

第六章 时序逻辑电路（一）

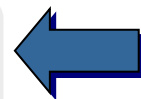
主讲教师：何云峰



提 纲

1

时序逻辑电路概述



2

同步时序逻辑电路分析

3

同步时序逻辑电路设计

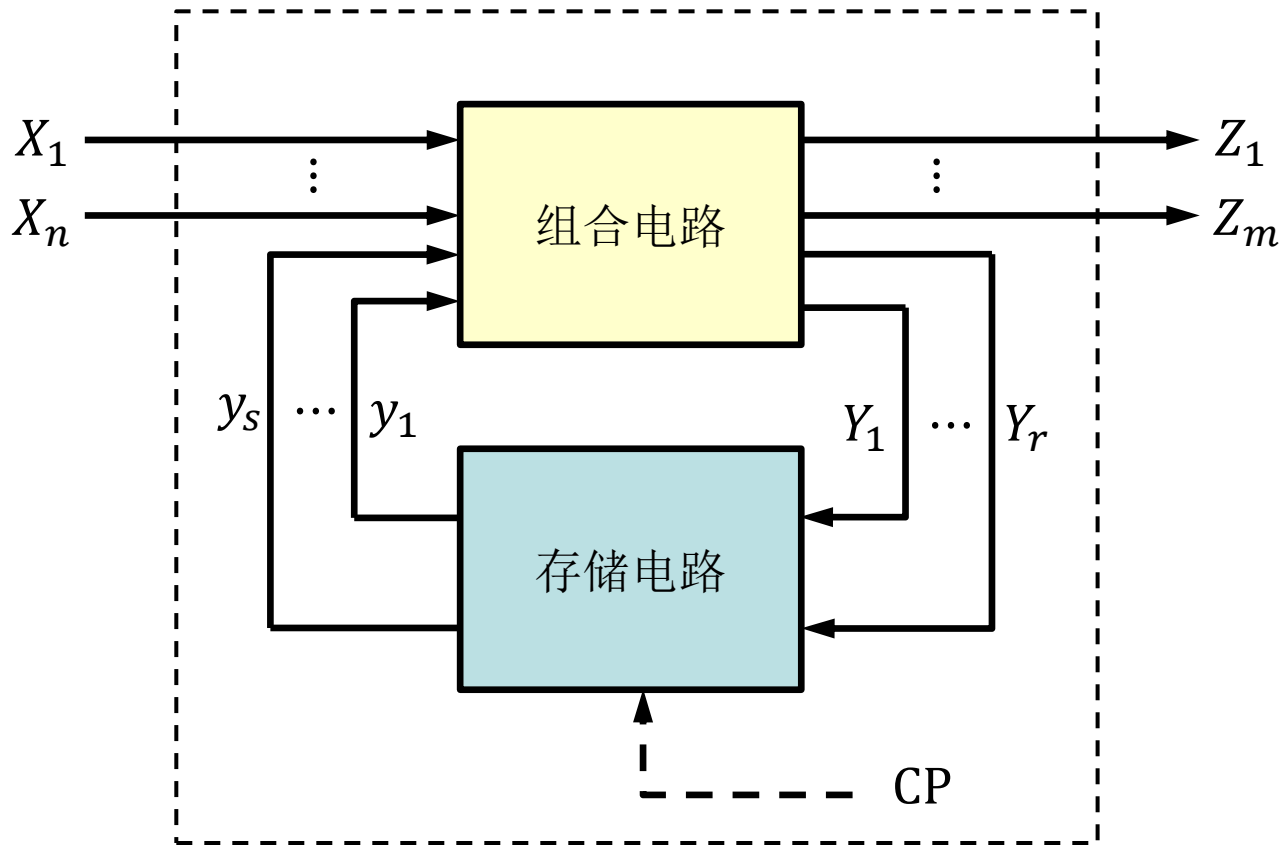
时序逻辑电路概述

□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。

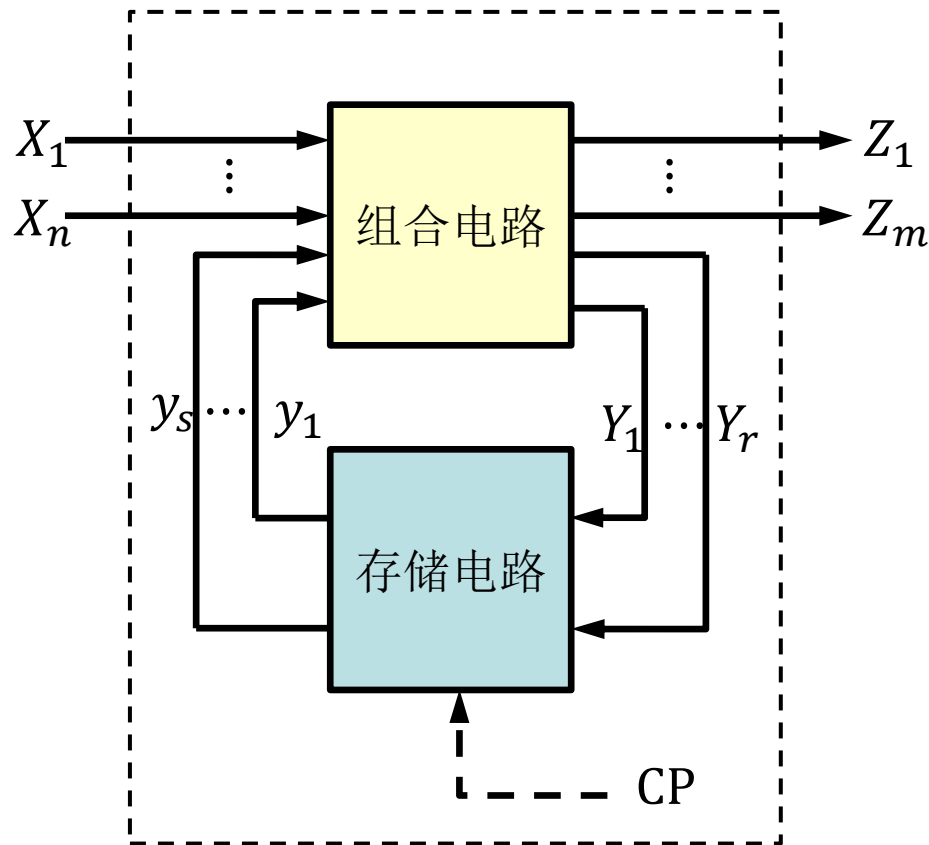
时序逻辑电路概述

□ 一般结构



时序逻辑电路概述

- 输入信号: X_1, \dots, X_n
- 输出信号: Z_1, \dots, Z_m
- “状态”: y_1, \dots, y_s
- 激励信号: Y_1, \dots, Y_r
- 时钟脉冲信号: CP
 - 是否存在取决于时序逻辑电路的类型



时序逻辑电路概述

□ 时序逻辑电路的状态

- 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
- 随着外部信号的作用而变化
- 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}

时序逻辑电路概述

□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(过去的输入)共同决定

时序逻辑电路概述

□ 时序逻辑电路的分类

- 电路的工作方式
- 电路输出对输入的依从关系
- 输入信号的形式

时序逻辑电路概述

□ 按照电路的工作方式分类

– 同步时序逻辑电路

- 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

– 异步时序逻辑电路

- 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

时序逻辑电路概述

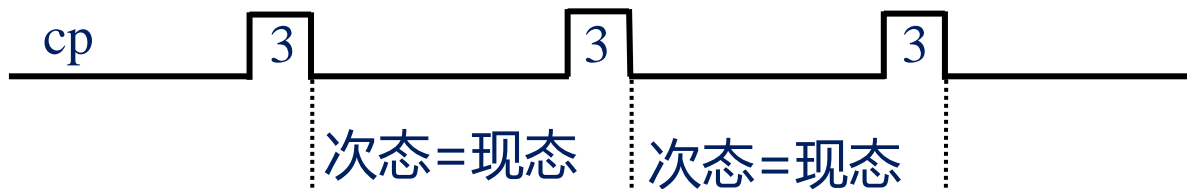
□ 同步时序逻辑电路

- 电路状态的改变依赖于输入信号和时钟脉冲信号
 - 状态变化时间
 - 取决于时钟信号
 - 状态如何变化
 - 取决于输入信号
 - 每个状态维持时间
 - 取决于时钟脉冲的周期
- 在研究同步时序逻辑电路时，通常不把同步时钟信号作为输入信号处理，而是将它当成一种默认的时间基准

