

## Lucrarea 3: Aplicații cu registre de deplasare (Lumină dinamică)

### Scopul lucrării:

- Înțelegerea modului de funcționare a registrelor de deplasare;
- Utilizarea registrelor de deplasare în implementarea luminilor dinamice.

## I. Noțiuni teoretice

### I.1. Registre de deplasare

La nivel de bit, memorarea informației se face cu ajutorul latch-urilor sau a bistabililor. La nivel de cuvânt, memorarea și procesarea informației se face cu ajutorul registrelor. Registrul este un circuit logic secvențial format dintr-o succesiune de  $n$  latch-uri sau bistabili.

Încărcarea unui registru (operația de introducere a cuvântului de  $n$  biți în registru) se poate face în două moduri:

- *serial* – încărcarea se face bit cu bit în ritmul unui semnal de ceas, motiv pentru care timpul de încărcare este egal cu  $n$  perioade de ceas;
- *paralel* – toți biții sunt introduși în același timp, pe tranziția activă a unui semnal de ceas.

Extragerea informației din registru (operație denumită citire) se poate face tot în două moduri:

- *serial* – citirea se face bit cu bit pe  $n$  tranziții active ale unui semnal de ceas;
- *paralel* – toți biții sunt citiți în același timp.

În general, un registru este definit ca o structură liniară de celule de memorare ce prezintă una sau mai multe din următoarele funcții: - acces serial; - acces paralel; - ieșire serială; - ieșire paralelă.

Registrele cu încărcare serială mai sunt denumite și *registre de deplasare* și, la rândul lor, pot fi împărțite în două categorii: a) registre cu deplasare la dreapta și b) registre bidirecționale.

Registrele prevăzute cu ambele tipuri de intrări (serie și paralel) și ieșire paralelă sunt denumite registre universale.

Încărcarea serială a unui registru pe 4 biți este ilustrată în figura 1.

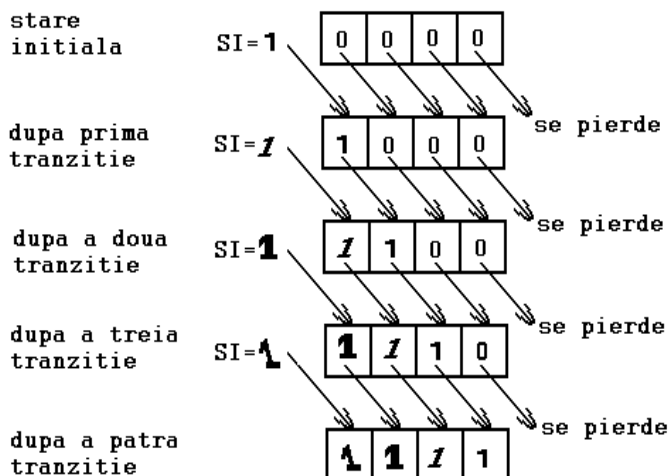


Fig. 1. Exemplu de încărcare serială a unui registru de 4 biți

### 1.2. Prezentarea circuitul 74198, registru universal pe 8 biți

Circuitul 74198 este un registru universal reversibil pe 8 biți. Modul de lucru al acestui circuit este prezentat în tabelul de mai jos.

Circuitul **74LS198** – registru universal pe 8 biți.

#### Caracteristici:

- modul de lucru este decis de codul binar aplicat pe intrările de selecție  $S_1 S_0$ , după cum urmează:
  - $S_1 S_0 = 00$ , memorare a informației existente în registru;
  - $S_1 S_0 = 01$ , deplasare serială de la  $Q_A \rightarrow Q_H$ , intrarea de date este SRSER;
  - $S_1 S_0 = 10$ , deplasare serială de la  $Q_H \rightarrow Q_A$ , intrarea de date este SLSER;
  - $S_1 S_0 = 11$ , încărcare paralelă sincronă ( $Q_H \dots Q_B Q_A = H \dots BA$ );
- intrarea de ștergere asincronă ( $\overline{CLR}$ ) este activă pe zero logic;
- operațiile de deplasare la stânga, deplasare la dreapta și încărcare paralelă sunt efectuate sincron cu semnalul de ceas  $CLK$ , pe tranziția pozitivă;
- pentru comanda de memorare, semnalul de ceas nu are efect asupra funcționării circuitului;

