STĂRI

S.l. Ing. Vlad-Cristian Miclea

Universitatea Tehnica din Cluj-Napoca
Departamentul Calculatoare

CUPRINS

PLAN CURS

- Partea 1 – FPGA si VHDL
  1. FPGA
  2. Limbajul VHDL – 1
  3. Limbajul VHDL – 2
  4. Limbajul VHDL – 3

- Partea 2 – Implementarea sistemelor numerice
  5. Microprogramare
  6. Partea 1 - Unitate de comanda + executie – exemplu impartitor
  6. Partea 2 – Cod UC + UE impartitor

- Partea 3 – Automate
  7. Automate finite
  8. **Stari**
  9. Automate sincrone
  10. Automate asincrone
  11. Identificarea automatelor; Automate fara pierderi
  12. Automate liniare + probleme si discutii

CONTEXT

Cursurile trecute

- Proiectarea unui sistem numeric complex
  - Exemplu – cuptor simplu
- Automate finite
  - Definitii, informatii generale
  - Reprezentari
  - Clasificarea automatelor (Moore, Mealy)
- Saptamana aceasta – stari ale automatelor

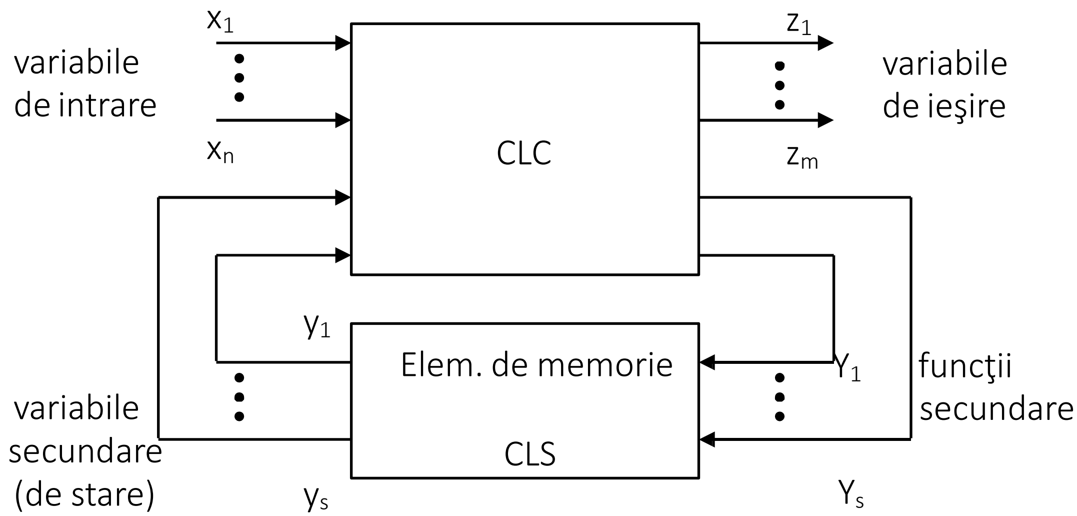
STĂRI ŞI VARIABILE DE STARE

- **Definiţie: Starea** unui automat este cantitatea de informaţie necesară determinării evoluţiei automatului începând cu acel moment
- **Identificarea stării** unui automat se realizează cu ajutorul unui număr de variabile numite ***variabile de stare*** sau ***variabile secundare***

STĂRI ŞI VARIABILE DE STARE

Automat finit

- Reprezentarea:



- $X = (x_1, x_2, ..., x_n)^T$ vectorul variabilelor de intrare
- $Z = (z_1, z_2, ..., z_m)^T$ vectorul variabilelor de ieşire
- $y = (y_1, y_2, ..., y_s)^T$ vectorul variabilelor de stare
  starea actuală a automatului
- $Y = (Y_1, Y_2, ..., Ys)^T$ vectorul funcţiilor secundare (de excitaţie)
  starea următoare a automatului

STĂRI ŞI VARIABILE DE STARE

Evoluţia automatului

- Sistem de ecuaţii - ţin cont de variabilele de intrare şi de stare pentru determinarea stării următoare Y şi a ieşirii Z

  - $Y_i(t) = f_i(x_1(t),\ldots, x_n(t), y_1(t), \ldots, y_s(t))$ $\quad i = \overline{1,s}$
  - $Z_j(t) = g_j(x_1(t),\ldots, x_n(t), y_1(t), \ldots, y_s(t))$ $\quad j=\overline{1,m}$
  - Relaţiile dintre funcţiile secundare Y şi variabilele de stare y determină configuraţia buclei de reacţie

