



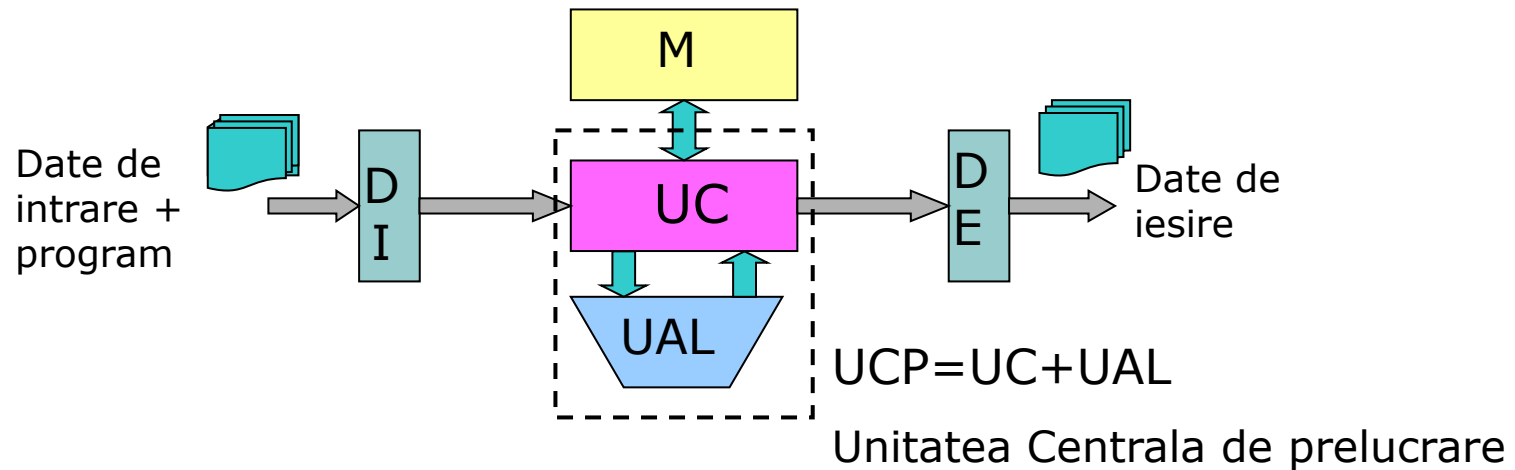
Sisteme cu microprocesoare

Cursul 2

Structura generala a unui calculator

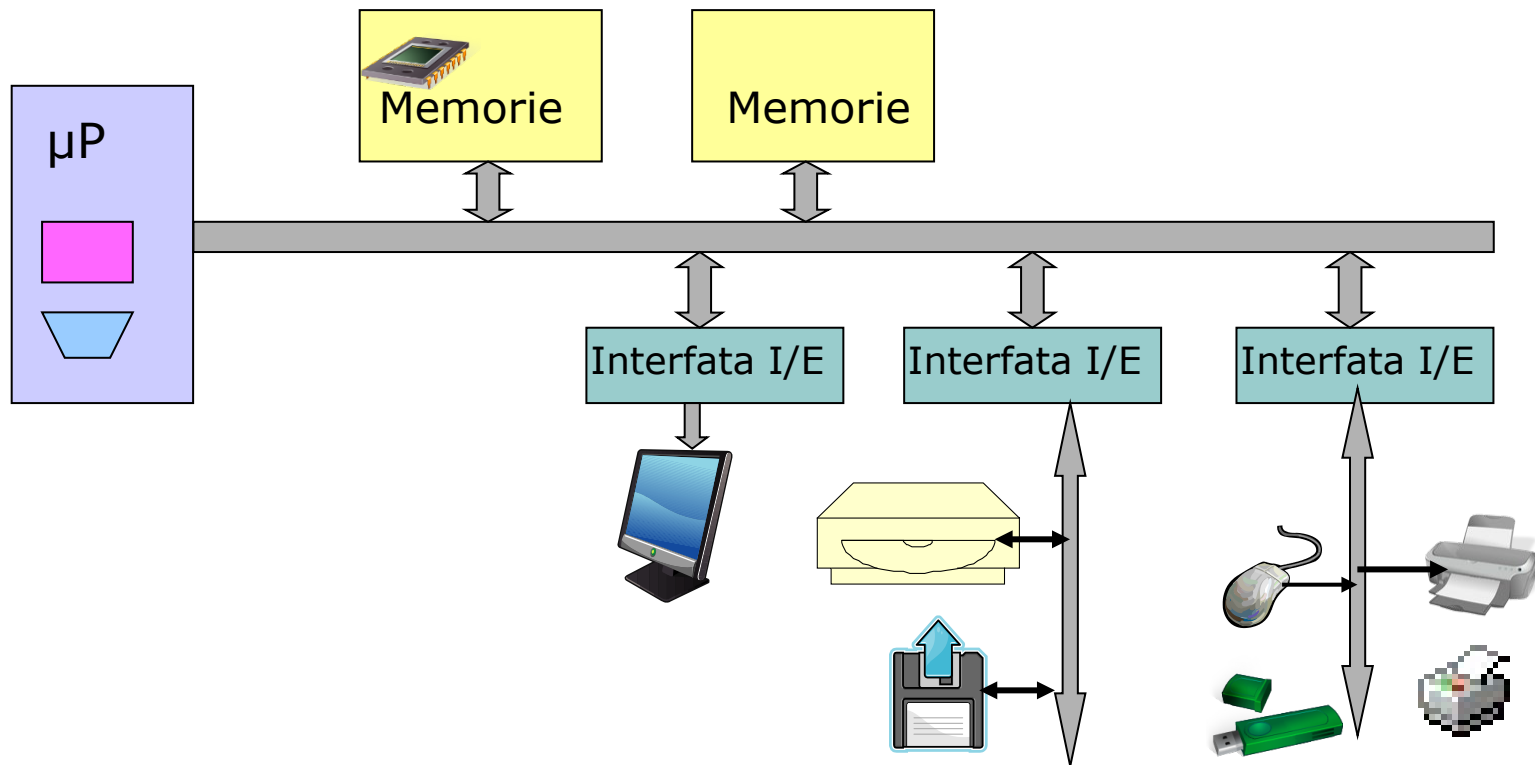
Componentele unui sistem de calcul

- Modelul clasic de calculator (J. von Neumann)
 - Unitatea de control (UC)
 - Unitatea aritmetico-logica (UAL)
 - Memoria (M)
 - Dispozitiv(e) de intrare (DI)
 - Dispozitiv(e) de iesire (DE)

