

Logică digitală

-Curs 11-
FSM (Automate cu
Stări finite)

Outline

- definiție FSM
 - Moore
 - Mealy
 - Exemplu
 - RTM (Register Transfer Methodology)
-

Registre

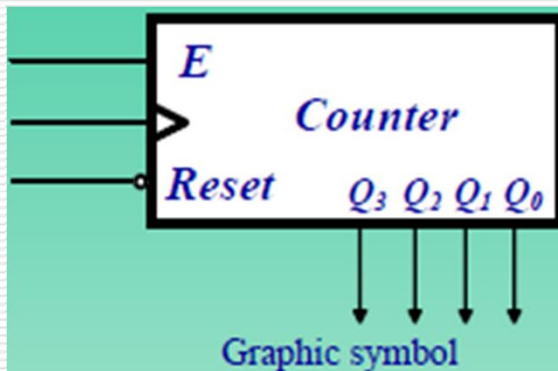
- ❑ Reprezinta o colectie/grupare de n bistabile
 - ❑ Nr maxim de valori a unui registru pe n biti – 2^n valori binare
 - ❑ Folosit pentru memorarea unui cuvant de date/unei stari curente a sistemului
-

Numărătoare

- ❑ circuite **secvențiale sincrone autonome** (mulțimea intrărilor vidă), care baleiază o secvență de stări impuse de proiectant.
 - ❑ de regulă este inițializat cu starea „0” , după care la fiecare impuls de numărare, comuta într-o nouă stare.
 - ❑ caracterul **asincron** al unui numărător este dat de faptul că impulsul de tact nu comandă simultan toate bistabilele numărătorului.
 - ❑ Funcție de direcția de parcurgere a secvenței de stări:
 - numărător în sens crescător,
 - numărător în sens descrescător,
 - numărător reversibil (ambele sensuri).
-

Numarator sincron

- Numărătoarele increm/decrem conținutul când primesc semnal de activare



<i>E</i>	<i>Operations</i>
0	No change
1	Count

Operation table

$$D_i = Q_i \oplus C_i$$
$$C_{i+1} = Q_i C_i$$

<i>Q_i</i>	<i>C_i</i>	<i>C_{i+1}</i>	<i>D_i</i>
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

HA truth table

Circuite secvențiale reprezentare

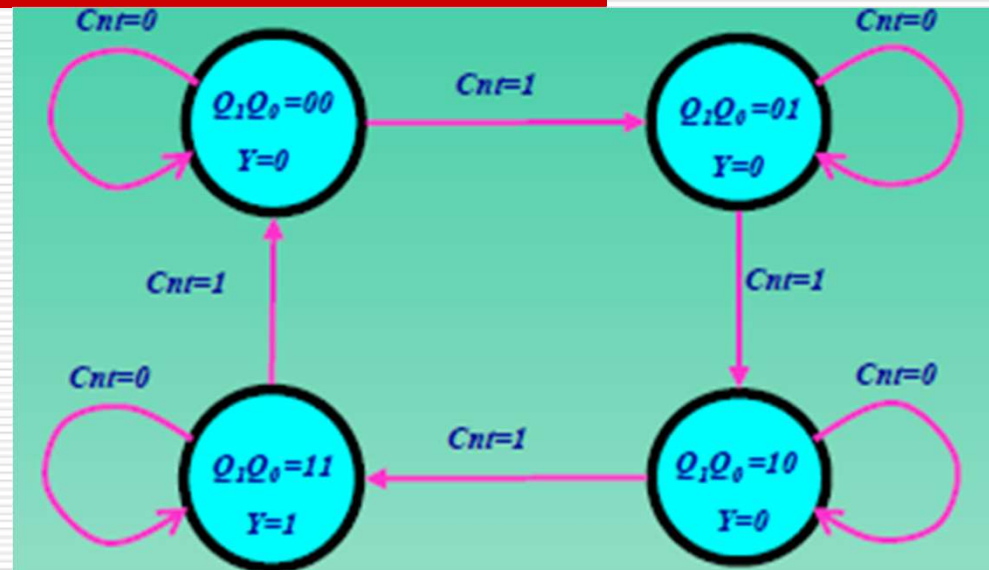
□ **Circuitele secvențiale:**

- **MEALY** sunt caracterizate prin faptul că starea următoare și ieșirea la un moment dat depind de starea prezentă și de intrarea prezentă;
- **MOORE** sunt caracterizate prin faptul că ieșirea depinde numai de starea circuitului. Starea următoare depinde de intrarea prezentă;

□ Modelele matematice ale circuitelor secvențiale se numesc în teoria comutațiilor **automate finite**.

Circuite secvențiale: diagrame e stare & tabelul tranzițiilor

□ Moore

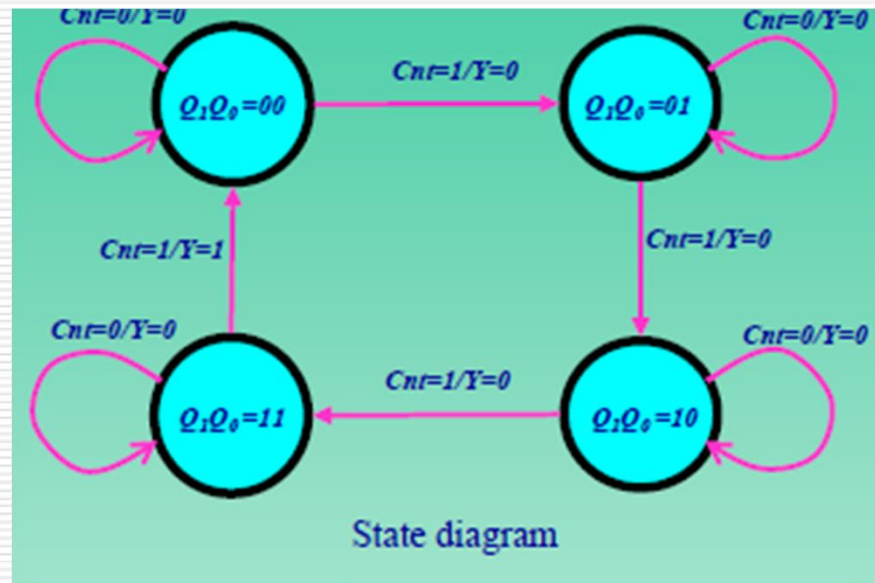


PRESENT STATE Q_1Q_0	NEXT STATE $Q_1(next) Q_0(next)$		OUTPUTS Y
	Cnt=0	Cnt=1	
00	00	01	0
01	01	10	0
10	10	11	0
11	11	00	1

State and output table

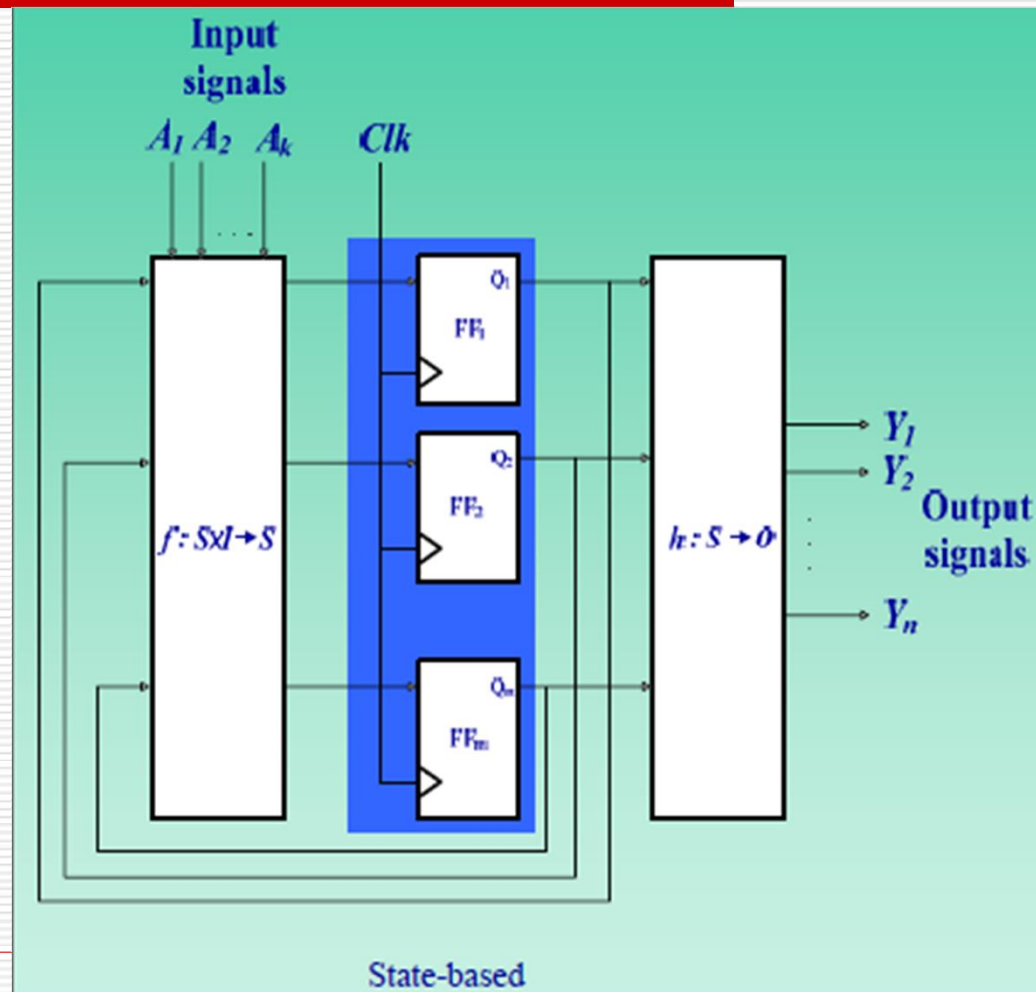
Circuite secvențiale: diagrame e stare & tabelul tranzițiilor