STĂRI ŞI VARIABILE DE STARE

Noţiuni

- Comportare deterministă - dacă pentru orice vector al stării interne există o singură tranziţie posibilă
- Stare stabilă - dacă pentru o anumită configuratie a intrărilor starea internă prezentă este identică cu starea următoare
- Stare instabilă - dacă pentru o anumită configuratie a intrărilor starea prezentă diferă de starea următoare
- Cursă - un eveniment în care se modifică mai multe variabile de stare

REDUCEREA NUMĂRULUI DE STĂRI

Automate complet definite

- Procesul de **reducere** a numărului de stări - determinarea acelor stări care realizează aceeaşi funcţie, adică nu se poate face distincţie între ieşirile rezultate în urma aplicării la intrările automatului, aflat în oricare dintre aceste stări, a aceleiaşi secvenţe de intrare
- Stările determinate se numesc stări echivalente şi pot fi înlocuite printr-o singură stare
- Definiţie: 2 stări s_i şi s_j ale aceluiaşi automat complet definit se numesc echivalente ($s_i = s_j$) dacă pentru orice secvenţă de intrare de lungime arbitrară aplicată automatului aflat în starea s_i, respectiv s_j, se obţine aceeaşi secvenţă de ieşire

REDUCEREA NUMĂRULUI DE STĂRI

Automate complet definite

- Două automate sunt echivalente dacă au toate stările echivalente
- Stările echivalente obţinute în urma unui procedeu de reducere formează clase de echivalenţă disjuncte
- Stările dintr-o clasă de echivalenţă pot fi înlocuite printr-o singură stare
- Problema reducerii numărului de stări se reduce la determinarea claselor de echivalenţă
- Obţinem un automat echivalent cu cel iniţial, dar cu un număr minim de stări

REDUCEREA NUMĂRULUI DE STĂRI

Automate complet definite

- Metoda de determinare a claselor de echivalenţă cu **tabelul implicaţiilor** - **Algoritmul Paull-Unger**
- În tabelul implicaţiilor se înscriu, pentru fiecare pereche de stări care ar putea fi echivalente, implicaţiile echivalenţei respective (adică se înscriu condiţiile care ar trebui îndeplinite pentru ca două stări să fie echivalente)

REDUCEREA NUMĂRULUI DE STĂRI

Automate complet definite

- Reguli de completare a tabelului implicaţiilor:
  - Se porneşte din colţul din stânga sus
  - Se bifează celulele în cazul stărilor evident echivalente
  - Celulele perechilor de stări evident neechivalente se barează cu un X
  - În celula unei perechi de stări care ar putea fi echivalente se trec implicaţiile respective (condiţiile de echivalenţă)
  - La a doua trecere se inspectează tabelul pornind din celula din dreapta jos şi se urmăreşte efectul neechivalenţelor sau incompatibilitatea stărilor
  - După o inspecţie completă se face o nouă inspecţie plecând din colţul din stânga sus şi se caută alte incompatibilităţi care ar fi putut să se piardă la celelalte verificări
  - Scrierea stărilor echivalente obţinute se face începând cu celula din dreapta jos

REDUCEREA NUMĂRULUI DE STĂRI

Exemplu

- Se dă automatul complet definit descris prin tabelul de tranziţii:

S \ X	0	1	2
1	2/1	2/0	5/0
2	1/0	4/1	4/1
3	2/1	2/0	5/0
4	3/0	2/1	2/1
5	6/1	4/0	3/0
6	8/0	9/1	6/1
7	6/1	2/0	8/0
8	4/1	4/0	7/0
9	7/0	9/1	7/1

REDUCEREA NUMĂRULUI DE STĂRI

Exemplu

- Tabelul implicaţiilor este:

2	X							
3	V	X						
4	X	1,3	X					
5	2,6⊗ 2,4 3,5	X	2,6⊗ 2,4	X				
6	X	1,8 4,9⊗ 4,6	X	3,8 2,9⊗ 2,6	X			
7	2,6⊗ 5,8	X	2,6⊗ 5,8	X	2,4 3,8	X		
8	2,4 5,7	X	2,4 5,7	X	4,6⊗ 3,7	X	4,6⊗ 2,4	
9	X	1,7 4,9⊗ 4,7	X	3,7 2,9 2,7⊗	X	7,8 6,7⊗	X	X
	1	2	3	4	5	6	7	8

S X	0	1	2
1	2/1	2/0	5/0
2	1/0	4/1	4/1
3	2/1	2/0	5/0
4	3/0	2/1	2/1
5	6/1	4/0	3/0
6	8/0	9/1	6/1
7	6/1	2/0	8/0
8	4/1	4/0	7/0
9	7/0	9/1	7/1

