

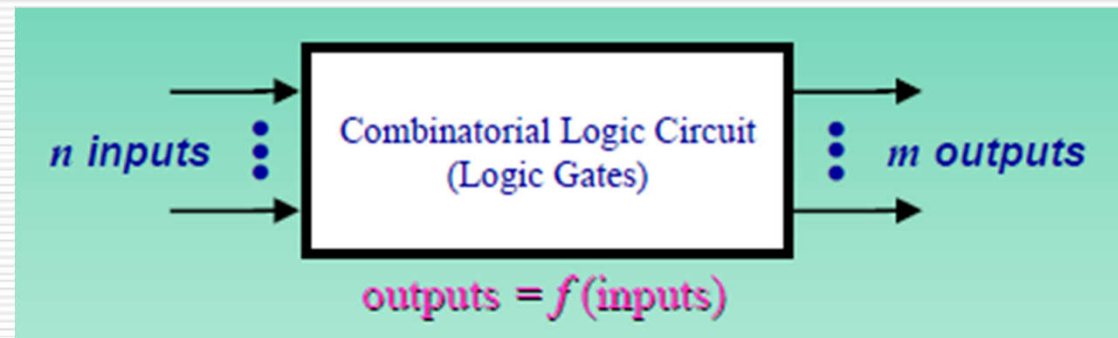
# Logică digitală

-Curs 9-  
Circuite logice  
secvențiale

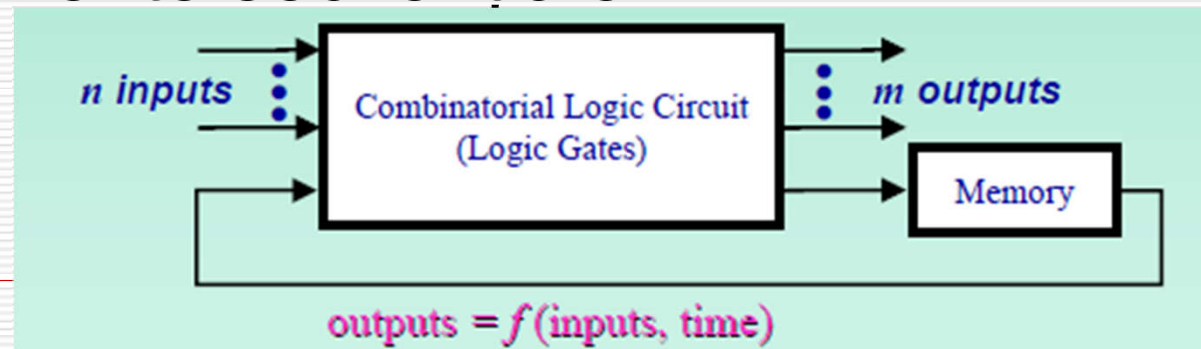
# Clasificare componente digitale

## □ Componente combinaționale

- Ușor de analizat, partiționat, verificat



## □ Componente secvențiale

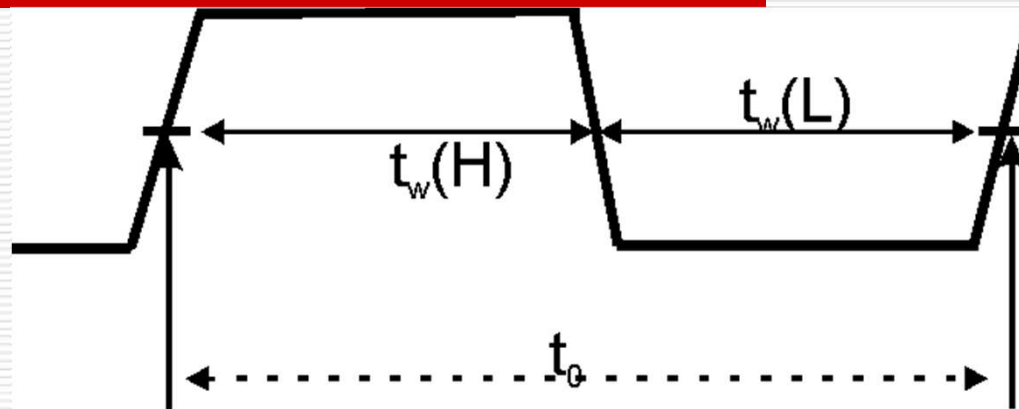


# Circuite secvențiale

---

- **Circuitele secvențiale se clasifică:**
    - **Asincrone**
    - **Sincrone**
  
  - Componentele secvențiale asincrone își modifică starea și valorile de ieșire funcție de modificările semnalelor de la intrare (**oricând!**) se modifică acestea.
  
  - Componentele secvențiale sincrone își modifică valoarea funcție de valoarea semnalelor de intrare la **momente bine definite de timp**, dictate de un semnal (de intrare) care se numește tact (*clock*)
-

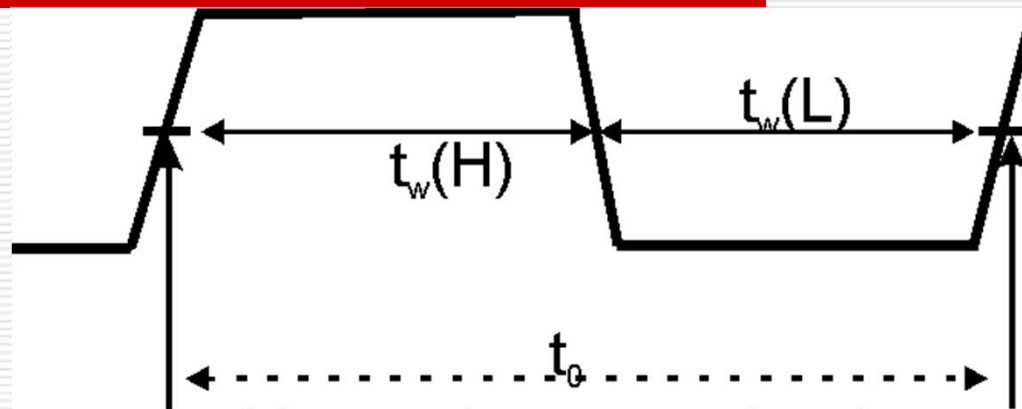
# Semnalul de tact



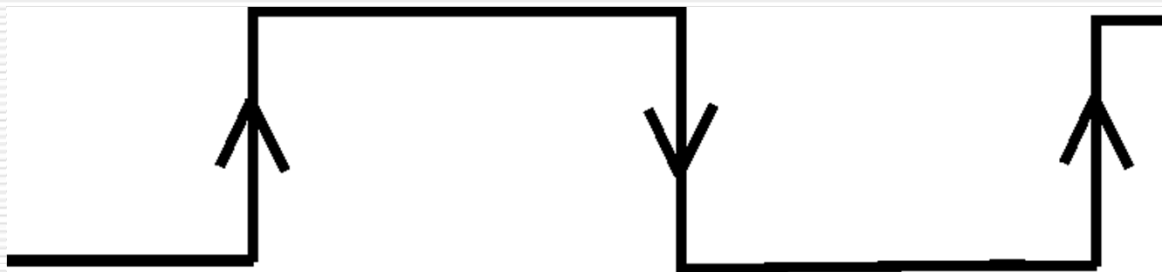
- ❑ **Palierul** unui semnal reprezintă porțiunea unde acesta rămâne constant 0 logic (palier negativ) și 1 logic (palier pozitiv).
- ❑ **Frontul crescător** se referă la porțiunea unde semnalul își modifică valoarea de la 0 logic la 1 logic (mai exact de la 10% din nivelul corespunzător lui 1 logic la 90% din nivelul corespunzător lui 1 logic)
- ❑  $t_0$  - perioada semnalului de tact,
- ❑  $t_w(H)$  și  $t_w(L)$  reprezintă durata unui impuls de 1 respectiv 0 logic

