

2. sklop

Naloga 1

Napišite program, ki v 8-bitni spremenljivki S1 naredi naslednje:

- pobriše (postavi na 0) bita 3 in 5
- bite 0, 4 in 6 postavi na 1

bita 2 in 7 'obrne', torej bit, ki je prej imel vrednost 1, mora biti sedaj 0 in obratno.

Navodilo: napišite dve različici programa. V prvi različici bite postavljajte in brišite z ukazoma OR in AND, v drugi pa z BSET in BCLR. Ko program napišete, ga izvajajte po korakih in preverite njegovo delovanje z različnimi števili.

```
spr1      org $2000
          fcb %10100011

          org $e000

zacet     ldaa spr
          anda #%11010111
          oraa #%01010001
          eora #%10000100
          staa spr1
          ldx #spr1
          bset 0,x,#%00001001
          bclr 0,x,#%00001001

          org $fffe
          fdb zacet
```

Naloga 2

Napišite program, ki 8-bitno spremenljivko S1 množi z 11. Rezultat naj bo v spremenljivki S1. Navodilo: uporabite ukaze za pomikanje. Množenje z 11 naredite na naslednji način:

$$S1 * 11 = S1 * (1 + 2 + 8) = S1 + 2 * S1 + 8 * S1$$

Namig: množenje s potencami števila 2 izvedite z ukazi z pomikanje...

```
          org $2000
S1        fcb 5
S2        fcb %10101110

          org $E000
start     ldaa S1
          ldab S1
          lsla
          adda S1
          addb S1
          lslb
          lslb
```

```
stab S1
```

```
adda S1  
stab S1
```

```
nop  
org $fffe  
fdb start
```

3. sklop

Naloga 1:

- Zapišite zaporedje ukazov za 68HC11, ki zamenja vrednosti akumulatorjev A in B. Nalogo rešite z uporabo sklada (4 ukazi). Vrh sklada nastavite na konec RAM-a za spremenljivke.
- Zapišite zaporedje ukazov za 68HC11, ki zamenja vsebini registrov X in Y preko sklada.
- Napišite zaporedje ukazov za 68HC11, ki v register D naloži **naslov** trenutno zadnjega podatka na skladu.

```
org $2000  
spr1   fcb 22  
spr2   fcb 43  
org $e000
```

```
start  ldaa spr1  
       ldab spr2  
       lds #$3fff  
       psha  
       pshb  
       pula  
       pulb  
       nop  
       org $fffe  
       fdb start
```

Naloga 2:

Na 68HC11 se izvede naslednje zaporedje ukazov:

```
LDS      #$3FFF  
LDX      #$0012  
LDD      #$ABCD  
PSHX
```

```

                JSR      PROC
                PULX
D      BRA      D
PROC   PSHA
                PULB
                PSHB
                PSHX
                INS
                PULX

```

Odgovorite:

- Na kateri naslov se shrani \$AB? Utemeljite.
- Kakšna je vrednost registra IX po koncu izvajanja (za izvedbo zadnjega ukaza PULX)? Zakaj?
- Kakšna je vrednost SP? Utemeljite.
- Na katerem naslovu je ukaz PULX v glavne programu? Kako to ugotovite s pomočjo sklada?
 - Dokončajte podprogram PROC. Pozorni bodite na to, da sklad kazalec na sklad pred ukazom RTS kaže na povratni naslov.

Naloga 3:

Napišite **podprogram** za deljenje z 8. Podprogram naj nepredznačeno število, ki je v akumulatorju A, deli z 8. Rezultat naj vrne v akumulatorju A, v akumulatorju B pa naj vrne ostanek. Namig: deljenje z 8 je pomik v desno za 3 bite. Ostanek so spodnji trije biti (bit0-2). Podprogram bo enostavnejši, če najprej izločite ostanek in šele nato pomikate število v desno. Ne pozabite na inicializacijo sklada.

```

                org $2000
spr1   fcb 33

                org $E000
start  LDS #$3FFF
                lda spr1
                JSR DELI

D      BRA      D

DELI   PSHA
                ANDA #%00000111
                TAB
                PULA
                ASRA
                ASRA
                ASRA
                RTS

                nop
                org $ffe

