第4章
同步时序电路

Video Image Processing (VIP)
Research Group @ Fudan
http://soc.fudan.edu.cn/vip/

范益波

2013.9

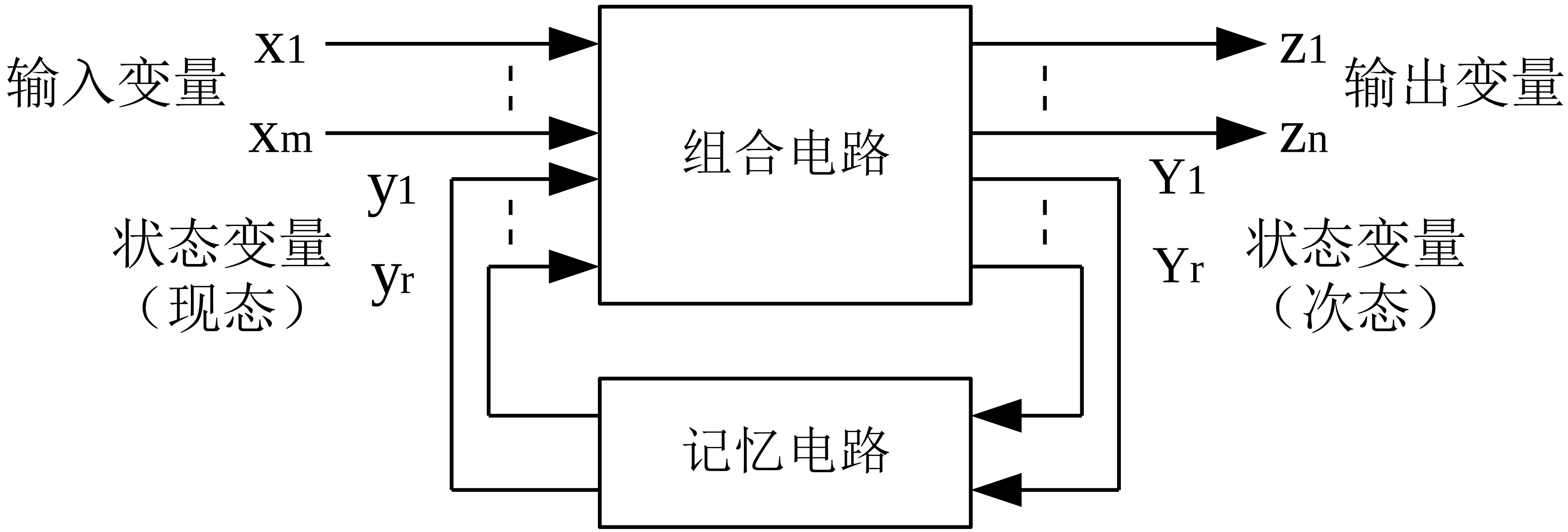
本章内容

- 什么是同步时序电路？
- 什么是电路的状态？
- 状态需要“分配”与“化简”么？
- 如何估算电路的时序是否满足要求？

本章要求

- 掌握同步时序电路的基本分析过程
- 掌握同步时序电路的设计原理
- 掌握状态表的化简过程

4.1 时序电路的描述



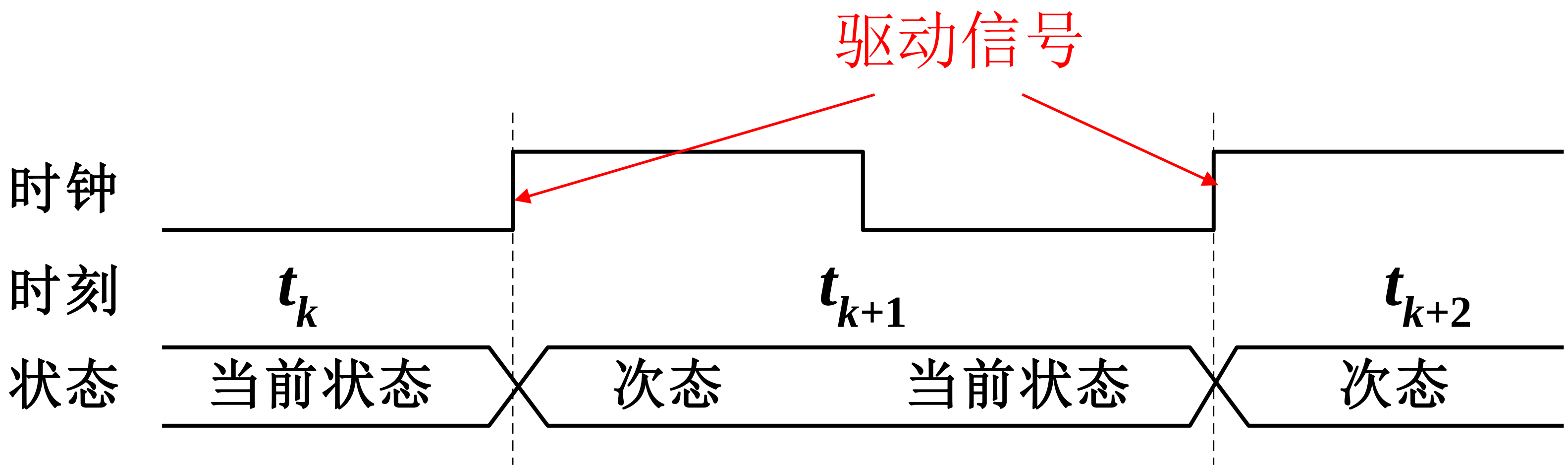
注：这是一个一般的结构，在实际的逻辑中可以合并某些输出和状态，也可以没有输入（如计数器）。

同步时序电路和异步时序电路

同步时序电路：记忆电路一般由触发器构成，记忆电路中所有触发器状态的变化都是在同一时钟信号操作下同时发生的。触发器的时钟信号不计在输入之内。

异步时序电路：记忆电路可以由触发器构成，也可以由组合电路的反馈构成。记忆电路状态的变化不是同时发生的，可能有公共的时钟信号，也可能没有公共的时钟信号。

现态与次态概念



以两次驱动（在同步时序逻辑中就是时钟）的间隔时间作为时序电路的定时单位，把某个间隔时刻 t_k 作为“当前时刻”，将下一个间隔时刻 t_{k+1} 称为“次时刻”。

对于“当前时刻”和“次时刻”的表述，都是相对于时刻 t_k 而言。当前时刻的状态为现态，次时刻的状态为次态。

时序电路的状态方程与输出方程

$$\boldsymbol{Y}(t_k) = f_1[\boldsymbol{x}(t_k), \boldsymbol{y}(t_k)]$$

$$\boldsymbol{z}(t_k) = f_2[\boldsymbol{x}(t_k), \boldsymbol{y}(t_k)]$$

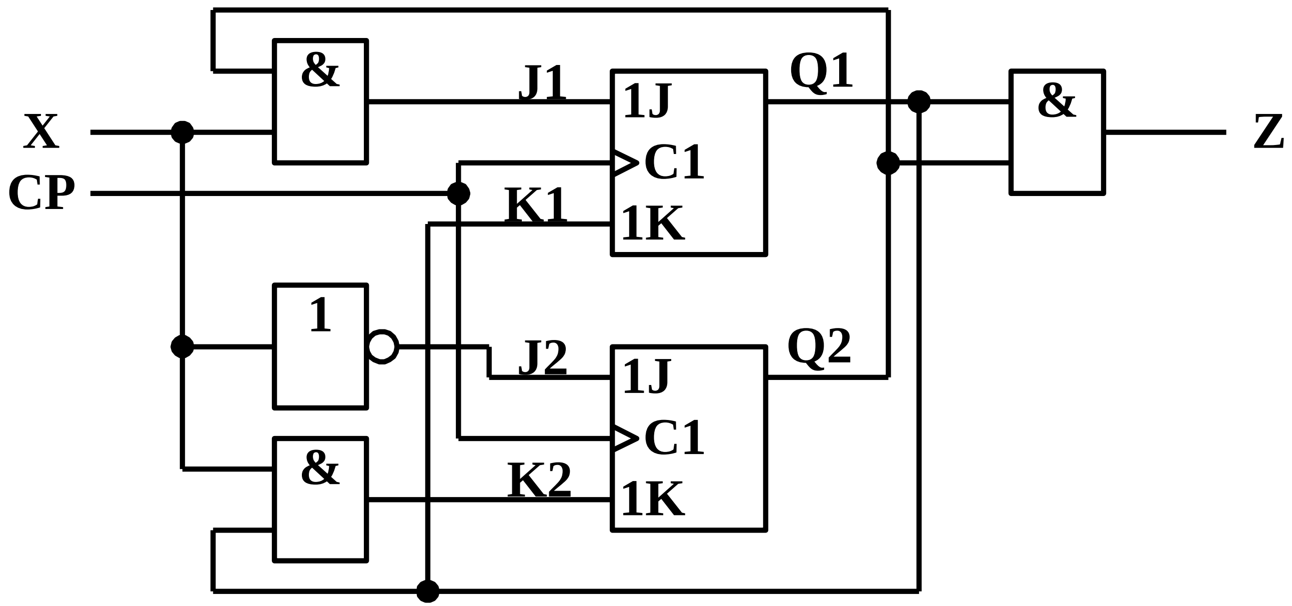
意义:

次态是输入与现态的函数（一般情况，也可以无输入）
输出是输入与现态的函数（一般情况，也可以无输入）

注意点:

Y 是次态变量，通常是一个隐含的变量，不一定是触发器的激励输入。只有记忆电路全部是 D 触发器时，次态才与激励相同。

例1 状态机



JK 触发器，Q1、Q2 是现态， X 是输入。

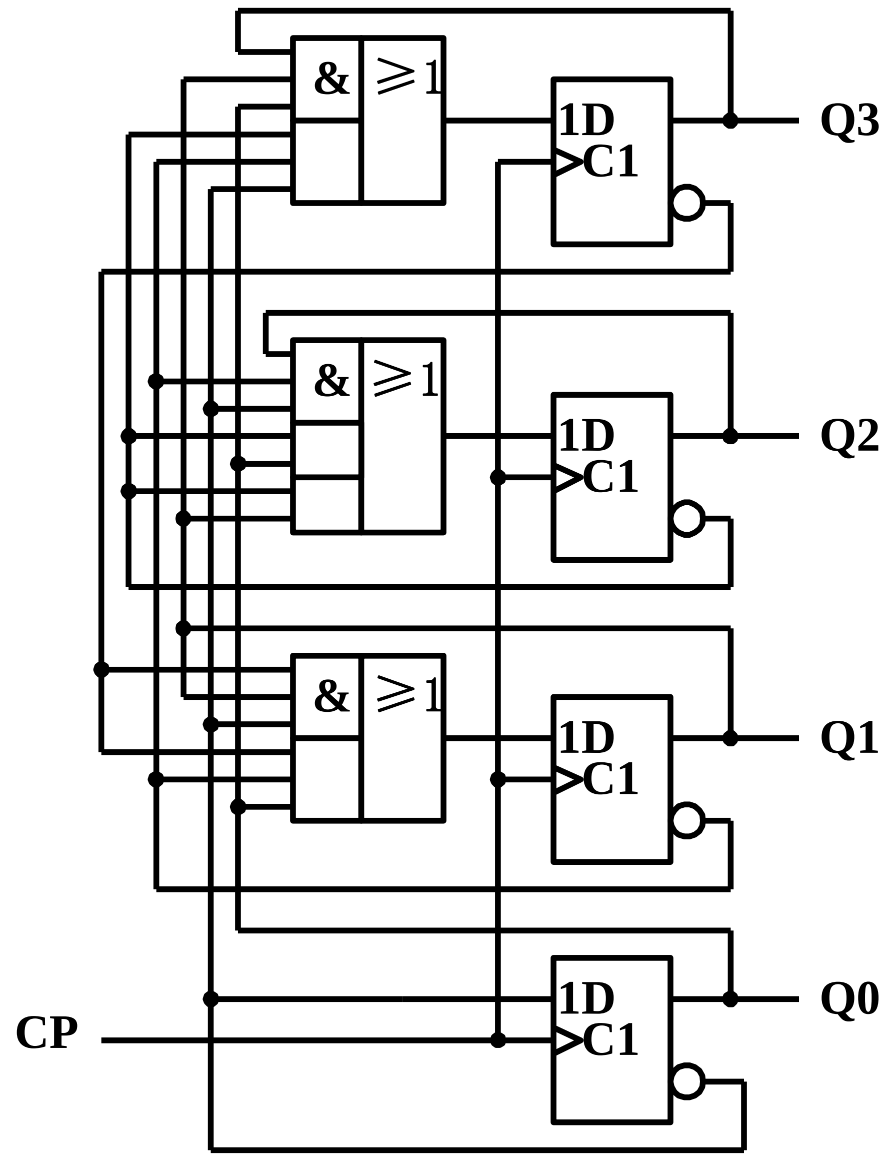
Z 是输出，仅是现态的函数。

次态隐含在 J1、K1、J2、K2 中。

例2 计数器

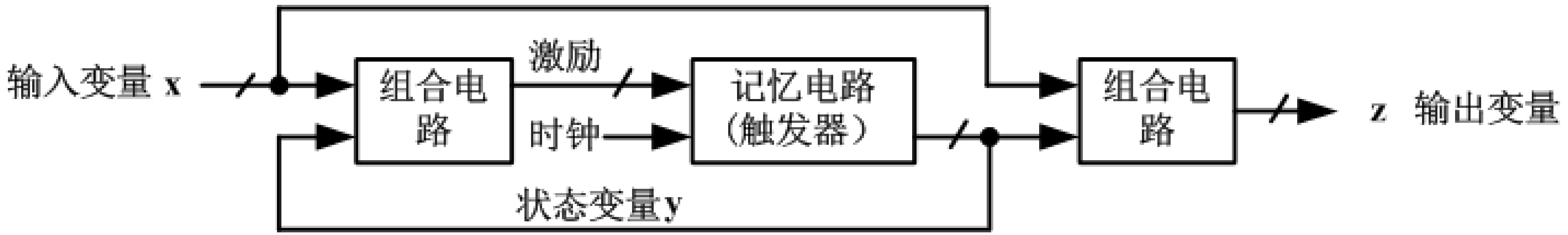
D 触发器，无输入，Q0 ~ Q3 为状态同时也是输出。

次态是 D0 ~ D3，可以通过组合逻辑显式地写出。



米利模型和摩尔模型

米利（Mealy）模型

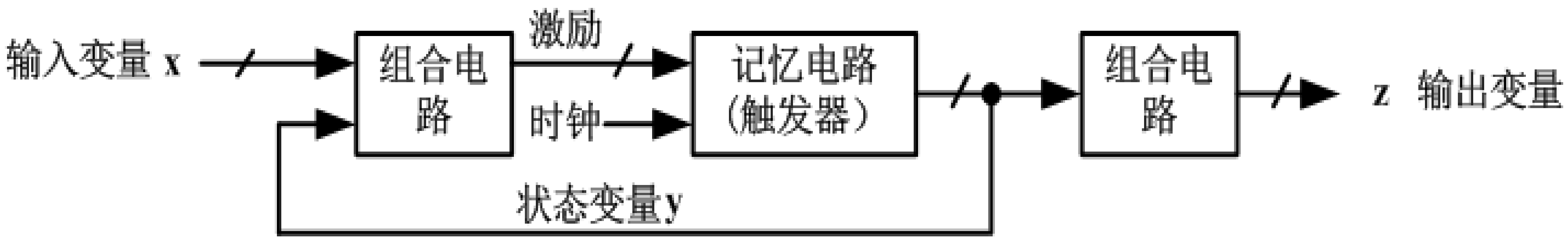


$$\boldsymbol{z}(t_k) = f_1[\boldsymbol{x}(t_k), \boldsymbol{y}(t_k)]$$

$$\boldsymbol{Y}(t_k) = f_2[\boldsymbol{x}(t_k), \boldsymbol{y}(t_k)]$$

输出：当前输入和现态的函数
次态：当前输入和现态的函数

摩尔(Moore)模型



$$z(t_k) = f_1[y(t_k)]$$

$$Y(t_k) = f_2[x(t_k), y(t_k)]$$

输出：现态的函数
次态：当前输入和现态的函数

米利模型和摩尔模型的区别：

一、米利模型的输出直接同输入有关，所以在输入变化时，不管状态是否改变，输出立即产生变化。即输入不仅影响次态，同时影响输出。

二、摩尔模型的输出只同状态有关，所以在整个状态保持期间保持输出不变。输入的变化只影响次态。

三、根据上述情况，若输入与时钟同步，则两种模型的输出在整个时钟周期内均保持不变，但米利模型比摩尔模型提前一个时钟周期改变输出。

四、若输入存在干扰，一般不会影响摩尔模型的输出，但可以影响米利模型的输出。

问题：第三章的数字序列检测器（P129）是什么类型的？

状态转换表

以表格的形式描述现态、输入与次态、输出的关系。

米利模型的表格形式是：

现态	次态 / 输出		
	输入1	输入2	输入n
现态1	次态11 / 输出11	次态12 / 输出12	次态1n / 输出1n
现态2	次态21 / 输出21	次态22 / 输出22	次态2n / 输出2n
现态m	次态m1 / 输出m1	次态m2 / 输出m2	次态mn / 输出mn

时序电路作为控制，就是在各状态间转换。

状态转换表

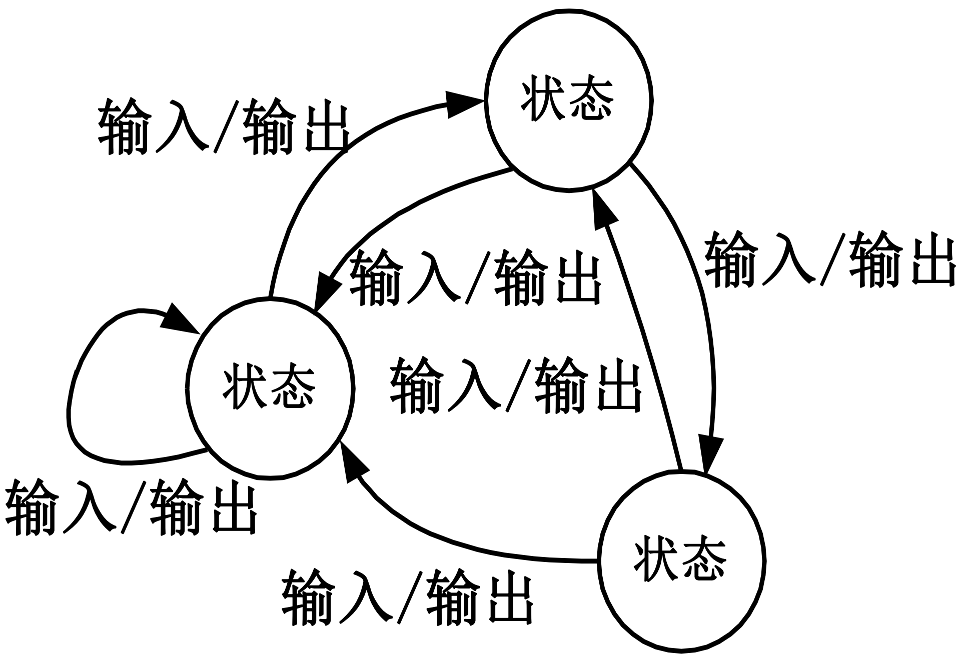
摩尔模型的表格形式是:

现态	次态			输出
	输入1	输入2	输入n	
现态1	次态11	次态12	次态1n	输出1
现态2	次态21	次态22	次态2n	输出2
现态m	次态m1	次态m2	次态mn	输出m

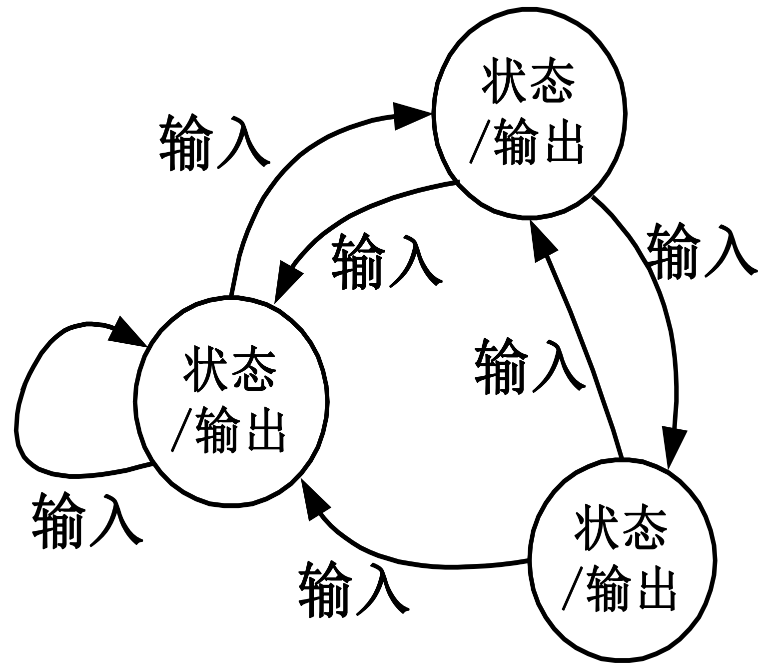
状态转换图

以信号流图形式显示状态转换关系。

米利模型形式将输出写在转换线上，摩尔模型形式将输出写在状态圈内。



米利模型



摩尔模型

状态转换图的特点

状态转换图中每个状态射出的状态转换线的根数同系统输入的组合数相同，转换条件包含了所有的输入组合。例如某系统输入组合有3种：00、01和10，则无论哪个模型，每个状态射出的状态转换线都是3根，分别对应3个输入组合。这个特点常常被用来检查状态转换图的正确性。

