

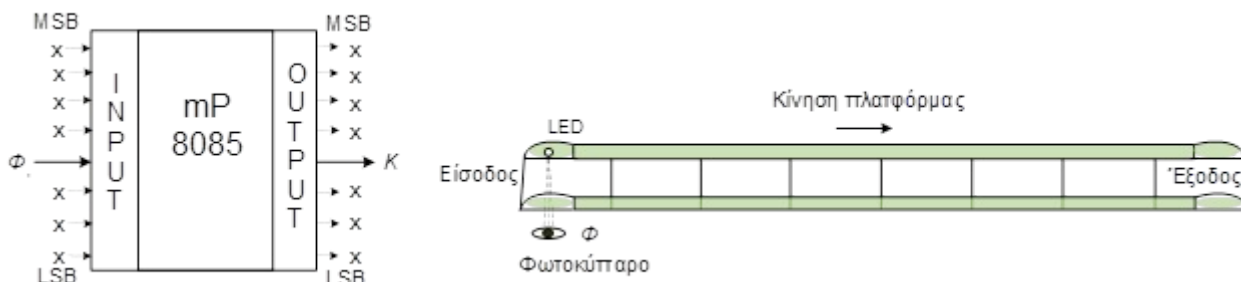
ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΣΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

(ΘΕΜΑ 1^ο – ΣΥΝΟΛΟ 3.5 Μονάδες)

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

ΟΝΟΜΑΤΕΠΩΝΥΜΟ:

ΘΕΜΑ 1α: (2 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 υλοποιηθεί ένα σύστημα οδήγησης κυλιόμενης πλατφόρμας μονής κατεύθυνσης η οποία να ενεργοποιείται από το φωτοκύτταρο Φ . Συγκεκριμένα, αν ένας επιβάτης εισέρχεται στην πλατφόρμα, όταν είναι ακίνητη, διακόπτει δέσμη φωτός (γίνεται $\Phi=0$) και τότε τίθεται σε κίνηση η πλατφόρμα με το σήμα εξόδου K (για $K=1$ έχουμε κίνηση). Η κίνηση να σταματά ~ 10 sec μετά την τελευταία διακοπή του φωτοκυττάρου Φ (χρόνος ικανός για να αδειάσει η πλατφόρμα από επιβάτες). Μπορείτε να κάνετε χρήση της ρουτίνας χρονοκαθυστέρησης DSEC των 40 msec.



START:

```
IN 20H
MOV B, A
ANI 8H ; Input & 0000 1000
CPI 8H
JNZ OUTPUT
```

```
JMP NO_OUTPUT
```

NO_OUTPUT:

```
MVI A, 00H
OUT 30H
JMP START
```

;; count 250 times – FA times

OUTPUT:

```
MVI A, 8H
OUT 30H
MVI B, 0H
JMP COUNT_250
```

COUNT_250:

```
MOV A,B
CPI FAH
JZ NO_OUTPUT
```

```
CALL DSEC
```

```
INR B
```

```
JMP COUNT_250
```

ΘΕΜΑ 1β: (0.8 ΜΟΝΑΔΕΣ): Απαντήστε στα παρακάτω ερωτήματα (σύντομα και αιτιολογημένα):

(i) Δώστε τη μακροεντολή *MIN* που μετακινεί τον ελάχιστον των καταχωρητών *D*, *C*, *L* στον καταχωρητή *A*, χωρίς να επηρεάζεται η τιμή των καταχωρητών (πλην φυσικά του *A*). (0.4 ΜΟΝΑΔΕΣ)

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

(iii) Πώς λειτουργούν οι εντολές σχετικού άλματος και κλήσης ρουτίνας; Εξηγήστε την χρησιμότητά τους.

