

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Εργαστήριο Μικροϋπολογιστών

1^η Εργαστηριακή Άσκηση

Ονοματεπώνυμο: **Μπέτζελος Χρήστος, Γιαννιός
Γεώργιος-Ταξιάρχης**

A.M. : **031 16 067, 031 16 156**

Ομάδα : **A 16**

Εξάμηνο: **7^ο**

1^η Άσκηση

Κατασκευάζεται χρονόμετρο δευτερολέπτων που απεικονίζει το χρόνο σε δυαδική μορφή πάνω στα LEDs εξόδου του mLab. Θεωρούμε ότι το άναμμα των LED αντιστοιχεί σε λογικό '1' του μετρητή και αντίστροφα. Για την υλοποίηση της χρονοκαθυστέρησης του 1 sec χρησιμοποιήσαμε την έτοιμη ρουτίνα DELB. Το χρονόμετρο όταν φτάνει στην τιμή $x \in [0...15]$ που καθορίζεται από τις τιμές των 4ων αριστερότερων (MSB) της θύρας 2000 Hex αρχίζει την μέτρηση προς τα κάτω έως το 0 και μετά προς τα πάνω συνεχόμενα. Τα παραπάνω με την προϋπόθεση το LSB των διακοπών να είναι ON. Αλλιώς η μέτρηση σταματάει μέχρι να ξανα-ενεργοποιηθεί ο LSB διακόπτης. Διευκρινίζεται ότι για την περίπτωση που αλλάξουν τιμή οι 4 MSB διακόπτες στην διάρκεια της μέτρησης, θεωρούμε ότι το x αποθηκεύει την τιμή του όταν περνάει από το '0' και κρατάει την τιμή αυτή σταθερή μέχρι να ξαναπεράσει από το '0'.

```
START:
IN 10H          ;Αίρει την προστασία της μνήμης
LXI B,03E8H     ;Εκχωρεί στον διπλό καταχωρητή BC την τιμή 03E8 για την DELB
                ;για καθυστέρηση (03E8)16 =(1000)10= 1s.
ZERO:           ;Κλειστοί διακόπτες -> όλα κλειστά
CMA
STA 3000H
CMA
LDA 2000H
CPI 00H
JZ ZERO

MAIN:
LDA 2000H       ; Φορτώνω την είσοδο που δίνω από τον διακόπτη στον A
CPI 00H
JZ ZERO
MOV D,A         ; Χρησιμοποιώ τον D καταχωρητή για να διατηρήσει προσωρινά τα
                ; δεδομένα του A
```

```

RAR          ; Περιστρέφω μια δεξιά τα bits του A ώστε να αποθηκευτεί στο CY
το LSB
JC MAIN      ; Αν το LSB είναι 1 τότε «πήδα» στην MAIN δηλαδή ξαναπάρε είσοδο
(μην        ; αλλάξεις κίνηση)

MOV A,D
RRC          ; κάνουμε 4 περιστροφές ώστε τα MSBs να γίνουν LSBs
RRC
RRC
RRC
ANI 0FH      ; Απομόνωση των 4LSB που είναι ο αρχικός αριθμός
MOV E,A
MVI D,00H    ; αρχικοποιώ στον D το μηδέν

UP:
LDA 2000H    ; έλεγχος LSB
RAR
JC UP
MOV A,D      ; φόρτωση μετρητή
CMA
STA 3000H    ; εμφάνιση στην έξοδο
CMA
CMP E        ; συγκρίνουμε με τον αριθμό εισόδου
JZ DOWN      ; αν είναι ίσος πήγαινε "κάτω"
CALL DELB    ; Καθυστερεί για 1 sec .
INR A        ; αυξάνουμε τον μετρητή κατά ένα
MOV D,A
JMP UP

DOWN:
LDA 2000H    ; έλεγχος LSB
RAR
JC DOWN
MOV A,D      ; φόρτωση μετρητή
CMA
STA 3000H    ; εμφάνιση στην έξοδο
CMA
CALL DELB    ; Καθυστερεί για 1 sec
DCR A        ; μειώνουμε τον μετρητή κατά ένα
JZ MAIN      ; DCR επηρεάζει σημαία Z οπότε όταν φτάσει στο μηδέν
              ; πήγαινε στην MAIN για να ξαναελέγξεις τον αρχικό αριθμό

MOV D,A
JMP DOWN
END