□ Mealy

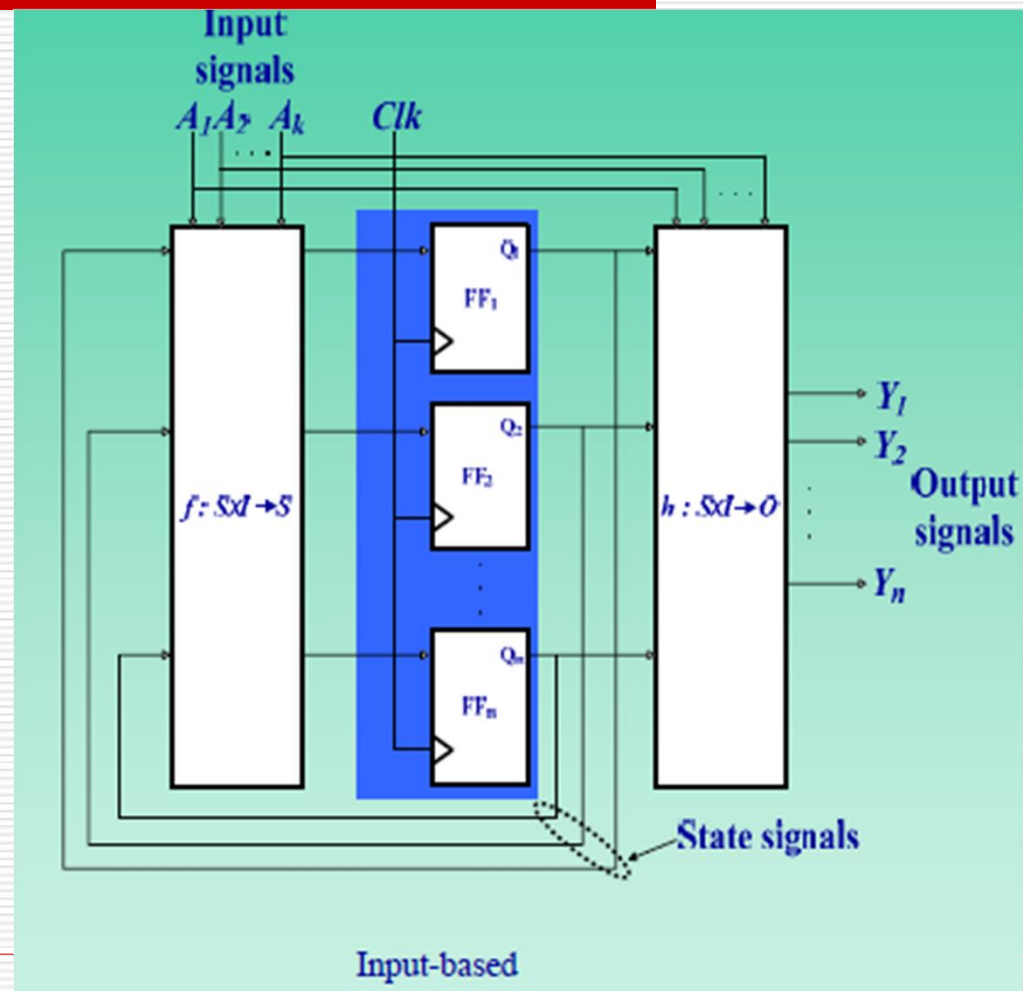


PRESENT STATE Q_1Q_0	NEXT STATE /OUTPUTS $Q_1(next) Q_0(next)/Y$	
	Cnt=0	Cnt=1
0 0	0 0 / 0	0 1 / 0
0 1	0 1 / 0	1 0 / 0
1 0	1 0 / 0	1 1 / 0
1 1	1 1 / 0	0 0 / 1

