

1. ALGEBRA LOGICII

1.1. Variabile și funcții ale algebrei logicii

Pentru descrierea proceselor de lucru ale dispozitivelor electronice numerice, construite din diode, tranzistori, rezistori, condensatori, etc se folosesc variabile și funcții numerice (logice). Dispozitivele electronice pot să se afle în una din următoarele 2 stări:

- în regim de lucru (conectate la sursa de energie electrică);
- în regim de repaos (deconectate de la sursa de energie electrică).

Dispozitivele electronice numerice funcționează prin aplicarea la intrarea/intrările lor semnal electric dreptunghiular. Prezența sau lipsa semnalelor electrice la intrările dispozitivelor electronice, considerate în algebra logicii drept „variabile logice” pot fi notate prin:

- DA** sau **NU**;
- 1** sau **0**;
- orice literă din alfabetul latin fără indice:
 - este semnal (starea directă) – prin $\{a, b, c, \dots, z\}$;
 - semnalul electric lipsește (starea inversată) – prin $\{\bar{a}, \bar{b}, \bar{c}, \dots, \bar{z}\}$;
- orice literă din alfabetul latin cu indice:
 - este semnal (starea directă) – prin $\{a_{n-1}, a_{n-2}, a_{n-3}, \dots, a_0\}$;
 - semnalul electric lipsește (starea inversată) – prin $\{\bar{a}_{n-1}, \bar{a}_{n-2}, \bar{a}_{n-3}, \dots, \bar{a}_0\}$.

Definiție. Numim variabilă numerică (logică) o mărime care poate avea numai două valori (două stări).

Pentru descrierea procesului de lucru al dispozitivelor numerice se folosesc funcții numerice (logice). Funcțiile logice pot prezenta în modul următor:

- $F(a, b, c, \dots, z)$;
- $F(a_{n-1}, a_{n-2}, a_{n-3}, \dots, a_0)$

Definiție. Numim funcție numerică (logică) o mărime care ca și variabilele sale poate avea numai două valori (două stări).

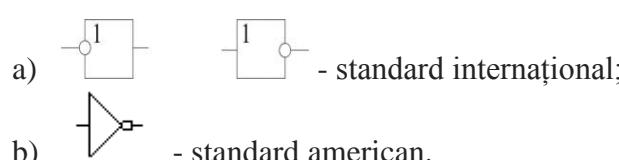
Pentru construirea dispozitivelor electronice numerice de orice coplexitate pot fi folosite microcircuite principiul de lucru al căroro sănt descrise numai de trei funcții logice elementare: **NU**, **ȘI**, **SAU**.

Definiție. Funcția NU este o funcție logică cu o variabilă (cu un singur argument) și este egală cu valoarea inversată a variabilei (argumentului).

Funcția NU se mai numește negare, iar dispozitivul care execută această operație se numește invertor. În formă algebraică funcția NU poate fi prezentată în modul următor

$$F(a) = \bar{a}.$$

Simbolul convențional și exemplu de funcționare al elementului NU (invertorului) sunt prezente în figura 1.1.



În fig. 1.1 sunt următoarele notări: +V_{CC} – sursa de tensiune/semnal; [A] – comutator.

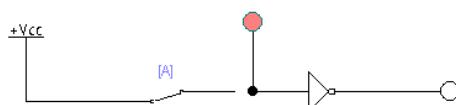


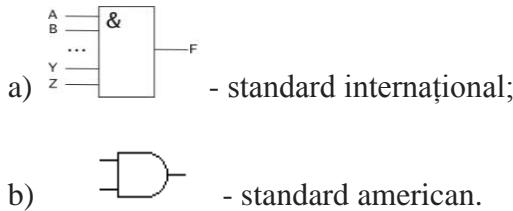
Fig. 1.1. Simbolul convențional al elementului NU (invertorului) și exemplu de funcționare.

Definiție. Funcția řI este o funcție logică cu două și mai multe variabile și este egală cu 1 numai atunci cînd toate variabilele ei sunt egale cu 1.

Această funcție se mai numește produs logic sau conjuncție. În formă algebrică funcția řI poate fi prezentată în modul următor

$$F(a, b, c, \dots, z) = a \times b \times c \times \dots \times z = a \wedge b \wedge c \wedge \dots \wedge z.$$

Simbolul convențional al elementului řI și exemplu de funcționare sunt prezentate în figura 1.2.



În fig. 1.2 sunt următoarele notări: $+V_{cc}$ – sursa de tensiune/semnal; [A], [B] – comutatoare.

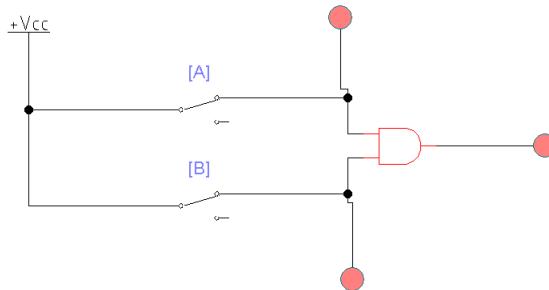


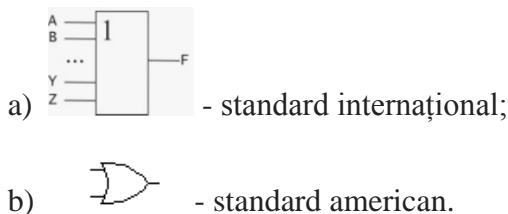
Fig. 1.2. Simbolul convențional și exemplu de funcționare al elementului řI.

Definiție. Funcția SAU este o funcție logică cu două și mai multe variabile și este egală cu 1 atunci, cînd măcar o variabilă a ei este egală cu 1.

Funcția SAU se mai numește sumă logică sau disjuncție. În formă algebrică funcția SAU poate fi prezentată în modul următor

$$F(a, b, c, \dots, z) = a + b + c + \dots + z = a \vee b \vee c \vee \dots \vee z.$$

Simbolul convențional și exemplu de funcționare al elementului SAU sunt prezentate în figura 1.3.



În fig. 1.3 sunt următoarele notări: $+V_{cc}$ – sursa de tensiune/semnal; [A], [B] – comutatoare.

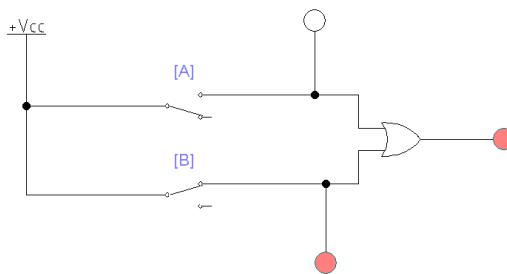


Fig. 1.3. Simbolul convențional și exemplu de funcționare al elementului SAU.

Stările funcțiilor logice pot fi prezentate și prin tabele numite tăbele de adevăr sau tăbele de stări.

Definiție. Numim tabel de adevăr un tabel în care sunt prezentate toate combinațiile posibile ale variabilelor și valorile funcției (funcțiilor).

În tabelul 1.1 sunt prezentate stările funcțiilor cu două variabile.

Tabelul 1.1. Stările funcțiilor cu două variabile

Variabile		Funcții logice					
		ȘI	SAU	ȘI-NU	SAU-NU	SAU-EX	Echivalență
a	b						
0	0	0	0	1	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	1	0	0	0	1

Numărul de funcții cu 2 variabile este $N = 2^{2^2}$, cu 3 variabile este $N = 2^{2^3}$, cu m variabile este $N = 2^{2^m}$.

1.2. Axiomele și teoremele principale ale algebrei logicii

Axiomele principale pentru funcțiile NU, ȘI, SAU sînt următoarele:

a) pentru funcția NU: $a = \bar{a}$.

b) pentru funcția ȘI: $a \times 0 = 0$ sau $a \wedge 0 = 0$;

$$a \times 1 = a \text{ sau } a \wedge 1 = a;$$

$$a \times \bar{a} = 0 \text{ sau } a \wedge \bar{a} = 0;$$

$$a \times a = a \text{ sau } a \wedge a = a;$$

$$a \times a \times \dots \times a = a \text{ sau } a \wedge a \wedge \dots \wedge a = a.$$

c) pentru funcția SAU: $a + 0 = a$ sau $a \vee 0 = a$;

$$a + 1 = 1 \text{ sau } a \vee 1 = 1;$$

$$a + \bar{a} = 1 \text{ sau } a \vee \bar{a} = 1;$$

$$a + a = a \text{ sau } a \vee a = a;$$

$$a + a + \dots + a = a \text{ sau } a \vee a \vee \dots \vee a = a.$$

Teoremele logice principale utilizate pentru transformarea funcțiilor logice și descrierea procelor de lucru ale dispozitivelor electrice numerice sînt: teorema comutativă; teorema asociativă; teorema distributivă; teorema asimilării; teorema de Morgan.

Teorema comutativă. Schimbarea poziției variabilor într-o relație logică nu aduce la schimbarea rezultatului funcției.

Exemplu:

$$a + b = b + a \text{ sau } a \vee b = b \vee a ;$$

$$a \times b = b \times a \text{ sau } a \wedge b = b \wedge a .$$

În fig. 1.4 sunt prezentate două scheme electrice echivalente, construite din elementul logic SAU, care demonstrează posibilitatea aplicării teoremei comutative pentru scheme electrice logice.

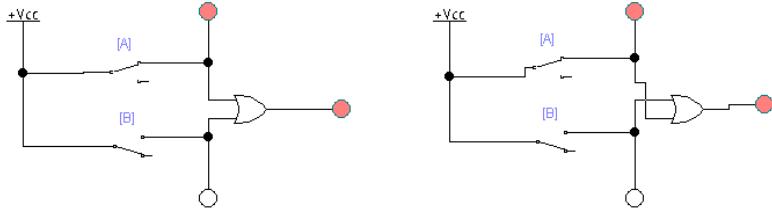


Fig. 1.4. Scheme echivalente pentru demonstrarea teoremei comutative.

Teorema asociativă. Asocierea variabilelor într-o relație logică nu aduce la schimbarea funcției.

Exemplu:

$$a + b + c = a + (b + c) \text{ sau } a \vee b \vee c = a \vee (b \vee c);$$

$$a \times b \times c = (a \times b) \times c \text{ sau } a \wedge b \wedge c = (a \wedge b) \wedge c .$$

În fig. 1.5 sunt prezentate două scheme electrice echivalente care demonstrează posibilitatea aplicării teoremei asociative pentru scheme electrice logice.

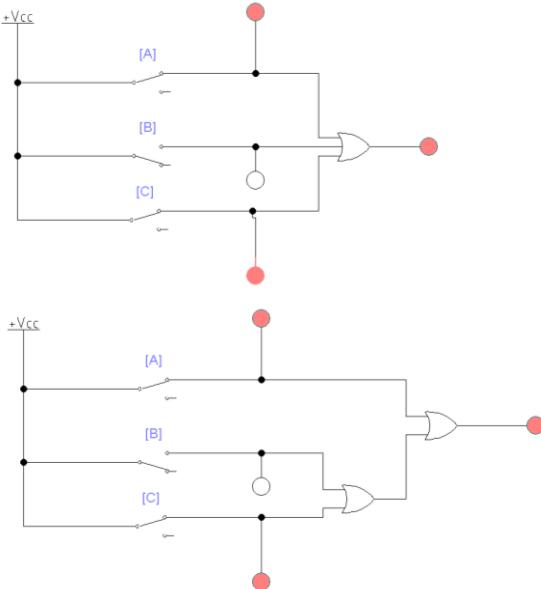


Fig. 1.5. Scheme echivalente pentru demonstrarea teoremei asociative.

Teorema distributivă. Rezultatul operației nu se va schimba dacă variabilele comune sunt scoase în afara parantezelor.

Exemplu:

$$a \times b + a \times c = a \times (b + c) \text{ sau } a \wedge b \vee a \wedge c = a \wedge (b \vee c);$$

Această teoremă permite micșorarea numărului de microcircuite într-un dispozitiv numeric. În fig. 1.6 sunt prezentate două scheme electrice echivalente care demonstrează posibilitatea aplicării teoremei distributive pentru scheme electrice logice.

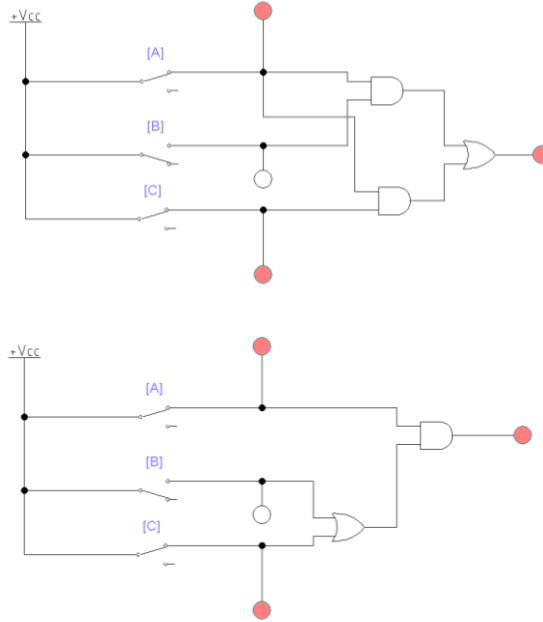


Fig. 1.6. Scheme echivalente pentru demonstrarea teoremei distributive.

Teorema asimilării permite micșorarea numărului de variabile.

Exemplu:

$$a + a \times c = a \times (1 + c) = a.$$

În fig. 1.7 sunt prezentate două scheme electrice echivalente care demonstrează posibilitatea aplicării teoremei asimilării pentru scheme electrice logice.

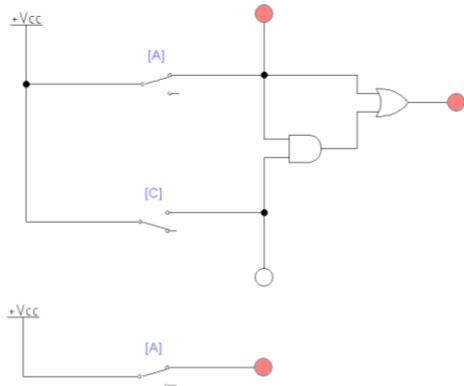


Fig. 1.7. Scheme echivalente pentru demonstrarea teoremei asimilării.

Această teoremă permite micșorarea numărului de microcircuite într-un dispozitiv numeric.

Teorema de Morgan.

a) **Suma inversată a variabilelor este egală cu produsul variabilelor inverse**

$$\overline{a + b + \dots + z} = \bar{a} \times \bar{b} \times \dots \times \bar{z};$$

b) **Produsul inversat al variabilelor este egal cu suma variabilelor inverse.**

$$\overline{a \times b \times \dots \times z} = \bar{a} + \bar{b} + \dots + \bar{z};$$

Demonstrarea teoremei este prezentată cu ajutorul tabelului de adevăr (vezi tabelul 1.2).

Tabelul 1.2. Demonstrarea teoremei de Morgan.

Nr. d/o	a	b	\bar{a}	\bar{b}	$\overline{a+b}$	$\overline{\bar{a} \times \bar{b}}$	$\overline{a \times b}$	$\overline{\bar{a} + \bar{b}}$
0	0	0	1	1	1	1	1	1
1	0	1	1	0	0	0	1	1
2	1	0	0	1	0	0	1	1
3	1	1	0	0	0	0	0	0

Din tabelul 1.2 obținem: $\overline{a+b} = \bar{a} \times \bar{b}$ și $\overline{a \times b} = \bar{a} + \bar{b}$.

În fig. 1.8 sunt prezentate două scheme electrice echivalente care demonstrează posibilitatea aplicării teoremei de Morgan pentru scheme electrice logice.

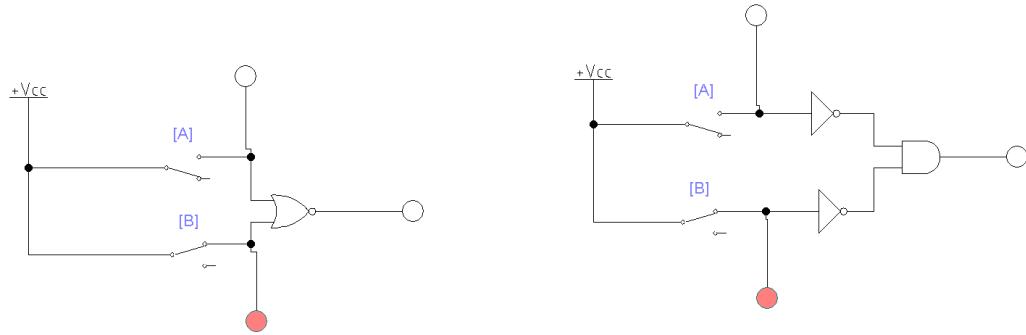


Fig. 1.8. Scheme echivalente pentru demonstrarea teoremei de Morgan.

Lucru independent nr. 1. De simplificat funcțiile și de construit schemele electrice pînă la simplificarea funcției și după simplificarea ei:

- a) $a + ab + a\bar{b} =$
- b) $a\bar{b} + b =$
- c) $(a + \bar{b})b =$

1.3. Familii de funcții logice fundamentale echivalente

Funcțiile NU, ȘI, SAU reprezintă familia fundamentală principală de funcții logice. Microcircuitele care execută aceste funcții se folosesc pentru construirea dispozitivelor numerice. Totuși, în procesul construirii dispozitivelor numerice, nu este rentabil de a folosi 3 microcircuite diferite. În majoritatea cazurilor, în procesul construirii dispozitivelor numerice, în loc de aceste 3 microcircuite NU, ȘI, SAU se folosesc numai microcircuite care execută funcția ȘI-NU sau microcircuite care execută funcția SAU-NU și altele. În prezent sunt utilizate cîteva zeci de familii de funcții logice fundamentale echivalente.

Definiție. Funcțiile logice care pot înlocui funcțiile NU, ȘI, SAU reprezintă familii de funcții logice fundamentale echivalente.

Pentru a demonstra că funcțiile ȘI-NU, SAU-NU tot prezintă familii de funcții logice fundamentale v-oam construi schemele electrice din ȘI-NU, SAU-NU care înlocuesc execută funcțiile NU, ȘI, SAU. Folosim microcircuite ȘI-NU, SAU-NU numai cu 2 intrări.

Prezentarea NU prin ȘI-NU.

Pentru schema din NU $F(a) = \bar{a}$, iar pentru schema din ȘI-NU cu 2 intrări $F(a) = \overline{\bar{a} \times \bar{a}} = \bar{a}$.

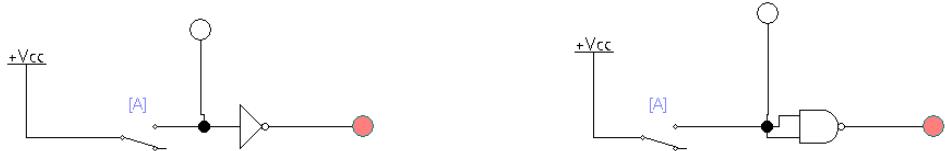


Fig. 1.9. Scheme echivalente prezentate prin NU și ȘI-NU.

Prezentarea ȘI prin ȘI-NU.

Pentru schema din ȘI $F(a, b) = a \times b$. Folosim axioma (2 inversări nu schimbă rezultatul) pentru a prezenta formula numai prin ȘI-NU $F(a, b) = a \times b = \overline{\bar{a} \times \bar{b}}$.

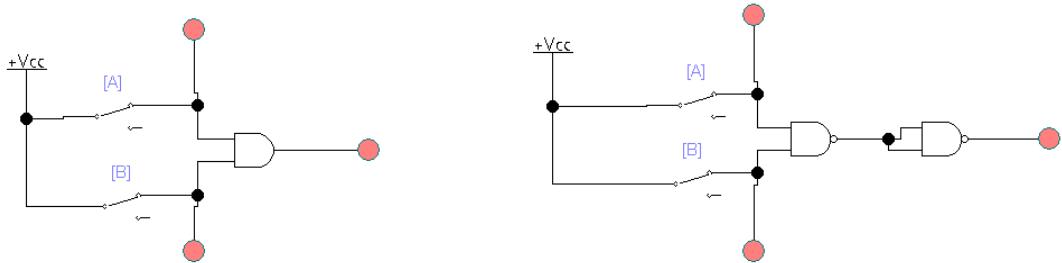


Fig. 1.10. Scheme echivalente prezentate prin ȘI și ȘI-NU.

Prezentarea SAU prin ȘI-NU.

Pentru schema din SAU $F(a, b) = a + b$. Folosim teorema de Morgan pentru a prezenta formula numai prin ȘI-NU $F(a, b) = a + b = \overline{\bar{a} + \bar{b}} = \bar{a} \times \bar{b}$.

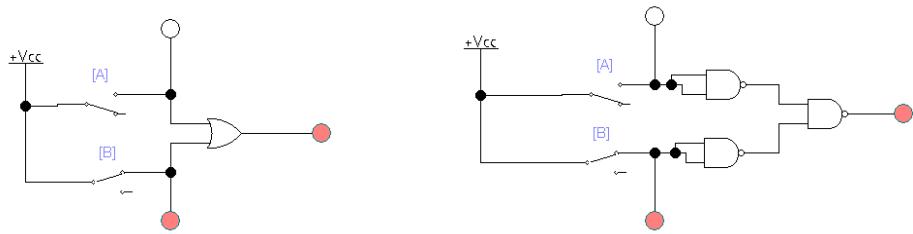


Fig. 1.11. Scheme echivalente prezentate prin SAU și ȘI-NU.

Prezentarea NU prin SAU-NU.

Pentru NU $F(a) = \bar{a}$, iar pentru SAU-NU cu 2 intrări $F(a) = \overline{\bar{a} + a} = \bar{a}$.

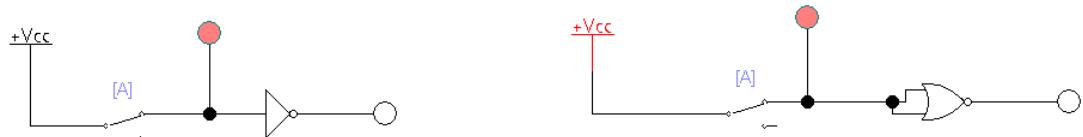


Fig. 1.12. Scheme echivalente prezentate prin NU și SAU-NU.

Prezentarea SAU prin SAU-NU.

Pentru schema din SAU $F(a, b) = a + b$. Folosim axioma (2 inversări nu schimbă rezultatul) pentru a prezenta formula numai prin SAU-NU $F(a, b) = a + b = \overline{\bar{a} + \bar{b}}$.

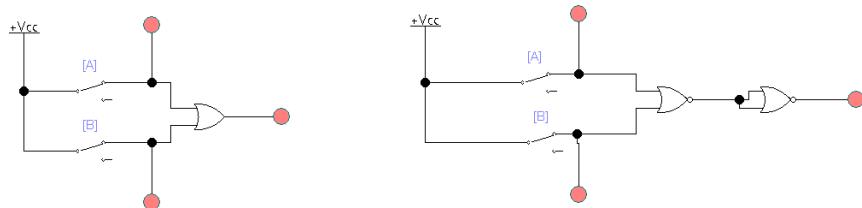


Fig. 1.13. Scheme echivalente prezentate prin SAU și SAU-NU.

Prezentarea ȘI prin SAU-NU.

Pentru schema din ȘI $F(a, b) = a \times b$. Folosim teorema de Morgan pentru a prezenta formula numai prin SAU-NU $F(a, b) = a \times b = \overline{\bar{a} \times \bar{b}} = \bar{a} + \bar{b}$.

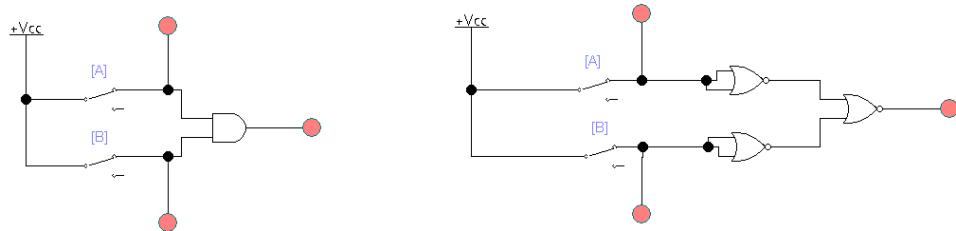


Fig. 1.14. Scheme echivalente prezentate prin ȘI și SAU-NU.

În multe scheme înlocuirea NU, řI, SAU prin řI-NU, SAU-NU nu duce la majorarea numărului de microcircuite.

Exemplu. De construit schema electrică logică numai din řI-NU pentru funcția $F(a, b, c, d) = a \times b + c \times d$.

Folosim teorema de Morgan pentru a prezenta funcția $F(a, b)$ numai prin řI-NU

$$F(a, b, c, d) = a \times b + c \times d = \overline{\overline{a} \times b + c \times d} = \overline{\overline{a} \times b} \times \overline{c \times d}.$$

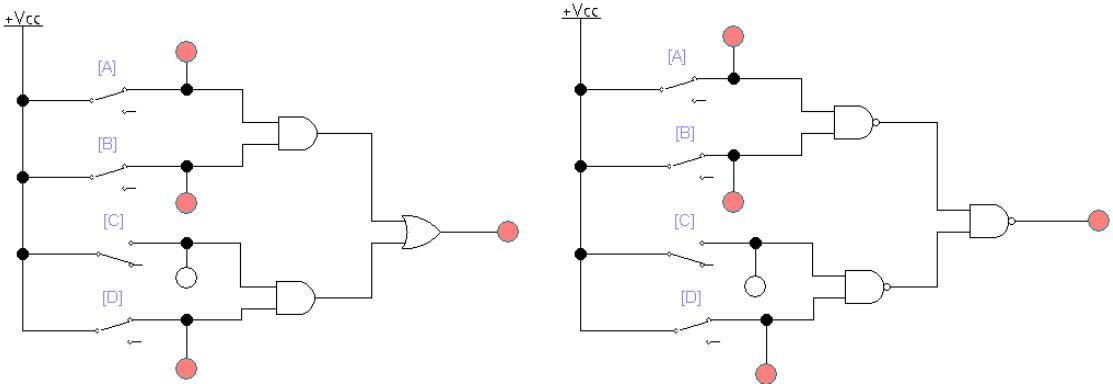


Fig. 1.15. Scheme echivalente prezentate pentru funcția $F(a, b, c, d) = a \times b + c \times d$.

Lucru independent nr. 2.

- a) De scris relațiile algebrice și de construit scemele electrice pentru funcțiile logice SAU-EX și Echivalența numai din elemente řI-NU, SAU-NU (4 scheme). Folosiți tabelul 1.1.
- b) De transformat funcția logică

$$F(a, b, c) = \bar{a} \times b \times \bar{c} + \bar{a} \times b \times c + a \times \bar{b} \times c + a \times b \times \bar{c} + a \times b \times c$$
și de construit scema electrică numai de elemente řI-NU.
- c) De transformat funcția logică

$$F(a, b, c) = \bar{a} \times b \times \bar{c} + \bar{a} \times b \times c + a \times \bar{b} \times c + a \times b \times \bar{c} + a \times b \times c$$
și de construit scema electrică numai de elemente SAU-NU.

1.4. Metodele de prezentare a funcțiilor logice

Funcțiile logice pot fi prezentate prin următoarele metode:

1. metoda scriptică (o descriere prin cuvinte a funcției/functiilor îndeplinite de dispozitivul numeric);
2. cu ajutorul tabelelor de adevăr (al stărilor);
3. metoda analitică (algebrică):
 - a) forma complet normală dizjunctivă (FCND);
 - b) forma complet normală conjunctivă (FCNC);
4. în formă numerică;
5. cu ajutorul schemelor electrice principale.

Metoda scriptică este o descriere prin cuvinte a funcției/functiilor îndeplinite de dispozitivul numeric. Este elaborată la prima etapă de proiectare a unui dispozitiv numeric și se mai numește „Sarcină tehnică”.

Exemplu. De construit un dispozitiv numeric care va monitoriza trei parametri într-un depozit: umiditatea, temperatura și iluminarea. Dispozitivul semnalează dacă parții monitorizați nu corespund standardului.

Metoda de prezentare a funcție logice cu ajutorul tabelelor de adevăr (al stărilor).
Tabelele de adevăr se completează în baza sarcinei tehnice.

Definiție. Numim tabel de adevăr un tabel în care sunt prezentate toate combinațiile posibile ale variabilelor și valorile funcției (funcțiilor).

Conform exemplului de mai sus notăm: umeditatea – variabila **a** (**a=0** dacă Umeditatea \leq 60% și **a=1** dacă Umeditatea $>$ 60%); temperatura – variabila **b** (**b=0** dacă $T \leq 6$ C și **b=1** dacă $T > 6$ C); iluminarea – variabila **c** (**c=0** dacă în depozit este lumină și **c=1** dacă în depozit este întuneric). În baza acestor date se completează tabelul de adevăr (vezi tabelul 1.3), funcția logică primește valoarea 1 dacă se schimbă temperatura sau oricare 2 parametri.

Tabelul 1.3. Stările posibile în depozit

Nr. d/o	a	b	c	F(a,b,c)
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Forma analitică (algebrică) permite prezentarea funcțiilor logice prin forma complet normală dizjunctivă (FCND) și forma complet normală conjunctivă (FCNC). FCND și FCNC se obțin din tabelul de adevăr.

Prezentarea FCND.

Definiție. Numim FCND suma logică a mintermenilor.

Definiție. Numim mintermen produsul logic al variabilelor luate cîte una în stare directă sau inversă.

Tabelul 1.4. Stările posibile în depozit cu indicarea mintermenilor

Nr. d/o	a	b	c	$F(a,b,c)$, f_i , unde $i = 0, 1, \dots, 7$	Mintermeni, m_i , unde $i = 0, 1, \dots, 7$
0	0	0	0	0	$m_0 = \bar{a} \times \bar{b} \times \bar{c}$
1	0	0	1	0	$m_1 = \bar{a} \times \bar{b} \times c$
2	0	1	0	1	$m_2 = \bar{a} \times b \times \bar{c}$
3	0	1	1	1	$m_3 = \bar{a} \times b \times c$
4	1	0	0	0	$m_4 = a \times \bar{b} \times \bar{c}$
5	1	0	1	1	$m_5 = a \times \bar{b} \times c$
6	1	1	0	1	$m_6 = a \times b \times \bar{c}$
7	1	1	1	1	$m_7 = a \times b \times c$

Din tabelul 1.4 obținem formula generală,

$$F(a,b,c) = f_0 \times m_0 + f_1 \times m_1 + f_2 \times m_2 + f_3 \times m_3 + f_4 \times m_4 + f_5 \times m_5 + f_6 \times m_6 + f_7 \times m_7.$$

Deoarece $f_0 = f_1 = f_4 = 0$ obținem

$$F(a,b,c) = f_2 \times m_2 + f_3 \times m_3 + f_5 \times m_5 + f_6 \times m_6 + f_7 \times m_7.$$

Funcția poate fi prezentată și în formă desfășurată (FCND)

$$F(a,b,c) = \bar{a} \times b \times \bar{c} + \bar{a} \times b \times c + a \times \bar{b} \times c + a \times b \times \bar{c} + a \times b \times c. \quad (1)$$

Prezentarea FCNC.

Definiție. Numim FCNC produsul logic al maxtermenilor.

Definiție. Numim maxtermen suma logică a variabilelor luate câte una în stare directă sau inversă.

Tabelul 1.4. Stările posibile în depozit cu indicarea mintermenilor

Nr. d/o	a	b	c	$F(a,b,c),$ $f_i, \text{ unde } i = 0, 1, \dots, 7$	Maxtermeni, $M_i, \text{ unde } i = 0, 1, \dots, 7$
0	0	0	0	0	$M_0 = a + b + c$
1	0	0	1	0	$M_1 = a + b + \bar{c}$
2	0	1	0	1	$M_2 = \bar{a} + b + \bar{c}$
3	0	1	1	1	$M_3 = a + \bar{b} + \bar{c}$
4	1	0	0	0	$M_4 = \bar{a} + b + c$
5	1	0	1	1	$M_5 = \bar{a} + b + \bar{c}$
6	1	1	0	1	$M_6 = \bar{a} + \bar{b} + c$
7	1	1	1	1	$M_7 = \bar{a} + \bar{b} + \bar{c}$

Din tabelul 1.4 alegem maxtermenii din rîndurile unde starea $f_i = 0$ și obținem

$$F(a,b,c) = M_0 \times M_1 \times M_4.$$

Funcția poate fi prezentată și în formă desfășurată (FCNC)

$$F(a,b,c) = (a + b + c) \times (a + b + \bar{c}) \times (\bar{a} + b + c). \quad (2)$$

Formulele (1) și (2) se utilizează pentru construirea schemelor electrice

Forma numerică de prezentare a funcțiilor logice este utilizată pentru a micșora procedura de scriere.

Drept exemplu:

a) pentru FCND

$F(a,b,c) = \bar{a} \times b \times \bar{c} + \bar{a} \times b \times c + a \times \bar{b} \times c + a \times b \times \bar{c} + a \times b \times c = \Sigma(2,3,5,6,7)$, unde, în paranteze sunt indicate numerele de ordine ale mintermenilor pentru care $f_i=1$.

b) pentru FCNC

$F(a,b,c) = (a + b + c) \times (a + b + \bar{c}) \times (\bar{a} + b + c) = \Pi(0,1,4)$ unde, în paranteze sunt indicate numerele de ordine ale maxtermenilor pentru care $f_i=0$.

Folosind teorema de Morgan din FCND se obține FCNC și invers.

Prezentarea funcțiilor logice cu ajutorul schemelor electrice principale. Schemele electrice principale se construiesc utilizînd FCND sau FCNC.

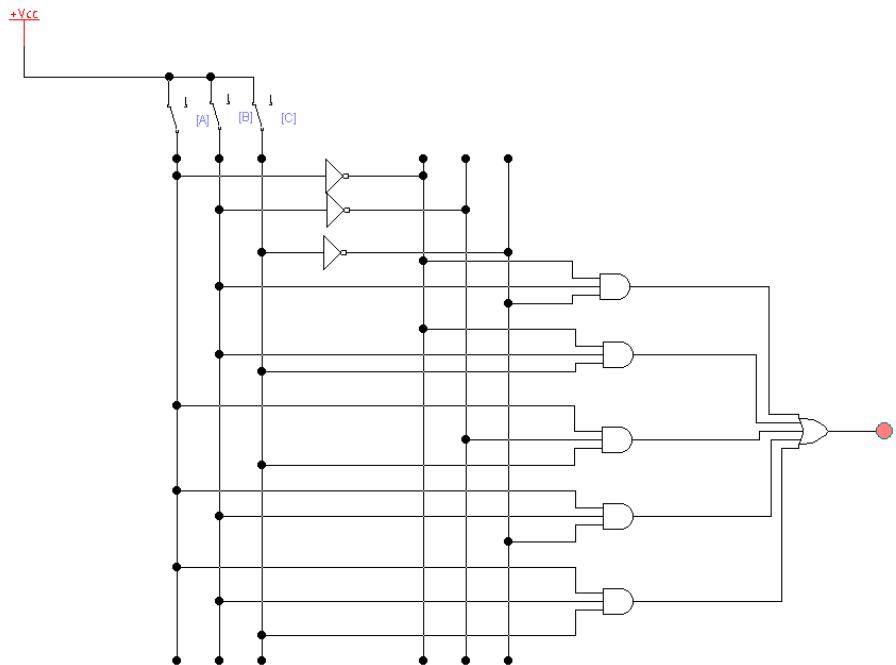


Fig. 1.15. Schema electrică construită conform FCND (formula (1)).

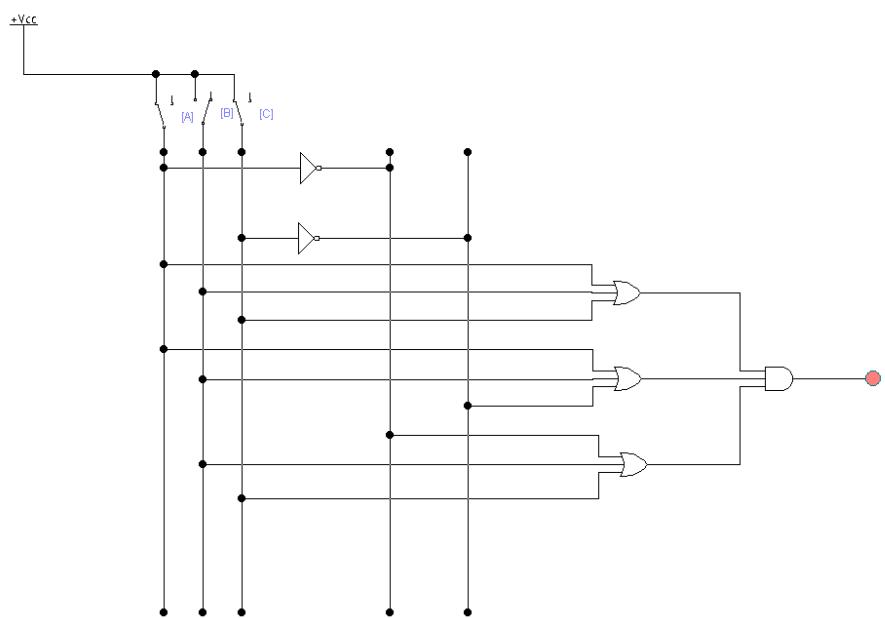


Fig. 1.16. Schema electrică construită conform FCNC (formula (2)).

Schemele din fig.1.15 și fig. 1.16 sunt echivalente deoarece execută aceeași funcție.

Lucru independent nr. 3.

1. Transformați funcția logică din FCNC în FCND și prezentați funcția obținută prin metoda numerică (Преобразовать логическую функцию из СКНФ в СДНФ и представить полученную функцию в цифровом виде)

$$f(a,b,c,d) = (a + b + c + d)(a + b + c + \bar{d})(a + \bar{b} + c + \bar{d})(a + \bar{b} + \bar{c} + \bar{d})(\bar{a} + b + c + d)$$

2. Scrieți FCND ale funcțiilor logice care corespund următorului tabel de adevăr (Напишите СДНФ логических функций, которые соответствуют следующей таблице истинности)

Nr. d/o	Argumenti				Funcția	
	a	b	c	d	F ₁ (a,b,c,d)	F ₂ (a,b,c,d)
0	0	0	0	0	0	1
1	0	0	0	1	1	0
2	0	0	1	0	0	0
3	0	0	1	1	1	0
4	0	1	0	0	0	0
5	0	1	0	1	1	1
6	0	1	1	0	1	1
7	0	1	1	1	0	1
8	1	0	0	0	0	0
9	1	0	0	1	1	1
10	1	0	1	0	0	0
11	1	0	1	1	1	1
12	1	1	0	0	1	0
13	1	1	0	1	0	0
14	1	1	1	0	0	1
15	1	1	1	1	1	0

3. Transformați funcția logică din FCND în FCNC și prezentați funcția obținută prin metoda numerică (Преобразовать логическую функцию из СДНФ в СКНФ и представить полученную функцию в цифровом виде)

$$f(a,b,c,d) = a\bar{b}\bar{c}\bar{d} + \bar{a}bcd + a\bar{b}cd + \bar{a}\bar{b}cd + abcd$$

1.5. Minimizarea funcțiilor logice

Elaborarea dispozitivelor numerice în baza funcțiilor prezentate în formatele FCND și FCNC în majoritatea cazurilor nu este justificată deoarece se obțin scheme electrice cu surplus de elemente logice. Pentru a micșora numărul de elemente logice în schema electrică se execută procedura de minimizare/simplificare a funcției logice. Minimizarea funcției logice este o procedură obligatorie a procesului de proiectare al dispozitivelor numerice. Dispozitivele numerice construite în baza funcțiilor logice minimeze spre deosebire de dispozitivele numerice construite în baza funcțiilor logice inițiale neminimizate au următoarele avantaje:

- conțin mai puține elemente logice;
- consumă mai puțină energie electrică;
- viteza de lucru a dispozitivului este mai mare;
- masa, gabaritele dispozitivului sunt mai mici;
- prețul dispozitivului devine mai mic.

Definiție. Minimizarea funcției logice reprezintă o procedură de simplificare cu scopul de a obține forma cea mai simplă a funcției logice.

În prezent sunt cunoscute mai multe metode de minimizare ale funcțiilor logice:

- metoda algebraică;
- metoda tabelelor Veitch-Karnaugh;
- metoda cuburilor;
- metoda Quin-McCluskey;
- metoda coeficientilor nedeterminați și altele.

Pentru majoritatea metodelor de minimizare ale funcțiilor logice sunt elaborate programe specializate.

În continuare v-or fi prezentate metoda algebraică și metoda tabelelor Veitch-Karnaugh de minimizare ale funcțiilor logice.

Metoda algebraică de minimizare a funcțiilor logice utilizează axiomele și teoremele algebrei logicii și poate fi folosită pentru minimizarea funcțiilor logice cu un număr de variabile relativ mic.

Exemplu. De minimizat funcția logică, utilizând metoda algebraică

$$F(a,b,c) = \bar{a} \times b \times \bar{c} + \bar{a} \times b \times c + a \times \bar{b} \times c + a \times \bar{b} \times \bar{c} = \bar{a} \times b \times (\bar{c} + c) + a \times \bar{b} \times (c + \bar{c}) = \bar{a} \times b + a \times \bar{b} = \bar{a}b + a\bar{b}$$

În fig. 1.18 sunt prezentate schemele electrice în baza funcției logice pînă și după minimizare.

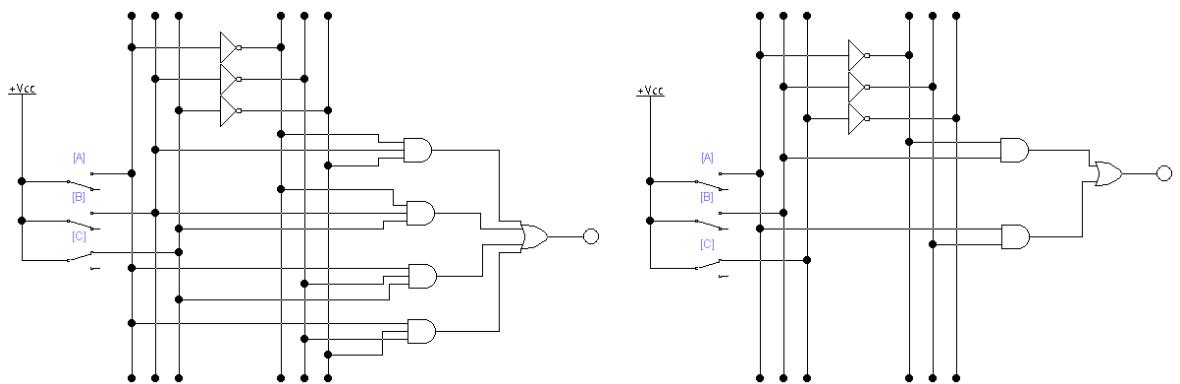


Fig. 1.18. Schemele electrice construite în baza funcției logice pînă și după minimizare.

Metoda tabelelor Veitch-Karnaugh utilizează pentru minimizarea funcțiilor logice tebele de mintermeni. Numărul de celule în tabelele de mintermeni depinde de numărul de variabile (ordinul) funcției logice supuse minimizării și este determinat conform formulei $N_{cel} = 2^m$, unde m reprezintă numărul de variabile. Mintermenii prezenți în formula pentru funcția logică sunt introdusi în tabelul de mintermeni ca 1.

1. Tabelul de mintermeni pentru o funcție logică cu o singură variabilă, $F(a) = \bar{a}$, are 2 celule (vezi fig. 1.19).

a	0	1
	1	

Fig. 1.19. Tabelul de mintermeni pentru o funcție logică cu o singură variabilă.

2. Tabelul de mintermeni pentru o funcție logică cu 2 variabile, $F(a, b) = \bar{a}\bar{b} + a\bar{b}$, are 4 celule (vezi fig. 1.20).

$b \backslash a$	0	1
0		1
1	1	

Fig. 1.20. Tabelul de mintermeni pentru o funcție logică cu 2 variabile.

3. Tabelul de mintermeni pentru o funcție logică cu 3 variabile, $F(a, b, c) = \bar{a}\bar{b}\bar{c} + a\bar{b}\bar{c}$, are 8 celule (vezi fig. 1.21). Tabelul de mintermeni poate fi prezentat în poziție orizontală sau în poziție verticală.

$c \backslash ab$	00	01	11	10
0				1
1		1		

Fig. 1.21. Tabelul de mintermeni pentru o funcție logică cu 3 variabile.

1. Tabelul de mintermeni pentru o funcție logică cu 4 variabile are 16 celule. Drept exemplu se propune pentru minimizare următoarea funcție logică

$$F(a, b, c, d) = \Sigma(0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 12, 13, 14). \quad (3)$$

Conform formulei (3) schema electrică conține 3 elemente NU, 13 elemente SI și un element SAU.

Introducem mintermenii în tabel (vezi fig. 1.22).

$cd \backslash ab$	00	01	11	10	
00	1	1	1	1	1 1 0 0
01	1		1	1	1 0 0 0
11	1	1			1 0 0 1
10	1	1	1	1	1 0 1 0

The diagram shows arrows originating from the minterm values in the table and pointing to their corresponding binary representations below. The first row (00, 01, 11, 10) points to 1100. The second row (00, 01, 11, 10) points to 1000. The third row (00, 01, 11, 10) points to 1001. The fourth row (00, 01, 11, 10) points to 1010. The fifth row (00, 01, 11, 10) points to 0100. The sixth row (00, 01, 11, 10) points to 0101. The seventh row (00, 01, 11, 10) points to 0010. The eighth row (00, 01, 11, 10) points to 0000.

Fig. 1.22. Tabelul de mintermeni pentru o funcție logică cu 4 variabile.

Proprietăile tabelului de mintermeni depind de scrierea corectă a consecutivității valorilor variabilelor poziționate orizontal și vertical: a) orizontal – ab (00, 01, 11, 10); vertical – cd (00, 01, 11, 10). Dacă consecutivitățile valorilor variabilelor poziționate orizontal sau/și vertical sunt scrise altfel, atunci acest tabel nu poate fi utilizat pentru minimizarea funcțiilor logice.

Conform tabelului de mintermeni pentru o funcție logică cu 4 variabile pot fi evidențiate următoarele proprietăți (proprietățile prezentate mai jos sunt valabile pentru tabelele de mintermeni/funcții logice cu orice număr de variabile):

1. mintermenii vecini, poziționați orizontal sau vertical, se deosebesc cu o singură stare;

2. mintermenii, poziționați în același rînd sau aceeași coloană la frontieră tabelului, se deosebesc cu o singură stare;
3. mintermenii care se deosebesc numai cu o singură stare, indiferent unde sunt poziționați în tabel, se mai numesc „vecini” și pot forma grupe de mintermeni;
4. mintermenii, poziționați pe diagonală, se deosebesc cu 2 și mai multe stări și nu sunt vecini.

Procesul de minimizare a funcției logice include următoarele etape:

1. Pregătirea tabelului de mintermeni (în fig. 1.23 este prezentat tabelul de mintermeni pentru o funcție logică cu 4 variabile).

cd\ab	00	01	11	10
00				
01				
11				
10				

Fig. 1.23. Tabelul de mintermeni pentru o funcție logică cu 4 variabile.

2. Introducerea mintermenilor în tabel (în tabelul din fig. 1.24 sunt introdusi mintermenii funcției (3)).

cd\ab	00	01	11	10
00	1	1	1	1
01	1		1	1
11	1	1		
10	1	1	1	1

Fig. 1.24. Tabelul de mintermeni completat cu unități.

3. Formarea grupurilor de mintermeni (vezi fig. 1.25). În procesul formării grupurilor de mintermeni trebuie de ținut cont de următoarele reguli:

- numărul de celule în grup poate avea numai următoarele valori $N_{cel} = 2^l$, unde $l = 0, 1, 2, 3, \dots$;
- în grup pot fi incluse numai celulele cu unități;
- în grup trebuie inclus un număr maximal posibil de celule;
- grupurile formate se pot suprapune (celulele unui grup pot fi parte componentă ale altor grupuri);
- de respectat condiția de simetrie a configurației grupului format în raport cu linia de centru a tabelului sau numărul de celule în grup de pe ambele părți a liniei de centru al tabelului trebuie să fie același.

4. Notarea grupurilor. Pentru notarea grupurilor pot fi utilizate oricare litere cu indice. În fig. 1.25 grupurile sunt notate prin k_0, k_1, k_2 și k_3 .

cd\ab	00	01	11	10
00	1	1	1	1
01	1		1	1
11	1	1		
10	1	1	1	1

Diagram illustrating the grouping of minterms in the table. The minterms are grouped as follows:

- k_0 covers the first column (minterms 00, 01, 11, 10).
- k_1 covers the second column (minterms 00, 01, 11, 10).
- k_2 covers the third column (minterms 00, 01, 11, 10).
- k_3 covers the fourth column (minterms 00, 01, 11, 10).

Fig. 1.25. Tabel cu grupuri din mintermeni egali cu 1.

5. Scrierea pentru fiecare grup de mintermeni a unui produs din variabilele care nu-și schimbă starea. Conform grupurilor formate în fig. 1.25 pentru k_0 , k_1 , k_2 și k_3 obținem:

$$k_0 = \bar{a}\bar{b}; \quad k_1 = \bar{a}c; \quad k_2 = a\bar{c}; \quad k_3 = \bar{d}.$$

6. Scrierea formulei pentru funcția logică minimizată.

$$F(a, b, c, d) = k_0 + k_1 + k_2 + k_3 = \bar{a}\bar{b} + \bar{a}c + a\bar{c} + \bar{d}. \quad (4)$$

Conform formulei (4) schema electrică conține 3 elemente NU, 3 elemente ȘI și un element SAU.

Formula (4) a fost obținută prin gruparea mintermenilor egali cu 1. Funcția logică poate fi minimizată și prin gruparea mintermenilor egali cu 0 (vezi fig. 1.26). În acest caz pentru fiecare grup este necesar de scris o sumă din variabilele care nu-și schimbă starea, iar variabilele se scriu în starea inversată în comparație cu starea care au avut-o în tabel.

$cd \backslash ab$	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

l_0

l_1

Fig. 1.26. Tabel cu grupuri din mintermeni egali cu 0.

