

数字逻辑设计

高翠芸

School of Computer Science

gaocuiyun@hit.edu.cn

同步时序逻辑设计

- 状态机基础
- 原始状态图和状态表
- 状态表化简
- 状态分配

状态机 (State machine) 定义

- 是一个**有向**图形，由一组**节点**和一组相应的**转移函数**组成。
- 状态机通过响应一系列事件而“运行”。
- 每个事件都在属于“当前”节点的转移函数的控制范围内，其中函数的范围是节点的一个子集。函数返回“下一个”（也许是同一个）节点。这些节点中至少有一个必须是终态。当到达终态， 状态机停止。

状态机 (State machine) 基础

- 包含一组**状态集** (states)、一个起始状态 (start state)、一组**输入符号集** (alphabet)、一个映射输入符号和当前状态到下一状态的**转换函数** (transition function) 的计算模型
- 状态是**状态变量 (state variable) 的集合**
 - 状态变量的值包含决定未来行为的所有信息
- 状态变化：从一个状态变为另一个状态
 - 变化的时间、如何变化

时序逻辑电路——状态机描述

- 时序电路的**状态 (state)**

- 具有 n 位二进制状态变量的电路有 2^n 种可能状态。
- 状态有限，可称为有限状态机 (FSM: Finite State Machine)
- 大多时序电路和几乎所有的状态机都会使用**边沿触发的D触发器**存储状态变量

- 时序电路状态变化

大多数时序电路状态发生变化的时间由**时钟信号**决定
状态变化函数——次态方程

状态机基础

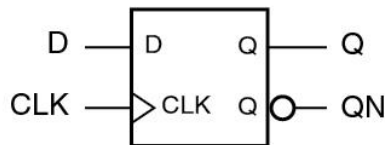
状态变化

- 大多数时序电路状态发生变化的时间由**时钟信号CLOCK**决定
 - 状态在时钟信号**上升沿**变化，称时钟信号**高电平有效**
 - 状态在时钟信号**下降沿**变化，称时钟信号**低电平有效**

触发器

- **大多时序电路和几乎所有的状态机都会使用边沿触发的D触发器存储状态变量**

(a)



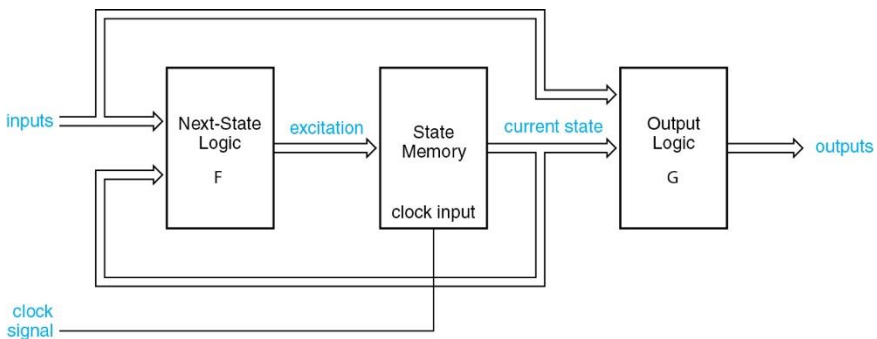
(b)

D	CLK	Q	QN
0		0	1
1		1	0
x	0	last Q	last QN
x	1	last Q	last QN

Mealy状态机 vs Moore状态机

状态机结构

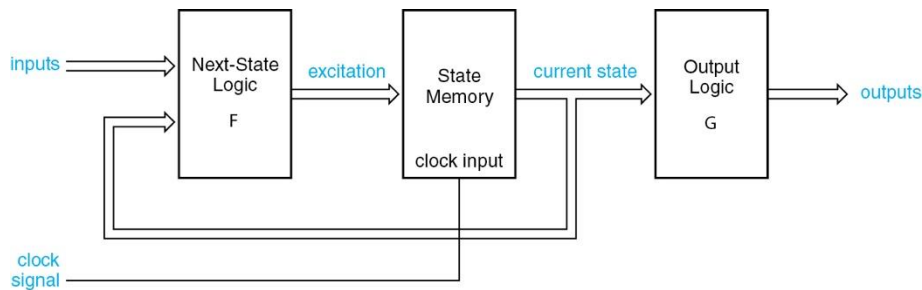
- 状态存储器 (state memory) 是存储状态机现态的一组触发器
- 状态机的次态, 由次态逻辑 (next-state logic) F 确定
- 状态机的输出, 由输出逻辑 (output logic) G 确定



Mealy状态机

次态 = F (现态, 输入)

输出 = G (现态, 输入)



Moore状态机

输出 = G (现态)

同步时序逻辑设计

- 状态机基础
- 原始状态图和状态表
- 状态表化简
- 状态分配

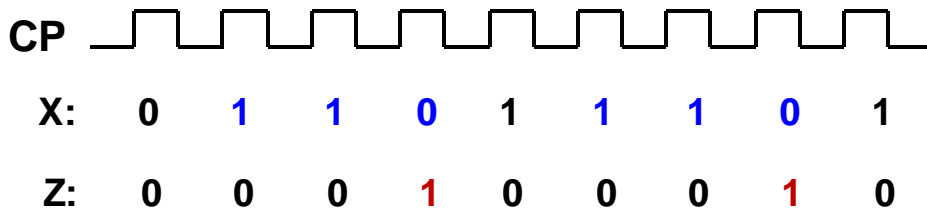
同步时序逻辑电路设计方法

利用触发器设计同步时序逻辑的方法

- (1) 根据需求 → 获得原始状态图、状态表
- (2) 最小化状态图、状态表
- (3) 状态编码（分配）→ 获得状态转移表
- (4) 状态转移表 } → 触发器激励（状态转移真值表）
触发器特征
- (5) 卡诺图化简 → { 激励（输入）函数表达式
输出函数表达式
- (6) 电路实现 (7) 检查无关项

示例：110序列检测器

例：利用JK触发器设计110序列检测器



1. 获得原始状态图和原始状态表

(1) 状态设定

S_0 ——初始状态，表示收到1位数据：“0”

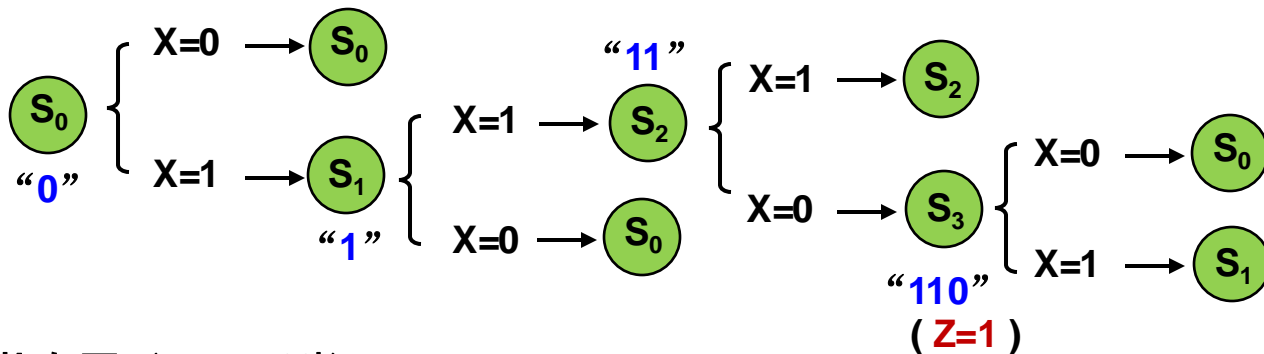
S_1 ——表示收到1位数据：“1”

S_2 ——表示收到2位数据：“11”

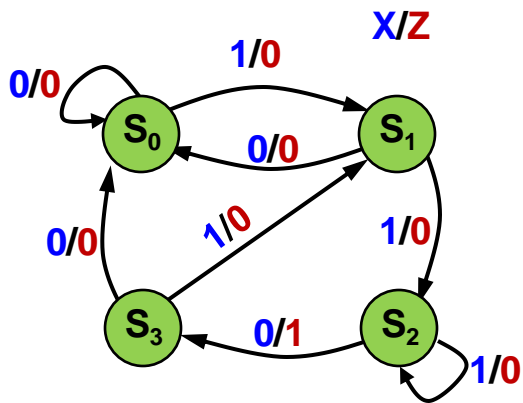
S_3 ——表示收到3位数据：“110”，此时输出标志 $Z=1$.