(0.2 ΜΟΝΑΔΕΣ)

```
i)
MOVING MACRO MIN
MVI A, D
CMP C
JNC E0 ;; C > D
;; ELSE C IS MIN (C,D)
JMP E1
E0:
    MOV C, D
    JMP E1
E1:
    MVI A,C
    CMP L
    JNC E2 ;; L > C
    MOV A, L
E2:
    MOV A, C

EDNM
```

ii) Η μακροεντολή ορίζει μια συγκεκριμένη λειτουργία, ορίζεται σε ένα μπλόκ κώδικα. Ουσιαστικά όταν γίνεται κλήση της τότε αντικαθιστάτε η μακροεντολή από το κείμενο που έχει οριστεί. Με αυτόν τον τρόπο γλυτώνουμε κόστος αποθήκευσης και αλλαγής του PC(Program counter). Αντίθετα, οι ρουτίνες χρησιμοποιούν στοίβα για αποθήκευσης της προηγούμενης θέσης του PC. Οπότε, οι μακροεντολές συνιστούν στην επιτάχυνση της εκτέλεσης.

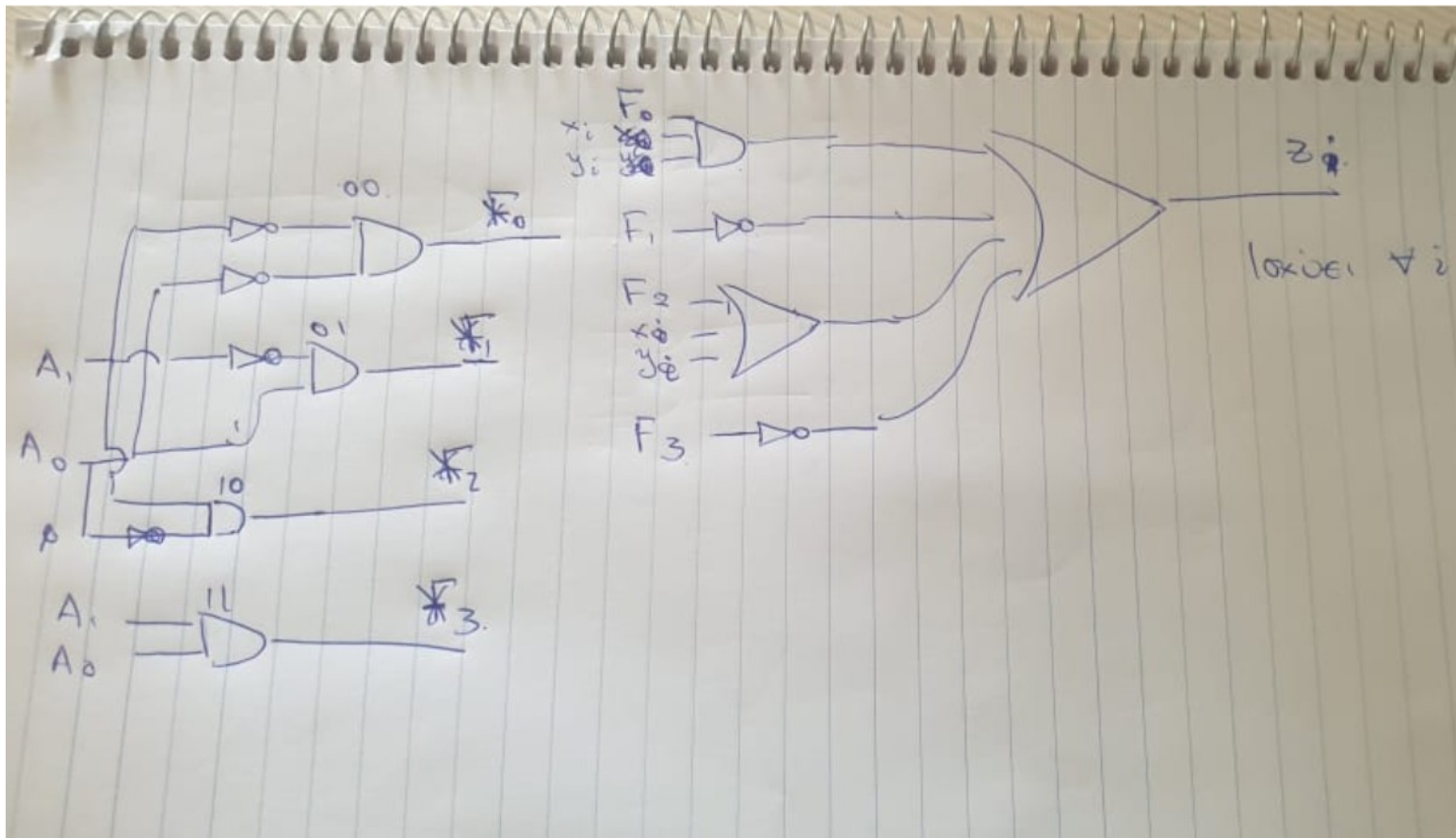
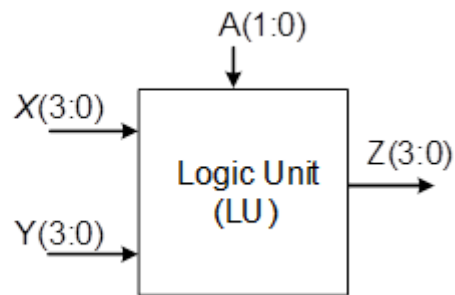
iii)

