AUTOMATE FINITE

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 VHDL pt 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

Data trecuta

- Proiectarea unui sistem numeric complex
  - Unitatea de executie
  - Unitatea de control/comanda
- Exemplu – cuptor simplu
  - INTREBARI???
- Incepand de sapt aceasta – abstractizarea metodelor/modelelor de realizare a (controlului) sistemelor
  - Automate finite

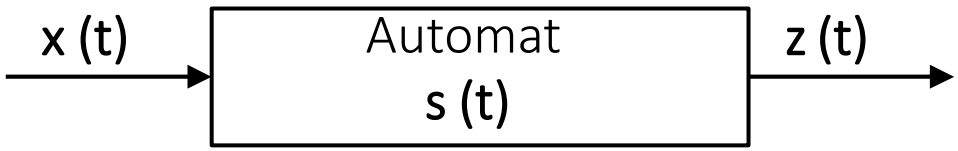
INTRODUCERE

- Teoria automatelor realizează studiul posibilităţilor şi limitărilor dispozitivelor de prelucrare a informaţiei, folosind un model abstract al tuturor dispozitivelor, model care imită activitatea şi aspectul funcţional al acestora
- Modelul abstract se numeşte AUTOMAT (maşină de stare) – state machine (FSM)
- **Definiţie:** Un automat este un *sistem dinamic* a cărui comportare se poate descrie ca o succesiune de evenimente numite *stări*, ce apar la *momente discrete ale variabilei timp*

INTRODUCERE

Automat finit

- Ca termen, este un concept abstract (obiect logico-matematic realizat sau realizabil), cu derivaţie directă din conceptul de **sistem dinamic cu reacţie negativă**
- Se reprezintă ca o cutie neagră:



- Automat finit - dacă mulţimea stărilor interne s(t) este finită
- Automatul finit interacţionează cu mediul:
  - La un anumit moment de timp t este supus unui semnal de intrare x(t)
  - La momentul t+dt oferă ca răspuns la ieşire semnalul z(t)

INTRODUCERE

Automat finit

- Semnalele de intrare şi de ieşire sunt de regulă succesiuni de valori binare (0 sau 1)
- Aplicarea intrărilor şi succesiunea ieşirilor se face în ordine secvenţială → se justifică şi denumirea de circuit logic secvenţial (CLS)
- Automatele finite sunt o reprezentare abstractă pentru circuitele logice secvenţiale

INTRODUCERE

Automat infinit

- Mulţimea stărilor şi mulţimea variabilelor de intrare şi de ieşire sunt infinite
- Nu există un automat infinit realizabil, dar trebuie luat ca limită spre care tind maşinile de calcul moderne
- În practică, faptul că maşinile de calcul sunt finite se datorează fie limitării în timp a funcţionării, fie limitării în complexitate a structurii sau în lungime a secvenţei de instrucţiuni
- Pentru a se putea studia un automat infinit se foloseşte un compromis: se studiază acele maşini care au la un moment dat o structură finită, structură pe care o pot extinde nedefinit în timp

INTRODUCERE

Definiţie

- **Automat finit** - cvintuplu **A = { X, Z, S, f, g }**
  - **X** , **Z** şi **S** sunt mulţimi finite nevide
  - **f** şi **g** sunt funcţii definite pe aceste mulţimi
- **X = { x_1, x_2, … x_n }** - mulţimea variabilelor de intrare
- **Z = { z_1, z_2, … z_n }** - mulţimea variabilelor de ieşire
- **S = { s_1, s_2, … s_n }** - mulţimea stărilor automatului (s_i este o stare)