示例：110序列检测器

(2) 分析状态转换情况



(3) 原始状态图 (Mealy型)




(4) 原始状态表

现态 Q ⁿ	Q ⁿ⁺¹ / Z	
	X=0	X=1
S ₀	S ₀ / 0	S ₁ / 0
S ₁	S ₀ / 0	S ₂ / 0
S ₂	S ₃ / 1	S ₂ / 0
S ₃	S ₀ / 0	S ₁ / 0

示例：110序列检测器

2. 状态化简

现态 Q^n	Q^{n+1}/Z		
	$X=0$	$X=1$	
S_0	$S_0/0$	$S_1/0$	✓
S_1	$S_0/0$	$S_2/0$	
S_2	$S_3/1$	$S_2/0$	✓
S_3	$S_0/0$	$S_1/0$	



现态 Q^n	Q^{n+1}/Z	
	$X=0$	$X=1$
S_0	$S_0/0$	$S_1/0$
S_1	$S_0/0$	$S_2/0$
S_2	$S_0/1$	$S_2/0$

3. 状态分配

使用 2个JK触发器

y_2y_1

S_0 — 00

S_1 — 10

S_2 — 11

4. 状态转换真值表

输入		现态		次态		触发器				输出
X		Y_2^n	Y_1^n	Y_2^{n+1}	Y_1^{n+1}	J_2	K_2	J_1	K_1	Z
0	0	0	0	0	0	0	X	0	X	0
0	1	0	0	0	0	X	1	0	X	0
0	1	1	1	0	0	X	1	X	1	1
1	0	0	0	1	0	1	X	0	X	0
1	1	0	0	1	1	X	0	1	X	0
1	1	1	1	1	1	X	0	X	0	0
0	0	1	1	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X

示例：110序列检测器

4. 状态转换真值表

输入	现态		次态		触发器				输出
X	Y_2^n	Y_1^n	Y_2^{n+1}	Y_1^{n+1}	J_2	K_2	J_1	k_1	Z
0	0	0	0	0	0	X	0	X	0
0	1	0	0	0	X	1	0	X	0
0	1	1	0	0	X	1	X	1	1
1	0	0	1	0	1	X	0	X	0
1	1	0	1	1	X	0	1	X	0
1	1	1	1	1	X	0	X	0	0
0	0	1	X	X	X	X	X	X	X
1	0	1	X	X	X	X	X	X	X

5. 卡诺图化简

$Y_2^n Y_1^n$	00	01	11	10
0	0	X	X	X
1	1	X	X	X

$$J_2 = X$$

$Y_2^n Y_1^n$	00	01	11	10
0	X	X	1	1
1	X	X	0	0

$$K_2 = \bar{X}$$

$Y_2^n Y_1^n$	00	01	11	10
0	0	X	X	0
1	0	X	X	1

$$J_1 = XY_2^n$$

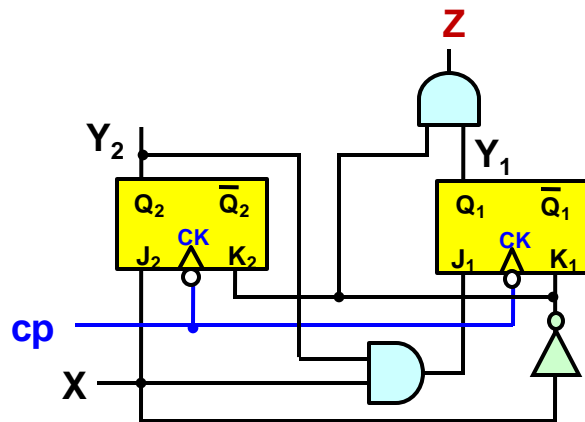
$Y_2^n Y_1^n$	00	01	11	10
0	X	X	1	X
1	X	X	0	X

$$K_1 = \bar{X}$$

$Y_2^n Y_1^n$	00	01	11	10
0	0	X	1	0
1	0	X	0	0

$$Z = \bar{X}Y_1^n$$

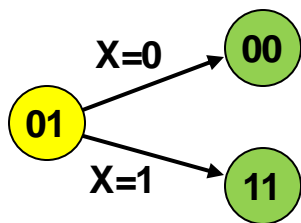
6. 电路实现



示例：110序列检测器

7. 检查无关项

$$\left\{ \begin{array}{l} J_1 = XY_2^n \\ K_1 = \bar{X} \\ J_2 = X \\ K_2 = \bar{X} \end{array} \right. \Rightarrow \left\{ \begin{array}{l} Y_1^{n+1} = XY_2^n \bar{Y}_1^n + XY_1^n \\ \quad = X(Y_1^n + Y_2^n) \\ Y_2^{n+1} = X\bar{Y}_2^n + XY_2^n \\ \quad = X \end{array} \right.$$



电路可以自启动

同步时序逻辑电路设计方法

利用触发器设计同步时序逻辑的方法

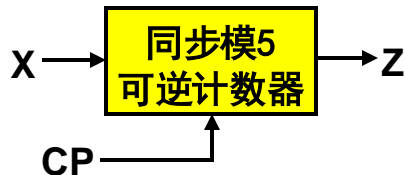
- (1) 根据需求 \rightarrow 获得原始状态图、状态表
- (2) 最小化状态图、状态表
- (3) 状态编码（分配） \rightarrow 获得状态转移表
- (4) 状态转移表
触发器特征 $\} \rightarrow$ 触发器激励
- (5) 卡诺图化简 \rightarrow $\left\{ \begin{array}{l} \text{激励（输入）函数表达式} \\ \text{输出函数表达式} \end{array} \right.$
- (6) 电路实现 (7) 检查无关项

直接构图法

直接构图法

- 1) 根据文字描述的设计要求, 先**假定一个初态**;
- 2) 从这个初态开始, **每加入一个输入取值**, 就可确定其次态和输出;
- 3) 该次态可能是现态本身, 也可能是已有的另一个状态, 或是新增加的一个状态。
- 4) 这个过程持续下去, 直至**每一个现态向其次态的转换都被考虑**, 并且不再构成新的状态。