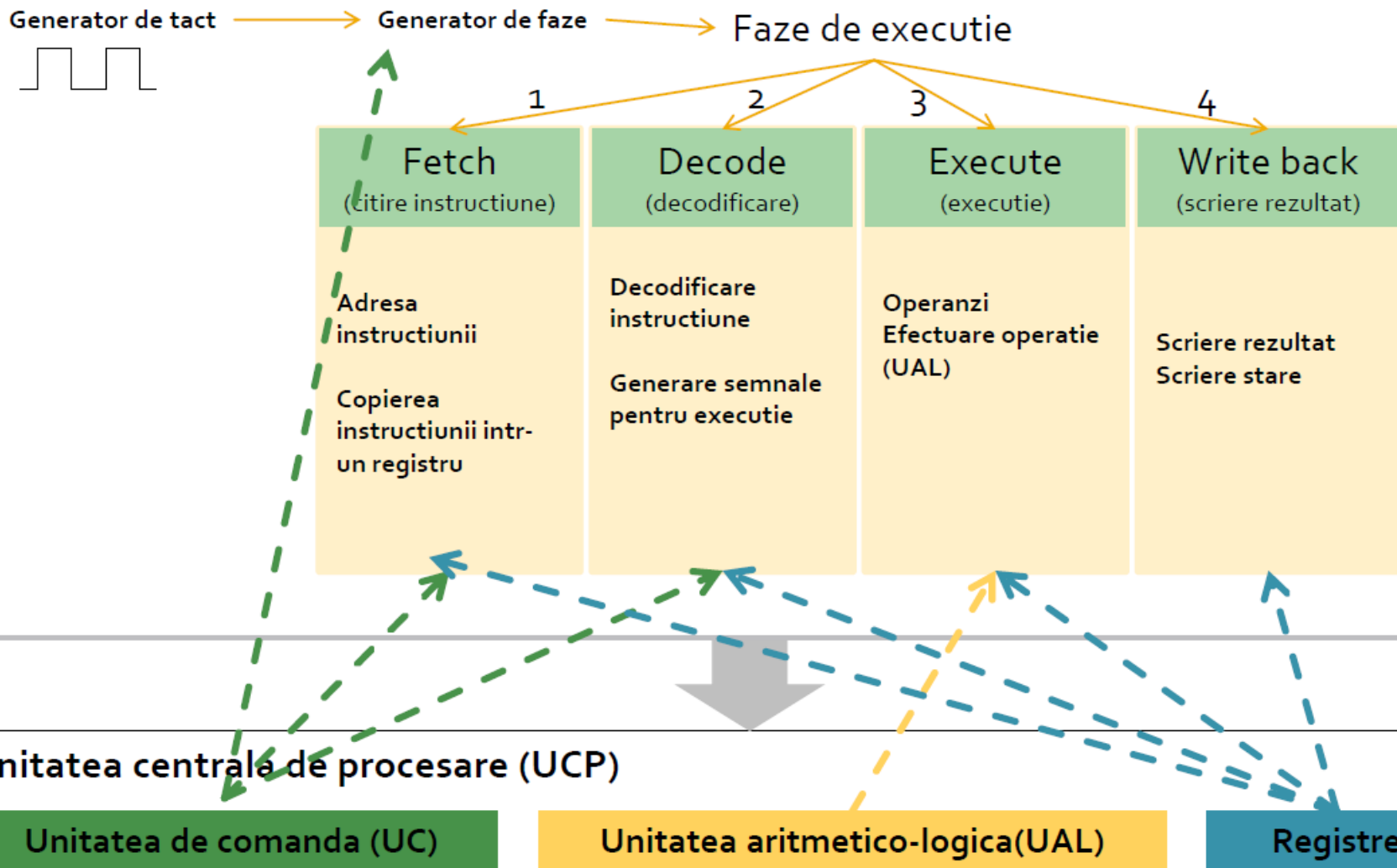


Componentele unui sistem de calcul

- Sistem de calcul bazat pe un microprocesor

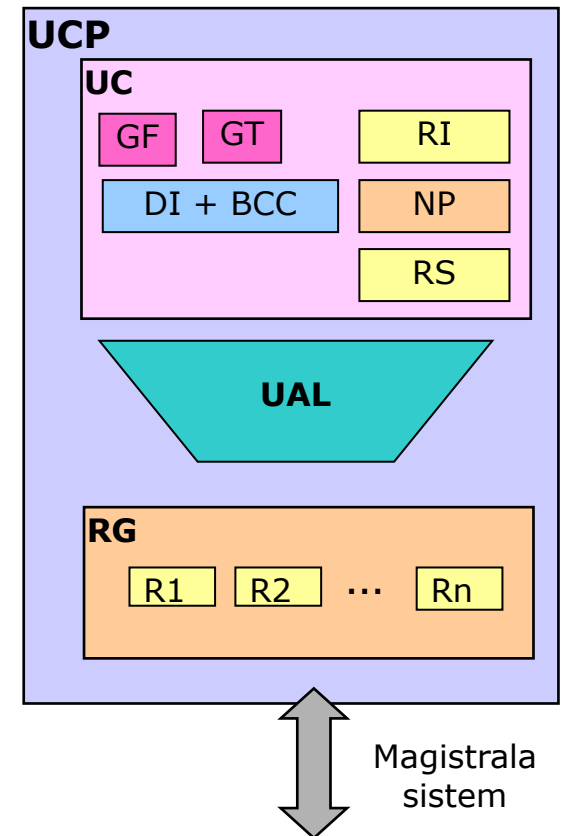


Executia instructiunilor



Unitatea Centrala de Prelucrare (UCP)

- Unitatea de Comanda (UC)
 - Responsabila pentru:
 - Citirea instructiunilor
 - Decodificarea instructiunilor
 - Generarea de semnale pentru executia instructiunilor
 - Este un circuit secvential (automat de stare)
- Unitatea Aritmetico-logica (UAL)
 - Efectueaza operatii aritmetice si logice:
 - Aritmetice: +, -, /, *, mod, comparare
 - Logice: SI, SAU, NU,
 - Deplasari/rotatii stanga/dreapta
 - Este (in general) un circuit combinational
- Registre
 - Registre generale (RG)
 - Pentru pastrarea temporara a datelor
 - Pentru efectuarea de operatii aritmetice si logice
 - Registre speciale (RS):
 - De adresare
 - De stare
 - De testare
 - De control