```

2^η Άσκηση

Μετατρέπεται το πρόγραμμα διακοπής (πίνακες 2.1, 2.3 και 2.4, σελ. 49, 55, 56 του παραπάνω βιβλίου των σημειώσεων του εργαστηρίου) έτσι ώστε να επιτρέπει διακοπές μόνο όταν το LSB της θύρας 2000 Hex είναι ON, αλλιώς όχι. Ο μετρητής που αποτελεί το κύριο πρόγραμμα απεικονίζει την μέτρηση στα 4 MSB των LEDs (στην θύρα εξόδου 3000 Hex) και τρέχει με ταχύτητα μίας μέτρησης ανά 0.2 του δευτερολέπτου. Αντί της λειτουργίας beep (πίνακας 2.4) ως ρουτίνας εξυπηρέτησης διακοπής ενσωματώθηκε μια άλλη που μετράει και απεικονίζει το πλήθος των διακοπών που προκαλούνται modulo 10, στο δεξιότερο 7-segment display σε δεκαδική μορφή ενός ψηφίου.

```

IN 10H       ;Αίρει την προστασία της μνήμης

```

```

LDA 2000H      ;Έλεγχος LSB
ANI 01H
CPI 01H
JNZ CONTINUE

;Ενεργοποίηση της διακοπής RST6.5
MVI A,0DH      ;A = 00001101hex
SIM            ;Set Interrupt Mask
EI            ;Ενεργοποίηση των διακοπών (Enable Interrupt)

;Αρχικοποίησης της 7-segment οθόνης
;και φόρτωση στη μνήμη του κωδικού 10H
;για κενό χαρακτήρα

;Αρίθμηση απο δεξιά προς τα αριστερά

CONTINUE:

LXI H,0BF1H      ;2ο ψηφίο
MVI M,10H

INX H            ;3ο ψηφίο
MVI M,10H

INX H            ;4ο ψηφίο
MVI M,10H

INX H            ;5ο ψηφίο
MVI M,10H

INX H            ;6ο ψηφίο
MVI M,10H

LXI H,0BF0H      ;1ο ψηφίο
MVI M,00H

LXI B,00C8H      ;Καθυστέρηση 200msec

COUNTER_FROM_ZERO:
MVI E,00H        ;Αρχικοποίηση μετρητή LED

COUNTER:
    LDA 2000H      ;Έλεγχος lsb
    ANI 01H
    CPI 01H
    JZ ENABLE_2    ;
    DI            ;lsb κλειστο -> απενεργοποίηση διακοπών
    MVI D,00H      ;D σημαία : 0, όταν οι διακοπές απενεργοποιημένες
                    ;                1, αλλιώς

CONT:
    MOV A,E

    CPI F8H        ;σύγκριση με επόμενη κατάσταση (απο F0H)
    JZ COUNTER_FROM_ZERO

    ANI F0H        ;απομόνωση 4 msb

    CMA
    CALL DELB

    STA 3000H      ;εμφάνιση στα led
    CMA
    ADI 10H        ;αύξηση κατά 8
    MOV E,A        ;ενημέρωση μετρητή
    CALL DCD       ;ανανέωση οθόνης
    JMP COUNTER

```

```

ENABLE_2:
    MVI A,0DH      ;A = 00001101hex
    SIM            ;Set Interrupt Mask
    EI             ;Ενεργοποίηση των διακοπών (Enable Interrupt)
    MVI D,01H
    JMP CONT

INTR_ROUTINE:
    PUSH PSW       ;Αποθήκευσης A,σημαίων στο stack
    MOV A,D        ;έλεγχος σημαίας D (Υπόδειξη 2)
    CPI 00H
    JZ NOTCASE
    INR M          ;αύξηση μνήμης κατά ένα (διακοπών)
    MOV A,M
    CPI 0AH        ;για μεγαλύτερες ή ίσες με 10 πήγαινε στο "greater"
    JNC GREATER    ;για υπολογισμό modulo
    CALL DCD       ;ανανέωση οθόνης
    EI
    POP PSW        ;επαναφορά A,σημαίων από stack
    RET            ;επιστροφή στο κυρίως πρόγραμμα

GREATER:
    SUI 0AH        ;Μεγαλύτερη τιμή απο 10-> Αφαίρεση 10
    MOV M,A
    CALL DCD       ;ανανέωση οθόνης
    EI
    POP PSW
    RET

NOTCASE:
    POP PSW        ;Όχι διακοπές όσο διακοπές απενεργοποιημένες
    RET            ;επιστροφή στο κυρίως πρόγραμμα

END

