



## 第五章

# 同步时序逻辑电路





## 本章知识要点：

- ▶ 时序逻辑电路的基本概念；
- ▶ 同步时序逻辑电路的分析和设计方法；
- ▶ 典型同步时序逻辑电路的分析和设计。





## 5.1 概述

### 5.1.1 时序逻辑电路的定义、结构和特点

#### 一、定义

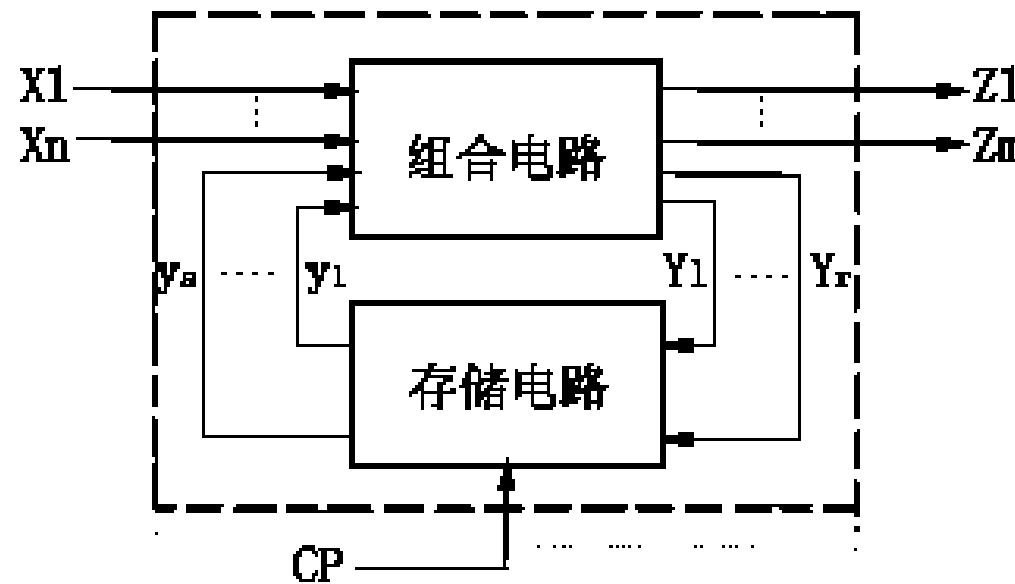
若逻辑电路在任何时刻产生的稳定输出信号不仅与电路该时刻的输入信号有关，还与电路过去的输入信号有关，则称为**时序逻辑电路**。





## 二、结构

时序逻辑电路由组合电路和存储电路两部分组成，通过反馈回路将两部分连成一个整体。



图中， $CP$ 为时钟脉冲信号，它是否存在取决于时序逻辑电路的类型。





时序逻辑电路的状态 $y_1, \dots, y_s$ 是存储电路对过去输入信号记忆的结果，它随着外部信号的作用而变化。

### 次态与现态的概念：

在对电路功能进行研究时，通常将某一时刻的状态称为“**现态**”，记作 $y^n$ ，简记为 $y$ ；

将在某一现态下，外部信号发生变化后到达的新的状态称为“**次态**”，记作 $y^{n+1}$ 。





### 三、特点

- ☆ 电路由组合电路和存储电路组成，具有对过去输入进行记忆的功能；
- ☆ 电路中包含反馈回路，通过反馈使电路功能与“时序”相关；
- ☆ 电路的输出由电路当时的输入和状态(对过去输入的记忆)共同决定。





## 5.1.2 时序逻辑电路的分类

### 一、按电路的工作方式分类

按照电路的工作方式，时序逻辑电路可分为同步时序逻辑电路和异步时序逻辑电路两种类型。

#### 1. 同步时序电路

**(1) 特点：**电路中有统一的定时信号，存储器件采用时钟控制触发器，电路状态在时钟脉冲控制下同时发生转换，即电路状态的改变依赖于输入信号和时钟脉冲信号。





## (2) 现态与次态

同步时序电路中的现态与次态是针对某个时钟脉冲而言的。

**现态**----指时钟脉冲作用之前电路所处的状态。

**次态**----指时钟脉冲作用之后电路到达的状态。

**注意：**前一个脉冲的次态即后一个脉冲的现态！

## (3) 对时钟的要求

脉冲的宽度：必须保证触发器可靠翻转；

脉冲的频率：必须保证前一个脉冲引起的电路响应完全结束后，后一个脉冲才能到来。

## 2. 异步时序逻辑电路

异步时序逻辑电路的存储电路可由触发器或延时元件组成，  
电路中**没有统一**的时钟信号同步，电路输入信号的变化将直接  
导致电路状态的变化。





## 二、按电路输出对输入的依从关系分类

根据电路的输出是否与输入直接相关，时序逻辑电路可以分为Mealy型和Moore型两种不同的模型。

1. **Mealy型电路**：若时序逻辑电路的输出是电路输入和电路状态的函数，则称为Mealy型时序逻辑电路。

2. **Moore型电路**：若时序逻辑电路的输出仅仅是电路状态的函数，则称为Moore型时序逻辑电路。

## 三、按输入信号形式分类

时序逻辑电路的输入信号可以是脉冲信号也可以是电平信号。根据输入信号形式的不同，时序逻辑电路通常又可分为脉冲型和电平型两种类型。





## 5.1.3 同步时序逻辑电路的描述方法

### 一、逻辑函数表达式

同步时序电路的结构和功能，可用三组逻辑函数表达式描述。

**1. 输出函数表达式：**是一组反映电路输出Z与输入x和状态y之间关系的表达式。

$$Z_i = f_i(x_1, \dots, x_n, y_1, \dots, y_s) \quad i=1,2,\dots,m \quad (\text{Mealy型电路})$$

$$Z_i = f_i(y_1, \dots, y_s) \quad i=1,2,\dots,m \quad (\text{Moore型电路})$$





**2. 激励函数表达式：**激励函数又称为控制函数，它反映了存储电路的输入Y与外部输入x和电路状态y之间的关系。其函数表达式为

$$Y_j = g_j(x_1, \dots, x_n, y_1, \dots, y_s) \quad j=1,2,\dots,r$$

**3. 次态函数表达式：**次态函数用来反映同步时序电路的次态 $y^{(n+1)}$ 与激励函数Y和电路现态y之间的关系，它与触发器类型相关。其函数表达式为

$$y_l^{n+1} = k_l(Y_j, y_l) \quad j=1,2,\dots,r ; \quad l=1,2,\dots,s$$





## 二、状态表

**状态表：**反映同步时序电路输出Z、次态 $y^{n+1}$ 与电路输入x、现态y之间关系的表格，又称为**状态转移表**。

Mealy型同步时序电路状态表的格式如作下表所示。

Mealy型电路状态表格式

| 现态 | 次态 / 输出 |               |
|----|---------|---------------|
|    | 输入 x    |               |
| y  |         | $y^{(n+1)}/Z$ |
|    |         |               |

表中，列数 = 输入的所有取值组合数；  
行数 = 触发器的状态组合数。





Moore型电路状态表的格式如左下表所示。

Moore型电路状态表格式

| 现 态 | 次态          |  | 输出  |
|-----|-------------|--|-----|
|     | 输入 $x$      |  |     |
| $y$ | $y^{(n+1)}$ |  | $z$ |
|     |             |  |     |
|     |             |  |     |

状态表是同步时序电路分析和设计中常用的工具，它非常清晰地给出了同步时序电路在不同输入和现态下的次态和输出。

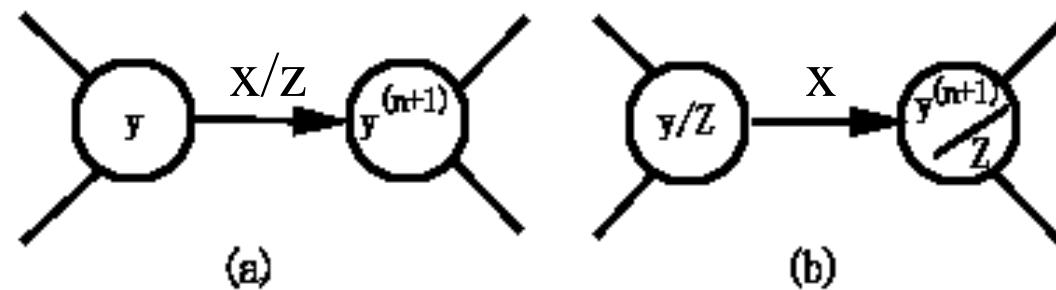




### 三、状态图

**状态图：**是一种反映同步时序电路状态转换规律及相应输入、输出取值关系的有向图。

Mealy型电路状态图的形式如图(a)所示。图中，在有向箭头的旁边标出发生该转换的输入条件以及在该输入和现态下的相应输出。



Moore型电路状态图的形式如图(b) 所示，电路输出标在圆圈内的状态右下方，表示输出只与状态相关。





## 四、时间图

时间图是用波形图的形式来表示输入信号、输出信号和电路状态等的取值在各时刻的对应关系，通常又称为工作波形图。在时间图上，可以把电路状态转换的时刻形象地表示出来。





## 5.2 同步时序逻辑电路分析

### 5.2.1 分析的方法和步骤

常用方法有表格法和代数法。

#### 一、表格分析法的一般步骤

1. 写出输出函数和激励函数表达式。  
↓
2. 借助触发器功能表列出电路次态真值表。  
↓
3. 作出状态表和状态图（必要时画出时间图）。  
↓
4. 归纳出电路的逻辑功能。





