

## **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 registru;
- transferării datelor dintr-un registru în alt registru;
- transferării datelor din registru pe magistrala de date.

#### **5.1.1. Transferul datelor de pe magistrala de date în registru**

Schema electrică care execută operația de transfer a datelor de pe magistrala de date în registru 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 registru;
- Clock – comanda de sincronizare a procesului de stocare a datelor în registru;
- [C] – comutatorul;
- $Q_2Q_1Q_0$  – ieșirile registrului RG.

Bistabili de tip D sunt utilizați 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 registru este necesar ca Move BUS to RG și Clock să primească valorile logice 1.

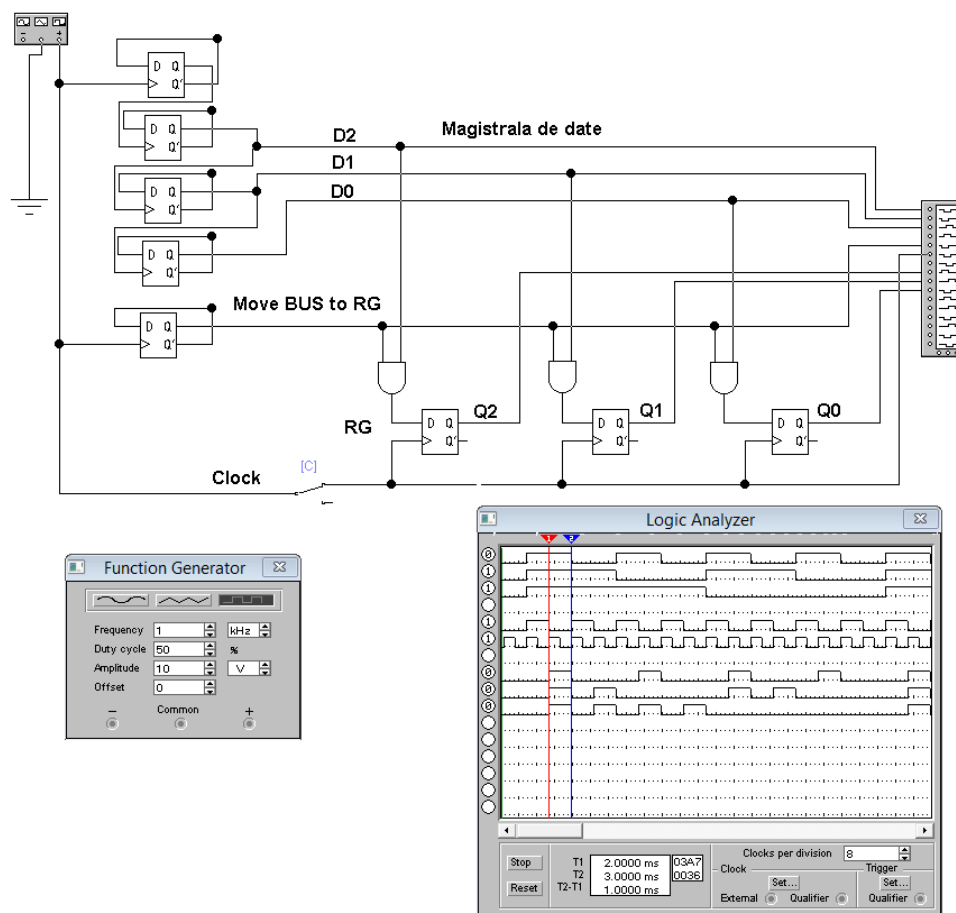


Fig. 5.1. Schema electrică care execută operația de transfer a datelor de pe magistrala de date în registru.