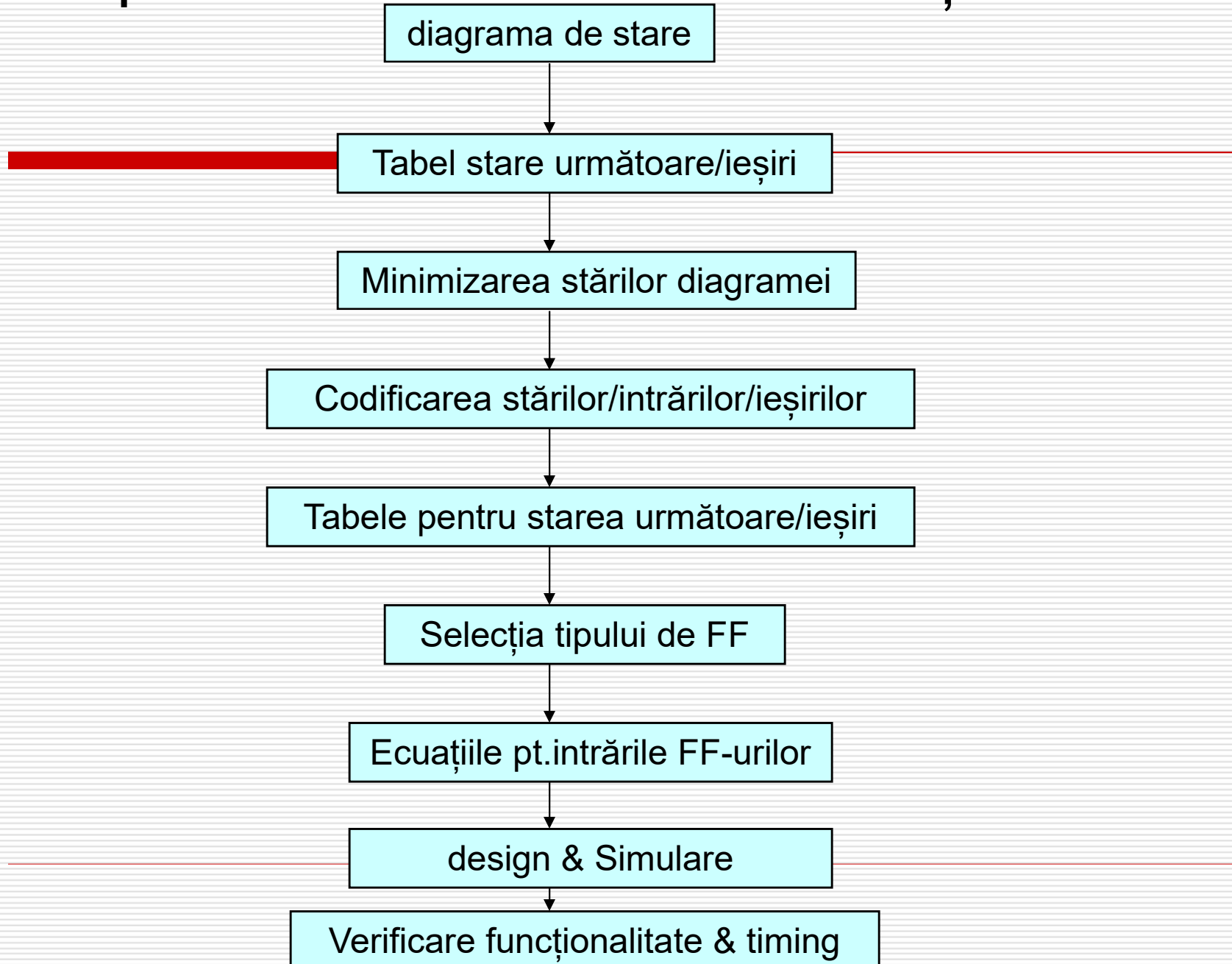
Implementare FSM Moore



Implementare FSM Mealy



Etape de sinteză circuit secvențial



Design-ul FSM-urilor folosind diagrame de stare

- ❑ Se preteaza pentru FSM-uri de dimensiune mici, respectiv medie;
 - ❑ Constructiv:
 - Un tabel de stari este o lista **exhaustiva** de stari urmatoare corespunzatoare unei combinatii: (stare curenta, intrare);
 - O diagrama de stare contine un set de arce etichetate cu conditia aferenta tranzitiei catre starile urmatoare.
-

Design-ul FSM-urilor folosind diagrame de stare

- ❑ Se preteaza pentru FSM-uri de dimensiune mici, respectiv medie;
- ❑ Constructiv:
 - Un tabel de stari este o lista exhaustiva de stari urmatoare corespunzatoare unei combinatii: (stare curenta, intrare);

Chiar daca sunt mai multe intrari, numai o singura expresie corespunzatoare tranzitiei este marcata pe arc!

Design-ul FSM-urilor folosind diagrame de stare

- ❑ Se preteaza pentru FSM-uri de dimensiune mica, respectiv medie;
- ❑ Constructiv:

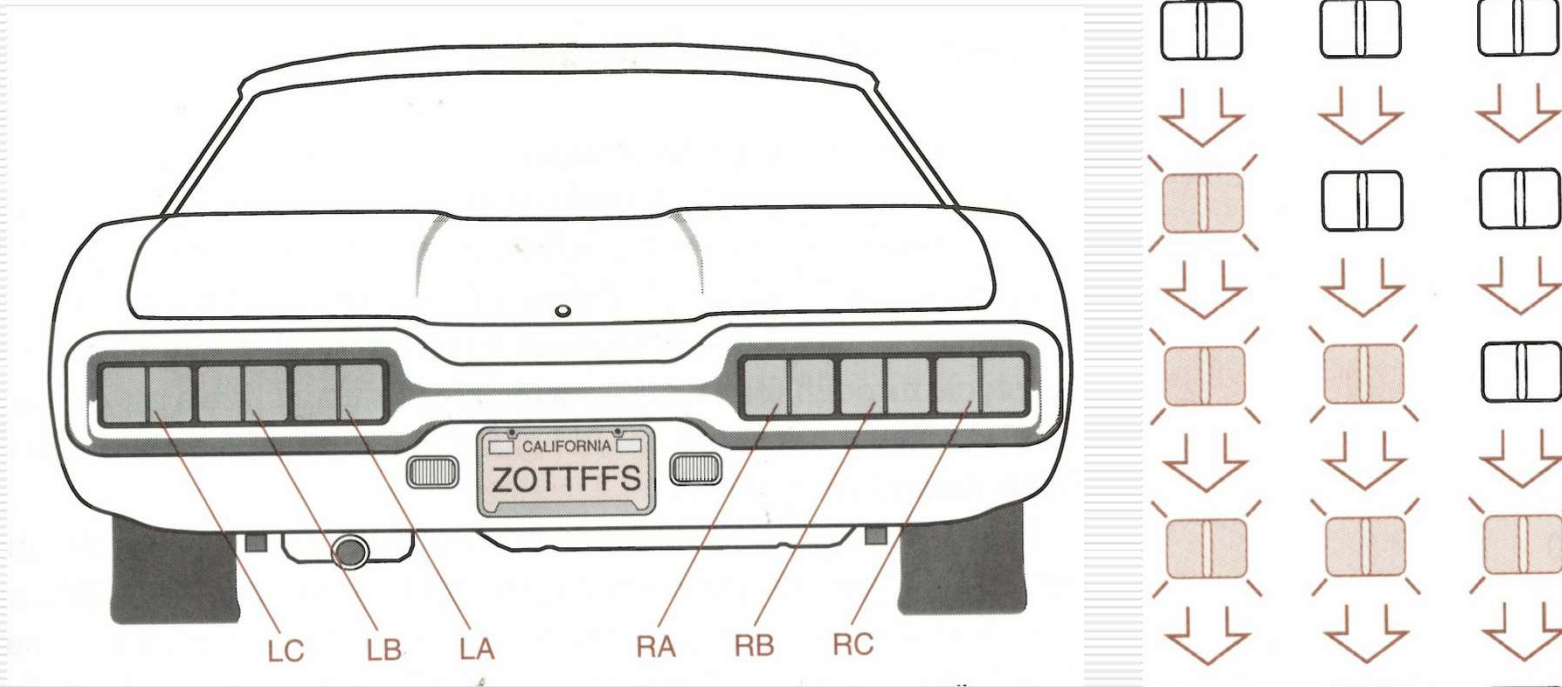
Probleme care pot aparea:

1. Diagrama de stare ambigua! Pot aparea situatii in care sa existe perechi (stare, intrari) pentru care sa nu fie specificata starea urmatoare.
2. Nu este garantat faptul ca conditiile pentru tranzitie asociate arcelor care pleaca dintr-o stare acopera toate combinatiile de intrare exact o data.

Exemplu: Ford Thunderbird 1965

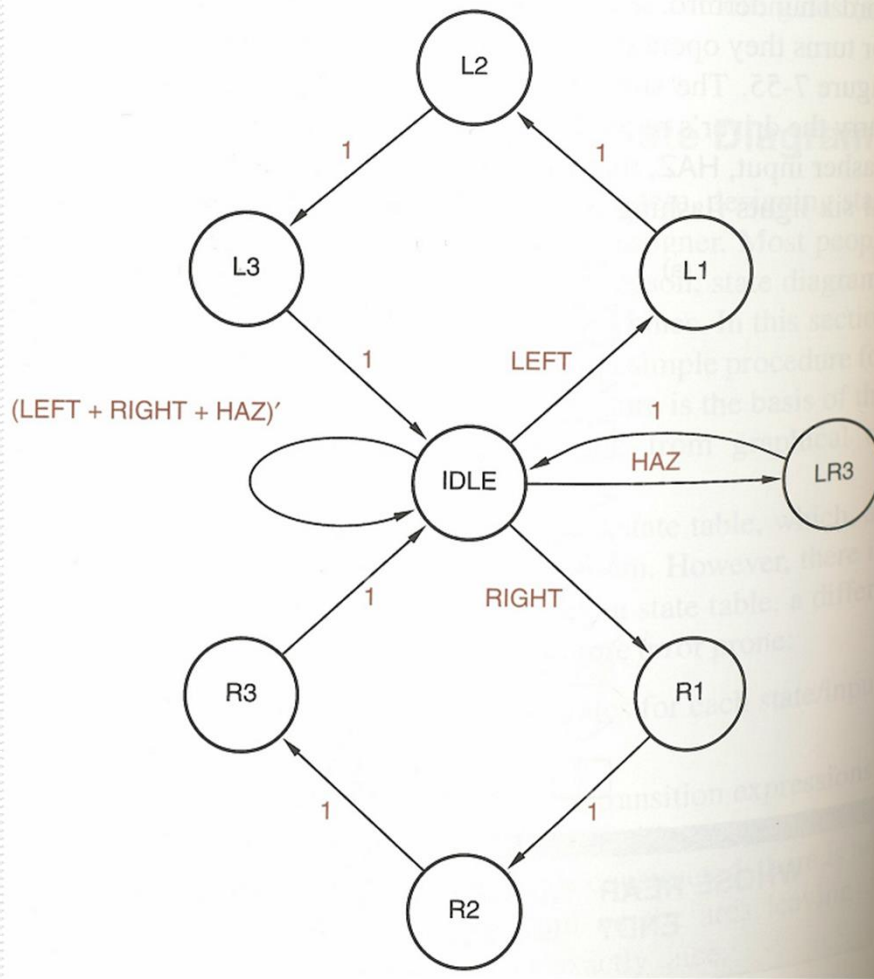
Ex.: semnalizare Ford Thunderbird 1965

- Realizati diagrama de stare pentru sistemul de semnalizare a unei masini Ford.



