

MIKROPROCESOROVÁ TECHNIKA

Třída: A3	Úloha č. 3	Název: Součet dvou vícebytových čísel	
Jméno: Jakub Tenk	Datum zadání: 22. 10. 2020	Datum odevzdání: 8.11.2020	Známka:

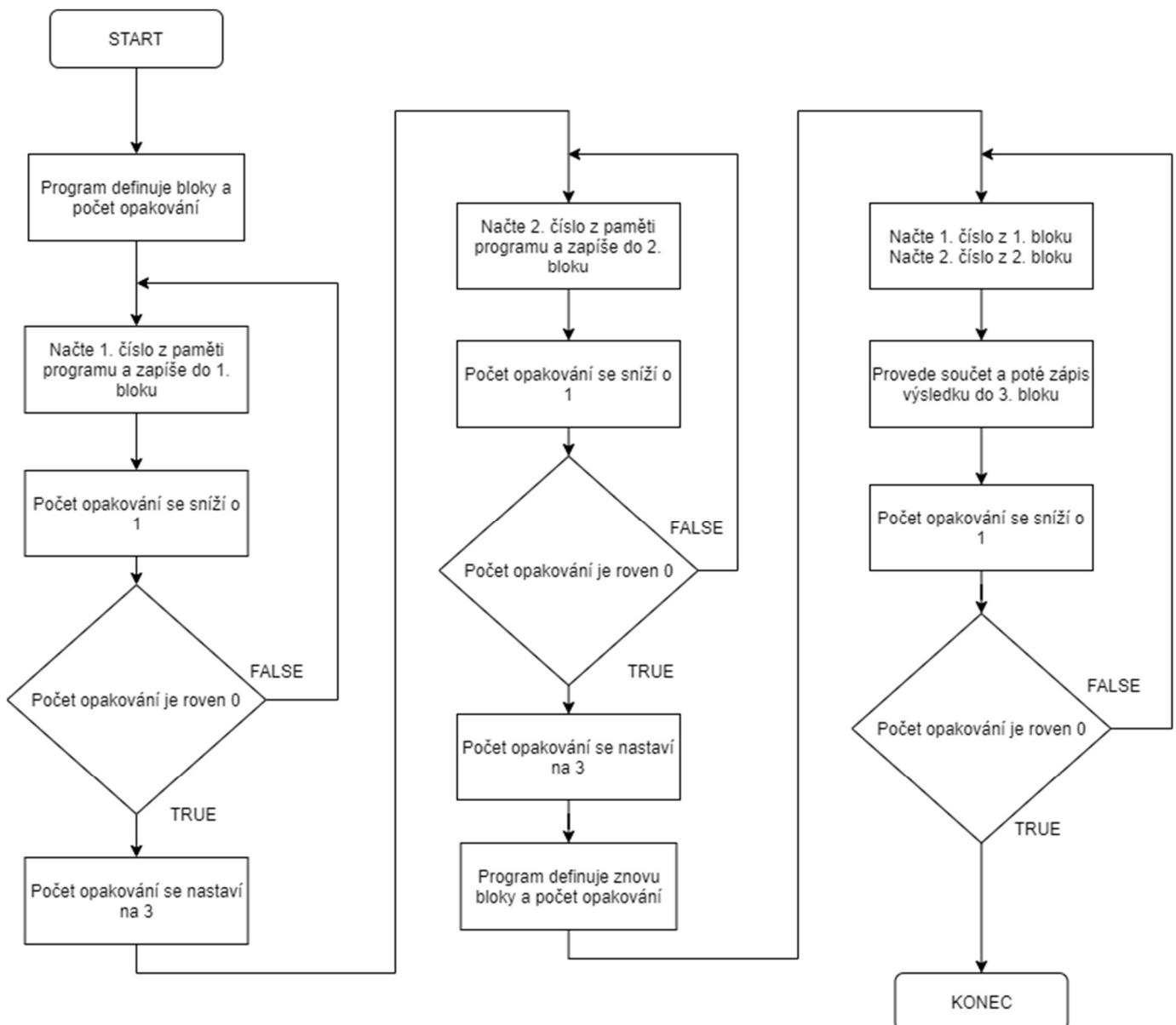
Zadání:

Vytvořte program v JSA, který bude sčítat 2 vícebytová čísla zapsaná v tabulce (využití pseudoinstrukce DB). Výsledek bude uložen na vámi určené adrese v paměti dat.