摩尔模型的状态数通常大于米利模型的状态数。形成这个特点的原因是由于米利模型中一个状态可以对应多个输出，而摩尔模型一个状态只能对应一个输出。

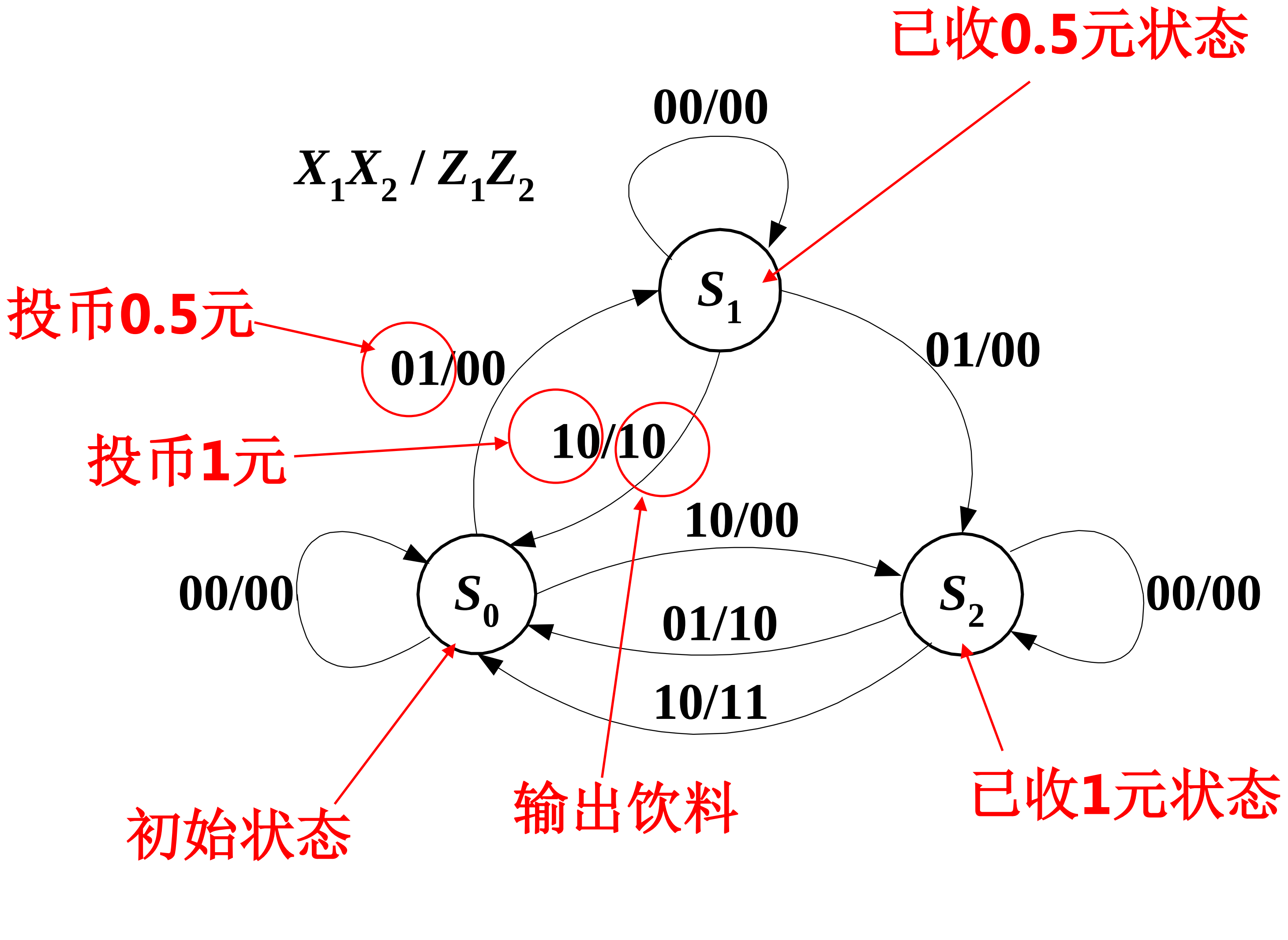
例4-1

自动售饮料机。可以投入1元或5角的硬币，饮料1.5元一杯。当先后投入的硬币满1元5角后，机器送出一杯饮料；当投入的硬币满2元后，机器送出一杯饮料以及送出一个5角硬币。作出上述自动售饮料机问题的状态转换图和状态转换表。

分析1：

- **输出**：设Z_1 =1→输出饮料；Z_2 =1→输出找零。所有的输出情况为$Z_1Z_2 = 00$、$Z_1Z_2 = 10$、$Z_1Z_2 = 11$。
- **输入**：当前投入的币值，$X_1X_2 = 00$、币值为0；$X_1X_2 = 01$、币值为5角；$X_1X_2 = 10$、币值为1元。
- **状态**：记录已经投入的币值，S_0＝0、 S_1 ＝5角、S_2 ＝1元。

米利模型的状态图


已收0.5元状态
00/00
X_1X_2 / Z_1Z_2
S_1
投币0.5元
01/00
01/00
投币1元
10/10
10/00
00/00
S_0
S_2
00/00
01/10
10/11
初始状态
输出饮料
已收1元状态

米利模型的状态转换表

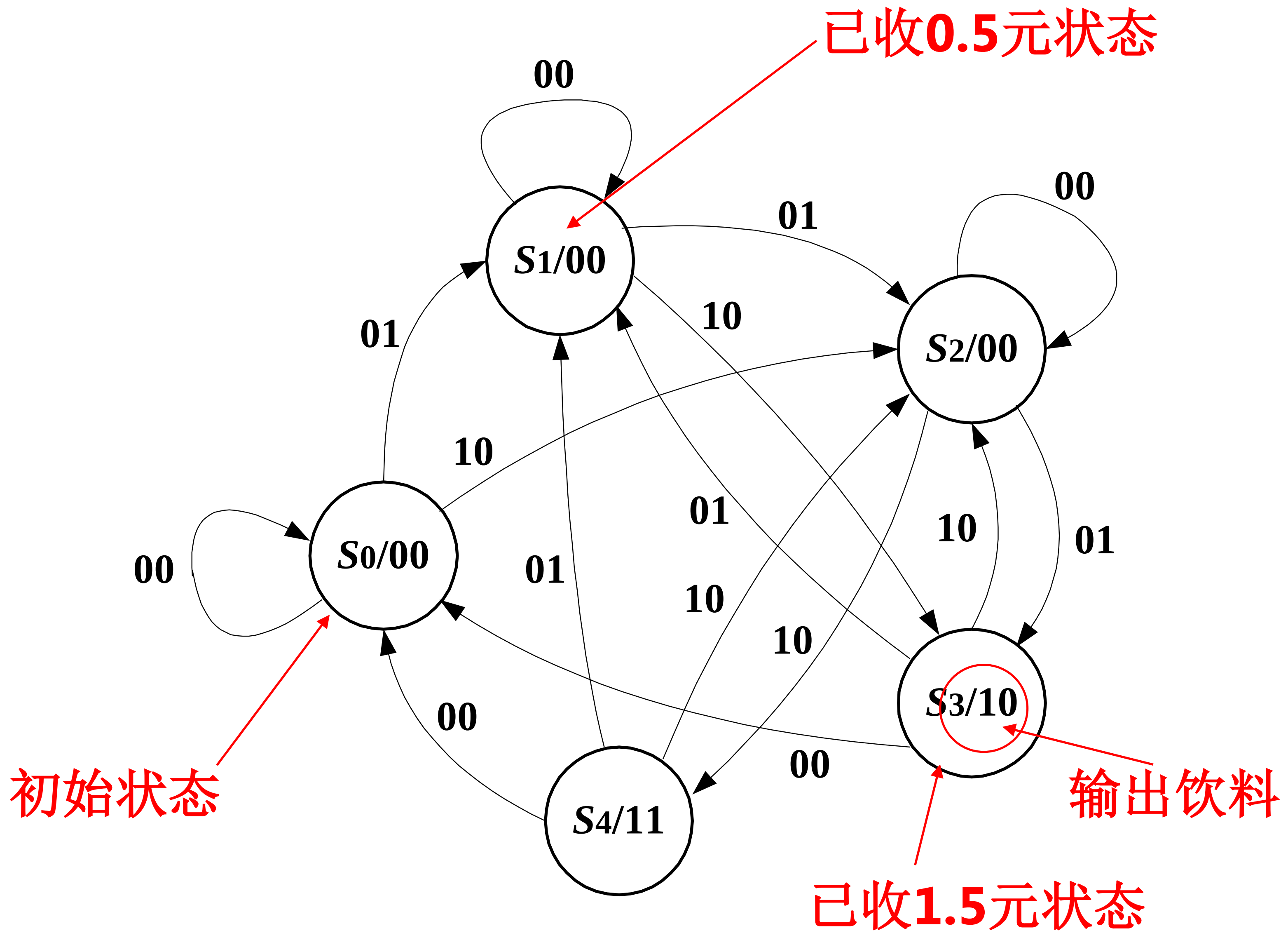
现态	次态 / 输出Z_1Z_2		
	$X_1X_2 = 00$	$X_1X_2 = 01$	$X_1X_2 = 10$
S_0	$S_0/00$	$S_1/00$	$S_2/00$
S_1	$S_1/00$	$S_2/00$	$S_0/10$
S_2	$S_2/00$	$S_0/10$	$S_0/11$

此处：注意此时状态还没有编码。要实现该电路，需要对状态编码

分析2：

- **输出**：设$Z_1=1 \rightarrow$输出饮料；$Z_2=1 \rightarrow$输出找零。所有的输出情况为$Z_1Z_2=00$、$Z_1Z_2=10$、$Z_1Z_2=11$。
- **输入**：当前投入的币值，$X_1X_2=00$、币值为0；$X_1X_2=01$、币值为5角；$X_1X_2=10$、币值为1元。
- **状态**：记录已经投入的币值，$S_0=0$、$S_1=5$角、$S_2=1$元、$S_2=1.5$元、$S_2=2$元。

摩尔模型的状态图



已收0.5元状态
00
S1/00
01
00
S2/00
01
10
01
10
S0/00
00
01
10
10
01
01
10
00
S3/10
00
初始状态
S4/11
输出饮料
已收1.5元状态

摩尔模型的状态转换表

现态	次态			输出 Z_1Z_2
	$X_1X_2 = 00$	$X_1X_2 = 01$	$X_1X_2 = 10$	
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

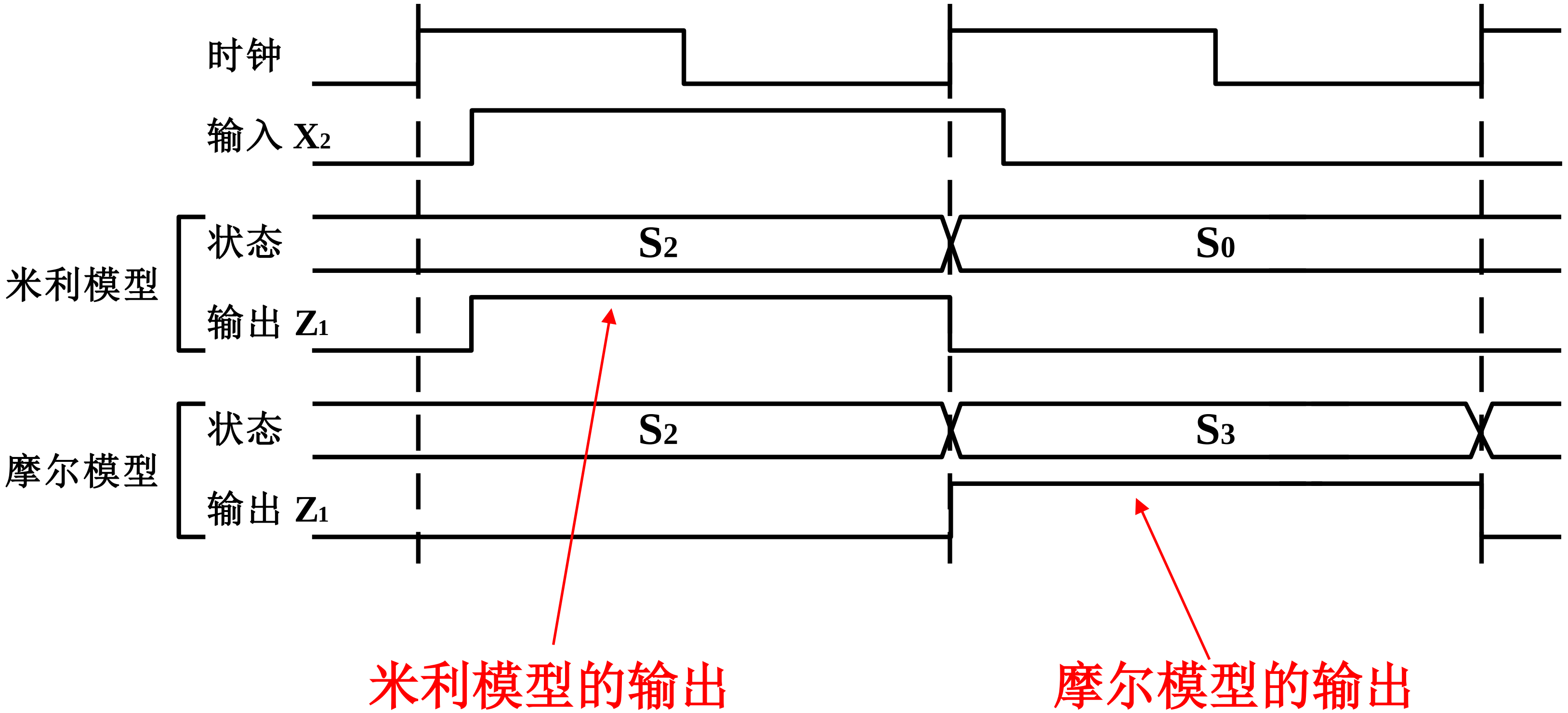
两个模型的时序图

状态：已经投入的硬币总值为1元

输入：再投入1个5角硬币

输出：一杯饮料，即Z_1 =1

X1，Z1一直为0，未标出



此处：状态变化都等下一个时钟沿，但米利型的输出提前发生变化。

两种基本模型的相互转换

1、摩尔模型转换为米利模型

将摩尔模型状态转换表的最后一列输出去掉。

在每个次态后面加上“/输出”。其中的输出对应于该次态在原模型中的输出。

观察修改后的状态转换表，合并相同的状态。

参考书p143，表4-3　**用状态转换表更方便。**

此处：什么是相同的状态？输出和次态在相同的输入下都相同。

2、米利模型转换为摩尔模型

- 输出同类状态：

  所有指向某个状态的状态转换线都具有相同的输出。

  这种类型的状态，次态和输出是统一的，所以只要将所有指向这个状态的状态转换线上的输出改写到表示状态的圆圈中，就可以将米利模型转换为摩尔模型。

参考书p144，图4-8 **用状态转换图更方便。**

- 输出非同类状态：

  指向某个状态的状态转换线具有几个不同的输出。

  显然这个状态转换成摩尔模型后将对应几个状态，所以按照下列步骤改画这种类型的状态：

  一、将此状态分成几个新状态。每个新状态对应一个输出，写在表示新状态的圆圈中。

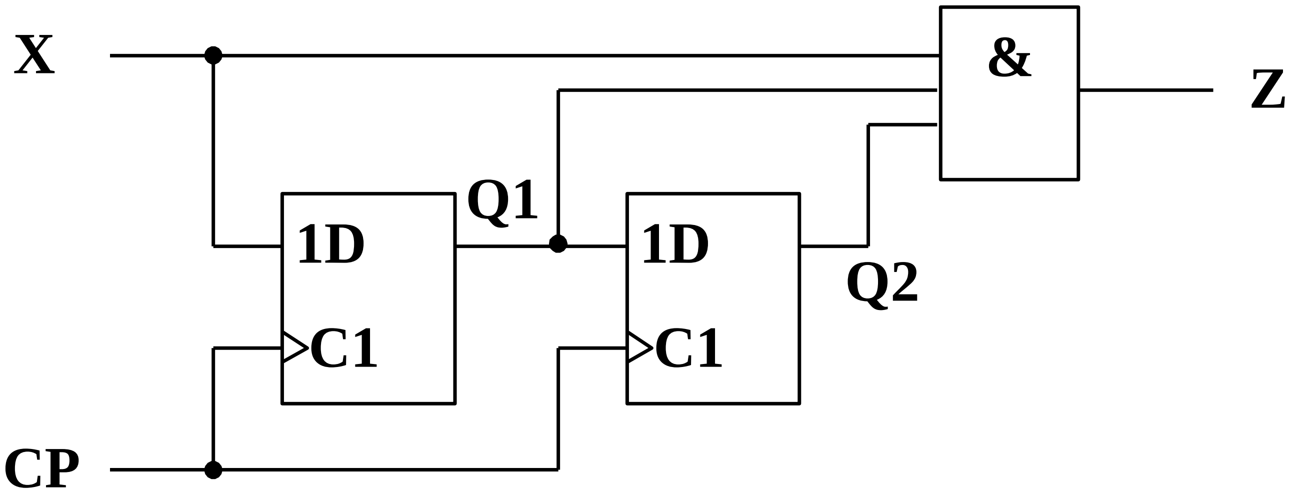
  二、按照不同的输出，将原来的状态转换线分别改画成指向具有对应输出的新状态。

  三、原来从输出非同类状态出发的所有状态转换线，都应该在每个新状态中重新画出来，并且它们的目的状态应该与原来的相同。

4.2 同步时序电路的分析

- 根据给定的电路，确定电路的类型，列出触发器的激励方程。
- 将激励方程代入触发器的特征方程，写出电路的状态方程，同时写出电路的输出方程。
- 由状态方程和输出方程，列出电路的状态转换表或状态转换图。
- 分析电路的状态转换表或状态转换图，得到电路的功能表示或者相应的时序图。如果已知电路的功能，可以通过这一步的分析，验证电路功能的正确性。

例1



米利型电路。

输出方程： $Z = XQ_1Q_2$

D触发器： $Q_1^{n+1} = D_1, \quad Q_2^{n+1} = D_2$

激励方程： $D_1 = X, \quad D_2 = Q_1$

次态方程： $Q_1^{n+1} = X, \quad Q_2^{n+1} = Q_1$

$$Z = XQ_1Q_2$$

$$Q_1^{n+1} = D_1, \quad Q_2^{n+1} = D_2$$

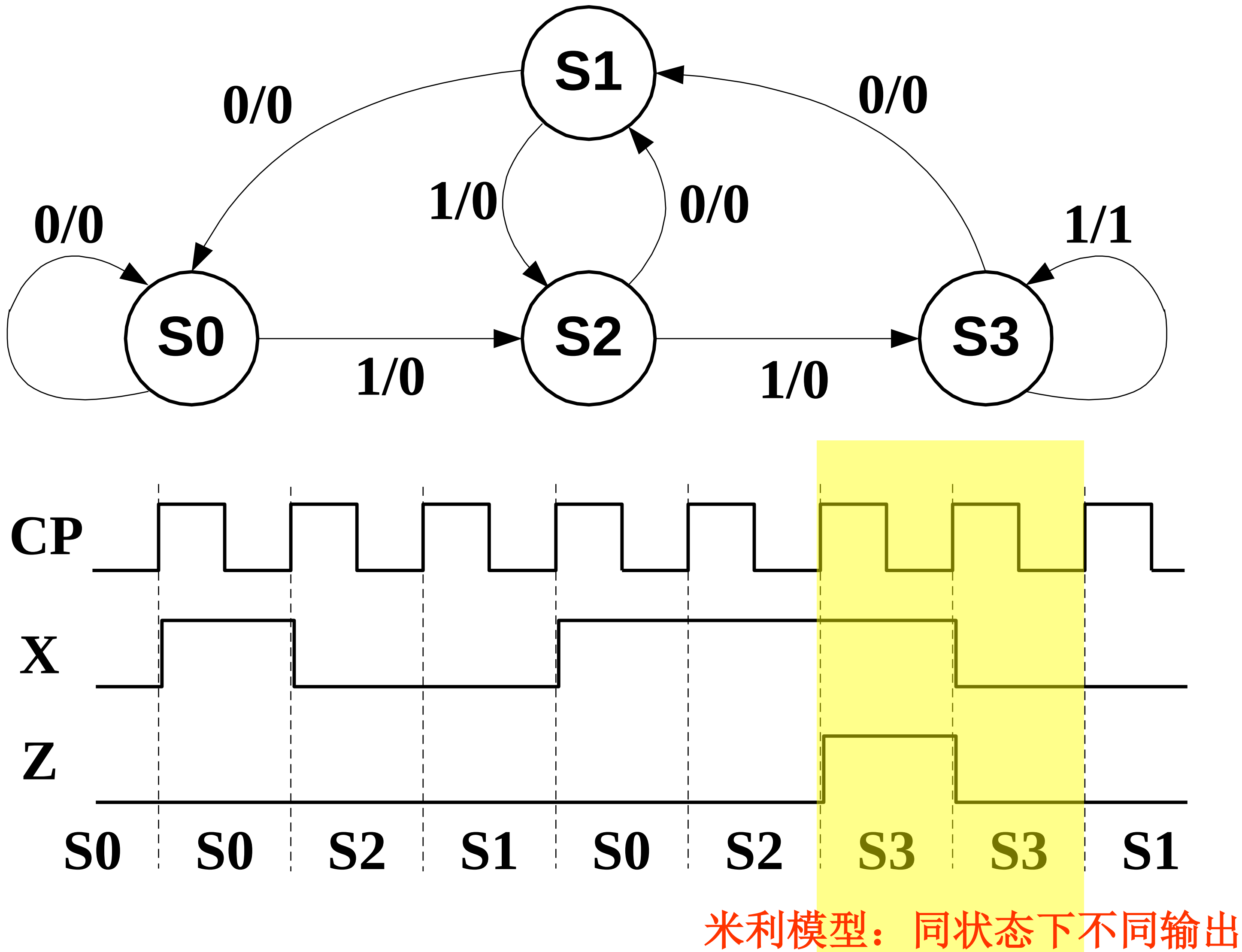
$$D_1 = X, \quad D_2 = Q_1$$

$$Q_1^{n+1} = X, \quad Q_2^{n+1} = Q_1$$

状态转换表

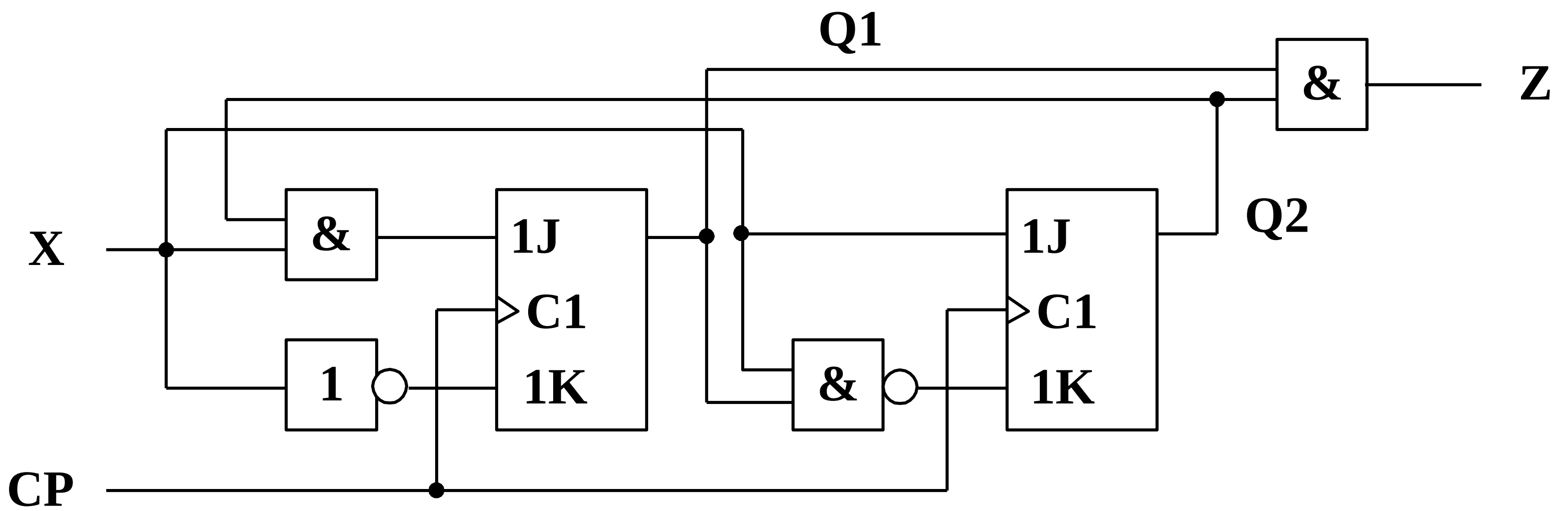
现态	次态/输出	次态/输出	状态编号
Q_1Q_2	X=0	X=1	
00	00/0	10/0	S0
01	00/0	10/0	S1
10	01/0	11/0	S2
11	01/0	11/1	S3