REDUCEREA NUMĂRULUI DE STĂRI

Exemplu

- Se obţin stările echivalente:
  - 5=7; 3=8; 2=4; 1=8; 1=3
  - Se ţine cont de relaţia de tranzitivitate a relaţiei de echivalenţă a stărilor şi rezultă din 3=8; 1=8 şi 1=3 echivalenţa (1,3,8)
  - Stările care nu apar în nici o clasă de echivalenţe, 6 şi 9, vor forma singure câte o clasă de echivalenţe: (6) (9)
- Clasele de echivalenţe disjuncte obţinute vor fi: (1,3,8), (2,4), (5,7), (6), (9)
  - Notăm clasele de echivalenţe: (1,3,8) = a; (2,4) = b; (5,7) = c; (6) = d; (9) = e

REDUCEREA NUMĂRULUI DE STĂRI

Exemplu

- Automatul echivalent cu cel iniţial (are aceeaşi funcţionalitate) are 5 stări:

S \ X	0	1	2
a	b/1	b/0	c/0
b	a/0	b/1	b/1
c	d/1	b/0	a/0
d	a/0	e/1	d/1
e	c/0	e/1	c/1

REDUCEREA NUMĂRULUI DE STĂRI

Automate incomplet definite

- În multe situaţii practice evoluţia automatului nu este descrisă complet
  - Există intrări pentru care nu se defineşte tranziţia
  - Există tranziţii pentru care nu se defineşte ieşirea
- În aceste situaţii 2 stări nu mai pot fi echivalente fiindcă pentru anumite intrări nu cunoaştem stările următoare sau ieşirile următoare → automat incomplet definit
- Definiţie: 2 stări s_i şi s_j ale aceluiaşi automat incomplet definit se numesc stări compatibile dacă şi numai dacă pentru orice secvenţă de intrare de lungime arbitrară aplicată automatului aflat în starea s_i, respectiv s_j, se obţin secvenţe de ieşiri compatibile (în care ieşirile corespondente sunt identice atunci când sunt specificate)

REDUCEREA NUMĂRULUI DE STĂRI

Automate incomplet definite

- **Definiţie**: O mulţime de stări ale unui automat incomplet definit formează o **clasă de compatibilităţi** dacă fiecare pereche de stări din mulţime are stări compatibile
- **Definiţie**: O **compatibilitate maximă** este o clasă de compatibilităţi care, dacă se mai adaugă vreo stare care nu face parte din clasa respectivă, nu mai rămâne clasă de compatibilităţi
- **Observaţie**: o stare care nu este compatibilă cu nici o altă stare formează o compatibilitate maximă
- Metoda de determinare a perechilor de stări compatibile cu **tabelul implicaţiilor** - **Algoritmul Paull-Unger**

REDUCEREA NUMĂRULUI DE STĂRI

Automate incomplet definite

- Pentru a găsi un automat cu un număr minim de stări, dar cu comportare identică cu cel iniţial, trebuie găsită o mulţime de clase de compatibilităţi minimă şi închisă
- Definiţie: O mulţime de **clase de compatibilităţi** se numeşte închisă dacă pentru oricare din clasele mulţimii toţi succesorii stărilor (stările următoare) din clasa respectivă, pentru fiecare din intrări, fac parte dintr-o singură clasă de compatibilităţi

REDUCEREA NUMĂRULUI DE STĂRI

Automate incomplet definite

- **Mulţimea de clase minimă şi închisă** trebuie să îndeplinească următoarele condiţii:
  - Toate stările să fie prezente în aceste clase
  - Compatibilitatea a două stări oarecare ale unei clase nu trebuie să implice compatibilitatea a două stări care nu aparţin aceleaşi clase
  - Numărul de clase de stări compatibile (compatibilităţi) trebuie să fie minim sau cel puţin egal cu cea mai mică putere posibilă a lui 2

REDUCEREA NUMĂRULUI DE STĂRI

Automate incomplet definite

- **Numărul** claselor de compatibilităţi al mulţimii închise şi minime este cuprins între două limite:
  - **Limita superioară**: dată de numărul de clase de compatibilităţi maxime
  - **Limita inferioară**: egală cu numărul de stări conţinute în cea mai mare clasă de compatibilităţi maximă

Exemplu

REDUCEREA NUMĂRULUI DE STĂRI

- Se dă automatul descris prin tabelul de tranziţii:

S \ X	0	1	2	3
1	2/0	-/-	3/-	2/0
2	3/0	5/1	2/0	-/-
3	3/0	4/1	-/-	5/0
4	-/-	1/1	2/-	-/-
5	-/-	-/-	1/1	-/-

- Aplicând algoritmul Paull Unger se obţine:

2	2,3			
3	2,3 2,5 ⊗	4,5		
4	2,3	1,5 ⊗	1,4	
5	1,3 ⊗	X	V	1,2
	1	2	3	4

  - Stările evident compatibile au aceleaşi intrări şi aceleaşi ieşiri: 3 cu 5 sunt evident compatibile
  - La pasul al doilea se urmăreşte efectul incompatibilităţilor (de exemplu, 2 cu 5 incompatibile)