```

3^η Άσκηση

Γράφουμε πρόγραμμα που εισάγει από το πληκτρολόγιο δύο δεκαεξαδικούς αριθμούς $x, y \in \{0-F\}$. Στη συνέχεια απεικονίζει στα τρία αριστερότερα 7-segment display την ποσότητα $x-y$ σε δεκαδική μορφή δύο ψηφίων με το πρόσημο. Η διαδικασία είναι συνεχόμενη δηλαδή κάθε φορά μετά από την εισαγωγή 2 HEX αριθμών ανανεώνεται η οθόνη.

```

IN 10H            ;αίρει την προστασία της μνήμης
START:
LXI H,0A05H      ;Θεση μνήμης που θα βάλουμε το αποτέλεσμα(τυχαία)
CALL KIND        ; Διάβασε 1ο πλήκτρο
MOV B,A
CALL KIND        ; Διάβασε 2ο πλήκτρο
MOV C,A          ; Το x στον A, το y στον B
MOV A,B          ; μετά από swap
MOV B,C
CMP B            ; CY == (x<y)
JC NEGATIVE

POSITIVE:        ; x-y>=0
SUB B            ; A = x-y
CPI 0AH          ; Αν A>=10 -> διψήφιος αριθμός
JNC DOUBLE_DIGIT_POSITIVE
MOV M,A          ;Αρχικοποίησης της 7-segment οθόνης
                ;όλα κενά εκτός από το πρώτο αριστερά

DCX H
MVI M,10H

```

```

DCX H
MVI M,10H
DCX H
MVI M,10H
DCX H
MVI M,10H
DCX H
MVI M,10H
LXI D,0A00H ;Βάζουμε στον D-E την θέση μνήμης όπου αρχίζουν τα
CALL STDM ;δεδομένα που θέλουμε να δείξουμε καλούμε την STDM
CALL DCD ;και εμφανίζουμε με την DCD
JMP START

NEGATIVE: ; x-y<0
MOV C,A ; μετά από swap καταλήγουμε στο A=y-x
MOV A,B
SUB C
CPI 0AH ; Αν A>=10 -> διψήφιος αριθμός
JNC DOUBLE_DIGIT_NEGATIVE
MVI M,1CH ;Αρχικοποίησης της 7-segment οθόνης
;όλα κενά εκτός από το πρώτο αριστερά (πρόσημο -) και το
;δεύτερο από αριστερά ο αριθμός A

DCX H
MOV M,A
DCX H
MVI M,10H
DCX H
MVI M,10H
DCX H
MVI M,10H
DCX H
MVI M,10H
LXI D,0A00H ;Βάζουμε στον D-E την θέση μνήμης όπου αρχίζουν τα
CALL STDM ;δεδομένα που θέλουμε να δείξουμε καλούμε την STDM
CALL DCD ;και εμφανίζουμε με την DCD
JMP START

DOUBLE_DIGIT_POSITIVE:
SUI 0AH ;A=A-10
MVI M,01H ;Αρχικοποίησης της 7-segment οθόνης
;όλα κενά εκτός από το πρώτο αριστερά ψηφίο 1 και το
;δεύτερο από αριστερά ο αριθμός A

DCX H
MOV M,A
DCX H
MVI M,10H
DCX H
MVI M,10H
DCX H
MVI M,10H
DCX H
MVI M,10H
LXI D,0A00H ;Βάζουμε στον D-E την θέση μνήμης όπου αρχίζουν τα
CALL STDM ;δεδομένα που θέλουμε να δείξουμε καλούμε την STDM
CALL DCD ;και εμφανίζουμε με την DCD
JMP START

DOUBLE_DIGIT_NEGATIVE:
SUI 0AH
MVI M,1CH ;Αρχικοποίησης της 7-segment οθόνης
;όλα κενά εκτός από το πρώτο αριστερά (πρόσημο -)
;το δεύτερο από αριστερά (ψηφίο 1)
;και το τρίτο από αριστερά ψηφίο ο αριθμός A

DCX H
MVI M,01H
DCX H

```

```

MOV M,A
DCX H
MVI M,10H
DCX H
MVI M,10H
DCX H
MVI M,10H
LXI D,0A00H ;Βάζουμε στον D-E την θέση μνήμης όπου αρχίζουν τα
CALL STDM ;δεδομένα που θέλουμε να δείξουμε καλούμε την STDM
CALL DCD ;και εμφανίζουμε με την DCD
JMP START

END

