

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ  
ΠΟΛΥΤΕΧΝΕΙΟ

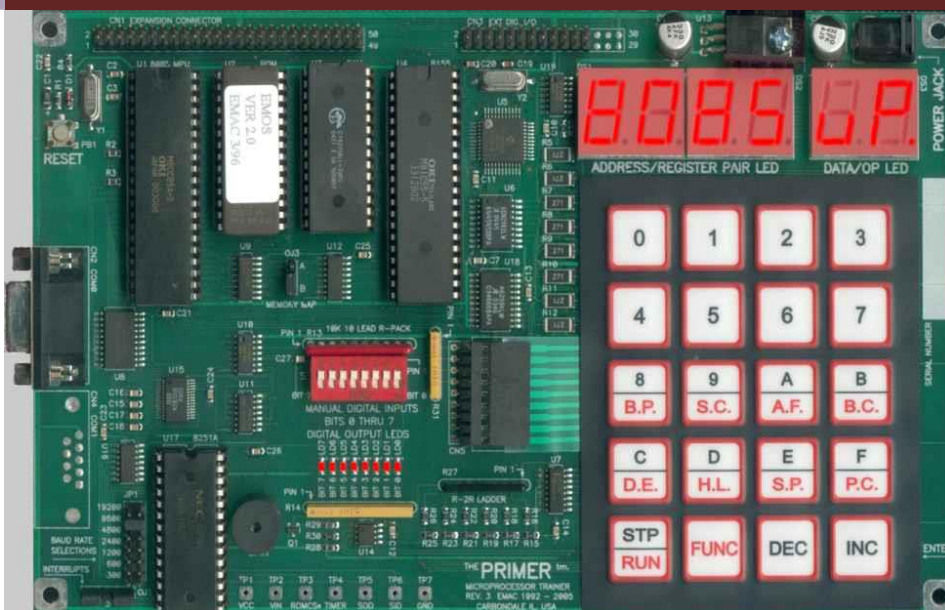
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ  
ΜΗΧΑΝΙΚΩΝ

