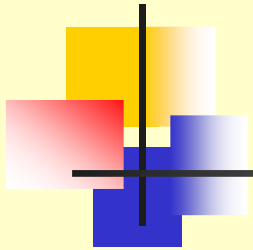


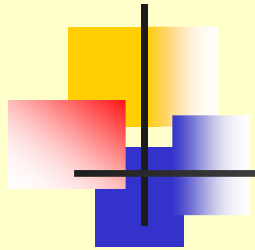


REGISTRE ȘI MEMORII RAM



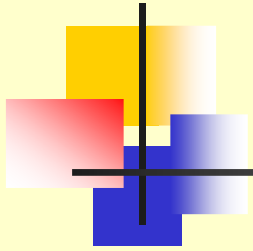
5.3. Registre

- **Definiție:** Registrele sunt circuite logice secvențiale (CLS) care permit stocarea și/sau deplasarea informației codificate binar
- Ele se realizează din celule de memorie binară (CBB) și din circuite logice combinaționale (CLC), care permit înscrierea, citirea și transferul informației
- **Capacitatea** unui registru este dată de numărul celulelor de memorie



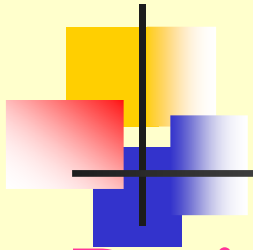
5.3. Registre

- **Clasificare**
- Registrele pot să fie de mai multe tipuri:
 - De memorie
 - De deplasare
 - Combinate
 - Universale



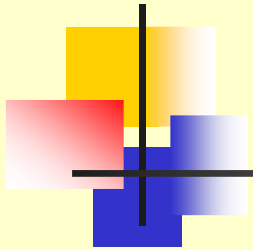
5.3. Registre

- **Registre de memorie**
- Memorează informația binară în celule de memorie binară
- În fiecare celulă de memorie se memorează 1 bit de informație
- Încărcarea în bistabile se poate face și paralel, prin intrările asincrone, de Set și Reset ale bistabilelor



5.3. Registre

- **Registre de deplasare**
- Realizează transferul informației
- Transferul se poate face:
 - stânga-dreapta
 - dreapta-stânga
 - reversibil – în ambele sensuri – cu o comandă
- La fiecare impuls de tact conținutul registrului se deplasează cu câte o celulă în sensul stabilit
- **Semnalul de ieșire este identic cu cel de intrare, dar întârziat** cu un număr de impulsuri de tact egal cu numărul de celule de memorie din care este format registrul



5.3. Registre

- **Registre de deplasare**
- Exceptând primul bistabil, ecuația de stare a unui registru de deplasare stânga-dreapta este dată de relația:

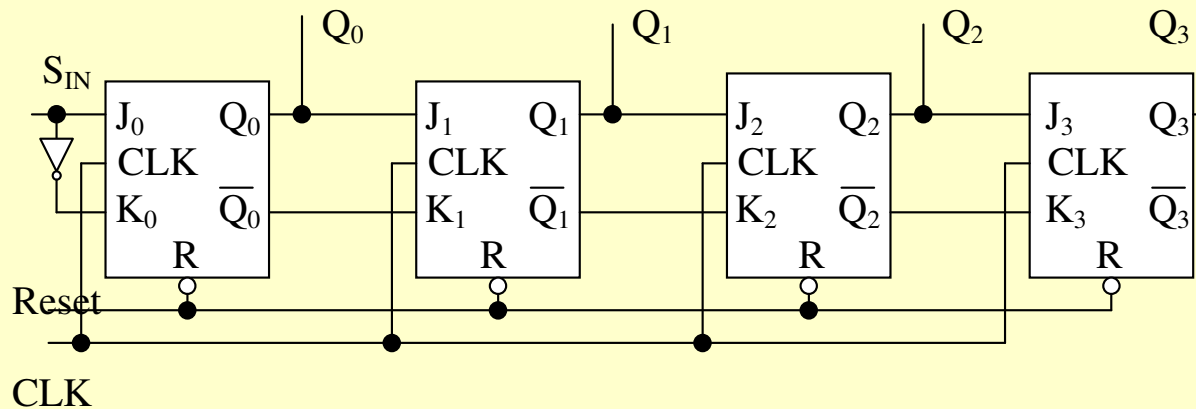
$$Q_i(t+1) = Q_{i-1}(t) \cdot CLK$$

(unde CLK = impulsul de tact)

