

# Računske operacije

## Sistemi 1 - teden 4

**Naloga 1.** Definirajte dve 16-bitni spremenljivki  $x$  in  $y$  ter ju inicializirajte s poljubnima vrednostima. Napišite program, ki izračuna  $x = x * y$ .

**Rešitev:**

```
JMP main

x: DW 35
y: DW 7

main:
    MOV A, [x]
    MUL [y]
    MOV [x], A
    HLT
```

**Naloga 2.** Naj bodo spremenljivke  $x$ ,  $y$  in  $z$  16-bitnega celoštevilskega tipa. Napišite program, ki izračuna  $z = x - \frac{y}{4}$ . Kaj lahko povemo o dobljenem rezultatu?

**Rešitev:**

```
JMP main

x: DW 7
y: DW 35
z: DW 0

main:
    MOV A, [x]
    MOV B, [y]
    SHR B, 2
    SUB A, B
    MOV [z], A
    HLT
```

Ker so spremenljivke celoštevilskega tipa, pri deljenju pride do izgube natančnosti. Rezultat je zaokrožen na celo število.

**Naloga 3.** Rešite prejšnjo nalogo tako, da ne pride do izgube natančnosti.

**Rešitev:** Spremenljivke  $x$ ,  $y$  in  $z$  definiramo kot realna števila, pri tem pa uporabimo zapis s fiksno vejico, kjer zadnja dva bita predstavljata neceli del števila. Premik dvojiške vejice v levo pomeni deljenje z 2, pomik v desno pa množenje z 2. V format s fiksno vejico torej pretvarjamo tako, da število pomnožimo z  $2^n$ , iz formata s fiksno vejico pa z deljenjem z  $2^n$ . Pri tem  $n$  pomeni število bitov za vejico. Računanje nato poteka enako kot s celimi števili.

```
JMP main

x: DW 28      ; x = 7 (7 * 4 = 28)
y: DW 140     ; y = 35 (35 * 4 = 140)
z: DW 0       ; The result must be interpreted as z / 4.

main:
    MOV A, [x]
    MOV B, [y]
    SHR B, 2
    SUB A, B
    MOV [z], A
    HLT
```

**Naloga 4.** Napišite program, ki izračuna  $x = \text{abs}(x)$ .

**Rešitev:**

Za implementacijo pogojne negacije bitov (eniškega komplementa) uporabimo logično operacijo XOR.

$x$	$y$	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Maskiranje z XOR obdrži bite, kjer je maskirni bit enak 0 in negira, kjer je maskirni bit enak 1.

Zgled:

$x$	1010 1001 1010 1010
maska	1111 0110 0000 1001
XOR	0101 1111 1010 0011

V našem primeru uporabimo masko, kjer so vsi biti enaki, torej masko 0x0000 ali 0xFFFF. Ta je odvisna od predznaka števila  $x$ . Masko nastavimo s preprostim trikom, tako da odštejemo  $0 - B$ , pri čemer register B hrani predznak števila  $x$  ( $B = 0$ , če  $x \geq 0$ ,  $B = 1$  sicer).

```
JMP main

x: DW 0xF63C ; x = -2500;

main:
    MOV A, [x] ; Get the value of x.
```

```

MOV B, A      ; Duplicate x.
SHR B, 15     ; Extract the sign bit (B = 1 if x is negative).
MOV C, 0      ; Make C = 000...0 if B = 0
SUB C, B      ; or C = 111...1 if B = 1
XOR A, C      ; One's complement if x is negative.
ADD A, B      ; Two's complement if x is negative.
MOV [x], A    ; Store the result back to x.
HLT           ; The result should be x = 0x09C4.

```

### **Domača naloga:**

Napišite program, ki izračuna  $y = -x^2$ .