INTRODUCERE

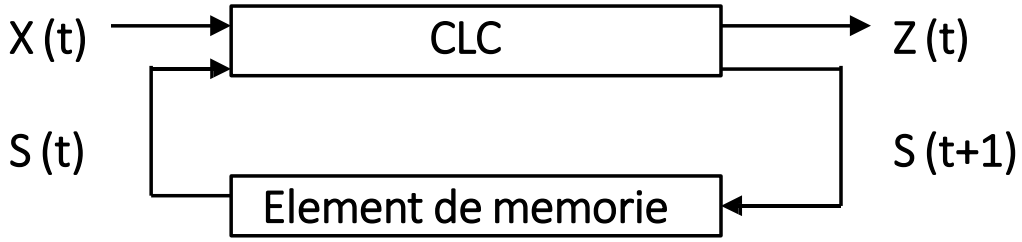
Definiţie

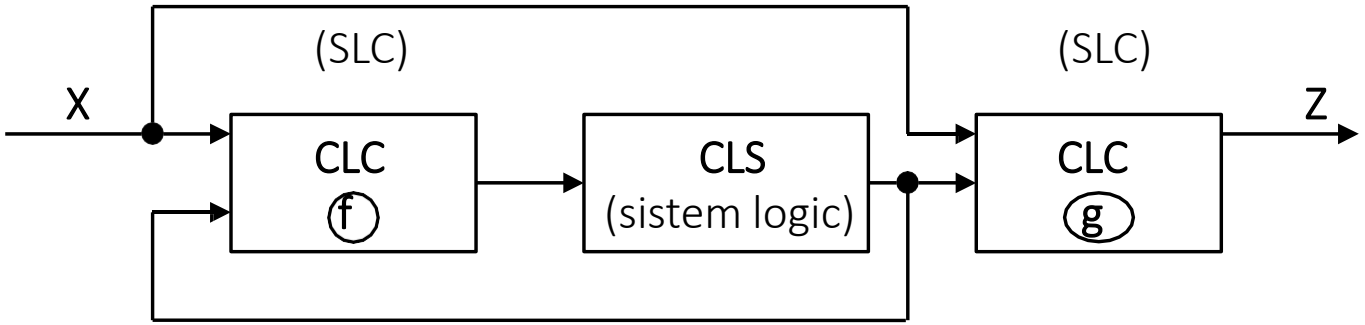
- Funcţia de tranziţie $f : S * X \rightarrow S$
  - Transformă mulţimea tuturor perechilor ordonate (s_i, x_j) în mulţimea S
  - Are rolul de a preciza starea în care ajunge automatul în urma aplicării unei intrări (dă starea următoare funcţie de intrare şi de starea prezentă)
- Funcţia de ieşire $g : S * X \rightarrow Z$
  - Transformă mulţimea tuturor perechilor ordonate (s_i, x_j) în mulţimea Z
  - Are rolul de a preciza ieşirea automatului în urma aplicării unei intrări

INTRODUCERE

Schema bloc

- 2 variante





INTRODUCERE

Definiţii

- Un automat finit se numeşte **iniţial** dacă în mulţimea stărilor S există o singură stare s_0 ca stare iniţială (din care se pune în funcţiune automatul); automatul este **slab iniţial** dacă există mai multe stări iniţiale
- Se numeşte **evoluţie** a unui automat o succesiune de stări $\{ s_1, s_2, \ldots s_n \}$
- Automatul **determinist** are **funcţii unice** de tranziţie **f**, respectiv de ieşire **g**, oricare ar fi $x \in X$ şi $s \in S$

INTRODUCERE

Definiţii

- Automatul **autonom** are evoluţia independentă de intrări
- Automatul este **combinaţional** dacă starea internă nu influenţează ieşirea (ieşirile sunt complet determinate în orice moment de timp numai de intrări)
- Automatul este **complet definit** (specificat) dacă sistemul de funcţii f şi g este definit pentru toate perechile (s_i, x_j)
- Automatul este **conex** dacă orice stare este accesibilă din orice altă stare

INTRODUCERE

Analiza automatelor

- Descriptivă - dacă automatele sunt cutii negre
  - Se vor trata problemele de *echivalare* a automatelor din punctul de vedere al *comportării exterioare* şi al *reducerii numărului de stări*
- Constructivă
  - Se face analiza în vederea **realizării** unor *automate echivalente* cu unul dat, pe baza unor cerinţe sau criterii impuse

INTRODUCERE

Utilizarea automatelor - roluri

- Traductor – dispozitiv care transformă intrările în ieşiri
  - Obiectiv – generarea corecta a iesirilor
  - Ex: divizor de frecventa/porti logice
- Acceptor – dispozitiv la care se analizează secvenţa de intrare care trebuie aplicată automatului în vederea atingerii unor stări finale
  - Obiectiv – generarea starilor
  - Ex: numarator

REPREZENTAREA AUTOMATELOR

- Evoluţia automatelor în timp este dată de funcţiile caracteristice:
  - Funcţia de tranziţie **f**
  - Funcţia de ieşire **g**
- Reprezentarea unui automat scoate în evidenţă intrările, stările şi ieşirile, cu precizarea funcţiilor de tranziţie şi de ieşire **f** şi **g**

