# 第九讲 有限状态机分析
# Finite State Machine Analysis

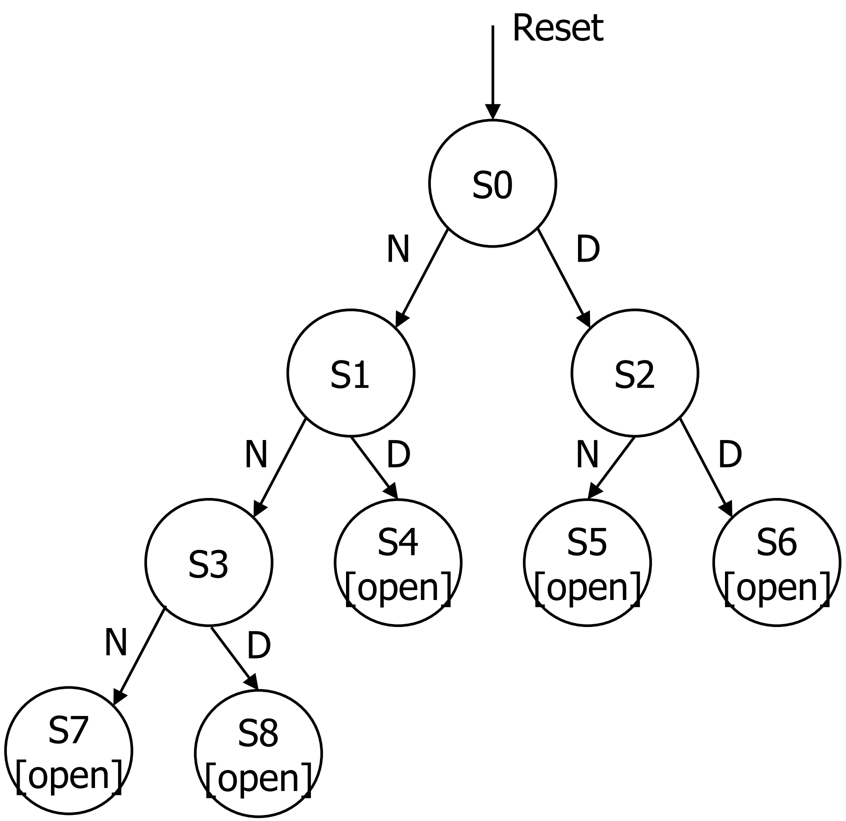
佟冬
tongdong@pku.edu.cn

微处理器研究开发中心（MPRC）
计算机科学技术系

北京大学

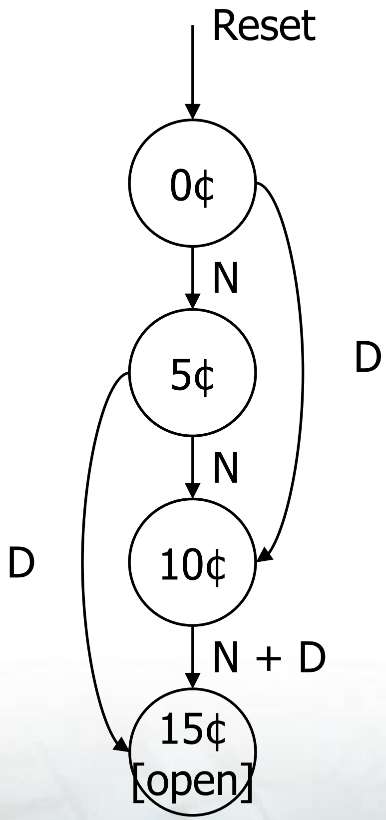
# 课程回顾：自动售货机

- 适当的抽象表示
  - 列出典型的硬币输入序列：
    - 3个五分
    - 一个五分接着一个十分
    - 一个十分接着一个五分
    - 两个十分
  - 画出状态图：
    - 输入: N, D, reset
    - 输出: open chute
  - 假设：
    - 每个时钟周期已能投入一个5分或者一个10分
    - 如果N = D = 0 （没有硬币）状态不变



# 自动售货机

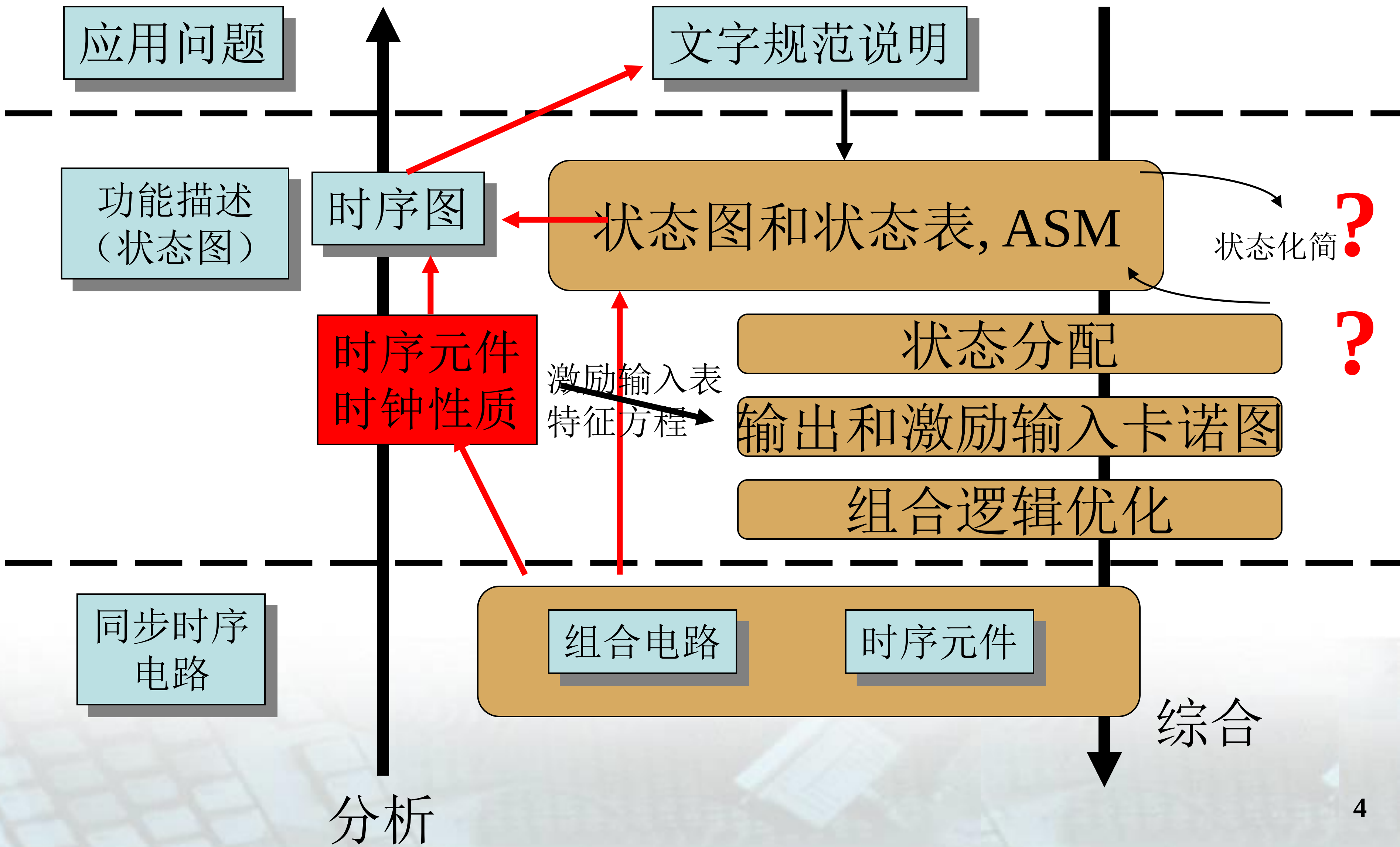
- 使状态个数最小，即可能的复用状态




| present state | inputs D | inputs N | next state | output open |
|---|---|---|---|---|
| 0¢ | 0 | 0 | 0¢ | 0 |
| | 0 | 1 | 5¢ | 0 |
| | 1 | 0 | 10¢ | 0 |
| | 1 | 1 | – | – |
| 5¢ | 0 | 0 | 5¢ | 0 |
| | 0 | 1 | 10¢ | 0 |
| | 1 | 0 | 15¢ | 0 |
| | 1 | 1 | – | – |
| 10¢ | 0 | 0 | 10¢ | 0 |
| | 0 | 1 | 15¢ | 0 |
| | 1 | 0 | 15¢ | 0 |
| | 1 | 1 | – | – |
| 15¢ | – | – | 15¢ | 1 |

symbolic state table

# 时序电路分析与综合总结



# 本章主要内容

- 冗余状态
- 完全确定状态机的化简
- 非完全确定状态机的化简
- 状态分配方法

# 1 冗余状态

- 消除冗余状态的重要性
  - 成本：存储元件的数目和状态的数目直接相关。
    - Binary编码：n = logN
    - One-hot编码：n = N
  - 复杂性：电路中的状态越多，设计实现越复杂。
  - 辅助故障分析：诊断程序一般都默认为电路中没有冗余状态。

