LABORATORIJSKE VAJE, 2. letnik - **07 -** Mikroprocesorji in mikrokontrolerji

**Arhitektura računalnikov  
 ATMEL AVR STUDIO**

**Makro ukazi, podprogrami**

# Makro ukazi

Kot že ime namiguje, je makro zaporedje več mikroukazov, ki skupaj tvorijo zaključeno celoto.

Makro se definira z direktivo .macro:

; loads 16bit constant into register X

.macro ldx

ldi xl, low(@0)

ldi xh, high(@0)

.endmacro

Primer uporabe:

ldix 0xaa55 ; v register X naložimo vrednost 0xAA55

@0, @1, itd. so parametri, ki nastopajo v makru. 0 je prvi parameter, 1 naslednji itd.

1. Vnesite zgornji macro in ga uporabite, kot kaže primer. Prilepite sliko, v kaj se je ta ukaz prevedel. (Windows -> Disassembly)
2. Napišite naslednje makro ukaze:

|  |  |  |
| --- | --- | --- |
| clrx | pobriše register X |  |
| clry | pobriše register X |  |
| lslx | rotira bite registra X za ena v levo |  |
| pushx | spravi register X na sklad |  |
| popx | pobere register X s sklada |  |
| cpix | primerja vrednost X z neko podano 16-bitno konstanto |  |
| iszero | testira X in nastavi zastavico Z |  |
| ldisp | naloži direkt vrednost  v register SP, ne paca po registrih | (\*) mogoče prej rešite naslednje poglavje...  (\*\*) predlagajte potencialne rešitve |

**Makro ukaze uporabljamo, kadar z njimi naredimo kodo bolj čitljivo, opravilo pa ne zahteva (bistveno) več prostora v pomnilniku, kot podprogram.**

# Podprogrami

Podprogrami so namenjeni izvajanjem opravil, ki se pogosto pojavljajo, a so predolgi, da bi jih uporabljali v obliki makro ukazov. Na njih vplivamo z vrednostmi v registrih in/ali na skladu.

1. Napišite podprogram **strlen\_p**, ki bo preštel, koliko znakov je v danem nizu v programskem pomnilniku in število vrne v registru r16. Na primeru spodnjega niza mora po klicu v r16 biti vrednost 0Dh.

hello db "Hello, world!", 0

1. Podprogram strcopy\_p2d prekopira niz znakov (do ASCII 0) iz programskega v podatkovni pomnilnik.

Prostor v podatkovnem pomnilniku rezervirate takole:

.cseg ; tu prideta program za testiranje in podprogram  
 .org 0x0000

............

.cseg

hello: db "Hello, world!", 0

.dseg

prostor: .byte 20 ; rezervirali smo 20 zlogov za svoje potrebe

1. Napišite podprogram is\_hex\_p, ki ugotovi, ali je na danem naslovu res šestnajstiško število (v ASCII obliki seveda..)

niz\_znakov: db "DEADBEEF",0 ; za tega vrne v zastavici Z ena   
niz\_znakov2: db "DeAdBeEf",0 ; za tega vrne v zastavici Z ena  
niz\_znakov2: db "20AHD",0 ; za tega vrne v zastavici Z nič (H ni heksadecimalna števka)

1. Napišite podprogram upper\_d, ki v danem nizu v podatkovnem pomnilniku male črke pretvori v velike.
2. Napišite podprogram strlen\_d, ki v r16 vrne dolžino niza v podatkovnem pomnilniku.
3. Napišite podprogram setbits\_r, ki v r16 vrne število enic v registru r16.   
   Za vrednost 0xFC vrne 6.