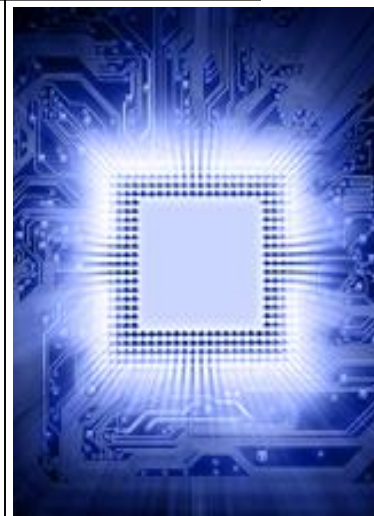


第7章 时序逻辑设计原理

(part-3)

武港山

南京大学人工智能学院





内容简介

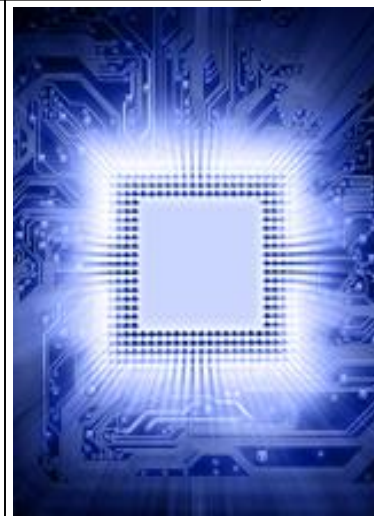


- 双稳态电路(Bi-stable Device)
- 锁存器与触发器
- 时钟同步状态机分析
- 时钟同步状态机设计
 - 用状态表设计状态机
 - 用状态图设计状态机
 - 用转移表设计状态机
- 反馈时序电路分析
- 反馈时序电路设计
- Verilog设计时序电路

分析工具
分析方法
例题讲解

7.5

用状态图设计状态机





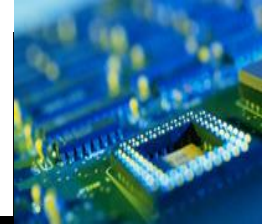
用状态图设计状态机



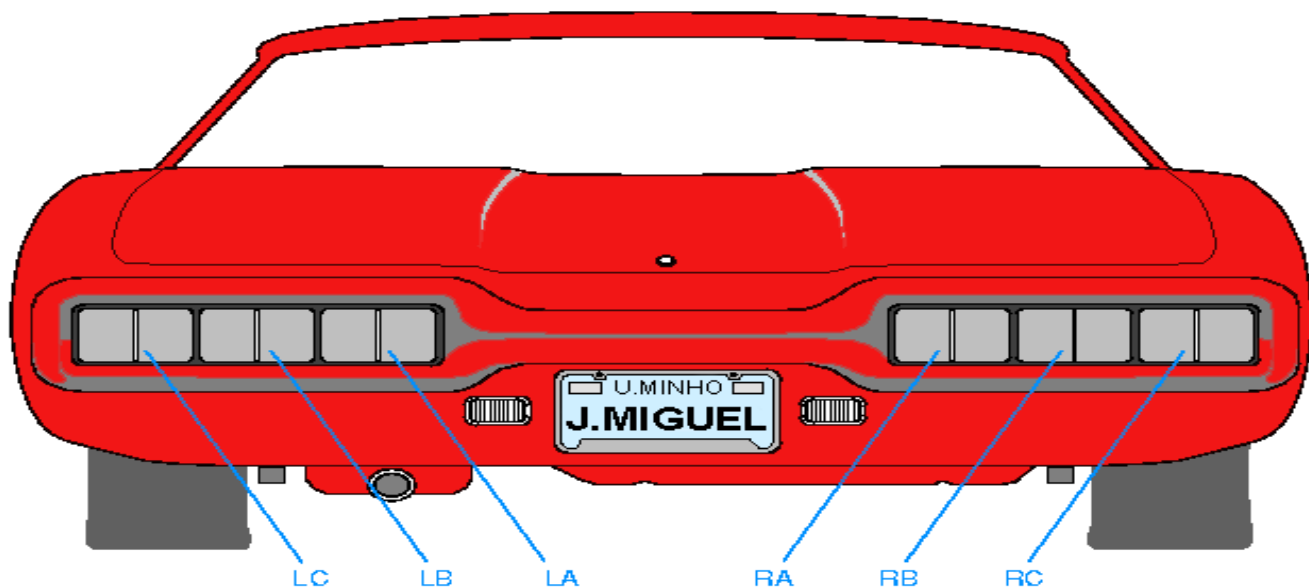
- 状态图与状态表的设计相似
 - 状态表采用穷举法，列出所有状态/输入组合的次态
 - 状态图包含一组标有转移表达式的弧线。
 - 简单、易出错。
 - 有些状态/输入组合可能没有确定的次态（不希望的）；二义性
 - 有些状态/输入组合可能对应于多个次态（错误的）。



雷鸟车尾灯设计

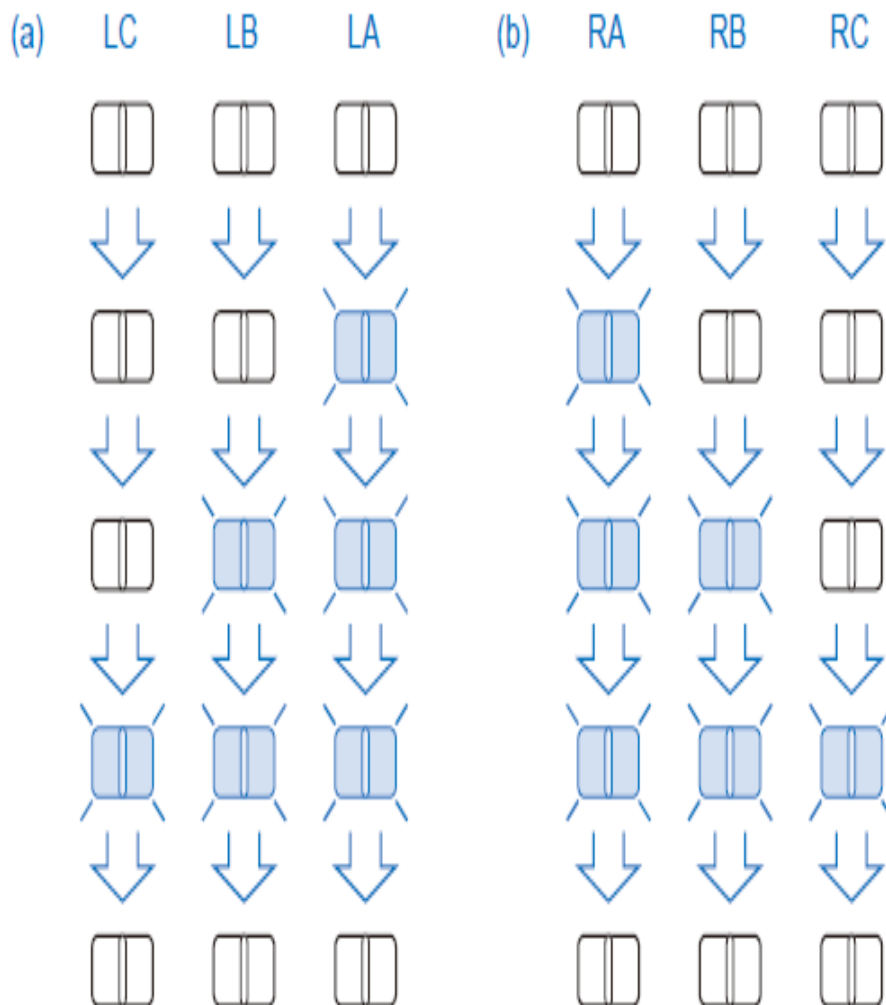


- 雷鸟车尾灯
 - 每边各有3个灯：轮流顺序亮起，表示车的转向
 - 三个输入：左转、右转、应急闪烁输入，时钟信号
- 功能
 - 转向状态：6个灯轮流协调闪烁
 - 告警状态





雷鸟车尾灯设计



Output Table

State	LC	LB	LA	RA	RB	RC
IDLE	0	0	0	0	0	0
L1	0	0	1	0	0	0
L2	0	1	1	0	0	0
L3	1	1	1	0	0	0
R1	0	0	0	1	0	0
R2	0	0	0	1	1	0
R3	0	0	0	1	1	1
LR3	1	1	1	1	1	1



雷鸟车尾灯设计



● 8状态

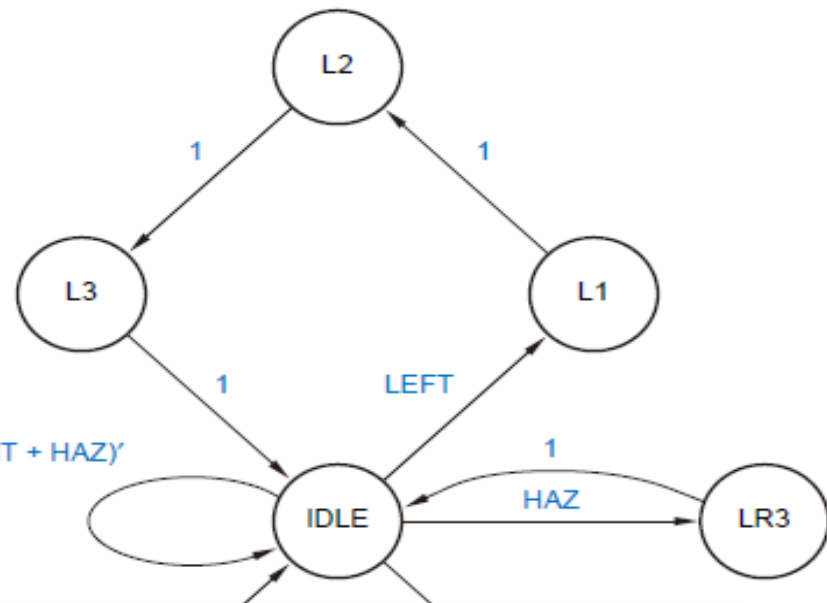
- 原始图有问题，无法处理多输入有效的情形，LEFT和HAZ同时有效？

I

Output Table

State	LC	LB	LA	RA	RB	RC
IDLE	0	0	0	0	0	0
L1	0	0	1	0	0	0
L2	0	1	1	0	0	0
L3	1	1	1	0	0	0
R1	0	0	0	1	0	0
R2	0	0	0	1	1	0
R3	0	0	0	1	1	1
LR3	1	1	1	1	1	1

(LEFT + RIGHT + HAZY)



$$LA=L1+L2+L3+LR3$$

$$LB=L2+L3+LR3$$

$$LC=L3+LR3$$

$$RA=R1+R2+R3+LR3$$

$$RB=R2+R3+LR3$$

$$RC=R3+LR3$$

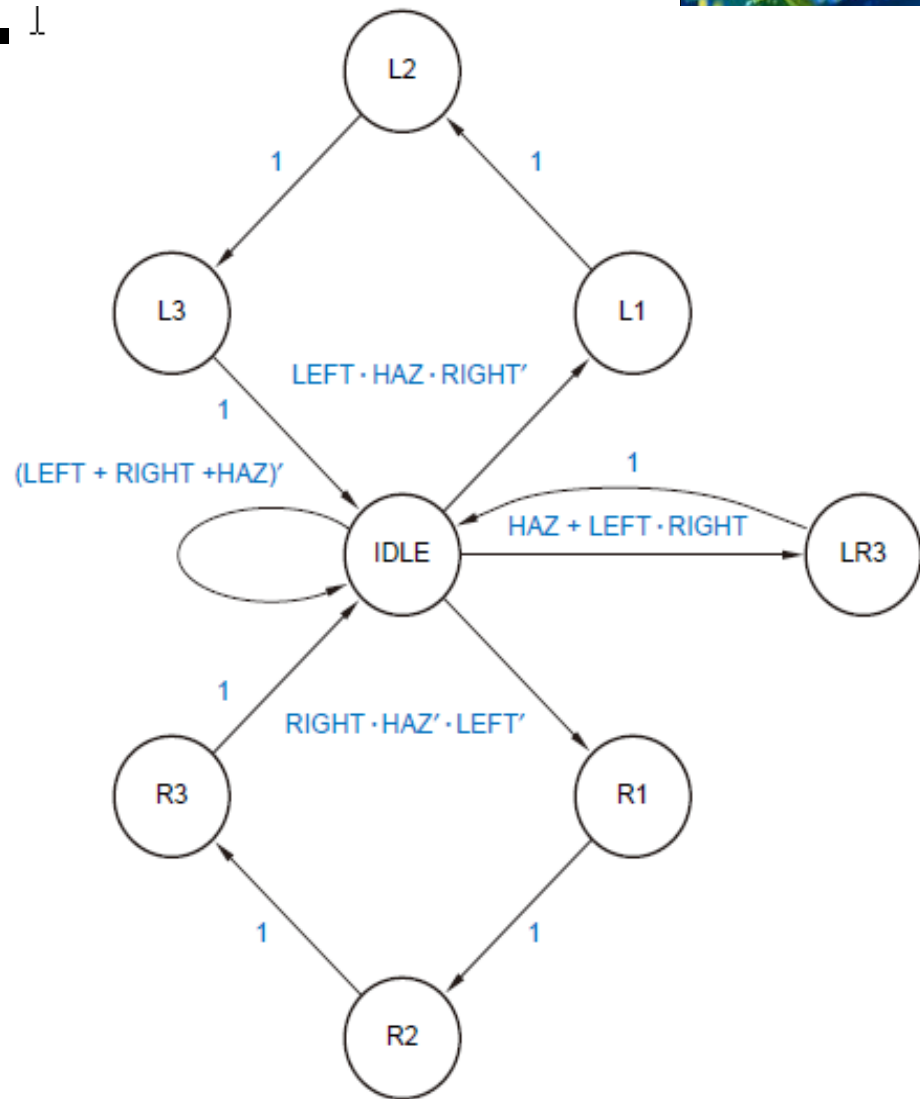


雷鸟车尾灯设计



↓

- 状态图改进1
 - 转移表达式都是互斥且完备的。
 - 但语义混乱：**HAZ**告警，且同时左转和右转。
- 状态图的要求：
 - 互斥性(mutual exclusion)：离开状态的弧线上所标的任意一对转移表达式的逻辑积等于0。