### 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.

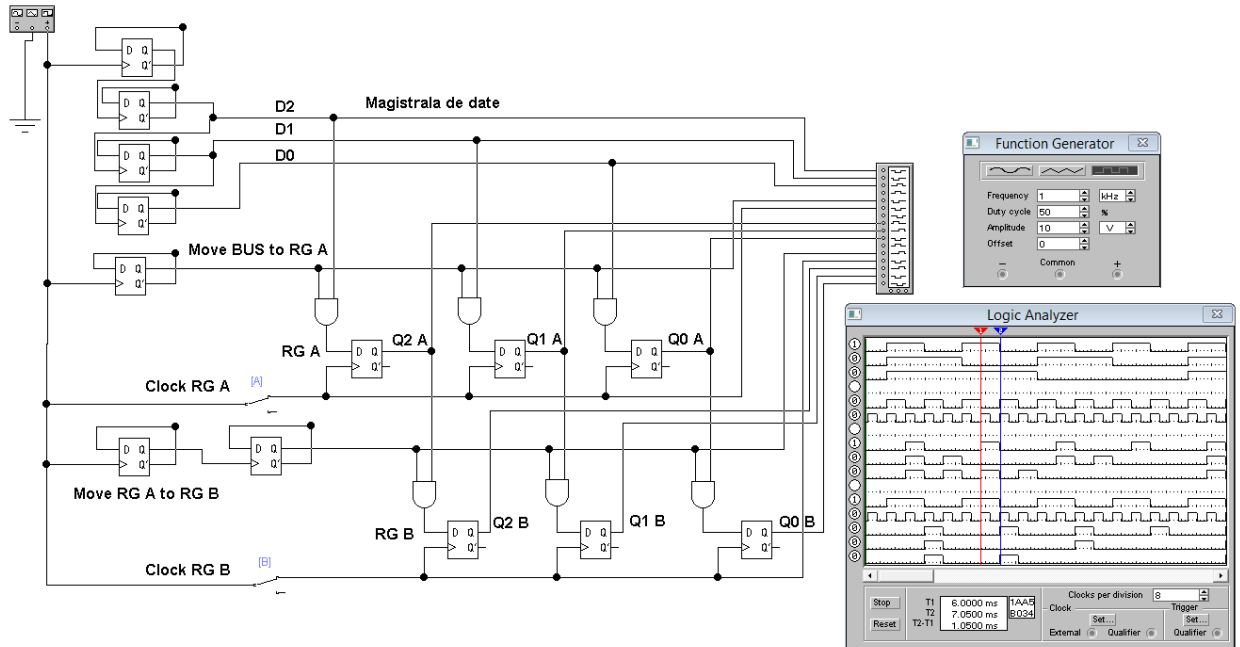


Fig. 5.2. Schema electrică care execută operațiile de transfer a datelor de pe magistrala de date în RG A și din RG A în RG B.