# Semnalul de tact

---



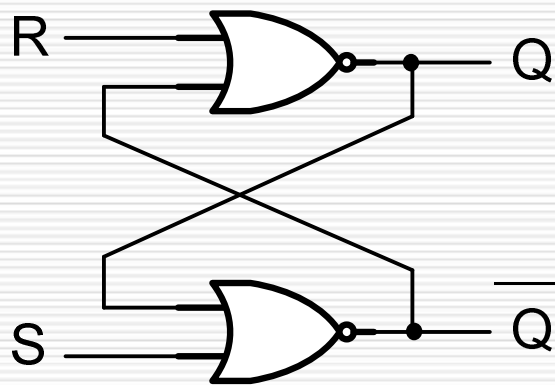
a) Semnal rectangular de tact



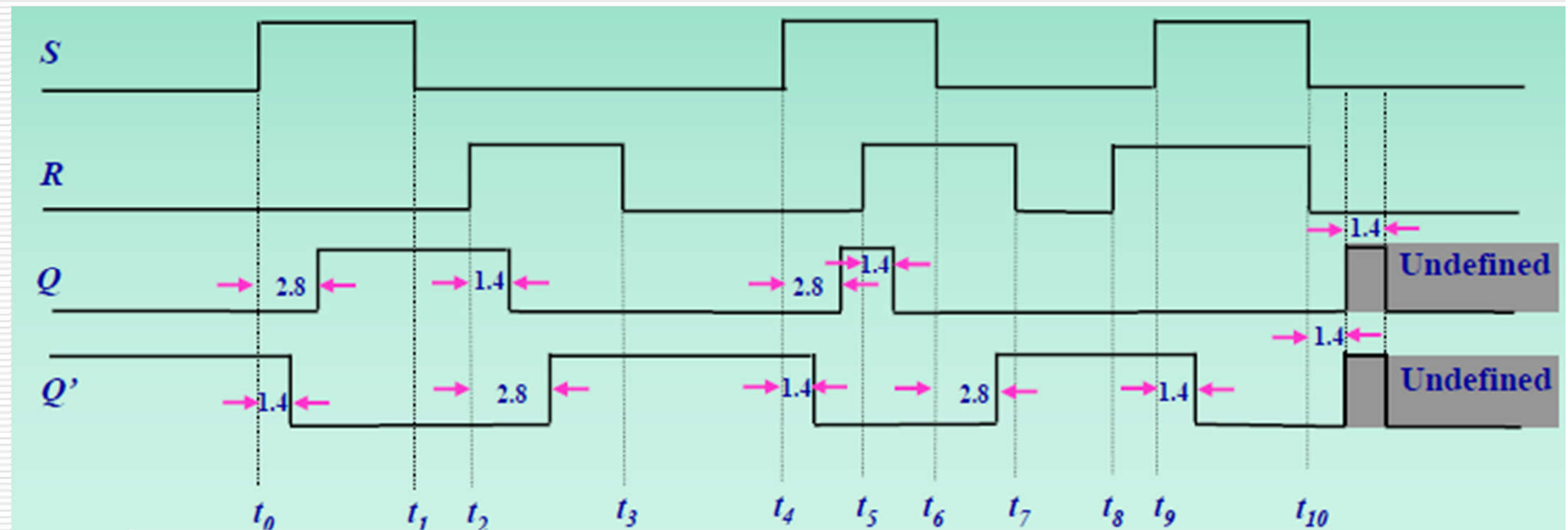
b) Formă idealizată a semnalului

---

# S-R Latch (SAU-NU) - asincron

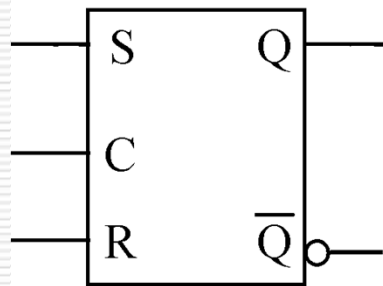


R	S	$Q_{t+1}$
0	0	$Q_t$
0	1	1
1	0	0
1	1	interzis



# Gated SR-latch

---

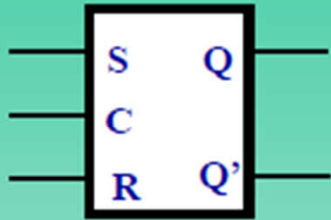


S	R	C	Q <sub>next</sub>	$\overline{Q}_{next}$
0	0	1	Q	$\overline{Q}$
0	1	1	0	1
1	0	1	1	0
1	1	1	-	-
*	*	0	Q	$\overline{Q}$

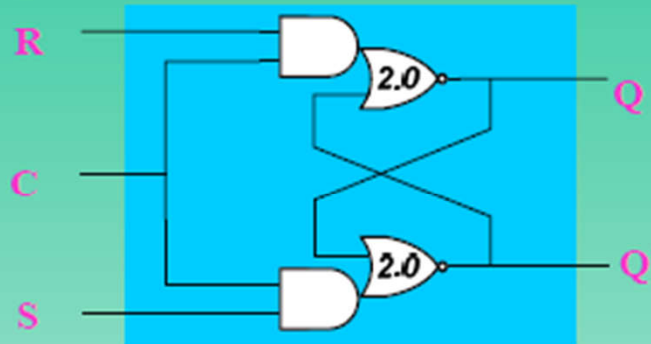
- ❑ Când semnalul C este activ valorile de la intrare sunt propagate prin latch
  - ❑ Semnalele de intrare nu trebuie să se modifice în intervalul  $t_{setup}$  și  $t_{hold}$  al frontului descrescător
-

# Gated SR-Latch

•Control signal C activates the latch



Graphic symbol



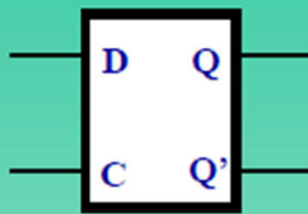
Logic schematic

<i>C</i>	<i>S</i>	<i>R</i>	<i>Q</i>	<i>Q(next)</i>	
0	X	X	0	0	(inactive)
0	X	X	1	1	(inactive)
1	0	0	0	0	(hold)
1	0	0	1	1	(hold)
1	0	1	X	0	(reset)
1	1	0	X	1	(set)
1	1	1	X	NA	(?)