状态转换图和时序图



此处：米利型输出与状态不同步的问题

例2



摩尔型电路。

输出方程： $Z = Q_1 Q_2$

JK触发器： $Q^{n+1} = J\overline{Q} + \overline{K}Q$

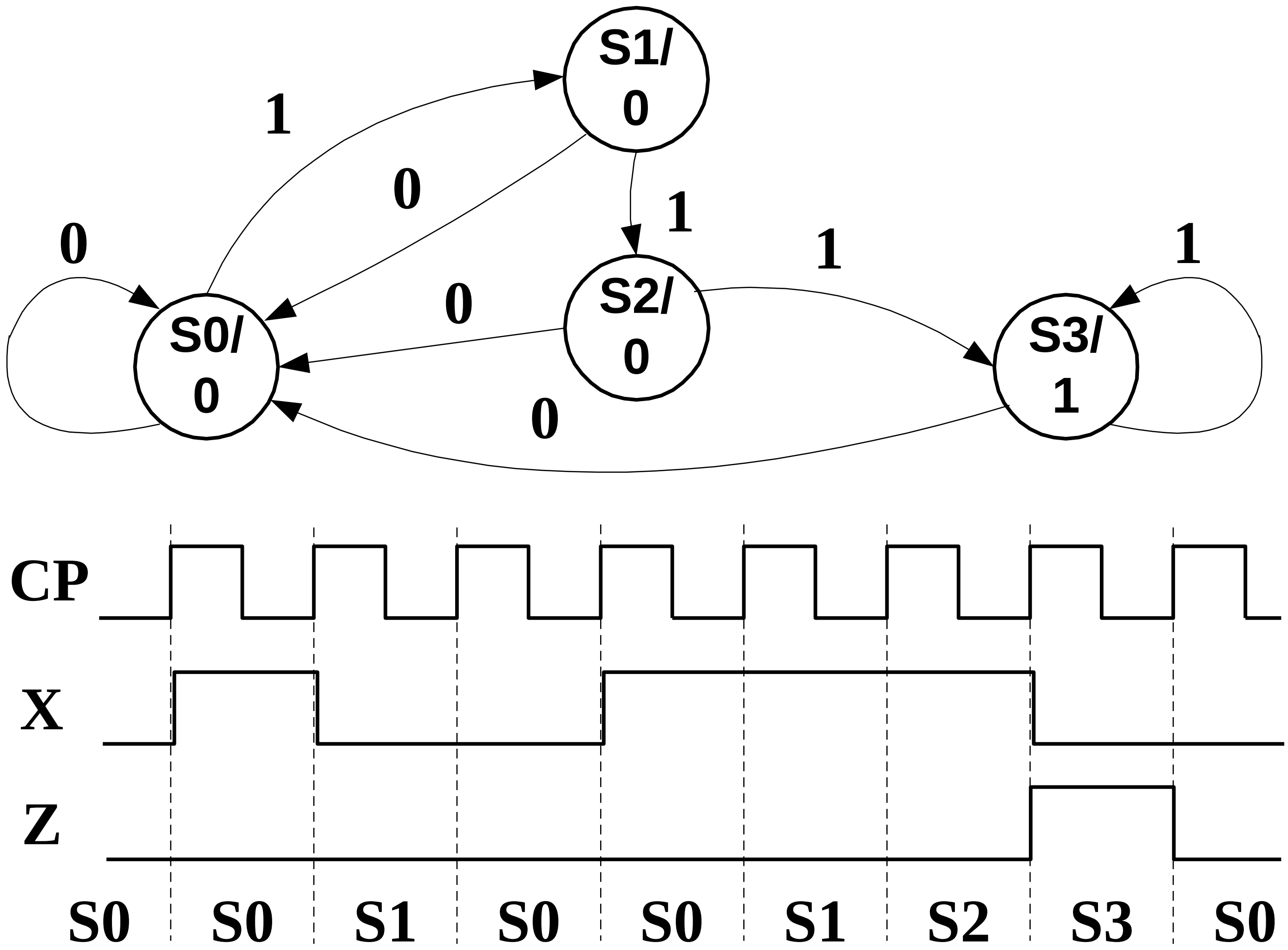
激励方程： $J_1 = XQ_2$， $K_1 = \overline{X}$， $J_2 = X$， $K_2 = \overline{XQ_1}$

次态方程： $Q_1^{n+1} = XQ_2\overline{Q}_1 + XQ_1$， $Q_2^{n+1} = X\overline{Q}_2 + XQ_1Q_2$

状态转换表

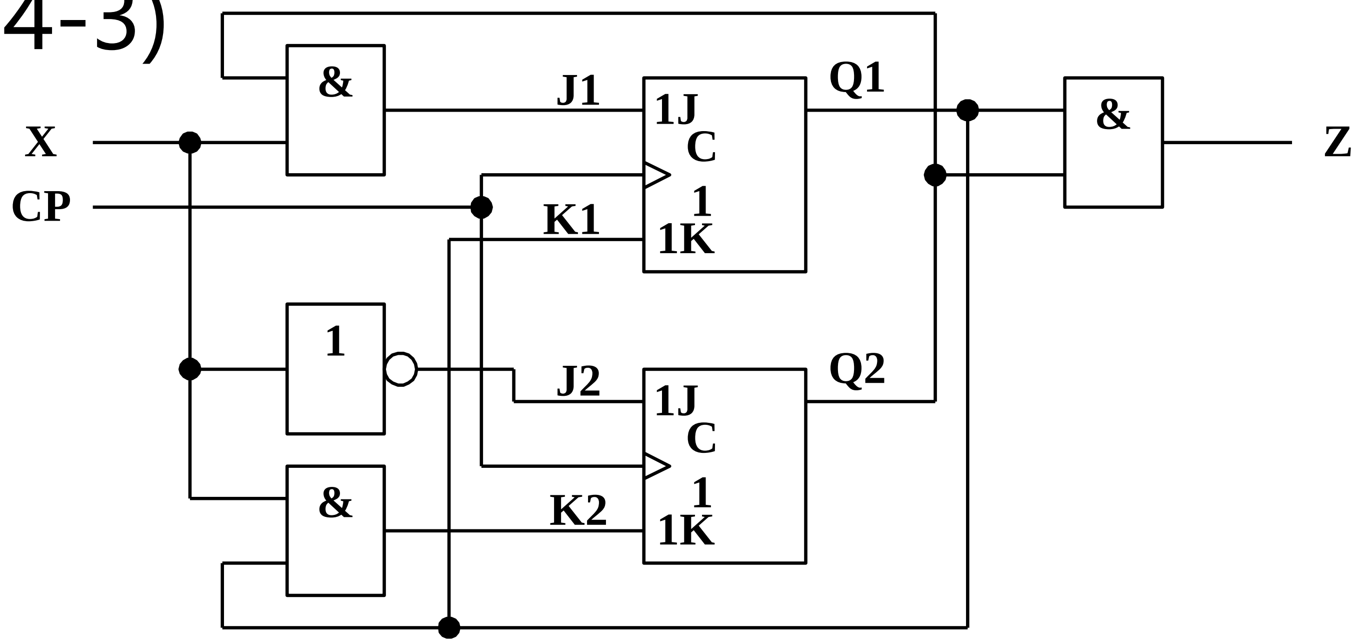
现态	次态		输出	编号
Q_1Q_2	X=0	X=1	Z	
00	00	01	0	S0
01	00	10	0	S1
10	00	11	0	S2
11	00	11	1	S3

状态转换图和时序图



此处：功能是什么？

例3(P146, 4-3)



摩尔型电路。

输出方程: $Z = Q_1 Q_2$

JK触发器: $Q^{n+1} = J\overline{Q} + \overline{K}Q$

激励方程: $J_1 = XQ_2$, $K_1 = Q_1$, $J_2 = \overline{X}$, $K_2 = XQ_1$

次态方程: $Q_1^{n+1} = XQ_2\overline{Q}_1$, $Q_2^{n+1} = \overline{X}\,\overline{Q}_2 + \overline{XQ_1}\,Q_2$

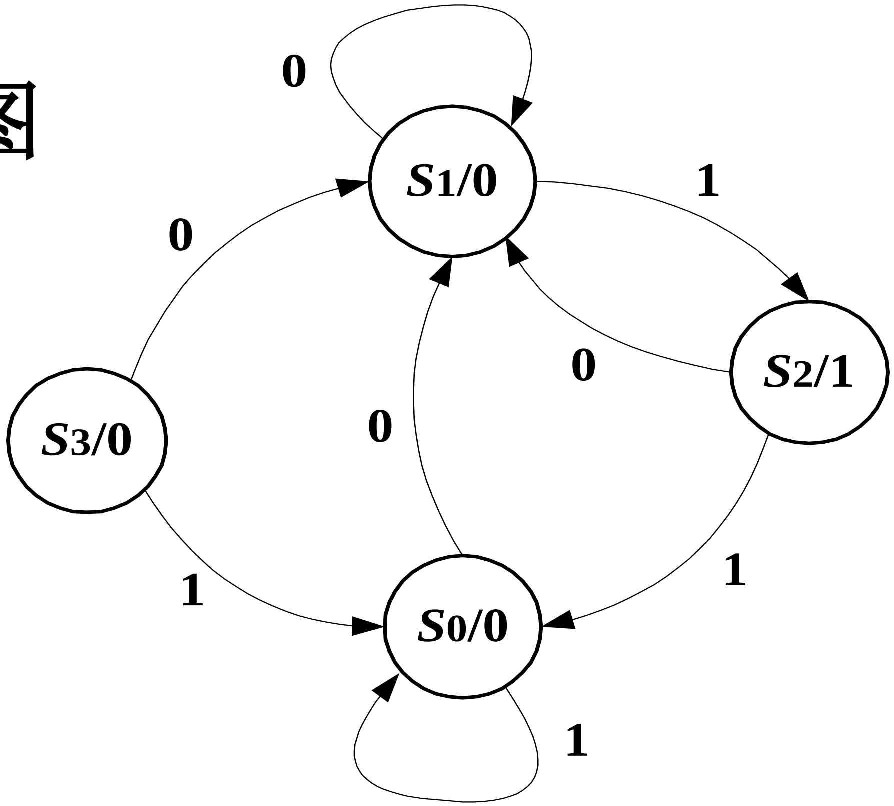
状态转换图和时序图

S0： Q_1Q_2=00

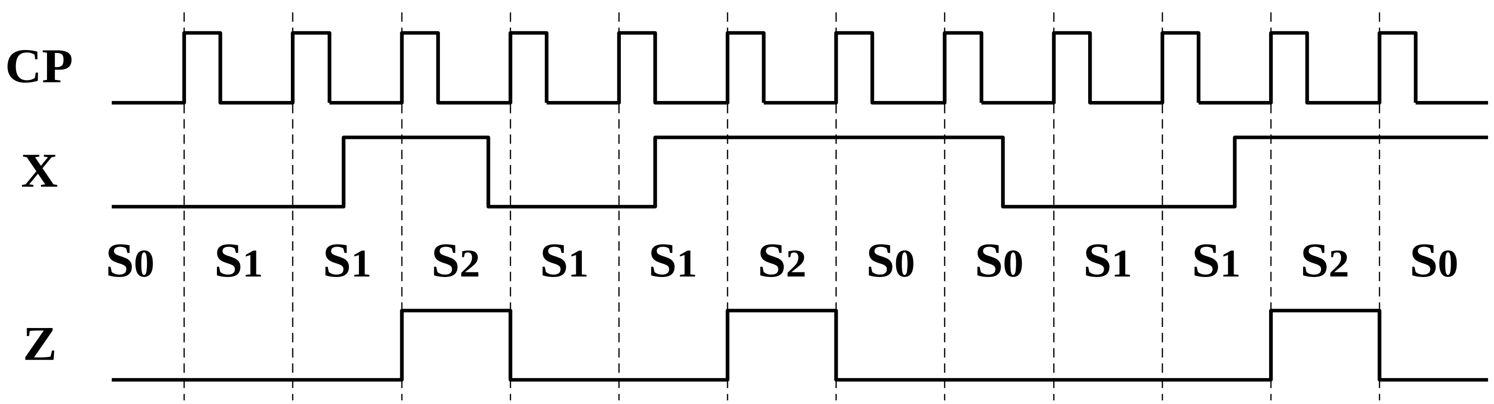
S1： Q_1Q_2=01

S2： Q_1Q_2=11

S3： Q_1Q_2=10

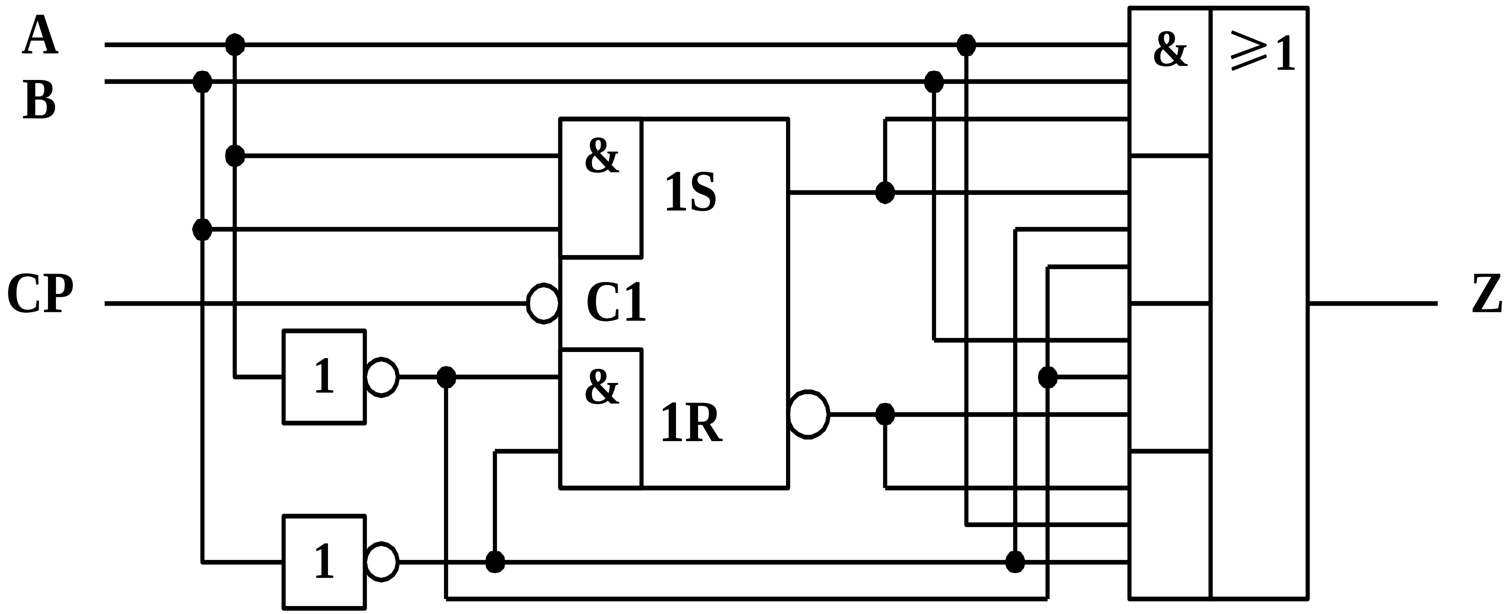


这里S3是多余状态，只出不进；
但一旦进入S3，也能进入有效循环。



此处：需要画状态转换表。有多余状态的要多加注意

P148 例4-4　串行加法器



激励方程

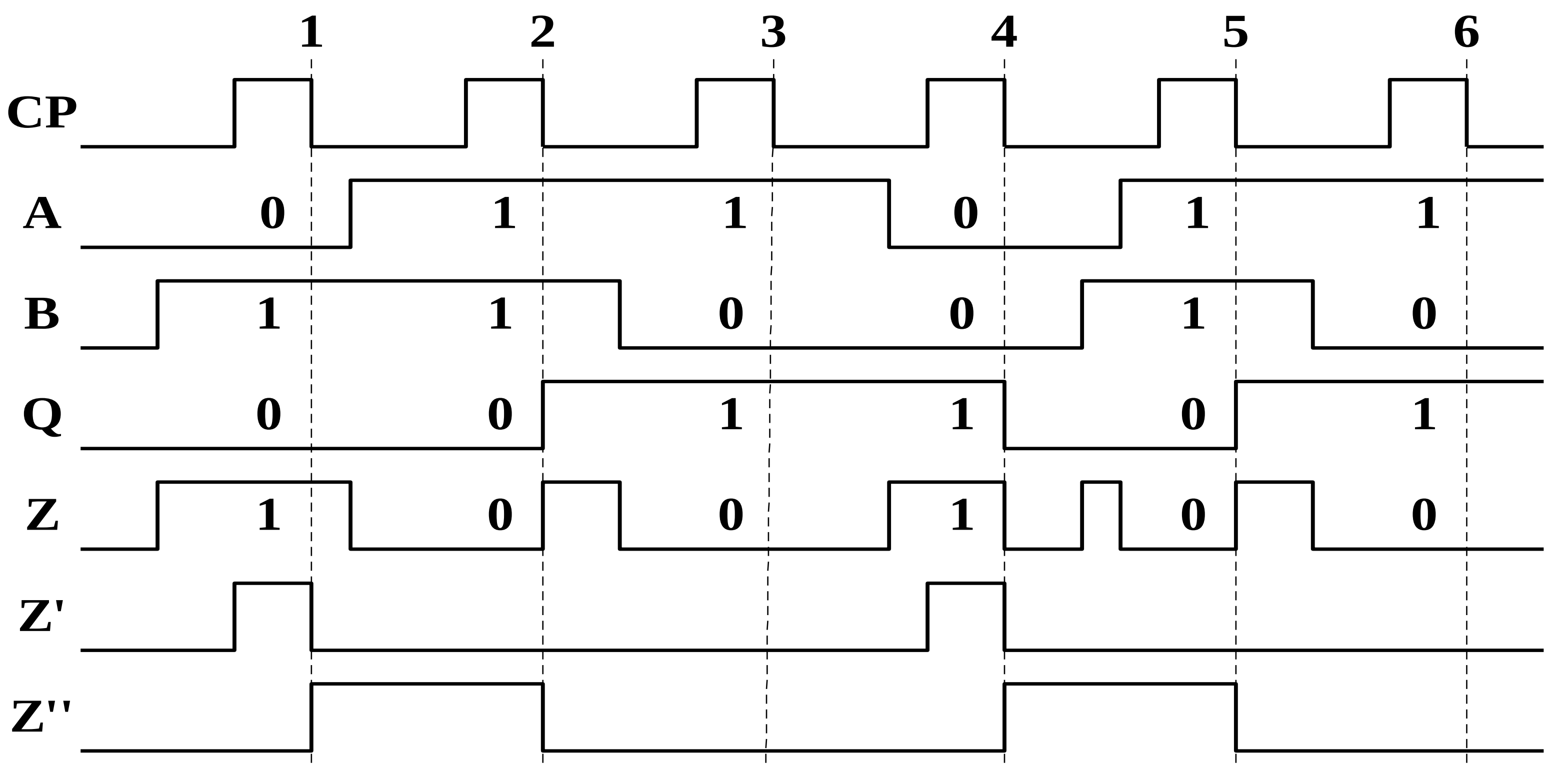
$$S = AB$$

$$R = \overline{A}\,\overline{B}$$

状态方程、输出方程和时序图

$$Q_{n+1} = S + \overline{R}Q_n = AB + (A + B)Q_n = AB + AQ_n + BQ_n$$

$$Z = Q_n(AB + \overline{A}\,\overline{B}) + \overline{Q}_n(A\overline{B} + A\overline{B}) = Q_n \oplus A \oplus B$$



- 一般来说，如果输入信号整个时钟周期内稳定，那么同步时序电路采用米利模型较为有利，因为米利模型状态少、输出快。若不能保证输入信号在整个时钟周期内保持稳定，采用摩尔模型比较有利。