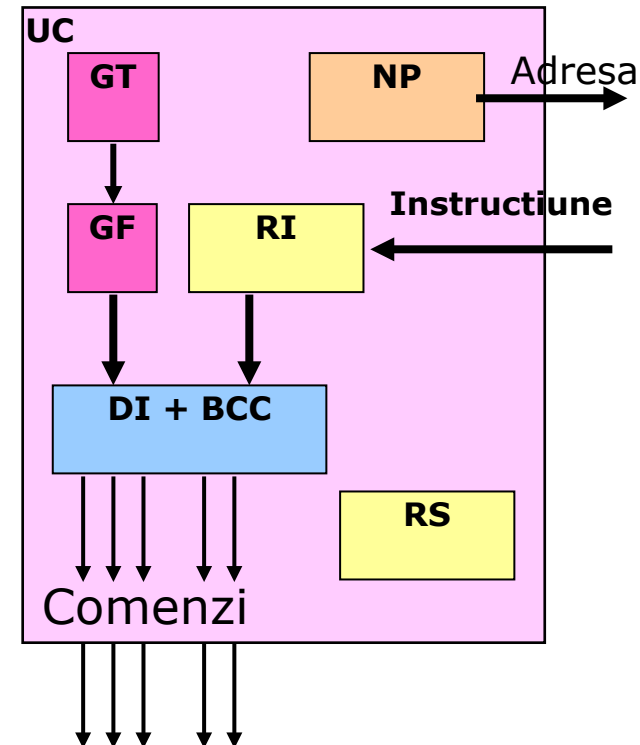


Unitatea de Comanda (UC)

Este creierul calculatorului

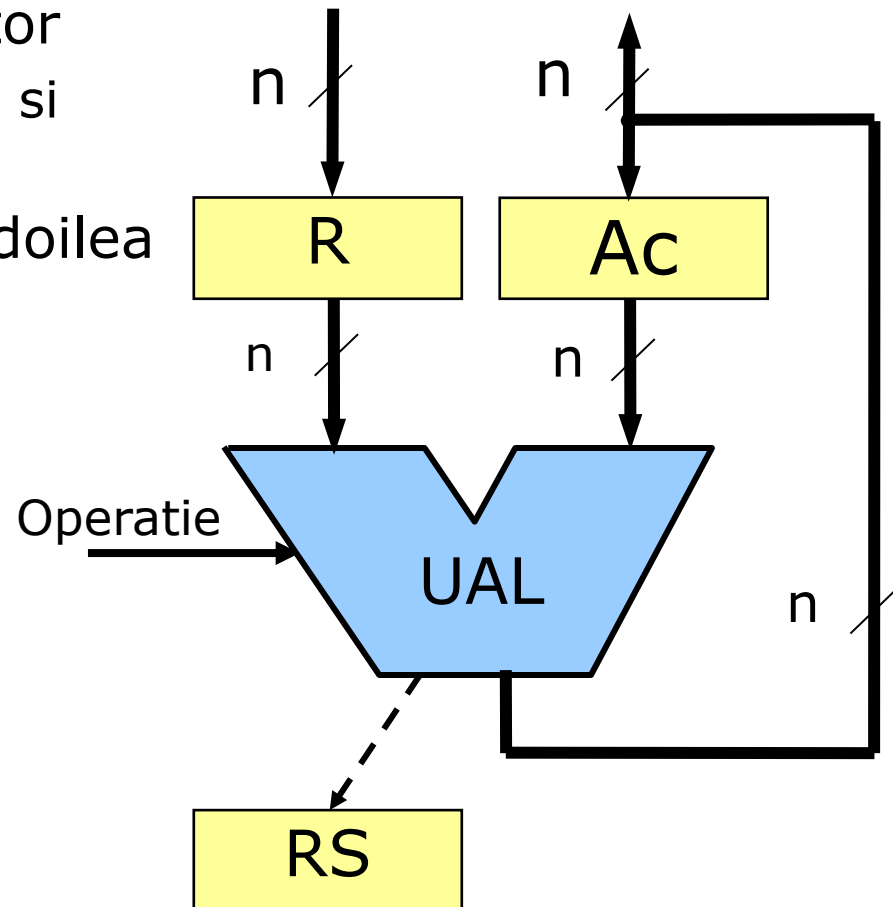
Se compune din:

- Generatorul de tact GT
 - Genereaza senalul de ceas
- Generatorul de faze GF
 - Genereaza fazele de executie
- Unitatea de decodificare instructiuni (DI) si blocul de comanda si control (BCC)
 - Interpreteaza instructiunile si genereaza semnalele de comanda necesare pentru executia lor
- Registrul de instructiuni RI
 - Pastreaza instructiunea curenta
- Numaratorul de program NP
 - Pastreaza adresa instructiunii urmatoare
- Registrul de stare program RS



Unitatea aritmetico-logica

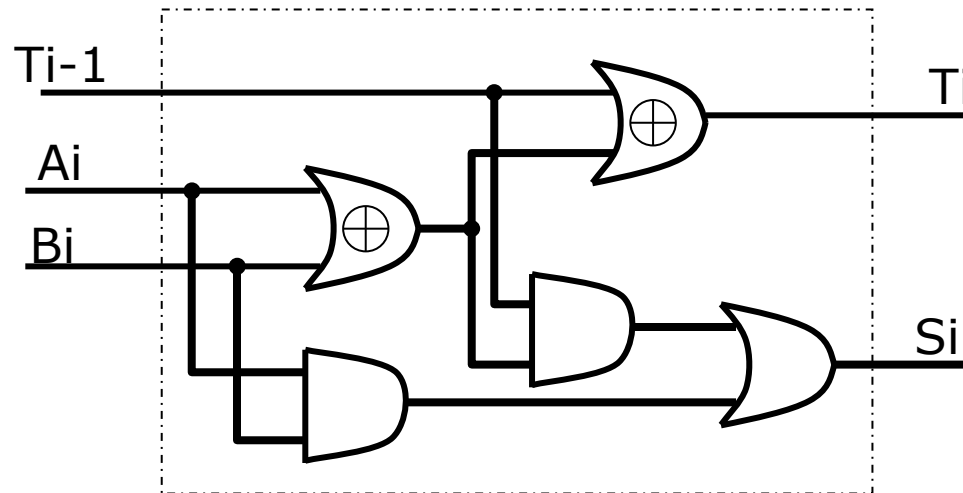
- Ac – registru acumulator
 - Pastreaza un operand si rezultatul operatiei
- R – registru pentru al doilea operand
- RS - registru de stare