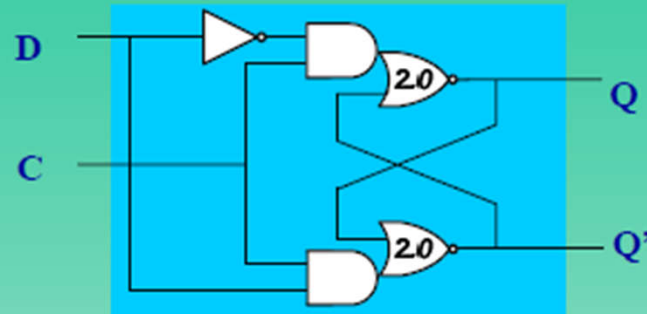
Truth table



# Gated D-latch



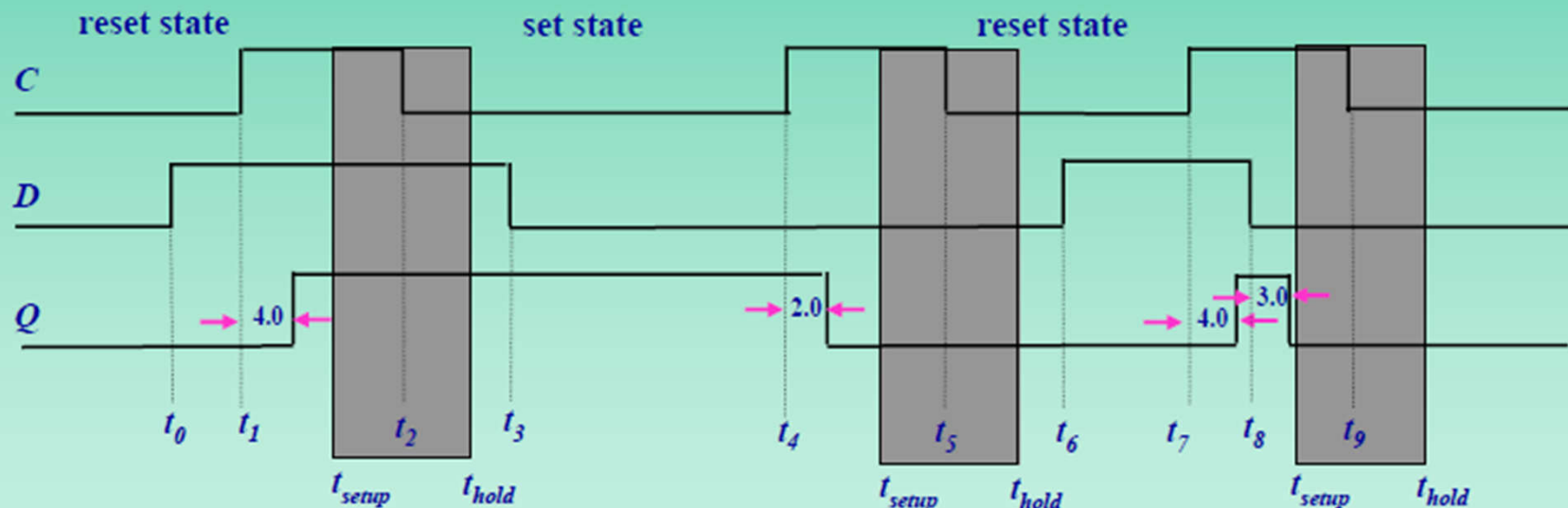
Graphic symbol



Logic schematic

<i>C</i>	<i>D</i>	<i>Q</i>	<i>Q(next)</i>
0	X	0	0
0	X	1	1
1	0	X	0
1	1	X	1

Truth table



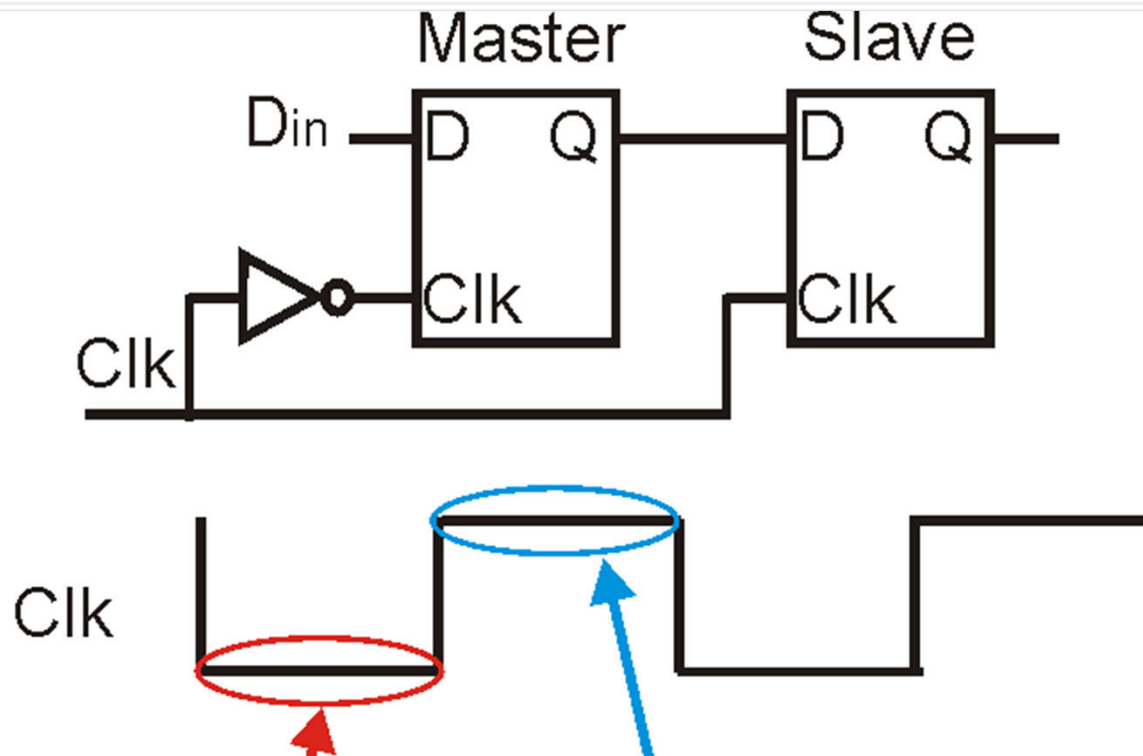
Timing diagram

# Flip-flop-uri

---

- ❑ Se mai numesc și latch-uri sensibile pe frontul semnalului de tact;
  - ❑ Bascularea se face pe frontul semnalului de tact (!nu pe palier – latch-uri)
  - ❑ Două variante de arhitecturi:
    - Configurația master-slave
    - Edge-triggered FF
-

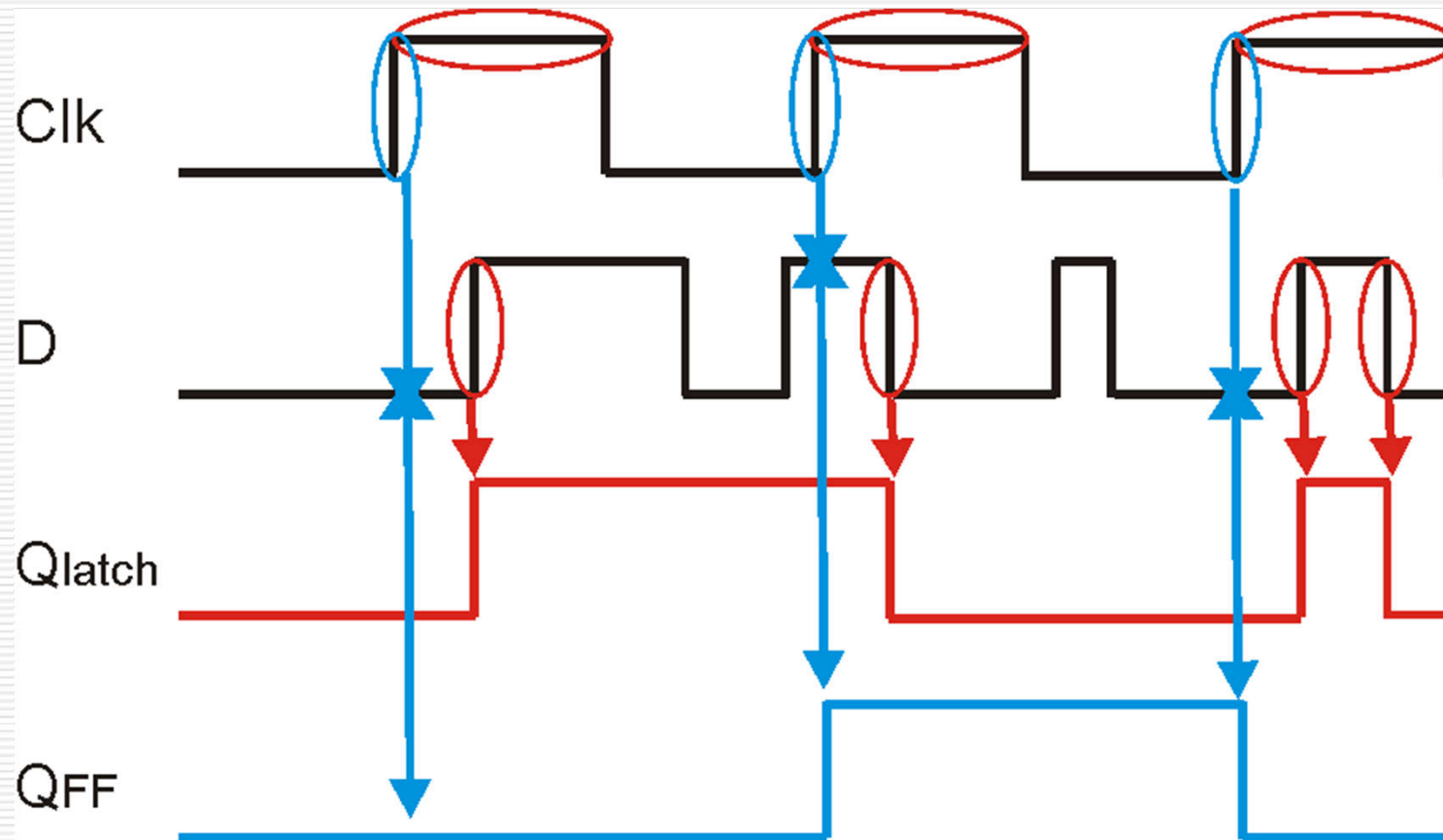
# FF-Master-Slave



Incarcare Master  
Latch cu  $D_{in}$   
Slave Latch  
memoreaza  
data anteriora

Incarcare Slave  
Latch cu  $Q_{master}$   
Master Latch  
memoreaza  
valoarea  $D_{in}$   
dinaintea comutarii  
semnalului  $Clk$

# Latch sincron vs. FF sincron

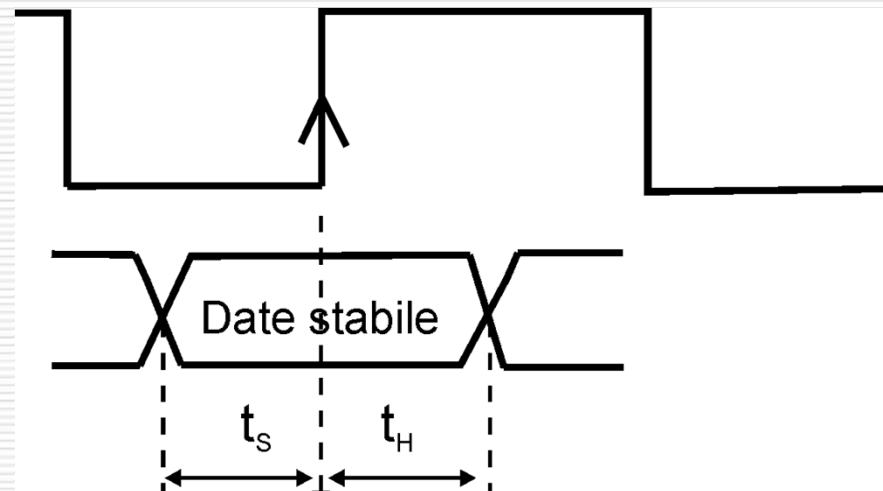


# FF-uri

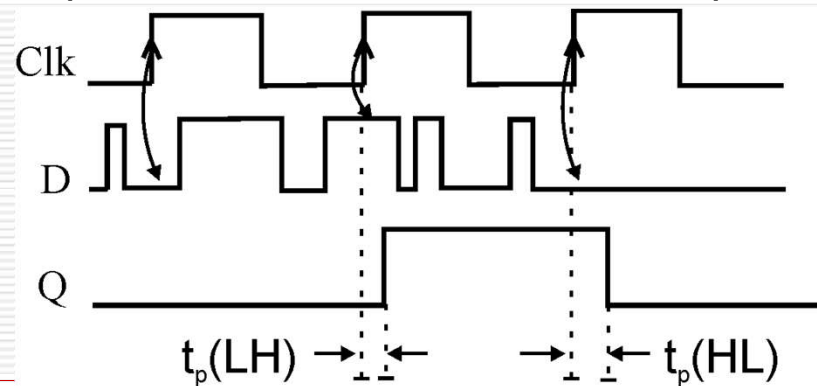
---

- contrângeri de timp in operarea FF-urilor - trebuie să aibe în vedere 3 timpi:
    - **Timpul de setup:** se referă la timpul necesar pentru ca semnalul de intrare (D) să rămână stabil înainte de apariția frontului semnalului de tact;
    - **Timpul de hold:** reprezintă timpul în care datele de intrare nu pot fi modificate după apariția frontului semnalului de tact în vederea încărcării corecte a acestora;
    - **Timpul aferent întârzierii datorate propagării ( $t_p$ ):** constituie timpul necesar basculării FF-ului (clock to Q delay);
-

