

LB  
(curs 9 - 59)

## Operația de reset a elementelor secvențiale

### □ Semnalul de reset (set)

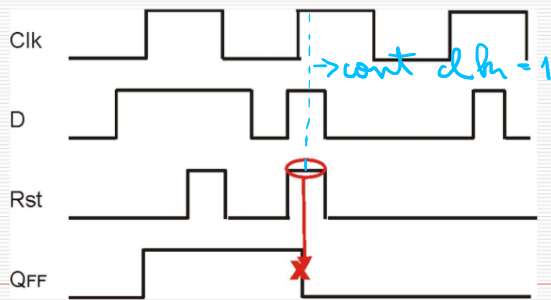
- Funcționalitate – aducerea bistabilului într-o stare “inițială” cunoscută (de obicei starea 0)
- Reset este un semnal global – este aplicat tuturor elementelor de memorie dintr-un sistem digital

#### ■ Tipuri de reset

- Reset sincron
- Reset asincron

### □ Reset sincron

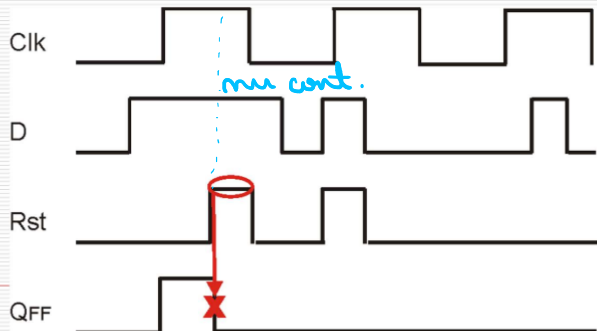
- Este activ doar pe palierul (latch) sau front-ul (FF) activ al semnalului de clock



doar dacă  $rst = 1$  pe frontedge  
sau la FF  
sau pe palier la latch

### □ Reset asincron

- Resetează elementul secvențial indiferent de valoarea semnalului de clock



### □ Reset sincron vs Reset asincron

```
always
  @posedge clk
begin
  if (rst)
    begin
      q <= 0;
    end
  else
    begin
      q <= d;
    end
end
```

```
always
  @posedge clk, posedge rst
begin
  if (rst)
    begin
      q <= 0;
    end
  else
    begin
      q <= d;
    end
end
```

Flip-flop name	Flip-flop symbol	Characteristic table	Characteristic equation	Excitation table																																			
SR		<table><tr><th>S</th><th>R</th><th>Q(next)</th></tr><tr><td>0</td><td>0</td><td>Q</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>NA</td></tr></table>	S	R	Q(next)	0	0	Q	0	1	0	1	0	1	1	1	NA	$Q(next)=S+R'Q$ $SR=0$	<table><tr><th>Q</th><th>Q(next)</th><th>S</th><th>R</th></tr><tr><td>0</td><td>0</td><td>0</td><td>d</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>d</td><td>0</td></tr></table>	Q	Q(next)	S	R	0	0	0	d	0	1	1	0	1	0	0	1	1	1	d	0
S	R	Q(next)																																					
0	0	Q																																					
0	1	0																																					
1	0	1																																					
1	1	NA																																					
Q	Q(next)	S	R																																				
0	0	0	d																																				
0	1	1	0																																				
1	0	0	1																																				
1	1	d	0																																				
JK		<table><tr><th>J</th><th>K</th><th>Q(next)</th></tr><tr><td>0</td><td>0</td><td>Q</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>Q'</td></tr></table>	J	K	Q(next)	0	0	Q	0	1	0	1	0	1	1	1	Q'	$Q(next)=JQ'+K'Q$	<table><tr><th>Q</th><th>Q(next)</th><th>J</th><th>K</th></tr><tr><td>0</td><td>0</td><td>0</td><td>d</td></tr><tr><td>0</td><td>1</td><td>1</td><td>d</td></tr><tr><td>1</td><td>0</td><td>d</td><td>1</td></tr><tr><td>1</td><td>1</td><td>d</td><td>0</td></tr></table>	Q	Q(next)	J	K	0	0	0	d	0	1	1	d	1	0	d	1	1	1	d	0
J	K	Q(next)																																					
0	0	Q																																					
0	1	0																																					
1	0	1																																					
1	1	Q'																																					
Q	Q(next)	J	K																																				
0	0	0	d																																				
0	1	1	d																																				
1	0	d	1																																				
1	1	d	0																																				
D		<table><tr><th>D</th><th>Q(next)</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	D	Q(next)	0	0	1	1	$Q(next)=D$	<table><tr><th>Q</th><th>Q(next)</th><th>D</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	Q	Q(next)	D	0	0	0	0	1	1	1	0	0	1	1	1														
D	Q(next)																																						
0	0																																						
1	1																																						
Q	Q(next)	D																																					
0	0	0																																					
0	1	1																																					
1	0	0																																					
1	1	1																																					
T		<table><tr><th>T</th><th>Q(next)</th></tr><tr><td>0</td><td>Q</td></tr><tr><td>1</td><td>Q'</td></tr></table>	T	Q(next)	0	Q	1	Q'	$Q(next)=TQ'+T'Q$	<table><tr><th>Q</th><th>Q(next)</th><th>T</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	Q	Q(next)	T	0	0	0	0	1	1	1	0	1	1	1	0														
T	Q(next)																																						
0	Q																																						
1	Q'																																						
Q	Q(next)	T																																					
0	0	0																																					
0	1	1																																					
1	0	1																																					
1	1	0																																					

