

Aplicații ale circuitelor logice combinate

Circuitele logice combinate (CLC) sunt circuite fără memorie (ieșirile există doar în prezența semnalelor de intrare), iar semnalul de la ieșire y_k este funcție doar de semnalele de intrare: $y_k = f_k(x_1, x_2, \dots, x_n)$.

Sinteza circuitelor logice combinate constă în:

- definirea funcției (funcțiilor) prin stabilirea tabelului de adevăr;
- minimizarea expresiei logice a funcțiilor logice;
- implementare schemei.

Exemple de CLC

1. **Convertoare de cod** - fac trecerea de la un cod la altul (ex. binar - Gray; binar - BCD).

Problemă: Sa se implementeze un convertor binar – Gray pe trei biți.

2. **Codificatoare**

- codificator: la activarea unei intrări, la ieșire apare un cuvânt de cod;
- codificator de adresă: la activarea unei intrări, la ieșire apare adresa (numărul de ordine) intrării activate.

Exemplu: 7 intrări => adresă cu trei biți

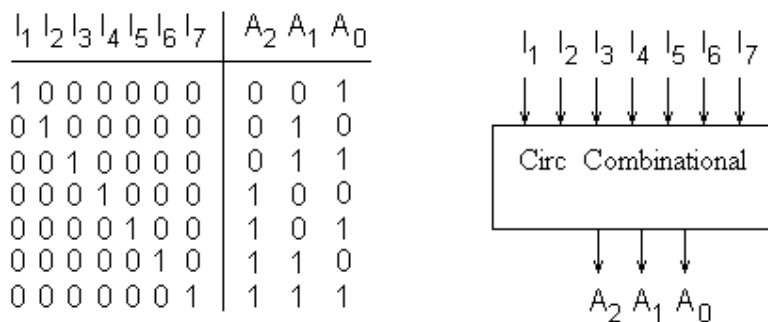


Fig.1. Codificator de 3 biți

$$A_0 = I_1 + I_3 + I_5 + I_7$$

$$A_1 = I_2 + I_3 + I_6 + I_7$$

$$A_2 = I_4 + I_5 + I_6 + I_7$$

Observație: Tabelul de adevăr este incomplet definit, foarte multe combinații din domeniul de definiție lipsind; dacă pentru combinațiile lipsă se completează cu indiferent și apoi relațiile s-ar minimiza cu diagrame Veight-Karnaugh ar rezulta aceleași ecuații.

Consecință: Dacă sunt activate simultan mai multe intrări vor rezulta adrese false.

Codificator de adresă prioritar

Să considerăm că nivelul activ de la intrare este simbolul 1 logic..

Un codificator de adresă prioritar va furniza la ieșire adresa intrării de rang maxim care este activată, vezi tabelul de adevăr de mai jos.

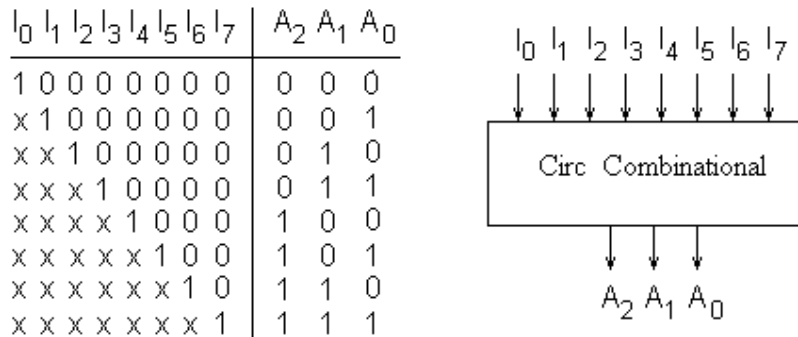


Fig.2. Codificator de adresă prioritar, de 3 biți

Observație: de data aceasta tabelul de adevăr este complet definit.

Circuitul astfel prezentat se poate completa cu

- intrare *chip select*
- ieșire care să semnaleze faptul că nu este activată nici o intrare

Problemă: Să se implementeze circuitul de mai sus și să se completeze circuitul cu EI (enable input), EO (enable output) sau CS (selectare chip).