时序逻辑电路概述

□ 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



时序逻辑电路概述

□ 同步时序逻辑电路对时钟的要求

- 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

时序逻辑电路概述

□ 异步时序逻辑电路

- 存储电路可由触发器或延时元件组成
- 将在第六章第二部分详细讲解

时序逻辑电路概述

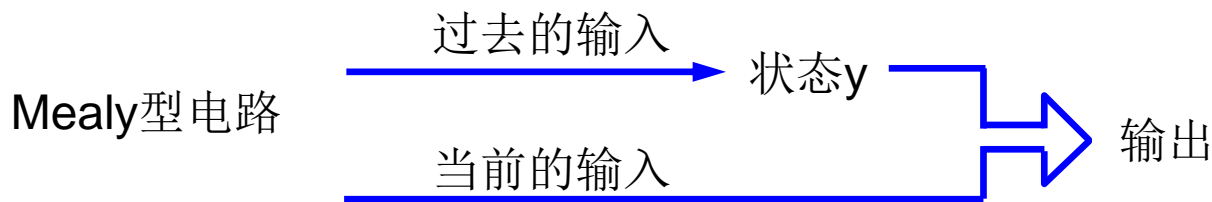
□按照电路输出与输入的关系分类

– Mealy型

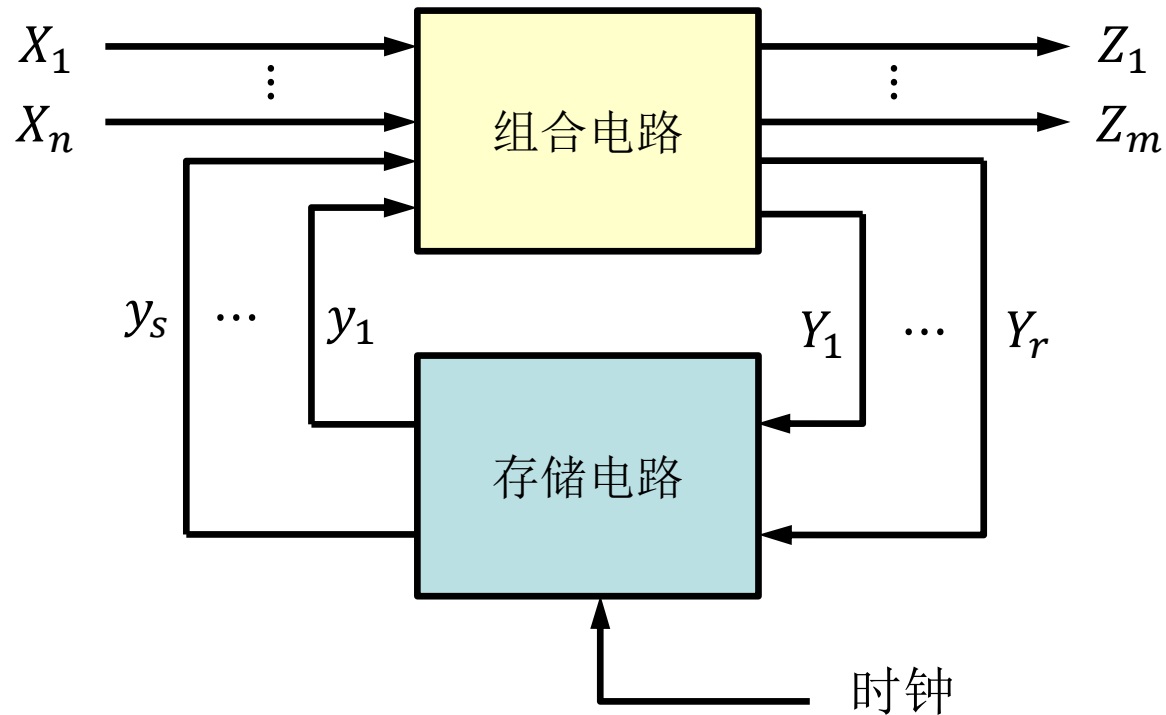
- 时序逻辑电路的输出是电路输入和电路状态的函数

– Moore型

- 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

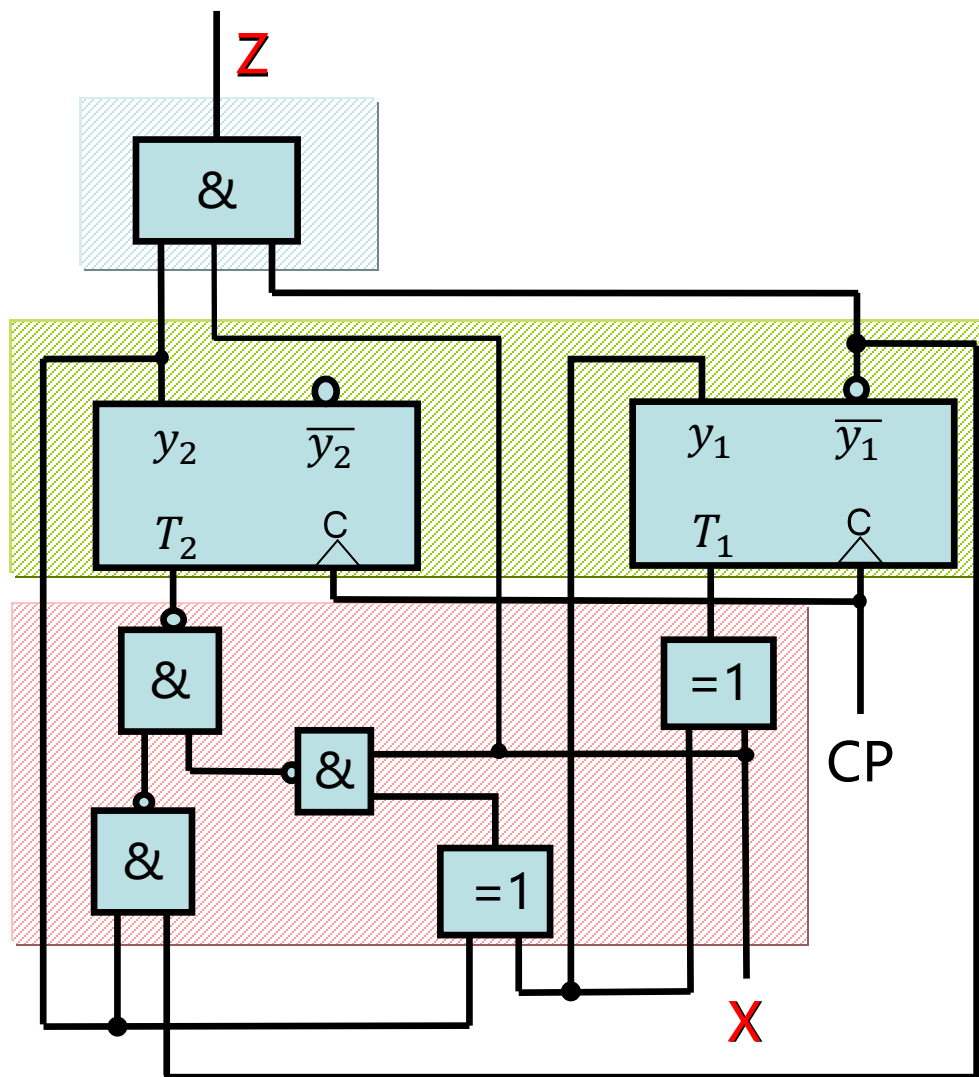


时序逻辑电路概述

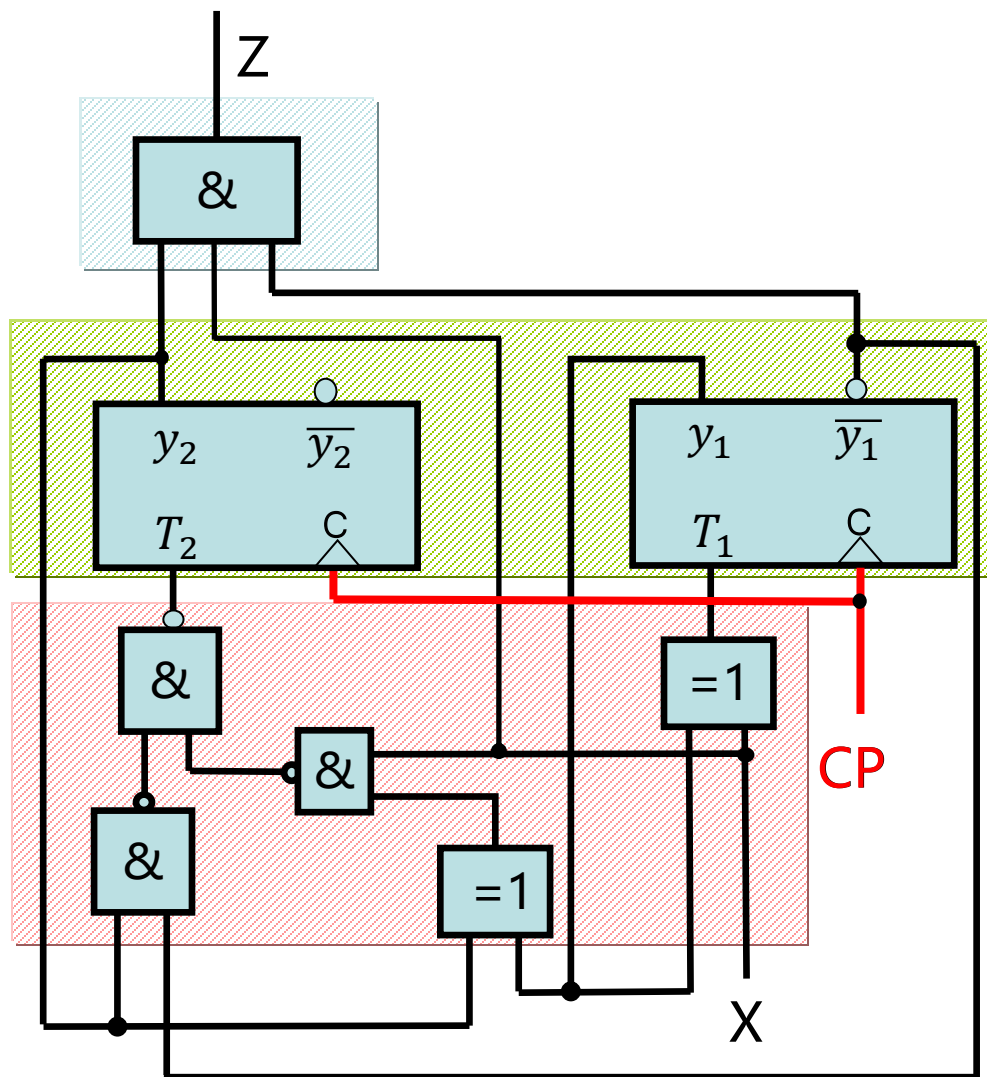


Mealy型电路

时序逻辑电路概述

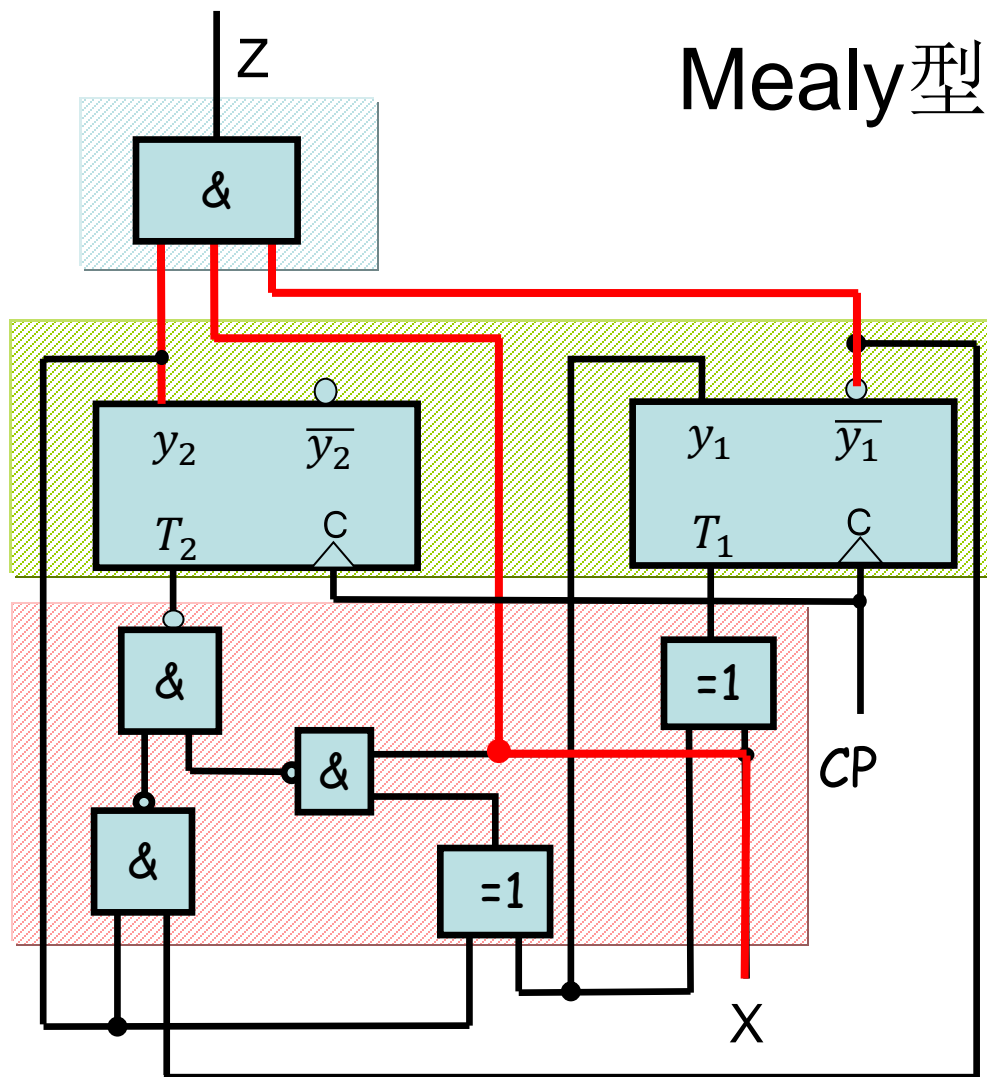


时序逻辑电路概述

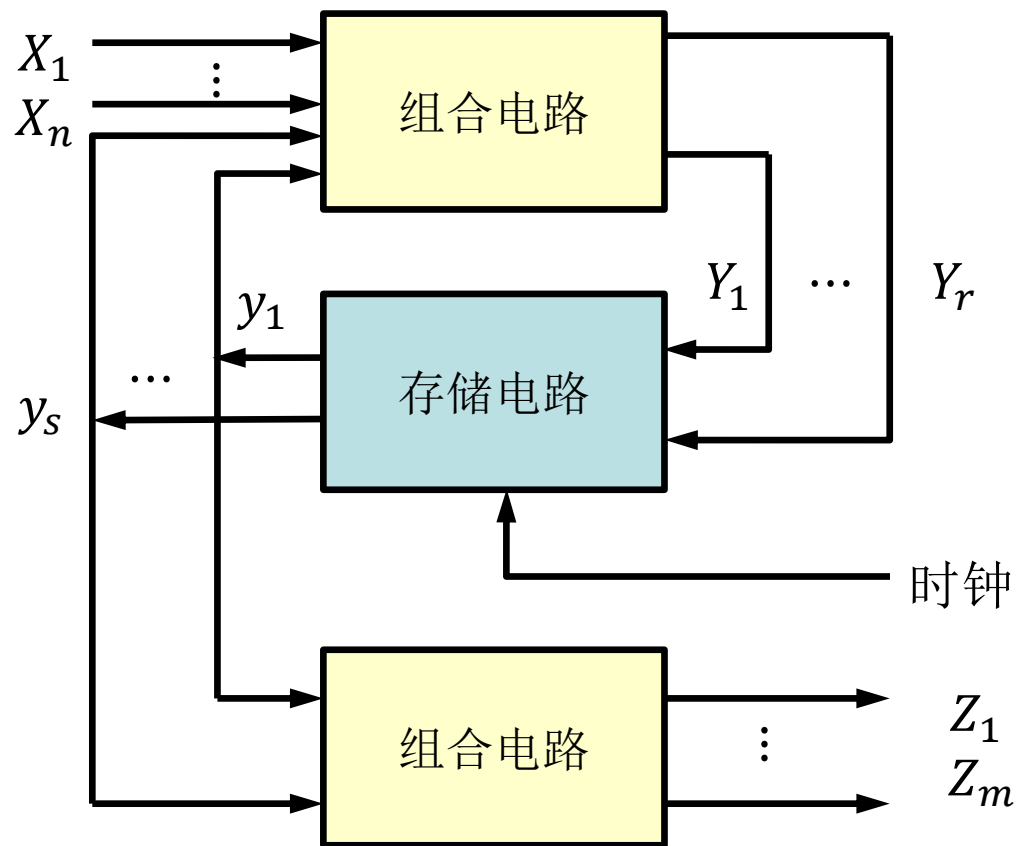


时序逻辑电路概述

Mealy型

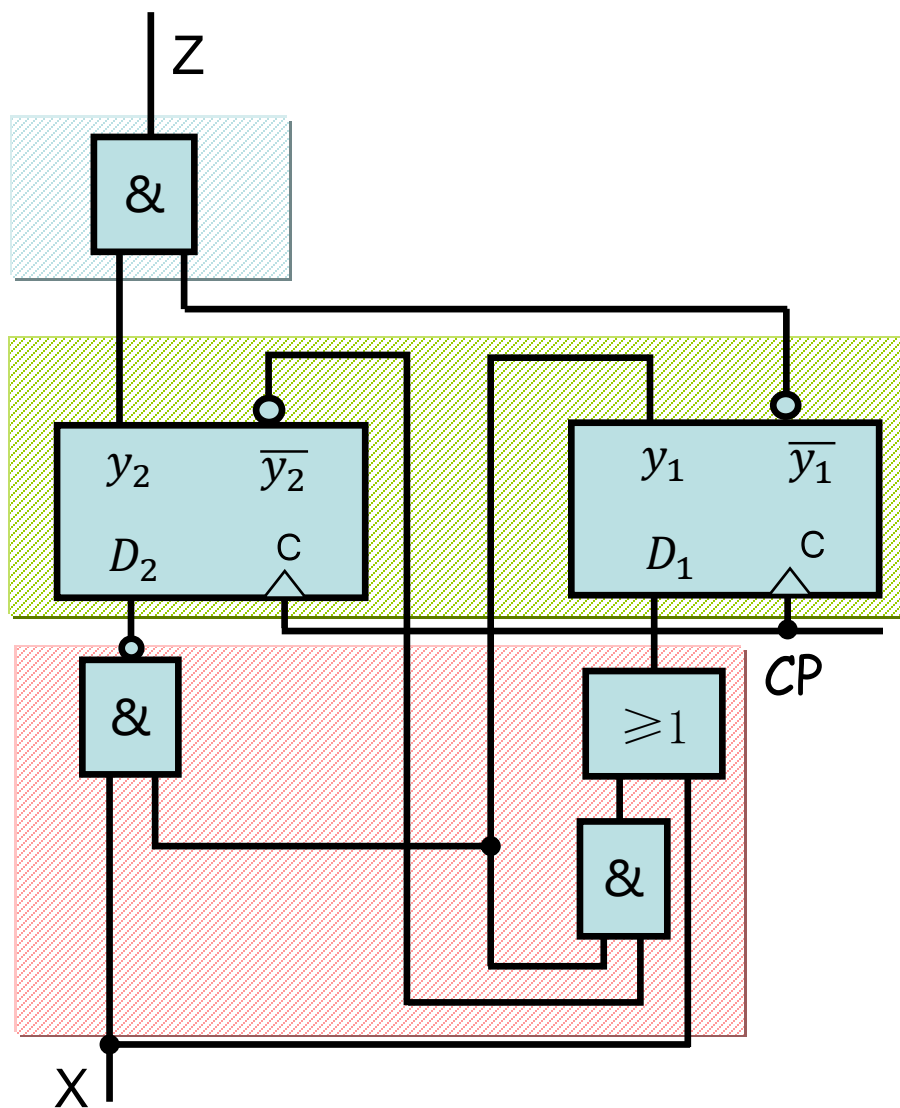


时序逻辑电路概述

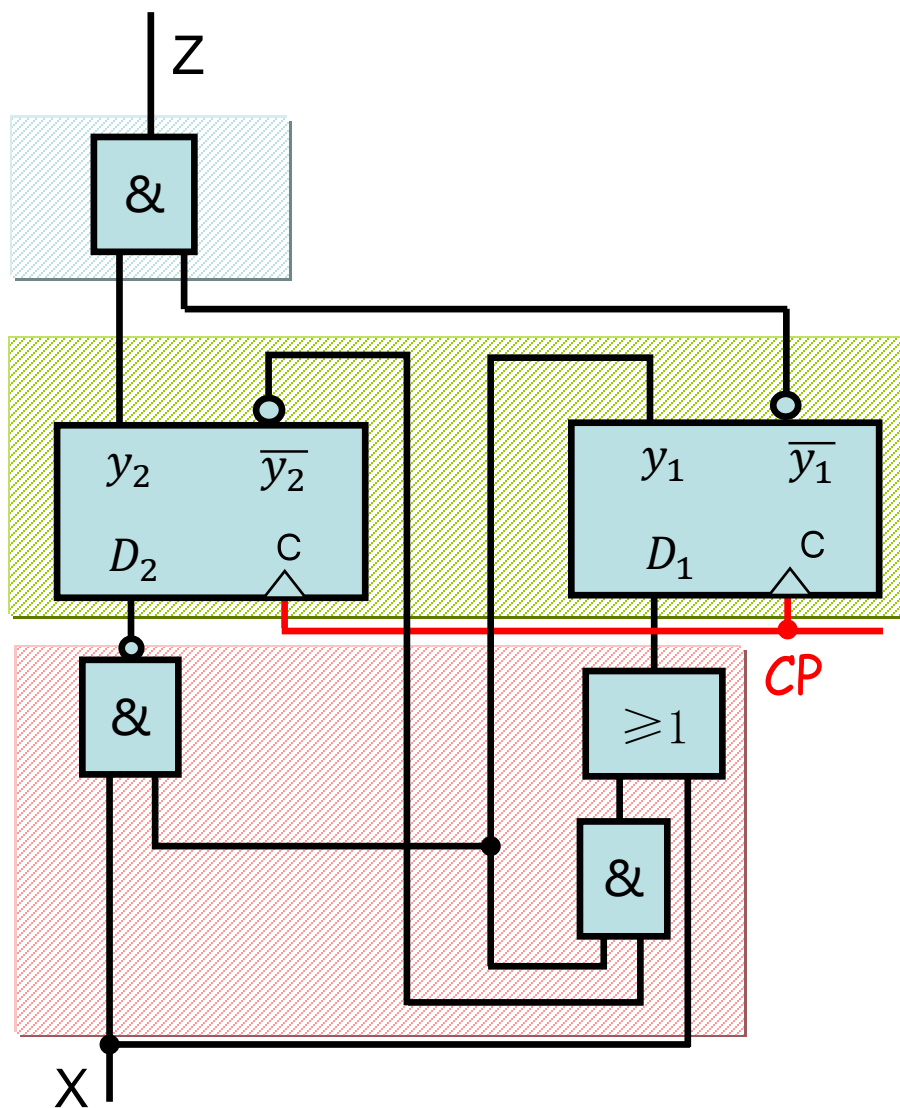


Moore型电路

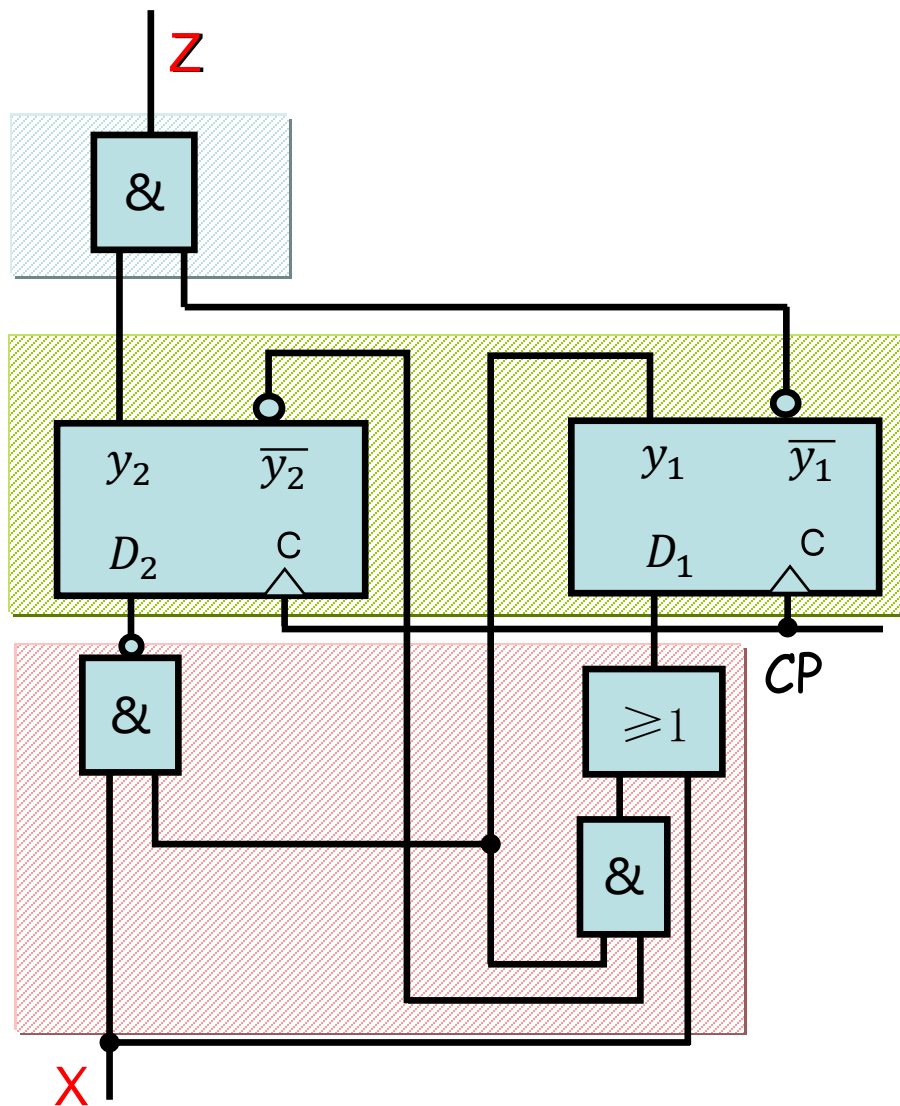
时序逻辑电路概述



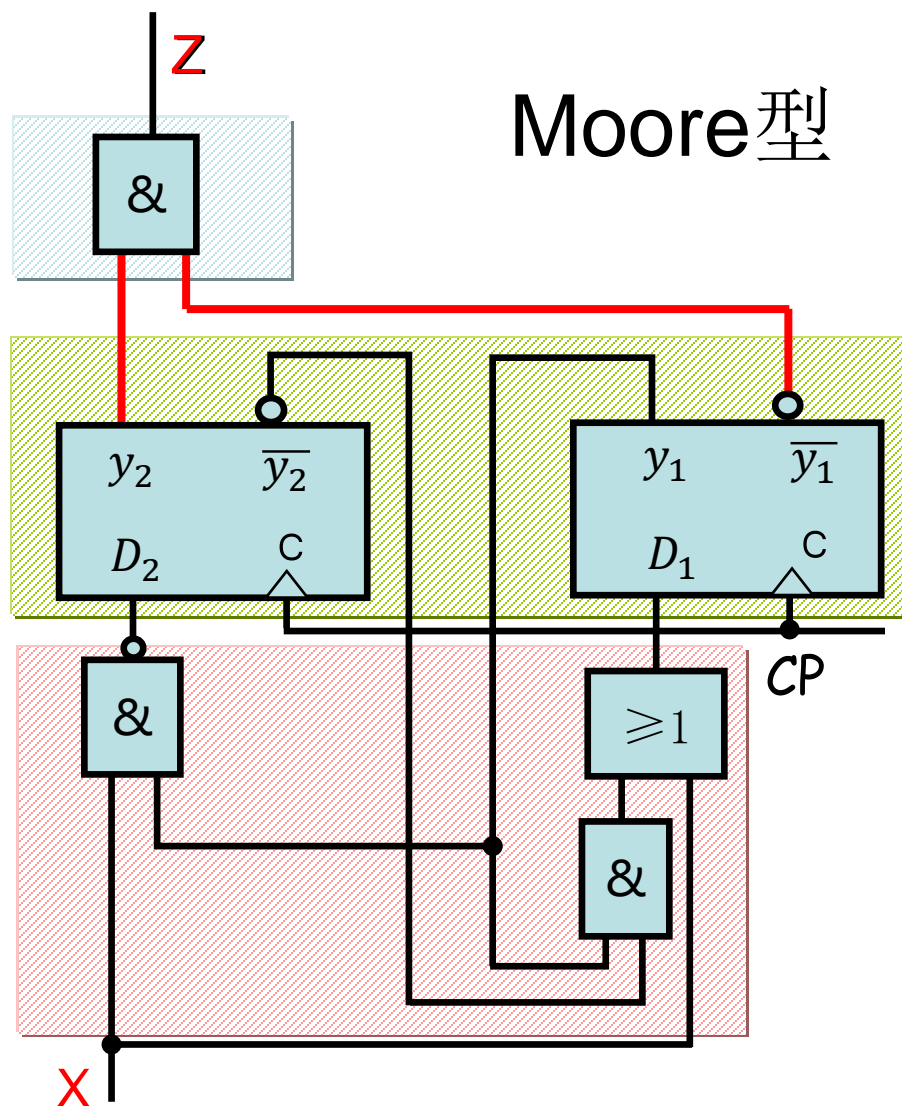
时序逻辑电路概述



时序逻辑电路概述



时序逻辑电路概述



时序逻辑电路概述

□ 区别

- Mealy型电路的输入和输出之间存在直接联系
- Moore型电路则是将全部输入转换成电路状态后再和输出建立联系

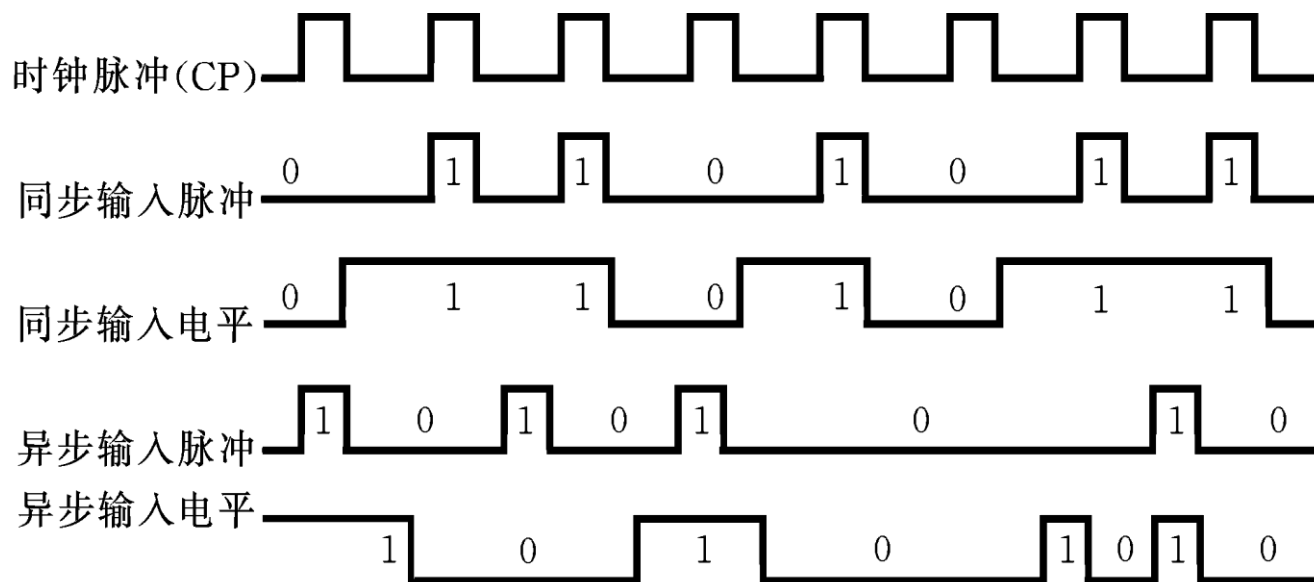
□ Moore型电路的特殊情况

- 时序逻辑电路没有专门的外部输出信号，而是以电路状态作为输出

时序逻辑电路概述

□按照电路输入信号的形式划分

- 脉冲型
- 电平型



时序逻辑电路概述

□ 典型的同步时序逻辑电路

- 锁存器
- 寄存器
- 移位寄存器
- 计数器
- 序列信号发生器

同步时序逻辑电路设计

□ 时序逻辑电路的描述

- 逻辑函数表达式
- 状态图
- 状态表
- 时间图

同步时序逻辑电路设计

□ 逻辑函数表达式

– 输出函数表达式

- 反映电路输出与外部输入和状态之间关系

– 激励函数表达式

- 反映了存储电路的输入与外部输入和电路状态之间的关系

– 次态函数表达式

- 同步时序电路的次态与激励函数和电路现态之间的关系
- 与触发器类型相关

函数表达式

□ Moore型电路

– 输出函数表达式

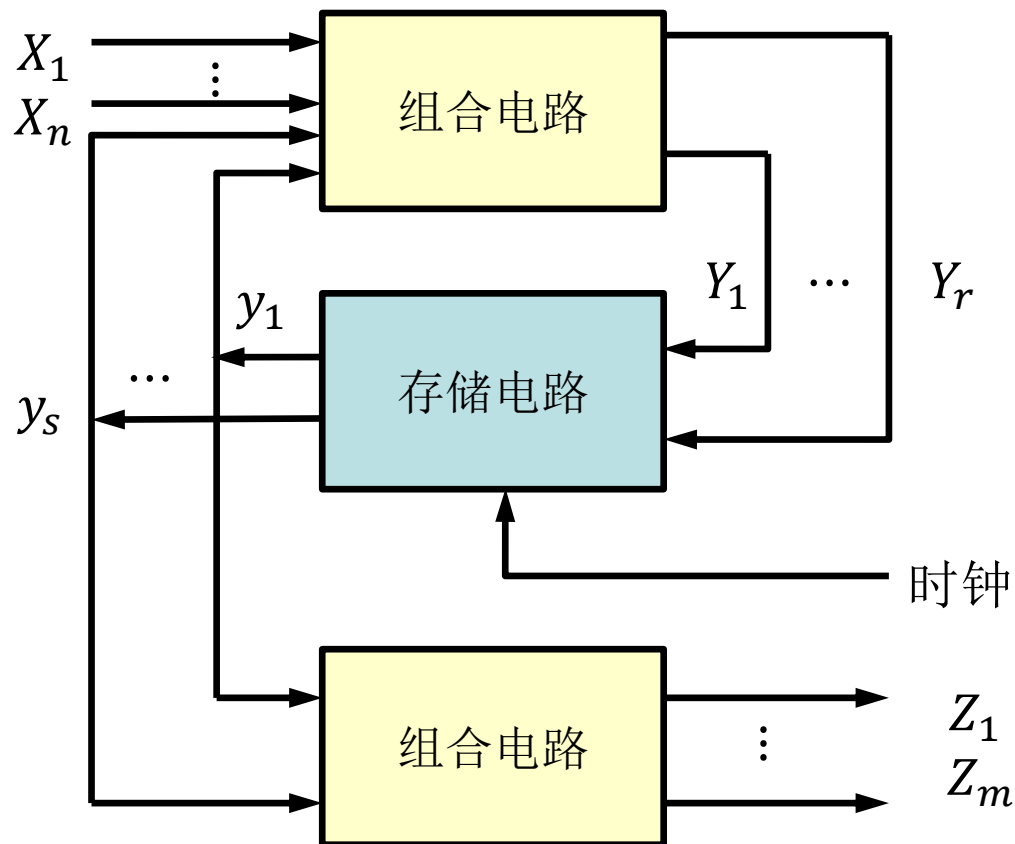
- $Z = f(y)$

– 激励函数表达式

- $Y = g(x, y)$

– 次态函数表达式

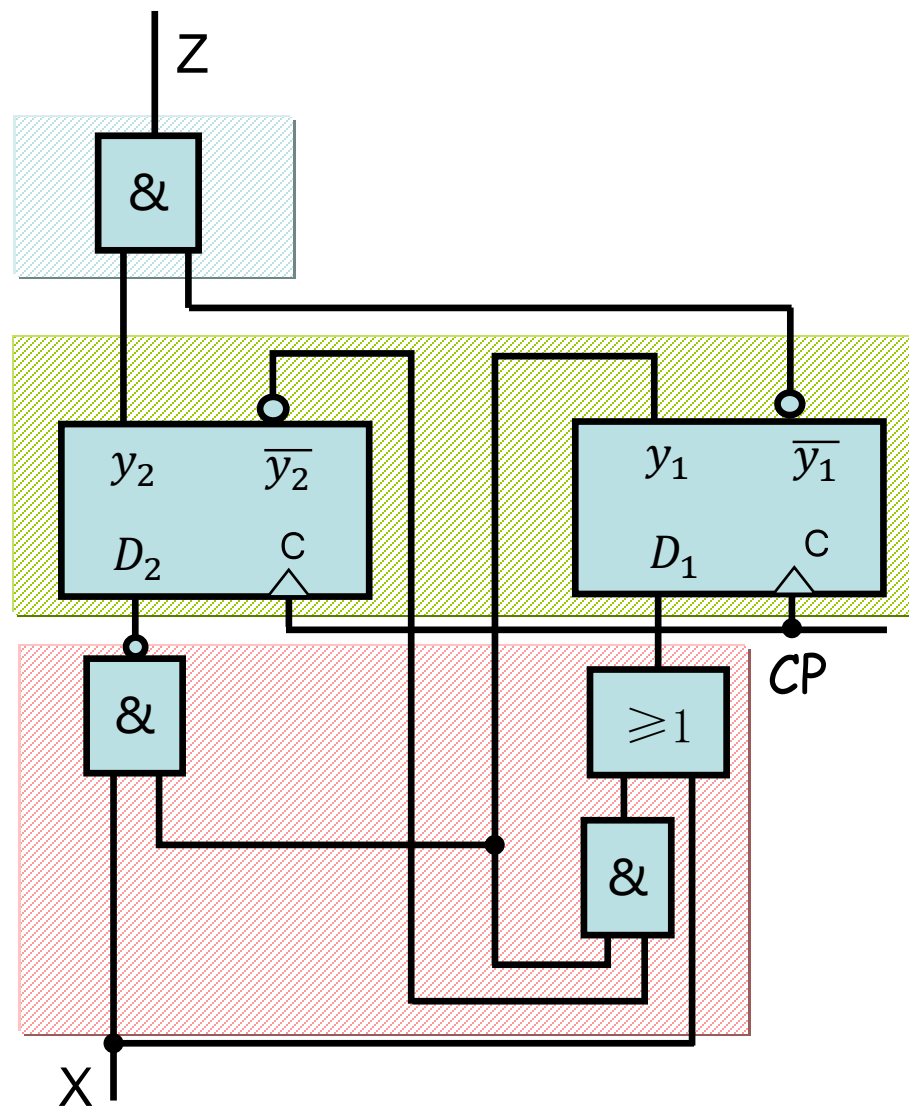
- $y^{n+1} = k(Y, y)$



函数表达式

□ Moore型电路

– 输出函数表达式



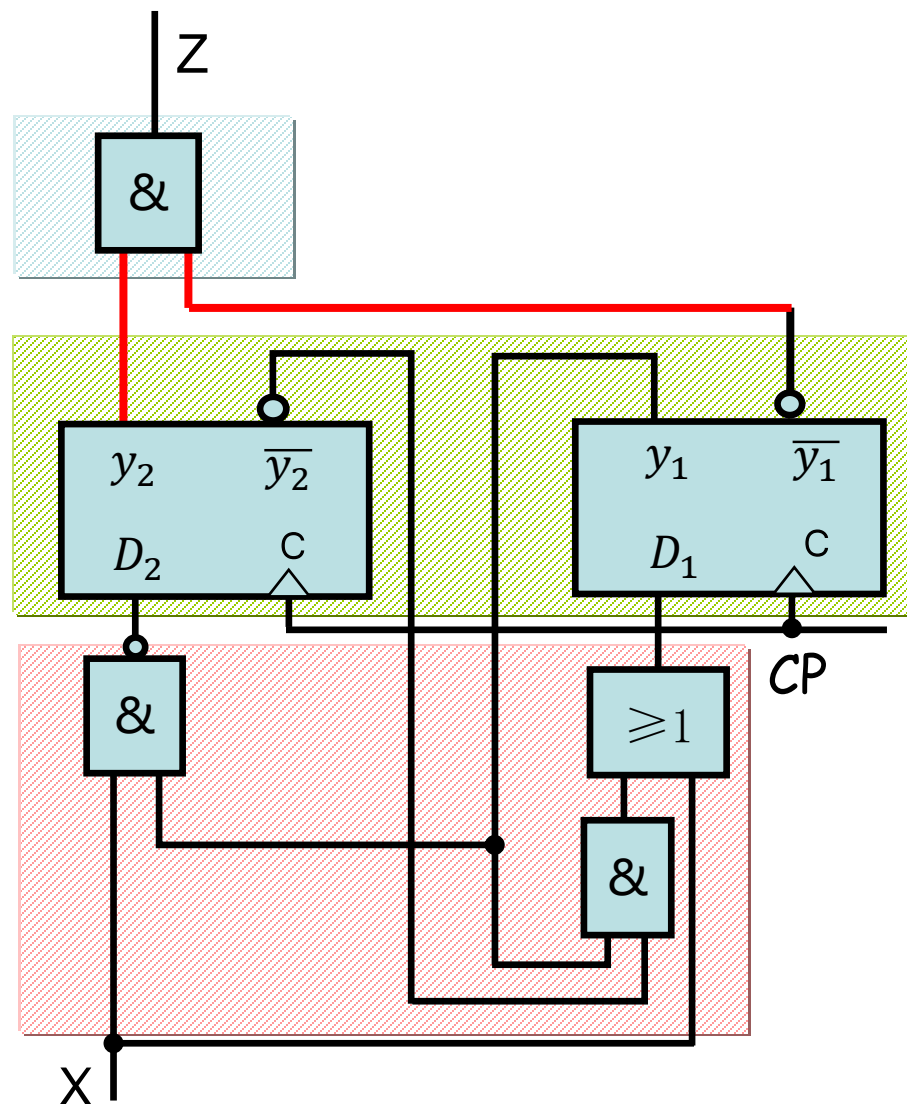
函数表达式

□ Moore型电路

– 输出函数表达式

- $Z = y_2 \overline{y_1}$

– 激励函数表达式



函数表达式

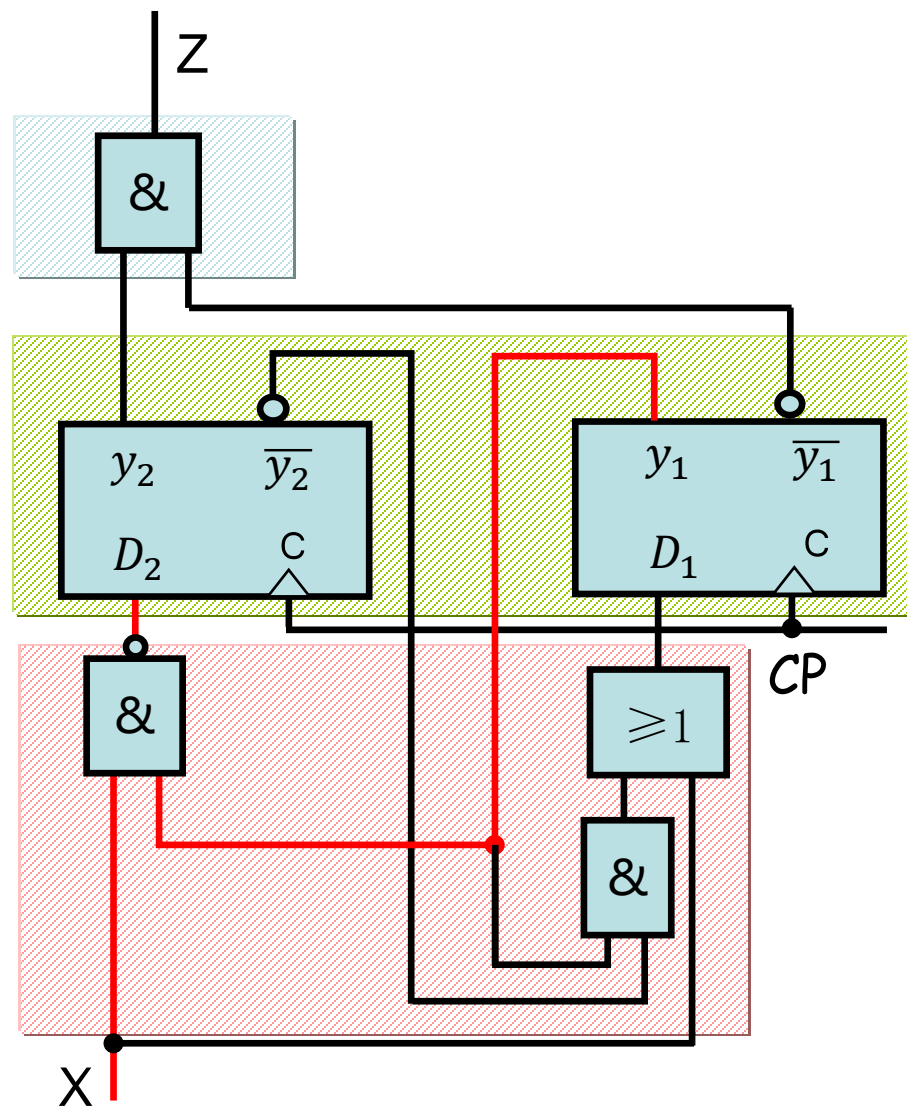
□ Moore型电路

– 输出函数表达式

- $Z = y_2 \overline{y_1}$

– 激励函数表达式

- $D_2 = \overline{x} y_1$



函数表达式

□ Moore型电路

– 输出函数表达式

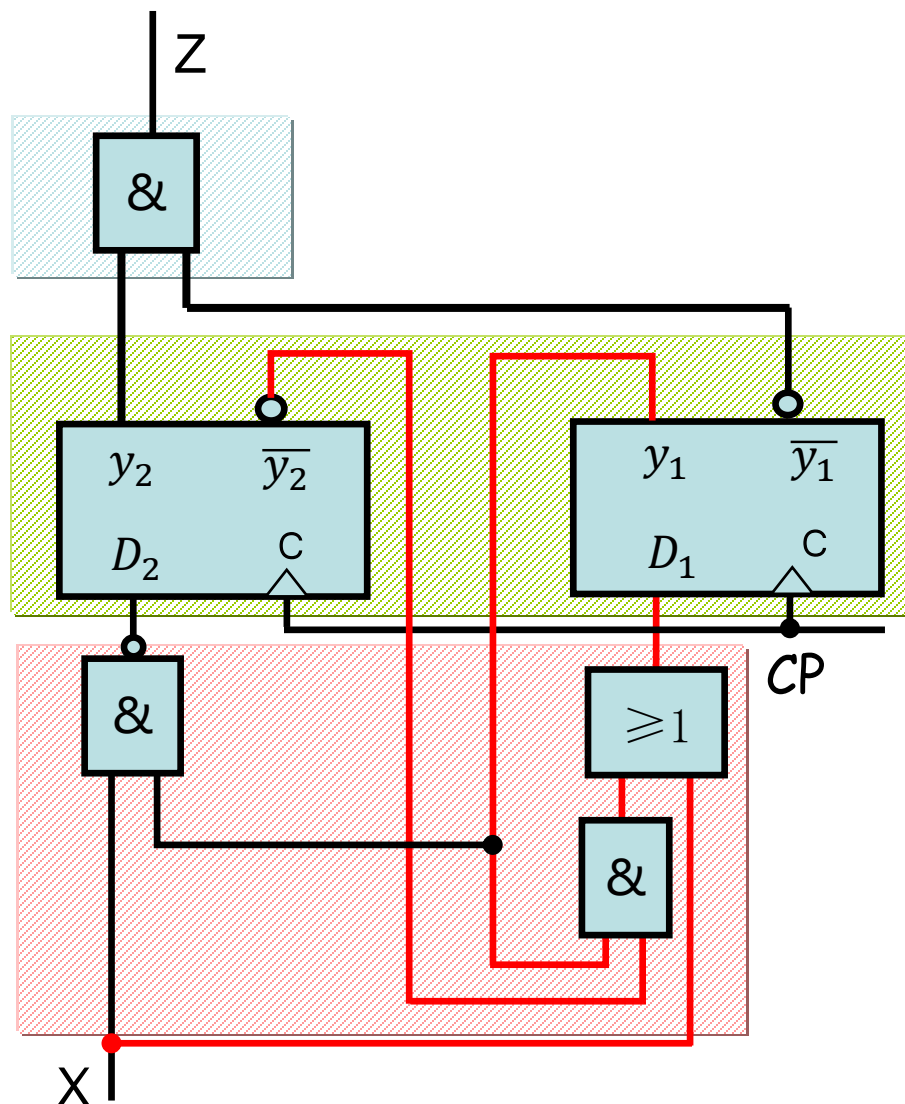
- $Z = y_2 \overline{y_1}$

– 激励函数表达式

- $D_2 = \overline{x} y_1$

- $D_1 = x + \overline{y_2} y_1$

– 次态函数表达式



函数表达式

□ Moore型电路

– 输出函数表达式

- $Z = y_2 \overline{y_1}$

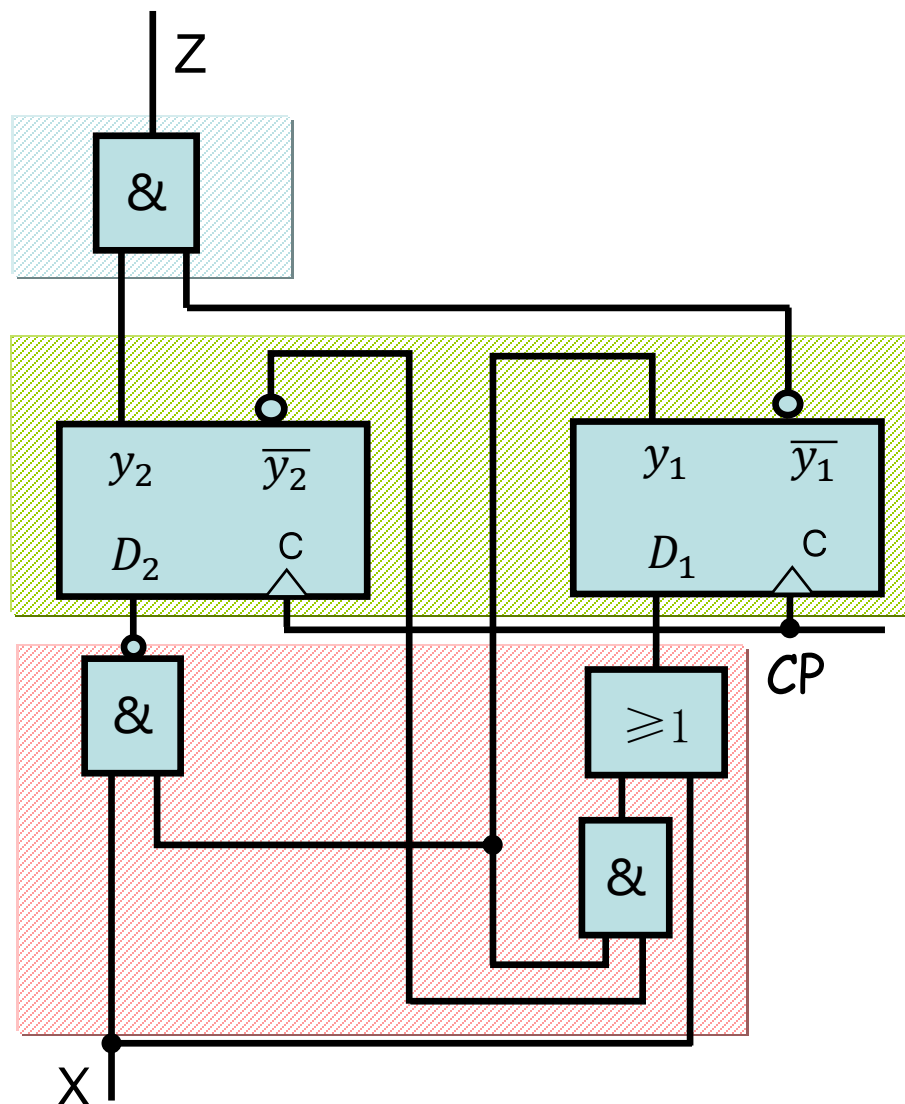
– 激励函数表达式

- $D_2 = \overline{x} y_1$

- $D_1 = x + \overline{y_2} y_1$

– 次态函数表达式

- $y^{n+1} = D$



函数表达式

□ Mealy型电路

– 输出函数表达式

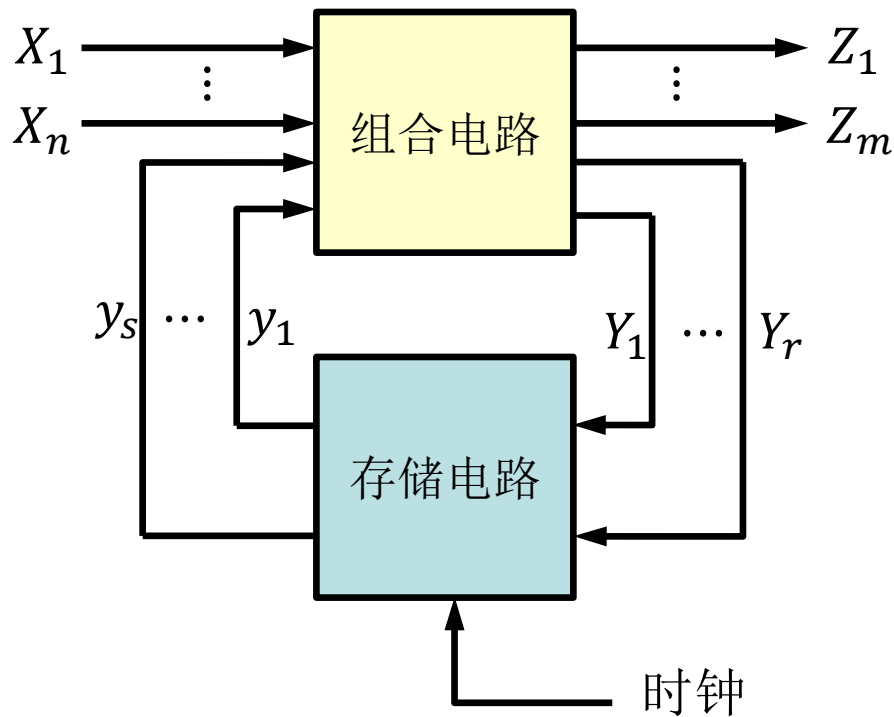
- $Z = f(x, y)$

– 激励函数表达式

- $Y = g(x, y)$

– 次态函数表达式

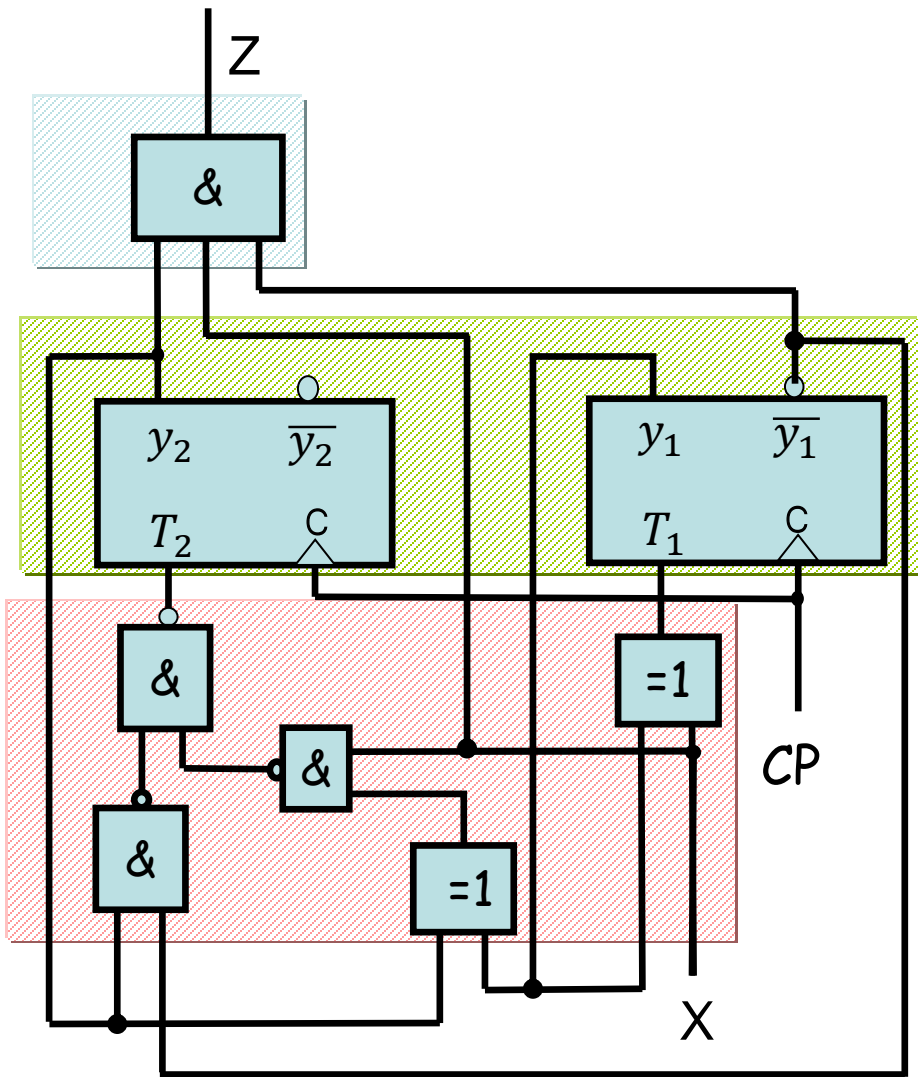
- $y^{n+1} = k(Y, y)$



函数表达式

□ Mealy型电路

– 输出函数表达式



函数表达式

□ Mealy型电路

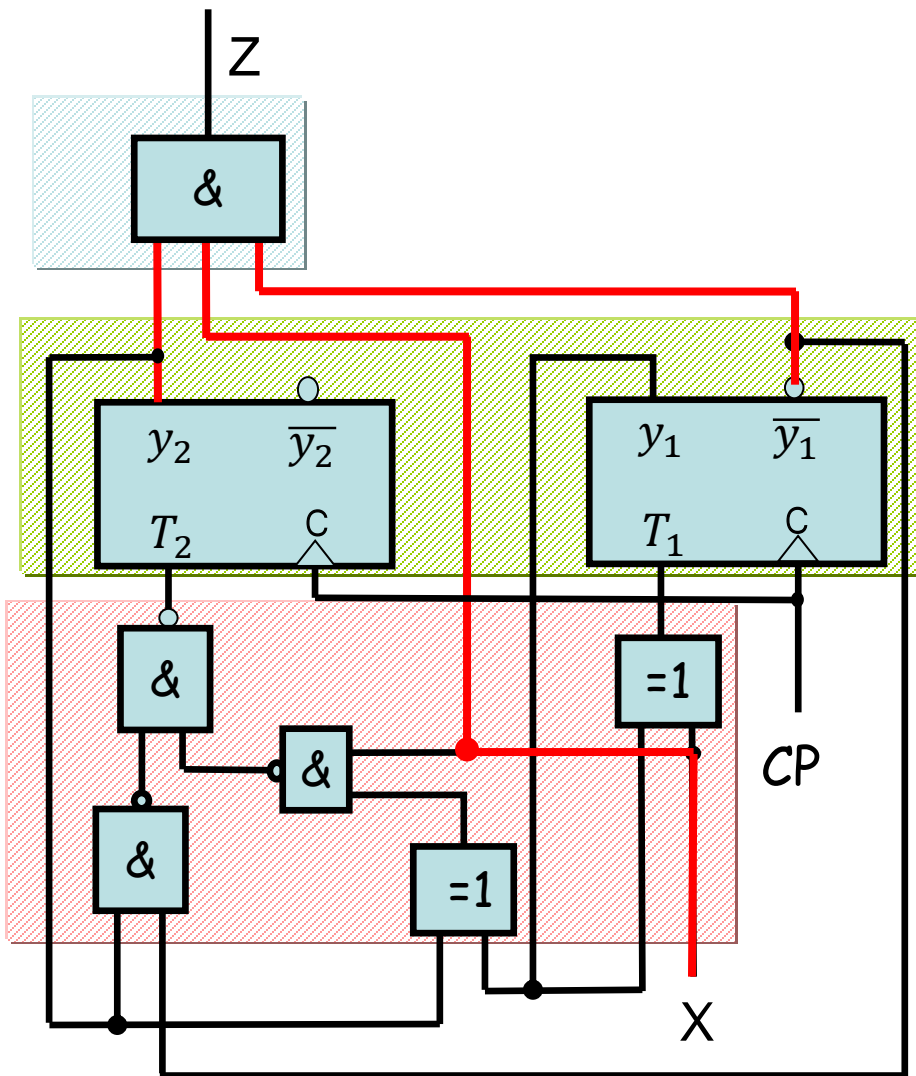
– 输出函数表达式

- $Z = xy_2\overline{y_1}$

– 次态函数表达式

- $y^{n+1} = y \oplus T$

– 激励函数表达式



函数表达式

□ Mealy型电路

– 输出函数表达式

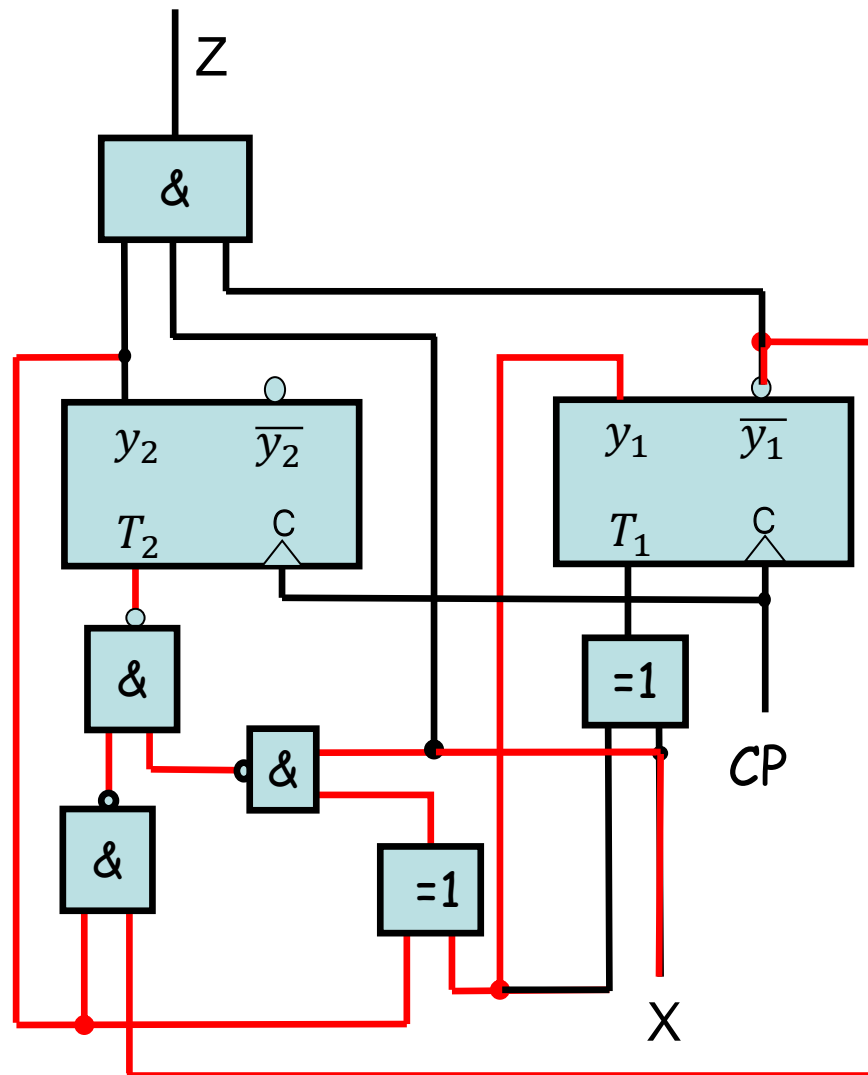
- $Z = xy_2\overline{y_1}$

– 次态函数表达式

- $y^{n+1} = y \oplus T$

– 激励函数表达式

- $$T_2 = \overline{\overline{y_2\overline{y_1}} \cdot \overline{(y_2 \oplus y_1)x}}$$
$$= y_2 \overline{y_1} + (y_2 \oplus y_1)x$$



函数表达式

Mealy型电路

– 输出函数表达式

- $Z = xy_2\bar{y}_1$

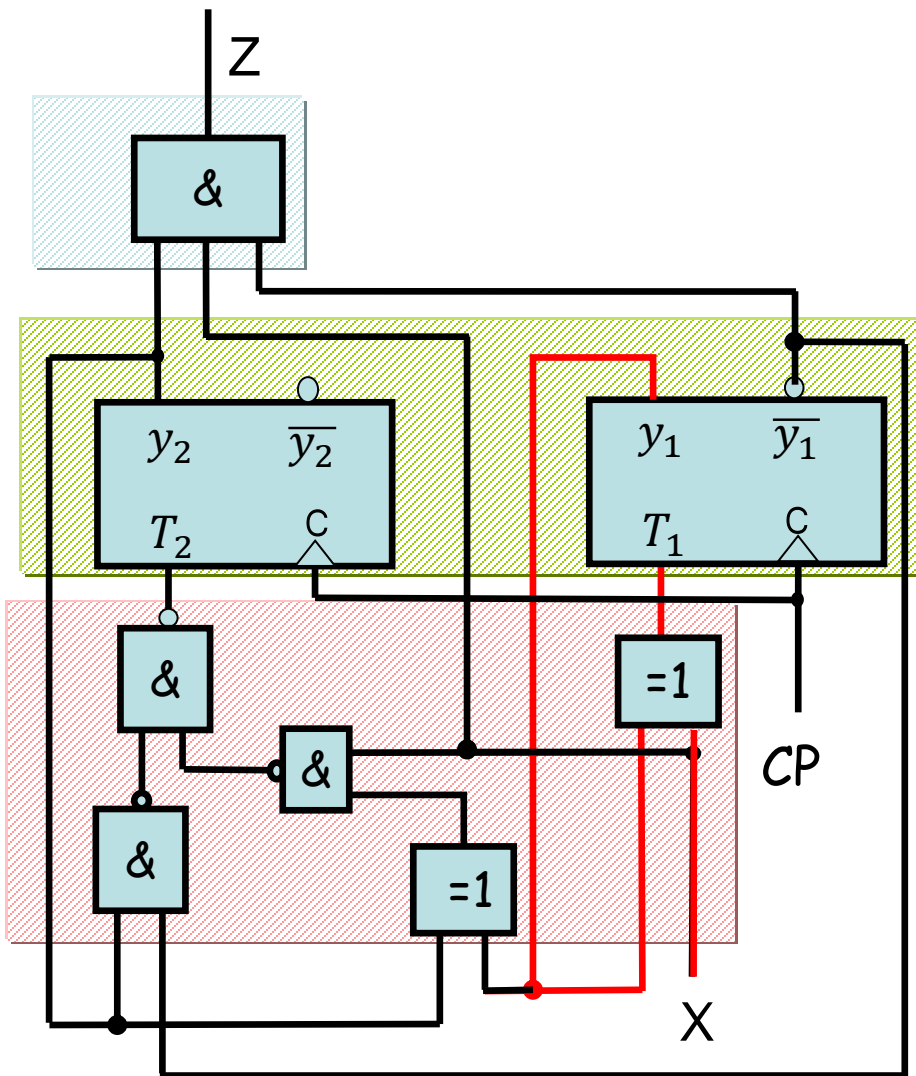
– 次态函数表达式

- $y^{n+1} = y \oplus T$

– 激励函数表达式

- $$T_2 = \overline{\overline{y_2\bar{y}_1} \cdot \overline{(y_2 \oplus y_1)x}}$$
$$= y_2 \bar{y}_1 + (y_2 \oplus y_1)x$$

- $T_1 = x \oplus y_1$



同步时序逻辑电路设计

□ 状态表

- 状态转移表
- 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
- 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

状态表

Mealy型

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

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

状态表

Moore型

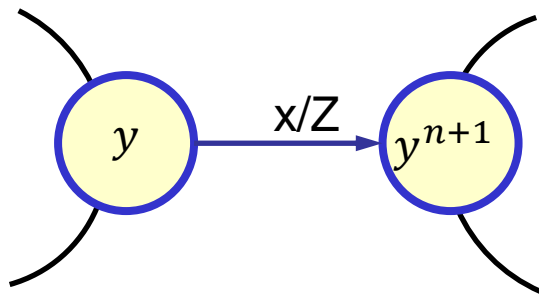
现态 y		次态 y^{n+1}		输出 Z
		输入 x		
y		y^{n+1}		Z