例1: 给出同步模5可逆计数器的**状态表**

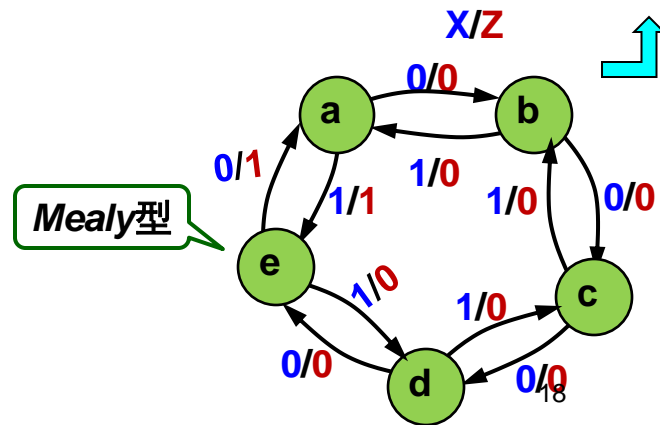


$X=0$: 加计数

$X=1$: 减计数

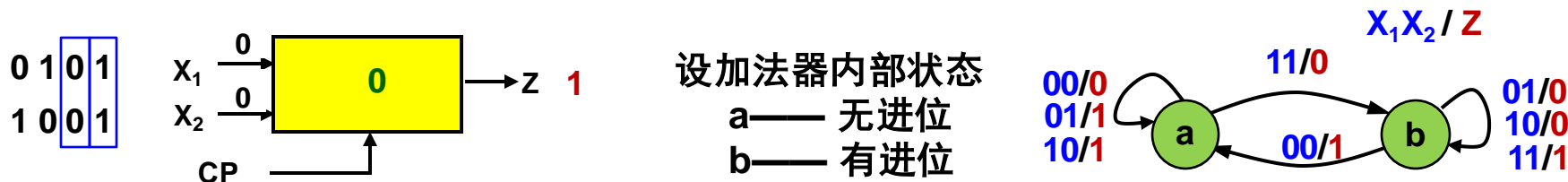
Z : 进位、借位输出标志

现态 Q^n	Q^{n+1} / Z	
	$X=0$	$X=1$
a	b / 0	e / 1
b	c / 0	a / 0
c	d / 0	b / 0
d	e / 0	c / 0
e	a / 1	d / 0



直接构图法

例2：给出同步二进制串行加法器的状态表



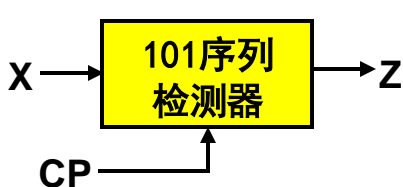
直接构图法

- 1) 根据文字描述的设计要求，先假定一个初态；
- 2) 从这个初态开始，每加入一个输入取值，就可确定其次态和输出；
- 3) 该次态可能是现态本身，也可能是已有的另一个状态，或是新增加的一个状态。
- 4) 这个过程持续下去，直至每一个现态向其次态的转换都已被考虑，并且不再构成新的状态。

现态 Q^n	Q^{n+1} / Z			
	$X_1X_2=00$	$X_1X_2=01$	$X_1X_2=10$	$X_1X_2=11$
a	a / 0	a / 1	a / 1	b / 0
b	a / 1	b / 0	b / 0	b / 1

序列检测—101序列检测器

例3：序列检测——给出同步Mealy型101序列检测器的状态表



X: 0 1 0 1 0 1 1 0 1
Z: 0 0 0 1 0 1 0 0 1

可重叠
检测

X: 0 1 0 1 0 1 0 1 1
Z: 0 0 0 1 0 0 0 1 0

不可重
叠检测

(1) 状态设定

S_0 ——初始状态，表示收到1位数据：“0”

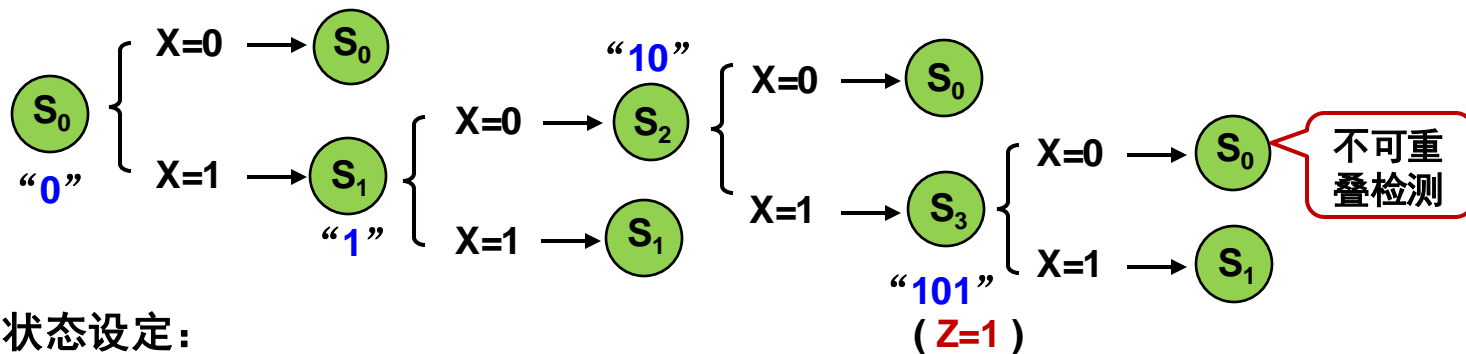
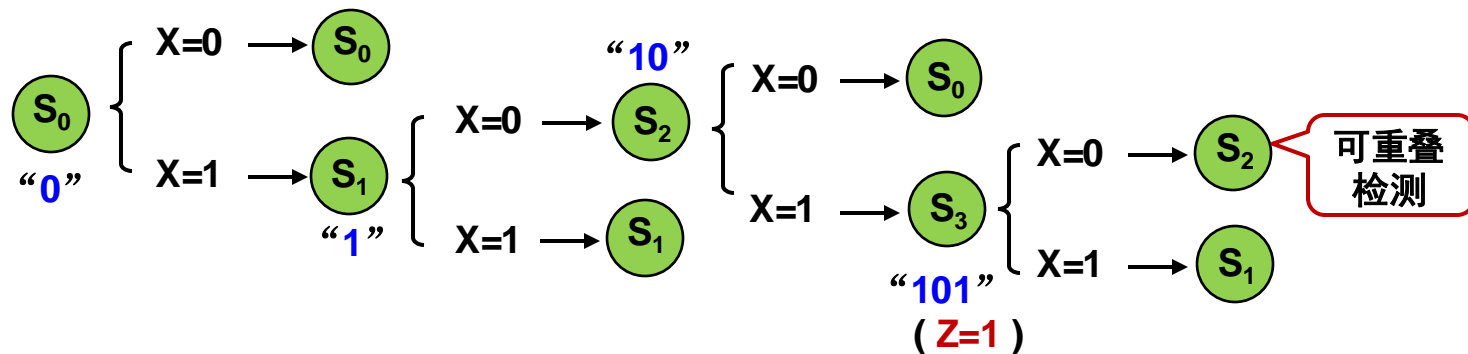
S_1 ——表示收到1位数据：“1”

S_2 ——表示收到2位数据：“10”

S_3 ——表示收到3位数据：“101”，此时输出标志 $Z=1$.

只标记感兴
趣的子串

101序列检测器

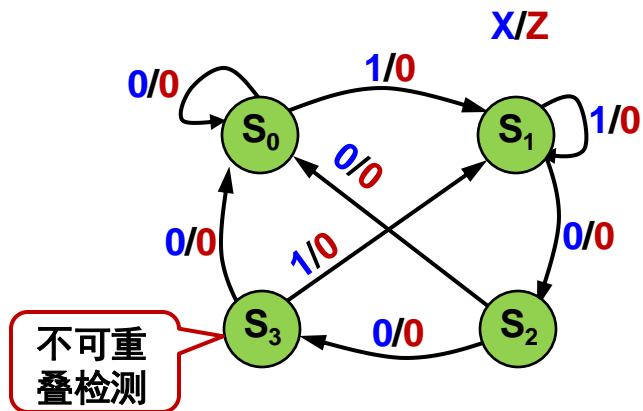


状态设定:

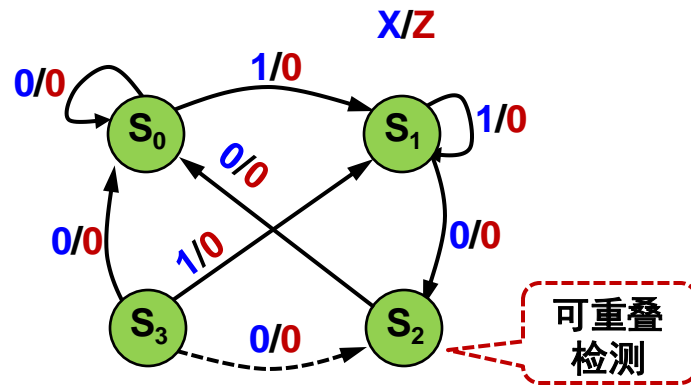
S_0 — 0 ; S_1 — 1 ;

S_2 — 10 ; S_3 — 101 , 且 $Z=1$

构造原始状态图和状态表



现态 Q^n	Q^{n+1} / Z	
	$X=0$	$X=1$
S_0	$S_0 / 0$	$S_1 / 0$
S_1	$S_2 / 0$	$S_1 / 0$
S_2	$S_0 / 0$	$S_3 / 1$
S_3	$S_0 / 0$	$S_1 / 0$



现态 Q^n	Q^{n+1} / Z	
	$X=0$	$X=1$
S_0	$S_0 / 0$	$S_1 / 0$
S_1	$S_2 / 0$	$S_1 / 0$
S_2	$S_0 / 0$	$S_3 / 1$
S_3	$S_2 / 0$	$S_1 / 0$

序列检测电路设计

序列检测的原始状态图构造方法总结

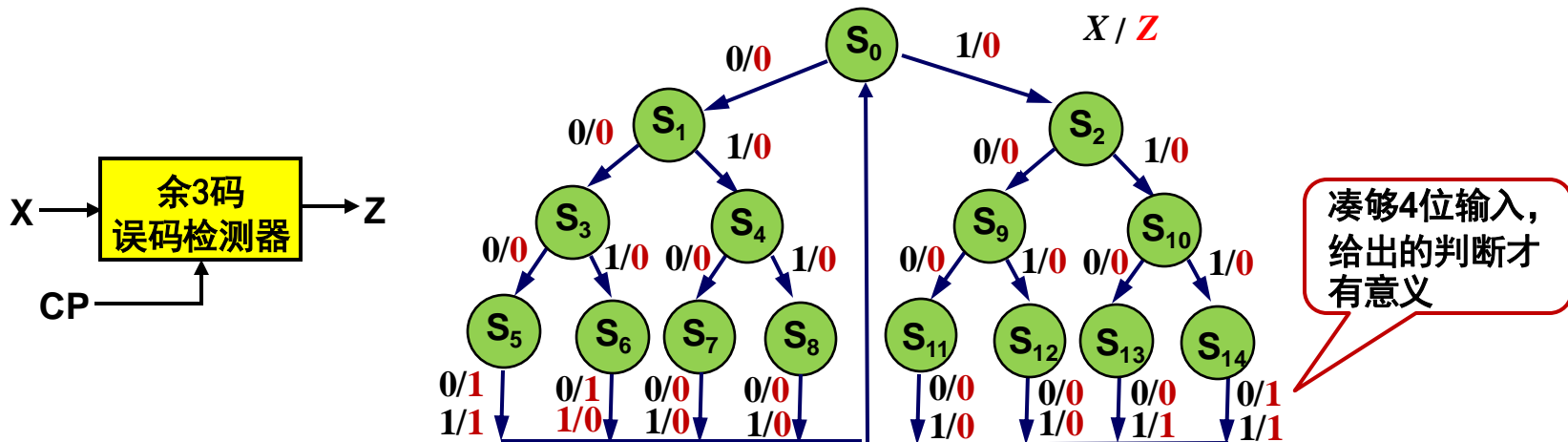
- (1) 检测器输入端收到1位数据时，有两种可能：0或1，分别用 S_0 和 S_1 标记这两个状态，通常用 S_0 表示**初始状态**。
- (2) 收到2位数据时，只标记我们**感兴趣的子串**，用 S_2 表示（例如 10）
- (3) 同理，收到3位数据时，只标记我们感兴趣的子串，用 S_3 表示（例如 101）……，直到把我们感兴趣的**完整子串也已标记为止**。
- (4) 从初始状态开始，采用**直接构图法**，将每一个当前状态在所有取值下的次态转换及输出情况已都考虑到，并且**没有遗漏为止**。

码制检测电路设计

例4：码制检测——建立一个余3码误码检测器的原始状态图和原始状态表

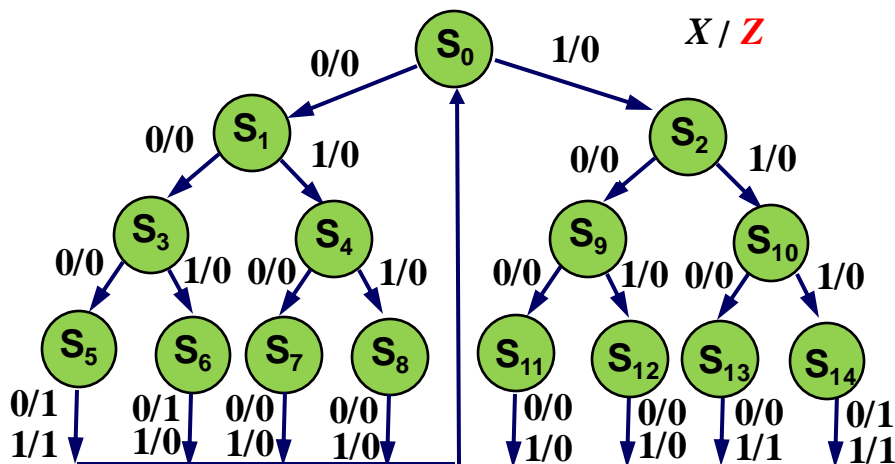
要求：

- 余3码高位在前、低位在后串行地加到检测器的输入端。
- 电路每接收一组代码（即在收到第4位代码时）判断。若是错误代码，则输出为1，否则输出为0，电路又回到初始状态并开始接收下一组代码。



原始状态图和状态表

原始状态图



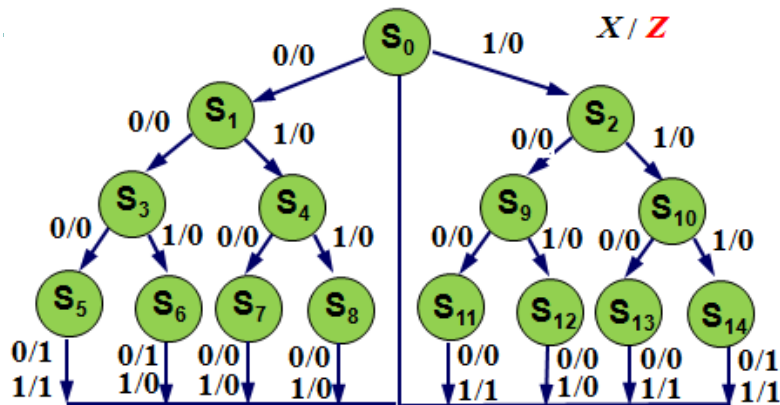
凑够4位输入，
给出的判断才
有意义

现态 Q^n	Q^{n+1} / Z	
	$X=0$	$X=1$
S_0	$S_1 / 0$	$S_2 / 0$
S_1	$S_3 / 0$	$S_4 / 0$
S_2	$S_9 / 0$	$S_{10} / 0$
S_3	$S_5 / 0$	$S_6 / 0$
S_4	$S_7 / 0$	$S_8 / 0$
S_5	$S_0 / 1$	$S_0 / 1$
S_6	$S_0 / 1$	$S_0 / 0$
S_7	$S_0 / 0$	$S_0 / 0$
S_8	$S_0 / 0$	$S_0 / 0$
S_9	$S_{11} / 0$	$S_{12} / 0$
S_{10}	$S_{13} / 0$	$S_{14} / 0$
S_{11}	$S_0 / 0$	$S_0 / 0$
S_{12}	$S_0 / 0$	$S_0 / 0$
S_{13}	$S_0 / 0$	$S_0 / 1$
S_{14}	$S_0 / 1$	$S_0 / 1$

