Laborator 2

Adunarea si scaderea in virgula fixa

Adunarea si scaderea in cod direct

Exemplu. Sa se efectueze operatia x-y=z, unde x = 11, y = -14, pe 6 biti (in continuarea exemplele vor fi de asemenea cu numere reprezentate in virgula fixa, pe 6 biti si nu se va mai preciza acest lucru).

```
[x]d = 001011
[y]d = 101110
```

Se calculeaza $opfin = x_s \oplus y_s \oplus op = 0 \oplus 1 \oplus 1 = 0$ (s-a considerat op=1, pentru scadere). Semnul rezultatului este z_s=x_s=0, iar modulul se calculeaza:

```
|x|+        01011+
|y|         01110
----   =>   -----
|z|         11001
```

deci, s-a obtinut $[z]_d$ = 011001 (z = 25).

Exemplu. Sa se efectueze operatia x+y=z, unde x = -29, y = 17.

```
[x]d = 111101
[y]d = 010001
```

Se calculeaza $opfin = x_s \oplus y_s \oplus op = 1 \oplus 0 \oplus 0 = 1$ (s-a considerat op=0, pentru adunare). Deoarece $|x|>|y|$, semnul rezultatului este z_s=x_s=1, iar modulul se calculeaza:

```
|x|-        11101-
|y|         10001
----   =>   -----
|z|         01100
```

deci, s-a obtinut $[z]_c$ = 101100 (z = -12).

Adunarea si scaderea in cod invers

Exemplu. Sa se efectueze operatia x + y = z, unde x = 25, y = -9, in cod invers.

```
[x]i = 011001
[y]d = 101001   =>    [y]i = 110110
```

```
[x]i+           011001+
[y]i            110110
------          ------
[z]i         1<-001111+
                     1
                ------
                010000
```

Deci, $[z]_i$ = 010000 (z = 16).

Exemplu. Se efectueaza operatia $x + y = z$, unde $x = 5$, $y = -29$, in cod invers.

```
[x]i = 000101
[y]d = 111101    =>     [y]i = 100010

[x]i+           000101+
[y]i            100010
------          ------
[z]i            100111
```

Deci, $[z]_i$ = 100111, $[z]_d$ = 111000 (z = -24).

Exemplu. Se efectueaza operatia $x + y = z$, unde $x = -11$, $y = -19$, in cod invers.

```
[x]d = 101011    =>     [x]i = 110100
[y]d = 110011    =>     [y]i = 101100

[x]i+           110100+
[y]i            101100
------          ------
[z]i         1<-100000+
                     1
                ------
                100001
```

Deci, $[z]_i$ = 100001, $[z]_d$ = 111110 (z = -30).

Adunarea si scaderea in cod complementar

Exemplu. Se efectueaza operatia $x + y = z$, unde $x = 22$, $y = -18$, in cod complementar.

```
[x]c = 010110
[y]d = 110010    =>     [y]c = 101110

[x]c+           010110+
[y]c            101110
------          ------
[z]c         1<-000100+
```

Deci, $[z]_c = 000100$ ($z = 4$).

Exemplu. Se efectueaza operatia $x + y = z$, unde $x = 20$, $y = -28$, in cod complementar.

$[x]_c = 010100$
$[y]_d = 111100 \quad \Rightarrow \quad [y]_c = 100100$

$$\begin{array}{ll} [x]_c+ & 010100+ \\ [y]_c & 100100 \\ \hline [z]_c & 111000 \end{array}$$

Deci, $[z]_c = 111000$, $[z]_d = 101000$ ($z = -8$).

Exemplu. Se efectueaza operatia $x + y = z$, unde $x = -4$, $y = -13$, in cod complementar.

$[x]_d = 100100 \quad \Rightarrow \quad [x]_c = 111100$
$[y]_d = 101101 \quad \Rightarrow \quad [y]_c = 110011$

$$\begin{array}{lr} [x]_c+ & 111100+ \\ [y]_c & 110011 \\ \hline [z]_c & 1\text{<-}101111 \end{array}$$

Deci, $[z]_c = 101111$, $[z]_d = 110001$ ($z = -17$).

