

Proiectare logică

Curs 9

Aplicații ale bistabilelor: registre,
convertoare, memorii RAM

Cristian Vancea

<https://users.utcluj.ro/~vcristian/PL.html>

Cuprins

Aplicații ale circuitelor basculante bistabile

- Registre
- Aplicații ale registrelor
 - Registre de deplasare cu reacție
 - Convertoare serial-paralel / paralel-serial
 - Generatoare de secvențe
- Memorii RAM

Aplicații ale circuitelor basculante bistabile

Registre

Definiție – registrele sunt circuitele logice secvențiale (CLS) care permit stocarea și/sau deplasarea datelor binare.

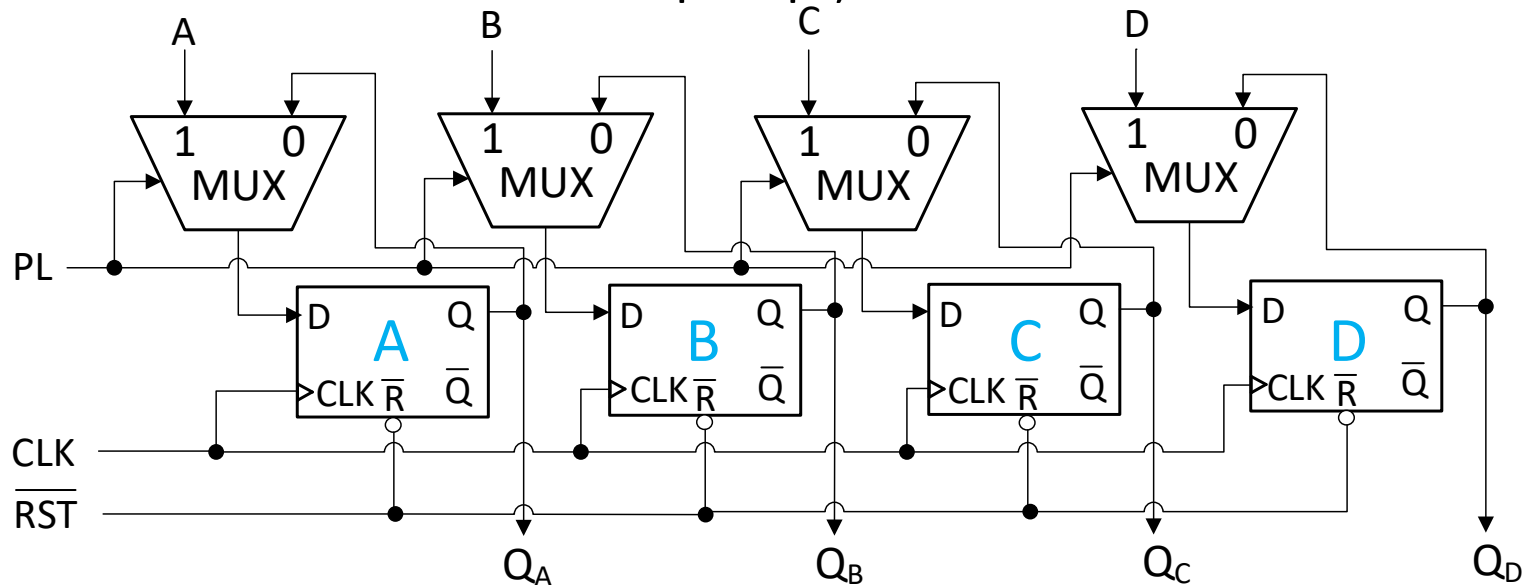
- Se implementează cu **bistabile** ca suport de memorare și **circuite logice combinaționale** care implementează diverse operații de scriere, citire, deplasare.
- Numărul de bistabile n = **capacitatea registrului**.
- Clasificare după funcționalitatea implementată:
 - de memorie;
 - de deplasare;
 - combinate;
 - universale.

Aplicații ale circuitelor basculante bistabile

Registre de memorie

- Memorează biții de date în bistabile, 1 bit / bistabil.
- Încarcă datele de intrare în paralel (toți biții sincron la fiecare impuls).
- O linie de comandă indică operația de efectuat: memorare sau încărcare.
- Pentru inițializare se pot utiliza intrările asincrone.

Ex: $n = 4$ – Sinteza cu bistabile D flip-flop și reset asincron:



Obs: PL (parallel load) \rightarrow PL = 0 \Rightarrow memorare ($Q_A Q_B Q_C Q_D^{t+1} = Q_A Q_B Q_C Q_D^t$);
 \rightarrow PL = 1 \Rightarrow încărcare ($Q_A Q_B Q_C Q_D^{t+1} = ABCD$).

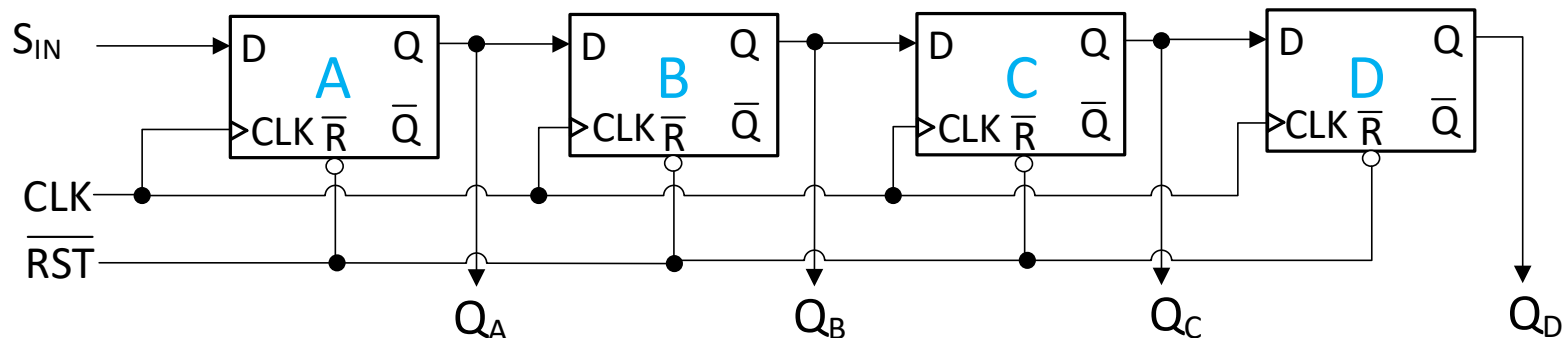
Aplicații ale circuitelor basculante bistabile

Registre de deplasare

- Realizează deplasarea biților unui număr binar la stânga, la dreapta sau în ambele sensuri (reversibil) în funcție de o comandă.
- Deplasarea biților se realizează cu câte o poziție la fiecare impuls de tact.
- Secvența introdusă pe intrarea serială la un capăt este reprodusă pe ieșirea bistabilului de la celălalt capăt cu o întârziere de n impulsuri de tact.

