

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ
ΠΟΛΥΤΕΧΝΕΙΟ

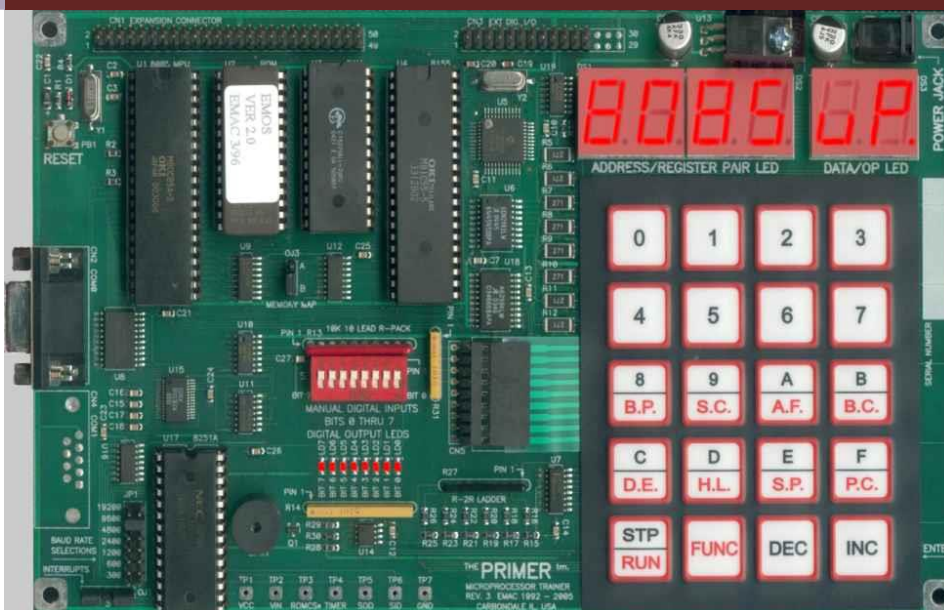
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ

&

ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

Εργαστήριο
Μικροϋπολογιστών

1^η Σειρά Ασκήσεων



7^ο Εξάμηνο

ΡΟΗ Υ

Αθανασίου
Νικόλαος

AM 03112074

Βαβουλιώτης
Γεώργιος

AM 03112083

Γιαννούλας
Βασίλειος

AM 03112117

Άσκηση 1ii

Σκοπός της άσκησης: Σκοπός της άσκησης είναι η υλοποίηση της λειτουργίας ενός χρονομέτρου συνεχούς λειτουργίας που μετράει ως το 15 με την αντίστοιχη μέτρηση να απεικονίζεται στα leds συνεχόμενα.

Επεξήγηση κώδικα: Αρχικά καλείται η συνάρτηση BEEP, η οποία παράγει ένα χαρακτηριστικό ήχο ο οποίος δηλώνει την έναρξη της εκτέλεσης του προγράμματος. Στην συνέχεια φορτώνω στον διπλό καταχωρητή BC τη τιμή 1000(στο δεκαδικό σύστημα) έτσι ώστε να ορίσω τη διάρκεια της καθυστέρησης που θα προκληθεί από την κλήση της συνάρτησης DELB, αφού κατά τα γνωστά η συνάρτηση DELB εισάγει καθυστέρηση ίση με $((B)(C)) * 1ms$. Έπειτα αρχικοποιείται ένας μετρητής στη τιμή 16 ο οποίος φροντίζει ώστε όταν φτάσουμε στον αριθμό 15,δηλαδή όταν ο μετρητής μηδενστεί, να αρχίζει από την αρχή η μέτρηση και να παίρνει εκ νέου την τιμή 16. Η απεικόνιση στα leds οφείλεται στον καταχωρητή A, ο οποίος αρχικά έχει την τιμή FFH, αλλά λόγω της αρνητικής λογικής των leds έχει ως αποτέλεσμα η εντολή STA 3000H να σβήνει όλα τα leds. Η απεικόνιση της σωστής μέτρητης στα leds επιτυγχάνεται κάθε φορά με μείωση του A κατά 1 και προφανώς ανανέωση αυτού στη τιμή FFH όταν η μέτρηση φτάσει στον αριθμό 15.

