Lucrarea nr. 2

Operaţii cu numere în diferite baze de numeraţie

Lucrarea îşi propune înţelegerea modului în care se efectuează operaţiile simple cu numere din diferite baze de numeraţie: adunarea, scăderea, înmulţirea şi împărţirea. De asemenea se doreşte aprofundarea modului în care sunt reprezentate numerele negative în alte baze de numeraţie.

2.1. Adunarea numerelor în diferite baze de numeraţie

În orice bază de numeraţie adunarea se face după aceleaşi reguli ca şi în zecimal, cu observaţia că cifra cea mai mare dintr-o bază “b” va fi $b-1$, adică 9 în zecimal, 7 în octal, 1 în binar şi F în hexazecimal. Dacă prin adunarea a două cifre de rang “i” se va obţine un rezultat mai mare decât $b-1$, va apare un “transport” spre cifra de rang următor $i+1$ iar pe poziţia de rang “i” va rămâne restul împărţirii rzultatului adunării cifrelor de rang $i+1$ la bază. Transportul spre cifra de rang $i+1$ va deveni o nouă unitate la suma cifrelor de rang $i+1$. În tabelul 2.1 este prezentat un exemplu de adunare a numerelor în binar, octal şi hexazecimal.

Tabelul 2.1. ***Adunarea numerelelor în diferite baze de numeraţie.***

Binar	**Octal**	**Hexazecimal**
$\dot{1}1101+$	$\dot{1}702+$	$\dot{1}C\dot{6}F+$
$\underline{11010}$	$\underline{2131}$	$\underline{1411}$
110111	4033	3080

2.2. Scăderea numerelor în diferite baze de numeraţie

Ca şi în cazul adunării, pentru scăderea numerelor în binar, octal şi hexazecimal se aplică regulile de la scăderea în zecimal: dacă nu se pot scădea două cifre de rang “i” (adică cifra descăzutului este mai mică decât a scăzătorului) se face “împrumut” o unitate din cifra de rang următor $i+1$. În cazul în care cifra din care se face “împrumutul” este 0, se face împrumutul mai departe din cifra de rang următor. În tabelul 2.2 este prezentat un exemplu de scădere a numerelor în binar, octal şi hexazecimal.

Tabelul 2.2. ***Scăderea numerelelor în diferite baze de numeraţie.***

Binar	**Octal**	**Hexazecimal**
$\dot{1}0\dot{1}01-$	$\dot{3}22-$	$A\dot{F}9-$
$\underline{110}$	$\underline{131}$	$\underline{13F}$
1111	171	$9BA$

2.3. Înmulţirea şi împărţirea numerelor în binar

Înmulţirea şi împărţirea numerelor în sistemul de numeraţie binar se efectuează ca şi în cazul numerelor din sistemul zecimal. În cazul înmulţirii în binar (ca şi în zecimal) $1 \times 1 = 1$, $0 \times 0 = 0$ iar $0 \times 1 = 0$. Împărţirea numerelor în sistemul de numeraţie binar are acelaşi rezultat ca şi în zecimal, adică un *cât* şi un *rest*. În tabelul 2.3 este prezentat câte un exemplu de înmulţire şi împărţire a numerelor în binar.

Tabelul 2.3. ***Înmulţirea şi împărţirea numerelor în binar.***

Înmulţire	Împărţire
1011×	11101|101
1101	101 |101
1011	0100
0000	000
1011	1001
1011	101
10001111	100

Notă:

Dacă în urma împărţirii a două numere în binar rezultă un rest diferit de zero şi mai mic decâ împarţitorul, pentru obţinerea părţii fracţionare se poate continua împărţirea astfel: se adauga cifra 0 la rest si virgula la cât si se continuă prin împărţirea restului la imparţitor, rezultatele fiind adăugate la cât după virgulă.

2.4. Reprezentarea binară a numerelor negative

Pentru reprezentarea numerelor negative în binar, bitul din stânga reprezentării numărului este folosit ca bit de semn.

Astfel avem bitul de semn:
0 - pentru numere pozitive (+)
1 - pentru numere negative (-)
Restul de $N - 1$ biţi sunt folosiţi pentru reprezentarea valorii.