```

4^η Άσκηση

Εξομοιώνεται ένας αυτοματισμός βαγονέτου που κινείται αρχικά από αριστερά προς τα δεξιά (MSB => LSB) και στη συνέχεια αντίστροφα συνεχώς, με την προϋπόθεση ότι είναι ON ο διακόπτης MSB της θύρας εισόδου 20H. Αν ο διακόπτης είναι OFF το βαγονέτο κινείται αντίστροφα. Το βαγονέτο φαίνεται στη μορφή ενός led (θύρα εξόδου 30H) και η κίνησή του γίνεται κατά μία θέση κάθε 0,5 sec. Το βαγονέτο ξεκινάει από την μία άκρη και θα καταλήγει στην άλλη όπου αναστρέφεται η κίνηση χωρίς να κάνει πρόσθετη στάση (συνολικά στα άκρα παραμένει 0,5 sec). Η κίνηση του βαγονέτου ελέγχεται από τον LSB διακόπτη της θύρας εισόδου 20H. Όταν αυτό είναι ON το βαγονέτο κινείται, όταν είναι OFF σταματάει στη θέση που είναι και “θυμάται” την κατεύθυνσή του όταν ο LSB διακόπτης επανέλθει στη θέση ON.

- Εκδοχή α) έλεγχος MSB μόνο στην αρχή:

```

START:
IN 10H
LXI B,01F4H ; Εκχωρεί στον διπλό καταχωρητή BC την τιμή (01F4)16 =(500)10
ms.
MVI E,80H ; Αρχικοποιείται στο δυαδικό 8.
MVI L,01H ; Αρχικοποιείται στο δυαδικό 1.

MAIN:
LDA 2000H
MOV D,A
RAR
JNC MAIN ; Έλεγχος για το LSB
MOV A,D
RAL ; Περιστρέφει τα bits του A αριστερά και αποθηκεύει στο CY το MSB
JC RIGHT ; Αν το MSB είναι 1 τότε κάνει δεξιά ολίσθηση

LEFT: ; Αλλιώς αριστερή
LDA 2000H
RAR
JNC LEFT ; Έλεγχος για το LSB
MOV A,L
CMA ; Συμπληρώνω το A ως προς 1 καθώς τα leds είναι αντίστροφης
λογικής
STA 3000H ; Οδηγώ στην έξοδο το A για να ανάψει το καταλληλο led
CMA ; Ξανασυμπληρώνω για να πάρω πάλι το A της εισόδου
RLC ; Κάνω περιστροφή αριστερά για να ορίσω το επόμενο led που θα
ανάψει

```

```

MOV L,A
CPI 01H      ; Συγκρίνω με το 1 που σημαίνει ότι έχει φτάσει στο άκρο οπότε
αλλά η φορά
JZ RIGHT
CALL DELB
JMP LEFT

RIGHT:      ; Ομοίως με πριν
LDA 2000H
RAR
JNC RIGHT   ; Έλεγχος για το LSB
MOV A,E
CMA
STA 3000H
CMA
RRC      ; Αυτή τη φορά περιστρέφομαι δεξιά για να ορίσω το επόμενο led
που ανάβει
MOV E,A
CPI 80H     ; Συγκρίνω με το 8 που σημαίνει ότι έχει φτάσει στο άκρο οπότε
αλλά η φορά
JZ LEFT
CALL DELB
JMP RIGHT
END

```

- Εκδοχή β) έλεγχος MSB καθόλη τη διάρκεια της κίνησης με χρήση διαγράμματος κατάστασης (βλ. τέλος):

```

;Αρχική κατάσταση

MVI A,80H
CMA
STA 3000H      ;Εμφάνισε το αρχικό αποτέλεσμα (Βαγονέτο τέρμα αριστερά)

CMA

MOV E,A        ;E,προσωρινός καταχωρητής

LXI B,01F4H    ;500msec για DELB

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
GO_RIGHT_MSB_IS_ON:      ;Κατάσταση 01

LDA 2000H      ;Διάβασμα LSB διακόπτη
ANI 01H        ;Όσο είναι OFF
CPI 01H        ;Ξαναδιάβασε διακόπτες
JNZ GO_RIGHT_MSB_IS_ON

MOV A,E        ;Ενημέρωση καταχωρητή A
RRC            ;Δεξιά ολίσθηση
CALL DELB
CMA
STA 3000H      ;Εμφάνισης αποτελέσματος
CMA

MOV E,A        ;Αποθήκευσε στον E την κατάσταση του A

READ_MSB:
LDA 2000H      ;Διάβασε την κατάσταση των διακοπών
ANI 80H
ADD E          ;Πρόσθεσε την κατάσταση του A
ANI 81H        ;Απομόνωσε πρώτο και τελευταίο ψηφίο

