



第六章 同步时序逻辑电路(二)

秦磊华 计算机学院

6.5 时序电路的描述工具

6.6 同步时序电路分析

6.7 同步时序电路设计基础

CONTENT

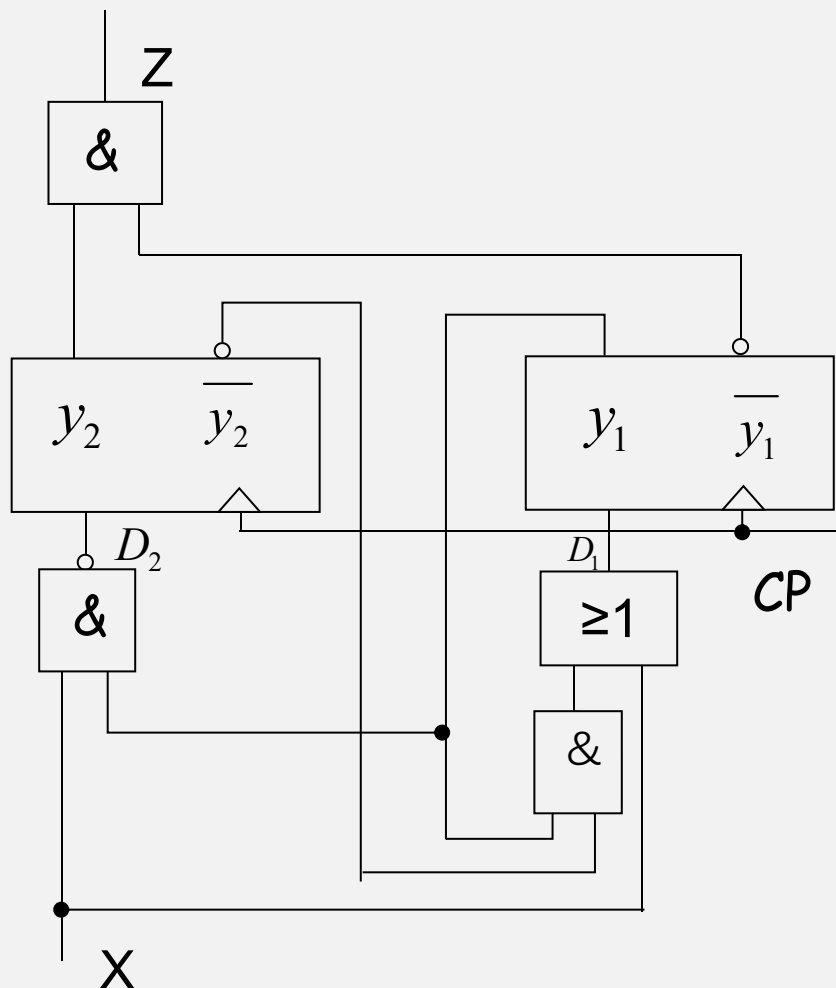


6.5 时序逻辑电路的描述工具

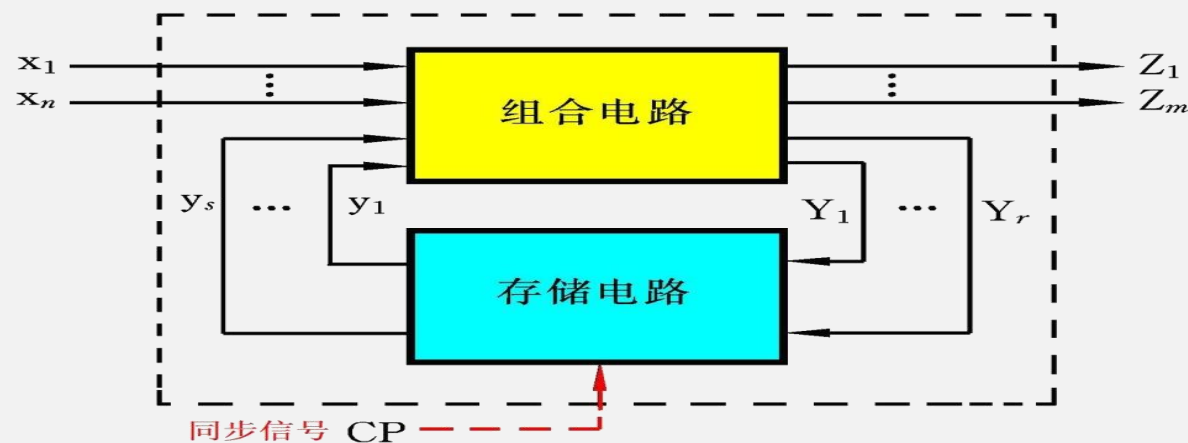
- 逻辑函数表达式
- 状态转移表
- 状态转换图
- 时间图

6.5 时序逻辑电路的描述工具

1. 逻辑函数表达式



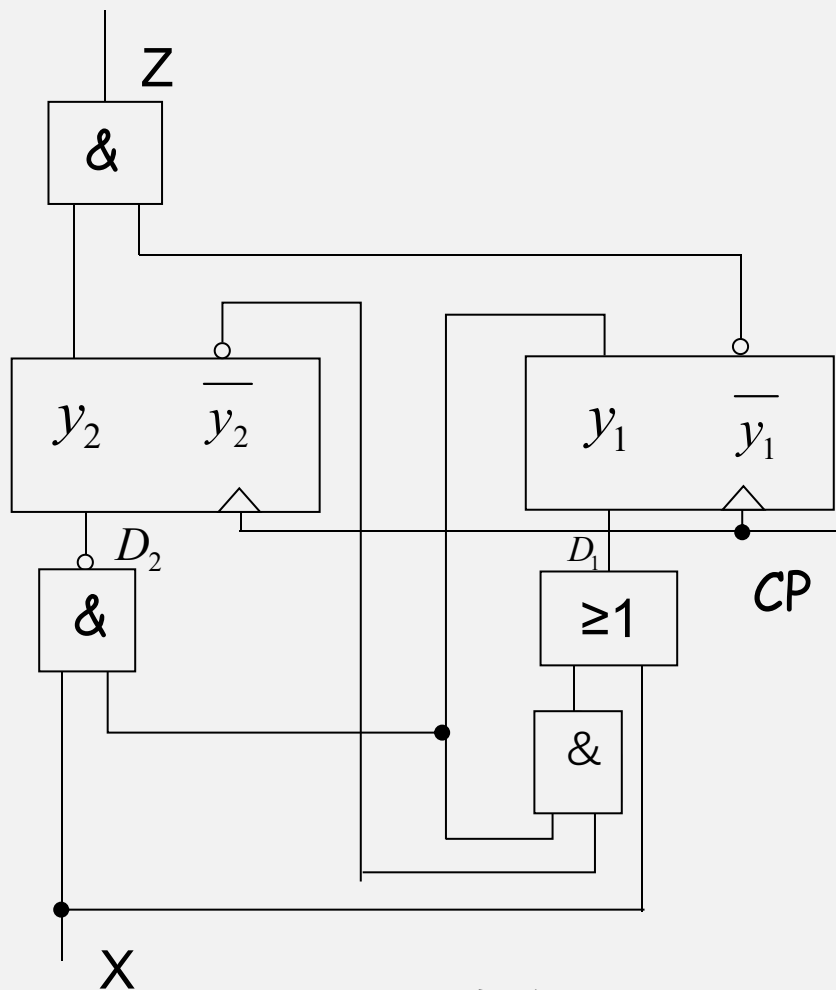
Moore型电路



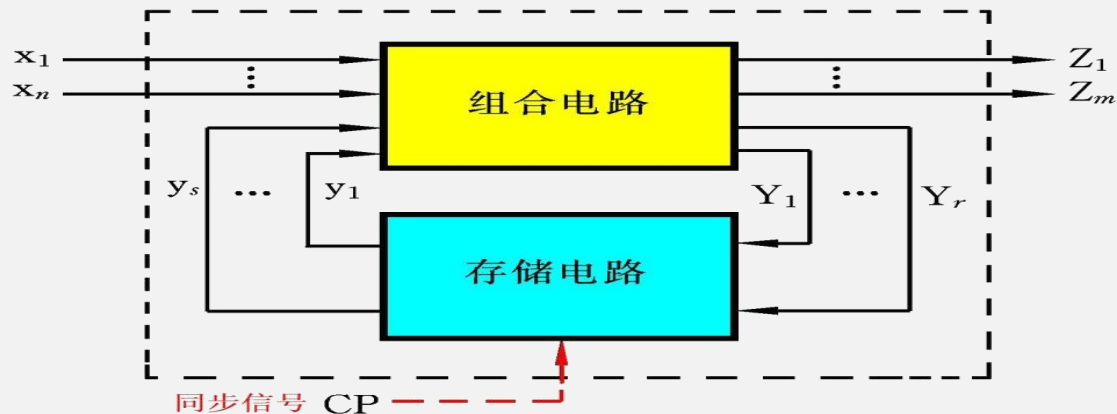
- x_1, \dots, x_n : (外)输入信号;
- Z_1, \dots, Z_m : (外)输出信号;
- y_1, \dots, y_s : (内输入)电路的“状态”-现态 y^n 或 y ;
- Y_1, \dots, Y_r : (内输出)“次态”(y^{n+1})-电路的激励信号;

6.5 时序逻辑电路的描述工具

1. 逻辑函数表达式



Moore型电路

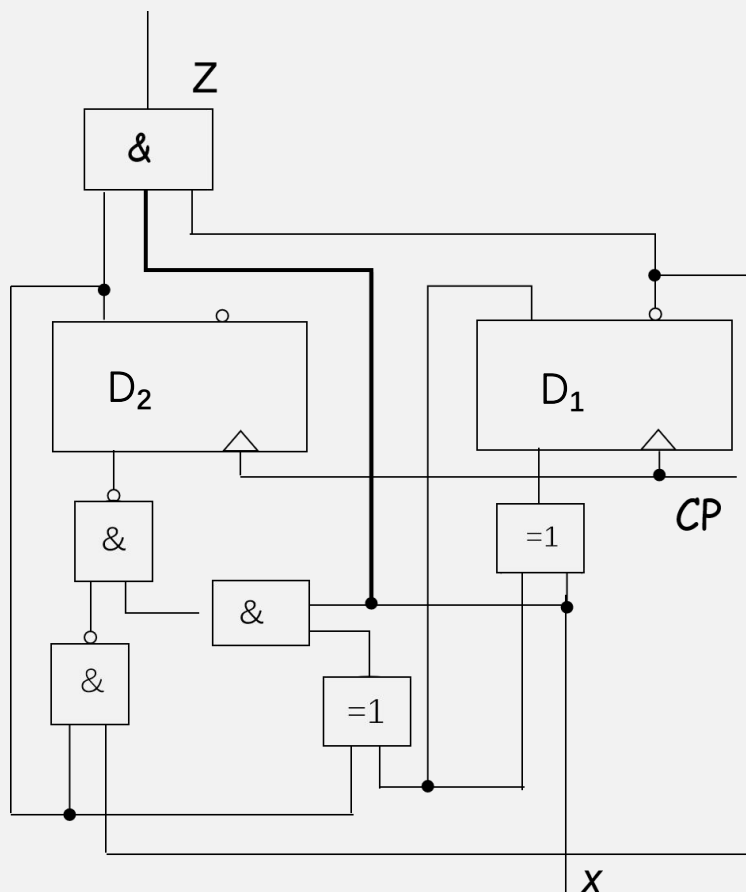


- (外)输出函数: $Z = y_2 \overline{y_1}$
- (内输入)电路的“状态”-现态: $y_2 y_1$
- (内输出)“次态”(yⁿ⁺¹): $y^{n+1} = D$
- (触发器的输入函数)激励函数:

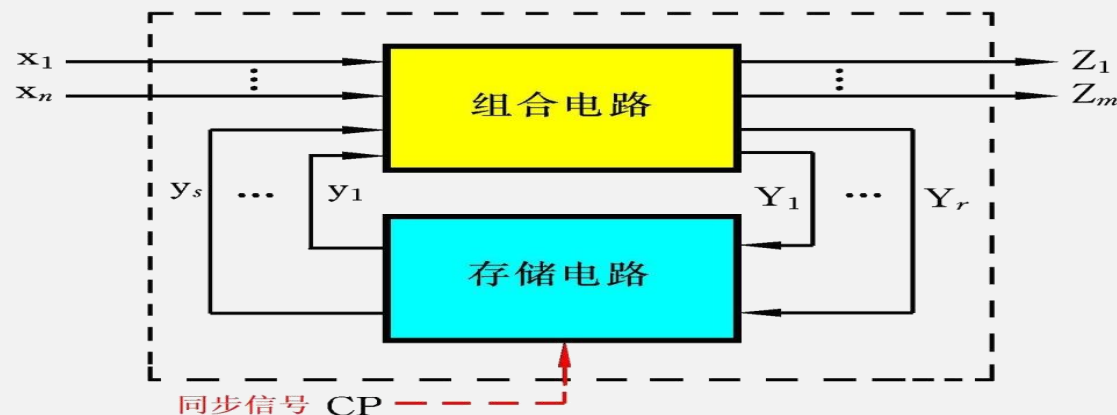
$$\begin{aligned} D_2 &= \overline{xy_1} \\ D_1 &= x + \overline{y_2 y_1} \end{aligned}$$