Ο κώδικας σε assembly 8085 φαίνεται παρακάτω :

```
START:
    CALL BEEP
    MVI B,03H ;bazw ston BC to 1000 decimal
    MVI C,E8H

LABEL1:
    MVI A,FFH
    MVI E,10H ;E=16

LOOP1:
    STA 3000H
    CALL DELB
    DCR A
    DCR E
    JNZ LOOP1
    JMP LABEL1

END
```

Άσκηση 2i

Σκοπός της άσκησης: Το παρακάτω πρόγραμμα αναβοσβήνει συνεχώς όλα τα leds, αλλά με μεταβαλλόμενη καθυστέρηση για το άναμμα και το σβήσιμο, η οποία καθορίζεται από τους 4 αριστερότερους διακόπτες για το άναμμα και τους 4 δεξιότερους για το σβήσιμο.

Επεξήγηση κώδικα : Από την εκφώνηση της άσκησης έχω ότι η ελάχιστη καθυστέρηση είναι 200ms(=2*100ms) και η μέγιστη είναι 1700ms(=15*100 + 200), δηλαδή το διάστημα 0-15 των διακοπών αντιστοιχεί στο 200-1700ms(ουσιαστικά γίνεται η πράξη 200+μέτρηση*100).

Υπο αυτό το πρίσμα, φορτώνω στον BC το 100 στο δεκαδικό σύστημα ώστε να υλοποιήσω την παραπάνω πράξη καθυστέρησης όταν θα καλώ την συνάρτηση DELB.

Η ιδέα του παρακάτω προγράμματος είναι ότι είτε ασχολούμαι με το άναμμα είτε με το σβήσιμο, αρχικά θα πρέπει να γίνει καθυστέρηση 200ms, η οποία επιτυγχάνεται με δυο συνεχόμενες κλήσεις της συνάρτησης DELB.

Στη συνέχεια αν εξετάζω το άναμμα τότε ασχολούμαι με τα MSB των διακοπών και ανάλογα με την τιμή που έχουν καλώ τόσες φορές την συνάρτηση DELB. Το ίδιο κάνω ακριβώς κάνω και για το σβήσιμο.

Ο κώδικας σε assembly 8085 φαίνεται παρακάτω :

```
START:
    MVI B,00H           ;bazw ston BC to 100 decimal gia thn ;ka8usterhsh
    MVI C,64H
    MVI A,00H           ;anabw ta leds
    STA 3000H           ;ta apeikonizw
    CALL DELB           ;ka8usterhsh 2*100ms
    CALL DELB
    LDA 2000H           ;diabasma apo diakoptes
    ANI F0H             ;krataw ta MSB
    RLC                 ;metafora sta LSB
    RLC
    RLC
    RLC
    MOV E,A             ;bazw sto E ton ari8mo pou apeikonizoun ta MSB
    CPI 00H             ;an einai 0 tote paw sto sbhsimo
    JZ LABEL2           ;alliws anabw ta leds gia oso leei o E

LOOP1:
    CALL DELB           ;trexei gia oso lene ta MSB
    DCR E
    JNZ LOOP1

LABEL2:
    MVI A,FFH           ;sbhnw ta leds
    STA 3000H           ;ta apeikonizw
    CALL DELB           ;ka8usterhsh 2*100ms
    CALL DELB
    LDA 2000H           ;diabasma apo diakoptes
    ANI 0FH             ;krataw ta LSB
    MOV E,A             ;bazw sto E ton ari8mo pou apeikonizoun ta LSB
    CPI 00H             ;an einai 0 tote paw sthn arxh
    JZ START           ;alliws sbhnw ta leds gia oso leei o E

LOOP2:
    CALL DELB           ;trexei gia oso lene ta LSB
    DCR E
    JNZ LOOP2

    JMP START

END
```

Άσκηση 2iia

Σκοπός της άσκησης: Σκοπός αυτού του προγράμματος είναι η υλοποίηση ενός μετρητή που θα εκτελεί μέτρηση συνεχώς από το 0 έως το 15 στα 4 LSB των LEDs και θα μετράει το πλήθος των διακοπών που έχουν προκληθεί και θα το απεικονίζει στα 4 MSB των LEDs, εφόσον το MSB των διακοπών είναι on.