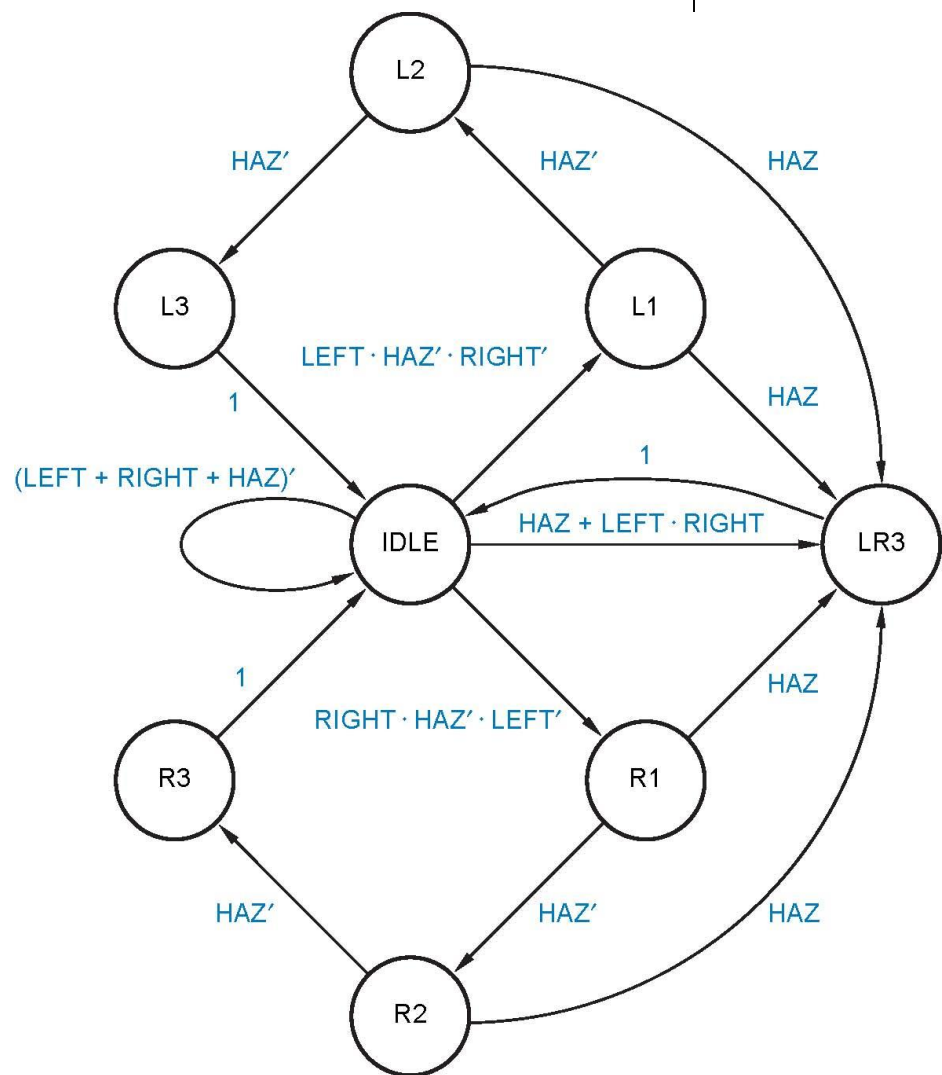




雷鸟车尾灯设计



- 状态图的要求（续）
 - 完备性（all inclusion）
：离开状态的弧线上所标的所有转移表达式的逻辑和等于1。
- 状态表的结构保证了无二义性
- 状态图需要证明
 - 改进状态图2
 - 在左右转状态可进入告警状态





雷鸟车尾灯设计



- 状态赋值

- 8个状态，3个触发器
Q1Q2Q3

- 选择一种

- 初始态为：000
- 左右转循环采用两位格雷码顺序（状态转移时发生变化的状态变量最少，简化激励逻辑）。
- 另一位状态变量用来区分左右转
- 剩下的状态变量组合表示状态LR3

State	Q2	Q1	Q0
IDLE	0	0	0
L1	0	0	1
L2	0	1	1
L3	0	1	0
R1	1	0	1
R2	1	1	1
R3	1	1	0
LR3	1	0	0



雷鸟车尾灯设计



- 状态转移列表：由转移表达式指定

<i>S</i>	<i>Q2</i>	<i>Q1</i>	<i>Q0</i>	<i>Transition Expression</i>	<i>S*</i>	<i>Q2*</i>	<i>Q1*</i>	<i>Q0*</i>
IDLE	0	0	0	$(\text{LEFT} + \text{RIGHT} + \text{HAZ})'$	IDLE	0	0	0
IDLE	0	0	0	$\text{LEFT} \cdot \text{HAZ}' \cdot \text{RIGHT}'$	L1	0	0	1
IDLE	0	0	0	$\text{HAZ} + \text{LEFT} \cdot \text{RIGHT}$	LR3	1	0	0
IDLE	0	0	0	$\text{RIGHT} \cdot \text{HAZ}' \cdot \text{LEFT}'$	R1	1	0	1
L1	0	0	1	HAZ'	L2	0	1	1
L1	0	0	1	HAZ	LR3	1	0	0
L2	0	1	1	HAZ'	L3	0	1	0
L2	0	1	1	HAZ	LR3	1	0	0
L3	0	1	0	1	IDLE	0	0	0
R1	1	0	1	HAZ'	R2	1	1	1
R1	1	0	1	HAZ	LR3	1	0	0
R2	1	1	1	HAZ'	R3	1	1	0
R2	1	1	1	HAZ	LR3	1	0	0
R3	1	1	0	1	IDLE	0	0	0
LR3	1	0	0	1	IDLE	0	0	0



7.6 用状态转移表综合状态机



- 得出转移方程：根据前面状态和输入来定义下一状态变量 V^*

$$V^* = \sum (\text{转移p项}), V^* = 1 \text{ 的行}$$

- 转移p项：当前行状态最小项与转移表达式的乘积。

$$Q2^* = Q2' \cdot Q1' \cdot Q0'$$

$$(HAZ + RIGHT) + Q2' \cdot Q0' \\ (HAZ) + Q2 \cdot Q0$$

$$Q1^* = Q0 \cdot HAZ'$$

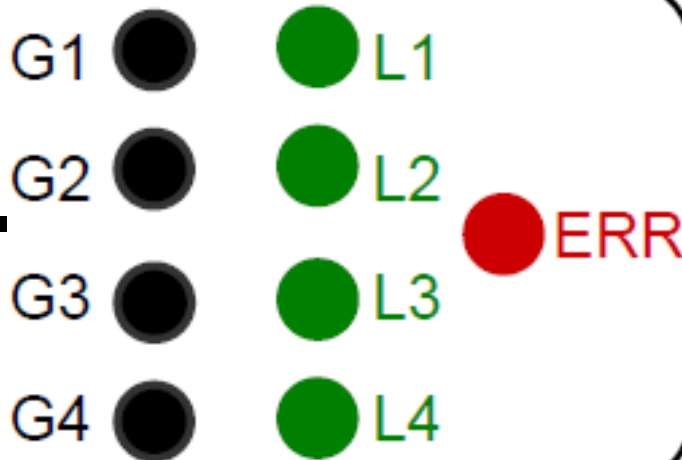
$$Q0^* = Q2' \cdot Q1' \cdot Q0' \cdot HAZ' \cdot \\ (LIGHT \oplus RIGHT) + Q1' \cdot Q0' \\ \cdot HAZ'$$

• 激励方程

• 逻辑电路

S	Q2	Q1	Q0	Transition Expression	S*	Q2*	Q1*	Q0*
IDLE	0	0	0	$(LEFT + RIGHT + HAZ)'$	IDLE	0	0	0
IDLE	0	0	0	$LEFT \cdot HAZ' \cdot RIGHT'$	L1	0	0	1
IDLE	0	0	0	$HAZ + LEFT \cdot RIGHT$	LR3	1	0	0
IDLE	0	0	0	$RIGHT \cdot HAZ' \cdot LEFT'$	R1	1	0	1
L1	0	0	1	HAZ'	L2	0	1	1
L1	0	0	1	HAZ	LR3	1	0	0
L2	0	1	1	HAZ'	L3	0	1	0
L2	0	1	1	HAZ	LR3	1	0	0
L3	0	1	0	1	IDLE	0	0	0
R1	1	0	1	HAZ'	R2	1	1	1
R1	1	0	1	HAZ	LR3	1	0	0
R2	1	1	1	HAZ'	R3	1	1	0
R2	1	1	1	HAZ	LR3	1	0	0
R3	1	1	0	1	IDLE	0	0	0
LR3	1	0	0	1	IDLE	0	0	0

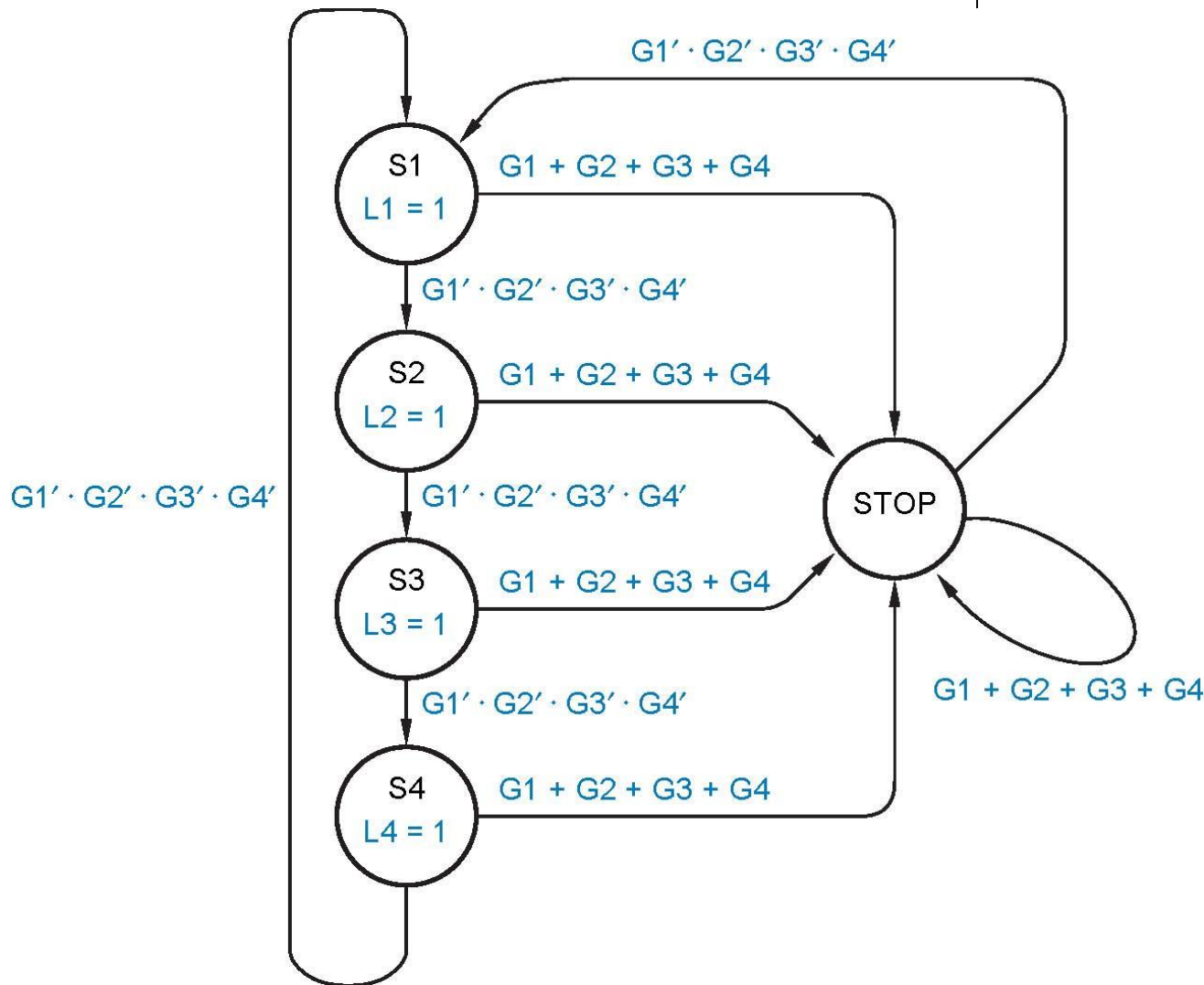
7.7 猜谜游戏



- 同步状态机
 - 4个按钮，与输入G1~G4联接
 - 1个ERR输出，与红色指示灯联接
 - 4个输出，与指示灯L1~L4联接，并与对应输入相邻
- 功能
 - 正常情况下，每经过1个时钟，模式旋转1个位置
 - 时钟频率4Hz
 - 猜谜：按下1个按钮，某输入Gi有效；若当前输入数Gi与时钟触发沿到来前有效的灯输出(状态)不同，则ERR有效，未能猜中；若相同为猜中。一旦完成1次猜测，游戏停止并且ERR输出会维持1个或多个时钟周期，直到输入Gi取消，游戏继续进行