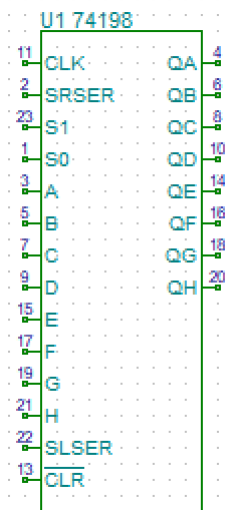


Fig.2: Circuitul 74198

**Atenție!** Schimbarea stării logice de pe intrările de selecție a modului de lucru trebuie realizată doar pe nivelul de unu logic al semnalului de ceas.

**I.3. Aplicația 1:****Realizarea unei lumini dinamice de tip "Progress bar" pe o bareță de 8 LED-uri**

Cu un registru universal pe 8 biți (spre exemplu, circuitul 74LS198), realizați o aplicație de tip "progressbar", pe o bareță formată din 8 LED-uri.

În funcționarea acestei aplicații distingem două etape:

- prima etapă constă în "umplerea" progresivă a baretei de LED-uri
- cealaltă etapă constă în "golirea" progresivă a baretei.

Modul de lucru este ilustrat grafic în figura 3, în care, LED-ul aprins este simbolizat printr-un punct negru.

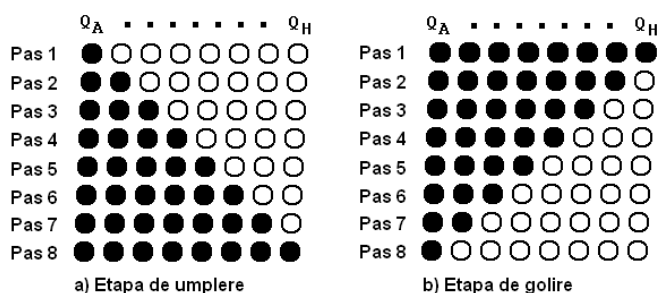
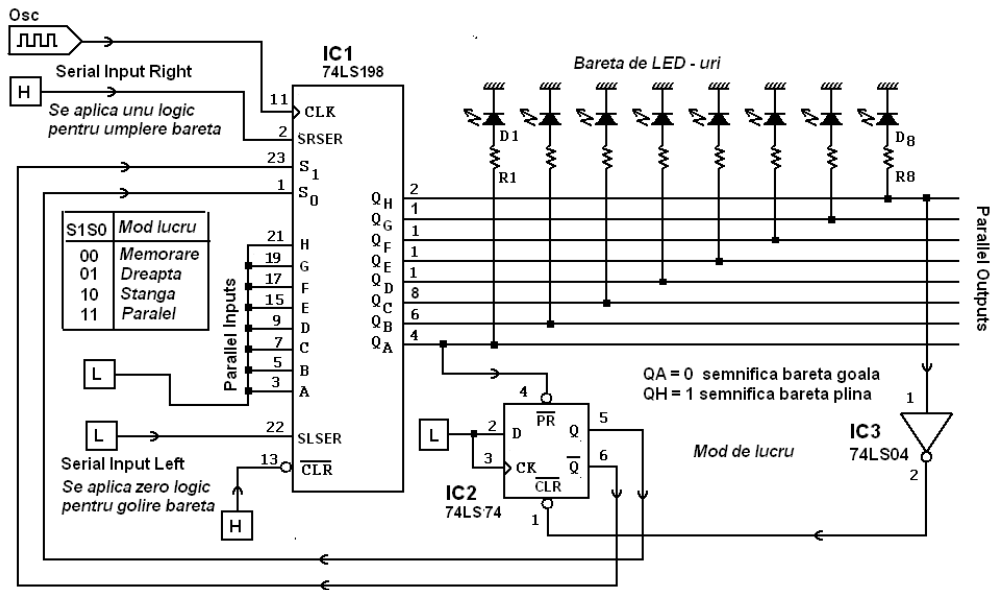


Fig. 3. Modul de umplere/golire a baretei de LED-uri

Pentru realizarea acestei aplicații trebuie să ținem cont de următoarele aspecte:

- umplerea/golirea baretei de LED-uri se face în ritmul unui semnal de ceas aplicat pe intrarea  $CLK$  a registrului de deplasare. Frecvența acestui semnal determină viteza de umplere/golire a baretei de LED-uri;
- umplerea progresivă a baretei de LED-uri se poate face printr-o comandă de încărcare serială cu deplasare la dreapta ( $S_1S_0=01$ ), dublată de aplicarea valorii de zero logic pe intrarea serială de date alocată pentru deplasarea la dreapta *Shift Right Serial* ( $SRSER=1$ );
- golirea progresivă a baretei de LED-uri se poate face printr-o comandă de încărcare serială cu deplasare la stânga ( $S_1S_0=10$ ), dublată de aplicarea valorii de zero logic pe intrarea serială de date alocată pentru deplasarea la stânga, *Shift Left Serial* ( $SLSER=0$ );
- comanda de umplere a baretei (deplasare la dreapta) trebuie menținută activă până când ieșirea  $QH=1$ ;
- comanda de golire a baretei (deplasare la stânga) trebuie menținută activă până când ieșirea  $QA=0$ ;
- putem constata că:
  - ieșirea  $QH$ , poate fi utilizată ca un senzor, activ pe unu logic, ce indică umplerea completă a baretei de LED-uri;
  - ieșirea  $QA$ , poate fi utilizată ca un senzor, activ pe zero logic, ce indică golirea completă a baretei de LED-uri;
  - deoarece un "senzor" este activ pe zero logic iar celălalt pe unu logic, avem nevoie de un inversor ca să le aducem, pe ambele, să fie active pe zero logic;

- pentru acțiunea de umplere registrul trebuie să primească comanda  $S_1S_0=01$ , iar pentru cea de golire trebuie să primească comanda  $S_1S_0=10$ . Din acest motiv, pe lângă registru, trebuie să mai adăugăm o logică suplimentară care să genereze semnale corecte pentru comanda intrărilor ce stabilesc modul de operare al registrului, ne referim aici la intrările  $S_1S_0$ ;



*Fig. 4. Aplicație de tip "progress bar" (umplere/golire bareta de LED-uri) cu registrul universal 74LS198:*

Rolul componentelor este următorul:

- registrul de deplasare este piesa centrală a schemei, este utilizat pentru comanda baretei de LED-uri;
- bistabilul de tip D este folosit pentru generarea și memorarea semnalelor ce stabilesc modul de lucru al registrului.
- Modificarea stării bistabilului se face doar în două situații:
  - când registrul este gol avem  $QA=0$  și  $QH=0$ , situație care aduce bistabilul în starea  $Q=1$ , iar mai departe se generează spre registru o comandă de deplasare serială spre dreapta. Această situație este menținută nemodificată până la umplerea baretei (aprinderea completa).
  - când registrul este plin avem  $QA=1$  și  $QH=1$ , situație care aduce bistabilul în starea  $Q=0$ , iar mai departe se generează spre registru o comandă de deplasare serială spre stânga. Această situație este menținută nemodificată până la golirea completa a baretei (stingerea completa).
- inversorul este folosit pentru a modifica nivelul activ al "senzorului" de bară plină;

Modul de conectare a componentelor prezentate anterior se poate vedea în figura 4.

**I.4. Aplicația 2:****Realizarea unei lumini dinamice în care se deplasează un singur punct luminos de la un capăt la altul al baretei de 8 LED-uri**

Pe baza unui registru universal pe 8 biți (spre exemplu, circuitul 74LS198), realizați o aplicație, în care, un singur punct luminos să se deplaseze de la stânga la dreapta iar apoi de la dreapta la stânga, pe o bară formată din 8 LED-uri.

În funcționarea acestei aplicații distingem două etape: în prima etapă un singur punct luminos (LED aprins) se deplasează de la stânga la dreapta iar în etapa a doua, punctul luminos se deplasează în sens invers. Modul de lucru este ilustrat grafic în figura 5, în care, LED-ul aprins este simbolizat printr-un punct negru.

Modul de funcționare pentru această aplicație este ilustrat în figura 5.

