

Εργαστήριο Προηγμένα Μικροϋπολογιστικά Συστήματα

2^η Εργαστηριακή Άσκηση

Ομάδα Β3

Κουκιάσας Δημήτριος 1072650

Μπασαγιάννη Γεωργία 1084016

Η συγκεκριμένη εργαστηριακή άσκηση πραγματοποιήθηκε με τη χρήση του μικροϋπολογιστικού συστήματος (kit) FLIGHT 8086 όπως και στην προηγούμενη άσκηση.

Από τη θέση \01-8086 Flight-86 Training System\00-Flight-86 Manual του CD εργαστηρίου Micro II, εκτελέστηκε το παράδειγμα 4 από το manual :

“03-FLIGHT (APPLICATIONS BOARD-EXPERIMENT MANUAL)”.

Για το παράδειγμα 4, έπρεπε να ρυθμίσουμε τα φανάρια κυκλοφορίας μίας διασταύρωσης ενός κύριου δρόμου και ενός παράδρομου. Το φανάρι του κύριου δρόμου είναι πράσινο για 35sec, ενώ το φανάρι του παράδρομου είναι πράσινο για 12sec. Και τα δύο φανάρια είναι πορτοκαλί για 4sec.

Για την επίλυση του προβλήματος δημιουργήσαμε 4+1 καταστάσεις.

1^η : ΚΔ-ΠΡΑΣΙΝΟ ΠΔ-ΚΟΚΚΙΝΟ

2^η : ΚΔ-ΠΟΡΤΟΚΑΛΙ ΠΔ-ΚΟΚΚΙΝΟ

3^η : ΚΔ-ΚΟΚΚΙΝΟ ΠΔ-ΠΡΑΣΙΝΟ

4^η : ΚΔ-ΚΟΚΚΙΝΟ ΠΔ-ΠΟΡΤΟΚΑΛΙ

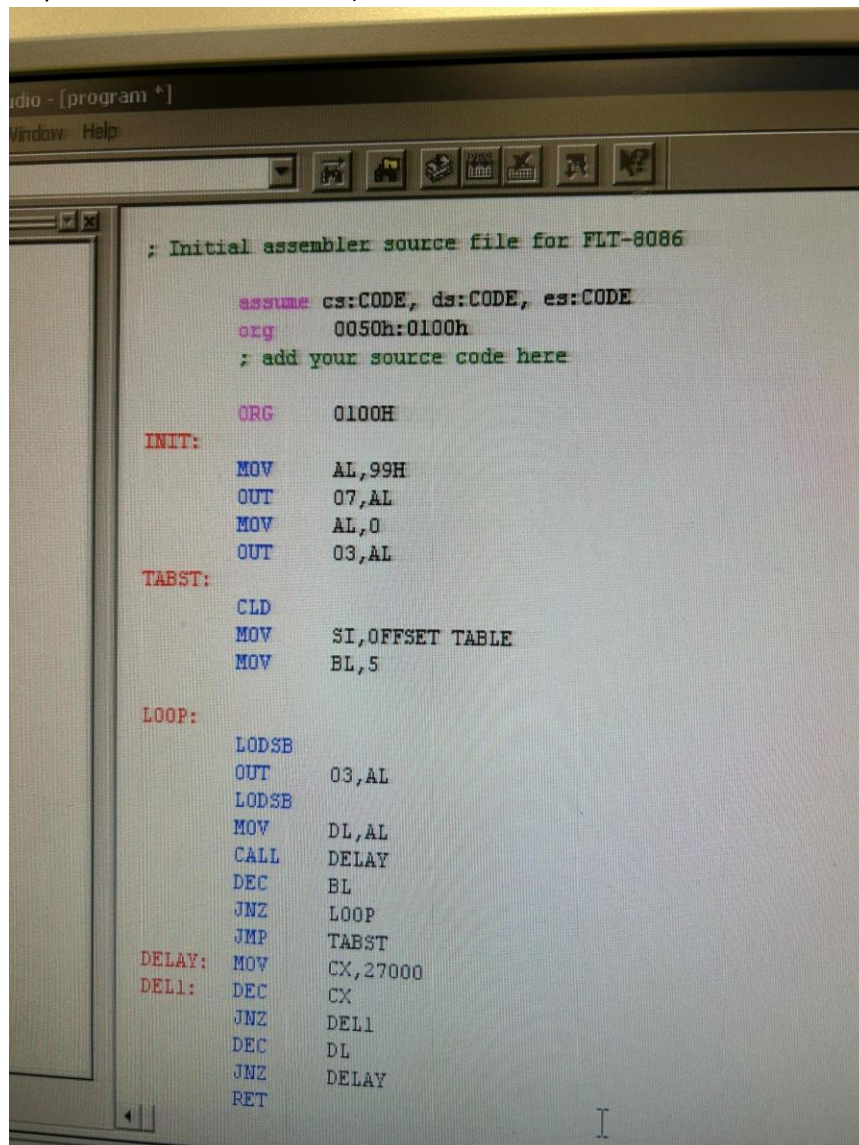
Ωστόσο, χρειάστηκε να «σπάσουμε» την 1^η κατάσταση σε 2 καταστάσεις καθώς η καθυστέρηση στην αλλαγή των χρωμάτων των φαναριών γίνεται μέσω DB(Delay Byte) και $2^8=256<350$ (που πρέπει να μπει για το πράσινο φανάρι του ΚΔ, και $350=35*10$ καθώς έτσι έχει ριστεί το delay στον κώδικα).