REPREZENTAREA AUTOMATELOR

Tabel de tranziţie

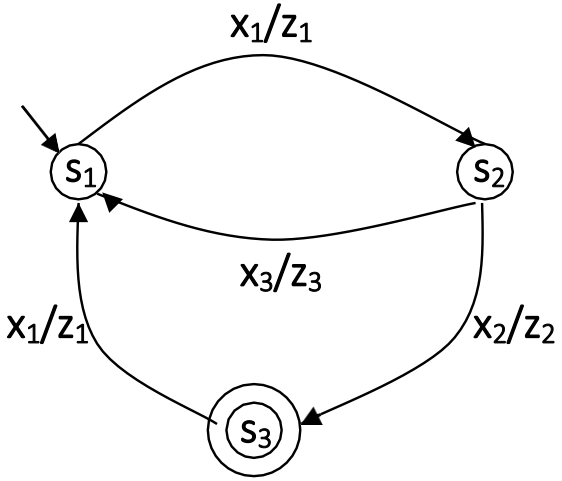
- Se reprezintă pe coloane variabilele de intrare şi pe linii stările
- În intersecţiile rezultate se marchează starea următoare şi ieşirea următoare
  - De exemplu, din s_1, pentru intrarea x_1 se trece în starea s_i cu ieşirea z_j

S \ X	x_1	...	x_n
s_1	s_i/z_j	...	
. . .		f, g	
s_q			

REPREZENTAREA AUTOMATELOR

Graf de tranziţie

- Fiecărei stări a sistemului îi corespunde un nod al grafului
- Fiecărei tranziţii de la starea s_i la starea s_j îi corespunde un arc orientat de la s_i la s_j care uneşte cele 2 noduri
- Pe fiecare arc se notează intrarea care determină tranziţia şi ieşirea următoare a automatului
  - Starea iniţială s_1 se marchează cu o săgeată
  - Starea finală s_3 se marchează cu un cercconcentric