Επεξήγηση κώδικα: Η λογική του παρακάτω κώδικα είναι ότι ενεργοποιώ από την αρχή του κώδικα της διακοπής (είτε το MSB είναι on είτε όχι) και μέσα στην ρουτίνα εξυπηρέτησης της διακοπής τσεκάρω το MSB και ανάλογα με αυτό αυξάνω ή όχι τον μετρητή των διακοπών. Για να αποφύγω πολλαπλές διακοπές που είναι πιθανό να συμβούν λόγω ευαισθησίας της εισόδου διακοπής χρησιμοποιώ στην ρουτίνα εξυπηρέτησης τον αλγόριθμο του εργαστηριακού οδηγού για να αποφύγω τη μέτρηση παραπάνω διακοπών.

Για να αυξήσουμε τον μετρητή των διακοπών απομονώνουμε τα 4 MSB του καταχωρητή D, οποίος περιέχει την τελευταία τιμή που εμφανίστηκε, τα μεταφέρουμε στα LSB, αυξάνουμε κατά 1, τα μεταφέρουμε πάλι στα MSB και τα ενώνουμε με τα LSB του κύριου μετρητή.

Τώρα για να μπορέσουμε να αυξήσουμε τον μετρητή στα 4 LSB, χωρίς όμως να χάσω το πλήθος των διακοπών αποθηκεύουμε πάντα το τελευταίο αποτέλεσμα που έχει εμφανιστεί στα leds στον καταχωρητή D και σε κάθε επανάληψη απομονώνουμε τα 4 LSB του D, τα αυξάνουμε κατά 1 (εκτός εάν είναι 15, όπου θα τα μηδενίσουμε), και στη συνέχεια τα συνενώνουμε με τα 4 MSB που περιέχουν το πλήθος των διακοπών. Πλέον το αποτέλεσμα που έχω είναι πάντα σωστό.

Ο κώδικας σε assembly 8085 φαίνεται παρακάτω :

```
START:
    MVI A,0DH                ;Energopoihsh tw n diakopwn
    SIM
    EI
    MVI D,FFH                ;Arxikopoiisi eksodou
    MOV A,D
    STA 3000H
    LXI B,0064H
    DI
    CALL DELB
    EI

COUNT:
    MOV A,D
    ANI 0FH                  ;4 LSB=metritis
    DCR A                     ;meiwse ena (logw tis sumplirwsis)
    MOV E,A
    MOV A,D
    ANI F0H                   ;Pare ta 4MSB pou antistoixoun ston metriti
                                ;diakopwn
    ADD E                     ;Sunenwse me ta 4LSB
    STA 3000H                 ;Emfanise sta LED
    MOV D,A
    DI                         ;Apothikeuesai tin teleutaia eksodo ston D
                                ;Apenergopoiise tis diakopes prin tin
                                ;kathusterisi
    LXI B,0064H               ;Kathusterisi isi me to 1/10 tou sec
    CALL DELB
    EI

CHECK_SWITCHES1:
    MVI A,00H
    CMP E                     ;ean o metritis den exei ftasei sto 15
    JNZ COUNT                 ;epanelave tin diadikasia
    MOV A,D
    ANI F0H
    ADI 0FH                   ;alliws ksekina ton apo tin arxi
    STA 3000H
    DI
    CALL DELB
    EI
```

```

        MOV D,A
        JMP COUNT                ;sunexomenh leitourgia
INTR_ROUTINE:                    ;routina eksupiretisis diakopis
        LXI B,0032H
        CALL DELB                ;kathsterisi 50msec
R_MASK:
        RIM                      ;elekse to 5o bit tis maskas
        ANI 20H                  ;kai sunexise parakataw otan ginei 0
        CPI 00H
        JNZ R_MASK
        LXI B,0032H
        CALL DELB                ;kathusterisi 50msec
        LDA 2000H
        RAL
        JNC LABEL11
        MOV A,D
        ANI F0H
        RRC                      ;metefere ta 4 MSB sta LSB
        RRC
        RRC
        RRC
        CPI 00H                  ;an eisai se pollaplasio tou 16 ksekina
        JZ RESTART              ;tin metrisi apo tin arxi
        DCR A                    ;alliws auksise to plithos tw n diakopwn
                                ;kata 1 (afairw logw apeutheias sumplirwsis
                                ;eksodou)
        RLC                      ;ksanametefere ta sta 4MSB
        RLC
        RLC
        RLC
        MOV E,A
        MOV A,D
        ANI 0FH                  ;kai topothetise ta 4LSB toy metriti
        ADD E
        JMP EXIT                ;emfanise to apotelesma
RESTART:
        MOV E,A
        MOV A,D
        ANI 0FH
        ADI F0H
EXIT:
        STA 3000H
        MOV D,A                  ;krata tin teleutaia egkuri eksodo ston D
LABEL11:
        EI
        RET
END

