

3. DISPOZITIVE COMBINAȚIONALE

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

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

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

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;
- unitățile aritmetic-logice.

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

- nu pot păstra informația sau nu au memorie;
- starea următoare nu depinde de starea precedentă;
- starea la ieșirile 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 intrări 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 definiției decoderului.

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

Nr. d/o	Variabile			Funcții							
	x_2	x_1	x_0	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

3. DISPOZITIVE COMBINAȚIONALE

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

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

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

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;
- unitățile aritmetic-logice.

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

- nu pot păstra informația sau nu au memorie;
- starea următoare nu depinde de starea precedentă;
- starea la ieșirile 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 intrări 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 definiției decoderului.

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

Nr. d/o	Variabile			Funcții							
	x_2	x_1	x_0	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

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.

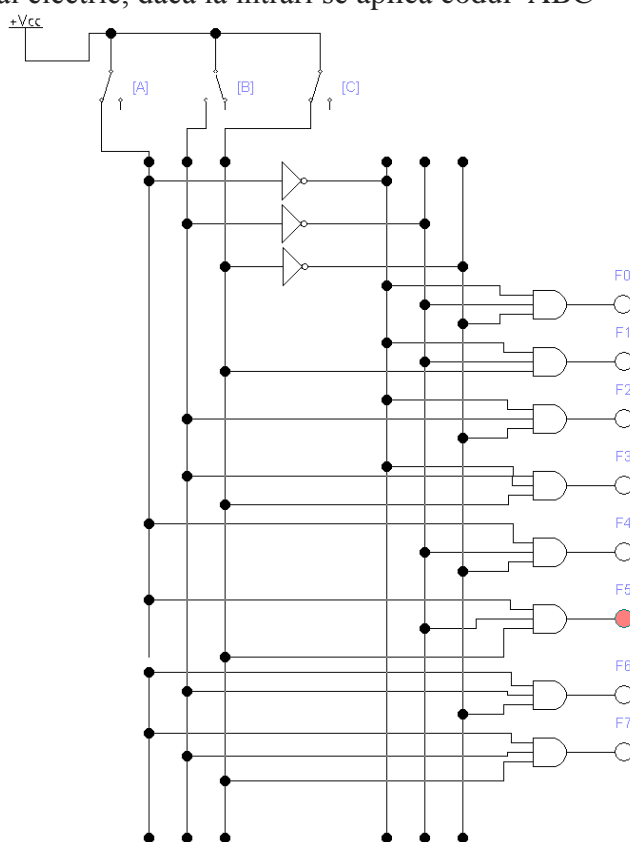


Fig. 3.1. Schema electrică a decoderului.

Simbolul convențional al decoderului este prezentat în figura 3.2.

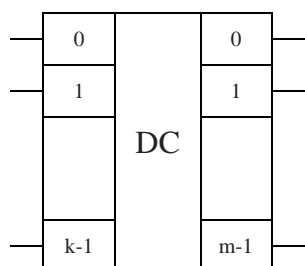


Fig. 3.2. Simbolul convențional al decoderului.

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, \dots, F_0 – ieșirile coderului sau magistrala de date. Din schemă se vede, că la ieșiri se obține codul 01112, dacă numai la intrarea cu numărul 7 se aplică zero.

