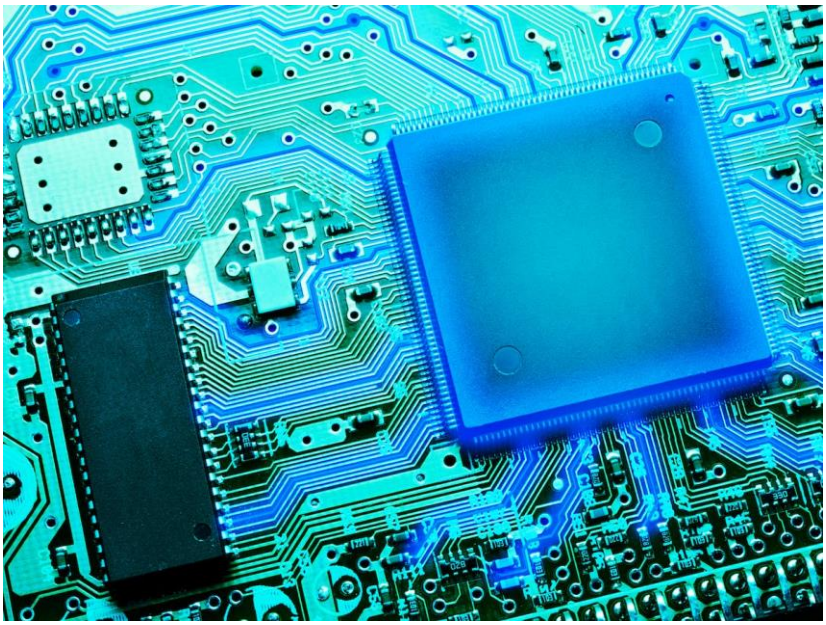
Ε.Μ.Π. - ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΑΚΑΔ. ΕΤΟΣ 2022-2023

ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ

3^{Η} ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ



Καμπουγέρης Χαράλαμπος | el20098
Κουστένης Χρίστος | el20227

Άσκηση 1

Ακολουθεί ο κώδικας για την άσκηση αυτή δοκιμασμένος στον Microlab προσομοιωτή του 8085.

```
      IN 10H                  ; Input, (A)<-data

      MVI A,10H               ;  Set up Display
      STA 0B00H               ; ((0B)(00))<-(A)
      STA 0B01H
      STA 0B02H
      STA 0B03H
      STA 0B04H
      STA 0B05H

      MVI A,0DH               ;Initialization of Interupt mask
      SIM                     ;Put  into INTERRUPT mask
      EI                      ;Enable interuptions RST 6.5
WAIT:
      JMP WAIT
INTR_ROUTINE:
      POP H             ;POP return address so that the stack doesn't fill up
      EI                ;Enable interrupts inside interrupt routine
      MVI A,00H         ;Turn on LEDs
      STA 3000H
      MVI H,06H         ;Counter for 6 iterations
      MOV A,H           ; (A)<- (H)
      DCR A             ;Set up tens
      STA 0B01H               ;Store tens in the 2nd segment display
SECONDS:
      MVI A,09H               ;Set up 9 secs (units)
LIGHTS_ON:
      STA 0B00H               ;Store units in the 1st segment display
      CALL DISPLAY
      DCR A
      CPI 00H                 ;Compare with zero
      JNZ LIGHTS_ON           ;If Z=0 then 9 seconds passed
      CALL ZERO               ;Display zero unit (1 sec)
      DCR H                   ;Decrease counter
      JZ EXIT                 ;If Z=0 end timer
      MOV A,H
      DCR A
      STA 0B01H
      JMP SECONDS       ;Repeat for 60 seconds
EXIT:
      MVI A,FFH         ;Turn off LEDs
      STA 3000H
      JMP WAIT                ;Return to wait (main program)
DISPLAY:
      LXI B,0064H             ;100 msec delay
      LXI D,0B00H             ;For STDM ((D)(E))<- 0B00H
      PUSH PSW                ; SAVES on stack the values of the registers
      PUSH H
      PUSH D
      PUSH B
      CALL STDM
      MVI A,0AH         ;10*100msec=1sec
1SEC:
      CALL DCD
      CALL DELB
      DCR A
      CPI 00H
```

```
      JNZ 1SEC
      POP B
      POP D
      POP H
      POP PSW
      RET
ZERO:                         ;Display zero in the 3rd segment display
      MVI A,00H
      STA 0B00H
      CALL DISPLAY
      CALL DELB
      RET

      END   IN 10H                  ; Input, (A)<-data

      MVI A,10H               ;  Set up Display
      STA 0B00H               ; ((0B)(00))<-(A)
      STA 0B01H
      STA 0B02H
      STA 0B03H
      STA 0B04H
      STA 0B05H

      MVI A,0DH               ;Initialization of Interupt mask
      SIM                     ;Put  into INTERRUPT mask
      EI                      ;Enable interuptions RST 6.5
WAIT:
      JMP WAIT
INTR_ROUTINE:
      POP H             ;POP return address so that the stack doesn't fill up
      EI                ;Enable interrupts inside interrupt routine
      MVI A,00H         ;Turn on LEDs
      STA 3000H
      MVI H,06H         ;Counter for 6 iterations
      MOV A,H           ; (A)<- (H)
      DCR A             ;Set up tens
      STA 0B01H               ;Store tens in the 2nd segment display
SECONDS:
      MVI A,09H               ;Set up 9 secs (units)
LIGHTS_ON:
      STA 0B00H               ;Store units in the 1st segment display
      CALL DISPLAY
      DCR A
      CPI 00H                 ;Compare with zero
      JNZ LIGHTS_ON           ;If Z=0 then 9 seconds passed
      CALL ZERO               ;Display zero unit (1 sec)
      DCR H                   ;Decrease counter
      JZ EXIT                 ;If Z=0 end timer
      MOV A,H
      DCR A
      STA 0B01H
      JMP SECONDS       ;Repeat for 60 seconds
EXIT:
      MVI A,FFH         ;Turn off LEDs
      STA 3000H
      JMP WAIT                ;Return to wait (main program)
DISPLAY:
      LXI B,0064H             ;100 msec delay
      LXI D,0B00H             ;For STDM ((D)(E))<- 0B00H
      PUSH PSW                ; SAVES on stack the values of the registers
      PUSH H
      PUSH D
```

```
      PUSH B
      CALL STDM
      MVI A,0AH          ;10*100msec=1sec
1SEC:
      CALL DCD
      CALL DELB
      DCR A
      CPI 00H
      JNZ 1SEC
      POP B
      POP D
      POP H
      POP PSW
      RET
ZERO:                          ;Display zero in the 3rd segment display
      MVI A,00H
      STA 0B00H
      CALL DISPLAY
      CALL DELB
      RET

      END
```

Άσκηση 2

```
IN 10H              ; Input data from port 10H

    MVI A,10H           ; Initialize the 7-segment display
    STA 0B00H           ; Store A at memory location 0B00H (units digit)
    STA 0B01H           ; Store A at memory location 0B01H (tens digit)
    STA 0B02H
    STA 0B03H
    STA 0B04H
    STA 0B05H

    MVI A,0DH           ; Enable RST 6.5 interrupt
    SIM
    EI

MAIN_LOOP:
    JMP MAIN_LOOP       ; Infinite loop

INTERRUPT_ROUTINE:
    CALL KIND           ; Input from keyboard -> Units digit
    STA 0B04H           ; Store the input in the units digit location
    CALL DISPLAY        ; Call the DISPLAY subroutine
    MOV B,A
    CALL KIND           ; Input from keyboard -> Tens digit
    STA 0B05H           ; Store the input in the tens digit location
    CALL DISPLAY        ; Call the DISPLAY subroutine
    MVI D,28H           ; K1 = 40
    MVI E,96H           ; K2 = 150
    RLC                 ; Rotate left
    RLC
    RLC
    RLC
    ORA B
    MOV B,A             ; The final number is stored in B
    EI
```

```
    MOV A,B
    CMP D               ; Compare with K1
    JC RANGE1
    JZ RANGE1
    CMP E               ; Compare with K2
    JC RANGE2
    JZ RANGE2
    MVI A,FBH           ; Open the 3rd LSB LED in any other case
    STA 3000H
    RET

DISPLAY:
    PUSH PSW
    PUSH H
    PUSH D
    PUSH B
    LXI D,0B00H
    CALL STDM
    CALL DCD
    POP B
    POP D
    POP H
    POP PSW
    RET

RANGE1:
    MVI A,FEH           ; Activate the 1st LSB LED
    STA 3000H
    RET

RANGE2:
    MVI A,FDH           ; Activate the 2nd LSB LED
    STA 3000H
    RET

    END
```

