

Lecția 5

Reprezentarea datelor în memoria calculatorului

În funcție de valoarea pe care o au datele numerice ele pot reprezenta numere naturale, numere întregi sau numere reale, iar pentru fiecare astfel de număr se vor aloca în memorie un număr diferit de octeți.

Reprezentarea numerelor naturale

Formatul de reprezentare întreg fără semn nu presupune nimic în plus decât trecerea de la baza 10 la baza 2. În aceasta reprezentare putem scrie spre exemplu:

$$\text{numărul } (188)_{10} = (10110110)_2$$

$$(1 \cdot 10_2 + 8 \cdot 10_1 + 8 \cdot 10_0)_{10} = (0 \cdot 2_0 + 1 \cdot 2_1 + 1 \cdot 2_2 + 0 \cdot 2_3 + 1 \cdot 2_4 + 1 \cdot 2_5 + 0 \cdot 2_6 + 1 \cdot 2_7)_2$$

O reprezentare a unui număr într-o bază de numerație notată B folosește doar cifre din intervalul $[0, B-1]$. Trecerea unui număr de la o bază la alta presupune aflarea unei secvențe unice de resturi conform algoritmului următor:

- numărul se împarte la baza și se reține restul obținut. Acesta constituie ultima cifră din reprezentare în acea bază.
- câtul obținut în urma acestei prime împărțiri este din nou împărțit la baza și se reține iarăși restul. Acest rest constituie a doua cifră, în ordine de la dreapta la stânga, a reprezentării în baza aleasă.
- algoritmul continuă până când câtul obținut este mai mic decât baza. Acest cât va reprezenta cea mai semnificativă cifră a numărului în baza dată.

În următoarea imagine este prezentat modul de transformare a numărului 25 în baza 2: Reprezentarea în baza 2 a numărului 25 se obține luând în ordine inversă resturile împărțirii la 2.

Pentru a putea memora un număr natural în memoria internă a calculatorului se pot alocă 1, 2 sau 4 octeți. Pe un octet se pot memora numere naturale cuprinse în intervalul $[0..255]$

adică numărul natural $0 = 0 \cdot 2_0 + 0 \cdot 2_1 + 0 \cdot 2_2 + 0 \cdot 2_3 + 0 \cdot 2_4 + 0 \cdot 2_5 + 0 \cdot 2_6 + 0 \cdot 2_7$.

adică numărul natural $255 = 1 \cdot 2_0 + 1 \cdot 2_1 + 1 \cdot 2_2 + 1 \cdot 2_3 + 1 \cdot 2_4 + 1 \cdot 2_5 + 1 \cdot 2_6 + 1 \cdot 2_7 = 1 + 2 + 4 + 8 + 16 + 32 + 64 + 128$

Propun ca exercițiu de calcul reprezentarea intervalului de numere naturale care pot fi memorate pe 2 octeți.

De exemplu, numerele întregi reprezentate pe 32 bits (4 octeți) pot lua valori cuprinse în intervalul $[-2147483648, 2147483647]$. Adunarea, scăderea și înmulțirea acestor numere returnează rezultate exacte atâta vreme cât nu se depășesc limitele precizate. În cazul în care apare o depășire (*overflow*) rezultatul este imprevizibil. Rezultatul împărțirii numerelor întregi reprezentate este tot un număr întreg obținut prin rotunjirea rezultatului corect la întregul imediat mai mic.

Reprezentarea numerelor întregi

Numerele întregi se reprezintă în memorie în complement față de 2. Reprezentarea în complement față de 2 este folosită de majoritatea procesoarelor actuale. Ideea acestei reprezentări se înțelege pe baza observației simple că numărul x adunat cu numărul $-x$ trebuie să dea întotdeauna valoarea 0. Întrebarea ce se pune este *cum se scrie $-x$* , deoarece pentru numerele întregi pozitive se păstrează regula de la paragraful precedent. Cel mai simplu este să folosim un bit pentru reprezentarea semnului (de ex., 0 înseamnă plus și 1 înseamnă minus) și ceilalți $(n-1)$ biți pentru reprezentarea valorii absolute.

Numerele negative se reprezintă astfel:

- plecăm de la reprezentarea în baza 2 a opusului (care este pozitiv),
- inversăm toate cifrele până la ultimul 1 (exclusiv)

Exemple:

$(22)_{10} = (00010110)_2$, număr pozitiv

$(-22)_{10} = (11101010)_2$, număr negativ

Primul bit (cel scris cu culoare roșie) se mai numește *bit de semn* deoarece el indică semnul numărului (este 0 dacă și numai dacă numărul este pozitiv). Avantajul acestei reprezentări este următorul: *algoritmii de adunare și de scădere a două numere reprezentate în cod complementar sunt identici cu algoritmii de adunare și de scădere pentru numere pozitive reprezentate în baza 2.*

Pe un octet se pot memora numere naturale cuprinse în intervalul [-128..+127]

Reprezentarea numerelor reale

Modul în care numerele reale sunt reprezentate în memorie este reglementat de *IEEE[1] Floating Point Standard[2]*, un set de reguli care ne ajută în prezicerea gradului de inexactitate a rezultatelor calculelor aritmetice. Una dintre sursele de erori în calculul științific este *eroarea de rotunjire* datorată calculelor aritmetice inexacte efectuate de calculator. Eroarea de rotunjire face parte din categoria erorilor inevitabile dar consecințele sale diferă de la o problema la alta, de la o strategie de rezolvare la alta. O singură operație aritmetică efectuată de calculator cu numere reale va da de cele mai multe ori un răspuns cu un grad relativ mare de precizie.

Pentru reprezentarea numerelor în calculator se folosesc două formate de reprezentare:

- *formatul cu virgulă fixă:*

± partea întreagă. partea zecimală de exemplu: 12.351 sau -54.789

- *formatul cu virgulă mobilă:* s-a impus datorită necesității de a reprezenta în sistemele de calcul numere foarte mari sau foarte mici cu un grad de precizie ridicat. La baza acestui mod de reprezentare se situează reprezentarea numerelor reale cu ajutorul mantisei (indică ordinul de mărime al numărului printr-o putere a bazei) și exponentului (mărima numărului în cadrul ordinului respective):

$$\pm \text{mantisă} \cdot E_{\pm \text{exponent}}$$

Mantisa în reprezentarea normalizată este un număr subunitar, *E* reprezintă valoarea bazei 10, iar *exponentul* este un număr întreg care stabilește valoarea numărului real. De exemplu: numărul real -102.224 scris în virgulă fixă va fi egal cu $0.102224 \cdot E_3$ în virgulă mobilă.

Reprezentarea caracterelor (valorilor alfanumerice)

În cazul reprezentării caracterelor, există un standard internațional care definește numerele, reprezentabile pe un octet, corespunzătoare fiecărui caracter în parte, numit standardul *ASCII*[3]. În acest mod sunt grupate un număr de 256 de caractere, fiecărui caracter fiindu-i asociat câte un număr natural din intervalul [0,255], număr care poartă denumirea de cod ASCII.

Astfel:

[1] Institute of Electric and Electronic Engineering

[2] IEEE Floating Point Standard — standardul IEEE de reprezentare a numerelor în virgulă mobilă este un set de convenții de reprezentare și de operare a numerelor reale în memoria calculatorului și care are câteva obiective:

- Creșterea preciziei operațiilor cu numere reale.
- Standardizarea operațiilor cu numere reale pe toate platformele de calcul.
- Stabilirea felului în care sunt tratate erorile.