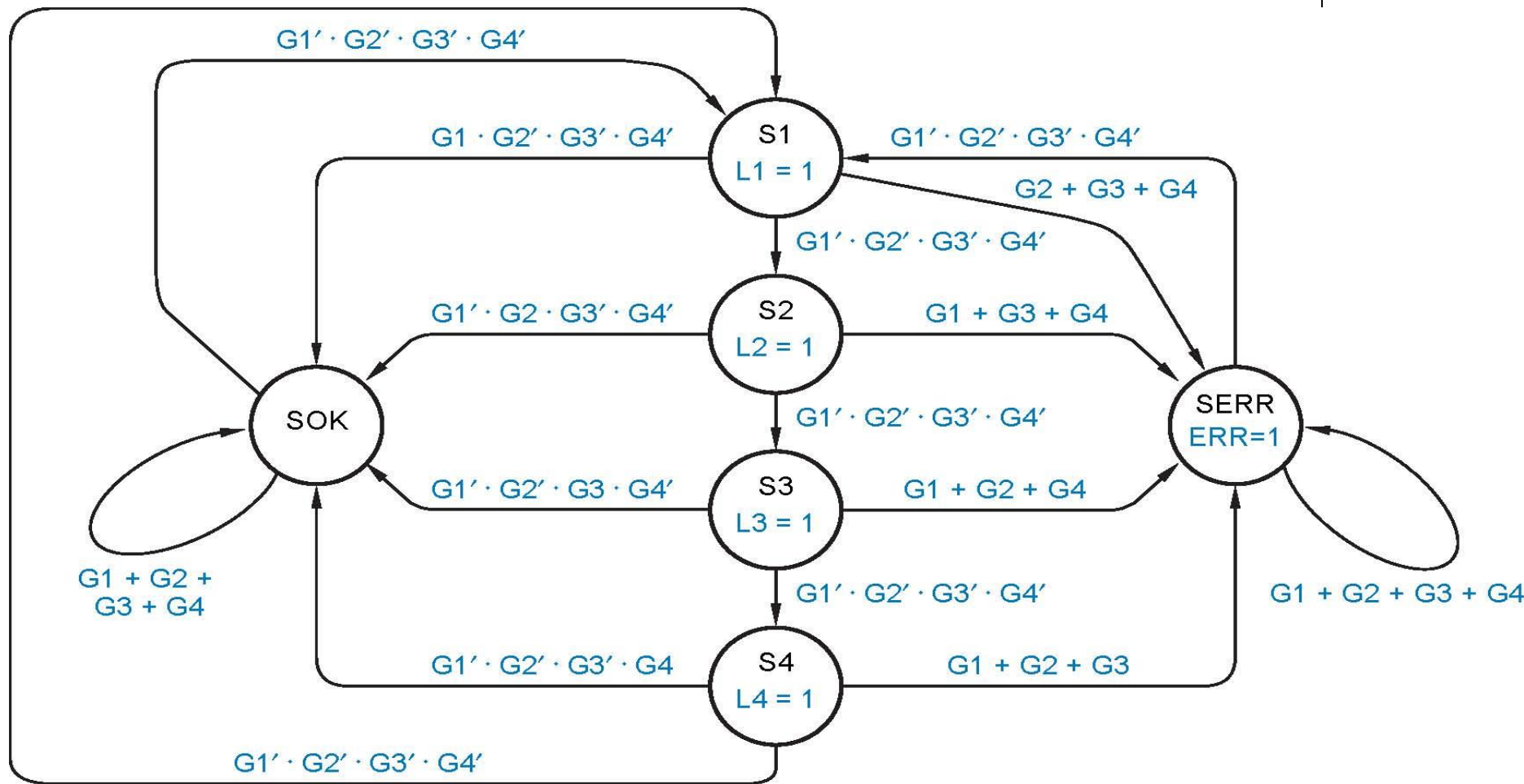


- 5状态
 - 无法指示猜测结果是否正确
- 使用者同时按下多个按键
 - 需克服并进入ERR状态





猜谜游戏



改进的猜谜游戏状态图



状态赋值：格雷码顺序对S1~S4编码

状态	Q2	Q1	Q0
S1	0	0	0
S2	0	0	1
S3	0	1	1
S4	0	1	0
SOK	1	0	0
SERR	1	0	1

Current State				Transition Expression	Next State				Output				
S	Q2	Q1	Q0		S*	Q2*	Q1*	Q0*	L1	L2	L3	L4	ERR
S1	0	0	0	$G1' \cdot G2' \cdot G3' \cdot G4'$	S2	0	0	1	1	0	0	0	0
S1	0	0	0	$G1 \cdot G2' \cdot G3' \cdot G4'$	SOK	1	0	0	1	0	0	0	0
S1	0	0	0	$G2 + G3 + G4$	SERR	1	0	1	1	0	0	0	0
S2	0	0	1	$G1' \cdot G2' \cdot G3' \cdot G4'$	S3	0	1	1	0	1	0	0	0
S2	0	0	1	$G1' \cdot G2 \cdot G3' \cdot G4'$	SOK	1	0	0	0	1	0	0	0
S2	0	0	1	$G1 + G3 + G4$	SERR	1	0	1	0	1	0	0	0
S3	0	1	1	$G1' \cdot G2' \cdot G3' \cdot G4'$	S4	0	1	0	0	0	1	0	0
S3	0	1	1	$G1' \cdot G2' \cdot G3 \cdot G4'$	SOK	1	0	0	0	0	1	0	0
S3	0	1	1	$G1 + G2 + G4$	SERR	1	0	1	0	0	1	0	0
S4	0	1	0	$G1' \cdot G2' \cdot G3' \cdot G4'$	S1	0	0	0	0	0	0	1	0
S4	0	1	0	$G1' \cdot G2' \cdot G3' \cdot G4$	SOK	1	0	0	0	0	0	1	0
S4	0	1	0	$G1 + G2 + G3$	SERR	1	0	1	0	0	0	1	0
SOK	1	0	0	$G1 + G2 + G3 + G4$	SOK	1	0	0	0	0	0	0	0
SOK	1	0	0	$G1' \cdot G2' \cdot G3' \cdot G4'$	S1	0	0	0	0	0	0	0	0
SERR	1	0	1	$G1 + G2 + G3 + G4$	SERR	1	0	1	0	0	0	0	1
SERR	1	0	1	$G1' \cdot G2' \cdot G3' \cdot G4'$	S1	0	0	0	0	0	0	0	1



- 状态方程
 - Q2
 - Q1
 - Q0
- 输出方程
 - $L1 = Q2' \cdot Q1' \cdot Q0'$
 - $L2 = Q2' \cdot Q1' \cdot Q0$
 - $L3 = Q2' \cdot Q1 \cdot Q0$
 - $L4 = Q2' \cdot Q1 \cdot Q0'$
 - $ERR = Q2 \cdot Q1' \cdot Q0$
- 未用状态：无关，隐含表示为0。



7.7.3 输出编码状态赋值



Current State						Transition Expression	Next State					
S	L1	L2	L3	L4	ERR		S*	L1*	L2*	L3*	L4*	ERR*
S1	1	0	0	0	0	$G1' \cdot G2' \cdot G3' \cdot G4'$	S2	0	1	0	0	0
S1	1	0	0	0	0	$G1 \cdot G2' \cdot G3' \cdot G4'$	SOK	0	0	0	0	0
S1	1	0	0	0	0	$G2 + G3 + G4$	SERR	0	0	0	0	1
S2	0	1	0	0	0	$G1' \cdot G2' \cdot G3' \cdot G4'$	S3	0	0	1	0	0
S2	0	1	0	0	0	$G1' \cdot G2 \cdot G3' \cdot G4'$	SOK	0	0	0	0	0
S2	0	1	0	0	0	$G1 + G3 + G4$	SERR	0	0	0	0	1
S3	0	0	1	0	0	$G1' \cdot G2' \cdot G3' \cdot G4'$	S4	0	0	0	1	0
S3	0	0	1	0	0	$G1' \cdot G2' \cdot G3 \cdot G4'$	SOK	0	0	0	0	0
S3	0	0	1	0	0	$G1 + G2 + G4$	SERR	0	0	0	0	1
S4	0	0	0	1	0	$G1' \cdot G2' \cdot G3' \cdot G4'$	S1	1	0	0	0	0
S4	0	0	0	1	0	$G1' \cdot G2' \cdot G3' \cdot G4$	SOK	0	0	0	0	0
S4	0	0	0	1	0	$G1 + G2 + G3$	SERR	0	0	0	0	1
SOK	0	0	0	0	0	$G1 + G2 + G3 + G4$	SOK	0	0	0	0	0
SOK	0	0	0	0	0	$G1' \cdot G2' \cdot G3' \cdot G4'$	S1	1	0	0	0	0
SERR	0	0	0	0	1	$G1 + G2 + G3 + G4$	SERR	0	0	0	0	1
SERR	0	0	0	0	1	$G1' \cdot G2' \cdot G3' \cdot G4'$	S1	1	0	0	0	0

用输出作为状态变量，有时能简化激励方程和输出方程。



7.7.3 输出编码状态赋值



- $L1^* = L1' \cdot L2' \cdot L3' \cdot L4 \cdot ERR' \cdot (G1' \cdot G2' \cdot G3' \cdot G4') + L1' \cdot L2' \cdot L3' \cdot L4' \cdot ERR' \cdot (G1' \cdot G2' \cdot G3' \cdot G4') + L1' \cdot L2' \cdot L3' \cdot L4' \cdot ERR \cdot (G1' \cdot G2' \cdot G3' \cdot G4')$
- $L2^* = L1 \cdot L2' \cdot L3' \cdot L4' \cdot ERR' \cdot (G1' \cdot G2' \cdot G3' \cdot G4')$
- $L3^* = L1' \cdot L2 \cdot L3' \cdot L4' \cdot ERR' \cdot (G1' \cdot G2' \cdot G3' \cdot G4')$
- $L4^* = L1' \cdot L2' \cdot L3 \cdot L4' \cdot ERR' \cdot (G1' \cdot G2' \cdot G3' \cdot G4')$
- $ERR^* = L1' \cdot L2' \cdot L3' \cdot L4' \cdot ERR' \cdot (G2 + G3 + G4) + L1' \cdot L2 \cdot L3' \cdot L4' \cdot ERR' \cdot (G1 + G3 + G4) + L1' \cdot L2' \cdot L3 \cdot L4' \cdot ERR' \cdot (G1 + G2 + G4) + L1' \cdot L2' \cdot L3' \cdot L4 \cdot ERR' \cdot (G1 + G2 + G3) + L1' \cdot L2' \cdot L3' \cdot L4' \cdot ERR \cdot (G1 + G2 + G3 + G4)$
- 方程组并不简单，但输出数变少



7.7.4 无关状态赋值



- 状态图有6个状态
 - 实际5个触发器有32个状态，未用状态作为“无关状态”
 - Karnaugh Map化简：只能处理简单问题
 - 计算机化简：许多综合软件容易处理大规模设计，但却无法处理“无关项”；
- 每一个未用的当前状态行为与附近的“正常”状态一样；机器不小心进入“未用状态”，可以自动回到“正常”状态
 - 自由度：通过引入“无关项”，允许对逻辑电路作一定的简化——Minimal cost approach



7.7.4 无关状态赋值

State	L1	L2	L3	L4	ERR
S1	1	x	x	x	x
S2	0	1	x	x	x
S3	0	0	1	x	x
S4	0	0	0	1	x
SOK	0	0	0	0	0
SERR	0	0	0	0	1

