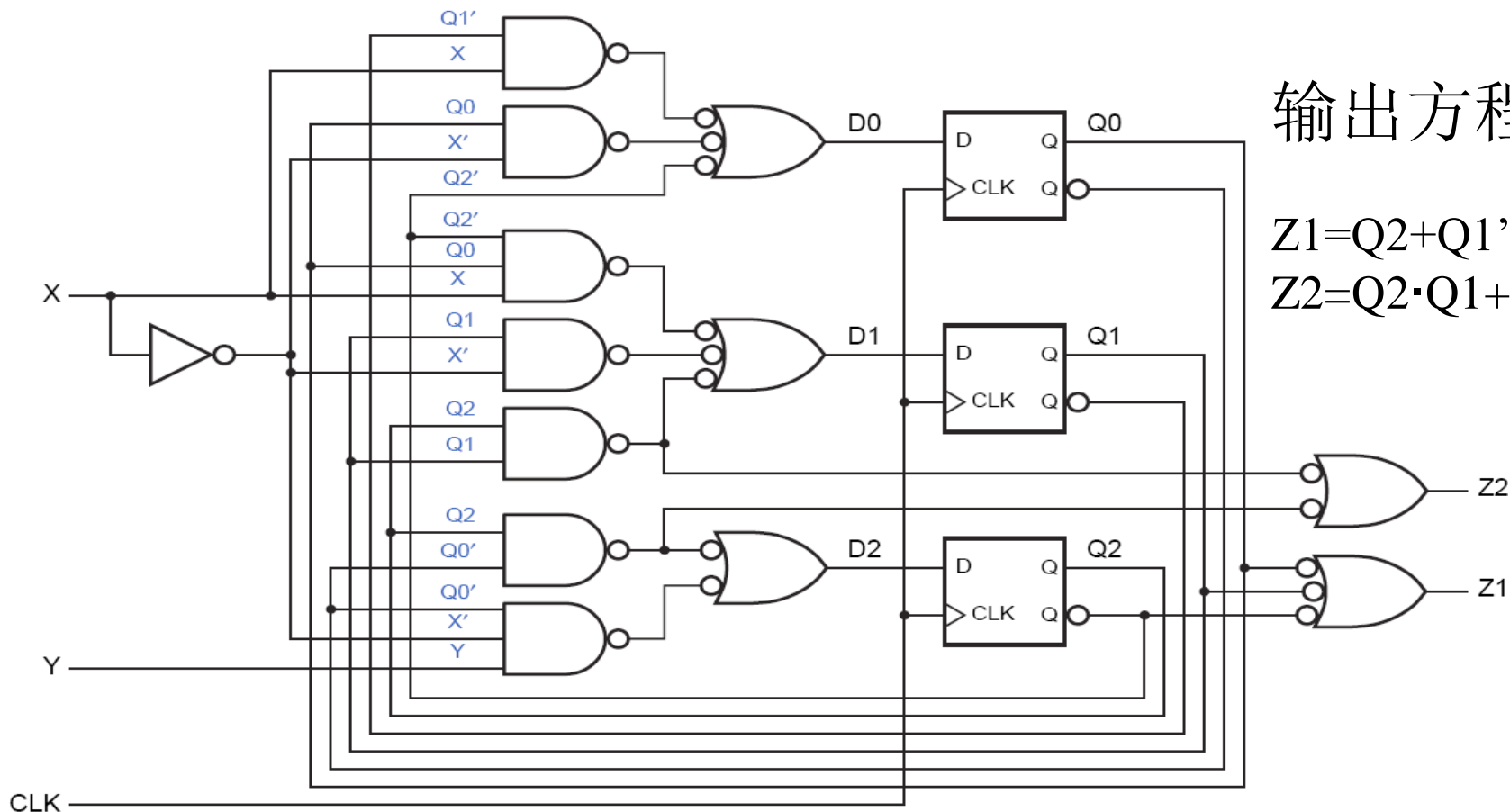
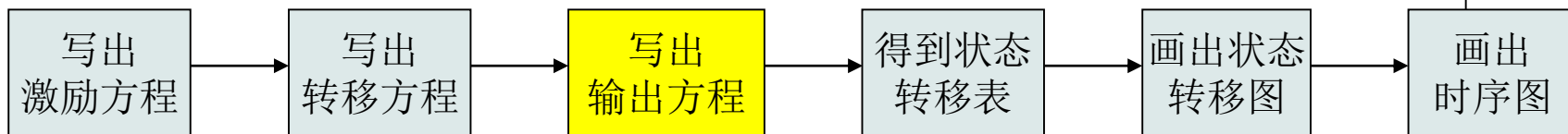
$$Q0^{n+1}=Q1' \cdot X + Q0 \cdot X' + Q2$$

$$Q1^{n+1}=Q2' \cdot Q0 \cdot X + Q1 \cdot X' + Q2 \cdot Q1$$

$$Q2^{n+1}=Q2 \cdot Q0' + Q0' \cdot X' \cdot Y$$

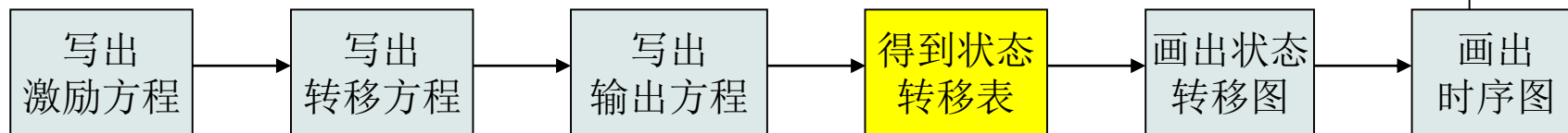


## 第三步：得到输出方程





## 第四步：构建转移表



$$Q0^{n+1} = Q1' \cdot X + Q0 \cdot X' + Q2$$

$$Q1^{n+1} = Q2' \cdot Q0 \cdot X + Q1 \cdot X' + Q2 \cdot Q1$$

$$Q2^{n+1} = Q2 \cdot Q0' + Q0' \cdot X' \cdot Y$$

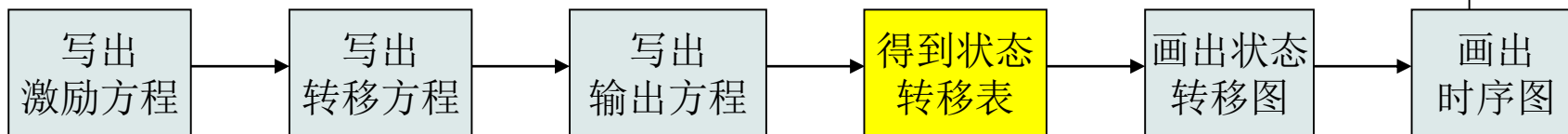
$$Z1 = Q2 + Q1' + Q0'$$

$$Z2 = Q2 \cdot Q1 + Q2 \cdot Q0'$$

|    |    |    | XY                           |    |    |    |    |    |
|----|----|----|------------------------------|----|----|----|----|----|
| Q2 | Q1 | Q0 | 00                           | 01 | 10 | 11 | Z1 | Z2 |
| 0  | 0  | 0  |                              |    |    |    |    |    |
| 0  | 0  | 1  |                              |    |    |    |    |    |
| 0  | 1  | 0  |                              |    |    |    |    |    |
| 0  | 1  | 1  |                              |    |    |    |    |    |
| 1  | 0  | 0  |                              |    |    |    |    |    |
| 1  | 0  | 1  |                              |    |    |    |    |    |
| 1  | 1  | 0  |                              |    |    |    |    |    |
| 1  | 1  | 1  |                              |    |    |    |    |    |
|    |    |    | $Q2^{n+1} Q1^{n+1} Q0^{n+1}$ |    |    |    |    |    |

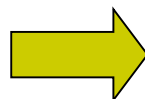


## 第四步：构建状态表



| XY                           |     |     |     |     |       |  |
|------------------------------|-----|-----|-----|-----|-------|--|
| Q2 Q1 Q0                     | 00  | 01  | 10  | 11  | Z1 Z2 |  |
| 000                          | 000 | 100 | 001 | 001 | 10    |  |
| 001                          | 001 | 001 | 011 | 011 | 10    |  |
| 010                          | 010 | 110 | 000 | 000 | 10    |  |
| 011                          | 011 | 011 | 010 | 010 | 00    |  |
| 100                          | 101 | 101 | 101 | 101 | 11    |  |
| 101                          | 001 | 001 | 001 | 001 | 10    |  |
| 110                          | 111 | 111 | 111 | 111 | 11    |  |
| 111                          | 011 | 011 | 011 | 011 | 11    |  |
| $Q2^{n+1} Q1^{n+1} Q0^{n+1}$ |     |     |     |     |       |  |

转移表



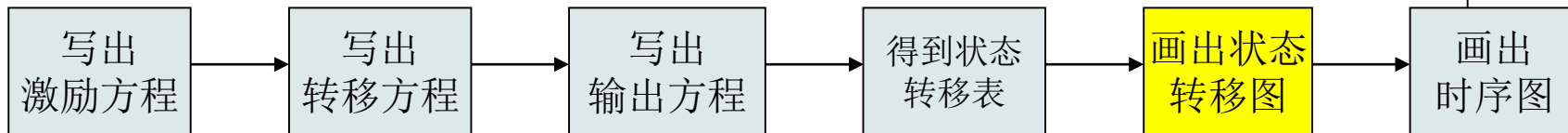
| XY        |    |    |    |    |       |
|-----------|----|----|----|----|-------|
| S         | 00 | 01 | 10 | 11 | Z1 Z2 |
| A         | A  | E  | B  | B  | 10    |
| B         | B  | B  | D  | D  | 10    |
| C         | C  | G  | A  | A  | 10    |
| D         | D  | D  | C  | C  | 00    |
| E         | F  | F  | F  | F  | 11    |
| F         | B  | B  | B  | B  | 10    |
| G         | H  | H  | H  | H  | 11    |
| H         | D  | D  | D  | D  | 11    |
| $S^{n+1}$ |    |    |    |    |       |

状态表





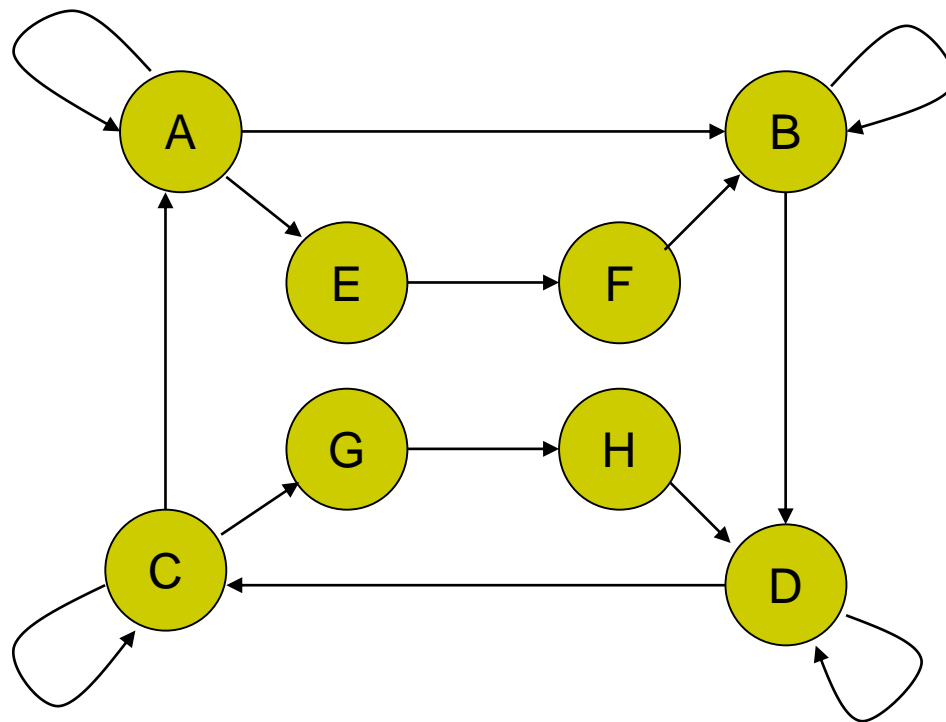
# 第五步：画出状态转移图



| S | XY |    |    |    | Z1 Z2 |
|---|----|----|----|----|-------|
|   | 00 | 01 | 10 | 11 |       |
| A | A  | E  | B  | B  | 10    |
| B | B  | B  | D  | D  | 10    |
| C | C  | G  | A  | A  | 10    |
| D | D  | D  | C  | C  | 00    |
| E | F  | F  | F  | F  | 11    |
| F | B  | B  | B  | B  | 10    |
| G | H  | H  | H  | H  | 11    |
| H | D  | D  | D  | D  | 11    |

$S^{n+1}$

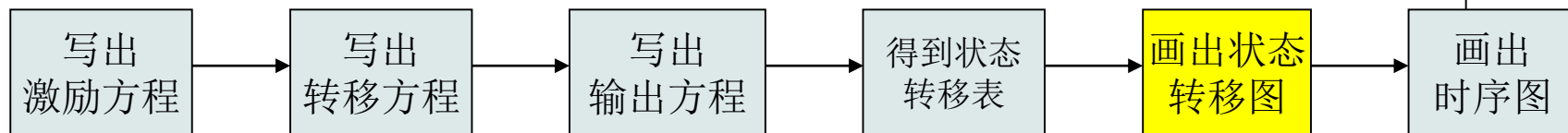
状态表



转移图



## 第五步：画出状态转移图

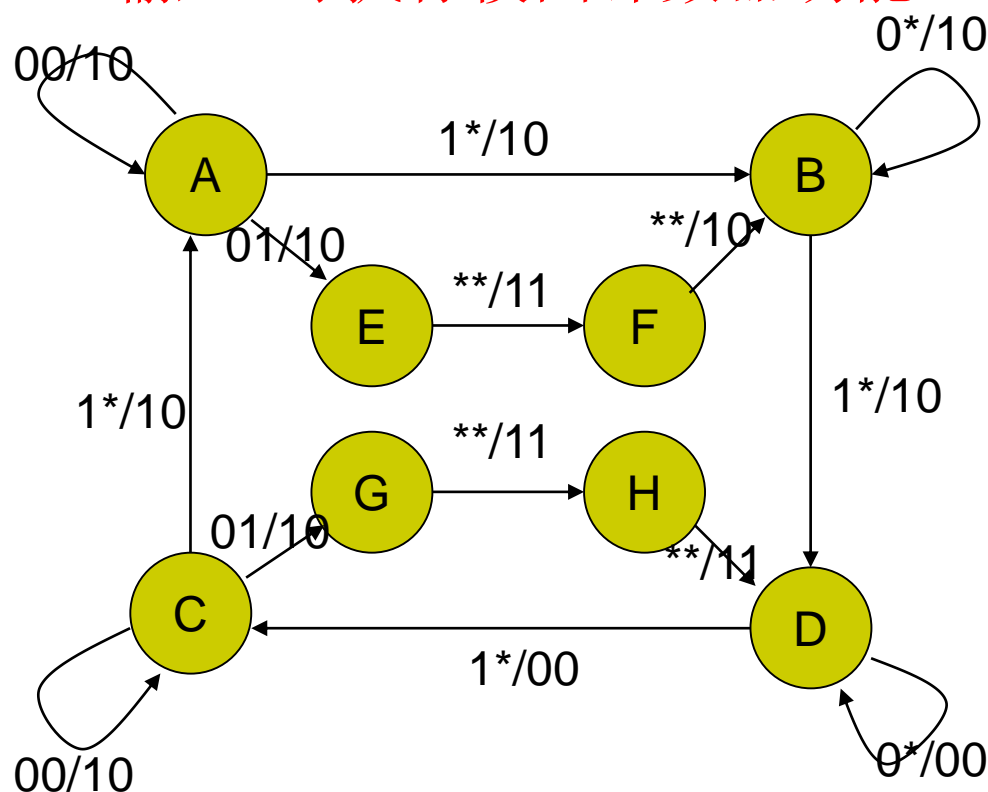


功能：X输入1时执行模四计数器功能。

| S | XY |    |    |    | Z1 Z2 |
|---|----|----|----|----|-------|
|   | 00 | 01 | 10 | 11 |       |
| A | A  | E  | B  | B  | 10    |
| B | B  | B  | D  | D  | 10    |
| C | C  | G  | A  | A  | 10    |
| D | D  | D  | C  | C  | 00    |
| E | F  | F  | F  | F  | 11    |
| F | B  | B  | B  | B  | 10    |
| G | H  | H  | H  | H  | 11    |
| H | D  | D  | D  | D  | 11    |

$S_{n+1}$

状态表



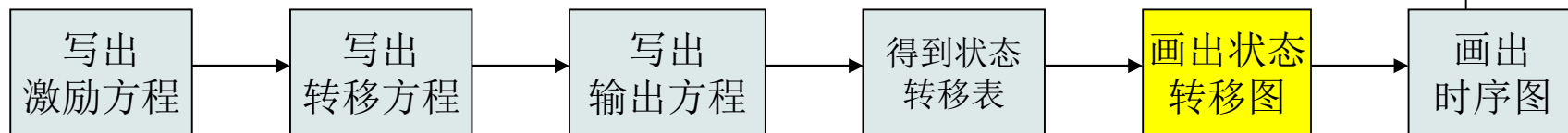
转移图

\*表示与该输入无关

转移表达式必须是互斥的，并且是完备的。



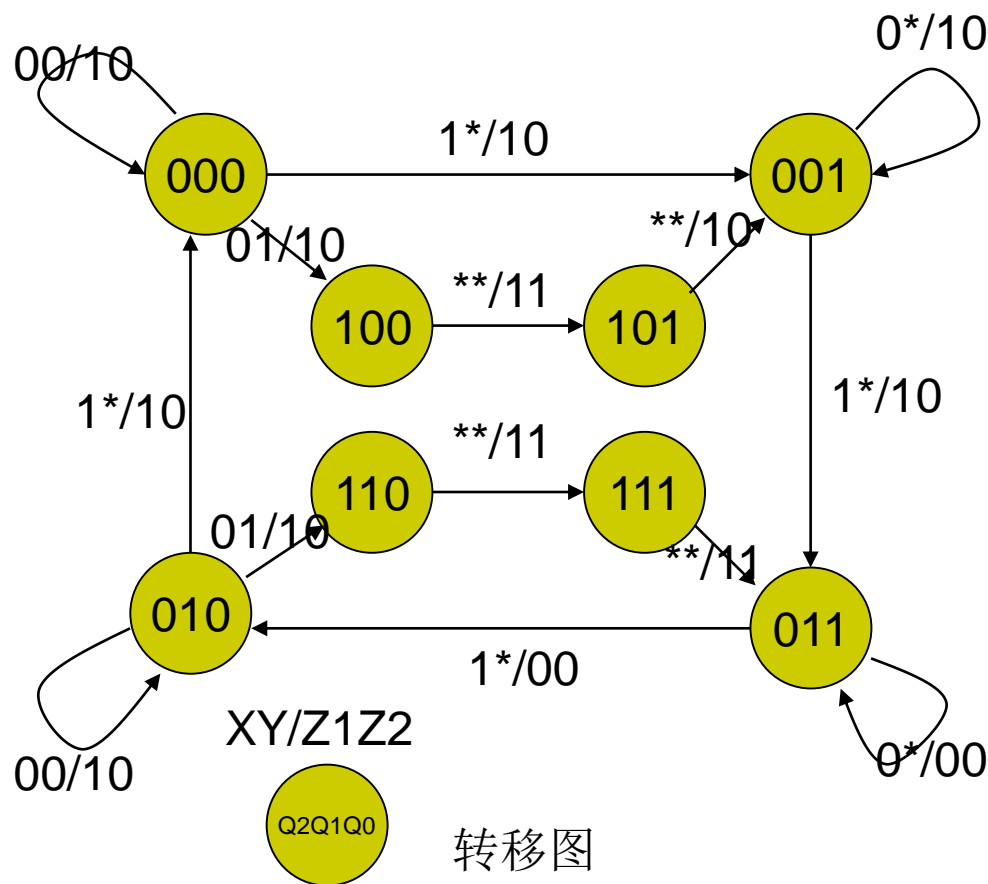
# 第五步：画出状态转移图



| Q <sub>2</sub> Q <sub>1</sub> Q <sub>0</sub> | XY  |     |     |     | Z <sub>1</sub> Z <sub>2</sub> |
|--|-----|-----|-----|-----|-------------------------------|
|  | 00  | 01  | 10  | 11  |                               |
| 000  | 000 | 100 | 001 | 001 | 10                            |
| 001  | 001 | 001 | 011 | 011 | 10                            |
| 010  | 010 | 110 | 000 | 000 | 10                            |
| 011  | 011 | 011 | 010 | 010 | 00                            |
| 100  | 101 | 101 | 101 | 101 | 11                            |
| 101  | 001 | 001 | 001 | 001 | 10                            |
| 110  | 111 | 111 | 111 | 111 | 11                            |
| 111  | 011 | 011 | 011 | 011 | 11                            |

Q<sub>2</sub><sup>n+1</sup>Q<sub>1</sub><sup>n+1</sup>Q<sub>0</sub><sup>n+1</sup>

状态表



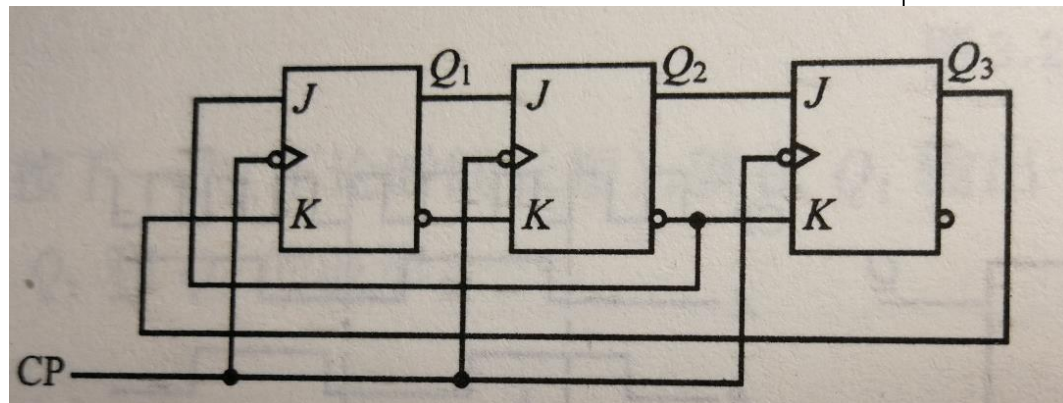
\*表示与该输入无关



# 补充例题1



● 分析如图所示时序电路：



● 激励方程：

- $J1=Q2'$   $K1=Q3$
- $J2=Q1$   $K2=Q1'$
- $J3=Q2$   $K3=Q2'$

● 转移方程：

- $Q1^*=Q2'Q1'+Q3'Q1$
- $Q2^*=Q1Q2'+Q1Q2=Q1$
- $Q3=Q2$

| $Q_3^n$ | $Q_2^n$ | $Q_1^n$ | $Q_3^{n+1}$ | $Q_2^{n+1}$ | $Q_1^{n+1}$ |
|---------|---------|---------|-------------|-------------|-------------|
| 0       | 0       | 0       | 0           | 0           | 1           |
| 0       | 0       | 1       | 0           | 1           | 1           |
| 0       | 1       | 0       | 1           | 0           | 0           |
| 0       | 1       | 1       | 1           | 1           | 1           |
| 1       | 0       | 0       | 0           | 0           | 1           |
| 1       | 0       | 1       | 0           | 1           | 0           |
| 1       | 1       | 0       | 1           | 0           | 0           |
| 1       | 1       | 1       | 1           | 1           | 0           |