## 二、代数分析法的一般步骤

1. 写出输出函数和激励函数表达式。



2. 把激励函数表达式代入触发器的次态方程，导出电路的次态方程组。



3. 作出状态表和状态图（必要画出时间图）。



4. 归纳出电路的逻辑功能。

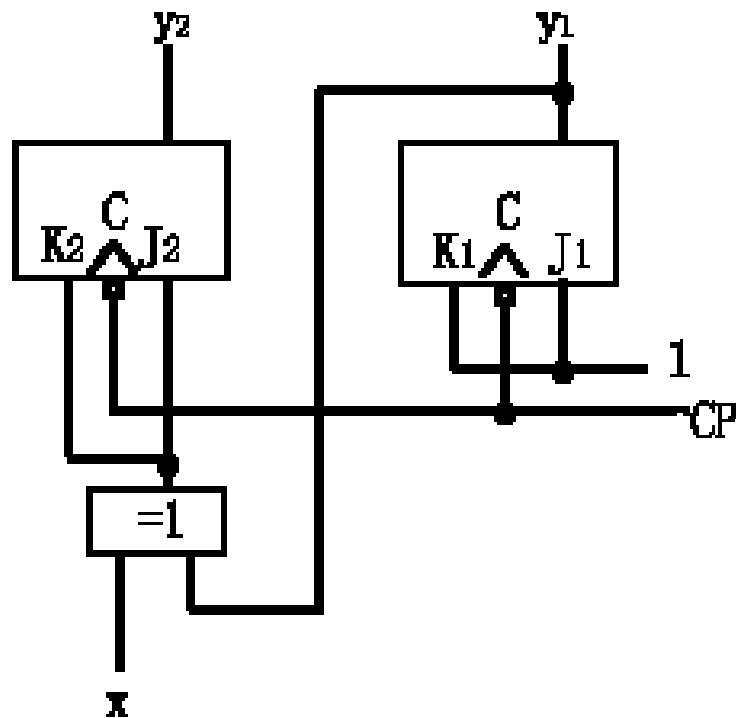
由分析步骤可知，两种方法仅第二步有所不同，分析中可视具体问题灵活选用。





## 5.2.2 分析举例

例 用表格法分析下图所示同步时序逻辑电路。



解 该电路的输出即状态变量，因此，该电路属于Moore型电路的特例。

1.写出输出函数和激励函数表达式

$$J_1 = K_1 = 1 \quad ; \quad J_2 = K_2 = x \oplus y_1$$





## 2. 列出电路次态真值表

次态真值表

| 输入<br>x | 现<br>态<br>$y_2$ | y <sub>1</sub> | 激励函数           |                |                |                | $y_2^{(n+1)}$ | $y_1^{(n+1)}$ |
|---------|-----------------|----------------|----------------|----------------|----------------|----------------|---------------|---------------|
|         |                 |                | J <sub>2</sub> | K <sub>2</sub> | J <sub>1</sub> | K <sub>1</sub> |               |               |
| 0       | 0               | 0              | 0              | 0              | 1              | 1              | 0             | 1             |
| 0       | 0               | 1              | 1              | 1              | 1              | 1              | 1             | 0             |
| 0       | 1               | 0              | 0              | 0              | 1              | 1              | 1             | 1             |
| 0       | 1               | 1              | 1              | 1              | 1              | 1              | 0             | 0             |
| 1       | 0               | 0              | 1              | 1              | 1              | 1              | 1             | 1             |
| 1       | 0               | 1              | 0              | 0              | 1              | 1              | 0             | 0             |
| 1       | 1               | 0              | 1              | 1              | 1              | 1              | 0             | 1             |
| 1       | 1               | 1              | 0              | 0              | 1              | 1              | 1             | 0             |

| J | K | $Q^{(n+1)}$    |
|---|---|----------------|
| 0 | 0 | Q              |
| 0 | 1 | 0              |
| 1 | 0 | 1              |
| 1 | 1 | $\overline{Q}$ |

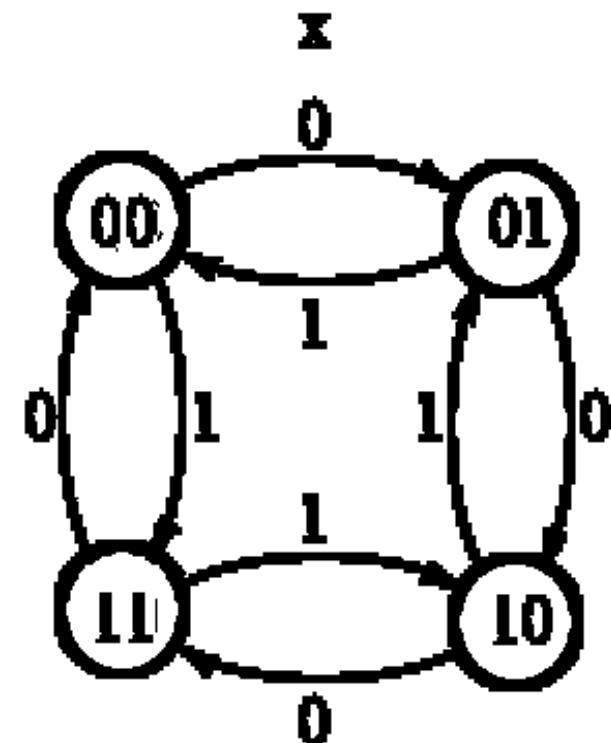




### 3. 作出状态表和状态图

状态表

| 现态<br>$y_2$ $y_1$ | 次态 $y_2^{(n+1)}y_1^{(n+1)}$ |     |
|-------------------|-----------------------------|-----|
|                   | X=0                         | X=1 |
| 0 0               | 0 1                         | 1 1 |
| 0 1               | 1 0                         | 0 0 |
| 1 0               | 1 1                         | 0 1 |
| 1 1               | 0 0                         | 1 0 |





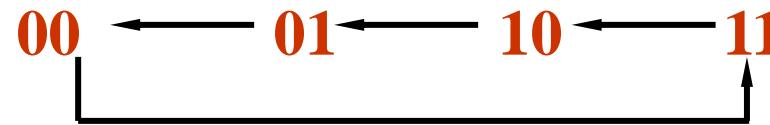
#### 4. 描述电路的逻辑功能。

由状态图可知，该电路是一个**2位二进制数可逆计数器**。

当输入 $x=0$ 时，可逆计数器进行加1计数，其计数序列为

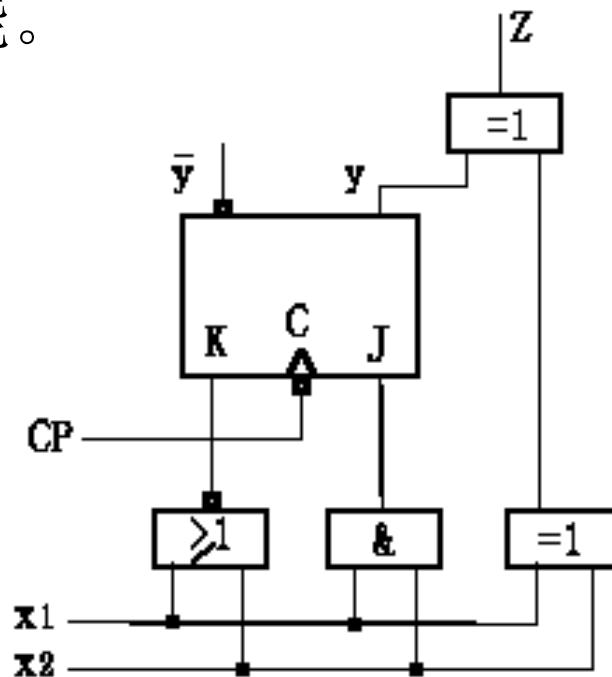


当输入 $x=1$ 时，可逆计数器进行减1计数，其计数序列为





例 试用代数法分析下图所示同步时序逻辑电路的逻辑功能。



解 该电路由一个J-K触发器和四个逻辑门构成，电路有两个输入端 $x_1$ 和 $x_2$ ，一个输出端Z。输出Z与输入和状态均有直接联系，属于Mealy型电路。

