

Logică digitală

-Curs 9-
Circuite logice
secvențiale: registre
numărătoare, pila de
registre

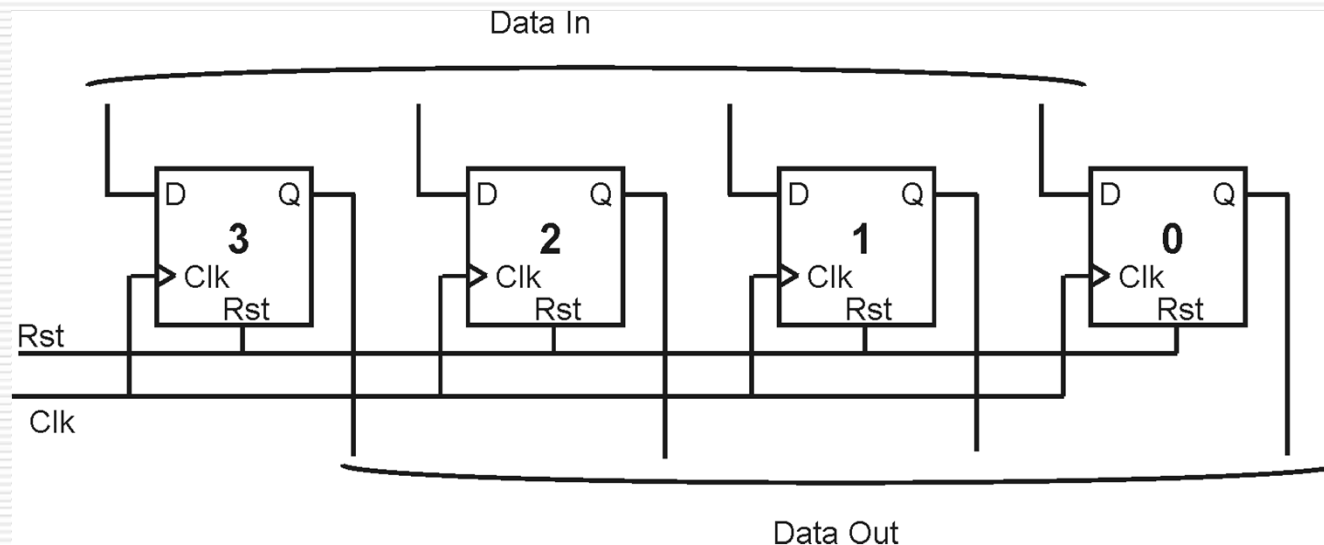
Outline

- ☐ Registre
 - Paralel
 - Serie
 - ☐ Numărătoare
 - Sincron
 - Asincron
 - ☐ Pila de registre
-

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 încărcare în paralel – iesire în paralel



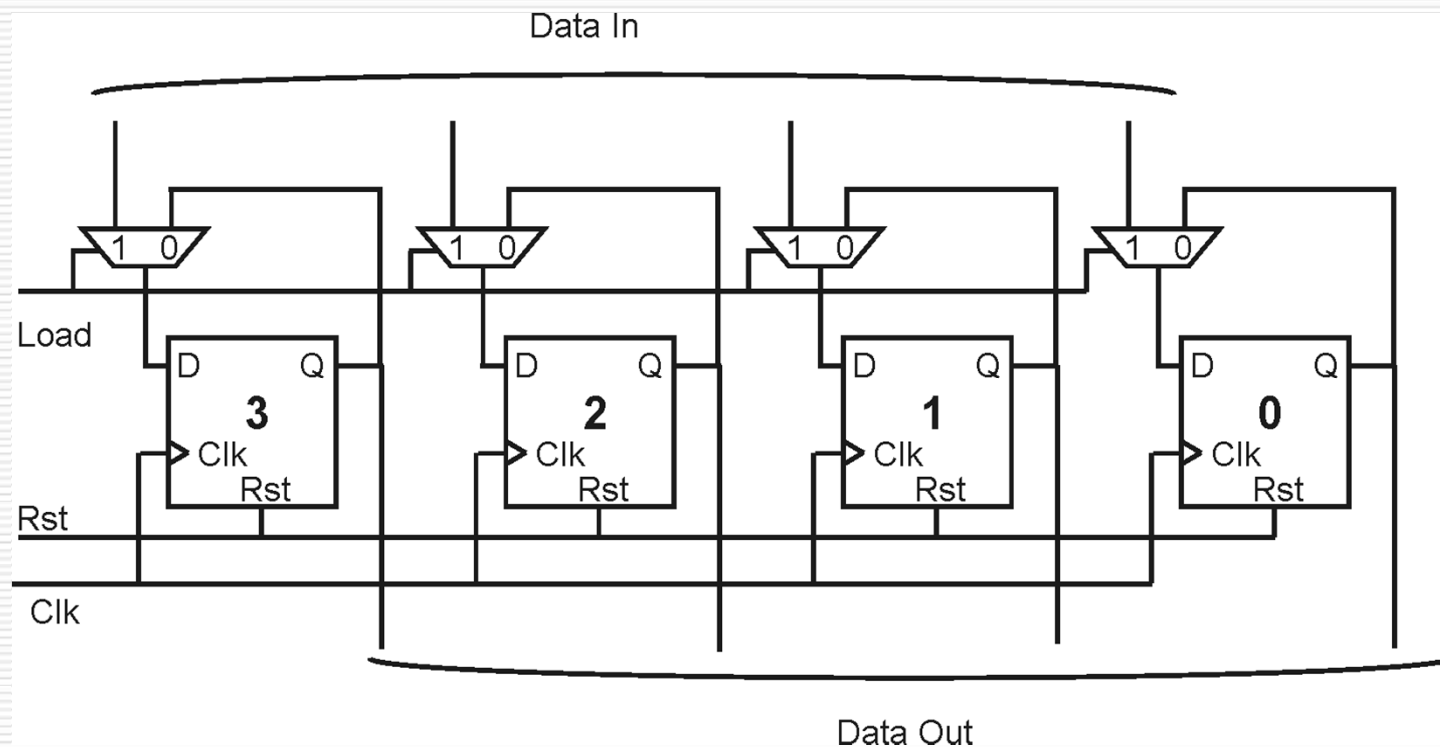
- ❑ La fiecare front crescator valoarea registrului se actualizeaza cu *Data In*
-

Registru cu încărcare în paralel – ieșire în paralel

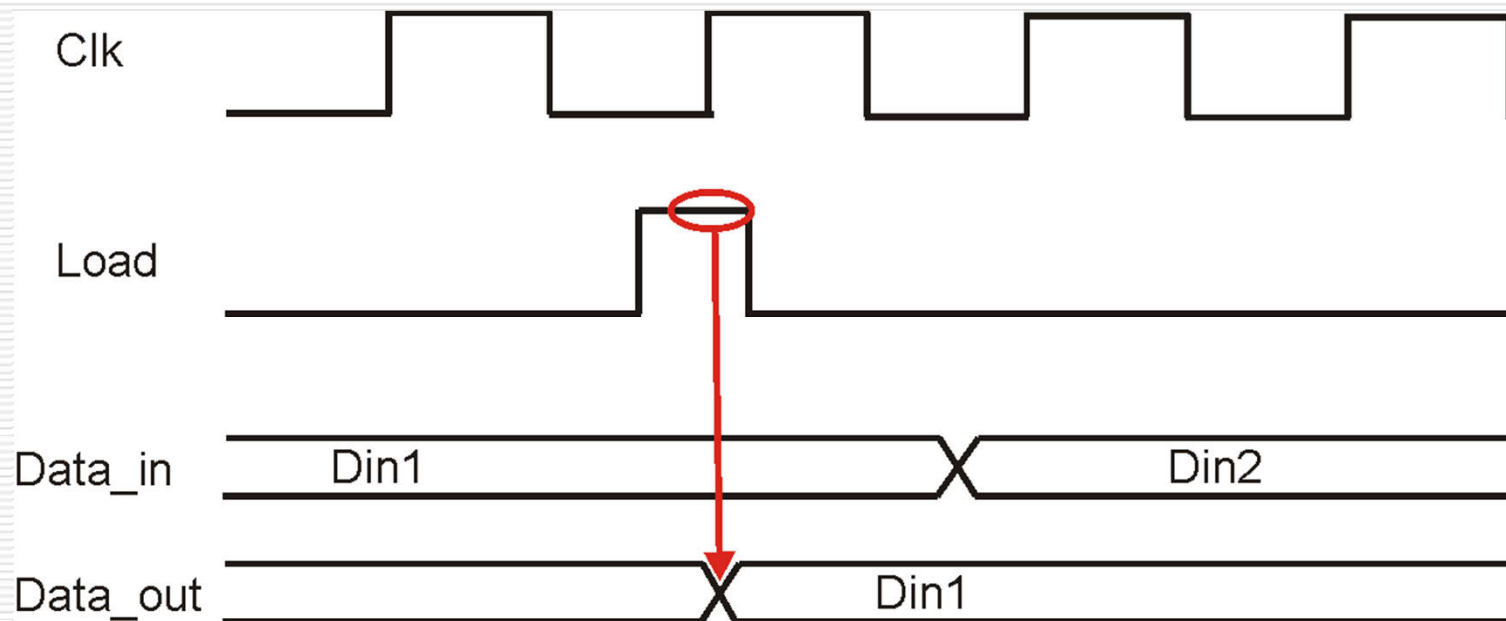
- ❑ Prezinta semnale de incarcare (*Load*)
- ❑ La fiecare front crescator valoarea registrului se actualizeaza cu *Data In*, daca este activ semnalul de *Load*