Ex. Preluat din Digital Design Principles and Practices, Cap 7, J.F. Wakerly, 4th Edition

Ex.: semnalizare Ford diagrama Moore



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

Ecuatiile iesirilor

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

$$LB = L2 + L3 + LR3$$

$$LC = L3 + LR3$$

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

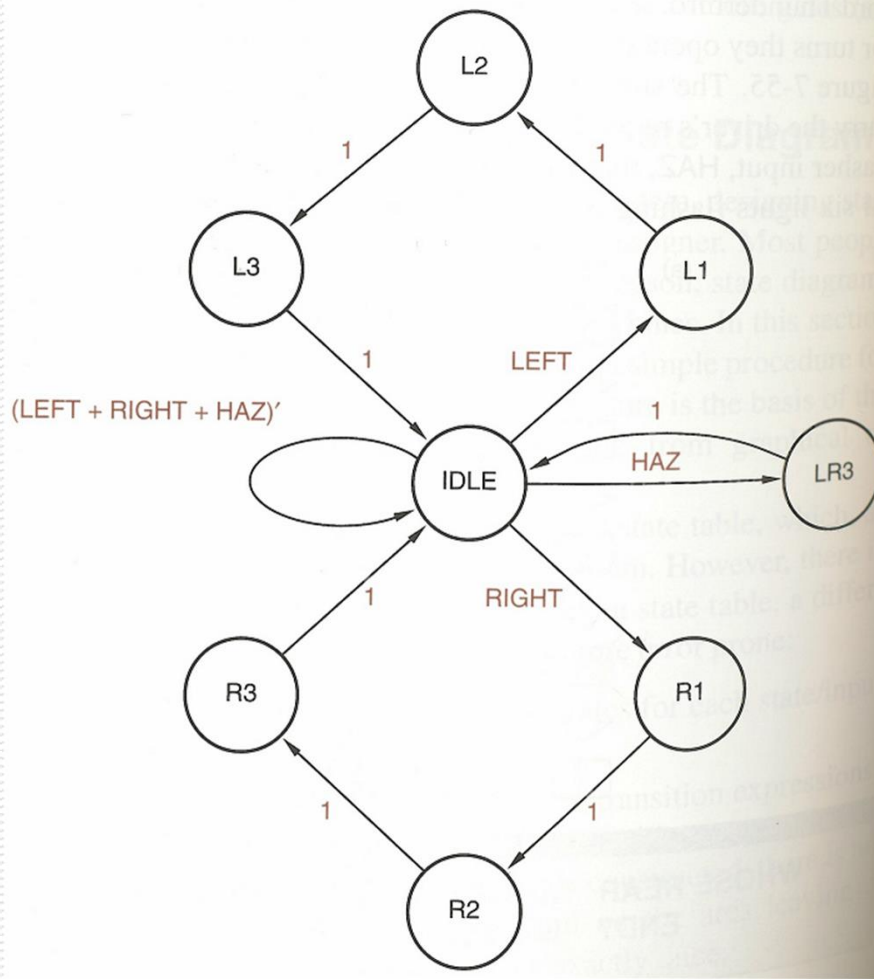
$$RB = R2 + R3 + LR3$$

$$RC = R3 + LR3$$

Output Table

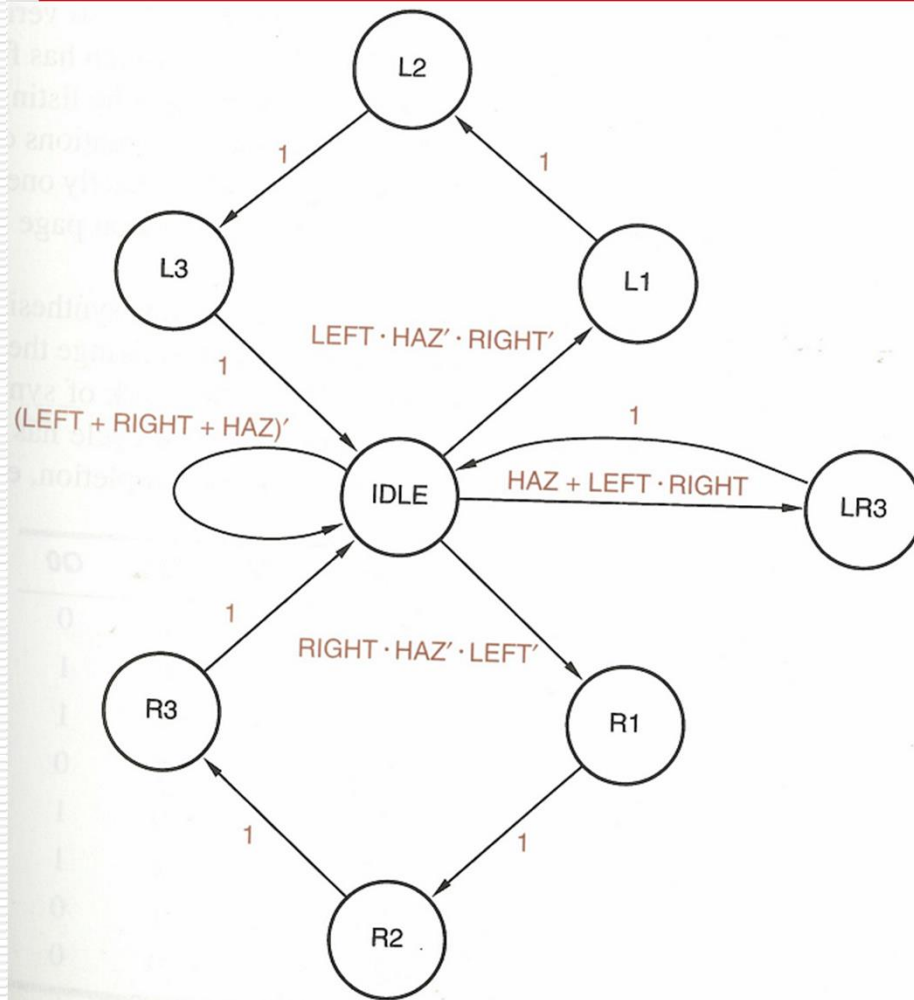
State	LC	LB	LA	RA	RB	RC
IDLE	0	0	0	0	0	0
L1	0	0	1	0	0	0
L2	0	1	1	0	0	0
L3	1	1	1	0	0	0
R1	0	0	0	1	0	0
R2	0	0	0	1	1	0
R3	0	0	0	1	1	1
LR3	1	1	1	1	1	1

Probleme?



- Ce se intampla daca **LEFT** si **HAZ** sunt asertuite simultan?

Probleme?