6.5 时序逻辑电路的描述工具

1. 逻辑函数表达式



Mealy型电路



- (外)输出函数: $Z = xy_2\overline{y_1}$
- (内输入)电路的“状态”-现态: $y_2 y_1$
- (内输出)“次态”(yⁿ⁺¹): $y_{n+1} = y_n \oplus T$
- (触发器的输入函数)激励函数:

$$\begin{aligned} T_2 &= \overline{y_2 y_1} \cdot (y_2 \oplus y_1)x \\ &= \overline{y_2 y_1} + (y_2 \oplus y_1)x \\ T_1 &= x \oplus y_1 \end{aligned}$$

6.5 时序逻辑电路的描述工具

2.状态转移表

反映同步电路输出Z、次态 y^{n+1} 与电路输入x、现态y之间关系的表
(分析与设计同步时序电路的必备工具)

现态 y	次态 y^{n+1} /输出Z		
		输入 x	
y		y^{n+1}/Z	

Mealy型

现态 $y_2 \quad y_1$	次态 $y_2^{n+1} \quad y_1^{n+1}$ /输出Z	
	x=0	x=1
0 0	0 1 /0	1 1 /1
0 1	1 0 /0	0 0 /0
1 0	1 1 /0	0 1 /0
1 1	0 0 /1	1 0 /0

6.5 时序逻辑电路的描述工具

2.状态转移表

反映同步电路输出Z、次态 y^{n+1} 与电路输入x、现态y之间关系的表
(分析与设计同步时序电路的必备工具)

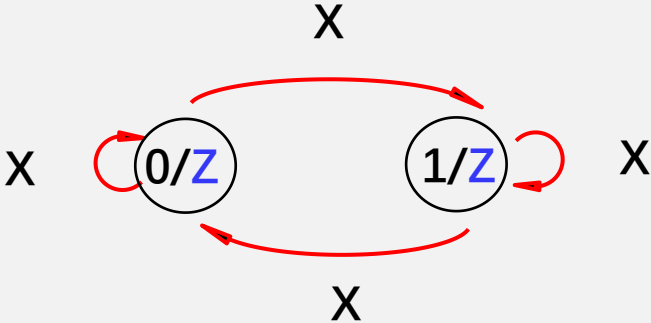
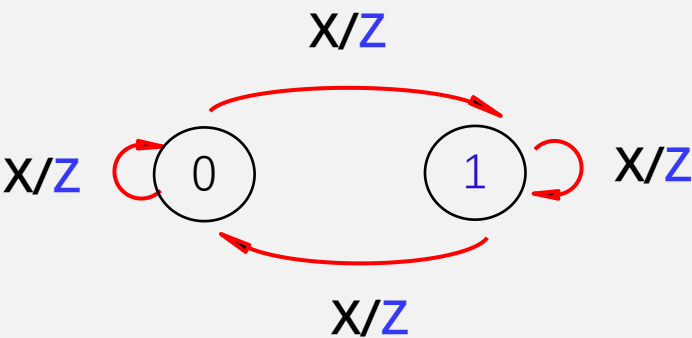
现态 y		次态 y^{n+1}		输出 Z
		输入 x		
y		y^{n+1}		Z

Moore型

现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$		输出 Z
	x=0	x=1	
0 0	0 0	0 1	0
0 1	1 0	0 1	0
1 0	0 0	1 1	0
1 1	1 0	0 1	1

6.5 时序逻辑电路的描述工具

2. 状态转移图



6.6 同步时序逻辑电路分析

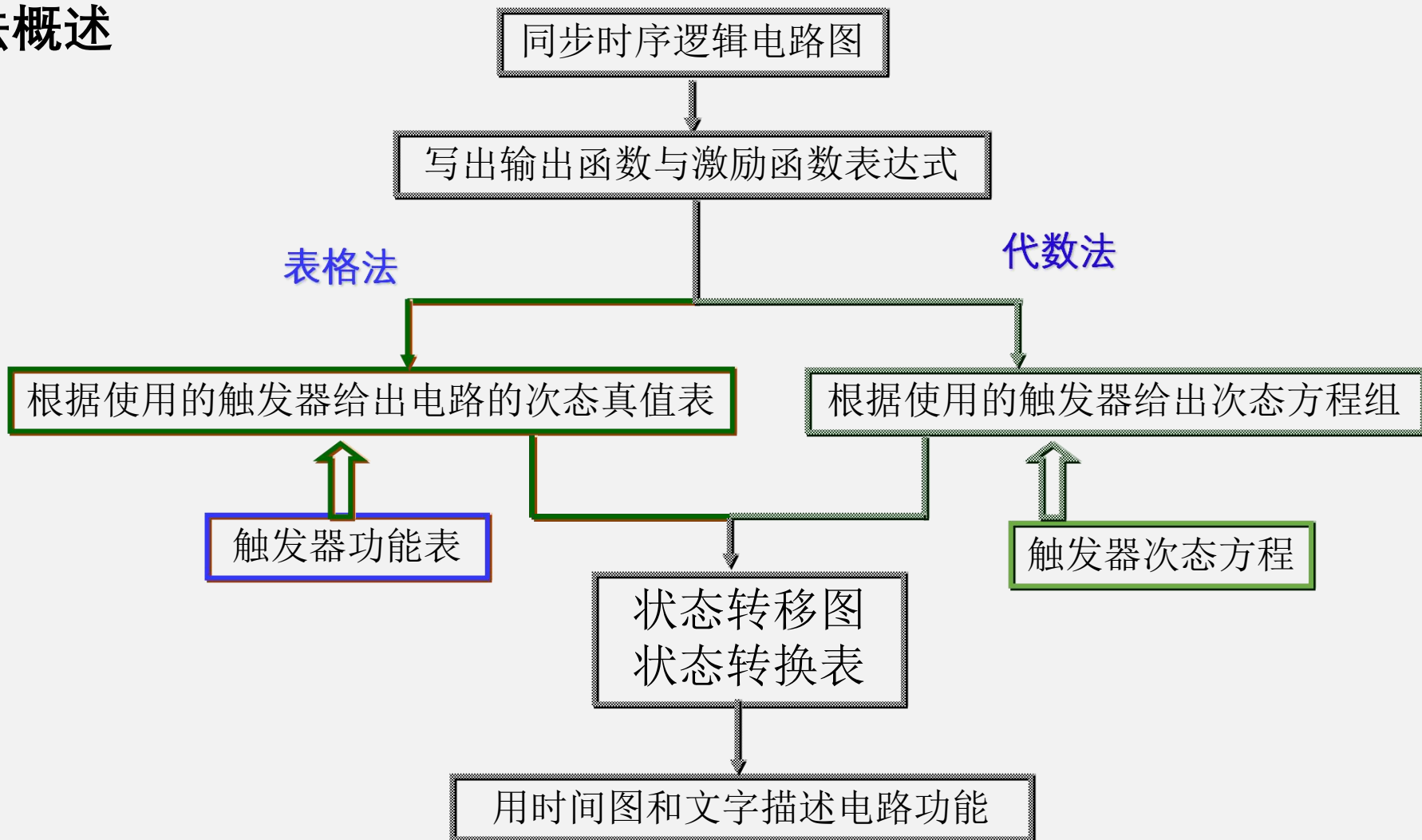
1. 分析方法概述

目标：找出电路状态和输出随输入变化而变化的规律

常用的方法：表格法、代数法

6.6 同步时序逻辑电路分析

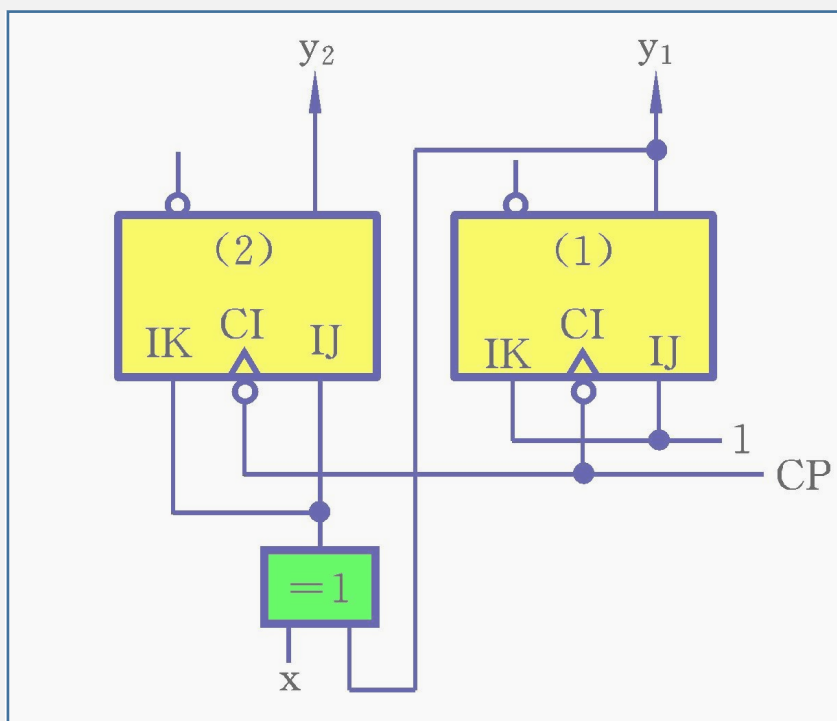
1. 分析方法概述



6.6 同步时序逻辑电路分析

2.表格法分析举例

例1 分析下图所示同步时序逻辑电路。

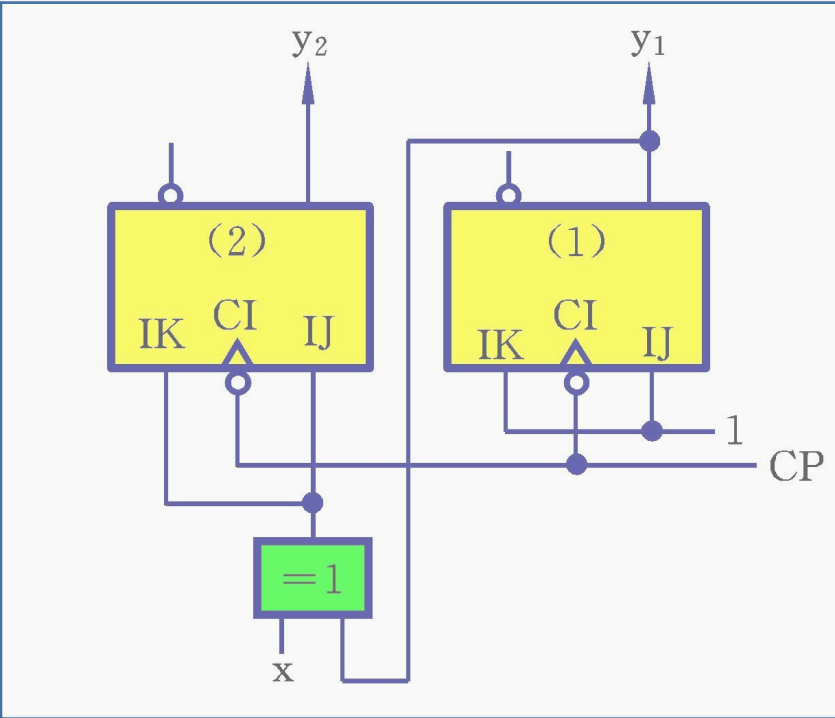


电路结构

- 2个J-K触发器
- 1个异或门
- 外输入:x
- 状态: y_2 、 y_1
- 输出: ? y_2y_1
- 电路类型: ? Moore型

2.表格法分析举例

例1 分析下图所示同步时序逻辑电路。



1)写出输出函数和激励函数表达式

$$J_2 = K_2 = x \oplus y_1 \qquad J_1 = K_1 = 1$$

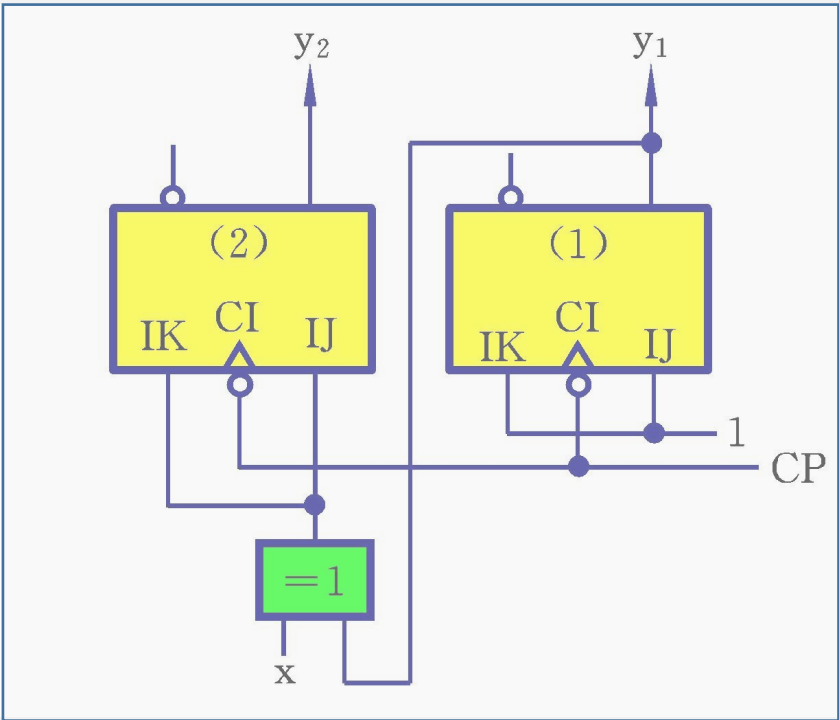
2)列出电路次态真值表

输入 x	现态 y ₂ y ₁	激励函数 J ₂ K ₂ J ₁ K ₁	次 态 y ₂ ⁿ⁺¹ y ₁ ⁿ⁺¹
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