2.4.1. Codul direct

Numerele întregi se reprezintă prin valoare absolută şi semn. În cazul codului direct, pentru numerele negative bitul de semn este 1 iar ceilalţi $n - 1$ biţi servesc pentru reprezentarea valorii absolute a numărului. De exemplu, numărul $N = -5$ se poate reprezenta pe 8 biţi astfel: $10000101_{(2)}$, unde valoarea absolută este $0000101_{(2)}$ iar primul bit este bitul de semn.

Domeniul de reprezentare în cazul codului direct va fi:

- 2^{n-1} valori pozitive de la 0 la $2^{n-1} - 1$
- 2^{n-1} valori negative de la $-(2^{n-1} - 1)$ la 0

Se poate observa că există două reprezentări ale lui zero, respectiv 00000000 şi 10000000, iar numărul maxim şi numărul minim dintr-un domeniu au aceeaşi valoare absolută, respectiv 01111111 şi 11111111.

2.4.2. Codul invers

Pentru numerele negative reprezentate în codul invers (complement faţă de 1) bitul de semn este 1 iar ceilalţi $n - 1$ biţi servesc la reprezentarea valorii absolute **negate** a numărului de reprezentat. Negarea se realizează la nivel de bit: biţii "0" devin "1" iar biţii "1" devin "0". De exemplu, numărul $N = -5$ se va reprezenta îin codul invers astfel: $11111010_{(2)}$, unde $1111010_{(2)}$ reprezintă valoarea absolută negată a numărului $0000101_{(2)}$.

Matematic, complementul faţă de 1 al unui număr negativ N care se reprezintă pe n biţi se defineşte astfel:

$$C_1(N) = 2^n - 1 - V \tag{2.1}$$

unde:
n - numărul de biţi al reprezentării;
V - valoarea absolută a numărului de reprezentat.

2.4.3. Codul complementar

Pentru reprezentarea numerelor negative în codul complementar (complement faţă de 2) se aplică următoarea regulă de complementare: se reprezintă numărul în valoare absolută, apoi se inversează bit cu bit(inclusiv bitul de semn care devine 1); la rezultatul obţinut se adună "1". Deci, complementul faţă de 2 se obţine din complementul faţă de 1 la care se adună "1". De exemplu, numărul $N = -5$ în codul complementar va avea valoarea:

$$\begin{array}{r} 00000101 \quad (inversare) \\ 11111010+ \\ \underline{\qquad\qquad 1\;\,} \\ 11111011\;\; \end{array}$$

Din punct de vedere matematic, complementul faţă de 2 al unui număr negativ N este:

$$C_2(N) = 2^n - V \tag{2.2}$$

unde:
n - numărul de biţi al reprezentării;
V - valoarea absolută a numărului de reprezentat.

În cazul codului complementar bitul din stânga rămâne tot timpul bit de semn. Avantajul reprezentării în complement faţă de 2 este că adunând un număr cu complementul său faţă de 2 rezultatul este 0 (ignorând depăşirea) ceea ce nu este valabil în cazul celorlalte reprezentări.

Codul complementar este cel mai utilizat în reprezentarea numerelor algebrice în calculator.

2.5. Exerciţii propuse

1) Să se efectueze următoarele operaţii:

a) $1101001_{(2)} + 1010111_{(2)}$
b) $1000100_{(2)} + 1001111_{(2)}$
c) $1733_{(8)} + 234_{(8)}$
d) $1022_{(8)} + 7721_{(8)}$
e) $AC97_{(16)} + 33ED_{(16)}$
f) $922A_{(16)} + 4522_{(16)}$
g) $10110_{(2)} - 1101_{(2)}$
h) $11101011_{(2)} - 11101_{(2)}$
i) $7100_{(8)} - 324_{(8)}$
j) $1021_{(8)} - 261_{(8)}$
k) $AA31_{(16)} - 2FC_{(16)}$
l) $FD124_{(16)} - AF3C_{(16)}$

2) Să se efectueze următoarele operaţii::

a) $110100110_{(2)} \times 11001_{(2)}$
b) $100101101_{(2)} \times 10011_{(2)}$
c) $111010001_{(2)} \times 1110_{(2)}$
d) $110111101_{(2)} \times 101_{(2)}$
e) $10111_{(2)} : 110_{(2)}$
f) $10101_{(2)} : 100_{(2)}$
g) $110011_{(2)} : 1101_{(2)}$
h) $100010_{(2)} : 101_{(2)}$

3) Să se reprezinte următoarele numere în sistemul binar, utilizând codul complementar:

a) -167
b) -622
c) -1125
d) -96
e) -101
f) -127
g) -23
h) -114