Universitatea Politehnica Timişoara
P-ţa Victoriei nr. 2
RO 300006 - Timişoara
Tel: +4 0256 403000
Fax: +4 0256 403021
rector@rectorat.upt.ro
www.upt.ro

Logică digitală

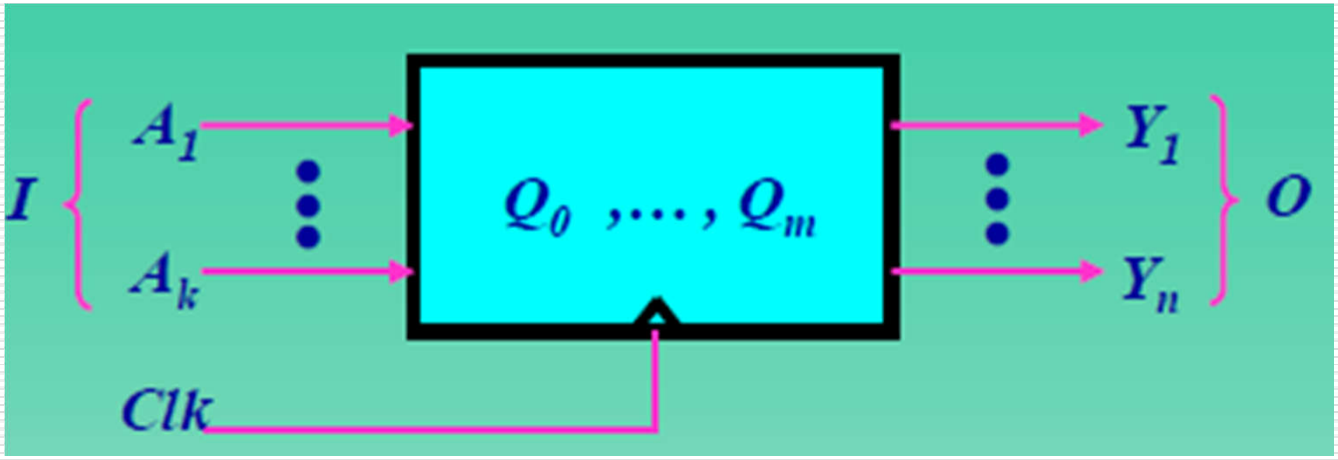
-Curs 10-11-

FSM (Automate cu Stări finite)

Outline

- Definiție FSM
  - Moore
  - Mealy
- Reducerea stărilor
  - Exemplu
- Codificarea stărilor
  - Număr minim de tranziții
  - Adiacență pe bază de priorități
  - One hot

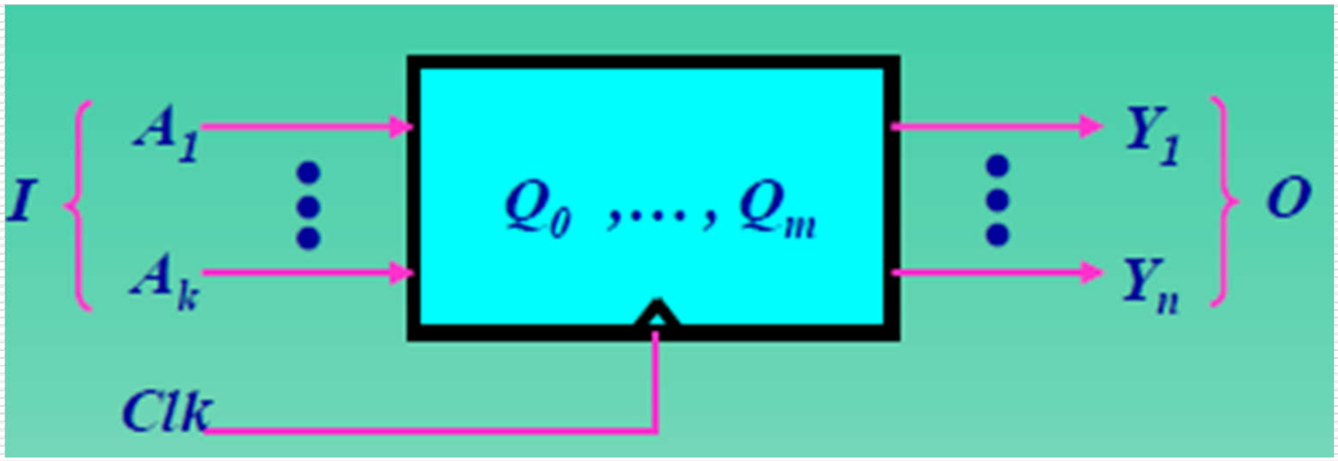
Automate cu stări finite



- Cvadruplul $\langle S, I, O, f, h \rangle$
  - S – mulțimea stărilor
  - I – mulțimea intrărilor
  - O – mulțimea ieșirilor
  - f- funcțiile pt.starea urm.; h – funcțiile pt.ieșire

$$S = Q_1 \times Q_2 \times \ldots \times Q_m ,$$
$$I = A_1 \times A_2 \times \ldots \times A_k ,$$
$$O = Y_1 \times Y_2 \times \ldots \times Y_n ,$$

Automate cu stări finite



- Cvadruplul $\langle S, I, O, f, h \rangle$
  - S – mulțimea stărilor
  - I – mulțimea intrărilor
  - O – mulțimea ieșirilor
  - f- funcțiile pt.starea urm.; h – funcțiile pt.ieșire

$f : S \times I \longrightarrow S$

$h : S \times I \longrightarrow O$ (Mealy-type)

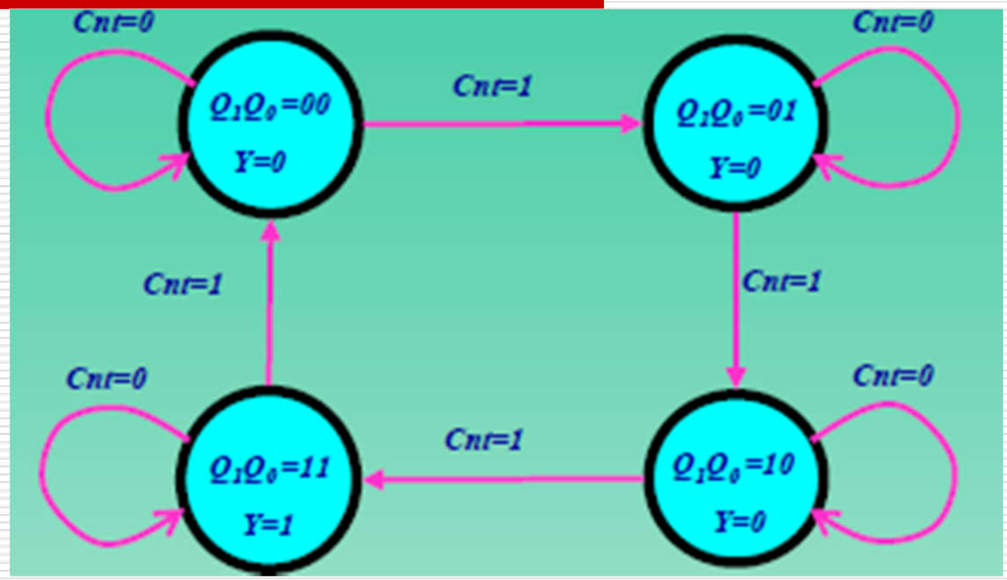
$S \longrightarrow O$ (Moore-type)