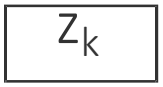
REPREZENTAREA AUTOMATELOR

Organigramă

- Elementele constitutive
  - Stare

  - Intrare

  - Ieşire



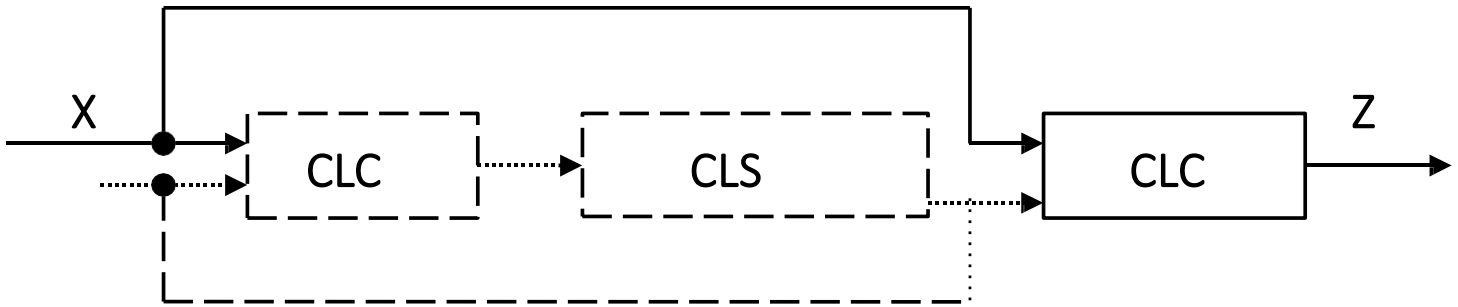
Cronogramă

- Forme de undă = diagrame de timp – arată evoluţia în timp a automatului

CLASIFICAREA AUTOMATELOR

Gradul de complexitate şi modul de realizare a funcţiilor caracteristice

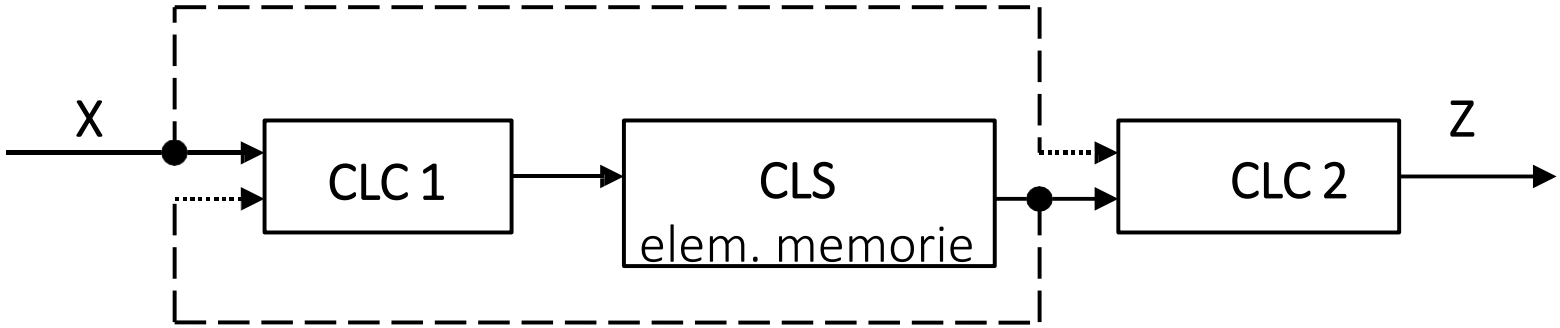
- 1) Automat de gradul 0 (automat combinaţional)
  - Caracterizat de absenţa variabilelor de stare
  - Funcţia de ieşire **g** - determinată numai de variabilele de intrare
  - Poate fi descris şi prin diagrame Karnaugh, funcţii de adevăr, ecuaţii etc.



CLASIFICAREA AUTOMATELOR

Gradul de complexitate şi modul de realizare a funcţiilor caracteristice

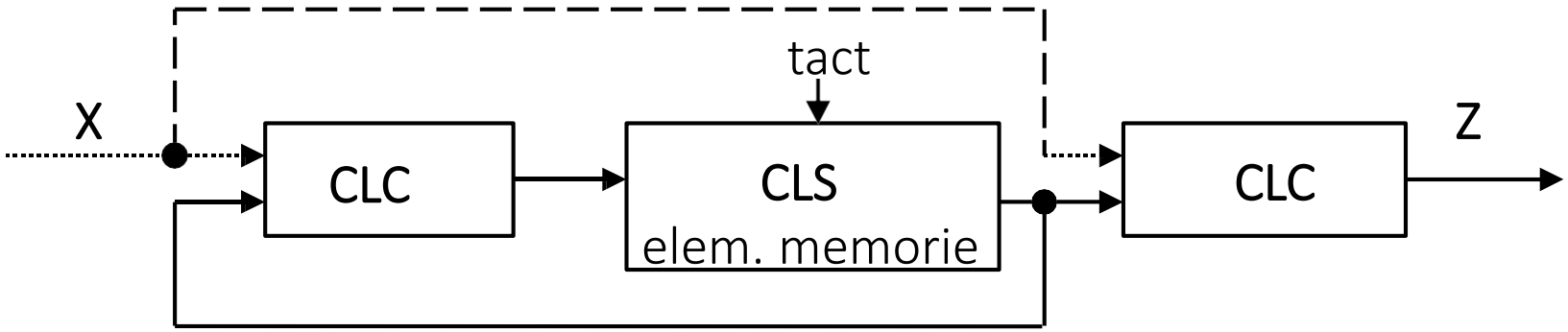
- 2) Automat de gradul 1
  - Apar şi elemente de memorie
  - Starea este generată de elementele de memorie, fără bucle de reacţie inversă
  - Funcţia de tranziţie **f** nu depinde de starea anterioară
  - Poate fi interpretat ca un CLC cu o unitate de întârziere