- dacă EI este inactivă - circuitul furnizează codul 000 la ieșire și simultan semnalează la ieșirea EO (indiferent de codul de intrare).

- ieșirea EO – ieșire care semnalează când circuitul este blocat de EI sau când nici o intrare nu este activă.

- CS - este activă când circuitul este selectat și se activează cel puțin una din intrări (de fapt are o funcționare similară cu EO).

3. Decodificatoare

- Activează una sau mai multe ieșiri în funcție de cuvântul de cod de la intrare.

Utilizări: - adresarea memoriilor;

- afișarea numerică;
- în construcția multiplexoarelor.

Decodificator de adresă

Un decodificator de adresă are n intrări și 2^n ieșiri. La aplicarea unui cod pe intrările de adresă este activată **numai** ieșirea care corespunde acelei intrări.

Exemplu: 3 intrări de adresă și $2^3=8$ ieșiri.

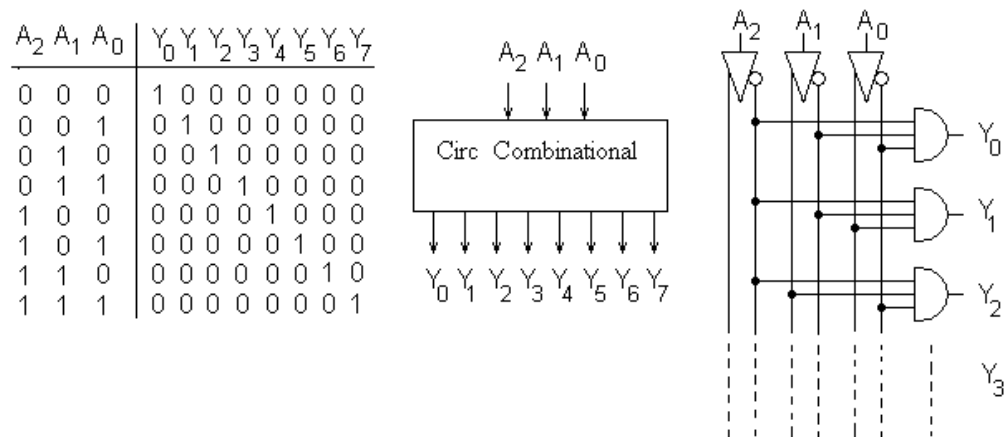


Fig.3. Decodificator de adresă de 3 biți

$$\begin{aligned}
 y_0 &= \bar{A}_2 \cdot \bar{A}_1 \cdot \bar{A}_0 & y_4 &= A_2 \cdot \bar{A}_1 \cdot \bar{A}_0 \\
 y_1 &= \bar{A}_2 \cdot \bar{A}_1 \cdot A_0 & y_5 &= A_2 \cdot \bar{A}_1 \cdot A_0 \\
 y_2 &= \bar{A}_2 \cdot A_1 \cdot \bar{A}_0 & y_6 &= A_2 \cdot A_1 \cdot \bar{A}_0 \\
 y_3 &= \bar{A}_2 \cdot A_1 \cdot A_0 & y_7 &= A_2 \cdot A_1 \cdot A_0
 \end{aligned}$$

Decodificator BCD - zecimal

Este identic cu cel prezentat anterior cu observația că la intrare se aplică un cod de 4 biți, numai până la combinația 9 (deci are A₃ A₂ A₁ A₀ - patru intrări de adresă). Evident tabelul de adevăr astfel descris este incomplet definit.

Variante - **cu rejecția** datelor false - pentru combinațiile de cod inexistente in BCD nu se activează nici o ieșire;

- **fără rejecția** datelor false – tabelul de adevăr se completează cu indiferent și apoi ecuațiile se minimizează.

Exemplu: Circuitul 5442 al cărui tabel de adevăr este prezentat mai jos.

D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

Observație: La ieșire nivelul activ este 0, deci ieșirile se notează cu bară.

Ecuatii :

$$\bar{0} = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$$

$$\bar{1} = A \cdot B \cdot C \cdot D, \text{ etc}$$

4. Circuite de multiplexare (MUX)

Multiplexorul este un circuit logic combinațional care permite trecerea datelor de la una din cele 2^n intrări spre o ieșire unică. Selecția intrării care este pusă în legătură cu ieșirea se face printr-un cuvânt de cod de selecție (adresă) de n biți.