Conform grupurilor formate în fig. 1.26 pentru l_0 și l_1 obținem:

$$l_0 = a + \bar{b} + c + \bar{d}; \quad l_1 = \bar{a} + \bar{c} + \bar{d};$$

Formula pentru funcția logică minimizată se scrie ca produsul mărimilor l_0 și l_1 :

$$F(a, b, c, d) = l_0 \times l_1 = (a + \bar{b} + c + \bar{d}) \times (\bar{a} + \bar{c} + \bar{d}). \quad (5)$$

Conform formulei (5) schema electrică conține 3 elemente NU, 1 element ȘI și 2 elemente SAU.

În fig. 1.27(a,b,c) sunt prezentate schemele electrice, construite conform funcțiilor (3), (4) și (5). Tabelele de adevăr, obținute cu ajutorul Convertorului Logic, demonstrează că schemele sunt echivalente.

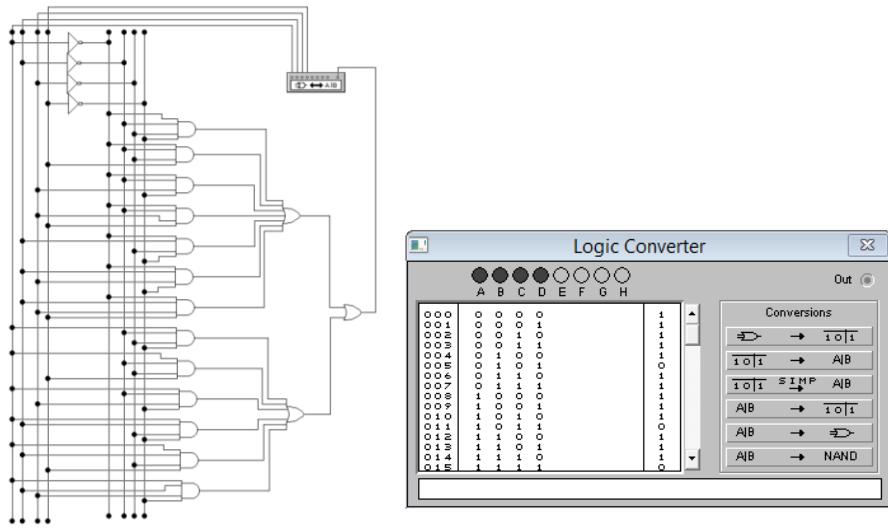


Fig. 1.27(a). Schema electrică, construită conform funcției (3) neminimizate.

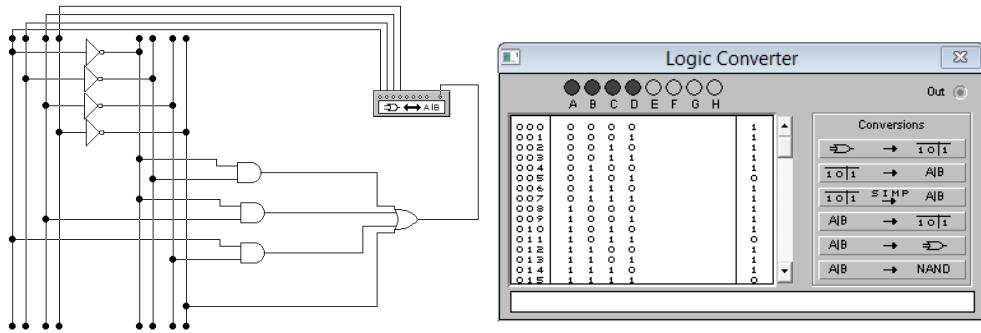


Fig. 1.27(b). Schema electrică, construită conform funcției (4).

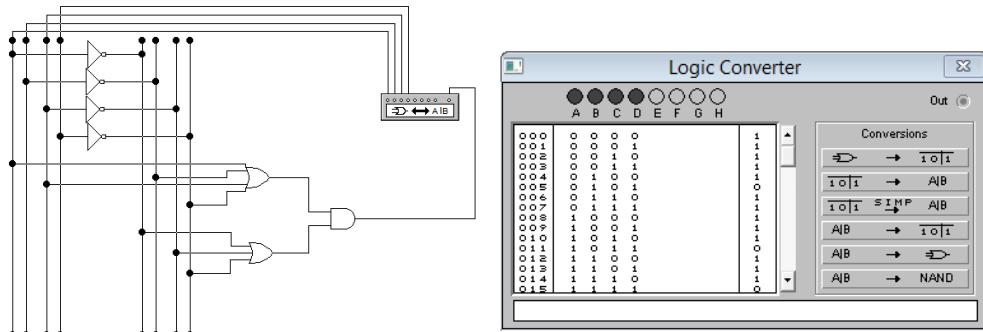


Fig. 1.27(c). Schema electrică, construită conform funcției (5).

În continuare vor fi prezentate tabelele de mintermeni pentru funcții logice cu 5 și 6 variabile.

5. Tabelul de mintermeni pentru o funcție logică cu 5 variabile are 32 de celule și poate fi prezentat în 2 variante:

a) 2 tabele separate cîte 16 celule fiecare (vezi fig. 1.28(a));

b) un singur tabel cu 32 de celule (vezi fig. 1.28(b)).

Drept exemplu, în tabelele din 1.28(a,b) sunt introdusi mintermenii funcției logice

$$F(a, b, c, d, e) = \bar{a}b\bar{c}d\bar{e} + a\bar{b}\bar{c}d\bar{e}$$

		\bar{a}				a					
		00	01	11	10	00	01	11	10	bc/de	
de\bc	00									00	
	01									01	
	11									11	
	10				1					10	
						1					

Fig. 1.28(a). Tabelul de mintermeni pentru o funcție logică cu 5 variabile (2 tabele separate).

În cazul utilizării configurației din două tabele separate minimizarea funcției logice cu 5 variabile se face pentru fiecare tabel aparte.

		000	001	011	010	110	111	101	100	abc/de	
		00								00	
de\abc	00									01	
	01									11	
	11									10	
	10			1							
						1					

Fig. 1.28(b). Tabelul de mintermeni pentru o funcție logică cu 5 variabile (un tabel unic).

6. Tabelul de mintermeni pentru o funcție logică cu 6 variabile are 64 de celule și poate fi prezentat în 2 variante:

- a) 4 tabele separate cîte 16 celule fiecare (vezi fig. 1.29(a));
- b) un singur tabel cu 64 de celule (vezi fig. 1.29(b)).

Drept exemplu, în tabelele din fig. 1.29(a,b) sunt introdusi mintermenii funcției logice

$$F(a, b, c, d, e, f) = \bar{a}bcd\bar{e}f + a\bar{b}\bar{c}def\bar{f}.$$

		\bar{a}				a					
		00	01	11	10	00	01	11	10	cd/ef	
\bar{b}	00									00	
	01									01	
	11									11	
	10									10	
						1					
		00								00	
		01			1					01	
		11								11	
		10								10	
										cd\ef	
		ef\cd	00	01	11	10	00	01	11	10	

Fig. 1.29(a). Tabelul de mintermeni pentru o funcție logică cu 6 variabile (4 tabele separate).

În cazul utilizării configurației din patru tabele separate minimizarea funcției logice cu 6 variabile se face pentru fiecare tabel aparte.

$def \setminus abc$	000	001	011	010	110	111	101	100	abc / def
$def \setminus abc$	000								000
000									001
001									011
011									010
010									110
110								1	111
111									101
101			1						100
100									
$def \setminus abc$	000	001	011	010	110	111	101	100	$abc \setminus def$

Fig. 1.29(b). Tabelul de mintermeni pentru o funcție logică cu 6 variabile (un tabel unic).

Lucru independent nr. 4.

1. De minimizat următoarea funcție logică prin metoda Veitch-Karnaugh
- $$F(a, b, c, d, e, f) = \Sigma(0, 1, 2, 6, 7, 8, 9, 10, 12, 19, 20, 21, 22, 23, 24, 25, 27, 34, 35, 36, 37, 38, 42, 43, 44, 45, 51, 52, 53, 54, 55, 56)$$
- a) prin gruparea mintermenilor egali cu 1;
 - b) prin gruparea mintermenilor egali cu 0;
 - c) construiți schemele electrice pentru varianta (a) și varianta (b) (utilizați Convertorul logic pentru a obține tabela de adevăr și demonstra că schemele sunt echivalente);
 - d) comparați numărul de elemente NU, ȘI, SAU pentru cazul funcției neminimizate, cazul (a) și cazul (b).

2. BISTABILI, REGISTRE ȘI CONTOARE

Bistabilii, registrele și contoarele (se mai numesc dispozitive secvențiale) sunt dispozitivele principale ale calculatorului și se caracterizează prin următoarele proprietăți:

- pot păstra informația sau au memorie;
- starea următoare depinde de starea precedentă;
- starea la ieșirile dispozitivului nu depinde univoc de starea aplicată la intrările lui.

2.1. Clasificarea bistabililor

Definiție. Numim **bistabil** un dispozitiv electronic logic care se poate afla în una din cele 2 stări posibile ($Q = 1$; $\bar{Q} = 0$) sau ($Q = 0$; $\bar{Q} = 1$) și se utilizează pentru înscrierea, păstrarea și prezentarea unui bit (doi biți) de informație.

În prezent există mai multe scheme electrice ale bistabililor. Bistabilii se clasifică după următoarele criterii:

- numărul de trepte;
- sincroni sau asincroni.

Clasificarea după criteriul „numărul de trepte”. Conform numărului de trepte există bistabili cu una, două și mai multe trepte. Din bistabilii cu o treaptă, drept exemplu, pot fi menționati bistabilii RS-asincron, RS-sincron și de tip D. Din bistabilii cu două trepte aici vor fi prezentati bistabilii MS, DV și JK.

Clasificarea după criteriul „sincron sau asincron”. Bistabilii care nu au la intrare canal de dirijare (canal de comandă) nu pot fi dirijați de program și se numesc bistabili asincroni. Bistabilii care au la intrare canal de dirijare (canal de comandă) pot fi dirijați de program și se numesc bistabili sincroni.

Bistabilii se utilizează pe larg pentru proiectarea și construirea dispozitivelor numerice cu memorie (registre, contoare etc).

2.2. Bistabili cu o treaptă RS-asincron, RS-sincron și de tip D

2.2.1. Bistabilul RS-asincron

Bistabilii pot fi construiți din elemente logice SAU-NU sau ȘI-NU.

Schema electrică a bistabilului RS-asincron construită din elemente logice SAU-NU este prezentată în fig. 2.1. Bistabilul RS-asincron construit din elemente logice SAU-UN are o singură treaptă.

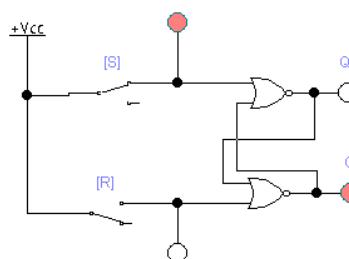


Fig. 2.1. Schema electrică a bistabilului RS-asincron construită din elemente logice SAU-NU.

În schema electrică a bistabilului RS-asincron construită din elemente logice SAU-NU sunt următoarele notări:

- +V_{cc} – sursa de alimentare;
S – set (introducerea, plasarea, înscrierea informației);

R – reset (inițializarea, instalarea bistabilului în starea inițială, ștergerea informației);

Q – ieșirea directă (canalul care prezintă informația în stare directă);

\bar{Q} – ieșirea inversă (canalul care prezintă informația în stare inversă).

Pentru a prezenta procesul de lucru al bistabilului, este necesar de scris funcțiile logice ale schemei, de aplicat la intrările S și R diferite combinații de 0 și 1, unde 0 – nu este tensiune, iar 1 – este tensiune. Pentru fiecare combinație aplicată la intrările S și R este necesar de a calcula Q și \bar{Q} și de a completa tabelul de adevăr al bistabilului.

Pentru schema electrică a bistabilului din fig. 2.1 pot fi scrise următoarele 2 funcții logice:

$$Q = \overline{R + \bar{Q}}; \quad (2.2.1)$$

$$\bar{Q} = \overline{S + Q}. \quad (2.2.2)$$

Calculele Q și \bar{Q} se efectuează în următoarea consecutivitate:

$$S = 0, R = 0 \rightarrow Q = \overline{R + \bar{Q}} = \overline{0 + \bar{Q}} = Q; \bar{Q} = \overline{\overline{S + Q}} = \overline{\overline{0 + Q}} = \bar{Q};$$

$$S = 0, R = 1 \rightarrow Q = \overline{R + \bar{Q}} = \overline{1 + \bar{Q}} = 0; \bar{Q} = \overline{\overline{S + Q}} = \overline{\overline{1 + Q}} = 1;$$

$$S = 1, R = 0 \rightarrow \bar{Q} = \overline{S + Q} = \overline{1 + Q} = 0; Q = \overline{\overline{R + \bar{Q}}} = \overline{\overline{0 + \bar{Q}}} = 1;$$

$$S = 1, R = 1 \rightarrow \bar{Q} = \overline{S + Q} = \overline{1 + Q} = 0; Q = \overline{\overline{R + \bar{Q}}} = \overline{\overline{1 + \bar{Q}}} = 0.$$

Completăm tabelul de adevăr (vezi tabelul 2.1).

Tabelul 2.1. Tabelul de adevăr al bistabilului RS-asincron construit din elemente logice SAU-UN.

Nr. d/o	S	R	Q	\bar{Q}	Regimul de lucru
0	0	0	Q	\bar{Q}	Păstrarea stării precedente
1	0	1	0	1	Inițializarea bistabilului
2	1	0	1	0	Înscrierea informației
3	1	1	0	0	Stare interzisă

Schema electrică a bistabilului RS-asincron, construit din elemente logice SAU-NU, are următoarele neajunsuri:

- nu are canal de dirijare, din aceasta cauză el nu poate fi dirijat de program;
- dacă se înscrie o informație nouă, atunci informația precedentă se sterge și nu poate fi restabilită (bistabilul are o singură treaptă);
- starea $S = 1, R = 1$ este interzisă.

Schema electrică a bistabilului RS-asincron construită din elemente logice ȘI-NU este prezentată în fig. 2.2. Bistabilul RS-asincron construit din elemente logice ȘI-NU are o singură treaptă.

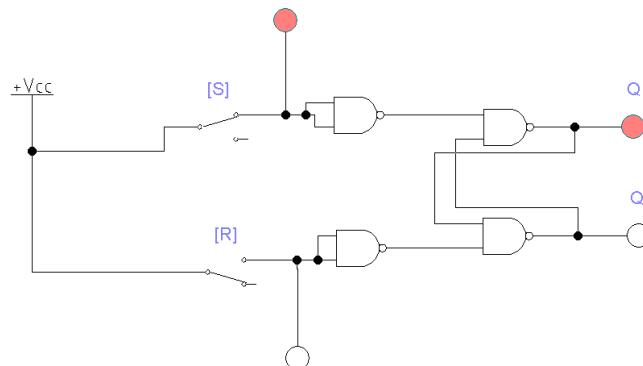


Fig. 2.2. Schema electrică a bistabilului RS-asincron construită din elemente logice ȘI-NU.

În schema electrică a bistabilului RS-asincron construită din elemente logice řI-NU sunt următoarele notări:

$+V_{cc}$ – sursa de alimentare;

S – set (introducerea, plasarea, înscrierea informației);

R – reset (inițializarea, instalarea bistabilului în stare inițială, ștergerea informației);

Q – ieșirea directă (canalul care prezintă informația în stare directă);

\bar{Q} – ieșirea inversă (canalul care prezintă informația în stare inversă).

Pentru a prezenta procesul de lucru al bistabilului, este necesar de scris funcțiile logice ale schemei, de aplicat la intrările S și R diferite combinații de 0 și 1, unde 0 – nu este tensiune, iar 1 – este tensiune. Pentru fiecare combinație aplicată la intrările S și R de calculat Q și \bar{Q} și de completat tabelul de adevăr al bistabilului.

Pentru schema electrică a bistabilului din fig. 2.2 pot fi scrise următoarele 2 funcții logice:

$$Q = \overline{\bar{S} \times \bar{Q}}; \quad (2.2.3)$$

$$\bar{Q} = \overline{\bar{R} \times Q}. \quad (2.2.4)$$

Calculele Q și \bar{Q} se efectueză în următoarea consecutivitate:

$$S = 0, R = 0 \rightarrow Q = \overline{\bar{S} \times \bar{Q}} = \overline{1 \times \bar{Q}} = Q; \bar{Q} = \overline{\bar{R} \times Q} = \overline{1 \times Q} = \bar{Q};$$

$$S = 0, R = 1 \rightarrow \bar{Q} = \overline{\bar{R} \times Q} = \overline{0 \times Q} = 1; Q = \overline{\bar{S} \times \bar{Q}} = \overline{1 \times \bar{Q}} = 0;$$

$$S = 1, R = 0 \rightarrow Q = \overline{\bar{S} \times \bar{Q}} = \overline{0 \times \bar{Q}} = 1; \bar{Q} = \overline{\bar{R} \times Q} = \overline{1 \times Q} = 0;$$

$$S = 1, R = 1 \rightarrow \bar{Q} = \overline{\bar{R} \times Q} = \overline{0 \times Q} = 1; Q = \overline{\bar{S} \times \bar{Q}} = \overline{0 \times \bar{Q}} = 1.$$

Completăm tabelul de adevăr (vezi tabelul 2.2).

Tabelul 2.2. Tabelul de adevăr al bistabilului RS-asincron construit din elemente logice řI-NU

Nr. d/o	S	R	Q	\bar{Q}	Regimul de lucru
0	0	0	Q	\bar{Q}	Păstrarea stării precedente
1	0	1	0	1	Inițializarea bistabilului
2	1	0	1	0	Înscrierea informației
3	1	1	1	1	Stare interzisă

Schema electrică a bistabilului RS-asincron, construit din elemente logice řI-NU, are următoarele neajunsuri:

- nu are canal de dirijare, din aceasta cauză el nu poate fi dirijat de program;
- dacă se înscrive o informație nouă, atunci informația precedentă se sterge și nu poate fi restabilită (bistabilul are o singură treaptă);
- starea $S = 1, R = 1$ este interzisă.

În fig. 2.3 este prezentat simbolul convențional al bistabililor asincroni construiți din elemente logice SAU-NU sau řI-NU.

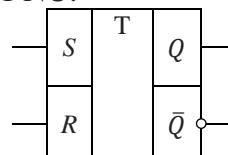


Fig. 2.3. Simbolul convențional al bistabililor RS-asincroni.

2.2.2. Bistabilul RS-sincron

Schema electrică a bistabilului RS-sincron construită din elemente logice ȘI-NU este prezentată în fig. 2.4. Bistabilul RS-sincron construit din elemente logice ȘI-NU are o singură treaptă.

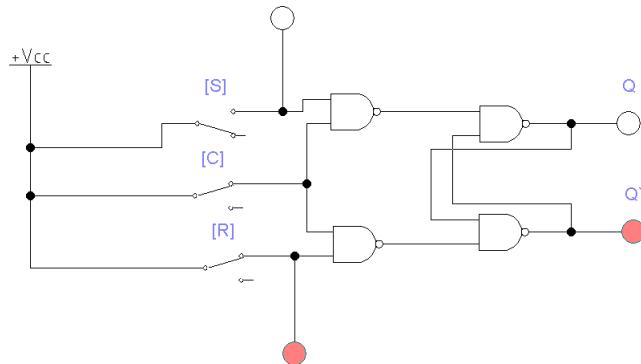


Fig. 2.4. Schema electrică a bistabilului RS-sincron construită din elemente logice ȘI-NU.

În schema electrică a bistabilului RS-sincron construită din elemente logice ȘI-NU sunt următoarele notări:

$+V_{cc}$ – sursa de alimentare;

S – set (introducerea, plasarea, înscrierea informației);

C – canalul de dirijare/comandă;

R – reset (inițializarea, instalarea bistabilului în stare inițială, stergerea informației);

Q – ieșirea directă (canalul care prezintă informația în stare directă);

\bar{Q} – ieșirea inversă (canalul care prezintă informația în stare inversă).

Pentru a prezenta procesul de lucru al bistabilului, este necesar de scris funcțiile logice ale schemei, de aplicat la intrările C , S și R diferite combinații de 0 și 1, unde 0 – nu este tensiune, iar 1 – este tensiune. Pentru fiecare combinație aplicată la intrările C , S și R de calculat Q și \bar{Q} și de completat tabelul de adevăr al bistabilului.

Pentru schema electrică a bistabilului din fig. 2.4 pot fi scrise următoarele 2 funcții logice:

$$Q = \overline{\overline{C} \times S \times \bar{Q}}; \quad (2.2.4)$$

$$\bar{Q} = \overline{\overline{C} \times R \times Q}. \quad (2.2.5)$$

Calculele Q și \bar{Q} se efectuează în următoarea consecutivitate:

$$C = 0, S = 0, R = 0 \rightarrow Q = \overline{\overline{C} \times S \times \bar{Q}} = \overline{1 \times \bar{Q}} = Q; \quad \bar{Q} = \overline{\overline{C} \times R \times Q} = \overline{1 \times Q} = \bar{Q};$$

$$C = 0, S = 0, R = 1 \rightarrow Q = \overline{\overline{C} \times S \times \bar{Q}} = \overline{1 \times \bar{Q}} = Q; \quad \bar{Q} = \overline{\overline{C} \times R \times Q} = \overline{1 \times Q} = \bar{Q};$$

$$C = 0, S = 1, R = 0 \rightarrow Q = \overline{\overline{C} \times S \times \bar{Q}} = \overline{1 \times \bar{Q}} = Q; \quad \bar{Q} = \overline{\overline{C} \times R \times Q} = \overline{1 \times Q} = \bar{Q};$$

$$C = 0, S = 1, R = 1 \rightarrow Q = \overline{\overline{C} \times S \times \bar{Q}} = \overline{1 \times \bar{Q}} = Q; \quad \bar{Q} = \overline{\overline{C} \times R \times Q} = \overline{1 \times Q} = \bar{Q};$$

$$C = 1, S = 0, R = 0 \rightarrow Q = \overline{\overline{C} \times S \times \bar{Q}} = \overline{1 \times \bar{Q}} = Q; \quad \bar{Q} = \overline{\overline{C} \times R \times Q} = \overline{1 \times Q} = \bar{Q};$$

$$C = 1, S = 0, R = 1 \rightarrow \bar{Q} = \overline{\overline{C} \times R \times Q} = \overline{0 \times Q} = 1; \quad Q = \overline{\overline{C} \times S \times \bar{Q}} = \overline{1 \times 1} = 0;$$

$$C = 1, S = 1, R = 0 \rightarrow Q = \overline{\overline{C} \times S \times \bar{Q}} = \overline{0 \times \bar{Q}} = 1; \quad \bar{Q} = \overline{\overline{C} \times R \times Q} = \overline{1 \times 1} = 0;$$

$$C = 1, S = 1, R = 1 \rightarrow \bar{Q} = \overline{\overline{C} \times R \times Q} = \overline{0 \times Q} = 1; \quad Q = \overline{\overline{C} \times S \times \bar{Q}} = \overline{0 \times 1} = 1.$$

Completăm tabelul de adevăr (vezi tabelul 2.3).

Tabelul 2.3. Tabelul de adevăr al bistabilului RS-sincron construit din elemente logice ȘI-NU

Nr. d/o	C	S	R	Q	\bar{Q}	Regimul de lucru
0	0	0	0	Q	\bar{Q}	Păstrarea stării precedente, $C = 0$
1	0	0	1	Q	\bar{Q}	Păstrarea stării precedente, $C = 0$
2	0	1	0	Q	\bar{Q}	Păstrarea stării precedente, $C = 0$
3	0	1	1	Q	\bar{Q}	Păstrarea stării precedente, $C = 0$
4	1	0	0	Q	\bar{Q}	Păstrarea stării precedente, $C = 1$
5	1	0	1	0	1	Inițializarea bistabilului
6	1	1	0	1	0	Înscrierea informației
7	1	1	1	1	1	Stare interzisă

Din tabelul 2.3 se vede, că pentru $C = 0$ schimbarea stărilor S și R nu schimbă regimul de lucru al bistabilului. Avantajul bistabilului RS-sincron – are canal de comandă și poate fi dirijat de program.

Schema electrică a bistabilului RS-sincron, construit din elemente logice ȘI-NU, are următoarele neajunsuri:

- a) dacă se înscrie o informație nouă, atunci informația precedentă se sterge și nu poate fi restabilită (bistabilul are o singură treaptă);
- b) starea $S = 1, R = 1$ este interzisă.

În fig. 2.5 este prezentat simbolul convențional al bistabililor RS-sincroni construiți din elemente logice ȘI-NU/SAU-NU.

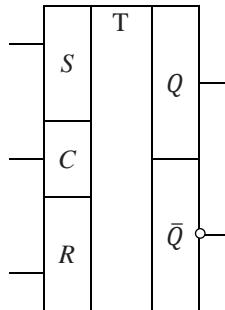


Fig. 2.5. Simbolul convențional al bistabililor RS-sincroni.

2.2.3. Bistabilul de tip D

Bistabilul de tip D a fost construit cu scopul înlăturării neajunsului bistabilului RS-sincron care nu permite aplicarea stării $S = 1, R = 1$.

Schema electrică a bistabilului de tip D construită din elemente logice ȘI-NU este prezentată în fig. 2.6. Bistabilul de tip D construit din elemente logice ȘI-NU are o singură treaptă.

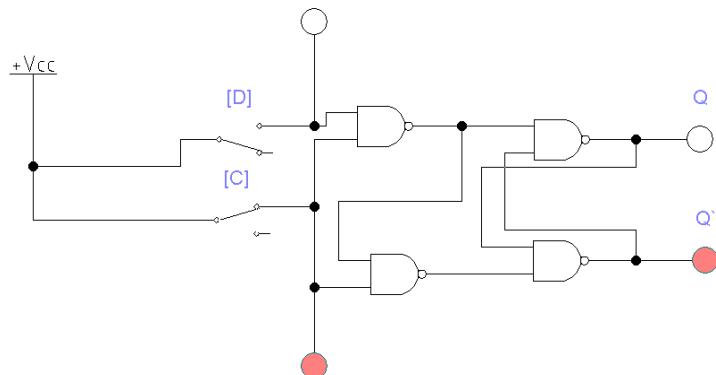


Fig. 2.6. Schema electrică a bistabilului de tip D construită din elemente logice řI-NU.

În schema electrică a bistabilului de tip D construită din elemente logice řI-NU sunt următoarele notări:

$+V_{cc}$ – sursa de alimentare;

D – introducerea, plasarea, înscrierea informației;

C – canalul de dirijare/comandă;

Q – ieșirea directă (canalul care prezintă informația în stare directă);

\bar{Q} – ieșirea inversă (canalul care prezintă informația în stare inversă).

Pentru a prezenta procesul de lucru al bistabilului, este necesar de scris funcțiile logice ale schemei, de aplicat la intrările C și D diferite combinații de 0 și 1, unde 0 – nu este tensiune, iar 1 – este tensiune. Pentru fiecare combinație aplicată la intrările C și D de calculat Q și \bar{Q} și de completat tabelul de adevăr al bistabilului.

Pentru schema electrică a bistabilului din fig. 2.6 pot fi scrise următoarele 2 funcții logice:

$$Q = \overline{\overline{C} \times \overline{D} \times \overline{Q}}; \quad (2.2.6)$$

$$\bar{Q} = \overline{C \times \overline{C} \times \overline{D}} \times Q. \quad (2.2.7)$$

Calculele Q și \bar{Q} se efectuează în următoarea consecutivitate:

$$C = 0, D = 0 \rightarrow Q = \overline{\overline{C} \times \overline{D} \times \overline{Q}} = \overline{0 \times 0 \times \overline{Q}} = \overline{1 \times \overline{Q}} = Q;$$

$$\bar{Q} = \overline{C \times \overline{C} \times \overline{D}} \times Q = \overline{0 \times \overline{0} \times \overline{0}} \times Q = \overline{1 \times Q} = \bar{Q};$$

$$C = 0, D = 1 \rightarrow Q = \overline{\overline{C} \times \overline{D} \times \overline{Q}} = \overline{0 \times 1 \times \overline{Q}} = \overline{1 \times \overline{Q}} = Q;$$

$$\bar{Q} = \overline{C \times \overline{C} \times \overline{D}} \times Q = \overline{0 \times \overline{0} \times 1} \times Q = \overline{1 \times Q} = \bar{Q};$$

$$C = 1, D = 0 \rightarrow \bar{Q} = \overline{C \times \overline{C} \times \overline{D}} \times Q = \overline{1 \times \overline{1} \times \overline{0}} \times Q = \overline{0 \times Q} = 1;$$

$$Q = \overline{\overline{C} \times \overline{D} \times \overline{Q}} = \overline{1 \times 0 \times 1} = \overline{1 \times 1} = 0;$$

$$C = 1, D = 1 \rightarrow Q = \overline{\overline{C} \times \overline{D} \times \overline{Q}} = \overline{1 \times 1 \times \overline{Q}} = \overline{0 \times \overline{Q}} = 1;$$

$$\bar{Q} = \overline{C \times \overline{C} \times \overline{D}} \times Q = \overline{1 \times \overline{1} \times 1} \times 1 = \overline{1 \times 1} = 0.$$

Introducem datele obținute în tabelul 2.4.

Tabelul 2.4. Tabelul de adevăr al bistabilului de tip D

Nr. d/o	C	D	Q	\bar{Q}	Regimul de lucru
0	0	0	Q	\bar{Q}	Păstrarea stării precedente, $C = 0$
1	0	1	Q	\bar{Q}	Păstrarea stării precedente, $C = 0$
2	1	0	0	1	Inițializarea bistabilului
3	1	1	1	0	Înscrierea informației

Din tabelul 2.4 pot fi determinate avantajele și neajunsurile bistabilului de tip D. Avantajele bistabilului de tip D sunt următoarele:

- nu are stări interzise;
- are canal de comandă și poate fi dirijat de program.

Bistabilul de tip D are următorul neajuns:

- dacă se înscrive o informație nouă, atunci informația precedentă se sterge și nu poate fi restabilită (bistabilul are o singură treaptă).

În fig. 2.7 este prezentat simbolul convențional al bistabilului de tip D.

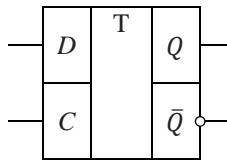


Fig. 2.7. Simbolul convențional al bistabililor de tip D.

Pentru construirea dispozitivelor logice din bistabilii cu o treaptă RS-asincron, RS-sincron și de tip D cel mai des se folosește bistabilul de tip D.

2.2.4. Divizor al frecvenței semnalului dreptunghiular

Într-un calculator se utilizează semnale dreptunghiulare cu frecvențe diferite. Pentru a obține semnale dreptunghiulare cu frecvențe diferite des se folosește bistabilul D cu circuit cu reacție care funcționează ca divizor de frecvență. Schema electrică a divizorului frecvenței construit din bistabilul D este prezentată în fig. 2.8.

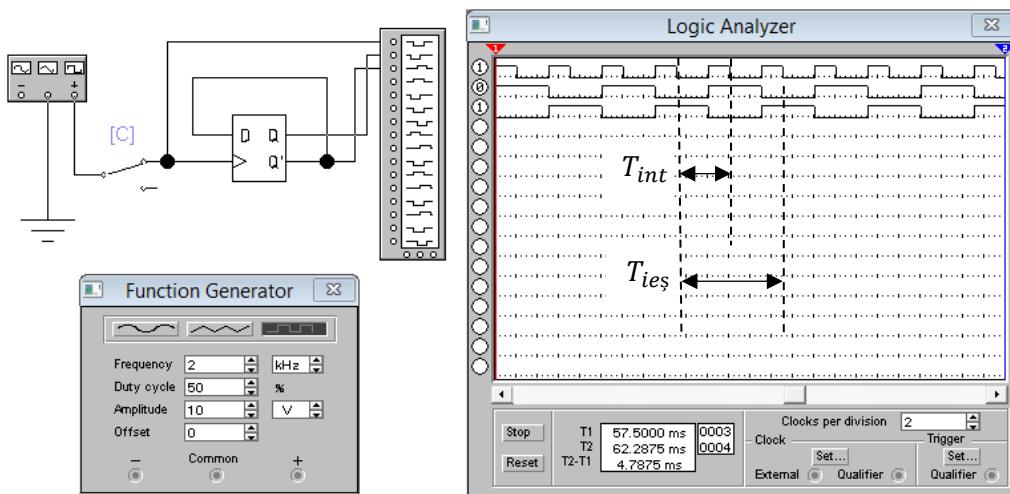


Fig. 2.8. Schema electrică a divizorului frecvenței construit din bistabilul D.

În schema electrică din fig. 2.8 sunt prezentați generatorul de semnale, comutatorul C, și bistabilul D. Ieșirea inversă \bar{Q} a bistabilului D este conectată cu intrarea lui. Pe ecranul analizatorului logic sunt prezentate diagramele temporale pentru canalele C , Q și \bar{Q} .

Din fig. 2.8 poate fi determinată valoarea coeficientului de divizare al frecvenței $K_{div} = f_{int} / f_{ies}$, unde f_{int} este frecvența semnalului aplicat la intrarea C a bistabilului, iar f_{ies} – frecvența semnalului obținut la ieșirile Q și \bar{Q} ale bistabilului. Conform schemei electrice generatorul emite un semnal f_C la intrarea bistabilului D care are frecvența $f_{int} = f_C = 2 \text{ kHz}$. Pe panoul analizatorului logic se vede, că perioada semnalului la ieșirea bistabilului $T_{ies} = 2T_{int}$. Deoarece $f_{ies} = 1/T_{ies}$ pentru K_{div} se obține:

$$K_{div} = \frac{f_{int}}{f_{ies}} = \frac{T_{ies}}{T_{int}} = 2 \text{ sau } f_{ies} = 1 \text{ kHz.}$$

Pentru a obține $K_{div} = 4, 8, 16, \dots$ bistabilii D cu circuit de reacție se conectează în schema electrică consecutiv.

2.3. Bistabili cu două trepte MS, DV și JK

2.3.1. Bistabilul de tip MS

Bistabilul de tip MS poate fi construit din doi bistabili RS-sincroni. Schema electrică a bistabilului MS, construită din elementele logice řI-NU, este prezentată în fig. 2.8.

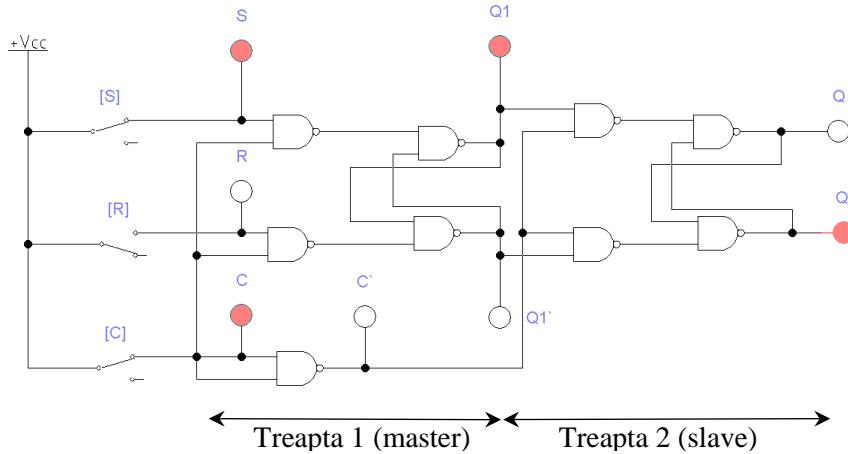


Fig. 2.8. Schema electrică a bistabilului MS.

În schema electrică a bistabilului MS construită din elemente logice řI-NU sunt următoarele notări:

$+V_{cc}$ – sursa de alimentare;

S – set (introducerea, plasarea, înscrierea informației);

R – reset (initializarea, instalarea bistabilului în stare inițială, stergerea informației);

C – canalul de dirijare/comandă;

Q_1 – ieșirea directă a primei trepte (canalul care prezintă informația în stare directă);

\bar{Q}_1 – ieșirea inversă a primei trepte (canalul care prezintă informația în stare inversă);

Q – ieșirea directă (canalul care prezintă informația în stare directă);

\bar{Q} – ieșirea inversă (canalul care prezintă informația în stare inversă).

Bistabilul MS are 2 trepte (vezi fig. 2.8): prima treaptă se numește „master” sau treaptă conducătoare; treapta a doua se numește „slave” sau treaptă condusă.

Pentru prezentarea procesului de lucru al bistabilului MS pot fi folosite funcțiile logice sau diagramele temporale. În rezultatul efectuării calculelor sau analizei diagramelor temporale este necesar de completat tabelul de adevăr.

Conform notărilor din fig. 2.8 funcțiile logice pentru prima treaptă (Q_1 și \bar{Q}_1) și ieșirile (Q și \bar{Q}) ale bistabilului MS sunt următoarele:

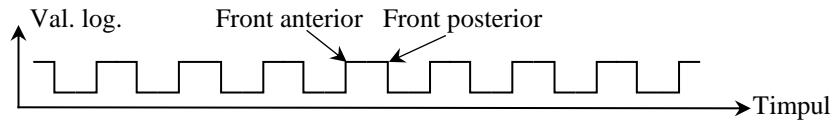
$$Q_1 = \overline{\bar{C} \times \bar{S} \times \bar{Q}_1}; \quad (2.3.1)$$

$$\bar{Q}_1 = \overline{\bar{C} \times R \times Q_1}; \quad (2.3.2)$$

$$Q = \bar{C} \times Q_1 \times \bar{Q}; \quad (2.3.3)$$

$$\bar{Q} = \bar{C} \times \bar{Q}_1 \times Q. \quad (2.3.4)$$

În continuare v-a fi prezentat procesul de lucru al bistabilului MS cu ajutorul diagramelor temporale. Diagrama temporală reprezintă un grafic al semnalului dreptunghiular (vezi fig. 2.9), unde pe axa verticală este valoarea logică (0, 1), iar pe axa orizontală – timpul.



Diagramele temporale se obțin cu ajutorul analizatorului logic, conectat la schema electrică a bistabilului MS (vezi fig. 2.10).

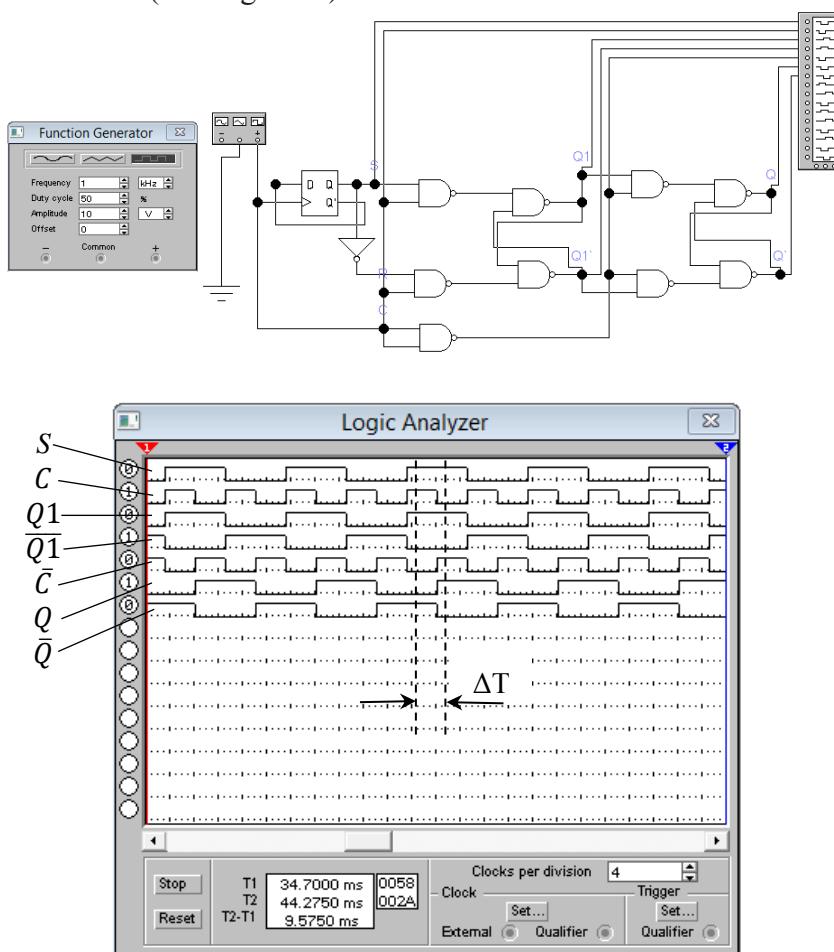


Fig. 2.10. Prezentarea procesului de lucru al bistabilului MS prin diagrame temporale.
În schema electrică din fig. 2.10 sunt prezentate următoarele dispozitive:

- generatorul de semnale (Function Generator);
- un bistabil de tip D necesar pentru a obține semnal la intrarea S a bistabilului MS;
- un inversor, conectat la ieșirea bistabilului D și intrarea R a bistabilului MS (este necesar pentru a exclude starea interzisă $S = 1, R = 1$);
- bistabilul MS construit din noua elemente logice řI-NU;
- un analizor logic (Logic Analyzer) necesar pentru a obține diagramele temporale.

Diagramele temporale, obținute de analizatorul logic sunt notate prin: S – canalul informației (canalul de date) care se înscrive în bistabil; C – canalul de dirijare; Q_1 – ieșirea directă a primei trepte a bistabilului; \bar{Q}_1 – ieșirea inversă a primei trepte a bistabilului;; \bar{C} – canalul de dirijare inversat; Q – ieșirea directă a treptei a doua a bistabilului; \bar{Q} – ieșirea inversă treptei a doua a bistabilului;

Analizând diagramele temporale se pot formula următoarele concluzii:

- starea primei trepte a bistabilului MS se schimbă la trecerea valorii logice a semnalului de comandă C de la 0 la 1 (după frontul anterior al semnalului de comandă C);

- starea treptei a doua a bistabilui MS se schimbă la trecerea valorii logice a semnalului de comandă \bar{C} de la 0 la 1 (după frontul anterior al semnalului de comandă \bar{C});
 - există intervale de timp ΔT când, după înscrierea informației noi în prima treaptă, informația precedentă din prima treaptă se păstrează în treapta a două;
 - păstrarea informației precedente în treapta a două, după înscrierea unei informații noi în prima treaptă, permite organizarea procedurii de restabilire a informației precedente.
- Completăm tabelul de adevăr (vezi tabelul 2.5).

Tabelul 2.5. Tabelul de adevăr al bistabilului MS construit din elemente logice řI-NU

Pentru prima treaptă a bistabilului MS						
Nr. d/o	C	S	R	Q_1	\bar{Q}_1	Regimul de lucru
0	0	0	0	Q	\bar{Q}	Păstrarea stării precedente, $C = 0$
1	0	0	1	Q	\bar{Q}	Păstrarea stării precedente, $C = 0$
2	0	1	0	Q	\bar{Q}	Păstrarea stării precedente, $C = 0$
3	0	1	1	Q	\bar{Q}	Păstrarea stării precedente, $C = 0$
4	1	0	0	Q	\bar{Q}	Păstrarea stării precedente, $C = 1$
5	1	0	1	0	1	Inițializarea bistabilului
6	1	1	0	1	0	Înscrierea informației
7	1	1	1	1	1	Stare interzisă
Pentru treapta a două a bistabilului MS						
Nr. d/o	\bar{C}	Q_1	\bar{Q}_1	Q	\bar{Q}	Regimul de lucru
0	0	0	0	Q	\bar{Q}	Păstrarea stării precedente, $\bar{C} = 0$
1	0	0	1	Q	\bar{Q}	Păstrarea stării precedente, $\bar{C} = 0$
2	0	1	0	Q	\bar{Q}	Păstrarea stării precedente, $\bar{C} = 0$
3	0	1	1	Q	\bar{Q}	Păstrarea stării precedente, $\bar{C} = 0$
4	1	0	0	Q	\bar{Q}	Păstrarea stării precedente, $\bar{C} = 1$
5	1	0	1	0	1	Inițializarea bistabilului
6	1	1	0	1	0	Înscrierea informației
7	1	1	1	1	1	Stare interzisă

Din tabelul 2.5 pot fi determinate avantajele și neajunsurile bistabilului MS. Avantajele bistabilului MS sunt următoarele:

- păstrarea informației precedente în treapta a două, după înscrierea unei informații noi în prima treaptă, permite organizarea procedurii de restabilire a informației precedente;
- are canal de comandă, este bistabil sincron și poate fi dirijat de program.

Bistabilul MS are următorul neajuns:

- starea $S = 1, R = 1$ este interzisă.

În fig. 2.11 este prezentat simbolul convențional al bistabililor MS construiți din elemente logice řI-NU/SAU-NU.

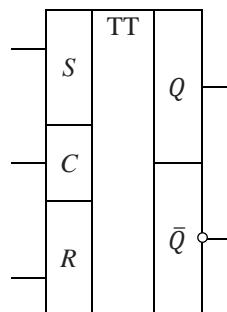


Fig. 2.11. Simbolul convențional al bistabililor MS.

2.3.2. Bistabilul de tip DV

În cazul bistabilului MS pentru înscrierea fiecărui bit de informație este necesar de a activa semnalul de comandă C . Pentru a înscri la rînd mai mulți biți cu aplicarea unui semnal de comandă schema bistabilului MS a fost modificată și a fost denumită bistabil DV.

Bistabilul DV are drept bază schema electrică a bistabilului MS în care este adăugat un element NU și un element SI (vezi fig. 2.12).

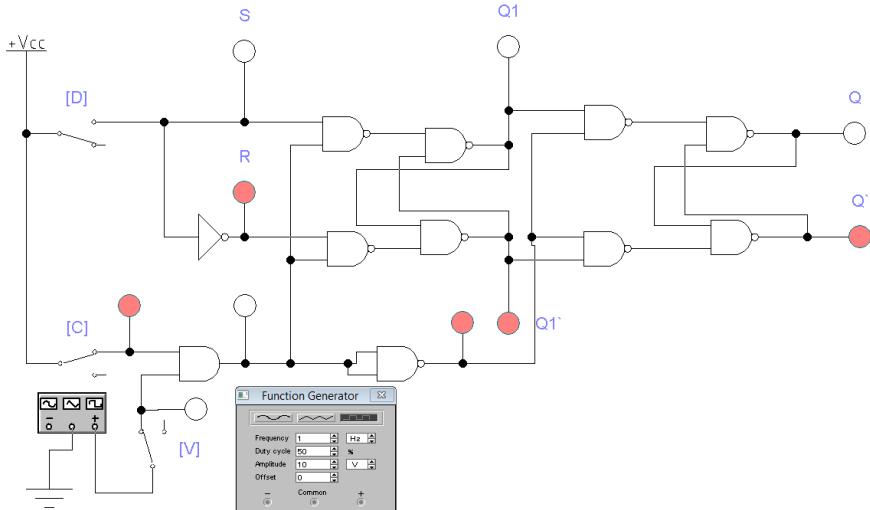


Fig. 2.12. Schema electrică a bistabilului MS.

Inversorul NU conectează intrările S și R ale bistabilului MS pentru a exclude aplicarea stării interzise $S = 1, R = 1$. La intrările elementului SI sunt conectate canalele C și V , unde C este canalul de comandă, iar pe canalul V se aplică permanent semnalul de tactare. Canalul D este canalul de date care pot fi înscrise în bistabilul DV. Semnalul de tactare este emis de generator.

Pentru bistabilul DV avem următoarele regimuri de lucru:

- $C = 0$, bistabilul este în regim de păstrare a informației;
- $C = 1$, bistabilul este în regim de înscriere a informației.

Simbolul convențional al bistabilului DV este prezentat în fig. 2.13.

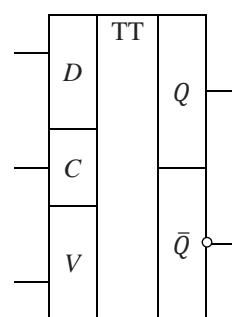


Fig. 2.13. Simbolul convențional al bistabililor DV.

2.3.3. Bistabilul de tip JK

Bistabilul JK are drept bază schema electrică a bistabilului MS în care sunt introduse două circuite cu reacție și două elemente SI-NU au trei intrări (vezi fig. 2.14).

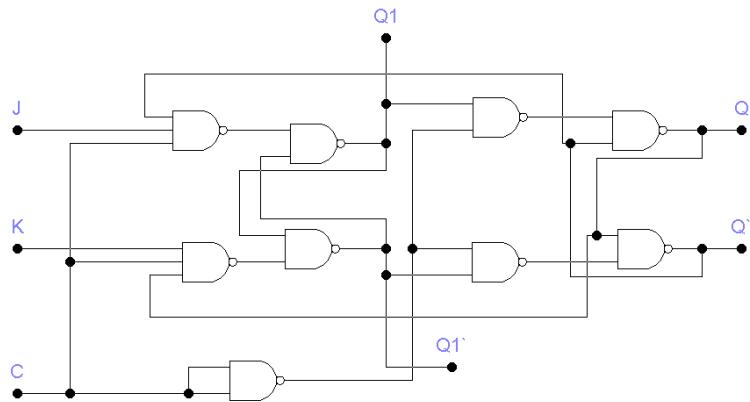


Fig. 2.14. Scema electrică a bistabilului JK.

În fig.2.14 sunt următoarele notări:

J – canalul de date;

K – se utilizează în special pentru inițializarea bistabilului;

C – canalul de comandă;

*Q*1 – ieșirea directă a primei trepte (canalul care prezintă informația în stare directă);

*Q*1' – ieșirea inversă a primei trepte (canalul care prezintă informația în stare inversă);

Q – ieșirea directă (canalul care prezintă informația în stare directă);

Q' – ieșirea inversă (canalul care prezintă informația în stare inversă).

Conform schemei din fig. 2.14 funcțiile logice executate de către bistabilul JK sunt următoarele:

$$Q_1 = \overline{C \times J \times \bar{Q}} \times \bar{Q}_1; \quad (2.3.5)$$

$$\bar{Q}_1 = \overline{C \times K \times Q} \times Q_1; \quad (2.3.6)$$

$$Q = \bar{C} \times Q_1 \times \bar{Q}; \quad (2.3.7)$$

$$\bar{Q} = \bar{C} \times \bar{Q}_1 \times Q. \quad (2.3.8)$$

Pentru prezentarea procesului de lucru al bistabilului JK poate fi utilizată schema electrică prezentată în fig. 2.15. Tabelul de adevăr se v-a completa, aplicînd la intrările bistabilului JK diferite combinații ale *C*, *J*, *K*, ținînd cont de stările anterioare *Q*_{an}, *Q*'_{an} (vezi tabelul 2.6).

