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

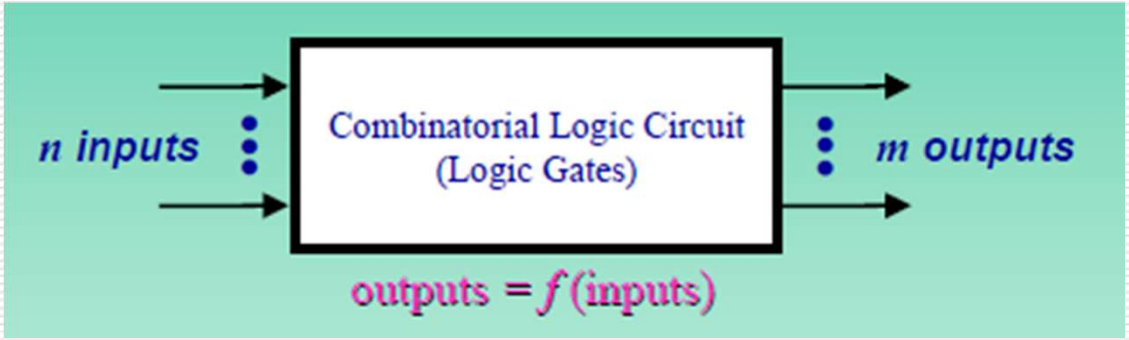
Logică digitală

-Curs 8-9-

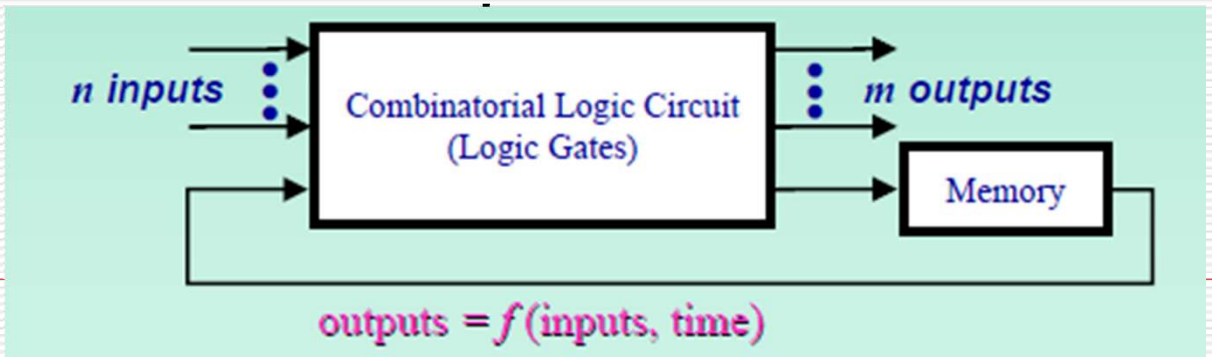
Circuite logice secvențiale

Clasificare componente digitale

- Componente combinaționale
  - Ușor de analizat, partiționat, verificat



- Componente secvențiale



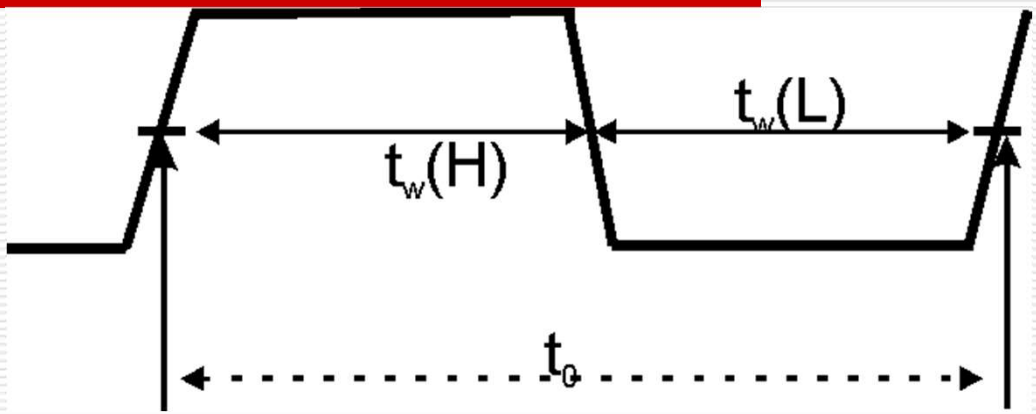
Circuite secvențiale

- **Circuitele secvențiale se clasifică:**
  - **Asincrone**
  - **Sincrone**

- Componenentele secvențiale asincrone își modifică starea și valorile de ieșire funcție de modificările semnalelor de la intrare (oricând!) se modifică acestea.

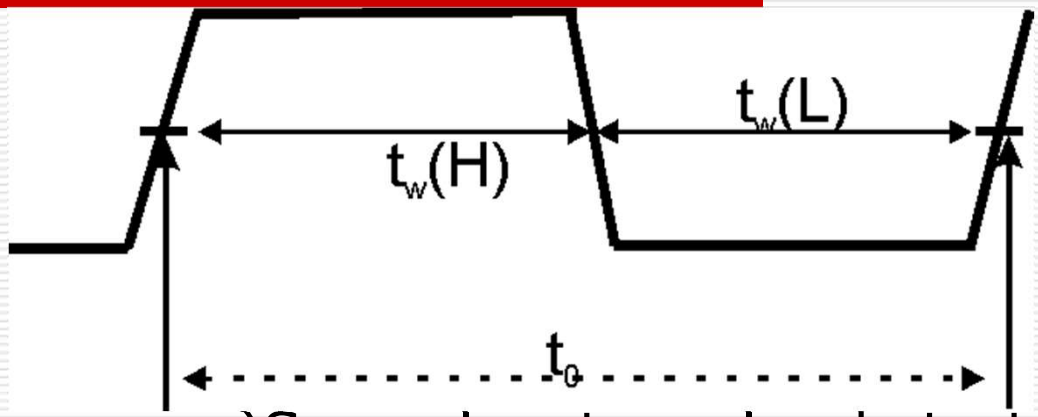
- Componentele secvențiale sincrone își modifcă valoarea funcție de valoarea semnalelor de intrare la momente bine definite de timp, dictate de un semnal (de intrare) care se numește tact (*clock*)

Semnalul de tact

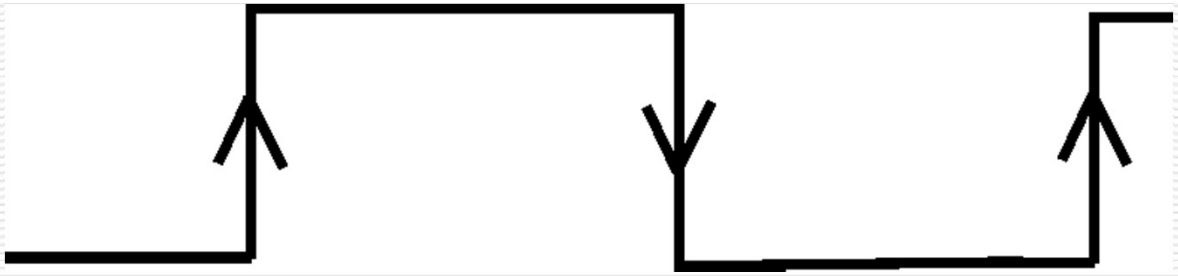


- **Palierul** uni semnal reprezintă porţiunea unde acesta rămâne constant o logic (palier negativ) şi 1 logic (palier pozitiv).
- **Frontul crescător** se referă la porţiunea unde senalul îşi modifică valoarea de la 0 logic la 1 logic (mai exact de la 10% din nivelul corespunzător lui 1 logic la 90% din nivelul corespunzător lui 1 logic
- t0 - perioada semnalului de tact,
- tw(H) şi tw(L) reprezintă durata unui impuls de 1 respectiv 0 logic