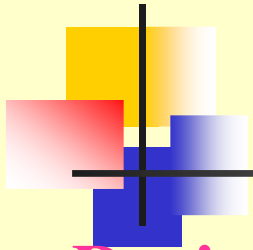
5.3. Registre

■ Registre de deplasare - exemplu

- Registru de deplasare stânga-dreapta cu bistabile JK



- La fiecare impuls de tact conținutul bistabilului Q_i se transferă în bistabilul Q_{i+1}
- În bistabilul Q_0 se introduce informația din exterior prin intrarea S_{IN} (serial input)
- Conținutul ultimului bistabil se pierde
- **Încărcarea** registrului se realizează în mod **serie**
- Inițializarea registrului se realizează prin semnalul de Reset (asincron), care forțează toate ieșirile registrului în 0 logic

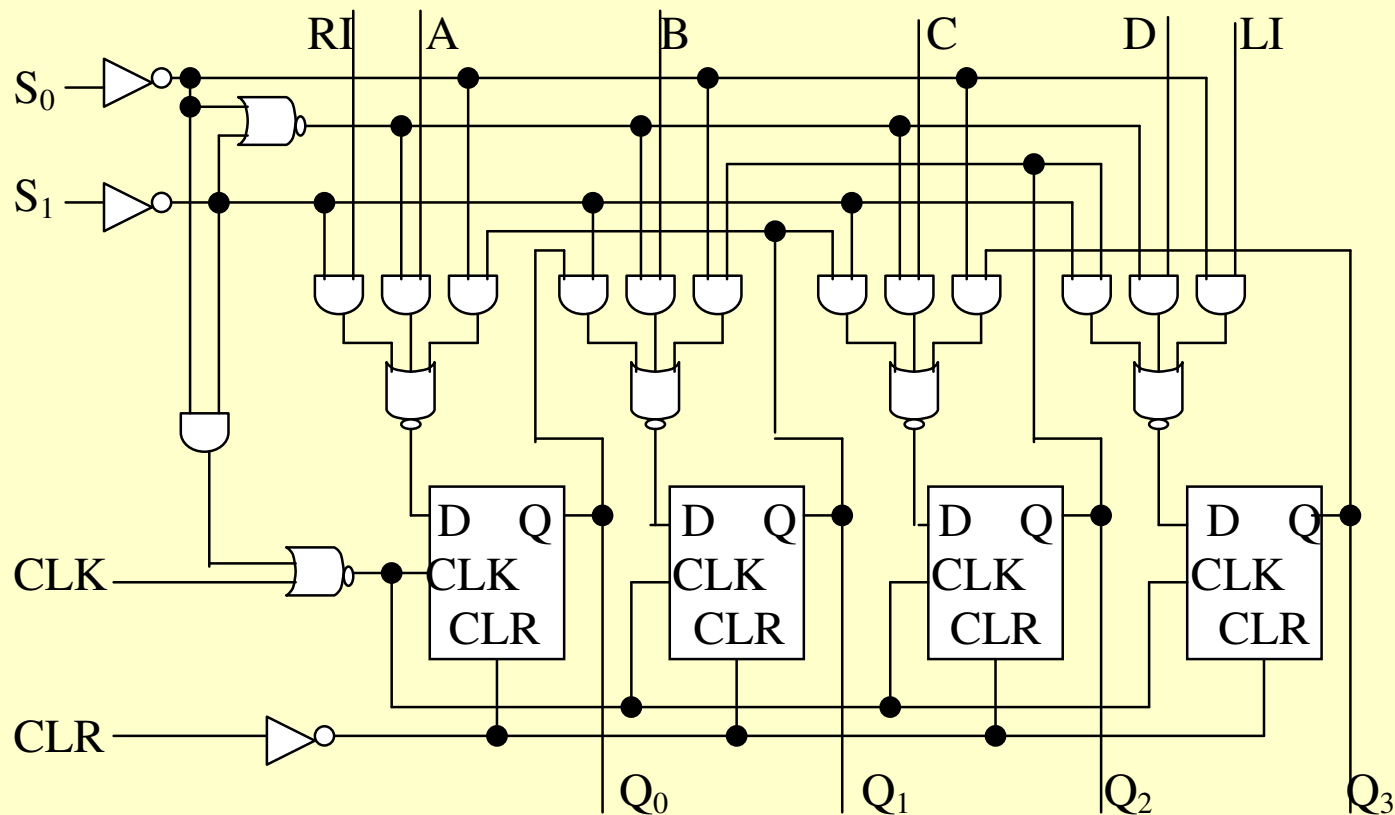


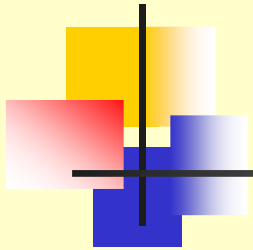
5.3. Registre

- **Registre de deplasare**
- Registrele de deplasare dreapta-stânga și reversibile se realizează folosind circuite logice combinaționale suplimentare
- **Registre combinate**
- Sunt cele care au și funcția de memorare și cea de deplasare
- **Registre universale**
- Cumulează toate funcțiile posibile ale unui registru:
 - deplasare stânga-dreapta
 - deplasare dreapta-stânga
 - încărcare serie sau paralelă a informației
 - citire serie sau paralelă a informației

5.3. Registre

- Registre universale - exemplu
- Registru universal pe 4 biți

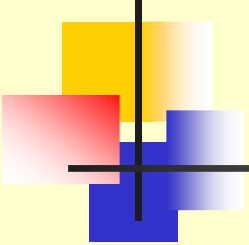




5.3. Registre

- **Registre universale - exemplu**
- **Registru universal pe 4 biți**