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

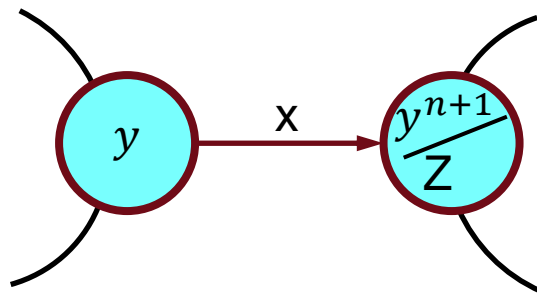
同步时序逻辑电路设计

□ 状态图

- 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



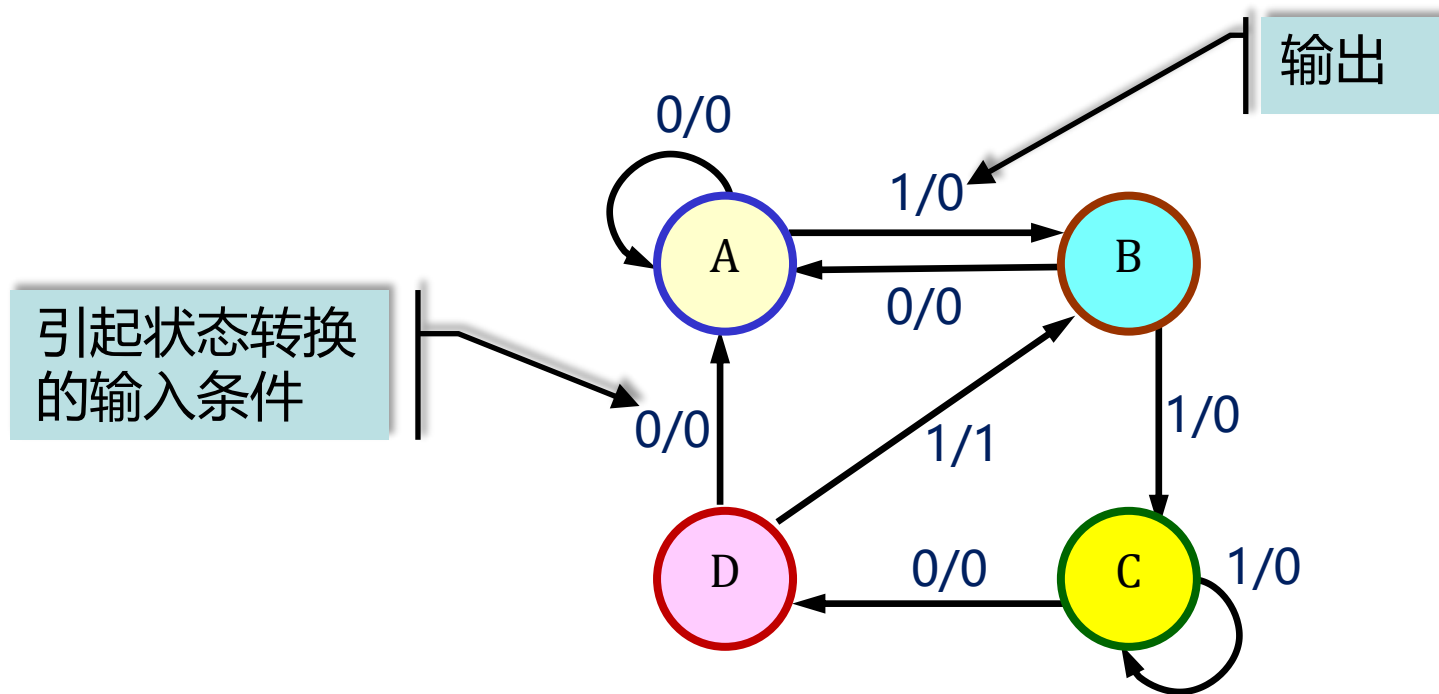
Mealy型电路状态图



Moore型电路状态图

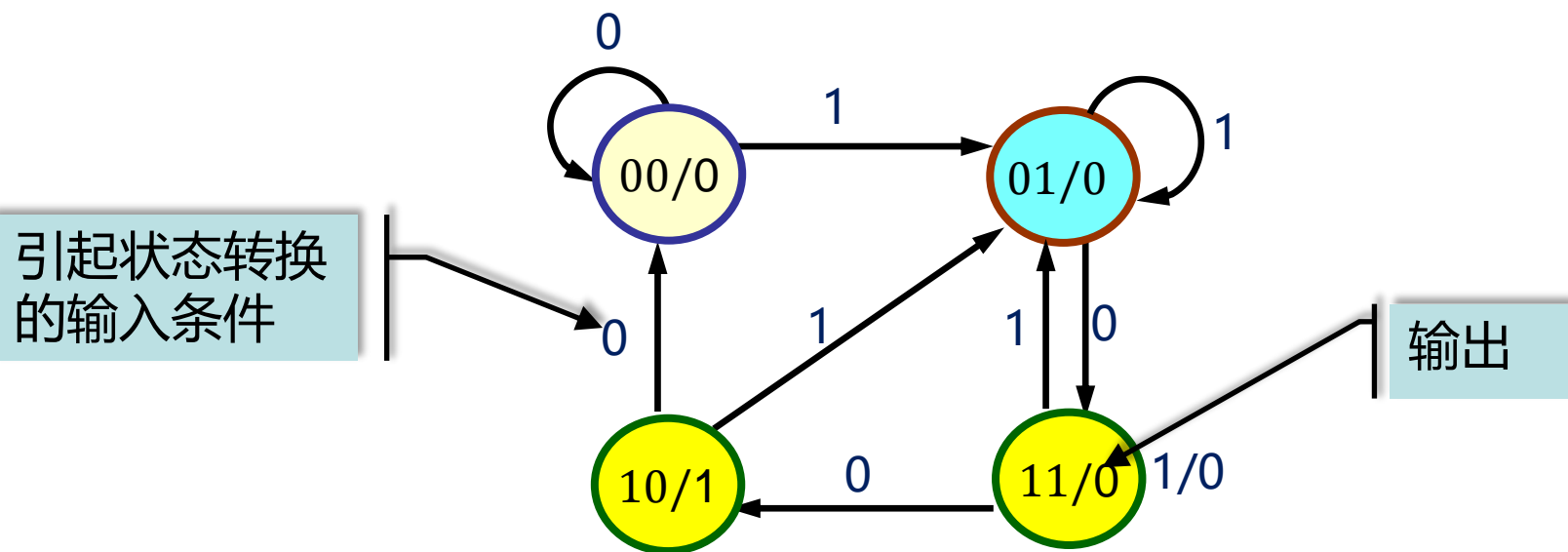
状态图

□ Mealy型电路状态图



状态图

□ Moore型电路状态图



状态图

□ 优点

- 直观、形象

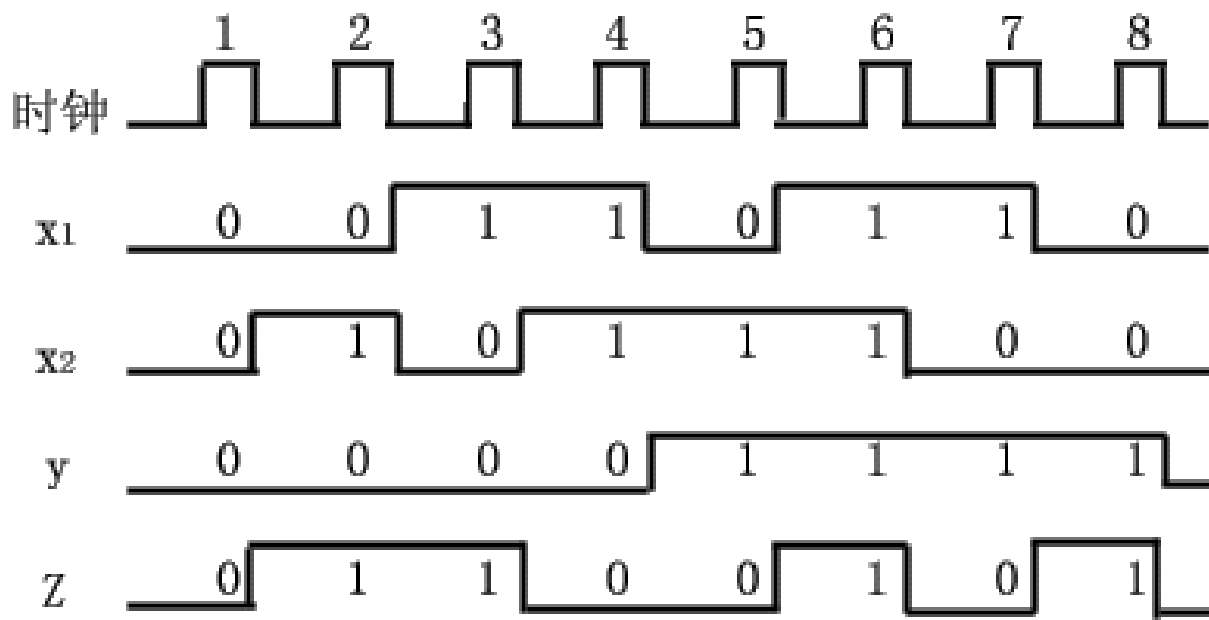
□ 状态图和状态表对比

- 状态表更规范
- 状态图更形象

同步时序逻辑电路设计

□时间图

- 表示输入信号、输出信号和电路状态的取值在各时刻的对应关系的波形图，通常又称为工作波形图



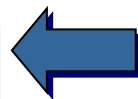
提纲

1

时序逻辑电路概述

2

同步时序逻辑电路分析



3

同步时序逻辑电路设计

同步时序逻辑电路分析

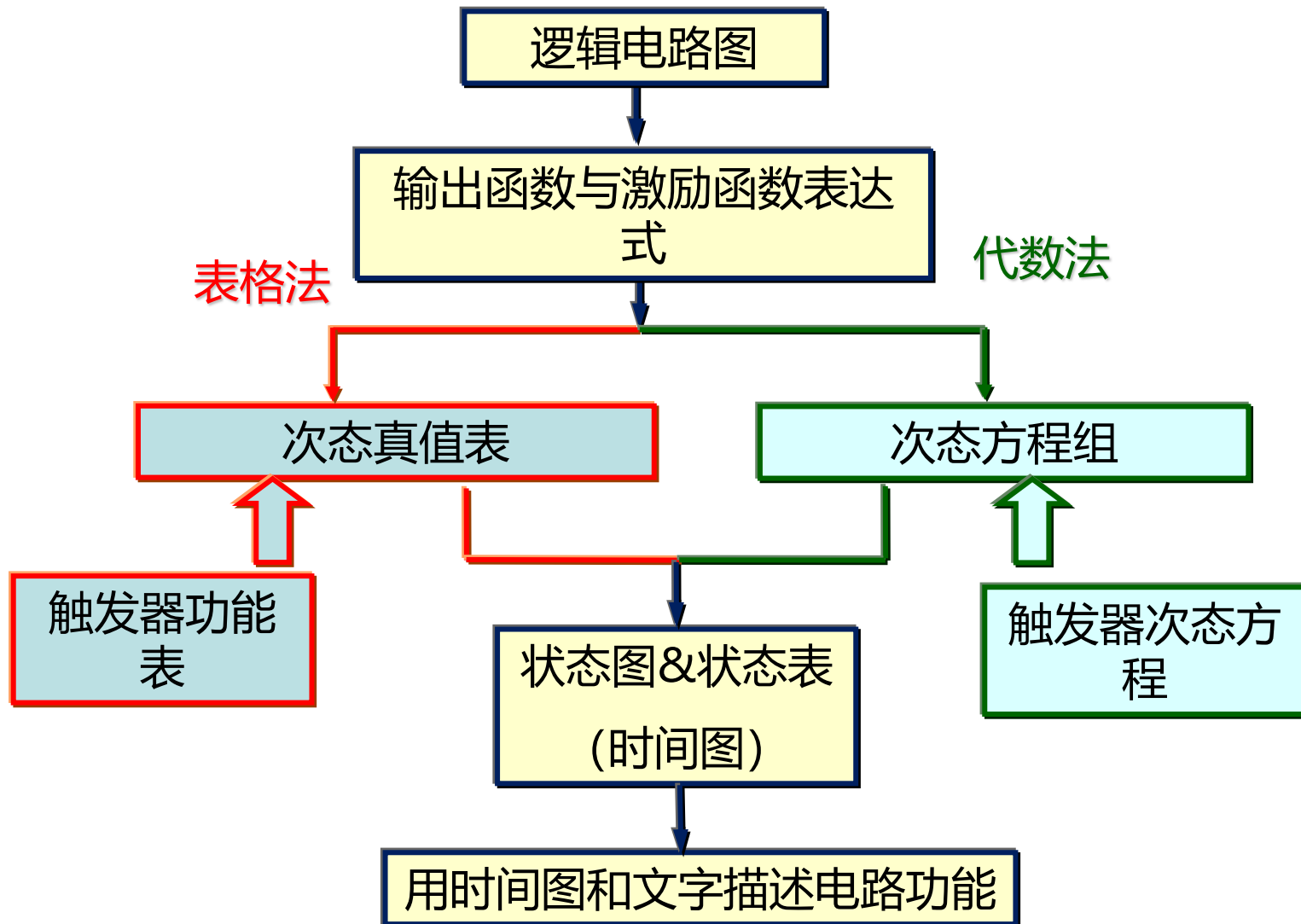
□ 分析的关键

- 找出电路状态和输出随输入变化而变化的规律，以便确定其逻辑功能

□ 分析的方法

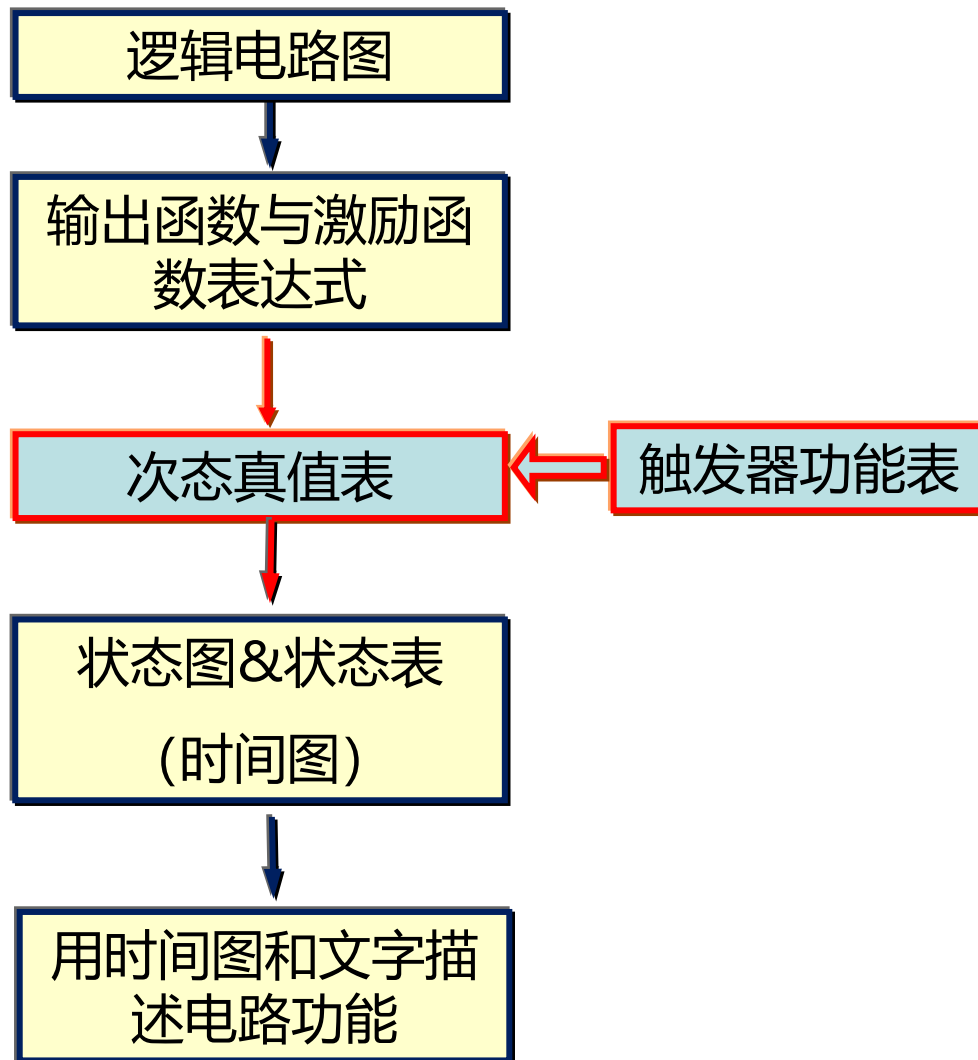
- 表格法
- 代数法

同步时序逻辑电路分析

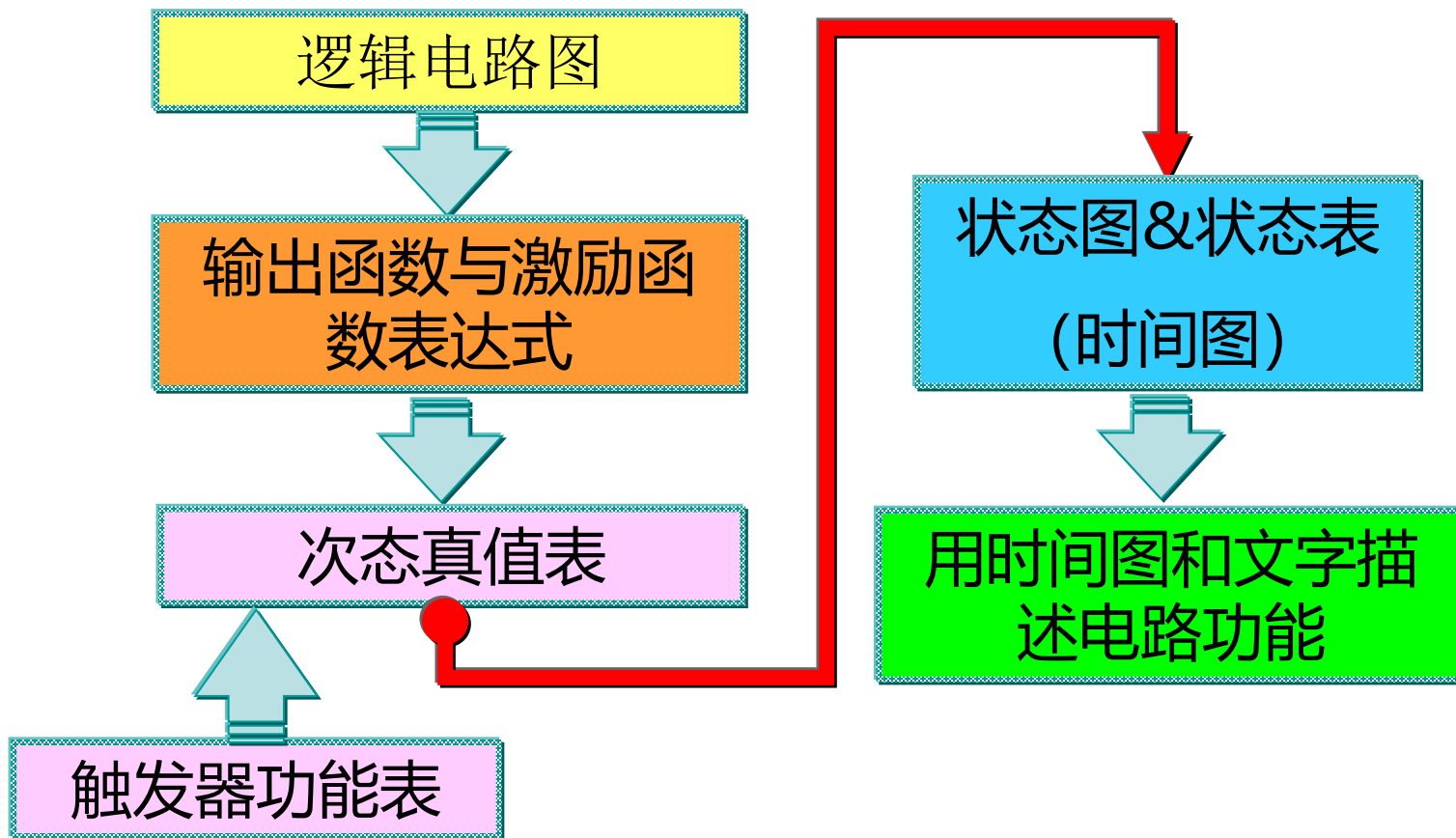


同步时序逻辑电路分析

□ 表格法

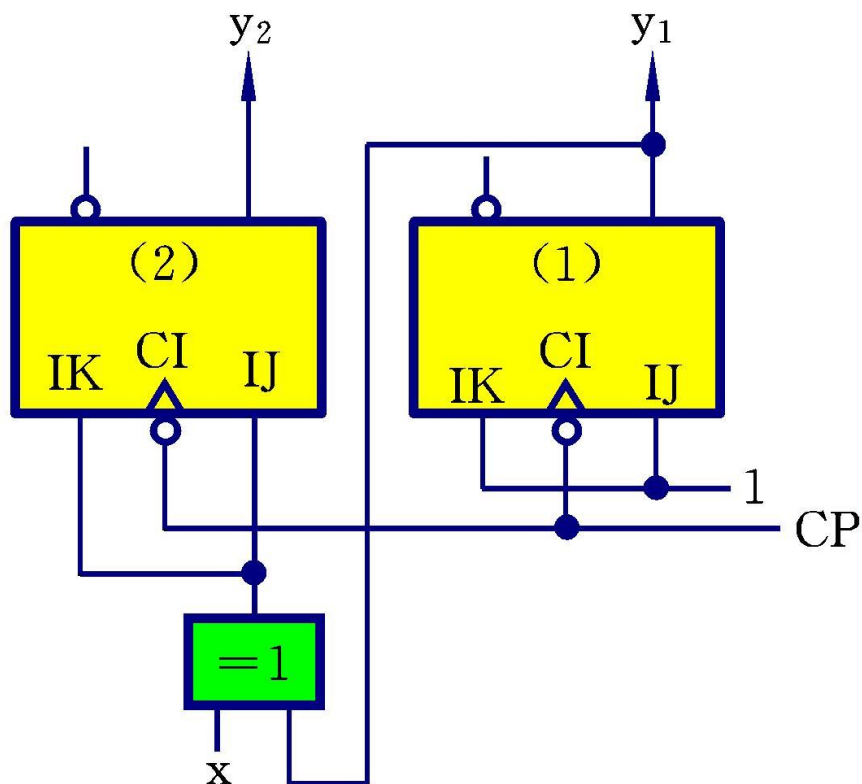


同步时序逻辑电路分析



表格分析法

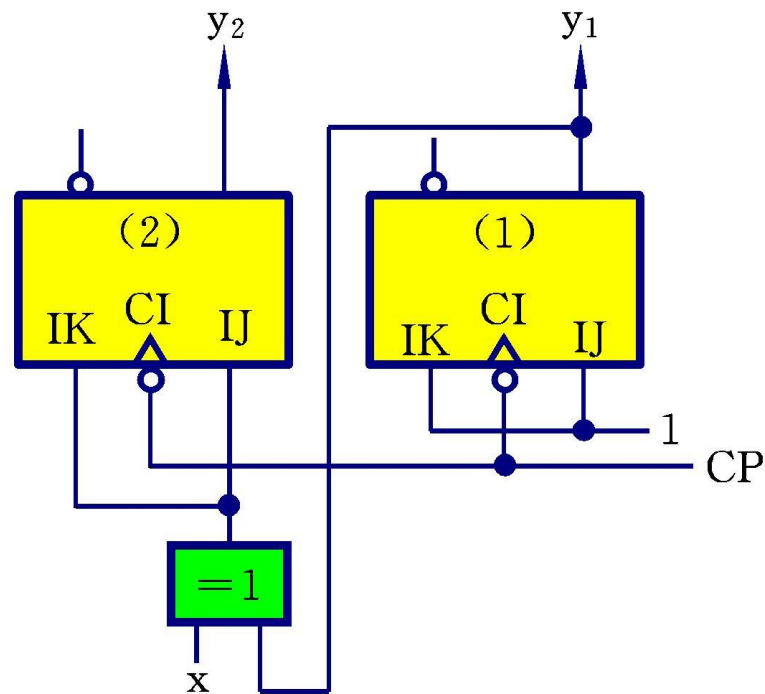
□ 例 分析下图所示同步时序逻辑电路



表格分析法

□分析：

- 两个J-K触发器
 - 相同的时钟端
 - 同步时序逻辑电路
- 一个异或门
- 输入：x
- 电路的状态： y_2 、 y_1
- 电路的输出： y_2 、 y_1
- Moore型电路

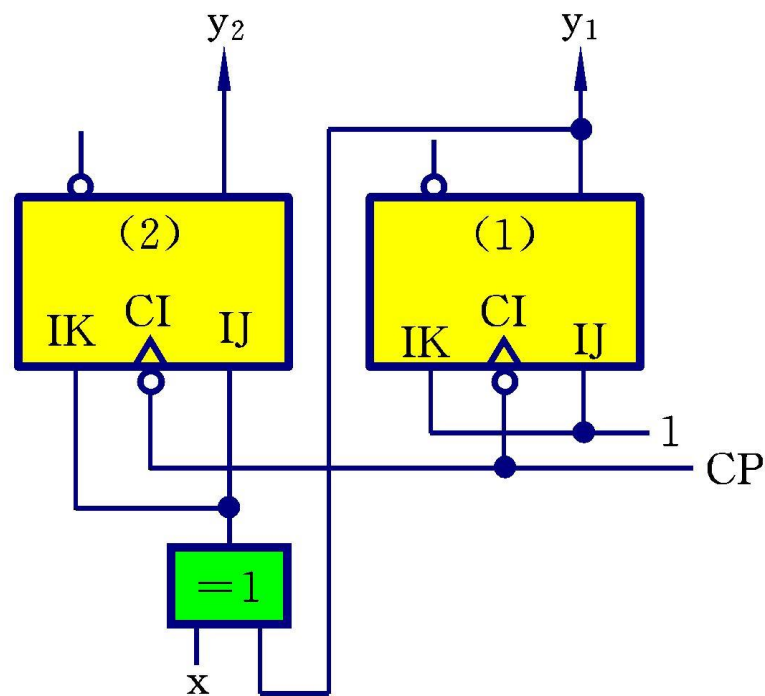


表格分析法

□ 分析

— 激励函数表达式

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



表格分析法

□分析：

— 激励函数表达式

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

— 次态真值表

$J \ K$	Q^{n+1}
0 0	Q
0 1	0
1 0	1
1 1	\bar{Q}

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	0 1	1 1	1 1	
0	1 0	0 0	1 1	
0	1 1	1 1	1 1	
1	0 0	1 1	1 1	
1	0 1	0 0	1 1	
1	1 0	1 1	1 1	
1	1 1	0 0	1 1	

表格分析法

□分析：

— 激励函数表达式

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

— 次态真值表

$J \ K$	Q^{n+1}
0 0	Q
0 1	0
1 0	1
1 1	\bar{Q}

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	0 1	1 1	1 1	
0	1 0	0 0	1 1	
0	1 1	1 1	1 1	
1	0 0	1 1	1 1	
1	0 1	0 0	1 1	
1	1 0	1 1	1 1	
1	1 1	0 0	1 1	

同步时序逻辑电路分析

□分析：

– 激励函数表达式

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

– 次态真值表

$J \ K$	Q^{n+1}
0 0	Q
0 1	0
1 0	1
1 1	\bar{Q}

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
0	0 1	1 1	1 1	
0	1 0	0 0	1 1	
0	1 1	1 1	1 1	
1	0 0	1 1	1 1	
1	0 1	0 0	1 1	
1	1 0	1 1	1 1	
1	1 1	0 0	1 1	

同步时序逻辑电路分析

□ 分析：

– 激励函数表达式

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

– 次态真值表

$J \ K$	Q^{n+1}
0 0	Q
0 1	0
1 0	1
1 1	\bar{Q}

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
0	0 1	1 1	1 1	
0	1 0	0 0	1 1	
0	1 1	1 1	1 1	
1	0 0	1 1	1 1	
1	0 1	0 0	1 1	
1	1 0	1 1	1 1	
1	1 1	0 0	1 1	

同步时序逻辑电路分析

□分析：

— 激励函数表达式

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

— 次态真值表

$J \ K$	Q^{n+1}
0 0	Q
0 1	0
1 0	1
1 1	\bar{Q}

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	
0	1 0	0 0	1 1	
0	1 1	1 1	1 1	
1	0 0	1 1	1 1	
1	0 1	0 0	1 1	
1	1 0	1 1	1 1	
1	1 1	0 0	1 1	

同步时序逻辑电路分析

□分析：

– 激励函数表达式

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

– 次态真值表

$J \ K$	Q^{n+1}
0 0	Q
0 1	0
1 0	1
1 1	\bar{Q}

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	
0	1 1	1 1	1 1	
1	0 0	1 1	1 1	
1	0 1	0 0	1 1	
1	1 0	1 1	1 1	
1	1 1	0 0	1 1	

同步时序逻辑电路分析

□ 分析：

– 激励函数表达式

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

– 次态真值表

$J \ K$	Q^{n+1}
0 0	Q
0 1	0
1 0	1
1 1	\bar{Q}

输入 x	现态 $y_2 \ y_1$
0	0 0
0	0 1
0	1 0
0	1 1
1	0 0
1	0 1
1	1 0
1	1 1

次态 $y_2^{n+1} y_1^{n+1}$
0 1
1 0
1 1
0 0
1 1
0 0
0 1
1 0

同步时序逻辑电路分析

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 0		
1 1		

输入 x	现态 $y_2 y_1$
0	0 0
0	0 1
0	1 0
0	1 1
1	0 0
1	0 1
1	1 0
1	1 1

次态 $y_2^{n+1} y_1^{n+1}$
0 1
1 0
1 1
0 0
1 1
0 0
0 1
1 0

同步时序逻辑电路分析

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

输入 x	现态 $y_2 y_1$
0	0 0
0	0 1
0	1 0
0	1 1
1	0 0
1	0 1
1	1 0
1	1 1

次态 $y_2^{n+1} y_1^{n+1}$
0 1
1 0
1 1
0 0
1 1
0 0
0 1
1 0

同步时序逻辑电路分析

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

输入 x	现态 $y_2 y_1$
0	0 0
0	0 1
0	1 0
0	1 1
1	0 0
1	0 1
1	1 0
1	1 1

次态 $y_2^{n+1} y_1^{n+1}$
0 1
1 0
1 1
0 0
1 1
0 0
0 1
1 0

同步时序逻辑电路分析

□分析：

— 激励函数表达式

- $J_2 = K_2 = x \oplus y_1$

- $J_1 = K_1 = 1$

— 次态真值表

— 状态表

— 状态图

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

同步时序逻辑电路分析

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

00

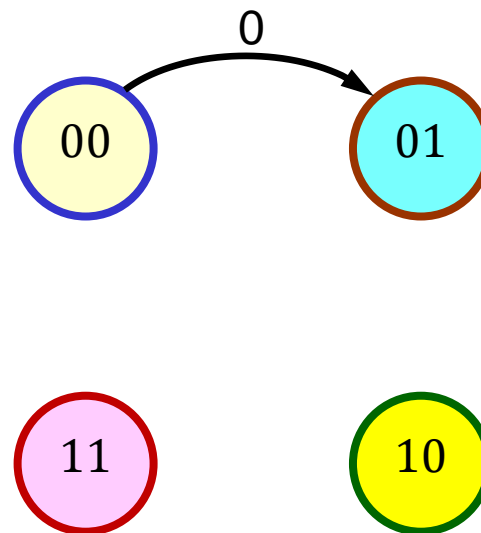
01

11

10

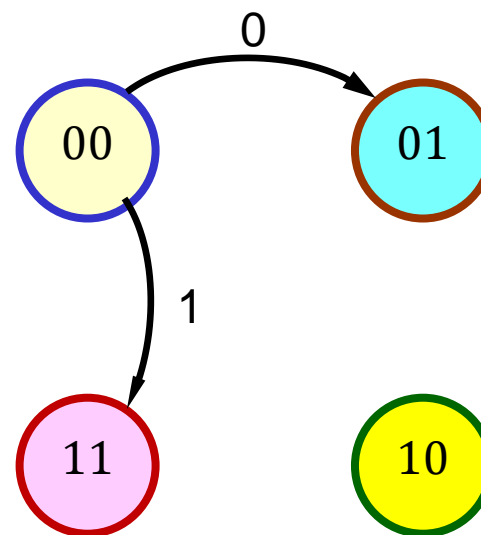
同步时序逻辑电路分析

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



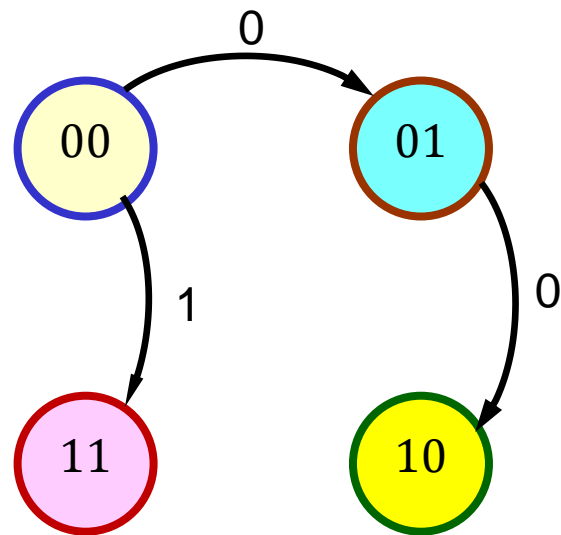
同步时序逻辑电路分析

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



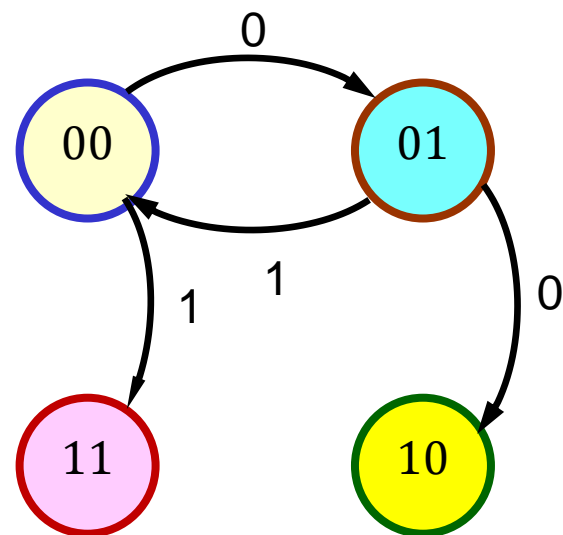
同步时序逻辑电路分析

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



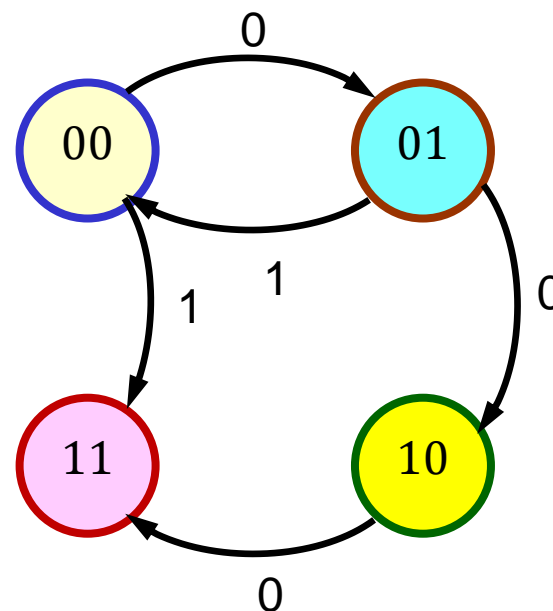
同步时序逻辑电路分析

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



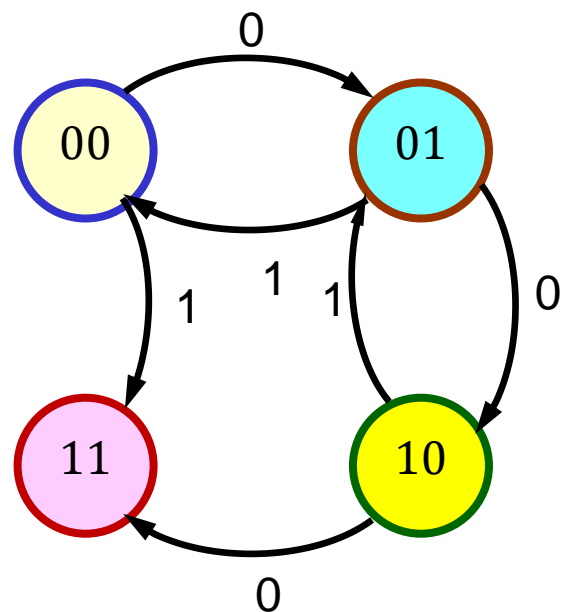
同步时序逻辑电路分析

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



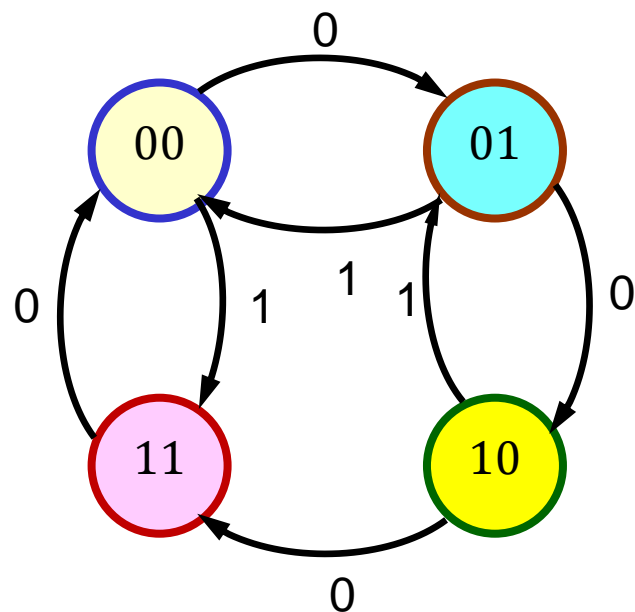
同步时序逻辑电路分析

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



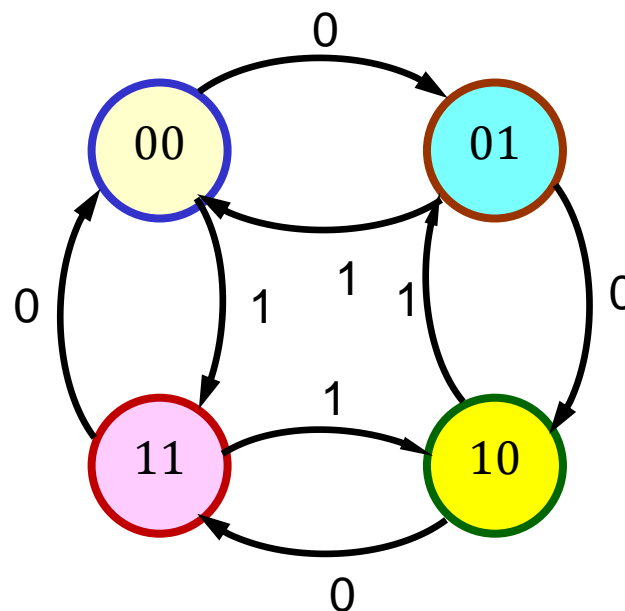
同步时序逻辑电路分析

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



同步时序逻辑电路分析

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



同步时序逻辑电路分析

□分析：

– 激励函数表达式

- $J_2 = K_2 = x \oplus y_1$

- $J_1 = K_1 = 1$

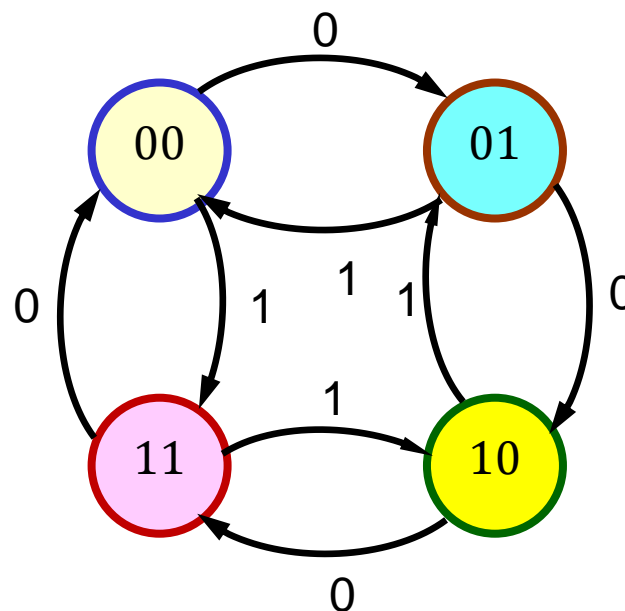
– 次态真值表

– 状态表

– 状态图

– 功能评述

- 2位二进制数可逆计数器



同步时序逻辑电路分析

□分析：

– 激励函数表达式

- $J_2 = K_2 = x \oplus y_1$

- $J_1 = K_1 = 1$

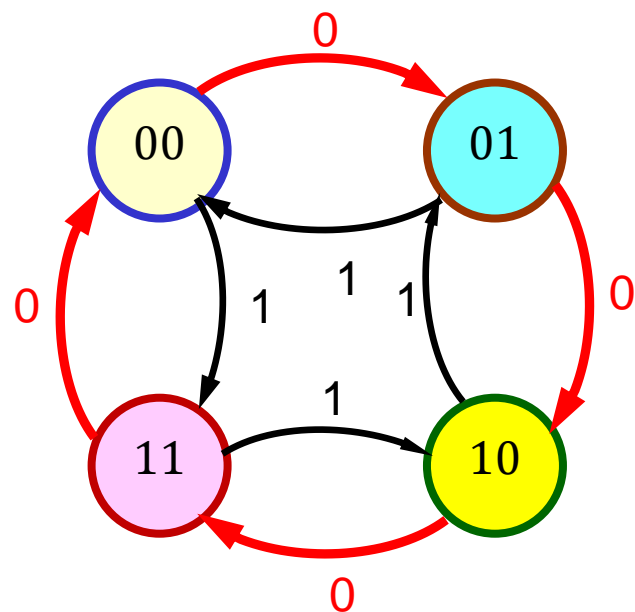
– 次态真值表

– 状态表

– 状态图

– 功能评述

- 2位二进制数可逆计数器



加1计数

同步时序逻辑电路分析

□分析：

– 激励函数表达式

- $J_2 = K_2 = x \oplus y_1$

- $J_1 = K_1 = 1$

– 次态真值表

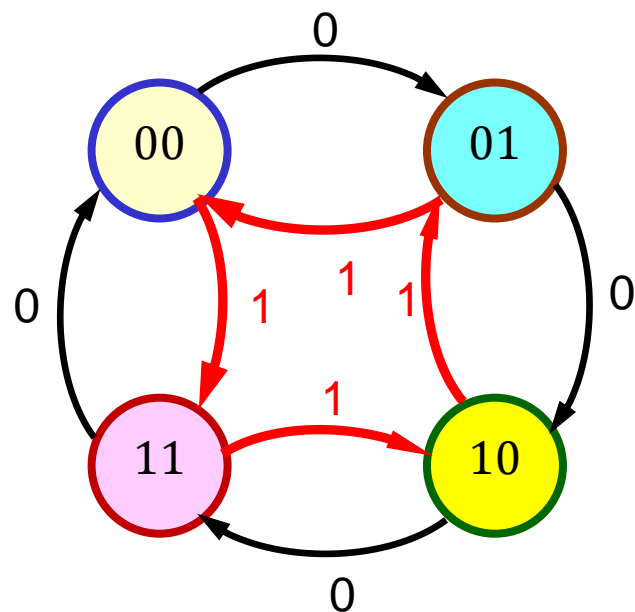
– 状态表

– 状态图

– 功能评述

- 2位二进制数可逆计数器

– 时间图



减1计数

同步时序逻辑电路设计

□时间图

- 在时序逻辑电路分析中，除了状态图和状态表之外，通常还用到时间图
- 时间图能较形象、生动地体现时序电路的工作过程，并可和实验观察的波形相比较，是描述时序电路工作特性的一种常用方式