Circuite secvenţiale reprezentare

- **Circuitele secvenţiale:**
  - **MEALY** sunt caracterizate prin faptul că starea următoare şi ieşirea la un moment dat depind de starea prezentă si de intrarea prezentă;
  - **MOORE** sunt caracterizate prin faptul că ieşirea depinde numai de starea circuitului. Starea următoare depinde de intrarea prezentă;
- Modelele matematice ale circuitelor secvenţiale se numesc in teoria comutaţiilor **automate finite**.

Circuite secvențiale: diagrame e stare & tabelul tranzițiilor

- Moore

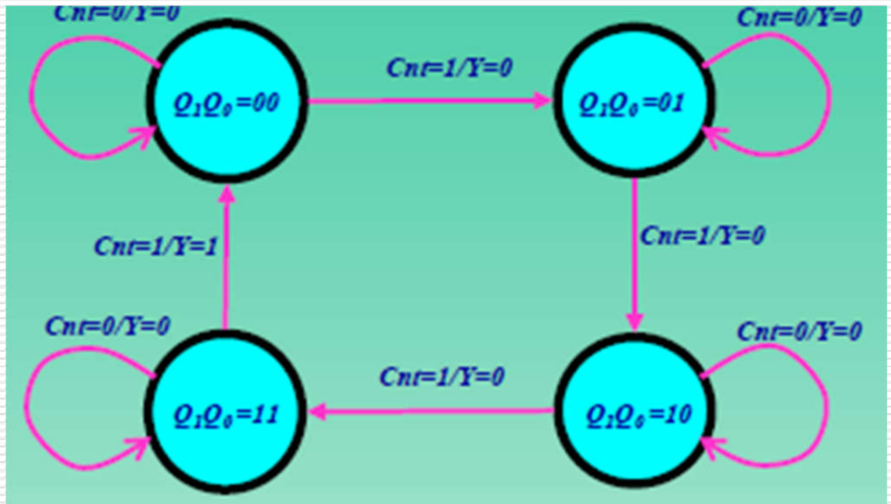


PRESENT STATE Q_1Q_0	NEXT STATE $Q_1(next)\ Q_0(next)$		OUTPUTS Y
	Cnt=0	Cnt=1	
0 0	0 0	0 1	0
0 1	0 1	1 0	0
1 0	1 0	1 1	0
1 1	1 1	0 0	1