### 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.

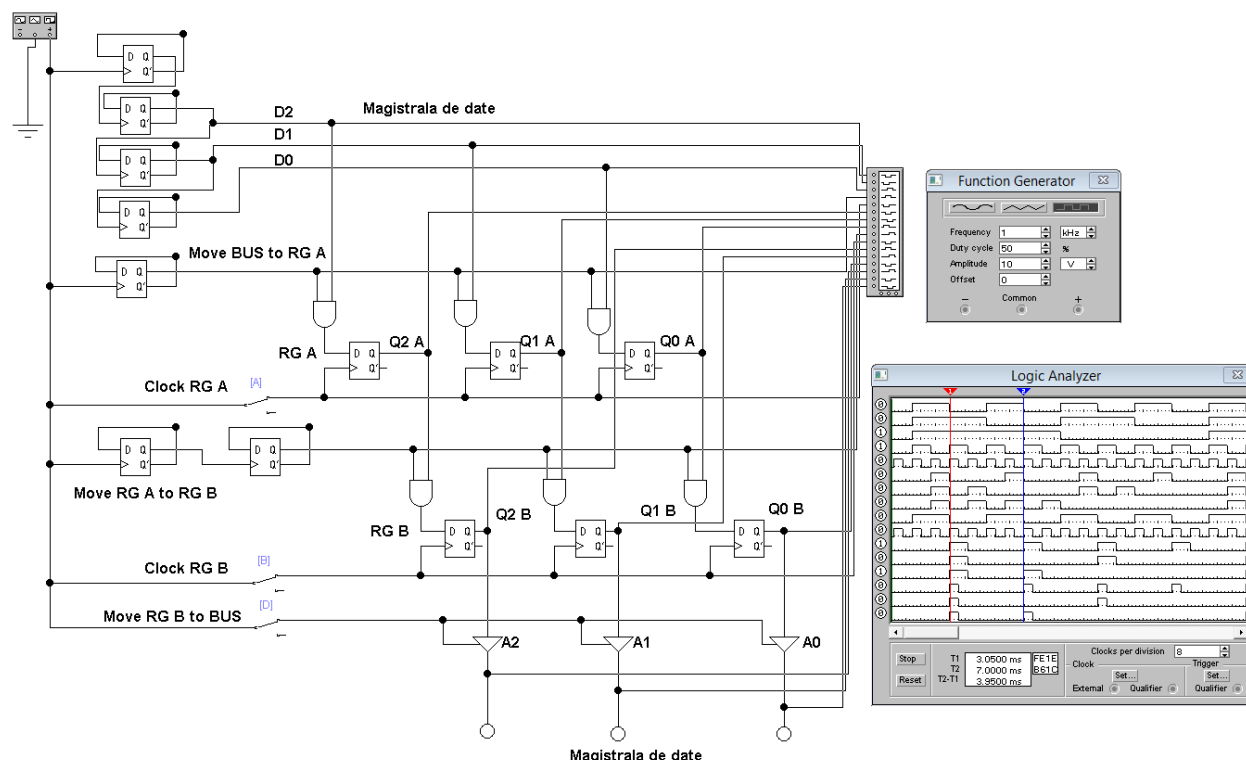


Fig. 5.3. Schema electrică care execută operațiile de transfer a datelor de pe magistrala de date în RG A, din RG A în RG B și din RG B pe magistrala de date.

## 5.2. Transfer cu inversarea datelor

Inversarea/complementarea datelor în procesul transferului din registru în registru 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 – 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], [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 cărora se aplică comanda Complement. Dacă Complement = 0, atunci datele nu sunt inversate, iar dacă Complement = 1, atunci datele sunt inversate.