6.6 同步时序逻辑电路分析

2.表格法分析举例

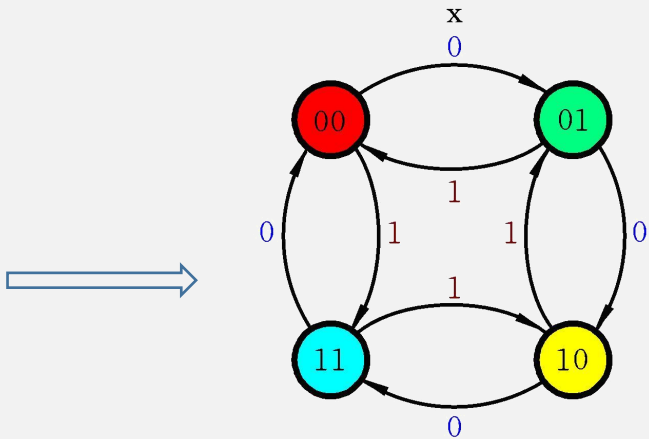
例1 分析下图所示同步时序逻辑电路。



3)作出状态表和状态图

输入 x	现态 $y_2 y_1$	激励函数 $J_2 K_2 J_1 K_1$	次态 $y_2^{n+1} y_1^{n+1}$
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

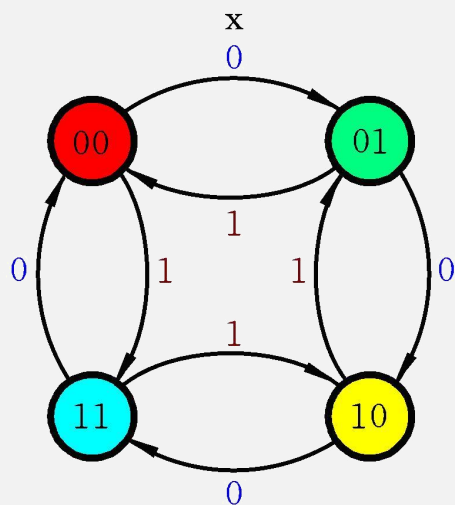
现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	
	x=0	x=1
0 0	0 1	1 1
0 1	1 0	0 0
1 0	1 1	0 1
1 1	0 0	1 0



2.表格法分析举例

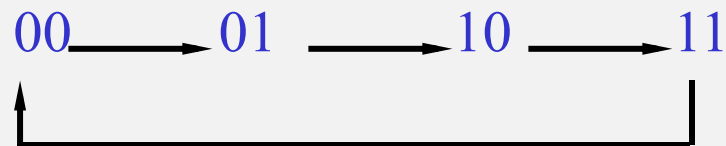
例1 分析下图所示同步时序逻辑电路。

4)功能评述

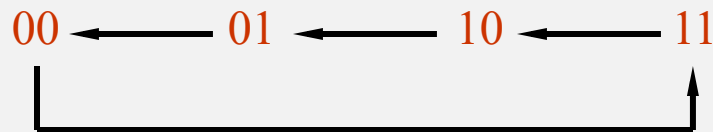


由状态图可知，该电路是一个2位二进制数可逆计数器。

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

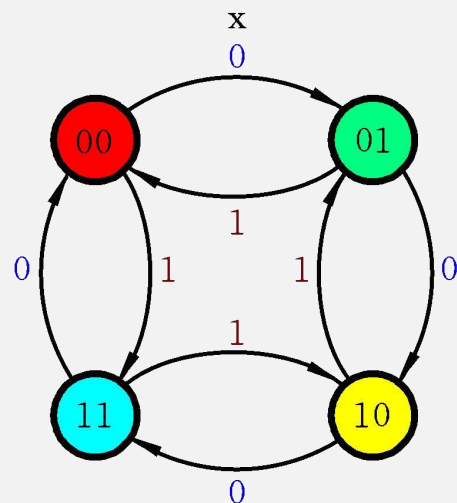


6.6 同步时序逻辑电路分析

2.表格法分析举例

例1 分析下图所示同步时序逻辑电路。 5)画典型时间图

设电路初始状态 $y_2y_1=00$, 典型输入序列为111100000,电路状态响应序列如下:



CP	1	2	3	4	5	6	7	8	9
X	1	1	1	1	0	0	0	0	0
y ₂	0								
y ₁	0						-		
y ₂ ⁿ⁺¹									
y ₁ ⁿ⁺¹									

6.6 同步时序逻辑电路分析

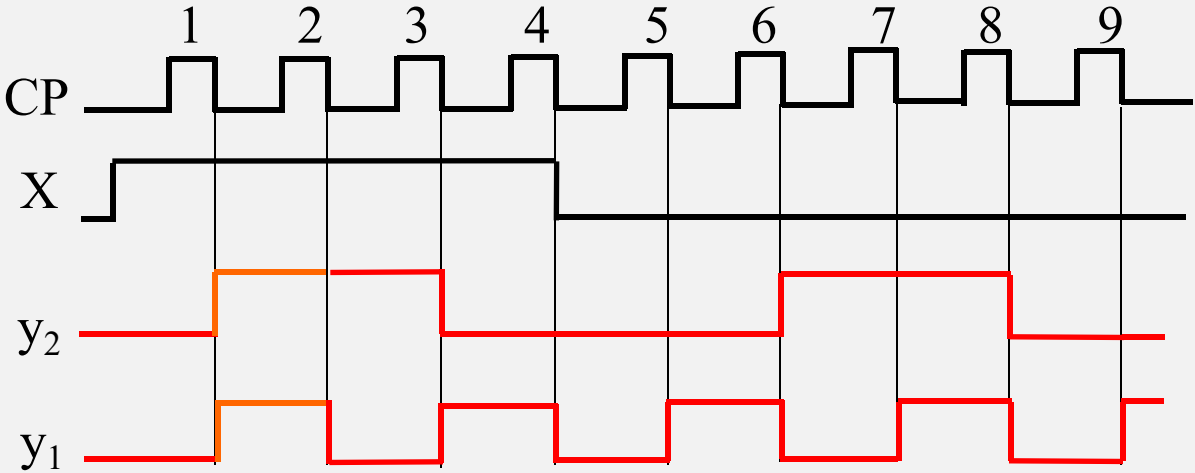
2.表格法分析举例

例1 分析下图所示同步时序逻辑电路。

5)画典型时间图

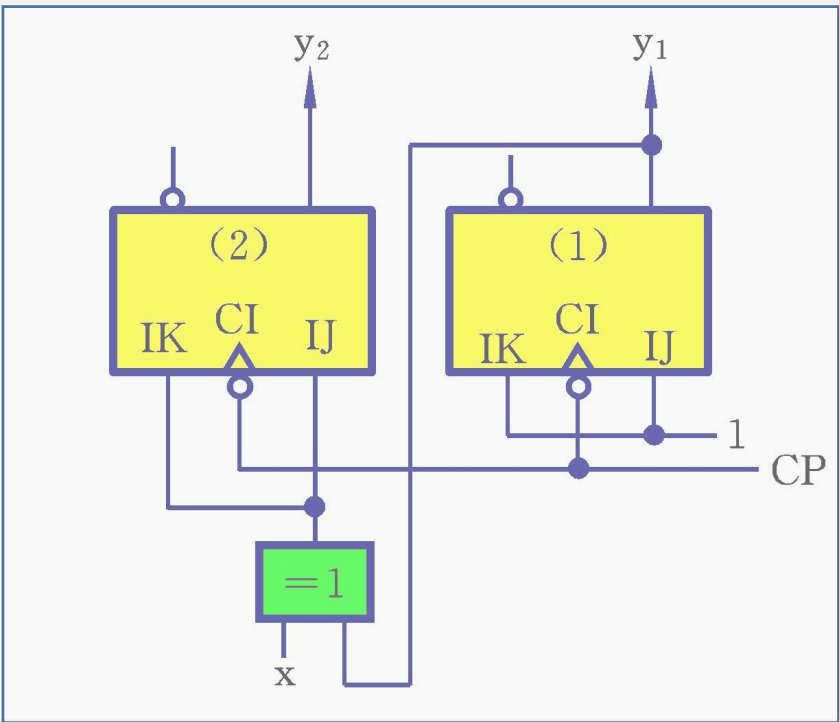
设电路初始状态 $y_2y_1=00$ ，典型输入序列为111100000,电路状态响应序列如下：

CP	1	2	3	4	5	6	7	8	9
X	1	1	1	1	0	0	0	0	0
Y_2	0	1	1	0	0	0	1	1	0
Y_1	0	1	0	1	0	1	0	1	0
Y_2^{n+1}	1	1	0	0	0	1	1	0	0
Y_1^{n+1}	1	0	1	0	1	0	1	0	1



3.代数分析法举例

例2 分析下图所示同步时序逻辑电路。



1)写出输出函数和激励函数表达式

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

2)把激励函数代入触发器次态方程

$y^{n+1} = J\bar{y} + \bar{K}y$ 得到电路次态方程组

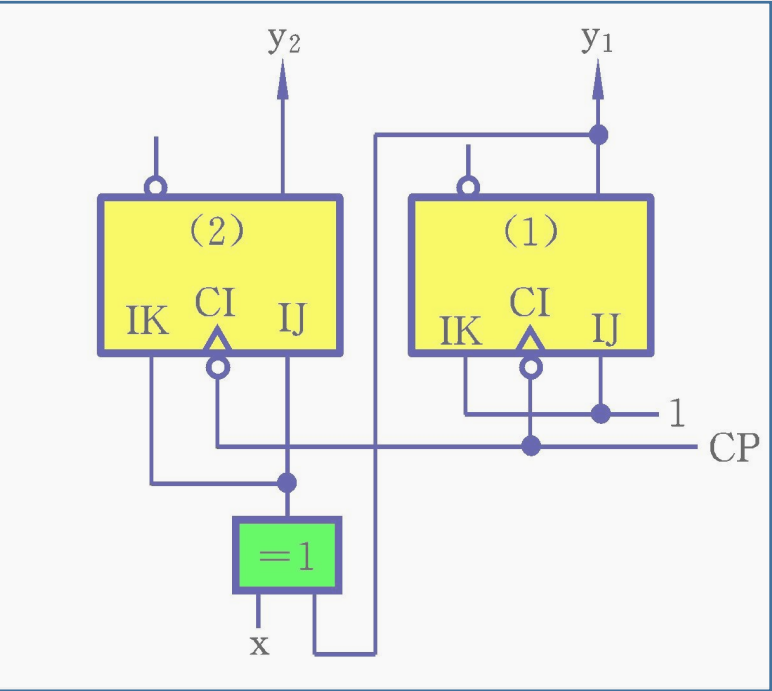
$$\begin{aligned} y_2^{n+1} &= J_2 \bar{y}_2 + \bar{K}_2 y_2 \\ &= (x \oplus y_1) \bar{y}_2 + \overline{(x \oplus y_1)} y_2 \\ &= \overline{xy_1} \bar{y}_2 + \bar{x} \bar{y}_1 \bar{y}_2 + xy_1 y_2 + \overline{\bar{x} \bar{y}_1} y_2 \end{aligned}$$

$$\begin{aligned} y_1^{n+1} &= J_1 \bar{y}_1 + \bar{K}_1 y_1 \\ &= \bar{y}_1 \end{aligned}$$

