ARHITECTURA SISTEMELOR DE CALCUL ŞI SISTEME DE OPERARE

LUCRĂRILE DE LABORATOR Nr. 6, 7 şi 8

REPREZENTAREA INFORMAŢIILOR NUMERICE ÎNTREGI ŞI REALE.

I. SCOPUL LUCRĂRILOR

Lucrările prezintă reprezentarea informaţiilor numerice întregi şi reale. Scopul lucrării constă în a familiariza studenţii cu modalităţile de implementare a datelor numerice întregi (codul direct, codul invers şi codul complementar) precum şi a informaţiilor numerice reale (reprezentarea în virgulă fixă şi virgulă mobilă).

II. NOŢIUNI TEORETICE

1 Reprezentarea informaţiei

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

Codificarea unei informaţii constă în a stabili o corespondenţă între reprezentarea externă a informaţiei (caracterul A sau numărul 33, de exemplu) şi reprezentarea sa internă, care este o secvenţă de biţi.

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

Informaţiile prelucrate în sistemele de calcul sunt de două tipuri: **instrucţiuni** şi **date.**

Instrucţiunile, scrise în limbaj maşină, reprezintă operaţiile efectuate în sistemul de calcul şi ele sunt compuse din mai multe câmpuri:

- **codul** operaţiei de efectuat;
- **operanzii** implicaţi în operaţie.

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

Datele sunt operanzii asupra cărora acţionează operaţiile (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 operaţii aritmetice, sau date nenumerice, de exemplu simbolurile care constituie un text.

2. Datele numerice

Datele numerice sunt de următoarele tipuri:

a) **numere întregi pozitive sau nule**: 0; 1; 315...
b) **numere intregi negative**: -1; -155...
c) **numere fracţionare**: 3.1415; -0.5...
d) **numere în notaţie ştiinţifică**: 4.9×10^7 ; 10^{23} ...

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

Tabele de adunare şi înmulţire binară

0 + 0 = 0	0 × 0 = 0
0 + 1 = 1	0 × 1 = 0
1 + 0 = 1	1 × 0 = 0
1 + 1 = 10	1 × 1 = 1

3. Numere întregi negative

Numerele întregi negative pot fi codificate prin trei metode:

- **semn şi valoare absolută (SVA)**;
- **complement logic sau restrâns sau faţă de 1 (C1)**;
- **complement aritmetic sau adevărat sau faţă de 2 (C2)**;

Prin metoda **"semn şi valoare absolută"**, numerele se codifică sub forma*: ± valoare absolută*.

Prin această reprezentare se sacrifică un bit pentru semn. În mod normal, 0 este codul semnului +, iar 1 este codul semnului -. În aceste condiţii, pe un cuvânt de k biţi se pot reprezenta numere întregi pozitive şi negative N, astfel încât: $-(2^{k-1}-1) \le N \le (2^{k-1}-1)$.

Această metodă de reprezentare prezintă unele inconveniente:

- numărul zero are două reprezentări distincte: 000...0 şi 100...0, adică +0 şi -0;
- tabelele de adunare şi înmulţire sunt complicate din cauza bitului de semn care trebuie tratat separat.

Complement logic şi aritmetic

Complementul logic (complement faţă de 1) se calculează înlocuind, pentru valorile negative, fiecare bit 0 cu 1 şi 1 cu 0.

Complementul aritmetic (complement faţă de 2) este obţinut adunând o unitate la valoarea complementului logic.

Exemplu: Reprezentarea numărului (-6) pe 4 biţi: + 6 = 0110

Semn şi valoare absolută: - 6 = 1110

Complement faţă de 1: - 6 = 1001

Complement faţă de 2: - 6 = 1010

Se poate uşor constata că intervalul numerelor întregi N care se pot reprezenta în complement faţă de 1 este acelaşi ca şi pentru reprezentarea **"**semn şi valoare absolută**"**.

Pentru reprezentarea în complement faţă de 2 există o valoare în plus, deci pentru k biţi vom avea: $-2^{k-1} \le N \le (2^{k-1}-1)$.

Se poate remarca faptul că bitul cel mai din stânga (bitul de semn) este întotdeauna 0 pentru numere pozitive şi 1 pentru cele negative şi aceasta pentru fiecare din cele trei reprezentări.

Tabelă de reprezentare, pe 16 biţi, a numerelor întregi cu semn
(16 biţi $\Rightarrow 2^{16} = 65536 = 2 \times 32768$ valori posibile)

zecimal	semn şi valoare absolută	complement faţă de 2	complement faţă de 1
+32767	0111...1...1111	0111...1...1111	0111...1...1111
+32766	0111...1...1110	0111...1...1110	0111...1...1110
...	...	...	...
+1	0000...0...0001	0000...0...0001	0000...0...0001
+0	0000...0...0000	0000...0...0000	0000...0...0000
-0	1000...0...0000	----------	1111...1...1111
-1	1000...0...0001	1111...1...1111	1111...1...1111
...	...	...	...
-32766	1111...1...1110	1000...0...0010	1000...0...0001
-32767	1111...1...1111	1000...0...0001	1000...0...0000
-32768	----------	1000...0...0000	----------

Reprezentarea în complement faţă de 1 recunoaşte două zerouri (+0 şi -0), dar este simetrică, deoarece aceleaşi numere pozitive şi negative sunt reprezentabile, iar această situaţie se poate uşor realiza electronic.