同步时序逻辑电路分析

□ 作时间图的一般步骤

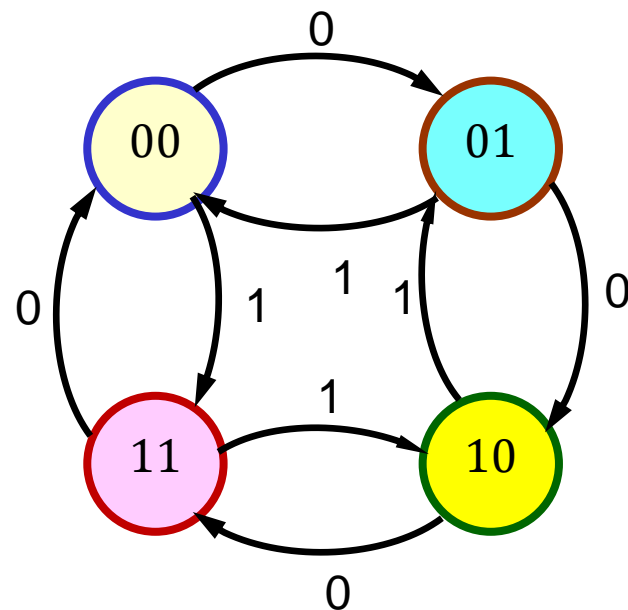
- 假设电路初始状态，并拟定一典型输入序列
- 作出状态和输出响应序列
- 根据响应序列画出波形图

同步时序逻辑电路分析

□ 做时间图

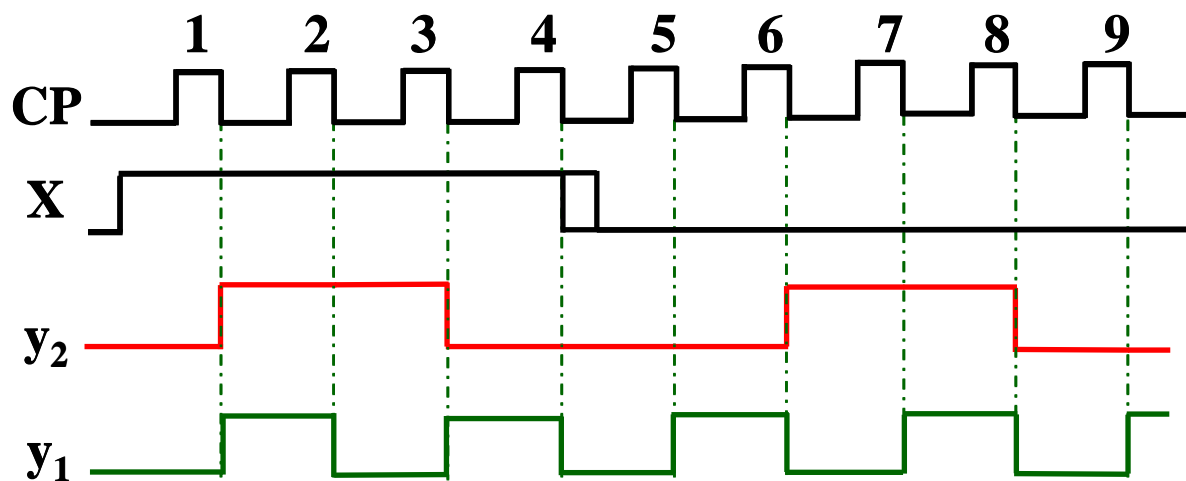
- 初始状态: $y_2 y_1 = 00$
- 输入 x 为电平信号, 典型输入序列为111100000
- 做电路的状态响应序列:

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

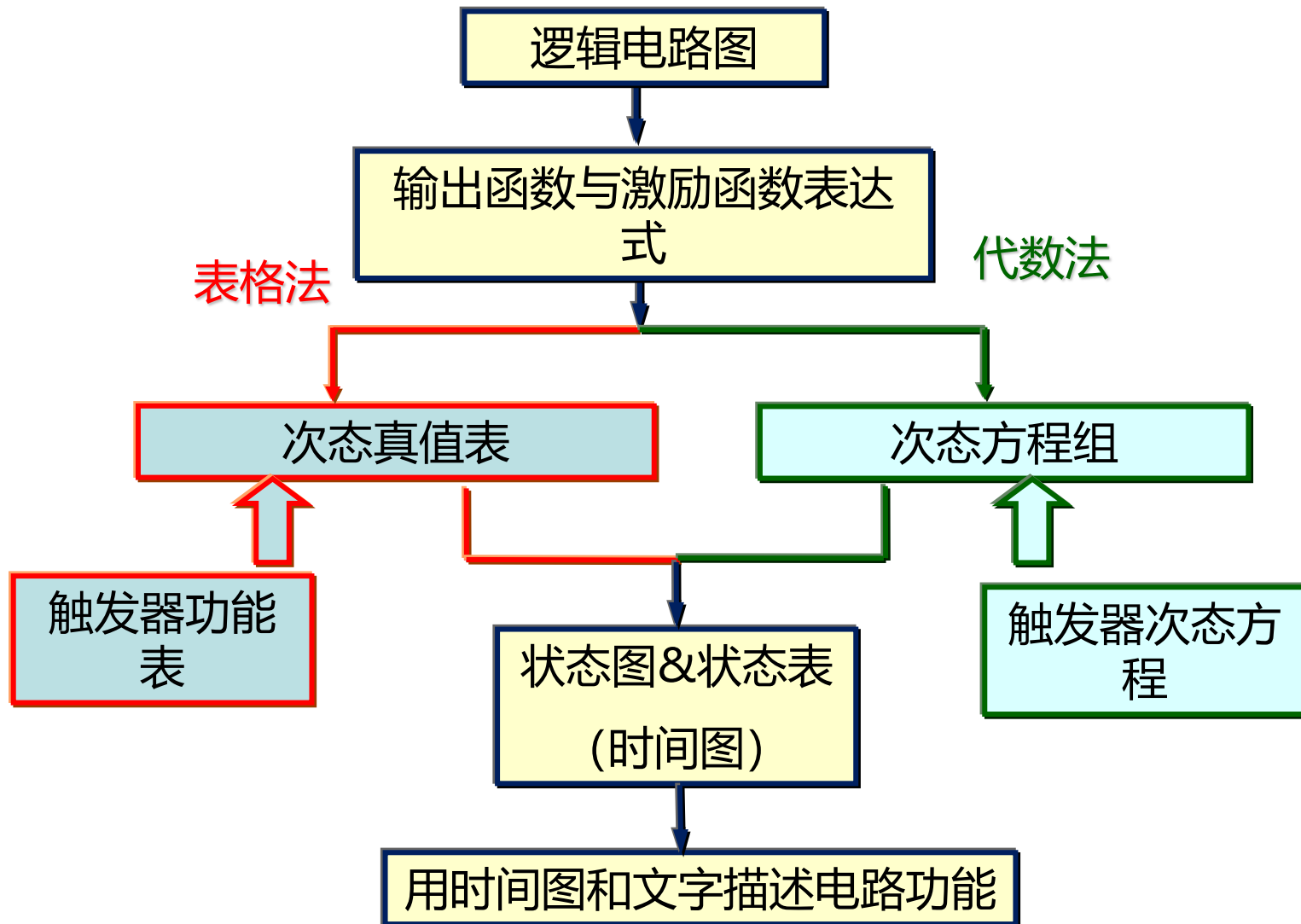


同步时序逻辑电路分析

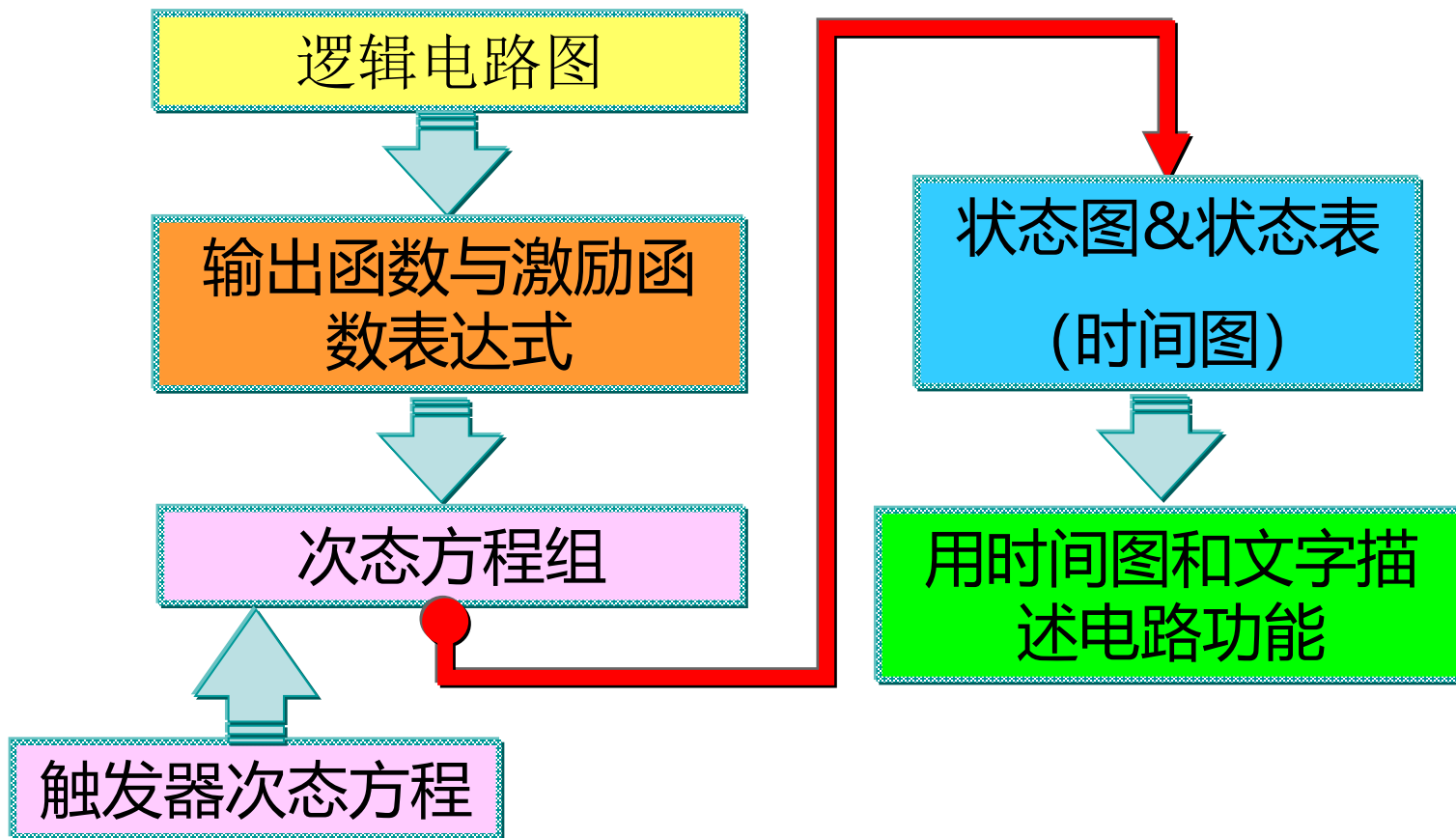
CP	1	2	3	4	5	6	7	8	9
x	1	1	1	1	0	0	0	0	0
y_2	0	1	1	0	0	0	1	1	0
y_1	0	1	0	1	0	1	0	1	0
y_2^{n+1}	1	1	0	0	0	1	1	0	0
y_1^{n+1}	1	0	1	0	1	0	1	0	1



同步时序逻辑电路分析

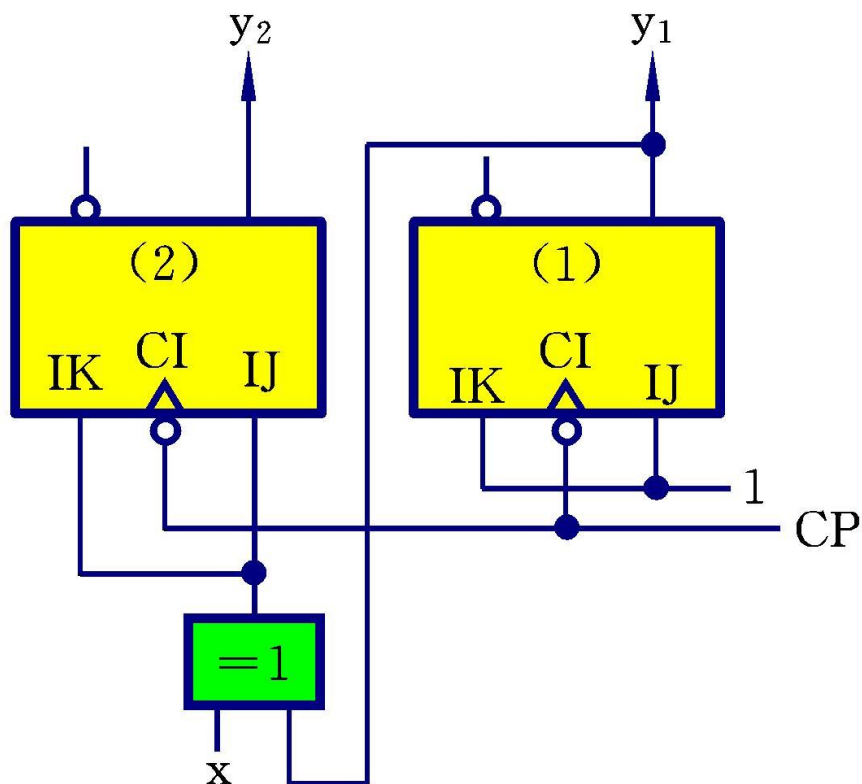


同步时序逻辑电路分析



代数分析法

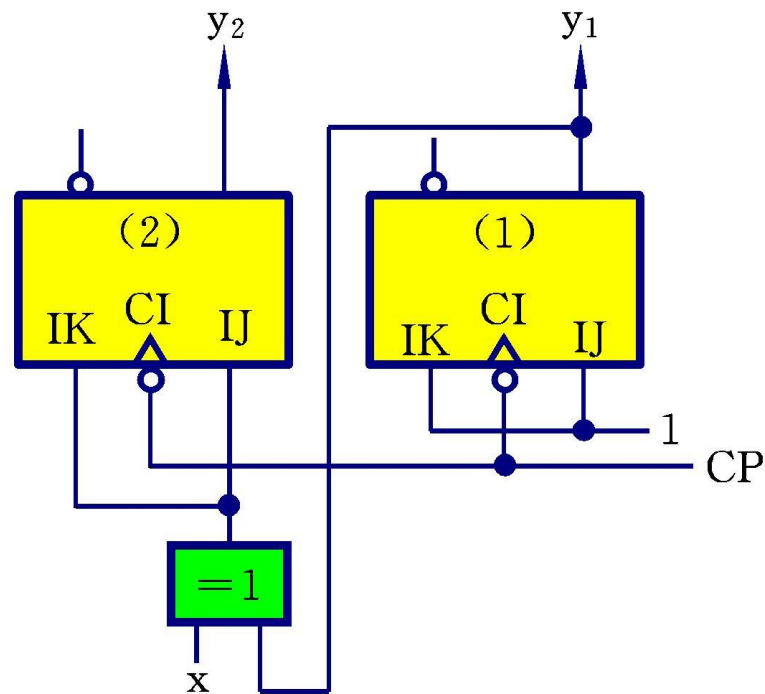
□ 例 分析下图所示同步时序逻辑电路



代数分析法

□分析：

- 两个J-K触发器
 - 相同的时钟端
 - 同步时序逻辑电路
- 一个异或门
- 输入：x
- 电路的状态： y_2 、 y_1
- 电路的输出： y_2 、 y_1
- Moore型电路



代数分析法

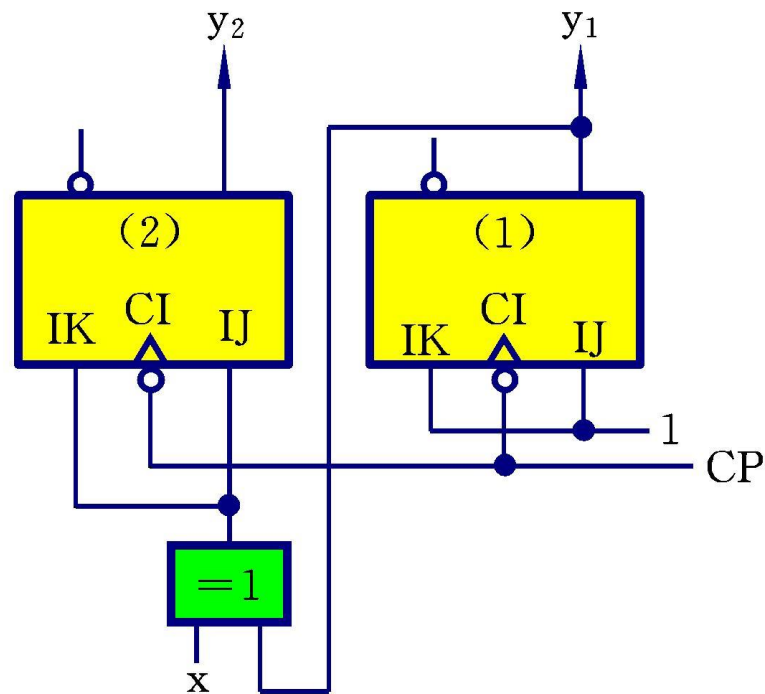
□ 函数表达式

— 激励函数表达式

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

— 次态方程组

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



代数分析法

□ 函数表达式

– 激励函数表达式

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

– 次态方程组

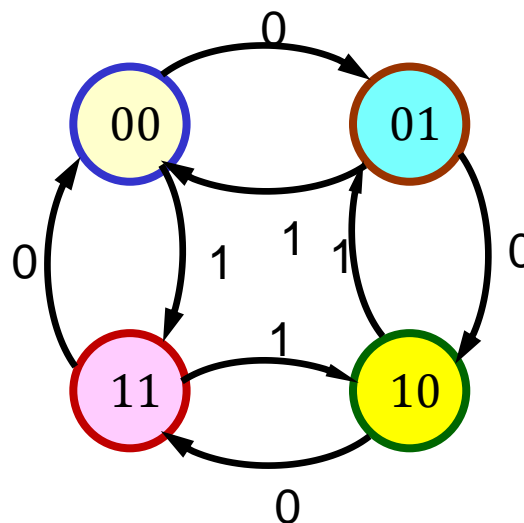
- $y_2^{n+1} = x \oplus y_1 \oplus y_2$
- $y_1^{n+1} = \overline{y_1}$

□ 状态表和状态图

□ 功能评述

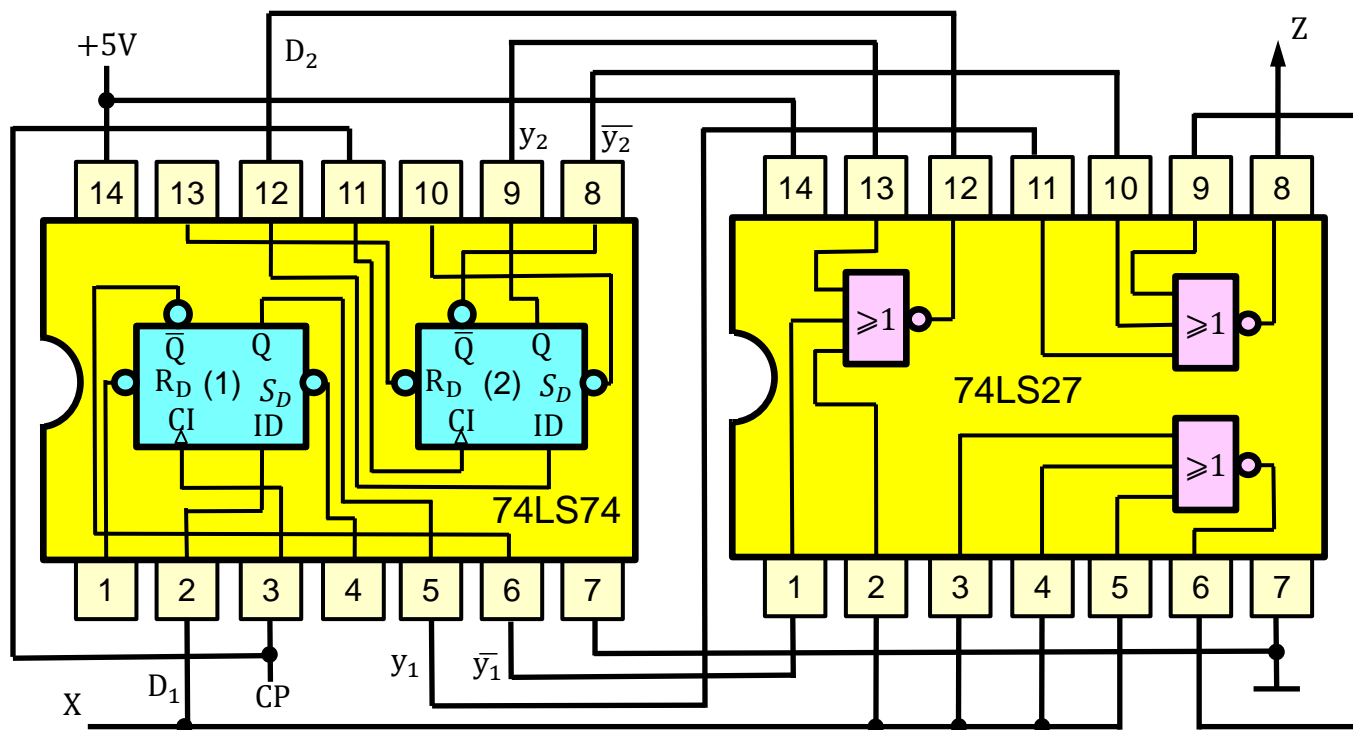
– 2位二进制数可逆计数器

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



同步时序逻辑电路分析

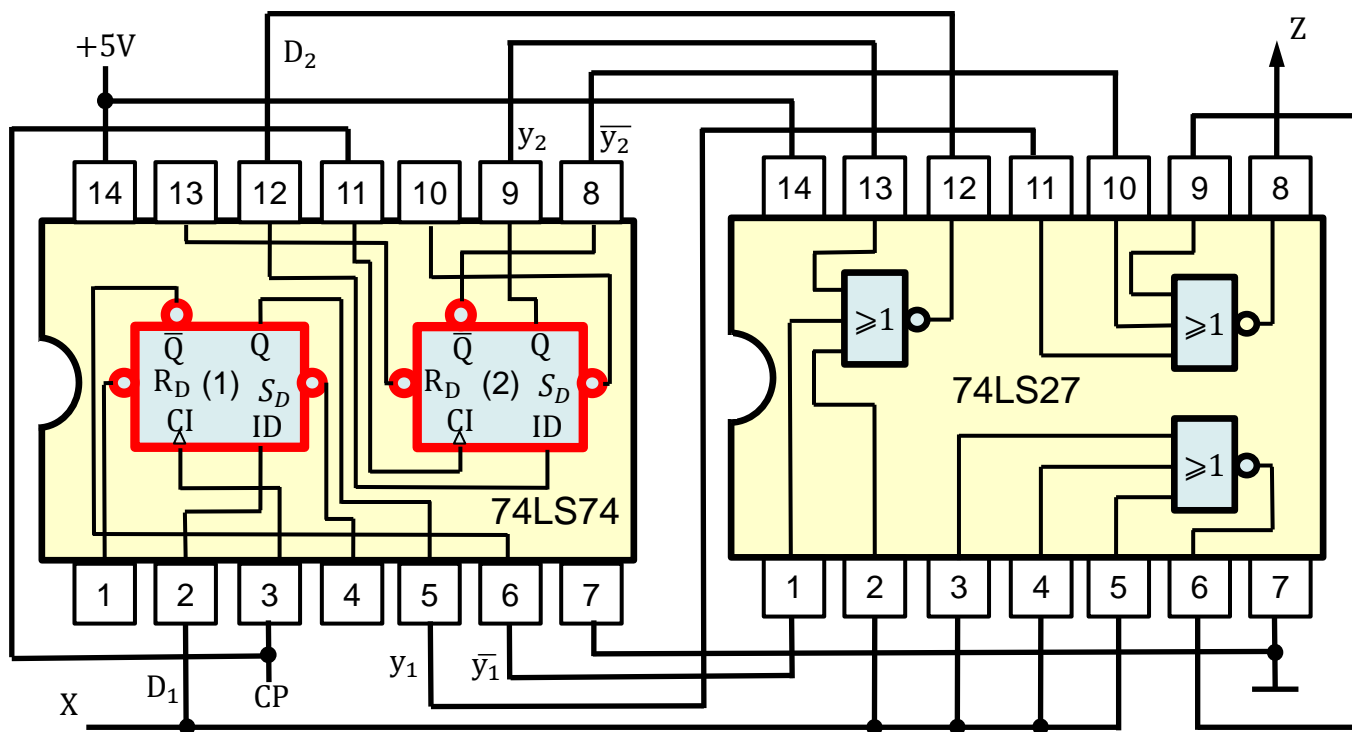
□ 例 分析下图所示同步时序逻辑电路



同步时序逻辑电路分析

□分析：

– 两个D触发器

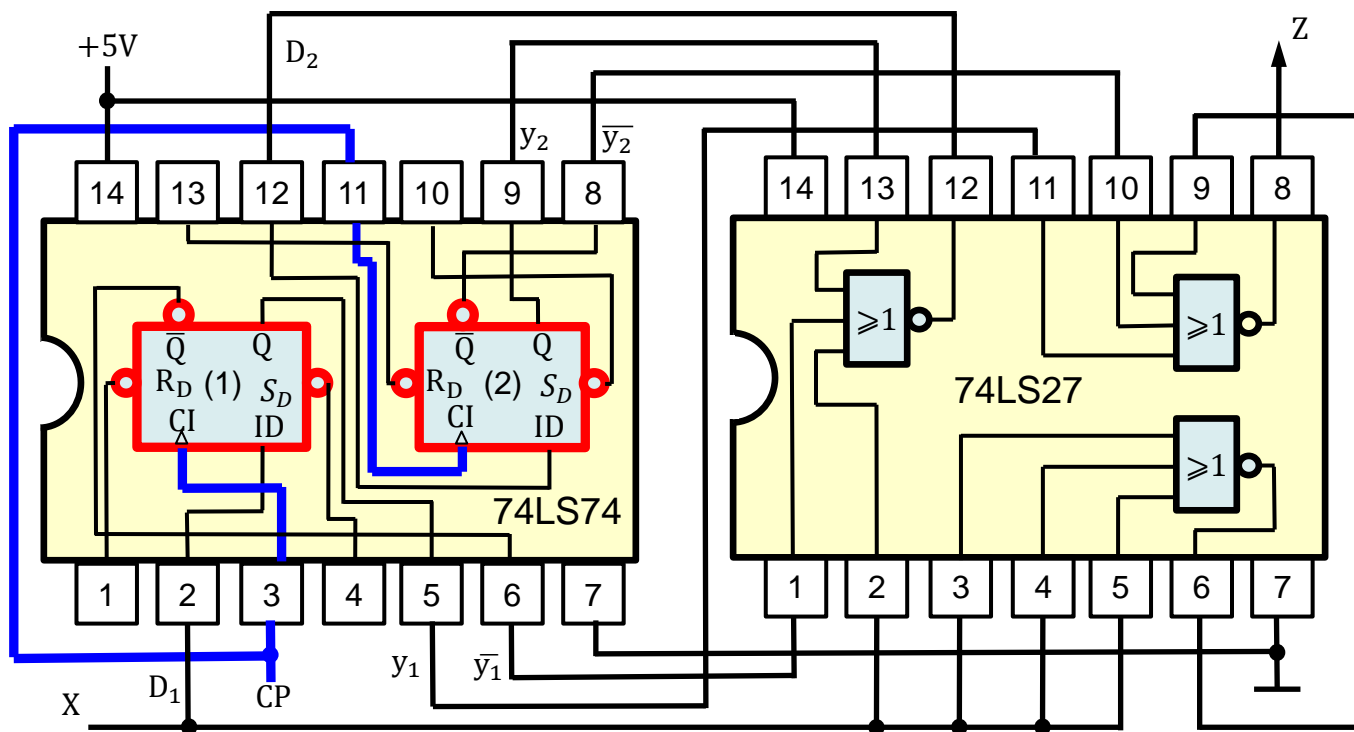


同步时序逻辑电路分析

□分析：

– 两个D触发器

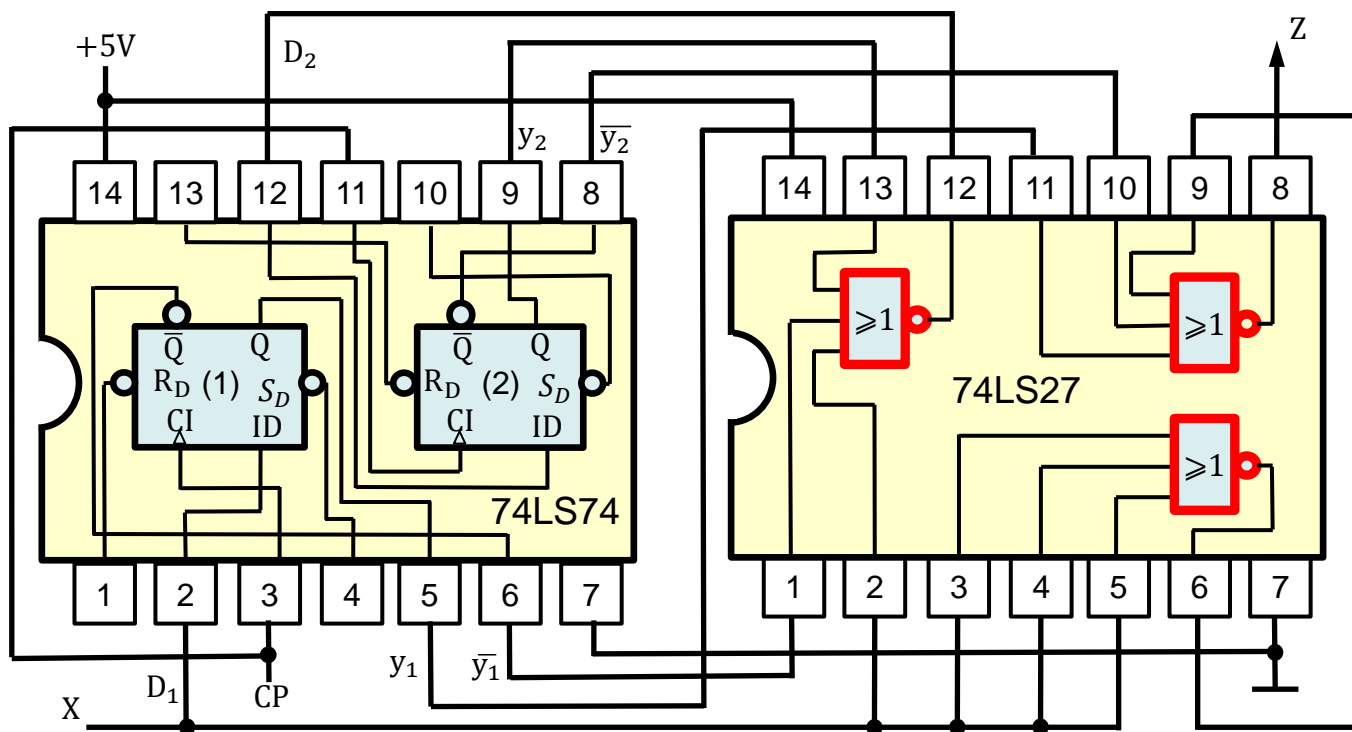
- 相同时钟端，同步时序逻辑电路



同步时序逻辑电路分析

□分析：

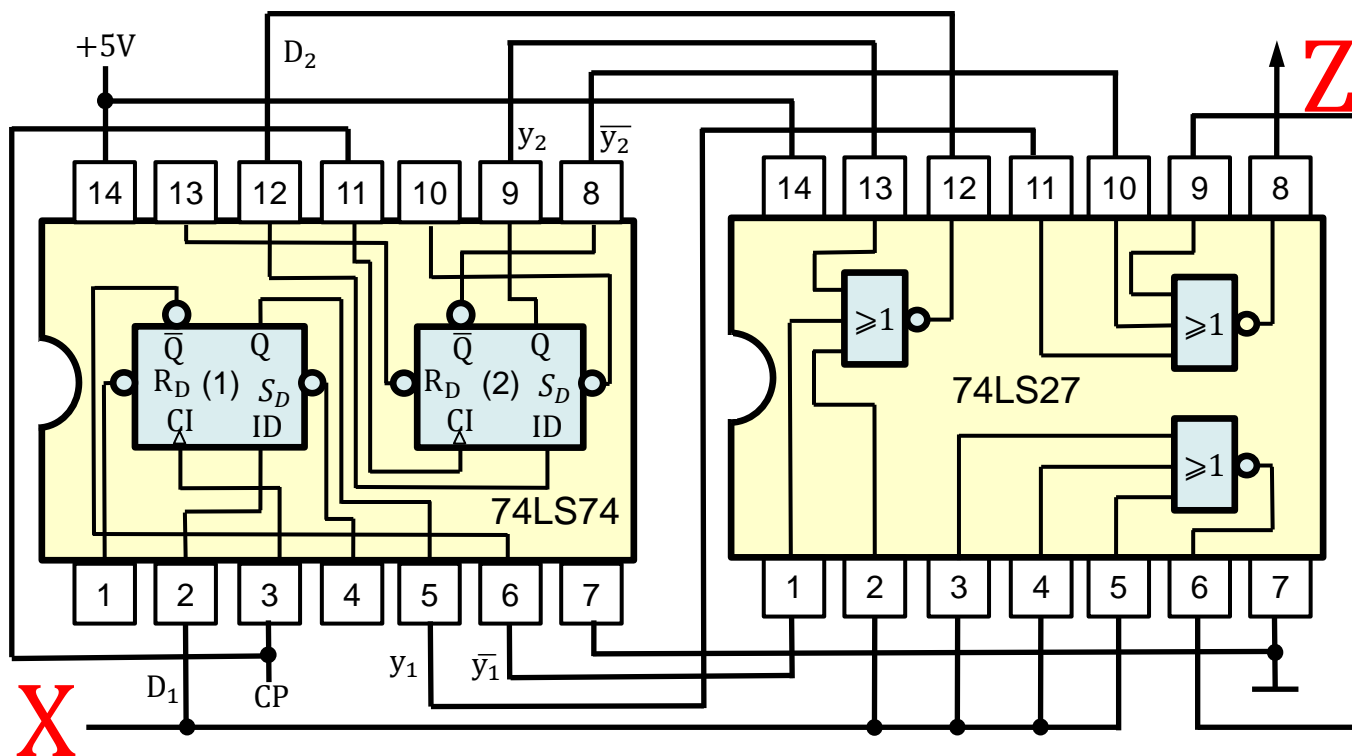
– 三个或非门



同步时序逻辑电路分析

□ 分析：

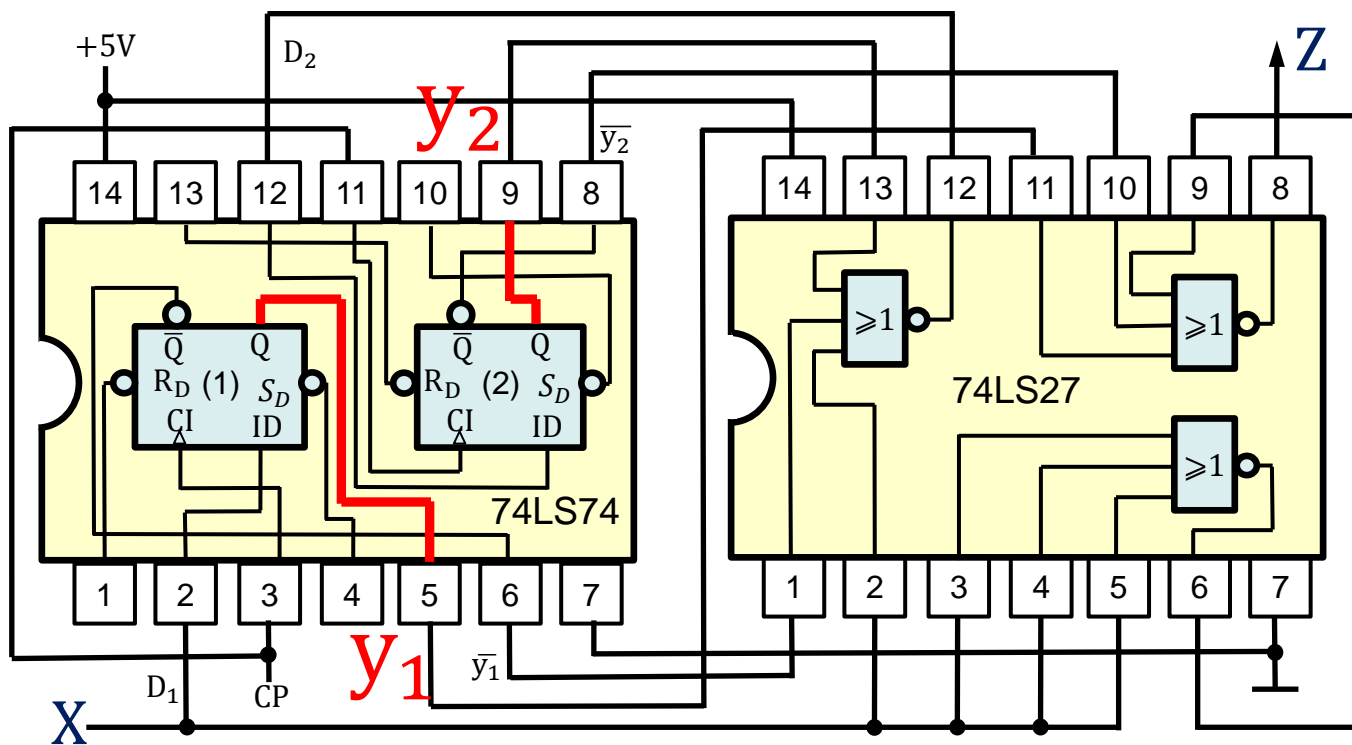
— 输入: x , 输出: z



同步时序逻辑电路分析

□分析：

– 电路的状态： y_2 、 y_1



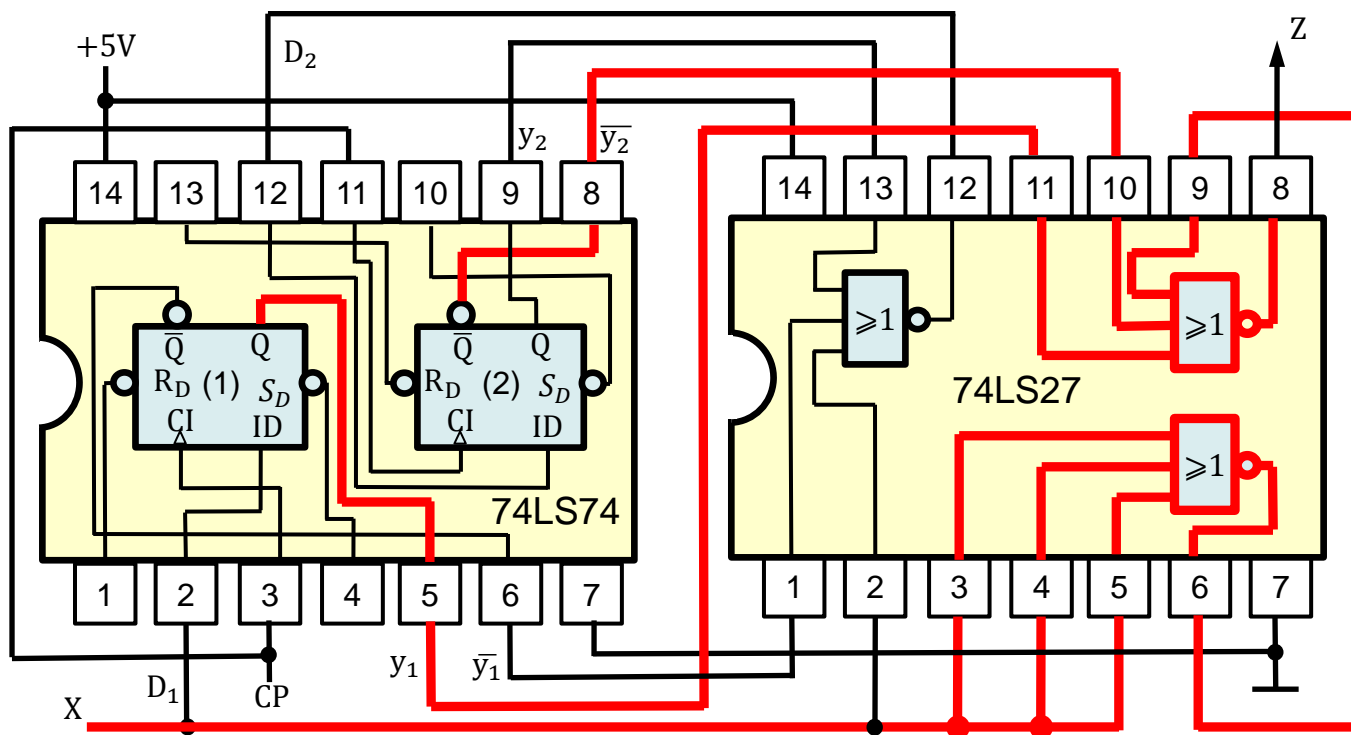
同步时序逻辑电路分析

□ 函数表达式

– 输出函数表达式

$$\bullet Z = \overline{x + x + x + y_2 + y_1} = \overline{\bar{x} + \overline{y_2} + y_1} = xy_2\overline{y_1}$$