CLASIFICAREA AUTOMATELOR

Gradul de complexitate şi modul de realizare a funcţiilor caracteristice

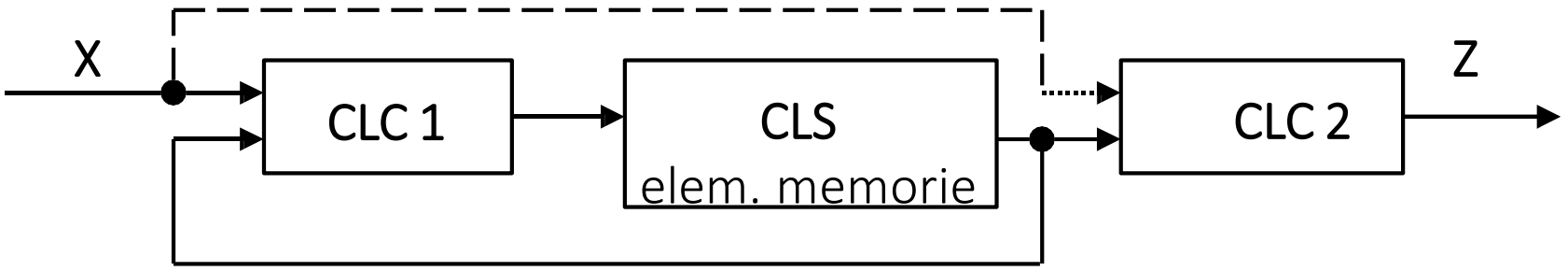
- 3) Automat de gradul 2
  - Introduce conceptul de succesiune de stări
  - Conţine bucle de reacţie
  - Nu are variabile de intrare, deci evoluţia nu depinde de variabilele de intrare
  - Trecerea dintr-o stare în alta se face în prezenţa unui semnal de ceas
  - Se numesc autonome
  - Exemple: generatoare de tact, divizoare de frecvenţă
  - Dacă mulţimea stărilor este identică cu mulţimea ieşirilor Z=S avem cazul bistabilelor şi a numărătoarelor



CLASIFICAREA AUTOMATELOR

Gradul de complexitate şi modul de realizare a funcţiilor caracteristice

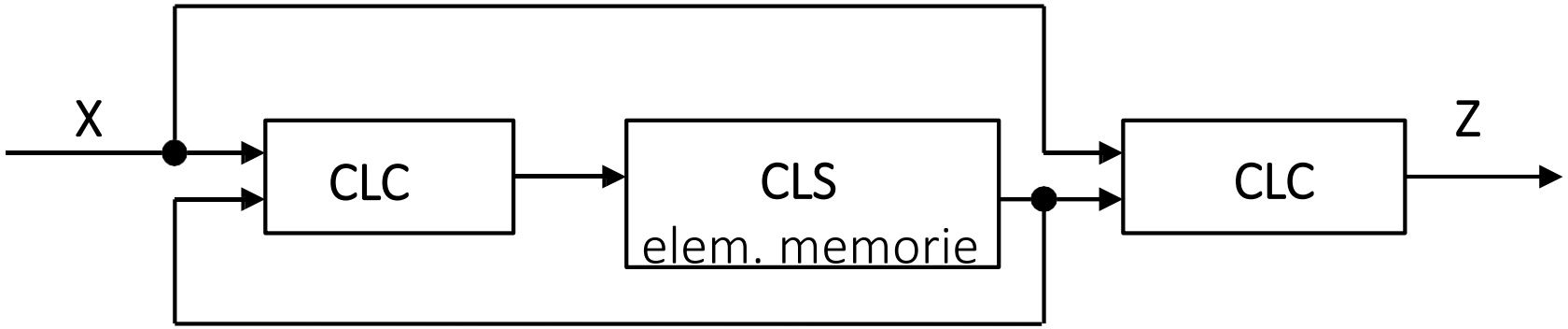
- 4) Automat de gradul 3 (Moore)
  - Funcţia de tranziţie **f** se obţine din variabilele de stare şi variabilele de intrare
  - Funcţia de ieşire se obţine numai din variabilele de stare **Z = g(s)**
  - Are două părţi de prelucrare combinaţională şi o parte de memorie care identifică starea actuală şi comportarea viitoare
  - Partea de memorie poate fi realizată cu memorii efective (bistabile, RAM, ROM) sau este un sistem logic cu funcţia de memorare obţinută prin buclă de reacţie proprie



CLASIFICAREA AUTOMATELOR

Gradul de complexitate şi modul de realizare a funcţiilor caracteristice

- 5) Automat de gradul 4 (Mealy)
  - Forma cea mai complexă
  - Funcţiile de tranziţie **f** şi de ieşire **g** sunt definite atât pe baza stării actuale cât şi pe baza variabilelor de intrare



- **Observaţie**: clasificarea nu ţine cont de restricţiile care apar referitor la tranziţiile variabilelor care intervin în evoluţia automatului

CLASIFICAREA AUTOMATELOR

Modul în care se efectuează tranziţiile variabilelor

- Automate sincrone
  - Tranziţiile se produc la momente de timp bine precizate, indicate printr-un semnal de tact (ceas) extern
  - Toate cele 5 tipuri de automate prezentate la clasificarea anterioară pot fi sincrone