Άρα, οι τελικές καταστάσεις με τους χρόνους καθυστέρησης τους είναι :

1 ^η : ΚΔ-ΠΡΑΣΙΝΟ	ΠΔ-ΚΟΚΚΙΝΟ	20sec
2 ^η : ΚΔ-ΠΡΑΣΙΝΟ	ΠΔ-ΚΟΚΚΙΝΟ	15sec
3 ^η : ΚΔ-ΠΟΡΤΟΚΑΛΙ	ΠΔ-ΚΟΚΚΙΝΟ	4sec
4 ^η : ΚΔ-ΚΟΚΚΙΝΟ	ΠΔ-ΠΡΑΣΙΝΟ	12sec
5 ^η : ΚΔ-ΚΟΚΚΙΝΟ	ΠΔ-ΠΟΡΤΟΚΑΛΙ	4sec

Οι επιμέρους καθυστερήσεις δεν έχουν συμπεριληφθεί όπως μας ζητήθηκε (όχι πρότυπο Αγγλίας).

Παρακάτω δίνεται ο κώδικας :



```
; Initial assembler source file for FLT-8086

assume cs:CODE, ds:CODE, es:CODE
org 0050h:0100h
; add your source code here

INIT:
    ORG 0100H
    MOV AL,99H
    OUT 07,AL
    MOV AL,0
    OUT 03,AL

TABST:
    CLD
    MOV SI,OFFSET TABLE
    MOV BL,5

LOOP:
    LODSB
    OUT 03,AL
    LODSB
    MOV DL,AL
    CALL DELAY
    DEC BL
    JNZ LOOP
    JMP TABST

DELAY:
    MOV CX,27000
DELL:
    DEC CX
    JNZ DELL
    DEC DL
    JNZ DELAY
    RET
```

```
ment Studio - [program *.]
Tools Window Help

OUT    07,AL
MOV    AL,0
OUT    03,AL

TABST:
      CLD
      MOV    SI,OFFSET TABLE
      MOV    BL,5

LOOP:
      LODSB
      OUT    03,AL
      LODSB
      MOV    DL,AL
      CALL   DELAY
      DEC    BL
      JNZ    LOOP
      JMP    TABST