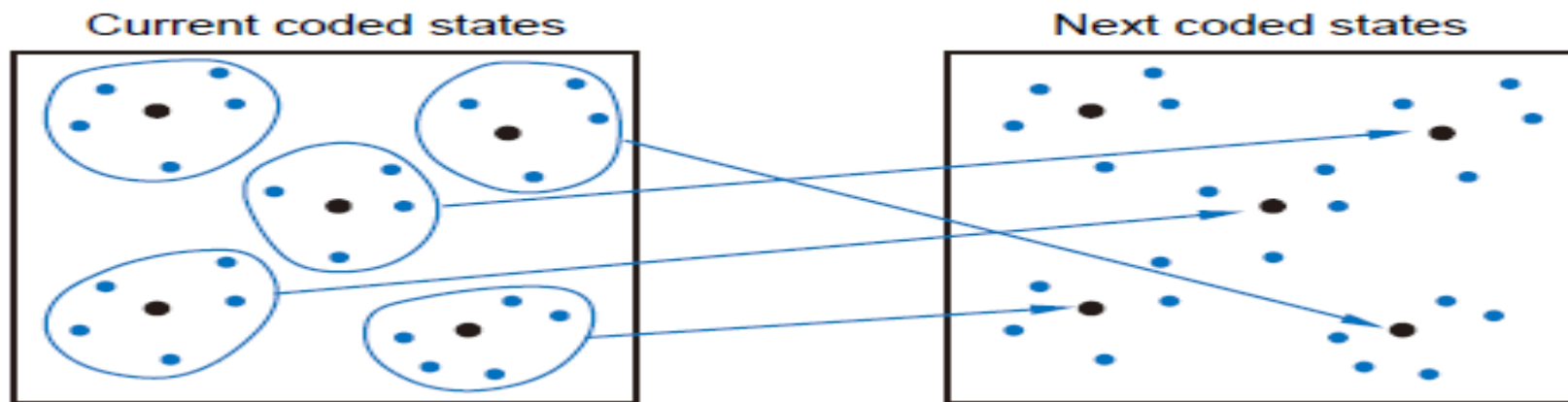
Current State						Transition Expression	Next State					
S	L1	L2	L3	L4	ERR		S*	L1*	L2*	L3*	L4*	ERR*
S	1	x	x	x	x	$G1' \cdot G2' \cdot G3' \cdot G4'$	S2	0	1	0	0	0
S	1	x	x	x	x	$G1 \cdot G2' \cdot G3' \cdot G4'$	SOK	0	0	0	0	0
S	1	x	x	x	x	$G2 + G3 + G4$	SERR	0	0	0	0	1
S2	0	1	x	x	x	$G1' \cdot G2' \cdot G3' \cdot G4'$	S3	0	0	1	0	0
S2	0	1	x	x	x	$G1' \cdot G2 \cdot G3' \cdot G4'$	SOK	0	0	0	0	0
S2	0	1	x	x	x	$G1 + G3 + G4$	SERR	0	0	0	0	1
S3	0	0	1	x	x	$G1' \cdot G2' \cdot G3' \cdot G4'$	S4	0	0	0	1	0
S3	0	0	1	x	x	$G1' \cdot G2' \cdot G3 \cdot G4'$	SOK	0	0	0	0	0
S3	0	0	1	x	x	$G1 + G2 + G4$	SERR	0	0	0	0	1
S4	0	0	0	1	x	$G1' \cdot G2' \cdot G3' \cdot G4'$	S1	1	0	0	0	0
S4	0	0	0	1	x	$G1' \cdot G2' \cdot G3' \cdot G4$	SOK	0	0	0	0	0
S4	0	0	0	1	x	$G1 + G2 + G3$	SERR	0	0	0	0	1
SOK	0	0	0	0	0	$G1 + G2 + G3 + G4$	SOK	0	0	0	0	0
SOK	0	0	0	0	0	$G1' \cdot G2' \cdot G3' \cdot G4'$	S1	1	0	0	0	0
SERR	0	0	0	0	1	$G1 + G2 + G3 + G4$	SERR	0	0	0	0	1
SERR	0	0	0	0	1	$G1' \cdot G2' \cdot G3' \cdot G4'$	S1	1	0	0	0	0



7.7.4 无关状态赋值



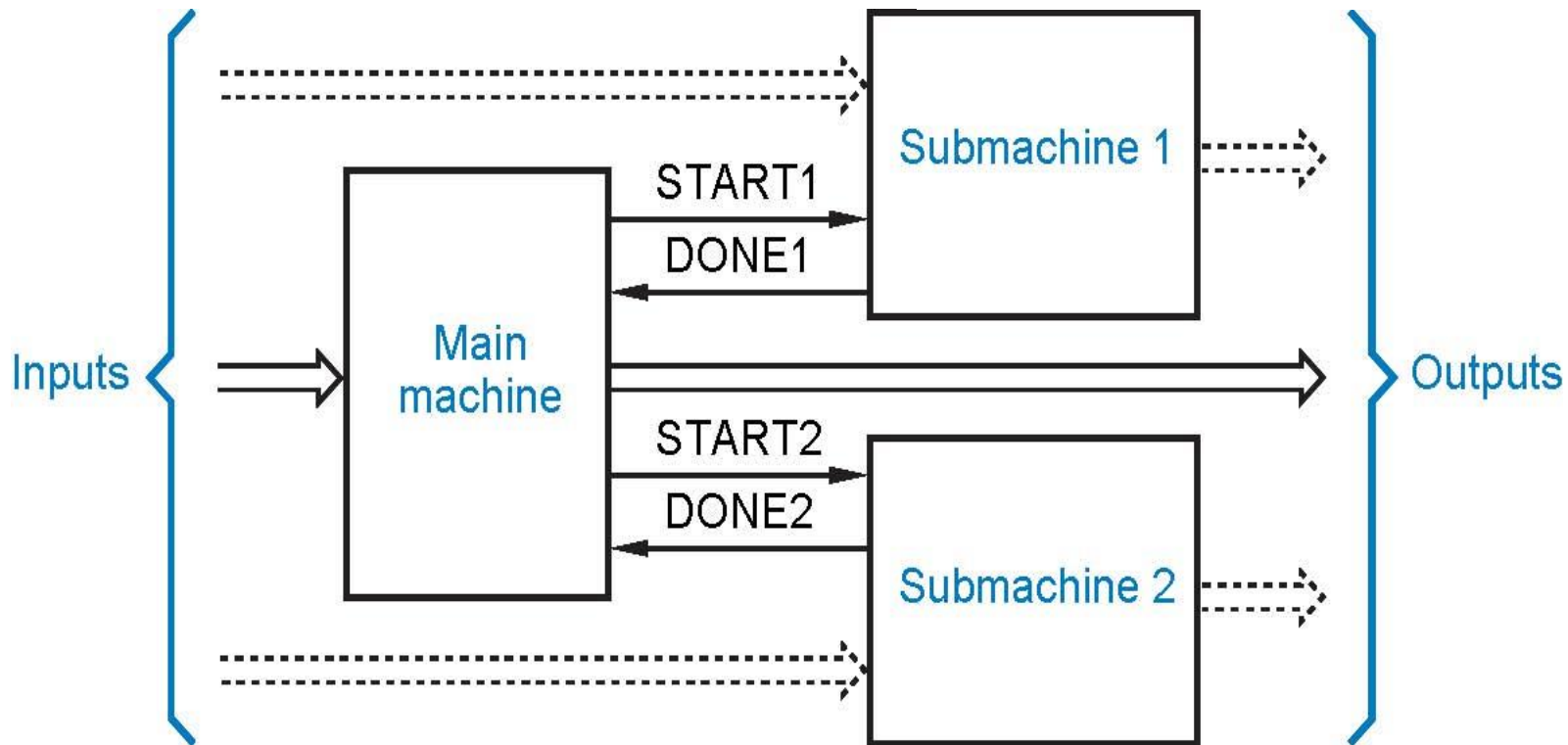
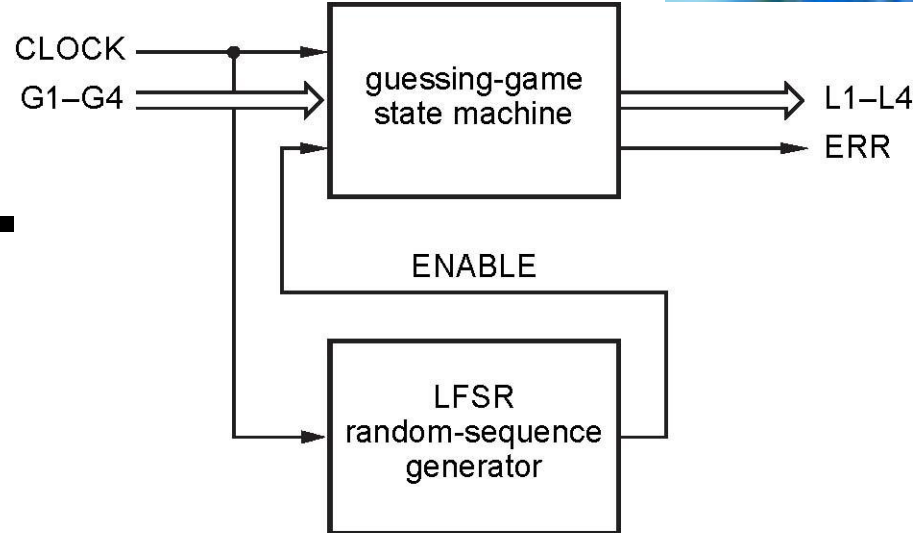
- $ERR^* = L1 \cdot (G2 + G3 + G4) + L1' \cdot L2 \cdot (G1 + G3 + G4) + L1' \cdot L2' \cdot L3 \cdot (G1 + G2 + G4) + L1' \cdot L2' \cdot L3' \cdot L4 \cdot (G1 + G2 + G3) + L1' \cdot L2' \cdot L3' \cdot L4' \cdot ERR \cdot (G1 + G2 + G3 + G4)$
- 输出简化
 - 最简“或-与式”只需5项，较“与-或式”需16项来得简单





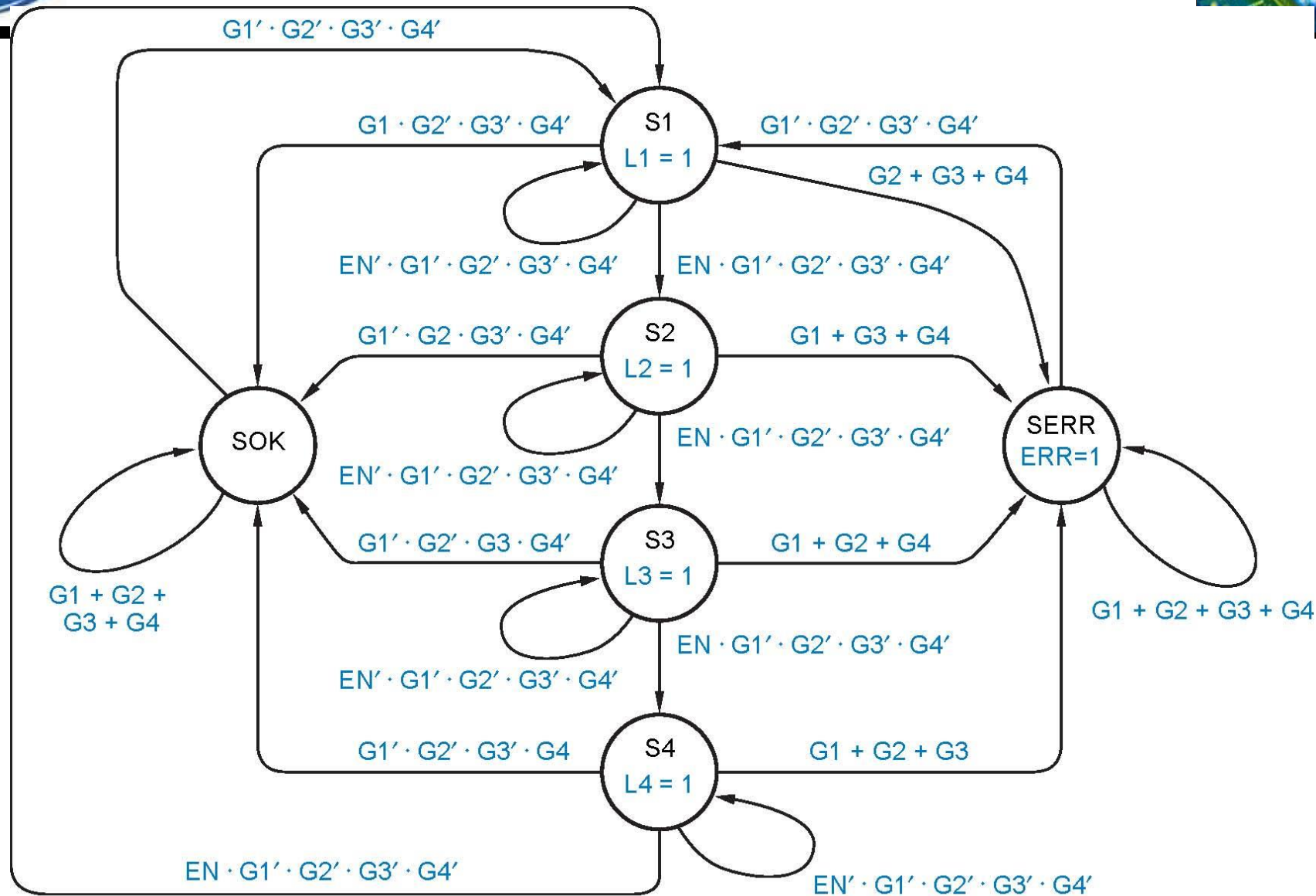
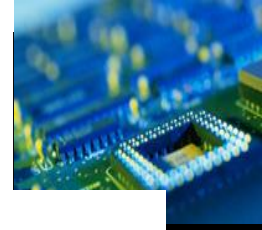
7.8 状态机分解