Semnalul de tact

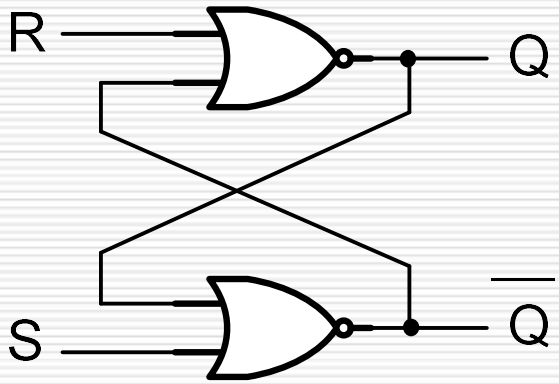


a)Semnal rectangular de tact

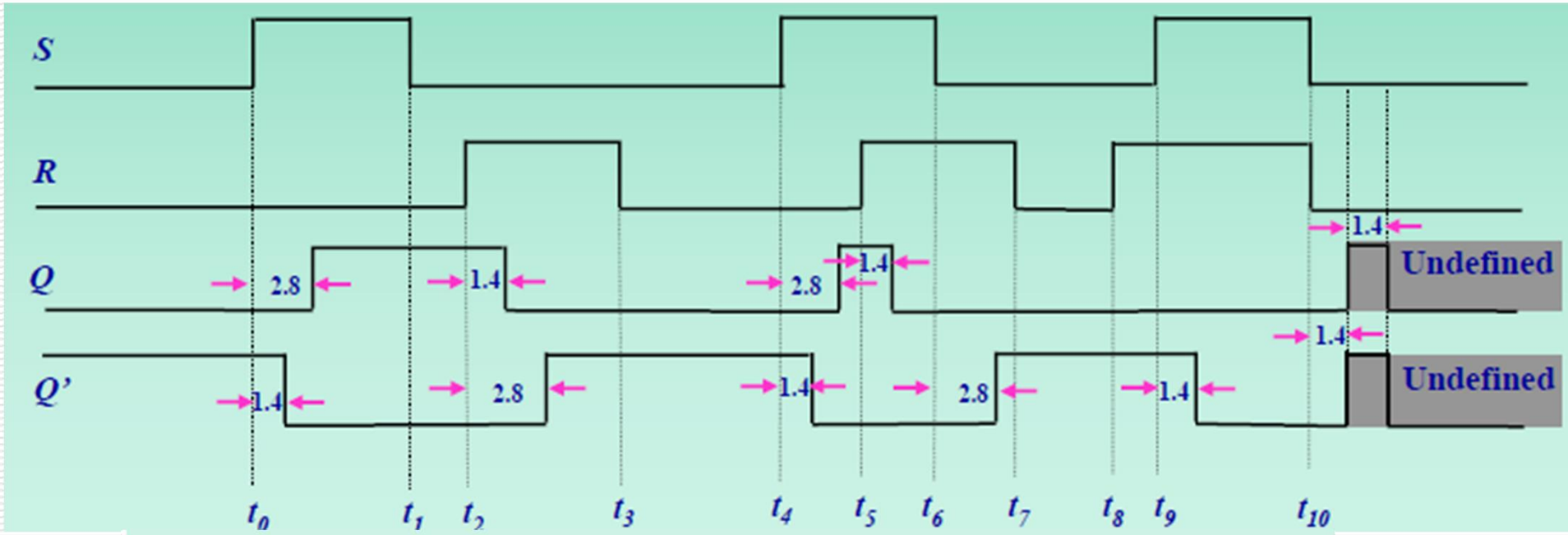


b)Formă idealizată a semnalului

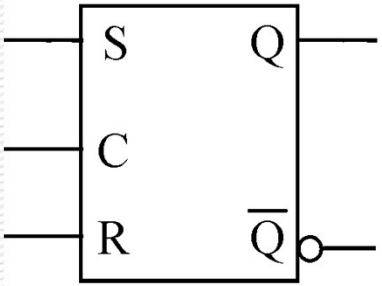
S-R Latch (SAU-NU) - asincron



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



Gated SR-latch

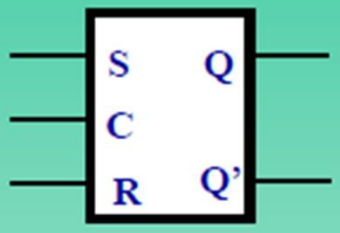


S	R	C	Q_next	$\overline{Q}$_next
0	0	1	Q	$\overline{Q}$
0	1	1	0	1
1	0	1	1	0
1	1	1	-	-
*	*	0	Q	$\overline{Q}$

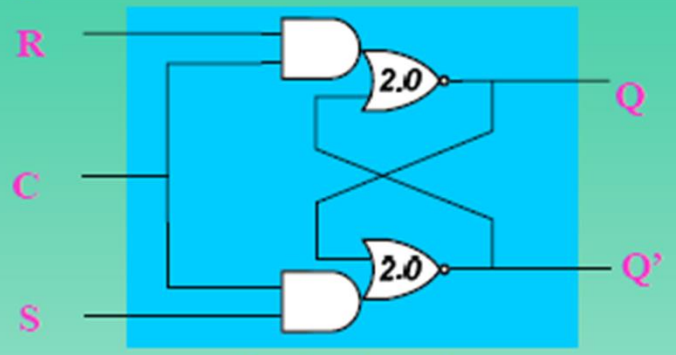
- Când semnalul C este activ valorile de la intrare sunt propagate prin latch
- Semnalele de intrare nu trebuie să se modifice în intervalul t_{setup} și t_{hold} al frontului descrescător