# Prezentarea celor 3 timpi de propagare



Timpii de setup și hold pentru un FF care basculează pe front crescător al tactului



Timpi de propagare  $t_p$

# Operatia de reset a elementelor secventiale

---

## ☐ Semnalul de reset (set)

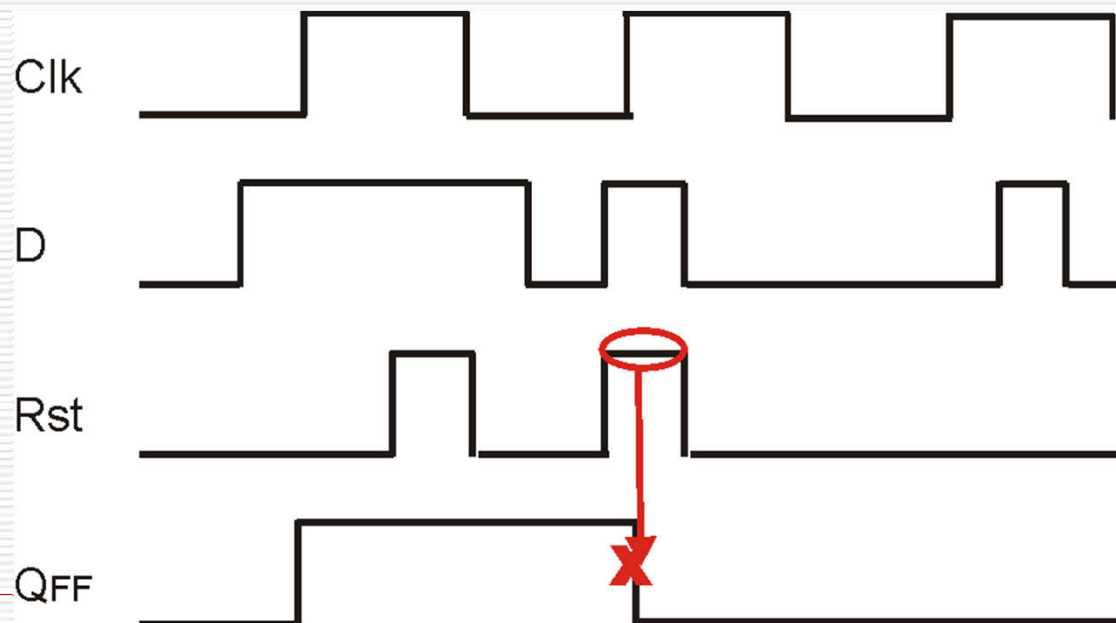
- Functionalitate – aducerea bistabilului intr-o stare “initiala” cunoscuta (de obicei starea 0)
- Reset este un semnal global – este aplicat tuturor elementelor de memorie dintr-un sistem digital
- Tipuri de reset
  - ☐ Reset sincron
  - ☐ Reset asincron