# 补充例题1



- 分析如图所示时序电路：

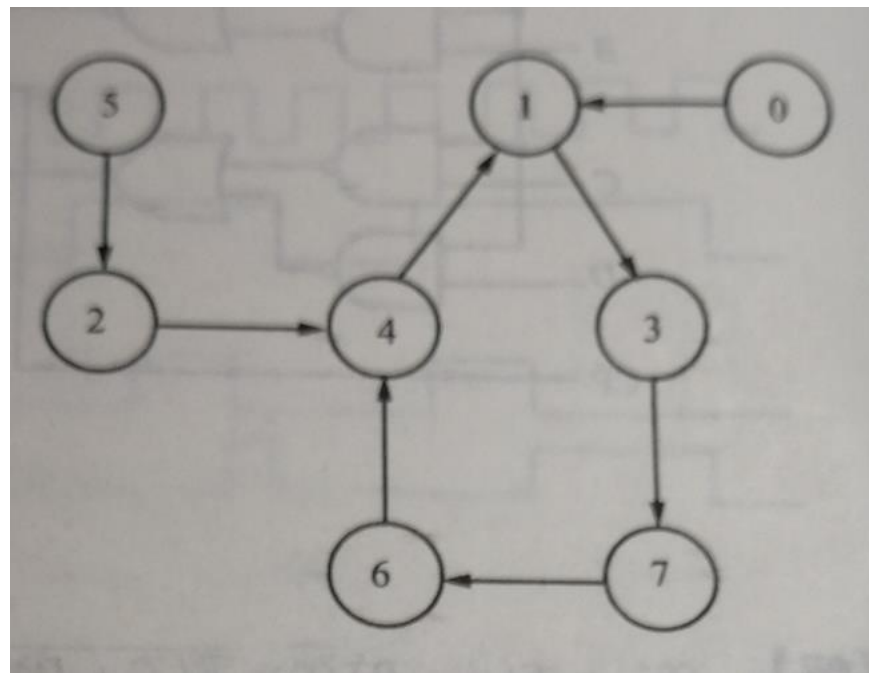
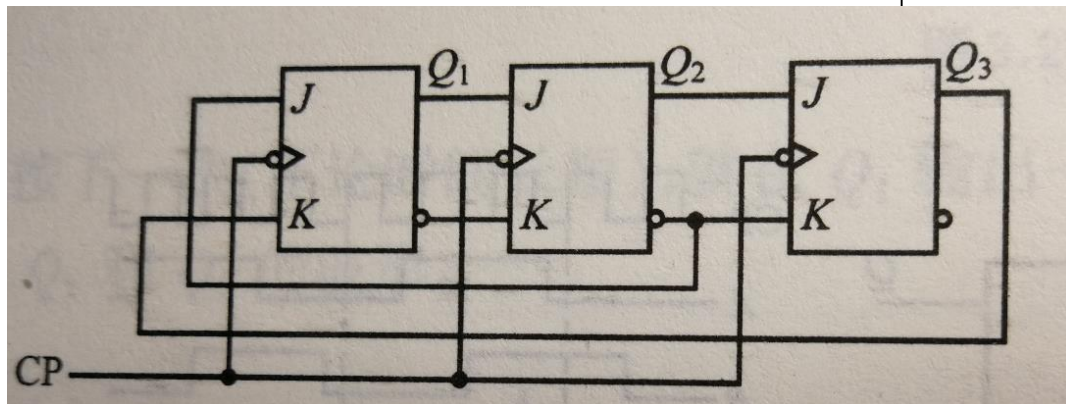
## 模5计数器

- 激励方程：

- $J_1 = Q_2'$   $K_1 = Q_3$
- $J_2 = Q_1$   $K_2 = Q_1'$
- $J_3 = Q_2$   $K_3 = Q_2'$

- 转移方程：

- $Q_1^* = Q_2'Q_1' + Q_3'Q_1$
- $Q_2^* = Q_1Q_2' + Q_1Q_2 = Q_1$
- $Q_3 = Q_2$

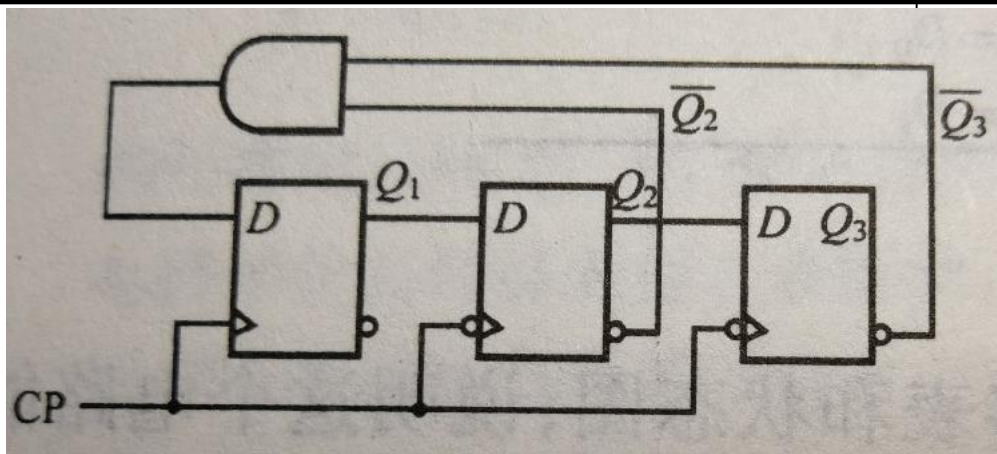




# 补充例题2



● 分析如图所示时序电路：



● 激励方程：

- $D1=Q2'Q3'$
- $D2=Q1$
- $D3=Q2$

● 转移方程：

- $Q1^*=Q2'Q3'$
- $Q2^*=Q1$
- $Q3^*=Q2$

| $Q_3^n$ | $Q_2^n$ | $Q_1^n$ | $D_1$ |
|---------|---------|---------|-------|
| 0       | 0       | 0       | 1     |
| 0       | 0       | 1       | 1     |
| 0       | 1       | 1       | 0     |
| 1       | 1       | 0       | 0     |
| 1       | 0       | 0       | 0     |
| 0       | 0       | 0       | 1     |
| 0       | 1       | 0       | 0     |
| 1       | 0       | 1       | 0     |
| 1       | 1       | 1       | 0     |

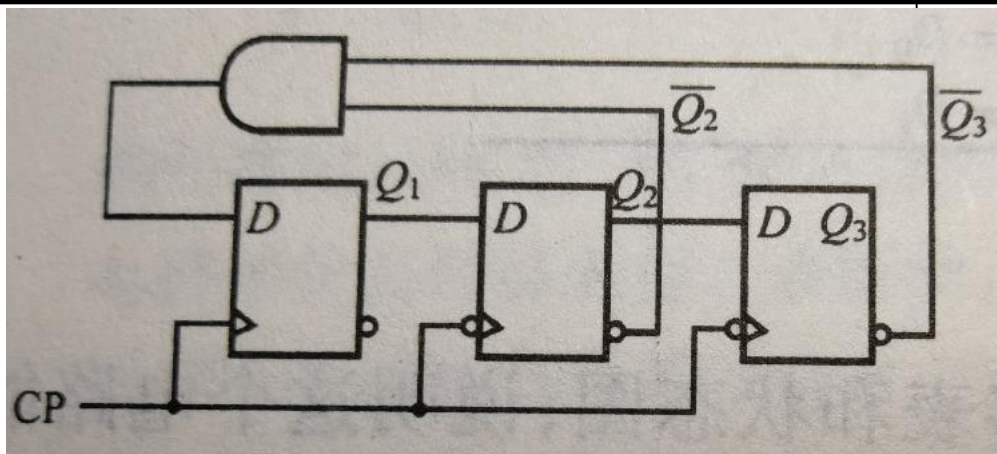
# 补充例题2



- 分析如图所示时序电路：

序列发生器

- 激励方程：
  - $D_1 = Q_2' Q_3'$
  - $D_2 = Q_1$
  - $D_3 = Q_2$
- 转移方程：
  - $Q_1^* = Q_2' Q_3'$
  - $Q_2^* = Q_1$
  - $Q_3^* = Q_2$

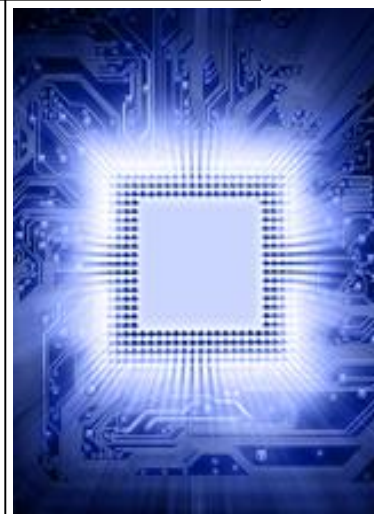


| $Q_3^n$ | $Q_2^n$ | $Q_1^n$ | $D_1$ |
|---------|---------|---------|-------|
| 0       | 0       | 0       | 1     |
| 0       | 0       | 1       | 1     |
| 0       | 1       | 1       | 0     |
| 1       | 1       | 0       | 0     |
| 1       | 0       | 0       | 0     |
| 0       | 0       | 0       | 1     |
| 0       | 1       | 0       | 0     |
| 1       | 0       | 1       | 0     |
| 1       | 1       | 1       | 0     |

# 7.4 时钟同步状态机设计

---

分析过程的逆过程







# 时钟同步状态机设计



1. 根据文字描述，构造状态/输出表
2. （可选）将状态/输出表中的状态数目最小化
3. 状态赋值，选择一组状态变量编码组合赋给状态
4. 将状态变量组合代入状态/输出表，建立转移/输出表。
5. 选择一种触发器作为状态存储器
6. 构造激励表
7. 由激励表推导出激励方程
8. 由转移/输出表推导出输出方程
9. 画出逻辑电路图。



# 设计流程中关键的三步



## 1. 逻辑抽象，原始状态转移图和状态/输出表

- 将每一个状态预设作为电路当前状态，按设计要求和各种可能的输入，确定当前输出和下一状态；注明输入和输出，给出原始状态图，画出状态/输出表。
- 为避免遗漏，可多设状态

## 2. 状态化简：状态数目最小化(State Minimization)

- 状态化简：电路状态数越少，存储电路越简单

## 3. 状态编码(State assignment)

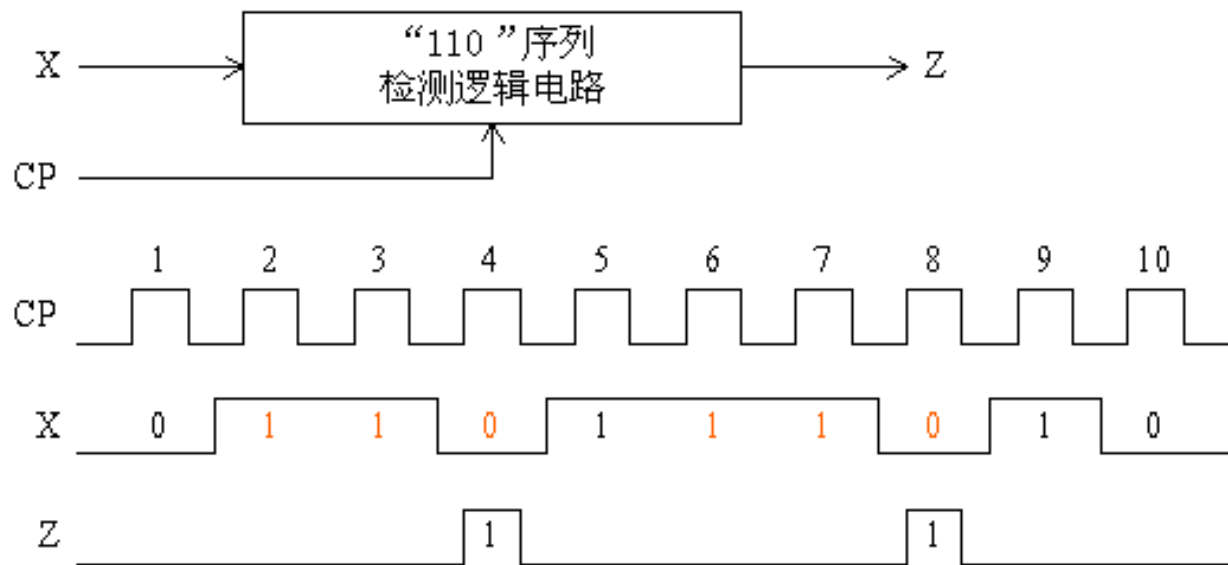
- 状态编码：编码方案选择得当，电路可以简化



# 【附例1】



- 用与非门和D触发器设计一个同步时序逻辑电路，以检测输入的信号序列是否为连续的“110”
- 第一步：确定输入变量和输出函数



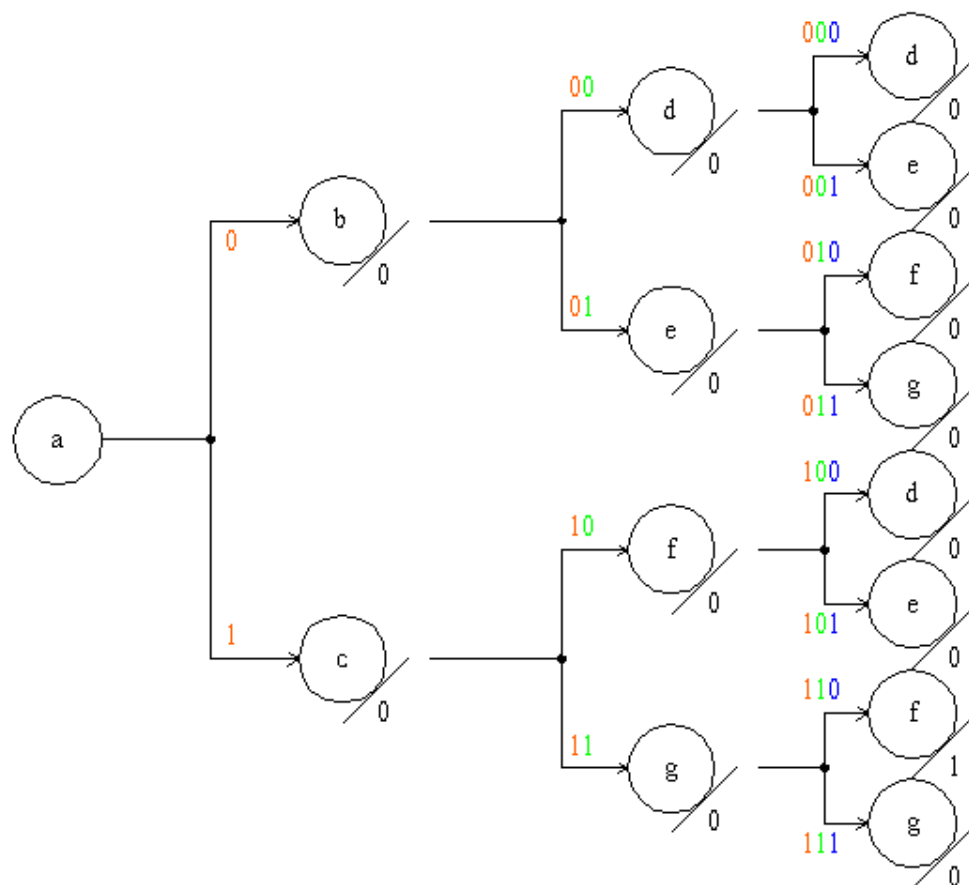


# 【附例1】



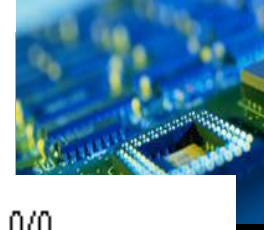
- 第二步：建立原始状态表  
宁多勿缺

| XS | 0   | 1   |
|----|-----|-----|
| a  | b,0 | c,0 |
| b  | d,0 | e,0 |
| c  | f,0 | g,0 |
| d  | d,0 | e,0 |
| e  | f,0 | g,0 |
| f  | d,0 | e,0 |
| g  | f,1 | g,0 |





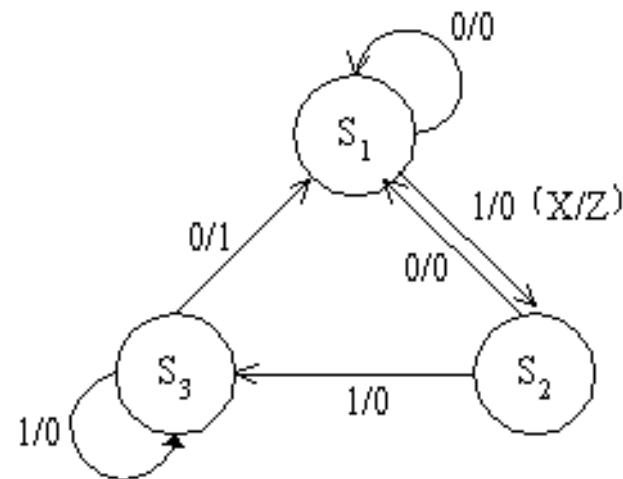
# 【附例1】



- 第三步：建立最简状态表及状态图

$q1=\{b,d,f\}$ ,  $q2 = \{c,e\}$

| XS    | 0       | 1       |
|-------|---------|---------|
| a     | $q_1,0$ | $q_2,0$ |
| $q_1$ | $q_1,0$ | $q_2,0$ |
| $q_2$ | $q_1,0$ | $g,0$   |
| g     | $q_1,1$ | $g,0$   |



| XS    | 0       | 1       |
|-------|---------|---------|
| $S_1$ | $S_1,0$ | $S_2,0$ |
| $S_2$ | $S_1,0$ | $S_3,0$ |
| $S_3$ | $S_1,1$ | $S_3,0$ |



## 【附例1】



- 第四步：状态编码

对状态S1—S3指定二进制代码：采用一对一法：三个状态使用三个触发器，设计简单，多用触发器。

采用计数器法：三个状态使用两个触发器，少用触发器，设计稍复杂。

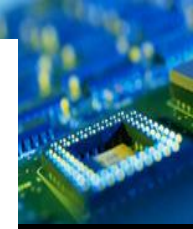
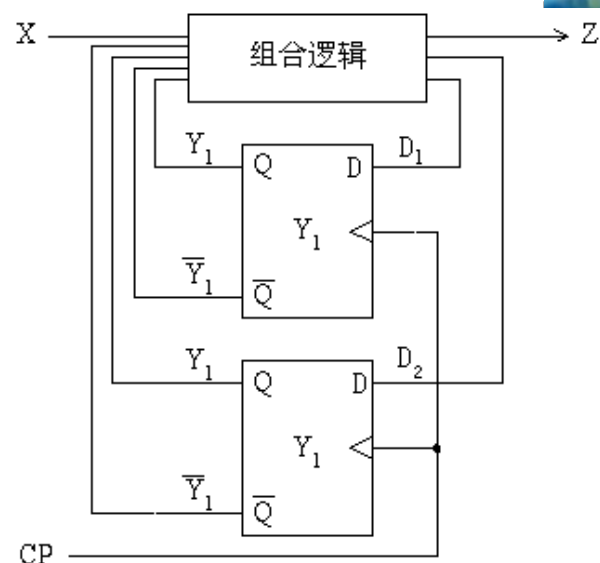
- 现采用计数器法，用两个D触发器y1,y2实现。
- 设：

- $S1 = y2'y1'$                    $S2 = y2y1'$                    $S3 = y2y1$



# 【附例1】

- 确定输出函数与激励函数



| C (条件) | PS (现态)       | NS (次态)               | 激励        | 输出  |
|--------|---------------|-----------------------|-----------|-----|
| $X$    | $y_2^n y_1^n$ | $y_2^{n+1} y_1^{n+1}$ | $D_2 D_1$ | $Z$ |
| 0      | 0 0           | 0 0                   | 0 0       | 0   |
| 0      | 1 0           | 0 0                   | 0 0       | 0   |
| 0      | 1 1           | 0 0                   | 0 0       | 1   |
| 1      | 0 0           | 1 0                   | 1 0       | 0   |
| 1      | 1 0           | 1 1                   | 1 1       | 0   |
| 1      | 1 1           | 1 1                   | 1 1       | 0   |