```

```

CPI 00H                ;Οχι άκρη, διακόπτης OFF
JZ GO_LEFT_MSB_IS_OFF ;κατάσταση 10

CPI 81H                ;Ακρη, διακόπτης ON
JZ GO_LEFT_MSB_IS_ON  ;κατάσταση 11

CPI 01H                ;Ακρη, διακόπτης OFF
JZ READ_MSB           ;κατάσταση STOP
                        ;Οχι ακρη διακόπτης, διακόπτης ON
JMP GO_RIGHT_MSB_IS_ON

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
GO_LEFT_MSB_IS_ON:    ;Κατάσταση 11

LDA 2000H              ;Διάβασμα LSB διακόπτη
ANI 01H               ;Όσο είναι OFF
CPI 01H               ;Ξαναδιάβασε διακόπτες
JNZ GO_LEFT_MSB_IS_ON

MOV A,E               ;Ενημέρωση καταχωρητή A

RLC                  ;Αριστερή ολίσθηση
CALL DELB
CMA
STA 3000H             ;Εμφάνισης αποτελέσματος
CMA

MOV E,A              ;Αποθήκευσε στον E την κατάσταση του A

READ_MSB_2:
LDA 2000H             ;Διάβασε την κατάσταση των διακοπών
ANI 80H
RLC
ANI 01H              ;Ελενξε αν έφτασε στην άκρη
ADD E                ;Πρόσθεσε την κατάσταση του A

ANI 81H              ;Απομόνωσε πρώτο και τελευταίο ψηφίο

CPI 00H              ;Οχι άκρη, διακόπτης OFF
JZ GO_RIGHT_MSB_IS_OFF ;Κατάσταση 00

CPI 81H              ;Ακρη, διακόπτης ON
JZ GO_RIGHT_MSB_IS_ON ;Κατάσταση 01

CPI 80H              ;Ακρη, διακόπτης OFF
JZ READ_MSB_2        ;Κατάσταση STOP
                        ;Οχι Ακρη, διακόπτης ON
JMP GO_LEFT_MSB_IS_ON ;Κατάσταση 11

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
GO_LEFT_MSB_IS_OFF:  ;Κατάσταση 10

LDA 2000H             ;Διάβασμα LSB διακόπτη
ANI 01H               ;Όσο είναι OFF
CPI 01H               ;Ξαναδιάβασε διακόπτες
JNZ GO_LEFT_MSB_IS_OFF

MOV A,E              ;Ενημέρωση καταχωρητή A

RLC                  ;Αριστερή ολίσθηση

CALL DELB
CMA
STA 3000H            ;Εμφάνισης αποτελέσματος

```



```

CMA

MOV E,A                ;Σώσε τη θέση του βαγονέτου

READ_MSB_3:
LDA 2000H              ;Διάβασε την κατάσταση των διακοπών
ANI 80H
RLC
ADD E                  ;Προστίθεται η πληροφορία για θέση
ANI 81H                ;Απομόνωσε πρώτο και τελευταίο ψηφίο

CPI 80H                ;Άκρη, διακόπτης OFF
JZ GO_RIGHT_MSB_IS_OFF ;Κατάσταση 00

CPI 81H                ;Άκρη, διακόπτης ON
JZ READ_MSB_3          ;Κατάσταση STOP

CPI 01H                ;Όχι άκρη, διακόπτης ON
JZ GO_RIGHT_MSB_IS_ON  ;Κατάσταση 01

JMP GO_LEFT_MSB_IS_OFF ;Όχι άκρη, διακόπτης OFF
                        ;κατάσταση 10

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
GO_RIGHT_MSB_IS_OFF:   ;Κατάσταση 00

LDA 2000H              ;Διάβασμα LSB διακόπτη
ANI 01H                ;Όσο είναι OFF
CPI 01H                ;Ξαναδιάβασε διακόπτες
JNZ GO_RIGHT_MSB_IS_OFF

MOV A,E                ;Ενημέρωση καταχωρητή A
RRC                    ;Σώσε τη θέση του βαγονέτου

CALL DELB
CMA
STA 3000H              ;Εμφάνισης αποτελέσματος
CMA

MOV E,A                ;Αποθήκευσε στον E την κατάσταση του A

READ_MSB_4:
LDA 2000H              ;Διάβασε την κατάσταση των διακοπών
ANI 80H
ADD E                  ;Πρόσθεσε την κατάσταση του A
ANI 81H                ;Απομόνωσε πρώτο και τελευταίο ψηφίο

CPI 80H                ;Όχι άκρη, διακόπτης ON
JZ GO_LEFT_MSB_IS_ON   ;Κατάσταση 11

CPI 81H                ;Άκρη, διακόπτης ON
JZ READ_MSB_4          ;Κατάσταση STOP

CPI 01H                ;Άκρη, διακόπτης OFF
JZ GO_LEFT_MSB_IS_OFF  ;Κατάσταση 10

JMP GO_RIGHT_MSB_IS_OFF ;Όχι άκρη, διακόπτης OFF
                        ;Κατάσταση 00

END