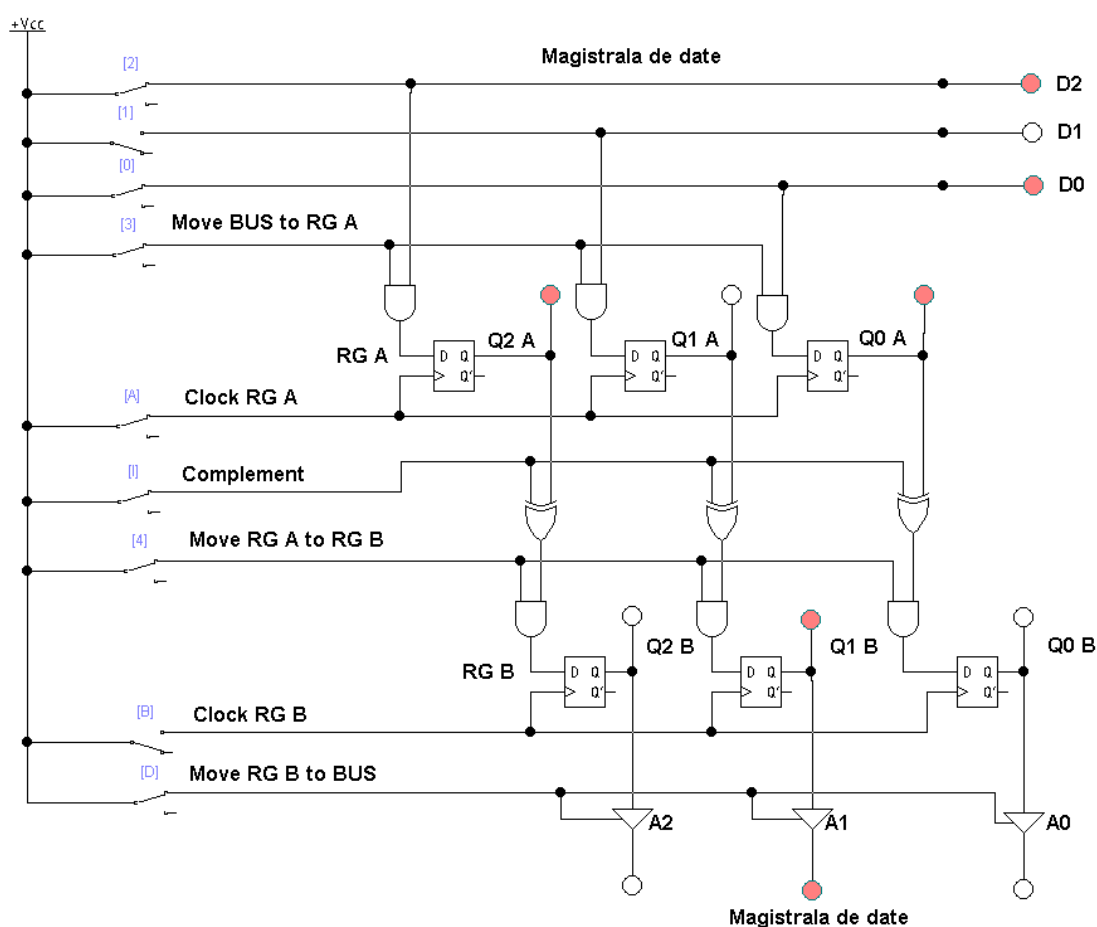


Fig. 5.4. Schema electrică care execută operația de transfer cu inversarea datelor.

Inversarea datelor poate fi efectuată cu ajutorul unui registru construit pe baza bistabililor JK. Dacă  $JK=11$  atunci la ieşirile registrului se obţin 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 operaţia de deplasare a cuvîntului binar cu un ordin sau mai multe ordine spre stînga sau dreapta în procesul transferului din registru în registru. Operaţia de deplasare a cuvîntului 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 şi 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;
- $Q_{2A}Q_{1A}Q_{0A}$  – ieşirile registrului RG A;
- $Q_{2B}Q_{1B}Q_{0B}$  – ieşirile 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 ŞI şi 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  $D_d = 1$  ( $D_o = 0$ ,  $D_s = 0$ ), transferul datelor din registrul RG A în registrul RG B fără deplasare se face prin conectarea comutatorului [5] sau  $D_o = 1$  ( $D_d = 0$ ,  $D_s = 0$ ), iar deplasarea datelor spre stînga, în procesul transferului din

registrul 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 ȘI în multiplexoare este egal cu ordinul registrelor.

