**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

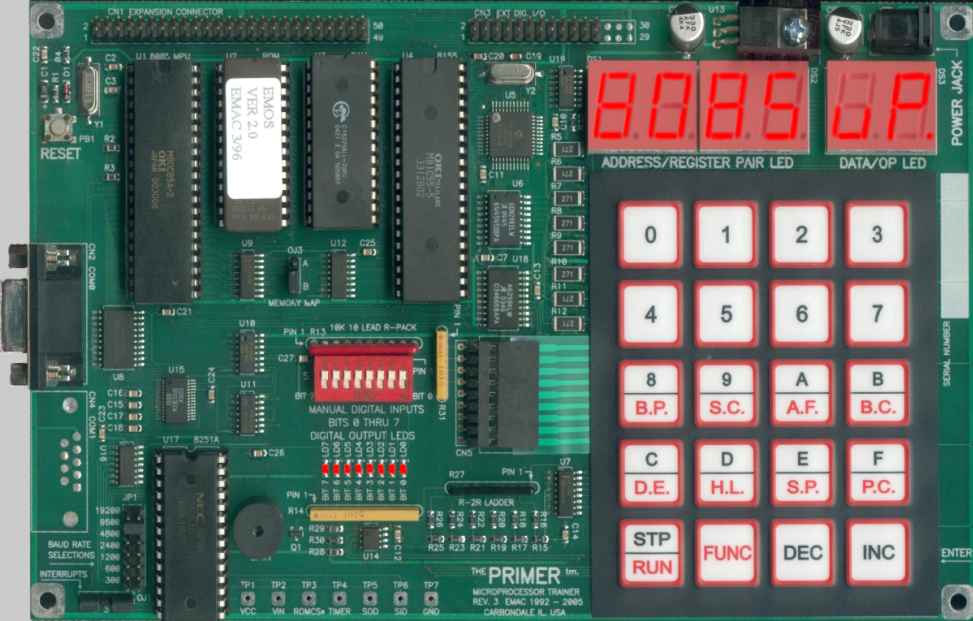
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

&

ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Συστήματα Μικροϋπολογιστών

*4η Σειρά Ασκήσεων*



*6Ο Εξάμηνο*

*ΡΟΗ Υ*

*Αθανασίου Νικόλαος*

*ΑΜ 03112074*

*Σταυρακάκης Δημήτριος*

*ΑΜ 03112017*

# Άσκηση 1η

Στην παρούσα άσκηση υλοποιείται ένα πρόγραμμα που ελέγχει τα φώτα ενός χώρου, με τη διαδικασία που ορίζεται από την εκφώνηση. Έχουμε λάβει υπόψη πως με το πάτημα του πλήκτρου INTRPT στο microlab simulator πραγματοποιούνται στην ουσία δύο διακοπές, δηλαδή δύο κλήσεις της συνάρτησης INTR\_ROUTINE. Παρακάτω παρατίθεται ο αντίστοιχος κώδικας:

LXI B,0014H ; για την καθυστέρηση των 20ms

IN 10H

MVI A,0DH ; φόρτωση στον Α του 00001101

SIM ;

EI ; Ενεργοποίηση διακοπών

MVI A,00H ;

ANAMONH:

LXI H,0910H ; Έλεγχος αν ο Α είναι 2 (διπλή κλήση της RST ;6.5)

MOV A,M ; με ένα πάτημα του ΙΝΤΕRRUPT

CPI 02H

JZ INTR

JMP ANAMONH

INTR:

LXI H,0910H ; μηδενισμός

MVI M,00H

LXI H,0900H ; 60 = τα δύο αριστερότερα του 7 segment

MVI M,10H ; σβηστά όλα τα άλλα ψηφία

INX H

MVI M,10H

INX H

MVI M,10H

INX H

MVI M,10H

INX H

MVI M,00H

INX H

MVI M,06H

MVI H,7AH ; ορισμός 120 επαναλήψεων

MVI A,FFH

CMA

STA 3000H ;αναμμα των leds όταν πατηθεί το INTRPT

DIS\_PREPARE:

MVI L,19H ; Αρχικοποίηση του L σε 25( 25\*20ms = 500ms delay)

DIS\_CHECK: DI ; απενεργοποίηση μάσκας διακοπών

PUSH H ;κραταω τις τιμες των καταχωρητών

PUSH B

MOV B,A

LXI H,0910H

MOV A,M

CPI 01H ;σύγκριση του Α με το 1 για έλεγχο των διακοπών

MOV A,B ;αν πατηθεί το INTRPT γίνεται μετάβαση στη anamonh

POP B ;ανακτώ τις τιμές των καταχωρητών

POP H

EI

JZ ANAMONH

CALL DELB ; καθυστέρηση των 20ms

JMP EMFANISH ; απεικόνιση υπολοιπόμενου χρόνου

FIX\_DELAY: DCR L

JNZ DIS\_CHECK

INR D

MOV E,A

MOV A,D ;( 500ms + 500ms = 1sec )

CPI 02H

JZ MAKE

KEEP\_ON:

MOV A,E

DCR H ; Συνέχισε να εμφανίζεις, εδώ μειώνω τις ;επαναλήψεις

JNZ DIS\_PREPARE

JMP ANAMONH ; αν περάσει ένα λεπτό χωρίς INTRPT πήγαινε στην ;αναμονή και δες αν ξαναέχω interrupt

EMFANISH:

DI

PUSH B ;κραταω τις τιμες των καταχωρητών

PUSH D ; BC DE HL

PUSH H

LXI D,0900H ;φόρτωση της διεύθυνσης με τους αριθμούς

MOV B,A

CALL STDM ;εμφάνιση των ψηφίων στον 7 segment

CALL DCD

MOV A,B

POP H ;ανακτώ τις τιμές των καταχωρητών

POP D ;ως είχαν πριν (BC,DE,HL)

POP B

EI

JMP FIX\_DELAY

MAKE:

DI

PUSH B

PUSH H

LXI H,0904H ; φόρτωση και μείωση στο δεξίοτερο ψηφίο που

DCR M ; εμφανίζεται στο 7SEGMENT DISPLAY

MOV B,A

MOV A,M ;

ELEGXOS\_DEKADWN:

CPI FFH ;Έλεγχος αν έφτασα στο 0 για να μειώσω τις δεκάδες

JZ ZERO

NEXT: MOV A,B

POP H

POP B

EI

MVI D,00H

JMP KEEP\_ON

ZERO:

MVI M,09H ;αν έφτασα στο 0 φόρτωση στις μονάδες το 9 και στις ;δεκάδες το δεκάδες-1

INX H

DCR M

JMP NEXT

INTR\_ROUTINE: ; ρουτίνας διακοπής τύπου

PUSH H ; RST 6.5

LXI H,0910H

INR M

POP H

EI

RET ;επιστροφή στο κυρίως πρόγραμμα

FINISH:

END

## Άσκηση 2η

Στην άσκηση αυτή, καλούμαστε να υλοποιήσουμε σε γλώσσα assembly του 8085 ένα πρόγραμμα που εκτελεί την παρακάτω λειτουργία: Όταν προκαλείται διακοπή τύπου RST 6.5 το πρόγραμμα διαβάζει δύο διαδοχικά ψηφία ενός δεκαεξαδικού αριθμού από το πληκτρολόγιο και τα απεικονίζει στα δύο δεξιότερα του 7 segment display. Επιπλέον, ανιχνεύει το πεδίο τιμών στο οποίο ανήκει ο αριθμός που πατήθηκε και ανάβει κάποιο ενδεικτικό LED. Συνολικά, έχουμε τρία πεδία τιμών: [00H,Κ1], (Κ1,Κ2] και (Κ2,FFH]. Οι τιμές τον K1, K2 περιέχονται στους καταχωρητές B και C αντίστοιχα. Σημειώνουμε, ότι το πρώτο πεδίο αντιστοιχεί στο 3ο  LED, το δεύτερο στο 2ο και το τρίτο στο 1ο LED. Πριν παραθέσουμε τον κώδικα του προγράμματος, αξίζει να αναφερθούμε σε ένα λεπτό σημείο. Όταν στο microlab πατάμε το κουμπί INTRPT, προκαλούνται δυο διακοπές και όχι μια, όπως αναφέραμε και στην προηγούμενη άσκηση. Για να λύσουμε ο πρόβλημα αυτό δουλέψαμε ως εξής: αρχικά ο καταχωρητής HL δείχνει τη θέση μνήμης που θα αποθηκευτεί ένα από τα δύο ψηφία που θέλουμε να διαβάσουμε. Κάθε φορά που καλείται η ρουτίνα INTR\_ROUTINE (δύο φορές για κάθε πάτημα του INTRPT) διαβάζουμε το ψηφίο, το αποθηκεύουμε στη διεύθυνση που δείχνει ο HL και συμπληρώνουμε το τελευταίο bit του HL. Με αυτόν τον τρόπο, για κάθε ζευγάρι INTR\_ROUTINE που θα εκτελεστεί τα δύο ψηφία που θα διαβαστούν αποθηκεύονται πάντα σε δύο κατάλληλες διαδοχικές θέσεις μνήμης (09A1H και 09A0H στο πρόγραμμά μας επομένως με την πρώτη συμπλήρωση του τελευταίου bit πάμε στη μία και με τη δεύτερη, επειδή η INTR\_ROUTINE καλείται 2 φορές, πάμε στην άλλη).Στη συνέχεια παρατίθεται ο κώδικας Assembly για την υλοποίηση των παραπάνω:

IN 10H ;αποκτούμε πρόσβαση σε όλη τη μνήμη

;επειδή θα έχουμε αριθμούς από το 0-255 θα χωρίσουμε το διάστημα αυτό

;σε 3 περίπου ίσα διαστήματα, γι’ αυτό επιλέγουμε Κ1=256/3=85.3

;δηλαδή θέτουμε (Β)=56Η=86 DEC

;και (C)=ACH=172 DEC

MVI B,56H ;αρχικοποιούμε τον B (όριο Κ1)

MVI C,ACH ;αρχικοποιούμε τον C (όριο Κ2)

MVI A,0DH ;ενεργοποιούμε τις διακοπές μέσω μάσκας

SIM

EI ;ενεργοποίηση διακοπών (Enable Interrupts)

MVI A,10H ;αποθηκεύουμε τον κενό χαρακτήρα, στις θέσεις

STA 09A5H ;μνήμης που αντιστοιχούν στα 7-segment display

STA 09A4H ;που δεν θα χρησιμοποιήσουμε, δηλαδή στις

STA 09A3H ;4 αριστερότερες θέσεις

STA 09A2H

LXI H,09A1H ;η διεύθυνση που θα απεικονιστεί στο 2ο από δεξία

;ψηφίο του 7 SEGMENT display

INR B ;μεγαλώνουμε τα όρια κατά ένα έτσι ώστε να λάβουμε

INR C ;την περίπτωση του κλειστού διαστήματος που ζητείται

INITIAL:

LDA 09A0H ;εδώ κατασκευάζουμε από τα δύο δεκαεξαδικά ψηφία

ANI 0FH ;τον συνολικό αριθμό με σκοπό να τον κατατάξουμε σε

MOV E,A ;κάποιο πεδίο

LDA 09A1H

RLC

RLC

RLC

RLC

ANI F0H ;κάνω and με τον αριθμό 11110000 για να κρατήσω τα 4 ;MSBits

ORA E ;δημιουργώ τον αριθμό συνολικά

CMP B ;ελέγχουμε εάν αν ανήκει στο πεδίο [0,Κ1]