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

Registru cu încărcare paralelă – iesire paralelă



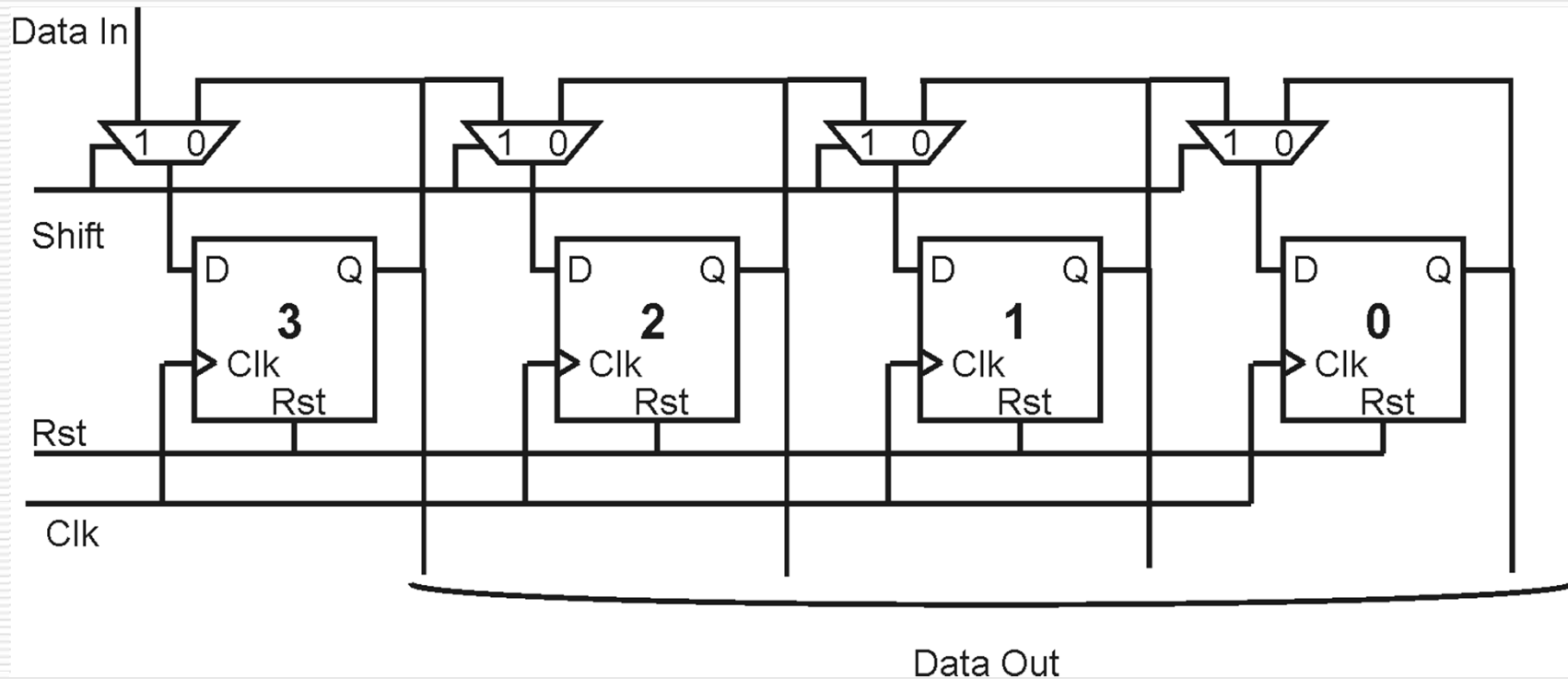
Registru cu încărcare în paralel – ieșire în paralel



Registru cu încărcare serie – ieșire în paralel

- ❑ Funcția de deplasare (shift-are) în interiorul registrului
 - ❑ Datele se introduc serial în registru – o singură intrare de date
 - ❑ La fiecare activare a semnalului de Load (Shift), datele se deplasează în cadrul registrului
 - ❑ Încărcarea a n biți necesită n cicluri de clock
-

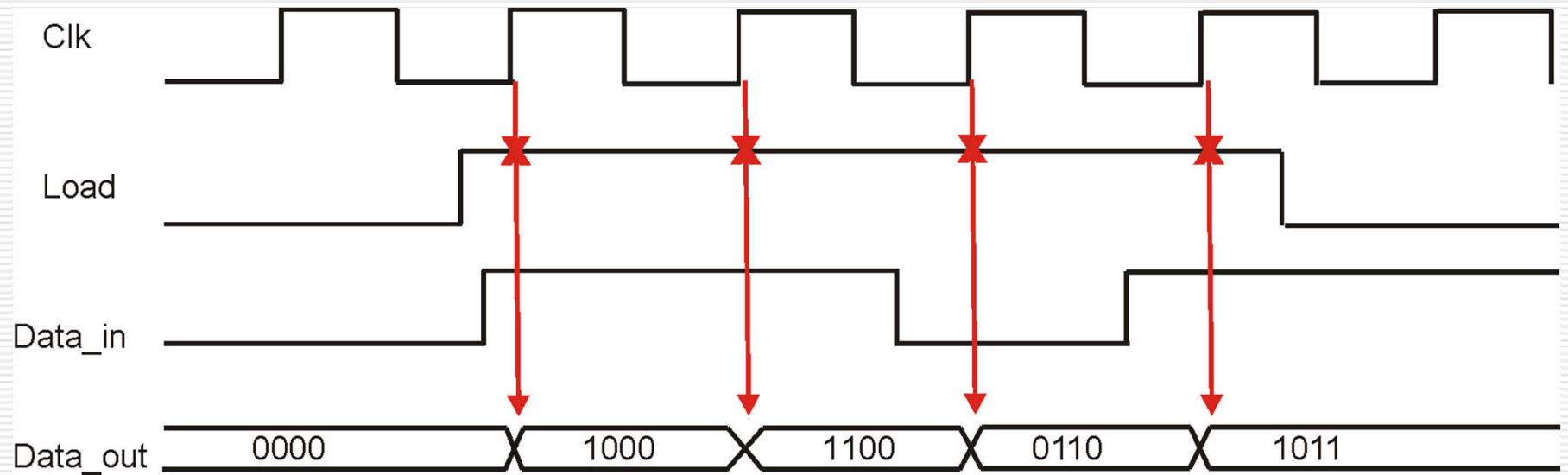
Registru cu încărcare în serie – ieșire în paralel



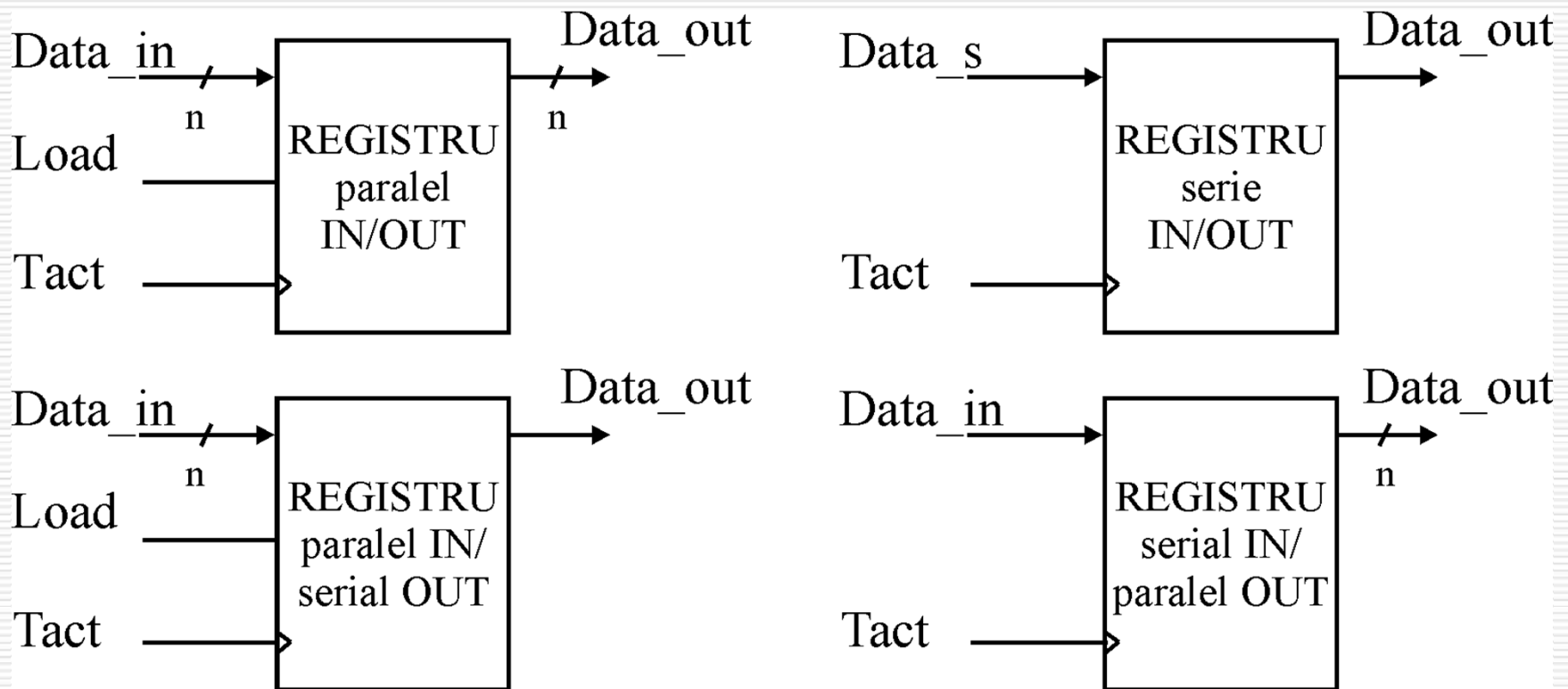
Registru cu încărcare serie – ieșire în paralel

