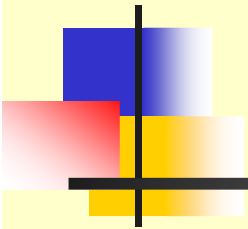
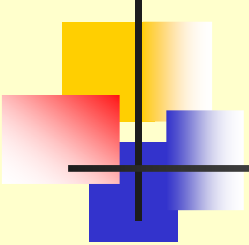


# **SINTEZA CIRCUITELOR LOGICE SECVENTIALE SINCRONE**



## 5.5. Sinteza circuitelor logice secvențiale sincrone

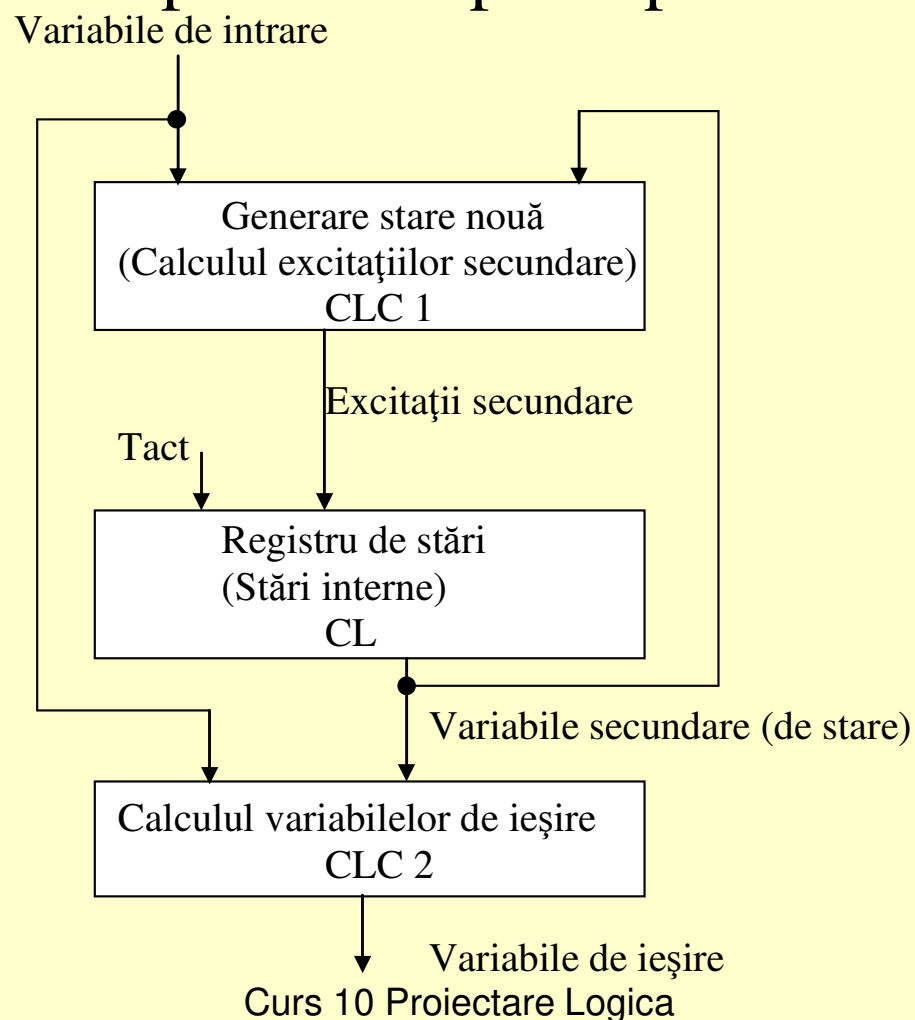


---

- Circuitele logice secvențiale **sincrone** trec dintr-o stare în alta la momente distincte de timp, determinate de impulsurile de tact (CLK)
- Între două impulsuri de tact starea circuitului nu se modifică!