## ❑ 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**.

## Registre

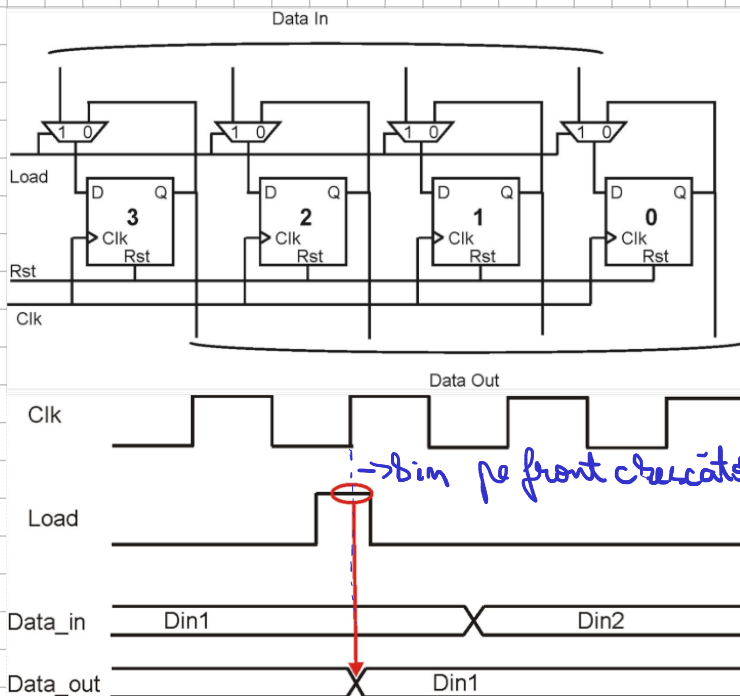
- ❑ Reprezintă o colecție/grupare de  $n$  bistabile
- ❑ Nr maxim de valori a unui registru pe  $n$  biti –  $2^n$  valori binare
- ❑ Folosit pentru memorarea unui cuvânt de date/unei stări curente a sistemului

## PIPO

Registru cu intrare paralela – ieșire paralela

- ❑ Prezintă semnale de încărcare (*Load*)
- ❑ La fiecare front crescător valoarea registrului se actualizează cu *Data In*, dacă este activ semnalul de *Load*

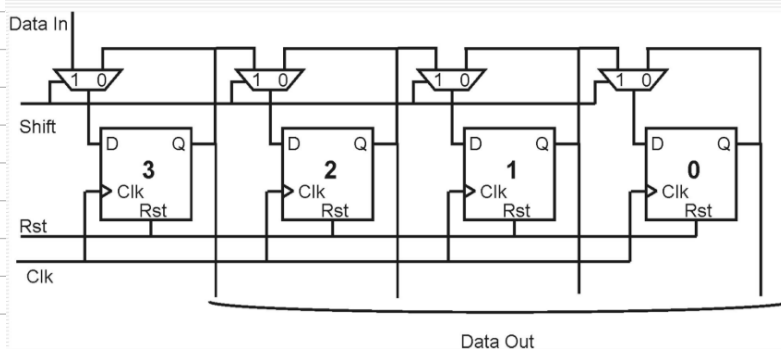
Load	Stare viitoare (Data Out)
0	Nu se schimbă
1	Data In