# Operatia de reset a elementelor secventiale

---

## ❑ Reset sincron

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



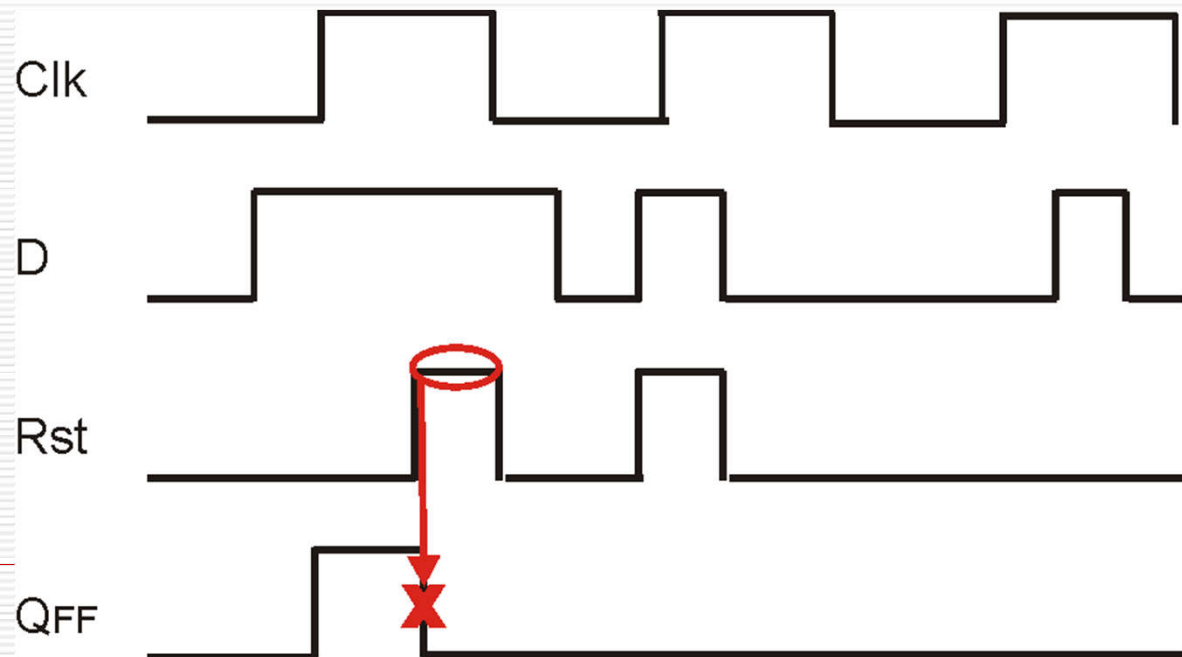


# Operatia de reset a elementelor secventiale

---

## □ Reset asincron

- Reseteaza elementul secvential indiferent de valoarea semnalului de clock



# Operatia de reset a elementelor secventiale

---

## □ 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
```

# Modalitati de descriere: circuite secvențiale

---

- Tabelul caracteristic:
    - Pentru fiecare combinație de intrare, funcție de starea curentă este precizată starea următoare
  - Ecuația caracteristică:
    - Ecuația rezultată în urma aplicării unei metode de minimizare
  - Tabelul excitațiilor:
    - Folosit la sinteza circuitului
    - Specifică intrările necesare pentru a trece din starea curentă în starea următoare
  - Diagrame/grafuri de stare:
    - Graf orientat, în care valorile posibile (stările) sunt reprezentate prin cercuri, iar tranzițiile prin arce
-

# Tipuri de FF-uri

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>X</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>X</td> <td>0</td> </tr> </table>	Q	Q(next)	S	R	0	0	0	X	0	1	1	0	1	0	0	1	1	1	X	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	X																																				
0	1	1	0																																				
1	0	0	1																																				
1	1	X	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>X</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>X</td> </tr> <tr> <td>1</td> <td>0</td> <td>X</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>X</td> <td>0</td> </tr> </table>	Q	Q(next)	J	K	0	0	0	X	0	1	1	X	1	0	X	1	1	1	X	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	X																																				
0	1	1	X																																				
1	0	X	1																																				