## 【附例1】



- 确定输出函数与激励函数
- 根据上表，用PS表示现态，NS表示次态，C表示输入条件，我们可得到一个重要的设计公式：

$$\mathbf{NS} = \sum \mathbf{PS} \cdot \mathbf{C}$$

$$y_2^{n+1} = D_2 = \bar{y}_2 \bar{y}_1 x + y_2 \bar{y}_1 x + y_2 y_1 x = (y_2 + \bar{y}_1) x = \overline{\bar{y}_2 y_1} x$$

$$y_1^{n+1} = D_1 = y_2 \bar{y}_1 x + y_2 y_1 x = y_2 x$$

- 输出函数： $Z = y_2 y_1 x'$

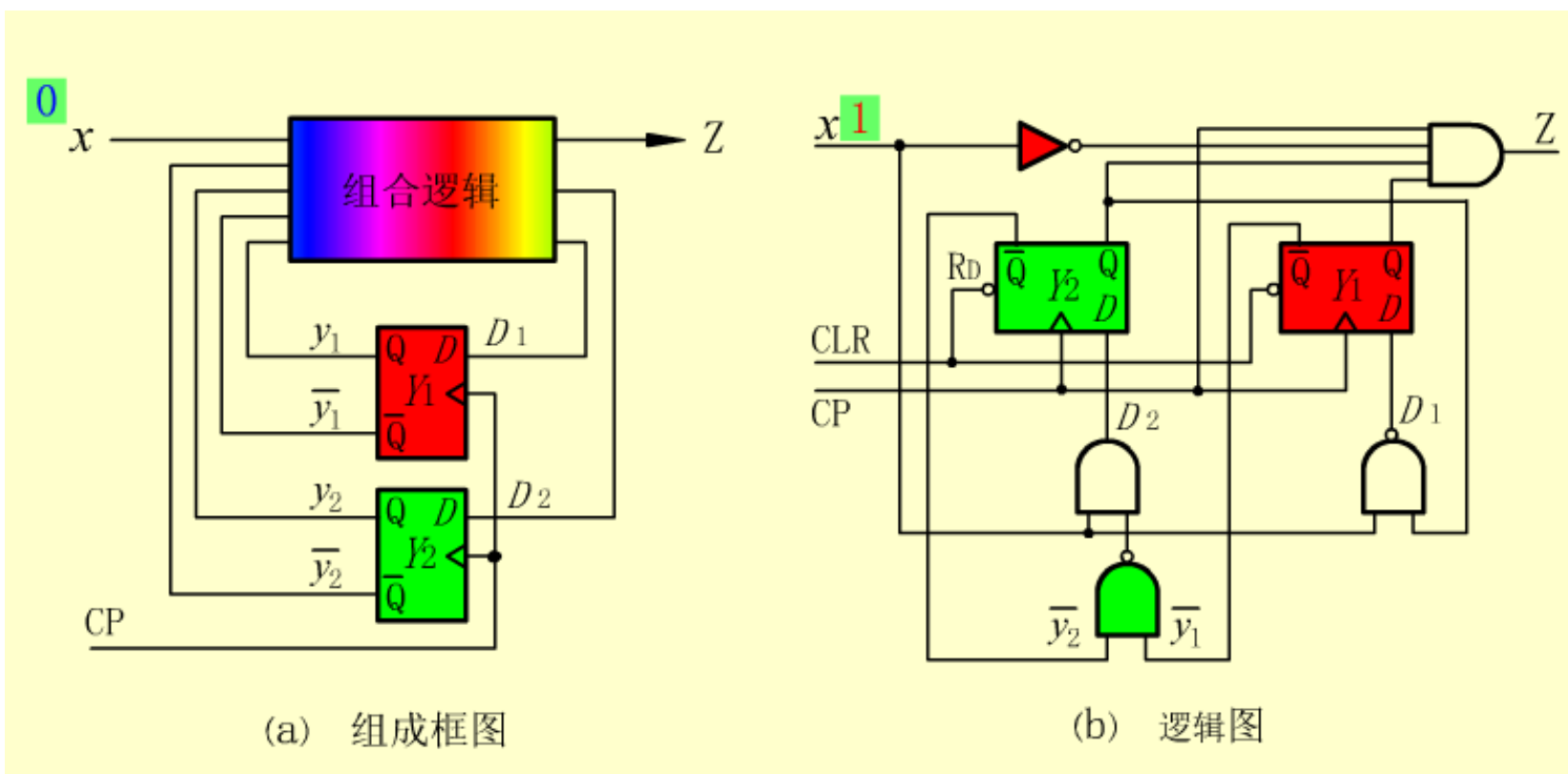




# 【附例1】



- 第五步，画逻辑图，考虑工程问题。

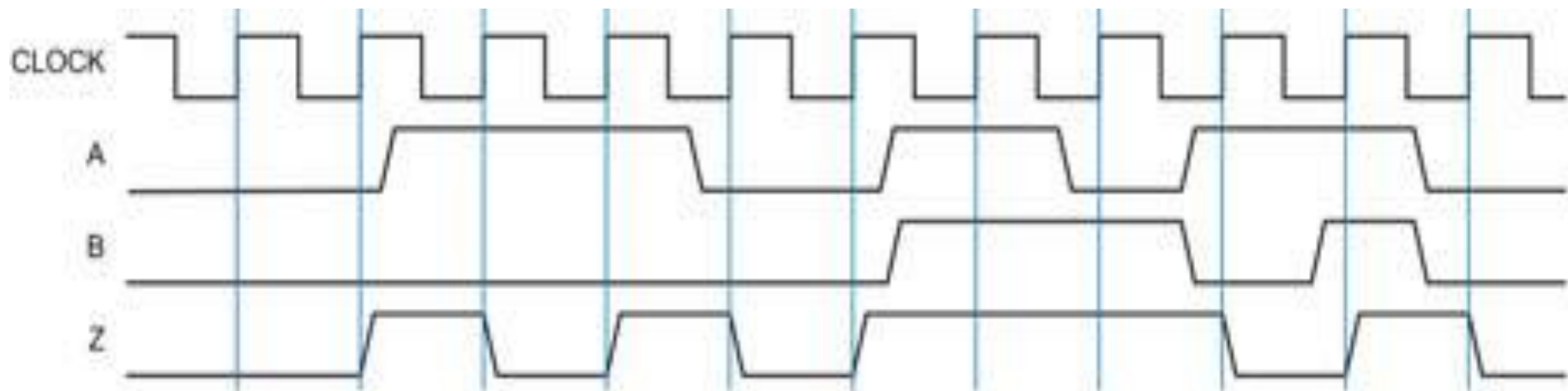




# 状态表设计举例



- 设计一个2输入(A和B), 1输出(Z)同步时序电路, Z为1的条件是
  - 情形1: 前两个时钟采样时刻, A输入值相同;
  - 情形2: 从情形1出现时刻起, B的值一直为1。



moore型电路

采样时刻: 时钟上升沿



# 状态表设计举例



- 一个2输入(A和B), 1输出(Z)同步时序电路, Z为1的条件是
  - 情形1: 前两个时钟采样时刻, A输入值相同;
  - 情形2: 从情形1出现时刻起, B的值一直为1。

| 含义                | State          | Inputs AB=<br>00 01 11 10                                   | Output<br>Z |
|-------------------|----------------|---|-------------|
| 初态                | Init           | S <sub>0</sub> S <sub>0</sub> S <sub>1</sub> S <sub>1</sub> | 0           |
| 时钟采样时刻, A=0       | S <sub>0</sub> | 0k 0k S <sub>1</sub> S <sub>1</sub>                         | 0           |
| 时钟采样时刻, A=1       | S <sub>1</sub> | S <sub>0</sub> S <sub>0</sub> 0k 0k                         | 0           |
| Got 00 or 11 on A | 0k             |   | 1           |



# 状态表设计举例



- 2输入(A和B), 1输出(Z)同步时序电路, Z为1的条件是
  - 情形1: 前两个时钟采样时刻, A输入值相同;
  - 情形2: 从情形1出现时刻起, B的值一直为1。

| 含义             | State  | Inputs AB=<br>00 01 11 10 | Output<br>Z |
|----------------|--------|---------------------------|-------------|
| 初态             | Init   | $S_0 S_0 S_1 S_1$         | 0           |
| 时钟采样时刻, A=0    | $S_0$  | $0k_0 0k_0 S_1 S_1$       | 0           |
| 时钟采样时刻, A=1    | $S_1$  | $S_0 S_0 0k_1 0k_1$       | 0           |
| 两个A输入相等, 上一A=0 | $0k_0$ | <b>AB=01→A=0</b>          | 1           |
| 两个A输入相等, 上一A=1 | $0k_1$ | <b>AB=11→A=1</b>          | 1           |
| 两个A输入相等, 上一A=0 | $0k_0$ | $0k_0 0k_0 0k_1 S_1$      | 1           |
| 两个A输入相等, 上一A=1 | $0k_1$ | $S_0 0k_0 0k_1 0k_1$      | 1           |



# 最简状态表(最小化状态表)



- 2输入(A和B), 1输出(Z)同步时序电路, Z为1的条件是
  - 情形1: 前两个时钟采样时刻, A输入值相同;
  - 情形2: 从情形1出现时刻起, B的值一直为1。

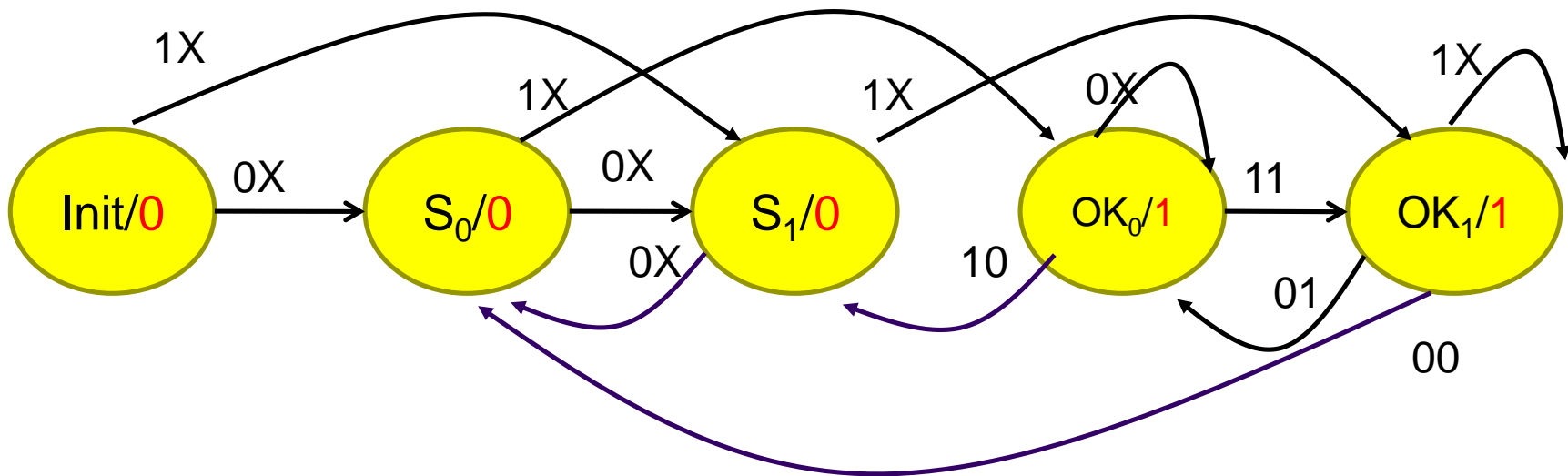
| 含 义            | State  | Inputs AB=<br>00 01 11 10 | Output<br>Z |
|----------------|--------|---------------------------|-------------|
| 初态             | Init   | $S_0 S_0 S_1 S_1$         | 0           |
| 时钟采样时刻, A=0    | $S_0$  | $0k_0 0k_0 S_1 S_1$       | 0           |
| 时钟采样时刻, A=1    | $S_1$  | $S_0 S_0 0k_1 0k_1$       | 0           |
| 两个A输入相等, 上一A=0 | $0k_0$ | $0k_0 0k_0 0k_1 S_1$      | 1           |
| 两个A输入相等, 上一A=1 | $0k_1$ | $S_0 0k_0 0k_1 0k_1$      | 1           |



# 最简状态



| 含 义            | State     | Inputs AB=<br>00 01 11 10 | Output<br>Z |
|----------------|-----------|---------------------------|-------------|
| 初态             | Init      | $S_0 S_0 S_1 S_1$         | 0           |
| 时钟采样时刻, A=0    | $S_0$     | $0k_0 0k_0 S_1 S_1$       | 0           |
| 时钟采样时刻, A=1    | $S_1$     | $S_0 S_0 0k_1 0k_1$       | 0           |
| 两个A输入相等, 上一A=0 | $0k_0$    | $0k_0 0k_0 0k_1 S_1$      | 1           |
| 两个A输入相等, 上一A=1 | $0k_{11}$ | $S_0 0k_0 0k_1 0k_1$      | 1           |





# 非最小化状态表示例



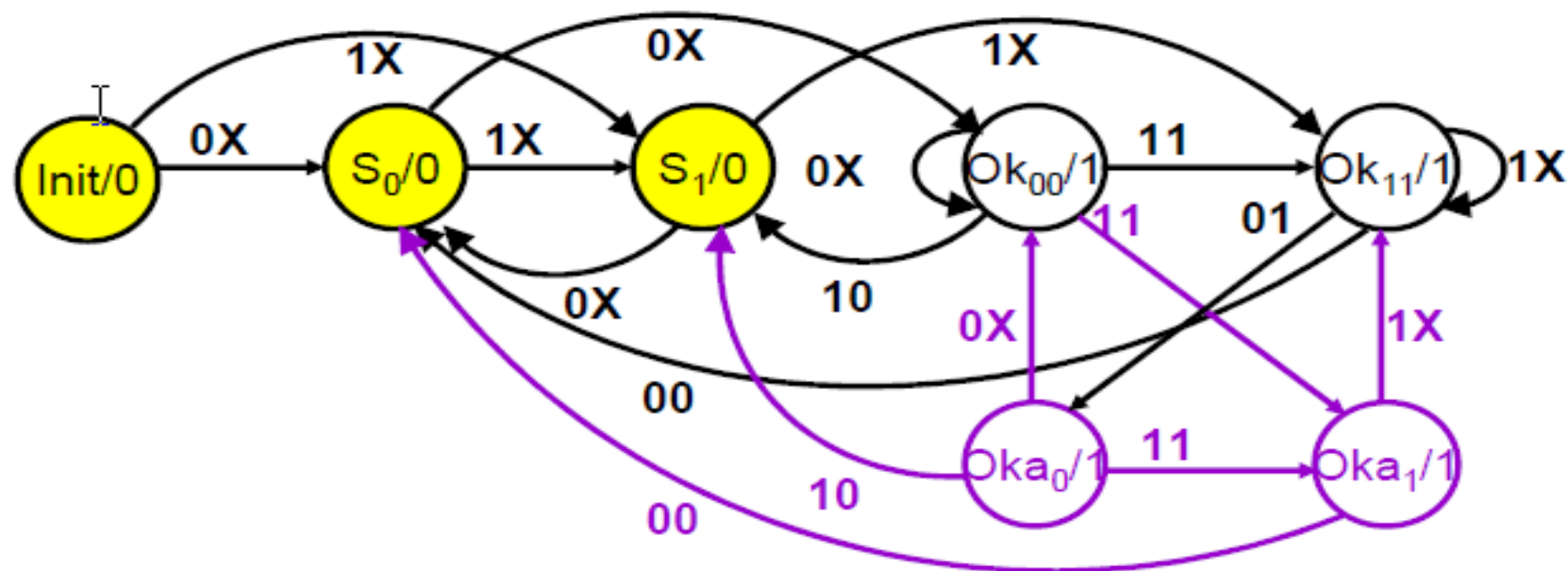
- 情形1：前两个时钟采样时刻，A输入值相同；
- 情形2：从情形1出现时刻起，B的值一直为1。

| 含义               | State            | Inputs AB=<br>00 01 11 10                                | Output<br>Z |
|------------------|------------------|--|-------------|
| 初态               | Init             | $S_0$ $S_0$ $S_1$ $S_1$                                  | 0           |
| 时钟采样时刻, A=0      | $S_0$            | Ok <sub>00</sub> Ok <sub>00</sub> $S_1$ $S_1$            | 0           |
| 时钟采样时刻, A=1      | $S_1$            | $S_0$ $S_0$ Ok <sub>11</sub> Ok <sub>11</sub>            | 0           |
| Got 00 on A      | Ok <sub>00</sub> | Ok <sub>00</sub> Ok <sub>00</sub> Oka <sub>1</sub> $S_1$ | 1           |
| Got 11 on A      | Ok <sub>11</sub> | $S_0$ Oka <sub>0</sub> Ok <sub>11</sub> Ok <sub>11</sub> | 1           |
| Ok, got a 0 on A | Oka <sub>0</sub> | Ok <sub>00</sub> Ok <sub>00</sub> Oka <sub>1</sub> $S_1$ | 1           |
| Ok, got a 1 on A | Oka <sub>1</sub> | $S_0$ Oka <sub>0</sub> Ok <sub>11</sub> Ok <sub>11</sub> | 1           |

# 非最小化

- 状态最小化
  - 识别等效状态

| 含义               | State     | Inputs AB=<br>00 01 11 10           | Output<br>Z |
|------------------|-----------|-------------------------------------|-------------|
| 初态               | Init      | $S_0$ $S_0$ $S_1$ $S_1$             | 0           |
| 时钟采样时刻, A=0      | $S_0$     | $Ok_{00}$ $Ok_{00}$ $S_1$ $S_1$     | 0           |
| 时钟采样时刻, A=1      | $S_1$     | $S_0$ $S_0$ $Ok_{11}$ $Ok_{11}$     | 0           |
| Got 00 on A      | $Ok_{00}$ | $Ok_{00}$ $Ok_{00}$ $Ok_{a1}$ $S_1$ | 1           |
| Got 11 on A      | $Ok_{11}$ | $S_0$ $Ok_{a0}$ $Ok_{11}$ $Ok_{11}$ | 1           |
| Ok, got a 0 on A | $Ok_{a0}$ | $Ok_{00}$ $Ok_{00}$ $Ok_{a1}$ $S_1$ | 1           |
| Ok, got a 1 on A | $Ok_{a1}$ | $S_0$ $Ok_{a0}$ $Ok_{11}$ $Ok_{11}$ | 1           |



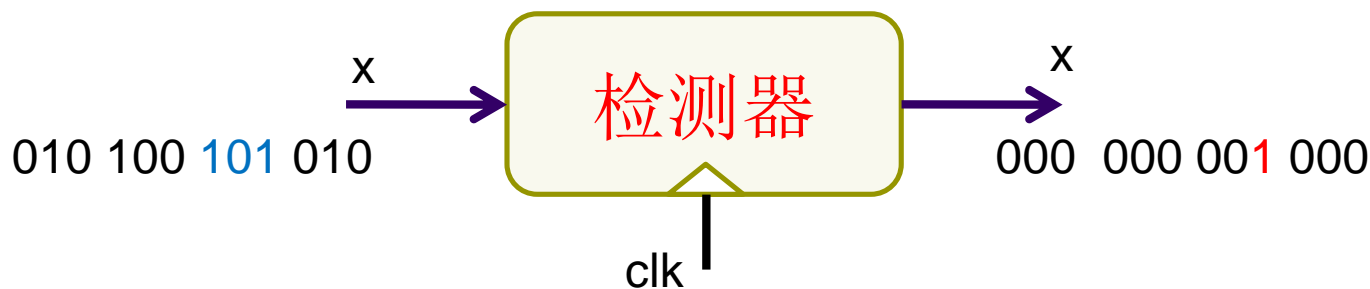




# 直接构图法设计时序逻辑电路



- 举例：设计判断输入序列为101的检测器。输入为X，输出为Z。画出状态图
  - 对输入序列**每3位**进行一次判决；若3位代码是101，则对应其最后一个1时，输出Z为1；其它情况Z为0
  - 例如X = 010 100 101 010  
Z = 000 000 00**1** 000

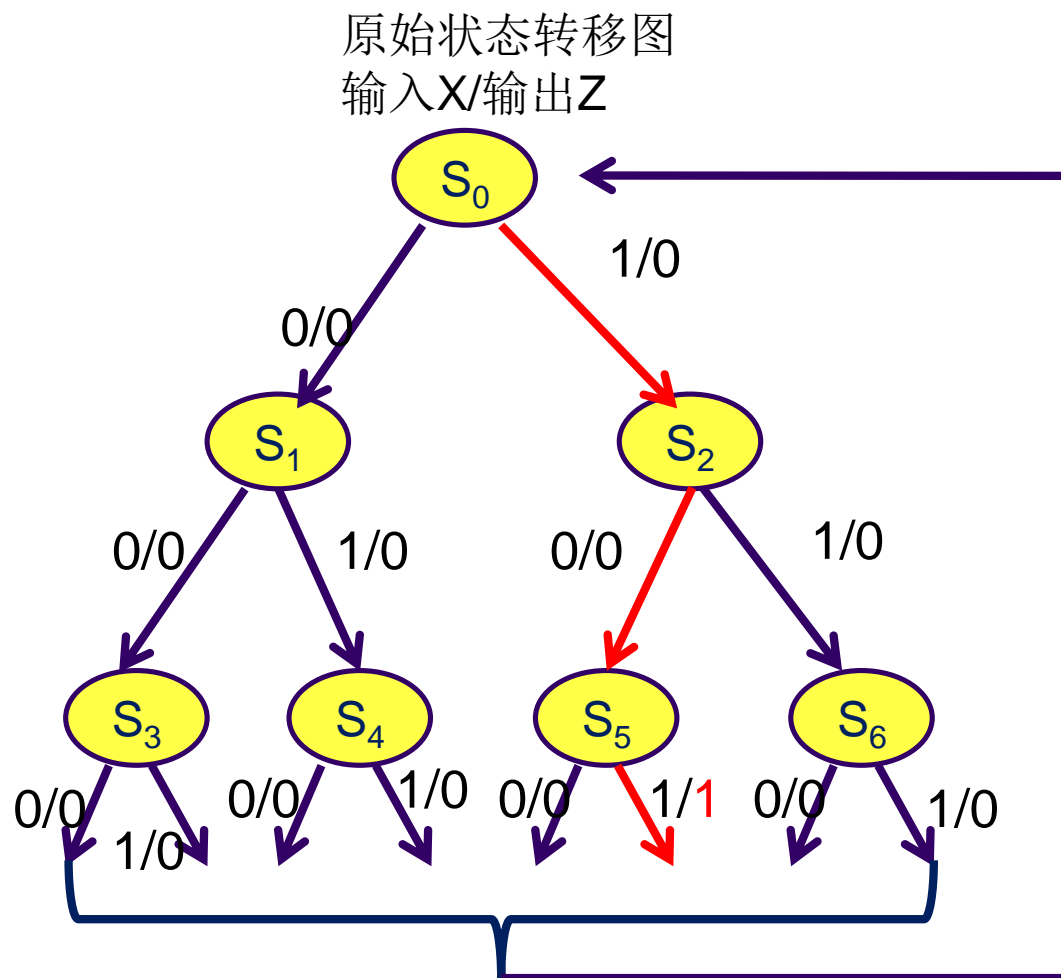




# 建立原始状态图(表)



- 将文字描述转化成的原始状态/输出表：相当于数学建模。
- 直接构图法：
  - 先设定一个初态，从初态开始，每加入一个输入，就得到一个次态；
  - 该次态可能就是现态本身，也可能是已有的另一个状态，或是新增加的一个状态。
  - 此过程一直持续下去，直到每一个现态向其次态的转换都被考虑，并且不再构成新的状态。
  - 最后确定需要多少个状态。

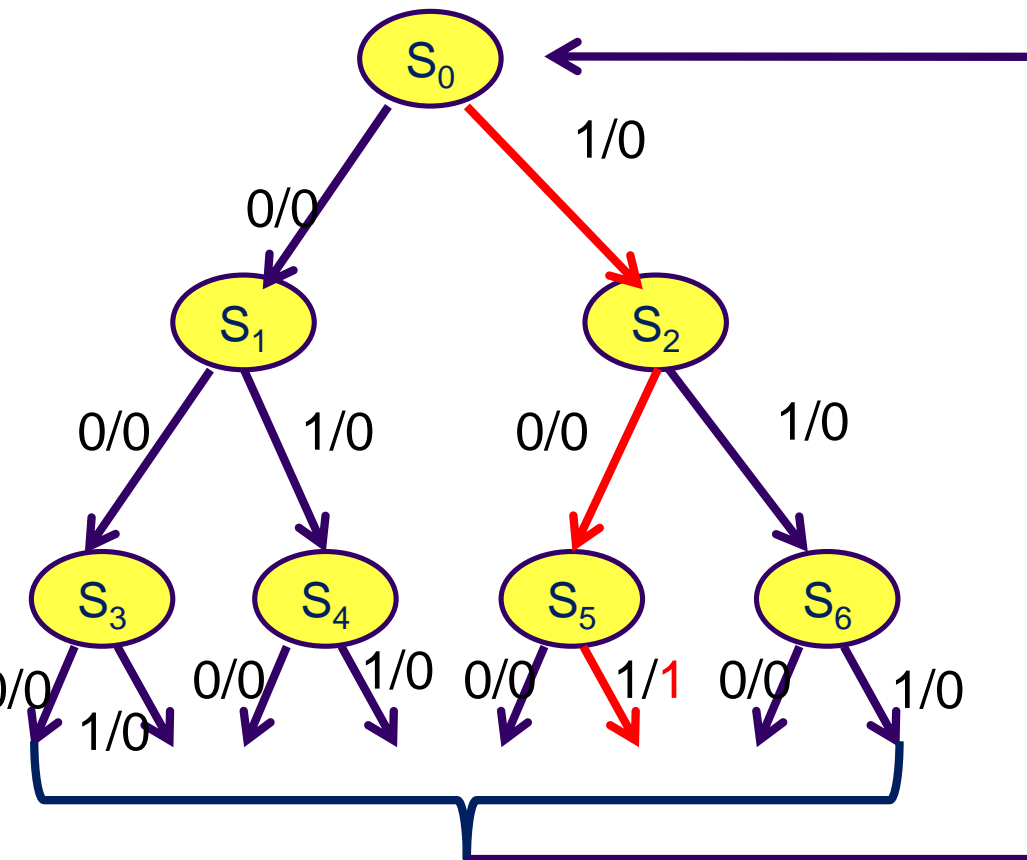




# 原始状态/输出表



原始状态转移图  
输入X/输出Z



原始状态/输出表

| 当前状态 | 次态  |     | 输出  |     |
|------|-----|-----|-----|-----|
|      | X=0 | x=1 | X=0 | X=1 |
| s0   | s1  | s2  | 0   | 0   |
| s1   | s3  | s4  | 0   | 0   |
| s2   | s5  | s6  | 0   | 0   |
| s3   | s0  | s0  | 0   | 0   |
| s4   | s0  | s0  | 0   | 0   |
| s5   | s0  | s0  | 0   | 1   |
| s6   | s0  | s0  | 0   | 0   |



