

CALCULATOARE NUMERICE

Calculatoare 2 2024-2025 Sem. 2

CURS 3 Sapt 3 13 martie 2025 12:00-14:00

serban@upit.ro

Regulament disciplina

Nota finala este formata din activitatile:

- Laborator 30%
- Lucrare control (midterm) 20%
- Examen 50%
- Bonus – prezenta activa curs 10%

Conditii pentru promovare:

- Nota de la laborator trebuie sa fie minim 5 (prezenta obligatorie la toate sedintele de laborator);
- Nota la lucrarea de control trebuie sa fie minim 5;
- Nota de la examen trebuie sa fie minim 5.

In cazul reluarii disciplinei intr-un alt an universitar, activitatile nepromovate trebuie parcuse din nou.

Continut disciplina

Structuri digitale folosite în calculatoare numerice

Memorii semiconductoare in calculatoare numerice

Memoria cache

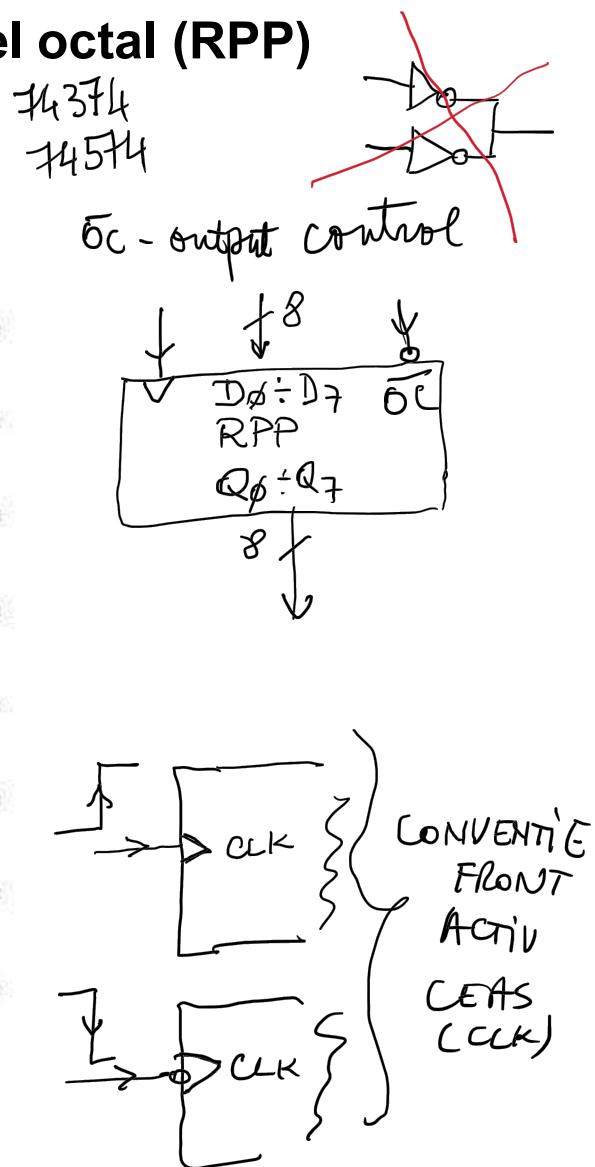
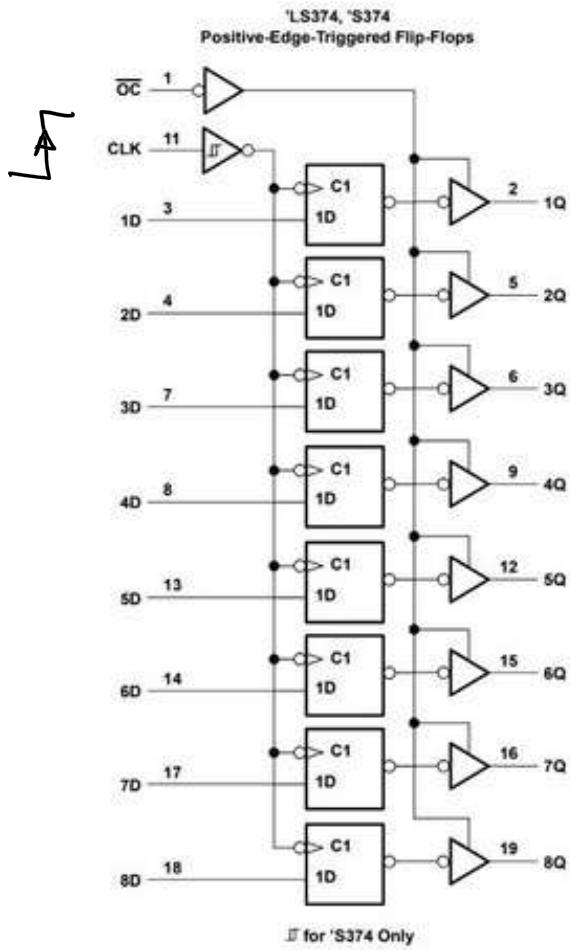
Memoria virtuală