3.代数分析法举例

例2 分析下图所示同步时序逻辑电路。

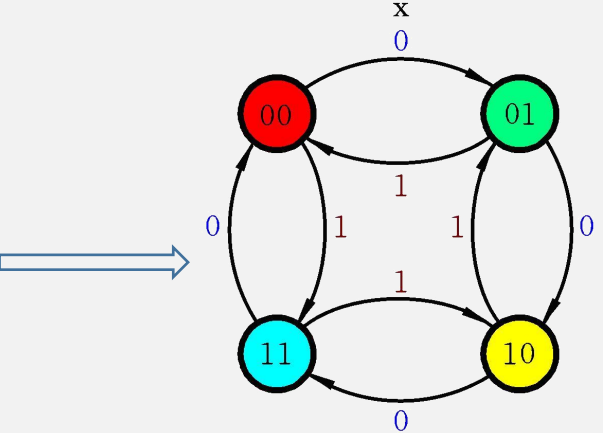
3)作出状态表和状态图



$$\begin{aligned}
 y_2^{n+1} &= y_2 \oplus y_1 \quad (x=0) \\
 y_2^{n+1} &= y_2 \odot y_1 \quad (x=1)
 \end{aligned}$$

$$\begin{aligned}
 y_2^{n+1} &= J_2 \overline{y_2} + \overline{K_2} y_2 \\
 &= (x \oplus y_1) \overline{y_2} + (\overline{x \oplus y_1}) y_2 \\
 &= \overline{x} \overline{y_1} \overline{y_2} + \overline{x} y_1 \overline{y_2} + x y_1 y_2 + x \overline{y_1} y_2 \\
 y_1^{n+1} &= J_1 \overline{y_1} + \overline{K_1} y_1 \\
 &= \overline{y_1}
 \end{aligned}$$

现态 $y_2 \quad y_1$		次态 $y_2^{n+1} \quad y_1^{n+1}$	
		$x=0$	$x=1$
0	0	0 1	1 1
0	1	1 0	0 0
1	0	1 1	0 1
1	1	0 0	1 0



3.代数分析法举例

例3 分析下图所示同步时序逻辑电路。



Z与状态和X直接相关，属于Mealy型电路。

1) 写出输出函数和激励函数表达式

$$Z = xy_2 \overline{y_1}$$

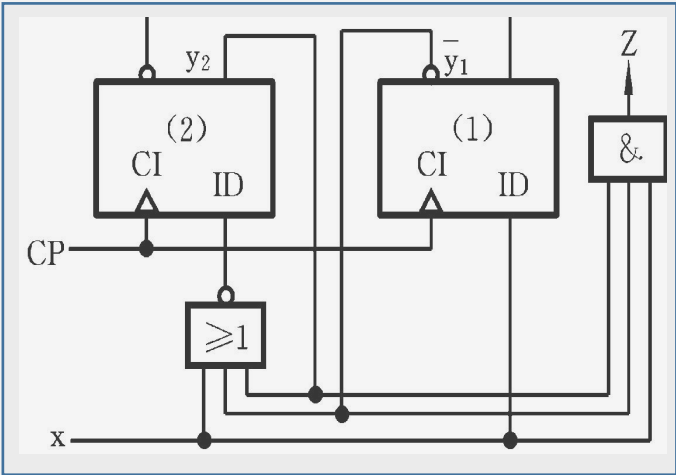
$$D_2 = \overline{x + y_2 + y_1} = \overline{\overline{x} \overline{y_2} \overline{y_1}}$$

$$D_1 = X$$

6.6 同步时序逻辑电路分析

3.代数分析法举例

例3 分析下图所示同步时序逻辑电路。



2)列出电路次态真值表

$$Z = xy_2\overline{y_1}$$
$$D_2 = \overline{x + y_2 + y_1} = \overline{x}y_2y_1$$
$$D_1 = x$$

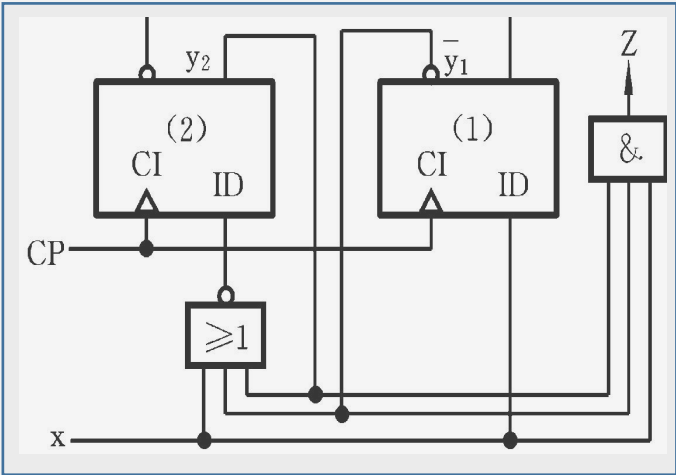
次态真值表

输 入 X	现 态 y ₂ y ₁	激励函数		次 态	
				y ₂ ⁽ⁿ⁺¹⁾	y ₁ ⁽ⁿ⁺¹⁾
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	0	0	0	0
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	0	1

6.6 同步时序逻辑电路分析

3.代数分析法举例

例3 分析下图所示同步时序逻辑电路。



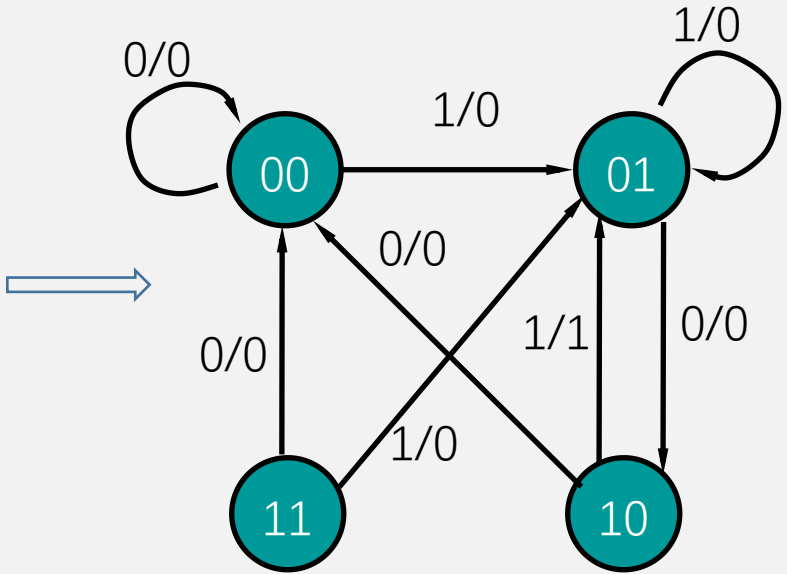
3)由次态真值表得到状态
(转换)表和状态(转换)图

次态真值表

输入 X	现 态 $y_2 y_1$	激励函数		次 态	
		D_2	D_1	$y_2^{(n+1)}$	$y_1^{(n+1)}$
0	0 0	0	0	0	0
0	0 1	1	0	1	0
0	1 0	0	0	0	0
0	1 1	0	0	0	0
1	0 0	0	1	0	1
1	0 1	0	1	0	1
1	1 0	0	1	0	1
1	1 1	0	1	0	1



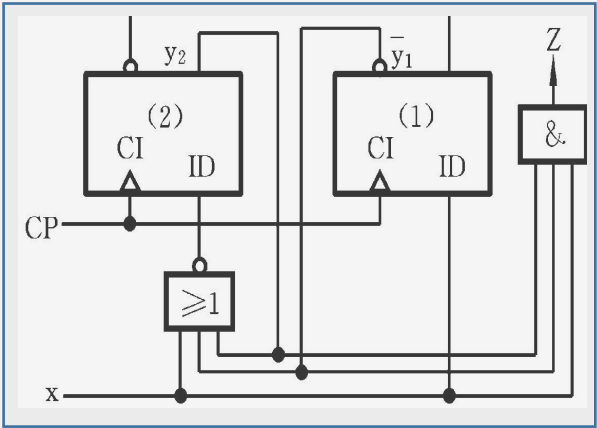
现 态 $y_2 y_1$	次态/输出($y_2^{n+1} y_1^{n+1} / Z$)	
	X=0	X=1
00	00/0	01/0
01	10/0	01/0
10	00/0	01/1
11	00/0	01/0



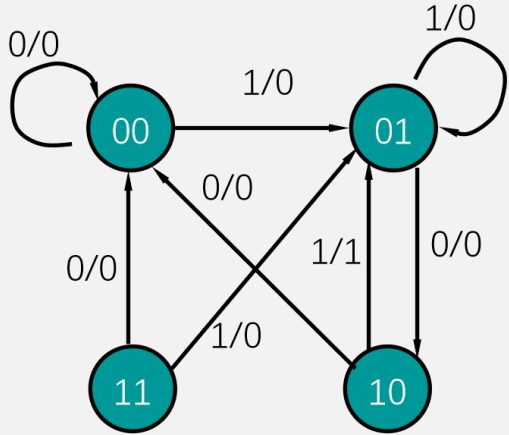
6.6 同步时序逻辑电路分析

3.代数分析法举例

例3 分析下图所示同步时序逻辑电路。



4)分析电路功能



分析各状态的意义
及状态转换的意义

从典型输入、输出序列中去找

CP	1	2	3	4	5	6	7	8	9
X	0	1	0	1	1	0	1	0	0
Y_2	0	0	0	1	0	0	1	0	1
Y_1	0	0	1	0	1	1	0	1	0
Y_2^{n+1}	0	0	1	0	0	1	0	1	0
Y_1^{n+1}	0	1	0	1	1	0	1	0	0
Z	0	0	0	1	0	0	1	0	0

CP	1	2	3	4	5	6	7	8
X	0	1	0	1	0	1	0	0
Y_2	0	0	0	1	0	1	0	1
Y_1	0	0	1	0	1	0	1	0
Y_2^{n+1}	0	0	1	0	1	0	1	0
Y_1^{n+1}	0	1	0	1	0	1	0	0
Z	0	0	0	1	0	1	0	0

存在局限性！

1.设计步骤概述(对应于分析步骤)

1)逻辑抽象，得出电路的状态转换图或状态转换表

(1)进行逻辑定义

确定输入、输出变量及电路状态数；

(2)定义输入、输出逻辑状态和每个电路状态的含义，并将电路状态顺序编号；

(3)列出电路的状态转换表或画出电路的状态转换图；

正确？完备？最简？

2)状态化简

电路的状态数越少，对应的电路也越简单。

1.设计步骤概述(对应于分析步骤)

3)状态分配及编码

(1)确定触发器数目 $n = \lceil \log_2 M \rceil$

(2)状态编码

给每个电路状态分配二进制编码(触发器状态组合)。编码方案得当有利于简化电路。存在编码总状态数大于电路状态数的情况，如电路有3个状态，用2个触发器时有00、01、10、11等四个状态编码，用其中的任何3种即可。

4)选定触发器的类型设计电路

结合选用触发器，求出电路的状态方程、激励函数和输出函数并设计对应的电路。

5)电路分析检查

2. 完全确定状态表的化简

1)几个概念

◆ 等效对的概念

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

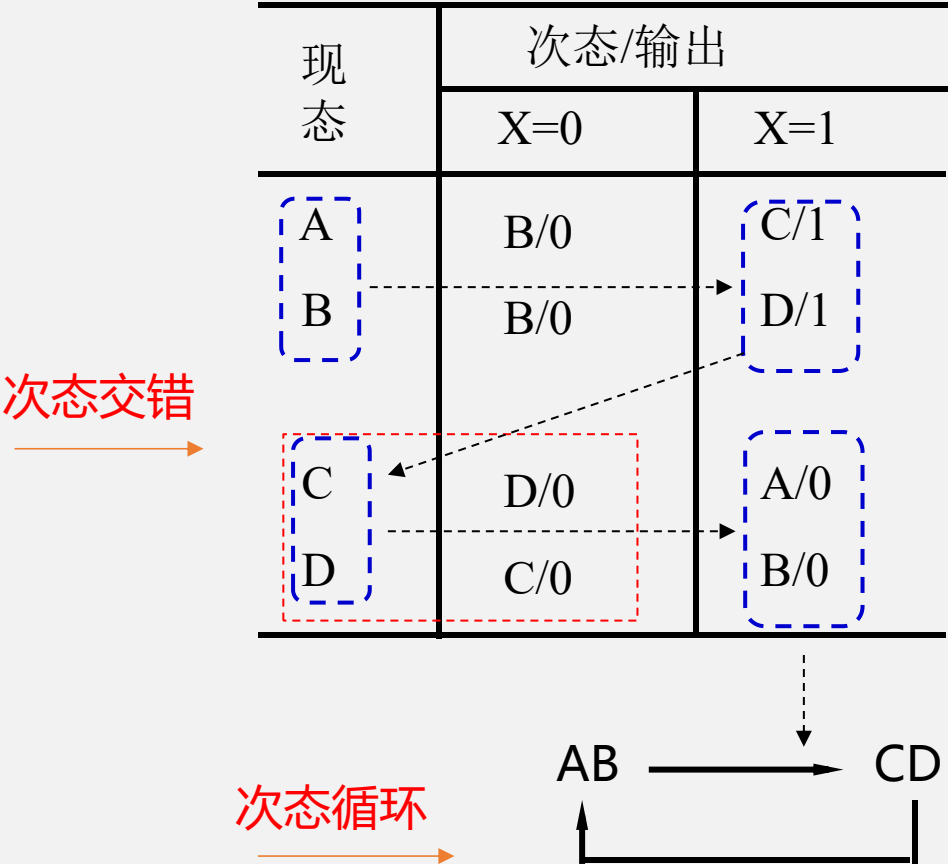
状态 S_i 和 S_j 是**完全确定状态**表中的两个状态，若对于所有可能的输入序列，分别从状态 S_i 和状态 S_j 出发，所得到的输出响应序列完全相同，则状态 S_i 和 S_j 等效，记作 (S_i, S_j) ，称为等效对。