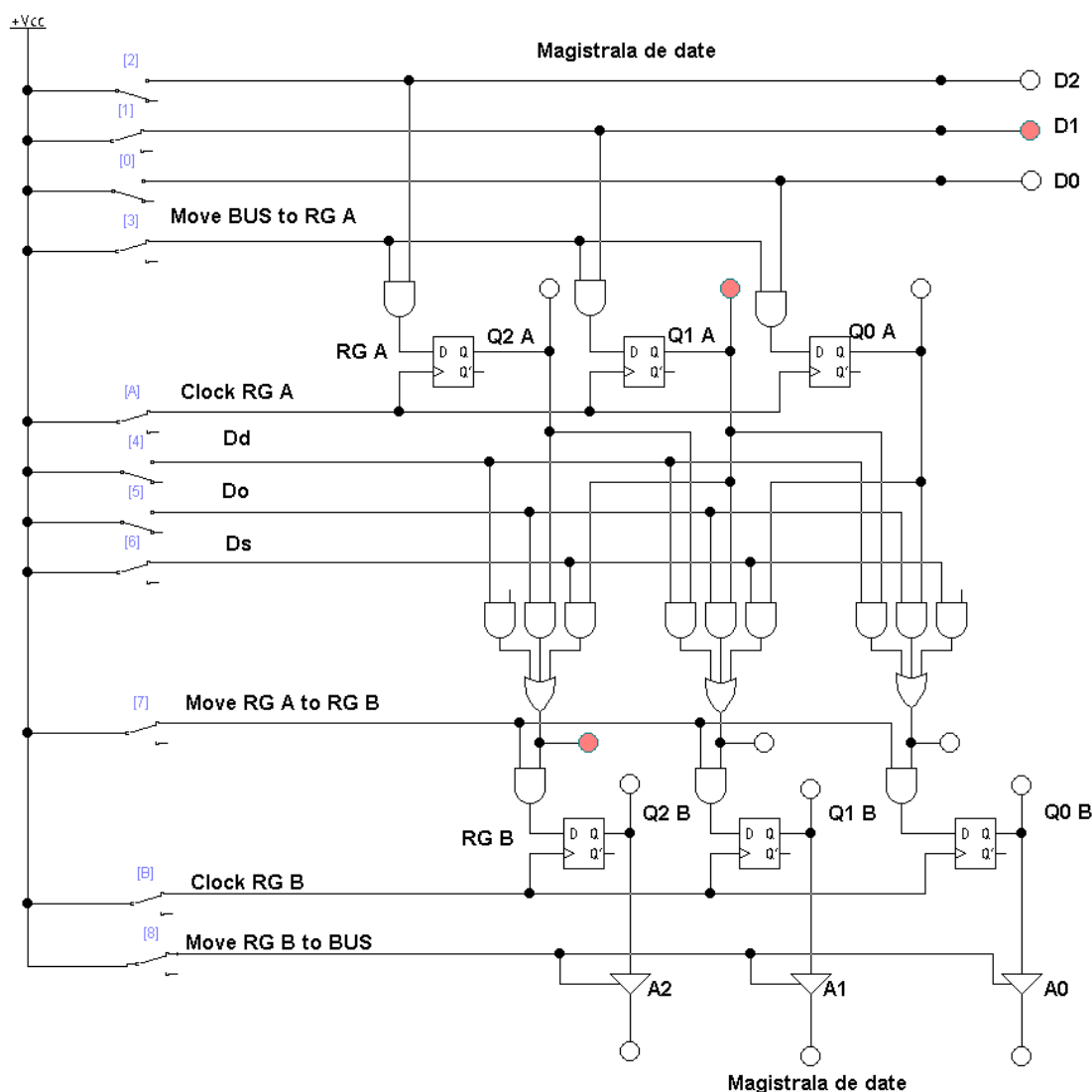


Fig. 5.5. Schema electrică care execută operația de transfer cu deplasarea datelor.

#### 5.4. Registru multioperațional

În componența microprocesoarelor și altor dispozitive ale calculatorului se utilizează registre multioperaționale care pot executa mai multe operații elementare. Pentru exemplificare, în fig. 5.6 este prezentată schema electrică a unui registru de tip paralel multioperațional 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  $D_2D_1D_0$ ;
- [W] – comanda de transfer a datelor de pe magistrala de date în registru sau stocarea/înscrierea datelor (Move BUS to RG/[Write]);
- [I] – comanda de incrementare (adăugarea unei unități) a datelor;
- [Z] – comanda de inițializare/resetare a registrului;
- [C] – comanda de complementare (inversare) a datelor;
- [T] – comanda de schimbare a stărilor registrului în conformitate cu datele aplicate la intrările lui (Clock);
- [R] – comanda de transfer a datelor din registru pe magistrala de date sau citirea datelor (Move RG to BUS/[Read]);
- TT2, TT1, TT0 – bistabilii registrului;
- G1, G0 – semnale utilizate pentru inițierea procesului de incrementare a datelor;
- $A_2A_1A_0$  – amplificatoare;
- $Q_2Q_1Q_0$  – ieșirile registrului (magistrala de date).

