

ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

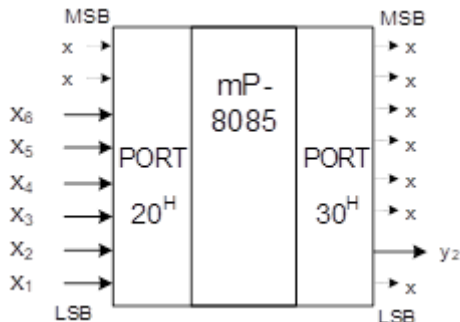
(ΘΕΜΑ 1^ο – ΣΥΝΟΛΟ 3.5 Μονάδες)

Έναρξη 11:30 - ΔΙΑΡΚΕΙΑ 50' + 10' Παράδοση: 12:30'

ΟΝΟΜΑΤΕΠΩΝΥΜΟ:

ΘΕΜΑ 1α: (1.5 ΜΟΝΑΔΕΣ):

Δίνεται μΥ-Σ που διαθέτει δύο 8-bit θύρες: μία εισόδου (διεύθ. 20^{HEX}) και μία εξόδου (διεύθ. 30^{HEX}). Να γραφεί πρόγραμμα assembly σε 8085 που να υπολογίζει τη λογική συνάρτηση $y_2 = x_1 \cdot x_2 \cdot x_3 \cdot x_4 + x_5 \cdot x_6$.

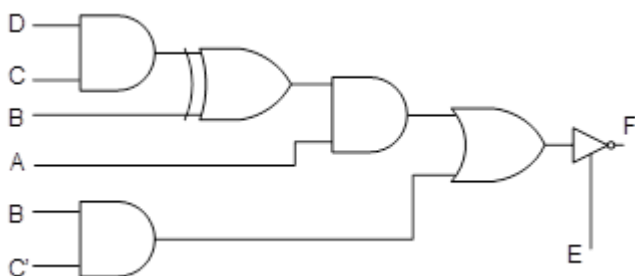


ΘΕΜΑ 1β: (1.3 ΜΟΝΑΔΑ): Απαντήστε στα παρακάτω ερωτήματα (σύντομα και αιτιολογημένα):

- (i) Δώστε τη μακροεντολή *MOVING n* που μετακινεί το περιεχόμενο ενός εκ των καταχωρητών *B, C, D, E* στον καταχωρητή *A*, για $n = 0, 1, 2, 3$ αντίστοιχα. Για άλλη τιμή του n να μην κάνει καμία λειτουργία. (0.5 ΜΟΝΑΔΕΣ)
- (ii) Να αναφέρετε τα πλεονεκτήματα που παρέχουν οι διακοπές στα μΥ-Σ. Τί πρόβλημα μπορεί να προκύψει αν μια διακοπή προκαλείται από παλμό μεγάλης ή και μικρής διάρκειας και γιατί; Να προτείνετε λύσεις για την αποφυγή των ενδεχόμενων προβλημάτων. (0.4 ΜΟΝΑΔΕΣ)
- (iii) Εξηγήστε τη λειτουργική διαφορά των καθυστερήσεων που προκαλούνται μέσω ρουτινών χρονοκαθυστέρησης και μέσω μετρητών-χρονιστών (πλεονεκτήματα, μειονεκτήματα). (0.2 ΜΟΝΑΔΕΣ)
- (iv) Πότε είναι χρήσιμη και πλεονεκτική η χρήση των Μακροεντολών σε σχέση με τις Ρουτίνες; (0.2 ΜΟΝΑΔΕΣ)

ΘΕΜΑ 1γ: (0.7 ΜΟΝΑΔΕΣ):

Δώστε την περιγραφή Verilog του παρακάτω κυκλώματος σε **επίπεδο πυλών** και σε μορφή **ροής δεδομένων**.



ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

(ΘΕΜΑ 2^ο – ΣΥΝΟΛΟ 4.5 Μονάδες)

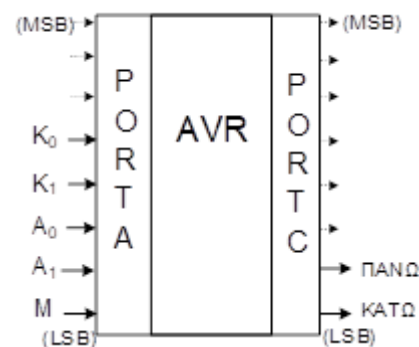
Έναρξη 12:30 - ΔΙΑΡΚΕΙΑ 60' + 10' Παράδοση: 13:40'

ΟΝΟΜΑΤΕΠΩΝΥΜΟ:

ΘΕΜΑ 2ο: (4.5 ΜΟΝΑΔΕΣ): Σε ένα μικροελεγκτή AVR Mega16 που αξιοποιεί μία θύρα εισόδου και μία εξόδου, όπως φαίνεται στο διπλανό σχήμα, να υλοποιηθεί ένα σύστημα οδήγησης ενός ανελκυστήρα δυο θέσεων (ισογείου και 1^{ου} ορόφου). Η κίνηση προς το ισόγειο ή τον 1^ο όροφο ελέγχεται από τους εξωτερικούς διακόπτες (Push-Buttons) K_0 και K_1 αντίστοιχα καθώς και από έναν εσωτερικό διακόπτη (Push-Button) M . Για να δοθεί εντολή από τους διακόπτες αυτούς, προϋπόθεση είναι το βαγόνι να είναι σταματημένο στο ισόγειο ή στον 1^ο όροφο. Όταν κινείται πρέπει να σταματάει από το πρόγραμμα με βάση τους αισθητήρες A_0 και A_1 που είναι τερματικοί διακόπτες και οι οποίοι δίνουν λογικό 1 αυτόματα όταν ο θάλαμος φτάνει στο ισόγειο ή στον 1^ο όροφο αντίστοιχα. Υποθέτουμε ότι κατά την εκκίνηση του συστήματος, ο θάλαμος πρέπει να βρίσκεται στο ισόγειο, αλλιώς πριν δεχτεί οποιαδήποτε εντολή να μεταφέρεται σε αυτή τη θέση αυτόματα.

Αναλυτικά, αν ο θάλαμος φτάσει στο ισόγειο, τότε πρέπει να σταματάει η κίνησή του και να ελέγχονται οι διακόπτες K_1 και M . Αν ένας από αυτούς είναι ενεργοποιημένος (=1) τότε έχουμε κίνηση προς τα πάνω. Αντίστοιχα αν ο θάλαμος φτάσει στον 1^ο όροφο, τότε πρέπει να σταματάει η κίνησή του και να ελέγχονται οι διακόπτες K_0 και M . Αν ένας από αυτούς είναι ενεργοποιημένος (=1) τότε έχουμε κίνηση προς τα κάτω. Δώστε το αντίστοιχο πρόγραμμα σε assembly και σε C.

(Assembly: 2.5 ΜΟΝΑΔΕΣ και C: 2 ΜΟΝΑΔΕΣ)



ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

(ΘΕΜΑ 3^ο – ΣΥΝΟΛΟ 2 Μονάδες)

Έναρξη 13:40' - ΔΙΑΡΚΕΙΑ 30' + 10' Παράδοση: 14:20'

ΟΝΟΜΑΤΕΠΩΝΥΜΟ:

ΘΕΜΑ 3ο: (2 ΜΟΝΑΔΕΣ): Σε ένα προσωπικό υπολογιστή, να γραφεί πρόγραμμα σε Assembly με 80x86 που να δέχεται από το πληκτρολόγιο τέσσερις (4) δεκαδικούς αριθμούς (D_3, D_2, D_1, D_0 με τη σειρά αυτή) για να αποτελέσουν ένα διψήφιο και δυο μονοψήφιους δεκαδικούς αριθμούς και να κάνει τον εξής υπολογισμό: $P = (D_3 \times 10 + D_2) \times (D_1 + D_0)$. Το πρόγραμμα τυπώνει στην οθόνη τα μηνύματα εισόδου και τους εισαγόμενους αριθμούς. Όταν συμπληρωθούν 4 έγκυροι δεκαδικοί αριθμοί να αναμένει τον χαρακτήρα 'h' και μετά να τυπώνει το αποτέλεσμα σε δεκαεξαδική μορφή 3 ψηφίων αν είναι <400Hex, αλλιώς το μήνυμα yperx, αυστηρά όπως φαίνεται παρακάτω:

DOSE 1ο ARITHMO = 58

DOSE 2ο ARITHMO = 7

DOSE 3ο ARITHMO = 9

APOTELESMA = 3A0 ή APOTELESMA = yperx

Να θεωρήσετε δεδομένες τις μακροεντολές (σελ. 361-2, 373) του βιβλίου και μπορείτε να κάνετε χρήση των ρουτινών DEC_KEYB και PRINT_HEX χωρίς να συμπεριλάβετε τον κώδικά τους. Για την διευκόλυνσή σας, δίνονται οι πρώτες εντολές που αποτελούν τον 'σκελετό' του ζητούμενου προγράμματος.

ΑΠΑΝΤΗΣΗ

INCLUDE MACROS

DATA_SEG SEGMENT

MSG1 DB 0AH,0DH, 'DOSE 1ο ARITHMO = \$'

MSG2 DB 0AH,0DH, 'DOSE 2ο ARITHMO = \$'

MSG3 DB 0AH,0DH, 'DOSE 3ο ARITHMO = \$'

MSG4 DB 0AH,0DH, 'APOTELESMA = \$'

DATA_SEG ENDS

CODE_SEG SEGMENT

ASSUME CS:CODE_SEG, DS:DATA_SEG

MAIN PROC FAR

MOV AX, DATA_SEG

MOV DS, AX

...