&

ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ

Εργαστήριο  
Μικροϋπολογιστών

## 2<sup>η</sup> Σειρά Ασκήσεων



7<sup>ο</sup> Εξάμηνο

ΡΟΗ Υ

Αθανασίου  
Νικόλαος

AM 03112074

Βαβουλιώτης  
Γεώργιος

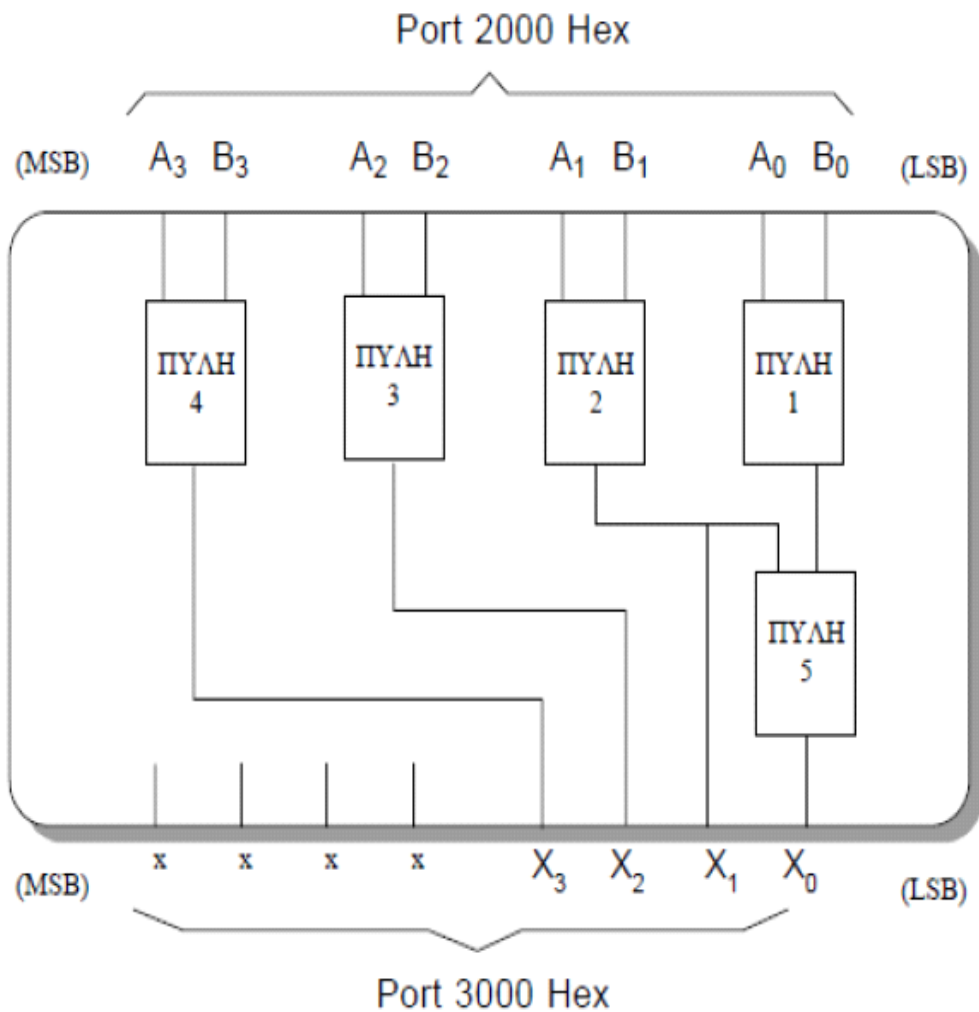
AM 03112083

Γιαννούλας  
Βασίλειος

AM 03112117

## Άσκηση 3i

**Σκοπός της άσκησης:** Να εξομοιωθεί η λειτουργία ενός υποθετικού I.C. με πύλες όπως φαίνεται στο σχήμα 8. Τα bits εισόδου πρέπει να αντιστοιχούν ακριβώς όπως φαίνονται στο σχήμα με τα dip switches της πόρτας εισόδου 2000 Hex, και τα LEDs πρέπει να είναι τα τέσσερα LSB της πόρτας εξόδου 3000 Hex. Οι πύλες 1, 2, 3, 4, 5 να υλοποιούν τις AND, AND, OR, OR και XOR αντίστοιχα.



**Επεξήγηση κώδικα :** Για να δημιουργήσουμε την έξοδο κάθε πύλης απομονώνουμε τα bit που είναι οι εισόδους της πύλης και με κατάλληλες ολισθήσεις τα φέρνουμε στις ίδιες θέσεις έτσι ώστε να μπορεί να γίνει η λογική πράξη .Επίσης ο καταχωρητής E κρατά κάθε στιγμή την τρέχουσα έξοδο .