码制检测电路设计

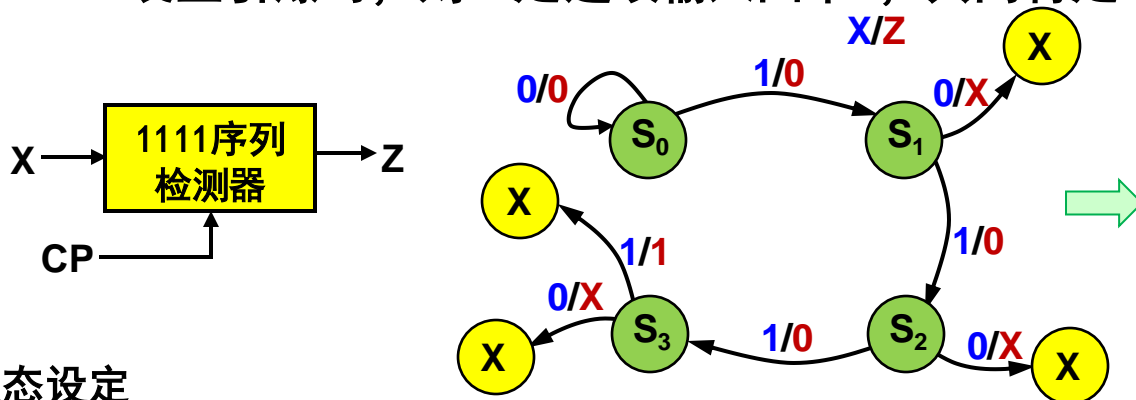
N位码制检测电路的原始状态图构造方法总结

- (1) 从初始状态 S_0 开始（这个初始状态没有特殊含义，仅仅代表一个起点），每来一个输入，次态总是分成左右两种情况。
- (2) 状态图由上至下分为N层：第一层代表起点；第二层代表检测器收到1位数据时，电路的状态情况；第三层代表检测器收到2位数据时，电路的状态情况……；直到第N层，代表检测器收到N-1位数据时，电路的状态情况。再来一位输入数据，则构成了N位待检测码制。此时，检测器可以给出判读，该码制正确还是错误。
- (3) 一轮检测结束，回到初始状态，等待下一组输入。



实例设计

例5：设计一个引爆装置的原始状态表。装置不引爆时，输入总为0；
装置引爆时，则一定连续输入四个1，其间肯定不再输入0。



现态 Q^n	Q^{n+1} / Z	
	$X=0$	$X=1$
S_0	$S_0 / 0$	$S_1 / 0$
S_1	X / X	$S_2 / 0$
S_2	X / X	$S_3 / 0$
S_3	X / X	$X / 1$

(1) 状态设定

S_0 ——初始状态，表示收到1位数据：“0”

S_1 ——表示收到1位数据：“1”

S_2 ——表示收到2位数据：“11”

S_3 ——表示收到3位数据：“111”

此时再收到一个“1”，输出标志 $Z=1$ 。

只标记感兴趣的子串

不完全定义状态表：包含任意项

状态表 { 完全定义状态表
不完全定义状态表

原始状态图的构造规律

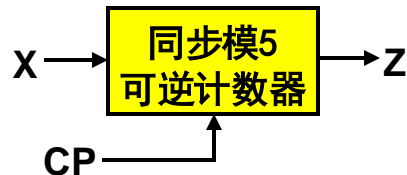
- 直接构图法
- 序列检测器
- N位码制检测电路

直接构图法

直接构图法

- 1) 根据文字描述的设计要求, 先**假定一个初态**;
- 2) 从这个初态开始, **每加入一个输入取值**, 就可确定其次态和输出;
- 3) 该次态可能是现态本身, 也可能是已有的另一个状态, 或是新增加的一个状态。
- 4) 这个过程持续下去, 直至**每一个现态向其次态的转换都被考虑**, 并且不再构成新的状态。

例1: 给出同步模5可逆计数器的**状态表**

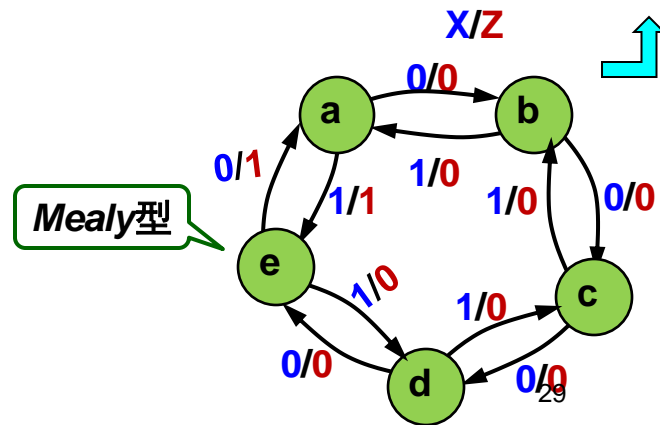


X=0: 加计数

X=1: 减计数

Z: 进位、借位输出标志

现态 Q^n	Q^{n+1} / Z	
	X=0	X=1
a	b / 0	e / 1
b	c / 0	a / 0
c	d / 0	b / 0
d	e / 0	c / 0
e	a / 1	d / 0



序列检测电路设计

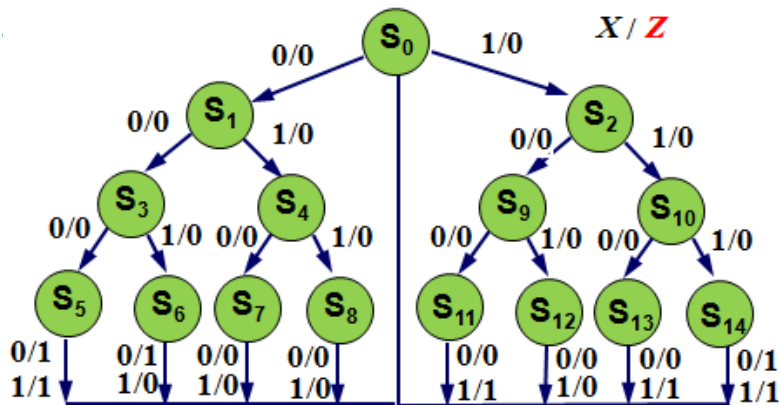
序列检测的原始状态图构造方法总结

- (1) 检测器输入端收到1位数据时，有两种可能：0或1，分别用 S_0 和 S_1 标记这两个状态，通常用 S_0 表示**初始状态**。
- (2) 收到2位数据时，只标记我们**感兴趣的子串**，用 S_2 表示（例如 10）
- (3) 同理，收到3位数据时，只标记我们感兴趣的子串，用 S_3 表示（例如 101）……，直到把我们感兴趣的**完整子串也已标记为止**。
- (4) 从初始状态开始，采用**直接构图法**，将每一个当前状态在所有取值下的次态转换及输出情况已都考虑到，并且**没有遗漏为止**。