Adresarea memoriei in sistemele de calcul

Bibliografie

1. David PATTERSON, John HENNESSY *Computer Organization and Design The Hardware/Software Interface*, 5th ed., Morgan Kaufmann Elsevier 2012 (a se vedea și traducerea în lb. română David A. Patterson, John L. Hennessy *Organizarea și proiectarea calculatoarelor, Interfața hardware/software*; Editura All, București, 2002);
2. John L. HENNESSY, David A. PATTERSON *Computer Architecture, A Quantitative Approach*, 5th ed., Morgan Kaufmann Publishers, Inc, San Francisco, 2012;
3. Bruce JACOB, Spencer NG, David WANG *Memory Systems Cache, DRAM, Disk* Morgan Kaufmann Elsevier 2008;
4. Miles MURDOCCA, Vincent HEURING *Principles of Computer Architecture*, Prentice Hill, 1999;
5. Jim HANDY *The Cache Memory*, 2nd ed., Academic Press Elsevier, 1998;
6. Andrew S. TANENBAUM, Todd AUSTIN *Structured Computer Organization*, 6th ed., Prentice-Hall, Inc., 2013 (a se vedea și traducerea în lb. română Andrew S. Tanenbaum *Organizarea Structurată a Calculatoarelor*, Agora, Tg. Mureș, 2004);
7. William STALLINGS *Computer Organization and Architecture: Designing for Performance*, 9th edition, Prentice-Hall Inc., 2013;
8. Vincent HEURING, Harry JORDAN *Computer Systems Design and Architecture*, 2nd ed., Person Prentice Hill, 2007;
9. Sajjan G. SHIVA *Computer Organization, Design, and Architecture*, 4th ed., CRC Press, Taylor & Francis Group, Boca Raton, USA, 2008;
10. Carl HAMACHER, Zvonko VRANESIC, Safwat ZAKY, Naraig MANJIKIAN *Computer organization and embedded systems, 6th edition* McGraw Hill, 2012

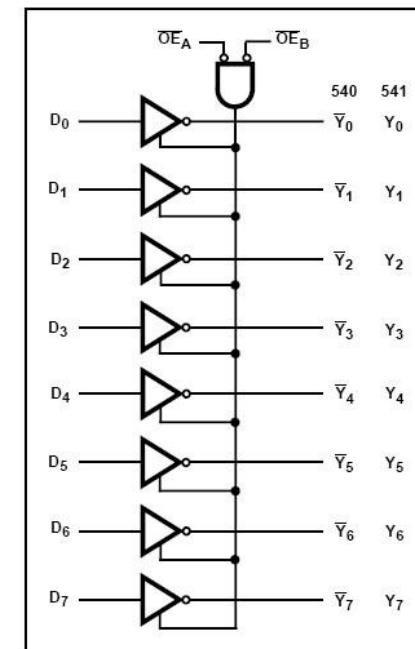
Registrul Paralel-Paralel octal (RPP)