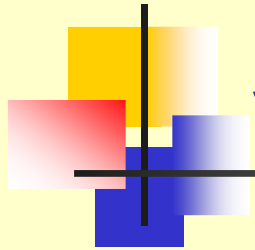
## 5.5. Sinteza circuitelor logice secvențiale sincrone

- **Schema bloc** cu componentele principale ale CLS sincrone



## 5.5. Sinteza circuitelor logice secvențiale sincrone

- **Variabilele de intrare** sunt în general sincrone cu impulsul de tact, dar pot fi și de tip asincron
- **CL** = circuit logic general care păstrează starea internă (variabilele de stare) → **registru de stări**
  - se poate implementa cu: bistabili RS, D, JK, T, registre, memorii; poate fi și un circuit logic secvențial cu buclă de reacție
- **CLC 1** = determină **funcțiile de excitație secundare** (condiționările intrărilor), care în prezența tactului determină trecerea circuitului în altă stare → se poate numi **generatorul stării noi**
  - se pot realiza cu porți logice sau cu circuite logice combinaționale specializate (multiplexoare, decodificatoare...)
- **CLC 2** = determină **valorile ieșirilor circuitului**
  - se pot realiza cu porți logice sau cu circuite logice combinaționale specializate (multiplexoare, decodificatoare...)



## 5.5.1 Etapele de sinteză

---

- 1. Expunerea condițiilor de funcționare (descrierea comportării circuitului)
- 2. Codificarea stărilor
- 3. Reducerea numărului de stări
- 4. Se decide modul de implementare a registrului de stări
- 5. Se determină funcțiile de excitație și funcțiile de ieșire
- 6. Se studiază problemele de hazard
- 7. Se desenează schema circuitului



## 5.5.1 Etapele de sinteză

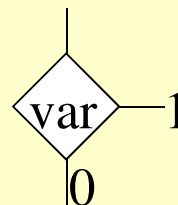
---

- **1. Expunerea condițiilor de funcționare** (descrierea comportării circuitului)
  - Stabilirea modalității de definire a circuitului care trebuie sintetizat prin:
    - tabel de tranziții
    - graf de tranziții
    - organigramă
    - forme de undă
  - Trebuie evidențiate:
    - stările prin care trece circuitul
    - valorile variabilelor de intrare pentru care se schimbă stările
    - valorile rezultate ale variabilelor de ieșire
  - Evoluția circuitului începe într-o stare inițială și de obicei se revine la această stare, după ultima stare a ciclului

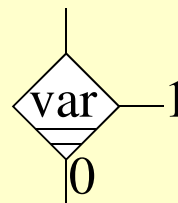
## 5.5.1 Etapele de sinteză

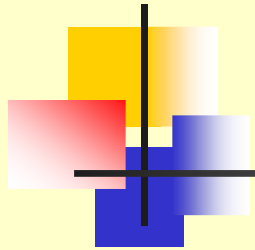
- Descrierea funcționării prin **organigramă**
  - Elementele componente ale organigramei de funcționare a oricărui circuit secvențial sincron:
    - 1. Elementul de intrare (control sau decizie)
      - Variabile de intrare

- Sincrone



- Asincrone

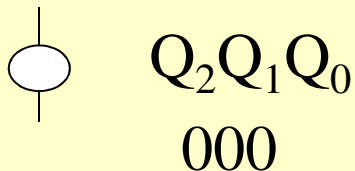




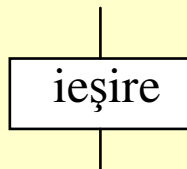
## 5.5.1 Etapele de sinteză

- Descrierea funcționării prin **organigramă**
  - Elementele componente ale organigramei de funcționare a oricărui circuit secvențial sincron:

- 2. Elementul de stare



- 3. Elementul de ieșire

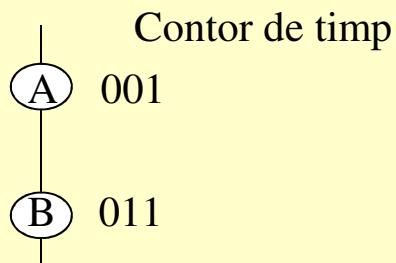




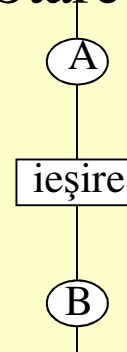
## 5.5.1 Etapele de sinteză

- Descrierea funcționării prin **organigramă**
  - Configurații elementare

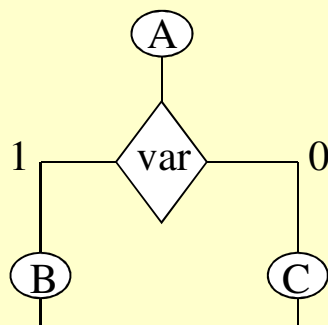
Tranziție simplă



Stare cu ieșire



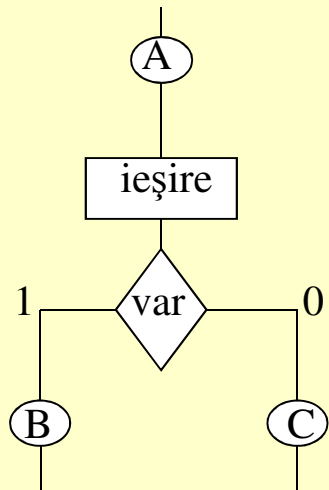
Stare cu decizie



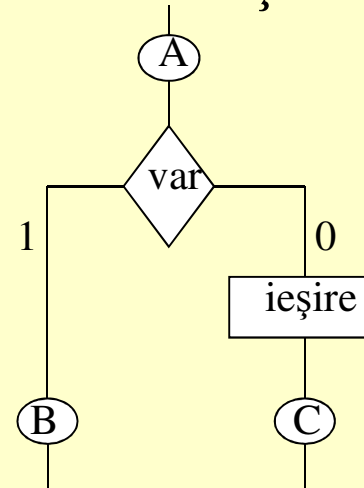
## 5.5.1 Etapele de sinteză

- Descrierea funcționării prin **organigramă**
  - Configurații elementare

Stare cu ieșire și decizie



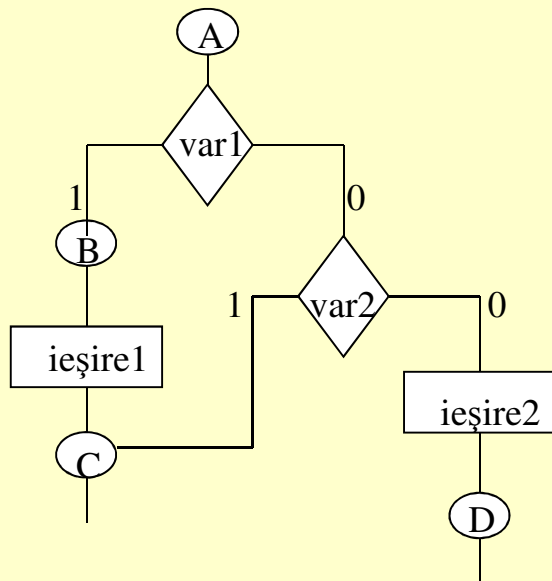
Stare cu decizie și ieșire condiționată



## 5.5.1 Etapele de sinteză

- Descrierea funcționării prin **organigramă**
  - Configurații elementare

Stări cu decizii multiple și ieșiri





## 5.5.1 Etapele de sinteză

---

### ■ 2. Codificarea stărilor - dificilă

#### ■ Funcționări defectuoase

##### ■ La tranziții greșite între stări

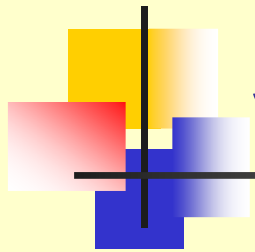
- Tranzițiile greșite între stări apar datorită prezenței **variabilelor de intrare asincrone**
  - Se elimină cel mai ușor prin sincronizarea variabilele de intrare cu semnalul de tact
- Codificarea stărilor se stabilește astfel încât, în orice stare, pentru toate combinațiile posibile de intrări asincrone, să nu fie mai mult decât o singură variabilă de stare dependentă de o variabilă de intrare asincronă  $\Rightarrow$  două stări rezultate din calea de ieșire a unei intrări asincrone vor avea codificare adiacentă

##### ■ Semnale greșite care apar la circuitul de generare a variabilelor de ieșire

- Ieșirile false pot să apară din cauză că la trecerea dintr-o stare în alta, **variabilele de stare practic nu se modifică simultan**
- Pentru evitarea tranzițiilor false ale ieșirilor se pot folosi metodele:
  - Se realizează o codificare adiacentă a stărilor