REDUCEREA NUMĂRULUI DE STĂRI

Exemplu

- Pentru determinarea claselor de compatibilităţi se construieşte un nou tabel, cu trei coloane
  - Pe prima coloană se trec stările de pe coloanele tabelului implicaţiilor (în ordinea de la dreapta la stânga)
  - Pe coloana a doua se trec stările compatibile cu stările din prima coloană
  - În coloana a treia apar clasele de compatibilităţi
  - În ultima linie a tabelului se obţin clasele de compatibilităţi maxime: {1,2}, {1,4}, {2,3}, {3,4,5}

4	5	{4,5}
3	4, 5	{4,5}, {3,4}, {3,5} $\Rightarrow$ {3,4,5}
2	3	{3,4,5}, {2,3}
1	2, 4	{3,4,5}, {2,3}, {1,2}, {1,4}

- Anumite stări apar în mai multe clase de compatibilităţi, deci **clasele de compatibilităţi nu mai sunt disjuncte**

REDUCEREA NUMĂRULUI DE STĂRI

Exemplu

- Considerăm ansamblul minimal format din trei clase de compatibilităţi: {1,2}, {1,4}, {3,4,5}
  - 1 cu 2 implică 2 cu 3 pentru a fi închisă, dar 2 cu 3 nu fac parte din aceeaşi clasă de compatibilităţi, deci mulţimea aleasă nu este soluţie!
- Dacă alegem: {1,4}, {2,3}, {3,4,5}
  - 4 cu 5 implică 1 cu 2, care nu fac parte din aceeaşi clasă, deci mulţimea aleasă nu este soluţie!
- Dacă alegem: {$\underline{1}$,2}, {2,3}, {3,$\underline{4}$,$\underline{5}$}
  - 3 cu 4 implică 1 cu 4, care nu fac parte din aceeaşi clasă, deci mulţimea nu este soluţie!

REDUCEREA NUMĂRULUI DE STĂRI

Exemplu

- Stările care apar o singură dată se numesc **stări esenţiale** (de exemplu stările 1, 4, 5 din ultima încercare)
- **Observaţie**: Pentru obţinerea mulţimii de clase de compatibilităţi cu număr minim de elemente:
  - Se pot înlătura unele dintre compatibilităţile maxime având însă grijă ca mulţimea obţinută să rămână închisă sau
  - Se pot înlătura anumite stări care nu sunt esenţiale, pentru a îndeplini condiţia de mulţime închisă
- La {1,2}, {2,3}, {3,4,5} putem elimina starea 3 din {3,4,5} şi vom obţine {**1,2**}, {**2,3**}, {**4,5**} care reprezintă o **compatibilitate minimă şi închisă**