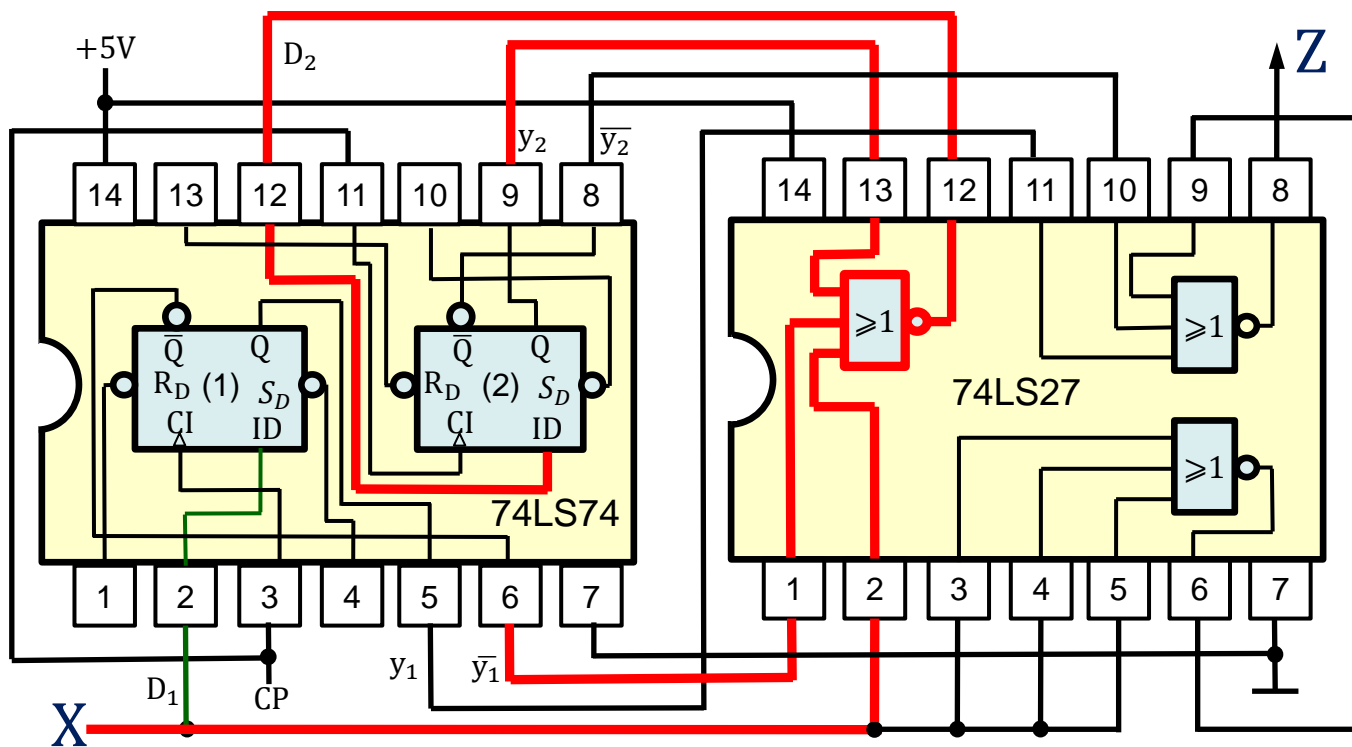
Mealy型



同步时序逻辑电路分析

激励函数表达式

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



同步时序逻辑电路分析

□ 函数表达式

$$- D_2 = \bar{x} \bar{y}_2 y_1$$

$$- D_1 = x$$

$$- Z = x y_2 \bar{y}_1$$

□ 次态真值表

□ 状态表

输入 x	现态 $y_2 y_1$	激励函数 $D_2 \quad D_1$		次态 $y_2^{n+1} y_1^{n+1}$	输出 Z
0	0 0	0	0	0 0	0
0	0 1	1	0	1 0	0
0	1 0	0	0	0 0	0
0	1 1	0	0	0 0	0
1	0 0	0	1	0 1	0
1	0 1	0	1	0 1	0
1	1 0	0	1	0 1	1
1	1 1	0	1	0 1	0

同步时序逻辑电路分析

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

输入 x	现态 $y_2 y_1$	
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

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

同步时序逻辑电路分析

□ 函数表达式

- $D_2 = \bar{x} \bar{y}_2 y_1$
- $D_1 = x$
- $Z = x y_2 \bar{y}_1$

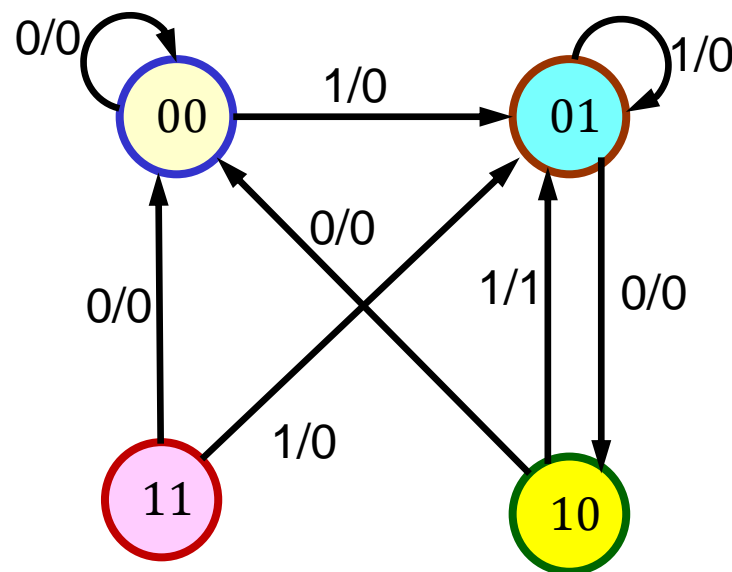
□ 次态真值表

□ 状态表

□ 状态图

□ 时间图

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

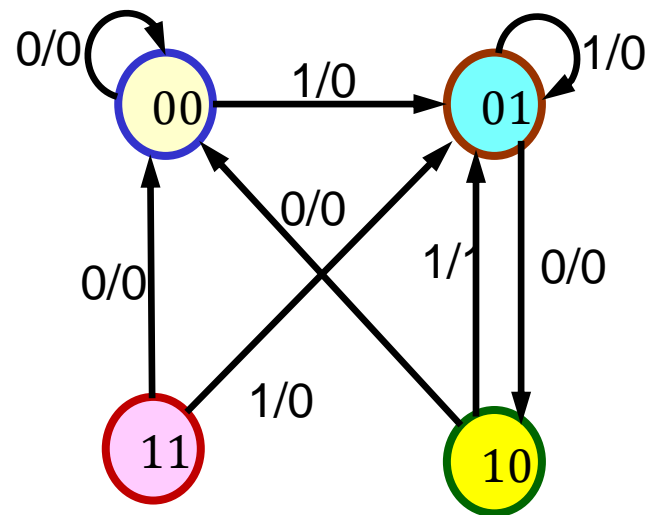


同步时序逻辑电路分析

□ 做时间图

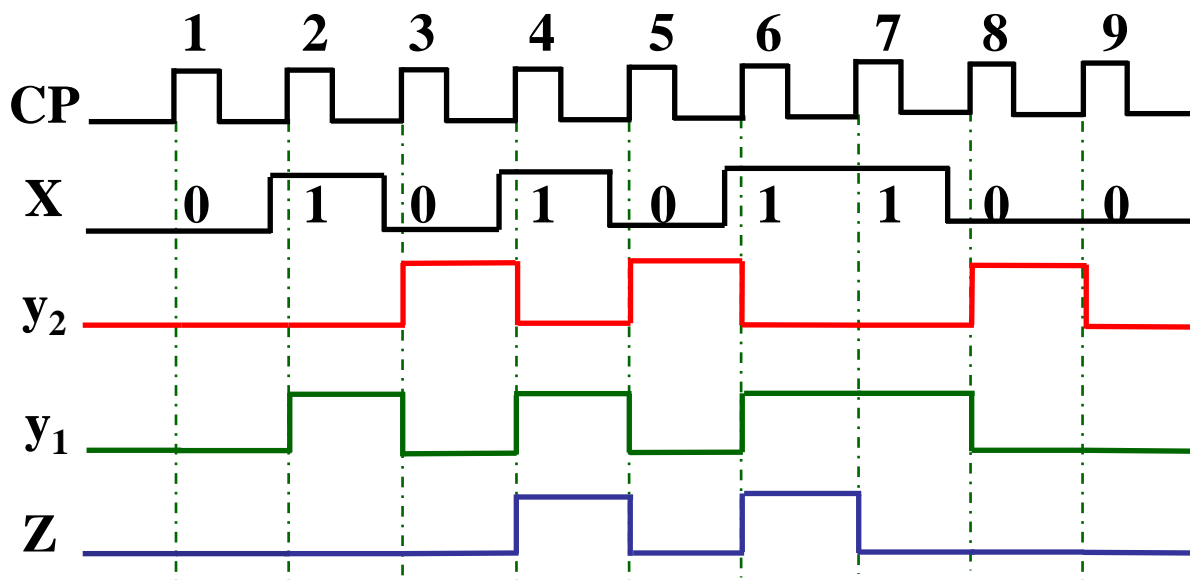
- 初始状态: $y_2 y_1 = 00$
- 输入 x 为电平信号, 典型输入序列为 010101100
- 做电路的状态响应序列:

CP	1	2	3	4	5	6	7	8	9
x	0	1	0	1	0	1	1	0	0
y_2	0	0	0	1	0	1	0	0	1
y_1	0	0	1	0	1	0	1	1	0
y_2^{n+1}	0	0	1	0	1	0	0	1	0
y_1^{n+1}	0	1	0	1	0	1	1	0	0
Z	0	0	0	1	0	1	0	0	0



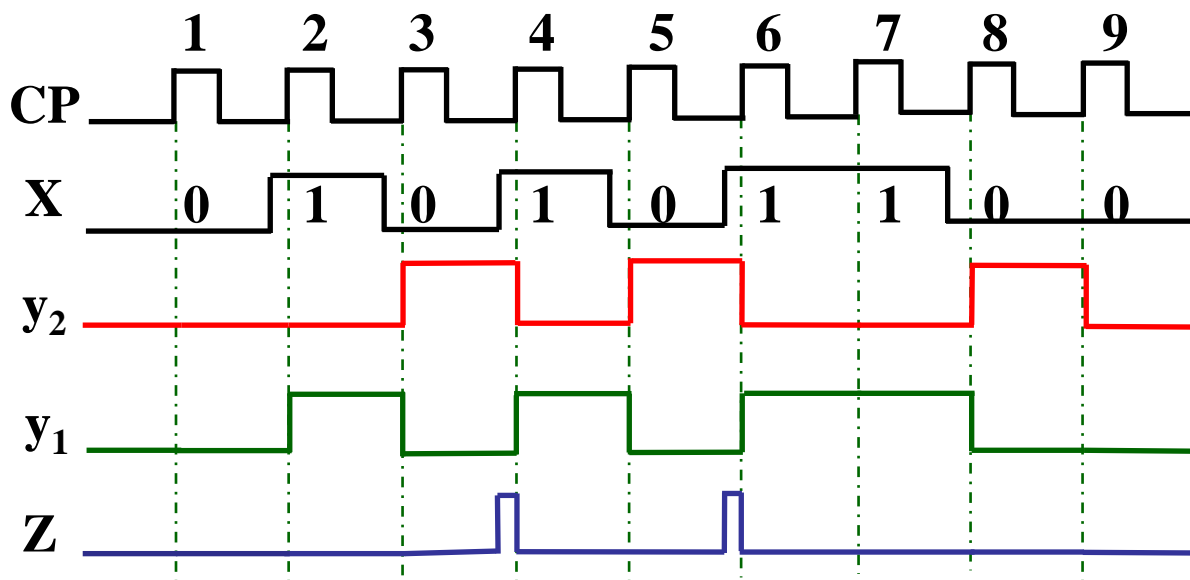
同步时序逻辑电路分析

CP	1	2	3	4	5	6	7	8	9
x	0	1	0	1	0	1	1	0	0
y_2	0	0	0	1	0	1	0	0	1
y_1	0	0	1	0	1	0	1	1	0
y_2^{n+1}	0	0	1	0	1	0	0	1	0
y_1^{n+1}	0	1	0	1	0	1	1	0	0
Z	0	0	0	1	0	1	0	0	0



同步时序逻辑电路分析

CP	1	2	3	4	5	6	7	8	9
x	0	1	0	1	0	1	1	0	0
y_2	0	0	0	1	0	1	0	0	1
y_1	0	0	1	0	1	0	1	1	0
y_2^{n+1}	0	0	1	0	1	0	0	1	0
y_1^{n+1}	0	1	0	1	0	1	1	0	0
Z	0	0	0	1	0	1	0	0	0



表格分析法

□ 函数表达式

$$- D_2 = \bar{x} \bar{y}_2 y_1 \quad D_1 = x \quad Z = xy_2 \bar{y}_1$$

□ 次态真值表

□ 状态表

□ 状态图

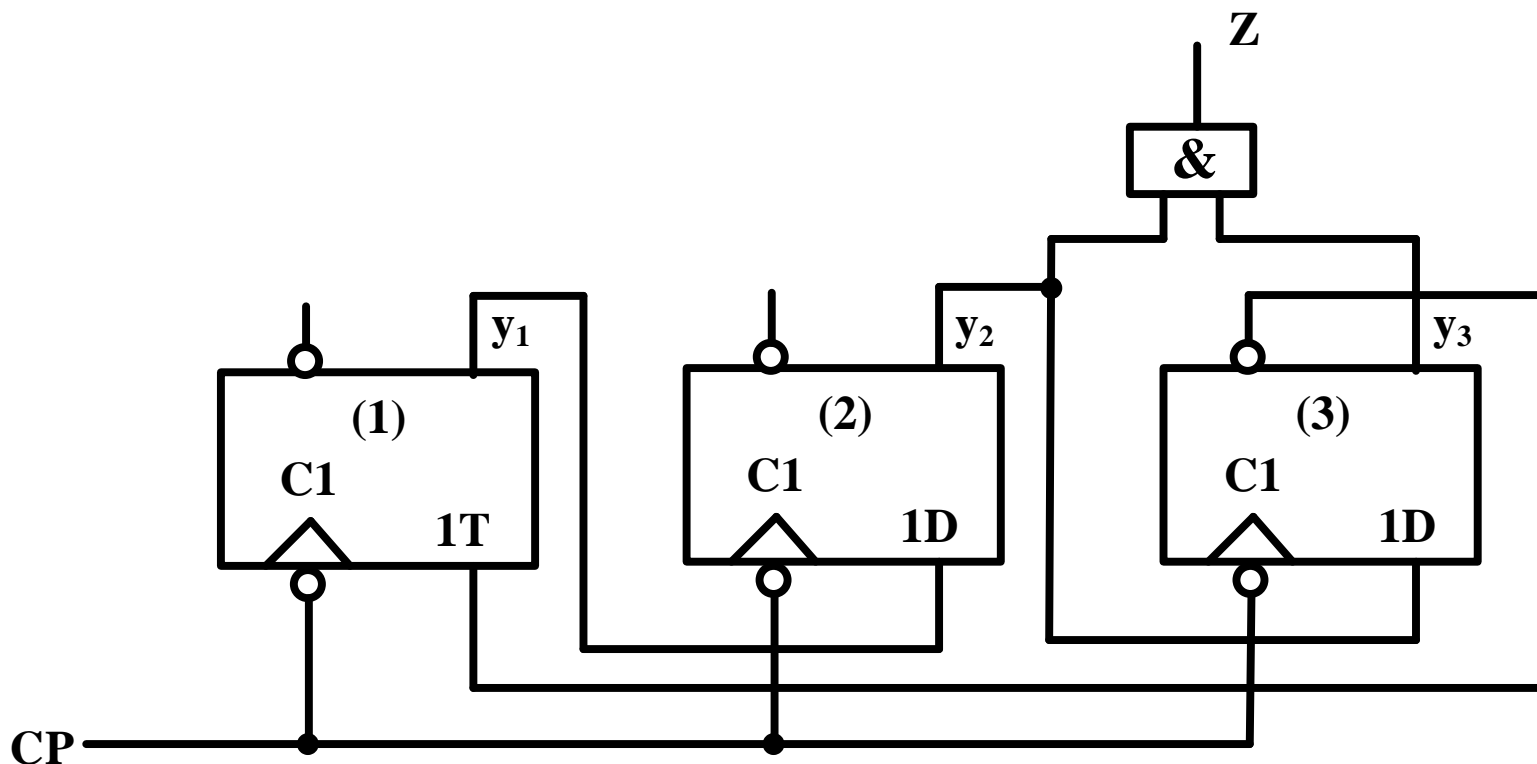
□ 时间图

□ 功能评述

– 可重叠101序列检测器

同步时序逻辑电路分析

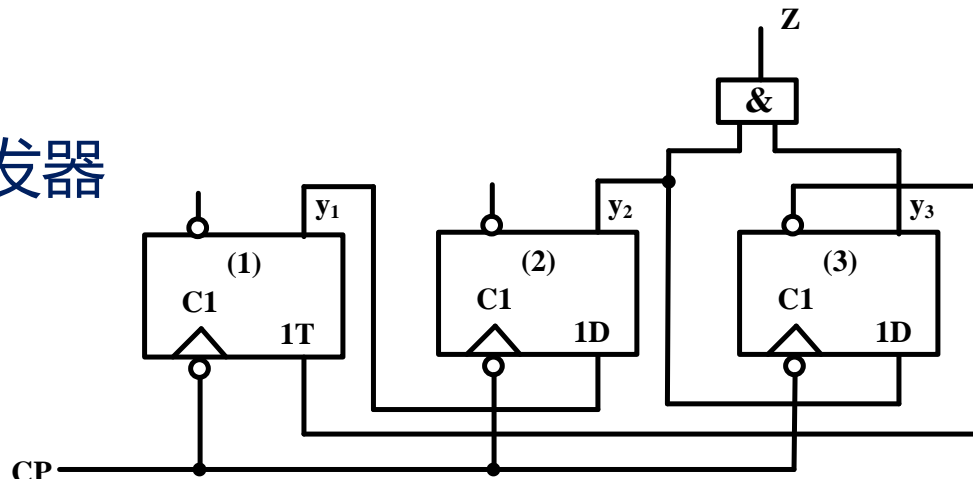
□ 例 分析下图所示同步时序逻辑电路



同步时序逻辑电路分析

□分析：

- 1个T触发器和2个D触发器
 - 同步时序逻辑电路
- 输入：无
- 电路的状态： $y_3y_2y_1$
- 电路的输出： Z
- Moore型电路



同步时序逻辑电路分析

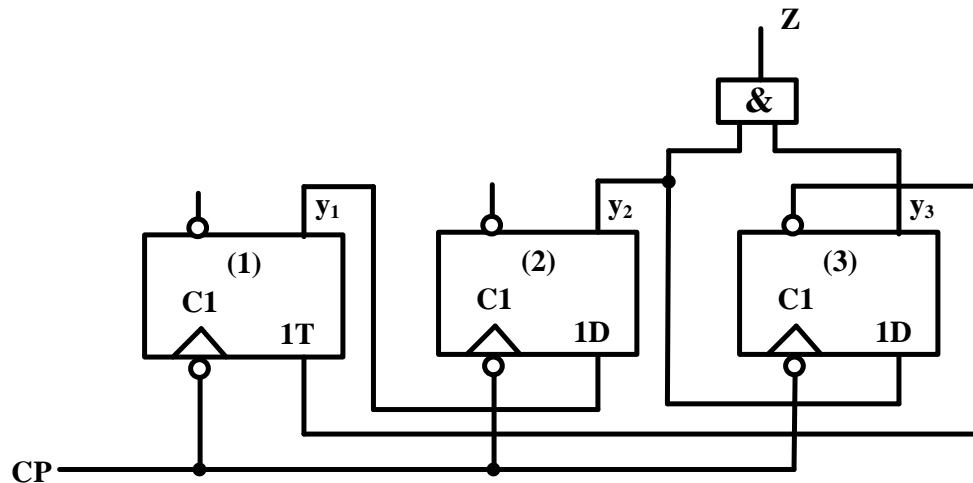
□分析：

– 函数表达式

- $T_1 = \overline{y_3}$
- $D_2 = y_1$
- $D_3 = y_2$
- $Z = y_3 y_2$

– 次态方程组

- $y_3^{n+1} = D_3 = y_2$
- $y_2^{n+1} = D_2 = y_1$
- $y_1^{n+1} = y_1 \oplus T_1 = y_1 \oplus \overline{y_3}$



同步时序逻辑电路分析

□分析：

$$y_3^{n+1} = y_2 \quad y_2^{n+1} = y_1 \quad y_1^{n+1} = y_1 \oplus \bar{y}_3 \quad Z = y_3 y_2$$

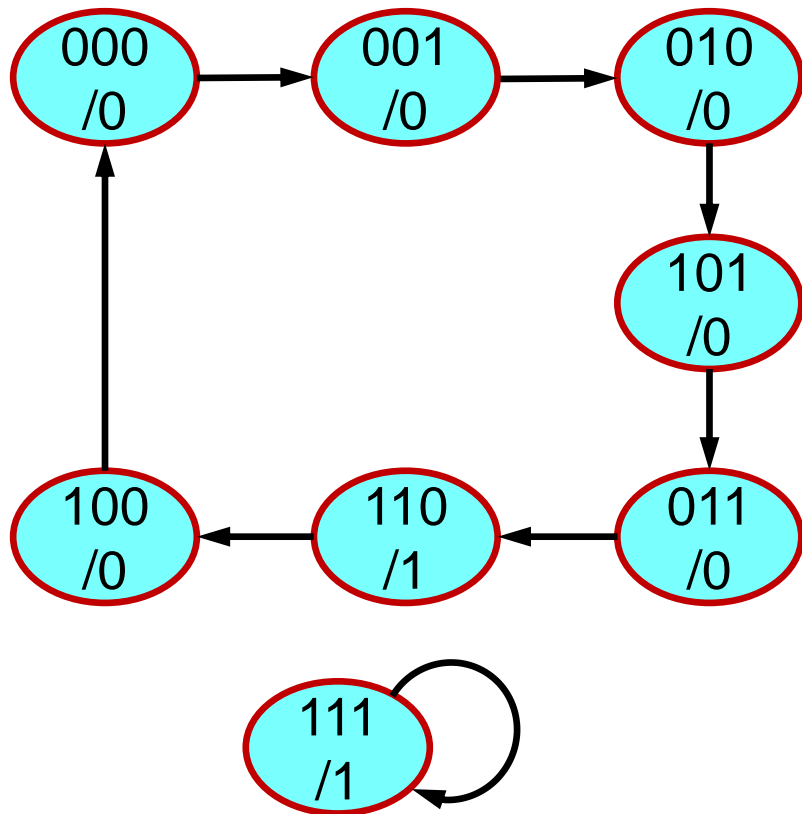
– 状态表

现态 $y_3 y_2 y_1$	次态 $y_3^{n+1} y_2^{n+1} y_1^{n+1}$	输出 Z
0 0 0	0 0 1	0
0 0 1	0 1 0	0
0 1 0	1 0 1	0
0 1 1	1 1 0	0
1 0 0	0 0 0	0
1 0 1	0 1 1	0
1 1 0	1 0 0	1
1 1 1	1 1 1	1

同步时序逻辑电路分析

□分析：

– 状态图



现态 $y_3y_2y_1$	次态 $y_3^{n+1}y_2^{n+1}y_1^{n+1}$	输出 Z
0 0 0	0 0 1	0
0 0 1	0 1 0	0
0 1 0	1 0 1	0
0 1 1	1 1 0	0
1 0 0	0 0 0	0
1 0 1	0 1 1	0
1 1 0	1 0 0	1
1 1 1	1 1 1	1

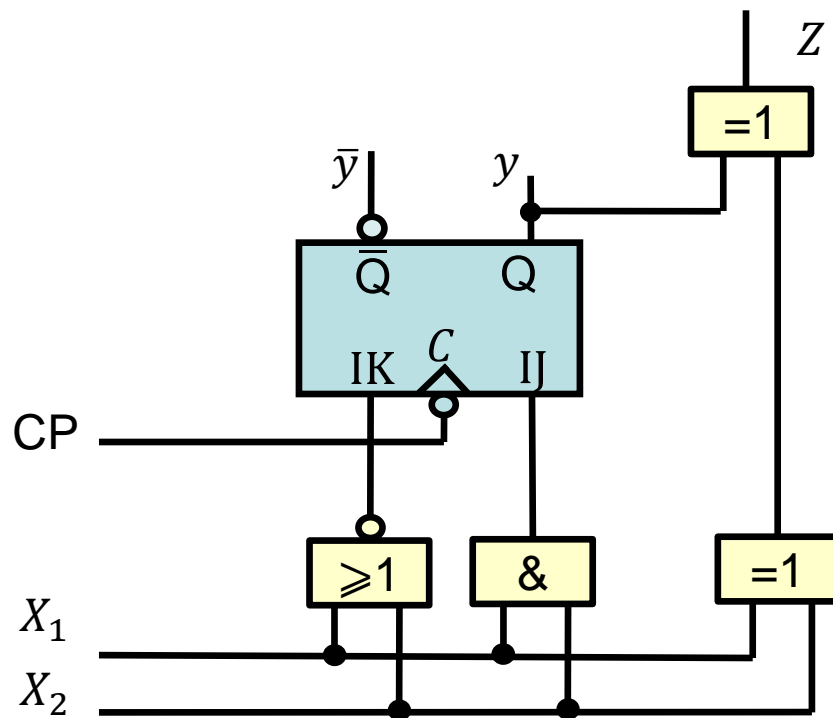
同步时序逻辑电路分析

□分析：

- 逻辑功能：不能自启动（存在挂起）的同步模7计数器，进位输出 $Z=1$

同步时序逻辑电路分析

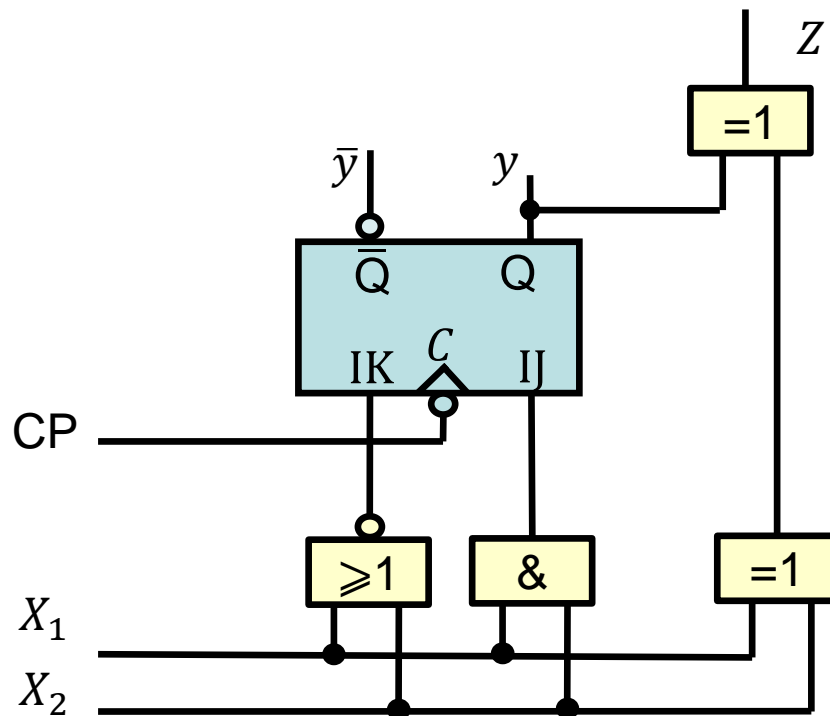
□ 例 分析下图所示同步时序逻辑电路



同步时序逻辑电路分析

□分析：

- 一个JK触发器
 - 同步时序逻辑电路
- 异或门+或非门+与门
- 输入： X_1, X_2
- 电路的状态： y
- 电路的输出： Z
- Mealy型电路



同步时序逻辑电路分析

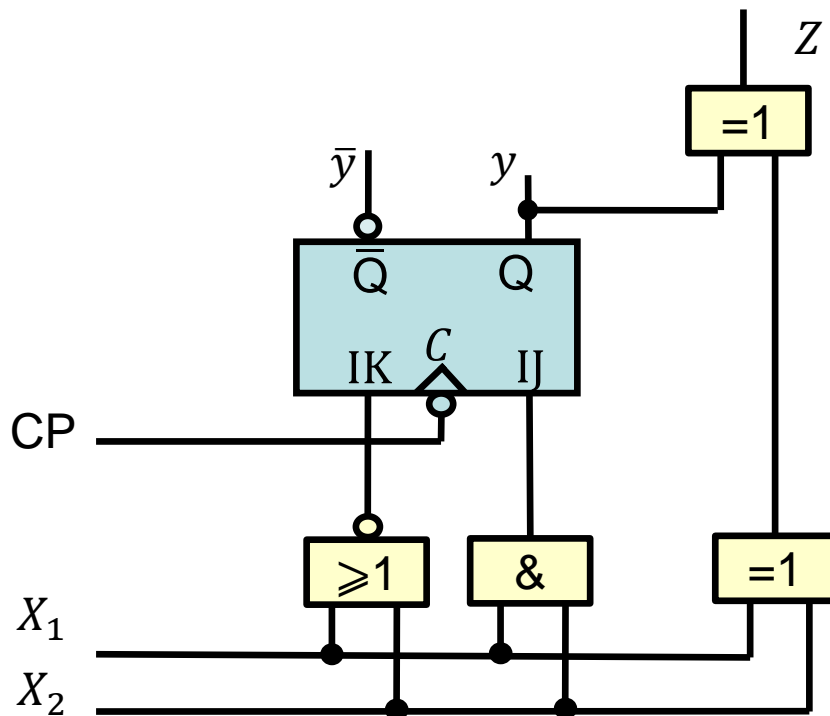
□分析：

– 函数表达式

- $J = X_1X_2$
- $K = \overline{X_1 + X_2}$
- $z = X_1 \oplus X_2 \oplus y$

– 次态方程组

$$\begin{aligned} y^{n+1} &= J\bar{y} + \bar{K}y \\ &= X_1X_2\bar{y} + \overline{\overline{X_1 + X_2}}y \\ &= X_1X_2\bar{y} + X_1y + X_2y \end{aligned}$$



同步时序逻辑电路分析

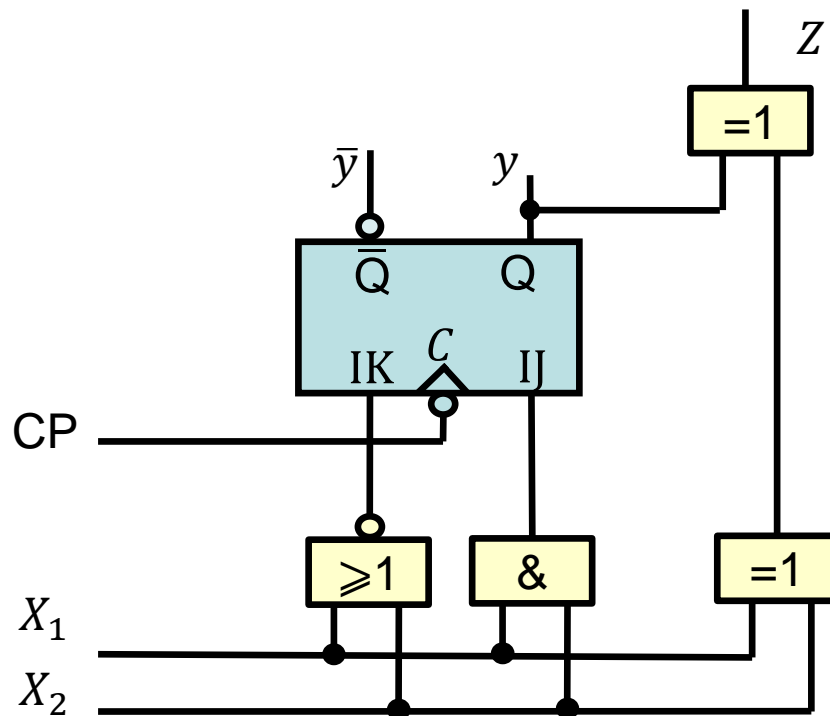
□分析：

– 次态方程组

- $y^{n+1} = X_1X_2\bar{y} + X_1y + X_2y$

– 状态表

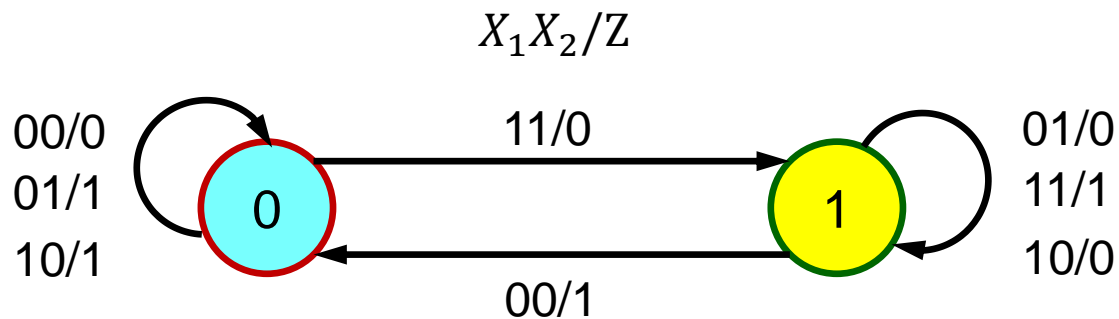
– 状态图



现态 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

同步时序逻辑电路分析

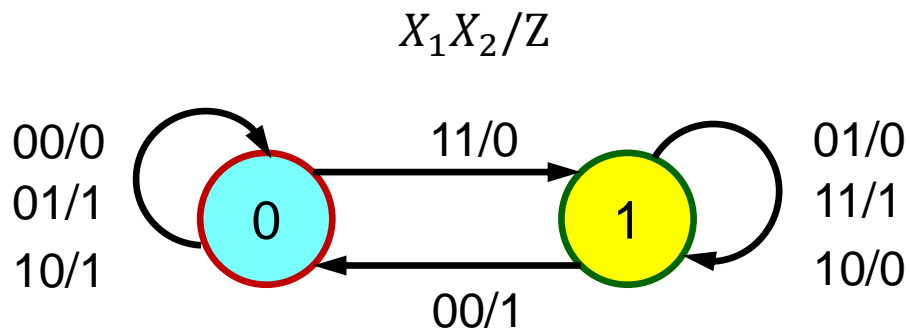
现态 y	次态/输出 (y^{n+1}/Z)			
	$X_1X_2=00$	$X_1X_2=01$	$X_1X_2=11$	$X_1X_2=10$
0	0/0	0/1	1/0	0/1
1	0/1	1/0	1/1	1/0



同步时序逻辑电路分析

□ 做时间图

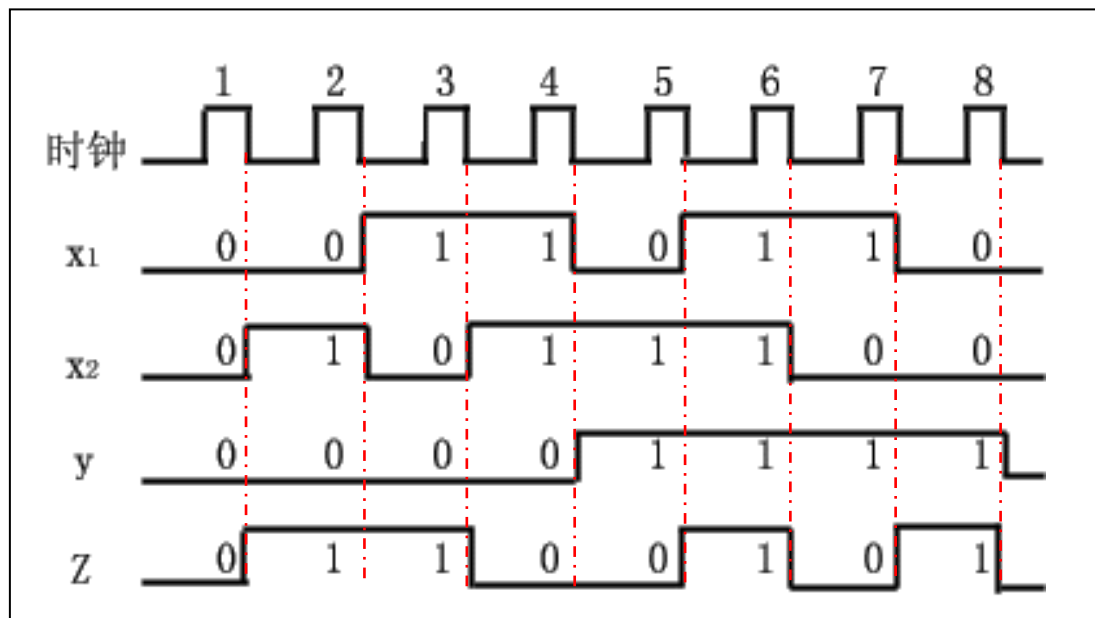
- 初始状态: $y=0$
- $X_1 = 00110110$
- $X_2 = 01011100$
- 做电路的状态响应序列:



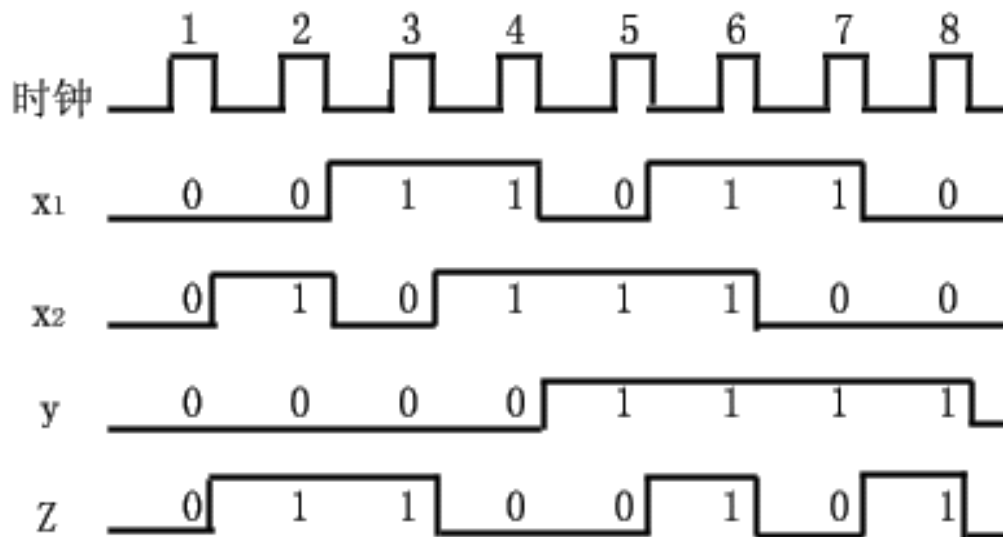
CP	1	2	3	4	5	6	7	8
X_1	0	0	1	1	0	1	1	0
X_2	0	1	0	1	1	1	0	0
y	0	0	0	0	1	1	1	1
Z	0	1	1	0	0	1	0	1