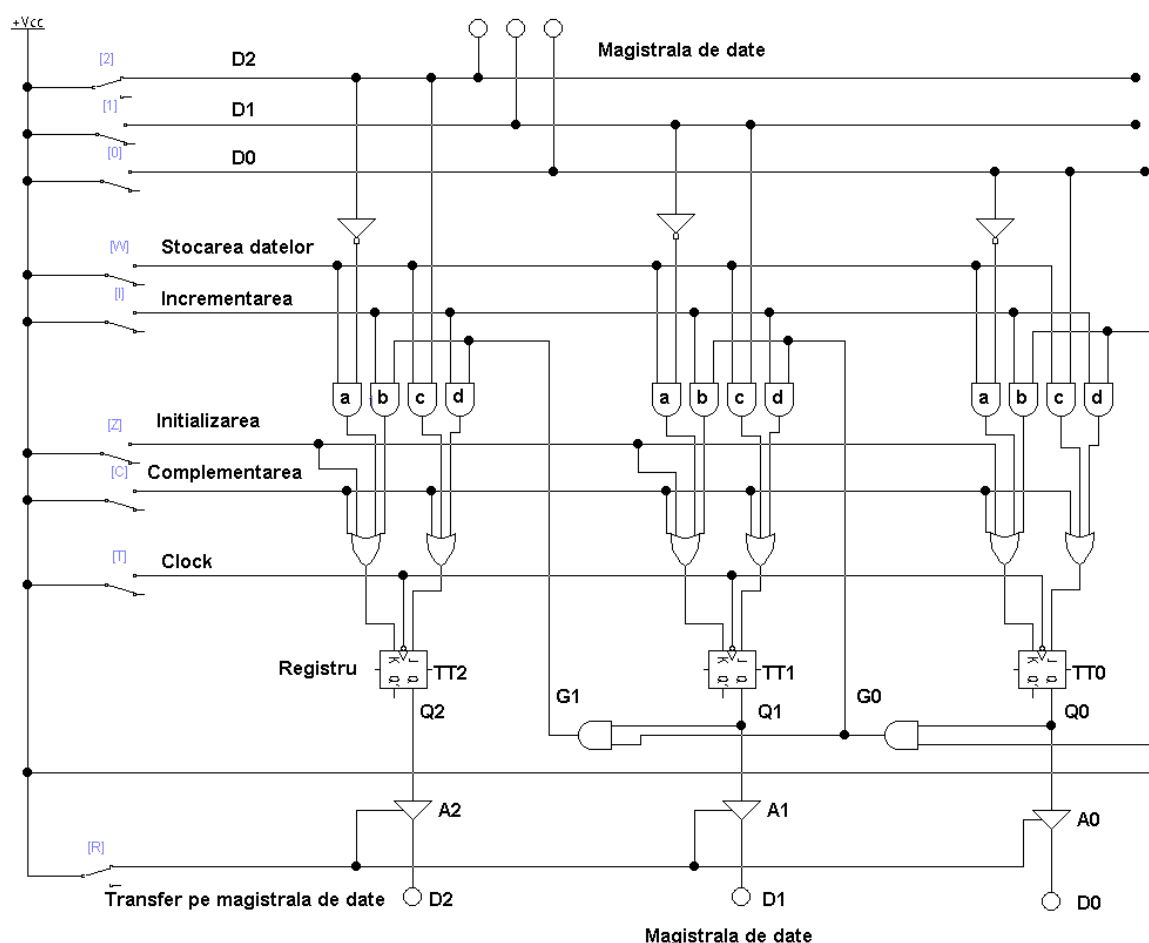


Fig. 5.6. Schema electrică a registrului multioperațional.

Registrul multioperațional poate executa următoarele operații:

- $Z = 1$ ,  $T = 1$ ,  $W = 0$ ,  $I = 0$ ,  $C = 0$ ,  $R = 0$  – inițializarea registrului 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 registru și, în acest caz, la intrările bistabililor TT2, TT1 și TT0 se aplică

$$\begin{aligned} J_2 &= D_2 \times W & K_2 &= \bar{D}_2 \times W, \\ J_1 &= D_1 \times W & K_1 &= \bar{D}_1 \times W, \\ J_0 &= D_0 \times W & K_0 &= \bar{D}_0 \times W; \end{aligned}$$

- $I = 1, T = 1, Z = 0, W = 0, C = 0, R = 0$  – incrementarea (adăugarea unei unități) datelor și, în acest caz, la intrările bistabililor TT2, TT1 și TT0 se aplică

$$\begin{aligned} J_2 &= G_1 \times I & K_2 &= G_1 \times I, \\ J_1 &= G_0 \times I & K_1 &= G_0 \times I, \\ J_0 &= 1 \times I & K_0 &= 1 \times I; \end{aligned}$$

- $C = 1, T = 1, Z = 0, W = 0, I = 0, R = 0$  – 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 registru pe magistrala de date  $D_2D_1D_0$

$$\begin{aligned} D_2 &= Q_2 \times W, \\ D_1 &= Q_1 \times W, \\ D_0 &= Q_0 \times W. \end{aligned}$$

Simbolul convențional al registrului multioperațional este prezentat în fig. 5.7.

