利用触发器设计同步时序逻辑——示例

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

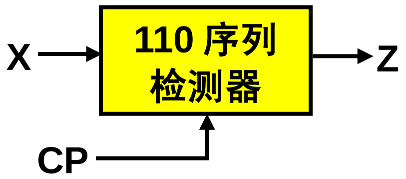
（4）状态转移表、触发器特征 → 触发器激励

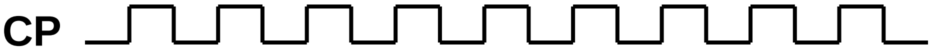
（5）卡诺图化简 → 激励（输入）函数表达式、输出函数表达式

（6）电路实现　（7）检查无关项

利用触发器设计同步时序逻辑——示例

例：利用JK触发器设计110序列检测器





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

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”

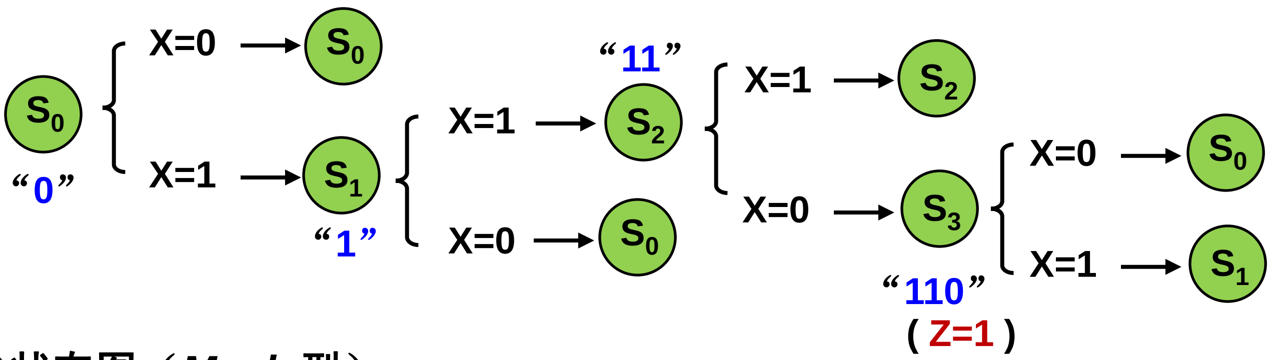
S_1——表示收到1位数据：“1”

S_2——表示收到2位数据：“11”

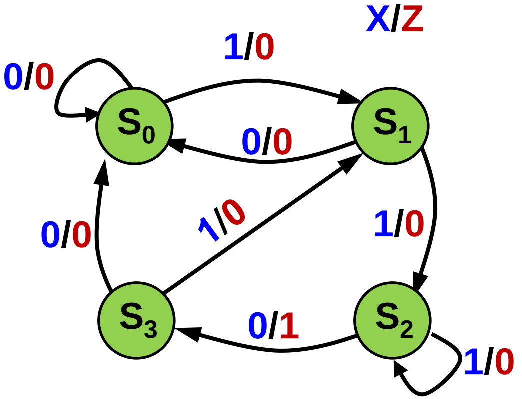
S_3——表示收到3位数据：“110”，此时输出标志 Z=1.

利用触发器设计同步时序逻辑——示例

（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

利用触发器设计同步时序逻辑——示例

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

利用触发器设计同步时序逻辑——示例

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	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

5. 卡诺图化简

$X \backslash Y_2^nY_1^n$	00	01	11	10
0	0	X	X	X
1	1	X	X	X

$$J_2 = X$$

$X \backslash Y_2^nY_1^n$	00	01	11	10
0	X	X	1	1
1	X	X	0	0

$$K_2 = \overline{X}$$

$X \backslash Y_2^nY_1^n$	00	01	11	10
0	0	X	X	0
1	0	X	X	1

$$J_1 = XY_2^n$$

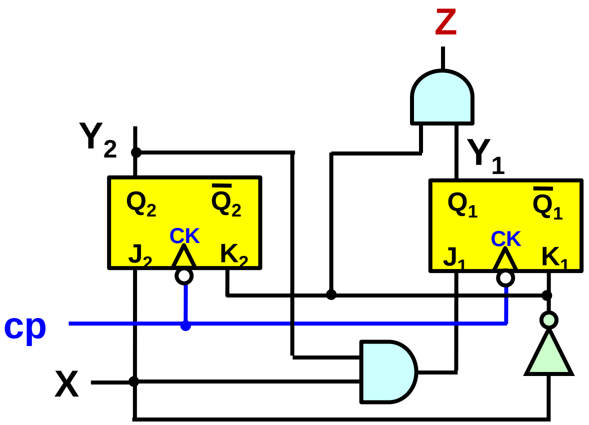
$X \backslash Y_2^nY_1^n$	00	01	11	10
0	X	X	1	X
1	X	X	0	X

$$K_1 = \overline{X}$$

$X \backslash Y_2^nY_1^n$	00	01	11	10
0	0	X	1	0
1	0	X	0	0

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

6. 电路实现

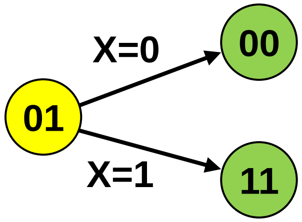


利用触发器设计时序逻辑——示例

7. 检查无关项

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

$$\begin{cases} Y_1^{n+1} = XY_2^n\overline{Y_1}^n + XY_1^n \\ \qquad\quad = X(Y_1^n + Y_2^n) \\ Y_2^{n+1} = X\overline{Y_2}^n + XY_2^n \\ \qquad\quad = X \end{cases}$$



电路可以自启动

利用触发器设计时序逻辑_构造原始状态图和状态表

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

构造原始状态图和状态表

直接构图法

1）根据文字描述的设计要求，先假定一个初态；

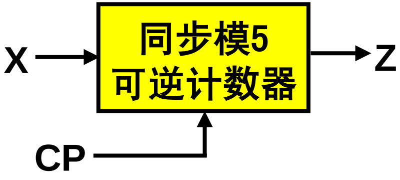
2）从这个初态开始，每加入一个输入取值，就可确定其次态和输出；

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=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

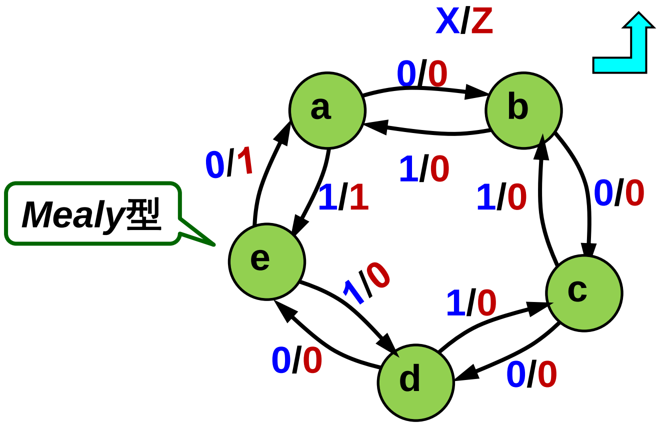
例1：给出同步模5可逆计数器的状态表



X=0： 加计数

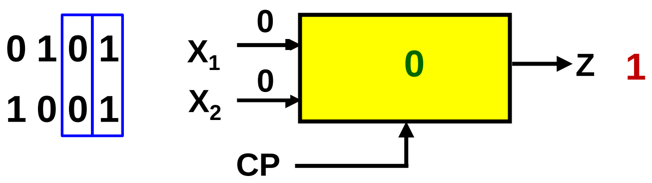
X=1： 减计数

Z: 进位、借位输出标志



构造原始状态图和状态表

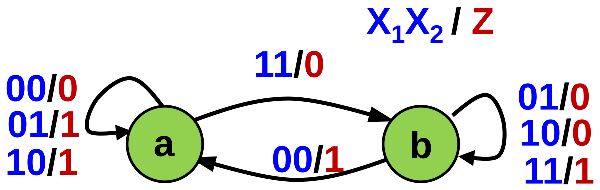
例2：给出同步二进制串行加法器的状态表



设加法器内部状态

a—— 无进位

b—— 有进位



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；

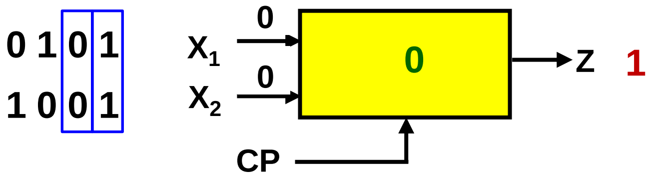
3）该次态可能是现态本身，也可能是已有的另一个状态，或是新增加的一个状态。

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

构造原始状态图和状态表

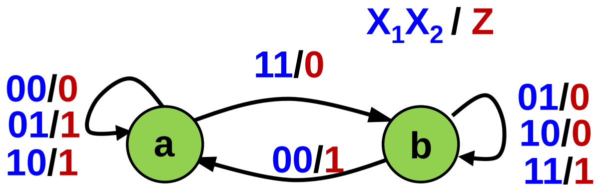
例2：给出同步二进制串行加法器的状态表



设加法器内部状态

a—— 无进位

b—— 有进位



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

利用触发器设计时序逻辑_构造原始状态图和状态表

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

构造原始状态图和状态表

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”

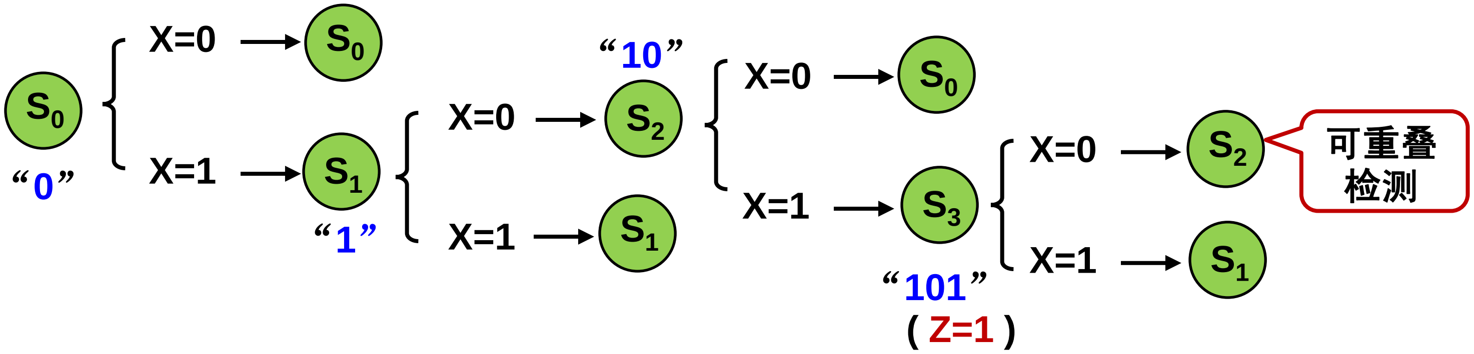
S_1——表示收到1位数据：“1”

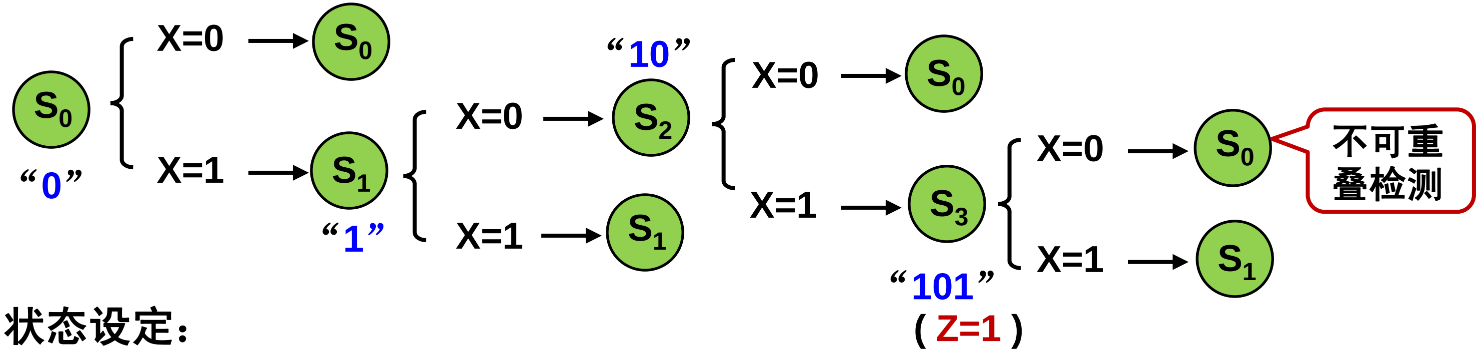
S_2——表示收到2位数据：“10”

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

构造原始状态图和状态表



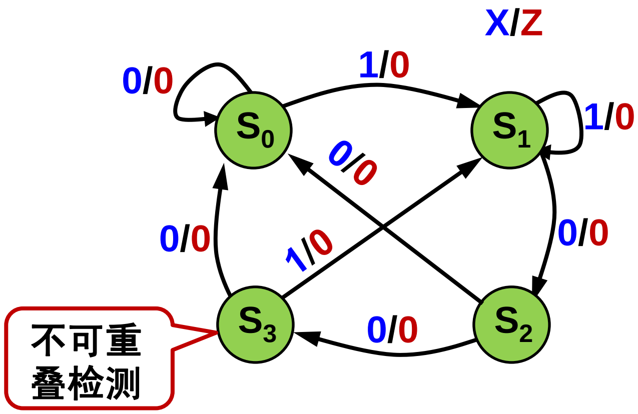


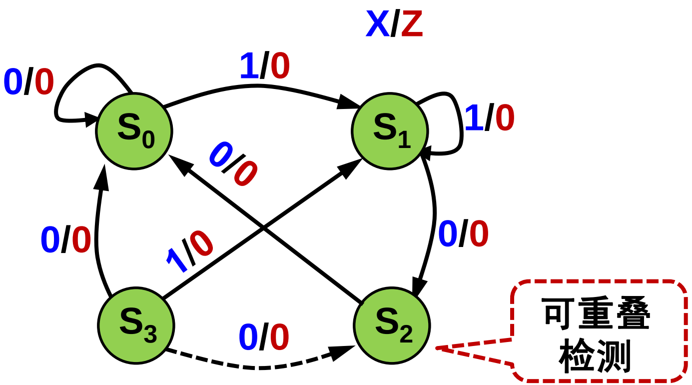
状态设定:

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

构造原始状态图和状态表

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。

利用触发器设计时序逻辑_构造原始状态图和状态表

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

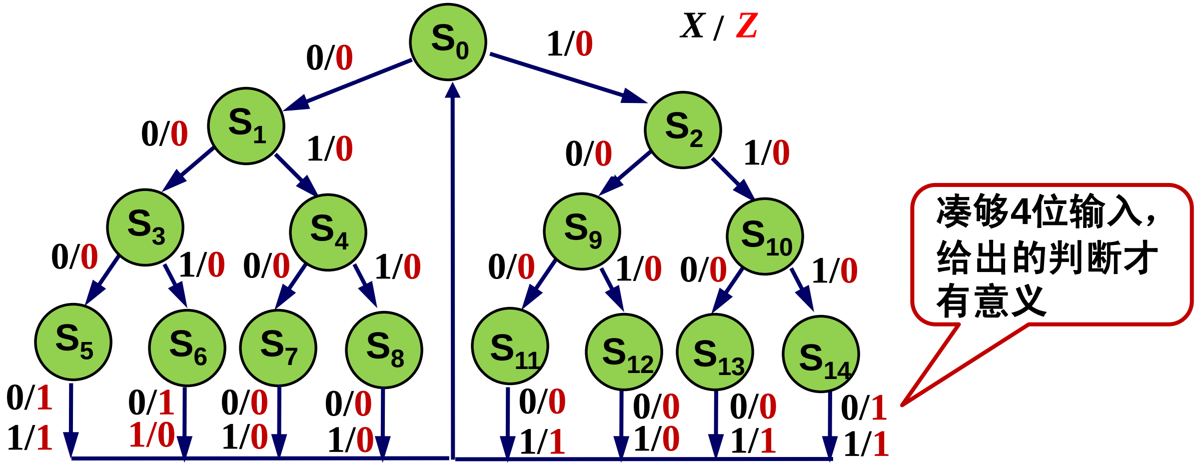
（6）电路实现　（7）　检查无关项

构造原始状态图和状态表

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

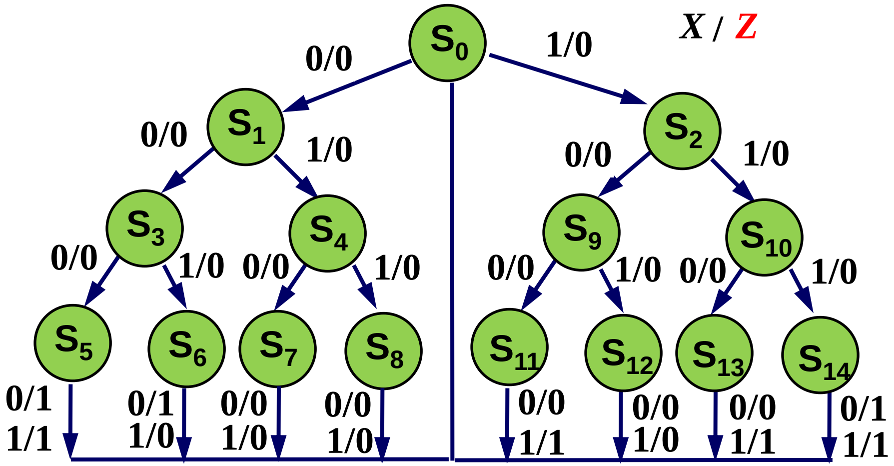
要求：

- 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。



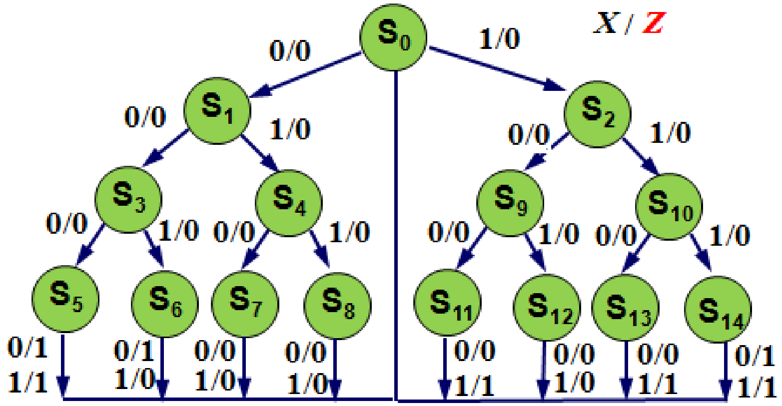
构造原始状态图和状态表

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	Q^{n+1}/ Z X=1
S_0	S_0 / 0	S_1 / 0
S_1	S_3 / 0	S_1 / 0
S_2	S_9 / 0	S_3 / 0
S_3	S_5 / 0	S_1 / 0
S_4	S_7 / 0	S_6 / 0
S_5	S_0 / 1	S_8 / 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_0 / 0
S_{10}	S_{13} / 1	S_{12} / 0
S_{11}	S_0 / 0	S_{14} / 0
S_{12}	S_0 / 0	S_0 / 0
S_3	S_0 / 0	S_0 / 0
S_{14}	S_0 / 0	S_0 / 1
S_{15}	S_0 / 1	S_0 / 1