Gated SR-Latch

•Control signal C activates the latch



Graphic symbol

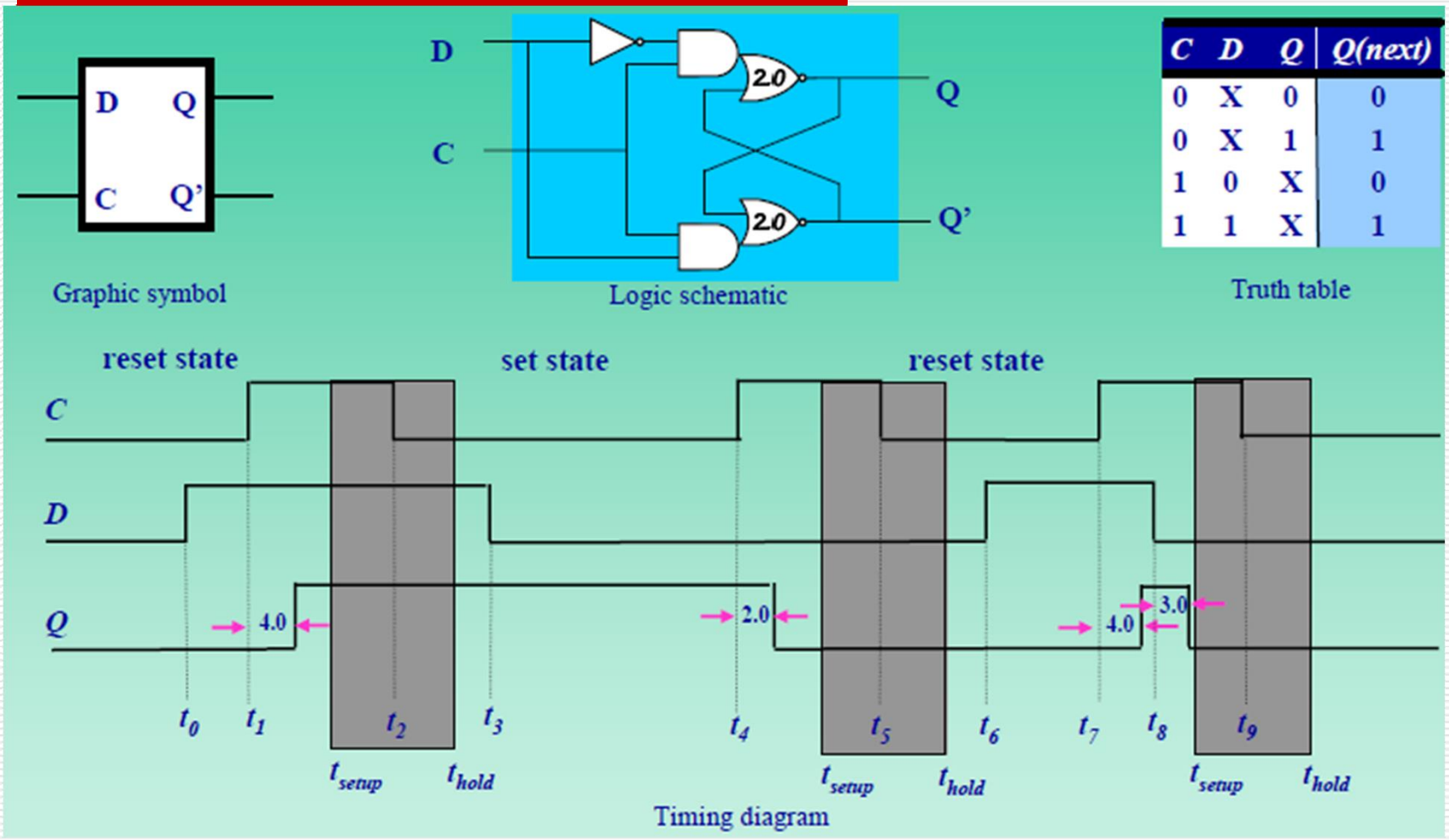


Logic schematic

C	S	R	Q	Q(next)	
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

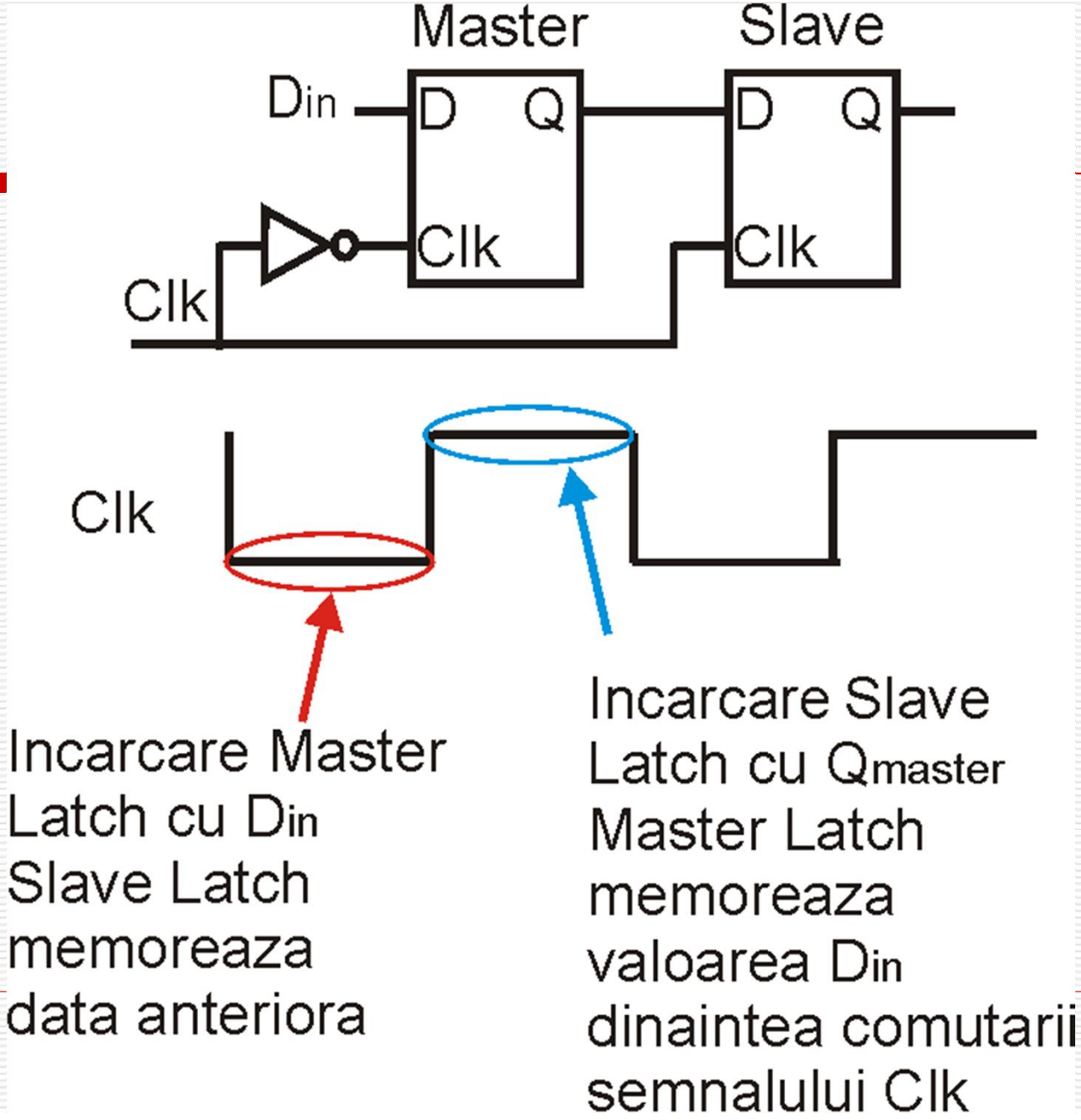
Gated D-latch



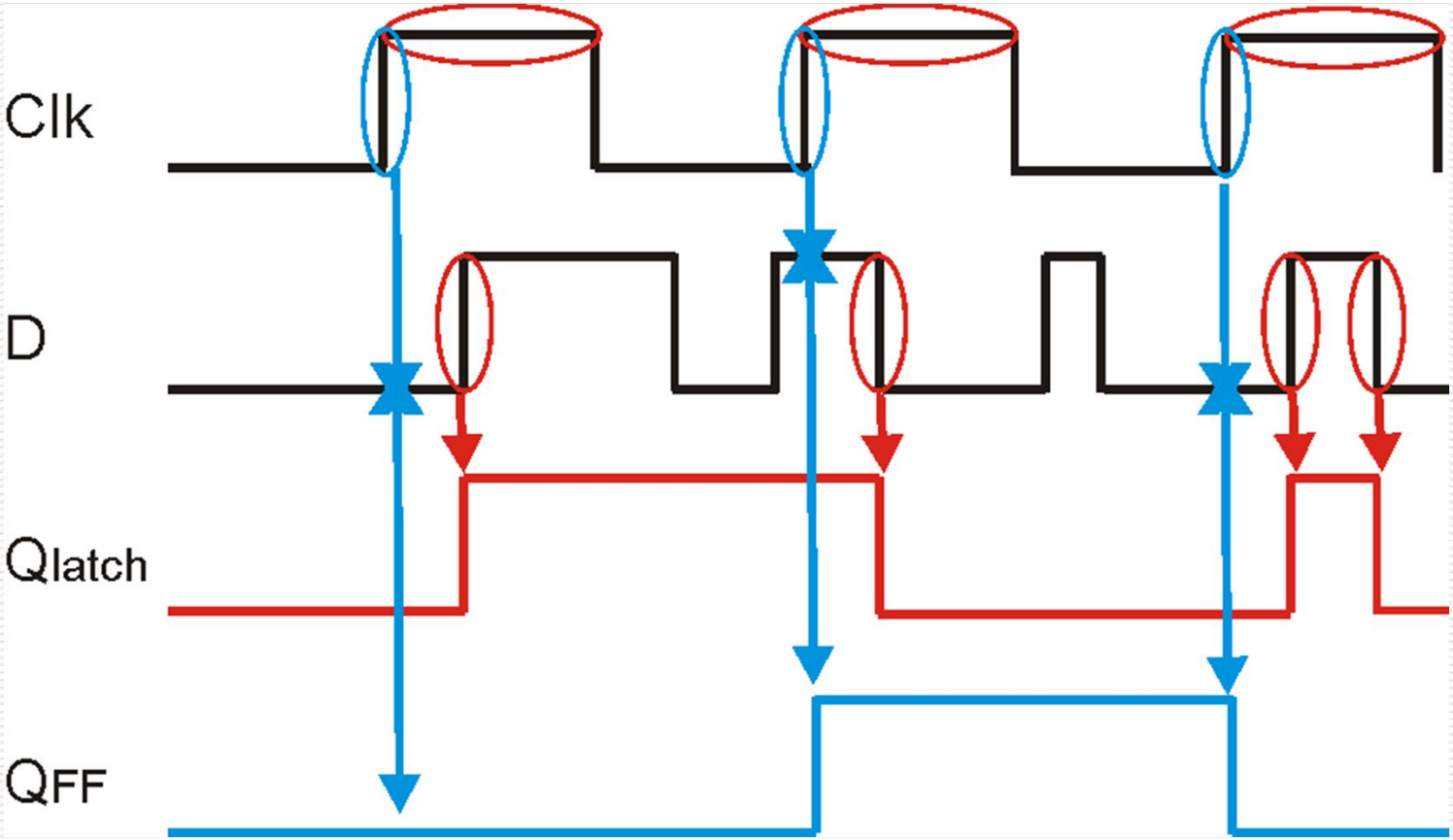
Flip-flop-uri

- 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



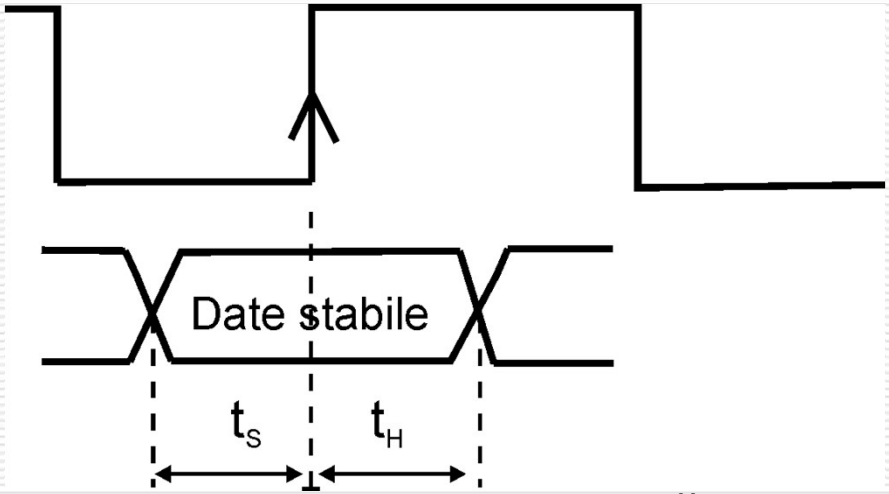
Latch sincron vs. FF sincron



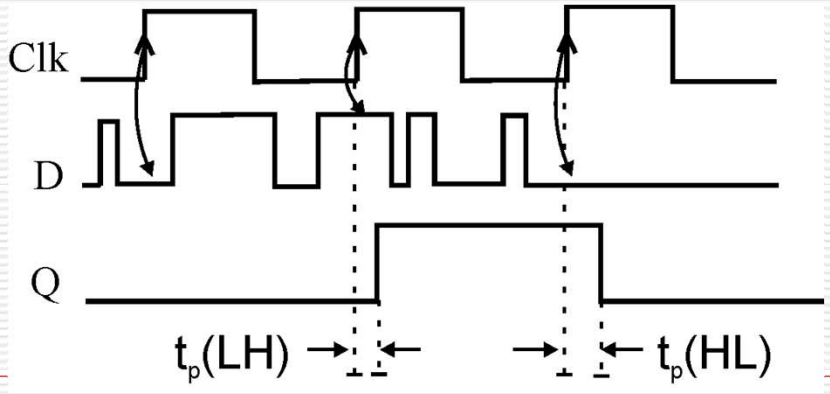
FF-uri

- 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 apariţia frontului semnalului de tact;