Άσκηση 3

(α)

```
SWAP Nible MACRO Q
      PUSH PSW
      MOV A,Q
      RRC
      RRC
      RRC
      RRC
      MOV Q,A
      MOV A,M
      RRC
      RRC
      RRC
      RRC
```

```
      MOV M,A
      POP PSW

      ENDM
```

(β)

```
FILL MACRO RP,X,K
    PUSH PSW          ; Push PSW register onto the stack
    PUSH H            ; Push H register onto the stack
    MOV H,R          ; Move the value of R register to H register
    MOV L,P          ; Move the value of P register to L register

LOOP:
    MOV M,K          ; Move the value of K to the memory location pointed by HL
    INX H            ; Increment the HL pair to point to the next memory location
    DCR X            ; Decrement the value of X by 1
    JNZ LOOP         ; Jump back to LOOP if X is not zero

    POP H             ; Pop the value from the stack to H register
    POP PSW           ; Pop the value from the stack to PSW register

    ENDM
```

(γ)

```
RHLR MACRO
      PUSH PSW
      MOV A,H  ; Copy the content of register H to register A
      RRC      ; Rotate the content of register A right
      MOV H,A  ; Copy the content of register A back to register H
      MOV A,L  ; Copy the content of register L to register A
      RRC      ; Rotate the content of register A right
      MOV L,A  ; Copy the content of register A back to register L
      POP PSW

      ENDM
```

Άσκηση 4

Τα βήματα της διαδικασίάς που περιγράφεται είναι τα εξής:

1. Η ρουτίνα εξυπηρέτησης της διακοπής **RST 5.5** εκτελείται αφού ολοκληρωθεί η εντολή **CALL 0900H.** Η εντολή αυτή αποθηκεύει στις θέσεις **(SP)-1** και **(SP)-2** τα 8 bits υψηλότερης και χαμηλότερης τάξης αντίστοιχα της διεύθυνσης της επόμενης εντολής που είναι η **0843H** και τοποθετεί στον PC την τιμή **0900H.**
2. Η ρουτίνα εξυπηρέτησης της διακοπής RST 5.5, αρχίζει από τη διεύθυνση **002CH** και αυτή η τιμή δίνεται στον PC αφού η παρούσα κατάσταση του δηλαδή η διεύθυνση **0900H** σωθεί στη στοίβα στις θέσεις **(SP)-3** και **(SP)-4**

ώστε να επανέλθει το κύριο πρόγραμμα μετά την επιστροφή **RET** της ρουτίνας της διακοπής **RST 5.5** σε αυτή τη διεύθυνση.

3. Αφού τελειώσει η ρουτίνα εξυπηρέτησης της **RST 5.5**, γίνεται **POP** από την στοίβα η προηγούμενη διεύθυνση του PC (δηλαδή η **0900H**) και ο PC παίρνει την τιμή αυτή και ο stack pointer ανεβαίνει 2 θέσεις μνήμης. Επομένως, **(SP) = (SP) + 2 = 2FFEH**.
4. Τέλος, ολοκληρώνεται η εκτέλεση της τελευταίας ρουτίνας στην διεύθυνση **0900H** και γίνεται **POP** η διεύθυνση **0843H** από την στοίβα και ο PC λαμβάνει την τιμή αυτή. Έτσι, **(SP) = (SP) + 2 = 3000H**

Συνεπώς το πρόγραμμα συνεχίζει την εκτέλεση του από τη εντολή που βρίσκεται στη διεύθυνση 0843H με τη στοίβα πλέον άδεια.

	addresses					
Stack	2FFCH			00H		
	2FFDH			09H		
	2FFEH		43H	43H	43H	
	2FFFH		08H	08H	08H	
	3000H					
PC			(PC)<-0900H	(PC)<-002CH	(PC)<-0900H	(PC)<-0843H
Steps			1(CALL 0900)	2(PUSH PSW)	3(POP PSW)	4(RET)