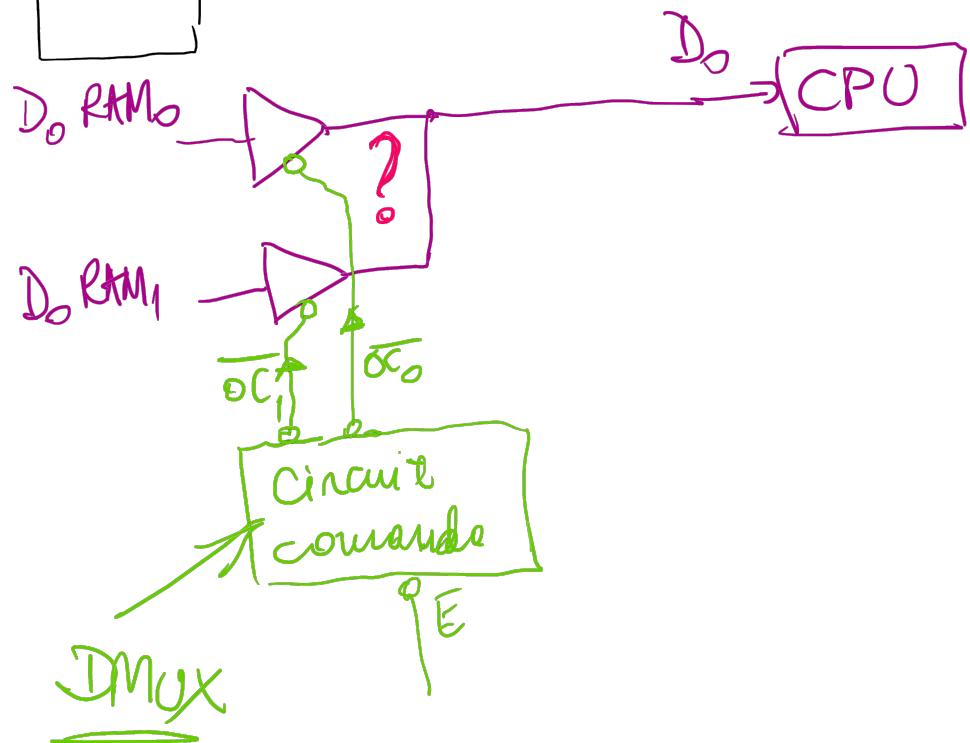
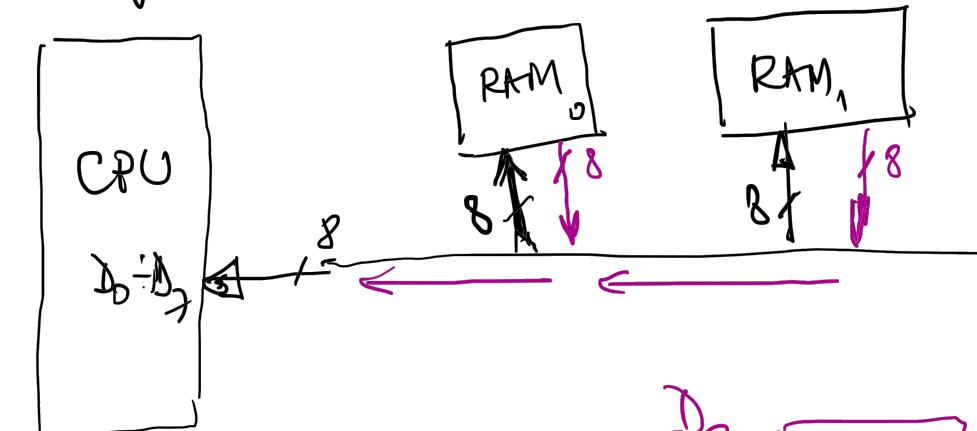
Buffer octal