State and output table

Circuite secvențiale: diagrame e stare & tabelul tranzițiilor

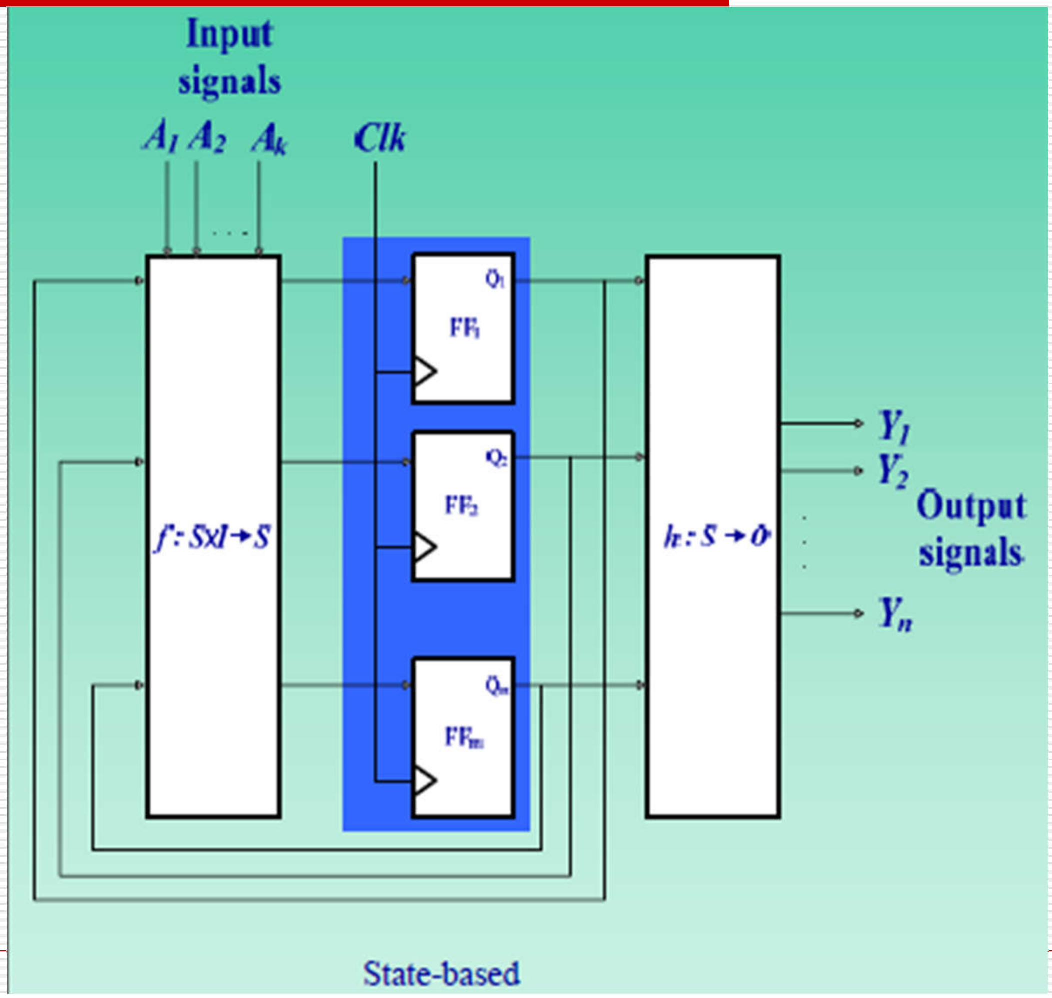
- Mealy



State diagram

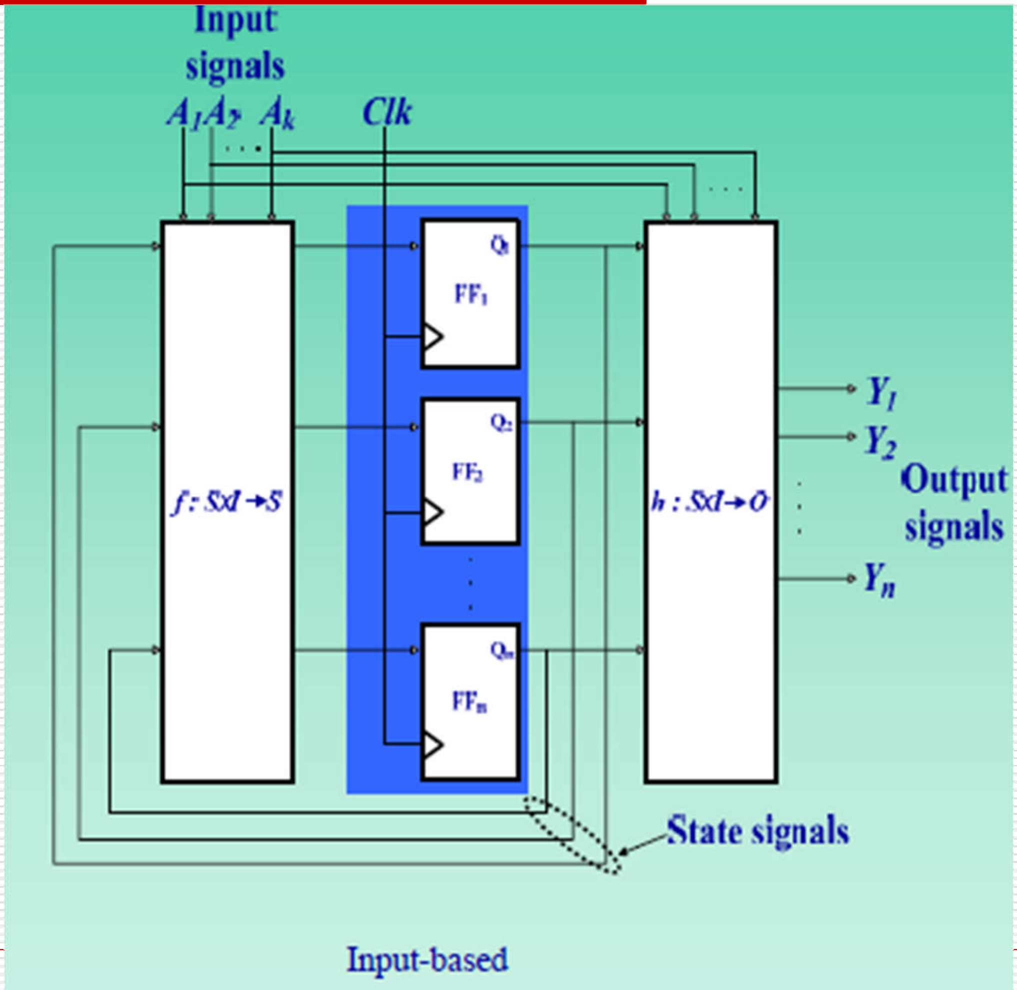
PRESENT STATE Q_1Q_0	NEXT STATE /OUTPUTS $Q_1(next)\ Q_0(next)/Y$	
	Cnt=0	Cnt=1
0 0	0 0 / 0	0 1 / 0
0 1	0 1 / 0	1 0 / 0
1 0	1 0 / 0	1 1 / 0
1 1	1 1 / 0	0 0 / 1

Implementare FSM Moore



 Slides by Philip Pham, University of California, Irvine

Implementare FSM Mealy



Etape de sinteză circuit secvențial

Diagrama de stare

Tabel stare următoare/ieșiri

Minimizarea stărilor diagramei

Codificarea stărilor/intrărilor/ieșirilor

Ecuațiile pentru starea următoare/ieșiri

Selecția tipului de FF

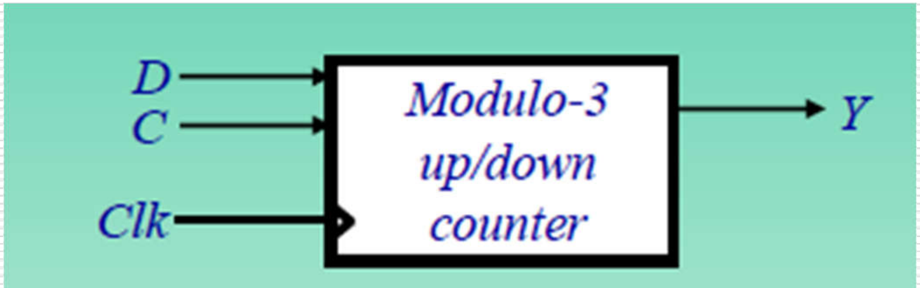
Ecuațiile pt.intrările FF-urilor

Design & Simulare

Verificare funcționalitate & timing

Exemplu: numărător modulo-3 în ambele sensuri – diagramă stare

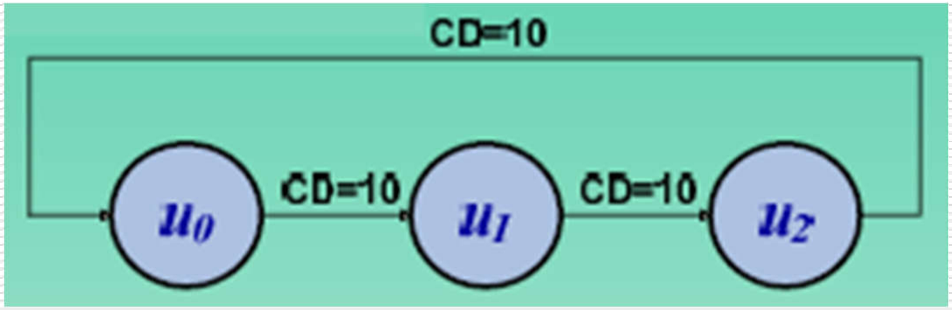
- Realizați diagrama de stare pentru un numărător care numără în ambele sensuri modulo 3. Numărătorul are 2 intrări: C – count enable care validează numărarea (pt. C=1), și D(direcție) care stabilește direcția de numărare D = 0 numără crescător, D=1 numără descrescător. Numărătorul are o ieșire Y care se setează pe 1 când numărătorul ajunge la 2 în situația în care numără crescător, sau la 0 pentru situația în care numără descrescător.