N位码制检测电路的原始状态图构造方法总结

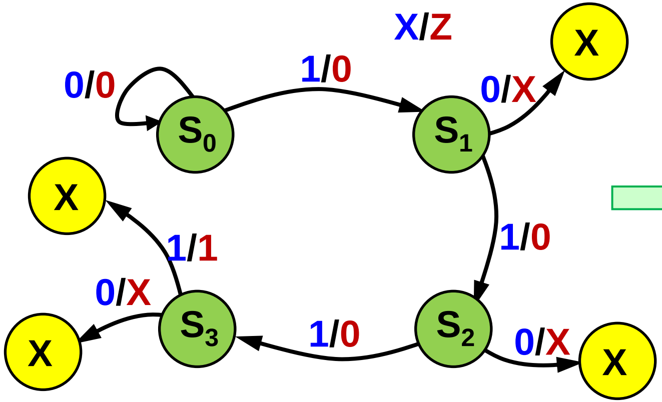
（1）从初始状态S_0开始（这个初始状态没有特殊含义，仅仅代表一个起点），每来一个输入，次态总是分成左右两种情况。

（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。

构造原始状态图和状态表

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

不完全定义状态表：包含任意项

（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

状态表 { 完全定义状态表 / 不完全定义状态表 }

利用触发器设计时序逻辑_状态表化简

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

利用触发器设计时序逻辑_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

利用触发器设计时序逻辑_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

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

利用触发器设计时序逻辑_状态表化简

隐含表(蕴含)法

例1：化简如下状态表

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

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

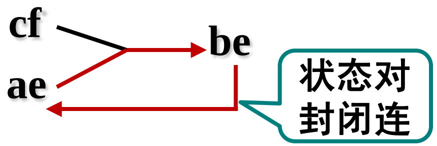
① 建立隐含表

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 cd~~ X	X	X
	a	b	c	d	e	f

竖列横排
掐头去尾



等价状态对

{ a ,b }、{ a ,e }

{ b ,e }、{ c ,f }

利用触发器设计时序逻辑_状态表化简

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
q_1 = { a , b ,e }
q_2 = { c ,f }
q_3 = d
q_4 = g

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： $\{q_1 , q_2 , q_3 , q_4\}$

利用触发器设计时序逻辑_状态表化简

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

b	X			
c	X	ae ✓		
d	X	X	X	
e	bc ✓	X	X	X
	a	b	c	d

$bc \rightleftarrows ae$

等价状态对:

{a,e}， {b,c}

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

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

不完全定义状态表的化简

（二）不完全定义状态表的化简方法——隐含（蕴含）表法

完全定义状态表化简：寻找等价状态；不完全定义状态表化简：寻找相容状态；

相容状态——输出与次态的确定部分满足合并条件的两个状态（如a和b）称为相容状态，或称相容状态对，记为(a, b)。

相容状态无传递性—— 若状态 S_i和 S_j相容，状态 S_j和 S_m相容，则状态 S_i和 S_m 不一定相容，即相容状态无传递性。

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

a和b相容，a和c相容，但b和c不相容

相容状态类——俩俩相容的状态集合

If：（S_i , S_j），（S_j , S_m），（S_i , S_m）

Then:（ S_i , S_j , S_m ）

最大相容状态类—— 某一相容状态类不属于其他任何相容状态类

不完全定义状态表的化简

例：化简如下状态表

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

① 建立隐含表

b	ac ✓			
c	ad ✓	X		
d	✓	✓	✓	
e	✓	X	ad ✓	bc X
	a	b	c	d

② 比较　　**③ 追踪**

④ 相容状态对

(ab)、(ac) 、(ad) 、

(ae) 、(bd) 、(cd)、 (ce)

⑤ 最大相容类

直观法

(ab)、(ad)、(bd) → (abd)

(ac)、(ad)、(cd) → (acd)

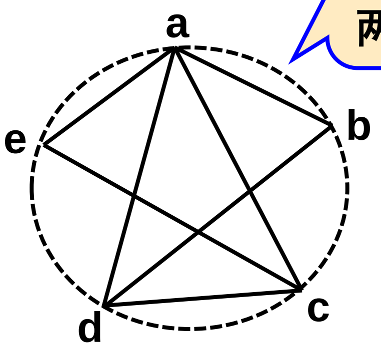
(ac)、(ae)、(ce) → (ace)

图形法

(abd)

(acd)

(ace)



- 圆周上的点：代表状态
- 点与点之间的连线：表示两个状态之间的相容关系

所有点之间都有连线的多边形构成一个最大相容类

不完全定义状态表的化简

⑥ 确定原始状态表的最小闭合覆盖集

最小闭合覆盖集应满足的三个条件

1. 满足覆盖性：覆盖全部原始状态，不得遗漏，即原始状态中的每个状态至少包含于该集的一个相容类（或最大相容类）
2. 满足闭合性：该集的任一个相容类（或最大相容类）在任何输入下所产生的次态应属于该集的某个相容类（或最大相容类）
3. 满足最小性：在满足上述两个要求的前提下，该集的相容类（或最大相容类）应为最少

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

最大相容类

(abd), (acd), (ace)

相容状态对

(ab), (ac), (ad),
(ae), (bd), (cd), (ce)

➢找出覆盖集，方案很多，如：

[abd , ace] [abd , ce] [acd , ab , ae] ……

➢为满足最小性，选取相容类（或最大相容类）个数最少的集合：

[abd , ace] [abd , ce] [ace , bd]

不完全定义状态表的化简

➢ 讨论闭合性：　分别考察[abd , ace], [abd , ce], [ace , bd]

满足闭合性

abd →

	X=0	X=1
次态	<u>ac</u>	b

b → 是abd的子集

ac → 是ace的子集

ace →

	X=0	X=1
次态	<u>ad</u>	c

c → 是ace的子集

ad → 是abd的子集

不满足闭合性

abd →

	X=0	X=1
次态	<u>ac</u>	b

b → 是abd的子集

ac → 不是abd的子集，也不是ce的子集

ce →

	X=0	X=1
次态	<u>ad</u>	c

c → 是ace的子集

ad → 是abd的子集

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

不完全定义状态表的化简

➢ 讨论闭合性： 分别考察[abd , ace], [abd , ce], [ace , bd]

不满足闭合性

bd →

	X=0	X=1
次态	c	b

b → 是bd的子集

c → 是ace的子集

ace →

	X=0	X=1
次态	ad	c

c → 是ace的子集

ad → 不是bd、也不是ace的子集

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

⇩

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

所以：最小闭合覆盖集为—— [abd， ace]

⑦ 建立状态表

设：q_1 = (a b d)
q_2 = (a c e)

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

利用触发器设计时序逻辑_状态编码

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）检查无关状态

利用触发器设计时序逻辑_状态编码

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

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	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

X \ $Y_2^nY_1^n$	00	01	11	10
0	0	X	X	0
1	0	X	X	1

$J_1 = XY_2^n$

X \ $Y_2^nY_1^n$	00	01	11	10
0	X	X	1	X
1	X	X	0	X

$K_1 = \overline{X}$

X \ $Y_2^nY_1^n$	00	01	11	10
0	0	X	1	0
1	0	X	0	0

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

X \ $Y_2^nY_1^n$	00	01	11	10
0	0	X	X	X
1	1	X	X	X

$J_2 = X$

X \ $Y_2^nY_1^n$	00	01	11	10
0	X	X	1	1
1	X	X	0	0

$K_2 = \overline{X}$

利用触发器设计时序逻辑_状态编码

分配方案(1)

S_0 —— 00
S_1 —— 10
S_2 —— 11

简单

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

分配方案(2)

S_0 —— 00
S_1 —— 11
S_2 —— 10

复杂

$$\begin{cases} J_1 = X\overline{Y_2^n} \\ K_1 = 1 \\ J_2 = X \\ K_2 = \overline{X}+\overline{Y_1^n} \\ Z = \overline{X}\,Y_2^n\,\overline{Y_1^n} \end{cases}$$

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

利用触发器设计时序逻辑_状态编码

状态分配

一种
经验法

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

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

利用触发器设计时序逻辑_状态编码

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1：次态相同，现态编码应相邻**

x=0时, 次态 (c,c) →现态 a,b
x=1时, 次态 (d,d) →现态 a,c

} ab,ac 应相邻

➢ **规则2：同一现态对应的次态应给予相邻编码**

现态		次态
a	→	(c,d)
b	→	(c,a)
c	→	(b,d)
d	→	(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

(a,b), (a,c) 应相邻, 满足规则1,2,3

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——00, b——01
c——10, d——11

	0	1
0	a	b
1	c	d

利用触发器设计时序逻辑_状态编码

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

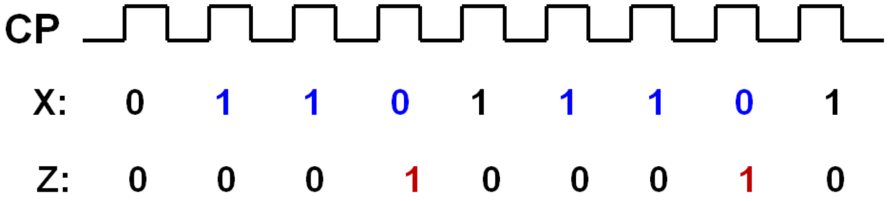
（4）状态转移表、触发器特征 ⟶ 触发器激励表

（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）检查无关状态

利用触发器设计时序逻辑_状态编码

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”

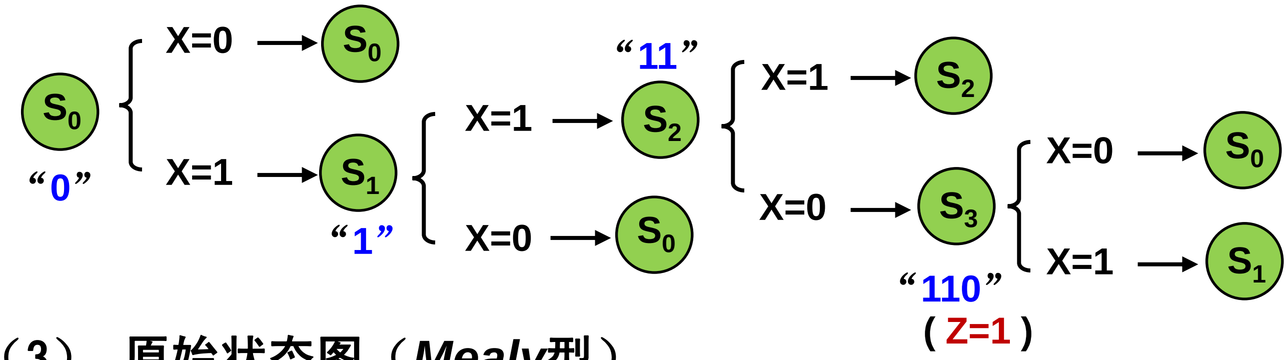
S_2——表示收到2位数据：“11”

S_3——表示收到3位数据：“110”，此时输出标志 Z=1.

只标记感兴趣的子串

利用触发器设计时序逻辑_状态编码

（2）分析状态转换情况



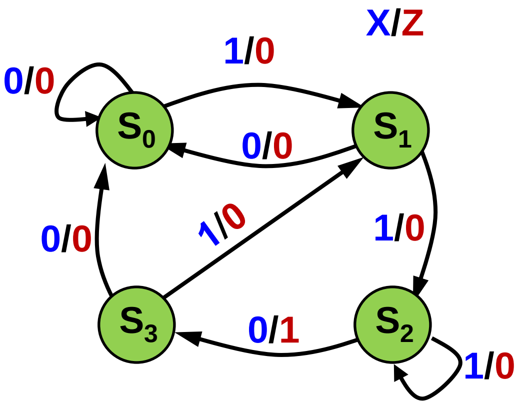
S_0: 0

S_1: 1

S_2: 11

S_3: 110, 且Z=1.

（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

利用触发器设计时序逻辑_状态编码

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

JK触发器驱动表

Q_n → Q_{n+1}	J	K
0 → 0	0	X
0 → 1	1	X
1 → 0	X	1
1 → 1	X	0

4. 状态转换真值表

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

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

利用触发器设计时序逻辑_状态编码

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

X \ $Y_2^nY_1^n$	00	01	11	10
0	0	X	X	0
1	0	X	X	1

$J_1 = XY_2^n$

X \ $Y_2^nY_1^n$	00	01	11	10
0	X	X	1	X
1	X	X	0	X

$K_1 = \overline{X}$

X \ $Y_2^nY_1^n$	00	01	11	10
0	0	X	1	0
1	0	X	0	0

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

5. 卡诺图化简

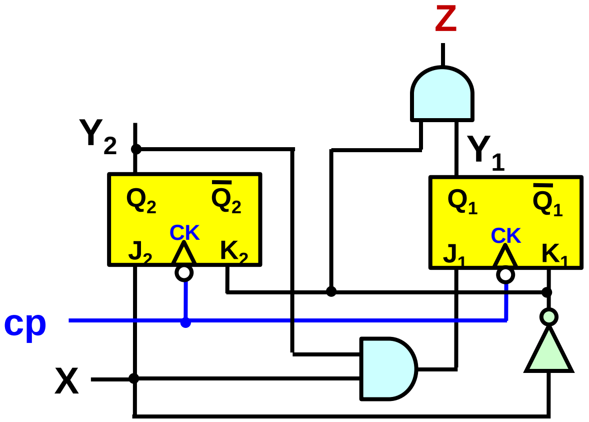
X \ $Y_2^nY_1^n$	00	01	11	10
0	0	X	X	X
1	1	X	X	X

$J_2 = X$

X \ $Y_2^nY_1^n$	00	01	11	10
0	X	X	1	1
1	X	X	0	0

$K_2 = \overline{X}$

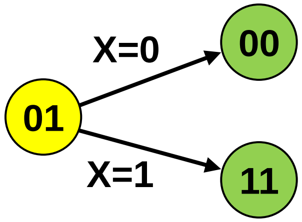
6. 电路实现



利用触发器设计时序逻辑_状态编码

7. 检查无关项

$$\begin{cases} J_1 = XY_2^n \\ K_1 = \overline{X} \\ J_2 = X \\ K_2 = \overline{X} \end{cases} \Rightarrow \begin{cases} Y_1^{n+1} = XY_2^n\overline{Y_1^n} + XY_1^n \\ \qquad = X(Y_1^n + Y_2^n) \\ Y_2^{n+1} = X\overline{Y_2^n} + XY_2^n \\ \qquad = X \end{cases}$$



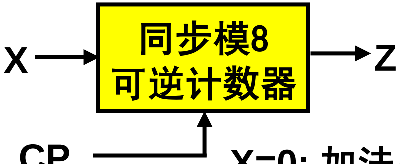
电路可以自启动

Some examples

- 模8可逆计数器
- 自动售卖机
- 时序锁
- 二进制串行加法器
- 串行输入的8421BCD码检测器
- 奇偶校验器
- 码制转换器
- 序列信号发生器

利用触发器设计同步时序逻辑_例1

例1：利用T触发器设计一个同步模8可逆计数器

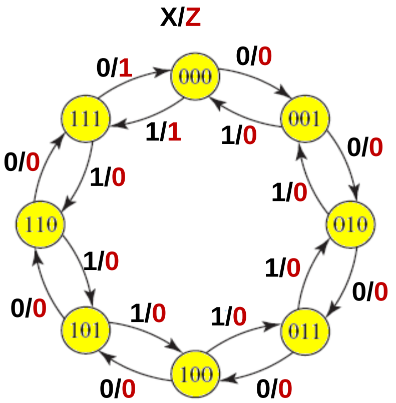


X=0: 加法；X=1: 减法

Z ： 进位及借位

1. 原始状态图及状态表

需要3个T触发器



T触发器驱动表

输入端T	次态 Q_{n+1}
0	Q_n
1	$\bar{Q}_n$

确定T_3：看$Q_3^n \to Q_3^{n+1}$
确定T_2：看$Q_2^n \to Q_2^{n+1}$
确定T_1：看$Q_1^n \to Q_1^{n+1}$

2. 状态转换真值表

输入	现态			次态			输入			输出
X	Q_3^n	Q_2^n	Q_1^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	T_3	T_2	T_1	Z
0	0	0	0	0	0	1	0	0	1	0
0	0	0	1	0	1	0	0	1	1	0
0	0	1	0	0	1	1	0	0	1	0
0	0	1	1	1	0	0	1	1	1	0
0	1	0	0	1	0	1	0	0	1	0
0	1	0	1	1	1	0	0	1	1	0
0	1	1	0	1	1	1	0	0	1	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	1	0
1	0	1	0	0	0	1	0	1	1	0
1	0	1	1	0	1	0	0	0	1	0
1	1	0	0	0	1	1	1	1	1	0
1	1	0	1	1	0	0	0	0	1	0
1	1	1	0	1	0	1	0	1	1	1
1	1	1	1	1	1	0	0	0	1	0

