



---

## ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΪΠΟΛΟΓΙΣΤΩΝ 3η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ

---

ΦΟΙΤΗΤΕΣ

Αθανασίου Ιωάννης / Α.Μ.:03117041 / 6<sup>ο</sup> Εξάμηνο  
Καραβαγγέλης Αθανάσιος / Α.Μ.:03117022 / 6<sup>ο</sup> Εξάμηνο

**ΑΣΚΗΣΗ 1<sup>η</sup>**

Παρακάτω παρατίθεται ο κώδικας σε assembly 8085 και ακόμα επισυνάπτεται στο αρχείο μας .zip το εκτελέσιμο αρχείο σε μLab με όνομα «s3askisi1.8085» .

```

        IN 10H
START:  MVI A,0DH
        SIM
        EI                      ;RTS 6.5 routine

INF:    JMP INF

INTR_ROUTINE:
        MVI C,61H
        LXI H,0B06H             ;IN ORDER TO HAVE EMPTY SPACES
        MVI M,10H               ;AND NOT 0'S IN THE OTHER POSITIONS
        LXI H,0B07H             ;OF THE 7 SEGMENT DISPLAY
        MVI M,10H
        LXI H,0B08H
        MVI M,10H
        LXI H,0B09H
        MVI M,10H

LEDS_ON:
        MVI B,00H
        MVI A,00H
        STA 3000H               ;TURN ON FOR 1/2 SEC
        EI                      ;ENABLE INTERRUPTS

FIX_TIMER:

        DCR C                   ;C <- C - 1
        MOV A,C
        CPI 00H
        JZ INF
        MOV A,C                 ; IF SUB LEADS TO RESULT F THEN DECREASE BY 6
        ANI 0FH                 ; 9 AS A DECIMAL BEHAVES
        CPI 0FH                 ; IF WE HAVE F AS SECOND DIGIT GO TO DECIMAL_DECR
        JZ FIX_FS
        JMP PRINT

LEDS_OFF:
        MVI B,11H
        MVI A,FFH
        STA 3000H               ;TURN OFF FOR 1/2 SEC
        JMP PRINT

FIX_FS:

        MOV A,C
        SUI 06H                 ;DECR BY 6
        MOV C,A

PRINT:  INR B
        MOV A,B                 ;STORE B IN A FOR COMPARISONS
        CPI 10H
        JZ LEDS_OFF
        CPI 20H
        JZ LEDS_ON
        MOV D,C                 ;STORE C TEMPORARILY IN D
        MVI B,01H               ;FIXING THE DELAY TO BE 01F4H = 500ms IN DECIMAL =

```

1/2 SEC

```

MVI C,F4H
CALL DELB          ;DELB ROUTINE
MOV C,D            ;RESTORE C
MOV B,A            ;RESTORE B
LXI H,0B04H
MOV A,C            ;STORE C IN A
ANI 0FH            ;ISOLATE THE 4 LSB'S
MOV M,A            ;STORE A IN MEMORY (HL)
INX H
MOV A,C            ;STORE C IN A
ANI F0H            ;ISOLATE THE 4 MSB'S
RRC                ;SHIFT LSB'S
RRC
RRC
MOV M,A            ;STORE A IN MEMORY (HL)
LXI D,0B04H        ;DISPLAY THEM
CALL STDM
CALL DCD
JMP PRINT

```

END

## ΑΣΚΗΣΗ 2<sup>η</sup>

Παρακάτω παρατίθεται ο κώδικας σε assembly 8085 και ακόμα επισυνάπτεται στο αρχείο μας .zip το εκτελέσιμο αρχείο σε uLab με όνομα «s3askisi2.8085» .

```

ARXH:          IN 10H

                ; Tha xrhsimopoihsoume:
                ;K1 = 32H (50)
                ;K2 = 64H (100)
                ;K3 = C8H (200)

                MVI A,0DH          ; ENABLE RST6.5
                SIM
                EI

WAIT:           JMP WAIT          ; INF WAIT FOR INTERRUPT

INTR_ROUTINE:  MVI A,0DH
                SIM
                EI                ; ENABLE INTERRUPT AGAIN

                PUSH PSW
                PUSH B
                PUSH D

                LXI H,0A00H
                MVI M,10H          ; keno dejia

                INX H
                MVI M,10H          ; keno dejia

                INX H
                MVI M,10H          ; keno dejia

                INX H
                MVI M,10H          ; keno dejia

