ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Εργαστήριο Μικροϋπολογιστών

$1^{η}$ εργαστηριακή άσκηση

mLab 8085

Αναστάσιος Λαγός - 03113531
Αντώνιος Δημήτριος Αλικάρης - 03118062

Άσκηση 1

Γενική Ιδέα

Αφού διαβάσουμε τα dip switches και ξεκινώντας απο το '0' (σβηστά όλα τα LEDs) αυξάνουμε τα LEDs κατα '1' και ελέγχουμε αν έφτασε στην τιμή των switches. Στην περίπτωση που έφτασε, μειώνουμε τα LEDs κατα '1' μέχρι να φτάσουν στο '0' και το πρόγραμμα ξεκινάει από την αρχή.

Κώδικας

```
LXI B,03E8H ; B = 1000d -> delay = 1s
MVI D,00H

START:
        LDA 2000H
        ANI 0FH
        JZ START        ;check if zero dip switches are turned on
        MOV E,A            ;make E the LSB value
        CALL CHECK        ;if MSB is on continue, else wait for it to turn on
INC:
        MOV A,D
        CMA
        STA 3000H        ;show the value of D in the LEDS
        CALL DELB        ;delay 1 sec
        CALL CHECK        ;same as before
        INR D               ;D++
        MOV A,D
        CMP E
        JC INC                ;while D<E go to INC
DEC:
        MOV A,D
        CMA
        STA 3000H
        CALL DELB
        CALL CHECK
        DCR D               ;D--
        JNZ DEC         ;while D>0 go to DEC
        JMP START        ;if D = 0 then go to Start to refresh the value of E



CHECK:
        LDA 2000H
        ANI 80H
        CPI 80H
        JNZ CHECK
        RET

```

```
END
```

Άσκηση 2

Γενική ιδέα

Αρχικά διαβάζουμε το input και εκτελούμε την πράξη $16x + y$. Έπειτα μετατρέπουμε το αποτέλεσμα στο δεκαδικό σύστημα και το εμφανίζουμε στο 7-seg display. Η μετατροπή γίνεται μετρώντας διαδοχικά τις εκατοντάδες/δεκάδες/μονάδες του αποτελέσματος.

Κώδικας

```
MVI A,10H        ;empty space on 7seg
STA 0B53H
STA 0B54H
STA 0B55H

START:
CALL KEYBOARDINPUT        ;read the first number
MOV B,A
CALL KEYBOARDINPUT        ;read the second number
MOV C,A

MOV A,B
RLC               ;4 RLC commands implement 16*x
RLC
RLC
RLC
ADD C             ; 16*x + y
CALL HEXTODEC
LXI D,0B50H
CALL STDM
CALL DCD
JMP START

KEYBOARDINPUT:        ;routine that reads a 1-bit number [0-F] from the keyboard
CALL KIND
CPI 10H
JNC KEYBOARDINPUT
RET

HEXTODEC:

MVI B,00H        ;counts the # of hundreds/tens/ones


HUNDREDS:
```

```
CPI 64H         ;if  A < 100 store B for hundreds
JC STOREHUNDREDS
; else inr # of hundreds and decrease number by 100 and repeat
INR B
SUI 64H
JMP HUNDREDS
STOREHUNDREDS:
MOV C,A         ;temp store number
MOV A,B         ;store # of hundreds in random address 0B52H for STDM
STA 0B52H
MOV A,C
MVI B,00H         ;reinitialize B for tens

TENS:
CPI 0AH         ;if A < 10
JC STORETENS
;same as hundreds
INR B
SUI 0AH
JMP TENS
STORETENS:
MOV C,A
MOV A,B
STA 0B51H
MOV A,C
MVI B,00H

ONES:
CPI 01H         ; if A<1
JC STOREONES
;same as ones
INR B
SUI 01H
JMP ONES
STOREONES:
MOV C,A
MOV A,B
STA 0B50H
MOV A,C

RET

END
```

Άσκηση 3

Γενική Ιδέα

Η λογική που ακολουθείται είναι παρόμοια της πρώτης άσκησης. Η κίνηση του LED γίνεται μέσω bit-wise shift operations. Επιπλέον, η χρονική καθυστέρηση λαμβάνει μέρος πριν πραγματοποιηθεί ο έλεγχος αφίξης σε κάποιο άκρο, έτσι επιτυγχάνουμε 0.5 sec επιπλέον καθυστέρηση στα άκρα. Ο έλεγχος για αλλαγή κατεύθυνσης έχει δυο συνθήκες με σειρά προτεραιότητας:

1. Το state του LSB να είναι διαφορετικό απο την προηγούμενη αποθυκευμένη τιμή.
2. Το νέο state να είναι '0'.

Κώδικας

```
LXI B,01F4H             ; B = 500d -> delay = 0.5s
MVI D,01H                ;counter
MVI E,01H                ;register to remember the state of the LSB

INC:
        MOV A,D
        CMA
        STA 3000H            ;show the value of D in the LEDS
        CALL DELB            ;delay 0.5 sec
INCN:
        CALL CHECK ;checking if the MSB is on, if not then we wait
        LDA 2000H
        ANI 01H
        CMP E                    ;checks if the state of the LSB has changed
        JNZ CHANGESTATE1    ;if yes, then goes to CHANGESTATE1
CONT:
        MOV A,D
        CPI 80H
        JZ DEC                ;if it has reached the MSB then go to DEC
        RLC                    ;else move it one position to the left
        MOV D,A
        JMP INC

DEC:
        MOV A,D
        CMA
        STA 3000H            ;show the value of D in the LEDS
        CALL DELB            ;delay 0.5 sec
DECN:                        ;checking if the MSB is on, if not then we wait
        CALL CHECK
        LDA 2000H
        ANI 01H
        CMP E                    ;checks if the state of the LSB has changed
        JNZ CHANGESTATE2 ;if yes, then goes to CHANGESTATE2
```

```
CONT2:
        MOV A,D
        CPI 01H
        JZ INC                ;if it has reached the LSB then go to INC
        RRC                    ;else move it one position to the right
        MOV D,A
        JMP DEC

;change the value of E and check if the change went from ON->OFF
CHANGESTATE1:
        MOV E,A             ;E changes State
        CPI 00H
        JZ DECN             ;if A=0 the LSB went from ON->OFF so we change
        ↪  direction
                                ;DECN if we don't want double delay for the
                                ↪  displayed led
                                ;if we don't mind we can simply put DEC
        JMP CONT         ;else it continues

;change the value of E and check if the change went from ON->OFF
;same as CHANGESTATE2
CHANGESTATE2:
        MOV E,A
        CPI 00H
        JZ INCN
        JMP CONT2

;CHECK POWER (ON-OFF) ROUTINE
CHECK:
    LDA 2000H
        ANI 80H
        CPI 80H
        JNZ CHECK
        RET

END
```