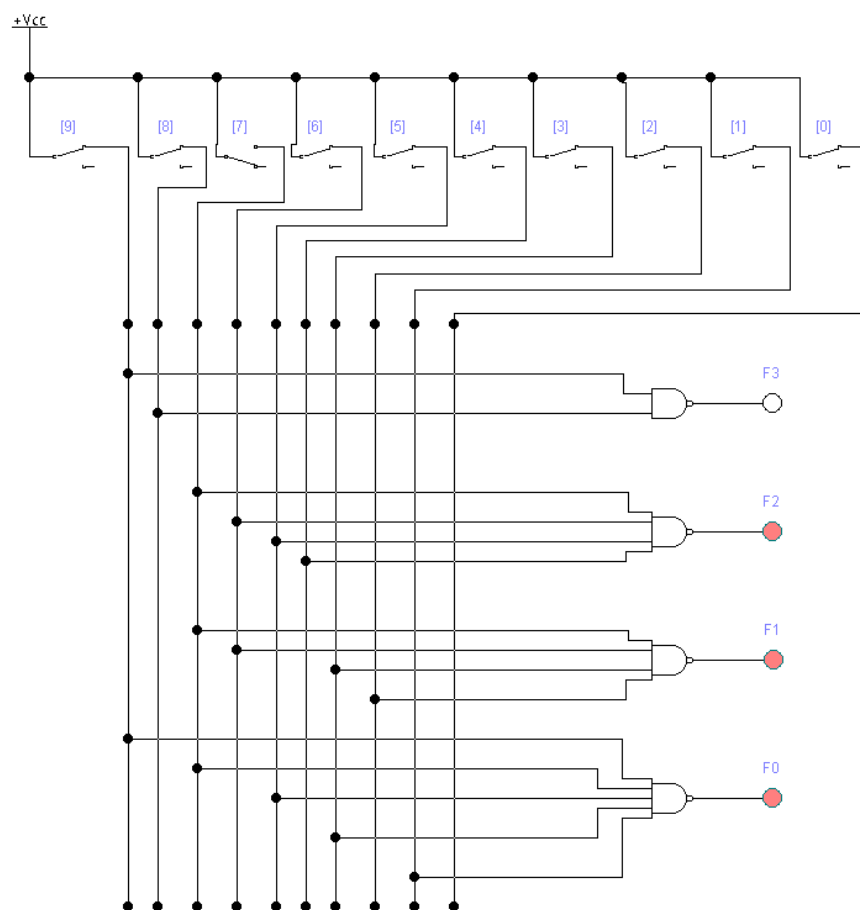


Fig. 3.3. Shema electrică a coderului.

Simbolul convențional al coderului este prezentat în figura 3.4.

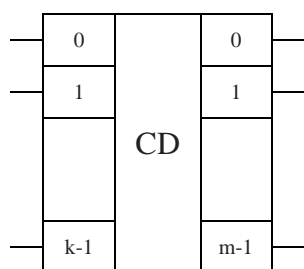


Fig. 3.4. Simbolul convențional al coderului.

Coderul este utilizat, în special, pentru a transforma un semnal electric în cod binar. Coderul este parte componentă a unității de comandă a microprocesorului, 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.

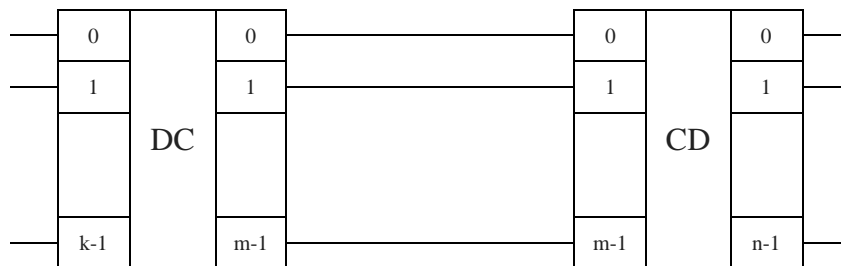


Fig. 3.5. Schema bloc a translatorului de cod.

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.

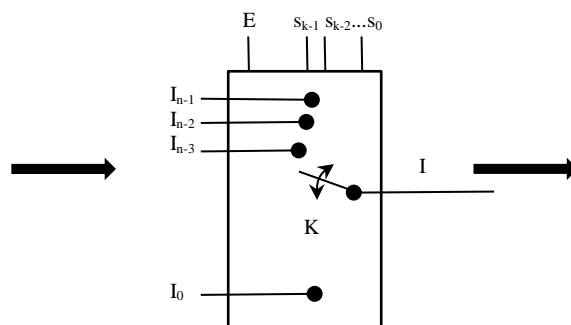


Fig. 3.6. Modelul structural al multiplexorului.

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 permite 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:

- a) $E = 0$, multiplexorul este în regim de repaus (nu transmite informația);
- b) $E = 1$, 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 $E = 1, s_1 s_0 = 10$ numai canalul de date I_2 se conectează cu ieșirea I .