码制检测电路设计

N位码制检测电路的原始状态图构造方法总结

- (1) 从初始状态 S_0 开始（这个初始状态没有特殊含义，仅仅代表一个起点），每来一个输入，次态总是分成左右两种情况。
- (2) 状态图由上至下分为N层：第一层代表起点；第二层代表检测器收到1位数据时，电路的状态情况；第三层代表检测器收到2位数据时，电路的状态情况……；直到第N层，代表检测器收到N-1位数据时，电路的状态情况。再来一位输入数据，则构成了N位待检测码制。此时，检测器可以给出判读，该码制正确还是错误。
- (3) 一轮检测结束，回到初始状态，等待下一组输入。



同步时序逻辑设计

- 状态机基础
- 原始状态图和状态表
- 状态表化简
- 状态分配

同步时序逻辑电路设计方法

利用触发器设计时序逻辑的方法

- (1) 根据需求 → 获得原始状态图、状态表
- (2) 最小化状态表
- (3) 状态编码（分配） → 获得状态转移表
- (4) 状态转移表
触发器特征 } → 触发器激励表
- (5) 卡诺图化简 → { 激励（输入）函数表达式
输出函数表达式
- (6) 电路实现 (7) 检查无关项

状态表化简

状态表的化简

时序电路的两个状态 S_i 和 S_j ，如果它们对每一个输入所产生的输出完全相同，且它们的次态等价，则这两个状态是等价的（可以合并为一个状态）——状态化简

（一）完全定义状态表的化简方法——隐含（蕴含）表法

- 俩俩比较原始状态表中的所有状态，找出能合并、不能合并、能否合并待定的状态对。
- 追踪能否合并待定的状态对，直至确定它们能合并或不能合并，从而找到原始状态表中的所有等价状态对。
- 基于这些等价状态对确定最大等价状态类，获得原始状态表的最小覆盖集，建立最简状态表

状态表化简

等价状态的判定条件

状态表中的任意两个状态 S_i 和 S_j 同时满足下列两个条件，它们可以合并为一个状态

1. 在所有不同的现输入下，**现输出**分别相同
2. 在所有不同的现输入下，**次态**分别为下列情况之一
 - (1) 两个次态完全相同
 - (2) 两个次态为其现态本身或交错
 - (3) 两个次态为状态对封闭链中的一个状态对
 - (4) 两个次态的某一后续状态对可以合并

状态合并的
必要条件

隐含表法化简状态表

隐含表(蕴含)法

等价状态的判定条件

状态表中的任意两个状态 s_i 和 s_j 同时满足下列两个条件，它们可以合并为一个状态

1. 在所有不同的现输入下，现输出分别相同

状态合并的
必要条件

2. 在所有不同的现输入下，次态分别为下列情况之一

(1) 两个次态完全相同

(2) 两个次态为其现态本身或交错

(3) 两个次态为状态对封闭链中的一个状态对

(4) 两个次态的某一后续状态对可以合并

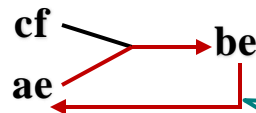
① 建立隐含表

② 比较

③ 追踪

b	cf✓					
c	X	X				
d	X	X	X			
e	be✓	ae✓ cf✓	X	X		
f	X	X	✓	X	X	
g	X	X	X	de	X	X
	a	b	c	d	e	f

竖列横排
掐头去尾



状态对
封闭链

等价状态对

{ a , b }、{ a , e }

{ b , e }、{ c , f }

隐含表法化简状态表

④ 获得最大等价状态类

等价状态类的定义——

If : $S_i \equiv S_j, S_j \equiv S_m$

Then: $S_i \equiv S_j \equiv S_m$, 即 $\{S_i, S_j, S_m\}$

最大等价状态类——

某一等价状态类不属于其他任何等价状态类

等价状态对:

$\{a, b\}$ 、 $\{a, e\}$

$\{b, e\}$ 、 $\{c, f\}$

最大等价状态类:

$\{a, b, e\}$ 、 $\{c, f\}$

Let $\begin{cases} q_1 = \{a, b, e\} \\ q_2 = \{c, f\} \\ q_3 = d \\ q_4 = g \end{cases}$

现态 Q^n	Q^{n+1}/Z	
	$X=0$	$X=1$
a	c / 0	b / 1
b	f / 0	a / 1
c	d / 0	g / 0
d	d / 1	e / 0
e	c / 0	e / 1
f	d / 0	g / 0
g	c / 1	d / 0

现态 Q^n	Q^{n+1}/Z	
	$X=0$	$X=1$
q_1	$q_2 / 0$	$q_1 / 1$
q_1	$q_2 / 0$	$q_1 / 1$
q_2	$q_3 / 0$	$q_4 / 0$
q_3	$q_3 / 1$	$q_1 / 0$
q_1	$q_2 / 0$	$q_1 / 1$
q_2	$q_3 / 0$	$q_4 / 0$
q_4	$q_2 / 1$	$q_3 / 0$

化简后的状态表

现态 Q^n	Q^{n+1}/Z	
	$X=0$	$X=1$
q_1	$q_2 / 0$	$q_1 / 1$
q_2	$q_3 / 0$	$q_4 / 0$
q_3	$q_3 / 1$	$q_1 / 0$
q_4	$q_2 / 1$	$q_3 / 0$

最小覆盖集: $\{q_1, q_2, q_3, q_4\}$

隐含表法化简状态表

例2：化简如下状态表

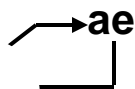
现态 Q^n	Q^{n+1}/Z			
	$X_1X_2=00$	$X_1X_2=01$	$X_1X_2=10$	$X_1X_2=11$
a	b/0	c/0	b/1	a/0
b	e/0	c/0	b/1	d/1
c	a/0	b/0	c/1	d/1
d	c/1	d/0	a/1	b/0
e	c/0	c/0	c/1	e/0



现态 Q^n	Q^{n+1}/Z			
	$X_1X_2=00$	$X_1X_2=01$	$X_1X_2=10$	$X_1X_2=11$
q_1	$q_2/0$	$q_2/0$	$q_2/1$	$q_1/0$
q_2	$q_1/0$	$q_2/0$	$q_2/1$	$q_3/1$
q_2	$q_1/0$	$q_2/0$	$q_2/1$	$q_3/1$
q_3	$q_2/1$	$q_3/0$	$q_1/1$	$q_2/0$
q_1	$q_2/0$	$q_2/0$	$q_2/1$	$q_1/0$



现态 Q^n	Q^{n+1}/Z			
	$X_1X_2=00$	$X_1X_2=01$	$X_1X_2=10$	$X_1X_2=11$
q_1	$q_2/0$	$q_2/0$	$q_2/1$	$q_1/0$
q_2	$q_1/0$	$q_2/0$	$q_2/1$	$q_3/1$
q_3	$q_2/1$	$q_3/0$	$q_1/1$	$q_2/0$



等价状态对:

$\{a,e\}$, $\{b,c\}$

Let $\begin{cases} q_1 = \{a,e\} \\ q_2 = \{b,c\} \\ q_3 = d \end{cases}$

同步时序逻辑设计

- 状态机基础
- 原始状态图和状态表
- 状态表化简
- 状态分配

状态编码（分配）

利用触发器设计时序逻辑的方法

- (1) 根据需求 \rightarrow 获得原始状态图、状态表
- (2) 最小化状态图、状态表
- (3) **状态编码（分配）** \rightarrow 获得状态转移表
- (4) 状态转移表
触发器特征 } \rightarrow 触发器激励表
- (5) 卡诺图化简 \rightarrow { 激励（输入）函数表达式
输出函数表达式
- (6) 电路实现 (7) 检查无关状态

状态编码（分配）

化简110 序列检测器的原始状态表

现态 Q^n	Q^{n+1} / Z	
	$X=0$	$X=1$
S_0	$S_0 / 0$	$S_1 / 0$
S_1	$S_0 / 0$	$S_2 / 0$
S_2	$S_3 / 1$	$S_2 / 0$
S_3	$S_0 / 0$	$S_1 / 0$



