

# ARHITECTURA SISTEMELOR DE CALCUL ȘI SISTEME DE OPERARE

## LUCRĂRILE DE LABORATOR Nr. 3, 4 și 5

### SISTEME DE NUMERAȚIE. CONVERSII. OPERAȚII CU BAZE MARI DE NUMERAȚIE

#### I. SCOPUL LUCRĂRILOR

Lucrările prezintă reprezentarea informațiilor numerice întregi pozitive sau cel puțin zero. Scopul lucrării constă în a familiariza studenții cu sistemele de numerație conversii între sisteme de numerație, operații aritmetice cu baze mari de numerație.

#### II. NOȚIUNI TEORETICE

##### 1 Reprezentarea informației

Informațiile prelucrate prin sistemele de calcul sunt de diverse tipuri dar ele sunt reprezentate la nivel elementar sub formă binară. O informație elementară corespunde deci unei cifre binare (0 sau 1) numită **bit**. O informație mai complexă (un caracter, un număr etc.) se exprimă printr-o mulțime de biți.

**Codificarea** unei informații constă în a stabili o corespondență între reprezentarea externă a informației (caracterul A sau numărul 33, de exemplu) și reprezentarea sa internă, care este o secvență de biți.

Avantajele reprezentării binare se referă în special la facilitatea de realizare tehnică cu ajutorul elementelor bistabile (sisteme cu 2 stări de echilibru) precum și la simplitatea efectuării operațiilor fundamentale sub forma unor circuite logice, utilizând logica simbolică cu două stări (0, 1).

Informațiile prelucrate în sistemele de calcul sunt de două tipuri: **instrucțiuni** și **date**.

**Instrucțiunile**, scrise în limbaj mașină, reprezintă operațiile efectuate în sistemul de calcul și ele sunt compuse din mai multe câmpuri:

- **codul** operației de efectuat;
- **operanzii** implicați în operație.

Codul operației trebuie să suporte o operație de **decodificare** (transformare inversă codificării) pentru a se putea efectiv executa.

**Datele** sunt operanzii asupra cărora acționează operațiile (prelucrările), sau sunt produse de către acestea. O adunare, de exemplu, se aplică la doi operanzi, furnizând un rezultat care este suma acestora.

Se pot distinge datele numerice, rezultat al unei operații aritmetice, sau date nenumerice, de exemplu simbolurile care constituie un text.

##### 2. Date nenumerice

Datele nenumerice corespund caracterelor alfanumerice: A, B, ..., Z, a, b, ..., z, 0, 1, ..., 9 și caracterelor speciale: ?, !, ", \$, ;, ...

Codificarea se realizează pe baza unei tabele de corespondență specifică fiecărui cod utilizat. Printre cele mai cunoscute coduri putem enumera:

- **BCD** [Binary Coded Decimal] prin care un caracter este codificat pe 6 biți;
- **ASCII** [American Standard Code for Information Interchange] (7 biți);
- **EBCDIC** [Extended Binary Coded Decimal Internal Code] (8 biți).

**Tabela de corespondență între diferite coduri**

caracter	BCD	ASCII	EBCDIC
0	000000	0110000	11110000
1	000001	0110001	11110001
2	000010	0110010	11110010
...	...	...	...
9	001001	0111001	11111001
A	010001	1000001	11000001
B	010010	1000010	11000010
C	010011	1000011	11000011
	(6 biți)	(7 biți)	(8 biți)

### 3. Datele numerice

Datele numerice sunt de următoarele tipuri:

- numere întregi pozitive sau nule:** 0; 1; 315...
- numere întregi negative:** -1; -155...
- numere fracționare:** 3.1415; -0.5...
- numere în notație științifică:**  $4.9 \times 10^7$ ;  $10^{23}$  ...

Codificarea se realizează cu ajutorul unui **algoritm de conversie** asociat tipului de dată corespunzător. Operațiile aritmetice (adunare, scădere, înmulțire, împărțire) care se pot aplica asupra acestor date se efectuează de regulă în aritmetica binară.

Tabele de adunare și înmulțire binară

$0 + 0 = 0$	$0 \times 0 = 0$
$0 + 1 = 1$	$0 \times 1 = 0$
$1 + 0 = 1$	$1 \times 0 = 0$
$1 + 1 = 10$	$1 \times 1 = 1$

**Numerele întregi pozitive sau nule** cuprind: 0, 1, 2, ..., N, N + 1, ...

#### Sisteme de numerație

Un sistem de numerație face să-i corespundă unui număr N, un anumit simbolism scris și oral. #ntr-un sistem de numerație cu baza  $p > 1$ , numerele 0, 1, 2, ...,  $p - 1$  sunt numite **cifre**.

Orice număr întreg pozitiv poate fi reprezentat astfel:  $N = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0$  cu  $a_i \in \{0, 1, 2, p-1\}$  și  $a_n \neq 0$ . Se utilizează de asemenea notația echivalentă  $N = a_n a_{n-1} \dots a_1 a_0$ .

Numerele scrise în sistenul de numerație cu baza 2 (binar) sunt adesea compuse dintr-un mare număr de biți, și de aceea se preferă exprimarea acestora în sistemele **octal** ( $p = 8$ ) și **hexazecimal** ( $p = 16$ ), deoarece conversia cu sistemul binar este foarte simplă.

#### 4. Schimbări de bază

a) **binar**  $\Rightarrow$  **zecimal**:

Conversia se realizează prin însumarea puterilor lui 2 corespunzătoare biților egali cu 1;

*Exemplu:*  $10101_2 = 2^4 + 2^2 + 2^0 = 16 + 4 + 1 = 21_{10}$

b) **zecimal  $\Rightarrow$  binar.**

Conversia se efectuează prin împărțiri întregi succesive cu 2. Testul de oprire corespunde situației câtului nul. Numărul binar este obținut considerând resturile în ordinea inversă.