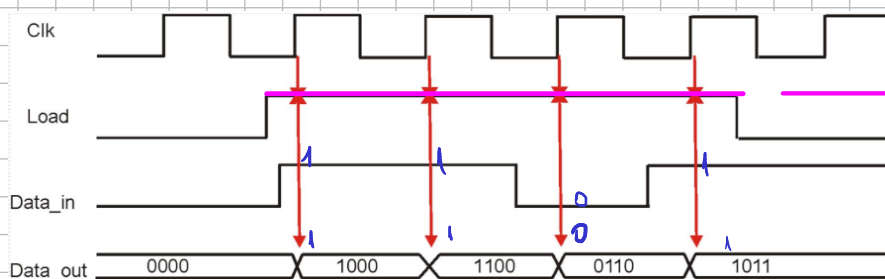
SIPO

## Registru cu intrare seriala – iesire paralela

- ☐ Functia de deplasare (shift-are) in interiorul registrului
- ☐ Datele se introduc serial in registru – o singura intrare de date
- ☐ La fiecare activare a semnalului de Load (Shift), datele se deplaseaza in cadrul registrului
- ☐ Incarcarea a  $n$  biti necesita  $n$  cicluri de clock

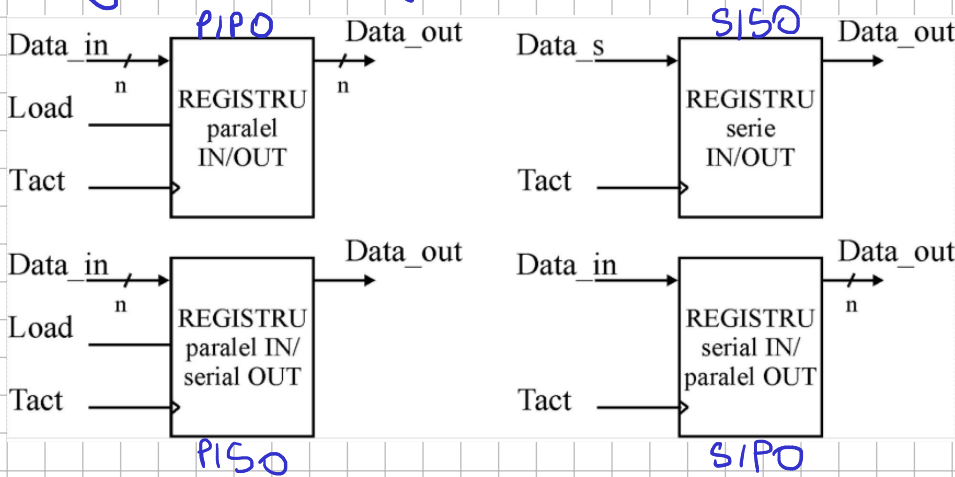


Shift	Starea Curenta	Starea viitoare
0	$Q_3Q_2Q_1Q_0$	$Q_3Q_2Q_1Q_0$ (nu se schimba)
1	$Q_3Q_2Q_1Q_0$	$DataInQ_3Q_2Q_1$



→ schimbări doar pe Load = 1

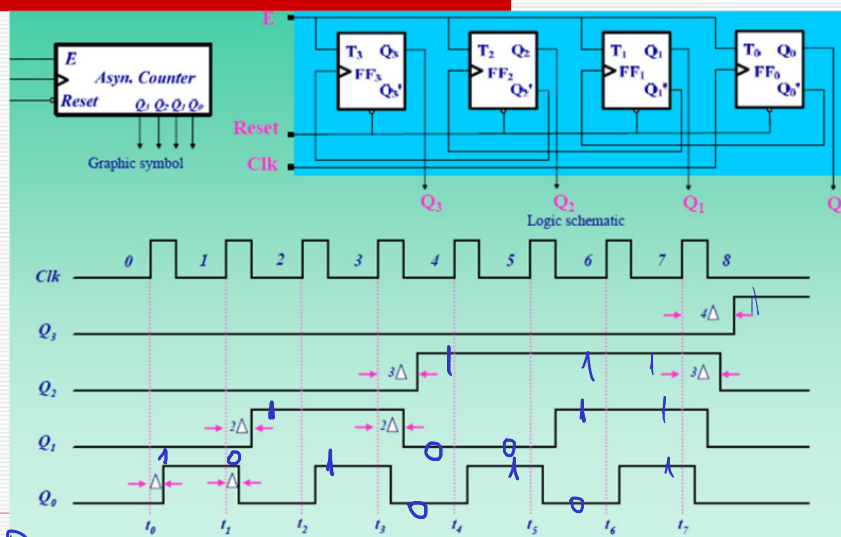
# Registre - clasificare



## Numărătoare

- circuite **secvențiale sincrone autonome** (mulțimea intrărilor de DATE 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 din modului în care se generează clk-ul.
- Funcție de direcția de parcurgere a secvenței de stări:
  - numărător în sens crescător,
  - numărător în sens descrescător,
  - numărător reversibil (ambele sensuri).