În complement faţă de 1 sau faţă de 2, operaţiile aritmetice sunt avantajoase, deoarece operaţia de scădere se realizează prin adunarea complementului.

Într-o adunare în complement faţă de 1, o cifră de transport către ordinul superior generată de bitul de semn trebuie adăugată la rezultatul obţinut, spre deosebire de complementul faţă de 2, când această cifră de transport se ignoră.

În complement faţă de 1 sau 2 nu se produce depăşire de capacitate decât în cazul în care cifrele de transport generate de bitul de semn şi de bitul anterior acestuia sunt diferite.

3. Numere fracţionare

Numerele fracţionare sunt numerele subunitare.

Schimbări de bază

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

Conversia se face adunând puterile (negative) 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 înmulţiri succesive cu 2 a numerelor pur fracţionare. Acest algoritm trebuie să se termine când se obţine o parte fracţionară nulă sau când numărul de biţi obţinuţi corespunde mărimii registrului sau a cuvântului de memorie în care se va stoca valoarea. Numărul binar se obţine citind părţile întregi în ordinea calculării lor.

Exemplu: 0.125 $\times$ 2 = 0.250 = 0 + 0.250
0.25 $\times$ 2 = 0.50 = 0 + 0.50
0.50 $\times$ 2 = 1.0 = 1 + 0.0

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

Pentru numerele fracţionare se pot remarca reprezentările în virgulă fixă şi virgulă mobilă.

4. Virgula fixă (VF)

Sistemele de calcul nu posedă virgula la nivelul maşinii, deci reprezentarea numerelor fracţionare se face ca şi când acestea ar fi întregi, cu o virgulă virtuală a cărei poziţie este controlată de către programator.

Datorită dificultăţii de gestionare a virgulei de către programator (pot apare frecvent situaţii de depăşire a capacităţii de memorare), se preferă soluţia aritmeticii în virgulă mobilă.

5. Virgula mobilă (VM)

Primele sisteme de calcul utilizau doar virgula fixă pentru efectuarea operaţiilor aritmetice, iar către sfârşitul anilor '50, în urma apariţiei logicii cablate s-a introdus pe scară largă reprezentarea în virgulă mobilă a numerelor fracţionare.

În majoritatea sistemelor de calcul actuale destinate în special aplicaţiilor de natură tehnico-ştiinţifică, cele două metode de reprezentare (virgula fixă şi virgula mobilă) coexistă şi sunt foarte utile.

Reprezentarea în virgulă mobilă constă în a reprezenta numerele sub forma următoare:

$N = M \times B^E$ cu: B = baza (2, 8, 10, 16...)
M = mantisa
E = exponentul

Exponentul este un număr întreg, mantisa **normalizată** este un număr pur fracţionar (fără cifre semnificative la partea întreagă). Cu excepţia numărului zero (în general reprezentat prin cuvântul 000...0), vom avea întotdeauna: $0.1_2 \leq |M| < 1_2$, sau $0.5_{10} \leq |M| < 1_{10}$.

Exponentul şi mantisa trebuie să poată reprezenta atât numere pozitive cât şi negative. Cel mai adesea, mantisa admite o reprezentare sub forma "semn şi valoare absolută", iar exponentul este fără semn, dar **decalat.**

Exemplu

SM	ED	M

unde SM este semnul mantisei, ED este exponentul decalat şi M mantisa.

Pentru nu număr de k biţi rezervaţi pentru ED se pot reprezenta fără semn 2^k valori, de la 0 la $2^k - 1$. Decalajul considerat este 2^{k-1}, ceea ce permite ca valorile de la 0 la $2^{k-1}-1$ pentru ED să corespundă unui exponent real (ER) negativ, iar valorile de la 2^{k-1} la $2^k - 1$ ale lui ED să corespundă unui exponent real (ER) pozitiv. Deci domeniul de valori reprezentabile pentru exponentul real este de la -2^{k-1} la $2^{k-1}-1$.

De exemplu, pentru k = 4, pe cei 4 biţi putem reprezenta fără semn numere de la 0 la 15 pentru ED. Decalajul considerat este $2^{k-1} = 2^3 = 8$, deci pentru exponentul real (ER) putem considera valori de la $-2^{k-1} = -2^3 = -8$ şi până la $2^{k-1} - 1 = 2^3 - 1 = 7$.

Relaţia existentă se poate scrie astfel: ER = ED - D.

Exponentul determină intervalul de numere reprezentabile în sistemul de calcul, iar numerele prea mari pentru a putea fi reprezentate corespund unei **"**depăşiri superioare**"** de

capacitate de memorare [overflow], iar numerele prea mici corespund unei **"**depăşiri inferioare**"** de capacitate de memorare [underflow].