CLASIFICAREA AUTOMATELOR

Modul în care se efectuează tranziţiile variabilelor

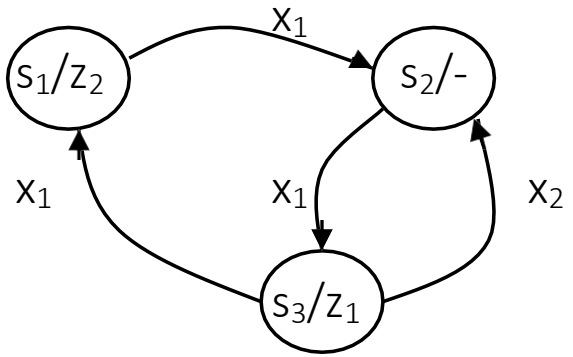
- Automate asincrone
  - Tranziţiile, modificările variabilelor de intrare, stare şi ieşire se fac la momente arbitrare de timp
  - Automatul de tip 2 nu poate fi asincron
    - Depinde direct de ceas

TRANSFORMAREA AUTOMATELOR

Transformare Moore → Mealy

- Metoda de transformare a automatelor complexe de tip Moore şi Mealy se bazează pe grafurile de tranziţie pentru cele două tipuri de automate
- Automatul Moore are notate în nodurile grafului stările (simbolic sau codificate binar) şi ieşirile corespunzătoare, iar pe arce intrările care produc tranziţia respectivă

S \ X	x_1	x_2	Z
s_1	s_2	-	z_2
s_2	s_3	-	-
s_3	s_1	s_2	z_1

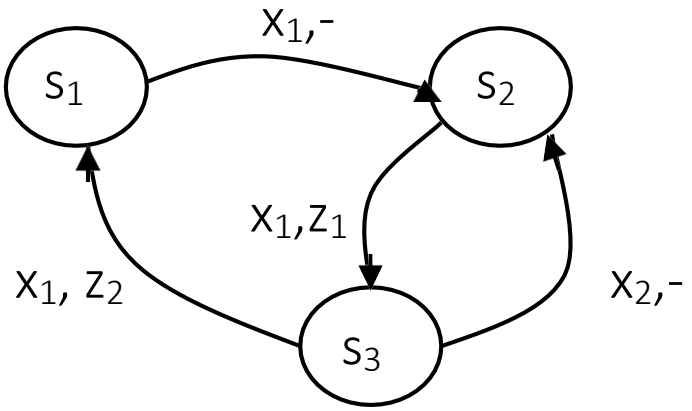


TRANSFORMAREA AUTOMATELOR

Transformare Moore → Mealy

- **Automatul Mealy** are înscrise în nodurile grafului stările, iar pe arce intrările care produc tranziţia şi ieşirile obţinute în cadrul tranziţiei respective