利用触发器设计同步时序逻辑_例1

3. 卡诺图化简

XQ_3^n \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	1	0	0	0
10	1	0	0	0

$$T_3 = \overline{X}Q_2^nQ_1^n + X\overline{Q_2^n}\,\overline{Q_1^n}$$

XQ_3^n \ $Q_2^nQ_1^n$	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	1

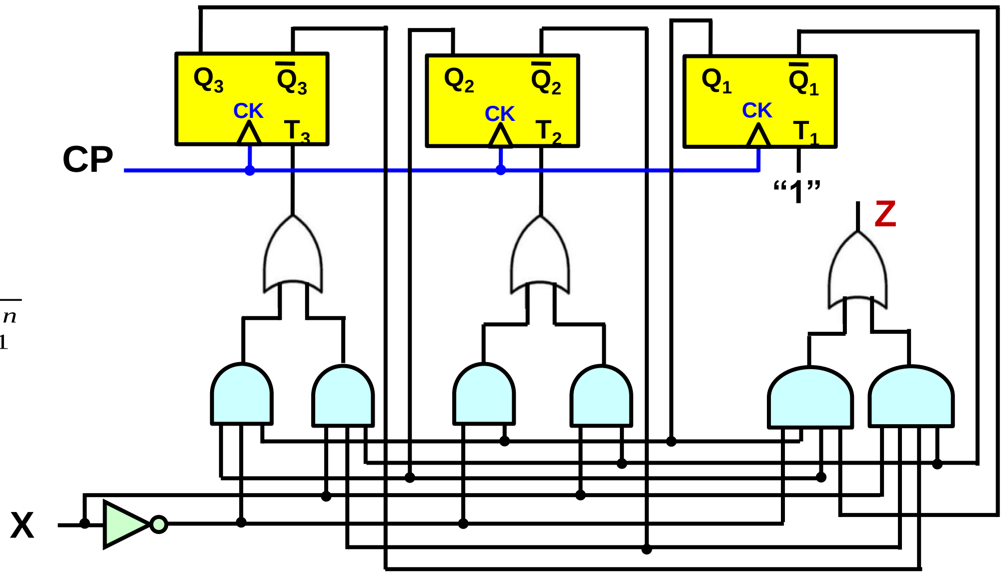
$$T_2 = \overline{X}Q_1^n + X\overline{Q_1^n}$$

XQ_3^n \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	0	0
10	1	0	0	0

$$T_1 = 1$$

$$Z = X\overline{Q_3^n}\,\overline{Q_2^n}\,\overline{Q_1^n} + \overline{X}Q_3^nQ_2^nQ_1^n$$

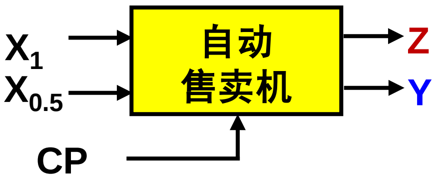
4. 电路实现



利用触发器设计同步时序逻辑_例2

例2：利用D触发器设计一个自动售卖机

- 只接收硬币：   0.5￥，1￥
- 每次投币只接收一枚硬币
- 机器收到1.5 ￥，给出一瓶饮料
- 机器收到2.0￥，给出一瓶饮料，找回0.5￥



$X_1 X_{0.5}$ ＝ 00: 0￥
$X_1 X_{0.5}$ ＝ 01: 0.5￥
$X_1 X_{0.5}$ ＝ 10: 1￥
Y=1/ 0: 给/不给 饮料
Z=1/ 0 :   找零/不找零

1. 原始状态图及状态表

① 状态设定

Solution 1:
Mealy circuit

S_0—初始状态，无投币
S_1—机器收到0.5￥
S_2—机器收到1.0￥ （2个 0.5￥, or 1个1.0￥ ）
  if （机器又收到1个0.5￥）
  then Y=1,且 Z=0, 回到 S_0
  Else If （机器又收到1个1￥）
  then Y=1, 且Z=1,回到S_0

利用触发器设计同步时序逻辑_例2

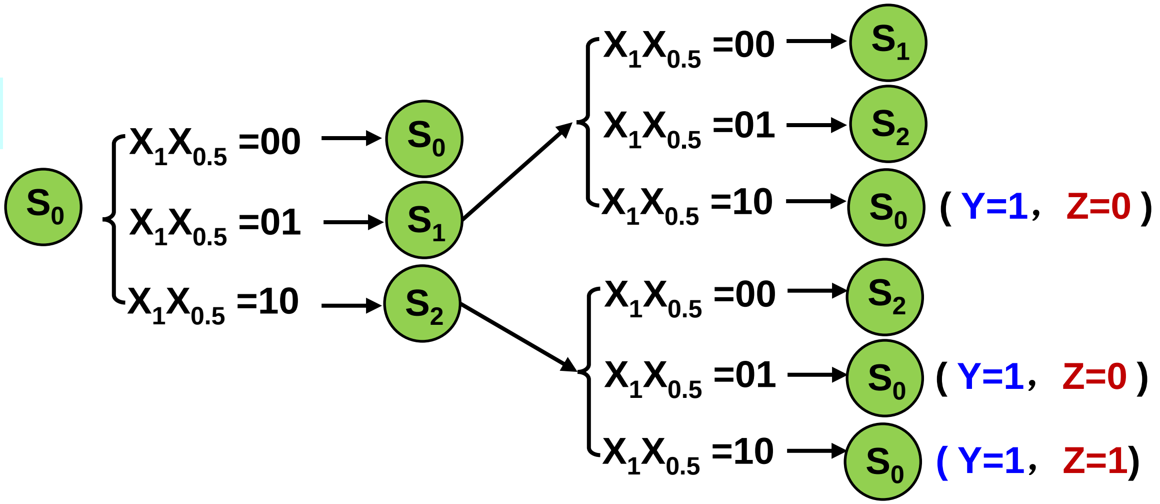
② 状态转换分析

Solution 1: Mealy circuit

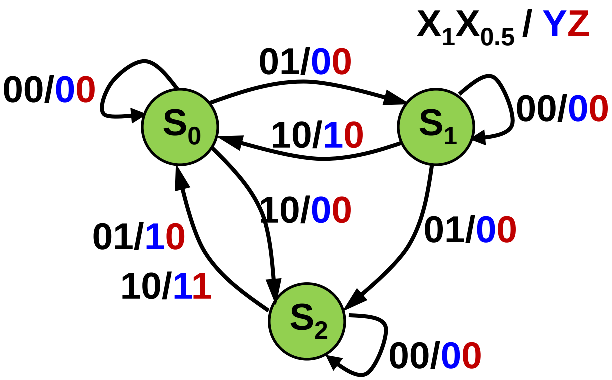
S_0—无投币

S_1—0.5￥

S_2—机器收到1.0￥



③ Mealy 状态图



④ 状态表

现态 S^n	S^{n+1}/ Z			
	$X_1X_{0.5}$=00	$X_1X_{0.5}$=01	$X_1X_{0.5}$=10	$X_1X_{0.5}$=11
S_0	S_0 / 00	S_1 / 00	S_2 / 00	X/ XX
S_1	S_1 / 00	S_2 / 00	S_0 / 10	X / XX
S_2	S_2 / 00	S_0 / 10	S_0 / 11	X / XX

利用触发器设计同步时序逻辑_例2

④ 状态表

现态 S^n	S^{n+1}/ Z			
	$X_1X_{0.5}$=00	$X_1X_{0.5}$=01	$X_1X_{0.5}$=10	$X_1X_{0.5}$=11
S_0	S_0 / 00	S_1 / 00	S_2 / 00	X/ XX
S_1	S_1 / 00	S_2 / 00	S_0 / 10	X / XX
S_2	S_2 / 00	S_0 / 10	S_0 / 11	X / XX

2. 状态化简

3. 状态分配

S_0 —— 00

S_1 —— 01

S_2 —— 10

	0	1
0	S_0	S_1
1	S_2	

需要2个D触发器

4. 状态转换真值

确定D_2：看Q_2^{n+1}
确定D_1：看Q_1^{n+1}

输入		现态		次态		输入		输出	
X_1	$X_{0.5}$	Q_2^n	Q_1^n	Q_2^{n+1}	Q_1^{n+1}	D_2	D_1	Y	Z
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	1	0	0
0	0	1	0	1	0	1	0	0	0
0	0	1	1	X	X	X	X	X	X
0	1	0	0	0	1	0	1	0	0
0	1	0	1	1	0	1	0	0	0
0	1	1	0	0	0	0	0	1	0
0	1	1	1	X	X	X	X	X	X
1	0	0	0	1	0	1	0	0	0
1	0	0	1	0	0	0	0	1	0
1	0	1	0	0	0	0	0	1	1
1	0	1	1	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X

利用触发器设计同步时序逻辑_例2

5. 卡诺图化简

$X_1X_{0.5}$ \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	1
01	0	1	X	0
11	X	X	X	X
10	1	0	X	0

$$D_2=\overline{X}_1\overline{X}_{0.5}Q_2^n+Q_1^n X_{0.5}+X_1\overline{Q}_1^n\overline{Q}_2^n$$

$X_1X_{0.5}$ \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	0
01	0	0	X	1
11	X	X	X	X
10	0	1	X	1

$$Y=Q_2^nX_{0.5}+Q_2^nX_1+X_1Q_1^n$$

$X_1X_{0.5}$ \ $Q_2^nQ_1^n$	00	01	11	10
00	0	1	X	0
01	1	0	X	0
11	X	X	X	X
10	0	0	X	0

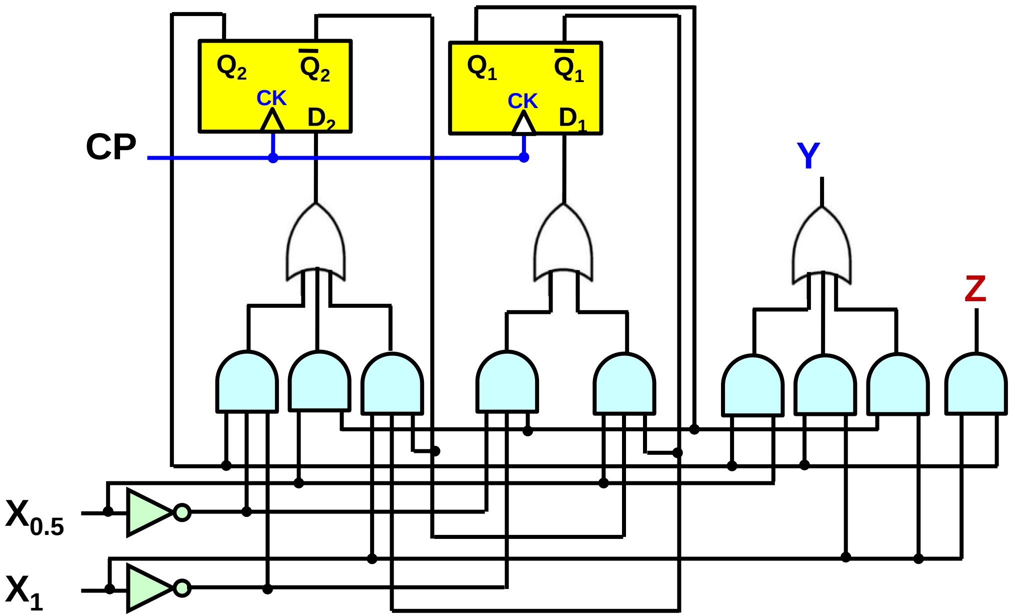
$$D_1=\overline{X}_1\overline{X}_{0.5}Q_1^n+X_{0.5}\overline{Q}_1^n\overline{Q}_2^n$$

$X_1X_{0.5}$ \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	0
01	0	0	X	0
11	X	X	X	X
10	0	0	X	1

$$Z=X_1Q_2^n$$

利用触发器设计同步时序逻辑_例2

6. 电路实现



! 电路需要预置

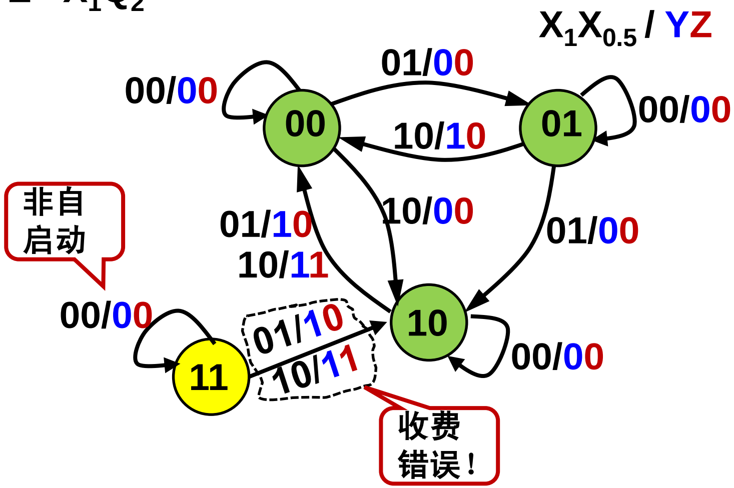
7. 检查无关项

无关状态：$Q_2^nQ_1^n=11$

$X_1X_{0.5}$ 分别为 00 ,01,10时，带入计算

$$\begin{cases} Q_2^{n+1}=D_2=\overline{X_1}\,\overline{X_{0.5}}Q_1^n+Q_1X_{0.5}+X_1\overline{Q_1^n}\,\overline{Q_2^n} \\ Q_1^{n+1}=D_1=\overline{X_1}\,\overline{X_{0.5}}Q_2^n+X_{0.5}\overline{Q_1^n}\,\overline{Q_2^n} \end{cases}$$

$$\begin{cases} Y=Q_2^nX_{0.5}+Q_2^nX_1+X_1Q_1^n \\ Z=X_1Q_2^n \end{cases}$$



利用触发器设计同步时序逻辑_例2

1. 原始状态图及状态表

① 状态设定（标记收到的钱数）

S_0—初始状态，机器收到0￥

S_1—机器收到0.5￥

S_2—机器收到1.0￥

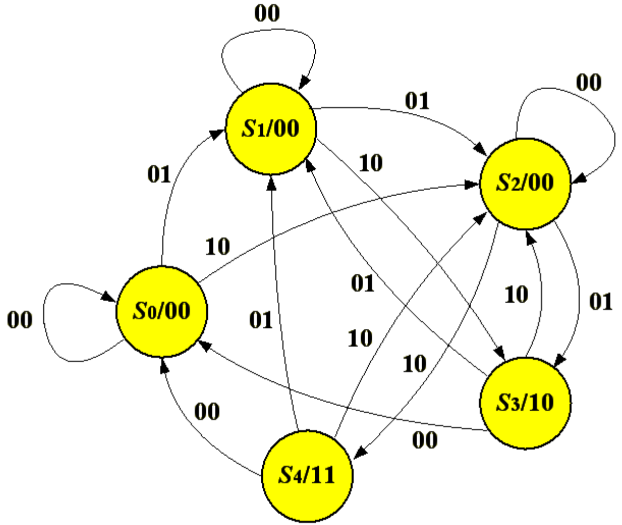
S_3—机器收到1.5￥

S_4—机器收到2.0￥

Solution 2:

Moor circuit

② Moor 状态图



③ Moor 状态表

现态 S_n	次态S_{n+1} $X_1X_2 = 00$	次态S_{n+1} $X_1X_2 = 01$	次态S_{n+1} $X_1X_2 = 10$	输出 YZ
S_0	S_0	S_1	S_2	00
S_1	S_1	S_2	S_3	00
S_2	S_2	S_3	S_4	00
S_3	S_0	S_1	S_2	10
S_4	S_0	S_1	S_2	11

2. 状态化简

3. 状态分配

需要3个D触发器

Q_3^n \ $Q_2^nQ_1^n$	00	01	11	10
0	S_0	S_3		S_1
1	S_4			S_2

S_0 —— 000

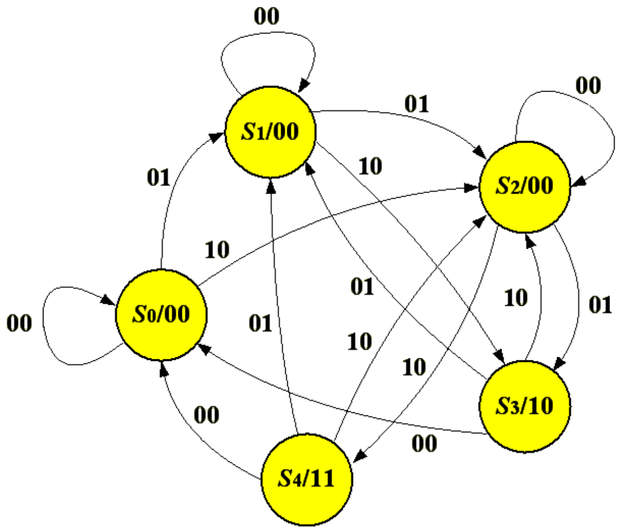
S_1 —— 010

S_2 —— 110

S_3 —— 001

S_4 —— 100

利用触发器设计同步时序逻辑_例2



S_0 —— 000

S_1 —— 010

S_2 —— 110

S_3 —— 001

S_4 —— 100

4. 状态转换真值表

输入		现态			次态			输入			输出	
X_1	$X_{0.5}$	Q_3^n	Q_2^n	Q_1^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	D_3	D_2	D_1	Y	Z
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	1	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	1	1	0	1	1	0	1	1	0	0	0
0	1	1	1	0	0	0	1	0	0	1	0	0
0	1	0	0	0	0	1	0	0	1	0	0	0
0	1	0	1	0	1	1	0	1	1	0	0	0
0	1	0	0	1	0	1	0	0	1	0	1	0
0	1	1	0	0	0	1	0	0	1	0	1	1
1	0	0	0	0	1	1	0	1	1	0	0	0
1	0	0	1	0	0	0	1	0	0	1	0	0
1	0	1	1	0	1	0	0	1	0	0	0	0
1	0	0	0	1	1	1	0	1	1	0	1	0
1	0	1	0	0	1	1	0	1	1	0	1	1
1	1	X	X	X	X	X	X	X	X	X	X	X

利用触发器设计同步时序逻辑_例2

5. 卡诺图化简

$X_1=0$

$X_{0.5}Q_3^n$ \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	0
01	0	X	X	1
11	0	X	X	0
10	0	0	X	1