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



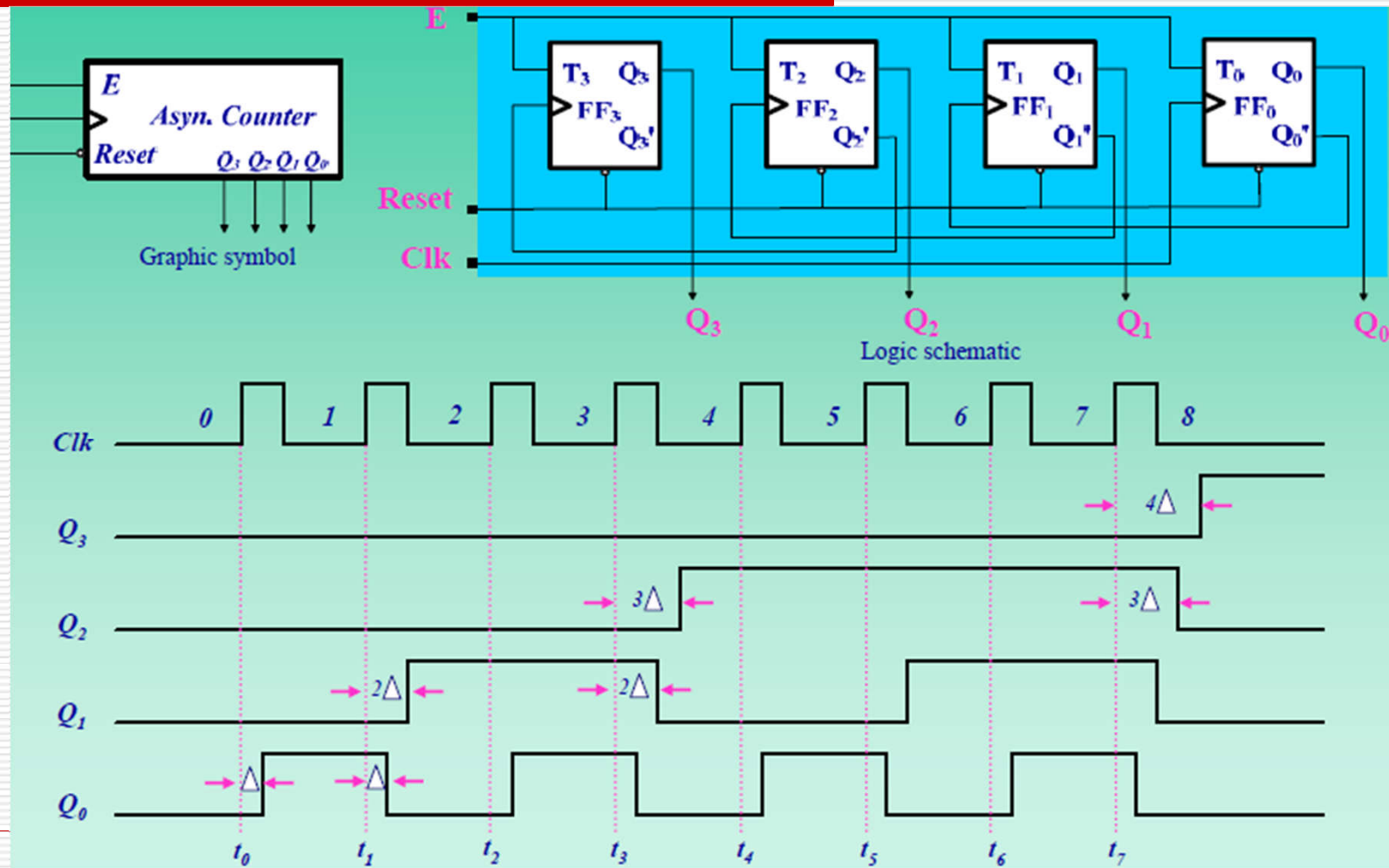
Registre - clasificare



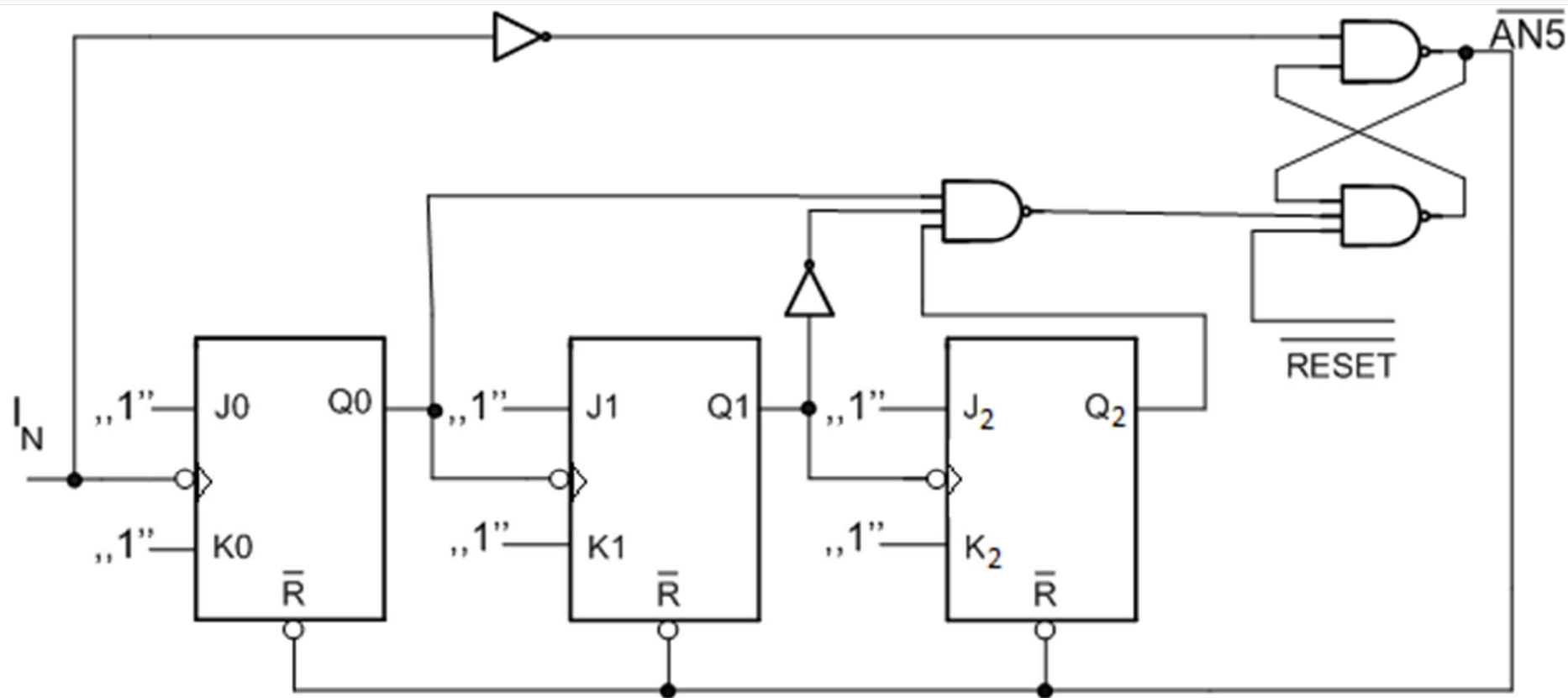
Numărătoare

- circuite **secvențiale sincrone** a caror diagrama de stare are un singur ciclu, si 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



Numărător asincron cu memorarea semnalului de anulare



Observație

- ❑ Dacă intrările $J=K=„1”$, bistabilul J-K are același comportament cu un bistabil T cu intrarea $T=„1”$;
 - ❑ Inițial prin activarea („0”) intrării asincrone de reset, numărătorul se aduce în starea 0 iar apoi sub comanda impulsului de numărare parcurge secvența de stări;
-

