

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχ. Και Μηχ. Υπολογιστών

Εργαστήριο Μικροϋπολογιστών , 7ο εξάμηνο - Ροή Υ

Ακαδημαϊκή Περίοδος : 2011-2012



2^η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ

Ομάδα: C16

Γερακάρης Βασίλης Α.Μ.: 03108092

Διβόλης Αλέξανδρος Α.Μ.: 03107238

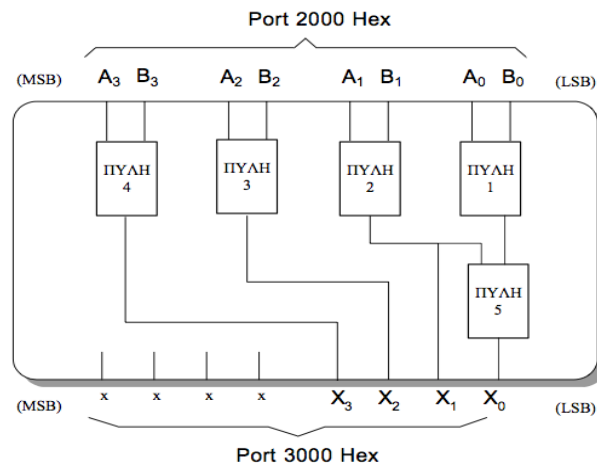
Τεντσεκ Στέργιος Α.Μ.: 03108690

Άσκηση 3i

Στην άσκηση αυτή πρέπει να υλοποιήσουμε το IC του σχήματος, με 4 πύλες XOR και μια OR. Για να το πετύχουμε αυτό παίρνουμε την είσοδο, την ολισθαίνουμε κατά μια θέση και κάνουμε XOR έτσι, ανάλογα με την ολίσθηση, δεξιά ή αριστερή, έχουμε στα άρτια ή στα περιττά bit αντίστοιχα, την έξοδο των XOR. Έπειτα περνούμε αυτά τα αποτελέσματα στις κατάλληλες θέσεις, και κάνουμε και μια OR μεταξύ των δυο δεξιών εξόδων των πυλών XOR. Έπειτα εμφανίζουμε το αποτέλεσμα. Ο κώδικας παρατίθεται παρακάτω:

```
START: MVI C,00H      ;Αρχικοποιούμε τον C ο οποίος θα έχει το αποτέλεσμα
        LDA 2000H     ;Διαβάζουμε την είσοδο
        MOV B,A       ;Την κρατάμε στον B
        RRC          ;Κάνουμε μια ολίσθηση δεξιά ώστε να έρθουν τα δεδομένα στη
                    ;κατάλληλη θέση
        XRA B         ;Κάνουμε XOR
        ANI 55H       ;Κρατάμε τα κατάλληλα αποτελέσματα 0 X3 0 X2 0 X1 0 X
        MOV B,A
        ANI 40H       ;Φέρνουμε το X3 στην κατάλληλη θέση εξόδου
        RRC
        RRC
        RRC
        ADD C         ;Το κρατάμε στον C
        MOV C,A
        MOV A,B
        ANI 10H       ;Φέρνουμε το X2 στην κατάλληλη θέση εξόδου
        RRC
        RRC
        ADD C         ;Το κρατάμε στον C
        MOV C,A
        MOV A,B
        ANI 04H       ;Φέρνουμε το X1 στην κατάλληλη θέση εξόδου
        RRC
        ADD C         ;Το κρατάμε στον C
        MOV C,A
        MOV A,B       ;Παίρνουμε την έξοδο των XOR
        RRC
        RRC
        ORA B         ;Κάνουμε OR μεταξύ των δυο τελευταίων
        ANI 01H       ;Φέρνουμε το X0 στην κατάλληλη θέση εξόδου
        ADD C         ;Το κρατάμε στον C
        CMA          ;Αντιστρέφουμε
        STA 3000H     ;Το εμφανίζουμε
        JMP START     ;Ξαναρχίζουμε

END
```