$X_1=1$

$X_{0.5}Q_3^n$ \ $Q_2^nQ_1^n$	00	01	11	10
00	1	1	X	0
01	1	X	X	1
11	X	X	X	X
10	X	X	X	X

$X_1=0$

$X_{0.5}Q_3^n$ \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	1
01	0	X	X	1
11	1	X	X	0
10	1	1	X	1

$X_1=1$

$X_{0.5}Q_3^n$ \ $Q_2^nQ_1^n$	00	01	11	10
00	1	1	X	0
01	1	X	X	0
11	X	X	X	X
10	X	X	X	X

$X_1=0$

$X_{0.5}Q_3^n$ \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	0
01	0	X	X	0
11	0	X	X	1
10	0	0	X	0

$X_1=1$

$X_{0.5}Q_3^n$ \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	1
01	0	X	X	0
11	X	X	X	X
10	X	X	X	X

$$D_3=\overline{X}_{0.5}Q_3^nQ_2^n+\overline{Q}_3^n X_{0.5}Q_2^n+X_1\overline{Q}_2^n$$

$$D_2=\overline{X}_{0.5}Q_3^n+\overline{Q}_2^n X_{0.5}+X_1\overline{Q}_2^n+\overline{X}_1\overline{X}_{0.5}Q_2^n$$

$$D_1=X_{0.5}Q_3^nQ_2^n+\overline{Q}_3^n X_1Q_2^n$$

利用触发器设计同步时序逻辑_例2

$X_1=0$

$X_{0.5}Q_3^n$ \ $Q_2^nQ_1^n$	00	01	11	10
00	0	1	X	0
01	1	X	X	0
11	1	X	X	0
10	0	1	X	1

$X_1=1$

$X_{0.5}Q_3^n$ \ $Q_2^nQ_1^n$	00	01	11	10
00	0	1	X	0
01	1	X	X	0
11	X	X	X	X
10	X	X	X	X

$$Y = \overline{Q_2}^nQ_3^n + Q_1^n$$

$X_1=0$

$X_{0.5}Q_3^n$ \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	0
01	1	X	X	0
11	1	X	X	0
10	0	0	X	0

$X_1=1$

$X_{0.5}Q_3^n$ \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	0
01	1	X	X	0
11	X	X	X	X
10	X	X	X	X

$$Z = \overline{Q_2}^nQ_3^n$$

$$\begin{cases} D_3 = \overline{X}_{0.5}Q_3^nQ_2^n+\overline{Q_3}^n X_{0.5}Q_2^n+ X_1\overline{Q_2}^n \\ D_2 = \overline{X}_{0.5}Q_3^n+\overline{Q_2}^n X_{0.5}+ X_1\overline{Q_2}^n +\overline{X}_1\overline{X}_{0.5}Q_2^n \\ D_1 = X_{0.5}Q_3^nQ_2^n+\overline{Q_3}^n X_1Q_2^n \\ Y = \overline{Q_2}^nQ_3^n + Q_1^n \\ Z = \overline{Q_2}^nQ_3^n \end{cases}$$

6. 电路实现(略)

7. 检查无关项（略）

Moor型电路与Mealy型电路比较

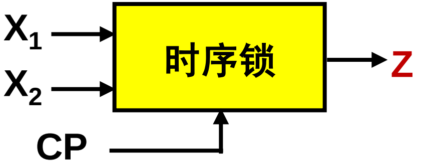
- Moor型电路中的状态总数相对要多一些，需要使用较多的触发器资源。
- Moor型电路的输出只与状态有关，输出没有毛刺。

利用触发器设计时序逻辑

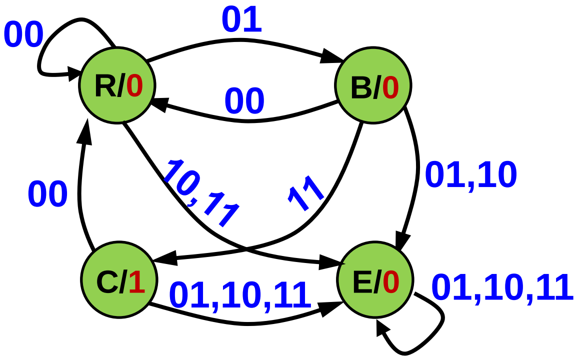
- 模8可逆计数器
- 自动售卖机
- 时序锁
- 二进制串行加法器
- 串行输入的8421BCD码检测器
- 奇偶校验器
- 码制转换器
- 序列信号发生器

利用触发器设计同步时序逻辑_例3

例3：利用JK触发器设计一个时序锁



- 输入: X_1X_2，输出: Z
- 该锁内部有四个状态R、B、C、E
- 依次输入00、01、11，时序锁从状态 R→B→C，并开锁（Z=1）
- 不是上述序列，进入状态 E（error）
- 任何时候只要输入00，都将返回状态 R



摩尔型

1. 原始状态图及状态表

① 状态设定

R—初始状态，输入00

B—输入00后，再输入01

C—输入00、01后，再输入11，且Z=1

E—错误状态

现态 S_n	次态S_{n+1}				输出 Z
	$X_1X_2 = 00$	$X_1X_2 = 01$	$X_1X_2 = 11$	$X_1X_2 = 10$	
R	R	B	E	E	0
B	R	E	C	E	0
C	R	E	E	E	1
E	R	E	E	E	0

利用触发器设计同步时序逻辑_例3

现态 S_n	次态 S_{n+1}				输出 Z
	X_1X_2 = 00	X_1X_2 = 01	X_1X_2 = 11	X_1X_2 = 10	
R	*R*	*B*	*E*	*E*	0
B	*R*	*E*	*C*	*E*	0
C	*R*	*E*	*E*	*E*	1
E	*R*	*E*	*E*	*E*	0

2. 状态化简

3. 状态分配

需要2个JK触发器

R: 00, B : 01

E : 10, C: 11

	0	1
0	R	B
1	E	C

4. 状态转换真值表

$J_2\,K_2$： 看$Q_2^n \rightarrow Q_2^{n+1}$

$J_1\,K_1$： 看$Q_1^n \rightarrow Q_1^{n+1}$

输入		现态		次态		输入				输出
X_1	X_2	Q_2^n	Q_1^n	Q_2^{n+1}	Q_1^{n+1}	J_2	K_2	J_1	K_1	Z
0	0	0	0	0	0	0	X	0	X	0
0	0	0	1	0	0	0	X	X	1	0
0	0	1	0	0	0	X	1	0	X	0
0	0	1	1	0	0	X	1	X	1	1
0	1	0	0	0	1	0	X	1	X	0
0	1	0	1	1	0	1	X	X	1	0
0	1	1	0	1	0	X	0	0	X	0
0	1	1	1	1	0	X	0	X	1	1
1	0	0	0	1	0	1	X	0	X	0
1	0	0	1	1	0	1	X	X	1	0
1	0	1	0	1	0	X	0	0	X	0
1	0	1	1	1	0	X	0	X	1	1
1	1	0	0	1	0	1	X	0	X	0
1	1	0	1	1	1	1	X	X	0	0
1	1	1	0	1	0	X	0	0	X	0
1	1	1	1	1	0	X	0	X	1	1

利用触发器设计同步时序逻辑_例3

5. 卡诺图化简

X_1X_2 \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	X
01	0	1	X	X
11	1	1	X	X
10	1	1	X	X

$$J_2 = X_2Q_1^n + X_1$$

X_1X_2 \ $Q_2^nQ_1^n$	00	01	11	10
00	X	X	1	1
01	X	X	0	0
11	X	X	0	0
10	X	X	0	0

$$K_2 = \overline{X}_2\,\overline{X}_1$$

X_1X_2 \ $Q_2^nQ_1^n$	00	01	11	10
00	0	X	X	0
01	1	X	X	0
11	0	X	X	0
10	0	X	X	0

$$J_1 = \overline{X}_1X_2\overline{Q}_2^n$$

X_1X_2 \ $Q_2^nQ_1^n$	00	01	11	10
00	X	1	1	X
01	X	1	1	X
11	X	0	1	X
10	X	1	1	X

$$K_1 = Q_2^n + \overline{X}_2 + \overline{X}_1$$

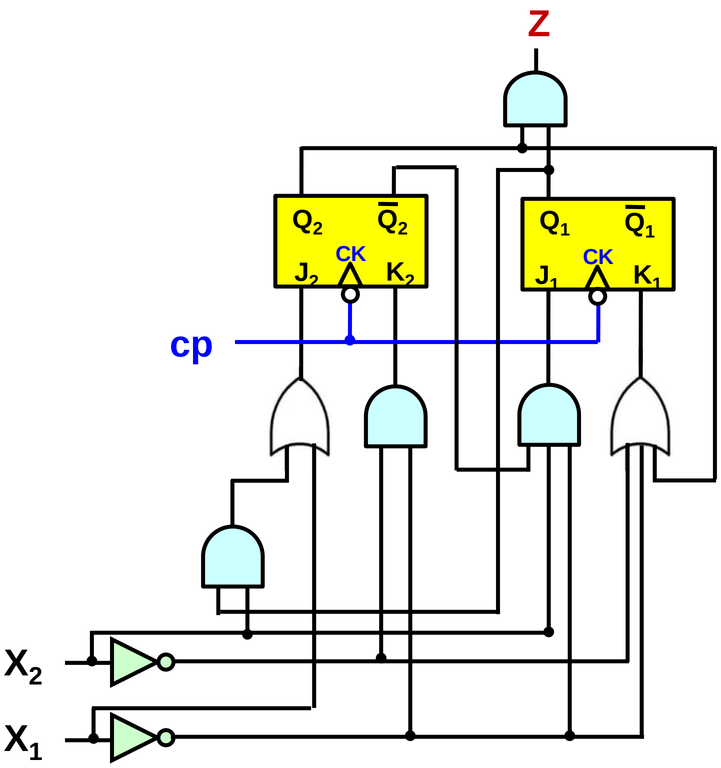
X_1Q_2 \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	0	0	1	0
10	0	0	1	0

$$Z = Q_2^nQ_1^n$$

利用触发器设计同步时序逻辑_例3

6. 电路实现

$$\begin{cases} J_2 = X_2Q_1^n + X_1 \\ K_2 = \overline{X}_2\overline{X}_1 \\ J_1 = \overline{X}_1X_2\overline{Q}_2^n \\ K_1 = Q_2^n + \overline{X}_2 + \overline{X}_1 \\ Z = Q_2^nQ_1^n \end{cases}$$



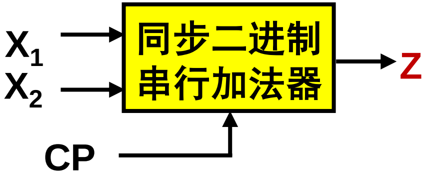
密码锁

- ■一维开锁：密码正确
- ■二维开锁：有限时间+密码正确
- ■三维开锁：

有限时间+有限按键次数+密码正确

利用触发器设计同步时序逻辑_例3

例4：利用JK触发器设计一个同步二进制串行加法器



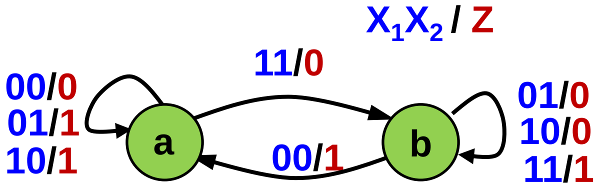
1. 原始状态图及状态表

① 设加法器内部状态

a—— 无进位

b—— 有进位

② Mealy 状态图



③ Mealy 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	a / 0	a / 1	a / 1	b / 0
b	a / 1	b / 0	b / 0	b / 1

2. 状态化简　　3. 状态分配　a=0, b=1

4. 状态转换真值表

输入 现态			次态	输入		输出
X_1	X_2	Q^n	Q^{n+1}	J	K	Z
0	0	0	0	0	X	0
0	0	1	0	X	1	1
0	1	0	0	0	X	1
0	1	1	1	X	0	0
1	0	0	0	0	X	1
1	0	1	1	X	0	0
1	1	0	1	1	X	0
1	1	1	1	X	0	1

利用触发器设计同步时序逻辑_例3

5. 卡诺图化简

X_1 \ X_2Q^n	00	01	11	10
0	0	X	X	0
1	0	X	X	1

$$J = X_1 X_2$$

X_1 \ X_2Q^n	00	01	11	10
0	X	1	0	X
1	X	0	0	X

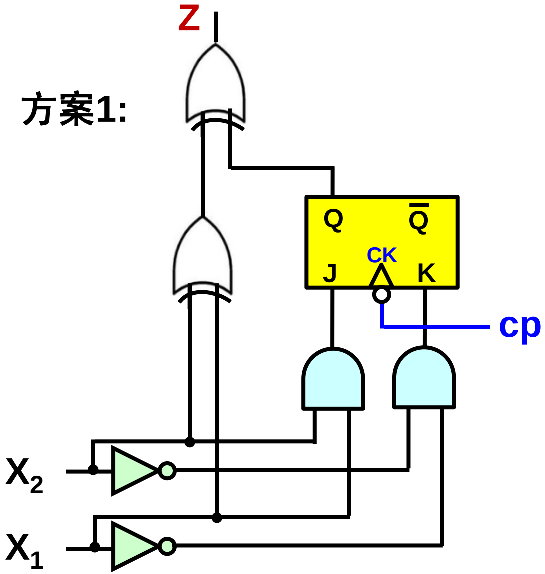
$$K = \overline{X}_1 \overline{X}_2$$

X_1 \ X_2Q^n	00	01	11	10
0	0	1	0	1
1	1	0	1	0

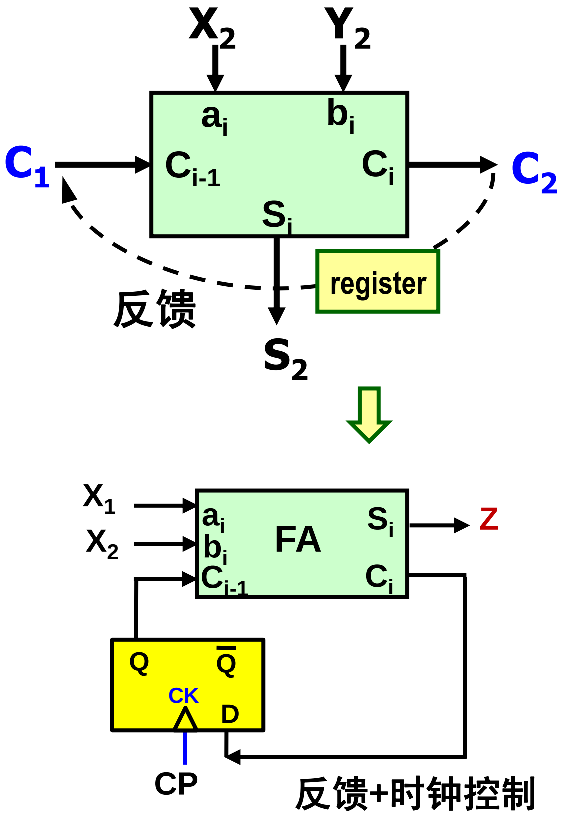
$$Z = X_1 \oplus X_2 \oplus Q^n$$

6. 电路实现

方案1:



方案2: 如何用一位全加器实现?



利用触发器设计时序逻辑

- 模8可逆计数器
- 自动售卖机
- 时序锁
- 二进制串行加法器
- 串行输入的8421BCD码检测器
- 奇偶校验器
- 码制转换器
- 序列信号发生器

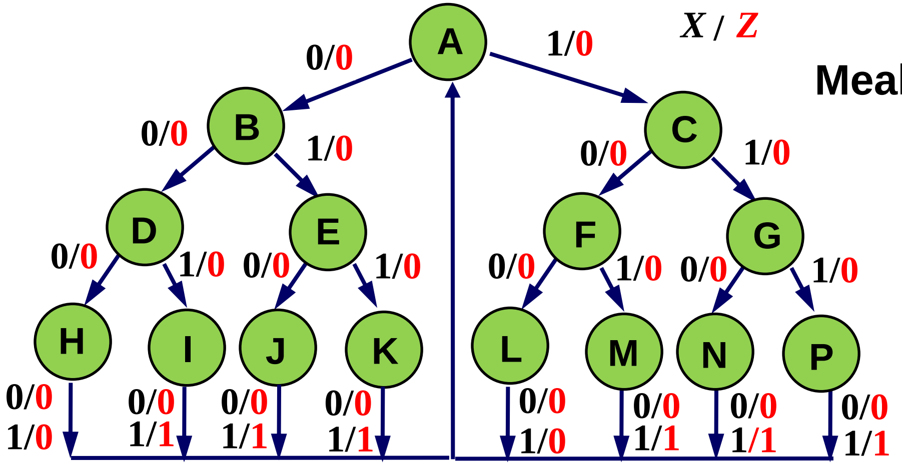
利用触发器设计同步时序逻辑_例5

例5：用D触发器设计一个串行输入的8421BCD码误码检测器

要求：

- 8421BCD码低位在前、高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。

1. 原始状态图及状态表



Mealy 状态图

利用触发器设计同步时序逻辑_例5

2. 状态化简

现态 Q^n	Q^{n+1}/ Z	
	X=0	X=1
A	B / 0	C / 0
B	D / 0	E / 0
C	F / 0	G / 0
D	H / 0	I / 0
E	J / 0	K / 0
F	L / 0	M / 0
G	N / 0	P / 0
~~H~~	~~A / 0~~	~~A / 0~~
~~I~~	~~A / 0~~	~~A / 1~~
~~J~~	~~A / 0~~	~~A / 1~~
~~K~~	~~A / 0~~	~~A / 1~~
~~L~~	~~A / 0~~	~~A / 0~~
~~M~~	~~A / 0~~	~~A / 1~~
~~N~~	~~A / 0~~	~~A / 1~~
~~P~~	~~A / 0~~	~~A / 1~~

⇒

现态 Q^n	Q^{n+1}/ Z	
	X=0	X=1
A	B / 0	C / 0
B	D / 0	E / 0
C	F / 0	G / 0
~~D~~	~~H / 0~~	~~I / 0~~
~~E~~	~~I / 0~~	~~I / 0~~
~~F~~	~~H / 0~~	~~I / 0~~
~~G~~	~~I / 0~~	~~I / 0~~
H	A / 0	A / 0
I	A / 0	A / 1

