



**3<sup>η</sup> ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ**  
**ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"**  
**Παράδοση 25/5/2020**

**Ασκήσεις προσομοίωσης** (να υλοποιηθούν και να δοκιμαστούν στο πρόγραμμα προσομοίωσης του εκπαιδευτικού συστήματος μLAB)

**1<sup>η</sup> ΑΣΚΗΣΗ:** Στο μLAB να γραφεί πρόγραμμα Assembly, που να ελέγχει μέσω της διακοπής τύπου RST 6.5 τα φώτα ενός χώρου. Όταν προκαλείται διακοπή τύπου RST 6.5 (πάτημα πλήκτρου INTR) να αναβοσβήνουν (με περίοδο  $\sim 1/2$  sec) όλα τα LED της πόρτας εξόδου. Αυτό να παραμένει για περίπου ένα (1) λεπτό της ώρας και μετά να σβήνει. Αν όμως ενδιάμεσα ξαναενεργοποιηθεί η διακοπή να ανανεώνεται ο χρόνος του ενός λεπτού. Ο χρόνος που υπολείπεται να απεικονίζεται σε sec συνεχώς στα 2 δεξιότερα δεκαεξαδικά ψηφία των 7-segments displays και σε δεκαδική μορφή.

Να γίνει χρήση των ρουτινών χρονοκαθυστέρησης του εκπαιδευτικού συστήματος μLAB.

**Σημ.:** Στον προσομοιωτή υπάρχει η "ιδιοτροπία" να προκαλούνται 2 διακοπές σε κάθε ενεργοποίηση της διακοπής (προκαλείται μία διακοπή στο πάτημα και μία στο άφημα του πλήκτρου ή του ποντικιού). Ένας τρόπος διαχείρισης του προβλήματος είναι στον έλεγχο λειτουργίας να κρατάτε πατημένο το πλήκτρο και όταν το αφήνετε να υπολογίζεται αυτό ως 2<sup>η</sup> διακοπή. Αλλιώς λόγω της μικρής διάρκειας μεταξύ πατήματος και απελευθέρωσης του πλήκτρου δεν αλλάζει τη λειτουργία η διπλή διακοπή.

**2<sup>η</sup> ΑΣΚΗΣΗ:** Να υλοποιηθεί και να εκτελεστεί στο μLAB πρόγραμμα σε assembly που όταν προκαλείται διακοπή τύπου RST 6.5 (πάτημα πλήκτρου INTR) να διαβάσει τα 2 διαδοχικά δεκαεξαδικά ψηφία ενός αριθμού (00-FFH => 0-255) που δίνονται στη συνέχεια από το πληκτρολόγιο (χρήση της ρουτίνας KIND) και να τα απεικονίζει στα 2 αριστερότερα 7-segment display (βάσει των ρουτινών DCD και STDM). Να συγκρίνει την τιμή αυτή με τρία κατώφλια K1, K2 και K3 με  $K1 < K2 < K3$ , που οι τιμές τους βρίσκονται στους καταχωρητές C, D και E αντίστοιχα. Για να είναι ο έλεγχος εύκολος, στον κώδικα βάλτε συγκεκριμένες τιμές στους καταχωρητές C, D και E. Στην συνέχεια να ανάβει ένα από τα τέσσερα LSB LED εξόδου που αντιστοιχούν στις περιοχές τιμών [0..K1], (K1..K2], (K2..K3] και (K3..FFH]. Το κύριο πρόγραμμα να είναι μετά τις αρχικοποιήσεις ένα ατέρμων βρόχος αναμονής (των διακοπών).

**Σημ.:** Και εδώ η διπλή διακοπή δεν δημιουργεί πρόβλημα στον έλεγχο της λειτουργίας (απλά η ρουτίνα εξυπηρέτησης της διακοπής θα τρέξει 2 φορές).

### Θεωρητικές Ασκήσεις

**3<sup>η</sup> ΑΣΚΗΣΗ:** α) Δώστε τη μακροεντολή INR16 ADDR που να αυξάνει έναν αριθμό X των 16 bit αποθηκευμένο σε 2 διαδοχικές θέσεις στη μνήμη ως εξής:  $X_{LOW} = (ADDR)$ ,  $X_{HIGH} = (ADDR+1)$ . Το αποτέλεσμα να επιστρέφει στις ίδιες θέσεις. Η εκτέλεση της μακροεντολής δεν πρέπει να επηρεάζει τα περιεχόμενα των υπολοίπων καταχωρητών γενικού σκοπού.

β) Δώστε τη μακροεντολή FILL ADDR, K, η οποία γεμίζει ένα τμήμα μνήμης με αρχική διεύθυνση ADDR και μήκος K με τους αριθμούς K, K-1, ..2, 1. Το μέγεθος του τμήματος K μπορεί να είναι από 0 έως 255. Θεωρείστε ότι για K=0 το μέγεθος του τμήματος να είναι ίσο με 256 και οι αριθμοί που θα αποθηκευθούν να είναι 0, 255, 254, ...1.

γ) Δώστε τη μακροεντολή RHLR Q, R που περιστρέφει τα περιεχόμενα του κρατουμένου CY, των καταχωρητών Q και R κατά μια θέση αριστερά. Οι καταχωρητές Q και R μπορεί να είναι ένας συνδυασμός εκ των B, C, D, E, H και L (φυσικά  $Q \neq R$ ). Η μακροεντολή συμπεριφέρεται στα CY, Q και R σαν να είναι ένας 17-bit καταχωρητής: CY(17° bit): Q(16° -9° bit): R(8° -1° bit). Μπορείτε να κάνετε χρήση της στοίβας για την αποθήκευση και επαναφορά τιμής καταχωρητών.

**4<sup>η</sup> ΑΣΚΗΣΗ:** Στο  $\mu\text{E}$  8085 εκτελείται η εντολή **JMP 0900H**. Ο μετρητής προγράμματος είναι (PC)=0800H και ο δείκτης σωρού (SP)=1FF0H. Στο μέσον της εκτέλεσης της εντολής συμβαίνει διακοπή RST 6.5. Δώστε τις νέες τιμές των PC, SP, το περιεχόμενο του σωρού καθώς και τις λειτουργίες που συμβαίνουν.

**5<sup>η</sup> ΑΣΚΗΣΗ:** Να γραφεί πρόγραμμα Assembly (και η ρουτίνα εξυπηρέτησης) σε  $\mu\text{Y}-\Sigma$  8085 που να λαμβάνει 16 δεδομένα των 8 bit από μια συσκευή. Το καθένα μεταφέρεται σε 2 βήματα (πρώτα τα 4 MSB και μετά τα 4 LSB – συνολικά θα χρειαστούν 32 βήματα) μέσω των ( $X_0-X_3$ ) της θύρας PORT\_IN (20<sup>H</sup>) ενώ τα υπόλοιπα MSbit της θύρας ( $X_4-X_7$ ) δεν χρησιμοποιούνται. Η συσκευή για κάθε 4 bit που αποστέλλει, προκαλεί πριν διακοπή RST5.5. Να υπολογιστεί ο μέσος όρος των 16 δεδομένων με ακρίβεια 8 bit.

