

# MIKROPROCESOROVÁ TECHNIKA

Třída: <b>A3</b>	Úloha č. 5	Název <b>Programové zpoždění</b>	
Jméno: <b>Jakub Tenk</b>	Datum zadání: <b>19. 11. 2020</b>	Datum odevzdání: <b>2. 12. 2020</b>	Známka:

## Zadání:

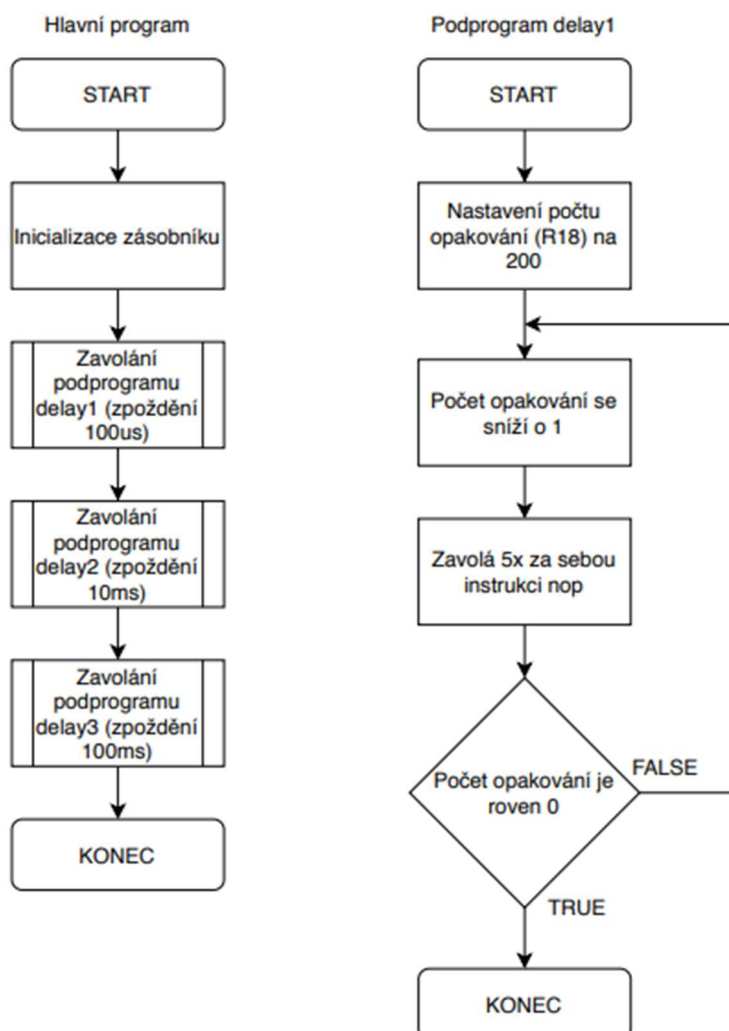
Navrhněte program pro programové zpoždění. V zadání uveďte vaše specifikované časy, které jsou realizovány v rámci podprogramu jednou, dvěma a třemi smyčkami. Časy volte ve stovkách mikrosekund, desítkách milisekund a stovkách milisekund.