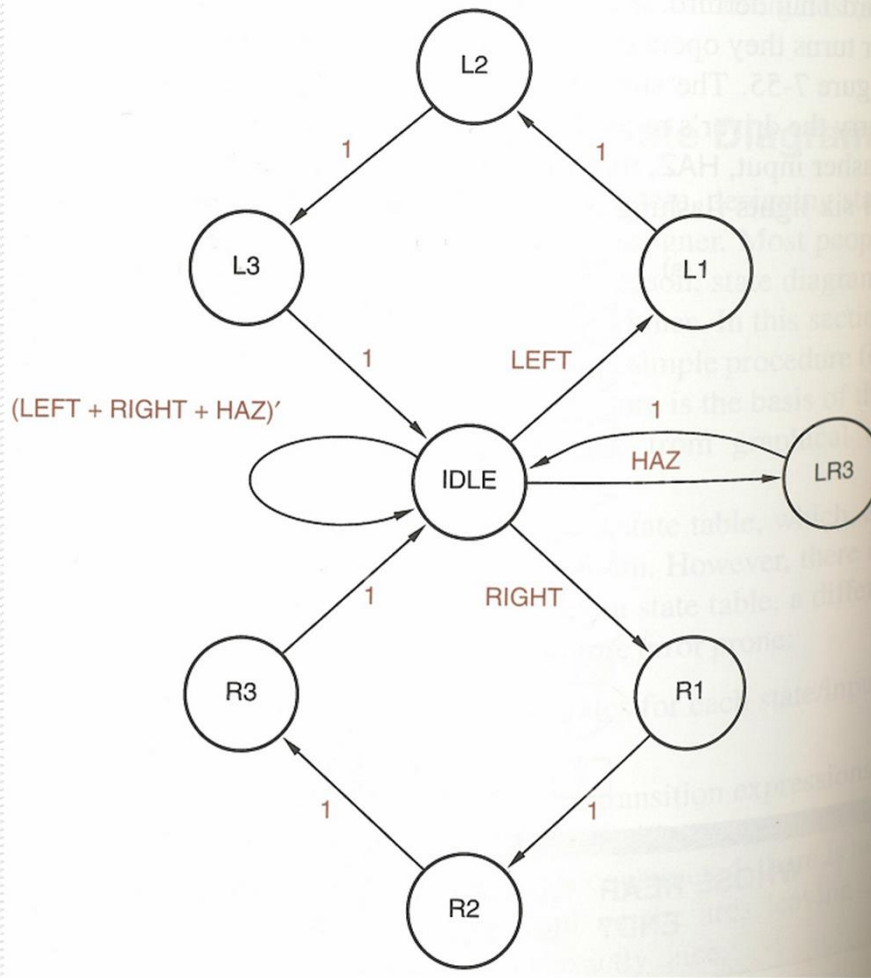


- ❑ Solutie: Ii dam la intrarea HAZ prioritate, iar cazul LEFT si RIGHT asertuite simulatan il tratam ca si hazard.
- ❑ Noua diagrama nu e ambigua: conditiile de pe arce se exclud reciproc si surprind toate combinatiile de intrare!

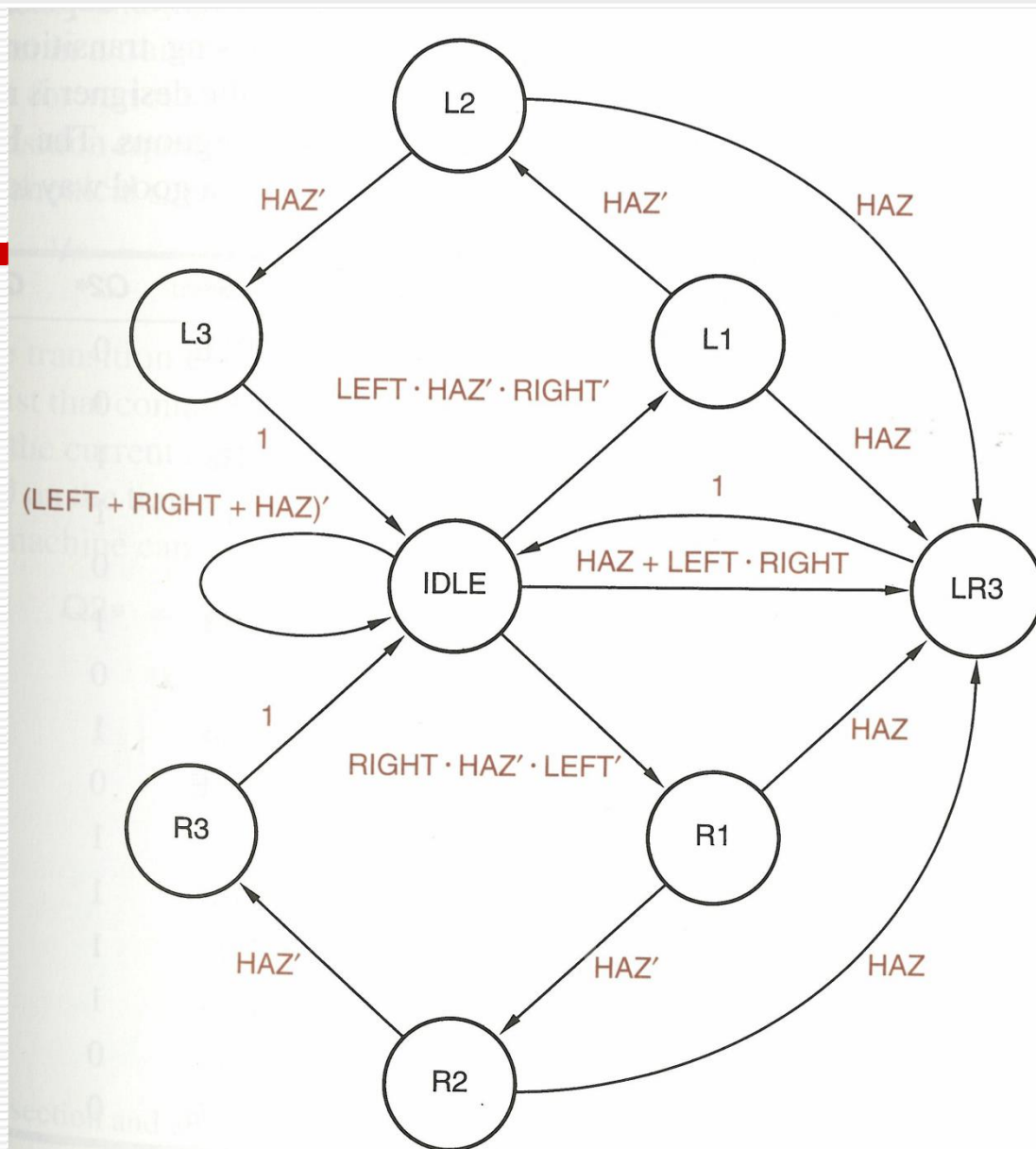
Diagrama Ne-amigua

- ❑ Mutual exclusion: produs logic intre oricare 2 expresii a tranzitiei pentru oricare 2 arce care pleaca dintr-o stare este 0.
 - ❑ All inclusion: suma logica a expresiilor tranzitiilor tuturor arcelor care pleaca dintr-o stare este 1.
-

Probleme?

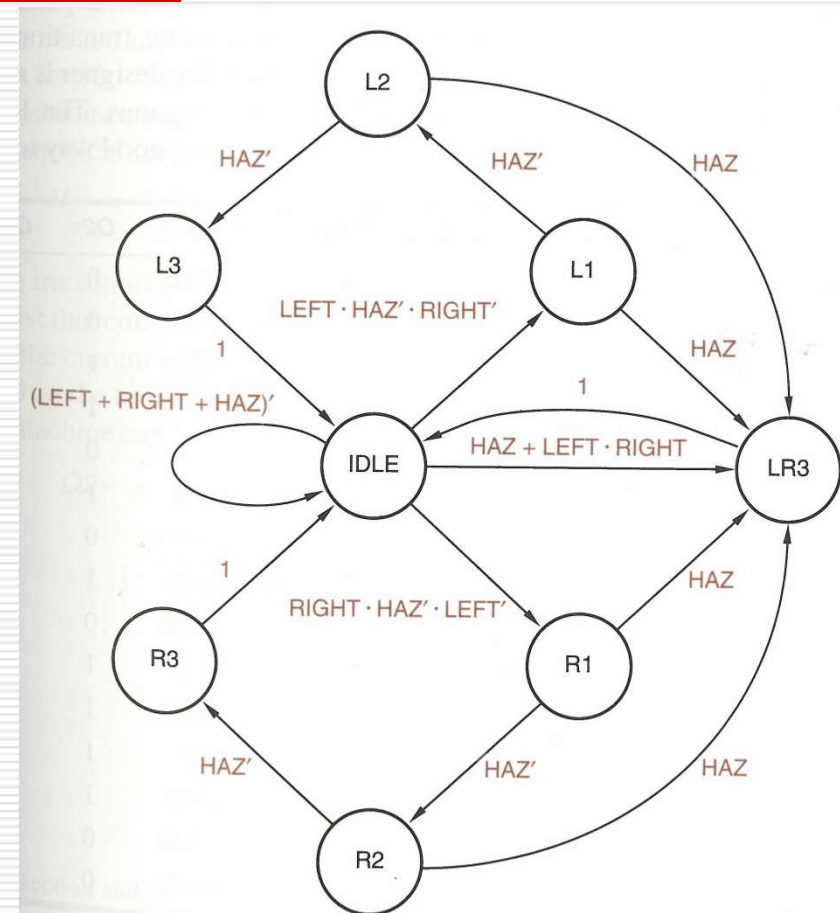


- Ar fi de dorit pentru utilizator ca semnalul de hazard sa fie prioritar.
- Adica daca esti intr-o secventa de semnalizare la stanga in starea L1 si HAZ este activ sa treci direct in starea de hazard LR3