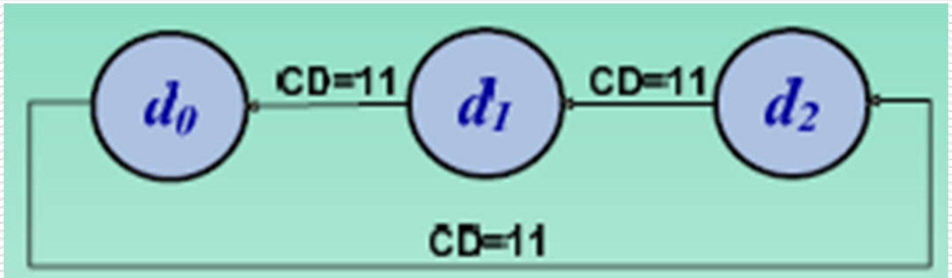


Counter symbol

Numărător modulo-3

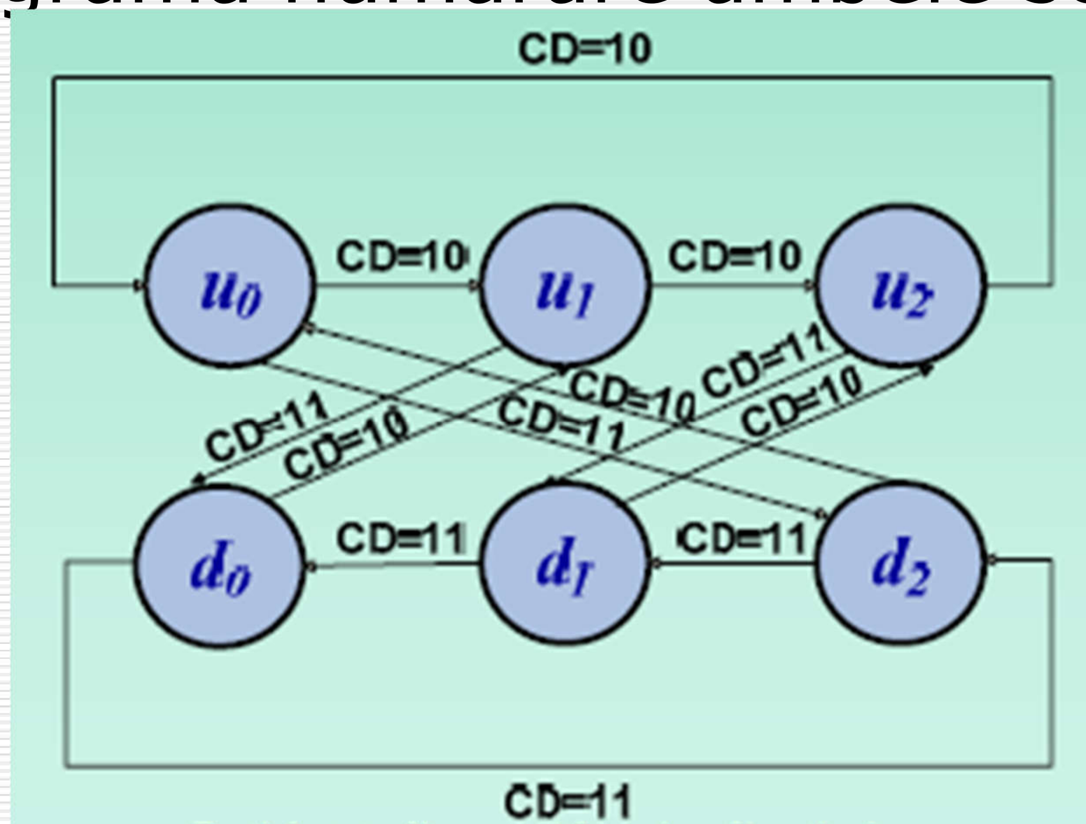
- Diagrame parțiale: secvență crescătoare/descrescătoare





Numărător modulo-3 ambele sensuri

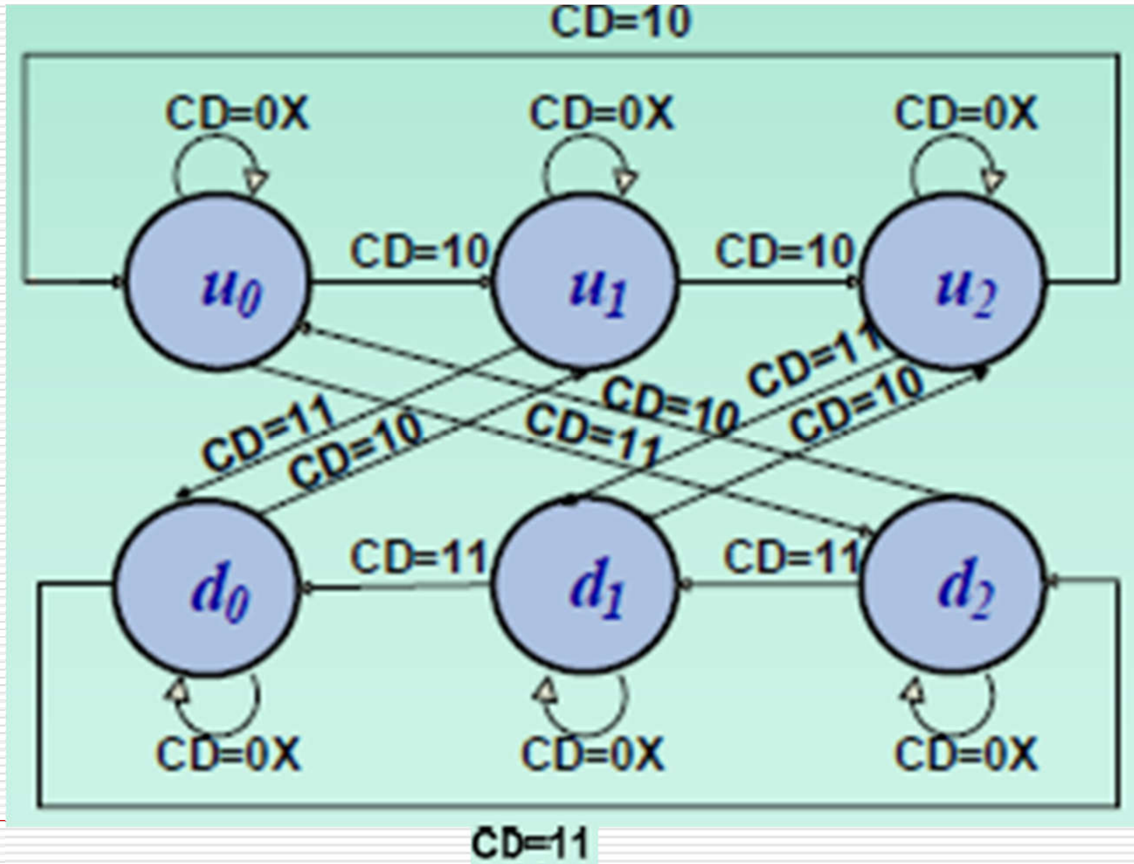
- Diagramă numărare ambele sensuri



Slides by Philip Pham, University of California, Irvine

Numărător modulo-3 ambele sensuri

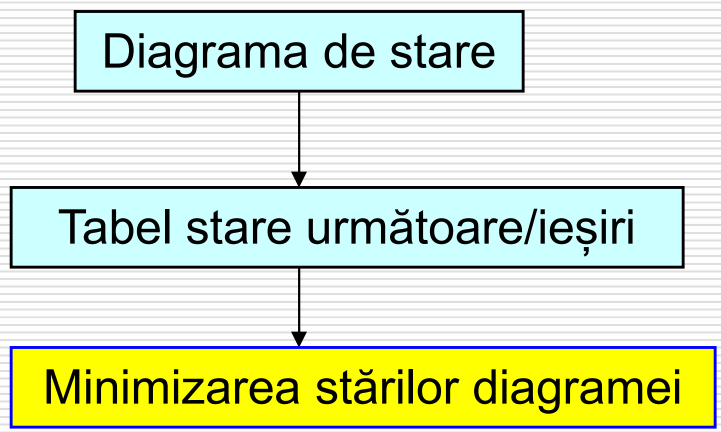
- Diagramă care ține cont si de semnal C inactiv