## 1. 写出输出函数和激励函数表达式

$$Z = x_1 \oplus x_2 \oplus y$$

$$J = x_1 x_2 \quad K = \overline{x_1 + x_2}$$





## 2. 把激励函数表达式代入触发器的次态方程，得到电路的次态方程组

该电路的存储电路只有一个触发器，因此，电路只有一个次态方程。

根据J-K触发器的次态方程和电路的激励函数表达式，可导出电路的次态方程如下：

$$\begin{aligned}y^{(n+1)} &= J\bar{y} + \bar{K}y \\&= x_1 x_2 \bar{y} + (\overline{x_1 + x_2})y \\&= x_1 x_2 \bar{y} + x_1 y + x_2 y \\&= x_1 x_2 + x_1 y + x_2 y\end{aligned}$$





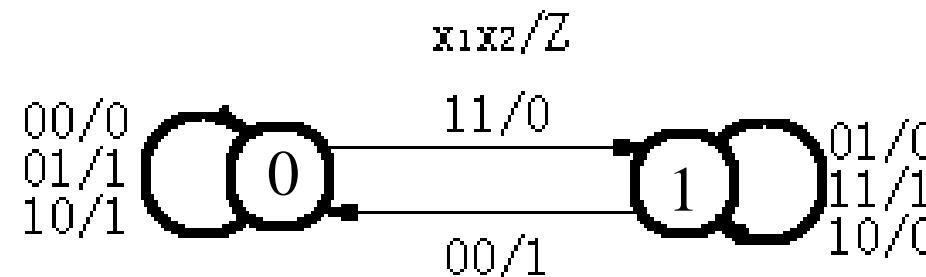
### 3. 根据次态方程和输出函数表达式作出状态表和状态图

根据次态方程和输出函数表达式，可以作出该电路的状态表和状态图如下。

$$\begin{aligned}y^{n+1} &= x_1 x_2 + x_1 y + x_2 y \\Z &= x_1 \oplus x_2 \oplus y\end{aligned}$$

状态表

| 现态<br>$y$ | 次态 / 输出 ( $y^{(n+1)}/Z$ ) |             |             |             |
|-----------|---------------------------|-------------|-------------|-------------|
|           | $x_1x_2=00$               | $x_1x_2=01$ | $x_1x_2=11$ | $x_1x_2=10$ |
| 0         | 0/0                       | 0/1         | 1/0         | 0/1         |
| 1         | 0/1                       | 1/0         | 1/1         | 1/0         |

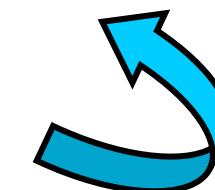
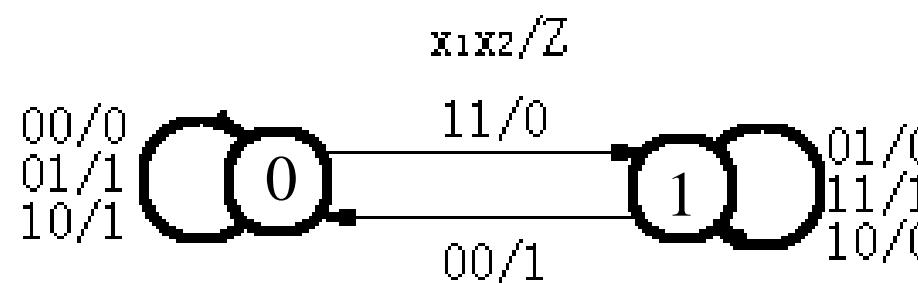
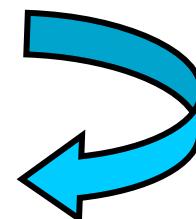




#### 4. 画出时间图，并说明电路的逻辑功能

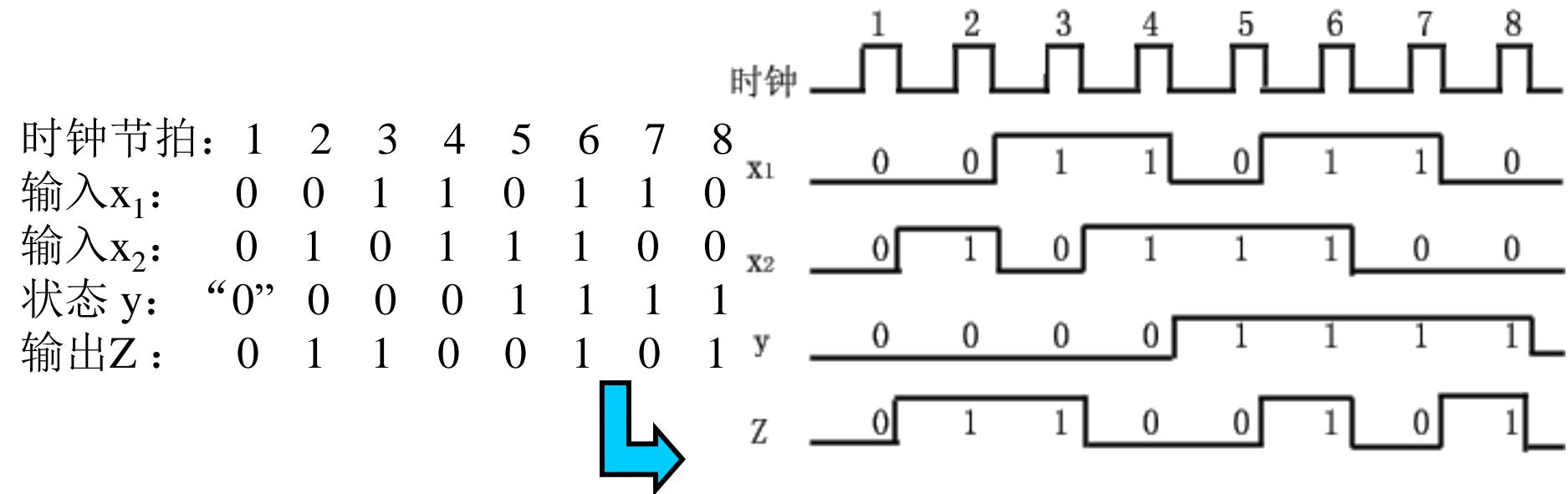
设电路初态为“0”，输入 $x_1$ 为**00110110**，输入 $x_2$ 为**01011100**，根据状态图可作出电路的输出和状态响应序列如下：

|            |     |   |   |   |   |   |   |   |
|------------|-----|---|---|---|---|---|---|---|
| 时钟节拍:      | 1   | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 输入 $x_1$ : | 0   | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 输入 $x_2$ : | 0   | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 状态y:       | “0” | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 输出Z:       | 0   | 1 | 1 | 0 | 0 | 1 | 0 | 1 |





根据状态响应序列可作出时间图如下：



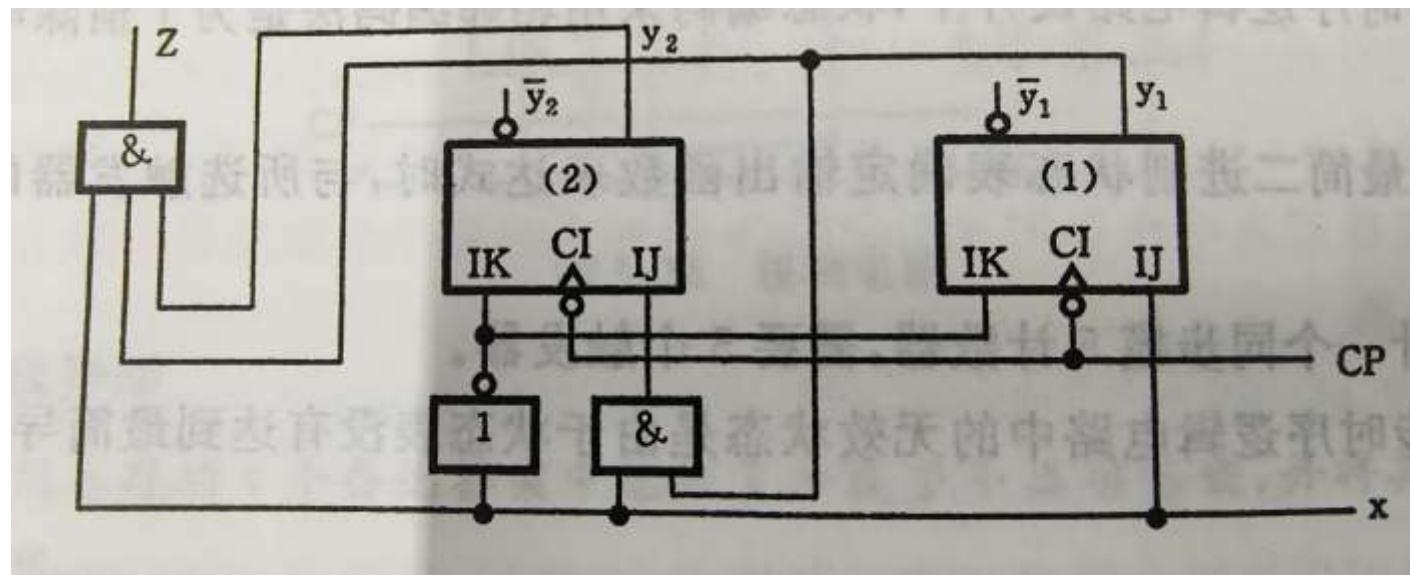
分析时间图可知，该电路实现了**串行加法器的功能**。其中 $x_1$ 为被加数， $x_2$ 为加数，它们按照先低位后高位的顺序串行地输入。每位相加产生的进位由触发器保存下来参加下一位相加，输出Z从低位到高位串行地输出“和”数。