  - **Timpul de hold:** reprezintă timpul în care datele de intrare nu pot fi modificate după apariţia frontului semnalului de tact în vederea încărcării corecte a acestora;
  - **Timpul aferent întârzirii datorate propagării (tp):** constituie timpul necesar basculării FF-ului (clock to Q delay);

Prezentarea celor 3 timpi de propagare



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



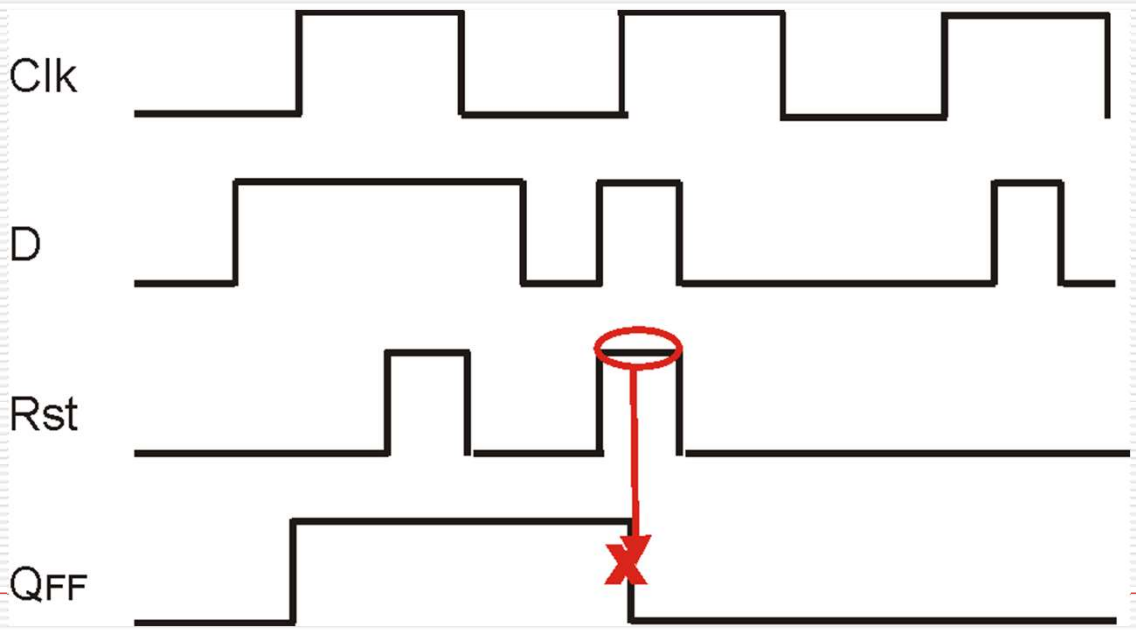
Timpi de propagare t_p

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

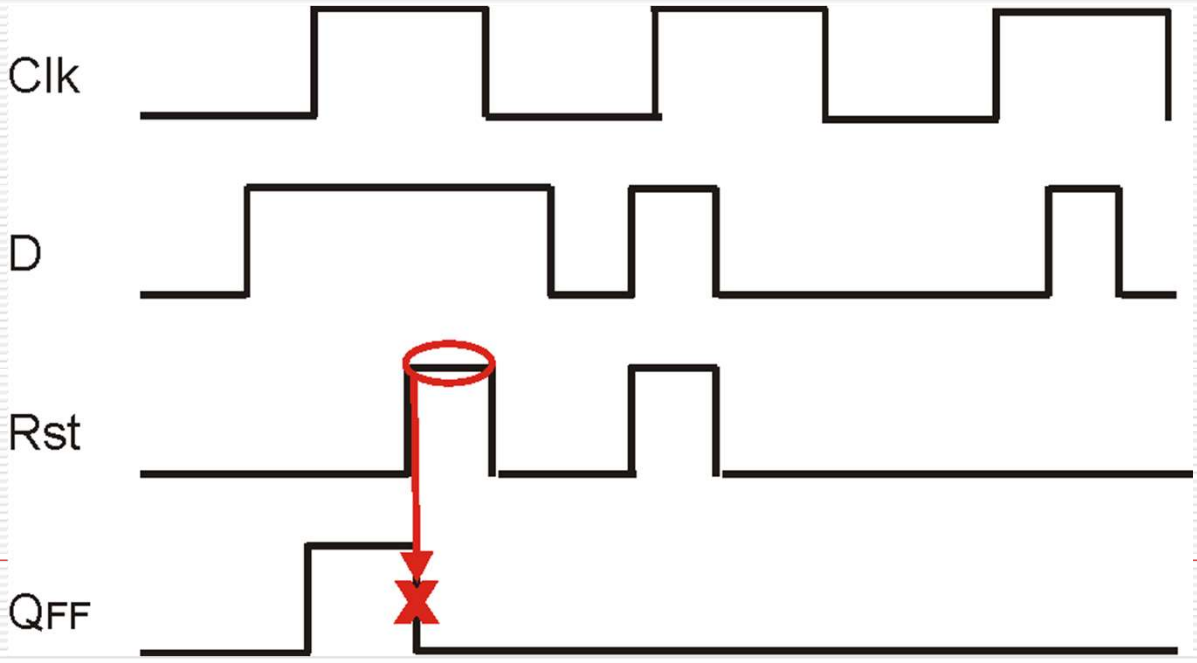
Operatia de reset a elementelor secventiale