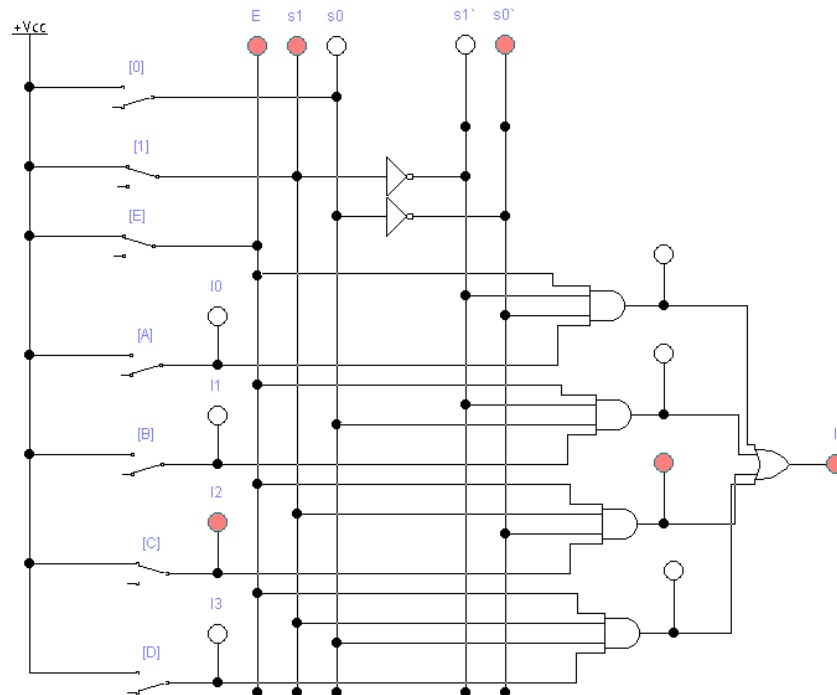


Fig. 3.7. Schema electrică a multiplexorului.

Simbolul convențional al multiplexorului este prezentat în figura 3.8.

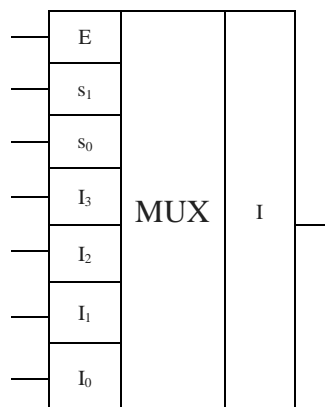


Fig. 3.8. Simbolul convențional al multiplexorului.

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 aplica un cod binar paralel care se menține un timp mai îndelungat și cu ajutorul codurilor de dirijare $s_{k-1}, s_{k-2}, \dots, s_0$ și comutatorului K se unesc pe rând canalele de la intrare se unesc pe rând 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 ciclic 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.

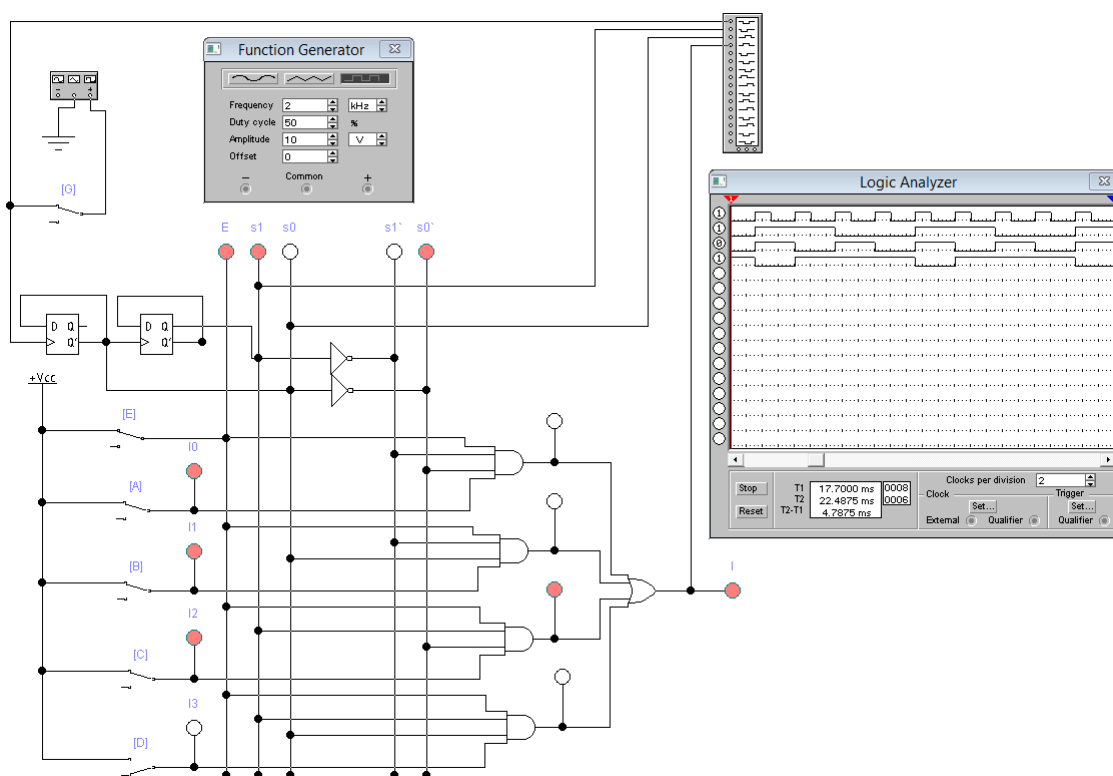


Fig. 3.9. Schema electrică a multiplexorului generator.