 Slides by Philip Pham, University of California, Irvine

Numărător modulo-3 ambele sensuri

- Diagrama completă nu prezintă numărul minim de stări



Minimizarea stărilor

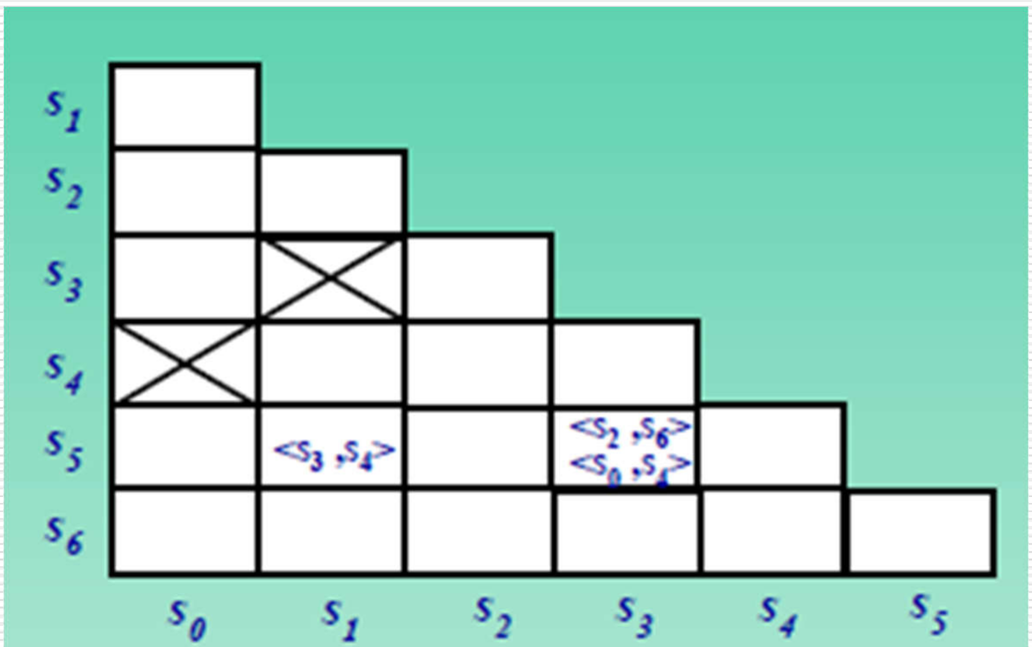
- Minimizarea stărilor își propune reducerea acestora → utilizarea unui număr mai mic de FF-uri
- Se bazează pe conceptul de comportament echivalent; două stări ale unui FSM au comportament echivalent dacă:
  - Au aceași secventă de valori de ieșire pentru aceași secventă de vectori de intrare
- Două stări s_j și s_k sunt evident echivalente ($s_j \equiv s_k$) dacă și numai dacă:
  - (1) au comportament echivalent: $\boldsymbol{h(s_j, i) = h(s_k, i)}$
  - (2) au aceleași stări următoare pt. toate secvențele de intrare $\boldsymbol{f(s_j, i) = f(s_k, i)}$

Minimizarea stărilor

- Două stări s_j și s_k sunt echivalente ($s_j \equiv s_k$) dacă și numai dacă:
  - (1) au comportament echivalent: ***h (s_j ,i) = h (s_k ,i)***
  - (2) au stări următoare diferite, dar acestea sunt echivalente
- Două automate A1 şi A2 sunt echivalente dacă pentru fiecare stare s_j din A2 există o stare echivalentă s_k în A1 şi invers pentru fiecare stare s_j din A1 există o stare echivalentă s_k în A2
- Echivalenţa stărilor unui automat complet definit împarte mulţimea stărilor acestuia în clase de echivalenţă disjuncte.
- Relaţia de echivalenţă a stărilor automatului complet definit are **proprietatea de tranzitivitate:** dacă şi atunci.

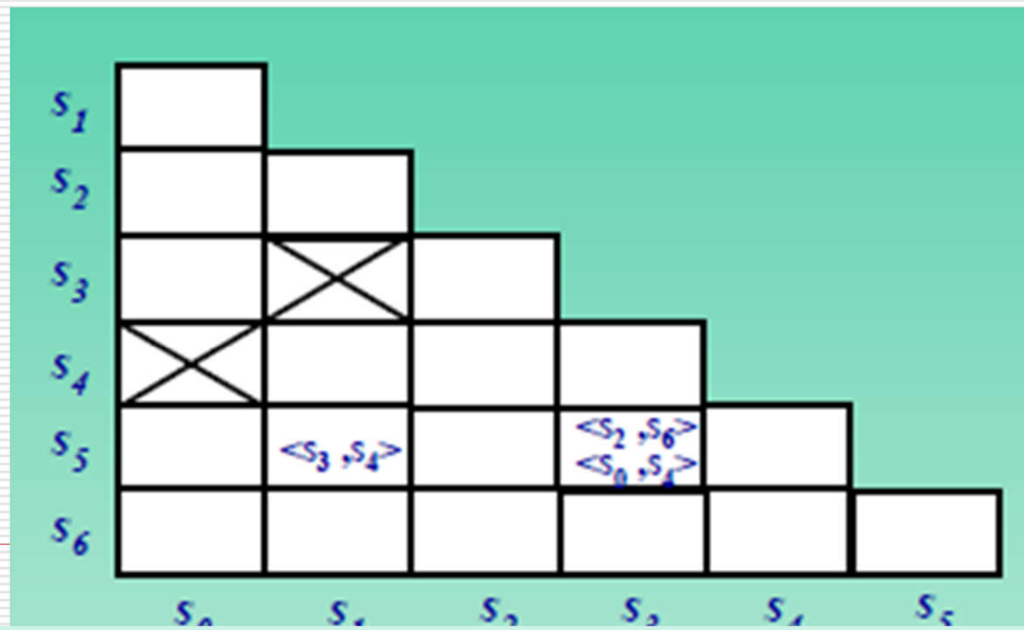
Metoda tabelului implicațiilor (Algoritmul PAULL-UNGER)

- tabel de formă triunghiulară având început de linii, stările automatului fără prima stare şi început de coloane stările automatului fără ultima stare.