Ο κώδικας σε assembly 8085 φαίνεται παρακάτω :

```

START:
    MVI E,00H
    LDA 2000H
    MOV B,A          ; (B)<-DIP_SWITCHES
    ANI 01H          ; krataw to B0
    MOV C,A          ; (C)<-B0
    MOV A,B
    RRC              ; fernw to A0 sto deksiotero bit
    ANI 01H          ; kai krataw to A0
    ANA C            ; (BO)AND(A0)
    MOV E,A          ; (E)<-(BO)AND(A0)

LABEL_X1_X0:
    MOV A,B          ; epanafora tw n diakoptwn ston A
    ANI 04H          ; krataw to B1
    MOV C,A          ; (C)<-B1
    MOV A,B
    RRC              ; fernw to A1 sto 3o bit apo aristera
    ANI 04H
    ANA C            ; (A)<-(B1)AND(A1)
    MOV C,A          ; krataw to (B1)AND(A1)giati to 8elw kai gia epomenh
prakh
    RRC              ; paw to apotelesma sto 2o bit apo aristera
    MOV D,A          ; swzw to apotelesma ston D gia na kanw thn prakh toy X0
    RRC              ; to paw sto 1o bit gia na kanw thn prakh toy X0
    XRA E            ; (B1 AND A1) XOR (BO AND AO)= X0
    ADD D            ; D=X1 ARA A=X1X0
    MOV E,A          ; (E)<-X1X0

LABEL_X2:
    MOV A,B          ; epanafora tw n diakoptwn ston A
    ANI 10H          ; krataw to B2
    MOV C,A          ; (C)<-B2
    MOV A,B
    RRC              ; fernw to A2 sto 5o apo deksia bit
    ANI 10H          ; krataw to A2
    ORA C            ; (A)<-(B2)OR(A2)
    ANI 10H          ; krataw mono to apotelesma thn NOR sto 5o bit
    RRC              ; kai to metaferw sto 3o bit
    RRC
    ADD E            ; (A)<-X2X1X0
    MOV E,A          ; (E)<-X2X1X0

LABEL_X3:
    MOV A,B          ; epanafora tw n diakoptwn ston A
    ANI 40H          ; krataw to B3
    MOV C,A          ; (C)<-B3
    MOV A,B
    RRC              ; paw sto A3 sto 7o bit
    ANI 40H          ; kai krataw mono to A3
    ORA C            ; kanw thn prakh (B3)OR(A3)
    ANI 40H          ; krataw mono to apotelesma thw OR
    RRC              ; metafora sto 4o bit
    RRC
    RRC
    ADD E            ; bazw to teliko apotelesma ston A

LABEL_END:
    CMA
    STA 3000H
    JMP START

END

```

## Άσκηση 4ii

**Σκοπός της άσκησης:** Σκοπός του προγράμματος είναι να απεικονίζει στα δύο αριστερότερα displays την τιμή του κωδικού του πλήκτρου που πατήθηκε. Χρησιμοποιούμε τη ρουτίνα KIND για τη λειτουργία ανάγνωσης του πληκτρολογίου και τις STDM και DCD για τη λειτουργία αποστολής των δεδομένων στα displays.

**Επεξήγηση κώδικα:** Αρχικά διαβάζουμε το πλήκτρο που πατήθηκε μέσω της ρουτίνας KIND η οποία αποθηκεύει στον A τον αριθμό που πατήθηκε, στην συνέχεια καθαρίζουμε τα τέσσερα δεξιότερα 7-segments, βάζοντας στον καταχωρητή A την τιμή 10H και αποθήκευση αυτής της τιμής στις αντίστοιχες διευθύνσεις, γιατί θέλουμε το αποτέλεσμα να εμφανιστεί μόνο στα δύο αριστερότερα displays. Τέλος με την χρήση των ρουτίνων STDM και DCD εμφανίζουμε τον κωδικό του πλήκτρου.

Ο κώδικας σε assembly 8085 φαίνεται παρακάτω :

BEGIN:

```
CALL KIND          ;apo8hkeush ston A tou kwdikou tou plhktrou pou
                   ;pathsa

MOV B,A            ;swzw ton A gia na ka8arish ta 7-segments pou den
                   ;8elw
MVI A,10H          ;(A)<-10H gia na ka8arish ta 7-segments
STA 0BF0H
STA 0BF1H
STA 0BF2H
STA 0BF3H
MOV A,B            ;epanafora periexomenoy tou A

LXI D,0BF4H        ;fortwnw ston DE thn address sthn
                   ;opoia 8a emfanish

ANI 0FH            ;krataw to lsb tou kwdikou pou diavasa me thn KIND
STAX D             ;to apo8hkeuw sthn dieu8unsh 0BF4H
INX D              ;((D)(E))<-0BF5H
MOV A,B            ;epanafora ston A tou kwdikou
ANI 0FH            ;krataw to msb tou kwdikou pou diavasa me thn KIND
STAX D             ;to apouhkeuw sth dieu8unsh 0BF5H
LXI D,0BF0H        ;thn arxikh dieu8unsh thn exei o DE pou prepei na
                   ;dei h STDM

CALL STDM          ;h STDM metakinei to munhma to opoio ksekina apo
                   ;thn dieu8unsh pou deixnei o DE
                   ;kai perimenei mexri na vrei thn rutina DCD

CALL DCD
JMP BEGIN          ;sunexomenh leitourgia
```

END

## Άσκηση 4iv

**Σκοπός της άσκησης:** Σκοπός της άσκησης είναι να γραφεί ρουτίνα που να απεικονίζει στα τέσσερα δεξιότερα 7-segment την τιμή που διαβάζεται από τη θύρα εισόδου 2000 σε δεκαδική μορφή (1 για το πρόσημο και 3 για την δεκαδική τιμή). Η διαδικασία πρέπει να είναι συνεχόμενη.