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.

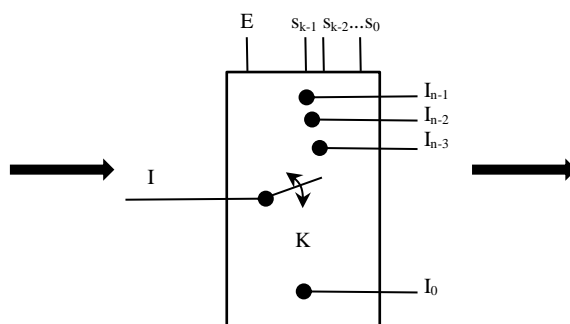


Fig. 3.10. Modelul structural al demultiplexorului.

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:

- a) $E = 0$, demultiplexorul este în regim de repaus (nu transmite informația);
- b) $E = 1$, 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

$$I_3 = E I s_1 s_0; \quad (3.15)$$

$$I_2 = E I s_1 \bar{s}_0; \quad (3.16)$$

$$I_1 = E I \bar{s}_1 s_0; \quad (3.17)$$

$$I_0 = E I \bar{s}_1 \bar{s}_0. \quad (3.18)$$

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 .

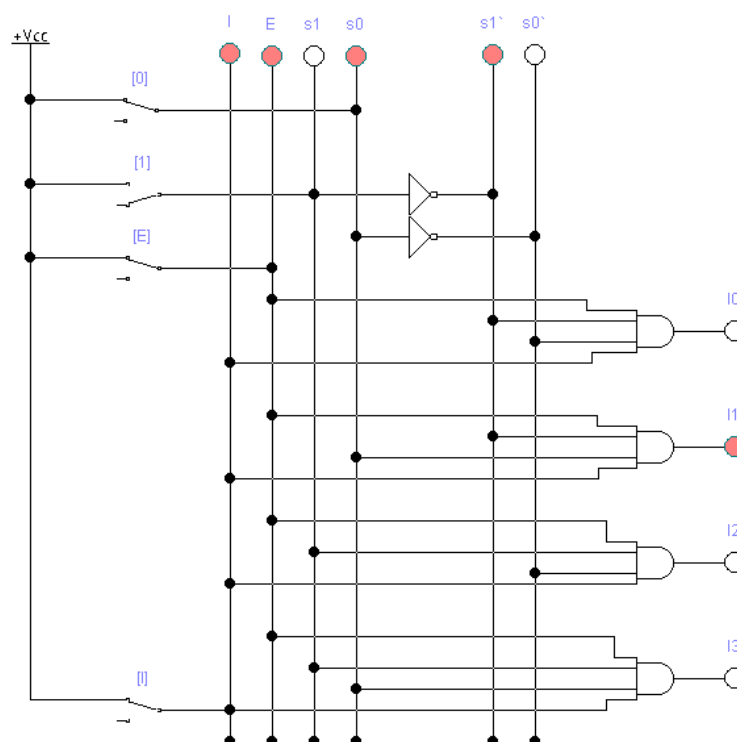


Fig. 3.11. Schema electrică a demultiplexorului.

Simbolul convențional al demultiplexorului este prezentat în figura 3.12.

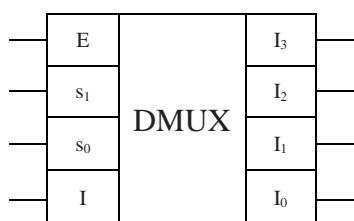


Fig. 3.12. Simbolul convențional al demultiplexorului.

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 rând 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 $\bar{B} \times B = 0$:

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

sau

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

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 $B(b_1, b_0)$.

