



ΑΘΗΝΑ 23-10-2-21

**1<sup>η</sup> ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ**  
**ΓΙΑ ΤΟ ΜΑΘΗΜΑ “Εργαστήριο Μικροϋπολογιστών”**  
(Στο Εκπαιδευτικό Σύστημα mLAB-8085)

**Ομάδα 41**

Σιόκουρου Αιμιλία – 03117703

Πεγειώτη Νάταλυ – 03117707

**Άσκηση 1**

Ο κώδικας της άσκησης φαίνεται αναλυτικά, με τα απαραίτητα σχόλια, παρακάτω:

IN 10H

LXI B,03E8H     ;delay 1000msec

MAIN:

```
LDA 2000H     ;read input x of dip switch
STA 0900H     ;store the number in a random memory place to check later the MSB
ANI 0FH       ;isolate the 4LSB
MOV H,A
LDA 0900H
ANI 80H       ;isolate the MSB of the dip switch
CPI 80H       ;if MSB is 1 then
JZ UPCOUNT
JMP MAIN
```

UPCOUNT:

MVI A,FFH

UP:

```
STA 3000H
CALL DELB
XRI FFH       ;inverse logic
MOV L,A
JMP MSBONUP
CONTINUEUP:
MOV A,L
CMP H         ;if reached the count number
JZ DOWN
INR A         ;else increase counter
XRI FFH
JMP UP         ;and repeat
```

DOWN:

```
XRI FFH      ;inverse logic on led
STA 3000H    ;LED=4LSB DIP SWITCH
CALL DELB
MOV L,A      ;move A to L to save the value inside it
JMP MSBON
CONTINUEDOWN:
    MOV A,L
    XRI FFH   ;inverse logic to get it to the right logic
    CPI 00H   ;if counter is 0 go to upcount
    JZ CHECK
    DCR A     ;else decrease the counter and repeat
    JMP DOWN
```

CHECK:

```
LDA 2000H    ;read the dip switch
ANI 0FH      ;isolate the 4LSB
CMP H        ;compare with the previous amount of the dip switch
JZ UPCOUNT   ;if it didn't change continue to upcount
MOV H,A      ;else change the new counter
JZ UPCOUNT
```

MSBON:

```
LDA 2000H
ANI 80H      ;check the MSB if it's ON
CPI 80H
JZ CONTINUEDOWN
JMP MSBON
```

MSBONUP:

```
LDA 2000H
ANI 80H      ;check the MSB if it's ON
CPI 80H
JZ CONTINUEUP
JMP MSBONUP
```

END

## Άσκηση 2

Ο κώδικας της άσκησης φαίνεται αναλυτικά, με τα απαραίτητα σχόλια, παρακάτω:

IN 10H

MAIN:

```
MVI L,00H
MVI C,00H
CALL KIND
MOV H,A      ;else just save the first number to register E
CALL KIND
MOV B,A      ;save the second number in B
MOV A,H
RLC          ;4 times rlc to multiply x by 16
RLC
RLC
RLC
ADD B        ;add y
HUNDREDS:
    CPI 64H      ;if not larger than 100
    JC DECADES   ;go to calculate the decades
    SUI 64H      ;else subtract 100 until not over 100
    INR L        ;counter for hundreds
    JMP HUNDREDS ;repeat
```

DECADES:

```
MOV H,A      ;move to register H the 2digit number without the hundreds
MOV A,L      ;move the amount of hundreds to A
STA 0902H    ;store the hundreds in the 3rd place
MOV A,H      ;restore A
```

DEC:

```
CPI 0AH      ;subtract 10 until it is no longer over 10
JC RES
SUI 0AH
INR C        ;counter for decades
JMP DEC
```