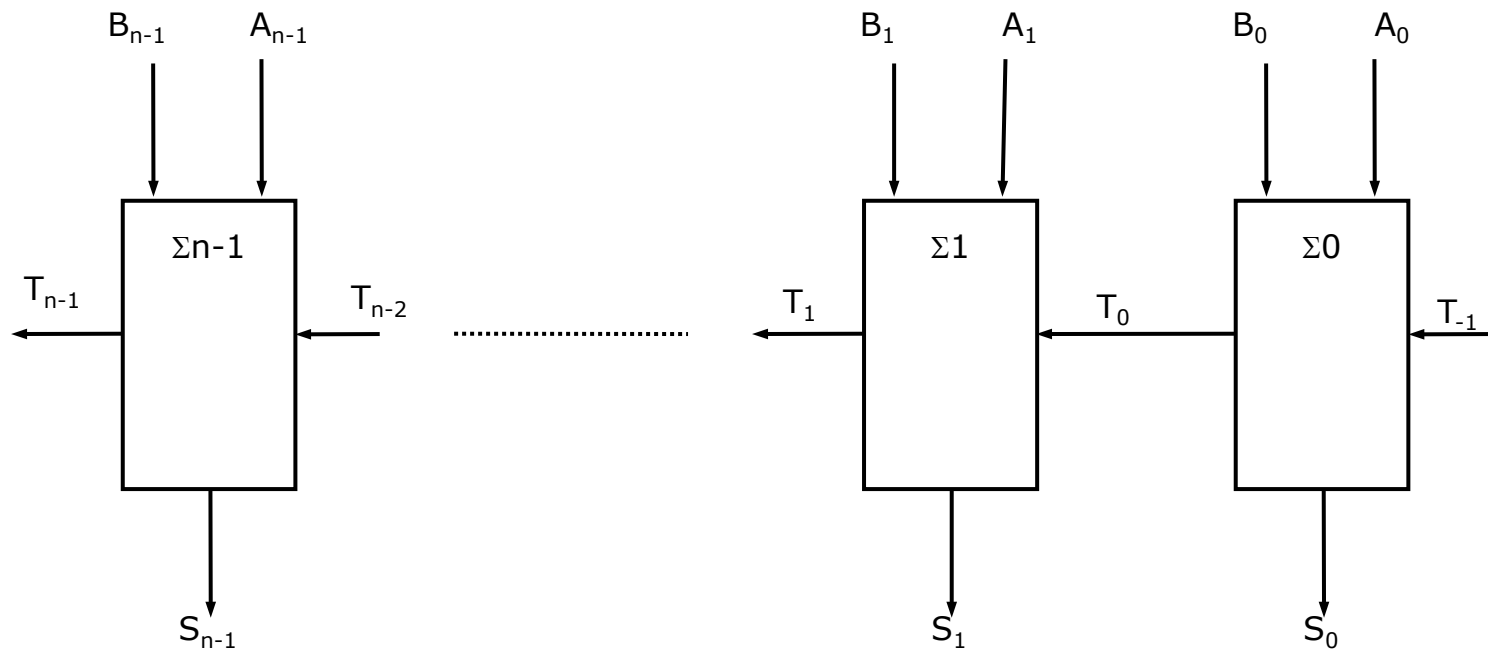
Unitatea Aritmetico-Logica

○ Sumator pe un bit

- $S_i = A_i \oplus B_i \oplus T_{i-1}$
- $T_i = A_i * B_i + T_{i-1} * (A_i \oplus B_i)$