⇒

现态 Q^n	Q^{n+1}/ Z	
	X=0	X=1
A	B / 0	C / 0
B	D / 0	E / 0
C	D / 0	E / 0
D	H / 0	I / 0
E	I / 0	I / 0
H	A / 0	A / 0
I	A / 0	A / 1

利用触发器设计同步时序逻辑_例5

2. 状态化简

现态 Q^n	Q^{n+1}/ Z	
	X=0	X=1
A	B / 0	C / 0
~~B~~	~~D / 0~~	~~E / 0~~
~~C~~	~~D / 0~~	~~E / 0~~
D	H / 0	I / 0
E	I / 0	I / 0
H	A / 0	A / 0
I	A / 0	A / 1

⇨

现态 Q^n	Q^{n+1}/ Z	
	X=0	X=1
A	B / 0	B / 0
B	D / 0	E / 0
D	H / 0	I / 0
E	I / 0	I / 0
H	A / 0	A / 0
I	A / 0	A / 1

3. 状态分配

规则1：次态相同，现态编码应相邻

HI, DE 应相邻

规则2：同一现态对应的次态应给予相邻编码

DE, HI 应相邻

规则3：输出相同，现态编码应相邻

ABDEH应相邻

⇨

Q_3^n \ $Q_2^nQ_1^n$	00	01	11	10
0	A	B	D	I
1			E	H

A: 000;　B: 001

D: 011;　I: 010

E: 111;　H: 110

Q_3^n \ $Q_2^nQ_1^n$	00	01	11	10
0	A	B	D	I
1			E	H

现态 Q^n	Q^{n+1}/ Z	
	X=0	X=1
A	B / 0	B / 0
B	D / 0	E / 0
D	H / 0	I / 0
E	I / 0	I / 0
H	A / 0	A / 0
I	A / 0	A / 1

4. 状态转换真值表

确定D_3：看Q_3^{n+1}
确定D_2：看Q_2^{n+1}
确定D_1：看Q_1^{n+1}

输入及现态				次态			输入			输出
X	Q_3^n	Q_2^n	Q_1^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	D_3	D_2	D_1	Z
0	0	0	0	0	0	1	0	0	1	0
0	0	0	1	0	1	1	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	1	1	0	0
0	1	0	0	X	X	X	X	X	X	X
0	1	0	1	X	X	X	X	X	X	X
0	1	1	0	0	0	0	0	0	0	0
0	1	1	1	0	1	0	0	1	0	0
1	0	0	0	0	0	1	0	0	1	0
1	0	0	1	1	1	1	1	1	1	0
1	0	1	0	0	0	0	0	0	0	1
1	0	1	1	0	1	0	0	1	0	0
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	0	1	0	0	1	0	0

利用触发器设计同步时序逻辑_例5

5. 卡诺图化简

XQ_3^n \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	1	0
01	X	X	0	0
11	X	X	0	0
10	0	1	0	0

$$D_3=\overline{Q_3^n}Q_2^nQ_1^n\overline{X}+X\overline{Q_2^n}Q_1^n$$

XQ_3^n \ $Q_2^nQ_1^n$	00	01	11	10
00	0	1	1	0
01	X	X	1	0
11	x	x	1	0
10	0	1	1	0

$$D_2=Q_1^n$$

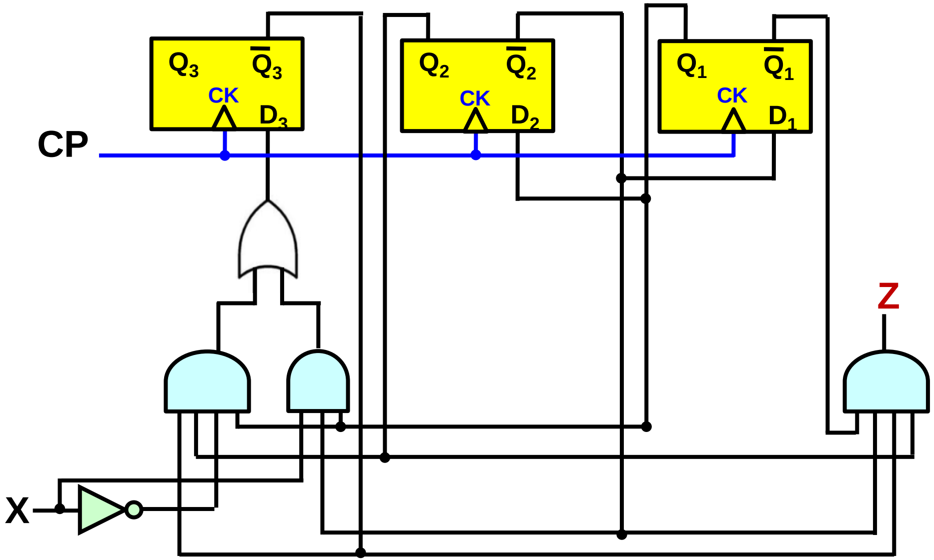
XQ_3^n \ $Q_2^nQ_1^n$	00	01	11	10
00	1	1	0	0
01	x	x	0	0
11	x	x	0	0
10	1	1	0	0

$$D_1=\overline{Q_2^n}$$

XQ_3^n \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	0	0
01	x	x	0	0
11	x	x	0	0
10	0	0	0	1

$$Z=X\overline{Q_3^n}Q_2^n\overline{Q_1^n}$$

6. 电路实现

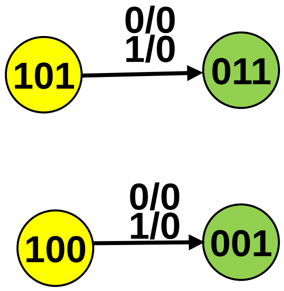


利用触发器设计同步时序逻辑_例5

7. 无关项检查

Q_3^n \ $Q_2^nQ_1^n$	00	01	11	10
0	A	B	D	I
1			E	H

将无关状态$Q_3^nQ_2^nQ_1^n$=100和101分别代入次态方程和输出方程计算

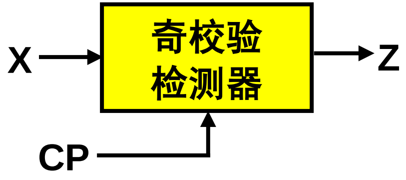


$$
\begin{cases}
Q_i^{n+1} = D_i \\
D_3 = \overline{Q_3^n} Q_2^n Q_1^n \overline{X} + X \overline{Q_2^n} Q_1^n \\
D_2 = Q_1^n \\
D_1 = \overline{Q_2^n} \\
Z = X \overline{Q_3^n} Q_2^n \overline{Q_1^n}
\end{cases}
$$

电路可以自启动

利用触发器设计同步时序逻辑_例6

例6：利用T触发器设计一个串行输入的奇校验检测器



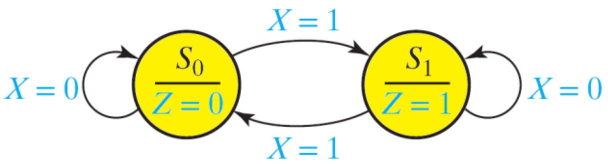
1. 原始状态图及状态表

① 状态设定

S_0——表示收到偶数个“1”，初始为0个“1”

S_1——表示收到奇数个“1”

② Moor 状态图



③状态表

现态 Q^n	次态Q^{n+1} X=0	次态Q^{n+1} X=1	输出 Z
S_0	S_0	S_1	0
S_1	S_1	S_0	1

2. 状态化简

3. 状态分配

S_0: 0；S_1: 1

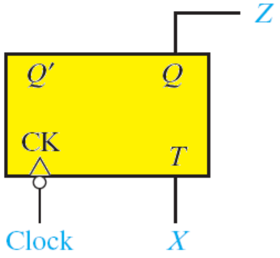
4. 状态转换真值表

输入 X	现态 Q^n	次态 Q^{n+1}	输入 T	输出 Z
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	0	1	1

5. 卡诺图化简

$T=X$; $Z=Q^n$

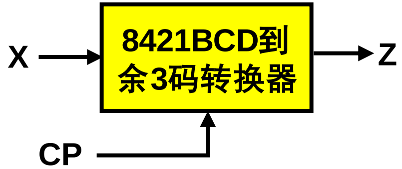
6. 电路实现



更复杂的同步时序设计_例7

例7：利用D触发器设计一个同步时序的码制转换器，将串行输入的8421BCD码转换为余3码。

- 转换器的输入和输出都是最低位优先



X Input (BCD)				Z Output (excess-3)			
t_3	t_2	t_1	t_0	t_3	t_2	t_1	t_0
	0	0	0		0	1	1
	0	0	1		1	0	0
	0	1	0		1	0	1
	0	1	1		1	1	0
	1	0	0		1	1	1
	1	0	1		0	0	0
	1	1	0		0	0	1
	1	1	1		0	1	0
	0	0	0		0	1	1
	0	0	1		1	0	0

- t_0时刻：

  输入为0，输出为1

  输入为1，输出为0

- t_1~t_3时刻：

  单纯看没有规律，要联合前一时刻的输入一同来看

更复杂的同步时序设计_例7

- t_0时刻：输入为0，输出为1；输入为1，输出为0
- t_1~t_3时刻：单纯看没有规律，要联合前一时刻的输入一同来看

$t_1\,t_0$时刻输入	$t_1\,t_0$时刻输出
00	11
01	00
10	01
11	10

$t_2\,t_1\,t_0$时刻输入	$t_2\,t_1\,t_0$时刻输出
000	011
001	100
010	101
011	110
100	111
101	000
110	001
111	010

$t_3\,t_2\,t_1\,t_0$时刻输入	$t_3\,t_2\,t_1\,t_0$时刻输出
0000	0011
0001	0100
0010	0101
0011	0110
0100	0111
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100

X Input (BCD)				Z Output (excess-3)			
t_3	t_2	t_1	t_0	t_3	t_2	t_1	t_0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

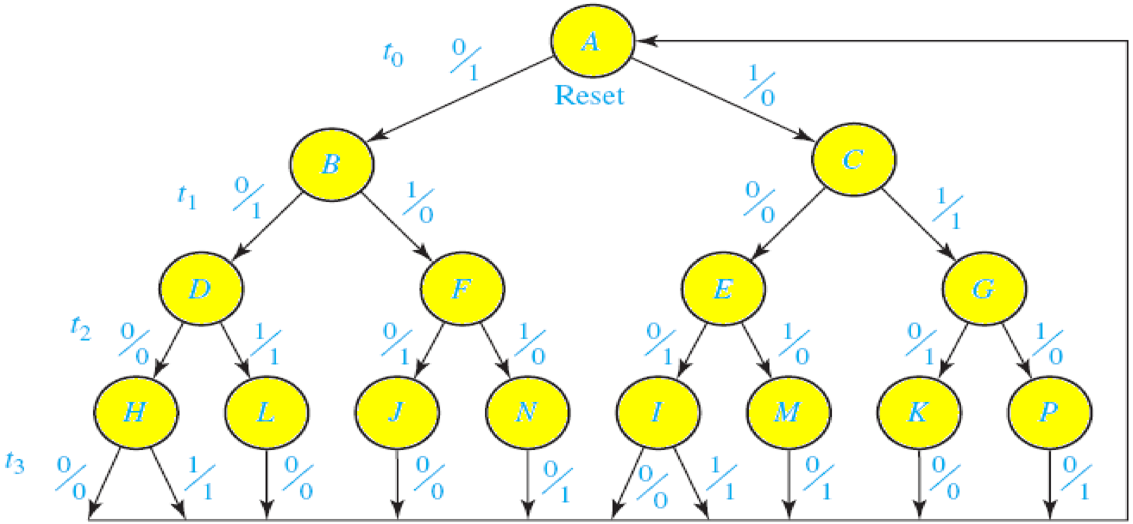
更复杂的同步时序设计_例7

1. 原始状态图及状态表

- t_0时刻：输入为0，输出为1；输入为1，输出为0
- t_1~t_3时刻：单纯看没有规律，要联合前一时刻的输入一同来看

$t_1 t_0$时刻输入	$t_1 t_0$时刻输出
00	11
01	00
10	01
11	10

$t_2 t_1 t_0$时刻输入	$t_2 t_1 t_0$时刻输出
000	011
001	100
010	101
011	110
100	111
101	000
110	001
111	010



$t_3 t_2 t_1 t_0$时刻输入	$t_3 t_2 t_1 t_0$时刻输出
0000	0011
0001	0100
0010	0101
0011	0110
0100	0111
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100

更复杂的同步时序设计_例7

2. 状态化简

Time	Input Sequence Received (Least Significant Bit First)	Present State	Next State $X=0$	Next State $X=1$	Present Output (Z) $X=0$	Present Output (Z) $X=1$
t_0	reset	A	B	C	1	0
t_1	0	B	D	F	1	0
	1	C	E	G	0	1
t_2	00	D	H	L	0	1
	01	E	I	M	1	0
	10	F	J	N	1	0
	11	G	K	P	1	0
t_3	000	H	A	A	0	1
	001	I	A	A	0	1
	010	J	A	–	0	–
	011	K	A	–	0	–
	100	L	A	–	0	–
	101	M	A	–	1	–
	110	N	A	–	1	–
	111	P	A	–	1	–

Time	Present State	Next State $X=0$	Next State $X=1$	Present Output (Z) $X=0$	Present Output (Z) $X=1$
t_0	A	B	C	1	0
t_1	B	D	E	1	0
	C	E	E	0	1
t_2	D	H	H	0	1
	E	H	M	1	0
t_3	H	A	A	0	1
	M	A	–	1	–

更复杂的同步时序设计_例7

3. 状态分配

Time	Present State	Next State $X=0$	Next State $X=1$	Present Output (Z) $X=0$	Present Output (Z) $X=1$
t_0	A	B	C	1	0
t_1	B	D	E	1	0
	C	E	E	0	1
t_2	D	H	H	0	1
	E	H	M	1	0
t_3	H	A	A	0	1
	M	A	–	1	–

Q_3 \ Q_2Q_1	00	01	11	10
0	A	B	E	M
1		C	D	H

4. 状态转换真值表

	$Q_1Q_2Q_3$	$Q_1^+Q_2^+Q_3^+$ $X=0$	$Q_1^+Q_2^+Q_3^+$ $X=1$	Z $X=0$	Z $X=1$
A	0 0 0	1 0 0	1 0 1	1	0
B	1 0 0	1 1 1	1 1 0	1	0
C	1 0 1	1 1 0	1 1 0	0	1
D	1 1 1	0 1 1	0 1 1	0	1
E	1 1 0	0 1 1	0 1 0	1	0
H	0 1 1	0 0 0	0 0 0	0	1
M	0 1 0	0 0 0	x x x	1	x
–	0 0 1	x x x	x x x	x	x

更复杂的同步时序设计_例7

4. 状态转换真值表

	$Q_1Q_2Q_3$	$Q_1^+Q_2^+Q_3^+$		Z	
		$X=0$	$X=1$	$X=0$	$X=1$
A	0 0 0	1 0 0	1 0 1	1	0
B	1 0 0	1 1 1	1 1 0	1	0
C	1 0 1	1 1 0	1 1 0	0	1
D	1 1 1	0 1 1	0 1 1	0	1
E	1 1 0	0 1 1	0 1 0	1	0
H	0 1 1	0 0 0	0 0 0	0	1
M	0 1 0	0 0 0	x x x	1	x
–	0 0 1	x x x	x x x	x	x

5. 卡诺图化简

$Q_2Q_3 \backslash XQ_1$	00	01	11	10
00	1	1	1	1
01	X	1	1	X
11	0	0	0	0
10	0	0	0	X

$D_1 = Q_1^+ = Q_2'$

$Q_2Q_3 \backslash XQ_1$	00	01	11	10
00	0	1	1	0
01	X	1	1	X
11	0	1	1	0
10	0	1	1	X

$D_2 = Q_2^+ = Q_1$

$Q_2Q_3 \backslash XQ_1$	00	01	11	10
00	0	1	0	1
01	X	0	0	X
11	0	1	1	0
10	0	1	0	X

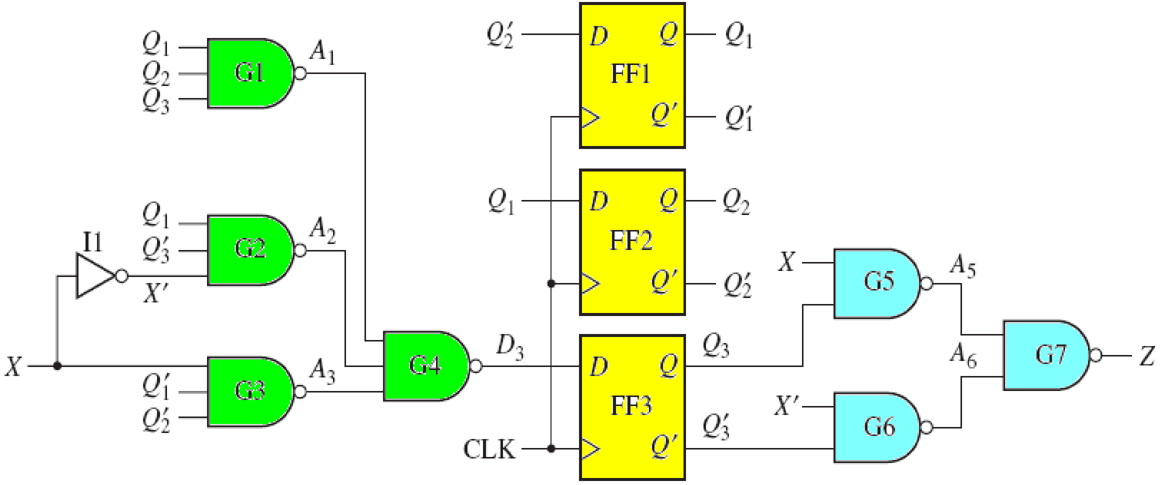
$D_3 = Q_3^+ = Q_1Q_2Q_3 + X'Q_1Q_3' + XQ_1'Q_2'$

$Q_2Q_3 \backslash XQ_1$	00	01	11	10
00	1	1	0	0
01	X	0	1	X
11	0	0	1	1
10	1	1	0	X

$Z = X'Q_3' + XQ_3$

更复杂的同步时序设计_例7

6. 电路实现

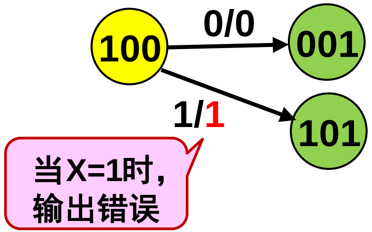


Q_2Q_3 \ Q_1	0	1
00	A	B
01		C
11	H	D
10	M	E

7. 无关项检查

$$\begin{cases} D_1 = Q_1^+ = Q_2' \\ D_2 = Q_2^+ = Q_1 \\ D_3 = Q_3^+ = Q_1Q_2Q_3 + X'Q_1Q_3' + XQ_1'Q_2' \\ Z = X'Q_3' + XQ_3 \end{cases}$$

将无关状态$Q_3Q_2Q_1$=100代入次态方程和输出方程计算

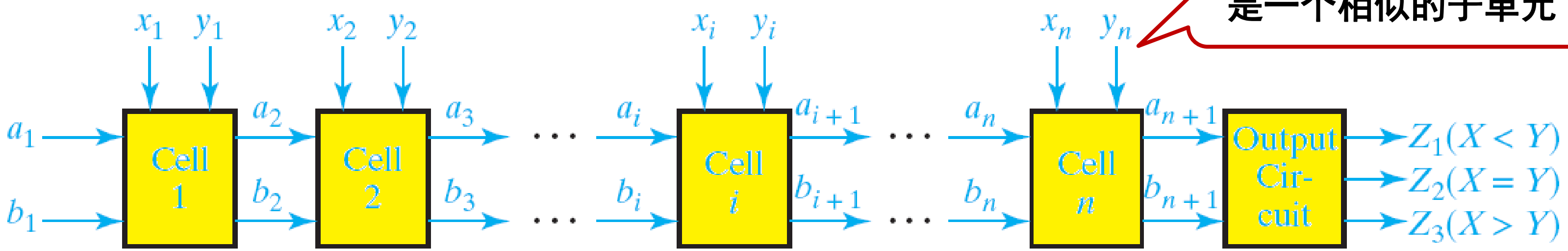


电路可以自启动

更复杂的同步时序设计_例8

例8：迭代电路设计——利用D触发器设计一个比较器，能对两个n位二进制数进行比较。

每个对应位的比较都是一个相似的子单元



1. 原始状态图及状态表

- 由n个比较子单元（cell）构成
- 从高位到低位，逐位对应比较，并将前一位比较的结果传送给下一位
- 第i个单元的比较结果： $X = Y$, $X > Y$, or $X < Y$.