S \ X	x_1	x_2
s_1	s_2, -	-
s_2	s_3, z_1	-
s_3	s_1, z_2	s_2, -



TRANSFORMAREA AUTOMATELOR

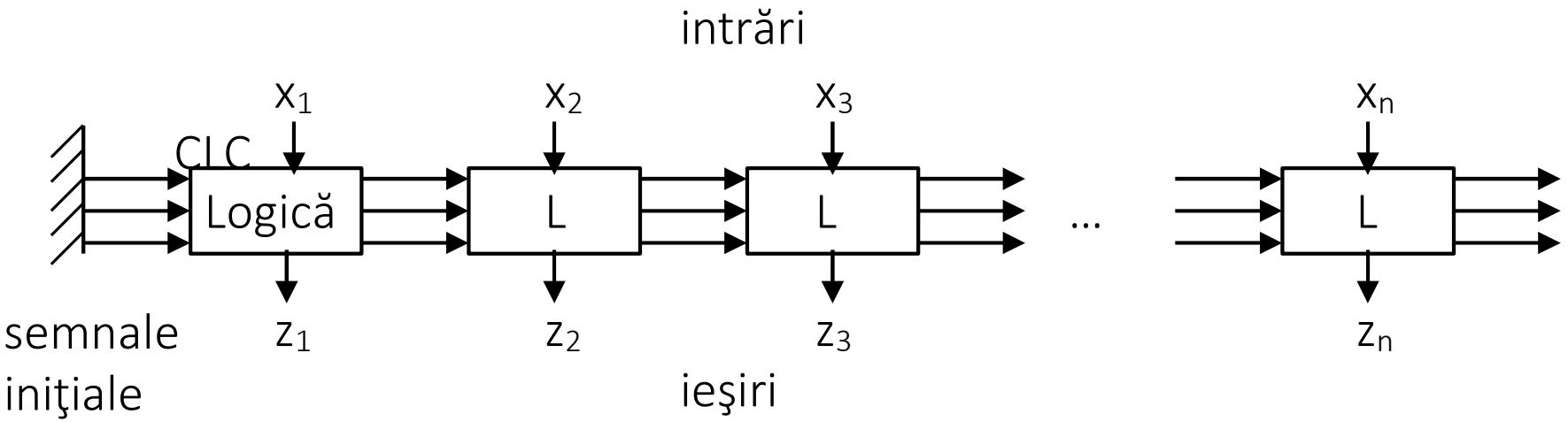
Transformare Moore → Mealy

- Condiţia care se impune este ca automatul Mealy obţinut în urma transformării automatului Moore să fie echivalent, adică să producă pentru orice succesiune de intrări aceeaşi succesiune de ieşiri ca şi automatul Moore iniţial
- Transformarea din automat Moore în automat Mealy presupune scoaterea în graf a ieşirii corespunzătoare unei anumite stări pe arcele care conduc la starea respectivă
- Dacă cele două automate au **aceeaşi comportare** pentru **orice secvenţă de intrare**, atunci sunt echivalente

REŢELE ITERATIVE

Definiţie

- Reţelele iterative sunt o structură de **reţea de componente combinaţionale identice** (toate blocurile sunt identice)



REŢELE ITERATIVE

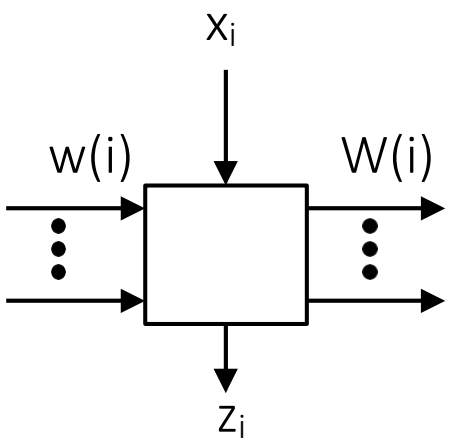
- **Semnalele iniţiale (de limită sau graniţă)** determină starea iniţială a reţelei
- **Semnalele de intrare** $x_1, x_2, \ldots x_n$ se aplică independent unul de celălalt, fiecare câte unei celule
- **Semnalele de ieşire** $z_1, z_2, \ldots z_n$ sunt produse de câte o celulă
- **Semnalele intercelule** sunt transmise doar celulelor adiacente

REŢELE ITERATIVE

Reţea iterativă unilaterală

- Semnalele dintre celule se transmit doar într-o singură direcţie
- Reprezentarea unei celule:

semnale intercelule (de intrare/ieşire)



celula i
W(i) = w(i+1)

- Considerăm un model ideal - semnalele se pot propaga prin reţea fără întârziere
- La modelul ideal noţiunea de timp nu e relevantă

REŢELE ITERATIVE

Reţea iterativă unilaterală

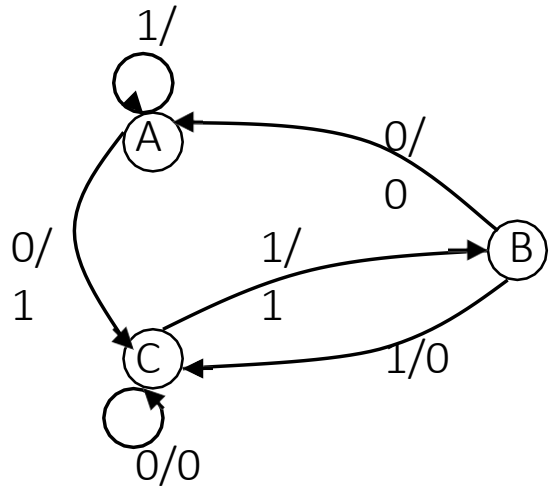
- Există o analogie apropiată între comportarea reţelelor iterative şi CLS-uri (automate)
- Semnalele de intrare aplicate simultan celulelor corespund secvenţei de intrări aplicate CLS-urilor
- Semnalele transmise fără întârziere de la stânga la dreapta între celulele adiacente corespund stărilor interne ale CLS-urilor
  - w(i) sunt stări interne la momentul t=i
- Semnalele limită corespund stării iniţiale