```

fdb start

Naloga 4*:

Napišite podprogram, ki v akumulatorju A določen bit postavi na 1. Ob vstopu v podprogram naj bo v akumulatorju B zaporedna številka bita (0-7), ki ga želimo postaviti na 1. Namig: uporabite logične ukaze in ukaze za pomikanje.

```
        ORG $FFFE
        FDB START
        ORG $2000
TMP     RMB 1
        ORG $E000
START   LDS #$3FFF
        LDAA #159
        LDAB #5 ; recimo postavimo bit 5 na 1 (159->191)
        JSR DELI
KONEC   BRA KONEC
DELI    PSHA
        LDAA #1
        TSTB
PONOVI  BEQ VRNISE
        ASLA
        DECB
        BRA PONOVI
VRNISE  STAA TMP
        PULA
        ORAA TMP
        RTS
```

5. sklop

Naloga 1

Napišite program, ki 4 bitna števila v 8-bitnem zapisu (mišljena so 8-bitna števila, ki so manjša od 16) 'pakira' tako, da zloži dve števili v eno 8-bitno. Predpostavite, da je števil, ki jih je potrebni 'pakirati' 8 in so podana v tabeli, kot v naslednjem primeru:

| | | |
|------|-----|--|
| TAB1 | FCB | \$01, \$02,\$05,\$03,\$04,\$0b,\$0f,\$08 |
| TAB2 | RMB | 4 |

Po izvajanju programa naj bodo v tabeli TAB2 naslednje vrednosti: \$12,\$53,\$4b,\$f8.

Nalogo rešite z uporabo zanke in indeksnega naslavljanja. Namig: v enem obhodu zanke obdelajte dve števili iz tabele TAB1 – zanko ponovite štirikrat.

Naloga 2

Program iz prve naloge predelajte v podprogram. Podprogram naj ima tri parametre: IX kaže na tabelo s števili, ki jih je potrebno pakirati, IY kaže na tabelo za pakirana števila, v akumulatorju A pa naj bo dolžina prve tabele (število števil, ki jih želimo pakirati). Podprogram preizkusite z dvema različno dolgima tabelama števil.

```

        org $2000
tab1    fcb $01,$02,$05,$03,$04,$0b,$0f,$08
TAB2    RMB 4
stevec  fcb 8
temp    fcb 0
        org $e000
start   ldx #tab1
        ldy #tab2
        ldab #8
        lds #$3fff

        bsr paser
        nop
paser

loop    ldaa 0,x
        lsla
        lsla
        lsla
        lsla
        adda 1,x

        staa 0,y

        inx
        inx
        iny
        decb
        decb
        cmpb #0
        bne loop
        rts

        nop
        org $fffe
```

fdb start

Naloga 3

Napišite podprogram, ki v znakovnem nizu, ki se zaključi z 0, vse začetne črke besed (prvo črko in črke, ki sledijo presledku) nadomesti z velikimi. Podprogram naj dobi v IX kazalec na znakovni niz. Delovanje preverite z naslednjim nizom:

```
NIZ   FCC      'janez novak'  
      FCB      0
```

Po izvajanju bo NIZ vseboval 'Janez Novak'. Namig: uporabite indeksno naslavljanje in zanko. Števca ponovitev zanke ne potrebujete – ko pridete do ničle (ne ASCII kode '0'), se zanka zaključi. Predpostavite lahko, da so v nizu samo črke in presledki.

```
      org      $ffe  
fdb   start  
  
      org      $2000  
NIZ   FCC      'dusan smitran '  
      FCB      0  
  
start  
      org      $e000  
      lds      #$3FFF  
      ldx      #niz  
  
      ldab     0,x          ;prva crka  
      andb     #%11011111 ;spremeni v velko  
      stab     0,x          ;shrani  
naprej  
      inx  
      ldab     0,x  
      cmpb     #' '  
      bne     preve_end  
  
      ldaa     1,x  
      cmpa     #0  
      beq     pon  
  
      anda     #%11011111  
      staa     1,x  
      inx  
      bra     naprej  
  
preve_end  
      cmpa     #0  
      bne     naprej  
  
pon   bra     pon
```