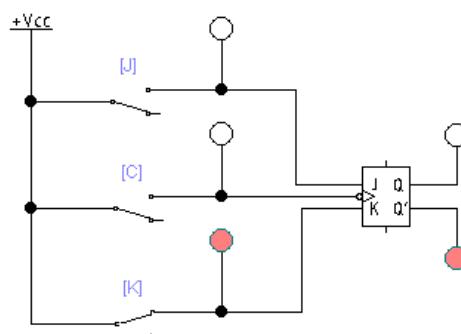


Fig. 2.15. Schema electrică pentru demonstrarea procesului de lucru al bistabilului JK.

Tabelul 2.6. Tabelul de adevăr al bistabilului JK

Nr. d/o	<i>C</i>	<i>J</i>	<i>K</i>	<i>Q</i> _{an}	<i>Q</i> ' _{an}	<i>Q</i>	<i>Q</i> '	Regim de lucru
------------	----------	----------	----------	------------------------	--------------------------	----------	------------	----------------

0	0	0	0	0	1	0	1	Păstrarea stării precedente, $C = 0$
1	0	0	0	1	0	1	0	Păstrarea stării precedente, $C = 0$
2	0	0	1	0	1	0	1	Păstrarea stării precedente, $C = 0$
3	0	0	1	1	0	1	0	Păstrarea stării precedente, $C = 0$
4	0	1	0	0	1	0	1	Păstrarea stării precedente, $C = 0$
5	0	1	0	1	0	1	0	Păstrarea stării precedente, $C = 0$
6	0	1	1	0	1	0	1	Păstrarea stării precedente, $C = 0$
7	0	1	1	1	0	1	0	Păstrarea stării precedente, $C = 0$
8	1	0	0	0	1	0	1	Păstrarea stării precedente, $C = 1$
9	1	0	0	1	0	1	0	Păstrarea stării precedente, $C = 1$
10	1	0	1	0	1	0	1	Inițializarea bistabilului
11	1	0	1	1	0	0	1	Inițializarea bistabilului
12	1	1	0	0	1	1	0	Înscrierea informației
13	1	1	0	1	0	1	0	Înscrierea informației
14	1	1	1	0	1	1	0	Inversarea stării anterioare/precedente
15	1	1	1	1	0	0	1	Inversarea stării anterioare/precedente

Conform datelor prezentate în tabelul 2.6 bistabilul JK se poate afla în următoarele stări/regimuri de lucru:

- a) $C = 0$ și JK – diferite combinații din 0 și 1, regim de păstrare a informației;
- b) $C = 1$ și $JK = 00$, regim de păstrare a informației;
- c) $C = 1$ și $JK = 01$, inițializarea bistabilului;
- d) $C = 1$ și $JK = 10$, înscrierea informației;
- e) $C = 1$ și $JK = 11$, inversarea stării anterioare/precedente.

Bistabilul JK are următoarele avantaje:

- păstrarea informației precedente în treapta a doua, după înscrierea unei informații noi în prima treaptă, permite organizarea procedurii de restabilire a informației precedente;
- are canal de comandă, este bistabil sincron și poate fi dirijat de program;
- nu are stări interzise.

Bistabilul JK se mai numește bistabil universal deoarece poate înlocui, cu mici modificări, alți bistabili și se utilizează pe larg pentru proiectarea și construirea dispozitivelor numerice.

Simbolul convențional al bistabilului JK este prezentat în fig. 2.16.

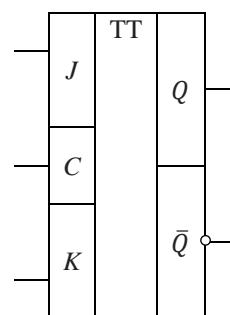


Fig. 2.16. Simbolul convențional al bistabililor JK.

Lucru independent nr. 5.

1. De construit schema bistabilului RS-sincron din elemente logice SAU-NU, de prezentat funcțiile logice și diagramele temporale, de completat tabelul de adevăr.

2. De construit schema electrică a unui divizor al frecvenței cu $K_{div} = 8$ și de prezentat diagramele temporale pentru intrarea C și toate ieșirile Q ale divizorului.

2.4. Registre

2.4.1. Definirea și clasificarea regisrelor

Bistabili permit înscrierea și păstrarea unui sau doi biți de informație. Pentru a înscri și păstra mai mulți biți de informație se folosesc regisrelor.

Definiție. Numim regiszru un dispozitiv electronic numeric construit din bistabili care se utilizează pentru înscrierea, păstrarea, prelucrarea/transformarea și prezentarea unui cuvînt de informație.

Regisrelle pot fi construite din orice tip de bistabili și se clasifică conform următoarelor criterii:

- a) conform modului de înscriere a informației;
- b) conform modului de prezentare a informației;
- c) conform modului de prelucrare a informației.

Conform modului de înscriere a informației există regisrelle de tip paralel, consecutiv și combinat. În regisrelle de tip paralel toți biții unui cuvînt de informație sunt înscrisi simultan sau în același moment de timp, în regisrelle de tip consecutiv biții unui cuvînt de informație sunt înscrisi pe rînd în diferite momente de timp. Regisrelle combine permit înscrierea unui cuvînt de informație în regim de lucru paralel și regim de lucru consecutiv.

Conform modului de prezentare a informației există regisrelle monopolare/monofazate și bipolare/bifazate. Regisrelle monopolare/monofazate pot prezenta informația numai în cod direct sau numai în cod inversat. Regisrelle bipolare/bifazate pot prezenta informația în cod direct și în cod inversat.

Conform modului de prelucrare a informației există regisrelle de deplasare, de transformare și de inversare a informației/codului binar:

a) **Regisrelle de deplasare** permit deplasarea directă sau/și deplasarea inversă a informației în regiszru. Deplasarea directă este o deplasare spre dreapta a biților cuvîntului de informație din bistabilul de ordin superior spre bistabilul de ordin inferior. Deplasarea inversă este o deplasare spre stînga a biților cuvîntului de informație din bistabilul de ordin inferior spre bistabilul de ordin superior.

b) **Regisrelle de transformare** a informației/codului binar pot efectua transformarea codului binar din paralel în consecutiv și/sau din consecutiv în paralel.

c) **Regisrelle de inversare** a informației/codului binar pot inversa informația înscrisă.

Regisrelle au următorii parametri principali:

a) **ordinul** – este determinat de numărul de biți al cuvîntului de informație care poate fi înscris în regiszru sau de numărul de bistabili ai regiszrului;

b) **viteza de lucru** – este măsurată în operații pe secundă și depinde de frecvența maximală posibilă a semnalului de tactare utilizat de regiszru;

c) **timpul de acces** – este un interval de timp din momentul aplicării la intrarea regiszrului a instrucțiunii de afișare a informației pînă în momentul afișării informației pe magistrala de date;

d) **parametrii electrici** – frecvența nominală și maximală a semnalului de tactare, mărimea tensiunii, intensitatea curentului, energia electrică consumată etc.

2.4.2. Regiszru de tip paralel

Drept exemplu, în fig. 2.17 este prezentată schema electrică a regiszrului de tip paralel de ordinul trei, construit din bistabili de tip D.

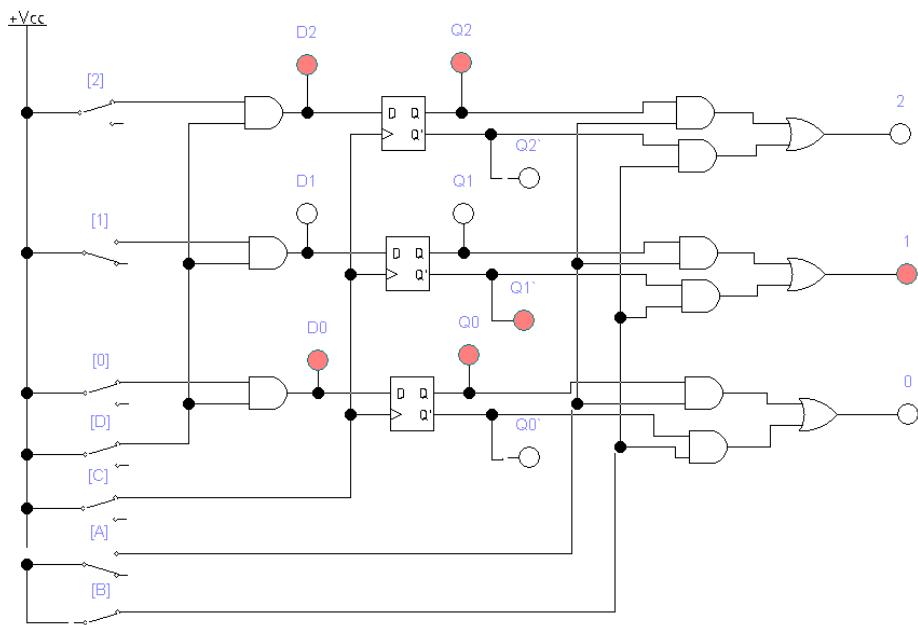


Fig. 2.17. Schema electrică a registrului de tip paralel.

În fig. 2.17 sunt următoarele notări: 2,1,0 – comutatoarele care permit conectarea magistralei de date la intrările elementelor řI și indicatoarele canalelor magistralei de date la ieșirile registrului; D – comutatorul care conectează magistrala de date cu intrările bistabililor; C – comutatorul care permite aplicarea semnalului de comandă pentru înscriverea datelor în registru; A și B – comutatoarele care permit aplicarea semnalelor cu scopul conecterii magistralei de date cu ieșirile registrului.

Schema electrică a registrului de tip paralel prezentat în fig. 2.17 are următoarele intrări și poate îndeplini operațiile:

- 2,1,0 – magistrala de date;
- D = 0 – magistrala de date este deconectată de la intrările registrului D₂D₁D₀;
- D = 1 – magistrala de date este conectată la intrările registrului D₂D₁D₀;
- C = 0 – registrul se află în regim de păstrare a informației;
- C = 1 – are loc înscriverea informației;
- A = 0 și B = 0 – la ieșirile registrului se obține 000 și magistrala de date este deconectată de la ieșirile registrului;
- A = 0 și B = 1 – magistrala de date este conectată la ieșirile registrului sau $Q_2Q_1Q_0 = 010$ (conform schemei din fig. 2.16);
- A = 1 și B = 0 – magistrala de date este conectată la ieșirile registrului sau $Q_2Q_1Q_0 = 101$ (conform schemei din fig. 2.16);
- A = 1 și B = 1 – magistrala de date este conectată la ieșirile registrului și pe magistrală se obține 111 (conform schemei din fig. 2.16).

Registrul are ieșirile:

- $Q_2Q_1Q_0$ – care servesc pentru prezentarea informației în cod paralel binar direct;
- $\bar{Q}_2\bar{Q}_1\bar{Q}_0$ – care servesc pentru prezentarea informației în cod paralel binar invers.

Registrul din în fig. 2.17 poate fi descris conform criteriilor de clasificare prezentate în paragraful 2.4.1:

- conform modului de înscrivere a informației – este un registru de tip paralel;
- conform modului de prezentare a informației – este bipolar deoarece poate prezenta datele în cod direct și în cod invers;
- conform modului de prelucrare a informației – este registru de inversare a datelor (alte operații de prelucrare a datelor nu poate efectua).

Simbolul convențional al registrului de tip paralel este prezentat în fig. 2.18.

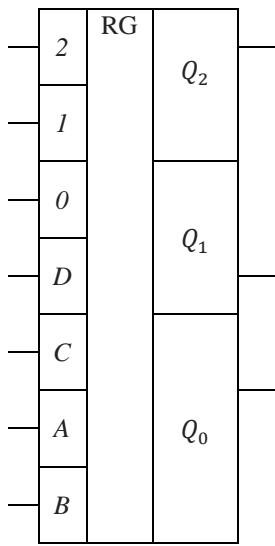


Fig. 2.18. Simbolul convențional al a registrului de tip paralel.

Registrele de tip paralel se utilizează pe larg în dispozitivele calculatoarelor deoarece numărul de biți prelucrați într-o unitate de timp de așa registrul este cu mult mai mare în comparație cu numărul de biți prelucrați într-o unitate de timp de către registrele de tip consecutiv.

2.4.3. Registrul consecutiv cu deplasare directă

Drept exemplu, în fig. 2.19 este prezentată schema electrică a registrului consecutiv cu deplasare directă de ordinul patru, construit din bistabili JK.

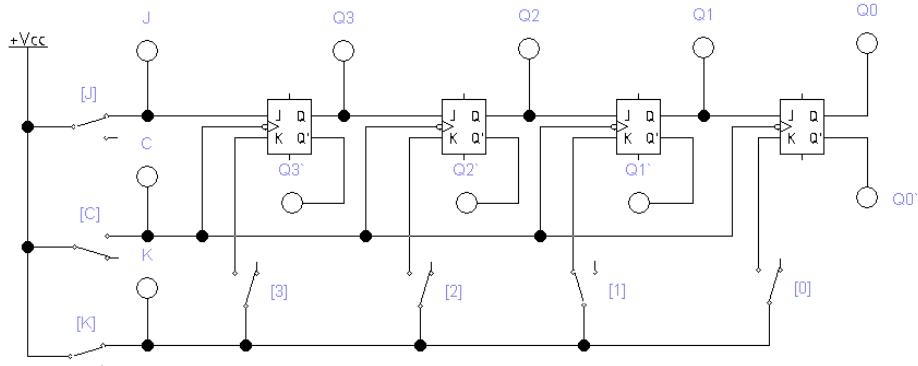


Fig. 2.19. Schema electrică a registrului consecutiv cu deplasare directă.

În fig. 2.19 sunt următoarele notări: J – comutatorul care conectează magistrala de date cu intrarea registrului; C - comutatorul care permite aplicarea semnalului de comandă pentru înscrierea datelor în registru; K și 3,2,1,0 – comutatoarele care permit aplicarea semnalelor cu scopul inițializării registrului.

Schema electrică a registrului de tip consecutiv prezentat în fig. 2.19 are intrările J, C și K și poate îndeplini următoarele operații:

- J = 0 – magistrala de date este deconectată de la intrările registrului;
- J = 1 – magistrala de date este conectată la intrările registrului;
- C = 0 – registrul se află în regim de păstrare a informației pentru orice combinații JK;
- C = 1 și JK = 00 – registrul se află în regim de păstrare a informației;
- C = 1, J = 0 și K = 1 – are loc inițializarea registrului (în registru se înscrive zero);
- C = 1, J = 1 și K = 0 – are loc înscrierea informației (în registru se înscrive unu);

g) $C = 1$, $J = 1$ și $K = 1$ – are loc inversarea stării precedente a registrului.

Registrul are ieșirile:

- Q_0 – pentru prezentarea datelor în cod binar consecutiv direct;
- \bar{Q}_0 – pentru prezentarea datelor în cod binar consecutiv invers;
- $Q_3 Q_2 Q_1 Q_0$ – care servesc pentru prezentarea informației în cod paralel binar direct;
- $\bar{Q}_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0$ – care servesc pentru prezentarea informației în cod paralel binar invers.

Pentru a demonstra, că în registrul consecutiv se realizează deplasarea datelor este utilizată schema electrică și diagramele temporale prezentate în fig. 2.20.

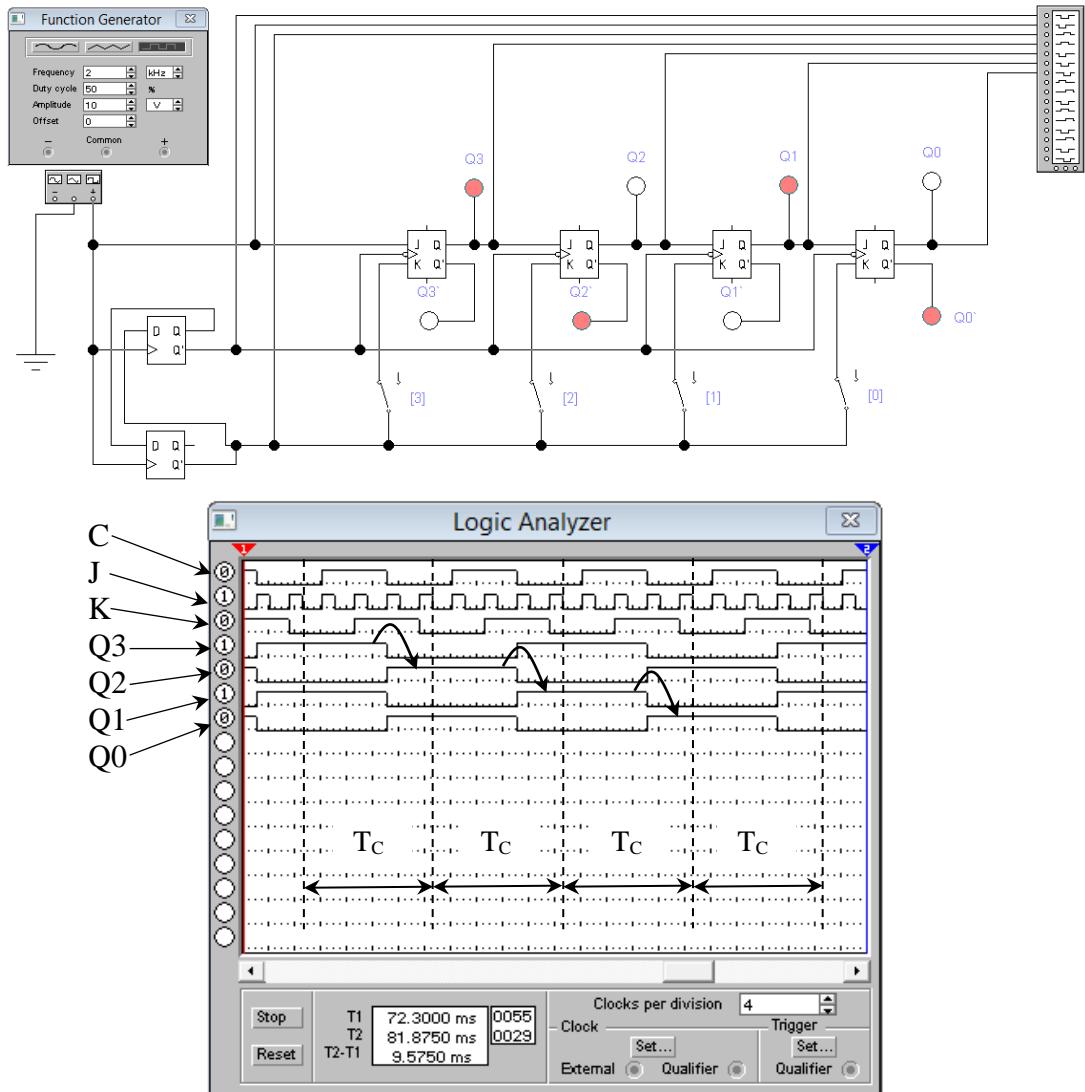


Fig. 2.20. Schema electrică a registrului consecutiv cu deplasare directă și diagramele temporale.

Conform diagramelor temporale prezentate în fig. 2.20 se vede, că în momentul de timp, cînd C trece din starea 1 în starea 0, stările la ieșirile bistabililor se schimbă. Schimbările stărilor bistabililor sunt prezentate în tabelul 2.7. din tabel se vede, că din bistabilul cu număr superior (Q_3) peste fiecare perioadă de timp T_c se deplasează spre dreapta sau se realizează deplasarea directă. În registrele cu deplasare directă înscrierea informației se începe cu bitul inferior.

Tabelul 2.7. Schimbarea stărilor bistabililor ale registrului consecutiv de deplasare directă

Nr. d/o	C	Q_3	Q_2	Q_1	Q_0
0	$1 \rightarrow 0$	1	0	1	0
1	$1 \rightarrow 0$	0	1	0	1
2	$1 \rightarrow 0$	1	0	1	0
3	$1 \rightarrow 0$	0	1	0	1

Registrul din în fig. 2.19 poate fi descris conform criteriilor de clasificare prezentate în paragraful 2.4.1 în modul următor:

- conform modului de înscriere a informației – este un registru de tip consecutiv (serie);
- conform modului de prezentare a informației – este bipolar deoarece poate prezenta datele în cod direct și în cod invers;
- conform modului de prelucrare a informației acest registru poate:
 - efectua deplasarea directă (spre dreapta) a datelor;
 - îndeplini inversarea datelor;
 - transforma datele din cod consecutiv în cod paralel.

Registrele consecutive îndeplinesc mai multe operații în comparație cu registrele de tip paralel însă numărul de biți prelucrați într-o unitate de timp de așa registre este cu mult mai mic.

2.4.4. Registrul consecutiv cu deplasare inversă

Drept exemplu, în fig. 2.21 este prezentată schema electrică a registrului consecutiv cu deplasare inversă de ordinul patru, construit din bistabili JK.

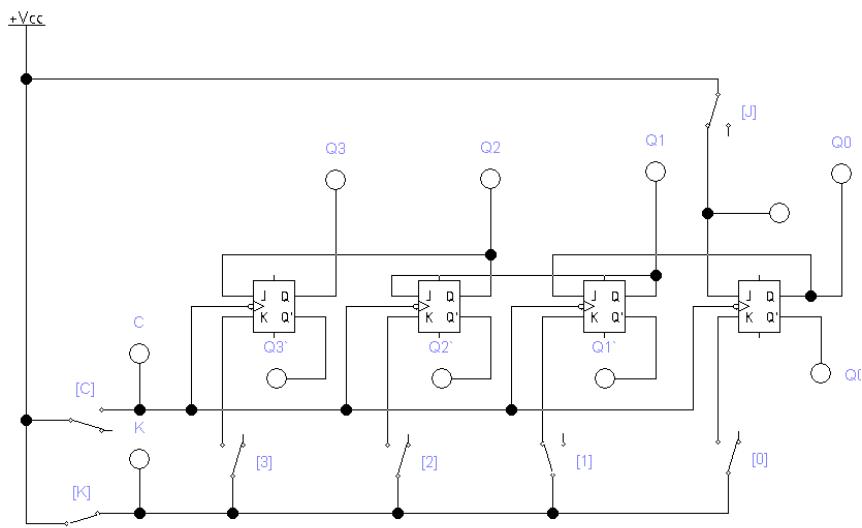


Fig. 2.21. Schema electrică a registrului consecutiv cu deplasare directă.

În fig. 2.21 sunt următoarele notări: J – comutatorul care conectează magistrala de date cu intrarea registrului; C - comutatorul care permite aplicarea semnalului de comandă pentru înscrierea datelor în registrul; K și 3,2,1,0 – comutatoarele care permit aplicarea semnalelor cu scopul inițializării registrului.

Schema electrică a registrului de tip consecutiv prezentat în fig. 2.21 are intrările J, C și K și poate îndeplini următoarele operații:

- $J = 0$ – magistrala de date este deconectată de la intrările registrului;
- $J = 1$ – magistrala de date este conectată la intrările registrului;
- $C = 0$ – registrul se află în regim de păstrare a informației pentru orice combinații JK;
- $C = 1$ și $JK = 00$ – registrul se află în regim de păstrare a informației;
- $C = 1$, $J = 0$ și $K = 1$ – are loc inițializarea registrului (în registru se înscrive zero);
- $C = 1$, $J = 1$ și $K = 0$ – are loc înscrierea informației (în registru se înscrive unu);
- $C = 1$, $J = 1$ și $K = 1$ – are loc inversarea stării precedente a registrului.

Registrul are ieșirile:

- Q_3 – pentru prezentarea datelor în cod binar consecutiv direct;
- \bar{Q}_3 – pentru prezentarea datelor în cod binar consecutiv invers;

- g) $Q_3 Q_2 Q_1 Q_0$ – care servesc pentru prezentarea informației în cod paralel binar direct;
 h) $\bar{Q}_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0$ – care servesc pentru prezentarea informației în cod paralel binar invers.

Pentru a demonstra, că în registrul consecutiv se realizează deplasarea datelor este utilizată schema electrică și diagramele temporale prezentate în fig. 2.22.

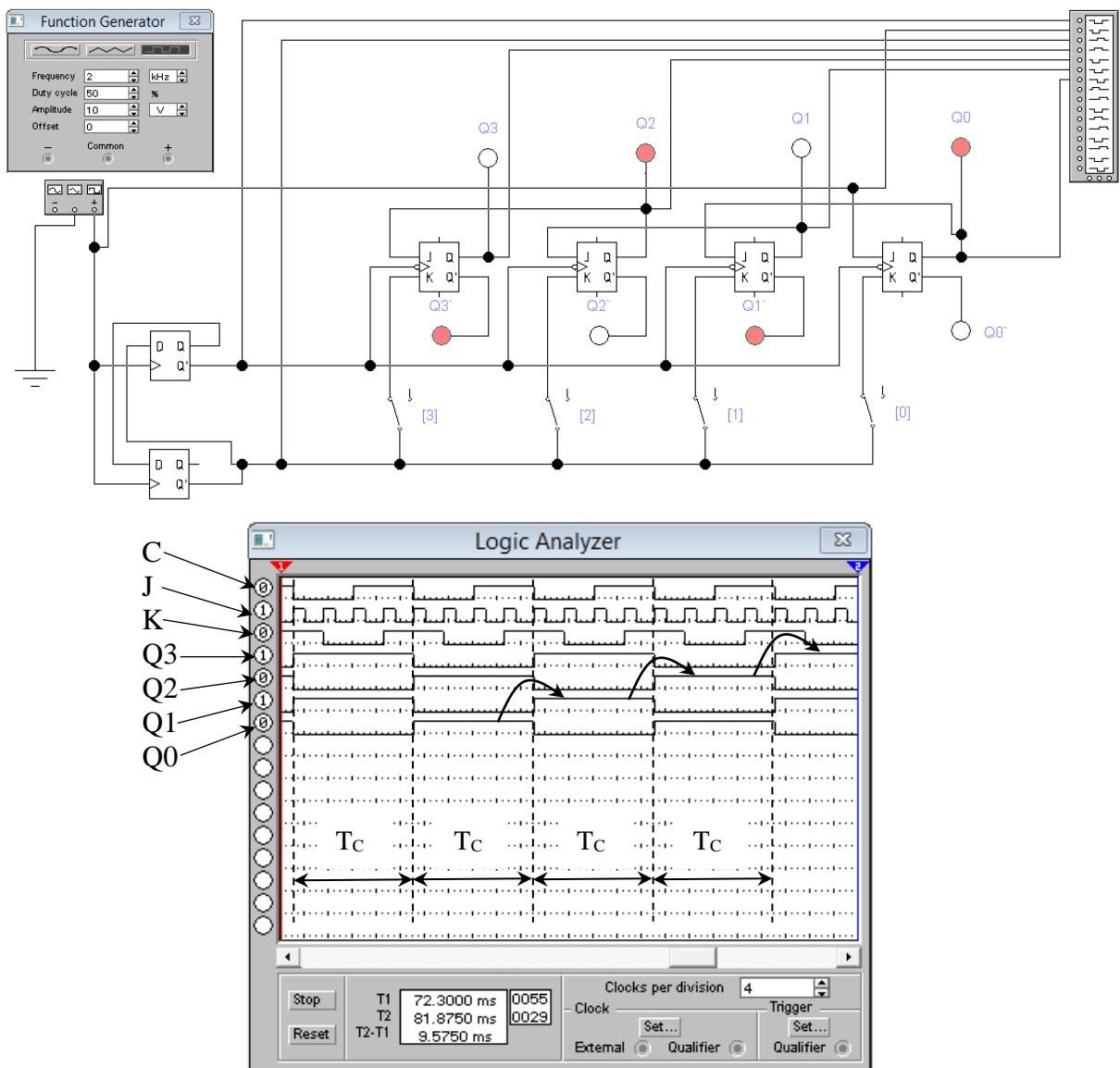


Fig. 2.22. Schema electrică a registrului consecutiv cu deplasare inversă și diagramele temporale.

Conform diagramelor temporale prezentate în fig. 2.18 se vede, că în momentul de timp, cînd C trece din starea 1 în starea 0, stările la ieșirile bistabililor se schimbă. Schimbările stărilor bistabililor sunt prezentate în tabelul 2.8. din tabel se vede, că din bistabilul cu număr inferior (Q0) peste fiecare perioadă de timp T_C se deplasează spre stînga sau se realizează deplasarea inversă. În registrele cu deplasare inversă înscrierea informației se începe cu bitul superior.

Tabelul 2.8. Schimbarea stărilor bistabililor ale registrului consecutiv de deplasare inversă

Nr. d/o	C	Q3	Q2	Q1	Q0
0	$1 \rightarrow 0$	0	1	0	1
1	$1 \rightarrow 0$	1	0	1	0
2	$1 \rightarrow 0$	0	1	0	1
3	$1 \rightarrow 0$	1	0	1	0

Registrul din în fig. 2.21 poate fi descris conform criteriilor de clasificare prezentate în paragraful 2.4.1 în modul următor:

- d) conform modului de înscriere a informației – este un registru de tip consecutiv (serie);
- e) conform modului de prezentare a informației – este bipolar deoarece poate prezenta datele în cod direct și în cod invers;
- f) conform modului de prelucrare a informației acest registru poate:
 - efectua deplasarea inversă (spre stânga) a datelor;
 - îndeplini inversarea datelor;
 - transforma datele din cod consecutiv în cod paralel.

2.4.5. Registrul ciclic

Schema electrică a registrului ciclic poate fi construită pe baza schemelor registrului consecutiv cu deplasare directă sau cu deplasare inversă. În fig. 2.23, drept exemplu, este prezentată schema electrică a registrului ciclic cu deplasare directă de ordinul patru construit din bistabili D.

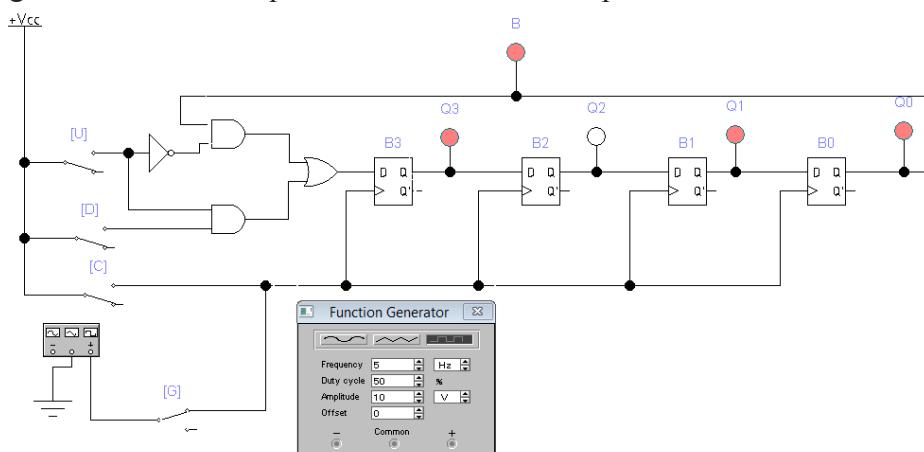


Fig. 2.23. Schema electrică a registrului ciclic cu deplasare directă.

În fig. 2.23 sunt următoarele notări: U – comutatorul care permite conectarea magistralei de date cu intrarea registrului D sau conectarea ieșirii Q0 a registrului cu intrarea D prin elementele řI, SAU; D – comutatorul pentru conectarea magistralei de date cu intrarea D a registrului prin elementele řI, SAU; C – comutatorul care permite aplicarea semnalului de comandă pentru înscrierea datelor în regisztr. În conformitate cu valoarea semnalului U pentru schema din fig. 2.23 se realizează următoarele regimuri de lucru:

- a) U = 1 sau comutatorul U conectat. În acest caz la ieșirea elementului NU se obține 0, la ieșirea elementului řI (de sus) permanent este 0 și ieșirea Q0 a registrului este deconectată de la intrarea D a bistabilului B3. Conectând/deconectând comutatorul D și aplicând semnalul de comandă C informația necesară poate fi înscrisă în regisztr. Semnalul de comandă poate fi generat de generator. În acest caz comutatorul C deconectat, iar comutatorul G conectat.
- b) U = 0 sau comutatorul U deconectat. În acest caz la ieșirea elementului NU se obține 1, la ieșirea elementului řI (de jos) permanent este 0 și magistrala de date este deconectată de la intrarea D a bistabilului B3 iar ieșirea Q0 este conectată de la intrarea D a regisztrului. Aplicând permanent semnalul de comandă C informația, prealabil înscrisă, va fi deplasată în ciclu prin circuitul B3, B2, B1, B0, řI (de sus), SAU, B3.

Schimbarea stărilor bistabililor regisztrului ciclic sunt prezentate în tabelul 2.9 (în tabel X poate fi 0 sau 1).

Tabelul 2.9. Schimbarea stărilor bistabililor registrului ciclic

Nr. d/o	U	C	D	Q3	Q2	Q1	Q0	Comentarii
0	1	0→1	1	1	0	0	0	Înscrierea datelor
1	1	0→1	0	0	1	0	0	
2	1	0→1	1	1	0	1	0	
3	1	0→1	1	1	1	0	1	
4	0	0→1	X	1	1	1	0	
5	0	0→1	X	0	1	1	1	
6	0	0→1	X	1	0	1	1	
7	0	0→1	X	1	1	0	1	

2.4.6. Registr cu intrare paralel-consecutivă

În procesul prelucrării datelor în sistemele numerice (calculator) apare necesitatea executării diferitor operații asupra lor: inversarea; deplasarea directă sau inversă; transformarea din cod consecutiv în cod paralel sau din cod paralel în cod consecutiv. Așa operații pot fi realizate de registrul cu intrare paralel-consecutivă (combinat). Drept exemplu, în fig. 2.24 este prezentată schema electrică a registrului cu intrare paralel-consecutivă de ordinul patru, construit din bistabili de tip D.

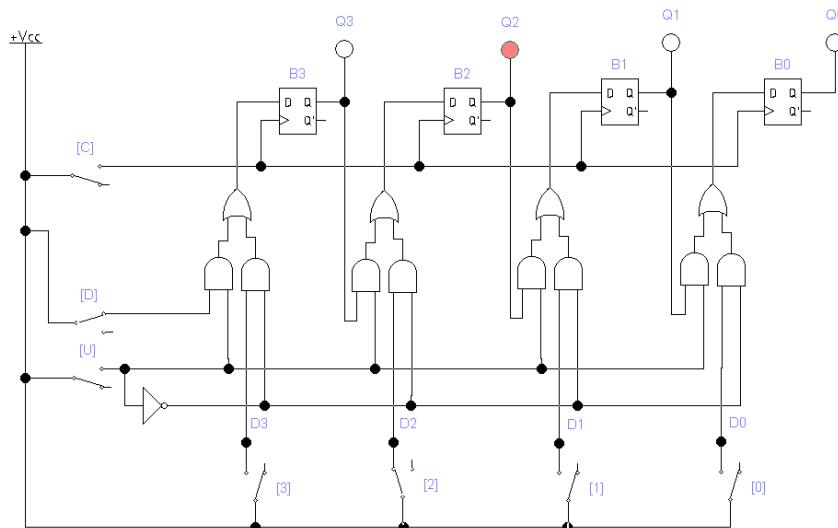


Fig. 2.24. Schema electrică a registrului cu intrare paralel-consecutivă.

În fig. 2.24 sunt următoarele notări:

- D – magistrala consecutivă de date;
- D3, D2, D1, D0 – magistrala paralelă de date;
- C – canalul de comandă;
- 3, 2, 1, 0 – comutatoare care se folosesc pentru a aplica pe magistrala paralelă diferite combinații ale semnalelor;
- U – se utilizează pentru schimbarea regimului de lucru al registrului:
 - U = 1, registrul funcționează în calitate de registru consecutiv;
 - U = 0, registrul funcționează în calitate de registru paralel.

Pentru cazul $U = 1$ (comutatorul U conectat) se obține următoarea configurație a schemei registrului: magistrala de date consecutivă D; elementul řI(din stînga); elementul SAU;

intrarea D a bistabilului B3, ieșirea Q3; elementul řI; elementul SAU; intrarea D a bistabilului B2, ieșirea Q2; elementul řI; elementul SAU; intrarea D a bistabilului B1, ieșirea Q1; elementul řI; elementul SAU; intrarea D a bistabilului B0, ieșirea Q0.

În acest caz registrul funcționează în regim de regisztru consecutiv și execută următoarele operații:

- D = 0 – magistrala consecutivă de date este deconectată de la intrarea registrului;
- D = 1 – magistrala de date consecutivă este conectată la intrările registrului;
- C = 0 – registrul se află în regim de păstrare a informației;
- C = 1, D = 0 – are loc inițializarea registrului (în regisztru se înscrive zero);
- C = 1, D = 1 – are loc înscrierea informației (în regisztru se înscrive unu).

În regim de înscrisire în regisztru are loc deplasarea datelor spre dreapta. Datele pot fi citite în cod consecutiv direct la ieșirea Q0 sau în cod consecutiv invers la ieșirea \bar{Q}_0 .

Pentru cazul $U = 0$ (comutatorul U deconectat) conexiunile dintre bistabili se deconectează, magistrala paralelă de date este conectată la intrările bistabililor și cu ajutorul comutatoarelor (3, 2, 1, 0) pot fi aplicate diferite combinații ale datelor. În acest caz se obține următoarea configurație a schemei registrului:

- canalul D3; elementul řI(din dreapta); elementul SAU; intrarea D a bistabilului B3, ieșirea Q3;
- canalul D2; elementul řI(din dreapta); elementul SAU; intrarea D a bistabilului B2, ieșirea Q2;
- canalul D1; elementul řI(din dreapta); elementul SAU; intrarea D a bistabilului B1, ieșirea Q1;
- canalul D0; elementul řI(din dreapta); elementul SAU; intrarea D a bistabilului B0, ieșirea Q0.

Datele pot fi citite în cod paralel direct la ieșirile $Q_3 Q_2 Q_1 Q_0$ sau în cod paralel invers la ieșirile $\bar{Q}_3 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0$.

Registrul combinat din fig. 2.24 poate fi descris conform criteriilor de clasificare prezentate în paragraful 2.4.1 în modul următor:

- conform modului de înscrisire a informației – este un regisztru de tip combinat: $U = 1$, regisztrul funcționează în calitate de regisztru consecutiv; $U = 0$, regisztrul funcționează în calitate de regisztru paralel;
- conform modului de prezentare a informației – este bipolar deoarece poate prezenta datele în cod direct și în cod invers;
- conform modului de prelucrare a informației acest regisztru poate:
 - efectua deplasarea directă (spre dreapta) a datelor;
 - îndeplini inversarea datelor;
 - transforma datele din cod consecutiv în cod paralel;
 - transforma datele din cod paralel în cod consecutiv.

2.4.7. Regisztru universal

În calculatoare registrele au sarcina să execute mai multe operații în comparație cu numărul de operații executat de regisztrul combinat. Pentru a majora numărul de operații/funcții ale regisztrului se folosesc scheme electrice speciale care permit comutarea mai multor regimuri de lucru. Registrele care pot funcționa în mai multe regimuri se mai numesc regisztre universale. Drept exemplu, în fig. 2.25 este prezentată schema electrică a regisztrului universal de ordinul patru, construit din bistabili de tip D.

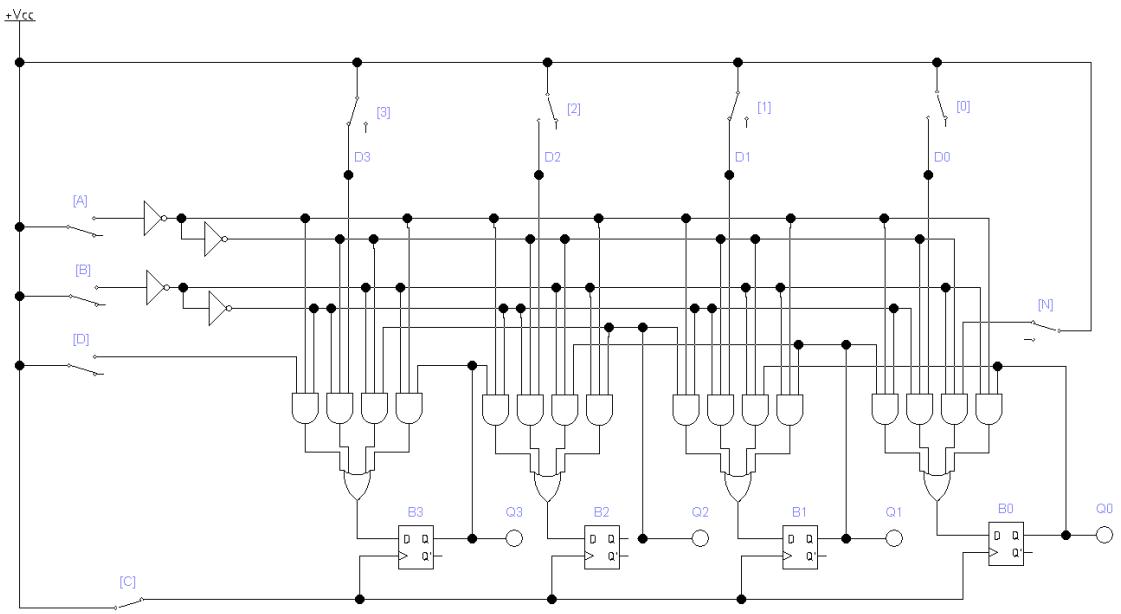


Fig. 2.25. Schema electrică a registrului universal.

În fig. 2.25 sunt următoarele notări:

- A, B – comutatoare utilizate pentru schimbarea regimurilor de lucru ale registrului;
- D – comutatorul care conectează magistrala consecutivă de date la intrarea D a bistabilului B3 în cazul funcționării registrului în regim de registru cu deplasare directă;
- N – comutatorul care conectează magistrala consecutivă de date la intrarea D a bistabilului B0 în cazul funcționării registrului în regim de registru cu deplasare inversă;
- 3, 2, 1 și 0 – comutatoarele utilizate pentru conectarea magistralei paralele de date cu intrările registrului D3, D2, D1 și D0;
- C – canalul de comandă.

În continuare v-ori fi prezentate regimurile de lucru ale registrului universal în conformitate cu stările A și B.

Fie $A = 0$ și $B = 1$. În acest caz registrul funcționează în regim de registru consecutiv cu deplasare directă. Conform schemei electrice din fig. 2.25 pentru registrul consecutiv cu deplasare directă se formează un circuit electric cu următoarea configurație (este funcțional numai primul element řI din toate grupele): magistrala de date consecutivă D; elementul řI; elementul SAU; intrarea D a bistabilului B3, ieșirea Q3; elementul řI; elementul SAU; intrarea D a bistabilului B2, ieșirea Q2; elementul řI; elementul SAU; intrarea D a bistabilului B1, ieșirea Q1; elementul řI; elementul SAU; intrarea D a bistabilului B0, ieșirea Q0.

Operațiile îndeplinite de registrul consecutiv cu deplasare directă sunt descrise în paragraful 2.4.3. Datele pot fi citite în cod consecutiv direct la ieșirea Q0 sau în cod consecutiv invers la ieșirea \bar{Q}_0 .

Fie $A = 1$ și $B = 1$. În acest caz registrul funcționează în regim de registru paralel. Conform schemei electrice din fig. 2.25 pentru registrul paralel se formează un circuit electric cu următoarea configurație (este funcțional numai al doilea element řI din toate grupele):

- canalul D3; elementul řI; elementul SAU; intrarea D a bistabilului B3, ieșirea Q3;
- canalul D2; elementul řI; elementul SAU; intrarea D a bistabilului B2, ieșirea Q2;
- canalul D1; elementul řI; elementul SAU; intrarea D a bistabilului B1, ieșirea Q1;
- canalul D0; elementul řI; elementul SAU; intrarea D a bistabilului B0, ieșirea Q0.

Datele pot fi citite în cod paralel direct la ieșirile $Q_3Q_2Q_1Q_0$ sau în cod paralel invers la ieșirile $\bar{Q}_3\bar{Q}_2\bar{Q}_1\bar{Q}_0$.

Fie $A = 1$ și $B = 0$. În acest caz registrul funcționează în regim de registru consecutiv cu deplasare inversă. Conform schemei electrice din fig. 2.25 pentru registrul consecutiv cu deplasare directă se formează un circuit electric cu următoarea configurație (este funcțional numai al treilea element řI din toate grupele): magistrala de date consecutivă N; elementul řI; elementul SAU; intrarea D a bistabilului B0, ieșirea Q0; elementul řI; elementul SAU; intrarea D a bistabilului B1, ieșirea Q1; elementul řI; elementul SAU; intrarea D a bistabilului B2, ieșirea Q2; elementul řI; elementul SAU; intrarea D a bistabilului B3, ieșirea Q3.

Operațiile îndeplinite de registrul consecutiv cu deplasare inversă sunt descrise în paragraful 2.4.4. Datele pot fi citite în cod consecutiv direct la ieșirea Q3 sau în cod consecutiv invers la ieșirea \bar{Q}_3 .

Fie $A = 0$ și $B = 0$. În acest caz registrul funcționează în regim de regenerare/reînoire a datelor. Conform schemei electrice din fig. 2.25 pentru acest regim de lucru al registrului se formează un circuit electric cu următoarea configurație (este funcțional numai elementul řI cu numărul patru din toate grupele):

- ieșirea Q3; elementul řI; elementul SAU; intrarea D a bistabilului B3, ieșirea Q3;
- ieșirea Q2; elementul řI; elementul SAU; intrarea D a bistabilului B2, ieșirea Q2;
- ieșirea Q1; elementul řI; elementul SAU; intrarea D a bistabilului B1, ieșirea Q1;
- ieșirea Q0; elementul řI; elementul SAU; intrarea D a bistabilului B0, ieșirea Q0.

Așa regim de lucru este frecvent utilizat în microcircuitele de memorie DRAM care pot păstra datele numai un interval de timp limitat (10...20 ms).

2.5. Contoare

2.5.1. Definirea și clasificarea contoarelor

Definiție. Numim **contor** un dispozitiv electronic la ieșirile căruia se obține un cod binar, determinat de numărul de impulsuri aplicate la intrare.

Contoarele pot fi clasificate conform:

- direcției de numărare** – sumative (de adunare), de numerare inversă (de scădere) și reversibile;
- modului de numărare** – binare, binar-zecimale, cu modul de numărare constant și cu modul de numărare arbitrar;
- metodei de transfer a datelor dintre bistabili** – cu transfer consecutiv, cu transfer paralel accelerat, cu transfer combinat și cu transfer în ciclu;
- procesului de schimbare a stărilor bistabililor** – sincroni și asincroni.

Contoarele au următorii parametri principali:

- modulul de numărare – este determinat de numărul de impulsuri N_{max} pe care le poate număra contorul;
- viteza de lucru – este măsurată în operații pe secundă și depinde de frecvența maximală posibilă a semnalului de tactare utilizat de regisztru;
- timpul de reacție – este un interval de timp din momentul aplicării impulsului la intrarea contorului pînă în momentul instalării stărilor stable pe toate ieșirile contorului ($\Delta t_{reac} = 1/f_{max}$);
- parametrii electrici – mărimea tensiunii, intensitatea curentului, energia electrică consumată etc.

2.5.2. Contor asincron sumativ

În fig. 2.26 este prezentată schema electrică a contorului asincron sumativ de ordinul trei, construit din bistabili de tip D cu circuit cu reacție (ieșirile \bar{Q} ale bistabililor B0 și B1 sunt conectate cu intrările următorilor bistabili).

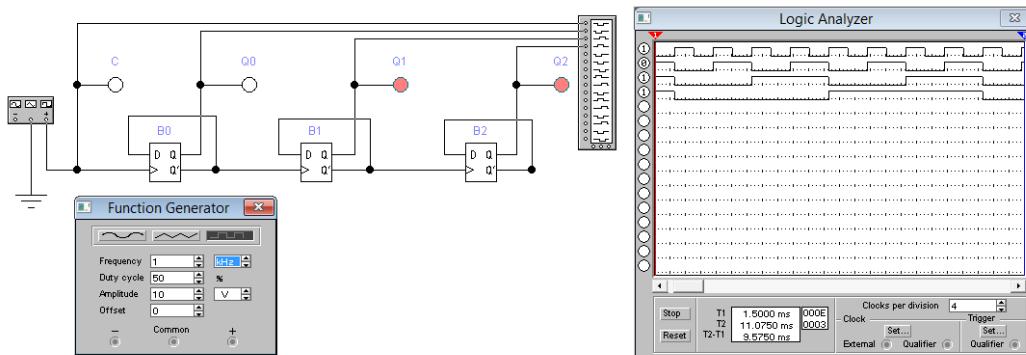


Fig. 2.26. Schema electrică a contorului sumativ de ordinul trei.

În fig. 2.26 sunt următoarele notări:

- C – canalul prin care se aplică semnalul de tactare;
- $Q_2 Q_1 Q_0$ – ieșirile contorului (magistrala de date) pentru prezentarea datelor în cod binar direct.

Este necesar să fie menționat, că numerarea bistabililor în schema de tip consecutiv a contorului începe cu bitul inferior, iar pentru registrele de tip consecutiv – cu bitul superior.

Principiul de lucru al contorului. Stările bistabililor se schimbă după frontul anterior al impulsului aplicat la intrările lor în momentul schimbării valorii impulsului din 0 în 1. Conform schemei electrice stările bistabilului B0 se schimbă prin aplicarea impulsurilor de la generator,

stările bistabilului B1 se schimbă cu o frecvență de 2 ori mai mică prin aplicarea impulsurilor de la ieșirea \bar{Q} a bistabilului B0, iar stările bistabilului B2 se schimbă cu o frecvență de 4 ori mai mică prin aplicarea impulsurilor de la ieșirea \bar{Q} a bistabilului B1 (vezi diagramele temporale din fig. 2.26).