- 大型设计小型化

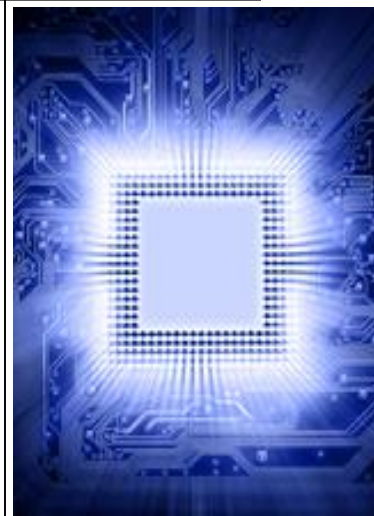




7.8 状态机分解



7.9 反馈时序电路





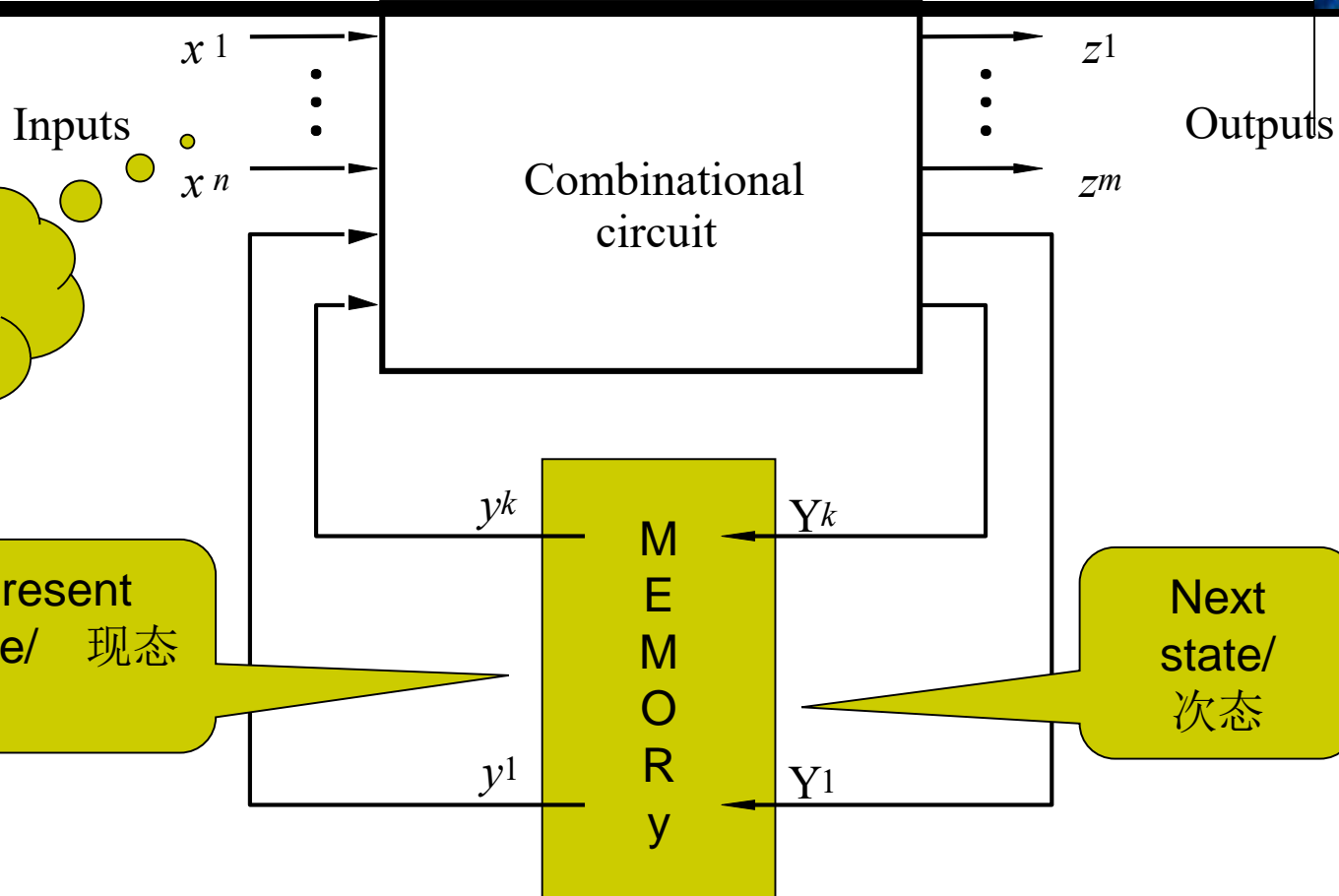
异步时序电路



- 异步时序电路特征
 - 状态变化不受“统一时钟信号控制”
- 依电路结构和输入信号形式不同，异步时序电路分类：
 - 脉冲型Pulsed Asynchronous Circuit(脉冲信号)
 - 记忆电路为“触发器”
 - Mealy型、Moore型
 - 电平型Level Asynchronous Circuit(电平信号)
 - 记忆电路为“反馈加延时”
 - Mealy型、Moore型



脉冲型异步时序电路

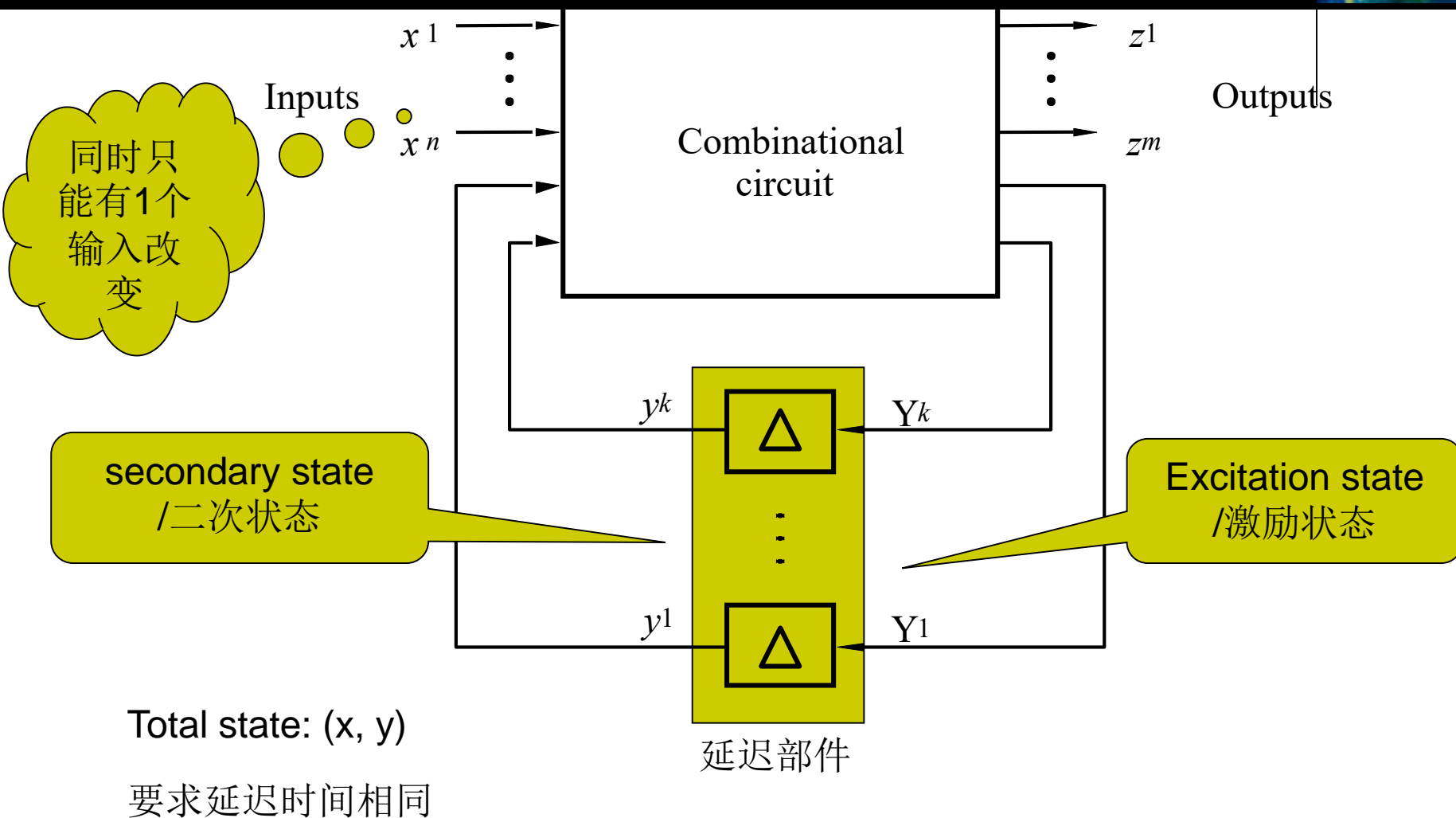


同时只能
有1个输入
起作用

非时钟型
记忆单元/
锁存器



电平型异步时序电路

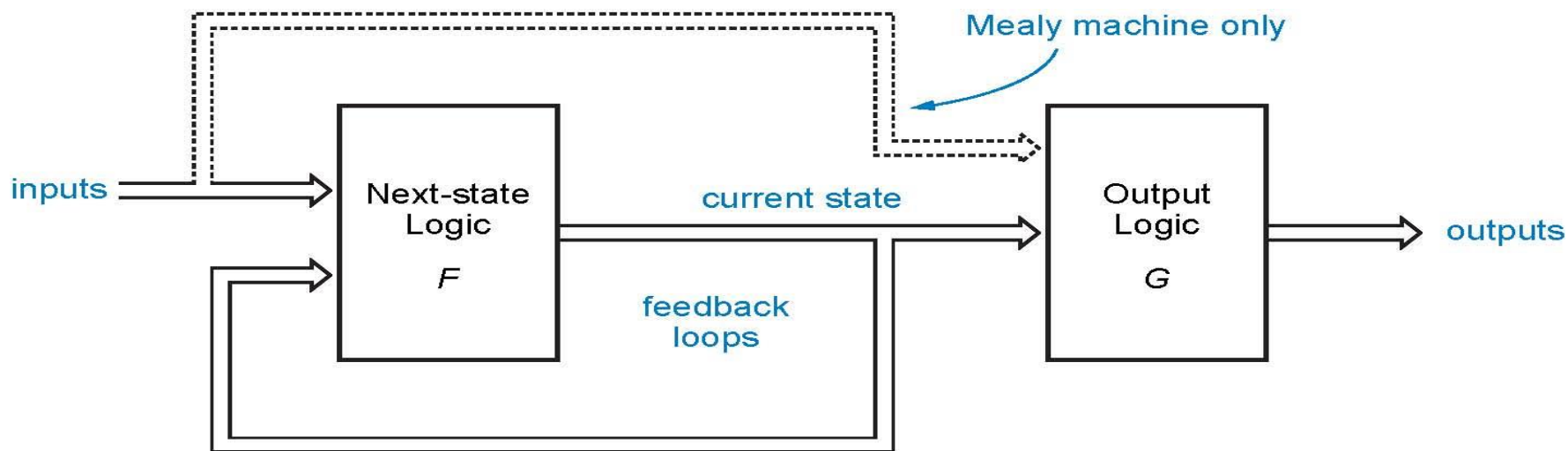




7.9 反馈时序电路



- 双稳态电路、锁存器和触发器等都是反馈时序电路。反馈回路是记忆元件，存储0或1。
- 基本模式电路，对输入信号的约束：
 - 不允许两个或两个以上输入同时变化(每时刻仅允许1个输入发生变化)
 - 仅当电路处于稳态时，允许输入信号发生变化





反馈时序电路的分析步骤



- 根据给定的逻辑电路图，找出它的流程表，作出输入序列的时间图，并说明电路的逻辑功能。

。

其步骤如下：

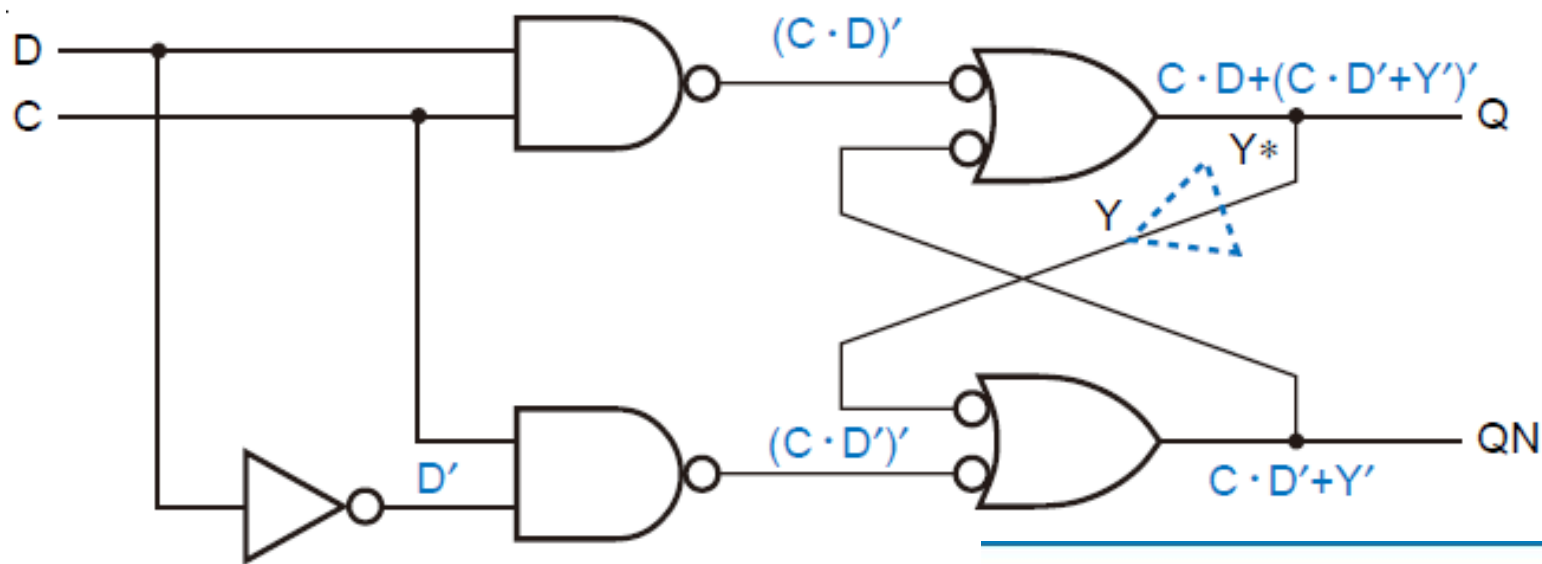
- 切断反馈路径，写出激励函数和输出函数
- 列出状态转移表
- 作出状态图和时序(间)图
- 说明电路的逻辑功能