```

Άσκηση 2iib

Σκοπός της άσκησης: Σκοπός αυτού του προγράμματος είναι η υλοποίηση ενός μετρητή που θα εκτελεί μέτρηση συνεχώς από το 0 έως το 15 στα 4 LSB των LEDs και θα μετράει (εφόσον το MSB των διακοπών είναι ON) το πλήθος των διακοπών που είναι on όταν προκληθεί μια διακοπή και θα το απεικονίζει στα 4 MSB των LEDs.

Επεξήγηση κώδικα: Αρχικά τσεκάρω αν το MSB είναι on και αν είναι τότε ενεργοποιώ τις διακοπές αλλιώς όχι. Στη συνέχεια η διαδικασία που ακολουθείται είναι όμοια με αυτή της άσκησης 2iia με την μόνη διαφορά ότι σ'αυτή την άσκηση υπάρχει ένας καταχωρητής ο οποίος μας πληροφορεί αν οι διακοπές έχουν ενεργοποιηθεί ή όχι.

Η μόνη ουσιαστική αλλαγή εντοπίζεται μέσα στην ρουτίνα εκυπηρέτησης της διακοπής στην οποία πλέον μετράμε το πλήθος των διακοπών που είναι on όταν έχει προκληθεί μια διακοπή. Αυτό το κάνουμε απομονώνοντας τα 4 MSB του καταχωρητή D που περιέχει την τελευταία τιμή που εμφανίσαμε, τα μεταφέρουμε στα LSB, αυξάνουμε κατά 1, μεταφέρουμε πάλι στα msb το πλήθος των ενεργοποιημένων διακοπών, συνενωνούμε με τον κύριο μετρητή και σώζουμε την τελευταία κατάσταση που εμφανίστηκε στον καταχωρητή D.

Ο κώδικας σε assembly 8085 φαίνεται παρακάτω :

```
        MVI D,FFH                ;Arxikopoiisi eksodou
CHECK_SWITCHES:
        LDA 2000H
        RAL                      ;Elekse to MSB
        JC INTER                 ;an einai 1 energopoihse tis diakopes
        MVI L,00H
        DI                      ;alliws mhn to kaneis
        JMP COUNT

INTER:
        MVI L,01H
        MVI A,0DH
        SIM
        EI

COUNT:
        MOV A,D
        ANI 0FH                 ;krataw ta 4 lsb
        DCR A                   ;meiwnw kata 1 logw sumplhrwshs
        MOV E,A
        MOV A,D
        ANI F0H                 ;pairnw ta 4 msb pou antistoixoun ston metrhth
        ADD E                   ;sunenwnw me ta lsb
        STA 3000H
        MOV D,A                 ;apo8hkeush teleutaias eksodou ston D
        LXI B,0AFFH             ;ka8usterhsh ish me 1/10 tou sec
        DI
        CALL DELB
        MVI A,00H
        CMP L                   ;an den exoun hdh energopoih8ei mhn to kaneis
        JZ CHECK_SWITCHES1
        EI

CHECK_SWITCHES1:
        MVI A,00H
        CMP E                   ;an o counter den exei ftasei sto 15
        JNZ CHECK_SWITCHES     ;epanelabe to loop
        MOV A,D
        ANI F0H
        ADI 0FH                 ;edw ksekiname apo thn arxh
        MOV D,A
        JMP CHECK_SWITCHES
```

```

INTR_ROUTINE:
    LDA 2000H          ;Diavase tin eisodo
    MVI B,09H          ;Arxikopoiisi metriti epanalipsis
    MVI L,00H          ;Arxikopoiisi metriti tw n ON diakoptwn
REPEAT:
    RAL                ;des to MSB
    JNC SKIP           ;an den einai 1 min ayksiseis to metriti
    INR L
SKIP:
    DCR B
    JNZ REPEAT         ;epanelave mexri na elegkseis ola ta psifia
                        ;eisodou
    MOV A,L
    RLC                ;metefere to metriti sta MSB
    RLC
    RLC
    RLC
    ANI F0H
    CMA                ;kai anestrepse gia energopoiisi tw n
                        ;katallilwn LEDS
    MOV E,A
    MOV A,D
    ANI 0FH
    ADD E              ;sunenwse me ton metriti tou programmatos
    STA 3000H          ;emfanise sta LEDS
    MOV D,A            ;apothikeuse ston D tin teleutaia
                        ;emfanizomeni katastasi
    RET
END