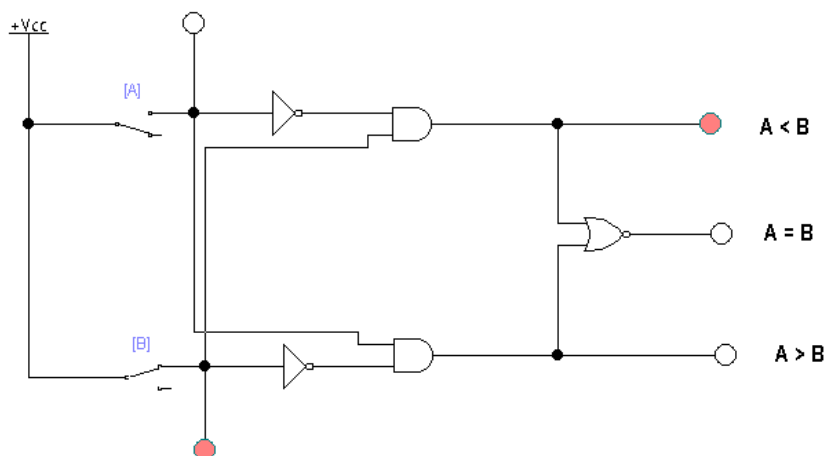


Fig. 3.13. Schema electrică a comparatorului de ordinul unu.

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:

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

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 minimizate prin metoda Veitch-Karnaugh (vezi fig. 3.14 și fig. 3.15).

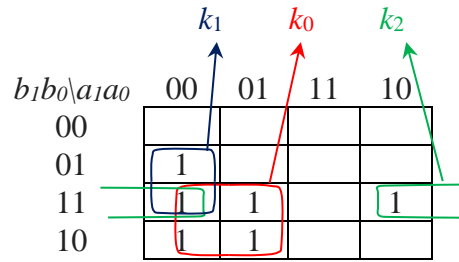


Fig. 3.14. Minimizarea funcției $F(A < B)$.

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

$$F(A < B) = k_2 + k_1 + k_0 = \bar{a}_1\bar{b}_1 + \bar{a}_1\bar{a}_0b_0 + \bar{a}_0b_1b_0. \quad (3.27)$$

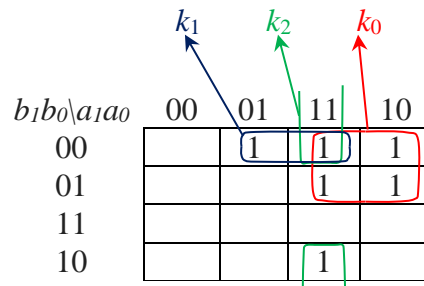


Fig. 3.15. Minimizarea funcției $F(A > B)$.

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

$$F(A > B) = k_2 + k_1 + k_0 = a_1\bar{b}_1 + a_0\bar{b}_1\bar{b}_0 + a_1a_0\bar{b}_0. \quad (3.28)$$

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

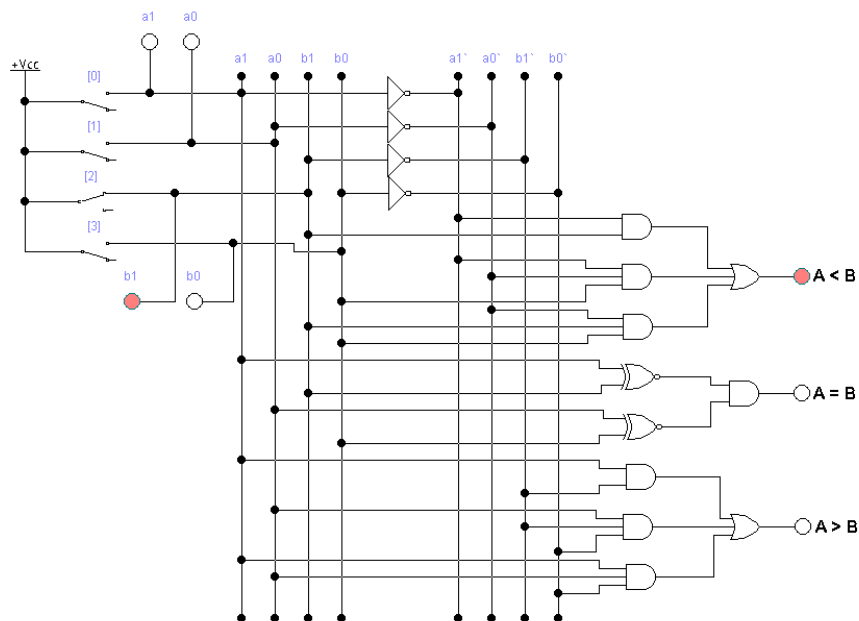


Fig. 3.16. Schema electrică a comparatorului de ordinul doi.

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.