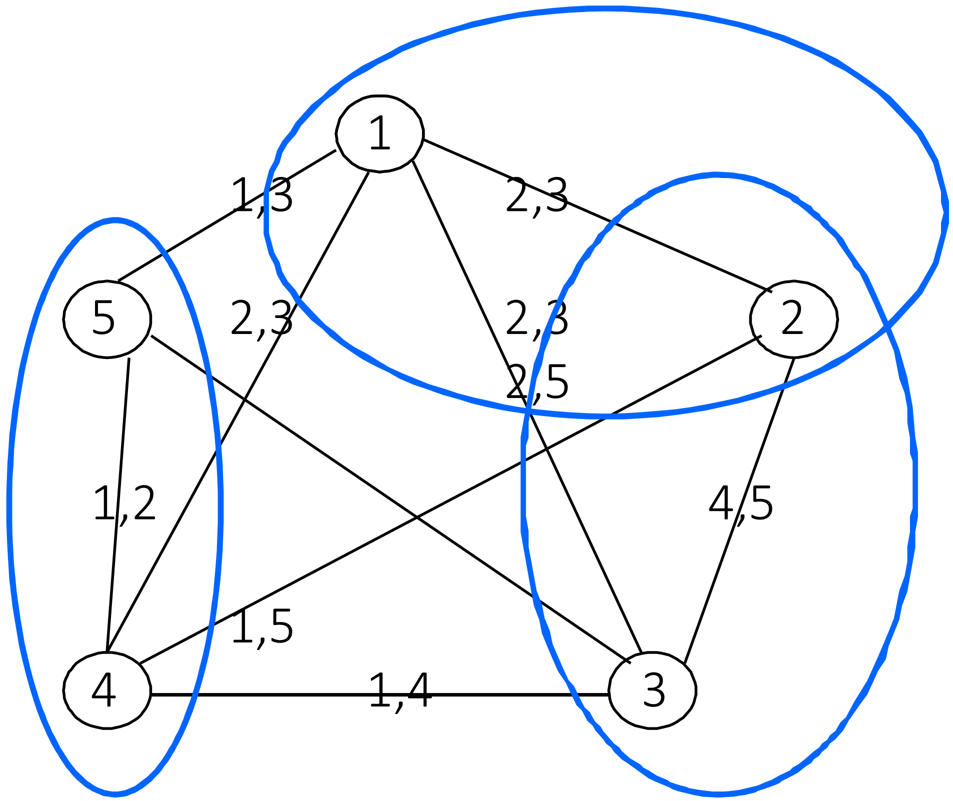
REDUCEREA NUMĂRULUI DE STĂRI

Automate incomplet definite

- Graf al implicaţiilor - metodă grafică de determinare a mulţimii minime şi închise
- Graful se construieşte după regulile:
  - Numărul nodurilor este egal cu numărul de stări ale automatului
  - Pentru fiecare pereche de stări compatibile se trasează un arc între stările respective
  - Pentru perechile de stări a căror compatibilitate implică alte compatibilităţi se va trasa un arc întrerupt pe care se notează compatibilitatea implicată
- Se poate pleca direct de la graful de tranziţie al automatului, nu este neapărată nevoie de tabelul implicaţiilor

REDUCEREA NUMĂRULUI DE STĂRI

Exemplu



S X	0	1	2	3
1	2/0	-/-	3/-	2/0
2	3/0	5/1	2/0	-/-
3	3/0	4/1	-/-	5/0
4	-/-	1/1	2/-	-/-
5	-/-	-/-	1/1	-/-

CODIFICAREA STĂRILOR

- Fiecărei **stări** din tabelul tranziţiilor unui automat i se atribuie un cod binar format din “n” cifre binare
- Dacă **m** = numărul de stări, avem $2^{n-1} < m \leq 2^n$
  - **n** = numărul de variabile binare care codifică stările
- La automatele cu multe stări atribuirea e importantă
- Pentru alegerea soluţiei optime de atribuire (codificare) se folosesc:
  - metode simple, care nu duc la soluţia optimă
  - metode mai complicate, care duc la o soluţie optimă, dar necesită mai multe calcule

CODIFICAREA STĂRILOR

Metode aproximative de codificare

- Codificarea descrie ecuaţiile stărilor următoare cu ajutorul unor variabile de stare explicitate prin variabile binare
- Considerăm automatul descris obişnuit prin tabelul de tranziţii:

S	X	00	01	11	10
	A	A/0	B/0	C/1	B/0
	B	B/0	C/0	A/0	C/0
	C	C/0	A/1	A/1	A/0

  - s, q = variabile binare; yi este un bit care intră în codificarea unei stări
  - Dacă:
    - A → s1q1 și, de exemplu: → 01, adică: s1=0 q1=1
    - B → s2q2 și, de exemplu: → 10, adică: s2=1 q2=0
    - C → s3q3 și, de exemplu: → 00, adică: s3=0 q3=0

S	X	00	01	11	10
01	s_1 q_1	s_1q_1/0	s_2q_2/0	s_3q_3/1	s_2q_2/0
10	s_2 q_2	s_2q_2/0	s_3q_3/0	s_1q_1/0	s_3q_3/0
00	s_3 q_3	s_3q_3/0	s_1q_1/1	s_1q_1/1	s_1q_1/0

CODIFICAREA STĂRILOR

Metode aproximative de codificare

- Diagramele pentru Y_1Y_2/Z, în funcţie de x_1x_2 şi y_1y_2

S \ X	00	01	11	10
01	01/0	10/0	00/1	10/0
10	10/0	00/0	01/0	00/0
00	00/0	01/1	01/1	01/0

S \ X	00	01	11	10
01	0/0	1/0	0/1	1/0
10	1/0	0/0	0/0	0/0
00	0/0	0/1	0/1	0/0

S \ X	00	01	11	10
01	1/0	0/0	0/1	0/0
10	0/0	0/0	1/0	0/0
00	0/0	1/1	1/1	1/0

- Ecuaţiile stării următoare:
  - $Y_1 = y_1\bar{y}_2\bar{x}_1\bar{x}_2 + \bar{y}_1y_2\bar{x}_1x_2 + \bar{y}_1y_2x_1\bar{x}_2$
  - $Y_2 = \bar{y}_1y_2\bar{x}_1\bar{x}_2 + \bar{y}_1\bar{y}_2\bar{x}_1x_2 + \bar{y}_1\bar{y}_2x_1x_2 + y_1\bar{y}_2x_1x_2 + \bar{y}_1\bar{y}_2x_1\bar{x}_2$