现态 Q^n	Q^{n+1} / Z	
	$X=0$	$X=1$
S_0	$S_0 / 0$	$S_1 / 0$
S_1	$S_0 / 0$	$S_2 / 0$
S_2	$S_0 / 1$	$S_2 / 0$

状态分配：

S_0 — 00

S_1 — 10

S_2 — 11



X	$Y_2^n Y_1^n$			
	00	01	11	10
0	0	X	X	0
1	0	X	X	1

$$J_1 = XY_2^n$$

X	$Y_2^n Y_1^n$			
	00	01	11	10
0	X	X	1	X
1	X	X	0	X

$$K_1 = \bar{X}$$

X	$Y_2^n Y_1^n$			
	00	01	11	10
0	0	X	1	0
1	0	X	0	0

$$Z = \bar{X}Y_1^n$$

X	$Y_2^n Y_1^n$			
	00	01	11	10
0	0	X	X	X
1	1	X	X	X

$$J_2 = X$$

X	$Y_2^n Y_1^n$			
	00	01	11	10
0	X	X	1	1
1	X	X	0	0

$$K_2 = \bar{X}$$

输入 X	现态		次态		触发器				输出 Z
	Y_2^n	Y_1^n	Y_2^{n+1}	Y_1^{n+1}	J_2	K_2	J_1	K_1	
0	0	0	0	0	0	X	0	X	0
0	1	0	0	0	X	1	X	1	1
0	1	0	0	1	X	1	0	X	0
1	0	1	0	1	1	X	0	X	0
1	1	1	1	0	X	0	X	0	0
1	1	1	1	0	X	0	1	X	0
0	0	X	X	X	X	X	X	X	X
1	0	X	X	X	X	X	X	X	X

状态编码（分配）

分配方案(1)

S_0 — 00

S_1 — 10

S_2 — 11



简单

$$\begin{cases} J_1 = XY_2^n \\ K_1 = \bar{X} \\ J_2 = X \\ K_2 = \bar{X} \\ Z = \bar{X}Y_1^n \end{cases}$$

分配方案(2)

S_0 — 00

S_1 — 11

S_2 — 10



复杂

$$\begin{cases} J_1 = X\bar{Y}_2^n \\ K_1 = 1 \\ J_2 = X \\ K_2 = \bar{X} + \bar{Y}_1^n \\ Z = \bar{X}Y_2^n\bar{Y}_1^n \end{cases}$$

状态分配

需要解决两个问题：

① 确定需要的触发器数量K

$$2^{K-1} \leq N \leq 2^K$$

K — 触发器数量

N — 最简状态数量

② 为状态表中的每一个状态分配二进制编码



力图获得一个最小代价的实现方案



电路实现代价与状态分配密切相关

状态编码（分配）

状态分配

规则

一种
经验法

1. 同一输入下，相同的次态所对应的**现态**应该给予相邻编码
2. 同一现态在不同输入下所对应的**次态**应给予相邻编码
3. 给定输入下，输出完全相同，**现态**编码应相邻

目的：尽量使卡诺图中更多的“1”（或“0”）相邻

注意：

- **初始状态**一般可以放在卡诺图的 0号单元格里
- 优先满足规则1和规则2
- 状态编码尽量按照相邻原则给予
- 对于多输出函数，规则3可以适当调高优先级

状态分配规则

➤ 规则1：次态相同，现态编码应相邻

$x=0$ 时, 次态 $(c,c) \rightarrow$ 现态 a,b
 $x=1$ 时, 次态 $(d,d) \rightarrow$ 现态 a,c

ab,ac 应相邻

现态 Q^n	Q^{n+1}/Z	
	$X=0$	$X=1$
a	c / 0	d / 0
b	c / 0	a / 0
c	b / 0	d / 0
d	a / 1	b / 1

➤ 规则2：同一现态对应的次态应给予相邻编码

现态 次态
 a \rightarrow (c,d)
 b \rightarrow (c,a)
 c \rightarrow (b,d)
 d \rightarrow (a,b)

cd,ca,bd,ab 应相邻

➤ 规则3：输出相同，现态编码应相邻

现态 输出
 a , b , c 0
ab,ac,bc 应相邻

规则

1. 同一输入下，相同的次态所对应的**现态**应该给予相邻编码
2. 同一现态在不同输入下所对应的**次态**应给予相邻编码
3. 给定输入下，输出完全相同，**现态**编码应相邻

很难找到一个最佳的状态分配方案

(a,b), (a,c) 应相邻, 满足规则1,2,3

a — 00, b — 01
 c — 10, d — 11

	0	1
0	a	b
1	c	d

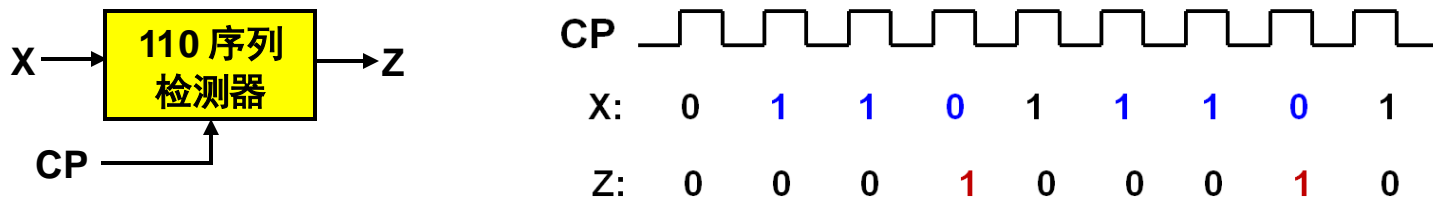
同步时序逻辑电路设计方法

利用触发器设计时序逻辑的方法

- (1) 根据需求 \rightarrow 获得原始状态图、状态表
- (2) 最小化状态图、状态表
- (3) 状态编码（分配） \rightarrow 获得状态转移表
- (4) 状态转移表
触发器特征 } \rightarrow 触发器激励表
- (5) 卡诺图化简 \rightarrow $\left\{ \begin{array}{l} \text{激励（输入）函数表达式} \\ \text{输出函数表达式} \end{array} \right.$
- (6) 电路实现 (7) 检查无关状态

完整电路设计过程示例

例：利用JK触发器设计110序列检测器



1. 获得原始状态图和原始状态表

(1) 状态设定

S_0 ——初始状态，表示收到1位数据：“0”

S_1 ——表示收到1位数据：“1”

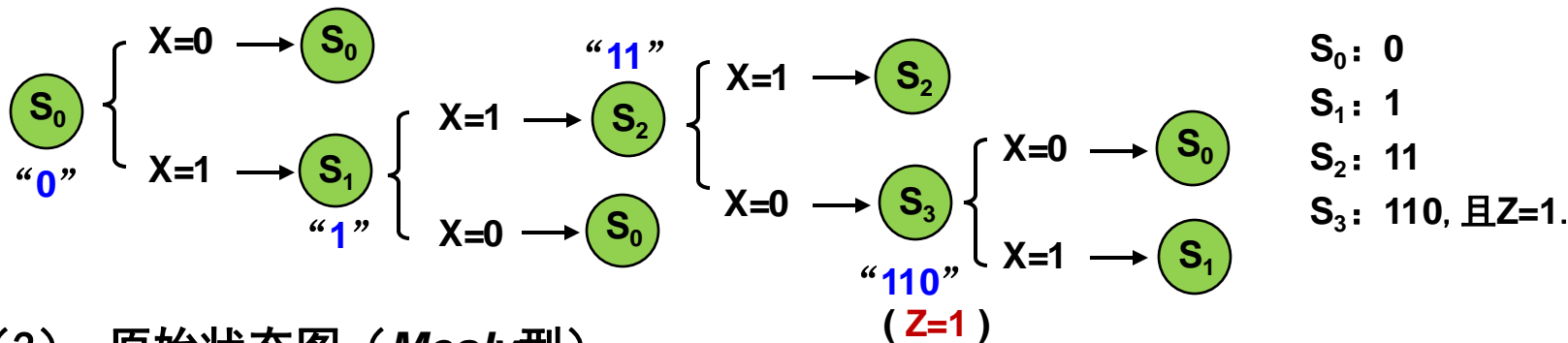
S_2 ——表示收到2位数据：“11”