6. sklop

1. Napišite podprogram NUMCOUNT , ki prešteje koliko ASCII znakov, ki ustrezajo števkam 0 – 9, se nahaja v znakovnem nizu, ki se zaključí z 0. V registru IX naj dobi podprogram naslov niza. Podprogram naj v registru D vrne število najdenih števk. Naloga mora vsebovati tudi glavni program v katerem kličete podprogram NUMCOUNT in po klicu podprograma vrnjeno vrednost shranite v spremenljivko COUNT. Testni niz v vašem programu naj bo
NIZ FCC "11 april 2005"
FCB 0

Rezultat mora biti 6. Namig: prešteti morate ASCII kode med '0' in '9'...

```
ORG $2000
NIZ   FCC "11 april 2005"
FCB   0

COUNT  FDB $0

ORG $E000
START  LDS #$3fff
      LDX #NIZ-1
      LDY #$00
      CLRA
      JSR FOR
      NOP

FOR    INX
      TST 0,x
      BNE FOR2
      RTS
FOR2
      LDB 0,x
      CMPB #0
      BEQ KONEC
      CMPB #$30
      BLO FOR ;CE JE MANJSI
      CMPB #$39
      BHI FOR ;CE JE VECJI
      BRA DODAJ ;SKOCI V DODAJ

DODAJ  INY
      STY COUNT
      BRA ALI
      BRA FOR ;VRNI NA ZACETEK
ALI
      CMPB #0
      BEQ KONEC
      BNE FOR
KONEC  RTS
      NOP
      ORG $FFFE
      FDB START
```

2. Napišite podprogram HANGCPY, ki iz izvornega znakovnega niza, ki se zaključí z 0, v ponornega prepíše samo določene črke; ostale znake naj nadomesti s podčrtajem (_). V registru IX naj dobi podprogram naslov izvornega niza, v registru IY ponorni naslov v pomnilniku, v akumulatorju A pa ASCII kodo znaka, ki ga kopiramo. Podprogram torej deluje enako, kot bi igrali igro vislice (hangman); glej primer. Naloga mora vsebovati tudi glavni program v katerem nekajkrat pokličite podprogram HANGCPY. Testni niz v vašem programu naj bo

```
    org    $fffe
    fdb    start

    org    $2000
niz    fcc    "tataratata"
    fcb    0
count  rmb    2

    org    $e000

start  lds    #$3fff
    ldx    #niz
    ldy    #'_'
    ldab   #'a'
    bsr    strcount
    std    count
konec  bra    konec
    end

strcount clry
    ldaa   0,x
loop1  cba
    bne   loop2
    iny
loop2  inx
    ldaa   0,x
    bne   loop1
    ldaa   #'_'
    staa   0,x
    xgdy
    rts
```

7. sklop

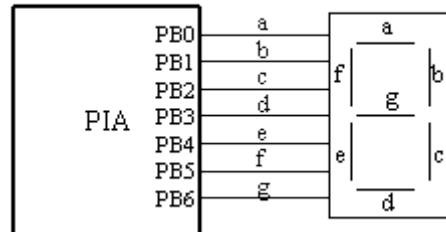
1. Napišite podprogram STRUP, ki v znakovnem nizu, ki se zaključí z 0, vse male črke nadomesti z velikimi. V registru IX naj dobi podprogram naslov niza. Naloga mora vsebovati tudi glavni program v katerem kličete podprogram STRUP. Upoštevajte, da v nizu niso samo črke, ampak tudi drugi znaki. Testni niz v vašem programu naj bo

```
org $2000
NIZ    FCC "ORA – vaje"
      FCB 0
      org $e000
start  ldx #niz
      lds #$3fff
      jsr strup
      nop
      jmp konec

strup
loop   ldaa 0,x
      cmpa #0
      beq stop1
      cmpa #$61 ;ce je manjsi
      blo skoci
      cmpa #$7A ;ce je vecji
      bhi skoci
      anda #%11011111 ;spremeni a->A
      bne skoci
skoci  staa 0,x
      cmpa #0
      inx
      beq stop1
      bne loop
stop1  rts

konec  nop
      org $fffe
      fdb start
```