\bar{OE} - output enable

74HCT541 Octal Buffer
Logic Diagram

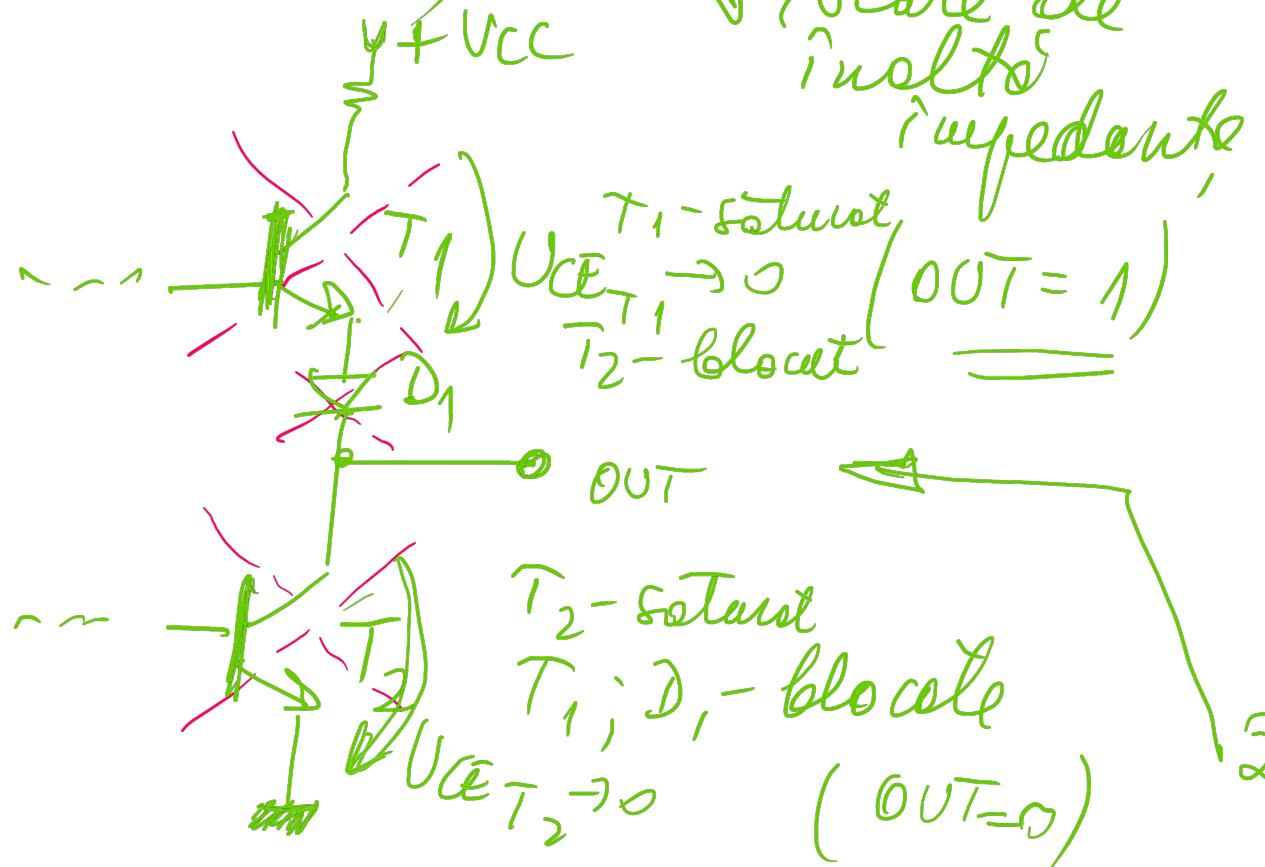


Etaje de ieșire TS (three state) în fizica de calcul



mai multe circuite cu etaje de ieșire TS se pot conecta în paralel dacă la un moment dat cel mult unul răngea direct scosă este activ, iar celealte sunt în stare TS.

Etoj ierre TS
(three state;
tree state)



2 stari logic
"0"-logic
"1"-logic
H: \mathbb{Z} high impedance
(\mathbb{Z})

