

ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

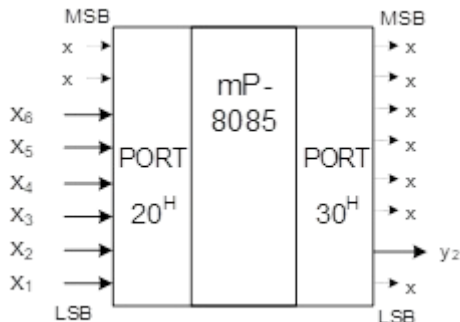
(ΘΕΜΑ 1^ο – ΣΥΝΟΛΟ 3.5 Μονάδες)

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

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΓΙΩΡΓΟΣ ΚΥΡΙΑΚΟΠΟΥΛΟΣ - el18153

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 $y_2 = x_1 \cdot x_2 \cdot x_3 \cdot x_4 + x_5 \cdot x_6$.

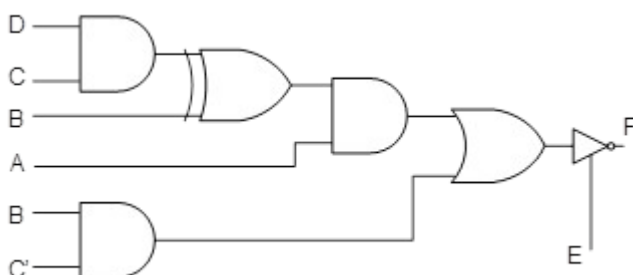


ΘΕΜΑ 1β: (1.3 ΜΟΝΑΔΑ): Απαντήστε στα παρακάτω ερωτήματα (σύντομα και αιτιολογημένα):

- (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**.



1α)

START:

```
LDA 2000H
MOV B, A
MOV C, A
RAR
ANA C
MOV C, A
```

```
MOV A, B
RAR
RAR
ANA C
MOV C, A
```

```
MOV A, B
RAR
RAR
RAR
ANA C
MOV C, A ; C(LSB) = x1x2x3x4
```

```
MOV A, B
RAR
RAR
RAR
RAR
MOV D, A
RAR
ANA D ; A(LSB) = x5x6
```

```
ORA C ; A(LSB) = x1x2x3x4 + x5x6
ANI 01H
RAL
```

```
STA 3000H
```

```
JMP START
```

```
END
```

1β1)

MOVING MACRO N

```
MVI A, N ; We only use A and return it so nothing to push/pop
```

```
CPI 00H
JZ MOVBA
CPI 01H
JZ MOVCA
CPI 02H
JZ MOVDA
CPI 03H
JZ MOVEA
JMP EXIT
```

MOVBA:

```
MOV A, B
JMP EXIT
```

MOVCA:

```
MOV A,
JMP EXIT
```

MOVDA:

```

MOV A,D
JMP EXIT
MOVEA:
MOV A,E
JMP EXIT
EXIT: ENDM

```

1γ)

```

module verilog (F, A, B, C, D, E);
  output F;
  input A, B, C, D, E;
  wire w1, w2, w3, w4, w5, w6;

  not G1 (w1, C);
  and
    G2 (w2, D, C),
    G3 (w3, B, w1);

  xor G4 (w4, w2, B);
  and G5 (w5, A, w4);
  or G6 (w6, w5, w3);
  notif(F, w6, E);
endmodule

```

Μοντελοποίησης

```

module verilog (F, A, B, C, D, E);
  output F;
  input A, B, C, D, E;

  assign F = (E)?(~((((D&C)^B)&A)|(B&(~C)))):1'bz;
endmodule

```

1β4)

Με τη χρήση ρουτινών έχουμε οικονομία στη μνήμη. Επίσης με χρήση μακροεντολών οι εντολές τους εισέρχονται στο κυρίως πρόγραμμα πριν το χρόνο μετάφρασης ενώ με τις ρουτίνες αυτό γίνεται κατά την εκτέλεση του προγράμματος. Από άποψη ταχύτητας οι μακροεντολές δίνουν καλύτερα αποτελέσματα αφού δεν επιβαρύνουν το πρόγραμμα με εντολές κλήσης και επιστροφής ρουτινών.

1β2)

Διαχείριση I/O με καλύτερο τρόπο.

Καλύτερη εκμετάλλευση του χρόνου του με δηλαδή υπολογιστικής ισχύος.

Ασύγχρονη ανταπόκριση.