同步时序逻辑电路分析

CP	1	2	3	4	5	6	7	8
X_1	0	0	1	1	0	1	1	0
X_2	0	1	0	1	1	1	0	0
y	0	0	0	0	1	1	1	1
Z	0	1	1	0	0	1	0	1



同步时序逻辑电路分析



时钟节拍:	8	7	6	5	4	3	2	1
x_1 (被加数):	0	1	1	0	1	1	0	0
x_2 (加数):	0	0	1	1	1	0	1	0
y (进位):	1	1	1	1	0	0	0	0
Z (和数):	1	0	1	0	0	1	1	0

同步时序逻辑电路分析

□分析：

– 次态方程组

- $y^{n+1} = X_1X_2\bar{y} + X_1y + X_2y$

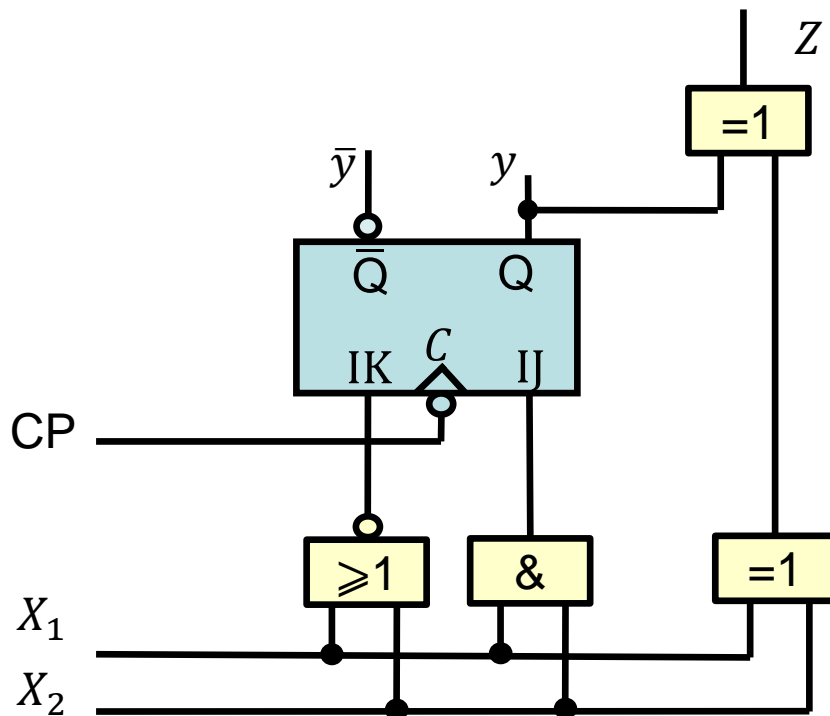
– 状态表

– 状态图

– 时间图

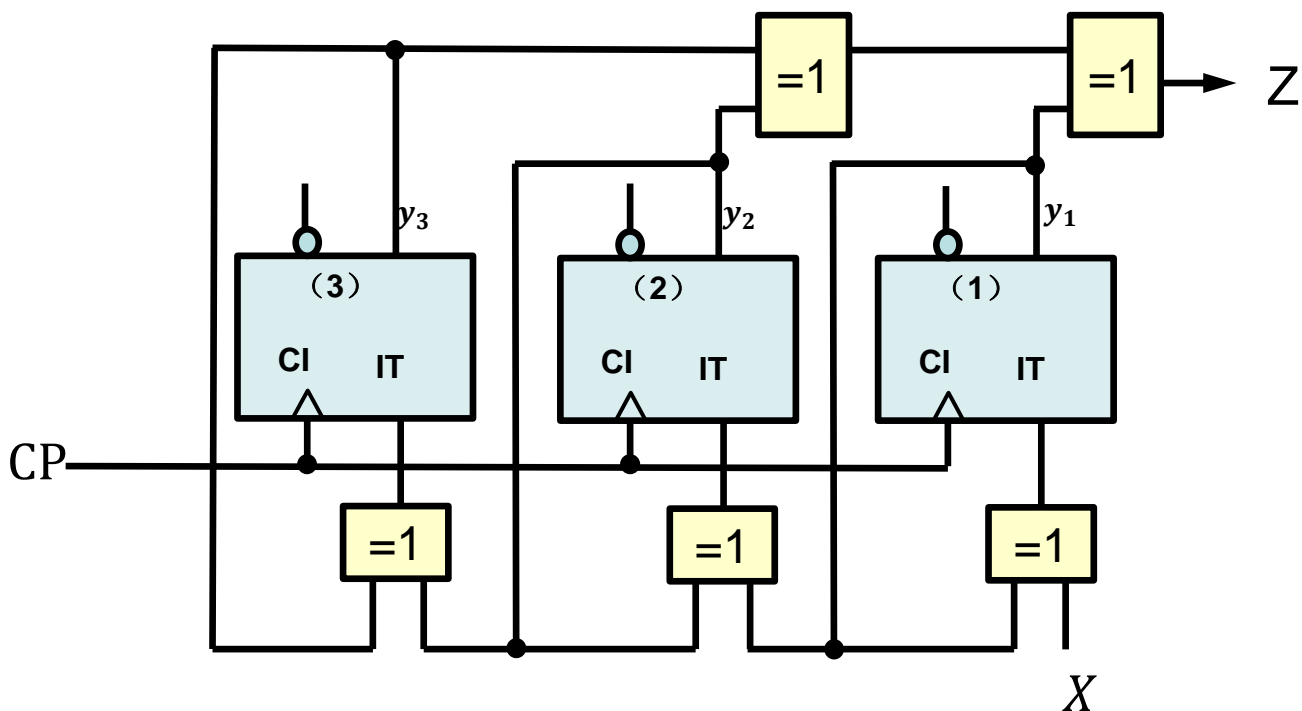
– 功能评述

- 串行加法器的功能



同步时序逻辑电路分析

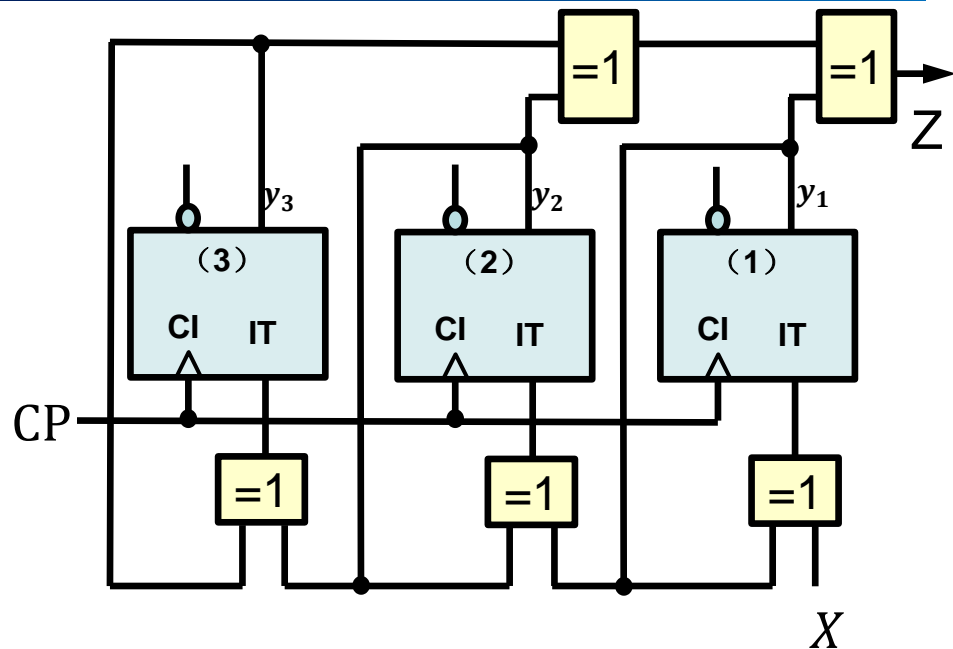
□例 试用代数法分析下图所示同步时序逻辑电路



同步时序逻辑电路分析

□ 分析：

- 三个T触发器
 - 相同的时钟端
 - 同步时序逻辑电路
- 五个异或门
- 输入：x
- 电路的状态： y_3 、 y_2 、 y_1
- 电路的输出： Z
- Moore型电路



同步时序逻辑电路分析

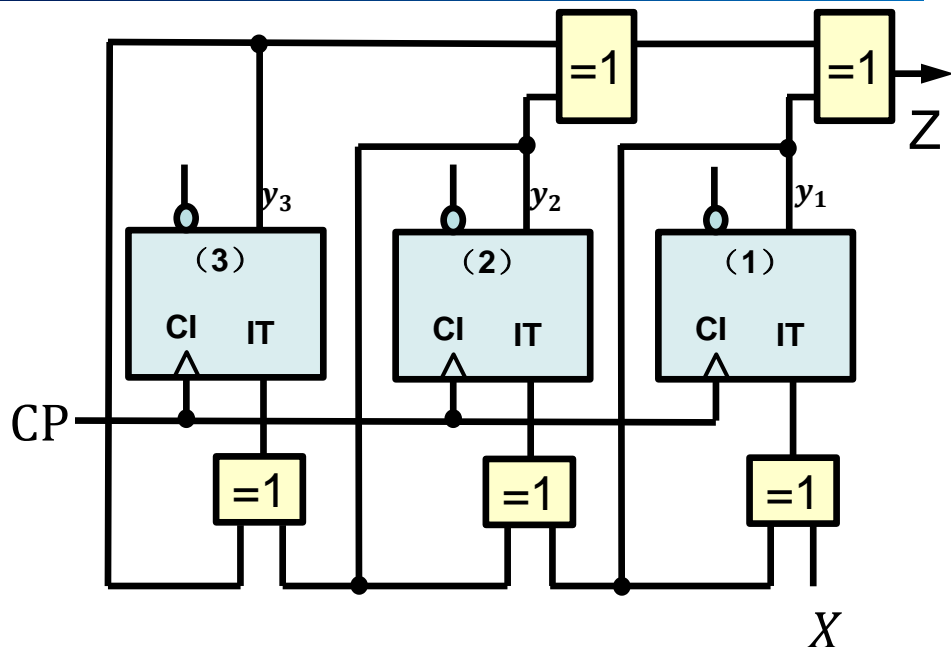
□ 分析：

— 激励和输出函数表达式

- $T_3 = y_3 \oplus y_2$
- $T_2 = y_2 \oplus y_1$
- $T_1 = y_1 \oplus x$
- $z = y_3 \oplus y_2 \oplus y_1$

— 次态方程组

- $y_3^{n+1} = T_3 \oplus y_3 = y_3 \oplus y_2 \oplus y_3 = y_2$
- $y_2^{n+1} = T_2 \oplus y_2 = y_2 \oplus y_1 \oplus y_2 = y_1$
- $y_1^{n+1} = T_1 \oplus y_1 = y_1 \oplus x \oplus y_1 = x$



同步时序逻辑电路分析

□ 分析：

– 次态方程组

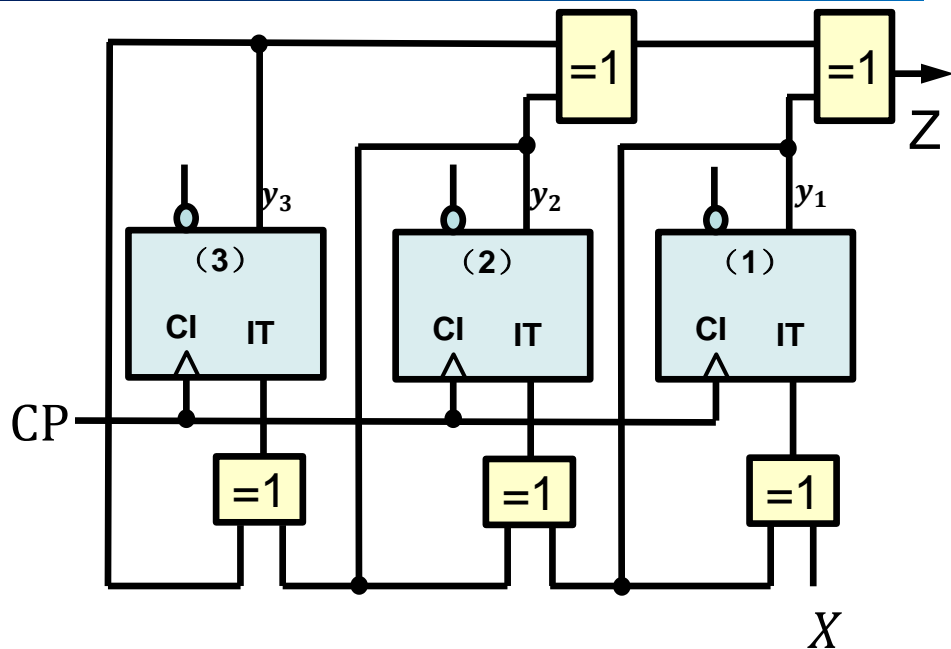
- $y_3^{n+1} = y_2$
- $y_2^{n+1} = y_1$
- $y_1^{n+1} = x$

– 输出函数

- $z = y_3 \oplus y_2 \oplus y_1$

– 功能分析

- 移位寄存器



同步时序逻辑电路设计

实际问题分析时，可视具体情况灵活运用，根据给定逻辑电路的复杂程度不同，通常可以省去某些步骤。例如，列次态真值表或画时间图等。

提纲

1

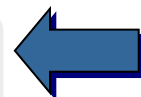
时序逻辑电路概述

2

同步时序逻辑电路分析

3

同步时序逻辑电路设计



同步时序逻辑电路设计

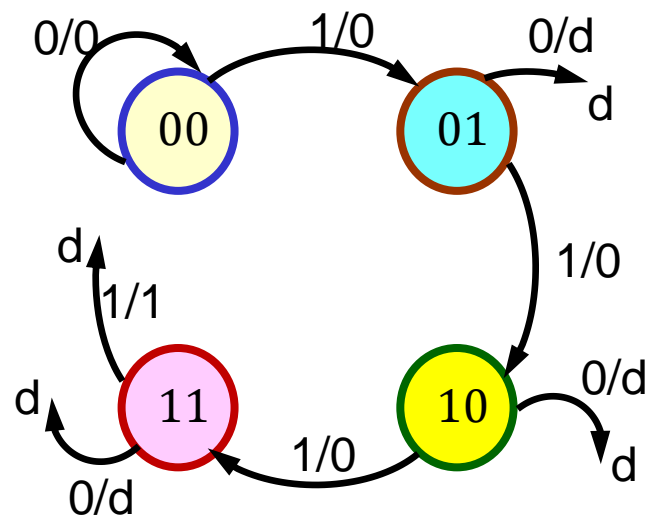
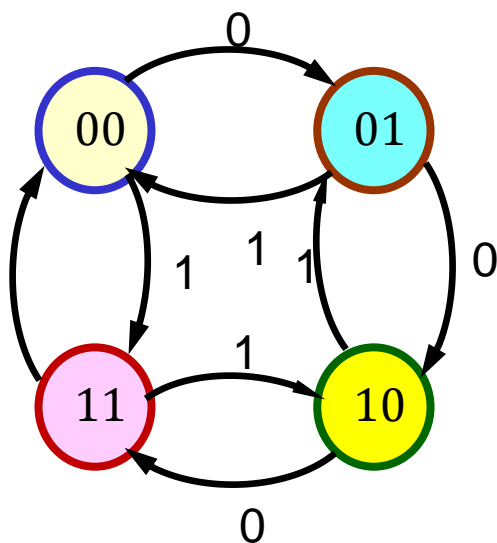
□ 完全给定同步时序逻辑电路

- 电路在不同输入取值下都有**确定**的次态和输出

□ 不完全给定同步时序逻辑电路

- 电路中存在**不确定**的次态或输出

同步时序逻辑电路设计



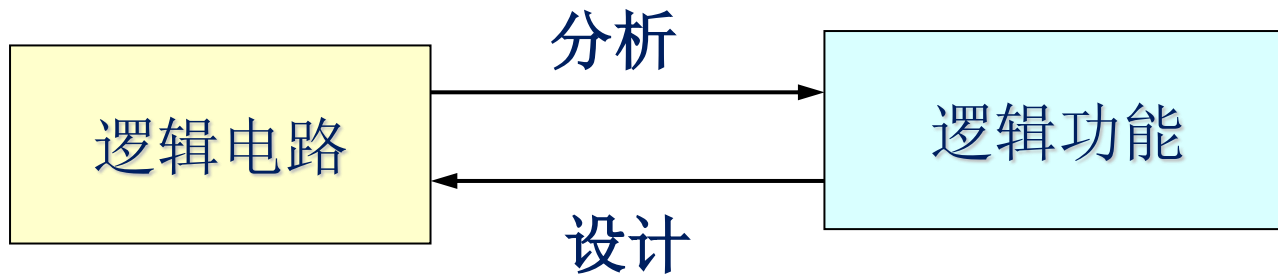
同步时序逻辑电路设计

□ 定义

- 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

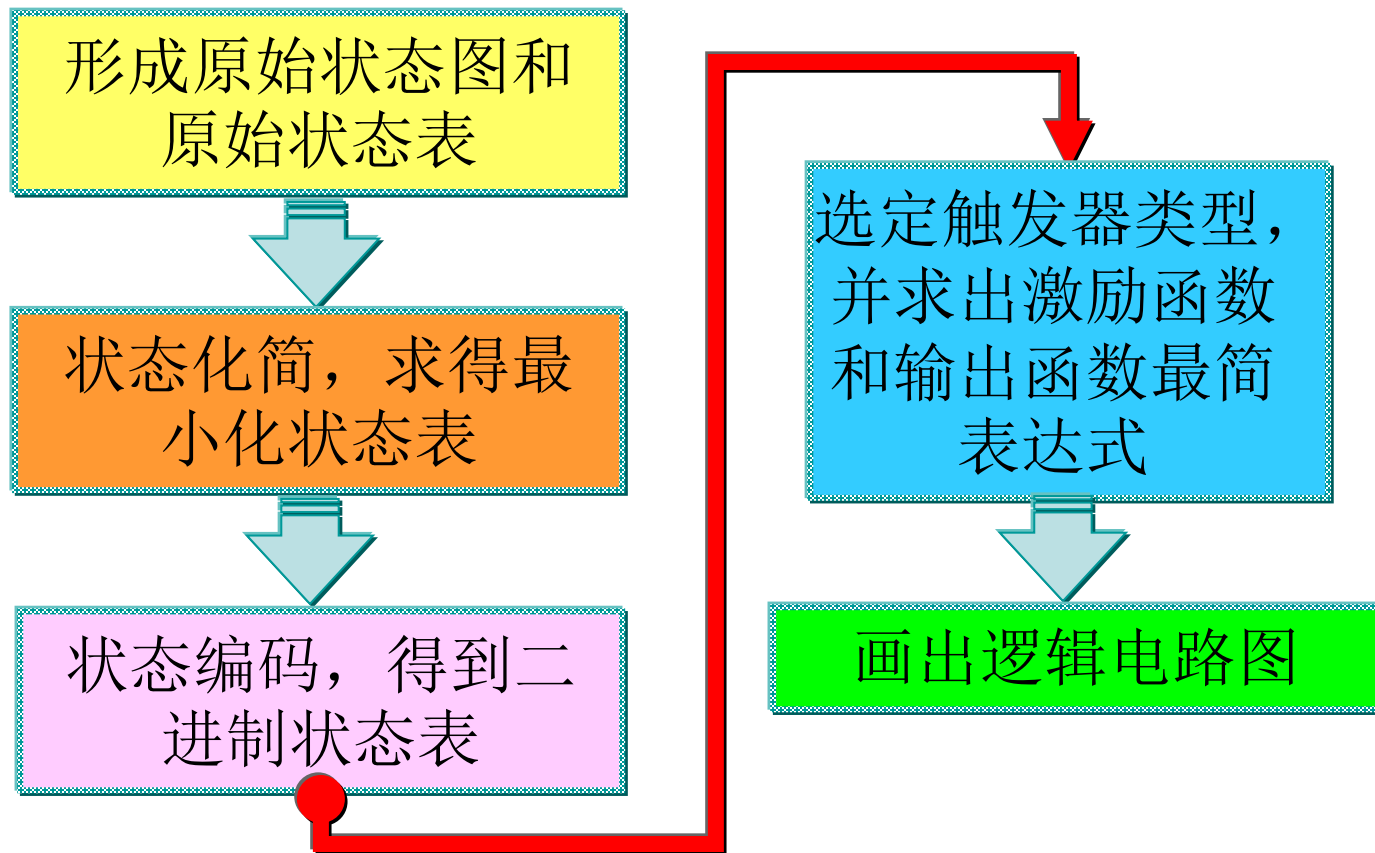
□ 目标

- 使用尽可能少的触发器和逻辑门实现预定的逻辑要求



同步时序逻辑电路设计

□ 设计的一般步骤



同步时序逻辑电路设计

1 原始状态图和状态表

2 状态化简

3 状态编码

4 激励函数和输出函数

5 逻辑电路图

同步时序逻辑电路设计

□ 形成原始状态图和原始状态表

- 根据对设计要求的文字描述，抽象出电路的输入、输出及状态之间的关系
- 形成状态图和状态表
- 开始得到的状态图和状态表可能包含多余的状态
- 先画状态图后作状态表
 - 状态图比状态表更形象、灵活
 - 一般用字母或数字表示状态

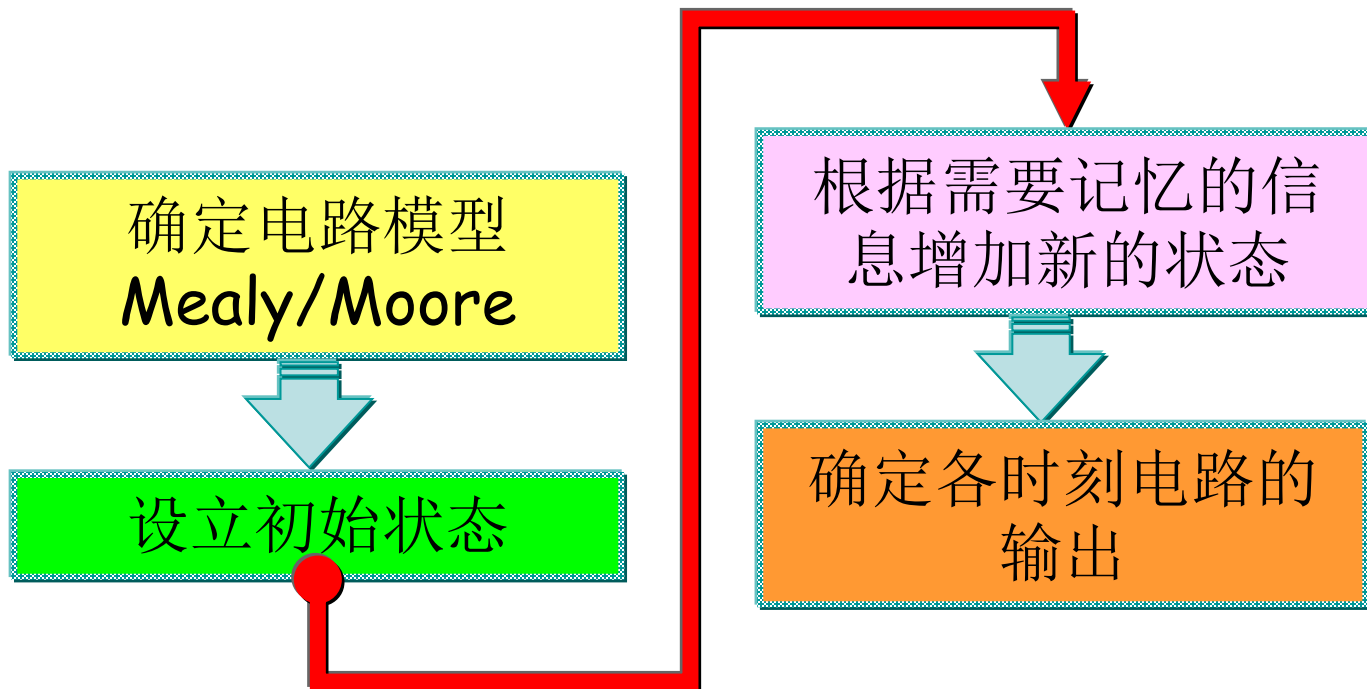
同步时序逻辑电路设计

□ 形成原始状态图和原始状态表

- 根据对设计要求的文字描述，抽象出电路的输入、输出及状态之间的关系
- 先画状态图后作状态表
- 任务
 - 确定电路模型
 - 设立初始状态
 - 增加新的状态
 - 确定各时刻电路的输出

原始状态表和状态图

□ 建立的一般步骤



原始状态表和状态图

□ 确定电路模型

- 将电路设计成哪种模型， Mealy/Moore?
 - 有的问题已由设计要求规定，有的问题可由设计者选择
 - 不同的模型对应的电路结构不同
 - 根据问题中的信号形式、电路所需器件的多少等综合考虑

□ 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
- 从初始状态出发考虑在各种输入作用下的状态转移和输出响应

原始状态表和状态图

□ 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
- 若在某个状态下出现的输入信号作用的结果能用已有状态表示，则应转向已有状态
- 仅当某个状态下出现的输入信号作用的结果不能用已有状态表示时，才令其转向新的状态

原始状态表和状态图

□ 确定各时刻电路的输出

- 时序逻辑电路的功能是通过输出对输入的响应来体现的
- 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

□ 注意

- 状态数目能否达到最少无关紧要，因为可以对它再进行状态化简
- 设计者应把清晰、正确地描述设计要求放在第一位

原始状态表和状态图

□ 设计一个模5可逆计数器

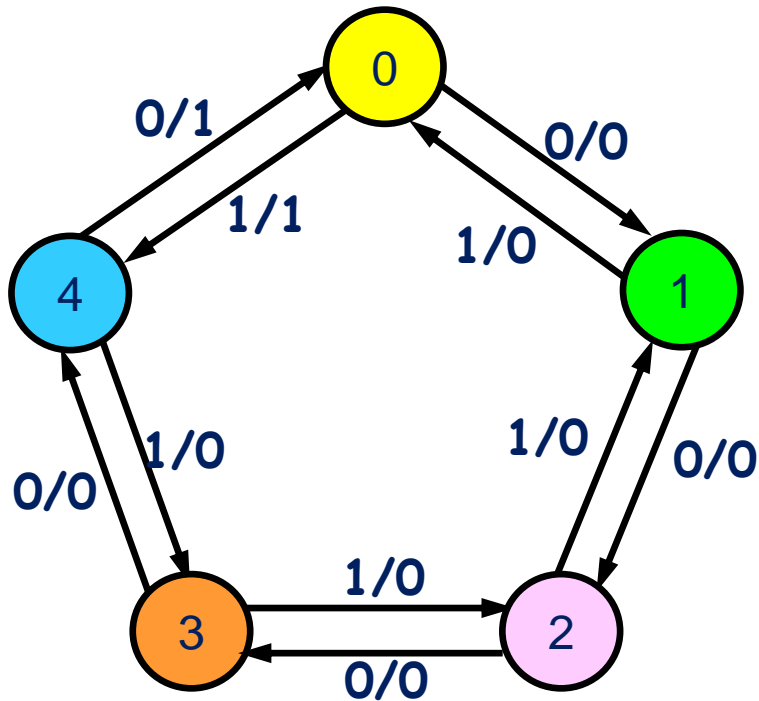
- 一个输入 x 和一个输出 Z
- 输入 x 为加、减控制信号
- 当 $x=0$ 时，计数器在时钟脉冲作用下进行加1计数
- 当 $x=1$ 时，计数器在时钟脉冲作用下进行减1计数
- 输出 Z 为进位或借位信号