Άσκηση 5

(α)

```
        MVI A,ODH           ;Μάσκα διακοπών
        SIM
        LXI H,00H           ;Συσσυρευτης Δεδομένων
        MVI C,64d           ;64 σε δεκαδικό στον μετρητή δεδομένων
        EI                  ;Ενεργοποιηση των διακοπων
ADDR:                       ;Αναμονη δεδομενων
        MVI A,C
        CPI 00H
        JNZ ADDR            ;Ελεγχος εισοδου των δεδομενων
        DI                  ;Απενεργοποιηση των διακοπων
        DAD H               ;Ολισθηση 3 φορες αριστερα (3 φορες προσθεση HL στον εαυτο του)
        DAD H
        DAD H
        MOV A,L
        ANI 80H
        MVI L,00H           ;L=00H για ακριβεια 8bit
        CPI 00H
        JNZ ROUNDING        ;Αν MSB του L ισο με ενα να τοτε θα γινει ανω στρογγυλοποιηση
BACK:
        HLT
ROUNDING:                   ;Ανω στρογγυλοποιηση
        INR H
        JMP BACK
0034:
        JMP RST6.5
RST6.5
        PUSH PSW
        MOV A,C
        ANI 00000001b       ;00000001 δυαδικο για το LSB
        JPO 4MSB            ;Ελεγχος για το αν λαβαμε τα LSB H τα MSB ΤΟΥ δεδομενου
        IN 20H              ;Ειδοδος των 4LSB του δεδομενου
        ANI 00001111b       ;00001111 δυαδικο για τα 4 LSB Της πορτας
        MOV B,A             ;Προσωρινη αποθηκευση εως οτου λαβουμε τα MSB
        JMP 4LSB            ;Επιστροφη στο προγραμμα εως οτου να ληφθουν τα MSB
4MSB:                       ;Επεξεργασια των 4 MSB του δεδομενου
        IN 20H
        ANI 00001111b
        RLC                 ;4 φορες αριστερη ολισθηση για τοποθετηση του τμηματος δεδομενου στα MSN
        RLC
        RLC
        RLC
        ORA B               ;Ενωση με τα LSB Του δεδομενου
        MVI D,00H
        MOV E,A
        DAD D               ;Προσθεση δεδομενων
4LSB:
        POP PSW
        DCR C               ;Μειωση μετρητη
        EI
        RET
```

(β)

```
        MVI A,0DH           ;Μάσκα διακοπών
        SIM
        LXI H,00H           ;Συσσωρευτης Δεδομένων
        MVI C,64d           ;64 σε δεκαδικό στον μετρητή δεδομένων
        EI                  ;Ενεργοποιηση των διακοπων
ADDR:                       ;Αναμονη δεδομενων
        MVI A,C
        CPI 00H
        JNZ ADDR            ;Ελεγχος εισοδου των δεδομενων
        DI                  ;Απενεργοποιηση των διακοπων
        DAD H               ;Ολισθηση 3 φορες αριστερα (3 φορες προσθεση HL στον εαυτο του)
        DAD H
        DAD H
        MOV A,L
        ANI 80H
        MVI L,00H           ;L=00H για ακριβεια 8bit
        CPI 00H
        JNZ ROUNDING        ;Αν MSB του L ισο με ενα να τοτε θα γινει ανω στρογγυλοποιηση
BACK:
        HLT
ROUNDING:                   ;Ανω στρογγυλοποιηση
        INR H
        JMP BACK
0034:
        JMP RST6.5
RST6.5
        PUSH PSW
        MOV A,C
        ANI 00000001b       ;00000001 δυαδικο για το LSB
        JPO 4MSB            ;Ελεγχος για το αν λαβαμε τα LSB Η τα MSB ΤΟΥ δεδομενου
        IN 20H              ;Ειδοδος των 4LSB του δεδομενου
        ANI 00001111b       ;00001111 δυαδικο για τα 4 LSB Της πορτας
        MOV B,A             ;Προσωρινη αποθηκευση εως οτου λαβουμε τα MSB
        JMP 4LSB            ;Επιστροφη στο προγραμμα εως οτου να ληφθουν τα MSB
4MSB:                       ;Επεξεργασια των 4 MSB του δεδομενου
        IN 20H
        ANI 00001111b
        RLC                 ;4 φορες αριστερη ολισθηση για τοποθετηση του τμηματος δεδομενου στα MSN
        RLC
        RLC
        RLC
        ORA B               ;Ενωση με τα LSB Του δεδομενου
        MVI D,00H
        MOV E,A
        DAD D               ;Προσθεση δεδομενων
4LSB:
        POP PSW
        DCR C               ;Μειωση μετρητη
        EI
        RET
```