Άσκηση 4ii

Στην άσκηση αυτή πρέπει να υλοποιήσουμε ένα πρόγραμμα το οποίο θα εμφανίζει τον κωδικό του πλήκτρου που πατάμε. Για να το πετύχουμε αυτό χρησιμοποιούμε την ρουτίνα KIND που μας επιστρέφει αυτό τον κωδικό, τον κρατάμε σε ένα καταχωρητή και μετά κάνοντας 4 ολισθήσεις και περνώντας τον από κατάλληλη μάσκα κρατάμε το πρώτο δεκαδικό του ψηφίο, έπειτα βάζουμε σε έξι συνεχόμενες θέσεις μνήμης τα αποτελέσματα που θέλουμε να εμφανιστούν στα 7 segment displays, δηλαδή το 10_H στα τέσσερα πρώτα ώστε να μείνουν σβηστά και τα δυο ψηφία που θέλουμε στα επόμενα. Καλούμε την ρουτίνα STDm έχοντας βάλει στον D-E τη θέση μνήμης στην οποία έχουμε αποθηκεύσει τα αποτελέσματα, ώστε να εμφανιστούν. Ο κώδικας παρατίθεται παρακάτω:

```
IN 10H          ;Άρση προστασίας μνήμης

LXI H,0A05H     ;Θέση μνήμης που θα βάλουμε το αποτέλεσμα
MVI M,10H       ;θέτουμε σβηστά τα 4 πρώτα 7 segment displays
DCX H
MVI M,10H
DCX H
MVI M,10H
DCX H
MVI M,10H

START: LXI H,0A01H
CALL KIND
MOV B,A
RRC             ;Κρατάμε τα 4 MSBits άρα το πρώτο δεκαδικό ψηφίο
RRC             ;του κωδικού
RRC
RRC
ANI 0FH
MOV M,A         ;βάζουμε το πρώτο ψηφίο του κωδικού
MOV A,B         ;Φέρνουμε τον B στον A
ANI 0FH         ;Κρατάμε τα 4 LSBits
DCX H           ;στη θέση 0A00H
MOV M,A         ;βάζουμε το δεύτερο ψηφίο του κωδικού

LXI D,0A00H     ;Βάζουμε στον D-E την θέση μνήμης όπου αρχίζουν τα
CALL STDm       ;δεδομένα που θέλουμε να δείξουμε καλούμε την STDm

JMP START

END
```

Άσκηση 4iv

Στην άσκηση αυτή υλοποιούμε ένα πρόγραμμα το οποίο εμφανίζει σε δεκαδική μορφή την είσοδο που παίρνει από τους διακόπτες. Πρέπει να λάβουμε υπόψη μας ότι η είσοδος δίνεται ως συμπλήρωμα του 2. Αρχικά τοποθετούμε στη μνήμη σε τρεις διαδοχικές θέσεις την τιμή 10_H ώστε μα μείνουν τα 3 πρώτα displays σβηστά. Έπειτα ελέγχουμε το MSB της εισόδου, για να δούμε αν ο αριθμός είναι αρνητικός, αν είναι βάζουμε στη θέση μνήμης που ελέγχει το 3^ο display το 1C_H ώστε να εμφανιστεί το σύμβολο πλην, και συμπληρώνουμε τον αριθμό προσθέτοντας του και ένα ώστε να πάρουμε το συμπλήρωμα ως προς 2. Αφού έχουμε λοιπόν τον θετικό ελέγχουμε αν είναι μεγαλύτερος του εκατό, αν είναι βάζουμε στην κατάλληλη θέση μνήμης το 1 και αφαιρούμε εκατό, αλλιώς βάζουμε το 0. Έπειτα με την επαναληπτική διαδικασία που περιγράφεται στο βιβλίο χωρίζουμε σε δεκάδες και μονάδες, τις αποθηκεύουμε στην κατάλληλη θέση μνήμης και εμφανίζουμε το αποτέλεσμα.

```
IN 10H ;Άρση προστασίας μνήμης
START: LXI H,0A05H ;Θέση μνήμης που θα βάλουμε το αποτέλεσμα
      LXI B,0000H ;Αρχικοποίηση καταχωρητων
      LXI D,0000H
      MVI A,00H
      MVI M,10H ;Θέτουμε σβηστά τα 3 πρώτα 7 segment displays
      DCX H
      MVI M,10H
      DCX H
      MVI M,10H
      LDA 2000H ;Διαβάζουμε την είσοδο
      MOV B,A ;Την κρατάμε στο B
      RAL ;Βλέπουμε αν είναι 1 το MSB
      CC TOPOS ;Αν είναι πρέπει να κάνουμε συμπλήρωμα 2
      MOV A,B ;Παίρνουμε τον θετικό πλέον αριθμό που έχουμε να απεικονίσουμε
      CPI 64H ;Τον συγκρίνουμε με το 100d
      CNC HUN ;Αν είναι μεγαλύτερος ή ισος του 100 πάμε στη ρουτίνα HUN
CHECK: CPI 0AH ;Συγκρίνουμε με το 10
      JNC GRTEN ;Αν είναι μεγαλύτερο ή ισο πάμε στο GRTEN
      MOV E,A ;Κρατάμε και τις μονάδες
      DCX H
      MOV M,C ;Βάζουμε τις εκατοντάδες στο 4ο 7 segment display
      DCX H
      MOV M,D ;Βάζουμε τις δεκάδες στο 5ο 7 segment display
      DCX H
      MOV M,E ;Βάζουμε τις μονάδες στο 6ο 7 segment display
      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
TOPOS: MOV A,B ;Παίρνουμε τον αριθμό
      CMA ;Τον αντιστρέφουμε
      ADI 01H ;Προσθέτουμε 1
      MOV B,A ;Τον κρατάμε στο B
      MVI M,1CH ;Βάζουμε στο 3ο BCD το μείον
      RET
HUN: MVI C,01H ;Βάζουμε στο C 1 όπου είναι οι εκατοντάδες
      SBI 64H ;Αφαιρούμε 100 και επιστρέφουμε
      RET
GRTEN: INR D ;Αυξάνουμε τις δεκάδες
      SBI 0AH ;Αφαιρούμε 10 από τον αριθμό και επιστρέφουμε
      JMP CHECK
END
```