



3^η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ
ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"
Παράδοση 2/5/2025

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

1^η 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 ανάβουν 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^η διακοπή. Αλλιώς λόγω της μικρής διάρκειας μεταξύ πατήματος και απελευθέρωσης του πλήκτρου δεν αλλάζει τη λειτουργία η διπλή διακοπή.

2^η 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 (πρώτα το χαμηλότερης αξίας και μετά το υψηλότερης αξίας) ενός αριθμού (0-255) που δίνονται στη συνέχεια από το πληκτρολόγιο (ρουτίνα KIND) και να τα απεικονίζει στα 2 **αριστερότερα** 7-segment display (βάσει των ρουτινών DCD και STDM). Να συγκρίνει την τιμή αυτή με δυο κατώφλια K1 και K2 με $K1 < K2$, που οι τιμές τους βρίσκονται στους καταχωρητές D και E αντίστοιχα. Για να είναι ο έλεγχος εύκολος, στον κώδικα βάλτε συγκεκριμένες τιμές στους καταχωρητές D και E. Στην συνέχεια να ανάβει ένα από τα τρία LED εξόδου (τα πρώτα LSB) που αντιστοιχούν στις περιοχές τιμών $[0...K1]$, $(K1...K2]$ και $(K2...FFH]$.

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

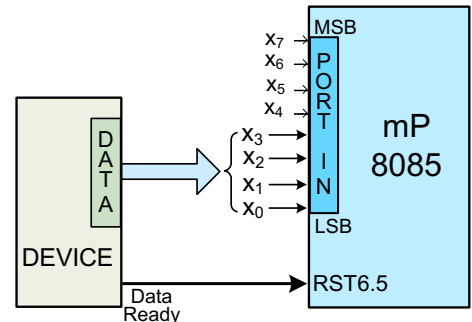
3^η ΑΣΚΗΣΗ: α) Τη μακροεντολή SWAP Nibble Q που εναλλάσσει το χαμηλότερης αξίας HEX ψηφίο με το υψηλότερης των καταχωρητών γενικού σκοπού B, C, D, E, H και L καθώς και της θέσης μνήμης που 'δείχνει' ο διπλός καταχωρητής H-L (δηλ. εναλλαγή των HEX ψηφίων της). Η εκτέλεση της μακροεντολής δεν πρέπει να επηρεάζει τα περιεχόμενα των υπολοίπων καταχωρητών γενικού σκοπού.

β) Τη μακροεντολή FILL RP, X, K, η οποία γεμίζει ένα τμήμα μνήμης με μια σταθερά K (0-255). Η παράμετρος X παίρνει τιμές 0-255. Το μέγεθος του τμήματος μπορεί να είναι από 1 έως 256, θεωρώντας για $(X)=0$ το μέγεθος του τμήματος ότι είναι ίσο με 256. Η αρχική διεύθυνση καθορίζεται από το διπλό καταχωρητή RP (BC, DE, HL), το μήκος (X) και η σταθερά K καθορίζονται από τις παραμέτρους της μακροεντολής.

γ) Επίσης δώστε τη μακροεντολή RHLR, που **περιστρέφει** τα περιεχόμενα των καταχωρητών H και L κατά μια θέση **δεξιά**. Έτσι η μακροεντολή συμπεριφέρεται στους H και L σαν να είναι ένας 16-bit καταχωρητής με το CY να παίρνει κάθε φορά την τιμή L_0 (LSB του καταχ. L).

4^η ΑΣΚΗΣΗ: Στο μΕ 8085 εκτελείται η εντολή **CALL 0900H**. Ο μετρητής προγράμματος είναι (PC)=0840H και ο δείκτης σωρού (SP)=3000H. Στο μέσον της εκτέλεσης της εντολής συμβαίνει διακοπή RST 5.5. Δώστε τις νέες τιμές των PC, SP, το περιεχόμενο του σωρού καθώς και τις λειτουργίες που συμβαίνουν στην αρχή και στην επιστροφή από την ρουτίνα εξυπηρέτησης.

5^η ΑΣΚΗΣΗ: (α) Να γραφεί πρόγραμμα Assembly (και η ρουτίνα εξυπηρέτησης) σε μΥ-Σ 8085 που να λαμβάνει 32 δεδομένα των 8 bit από μια συσκευή. Το καθένα δεδομένο μεταφέρεται σε 2 βήματα (πρώτα τα 4 LSB και μετά τα 4 MSB – συνολικά θα χρειαστούν 64 βήματα) μέσω των (X_0 - X_3) της θύρας *PORT_IN* (20^H) ενώ τα υπόλοιπα MSbit της θύρας (X_4 - X_7) δεν χρησιμοποιούνται. Η συσκευή για κάθε 4 bit που αποστέλλει, προκαλεί πριν διακοπή RST6.5 μέσω του σήματος *Data Ready*. Επίσης το πρόγραμμα να υπολογίζει το μέσο όρο των 32 δεδομένων με ακρίβεια 8 bit (να γίνει στρογγύλευση).



(β) Να επιλύσετε το ίδιο θέμα θεωρώντας ότι δεν υπάρχουν διακοπές και πως ο έλεγχος νέου δεδομένου γίνεται μέσω του x_7 στην οποίαν συνδέουμε τη γραμμή *Data Ready*. Θα πρέπει να ελέγχουμε κάθε φορά το θετικό μέτωπό του σήματος αυτού για νέο δεδομένο καθώς και την επαναφορά του στη στάθμη '0'.