

ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

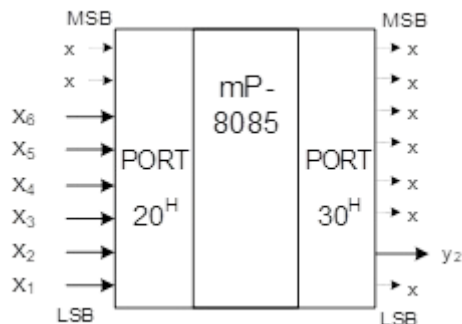
(ΘΕΜΑ 1^ο – ΣΥΝΟΛΟ 3.5 Μονάδες)

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

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΣΤΕΦΑΝΑΚΗΣ ΓΕΩΡΓΙΟΣ el18436

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



IN 20H

MOV B,A ; Save input to B TO REUSE

ANI 01H ; MASK LSB

MOV C,A ; C = Bit(0)

MOV A,B ; A = input

RRC ; rotate right

ANI 01H ; MASK

ANA C ; A AND C

MOV C,A ; C = x1*x2

MOV A,B ; A = input

RRC

RRC

ANI 01H

ANA C ; A = x1*x2*x3

MOV C,A ; C = x1*x2*x3

MOV A,B

RRC

RRC

RRC

ANI 01H

ANA C ; A = x1*x2*x3*x4

MOV C,A ; C = x1*x2*x3*x4

MOV A,B

RRC

RRC

RRC

RRC

ANI 01H ; A = x5

MOV D,A ; D = x5

MOV A,B

RRC

RRC

RRC

RRC

RRC

ANI 01H ; A = x6
ANA D ; A = x5*x6
ORA C ; A OR C = x5*x6 || x1*x2*x3*x4
RLC ; ROTATE TO BRING BIT TO y2
OUT 30H

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)

MOVING MACRO n

MOV A,n ; A =n

CPI 00H ; if n !=0 jump to next check

JNZ MOVC ; if n = 0

MOV A,B ; A = B

JUMP ENDMAC ; jump to label to end macroinstruction

MOVC: CPI 01H

JNZ MOVD

MOV A,C

JUMP ENDMAC

MOVD: CPI 02H

JNZ MOVE

MOV A,D

JMP ENDMAC

MOVE: CPI 03H

JNZ ENDMAC

MOV A,E

ENDMAC:

END ; Χρειάζεται να γίνουν push τα flags με καποιον τροπο που δεν προλαβα να υλοποιησω

(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)

1) Αυξάνει την επίδοση ενός επεξεργαστή (Διαχείριση εισόδου εξόδου με αποδοτικό τρόπο)

2) Μειώνει τον χρόνο αναμονής του επεξεργαστή (καλύτερη εκμετάλλευση του χρόνου το μικροεπεξεργαστή)

3) Ασύγχρονη ανταπόκριση (η διακοπή ψάχνει για το σήμα εισόδου/εξόδου μέσω hardware και μόλις γίνει η ανταπόκριση συνεχίζεται η εντολή που τρέχει μέχρι να ολοκληρωθεί)

(iii) Εξηγήστε τη λειτουργική διαφορά των καθυστερήσεων που προκαλούνται μέσω ρουτινών χρονοκαθυστέρησης και μέσω μετρητών-χρονιστών (πλεονεκτήματα, μειονεκτήματα). (0.2 ΜΟΝΑΔΕΣ)

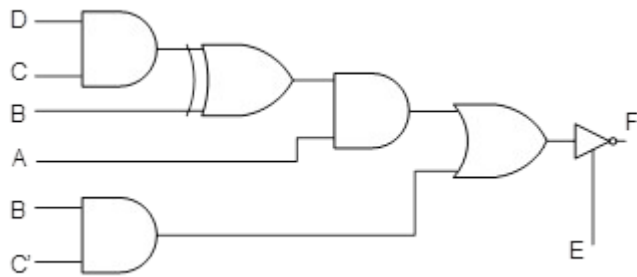
(iv) Πότε είναι χρήσιμη και πλεονεκτική η χρήση των Μακροεντολών σε σχέση με τις Ρουτίνες; (0.2 ΜΟΝΑΔΕΣ)

1) Χρονος εκτέλεσης macro μικροτερος απο χρονο εκτέλεσης ρουτινας καθώς οι εντολές εισέρχονται στο κυρίως πρόγραμμα πριν το χρόνο μετάφρασης σε αντίθεση με τις ρουτίνες που εισέρχονται κατά τον χρόνο εκτέλεσης

2) Τμηματικός προγραμματισμός: Για μακροεντολές, ένα κομμάτι κωδικα χρησιμοποιείται πολλές φορές όταν χρειαστεί και δεν έχουμε επαναληψη κωδικα (κερδίζουμε σε μνημη)

ΘΕΜΑ 1γ: (0.7 ΜΟΝΑΔΕΣ):

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



```
module thema_1g_gates (F, A, B, C, D, E);
```

```
    Input A, B, C, D, E;
```

```
    Output F;
```

```
    wire x, y, z, w, l, k;
```

```
    not (x, C); // x = not C
```

```
    and (y, x, B);
```

```
    and (z, C, D);
```

```
    xor (w, z, B);
```

```
    and (l, w, A);
```

```
    or (k, l, y);
```

```
    notif1(F, k, E);
```

```
endmodule
```

```
module thema_1g_dataflow (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
```