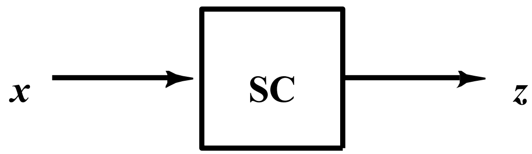
# 1.1 状态等价性

- 定义1：完全确定的时序电路中状态$S_1, S_2, \ldots, S_j$被称为等价的，当且仅当对于任意的输入序列，将$S_1, S_2, \ldots, S_j$中的任意状态作为初始状态，电路的输出序列都是相同的。
- 定义2：设$S_i$和$S_j$是完全确定时序电路的两个状态，$S_k$和$S_l$是在输入$I_p$时$S_i$和$S_j$的下一个状态，$S_i$和$S_j$是等价的当且仅当对于每一可能的$I_p$满足下列的条件：
  - 1. $S_i$和$S_j$的输出相同；
  - 2. 下一个状态$S_k$和$S_l$是等价的。

# 定义1和定义2等价

- 充分性：
  - 假设定义2不成立，则定义1不成立。
- 必要性
  - 假设定义1不成立，则定义2的两个条件必然有一个不成立，所以定义2不成立。
- 因此：
  - 定义1和定义2是同义的(synonymous)

# 等价状态的例子




(a)

| | $x$ = 0 | $x$ = 1 |
|---|---|---|
| *A* | C/1 | B/0 |
| *B* | C/1 | E/0 |
| *C* | B/1 | E/0 |
| *D* | D/0 | B/1 |
| *E* | E/0 | A/1 |

(b)

| Initial State | Input Sequences 00 | 01 | 10 | 11 |
|---|---|---|---|---|
| *A* | 11 | 10 | 01 | 00 |
| *B* | 11 | 10 | 00 | 01 |
| *C* | 11 | 10 | 00 | 01 |
| *D* | 00 | 01 | 11 | 10 |
| *E* | 00 | 01 | 11 | 10 |

(c)

| Initial State | Input Sequences 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| *A* | 111 | 110 | 100 | 101 | 011 | 010 | 000 | 001 |
| *B* | 111 | 110 | 100 | 101 | 000 | 001 | 011 | 010 |
| *C* | 111 | 110 | 100 | 101 | 000 | 001 | 011 | 010 |
| *D* | 000 | 001 | 011 | 010 | 111 | 110 | 100 | 101 |
| *E* | 000 | 001 | 011 | 010 | 111 | 110 | 101 | 100 |

(d)

# 1.2 等价和等价关系

- **等价关系：**设R是集合S上定义的关系(Relation)，R是等价的当且仅当R是自反的(reflexive)、对称的(symmetric)、传递的(transitive)。一个集合的等价关系可以将集合划分(partition)为不相交的(disjoint)等价类。
- 例：let $S = \{A,B,C,D,E,F,G,H\}$ and $R = \{(A,A),(B,B),(B,H),(C,C),(D,D),(D,E),(E,E),(E,D),(F,F),(G,G),(H,H),(H,B)\}$. Then $P = (A)(BH)(C)(DE)(F)(G)$
- **定理：时序电路的状态等价是状态集合的一个等价关系。**
- **定理：时序电路状态等价定义的等价类可以用来表示等价电路中的状态。**

# 2 完全确定电路的状态化简

- 三种方法
  - 1 观察法
  - 2 划分法
  - 3 蕴含表法

# 2.1 观察法

| | $x=0$ | $x=1$ |
|---|---|---|
| $A$ | $B/0$ | $C/1$ |
| $B$ | $C/0$ | $A/1$ |
| $C$ | $D/1$ | $B/0$ |
| $D$ | $C/0$ | $A/1$ |

(a)

| | $x=0$ | $x=1$ |
|---|---|---|
| $A$ | $B/0$ | $C/1$ |
| $B$ | $C/0$ | $A/1$ |
| $C$ | $B/1$ | $B/0$ |

(b)

| | $x=0$ | $x=1$ |
|---|---|---|
| $A$ | $B/0$ | $C/1$ |
| $B$ | $B/0$ | $A/1$ |
| $C$ | $D/1$ | $B/0$ |
| $D$ | $D/0$ | $A/1$ |

(c)

| | $x=0$ | $x=1$ |
|---|---|---|
| $A$ | $B/0$ | $C/1$ |
| $B$ | $B/0$ | $A/1$ |
| $C$ | $B/1$ | $B/0$ |

(d)

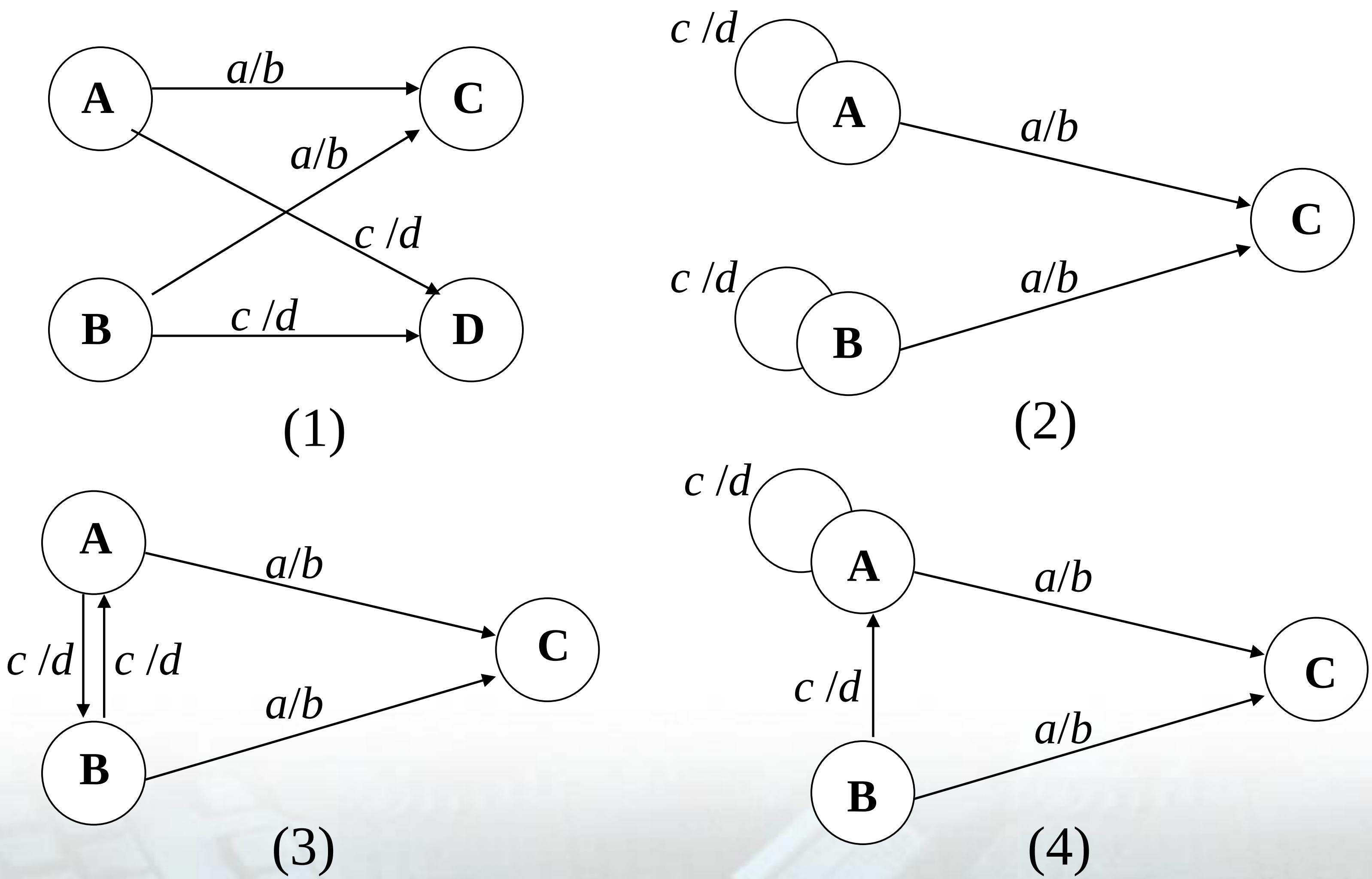
| | $x=0$ | $x=1$ |
|---|---|---|
| $A$ | $B/0$ | $C/1$ |
| $B$ | $D/0$ | $A/1$ |
| $C$ | $D/1$ | $B/0$ |
| $D$ | $B/0$ | $A/1$ |