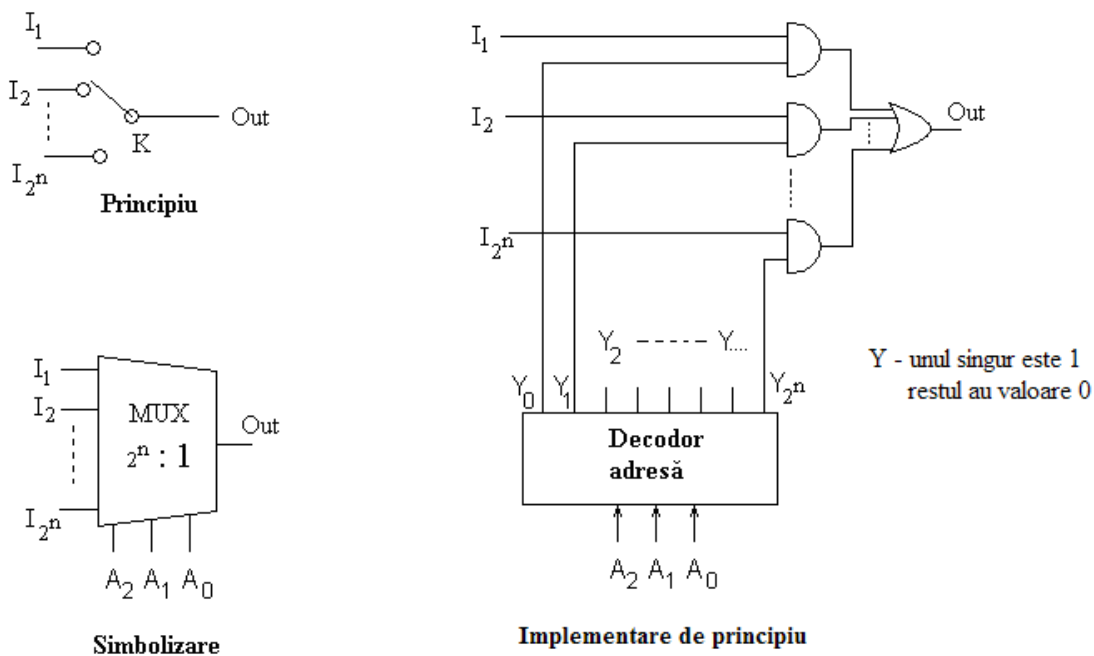


Fig.4. Multiplexor

Construcție : implementarea de principiu, fig.4., presupune utilizarea unui decodor de adresă și a unui ansamblu de circuite SI-SAU.

Funcționare: la aplicarea unui cod de adresă pe intrările A_i , decodorul de adresă furnizează la ieșirile Y un singur simbol 1 logic și restul 0; simbolul 1 logic validează un singur circuit SI; toate celelalte simboluri 0 blochează porțile SI; în acest fel o singură intrare I este pusă în legătură cu ieșirea Out .

Observație: Circuitul este unidirecțional în sensul că datele se aplică la intrări și se obțin la ieșire (în fond orice circuit logic este unidirecțional).