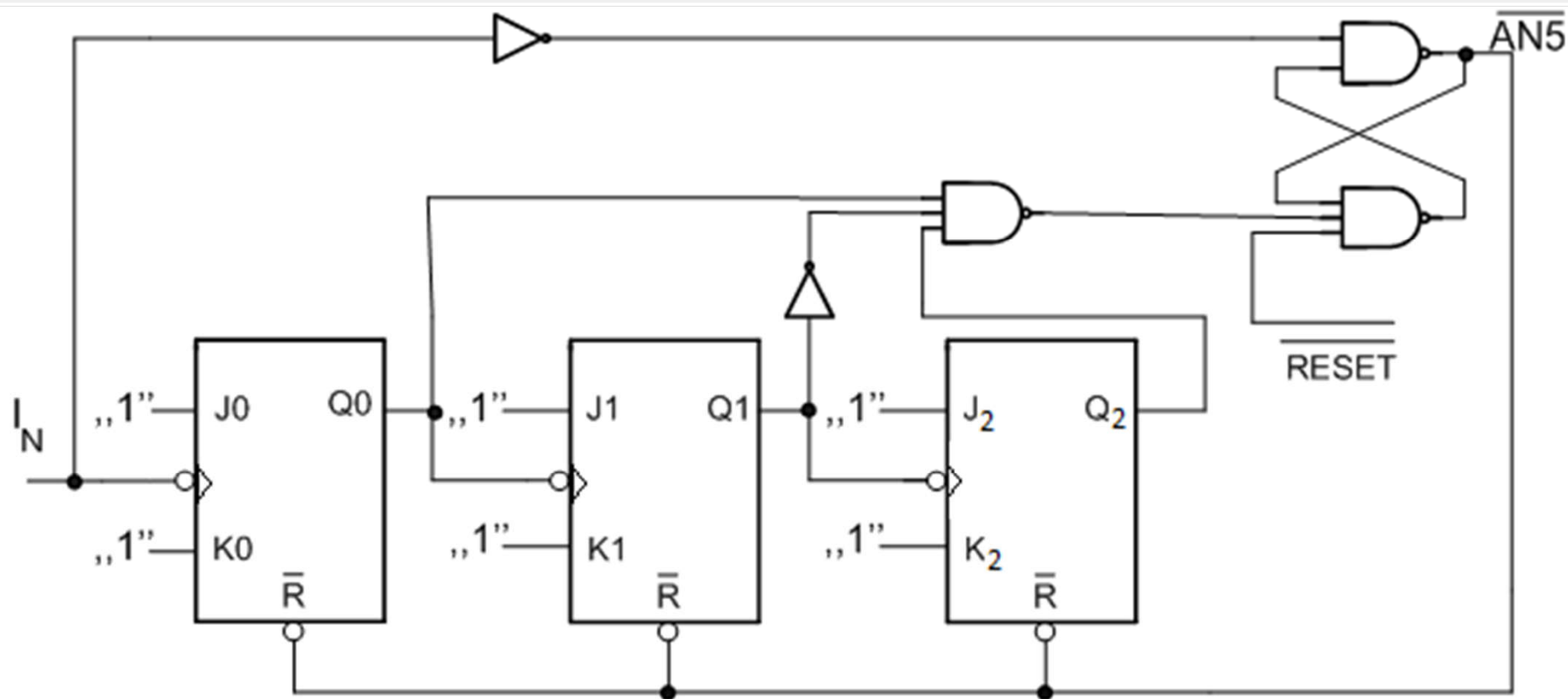
Determinați:

- ☐ Sensul de numărare și secvența de stări parcurse;
 - ☐ Elaborați diagrama de timp pentru o secvență completă de numărare.
-

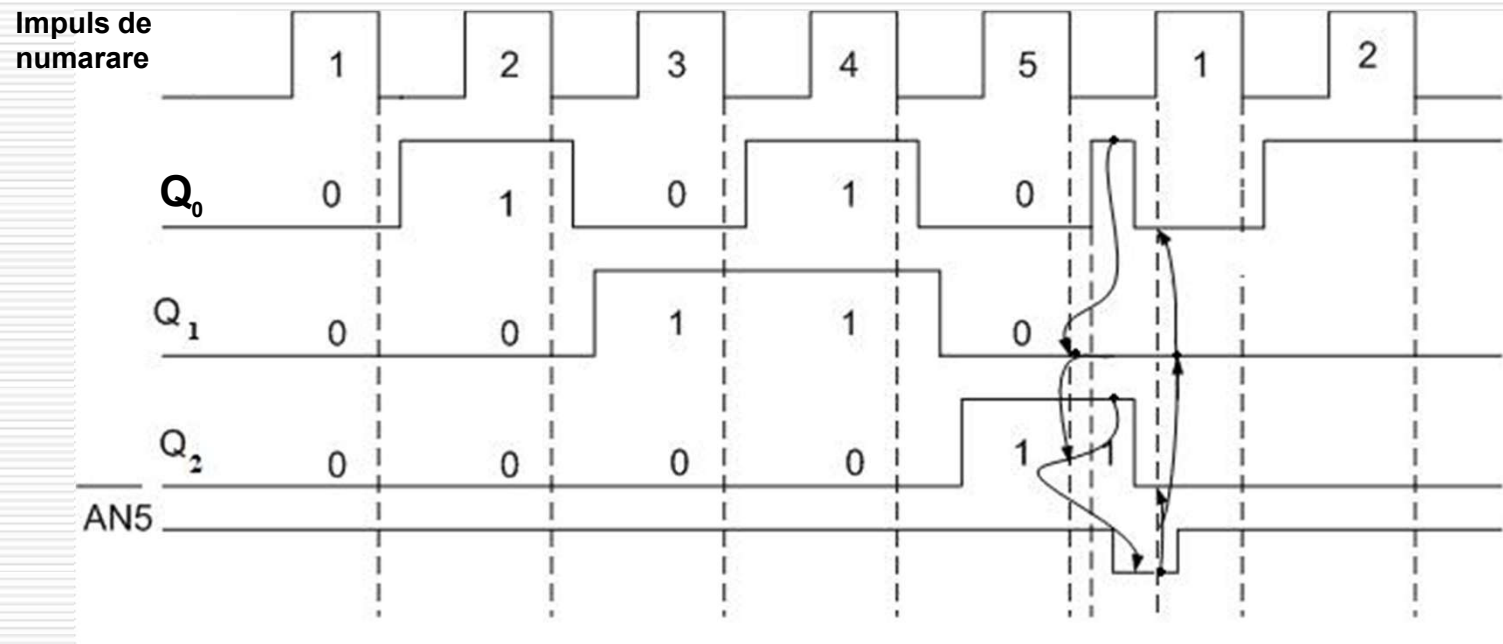
Numărător asincron crescător modulo 5

Starea	Q_2	Q_1	Q_0
	2^2	2^1	2^0
000	0	0	0
001	0	0	1
010	0	1	0
011	0	1	1
100	1	0	0
101	1	0	1
110	1	1	0
111	1	1	1

Numărător asincron cu memorarea semnalului de anulare



Numărător asincron crescător modulo 5 cu memorarea semnalului de anulare

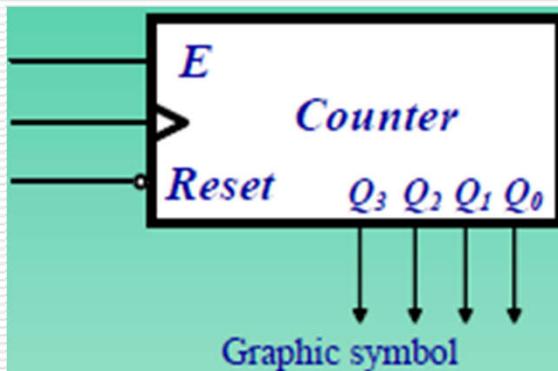


Numărătoare asincrone: sumar

- ❑ Impulsul de numărare nu comandă simultan toate bistabilele;
 - ❑ Dezavantaj: întârzierile introduse de bistabilele numărătorului pot genera impulsuri parazite la decodificare;
 - ❑ Dacă nu sunt filtrate aceste impulsuri pot genera comenzi eronate în sistem.
-

Numarator sincron

- Numărătoarele își increm/decrem conținutul când primesc semnal de activare/numărare (E)