Ex: $n = 4$

Sinteza cu bistabile D flip-flop și reset asincron:



Obs₁: Funcționare în timp: $Q_A^{t+1} = S_{IN}$, $Q_B^{t+1} = Q_A^t$, $Q_C^{t+1} = Q_B^t$, $Q_D^{t+1} = Q_C^t$.

Obs₂: Valorile introduse serial (1 bit/impuls) pe intrarea S_{IN} (Serial Input) vor apărea pe ieșirea Q_D în aceeași ordine, după 4 impulsuri, apoi se pierd. 5

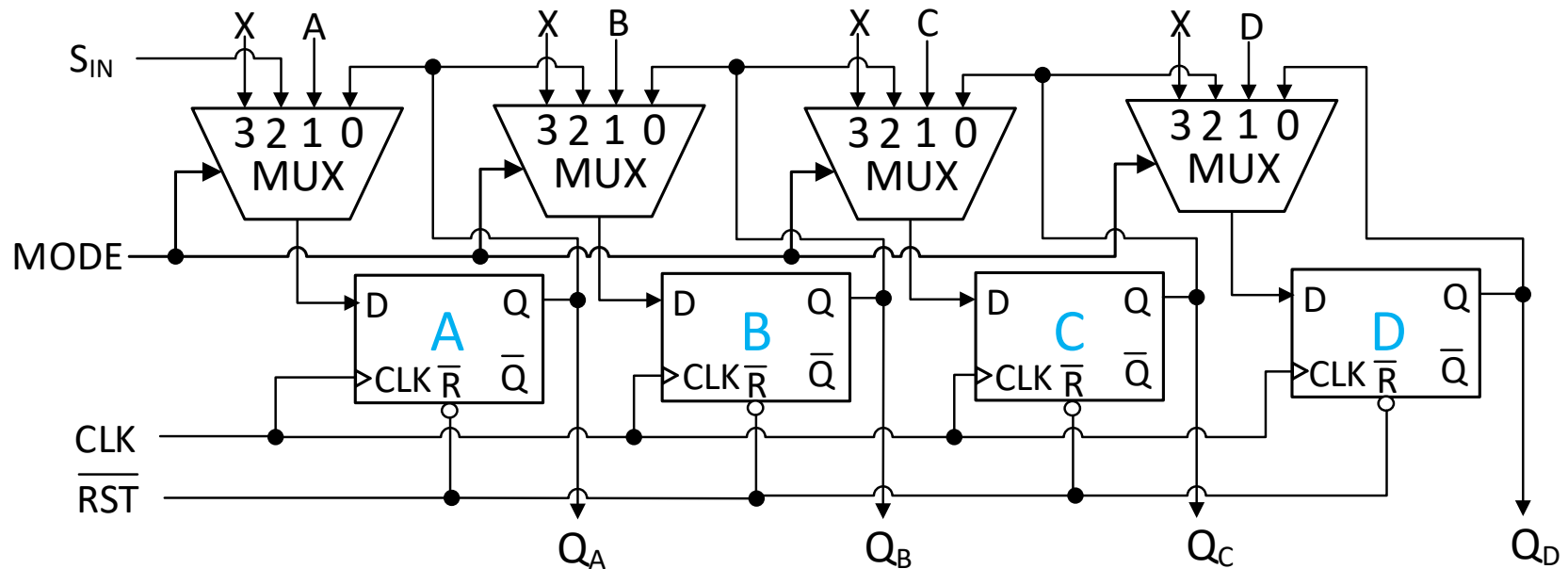
Aplicații ale circuitelor basculante bistabile

Registre combinate

- Registru de memorie + registru de deplasare = registru combinat.

Ex: $n = 4$

Sinteza cu bistabile D flip-flop și reset asincron:



Funcționare în timp:

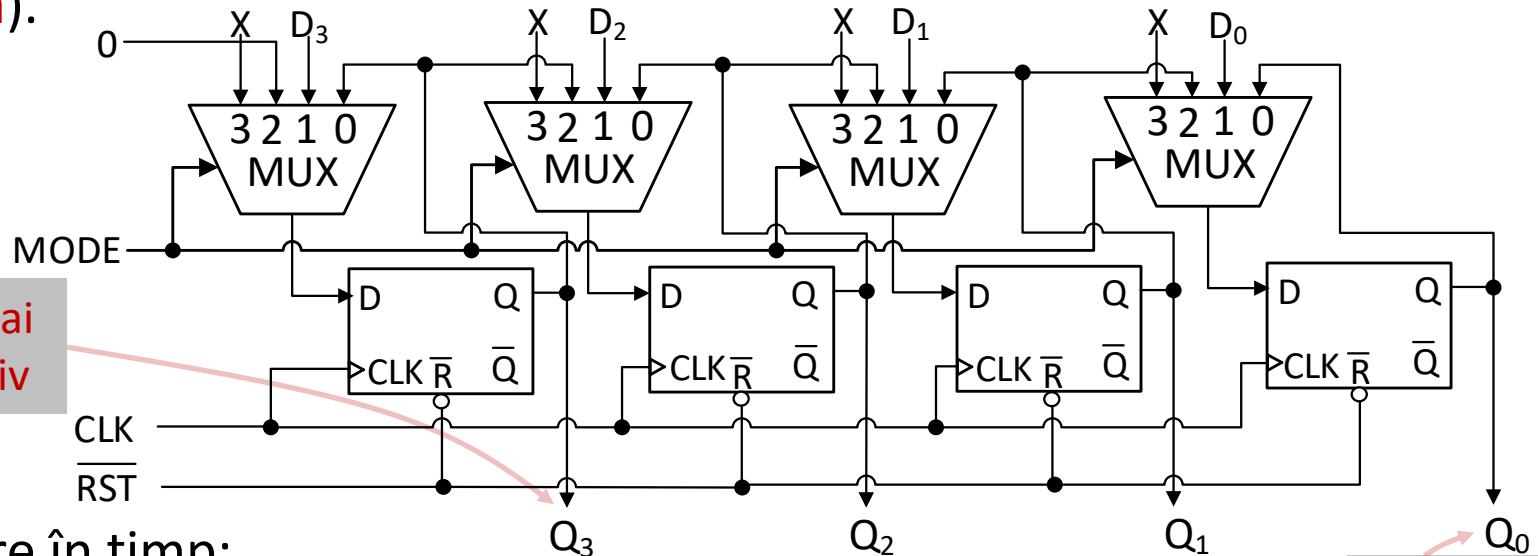
- $\rightarrow \text{MODE} = 00 \Rightarrow \text{memorare } (Q_A Q_B Q_C Q_D^{t+1} = Q_A Q_B Q_C Q_D^t);$
- $\rightarrow \text{MODE} = 01 \Rightarrow \text{încărcare paralelă } (Q_A Q_B Q_C Q_D^{t+1} = ABCD);$
- $\rightarrow \text{MODE} = 10 \Rightarrow \text{încărcare serială+deplasare } (Q_A Q_B Q_C Q_D^{t+1} = S_{IN} Q_A Q_B Q_C^t).$

Aplicații ale circuitelor basculante bistabile

Registre combinate (caz de utilizare)

- *Deplasare la dreapta* => valoarea din registru se împarte la 2 (**împărțire întreagă**).