- 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

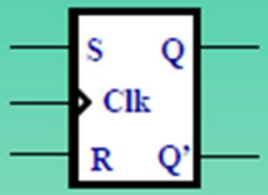
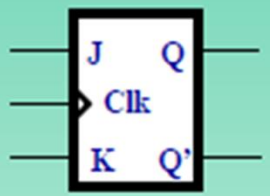
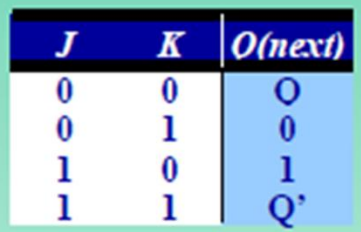
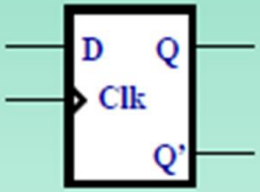
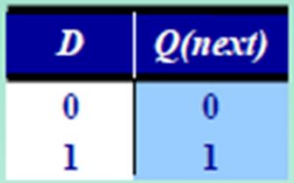
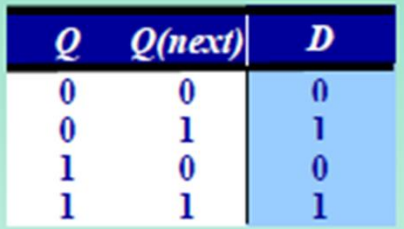
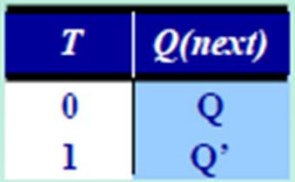
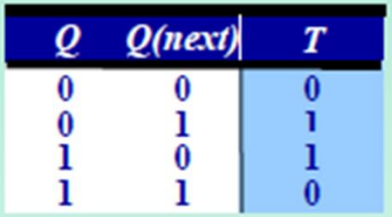
```
always
    @(posedge clk)
begin
     if (rst)
         begin
             q <= 0;
         end
     else
         begin
             q <= d;
         end
end
```

```
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

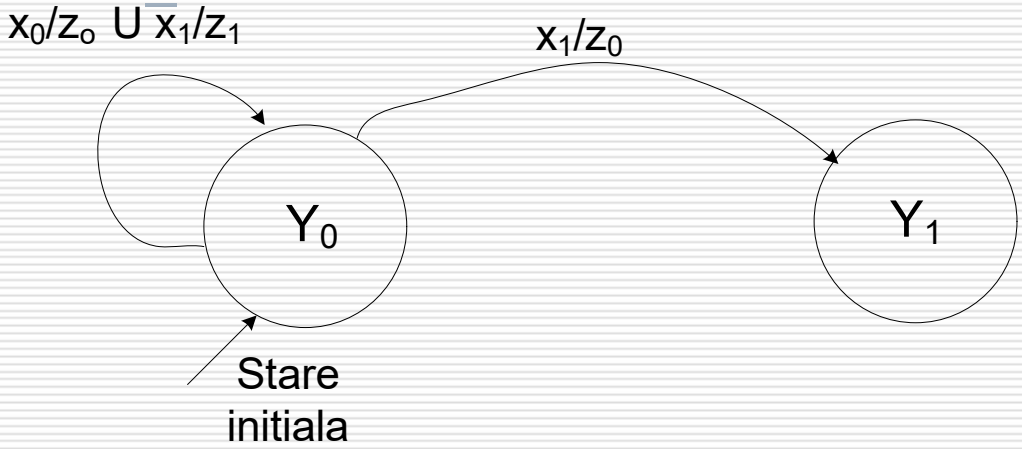
Flip-flop name	Flip-flop symbol	Characteristic table	Characteristic equation	Excitation table
SR			$Q(next)=S+R'Q$ $SR=0$	
JK			$Q(next)=JQ'+K'Q$	
D			$Q(next)=D$	
T			$Q(next)=TQ'+T'Q$	

Circuite secvențiale 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 de stare

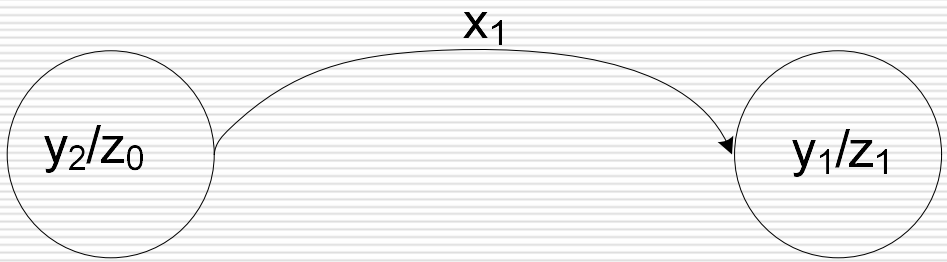
- **circuite de tip Mealy:**
  - fiecare nod se notează cu simbolul stării pe care o reprezintă,
  - arcul care pleacă din nod se notează cu perechea intrarea care a generat tranziţia circuitului/ ieşirea generată în timpul tranziţiei.