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

Operation table

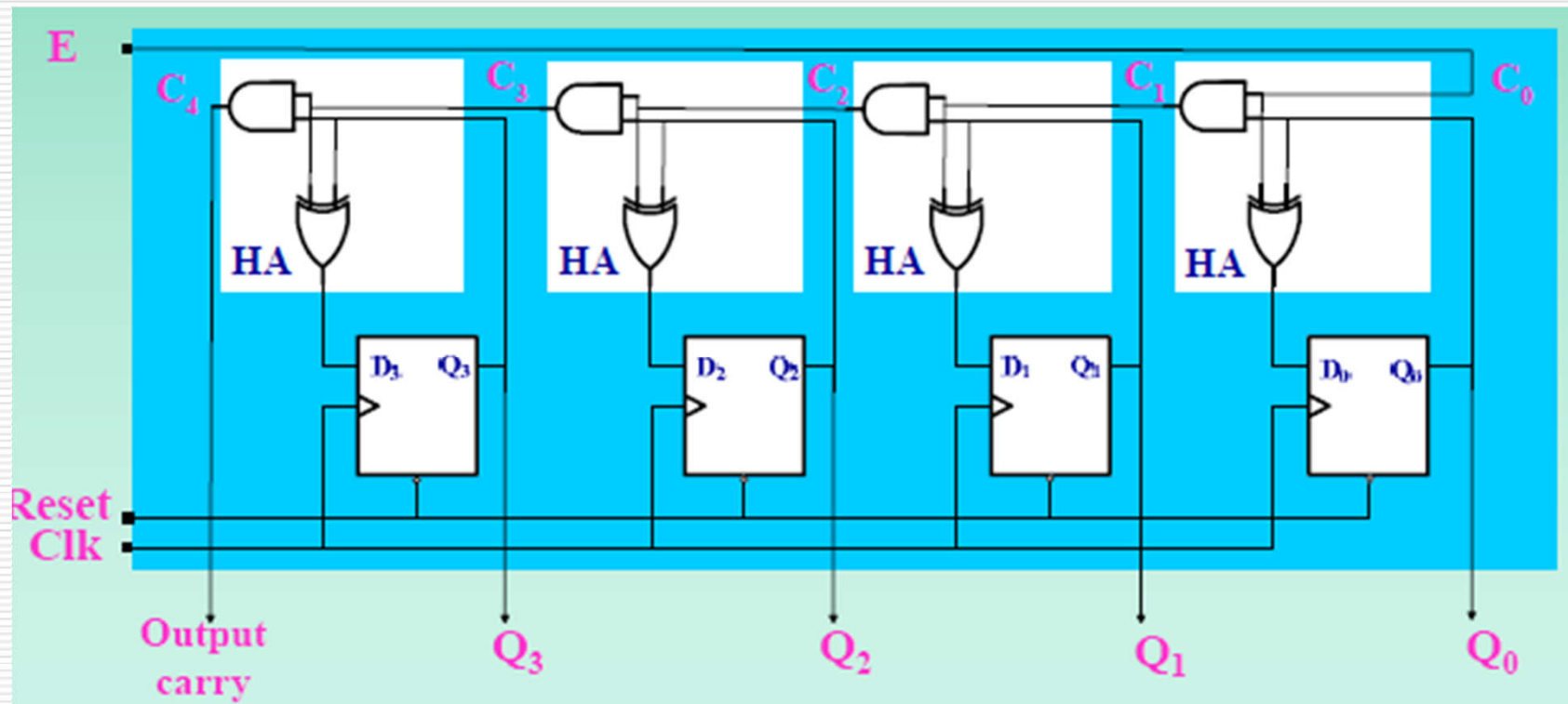
$$D_i = Q_i \oplus C_i$$

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

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

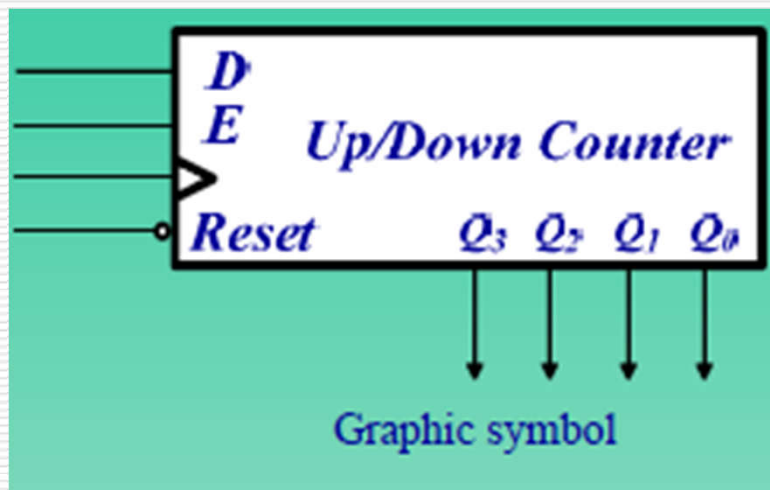
HA truth table

Numarator sincron



Numărător sincron în ambele sensuri

- Se obține din numărătorul anterior înlocuind HS cu half add/subtract cell



<i>E</i>	<i>D</i>	<i>Operations</i>
0	X	No change
1	0	Count up
1	1	Count down

Operation table

Numărător sincron în ambele sensuri

□ Tabel de adevăr:

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

HAS truth table

E	D	Operations
0	X	No change
1	0	Count up
1	1	Count down

Operation table

Numărător sincron în ambele sensuri

□ Tabel de adevăr:

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

HAS truth table

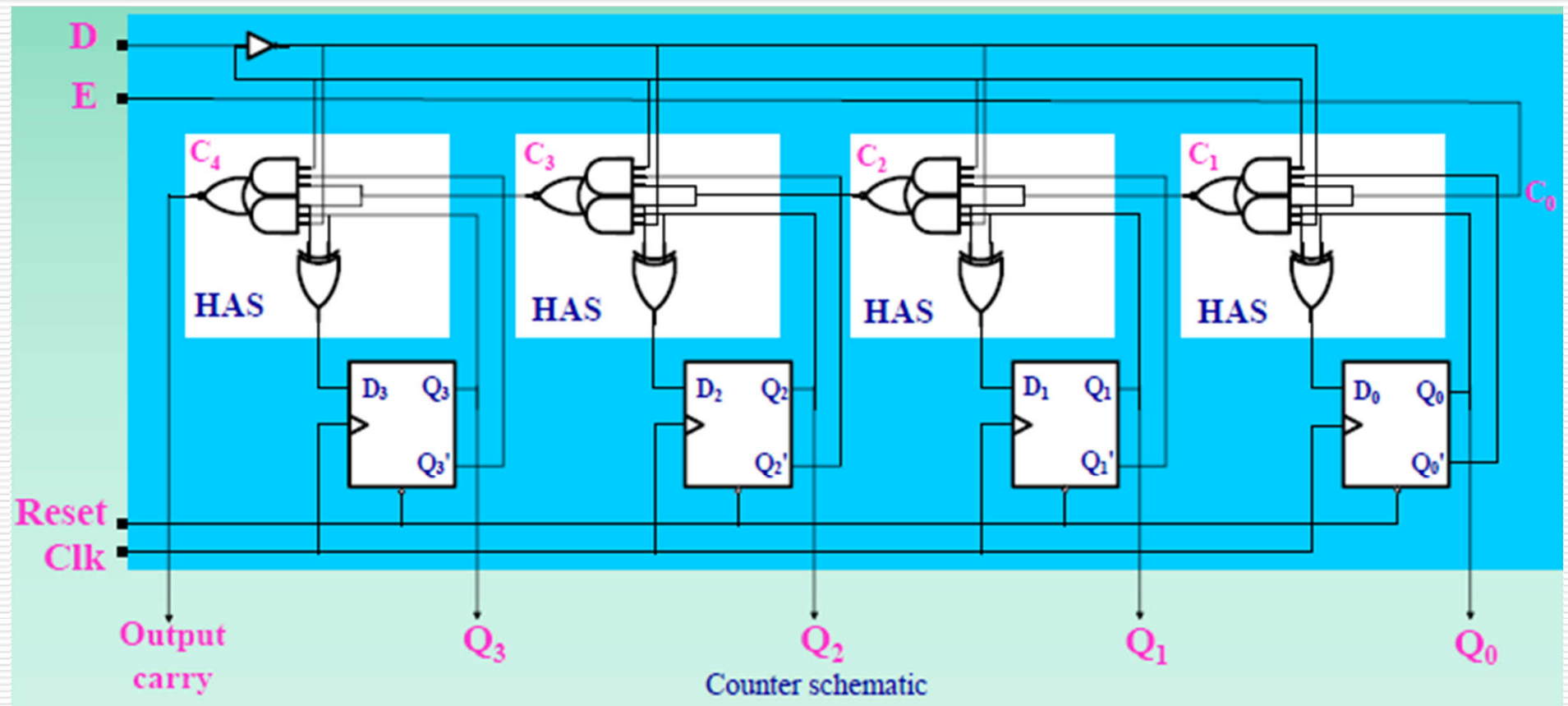
E	D	Operations
0	X	No change
1	0	Count up
1	1	Count down

Operation table

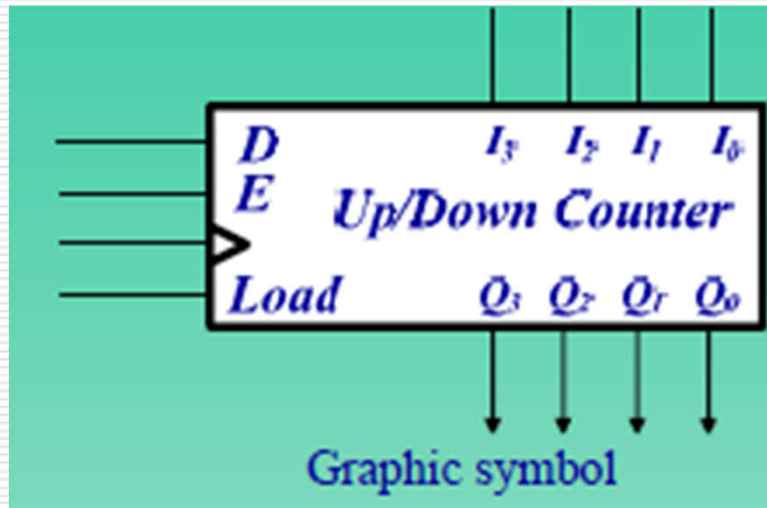
$$D_i = Q_i \oplus C_i$$

$$C_{i+1} = D'Q_iC_i + DQ_i'C_i$$

Numărător sincron în ambele sensuri



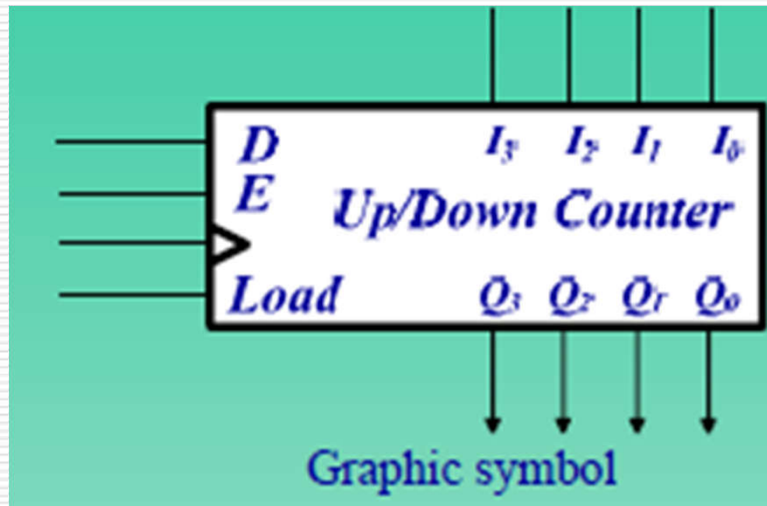
Numărător sincron în ambele sensuri cu o valoare de start încărcabilă