- 其它例子，参考书P150例4-5

常见的同步时序电路分析

1、计数器类电路

4位二进制同步加法计数器

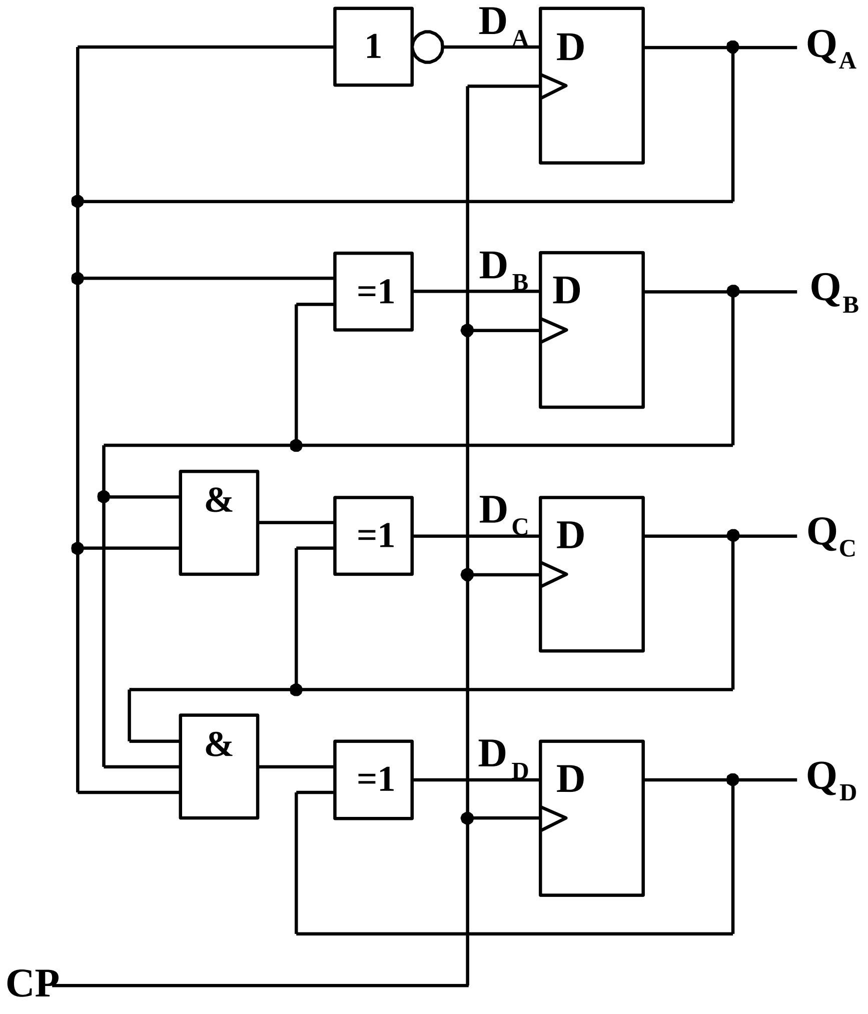
状态方程：

$$Q_{A(n+1)} = \overline{Q}_A$$

$$Q_{B(n+1)} = Q_B \oplus Q_A$$

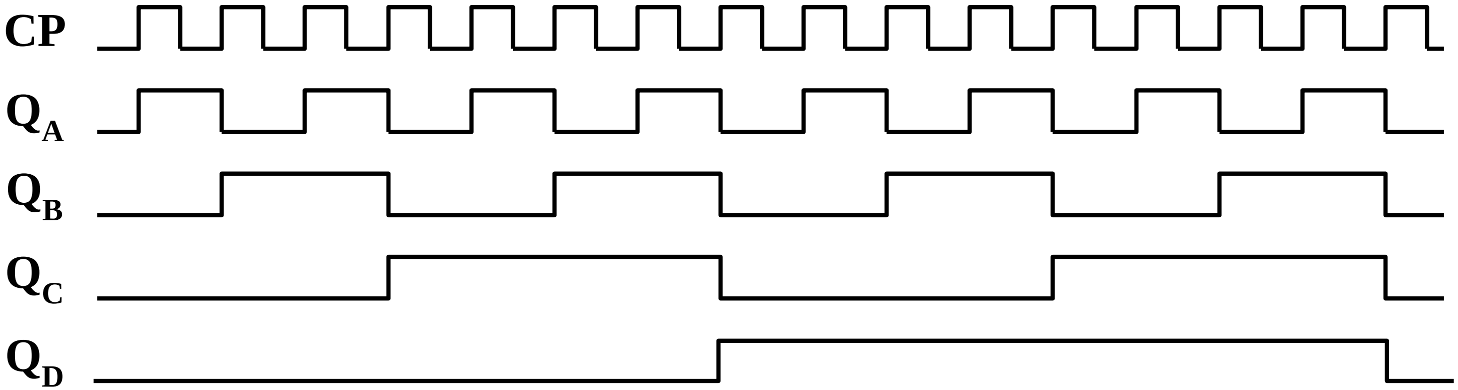
$$Q_{C(n+1)} = Q_C \oplus Q_A Q_B$$

$$Q_{D(n+1)} = Q_D \oplus Q_A Q_B Q_C$$



画一下状态转换表

时序图



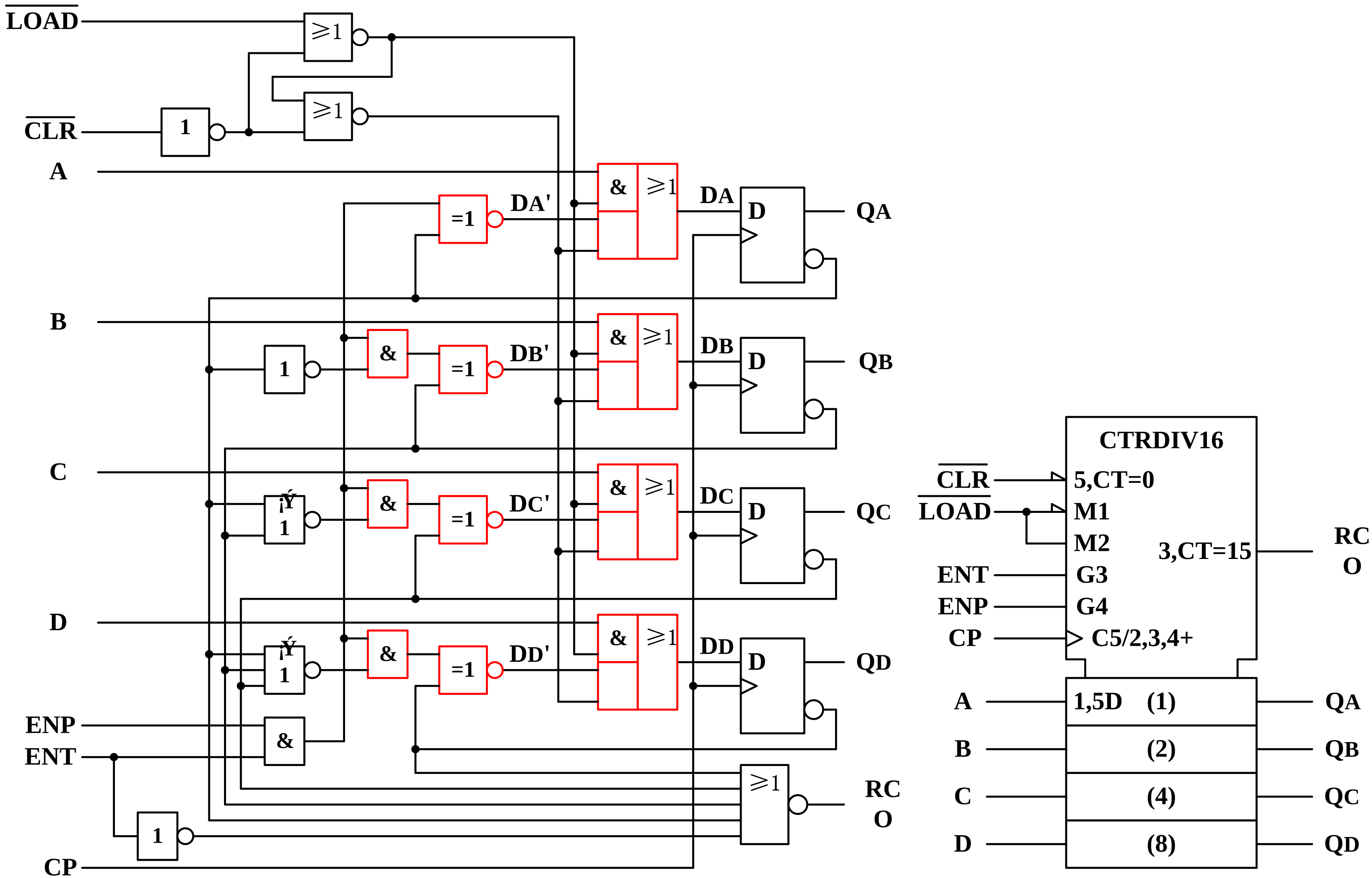
二进制同步加法计数器的状态方程的一般形式

$$Q_{0(n+1)} = \overline{Q}_0$$

$$Q_{i(n+1)} = Q_i \oplus \left(\prod_{j=0}^{i-1} Q_j\right), \quad i \neq 0$$

与行波计数器的区别？同步vs异步，电路简单程度

带同步置数、同步复位、保持等多种功能的4位二进制同步加法计数器 (P152)



利用与或门作为数据选择器实现多种逻辑功能转换

$$D_n = LOAD \cdot \overline{CLR} \cdot n + \overline{LOAD} \cdot \overline{CLR} \cdot D_n'$$

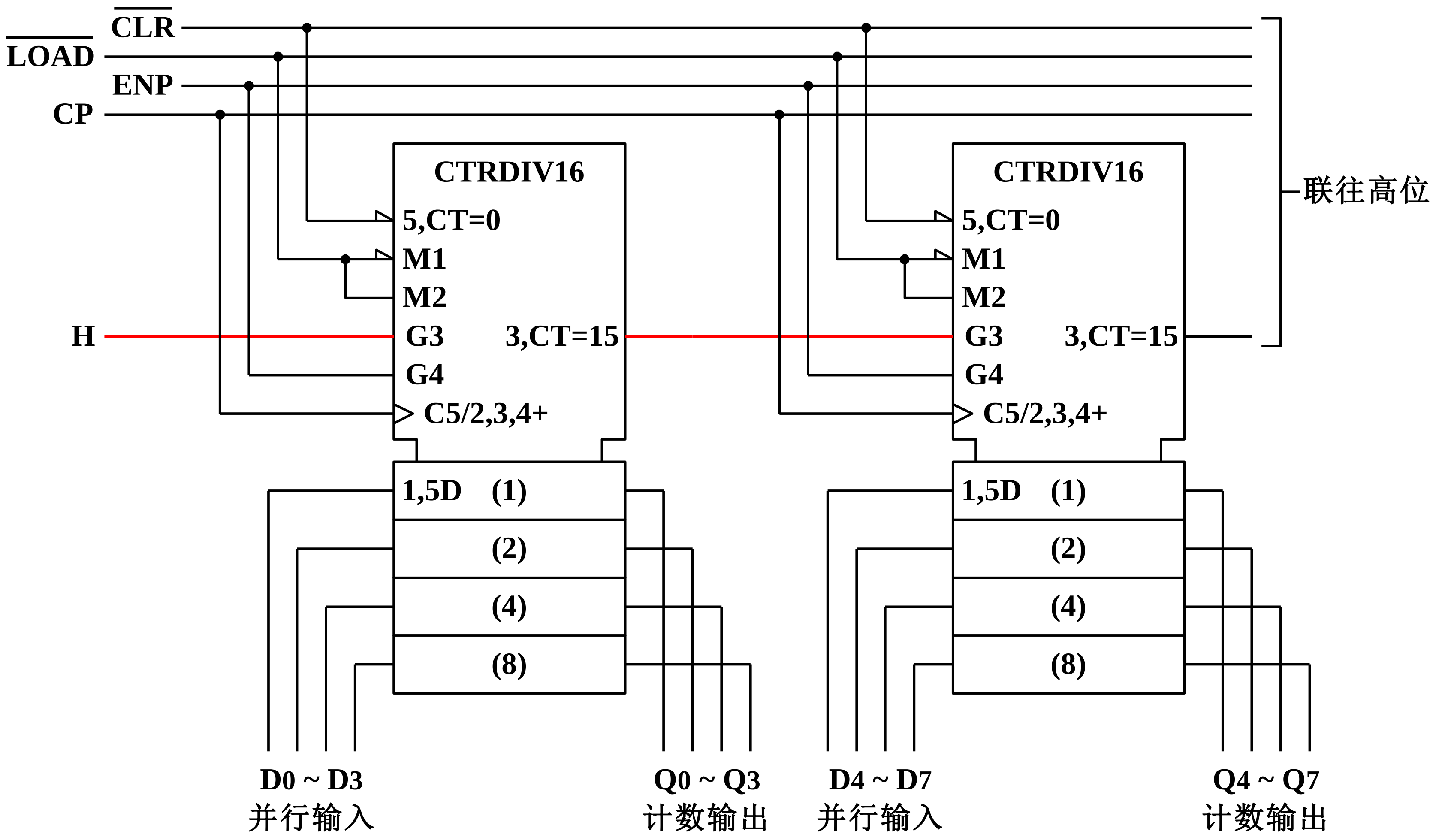
$$D_A' = (ENP \cdot ENT) \square\ \overline{Q}_A$$

$$D_n' = [(ENP \cdot ENT) \cdot (\prod_{j=0}^{i-1} Q_j)] \square\ \overline{Q}_n, \quad n \neq A$$

$\overline{CLR}$	$\overline{LOAD}$	ENP	ENT	功能
0	X	X	X	复位（清零）
1	0	X	X	加载（置数）
1	1	1	1	计数
1	1	0	X	保持
1	1	X	0	保持

回顾数据选择器的逻辑表达式

二进制加法计数器的串联



二进制减法计数器

状态方程

$$Q_{A(n+1)} = \overline{Q}_A$$

$$Q_{B(n+1)} = Q_B \oplus \overline{Q}_A$$

$$Q_{C(n+1)} = Q_C \oplus \overline{Q}_A \overline{Q}_B$$

$$Q_{D(n+1)} = Q_D \oplus \overline{Q}_A \overline{Q}_B \overline{Q}_C$$

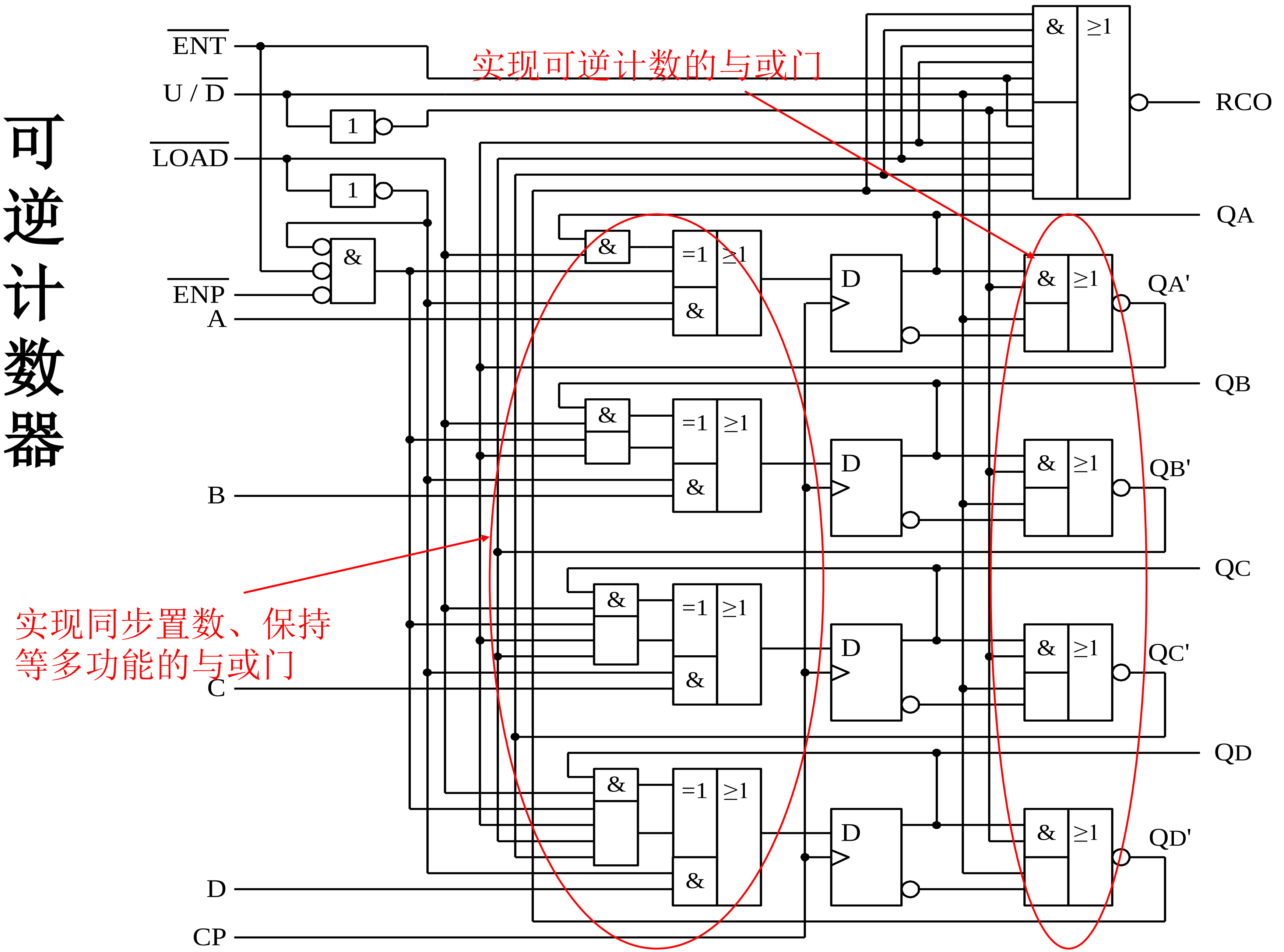
二进制减法计数器的状态方程的一般形式

$$Q_{0(n+1)} = \overline{Q}_0$$

与加法计数器的差别仅在于这个“非”

$$Q_{i(n+1)} = Q_i \oplus \left(\prod_{j=0}^{i-1} \overline{Q}_j\right), \quad i \neq 0$$

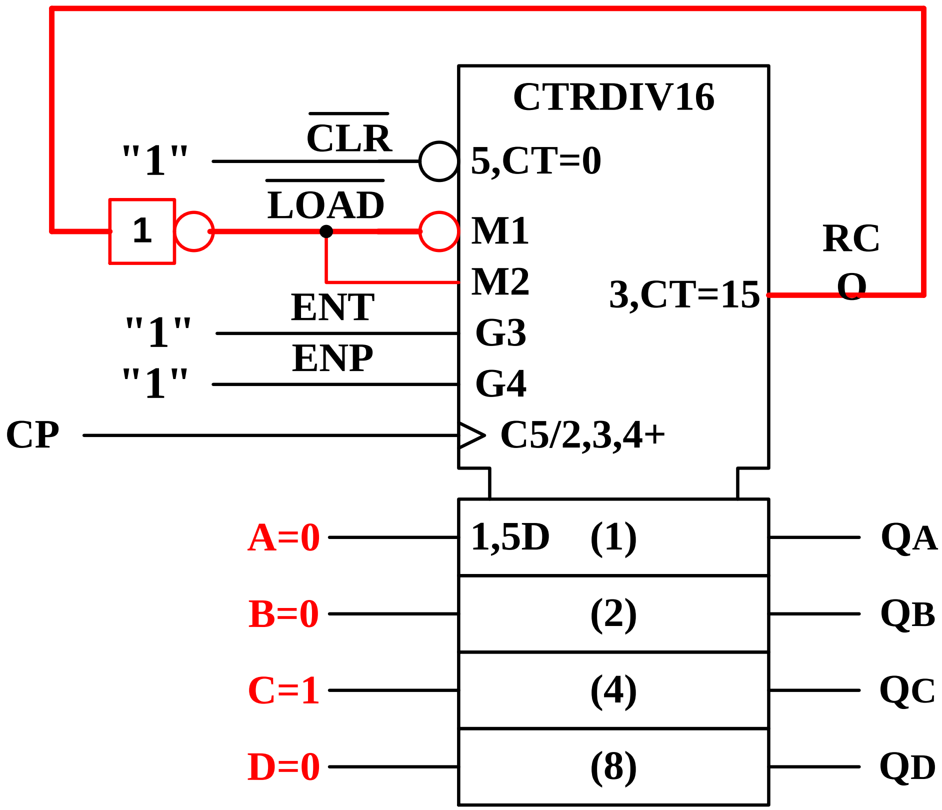
可逆计数器


ENT
U / D
LOAD
ENP
A
B
C
D
CP
RCO
QA
QB
QC
QD
QA'
QB'
QC'
QD'
实现可逆计数的与或门
实现同步置数、保持
等多功能的与或门