CODIFICAREA STĂRILOR

Metode aproximative de codificare

- Prin minimizare se obţine:

x_1x_2

S X	00	01	11	10
00				
01		1		1
11	x	x	x	x
10	1			

y_1y_2

x_1x_2

S X	00	01	11	10
00		1	1	1
01	1			
11	x	x	x	x
10			1	

y_1y_2

  - $Y_1 = y_1\overline{x_1}\,\overline{x_2} + y_2(\overline{x_1}x_2 + x_1\overline{x_2})$
  - $Y_2 = y_2\overline{x_1x_2} + y_1x_1x_2 + \overline{y_1y_2}(x_1 + x_2)$

- **Concluzie**:
  - Este avantajos ca un număr cât mai mare din coeficienţii s_iq_i să fie egali cu 0, pentru că astfel se anulează termenii respectivi din ecuaţia stării următoare
  - Deoarece nu pot fi făcuţi toţi coeficienţii 0, este important să se facă apoi minimizare pe baza principiului absorbţiei – termeni grupati!

CODIFICAREA STĂRILOR

Metode aproximative de codificare

- Regula 1: Stările care au aceiasi succesori primesc coduri adiacente dacă succesorii lor sunt **codificati similar** pentru aceeaşi variabilă de intrare (grupare pe aceeaşi coloană)
- Regula 2: Succesorilor unei stări li se atribuie coduri adiacente (schimbari minime pe linie!)
  - Prima regulă este prioritară faţă de a doua
  - Dacă cele 2 reguli nu duc la aceeaşi codificare, se alege codificarea rezultată din prima regulă
- **IDEE: Construirea unui nou tabel cu starile anterioare si starile urmatoare**
  - Prima regulă impune adiacenţa stărilor din coloana “stării prezente” fata de “starea anterioara”
  - A doua regulă impune adiacenţa stărilor din coloana “stării următoare” fata de “starea curenta”

CODIFICAREA STĂRILOR

Exemplu

- Se dă automatul cu tabelul de tranziţii:

S \ X	0	1
A.	D	B
B.	A	C
C	B	D
D	C	A

Stări precedente	Starea prezentă	Stări următoare
B,D	A	B,D
A,C	B	A,C
B,D	C	B,D
A,C	D	A,C

- Pentru regula 1 ne uităm în coloana “prezentă” (fata de precedenta), iar pentru regula 2 ne uităm în coloana “următoare” (fata de urmatoarea)
  - A şi C, respectiv B şi D, trebuie să aibă coduri adiacente conform regulii 1
  - B şi D, respectiv A şi C, trebuie să aibă coduri adiacente conform regulii 2
  - Cele două reguli conduc la aceeaşi codificare

CODIFICAREA STĂRILOR

Exemplu

- Se poate alege:
  - A = 00, C = 10, B = 01, D = 11 şi rezultă tabelul tranziţiilor:

y_1y_2	Y_1Y_2 pentru X	
	0	1
00 (A)	11	01
01 (B)	00	10
11 (D)	10	00
10 (C)	01	11

CODIFICAREA STĂRILOR

Metodă de codificare pe baza partiţiei stărilor

- O posibilitate de simplificare a ecuaţiei stării următoare este reducerea dependenţei unei variabile de stare de alte variabile de stare
- Ideal este ca o variabilă de stare să depindă numai de ea însăşi şi de variabilele de intrare
- Se încearcă împărţirea mulţimii stărilor în partiţii închise astfel încât toţi succesorii unei stări să facă parte dintr-o singură partiţie (aceeaşi partiţie)
  - Starile dintr-o grupare vor avea coduri adiacente, avand o parte din variabilele de stare valori comune
  - Ex: partitia 1: {00, 01}; partitia 2: {10,11}
    - Prima variabila de stare indica diferenta de partitie

CODIFICAREA STĂRILOR

Exemplu

- Se dă automatul cu tabelul de tranziţii:

S \ X	x_1	x_2
	0	1
A	D	A
B	E	C
C	F	A
D	B	E
E	A	F
F	B	D

- Mulţimea stărilor se poate împărţi în două partiţii
- Variabila de stare y3 va define schimbarea de partitie

  - (A,B,C) (D,E,F)
  - y_3 $\overline{y_3}$

CODIFICAREA STĂRILOR

Exemplu

S \ X	x_1	x_2
	0	1
A	D	A
B	E	C
C	F	A
D	B	E
E	A	F
F	B	D

- f fiind funcţia de tranziţie
  - Succesorii pentru intrarea x1 pentru prima partiţie (ABC) sunt: {f(0,A), f(0,B), f(0,C)} = {D,E,F} şi pentru a doua partiţie (DEF) sunt: {f(0,D), f(0,E), f(0,F)} = {A,B}
  - Succesorii pentru intrarea x2 pentru prima partiţie (ABC) sunt: {f(1,A), f(1,B), f(1,C)} = {A,C} şi pentru a doua partiţie (DEF) sunt: {f(1,D), f(1,E), f(1,F)} = {D,E,F}
  - Cele două partiţii determinate sunt partiţii închise