(e)

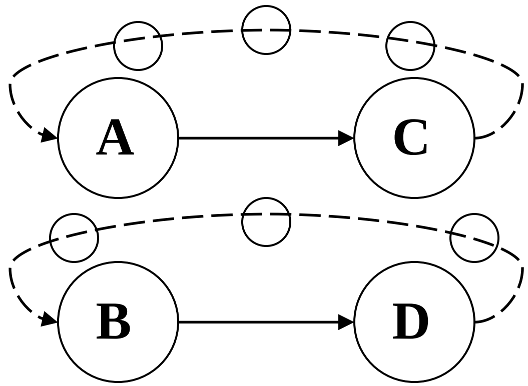
# 观察法的原则

- 两个状态等价
  - 在相同输入和输出的前提下
    - 下一个状态是相同的
    - 下一个状态是这两个状态之一(self-loop-back)

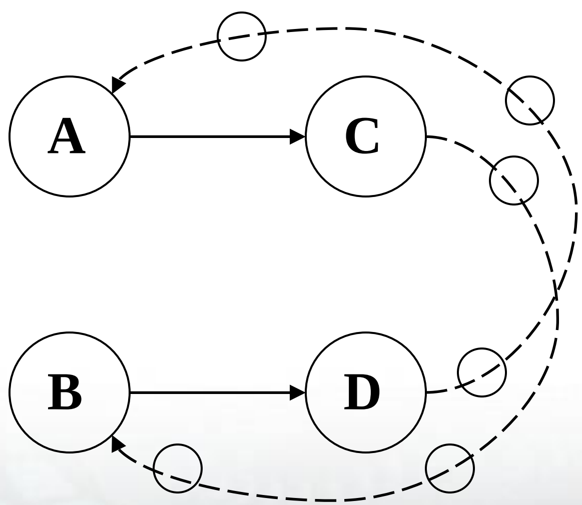
# 观察法的4种模式（A和B等价）



# 另两种模式（A和B等价）




(5)



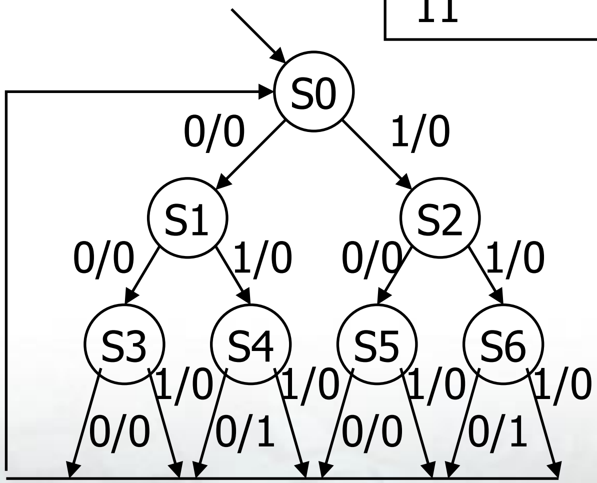

(6)

- 对于这两种模式，采用观察法很难发现

# 状态化简例子

- 序列识别010 或者110 （不重叠）

| Input Sequence | Present State | Next State X=0 | Next State X=1 | Output X=0 | Output X=1 |
|---|---|---|---|---|---|
| Reset | S0 | S1 | S2 | 0 | 0 |
| 0 | S1 | S3 | S4 | 0 | 0 |
| 1 | S2 | S5 | S6 | 0 | 0 |
| 00 | S3 | S0 | S0 | 0 | 0 |
| 01 | S4 | S0 | S0 | 1 | 0 |
| 10 | S5 | S0 | S0 | 0 | 0 |
| 11 | S6 | S0 | S0 | 1 | 0 |



# 状态化简例子

| Input Sequence | Present State | Next State X=0 | Next State X=1 | Output X=0 | Output X=1 |
|---|---|---|---|---|---|
| Reset | S0 | S1 | S2 | 0 | 0 |
| 0 | S1 | S3 | S4 | 0 | 0 |
| 1 | S2 | S5 | S6 | 0 | 0 |
| 00 | S3 | S0 | S0 | 0 | 0 |
| 01 | S4 | S0 | S0 | 1 | 0 |
| 10 | S5 | S0 | S0 | 0 | 0 |
| 11 | S6 | S0 | S0 | 1 | 0 |

( S0 S1 S2 S3 S4 S5 S6 )

( S0 S1 S2 S3 S5 ) ( S4 S6 )

( S0 S3 S5 ) ( S1 S2 ) ( S4 S6 )

( S0 ) ( S3 S5 ) ( S1 S2 ) ( S4 S6 )

S3与S5等价

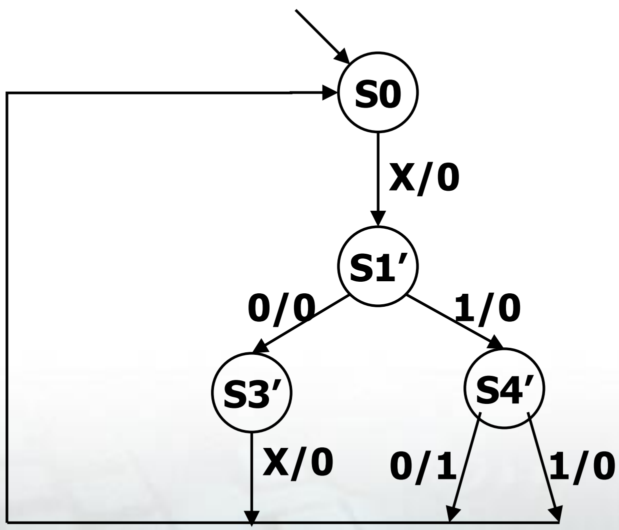
S4与S6等价

S1与S2等价

# 最简化FSM

- 状态最少的序列识别010或110

| Input Sequence | Present State | Next State X=0 | Next State X=1 | Output X=0 | Output X=1 |
|---|---|---|---|---|---|
| Reset | S0 | S1' | S1' | 0 | 0 |
| 0 + 1 | S1' | S3' | S4' | 0 | 0 |
| X0 | S3' | S0 | S0 | 0 | 0 |
| X1 | S4' | S0 | S0 | 1 | 0 |



# 例子中的几个概念

- 1-等价 1-equivalent
- 2-等价 2-equivalent
- …
- K-等价 K-equivalent
  - 两个状态Si和Sj 输入*K*个输入序列，输出序列是相同的。
- ∞ 等价 = 状态等价

# 2.3 划分法(Partitioning)

- 划分法是一组连续的过程。
- 每步形成的划分$P_k$，由一些块(block)组成，每个块中的状态是K-等价的。
- 划分步骤
  - 第一步：将输出相同的状态分在一个块内，形成1-等价划分。
  - 第二步：用下面的方法连续得到$P_k$, k=2,3,4,5...; 直到$P_{k-1}=P_k$。
    - 若对于每个输入,Si与Sj的次态都在同一个$P_{k-1}$的块内,则Si与Sj放在$P_k$的同一块内

# 例1

| | Partition blocks | Action |
|---|---|---|
| Partition $P_0$ | $(ABCDE)$ | |
| Output for $x=0$ | 11100 | Separate $(ABC)$ and $(DE)$ |
| Output for $x=1$ | 00011 | Separate $(ABC)$ and $(DE)$ |
| Partition $P_1$ | $(ABC)$ $(DE)$ | |
| Next state for $x=0$ | $CCB$ $DE$ | |
| Next state for $x=1$ | $BEE$ $BA$ | Separate $(A)$ and $(BC)$ |
| Partition $P_2$ | $(A)$ $(BC)$ $(DE)$ | |
| Next state for $x=0$ | $C$ $CB$ $DE$ | |
| Next state for $x=1$ | $B$ $EE$ $BA$ | Separate $(D)$ and $(E)$ |
| Partition $P_3$ | $(A)$ $(BC)$ $(D)$ $(E)$ | |
| Next state for $x=0$ | $C$ $CB$ $D$ $E$ | |
| Next state for $x=1$ | $B$ $EE$ $B$ $A$ | |
| Partition $P_4=P_3$ | $(A)$ $(BC)$ $(D)$ $(E)$ | |