**Επεξήγηση κώδικα:** Αρχικά διαβάζεται ο αριθμός που δίνετε από τους διακόπτες και ελέγχεται αν είναι θετικός ή αρνητικός( αν είναι αρνητικός παίρνουμε το συμπλήρωμα του ως προς 2 για να υπολογίσουμε τα ψηφία ). Στην συνέχεια αφαιρούμε συνεχώς το 100 μέχρι να φτάσουμε σε αριθμό μικρότερο του 100 και για κάθε εκατοντάδα που αφαιρούμε αυξάνουμε κατά 1 τον καταχωρητή B, συνεχίζουμε την ίδια διαδικασία για να υπολογίσουμε και τις δεκάδες αφαιρώντας 10 αυτή την φορά και αποθηκεύοντας τις δεκάδες στον καταχωρητή C και όποιος αριθμός περισσέψει από την παραπάνω διαδικασία είναι οι μονάδες που αποθηκεύονται στον καταχωρητή L. Το θετικό πρόσημο το απεικονίζουμε με ένα μηδενικό γιατί δεν υπήρχε δυνατότητα απεικόνισης του συμβόλου '+' στον ενδείκτη , ενώ το αρνητικό πρόσημο το απεικονίζουμε κανονικά .

Ο κώδικας σε assembly 8085 φαίνεται παρακάτω :

```
BEGIN:
    LDA 2000H          ;diavasma eisodou
    MVI B,00H          ;ekatontades
    MVI C,00H          ;dekades
    MOV H,A
    MVI E,00H
    RAL
    JNC GREATER_THAN_100
    MVI E,01H
    MOV A,H
    CMA
    ADI 01H
    JMP LABEL--

GREATER_THAN_100:
    MOV A,H
LABEL--:
    CPI 64H            ;sugkrinw me to 100
    JNC BIGGER_100    ;phgainw sthn etiketa BIGGER_100 an o ari8mos
                        ;einai megalyteros toy 100

GREATER_THAN_10:
                        ;an exw er8ei edw sigoura o ari8mos einai <100
    CPI 0AH           ;sugkrinw me to 10
    JNC BIGGER_10     ;phgainw sthn etiketa BIGGER_10 an o ari8mos
                        ;einai megaluteros tou 10

LABEL_SEGM:
    MOV L,A
    MVI A,10H          ;ka8arizw ta pshfia pou den 8elw na emfanisw
    STA 0BF4H
    STA 0BF5H

    LXI D,0BF0H
    MOV A,L            ;apo8hkeuw tis monades
    STAX D
    INX D

    MOV A,C
    STAX D            ;apo8hkeuw tis dekades
    INX D

    MOV A,B
    STAX D            ;apo8hkeuw tis ekatontades
```

```

        INX D

        MOV A,H
        ANI 80H
        CPI 80H
        JNZ LABEL+
        MVI A,1CH
        STAX D
        JMP CALL_ROUTINES

LABEL+:
        MVI A,00H
        STAX D
        INX D

CALL_ROUTINES:
        LXI D,0BF0H
        CALL STDH          ;plhrhs emfanish
        CALL DCD
        JMP BEGIN

BIGGER_100:
        SUI 64H            ;afairw 100 apo ton ari8mo
        INR B              ;ekatontades+=1
        JMP LABEL--

BIGGER_10:
        SUI 0AH            ;afairw 10 apo ton ari8mo
        INR C              ;dekades+=1
        JMP GREATER_THAN_10

END

```