CODIFICAREA STĂRILOR

Exemplu

- Pentru 6 stări este nevoie de 3 variabile de stare $y_3 y_2 y_1$
- Alegem variabila de stare y_3 pentru **a deosebi între ele cele 2 partiţii** (ABC) şi (DEF)
- Cu celelalte 2 variabile de stare y_2 y_1 vom **distinge stările din interiorul fiecărei partiţii** (4 posibilităţi de atribuire)
  - Pentru a vedea unde este necesară **adiacenţa** în cadrul fiecărui bloc se aplică şi aici o codificare a stărilor (trebuie sa se **modifice un număr minim de variabile** de stare la **trecerea** dintr-o stare în alta)

CODIFICAREA STĂRILOR

Exemplu

- Se construiesc **diagrame de tranziţii** care vor avea în noduri stările
- Stările între care sunt posibile tranziţii sunt legate prin arce orientate

- Codificarea în partiţia a doua nu este optima (F -> D implica 2 schimbari) deci trebuie să se utilizeze o metodă pentru o codificare adiacentă!!!
- Dacă nu se poate respecta codificarea adiacentă pentru toate stările se va încerca asigurarea măcar a adiacenţei între stările care au între ele tranziţii duble (în ambele sensuri)

CODIFICAREA STĂRILOR

Metodă de codificare pe baza partiţiei stărilor

- Dificultatea acestei metode constă în determinarea rapidă a partiţiilor închise
- Determinarea sistematică a partiţiilor închise
  - Se pleacă de la o anumită pereche de stări
  - Pe baza tabelului de tranziţii se găsesc succesorii stărilor pentru toate intrările posibile
- Mulţimea perechilor de stări care se obţin va fi închisă, în cazul în care cuprinde toate stările

CODIFICAREA STĂRILOR

Exemplu

- Se analizează automatul cu tabelul de tranziţii:

S \ X	00	01	11	10
1	2/0	3/0	2/0	3/0
2	6/0	4/0	6/0	4/0
3	4/1	4/0	4/1	4/0
4	5/0	5/0	5/0	5/0
5	1/0	1/0	1/0	1/0
6	7/1	5/0	7/1	5/0
7	1/0	1/0	1/0	1/0

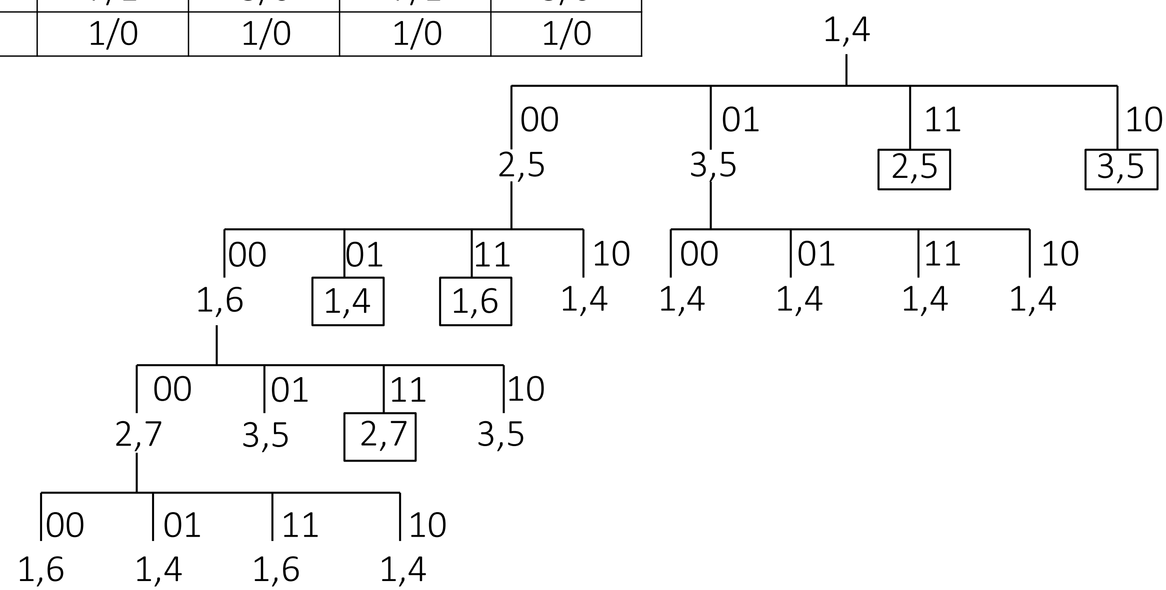
- Căutăm succesorii pentru toate combinaţiile de intrare posibile, până avem perechi de succesori care se repetă şi care se iau o singură dată
- Perechile pentru care s-a încheiat căutarea, pentru că au mai apărut o dată, se încadrează într-un dreptunghi
- Se continuă cu căutarea succesorilor pentru perechile neîncadrate
- Pentru a face o codificare corectă trebuie să se obţină toate stările automatului!