Fie timpul de propagare a semnalului de la intrarea C pînă la ieșirea \bar{Q}_0 a bistabilului B0 este Δt_0 , iar timpul de propagare a semnalului prin bistabilul B1 este Δt_1 . Din aceasta rezultă, că frontul anterior al semnalului \bar{Q}_0 nu coincide în timp cu frontul anterior al semnalului C sau întîrzie cu Δt_0 și prin urmare bistabilul B1 își schimbă starea peste un interval de timp Δt_0 . Respectiv frontul anterior al semnalului \bar{Q}_1 nu coincide în timp cu frontul anterior al semnalului C sau întîrzie cu $\Delta t_0 + \Delta t_1$ și prin urmare bistabilul B2 își schimbă starea peste un interval de timp $\Delta t_0 + \Delta t_1$. Această demonstrează că bistabilii unui contor de tip consecutiv, fiind comandanți de semnale diferite, un pot să-și schimbe stările sincron cu semnalul de tactare aplicat la intrarea C a bistabilului B0. Din acest motiv aşa contoare se numesc contoare asincrone.

Conform diagramelor temporale din fig. 2.26 este completat tabelul de adevăr al contorului sumativ (vezi tabelul 2.10). Stările $Q_2 Q_1 Q_0$ sunt luate după fiecare impuls de tactare C.

Tabelul 2.10. Tabelul de adevăr al contorului sumativ de ordinul trei

Nr. d/o	C	Q_2	Q_1	Q_0	\bar{Q}_2	\bar{Q}_1	\bar{Q}_0
0	0→1	0	0	0	1	1	1
1	0→1	0	0	1	1	1	0
2	0→1	0	1	0	1	0	1
3	0→1	0	1	1	1	0	0
4	0→1	1	0	0	0	1	1
5	0→1	1	0	1	0	1	0
6	0→1	1	1	0	0	0	1
7	0→1	1	1	1	0	0	0
8	0→1	0	0	0	1	1	1

Din tabelul de adevăr se vede, că numărul maximal prezentat de contor este $N_{max} = 111_2$ și la ieșirile $\bar{Q}_2 \bar{Q}_1 \bar{Q}_0$ ale contorului se realizează numărarea inversă.

Contoarele asincrone cu N_{max} de ordin mare au viteza de lucru relativă mică. Unicul avantaj al contoarelor asincrone constă în simplitatea schemei electrice.

2.5.3. Contor asincron cu numărare inversă

În fig. 2.27 este prezentată schema electrică a contorului asincron cu numărare inversă de ordinul trei, construit din bistabili de tip D cu circuit cu reacție (ieșirile Q ale bistabililor B0 și B1 sunt conectate cu intrările următorilor bistabili).

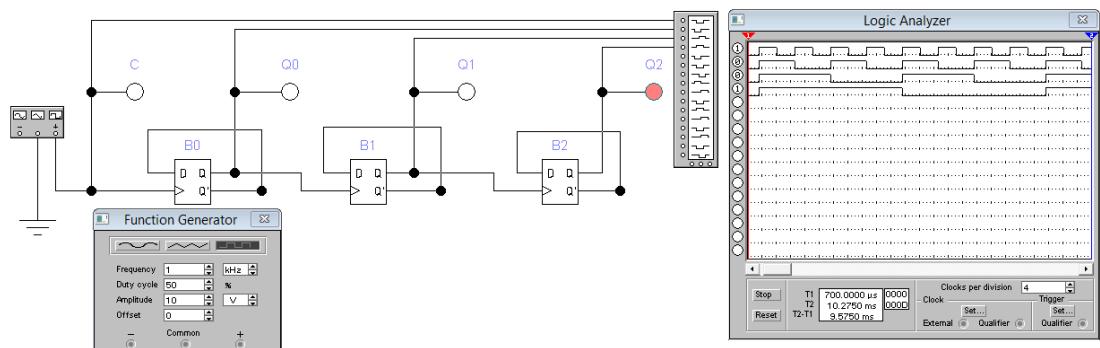


Fig. 2.27. Schema electrică a contorului sumativ de ordinul trei.

În fig. 2.27 sunt următoarele notări:

- c) C – canalul prin care se aplică semnalul de tactare;
- d) $Q_2 Q_1 Q_0$ – ieșirile contorului (magistrala de date) pentru prezentarea datelor în cod binar direct.

Principiul de lucru al contorului. Stările bistabililor se schimbă după frontul anterior al impulsului aplicat la intrările lor în momentul schimbării valorii impulsului din 0 în 1. Conform schemei electrice stările bistabilului B0 se schimbă prin aplicarea impulsurilor de la generator, stările bistabilului B1 se schimbă cu o frecvență de 2 ori mai mică prin aplicarea impulsurilor de la ieșirea Q a bistabilului B1, iar stările bistabilului B2 se schimbă cu o frecvență de 4 ori mai mică prin aplicarea impulsurilor de la ieșirea Q a bistabilului B1 (vezi diagramele temporale din fig. 2.27).

Conform diagramelor temporale din fig. 2.27 este completat tabelul de adevăr al contorului cu numărare inversă (vezi tabelul 2.11). Stările $Q_2 Q_1 Q_0$ sunt luate după fiecare impuls de tactare C.

Tabelul 2.11. Tabelul de adevăr al contorului cu numărare inversă de ordinul trei

Nr. d/o	C	Q_2	Q_1	Q_0	\bar{Q}_2	\bar{Q}_1	\bar{Q}_0
0	0→1	1	1	1	0	0	0
1	0→1	1	1	0	0	0	1
2	0→1	1	0	1	0	1	0
3	0→1	1	0	0	0	1	1
4	0→1	0	1	1	1	0	0
5	0→1	0	1	0	1	0	1
6	0→1	0	0	1	1	1	0
7	0→1	0	0	0	1	1	1
8	0→1	1	1	1	0	0	0

Din tabelul de adevăr se vede, că numărul maximal prezentat de contor este $N_{\max} = 111_2$ și la ieșirile $\bar{Q}_2 \bar{Q}_1 \bar{Q}_0$ ale contorului se realizează numărarea directă (sumativă).

2.5.4. Contor asincron reversibil

În fig. 2.28 este prezentată schema electrică a contorului asincron reversibil de ordinul trei, construit din bistabili de tip D cu circuit cu reacție.

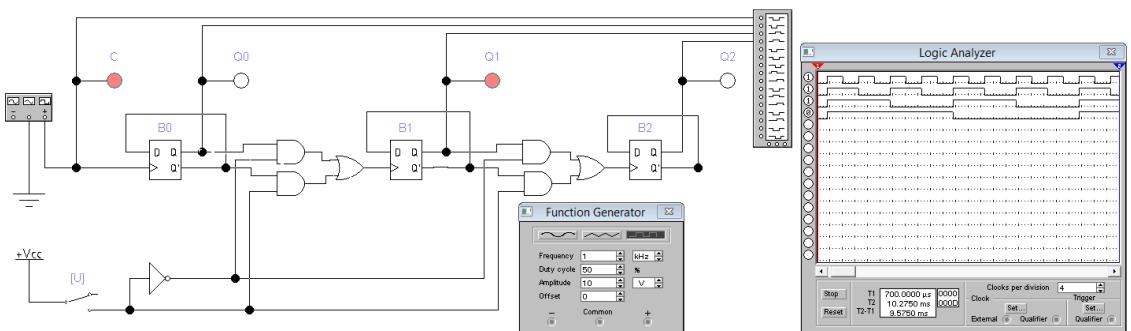


Fig. 2.28. Schema electrică a contorului reversiv de ordinul trei.

În schema electrică, prezentată în fig. 2.28, ieșirile Q , \bar{Q} ale bistabililor B0 și B1 sunt conectate cu intrările bistabililor B1 și B2 cu ajutorul porturilor construite din un element NU,

două elemente řI și un element SAU. Comutatorul U se utilizează pentru schimbarea regimului de lucru al contorului.

Fie $U = 1$ (comutatorul conectat). În acest caz ieširile \bar{Q} ale bistabililor B0 și B1 sunt conectate cu intrările bistabililor B1 și B2. Contorul funcționează în regim de contor asincron sumativ (vezi paragraful 2.5.2).

Fie $U = 0$ (comutatorul deconectat). În acest caz ieširile Q ale bistabililor B0 și B1 sunt conectate cu intrările bistabililor B1 și B2. Contorul funcționează în regim de contor asincron cu numărare inversă (vezi paragraful 2.5.3).

2.5.5. Contor asincron cu modul de numărare

În fig. 2.29 este prezentată schema electrică a contorului asincron cu modul de numărare variabil $M < N_{\max}$ (pentru acest contor $N_{\max} = 1111_2$) de ordinul patru, construit din bistabili de tip D cu circuit cu reacție. Valoarea modulului de numărare M depinde de stările comutatoarelor 3, 2, 1 și 0 (conectate/deconectate).

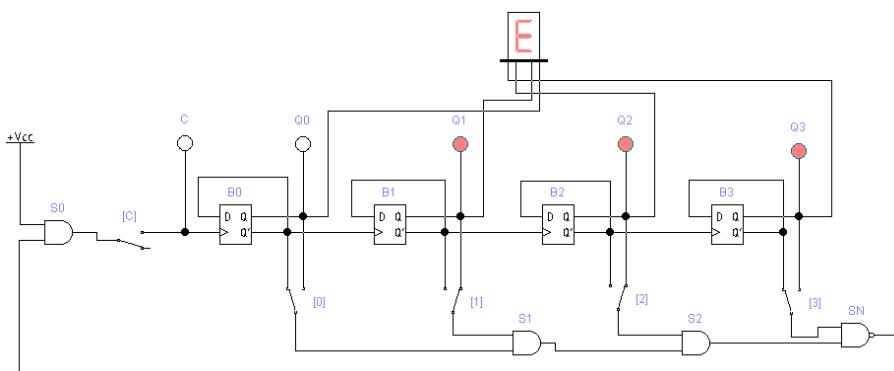


Fig. 2.29. Schema electrică a contorului asincron cu modul de numărare M.

În fig. 2.29 sunt următoarele notări:

- C – comutatorul utilizat pentru aplicarea impulsurilor la intrarea contorului;
- 3, 2, 1 și 0 – comutatoarele utilizate pentru conectarea ieširilor Q , \bar{Q} ale bistabililor B0, B1, B2, și B3 la intrările elementelor logice S1(SI), S2(SI), SN(SI-NU);
- S0 – elementul řI care transmite la intrarea C a contorului semnalul obtinut în rezultatul înmulțirii logice a semnalelor $+V_{cc}$ și SN:
 - dacă $SN = 1$, atunci $+V_{cc} \times SN = +V_{cc}$ și la intrarea contorului se aplică impulsul sau contorul este în regim de numărare;
 - dacă $SN = 0$, atunci $+V_{cc} \times SN = 0$ și la intrarea contorului impulsul nu se aplică sau contorul este în regim de pauză;
- $Q_3 Q_2 Q_1 Q_0$ – ieširile contorului.

Pentru circuitul format din elementele logice S1(SI), S2(SI), SN(SI-NU) este valabilă următoarea formulă

$$\begin{aligned} SN &= S2 \times (Q_3 \text{ sau } \bar{Q}_3) = S1 \times (Q_2 \text{ sau } \bar{Q}_2) \times (Q_3 \text{ sau } \bar{Q}_3) = \\ &= (Q_0 \text{ sau } \bar{Q}_0) \times (Q_1 \text{ sau } \bar{Q}_1) \times (Q_2 \text{ sau } \bar{Q}_2) \times (Q_3 \text{ sau } \bar{Q}_3) \end{aligned}$$

Valoarea logică a semnalului SN depinde de valoarea modulului M și respectiv de conectarea comutatoarelor 0, 1, 2, 3 la ieširile Q_i sau \bar{Q}_i ale contorului, unde $i = 0, 1, 2, 3$. Pentru funcționarea contorului în regim de numărare este necesar ca valoarea logică SN să fie 1 pînă în momentul de timp, cînd la ieširile contorului se instalează numărul M .

Drept exemplu, $M = 1011_2 = 11_{10}$. În acest caz pentru SN se obține următoarea formulă

$$SN = (Q_0) \times (Q_1) \times (\bar{Q}_2) \times (Q_3).$$

Conform acestei formule comutatoarele 3, 2, 1 și 0 trebuie conectate la ieșirile Q_3 , \bar{Q}_2 , Q_1 și Q_0 respectiv.

În conformitate cu mărimea modului M și configurarea conectărilor între ieșirile contorului și comutatoarele 3, 2, 1 și 0 schema prezentată în fig. 2.29 poate fi numită:

- a) $M = 1001_2 = 9_{10}$ – contor binar-zecimal (poate număra pînă la 9);
- b) M fixat, unde $M < N_{max}$ și $M \neq 1001_2$ – contor cu modul de numărare constant;
- c) valoarea modulului $M < N_{max}$ poate primi diferite valori conform programului care poate schimba stările comutatoarelor 3, 2, 1 și 0 – contor cu modul de numărare arbitrar.

2.5.6. Contor sincron

În compartimentul 2.5.2 a fost menționat faptul, că contoarele asincrone cu N_{max} de ordin mare au viteză de lucru relativ mică. În scopul majorării vitezei de lucru a contoarelor au fost elaborate contoare sincrone. Drept exemplu, în fig. 2.30 este prezentată schema electrică a contorului sincron, construit din bistabili de tip JK.

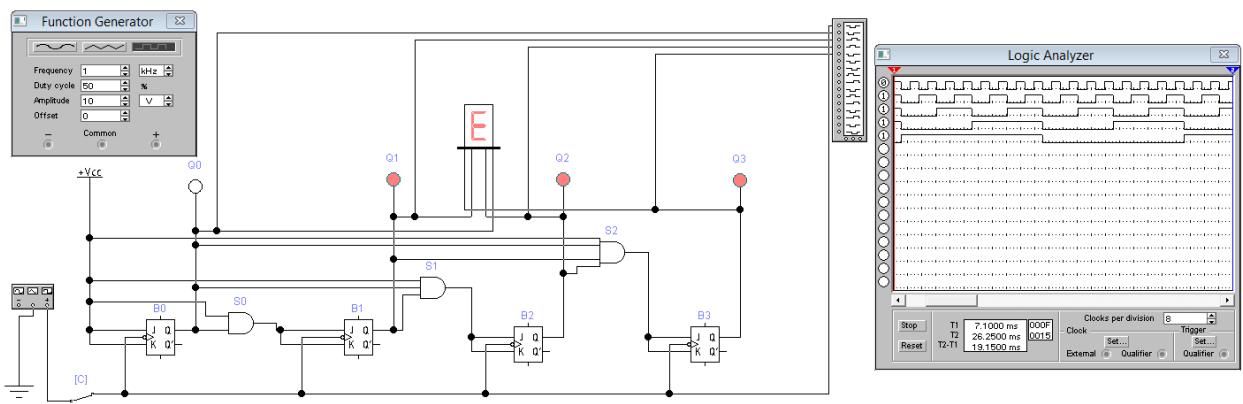


Fig. 2.30. Schema electrică a contorului sincron.

În schema din fig. 2.30 toți bistabilii își schimbă starea în același moment de timp, deoarece semnalul de tactare C, emis de generator, este aplicat simultan la toate intrările C ale bistabililor. Permișuna de schimbare a stării bistabililor B3, B2, B1 este formată de elementele logice řI (S2 = 1, S1 = 1, S0 = 1).

Drept exemplu, bistabilul B3 își va schimba starea la ieșirile Q_3 , \bar{Q}_3 după frontul posterior al semnalului de tactare C ($1 \rightarrow 0$), numai în cazul cînd $S2 = S1 \times S0 \times (+V_{cc}) = 1$.

3. DISPOZITIVE COMBINATORIALE

În calculator există un set de dispozitive care execută operații cu datele fără memorarea lor. Din astăzi operații pot fi menționate următoarele:

- a. transferul informației din mai multe canale într-un canal și invers;
 - b. codificarea și decodificarea informației;
 - c. executarea operațiilor aritmetice,
 - d. executarea operațiilor logice.

Pentru executarea acestor operații se folosesc dispozitive electronice numerice numite dispozitive combinataionale.

Definiție. Numim dispozitiv combinational un dispozitiv electronic numeric utilizat pentru prelucrarea informației fără memorizarea ei.

Din dispozitivele combinationale pot fi numite următoarele:

- decoderul;
 - coderul;
 - translatorul de cod;
 - multiplexorul;
 - demultiplexor;
 - comparatorul;
 - unitatile aritmetico-logice

Dispozitivele combinaționale se deosebesc de dispozitivele care pot păstra informația (bistabili, registre, contoare) prin următoarele proprietăți:

- a) nu pot păstra informația sau nu au memorie;
 - b) starea următoare nu depinde de starea precedentă;
 - c) starea la iesirile dispozitivului depinde univoc de starea aplicată la intrările lui.

3.1. Decoder

Definiție. Numim decoder un dispozitiv electronic combinațional care are k intrări și $m=2^k$ ieșiri și este utilizat pentru transformarea unui cod binar aplicat la intrările în semnal electric (logic 1 sau 0) numai la una din ieșiri.

Deosebim decoder complet, dacă $m=2^k$, și decoder necomplet, dacă $m < 2^k$.

Fie, drept exemplu, este dat tabelul de adevăr al unui decoder complet pentru care $k = 3$ și $m = 8$ (vezi tabelul 3.1). Tabelul 3.1 este completat conform definitiei decoderului.

Tabelul 3.1. Tabelul de adevăr al unui decoder complet pentru care $k = 3$ și $m = 8$

3. DISPOZITIVE COMBINATORALE

În calculator există un set de dispozitive care execută operații cu datele fără memorarea lor. Din astăzi operații pot fi menționate următoarele:

- a. transferul informației din mai multe canale într-un canal și invers;
 - b. codificarea și decodificarea informației;
 - c. executarea operațiilor aritmetice,
 - d. executarea operațiilor logice.

Pentru executarea acestor operații se folosesc dispozitive electronice numerice numite dispozitive combinataionale.

Definiție. Numim dispozitiv combinational un dispozitiv electronic numeric utilizat pentru prelucrarea informației fără memorizarea ei.

Din dispozitivele combinaționale pot fi numite următoarele:

- decoderul;
 - coderul;
 - translatorul de cod;
 - multiplexorul;
 - demultiplexor;
 - comparatorul;
 - unitatile aritmetic-hec;

Dispozitivele combinaționale se deosebesc de dispozitivele care pot păstra informația (bistabili, registre, contoare) prin următoarele proprietăți:

- a) nu pot păstra informația sau nu au memorie;
 - b) starea următoare nu depinde de starea precedentă;
 - c) starea la iesirile dispozitivului depinde univoc de starea aplicată la intrările lui.

3.1. Decoder

Definiție. Numim decoder un dispozitiv electronic combinațional care are k intrări și $m=2^k$ ieșiri și este utilizat pentru transformarea unui cod binar aplicat la intrările în semnal electric (logic 1 sau 0) numai la una din ieșiri.

Deosebim decoder complet, dacă $m=2^k$, și decoder necomplet, dacă $m < 2^k$.

Fie, drept exemplu, este dat tabelul de adevăr al unui decoder complet pentru care $k = 3$ și $m = 8$ (vezi tabelul 3.1). Tabelul 3.1 este completat conform definitiei decoderului.

Tabelul 3.1. Tabelul de adevăr al unui decoder complet pentru care $k = 3$ și $m = 8$

Din tabelul 3.1 de adevăr se obțin FCND ale următoarelor funcții logice:

$$F_0(x_2, x_1, x_0) = \bar{x}_2 \times \bar{x}_1 \times \bar{x}_0; \quad (3.1)$$

$$F_1(x_2, x_1, x_0) = \bar{x}_2 \times \bar{x}_1 \times x_0; \quad (3.2)$$

$$F_2(x_2, x_1, x_0) = \bar{x}_2 \times x_1 \times \bar{x}_0; \quad (3.3)$$

$$F_3(x_2, x_1, x_0) = \bar{x}_2 \times x_1 \times x_0; \quad (3.4)$$

$$F_4(x_2, x_1, x_0) = x_2 \times \bar{x}_1 \times \bar{x}_0; \quad (3.5)$$

$$F_5(x_2, x_1, x_0) = x_2 \times \bar{x}_1 \times x_0; \quad (3.6)$$

$$F_6(x_2, x_1, x_0) = x_2 \times x_1 \times \bar{x}_0; \quad (3.7)$$

$$F_7(x_2, x_1, x_0) = x_2 \times x_1 \times x_0. \quad (3.8)$$

În fig. 3.1 este prezentată schema electrică a decoderului, construită conform funcțiilor logice (3.1)...(3.8). În fig. 3.1 sunt următoarele notări: $+V_{cc}$ – sursa de semnal; A, B, C – comutatoarele magistralei de date; F7, ..., F0 – ieșirile decoderului. Din schemă se vede, că la ieșirea F5 se obține semnal electric, dacă la intrări se aplică codul ABC = 101₂.

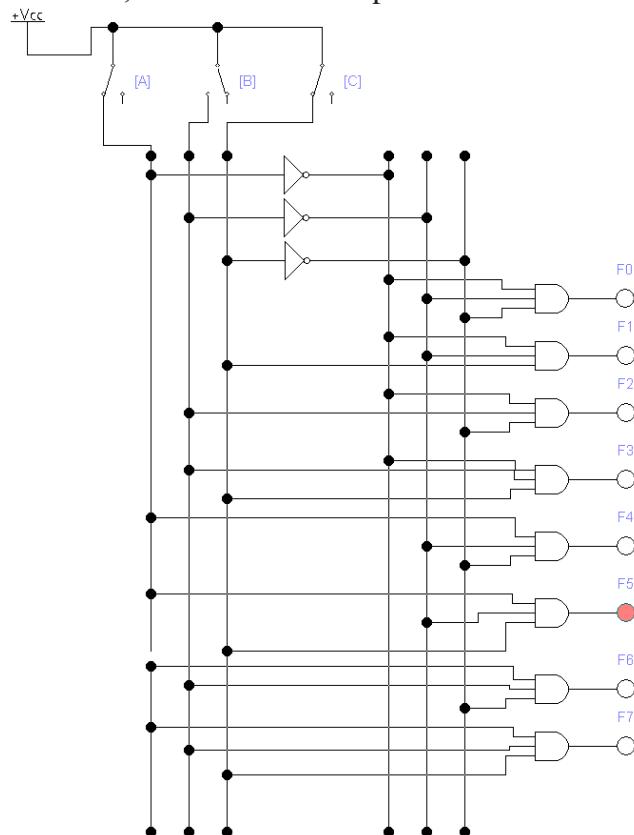


Fig. 3.1. Schema electrică a decoderului.

Simbolul convențional al decoderului este prezentat în figura 3.2.

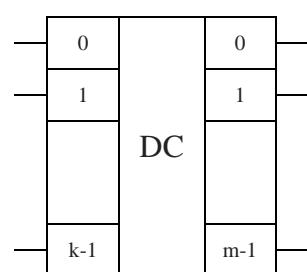


Fig. 3.2. Simbolul convențional al decoderului.

Decoderul este utilizat, în special, pentru a transforma codul binar în impuls de comandă. Impulsurile de comandă se folosesc pentru dirijarea stărilor de lucru ale dispozitivelor calculatorului. Decoderul este parte componentă a unității de comandă a microprocesorului, al memoriei permanente și convertorului numeric-analogic (CNA).

3.2. Coder

Definiție. Numim coder un dispozitiv electronic combinațional care are $k = 2^m$ intrări și m ieșiri și este utilizat pentru transformarea semnalului electric (logic 1 sau 0) aplicat la una din intrări în cod binar prezentat la ieșiri.

Deosebim coder complet, dacă $k=2^m$, și coder necomplet, dacă $k < 2^m$.

Fie, drept exemplu, este dat tabelul de adevăr al unui coder necomplet care are $k = 10$ intrări și $m = 4$ ieșiri (vezi tabelul 3.2). Tabelul 3.2 este completat conform definiției coderului. În tabel este utilizată logica „negativă”.

Tabelul 3.2. Tabelul de adevăr al unui coder necomplet pentru care $k = 10$ și $m = 4$

Nr. d/o	Variabile										Funcții			
	x_9	x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	F_3	F_2	F_1	F_0
0	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	1	0	0	0	1
2	1	1	1	1	1	1	1	0	1	1	0	0	1	0
3	1	1	1	1	1	1	0	1	1	1	0	0	1	1
4	1	1	1	1	1	0	1	1	1	1	0	1	0	0
5	1	1	1	1	0	1	1	1	1	1	0	1	0	1
6	1	1	1	0	1	1	1	1	1	1	0	1	1	0
7	1	1	0	1	1	1	1	1	1	1	0	1	1	1
8	1	0	1	1	1	1	1	1	1	1	1	0	0	0
9	0	1	1	1	1	1	1	1	1	1	1	0	0	1

Din tabelul 3.2 de adevăr se obțin următoarele funcții logice:

$$F_3(x_9, x_8, x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0) = \bar{x}_9 \times x_8 \times x_7 \times x_6 \times x_5 \times x_4 \times x_3 \times x_2 \times x_1 \times x_0 + \\ + x_9 \times \bar{x}_8 \times x_7 \times x_6 \times x_5 \times x_4 \times x_3 \times x_2 \times x_1 \times x_0 = \bar{x}_9 + \bar{x}_8 = \overline{x_9 \times x_8}; \quad (3.9)$$

$$F_2(x_9, x_8, x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0) = \overline{x_7 \times x_6 \times x_5 \times x_4}; \quad (3.10)$$

$$F_1(x_9, x_8, x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0) = \overline{x_7 \times x_6 \times x_3 \times x_2}; \quad (3.11)$$

$$F_0(x_9, x_8, x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0) = \overline{x_9 \times x_7 \times x_5 \times x_3 \times x_1}. \quad (3.12)$$

În fig. 3.3 este prezentată schema electrică a coderului, construită conform funcțiilor logice (3.9), ..., (3.12). În fig. 3.3 sunt următoarele notări: $+V_{cc}$ – sursa de semnal; 9, ..., 0 – comutatoarele canalelor semnalelor electrice; F_3 , ..., F_0 – ieșirile coderului sau magistrala de date. Din schemă se vede, că la ieșiri se obține codul 0111_2 , dacă numai la intrarea cu numărul 7 se aplică zero.

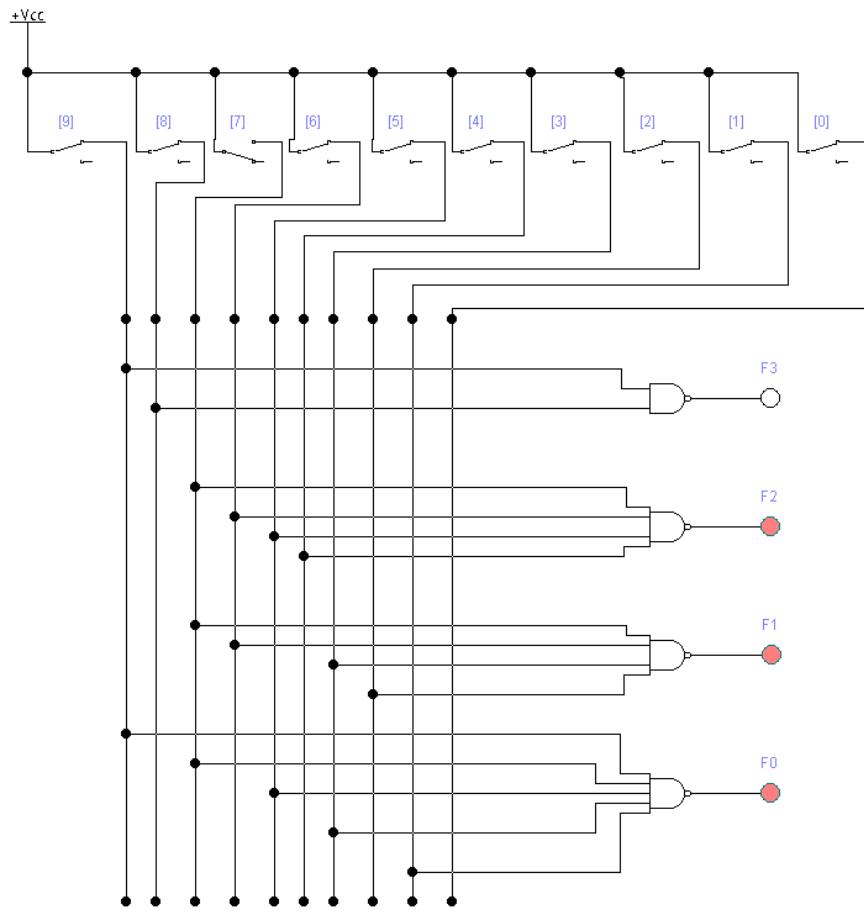


Fig. 3.3. Shema electrică a coderului.

Simbolul convențional al coderului este prezentat în figura 3.4.

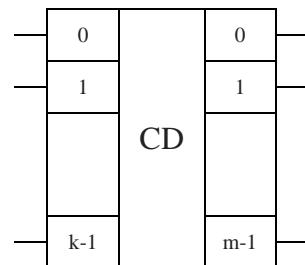


Fig. 3.4. Simbolul convențional al coderului.

Coderul este utilizat, în special, pentru a transforma un semnal electric în cod binar. Coderul este parte componentă a unității de comandă a micropresorului, al memoriei permanente și convertorului analogic-numeric (CAN).

3.3. Translator de cod

Translatorul de cod este construit din decoder și coder (vezi fig. 3.5) și servește pentru conectarea dispozitivelor care au magistrale de date de ordin diferit. Translatorul de cod trebuie să îndeplinească următoarele condiții:

- numărul lui de intrări este mai mic față de numărul de ieșiri sau $k < n$;
- numărul de ieșiri al decoderului este egal cu numărul de intrări al coderului.

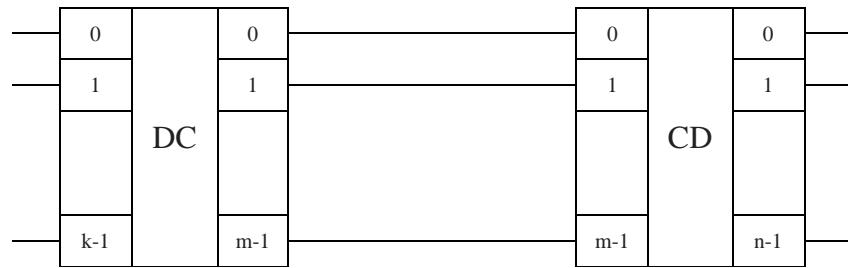


Fig. 3.5. Schema bloc a translatorului de cod.

3.3. Multiplexor

Definiție. Numim multiplexor un dispozitiv electronic combinațional utilizat pentru transmiterea informației din n ($n > 1$) canale într-un singur canal.

Operația de transmitere a informației din mai multe canale într-un singur canal poate fi efectuată de modelul structural, prezentat în fig. 3.6.

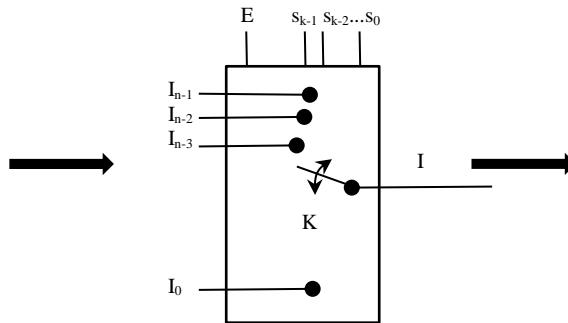


Fig. 3.6. Modelul structural al multiplexorului.

În fig. 3.6 sunt următoarele notări: I_{n-1} , I_{n-2} , I_{n-3}, \dots, I_0 – magistrala de date la intrarea multiplexorului; E – canalul de comandă (permite sau nu transmiterea informației); S_{k-1} , S_{k-2}, \dots, S_0 – magistrala de dirijare a poziției comutatorului K ; K – comutator; I – canalul de date la ieșirea multiplexorului.

Modelul structural al multiplexorului are următoarele regimuri de lucru:

- $E = 0$, multiplexorul este în regim de repaus (nu transmite informația);
- $E = 1$, multiplexorul este în regim de lucru (transmite informația aplicată la una din intrări spre ieșire conform conectării efectuate de comutatorul K).

Drept exemplu, v-a fi construit un multiplexor cu 4 canale de date la intrare (se mai spune „din patru în unu”). Stările multiplexorului sunt prezentate în tabelul de adevăr, unde x poate fi 0 sau 1 (vezi tabelul 3.3).

Tabelul 3.3. Tabelul de adevăr al multiplexorului din patru în unu

E	0	1	1	1	1
S_1	x	0	0	1	1
S_0	x	0	1	0	1
I	0	I_0	I_1	I_2	I_3

Din tabelul de adevăr al multiplexorului se obține următoarea funcție logică

$$I = EI_0\bar{S}_1\bar{S}_0 + EI_1\bar{S}_1S_0 + EI_2S_1\bar{S}_0 + EI_3S_1S_0. \quad (3.13)$$

În fig. 3.7 este prezentată schema electrică a multiplexorului, construită conform funcției logice (3.13). În schema electrică sunt următoarele notări: I_3, I_2, I_1, I_0 – magistrala de date la intrarea multiplexorului; E – canalul de comandă (permite ($E = 1$) sau nu permite ($E = 0$) transmiterea informației); s_1, s_0 – canale responsabile de comutarea canalelor magistralei de date I_3, I_2, I_1, I_0 la ieșirea I (în același timp s_1, s_0 determină adresele canalelor magistralei de date); I – canalul de date la ieșirea multiplexorului. Conform schemei electrice din fig. 3.7 pentru $E = 1$, $s_1s_0 = 10$ numai canalul de date I_2 se conectează cu ieșirea I .

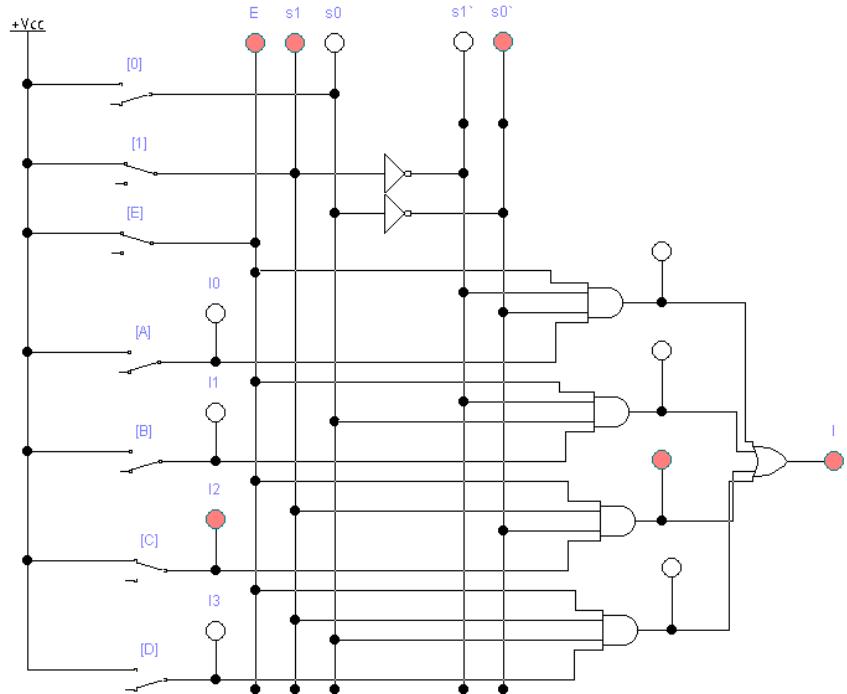


Fig. 3.7. Schema electrică a multiplexorului.

Simbolul convențional al multiplexorului este prezentat în figura 3.8.

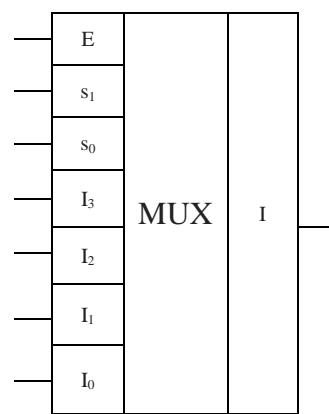


Fig. 3.8. Simbolul convențional al multiplexorului.

Multiplexorul poate fi utilizat pentru executarea următoarelor operații:

- a) transmiterea informației din mai multe canale într-un canal sau multiplexarea informației;

b) transformarea unui cod binar paralel în cod binar consecutiv. În acest caz la intrări se aplică un cod binar paralel care se menține un timp mai indelungat și cu ajutorul codurilor de dirijare $s_{k-1}, s_{k-2}, \dots, s_0$ și comutatorului K se unesc pe rind canalele de la intrare se unesc pe rind cu canalul de la ieșire (după finalizarea transmiterii codului la intrările multiplexorului se aplică alt cod binar);

c) pentru generarea codurilor binare în ciclu – în acest caz la intrare se aplică permanent un cod binar, iar pozițiile comutatorului K se schimbă ciclul sub acțiunea semnalelor de dirijare $s_{k-1}, s_{k-2}, \dots, s_0$.

3.4. Multiplexor generator

În fig. 3.9 este prezentată schema electrică a multiplexorului care poate genera semnale în ciclu. Forma semnalului la ieșirea I a multiplexorului este determinată de codul binar aplicat la intrările I_3, I_2, I_1, I_0 . În schema electrică sunt următoarele notări: I_3, I_2, I_1, I_0 – magistrala de date la intrarea multiplexorului; E – canalul de comandă (permite ($E = 1$) sau nu permite ($E = 0$) transmiterea informației); s_1, s_0 – canale responsabile de comutarea canalelor magistralei de date I_3, I_2, I_1, I_0 la ieșirea I (în același timp s_1, s_0 determină adresele canalelor magistralei de date); G – comutatorul care permite comutarea regimului de generare în ciclu a semnalului la ieșirea I a multiplexorului; I – canalul de date la ieșirea multiplexorului.

Pentru a obține la ieșirea I a multiplexorului semnale în ciclu este necesar de comutat comutatoarele A, B, C, D în conformitate cu valorile I_0, I_1, I_2, I_3 apoi de conectat generatorul și bistabilii cu ajutorul comutatorului G. Semnale formate de bistabili schimbă s_1, s_0 în ciclu.

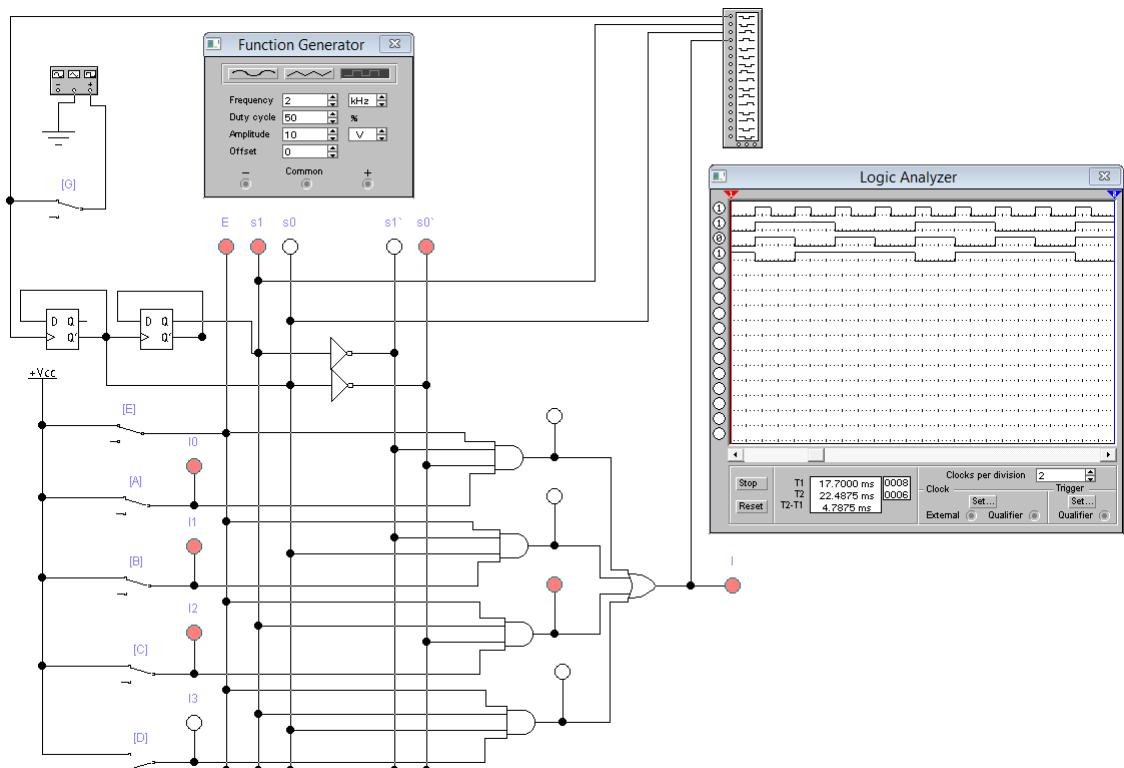


Fig. 3.9. Schema electrică a multiplexorului generator.

În schema din fig. 3.9 $E = 1$, iar $I_3 = 0, I_2 = 1, I_1 = 1$ și $I_0 = 1$. Introducând aceste date în formula (3.13) se obține funcția logică a semnalelor la ieșirea I a multiplexorului

$$I = \bar{s}_1 \bar{s}_0 + \bar{s}_1 s_0 + s_1 \bar{s}_0. \quad (3.14)$$

Pentru alte combinații I_3, I_2, I_1, I_0 se obțin alte funcții I sau alte semnale la ieșirea multiplexorului.

3.5. Demultiplexor

Definiție. Numim demultiplexor un dispozitiv electronic combinațional utilizat pentru transmiterea informației dintr-un canal în mai multe canale.

Operația de transmitere a informației dintr-un canal în mai multe canale poate fi efectuată de modelul structural, prezentat în fig. 3.10.

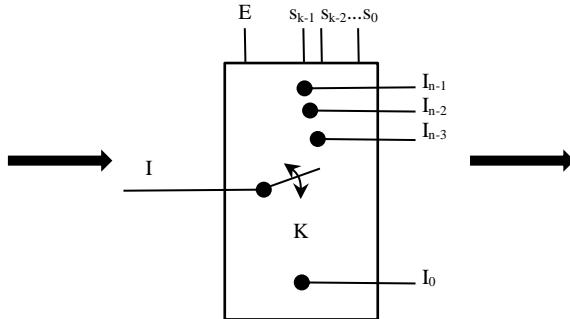


Fig. 3.10. Modelul structural al demultiplexorului.

În fig. 3.10 sunt următoarele notări: I – canalul de date la intrarea demultiplexorului;; E – canalul de comandă (permite sau nu permite transmiterea informației); $S_{k-1}, S_{k-2}, \dots, S_0$ – magistrala de dirijare a poziției comutatorului K; K – comutatorul; $I_{n-1}, I_{n-2}, I_{n-3}, \dots, I_0$ – magistrala de date la ieșirea demultiplexorului.

Modelul structural al demultiplexorului are următoarele regimuri de lucru:

- E = 0, demultiplexorul este în regim de repaus (nu transmite informația);
- E = 1, demultiplexorul este în regim de lucru (transmite informația aplicată la intrare spre una din ieșiri conform conectării efectuate de comutatorul K).

Pentru exemplificare, v-a fi construit un demultiplexor cu 4 canale de date la ieșire (se mai spune „din unu în patru”). Stările demultiplexorului sunt prezentate în tabelul de adevăr, unde x poate fi 0 sau 1 (vezi tabelul 3.4).

Tabelul 3.4. Tabelul de adevăr al demultiplexorului din unu în patru

E	0	1	1	1	1
S_1	x	0	0	1	1
S_0	x	0	1	0	1
I_3	0	0	0	0	I
I_2	0	0	0	I	0
I_1	0	0	I	0	0
I_0	0	I	0	0	0

Din tabelul de adevăr al multiplexorului se obțin următoarele funcții logice

$$I_3 = EI S_1 S_0; \quad (3.15)$$

$$I_2 = EI S_1 \bar{S}_0; \quad (3.16)$$

$$I_1 = EI \bar{S}_1 S_0; \quad (3.17)$$

$$I_0 = EI \bar{S}_1 \bar{S}_0. \quad (3.18)$$

În fig. 3.11 este prezentată schema electrică a demultiplexorului, construită conform funcțiilor logice (3.15), ..., (3.18). În schema electrică sunt următoarele notări: I – magistrala de date consecutivă la intrarea multiplexorului; E – canalul de comandă (permite ($E = 1$) sau nu

permite ($E = 0$) transmiterea informației); s_1, s_0 – canale responsabile de comutarea canalului magistralei de date I la una din ieșiri I_3, I_2, I_1, I_0 (în același timp s_1, s_0 determină adresele canalelor I_3, I_2, I_1, I_0). Conform schemei electrice din fig. 3.11 pentru $E = 1$ și $s_1s_0 = 01$ canalul de date I se conectează cu ieșirea I_1 .

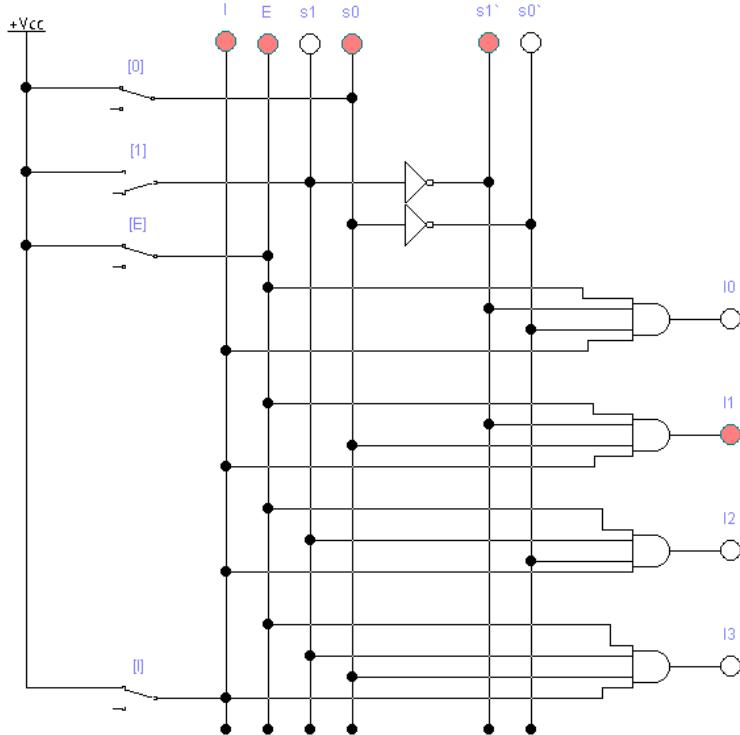


Fig. 3.11. Schema electrică a demultiplexorului.

Simbolul convențional al demultiplexorului este prezentat în figura 3.12.

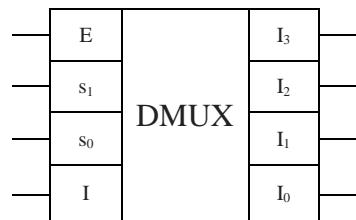


Fig. 3.12. Simbolul convențional al demultiplexorului.

Demultiplexorul poate fi utilizat pentru executarea următoarelor operații:

- transmiterea informației dintr-un canal în mai multe canale sau demultiplexarea (fragmentarea) informației;
- transformarea unui cod binar consecutiv în cod binar pseudoparalel. În acest caz cu ajutorul codurilor de dirijare $s_{k-1}, s_{k-2}, \dots, s_0$ și comutatorului K se unesc pe rind canalele de la ieșire cu canalul de la intrare (după finalizarea transmiterii codului la intrarea demultiplexorului se aplică alt cod binar).

3.6. Comparator

Definiție. Numim comparator un dispozitiv electronic combinațional utilizat pentru compararea codurilor binare.

Un comparator digital sau un comparator de cod este un dispozitiv logic cu intrări la care pot fi aplicate două coduri binare – $A(a_{n-1}, a_{n-2}, \dots, a_0)$ și $B(b_{n-1}, b_{n-2}, \dots, b_0)$. De obicei, microcircuitele comparatoarelor au trei ieșiri binare, la unul, din care, este emis un semnal, obținut în rezultatul comparării codurilor aplicate la intrări:

- a) $A > B$;
- b) $A = B$;
- c) $A < B$.

Unele microcircuite ale comparatoarelor au numai ieșirea $A = B$.

În comparator se compară biții codurilor. Drept exemplu, în tabelul 3.5 sunt prezentate stările comparatorului de ordinul unu care compară doi biți $A = a_0$ și $B = b_0$.

Tabelul 3.5. Tabelul de adevăr al comparatorului de ordinul unu

Nr. d/o	Intrări		Ieșiri		
	A	B	$A < B$	$A = B$	$A > B$
0	0	0	0	1	0
1	0	1	1	0	0
2	1	0	0	0	1
3	1	1	0	1	0

Din tabelul 3.5 se obțin următoarele funcții logice:

$$F(A < B) = \bar{A} \times B; \quad (3.19)$$

$$F(A = B) = \bar{A} \times \bar{B} + A \times B; \quad (3.20)$$

$$F(A > B) = A \times \bar{B}. \quad (3.21)$$

Conform formulelor (3.19), ..., (3.21) se obțin următoarele stări:

- a) $F(A < B) = 1$ dacă $A = 0$ și $B = 1$;
- b) $F(A = B) = 1$ dacă $A = B$ și $F(A = B) = 0$ dacă $A \neq B$;
- c) $F(A > B) = 1$ dacă $A = 1$ și $B = 0$.

Dependența dintre funcțiile $F(A = B)$, $F(A < B)$ și $F(A > B)$ poate fi obținută prin transformarea formulei (3.20), folosind axioma $\bar{A} \times A = 0$ sau $\bar{B} \times B = 0$:

$$\begin{aligned} F(A = B) &= \bar{A} \times \bar{B} + A \times B = \bar{A} \times A + \bar{A} \times \bar{B} + \bar{B} \times B + A \times B = \\ &= \cancel{\bar{A} \times A} + \cancel{A \times B} + \bar{A} \times \bar{B} + \cancel{\bar{B} \times B} = A \times (\bar{A} + B) + \bar{B} \times (\bar{A} + B) = (\bar{A} + B) \times (A + \bar{B}) = \\ &= \overline{(\bar{A} + B) \times (A + \bar{B})} = \overline{(\bar{A} + B)} + \overline{(A + \bar{B})} = \bar{A} \times B + A \times \bar{B} \end{aligned}$$

sau

$$F(A = B) = \overline{\bar{A} \times B + A \times \bar{B}} = \overline{F(A < B) + F(A > B)}. \quad (3.22)$$

În fig. 3.13 este prezentată schema electrică a comparatorului de ordinul unu, construită conform funcției logice (3.22).