P158, 例4-6

还有没有其他方式实现？

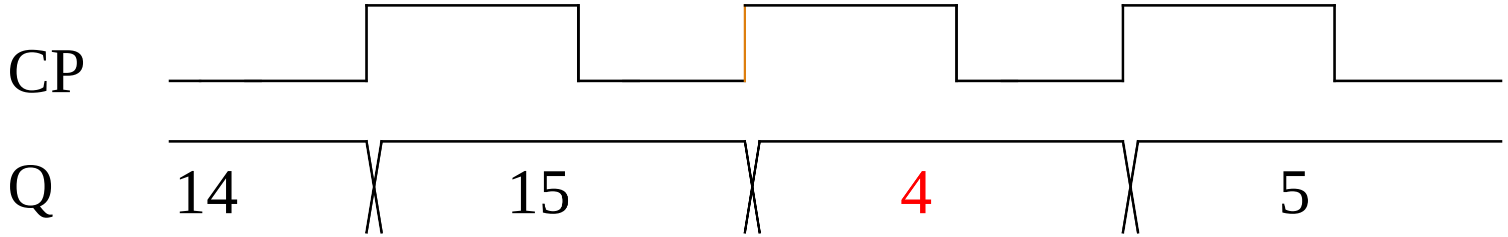
用二进制同步计数器构成其他进制计数器



A、同步置数法

利用进位信号进行同步置数，跳过若干状态。

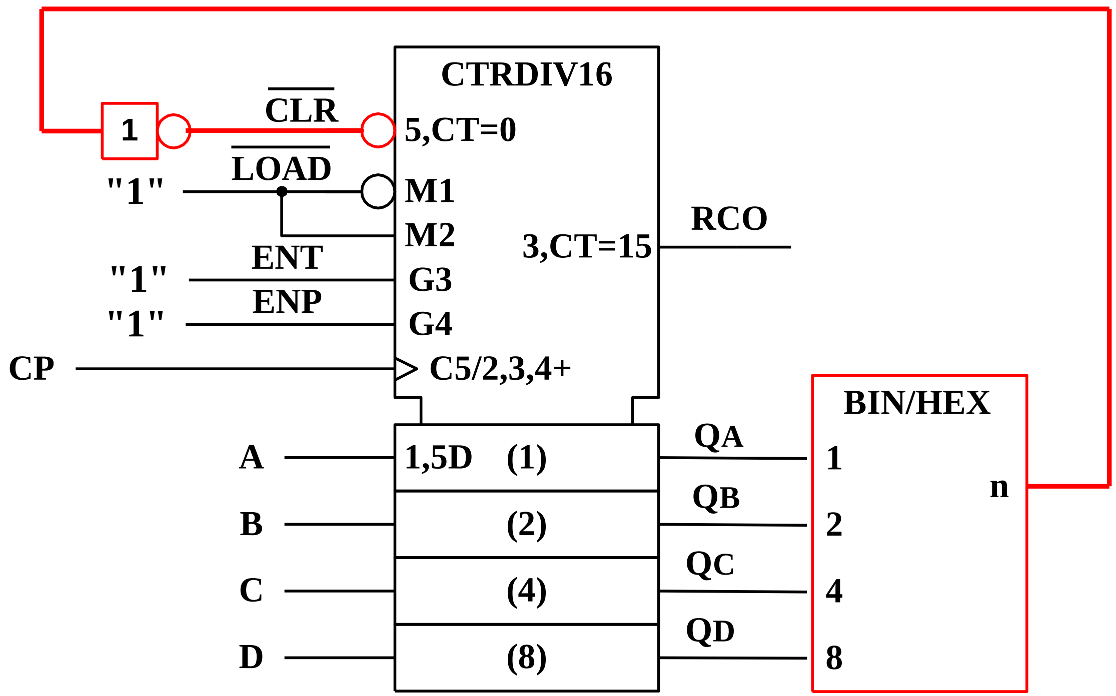
例如 **DCBA＝0011＝3，** 则有以下时序图：

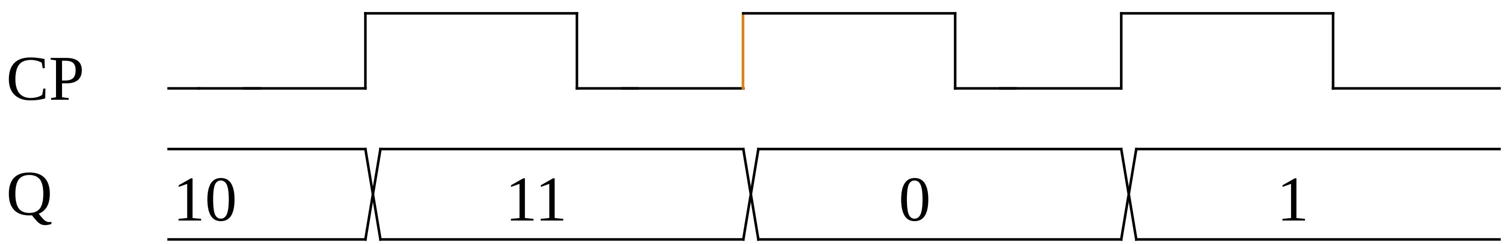


B、同步复位法

利用输出译码进行同步复位，跳过若干状态。

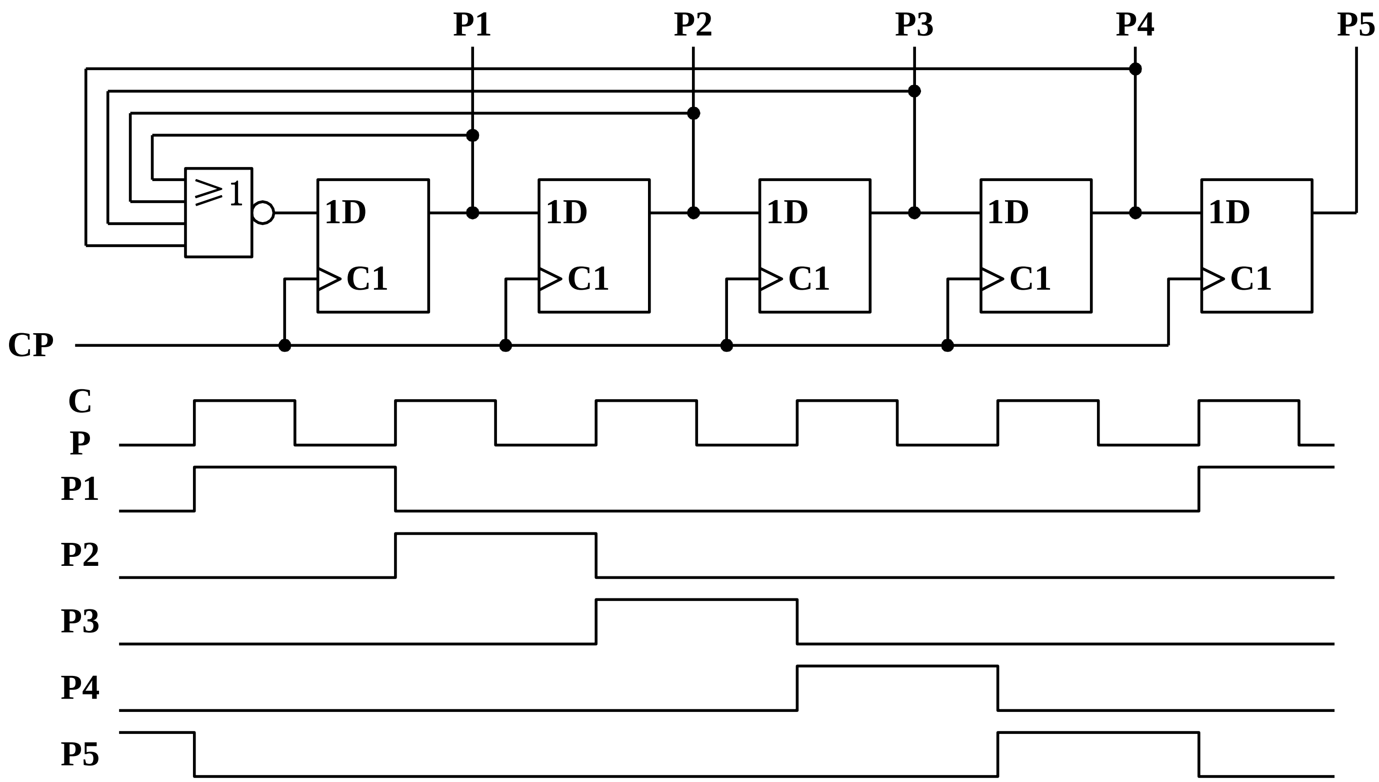
例如 n＝11，则有以下时序图：





2、移位寄存器类电路

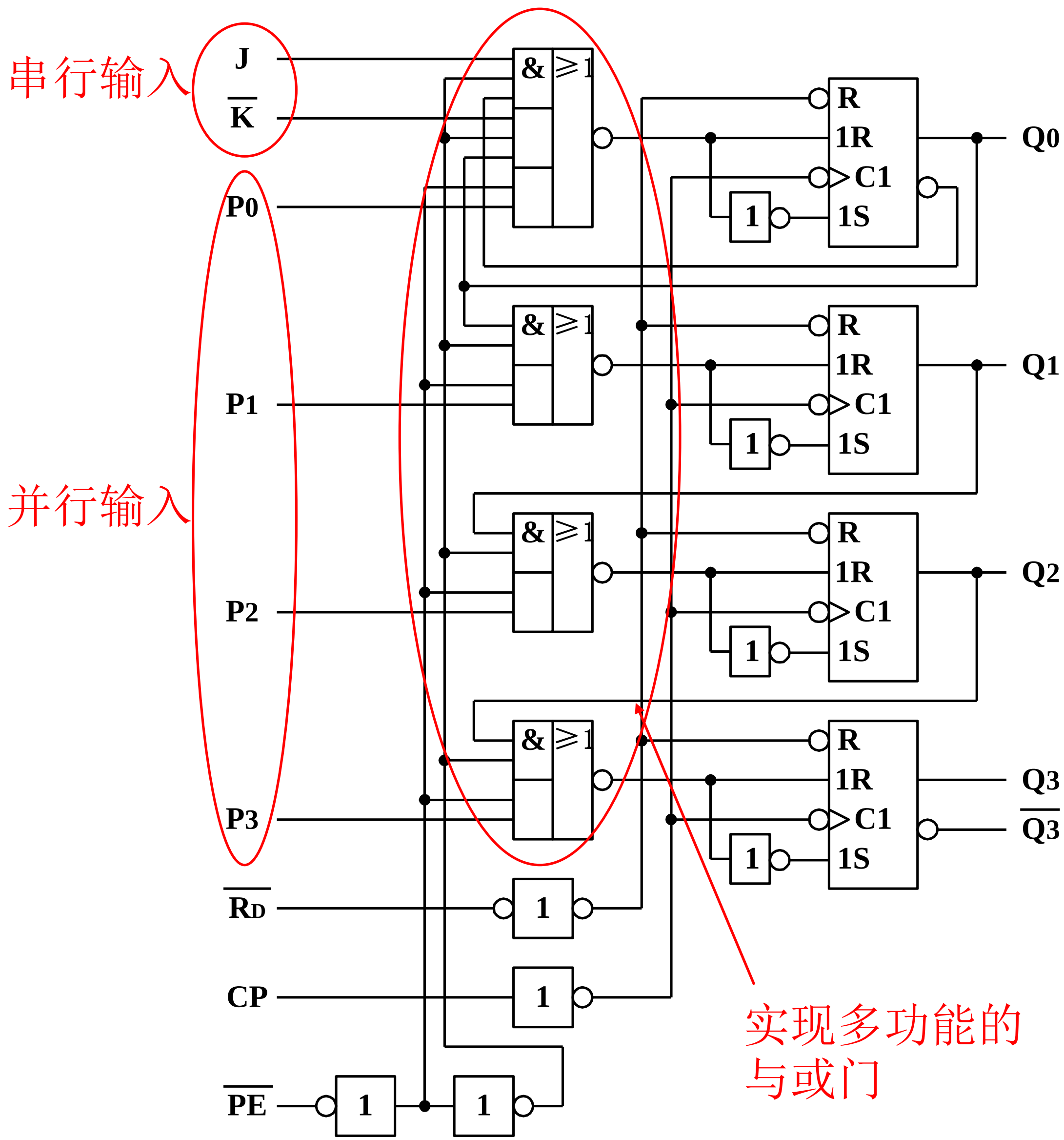
单活跃（One-hot）电路

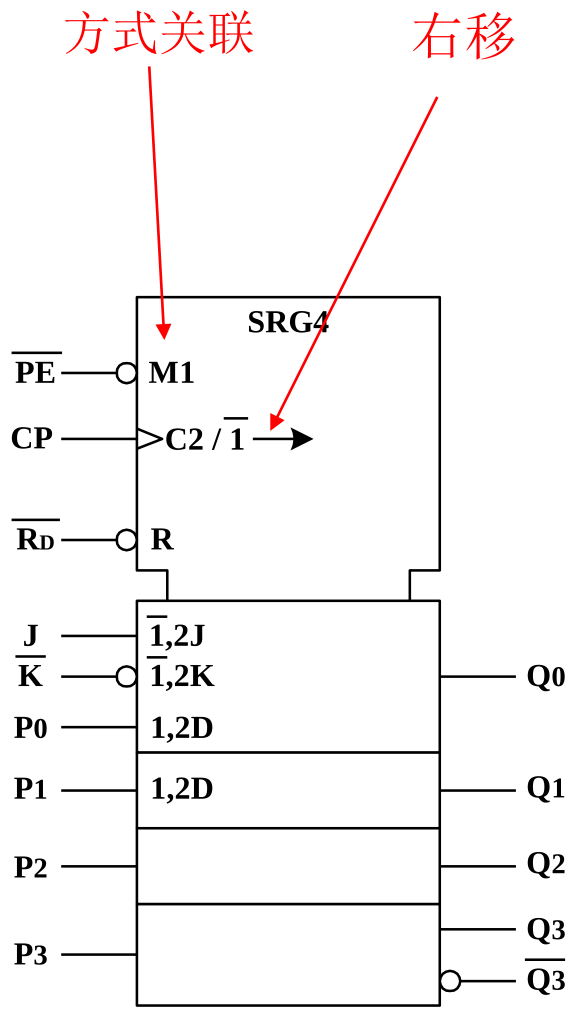


单活跃电路常常用来作为状态机的状态输出

对比第三章的环形计数器，需要置数。此处有冗余态，但能恢复。

通用移位寄存器

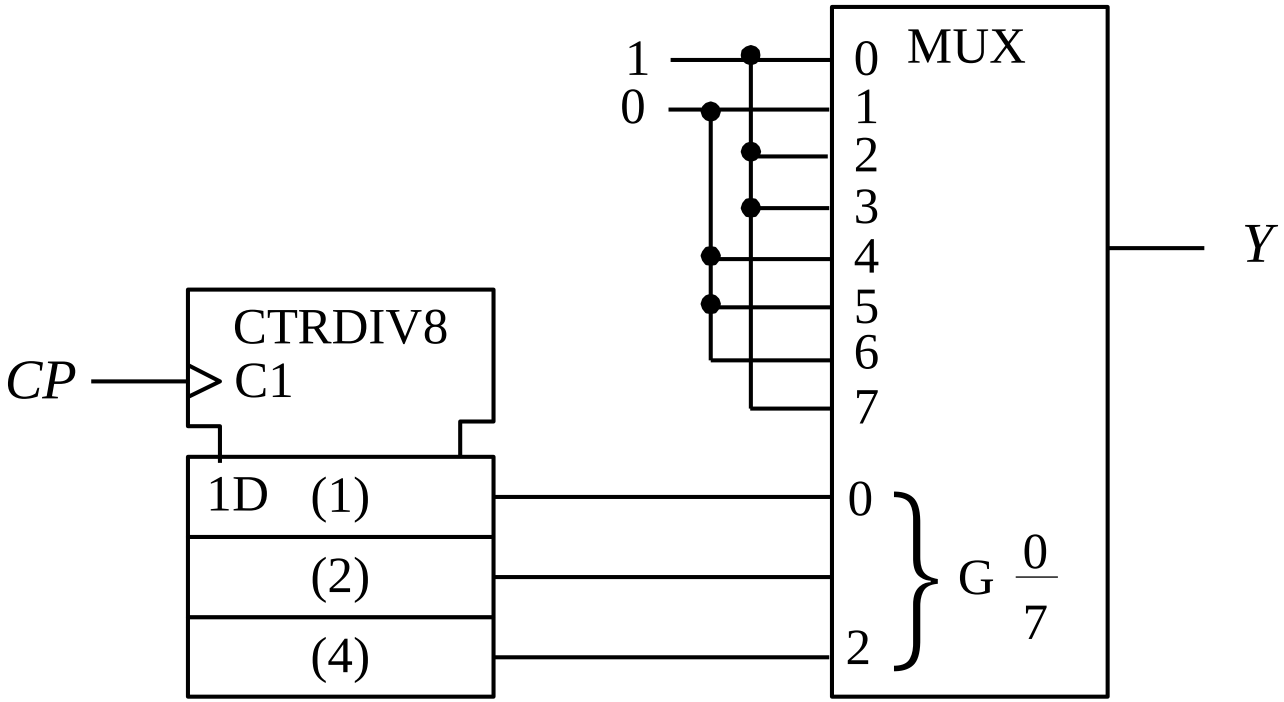




ＰＥ作为串行、并行的选择控制。

由同步时序电路模块构成的电路举例

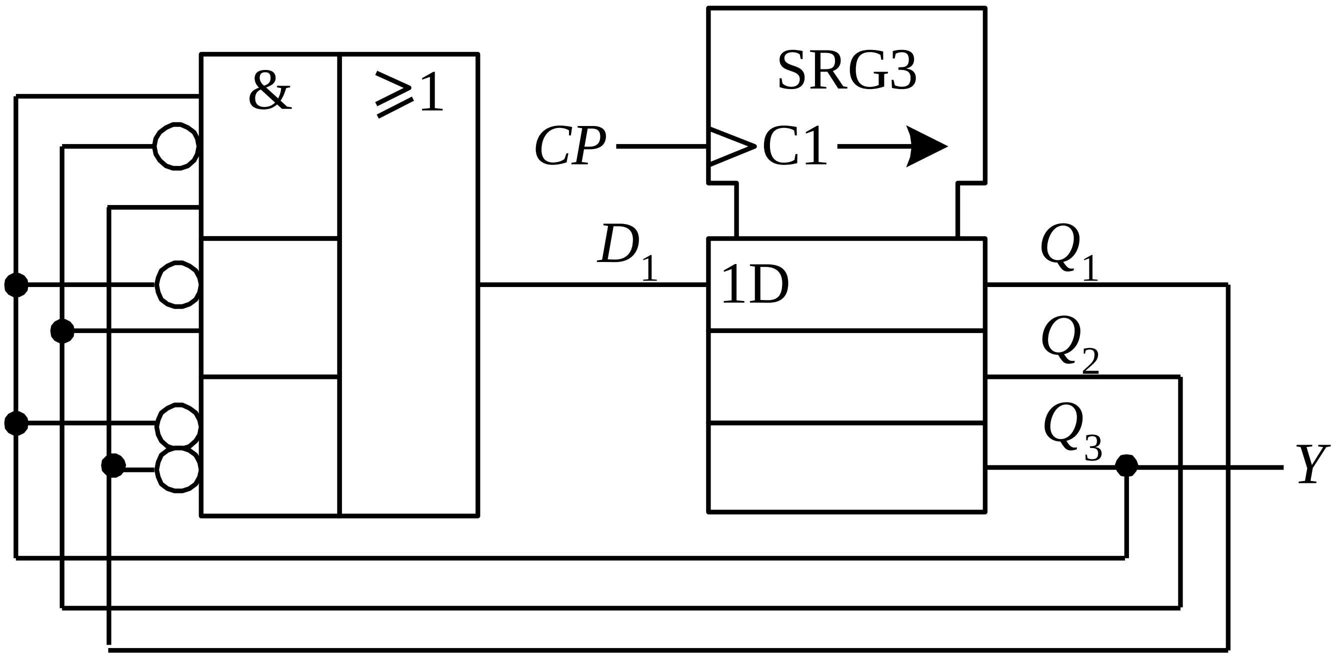
基于计数器的序列信号发生器



本例产生序列为10110001

由同步时序电路模块构成的电路举例

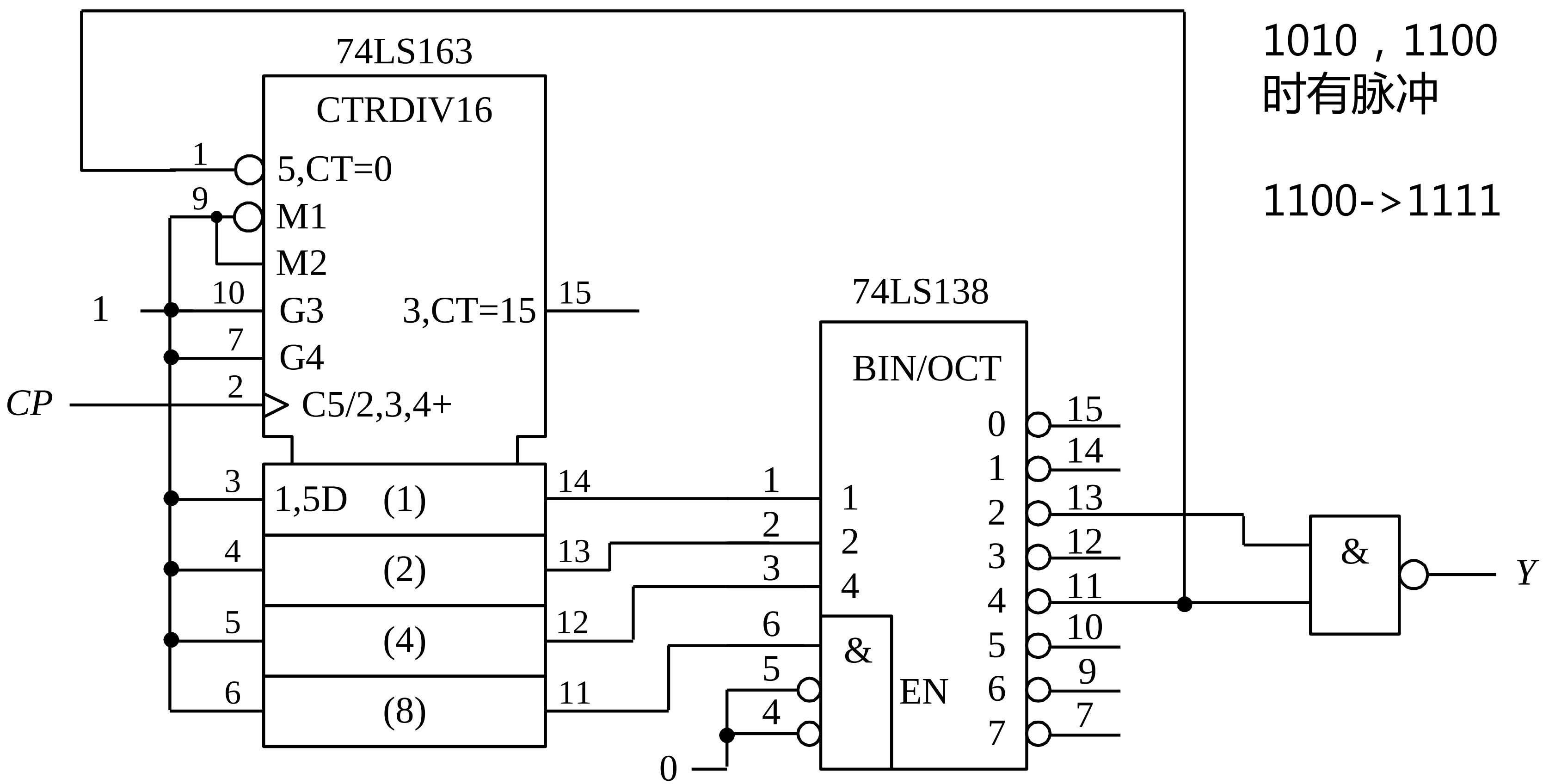
基于移位寄存器的序列信号发生器



请问产生的是什么序列？

由同步时序电路模块构成的电路举例

基于计数器的脉冲发生器



1010，1100
时有脉冲

1100->1111

4.3 同步时序电路设计

设计一个同步时序电路的一般步骤

✓ 分析要求，描述状态机

- 确定采用何种模型
- 状态转换表或状态转换图
- 化简冗余状态

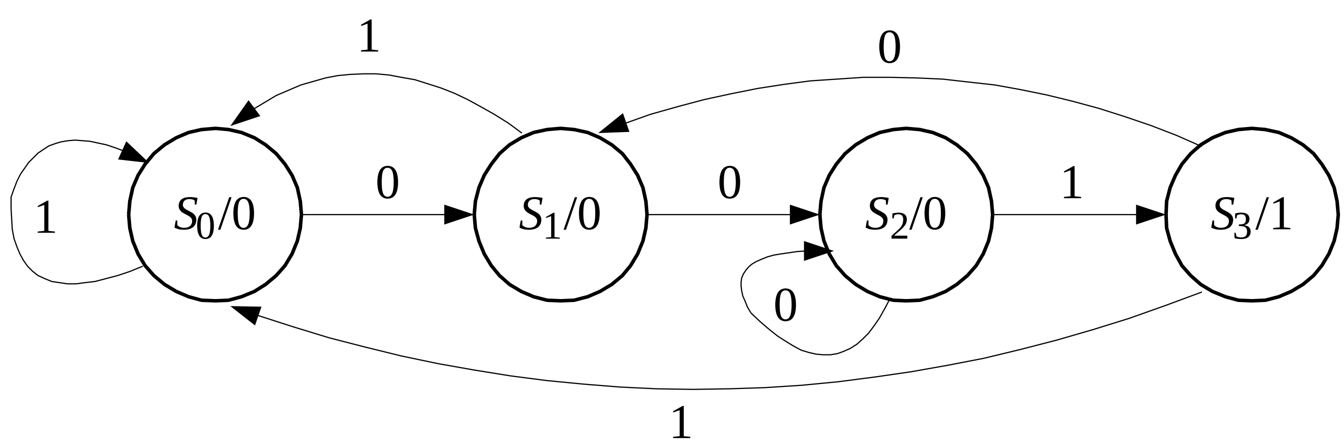
✓ 用电路实现状态机

- 状态编码
- 确定触发器类型，状态激励表
- 触发器的激励方程，电路的输出方程
- 最终的逻辑电路图

例4-7(p165)：设计外部事件检测电路

问题：在每个时钟脉冲的有效边沿检测输入信号。输入信号通常为1，当它变为0时表示有外部事件发生。为了避免由于干扰而发生虚假输出，要求至少检测到2个连续的0才认为输入有效。在检测到有效输入后，当输入重新回到1后输出一个逻辑1信号并将该信号保持一个时钟周期。