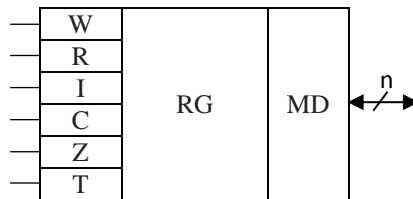


Fig. 5.7. Simbolul convențional al registrului multioperațional.

## 5.5. Unitatea aritmetică

Pentru executarea operațiilor aritmetice sau logice sunt utilizate unități aritmetico-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, registrelor și sumatorului;

- W – înscrierea datelor în registrul multiopțional de pe magistrala de date MD<sub>ex</sub>;
- R – citirea datelor din registrul multiopțional sau transmiterea datelor în sumator;
- I – incrementarea datelor;
- C – complementarea datelor;
- Z – inițializarea registrului multiopțional;
- W<sub>AC</sub> – înscrierea datelor în registrul acumulator;
- R<sub>AC</sub> – citirea datelor din registrul acumulator sau transmiterea datelor pe magistralele de date MD<sub>ex</sub> și MD<sub>in</sub>;
- Z<sub>AC</sub> – inițializarea registrului acumulator;
- W<sub>A</sub> – înscrierea datelor în registrul A;
- R<sub>A</sub> – citirea datelor din registrul A sau transmiterea datelor pe magistrala de date MD<sub>ex</sub>;
- W<sub>B</sub> – înscrierea datelor în registrul B;
- R<sub>B</sub> – citirea datelor din registrul B sau transmiterea datelor pe magistrala de date MD<sub>ex</sub>.

