



ΑΘΗΝΑ 28 - 4 - 2017

3^η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ
ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

Παράδοση 21/5/2017

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

1^η - 4^η ΑΣΚΗΣΗ: Να γραφούν σε assembly 8085 και να εκτελεστούν στο μLAB , 4 προγράμματα με τις εξής λειτουργίες:

- i. Διαβάζει την πόρτα εισόδου των dip switches και με βάση το 1^ο δεξιότερο ON, ανάβει το αντίστοιχης τάξης led και όλα τα υψηλότερης τάξης led μετά από αυτό (π.χ. για 1011 0100 => XXXX XX00). Το πρόγραμμα να είναι συνεχούς λειτουργίας.
- ii. Να αναμένει το πάτημα του δεκαεξαδικού πληκτρολογίου και μόνο των αριθμών 1 έως 8. Κάθε φορά να ανάβει το led της αντίστοιχης θέσης (1=>LSB, 8=>MSB). Να γίνει χρήση της ρουτίνας KIND που υπάρχει στο παράρτημα 1 των σημειώσεων του μLAB . Το πρόγραμμα να είναι συνεχούς λειτουργίας.
- iii. Με βάση την ύλη των σελ. 76 – 79 (των σημειώσεων του μLAB) να γίνει απευθείας ανάγνωση του πληκτρολογίου χωρίς τη χρήση της ρουτίνας KIND. Το αποτέλεσμα του κωδικού (βάσει του πίνακα 1 της σελ. 74) να εμφανίζεται στα 2 αριστερότερα 7-segment display με βάση τις ρουτίνες DCD (Display Character Decoder) και STDN (Store Display Message), σελ. 80-82.

Σημ.: Δεν χρειάζεται η εύρεση μόνο του κωδικού του πλήκτρου HDWR STEP λόγω της μη υλοποίησης της λειτουργίας του στον προσομοιωτή.

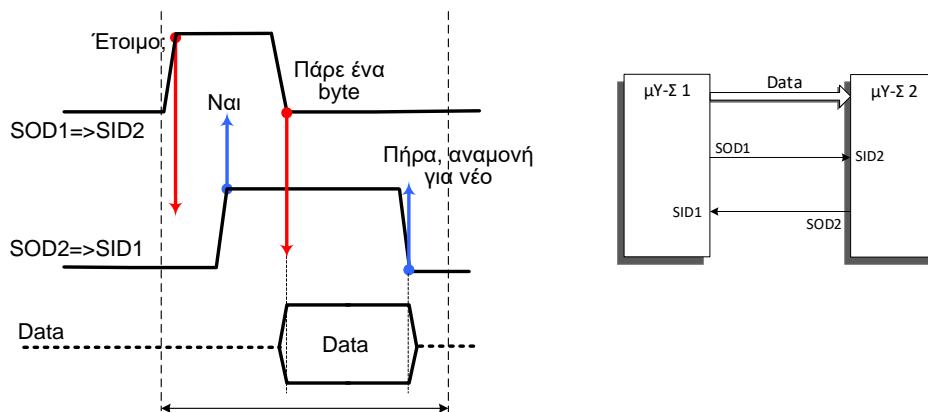
- iv. Να γίνεται η απεικόνιση των κωδικών της προηγούμενης άσκησης με την χρήση της τεχνικής απευθείας απεικόνισης χωρίς τη χρήση των ρουτινών DCD και STDN όπως περιγράφεται στις σελ. 83 – 87 (των σημειώσεων του μLAB).

Να σημειωθεί ότι χρειάζεται να δίνεται στην αρχή ενός προγράμματος η εντολή IN 10, που αίρει την προστασία της μνήμης του εκπαιδευτικού συστήματος μLAB , επιτρέποντας έτσι πρόσβαση για αποθήκευση μεταβλητών και δεδομένων οπουδήποτε στην διαθέσιμη μνήμη RAM του συστήματος (0800 – 0BFF Hex) βλ. χάρτη μνήμης μLAB - σελ. 7 των σημειώσεων - *Εισαγωγή στο Εκπαιδευτικό Σύστημα μLAB* .