```

```

        INX H
        INX H
        CALL KIND
        MOV M,A                ; MSB sto pio aristero 7-segment

        RLC
        RLC
        RLC
        RLC
        MOV B,A                ; MSB sto B metatopismeno

        DCX H
        CALL KIND
        MOV M,A                ; LSB sto 2o pio aristero 7-segment

        ADD B                  ; A <- A + B = MSB(metatopismeno) + LSB

        POP D                  ; epanaferw D = K2 kai E = K3
        POP B                  ; epanaferw C = K1

        MOV B,A                ; B <- o arithmos
        POP PSW
        MOV A,B                ; A <- o arithmos

        MVI C,32H
        CMP C                  ; arithmos <= K1
        JC PERIOXH_0
        JZ PERIOXH_0

        MVI D,64H
        CMP D                  ; arithmos <= K2
        JC PERIOXH_1
        JZ PERIOXH_1

        MVI E,C8H
        CMP E                  ; arithmos <= K3
        JC PERIOXH_2
        JZ PERIOXH_2

        JMP PERIOXH_3          ; arithmos > K3

PERIOXH_0:  MVI A,FEH
             JMP ROUTINE_END
PERIOXH_1:  MVI A,FDH
             JMP ROUTINE_END
PERIOXH_2:  MVI A,FBH
             JMP ROUTINE_END
PERIOXH_3:  MVI A,F7H
             JMP ROUTINE_END

ROUTINE_END: STA 3000H          ; anabei to katallhlo bit
             LXI D,0A00H
             CALL STDM

SHOW_LOOP:
             CALL DCD          ; emfanizetai h eisodos sto 7-segment
             MVI B,01H
             JMP SHOW_LOOP

END

```

## ΑΣΚΗΣΗ 3<sup>η</sup>

**α)**

```
INR16 MACRO ADDR:
    PUSH H
    PUSH B

    LXI H,ADDR      ; (H) (L) <- ADDR
    MOV C,M          ; C <- Xlow
    INX H             ; (H) (L) <- ADDR+1
    MOV B,M          ; B <- Xhigh
    INX B             ; BC <- BC + 1
    MOV M,B          ; (ADDR+1) <- B
    DCX H             ; (H) (L) -> ADDR
    MOV M,C          ; (ADDR) <- C

    POP B
    POP H
ENDM
```

**β)**

```
FILL MACRO ADDR,K
    PUSH H
    PUSH B
    PUSH PSW

    LXI H,ADDR      ;address in registers H,L
    MVI B,K          ;C<-K
    MVI A,00H        ;A = 0

    CMP B
    JZ FILL_IF_K_ZERO ;IF K=0 THEN FILL 256 POSITIONS

FILL_MEMORY:
    MOV M,B          ;M<-C
    INX H             ;NEXT ADDRESS
    DCR B             ;C<-C-1
    CMP B
    JNZ FILL_MEMORY  ;IF C!=0 CONTINUE
    JZ END            ;ELSE END IF C=0 -> END

FILL_IF_K_ZERO:
    MVI M,K          ;M<-0
    INX H             ;HL<-HL + 1 (GO TO NEXT ADDRESS)
    MOV B,FFH        ;C=255
    JMP FILL_MEMORY  ;START FILLING STARTING FROM 255

END:
    POP PSW
    POP B
    POP H
ENDM
```

**γ)**

```
RHLR MACRO Q,R
    PUSH PSW
    MOV A,R          ;A<-R
    RAL              ;CY = a7 and CY is stored in position 0 of register R
    MOV R,A          ;R<-A
    MOV A,Q          ;A<-Q
```

```

        RAL                ;CY = b7 and a7 is stored in position 0 of register Q
        MOV Q,A            ;Q<-A
        POP PSW
    ENDM

```

## ΑΣΚΗΣΗ 4<sup>η</sup>

Όταν συμβαίνει η διακοπή η εντολή JUMP ολοκληρώνεται κανονικά αφού η εκτέλεσή της έχει ήδη ξεκινήσει. Ωστόσο, όταν η εκτέλεση της ολοκληρωθεί τότε ξεκινά να εξυπηρετείται η διακοπή. Όσο αυτή εξυπηρετείται αποθηκεύεται στην στοίβα ο PC ώστε να επιστρέψει ομαλά το πρόγραμμα στην κανονική του ροή, γεγονός που συνεπάγεται μία αλλαγή στον SP.