  - Se forțează trecerea circuitului prin stări suplimentare
  - Se sincronizează variabilele de ieșire

### ■ 3. Reducerea numărului de stări

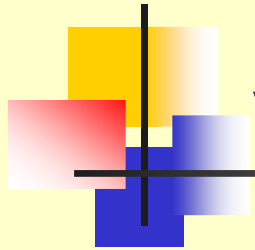
- Se aplică metode de reducere a numărului de stări astfel încât să nu se modifice funcționarea circuitului



## 5.5.1 Etapele de sinteză

---

- 4. Se decide modul de **implementare a registrului de stări interne**
  - Se aleg tipurile de circuite folosite: bistabile, registre, numărătoare
- 5. Se determină **funcțiile de excitație** și **funcțiile de ieșire**
- 6. Se studiază problemele de **hazard**
  - Se urmăresc ieșirile false sau tranzițiile false
- 7. Se desenează **schema circuitului**



## 5.5.1 Etapele de sinteză

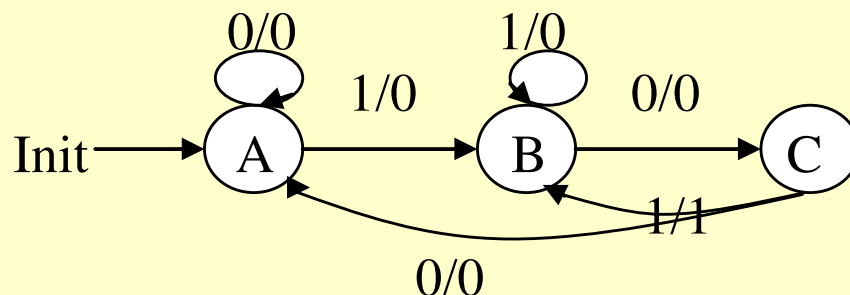
---

### ■ Condiții impuse:

- 1. Orice tranziție între 2 stări ale circuitului se face într-un singur impuls de tact
- 2. La un moment dat circuitul se poate găsi într-o singură stare
- 3. Un circuit care se găsește la un moment dat într-o stare dată, cu un set de intrări dat, poate avea o singură stare următoare

## 5.5.2 Sinteza CLS sincrone cu elemente de memorie

- Implementarea registrului de stări interne se realizează cu bistabile de tip D sau JK
- **Exemplu:** Să se recunoască secvența 101 în șirul de cifre binare 10101
- Reprezentăm funcționarea prin **graf de tranziții**
  - Graful de tranziții are în noduri stările circuitului
  - Pe arce avem tranziția dintr-o stare în alta pentru o anumită intrare, cu o anumită ieșire



## 5.5.2 Sinteza CLS sincrone cu elemente de memorie

- Notăm:
  - $x$  = intrare
  - $z$  = ieșire
- Avem 3 stări notate A, B, C  $\Rightarrow$  este nevoie de 2 variabile de stare pentru codificarea stărilor
- Alegem **codificarea**: A = 00, B = 01 și C = 11
- Reprezentăm funcționarea prin **tabel de tranziții**:

$Q_1Q_0$	$S_t$	$S_{t+1}, z$	
		$x=0$	$x=1$
00	A	A,0	B,0
01	B	C,0	B,0
11	C	A,0	B,1



## 5.5.2 Sinteza CLS sincrone cu elemente de memorie

### ■ Implementarea registrului de stare cu bistabile D

- Funcțiile de excitație (condiționările pentru intrările bistabilelor,  $D_1$  și  $D_0$ ) se deduc explicitând stările circuitului în momentul  $t$  și în momentul  $t+1$
- Stările se vor înlocui cu codurile lor ( $A=00$ ,  $B=01$ ,  $C=11$ )
- Tabelul de excitație al bistabilului D ne dă valorile care trebuie aplicate pe intrările bistabilelor D, ca să se obțină ieșirile dorite

$S_t$ $(Q_1 Q_0)^t$	$S_{t+1} (Q_1 Q_0)^{t+1}$		$z$	
	$D_1 D_0$ $x=0$	$D_1 D_0$ $x=1$	$x=0$	$x=1$
00 (A)	00 (A)	01 (B)	0	0
01 (B)	11 (C)	01 (B)	0	0
11 (C)	00 (A)	01 (B)	0	1

## 5.5.2 Sinteza CLS sincrone cu elemente de memorie

- Implementarea registrului de stare cu bistabile D
  - Generatorul noii stări se realizează cu **porți logice**
  - Se fac DK pentru intrările D1 și D0 și se obține forma minimizată pentru funcții

D<sub>1</sub>:

Q <sub>1</sub> Q <sub>0</sub> \ x	0	1
	0	0
00	0	0
01	1	0
11	0	0
10	x	x

$$D_1 = \overline{Q_1} \cdot Q_0 \cdot \overline{x}$$

D<sub>0</sub>:

Q <sub>1</sub> Q <sub>0</sub> \ x	0	1
	0	1
00	0	1
01	1	1
11	0	1
10	x	x

## 5.5.2 Sinteza CLS sincrone cu elemente de memorie

- Implementarea registrului de stare cu bistabile D
  - Ieșirea se implementează cu **porți logice**
  - Se face DK pentru ieșire și se obține forma minimizată

z:

Q <sub>1</sub> Q <sub>0</sub> \ x	0	1
	0	1
00	0	0
01	0	0
11	0	1
10	x	x

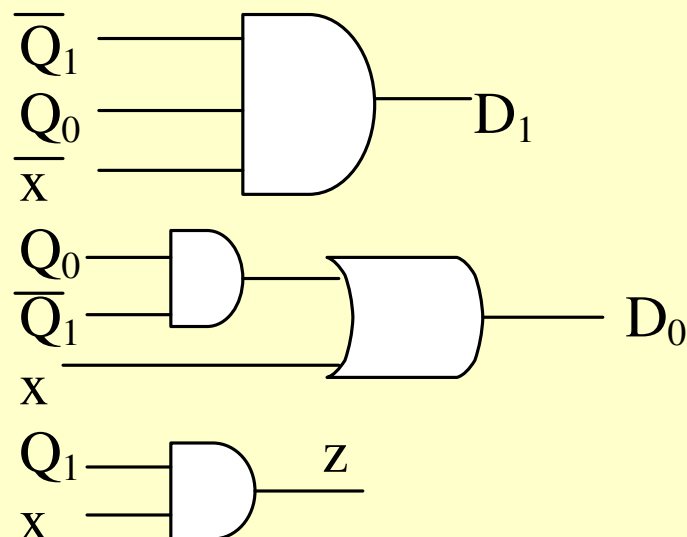
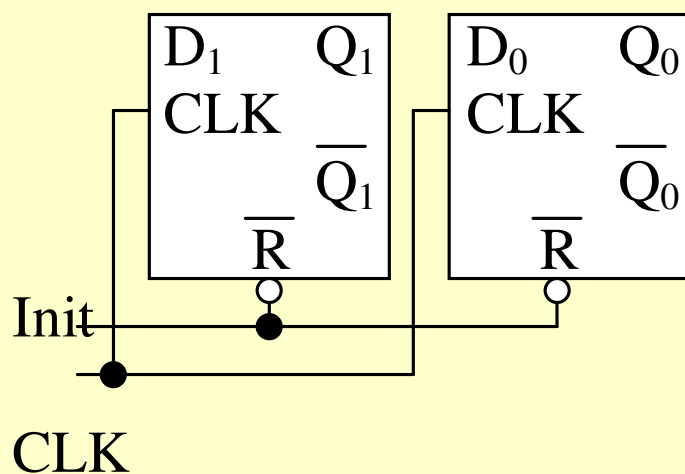
$$z = Q_1 \cdot x$$