Θεωρητικές Ασκήσεις

5^η ΑΣΚΗΣΗ: Σχεδιάστε ένα σύστημα που υλοποιεί μεταφορά δεδομένων από ένα $\mu\text{Y}-\Sigma$ 8085 σε ένα άλλο επίσης $\mu\text{Y}-\Sigma$ 8085 που η “χειραγία” (handsake) να βασίζεται στη χρήση των γραμμών σειριακής E/E (SID και SOD). Το $\mu\text{Y}-\Sigma$ 1 από την πόρτα εξόδου DATA1 στέλνει δεδομένα (κάθε φορά 1 byte) στην πόρτα εισόδου DATA2 του άλλου ($\mu\text{Y}-\Sigma$ 2). Για κάθε δεδομένο το $\mu\text{Y}-\Sigma$ 1 μέσω της σειριακής εξόδου SOD1 που συνδέεται στην είσοδο SID2 του $\mu\text{Y}-\Sigma$ 2 δημιουργεί μια θετική ακμή. Μόλις (το $\mu\text{Y}-\Sigma$ 2) το αναγνωρίζει, θέτει τη δική του γραμμή SOD2, περιμένει να μηδενιστεί το σήμα SOD1 (δυνατότητας λήψης του δεδομένου) και διαβάζει το δεδομένο. Το δεδομένο αυτό το αποθηκεύει στη μνήμη του με βάση το δικό του καταχωρητή H-L. Στη συνέχεια μηδενίζει το σήμα SOD2 για νέο κύκλο μεταφοράς δεδομένου.

Να δοθούν τα προγράμματα Assembly και στους δυο μE που επιτρέπουν την μεταφορά 256 δεδομένων που βρίσκονται στη μνήμη του $\mu\text{Y}-\Sigma$ 1 με αρχή τη θέση που δείχνει ο καταχωρητής H-L. Αποθηκεύονται στη μνήμη του $\mu\text{Y}-\Sigma$ 2 με βάση το δικό του καταχωρητή H-L. Η μικρότερη τιμή από τις 256 να αποθηκευτεί στον καταχωρητή C του $\mu\text{Y}-\Sigma$ 2.



6^η ΑΣΚΗΣΗ: Σχεδιάστε ένα $\mu\text{Y}-\Sigma$ 8085 που να έχει τον εξής χάρτη μνήμης:

0000-1FFF Hex	: ROM (8Kbytes)
2000-4FFF Hex	: RAM (12Kbytes)
5000-6FFF Hex	: ROM (8Kbytes)
7000 Hex	: θύρα εξόδου (Memory map I/O)
70 Hex	: θύρα εισόδου (Standard I/O)

Παρέχονται ολοκληρωμένα κυκλώματα: ROMs των 16KBytes, RAMs 4Kbytes, μE 8085, καταχωρητές και απομονωτές (απλής και διπλής κατεύθυνσης) των 8 bits, κωδικοποιητές 3 σε 8 και οι βασικές λογικές πύλες. Χρησιμοποιείτε μόνο όσα σας χρειάζονται και στην ποσότητα που θέλετε. Σχεδιάστε τα κυκλώματα διασύνδεσης. Τα βασικά σήματα ελέγχου του 8085 είναι $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\text{IO} / \overline{\text{M}}$ και ALE.

7^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 ADDR, L, K, η οποία γεμίζει ένα τμήμα μνήμης με μια σταθερά K. Το μέγεθος του τμήματος μπορεί να είναι από 1 έως 256. Η αρχική διεύθυνση ADDR, το μήκος (L) και η σταθερά K καθορίζονται από τις παραμέτρους της μακροεντολής. Θεωρείστε για (L)=0 το μέγεθος του τμήματος ίσο με 256.

γ) Επίσης δώστε τη μακροεντολή RHLL n, που περιστρέφει τα περιεχόμενα του κρατούμενου CY, των καταχωρητών H και L κατά n ψηφία αριστερά. Έτσι η μακροεντολή συμπεριφέρεται στα CY, H και L σαν να είναι ένας 17-bit καταχωρητής με το CY σαν το πιο σημαντικό ψηφίο.

Παρατήρηση: Όλα τα προγράμματα να συνοδεύονται υποχρεωτικά στα κυριότερα σημεία τους από πολύ σύντομα σχόλια.