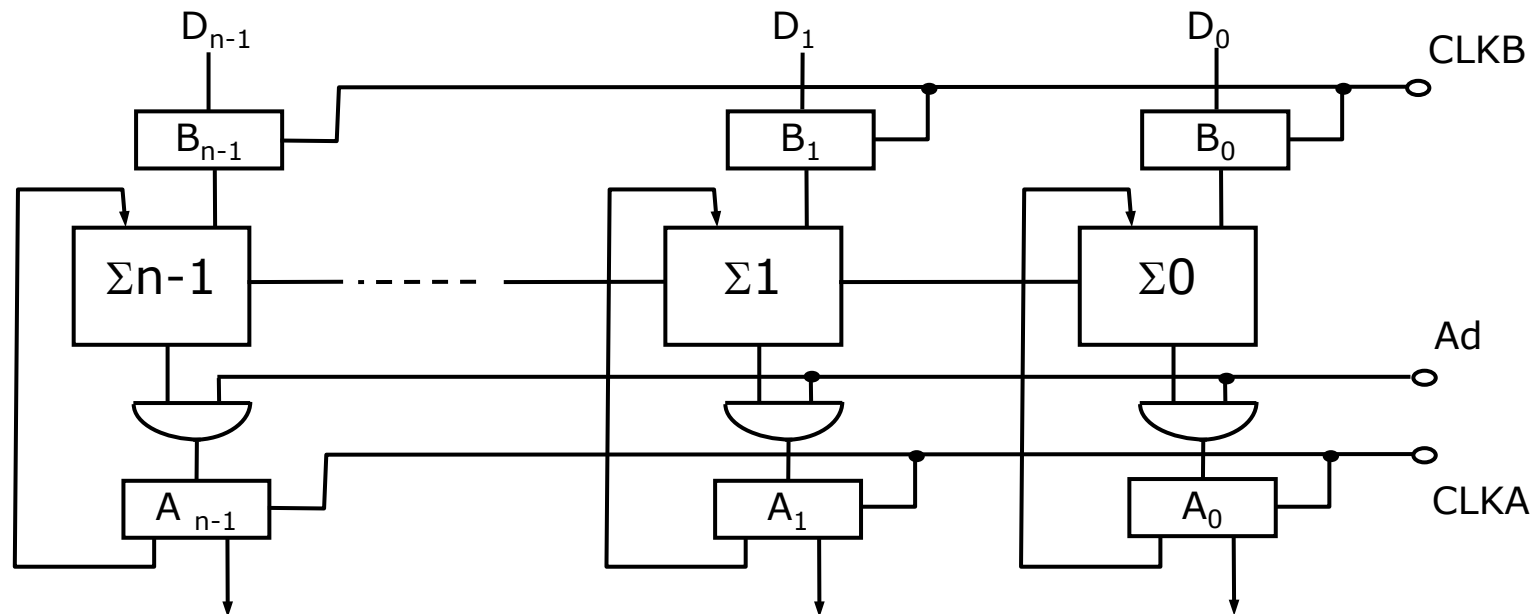


Sumator paralel pe n biti



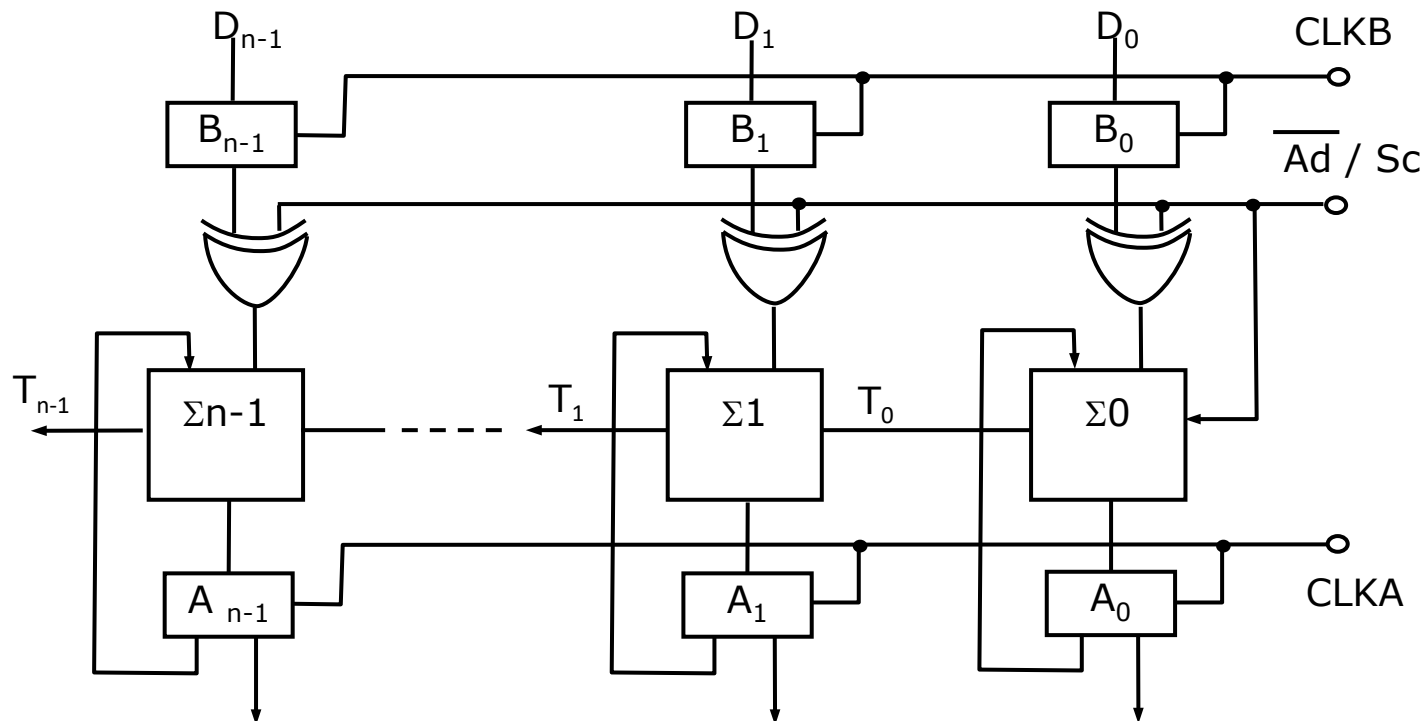
Operație de adunare cu operanzi în 2 registre

Pas	Operație	Comenzi
1	Șterge registrul A Înscrie registrul $B \leftarrow D$ (operand)	$Ad = 0, CLK_A = \Pi$ (impuls) $CLK_B = \Pi$
2	Transfer $B \rightarrow A$	$Ad = 1, CLK_A = \Pi$
3	$A \leftarrow A + B$	$Ad = 1, CLK_A = \Pi$