Metoda tabelului implicaţiilor

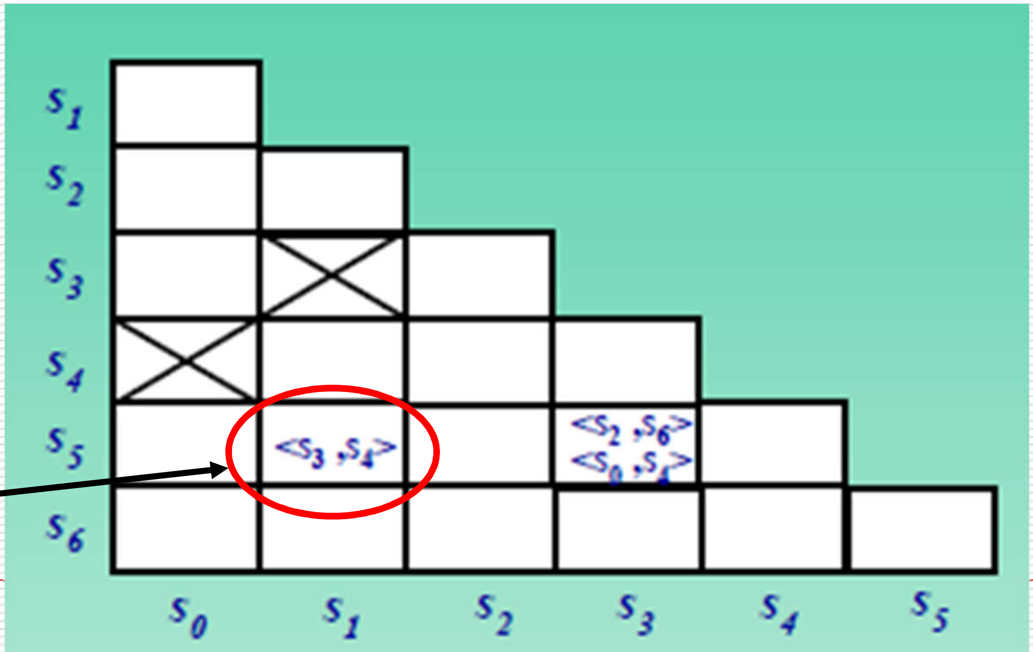
- la intersecţia unei linii cu o coloană :
  - X dacă stările din perechea respectivă sunt **evident neechivalente** (pentru aceeaşi intrare au ieşiri diferite)
  - **implicaţiile** privind echivalenţa succesorilor dacă stările din perechea respectivă au aceleaşi ieşiri pentru aceeaşi intrare (sunt 1 echivalente), dar succesori diferiţi



Metoda tabelului implicațiilor

- Se parcurge tabelul de jos în sus și de la stânga la dreapta și se introduce X în momentul identificării cel puțin unei perechi diferite de stări de ieșire
- Se reiterează până când în cadrul iterației curente nu se mai completează nici un X.

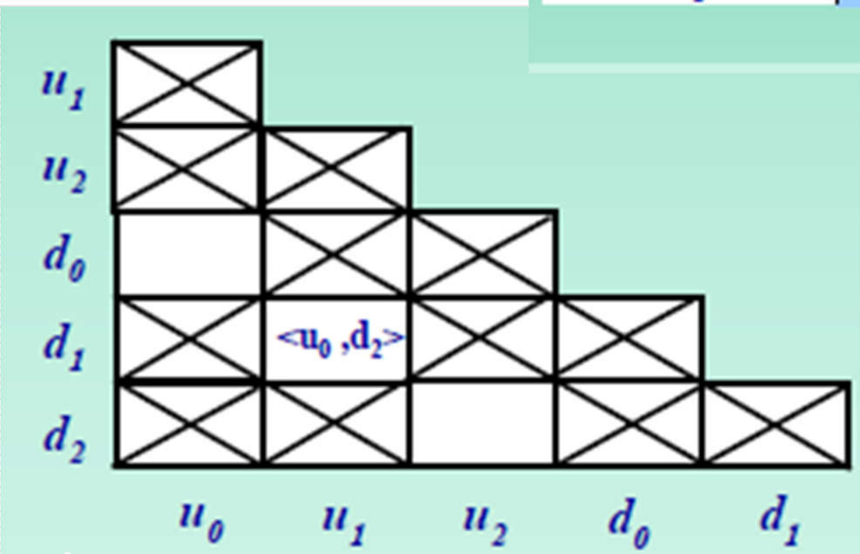
$s_1 \equiv s_5$ dacă
$s_3 \equiv s_4$



Metoda tabelului implicațiilor
Modulo 3-counter

PRESENT STATE	NEXT STATE		
	CD=0X	CD=01	CD=11
u_0	$u_0 / 0$	$u_1 / 0$	$d_2 / 1$
u_1	$u_1 / 0$	$u_2 / 0$	$d_0 / 0$
u_2	$u_2 / 0$	$u_0 / 1$	$d_1 / 0$
d_0	$d_0 / 0$	$u_1 / 0$	$d_2 / 1$
d_1	$d_1 / 0$	$u_2 / 0$	$d_0 / 0$
d_2	$d_2 / 0$	$u_0 / 1$	$d_1 / 0$

Next-state and output table

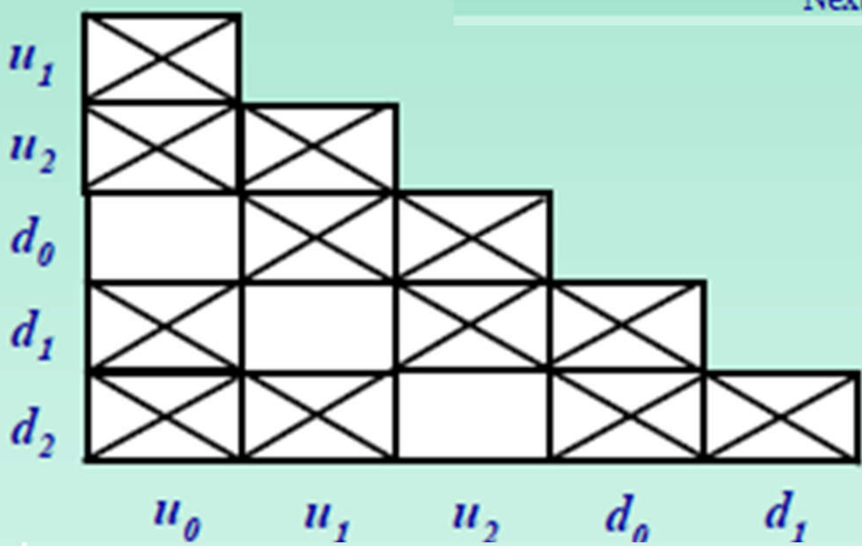


Se scriu clasele de echivalență:

Metoda tabelului implicațiilor
Modulo 3-counter

PRESENT STATE	NEXT STATE		
	CD=0X	CD=01	CD=11
u_0	$u_0/0$	$u_1/0$	$d_2/1$
u_1	$u_1/0$	$u_2/0$	$d_0/0$
u_2	$u_2/0$	$u_0/1$	$d_1/0$
d_0	$d_0/0$	$u_1/0$	$d_2/1$
d_1	$d_1/0$	$u_2/0$	$d_0/0$
d_2	$d_2/0$	$u_0/1$	$d_1/0$

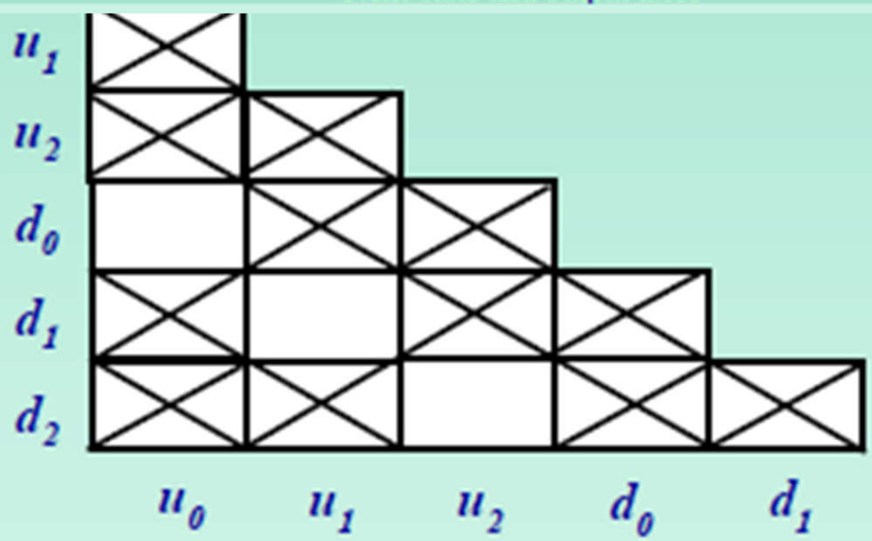
Next-state and output table



Metoda tabelului implicațiilor
Modulo 3-counter

PRESENT STATE	NEXT STATE		
	CD=0X	CD=01	CD=11
u_0	$u_0/0$	$u_1/0$	$d_2/1$
u_1	$u_1/0$	$u_2/0$	$d_0/0$
u_2	$u_2/0$	$u_0/1$	$d_1/0$
d_0	$d_0/0$	$u_1/0$	$d_2/1$
d_1	$d_1/0$	$u_2/0$	$d_0/0$
d_2	$d_2/0$	$u_0/1$	$d_1/0$

Next-state and output table



Se scriu clasele de echivalență:

s_0: <u_0, d_0>

s_1: <u_1, d_1>

s_2: <u_2, d_2>

Exemplul 2: Reducerea numărului stărilor automatelor finite

Să se determine tabelul tranziţiilor automatului cu număr redus de stări, folosind metoda tabelului implicaţiilor, având dat iniţial automatul:

Stare prezentă	Intrări			
	1	2	3	4
1	1/0	3/0	4/0	6/1
2	4/1	2/0	5/0	7/1
3	2/0	5/1	3/0	4/1
4	2/1	4/0	5/0	8/1
5	5/1	2/0	4/0	7/1
6	4/0	2/1	5/0	6/1
7	1/0	6/1	7/1	4/0
8	1/0	6/1	7/1	4/0

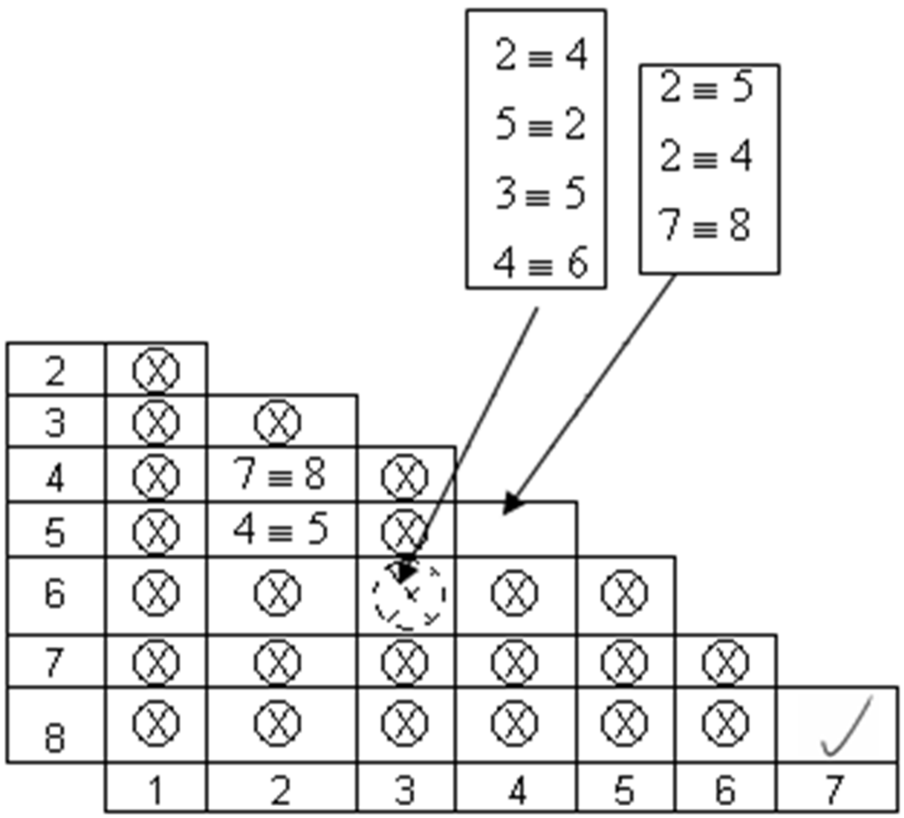
Exemplu 2 este preluat din cursul Logica Digitala 2013 – Adrian Mihăilescu

Exemplul 2: Reducerea numărului stărilor automatelor finite

Observaţie: în tabelul de mai sus s-au păstrat ca notaţii doar indicii stărilor şi intrărilor.

2	X						
3	X	X					
4	X	$7 \equiv 8$	X				
5	X	$4 \equiv 5$	X	$2 \equiv 5$ $2 \equiv 4$ $7 \equiv 8$			
6	X	X	$2 \equiv 4$ $5 \equiv 2$ $3 \equiv 5$ $4 \equiv 6$	X	X		
7	X	X	X	X	X	X	
8	X	X	X	X	X	X	✓
	1	2	3	4	5	6	7

Exemplul 2: Reducerea numărului stărilor automatelor finite



Exemplul 2: Reducerea numărului stărilor automatelor finite

Stări echivalente: $7 \equiv 8, 4 \equiv 5 \quad 2 \equiv 4 \quad 2 \equiv 5$.