Ex: $n = 4$



Funcționare în timp:

- $\text{MODE} = 00 \Rightarrow$ memorare ($Q_3 Q_2 Q_1 Q_0^{t+1} = Q_3 Q_2 Q_1 Q_0^t$);
- $\text{MODE} = 01 \Rightarrow$ încărcare paralelă ($Q_3 Q_2 Q_1 Q_0^{t+1} = D_3 D_2 D_1 D_0$);
- $\text{MODE} = 10 \Rightarrow$ **deplasare la dreapta** ($Q_3 Q_2 Q_1 Q_0^{t+1} = 0 Q_3 Q_2 Q_1^t$);

Poziția 3 2 1 0 3 2 1 0

Exemple deplasare la dreapta: **0110** → **0011** ($6 \rightarrow 3$)₁₀

Obs₁: Se pierde bitul Q_0 cel mai puțin semnificativ. **1100** → **0110** ($12 \rightarrow 6$)₁₀ Obs₂: Apare 0 la stânga.

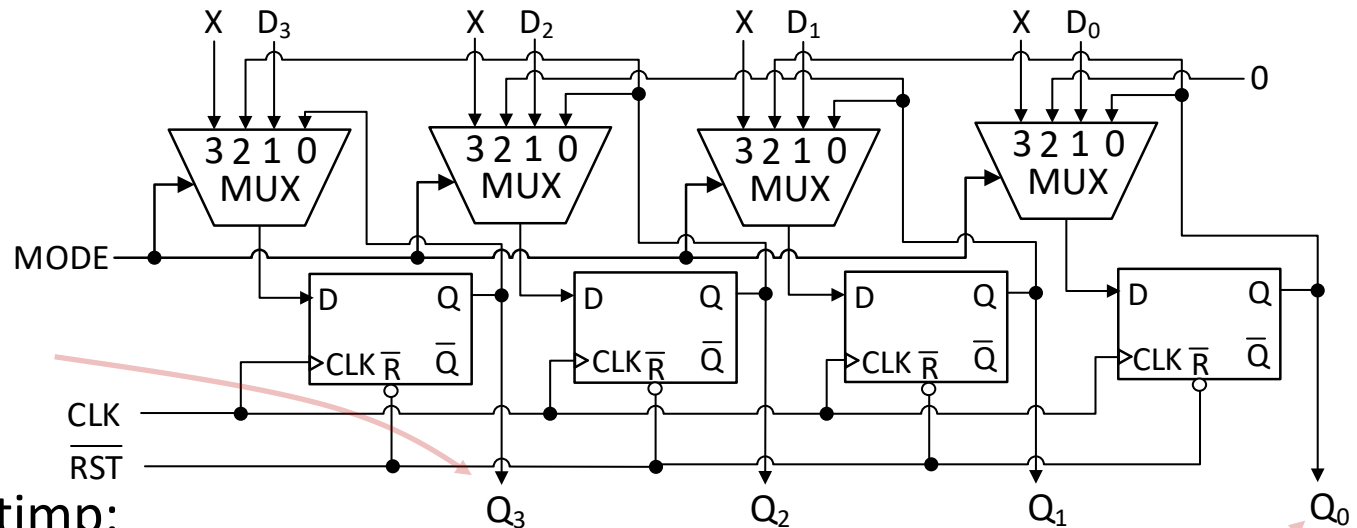
0011 → **0001** ($3 \rightarrow 1$)₁₀

Aplicații ale circuitelor basculante bistabile

Registre combinate (caz de utilizare)

- *Deplasare la stânga* => valoarea din registru **considerând bitul cel mai semnificativ 0** se înmulțește cu 2.

Ex: $n = 4$



bitul cel mai semnificativ

bitul cel mai puțin semnificativ

Funcționare în timp:

- $\text{MODE} = 00 \Rightarrow$ memorare ($Q_3 Q_2 Q_1 Q_0^{t+1} = Q_3 Q_2 Q_1 Q_0^t$);
- $\text{MODE} = 01 \Rightarrow$ încărcare paralelă ($Q_3 Q_2 Q_1 Q_0^{t+1} = D_3 D_2 D_1 D_0$);
- $\text{MODE} = 10 \Rightarrow$ **deplasare la stânga** ($Q_3 Q_2 Q_1 Q_0^{t+1} = Q_2 Q_1 Q_0 0^t$);

Poziția 3 2 1 0 3 2 1 0

Exemple deplasare la stânga: 0110 → 1100 ($6 \rightarrow 12$)₁₀

Obs₁: Se pierde bitul Q_3
cel mai semnificativ.

1100 → 1000 ($4 \rightarrow 8$)₁₀

0011 → 0110 ($3 \rightarrow 6$)₁₀

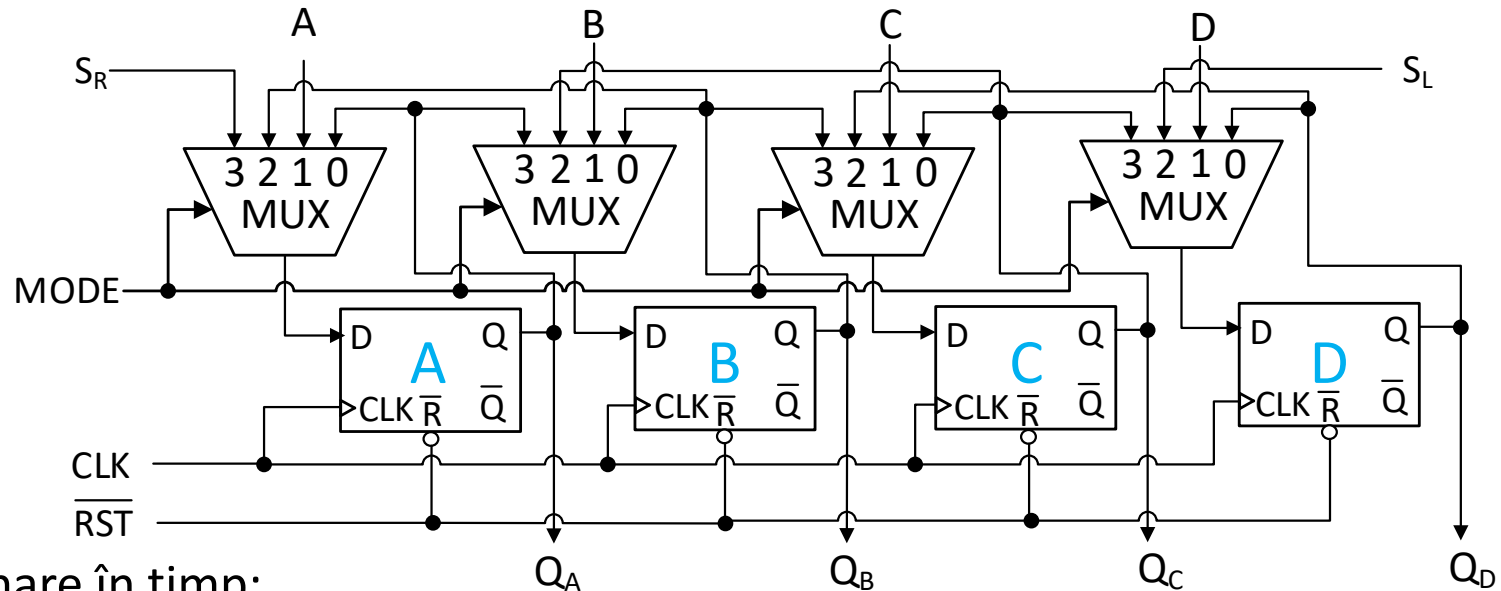
Obs₂: Apare 0 la dreapta.