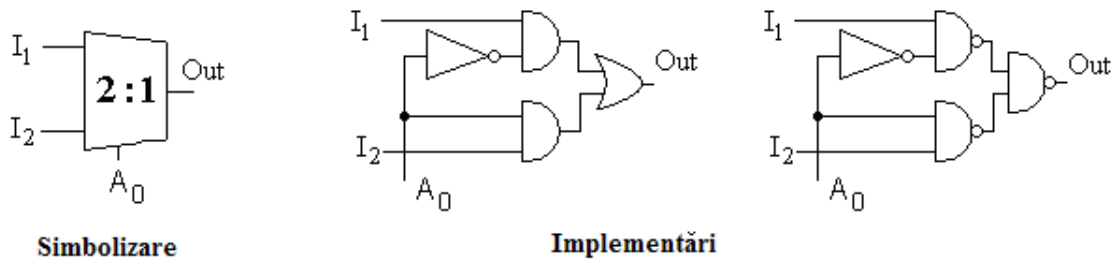
Exemple

Fig.5. MUX cu 2 intrări

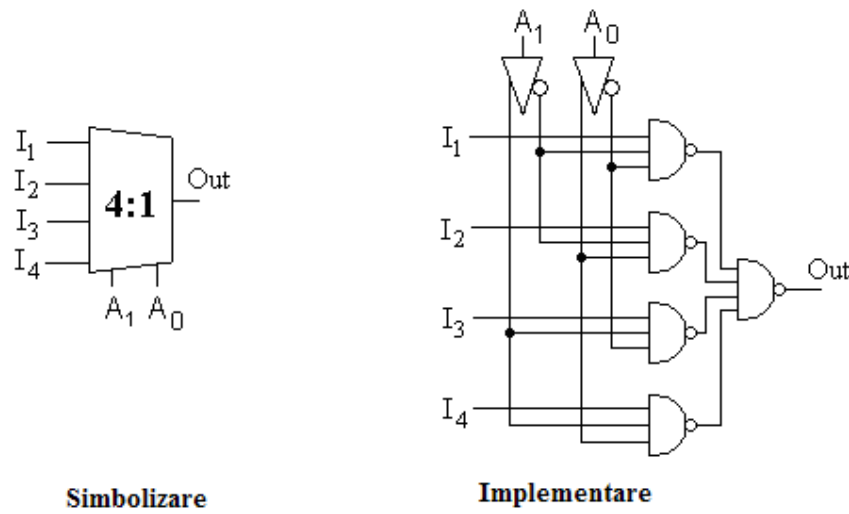


Fig.6. MUX cu 4 intrări

5. Circuit de demultiplexare (DEMUX)

Este un circuit logic combinațional care permite transmiterea datelor de pe o **singură** intrare pe **una** din ieșirile selectate. Selecția ieșirii se face prin intrări de adresă.

În fapt, așa cum arată și numele, circuitul realizează funcționarea inversă față de multiplexor.

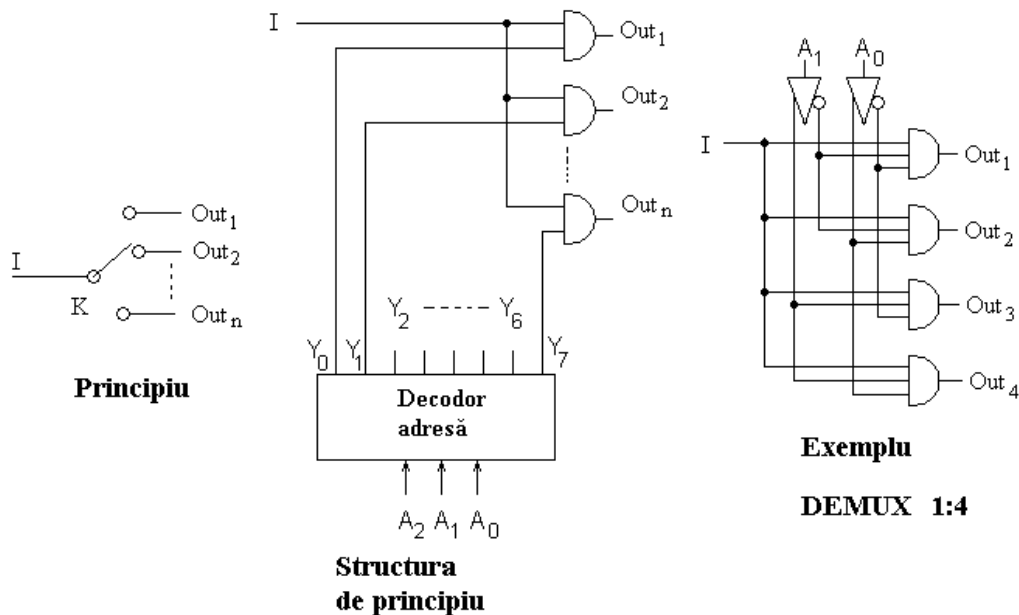


Fig.7. Demultiplexor

5.1. Circuit multiplexor/demultiplexor

În tehnologie CMOS, pe lângă porțile logice, există și porțile de transmisie care acționează similar unor comutatoare cu conducție bidirecțională. În consecință se poate construi circuitul din fig.8. care permite funcționarea fie ca multiplexor (dacă semnalul de intrare se aplică în partea stângă și ieșirea se face în dreapta) fie ca demultiplexor (intrarea în dreapta și ieșirea în stânga). Evident că, pentru o funcționare corectă a circuitului, trebuie avut grijă ca în orice moment **să fie deschis un singur comutator și restul blocate**. Acest lucru este asigurat de circuitul decodor de adresă.