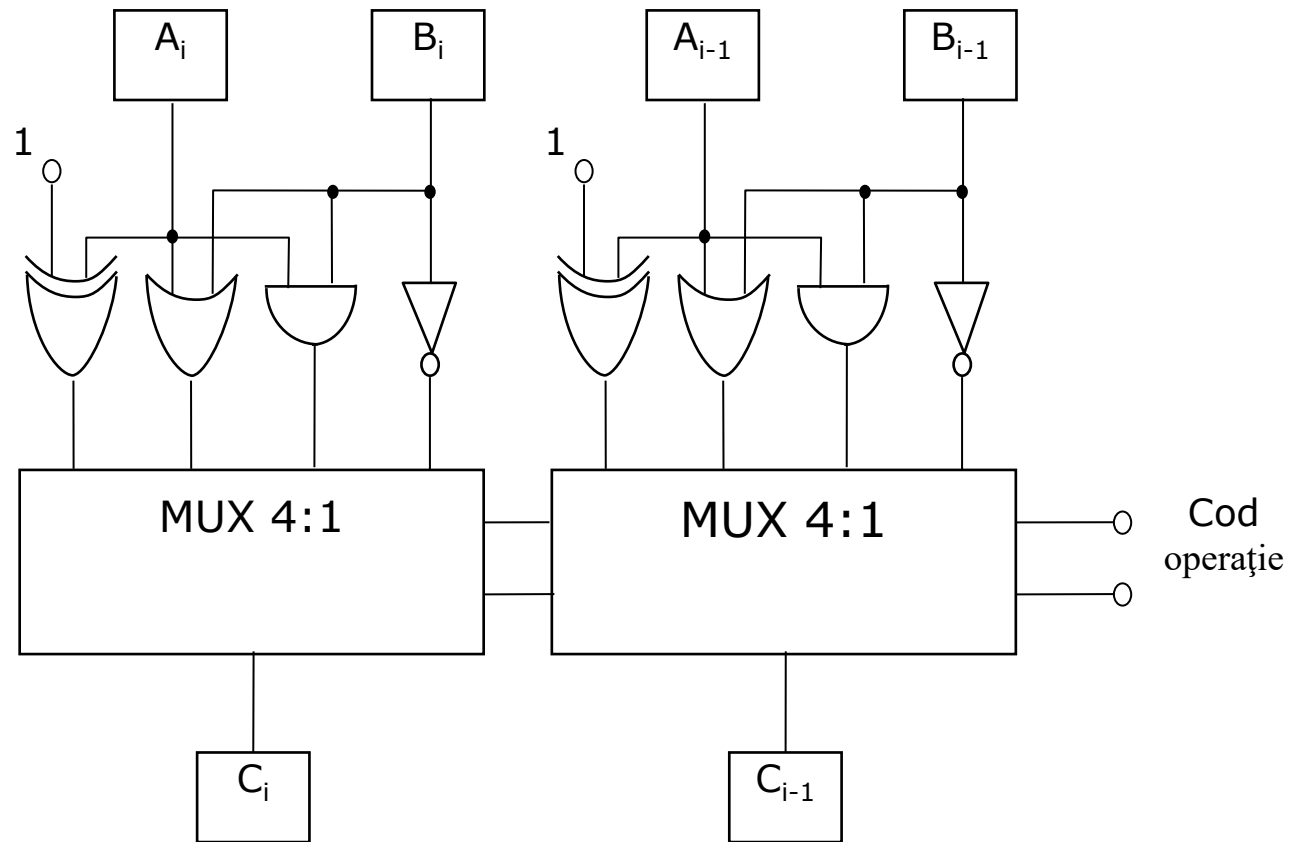


Circuit de adunare si scadere in complement fata de 2

- La scadere al doilea operand se complementeaza



Unitate logica pe un bit



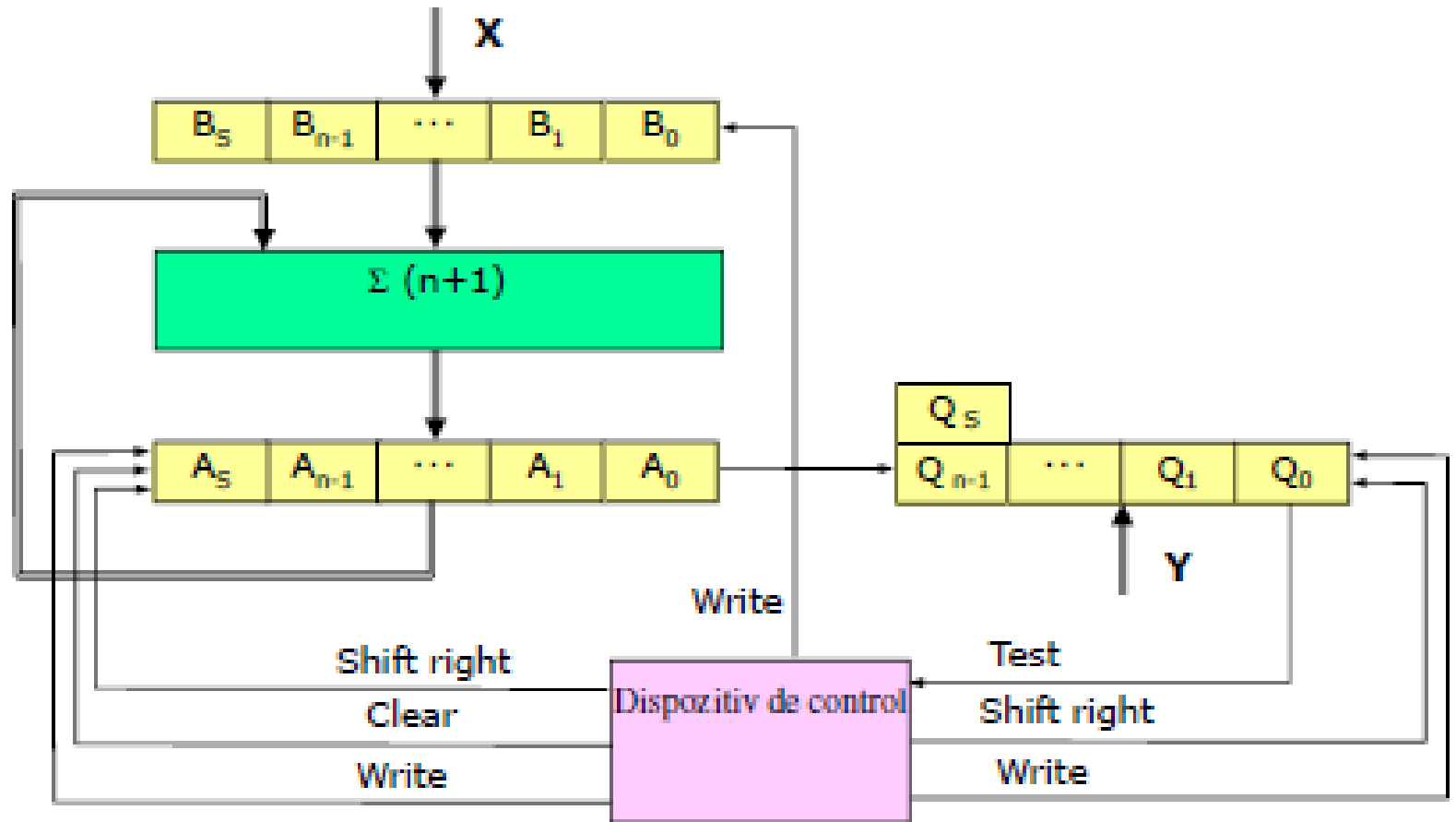
Operatia de inmultire

$$\begin{array}{r} 1100 * 12 * \\ 1010 \quad 10 \\ \hline 0000 \\ 1100 \\ 0000 \\ 1100 \\ \hline 1111000 = 78H = 120 \end{array}$$

Algoritm de înmulțire modificat:

	00000000		Acumulator (AC)
"0" →	0000000	0	Depl. dr.
"1" →	1100		Adunare
	0001100	0	Produs parțial
	000110	00	Depl. dr.
"0" →	00011	000	Depl. dr.
"1" →	1100		Adunare
	1111	000	Produs final