## 5.5.2 Sinteza CLS sincrone cu elemente de memorie

- Implementarea registrului de stare cu bistabile D
  - Tranziții false:
    - La trecerea din starea C (11) în starea A (00) se poate trece prin starea B (01) (deoarece variabilele de stare nu se modifică simultan), ceea ce nu corespunde funcționării normale a circuitului
    - Pentru a rezolva această situație se poate introduce o stare suplimentară

## 5.5.2 Sinteza CLS sincrone cu elemente de memorie

- Implementarea registrului de stare cu bistabile D
  - **Schema** pentru circuitul logic secvențial sincron se desenează făcând conexiuni de tip logic între semnale



## 5.5.2 Sinteza CLS sincrone cu elemente de memorie

- Implementarea registrului de stare cu bistabile JK
  - Tabelul de tranziții se completează ținând cont de tabelul de excitație pentru bistabilul JK

$Q_t$	$Q_{t+1}$	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

$Q_1Q_0$	$S_t$	$S_{t+1}, z$	
		x=0	x=1
00	A	A,0	B,0
01	B	C,0	B,0
11	C	A,0	B,1

$S_t$ $Q_1Q_0$	$S_{t+1}(Q_1Q_0)^{t+1}$		$J_1K_1$		$J_0K_0$		$J_1K_1$		$J_0K_0$		$z$	
	x=0	x=1	x=0		x=1		x=0		x=1		x=0	x=1
00 (A)	00	01	0x	0x	0x	1x	0	0	0	0	0	0
01 (B)	11	01	1x	x0	0x	x0	0	0	0	0	0	0
11 (C)	00	01	x1	x1	x1	x0	0	1	0	1	0	1

## 5.5.2 Sinteza CLS sincrone cu elemente de memorie

### ■ Implementarea registrului de stare cu bistabile JK

- Generatorul noii stări se realizează cu **porți logice**
- Se fac DK și se obține forma minimizată pentru funcții

$J_1$ :

$Q_1Q_0 \backslash x$	0	1
00	0	0
01	1	0
11	x	x
10	x	x

$$J_1 = Q_0 \cdot \overline{x}$$

$K_1$ :

$Q_1Q_0 \backslash x$	0	1
00	x	x
01	x	x
11	1	1
10	x	x

$$K_1 = 1$$

$J_0$ :

$Q_1Q_0 \backslash x$	0	1
00	0	1
01	x	x
11	x	x
10	x	x

$$J_0 = x$$

$K_0$ :

$Q_1Q_0 \backslash x$	0	1
00	x	x
01	0	0
11	1	0
10	x	x

$$K_0 = Q_1 \cdot \overline{x}$$

## 5.5.3 Sinteza CLS sincrone – generatorul noii stări

- **Implementarea generatorului noii stări** – se poate realiza cu:
  - Porți logice (vezi exemplul anterior)
  - Multiplexoare
  - Decodificatoare
  - Memorii și multiplexoare