1	1	X	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																																					

# Circuite secvențiale reprezentare

---

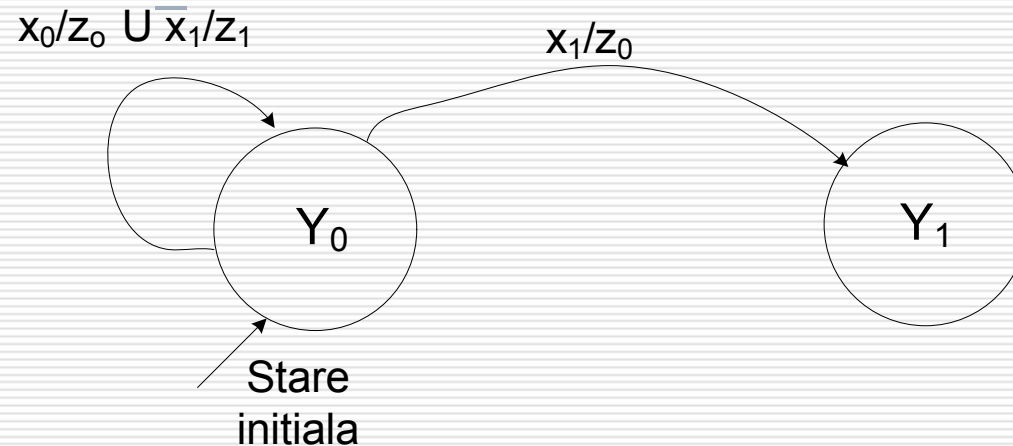
## □ Circuitele secvențiale:

- **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

## □ circuite de tip Mealy:

- fiecare nod se notează cu simbolul stării pe care o reprezintă,
- arcul care pleacă din nod se notează cu perechea intrarea care a generat tranziția circuitului/ ieșirea generată în timpul tranziției.



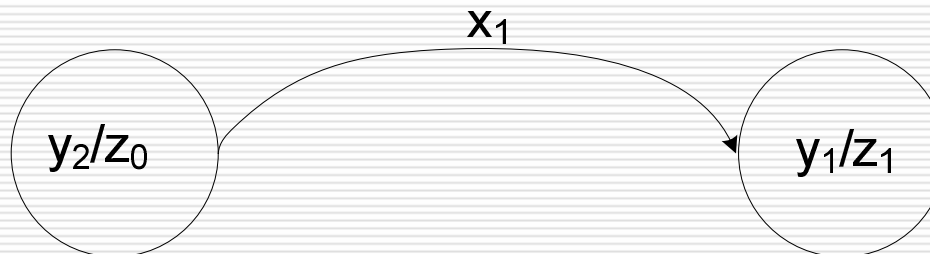
**Starea inițială** se marchează printr-o săgeată aplicată nodului respectiv.

# Circuite secvențiale: diagrame e stare

---

## □ **circuite de tip Moore:**

- nodurile diagramei de stări se notează simbolul stării corespondente și ieșirile
- arcul are notată intrarea care a generat tranziția.



# 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
-



# Registru cu intrare paralela – iesire paralela

---

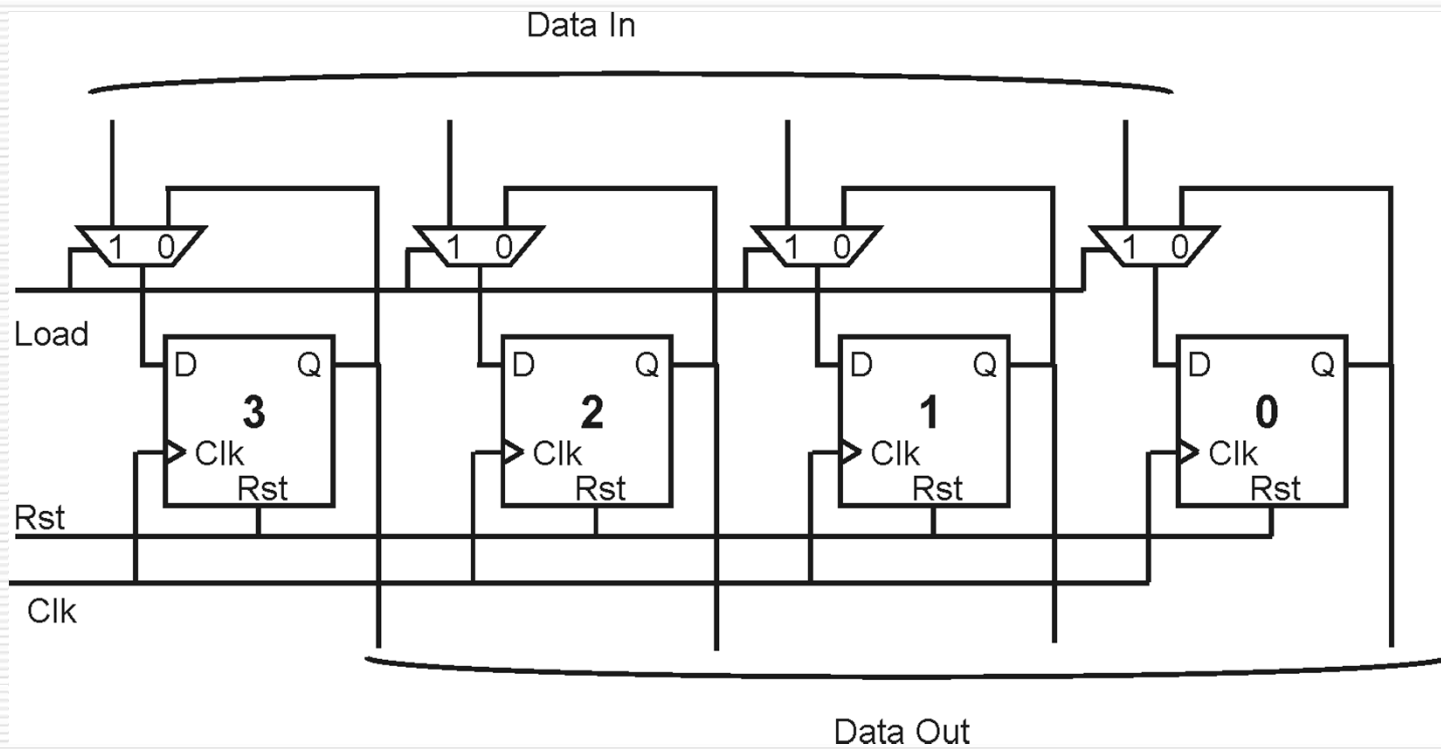
- ❑ Prezinta semnale de incarcare (*Load*)
- ❑ La fiecare front crescator valoarea registrului se actualizeaza cu *Data In*, daca este activ semnalul de *Load*

<i>Load</i>	Stare viitoare ( <i>Data Out</i> )
0	Nu se schimba
1	<i>Data In</i>

---

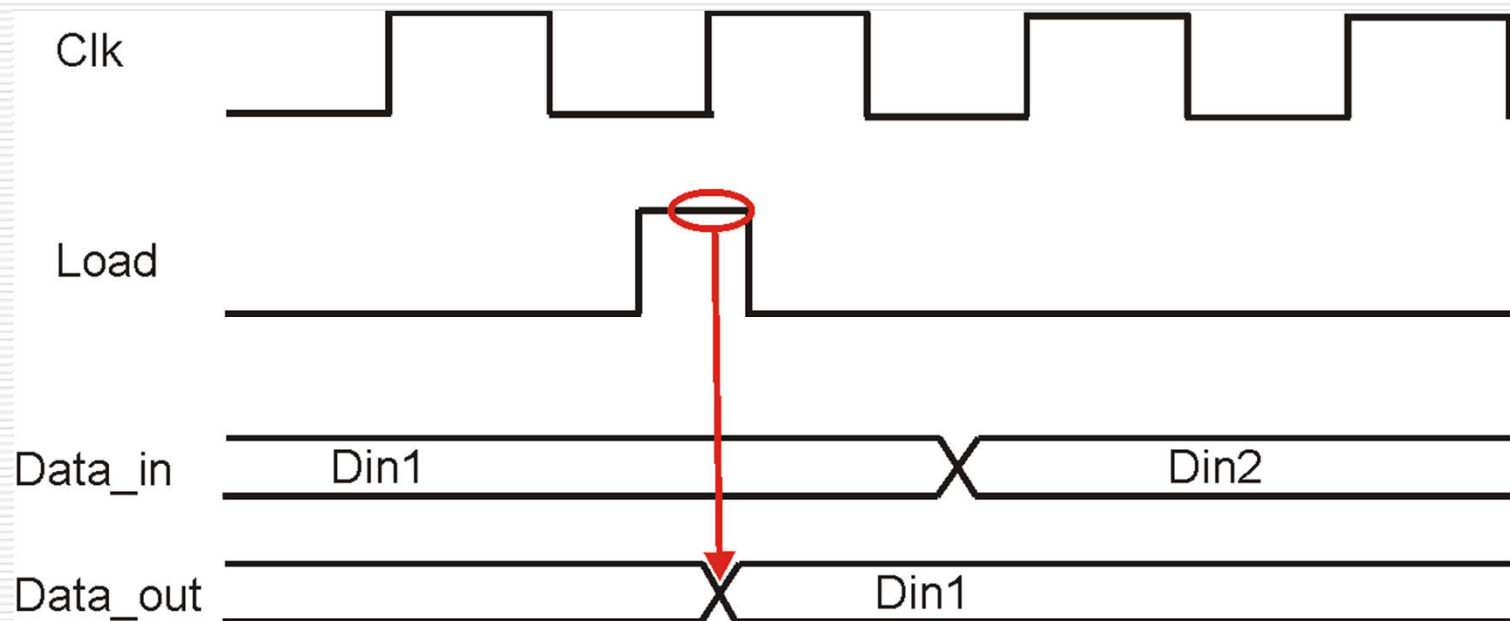
# Registru cu intrare paralela – iesire paralela

---



# Registru cu intrare paralela – iesire paralela

---

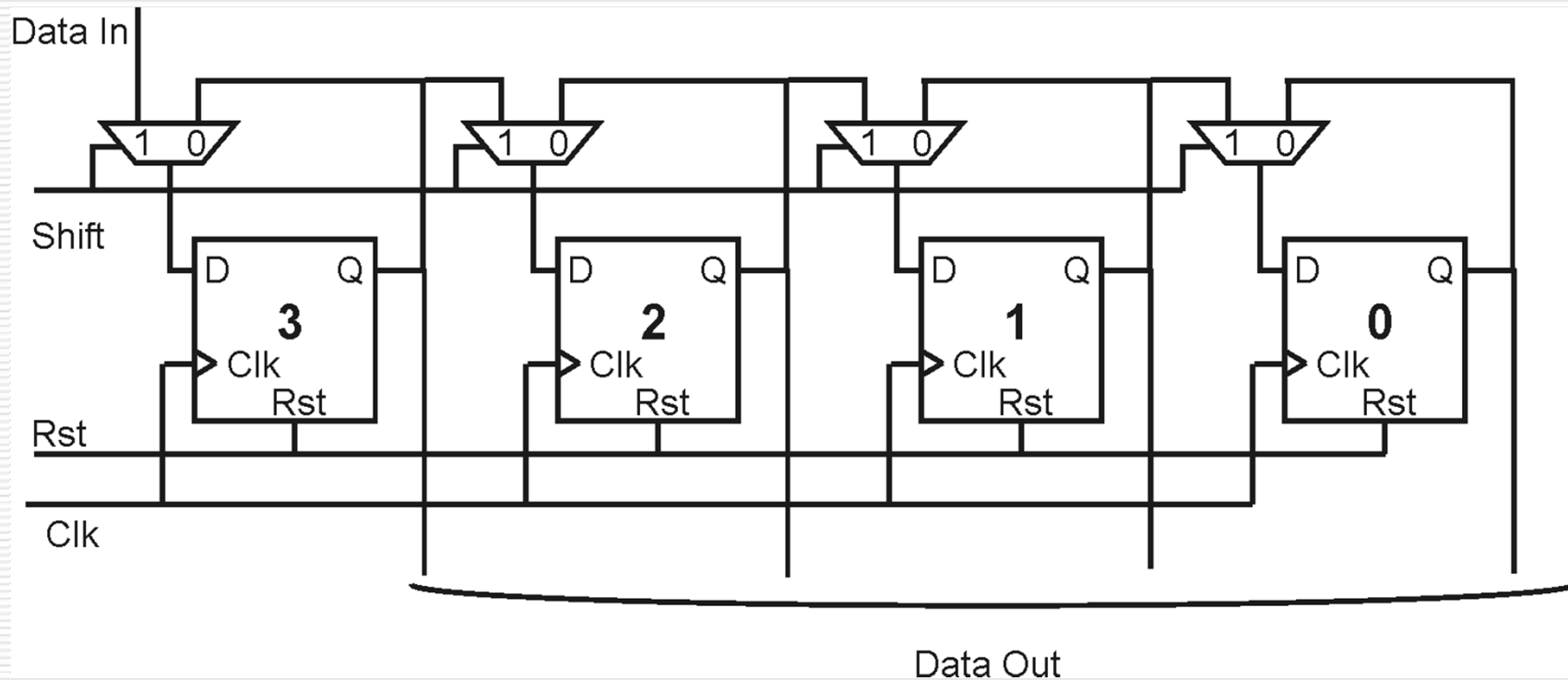


# Registru cu intrare seriala – iesire paralela

---

- ❑ 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
-

# Registru cu intrare seriala – iesire paralela



# Registru cu intrare seriala – iesire paralela

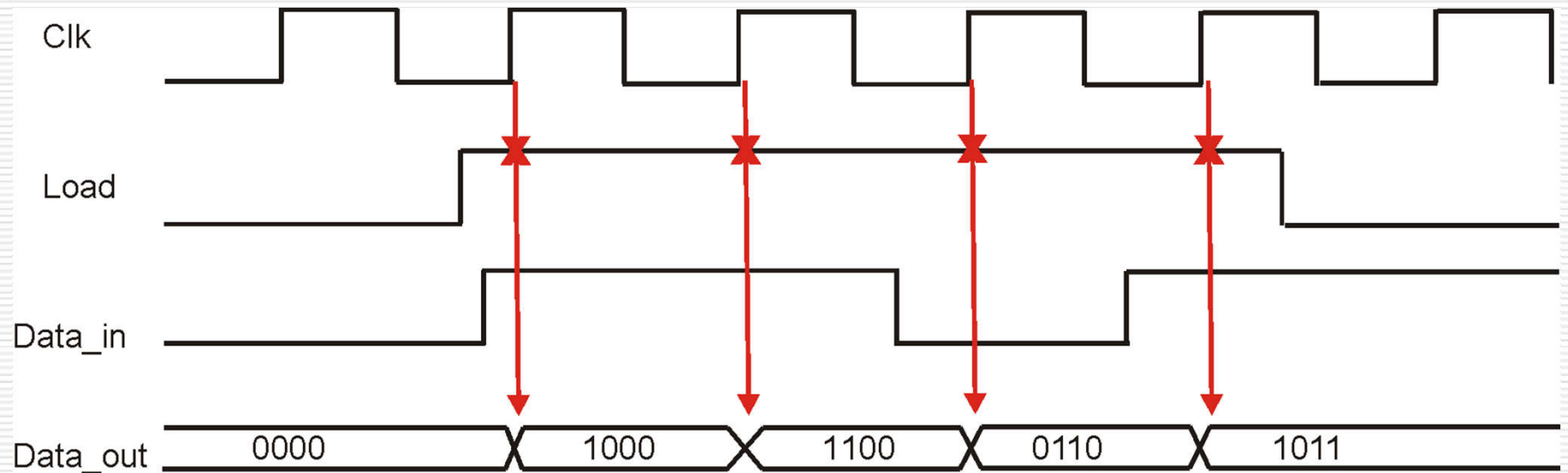
---

Shift	Starea Curenta	Starea viitoare
0	$Q_3Q_2Q_1Q_0$	$Q_3Q_2Q_1Q_0$ (nu se schimba)
1	$Q_3Q_2Q_1Q_0$	$\text{DataIn}Q_3Q_2Q_1$

---

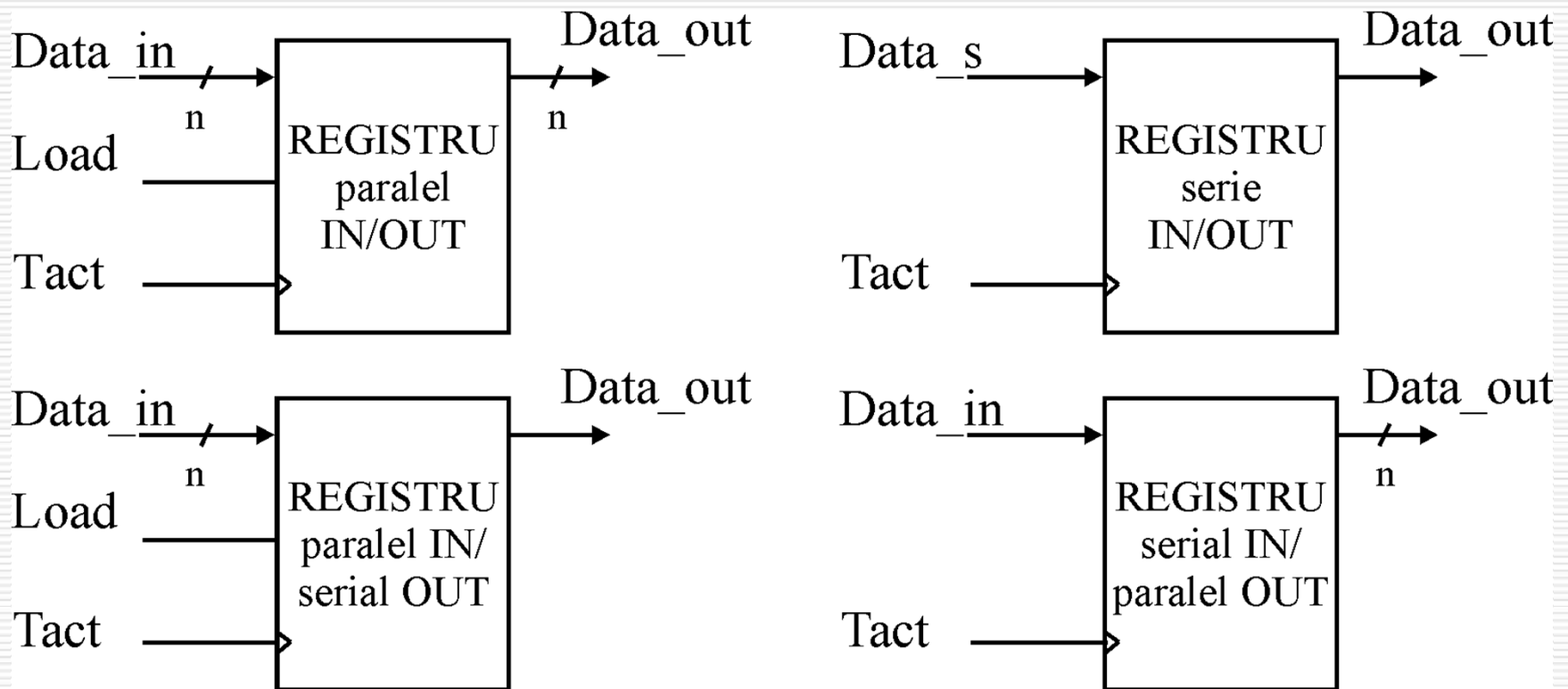
# Registru cu intrare seriala – iesire paralela

---



# Registre - clasificare

---