JNC FIELD1 ;αν δεν ανήκει τσεκάρουμε το επόμενο πεδίο

MVI A,FBH ;μεταφέρω στον Α την τιμή 11111011 για το άναμμα του 3ου ;LSB

JMP CHOSEN

FIELD1: ;ελέγχουμε εάν αν ανήκει στο πεδίο (Κ1,Κ2]

CMP C

JNC FIELD2

MVI A,FDH ;μεταφέρω στον Α την τιμή 11111101 για το άναμμα του 2ου ;LSB

JMP CHOSEN

FIELD2: ;ανήκει σίγουρα στο πεδίο (K2,FFH]

MVI A,FEH ;μεταφέρω στον Α την τιμή 11111110 για το άναμμα του 1ου ;LSB

CHOSEN:

STA 3000H ;ανάβουμε το LED του πεδίου που ανήκει ο αριθμός

PUSH B ;κάνουμε Push στη στοίβα τους διπλούς καταχωρητές

PUSH H ;καθώς αυτοί επηρεάζονται από την STDM και DCD

LXI D,09A0H ;εμφανίζουμε τα ψηφία του αριθμού στα 7 segment display

CALL STDM

CALL DCD

POP H ;επαναφέρουμε τις τιμές των καταχωρητών

POP B

JMP INITIAL ;επαναλαμβάνουμε την όλη διαδικασία

INTR\_ROUTINE: ;σε περίπτωση που γίνει διακοπή

PUSH PSW ;αποθηκεύουμε τον καταχωρητή A και τα FLAGS στη στοίβα

MVI A,00H ;ανάβουμε όλα τα LEDs ως ένδειξη αναγνώρισης διακοπής

STA 3000H

CALL KIND ;διαβάζουμε τον κωδικό του ενός πλήκτρου από το

;πληκτρολόγιο

MOV M,A ;και τον αποθηκεύουμε στη διεύθυνση HL

MOV A,L ;συμπληρώνουμε το LSB του καταχωρητή HL

XRI 01H ;για να διαβαστεί το δεύτερο ψηφίο κάνουμε xor με τον ;αριθμό 00000001 για να συμπληρωθεί το τελευταίο ψηφίο ;του L για την αποφυγή του προβλήματος που περιγράφεται ;και αρχικά

MOV L,A ;υπενθυμίζουμε ότι το κουμπί INTRPT προκαλεί δύο διακοπές

POP PSW ;επαναφέρουμε τις τιμές των flags και του accumulator

EI ;ενεργοποιούμε τις διακοπές

RET ;επαναφορά

END

# Άσκηση 3η

Στην αρχή, η τιμή του Program Counter (PC) ισούται με 2000Η όπως μας δίνεται στην εκφώνηση. Επίσης, ο δείκτης της στοίβας (Stack Pointer=SP) έχει την τιμή 4000Η. Επειδή η διακοπή πραγματοποιείται στη μέση της εντολής CALL 3000Η, δηλαδή έχει ήδη αρχίσει να εκτελείται η συγκεκριμένη εντολή, η διακοπή στην ουσία θα αναγνωριστεί από το σύστημα αφού ολοκληρωθεί και ο τελευταίος κύκλος της εντολής CALL. Επομένως, ο PC θα πάρει πρώτα την τιμή 3000Η όπως του ορίζει η εντολή CALL 3000Η. Αμέσως μετά θα αναγνωριστεί η διακοπή, θα απενεργοποιηθούν αυτόματα οι διακοπές και θα αποθηκευτεί η τιμή του PC στη στοίβα. Η τιμή του PC αποτελείται από 16 Bits επομένως για να αποθηκευτεί θέλει 2 θέσεις της στοίβας. Άρα οι μεταβολές που θα πραγματοποιηθούν στη στοίβα είναι οι ακόλουθες:

* Στη θέση (SP)-1 θα αποθηκευτεί η τιμή 30Η (τα 8 σημαντικότερα bits του PC)
* Στη θέση (SP)-2 θα αποθηκευτεί η τιμή 00Η (τα 8 λιγότερο σημαντικά bits του PC)
* Η τιμή του δείκτη στοίβας (SP) θα μειωθεί κατά δύο για να δείχνει την επόμενη θέση που τυχόν θα εισαχθεί το επόμενο στοιχείο στη στοίβα

Πιο μαθηματικοποιημένα θα πραγματοποιηθούν τα ακόλουθα:

* ((SP)-1) 🡨 30H (PCHigh)
* ((SP)-2 ) 🡨 00H (PCLow)
* (SP) = (SP)-2 (δηλαδή (SP)=3FFEH)

Τέλος, ο PC θα πάρει την διεύθυνση της ρουτίνας RST 6.5 (0034H) και θα εκτελεστεί η ρουτίνα εξυπηρέτησης της διακοπής.

# Άσκηση 4η

Σε αυτήν την άσκηση, ζητείται να γράψουμε ένα πρόγραμμα σε Assembly 8085 το οποίο θα λαμβάνει 32 δεδομένα των 8 Bit από μία συσκευή και καθένα από αυτά να το μεταφέρει με 2 βήματα, πρώτα τα 4 LSB και έπειτα τα 4 MSB, μέσω των Bits (X0-X3) της θύρας PORT\_IN. Τα υπόλοιπα 4 MSBits δε χρησιμοποιούνται. Στο σύστημα αυτό, μετά από κάθε αποστολή, η συσκευή προκαλεί διακοπή τύπου RST 5.5. Τέλος, ζητείται να υπολογιστεί ο μέσος όρος των 32 δεδομένων με ακρίβεια των 8 Bit.

MVI A,0EH ; μάσκα διακοπών = 00001110

SIM

LXI H,0000H ; (Η)(L)<-0

MVI C,40H ; μεταφέρω στον C την τιμή 64 για τις επαναλήψεις

EI ;ενεργοποίηση διακοπών

FIRST: ; αναμονή μέχρι C=0

MVI A,C ; εκτέλεση 64 φορές

CPI 00H

JNZ FIRST

DI ; απενεργοποίηση διακοπών

DAD H ; ολίσθηση 3 θέσεις αριστερά

DAD H

DAD H

HLT

RST6.5:

PUSH PSW ; κρατάω το περιεχόμενο του Α και των flags

MOV A,C

ANI 01H

CPI 00H ; έλεγχος αν πρόκειται για LSB ή MSB

JZ SECOND

IN PORT\_IN ; ή IN 80H

ANI 0FH ; Αν διαβάζεται LSB μηδενίζονται τα 4 MSB

MVI D,00H

MOV E,A

JMP FINISH

SECOND:

IN PORT\_IN ; ή IN 80H

RRC ; δεδομένα στα 4 ΜSΒ

RRC

RRC

RRC

ANI F0H ; μηδενισμός των 4 LSB

MVI D,00H

MOV E,A

FINISH:

DAD D ; sum (HL) += D

DCR C

POP PSW ;επαναφορά του Α και των flags

EI ; ενεργοποίηση διακοπών

RET

Σημείωση: Για το μέσο όρο των 32 δεδομένων, κάνουμε 3 ολισθήσεις προς τα αριστερά του HL. Αυτό έχει ως αποτέλεσμα το ακέραιο μέρος του Μ.Ο. να βρίσκεται στον καταχωρητή Η και το κλασματικό μέρος στον καταχωρητή L.

Παράδειγμα:

Αν τα δεδομένα μας έχουνε άθροισμα στον HL τον αριθμό

00000000 – 10010100 = Η – L =14810

Με την κατάλληλη ολίσθηση γίνεται:

00000100 – 10100000 = Η – L

Δηλαδή 4 είναι το ακέραιο μέρος του Μ.Ο. και το κλασματικό μέρος είναι :

2-1+2-3=0,5+0,125=0,625

Που είναι σωστό αφού : 148/32= 4,625

Έτσι υπολογίζεται ο μέσος όρος που ζητείται.