Starea iniţială se marchează printr-o săgeată aplicată nodului respectiv.

Circuite secvențiale: diagrame de stare

- **circuite de tip Moore:**
  - nodurile diagramei de stări se notează simbolul stării corespondente şi ieşirile
  - arcul are notată intrarea care a generat tranziția.



Registre

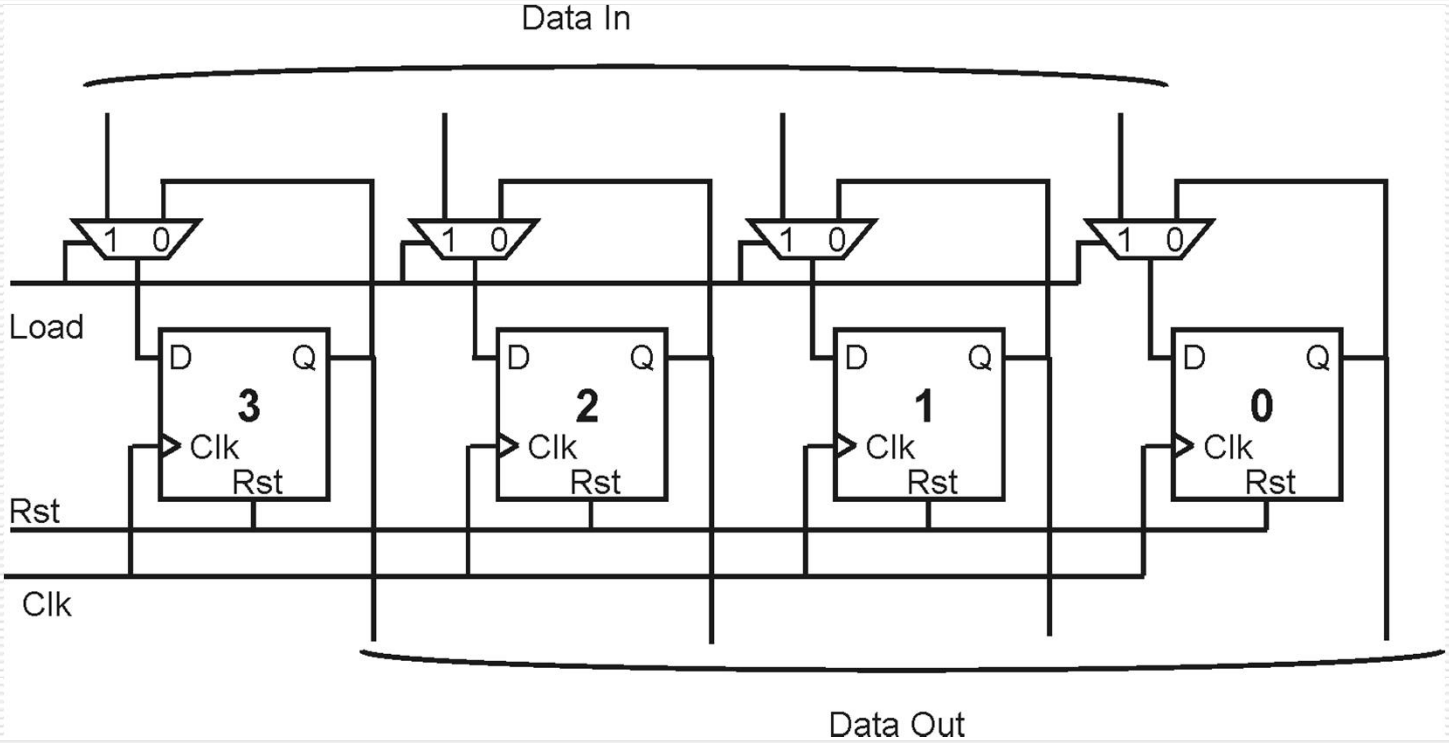
- Reprezinta o colectie/grupare de *n* bistabile
- Nr maxim de valori a unui registru pe *n* biti – 2^n valori binare
- Folosit pentru memorarea unui cuvant de date/unei stari curente a sistemului

Registru cu intrare paralela – iesire paralela

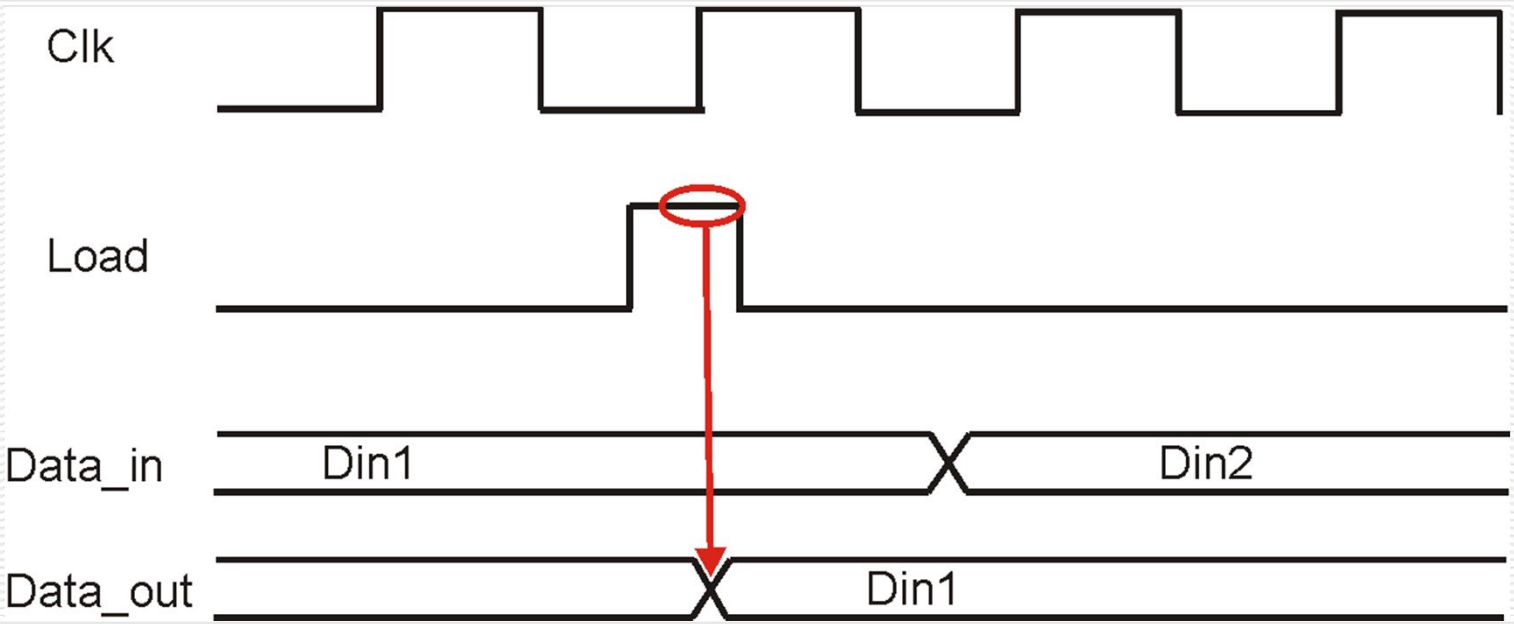
- Prezinta semnale de incarcare (*Load*)
- La fiecare front crescator valoarea registrului se actualizeaza cu *Data In*, daca este activ semnalul de *Load*

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

Registru cu intrare paralela – iesire paralela



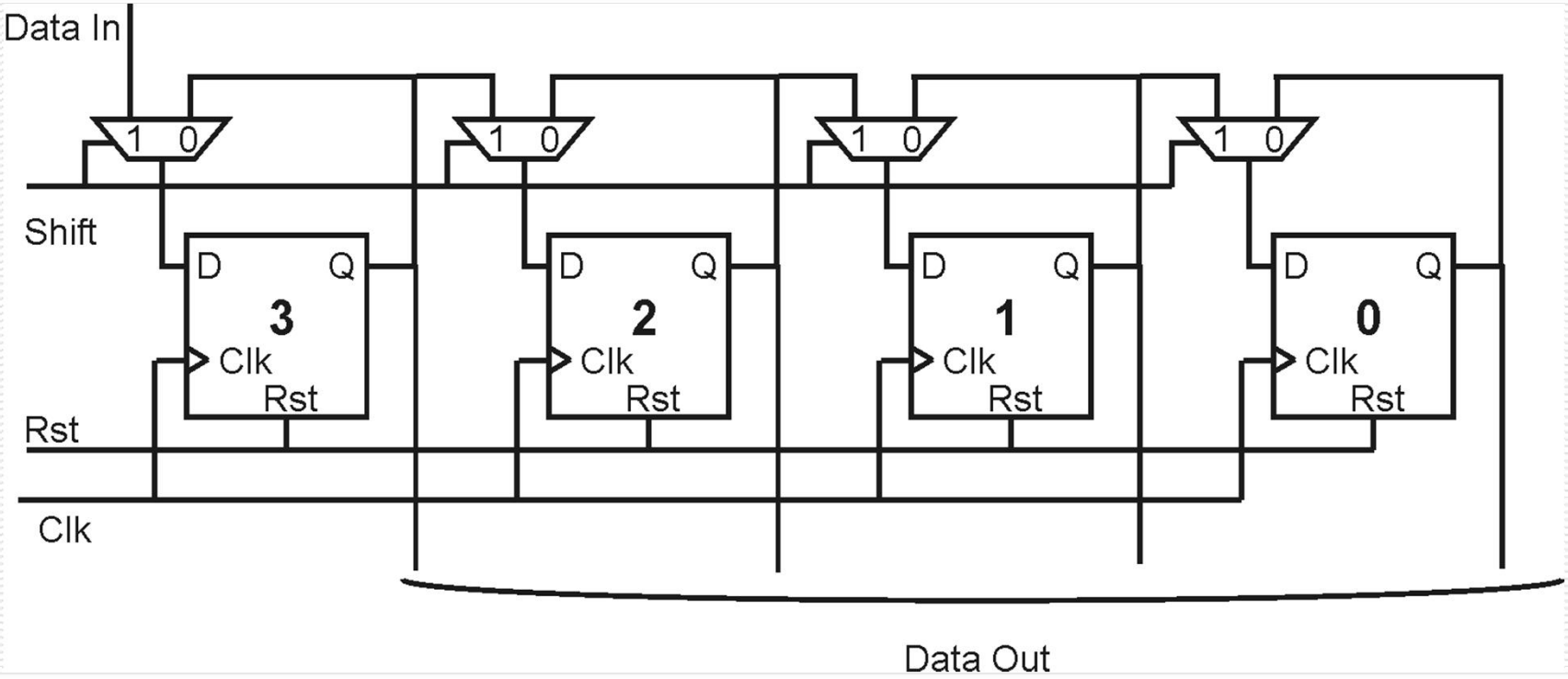
Registru cu intrare paralela – iesire paralela



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* ciclii de clock

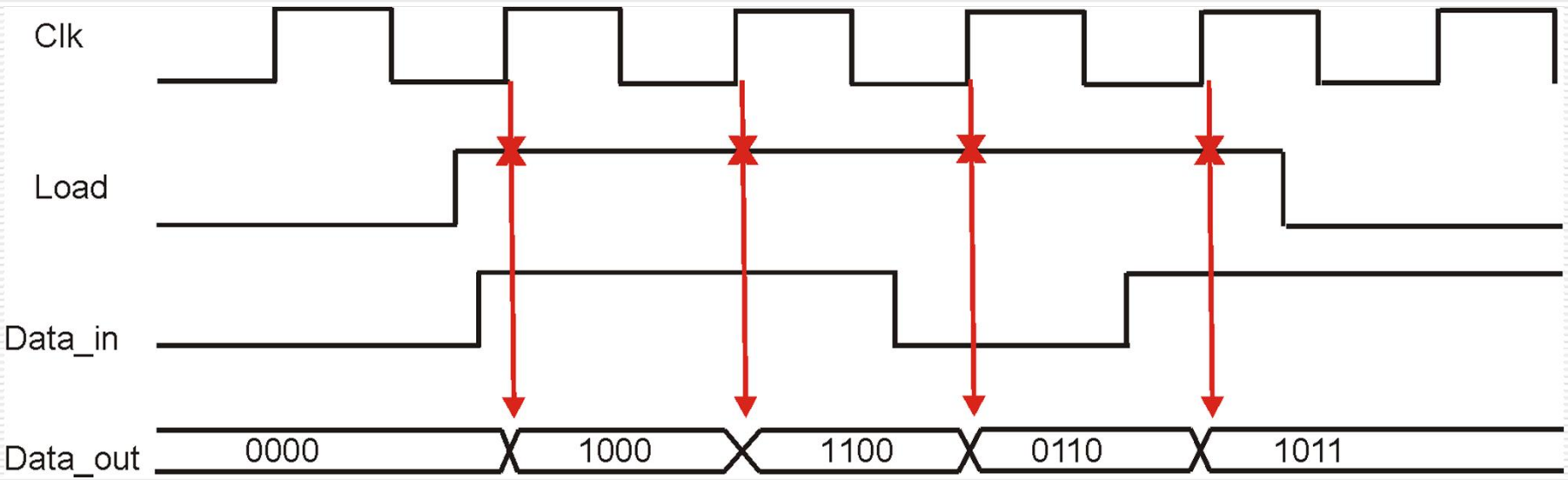
Registru cu intrare seriala – iesire paralela



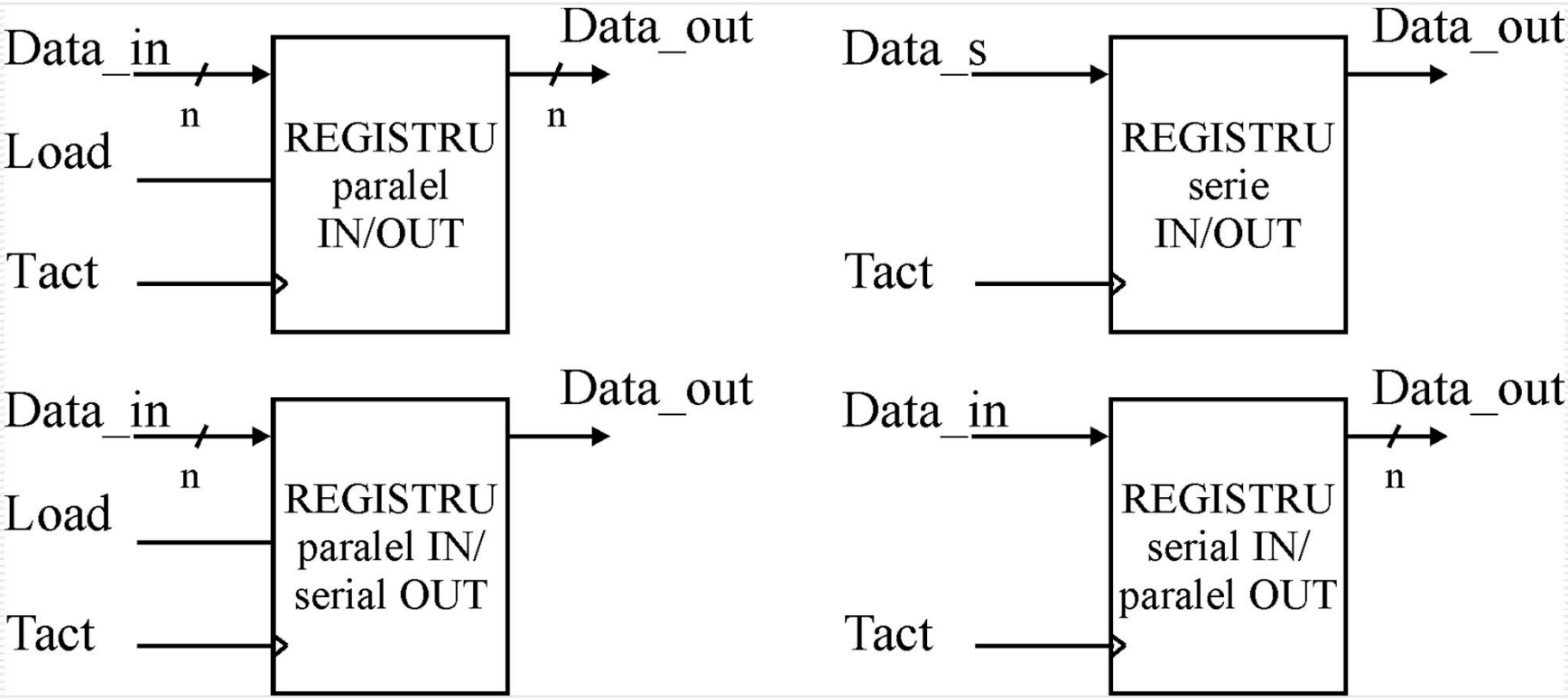