stare de
inalter
impedanta

$U_{CE} \xrightarrow{T_1-\text{saturat}} 0$
 $T_2-\text{blocat}$

$(OUT = 1)$

$T_2-\text{saturat}$

$T_1; D_1 - \text{blocat}$

$(OUT = 0)$

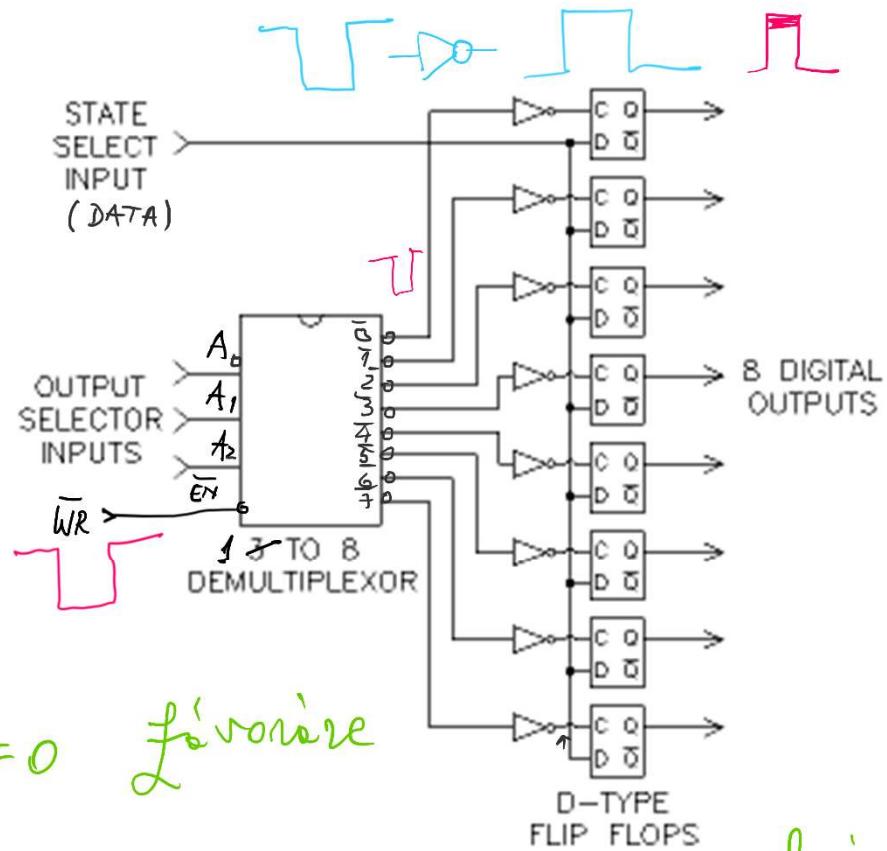
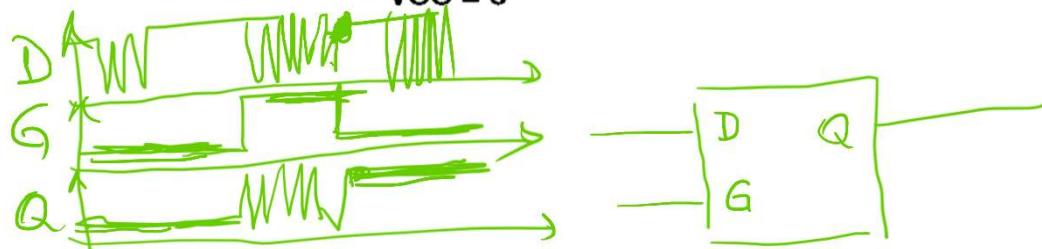
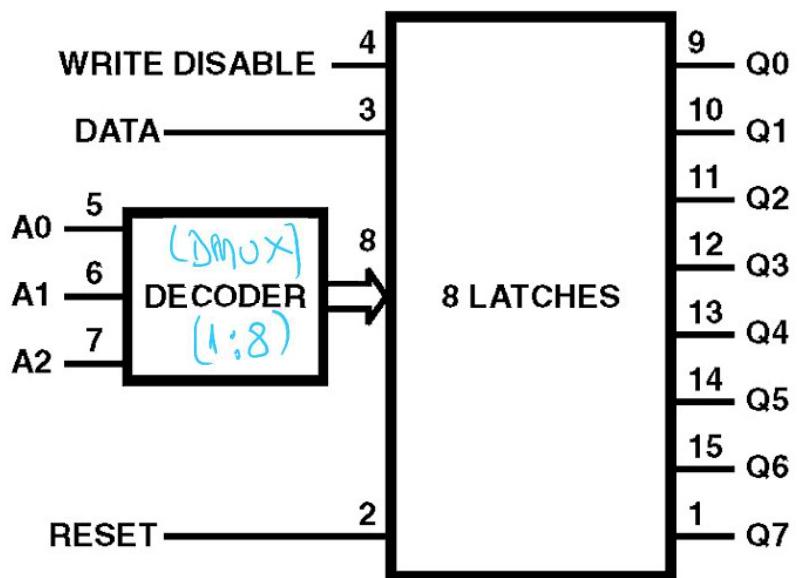
$T_1, T_2 - \text{blocate} \Rightarrow$

\Rightarrow stare de
inalter rezpect.

$Z_{out} \rightarrow \infty$

Latch octal adresabil 4099

LATCH ADRESABIL



- 1) $G = 1 \Rightarrow Q = D$ (orică schimbare pe linia D se transmite la linia Q)
- 2) $G \downarrow \rightarrow 0 \Rightarrow$ latch-ul memorizează

Un grup de celule de memorie adresate simultan intr-o operatie formeaza o locatie de memorie.