```

Παρατηρήσεις-Σημαντικές Λεπτομέρειες Υλοποίησης για τις ασκήσεις 2iia,2iib :

- Από τη στιγμή που θα πατηθεί το πλήκτρο διακοπής μέχρι να υπάρξει σταθεροποίηση του bit I6.5 στην τιμή 1 μεσολαβεί κάποιο χρονικό διάστημα (τάξης 50 ms), επειδή οι παλμοί που παράγονται στο σύστημα δεν είναι ιδανικοί και η τάση που θα αποκτήσει το συγκεκριμένο bit θα αυξάνεται σταδιακά, μέχρι να περάσει μια οριακή τιμή και να μπορεί να θεωρηθεί λογικό 1.
- Αν προκληθεί μια διακοπή κατά τη διάρκεια κλήσης μιας DELB, είναι πολύ επικίνδυνο να επηρεαστεί η λειτουργία της DELB και να έχουμε μη επιθυμητό αποτέλεσμα επειδή η ρουτίνα εξυπηρέτησης μπορεί να μεταβάλλει τα περιεχόμενα των καταχωρητών B,C που χρειάζεται η delb. Για το λόγο αυτό πριν τη κλήση της ρουτίνας DELB, απενεργοποιούμε τις διακοπές και τις ενεργοποιούμε εκ νέου με τη λήξη της DELB όπως είναι εμφανές να δει κανείς τον παραπάνω κώδικα.

Εξήγηση λειτουργίας λογικών διαγραμμάτων

- Το **πρώτο λογικό διάγραμμα** περιμένει να περάσει ο παλμός που προκάλεσε τη διακοπή του. Μετά τα 50 ms(διάστημα που χρειάζεται για να αλλάξει το bit της διακοπής) ελέγχει μέχρι να βεβαιωθεί ότι δεν έχει προκληθεί άλλη διακοπή μέσα στην παρούσα διακοπή. Τέλος, εκτελεί το πρόγραμμα διακοπής και επιστρέφει τελικά το πρόγραμμα στην κύρια ροή του.
- Το **δεύτερο λογικό διάγραμμα** σε αντίθεση με τον πρώτο, περιμένει αρχικά 50 ms, για να ολοκληρωθεί ο παλμός που προκάλεσε την παρούσα διακοπή. Στη συνέχεια, εκτελεί το πρόγραμμα διακοπής και στη περίπτωση που ανιχνεύσει κάποια επιπλέον διακοπή, ελέγχει συνεχώς τη μάσκα διακοπών μέχρι να ολοκληρωθεί ο επιπλέον παλμός και τέλος δίνοντας μια επιπλέον καθυστέρηση 50 ms επιστρέφει στην κύρια ροή. Πρέπει να τονιστεί ότι η επιπλέον καθυστέρηση γίνεται για επιβεβαίωση ολοκλήρωσης του τελευταίου παλμού.