## 练习题5-3：

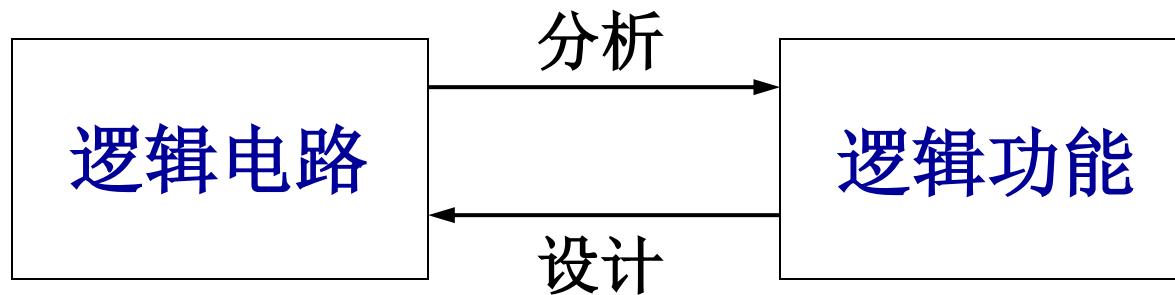
分析下图所示逻辑电路，设电路的初始状态为“00”，输入序列  $x=10011110110$ ，作出输出响应序列，并说明电路功能。





## 5.3 同步时序逻辑电路的设计

同步时序逻辑电路的设计是指根据特定的逻辑要求，设计出能实现其逻辑功能的时序逻辑电路。显然，设计是分析的逆过程，即：



同步时序逻辑电路设计追求的目标是，使用尽可能少的触发器和逻辑门实现预定的逻辑要求！





设计的一般步骤如下：

1. 形成原始状态图和原始状态表；  
↓
2. 状态化简，求得最小化状态表；  
(完全确定状态和不确定状态表的化简)  
↓
3. 状态编码，得到二进制状态表；  
↓
4. 选定触发器的类型，并求出激励函数和输出函数最简表达式；  
↓
5. 画出逻辑电路图。





### 5.3.1 建立原始状态图和原始状态表

原始状态图和原始状态表是对设计要求的最原始的抽象。建立正确的原始状态图和状态表是同步时序电路设计中最关键的一步。

形成原始状态图时一般应考虑如下几个方面问题：

#### 一、 确定电路模型

设计成Mealy型？ Moore型？

#### 二、 设立初始状态

时序逻辑电路在输入信号开始作用之前的状态称为初始状态。





### 三、根据需要记忆的信息增加新的状态

同步时序电路中状态数目的多少取决于需要记忆和区分的信息量。

### 四、确定各时刻电路的输出

在建立原始状态图时，必须确定各时刻的输出值。在Moore型电路中，应指明每种状态下对应的输出；在Mealy型电路中应指明从每一个状态出发，在不同输入作用下的输出值。





**例** 某序列检测器有一个输入端x和一个输出端Z。输入端x输入一串随机的二进制代码，当输入序列中出现“011”时，输出Z产生一个1输出，平时Z输出0。典型输入、输出序列如下。

输入x: 1 0 1 0 1 1 1 0 0 1 1 0  
输出Z: 0 0 0 0 0 1 0 0 0 0 1 0

试作出该序列检测器的**原始状态图**和**原始状态表**。

**解 1.** 假定用Mealy型同步时序逻辑电路实现该序列检测器的逻辑功能。

设：

状态A-----电路的初始状态；

状态B-----表示收到了序列“011”中的第一个信号“0”；

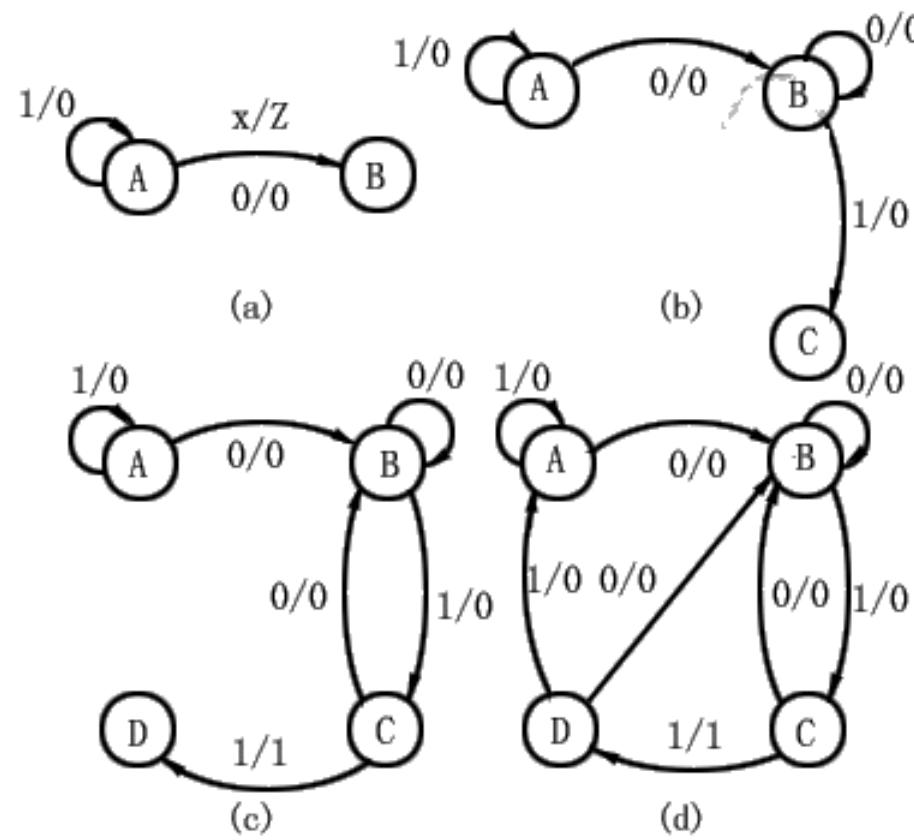
状态C-----表示收到了序列“011”中的前面两位“01”；

状态D-----表示收到了序列“011”。



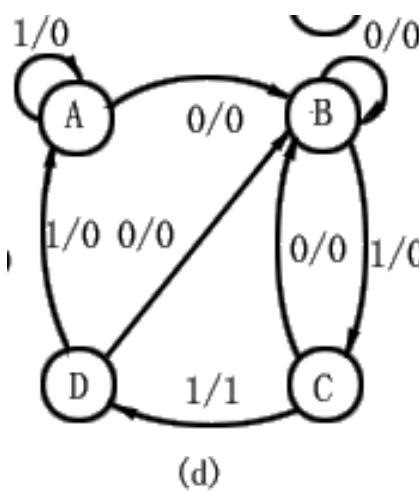


可画出状态图如下：





相应状态表如下：



Mealy型状态表

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





## 2. 假定用Moore型同步时序逻辑电路实现该序列检测器的逻辑功能。

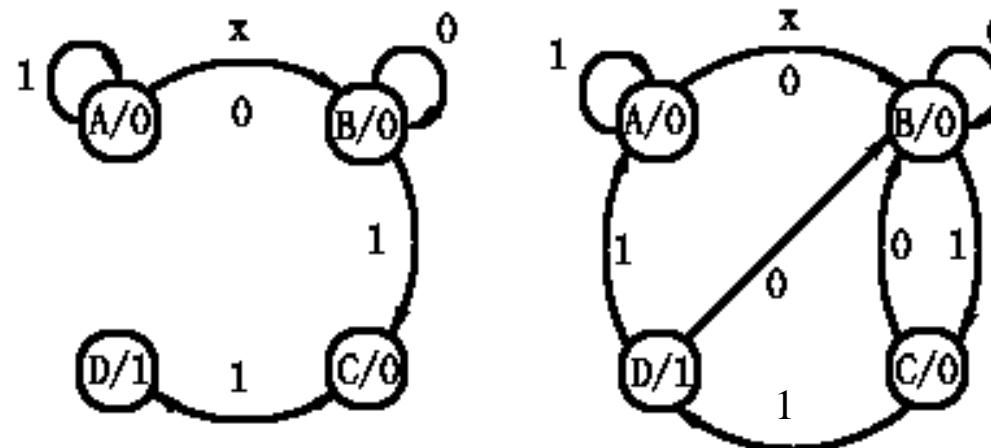
由于电路输出完全取决于状态，而与输入无直接联系。在作状态图时，应将输出标记在代表各状态的圆圈内。