Pentru compararea codurilor binare de ordin mai mare pot fi construite comparatoare din comparatoare de ordinul unu conectate în serie sau conform funcțiilor logice din tabele de adevăr. Pentru exemplificare, în tabelul 3.6 sunt prezentate stările comparatorului de ordinul doi, utilizat pentru compararea codurilor $A(a_1, a_0)$ și $B(b_1, b_0)$.

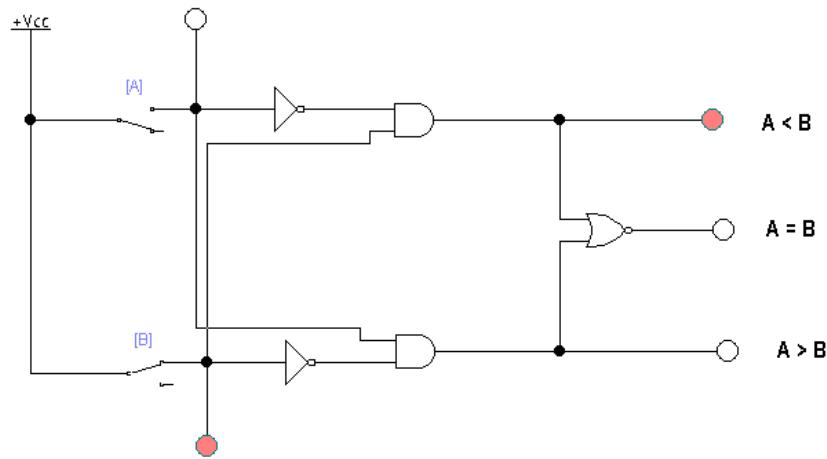


Fig. 3.13. Schema electrică a comparatorului de ordinul unu.

Tabelul 3.6. Stările comparatorului de ordinul doi

Nr. d/o	Intrări				Ieșiri		
	a_1	a_0	b_1	b_0	$A < B$	$A = B$	$A > B$
0	0	0	0	0	0	1	0
1	0	0	0	1	1	0	0
2	0	0	1	0	1	0	0
3	0	0	1	1	1	0	0
4	0	1	0	0	0	0	1
5	0	1	0	1	0	1	0
6	0	1	1	0	1	0	0
7	0	1	1	1	1	0	0
8	1	0	0	0	0	0	1
9	1	0	0	1	0	0	1
10	1	0	1	0	0	1	0
11	1	0	1	1	1	0	0
12	1	1	0	0	0	0	1
13	1	1	0	1	0	0	1
14	1	1	1	0	0	0	1
15	1	1	1	1	0	1	0

Din tabelul 3.6 se obțin următoarele funcții logice:

$$F(A < B) = \sum(1, 2, 3, 6, 7, 11); \quad (3.23)$$

$$F(A = B) = \sum(0, 5, 10, 15); \quad (3.24)$$

$$F(A > B) = \sum(4, 8, 9, 12, 13, 14). \quad (3.25)$$

Funcția $F(A = B)$ poate fi simplificată:

$$\begin{aligned} F(A = B) &= \bar{a}_1 \bar{a}_0 \bar{b}_1 \bar{b}_0 + \bar{a}_1 a_0 \bar{b}_1 b_0 + a_1 \bar{a}_0 b_1 \bar{b}_0 + a_1 a_0 b_1 b_0 = \\ &= \bar{a}_1 \bar{b}_1 (\bar{a}_0 \bar{b}_0 + a_0 b_0) + a_1 b_1 (\bar{a}_0 \bar{b}_0 + a_0 b_0) = (\bar{a}_1 \bar{b}_1 + a_1 b_1) \times (\bar{a}_0 \bar{b}_0 + a_0 b_0) = \\ &= a_1 \oplus b_1 \times a_0 \oplus b_0 \end{aligned}$$

sau

$$F(A = B) = \overline{a_1 \oplus b_1} \times \overline{a_0 \oplus b_0}. \quad (3.26)$$

Funcțiile $F(A < B)$ și $F(A > B)$ pot fi minimize prin metoda Veitch-Karnaugh (vezi fig. 3.14 și fig. 3.15).

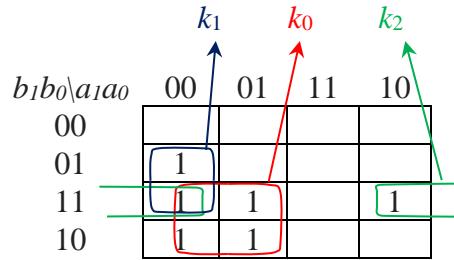


Fig. 3.14. Minimizarea funcției $F(A < B)$.

Din fig. 3.14 pentru $F(A < B)$ se obține:

$$F(A < B) = k_2 + k_1 + k_0 = \bar{a}_1 b_1 + \bar{a}_1 \bar{a}_0 b_0 + \bar{a}_0 b_1 b_0. \quad (3.27)$$

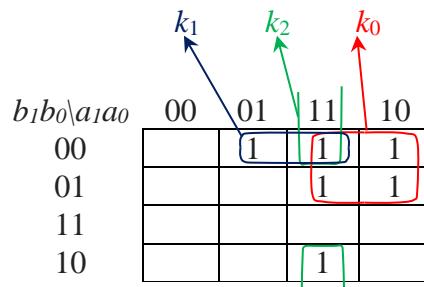


Fig. 3.15. Minimizarea funcției $F(A > B)$.

Din fig. 3.15 pentru $F(A > B)$ se obține:

$$F(A > B) = k_2 + k_1 + k_0 = a_1 \bar{b}_1 + a_0 \bar{b}_1 \bar{b}_0 + a_1 a_0 \bar{b}_0. \quad (3.28)$$

În fig. 3.16 este prezentată schema electrică a comparatorului de ordinul doi, construită conform funcțiilor logice (3.26), (3.27) și (3.28).

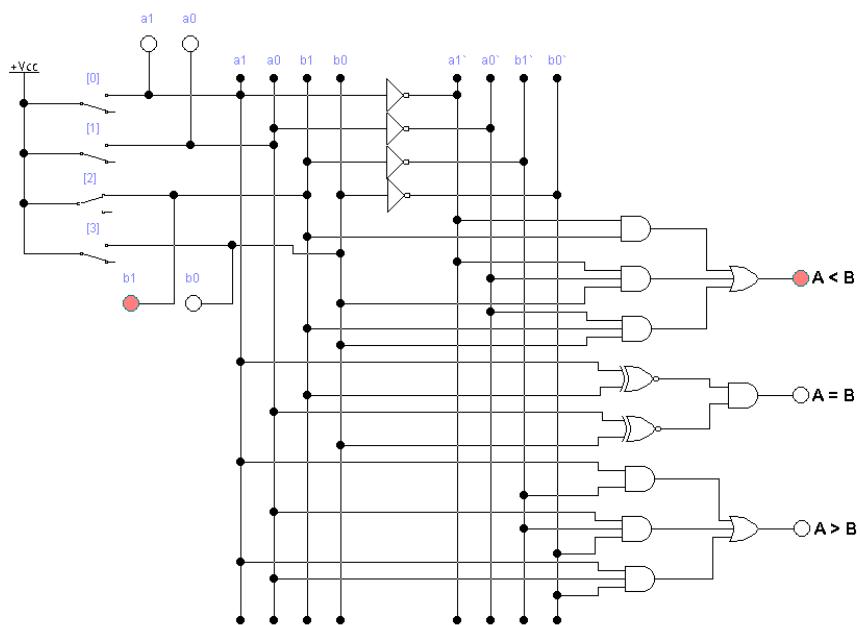


Fig. 3.16. Schema electrică a comparatorului de ordinul doi.

În fig. 3.16 este prezentată starea $A < B$, unde $A(a_1, a_0) = 00$ și $B(b_1, b_0) = 10$.
Simbolul convențional al comparatorului de ordinal 4 este prezentat în figura 3.17.

Intrările și ieșirile $A < B$, $A=B$, $A>B$ se folosesc pentru a conecta comparatoarele între ele cu scopul formării schemelor comparatoarelor de ordinul 8, 16 etc.

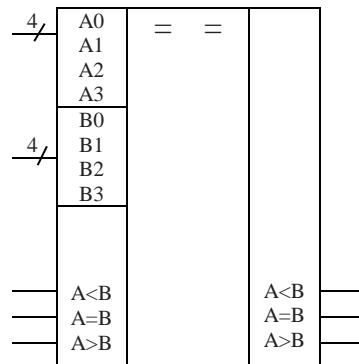


Fig. 3.17. Simbolul convențional al comparatorului de ordinal patru.

Comparatoarele sunt utilizate pe scară largă în calculatoare, în dispozitive numerice de măsurare, în rețele de comunicații radio și prin cablu, în aparate de uz casnic. Drept exemplu, un ceas digital cu alarmă conține un comparator digital care inițiază un semnal sonor, când ora curentă coincide cu cea specificată.

3.7. Definirea și clasificarea sumatorilor

Definiție. Numim sumator un dispozitiv electronic combinațional utilizat pentru sumarea codurilor.

Sumatorii pot fi clasificați după următoarele criterii:

- a) conform particularităților arhitecturale/constructive;
- b) conform modului de realizare a procesului adunării codurilor;
- c) conform metodei de transfer al depășirilor;
- d) conform sistemului de numerație utilizat.

A. Conform particularităților arhitecturale/constructive există:

- **sumator după modul** sau elementul logic SAU-EX cu două intrări și o singură ieșire (la intrări se aplică cite o cifră binară, iar la ieșire se obține rezultatul sumării după modul);
- **semisumator** cu două intrări și două ieșiri (la intrări se aplică cite o cifră binară, la o ieșire se obține rezultatul sumării după modul, iar la ieșirea a două se obține depășirea);
- **sumator complet (celulă de sumare)** cu trei intrări și două ieșiri (la două intrări se aplică cite o cifră binară din ordinul respectiv ale numerelor supuse sumării, la intrarea a treea se aplică depășirea din ordinul inferior, la o ieșire se obține rezultatul sumării după modul, iar la ieșirea a două se obține depășirea care este transmisă în ordinul următor superior);
- **sumator acumulator** care are memorie proprie.

B. Conform modului de realizare a procesului sumării codurilor sumatorii sunt numiți:

- **sumator consecutiv (în serie)**, în care sumarea cifrelor se efectuează consecutiv, bit cu bit în fiecare ordin al numerelor cu un singur sumator complet ;
- **sumator paralel-consecutiv**, în care mai mulți biți dintr-o pereche de numere sunt sumați în mod paralel, iar depășirile sunt transferate consecutiv;
- **sumator paralel**, în care toate perechile de cifre cu același ordin și depășirile se sumează concomitent.

C. Conform metodei de transfer al depășirilor există:

- **sumator cu transfer consecutiv al depășirilor (Ripple-carry adder)**;

- sumator cu transfer paralel sau accelerat în grup al depășirilor ([Carry-lookahead adders](#));
- sumator cu ocolirea transferului depășirilor ([Carry-skip adder](#));
- sumator cu sumare condiționată (Conditional sum adder);
- sumator cu transfer al depășirilor selectate ([Carry-select adder](#));
- sumator cu păstrarea depășirilor ([Carry-save adder](#)).

D. Conform sistemului de numerație utilizat sumatorii sunt clasificați în sumatori binari, ternari, cuaternari, octali, zecimali și hexazecimali.

3.8. Semisumator

Primul semisumator binar „Model K Adder” pe două relee electromecanice (semisumatorul Stibitz) a fost elaborat de George Robert Stibitz (1904 - 1995). El este unul dintre fondatorii recunoscuți pe plan internațional ale calculatoarelor digitale moderne. A lucrat ca cercetător la Bell Labs, cunoscut pentru munca sa din anii 1930 și 1940 privind implementarea logicii booleene în circuite digitale folosind relee electromecanice ca elemente de comutare.

Definiție. Numim semisumator un dispozitiv electronic combinațional care sumează două cifre binare.

Rezultatul sumării aritmetice pentru două cifre binare este prezentat în tabelul 3.7, unde: a, b – sunt cifre binare care se sumează; C – depășirea; S – suma după modul.

Tabelul 3.7. Rezultatul sumării aritmetice pentru două cifre binare

Nr. d/o	Intrări		Ieșiri	
	a	b	C	S
0	0	0	0	0
1	0	1	0	1
2	1	0	0	1
3	1	1	1	0

Din tabelul 3.7 se obțin următoarele funcții logice:

$$C = a \times b = \overline{a} \times \overline{b} = \overline{a + b}; \quad (3.29)$$

$$S = \overline{a} \times b + a \times \overline{b} = \overline{\overline{a}} \times \overline{b} + \overline{a} \times \overline{\overline{b}} = \overline{a + b} + \overline{a} \times \overline{b}. \quad (3.30)$$

Schema electrică a semisumatorului, construită conform formulelor (3.29) și (3.30) este prezentată în fig. 3.18.

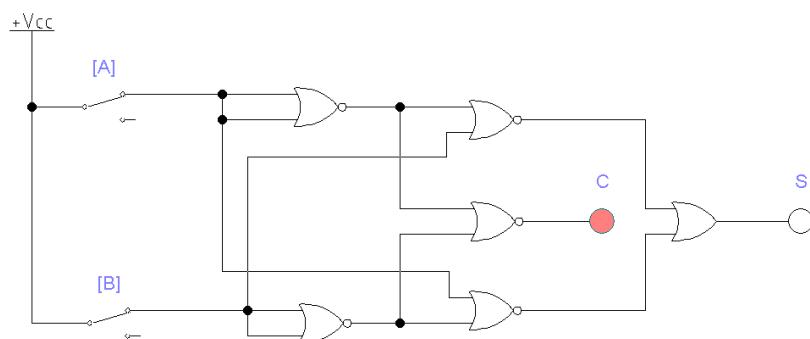


Fig. 3.18. Schema electrică a semisumatorului.

Este cunoscut faptul, că, în procesul sumării a două numere, în fiecare ordin se adună două cifre și cifra depășirii care este transferată din ordinul anterior inferior. Semisumatorul nu poate fi utilizat pentru construirea sumatorilor, deoarece are numai două intrări și nu poate lua în considerație depășirea.

Simbolul convențional al semisumatorului este prezentat în figura 3.19.

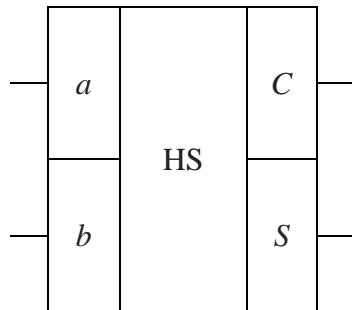


Fig. 3.19. Simbolul convențional al semisumatorului.

3.9. Sumator complet

Definiție. Numim **sumator complet** un dispozitiv electronic combinațional care sumează trei cifre binare.

Din cele prezentate în paragraful 3.8 urmează, că pentru construirea sumatorilor de ordinul $n \geq 2$ este necesar de un dispozitiv elementar de sumare cu trei intrări. Stările acestui dispozitiv, numit sumator complet, care sumează două cifre binare a_i, b_i și depășirea C_i transferată din ordinul anterior inferior sunt prezentate în tabelul 3.8. În tabelul 3.8 sunt noteate: C_{i+1} – depășirea pentru transfer în ordinul următor superior; S_i – suma după modul; i – ordinul.

Tabelul 3.8. Stările sumatorului complet

Nr. d/o	Intrări			Ieșiri	
	C_i	a_i	b_i	C_{i+1}	S_i
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

Din tabelul 3.8 obținem următoarele funcții logice:

$$C_{i+1} = \bar{C}_i a_i b_i + C_i \bar{a}_i b_i + C_i a_i \bar{b}_i + C_i a_i b_i; \quad (3.31)$$

$$S_i = \bar{C}_i \bar{a}_i b_i + \bar{C}_i a_i \bar{b}_i + C_i \bar{a}_i \bar{b}_i + C_i a_i b_i. \quad (3.32)$$

Funcțiile logice (3.31) și (3.32) pot fi prezentate și în modul următor:

$$C_{i+1} = a_i b_i + C_i (a_i \oplus b_i); \quad (3.33)$$

$$S_i = C_i \oplus a_i \oplus b_i. \quad (3.34)$$

Funcțiile logice (3.31) și (3.32) ale sumatorului complet (celulei de sumare) sunt executate de schema electrică din fig. 3.20, construită din doi semisumatori.

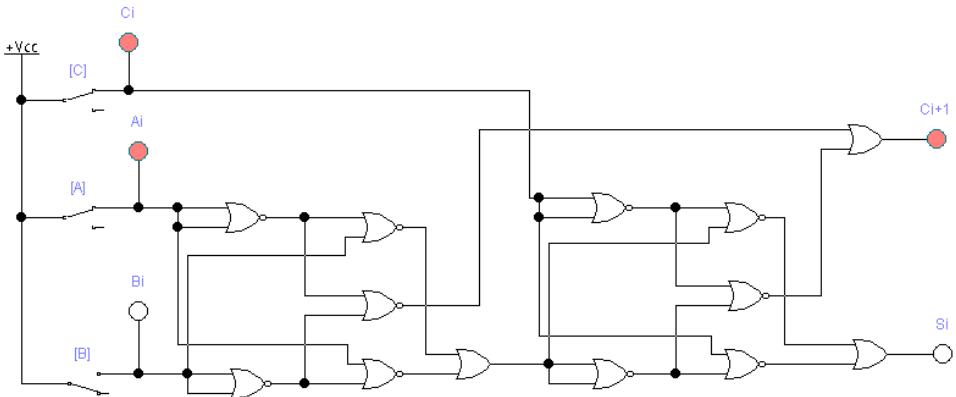


Fig. 3.20. Schema electrică a sumatorului complet.

Simbolul convențional al sumatorului complet este prezentat în figura 3.21.

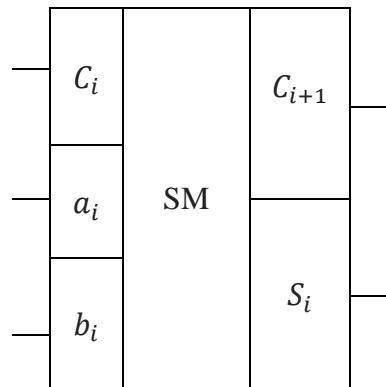


Fig. 3.21. Simbolul convențional al sumatorului complet.

Sumatorul complet se folosește pentru construirea sumatorilor consecutivi (în serie), sumatorilor paralel-consecutivi, sumatorilor paraleli și altor dispozitive care execută operații aritmetice și/sau logice.

3.10. Sumator consecutiv

Schema electrică a sumatorului consecutiv de ordinul patru este prezentată în fig. 3.22. Sumatorul consecutiv constă din următoarele dispozitive:

- trei registre consecutive cu deplasare directă de ordinul patru pentru stocarea numerelor $A(a_3, a_2, a_1, a_0)$, $B(b_3, b_2, b_1, b_0)$ și rezultatului sumării lor $S(s_3, s_2, s_1, s_0)$;
- sumatorul complet SM care execută operația de sumare a numerelor $A(a_3, a_2, a_1, a_0)$ și $B(b_3, b_2, b_1, b_0)$;
- un bistabil D necesar pentru stocarea depășirilor.

Procesul de lucru al sumatorului consecutiv include etapele:

- comutatorul S deconectat** – etapa înscrerii numerelor $A(a_3, a_2, a_1, a_0)$ și $B(b_3, b_2, b_1, b_0)$ în registre;
- comutatorul S conectat** – în acest caz poate fi executat procesul de sumare a numerelor și stocarea rezultatului prin aplicarea unui set de semnale de comandă C la intrările

tuturor regiszrelor (sumarea, executată de un singur sumator complet SM, are loc în regim consecutiv, începînd cu perechea de biți inferiori a_0, b_0 și depășirea C_0).

Dezavantajul principal al sumatorului consecutiv constă în faptul, că intervalul de timp, necesar pentru executarea operației de sumare a numerelor, include un set de cicluri de ceas, numărul cărora este direct proporțional cu ordinul numerelor care se sumează. Timpul necesar pentru sumarea numerelor de ordinul n poate fi calculat conform formulei:

$$T_{sum}(n) = n \times (T_s + T_d), \quad (3.35)$$

unde T_{sum} – intervalul de timp necesar pentru executarea operației de sumare a numerelor, n – ordinul numerelor sumate, T_s – intervalul de timp necesar pentru sumarea unei perechi de biți a_i și b_i , iar T_d – intervalul de timp necesar pentru transferul depășirii de la ieșirea C_0 la intrarea C_i a sumatorului complet de către bistabilul D (aici C_0 și C_i sunt prezentate conform notărilor în programul Electronics Workbench).

Pentru micșorarea intervalului de timp, necesar pentru executarea operației de sumare a numerelor, se utilizează sumatori de tip paralel.

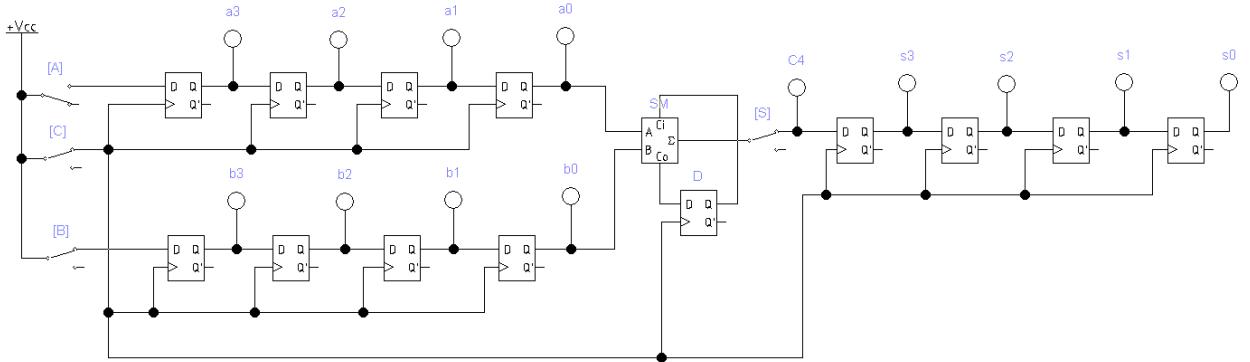


Fig. 3.22. Schema electrică a sumatorului consecutiv.

Simbolul convențional al sumatorului de ordinul patru este prezentat în fig. 3.23.

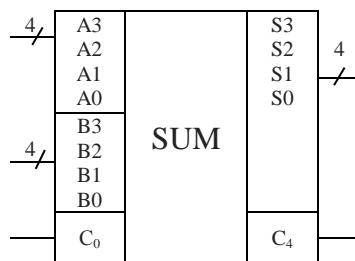


Fig. 3.23. Simbolul convențional al sumatorului de ordinul patru.

3.11. Sumator paralel cu transfer consecutiv al depășirilor

Sumatorul paralel cu transfer consecutiv al depășirilor se construiește din sumatoare complete conectate consecutiv. Pentru exemplificare, în fig. 3.22 este prezentată schema electrică a sumatorului consecutiv de ordinul patru. În schema electrică sunt următoarele notări: 0 – comutatorul canalului depășirii C_0 ; 2, 4, 6 și 8 – comutatoarele canalelor numărului A(a_3, a_2, a_1, a_0); 1, 3, 5 și 7 – comutatoarele canalelor numărului B(b_3, b_2, b_1, b_0); C_4, C_3, C_2, C_1, C_0 – indicatoarele depășirilor; s_3, s_2, s_1, s_0 – indicatoarele rezultatelor sumării după modul.

În cazul sumatorului paralel cu transfer consecutiv al depășirilor perechile de cifre $a_i b_i$, unde $i = 0, 1, \dots, n-1$, se aplică concomitent la intrările sumatorului și sumarea acestor cifre are loc tot concomitent, însă rezultatul corect al sumării în fiecare ordin se obține numai după sosirea

depăşirii C_i din ordinul anterior inferior. Timpul necesar pentru sumarea numerelor de ordinul n poate fi calculat conform formulei:

$$T_{sum}(n) = T_s + n \times T_d, \quad (3.36)$$

unde T_{sum} – intervalul de timp necesar pentru executarea operaţiei de sumare pentru două numere, n – ordinul numerelor sumate, T_s – intervalul de timp necesar pentru sumarea tuturor perechilor de biţi a_i şi b_i (sumarea are loc în primul semisumator al sumatorului complet), iar T_d – intervalul de timp necesar pentru transferul depăşirii C_i din ordinul anterior inferior şi finalizarea operaţiei de sumare de către semisumatorul următor al sumatorului complet.

În fig. 3.22 este prezentat rezultatul sumării $S(C_4, s_3, s_2, s_1, s_0) = 11000_2$ al numerelor $A(a_3, a_2, a_1, a_0) = 1101_2$ şi $B(b_3, b_2, b_1, b_0) = 1011_2$.

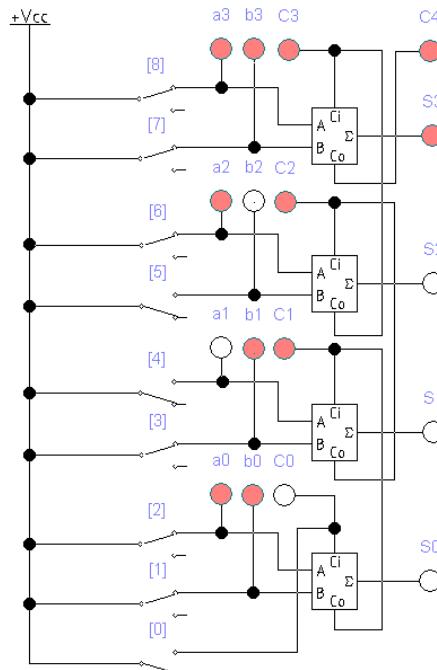


Fig. 3.22. Schema electrică a sumatorului paralel cu transfer consecutiv al depăşirilor.

Simbolul convențional al sumatorului paralel de ordinul patru cu transfer consecutiv al depăşirilor este prezentat în fig. 3.23.

3.12. Sumator paralel cu transfer paralel al depăşirilor

Pentru a micșora timpul necesar pentru executarea operaţiei de sumare a numerelor a fost elaborată schema sumatorului paralel cu transfer paralel (accelerat în grup) al depăşirilor.

Potențialitatea de a construi un sumator paralel cu transfer paralel al depăşirilor poate fi demonstrată prin analiza procesului de sumare într-un sumator complet prezentat în fig. 3.23.

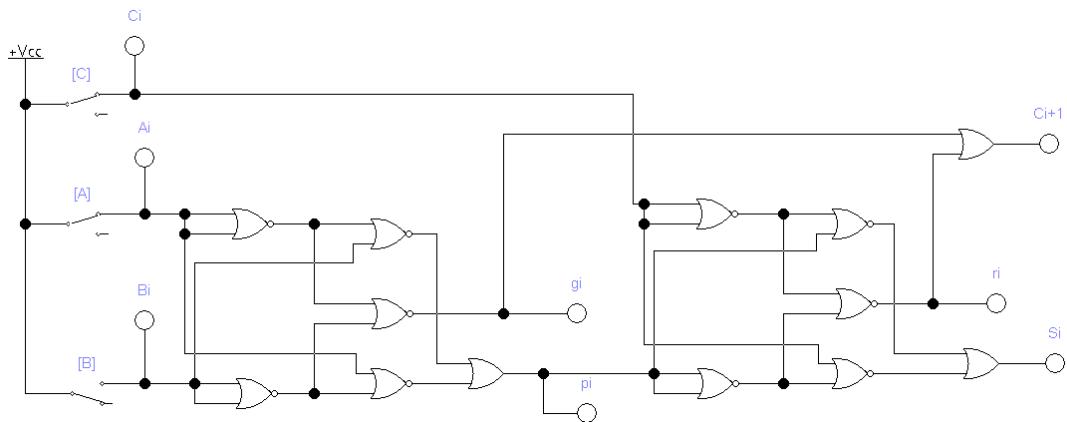


Fig. 3.23. Schema electrică a sumatorului complet de ordinul „ i ”.

În schema electrică, prezentată în fig. 3.23, sunt următoarele notări: $+V_{cc}$ – sursa de tensiune; C , A , B – comutatoare; a_i , b_i și C_i – canalele pentru aplicarea perechii de cifre cu același ordin ale numerelor și depășirii din ordinul anterior inferior; g_i , p_i – depășirea și suma după modul ale cifrelor a_i , b_i , obținute de semisumatorul din prima treaptă al sumatorului complet; r_i – depășirea obținută de către semisumatorul treptei a doua al sumatorului complet în procesul sumării C_i și p_i ; $C_{i+1}s_i$ – depășirea și suma după modul ale cifrelor a_i , b_i și C_i .

Stările realizate în schema sumatorului complet de ordinul „ i ” sunt prezentate în tabelul 3.9.

Tabelul 3.9. Stările realizate în schema a sumatorului complet de ordinul „ i ”

Nr. d/o	Stările la intrări			Stările intermediare			Stările la ieșiri	
	C_i	a_i	b_i	g_i	p_i	r_i	C_{i+1}	s_i
0	0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	1
2	0	1	0	0	1	0	0	1
3	0	1	1	1	0	0	1	0
4	1	0	0	0	0	0	0	1
5	1	0	1	0	1	1	1	0
6	1	1	0	0	1	1	1	0
7	1	1	1	1	0	0	1	1

Pentru un sumator complet (vezi paragraful 3.9) a fost obținută următoarea funcție logică:

$$C_{i+1} = a_i b_i + C_i(a_i \oplus b_i). \quad (3.37)$$

Din tabelul 3.9 se vede, că

$$g_i = a_i b_i, \quad (3.38)$$

$$p_i = a_i \oplus b_i \quad (3.39)$$

sau pentru C_{i+1} , exprimat prin g_i și p_i , se obține:

$$C_{i+1} = g_i + p_i C_i. \quad (3.40)$$

Conform formulei (3.40) se obține următorul algoritm de calcul al depășirilor C_1, C_2, C_3, \dots pentru $i = 0, 1, 2, 3, \dots$:

$$\begin{aligned}
i = 0 \rightarrow C_1 &= g_0 + p_0 C_0; \\
i = 1 \rightarrow C_2 &= g_1 + p_1 C_1 = g_1 + p_1(g_0 + p_0 C_0) = g_1 + p_1 g_0 + p_1 p_0 C_0; \\
i = 2 \rightarrow C_3 &= g_2 + p_2 C_2 = g_2 + p_2(g_1 + p_1 C_1) = g_2 + p_2 g_1 + p_2 p_1 C_1 = \\
&= g_2 + p_2 g_1 + p_2 p_1(g_0 + p_0 C_0) = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 C_0; \\
i = 3 \rightarrow C_4 &= g_3 + p_3 C_3 = g_3 + p_3(g_2 + p_2 C_2) = g_3 + p_3 g_2 + p_3 p_2 C_2 = \\
&= g_3 + p_3 g_2 + p_3 p_2(g_1 + p_1 C_1) = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 C_1 = \\
&= g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1(g_0 + p_0 C_0) = \\
&= g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 C_0.
\end{aligned}$$

Algoritmul de calcul a depășirilor poate fi continuat și pentru $i = 4, 5, 6, 7, \dots$. Din formulele (3.38), (3.39) și fig. 3.23 a sumatorului complet observăm, că mărimele intermediare g_i, p_i sunt determinate de primul semisumator și valorile lor depind numai de a_i și b_i . Din aceasta rezultă, că toate depășirile $C_1, C_2, C_3, C_4, \dots$ pot fi calculate concomitent, dacă sunt cunoscute mărimele intermediare g_i, p_i , unde $i = 0, 1, 2, 3, \dots$

În conformitate cu algoritmul, prezentat mai sus, este construită schema electrică a sumatorului paralel cu transfer paralel (accelerat în grup) al depășirilor (vezi fig. 3.24).

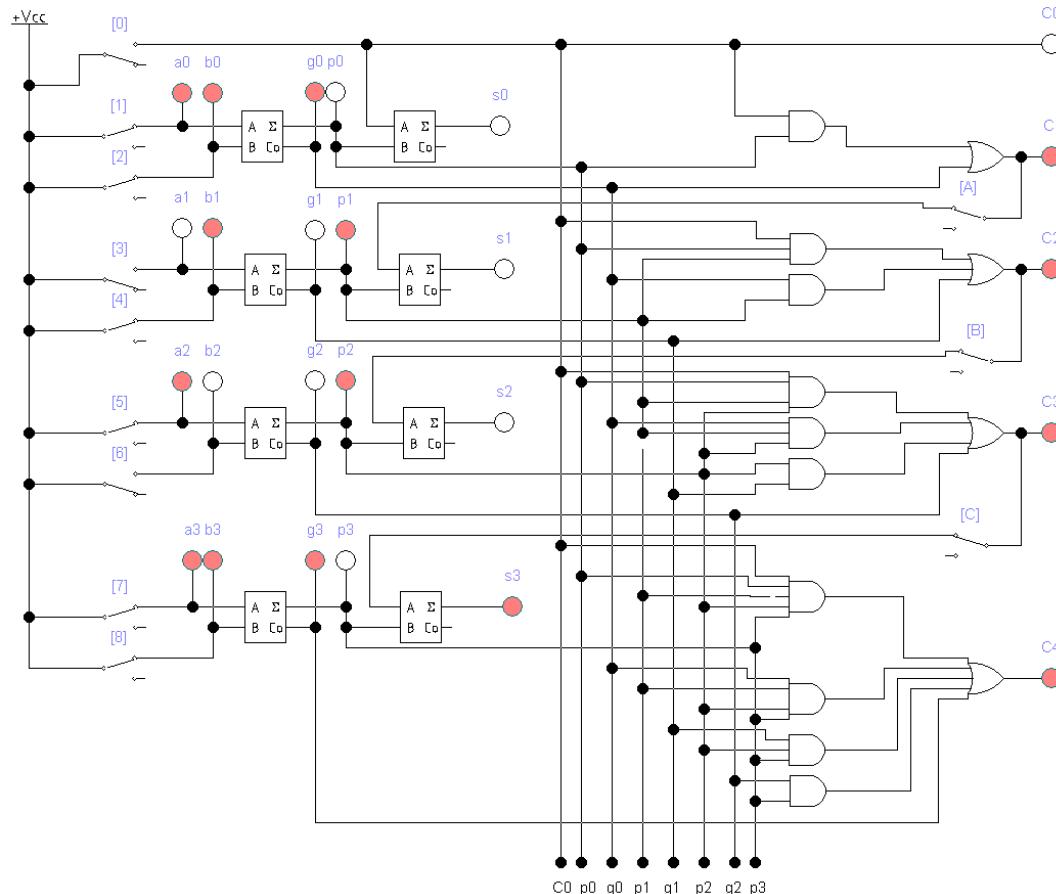


Fig. 3.24. Schema electrică a sumatorului paralel cu transfer paralel al depășirilor.

În schema electrică, prezentată în fig. 3.24, sunt următoarele notări: $+V_{cc}$ – sursa de tensiune; 0 – comutatorul canalului depășirii C_0 ; 1, 3, 5, 7 – comutatoarele magistralei numărului $A(a_3, a_2, a_1, a_0)$; 2, 4, 6, 8 – comutatoarele magistralei numărului $B(b_3, b_2, b_1, b_0)$; A, B, C – comutatoare.

Sumarea numerelor $A(a_3, a_2, a_1, a_0)$ și $B(b_3, b_2, b_1, b_0)$ are loc în trei etape:

- 1) la intrările sumatorului se aplică numerele $A(a_3, a_2, a_1, a_0)$, $B(b_3, b_2, b_1, b_0)$ și semisumatorii din prima treaptă a sumatorului determină concomitent toate mărimele intermediare g_i, p_i , unde $i = 0, 1, 2, 3$;

2) schema construită din elemente řI, SAU, numită schema de transfer paralel, folosind numai mărimile intermediare g_i, p_i , concomitent determină toate depăširile C_4, C_3, C_2, C_1 ;

3) depăširile C_4, C_3, C_2, C_1 se aplică la intrările A ale semisumatorilor treptei a doua care determină sumele după modul s_3, s_2, s_1, s_0 .

Rezultatul sumării numerelor $A(a_3, a_2, a_1, a_0)$ și $B(b_3, b_2, b_1, b_0)$ este determinat de codul C_4, s_3, s_2, s_1, s_0 sau

$$A(a_3, a_2, a_1, a_0) + B(b_3, b_2, b_1, b_0) = S(C_4, s_3, s_2, s_1, s_0). \quad (3.41)$$

În schema electrică, prezentată în fig. 3.24 este prezentat cazul sumării numerelor $A(a_3, a_2, a_1, a_0) = 1001_2$ și $B(b_3, b_2, b_1, b_0) = 1101_2$, în rezultatul căreea se obține $S(C_4, s_3, s_2, s_1, s_0) = 10110_2$.

Timpul necesar pentru sumarea numerelor nu depinde de ordinul lor și poate fi calculat conform formulei:

$$T_{sum} = T_1 + T_2 + T_3, \quad (3.42)$$

unde T_{sum} – intervalul de timp necesar pentru executarea operației de sumare pentru două numere, T_1 – intervalul de timp necesar pentru determinarea concomitentă a mărimilor intermediare g_i, p_i , unde $i = 0, 1, 2, 3$ de către semisumatorii primei trepte a sumatorului, T_2 – intervalul de timp necesar pentru determinarea depăširilor de către schema de transfer parallel și T_3 – intervalul de timp necesar pentru calcularea rezultatului final prin intermediul semisumatorilor treptei a doua a sumatorului.

Sumatorului paralel cu transfer paralel al depăširilor obține sumele numerelor într-un interval de timp cu mult mai mic în comparație cu intervalul de timp necesar pentru sumare în cazul sumatorului consecutiv sau sumatorului paralel cu transfer consecutiv al depăširilor.

Neajunsul principal al sumatorului paralel cu transfer paralel al depăširilor constă în creșterea complexității schemei electrice odată cu majorarea ordinului numerelor sumate.

Simbolul convențional al sumatorului de ordinul patru este prezentat în fig. 3.23.

Pentru sumarea numerelor de ordin mai mare sumatorii paraleli cu transfer paralel al depăširilor sunt conectați consecutiv sau paralel.

3.13. Unitatea de scădere

Unitatea de scădere poate fi construită din sumator cu mici modificări în schema electrică. Modificările care sunt necesare să fie introduse în sumator pot fi determinate prin analiza operației de scădere a numerelor binare. Diferența dintre două numere binare $A(a_{n-1}, \dots, a_1, a_0)$ și $B(b_{n-1}, \dots, b_1, b_0)$ este prezentată prin formula:

$$D(d_{n-1}, \dots, d_1, d_0) = A(a_{n-1}, \dots, a_1, a_0) - B(b_{n-1}, \dots, b_1, b_0). \quad (3.43)$$

Deoarece sumatorul îndeplinește operația de adunare este necesar ca operația de scădere din formula (3.43) să fie înlocuită prin operația de adunare.

Pentru diferența $D(d_{n-1}, d_{n-2}, \dots, d_1, d_0)$ se pot realiza următoarele variante:

- 1) $D(d_{n-1}, \dots, d_1, d_0) \geq 0$;
- 2) $D(d_{n-1}, \dots, d_1, d_0) < 0$.

3.13.1. Unitatea de scădere pentru $D(d_{n-1}, \dots, d_1, d_0) \geq 0$

Fie $D(d_{n-1}, \dots, d_1, d_0) \geq 0$. Introducem în formula (3.43) un număr $C(c_{n-1}, c_{n-2}, \dots, c_1, c_0)$ fără a schimba valoarea diferenței $D(d_{n-1}, \dots, d_1, d_0)$:

$$D = A + C - B - C = A + (C - B) - C. \quad (3.44)$$

În schema electrică a sumatorului v-or fi introduse modificări minimale dacă:

- a) $C = 2^n$;
- b) $C = 2^n - 1$.

Fie $C = 2^n$. În acest caz conform (3.44) se obține

$$D = A + (2^n - B) - 2^n = A + B^{(2)} - 2^n, \quad (3.45)$$

unde $B^{(2)}$ este numărul complementar. Numărul complementar $B^{(2)} = B^{(1)} + 1$, unde $B^{(1)}$ este numărul inversat.

Exemplu. Fie avem două numere $A(a_3, a_2, a_1, a_0) = 1101_2$ și $B(b_3, b_2, b_1, b_0) = 1000_2$ de ordinul $n = 4$. În acest caz $2^n = 2^4 = 16_{10} = 10000_2$, iar numărul complementar $B^{(2)}$ se obține în modul următor

$$B^{(2)} = 2^4 - B = 10000_2 - 1000_2 = \textcolor{red}{1000}_2.$$

Numărul complementar poate fi prezentat și prin numărul inversat $\bar{B} = B^{(1)}$ (aici numărul inversat este notat $B^{(1)}$):

$$B^{(1)} = \bar{B} = \overline{1000}_2 = 0111_2.$$

$$B^{(2)} = \bar{B} + 0001_2 = B^{(1)} + 0001_2 = \overline{1000}_2 + 0001_2 = 0111_2 + 0001_2 = \textcolor{red}{1000}_2.$$

Transformarea numărului B în număr complementar $B^{(2)}$ sau în număr inversat $B^{(1)}$ permite înlocuirea operației de scădere în operația de adunare. Pentru acest exemplu diferența dintre numerele $A(a_3, a_2, a_1, a_0) = 1101_2$ și $B(b_3, b_2, b_1, b_0) = 1000_2$ este

$$D = A + B^{(2)} - 2^4 = 1101_2 + 1000_2 - 10000_2 = 0101_2$$

$$\text{sau } D = A + B^{(2)} - 2^4 = 13_{10} + 8_{10} - 16_{10} = 5_{10}$$

După introducerea numărului inversat $B^{(1)}$ în formula (3.45) se obține

$$D = A + B^{(1)} + 1 - 2^n. \quad (3.46)$$

Fie $C = 2^n - 1$. În acest caz conform (3.44)

$$D = A + (2^n - 1 - B) - 2^n + 1 = A + B^{(2)} - 1 - 2^n + 1$$

sau se obține o formulă identică cu formula (3.46)

$$D = A + B^{(1)} + 1 - 2^n. \quad (3.46.A)$$

Conform formulelor (3.46) și (3.46.A) sumatorul v-a calculat corect diferența $D(d_{n-1}, \dots, d_1, d_0) \geq 0$ ale numerelor $A(a_{n-1}, \dots, a_1, a_0)$ și $B(b_{n-1}, \dots, b_1, b_0)$ dacă:

- a) numărul $B(b_{n-1}, b_{n-2}, \dots, b_1, b_0)$ v-a fi inversat sau la intrările magistralei B a sumatorului v-or fi instalată inversori;
- b) la intrarea C_0 a sumatorului v-a fi aplicată unitatea sau $C_0 = 1$;
- c) în rezultatul scăderii 2^n cifra cu ordinul superior n al diferenței se inversează ($d_n \rightarrow \bar{d}_n$) sau la ieșirea d_n este necesar să fie introdus un inversor.

Exemplu. Sunt date două numere de ordinul patru $A(a_3, a_2, a_1, a_0) = 1101_2$ și $B(b_3, b_2, b_1, b_0) = 1000_2$. De calculat diferența acestor numere conform formulelor (3.43) și (3.46).

Conform formulei (3.43) pentru $D(d_3, d_2, d_1, d_0)$ se obține următorul rezultat:

$$D(d_3, d_2, d_1, d_0) = A(a_3, a_2, a_1, a_0) - B(b_3, b_2, b_1, b_0) = 1101_2 - 1000_2 = 0100_2. \quad (3.47)$$

Pentru a calcula diferența $D(d_3, d_2, d_1, d_0)$ conform formulei (3.46) o transcriem cu indicarea ordinului numerelor ($n = 4$):

$$D = A + B^{(1)} + 1 - 2^4. \quad (3.48)$$

În continuare trebuie de efectuat următoarele operații:

- a) de inversat numărul $B(b_3, b_2, b_1, b_0)$

$$B(b_3, b_2, b_1, b_0) = 1001_2 \rightarrow B^{(1)}(b_3, b_2, b_1, b_0) = 0110_2;$$

- b) de calculat numărul complementar $B^{(2)}(b_3, b_2, b_1, b_0)$ (de adăugat unitatea la numărul inversat)

$$B^{(2)}(b_3, b_2, b_1, b_0) = B^{(1)}(b_3, b_2, b_1, b_0) + 0001_2 = 0110_2 + 0001_2 = 0111_2;$$

- c) de sumat numerele

$$A(a_3, a_2, a_1, a_0) + B^{(1)}(b_3, b_2, b_1, b_0) + 0001_2 = 1101_2 + 0110_2 + 0001_2 = 10100_2;$$

- d) de scăzut $2^4 = 10000_2$ sau de aflat diferența

$$D(d_3, d_2, d_1, d_0) = 10100_2 - 10000_2 = 0100_2. \quad (3.49)$$

Comparînd rezultatele obținute conform formulelor (3.43), (3.46) observăm, că ele sunt identice.

Schema electrică a unității de scădere de ordinul patru pentru $D(d_{n-1}, \dots, d_1, d_0) \geq 0$, construită din sumator paralel cu transfer consecutiv al depășirilor, este prezentată în fig. 3.25. În schema electrică sunt următoarele notări: $+V_{cc}$ – sursa de tensiune; 0 – comutatorul canalului depășirii C_0 ; 2, 4, 6 și 8 – comutatoarele canalelor numărului $A(a_3, a_2, a_1, a_0)$; 1, 3, 5 și 7 – comutatoarele canalelor numărului $B(b_3, b_2, b_1, b_0)$; C_4, C_3, C_2, C_1, C_0 – indicatoarele depășirilor; d_3, d_2, d_1, d_0 – indicatoarele rezultatelor scăderii. În schema electrică a unității de scădere este prezentat rezultatul operației de scădere

$$D(d_3, d_2, d_1, d_0) = A(a_3, a_2, a_1, a_0) - B(b_3, b_2, b_1, b_0) = 1101_2 - 1001_2 = 0100_2.$$

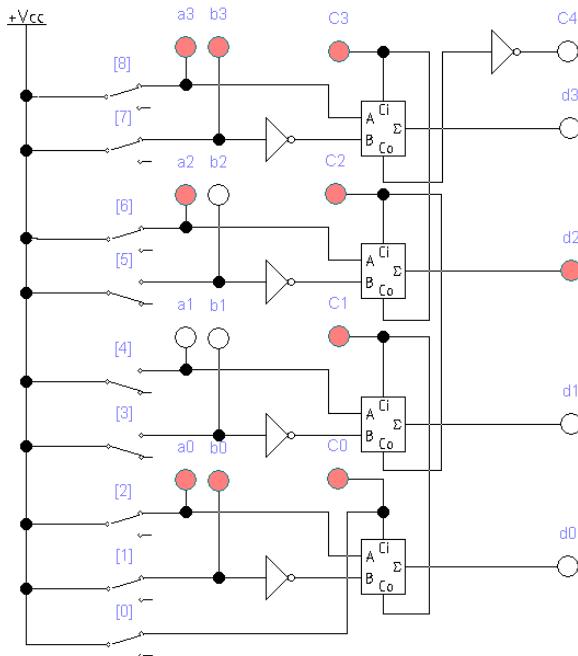


Fig. 3.25. Schema electrică a unității de scădere de ordinul patru pentru $D > 0$.

Schema electrică a unității de scădere poate fi construită și din sumatori paraleli cu transfer paralel al depășirilor. Simbolul convențional al unității de scădere de ordinul patru este prezentat în fig. 3.26.

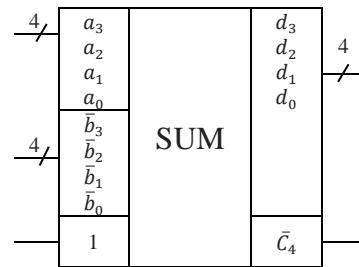


Fig. 3.26. Simbolul convențional al unității de scădere pentru $D(d_3, d_2, d_1, d_0) \geq 0$.

3.13.2. Unitatea de scădere pentru $D(d_{n-1}, d_{n-2}, \dots, d_1, d_0) < 0$

Unitatea de scădere prezentată în fig. 3.26 nu calculează corect dacă $|A| < |B|$. Modificările, necesare de a fi introduse în schema electrică a sumatorului cu scopul obținerii unității de scădere pentru cazul $|A| < |B|$ sau $D < 0$, pot fi determinate din formula (3.43) după schimbarea structurii formulei în modul următor:

$$|D| = |A - B| = |B - A|. \quad (3.50)$$

Introducem în formula (3.50) un număr $C(c_{n-1}, \dots, c_1, c_0)$ fără a schimba valoarea diferenței $D(d_{n-1}, \dots, d_1, d_0)$:

$$|D| = |B - A| = |C + B - A - C| = |C - (A + C - B)|. \quad (3.51)$$

Fie $C = 2^n - 1$. În acest caz conform (3.51)

$$|D| = |2^n - 1 - (A + 2^n - 1 - B)|. \quad (3.52)$$

În formula (3.51) expresia $(2^n - B)$ poate fi înlocuită prin $(B^{(1)} + 1)$, deoarece

$$2^n - B = B^{(2)} = B^{(1)} + 1. \quad (3.53)$$

În rezultatul înlocuirii se obține

$$|D| = |2^n - 1 - (A + B^{(1)})|. \quad (3.54)$$

În formula (3.54) expresia $(2^n - (A + B^{(1)}))$ poate fi înlocuită prin $((A + B^{(1)})^{(1)} + 1)$, deoarece

$$2^n - (A + B^{(1)}) = (A + B^{(1)})^{(2)} = (A + B^{(1)})^{(1)} + 1. \quad (3.55)$$

În rezultatul înlocuirii se obține formula finală

$$|D| = |(A + B^{(1)})^{(1)}|. \quad (3.56)$$

Conform formulei (3.56) sumatorul v-a calculat corect diferența $D(d_{n-1}, \dots, d_1, d_0) < 0$ ale numerelor $A(a_{n-1}, \dots, a_1, a_0)$ și $B(b_{n-1}, \dots, b_1, b_0)$ dacă:

a) numărul $B(b_{n-1}, \dots, b_1, b_0)$ v-a fi inversat sau la intrările magistralei B a sumatorului v-or fi instalați inversori;

b) rezultatul scăderii $D(d_{n-1}, \dots, d_1, d_0)$ v-a fi inversat sau la ieșirile magistralei D a sumatorului v-or fi instalați inversori.