REŢELE ITERATIVE

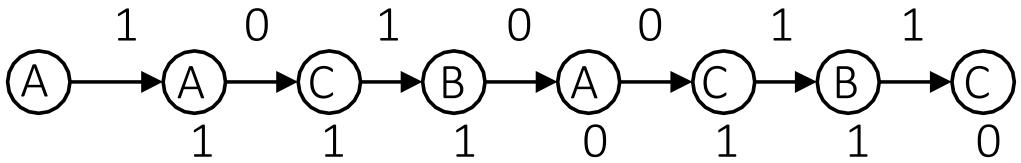
Exemplu

- Avem un automat descris de tabelul şi graful de tranziţie:

S \ X	0	1
A	C,1	A,1
B	A,0	C,0
C	C,0	B,1



- Aplicăm automatului o secvenţă de intrări: 1010011



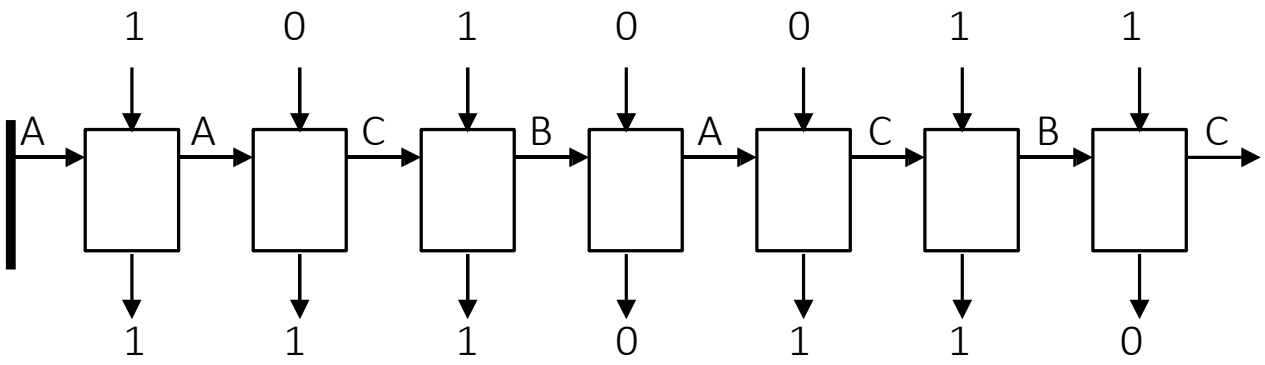
- Obţinem secvenţa de ieşiri generate: 1110110

REŢELE ITERATIVE

Exemplu

- Construim reţeaua iterativă analogă
- Descriem comportarea reţelei prin tabelul de adevăr al unei celule oarecare (trebuie să fie identic cu tabelul de tranziţii al automatului)

w(i) \ x_i	0	1
A	C,1	A,1
B	A,0	C,0
C	C,0	B,1



REŢELE ITERATIVE

Exemplu

- Aplicând simultan succesiunea variabilelor de intrare obţinem o funcţionare identică cu a automatului
- Pentru obţinerea fizică a reţelei iterative semnalele intercelulare trebuie analizate ca nivele logice nu ca şi impulsuri
- O celulă se construieşte pe baza tabelului ei de adevăr

REŢELE ITERATIVE

Diferenţe între reţele iterative şi automate

- Din cauza lungimii secvenţei de intrare, reţelele iterative pot fi realizate pentru un număr finit şi relativ mic de celule
- Costul unei reţele iterative este proporţional cu lungimea reţelei şi mult mai mare decât a unui automat
- Reţeaua iterativă poate fi şi bidirecţională (conectare de la dreapta la stânga); la un CLS asta ar însemna că starea prezentă depinde de starea viitoare, deci ar trebui să fie bidimensional în timp!!!
- Reţelele iterative au avantajul că nu necesită elemente de memorie sau sincronizări externe, deci nu apar hazarduri sau curse critice

CONCLUZII

- Automate finite
  - Definitie, tipuri
  - Functii de tranzitie, functii de iesire
  - Reprezentarea automatelor
- Clasificarea automatelor
  - Grad 0, 1, 2, 3, 4
  - Moore vs Mealy
  - Transformarea automatelor
- Retele iterative
  - Retea de componente combinationale
  - Alternativa la automate
  - Avantaje si dezavantaje wrt automate