ΘΕΜΑ 1γ: (0.7 ΜΟΝΑΔΕΣ):

Να δοθεί το κύκλωμα (σχηματικό διάγραμμα) και η δομική περιγραφή σε Verilog της μονάδας LU που η λειτουργία της φαίνεται στο διπλανό πίνακα και σχήμα. Μπορείτε να κάνετε χρήση των βασικών πυλών: XOR(*z,x,y*), OR(*z,x,y*), AND(*z,x,y*) και INV(*z,x*) θεωρώντας τις μεταβλητές *a* και *b* ως εισόδους. Το σύμβολο ‘∨’ δηλώνει την πράξη OR ενώ το ‘·’ την πράξη AND. Υποθέτουμε ότι οι μεταβλητές *X*, *Y* και *Z* είναι των 4-bit. Η λέξη ελέγχου *A*(1:0) είναι των 2-bit.

Επίσης να δοθεί η περιγραφή Verilog του ίδιου κυκλώματος σε μορφή ροής δεδομένων ή σε μοντελοποίηση συμπεριφοράς.

A_1A_0	Έξοδος Z
00	$X \cdot Y$
01	'1111'
10	$X \vee Y$
11	'0000'



```

module Circuit 1_g_gates(z0,z1,z2,z3, a0,a1, x0,x1,x2,x3, y0,y1,y2,y3);
    output z0;
    output z1;
    output z2;
    output z3;
    input a0,a1, x0,x1,x2,x3, y0,y1,y2,y3;
    wire w1,w2,w3,w4,w5,w6,w7,w8,w9,w10;

    inv g1(w1, a0)
    inv g2(w2, a1)
    and g3(w3,w1,w2) ; // 00
    and g4(w4,w1,a0) ; // 01
    and g5(w5,a1,w2) ; // 10
    and g6(w6,a1,a0) ; // 11

    and g7(w7,x0,y0);
    and g8(w8,x1,y1); ;
    and g9(w9,x2,y2);
    and g10(w10,x3,y3);

    // ola ta F0 xi yi
    and g11(w11, w7, w3);
    and g12(w12, w8, w3);
    and g13(w13, w9, w3);

```

```
and g14(w14, w10, w3);
```

```
or g15(w15, x0,y0 );  
or g16(w16, x1,y1 );  
or g17(w17, x2,y2 );  
or g18(w18, x3,y3 );
```

```
// ola ta F2, xi, yi  
or g19(w19, w15, w5);  
or g20(w20, w16, w6);  
or g21(w21, w17, w7);  
or g22(w22, w18, w8);
```

```
inv g23(w23, w4);  
inv g24(w24, w6);
```

```
// afto gia to z0  
or g21(w25, w11, w19);  
or g22(w26, w23, w25);  
or g23(w27, w26, w19);  
or g24(w28,w27, w24);  
notifl(z0, w28);
```

```
// z1  
or g21(w28, w11, w19);  
or g22(w30, w23, w28);  
or g23(w31, w30 w19);  
or g24(w32,w31, w24);  
notifl(z1, w32);
```

```
// z2  
or g21(w33, w11, w19);  
or g22(w34, w23, w25);  
or g23(w35, w26, w19);  
or g24(w36,w27, w24);  
notifl(z2, w36);
```

```
// z3  
or g21(w37, w11, w19);  
or g22(w38, w23, w25);  
or g23(w39, w26, w19);  
or g24(w40,w27, w24);  
notifl(z3, w40);
```

```
endmodule
```