DELAY: MOV    CX,27000
DELI:  DEC    CX
      JNZ    DELI
      DEC    DL
      JNZ    DELAY
      RET

TABLE:
      DB     28H,200
      DB     28H,150
      DB     48H,40
      DB     82H,120
      DB     84H,40

      END    INIT

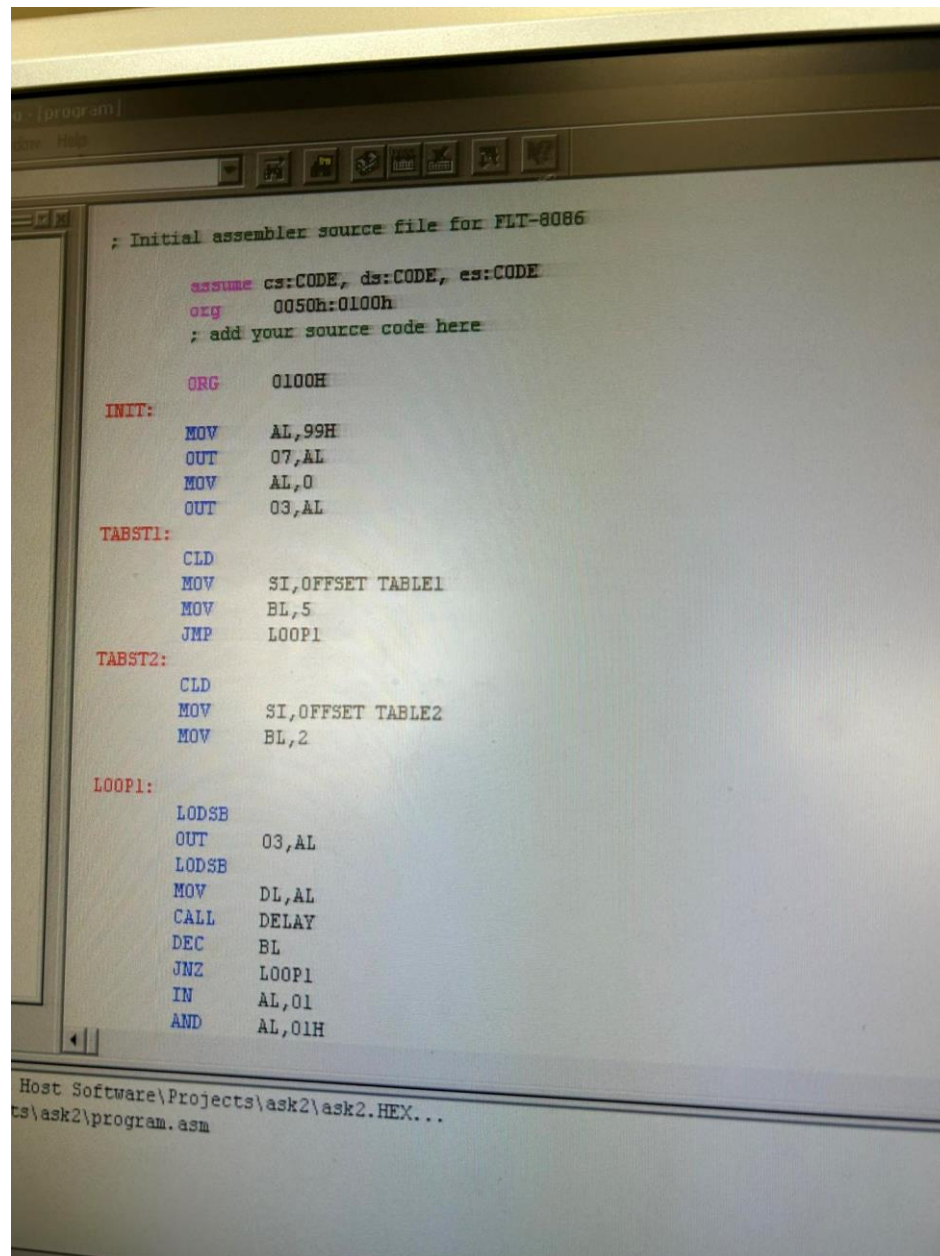
:\ESTUDIO PC Host Software\Projects\ask2\ask2.HEX...
ware\Projects\ask2\program.asm
te
```

Έτσι, είδαμε στο kit στο μέρος των 8 led, την 3αδα αριστερά ως το φανάρι του ΚΔ και την 3αδα δεξιά ως το φανάρι του ΠΔ.

Πράγματι, για 35sec μένει αναμμένο το πράσινο του ΚΔ (00101000=28H) και το κόκκινο του ΠΔ. Μετά για 4sec μένει αναμμένο το πορτοκαλί του ΚΔ (01001000=48 H) και το κόκκινο του ΠΔ. Έπειτα κόκκινο του ΚΔ (10000010=82 H) και πράσινο του ΠΔ. Τέλος, πορτοκαλί του ΠΔ και κόκκινο του ΚΔ (10000100=84 H) και πάλι από την αρχή οι καταστάσεις λόγω του loop.

Στη συνέχεια, ζητήθηκε κώδικας με switch είσοδο για να αλλάζει η λειτουργία των φαναριών σε mode μέρας και νύχτας. Στο mode μέρας γίνεται η λειτουργία του προηγούμενου παραδείγματος, ενώ στο mode νύχτας, το φανάρι του ΚΔ είναι πάντα σβηστό και το φανάρι του ΠΔ αναβοσβήνει πορτοκαλί με καθυστέρηση 2sec.