- 思考：

- 上述问题是三个一组测试，如果组间可以重叠，这样的序列检测器设计，其状态图会是如何？



## 7.4.2、状态化简



一般情况下，原始状态图和原始状态表中存在着多余的状态。状态个数越多，电路中所需的触发器的数目也越多，制造成本就越高。为降低制造成本，需要去掉多余的状态，即要进行**状态简化**。

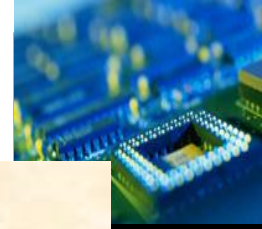
所谓**状态简化**，就是要获得一个最小化的状态表。这个表不仅能正确地反映设计的全部要求，而且状态的数目最少。



# 一、完全确定状态表的简化



**等价状态：** 设状态 $S_1$ 和 $S_2$ 是完全确定状态表中的两个状态,如果对于所有可能的输入序列，分别从状态 $S_1$  和状态 $S_2$ 出发，所得到的**输出响应序列完全相同**， 则状态 $S_1$ 和 $S_2$ 是等价的，记作 $(S_1, S_2)$  .或说，状态 $S_1$ 和 $S_2$ 是等价对。等价状态可以合并。



- 等价状态传递性:  $(S_1, S_2), (S_2, S_3) \rightarrow (S_1, S_3)$
- 等价类: 彼此等价的状态集合
- 最大等价类: 不被其它等价类所包含的等价类。  
一个状态也可能是一个最大等价类。

状态简化的任务是要在原始状态表中找出  
**全部最大等价类**(最大等价类集合), 并将每一个  
最大等价类用一个状态来表示。



## 判别方法:



假定状态 $S_1$ 和 $S_2$ 是完全确定原始状态表中的两个现态，那么 $S_1$ 和 $S_2$ 等价的条件可归纳为在输入的各种取值组合下：

第一、它们的输出完全相同；

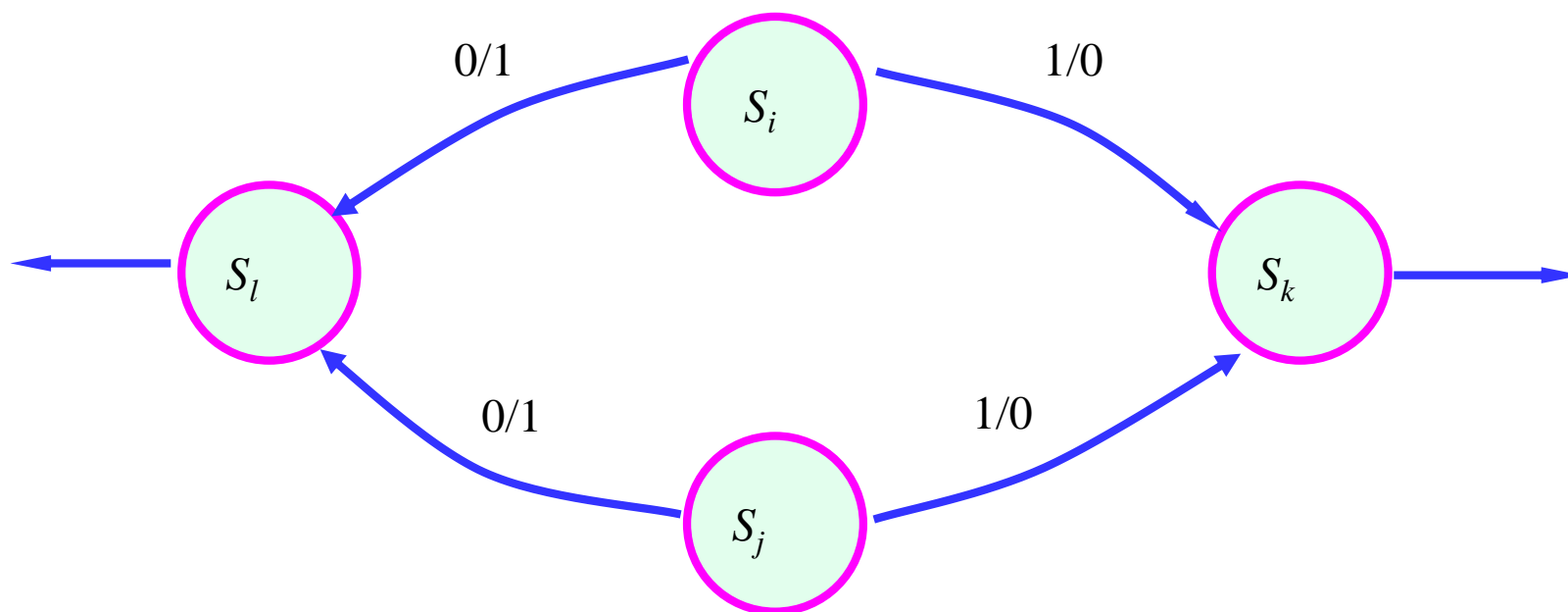
第二、它们的次态满足下列条件之一，即

- (1) 次态相同；
- (2) 次态交错；
- (3) 次态循环；
- (4) 次态对等价。



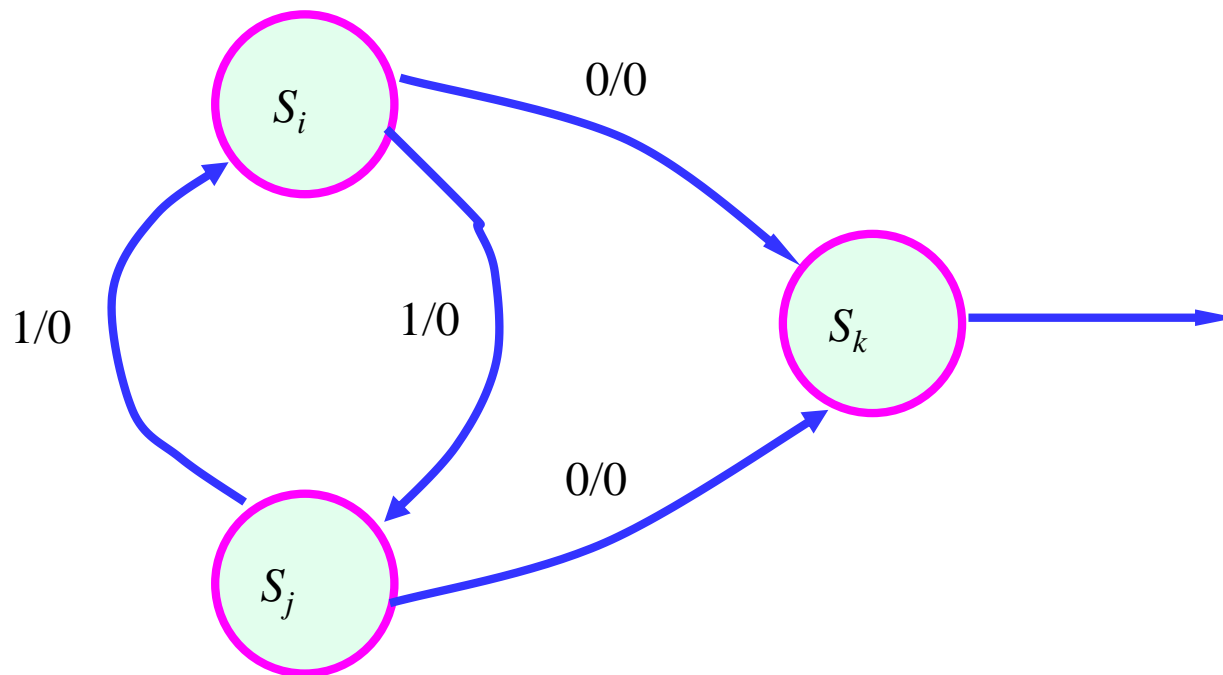


# 次态相同



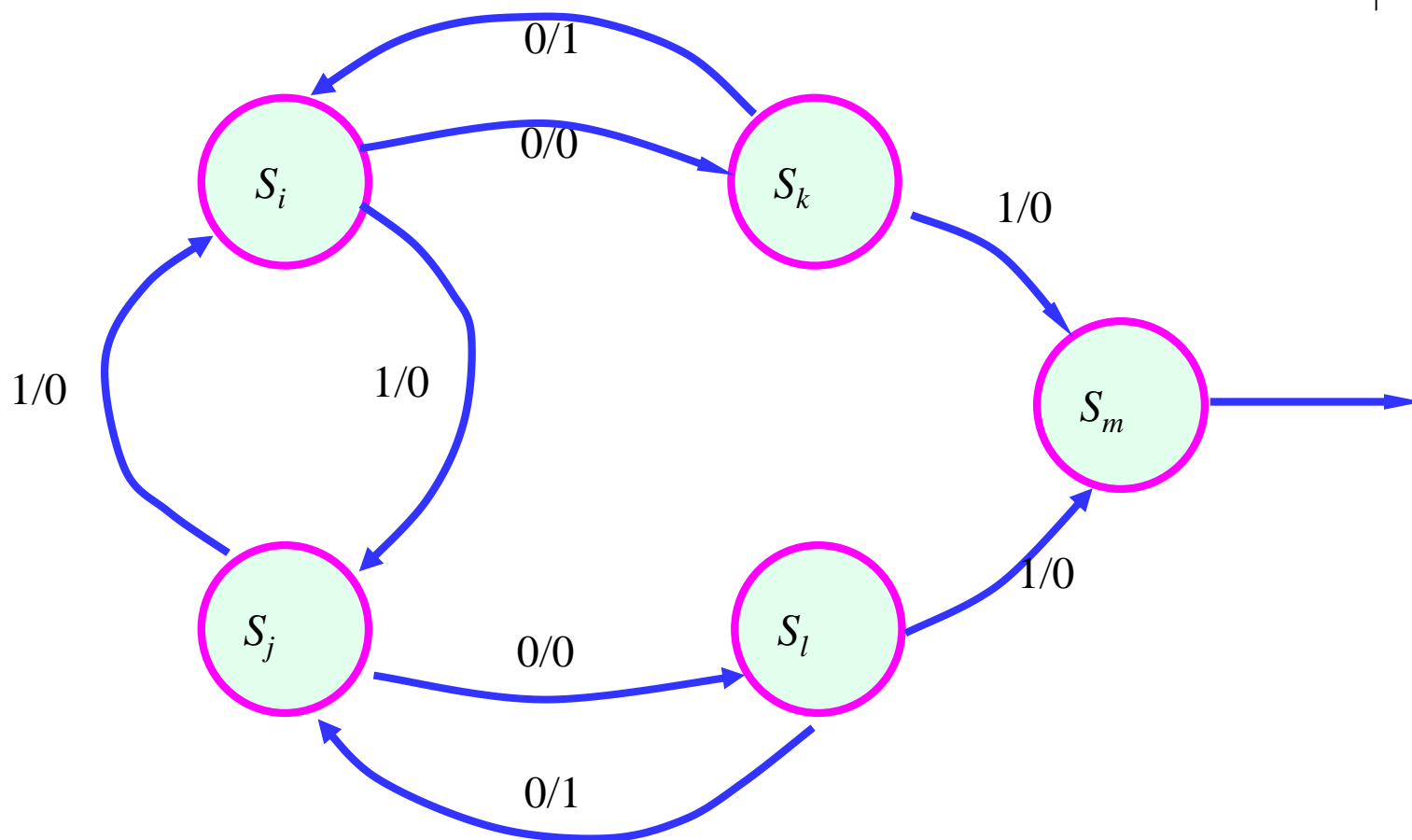


# 次态相同或交错



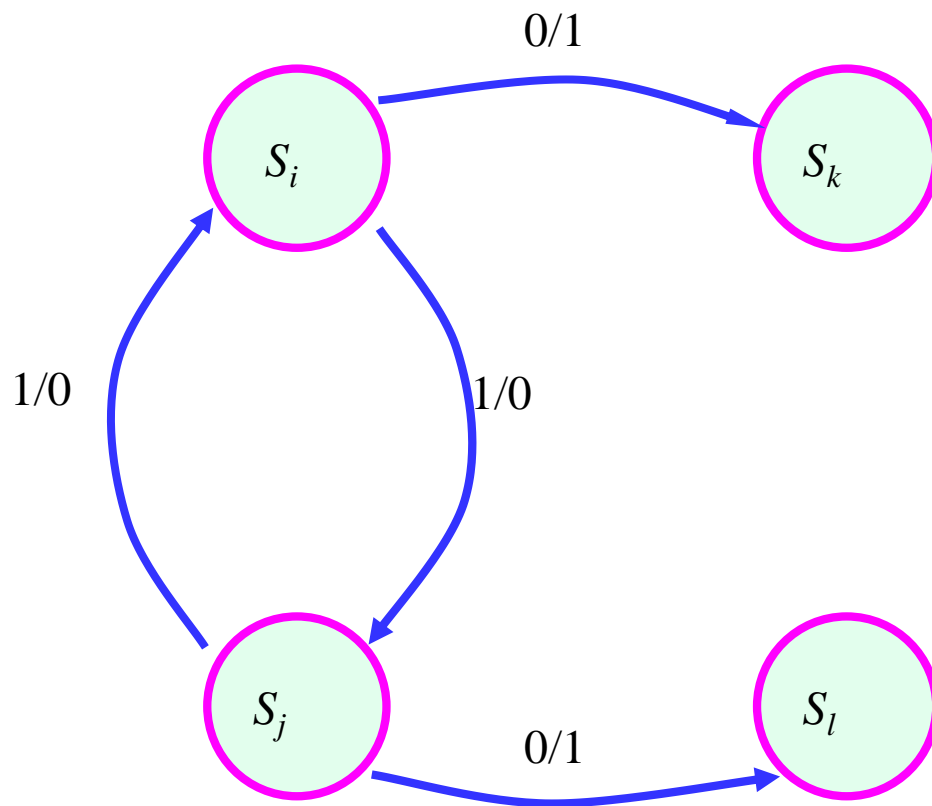


# 次态交错或相同或循环





# 次态交错或等价( $S_k, S_l$ 等价)





# 1. 观察法化简

例：简化下表所示的状态表

| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | A/0   | B/0   |
| B  | A/0   | C/0   |
| C  | A/0   | D/1   |
| D  | A/0   | D/1   |



# 解：



| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | A/0   | B/0   |
| B  | A/0   | C/0   |
| C  | A/0   | D/1   |
| D  | A/0   | D/1   |

- A和B，C和D的输出完全相等；
- C和D在输入的各种取值组合下，次态相同，因此C和D等价；
- A和B在 $x=1$ 时的次态不满足四条件之一，因此A和B不等价；
- 最大等价类为{A}，{B}，{C,D}，分别用A', B', C'表示；



最小化状态表为：

| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | A/0   | B/0   |
| B  | A/0   | C/0   |
| C  | A/0   | D/1   |
| D  | A/0   | D/1   |

| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A' | A'/0  | B'/0  |
| B' | A'/0  | C'/0  |
| C' | A'/0  | C'/1  |

这个时序电路执行的是什么功能？



## 1. 隐含表法化简



例：简化下表所示的状态表

| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | C/0   | B/1   |
| B  | F/0   | A/1   |
| C  | D/0   | G/0   |
| D  | D/1   | E/0   |
| E  | C/0   | E/1   |
| F  | D/0   | G/0   |
| G  | C/1   | D/0   |



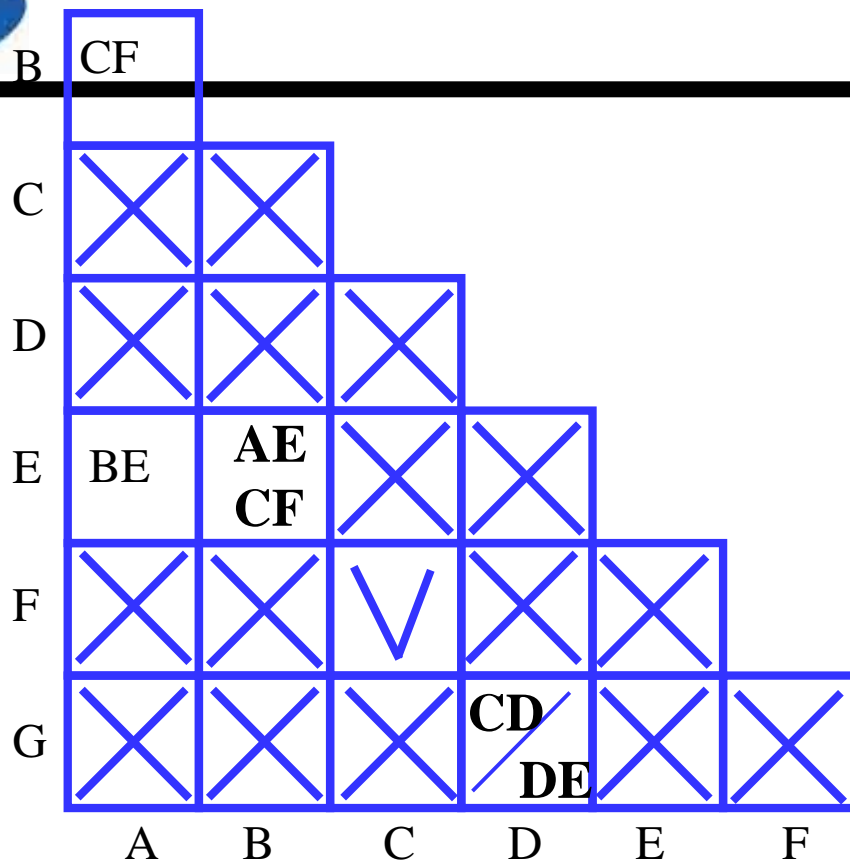


# 解：• 作隐含表

- 顺序比较，寻找等价状态对
- 状态对等价，打“√”；
- 状态对不等价，打“×”；
- 状态对是否等价需进一步检查，则标记次态对。

| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | C/0   | B/1   |
| B  | F/0   | A/1   |
| C  | D/0   | G/0   |
| D  | D/1   | E/0   |
| E  | C/0   | E/1   |
| F  | D/0   | G/0   |
| G  | C/1   | D/0   |

|   |    |          |   |          |   |   |
|---|----|----------|---|----------|---|---|
| B | CF |          |   |          |   |   |
| C | ×  | ×        |   |          |   |   |
| D | ×  | ×        | × |          |   |   |
| E | BE | AE<br>CF | × | ×        |   |   |
| F | ×  | ×        | √ | ×        | × |   |
| G | ×  | ×        | × | CD<br>DE | × | × |



- 关联比较，  
确定等价状态对

$$AB \rightarrow CF \checkmark$$

$$AE \rightarrow BE \rightarrow CF \checkmark$$

处于循环链中的每一个状态对都是等价状态对，一共四个等价对 (A,B), (A,E), (B,E), (C,F)。



## 确定最大等价类，作最小化状态表：



- 四个等价对 (A,B), (A,E), (B,E), (C,F)
- 四个最大等价类 (A,B,E), (C,F), (D), (G)
- 令以上四个最大等价类依次为a, b, c, d.

| 现态 | 次态/输出 |     |
|----|-------|-----|
|    | x=0   | x=1 |
| A  | C/0   | B/1 |
| B  | F/0   | A/1 |
| C  | D/0   | G/0 |
| D  | D/1   | E/0 |
| E  | C/0   | E/1 |
| F  | D/0   | G/0 |
| G  | C/1   | D/0 |

| 现态 | 次态/输出 |     |
|----|-------|-----|
|    | x=0   | x=1 |
| a  | b/0   | a/1 |
| b  | c/0   | d/0 |
| c  | c/1   | a/0 |
| d  | b/1   | c/0 |



## 二、不完全确定状态表的简化

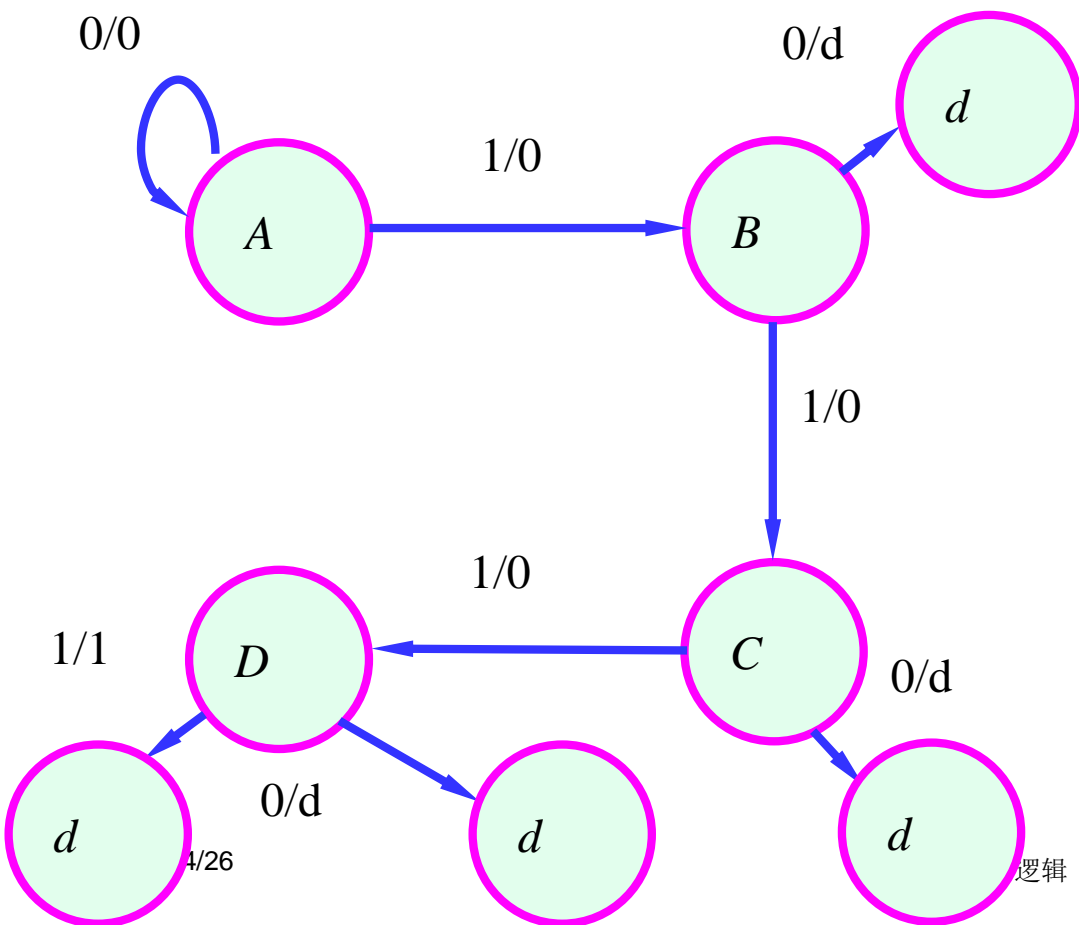


**不完全确定状态表：** 状态表中存在不确定的次态或输出，这些不确定的次态或输出将有利于状态简化。

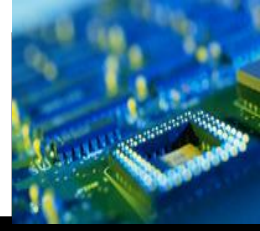
- **相容状态：** 设状态 $S_1$ 和 $S_2$ 是不完全确定状态表中的两个状态，如果对于所有的有效输入序列，分别从状态 $S_1$ 和 $S_2$ 出发，所得到的输出响应序列(除不确定的那些位之外)是完全相同的，那么状态 $S_1$ 和 $S_2$ 是相容的，或者说状态 $S_1$ 和 $S_2$ 是相容对，记作 $(S_1, S_2)$ 。相容状态可以合并。