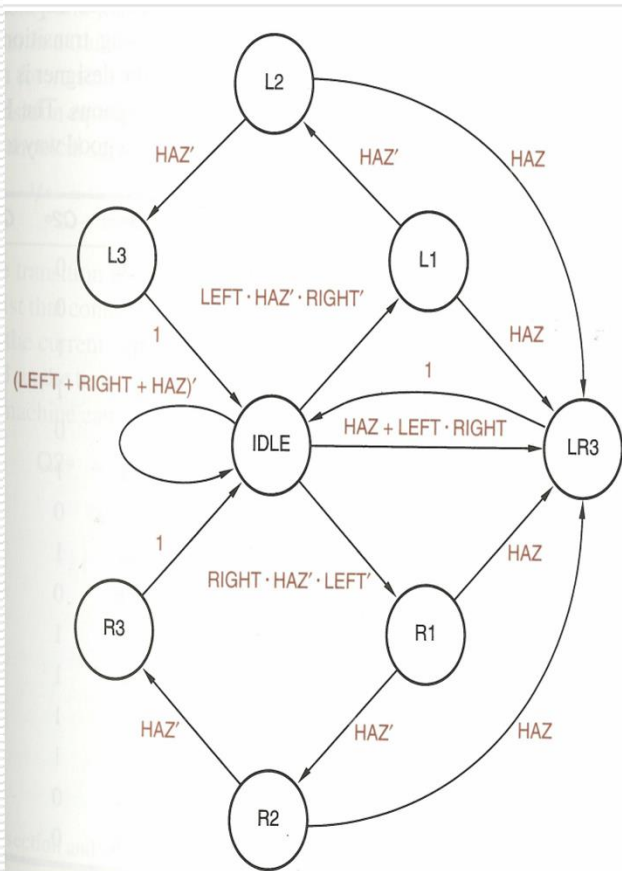


Codificarea starilor

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



Realizarea unei liste de tranzitii



S	Q2	Q1	Q0	Transition Expression	S*	Q2*	Q1*	Q0*
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

Sinteza

- ❑ Model 2 segment;
 - ❑ Responsabilitatea design-erului este scrierea unei descrieri de automat de stari neambigua;
 - ❑ Realizata de CAD-uri;
 - Eliminarea tranzitiilor duplicate;
-

RTM (Register Transfer Methodology)

- ❑ descriere foarte similară cu cea utilizată pentru algoritmi, și anume descrie fluxul de date prin prisma operațiilor care au loc la nivel de registre.
- ❑ se poate realiza o analogie între operații la nivel de registru și variabilele dintr-un algoritm.
- ❑ Nivelul RT (register transfer) de abstractizare este situat între nivelul poartă logică și nivelul procesor.

RTM

- ❑ situat între nivelul poartă logică și nivelul procesor.
 - ❑ este descris în termeni de operație RT de bază.
 - ❑ o unitate digitală este descrisă ca o succesiune de operații RT de bază, succesiune dictată de o logică (modul) de control.
-

Operația RT elementară

- este în esență o funcție simplă care calculează valoarea a unui registru destinație pe baza conținutului registrelor sursă și a semnalelor de intrare.

$$R_{DEST} \leftarrow f(R_{SRC1}, R_{SRC2}, \dots, R_{SRC_M})$$

Operația RT elementară

- ☐ Pe frontul crescător al tactului, informația disponibilă la intrare este încărcată în registre. După o întârziere (timp de propagare) ea este disponibilă la ieșiri.
 - ☐ O logică combinațională calculează funcția f funcție de intrări și valorile încărcate în registru.
 - ☐ Rezultatul calc. este trimis printr-o logică combinațională la intrările registrului destinație (ex. MUX).
 - ☐ La proximal front crescător al semnalului de tact, rezultatul de la intrarea registrului destinație este încărcat în registru.
-

RT exemplu: suma

$$R_1 \leftarrow R_1 + R_2$$



Considerăm următoarele sufixe pentru registre:

Sufix `_reg` – face referire la ieșirea curentă (din timpul impulsului de tact curent) a registrelor;

Sufix `_next` – face referire la datele care sunt la intrarea registrelor, calculații complete sau rezultate intermediare (din timpul impulsului de tact curent);

Atragem atenția asupra comportamentului registrelor de a încărca datele primite la intrare **numai** la momente bine stabilite de timp (în discuția noastră pe frontul crescător al impulsului de tact).

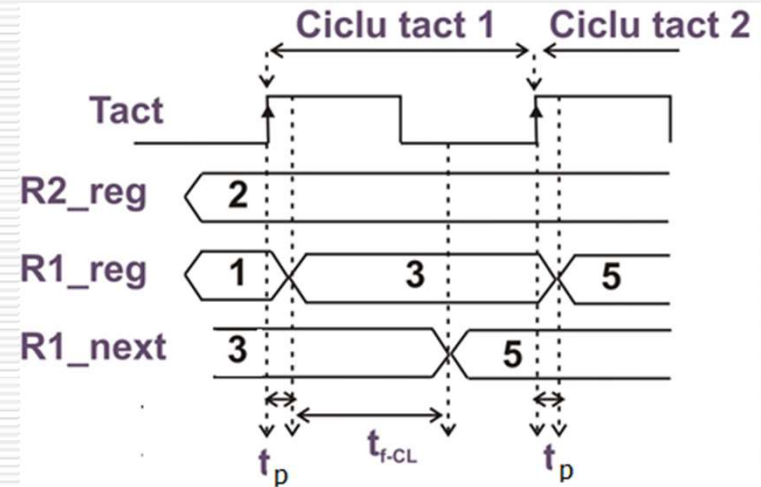
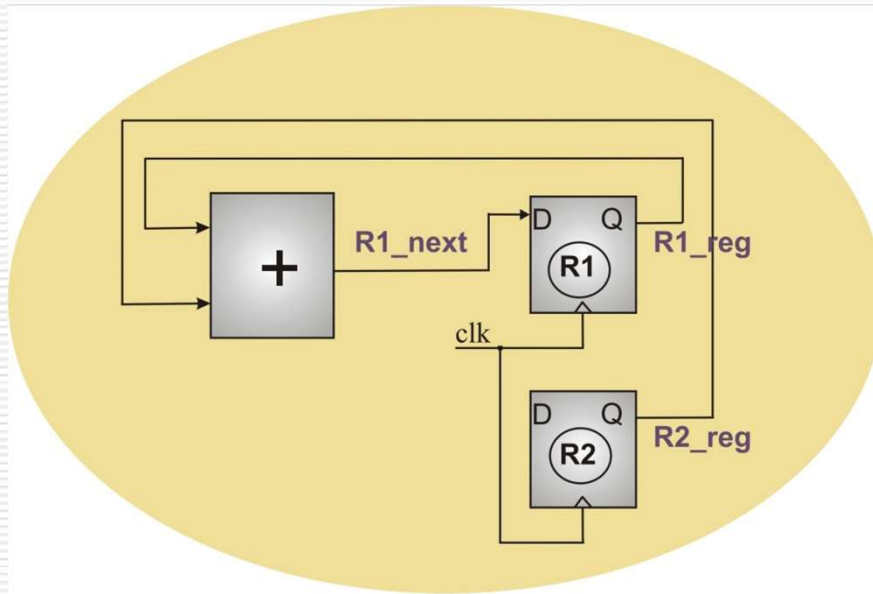
Așadar operația RT se poate exprima astfel:

$$R_1_next \leftarrow R_1_reg + R_2_reg$$

$$R_1_reg \leftarrow R_1_next \quad \text{la intalnirea primului front crescator al tactului}$$

Ex.: suma

□ Rezultat sinteza & diagrama de timp



Mai multe operatii RT

Ex.:

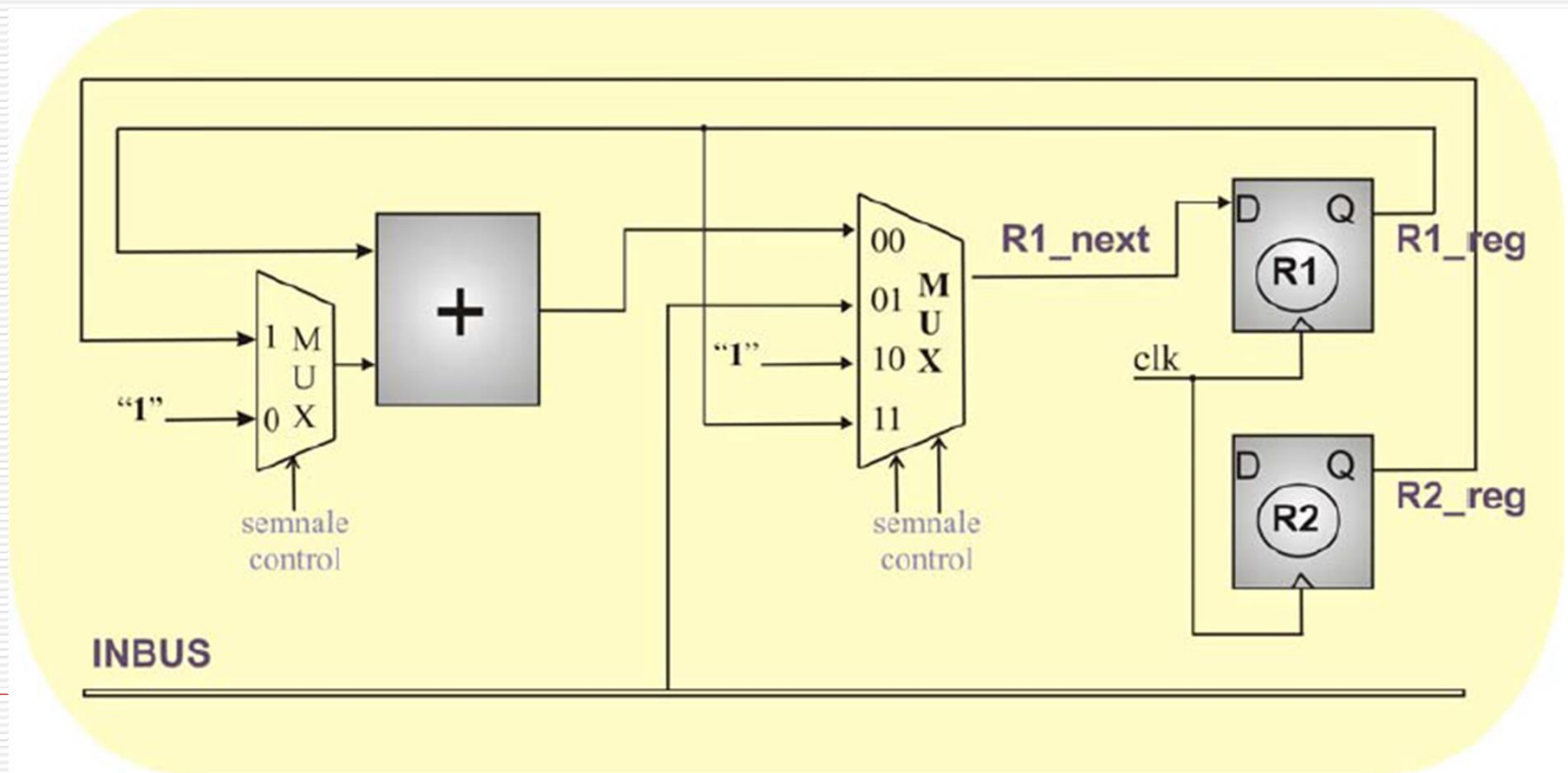
$$R_1 \leftarrow R_1 + R_2$$

$$R1 \leftarrow R1 + 1$$

$$R1 \leftarrow InBus$$

$$R1 \leftarrow 1$$

$$R1 \leftarrow R1 \quad - \text{nop}$$

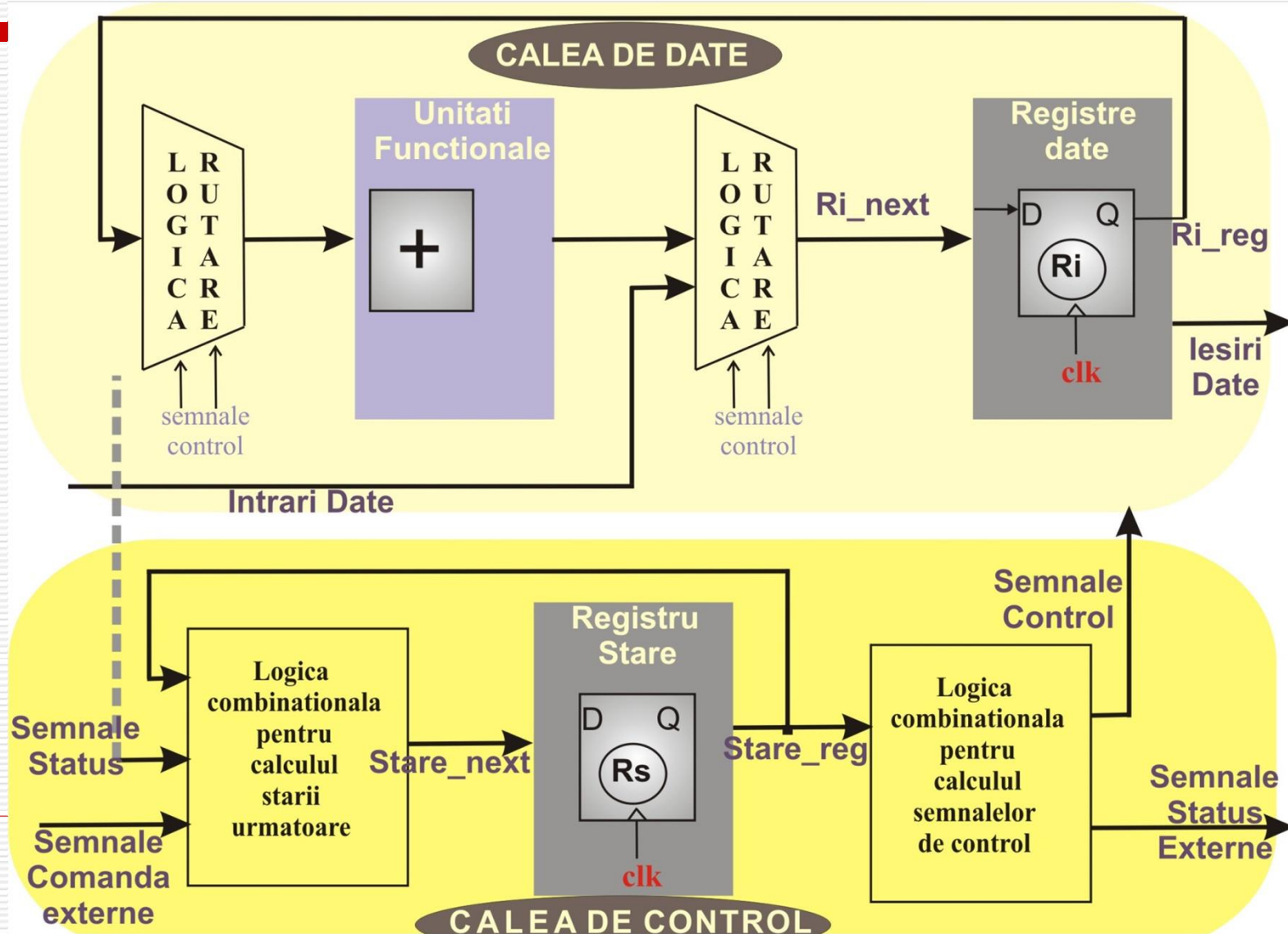


Automat cu stări finite și cale de prelucrare a datelor (FSMD)

O implementare RTL necesită:

- Cale date:
 - prelucrarea și rutarea datelor de către elemente secventiale de memorare.
 - unități funcționale de prelucrare a datelor
 - logică de rutare (de regulă reprezentată de multiplexoare)
 - registre pentru stocarea datelor
 - cale de control: dictează când și ce operație RT se execută (FSM).
-