2. Napišite podprogram TO7SEG, ki pretvori število 0-9 v primerno kodo za krmiljenje 7-segmentnega prikazovalnika na sliki. Podprogram naj v akumulatorju A dobi število, ki ga želimo prikazati na prikazovalniku. Rezultat naj vrne v akumulatorju B. Pri pretvorbi upoštevajte, da svetijo tisti segmenti, pri katerih je na ustreznem signalu (npr. a) logična enica. Signali ustrezajo posameznim bitom, kot je to razvidno s spodnje slike. Primer: število 0 je potrebno pretvoriti v %00111111. Namig: za pretvarjanje uporabite tabelo, v kateri so kode, ki ustrezajo številom 0-9. Uporabite indeksno naslavljanje in ukaz ABX.



```

    org    $ffe
    fdb    start
spr    fcb    2
    org    $2000
tabela fcb    %00111111
    fcb    %01010101
    fcb    %11111000
    fcb    %11001100
    fcb    %00000111
    fcb    %11111111
    fcb    %00000001
    fcb    %11111110
    fcb    %10011011
    fcb    %00011111

    org    $e000
start  lds    #$3fff
    ldaa   #7

    bsr    T07SEG ;skok na podprogram
konec  bra    konec
    end

T07SEG ldx    #tabela
    tab
    abx
    ;staa   spr
    ;ldx   spr
    cmpa   #$00
    beq    neki
    abx

neki   ldab   0,x
    rts

```

8. sklop

1) Napišite podprogram za zakasnitev $N * 6.66\text{ms}$ z imenom DELAY, ki parameter N dobi v registru D . Za čakanje naj podprogram uporablja programsko preverjanje (polling) zastavice sistema ure realnega časa RTI. Registri, ki jih potrebujete za delo z uro realnega časa:

```
PACTL    $1026
TFLG2    $1025
TMSK2    $1024
```

```
        org    $fffe
        fdb start

        org $e000
        org $2000
spr1    fcb 4
PACTL  fdb $1026
TFLG2  fdb $1025
TMSK2  fdb $1024
start  lds #$3fff
        cra
        jsr caki
d      bra d

caki   ldd spr1
        xgdy
        ldy #3
for
        dey
        ldaa PACTL
        anda #%11111100
        staa PACTL
        ldaa #%01000000
        staa TFLG2
        ldx #TFLG2
        brclr $0,x,$40,primerjaj

primerjaj cpy #$00
        bne for
        rts
konec    nop
```

2) Napišite podprogram, ki dobi parameter v akumulatorju A. Parameter je lahko '.' ali '-'. V primeru '.' naj podprogram za približno 150 ms postavi spremenljivko ZVOK na vrednost %00010000. Če sprejme '-', naj postavi spremenljivko ZVOK na vrednost %00010000 za približno 300 ms. Nato naj postavi ZVOK na vrednost 0 in počaka približno 150ms.

```

    org $ffe
    fdb start

    org $e000
    org $2000
zvok    fcb 0
PACTL   fdb $1026
TFLG2   fdb $1025
TMSK2   fdb $1024
start   lds #$3fff
        ldaa #'_'
        ;ldy #2

        jsr podprog
        bne konec
d        bra d

podprog ldy #4
        ldab #%00010000
        cmpa #'.'
        beq cakaj ;.....
        ldy #7
        ldab #%00001000
        cmpa #'_'
        beq cakaj ;_____
;crta   ldab #%00010000
;       bne cakaj
;pika   ldab #%00001000
;       bne cakaj
cakaj
        stab zvok
        dey
        ldaa PACTL
        oraa #%00000011
        staa PACTL
        ldaa #%01000000
        staa TFLG2
        ldx #TFLG2
        brclr $0,x,$40,primerjaj

primerjaj cpy #$00
         bne cakaj
         rts
konec    nop