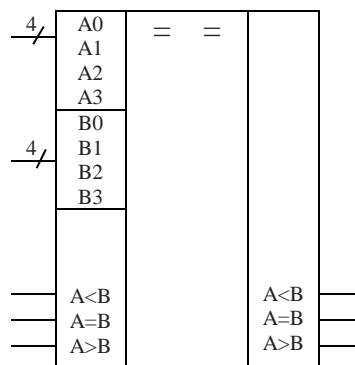


Fig. 3.17. Simbolul convențional al comparatorului de ordinal patru.

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 clasificații 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 doua 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 treia se aplică depășirea din ordinul inferior, la o ieșire se obține rezultatul sumării după modul, iar la ieșirea a doua 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:

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

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

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

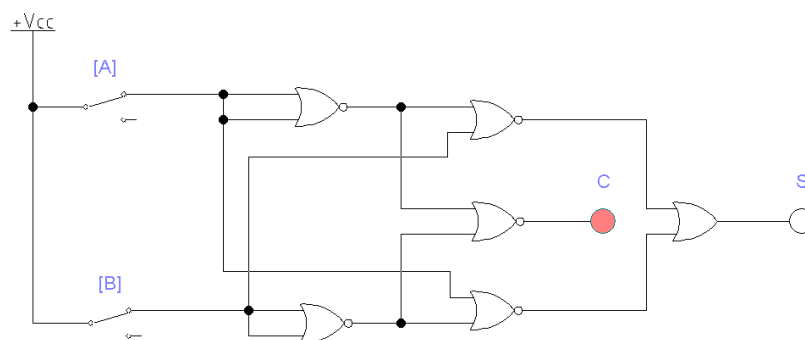


Fig. 3.18. Schema electrică a semisumatorului.

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.

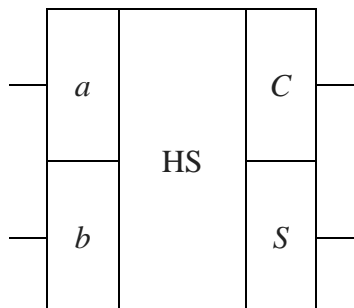


Fig. 3.19. Simbolul convențional al semisumatorului.

3.9. Sumator complet

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

Din cele prezentate în pagrafu 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 sunează 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 notate: 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.

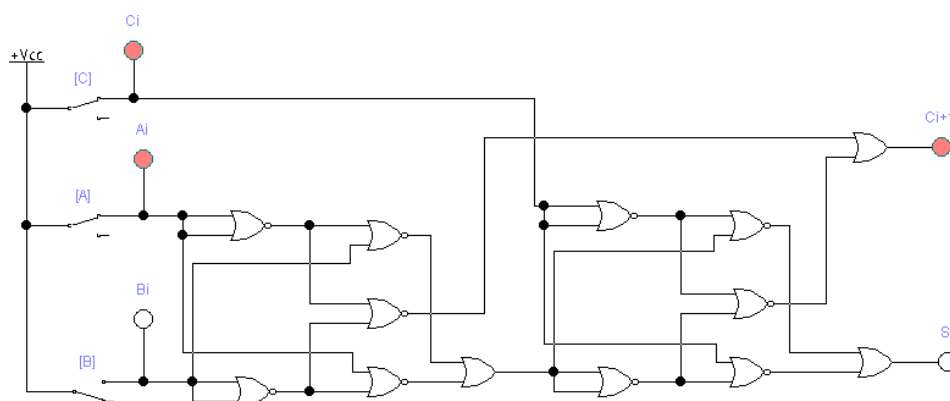


Fig. 3.20. Schema electrică a sumatorului complet.

Simbolul convențional al sumatorului complet este prezentat în figura 3.21.

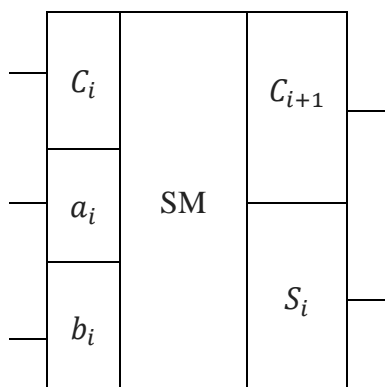


Fig. 3.21. Simbolul convențional al sumatorului complet.

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 înscririi 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 registrelor (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 proportional 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.