对于第 i 个单元，设状态——

S_0 : $X = Y$ 时

S_1 : $X > Y$ 时

S_2 : $X < Y$ 时

Z_2、Z_3、Z_1 分别取值为1

更复杂的同步时序设计_例8

1. 原始状态图及状态表

	S_i	S_{i+1}: $x_iy_i=00$	01	11	10	$Z_1Z_2Z_3$
$X=Y$	S_0	S_0	S_2	S_0	S_1	0 1 0
$X>Y$	S_1	S_1	S_1	S_1	S_1	0 0 1
$X<Y$	S_2	S_2	S_2	S_2	S_2	1 0 0

在第i个（前一个）单元有比较结果的前提下，根据输入取值，可以确定第$i+1$个单元的比较结果

对于第 i 个单元，设状态——
S_0 : $X=Y$ 时
S_1 : $X>Y$ 时
S_2 : $X<Y$ 时
Z_1 、Z_2、 Z_3 分别取值为1

2. 状态化简

3. 状态分配

S_0 : 00

S_1 : 01

S_2 : 10

需要两个触发器，用 a,b来表示

4. 状态转换真值表

a_ib_i	$a_{i+1}b_{i+1}$: $x_iy_i=00$	01	11	10	$Z_1Z_2Z_3$
0 0	00	10	00	01	0 1 0
0 1	01	01	01	01	0 0 1
1 0	10	10	10	10	1 0 0

更复杂的同步时序设计_例8

5. 卡诺图化简

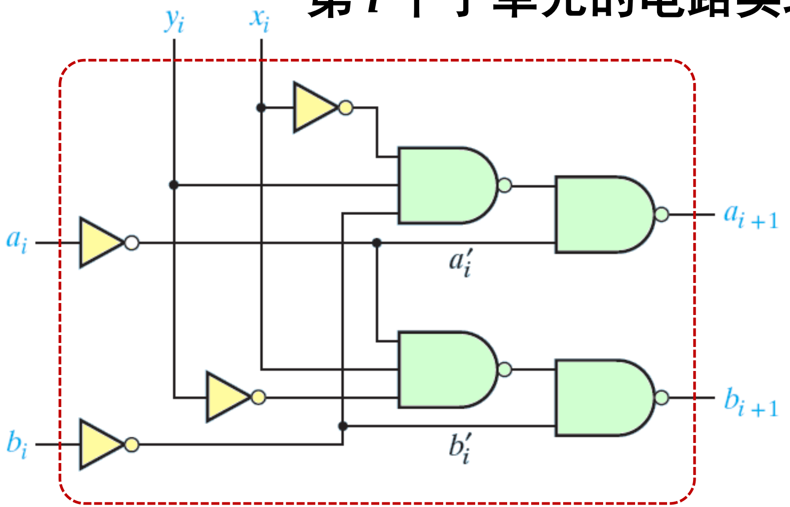
a_ib_i \ x_iy_i	00	01	11	10
00	0	1	0	0
01	0	0	0	0
11	X	X	X	X
10	1	1	1	1

$a_{i+1} = a_i + x_i' y_i b_i'$

a_ib_i \ x_iy_i	00	01	11	10
00	0	0	0	1
01	1	1	1	1
11	X	X	X	X
10	0	0	0	0

$b_{i+1} = b_i + x_i y_i' a_i'$

第 *i* 个子单元的电路实现



b_{n+1} \ a_{n+1}	0	1
0		1
1		X

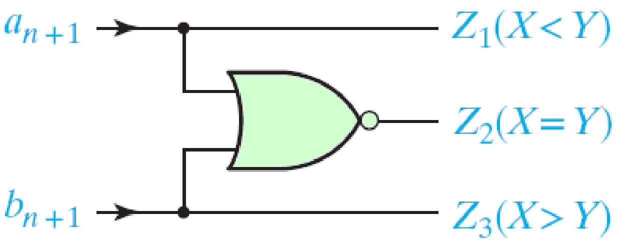
$Z_1 = a_{n+1}$

b_{n+1} \ a_{n+1}	0	1
0	1	
1		X

$Z_2 = a_{n+1}' b_{n+1}'$

b_{n+1} \ a_{n+1}	0	1
0		
1	1	X

$Z_3 = b_{n+1}$

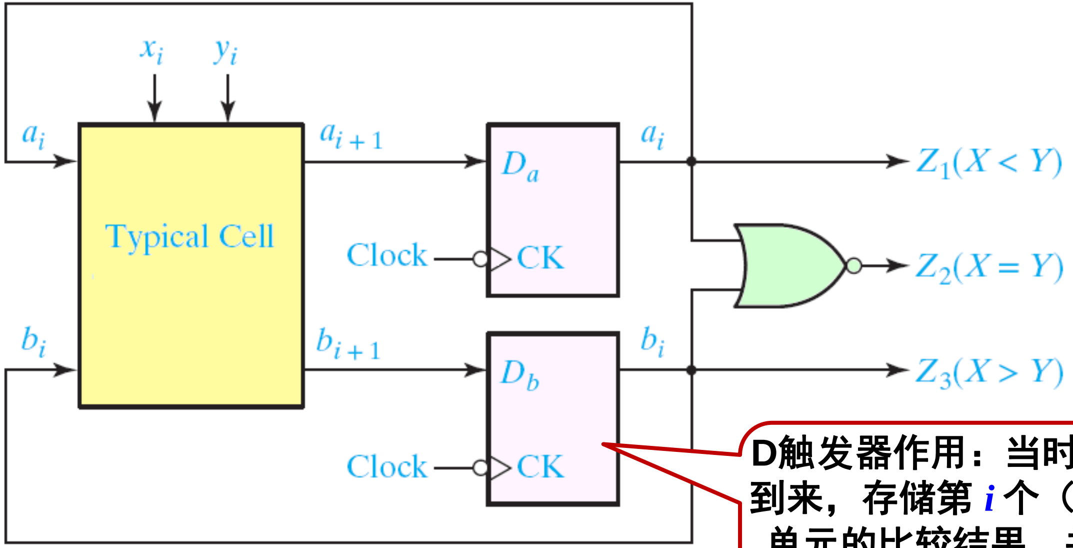


更复杂的同步时序设计_例8

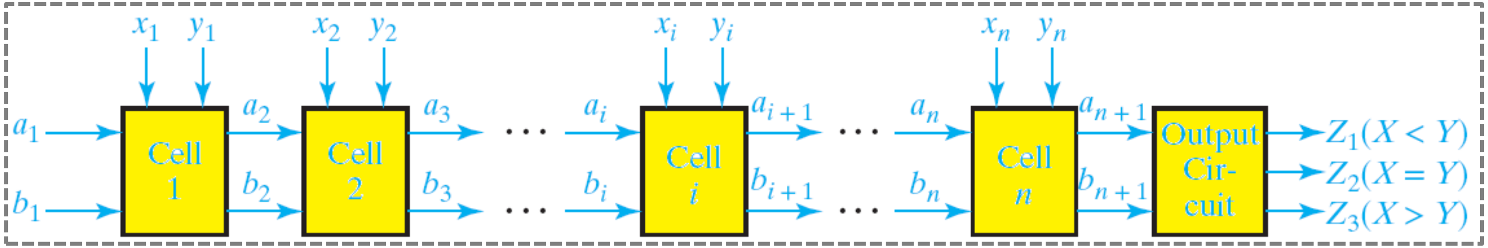
6. 电路实现

7. 无关项检查

（略）

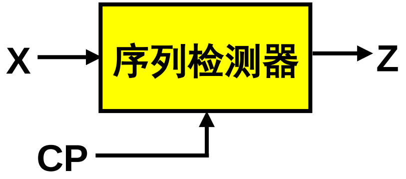


D触发器作用：当时钟信号到来，存储第 *i* 个（前一个）单元的比较结果，并产生第 *i* +1个单元的比较结果



更复杂的同步时序设计_例9

例9：利用D触发器设计一个同步时序电路，当输入序列以010或1001结尾时（允许重叠检测），输出Z为1，否则Z=0.



X = 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 1 0

↑a ↑b ↑c ↑d ↑e ↑f

Z = 0 0 0 1 0 1 0 1 1 0 0 0 1 0 1 0 0

1. Mealy型原始状态图构建

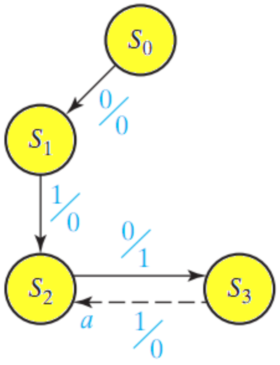
（1）子序列010检测的状态设定

S_0——初始复位状态，表示没有任何输入

S_1——表示序列以“0”结束

S_2——表示序列以“10”结束

S_3——表示序列以“010”结束，此时输出标志 Z=1。



（1）010检测的局部状态图

更复杂的同步时序设计_例9

（2）子序列1001检测的状态设定

S_0——初始复位状态，表示没有任何输入

S_1——表示序列以“0”结束

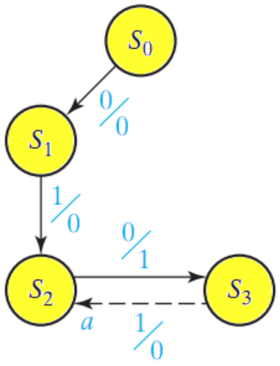
S_2——表示序列以“10”结束

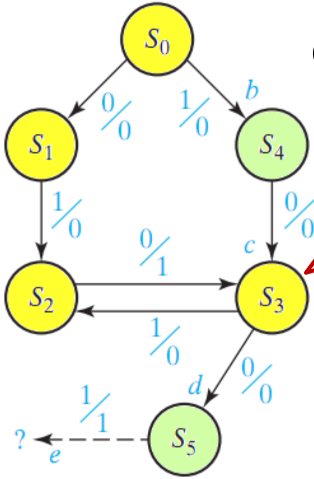
S_3——表示序列以“010”结束，此时输出标志 Z=1。

S_4——表示接收到1001序列的第一个“1”

S_5——表示序列以“100”结束。

重叠检测：010中的10
可以被1001检测重用





（2）1001检测的局部状态图

重叠检测：010中的10
可以被1001检测重用

X = 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 1 0

↑ a　↑ b　↑ c　↑ d　↑ e　↑ f

Z = 0 0 0 1 0 1 0 1 1 0 0 0 1 0 1 0 0

更复杂的同步时序设计_例9

（2）子序列1001检测的状态设定

S_0——初始复位状态，表示没有任何输入

S_1——表示序列以“0”结束

S_2——表示序列以“10”结束

S_3——表示序列以“010”结束，此时输出标志 Z=1。

S_4——表示接收到1001序列的第一个“1”

S_5——表示序列以“100”结束。

重叠检测：010中的10可以被1001检测重用

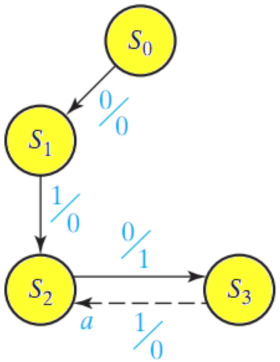
2. 状态化简（略）

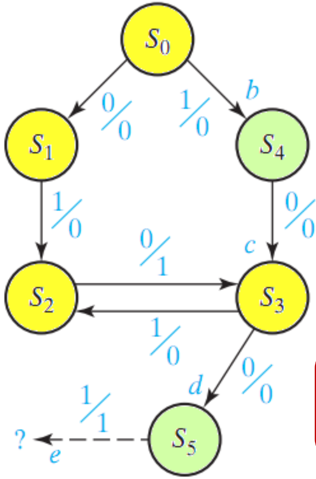
3.状态分配（略）

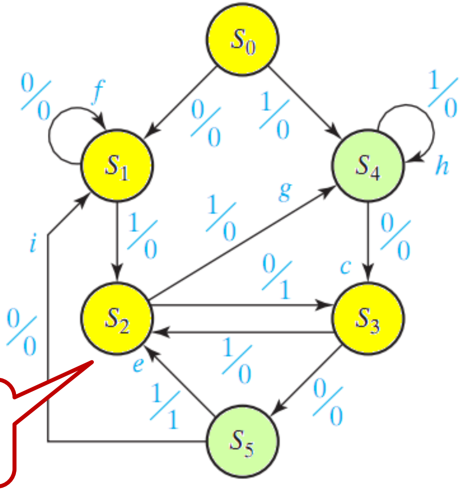
4.状态转换真值表（略）

5.卡诺图化简（略）

6. 电路实现（略）





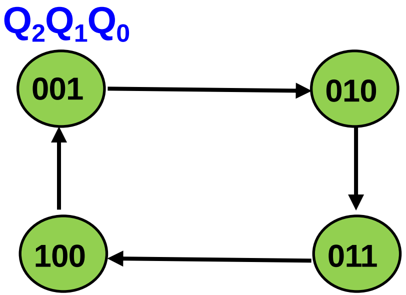


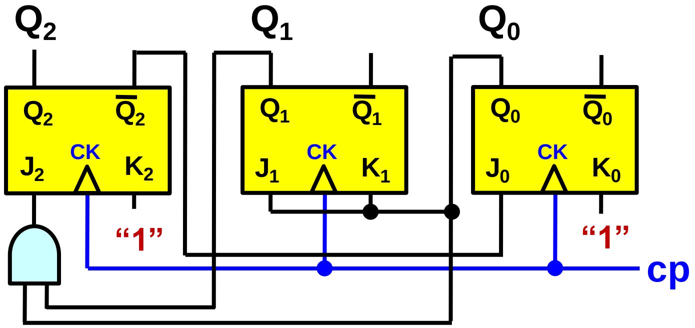
重叠检测：1001中的01可以被010检测重用

（3）010及1001检测的完整状态图

更复杂的同步时序设计_例10

例10:某同步时序电路如下所示，按图接线后，试验得到如下的循环状态。经检查：触发器工作正常，试分析故障所在。





1. 获得正确状态图

① 输入方程

$J_0=\overline{Q_2}^n$， $K_0=1$

$J_1=K_1=Q_0^n$

$J_2=Q_0^n Q_1^n$， $K_2=1$

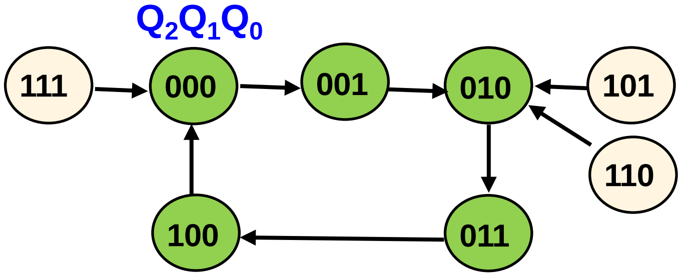
② 次态方程

$Q_0^{n+1}=\overline{Q_0}^n\,\overline{Q_2}^n$

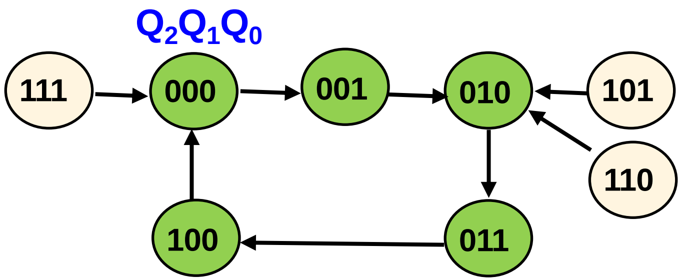
$Q_1^{n+1}=Q_1^n \oplus Q_0^n$

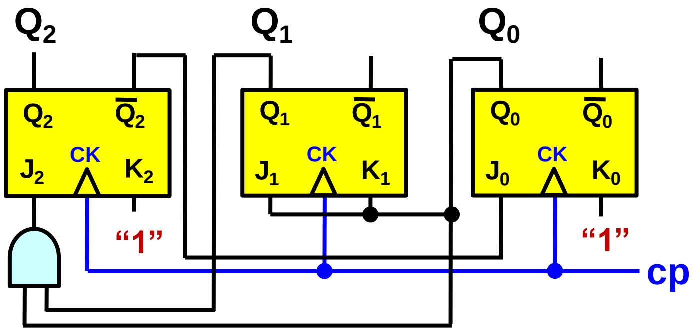
$Q_2^{n+1}=Q_0^n Q_1^n\,\overline{Q_2}^n$