2. 完全确定状态表的化简

1)几个概念

◆ 等效判断方法

- 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。



2. 完全确定状态表的化简

◆ 等效对具有传递性 $(S_1, S_2), (S_2, S_3) \rightarrow (S_1, S_3)$

2) 等效类 - 等效状态的集合

$$\underbrace{(S_1, S_2), (S_2, S_3) \rightarrow (S_1, S_3)}_{\{S_1, S_2, S_3\}}$$

↓

等效类是广义概念，两个状态或多个状态均可以组成一个等效类，甚至一个状态也可以构成等效类。

2. 完全确定状态表的化简

3)最大等效类

不被任何其它等效类包含的等效类。

---(不是指状态最多，而是指其独立性)

原始状态表的化简过程，就是寻找所有最大等效类的过程，每个最大等效类中的状态合并为一个新的状态。

简化后的状态数等于最大等效类的个数！

6.7 同步时序逻辑电路设计基础

2. 完全确定状态表的化简

4) 化简过程

作隐含表



寻找等效对



找出所有最大等效类



状态合并，作出最小化状态表

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

◆按等效对判断规则，对原始状态表任意两原态逐一检查比较，将结果标注在隐含表中相应方格：

等效：“√”；

不等效：“×”；

无法判断：填相关的状态对

B						
C						
D						
E						
F						
G						
	A	B	C	D	E	F

6.7 同步时序逻辑电路设计基础

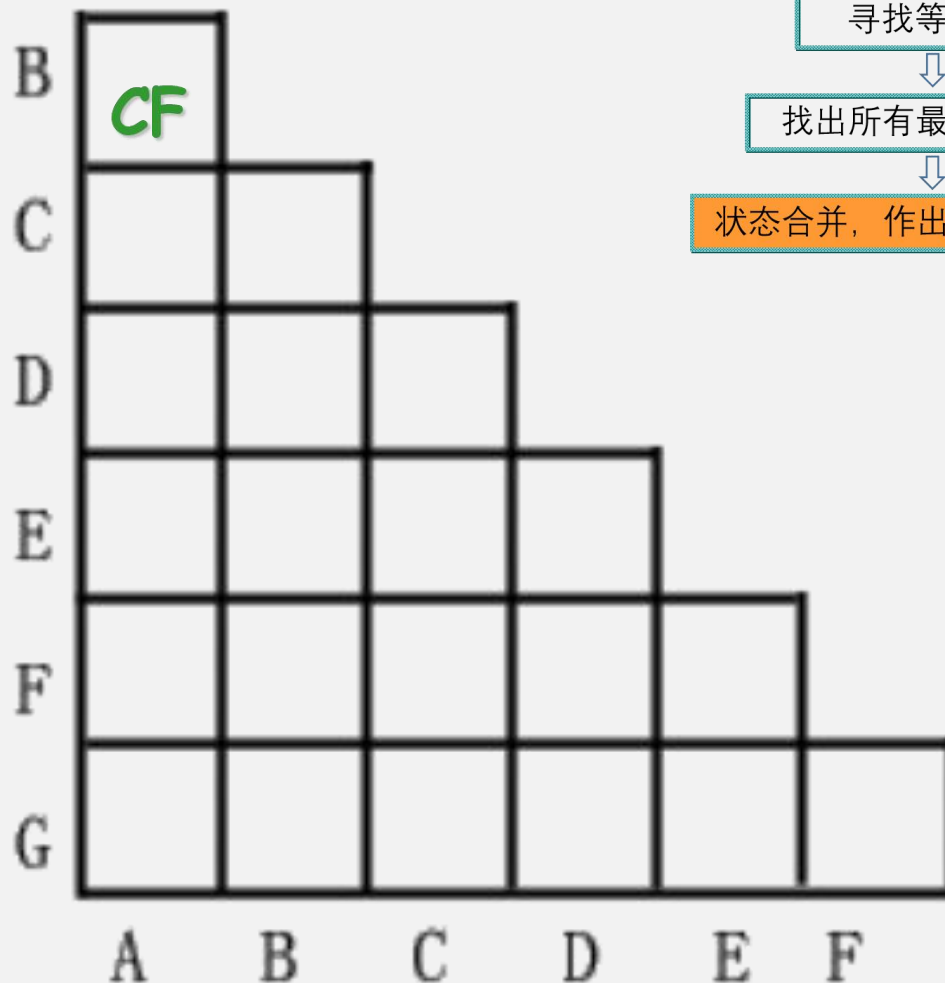
2. 完全确定状态表的化简

4) 化简过程

例1 化简下表所示原始状态表

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



作隐含表

寻找等效对

找出所有最大等效类

状态合并，作出最小化状态表

6.7 同步时序逻辑电路设计基础

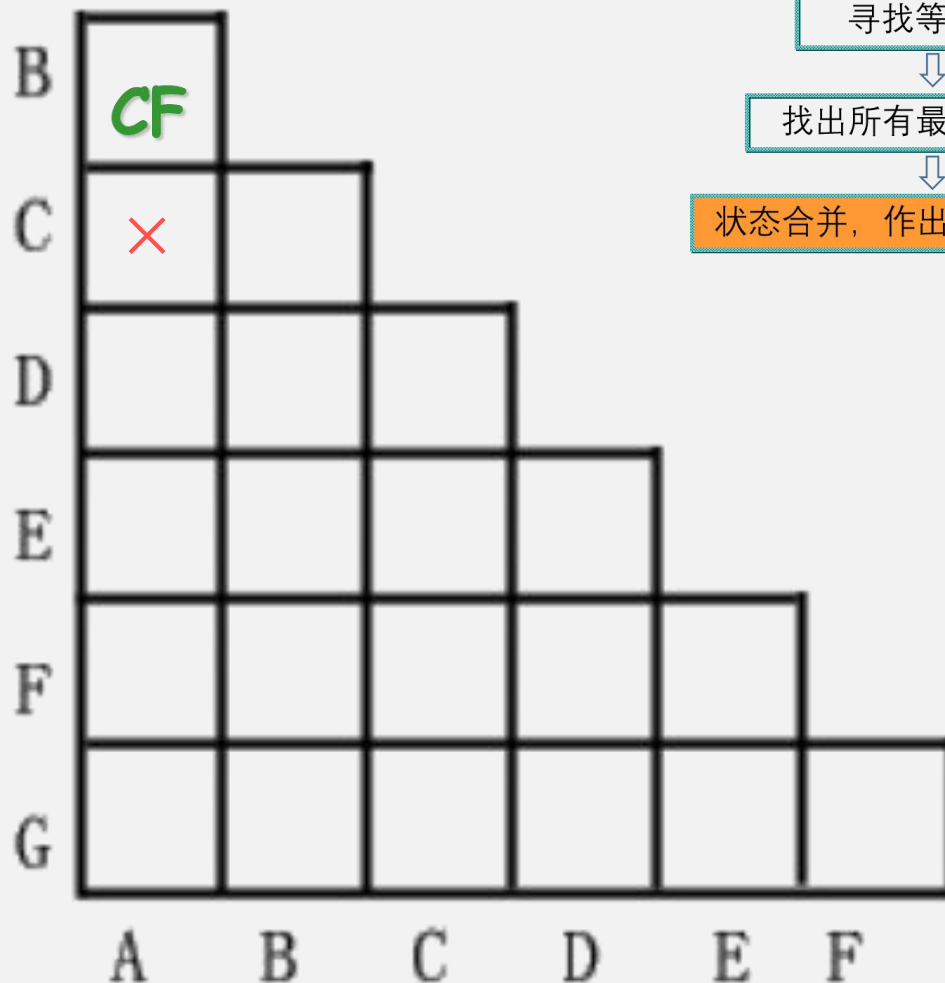
2. 完全确定状态表的化简