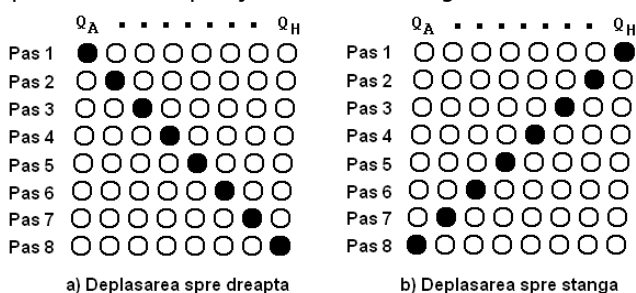


Fig. 5. Modul de deplasare a punctului luminos pe bareta de LED-uri

În realizarea acestei aplicații trebuie să ținem cont de următoarele aspecte:

- deplasarea punctului luminos pe bareta de LED-uri se face în ritmul unui semnal de ceas aplicat pe intrarea CLK a registrului de deplasare. Frecvența acestui semnal determină viteza de mișcare a punctului luminos;
- deplasarea de la stânga la dreapta a punctului luminos se poate face dacă avem simultan îndeplinite următoarele două acțiuni:
  - comandă de încărcare serială cu deplasare la dreapta ( $S_1S_0=01$ );
  - pe intrarea serială *Shift Right Serial* (*SRSER*) se aplică valoarea de unu logic în prima perioadă a semnalului de ceas iar apoi se aplică doar valori de zero până când punctul luminos ajunge în  $Q_H$ ;
- deplasarea de la dreapta la stânga a punctului luminos se poate face dacă avem simultan îndeplinite următoarele două acțiuni:
  - comandă de încărcare serială cu deplasare la stânga ( $S_1S_0=10$ );
  - pe intrarea serială *Shift Left Serial* (*SLSER*) se aplică doar valori de zero logic până când punctul luminos ajunge în  $Q_A$ ;
- comanda de deplasarea de la stânga la dreapta trebuie menținută activă până când ieșirea  $Q_H=1$ ;
- comanda de deplasarea de la dreapta la stânga trebuie menținută activă până când ieșirea  $Q_A=1$ ;
- putem constata că:
  - ieșirea  $Q_H=1$ , poate fi utilizată ca un senzor, activ pe unu logic, ce indică momentul de începere a sensului de deplasare de la dreapta la stânga;

- ieșirea  $QA=1$ , poate fi utilizată ca un senzor, activ pe unu logic, ce indică momentul de începere a sensului de deplasare de la stânga la dreapta;
- la fel ca în exemplul anterior, este nevoie de un bistabil pentru comanda intrărilor  $S_1S_0$ ;
- deoarece informația serială ce trebuie aplicată pe intrarea serială  $SRSER$  se modifică în timp, este nevoie de un bistabil suplimentar față de aplicația anterioară;
- deoarece avem "senzori" activi pe zero logic iar comenzile de set/reset la bistabili sunt active pe zero logic, avem nevoie de două inversoare;
- spre deosebire de aplicația anterioară, acum avem nevoie și de o logică de inițializare a întregii scheme, prin acționarea temporară a unui comutator cu revenire;