```

3) Napišite podprogram, ki kot parameter dobi kazalec na niz, v katerem se pojavljajo samo '.' in '-', ter črka E, ki označuje konec niza. Program naj po za vsaki './-' pokliče podprogram iz točke 2, ko naleti na 'E', pa naj počaka približno 300ms in se zaključi.

```

org $2000

spr1   fcc ".-E"
n      equ 22           ;konstanta za zakasnitev èe je znak '.'
m      equ 45           ;konstanta za zakasnitev èe je znak '-'
k      equ 45           ;konstanta za zakasnitev èe je konèni znak 'E'
pactl  equ $1026        ;nastavitev intervala
tflg2  equ $1025        ;brisanje zastavic
zvok   equ %00010000

org $e000
start  lds #$3fff
        ldy #spr1       ;v akumulator a naložimo (. ali -)
        bsr podprog2    ;klic podprograma

koniec bra konec       ;konec programa

delay  ldaa pactl       ;v a naložimo kar je na naslovu pactl
        anda #%11111100 ;nastavitev intervala
        staa pactl
        xgdx           ;iz akumulatorja d damo vrednost v x

zanka  ldab #%01000000  ;brisanje zastavic
        stab tflg2
        cmpx #0         ;primerjanje èe je konstanta enaka 0
        ldy #tflg2

skok   brclr 0,y,#%01000000,skok ;èakanje na zastavico
        dex             ;zmanjšanje števca
        bne zanka      ;ponavljanje zanke dokler ni x=0
        rts

podprog ;zaèetek podprograma
pika   cmpa #'.'       ;kar je v a primerjamo z '.'
        bne crtica     ;èe ni '.' preskoèimo na crtica
        ldd #n         ;èe je '.' se d nastavi na n=22(zakasnitev)
        bsr delay      ;klic podprograma delay, ki se izvede n-krat
crtica cmpa #'-'       ;primerjamo ali je znak v a '-'
        bne koncni     ;èe znak ni '-' se program zakljuèi
        ldd #m         ;èe je '-' se d nastavi na m=45(zakasnitev)
        bsr delay      ;klic podprograma delay, ki se izvede m-krat
koncni cmpa #'E'
        bne zanka2
        ldd #k

```

```

        bsr delay
zanka2  rts

podprog2  ldaa 0,y          ;v a damo znak iz spr1
        pshy              ;vsebino y damo na sklad
        psha              ;vsebino a damo na sklad
        bsr podprog      ;klic podprog (za zakasnitve)
        pula              ;iz sklada vzamemo a
        puly              ;iz sklada vzamemo y
        iny               ;povečamo y
        cmpa #'E'         ;primerjamo èe je 'E'
        bne podprog2     ;klic podprog2 (za naslednji znak)

rts      ;vrnemo se v glavni program
nop
org $ffe
fdb start

```

Naloge s preverjanja znanja iz leta 2003/04

1. Napišite podprogram, ki sešteva negativna predznačena števila v tabeli in vrne zaporedno številko (indeks) elementa, pri katerem je vsota preseгла -100. Števila v tabeli in vsota naj bodo 8-bitna. Prvi element v tabeli ima zaporedno številko 0, naslednji 1 itd... Števila v tabeli naj bodo naključna (RMB), v tabeli naj bo 100 števil.
2. Napišite podprogram STRLEN, ki v akumulatorju A vrne dolžino niza, ki se končuje z 0. Ob vstopu v podprogram IX kaže na začetek niza. Primer:

```
NIZ      FCC 'ORA-PREVERJANJE'  
        FCB 0  
  
        ...  
        LDX #NIZ  
        BSR STRLEN  
        ; sedaj je v A 15
```

3. Napišite podprogram STRCNT, ki v akumulatorju A vrne število, ki je enako številu ponovitev določene črke v nizu, ki se končuje z 0. Ob vstopu v podprogram IX kaže na začetek niza, v akumulatorju A pa je ASCII koda črke, za katero nas zanima, kolikokrat se pojavi v nizu. Primer:

```
NIZ      FCC 'ORA-PREVERJANJE'  
        FCB 0  
  
        ...  
        LDX #NIZ  
        LDAA #'E'  
        BSR STRCNT  
        ; sedaj je v A 3
```

4. Napišite podprogram, ki sešteva pozitivna predznačena števila v tabeli in vrne zaporedno številko (indeks) elementa, pri katerem je vsota preseгла 100. Števila v tabeli in vsota naj bodo 8-bitna. Prvi element v tabeli ima zaporedno številko 0, naslednji 1 itd... Števila v tabeli naj bodo naključna (RMB), v tabeli naj bo 100 števil.