反馈时序电路的分析



- 反馈时序电路的分析：断开反馈，加上延时。



$$\begin{aligned} Y^* &= (C \cdot D) + (C \cdot D' + Y')' \\ &= C \cdot D + C' \cdot Y + D \cdot Y \end{aligned}$$

		C D			
Y		00	01	11	10
0		0	0	1	0
1		1	1	1	0
		Y*			



反馈时序电路的分析



- 现态和次态只是在时间上有延迟，经过一段时间后，现态和次态的值趋于一致。
- 总态**Total State**：用来描述异步时序机的行为。包括输入状态I和次态S，并记为(I, S)。
- 稳定状态：如某一现态y在输入x发生变化时，若次态Y和现态y的值相同，则称次态Y为稳定状态；若次态Y和现态y的值不等，则称进入的次态Y为不稳定状态。不稳定状态是不能保持不变，经过一段延迟时间后，现态和次态趋于一致，进入稳定状态，则称为状态迁移。
- 状态表：状态命名，且将稳定的总状态用圆圈圈住。
- 输出方程：

$$Q = C \cdot D + C' \cdot Y + D \cdot Y$$
$$QN = C \cdot D' + Y'$$

S	C D			
	00	01	11	10
S0	(S0)	(S0)	S1	(S0)
S1	(S1)	(S1)	(S1)	S0

S*



反馈时序电路的分析



- 总态可以是稳定的也可以是不稳定的，不稳定的总态要发生状态迁移，而达到一个稳定的状态或循环不稳态。稳定的总态，如果输入不改变，则始终保存稳定状态，不会发生状态迁移。
- 异步电路的基本工作方式是保证电路稳定工作，使电路状态的转移是可以预测的。
- 输入状态的改变仅能引起次态在状态表作相邻方格的水平移动。
- 二次状态的改变则引起次态在状态表中作垂直方向的移动。



反馈时序电路的分析

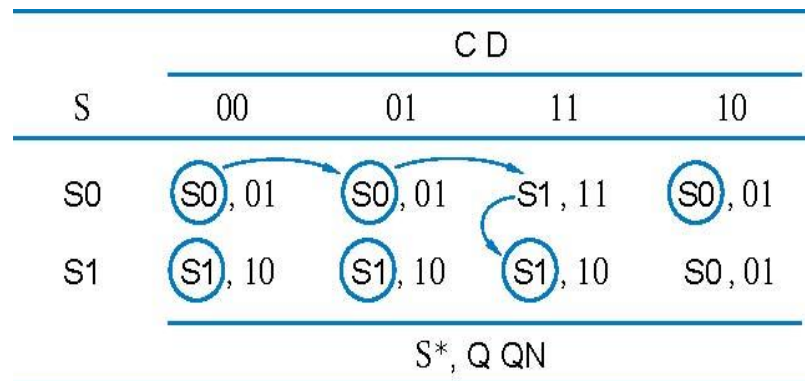


- 通过状态和输出表来分析电路特性

- 从稳定总状态开始:

$S=S0, CD=00$

$S0/00 \rightarrow S0/01 \rightarrow S1/11 \rightarrow S1/10$



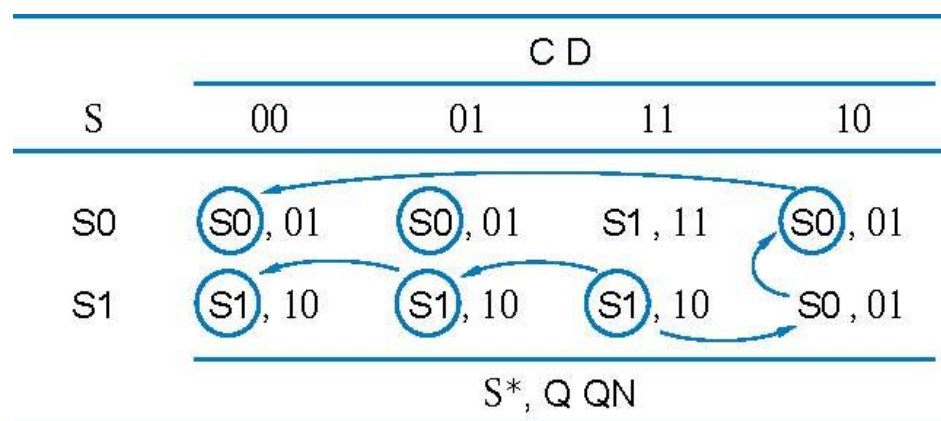
- 对任何期望的输入序列所引起的电路行为进行跟踪

。

- 输入同时变化

- $S1/11 \rightarrow */00$

- 亚稳定

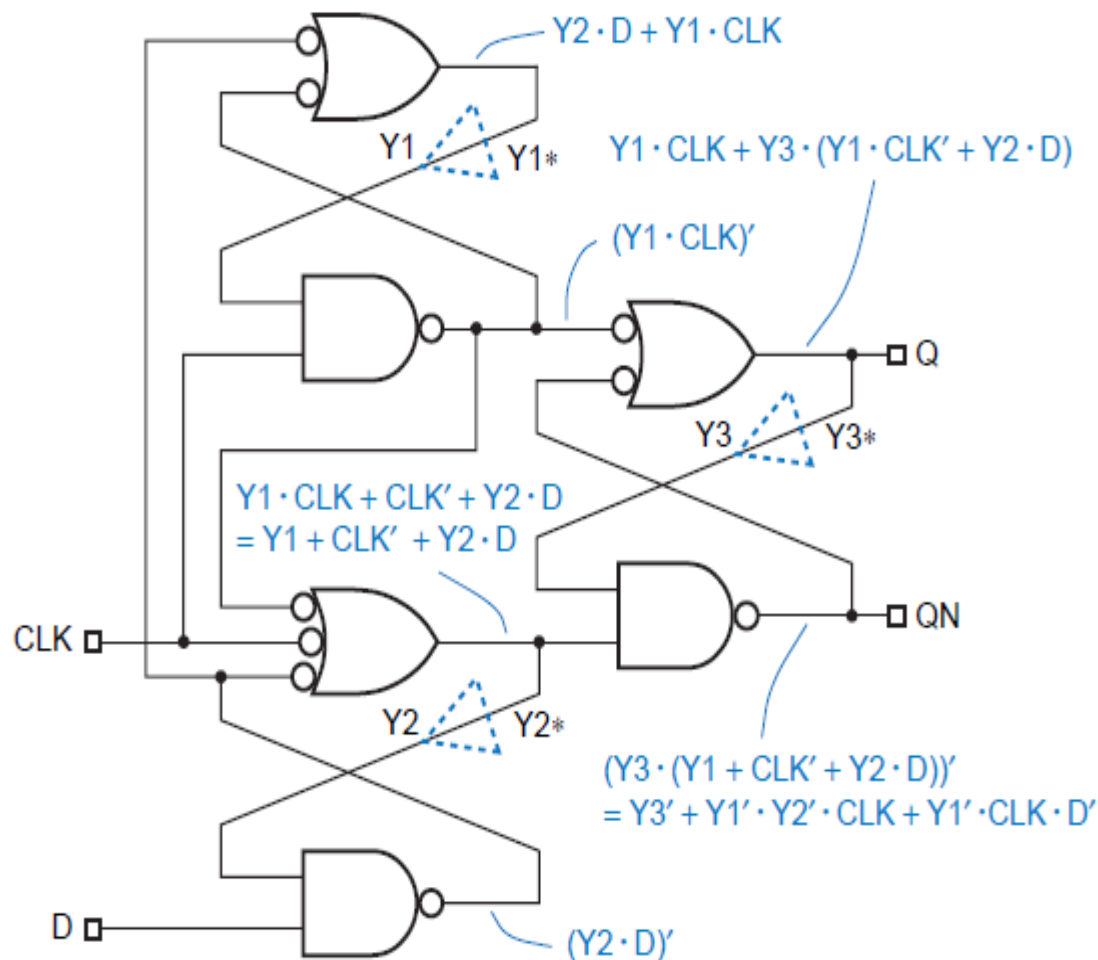




7.9.2 分析多反馈回路的电路



- 多反馈回路中，必须断开所有的反馈，设置虚构的缓冲器和状态变量。
- 最小割集(cut set)
 - 断点数最少的集合
- 3条反馈回路
 - 断点数越少，
 - 分析所用状态数就越少





$$Y1^* = Y2 \cdot D + Y1 \cdot CLK$$

$$Y2^* = Y1 + CLK' + Y2 \cdot D$$

$$Y3^* = Y1 \cdot CLK + Y1 \cdot Y3 + Y3 \cdot CLK' + Y2 \cdot Y3 \cdot D$$

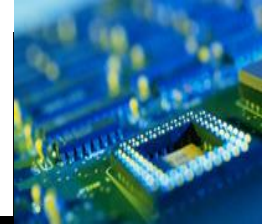
$$Q = Y1 \cdot CLK + Y1 \cdot Y3 + Y3 \cdot CLK' + Y2 \cdot Y3 \cdot D$$

$$QN = Y3' + Y1' \cdot Y2' \cdot CLK + Y1' \cdot CLK \cdot D'$$

Y1 Y2 Y3	CLK D			
	00	01	11	10
000	010	010	000	000
001	011	011	000	000
010	010	110	110	000
011	011	111	111	000
100	010	010	111	111
101	011	011	111	111
110	010	110	111	111
111	011	111	111	111
Y1* Y2* Y3*				



竞争races



- 一个输入信号的变化，引起多个内部状态变量改变，称之为发生了竞争
 - 非临界竞争(Noncritical race)，最终状态与状态变量变化顺序无关，结果可预测。
 - 临界竞争(Critical race)，最终状态取决于状态变量变化顺序和速度，结果不可预测。
- 对竞争的处理
 - 允许非临界竞争
 - 避免临界竞争



竞争races

CLK从0到1的变化, 状态可能会经过001或010, 最终结果到达000或111

非临界竞争

CLK D

Y1 Y2 Y3

00

01

11

10

000

010

010

000

000

001

011

011

000

000

010

010

110

110

000

011

011

111

111

000

Y1* Y2* Y3*

CLK从0到1的变化, 状态从011暂时经过001或010, 最终结果都是到达000

临界竞争

CLK D

Y1 Y2 Y3

00

01

11

10

000

010

010

000

000

001

011

011

000

000

010

010

110

110

110

011

011

111

111

000

100

010

010

111

111

101

011

011

111

111

110

010

110

111

111

111

011

111

111

111

Y1* Y2* Y3*



7.9.4 状态表与流程表



- 状态表：状态命名，且将稳定的总状态用圆圈圈住。
- 跳程hop：单个输入变量变化时，电路达到新的稳定总状态时所发生的不稳定的状态。
- 流程表**Flow table**：由现态和在不同输入条件下的次态及输出组成。删除状态表中的跳程，只表示出每次转移过程的最终目标，去除未用内部状态的那些行，去掉那些稳定的总状态在单个输入变化时从不会到达的下一状态项。



7.9.4 状态表与流程表



S	CLK D			
	00	01	11	10
S0	S2 , 01	S2 , 01	(S0) , 01	(S0) , 01
S1	S3 , 10	S3 , 10	S0 , 10	S0 , 10
S2	(S2) , 01	S6 , 01	S6 , 01	S0 , 01
S3	(S3) , 10	S7 , 10	S7 , 10	S0 , 01
S4	S2 , 01	S2 , 01	S7 , 11	S7 , 11
S5	S3 , 10	S3 , 10	S7 , 10	S7 , 10
S6	S2 , 01	(S6) , 01	S7 , 11	S7 , 11
S7	S3 , 10	(S7) , 10	(S7) , 10	(S7) , 10
S* , Q QN				