Δίνεται ο κώδικας:



```
; Initial assembler source file for FLT-8086

assume cs:CODE, ds:CODE, es:CODE
org 0050h:0100h
; add your source code here

ORG 0100H

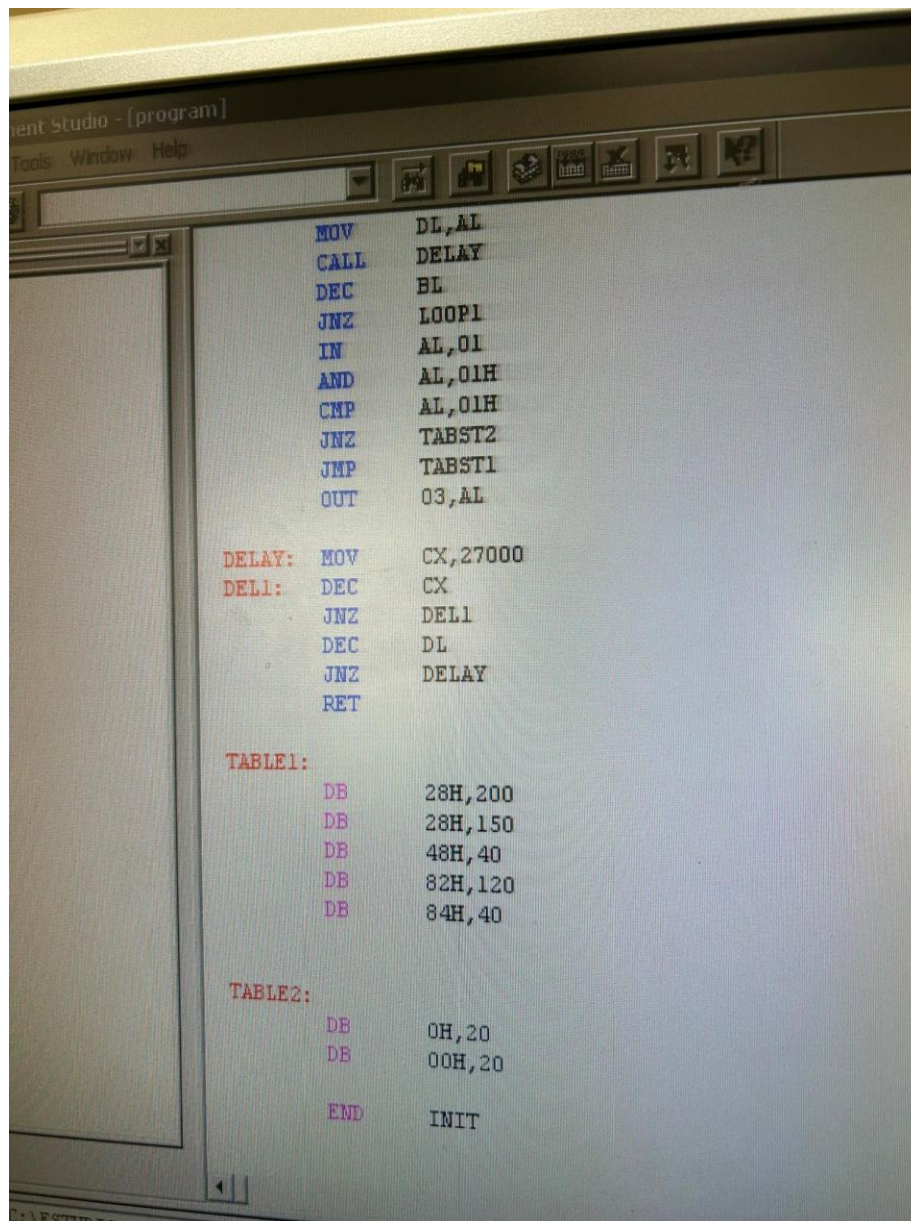
INIT:
MOV AL,99H
OUT 07,AL
MOV AL,0
OUT 03,AL

TABST1:
CLD
MOV SI,OFFSET TABLE1
MOV BL,5
JMP LOOP1

TABST2:
CLD
MOV SI,OFFSET TABLE2
MOV BL,2

LOOP1:
LODSB
OUT 03,AL
LODSB
MOV DL,AL
CALL DELAY
DEC BL
JNZ LOOP1
IN AL,01
AND AL,01H
```

Host Software\Projects\ask2\ask2.HEX...
ts\ask2\program.asm



Τώρα μετονομάσαμε τον TABLE σε TABLE1 και δημιουργήσαμε έναν επιπλέον για το mode νύχτας τον TABLE 2.

Αντίστοιχα μετονομάσαμε τον TABST1 και φτιάξαμε τον TABST2 για τον TABLE2.

Προσθέσαμε στον κώδικα της loop την είσοδο IN AL,01 για να παίρνει ως είσοδο το ζητούμενο switch και με AND AL,01H και CMP AL,01H ελέγχουμε αν είναι κατεβασμένο το τέρμα δεξιά switch.