CODIFICAREA STĂRILOR

Exemplu

- Se analizează automatul cu tabelul de tranziţii

S \ X	00	01	11	10
1	2/0	3/0	2/0	3/0
2	6/0	4/0	6/0	4/0
3	4/1	4/0	4/1	4/0
4	5/0	5/0	5/0	5/0
5	1/0	1/0	1/0	1/0
6	7/1	5/0	7/1	5/0
7	1/0	1/0	1/0	1/0

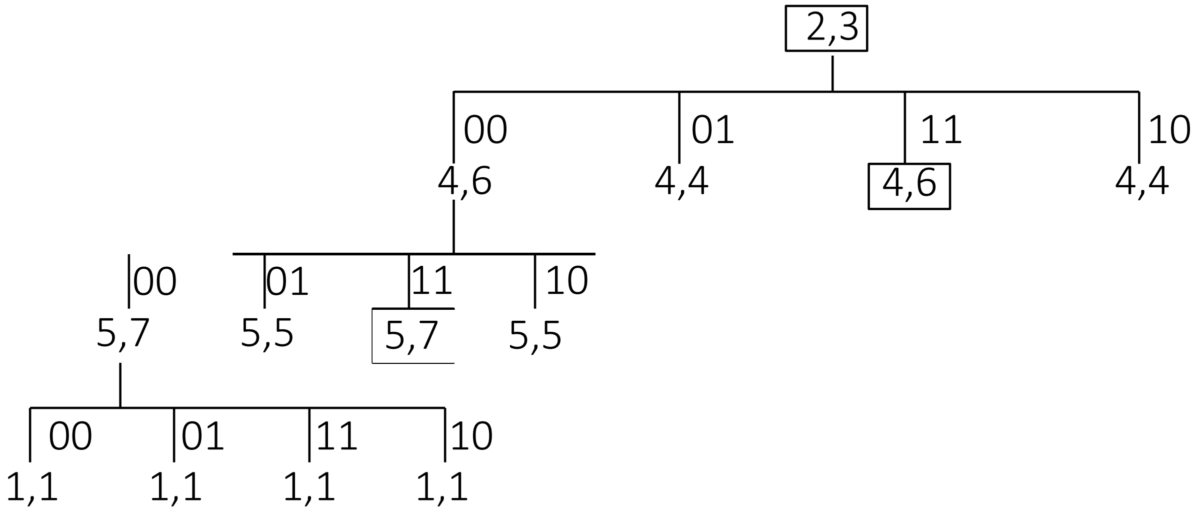


- Se obţin perechile de succesori: (1, 4), (1, 6), (2, 5), (2 ,7), (3,5)

CODIFICAREA STĂRILOR

Exemplu

- O altă variantă este dacă se pleacă de la perechea (2,3)



- Se ajunge la perechile de succesori (1), (2, 3), (4, 6), (5, 7)

Partiţia închisă este: P = {1, (2, 3), (4, 6), (5, 7)}

CODIFICAREA STĂRILOR

Exemplu

- Acestea sunt singurele partiţii închise care se pot găsi pentru automatul dat (exceptând partiţiile banale în care toate stările formează un singur bloc sau fiecare stare formează un bloc)
- Dacă se alege pentru codificare cea de-a doua partiţie închisă obţinută, pentru a distinge între 4 blocuri e nevoie de 2 variabile de stare y_2y_1
- Putem codifica, de exemplu, blocurile astfel:
  - (1) 10
  - (2, 3) 11
  - (4, 6) 01
  - (5, 7) 00

CODIFICAREA STĂRILOR

- **Exemplu**
  - Pentru distingerea stărilor în cadrul fiecărui bloc se foloseşte a treia variabilă de stare y_3
  - O codificare posibilă poate fi:

Starea	y3y2y1
1	010
2	011
3	111
4	001
5	000
6	101
7	100

- Există **mai multe posibilităţi de codificare**, dacă se codifică altfel stările din interiorul blocurilor

Concluzii

- Stari – definitie, variabile de stare
- Reducerea numarului de stari
  - Alg. Paull-Unger
  - Automate complet definite
  - Automate incomplete definite
- Codificarea starilor
  - Metode aproximative de codificare
  - Codificare pe baza partitiei starilor
- Data viitoare – automate sincrone