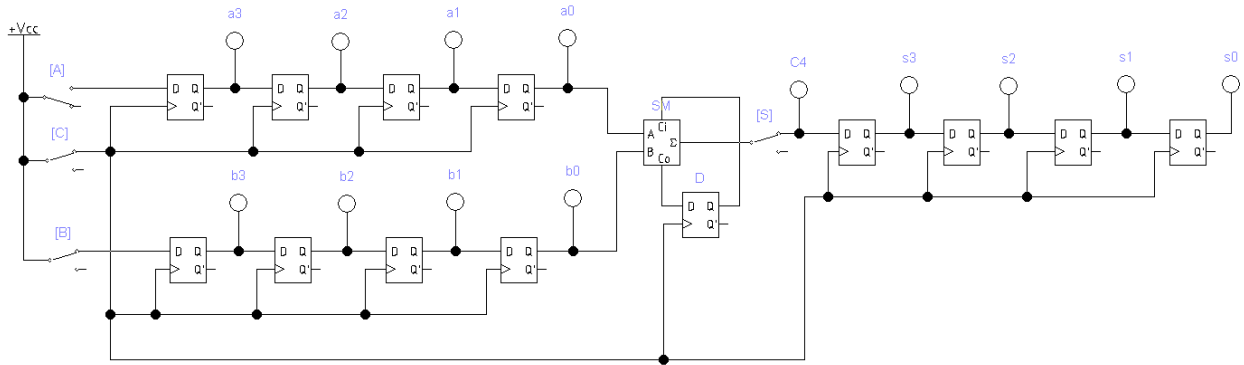


Fig. 3.22. Schema electrică a sumatorului consecutiv.

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

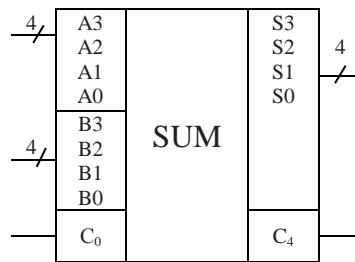


Fig. 3.23. Simbolul convențional al sumatorului de ordinul patru.

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

depășirii 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 operației de sumare pentru două numere, n – ordinul numerelor sumate, T_s – intervalul de timp necesar pentru sumarea tuturor perechilor de biți a_i și b_i (sumarea are loc în primul semisumator al sumatorului complet), iar T_d – intervalul de timp necesar pentru transferul depășirii C_i din ordinul anterior inferior și finalizarea operației 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 $A(a_3, a_2, a_1, a_0) = 1101_2$ și $B(b_3, b_2, b_1, b_0) = 1011_2$.

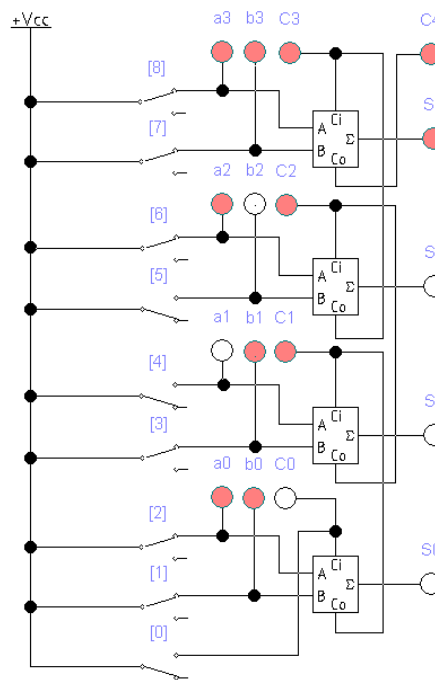


Fig. 3.22. Schema electrică a sumatorului paralel cu transfer consecutiv al depășirilor.