- Intrările de selecție S_1S_0 condiționează modul de funcționare a registrului:
 - $S_1S_0 = 00$ memorare – păstrează conținutul nemodificat
 - $S_1S_0 = 01$ deplasare stânga-dreapta
 - $S_1S_0 = 10$ deplasare dreapta-stânga
 - $S_1S_0 = 11$ încărcare paralelă
- Acționarea registrului se face sincron prin semnalul de CLK aplicat simultan pe toate bistabilele interne
- Ștergerea registrului se face asincron, prin semnalul CLR
- Încărcarea paralelă se realizează prin intrările paralele A, B, C, D
- Deplasarea stânga-dreapta se realizează prin încărcare serială pe intrarea RI
- Deplasarea dreapta-stânga se realizează cu intrarea serială LI

5.3.1. Aplicații ale registrelor

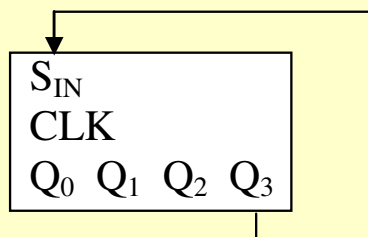
- 
- Registrele sunt utilizate în mai multe tipuri de aplicații, după funcțiile pe care pot să le îndeplinească:
 - Registre de deplasare cu reacție
 - Memorie FIFO
 - Memorie LIFO
 - Conversie serie–paralel și paralel–serie a informației
 - Generatoare de secvențe

5.3.1. Aplicații ale registrelor

- **Registre de deplasare cu reacție**
- Au ieșirile conectate la intrări
- Pot fi:
 - Registre de deplasare **în inel** – conținutul ultimei celule de memorie se înscrie în prima celulă de memorie
 - Registre (numărătoare) **Johnson** – în prima celulă se introduce conținutul negat al ultimei celule

