




**Εθνικό Μετσόβιο Πολυτεχνείο**

**Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών**

**6<sup>ο</sup> Εξάμηνο**

**Συστήματα Μικροϋπολογιστών**

**3<sup>η</sup> ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ**

  
Χατζή Ήβη

## 1<sup>η</sup> ΑΣΚΗΣΗ

```
START:
IN 10H          ;Remove memory protection
LXI H,0A00H     ;HL(for output byte) = start of memory

MVI A,0DH       ;0DH = 00001101. (Enable interrupts)
SIM
EI

WAIT: JMP WAIT  ;Wait for interupt

INTR_ROUTINE:
MVI A,FFH       ;A=11111111
CMA
STA 3000H       ;All lights ON
MVI E,3CH       ;E=60 (counter)
EI

COUNTER:
CALL HEX_TO_BCD
DCR E           ;E--
MOV A,E
CPI 00H         ;If E=00
JNZ COUNTER     ;Repeat
MVI A,00H       ;A=00000000;
CMA
STA 3000H       ;All lights off after 60 seconds
JMP WAIT

HEX_TO_BCD:
PUSH PSW
PUSH B
PUSH D
PUSH H
MVI B,00H       ;B holds decades, start with 0
MOV A,E         ;A holds units

DEC:
CPI 0AH         ;Compare with 10
JC DONE         ;If smaller then it is the last digit
SUI 0AH         ;Else subtract 10
INR B           ;Increase dacades by one
JMP DEC

DONE:
LXI H,0A02H
MOV M,A         ;Units in 3rd digit of LCD
INX H
MOV M,B         ;Decades in 4th digit of LCD

DISPLAY:
LXI D,0A00H     ;Move the block 0A00H-0A05H to
CALL STDM       ;Memory, where DCD routine reads
```

```
DELAY1:
LXI B,0032H      ;BC = 50
MVI A,14H
DELAY:
CALL DCD          ;Print
CALL DELB          ;Delay for 50 msec
DCR A             ;50 msec X 20 = 1 sec delay
CPI 00H
JNZ DELAY

POP H
POP D
POP B
POP PSW
EI
RET

END
```

## 2<sup>η</sup>ΑΣΚΗΣΗ

```
IN 10H
MVI D,3CH      ;values for thresholds D,E
MVI E,A1H

MVI A,10H      ;digits 1-3 show nothing
STA 0B00H      ;output at 0B00H-0B05H
STA 0B01H
STA 0B02H
STA 0B03H

MVI A,0DH      ;interrupt mask to allow rst 6.5
SIM
EI             ;enable interrupts

WAIT: JMP WAIT

INTR ROUTINE:
POP H
CALL KIND      ;input msb at A
STA 0B05H      ;store msb
RLC
RLC
RLC
RLC            ;move it 4 bits left
MOV B,A
CALL KIND      ;input lsb at A
STA 0B04H      ;store lsb
ADD B          ;get total number and store in B
MOV B,A
PUSH D         ;temporarily save D and E
LXI D,0B00H    ;address for STD
CALL STD
CALL DCD
POP D          ;restore D and E

MOV A,B
CMP D          ;compare with lower threshold D
JC LOW         ;A<D
JZ LOW         ;A=D
CMP E          ;compare with upper threshold E
JC MID         ;A<E
JZ MID         ;A=E

MVI A,04H      ;A>E, set 3rd lsb LED
JMP LIGHT

LOW:
MVI A,01H      ;A<=D, set lsb LED
JMP LIGHT

MID:
MVI A,02H      ;D<A<=E, set 2nd lsb LED
JMP LIGHT
```

```

LIGHT:
CMA
STA 3000H      ;light LED
EI             ;wait for next interrupt
JMP WAIT

END

```

### 3<sup>η</sup> ΑΣΚΗΣΗ

**α)**

```

SWAP nibble MACRO Q      ;Q is register to swap msb,lsb
    PUSH PSW

    MOV A,Q
    RLC
    RLC
    RLC
    RLC
    MOV Q,A

    MOV A,M
    RLC
    RLC
    RLC
    RLC
    MOV M,A

    POP PSW
ENDM

```

**β)**

```

FILL MACRO RP, X, K
    PUSH PSW
    PUSH H
    LXI H,ADDR
    MVI A, X

    LOOP1:
        MOV M,K
        INX H
        DCR A
        JNZ LOOP1

    POP H
    POP PSW
ENDM

```

y)

```
RHLR MACRO n
    PUSH PSW
    PUSH B
    MVI A,n
    CPI 00H
    JZ FINISH
    MVI B,n
```

```
LOOP1:
    MOV A,L
    RAR
    MOV L,A
    MOV A,H
    RAR
    MOV H,A
    DCR B
    JNZ LOOP1
```

```
FINISH:
    POP B
    JNC ZERO
```

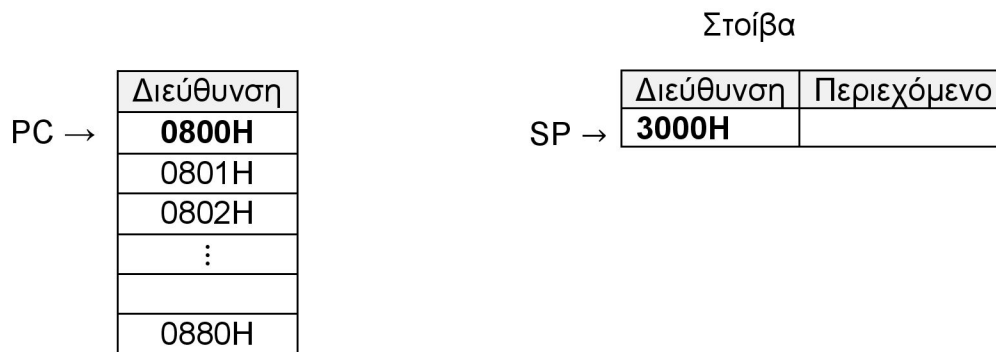
```
ONE:
    POP PSW    ;we do this cause pop psw modifies CY
    STC
    JMP FINISH2
```

```
ZERO:
    POP PSW
    STC
    CMC
```

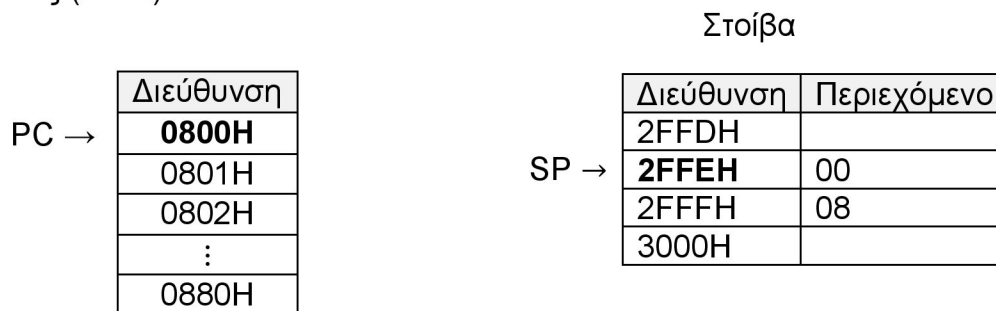
```
FINISH2:
ENDM
```

#### 4<sup>η</sup> ΑΣΚΗΣΗ

Στο μέσον της εκτέλεσης της εντολής CALL 0880H συμβαίνει διακοπή RST 7.5.  
Ο μετρητής προγράμματος αρχικά είναι (PC)=0800H και ο δείκτης σωρού (SP)=3000H.

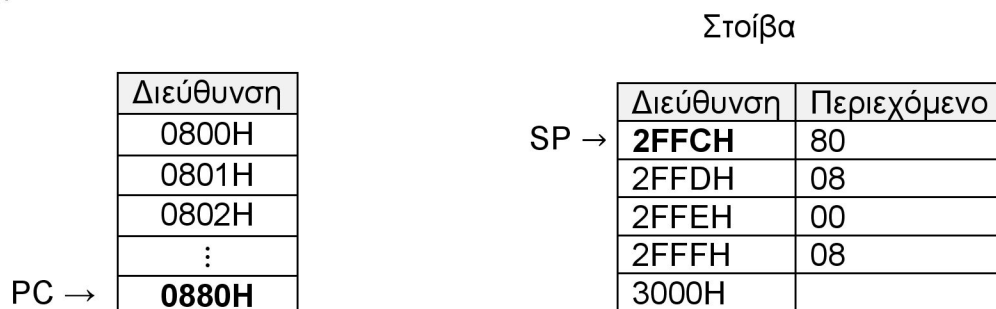


Αρχικά, ολοκληρώνεται η εκτέλεση της τρέχουσας εντολής, δηλαδή, της CALL 0880H.  
Η αρχική τιμή του PC (0800H) αποθηκεύεται στη στοίβα και ο δείκτης στοίβας ανεβαίνει δύο θέσεις (SP-2).

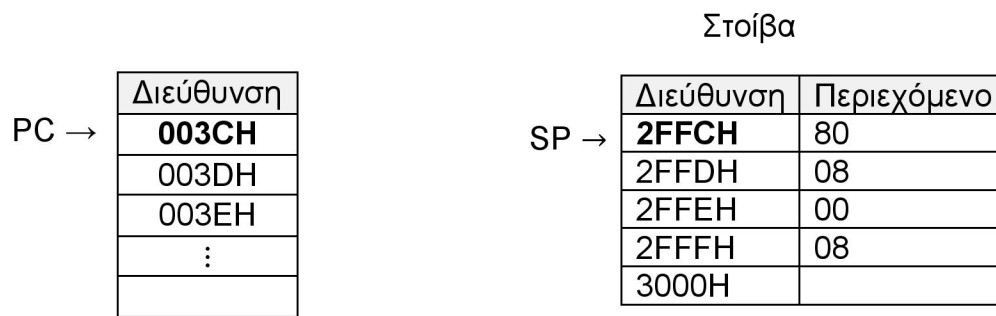


Στη συνέχεια, ο PC λαμβάνει την τιμή 0880H.

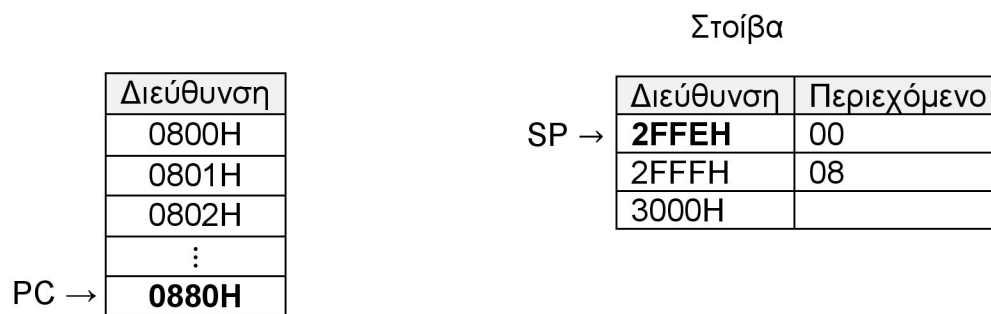
Η νέα αυτή τιμή αποθηκεύεται στη στοίβα και ο δείκτης στοίβας ανεβαίνει άλλες 2 θέσεις (SP-4).



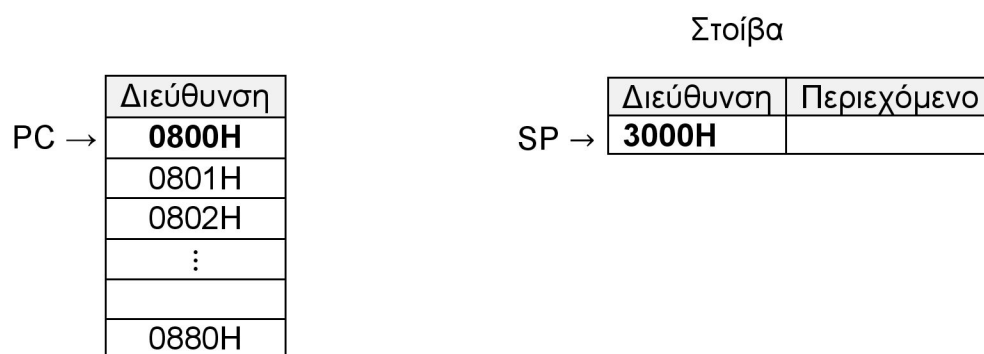
Ακολούθως, ο PC λαμβάνει την τιμή 003C (RST 7.5) και ο έλεγχος δίνεται στη ρουτίνα εξυπηρέτησης.



Μετά την ολοκλήρωση της ρουτίνας εξυπηρέτησης, επαναφέρεται από τη στοίβα η τιμή που είχε το PC πριν συμβεί η διακοπή (0880H) και ο δείκτης στοίβας κατεβαίνει 2 θέσεις (SP-2).



Έτσι, εκτελείται η ρουτίνα που βρίσκεται στη διεύθυνση 0880H. Όταν αυτή ολοκληρωθεί, επαναφέρεται στον PC αρχική του τιμή (0800H) και ο δείκτης στοίβας κατεβαίνει 2 θέσεις, μεταβαίνοντας στην αρχική του θέση. Επομένως, επιστρέφει ο έλεγχος στην επόμενη εντολή από αυτή που έγινε η διακοπή.





## 5<sup>η</sup> ΑΣΚΗΣΗ

(α)

```
START:
MVI A,0EH      ;interrupt mask
SIM
LXI H,0000H
MVI C,40H      ;C counter 40H = 64(dec)
EI             ;enable interrupts

ADDR:
MOV A,C        ;read all data
CPI 00H        ;until C is 0
JNZ ADDR
DI             ;disable interrupts
DAD H          ;average is found by shifting HL three times to the left
DAD H
DAD H
HLT

; in 0034H there is JMP RST6.5 in routine
;0034: JMP RST6.5
RST6.5:
PUSH PSW
MOV A,C
ANI 01H        ;get LSB of counter to see if its odd or even
CPI 00H        ;even 0, input bits are 4 msb
JNZ GET_LSBs  ;else 4 LSBs

GET_MSBs:
IN 20H         ;put 4 MSBs (M3M2M1M0) in A
ANI 0FH        ;isolate A0-A3
RLC            ;shift 4 times
RLC
RLC
RLC
MVI D,00H      ;D = 00000000
MOV E,A        ;DE = 00000000 M3M2M1M0 0000
JMP FINISH

GET_LSBs:
IN 20H         ;put 4 LSBs (L3L2L1L0) in A
ANI 0FH        ;isolate A0-A3
ORA E          ;A = A OR E=(0000L3L2L1L0)OR (M3M2M1M0 0000)=M3M2M1M0L3L2L1L0
MVI D,00H      ;D = 00000000
MOV E,A        ;E = M3M2M1M0L3L2L1L0
DAD D          ;add
DCR C          ;C--

FINISH:
POP PSW
EI             ;enable interrupts
RET
```

(β)

START:

LXI H,0000H

MVI C,40H ;counter C 40H = 64(dec)

WAIT\_FOR\_DATA:

MOV A,C ;get data until counter is 0

CPI 00H

JZ AVERAGE ;counter is 0, go calculate average

IN 20H ;isolate msb (data ready) until it is 1

RLC

JNC WAIT\_FOR\_DATA

RRC

MOV B,A

MOV A,C

ANI 01H ;get counter LSB to see if odd or even

CPI 00H ;if even 0, then input bits are 4 MSBs

JNZ GET\_LSBBS ;else 4 LSBs

GET\_MSBS:

MOV A,B

ANI 0FH ;isolate A0-A3

RLC ;shift 4 times

RLC

RLC

RLC

MVI D,00H ;D = 00000000

MOV E,A ;DE = 00000000 M3M2M1M0 0000

DCR C ;C--

JMP FINISH

GET\_LSBBS:

MOV A,B

ANI 0FH ;isolate A0-A3

ORA E ;A = A or E=(0000L3L2L1L0)or(M3M2M1M0 0000)=M3M2M1M0L3L2L1L0

MVI D,00H ;D = 00000000

MOV E,A ;E = M3M2M1M0L3L2L1L0

DAD D ;add

DCR C ;C--

FINISH:

IN 20H

RLC

JC FINISH

JMP WAIT\_FOR\_DATA

AVERAGE:

DAD H ;average is found by shifting HL three times to the left

DAD H

DAD H

HLT

END