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

3.12. Sumator paralel cu transfer paralel al depășirilor

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

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

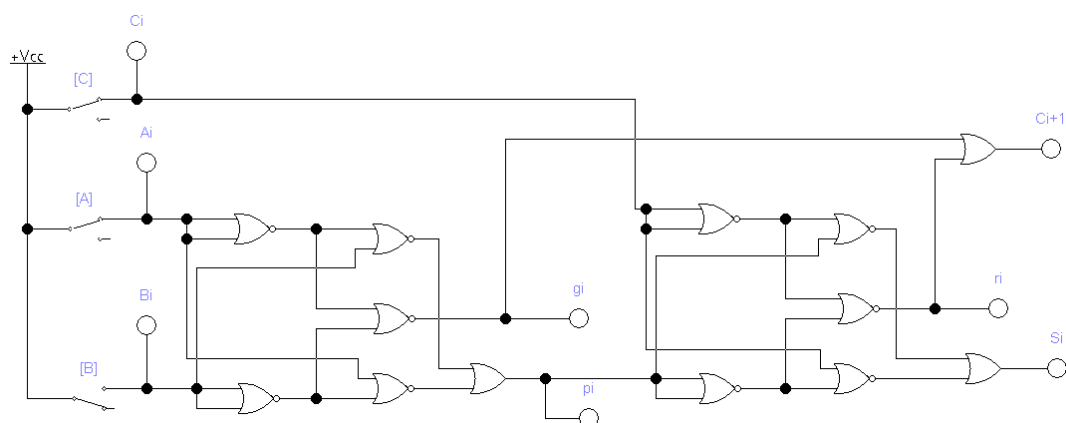


Fig. 3.23. Schema electrică a sumatorului complet de ordinul „i”.

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 mărimile 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 mărimile 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).

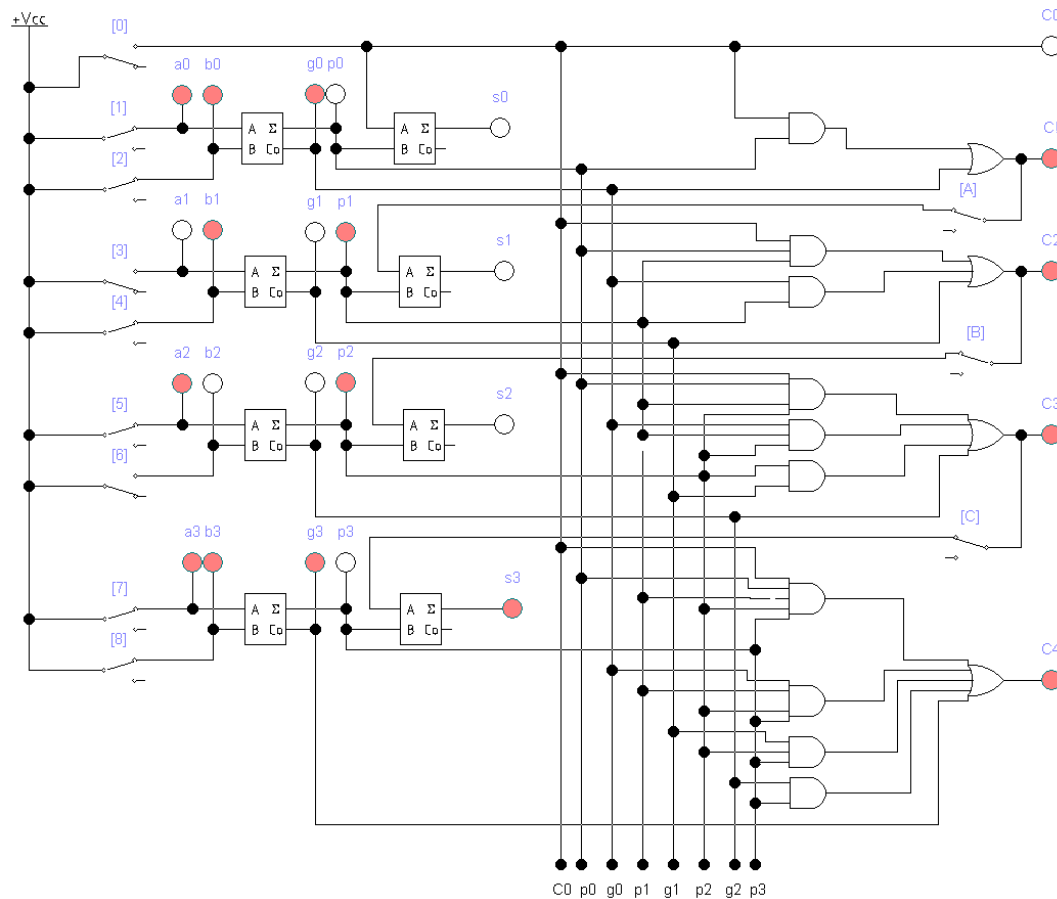


Fig. 3.24. Schema electrică a sumatorului paralel cu transfer paralel al depășirilor.

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 mărimile intermediare g_i, p_i , unde $i = 0, 1, 2, 3$;

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

3) depășirile 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 căreia 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 depășirilor de către schema de transfer paralel 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 depășirilor 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 depășirilor.

Neajunsul principal al sumatorului paralel cu transfer paralel al depășirilor 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 depășirilor sunt conectați consecutiv sau paralel.