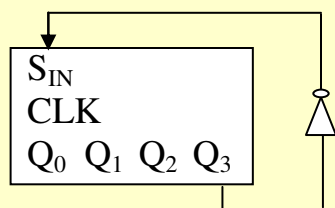
5.3.1. Aplicații ale registrelor

- **Registre de deplasare cu reacție**
- **Registre de deplasare în inel**



Q_0	Q_1	Q_2	Q_3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

- **Registre (numărătoare) Johnson**



Q_0	Q_1	Q_2	Q_3
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
0	0	0	0

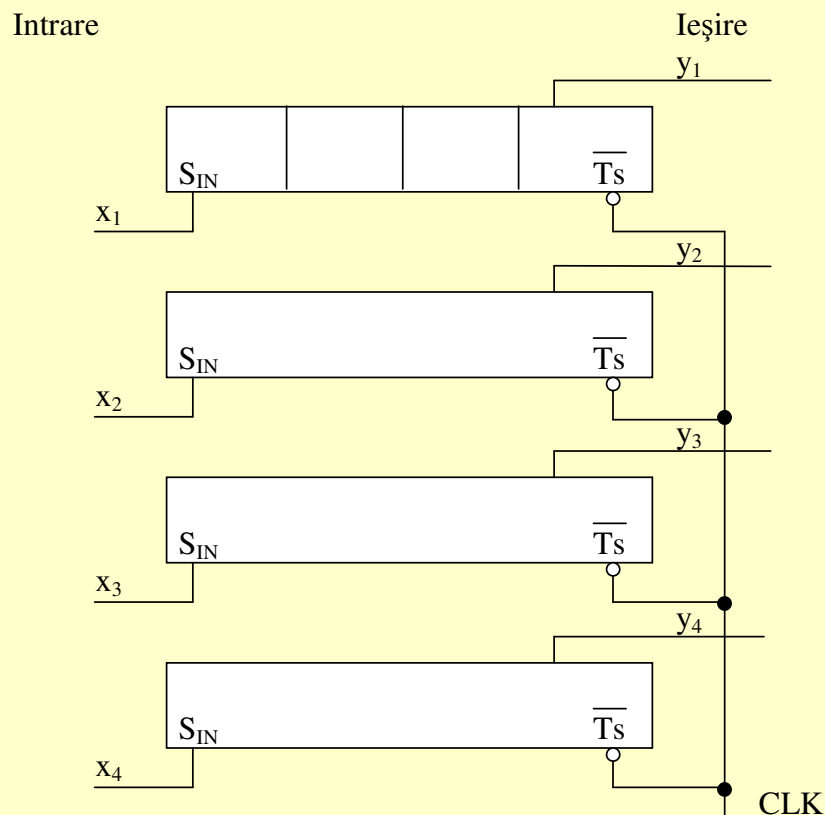
5.3.1. Aplicații ale registrelor

- **Memorie FIFO** (First In, First Out), primul înscris – primul citit (memorie coadă)
- Se realizează cu registre de deplasare stânga-dreapta
- Numărul registrelor utilizate depinde de lungimea cuvintelor ce urmează a fi memorate
- Capacitatea memoriei FIFO depinde de lungimea registrelor
- Exemplu: dacă registrele sunt de 8 biți, capacitatea memoriei este de 8 cuvinte
- La fiecare impuls de tact în memorie se introduc datele (cuvintele de memorat) utilizând intrarea serială a fiecărui registru
- Ieșirea (data memorată) se citește de pe ultimele ieșiri ale registrelor utilizate