Aplicații ale circuitelor basculante bistabile

Registre universale

- Registru de memorie + deplasare în ambele direcții = registru universal.

Ex: $n = 4$ – Sinteza cu bistabile D flip-flop și reset asincron:



Funcționare în timp:

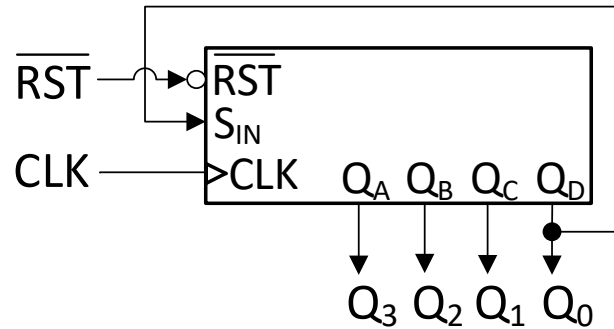
- $\rightarrow \text{MODE} = 00 \Rightarrow \text{memorare } (Q_A Q_B Q_C Q_D^{t+1} = Q_A Q_B Q_C Q_D^t);$
- $\rightarrow \text{MODE} = 01 \Rightarrow \text{încărcare paralelă } (Q_A Q_B Q_C Q_D^{t+1} = ABCD);$
- $\rightarrow \text{MODE} = 10 \Rightarrow \text{încărcare serială prin intrarea } S_L \text{ (serial left) +}$
deplasare stânga D \rightarrow A ($Q_A Q_B Q_C Q_D^{t+1} = Q_B Q_C Q_D S_L^t$);
- $\rightarrow \text{MODE} = 11 \Rightarrow \text{încărcare serială prin intrarea } S_R \text{ (serial right) +}$
deplasare dreapta A \rightarrow D ($Q_A Q_B Q_C Q_D^{t+1} = S_R Q_A Q_B Q_C^t$).

Aplicații ale circuitelor basculante bistabile

Aplicații ale registrelor – registre de deplasare cu reacție

- *Registru de deplasare în inel* (straight ring counter)

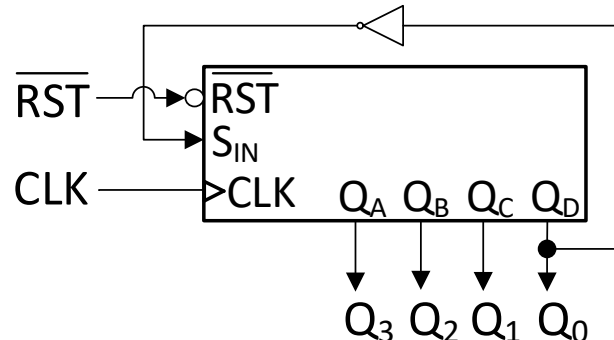
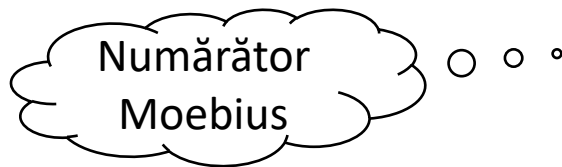
Ex: $n = 4$ – Sinteza cu *registru de deplasare*. Necesită reset asincron la valoarea $(Q_A Q_B Q_C Q_D)_2 = 1000$:



Graful de tranziții: $\overline{RST} \rightarrow 1000 \rightarrow 0100 \rightarrow 0010 \rightarrow 0001$

- *Registru Johnson de deplasare în inel cu inversare* (twisted ring counter)

Ex: $n = 4$



Graful de tranziții:

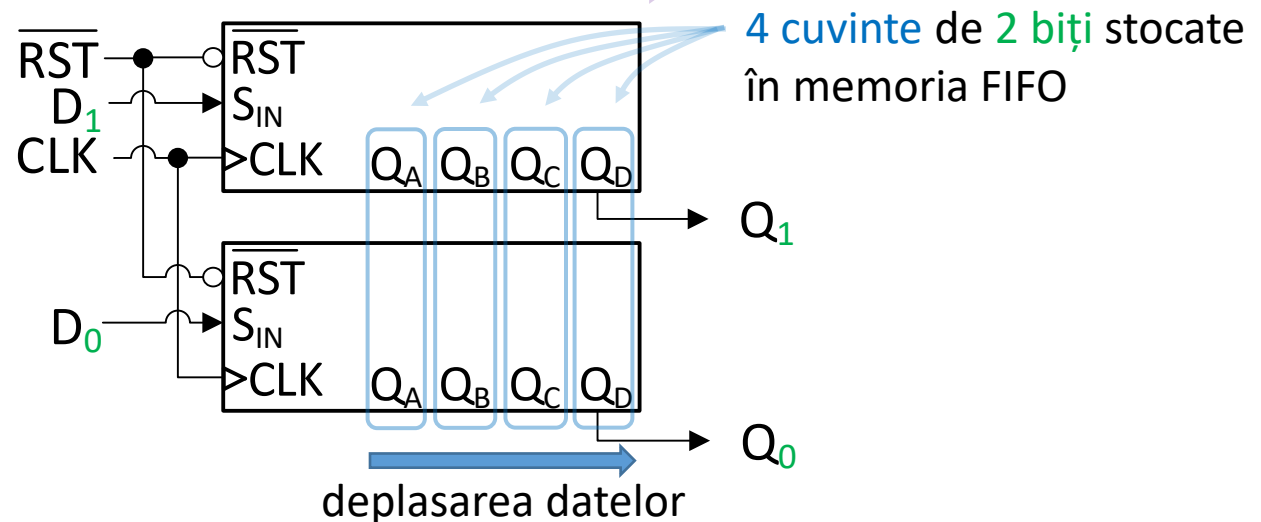
$\overline{RST} \rightarrow 0000 \rightarrow 1000 \rightarrow 1100 \rightarrow 1110 \rightarrow 1111 \rightarrow 0111 \rightarrow 0011 \rightarrow 0001$

Aplicații ale circuitelor basculante bistabile

Aplicații ale registrelor – **memorie FIFO (first-in, first-out)**

- Sunt memorii în care datele se obțin la ieșire în ordinea în care s-au scris.
- Se implementează cu **registre de deplasare**.
- Numărul maxim de cuvinte stocate = **capacitatea memoriei**.
- Capacitatea memoriei n este dată de capacitatea registrelor de deplasare.
- Dimensiunea m a unui cuvânt stocat este dată de numărul de registre utilizate.