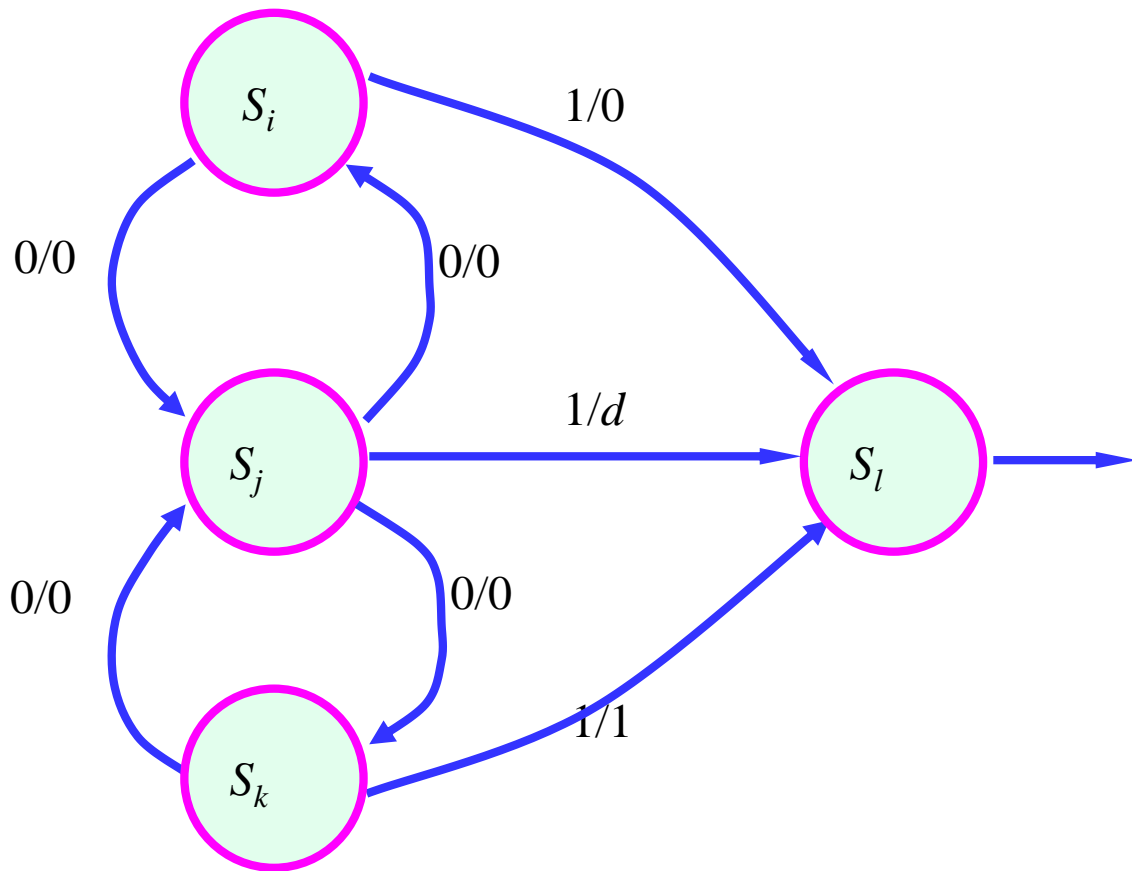
设计一个“1111”序列检测器，使其成为爆炸装置的引爆控制器。假定工作条件为：平时无1输入，Z一直处于0状态；当连续输入4个1时（不允许出现0），Z=1引爆，整个装置不存在。



| 现态 | 次态/输出 |     |
|----|-------|-----|
|    | x=0   | x=1 |
| A  | A/0   | B/0 |
| B  | d/d   | C/0 |
| C  | d/d   | D/0 |
| D  | d/d   | d/1 |



- 相容状态无传递性：

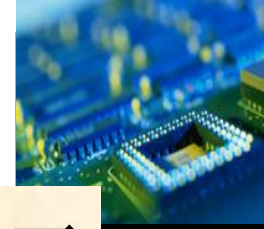


$S_i$ 和 $S_j$ 相容；  
 $S_j$ 和 $S_k$ 相容；  
但 $S_i$ 和 $S_k$ 不相容。

- 相容类：彼此相容的状态集合
- 最大相容类：不被其它相容类所包含的相容类



## 判别方法:



在不完全确定状态表中判断两个状态是否相容也是根据表中给出的次态和输出来决定的。假定状态 $S_i$ 和 $S_j$ 是不完全确定状态表中的两个现态，那么状态 $S_i$ 和 $S_j$ 相容的条件可归纳为在输入的各种取值组合下：

第一、它们的输出完全相同，或者其中的一个(或两个)输出为任意值。

第二、它们的次态满足下列条件之一：

- (1) 次态相同；
- (2) 次态交错；
- (3) 次态循环；
- (4) 其中的一个(或两个)为任意状态；
- (5) 次态相容；



## 例：简化下表所示的状态表



| 现 态 | 次 态   |       | 输 出 |
|-----|-------|-------|-----|
|     | $x=0$ | $x=1$ |     |
| A   | B     | D     | 0   |
| B   | B     | D     | d   |
| C   | A     | E     | 1   |
| D   | d     | E     | 1   |
| E   | F     | d     | 1   |
| F   | d     | C     | d   |





解:

• 作隐含表:

• 顺序比较,  
寻找相容对;



| 现 态 | 次 态   |       | 输 出 |
|-----|-------|-------|-----|
|     | $x=0$ | $x=1$ |     |
| A   | B     | D     | 0   |
| B   | B     | D     | d   |
| C   | A     | E     | 1   |
| D   | d     | E     | 1   |
| E   | F     | d     | 1   |
| F   | d     | C     | d   |

|   |    |          |    |    |   |
|---|----|----------|----|----|---|
| B | ✓  |          |    |    |   |
| C | ✗  | AB<br>DE |    |    |   |
| D | ✗  | DE       | ✓  |    |   |
| E | ✗  | BF       | AF | ✓  |   |
| F | CD | CD       | CE | CE | ✓ |
|   | A  | B        | C  | D  | E |



|   |    |          |    |    |   |
|---|----|----------|----|----|---|
| B | ✓  |          |    |    |   |
| C | ✗  | AB<br>DE |    |    |   |
| D | ✗  | DE       | ✓  |    |   |
| E | ✗  | BF       | AF | ✓  |   |
| F | CD | CD       | CE | CE | ✓ |
|   | A  | B        | C  | D  | E |

$$AF \rightarrow CD \checkmark$$

$$BC \rightarrow AB \checkmark$$

$$\quad \quad \quad \rightarrow DE \checkmark$$

$$BD \rightarrow DE \checkmark$$

$$BE \rightarrow BF \rightarrow CD \checkmark$$

$$CE \rightarrow AF \checkmark$$

$$CF \rightarrow CE \checkmark$$

$$DF \rightarrow CE \checkmark$$

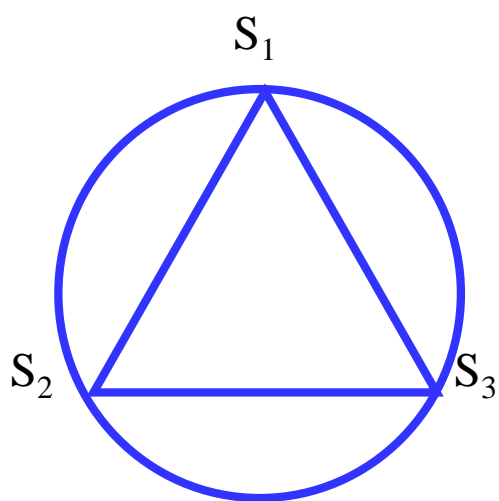
- 关联比较，确定相容对；

以上三步与确定状态表的化简相同

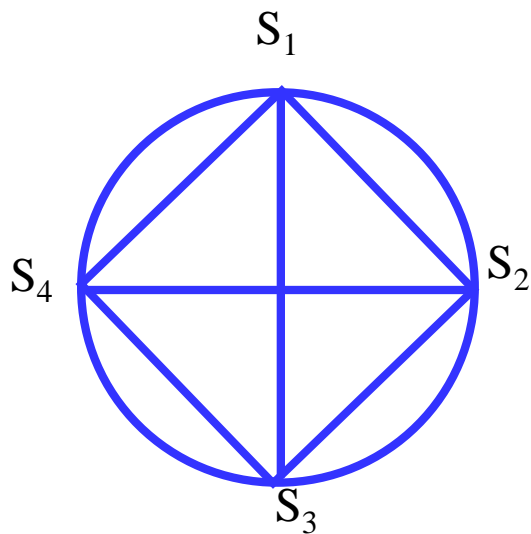
全部相容对：(A,B), (A,F), (B,C), (B,D), (B,E), (B,F),  
(C,D), (C,E), (C,F), (D,F), (D,E), (E,F)。



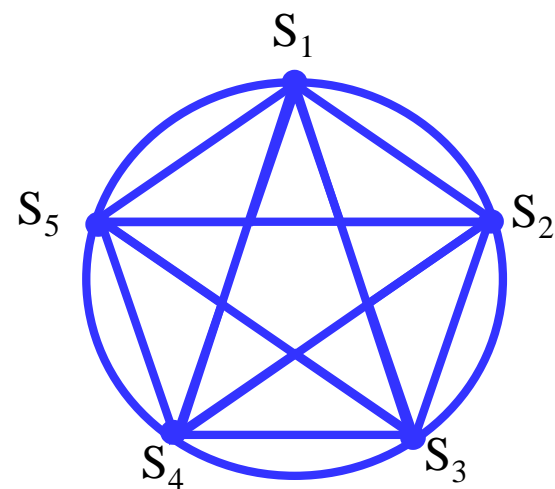
- 作状态合并图，求最大相容类。



3状态相容



4状态相容

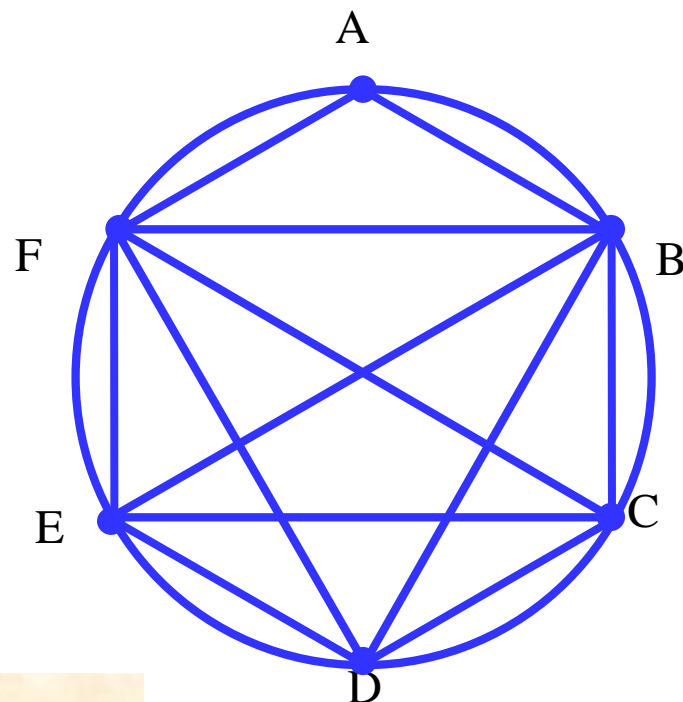


5状态相容



全部相容状态对:

$(A,B), (A,F), (B,C),$   
 $(B,D), (B,E), (B,F),$   
 $(C,D), (C,E), (C,F),$   
 $(D,F), (D,E), (E,F)。$



本例状态合并图，最大相容类是 $(A,B,F), (B,C,D,E,F)。$



**作最小化状态表:** 最小化状态表(又称最小闭覆盖) 应满足下列三个条件:

- 覆盖性——所选相容类集合应包含原始状态表中的全部状态。
- 最小性——所选相容类集合中相容类的个数应最少。
- 闭合性——所选相容类集合中的任一相容类, 在原始状态表中任一输入条件下产生的次态应该属于该集合中的某一个相容类。



采用闭覆盖表来反映所选相容类集合的覆盖和闭合情况。本例的闭覆盖表为



| 现态 | 次态    |       | 输出 |
|----|-------|-------|----|
|    | $x=0$ | $x=1$ |    |
| A  | B     | D     | 0  |
| B  | B     | D     | d  |
| C  | A     | E     | 1  |
| D  | d     | E     | 1  |
| E  | F     | d     | 1  |
| F  | d     | C     | d  |

| 最大相容类 | 覆 盖 |   |   |   |   |   | 闭 合   |       |
|-------|-----|---|---|---|---|---|-------|-------|
|       | A   | B | C | D | E | F | $x=0$ | $x=1$ |
| ABF   | √   | √ |   |   |   | √ | B     | CD    |
| BCDEF |     | √ | √ | √ | √ | √ | ABF   | CDE   |

所选相容类集合  $\{(A,B,F), (B,C,D,E,F)\}$  满足最小闭覆盖条件，令A表示(A,B,F)，C表示(B,C,D,E,F)可得：

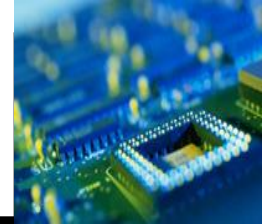
| 最大相容类 | 覆盖 |   |   |   |   |   | 闭 合 |     |
|-------|----|---|---|---|---|---|-----|-----|
|       | A  | B | C | D | E | F | x=0 | x=1 |
| ABF   | √  | √ |   |   |   | √ | B   | CD  |
| BCDEF |    | √ | √ | √ | √ | √ | ABF | CDE |

| 现态 | 次态  |     | 输出 |
|----|-----|-----|----|
|    | x=0 | x=1 |    |
| A  | B   | D   | 0  |
| B  | B   | D   | d  |
| C  | A   | E   | 1  |
| D  | d   | E   | 1  |
| E  | F   | d   | 1  |
| F  | d   | C   | d  |

| 现 态 | 次 态 |     | 输 出 |
|-----|-----|-----|-----|
|     | x=0 | x=1 |     |
| A   | A,C | C   | 0   |
| C   | A   | C   | 1   |

由于该表中只有两个状态，进一步可以得到：

| 现 态 | 次 态 |     | 输 出 |
|-----|-----|-----|-----|
|     | x=0 | x=1 |     |
| A   | d   | C   | 0   |
| C   | A   | C   | 1   |



例：化简下表所示的状态表

| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | D/d   | A/d   |
| B  | E/0   | A/d   |
| C  | D/0   | B/d   |
| D  | C/d   | C/d   |
| E  | C/1   | B/d   |





解： 作隐含表，寻找相容状态对



| 现态 | 次态/输出 |     | B  | C        | D  | E  | A  | B  | C | D | E |
|----|-------|-----|----|----------|----|----|----|----|---|---|---|
|    | x=0   | x=1 |    |          |    |    |    |    |   |   |   |
| A  | D/d   | A/d | DE | AB       | CD | AC | AB | CD |   |   |   |
| B  | E/0   | A/d |    | AB<br>DE | AC | CE |    |    |   |   |   |
| C  | D/0   | B/d |    |          |    |    |    |    |   |   |   |
| D  | C/d   | C/d |    |          |    |    |    |    |   |   |   |
| E  | C/1   | B/d |    |          |    |    |    |    |   |   |   |

AB→DE→BC ✓

AC→AB ✓

AD→CD→BC ✓

AD→AC ✓

AE→AB ✓

AE→CD ✓

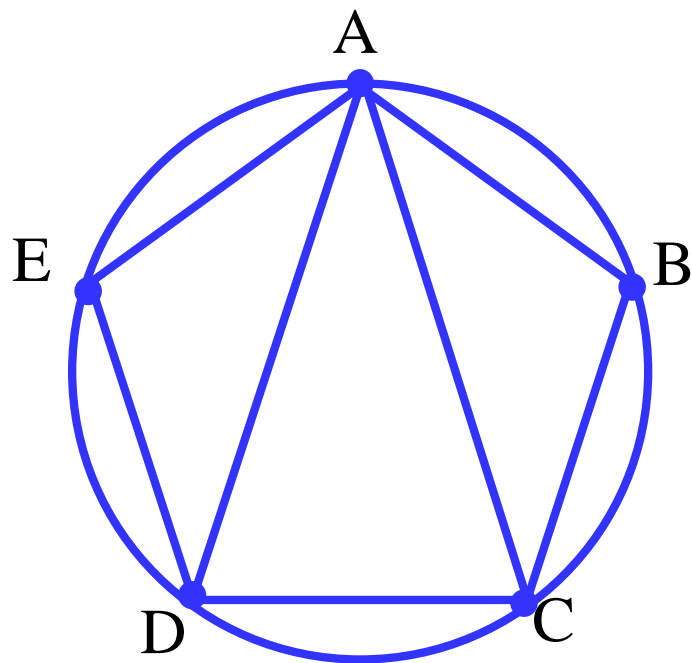
BD→AC ✓

BD→CE ✗

由上图得相容状态对为(A,B), (A,C), (A,D), (A,E),  
(B,C), (C,D), (D,E)



- 作状态合并图，寻找最大相容类



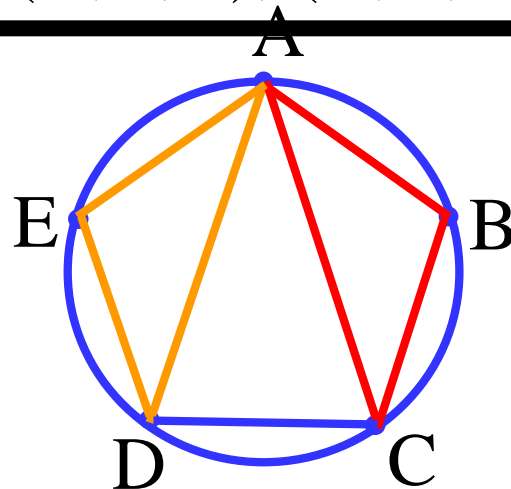
得最大相容类为(A,B,C), (A,C,D), (A,D,E)



# 作最小化状态表

$(A,B,C), (A,C,D), (A,D,E)$

若选相容类集合为 $\{(A,B,C), (A,D,E)\}$ 则下表表明它不满足闭合要求



次态/输出

现态

$x=0$

$x=1$

A

D/d

A/d

B

E/0

A/d

C

D/0

B/d

D

C/d

C/d

E

C/1

B/d

相容类

覆盖

闭合

A

B

C

D

E

$x=0$

$x=1$

ABC

√

√

√

DE

AB

ADE

√

√

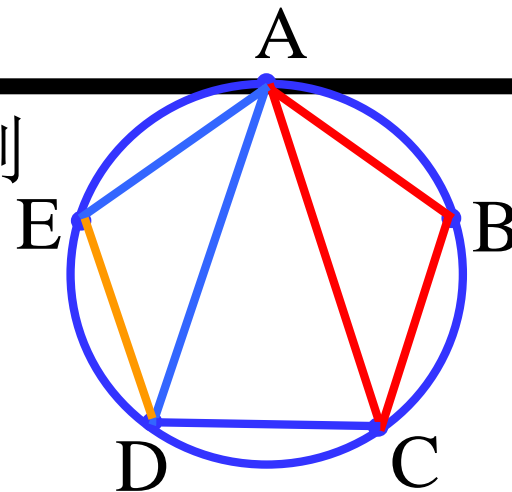
√

CD

ABC



但如果选相容类 (A,B,C)和 (D,E)则  
能满足最小闭覆盖的要求



| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | D/d   | A/d   |
| B  | E/0   | A/d   |
| C  | D/0   | B/d   |
| D  | C/d   | C/d   |
| E  | C/1   | B/d   |

| 相容类 | 覆 盖 |   |   |   |   | 闭 合   |       |
|-----|-----|---|---|---|---|-------|-------|
|     | A   | B | C | D | E | $x=0$ | $x=1$ |
| ABC | A   | B | C |   |   | DE    | AB    |
| DE  |     |   |   | D | E | C     | BC    |



| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | D/d   | A/d   |
| B  | E/0   | A/d   |
| C  | D/0   | B/d   |
| D  | C/d   | C/d   |
| E  | C/1   | B/d   |

令  $A'=(A,B,C)$ ,  $B'=(D,E)$ , 进一步可得:

| 现态   | 次态/输出    |          |
|------|----------|----------|
|      | $x=0$    | $x=1$    |
| $A'$ | $B' / 0$ | $A' / d$ |
| $B'$ | $A' / 1$ | $A' / d$ |

寻找最小闭覆盖通常不是一件容易的事情，其结果往往不唯一。



# 状态化简真的必要吗？



- 对于大多数设计者而言，**很少**采用这些正规化的过程。
- 通过将状态的意义与问题的要求进行细心地匹配，有经验的设计者想得到最少或接近最少数目的状态，几乎不成问题，而无需最小化过程。
- 在有些情况下，增加状态的数目还可能**简化设计**过程或**降低设计成本**。即使是自动的状态最小化过程也是不一定有帮助的。



## 7.4.3、状态赋值(状态编码)



- 确定表示状态表的状态需要多少位二进制变量，并且对每一个已命名的状态赋予一个特定的组合（编码）。
- 状态编码(coded state): 赋给一个特定状态的二进制数的组合。
- 状态编码要解决两个问题:
  - 根据所要求的状态数确定触发器的数目。
    - $s$ 个状态，触发器的个数为： $\lfloor \log_2^s \rfloor$
  - 指定每个状态的二进制代码，使所设计的电路尽量简单。



# 状态赋值



- 状态的赋值方式通常对电路的成本有很大影响。
- 状态赋值与其他因素相互作用：触发器的选择、激励和输出逻辑的实现方式。
- 状态分配方案：状态分配排列
  - 对使用K个状态变量(即K个触发器，K位二进制编码)实现N个状态时，可能的状态分配数为： $N_{SA} = 2^K! / (2^K - N)!$   
例：N=4，K=2共有24种分配方案。
  - 如果两种状态分配在实现逻辑时产生相同的结果，则认为它们是等价的。状态编码分配互补，或状态编码左右互换，都是等价的。