States B and C are equivalent

# 例2

- P1 = (AD)(BE)(CF)(GH)

  P2 = (AD)(BE)(CF)(G)(H)

  P3 = P2

| | $x$ = 0 | $x$ = 1 |
|---|---|---|
| $A$ | E/0 | D/0 |
| $B$ | A/1 | F/0 |
| $C$ | C/0 | A/1 |
| $D$ | B/0 | A/0 |
| $E$ | D/1 | C/0 |
| $F$ | C/0 | D/1 |
| $G$ | H/1 | G/1 |
| $H$ | C/1 | B/1 |

(a)

| | $x$ = 0 | $x$ = 1 |
|---|---|---|
| $A'$ | B'/0 | A'/0 |
| $B'$ | A'/1 | C'/0 |
| $C'$ | C'/0 | A'/1 |
| $D'$ | E'/1 | D'/1 |
| $E'$ | C'/1 | B'/1 |

(b)

# 例3

- P1 = (ACG)(BDEH)(F)

  P2 = (A)(CG)(BH)(DE)(F)

  = (A)(C)(G)(BH)(DE)(F)

  P3 = P2

| | $x$ = 0 | 1 |
|---|---|---|
| $A$ | A/0 | B/0 |
| $B$ | H/1 | C/0 |
| $C$ | E/0 | B/0 |
| $D$ | C/1 | D/0 |
| $E$ | C/1 | E/0 |
| $F$ | F/1 | G/1 |
| $G$ | B/0 | F/0 |
| $H$ | H/1 | C/0 |

(a)

| | $x$ = 0 | 1 |
|---|---|---|
| $A'$ | A'/0 | E/0 |
| $B'$ | B'/1 | D'/1 |
| $C'$ | F/0 | E/0 |
| $D'$ | E/0 | B'/0 |
| $E'$ | E/1 | C/0 |
| $F'$ | C/1 | F/0 |

(b)

# 例4

- P1 = (ADFG)(BCEH)

  P2 = (AFG)(D)(BCEH)

  P3 = (AF)(G)(D)(BCH)(E)

  P4 = P3

| | $x_1x_2$ 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| $A$ | D/0 | D/0 | F/0 | A/0 |
| $B$ | C/1 | D/0 | E/1 | F/0 |
| $C$ | C/1 | D/0 | E/1 | A/0 |
| $D$ | D/0 | B/0 | A/0 | F/0 |
| $E$ | C/1 | F/0 | E/1 | A/0 |
| $F$ | D/0 | D/0 | A/0 | F/0 |
| $G$ | G/0 | G/0 | A/0 | A/0 |
| $H$ | B/1 | D/0 | E/1 | A/0 |

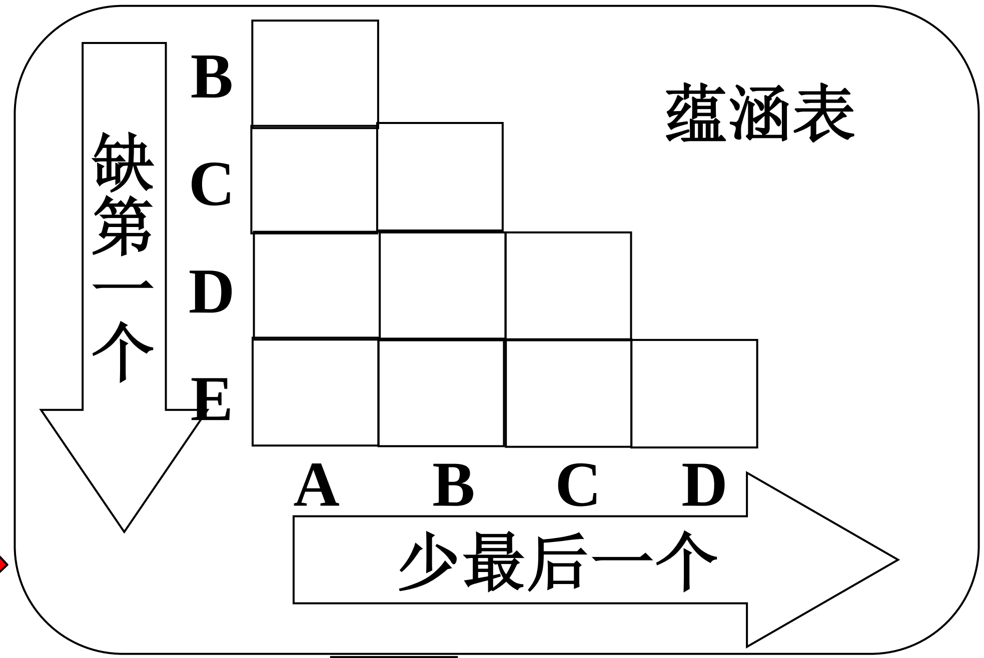
(a)

| | $x_1x_2$ 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| $A'$ | C'/0 | C'/0 | A'/0 | A'/0 |
| $B'$ | B'/1 | C'/0 | D'/1 | A'/0 |
| $C'$ | C'/0 | B'/0 | A'/0 | A'/0 |
| $D'$ | B'/1 | A'/0 | D'/1 | A'/0 |
| $E'$ | E'/0 | E'/0 | A'/0 | A'/0 |

(b)

# 2.4 蕴涵表法

| | X = 0 | X = 1 |
|---|---|---|
| A | C/1 | B/0 |
| B | C/1 | E/0 |
| C | B/1 | E/0 |
| D | D/0 | B/1 |
| E | E/0 | A/1 |




| | A | B | C | D |
|---|---|---|---|---|
| B | BE | | | |
| C | BC, BE | √ | | |
| D | × | × | × | |
| E | × | × | × | BA |

| | A | B | C | D |
|---|---|---|---|---|
| B | ~~BE~~ | | | |
| C | ~~BC, BE~~ | √ | | |
| D | × | × | × | |
| E | × | × | × | ~~BA~~ |

# 蕴含表法

(1)画出空的蕴涵表，蕴含表的方格表示所有可能的状态对。

(2)检查蕴涵表的每个方块，当对应的两状态输出不相同时，在该方块内打“×”。

(3)添完蕴涵表:对于每一种输入可能，将输出相同的次态对添入对应的蕴涵单元内。当蕴涵表单元的蕴涵项等于相对应的两状态或为同一个状态时：打对号“√”；当单元包含的所有蕴涵对都变为“√”时，其对应两状态位等价，对应的单元可变为“√”。

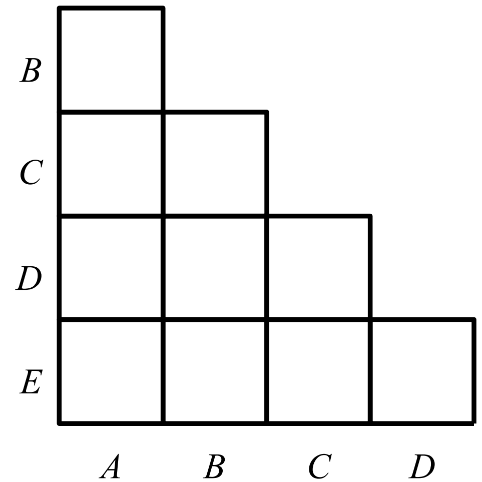
(4)处理蕴涵表，确定每个状态对的等价性。当蕴涵单元包含的蕴涵项有一个为“×”时,则其对应的状态对不等价，该单元画“×”.

(5)从蕴涵表得到等价状态对，导出等价划分：没有画“×”的单元对应的状态对为等价对.