RES:

```
STA 0900H
MOV A,C
STA 0901H
LXI D,0900H
CALL STDM
CALL DCD
JMP MAIN
```

END

### Άσκηση 3

Ο κώδικας της άσκησης φαίνεται αναλυτικά, με τα απαραίτητα σχόλια παρακάτω:

```
BEGIN:
LDA 2000H      ;Φόρτωσε στον A την είσοδο
RRC            ;Δεξιά ολίσθηση του A και αποθήκευση του LSB στον CY
JNC BEGIN      ;Αν CY = 0 άλμα στο BEGIN

MVI A,01H      ;Αποθήκευση της θέσης του αναμμένου LED
MVI H,01H      ;Αποθήκευση της τελευταίας κατάστασης του LSB (ON)

LSB2MSB:
CALL CHECK_MSB ;Κλήση της CHECK_MSB

GO_UP:
MVI E,00H      ;Αποθήκευση της κατεύθυνσης 0: LSB->MSB
CALL CHECK_MSB ;Κλήση της CHECK_MSB
CALL CHECK_LSB ;Κλήση της CHECK_LSB
RLC            ;Αριστερή ολίσθηση του A και αποθήκευση του MSB στον CY

CPI 80H        ;Σύγκριση του A με το 80H (1000 0000)
JZ MSB2LSB     ;Αν ισοούνται, άλμα στο MSB2LSB (αλλαγή κατεύθυνσης)
JMP GO_UP      ;Αλλιώς άλμα στο GO_UP (συνεχίζει στην ίδια κατεύθυνση)

MSB2LSB:
CALL CHECK_MSB ;Κλήση της CHECK_MSB

GO_DOWN:
MVI E,01H      ;Αποθήκευση της κατεύθυνσης 1: MSB->LSB
CALL CHECK_MSB ;Κλήση της CHECK_MSB
CALL CHECK_LSB ;Κλήση της CHECK_LSB
RRC            ;Δεξιά ολίσθηση του A και αποθήκευση του LSB στον CY

CPI 01H        ;Σύγκριση του A με το 01H (0000 0001)
JZ LSB2MSB     ;Αν ισοούνται άλμα στο LSB2MSB (αλλαγή κατεύθυνσης)
JMP GO_DOWN    ;Αλλιώς άλμα στο GO_DOWN(συνεχίζει στην ίδια κατεύθυνση)

CHECK_MSB:
    MOV D,A      ;Αποθήκευση της τιμής του A
    STALL:
    LDA 2000H    ;Φόρτωση της εισόδου στον A
    RLC          ;Αριστερή ολίσθηση του A και αποθήκευση του MSB στον CY
    JNC STALL    ;Αν CY = 0, άλμα στο STALL
    MOV A,D      ;Επαναφορά της τιμής του A

    CMA          ;Συμπλήρωμα ως προς 1 του A και καταχώρηση του στον ίδιο
    STA 3000H    ;Αποθήκευση της τιμής του A στην έξοδο (άναμμα LED)
    LXI B,01F4H ;Φόρτωση στον BC της τιμής 500
    CALL DELB    ;Καθυστέρηση BC*1ms
    CMA          ;Συμπλήρωμα ως προς 1 του A και καταχώρηση του στον ίδιο
RET
```

CHECK\_LSB:

MOV D,A ;Αποθήκευση της τιμής του A  
LDA 2000H ;Φόρτωση της εισόδου στον A  
RRC ;Δεξιά ολίσθηση και αποθήκευση του LSB στον CY  
JC LSB\_ON ;Αν CY = 1, άλμα στο LSB\_ON

MOV A,H ;Αποθήκευση στον A της τελευταίας τιμής του LSB  
CPI 00H ;Σύγκριση του A με το 0  
JZ SKIP ;Αν ισούνται άλμα στο SKIP

MVI H,00H ;Αλλιώς αποθήκευση της νέας τιμής του LSB (OFF -> αλλαγή κατεύθυνσης)  
MOV A,D ;Επαναφορά της τιμής του A  
CPI 80H ;Σύγκριση του A με το 80H (1000 0000)  
JZ SKIP ;Αν ισούνται άλμα στο SKIP  
CPI 01H ;Σύγκριση του A με το 01H (0000 0001)  
JZ SKIP ;Αν ισούνται άλμα στο SKIP  
MOV A,E ;Αλλιώς φόρτωση του E (κατεύθυνση) στον A  
CPI 00H ;Σύγκριση του A με το 00H  
MOV A,D ;Επαναφορά του A  
JZ GO\_DOWN ;Αν η κατεύθυνση ήταν 0 τότε άλμα στο GO\_DOWN  
JMP GO\_UP ;Αλλιώς άλμα στο GO\_UP

LSB\_ON:

MVI H,01H ;Μετατροπή της τελευταίας κατάστασης του LED (ON)

SKIP:

MOV A,D ;Επαναφορά της τιμής του A

RET

END