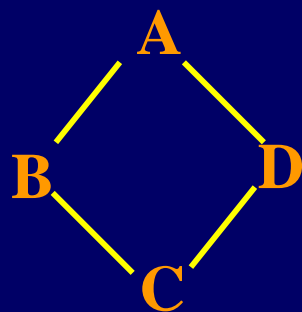




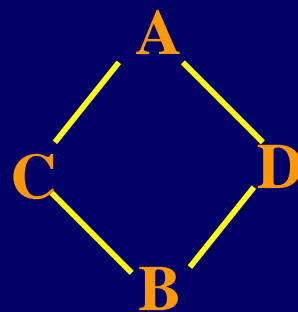
# n = 4, K = 2 全部状态分配方案

| 第一组  | 方案 | 1         | 2                    | 3                    | 4                               | 5         | 6                    | 7                    | 8                               |
|------|----|-----------|----------------------|----------------------|---------------------------------|-----------|----------------------|----------------------|---------------------------------|
|      | A  | 00        |                      |                      |                                 |           |                      |                      |                                 |
|      | B  | 01        |                      |                      |                                 |           |                      |                      |                                 |
|      | C  | 11        |                      |                      |                                 |           |                      |                      |                                 |
|      | D  | 10        |                      |                      |                                 |           |                      |                      |                                 |
| 第二组  | 方案 | 1         |                      |                      |                                 |           |                      |                      |                                 |
|      | A  | 00        |                      |                      |                                 |           |                      |                      |                                 |
|      | B  | 11        |                      |                      |                                 |           |                      |                      |                                 |
|      | C  | 01        |                      |                      |                                 |           |                      |                      |                                 |
|      | D  | 10        |                      |                      |                                 |           |                      |                      |                                 |
| 第三组  | 方案 | 1         |                      |                      |                                 |           |                      |                      |                                 |
|      | A  | 00        |                      |                      |                                 |           |                      |                      |                                 |
|      | B  | 10        |                      |                      |                                 |           |                      |                      |                                 |
|      | C  | 01        |                      |                      |                                 |           |                      |                      |                                 |
|      | D  | 11        |                      |                      |                                 |           |                      |                      |                                 |
| 状态变量 |    | $y_1 y_0$ | $\overline{y_1} y_0$ | $y_1 \overline{y_0}$ | $\overline{y_1} \overline{y_0}$ | $y_0 y_1$ | $y_0 \overline{y_1}$ | $\overline{y_0} y_1$ | $\overline{y_0} \overline{y_1}$ |

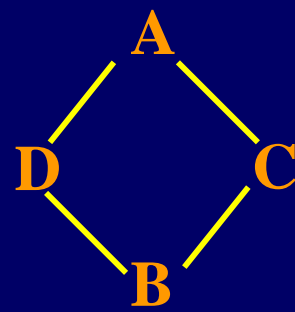
实际上，24种方案仅有 **3种** 分配方案是完全独立的，即表中的三个大组，每大组对应着一个电路设计。它们的主要差别表现在：



第一组



第二组



第三组



# 真正独立的状态分配方案总数

## *Unique State Assignment*



如果触发器个数为 **K**，有  $2^K$  种二进制组合，  
用来对 **n** 个状态进行分配时就有  $N_{uA}$  种独立的分配  
方案：

$$N_{uA} = \frac{(2^K - 1)!}{(2^K - n)! K!}$$

状态分配数 *Number of State Assignments*

| n | K | $N_{SA}$ | $N_{UA}$ | n  | K | $N_{SA}$              | $N_{UA}$ |
|---|---|----------|----------|----|---|-----------------------|----------|
| 2 | 1 | 2        | 1        | 7  | 3 | 40320                 | 840      |
| 3 | 2 | 24       | 3        | 8  | 3 | 40320                 | 840      |
| 4 | 2 | 24       | 3        | 9  | 4 | $4.15 \times 10^9$    | 10810800 |
| 5 | 3 | 6720     | 140      | 10 | 4 | $2.91 \times 10^{10}$ | 75675600 |
| 6 | 3 | 20160    | 420      |    |   |                       |          |



# 哪种编码方案最佳？



- 在同步时序电路中状态分配目的在于：在逻辑化简时，生成尽可能大的必要质蕴含。状态分配的标准就是使得质蕴含达到最大程度。寻找较好的结果。
- 状态编码：从 $2^n$ 种可能组合中选择 $S$ 种编码
  - 采用相邻状态编码方案，或可根据应用选用两种处理方法
- 遗憾的是：至今没有找到普遍有效的算法实现最佳状态分配，唯一途径是将所有分配方案都试个遍
- 次佳状态分配方案：相邻状态分配法，建立通用方程法，减少相关性



# 次佳状态分配方案



- 相邻状态分配法

- 次态相同，现态相邻

使下一个状态较少依赖于当前状态变量

$$Q_{n+1} = Q(Q_n, W_n)$$

- 同一现态，次态相邻

使下一个状态较少依赖于输入变量

$$Q_{n+1} = Q(Q_n, X_n)$$

- 输出相同，现态相邻

使输出较少依赖于当前状态变量

$$Z_n = Z(Q_n, X_n)$$



# 次佳状态分配示例



| Pre-sent State | Next State |     | OutPut |     |
|----------------|------------|-----|--------|-----|
|                | X=0        | X=1 | X=0    | X=1 |
| S1             | S3         | S2  | 1      | 1   |
| S2             | S7         | S8  | 0      | 0   |
| S3             | S6         | S1  | 0      | 0   |
| S4             | S4         | S5  | 0      | 0   |
| S5             | S3         | S2  | 0      | 0   |
| S6             | S4         | S5  | 1      | 1   |
| S7             | S6         | S1  | 1      | 1   |
| S8             | S7         | S8  | 1      | 1   |

| 状态 | 状态代码 |   |   |
|----|------|---|---|
|    | A    | B | C |
| S1 | 0    | 0 | 0 |
| S2 | 1    | 0 | 1 |
| S3 | 1    | 1 | 1 |
| S4 | 1    | 1 | 0 |
| S5 | 1    | 0 | 0 |
| S6 | 0    | 1 | 0 |
| S7 | 0    | 1 | 1 |
| S8 | 0    | 0 | 1 |

- 规则1(次态相同): S1S5, S2S8 , S3S7 , S4S6
- 规则2(现态相同): S1S6 , S4S5, S2S3, S7S8
- 规则3(输出相同): S1S6 S7S8, S2S3 S4S5



# 状态赋值-处理未用状态方法 $2^n > S$



- 最小风险法(Minimal risk)
  - 基于状态机有可能进入未用状态的假设
  - 状态机每种未用状态，都明确其下一个状态为初始态、空闲态或其它“安全”状态  
常用设置——全0状态
- 最小成本法(Minimal cost)
  - 未用状态标识为无关项，利用无关项，可以达到化简目的
  - 基于状态机不可能进入未用状态这一前提
  - 一旦状态机进入未用状态，行为不用预知
  - 因此，需要增加检查步骤：确认能进入正常状态



# 常用状态编码方案



- Sequential state assignment (顺序赋值)
- Decomposed state assignment (分解赋值)
  - 一个大型状态机用若干小型状态机的集合来实现
- Cyclic-code assignment (循环码)
  - 多余状态较多
  - 激励方程简单
- One-hot assignment (单热点)
  - 多余状态很多
  - 激励方程简单，尤其适合 **s选1编码输出情形**
- Almost one-hot assignment (准单热点)
  - 相对于 **One-hot assignment**，增加全**0**状态



# 状态赋值



| 含 义            | State     | Inputs AB=<br>00 01 11 10 | Output<br>Z |
|----------------|-----------|---------------------------|-------------|
| 初态             | Init      | $S_0 S_0 S_1 S_1$         | 0           |
| 时钟采样时刻, A=0    | $S_0$     | $0k_0 0k_0 S_1 S_1$       | 0           |
| 时钟采样时刻, A=1    | $S_1$     | $S_0 S_0 0k_1 0k_1$       | 0           |
| 两个A输入相等, 上一A=0 | $0k_0$    | $0k_0 0k_0 0k_1 S_1$      | 1           |
| 两个A输入相等, 上一A=1 | $0k_{11}$ | $S_0 0k_0 0k_1 0k_1$      | 1           |

## Assignment

| State<br>Name | Simplest<br>Q1-Q3 | Decomposed<br>Q1-Q3 | One-hot<br>Q1-Q5 | Almost One-hot<br>Q1-Q4 |
|---------------|-------------------|---------------------|------------------|-------------------------|
| INIT          | 000               | 000                 | 00001            | 0000                    |
| A0            | 001               | 100                 | 00010            | 0001                    |
| A1            | 010               | 101                 | 00100            | 0010                    |
| OK0           | 011               | 110                 | 01000            | 0100                    |
| OK1           | 100               | 111                 | 10000            | 1000                    |





## 7.4.4 采用D触发器综合



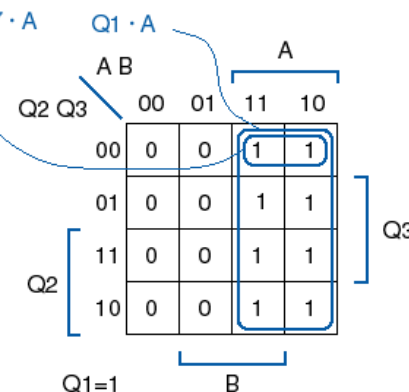
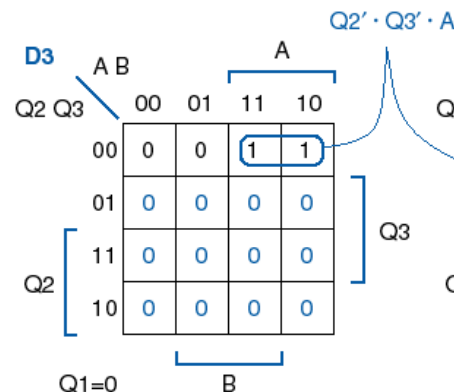
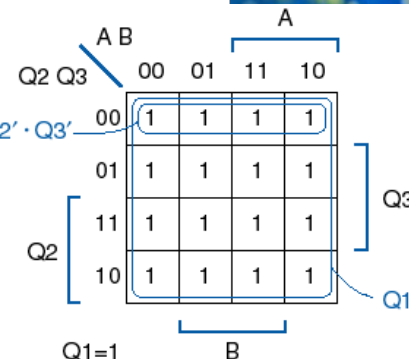
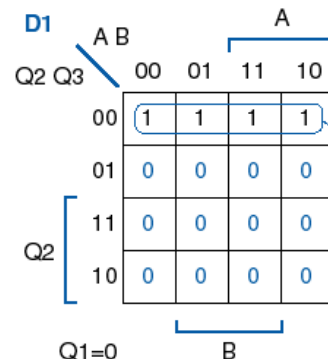
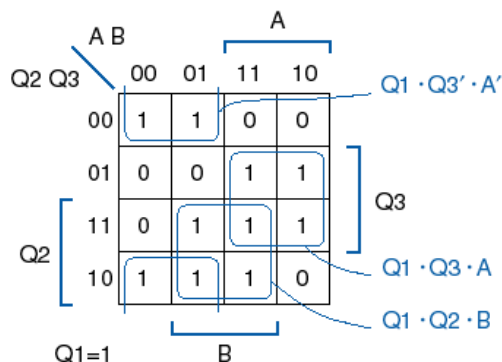
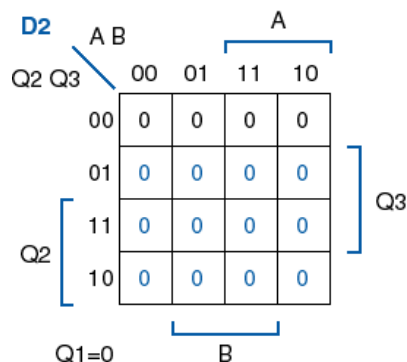
| S    | A B |     |     |     | Z |
|------|-----|-----|-----|-----|---|
|      | 00  | 01  | 11  | 10  |   |
| INIT | A0  | A0  | A1  | A1  | 0 |
| A0   | OK0 | OK0 | A1  | A1  | 0 |
| A1   | A0  | A0  | OK1 | OK1 | 0 |
| OK0  | OK0 | OK0 | OK1 | A1  | 1 |
| OK1  | A0  | OK0 | OK1 | OK1 | 1 |
| S*   |     |     |     |     |   |

| Q1 Q2 Q3    | A B |     |     |     | Z |
|-------------|-----|-----|-----|-----|---|
|             | 00  | 01  | 11  | 10  |   |
| 000         | 100 | 100 | 101 | 101 | 0 |
| 100         | 110 | 110 | 101 | 101 | 0 |
| 101         | 100 | 100 | 111 | 111 | 0 |
| 110         | 110 | 110 | 111 | 101 | 1 |
| 111         | 100 | 110 | 111 | 111 | 1 |
| Q1* Q2* Q3* |     |     |     |     |   |

D1 D2 D3

- 根据编码状态取代命名的状态，得到转移和输出表
- 根据采用触发器特征方程 ( $Q^*=D$ )，得到激励表和输出表。五个输入变量(Q1、Q2、Q3、A、B)三个输出变量 (D1、D2、D3)。

# 激励方程



- 假设最小风险法，未用状态的次态为000

$$D1 = Q1 + Q2' \cdot Q3'$$

$$D2 = Q1 \cdot Q3' \cdot A' + Q1 \cdot Q3 \cdot A + Q1 \cdot Q2 \cdot B$$

$$D3 = Q1 \cdot A + Q2' \cdot Q3' \cdot A$$



# 最小成本法化简

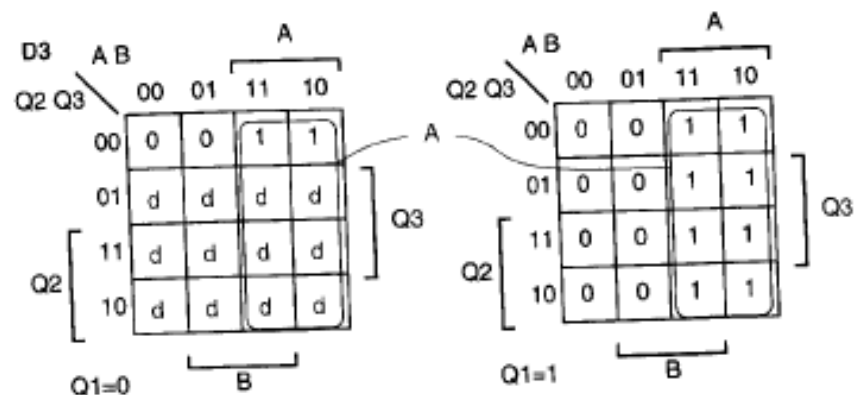
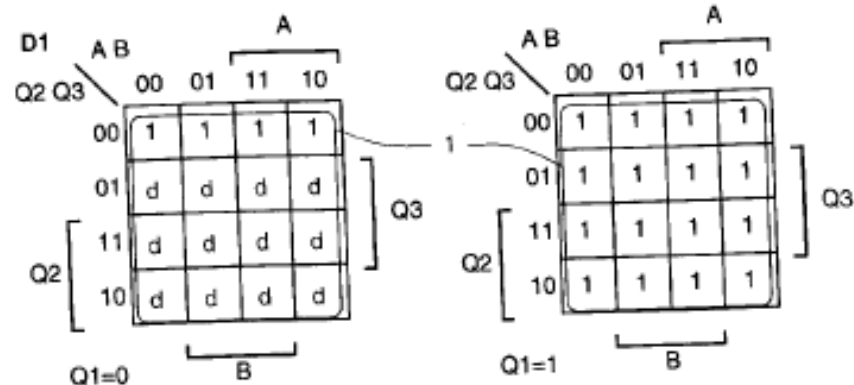
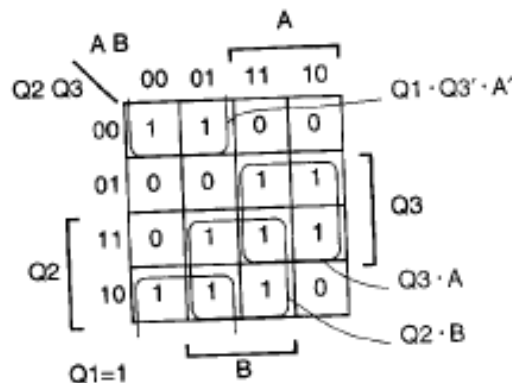
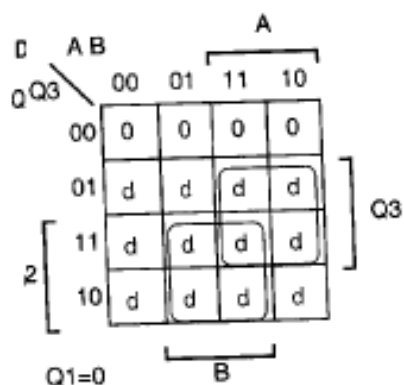


## 使用无关项优化

**D=1**

**D2=Q1 • Q3' • A' + Q3 • A + Q2 • B**

**D3= A**

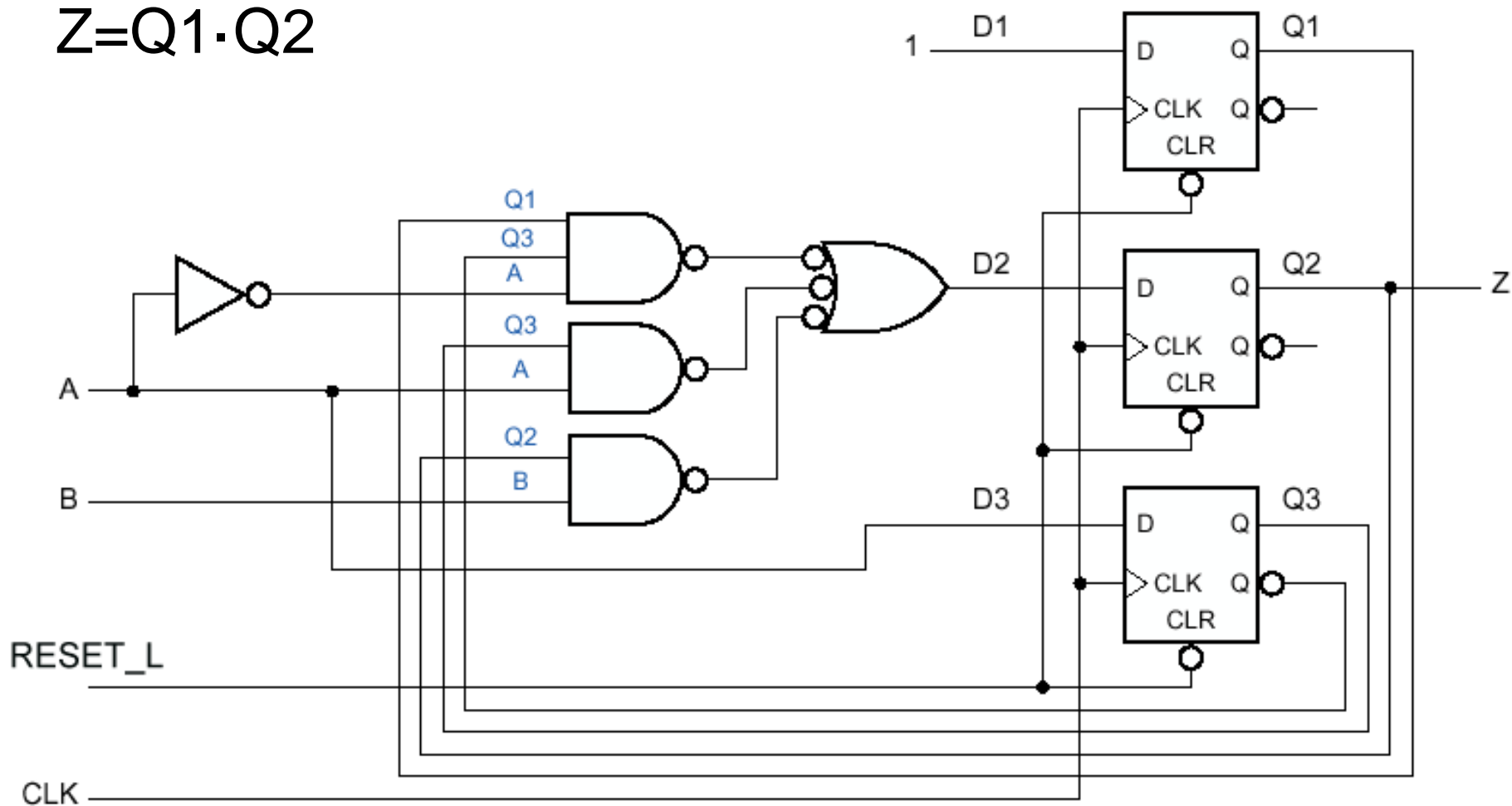




# 最小化电路



$$Z = Q1 \cdot Q2$$





## 7.4.5 采用JK触发器综合



- 在小规模集成电路设计中，有一段时期流行采用JK触发器，因为JK触发器所包含的功能较多（置位/复位/翻转），对于同样的状态机，JK触发器的激励逻辑比D触发器要简单，可以减少组件的数目。
- 在现代设计中，使用PLD、FPGA和ASIC，为JK触发器的输入端JK提供独立的AND-OR阵列比较困难。JK触发器占用的芯片面积比D触发器大25%。



# JK触发器转移表

| $Q$ | $Q^*$ | $J$ | $K$ |
|-----|-------|-----|-----|
| 0   | 0     | 0   | d   |
| 0   | 1     | 1   | d   |
| 1   | 0     | d   | 1   |
| 1   | 1     | d   | 0   |

| $AB$                     |            |            |            |            |     |
|--------------------------|------------|------------|------------|------------|-----|
| $Q1\ Q2\ Q3$             | $00$       | $01$       | $11$       | $10$       | $Z$ |
| 000                      | 1d, 0d, 0d | 1d, 0d, 0d | 1d, 0d, 1d | 1d, 0d, 1d | 0   |
| 100                      | d0, 1d, 0d | d0, 1d, 0d | d0, 0d, 1d | d0, 0d, 1d | 0   |
| 101                      | d0, 0d, d1 | d0, 0d, d1 | d0, 1d, d0 | d0, 1d, d0 | 0   |
| 110                      | d0, d0, 0d | d0, d0, 0d | d0, d0, 1d | d0, d1, 1d | 1   |
| 111                      | d0, d1, d1 | d0, d0, d1 | d0, d0, d0 | d0, d0, d0 | 1   |
| $J1\ K1, J2\ K2, J3\ K3$ |            |            |            |            |     |



## 7.4.6 采用D触发器的其他例子



- 1计数器

- 当X和Y输入1的个数为4的整数倍是，输出为1，否则为0

| Meaning                 | S  | XY |    |    |    | Z |
|-------------------------|----|----|----|----|----|---|
|                         |    | 00 | 01 | 11 | 10 |   |
| Got zero 1s (modulo 4)  | S0 | S0 | S1 | S2 | S1 | 1 |
| Got one 1 (modulo 4)    | S1 | S1 | S2 | S3 | S2 | 0 |
| Got two 1s (modulo 4)   | S2 | S2 | S3 | S0 | S3 | 0 |
| Got three 1s (modulo 4) | S3 | S3 | S0 | S1 | S0 | 0 |
|                         |    | S* |    |    |    |   |

| Q1 Q2 | XY |                  |    |    | Z |
|-------|----|------------------|----|----|---|
|       | 00 | 01               | 11 | 10 |   |
| 00    | 00 | 01               | 11 | 01 | 1 |
| 01    | 01 | 11               | 10 | 11 | 0 |
| 11    | 11 | 10               | 00 | 10 | 0 |
| 10    | 10 | 00               | 01 | 00 | 0 |
|       |    | Q1* Q2* or D1 D2 |    |    |   |