În consecință circuitul va purta numele MUX/DEMUX.

Observație: poarta de transmisie nu numai că este bidirecțională, dar poate transmite și **semnale analogice** (acest lucru numai cu unele precauții privind polarizarea/alimentarea circuitului - există circuite special construite în acest scop)).

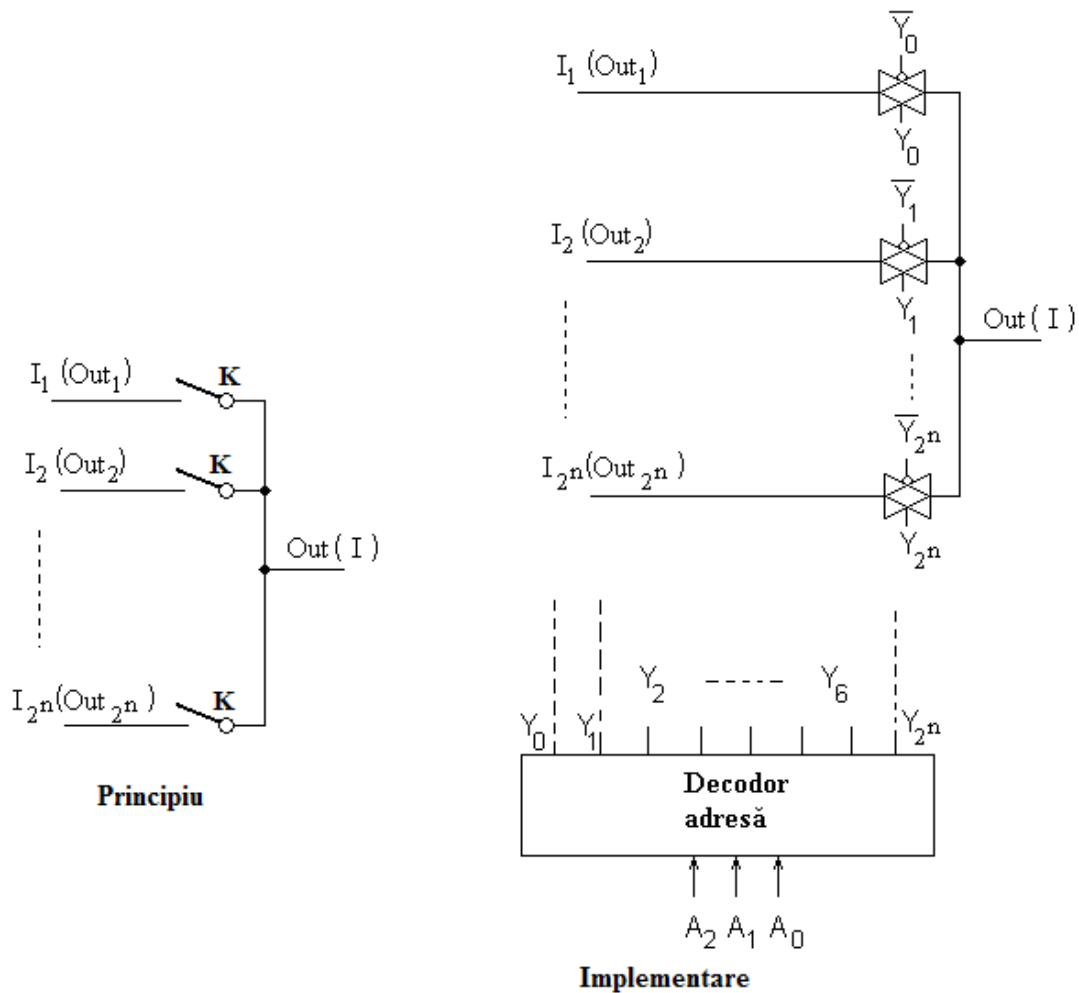


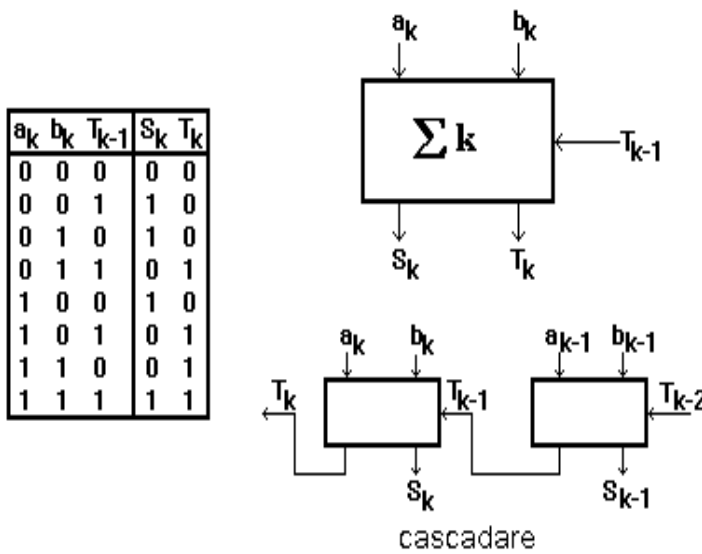
Fig.8. MUX/DEMUX în tehnologie CMOS

6. Circuit pentru adunarea a două numere

Un circuit sumator face adunarea a două numere binare și totodată poate să preia o valoare de „transport” rezultată de la o adunare precedentă; la ieșire circuitul furnizează o sumă-rezultat și un eventual „transport” către o unitate superioară. Prin legarea în cascadă a mai multor astfel de circuite se poate realiza un ansamblu care să suneze numere exprimate cu oricât de mulți biți.

Intrările circuitului sunt frecvent notate cu A, B – așa cum s-a procedat și în exemplul prezentat mai jos.

Observație: legarea în cascadă a circuitelor presupune o însumare a timpilor de propagare astfel încât rezultatul final este furnizat cu destul de mare întârziere față de semnalul de intrare. Din acest motiv o astfel de abordare mai poartă numele de „sumator secvențial”.