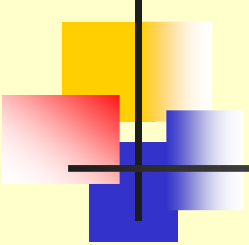
5.3.1. Aplicații ale registrelor

■ Memorie FIFO

■ Exemplu: Schema bloc a unei memorii FIFO pe 4 biți



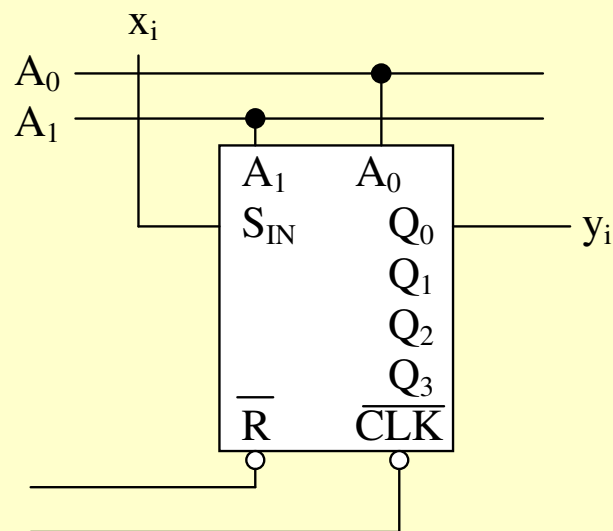
5.3.1. Aplicații ale registrelor

- 
- **Memorie LIFO** (Least In, First Out), ultimul introdus – primul citit (memorie stivă)
 - Realizarea memoriilor LIFO se face cu registre combinate
 - Numărul registrelor utilizate este dat de lungimea cuvântului de memorat
 - Lungimea registrelor determină capacitatea de memorie
 - Funcționarea registrelor din memoria stivă trebuie condiționată cu selecții, care să permită efectuarea celor 2 operații care apar la acest tip de memorie

5.3.1. Aplicații ale registrelor

■ Memorie LIFO

■ Exemplu: Schema bloc a unei celule de memorie LIFO



A_1	A_0	
0	1	deplasare stânga-dreapta → înscriere
1	0	citire

5.3.1. Aplicații ale registrelor

- **Conversie serie – paralel și paralel – serie a informației**
- **Conversia serie-paralel**
 - Se face încărcarea unui registru de deplasare prin intrarea serială și cu comandă de tact
 - Apoi se citește deodată informația de pe ieșirile registrului, în paralel
- **Conversia paralel-serie**
 - Se face întâi încărcarea paralelă a informației în registru, cu comandă de tact
 - Apoi se deplasează informația stânga-dreapta, cu comandă de tact și se citește informația serial de pe ultima ieșire

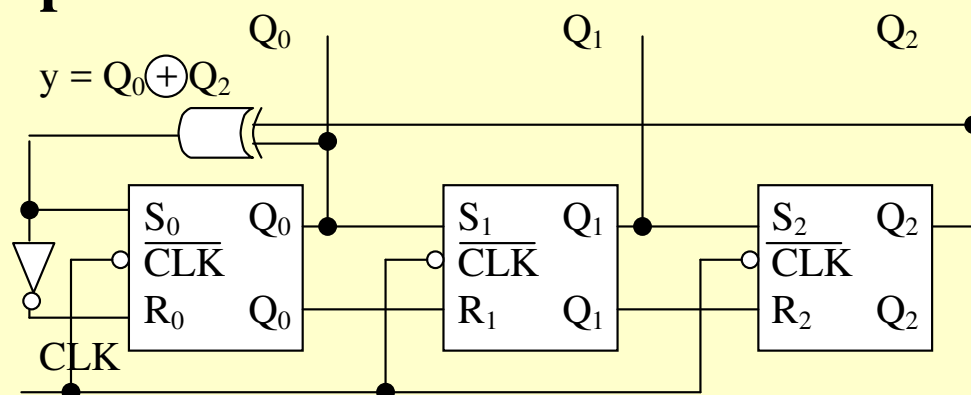
5.3.1. Aplicații ale registrelor

■ Generatoare de secvențe

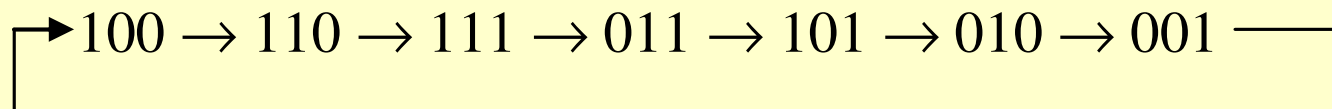
- Generatoarele de secvențe generează o succesiune de secvențe binare (succesiune de 1 și 0) de lungime dată (prestabilite)
- Lungimea secvenței reprezintă numărul după care se repetă întreaga secvență
- Secvențele binare pot fi:
 - Aleatoare, de lungime infinită
 - Deterministe, de lungime finită
- Secvențele binare deterministe de lungime maximă se numesc **secvențe pseudoaleatoare**
- Realizarea generatoarelor de secvențe se face cu registre care au în calea de reacție circuite logice combinaționale adecvate

5.3.1. Aplicații ale registrelor

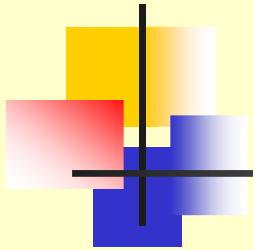
- Generatoare de secvențe pseudoaleatoare
- Exemplu