FSMD



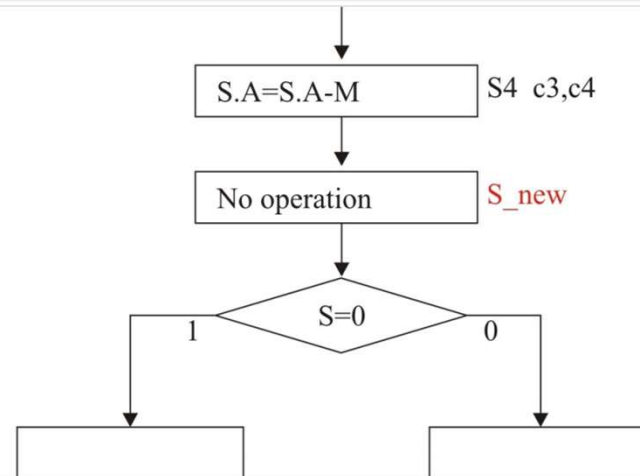
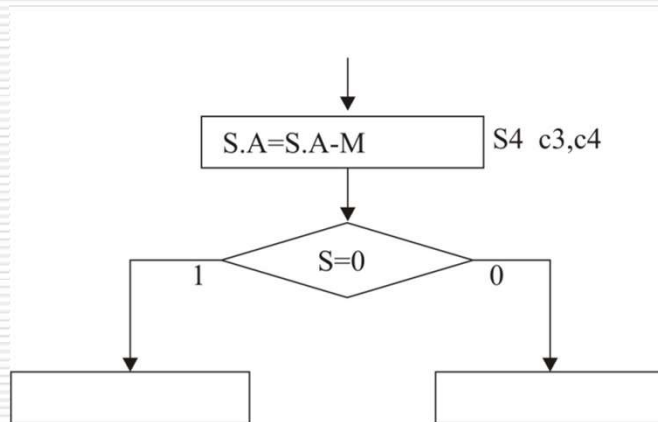
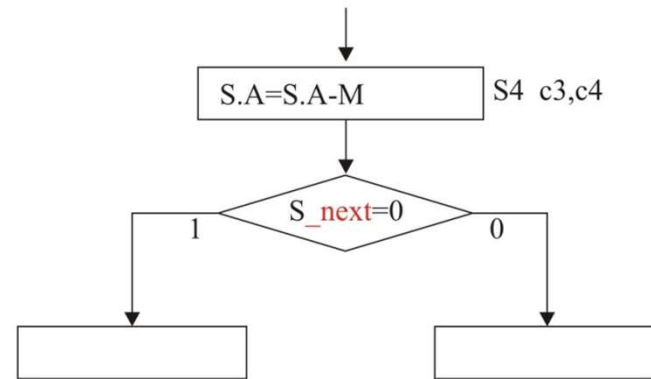
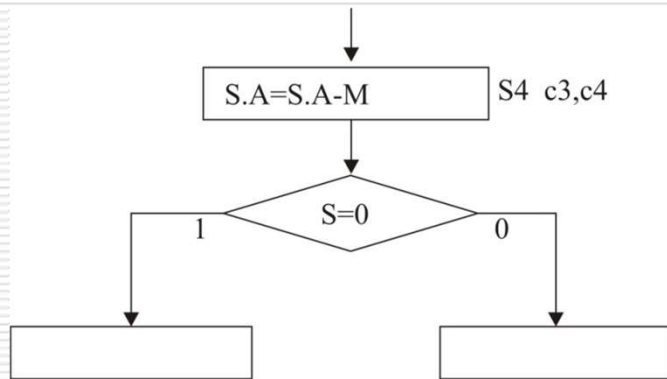
Algorithmic state machine

- ❑ constă din blocuri de decizie, blocuri în care se efectuează secvențe de RT ops, blocuri de start/stop.
 - ❑ uneori blocurile de test au nevoie să verifice condiția pe valoarea de intrare a unui registru.
-

ASM charts

- testează informația din registru care a fost modificată în *pasul curent* (așadar nu a beneficiat de un front crescător care să ducă la încărcarea ei în registru corespunzător). Acest scenariu prezintă două soluții:
 - Introducerea unei stări echivalente unui nop (no operation) care să ofere timpul necesar update-ului;
 - Folosirea pentru testare a valorii de *_next* (valoarea de la intrare care încă nu este încărcată în registru);
-

ASM charts



Ex.: Fibonacci

$$fib(n) = \begin{cases} 0, & n=0 \\ 1, & n=1 \\ fib(n-1) + fib(n-2), & n>1 \end{cases}$$

INPUT:

An[31:0]=1;

An-1[31:0]=1;

An-2[31:0]=0;

N[7:0]=INBUS;

{S1, c0}

BEGIN:

If (N=0) then goto OUTPUT;

An=An-1+An-2; {S2,c1}

An-2=An-1; {S3,c2}

An-1=An; }
N=N-1; } {S4,c3}

goto BEGIN

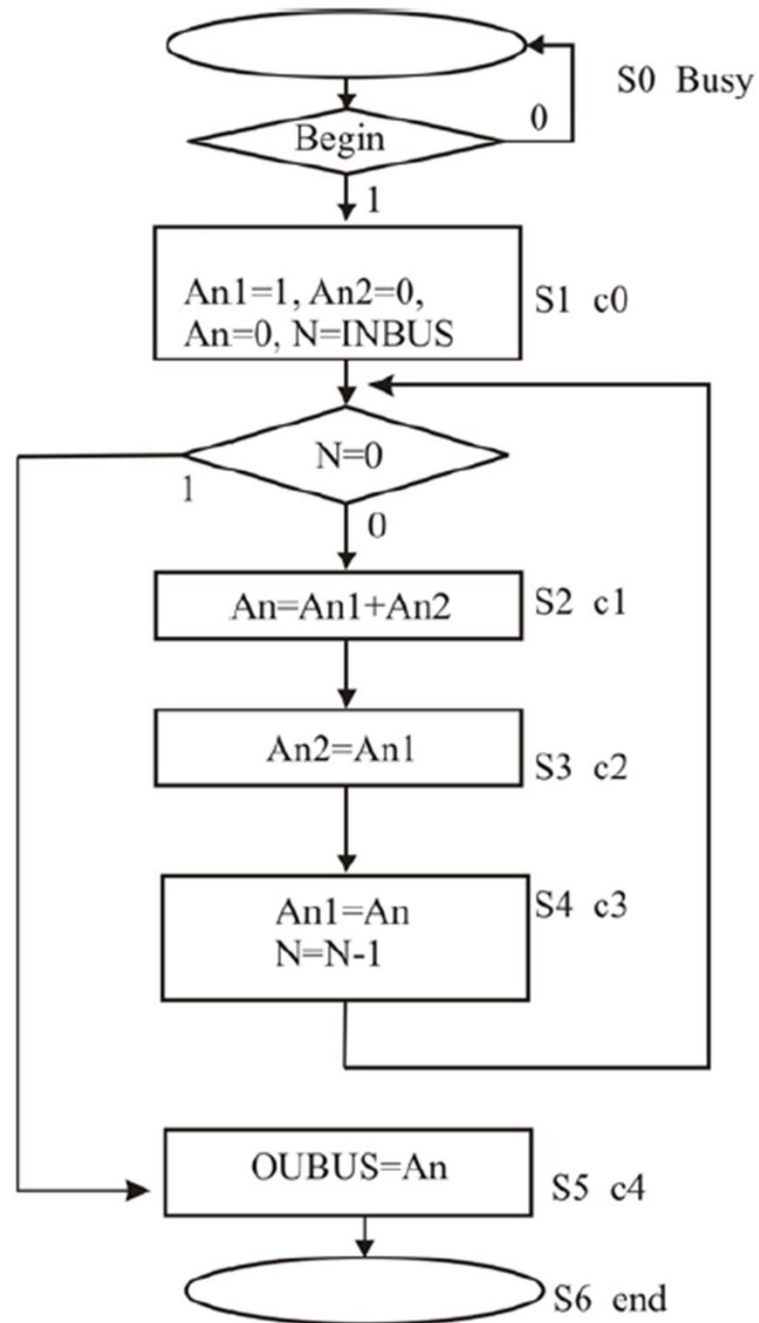
OUTPUT:

OUTBUS=An;

{S5,c4}

c5=END

ASM chart



Întrebări?

**Enough Talking Let's Get To It
!!001race Yourself!!**