Registru cu intrare seriala – iesire paralela

Shift	Starea Curenta	Starea viitoare
0	$Q_3Q_2Q_1Q_0$	$Q_3Q_2Q_1Q_0$ (nu se schimba)
1	$Q_3Q_2Q_1Q_0$	$DataIn Q_3Q_2Q_1$

Registru cu intrare seriala – iesire paralela



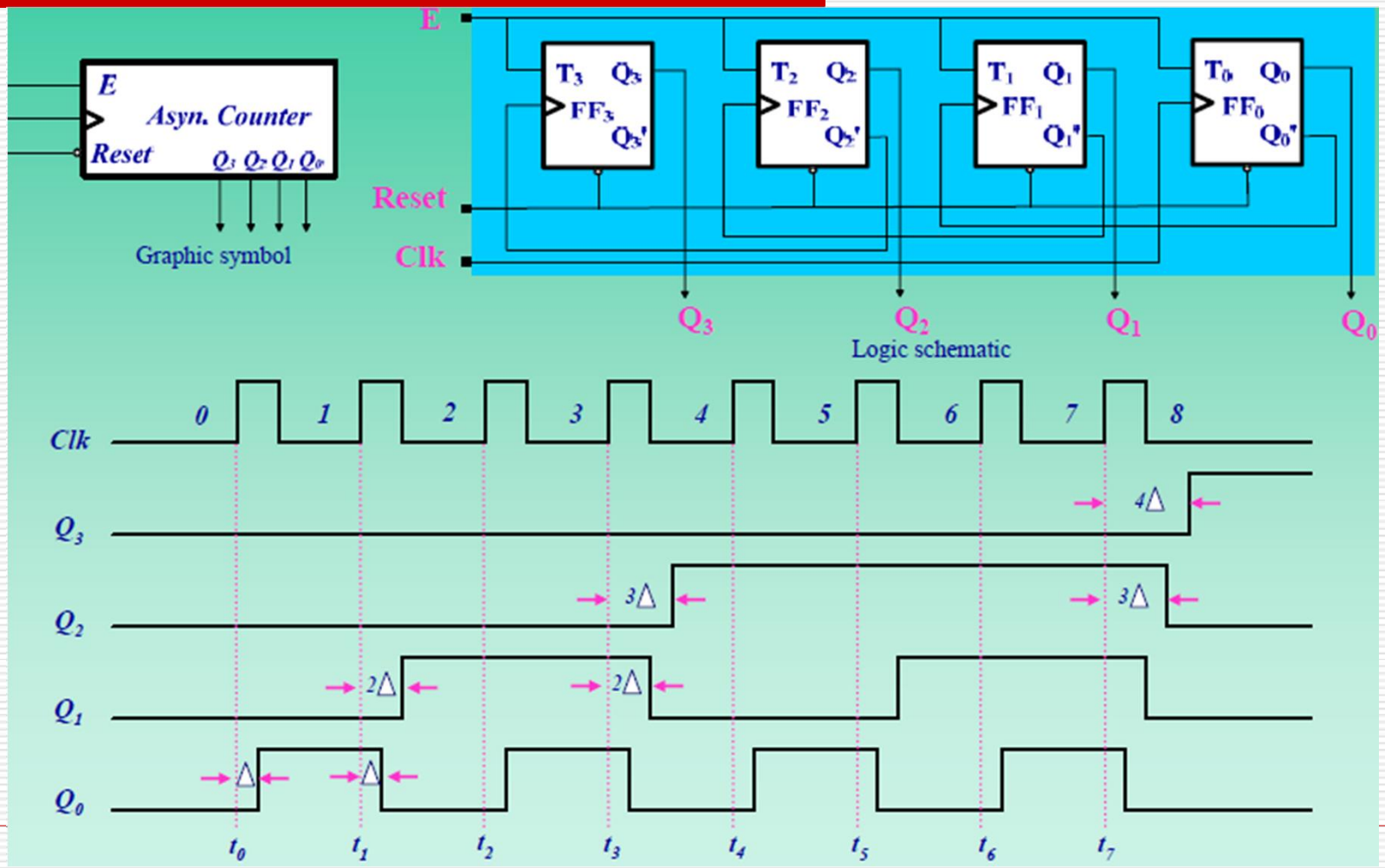
Registre - clasificare



Numărătoare

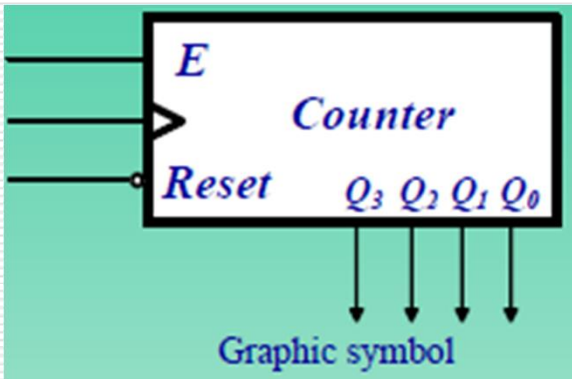
- circuite **secvenţiale sincrone autonome** (mulţimea intrărilor de DATE vidă), care baleiază o secvenţă de stări impuse de proiectant.
- de regulă este inițializat cu starea ,,0” , după care la fiecare impuls de numărare, comuta într-o nouă stare.
- caracterul **asincron** al unui numărător este dat de faptul că impulsul de tact “nu comandă” simultan toate bistabilele numărătorului din modului in care se genereaza 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



Numarator sincron

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



Graphic symbol

E	Operations
0	No change
1	Count

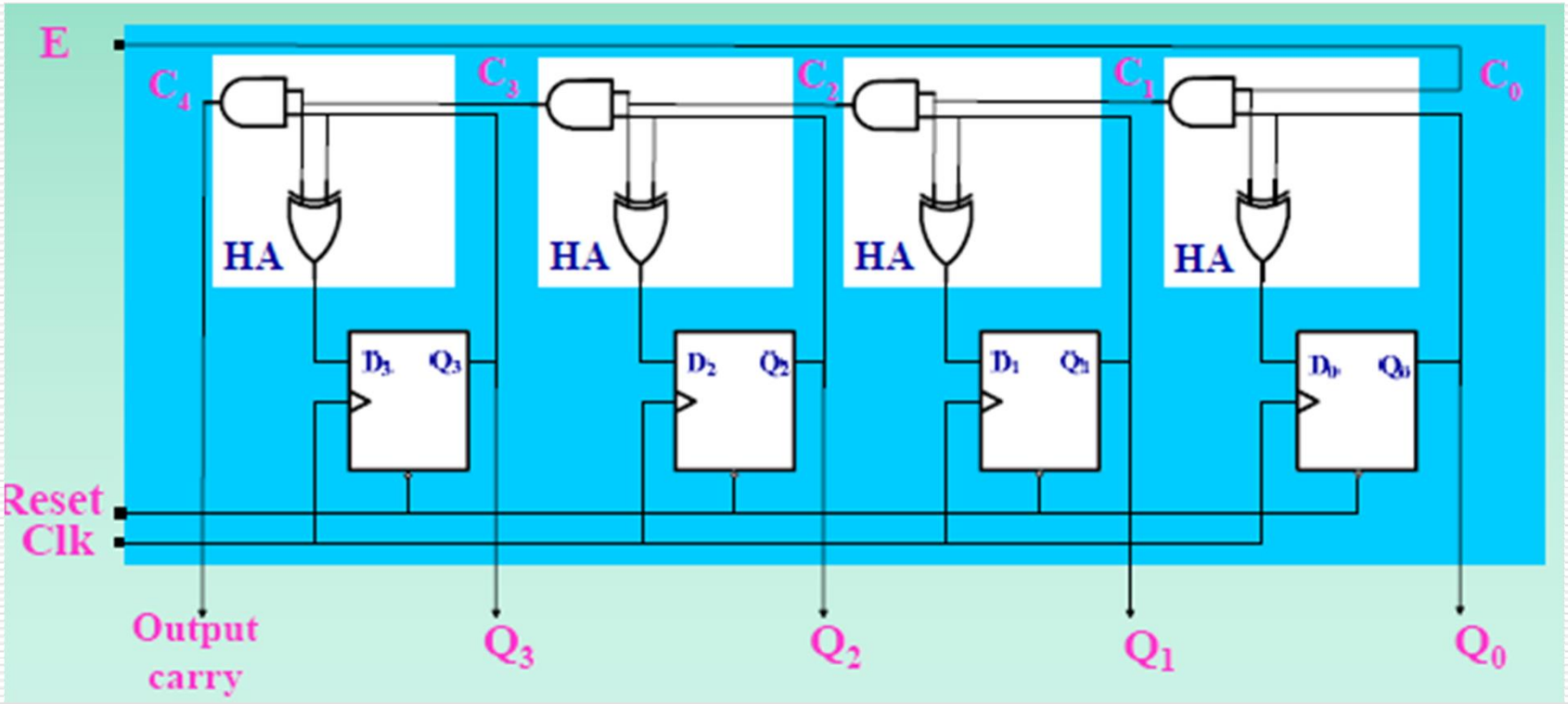
Operation table

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

Q_i	C_i	C_{i+1}	D_i
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

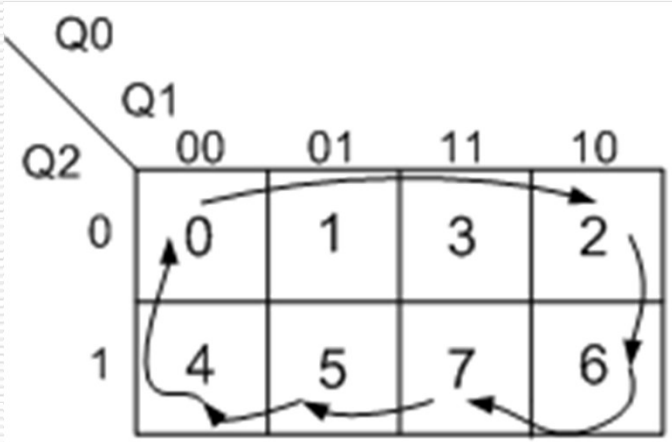
HA truth table

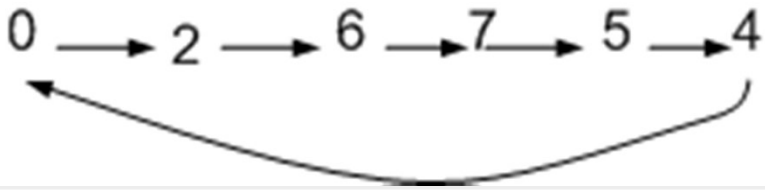