<i>Load</i>	<i>E</i>	<i>D</i>	<i>Operations</i>
0	0	X	No change
0	1	0	Count up
0	1	1	Count down
1	X	X	Load the input

Operation table

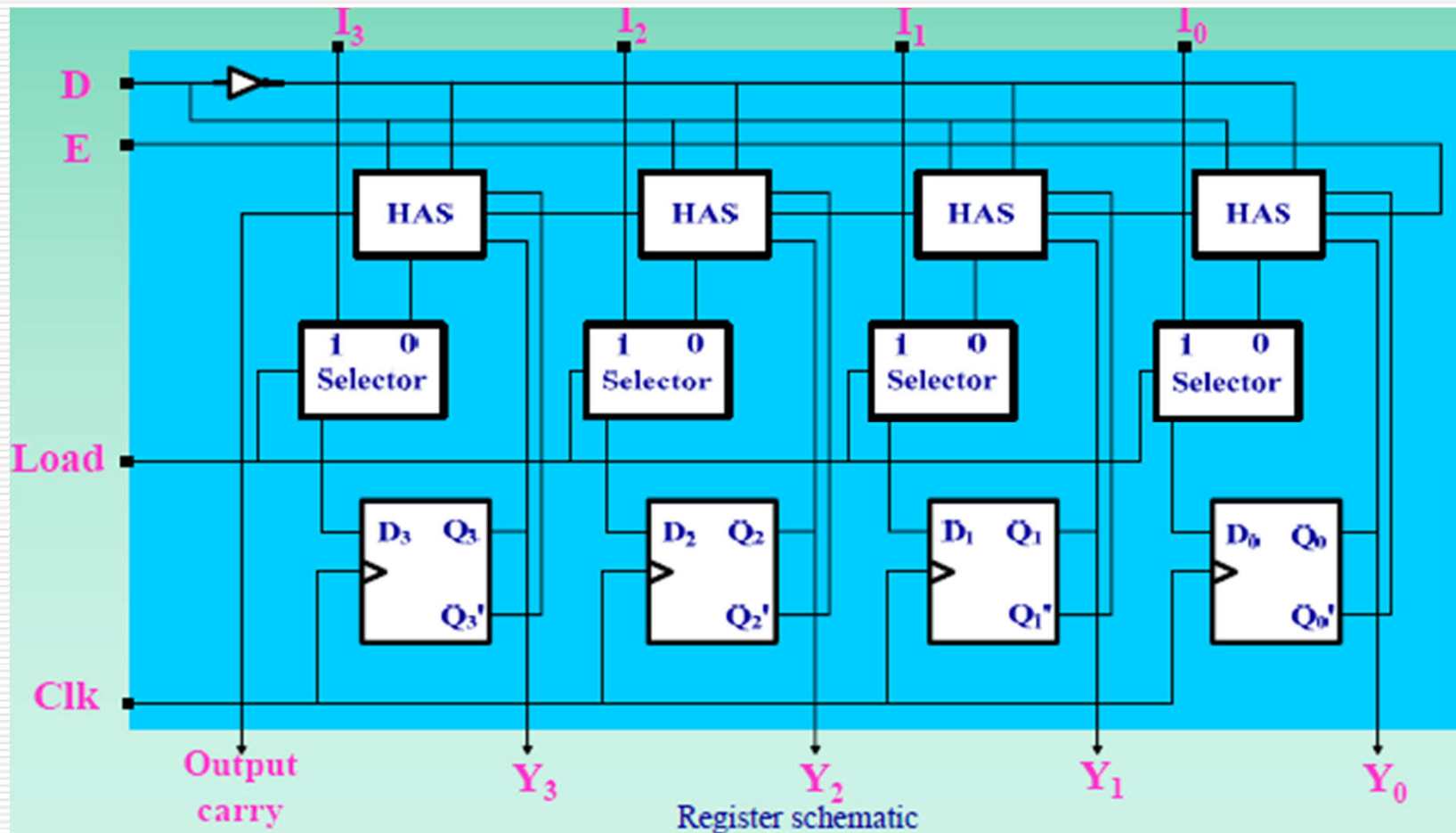
Numărător sincron în ambele sensuri cu o valoare de start încărcabilă



<i>Load</i>	<i>E</i>	<i>D</i>	<i>Operations</i>
0	0	X	No change
0	1	0	Count up
0	1	1	Count down
1	X	X	Load the input

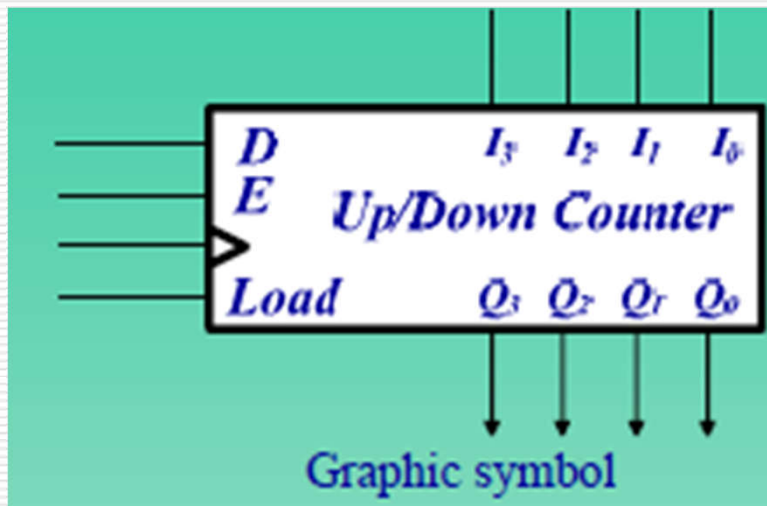
Operation table

Numărător sincron în ambele sensuri cu o valoare de start încărcabilă



Aplicație

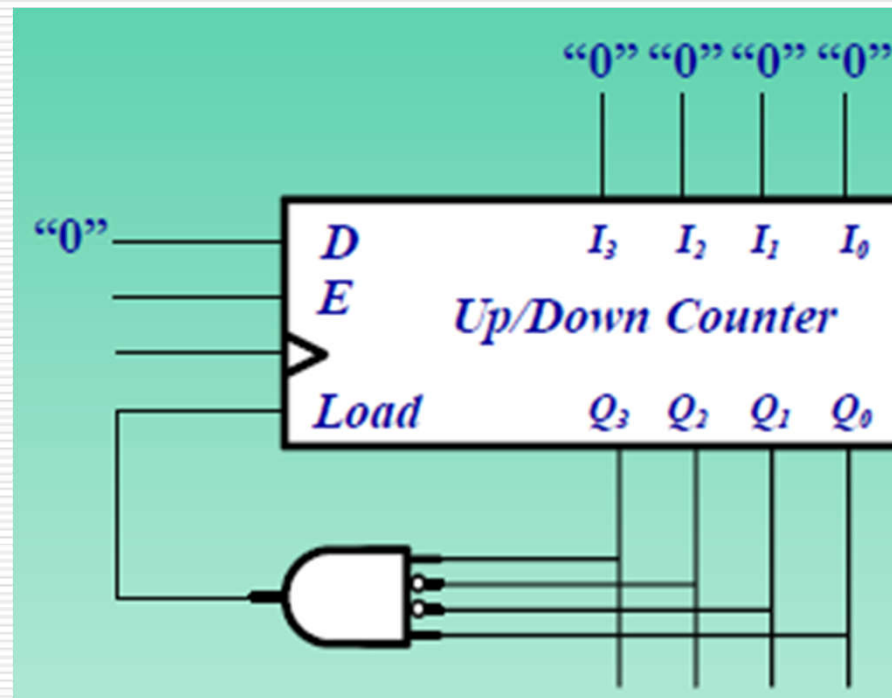
- Să se implementeze un numărător crescător modulo $M=10$ folosind următorul numărător:



Load	E	D	Operations
0	0	X	No change
0	1	0	Count up
0	1	1	Count down
1	X	X	Load the input