③ 正确的状态转换图



更复杂的同步时序设计_例10





④ 电路功能：模5加法计数器，可自启动

2. 故障分析

① 触发器工作正常：　说明——电源和地线接触良好、时钟信号CP正常送入
故障只可能在进位链或驱动回路中

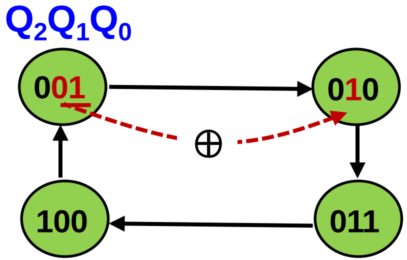
② 分析各触发器状态：　次态方程

$$Q_0^{n+1} = \overline{Q_0^n}\,\overline{Q_2^n}$$

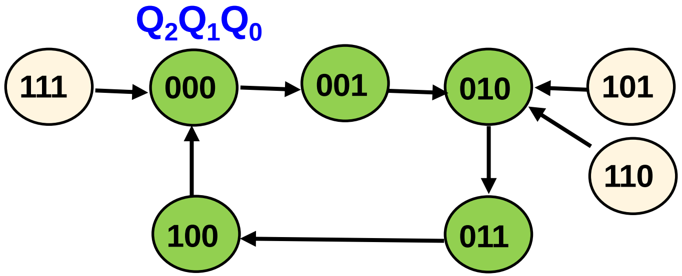
触发器FF1
没有问题

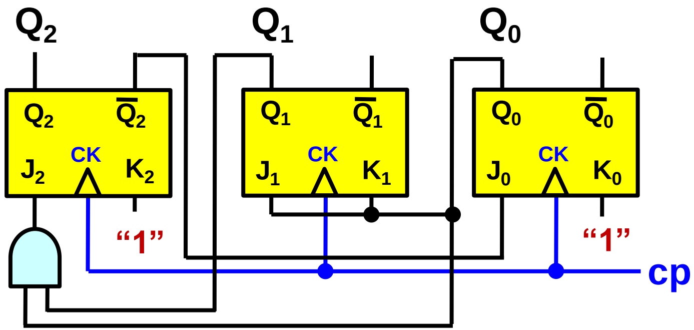
$$Q_1^{n+1} = Q_1^n \oplus Q_0^n$$

$$Q_2^{n+1} = Q_0^n\, Q_1^n\, \overline{Q_2^n}$$



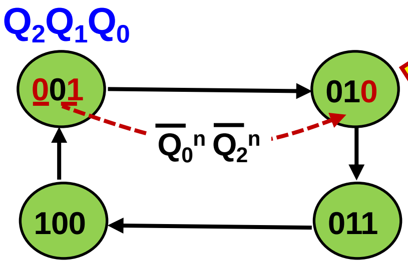
更复杂的同步时序设计_例10





2. 故障分析

② 分析各触发器状态：



触发器FF0
有问题

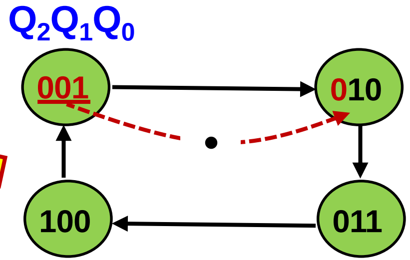
次态方程

$$Q_0^{n+1} = \overline{Q}_0^n\,\overline{Q}_2^n$$

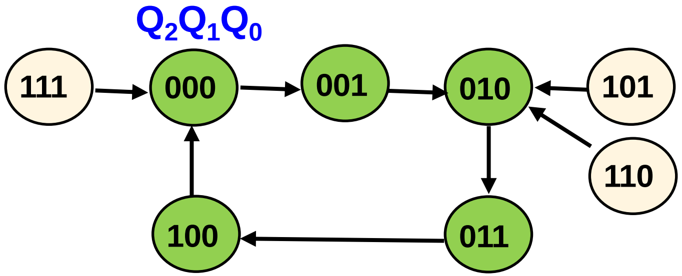
$$Q_1^{n+1} = Q_1^n \oplus Q_0^n$$

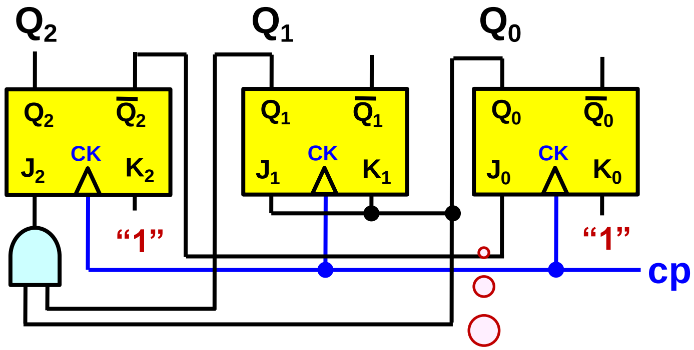
$$Q_2^{n+1} = Q_0^n\,Q_1^n\,\overline{Q}_2^n$$

触发器FF2
没有问题



更复杂的同步时序设计_例10





结论：$\overline{Q}_2$ 没有接入，J_0悬空

2. 故障分析

③ 针对触发器0分析：

K_0接触不良？

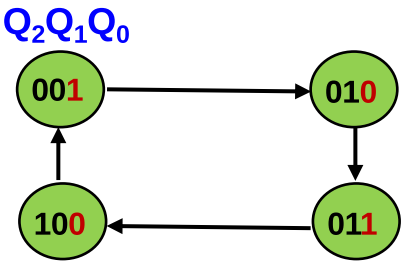
TTL电路管脚悬空等效为高电平1

K_0没问题

J_0接触不良？

$\overline{Q}_2$ 没有接入，J_0悬空等效为高电平1

触发器变成T',符合故障现象



利用触发器设计异步时序逻辑

异步时序逻辑设计的特点

- 异步时序电路中，没有统一的时钟脉冲
- 异步时序电路中要求每次输入信号发生变化后，必须等电路进入稳定状态，才允许输入信号再次发生改变
- 时钟脉冲作为一个输入变量考虑
- 为避免电路中出现竞争冒险，异步时序电路中每一时刻仅允许一个输入信号发生变化，不允许两个脉冲同时输入。n 个输入端有n+1个输入组合

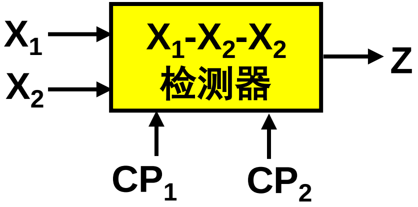
如：异步时序中，$X_1X_2X_3$是三个输入端，有四种输入组合：000、001、010、100。

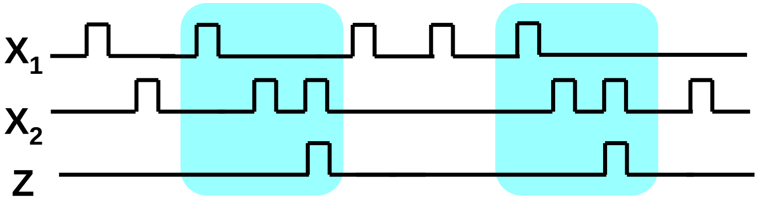
000——表示没有脉冲输入。

011、101、110、111是不允许出现的组合

利用触发器设计异步时序逻辑

例1：用D触发器设计一个X_1 - X_2 - X_2脉冲序列检测器，其中X_1、X_2为不同时出现的脉冲。





1. 建立原始状态表

① 设状态

S_0： 初始状态, X_1X_2 =00

S_1： 收到X_1, X_1X_2 =10

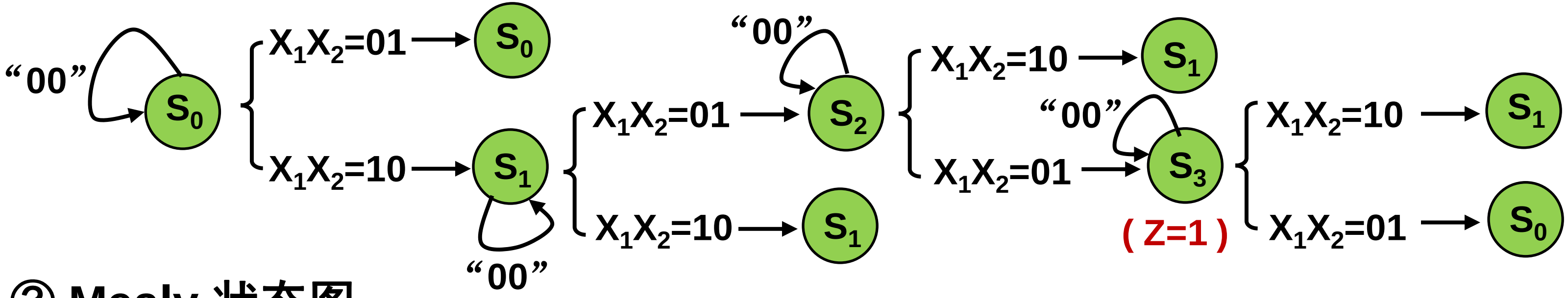
S_2： 收到X_1-X_2, 即10 →01

S_3： 收到X_1-X_2-X_2, 即10→01→01, 且Z=1。

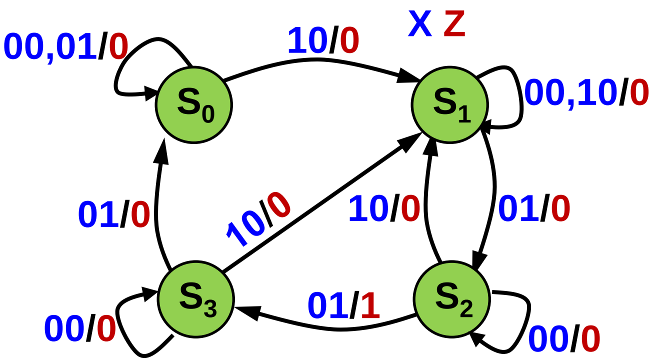
只标记感兴趣的子序列

利用触发器设计异步时序逻辑

② 状态转换情况



③ Mealy 状态图



④ 状态表

现态 Q^n	Q^{n+1}/ Z		
	$X_1X_2=00$	$X_1X_2=01$	$X_1X_2=10$
S_0	S_0 / 0	S_0 / 0	S_1 / 0
S_1	S_1/ 0	S_2 / 0	S_1 / 0
S_2	S_2 / 0	S_3 / 1	S_1 / 0
S_3	S_3 / 0	S_0 / 0	S_1/ 0

利用触发器设计异步时序逻辑

2. 状态表化简

现态 Q^n	Q^{n+1}/ Z			
	X_1X_2=00	X_1X_2=01	X_1X_2=10	
S_0	S_0 / 0	S_0 / 0	S_1 / 0	√
S_1	S_1 / 0	S_2 / 0	S_1 / 0	
S_2	S_2 / 0	S_3 / 1	S_1 / 0	
S_3	S_3 / 0	S_0 / 0	S_1 / 0	√

⇨

现态 Q^n	Q^{n+1}/ Z		
	X_1X_2=00	X_1X_2=01	X_1X_2=10
S_0	S_0 / 0	S_0 / 0	S_1 / 0
S_1	S_1 / 0	S_2 / 0	S_1 / 0
S_2	S_2 / 0	S_0 / 1	S_1 / 0

3. 状态编码

原则1：S_0S_2、S_0S_1、S_1S_2应取相邻编码

原则2：S_0S_1 、 S_1S_2 、 S_0S_2 应取相邻编码 ⇨

原则3：S_0S_2、 S_0S_1 、 S_1S_2应取相邻编码

	0	1
0	S_0	S_1
1	S_2	

S_0：00

S_1： 01

S_2：10

4、 D触发器的激励表

将CP看作控制函数，D触发器的特征表达式为：

$$Q^{n+1} = D.CP + Q^n.\overline{CP}$$

CP=1, $Q^{n+1} = D$
CP=0, $Q^{n+1} = Q$

驱动表

Q_n → Q_{n+1}	CP	D
0 → 0	0	X
0 → 1	1	1
1 → 0	1	0
1 → 1	0	X

	0	1
0	S_0	S_1
1	S_2	

S_0：00
S_1：01
S_2：10

现态 Q^n	Q^{n+1}/ Z		
	X_1X_2=00	X_1X_2=01	X_1X_2=10
S_0	S_0 / 0	S_0 / 0	S_1 / 0
S_1	S_1 / 0	S_2 / 0	S_1 / 0
S_2	S_2 / 0	S_0 / 1	S_1 / 0

确定CP_2：看$Q_2^n \rightarrow Q_2^{n+1}$
确定CP_1：看$Q_1^n \rightarrow Q_1^{n+1}$

确定D_2：看CP_2和Q_2^{n+1}
确定D_1：看CP_1和Q_1^{n+1}

输入及现态				次态		输入				输出
X_1	X_2	Q_2^n	Q_1^n	Q_2^{n+1}	Q_1^{n+1}	CP_2	D_2	CP_1	D_1	Z
0	0	0	0	0	0	0	X	0	X	0
0	0	0	1	0	1	0	X	0	X	0
0	0	1	0	1	0	0	X	0	X	0
0	0	1	1	X	X	X	X	X	X	X
0	1	0	0	0	0	0	X	0	X	0
0	1	0	1	1	0	1	1	1	0	0
0	1	1	0	0	0	1	0	0	X	1
0	1	1	1	X	X	X	X	X	X	X
1	0	0	0	0	1	0	X	1	1	0
1	0	0	1	0	1	0	X	0	X	0
1	0	1	0	0	1	1	0	1	1	0
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

利用触发器设计异步时序逻辑

5. 卡诺图化简

X_1X_2 \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	0
01	0	1	X	1
11	X	X	X	X
10	0	0	X	1

$$CP_2 = X_2Q_1^n + Q_2^n X_2 + X_1Q_2^n$$

X_1X_2 \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	0
01	0	1	X	0
11	X	X	X	X
10	1	0	X	1

$$CP_1 = \overline{Q_1}^n X_1 + Q_1^n X_2$$

X_1X_2 \ $Q_2^nQ_1^n$	00	01	11	10
00	X	X	X	X
01	X	0	X	X
11	X	X	X	X
10	1	X	X	1

$$D_1 = \overline{Q_1}^n$$

X_1X_2 \ $Q_2^nQ_1^n$	00	01	11	10
00	X	X	X	X
01	X	1	X	0
11	X	X	X	X
10	X	X	X	0

$$D_2 = Q_1^n$$

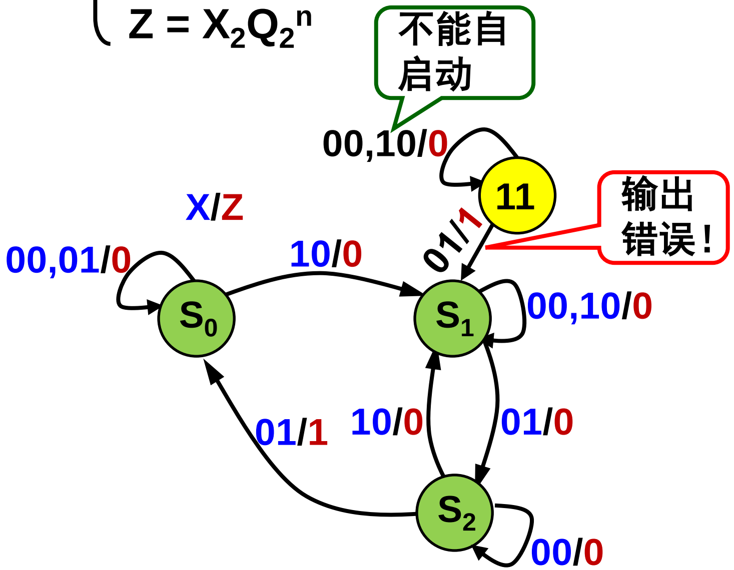
X_1X_2 \ $Q_2^nQ_1^n$	00	01	11	10
00	0	0	X	0
01	0	0	X	1
11	X	X	X	X
10	0	0	X	0

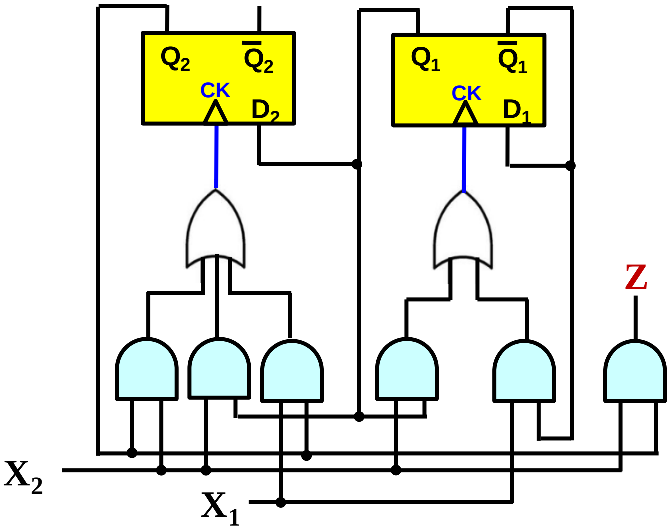
$$Z = X_2Q_2^n$$

利用触发器设计异步时序逻辑

6. 逻辑图

$$\begin{cases} CP_2 = X_2Q_1^n + Q_2^n X_2 + X_1Q_2^n \\ CP_1 = \overline{Q_1}^n X_1 + Q_1^n X_2 \\ D_2 = Q_1^n \\ D_1 = \overline{Q_1}^n \\ Z = X_2Q_2^n \end{cases}$$