Schema electrică a unității de scădere de ordinul patru pentru $D(d_{n-1}, \dots, d_1, d_0) < 0$, construită din sumator paralel cu transfer consecutiv al depășirilor, este prezentată în fig. 3.27. În schema electrică sunt următoarele notări: $+V_{cc}$ – sursa de tensiune; 0 – comutatorul canalului depășirii C_0 ; 2, 4, 6 și 8 – comutatoarele canalelor numărului $A(a_3, a_2, a_1, a_0)$; 1, 3, 5 și 7 – comutatoarele canalelor numărului $B(b_3, b_2, b_1, b_0)$; C_4, C_3, C_2, C_1, C_0 – indicatoarele depășirilor; d_3, d_2, d_1, d_0 – indicatoarele rezultatelor scăderii. În schema electrică a unității de scădere este prezentat rezultatul operației de scădere

$$|D(d_3, d_2, d_1, d_0)| = |A(a_3, a_2, a_1, a_0) - B(b_3, b_2, b_1, b_0)| = |0101_2 - 1101_2| = 1000_2.$$

Notă: rezultatul operației de scădere a numerelor este prezentat în cod binar direct.

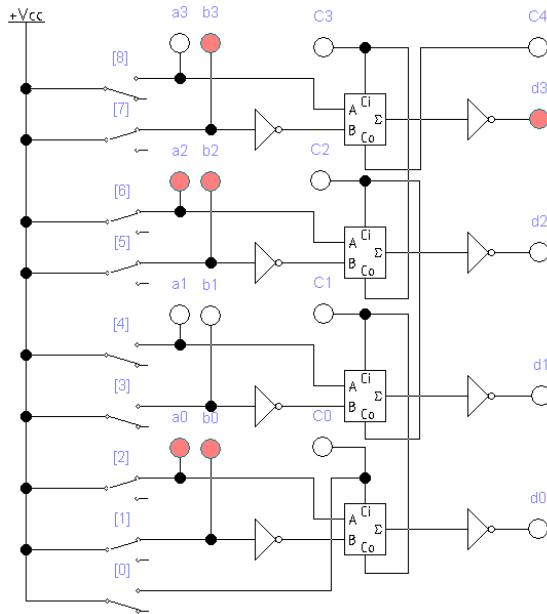


Fig. 3.27. Schema electrică a unității de scădere de ordinul patru pentru $D < 0$.

Simbolul convențional al unității de scădere de ordinul patru este prezentat în fig. 3.28.

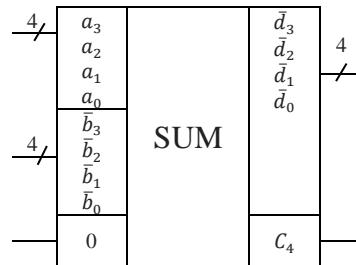


Fig. 3.28. Simbolul convențional al unității de scădere pentru $D(d_3, d_2, d_1, d_0) < 0$.

3.14. Unitatea de sumare a numerelor cu semn

Unitățile de scădere pentru $D > 0$ și $D < 0$, descrise în paragraful 3.13, prezintă rezultatul în cod binar direct fără indicarea semnului (semnul „+” sau semnul „-”).

Este cunoscut faptul, că în dispozitivele numerice semnul „+” este prezentat de valoarea logică „0”, iar semnul „-” este prezentat de valoarea logică „1” (vezi tabelul 3.10).

Tabelul 3.10. Prezentarea numerelor binare cu semn

Număr zecimal	Număr binar în cod direct (prezentarea pe 8 biți)
+127	0111 1111
+100	0110 0100
+0	0000 0000
-0	1000 0000
-100	1110 0100
-127	1111 1111

În schemele sumatorilor, prezentate în paragrafele precedente, semnele numerelor nu se iau în considerație. De aici rezultă, că scemele sumatorilor trebuie modificate sau completate.

Exemplu. Fie două numere binare cu semn:

$$A = +1101_2 = 01101_2 \text{ și } B = +1001_2 = 01001_2 - \text{cu roșu este indicat semnul „+”}.$$

$$S = A + B = 01101_2 + 01001_2 = 10110_2 - \text{rezultatul sumării este negativ „-”}.$$

Din acest exemplu se poate formula concluzia, că pentru obținerea unui rezultat corect, după sumarea numerelor cu semn, este interzis transferul depășirilor din ordinul cifrei în ordinul semnului.

Proiectarea unui sumator a numerelor cu semn se începe cu completarea tabelului de adevăr. Drept exemplu, v-a fi proiectat un sumator al numerelor cu semn de ordinul $n = 7$ $A(V_a, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$ și $B(V_b, b_6, b_5, b_4, b_3, b_2, b_1, b_0)$, unde V_a și V_b sunt semnele numerelor. Pentru aceste numere pot fi patru variante:

- a) $A > 0$ și $B > 0$;
- b) $A > 0$ și $B < 0$,
- c) $A < 0$ și $B > 0$;
- d) $A < 0$ și $B < 0$

În cadrul fiecărei variante trebuie de luat în considerație prezența sau lipsa transferului depășirii C_7 din ordinul cifrei în ordinul semnului. Rezultatele care se obțin pentru semnul V_s al sumei $S(V_s, s_6, s_5, s_4, s_3, s_2, s_1, s_0)$, pentru toate variantele posibile de sumare ale numerelor cu semn, sunt prezentate în tabelul 3.11.

Tabelul 3.11. Variantele posibile de sumare ale numerelor cu semn

Varianta	Semnul V_a	Semnul V_b	Depășirea C_7	Sumarea	Semnul V_s
$A > 0$ $B > 0$	0	0	0	$0, a_6, a_5, a_4, a_3, a_2, a_1, a_0 +$ $0, b_6, b_5, b_4, b_3, b_2, b_1, b_0 =$ $0, s_6, s_5, s_4, s_3, s_2, s_1, s_0$	$0 \rightarrow (V_s = C_7)$
	0	0	1	$0, a_6, a_5, a_4, a_3, a_2, a_1, a_0 +$ $0, b_6, b_5, b_4, b_3, b_2, b_1, b_0 =$ $1, s_6, s_5, s_4, s_3, s_2, s_1, s_0$	$1 \rightarrow (V_s = C_7)$
$A > 0$ $B < 0$	0	1	0	$0, a_6, a_5, a_4, a_3, a_2, a_1, a_0 +$ $1, b_6, b_5, b_4, b_3, b_2, b_1, b_0 =$ $1, s_6, s_5, s_4, s_3, s_2, s_1, s_0$	$1 \rightarrow (V_s = \bar{C}_7)$
	0	1	1	$0, a_6, a_5, a_4, a_3, a_2, a_1, a_0 +$ $1, b_6, b_5, b_4, b_3, b_2, b_1, b_0 =$ $0, s_6, s_5, s_4, s_3, s_2, s_1, s_0$	$0 \rightarrow (V_s = \bar{C}_7)$
$A < 0$ $B > 0$	1	0	0	$1, a_6, a_5, a_4, a_3, a_2, a_1, a_0 +$ $0, b_6, b_5, b_4, b_3, b_2, b_1, b_0 =$ $1, s_6, s_5, s_4, s_3, s_2, s_1, s_0$	$1 \rightarrow (V_s = \bar{C}_7)$
	1	0	1	$1, a_6, a_5, a_4, a_3, a_2, a_1, a_0 +$ $0, b_6, b_5, b_4, b_3, b_2, b_1, b_0 =$ $0, s_6, s_5, s_4, s_3, s_2, s_1, s_0$	$0 \rightarrow (V_s = \bar{C}_7)$
$A < 0$ $B < 0$	1	1	0	$1, a_6, a_5, a_4, a_3, a_2, a_1, a_0 +$ $1, b_6, b_5, b_4, b_3, b_2, b_1, b_0 =$ $0, s_6, s_5, s_4, s_3, s_2, s_1, s_0$	$0 \rightarrow (V_s = C_7)$
	1	1	1	$1, a_6, a_5, a_4, a_3, a_2, a_1, a_0 +$ $1, b_6, b_5, b_4, b_3, b_2, b_1, b_0 =$ $1, s_6, s_5, s_4, s_3, s_2, s_1, s_0$	$1 \rightarrow (V_s = C_7)$

Din tabelul 3.11 se obține formula

$$V_s = V_a \oplus V_b \oplus C_7 \quad (3.57)$$

Conform formulei (3.57) se poate afirma, că dispozitivul care determină valoarea semnului V_s constă din două elemente logice SAU-EX.

Funcția logică prezentată de formula (3.57) poate fi executată și de sumatorul complet (vezi paragraful 3.9, formula (3.34) $\rightarrow S_i = C_i \oplus a_i \oplus b_i$), dacă intrările și ieșirile lui v-or fi notate în modul următor (vezi fig. 3.29): $C_i = C_7$; $a_i = V_a$; $b_i = V_b$; $S_i = V_s$.

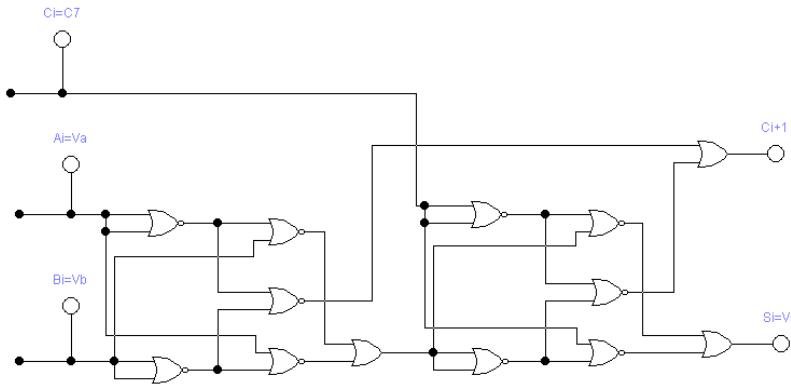


Fig. 3.29. Sumator complet pentru determinarea semnului.

Sumatorul complet determină semnul V_s fără a lua în considerație variantele sumării cu transfer ale depășirilor din ordinul cifrei în ordinul semnului. Pentru asigurarea corectitudinii determinării valorii semnului V_s , rezultatului sumării $S(V_s, s_6, \dots, s_1, s_0)$, identificării și excluderii variantelor de transfer ale depășirilor din ordinul cifrei în ordinul semnului schema sumatorului complet, prezentată în fig. 3.29, trebuie completată.

Din tabelul 3.11 pentru cinci combinații V_a , V_b și C_7 semnul V_s este determinat cu eroare și pentru funcția ER care determină stările în care se obține semnul V_s cu eroare se obține următoarea expresie logică:

$$ER = C_7 \bar{V}_a \bar{V}_b V_s + C_7 \bar{V}_a V_b \bar{V}_s + C_7 V_a \bar{V}_b \bar{V}_s + \bar{C}_7 V_a V_b \bar{V}_s + C_7 V_a V_b V_s. \quad (3.58)$$

Din formula (3.58) rezultă, că schema electrică a sumatorului complet trebuie completată cu cinci elemente SI și un element SAU. În fig. 3.30 este prezentată schema electrică a sumatorului paralel cu transfer consecutiv al depășirilor care execută sumarea numerelor cu semne pozitive de ordinul șapte $A(V_a, a_6, \dots, a_1, a_0)$ și $B(V_b, b_6, \dots, b_1, b_0)$.

Numerele cu semne negative trebuie transformate conform algoritmilor prezentați în paragraful 3.13 și de folosit unitatea de scădere.

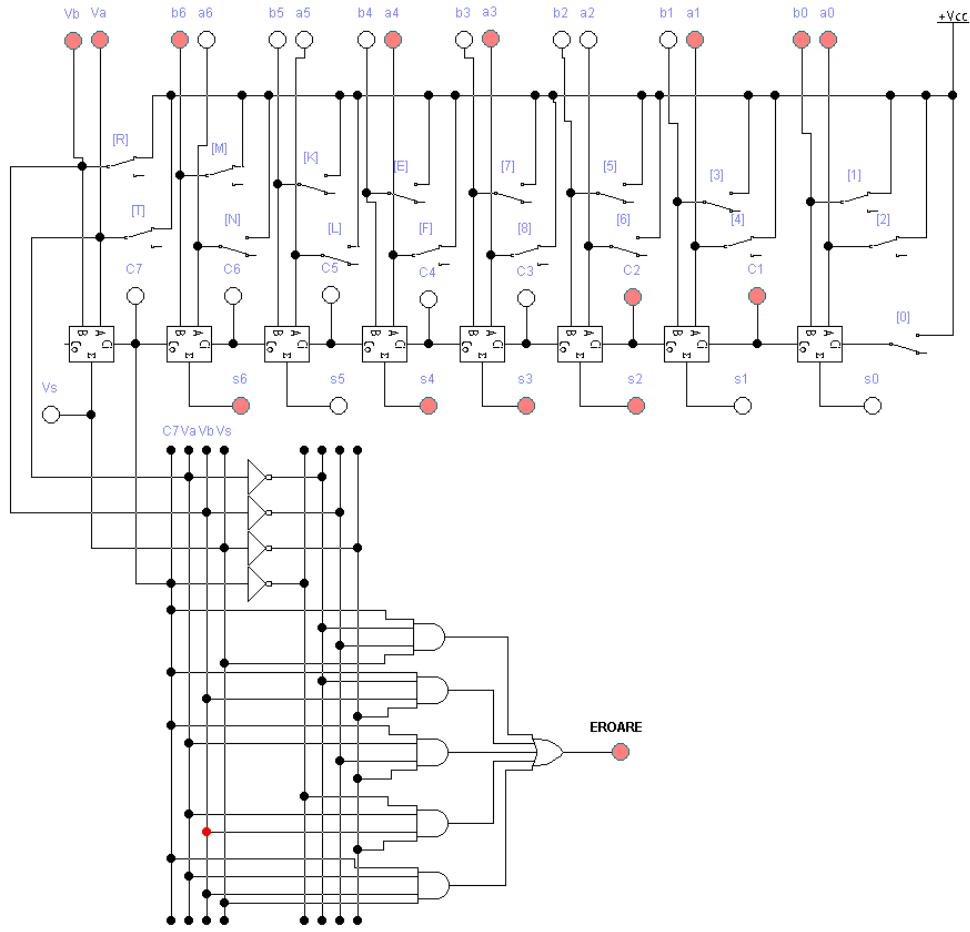


Fig. 3.30. Schema sumatorului numerelor cu semne pozitive de ordinul şapte.

3.15. Unitatea de înmulțire

Prealabil, în scopul construirii unității de înmulțire, v-a fi analizată procedura de înmulțire a două numere binare $A = 1101_2$ $B = 1011_2$.

				1	1	0	1	\times
			1	1	0	1	1	
		0	0	1	1	0	1	
	1	1	0	0	0	0		
1	0	0	0	1	1	1	1	

Fig. 3.31. Exemplu de înmulțire a numerelor binare $A = 1101_2$ și $B = 1011_2$.

Conform exemplului prezentat în fig. 3. 31 pot fi formulate următoarele:

- după înmulțirea la „1” a numărului $A = 1101_2$ se obține 1101_2 ;
- după înmulțirea la „0” a numărului $A = 1101_2$ se obține 0000_2 ;
- după fiecare înmulțire la „1” sau „0” numărul obținut este deplasat la stînga cu o poziție;
- numerele, obținute în rezultatul înmulțirilor, se sumează.

În conformitate cu condițiile condițiile a) a), b), c) și d) unitatea de înmulțire trebuie să execute următoarele operații:

- a) după înmulțirea la „1” a numărului $A = 1101_2$ să asigure pe magistrala de date valoarea 1101_2 ;
- b) după înmulțirea la „0” a numărului $A = 1101_2$ să asigure pe magistrala de date valoarea 0000_2 (condițiile (a) și (b) pot fi executate de elementele řI),
- c) după fiecare înmulțire la „1” sau „0” numărul obținut trebuie deplasat spre stînga (această condiție poate fi realizată de construcția unității de înmulțire);
- d) numerele, obținute în rezultatul înmulțirilor, să fie sumate sau în unitatea de înmulțire ca element principal trebuie utilizat sumatorul.

În fig. 3.32 este prezentată schema electrică a unității de înmulțire utilizată pentru înmulțirea numerelor binare de ordinul patru. În schema electrică sunt următoarele notări: $+V_{cc}$ – sursa de tensiune; $0, 1, 2, 3$ – comutatoarele canalelor numărului $A(a_3, a_2, a_1, a_0)$; $4, 5, 6$ și 7 – comutatoarele canalelor numărului $B(b_3, b_2, b_1, b_0)$; $p_7, p_6, p_5, p_4, p_3, p_2, p_1, p_0$ – indicatoarele rezultatului înmulțirii.

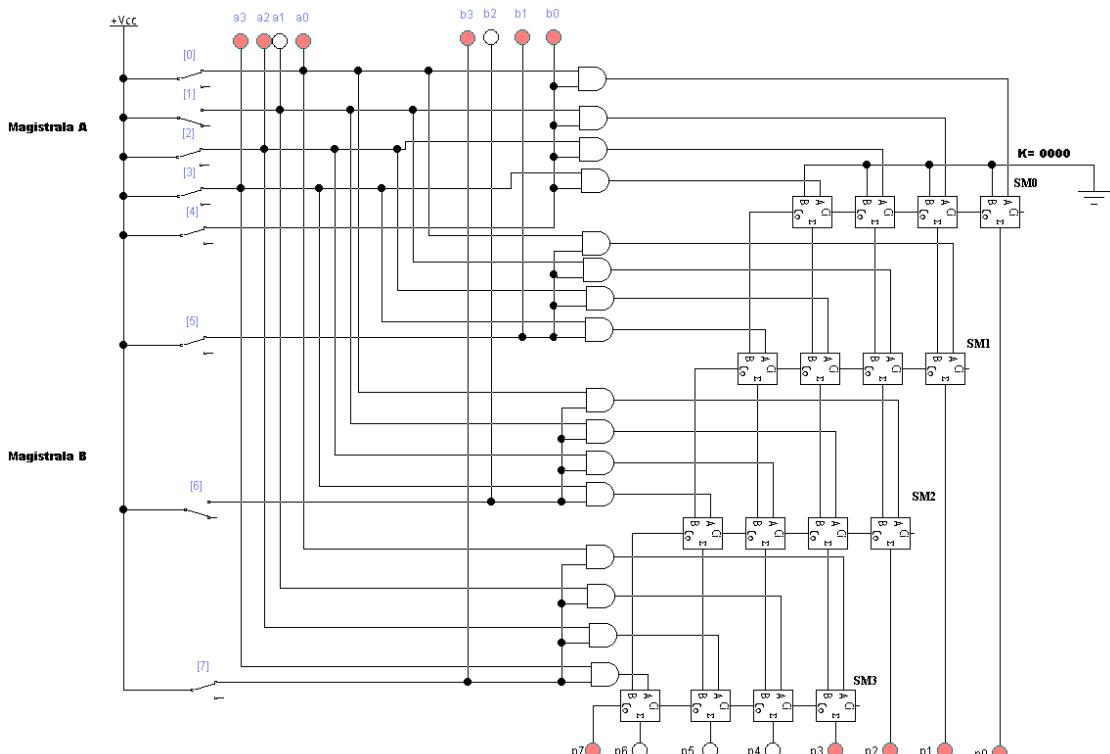


Fig. 3.32. Schema electrică a unității de înmulțire de ordinul patru.

În schema electrică a unității de înmulțire este prezentat rezultatul operației de înmulțire

$$\Pi(p_7, p_6, p_5, p_4, p_3, p_2, p_1, p_0) = A(a_3, a_2, a_1, a_0) \times B(b_3, b_2, b_1, b_0) =$$

$$= 1101_2 \times 1011_2 = 10001111_2.$$

Conform exemplului prezentat în fig. 3.32 în schema electrică sunt îndeplinite următoarele operații:

- a) **Sumatorul SM0.** $b_0 = 1$ și la intrările sumatorului SM0 sunt aplicate numerele

$$A(a_3, a_2, a_1, a_0) + K_0(0, 0, 0, 0) = 1101_2 + 0000_2 = 0110\textcolor{red}{1}_2;$$

- b) **Sumatorul SM1.** $b_1 = 1$ și la intrările sumatorului SM1 sunt aplicate numerele

$$A(a_3, a_2, a_1, a_0) + K_1(0, 1, 1, 0) = 1101_2 + 0110_2 = 1001\textcolor{red}{1}_2;$$

c) **Sumatorul SM2.** $\mathbf{b}_2 = \mathbf{0}$ și la intrările sumatorului SM2 sunt aplicate numerele

$$A(a_3, a_2, a_1, a_0) + K_2(1, 0, 0, 1) = 0000_2 + 1001_2 = 0100\textcolor{red}{1}_2;$$

d) **Sumatorul SM3.** $\mathbf{b}_3 = \mathbf{1}$ și la intrările sumatorului SM3 sunt aplicate numerele

$$A(a_3, a_2, a_1, a_0) + K_3(0, 1, 1, 0) = 1101_2 + 0110_2 = \textcolor{red}{10001}_2.$$

În final se obține $\Pi(p_7, p_6, p_5, p_4, p_3, p_2, p_1, p_0) = \textcolor{red}{10001111}_2$

4. MEMORIA CALCULATORULUI

4.1. Structura ierarhică a memoriei calculatorului

Memoria (un dispozitiv fizic sau un mediu de stocare) este o parte obligatorie a unui calculator, are o structură ierarhică și este utilizată pentru înscierea, păstrarea și prezentarea informației.

Majoritatea calculatoarelor moderne folosesc următoarea ierarhie a memoriei:

1. memoria internă este situată în interiorul calculatorului și este destinată pentru stocarea programelor și datelelor acestora în timpul funcționării calculatorului;
2. memoria externă este organizată de diferite dispozitive periferice (unități de disc magnetice, unități de disc optice, sticul etc) și este destinată pentru stocarea programelor și datelelor acestora pe termen lung.

Structura ierarhică a memoriei calculatorului (vezi fig. 4.1) este determinată de trei parametri: volumul (capacitatea) memoriei; timpul de acces; prețul/bit.

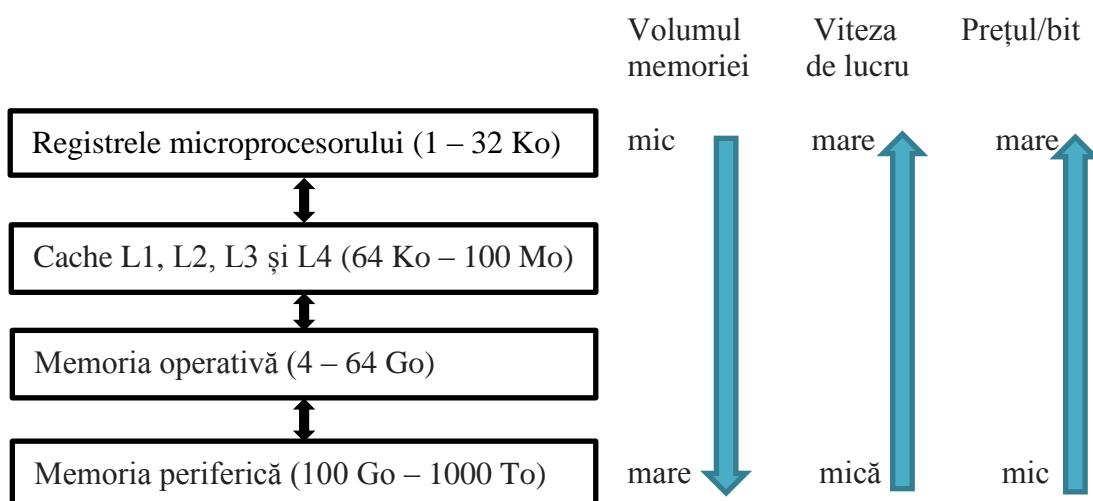


Fig. 4.1. Structura ierarhică a memoriei calculatorului.

4.1.1. Memoria internă a calculatorului

Memoria internă a calculatorului este distribuită în diferite dispozitive și include următoarele parți:

- a. memoria micropresorului (fig. 4.2);
- b. memoria internă centrală (memoria permanentă și memoria operativă);
- c. memoria dispozitivelor intrare/iesire.

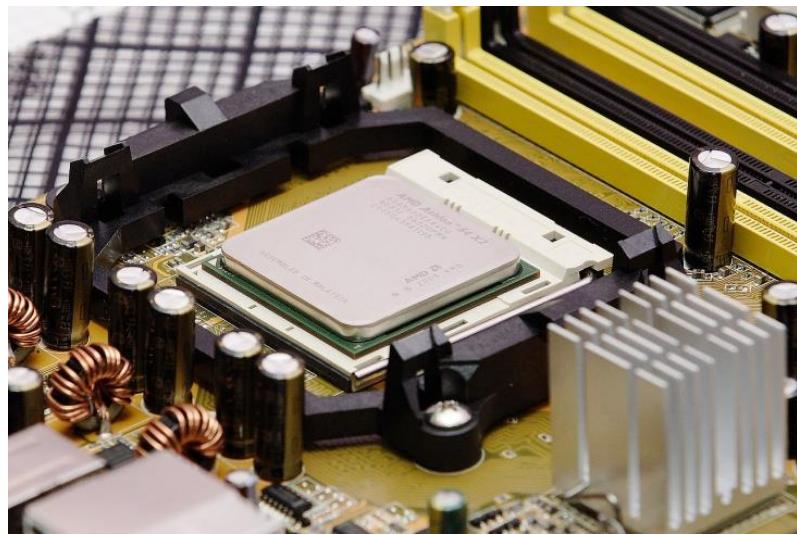


Fig. 4.2. Imaginea microprocesorului.

Memoria microprocesorului include registre de comandă, registre de date, acumulatorul, memoriile Cache de diferite niveluri - L1, L2, L3 și L4.

Registrele microprocesorului sunt cele mai rapide (au un timp de acces aproximativ egal cu perioada semnalului de tactare sau a unui ciclu de ceas), dar un volum mic (câteva sute sau, rareori, mii de octeți).

Cache este o unitate de memorie, concepută pentru a accelera accesul la datele care sunt plasate în unutăți de memorie cu o viteză de lucru mai mică, drept exemplu în memoria operativă. Unitățile de memorie Cache sunt utilizate de microprocesoare, hard disk-uri, browsere, servere web, serviciile DNS și WINS.

Memoria Cache a microprocesorului este împărțită în mai multe niveluri. Numărul maxim de cache-uri este de patru. În prezent, numărul de nivelurile dintr-un procesor de uz general poate ajunge la trei. Cache-urile de nivel $n + 1$ sunt, în general, mai mari după volum și mai lente în ceea ce privește accesul și viteza de transfer de date decât cache-urile de nivel n .

Cache L1 (Cache de nivel 1) este o unitate de memorie încorporată direct în microprocesor și are un volum de cîțiva zeci de chilioocteți. Cache L1 funcționează la frecvența microprocesorului și, în general, poate fi accesată la fiecare impuls de tactare (ciclu de ceas). În procesoarele moderne, Cache L1 este de obicei împărțită în două cache - cache-ul de instrucțiuni și cache-ul de date. Majoritatea procesoarelor fără Cache L1 nu pot funcționa.

Cache L2 este, de obicei, ca și Cache L1, încorporată direct în microprocesor. În primele versiuni de procesoare, Cache L2 a fost implementată ca un set separat de microcircuite de memorie pe placă de bază. Volumul Cache L2 este de la 128 Ko la 1-12 Mo și de 2 până la 10 ori mai lentă decât Cache L1. În microprocesoarele moderne cu multe nuclee, memoria Cache L2 este divizată uniform la numărul de nuclee.

Cache L3 este implementată pe un microcircuit separat, de 5-10 ori mai lentă decât Cache L2, dar încă semnificativ mai rapidă decât memoria operativă și poate avea un volum de peste 24 Mo. În sistemele cu multe nuclee, este utilizată în mod obișnuit și este proiectată pentru a sincroniza datele diferitor Cache L2.

Cache L4 este implementată pe un microcircuit separat, de 2-5 ori mai lentă decât Cache L3, dar mai rapidă decât memoria operativă și poate avea un volum de peste sute de Mo. Utilizarea Cache L4 este justificată doar pentru serverele și mainframe-urile multiprocesor de înaltă performanță.

Memoria internă centrală include:

- a) un microcircuit de memorie permanentă;
- b) blocuri din microcircuite de memorie operativă care ocupă aproximativ 99,9% din memoria totală a calculatorului.

Memoria permanentă (ROM – Read Only Memory), care, în special, stochează informațiile necesare pentru pornirea inițială a calculatorului. După cum sugerează și numele, informațiile din memoria permanentă se păstrează și după deconectarea calculatorului.

În prezent există microcircuite de memorie permanentă de tipul MROM, PROM, EPROM, EEPROM și Flash.

Memoria MROM (Mask Read Only Memory) este organizată de o clasă de microcircuite al căror conținut informațional este programat (construit din circuite electrice) de producătorul circuitului integrat. Terminologia „mască” a provine din procesul de fabricare a circuitului integrat, unde regiunile cipului sunt măscate în timpul procesului de fotolitografie. Principalul avantaj al microcircuitalor de tip MROM este costul acestora și sunt semnificativ mai ieftine decât orice alt tip de microcircuite de memorie construite din tranzistori. Memorii de tip MROM se folosesc în multe microprocesoare și unele microcontrolere.

Memoria PROM (Programmable Read Only Memory) este organizată de o clasă de microcircuite de memorie construite din semiconductoare (vezi fig. 4.3). Mediul de memorare reprezintă o matrice bidimensională de conductori (rânduri și coloane), la intersecția cărora se află o diodă (sau joncțiunea p-n a unui tranzistor) și un jumper special realizat din metal (de exemplu, nicrom sau aliaj de titan-tungsten) sau siliciu amorf. Programarea constă în trecerea unui curent prin jumperul corespunzător, care îl topește sau îl evaporă. Reconstrucția podurilor topite nu este posibilă.

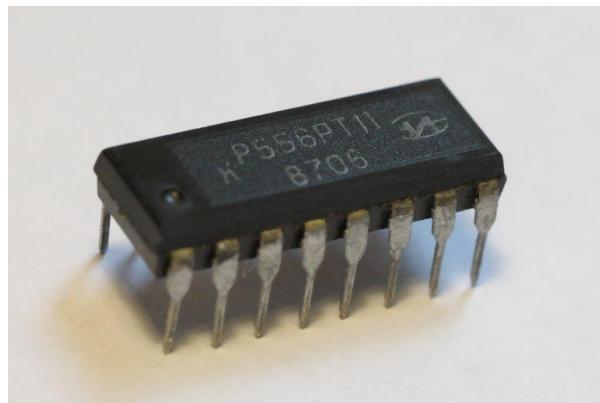


Fig. 4.3. Imaginea microcircuitului de tip PROM.

În ciuda fiabilității aparente a acestei soluții, această tehnologie s-a dovedit a fi destul de capricioasă. În timpul programării, punțile metalice au format picături și vaporii de metal, care s-au aşezat înapoi pe cristal în cele mai neașteptate locuri, cu consecințe neplăcute corespunzătoare. Podurile din polisilicon sunt capabile de auto-vindecare datorită migrației atomilor. Din acest motiv, microcircuitele după programare trebuiau menținute la temperaturi ridicate pentru o lungă perioadă de timp pentru a identifica potențialele defecte de acest tip.

În cele din urmă, microcircuitele de tip PROM au fost înlocuite în majoritatea dispozitivelor numerice prin microcircuite EPROM, EEPROM și Flash.

Memoria EPROM (Erasable Programmable Read Only Memory) este organizată de o clasă de microcircuite de memorie construite din tranzistori. Pentru înscrierea informației este utilizat un dispozitiv electronic numit programator. Spre deosebire de microcircuitele PROM, după programare informația din microcircuitele EPROM poate fi ştearsă (cu lumină ultravioletă puternică) și înscrisă altă informație cu ajutorul programatorului. Microcircuitele EPROM sunt ușor de recunoscut prin fereastra transparentă din sticlă de quarț din partea superioară a carcasei, prin care este vizibil cipul de siliciu și prin care lumina ultravioletă este iradiată în timpul ștergerii informației (vezi fig. 4.4).



Fig. 4.4. Microcontrolerul 8749 cu memorie EPROM.

Memoria EEPROM sau E²PROM (Electrically Erasable Programmable Read Only Memory) este organizată de o clasă de microcircuite de memorie construite din tranzistori. Informația în aşa microcircuite poate fi ştersă (cu semnale electrice) și înscrisă de sute de mii de ori atunci când sunt conectate la magistrala de sistem standardă microprocesor. Pentru EEPROM, fiecare celulă de memorie este ştersă automat atunci când sunt scrise informații noi, adică se pot schimba datele din orice celulă fără a afecta restul.

Memoria Flash este organizată de o clasă de microcircuite de memorie construite pe baza tehnologiei EEPROM. Datorită costului redus, a rezistenței mecanice, a posibilității stocării unui volum mare de informație, al timpului de acces relativ mic și a consumului redus de energie, memoria flash este utilizată pe scară largă în dispozitivele digitale portabile ca dispozitive de stocare (vezi fig. 4.5.). Un dezavantaj grav al acestei tehnologii este durata de viață limitată de microcircuitelor (5 -10 ani), precum și sensibilitatea la descărcări electrostatice.



Fig. 4.5. Dispozitive cu memorie Flash.

Memoria operativă (RAM – Random Access Memory) este organizată, în majoritatea calculatoarelor, de microcircuite de memorie de tipul SRAM (Static Random Access Memory) și DRAM (Dynamic Random Access Memory):

- microcircuitele SRAM sunt construite din bistabili;
- microcircuitele DRAM sunt construite din condensatori.

Microcircuitele SRAM sunt mai scumpe, în ceea ce privește stocarea 1 bit de informație, au un consum de energie mai mare, dar, de regulă, au un timp de acces mai scurt decât microcircuitele DRAM. În calculatoarele moderne, microcircuitele SRAM sunt des folosite în calitate de Cache ale procesoarelor. În microcircuitele SRAM informația se păstrează atît timp cît este alimentare electrică.

Microcircuitele DRAM stochează un bit de date sub formă de încărcare a condensatorului. O celulă de memorie conține un condensator și un tranzistor. Condensatorul este încărcat la tensiune înaltă sau joasă (logic 1 sau 0). Tranzistorul acționează ca un comutator

care conectează condensatorul la circuitul de dirijare situat în același microcircuit. Circuitul de dirijare permite citirea stării de încărcare a condensatorului sau modificarea ei. Deoarece păstrarea 1 bit de informație necesită mai puțină energie, în comparație cu cantitatea de energie consumată de microcircuitele SRAM, microcircuitele DRAM predomină pe calculatoarele moderne. Neajunsul principal al microcircuitelor DRAM constă în aceea, că condensatorul se descarcă și informația se păstrează un timp scurt (1...2 ms). Din această cauză așa microcircuite necesită un sistem de reînoire a informației.

În calculatoare se folosesc blocuri de microcircuite de memorie operativă. În fig. 4.6 este prezentat procesul evoluției constructive a memoriei operative (de sus în jos - [DIP](#), [SIPP](#), [SIMM](#) 30 pin, [SIMM](#) 72 pin, [DIMM](#), [DDR DIMM](#)).

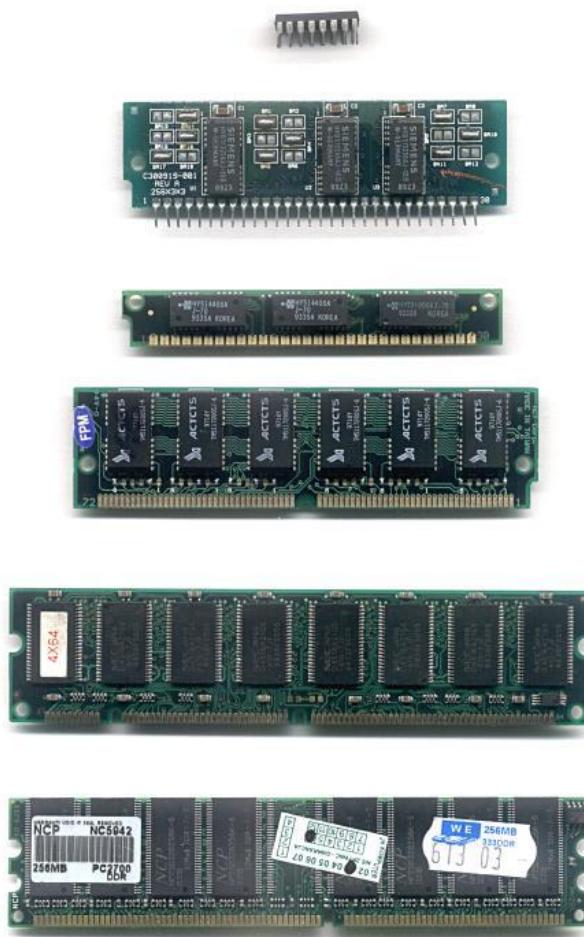


Fig. 4.6. Evoluția constructivă a memoriei operative.

4.1.2. Memoria externă a calculatorului

Memoria externă a calculatorului este realizată sub formă de dispozitive externe cu diferite principii de stocare a informațiilor. Dispozitivele externe de stocare a informației sunt utilizate pentru stocarea pe termen lung a informațiilor de orice fel și se caracterizează, în

comparație cu microcircuitele RAM, printr-o cantitate mare de memorie și viteză redusă de transfer. Memoria externă a unui calculator înseamnă, de obicei, atât dispozitive pentru citirea / scrierea informațiilor - dispozitive de stocare, cât și dispozitive în care informațiile sunt stocate direct - purtători de informații.

În calculatoarele personale, dispozitivele de memorie externă includ:

- unități de disc concepute pentru citirea / scrierea informațiilor pe discuri magnetice;
- unități de disc pentru lucrul cu discuri laser (optice);
- streamere concepute pentru a citi / scrie informații pe benzi magnetice;
- unități magneto-optice pentru lucrul cu discuri magneto-optice;
- dispozitive de memorie nevolatile (memorie flash).

Parametrii principali ale dispozitivelor externe de stocare a informației sunt:

- capacitatea (volumul) - cantitatea maximă de informații (volumul de date) care poate fi scrisă pe suport;
- timpul de acces la informații;
- rata de transfer a datelor.

Capacitatea memoriei externe este de sute și mii de ori mai mare decât capacitatea memoriei interne RAM, sau chiar nelimitată atunci când vine vorba de unități cu suporturi amovibile (сменные носители). Dar accesul la memoria externă necesită mult mai mult timp, deoarece viteza de transfer a memoriei externe este semnificativ mai mică decât a memoriei RAM.

4.2. Unitatea de memorie permanentă

Unitatea de memorie permanentă, de obicei, include un decoder, celule de memorare și un coder. În calitate de cellule de memorare poate servi o matrice din tranzistori sau diode (vezi fig. 4.7).

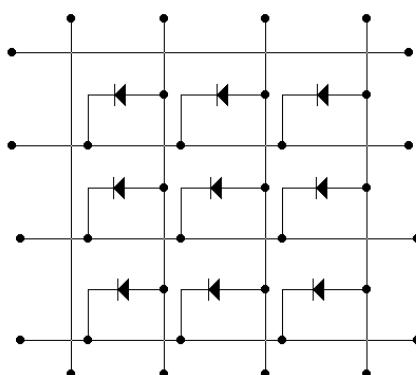


Fig. 4.7. Matricea cu diode.

În procesul înscrierii informației, drept exemplu pentru microcircuitele de tip PROM, unele diode sunt scurtcircuitate. Înscrierea informației în microcircuite de tip PROM se face cu dispozitive speciale numite „programatoare”.

Pentru exemplificare vom construi o unitate de memorie permanentă în care v-a fi stocată informația prezentată în tabelul 4.1.

Tabelul 4.1. Date pentru stocare în unitatea de memorie permanentă

Nr. d/o	Locația memoriei	Adresa, $A_2A_1A_0$	Date, $D_3D_2D_1D_0$
0	M_0	000	0100
1	M_1	001	0111
2	M_2	010	1010
3	M_3	011	1101
4	M_4	100	0010
5	M_5	101	1011
6	M_6	110	0111
7	M_7	111	0100

Conform tabelului 4.1 unitatea de memorie permanentă trebuie să includă următoarele componente: un decoder din 3 în 8; 8 celule de memorie; un coder din 8 în 4 (vezi fig. 4.8).

Schema unității de memorie permanentă este prezentată în fig. 4.8. În schemă sunt următoarele notări:

- $+V_{cc}$ – sursa de tensiune;
- [2], [1], [0] – comutatoarele magistralei adreselor $A_2A_1A_0$;
- $M_7, M_6, M_5, M_4, M_3, M_2, M_1, M_0$ – celule de memorie;
- D_3, D_2, D_1, D_0 – indicatoarele stărilor canalelor magistralei de date.

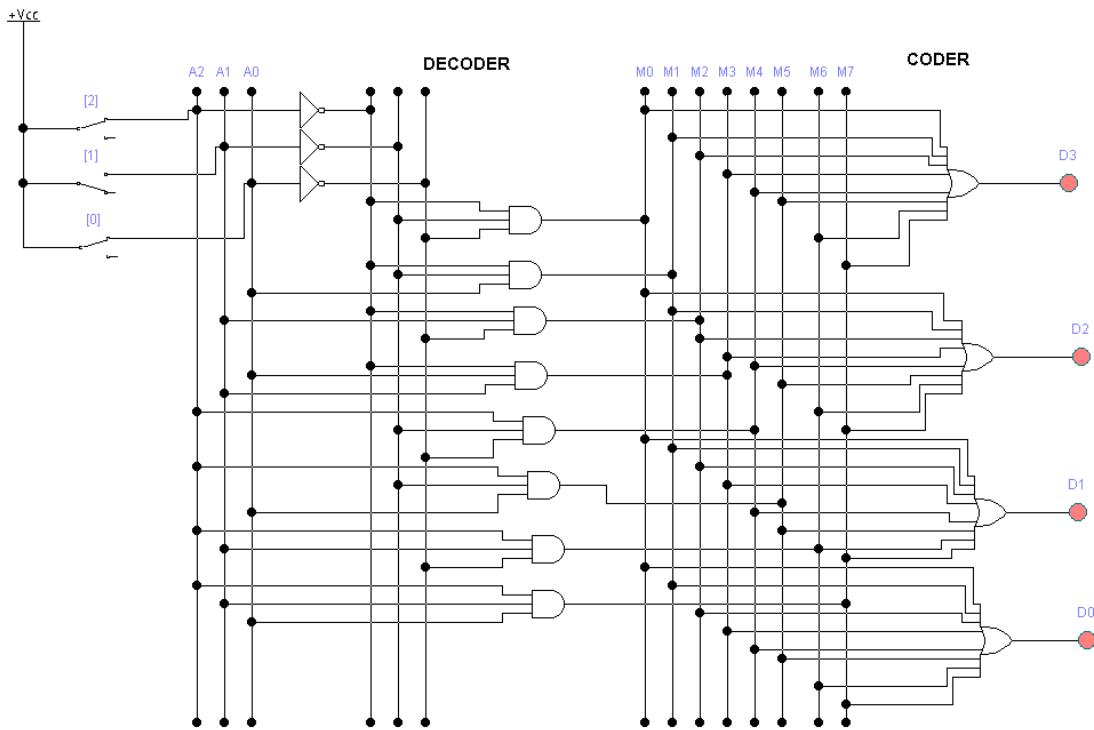


Fig. 4.8. Unitatea de memorie permanentă fără informație.

Pentru a stoca datele în shema unității de memorie permanentă sunt scoase conexiunile electrice dintre elementele SAU și celulele de memorie M₀, M₁, M₂, M₃, M₄, M₅, M₆, M₇. Drept exemplu, pentru canalul D₃ în tabelul 4.1 sunt următoarele date – 00110100 (vezi prima coloană din stînga a compartimentului Date). În acest caz sunt eliminate conexiunile electrice dintre elementul SAU al canalului D₃ și celulele de memorie M₀, M₁, M₄, M₆, M₇ sau sunt eliminate conexiunile electrice pentru care D₃ = 0 (vezi fig. 4.9).

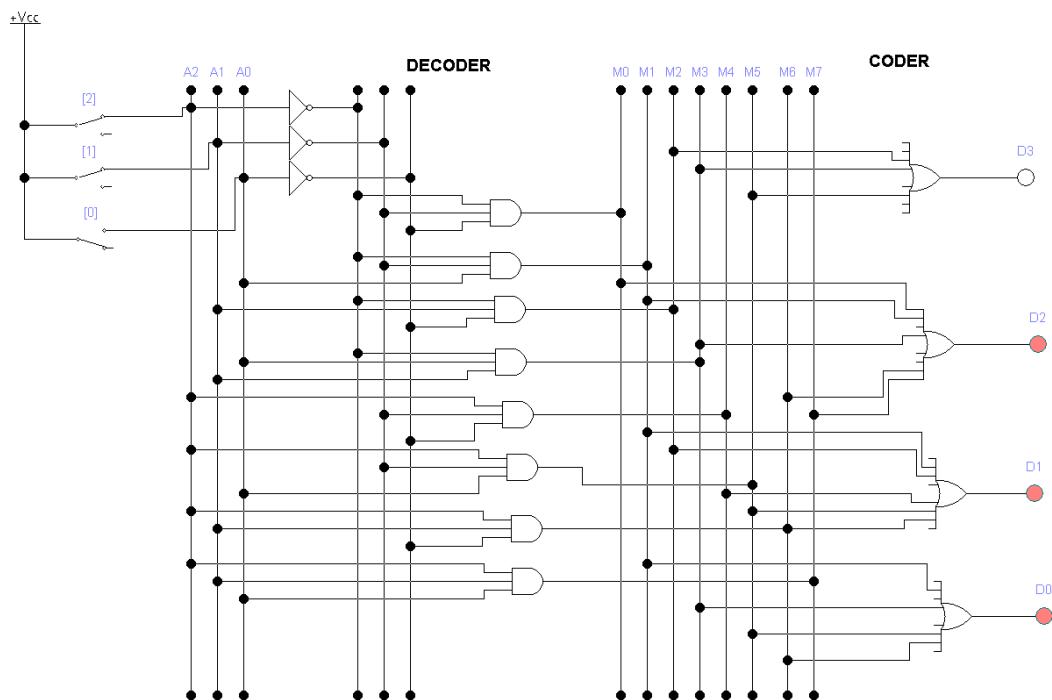


Fig. 4.9. Unitatea de memorie permanentă cu informație.

Simbolul convențional al microcircuitului memoriei permanente este prezentat în figura 4.10, unde k este numărul de canale a magistralei de adrese $A_{k-1} \dots A_1 A_0$, n - este numărul de canale a magistralei de date $D_{n-1} \dots D_1 D_0$, CS (chip select) se utilizează pentru conectarea schemei microcircuitului la magistrale, iar 1024×32 reprezintă numărul de cuvinte × lungimea cuvântului sau volumul memoriei.

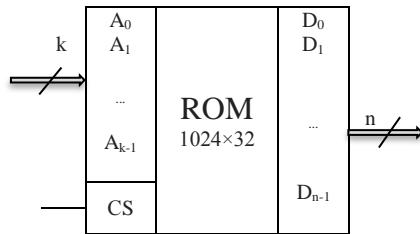


Fig. 4.10. Simbolul convențional al microcircuitului memoriei permanente.

4.3. Unitatea de memorie SRAM

Schema unității de memorie cu acces aleatoriu (SRAM) formată din decoder și patru registre de ordinul doi este prezentată în fig. 4.11. În schemă sunt următoarele notări:

- $+V_{cc}$ – sursa de tensiune;
- 1 V – comutatoare dirijate de tensiune;
- $[1], [0]$ – comutatoarele magistralei adreselor $A_1 A_0$;
- $\text{SI}0, \text{SI}1, \text{SI}2, \text{SI}3$ – elemente SI ;
- K_{in} – comutatorul utilizat pentru conectarea intrărilor tuturor registrelor la magistrala de date D ;
- K_{ies} – comutatorul utilizat pentru conectarea ieșirilor tuturor registrelor la magistrala de date Q ;
- $\text{RG}3, \text{RG}2, \text{RG}1, \text{RG}0$ – registre de tip consecutiv cu deplasare directă;
- K_0, K_0^* – comutatoarele utilizate pentru conectarea registrului $\text{RG}0$ la magistralele de date D și Q , respectiv;
- K_1, K_1^* – comutatoarele utilizate pentru conectarea registrului $\text{RG}1$ la magistralele de date D și Q , respectiv;
- K_2, K_2^* – comutatoarele utilizate pentru conectarea registrului $\text{RG}2$ la magistralele de date D și Q , respectiv;
- K_3, K_3^* – comutatoarele utilizate pentru conectarea registrului $\text{RG}3$ la magistralele de date D și Q , respectiv;
- $[\text{C}], [\text{D}], [\text{E}], [\text{F}]$ – comutatoarele canalelor de comandă ale registrelor;

- D – magistrala de date (intrarea);
- Q – magistrala de date (iesirea);
- [2] – comutatorul de comutare a regimurilor citire/înscriere R/W;
- [3] – comutatorul CS necesar pentru conectarea registrelor la magistralele de date.

Principiul de lucru al schemei. Schema unității de memorie SRAM, prezentată în fig. 4.11, se poate afla în următoarele regimuri de lucru:

- înscrierea informației primită prin magistrala D;
- păstrarea informației;
- citirea informației sau transmiterea informației pe magistrala Q.