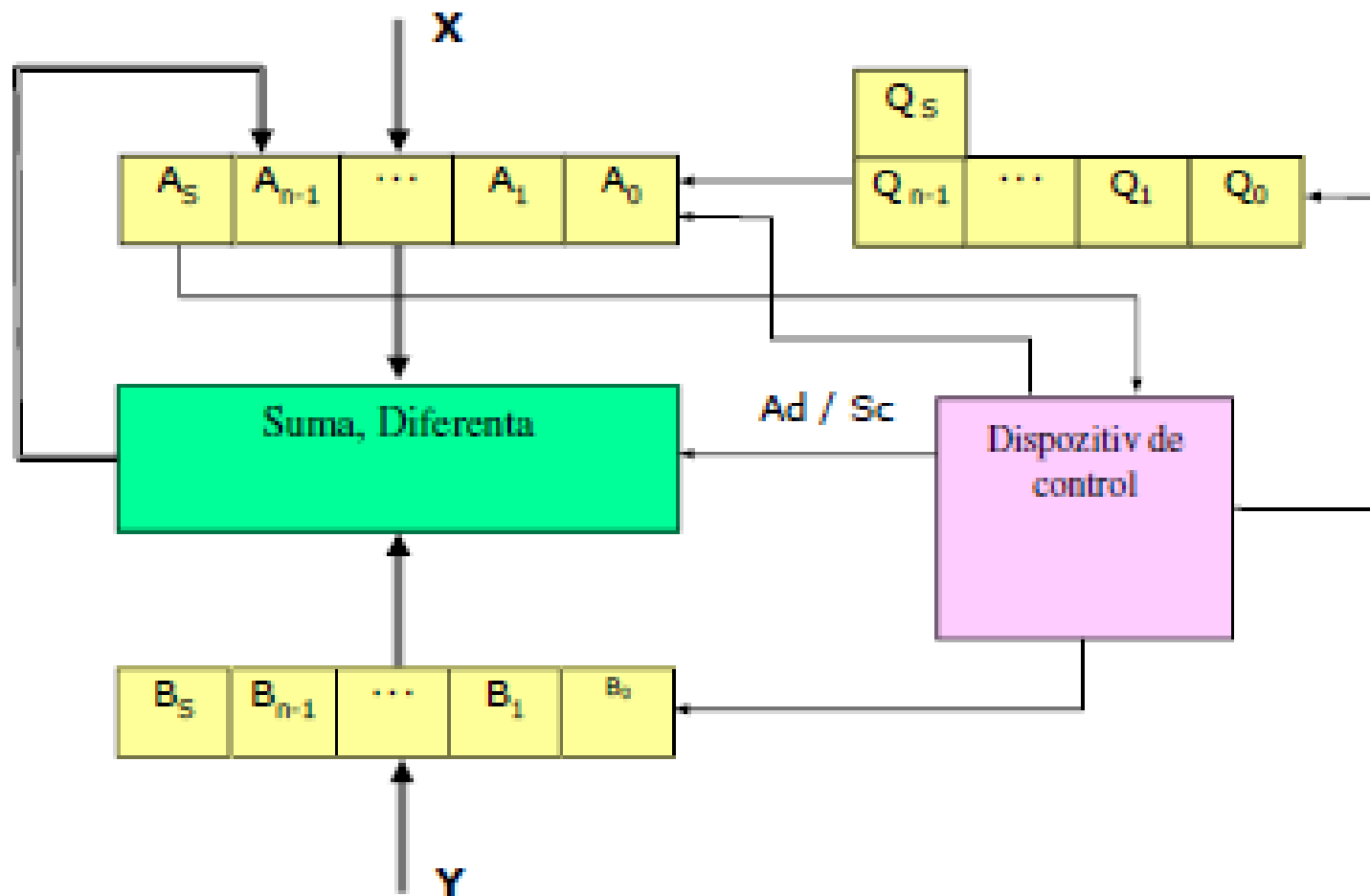
Implementarea operatiei de inmultire



Algoritm de inmultire

1. Înscriere operanzi în registre $B \leftarrow X, Q \leftarrow Y$, șterge acumulatorul $A \leftarrow 0$
2. Complementare operanzi dacă semnul este negativ „-”
3. Test Q_0
 - $Q_0 = 0$, deplasare dreapta A și Q
 - $Q_0 = 1$, adunare $A = B + A$ și deplasare dreapta A și Q
4. Se repetă pasul 3 până când Y_{n-1} ajunge în Q_0 . După ultima eventuală adunare nu se mai face deplasare.
5. $A_s = B_s + Q_s$
6. Complementare rezultat dacă $A_s = 1$