- Secvența pseudoaleatoare generată la ieșirile $Q_0Q_1Q_2$ este:

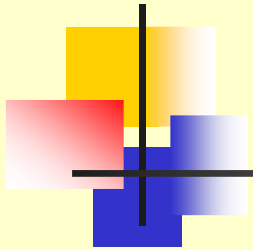


- Lungimea secvenței pseudoaleatoare este de 7
- **Observație:** Dacă sistemul pornește din starea inițială 000 se va bloca în această stare! => autocorecție



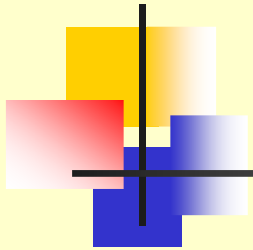
5.4. Memorii RAM

- Memoriile de tip RAM (random access memory) sunt memorii de tip citește-scrie, cu acces aleatoriu
- Tipuri de memorii RAM:
 - **SRAM** (static RAM)
 - **DRAM** (dynamic RAM)
- SRAM - după înscriserea unei informații într-o locație aceasta se păstrează atâta timp cât memoria este alimentată și nu se face o nouă scriere
- DRAM - datele înscrise trebuie reactualizate (refresh) periodic, prin citirea și reînscriserea lor, altfel dispar, chiar în prezența tensiunii de alimentare



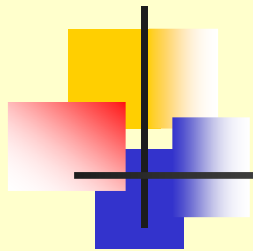
5.4. Memorii RAM

- În general memoriile RAM sunt **volatile**, nu-și păstrează informația după întreruperea tensiunii de alimentare
- Există memorii CMOS statice (nevolatile) prevăzute cu baterie, care pot să-și păstreze informațiile
- Memorii RAM nevolatile
 - feroelectrice - combinarea elementelor magnetice și electronice
 - nanotuburi de carbon