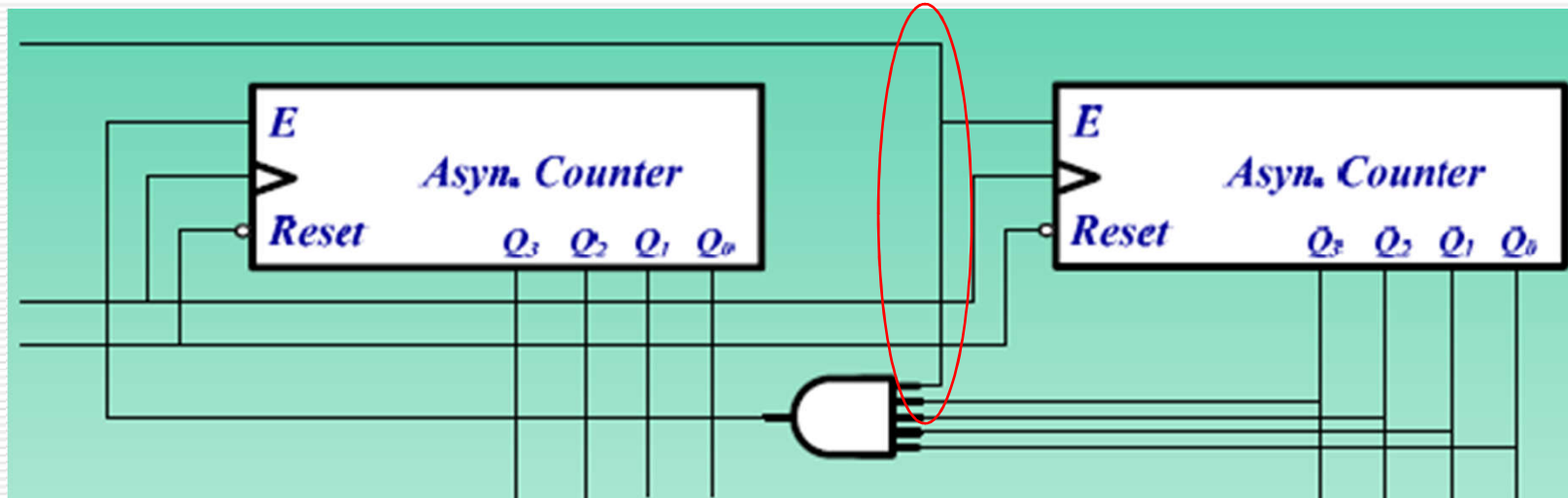
Operation table

Aplicație: soluție



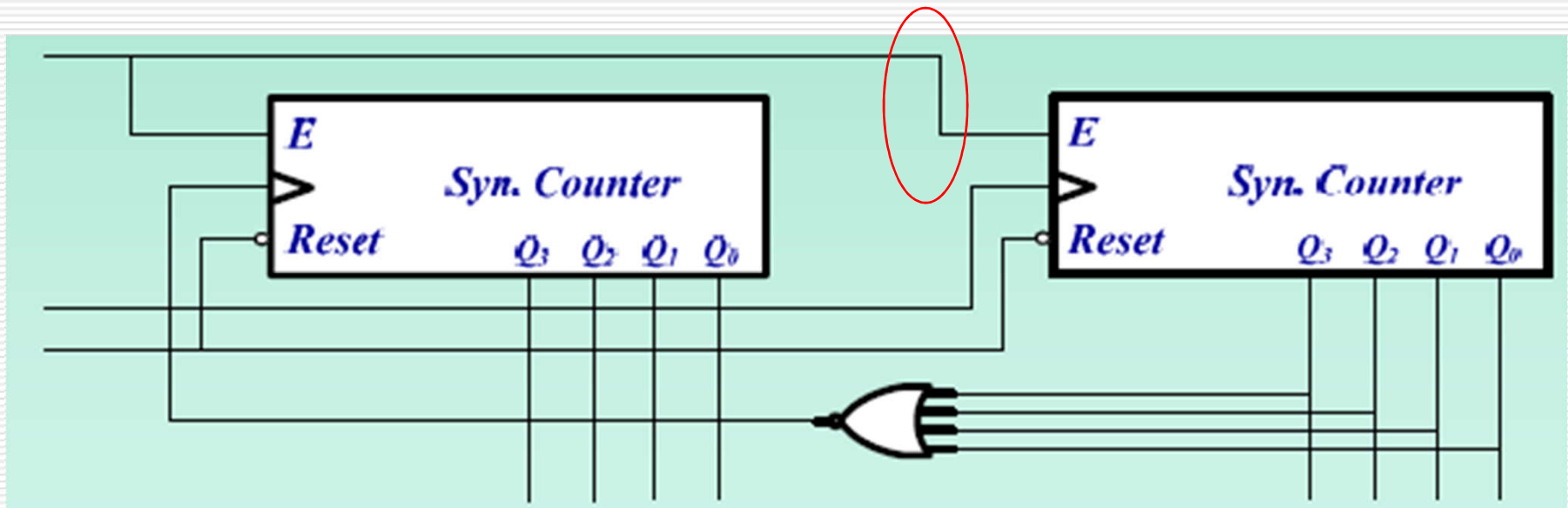
Numărătoare mixte (I)

- ❑ Blocuri asincrone conectate sincron



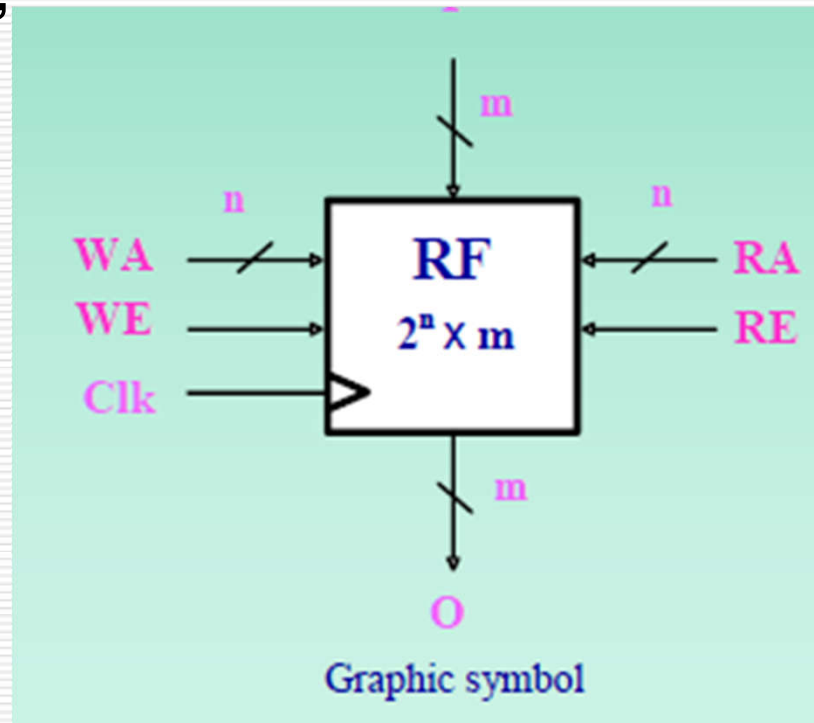
Numărătoare mixte (II)

- ❑ Blocuri sincrone conectate asincron



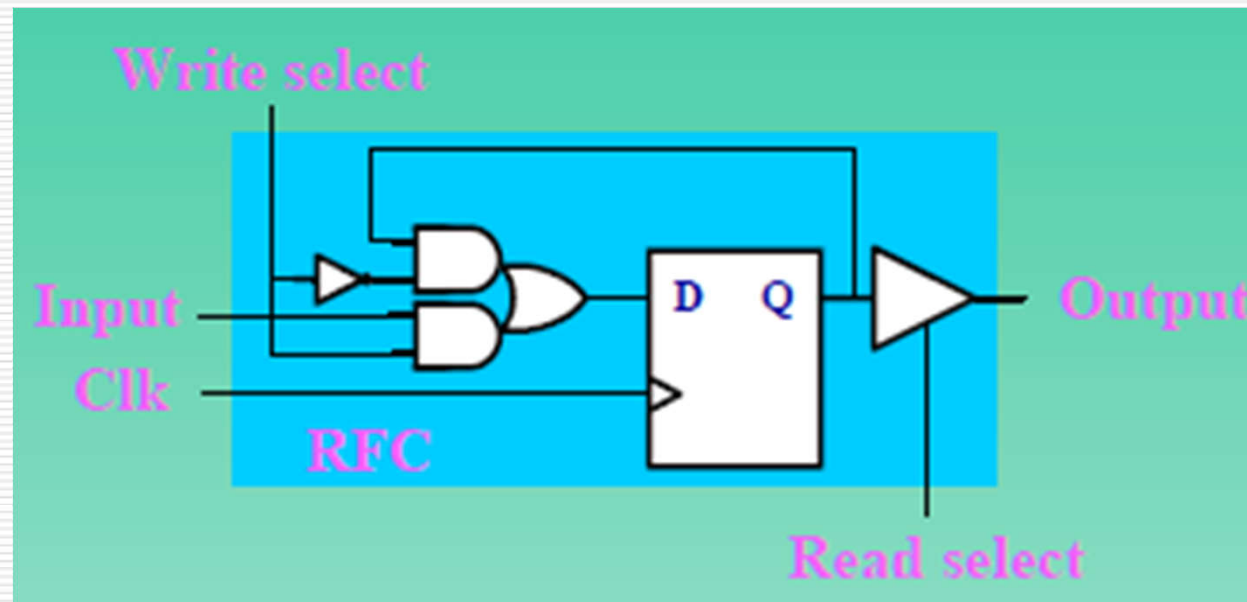
Pila de registre

- Colecție de registre pentru accesul rapid al informației:

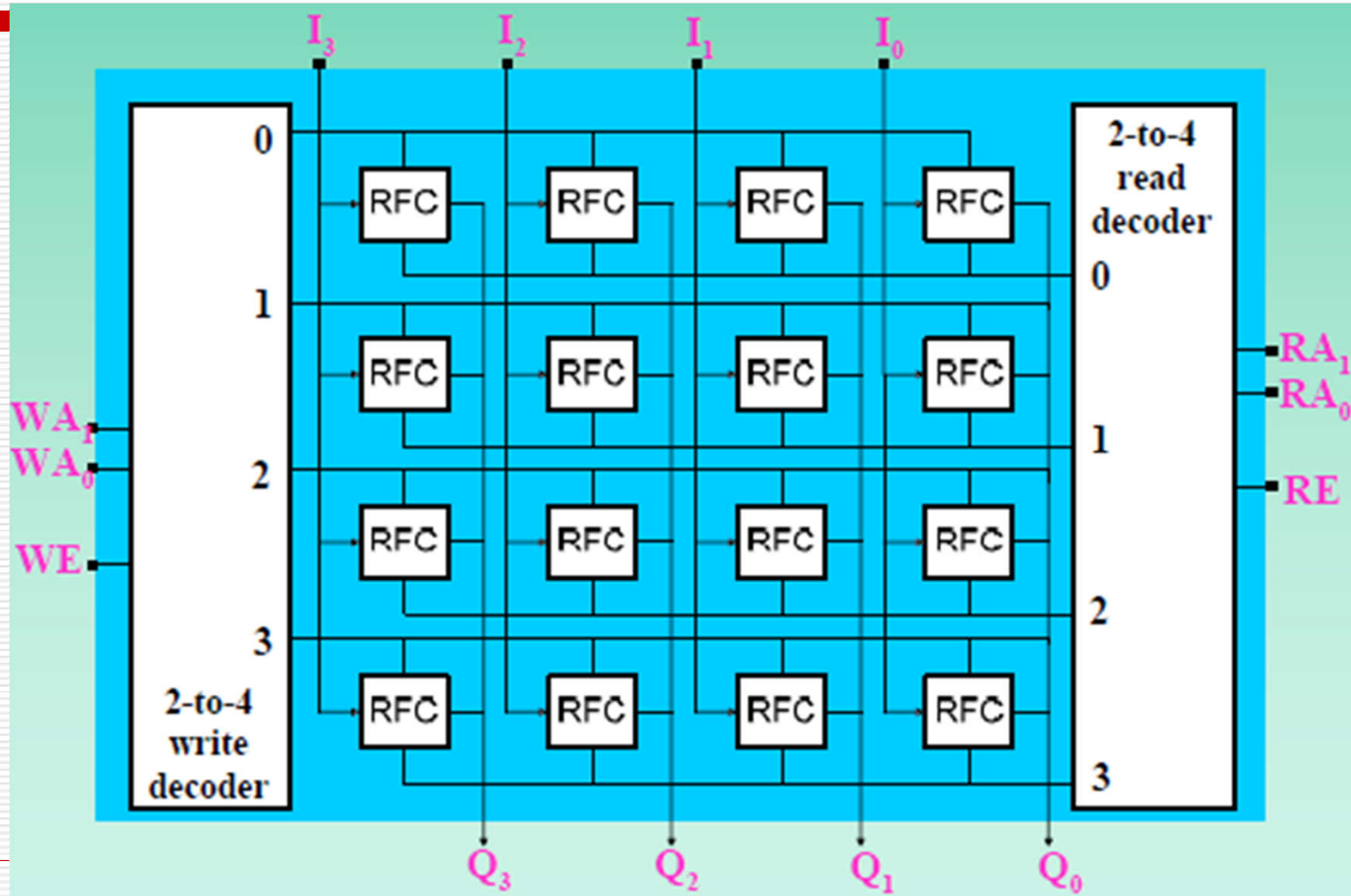


Pila de registre

- Celula pilei de registre 1 port citire/ 1 port scriere:

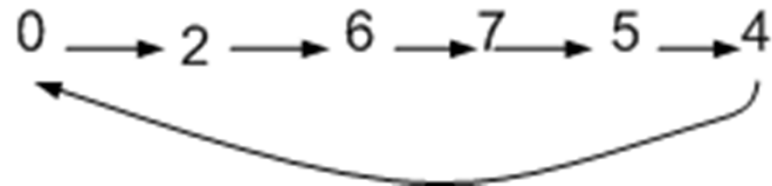
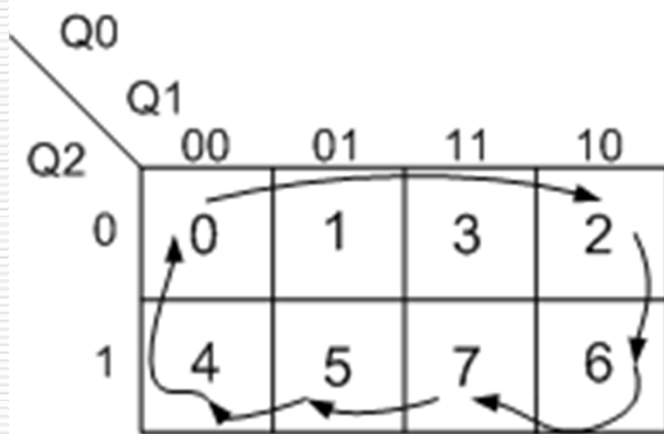


Pila de registre: aritectura



Aplicație 1*:

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



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.)

[illegible]

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 ₂	Q ₁	Q ₀	J ₂	K ₂	J ₁	K ₁	J ₀	K ₀
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 ₀	x ₁	x ₃	0 ₂
1	0 ₄	x ₅	x ₇	1 ₆

$$J_0 = Q_2 Q_1$$

K_0

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

$$K_0 = \overline{Q_1}$$

Aplicație 1:

J_1

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

$$J_1 = \overline{Q_2}$$

K_1

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

$$K_1 = Q_0$$

J_2

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

$$J_2 = Q_1$$

K_2

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

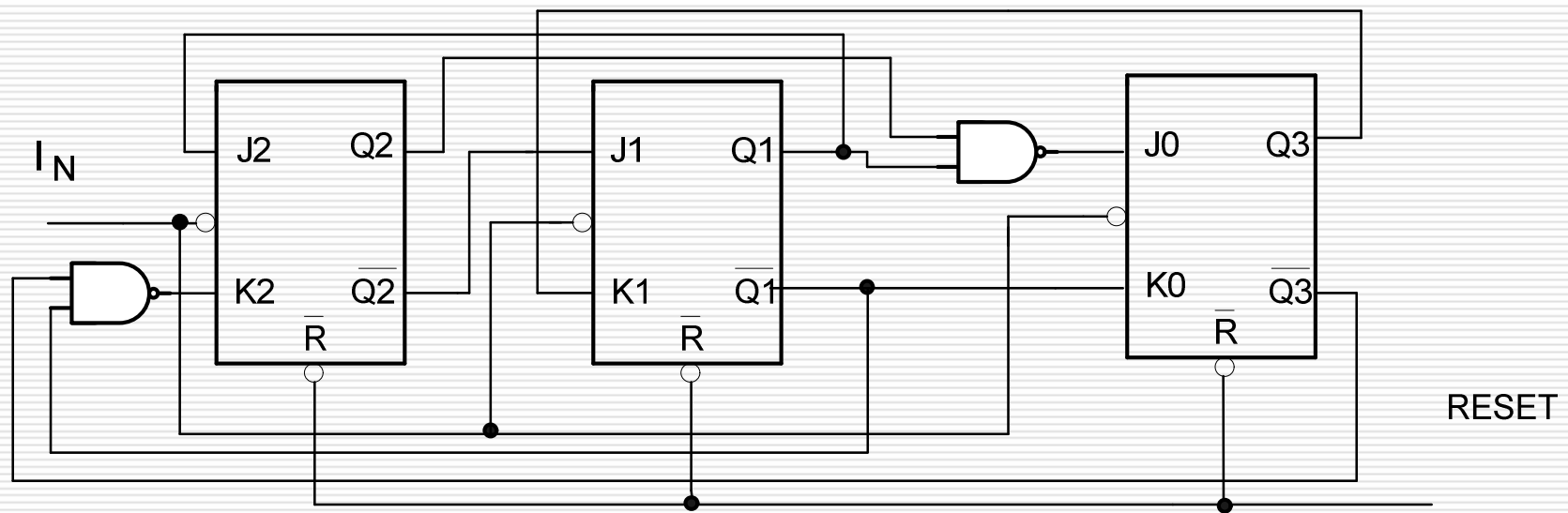
$$K_2 = \overline{Q_0} \overline{Q_1}$$

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!!**