Circuit de impartire



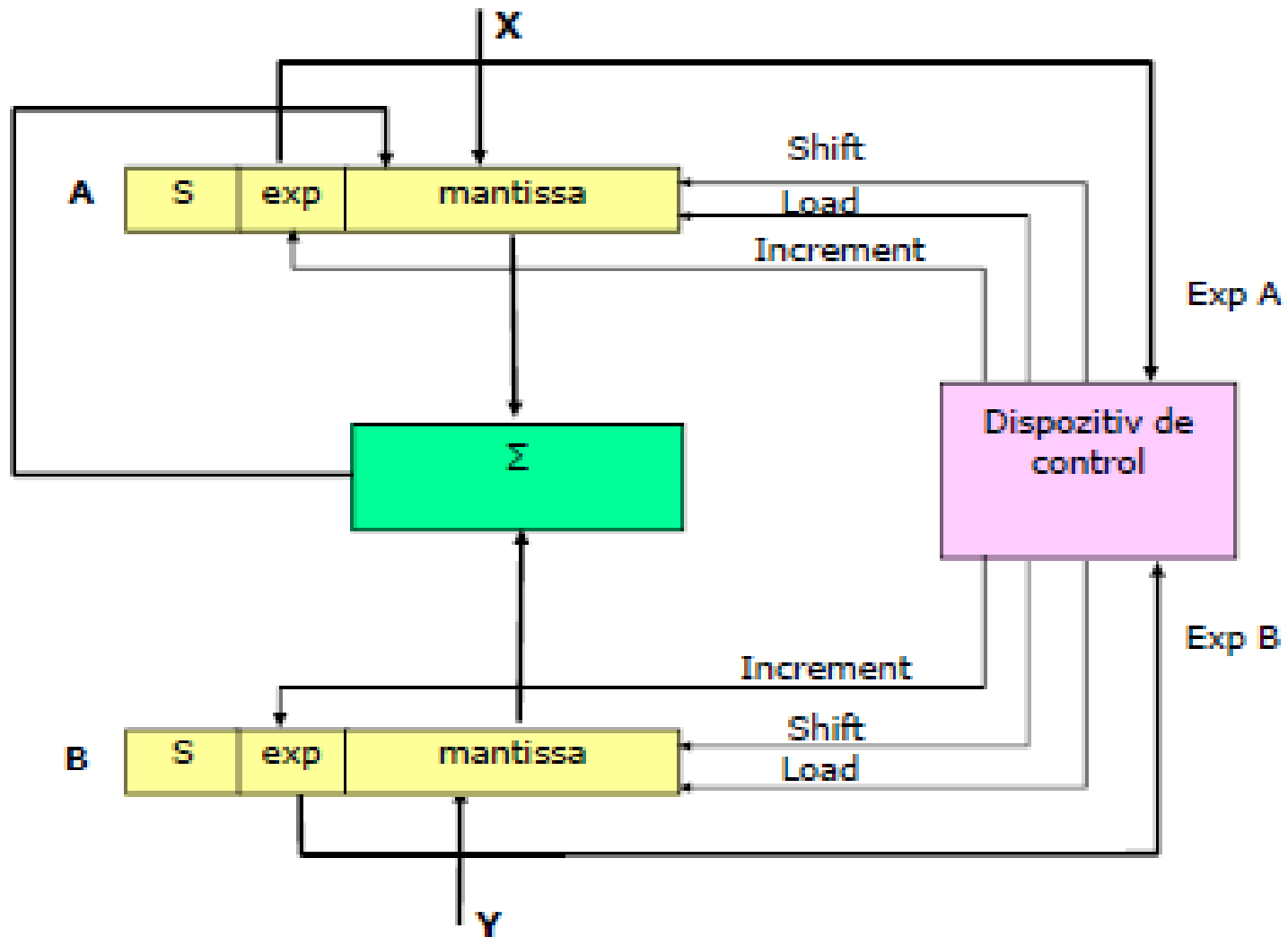
Secventa de impartire

1. Încarcă deîmpărțitul în A și Q
 Încarcă împărțitorul în B
2. Memorează $A_s + B_s$ în Q_s . Dacă
 $A_s = 1$, complementează A, Q
 $B_s = 1$, complementează B
3. Teste:
 - a. $A \geq B$, depășire de capacitate
 - b. $B = 0$, împărțire cu 0
 - c. $A = 0$ și $Q < B$, rezultat = 0
4. Deplasare A, Q la stânga și pune 0 în Q_0
5. Se scade B din A și rezultatul se pune în A. Dacă
 $A_s = 0$, deplasare stânga A, Q și pune 1 în Q_0
 $A_s = 1$, adună B la A, deplasare stânga A, Q și pune 0 în Q_0
6. Repetă pasul 5 de n ori
7. Rotunjirea rezultatului. Dacă $A \geq B$, se adaugă 1 la complementul registrului Q
8. Dacă $Q_s = 1$ se complementează registrul Q

Adunarea in virgula flotanta

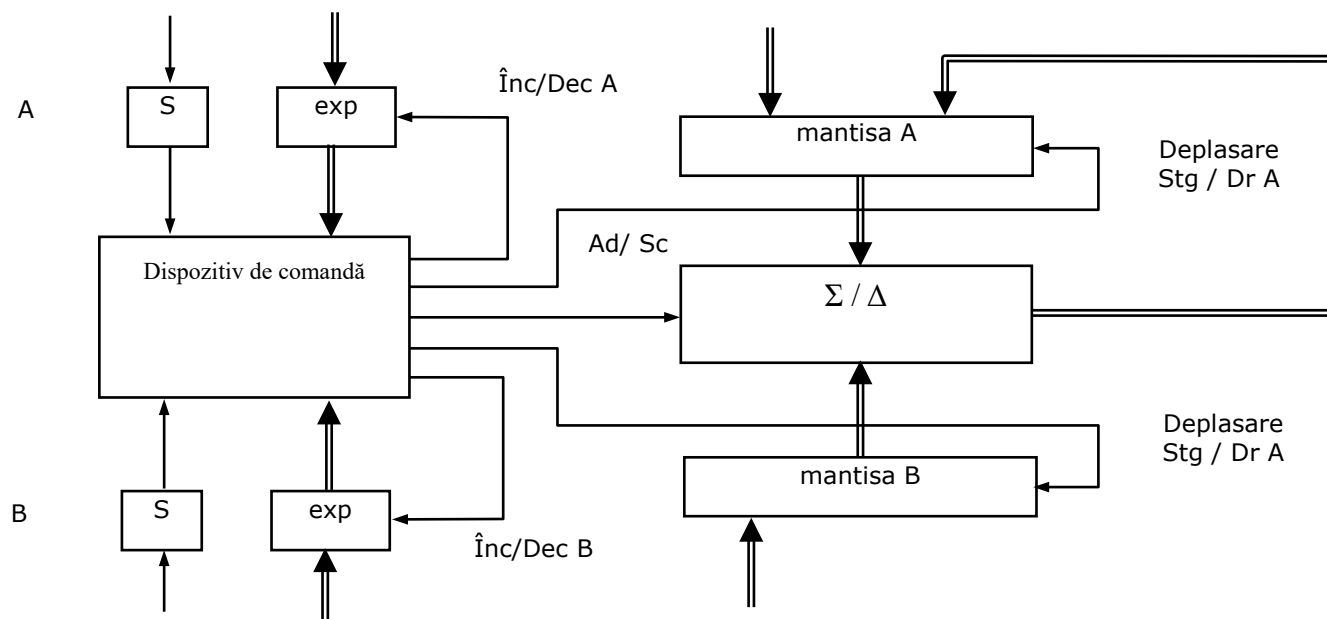
1. Încărcare operanzi
2. Se compară exponenții (5 cazuri):
 - $ex = ey$, se adună mantisele si se copiază exponentul
 - $ex > ey$ și $(ex - ey) < \text{numărul pozițiilor binare din mantisă}$, atunci mantisa my se aliniază prin deplasare la dreapta cu $ex-ey$ poziții și se adună mx cu my
 - $ex >> ey$ și $(ex - ey) \geq \text{numărul pozițiilor binare din mantisă}$, atunci se copiază X în rezultat
 - $ex < ey$ și $(ey - ex) < \text{numărul pozițiilor binare din mantisă}$, atunci mantisa mx se aliniază prin deplasare la dreapta cu $ey-ex$ poziții și se adună mx cu my
 - $ex << ey$ și $(ey - ex) \geq \text{numărul pozițiilor binare din mantisă}$, atunci se copiază Y în rezultat
3. Adunare
4. Realiniere. Test bit de după și dinainte de virgulă, deplasare stânga sau dreapta, incrementare sau decrementare exponent

Circuit de adunare in Virgula flotanta



Inmultirea si impartirea in virgula flotanta

- Înmulțirea se realizează prin următoarea secvență:
 - Adunare exponenți
 - Multiplicare mantise
 - Realiniere mantisă rezultat si decrementare exponent
- Împărțire:
 - Scădere exponenți
 - Împărțire mantise
 - Realiniere mantise



Proiectarea unui calculator simplu

Etape de proiectare:

1. Se stabilesc destinația și domeniul de utilizare al calculatorului;
- 2° Se definește setul de instrucțiuni și formatul instrucțiune;
- 3° Se definește schema bloc a unității centrale;
- 4° Se descompune fiecare instrucțiune în faze și microoperații;
- 5° Se determină ecuațiile microcomenzilor generate de unitatea de comanda;
- 6° Se construiește schema logică a modulelor GF și BCC;
- 7° Se construiesc schemele logice pentru alte module: RI, NP, RG, UAL, RS, GT;
- 8° Proiectarea memoriei;
- 9° Proiectarea interfețelor de intrare/ ieșire.
- 10° Se reiau etapele 2, 3 și 4 până la obținerea unor rezultate optime conform criteriilor stabilite în etapa 1.

Calculator simplu:

Pasi de proiectare

- Destinatia:
 - Calculatoare de uz general
 - Sisteme de calcul cu destinatie speciala:
 - Sisteme incapsulate
 - Sisteme de procesare a semnalelor
 - Sisteme de control
 - Sisteme de calcul de inalta performanta:
 - Sisteme paralele
- Setul de instructiuni:
 - Formatul instructiunilor: (lungime si campuri)
 - Fix:
 - variabil
 - Tipuri de operatii:
 - Aritmetice
 - Logice
 - De transfer
 - De salt
 - Operatii cu stiva