Inmultirea in virgula fixa

Inmultirea in cod direct

Exemplu. Sa se efctueze inmultirea $x \bullet y = z$ in virgula fixa, cod direct, unde $x = 20/32$ si $y = -19/32$.

$[x]_d = 0.10100$
$[y]_d = 1.10011$

Se parcurg etapele:

1) Semnul rezultatului:

$$z_s = x_s \oplus y_s = 0 \oplus 1 = 1$$

2) Modulul rezultatului:

$|z| = |x| \bullet |y|$

```
        .10100•
        .10011
    ─────────────
    .0000010100+   |x|•y₋₅•2⁻⁵=|x|•2⁻⁵
    .0000101000    |x|•y₋₄•2⁻⁴=|x|•2⁻⁴
    .0000000000    |x|•y₋₃•2⁻³=0
    .0000000000    |x|•y₋₂•2⁻²=0
    .0101000000    |x|•y₋₁•2⁻¹=|x|•2⁻¹
    ─────────────
    .0101111100
```

Acesta este rezultatul exact si tinand cont si de bitul de semn rezulta $[z]_d$ = 1.0101111100. Valoarea in zecimal se obtine cu relatia:

$z = -(0\bullet 2^{-1}+1\bullet 2^{-2}+0\bullet 2^{-3}+1\bullet 2^{-4}+1\bullet 2^{-5}+1\bullet 2^{-6}+1\bullet 2^{-7}+1\bullet 2^{-8}+0\bullet 2^{-9}+0\bullet 2^{-10}) =$
$= -(0\bullet 2^{9}+1\bullet 2^{8}+0\bullet 2^{7}+1\bullet 2^{6}+1\bullet 2^{5}+1\bullet 2^{4}+1\bullet 2^{3}+1\bullet 2^{2}+0\bullet 2^{1}+0\bullet 2^{0})/2^{10} =$
$= -(256+64+32+16+8+4)/1024 = -380/1024$
(rezultat corect !)

3) Trunchierea si rotunjirea modulului rezultatului:

```
.01011 11100
     ↓
.01011+
     1
────────
.01100
```

Rezultatul aproximativ este $[z]_d$ = 1.01100, deci z = -12/32 (s-a obtinut rezultatul -0.375, fata de cel exact 0.37109375).

Inmultirea in cod invers

Exemplu. Sa se efectueze inmultirea $x \bullet y = z$, unde $x = -25/32$, $y = 13/32$.

Se reprezinta cei doi operanzi in cod invers:

$[x]_d = 1.11001$
$[x]_i = 1.00110$

$[y]_i = 0.01101$

```
                 1.00110•
                 0.01101
            ─────────────
            1.1111100110+   x•y-5•2^-5=x•2^-5
            0.0000000000    x•y-4•2^-4=0
            1.1110011011    x•y-3•2^-3=x•2^-3
            1.1100110111    x•y-2•2^-2=x•2^-2
            0.0000000000    x•y-1•2^-1=0
            ─────────────
      1 <--- 1.1010111000+
      1 --------------> 1
                        1
            ─────────────
            1.1010111010 = [z]i
```

S-a obtinut $[z]_d$ = 1.0101000101 (z = -325/1024).

Inmultirea in cod complementar

Exemplu. Sa se efectueze inmultirea $x \bullet y = z$, unde x = -12/32, y = 21/32.

Se reprezinta cei doi operanzi in cod complementar:

```
[x]d = 1.01100
[x]c = 1.10100

[y]c = 0.10101
```

```
                 1.10100•
                 0.10101
            ─────────────
            1.1111110100+   x•y-5•2^-5=x•2^-5
            0.0000000000    x•y-4•2^-4=0
            1.1111010000    x•y-3•2^-3=x•2^-3
            0.0000000000    x•y-2•2^-2=0
            1.1101000000    x•y-1•2^-1=x•2^-1
            ─────────────
      1 <--- 1.1100000100 = [z]c
      1
```

S-a obtinut $[z]_d$ = 1.0011111100 (z = -252/1024).

Tema

Sa se scrie un program pentru adunarea/inmultirea numerelor intregi in baza 2/10/16 pe lungime mare de cifre.