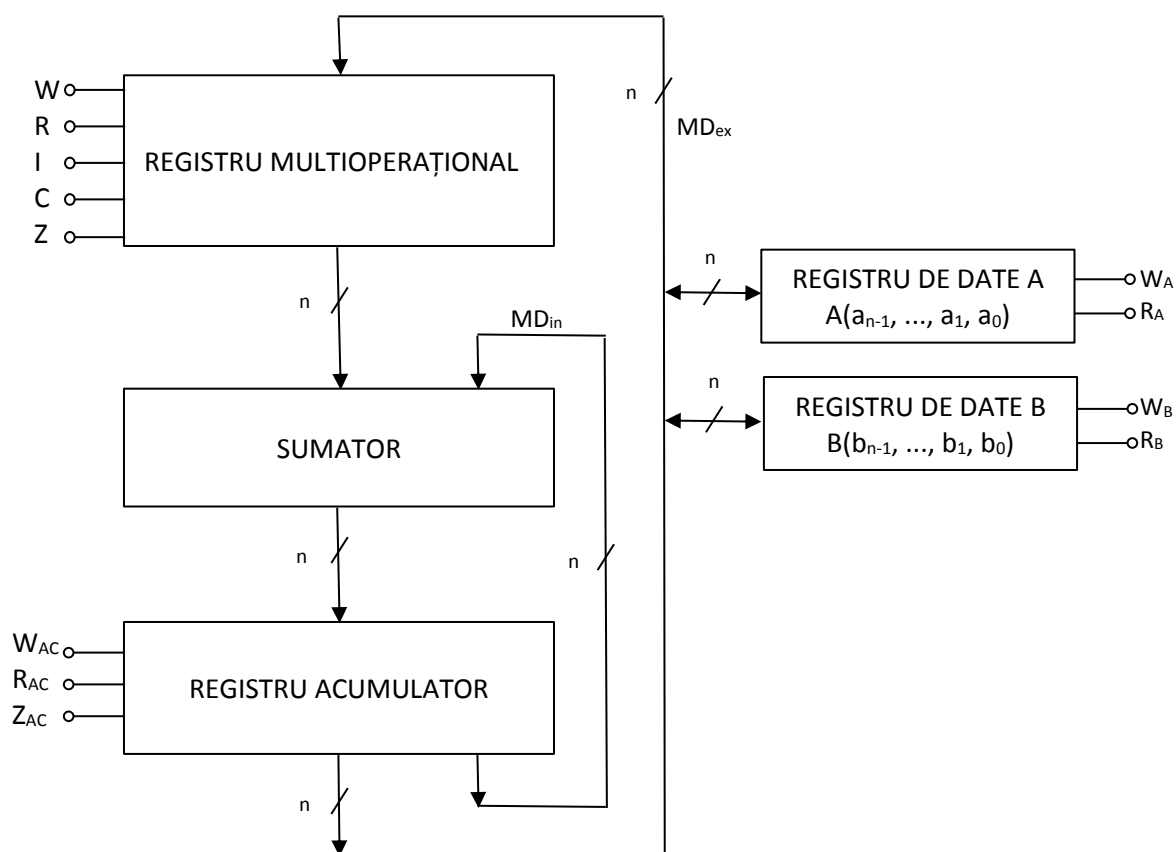


Fig. 5.8. Schema-bloc a unității aritmetice.

Fie în registrul A este stocat numărul binar  $A(a_{n-1}, \dots, a_1, a_0)$ , iar în registrul B – numărul binar  $B(b_{n-1}, \dots, b_1, b_0)$ . 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 multipoțional;
- 2) **Z<sub>AC</sub> = 1** – inițializarea registrului acumulator;
- 3) **R<sub>A</sub> = 1** – citirea datelor din registrul A (transmiterea numărului  $A(a_{n-1}, \dots, a_1, a_0)$  pe magistrala de date MD<sub>ex</sub>);
- 4) **W = 1** – înscrierea numărului  $A(a_{n-1}, \dots, a_1, a_0)$  în registrul multipoțional de pe magistrala de date MD<sub>ex</sub>;
- 5) **R = 1** – citirea numărului  $A(a_{n-1}, \dots, a_1, a_0)$  din registrul multipoțional 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, ..., 0,0) de pe magistrala de date MD<sub>in</sub>);
- 6) **W<sub>AC</sub> = 1** – înscrierea numărului  $A(a_{n-1}, \dots, a_1, a_0)$ , obținut la ieșirea sumatorului, în registrul acumulator;
- 7) **R<sub>AC</sub> = 1** – citirea numărului  $A(a_{n-1}, \dots, a_1, a_0)$  din registrul acumulator sau transmiterea lui pe magistralele de date MD<sub>ex</sub> și MD<sub>in</sub>;
- 8) **R<sub>B</sub> = 1** – citirea datelor din registrul B (transmiterea numărului  $B(b_{n-1}, \dots, b_1, b_0)$  pe magistrala de date MD<sub>ex</sub>);
- 9) **W = 1** – înscrierea numărului  $B(b_{n-1}, \dots, b_1, b_0)$  în registrul multipoțional de pe magistrala de date MD<sub>ex</sub>;
- 10) **R = 1** – citirea numărului  $B(b_{n-1}, \dots, b_1, b_0)$  din registrul multipoțional 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<sub>in</sub> și la ieșirea sumatorului se obține  $S(s_{n-1}, \dots, s_1, s_0)$ );
- 11) **W<sub>AC</sub> = 1** – înscrierea numărului  $S(s_{n-1}, \dots, s_1, s_0)$ , obținut la ieșirea sumatorului, în registrul acumulator;
- 12) **R<sub>AC</sub> = 1** – citirea numărului  $S(s_{n-1}, \dots, s_1, s_0)$  din registrul acumulator sau transmiterea lui pe magistralele de date MD<sub>ex</sub> și MD<sub>in</sub>;
- 13) **W<sub>A</sub> = 1** – înscrierea numărului  $S(s_{n-1}, \dots, s_1, s_0)$ , afișat pe magistrala de date MD<sub>ex</sub> 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.