试建立该计数器的Mealy型原始状态图和状态表。

原始状态表和状态图

□分析

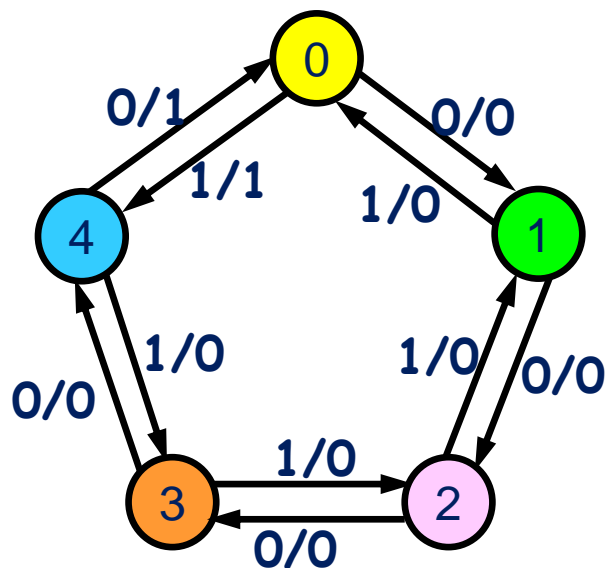
- 电路类型: Mealy
- 初始状态: 0



原始状态表和状态图

□分析

- 电路类型: Mealy
- 初始状态: 0
- 原始状态表



现态	次态/输出	
	X=0	X=1
0	1/0	4/1
1	2/0	0/0
2	3/0	1/0
3	4/0	2/0
4	0/1	3/0

原始状态表和状态图

- 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

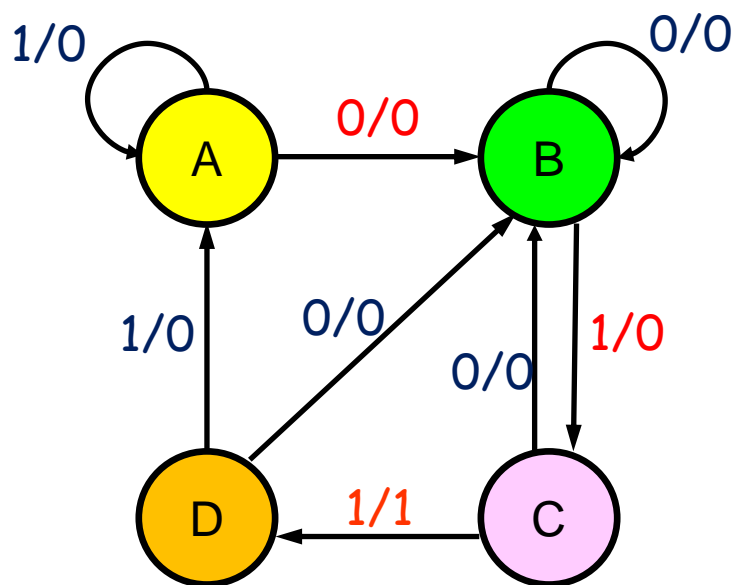
输入x: 1 0 1 0 1 1 1 0 0 1 1 0

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。

原始状态表和状态图

□ Mealy型

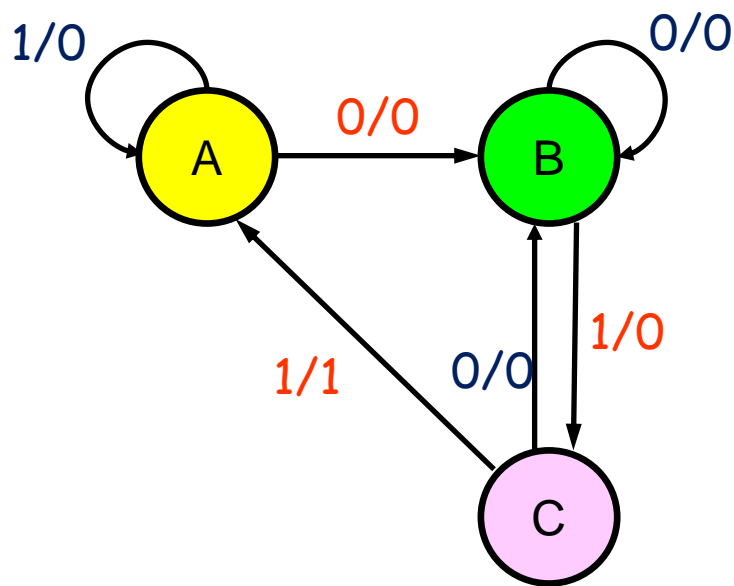


现态	次态/输出	
	X=0	X=1
A	B/0	A/0
B	B/0	C/0
C	B/0	D/1
D	B/0	A/0

思考：需要几个触发器？D是否可以省略？

原始状态表和状态图

□ Mealy型

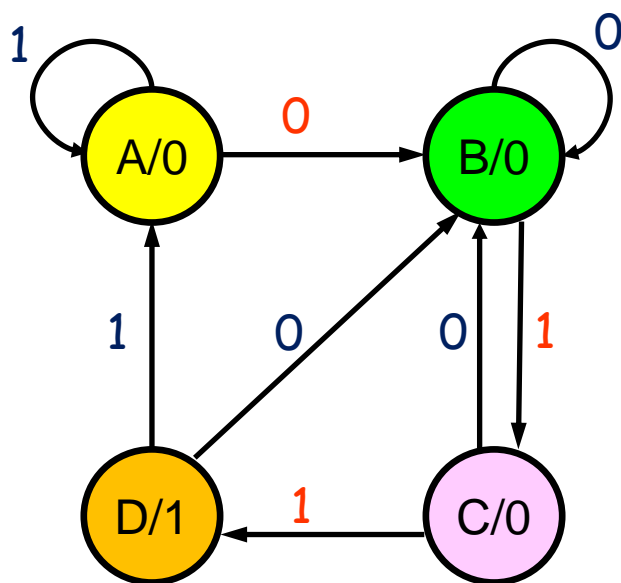


现态	次态/输出	
	X=0	X=1
A	B/0	A/0
B	B/0	C/0
C	B/0	A/1

思考：需要几个触发器？

原始状态表和状态图

□ Moore型

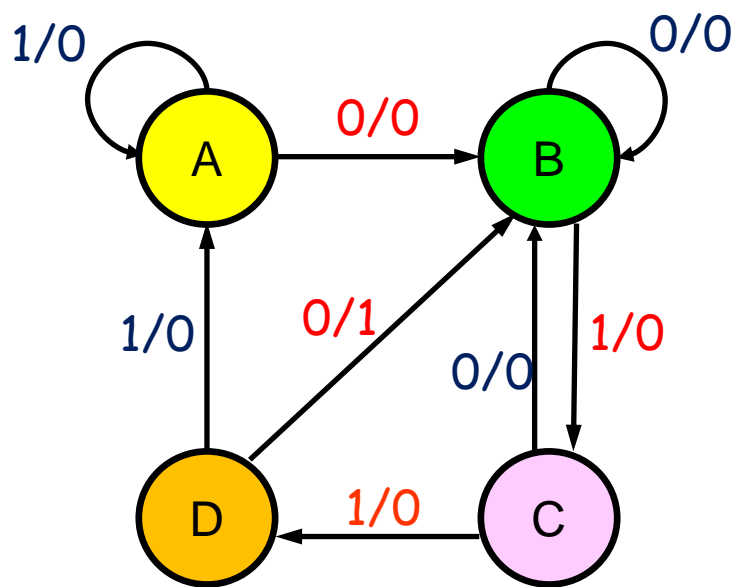


现态	次态/		输出
	X=0	X=1	Z
A	B	A	0
B	B	C	0
C	B	D	0
D	B	A	1

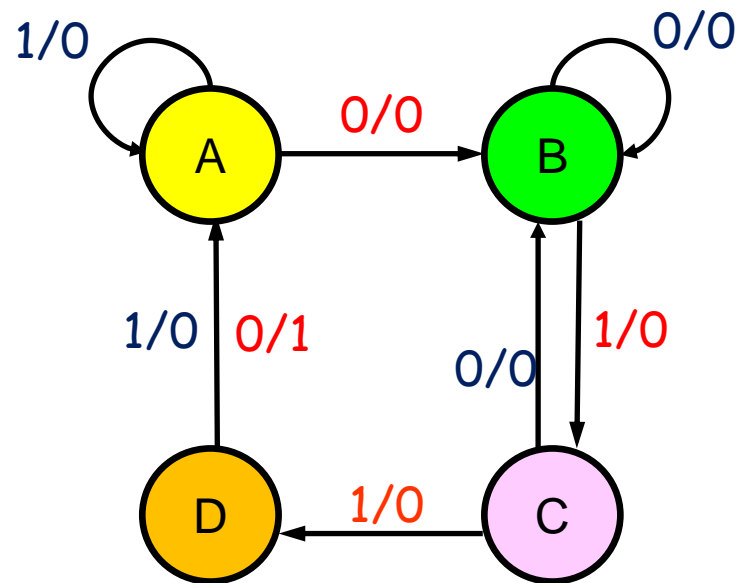
思考：需要几个触发器？D是否可以省略？

原始状态表和状态图

□思考：如果是0110序列检测器（Mealy型）？



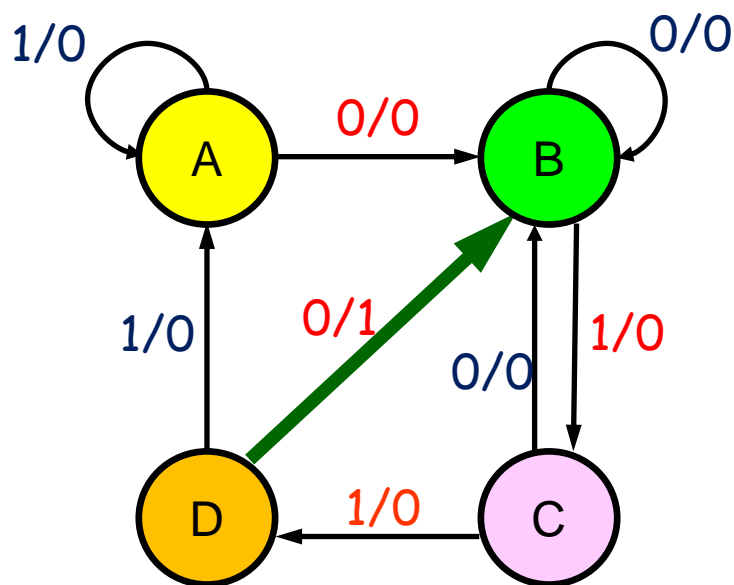
可重叠的0110



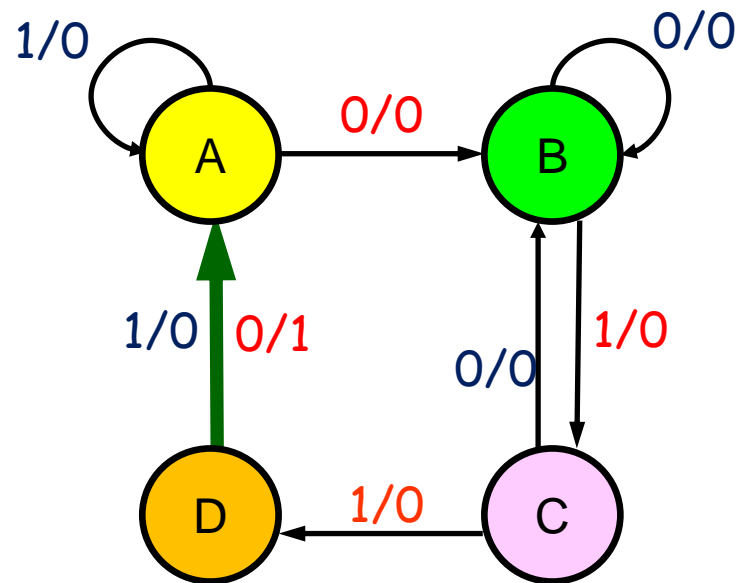
不可重叠的0110

原始状态表和状态图

□思考：如果是0110序列检测器（Mealy型）？



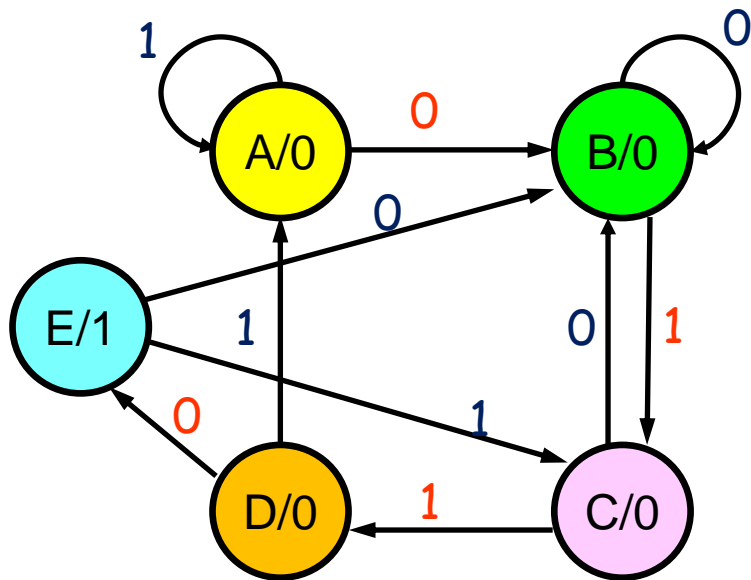
可重叠的0110



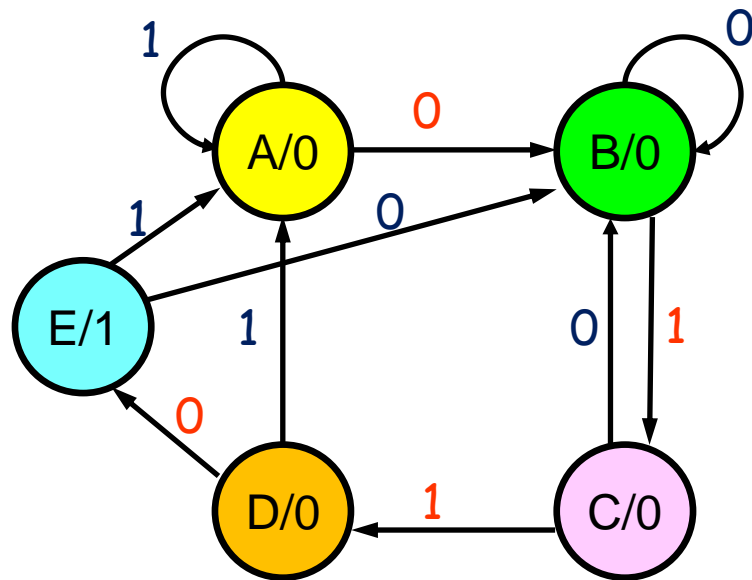
不可重叠的0110

原始状态表和状态图

□思考：如果是0110序列检测器（Moore型）？



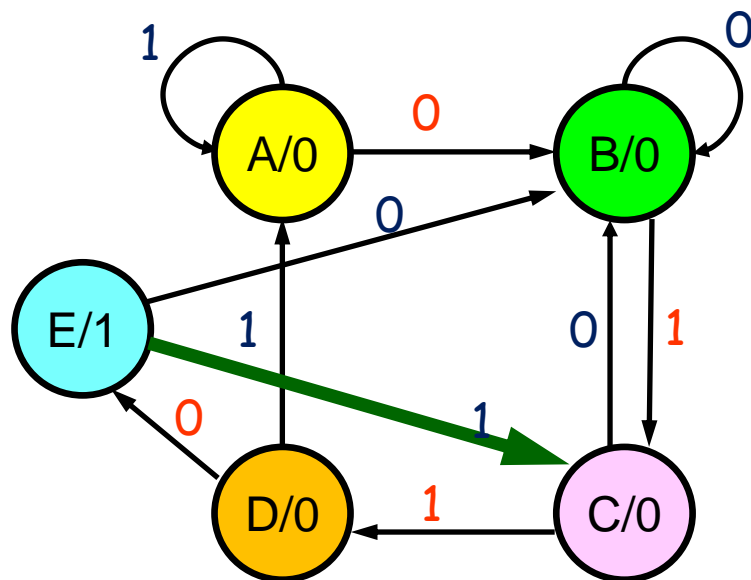
可重叠的0110



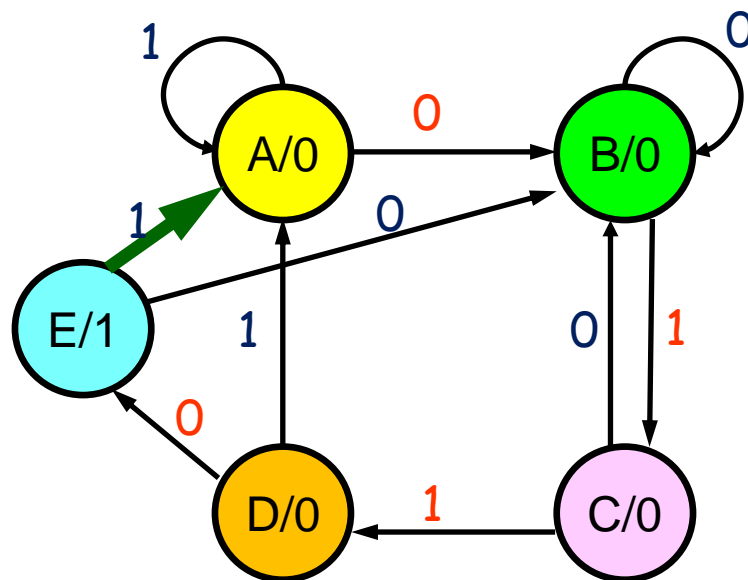
不可重叠的0110

原始状态表和状态图

□思考：如果是0110序列检测器（Moore型）？



可重叠的0110



不可重叠的0110

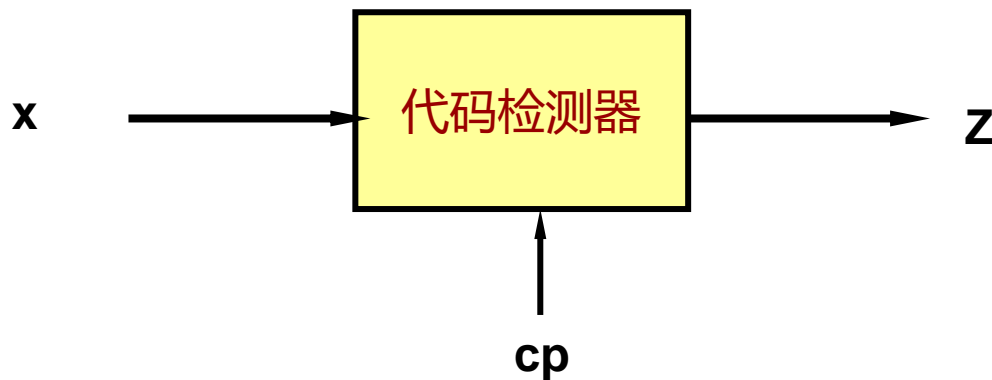
原始状态图和状态表

□ 总结

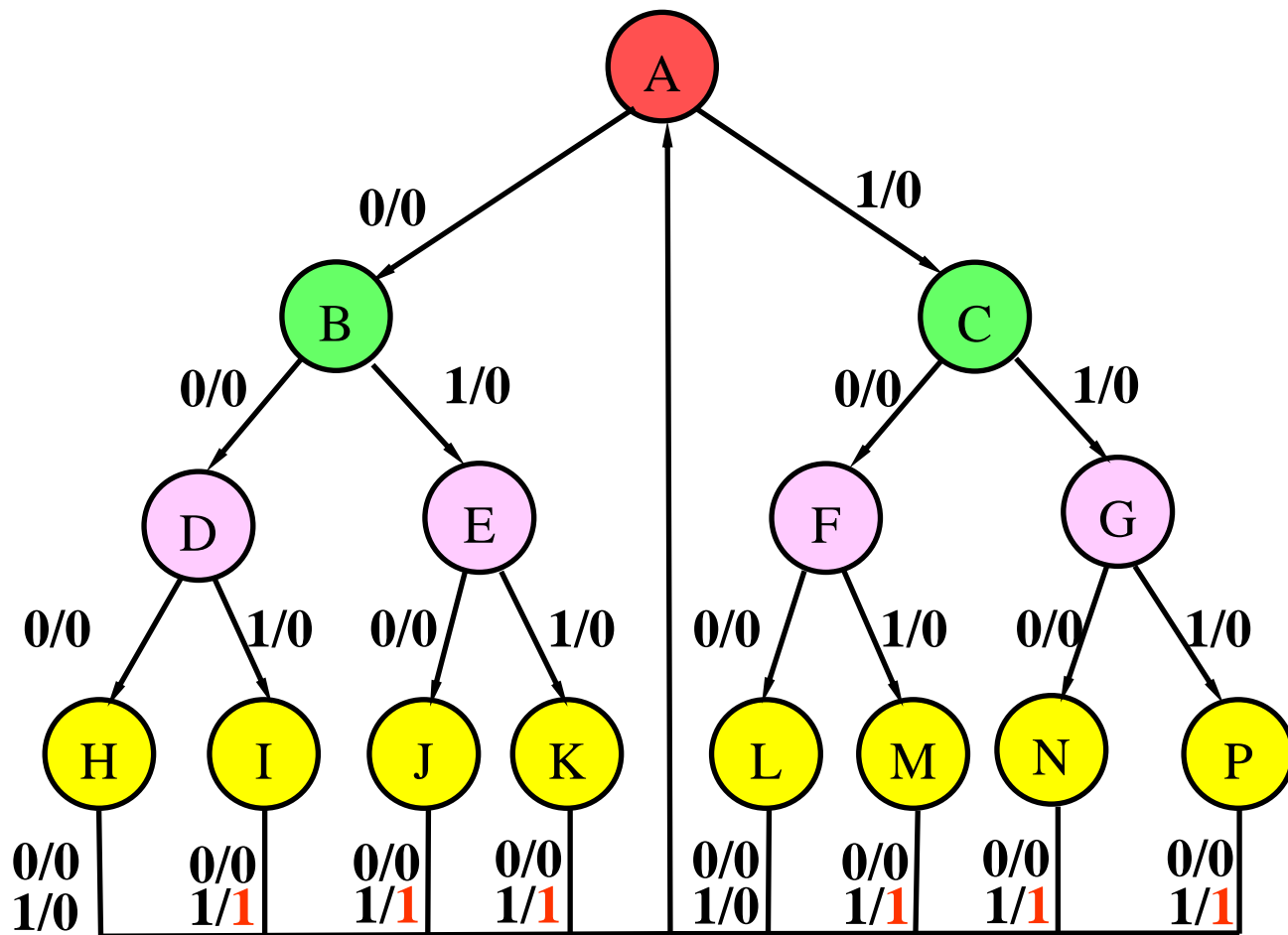
- 一个序列检测器所需要的状态数与要识别的序列长度相关
- 序列越长，需要记忆的代码位数越多，状态数也就越多
- Mealy型一般比Moore型所需状态数少
- Mealy型不一定比Moore型所需触发器少

原始状态图和状态表

- 设计一个代码检测器，用于检测串行输入的8421码
 - 其输入的顺序是先低位后高位
 - 当出现非法数字(即输入1010, 1011, 1100, 1101, 1110, 1111)时，电路的输出为1
 - 试作出该时序电路的Mealy模型状态图和状态表



原始状态图和状态表



原始状态图和状态表

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

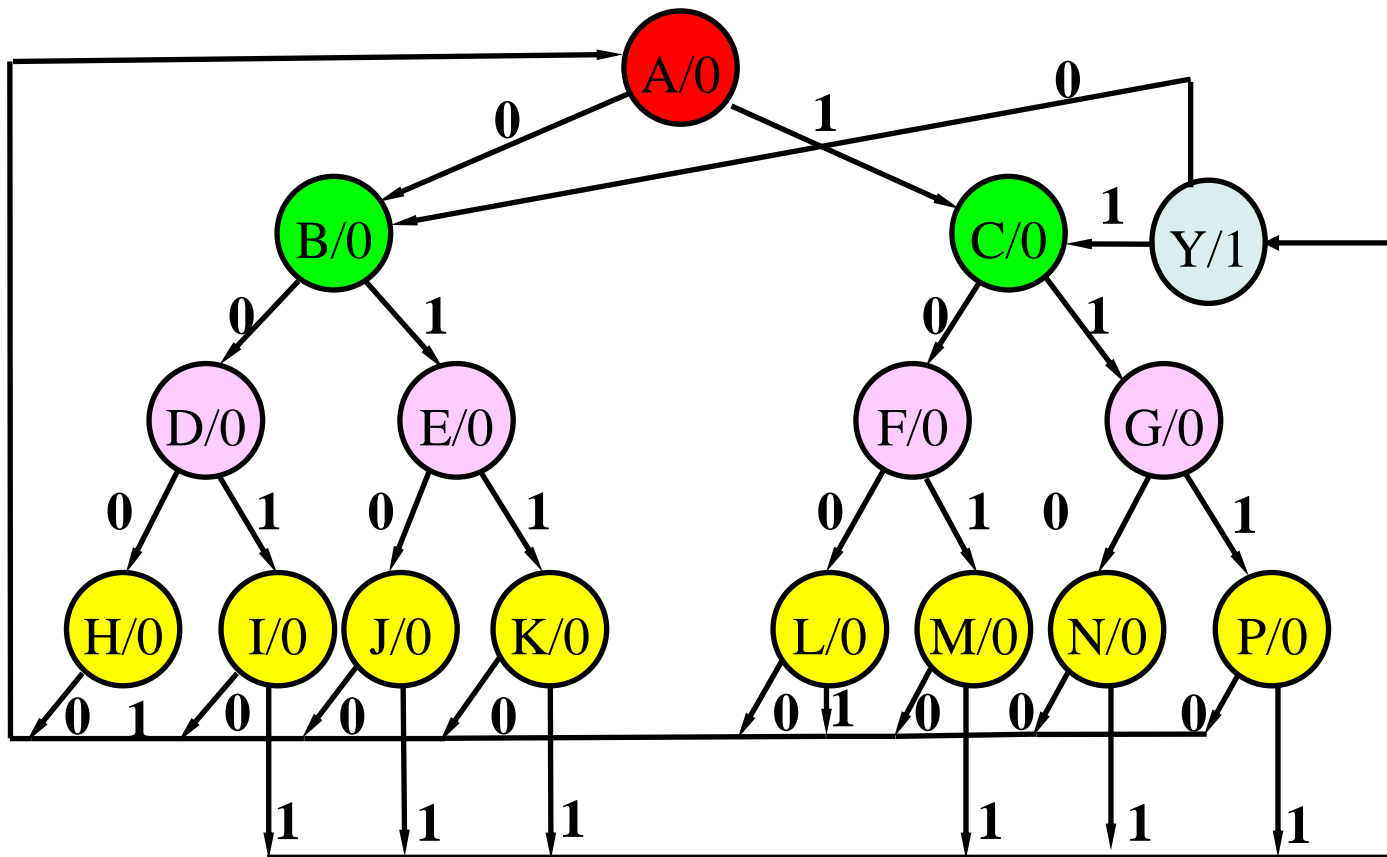
原始状态图和状态表

□思考:

- 代码检测器与序列检测器的主要区别是什么?
- 若将该代码检测器设计成Moore型同步时序电路
 - 该如何建立原始状态图?
 - 需增加几个状态?

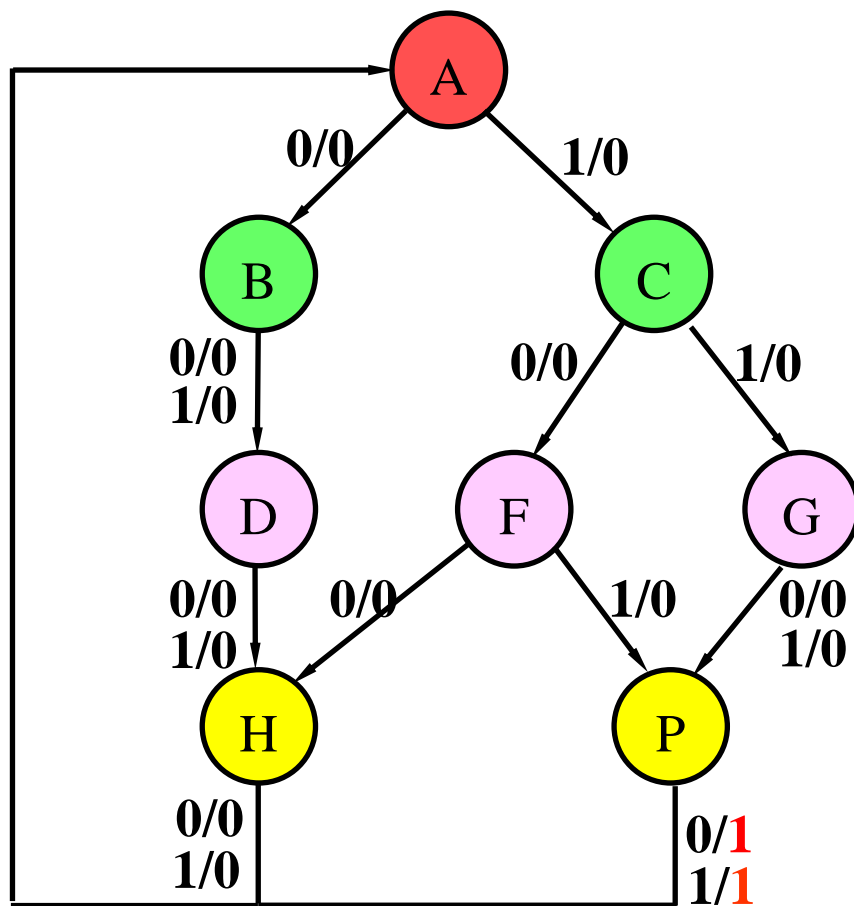
原始状态图和状态表

□ Moore型代码检测器



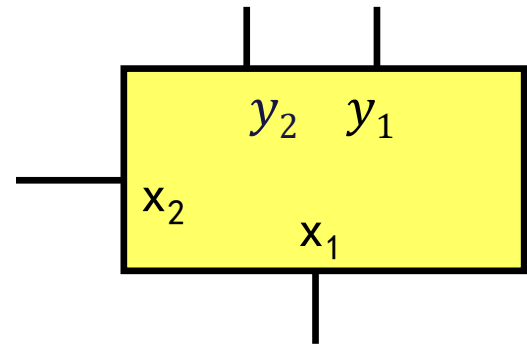
原始状态图和状态表

□思考：如果先输入高位，再输入低位如何做Mealy状态图？



原始状态图和状态表

- 设计一个两位串行输入，并行输出双向移位寄存器
 - 两位串行输入 x_1x_2 ,
 - x_1 为数据输入端， x_2 为控制端
 - $x_2 = 0$ ， x_1 往寄存器高位串行送数，寄存器的数据从高位移向低位
 - $x_2 = 1$ ， x_1 往寄存器低位串行送数，寄存器的数据从低位移向高位
 - 寄存器的输出为触发器状态本身



原始状态图和状态表

□分析

- 设并行输出端为 y_2y_1 , y_2 为高位, y_1 为低位
- $x_2 = 0$, x_1 往寄存器高位串行送数, 寄存器的数据从高位移向低位
 - $x_2 = 0 \quad y_2^{n+1}y_1^{n+1} = x_1y_2$
- $x_2 = 1$, x_1 往寄存器低位串行送数, 寄存器的数据从低位移向高位
 - $x_2 = 1 \quad y_2^{n+1}y_1^{n+1} = y_1x_1$

原始状态图和状态表

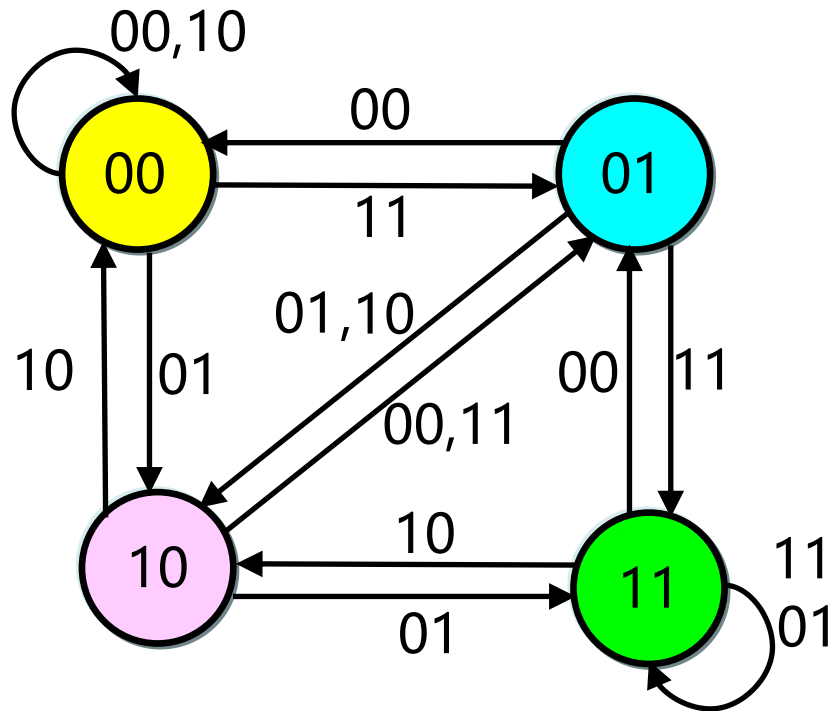
$$\square x_2 = 0 \quad y_2^{n+1} y_1^{n+1} = x_1 y_2$$

$$\square x_2 = 1 \quad y_2^{n+1} y_1^{n+1} = y_1 x_1$$

现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$			
	$x_2 x_1 = 00$	$x_2 x_1 = 01$	$x_2 x_1 = 10$	$x_2 x_1 = 11$
00	00	10	00	01
01	00	10	10	11
11	01	11	10	11
10	01	11	00	01

原始状态图和状态表

现态 y_2y_1	次态 $y_2^{n+1}y_1^{n+1}$			
	x_2x_1 00	x_2x_1 01	x_2x_1 10	x_2x_1 11
00	00	10	00	01
01	00	10	10	11
11	01	11	10	11
10	01	11	00	01



同步时序逻辑电路设计

1 原始状态图和状态表

2 状态化简

3 状态编码

4 激励函数和输出函数

5 逻辑电路图

同步时序逻辑电路设计

□ 状态化简

- 从原始状态表中消去多余状态
- 得到一个描述给定的逻辑功能的包含状态数目达到最少的状态表
- 最简状态表或最小化状态表
- 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

同步时序逻辑电路设计

□ 状态化简

– 状态数和所需触发器的关系

- 状态数: n ,
- 所需触发器: m
- $2^m \geq n > 2^{m-1}$

– 方法

- 观察法
- 输出分类法
- 隐含表法

状态化简

□ 等效状态

- 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

状态化简

□ 等效状态的判断方法

- 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 等效的条件为在一位输入的各种取值组合下满足两条:
 - 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

状态化简

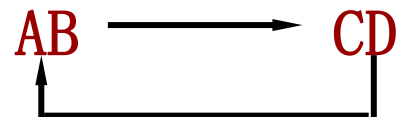
□ 次态交错

- 在右表中当 $X=0$ 时，现态A、B的次态相同，现态C、D的次态交错；

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

□ 次态循环

- 当 $X=1$ 时，现态A、B的次态为C、D,而现态C、D的次态为A、B，构成次态循环



状态化简

□ 等效状态的性质

– 等效状态具有传递性

- 假若 S_1 和 S_2 等效, S_2 和 S_3 等效, 则 S_1 和 S_3 等效

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

状态化简

□ 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
- $(S_1, S_2), (S_2, S_3) \longrightarrow \{S_1, S_2, S_3\}$
- 广义的概念，两个及以上状态可以组成一个等效类
 - 一个状态也可以称为等效类
 - 任何状态和它自身必然是等效的

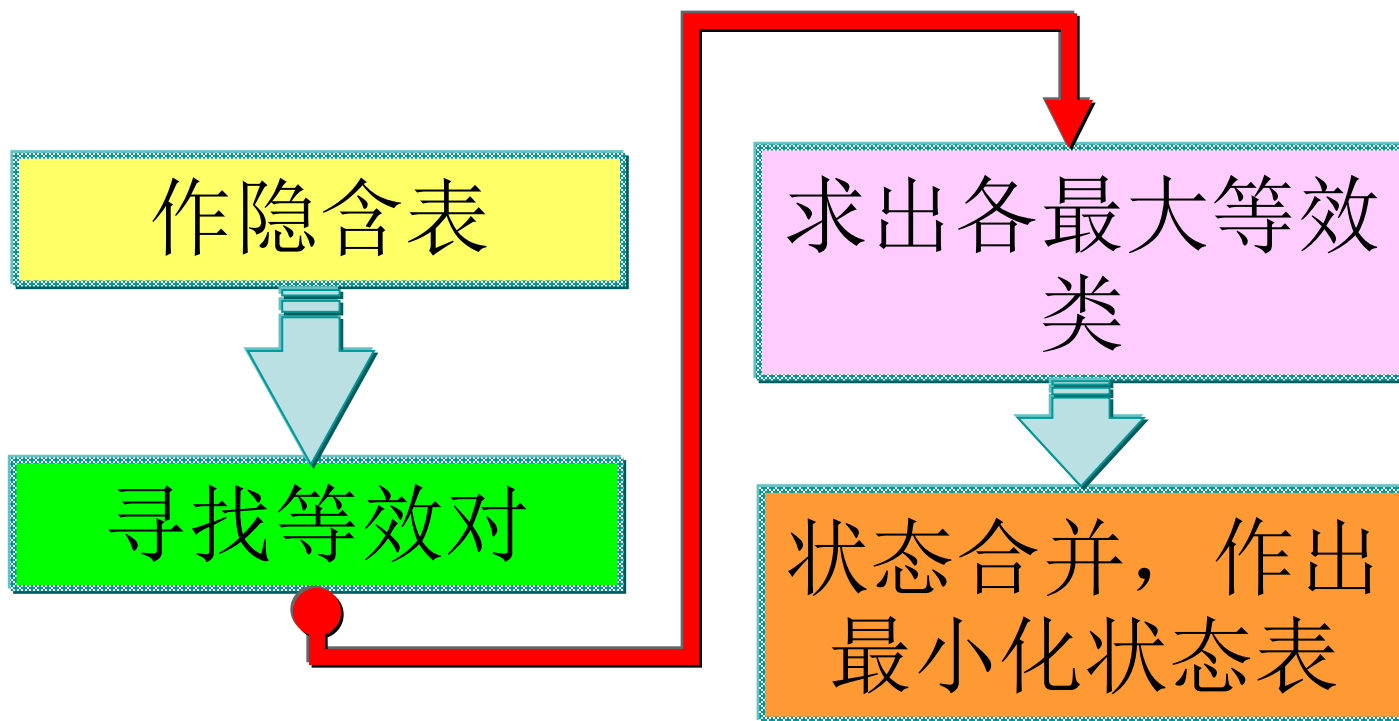
状态化简

□ 最大等效类

- 不被任何别的等效类所包含的等效类
- 注意
 - 最大，并不是指包含的状态最多，而是指它的独立性
 - 即使是一个状态，只要它不被包含在别的等效类中，也是最大等效类

状态化简

□化简过程



状态化简

- 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！

状态化简

□ 隐含表

– 一个直角三角形阶梯网格

- 横向和纵向的网格数等于原始状态表中的状态数 $n-1$
- 横向从左到右依次标上原始状态表中的前 $n-1$ 个状态
- 纵向自上到下依次标上原始状态表中的后 $n-1$ 个状态
- 每个方格代表一个状态对

状态化简

□ 寻找等效对

– 顺序比较

- 对原始状态表所有“状态对”进行逐一检查和比较内
 - 等效: “√” ;
 - 不等效: “×”
 - 与其他状态对相关: 填上相关的状态对

– 关联比较

- 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

状态化简

□ 求最大等效类

- 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
- 注意
 - 各最大等效类之间不应出现相同状态
 - 原始状态表中的每一个状态都必须属于某一个最大等效类，否则，化简后的状态表不能描述原始状态表所描述的功能。

□ 状态合并

- 将每个最大等效类中的全部状态合并为一个状态

状态化简

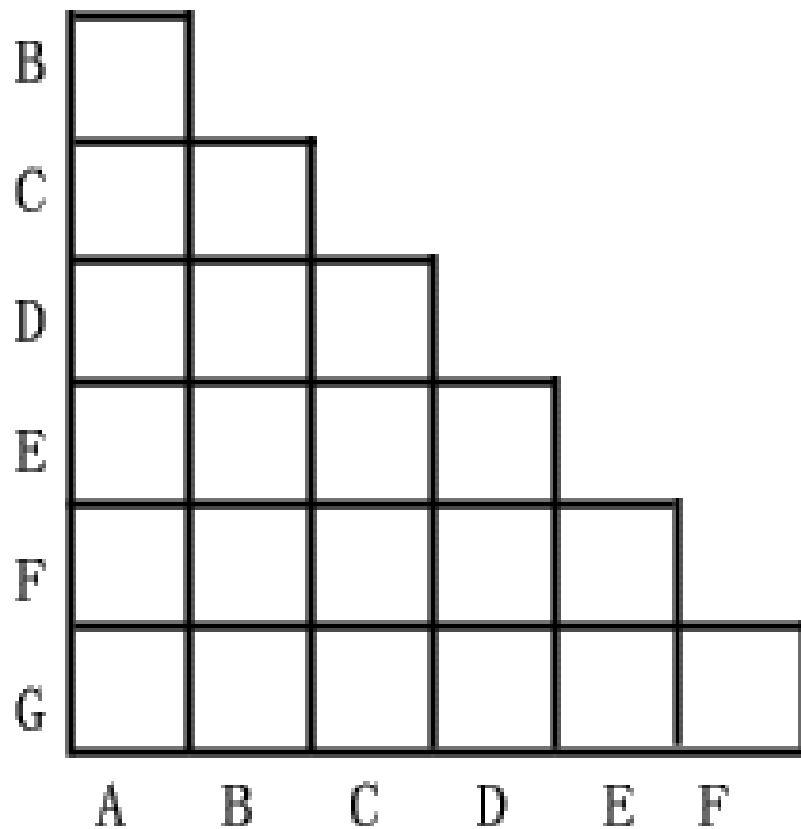
□例 化简下表所示原始状态表。

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

状态化简

□ ① 作隐含表

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



状态化简

□② 寻找等效对---顺序比较

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

B	CF					
C	×	×				
D	×	×	×			
E	BE	$\frac{AE}{CF}$	×	×		
F	×	×	✓	×	×	
G	×	×	×	$\frac{CD}{DE}$	×	×
	A	B	C	D	E	F

状态化简

□② 寻找等效对---关联比较

- AB→CF

- (A, B)

- AE→BE \Rightarrow $\begin{cases} AE \\ CF \end{cases}$ ✓

- (A, E)

- (B, E)

- DG \Rightarrow $\begin{cases} CD \times \\ DE \times \end{cases}$

B	\checkmark					
C	\times	\times				
D	\times	\times	\times			
E	\checkmark	\checkmark	\times	\times		
F	\times	\times	\checkmark	\times	\times	
G	\times	\times	\times	\times	\times	\times
	A	B	C	D	E	F






状态化简

□ ② 寻找等效对---等效对

- (A, B)
- (A, E)
- (B, E)
- (C, F)

□ ③ 最大等效类

- {A, B, E}
- {C, F}
- {D} {G}

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

状态化简

□④ 状态合并

- {A, B, E} → a
- {C, F} → b
- {D} → c
- {G} → d

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

状态化简

□④ 状态合并

- {A, B, E} → a
- {C, F} → b
- {D} → c
- {G} → d

现态	次态/输出	
	X=0	X=1
a	C/0	a/1
a	F/0	a/1
C	F/0	G/0
D	D/1	a/0
a	C/0	a/1
F	C/0	G/0
G	C/1	D/0

状态化简

□④ 状态合并

- {A, B, E} → a
- {C, F} → b
- {D} → c
- {G} → d

现态	次态/输出	
	X=0	X=1
a	b/0	a/1
a	b/0	a/1
b	b/0	G/0
D	D/1	a/0
a	b/0	a/1
b	b/0	G/0
G	b/1	D/0

状态化简

□④ 状态合并

- {A, B, E} → a
- {C, F} → b
- {D} → c
- {G} → d

现态	次态/输出	
	X=0	X=1
a	b/0	a/1
a	b/0	a/1
b	b/0	G/0
c	c/1	a/0
a	b/0	a/1
b	b/0	G/0
G	b/1	c/0

状态化简

□④ 状态合并

- {A, B, E} → a
- {C, F} → b
- {D} → c
- {G} → d

现态	次态/输出	
	X=0	X=1
a	b/0	a/1
a	b/0	a/1
b	b/0	d/0
c	c/1	a/0
a	b/0	a/1
b	b/0	d/0
d	b/1	c/0

状态化简

□④ 状态合并

- {A, B, E} → a
- {C, F} → b
- {D} → c
- {G} → d

现态	次态/输出	
	X=0	X=1
a	b/0	a/1
a	b/0	a/1
b	b/0	d/0
c	c/1	a/0
a	b/0	a/1
b	b/0	d/0
d	b/1	c/0

状态化简

□ ④ 状态合并

- $\{A, B, E\} \rightarrow a$
- $\{C, F\} \rightarrow b$
- $\{D\} \rightarrow c$
- $\{G\} \rightarrow d$

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

同步时序逻辑电路设计

1 原始状态图和状态表

2 状态化简

3 状态编码

4 激励函数和输出函数

5 逻辑电路图

同步时序逻辑电路设计

□ 状态编码

- 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
- 也称状态分配，或者状态赋值
- 任务
 - 确定状态编码的长度(所需触发器个数)
 - 寻找一种最佳的或接近最佳的状态分配方案
 - 使所设计的时序电路最简单

状态编码

□ 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
- $2^m \geq N > 2^{m-1}$

□ 确定状态分配方案

- 状态与代码之间的对应关系可以有多种
- 分配方案数 K_s

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

状态编码

- 随着状态数目的增加，分配方案的数目急剧增加
- 如何从众多的分配方案中寻找出一种最佳方案？
- 在实际工作中，工程技术人员通常按照一定的原则、凭借设计的经验去寻找相对最佳的编码方案
- 一种常用方法称为相邻分配法

状态编码

□ 相邻分配法的基本思想

- 在选择状态编码时，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，从而有利于激励函数和输出函数的化简

状态编码

□ 相邻分配法的状态编码原则

- 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 同一现态，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。)

状态编码

□ 注意

- 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
- 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

状态编码

□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

状态编码

□分析

- $N = 4$
- $m = 2$
- 需要两个触发器

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

状态编码

□分析

- 原则①
 - BC应相邻
- 原则②
 - BC、AD、CD应相邻
- 原则③
 - AD应相邻
- 总结
 - BC、AD、CD相邻

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

状态编码

□ 状态分配的工具

- 卡诺图

□ 假定状态变量用 y_2y_1 表示

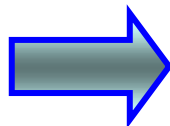
- 初始状态A
- $A=00$
- $B=01$
- $C=11$
- $D=10$

		y_2	
		0	1
y_1	0	A	D
	1	B	C

状态编码

□ A=00 B=01 C=11 D=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



现态	次态/输出	
	x=0	x=1
00	11/1	01/0
01	00/0	00/1
11	00/1	10/1
10	10/1	11/0

状态编码

□ 确定触发器数目和类型

- 根据二进制状态表中二进制代码的位数确定电路中所需触发器数目
- 选择合适的触发器类型

□ 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
- 化简
- 工具：激励表

同步时序逻辑电路设计

□ 触发器的激励表

- 激励表反应了触发器从现态转移到某种次态时，对输入条件的要求
- 把触发器的现态和次态作为自变量，而把触发器的输入(或激励)作为因变量

同步时序逻辑电路设计

□ 四种时钟控制触发器的激励表

钟控R-S触发器激励表		
现态Q	次态 Q^{n+1}	RS
0	0	d 0
0	1	0 1
1	0	1 0
1	1	0 d

钟控D触发器激励表		
现态Q	次态 Q^{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

钟控J-K触发器激励表		
现态Q	次态 Q^{n+1}	J K
0	0	0 d
0	1	1 d
1	0	d 1
1	1	d 0

钟控T触发器激励表		
现态Q	次态 Q^{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

同步时序逻辑电路设计

□ 步骤

- 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

同步时序逻辑电路设计

□ 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 $y_2 y_1$	次态 $y_2^{n+1}y_1^{n+1}$ /输出Z	
	x=0	x=1
00	11/0	01/0
01	00/0	00/1
10	01/0	11/0
11	00/1	10/1

同步时序逻辑电路设计

激励函数和输出函数真值表

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
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

现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$ /输出 Z	
	$x=0$	$x=1$
00	11/0	01/0
01	00/0	00/1
10	01/0	11/0
11	00/1	10/1

钟控J-K触发器激励表		
现态 Q	次态 Q^{n+1}	J K
0	0	0 d
0	1	1 d
1	0	d 1
1	1	d 0

同步时序逻辑电路设计

激励函数和输出函数

输入 x	现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	激励函数 $J_2 K_2 \quad J_1 K_1$	输出 Z
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_2			
		00	01	11	10
x	0	1	0	d	d
	1	0	0	d	d

$$J_2 = \bar{x} \bar{y}_1$$

同步时序逻辑电路设计

激励函数和输出函数

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

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

$$K_2 = \bar{x}$$

同步时序逻辑电路设计

激励函数和输出函数

输入 x	现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	激励函数 $J_2 K_2 \quad J_1 K_1$	输出 Z
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			
x	$y_2 y_1$	00	01	11	10
	0	1	d	d	1
1	1	1	d	d	1

$$J_1 = 1$$

同步时序逻辑电路设计

激励函数和输出函数

输入 x	现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	激励函数 $J_2 K_2 \quad J_1 K_1$	输出 Z
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		K_1			
		00	01	11	10
0		d	1	1	d
1		d	1	1	d

$$K_1 = 1$$

同步时序逻辑电路设计

激励函数和输出函数

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

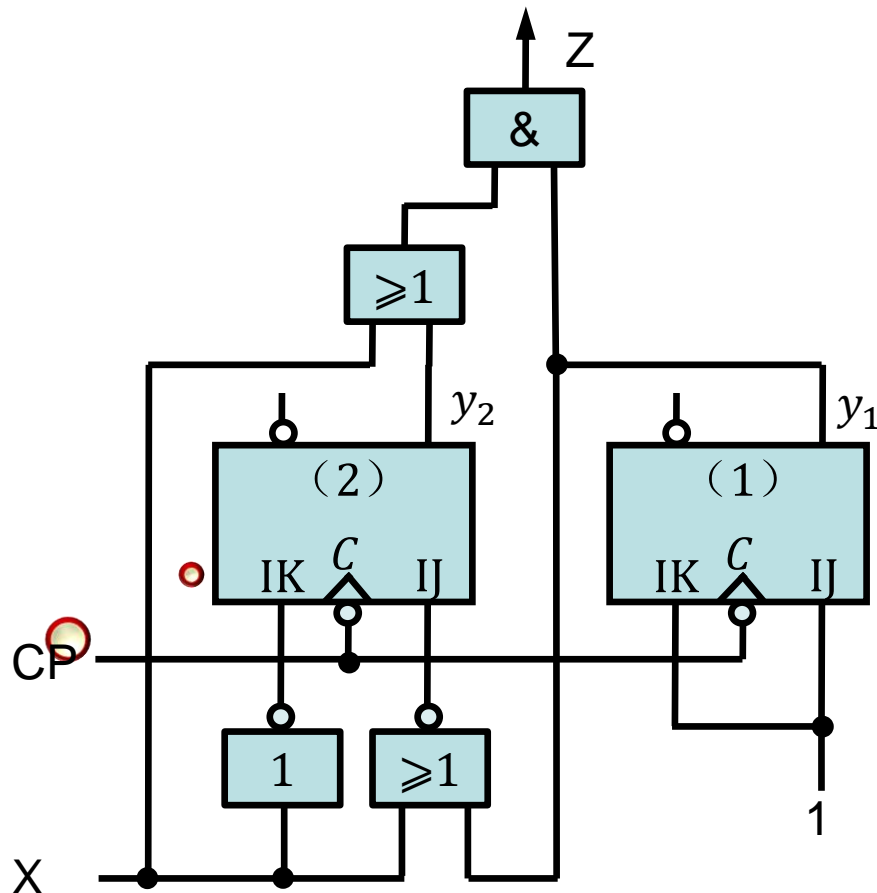
$$Z = y_2 y_1 + x y_1$$

同步时序逻辑电路设计

□ 逻辑电路图

- $J_2 = \bar{x} \overline{y_1}$
- $K_2 = \bar{x}$
- $J_1 = 1$
- $K_1 = 1$
- $Z = y_2 y_1 + x y_1$