1 bit **b**

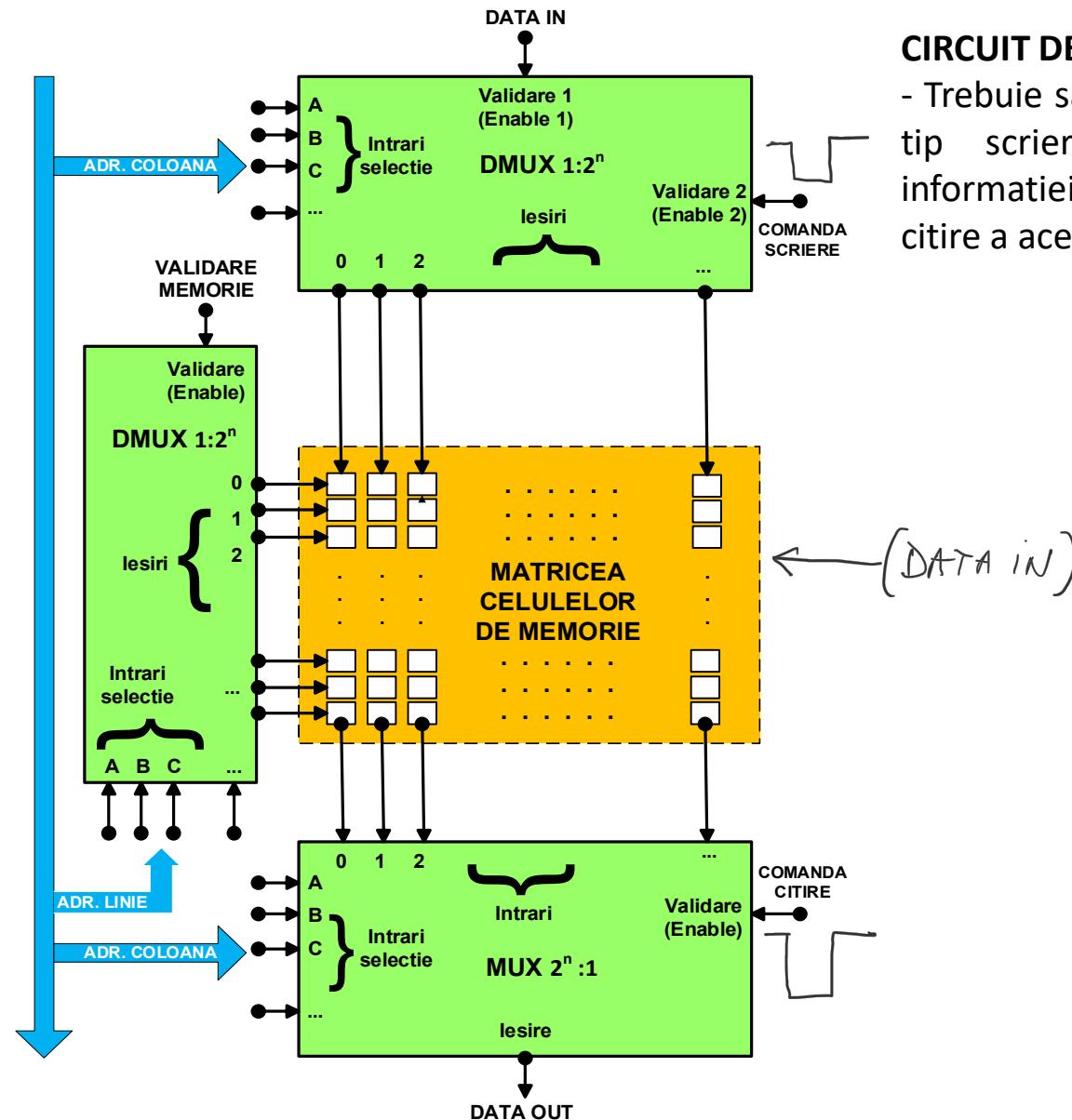
4 biti – nibble (tetradă)

8 biti – octet (byte) **B**

16 biti – cuvant (Word) **W**

32 biti – double word **DW**

64 biti – quad word **QW**



CIRCUIT DE MEMORIE

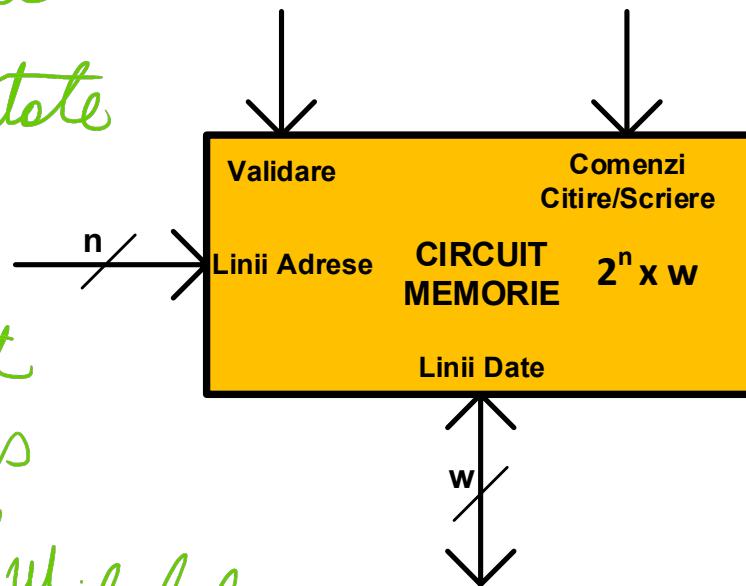
- Trebuie sa permita operatii de tip scriere (cu memorarea informatiei), respectiv de tip citire a acestieia

$$10^3 \neq 2^{10}$$

$$1000 \neq 1024$$

SIMBOL CIRCUIT DE MEMORIE

locatii - celele de memorie accesate simultan
 n - linii adrese $\Rightarrow 2^n$ locatii adresabile
 w - celele de memorie selectate
 intr-o locatie



Organizare circuit
memorie = produs
 neefectuat intre
 numar locatii n si
 celele de memorie dintr-o locatie
 $2^n \times w$

$$2^{14} = 2^4 \cdot 2^{10} = 16K$$

$$2^{22} \cdot 2^{20} = 2 \cdot 2 = 4M$$

$$n = 10 \quad 2^{10} = 1K$$

$$2^{10} \times 8 = 1K \times 8 =$$

$$= 1K_8 = 1KB = 8Kb$$

(Org.) (cap)

$$2^{20} = 1M$$

$$2^{30} = 1G$$

$$2^{40} = 1T$$

Organizarea circuitului de memorie – produsul neefectuat dintre numarul locatiilor de memorie si numarul de biti dintr-o locatie.