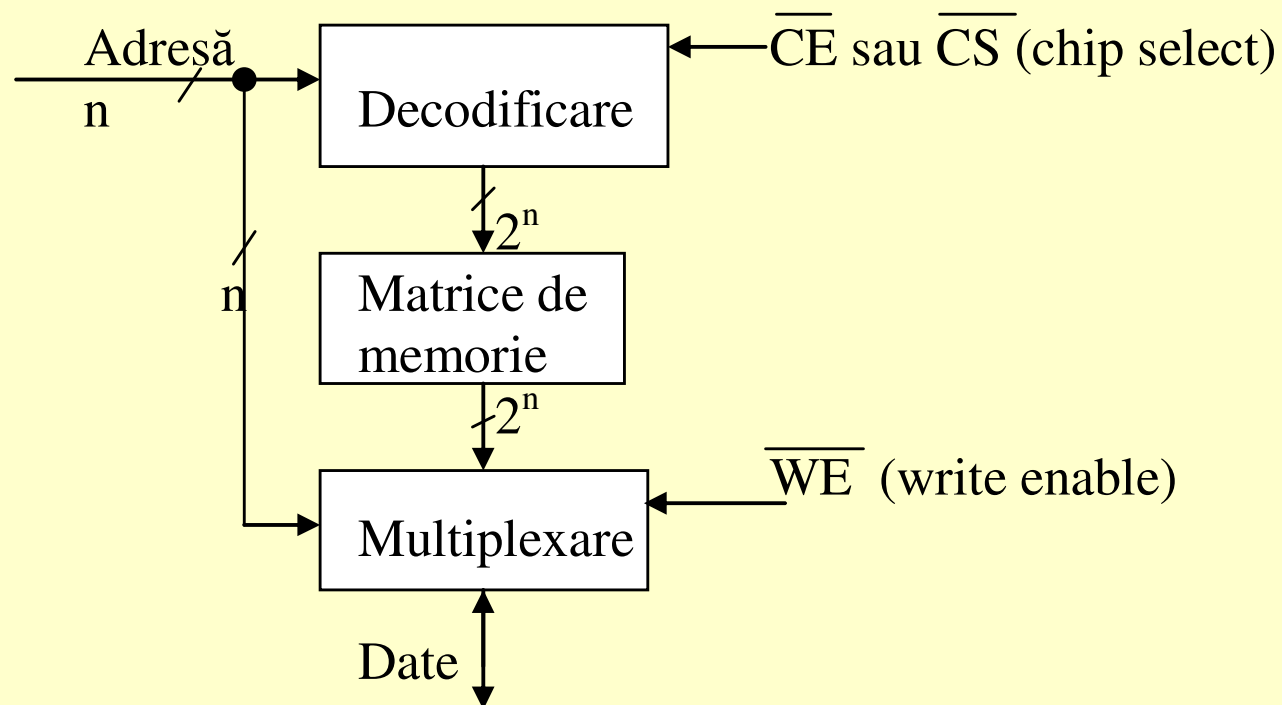
5.4. Memorii RAM

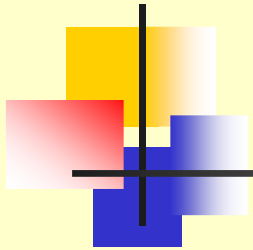
- **Dimensiunea** memoriei este specificată prin:
 - numărul de cuvinte (stabilit cu ajutorul adreselor)
 - numărul de biți pe cuvânt
- **Capacitatea** memoriei este dată de numărul total de biți memorați



5.4.1. Memorii SRAM

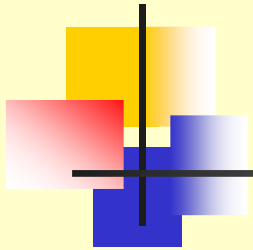
- **Schema funcțională** de principiu a unei memorii SRAM





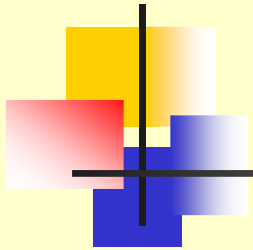
5.4.1. Memorii SRAM

- Memoria SRAM este formată din:
 - nivel de decodificare
 - matrice de memorie realizată cu celule de memorie binară de tip “latch”
 - nivel de multiplexare
- Decodificarea are ca intrări adresele memoriei și acționează pentru selectarea fiecărui cuvânt de memorie
- Multiplexarea permite selectarea oricărei celule de memorie la intrare sau la ieșire



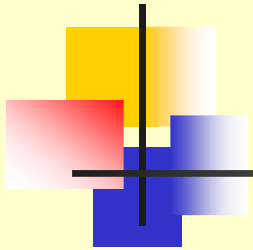
5.4.1. Memorii SRAM

- Validarea memoriei se face prin intrarea de enable CS (chip select)
- Pentru citirea și scrierea memoriei de folosește un **semnal de comandă** suplimentar **WE**:
 - Dacă $WE = 1$ se face citire
 - Dacă $WE = 0$ se face scriere
- Datele de intrare și ieșire pot să folosească linii diferite sau aceeași linie bidirecțională
- Memoriile pot avea un număr diferit de căi de date (de obicei cuvinte de un număr de biți multiplu de 2)



5.4.1. Memorii SRAM

- Memorii SSRAM (synchronous SRAM) - o variantă îmbunătățită de memorii SRAM
 - Folosesc tot latch-uri în matricea de memorie
 - Există în plus o interfață pentru control, adrese și date, controlată prin semnal de tact (CLK)
- Extinderea capacității memoriilor se face:
 - Prin extinderea dimensiunii cuvântului de memorie (numărul de biți din cuvânt)
 - Și prin extinderea numărului de cuvinte ale memoriei (adresa de memorie)



5.4.2. Memorii DRAM

- Memoriile DRAM sunt realizate cu condensatoare și tranzistoare MOS (pentru a fi mai rapide)
- Ele au nevoie de reîmprospătarea informației memorate la diferite intervale de timp, pentru a se evita pierderea ei
- Au 2 intrări de comandă în plus:
 - **RAS** (row address strobe)
 - **CAS** (column address strobe)
- Există și memorii SDRAM (synchronous DRAM), care au o interfață controlată cu semnal de tact