## 7.4.6 采用D触发器的其他例子



- 序列检测器（密码锁）
  - 设计一个1输入X和2输出(Unlk和Hint)同步时序电路。当且仅当
    - X=0且前面7个时钟采样时刻X收到的输入序列为0110111时，输出Unlk为1
    - X当前输入是上述序列(01101110)中以使状态机逐步接近于解锁状态即Unlk=1的比特时，输出Hint为1。





# 序列检测器（密码锁）



- 1输入(X)2输出(Unlk, Hint)同步时序电路.  
当且仅当
  - $X=0$ 且之前时刻 $X$ 收到的输入序列为0110111，输出Unlk为1
  - $X$ 当前输入是上述序列中以使Unlk=1的比特时，输出Hint为1

| <i>Meaning</i> | <i>S</i> | <i>X</i>              |          |
|----------------|----------|-----------------------|----------|
|                |          | <i>0</i>              | <i>1</i> |
| Got zip        | A        | B, 01                 | A, 00    |
| Got 0          | B        | B, 00                 | C, 01    |
| Got 01         | C        | B, 00                 | D, 01    |
| Got 011        | D        | E, 01                 | A, 00    |
| Got 0110       | E        | B, 00                 | F, 01    |
| Got 01101      | F        | B, 00                 | G, 01    |
| Got 011011     | G        | E, 00                 | H, 01    |
| Got 0110111    | H        | B, 11                 | A, 00    |
|                |          | <i>S*</i> , UNLK HINT |          |



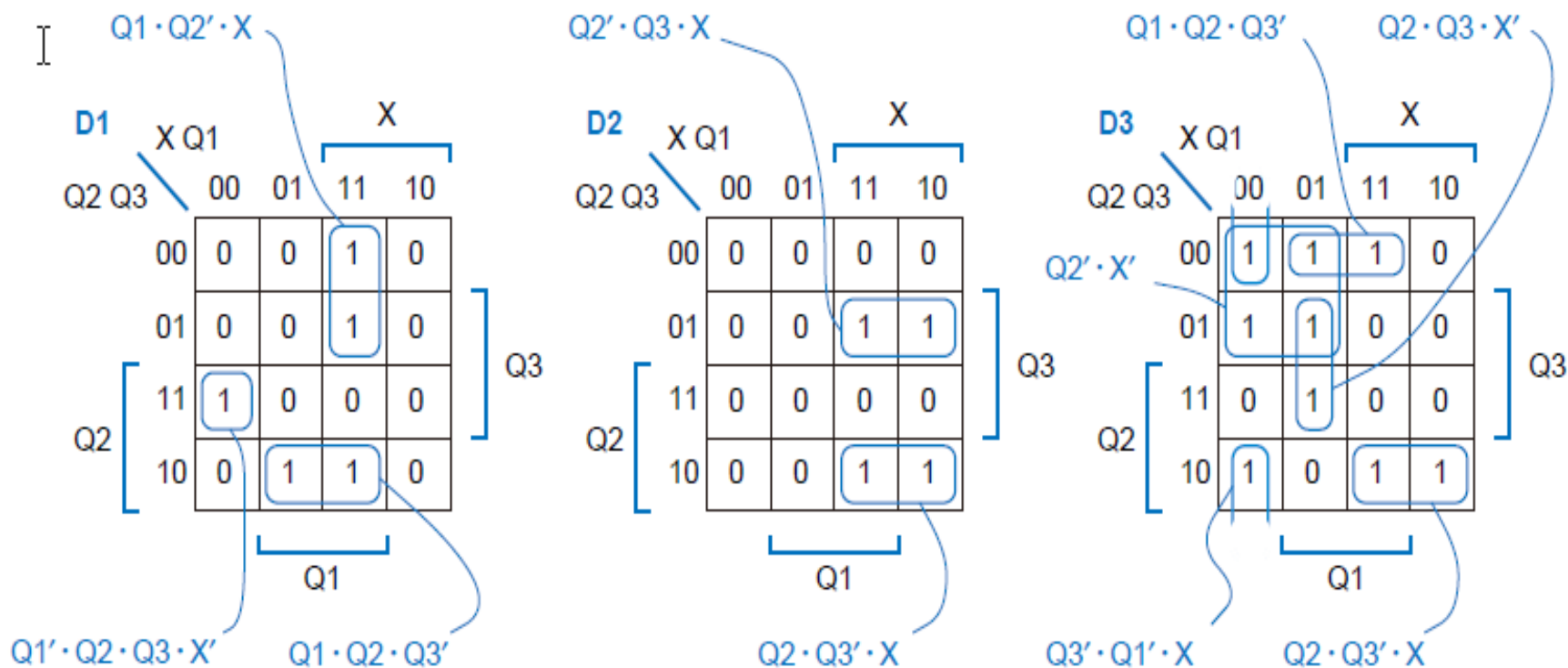
# 序列检测器（密码锁）



| <i>Q1 Q2 Q3</i>               | <i>X</i> |          |
|-------------------------------|----------|----------|
|                               | <i>0</i> | <i>1</i> |
| 000                           | 001, 01  | 000, 00  |
| 001                           | 001, 00  | 010, 01  |
| 010                           | 001, 00  | 011, 01  |
| 011                           | 100, 01  | 000, 00  |
| 100                           | 001, 00  | 101, 01  |
| 101                           | 001, 00  | 110, 01  |
| 110                           | 100, 00  | 111, 01  |
| 111                           | 001, 11  | 000, 00  |
| <i>Q1* Q2* Q3*, UNLK HINT</i> |          |          |



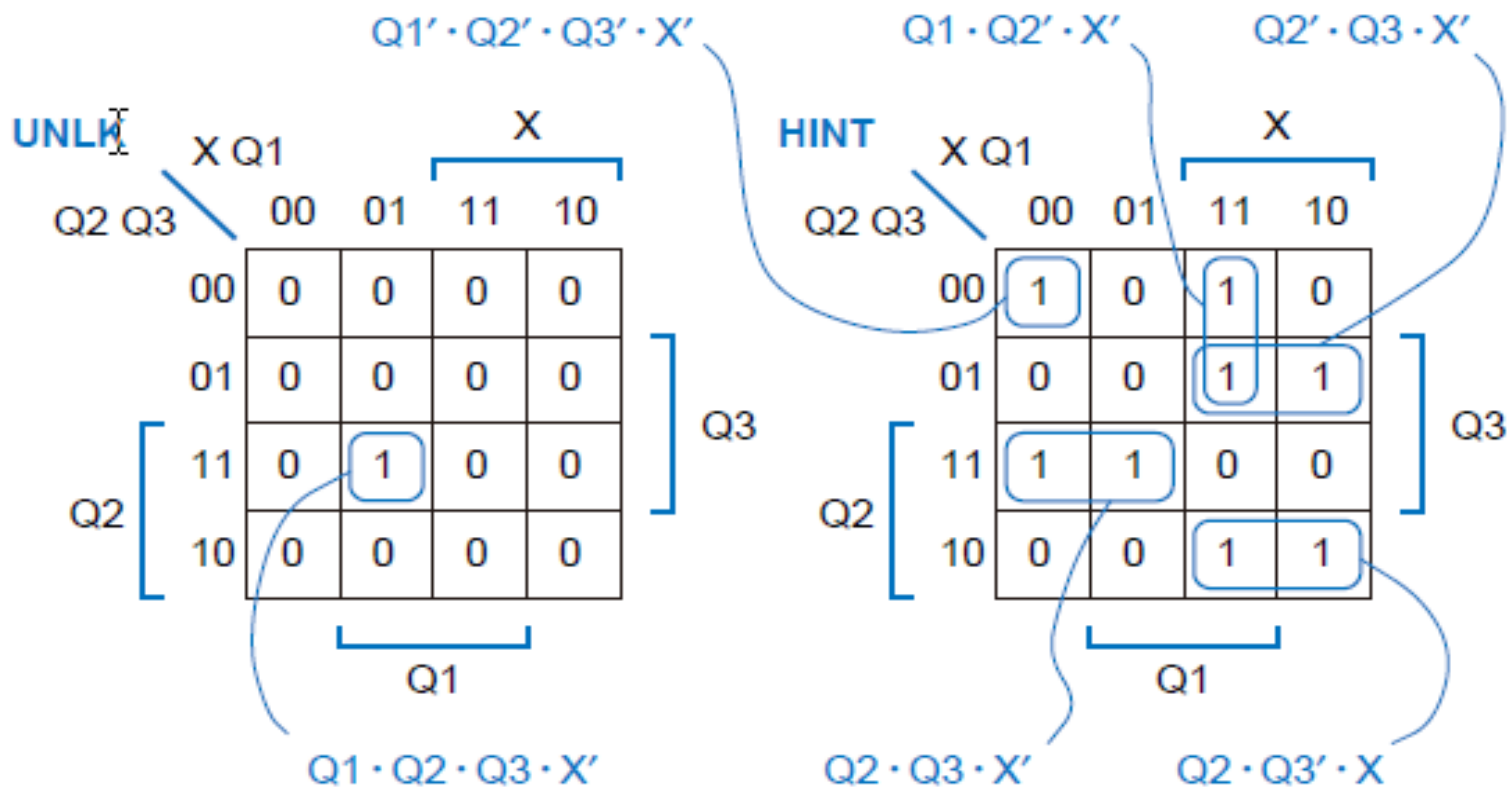
# 序列检测器（密码锁）



- $D1 = Q1 \cdot Q2' \cdot X + Q1' \cdot Q2 \cdot Q3 \cdot X' + Q1 \cdot Q2 \cdot Q3'$
- $D2 = Q2' \cdot Q3 + Q2 \cdot Q3' \cdot X$
- $D3 = Q1 \cdot Q2' \cdot Q3' + Q1 \cdot Q3 \cdot X' + Q2' \cdot X' + Q3' \cdot Q1' \cdot X' + Q2 \cdot Q3' \cdot X$



# 序列检测器（密码锁）



- $UNLK = Q1 \cdot Q2 \cdot Q3 \cdot X'$
- $HINT = Q1' \cdot Q2' \cdot Q3' \cdot X' + Q1 \cdot Q2' \cdot X + Q2' \cdot Q3 \cdot X + Q2 \cdot Q3 \cdot X' + Q2 \cdot Q3' \cdot X$



# 补充：同步时序逻辑电路设计举例



例：序列检测器（密码锁）

设计一个“111...”序列检测器，用来检测串行二进制序列，要求每当连续输入3个(或3个以上)1时，检测器输出为1，否则输出为0。其典型输入输出序列如下：

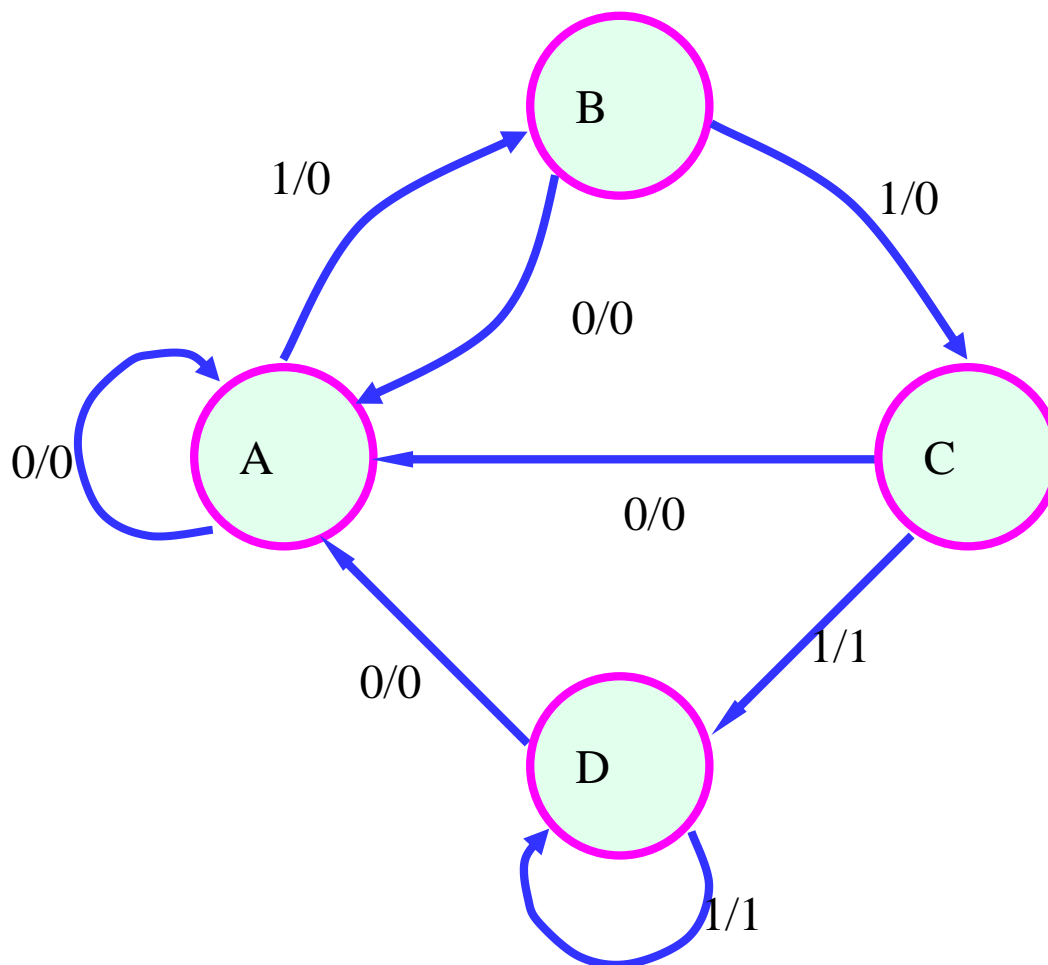
输入x: 0    1    1    1    0    1    1    1    1    0

输出Z: 0    0    0    1    0    0    0    1    1    0



解：

- 作状态图和状态表





| 现 态 | 次态/输出Z |       |
|-----|--------|-------|
|     | $x=0$  | $x=1$ |
| A   | A/0    | B/0   |
| B   | A/0    | C/0   |
| C   | A/0    | D/1   |
| D   | A/0    | D/1   |



## • 状态化简



用观察法可得最大等价类为: (A), (B), (C,D)

令 $C=(C,D)$ ,可得下列最简状态表

| 现 态 | 次态/输出Z |       |
|-----|--------|-------|
|     | $x=0$  | $x=1$ |
| A   | A/0    | B/0   |
| B   | A/0    | C/0   |
| C   | A/0    | D/1   |
| D   | A/0    | D/1   |

| 现 态 | 次态/输出Z |       |
|-----|--------|-------|
|     | $x=0$  | $x=1$ |
| A   | A/0    | B/0   |
| B   | A/0    | C/0   |
| C   | A/0    | C/1   |

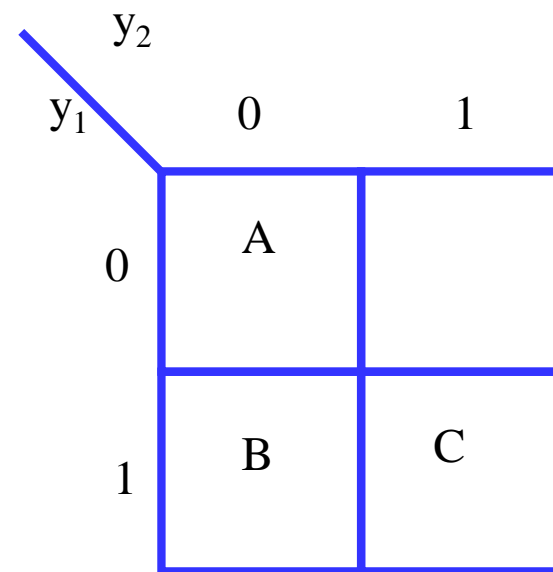


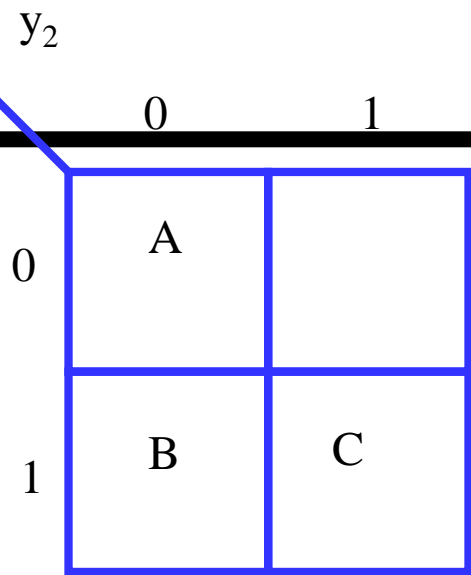


状态分配:

| 现 态 | 次态/输出Z |       |
|-----|--------|-------|
|     | $x=0$  | $x=1$ |
| A   | A/0    | B/0   |
| B   | A/0    | C/0   |
| C   | A/0    | C/1   |

- AB, BC, AC应相邻
- AB, AC应相邻
- AB应相邻
- A应为逻辑0





| 现 态 | 次态/输出Z            |                   |
|-----|-------------------|-------------------|
|     | $x=0$             | $x=1$             |
|     | A/0<br>A/0<br>A/0 | B/0<br>C/0<br>C/1 |

| 现态<br>$y_2 \ y_1$ | 次态 $y_2^{(n+1)}y_1^{(n+1)}$ /输出Z |       |
|-------------------|----------------------------------|-------|
|                   | $x=0$                            | $x=1$ |
| 0 0               | 00/0                             | 01/0  |
| 0 1               | 00/0                             | 11/0  |
| 1 1               | 00/0                             | 11/1  |

确定激励函数和输出函数表达式：

选用J—K触发器

| Q | Q <sup>(n+1)</sup> | J | K |
|---|--------------------|---|---|
| 0 | 0                  | 0 | d |
| 0 | 1                  | 1 | d |
| 1 | 0                  | d | 1 |
| 1 | 1                  | d | 0 |

| 输 入<br>x | 现 态<br>y <sub>2</sub> y <sub>1</sub> |   | 次 态<br>y <sub>2</sub> <sup>(n+1)</sup> y <sub>1</sub> <sup>(n+1)</sup> |   | 激励函数<br>J <sub>2</sub> K <sub>2</sub> J <sub>1</sub> K <sub>1</sub> |   |   |   | 输 出<br>Z |
|----------|--------------------------------------|---|--|---|---|---|---|---|----------|
| 0        | 0                                    | 0 | 0  | 0 | 0   | d | 0 | d | 0        |
| 0        | 0                                    | 1 | 0  | 0 | 0   | d | d | 1 | 0        |
| 0        | 1                                    | 1 | 0  | 0 | d   | 1 | d | 1 | 0        |
| 0        | 1                                    | 0 | d  | d | d   | d | d | d | d        |
| 1        | 0                                    | 0 | 0  | 1 | 0   | d | 1 | d | 0        |
| 1        | 0                                    | 1 | 1  | 1 | 1   | d | d | 0 | 0        |
| 1        | 1                                    | 1 | 1  | 1 | d   | 0 | d | 0 | 1        |
| 1        | 1                                    | 0 | d  | d | d   | d | d | d | d        |

| 输入<br>x | 现态             |                | 激励函数           |                |                |                | 输 出<br>Z |   |
|---------|----------------|----------------|----------------|----------------|----------------|----------------|----------|---|
|         | y <sub>2</sub> | y <sub>1</sub> | J <sub>2</sub> | K <sub>2</sub> | J <sub>1</sub> | K <sub>1</sub> |          |   |
| 0       | 0              | 0              | 0              | 0              | d              | 0              | d        | 0 |
| 0       | 0              | 1              | 1              | 0              | d              | d              | 1        | 0 |
| 0       | 1              | 1              | 1              | d              | 1              | d              | 1        | 0 |
| 0       | 1              | 0              | 0              | d              | d              | d              | d        | d |
| 1       | 0              | 0              | 0              | 0              | d              | 1              | d        | 0 |
| 1       | 0              | 1              | 1              | 1              | d              | d              | 0        | 0 |
| 1       | 1              | 1              | 1              | d              | 0              | d              | 0        | 1 |
| 1       | 1              | 0              | 0              | d              | d              | d              | d        | d |

| $y_1 \backslash xy_2$ | 00 | 01 | 11 | 10 |
|-----------------------|----|----|----|----|
| 0                     | 0  | d  | d  | 0  |
| 1                     | 0  | d  | d  | 1  |

| $y_1 \backslash xy_2$ | 00 | 01 | 11 | 10 |
|-----------------------|----|----|----|----|
| 0                     | d  | d  | d  | d  |
| 1                     | d  | 1  | 0  | d  |

| $y_1 \backslash xy_2$ | 00 | 01 | 11 | 10 |
|-----------------------|----|----|----|----|
| 0                     | 0  | d  | d  | 1  |
| 1                     | d  | d  | d  | d  |

| $y_1 \backslash xy_2$ | 00 | 01 | 11 | 10 |
|-----------------------|----|----|----|----|
| 0                     | d  | d  | d  | d  |
| 1                     | 1  | 1  | 0  | 0  |

| $y_1 \backslash xy_2$ | 00 | 01 | 11 | 10 |
|-----------------------|----|----|----|----|
| 0                     | 0  | d  | d  | 0  |
| 1                     | 0  | 0  | 1  | 0  |

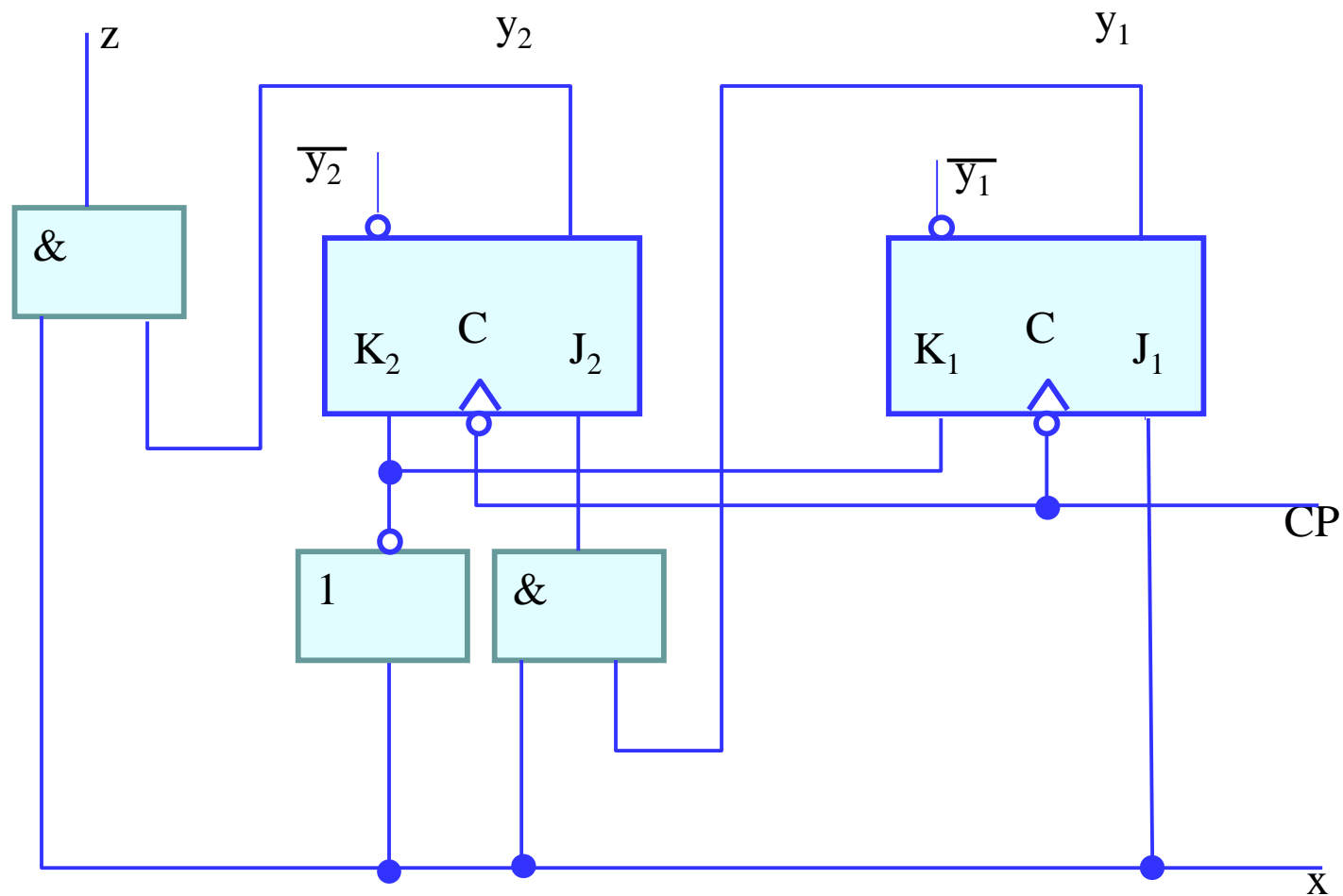
$$\begin{aligned}
 J_2 &= xy_1, & k_2 &= \bar{x} \\
 J_1 &= x, & k_1 &= x \\
 Z &= xy_2
 \end{aligned}$$