Ex: FIFO pe 2 biți de capacitate 4 ($m = 2$, $n = 4$)



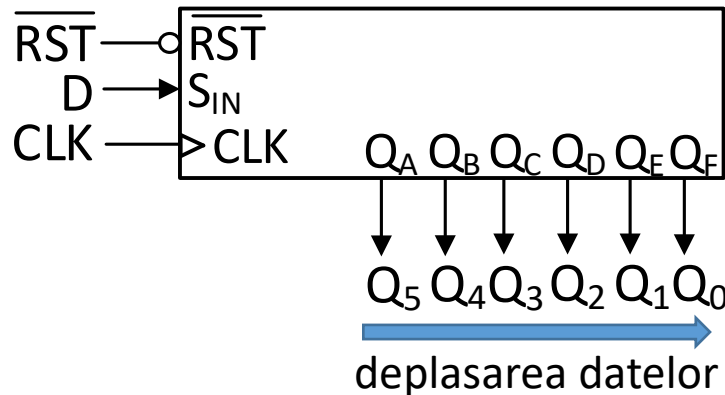
Obs: Datele de dimensiune $m=2$ biți se înscriu pe intrările seriale S_{IN} și se citesc după $n=4$ impulsuri de tact pe ieșirile Q_D ale registrelor.

Aplicații ale circuitelor basculante bistabile

Aplicații ale registrelor – **conversie serial-paralel (SIPO – serial-input, parallel-output)**

- Se implementează cu **registru de deplasare** de capacitate n .
- La fiecare n operații de scriere serială pe intrarea S_{IN} se realizează o citire paralelă a biților înscriși.

Ex: $n = 6$



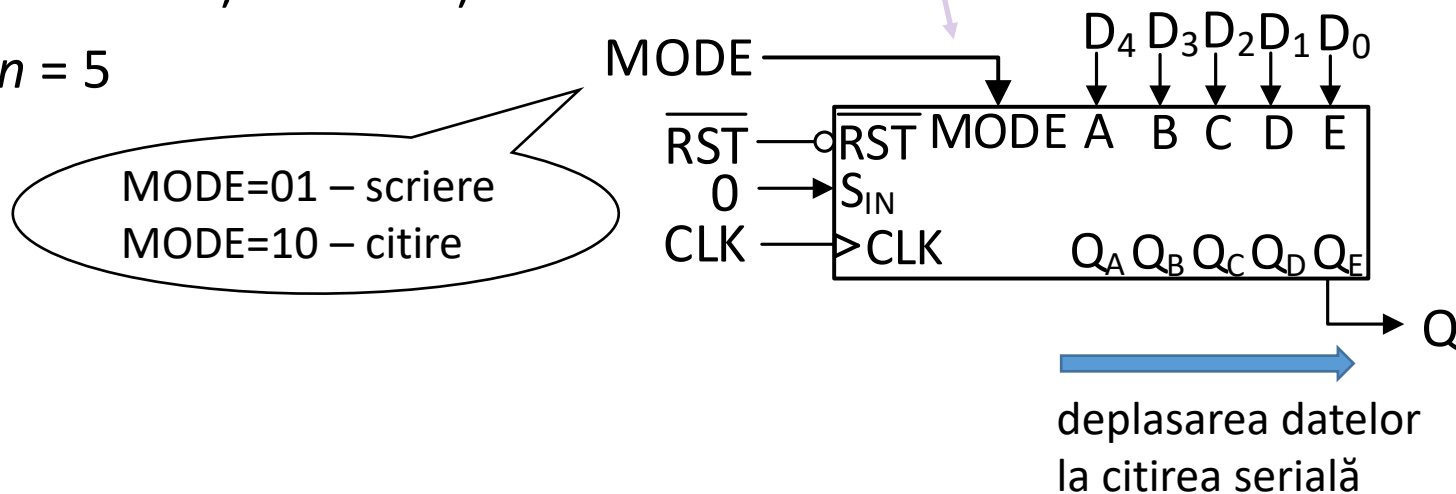
Obs: Datele se înscriu serial în $n=6$ impulsuri de tact începând cu bitul cel mai puțin semnificativ și terminând cu cel mai semnificativ, iar apoi sunt disponibile pe ieșiri pentru citire paralelă numai pe durata următorului impuls.

Aplicații ale circuitelor basculante bistabile

Aplicații ale registrelor – **conversie paralel-serial (PISO – parallel-input, serial-output)**

- Se implementează cu **registru combinat** de capacitate n .
- La fiecare operație de scriere paralelă se realizează n operații de citire serială a biților înscriși.

Ex: $n = 5$



Obs₁: Când $MODE=01$ numărul $D_4D_3D_2D_1D_0$ se înregistrează paralel într-un impuls de tact, iar apoi, în configurația de deplasare $MODE=10$, biții se pot citi serial de pe ieșirea Q_E pe durata a $n=5$ impulsuri de tact.

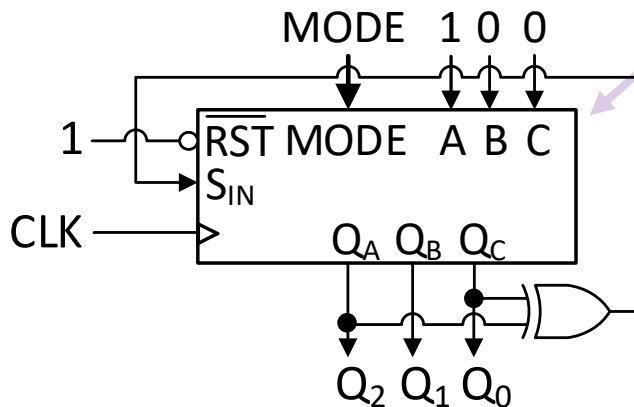
Obs₂: Citirea se realizează începând cu bitul cel mai puțin semnificativ și terminând cu cel mai semnificativ.