*Exemplu:* Conversia lui 26:

$$\begin{array}{r} 26 : 2 = 13 \quad \text{rest } 0 \\ 13 : 2 = 6 \quad \text{rest } 1 \\ 6 : 2 = 3 \quad \text{rest } 0 \\ 3 : 2 = 1 \quad \text{rest } 1 \\ 1 : 2 = 0 \quad \text{rest } 1 \end{array}$$

Se obține (de jos în sus):  $26_{10} = 11010_2$ .

c) **octal (hexazecimal)  $\Rightarrow$  zecimal**

Conversia se reduce la însumarea puterilor lui 8 (16).

d) **zecimal  $\Rightarrow$  octal (hexazecimal)**

Conversia se efectuează prin împărțiri întregi succesive prin 8 (16). Testul de oprire corespunde situației câtului nul. Numărul octal (hexazecimal) este obținut considerând resturile obținute de la ultimul către primul.

e) **Octal (hexazecimal)  $\Rightarrow$  binar**

Conversia corespunde dezvoltării fiecărei cifre octale (hexazecimale) în echivalentul ei binar pe 3 (4) biți.

*Exemplu:*

$$27_8 = 010'111_2 \text{ deoarece } 2_8 = 010_2 \text{ și } 7_8 = 111_2.$$

$$3A_{16} = 0011'1010_2 \text{ deoarece } 3_{16} = 0011_2 \text{ și } A_{16} = 1010_2.$$

f) **binar  $\Rightarrow$  octal (hexazecimal)**

Conversia se realizează înlocuind de la dreapta la stânga, grupele de 3 (4) biți prin cifra octală (hexazecimală) corespunzătoare. Dacă numărul de biți nu este multiplu de 3 (4) se completează configurația binară la stânga cu zerouri.

*Exemplu:*  $101011_2 = 53_8 = 2B_{16}$ .

## 5. Numere fracționare

Numerele fracționare sunt numerele subunitare.

### Schimbări de bază

a) **binar  $\Rightarrow$  zecimal**

Conversia se face adunând puterile corespunzătoare ale lui 2.

*Exemplu:*  $0.01_2 = 0 \times 2^{-1} + 1 \times 2^{-2} = 0.25_{10}$ .

b) **zecimal  $\Rightarrow$  binar**

Conversia se efectuează prin înmulțiri succesive cu 2 a numerelor pur fracționare. Acest algoritm trebuie să se termine când se obține o parte fracționară nulă sau când numărul de biți obținuți corespunde mărimii registrului sau a cuvântului de memorie în care se va stoca valoarea. Numărul binar se obține citind părțile întregi în ordinea calculării lor.

*Exemplu:*

$$\begin{array}{r} 0.125 \times 2 = 0.250 \quad = 0 + 0.250 \\ 0.25 \times 2 = 0.50 \quad = 0 + 0.50 \\ 0.50 \times 2 = 1.0 \quad = 1 + 0.0 \end{array}$$

Vom considera părțile întregi de sus în jos, deci:  $0.25_{10} = 0.001_2$ .

### III. MODUL DE LUCRU

Se exemplifică reprezentarea informațiilor în sisteme de numerație. Se realizează implementarea algoritmilor de împărțiri succesive, de înmulțiri succesive și algoritmul de substituție. Se aprofundează operațiile aritmetice cu numere mari scrise în diverse baze de numerație.

### IV. CONȚINUTUL REFERATULUI

1. Sumarul noțiunilor întâlnite.
2. Rezolvați următoarele probleme:
  - 2.1 Exprimați în binar, octal, hexazecimal numerele zecimale următoare: 32.625 și 128.75.
  - 2.2 Converteți în zecimal numerele următoare, baza fiind indicată în indice:  $DADA.C_{16}$ ,  $270.14_8$ ,  $11011.01111_2$ .
  - 2.3 Program de calcul al corespondentului binar, octal, hexazecimal pentru un număr natural dat, introdus de la tastatură.
  - 2.4 Program care realizează conversia în zecimal a unui număr scris într-un sistem de numerație cu o bază oarecare  $p$  (se citesc de la tastatură baza  $p$  și cifrele numărului în baza  $p$ ).
  - 2.5 Program care realizează conversia unui număr întreg din baza  $p$  în baza  $q$  (se citesc de la tastatură bazele  $p$  și  $q$  precum și cifrele numărului în baza  $p$ ).
  - 2.6 Program care realizează conversia unui număr fracționar din baza  $p$  în baza  $q$  (se citesc de la tastatură bazele  $p$  și  $q$  precum și cifrele numărului în baza  $p$ ).
  - 2.7 Program care realizează suma a două numere scrise în baza  $p$  (se citesc de la tastatură baza  $p$  precum și cifrele celor numere în baza  $p$ ).
  - 2.8 Program care realizează diferența a două numere scrise în baza  $p$  (se citesc de la tastatură baza  $p$  precum și cifrele celor numere în baza  $p$ ).
  - 2.9 Program care realizează produsul a două numere scrise în baza  $p$  (se citesc de la tastatură baza  $p$  precum și cifrele celor numere în baza  $p$ ).
  - 2.10 Program care realizează câtul a două numere scrise în baza  $p$  (se citesc de la tastatură baza  $p$  precum și cifrele celor numere în baza  $p$ ).
3. Observații și concluzii personale.