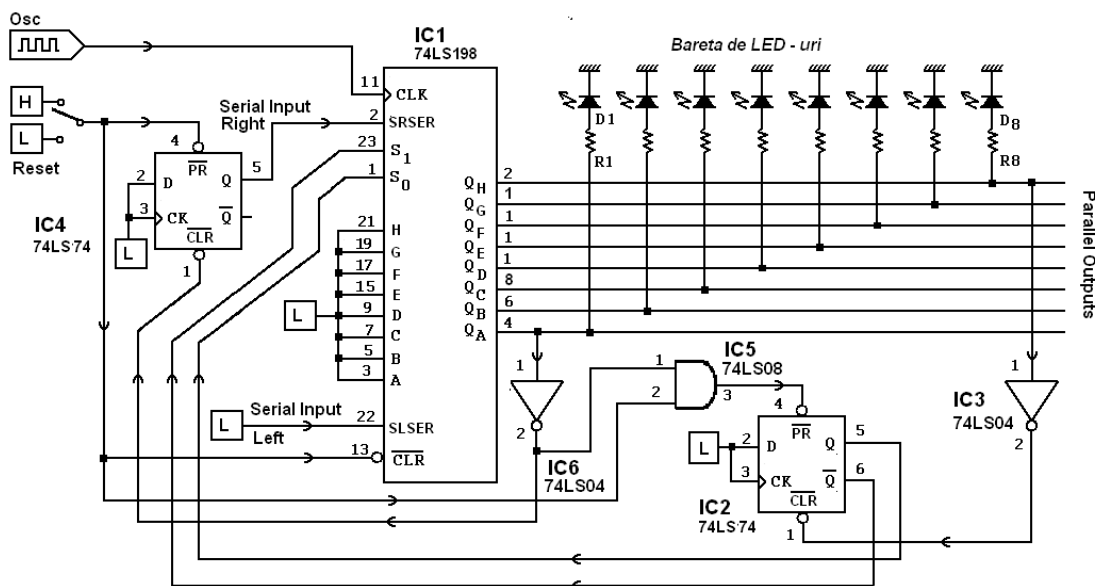


Fig. 6. Aplicație cu 74LS198: deplasare stânga/dreapta a unui singur LED aprins

Rolul componentelor este următorul:

- registrul de deplasare este piesa centrală a schemei, este utilizat pentru comanda baretei de LED-uri;
- bistabilul de tip D din circuitul IC2, este folosit pentru generarea și memorarea semnalelor ce stabilesc modul de lucru al registrului;
- bistabilul de tip D din circuitul IC4, este folosit pentru schimbarea informației aplicate la intrarea serială  $SRSER$ ;
- inversoarele IC3 și IC6 sunt folosite pentru a schimba nivelul activ al "senzorilor" ce indică atingerea capetelor de baretă de către punctul luminos;
- poarta AND din IC5 este folosită pentru preluarea celor două comenzi de preset necesare bistabilului IC2 (o comandă de preset de la circuitul de inițializare iar cealaltă comandă de preset vine de la inversorul IC6);

Modul de lucru este următorul:

- **partea de inițializare** este activată la apăsarea temporară a butonului RESET, acțiune ce are următoarele efecte:
  - ștergerea informației din registrului de deplasare ( $QA=QB= \dots QH=0$ );
  - aducerea bistabilului IC2 în starea  $Q=1$ , cea ce are ca efect apariția unei comenzi de deplasare spre dreapta ( $S_1S_0=01$ );
  - aducerea bistabilului IC4 în starea  $Q=1$ , în vederea introducerii valorii inițiale de unu logic pe intrarea serială *SRSER*;
- **deplasarea de la stânga la dreapta** a punctului luminos:
  - începe imediat după eliberarea butonului de RESET;
  - în această etapă avem  **$S_1S_0=01$** , comandă obținută în etapa de inițializare;
  - la prima tranziție activă a semnalului de ceas, în registru este introdusă valoarea de unu logic preluată din bistabilul IC4;
  - după prima tranziție activă a semnalului de ceas, valoarea de unu abia introdusă în registru determină  $QA=1$ , iar mai departe, prin IC6, determină trecerea în zero a bistabilului IC4. Din acest moment, pe intrarea *SRSER* a registrului se vor aplica numai valori de zero logic. Odată cu comanda de ștergere a bistabilului IC4 se mai generează și o comandă de set pentru bistabilul IC2, însă efectul ei este identic cu cel din etapa de inițializare (nu modifică starea logică a informației din IC2);
  - pentru următoarele perioade ale semnalului de ceas, valoarea de unu logic introdusă anterior, se deplasează spre dreapta și, în spatele ei, sunt introduse valori de zero;
  - terminarea acestei etape se face atunci când valoarea de unu logic ajunge în QH;
- **deplasarea de la dreapta la stânga** a punctului luminos:
  - începe imediat după ce punctul luminos a atins capătul din dreapta al barei, adică în momentul în care avem  $QH=1$ ;
  - în momentul în care  $QH=1$ , prin intermediul IC4 se generează o comandă de reset la bistabilul IC2. Din acest moment, registrul primește comanda  **$S_1S_0=10$** , ceea ce înseamnă o comandă de deplasare la stânga. Cu alte cuvinte începe deplasarea în sens invers.
  - pentru următoarele perioade ale semnalului de ceas, valoarea de unu logic care a ajuns în QH, este deplasată pas cu pas spre stânga până când ajunge în QA. În spatele valorii de unu sunt introduse valori de zero prin faptul că *SLSER* este conectat la masă. În acest fel ne asigurăm că avem doar un punct luminos deplasat spre stânga.
  - terminarea acestei etape se face atunci când valoarea de unu logic ajunge în QA;
- **reluarea deplasării de la stânga la dreapta** a punctului luminos:
  - începe imediat după ce punctul luminos atinge capătul stâng al baretei de LED-uri ( $QA=1$ );
  - evenimentul  $QA=1$ , are două efecte:
    - resetarea lui IC4, fără efect deoarece a mai fost resetat anterior;
    - presetarea lui IC2, ceea ce are ca efect schimbarea sensului de deplasare a informației din registrul de deplasare;
  - la reluarea deplasării de la stânga la dreapta, valoarea de unu nu mai este luată din IC4, este valoarea de unu care a ajuns în QA ca urmare a deplasării în sens invers din etapa anterioară;