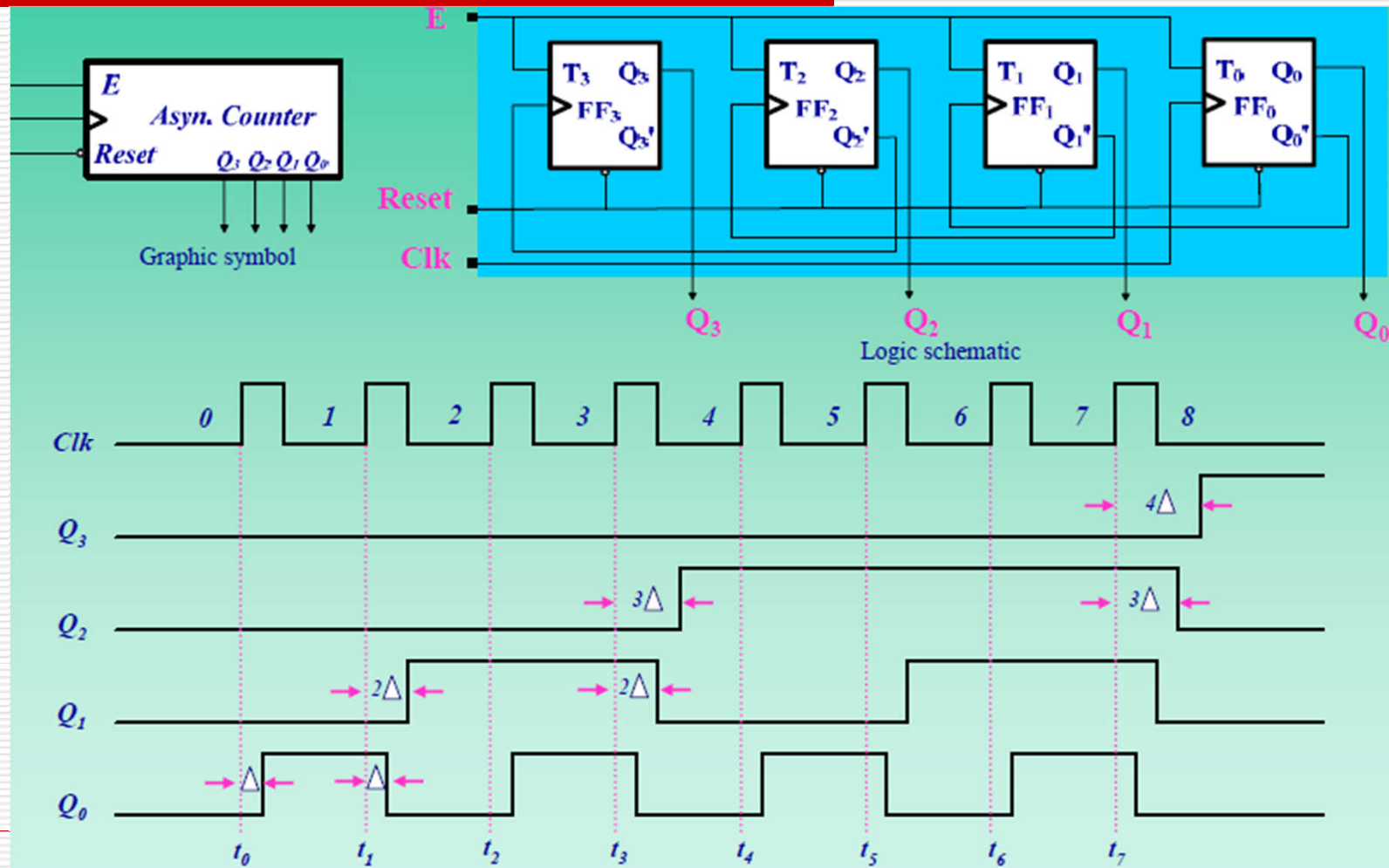


# Numărătoare

---

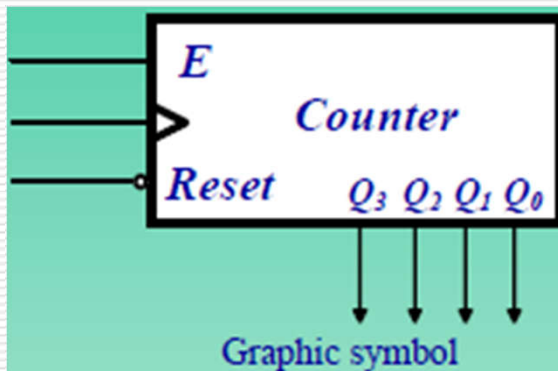
- ❑ circuite **secvențiale sincrone autonome** (mulțimea intrărilor vidă), care baleiază o secvență de stări impuse de proiectant.
  - ❑ de regulă este inițializat cu starea „0” , după care la fiecare impuls de numărare, comuta într-o nouă stare.
  - ❑ caracterul **asincron** al unui numărător este dat 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 asincron- realizat cu bistabile T



# Numarator sincron

- Numărătoarele increm/decrem conținutul când primesc semnal de activare



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

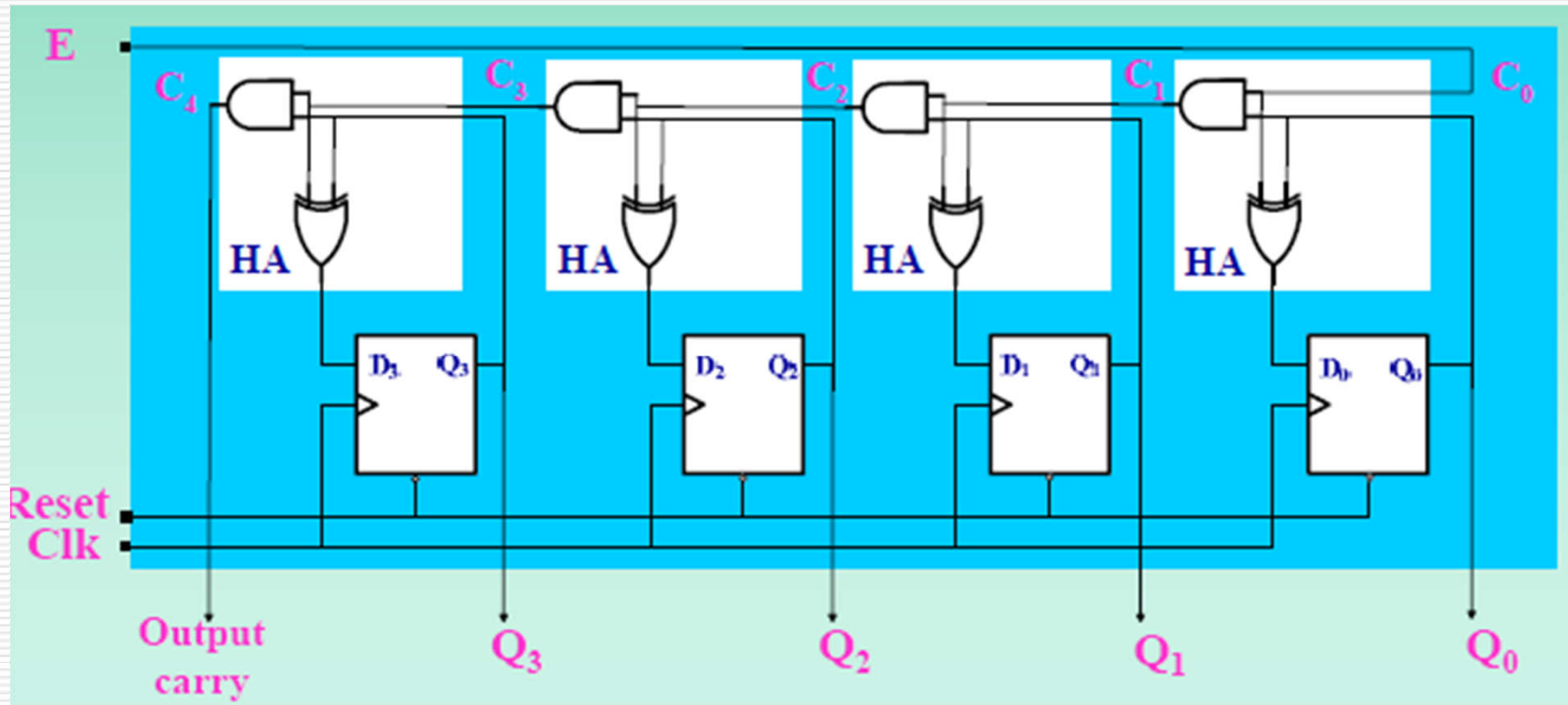
Operation table

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

<i>Q<sub>i</sub></i>	<i>C<sub>i</sub></i>	<i>C<sub>i+1</sub></i>	<i>D<sub>i</sub></i>
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

HA truth table

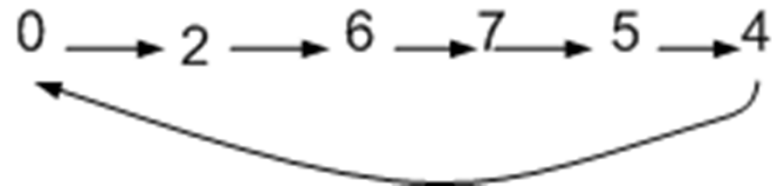
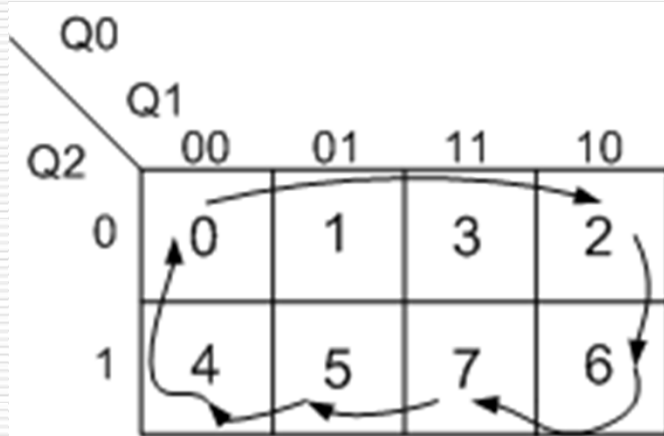
# Numarator sincron



# Aplicație 1\*:

---

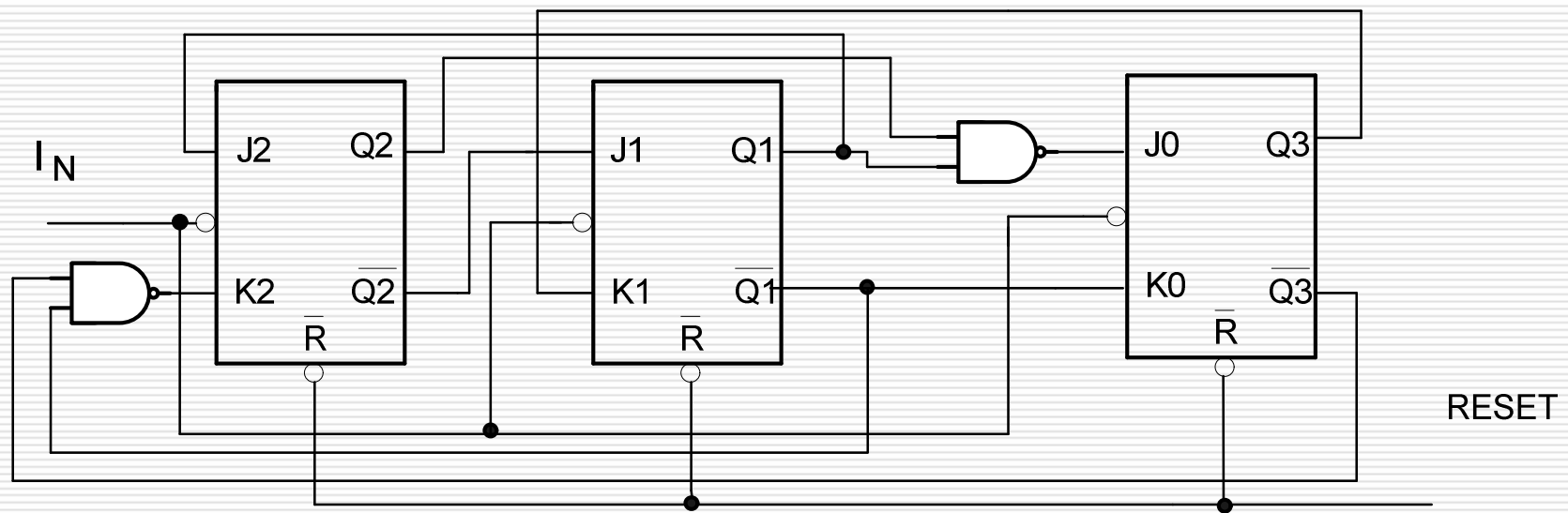
- ❑ Realizați un numărător folosind FF-uri de tip J-K care numără după următoarea secvență:



# Aplicație 1:

---

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



# Aplicație 1:

---

□ Indicații:

1. construiți tabelul de adevăr pentru determinarea expresiei intrărilor J-K
2. 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	x
0	1	1	x
1	0	x	1
1	1	x	0

## Aplicație 1:

**Indicații:**

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

Starea	O <sub>2</sub>	O <sub>1</sub>	O <sub>0</sub>	J <sub>2</sub>	K <sub>2</sub>	J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>
0	0	0	0	0	x	1	x	0	x
2	0	1	0	1	x	x	0	0	x



# Aplicație 1:

---

□ Indicații:

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

Starea	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	J <sub>2</sub>	K <sub>2</sub>	J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>