Aplicații ale circuitelor basculante bistabile

Aplicații ale registrelor – **generatoare de secvențe**

- Generează un set de numere binare pe n biți care se repetă în aceeași ordine. În cadrul setului (secvenței) numerele apar o singură dată.
- Secvențele de lungime finită = **secvențe deterministe**.
- Secvențele deterministe, aparent aleatoare, de lungime maximă = **secvențe pseudoaleatoare**.
- Sinteza se face cu **registru de deplasare combinat**, care conțin circuite logice combinaționale adecvate în calea de reacție.

Ex: Generator de secvențe pseudoaleatoare pe $n = 3$ biți



- MODE = 01 → Se încarcă valoarea 100
- MODE = 10 → Funcționare în bucla repetitivă:
100 → 110 → 111 → 011 → 101 → 010 → 001
(secvență pseudoaleatoare)

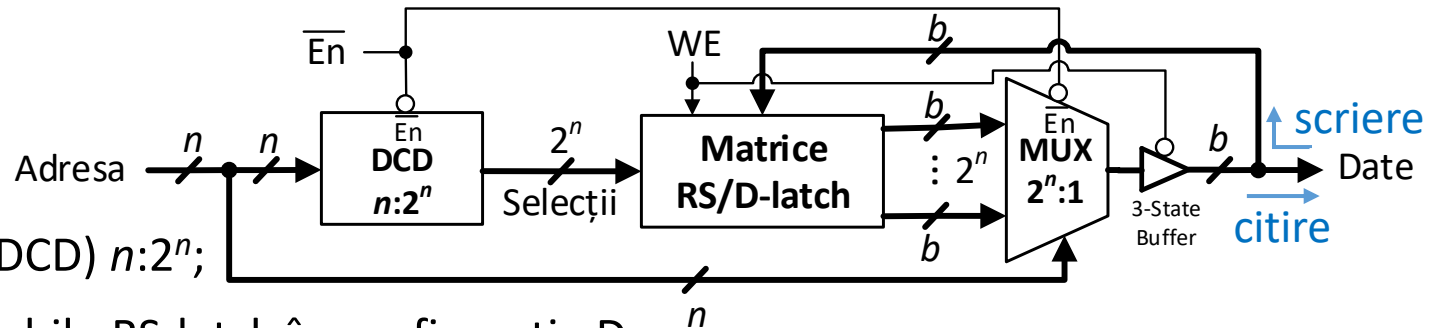
Obs: **Se evită intrarea în starea 000.**

Aplicații ale circuitelor basculante bistabile

Memorii RAM (Random Access Memory)

- Memoria RAM realizează operații de scriere și citire a datelor la adresa indicată pe liniile de adresă => memorie ROM + scriere.

- Structura:



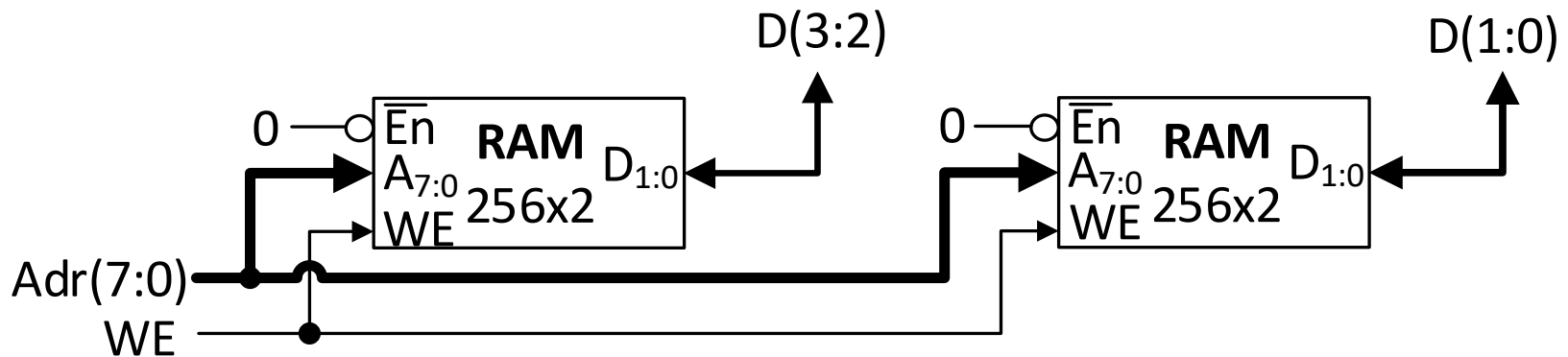
1. Decodificator (DCD) $n:2^n$;
 2. Matrice de bistabile RS-latch în configurație D;
 3. Multiplexor (MUX) $2^n:1$ cu calea de date pe b biți.
- Are o **intrare de activare** (pe 0 logic) numită \overline{En} (Enable).
 - Are o intrare de **comandă pentru scriere** numită WE (Write Enable):
 - WE = 0 – se citesc datele la adresa specificată pe liniile de adresă;
 - WE = 1 – se scriu datele de la adresa specificată pe liniile de adresă.
 - Are o **intrare de adresă** pe n biți – magistrală (bus) de adresă.
 - Are o **magistrală de date** pe b biți bidirecțională utilizată atât la scriere cât și la citire. Există și variante cu 2 magistrale unidirecționale pentru scriere și citire.

Aplicații ale circuitelor basculante bistabile

Memorii RAM (Random Access Memory)

- Organizarea memoriei:

- n – numărul de biți de adresă \Rightarrow numărul de cuvinte de memorie $= 2^n$
- b – numărul de biți ai unui cuvânt de memorie (de obicei putere a lui 2)
- **capacitatea memoriei $= 2^n \times b$**
- se poate mări capacitatea folosind mai multe memorii:
Ex₁: Mărirea cuvântului de memorie de la 2 la 4 biți pentru o memorie cu 256 cuvinte.