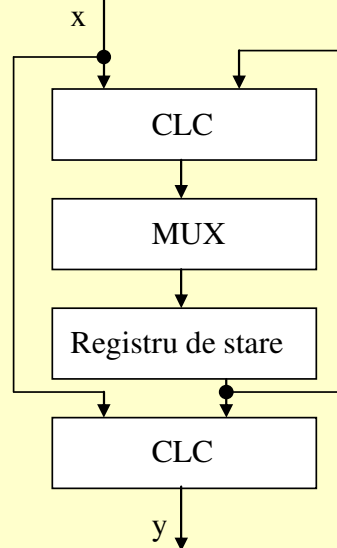


## 5.5.3 Sinteza CLS sincrone – generatorul noii stări

### ■ Generatorul noii stări

#### ■ Multiplexoare

- Funcțiile de excitație pentru intrările bistabilelor se scriu în forma canonică (fără a se minimiza)
- Dacă  $x$  = vectorul intrărilor și  $y$  = vectorul ieșirilor, schema bloc a unui sistem secvențial sincron devine:

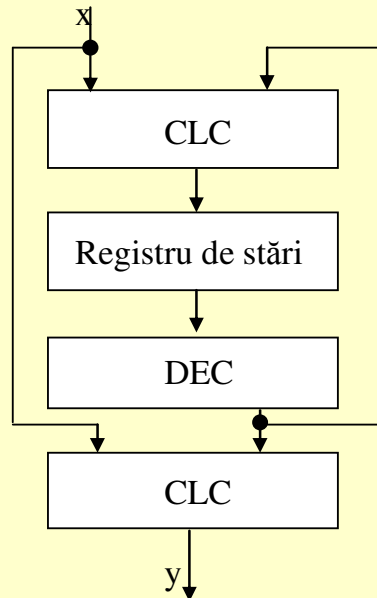


## 5.5.3 Sinteza CLS sincrone – generatorul noii stări

### ■ Generatorul noii stări

#### ■ Decodificatoare

- La intrarea decodicatorului se aplică variabilele de stare (Q)
- La ieșire se obțin stările interne individualizate
- Schema bloc a unui sistem secvențial sincron devine:

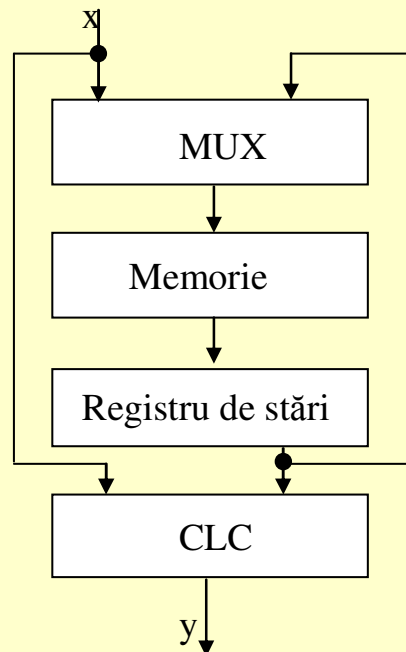


## 5.5.3 Sinteza CLS sincrone – generatorul noii stări

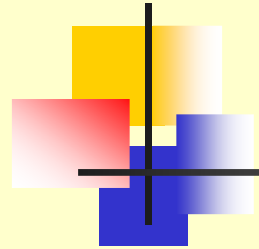
### ■ Generatorul noii stări

#### ■ Memorii și multiplexoare

- Se folosește la circuite complexe
- Schema bloc a unui sistem secvențial sincron devine:



## 5.5.3 Sinteza CLS sincrone – ieșirile



### ■ Implementarea ieșirilor

- Se poate realiza cu:
  - Porți logice (vezi exemplul anterior)
  - Multiplexoare
  - Decodificatoare
  - Memorii
  - PLA-uri
  - Etc.