设电路初始状态为A，并用状态B、C、D分别表示收到了输入x送来的0、01、011。显然，根据题意，仅当处于状态D时电路输出为1，其他状态下输出均为0。





构造Moore型原始状态图如下：



相应的原始状态表如下表所示。

Moore型状态表

| 现 态              | 次 态              |                  | 输出<br>Z          |
|------------------|------------------|------------------|------------------|
|                  | x=0              | x=1              |                  |
| A<br>B<br>C<br>D | B<br>B<br>B<br>B | A<br>C<br>D<br>A | 0<br>0<br>0<br>1 |





### 5.3.2 状态化简

**什么叫状态化简？** 所谓状态化简，是指采用某种化简技术从原始状态表中**消去多余状态**，得到一个既能正确地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，通常称这种状态表为最小化状态表。

**目的：**简化电路结构。状态数目的多少直接决定电路中所需触发器数目的多少。设状态数目为n,所需触发器数目为m,则应满足如下关系：

$$2^m \geq n > 2^{m-1}$$

**方法：**常用方法有**观察法、输出分类法、隐含表法**等。下面讨论最常用的一种方法----**隐含表法**。





## 一、完全确定状态表的化简

### 1. 几个概念

#### (1) 等效状态

请注意掌握三点：定义、判断方法和性质。

##### ① 定义

设状态 $S_i$ 和 $S_j$ 是完全确定状态表中的两个状态，若对于所有可能的输入序列，分别从状态 $S_i$ 和状态 $S_j$ 出发，所得到的输出响应序列完全相同，则状态 $S_i$ 和 $S_j$ 是等效的，记作 $(S_i, S_j)$ ，又称状态 $S_i$ 和 $S_j$ 为等效对。





## ② 判断方法

若状态  $S_i$  和  $S_j$  是完全确定的原始状态表中的两个现态，则  $S_i$  和  $S_j$  等效的条件可归纳为在一位输入的各种取值组合下满足如下两条：

第一，输出相同；

第二，次态属于下列情况之一：

a. 次态相同；

b. 次态交错或为各自的现态；

c. 次态循环或为等效对。

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





### ③ 性质

等效状态具有**传递性**。即假若 $S_1$ 和 $S_2$ 等效， $S_2$ 和 $S_3$ 等效，那么，一定有 $S_1$ 和 $S_3$ 等效。记作

$$(S_1, S_2), (S_2, S_3) \longrightarrow (S_1, S_3)$$

### (2) 等效类

**等效类：**由若干彼此等效的状态构成的集合。在同一个等效类中的任意两个状态都是等效的。

例如，由 $(S_1, S_2)$ 和 $(S_2, S_3)$ 可以推出 $(S_1, S_3)$ ，进而可知 $S_1$ 、 $S_2$ 、 $S_3$ 属于同一等效类，记作 $\{ S_1, S_2, S_3 \}$ ，即

$$(S_1, S_2), (S_2, S_3) \longrightarrow \{ S_1, S_2, S_3 \}$$





### (3) 最大等效类

所谓最大等效类，是指不被任何别的等效类所包含的等效类。

换而言之，如果一个等效类不是任何其他等效类的子集，则该等效类称为最大等效类。

完全给定原始状态表的化简过程，就是寻找出表中的所有**最大等效类**，然后将每个最大等效类中的状态合并为一个新的状态，从而得到最小化状态表。

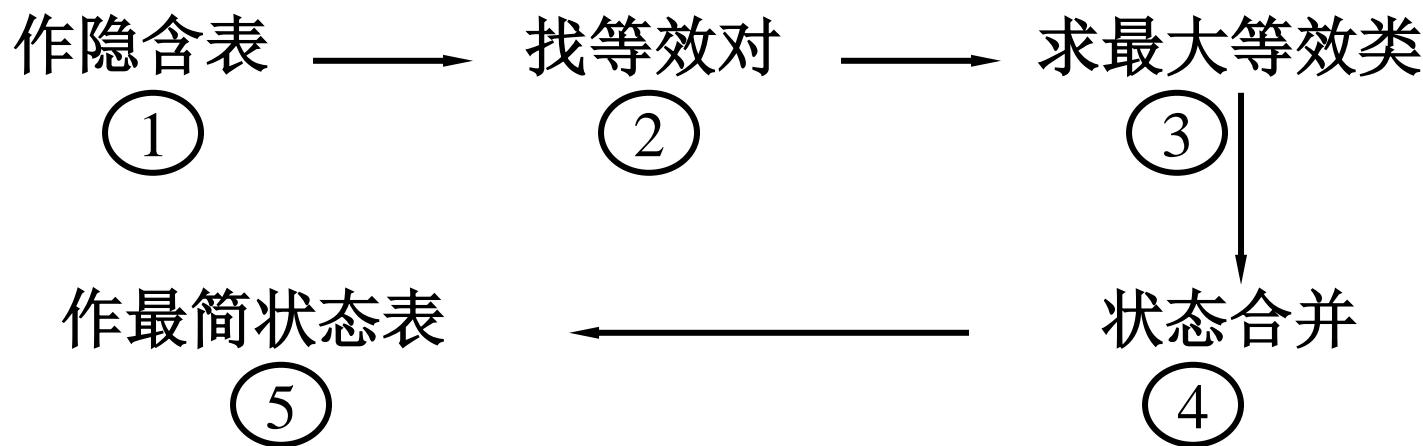
简化后的状态数等于最大等效类的个数！





## 2. 状态化简

### (1) 隐含表化简法的一般步骤



#### ① 作隐含表

隐含表是一个直角三角形阶梯网格，表中每个方格代表一个状态对。（状态的两两对应关系汇总）





## ②寻找等效对

**顺序比较：**按照隐含表中从上至下、从左至右的顺序，对照原始状态表依次对所有“状态对”进行逐一检查和比较，并将检查结果标注在隐含表中的相应方格内。

比较结果标注如下：

**等效** ----- 在相应方格内填上“√”；

**不等效**----- 在相应方格内填上“×”；

**与其他状态对相关** ---- 在相应方格内填上相关的状态对。

**关联比较：**指对那些在顺序比较时尚未确定是否等效的状态对作进一步检查。直到判别出状态对等效或不等效为止。





### ③求出最大等效类

在找出原始状态表中的所有等效对之后，可利用等效状态的传递性，求出各最大等效类。

### ④状态合并，作出最小化状态表

将每个最大等效类中的全部状态合并为一个状态，即可得到和原始状态表等价的最小化状态表。





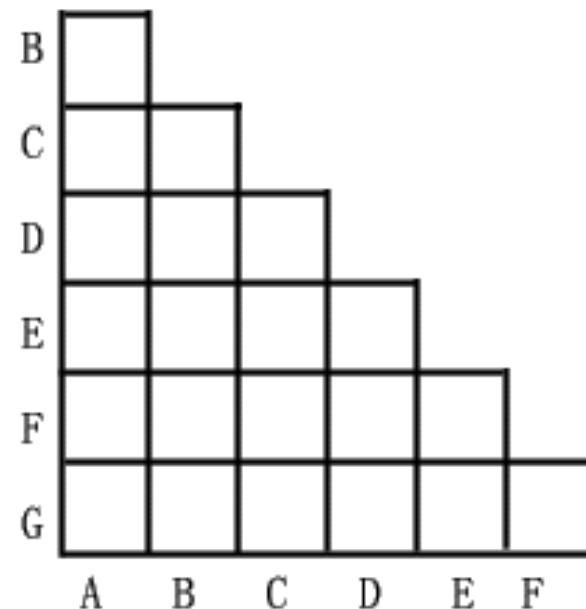
## (2)化简举例

例 化简下表所示原始状态表。

解 ① 作隐含表

给定原始状态表具有7个状态，根据画隐含表的规则，可画出隐含表框架如下。

| 现 态 | 次态 / 输出 |     |
|-----|---------|-----|
|     | x=0     | 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 |