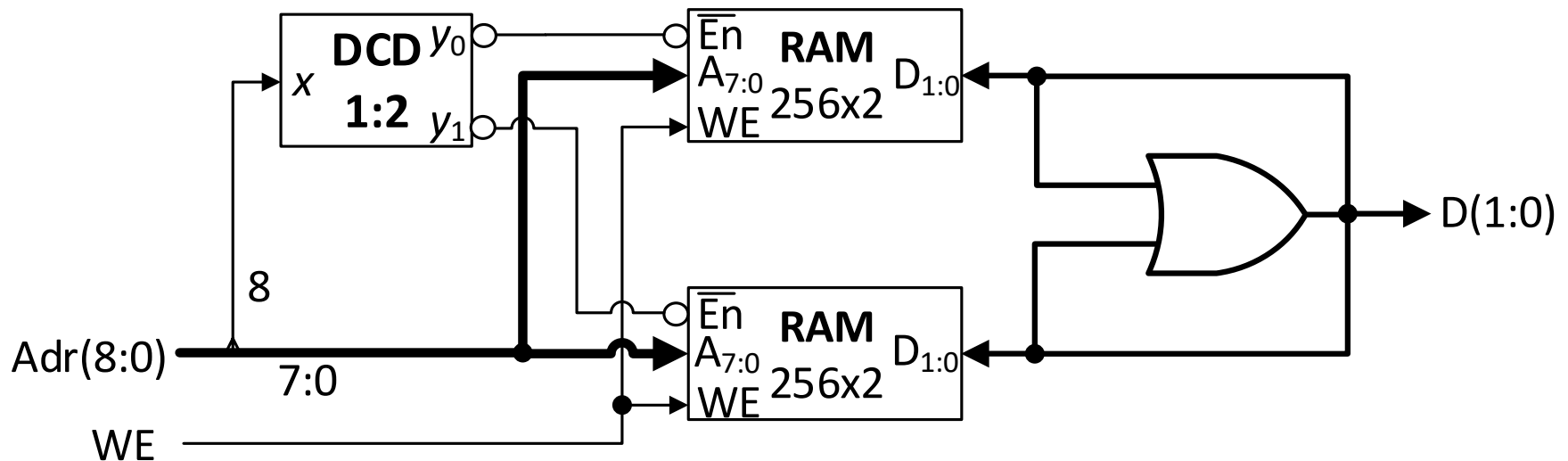
Aplicații ale circuitelor basculante bistabile

Memorii RAM (Random Access Memory)

- Organizarea memoriei:

- se poate mări capacitatea folosind mai multe memorii:

Ex₂: Dublarea numărului de cuvinte pe 2 biți de la 256 la 512.



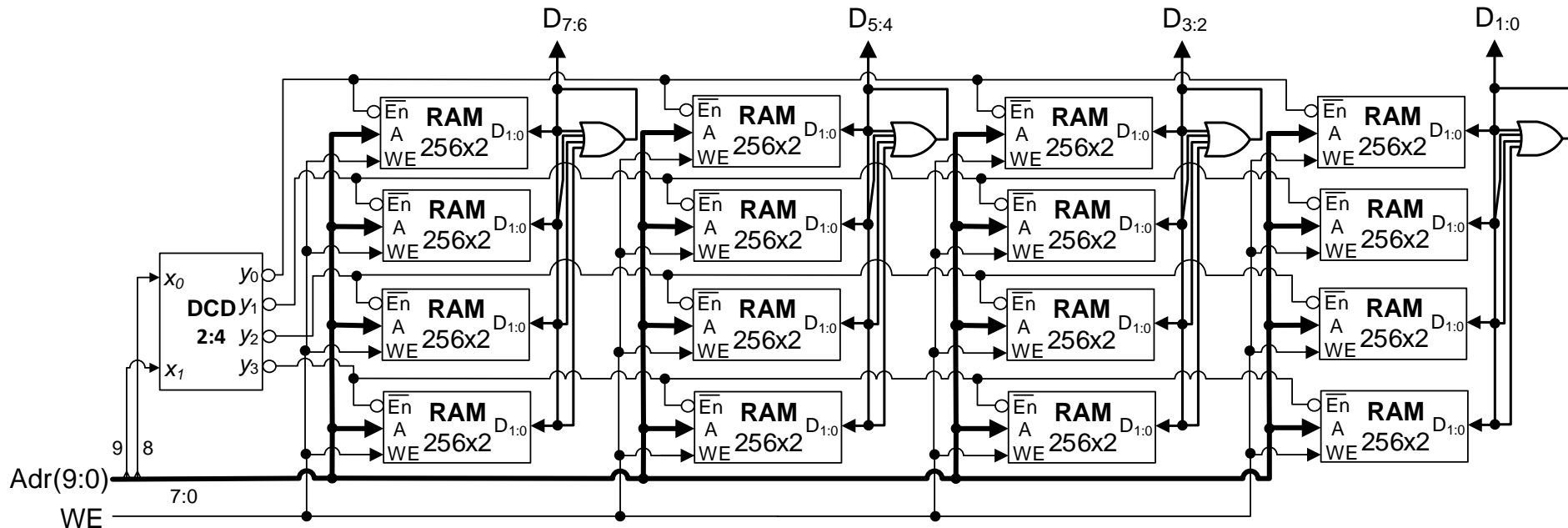
Aplicații ale circuitelor basculante bistabile

Memorii RAM (Random Access Memory)

- Organizarea memoriei:

- se poate mări capacitatea folosind mai multe memorii:

Ex₃: 4 x numărul de cuvinte + 4 x numărul de biți / cuvânt => RAM1024x8 (1KB)



Aplicații ale circuitelor basculante bistabile

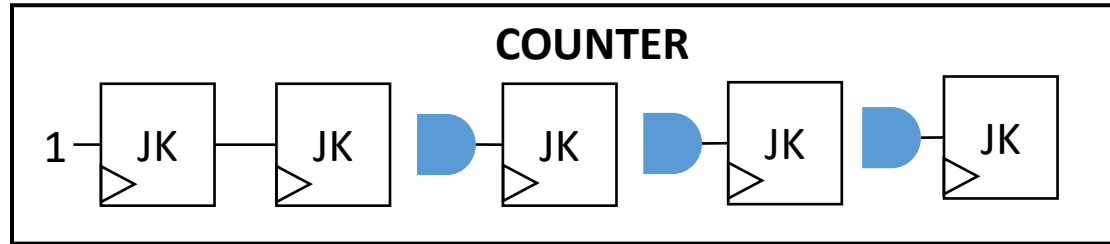
Memorii RAM (Random Access Memory)

- Sunt **memorii volatile** – pierd informația stocată după oprirea alimentării.
- Clasificare:
 - DRAM (Dynamic RAM)
 - Conține celule de memorie în tehnologie bazată pe condensatoare;
 - Necesită operații de reactualizare (refresh) a conținutului memoriei la anumite intervale de timp pentru a preîntâmpina descărcarea condensatoarelor și deci pierderea datelor => sunt mai lente, au capacitate mai mare și se folosesc pe post de memorie de bază în calculatoare;
 - Varianta sincronă cu semnal de tact CLK => SDRAM (Synchronous DRAM).
 - SRAM (Static RAM)
 - Conține latch-uri realizate cu tranzistoare care nu pierd informația cât timp sunt alimentate;
 - Sunt mai rapide, au capacitate mai mică și se folosesc pe post de cache în procesoare;
 - Nu necesită refresh;
 - Există și în varianta sincronă cu semnal de tact CLK => SSRAM (Synchronous SRAM).