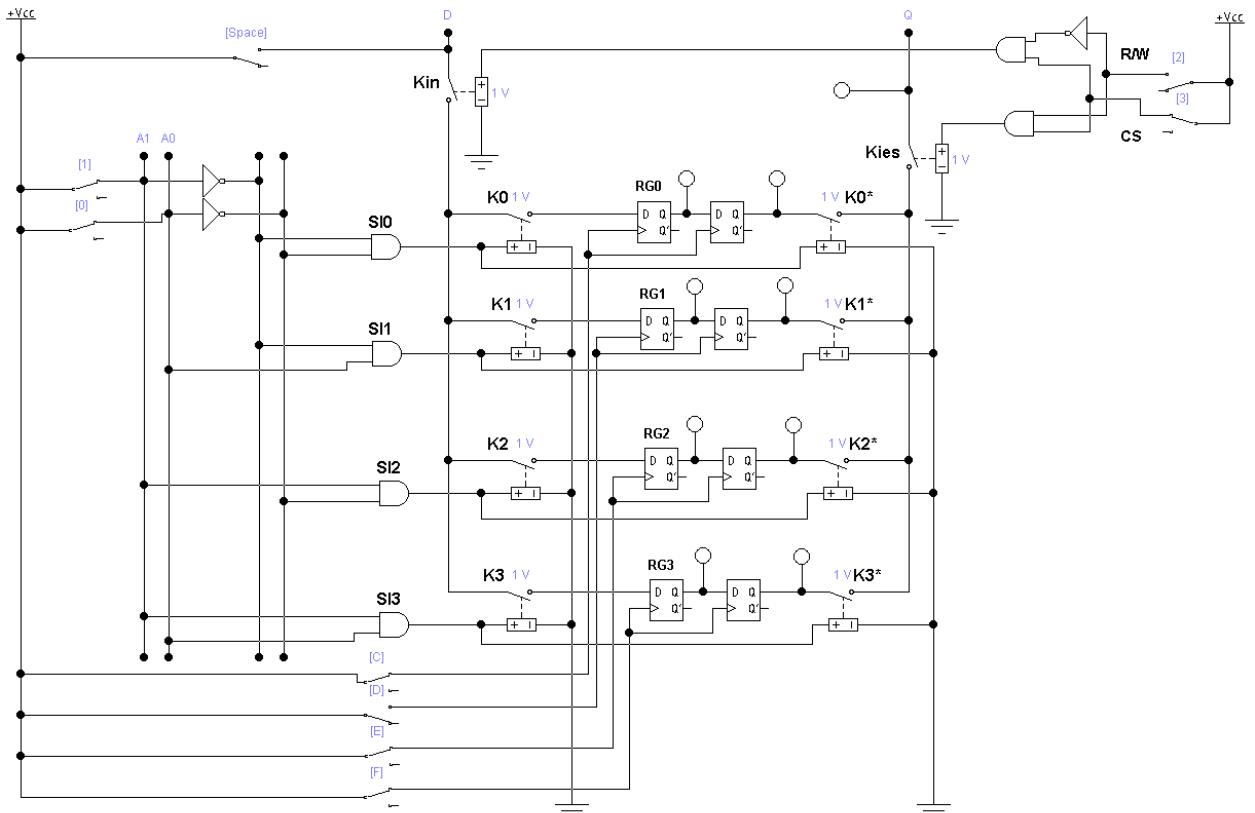


Fig. 4.11. Schema unității de memorie cu acces aleatoriu (SRAM).

Pentru a înscri o informație în registre este necesar ca comutatorul CS să fie conectat ($CS = 1$), iar comutatorul R/W deconectat ($R/W = 0$). Drept rezultat comutatorul Kin devine în stare conectat, iar comutatorul Kies în stare deconectat. Aplicând pe magistrala adreselor diferite combinații A1, A0 se alege registrul în care v-a fi înscrisă informația. Drept exemplu, dacă $A1 = 0$ și $A0 = 0$, atunci la magistrala de date D este cotectat registrul RG0, în care se poate înscri informația prin aplicarea semnalului de comandă cu ajutorul comutatorului [C].

Regimul de păstrare a informației se realizează dacă comutatorul CS este deconectat ($CS = 0$) și comutatoarele canalelor de comandă ale registrelor [C], [D], [E], [F] tot sunt deconectate.

Pentru citirea informației din registre sau transmiterea informației pe magistrala Q este necesar ca comutatoarele CS și R/W să fie conectate (CS = 1, (R/W = 1). Drept rezultat comutatorul K_{in} devine în stare deconectat, iar comutatorul K_{ies} în stare conectat. Aplicînd pe magistrala adreselor diferite combinații A₁, A₀ se alege registrul din care informația v-a fi transmisă pe magistrala Q. Drept exemplu, dacă A₁ = 1 și A₀ = 1, atunci la magistrala de date Q este cotectat registrul RG₃, din care se poate transmite informația prin aplicarea semnalului de comandă cu ajutorul comutatorului [F].

Adresarea regisrelor memoriei SRAM cu configurația 4×2 este prezentată în tabelul 4.2.

Tabelul 4.2. Adresele regisrelor memoriei cu configurația 4×2

Nr. d/0	Adresa, A ₁ A ₀	Registrul
0	00	RG0
1	01	RG1
2	10	RG2
3	11	RG3

Magistralele de date D și Q pot fi conectate la o magistrală de date comună (bidirectională).

Simbolul convențional al microcircuitului memoriei cu acces aleatoriu (SRAM) este prezentat în figura 4.12, unde k este numărul de canale a magistralei de adrese A_{k-1} ... A₁A₀, n - este numărul de canale a magistralei de date D_{n-1} ... D₁D₀, CS (chip select) se utilizează pentru conectarea schemei microcircuitului la magistrale, R/W se utilizează pentru comutarea regimurilor citire/înscriere, iar 1024×32 reprezintă numărul de cuvinte × lungimea cuvîntului sau volumul memoriei.

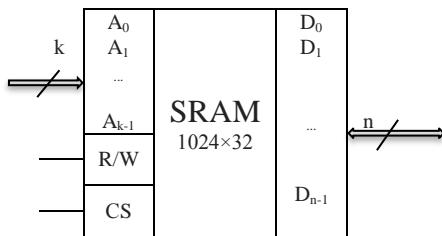


Fig. 4.12. Simbolul convențional al microcircuitului memoriei cu acces aleatoriu (SRAM).

4.4. Structura mono-, bi- și tridimensională a memoriei SRAM

Registrele într-un microcircuit de memorie SRAM pot fi aranjate în mod diferit, formînd următoarele structuri:

- monodimensională (în linie);

- bidimensională (matricială);
- tridimensională (în spațiu).

Tipul structurii determină numărul de elemente utilizate, cantitatea de energie electrică consumată, masa și gabaritele microcircuitului.

Structura monodimensională a memoriei SRAM. Un microcircuit de memorie SRAM cu structură monodimensională include un decoder, comutatoare și registre. În fig. 4.13 este prezentată schema electrică a microcircuitului de memorie SRAM cu structură monodimensională formată din decoder și patru registre de ordinul doi. În schemă sunt următoarele notări:

- $+V_{cc}$ – sursa de tensiune;
- 1 V – comutatoare dirijate de tensiune;
- [1], [0] – comutatoarele magistralei adreselor A_1A_0 ;
- SI_0, SI_1, SI_2, SI_3 – elemente $\bar{S}I$;
- K_{in} – comutatorul utilizat pentru conectarea intrărilor tuturor registrelor la magistrala de date D;
- K_{ies} – comutatorul utilizat pentru conectarea ieșirilor tuturor registrelor la magistrala de date Q;
- RG_3, RG_2, RG_1, RG_0 – registre de tip consecutiv cu deplasare directă;
- K_0, K_0^* – comutatoarele utilizate pentru conectarea registrului RG_0 la magistralele de date D și Q, respectiv;
- K_1, K_1^* – comutatoarele utilizate pentru conectarea registrului RG_1 la magistralele de date D și Q, respectiv;
- K_2, K_2^* – comutatoarele utilizate pentru conectarea registrului RG_2 la magistralele de date D și Q, respectiv;
- K_3, K_3^* – comutatoarele utilizate pentru conectarea registrului RG_3 la magistralele de date D și Q, respectiv;
- [C], [D], [E], [F] – comutatoarele canalelor de comandă ale registrelor;
- D – magistrala de date (intrarea);
- Q – magistrala de date (ieșirea);
- [2] – comutatorul de comutare a regimurilor citire/înscriere R/W;
- [3] – comutatorul CS necesar pentru conectarea registrului la magistralele de date.

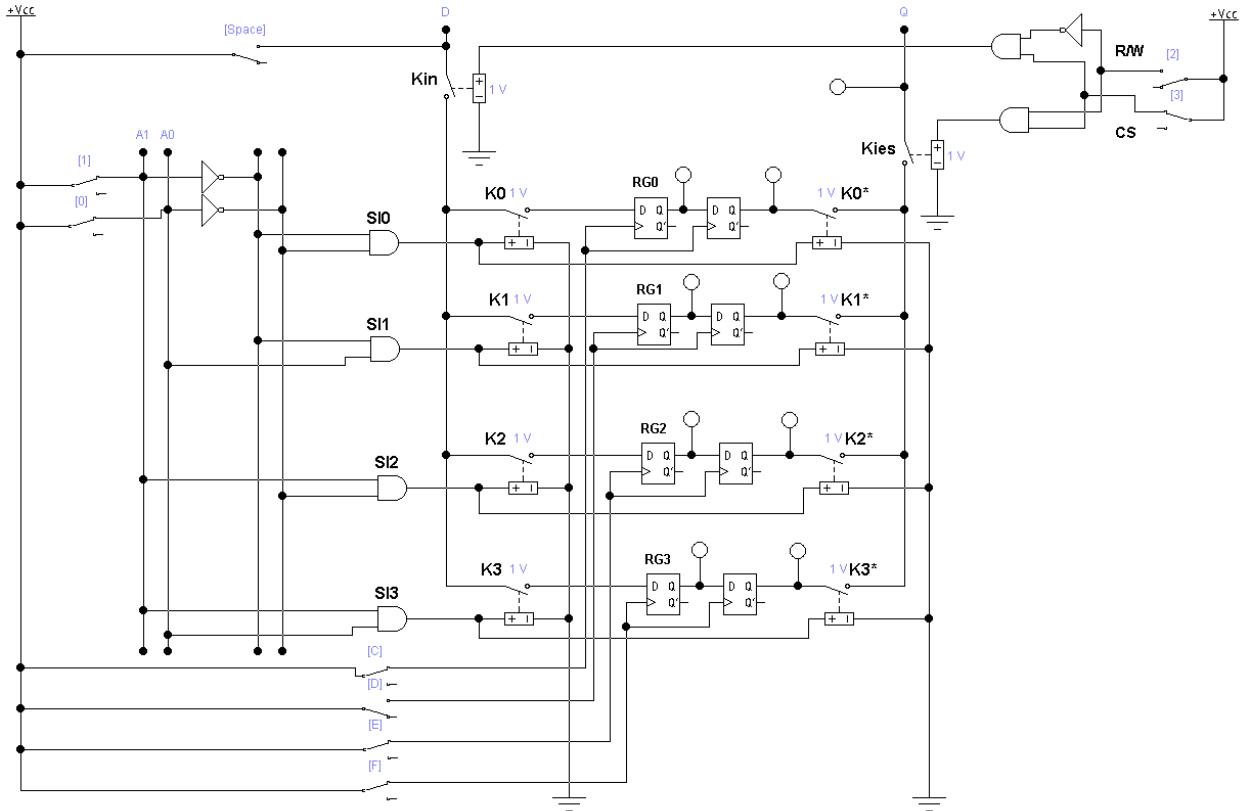


Fig. 4.13. Structura monodimensională a microcircuitului de memorie SRAM.

Adresarea registrelor memoriei cu structură monodimensională și configurație 4×2 este prezentată în tabelul 4.3.

Tabelul 4.3. Adresele registrelor memoriei cu structură monodimensională și configurație 4×2

Nr. d/0	Adresa, A ₁ A ₀	Registrul
0	00	RG0
1	01	RG1
2	10	RG2
3	11	RG3

Principiul de lucru al schemei memoriei SRAM cu structură monodimensională este prezentat în paragraful 4.3.

Structura bidimensională a memoriei SRAM. În fig. 4.14 este prezentată schema electrică a microcircuitului de memorie SRAM cu structură bidimensională (2D), formată din patru registre de ordinul doi. În schemă sunt următoarele notări:

- $+V_{cc}$ – sursa de tensiune;
- 1 V – comutatoare dirijate de tensiune;
- [1], [0] – comutatoare magistralei adreselor A₁A₀ ([0] – comutatorul utilizat pentru comutarea rândurilor de registre, [1] – comutatorul utilizat pentru comutarea coloanelor de registre);
- Kin – comutatorul utilizat pentru conectarea intrărilor tuturor registrelor la magistrala de date D;

- Kies – comutatorul utilizat pentru conectarea ieșirilor tuturor registrelor la magistrala de date Q;
- RG00, RG01, RG10, RG11 – registre de tip consecutiv cu deplasare directă;
- K00, K00* – comutatoarele utilizate pentru conectarea registrului RG00 la magistralele de date D și Q, respectiv;
- K01, K01* – comutatoarele utilizate pentru conectarea registrului RG01 la magistralele de date D și Q, respectiv;
- K10, K10* – comutatoarele utilizate pentru conectarea registrului RG10 la magistralele de date D și Q, respectiv;
- K11, K11* – comutatoarele utilizate pentru conectarea registrului RG11 la magistralele de date D și Q, respectiv;
- [C], [D], [E], [F] – comutatoarele canalelor de comandă ale registrelor;
- D – magistrala de date (intrarea);
- Q – magistrala de date (ieșirea);
- [2] – comutatorul de comutare a regimurilor citire/înscriere R/W;
- [3] – comutatorul CS necesar pentru conectarea registrului la magistralele de date.

Adresarea registrelor memoriei cu structură bidimensională și configurație 4×2 este prezentată în tabelul 4.4.

Tabelul 4.4. Adresele registrelor memoriei cu structură bidimensională și configurație 4×2

Nr. d/0	Adresa, A_1A_0	Registrul
0	00	RG00
1	01	RG01
2	10	RG10
3	11	RG11

Principiul de lucru al schemei cu structura bidimensională. Schema unității de memorie SRAM, prezentată în fig. 4.14, se poate afla în următoarele regimuri de lucru:

- înscrierea informației primită prin magistrala D;
- păstrarea informației;
- citirea informației sau transmiterea informației pe magistrala Q.

Pentru a înscrie o informație în registre este necesar ca comutatorul CS să fie conectat ($CS = 1$), iar comutatorul R/W deconectat ($R/W = 0$). În rezultat comutatorul K_{in} devine în stare conectată, iar comutatorul K_{ies} în stare deconectată. Registrul în care va fi înscrișă informația se alege prin indicarea adresei A_1, A_0 , utilizând comutatoarele [1], [0], și aplicând semnalul de comandă cu ajutorul comutatoarelor [C], [D], [E], [F]. În exemplu, dacă $A_1 = 0$ și $A_0 = 0$,

(comutatoarele K00, K10 sunt conectate) magistrala de date D este cotateată la intrările registrelor RG00 și RG10. În acest caz informația poate fi înscrisă în registrul RG00, aplicând semnalul de comandă cu ajutorul comutatorului [C], sau în registrul RG10, aplicând semnalul de comandă cu ajutorul comutatorului [E].

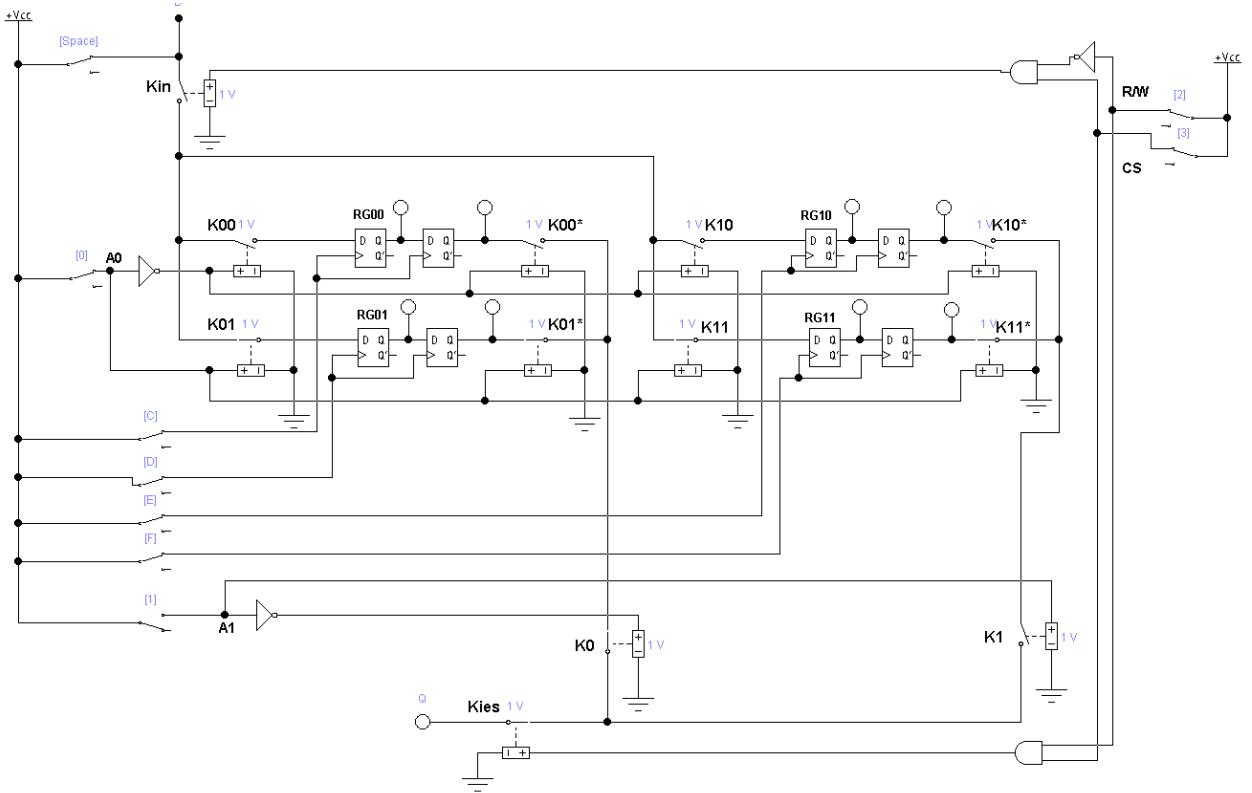


Fig. 4.14. Structura bidimensională a microcircuitului de memorie SRAM.

Regimul de păstrare a informației se realizează dacă comutatorul CS este deconectat ($CS = 0$) și comutatoarele canalelor de comandă ale registrelor [C], [D], [E], [F] tot sunt deconectate.

Pentru citirea informației din registre sau transmiterea informației pe magistrala Q este necesar ca comutatorul CS și R/W să fie conectate ($CS = 1$, $(R/W = 1)$). Drept rezultat comutatorul Kin devine în stare deconectat, iar comutatorul Kies în stare conectat. Registrul din care se citește informația se alege prin indicarea adresei A1,A0, utilizând comutatoarele [1], [0], și aplicând semnalul de comandă cu ajutorul comutatoarelor [C], [D], [E], [F]. Drept exemplu, dacă $A1 = 0$ și $A0 = 1$, atunci comutatoarele K00*, K10* și K1 sunt deconectate, iar comutatoarele K01*, K11* și K0 sunt conectate. În acest caz informația poate fi citită (transmisă pe magistrala de date Q) numai din registrul RG01 prin aplicarea semnalului de comandă cu ajutorul comutatorului [D].

Structura tridimensională a memoriei SRAM. Structura tridimensională (3D) a memoriei SRAM include trei decodere și matrice de registre plasate în spațiu. În fig. 4.15 este

prezentată schema convențională a microcircuitului de memorie SRAM cu structură tridimensională. În schemă sunt următoarele notări: A_3 , A_2 , A_1 – magistralele adreselor; n , k – numărul de canale ale magistralelor adreselor; M_n , ..., M_2 , M_1 – matrice cu registre; DC_x – decoder, utilizat pentru adresarea rândurilor de registre; DC_y – decoder, utilizat pentru adresarea coloanelor de registre; DC_z – decoder, utilizat pentru adresarea matricelor de registre.

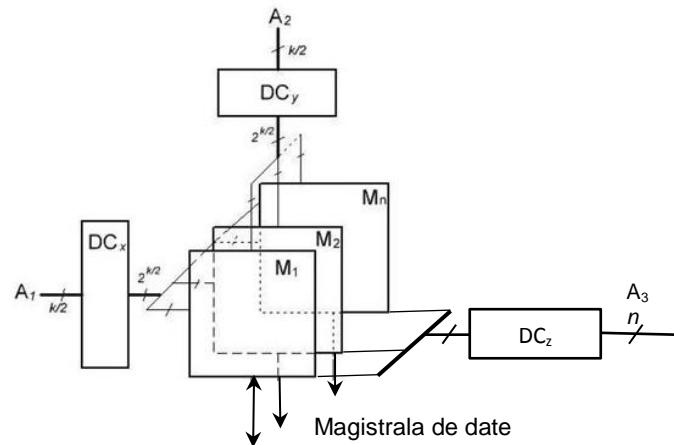


Fig. 4.15. Schema convențională a microcircuitului de memorie SRAM cu structură 3D.

4.5. Evoluția memoriei DRAM

Microcircuitele de memorie DRAM, folosite și în prezent, au fost inventate în 1966 de Robert Dennard de la Centrul de Cercetare Thomas Watson al companiei IBM. Primul microcircuit comercial de memorie a fost Intel 1103 cu un volum de 1 Ko, lansat spre vânzare în octombrie 1970.

Spre deosebire de microcircuitele memoriei SRAM care au o schemă electrică mai complexă, sunt mai scumpe dar mai rapide și sunt utilizate, în principal, în microprocesoare și în memoria Cache, microcircuitele DRAM sunt mai lente, dar mai ieftine și sunt construite pe baza condensatoarelor de capacitate mică. Astfel de condensatori își pierd rapid încărcătura, și, pentru a evita pierderea datelor stocate, condensatorii trebuie reîncărcați la intervale de timp regulate. Acest proces se numește regenerarea/reînoirea memoriei și este realizat de un controler special instalat fie pe placă de bază, fie în microprocesor.

În ultimii 30 de ani, dezvoltatorii au creat diferite tipuri de DRAM, folosind diferite soluții tehnice. Principala motră a dezvoltării microcircuitelor DRAM a fost dorința de a mări viteza de lucru și volumul memoriei operative.

PM DRAM (Page Mode DRAM) este una dintre primele memorii de tipul DRAM. Acest tip de memorie a fost produs la începutul anilor 1990.

FPM DRAM (Fast Page Mode DRAM) sunt microcircuite bazate pe PM DRAM și se caracterizează prin performanțe sporite. Acest tip de memorie a fost popular în prima jumătate a anilor 1990, iar în 1995 a ocupat 80% din piața de memorie pentru calculatoare. Microcircuitele FPM DRAM au fost utilizate în principal pentru calculatoare cu microprocesoare Intel 80486 sau procesoare similare de la alte companii. Funcționau la frecvențe de 25 și 33 MHz cu tempi de acces de 70 și 60 ns și cu un ciclu de regenerare de 40 și respectiv 35 ns. În 1996-1997 au fost înlocuite de EDO DRAM și SDR SDRAM. În 1997, cota de piață FPM DRAM a scăzut la 10%.

EDO DRAM (Extended Data Out DRAM – DRAM cu ieșire extinsă de date) este un tip de DRAM creat pentru a înlocui FPM DRAM datorită ineficienței FPM DRAM atunci când s-a trecut la procesoare Intel Pentium. Acest tip de memorie a apărut pe piață în 1996 și a fost folosit pe calculatoare cu procesoare Intel Pentium sau mai performante. A depășit performanțele FPM DRAM cu 10-15%. A funcționat la frecvențe de 40 și 50 MHz cu tempi de acces de 60 și 50 ns și cu un ciclu de regenerare de 25 ns și respectiv 20 ns.

SDR SDRAM (Single Data Rate Synchronous DRAM DRAM - DRAM sincron cu o singură frecvență) este un tip de DRAM creat pentru a înlocui EDO DRAM în legătură cu o scădere a stabilității EDO DRAM cu procesoare noi și o creștere a frecvențelor de operare ale magistralelor de sistem. Noile particularități ale acestui tip de memorie sunt utilizarea unui generator de semnale pentru sincronizarea proceselor înscriere/citire și prelucrarea informațiilor în torrent. Acest tip de memorie a funcționat în mod fiabil la frecvențele magistralei de sistem de 100 MHz și mai mari. Frecvențele de lucru ale microcircuitelor SDRAM sunt de 66, 100 sau 133 MHz, timpul de acces – de 40 și 30 ns, iar ciclul de regenerare – 10 și 7,5 ns.

Tehnologia VCM (Virtual Channel Memory) a fost utilizată împreună cu SDRAM. VCM utilizează o arhitectură de canale virtuale care permite un transfer de date mai flexibil. Această arhitectură este integrată în SDRAM. Utilizarea VCM a majorat rata de transfer a datelor. SDRAM-urile VCM și non-VCM au fost interoperabile, permitând modernizarea sistemelor fără costuri sau modificări semnificative.

ESDRAM (Enhanced SDRAM – SDRAM îmbunătățit) este un tip de DRAM conceput pentru a rezolva unele probleme legate de frânera semnalelor în microcircuitele DRAM standarde. În esență, microcircuitele SDRAM conțin memorii SRAM (10-15% din volumul total) care sunt folosite pentru a stoca și prelua datele cele mai frecvent utilizate, reducând astfel timpul de acces la datele stocate în DRAM lent. Memoria de acest tip funcționa la frecvențe de până la 200 MHz.

BEDO DRAM (Burst EDO DRAM - Bloc EDO RAM) este un tip de DRAM bazat pe EDO DRAM și care oferă suport pentru citirea datelor bloc cu bloc (un bloc de date este citit într-o perioadă a semnalului de tactare). Modulele de memorie de acest tip, datorită citirii bloc cu

bloc, funcționează mai repede decât SDRAM și au devenit o alternativă ieftină a memoriei SDRAM, dar datorită incapacității de a lucra la frecvențe de magistrală de sistem care depășesc 66 MHz, nu au devenit populare.

VRAM (Video RAM) este un tip de memorie DRAM dezvoltat pe baza SDRAM special pentru utilizarea pe plăci video. Memoria de acest tip, datorită unor modificări tehnice performante, a depășit SDRAM cu 25%. Acest tip de memorie permite furnizarea unui flux continuu de date în procesul de actualizării imaginii de înaltă calitate. VRAM a devenit baza tipului de memorie **WRAM** (Windows RAM), care este uneori asociată greșit cu sistemele de operare din familia Windows.

DDR SDRAM (Double Data Rate SDRAM – SDRAM cu viteză dublă de transfer a de datelor sau SDRAM II) este un tip de DRAM bazat pe SDR SDRAM și caracterizat printr-o rată de transfer de date dublă (lățime de bandă dublă). Acest tip de memorie a fost utilizat inițial în plăcile video, ulterior a început să fie folosit și în microcircuitele memoriei operative.

Frecvențele de funcționare ale memoriei SDRAM DDR sunt 100, 133, 166 și 200 MHz, iar timpul de acces este de 30 și 22,5 ns.

Deoarece frecvențele de lucru se situează în intervalul de la 100 la 200 MHz, iar datele sunt transmise câte 2 biți pe un impuls de tactare, atât după frontul anterior, cât și după frontul posterior al impulsului, frecvența efectivă de transmitere a datelor se află în intervalul de la 200 la 400 MHz. Modulele de memorie care funcționează la astfel de frecvențe sunt denumite „DDR200”, „DDR266”, „DDR333”, „DDR400”.

RDRAM (Rambus DRAM) este un tip de DRAM dezvoltat de compania Rambus. Memoria de acest tip s-a remarcat prin performanțe ridicate datorită unui număr de caracteristici care nu se regăsesc în memoriile altor tipuri. Această memorie funcționa la 400, 600 și 800 MHz cu timpi de acces până la 30 ns. Inițial, era foarte scump, motiv pentru care producătorii de calculatoare au preferat SDRAM DDR mai lent dar mai ieftin.

DDR2 SDRAM este un tip de memorie DRAM bazat pe DDR SDRAM și lansat în 2004. Memoria de acest tip, în comparație cu DDR SDRAM, datorită modificărilor tehnice, are performanțe mai mari. Acest tip de memorie funcționează la frecvențe de 200, 266, 333, 337, 400, 533, 575 și 600 MHz. În acest caz, frecvența efectivă de transmitere a datelor este 400, 533, 667, 675, 800, 1066, 1150 și 1200 MHz, iar timpii de acces sunt 25, 11,25, 9, 7,5 ns sau mai puțin.

Unii producători de module de memorie, pe lângă modulele care funcționează la frecvențe standard, au produs module care funcționează la frecvențe nestandard (intermediare); astfel de module au fost destinate utilizării în sistemele accelerate în care era necesară marja de frecvență.

DDR3 SDRAM este un tip de memorie DRAM bazat pe DDR2 SDRAM, care oferă o viteză dublă ratei de transfer de date pe magistrala de memorie și un consum redus de energie. Acest tip de memorie funcționează la frecvențe în intervalul 800...3000 MHz.

DDR4 SDRAM (DDR four SDRAM) este un tip de memorie SDRAM, lansat în 2014 și bazat pe tehnologiile generațiilor anterioare de DDR și caracterizat prin caracteristici de frecvență crescute și tensiune redusă de alimentare. Principala diferență între DDR4 și standardul anterior (DDR3) este că numărul blocurilor de date a fost dublat și a devenit egal cu 16, ceea ce a făcut posibilă creșterea vitezei de transfer a datelor. Viteza de transfer a datelor poate atinge mărimea de 25,6 GB/s. Fiabilitatea DDR4 a fost sporită prin introducerea unui mecanism de control al parității pe magistrala adreselor și magistrala de comandă. Memoria DDR4 funcționează la frecvențele 1600, 2400 și 3200 MHz.

GDDR (Graphics Double Data Rate) este un tip de memorie RAM concepută pentru plăci video și bazată pe tehnologia DDR (vezi fig. 4.16). GDDR este destinată să înlocuiască VRAM (RAM video) învechit. În prezent sunt dezvoltate cinci generații – GDDR2, GDDR3, GDDR4, GDDR5 și GDDR6. Microcircuitele GDDR6 au apărut în anul 2017. Acest tip de memorie este utilizat în plăcile video pentru stocarea temporară a informațiilor necesare pentru construirea și afișarea datelor pe ecranul monitorului conform descrierii sale în baza de date. Drept exemplu, GDDR este folosit pentru a stoca textura unei imagini, ca microprocesorul să nu apeleze la memoria operativă, deoarece astfel de apeluri durează mai mult timp.



Fig. 4.16. Microcircuit de memorie GDDR.

Deși microcircuitele DRAM, în prezent, sunt cele mai populare și sunt utilizate pe larg în calculatoare, experții prezic dispariția lor. Motivul principal este imposibilitatea de a reduce dimensiunea celulei de memorie. Printre cei mai probabili succesorii ai DRAM se numește

„memoria cu corp plutitor” (FBM), a cărei avantaj principal este absența unui condensator în celula de memorie.

4.6. Realizarea constructivă a memoriei DRAM

Constructiv memoria DRAM este realizată atât sub formă de microcircuite separate (DIP, SOIC, BGA), cât și sub formă de module de memorie (SIPP, SIMM, DIMM, RIMM).

Microcircuite separate de memorie DRAM. Până în prezent au fost produse microcircuite DRAM în corpuri de tip DIP, SOIC și BGA.

Inițial, microcircuitele de memorie au fost produse în corpuri de tip **DIP** (Dual In-line Package, vezi fig. 4.17), ulterior au început să fie produse în corpuri care au fost mai avansate din punct de vedere tehnologic pentru utilizare în module de memorie.

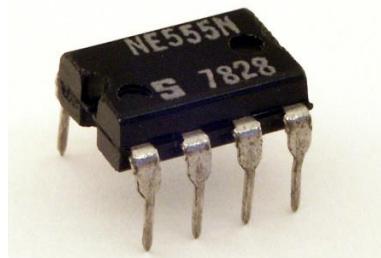


Fig. 4.17. Microcircuit DRAM în corp de tip DIP.

SOIC (Small-Outline Integrated Circuit) este un tip de corp de microcircuit conceput pentru montarea pe suprafață (vezi fig. 4.18). Are forma unui dreptunghi cu două rânduri de pinuri pe laturile lungi. Microcircuitele dintr-un corp SOIC ocupă cu 30-50% mai puțină suprafață decât omologii lor dintr-un corp DIP și, de obicei, sunt cu 70% mai subțiri. De regulă, numerotarea pinilor microcircuitelor DIP și SOIC sunt identice.

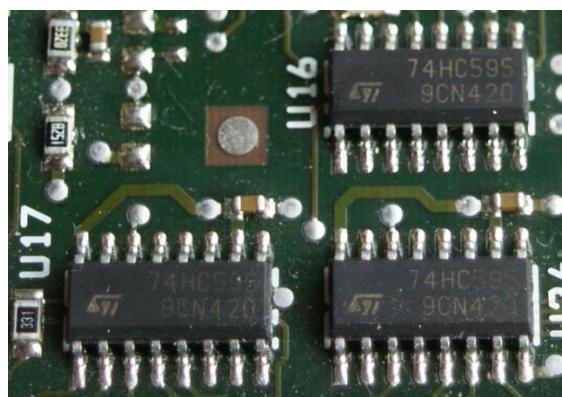


Fig. 4.18. Microcircuit DRAM în corp de tip SOIC.

BGA (Ball grid array) este un tip de corp pentru circuite integrate montate la suprafață (vezi fig. 4.19). Pinii BGA sunt niște bile de lipit aplicate pe suprafetele de contact de pe spatele microcircuitului. Microcircuitul este amplasat pe placă de montaj, conform marcajului primului contact de pe microcircuit și de pe placă. Microcircuitul este apoi încălzit folosind o stație de lipit sau o sursă cu infraroșu, astfel încât bilele să se topească. Tensiunea superficială a bilelor topite ține microcircuitul exact în locul în care ar trebui să fie pe placă de montaj și împiedică deformarea bilelor.

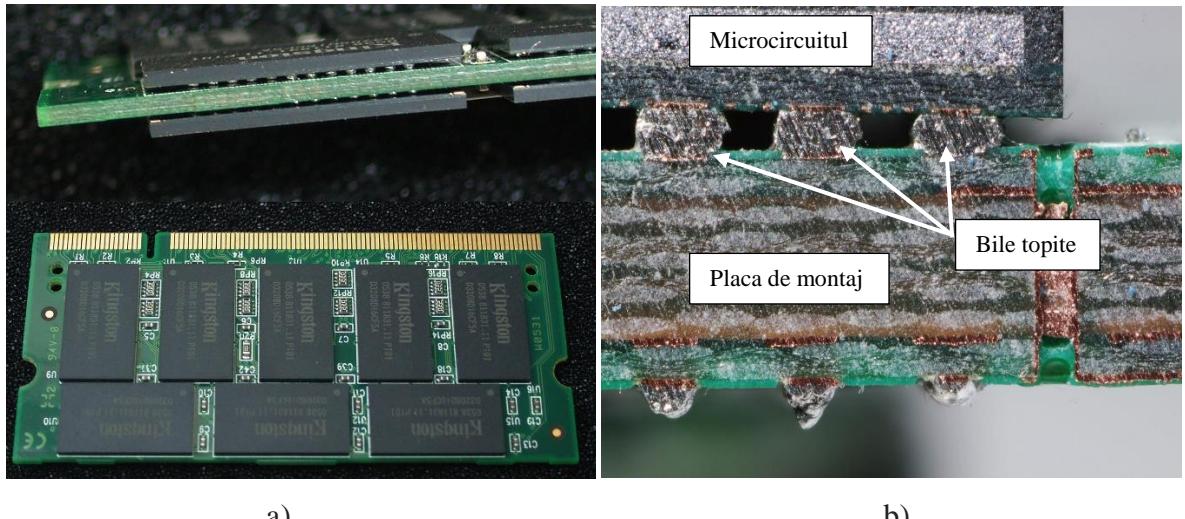


Fig. 4.19. Microcircuite DRAM în corp de tip BGA: a) – microcircuite cu pini BGA instalate pe placă; b) – secțiunea transversală a unei plăci de circuite cu un corp BGA.

Module de memorie DRAM. La momentul dat sunt cunoscute următoarele tipuri de module de memorie DRAM: SIPP; SIMM; DIMM; RIMM.

SIPP (Single In-line Pin Package) - module de memorie cu un singur aranjament în linie de pini (vezi fig. 4.20). Modulul este format dintr-o mică placă de microcircuite cu 30 de pini pe un rând, care sunt instalati în orificiile corespunzătoare de pe placă de bază a calculatorului.

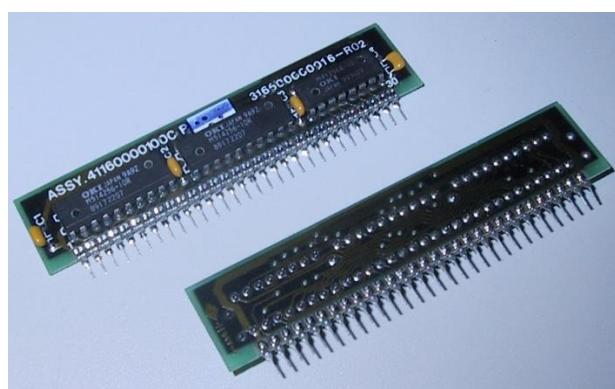


Fig. 4.20. Modul de memorie SIPP.

Acstea module de memorie au fost utilizate în calculatoare cu microprocesoare Intel 80286 și unele 80386. Ulterior au fost înlocuite cu SIMM-uri, care s-au dovedit a fi mai ușor de instalat.

SIMM (Single In-line Memory Module) este numele modulelor de memorie cu un singur aranjament în linie de contacte, care au fost utilizate pe scară largă în calculatoarele din anii '90. Modulele SIMM au avut mai multe modificări, cu 30, 68 și 72 de contacte (vezi fig. 4.21).



Fig. 4.21. Module de memorie SIMM cu 30, 68 și 72 de contacte.

SIMM-urile au fost dezvoltate și brevetate în 1983 de compania Wang Laboratories. Modulele erau inițial ceramice și aveau știfturi. Inițial SIMM-urile erau instalate în sloturi care nu aveau mecanisme de blocare, apoi au fost elaborate sloturile ZIF cu mechanism de blocare.

Primele SIMM-uri aveau 30 de contacte și un volum de memorie cuprins între 64 Ko și 16 Mo și o magistrală de date pe opt biți. Aceste module de memorie au fost utilizate în calculatoare cu microprocesoare Intel 8088, 80286, 80386.

Pentru calculatoarele cu microprocesoare Intel 80486 și Pentium au fost elaborate SIMM-uri cu 72 de contacte. Modulele cu 72 de contacte constau în esență din patru module cu 30 de contacte și aveau un volum de memorie de la 1 Mo până la 64 Mo.

DIMM (Dual In-line Memory Module – modul de memorie față-verso) este numele modulelor de memorie care au contactele situate pe ambele suprafete a plăcii de montaj. Microcircuitele de memorie pot fi plasate pe una sau pe ambele părți ale plăcii de montaj. Există și module SO-DIMM (Small Outline DIMM) concepute pentru utilizare pe laptopuri.

Modulele de tip DIMM sunt construite din microcircuite de memorie DDR, DDR2, DDR3 și DDR4.

În fig. 4.22 sunt prezentate imaginele schematicice ale modulelor DIMM și SO-DIMM.

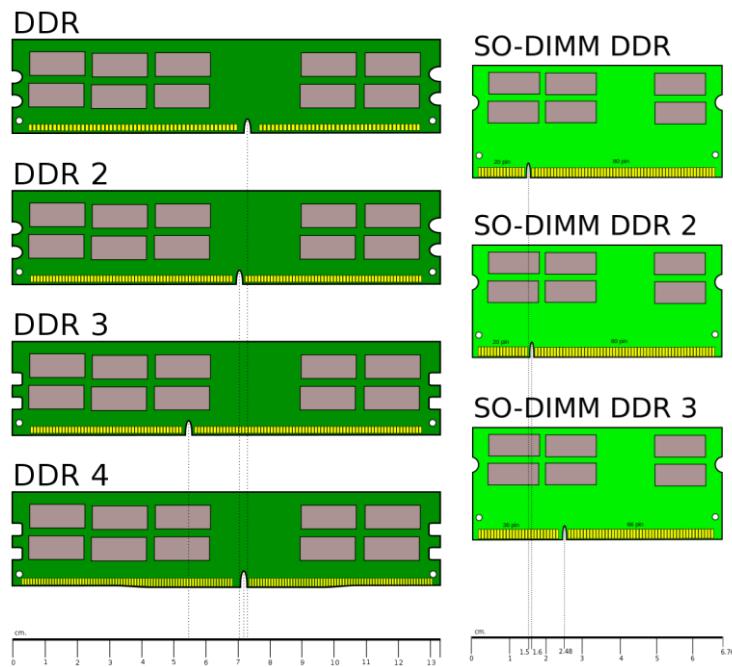


Fig. 4.22. Imaginele schematicice ale modulelor DIMM și SO-DIMM.

Există următoarele tipuri de module DIMM:

- 72-contacte SO-DIMM (nu este compatibil cu 72-contacte SIMM) – format din FPM DRAM și EDO DRAM;
- 100-contacte DIMM – format din SDRAM;
- 144-contacte SO-DIMM – format din SDR SDRAM;
- 168-contacte DIMM – format din SDR SDRAM;
- 172-contacte MicroDIMM – format din DDR SDRAM;
- 184-contacte DIMM – format din DDR SDRAM;
- 200-contacte SO-DIMM – format din DDR SDRAM și DDR2 SDRAM;
- 204-contacte SO-DIMM – format din DDR3 SDRAM;
- 214-contacte MicroDIMM – format din DDR2 SDRAM;
- 240-contacte DIMM – format din DDR2 SDRAM și DDR3 SDRAM;
- 260-contacte SO-DIMM – format din DDR4 SDRAM;
- 288-contacte DIMM – format din DDR4 SDRAM.

Trebuie remarcat faptul, că modulele cu standarde diferite sunt incompatibile între ele, atât mecanic, cât și electric. Pentru a proteja utilizatorii de posibile erori, toate cele patru standarde DDR SDRAM au decupaje/crestături pe placă DIMM situată în locuri diferite (vezi fig. 4.22). Prin urmare, devine imposibilă instalarea unui modul DIMM într-un slot de alt standard. SDRAM DIMM avea chiar și două decupaje/crestături.

Configurarea maximă a memoriei operative a calculatorului depinde de numărul de sloturi DIMM de pe placă de bază și de capacitatea controlerului de memorie pe care îl acceptă. Cu fiecare generație de memorie DDR, capacitatea maximă a modulelor a crescut, de asemenea, a crescut și volumul memoriei suportată de controlere. Pentru prima generație DDR SDRAM un modul avea maximum 1 Go, pentru a doua generație DDR a fost obținut un volum de 4 Go pe modul. Cu standardul DDR3 SDRAM volumul memoriei modulului DIMM a crescut la 8 Go, iar cu standardul DDR4 SDRAM la 16 Go per modul, deși ulterior au apărut module de 32 Go și 64 Go. În funcție de microprocesor, pentru module formate din DDR4 se utilizează frecvențe efective cu mărimea de 2666 MHz, 2933 MHz sau 3 200 MHz.

RIMM (Rambus In-line Memory Module). Module de memorie utilizate în perechi, construite din RDRAM cu 168 sau 184 contacte. Datorită caracteristicilor de proiectare, acestea trebuie să fie instalate pe plăcile de bază ale calculatorului numai în perechi, în caz contrar, trebuie instalate module fictive speciale în sloturi goale. Așa module sunt rar utilizate.

4.7. Schema electrică și principiul de funcționare ale elementelor memoriei DRAM

Elementul principal al celulei de memorie DRAM este condensatorul care în stare încărcată cu sarcină electrică păstrează valoarea logică 1, iar în stare descărcată păstrează valoarea logică 0. În fig. 4.23 este prezentată schema electrică a unei celule de memorie DRAM cu următoarele notări:

- D – magistrala de date la intrarea celulei de memorie;
- Q – magistrala de date la ieșirea celulei de memorie;
- R/W – canalul de comandă „citire/înscriere”;
- Ain – amplificator-comutator la intrarea celulei de memorie;
- Aies – amplificator-comutator la ieșirea celulei de memorie;
- C – condensatorul.

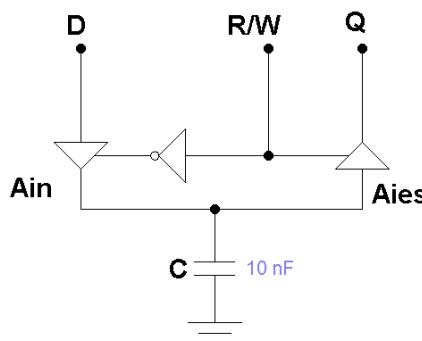


Fig. 4.23. Schema electrică a unei celule de memorie DRAM.

Celula de memorie este dirijată de semnalul electric transmis prin canalul R/W și poate fi plasată în regim de înscriere a informației sau în regim de citire a informației.

Regimul de înscriere a informației este realizat dacă $R/W = 0$. În acest caz amplificatorul-comutator A_{ies} la ieșirea celulei de memorie deconectează condensatorul C de la magistrala Q, iar amplificatorul-comutator A_{in} la intrarea celulei de memorie conectează condensatorul C la magistrala D deoarece inversorul transformă $R/W = 0$ în $R/W = 1$. Dacă pe magistrala D este tensiune (logic 1), atunci condensatorul C se încarcă sau se înscrive valoarea logică 1. Dacă pe magistrala D nu este tensiune (logic 0), atunci condensatorul C nu se încarcă sau se înscrive valoarea logică 0.

Regimul de citire a informației este realizat dacă $R/W = 1$. În acest caz amplificatorul-comutator A_{ies} la ieșirea celulei de memorie conectează condensatorul C la magistrala Q, iar amplificatorul-comutator A_{in} la intrarea celulei de memorie deconectează condensatorul C de la magistrala D deoarece inversorul transformă $R/W = 1$ în $R/W = 0$. Dacă pe condensatorul C este sarcină electrică (logic 1), atunci pe magistrala Q apare tensiune (logică 1). Dacă pe condensatorul C nu este sarcină electrică (logic 0), atunci pe magistrala Q nu apare tensiune (logic 0).

În fig. 4.24 este prezentată schema electrică a celulei de memorie DRAM care permite înscrierea și citirea informației. Semnalele de la generator sunt aplicate prin bistabil la magistrala de date D și direct pe canalul de dirijare R/W. Pe ecranul analizatorului logic sunt prezentate următoarele diagrame temporare (în ordinea de sus în jos): a magistralei de date D; a celulei de memorie (condensatorului) C; a canalului de comandă R/W; a magistralei de date Q. Linia verticală roșie indică momentul de timp de transmitere din condesator pe magistrala Q a valorii logice 1, iar linia verticală albastră indică momentul de timp de transmitere din condesator pe magistrala Q a valorii logice 0.

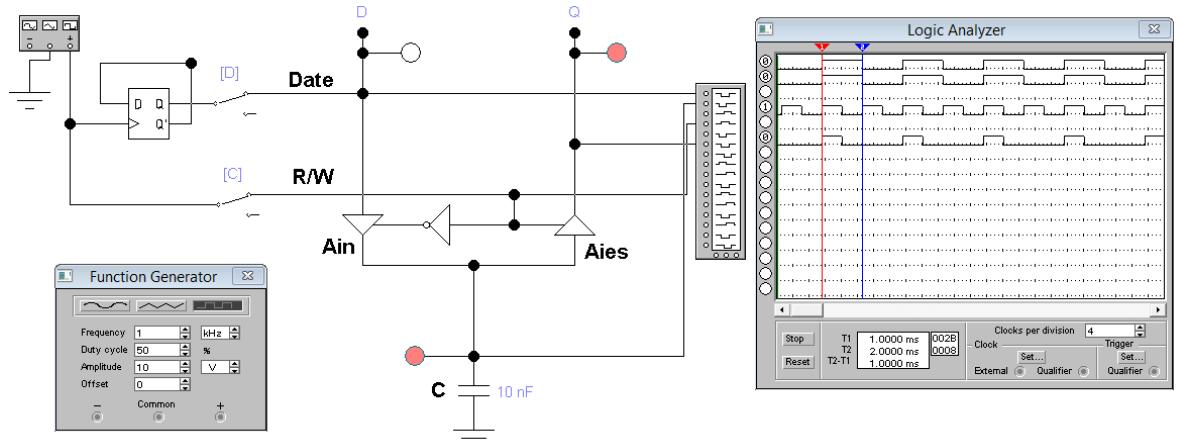


Fig. 4.24. Demonstrarea procesului de lucru a celulei de memorie DRAM.

Celulele de memorie DRAM se utilizează, în special, pentru construirea registrelor și microcircuitelor de memorie DRAM. În fig. 4.25 este prezentată schema electrică a memoriei DRAM formată din patru registre de ordinul patru – RG3, RG2, RG1 și RG0. În schemă sunt următoarele notări:

- $+V_{cc}$ – sursa de tensiune;
- 1 V – comutatoare dirijate de tensiune;
- [1], [0] – comutatoarele magistralei adreselor A₁A₀;
- SI0, SI1, SI2, SI3 – elementele řI ale decoderului;
- RG3, RG2, RG1, RG0 – registre de tip paralel;
- D₃, D₂, D₁, D₀ – magistrala de date la intrarea registrelor;
- Q₃, Q₂, Q₁, Q₀ – magistrala de date la ieșirea registrelor;
- R/W – canalul de comandă „citire/înscriere”.