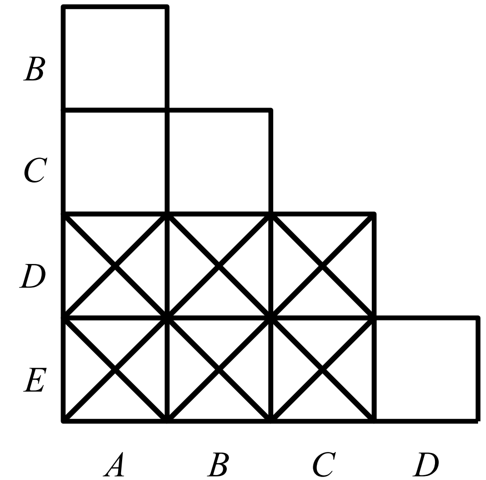
# 例1

| | $x=0$ | $x=1$ |
|---|---|---|
| $A$ | $C/1$ | $B/0$ |
| $B$ | $C/1$ | $E/0$ |
| $C$ | $B/1$ | $E/0$ |
| $D$ | $D/0$ | $B/1$ |
| $E$ | $E/0$ | $A/1$ |

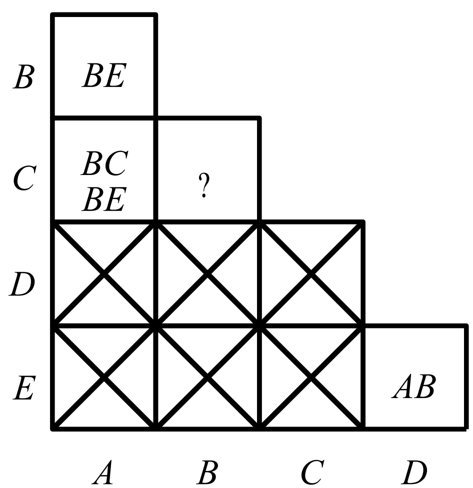
(a)



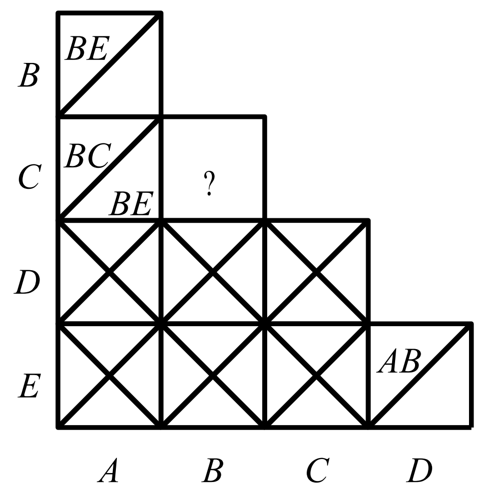

(b)




(c)




(d)




(e)

| | |
|---|---|
| $A$ | - |
| $B$ | $(BC)$ |
| $C$ | - |
| $D$ | - |

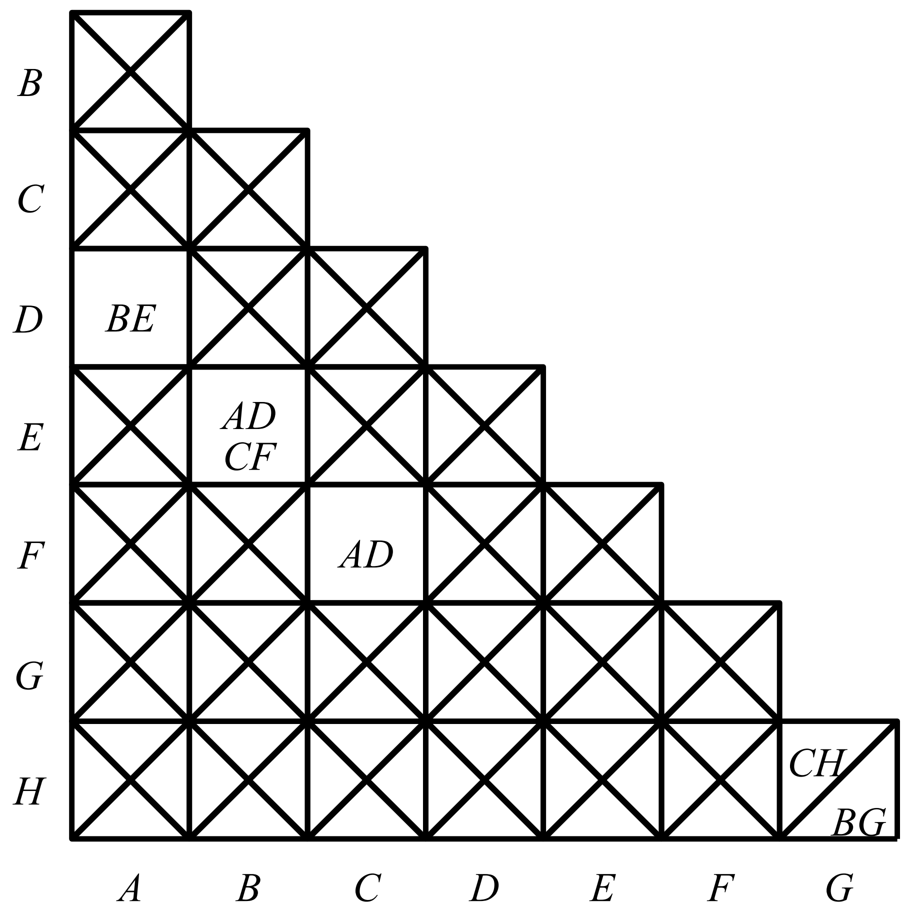
$$P_K = (A)(BC)(D)(E)$$

(f)

# 例2

| | $x=0$ | $x=1$ |
|---|---|---|
| $A$ | $E/0$ | $D/0$ |
| $B$ | $A/1$ | $F/0$ |
| $C$ | $C/0$ | $A/1$ |
| $D$ | $B/0$ | $A/0$ |
| $E$ | $D/1$ | $C/0$ |
| $F$ | $C/0$ | $D/1$ |
| $G$ | $H/1$ | $G/1$ |
| $H$ | $C/1$ | $B/1$ |

(a)




(b)

| | |
|---|---|
| $A$ | $(AD)$ |
| $B$ | $(BE)$ |
| $C$ | $(CF)$ |
| $D$ | - |
| $E$ | - |
| $F$ | - |
| $G$ | - |

$$P_K = (AD)(BE)(CF)(G)(H)$$

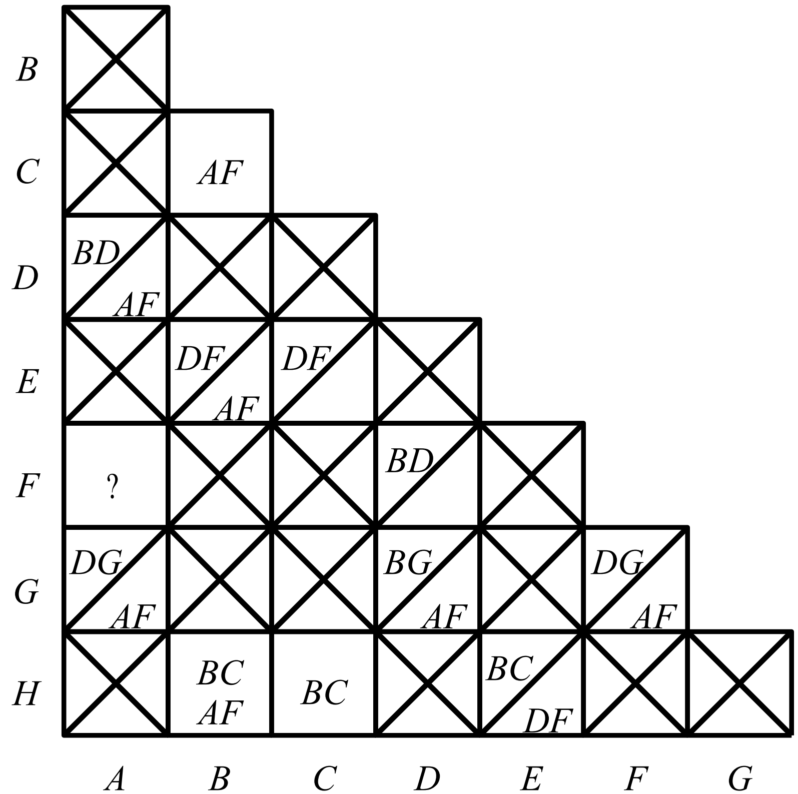
(c)

# 例3

$x_1x_2$

| | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| *A* | *D*/0 | *D*/0 | *F*/0 | *A*/0 |
| *B* | *C*/1 | *D*/0 | *E*/1 | *F*/0 |
| *C* | *C*/1 | *D*/0 | *E*/1 | *A*/0 |
| *D* | *D*/0 | *B*/0 | *A*/0 | *F*/0 |
| *E* | *C*/1 | *F*/0 | *E*/1 | *A*/0 |
| *F* | *D*/0 | *D*/0 | *A*/0 | *F*/0 |
| *G* | *G*/0 | *G*/0 | *A*/0 | *A*/0 |
| *H* | *B*/1 | *D*/0 | *E*/1 | *A*/0 |

(a)




(b)

| | |
|---|---|
| *A* | (*AF*) |
| *B* | (*BC*)(*BH*) |
| *C* | (*CH*) |
| *D* | - |
| *E* | - |
| *F* | - |
| *G* | - |

Note: (*BC*)(*BH*)(*CH*) = (*BCH*)

$P_K$ = (*AF*)(*BCH*)(*D*)(*E*)(*G*)