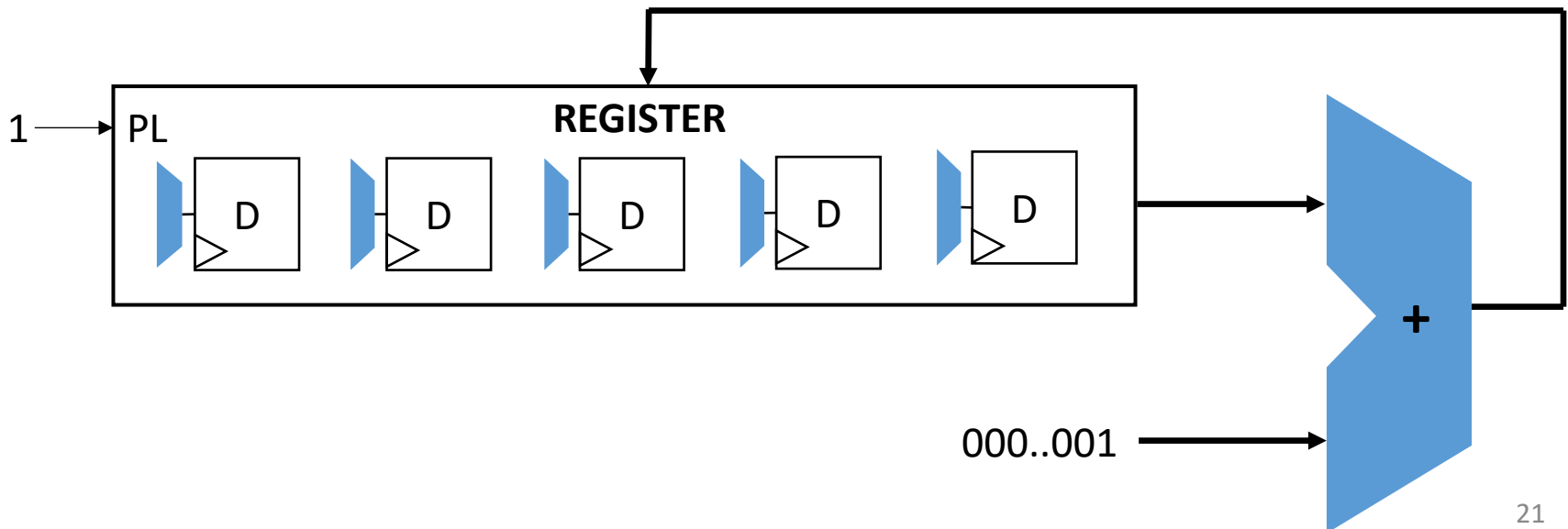
Studiu comparativ

Implementarea unui numărător

- Cu bistabile JK (**eficient**)



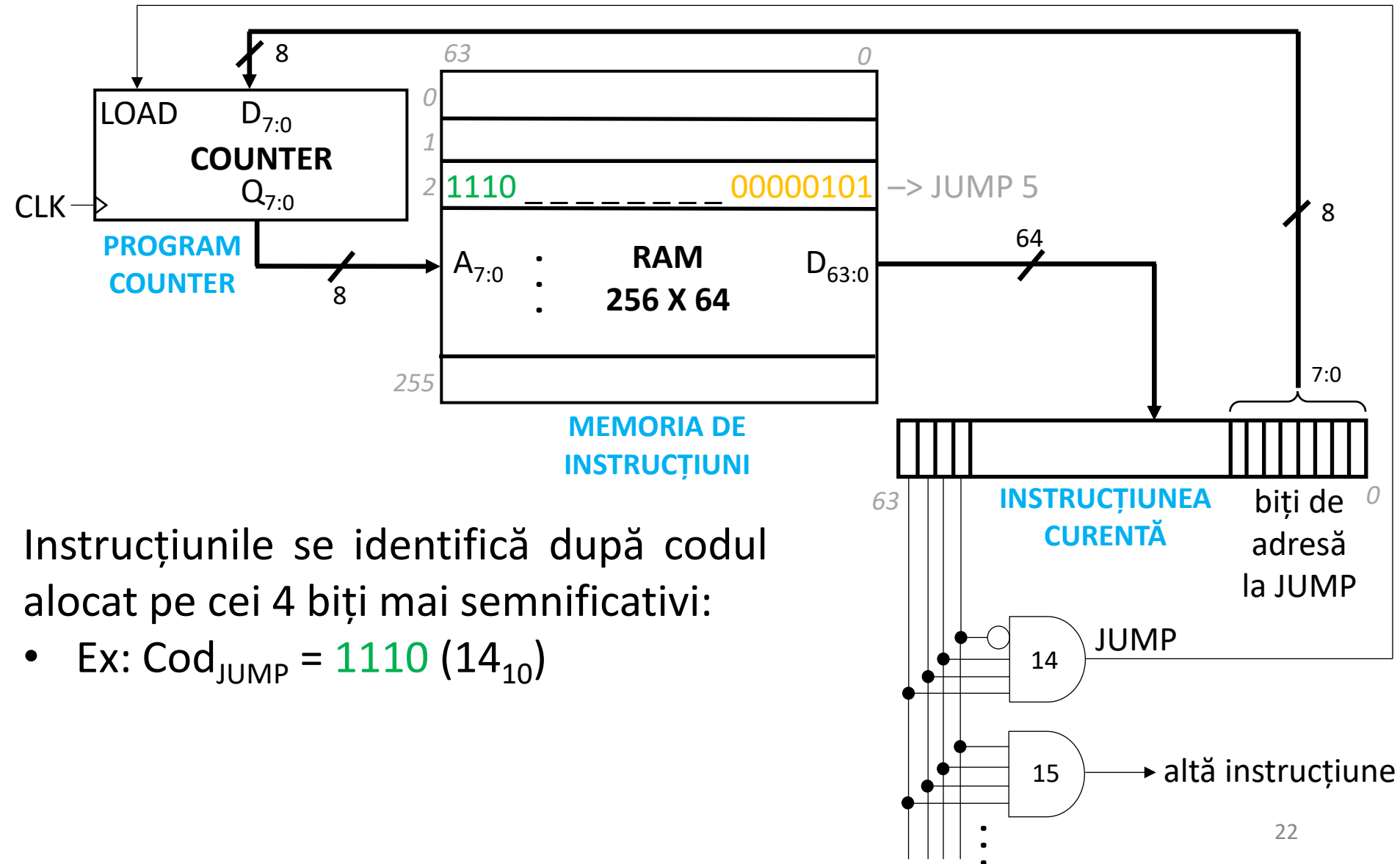
- Cu registru și sumator (**resurse multe => cost și consum ridicate; calea de propagare lungă => frecvență CLK mică**)



Studiu de caz

Optional

Implementarea instrucțiunii de salt JUMP (GO-TO, bucle) în procesoare



Instrucțiunile se identifică după codul alocat pe cei 4 biți mai semnificativi:

- Ex: Cod_{JUMP} = 1110 (14₁₀)