若选D触发器作为
存储元件是否更简
呢？



同步时序逻辑电路设计

□若选用D触发器作为存储元件实现

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

		D_2			
x	$y_2 y_1$	00	01	11	10
	0	1	0	0	0
1	1	0	0	1	1

$$\begin{aligned}
 D_2 &= \bar{x} \bar{y}_2 \bar{y}_1 + x y_2 \\
 &= \overline{x + y_2 + y_1} + x y_2
 \end{aligned}$$

同步时序逻辑电路设计

□若选用D触发器作为存储元件实现

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

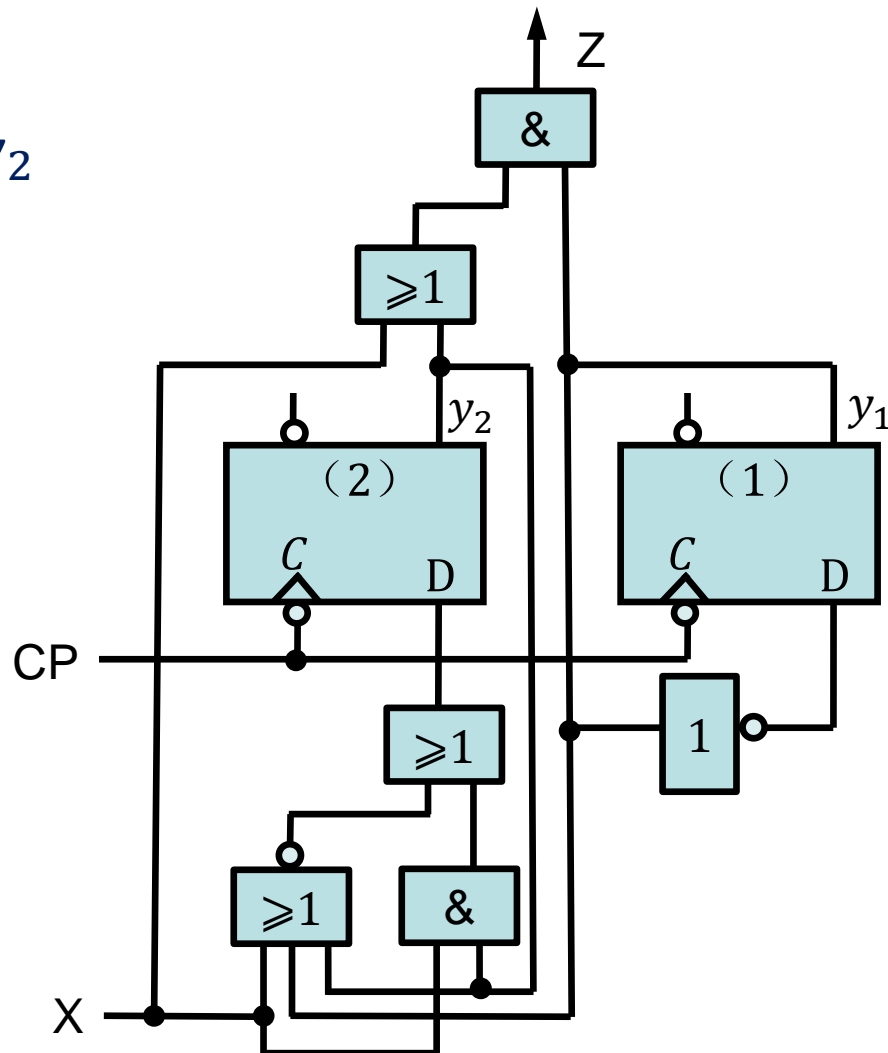
		D_1			
x	$y_2 y_1$	00	01	11	10
		1	0	0	1
		1	0	0	1

$$D_1 = \overline{y_1}$$

同步时序逻辑电路设计

□ 逻辑电路图

- $D_2 = \overline{x + y_2 + y_1 + xy_2}$
- $D_1 = \overline{y_1}$
- $Z = y_2y_1 + xy_1$



同步时序逻辑电路设计

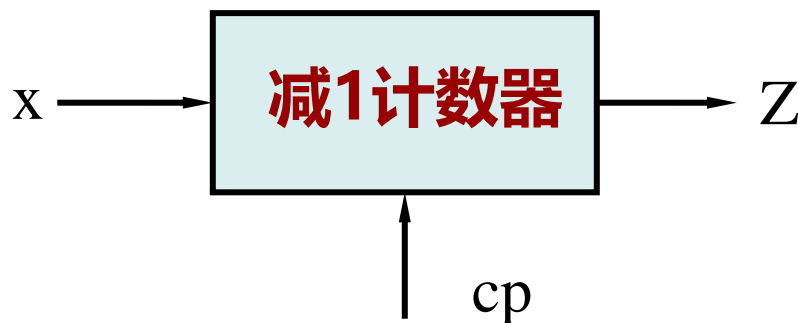
□例 用T触发器作为存储元件，设计一个2位二进制减1计数器

- 电路工作状态受输入信号 x 的控制
- 当 $x=0$ 时，电路状态不变
- 当 $x=1$ 时，在时钟脉冲作用下进行减1计数
- 计数器有一个输出 Z
- 当产生借位时 Z 为1
- 其他情况下 Z 为0

同步时序逻辑电路设计

□分析：

- 该电路的逻辑框图如下：

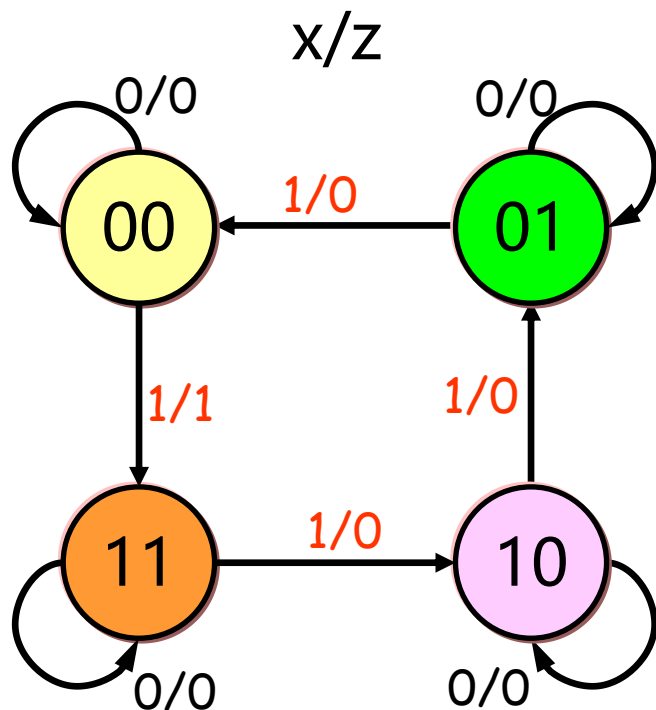


- 该问题对电路所要求的状态数目及状态转换关系均十分清楚，故可直接作出计数器的二进制状态图和二进制状态表

同步时序逻辑电路设计

□ ①作出状态图和状态表

– 设状态变量用 y_2 、 y_1 表示



现态 $y_2 y_1$	次态 $y_2^{n+1}y_1^{n+1}$ /输出	
	$x=0$	$x=1$
00	00/0	11/1
01	01/0	00/0
10	10/0	01/0
11	11/0	10/0

同步时序逻辑电路设计

□ 激励函数和输出函数真值表

输入 x	现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	激励函数 $T_2 T_1$		输出 Z
0	0 0	0 0	0	0	0
0	0 1	0 1	0	0	0
0	1 0	1 0	0	0	0
0	1 1	1 1	0	0	0
1	0 0	1 1	1	1	1
1	0 1	0 0	0	1	0
1	1 0	0 1	1	1	0
1	1 1	1 0	0	1	0

现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$ / 输出	
	x=0	x=1
00	00/0	11/1
01	01/0	00/0
10	10/0	01/0
11	11/0	10/0

钟控T触发器激励表		
现态Q	次态 Q^{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

同步时序逻辑电路设计

激励函数和输出函数

输入 x	现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	激励函数 $T_2 \quad T_1$		输出 Z
0	0 0	0 0	0	0	0
0	0 1	0 1	0	0	0
0	1 0	1 0	0	0	0
0	1 1	1 1	0	0	0
1	0 0	1 1	1	1	1
1	0 1	0 0	0	1	0
1	1 0	0 1	1	1	0
1	1 1	1 0	0	1	0

		T_2			
x	$y_2 y_1$	00	01	11	10
	0				
1		1			1

$$T_2 = x\overline{y_1}$$

同步时序逻辑电路设计

激励函数和输出函数

输入 x	现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	激励函数 $T_2 \quad T_1$		输出 Z
0	0 0	0 0	0	0	0
0	0 1	0 1	0	0	0
0	1 0	1 0	0	0	0
0	1 1	1 1	0	0	0
1	0 0	1 1	1	1	1
1	0 1	0 0	0	1	0
1	1 0	0 1	1	1	0
1	1 1	1 0	0	1	0

		T_1			
x	$y_2 y_1$	00	01	11	10
	0				
1		1	1	1	1

$$T_1 = x$$

同步时序逻辑电路设计

□ 激励函数和输出函数

输入 x	现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	激励函数 $T_2 \quad T_1$		输出 Z
0	0 0	0 0	0	0	0
0	0 1	0 1	0	0	0
0	1 0	1 0	0	0	0
0	1 1	1 1	0	0	0
1	0 0	1 1	1	1	1
1	0 1	0 0	0	1	0
1	1 0	0 1	1	1	0
1	1 1	1 0	0	1	0

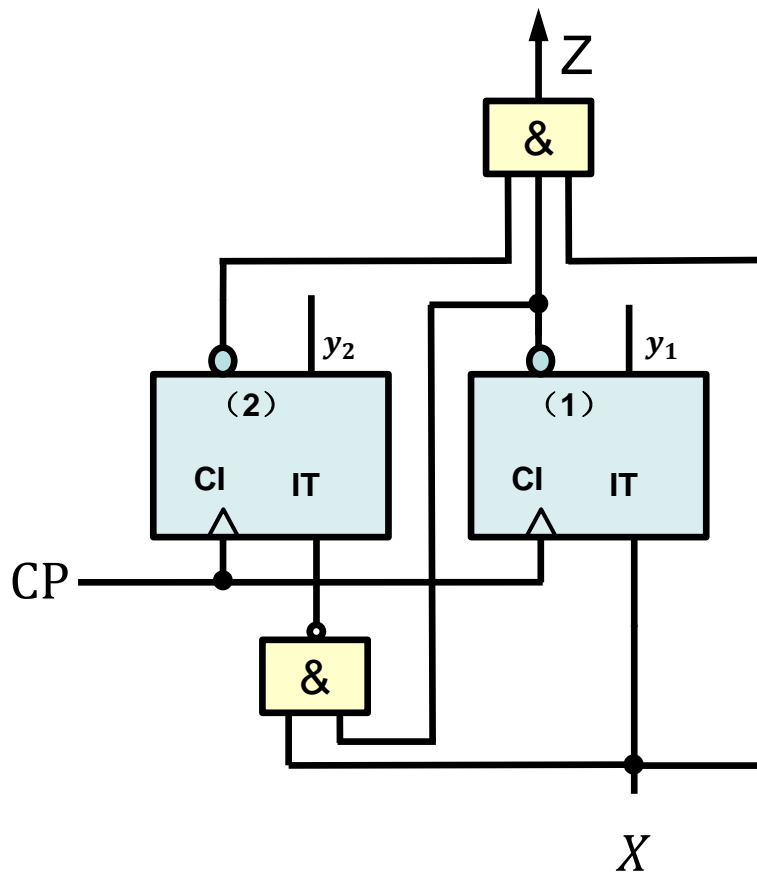
		Z			
x	$y_2 y_1$	00	01	11	10
0					
1		1			

$$Z = x \overline{y_2} \overline{y_1}$$

同步时序逻辑电路设计

□ 逻辑电路图

- $T_2 = x\overline{y_1}$
- $T_1 = x$
- $Z = x\overline{y_2} \overline{y_1}$



同步时序逻辑电路设计

□例 设计一个 “101” 序列检测器

- 采用 J-K触发器作为存储元件
- 一个输入x
- 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

同步时序逻辑电路设计

□分析：

- 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

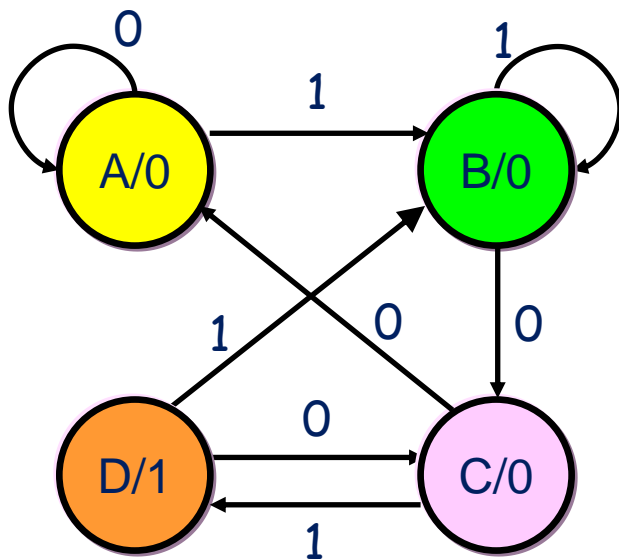
输出Z: 0 0 0 0 1 0 1 0 0 1 0 0

- 可重叠101序列检测器

同步时序逻辑电路设计

□ 作出原始状态图和状态表

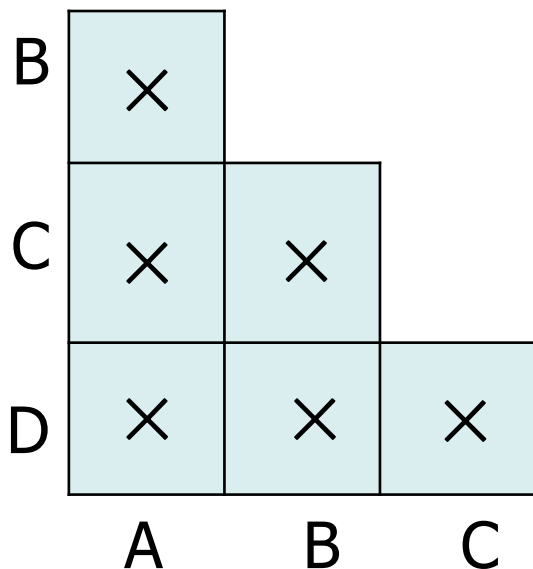
– 假定：Moore型



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

同步时序逻辑电路设计

□ 状态化简



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

— 所得原始状态表已是最小化状态表

同步时序逻辑电路设计

□ 状态编码

- 4个状态
- 2位二进制代码
- 电路中要有2个触发器。
- 设状态变量为 y_2 、 y_1
- $A=00$, $B=01$
- $C=10$, $D=11$

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

同步时序逻辑电路设计

□ 状态编码

- 4个状态
- 2位二进制代码
- 电路中要有2个触发器。
- 设状态变量为 y_2 、 y_1
- $A=00$, $B=01$
- $C=10$, $D=11$

现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$		输出
	$x=0$	$x=1$	Z
00	00	01	0
01	10	01	0
10	00	11	0
11	10	01	1

同步时序逻辑电路设计

□ 输出函数和激励函数真值表

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
0	0 0	0 0	0 d 0 d	0
0	0 1	1 0	1 d d 1	0
0	1 0	0 0	d 1 0 d	0
0	1 1	1 0	d 0 d 1	1
1	0 0	0 1	0 d 1 d	0
1	0 1	0 1	0 d d 0	0
1	1 0	1 1	d 0 1 d	0
1	1 1	0 1	d 1 d 0	1

现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$		输出 Z
	$x=0$	$x=1$	
00	00	01	0
01	10	01	0
10	00	11	0
11	10	01	1

钟控J-K触发器激励表		
现态Q	次态 Q^{n+1}	J K
0	0	0 d
0	1	1 d
1	0	d 1
1	1	d 0

同步时序逻辑电路设计

□ 输出函数和激励函数真值表

输入 x	现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	激励函数 $J_2 K_2 \quad J_1 K_1$	输出 Z
0	0 0	0 0	0 d 0 d	0
0	0 1	1 0	1 d d 1	0
0	1 0	0 0	d 1 0 d	0
0	1 1	1 0	d 0 d 1	1
1	0 0	0 1	0 d 1 d	0
1	0 1	0 1	0 d d 0	0
1	1 0	1 1	d 0 1 d	0
1	1 1	0 1	d 1 d 0	1

		J_2			
		y_2y_1	00	01	11
x	0	0	1	d	d
	1	0	0	d	d

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

同步时序逻辑电路设计

□ 输出函数和激励函数真值表

输入 x	现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	激励函数 $J_2 K_2 \quad J_1 K_1$	输出 Z
0	0 0	0 0	0 d 0 d	0
0	0 1	1 0	1 d d 1	0
0	1 0	0 0	d 1 0 d	0
0	1 1	1 0	d 0 d 1	1
1	0 0	0 1	0 d 1 d	0
1	0 1	0 1	0 d d 0	0
1	1 0	1 1	d 0 1 d	0
1	1 1	0 1	d 1 d 0	1

		K_2				
		y_2y_1	00	01	11	10
x	0	d	d	0	1	
	1	d	d	1	0	

$$\begin{aligned} K_2 &= xy_1 + \bar{x} \bar{y}_1 \\ &= \bar{x} \oplus y_1 \end{aligned}$$

同步时序逻辑电路设计

□ 输出函数和激励函数真值表

输入 x	现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	激励函数 $J_2 K_2 \quad J_1 K_1$	输出 Z
0	0 0	0 0	0 d 0 d	0
0	0 1	1 0	1 d d 1	0
0	1 0	0 0	d 1 0 d	0
0	1 1	1 0	d 0 d 1	1
1	0 0	0 1	0 d 1 d	0
1	0 1	0 1	0 d d 0	0
1	1 0	1 1	d 0 1 d	0
1	1 1	0 1	d 1 d 0	1

$y_2 y_1$ x		J_1			
		00	01	11	10
0		0	d	d	0
1		1	d	d	1

$$J_1 = x$$

同步时序逻辑电路设计

□ 输出函数和激励函数真值表

输入 x	现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	激励函数 $J_2 K_2 \quad J_1 K_1$	输出 Z
0	0 0	0 0	0 d 0 d	0
0	0 1	1 0	1 d d 1	0
0	1 0	0 0	d 1 0 d	0
0	1 1	1 0	d 0 d 1	1
1	0 0	0 1	0 d 1 d	0
1	0 1	0 1	0 d d 0	0
1	1 0	1 1	d 0 1 d	0
1	1 1	0 1	d 1 d 0	1

$y_2 y_1$ x		K_1			
		00	01	11	10
0		d	1	1	d
1		d	0	0	d

$$K_1 = \bar{x}$$

同步时序逻辑电路设计

□ 输出函数和激励函数真值表

输入 x	现态 $y_2 y_1$	次态 $y_2^{n+1} y_1^{n+1}$	激励函数 $J_2 K_2 \quad J_1 K_1$	输出 Z
0	0 0	0 0	0 d 0 d	0
0	0 1	1 0	1 d d 1	0
0	1 0	0 0	d 1 0 d	0
0	1 1	1 0	d 0 d 1	1
1	0 0	0 1	0 d 1 d	0
1	0 1	0 1	0 d d 0	0
1	1 0	1 1	d 0 1 d	0
1	1 1	0 1	d 1 d 0	1

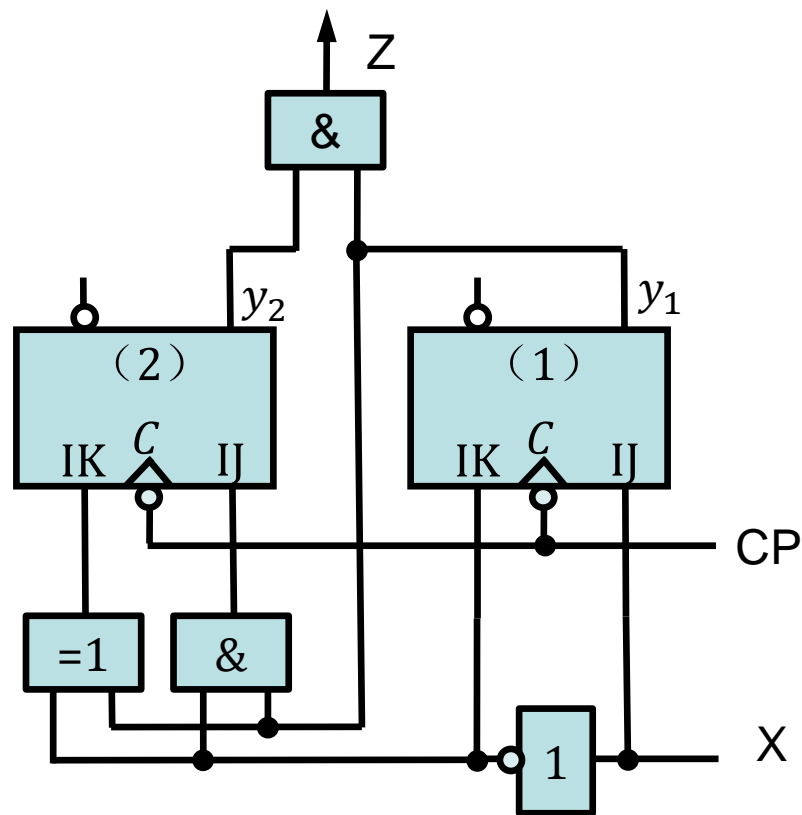
		Z			
x	$y_2 y_1$	00	01	11	10
0		0	0	1	0
1		0	0	1	0

$$Z = y_2 y_1$$

同步时序逻辑电路设计

□ 逻辑电路图

- $J_2 = \bar{x}y_1$
- $K_2 = \bar{x} \oplus y_1$
- $J_1 = x$
- $K_1 = \bar{x}$
- $Z = y_2y_1$



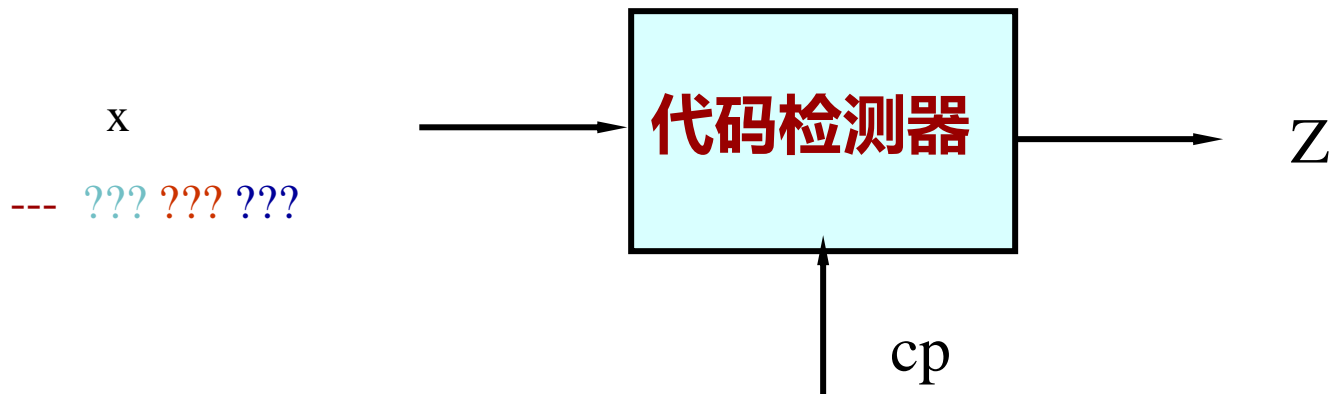
同步时序逻辑电路设计

□例3 设计一个3位二进制码的串行奇偶检测器。

- 该电路从输入端x串行输入二进制代码
- 每三位为一组
- 当三位代码中含1的个数为偶数时，Z输出1
- 平时Z输出为0

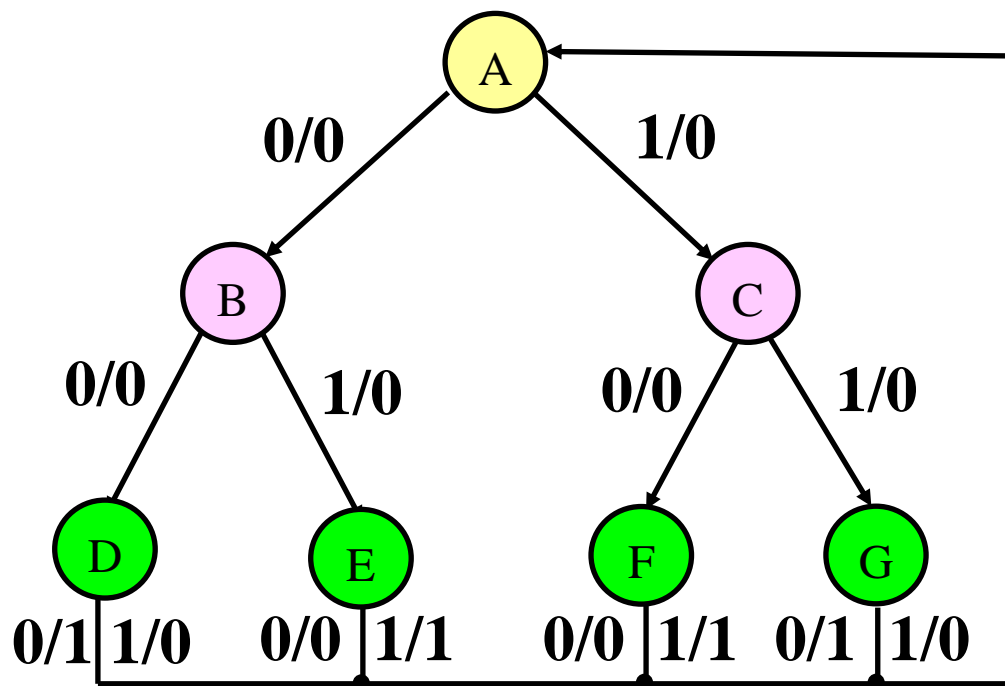
同步时序逻辑电路设计

- 解 代码检测器的特点是输入信号是按位分组的，每组的检测过程相同，即一组检测完后，电路回到初始状态，接着进行下一组的检测。



同步时序逻辑电路设计

□ ① 建立原始状态图和原始状态表



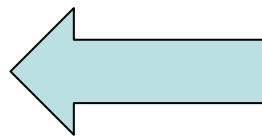
现态	次态/输出	
	x=0	x=1
A	B/0	C/0
B	D/0	E/0
C	F/0	G/0
D	A/1	A/0
E	A/0	A/1
F	A/0	A/1
G	A/1	A/0

同步时序逻辑电路设计

□② 状态化简

- 观察法, D和G等效, E和F等效
- (D,G) --- D (E,F) --- 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



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

同步时序逻辑电路设计

□ ③ 状态编码

- 5个状态，需用3位二进制码表示
- 设状态变量用 y_3 、 y_2 、 y_1 表示
- 原则①
 - DE应相邻
- 原则②
 - BC、DE应相邻
- 原则③
 - AB、AC、BC应相邻

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_2			
		00	01	11	10
y_1	0	A	B	C	D
	1				E

同步时序逻辑电路设计

$y_3y_2y_1$	$y_3^{n+1}y_2^{n+1}y_1^{n+1}/Z$	
	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

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_2			
		00	01	11	10
y_1	0	A	B	C	D
	1				E

同步时序逻辑电路设计

□ ④确定激励函数和输出函数

– 假定用D触发器作为存储元件

$y_3y_2y_1$	$y_3^{n+1}y_2^{n+1}y_1^{n+1}/Z$	
	$x=0$	$x=1$
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

D_3

xy_3 y_2y_1		00	01	11	10
		00	01	11	10
00		0	0	0	1
01		d	0	0	d
11		d	d	d	d
10		1	1	1	1

$$D_3 = y_2 + x\overline{y_3}$$

同步时序逻辑电路设计

□ ④确定激励函数和输出函数

– 假定用D触发器作为存储元件

$y_3 y_2 y_1$	$y_3^{n+1} y_2^{n+1} y_1^{n+1} / Z$	
	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

D_2

xy_3	00	01	11	10
$y_2 y_1$	1	0	0	1
00	d	0	0	d
01	d	d	d	d
11	0	0	0	0
10				

$$D_2 = \overline{y_3} \overline{y_2} = \overline{y_3 + y_2}$$

同步时序逻辑电路设计

□ ④确定激励函数和输出函数

– 假定用D触发器作为存储元件

$y_3y_2y_1$	$y_3^{n+1}y_2^{n+1}y_1^{n+1}/Z$	
	$x=0$	$x=1$
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

		D_1			
y_2y_1	xy_3	00	01	11	10
	00	0	0	0	0
01	d	0	0	d	d
11	d	d	d	d	d
10	0	1	0	1	1

$$\begin{aligned}D_1 &= \bar{x}y_3y_2 + x\bar{y}_3y_2 \\ &= (x \oplus y_3)y_2\end{aligned}$$

同步时序逻辑电路设计

□ ④确定激励函数和输出函数

– 假定用D触发器作为存储元件

$y_3y_2y_1$	$y_3^{n+1}y_2^{n+1}y_1^{n+1}/Z$	
	$x=0$	$x=1$
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

		Z			
xy_3	y_2y_1	00	01	11	10
		0	1	0	0
	00	0	1	0	0
	01	d	0	1	d
	11	d	d	d	d
	10	0	0	0	0

$$Z = \bar{x}y_3\bar{y}_2y_1 + xy_1$$

同步时序逻辑电路设计

□ ⑤ 讨论

- 当电路中包含多个无效状态时，往往又将无效状态构成的集合称作状态的无效序列
- 相应地将正常工作下的状态集合称为状态的有效序列
- 要讨论在无效状态时，电路的输出和状态转移

同步时序逻辑电路设计

□ ⑤ 讨论

– 1. 电路是否具有自恢复功能

- 电路万一偶然进入无效状态，如果能在输入信号和时钟脉冲作用下自动进入有效状态，则称为具有自恢复功能，否则，称为“挂起”
- 存在挂起，必须修正

– 2. 电路是否会产生错误输出信号

- 电路万一处在无效状态，是否会在输入信号和时钟脉冲作用下产生错误输出信号
- 存在错误输出信号，必须修正

同步时序逻辑电路设计

□ ⑤ 讨论

- 讨论时，可根据确定的激励函数和输出函数表达式作出相应状态表和状态图，并作出结论。
- 当设计方案需要修改时，只需要考虑激励函数和输出函数化简时对无效状态下任意项的处理问题

同步时序逻辑电路设计

□讨论

- 本例的状态无效序列中包含001、011和111
- 它们在化简激励函数和输出函数时被作为无关条件处理
- 实际上，在卡诺图上和1圈在一起即指定为1，否则为0
- 讨论时只需检查卡诺图，便可知道无效状态下的激励函数和输出函数的取值
- 并推出相应次态和输出
- 进而作出与设计方案对应的状态图或状态表
- 得出讨论结果

同步时序逻辑电路设计

		D_3			
xy_3	y_2y_1	00	01	11	10
00	00	0	0	0	1
01	01	d	0	0	d
11	11	d	d	d	d
10	10	1	1	1	1

		D_2			
xy_3	y_2y_1	00	01	11	10
00	00	1	0	0	1
01	01	d	0	0	d
11	11	d	d	d	d
10	10	0	0	0	0

		D_1			
xy_3	y_2y_1	00	01	11	10
00	00	0	0	0	0
01	01	d	0	0	d
11	11	d	d	d	d
10	10	0	1	0	1

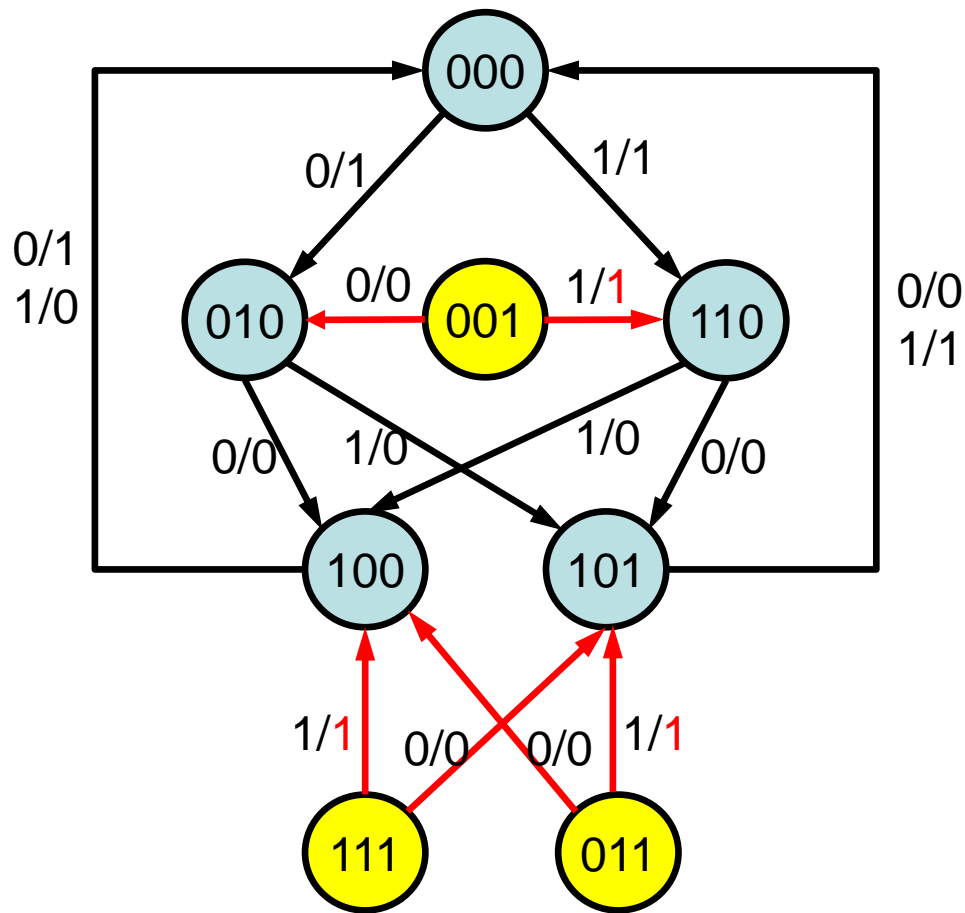
		Z			
xy_3	y_2y_1	00	01	11	10
00	00	0	1	0	0
01	01	d	0	1	d
11	11	d	d	d	d
10	10	0	0	0	0

输入 x	现态 $y_3 y_2 y_1$	激励函数 $D_3 D_2 D_1$	次态 $y_3^{n+1} y_2^{n+1} y_1^{n+1}$	输出 Z
0	0 0 1	0 1 0	0 1 0	0
0	0 1 1	1 0 0	1 0 0	0
0	1 1 1	1 0 1	1 0 1	0
1	0 0 1	1 1 0	1 1 0	1
1	0 1 1	1 0 1	1 0 1	1
1	1 1 1	1 0 0	1 0 0	1

同步时序逻辑电路设计

$y_3y_2y_1$	$y_3^{n+1}y_2^{n+1}y_1^{n+1}/Z$	
	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

输入 x	现态 $y_3y_2y_1$	激励函数 $D_3D_2D_1$	次态 $y_3^{n+1}y_2^{n+1}y_1^{n+1}$	输出 Z
0	001	010	010	0
0	011	100	100	0
0	111	101	101	0
1	001	110	110	1
1	011	101	101	1
1	111	100	100	1



无挂起，有错误输出，需修正。

同步时序逻辑电路设计

□ 输出函数修正

		Z			
		00	01	11	10
y_2y_1	00	0	1	0	0
	01	d	0	1	d
	11	d	d	d	d
	10	0	0	0	0

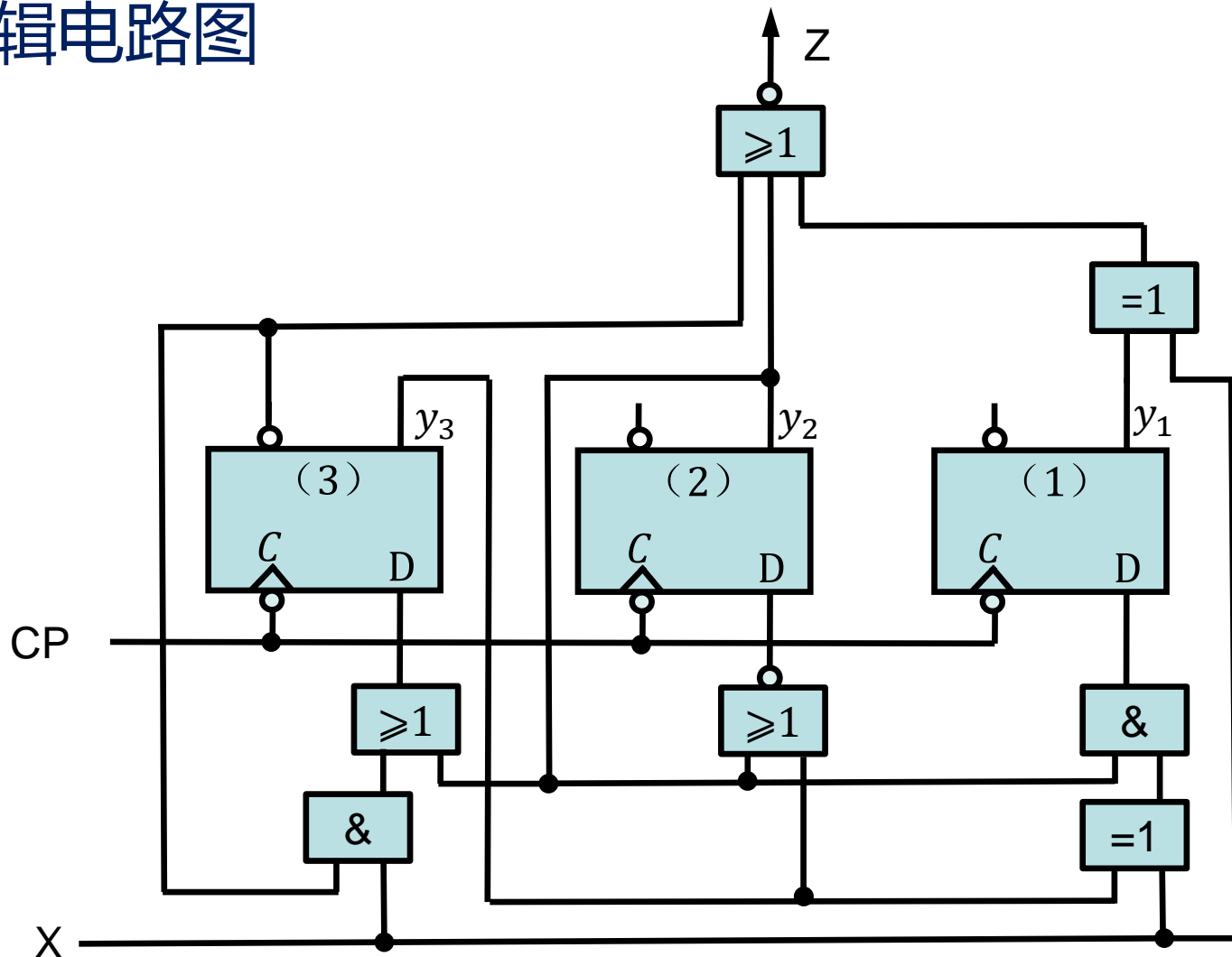
$$Z = \bar{x}y_3\bar{y}_2y_1 + xy_1$$

		Z			
		00	01	11	10
y_2y_1	00	0	1	0	0
	01	d	0	1	d
	11	d	d	d	d
	10	0	0	0	0

$$\begin{aligned} Z &= \bar{x}y_3\bar{y}_2y_1 + xy_3\bar{y}_2y_1 \\ &= y_3\bar{y}_2(x \odot y_1) \\ &= y_3\bar{y}_2 \cdot \overline{x \oplus y_1} \end{aligned}$$

同步时序逻辑电路设计

□ 逻辑电路图



Thank You!