## ② 寻找等效对

根据等效状态的判断标准，依次检查每个状态对，可得到顺序比较结果如图(a)所示。

关联比较的结果如图(b)所示。

|   |    |          |   |          |   |   |
|---|----|----------|---|----------|---|---|
| B | CF |          |   |          |   |   |
| C | X  | X        |   |          |   |   |
| D | X  | X        | X |          |   |   |
| E | BE | AE<br>CF | X | X        |   |   |
| F | X  | X        | ✓ | X        | X |   |
| G | X  | X        | X | CD<br>DE | X | X |
|   | A  | B        | C | D        | E | F |

图(a)

|   |    |          |   |          |   |   |
|---|----|----------|---|----------|---|---|
| B | CF |          |   |          |   |   |
| C | X  | X        |   |          |   |   |
| D | X  | X        | X |          |   |   |
| E | BE | AE<br>CF | X | X        |   |   |
| F | X  | X        | ✓ | X        | X |   |
| G | X  | X        | X | CD<br>DE | X | X |
|   | A  | B        | C | D        | E | F |

图(b)





|   |     |          |          |   |   |   |
|---|-----|----------|----------|---|---|---|
|   |     | CF       |          |   |   |   |
| B |     | X X      |          |   |   |   |
| C |     | X X      |          |   |   |   |
| D |     | X X      | X        |   |   |   |
| E | BE  | AE<br>CF | X X      |   |   |   |
| F | X X | ✓        | X        | X |   |   |
| G | X X | X        | CF<br>DE | X | X |   |
|   | A   | B        | C        | D | E | F |

由判断结果可知，原始状态表中的7个状态共有四个等效对：(A, B), (A, E), (B, E), (C, F)。





### ③求出最大等效类

由所得到的等效对和最大等效类的定义可知，原始状态表中的7个状态共构成 $\{A,B,E\}, \{C,F\}, \{D\}, \{G\}$ 四个最大等效类。

### ④状态合并，作出最小化状态表

令 $\{A,B,E\}---a$ 、 $\{C,F\}---b$ 、 $\{D\}---c$ 、 $\{G\}---d$ ，并代入原始状态表中，即可得到化简后的状态表如下边右表所示。

原始状态表

| 现态 | 次态 / 输出 |       |
|----|---------|-------|
|    | $x=0$   | $x=1$ |
| 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   |



最小化状态表

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





## 二、不完全确定状态表的化简

**例** 设计一个用于引爆控制的同步时序电路，该电路有一个输入端x和一个输出端Z。平时输入x始终为0，一旦需要引爆，则从x连续输入4个1信号(不被0中断)，电路收到第四个1后在输出端Z产生一个1信号点火引爆，该电路连同引爆装置一起被炸毁。试建立该电路的Mealy型状态图和状态表。

(该问题的实际意义？-----施工的安全性！)

**解 分析：**该电路实际上是一个用于特殊场所的“1111”序列检测器。它与一般序列检测器有两点不同：

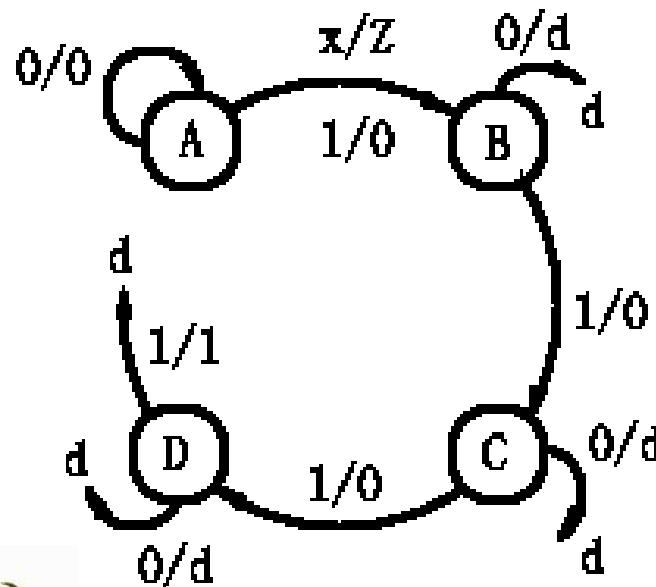
1. 输入带有约束条件，即一旦输入出现1，则一定是不被0中断的连续4个1；
2. 收到4个1后，输出产生的引爆信号使电路自毁，故此时不再存在次态问题。





设：**状态A**---电路初始状态；  
**状态B**---表示收到了第一个1输入；  
**状态C**---表示收到了连续2个1输入；  
**状态D**---表示收到了连续3个1输入。

根据题意，可得到该电路的Mealy型原始状态图和原始状态表如下。图、表中用“d”表示不确定次态或不确定输出。



原始状态表

| 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   |





## 如何化简具有不确定状态的状态表?

### 1. 几个概念

#### (1) 相容状态

① 定义：设状态  $S_i$  和  $S_j$  是不完全确定状态表中的两个状态，若对于所有可能的输入序列，分别从状态  $S_i$  和状态  $S_j$  出发，所得到的确定输出序列（不确定的位除外）完全相同，则状态  $S_i$  和  $S_j$  是相容的，记作  $(S_i, S_j)$ ，又称状态  $S_i$  和  $S_j$  为相容对。





## ② 判断方法

若状态  $S_i$  和  $S_j$  是不完全确定的原始状态表中的两个现态，则  $S_i$  和  $S_j$  相容的条件可归纳为在一位输入的各种取值组合下满足如下两条：

第一，输出相同，或其中一个（或两个）输出不确定；

第二，次态属于下列情况之一：

- a. 次态相同；
- b. 次态交错或为各自的现态；
- c. 次态循环或为相容对；
- d. 其中的一个（或两个）为不确定状态。





### ③ 性质

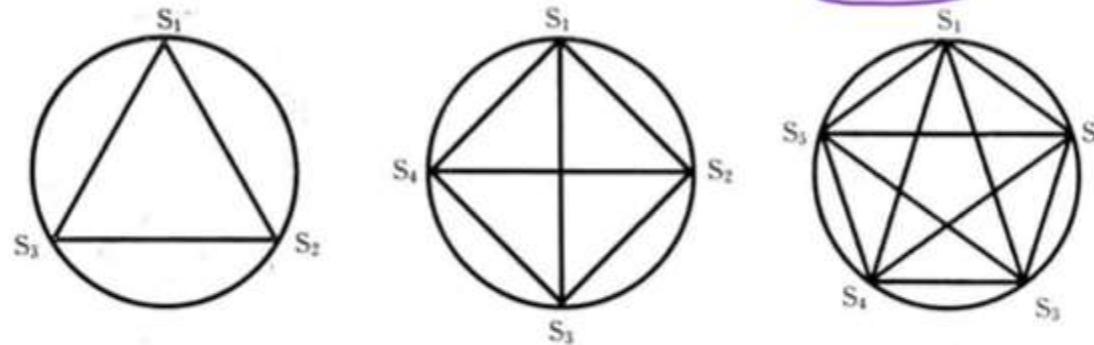
相容状态不具有传递性。即假若 $S_1$ 和 $S_2$ 相容， $S_2$ 和 $S_3$ 相容，那么，不能推出 $S_1$ 和 $S_3$ 相容。