4) 化简过程

例1 化简下表所示原始状态表

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



作隐含表



寻找等效对



找出所有最大等效类



状态合并，作出最小化状态表

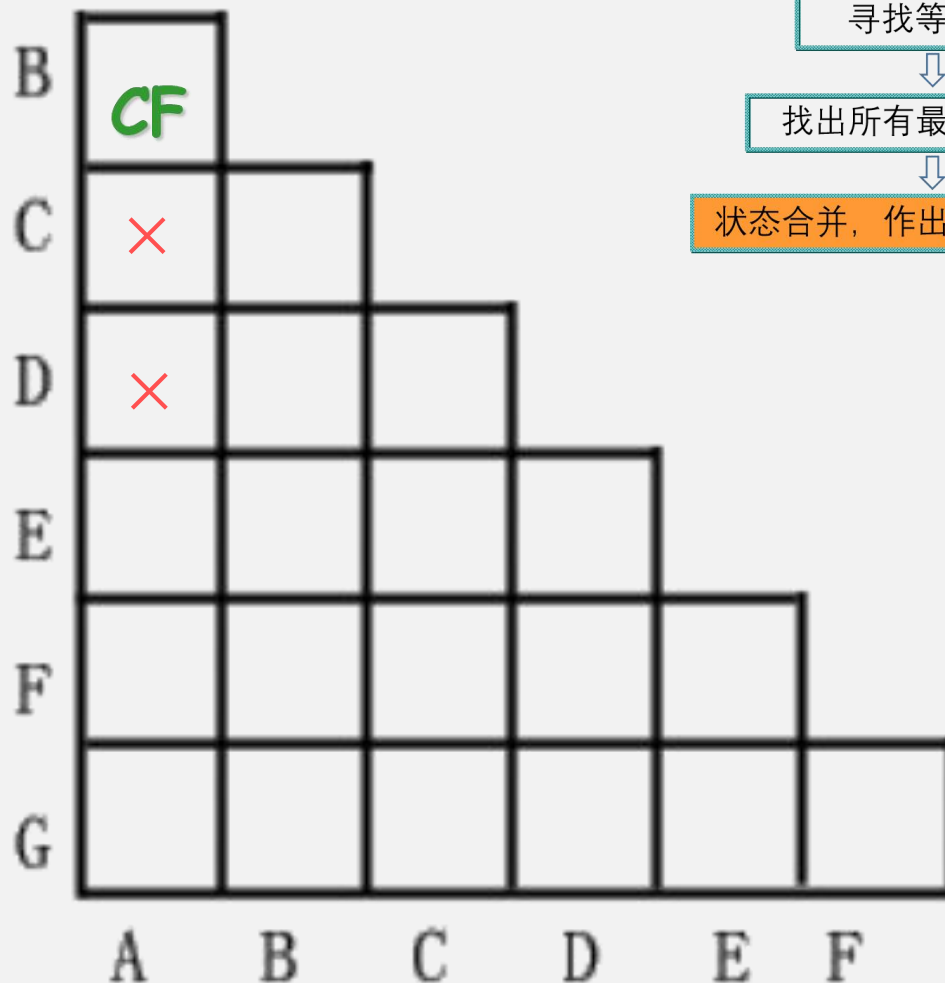
6.7 同步时序逻辑电路设计基础

2. 完全确定状态表的化简

4) 化简过程

例1 化简下表所示原始状态表

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



作隐含表

寻找等效对

找出所有最大等效类

状态合并，作出最小化状态表

6.7 同步时序逻辑电路设计基础

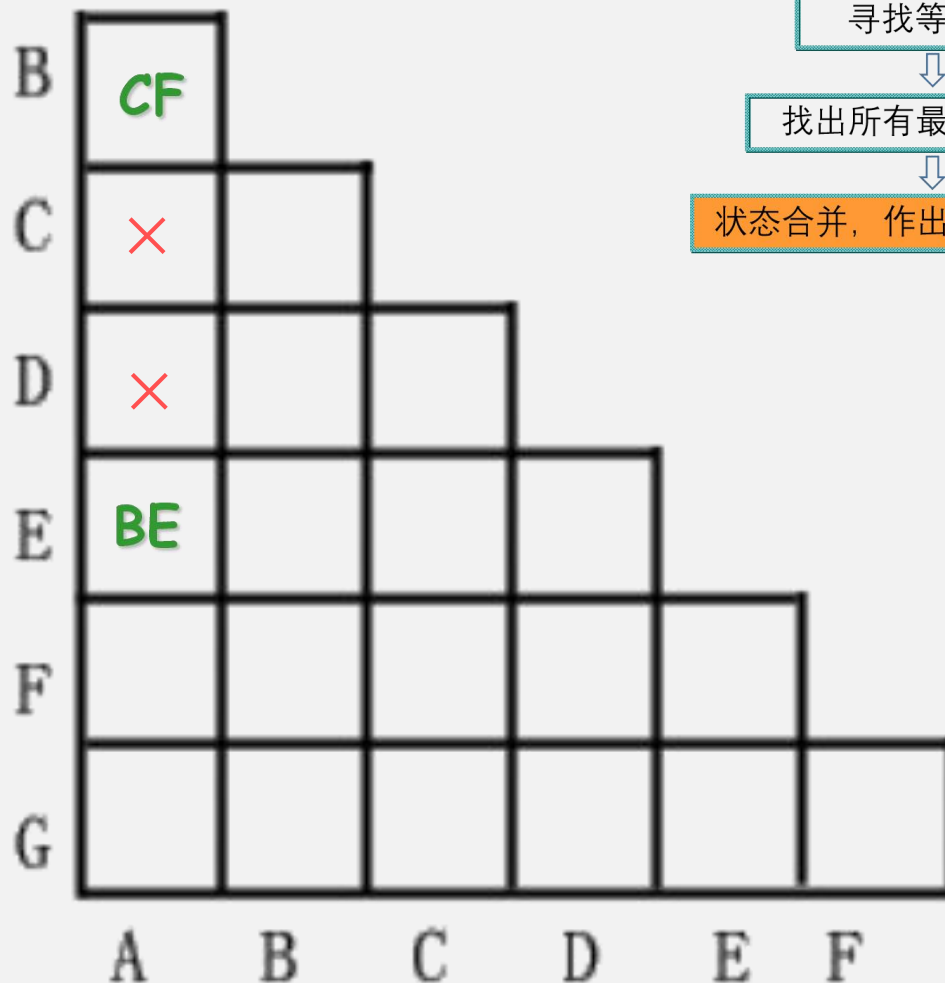
2. 完全确定状态表的化简

4) 化简过程

例1 化简下表所示原始状态表

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



作隐含表

寻找等效对

找出所有最大等效类

状态合并，作出最小化状态表

6.7 同步时序逻辑电路设计基础

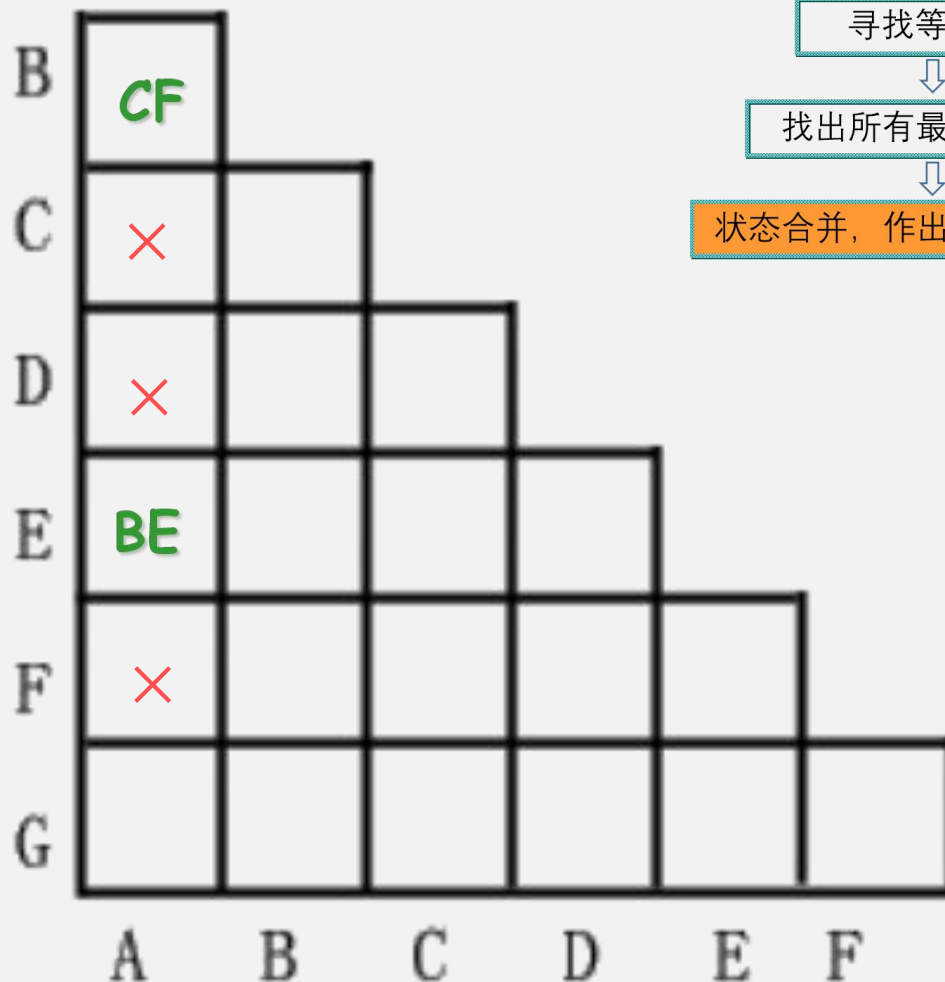
2. 完全确定状态表的化简

4) 化简过程

例1 化简下表所示原始状态表

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



作隐含表

寻找等效对

找出所有最大等效类

状态合并，作出最小化状态表

6.7 同步时序逻辑电路设计基础

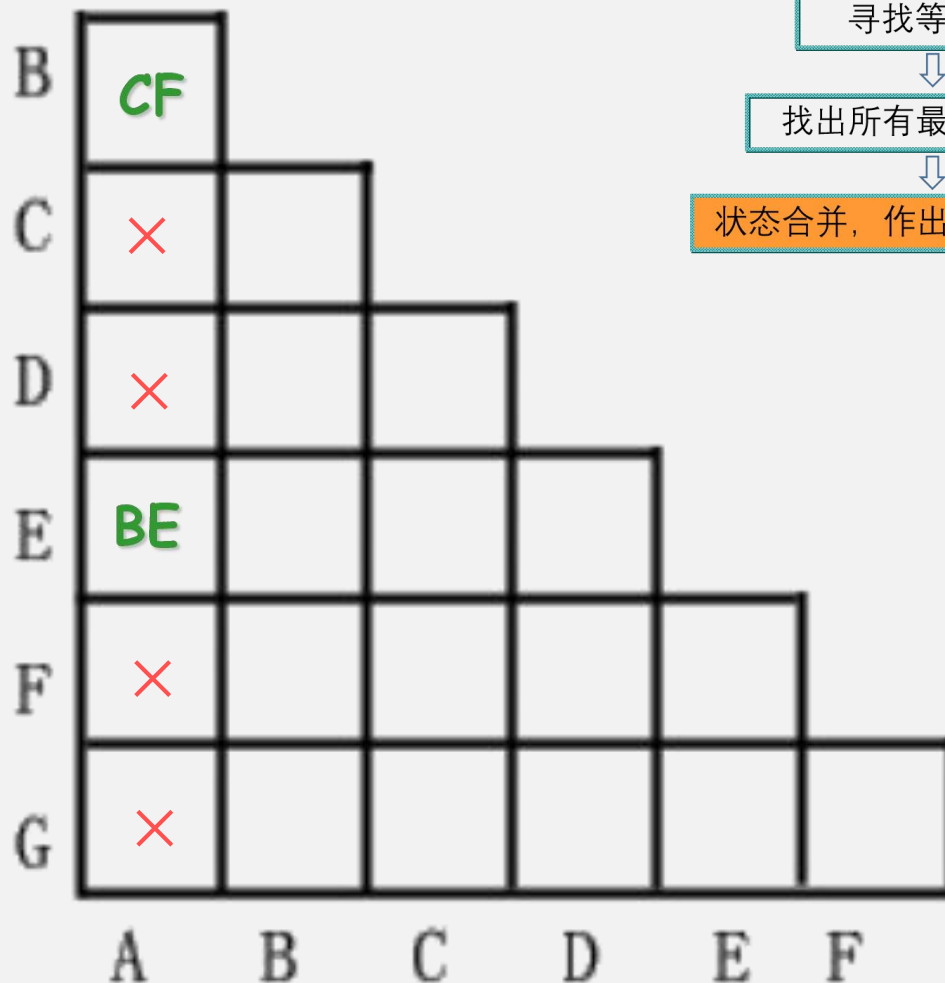
2. 完全确定状态表的化简

4) 化简过程

例1 化简下表所示原始状态表

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



作隐含表

寻找等效对

找出所有最大等效类

状态合并，作出最小化状态表

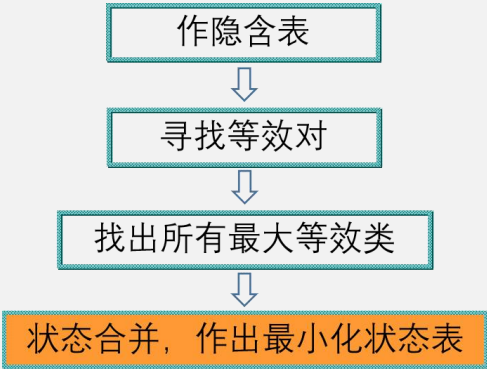
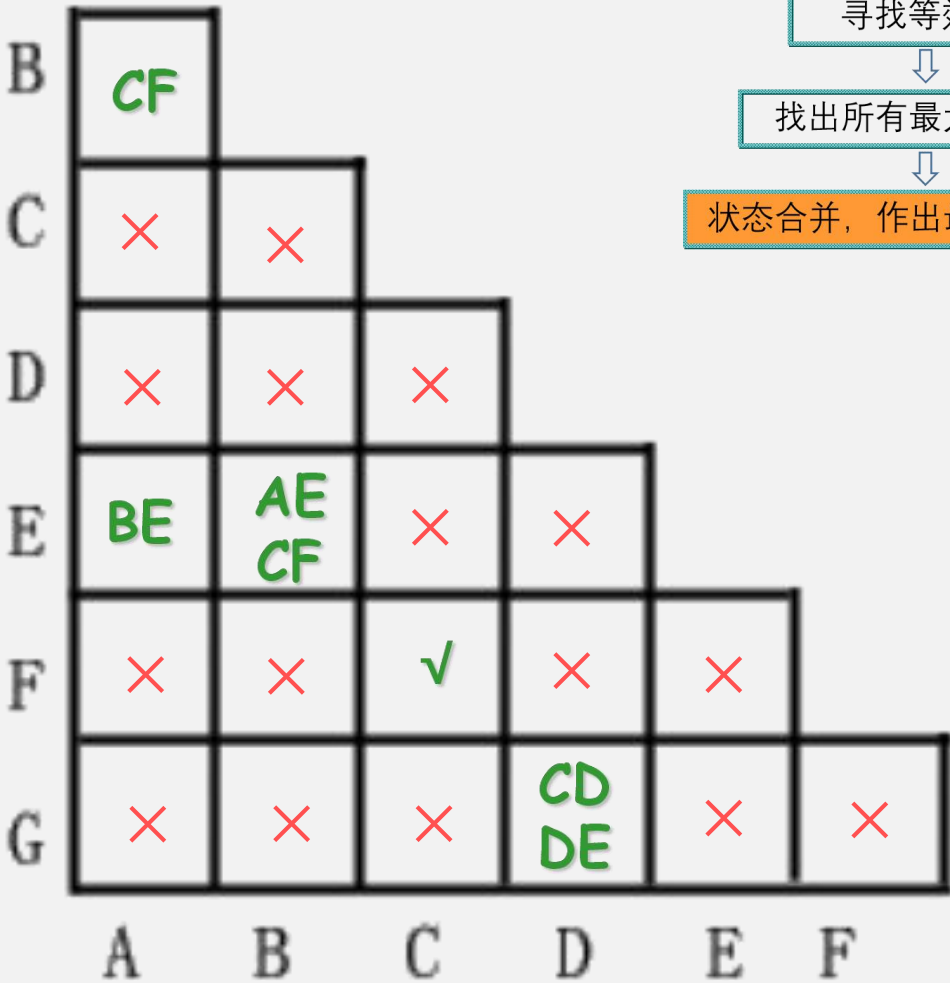
6.7 同步时序逻辑电路设计基础