只标记感兴趣的子串

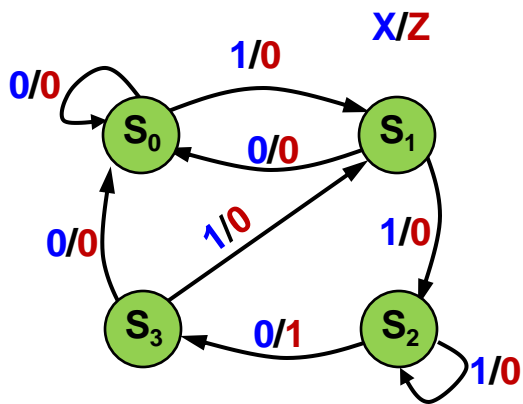
S_3 ——表示收到3位数据：“110”，此时输出标志 $Z=1$.

利用JK触发器设计110序列检测器

(2) 分析状态转换情况



(3) 原始状态图 (Mealy型)



(4) 原始状态表

现态 Q^n	Q^{n+1} / Z	
	$X=0$	$X=1$
S_0	$S_0 / 0$	$S_1 / 0$
S_1	$S_0 / 0$	$S_2 / 0$
S_2	$S_3 / 1$	$S_2 / 0$
S_3	$S_0 / 0$	$S_1 / 0$

利用JK触发器设计110序列检测器

$J_2 K_2$: 看 $Q_2^n \rightarrow Q_2^{n+1}$
 $J_1 K_1$: 看 $Q_1^n \rightarrow Q_1^{n+1}$

2. 状态化简

现态 Q^n	Q^{n+1}/Z	
	$X=0$	$X=1$
S_0	$S_0/0$	$S_1/0$
S_1	$S_0/0$	$S_2/0$
S_2	$S_3/1$	$S_2/0$
S_3	$S_0/0$	$S_1/0$



现态 Q^n	Q^{n+1}/Z	
	$X=0$	$X=1$
S_0	$S_0/0$	$S_1/0$
S_1	$S_0/0$	$S_2/0$
S_2	$S_0/1$	$S_2/0$

3. 状态分配

使用2个JK触发器

$Y_2 Y_1$
 S_0 — 00
 S_1 — 10
 S_2 — 11

JK触发器驱动表

Q_n	\rightarrow	Q_{n+1}	J	K
0	\rightarrow	0	0	X
0	\rightarrow	1	1	X
1	\rightarrow	0	X	1
1	\rightarrow	1	X	0

4. 状态转换真值表

输入 现态			次态		触发器				输出
X	Y_2^n	Y_1^n	Y_2^{n+1}	Y_1^{n+1}	J_2	K_2	J_1	K_1	Z
0	0	0	0	0	0	X	0	X	0
0	1	0	0	0	X	1	0	X	0
0	1	1	0	0	X	1	X	1	1
1	0	0	1	0	1	X	0	X	0
1	1	0	1	1	X	0	1	X	0
1	1	1	1	1	X	0	X	0	0
0	0	1	X	X	X	X	X	X	X
1	0	1	X	X	X	X	X	X	X

规则

1. 同一输入下，相同的次态所对应的**现态**应该给予相邻编码
2. 同一现态在不同输入下所对应的**次态**应给予相邻编码
3. 给定输入下，输出完全相同，**现态**编码应相邻⁵³

利用JK触发器设计110序列检测器

4. 状态转换真值表

输入	现态		次态		触发器				输出
X	Y_2^n	Y_1^n	Y_2^{n+1}	Y_1^{n+1}	J_2	K_2	J_1	k_1	Z
0	0	0	0	0	0	X	0	X	0
0	1	0	0	0	X	1	0	X	0
0	1	1	0	0	X	1	X	1	1
1	0	0	1	0	1	X	0	X	0
1	1	0	1	1	X	0	1	X	0
1	1	1	1	1	X	0	X	0	0
0	0	1	X	X	X	X	X	X	X
1	0	1	X	X	X	X	X	X	X

5. 卡诺图化简

$Y_2^n Y_1^n$	00	01	11	10
0	0	X	X	X
1	1	X	X	X

$$J_2 = X$$

$Y_2^n Y_1^n$	00	01	11	10
0	X	X	1	1
1	X	X	0	0

$$K_2 = \bar{X}$$

$Y_2^n Y_1^n$	00	01	11	10
0	0	X	X	0
1	0	X	X	1

$$J_1 = XY_2^n$$

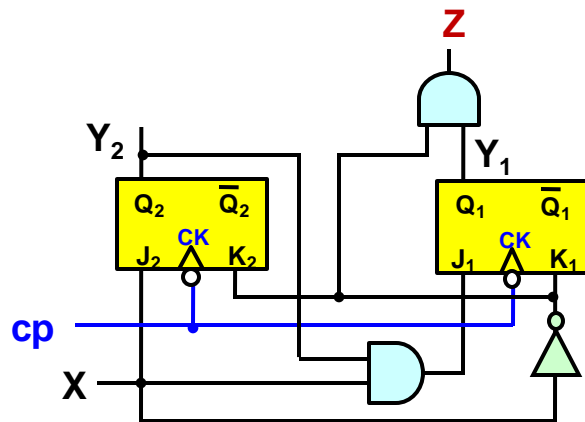
$Y_2^n Y_1^n$	00	01	11	10
0	X	X	1	X
1	X	X	0	X

$$K_1 = \bar{X}$$

$Y_2^n Y_1^n$	00	01	11	10
0	0	X	1	0
1	0	X	0	0

$$Z = \bar{X}Y_1^n$$

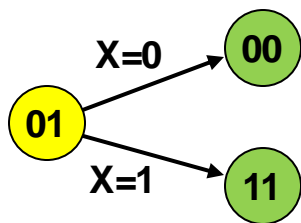
6. 电路实现



利用JK触发器设计110序列检测器

7. 检查无关项

$$\left\{ \begin{array}{l} J_1 = XY_2^n \\ K_1 = \bar{X} \\ J_2 = X \\ K_2 = \bar{X} \end{array} \right. \Rightarrow \left\{ \begin{array}{l} Y_1^{n+1} = XY_2^n \bar{Y}_1^n + XY_1^n \\ \quad = X(Y_1^n + Y_2^n) \\ Y_2^{n+1} = X\bar{Y}_2^n + XY_2^n \\ \quad = X \end{array} \right.$$



电路可以自启动

同步时序逻辑电路设计方法

利用触发器设计时序逻辑的方法

- (1) 根据需求 \rightarrow 获得原始状态图、状态表
- (2) 最小化状态图、状态表
- (3) 状态编码（分配） \rightarrow 获得状态转移表
- (4) 状态转移表
触发器特征 } \rightarrow 触发器激励表
- (5) 卡诺图化简 \rightarrow { 激励（输入）函数表达式
输出函数表达式
- (6) 电路实现 (7) 检查无关状态

时序逻辑电路设计

- 状态机基础
- 原始状态图和状态表
- 状态表化简
- 状态分配