1. Zpoždění – 100us
2. Zpoždění – 10ms
3. Zpoždění – 100ms

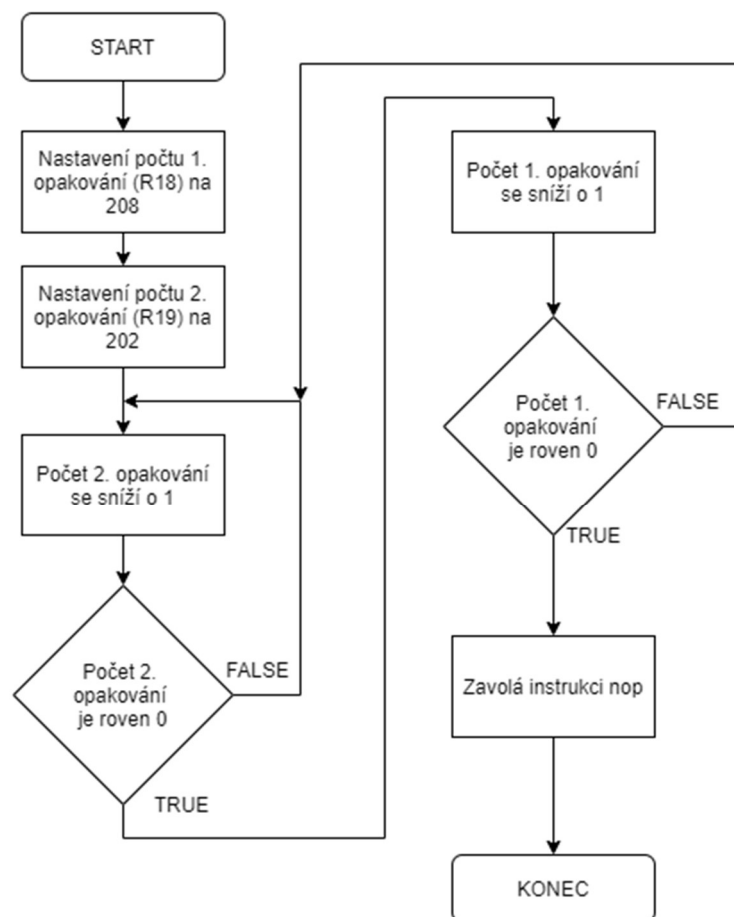
## Postup:

1. Při spuštění program inicializuje zásobník.
2. Pokračuje do hlavního programu jménem prog.
3. Zpustí podprogram delay1, který provede zpoždění o 100us a z podprogramu se vrátí do hlavního programu.
4. Pokračuje dále v hlavním programu, kde zpustí podprogram delay2, který provede zpoždění o 10ms a z podprogramu se vrátí do hlavního programu kde pokračuje dále.
5. Zpustí podprogram delay3, který provede zpoždění o 100ms a z podprogramu se vrátí do hlavního programu.
6. Program se ukončí.

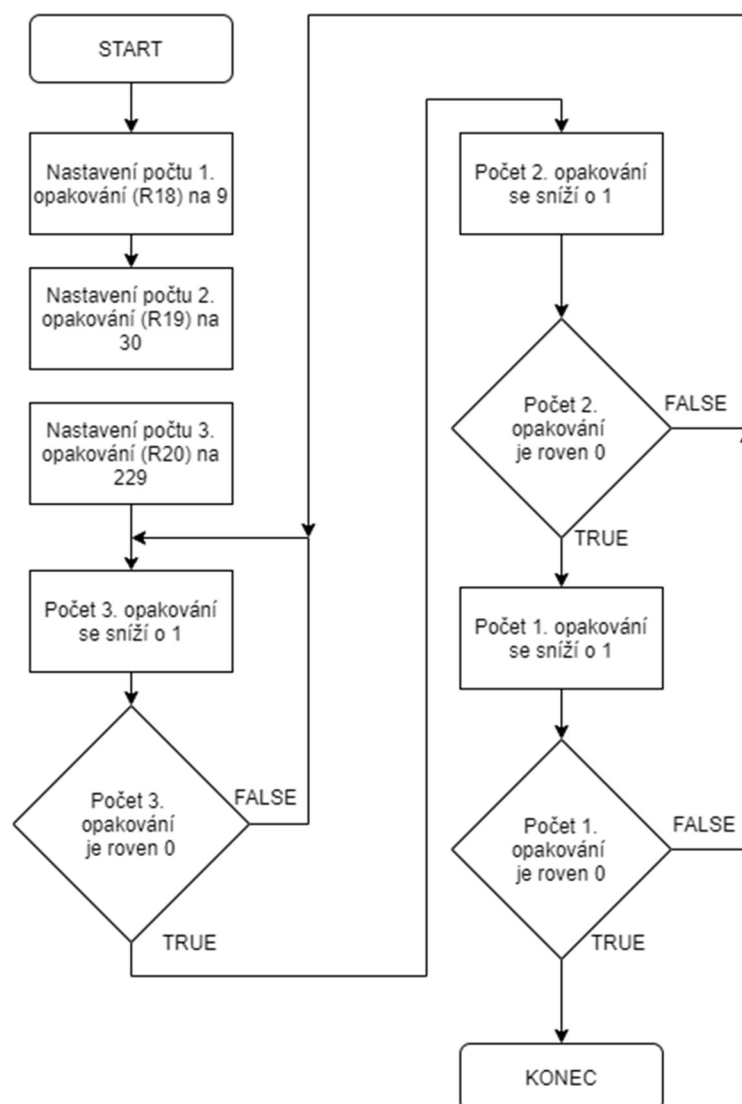
## Vývojový diagram:



Podprogram delay2



Podprogram delay3



## Ukázka z kódu:

```
; >> <<
; >> Programové zpoždění <<
; >> Vytvořil Jakub Tenk <<
; >> <<

.nolist ;vypne debug
.include "m128def.inc"
.list ;zapne debug

.def temp = R16
.dseg
.org 0x100
.cseg
.org 0x0000

    ldi temp, LOW(RAMEND) ; >> Inicializace zásobníku
    out SPL, temp
    ldi temp, HIGH(RAMEND)
    out SPH, temp

prog:
    call delay1 ; >> 1. podprogram zpoždění - 100us (provede 1600 cyklů)
    call delay2 ; >> 2. podprogram zpoždění - 10ms (provede 160000 cyklů)
    call delay3 ; >> 3. podprogram zpoždění - 100ms (provede 1600000 cyklů)
konec: rjmp konec

delay1: ldi r18, 200 ; >> Nastavení počtu opakování
d1:     dec r18 ; >> Snižuje počet opakování o 1
        nop ; >> Zavolání prázdné instrukce
        nop
        nop
        nop
        brne d1 ; >> Skok do cyklu d1
        ret ; >> Návrat z podprogramu

delay2: ldi r18, 208 ; >> Nastavení počtu opakování
        ldi r19, 202 ; >> Nastavení počtu opakování
d2:     dec r19 ; >> Snižuje počet opakování o 1
        brne d2 ; >> Skok do cyklu d2
        dec r18 ; >> Snižuje počet opakování o 1
        brne d2 ; >> Skok do cyklu d2
        nop ; >> Zavolání prázdné instrukce
        ret ; >> Návrat z podprogramu

delay3: ldi r18, 9 ; >> Nastavení počtu opakování
        ldi r19, 30 ; >> Nastavení počtu opakování
        ldi r20, 229 ; >> Nastavení počtu opakování
d3:     dec r20 ; >> Snižuje počet opakování o 1
        brne d3 ; >> Skok do cyklu d3
        dec r19 ; >> Snižuje počet opakování o 1
        brne d3 ; >> Skok do cyklu d3
        dec r18 ; >> Snižuje počet opakování o 1
        brne d3 ; >> Skok do cyklu d3
        ret ; >> Návrat z podprogramu
```

## Závěr:

Tento program splňuje funkčnost podle zadání, program zavolá podprogram, který zpomalí celý program. Vypracování úlohy bylo podle mě lehké, jelikož mi stačilo vytvořit kostru programu a podprogramu pro zpoždění, dále jen mi stačilo použít chytrou kalkulačku na zpoždění, která mi vypsalala celý kód pro zpomalení. Kalkulačku jsem použil na 2. a 3. podprogram, 1. podprogram jsem musel vymyslet a vypočítat sám, protože jsme mohli použít jen jeden cyklus. Jediný problém, na který jsem narazil byl ten, že jsem ze začátku debugoval program při rychlosti procesoru 1 MHz a divil jsem se proč to nevychází. Naštěstí jsem si toho rychle všimnul a nastavil rychlost na 16 MHz.