Modul de conectare a componentelor prezentate anterior se poate vedea în figura 6.

## II. Desfășurarea lucrării

### II.1. Studiarea modului de lucru al registrului universal 74198

Se cere studiarea modurilor de lucru specifice registrului 74198, folosind schema de test de mai jos:

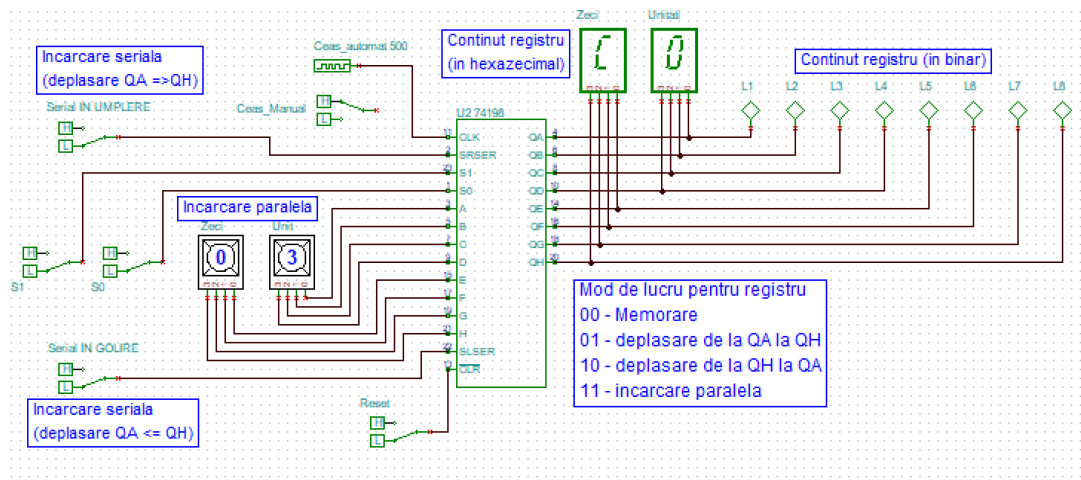


Fig. 7. Schemă pentru studiarea modurilor de lucru specifice circuitului 74198 (codul sursă pentru această schemă de test este disponibil pe platforma).

### II.2. Simularea aplicației "progress bar"

- Se cere simularea schemei din figura 4;

### II.3. Reproiectarea aplicației progress bar

- Se cere reproiectarea și apoi simularea schemei din figura 4, astfel încât bareta să aibă 16 LED-uri;

### II.4. Simularea aplicației deplasare „deplasare punct luminos”

- Se cere simularea schemei din figura 6;

### II.5. Reproiectarea aplicației „deplasare punct luminos”

- Se cere reproiectarea și apoi simularea schemei din figura 6, astfel încât bareta să aibă 16 LED-uri;

### II.6. Folosind un registru în inel, pe 8 biți, proiectați o lumină dinamică în care se deplasează segmente de 2 LED-uri aprinse și 2 LED-uri stinse

Indicații:

- Realizarea unui registru în inel presupune reintroducerea informației de pe ieșirea serială (adică QA sau QH, depinde de sensul de deplasare) pe intrarea serială (SRSER sau SLSER, depinde de sensul de deplasare ales);
- Informația ce trebuie deplasată se încarcă paralel;