#### (2) 相容类

**相容类：**由若干彼此相容的状态构成的集合。在同一个相容类中的状态都是**两两相容**的。

例如，若有相容对 $(S_1, S_2)$ 和 $(S_2, S_3)$ 和 $(S_1, S_3)$ ，进而可知 $S_1$ 、 $S_2$ 、 $S_3$ 属于同一相容类，记作 $\{S_1, S_2, S_3\}$ ，即

$$(S_1, S_2), (S_2, S_3), (S_1, S_3) \longrightarrow \{S_1, S_2, S_3\}$$



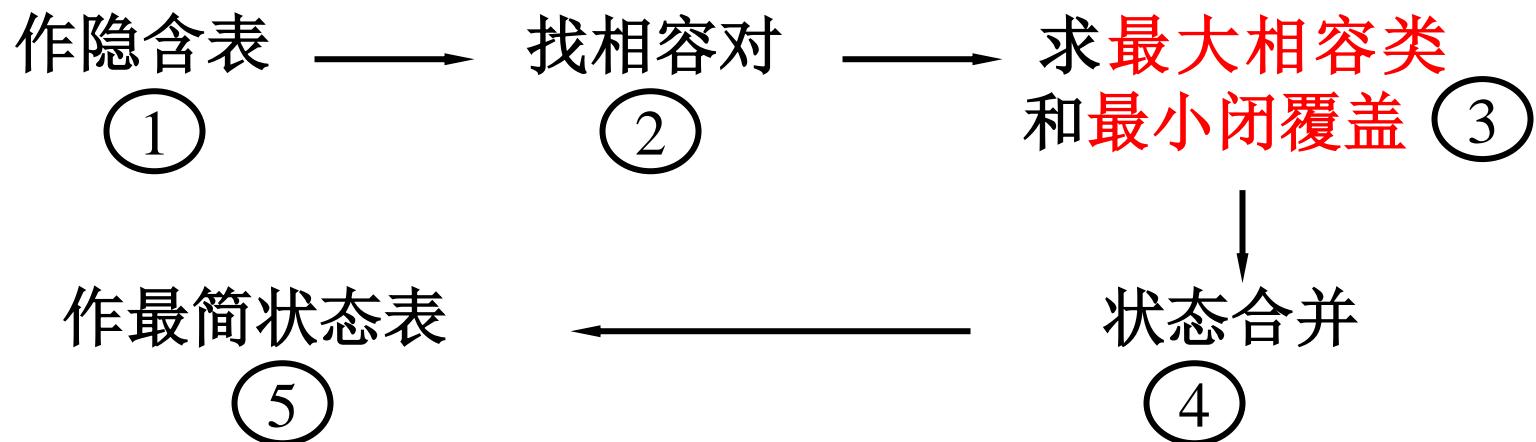
状态合并图





### (3) 最大相容类

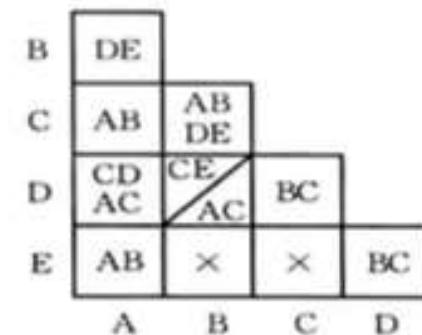
如果一个相容类不是任何其他相容类的子集，则该相容类称为最大相容类。由于相容状态不具备传递性，所以，同一状态可能出现在不同的最大相容类中。



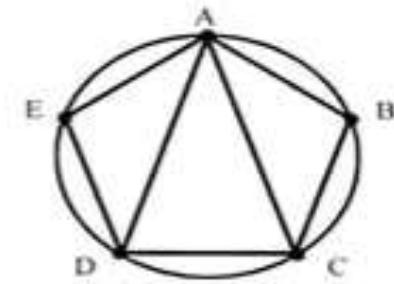


## 原始状态表

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



隐含表



状态合并图

根据状态合并图，可以获得最大相容类为：{A,B,C}, {A,C,D}, {A,D,E}





根据状态合并图，可以获得最大相容类为：  
 $\{A, B, C\}$ ,  $\{A, C, D\}$ ,  $\{A, D, E\}$

原始状态表

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

表 5.28 闭覆盖表

| 最大相容类 | 覆盖 |   |   |   |   | 闭合    |       |
|-------|----|---|---|---|---|-------|-------|
|       | A  | B | C | D | E | $x=0$ | $x=1$ |
| ABC   | A  | B | C |   |   | DE    | AB    |
| ACD   | A  |   | C | D |   | CD    | ABC   |
| ADE   | A  |   |   | D | E | CD    | ABC   |

表 5.29 最小化状态表

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

最小化状态表是最简的吗？

最小闭覆盖



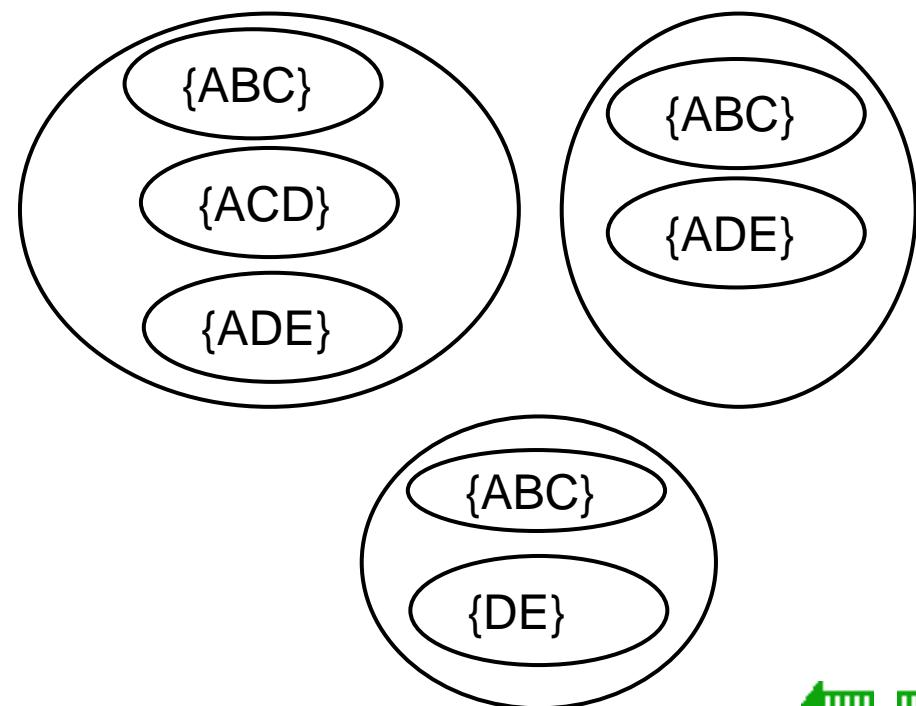


## 最小闭覆盖:

- a. 覆盖: 所选相容类集合应包含原始状态表的全部状态
- b. 最小: 所选相容类集合中相容类个数应最小
- c. 闭合: 所选相容类集合中任一相容类, 在原始状态表中任一输入条件下产生的次态组合应属于该集合中的某一个相容类。

原始状态表

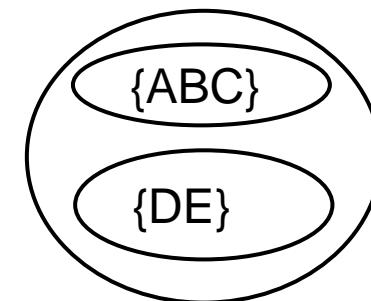
| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | D/d   | A/0   |
| B  | E/0   | A/d   |
| C  | D/0   | B/0   |
| D  | C/d   | C/d   |
| E  | d/1   | B/d   |





## 原始状态表

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



为最小闭覆盖

表 5.30 闭覆盖表

| 最大相容类 | 覆盖 |   |   |   |   | 闭合  |     |
|-------|----|---|---|---|---|-----|-----|
|       | A  | B | C | D | E | x=0 | x=1 |
| ABC   | A  | B | C |   | E | DE  | AB  |
| DE    |    |   |   | D |   | C   | BC  |

$$\begin{aligned}\{A, B, C\} &\rightarrow a \\ \{D, E\} &\rightarrow b\end{aligned}$$

表 5.31 最小化状态表

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





### 5.3.3 状态编码

**状态编码：**是指给最小化状态表中用字母或数字表示的状态，指定一个二进制代码，形成二进制状态表。状态编码也称状态分配，或者状态赋值。

状态编码的任务是：

- ①确定状态编码的长度(即二进制代码的位数，或者说所需触发器个数);
- ②寻找一种最佳的或接近最佳的状态分配方案。以便使所设计的时序电路最简单。





## 一、确定二进制代码的位数

设最小化状态表的状态数为N，状态编码的长度为m，则状态数N与状态编码长度m的关系为：

$$2^{m-1} < N \leq 2^m$$

例如，若某状态表的状态数N = 7，则状态分配时，二进制代码的位数应为 m = 3。或者说状态变量个数为3。

## 二、确定状态分配方案

状态与代码之间的对应关系可以有许多种。一般说来，用m位二进制代码的 $2^m$ 种组合来对N个状态进行分配时，可能出现的状态分配方案数K<sub>s</sub>为

$$K_s = A_2^N = \frac{2^m!}{(2^m - N)!}$$

例如，当 N = 4,  
m = 2 时， K<sub>s</sub> = 24。

一种常用方法称为**相邻分配法**。





相邻分配法的状态编码原则如下：

- ① 次态相同，现态相邻。（即在相同输入条件下，具有相同次态的现态应尽可能分配相邻的二进制代码；）
- ② 现态相同，次态相邻。（即在相邻输入条件下，同一现态的次态应尽可能分配相邻的二进制代码；）
- ③ 输出相同，现态相邻。（即在每一种输入取值下均具有相同输出的现态应尽可能分配相邻的二进制代码。）

某些状态表常常出现不能同时满足3条原则的情况。此时，可按从①至③的优先顺序考虑。

此外，从电路实际工作状态考虑，一般将初始状态分配“0”状态。





### 三、举例

例 对如下状态表进行状态编码（设A为初始状态）。

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

解 所示状态表中，状态数  $N = 4$ ，故状态编码的长度应为  $m=2$ 。即实现该状态表的功能需要两个触发器。





根据相邻法的编码原则，4个状态的相邻关系如下：

根据原则①，状态B和C应分配相邻的二进制代码；

根据原则②，状态B和C、A和D、C和D应分配相邻的二进制代码；

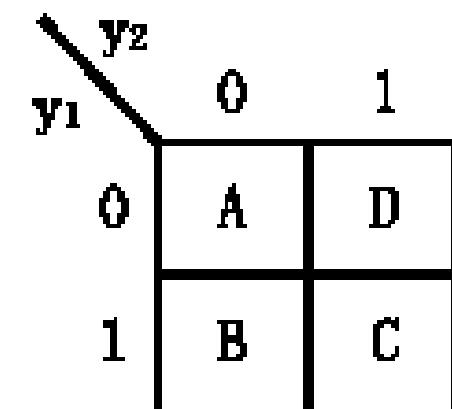
根据原则③，状态A和D应分配相邻的二进制代码。

综合①~③可知，状态分配时要求满足**B和C、A和D、C和D相邻**。

在进行状态分配时，为了使状态之间的相邻关系一目了然，通常将卡诺图作为状态分配的工具。

假定状态变量用 $y_2y_1$ 表示，并将A分配“0”，一种满足上述相邻关系的分配方案如右图所示。即状态**A、B、C、D**的状态编码依次为 $y_2y_1$ 的取值**00、01、11、10**。

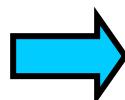
| 现态 | 次态/输出 |       |
|----|-------|-------|
|    | $x=0$ | $x=1$ |
| A  | C/1   | B/0   |
| B  | A/0   | A/1   |
| C  | A/1   | D/1   |
| D  | D/1   | C/0   |





将状态表中的状态A、B、C、D分别用编码00、01、11、10代替，即可得到该状态表的二进制状态表如右下表所示。

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



| 现态<br>$y_2y_1$ | 次态 $y_2^{(n+1)}y_1^{(n+1)}$ /输出 |      |
|----------------|---------------------------------|------|
|                | x=0                             | x=1  |
| 00             | 11/0                            | 01/0 |
| 01             | 00/0                            | 00/1 |
| 11             | 00/1                            | 10/1 |
| 10             | 10/0                            | 11/0 |

**注意：**满足分配原则的方案通常可以有多种，设计者可从中任选一种。





### 5.3.4 确定激励函数和输出函数并画出逻辑电路图

**任务：**根据二进制状态表和所选触发器的激励表，求出触发器的激励函数表达式和电路的输出函数表达式，并予以化简。以便用适当的逻辑门和所选定的触发器构成实现给定逻辑功能的逻辑电路。





根据二进制状态表和触发器激励表，求激励函数和输出函数的最简表达式一般分为**两步**：

- 列出激励函数和输出函数真值表；
- 用卡诺图化简后写出最简表达式。

**例** 用J-K触发器和适当的逻辑门实现如下二进制状态表的功能。

| 现态<br>$y_2y_1$ | 次态 $y_2^{(n+1)}y_1^{(n+1)}$ /输出 |       |
|----------------|---------------------------------|-------|
|                | $x=0$                           | $x=1$ |
| 00             | 11/0                            | 01/0  |
| 01             | 00/0                            | 00/1  |
| 10             | 01/0                            | 11/0  |
| 11             | 00/1                            | 10/1  |





| 现态<br>$y_2y_1$ | 次态 $y_2^{(n+1)}y_1^{(n+1)}$ /输出 |       |
|----------------|---------------------------------|-------|
|                | $x=0$                           | $x=1$ |
| 00             | 11/0                            | 01/0  |
| 01             | 00/0                            | 00/1  |
| 10             | 01/0                            | 11/0  |
| 11             | 00/1                            | 10/1  |

| $QQ^{(n+1)}$ | J K |
|--------------|-----|
| 00           | 0 d |
| 01           | 1 d |
| 10           | d 1 |
| 11           | d 0 |

**解** 根据给定的二进制状态表和J-K触发器的激励表可列出激励函数和输出函数的真值表如右下表所示。

激励函数和输出函数真值表

| $x$ | $y_2$ | $y_1$ | 激励函数<br>$J_2K_2 J_1K_1$ | 输出函数<br>Z |
|-----|-------|-------|-------------------------|-----------|
| 0   | 0     | 0     | 1 d 1 d                 | 0         |
| 0   | 0     | 1     | 0 d d 1                 | 0         |
| 0   | 1     | 0     | d 1 1 d                 | 0         |
| 0   | 1     | 1     | d 1 d 1                 | 1         |
| 1   | 0     | 0     | 0 d 1 d                 | 0         |
| 1   | 0     | 1     | 0 d d 1                 | 1         |
| 1   | 1     | 0     | d 0 1 d                 | 0         |
| 1   | 1     | 1     | d 0 d 1                 | 1         |





由真值表可作出激励函数和输出函数的卡诺图如下图所示。

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

$$J_2 = \overline{x} \cdot \overline{y}_1$$

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

$$K_2 = \overline{x}$$

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

$$J_1=1$$

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

$$K_1=1$$

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

$$Z$$

$$Z = y_2 y_1 + xy_1 = (y_2 + x)y_1$$



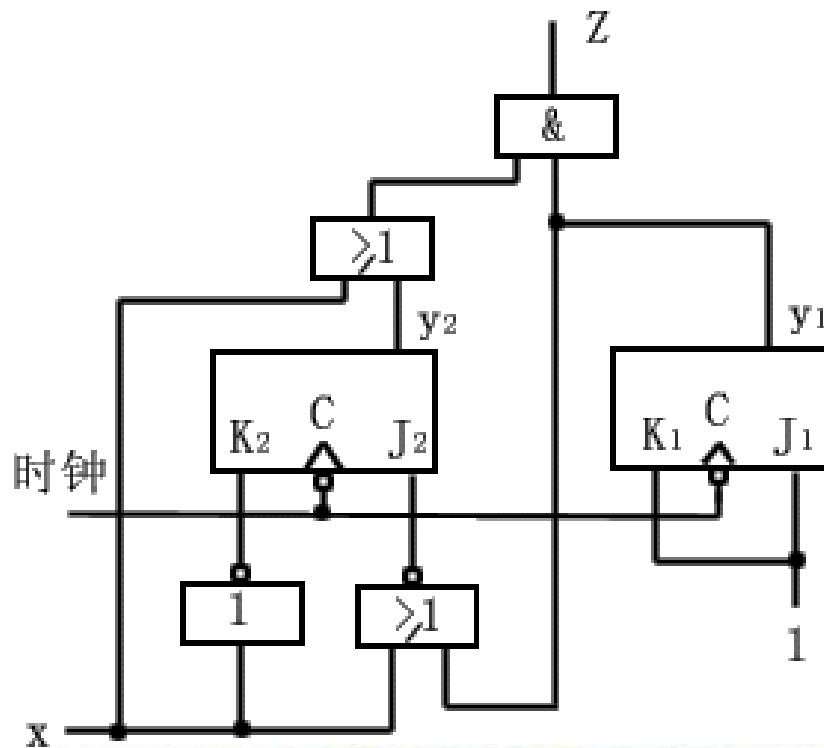


经化简后得到激励函数和输出函数的最简表达式如下：

$$J_2 = \overline{x}y_1 ; \quad J_1 = K_1 = 1$$

$$K_2 = \overline{x} ; \quad Z = y_2y_1 + xy_1 = (y_2 + x)y_1$$

相应逻辑电路图如下图所示。





## 5.4 同步时序逻辑电路设计举例

例5.16：用JK触发器作为存储元件，设计一个“101”序列检测器。该电路有一个输入x和一个输出z，当随机输入信号中出现“101”序列时，输出一个1信号。典型输入/输出序列如下：

输入x: 0 0 1 0 1 0 1 1 0 1 0 0

输出z: 0 0 0 0 1 0 1 0 0 1 0 0