采用摩尔模型，状态转换图：



状态编码如下：S_0= 00，S_1= 01，S_2= 10，S_3= 11

状态转换表

状态	状态（编码）	次态（编码）		输出
		$X=0$	$X=1$	
S_0	$Q_1Q_0=00$	01	00	0
S_1	$Q_1Q_0=01$	11	00	0
S_2	$Q_1Q_0=11$	11	10	0
S_3	$Q_1Q_0=10$	01	00	1

采用JK触发器的状态激励表

输入	激励 J_1K_1，J_0K_0			
	$Q_1Q_0=00$	$Q_1Q_0=01$	$Q_1Q_0=11$	$Q_1Q_0=10$
X = 0	0d，1d	1d，d0	d0，d0	d1，1d
X = 1	0d，0d	0d，d1	d0，d1	d1，0d

熟练J K、D触发器的激励表。

激励函数

X \ Q_1Q_0	00	01	11	10
0	0	1	d	d
1	0	0	d	d

J_1

X \ Q_1Q_0	00	01	11	10
0	d	d	0	1
1	d	d	0	1

K_1

X \ Q_1Q_0	00	01	11	10
0	1	d	d	1
1	0	d	d	0

J_0

X \ Q_1Q_0	00	01	11	10
0	d	0	0	d
1	d	1	1	d

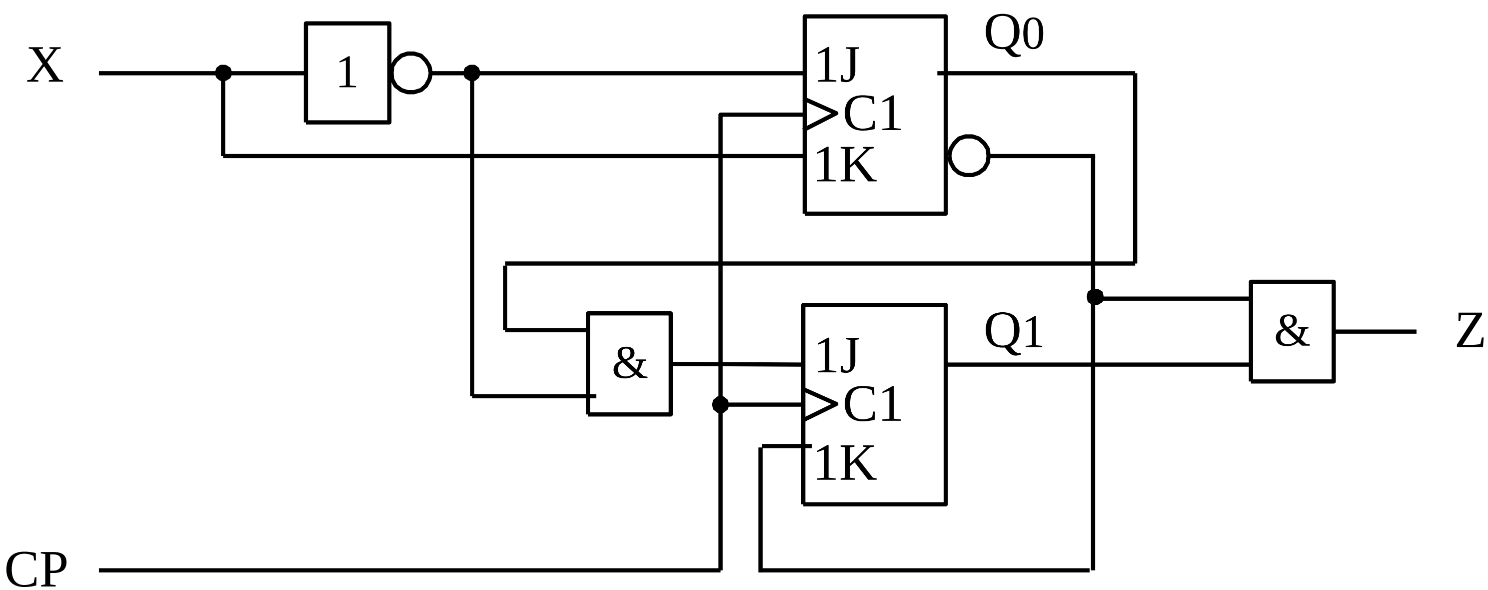
K_0

$$J_1 = \overline{X}Q_0, \quad K_1 = \overline{Q}_0, \quad J_0 = \overline{X}, \quad K_0 = X$$

输出函数

$$Z = Q_1\overline{Q}_0$$

逻辑图



例2：设计8进制计数器

状态转换表

状态	$Q_2Q_1Q_0$	$Q_{2(n+1)}Q_{1(n+1)}Q_{0(n+1)}$
S0	000	001
S1	001	010
S2	010	011
S3	011	100
S4	100	101
S5	101	110
S6	110	111
S7	111	000

8 个状态。激励表对于无输入的画法，直接画在表上。

采用D触发器的激励卡诺图

Q2 \ Q1Q0	00	01	11	10
0	0	0	1	0
1	1	1	0	1

D2

Q2 \ Q1Q0	00	01	11	10
0	0	1	0	1
1	0	1	0	1

D1

Q2 \ Q1Q0	00	01	11	10
0	1	0	0	1
1	1	0	0	1

D0

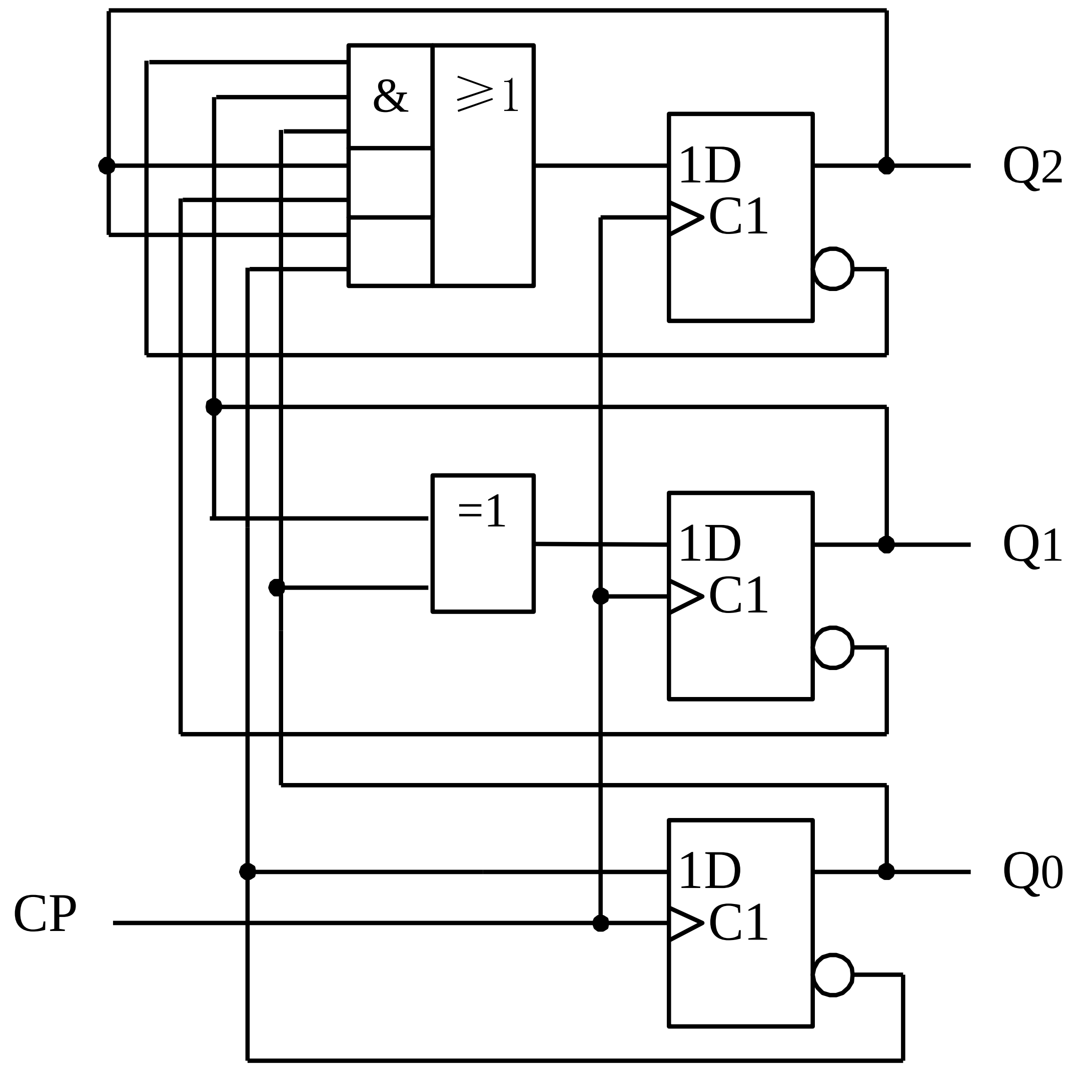
激励方程

$$D_2 = \overline{Q}_2 Q_1 Q_0 + Q_2 \overline{Q}_1 + Q_2 \overline{Q}_0$$

$$D_1 = Q_1 \oplus Q_0$$

$$D_0 = \overline{Q}_0$$

逻辑图



此例：状态全用上，没有冗余状态。

带有冗余状态的同步时序电路设计

一个具有n个状态的同步时序电路，如果n不是恰巧等于2^m，一般总有2^m-n个冗余状态。

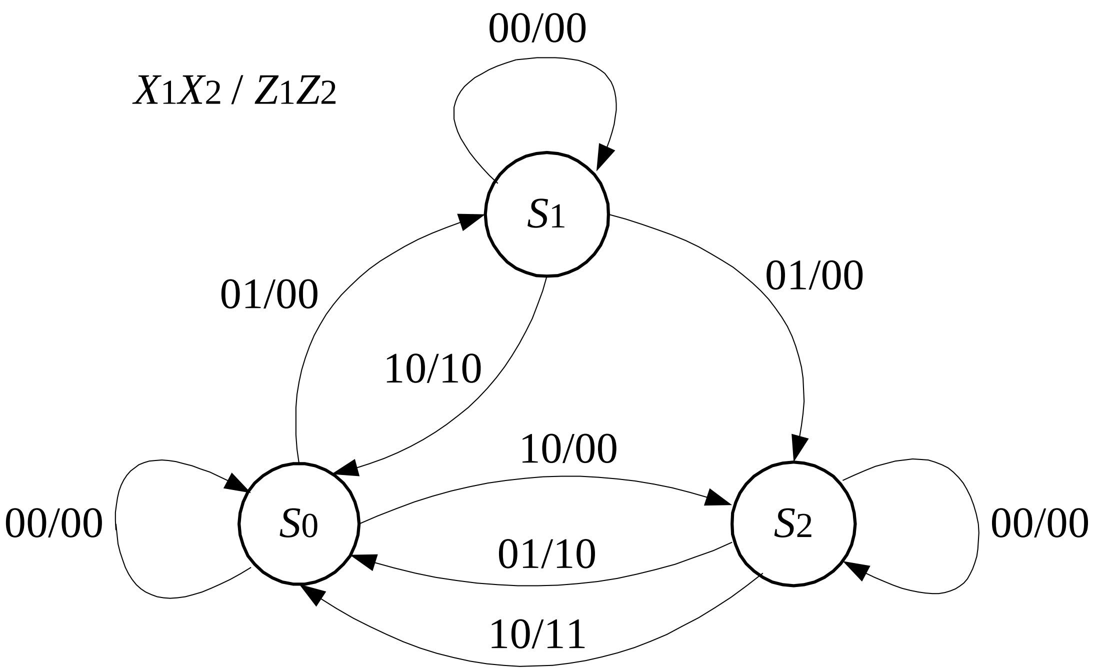
这些冗余状态在设计时必须加以处理。

若处理不当，会造成严重后果。

例4-8(p170)：自动售饮料机

米利模型，

状态转换图：



已是最简状态，无需化简

状态编码如下：S_0 = 00，S_1= 01，S_2= 11

S_3=10是冗余状态。

状态转换表

现态	编码 Q_1Q_2	次态 $Q_{1(N+1)}Q_{2(N+1)}$ / 输出 Z_1Z_2			
		$X_1X_2 = 00$	$X_1X_2 = 01$	$X_1X_2 = 11$	$X_1X_2 = 10$
S_0	00	00 / 00	01 / 00	dd / dd	11 / 00
S_1	01	01 / 00	11 / 00	dd / dd	00 / 10
S_2	11	11 / 00	00 / 10	dd / dd	00 / 11
S_3	10	dd / dd	dd / dd	dd / dd	dd / dd

采用D触发器设计

$Q_{n+1}=D$

直接从上表得到激励表

激励与输出卡诺图

Q1Q2 \ X1X2	00	01	11	10
00	0	0	d	1
01	0	1	d	0
11	1	0	d	0
10	d	d	d	d

D1

Q1Q2 \ X1X2	00	01	11	10
00	0	1	d	1
01	1	1	d	0
11	1	0	d	0
10	d	d	d	d

D2

Q1Q2 \ X1X2	00	01	11	10
00	0	0	d	0
01	0	0	d	1
11	0	1	d	1
10	d	d	d	d

Z1

Q1Q2 \ X1X2	00	01	11	10
00	0	0	d	0
01	0	0	d	0
11	0	0	d	1
10	d	d	d	d

Z2

激励函数与输出函数

$$D_1 = Q_1 \overline{X}_1 \overline{X}_2 + \overline{Q}_1 Q_2 X_2 + \overline{Q}_2 X_1$$

$$D_2 = Q_2 \overline{X}_1 \overline{X}_2 + \overline{Q}_1 X_2 + \overline{Q}_2 X_1$$

$$Z_1 = Q_1 X_2 + Q_2 X_1$$

$$Z_2 = Q_1 X_1$$

疑问：冗余状态是$Q_1Q_2 = 10$，我们将此冗余状态代入上式。看输出方程：当$Q_1Q_2 = 10$时，若输入$X_1 = 1$，则$Z_1 = 0$，$Z_2 = 1$。若输入$X_2 = 1$，则$Z_1 = 1$，$Z_2 = 0$。换句话说，若系统进入冗余状态，那么投入1圆硬币将没有饮料但有5角找零；而只要投入5角硬币，将会得到一杯饮料？（**系统BUG!!**）

发生错误的原因：

按照上面化简后的激励函数重新得到实际的状态转换表

现态	编码 Q_1Q_2	次态 $Q_{1(N+1)}Q_{2(N+1)}$ / 输出 Z_1Z_2			
		$X_1X_2 = 00$	$X_1X_2 = 01$	$X_1X_2 = 11$	$X_1X_2 = 10$
S_0	00	00 / 00	01 / 00	11 / 00	11 / 00
S_1	01	01 / 00	11 / 00	11 / 10	00 / 10
S_2	11	11 / 00	00 / 10	00 / 11	00 / 11
S_3	10	10 / 00	00 / 10	11 / 11	11 / 01

错误的冗余状态：

例如：$X_1X_2 = 00$, 则 S3 → S3

修改方法：将冗余状态的次态和输出按照实际情况填入合理的值，修改后的状态转换表如下。按照此表进行设计，没有不合理的冗余状态。

现态	编码 Q_1Q_2	次态 $Q_{1(N+1)}Q_{2(N+1)}$ / 输出 Z_1Z_2			
		$X_1X_2 = 00$	$X_1X_2 = 01$	$X_1X_2 = 11$	$X_1X_2 = 10$
S_0	00	00 / 00	01 / 00	dd / dd	11 / 00
S_1	01	01 / 00	11 / 00	dd / dd	00 / 10
S_2	11	11 / 00	00 / 10	dd / dd	00 / 11
S_3	10	00 / 00	01 / 00	dd / dd	11 / 00

由于输入$X_1X_2 = 11$不可能发生，所以有关$X_1X_2 = 11$的次态和输出不作调整。

例2：五进制计数器

为了避免出现自启动问题，在开始设计时将冗余状态指向S_0

现态	$Q_2Q_1Q_0$	$Q_{2(n+1)}Q_{1(n+1)}Q_{0(n+1)}$
S_0	000	001
S_1	001	010
S_2	010	011
S_3	011	100
S_4	100	000
S_5	101	000
S_6	110	000
S_7	111	000

Q2 \ Q1Q0	00	01	11	10
0	0	0	1	0
1	0	0	0	0

D2

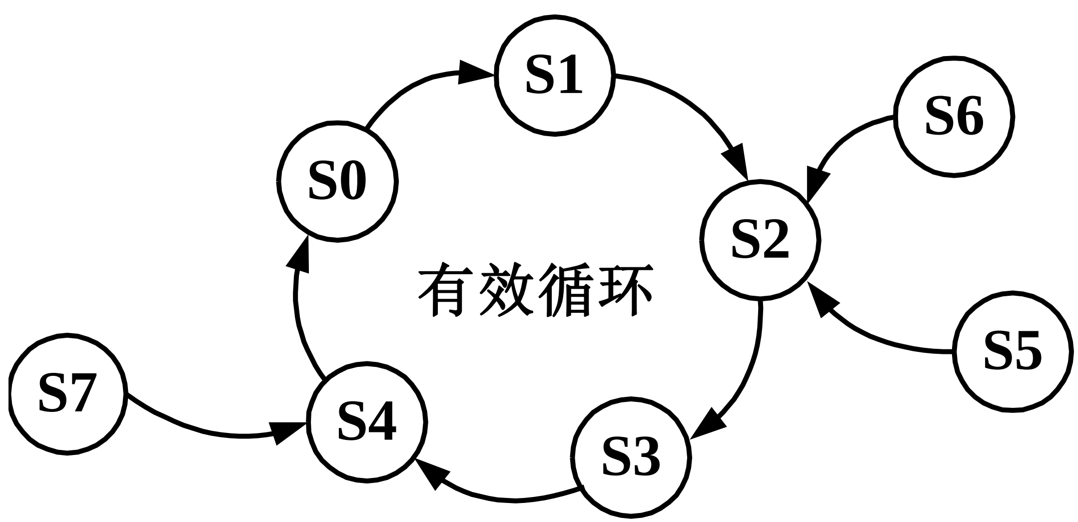
Q2 \ Q1Q0	00	01	11	10
0	0	1	0	1
1	0	0	0	0

D1

Q2 \ Q1Q0	00	01	11	10
0	1	0	0	1
1	0	0	0	0

D0

卡诺图中红色的0是原来设计的冗余项。但是按照原来的设计，卡诺圈太小。所以可以修改设计，按照上图加大卡诺圈（蓝色），相当于改变次态，仍然满足自启动要求。



冗余状态的处理规则

1. 如果问题要求所有冗余状态都具有特定的输出和次态，则在开始进行设计时，除了明确不可能出现的状态以外，应该将所有的冗余状态的输出和次态考虑在状态转换表或状态转换图中。这样得到的设计可以满足问题的原始要求。
2. 如果问题只要求满足自启动条件，则可以以任意项方式处理冗余状态，但是最后要进行自启动检查。
3. 也可以以确定方式处理冗余状态，但可能得到的结果不是比较好的结果，所以应该进行优化检查。
4. 无论上述哪种方法，若在检查后发现问题，都需要按照检查的结果重新修改设计。

例：十进制计数器问题

现态	$Q_3Q_2Q_1Q_0$	$Q_{3(n+1)}Q_{2(n+1)}Q_{1(n+1)}Q_{0(n+1)}$
S_0	0000	0001
S_1	0001	0010
S_2	0010	0011
S_3	0011	0100
S_4	0100	0101
S_5	0101	0110
S_6	0110	0111
S_7	0111	1000
S_8	1000	1001
S_9	1001	0000
S_{10}	1010	dddd
S_{11}	1011	dddd
S_{12}	1100	dddd
S_{13}	1101	dddd
S_{14}	1110	dddd
S_{15}	1111	dddd

Q3Q2 \ Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	d	d	d	d
10	1	0	d	d

D3

Q3Q2 \ Q1Q0	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	d	d	d	d
10	0	0	d	d

D2

Q3Q2 \ Q1Q0	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	d	d	d	d
10	0	0	d	d

D1

Q3Q2 \ Q1Q0	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	d	d	d	d
10	1	0	d	d