Mărimea mantisei exprimă precizia de reprezentare a numerelor.

Avantajul utilizării virgulei mobile faţă de virgula fixă constă în intervalul mult mai extins al valorilor posibile de reprezentat.

Standardul IEEE 754

Standardul IEEE [Institute of Electrical and Electronics Engineers] defineşte trei formate de reprezentare a numerelor în virgulă mobilă:

a) **simplă precizie** pe 32 de biţi (1 bit pentru SM, 8 biţi pentru ED şi 23 pentru M);
b) **dublă precizie** pe 64 biţi (1 bit pentru SM, 11 biţi pentru ED şi 52 biţi pentru M);
c) **dublă precizie extinsă** pe 96 biţi (1 bit pentru SM, 15 biţi pentru ED şi 80 biţi pentru M) .
d) **precizie cvadruplă** pe 128 biţi (1 bit pentru SM, 15 biţi pentru ED şi 112 biţi pentru M)

Exponentul este decalat cu 128 pentru reprezentarea în simplă precizie şi cu 1024 pentru reprezentarea în dublă precizie. Mantisa fiind normalizată, există siguranţa că primul bit al mantisei are valoarea 1, ceea ce permite omiterea sa (**bit ascuns**) pentru creşterea preciziei de reprezentare, dar complică prelucrarea informaţiei.

III. MODUL DE LUCRU

Se realizează implementarea algoritmilor de înmulţiri succesive pentru conversia parţii fracţionare a unui număr, algoritmii de reprezentare a numerelor întregi negative (cod direct, complement faţă de 1 şi complement faţă de 2) precum şi reprezentarea numerelor în virgulă fixă şi mobilă.

IV. CONŢINUTUL REFERATULUI

1. Sumarul noţiunilor întâlnite.
2. Rezolvaţi următoarele probleme:
   2.1 Daţi pe 8 biţi, reprezentările în cod direct, cod invers şi cod complementar a valorilor întregi următoare: -32 şi -128.
   2.2 Daţi valoarea zecimală cu semn a numărului $B7_{16}$ codificat în cod complementar pe 8 biţi.
   2.3 Conform diferitelor metode de reprezentare a întregilor pe care le cunoaşteţi care este valoarea zecimală corespunzătoare valorii binare conţinute într-un registru pe 8 biţi: 11111111?
   2.4 Dispunem de o maşină unde valorile numerice sunt reprezentate pe 32 de biţi în virgulă mobilă simplă precizie (VMSP). Daţi sub forma $\pm a \times 2^b$ (a si b zecimale), valoarea care corespunde celor 32 biţi următori (sub formă octală): 27632000000 şi 30364000000.
   2.5 Care este cel mai mare şi cel mai mic numărul reprezentabil în virgulă mobilă simplă precizie?
   2.6 Dispunem de o maşină unde valorile numerice sunt reprezentate pe 32 de biţi în virgulă mobilă simplă precizie (VMSP). Daţi sub forma $\pm a \times 2^b$ (a si b zecimale), valoarea care corespunde celor 32 biţi următori (sub formă zecimală): 278, -6.53125, 2.5 şi 0.00005.

2.7 Dispunem de o maşină unde valorile numerice sunt reprezentate pe 12 de biţi numerotaţi de la dreapta spre stânga 0-11, unde:
- o mantisă normalizată pe 7 biţi (0-6);
- un exponent decalat reprezentând o putere a lui 2, codificat pe 4 biţi (7-10);
- un bit pentru semnul mantisei (11).

a) Găsiţi intervalul închis al valorilor strict pozitive reprezentabile pe această maşină; marginile vor fi puse sub forma $\pm a \times 2^b$ (a si b zecimale). Simplificaţi pe cât posibil.
b) Puneţi sub forma $\pm a \times 2^b$ (a si b zecimale) următoarele numere date sub forma hexazecimală: X = AE8, şi Y = 9D0.
c) Daţi sub formă octală, reprezentarea corespunzând numărului în zecimal: -32.625.

2.8 Fie cei 32 biţi următori scrişi sub formă octală: 37724000000. Ce reprezintă în zecimal această informaţie dacă se consideră un număr întreg în cod invers (C1), în cod complementar (C2) şi ca un număr real cu o reprezentare analoagă celei descrise în exerciţiul 2.6?

2.9 Program care realizează conversia unui număr fracţionar din baza p în baza q (se citesc de la tastatură bazele p şi q precum şi cifrele numărului în baza p)

2.10 Program care realizează conversia unui număr întreg (citit de la tastatură) în format intern în cod direct, cod invers şi cod complementar.

2.11 Program care realizează conversia unui număr în format intern în cod direct în format extern exprimat în zecimal., cod invers şi cod complementar.

2.12 Program care realizează conversia unui număr în format intern în cod invers în format extern exprimat în zecimal.

2.13 Program care realizează conversia unui număr în format intern în cod complementar în format extern exprimat în zecimal.

3. Observaţii şi concluzii personale.