7. 检查无关项

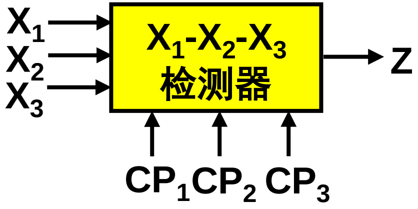
无关状态：$Q_2^nQ_1^n$=11

X_1X_2 分别为 00, 01,10时，带入计算

$$\begin{cases} Q_2^{n+1} = D_2 = Q_1^n\text{；} \quad CP_2 = X_2Q_1^n + Q_2^n X_2 + X_1Q_2^n \\ Q_1^{n+1} = D_1 = \overline{Q_1}^n\text{；} \quad CP_1 = \overline{Q_1}^n X_1 + Q_1^n X_2 \\ Z = X_2Q_2^n \end{cases}$$

利用触发器设计异步时序逻辑

例2：用D触发器设计一个X_1-X_2-X_3异步脉冲序列检测器，其中X_1、X_2、X_3为不同时出现的脉冲



1. 建立原始状态表

① 设状态

S_0：初始状态, $X_1X_2X_3$=000

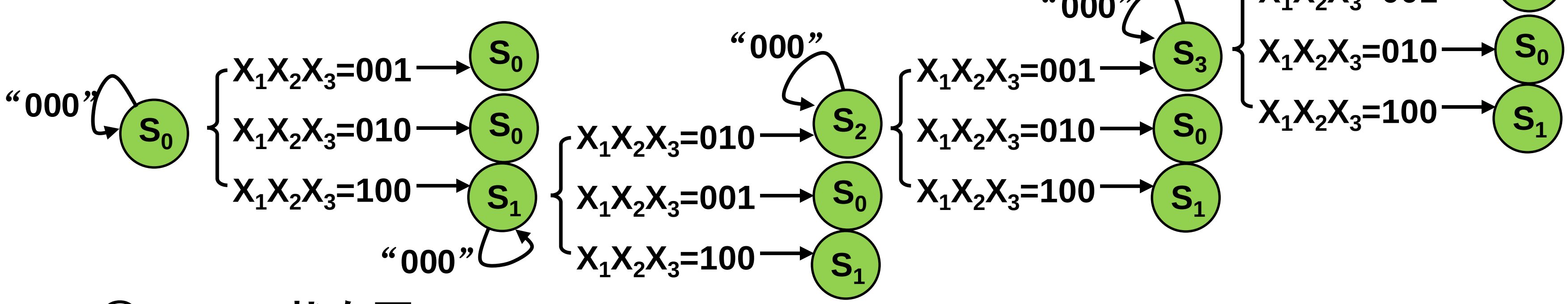
S_1：收到X_1, $X_1X_2X_3$=100

S_2：收到X_1-X_2, 即100 →010

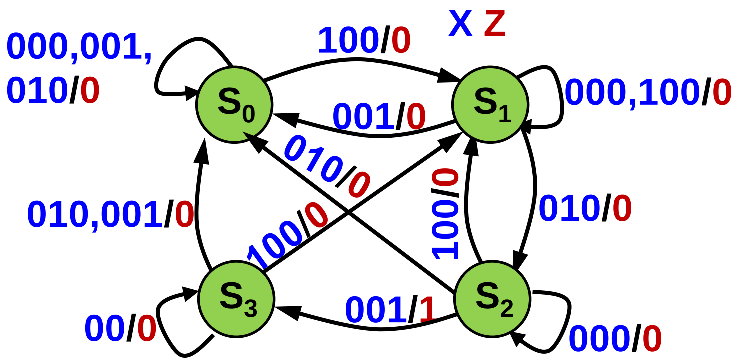
S_3：收到X_1-X_2-X_3, 即100→010→001, 且Z=1。

利用触发器设计异步时序逻辑

② 状态转换情况



③ Mealy 状态图



④ 状态表

现态 Q^n	Q^{n+1}/ Z			
	$X_1X_2X_3$=000	$X_1X_2X_3$=100	$X_1X_2X_3$=010	$X_1X_2X_3$=001
S_0	S_0 / 0	S_1 / 0	S_0 / 0	S_0 / 0
S_1	S_1/ 0	S_1 / 0	S_2 / 0	S_0 / 0
S_2	S_2 / 0	S_1 / 0	S_0/ 0	S_3/ 1
S_3	S_3 / 0	S_1 / 0	S_0/ 0	S_0/ 0

利用触发器设计异步时序逻辑

2. 状态表化简

现态 Q^n	Q^{n+1}/ Z			
	$X_1X_2X_3$=000	$X_1X_2X_3$=100	$X_1X_2X_3$=010	$X_1X_2X_3$=001
S_0	S_0 / 0	S_1 / 0	S_0 / 0	S_0 / 0
S_1	S_1/ 0	S_1 / 0	S_2 / 0	S_0 / 0
S_2	S_2 / 0	S_1 / 0	S_0/ 0	S_3/ 1
S_3	S_3 / 0	S_1 / 0	S_0/ 0	S_0/ 0

现态 Q^n	Q^{n+1}/ Z			
	$X_1X_2X_3$=000	$X_1X_2X_3$=100	$X_1X_2X_3$=010	$X_1X_2X_3$=001
S_0	S_0 / 0	S_1 / 0	S_0 / 0	S_0 / 0
S_1	S_1/ 0	S_1 / 0	S_2 / 0	S_0 / 0
S_2	S_2 / 0	S_1 / 0	S_0/ 0	S_0/ 1

3. 状态编码

原则1：S_0S_2、S_0S_1、S_1S_2应取相邻编码

原则2：S_0S_1 、 S_1S_2 、 S_0S_2 应取相邻编码

原则3：S_0S_2、 S_0S_1 、 S_1S_2应取相邻编码

	0	1
0	S_0	S_1
1	S_2	

S_0： 00

S_1： 01

S_2： 10

利用触发器设计异步时序逻辑

4、 状态转换真值表

D触发器驱动表

$Q_n \rightarrow Q_{n+1}$	CP	D
0 → 0	0	X
0 → 1	1	1
1 → 0	1	0
1 → 1	0	X

	0	1
0	S_0	S_1
1	S_2	

S_0： 00
S_1： 01
S_2： 10

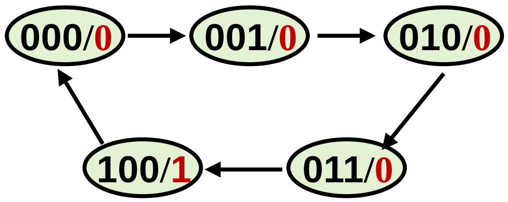
状态转换真值表？

利用触发器设计异步计数器

例1：试用JK触发器设计异步模5加法计数器

① 确定触发器个数：需要3个JK触发器，↓触发

② 画状态转换图



③ 确定触发器CP的接法

CP	Q_3	Q_2	Q_1
↓	0	0	0
↓	0	0	1
↓	0	1	0
↓	0	1	1
↓	1	0	0
↓	0	0	0

设计原则

- 时序图中，凡是触发器状态翻转的地方，都必须为其提供时钟脉冲。
- 在满足翻转的前提下，时钟脉冲越少越好

对触发器而言：只要提供时钟，状态的保持就必须依靠输入端（如J、K）的控制来实现。

Q_1——由CP提供下降沿，CP_1=CP

Q_2——翻转两次，需两个下降沿，恰好此时Q_1有两个下降沿，CP_2= Q_1↓

Q_3——翻转两次，需两个下降沿，此时Q_2、Q_1都不能提供，CP_3只能接CP

利用触发器设计异步计数器

④ 状态转换真值表

$CP_1 = CP_3 = CP\downarrow$, $CP_2 = Q_1\downarrow$

确定J_3K_3： 看$Q_3^n \to Q_3^{n+1}$
确定J_1K_1： 看$Q_1^n \to Q_1^{n+1}$

现态			次态			输入						输出
Q_3^n	Q_2^n	Q_1^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	J_3	K_3	J_2	K_2	J_1	K_1	Z
0	0	0	0	0	1	0	X	X	X	1	X	0
0	0	1	0	1	0	0	X	1	X	X	1	0
0	1	0	0	1	1	0	X	X	X	1	X	0
0	1	1	1	0	0	1	X	X	1	X	1	0
1	0	0	0	0	0	X	1	X	X	0	X	1

此时Q_1无下降沿, J_2 K_2为任意

确定J_2K_2：
看$Q_1^n \to Q_1^{n+1}$

$$\begin{cases} J_3 = Q_2^nQ_1^n\ ,\ K_3 = 1 \\ J_2 = 1\ ,\ K_2 = 1 \\ J_1 = \overline{Q}_3^n,\ K_1 = 1 \\ Z = Q_3^n \quad ,CP_2 = Q_1\downarrow \quad ,CP_3 = CP_1 = CP \end{cases}$$

⑤ 卡诺图化简

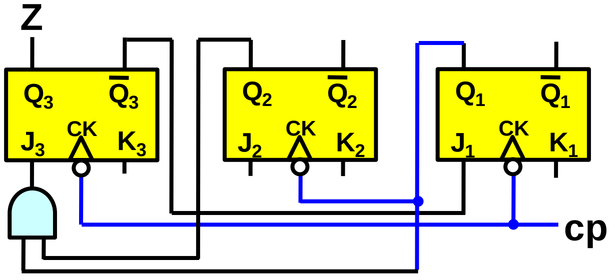
Q_3^n \ $Q_2^nQ_1^n$	00	01	11	10
0	0	0	1	0
1	X	X	X	X

$J_3 = Q_2^nQ_1^n$

Q_3^n \ $Q_2^nQ_1^n$	00	01	11	10
0	1	X	X	1
1	0	X	X	X

$J_1 = \overline{Q}_3^n$

⑥ 逻辑图

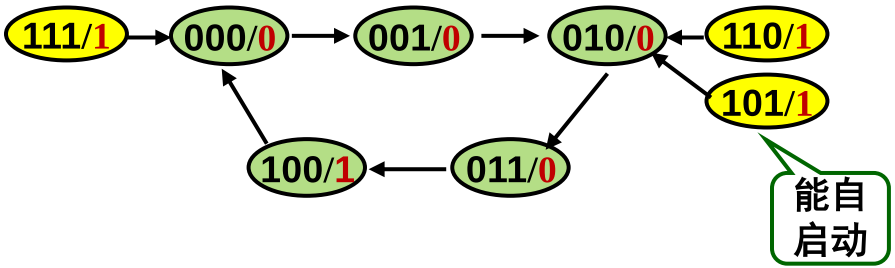


利用触发器设计异步计数器

⑦检查无关项

现态			次态			输入			输出
Q_3^n	Q_2^n	Q_1^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	CP_3	CP_2	CP_1	Z
1	0	1	0	1	0	↓	↓	↓	1
1	1	0	0	1	0	↓	0	↓	1
1	1	1	0	0	0	↓	↓	↓	1

$$\begin{cases} J_3 = Q_2^n Q_1^n \text{ , } K_3 = 1 \\ J_2 = 1 \text{ , } K_2 = 1 \\ J_1 = \overline{Q}_3^n \text{, } K_1 = 1 \\ Z = Q_3^n \quad \text{,} CP_2 = Q_1 \downarrow \quad \text{,} CP_3 = CP_1 = CP \end{cases}$$



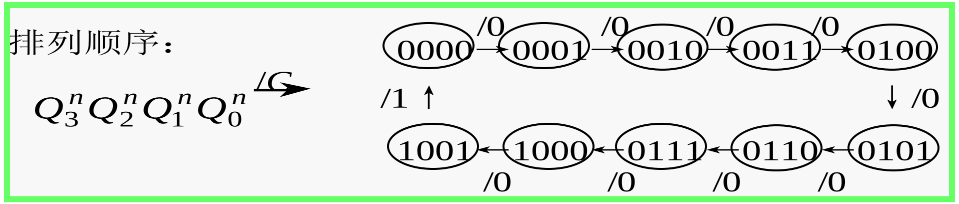
利用触发器设计异步计数器

例2：用*D*触发器设计实现十进制异步加法计数器

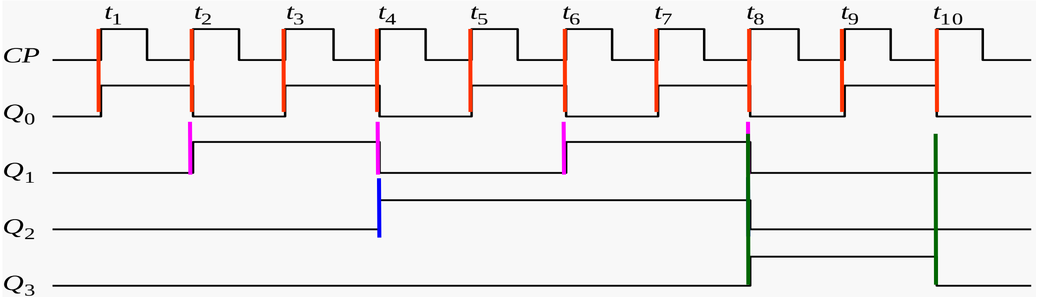
① 确定触发器个数：需要4个D触发器，↑触发

② 画状态转换图

输出方程： $C = Q_3^n Q_0^n$



③ 确定触发器CP的接法



$$
\begin{cases}
CP_0 = CP \\
CP_1 = \overline{Q_0} \\
CP_2 = \overline{Q_1} \\
CP_3 = \overline{Q_0}
\end{cases}
$$

选择时钟脉冲的基本原则：在满足翻转要求的条件下，触发沿越少越好。

利用触发器设计异步计数器

④ 状态转换真值表

$$\begin{cases} CP_0 = CP \\ CP_1 = \overline{Q_0} \\ CP_2 = \overline{Q_1} \\ CP_3 = \overline{Q_0} \end{cases}$$

现态				次态				输入							
Q_3^n	Q_2^n	Q_1^n	Q_0^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	CP_3	CP_2	CP_1	CP_0	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	1	0	0	0	↑	X	X	X	1
0	0	0	1	0	0	1	0	↑	0	↑	↑	0	X	1	0
0	0	1	0	0	0	1	1	0	0	0	↑	X	X	X	1
0	0	1	1	0	1	0	0	↑	↑	↑	↑	0	1	0	0
0	1	0	0	0	1	0	1	0	0	0	↑	X	X	X	1
0	1	0	1	0	1	1	0	↑	0	↑	↑	0	X	1	0
0	1	1	0	0	1	1	1	0	0	0	↑	X	X	X	1
0	1	1	1	1	0	0	0	↑	↑	↑	↑	1	0	0	0
1	0	0	0	1	0	0	1	0	0	0	↑	X	X	X	1
1	0	0	1	0	0	0	0	↑	0	↑	↑	0	X	0	0

⑤ 卡诺图化简

$Q_3^nQ_2^n$ \ $Q_1^nQ_0^n$	00	01	11	10
00	X	0	0	X
01	X	0	1	X
11	X	X	X	X
10	X	0	X	X

$D_3 = Q_2^nQ_1^n$

$Q_3^nQ_2^n$ \ $Q_1^nQ_0^n$	00	01	11	10
00	X	X	1	X
01	X	X	0	X
11	X	X	X	X
10	X	X	X	X

$D_2 = \overline{Q_2}^n$

$Q_3^nQ_2^n$ \ $Q_1^nQ_0^n$	00	01	11	10
00	X	1	0	X
01	X	1	0	X
11	X	X	X	X
10	X	0	X	X

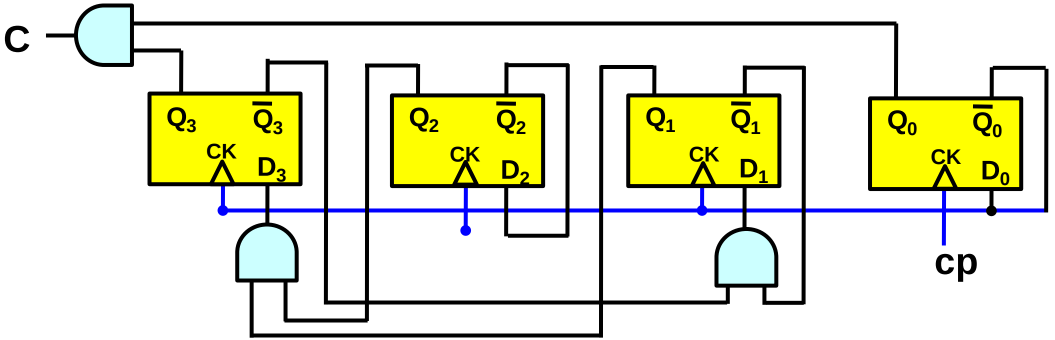
$D_1 = \overline{Q_3}^n\,\overline{Q_1}^n$

$Q_3^nQ_2^n$ \ $Q_1^nQ_0^n$	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

$D_0 = \overline{Q_0}^n$

利用触发器设计异步计数器

⑥ 逻辑图



$$\begin{cases} CP_0 = CP \\ CP_1 = \overline{Q_0} \\ CP_2 = \overline{Q_1} \\ CP_3 = \overline{Q_0} \end{cases}$$

$$\begin{cases} D_3 = Q_2^n Q_1^n \\ D_2 = \overline{Q_2^n} \\ D_1 = \overline{Q_3^n}\,\overline{Q_1^n} \\ D_0 = \overline{Q_0^n} \\ C = Q_3^n Q_0^n \end{cases}$$

⑦检查无关项

将无效状态1010～1111分别代入状态方程,可以验证该电路能够自启动。

现态				次态				输入			
Q_3^n	Q_2^n	Q_1^n	Q_0^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	CP_3	CP_2	CP_1	CP_0
1	0	1	0	0	0	0	1	0	0	0	↑
1	0	1	1	0	1	0	0	↑	↑	↑	↑
1	1	0	0	1	1	0	1	0	0	0	↑
1	1	0	1	0	1	0	0	↑	0	↑	↑
1	1	1	0	1	1	1	1	0	0	0	↑
1	1	1	1	0	0	0	0	↑	↑	↑	↑