Δυστυχώς δεν είδαμε το επιθυμητό αποτέλεσμα καθώς έχουμε βάλει JNZ TABST2, ενώ θα έπρεπε JZ, αλλιώς να κάνει JMP TABST1. Γι αυτό δεν είδαμε να αλλάζει το mode σε νύχτα.

Επίσης, υπάρχει τυπογραφικό DB 0H,20 αντί για DB 04H καθώς σβήστηκε καταλάθος στο τέλος λόγω πίεσης του χρόνου, αλλά κατά την εκτέλεση ήταν γραμμένο.

QUESTION 4

Για τη μέτρηση καθυστέρησης χρησιμοποιούμε το application board του FLIGHT 8086 και ένα stopwatch για χρονομέτρηση (πχ ένα smartphone).

Γράφουμε τον κώδικα Assembly, ώστε να περιλαμβάνει μια λογική μέτρησης χρόνου γύρω από το τμήμα κώδικα που εισάγει την καθυστέρηση. Με χρήση ενός smartwatch η εκκίνηση και ο τερματισμός του χρονομέτρου αφορούν τον χρήστη αν και ανάλογα το board μπορεί να συνδεθεί ο μετρητής στην πλακέτα. Έπειτα εκτελούμε το πρόγραμμα. Ξεκινάμε το χρονόμετρο μέσω του stopwatch λίγο πριν την εκκίνηση του μέρους του προγράμματος που αφορά την καθυστέρηση (για αποφυγή καθυστέρησης κατά την εκκίνηση ενεργοποιούμε το τμήμα κώδικα που εισάγει την καθυστέρηση). Μόλις παρατηρήσουμε την ολοκλήρωση της διαδικασίας που προκαλεί καθυστέρηση σταματάμε τη χρονομέτρηση. Τέλος, καταγράφουμε τον χρόνο που μετρήσαμε με το stopwatch και είναι η καθυστέρηση που επιθυμούσαμε να μετρήσουμε.

Οι κατάλληλες τιμές που πρέπει να αποθηκευτούν στον καταχωρητή για το delay ώστε να έχουμε καθυστέρηση 10ms είναι 2700 και για καθυστέρηση 100ms είναι 27000 (χρησιμοποιήθηκε και στον κώδικα παραπάνω). Προκύπτει από $\text{clock_cycles}(16\text{-JNP και } 2\text{-DEC}) * t (=1/f, \text{ όπου } f=14,7456/3) * \text{ZHTOYMENO} = \text{delay}(10 \text{ και } 100 \text{ ms αντίστοιχα})$.

```
DELAY: MOV     CX,27000
DEL1:  DEC     CX
        JNZ     DEL1
```

QUESTION 5

Για να μετρήσουμε λεπτά με το πρόγραμμα της ερώτησης 4 μπορούμε το DELAY loop να το βάλουμε μέσα σε ένα 2^ο loop χρησιμοποιώντας ένας καταχωρητή-μετρητή του οποίου η τιμή θα αρχικοποιηθεί στο 6000(=60/0.01) για την περίπτωση των 10ms, και στο 600(=60/0.1) για την περίπτωση των 100ms και κάθε φορά που θα τελειώνει το εσωτερικό DELAY loop θα μειώνεται η τιμή του κατά ένα και θα γίνεται έλεγχος αν έχουμε μηδέν.

QUESTION 6

Για την εγκατάσταση πεζών που διασχίζουν τον ΚΔ φτιάχνουμε έναν καινούργιο πίνακα με 7 καταστάσεις.

- | | | |
|--------------------------|----------------|-------------------------------------------------------|
| 1 ^η : LED OFF | (ΚΔ-ΠΡΑΣΙΝΟ) | 20sec |
| 2 ^η : LED OFF | (ΚΔ-ΠΡΑΣΙΝΟ) | 15sec |
| 3 ^η : LED OFF | (ΚΔ-ΠΟΡΤΟΚΑΛΙ) | 4sec |
| 4 ^η : LED ON | (ΚΔ-ΚΟΚΚΙΝΟ) | 16-5=11sec μέχρι να μείνουν τα τελευταία sec |
| 5 ^η : LED OFF | (ΚΔ-ΚΟΚΚΙΝΟ) | 1sec σβήνει ξαφνική αλλαγή |
| 6 ^η : LED ON | (ΚΔ-ΚΟΚΚΙΝΟ) | 4-2=2sec ανάβει |
| 7 ^η : LED OFF | (ΚΔ-ΚΟΚΚΙΝΟ) | 2sec τελευταία κι έτσι αναβοσβήνει τα τελευταία 5sec. |