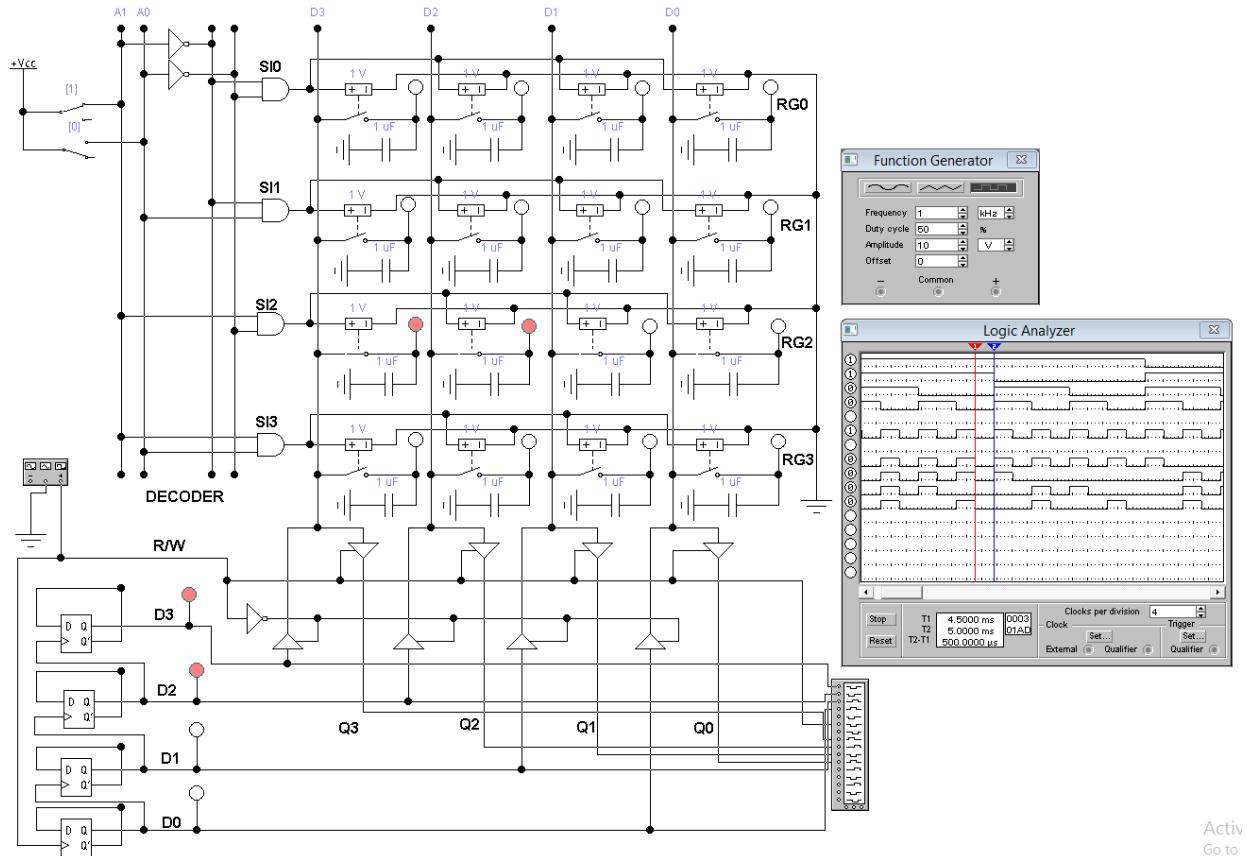


Fig. 4.25. Memorie DRAM cu structura 4×4.

Decoderul din schemă (vezi fig. 4.25) servește pentru comutarea registrelor la magistrala de date D₃, D₂, D₁, D₀. Comutarea registrelor la magistrala de date se face cu ajutorul comutatoarelor dirijate de tensiune. Semnalele de la generator sunt aplicate prin bistabili la magistrala de date D₃, D₂, D₁, D₀, pentru a fi înschise în registre, și direct pe canalul de comandă R/W. Adresele A₁A₀ registrelor memoriei DRAM sunt determinate de comutatoarele [1], [0] (vezi tabelul 4.5).

Tabelul 4.5. Adresele registrelor memoriei DRAM

Nr. d/0	Adresa, A ₁ A ₀	Registrul
0	00	RG0
1	01	RG1
2	10	RG2
3	11	RG3

Pe ecranul analizatorului logic sunt prezentate următoarele diagrame temporare (în ordinea de sus în jos): ale canalelor magistralei de date D₃, D₂, D₁, D₀ la intrările registrelor; a canalului de dirijare R/W; ale canalelor magistralei de date Q₃, Q₂, Q₁, Q₀ la ieșirile registrelor.

Linia verticală roșie de pe ecranul analizatorului logic indică momentul de timp de înscriere a informației D₃D₂D₁D₀ = 1100₂ în registrul RG2, iar linia verticală albastră indică momentul de timp de transmitere din registrul RG2 pe magistrala Q₃Q₂Q₁Q₀ a informației 1100₂.

4.8. Configurarea modulelor din microcircuite ale memoriei DRAM

În prezent microcircuitele memoriei operative sunt confecționate conform standardelor care determină numărul și ordinul cuvintelor. În cazul, când sunt necesare alte configurații ale memoriei operative, microcircuitele standarde pot fi conectate în paralel, consecutiv sau paralel/consecutiv (combinat) cu scopul majorării numărului de cuvinte, ordinului/lungimii cuvintelor sau și numărului de cuvinte și ordinului/lungimii cuvintelor.

Pentru exemplificare v-or fi prezentate două metode de configurare a modulelor din microcircuite DRAM. Fie, sunt date două microcircuite DRAM cu structura 8×4 (8 cuvinte de ordinul 4, vezi fig. 4.26).

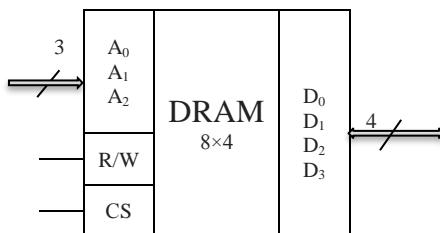


Fig. 4.26. Simbolul convențional al microcircuitului memoriei DRAM cu structura 8×4.

Din două microcircuite DRAM cu structura 8×4 pot fi formate/construite următoarele configurații ale modulelor:

- 8×8 (8 cuvinte de ordinul 8);

- 16×4 (16 cuvinte de ordinul 4).

Configurația 8×8 . Pentru a forma modulul cu configurația 8×8 se procedează astfel:

- 1) se conectează intrările CS ale microcircuitelor DRAM0 și DRAM1, deoarece microcircuitele trebuie să funcționeze sincron;
- 2) se conectează împreună intrările R/W ale microcircuitelor DRAM0 și DRAM1, deoarece operațiile de înscrriere/citire trebuie efectuat în/din ambele microcircuite;
- 3) se conectează împreună canalele de ordinul respectiv a magistralelor adreselor A₂A₁A₀ microcircuitelor DRAM0 și DRAM1, deoarece configurația formată are 8 cuvinte (8 registre);
- 4) canalele magistralei de date a microcircuitului SRAM1 se redenumesc pentru a forma magistrala comună de date D₇D₆D₅D₄D₃D₂D₁D₀.

Schema electrică a modulului de memorie cu configurația 8×8 , formată din două microcircuite DRAM cu structura 8×4 , este prezentată în fig. 4.27.

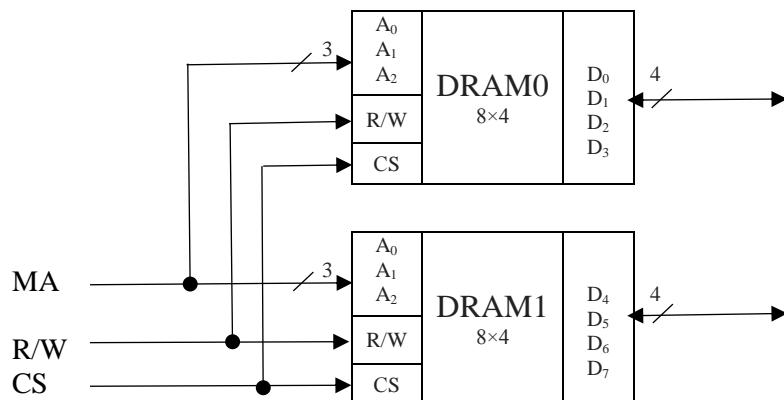


Fig. 4.27. Schema electrică a memoriei cu configurația 8×8 .

Adresarea registrelor memoriei cu configurația 8×8 este prezentată în tabelul 4.6.

Tabelul 4.6. Adresele registrelor memoriei cu configurația 8×8

Nr. d/0	Adresa, A ₂ A ₁ A ₀	Registrul
0	000	RG0
1	001	RG1
2	010	RG2
3	011	RG3
4	100	RG4
5	101	RG5
6	110	RG6
7	111	RG7

Configurația 16×4 . Pentru a forma modulul cu configurația 16×4 se procedează astfel:

1) se conectează intrările CS ale microcircuitelor DRAM0 și DRAM1 printr-un inversor, deoarece microcircuitele trebuie să funcționeze în perioade de timp diferit (8 registre în DRAM0 și 8 registre în DRAM1);

2) intrările R/W ale microcircuitelor DRAM0 și DRAM1 pot fi conectate împreună, deoarece operațiile de înscrriere/citire se îndeplinesc numai în cazul cînd $CS = 1$ pentru DRAM0 sau pentru DRAM1;

3) canalele de ordinul respectiv a magistralelor adreselor $A_2A_1A_0$ ale microcircuitelor DRAM0 și DRAM1 tot pot fi conectate împreună, deoarece microcircuitele funcționează în perioade de timp diferit cînd $CS = 1$ pentru DRAM0 sau pentru DRAM1;

4) canalele de ordinul respectiv ale magistralelor de date ale microcircuitelor DRAM0 și DRAM1 pot fi conectate împreună deoarece cuvintele sunt de ordinul patru..

Schema electrică a modulului de memorie cu configurația 16×4 , formată din două microcircuite DRAM cu structura 8×4 , este prezentată în fig. 4.28.

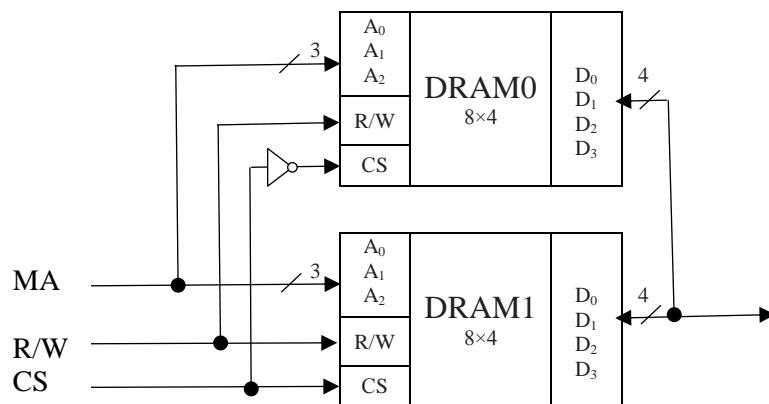


Fig. 4.24. Schema electrică a memoriei cu configurația 16×4 .

Adresarea registrelor memoriei cu configurația 16×4 este prezentată în tabelul 4.7.

Tabelul 4.7. Adresele registrelor memoriei cu configurația 16×4

Nr. d/0	Adresa, CS $A_2A_1A_0$	Registrul	Microcircuitul
0	0000	RG0	DRAM0
1	0001	RG1	
2	0010	RG2	
3	0011	RG3	
4	0100	RG4	
5	0101	RG5	
6	0110	RG6	
7	0111	RG7	
8	1000	RG8	DRAM1
9	1001	RG9	

10	1010	RG10	
11	1011	RG11	
12	1100	RG12	
13	1101	RG13	
14	1110	RG14	
15	1111	RG15	

Lucru independent nr. 6. De construit 2 module de memorie din microcircuite DRAM cu structura 4×4 :

- a) primul modul de memorie cu configurația 16×8 ;
- b) modulul doi de memorie cu configurația 8×16 .

Pentru adresarea microcircuitelor folosiți decodere.

5. DISPOZITIVE RESPONZABILE DE OPERAȚII ELEMENTARE

Orice aplicație de program, scrisă în limbaj de nivel înalt și executată pe calculator, se reduce la un set de operații logice și aritmetice elementare. Cele mai frecvente operații logice și aritmetice elementare executate asupra datelor la nivel de dispozitive fizice sunt:

- transferul datelor;
- transfer cu inversarea datelor;
- transfer cu deplasarea datelor;
- incrementarea datelor;
- decrementarea datelor;
- resetarea/inițializarea registrelor;
- sumarea datelor.

5.1. Transferul datelor

Operația de transfer a datelor este executată în cazurile:

- transferării datelor de pe magistrala de date în registrul;
- transferării datelor dintr-un registrul în alt registrul;
- transferării datelor din registrul pe magistrala de date.

5.1.1. Transferul datelor de pe magistrala de date în registrul

Schema electrică care execută operația de transfer a datelor de pe magistrala de date în registrul este prezentată în fig. 5.1. În fig. 5.1 sunt următoarele notări:

- $D_2D_1D_0$ – magistrala de date;
- RG – registrul;
- Move BUS to RG – comanda de transfer a datelor de pe magistrala de date în registrul;
- Clock – comanda de sincronizare a procesului de stocare a datelor în registrul;
- [C] – comutatorul;
- $Q_2Q_1Q_0$ – ieșirile registrului RG.

Bistabilii de tip D sunt utilizati pentru formarea semnalelor (datelor) pe magistrala de date și activarea instrucțiunii Move BUS to RG. Pentru a transfera datele de pe magistrala de date în registrul este necesar ca Move BUS to RG și Clock să primească valorile logice 1.

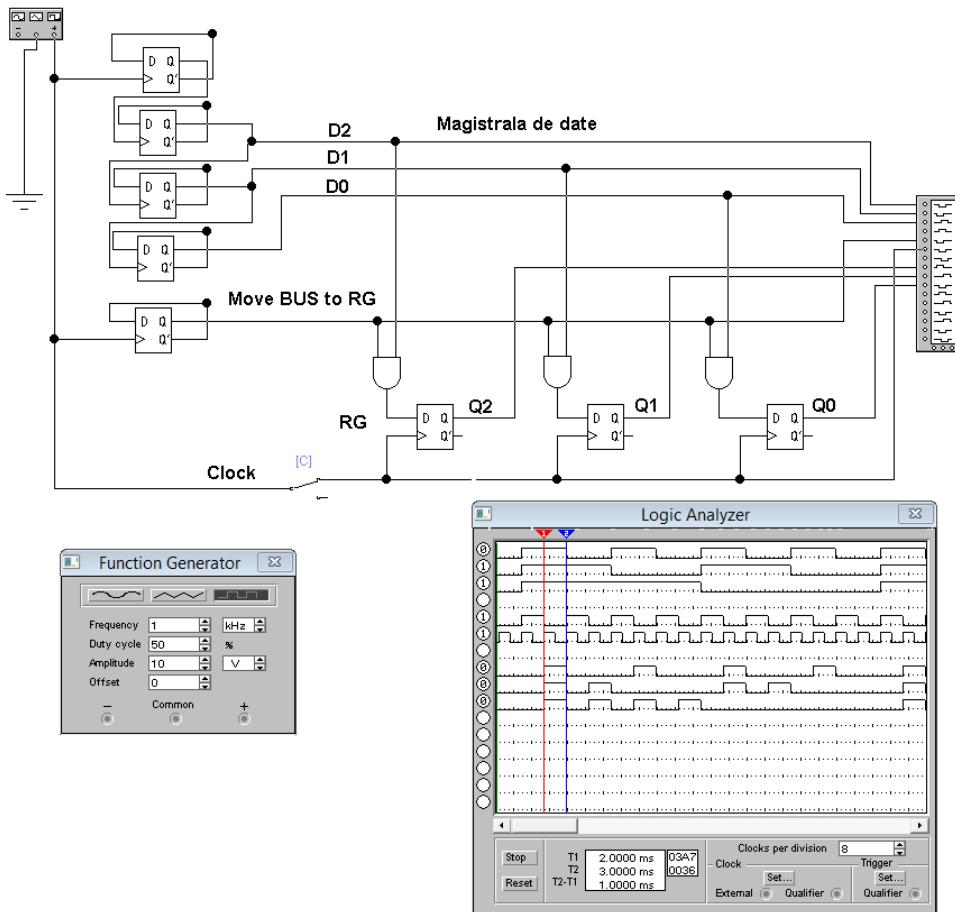


Fig. 5.1. Schema electrică care execută operația de transfer a datelor de pe magistrala de date în registrul.

5.1.2. Transferul datelor dintr-un registru în alt registru

Schema electrică care execută operațiile de transfer a datelor de pe magistrala de date în registrul RG A și din registrul RG A în registrul RG B este prezentată în fig. 5.2. În fig. 5.2 sunt următoarele notări:

- $D_2D_1D_0$ – magistrala de date;
- RG A, RG B – registrul A și registrul B;
- Move BUS to RG A – comanda de transfer a datelor de pe magistrala de date în registrul A;
- Clock RG A – comanda de sincronizare a procesului de stocare a datelor în registrul RG A;
- Move RG A to RG B – comanda de transfer a datelor din registrul RG A în registrul RG B;
- Clock RG B – comanda de sincronizare a procesului de stocare a datelor în registrul RG B;

- [A], [B] – comutatoare;
- $Q_{2A}Q_{1A}Q_{0A}$ – ieșirile registrului RG A;
- $Q_{2B}Q_{1B}Q_{0B}$ – ieșirile registrului RG B.

Pentru a transfera datele din registrul RG A în registrul RG B este necesar ca Move RG A to RG B și Clock RG B să primească valorile logice 1.

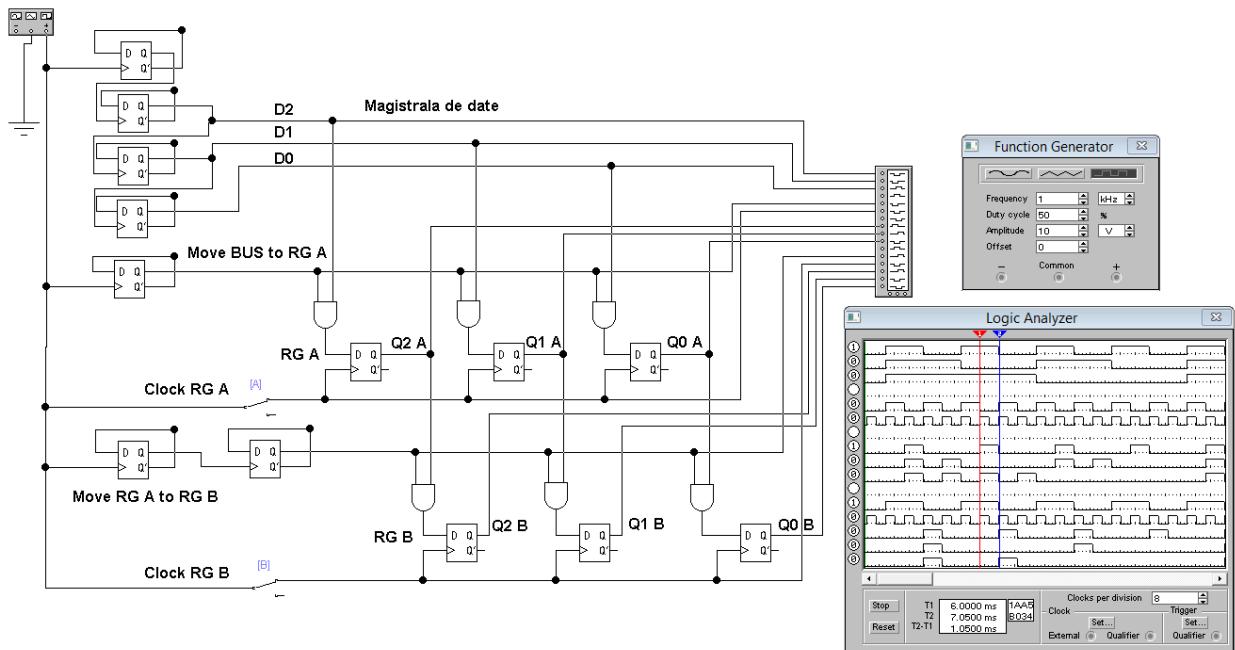


Fig. 5.2. Schema electrică care execută operațiile de transfer a datelor de pe magistrala de date în RG A și din RG A în RG B.

5.1.3. Transferul datelor din registru pe magistrala de date

Schema electrică care execută operațiile de transfer a datelor de pe magistrala de date în registrul RG A, din registrul RG A în registrul RG B și din registrul RG B pe magistrala de date este prezentată în fig. 5.3. În fig. 5.3 sunt următoarele notări:

- $D_2D_1D_0$ – magistrala de date;
- RG A, RG B – registrul A și registrul B;
- Move BUS to RG A – comanda de transfer a datelor de pe magistrala de date în registrul A;
- Clock RG A – comanda de sincronizare a procesului de stocare a datelor în registrul RG A;
- Move RG A to RG B – comanda de transfer a datelor din registrul RG A în registrul RG B;

- Clock RG B – comanda de sincronizare a procesului de stocare a datelor în registrul RG B;
- [A], [B], [B] – comutatoare;
- $Q_2A Q_1A Q_0A$ – ieșirile registrului RG A;
- $Q_2B Q_1B Q_0B$ – ieșirile registrului RG B.
- $A_2 A_1 A_0$ – amplificatoare;
- Move RG B to BUS – comanda de transfer a datelor din registrul RG B pe magistrala de date.

Pentru a transfera datele din registrul RG B pe magistrala de date este necesar ca Move RG B to BUS să primească valoarea logică 1.

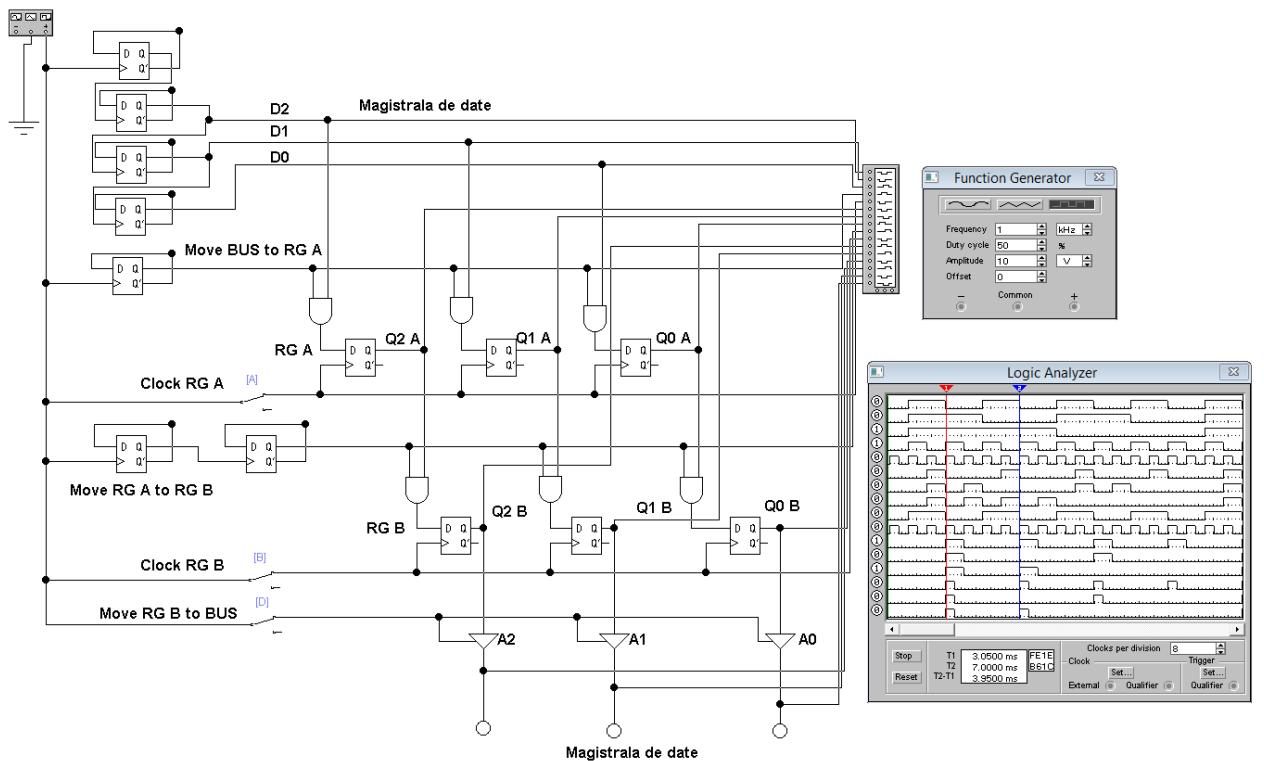


Fig. 5.3. Schema electrică care execută operațiile de transfer a datelor de pe magistrala de date în RG A, din RG A în RG B și din RG B pe magistrala de date.

5.2. Transfer cu inversarea datelor

Inversarea/complementarea datelor în procesul transferului din regiszr în regiszr poate fi efectuată cu ajutorul schemei electrice prezentate în figura 5.4. În fig. 5.4 sunt următoarele notări:

- $D_2 D_1 D_0$ – magistrala de date;
- RG A, RG B – regiszrul A și regiszrul B;

- Move BUS to RG A – comanda de transfer a datelor de pe magistrala de date în registrul A;
- Clock RG A – comanda de sincronizare a procesului de stocare a datelor în registrul RG A;
- Complement – comanda de inversare a datelor, aplicată la elementele SAU-EX;
- Move RG A to RG B – comanda de transfer a datelor din registrul RG A în registrul RG B;
- Clock RG B – comanda de sincronizare a procesului de stocare a datelor în registrul RG B;
- [A], [B], [D], [I], [0], [1], [2], [3], [4] – comutatoare;
- $Q_{2A}Q_{1A}Q_{0A}$ – ieșirile registrului RG A;
- $Q_{2B}Q_{1B}Q_{0B}$ – ieșirile registrului RG B.
- $A_2A_1A_0$ – amplificatoare;
- Move RG B to BUS – comanda de transfer a datelor din registrul RG B pe magistrala de date.

Pentru a inversa datele la ieșirile registrului RG A se folosesc elementele logice SAU-EX, la o intrare a căroră se aplică comanda Complement. Dacă Complement = 0, atunci datele nu sunt inverse, iar dacă Complement = 1, atunci datele sunt inverse.

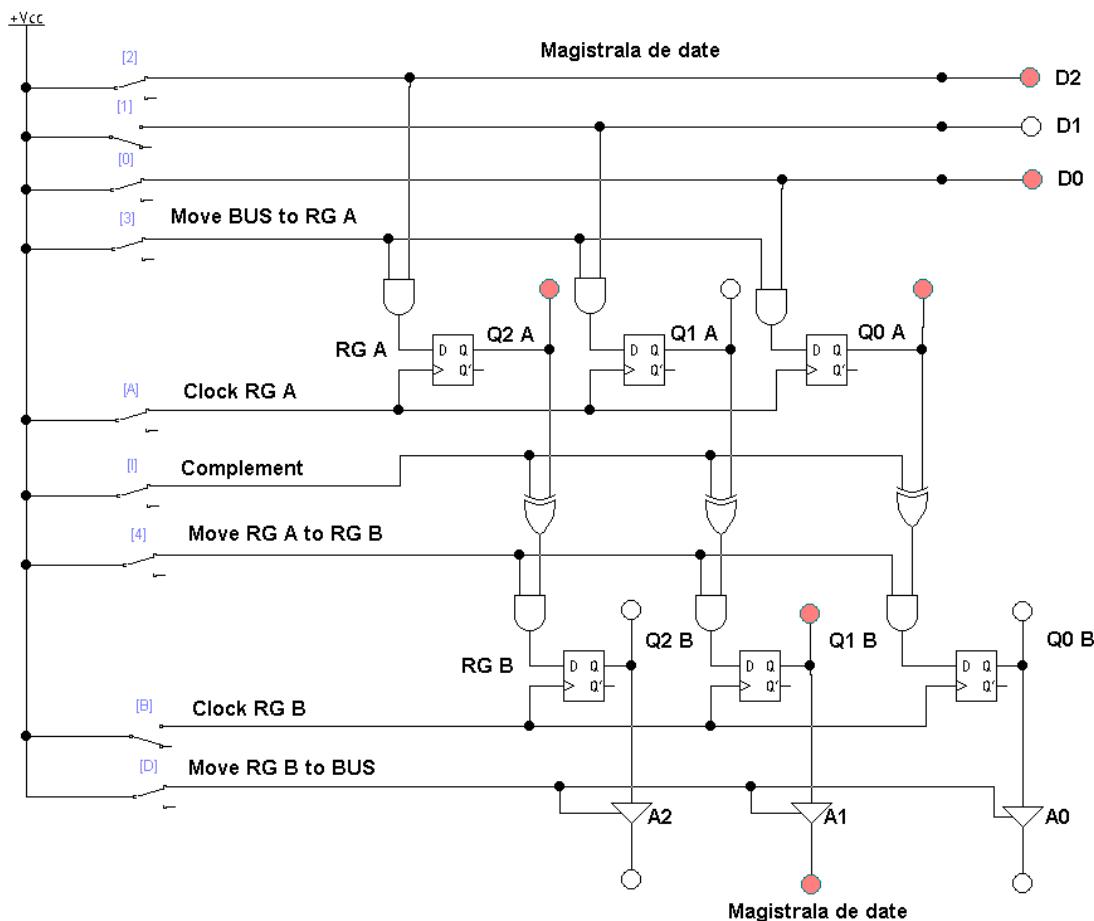


Fig. 5.4. Schema electrică care execută operația de transfer cu inversarea datelor.

Inversarea datelor poate fi efectuată cu ajutorul unui registru construit pe baza bistabililor JK. Dacă JK=11 atunci la ieșirile registrului se obțin stările inversate ale datelor care pot fi transferate în alt registru.

5.3. Transfer cu deplasarea datelor

Destul de des, în procesul de lucru al calculatorului, apare necesitatea de a efectua operația de deplasare a cuvântului binar cu un ordin sau mai multe ordine spre stînga sau dreapta în procesul transferului din registru în registru. Operația de deplasare a cuvântului binar cu un ordin sau mai multe ordine spre stînga sau dreapta în procesul transferului din registru în alt registru poate fi efectuată cu ajutorul schemei prezentate în figura 5.5. În fig. 5.5 sunt următoarele notări:

- $D_2D_1D_0$ – magistrala de date;
- RG A, RG B – registrul A și registrul B;
- Move BUS to RG A – comanda de transfer a datelor de pe magistrala de date în registrul A;
- Clock RG A – comanda de sincronizare a procesului de stocare a datelor în registrul RG A;
- Complement – comanda de inversare a datelor, aplicată la elementele SAU-EX;
- Move RG A to RG B – comanda de transfer a datelor din registrul RG A în registrul RG B;
- Clock RG B – comanda de sincronizare a procesului de stocare a datelor în registrul RG B;
- [A], [B], [0], [1], [2], [3], [4], [5], [6], [7], [8] – comutatoare;
- $Q_{2A}Q_{1A}Q_{0A}$ – ieșirile registrului RG A;
- $Q_{2B}Q_{1B}Q_{0B}$ – ieșirile registrului RG B.
- $A_2A_1A_0$ – amplificatoare;
- Move RG B to BUS – comanda de transfer a datelor din registrul RG B pe magistrala de date.

Pentru executarea deplasărilor datelor în procesul transferului lor din registrul RG A în registrul RG B sunt utilizate multiplexoare, formate din trei elemente logice SI și un element logic SAU. Deplasarea datelor spre dreapta, în procesul transferului din registrul RG A în registrul RG B, se face prin conectarea comutatorului [4] sau $Dd = 1$ ($D_0 = 0, D_s = 0$), transferul datelor din registrul RG A în registrul RG B fără deplasare se face prin conectarea comutatorului [5] sau $D_0 = 1$ ($Dd = 0, D_s = 0$), iar deplasarea datelor spre stînga, în procesul transferului din

registru RG A în registrul RG B, se face prin conectarea comutatorului [6] sau $D_s = 1$ ($D_d = 0$, $D_o = 0$). Numărul de elemente logice SI în multiplexoare este egal cu ordinul regisrelor.

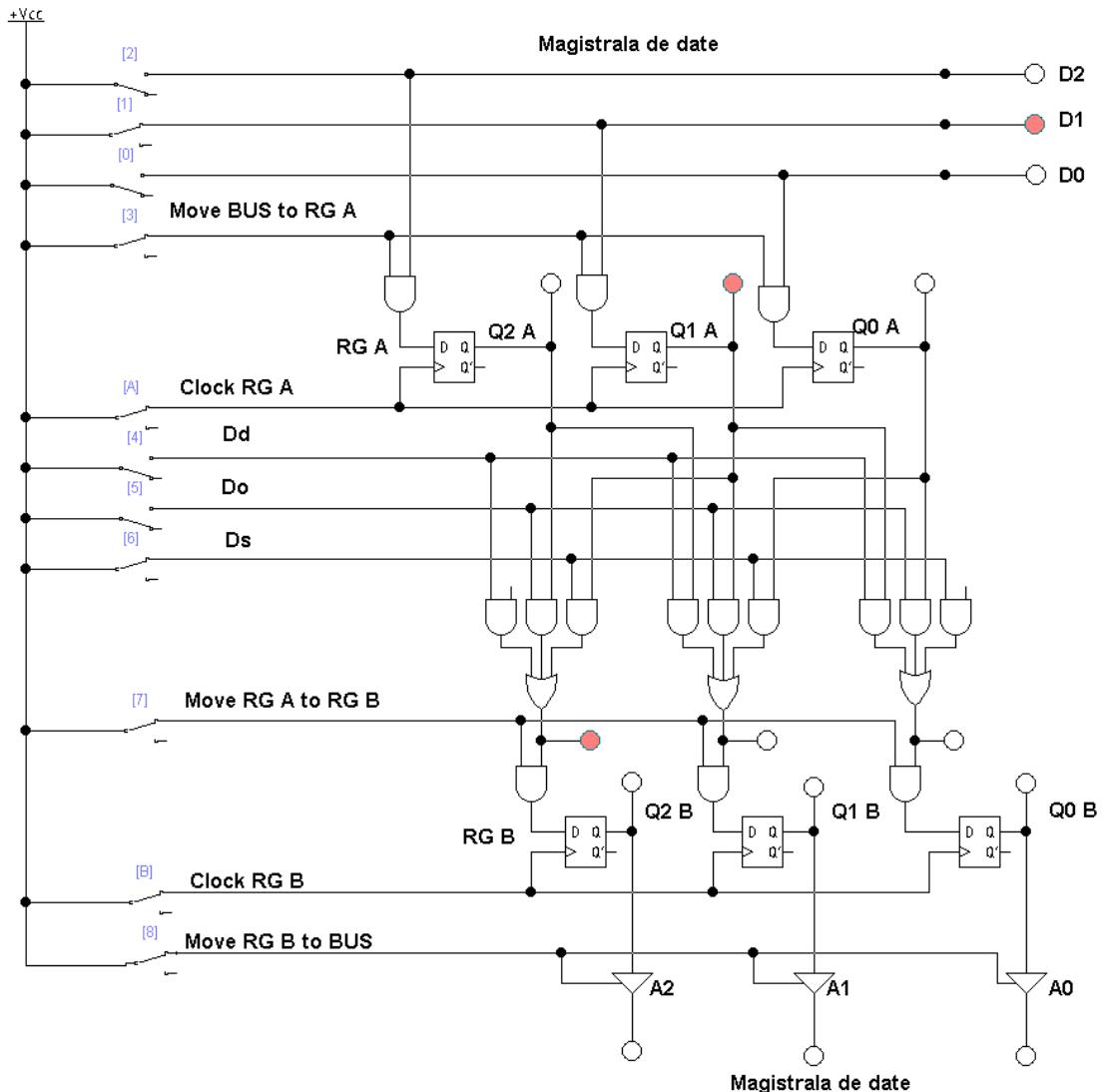


Fig. 5.5. Schema electrică care execută operația de transfer cu deplasarea datelor.

5.4. Registrul multioperational

În componentă microprocesoarelor și altor dispozitive ale calculatorului se utilizează registre multioperationali care pot executa mai multe operații elementare. Pentru exemplificare, în fig. 5.6 este prezentată schema electrică a unui registru de tip paralel multioperational de ordinul trei, construit din bistabili JK, care poate executa următoarele operații elementare: inițializarea registrului; transferul datelor de pe magistrală în registru; incrementarea datelor; complementarea datelor; transferul datelor din registru pe magistrala de date.

În fig. 5.6 sunt următoarele notări:

- [2], [1], [0] – comutatoarele magistralei de date D₂D₁D₀;
- [W] – comanda de transfer a datelor de pe magistrala de date în regisztr sau stocarea/înscrierea datelor (Move BUS to RG/[Write]);
- [I] – comanda de incrementare (adăugarea unei unități) a datelor;
- [Z] – comanda de inițializare/resetare a regisztrului;
- [C] – comanda de complementare (inversare) a datelor;
- [T] – comanda de schimbare a stărilor regisztrului în conformitate cu datele aplicate la intrările lui (Clock);
- [R] – comanda de transfer a datelor din regisztr pe magistrala de date sau citirea datelor (Move RG to BUS/[Read]);
- TT2, TT1, TT0 – bistabilii regisztrului;
- G1, G0 – semnale utilizate pentru inițierea procesului de incrementare a datelor;
- A₂A₁A₀ – amplificatoare;
- Q₂Q₁Q₀ – ieșirile regisztrului (magistrala de date).

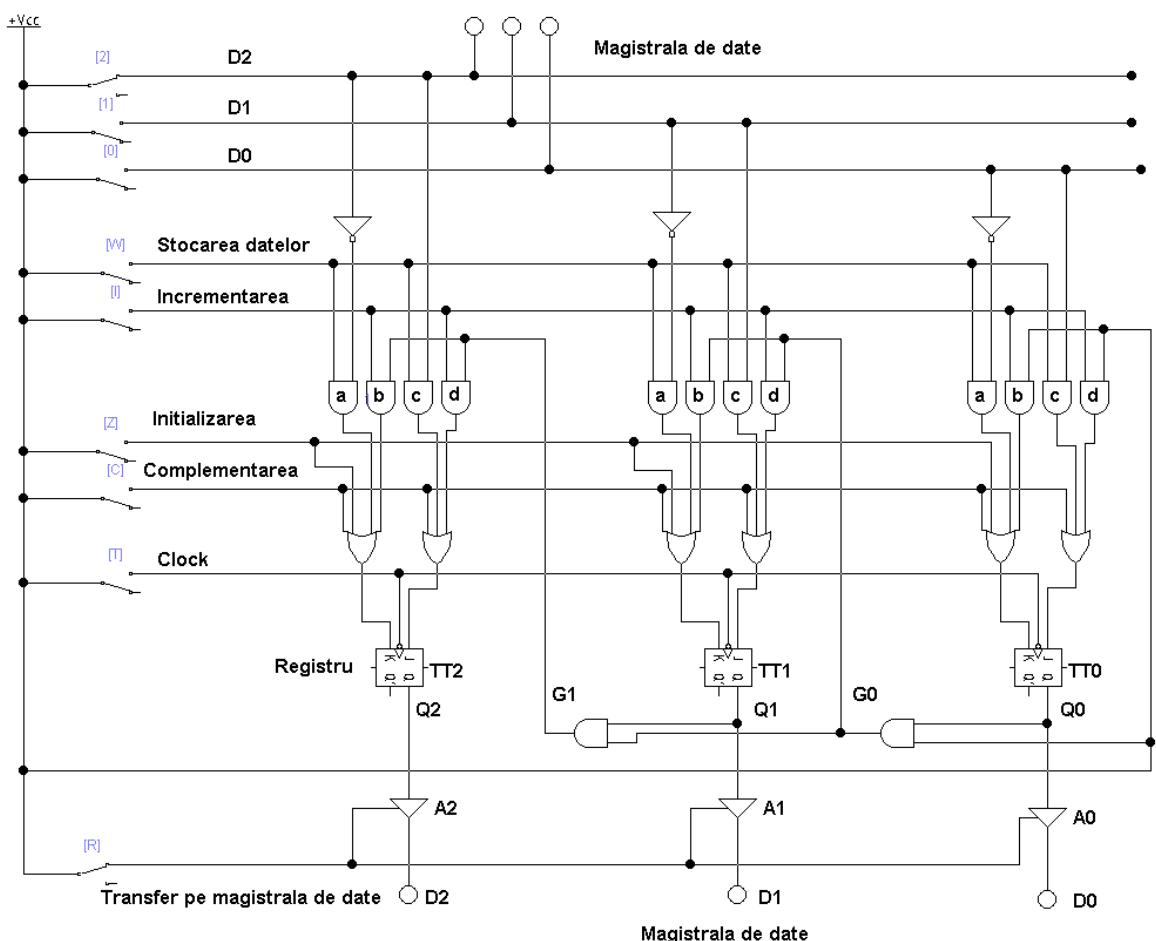


Fig. 5.6. Schema electrică a regisztrului multioperational.

Regiszrul multioperational poate executa următoarele operații:

- Z = 1, T = 1, W = 0, I = 0, C = 0, R = 0 – inițializarea regisztrului și, în acest caz, la intrările bistabililor TT2, TT1 și TT0 se aplică

$$J_2 = J_1 = J_0 = 0, \text{ iar } K_2 = K_1 = K_0 = 1;$$

- $W = 1, T = 1, Z = 0, I = 0, C = 0, R = 0$ – transferul datelor din magistrala de date $D_2D_1D_0$ în registrul și, în acest caz, la intrările bistabililor TT2, TT1 și TT0 se aplică

$$\begin{array}{ll} J_2 = D_2 \times W & K_2 = \bar{D}_2 \times W, \\ J_1 = D_1 \times W & K_1 = \bar{D}_1 \times W, \\ J_0 = D_0 \times W & K_0 = \bar{D}_0 \times W; \end{array}$$

- $I = 1, T = 1, Z = 0, W = 0, C = 0, R = 0$ – incrementarea (adăugarea unei unități) datelor și, în acest caz, la intrările bistabililor TT2, TT1 și TT0 se aplică

$$\begin{array}{ll} J_2 = G_1 \times I & K_2 = G_1 \times I, \\ J_1 = G_0 \times I & K_1 = G_0 \times I, \\ J_0 = 1 \times I & K_0 = 1 \times I; \end{array}$$

- $C = 1, T = 1, Z = 0, W = 0, I = 0, R = 0$ – complementarea (inversarea) datelor și, în acest caz, la intrările bistabililor TT2, TT1 și TT0 se aplică

$$J_2 = J_1 = J_0 = C = 1, K_2 = K_1 = K_0 = C = 1;$$

- $R = 1, T = 1, Z = 0, W = 0, I = 0, C = 0$ – transferul datelor din registrul pe magistrala de date $D_2D_1D_0$

$$\begin{array}{l} D_2 = Q_2 \times W, \\ D_1 = Q_1 \times W, \\ D_0 = Q_0 \times W. \end{array}$$

Simbolul convențional al registrului multioperațional este prezentat în fig. 5.7.

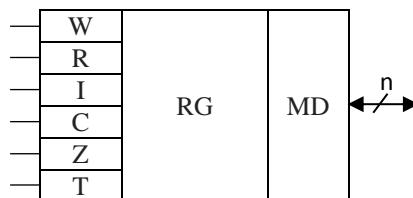


Fig. 5.7. Simbolul convențional al registrului multioperațional.

5.5. Unitatea aritmetică

Pentru executarea operațiilor aritmetice sau logice sunt utilizate unități aritmético-logice integrate în procesoare sau în microcircuite aparte. Drept exemplu, în fig. 5.8 este prezentată schema-bloc a unității aritmetice care poate calcula $S(s_{n-1}, \dots, s_1, s_0) = \pm A(a_{n-1}, \dots, a_1, a_0) \pm B(b_{n-1}, \dots, b_1, b_0)$. Schema-bloc permite urmărirea procesului de sumare a două numere binare la nivel de operații elementare executate de dispozitivele fizice.

În fig. 5.8 sunt următoarele notări:

- n – ordinul magistralelor de date, a registrelor și a sumatorului;

- W – înscrierea datelor în registrul multiopțional de pe magistrala de date MD_{ex} ;
- R – citirea datelor din registrul multiopțional sau transmiterea datelor în sumator;
- I – incrementarea datelor;
- C – complementarea datelor;
- Z – inițializarea registrului multiopțional;
- W_{AC} – înscrierea datelor în registrul acumulator;
- R_{AC} – citirea datelor din registrul acumulator sau transmiterea datelor pe magistralele de date MD_{ex} și MD_{in} ;
- Z_{AC} – inițializarea registrului acumulator;
- W_A – înscrierea datelor în registrul A;
- R_A – citirea datelor din registrul A sau transmiterea datelor pe magistrala de date MD_{ex} ;
- W_B – înscrierea datelor în registrul B;
- R_B – citirea datelor din registrul B sau transmiterea datelor pe magistrala de date MD_{ex} .

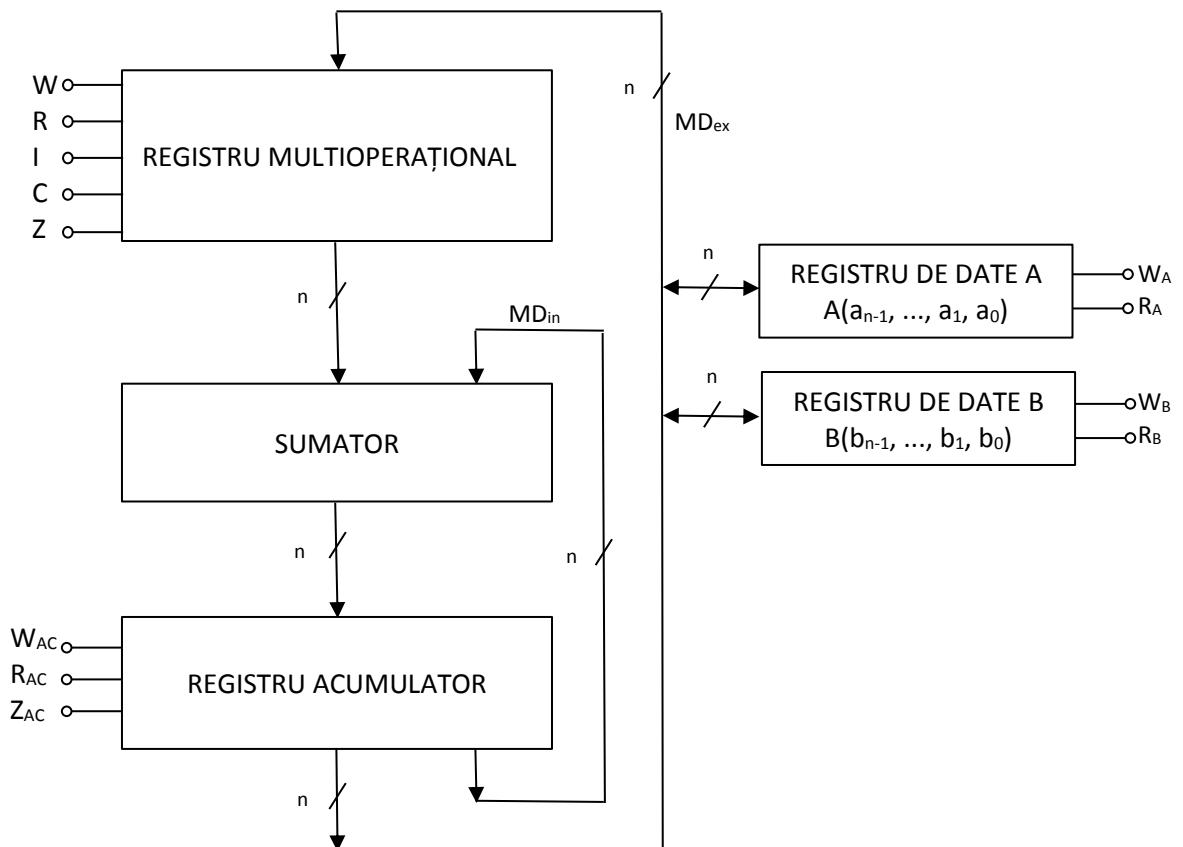


Fig. 5.8. Schema-bloc a unității aritmetice.

Fie în registrul A este stocat numărul binar $A(a_{n-1}, \dots, a_1, a_0)$, iar în registrul B – numărul bunar $B(b_{n-1}, \dots, b_1, b_0)$. Registrele A și B aparțin memoriei operative.

Pentru a calcula suma numerelor

$$S(s_{n-1}, \dots, s_1, s_0) = A(a_{n-1}, \dots, a_1, a_0) + B(b_{n-1}, \dots, b_1, b_0),$$

la nivel de operații elementare ale dispozitivelor fizice, trebuie executat următorul algoritm:

- 1) **Z = 1** – inițializarea registrului multiopțional;
- 2) **ZAC = 1** – inițializarea registrului acumulator;
- 3) **RA = 1** – citirea datelor din registrul A (transmiterea numărului $A(a_{n-1}, \dots, a_1, a_0)$ pe magistrala de date MD_{ex});
- 4) **W = 1** – înscrierea numărului $A(a_{n-1}, \dots, a_1, a_0)$ în registrul multiopțional de pe magistrala de date MD_{ex} ;
- 5) **R = 1** – citirea numărului $A(a_{n-1}, \dots, a_1, a_0)$ din registrul multiopțional sau transmiterea lui la intrarea sumatorului (sumatorul execută operația de sumare a numărului $A(a_{n-1}, \dots, a_1, a_0)$ cu $0(0, \dots, 0, 0)$ de pe magistrala de date MD_{in});
- 6) **WAC = 1** – înscrierea numărului $A(a_{n-1}, \dots, a_1, a_0)$, obținut la ieșirea sumatorului, în registrul acumulator;
- 7) **RAC = 1** – citirea numărului $A(a_{n-1}, \dots, a_1, a_0)$ din registrul acumulator sau transmiterea lui pe magistralele de date MD_{ex} și MD_{in} ;
- 8) **RB = 1** – citirea datelor din registrul B (transmiterea numărului $B(b_{n-1}, \dots, b_1, b_0)$ pe magistrala de date MD_{ex});
- 9) **W = 1** – înscrierea numărului $B(b_{n-1}, \dots, b_1, b_0)$ în registrul multiopțional de pe magistrala de date MD_{ex} ;
- 10) **R = 1** – citirea numărului $B(b_{n-1}, \dots, b_1, b_0)$ din registrul multiopțional sau transmiterea lui la intrarea sumatorului (sumatorul execută operația de sumare a numărului $B(b_{n-1}, \dots, b_1, b_0)$ cu $A(a_{n-1}, \dots, a_1, a_0)$ de pe magistrala de date MD_{in} și la ieșirea sumatorului se obține $S(s_{n-1}, \dots, s_1, s_0)$;
- 11) **WAC = 1** – înscrierea numărului $S(s_{n-1}, \dots, s_1, s_0)$, obținut la ieșirea sumatorului, în registrul acumulator;
- 12) **RAC = 1** – citirea numărului $S(s_{n-1}, \dots, s_1, s_0)$ din registrul acumulator sau transmiterea lui pe magistralele de date MD_{ex} și MD_{in} ;
- 13) **WA = 1** – înscrierea numărului $S(s_{n-1}, \dots, s_1, s_0)$, afișat pe magistrala de date MD_{ex} în registrul A al memoriei operative.

În urma analizei algoritmului prezentat se poate formula concluzia, că majoritatea operațiilor elementare efectuate în procesul adunării numerelor $A(a_{n-1}, \dots, a_1, a_0)$ și $B(b_{n-1}, \dots, b_1, b_0)$ sunt operații de transfer.