0	0	0	0	0	x	1	x	0	x
2	0	1	0	1	x	x	0	0	x
6	1	1	0	x	0	x	0	1	x
7	1	1	1	x	0	x	1	x	0
5	1	0	1	x	0	0	x	x	1
4	1	0	0	x	1	0	x	0	x
0	0	0	0						

# Aplicație 1:

---

□ Indicații:

3. Minimizăm funcțiile  $J_i(Q_0, Q_1, Q_2)$   $K_i(Q_0, Q_1, Q_2)$   
 . Stările prin care nu trece numărătoul sunt notate cu “don't care”

$J_0$

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	0 <sub>0</sub>	x <sub>1</sub>	x <sub>3</sub>	0 <sub>2</sub>
1	0 <sub>4</sub>	x <sub>5</sub>	x <sub>7</sub>	1 <sub>6</sub>

$$J_0 = Q_2 Q_1$$

$K_0$

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	x <sub>0</sub>	x <sub>1</sub>	x <sub>3</sub>	x <sub>2</sub>
1	x <sub>4</sub>	1 <sub>5</sub>	0 <sub>7</sub>	x <sub>6</sub>

$$K_0 = \overline{Q_1}$$


---

# Aplicație 1:

---

$J_1$

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	1 <sub>0</sub>	x <sub>1</sub>	x <sub>3</sub>	x <sub>2</sub>
1	0 <sub>4</sub>	0 <sub>5</sub>	x <sub>7</sub>	x <sub>6</sub>

$$J_1 = \overline{Q_2}$$

$K_1$

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	x <sub>0</sub>	x <sub>1</sub>	x <sub>3</sub>	0 <sub>2</sub>
1	x <sub>4</sub>	x <sub>5</sub>	1 <sub>7</sub>	0 <sub>6</sub>

$$K_1 = Q_0$$

$J_2$

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	0 <sub>0</sub>	x <sub>1</sub>	x <sub>3</sub>	1 <sub>2</sub>
1	x <sub>4</sub>	x <sub>5</sub>	x <sub>7</sub>	x <sub>6</sub>

$$J_2 = Q_1$$

$K_2$

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	x <sub>0</sub>	x <sub>1</sub>	x <sub>3</sub>	x <sub>2</sub>
1	1 <sub>4</sub>	0 <sub>5</sub>	0 <sub>7</sub>	0 <sub>6</sub>

$$K_2 = \overline{Q_0} \overline{Q_1}$$


---

# Aplicație 1:

---

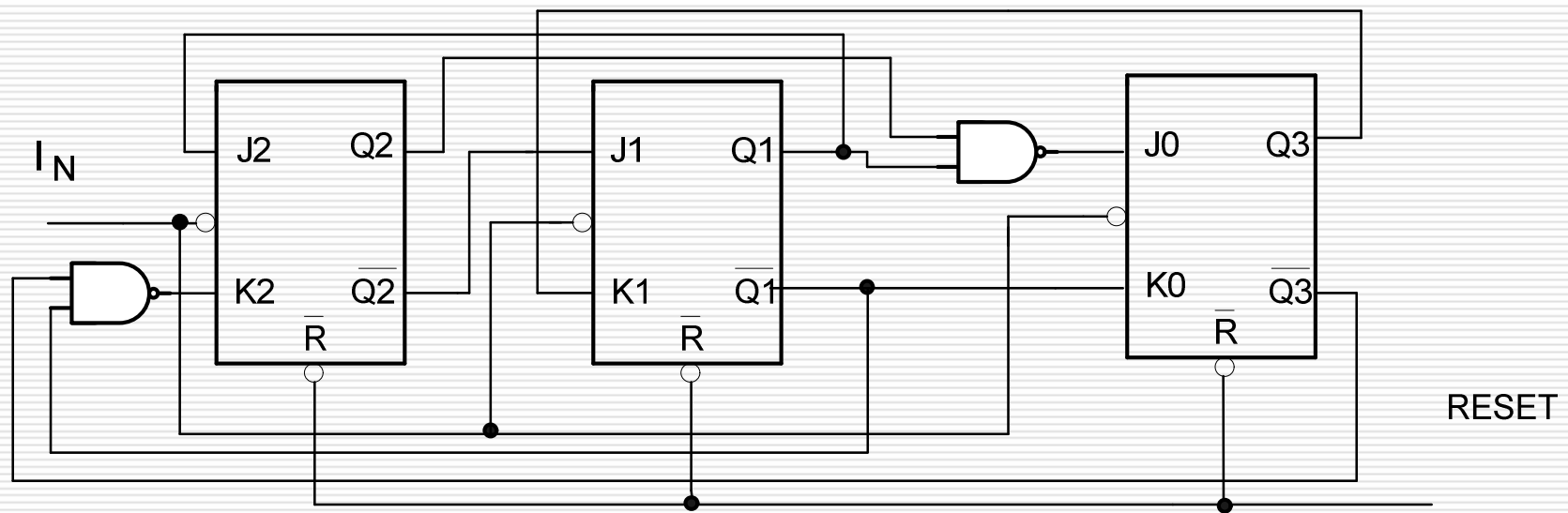
4. Implementare cu FF-uri J-K M-S și porți logice ȘI:

---

# Aplicație 1:

---

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



# Întrebări?

---

---

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