状态/输出表

S	CLK D			
	00	01	11	10
S0	S2 , 01	S6 , 01	(S0) , 01	(S0) , 01
S2	(S2) , 01	S6 , 01	— , —	S0 , 10
S3	(S3) , 10	S7 , 10	— , —	S0 , 01
S6	S2 , 01	(S6) , 01	S7 , 11	— , —
S7	S3 , 10	(S7) , 10	(S7) , 10	(S7) , 10
S* , Q QN				

流程/输出表



7.9.4 状态表与流程表



- 起始总稳态
S0/10, Q=0, CLK=1, D=0
 - D: 0 → 1
 - CLK: 1 → 0
 - 输出值未变
 - CLK: 0 → 1, 状态返回 S0 或 S7
- S0 和 S2 相容, 合并为 SB。

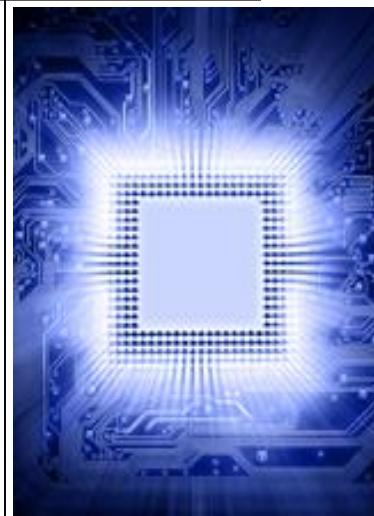
CLK D				
S	00	01	11	10
S0	S2, 01	S6, 01	(S0), 01	(S0), 01

CLK D				
S	00	01	11	10
SB	(SB), 01	S6, 01	(SB), 01	(SB), 01
S3	(S3), 10	S7, 10	—, —	SB, 01
S6	SB, 01	(S6), 01	S7, 11	—, —
S7	S3, 10	(S7), 10	(S7), 10	(S7), 10

S*, Q QN				
----------	--	--	--	--

7.10

反馈时序电路设计





反馈时序电路设计



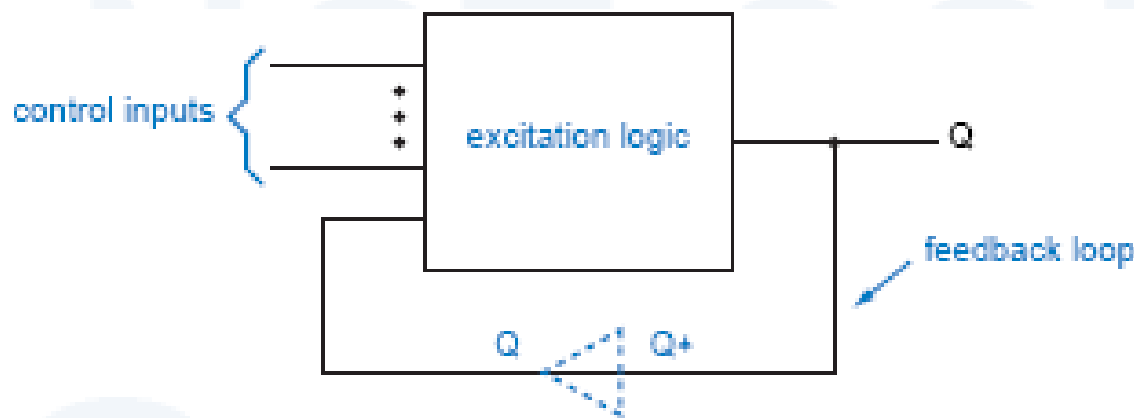
- 主要设计步骤：
 1. 根据逻辑要求，建立原始流程表
 2. 将原始状态表简化，得到最简流程表
 3. 对最简流程表进行状态分配
 4. 建立激励表和输出表
 5. 列出激励函数和输出函数表达式
 6. 画出逻辑电路图
- 原始流程表: 每一行含有一个稳态
- 每个状态决定于上一个状态和输入



锁存器的设计



- 只有一个反馈回路的电路
- 激励方程的一般形式：
 - $Q^* = (\text{强制项}) + (\text{保持项}) Q$

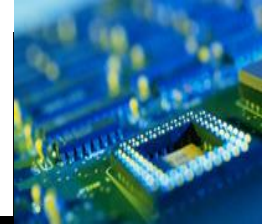


$$Q^* = S + R' \cdot Q$$

$$Q^* = C \cdot D + C' \cdot Q$$



锁存器的设计



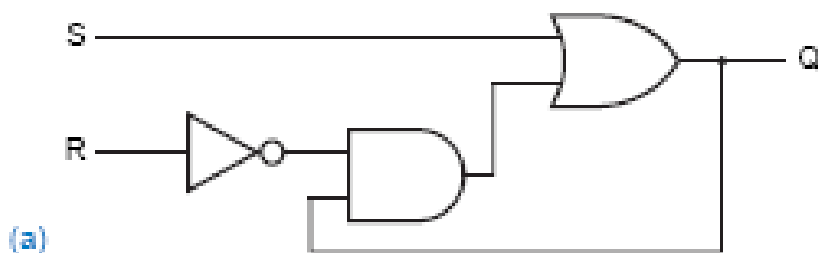
- 只有一个反馈回路的电路

- 激励方程的一般形式：

$$Q^* = S + R' \cdot Q$$

$$Q^* = C \cdot D + C' \cdot Q$$

- $Q^* = (\text{强制项}) + (\text{保持项}) Q$

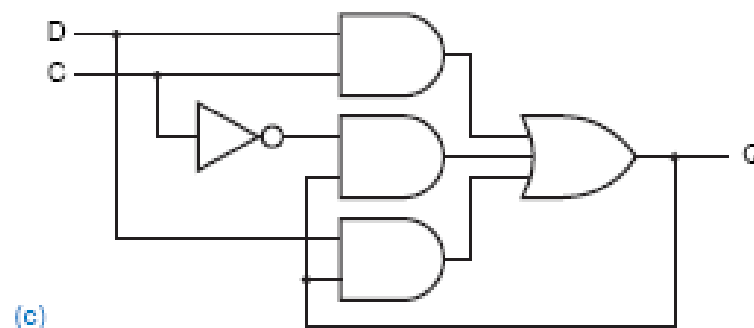
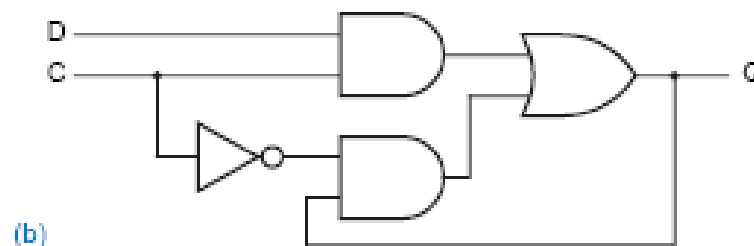


Latch circuits:

(a) S-R latch;

(b) unreliable D latch;

(c) hazard-free D latch.





锁存器的设计



- 只有一个反馈回路的电路
- 激励方程的一般形式：
 - $Q^* = (\text{强制项}) + (\text{保持项}) Q$

$$Q^* = S + R' \cdot Q$$

$$Q^* = C \cdot D + C' \cdot Q$$

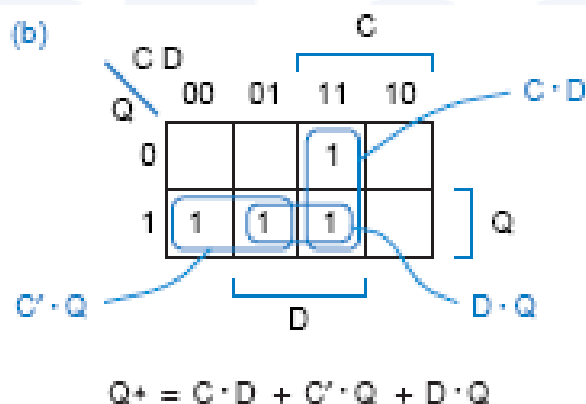
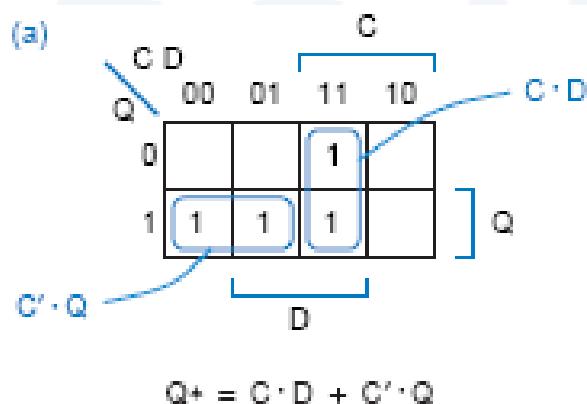


Figure 7-89
Karnaugh maps for
D-latch excitation
functions: (a) original,
containing a static-1
hazard; (b) hazard
eliminated.



设计基本模式流程表



- 原始流程表：每一行都只有一个稳定的总态。
 - 输出是状态的函数。
- 通常需要简化。
- 设计问题：

Design a feedback sequential circuit with two inputs, P (pulse) and R (reset), and a single output Z that is normally 0. The output should be set to 1 whenever a 0-to-1 transition occurs on P, and should be reset to 0 whenever R is 1. Typical functional behavior is shown in Figure 7-90.

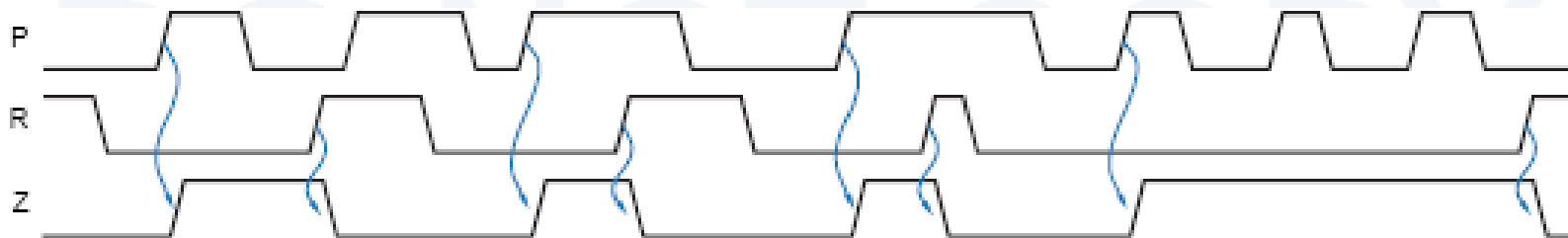


Figure 7-90 Typical functional behavior of a pulse-catching circuit.



设计基本模式流程表



- 原始流程表：每一行都只有一个稳定的总态。

Meaning	S	P R				Z
		00	01	11	10	
Idle, waiting for pulse	IDLE	IDLE	RES1	—	PLS1	0
Reset, no pulse	RES1	IDLE	RES1	RES2	—	0
Got pulse, output on	PLS1	PLS2	—	RES2	PLS1	1
Reset, got pulse	RES2	—	RES1	RES2	PLSN	0
Pulse gone, output on	PLS2	PLS2	RES1	—	PLS1	1
Got pulse, but output off	PLSN	IDLE	—	RES2	PLSN	0
S*						

Figure 7-91 Primitive flow table for pulse-catching circuit.



设计基本模式流程表



- 原始流程表最小化

S	P R				Z
	00	01	11	10	
IDLE	IDLE	IDLE	RES	PLS	0
PLS	PLS	IDLE	RES	PLS	1
RES	IDLE	IDLE	RES	RES	0
S*					



设计基本模式流程表



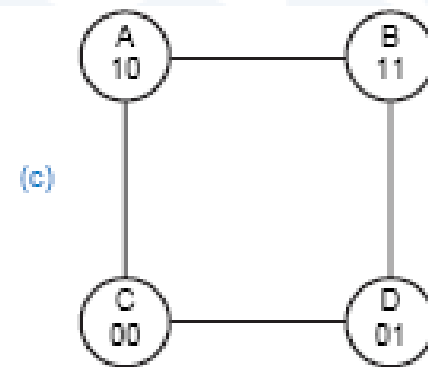
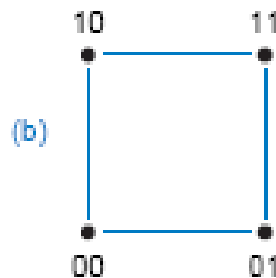
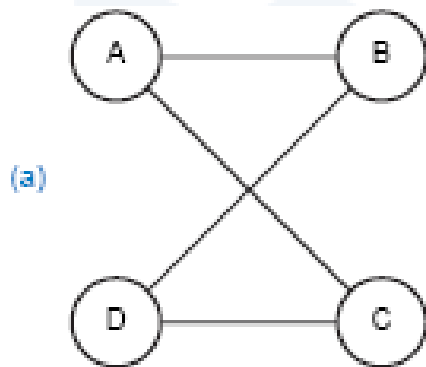
- 无竞争状态赋值法

- 例:

S	X Y			
	00	01	11	10
A	A	B	A	B
B	B	B	D	B
C	C	A	A	C
D	D	B	D	C

- 方法:

- 画状态相邻图
- 相邻状态的编码只能有一位不同

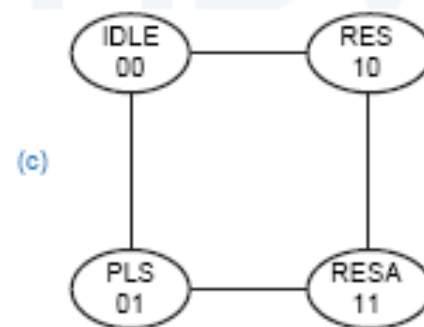
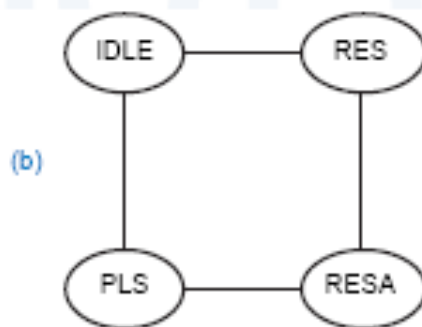
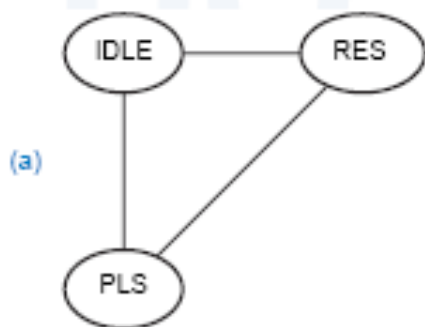




设计基本模式流程表



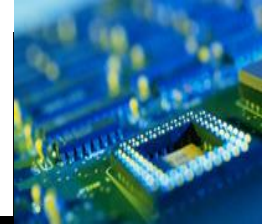
- 无竞争状态赋值法
 - 脉冲捕捉电路相邻图:



S	P R				Z
	00	01	11	10	
IDLE	IDLE	IDLE	RES	PLS	0
PLS	PLS	IDLE	RESA	PLS	1
RESA	--	--	RES	--	-
RES	IDLE	IDLE	RES	RES	0
S*					

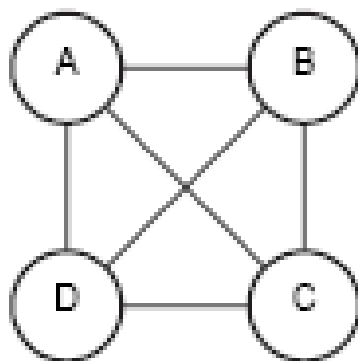


设计基本模式流程表

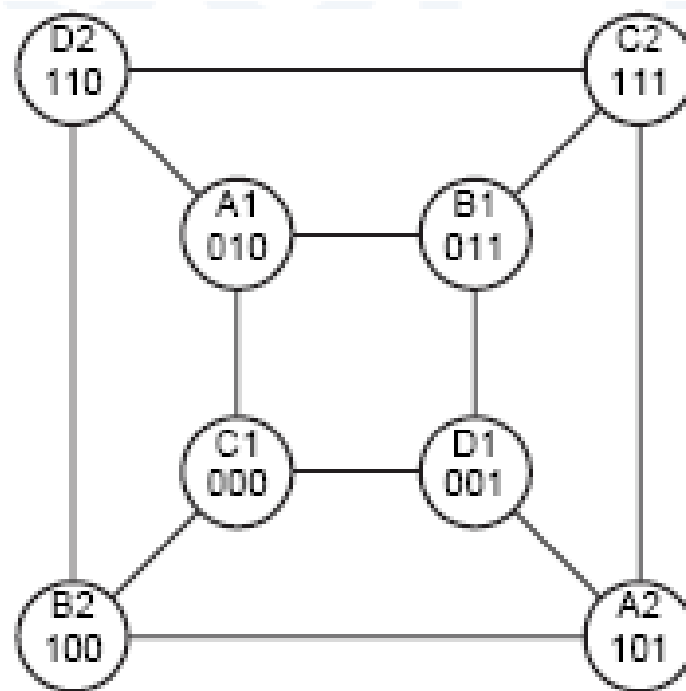


- 无竞争状态赋值法
 - 其它相邻图的处理方法：

(a)



(b)





设计基本模式流程表



- 激励方程:

Y1 Y2	P R				Z
	00	01	11	10	
00	00	00	10	01	0
01	01	00	11	01	1
11	--	--	10	—	—
10	00	00	10	10	0
Y1* Y2*					

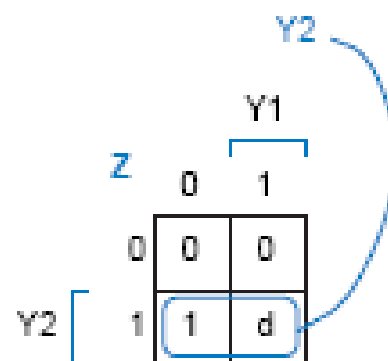
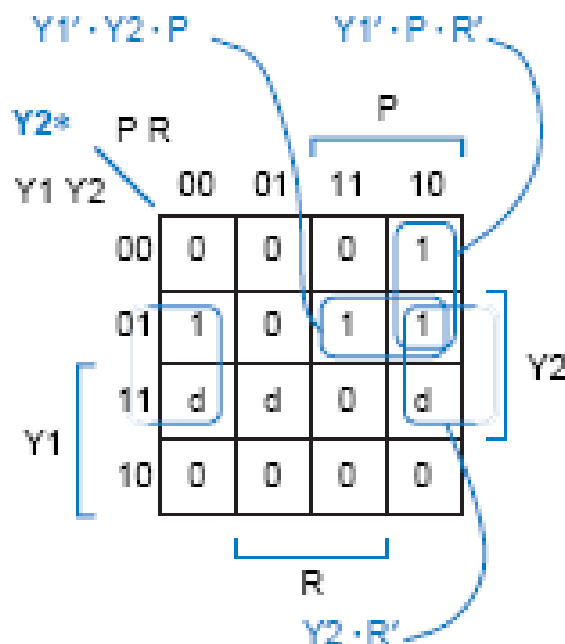
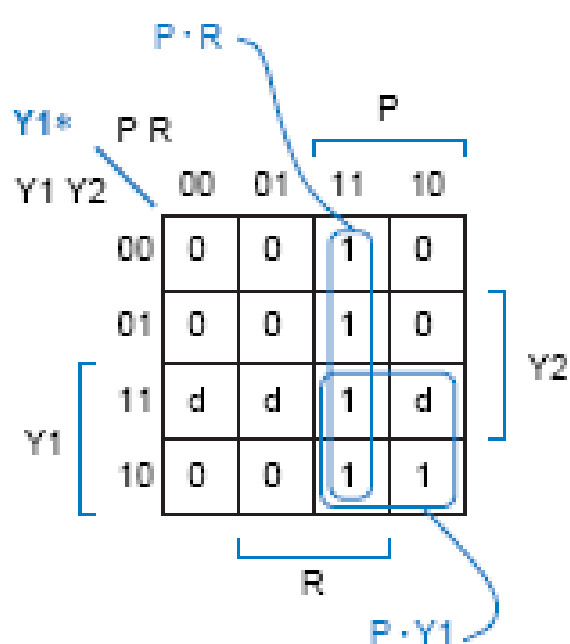
Figure 7-98
Race-free transition
table for the pulse-
catching circuit.



设计基本模式流程表



- 激励方程:



$$Y1^* = P \cdot R + P \cdot Y1$$

$$Y2^* = Y2 \cdot R' + Y1' \cdot Y2 \cdot P + Y1' \cdot P \cdot R'$$

$$Z = Y2$$



本质冒险



- 基本模式电路要能正常工作需要满足5个条件：

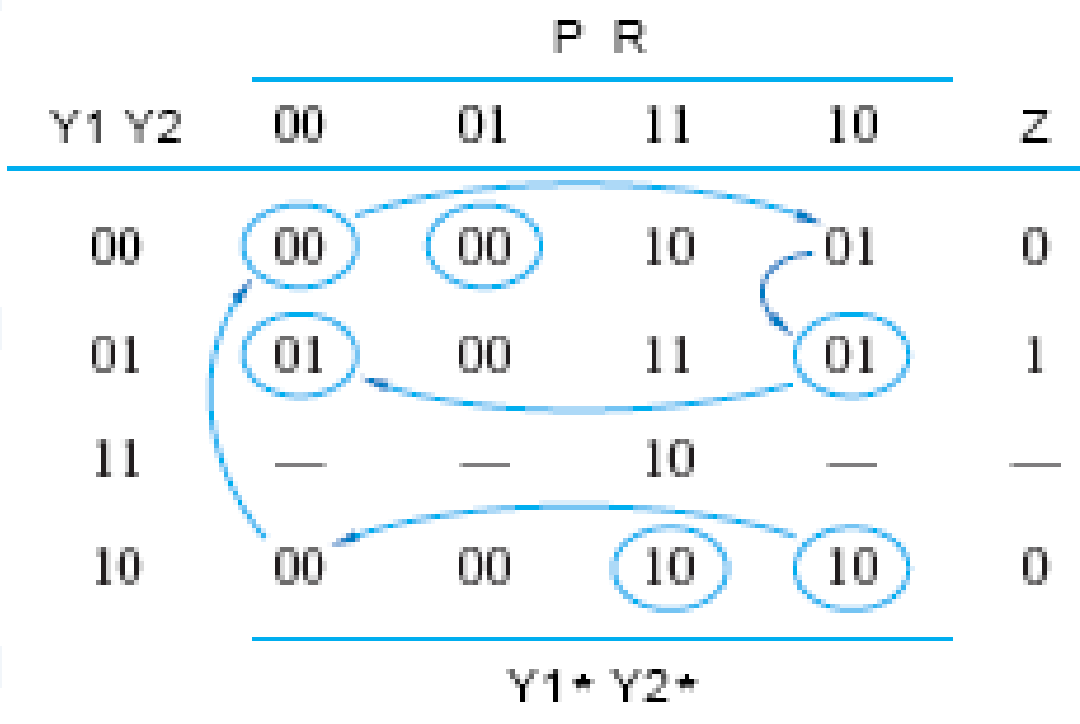
1. Only one input signal may change at a time, with a minimum bound between successive input changes.
2. There must be a maximum propagation delay through the excitation logic and feedback paths; this maximum must be less than the time between successive input changes.
3. The state assignment (transition table) must be free of critical races.
4. The excitation logic must be hazard free.
5. The minimum propagation delay through the excitation logic and feedback paths must be greater than the maximum timing skew through the "input logic."



本质冒险



- 本质冒险：当输入信号变化时，电路进入错误状态的可能性。
 - 如果最终状态变量的变化被传回到激励电路输入端之前，输入的变化产生错误。
- 通俗定义：
 - 如果从状态S出果状态不一致，





- 例子：

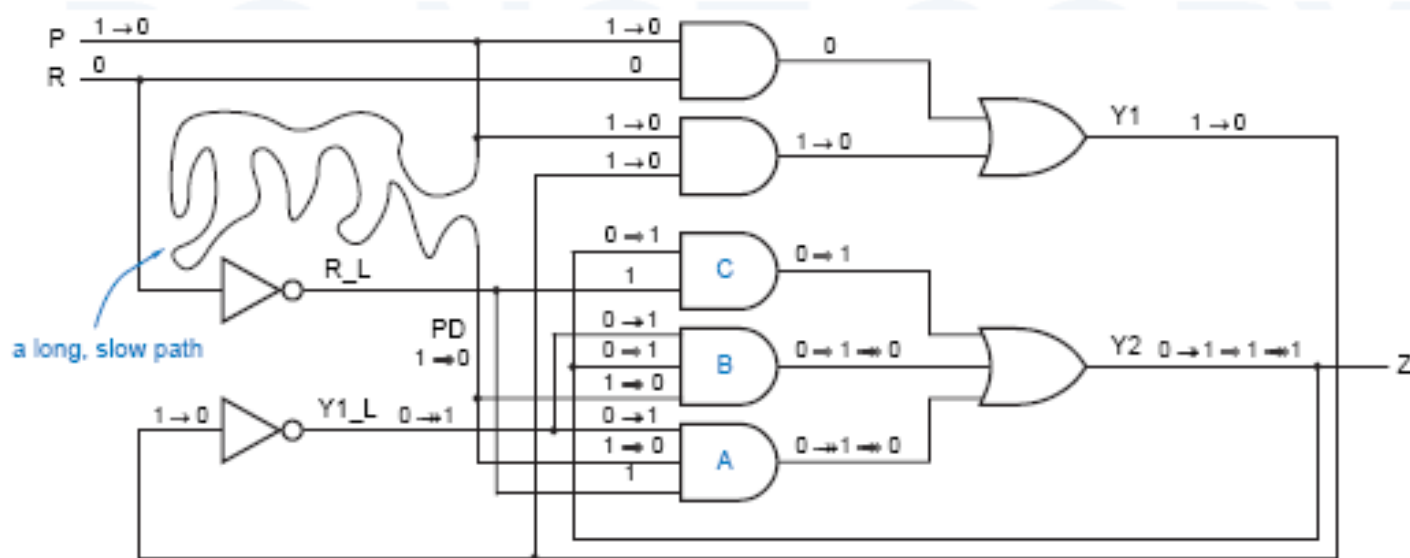


Figure 7-101 Physical conditions in pulse-catching circuit for exhibiting an essential hazard.

- 本质冒险是电路中固有的问题，可通过增加延迟解决问题。



- 7.4 7.7 7.12 7.19 7.21b 7.21c
- 7.41 7.44 7.46
- 7.49 7.54 7.59 7.66
- 7.70 7.87 7.92