- 画电路图

$$\begin{aligned}J_2 &= xy_1, \\ J_1 &= x, \\ Z &= xy_2\end{aligned}$$

$$\begin{aligned}k_2 &= \bar{x} \\ k_1 &= \bar{x}\end{aligned}$$





- 分析：

由于电路有冗余状态“10”，一旦电路进入“10”状态，不管输入为0还是1，经过一个时钟周期后，电路应自动进入有效状态，否则电路存在“挂起”现象。

## 分析方法为：

确定无效状态的次态，由于无效状态的次态为d，所以在化简的卡诺图中，被卡诺圈圈起的d为1，没有被卡诺圈圈起的d为0。然后判断无效状态的次态是否为有效状态或是否存在“挂起”现象。



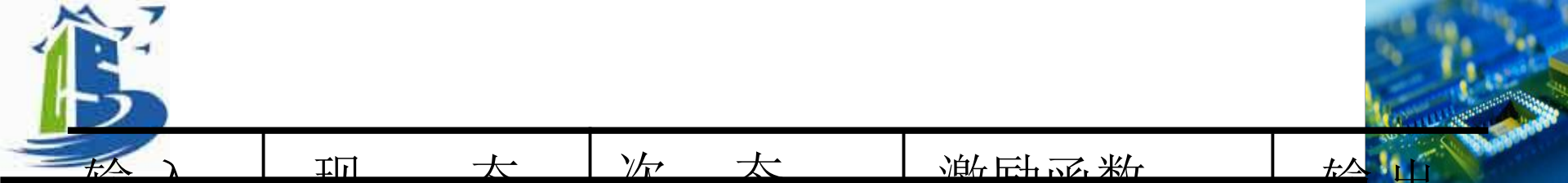
|       |   | $xy_2$ |    |    |    |
|-------|---|--------|----|----|----|
|       |   | 00     | 01 | 11 | 10 |
| $y_1$ | 0 | 0      | 0  | 0  | 0  |
|       | 1 | 0      | d  | d  | 1  |

|       |   | $xy_2$ |    |    |    |
|-------|---|--------|----|----|----|
|       |   | 00     | 01 | 11 | 10 |
| $y_1$ | 0 | d      | 1  | 0  | d  |
|       | 1 | d      | 1  | 0  | d  |

|       |   | $xy_2$ |    |    |    |
|-------|---|--------|----|----|----|
|       |   | 00     | 01 | 11 | 10 |
| $y_1$ | 0 | 0      | 0  | 1  | 1  |
|       | 1 | d      | d  | d  | d  |

|       |   | $xy_2$ |    |    |    |
|-------|---|--------|----|----|----|
|       |   | 00     | 01 | 11 | 10 |
| $y_1$ | 0 | d      | 1  | 0  | d  |
|       | 1 | 1      | 1  | 0  | 0  |

|       |   | $xy_2$ |    |    |    |
|-------|---|--------|----|----|----|
|       |   | 00     | 01 | 11 | 10 |
| $y_1$ | 0 | 0      | 0  | 1  | 0  |
|       | 1 | 0      | 0  | 1  | 0  |

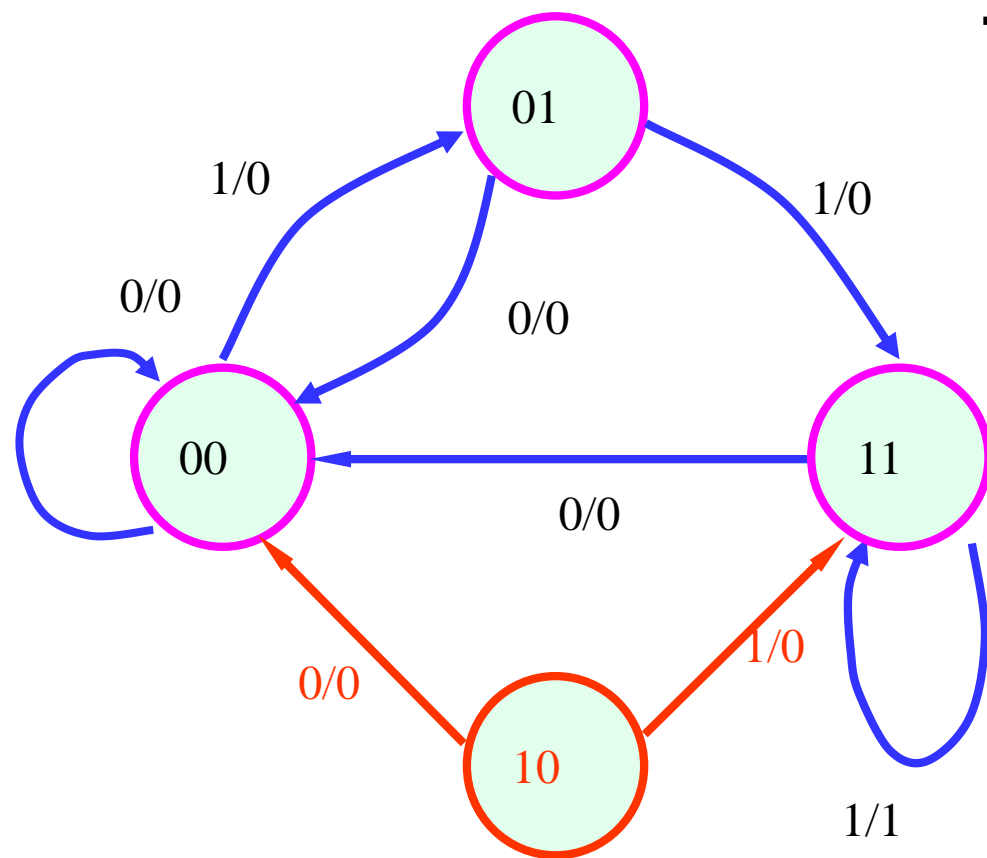


| 输入 | 现 态            |                | 次 态                             |                                 | 激励函数           |                |                |                | 输出 |
|----|----------------|----------------|---------------------------------|---------------------------------|----------------|----------------|----------------|----------------|----|
| X  | y <sub>2</sub> | y <sub>1</sub> | y <sub>2</sub> <sup>(n+1)</sup> | y <sub>1</sub> <sup>(n+1)</sup> | J <sub>2</sub> | K <sub>2</sub> | J <sub>1</sub> | K <sub>1</sub> | Z  |
| 0  | 0              | 0              | 0                               | 0                               | 0              | d              | 0              | d              | 0  |
| 0  | 0              | 1              | 0                               | 0                               | 0              | d              | d              | 1              | 0  |
| 0  | 1              | 1              | 0                               | 0                               | d              | 1              | d              | 1              | 0  |
| 0  | 1              | 0              | 0                               | 0                               | 0              | 1              | 0              | 1              | 0  |
| 1  | 0              | 0              | 0                               | 1                               | 0              | d              | 1              | d              | 0  |
| 1  | 0              | 1              | 1                               | 1                               | 1              | d              | d              | 0              | 0  |
| 1  | 1              | 1              | 1                               | 1                               | d              | 0              | d              | 0              | 1  |
| 1  | 1              | 0              | 1                               | 1                               | 0              | 0              | 1              | 0              | 1  |

|   |                | xy <sub>2</sub> |    |    |    |
|---|----------------|-----------------|----|----|----|
|   |                | 00              | 01 | 11 | 10 |
| Z | y <sub>1</sub> | 0               | d  | d  | 0  |
|   | 1              | 0               | 0  | 1  | 0  |

$$Z = xy_1y_2$$

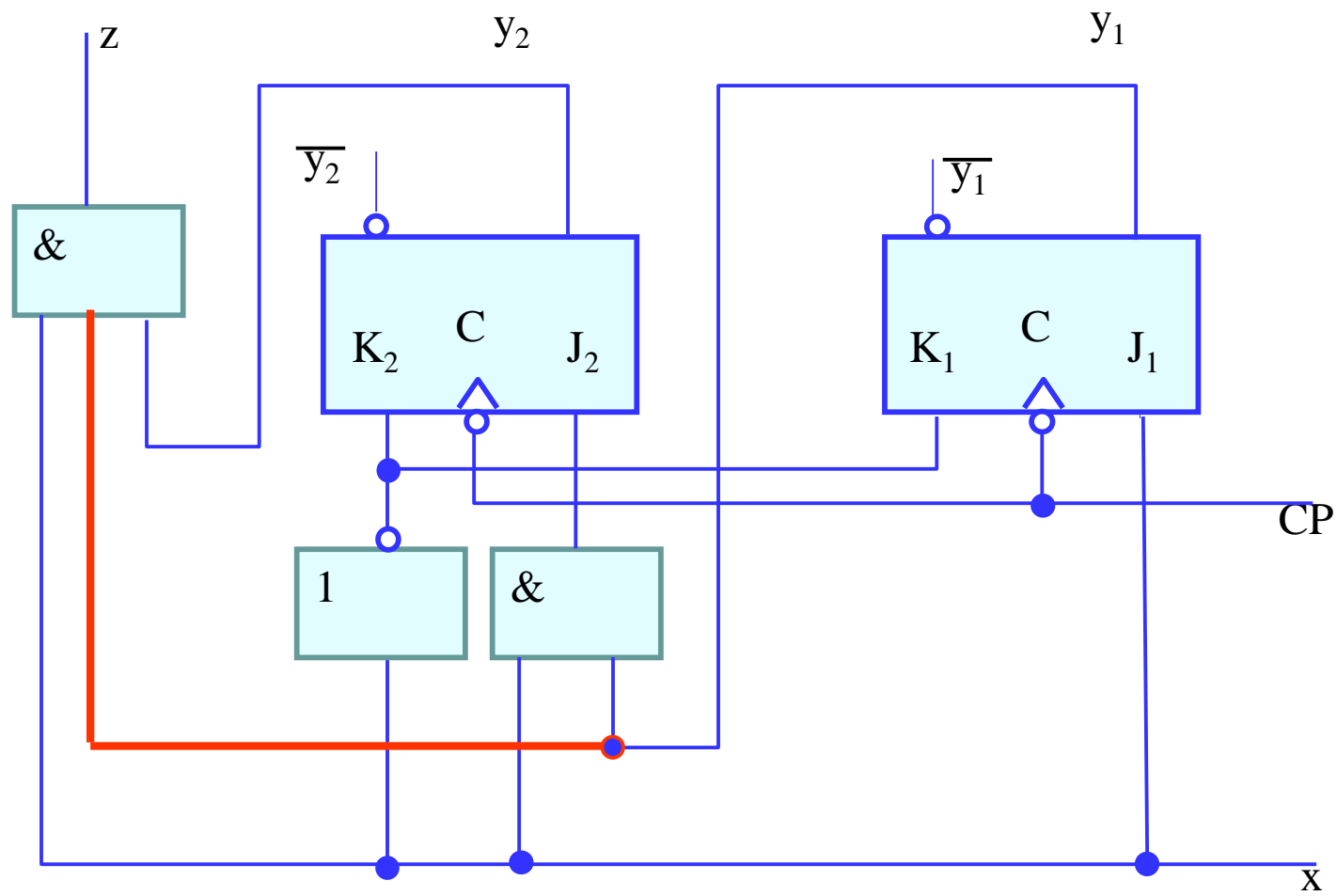




| 现<br>态 | 次态 $y_2^{(n+1)}y_1^{(n+1)}/Z$ |       |
|--------|-------------------------------|-------|
|        | $x=0$                         | $x=1$ |
|        | $y_2 \ y_1$                   |       |
| 0 0    | 00/0                          | 01/0  |
| 0 1    | 00/0                          | 11/0  |
| 1 1    | 00/0                          | 11/1  |
| 1 0    | 00/0                          | 11/0  |



- 修改后的电路图:

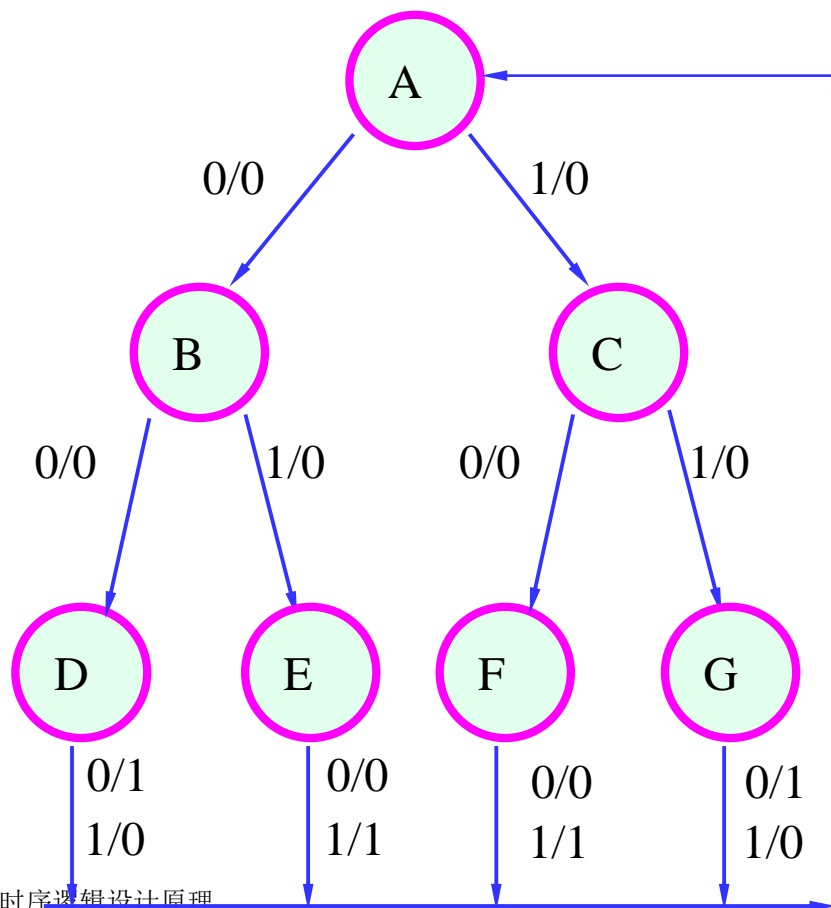




**例：**设计一个三位串行奇偶校验电路。当电路串行接收了三位二进制数后，如果1的个数为偶数，则电路输出为1；否则为0。当接收了三位二进制数后，电路返回初始状态。

**解：**

- 作状态图和状态表





## • 状态化简



| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | B/0   | C/0   |
| B  | D/0   | E/0   |
| C  | F/0   | G/0   |
| D  | A/1   | A/0   |
| E  | A/0   | A/1   |
| F  | A/0   | A/1   |
| G  | A/1   | A/0   |

| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | B/0   | C/0   |
| B  | D/0   | E/0   |
| C  | E/0   | D/0   |
| D  | A/1   | A/0   |
| E  | A/0   | A/1   |



- 状态分配:

| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | B/0   | C/0   |
| B  | D/0   | E/0   |
| C  | E/0   | D/0   |
| D  | A/1   | A/0   |
| E  | A/0   | A/1   |

- DE应相邻

- BC, DE应相邻

- ABC应两两相邻

- A应为逻辑0

| $y_3y_2$ |   | 00 | 01 | 11 | 10 |
|----------|---|----|----|----|----|
| $y_1$    | 0 | A  | B  | C  | D  |
|          | 1 |    |    |    | E  |



| $y_3y_2$ |   | 00 | 01 | 11 | 10 |
|----------|---|----|----|----|----|
| $v_1$    | 0 | A  | B  | C  | D  |
|          | 1 |    |    |    | E  |

| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | B/0   | C/0   |
| B  | D/0   | E/0   |
| C  | E/0   | D/0   |
| D  | A/1   | A/0   |
| E  | A/0   | A/1   |

| $Y_3Y_2Y_1$ | 次态/输出 |       |
|-------------|-------|-------|
|             | $x=0$ | $x=1$ |
| 000         | 010/0 | 110/0 |
| 010         | 100/0 | 101/0 |
| 110         | 101/0 | 100/0 |
| 100         | 000/1 | 000/0 |
| 101         | 000/0 | 000/1 |



| $Y_3 Y_2 Y_1$ | 次态/输出 |       |
|---------------|-------|-------|
|               | $x=0$ | $x=1$ |
|               |       |       |
| 000           | 010/0 | 110/0 |
| 001           | ddd/d | ddd/d |
| 010           | 100/0 | 101/0 |
| 011           | ddd/d | ddd/d |
| 100           | 000/1 | 000/0 |
| 101           | 000/0 | 000/1 |
| 110           | 101/0 | 100/0 |
| 111           | ddd/d | ddd/d |

| $Y_3Y_2Y_1$ | 次态/输出 |       |
|-------------|-------|-------|
|             | $x=0$ | $x=1$ |
| 000         | 010/0 | 110/0 |
| 001         | ddd/d | ddd/d |
| 010         | 100/0 | 101/0 |
| 011         | ddd/d | ddd/d |
| 100         | 000/1 | 000/0 |
| 101         | 000/0 | 000/1 |
| 110         | 101/0 | 100/0 |
| 111         | ddd/d | ddd/d |

$D_3$

| $y_2y_1 \backslash xy_3$ | 00 | 01 | 11 | 10 |
|--------------------------|----|----|----|----|
| 00                       | 0  | 0  | 0  | 1  |
| 01                       | d  | 0  | 0  | d  |
| 11                       | d  | d  | d  | d  |
| 10                       | 1  | 1  | 1  | 1  |

$D_2$

| $y_2y_1 \backslash xy_3$ | 00 | 01 | 11 | 10 |
|--------------------------|----|----|----|----|
| 00                       | 1  | 0  | 0  | 1  |
| 01                       | d  | 0  | 0  | d  |
| 11                       | d  | d  | d  | d  |
| 10                       | 0  | 0  | 0  | 0  |

$D_1$

| $y_2y_1 \backslash xy_3$ | 00 | 01 | 11 | 10 |
|--------------------------|----|----|----|----|
| 00                       | 0  | 0  | 0  | 0  |
| 01                       | d  | 0  | 0  | d  |
| 11                       | d  | d  | d  | d  |
| 10                       | 0  | 1  | 0  | 1  |

$Z$

| $y_2y_1 \backslash xy_3$ | 00 | 01 | 11 | 10 |
|--------------------------|----|----|----|----|
| 00                       | 0  | 1  | 0  | 0  |
| 01                       | d  | 0  | 1  | d  |
| 11                       | d  | d  | d  | d  |
| 10                       | 0  | 0  | 0  | 0  |





- 列出激励函数和输出函数表达式：

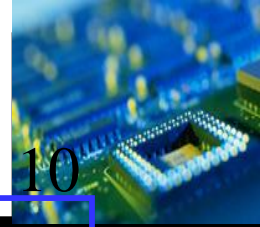
$$D_3 = Y_2 + X\bar{Y}_3$$

$$D_2 = \bar{Y}_3\bar{Y}_2$$

$$\begin{aligned} D_1 &= \bar{X}Y_3Y_2 + X\bar{Y}_3Y_2 \\ &= Y_2(X \oplus Y_3) \end{aligned}$$

$$Z = XY_1 + \bar{X}Y_3\bar{Y}_2\bar{Y}_1$$

- 画逻辑电路图：（略）。



Analysis:

| $y_2y_1 \backslash xy_3$ | 00 | 01 | 11 | 10 |
|--------------------------|----|----|----|----|
| 00                       | 0  | 0  | 0  | 1  |
| 01                       | 0  | 0  | 0  | 1  |
| 11                       | 1  | 1  | 1  | 1  |
| 10                       | 1  | 1  | 1  | 1  |

 $D_3$ 

| $y_2y_1 \backslash xy_3$ | 00 | 01 | 11 | 10 |
|--------------------------|----|----|----|----|
| 00                       | 1  | 0  | 0  | 1  |
| 01                       | 1  | 0  | 0  | 1  |
| 11                       | 0  | 0  | 0  | 0  |
| 10                       | 0  | 0  | 0  | 0  |

 $D_2$ 

| $y_2y_1 \backslash xy_3$ | 00 | 01 | 11 | 10 |
|--------------------------|----|----|----|----|
| 00                       | 0  | 0  | 0  | 0  |
| 01                       | 0  | 0  | 0  | 0  |
| 11                       | 0  | 1  | 0  | 1  |
| 10                       | 0  | 1  | 0  | 1  |

 $D_1$ 

| $y_2y_1 \backslash xy_3$ | 00 | 01 | 11 | 10 |
|--------------------------|----|----|----|----|
| 00                       | 0  | 1  | 0  | 0  |
| 01                       | 0  | 0  | 1  | 1  |
| 11                       | 0  | 0  | 1  | 1  |
| 10                       | 0  | 0  | 0  | 0  |

 $Z$



| $Y_3 Y_2 Y_1$ | 次态/输出 |       |
|---------------|-------|-------|
|               | $x=0$ | $x=1$ |
| 000           | 010/0 | 110/0 |
| 001           | 010/0 | 110/1 |
| 010           | 100/0 | 101/0 |
| 011           | 100/0 | 101/1 |
| 100           | 000/1 | 000/0 |
| 101           | 000/0 | 000/1 |
| 110           | 101/0 | 100/0 |
| 111           | 101/0 | 100/1 |



| $y_2y_1 \backslash xy_3$ |   | 00 | 01 | 11 | 10 |
|--------------------------|---|----|----|----|----|
| 00                       | 0 | 1  | 0  | 0  |    |
| 01                       | 0 | 0  | 1  | 0  |    |
| 11                       | 0 | 0  | 0  | 0  |    |
| 10                       | 0 | 0  | 0  | 0  |    |

$Z$

$$Z = \bar{X}Y_3\bar{Y}_2\bar{Y}_1 + XY_3\bar{Y}_2Y_1$$

修改后的画逻辑电路图：（略）。



- 1) 原始状态转移图
- 2) 状态化简
- 3) 状态编码
- 4) 状态方程，激励方程和输出方程
- 5) 画时序逻辑电路图