Postup:

1. Program definuje bloky
2. V cyklu skok1 načte první číslo a запиše jej do prvního bloku a sníží počet opakování o 1 (tento cyklus opakuje třikrát).
3. Počet opakování se nastaví na 3.
4. V cyklu skok2 načte druhé číslo a запиše jej do druhého bloku a sníží počet opakování o 1 (tento cyklus opakuje třikrát).
5. Počet opakování se nastaví na 3.
6. V cyklu soucet načte první číslo, druhé číslo z bloků a sečte je i s carry. Zapiše do adresy paměti dat, která začíná od 120. Počet opakování sníží o 1 (tento cyklus opakuje třikrát).
7. Program se ukončí

Vývojový diagram:



Ukázka z kódu:

```
.nolist ;neprovádí výpis na obrazovku (debug)
.include "m128def.inc"
.list ;po include začne znovu výpis na obrazovku (debug)

.def poc_opak = R22

.cseg

    ldi XL, 0x00 ; >> Definování 1. bloku pro ukládání prvních čísel
    ldi XH, 0x01 ; >>
    ldi YL, 0x00 ; >> Definování 2. bloku pro ukládání druhých čísel
    ldi YH, 0x02 ; >>
    ldi poc_opak, 3 ; >> Nastavení hodnoty počtu opakování (3x)

    ldi ZL, LOW(cislo1<<1) ; >> odkázání ukazatele Z do programové paměti
    ldi ZH, HIGH(cislo1<<1) ; >> na místo (adresa) kde se nachází cislo1

skok1: lpm R16, Z+

    st X+,R16 ; >> Zápis 1. čísla do 1. bloku
    dec poc_opak
    brne skok1

    ; >> Resetování hodnoty počtu opakování, nastavení na 3
    clr poc_opak
    ldi poc_opak,3
    ; >>

    ldi ZL, LOW(cislo2<<1) ; >> odkázání ukazatele Z do programové paměti
    ldi ZH, HIGH(cislo2<<1) ; >> na místo (adresa) kde se nachází cislo2

skok2: lpm R17, Z+

    st Y+,R17 ; >> Zápis 2. čísla do 2. bloku

    dec poc_opak
    brne skok2

    ; >> Resetování hodnoty počtu opakování, nastavení na 3
    clr poc_opak
    ldi poc_opak,3
    ; >>

    ldi XL, 0x00 ; >> Znovu definování 1. bloku
    ldi XH, 0x01 ; >>
    ldi YL, 0x00 ; >> Znovu definování 2. bloku
    ldi YH, 0x02 ; >>
    ldi ZL, 0x20 ; >> Definování 3. bloku kam budeme ukládat výpočty
    ldi ZH, 0x01 ; >>

soucet:
    ld R16, X+ ; >> načtení prvních čísel hodnot
    ld R17, Y+ ; >> načtení druhých čísel hodnot

    adc R17, R16 ; >> R17 = R17+R16+C
    st Z+,R17 ; >> Uložení součtu na adresy v paměti od 120

    dec poc_opak
    brne soucet

cislo1: .db 0x1A, 0x2B, 0x3C
cislo2: .db 0x4D, 0x5E, 0x6F

konec: rjmp konec
```

Závěr:

Tento program splňuje funkčnost zadání. S programem jsem spokojený, ale podle mě by to šlo i trochu jinak a snadněji. Furt si zvykám na JSA, protože v C# nebo v Lue by to podle mě bylo jednodušší. Taky mě překvapila funkčnost pseudoinstrukce DB, z mého pohledu je to podobné jako poli v C# nebo v Lue, akorát v ostatních jazycích to lze jednodušeji procházet.