```

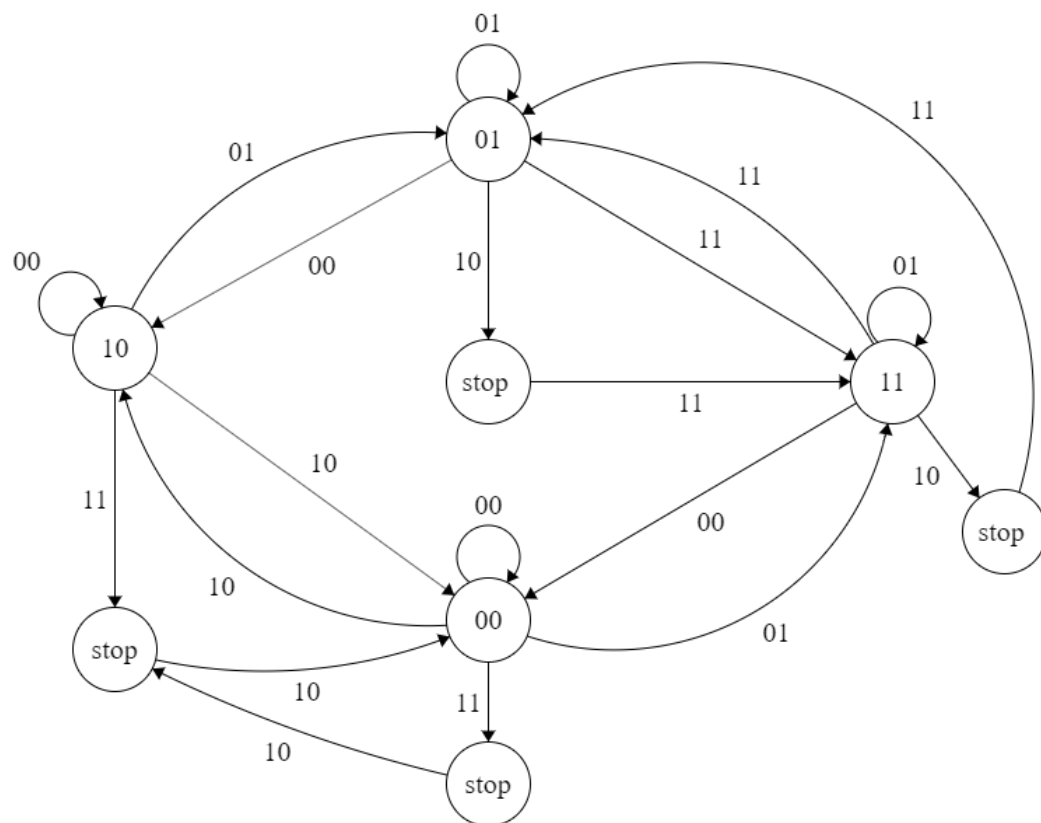
Διάγραμμα καταστάσεων :

Σύνολο Καταστάσεων :

Κατάσταση	Ερμηνεία
00	Δεξιά κίνηση, διακόπτης OFF
01	Δεξιά κίνηση, διακόπτης ON
10	Αριστερή κίνηση, διακόπτης OFF
11	Αριστερή κίνηση, διακόπτης ON

Σύνολο Εισόδων :

Είσοδος	Ερμηνεία
00	Όχι άκρη ,διακόπτης γίνεται OFF
01	Όχι άκρη ,διακόπτης γίνεται ON
10	Άκρη, διακόπτης γίνεται OFF
11	Άκρη ,διακόπτης γίνεται ON



Τέλος εργαστηριακής Αναφοράς