Numarator sincron



Aplicație 1*:

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





*Exemplu preluat din curs LD 2013 – Adrian Mihailescu

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	d
0	1	1	d
1	0	d	1
1	1	d	0

Aplicație 1:

- Indicații:

1. construiți tabelul de adevăr pentru determinarea expresiei intrărilot 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_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

Aplicație 1:

- Indicații:

1. construiți tabelul de adevăr pentru determinarea expresiei intrărilot 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_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						

Aplicație 1:

- Indicații:

3. Minimizăm funcţiile Ji(Q0,Q1,Q2) Ki(Q0,Q1,Q2) . Stările prin care nu trece numărătoul sunt notate cu “don’t care”

$$J_0 = Q_2 Q_1$$

$$K_0 = \overline{Q_1}$$

Aplicație 1:

$J_1 = \overline{Q_2}$

$K_1 = Q_0$

$J_2 = Q_1$

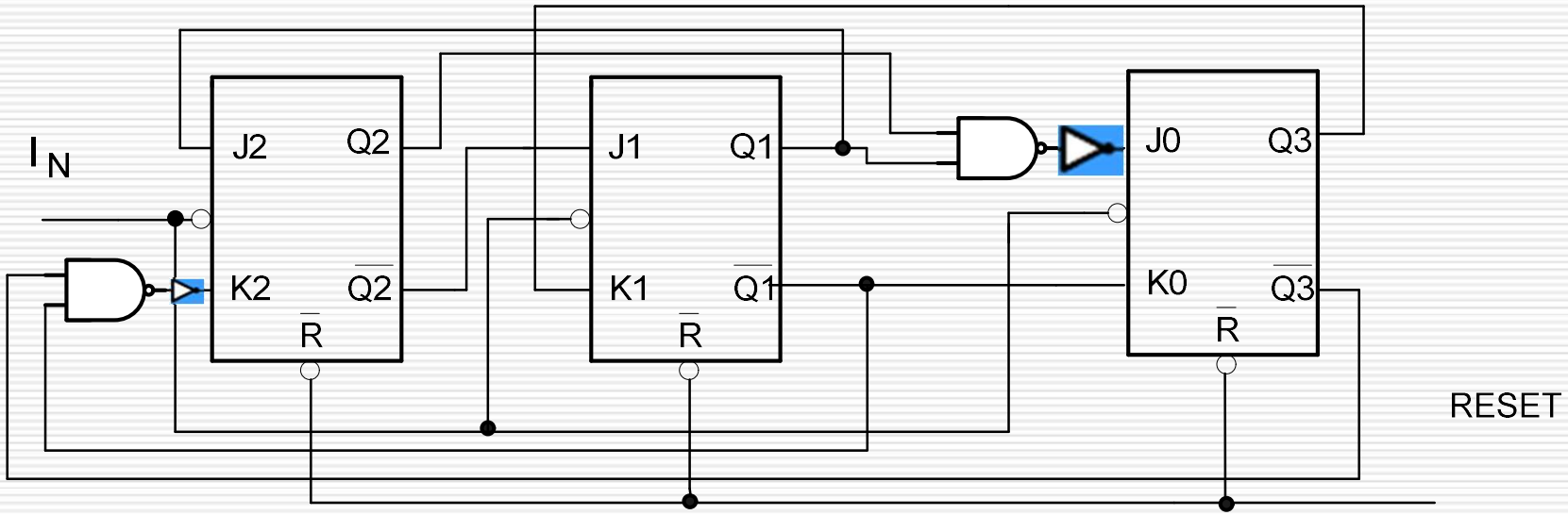
$K_2 = \overline{Q_0}\,\overline{Q_1}$

Aplicație 1:

4. Implementare cu FF-uri J-K M-S şi porţi logice ŞI:

Aplicație 1:

4. Implementare cu FF-uri J-K M-S şi porţi logice ŞI-NU:



Întrebări?

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