Obţinem clasele echivalente: {1}, {2, 4, 5}, {3}, {6}, {7, 8}. Dacă înlocuim fiecare clasă de echivalenţă cu stările obţinem:

{1} $\subseteq$ Y1r

{2, 4, 5} $\subseteq$ Y2r

{3} $\subseteq$ Y3r

{6} $\subseteq$ Y4r

{7, 8} $\subseteq$ Y5r

Tabelul tranziţiilor automatului cu număr redus de stări devine:

Stare prezentă	Intrări			
	1	2	3	4
Y_{1r}	$Y_{1r}/0$	$Y_{3r}/0$	$Y_{2r}/0$	$Y_{4r}/1$
Y_{2r}	$Y_{2r}/1$	$Y_{2r}/0$	$Y_{2r}/0$	$Y_{5r}/1$
Y_{3r}	$Y_{2r}/0$	$Y_{2r}/1$	$Y_{3r}/0$	$Y_{2r}/1$
Y_{4r}	$Y_{2r}/0$	$Y_{2r}/1$	$Y_{2r}/0$	$Y_{4r}/1$
Y_{5r}	$Y_{1r}/0$	$Y_{4r}/1$	$Y_{5r}/1$	$Y_{2r}/0$

Diagrama de stare

↓

Tabel stare următoare/ieșiri

↓

Minimizarea stărilor diagramei

↓

Codificarea stărilot/intrărilor/ieșirilor

Codificarea stărilor

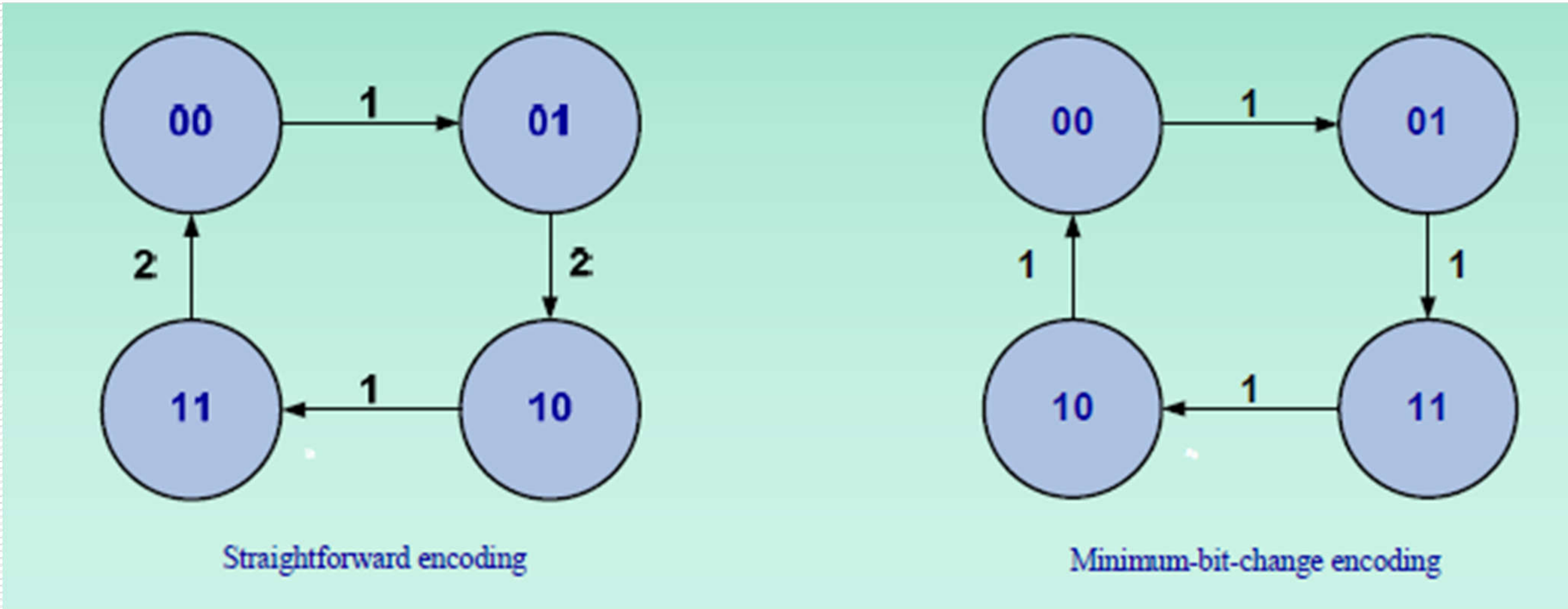
ENCODING NUMBER	s_0	s_1	s_2	s_3
1	00	01	10	11
2	00	01	11	10
3	00	10	01	11
4	00	10	11	01
5	00	11	01	10
6	00	11	10	01
7	01	00	10	11
8	01	00	11	10
9	01	10	00	11
10	01	10	11	00
11	01	11	00	10
12	01	11	10	00
13	10	00	01	11
14	10	00	11	01
15	10	01	00	11
16	10	01	11	00
17	10	11	00	01
18	10	11	01	00
19	11	00	01	10
20	11	00	10	01
21	11	01	00	10
22	11	01	10	00
23	11	10	00	01
24	11	10	01	00

- Costul/întârzierea unei implementări FSM depind de codificarea stărilor;
- 4! Posibilități de codificare pentru 4 stări.
- 3 euristici:
  - **Număr minim de tranziții**
  - **Adiacență pe bază de priorități**
  - **one-hot**

Codificarea stărilor

- **Număr minim de tranziții (minimum bit change):**
  - Stările sunt astfel codificate încât să fie minimizate tranzițiile bițiilor registrului de stare;
  - Fiecărui arc i se alocă un cost egal cu numărul de biți care diferă la tranziția dintre stări în registrul de stare;
  - Se minimizează suma costurilor la parcurgerea grafului
- **Adiacență pe bază de priorități**
- **one-hot**

Număr minim de tranziții



Codificarea stărilor

- **Număr minim de tranziții**
- **Adiacență pe bază de priorități (Prioritized adjancency strategy):**
  - **Codificări adiacente pentru stările:**
    - **Destinație comună**
    - **Sursă comună**
    - **Ieșire comună**
  - **Next state va apărea căsuțe adiacente în K-map;**
- **one-hot**

Codificarea stărilor

- **Număr minim de tranziții**
- **Adiacență pe bază de priorități**
- **one-hot**
  - Fiecărei stări i se alocă un bit din registrul de stare → registrul de stare are atâția biți câte stări sunt în diagramă
  - Nu se pretează pentru diagrame cu multe stări;
  - La un moment dat un singur bit (cel corespunzător stării este pe 1)

Întrebări?

Enough Talking Let's Get To It
!!Brace Yourselves!!

Imagini preluate din ECE 448 – FPGA and ASIC Design with VHDL