2. 完全确定状态表的化简

4)化简过程

例1 化简下表所示原始状态表

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

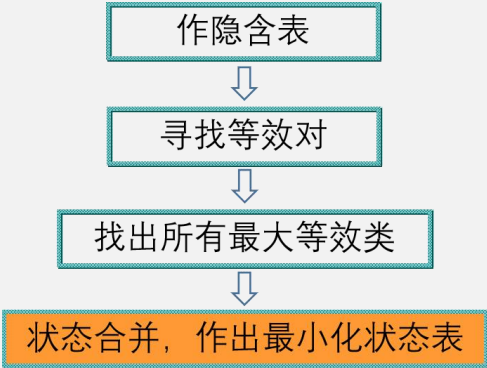
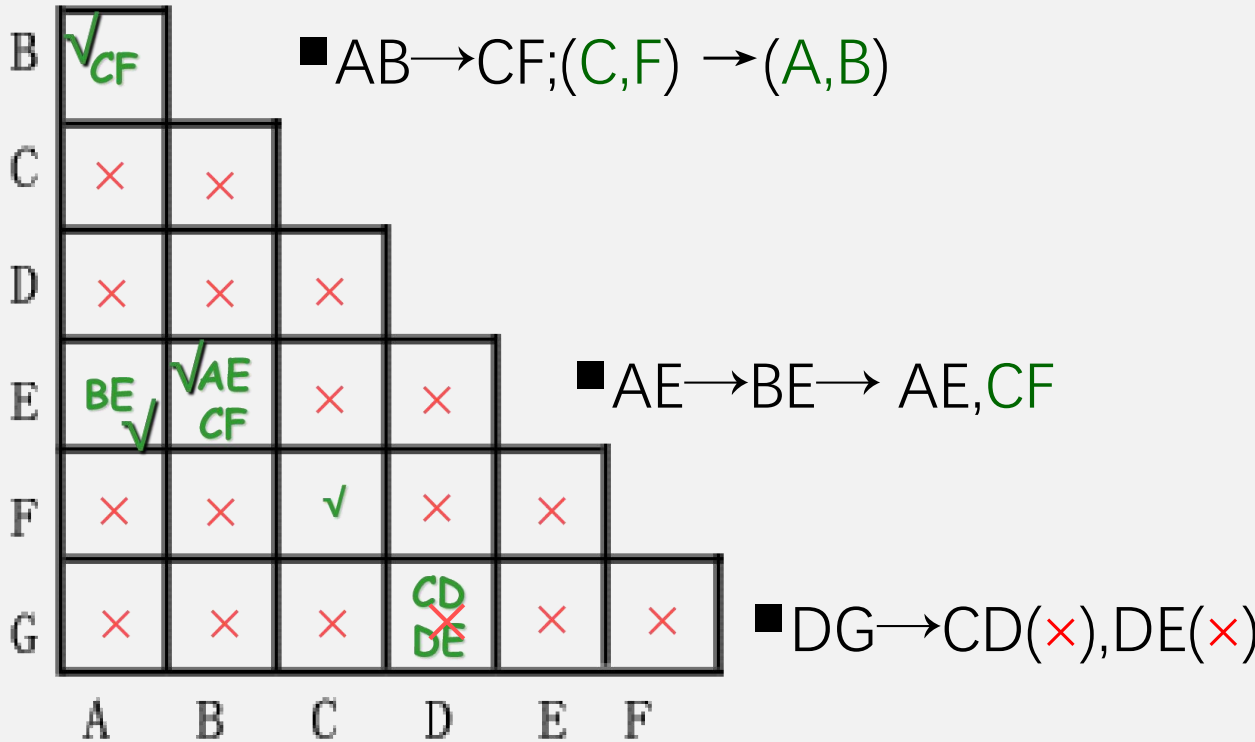


6.7 同步时序逻辑电路设计基础

2. 完全确定状态表的化简

4)化简过程

例1 化简下表所示原始状态表



原始状态表中共有四个等效对



(A, B)、(A, E)、(B, E)、(C, F)

6.7 同步时序逻辑电路设计基础

2. 完全确定状态表的化简

4)化简过程

例1 化简下表所示原始状态表

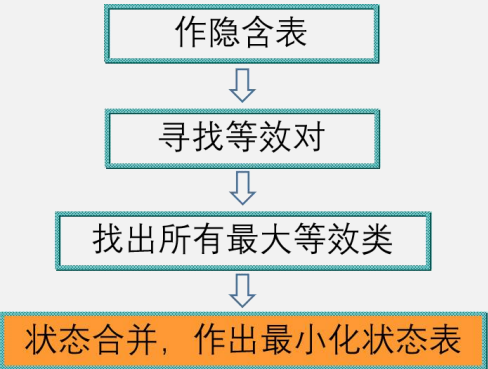
原始状态表		
现 态	次态 / 输出	
	x=0	x=1
A	C/0	B/1
B	F/0	A/1
C	F/0	G/0
D	D/1	E/0
E	C/0	E/1
F	C/0	G/0
G	C/1	D/0

$(A, B), (A, E), (B, E) \longrightarrow \{A, B, E\}$

$(C, F) \longrightarrow \{C, F\}$

$(D) \longrightarrow \{D\}$

$(G) \longrightarrow \{G\}$



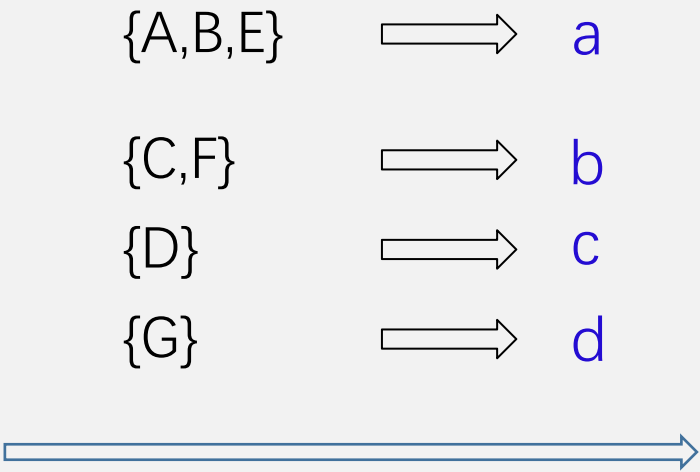
6.7 同步时序逻辑电路设计基础

2. 完全确定状态表的化简

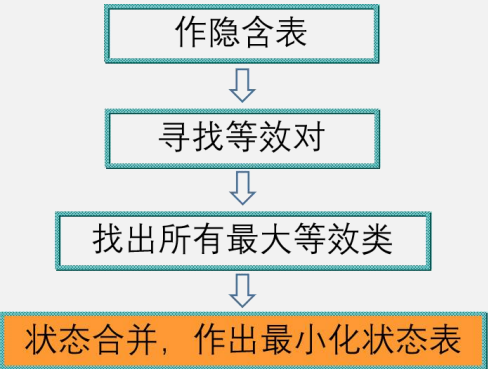
4)化简过程

例1 化简下表所示原始状态表

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



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



3. 完全确定状态表的编码

1) 确定二进制代码的位数/寄存器个数

$$m = \lceil \log_2 N \rceil$$

2) 确定状态分配方案(相邻分配法)

尽可能使激励函数和输出函数在卡诺图上的“1”方格处在相邻位置。

(1) 次态相同，现态相邻。

相同输入条件下，次态相同的现态分配相邻二进制代码；

(2) 同一现态，次态相邻。

同一现态在相邻输入条件下的次态分配相邻二进制代码；

初态分配 “0”

(3) 输出相同，现态相邻。

所有输入均具有相同输出的现态分配相邻二进制代码。

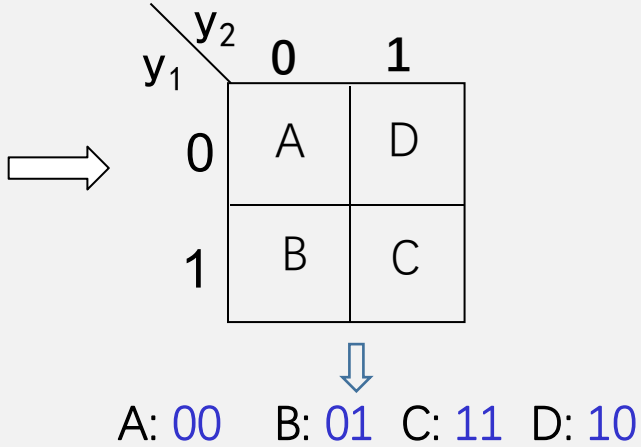
6.7 同步时序逻辑电路设计基础

3. 完全确定状态表的编码

例2 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

- 原则①：BC相邻；
原则②：BC、AD、CD相邻；
原则③：AD相邻。



6.7 同步时序逻辑电路设计基础

3. 完全确定状态表的编码

例2 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

→

现态 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

$y_1 \backslash y_2$	0	1
0	A	D
1	B	C

A: 00 B: 01 C: 11 D: 10

4. 确定激励函数和输出函数并画出逻辑电路图

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最简表达式：

- 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。

熟练时可直接根据作出激励函数和输出函数卡诺图化简。

6.7 同步时序逻辑电路设计基础

4. 确定激励函数和输出函数并画出逻辑电路图

四种时钟控制触发器的激励表如下：

R-S 触发器激励表			
$Q \rightarrow Q^{n+1}$		R	S
0	0	d	0
0	1	0	1
1	0	1	0
1	1	0	d

D触发器激励表		
$Q \rightarrow Q^{n+1}$		D
0	0	0
0	1	1
1	0	0
1	1	1

J-K 触发器激励表			
$Q \rightarrow Q^{n+1}$		J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

T触发器激励表		
$Q \rightarrow Q^{n+1}$		T
0	0	0
0	1	1
1	0	1
1	1	0

6.7 同步时序逻辑电路设计基础

4. 确定激励函数和输出函数并画出逻辑电路图

例3 用JK触发器和逻辑门设计同步时序逻辑电路，实现下表对应的功能。

现态 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

J-K 触发器激励表			
$Q \rightarrow Q^{n+1}$		J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

输 入 X	现 态 $y_2 \ y_1$		次 态 $y_2^{(n+1)}y_1^{(n+1)}$		激励函数 $J_2 \ K_2 \ J_1 \ K_1$				输出 Z
	y_2	y_1	$y_2^{(n+1)}$	$y_1^{(n+1)}$	J_2	K_2	J_1	K_1	
0	0	0	1	1	1	d	1	d	0
0	0	1	0	0	0	d	d	1	0
0	1	0	0	1	d	1	1	d	0
0	1	1	0	0	d	1	d	1	1
1	0	0	0	1	0	d	1	d	0
1	0	1	0	0	0	d	d	1	1
1	1	0	1	1	d	0	1	d	0
1	1	1	1	0	d	0	d	1	1

6.7 同步时序逻辑电路设计基础

4. 确定激励函数和输出函数并画出逻辑电路图

例3 用JK触发器和逻辑门设计同步时序逻辑电路，实现下表对应的功能。

输入 X	现 态 $y_2 y_1$		次 态 $y_2^{(n+1)} y_1^{(n+1)}$		激励函数				输出 Z
	y_2	y_1	$y_2^{(n+1)}$	$y_1^{(n+1)}$	J_2	K_2	J_1	K_1	
0	0	0	1	1	1	d	1	d	0
0	0	1	0	0	0	d	d	1	0
0	1	0	0	1	d	1	1	d	0
0	1	1	0	0	d	1	d	1	1
1	0	0	0	1	0	d	1	d	0
1	0	1	0	0	0	d	d	1	1
1	1	0	1	1	d	0	1	d	0
1	1	1	1	0	d	0	d	1	1

		$y_2 y_1$				
X		00	01	11	10	
J_1	0	1	d	d	1	$J_1=1$
	1	1	d	d	1	

		$y_2 y_1$				
X		00	01	11	10	
K_1	0	d	1	1	d	$K_1=1$
	1	d	1	1	d	

6.7 同步时序逻辑电路设计基础

4. 确定激励函数和输出函数并画出逻辑电路图

例3 用JK触发器和逻辑门设计同步时序逻辑电路，实现下表对应的功能。

输 入 X	现 态 y ₂ y ₁		次 态		激励函数				输出 Z
			y ₂ ⁽ⁿ⁺¹⁾	y ₁ ⁽ⁿ⁺¹⁾	J ₂	K ₂	J ₁	K ₁	
0	0	0	1	1	1	d	1	d	0
0	0	1	0	0	0	d	d	1	0
0	1	0	0	1	d	1	1	d	0
0	1	1	0	0	d	1	d	1	1
1	0	0	0	1	0	d	1	d	0
1	0	1	0	0	0	d	d	1	1
1	1	0	1	1	d	0	1	d	0
1	1	1	1	0	d	0	d	1	1