D0

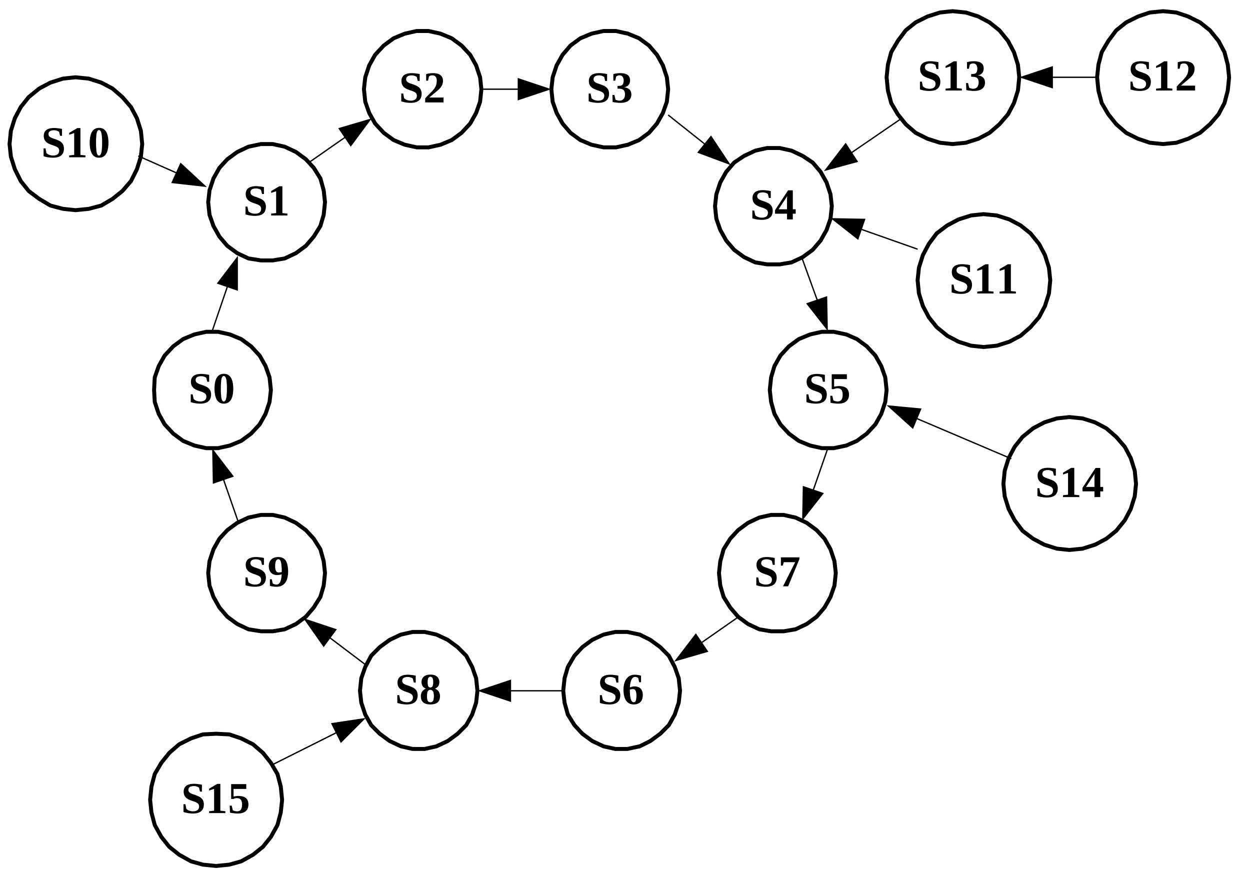
$$D_0 = \overline{Q}_0$$

$$D_1 = Q_0\overline{Q}_1\overline{Q}_3 + \overline{Q}_0 Q_1$$

$$D_2 = \overline{Q}_1 Q_2 + \overline{Q}_0 Q_2 + Q_0 Q_1 \overline{Q}_2$$

$$D_3 = Q_0 Q_1 Q_2 + \overline{Q}_0 \overline{Q}_1 Q_3$$

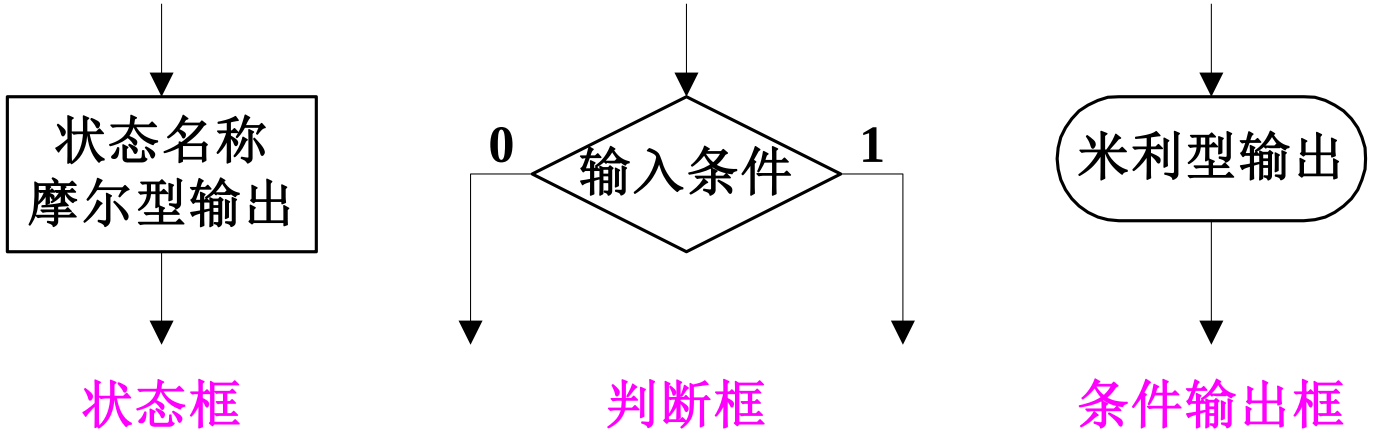
自启动检查：通过

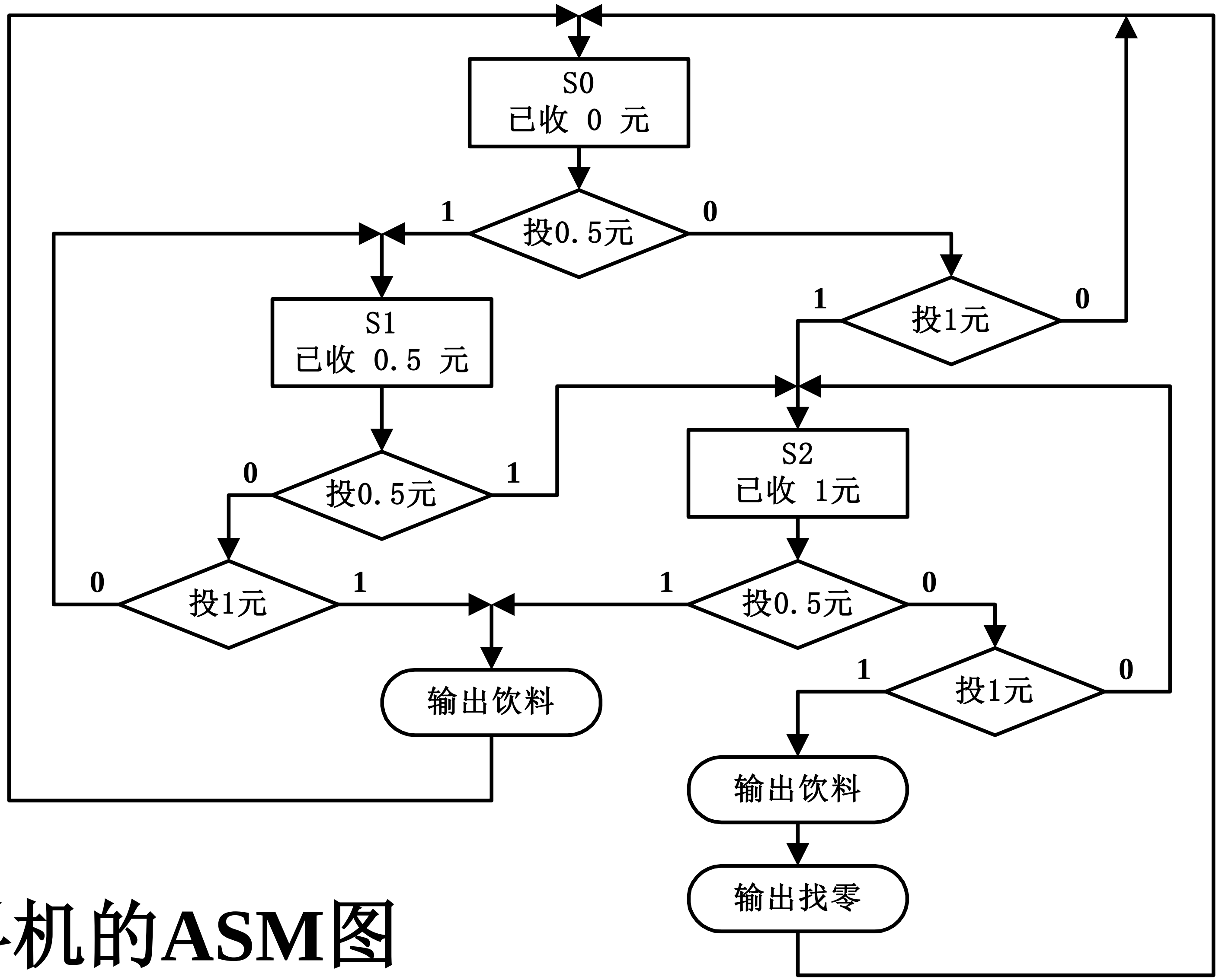


用算法状态机方法设计同步时序电路

算法状态机方法是从计算机程序设计者那里借用了流程图的一些符号，构成算法状态机图（**ASM**图）。

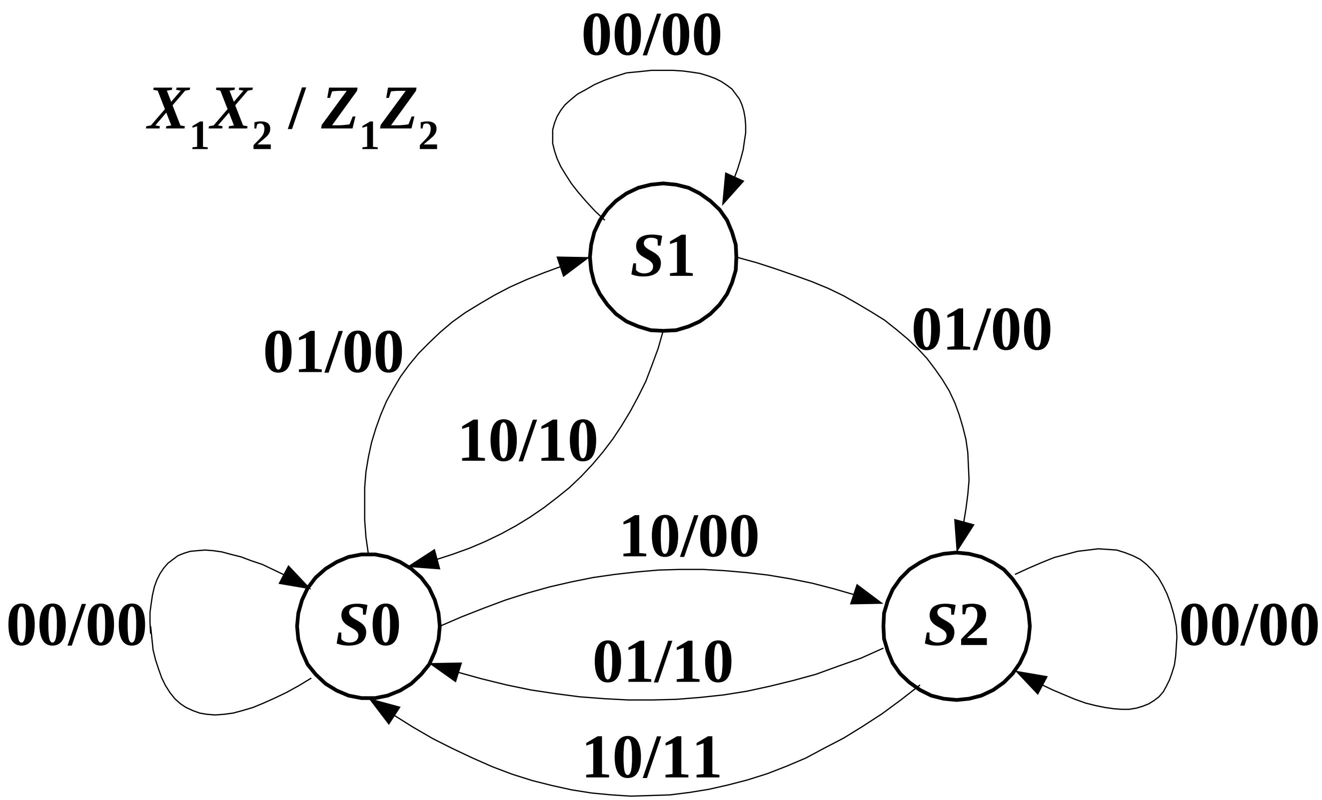
ASM图的主要包括以下**3**种元件




S0
已收 0 元
投0.5元
1
0
投1元
1
0
S1
已收 0.5 元
投0.5元
0
1
投1元
0
1
S2
已收 1元
投0.5元
1
0
投1元
1
0
输出饮料
输出饮料
输出找零

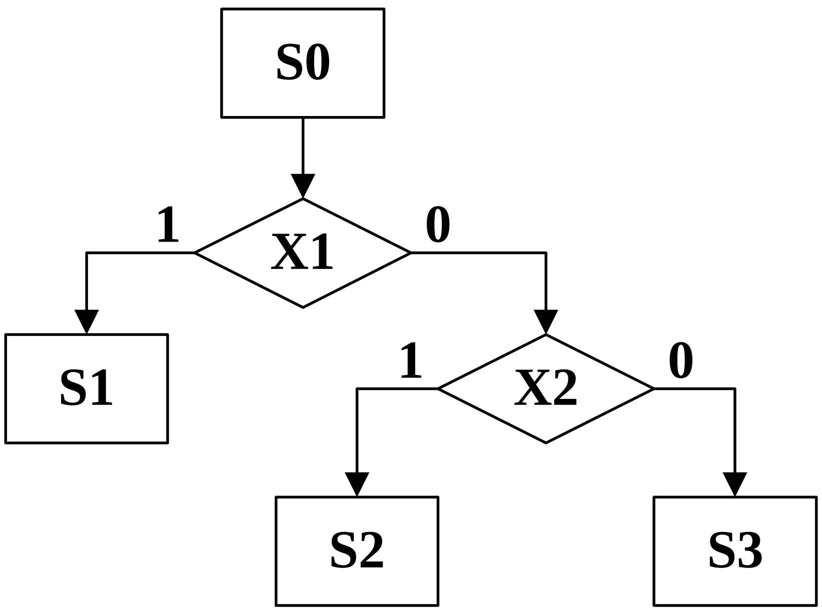
售饮料机的ASM图

售饮料机的状态转换图

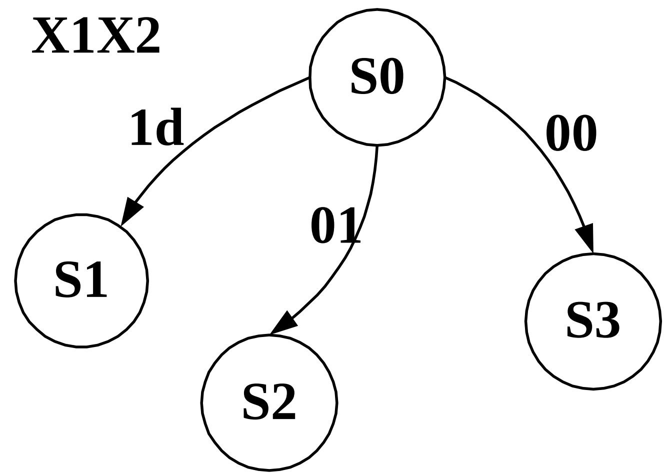


00/00
X_1X_2 / Z_1Z_2
S1
01/00
01/00
10/10
10/00
00/00
S0
S2
00/00
01/10
10/11

优先级的处理



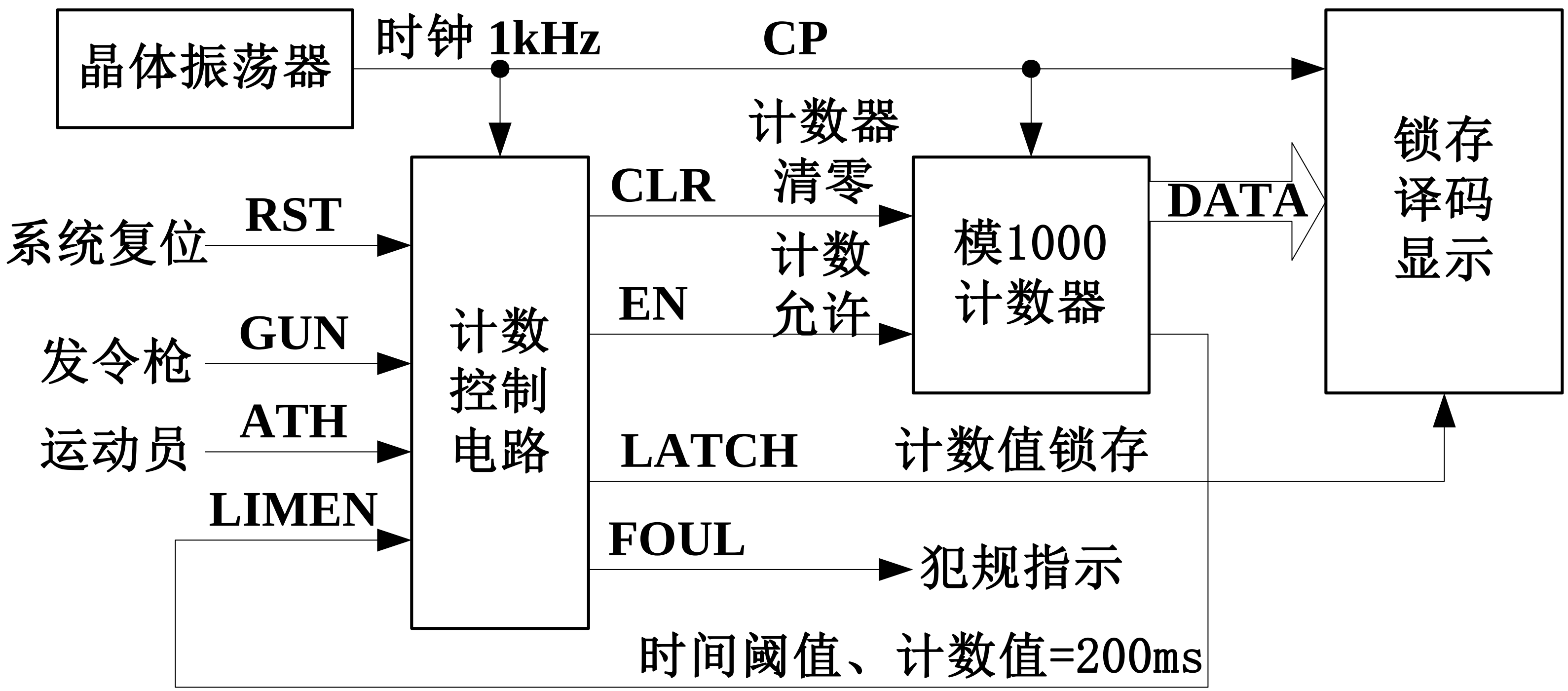
ASM图

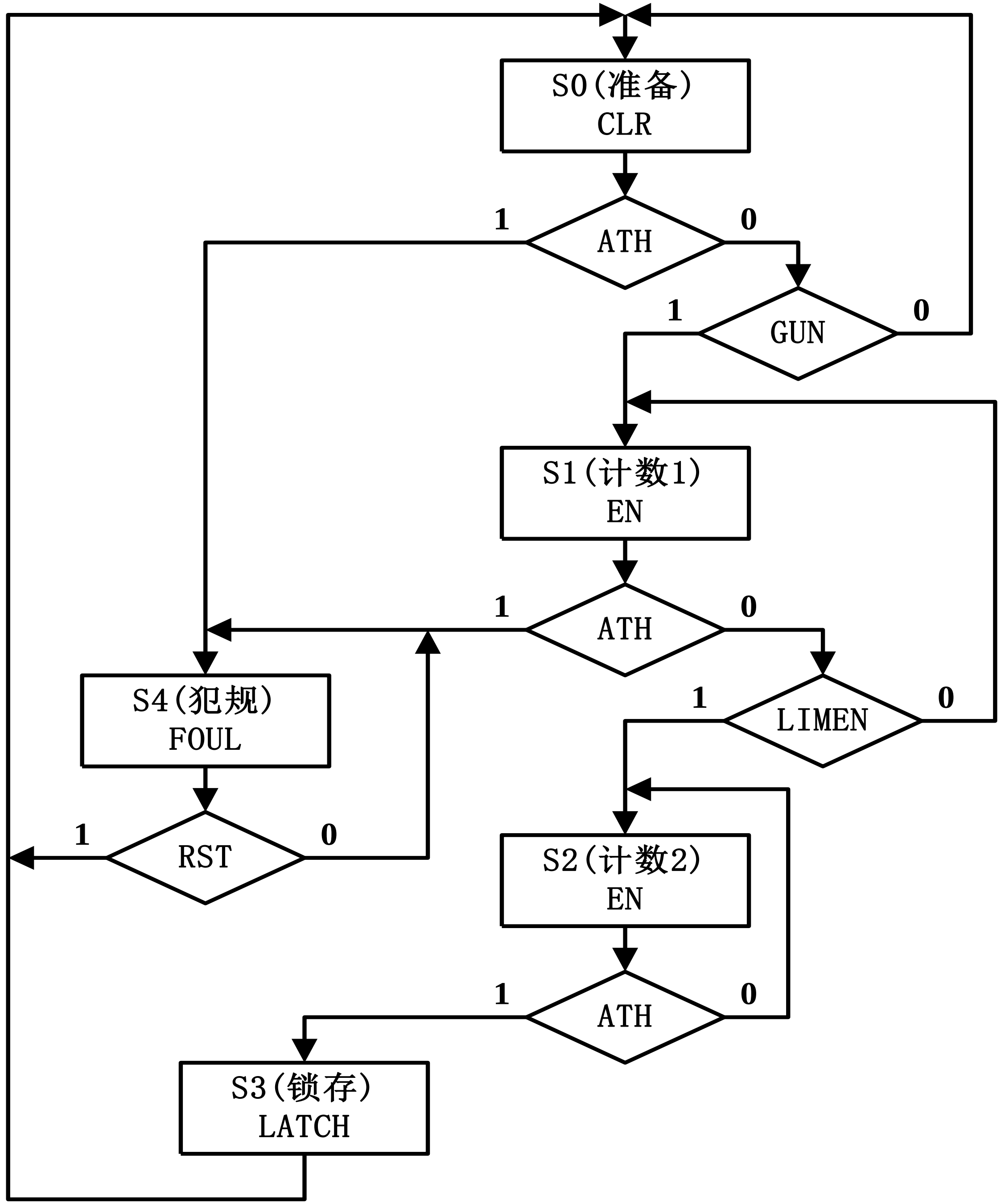


状态转换图

例：反应时间测试电路

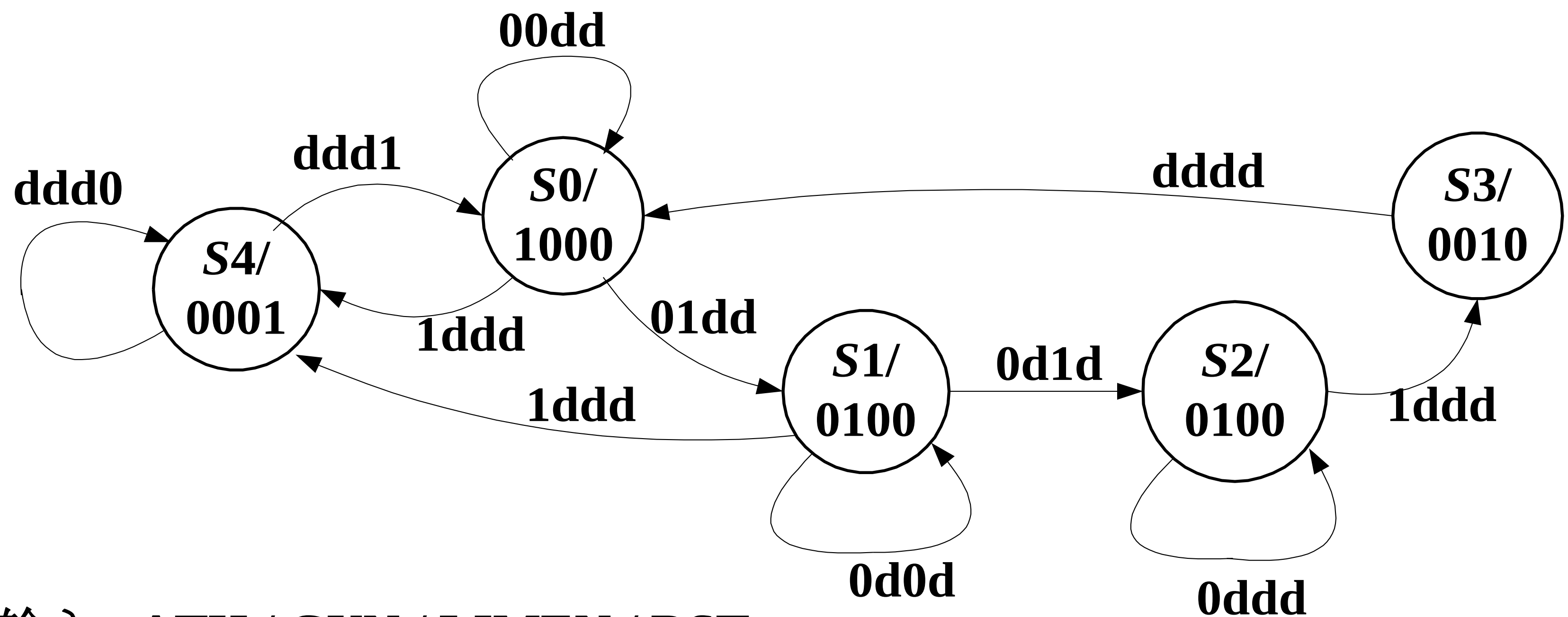
该电路用来测量短跑运动员的反应速度，要求时间测量精确到毫秒。由于运动员的反应时间不可能小于200ms，所以要求当反应时间小于200ms时，要给出犯规信号。