Άρα θα έχουμε τις εξής αλλαγές :

- **SP -> 1FEEH**, αφού η στοίβα μεγαλώνει από υψηλότερες σε χαμηλότερες διευθύνσεις και αποθηκεύουμε σε διαδοχικές θέσεις με τη σειρά το byte υψηλότερης τάξης και έπειτα το δεύτερο χαμηλότερης τάξης byte.
- **PC -> 0034H**, αφού πλέον ο PC μας θα δείχνει στη διεύθυνση όπου υπάρχει η ρουτίνα εξυπηρέτησης της διακοπής RST 6.5.

## ΑΣΚΗΣΗ 5<sup>η</sup>

```

MAIN:      MVI B,00H
           MVI C,00H      ; ston diplo register BC tha kratame thn ekastote eisodo

           MVI H,00H
           MVI L,00H      ; ston diplo register HL tha kratame to ews twra sum

           MVI D,00H      ; ston register D tha kratame metrhth tw n ews twra eisodwn

           MVI A,0EH
           SIM
           EI              ; epitrepsh diakophs 5.5

WAIT:      JMP WAIT

002C:      JMP RST5.5

RST5.5:    PUSH PSW
           INR D           ;auksanw ton metrhth tw n eisodwn
           JMP ODD_EVEN    ;elegxw an einai peritth h artia eisodos

EVEN:      ; artios arithmos eisodwn mexri kai aythn
           IN 20H          ; A <- ta 4 LSB
           ANI 0FH         ; apomonwnw ta 4 bits pou dinontai
           MOV C,A         ; C <- 0000|LSB
           MVI B,00H       ; B <- 0000|0000
           ; ara BC <- 0000|0000|0000|LSB

```

```

DAD B          ; ( H ) ( L ) <- ( H ) ( L ) + BC

MOV A,D        ; A <- eisodoi ews twra (bhmata)
CPI 20H        ; sygkrish me to 32 (synolika bhmata pou prepei na
ginoun)
JZ END_INPUT  ; eginan kai ta 32 bhmata
POP PSW
EI
JMP WAIT      ; alliws perimene gia thn epomenh eisodo

ODD:           ; perittos arithmos eisodwn mexri kai aythn
IN 20H        ; A <- ta 4 MSB
ANI 0FH      ; apomonwnw ta 4 bits pou dinontai
RLC
RLC
RLC
RLC          ; A <- MSB|0000
MOV C,A      ; C <- MSB|0000
MVI B,00H    ; B <- 0000|0000
              ; ara BC <- 0000|0000 | MSB|0000
DAD B        ; ( H ) ( L ) <- ( H ) ( L ) + BC
MOV A,D      ; den xreiazetai elegxos gia to plithos twm mexri twra
              ; vhmaw, afou tha termatisoume se artio plithos (32)

POP PSW
EI
JMP WAIT

ODD_EVEN:     PUSH PSW      ; krataw A
              PUSH B        ; krataw BC
              PUSH H        ; krataw HL

WHILE_LOOP:   MOV A,D      ; sthn while_loop vriskw an o D einai artios h
CPI 00H      ; perittos
JZ IS_EVEN
JC IS_ODD
DCR A
DCR A
JMP WHILE_LOOP

IS_EVEN:      POP H
              POP B
              POP PSW
              JMP EVEN

IS_ODD:       POP H
              POP B
              POP PSW
              JMP ODD

END_INPUT:    DI          ; den dexomaste alles eisodous
              ; vriskw meso oro, diairwntas to athroisma
              ; pou exw ston diplo HL me to 16

DAD H        ; to sum katalamvanei ta 12 LSB
DAD H        ; apo ta 16 bits, ara gia ton MO xreiazontai
DAD H        ; 4 olisthiseis pros ts aristera, me
DAD H        ; DAD H = 1 olisthisi

RST 1        ; o MO telika ston diplo register HL
END

```