Capacitatea circuitului de memorie – produsul efectuat dintre numarul locatiilor de memorie si numarul de biti dintr-o locatie.

Liniile de adrese ale unui circuit de memorie au notatia: $A_0, A_1, A_2, A_3, \dots$ (unde A_0 – reprezinta bcmps, iar linia de adrese notata cu indexul cel mai mare reprezinta bcms).

O memorie cu liniile de adrese notate $A_0 - A_{n-1}$ are n linii de adrese (A_0 – reprezinta bcmps, iar A_{n-1} este bcms). Cu n linii de adrese se pot adresa 2^n locatii de memorie.

Pentru numarul locatiilor de memorie se folosesc: $2^{10} = 1024$ – kilo (k); 2^{20} – Mega (M); 2^{30} – Giga (G); 2^{40} – Tera (T);

Liniile de date ale unui circuit de memorie au notatia: $D_0, D_1, D_2, D_3, \dots$ (unde D_0 – reprezinta bcmps, iar linia de date notata cu indexul cel mai mare reprezinta bcms).

O memorie cu liniile de date notate $D_0 - D_{w-1}$ are w linii de date (D_0 – reprezinta bcmps, iar D_{w-1} este bcms).

Pentru bitii dintr-o locatie de memorie avem notatiile: nibble sau tetrada – 4 biti; octet (Byte – B) – 8 biti; cuvant (Word – W) – 16 biti; dublu cuvant (Double Word – DW) – 32 biti; cvadruplu cuvant (Quad Word – QW) – 64 de biti.

Linii de control la circuite de memorie

La memorii ROM: \overline{CE} , \overline{OE} (mai apar si linii folosite la programarea circuitelor – nu le vom utiliza in acest curs);

La memorii RAM: \overline{CE} , \overline{WE} , \overline{OE}

\overline{CE} – chip enable – valideaza circuitul (ROM, RAM) pentru a se putea face operatii cu el – citirea, scrierea unei locatii adresate;

\overline{OE} – output enable - valideaza iesirile de date permitand trecerea intr-o stare activa din starea HiZ (la memorii ROM, RAM);

\overline{WE} – write enable - valideaza operatia de scriere intr-o locatie de memorie RAM adresata.

AUTOMATE

Automat – o structura digitala care poate evoluă într-o multime de stări logice, cu trecerea dintr-o stare anterioară într-o stare urmatoare, în conformitate cu un algoritm bine stabilit, tinând seama de valorile logice ale unor semnale de intrare și cu generarea unor semnale de ieșire.

Un automat este un cvintuplu: $A = (S, I, O, f, g)$ unde

S – multimea finită a stărilor posibile

I – multimea finită a semnalelor de intrare

O – multimea finită a semnalelor de ieșire

f – multimea funcțiilor de tranzitie care precizează stările viitoare în care ajunge automatul ca urmare a aplicării unui semnal de intrare ($f : S \times I \rightarrow S$)

g – multimea funcțiilor de ieșire care precizează semnalul de ieșire generat în cazul aplicării unui semnal de intrare

($g : S \times I \rightarrow O$)

Tipuri de automate: cu stări finite de tip Mealy, Moore, microprogramate

AUTOMATE MICROPROGRAMATE

Un Automat Microprogramat evoluează pe baza unui Microprogram scris într-o memorie (ROM) format din Microinstructiuni. Fiecare stăru din evoluția automatului îi corespunde o microinstructiune prin care se rezolvă mai multe sarcini: selectarea și testarea unui semnal de intrare specific stării respective, generarea corespunzătoare a unui semnal de ieșire (comanda), pregătirea trecerii în starea următoare, corespunzătoare algoritmului implementat.