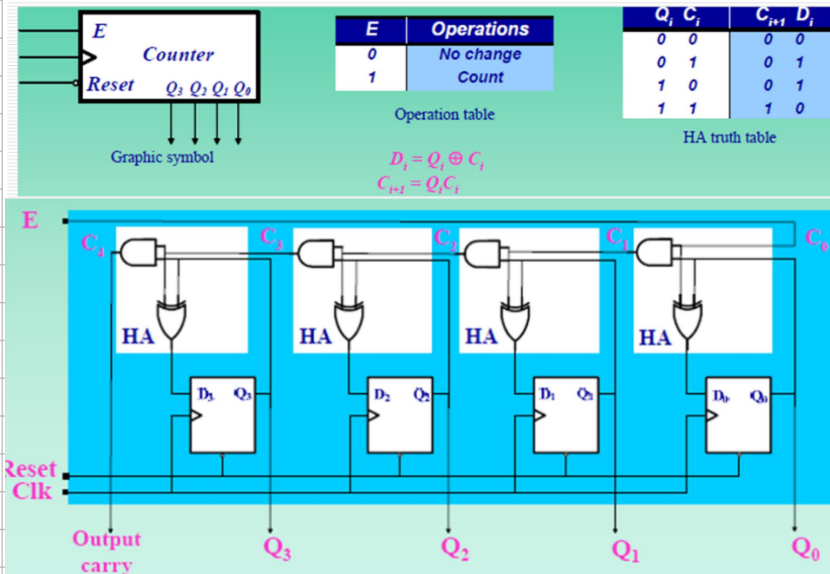
## Numarator asincron- realizat cu bistabile T



0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000

# Numarator sincron

- Numărătoarele increm/decrem conținutul când primesc semnal de activare; Toate elem secventiale au acelasi tact!

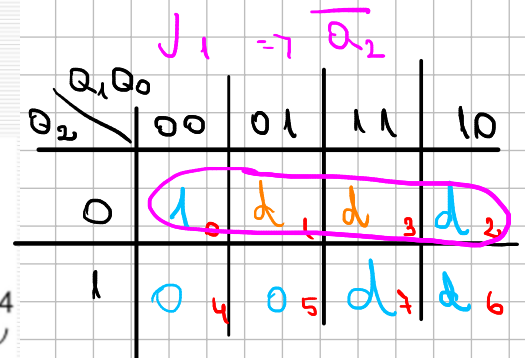
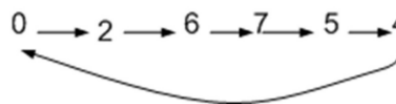
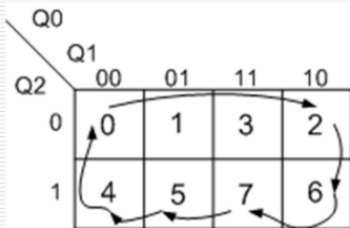


$$b_i = Q_i \oplus C_i = Q_i' C_i + Q_i C_i'$$

$$C_{i+1} = Q_i C_i$$

## Aplicație

- Realizați un numărător folosind FF-uri de tip J-K care numără după următoarea secvență:



- Indicații:

- construiți tabelul de adevăr pentru determinarea expresiei intrărilor J-K
- Completați J-K funcție de starea următoare (ex. starea curentă 0, starea următoare e trecută pe rândul următor: st. 2) și de tabelul excitațiilor

$Q_n$	$Q_{n+1}$	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

Starea	$Q_2$	$Q_1$	$Q_0$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	d	1	d	0	d
2	0	1	0	1	d	d	0	0	d
6	1	1	0	d	0	d	0	1	d
7	1	1	1	d	0	d	1	d	0
5	1	0	1	d	0	0	d	d	1
4	1	0	0	d	1	0	d	0	d
0	0	0	0						

- Minimizăm funcțiile  $J_i(Q_0, Q_1, Q_2)$   $K_i(Q_0, Q_1, Q_2)$ . Stările prin care nu trece numărătorul sunt notate cu "don't care"

$$J_0 = Q_2 Q_1 \quad K_0 = \bar{Q}_1$$

$$J_1 = Q_2' Q_1' Q_0' = Q_2' \text{ (după Karnaugh)}$$

și tot așa

$$K_1 = \dots$$

4. Implementare cu FF-uri J-K M-S și porți logice ȘI-NU:

