

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

Κώδικας

```
1  LXI B,03E8H ; B = 1000d -> delay = 1s
2  MVI D,00H
3
4  START:
5      LDA 2000H
6      ANI 0FH
7      JZ START      ;check if zero dip switches are turned on
8      MOV E,A        ;make E the LSB value
9      CALL CHECK     ;if MSB is on continue, else wait for it to turn on
10
11 INC:
12     MOV A,D
13     CMA
14     STA 3000H      ;show the value of D in the LEDS
15     CALL DELB      ;delay 1 sec
16     CALL CHECK     ;same as before
17     INR D          ;D++
18     MOV A,D
19     CMP E
20     JC INC         ;while D<E go to INC
21
22 DEC:
23     MOV A,D
24     CMA
25     STA 3000H
26     CALL DELB
27     CALL CHECK
28     DCR D          ;D--
29     JNZ DEC        ;while D>0 go to DEC
30     JMP START     ;if D = 0 then go to Start to refresh the value of E
31
32 CHECK:
33     LDA 2000H
34     ANI 80H
35     CPI 80H
36     JNZ CHECK
37     RET
38
```

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

```

1  MVI A,10H           ;empty space on 7seg
2  STA 0B53H
3  STA 0B54H
4  STA 0B55H
5
6  START:
7  CALL KEYBOARDINPUT   ;read the first number
8  MOV B,A
9  CALL KEYBOARDINPUT   ;read the second number
10 MOV C,A
11
12 MOV A,B
13 RLC                  ;4 RLC commands implement 16*x
14 RLC
15 RLC
16 RLC
17 ADD C                ; 16*x + y
18 CALL HEXTODEC
19 LXI D,0B50H
20 CALL STDM
21 CALL DCD
22 JMP START
23
24 KEYBOARDINPUT:       ;routine that reads a 1-bit number [0-F] from the keyboard
25 CALL KIND
26 CPI 10H
27 JNC KEYBOARDINPUT
28 RET
29
30 HEXTODEC:
31
32 MVI B,00H            ;counts the # of hundreds/tens/ones
33
34
35 HUNDREDS:

```

```

36 CPI 64H           ;if A < 100 store B for hundreds
37 JC STOREHUNDREDS
38 ; else inr # of hundreds and decrease number by 100 and repeat
39 INR B
40 SUI 64H
41 JMP HUNDREDS
42 STOREHUNDREDS:
43 MOV C,A           ;temp store number
44 MOV A,B           ;store # of hundreds in random address 0B52H for STDM
45 STA 0B52H
46 MOV A,C
47 MVI B,00H         ;reinitialize B for tens
48
49 TENS:
50 CPI 0AH           ;if A < 10
51 JC STORETENS
52 ;same as hundreds
53 INR B
54 SUI 0AH
55 JMP TENS
56 STORETENS:
57 MOV C,A
58 MOV A,B
59 STA 0B51H
60 MOV A,C
61 MVI B,00H
62
63 ONES:
64 CPI 01H           ; if A<1
65 JC STOREONES
66 ;same as ones
67 INR B
68 SUI 01H
69 JMP ONES
70 STOREONES:
71 MOV C,A
72 MOV A,B
73 STA 0B50H
74 MOV A,C
75
76 RET
77
78 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 από την προηγούμενη αποθηκευμένη τιμή.
2. Το νέο state να είναι '0'.

Κώδικας

```
1 LXI B,01F4H          ; B = 500d -> delay = 0.5s
2 MVI D,01H            ;counter
3 MVI E,01H            ;register to remember the state of the LSB
4
5 INC:
6     MOV A,D
7     CMA
8     STA 3000H          ;show the value of D in the LEDS
9     CALL DELB          ;delay 0.5 sec
10
11 INCN:
12     CALL CHECK ;checking if the MSB is on, if not then we wait
13     LDA 2000H
14     ANI 01H
15     CMP E              ;checks if the state of the LSB has changed
16     JNZ CHANGESTATE1   ;if yes, then goes to CHANGESTATE1
17
18 CONT:
19     MOV A,D
20     CPI 80H
21     JZ DEC              ;if it has reached the MSB then go to DEC
22     RLC                 ;else move it one position to the left
23     MOV D,A
24     JMP INC
25
26 DEC:
27     MOV A,D
28     CMA
29     STA 3000H          ;show the value of D in the LEDS
30     CALL DELB          ;delay 0.5 sec
31
32 DECN:
33     CALL CHECK ;checking if the MSB is on, if not then we wait
34     LDA 2000H
35     ANI 01H
36     CMP E              ;checks if the state of the LSB has changed
37     JNZ CHANGESTATE2   ;if yes, then goes to CHANGESTATE2
```

```

35  CONT2:
36      MOV A,D
37      CPI 01H
38      JZ INC          ;if it has reached the LSB then go to INC
39      RRC              ;else move it one position to the right
40      MOV D,A
41      JMP DEC
42
43      ;change the value of E and check if the change went from ON->OFF
44  CHANGESTATE1:
45      MOV E,A          ;E changes State
46      CPI 00H
47      JZ DECN          ;if A=0 the LSB went from ON->OFF so we change
        ↪ direction
48                      ;DECN if we don't want double delay for the
        ↪ displayed led
49                      ;if we don't mind we can simply put DEC
50      JMP CONT        ;else it continues
51
52      ;change the value of E and check if the change went from ON->OFF
53      ;same as CHANGESTATE2
54  CHANGESTATE2:
55      MOV E,A
56      CPI 00H
57      JZ INCN
58      JMP CONT2
59
60      ;CHECK POWER (ON-OFF) ROUTINE
61  CHECK:
62      LDA 2000H
63      ANI 80H
64      CPI 80H
65      JNZ CHECK
66      RET
67
68  END

```