



3^η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ
ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"
Παράδοση 8/6/2022

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 αναβοσβήνουν (με περίοδο $\sim 1/2$ sec) όλα τα LED της πόρτας εξόδου. Αυτό να διαρκεί για περίπου 45 sec και μετά να σβήνουν. Αν όμως ενδιάμεσα ξαναενεργοποιηθεί η διακοπή να ανανεώνεται ο χρόνος των 45 sec. Ο χρόνος που υπολείπεται, να απεικονίζεται σε sec συνεχώς στα 2 αριστερότερα 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 ενός αριθμού (00-FFH \Rightarrow 0-255) που δίνονται στη συνέχεια από το πληκτρολόγιο (χρήση της ρουτίνας KIND) και να τα απεικονίζει στα 2 μεσαία 7-segment display (βάσει των ρουτινών DCD και STDM). Να συγκρίνει την τιμή αυτή με τρία κατώφλια K1, K2 και K3 θετικοί αριθμοί $\in \{0-255\}$, με $K1 < K2 < K3$, που οι τιμές τους βρίσκονται στους καταχωρητές C, D και E αντίστοιχα. Για να είναι ο έλεγχος, 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^η ΑΣΚΗΣΗ: α) Δώστε τη μακροεντολή INR16 ADDR που να αυξάνει κατά 1, έναν αριθμό X των 16 bit αποθηκευμένο σε 2 διαδοχικές θέσεις στη μνήμη ως εξής: $X_{\text{LOW}} = (\text{ADDR})$, $X_{\text{HIGH}} = (\text{ADDR} + 1)$. Το αποτέλεσμα να επιστρέφει στις ίδιες θέσεις. Η εκτέλεση της μακροεντολής δεν πρέπει να επηρεάζει τα περιεχόμενα των υπολοίπων καταχωρητών γενικού σκοπού.

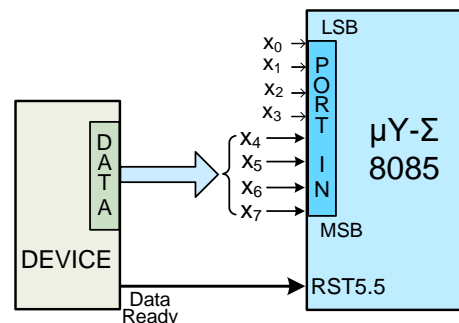
β) Δώστε τη μακροεντολή FILL ADDR, K, η οποία γεμίζει ένα τμήμα μνήμης με αρχική διεύθυνση ADDR και μήκος K με τους αριθμούς K, K-1, ...1. Το μέγεθος του τμήματος μπορεί να είναι από 1 έως 256. Θεωρείστε επίσης ότι για K=0 το μέγεθος του τμήματος να είναι ίσο με 256 και οι αριθμοί που θα αποθηκευθούν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

γ) Δώστε τη μακροεντολή RHLL Q, R που περιστρέφει τα περιεχόμενα του κρατούμενου 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: $CY(17^\circ \text{ bit})$: $Q(16^\circ - 9^\circ \text{ bit})$: $R(8^\circ - 1^\circ \text{ bit})$. Μπορείτε να κάνετε χρήση της στοίβας για την αποθήκευση και επαναφορά τιμής καταχωρητών.

4^η ΑΣΚΗΣΗ: Στο μE 8085 εκτελείται η εντολή **JMP 2200H**. Ο κωδικός της εντολής είναι στη θέση 2000H, δηλ. ο μετρητής προγράμματος είναι $(PC)=2000H$ και ο δείκτης σωρού $(SP)=3000H$. Στο μέσον της εκτέλεσης της εντολής αυτής συμβαίνει διακοπή **RST 6.5**. Δώστε τις νέες τιμές των PC , SP , το περιεχόμενο του σωρού μετά την εκτέλεση της διακοπής καθώς και τις λειτουργίες που συμβαίνουν.

5^η ΑΣΚΗΣΗ: Να γραφεί πρόγραμμα Assembly (και η ρουτίνα εξυπηρέτησης διακοπής) σε μY -Σ 8085 που να λαμβάνει 16 δεδομένα των 8 bit από μια συσκευή. Το καθένα δεδομένο μεταφέρεται σε 2 βήματα (πρώτα τα 4 LSB και μετά τα 4 MSB κάθε φορά) μέσω των (X_4-X_7) της θύρας **PORT_IN** (20^H) ενώ τα υπόλοιπα LSbit της θύρας (X_0-X_3) δεν χρησιμοποιούνται. Συνολικά θα χρειαστούν 32 βήματα. 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 (να γίνει με περικοπή).



(β) Να επιλύσετε το ίδιο θέμα θεωρώντας ότι δεν υπάρχουν διακοπές και πως ο έλεγχος νέου δεδομένου γίνεται μέσω του x_0 στην οποίαν συνδέουμε τη γραμμή *Data Ready*. Θα πρέπει να ελέγχουμε κάθε φορά το αρνητικό μέτωπό του σήματος αυτού για νέο δεδομένο καθώς και την επαναφορά του στη στάθμη '1'.