(c)

# 三种方法的比较

- 观察法：直观，功能不强。
- 划分法：功能强，可编程性强。
- 蕴含表法：功能强，可编程性强，步骤繁琐，所用时间多。

# 3 非完全确定状态机的化简

| | X = 0 | X = 1 |
|---|---|---|
| A | B/- | E/0 |
| B | B/1 | E/- |
| C | F/0 | C/0 |
| D | B/1 | A/1 |
| E | D/0 | C/- |
| F | D/- | C/1 |

| | X = 0 | X = 1 |
|---|---|---|
| A | B/1 | E/0 |
| B | B/1 | E/0 |
| C | F/0 | C/0 |
| D | B/1 | A/1 |
| E | D/0 | C/1 |
| F | D/0 | C/1 |

(AB)(C)(D)(EF)

| | X = 0 | X = 1 |
|---|---|---|
| A’ | B’/1 | D’/0 |
| B’ | D’/0 | B’/0 |
| C’ | A’/1 | A’/1 |
| D’ | C’/0 | B’/1 |

# 另一个结果

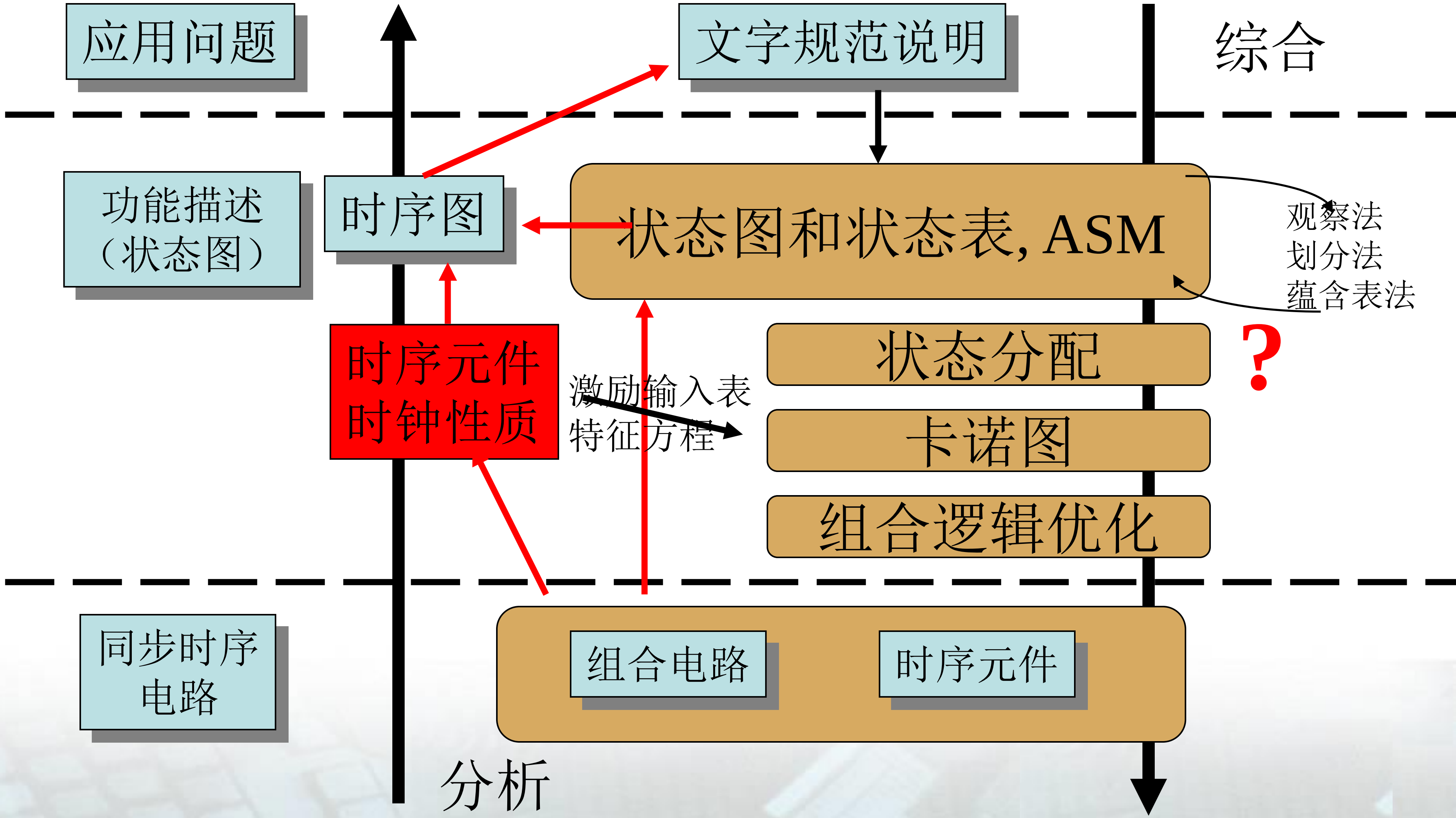
| | X=0 | X=1 |
|---|---|---|
| A | B/- | E/0 |
| B | B/1 | E/- |
| C | F/0 | C/0 |
| D | B/1 | A/1 |
| E | D/0 | C/- |
| F | D/- | C/1 |

| | X=0 | X=1 |
|---|---|---|
| A | B/0 | E/0 |
| B | B/1 | E/1 |
| C | F/0 | C/0 |
| D | B/1 | A/1 |
| E | D/0 | C/0 |
| F | D/1 | C/1 |

(ACE)(BDF)

| | X=0 | X=1 |
|---|---|---|
| A’ | B’/0 | A’/0 |
| B’ | B’/1 | A’/1 |

# 同步时序电路的分析与设计



# 4 优化的状态分配方法

- 例1

| Present state | $x$ = 0 | $x$ = 1 |
|---|---|---|
| $A$ | $B$/0 | $E$/0 |
| $B$ | $C$/0 | $G$/0 |
| $C$ | $D$/0 | $F$/0 |
| $D$ | $A$/1 | $A$/0 |
| $E$ | $G$/0 | $C$/0 |
| $F$ | $A$/0 | $A$/1 |
| $G$ | $F$/0 | $D$/0 |

Next state/output

# 例1的两种状态分配

分配1

| | y1 | y2 | y3 |
|---|---|---|---|
| A: | 0 | 0 | 0 |
| B: | 0 | 0 | 1 |
| C: | 0 | 1 | 1 |
| D: | 0 | 1 | 0 |
| E: | 1 | 0 | 1 |
| F: | 1 | 1 | 0 |
| G: | 1 | 1 | 1 |

分配2

| | y1 | y2 | y3 |
|---|---|---|---|
| A: | 0 | 0 | 0 |
| B: | 0 | 0 | 1 |
| C: | 0 | 1 | 0 |
| D: | 0 | 1 | 1 |
| E: | 1 | 0 | 0 |
| F: | 1 | 0 | 1 |
| G: | 1 | 1 | 0 |

$$J_1 = \bar{y}_2 x + y_3 x, K_1 = \bar{y}_3 + x$$

$$J_2 = y_3, K_2 = \bar{y}_3$$

$$J_3 = \bar{y}_2, K_3 = y_2$$

$$z = \bar{y}_3 y_2 \bar{y}_1 x + \bar{y}_3 y_1 x$$

3个OR；4个AND

$$J_1 = x\bar{y}_3 + x\bar{y}_2, K_1 = x + y_3$$

$$J_2 = y_1\bar{y}_3 + \bar{y}_1 y_3, K_2 = y_3 + \bar{x}y_1 + x\bar{y}_1$$

$$J_3 = y_2 + \overline{xy}, K_3 = 1$$

$$z = xy_3 + xy_2y_3$$

6个OR；9个AND

# 4.1 状态分配

- 可选择的状态分配方案数目十分巨大。

$$2^{N_{ff}-1} < N_s \le 2^{N_{ff}}$$

- 所有可能的状态分配方案数：

$$N_{SA} = \frac{2^{N_{ff}}!}{(2^{N_{ff}} - N_S)!}$$

# 例2

- 方案2将方案1的y1置取补
- 方案3将方案1的y1和y2交换

| Present state | $x$ = 0 | $x$ = 1 |
|---|---|---|
| $A$ | $A$/0 | $B$/0 |
| $B$ | $A$/0 | $C$/0 |
| $C$ | $C$/0 | $D$/0 |
| $D$ | $C$/1 | $A$/0 |

(a)

| States | Assignments 1 $y_1 y_2$ | Assignments 2 $y_1 y_2$ | Assignments 3 $y_1 y_2$ |
|---|---|---|---|
| $A$ | 00 | 10 | 00 |
| $B$ | 01 | 11 | 10 |
| $C$ | 11 | 01 | 11 |
| $D$ | 10 | 00 | 01 |

(b)

# 例2：结论

- 通过简单的将状态变量取补或者交换状态变量，不能化简电路。
- 可选择的状态分配方案：

$$N_{UA} = \frac{(2^{N_{ff}} - 1)!}{(2^{N_{ff}} - N_S)!\,N_{ff}!}$$

# 状态分配策略

- 可能的策略
  - 顺序分配 – 状态表中出现的次序
  - 随机分配 – 随机的选择编码
  - one-hot – 表示状态的位数和状态个数相同
  - 输出 – 利用输出帮助状态编码
  - 启发式 – 利用一系列规则进行编码
- 没有最优解

# 4.2 状态分配指导原则

- 问题：如何指导选择状态分配方案来降低次态生成逻辑的复杂性。
- 总体思路：选择一种状态编码方式，使在状态转换表（卡诺图）中的“1” 组成尽可能大的组（质蕴含项）。
- D触发器的使用。

# 例3：4状态状态机的例子

| | $x$ = 0 | 1 |
|---|---|---|
| $A$ | C/0 | D/0 |
| $B$ | C/0 | A/0 |
| $C$ | B/0 | D/0 |
| $D$ | A/1 | B/1 |

# 分配方案1

$A \rightarrow 00$

$B \rightarrow 01$

$C \rightarrow 11$

$D \rightarrow 10$

| $y_2y_1$ \ $x$ | 0 | 1 |
|---|---|---|
| 00 | **11/0** | **10/0** |
| 01 | **11/0** | **00/0** |
| 11 | **01/0** | **10/0** |
| 10 | **00/1** | **01/1** |

$Y_2\,Y_1/z$

(a)

$$D_2 = \bar{y}_1\bar{y}_2 + \bar{x}\bar{y}_2 + xy_1y_2$$

$$D_1 = \bar{x}\bar{y}_2 + \bar{x}y_1 + x\bar{y}_1y_2$$

$$z = \bar{y}_1y_2$$

| $y_2y_1$ \ $x$ | 0 | 1 |
|---|---|---|
| 00 | **0** | **0** |
| 01 | **0** | **0** |
| 11 | **0** | **0** |
| 10 | **1** | **1** |

y1 $y_2$

$z$

(b)

| $y_2y_1$ \ $x$ | 0 | 1 |
|---|---|---|
| 00 | **1** | **1** |
| 01 | **1** | **0** |
| 11 | **0** | **1** |
| 10 | **0** | **0** |

$y_1$ $y_2$

$D_2$

(c)

| $y_2y_1$ \ $x$ | 0 | 1 |
|---|---|---|
| 00 | **1** | **0** |
| 01 | **1** | **0** |
| 11 | **1** | **0** |
| 10 | **0** | **1** |

$y_1$ $y_2$

$D_1$

(d)

# 分配方案2

$A \rightarrow 00$

$B \rightarrow 11$

$C \rightarrow 01$

$D \rightarrow 10$

| $y_2 y_1$ \ $x$ | 0 | 1 |
|---|---|---|
| 00 | **01/0** | **10/0** |
| 01 | **11/0** | **10/0** |
| 11 | **01/0** | **00/0** |
| 10 | **00/1** | **11/1** |

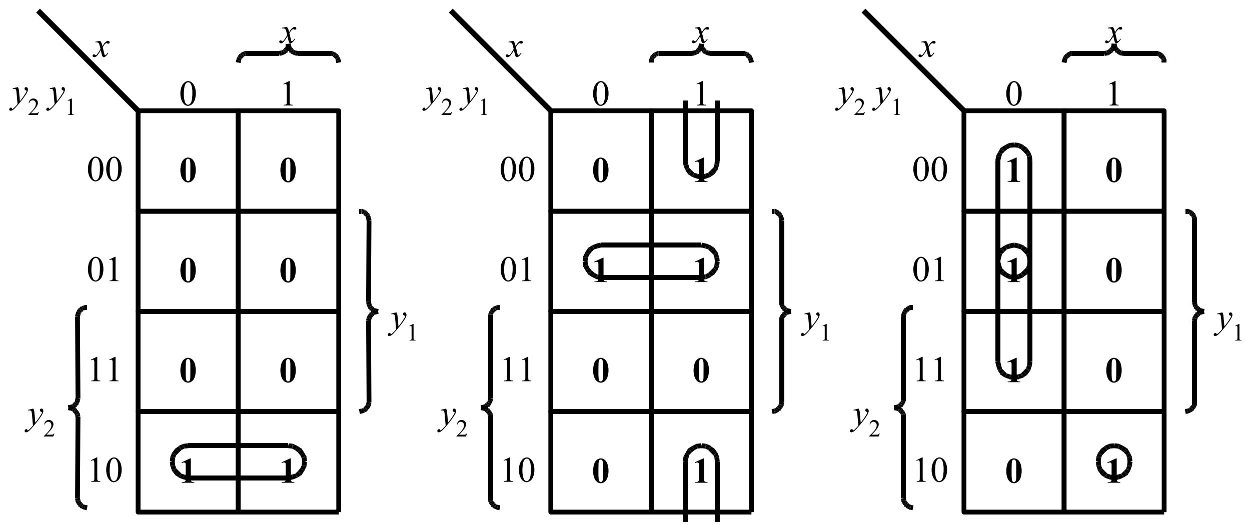
$Y_2\ Y_1/z$

(a)

$$D_2 = x\bar{y}_1 + y_1\bar{y}_2$$

$$D_1 = \bar{x}\bar{y}_2 + \bar{x}y_1 + x\bar{y}_1 y_2$$

$$z = \bar{y}_1 y_2$$




$z$ (b) $D_2$ (c) $D_1$ (d)

# 分配方案3

$A \rightarrow 00$

$B \rightarrow 10$

$C \rightarrow 01$

$D \rightarrow 11$

| $y_2y_1$ \ $x$ | 0 | 1 |
|---|---|---|
| 00 | **01/0** | **11/0** |
| 01 | **10/0** | **11/0** |
| 11 | **00/1** | **10/1** |
| 10 | **01/0** | **00/0** |

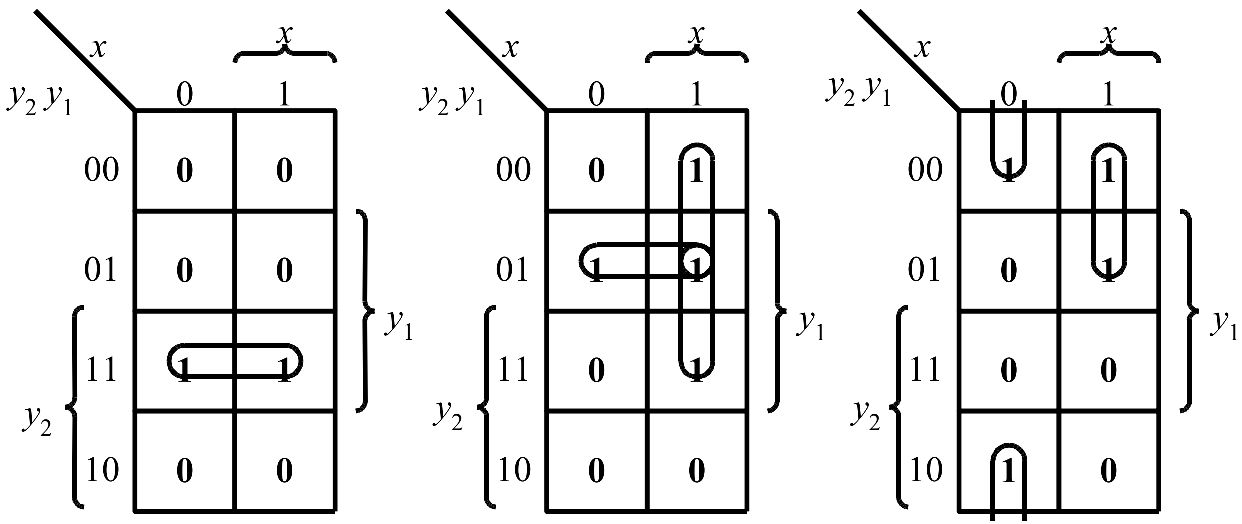
$Y_2\ Y_1/z$

(a)

$$D_2 = y_1\bar{y}_2 + x\bar{y}_2 + xy_1$$

$$D_1 = x\bar{y}_2 + \bar{x}\bar{y}_1$$

$$z = y_1y_2$$




$z$ (b)

$D_2$ (c)

$D_1$ (d)

# 状态分配规则

- 规则1：对于给定的输入，具有相同次态的状态应该分配逻辑相邻的编码。
- 规则2：对于一个现态，在逻辑相邻输入下的次态，应该分配逻辑相邻的编码

- 例3：
  - 规则1：A adj B; A adj C
  - 规则2：A adj B; A adj C; B adj D; C adj D

# 状态分配卡诺图

| $y_1 \backslash y_2$ | 0 | 1 |
|---|---|---|
| 0 | $A$ | $D$ |
| 1 | $B$ | $C$ |

(a)

分配 1

| $y_1 \backslash y_2$ | 0 | 1 |
|---|---|---|
| 0 | $A$ | $D$ |
| 1 | $C$ | $B$ |

(b)

分配 2

| $y_1 \backslash y_2$ | 0 | 1 |
|---|---|---|
| 0 | $A$ | $B$ |
| 1 | $C$ | $D$ |

(c)

分配 3

# 状态分配规则

- 规则3：输出规则，在相同输入情况下，具有相同输出的状态应该分配相邻的编码。
- 规则4：已经利用规则1、2、3确定分配相邻编码的状态对，其次态对应该分配相邻的编码。
- 规则5：找到在化简的状态表中作为次态次数最多的状态，分配“全0”的编码，然后利用规则1、2、3、4进行状态分配。尽可能的满足所有的规则。

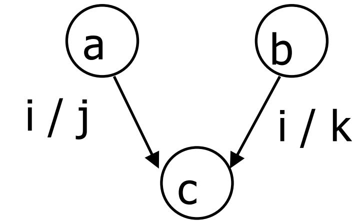
# 启发式状态分配

- 规则1

  –

| I | Q | $Q^+$ | O |
|---|---|---|---|
| i | a | c | j |
| i | b | c | k |

$c = i * a + i * b$

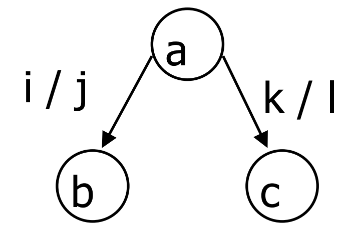



- 规则2

  –

| I | Q | $Q^+$ | O |
|---|---|---|---|
| i | a | b | j |
| k | a | c | l |

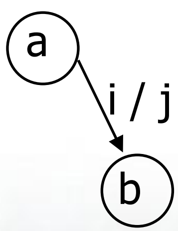
$b = i * a$
$c = k * a$

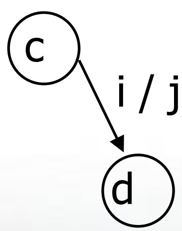



- 输出规则

| I | Q | $Q^+$ | O |
|---|---|---|---|
| i | a | b | j |
| i | c | d | j |

$j = i * a + i * c$
$b = i * a$
$d = i * c$



# 基于输出的状态编码

北京大学

- 利用输出作为状态的编码 – 用输出帮助区分状态
  - 复用计算输出的函数
  - 适合同步Mealy机的设计

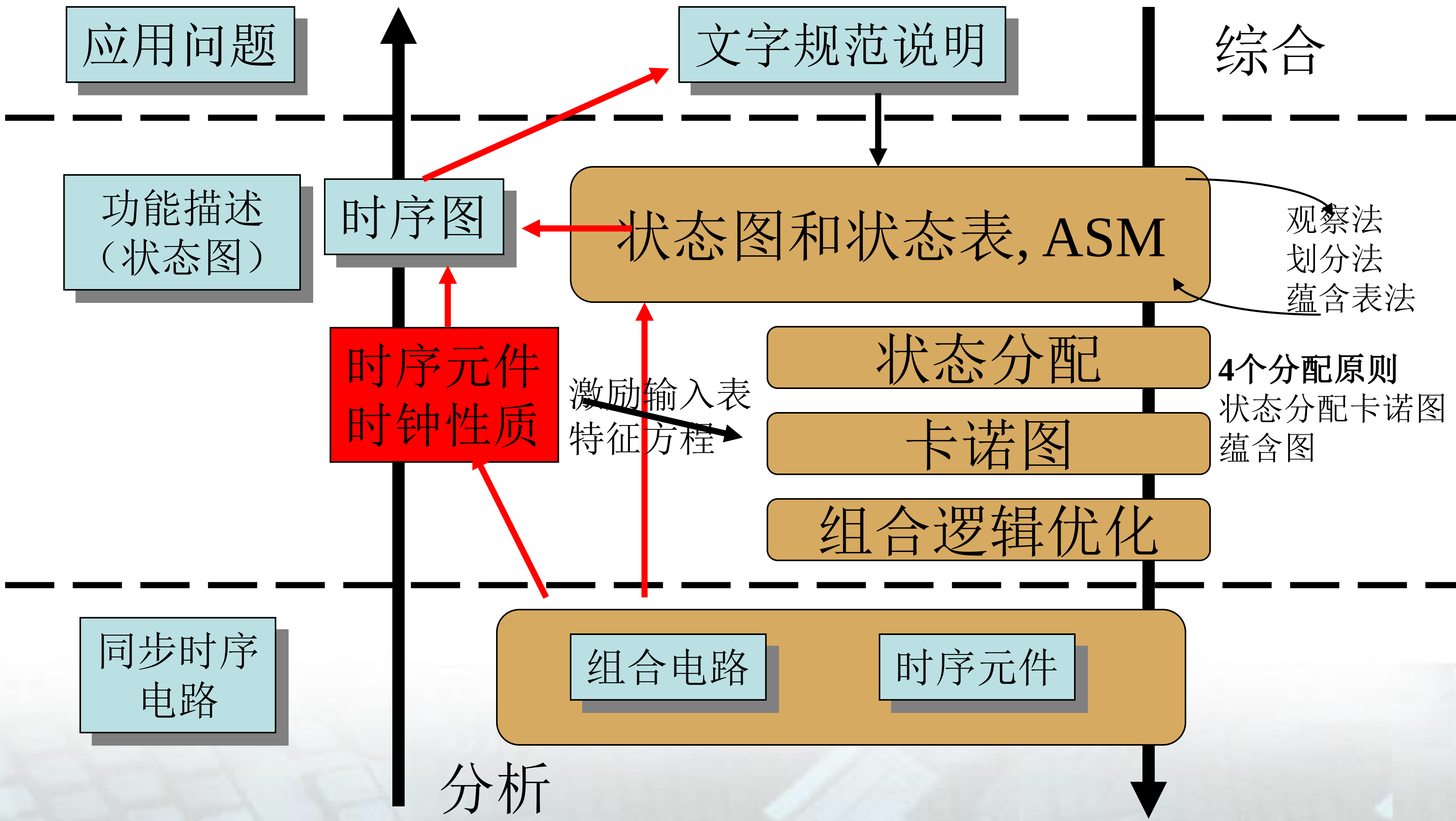
| Inputs | | | Present State | Next State | Outputs | | |
|---|---|---|---|---|---|---|---|
| C | TL | TS | | | ST | H | F |
| 0 | – | – | HG | HG | 0 | 00 | 10 |
| – | 0 | – | HG | HG | 0 | 00 | 10 |
| 1 | 1 | – | HG | HY | 1 | 00 | 10 |
| – | – | 0 | HY | HY | 0 | 01 | 10 |
| – | – | 1 | HY | FG | 1 | 01 | 10 |
| 1 | 0 | – | FG | FG | 0 | 10 | 00 |
| 0 | – | – | FG | FY | 1 | 10 | 00 |
| – | 1 | – | FG | FY | 1 | 10 | 00 |
| – | – | 0 | FY | FY | 0 | 10 | 01 |
| – | – | 1 | FY | HG | 1 | 10 | 01 |

HG = ST′ H1′ H0′ F1 F0′ + ST H1 H0′ F1′ F0
HY = ST H1′ H0′ F1 F0′ + ST′ H1′ H0 F1 F0′
FG = ST H1′ H0 F1 F0′ + ST′ H1 H0′ F1′ F0′
HY = ST H1 H0′ F1′ F0′ + ST′ H1 H0′ F1′ F0

输出与状态一一对应，不需要任何计算状态的函数

只需要5个函数（一个输出一个）而不是7个函数

# 总结


应用问题
文字规范说明
综合
功能描述
（状态图）
时序图
状态图和状态表, ASM
观察法
划分法
蕴含表法
时序元件
时钟性质
激励输入表
特征方程
状态分配
卡诺图
组合逻辑优化
4个分配原则
状态分配卡诺图
蕴含图
同步时序
电路
组合电路
时序元件
分析