



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ – 7^ο ΕΞΑΜΗΝΟ

Δημητρίου Αγγελική – ΑΜ: 03117106

Τζομάκα Αφροδίτη – ΑΜ: 03117107

Ομάδα Β8

1^η Εργαστηριακή Άσκηση

ΑΣΚΗΣΗ 1

Ξεκινάμε ανιχνεύοντας τον αριθμό x από τα 4 LSBs και ελέγχοντας το MSB. Αν αυτό είναι 0, περιμένουμε να γίνει 1 (κατάσταση ON) ελέγχοντας και τυχόν αλλαγές στο x. Αν αυτό είναι 1, ξεκινάει κανονικά η μέτρηση προς τα άνω μέχρι τον x και ύστερα προς τα κάτω μέχρι το 0. Μόλις φτάσουμε στο 0 ξεκινάμε από την αρχή έτσι ώστε να ανιχνεύσουμε πιθανές αλλαγές στο x.

```
LXI B,03E8H      ;BC=1000 for 1 sec delay
MVI D,FFH        ;D is the actual counter
START: LDA 2000H
ANI 0FH           ;keep the 4 LSBs
MOV E,A           ;store x in E
UP_COUNT: LDA 2000H
ANI 80H
JZ START          ;check if MSB is 0 and if x has changed
INR D
MOV A,D
CMA
STA 3000H
CMA
CALL DELB
CMP E              ;check if we reached x
JNZ UP_COUNT
DN_COUNT: LDA 2000H
ANI 80H
JZ DN_COUNT       ;check if MSB is 0
DCR D
MOV A,D
CMA
STA 3000H
CMA
CALL DELB
CPI 00H           ;check if we reached 0
JNZ DN_COUNT
JMP START
END
```

ΑΣΚΗΣΗ 2

Κατά τα γνωστά, θεωρώντας ότι οι διευθύνσεις από 0900^H έως 0905^H αντιστοιχίζονται στο 7 segment display από δεξιά προς αριστερά, γεμίζουμε με κενά τα 3 αριστερότερα 7 segments και θα αποθηκεύσουμε το αποτέλεσμα στα 3 δεξιότερα. Διαβάζουμε τα x,y από το πληκτρολόγιο και κάνουμε τις αντίστοιχες πράξεις σε δεκαεξαδική μορφή. Ύστερα μετατρέπουμε το αποτέλεσμα σε δεκαδική μορφή, το αποθηκεύουμε όπως περιγράψαμε παραπάνω και καλούμε τις ρουτίνες STDM και DCD για το display στα 7segments.

```
START: IN 10H
        MVI A,10H          ;fill the 3 memory positions
        STA 0903H          ;that correspond to the left most
        STA 0904H          ;7-seg displays with empty spaces
        STA 0905H

        CALL KIND           ;read x from keyb
        RLC
        RLC
        RLC
        RLC
        MOV B,A            ;B holds 16*x
        CALL KIND           ;read y from keyb
        ADD B              ;A holds 16*x+y in hex

TODEC:  MOV C,A            ;C holds the hex result
        XRA A
        MVI D,00H         ;D holds the hundreds
LOOP1:  ADI 01H
        DAA               ;decimal adjustment
        JNC CONT          ;if MS hex digit is >9 CY=1
        INR D             ;so we increment the hundreds
CONT:   DCR C              ;like a counter
        JNZ LOOP1

        MOV B,A
        ANI 0FH
        STA 0900H          ;units at the last 7-seg
        MOV A,B
        ANI F0H
        RRC
        RRC
        RRC
        RRC
        STA 0901H          ;decades
        MOV A,D
        STA 0902H          ;hundreds

        LXI D,0900H        ;load DE with the starting address of the message for STDM
        CALL STDM

        CALL DCD
        JMP START

END
```

ΑΣΚΗΣΗ 3

Ο κώδικας ελέγχει συνεχώς τα MSB και LSB της εισόδου. Το βαγονέτο βρίσκεται τέρμα δεξιά και ξεκινάει να κινείται προς τα αριστερά μόνο όταν και τα δύο αυτά bits είναι ON. Αρχικά ελέγχεται η εναλλαγή του LSB που σηματοδοτεί αλλαγή κατεύθυνσης. Αν υπάρχει, ανανεώνεται ο καταχωρητής θέσης κατάλληλα και στην συνέχεια μεταβαίνουμε στο αντίστοιχο τμήμα κώδικα που ελέγχει την ανάποδη κίνηση. Στη συνέχεια ελέγχεται αν το MSB είναι ON. Αν δεν είναι, το βαγονέτο σταματάει, ωστόσο η κατεύθυνσή του έχει αποθηκευτεί και θα συνεχίσει κανονικά όταν αυτό γίνει ON. Σε περίπτωση που είμαστε σε λειτουργία OFF και το LSB αλλάξει μονές φορές η εναλλαγή στην κατεύθυνση θα ανιχνευθεί στον επόμενο 'κύκλο' (εφόσον η άσκηση ορίζει ότι όταν είμαστε OFF πρέπει να διατηρείται η μνήμη της κατεύθυνσης του βαγονέτου).

```
LXI B,01F4H          ;delay
MVI E,01H            ;direction
MVI D,01H            ;cart position - starts from the LSB

MOV A,D              ;show the starting point
CMA
STA 3000H
MVI D,02H            ;load next pos

START: LDA 2000H
ANI 81H              ;check if lsb and msb are both on to start
CPI 81H
JNZ START
;right to left
D1: LDA 2000H
ANI 01H              ;checking for transition (on->off, off->on)
CMP E
JZ CONT
MOV E,A              ;transition found,change direction and save it
FIX1: MOV A,D
RRC                  ;moving cart back to its current position
JC S1                ;checking if cart at far right
RRC                  ;if not, move cart one position to the right (now
switched)
JMP S1
S1: RLC               ;if yes, move back to starting position
SS1: MOV D,A
JMP CONT2            ;and finally switch direction
CONT: LDA 2000H
RLC                  ;checking for off
JNC CONT             ;msb off - stop and loop
MOV A,D
CMA
STA 3000H            ;show
CALL DELB
CMA
RLC                  ;save next position
MOV D,A
JNC D1
MVI D,80H            ;dont forget, it stays at the ends for double the time

;left to right (same logic)
D2: LDA 2000H
ANI 01H
CMP E
JZ CONT2
MOV E,A
FIX2: MOV A,D
```

```
        RLC
        JC S2
        RLC
        JMP SS2
S2:      RRC
SS2:     MOV D,A
        JMP CONT
```

```
CONT2:  LDA 2000H
        RLC
        JNC CONT2
        MOV A,D
        CMA
        STA 3000H
        CALL DELB
        CMA
        RRC
        MOV D,A
        JNC D2
        MVI D,01H
        JMP D1
```

```
END
```