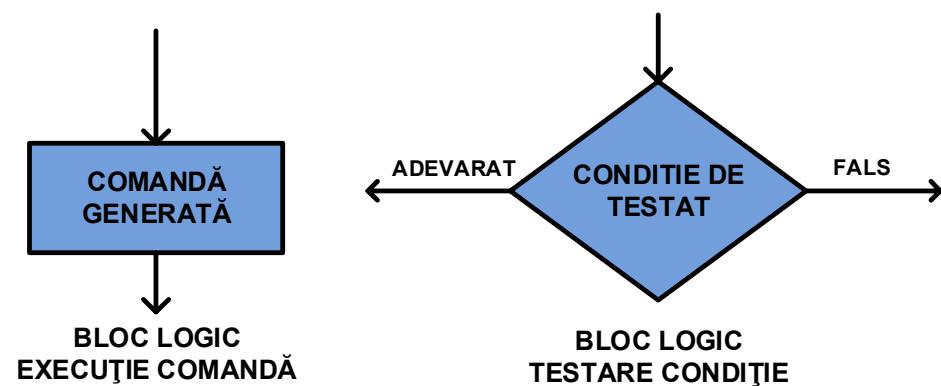
2 tipuri de automate microprogramate:

- cu format fix al microinstructiunilor
- cu format variabil al microinstructiunilor

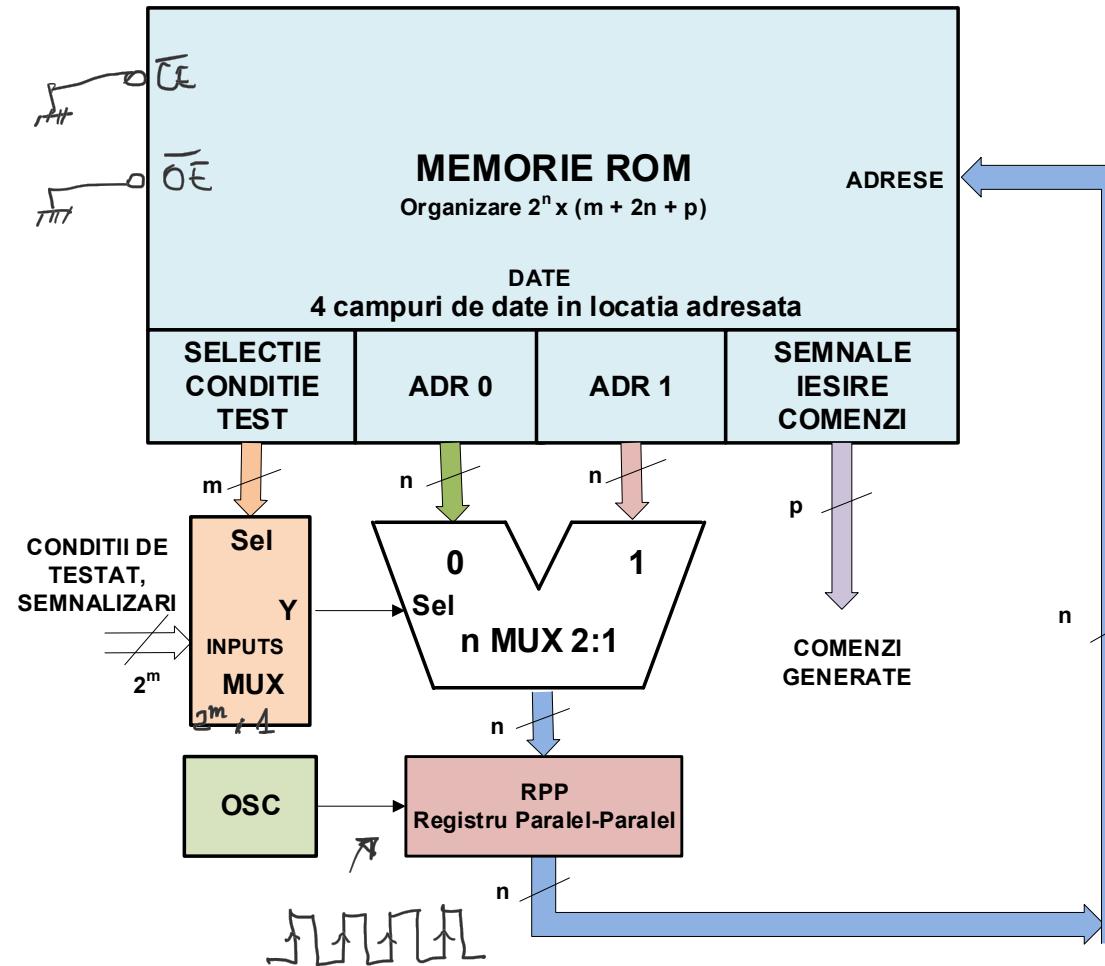
Functionarea unui automat se poate descrie prin mai multe forme:

- Limbaj natural;
- Diagrame de semnale;
- Diagrame de tranzitii a statilor si iesirilor;
- Scheme logice (organigrame).

Blocuri care apar in scheme logice (organigrame)



Automat microprogrammat cu format fix al microinstructiunilor



End 13 mar 2025