S0(准备)
CLR
ATH
1
0
GUN
1
0
S1(计数1)
EN
ATH
1
0
LIMEN
1
0
S4(犯规)
FOUL
RST
1
0
S2(计数2)
EN
ATH
1
0
S3(锁存)
LATCH

状态转换图



输入　ATH / GUN / LIMEN / RST
输出　CLR / EN / LATCH / FOUL

状态分配

时序电路的状态由状态变量（时序电路中记忆电路的输出）组合确定，将每个状态对应的状态变量组合分配一个唯一的二进制码的过程称为进行状态分配。

在时序电路设计中必须进行状态分配。分配之后用以代表状态变量的实际二进制码对于最终实现电路的代价有着重要影响。

状态分配问题没有通解。但是根据前人的经验，按照一定的分配规则，可以得到比较好的分配结果。

用m个状态变量实现n个状态时，可能的状态分配数

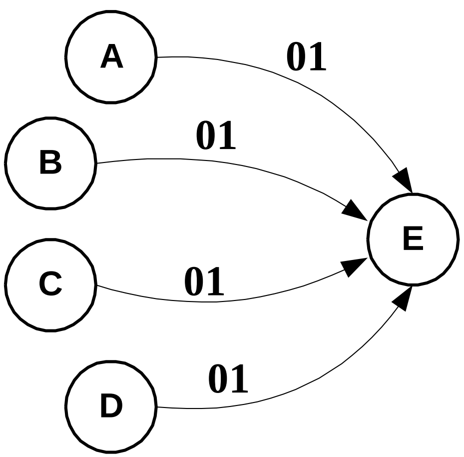
$$S = \frac{(2^m)!}{(2^m - n)!}$$

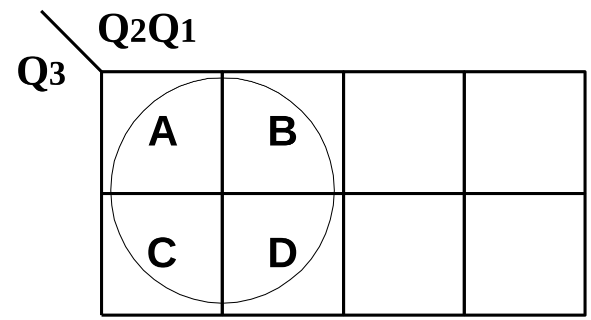
将等价的状态分配方案剔除后，以下函数给出了不等价的状态分配方案数。

$$S = \frac{(2^m - 1)!}{(2^m - n)!\, m!}$$

状态分配规则-1

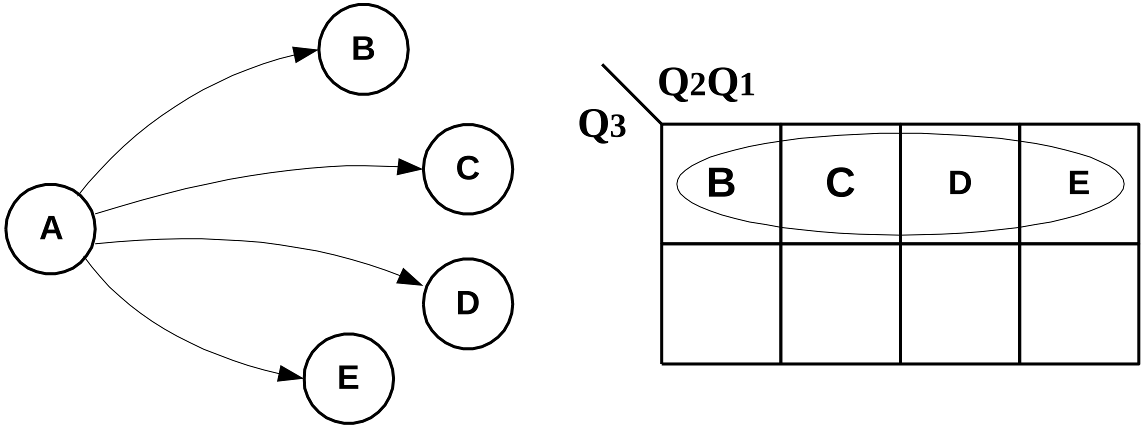
对于在相同输入条件下有相同次态的所有状态，在进行状态分配后，应当能形成一个质蕴含。





状态分配规则-2

对于一个现态的所有次态，在进行状态分配后，最好能够形成一个质蕴含。



状态分配规则-3

有相同输出的状态最好给予相邻的状态分配。

对于状态分配的说明：

1、上述3个规则，是一些经验规律，大部分情况下适用，可能有不适用的例子

2、对于大部分状态机，不可能全部满足3条规则，只能按照实际情况运用。规则1比规则2的优先级要高，规则3的优先级最低。除了涉及有大量输出的情况之外，一般很少考虑规则3

3、有些资料根本就忽略规则3

例4-13 (p182) 状态分配的例子-1

现态	次态	
	$X=0$	$X=1$
S_0	S_0	S_2
S_1	S_2	S_3
S_2	S_0	S_5
S_3	S_1	S_4
S_4	S_2	S_2
S_5	S_3	S_4

规则1：相同输入、相同次态的现态形成一个质蕴含

$X=0$：{$S0$，$S2$} 与 {$S1$，$S4$}

$X=1$：{$S0$，$S4$} 与 {$S3$，$S5$}

规则2：状态表中的每一行的次态要求相邻

{$S0$，$S2$}；{$S2$，$S3$}；{$S0$，$S5$}；{$S1$，$S4$}；{$S3$，$S4$}

结合规则1和规则2，要求相邻的状态集合排队

$\{S0，S2\}$；$\{S1，S4\}$；$\{S0，S4\}$；$\{S3，S5\}$；$\{S2，S3\}$；$\{S0，S5\}$；$\{S3，S4\}$

可能的分配方案：

Q_3 \ Q_2Q_1	00	01	11	10
0	S0	S4	S1	S5
1	S2			S3

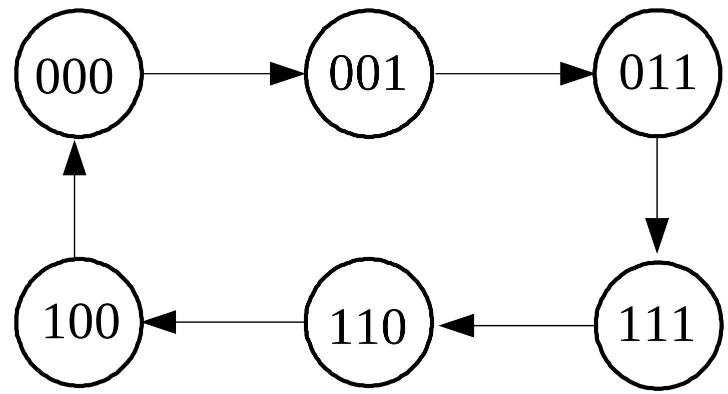
Q_3 \ Q_2Q_1	00	01	11	10
0	S0	S2		S5
1	S4	S3		S1

Q_1 \ Q_3Q_2	00	01	11	10
0	S0	S4	S3	S2
1	S5	S1		

状态分配的例子-2

- 试用3个JK触发器（每个只有1个J端和1个K端）构成一个同步模5计数器，不得增加其他门电路。提示：先构成有6个状态的扭环形计数器，再设法去除一个状态。

扭环形计数器的状态转换图：



扭环形计数器的状态转换卡诺图和激励卡诺图

Q_2 \ Q_1Q_0	00	01	11	10
0	00α	$0\alpha 1$	$\alpha 11$	
1	$\beta 00$		11β	$1\beta 0$

状态转换关系	J	K
0（0→0）	0	d
α（0→1）	1	d
β（1→0）	d	1
1（1→1）	d	0

J_2:

Q_2 \ Q_1Q_0	00	01	11	10
0	0	0	1	
1	d		d	d

K_2:

Q_2 \ Q_1Q_0	00	01	11	10
0	d	d	d	
1	1		0	0

J_1:

Q_2 \ Q_1Q_0	00	01	11	10
0	0	1	d	
1	0		d	d

K_1:

Q_2 \ Q_1Q_0	00	01	11	10
0	d	d	0	
1	d		0	1

J_0:

Q_2 \ Q_1Q_0	00	01	11	10
0	1	d	d	
1	0		d	0

K_0:

Q_2 \ Q_1Q_0	00	01	11	10
0	d	0	0	
1	d		1	d

此处：卡诺圈包含 4 格，才不需额外电路。

不增加附加的组合电路的关键：

对于 n 个变量的卡诺图，所有卡诺圈必须包含 2^n 或 2^{n-1} 个方格

对于本例题，需要在原来的卡诺图中去掉一个状态。关键是寻找一个合适的状态分配，使得剩余的5个状态的激励卡诺图中，所有卡诺圈仍然只包含4个方格

可能的方案有：

去掉000状态，从100直接转到001状态

去掉001状态，从000直接转到011状态

去掉011状态，从001直接转到111状态，等

这里，是否去掉任意一个都可以？

去掉000状态，从100直接转到001状态的过程

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	00α	$0\alpha 1$	$\alpha 11$	
1	$\beta 00$		11β	$1\beta 0$

原来的状态转换卡诺图

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0		$0\alpha 1$	$\alpha 11$	
1	$\beta 0\alpha$		11β	$1\beta 0$

修改后的状态转换卡诺图

修改的状态

要修改的激励

J_2:

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	0	0	1	
1	d		d	d

K_2:

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	d	d	d	
1	1		0	0

J_1:

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	0	1	d	
1	0		d	d

K_1:

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	d	d	0	
1	d		0	1

J_0:

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	1	d	d	
1	0		d	0

K_0:

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	d	0	0	
1	d		1	d

Ｊ２／Ｊ１都没变，其实０００时应为ｄ。修改后的没画出来，应为１。

4.4 时序电路的状态化简

一、完全描述状态表的化简

“完全描述”是指在给定的输入条件下，表中所有的次态和输出均有确定值。

完全描述状态表的例：

现态	次态 / 输出	
	X=0	X=1
S_0	S_0/0	S_2/0
S_1	S_0/0	S_2/0
S_2	S_1/0	S_3/1
S_3	S_1/0	S_3/0

状态的等价：

设某一时序电路（系统）内的两个状态 *Si* 和 *Sj*，如果用任意序列的输入加到此电路上，从 *Si* 或 *Sj* 出发所得到的输出序列都相同，则称状态 *Si* 和状态 *Sj* 等价。

可以证明，为了判别两个状态是否等价，所输入的任意序列只需要有限长度。具体地说，对于具有 n 个状态的电路，最多只需要输入 n-1个符号的任意输入序列，即可判别两个状态的等价与否。

等价的传递性：

若 *Si* 等价于 *Sj*，而 *Sj* 等价于 *Sk*，则 *Si* 等价于 *Sk*。

划分等价状态的规律：

1、如果某两个状态对应的输出不同，则它们显然是不等价的。

2、如果某两个状态在相同的输入下有相同的输出，并且次状态完全相等或为原状态时，这两个状态等价。

3、如果某两个状态在相同的输入下有相同的输出，但是次状态不相同，则此二状态等价与否还得视它们的次状态是否等价而定。

逐次分割法（p186）

化简的方法：隐含表法

1. 输出不同，肯定不等价，在相应的小方格中记以“×”号。
2. 输出相同，并且在相同的输入下次态相同，或者仍为原状态对。肯定等价，在对应的小方格中记以“√”号 。
3. 输出相同，次态不同，记录次态对。
4. 观察该次态对是否等价（等价传递）。若是则钩，若非则叉。

状态	次态		输出	
	X=0	X=1	X=0	X=1
S1	S1	S7	0	1
S2	S2	S4	0	0
S3	S4	S5	1	0
S4	S7	S5	1	1
S5	S5	S7	0	1
S6	S6	S4	0	0
S7	S3	S6	0	1

S2	×					
S3	×	×				
S4	×	×	×			
S5	√	×	×	×		
S6	×	√	×	×	×	
S7	**S1S3 S7S6**	×	×	×	**S5S3 S7S6**	×
	S1	**S2**	**S3**	**S4**	**S5**	**S6**

S2	×					
S3	×	×				
S4	×	×	×			
S5	√	×	×	×		
S6	×	√	×	×	×	
S7	×	×	×	×	×	×
	S1	**S2**	**S3**	**S4**	**S5**	**S6**

最后将所有等价态合并，就得到化简后的状态表。

对于上例，由于状态 S_1 与 S_5 等价，状态 S_2 与 S_6 等价，所以可以将这 4 个状态化简成 2 个状态，而其余3个状态则无法进一步化简。

原状态	化简后状态	次态		输出	
		X=0	X=1	X=0	X=1
S_1，S_5	S_A	S_A	S_E	0	1
S_7	S_E	S_C	S_B	0	1
S_2，S_6	S_B	S_B	S_D	0	0
S_3	S_C	S_D	S_A	1	0
S_4	S_D	S_E	S_A	1	1

二、不完全描述状态表的化简

不完全描述状态表：

表中某些状态在某些输入情况下的次态或输出没有确定的值，它们可以取0也可以取1，

状态	次态		输出
	X=0	X=1	
S1	S2	S4	0
S2	S1	S4	d
S3	S3	S5	1
S4	d	S5	1

相容：

1、相容：

如果两个输出序列的每一对有确定值的对应输出均相同，则称此两输出相容。

例：

$$Z_i = 1010d1$$

$$Z_j = 10d011$$

$$Z_k = 1d0011$$

Z_i与Z_j相容，Z_j与Z_k也相容。而Z_i与Z_k则不相容。可见，相容不具有传递性。

2、状态相容：

两个状态 *Si* 和 *Sj*，如果用任意序列的输入加到此系统上，从 *Si* 或 *Sj* 出发所得到的输出序列都相容，则称状态 *Si* 和状态 *Sj* 相容。

- 简单不相容：

在相同输入下，状态 *Si* 和 *Sj* 输出不相容。

- 简单相容：

对于任意输入，输出相容，有确定值的次状态相等或者仍为原状态。

例：用隐含表化简

状态	次态		输出
	X=0	X=1	
S_1	S_3	d	1
S_2	d	S_3	d
S_3	d	S_2	d
S_4	S_4	d	0
S_5	S_5	S_4	d

简单相容

简单相容

简单相容

简单相容

简单不相容

简单相容

简单相容

S2	√			
S3	√	√		
S4	×	√	√	
S5	**S3S5**	**S3S4**	**S2S4**	√
	S1	**S2**	**S3**	**S4**

步骤1、寻找相容关系

打钩的为简单相容，打叉的为简单不相容。

其余三个状态对，既不是简单相容，也不是简单不相容。写上它们有确定值的次态对。

由于相容不具有传递性，所以不能简单地由简单相容外推两个不简单相容的状态是否相容。

例如上表中，S_1与S_3相容，S_3与S_4相容，但是S_1与S_4简单不相容。

同样，也不能推断上表中写上次态对的3个方格：S_1与S_5、S_2与S_5以及S_3与S_5的相容性。

若两个状态的输出相容，有确定值的次态不同，但是该次态对是简单相容的，则此两个状态仍然可能相容。我们称此两个状态潜在相容。

考察上述3个状态对的次态对是否相容：其中两个次态对简单相容，一个次态对的次态对简单相容（潜在相容），所以上述3个状态对均潜在相容。

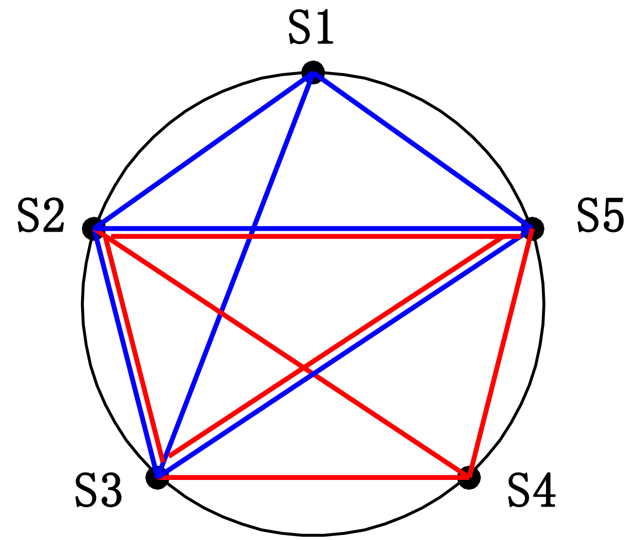
步骤2、寻找最大相容类集合

所谓最大相容类集合，是含有不被其它相容类集合所覆盖的相容的一组状态或状态对。寻找最大相容类集合的目的是要将全体状态划分为尽可能少的组，每组内的状态都彼此相容。

利用合并图来寻找最大相容类集合：
合并图是将状态表中所有的状态以点的形式画在一个圆周上，然后将所有的相容状态对（包括潜在相容对）用直线连接起来。若在一组状态点中两两之间都有连线，表示该组状态中所有状态两两相容，可以形成最大相容类。在图形上，它们将形成一个内部两两相连的最大的多边形，该多边形的各个顶点就形成一个最大相容类集合。

潜在相容

S_2	√			
S_3	√	√		
S_4	×	√	√	
S_5	S_3S_5	S_3S_4	S_2S_4	√
	S_1	S_2	S_3	S_4



在本例题中，共有5个状态，构成的相容状态对有：

{$S1$，$S2$}、{$S1$，$S3$}、{$S1$，$S5$}

{$S2$，$S3$}、{$S2$，$S4$}、{$S2$，$S5$}

{$S3$，$S4$}、{$S3$，$S5$}

{$S4$，$S5$}

最大相容类集合：

{ $S1$，$S2$，$S3$，$S5$ } 和 { $S2$，$S3$，$S4$，$S5$ }

步骤3、构成简化状态表

最大相容类集合 $\{A\}$，$\{B\}$，…，$\{K\}$ 可能有状态重叠，所以可以用其中的一部分作为一个简化状态。若用 $\{A'\}$ 表示 $\{A\}$ 的一个子集，$\{B'\}$ 表示 $\{B\}$ 的一个子集，等，则可以用 $\{A'\}$，$\{B'\}$，…，$\{K'\}$ 中的一部分 $\{A'\}$，$\{B'\}$，…，$\{N'\}$作为化简后的状态。其中 $\{A'\}$，$\{B'\}$，…，$\{N'\}$ 应满足下列三个条件：

1、**覆盖化**，$\{A'\}$，$\{B'\}$，…，$\{N'\}$中必须包含原状态表中所有的状态。

2、**最小化**，由$\{A'\}$，$\{B'\}$，…，$\{N'\}$构成的状态数目最少。也就是说，若在第一步得到k个最大相容类集合，最后的简化状态个数可以小于k。

3、**闭合性**，$\{A'\}$，$\{B'\}$，…，$\{N'\}$中任何一个新状态，它所包含的几个原来的状态在一定的输入下对应的次状态必须属于合并后的同一状态。否则简化表的次状态无法确定。

根据上述规则从最大相容类中选择化简后的状态集合来构成简化状态表，其方法不是唯一的，部分地要凭经验进行试验。

此处：闭合性，次态属于同一状态就可以，不是非得在自己所在的状态。

原来的例子已经得到2个最大相容类集合，分别是

$\{S1, S2, S3, S5\}\{S2, S3, S4, S5\}$

根据前面所说的三个条件中的前面二个——覆盖性和最小化考虑，选取上述2个最大相容类集合中的部分子集，可以按下面的组合来选取简化后的状态组合：

(1) $\{S1, S5\}\{S2, S3, S4\}$

(2) $\{S1\}\{S2, S3, S4, S5\}$

(3) $\{S1, S2\}\{S3, S4, S5\}$

(4) $\{S1, S2, S3\}\{S4, S5\}$

上述4个组合，每个组合中都包含了原来的所有状态，所以都满足覆盖性要求。这样组合以后的简化状态只有2个，并且不可能进一步减少，所以满足最小化要求。

但是，方案(1)和(3)均不满足闭合性。

以方案(1)为例，原始问题S_1，S_5在输入$X = 0$时对应的次状态分别为S_3和S_5。但如果按照方案(1)简化以后，S_1和S_5将合并成一个状态，而它们的次状态S_3和S_5将分属于化简后的两个不同状态，这样一来，在化简后的状态表中新状态 { S_1，S_5 } 在输入$X = 0$时的次态将无法选择。

状态	次态
	X=0
S_1	S_3
S_5	S_5

原始的状态转换关系

化简后的状态	次态
	X=0
S_A={S_1 , S_5}	$S_A(S_5)$? $S_B(S_3)$?
S_B={S_2 , S_3 , S_4}	

方案（1）的状态转换关系

反之，方案(2)和(4)则可以满足上述三个条件。所以这两种简化方案是可取的。假定取第二种方案，以S_A代替{ $S1$ }而以S_B代替{ $S2$，$S3$，$S4$，$S5$ }，于是本问题的状态转换表便简化成只有两个状态，如下表所示。

状态	次态		输出
	X=0	X=1	
S_A	S_B	d	1
S_B	S_B	S_B	0

此处：前面过程对输出值未讨论，最后赋了确定值。

VIP

Video Image Processing

Research Group @ Fudan

http://soc.fudan.edu.cn/vip/

Thank you !

Acknowledgement: Part of contents are referenced from Yufeng Xie, Ting Yi PPT