Z	x	$y_2 y_1$			
		00	01	11	10
	0	0	0	1	0
	1	0	1	1	0

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

J_2	x	$y_2 y_1$			
		00	01	11	10
	0	1	0	d	d
	1	0	0	d	d

$$J_2 = \overline{x} \overline{y_1}$$

K_2	x	$y_2 y_1$			
		00	01	11	10
	0	d	d	1	1
	1	d	d	0	0

$$K_2 = \overline{x}$$

6.7 同步时序逻辑电路设计基础

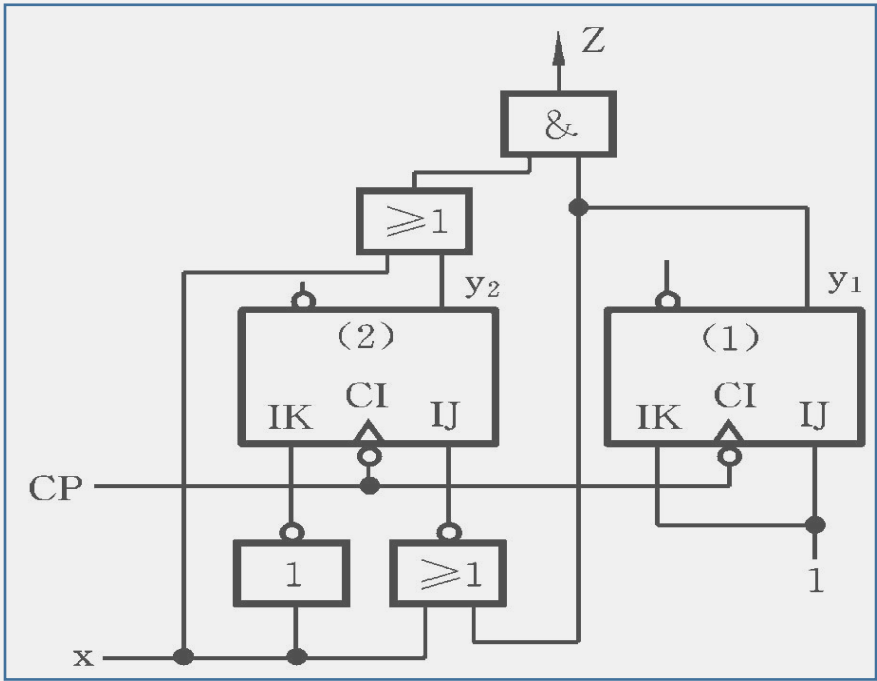
4. 确定激励函数和输出函数并画出逻辑电路图

例3 用JK触发器和逻辑门设计同步时序逻辑电路，实现下表对应的功能。

输入 X	现 态 $y_2 \ y_1$		次 态		激励函数				输出 Z
			$y_2^{(n+1)} y_1^{(n+1)}$		J_2	K_2	J_1	K_1	
0	0	0	1	1	1	d	1	d	0
0	0	1	0	0	0	d	d	1	0
0	1	0	0	1	d	1	1	d	0
0	1	1	0	0	d	1	d	1	1
1	0	0	0	1	0	d	1	d	0
1	0	1	0	0	0	d	d	1	1
1	1	0	1	1	d	0	1	d	0
1	1	1	1	0	d	0	d	1	1

$$J_1=1 \quad K_1=1 \quad J_2=\overline{\overline{x}y_1} \quad K_2=\overline{x}$$

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



4. 确定激励函数和输出函数并画出逻辑电路图

例3 用D触发器和逻辑门设计同步时序逻辑电路，实现下表对应的功能。

现态 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

D触发器激励表		
$Q \rightarrow Q^{n+1}$		D
0	0	0
0	1	1
1	0	0
1	1	1

输 入 X	现 态 $y_2 \ y_1$		次 态 $y_2^{(n+1)}y_1^{(n+1)}$	激励函数 $D_2 \ D_1$		输出 Z
0	0	0	1 1	1 1		0
0	0	1	0 0	0 0		0
0	1	0	0 1	0 1		0
0	1	1	0 0	0 0		1
1	0	0	0 1	0 1		0
1	0	1	0 0	0 0		1
1	1	0	1 1	1 1		0
1	1	1	1 0	1 0		1

6.7 同步时序逻辑电路设计基础

4. 确定激励函数和输出函数并画出逻辑电路图

例3 用D触发器和逻辑门设计同步时序逻辑电路，实现下表对应的功能。

输入 X	现 态 y ₂ y ₁		次 态 y ₂ ⁽ⁿ⁺¹⁾ y ₁ ⁽ⁿ⁺¹⁾		激励函数 D ₂ D ₁		输出 Z
	y ₂	y ₁	y ₂ ⁽ⁿ⁺¹⁾	y ₁ ⁽ⁿ⁺¹⁾	D ₂	D ₁	
0	0	0	1	1	1	1	0
0	0	1	0	0	0	0	0
0	1	0	0	1	0	1	0
0	1	1	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	1	0	0	0	0	1
1	1	0	1	1	1	1	0
1	1	1	1	0	1	0	1

D_2

x \ y ₂ y ₁				
	00	01	11	10
0	1	0	0	0
1	0	0	1	1

$$\begin{aligned} D_2 &= \overline{x} \cdot \overline{y_2} \cdot \overline{y_1} + xy_2 \\ &= \overline{x + y_2 + y_1} + xy_2 \end{aligned}$$

D_1

x \ y ₂ y ₁				
	00	01	11	10
0	1	0	0	1
1	1	0	0	1

$$D_1 = \overline{y_1}$$

6.7 同步时序逻辑电路设计基础

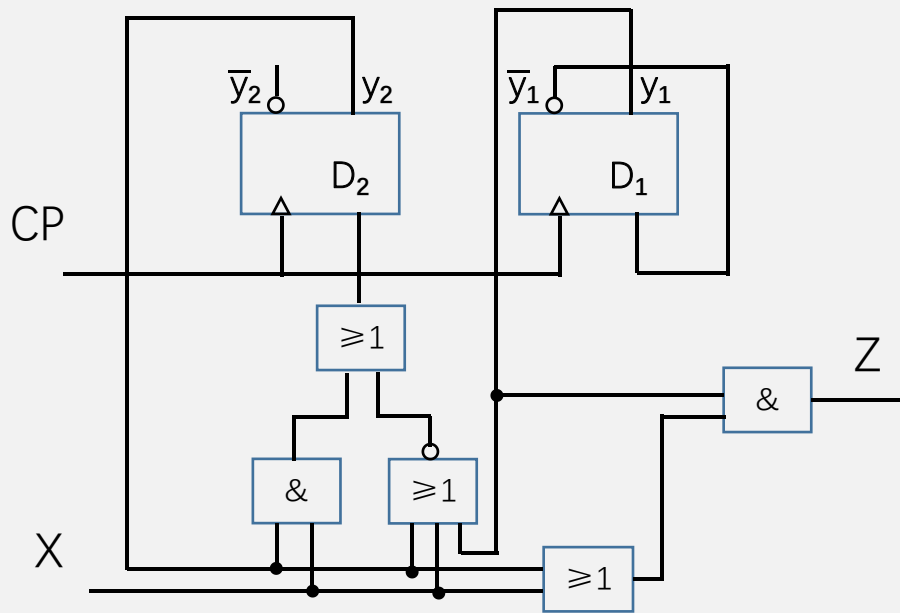
4. 确定激励函数和输出函数并画出逻辑电路图

例3 用D触发器和逻辑门设计同步时序逻辑电路，实现下表对应的功能。

输入 X	现 态 $y_2 \ y_1$		次 态 $y_2^{(n+1)} y_1^{(n+1)}$		激励函数 $D_2 \ D_1$		输出 Z
	y_2	y_1	$y_2^{(n+1)}$	$y_1^{(n+1)}$	D_2	D_1	
0	0	0	1	1	1	1	0
0	0	1	0	0	0	0	0
0	1	0	0	1	0	1	0
0	1	1	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	1	0	0	0	0	1
1	1	0	1	1	1	1	0
1	1	1	1	0	1	0	1

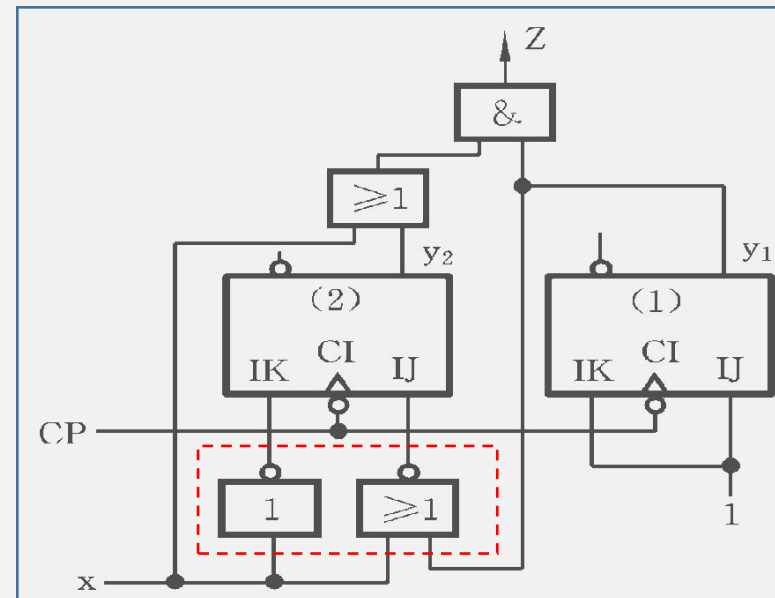
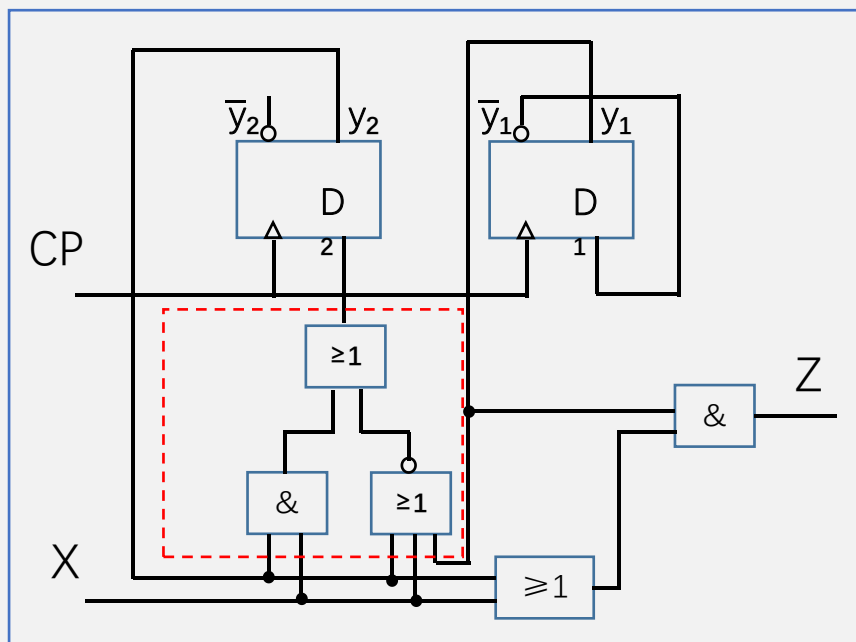
$$D_2 = \overline{\overline{x}} \cdot \overline{\overline{y_2}} \overline{\overline{y_1}} + x y_2$$
$$= x + y_2 + y_1 + x y_2$$
$$D_1 = \overline{y_1}$$

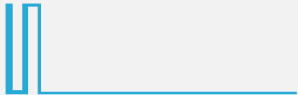
$$Z = y_2 y_1 + x y_1 = (y_2 + x) y_1$$



6.7 同步时序逻辑电路设计基础

输入 X	现 态 $y_2 y_1$	次 态		激励函数 $D_2 D_1$		输出 Z
		$y_2^{(n+1)}$	$y_1^{(n+1)}$			
0	0 0	1	1	1	1	0
0	0 1	0	0	0	0	0
0	1 0	0	1	0	1	0
0	1 1	0	0	0	0	1
1	0 0	0	1	0	1	0
1	0 1	0	0	0	0	1
1	1 0	1	1	1	1	0
1	1 1	1	0	1	0	1





第二部分完