$a_k b_k$	00	01	11	10
T_{k-1}	0	1	0	1
0	0	1	0	1
1	1	0	1	0

$a_k b_k$	00	01	11	10
T_{k-1}	0	0	1	0
0	0	0	1	0
1	0	1	1	1

$T_k T_{k-1}$	00	01	11	10
$a_k b_k$	0	1	x	x
00	0	1	x	x
01	1	x	0	x
11	x	x	1	0
10	1	x	0	x

$$S_k = \bar{a}_k b_k \bar{T}_{k-1} + a_k \bar{b}_k \bar{T}_{k-1} + \bar{a}_k \bar{b}_k T_{k-1} + a_k b_k T_{k-1} = a_k \oplus b_k \oplus T_{k-1}$$

$$T_k = a_k b_k + T_{k-1} a_k + T_{k-1} b_k$$

Pentru S_k se mai poate folosi și o altă implementare $S_k = f(a_k, b_k, T_{k-1}, T_k)$:

$$S_k = a_k b_k T_{k-1} + \bar{T}_k T_{k-1} + \bar{T}_k b_k + \bar{T}_k a_k = a_k b_k T_{k-1} + \bar{T}_k (T_{k-1} + b_k + a_k)$$

Temă pentru pregătirea lucrării

1. Se descarcă Data Sheets pentru circuite TTL și CMOS care realizează funcțiile codificator de adresă, codificator de adresă prioritar, decodor de adresă, decodor BCD-zecimal, MUX, DEMUX, sumator.
2. Se analizează și comentează facilitățile pe care le permit aceste circuite.
3. Se notează dispunerea pinilor pentru circuitele analizate

Desfășurarea lucrării

1. Cu ajutorul montajului test se verifică funcționarea circuitelor SN54153, SN54151, SN5442 .