



# **ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ ΗΜ&ΜΥ**  
**Συστήματα Μικροϋπολογιστών**

**1<sup>η</sup> Άσκηση**  
**Ακ. έτος 2011-2012**

Γρηγόρης Λύρας    Α.Μ.: 03109687

29 Απριλίου 2012

## Άσκηση (i)

Στην άσκηση αυτή μας ζητείται να αποκωδικοποιήσουμε το πρόγραμμα που μας δίνεται από γλώσσα μηχανής σε assembly. Από τη δοσμένη ακολουθία έχουμε τον εξής κώδικα: Κυρίως κώδικας:

```
1  START:
2      MVI B,01H      ;06 00
3      LDA 2000H      ;3A 00 20
4      CPI 00H        ;FE 00
5      JZ 13          ;CA 13 08
6
7  LOA:
8      RAR            ;1F
9      JC 12          ;DA 12 08
10     INR B          ;04
11     JNZ 0A         ;C2 0A 08
12  L12:
13     MOV A,B        ;78
14  L13:
15     CMA            ;2F
16     STA 3000H      ;32 00 30
17     RST 1          ;CF
18     ;JMP START
19     END
```

Το πρόγραμμα αυτό παίρνει είσοδο από τα dip switches (2000H) και βρίσκει τη θέση του πρώτου αναμένου bit από τα δεξιά. Τη θέση αυτή την εμφανίζει στα leds (3000H) στο δυαδικό σύστημα.

Για να έχουμε συνεχόμενη λειτουργία αρκεί το RST 1 στο τέλος του προγράμματος να γίνει JMP START.

## Άσκηση (ii)

Εδώ μας ζητείται να υποποιήσουμε ένα πρόγραμμα ανάβει κυκλικά με χρονική καθυστέρηση 0.5 sec. Αν ο δεξιότερος διακόπτης στα dip switches είναι ON τα leds ανάβουν από τα δεξιά προς τα αριστερά και αντίστροφα.

Κυρίως κώδικας:

```
1  START:
2      MVI A,FEH      ;1111 1110
3      LXI B,01F4H    ;500ms
4  LOOP_L:
5      STA 3000H      ;LED on
6      CALL DELB
7      MOV C,A        ;Save status
8      RLC
9      JNC CHECK      ;if at left check DIPs
10     JMP LOOP_L      ;else, loop
11  CHECK:
12     LDA 2000H
13     RRC             ;If LSB == 0
14     JNC START       ;Cycle
15     MOV A,C         ;else move right
16     RRC
17  LOOP_R:
18     STA 3000H      ;store to led
```

```

19      CALL DELB    ;wait a bit
20      RRC
21      JNC RESTART  ;If at rightmost LED, go left
22      JMP LOOP_R   ;Else, repeat
23 RESTART:
24      MVI A,FDH    ;1111 1101
25      JMP LOOP_L   ;Just so 1st LED is only 0.5s
26 END

```

### Άσκηση (iii)

Το πρόγραμμα που υλοποιούμε μετατρέπει τον αριθμό που διαβάζει από τα dip switches στη δεκαδική του μορφή και τον εμφανίζει στα leds, με τις δεκάδες να απεικονίζονται στα 4 MSB και τις μονάδες στα 4 LSB. Για να έχουμε συνεχή λειτουργία χρησιμοποιούμε το JMP START. Αν ο αριθμός της εισόδου είναι τριψήφιος ανάβουν όλα τα leds.

Κυρίως κώδικας:

```

1  START:
2      LDA 2000H      ;Read
3      MVI B,00H
4
5  HUN:
6      SUI 64H ;while number >= 0
7      JNC HUN ;Subtract 100 until
8      ADI 64H ;Add 100
9
10 DEC:
11     INR B          ;Count tenths
12     SUI 0AH
13     JNC DEC
14     ADI 0AH
15     DCR B
16
17     MOV C,A ;Keep units on C
18     MOV A,B ;Get tens
19     RLC          ;4 shifts to
20     RLC
21     RLC
22     ADD C
23
24     CMA
25     STA 3000H      ;Show LEDs
26     JMP START
27 END

```

### Άσκηση (iv)

Σε αυτή την άσκηση καλούμαστε να υλοποιήσουμε το κύκλωμα που μας δίνεται. Η είσοδος δίνεται από τα dip switches, και η έξοδος εμφανίζεται στα leds. Το πρόγραμμα έχει συνεχή λειτουργία.

Κυρίως κώδικας:

```

1  START:
2      LDA 2000H      ;eisodos apo dip switches
3      MOV B,A        ;B=A
4      MVI D,00H      ;D=0
5      RRC            ;right shift
6      ANA B          ;A and B

```

```

7      ANI 41H      ;kratame mono ta sto 1o kai to 7o
8      MOV D,A      ;D= 0 Y1 0 0 0 0 0 C0
9      RRC
10     RRC
11     RRC          ;A= 0 0 C0 0 Y1 0 0 0
12     ORA D
13     MOV D,A      ;D= 0 Y1 0 C0 Y1 0 0 C0
14     MOV A,B      ;epanafora A
15     RRC
16     XRA B        ;A xor B
17     ANI 10H      ;mono to 5o
18     RRC          ;right shift (4o lsb)
19     MOV C,A      ;apo8hkeuoume to Y0
20     ORA D        ;A = 0 Y1 0 C0 C3 0 0 C0
21     MOV D,A      ;
22     MOV A,C      ;A = 0 0 0 0 Y0 0 0 0
23     RRC          ;shift right 3o b
24     ORA D        ;A = 0 Y1 0 C0 C3 C2 0 C0
25     MOV D,A      ;
26     MOV A,B
27     RRC
28     ORA B        ;A or B gia to C1=A1 or B1
29     ANI 04H      ;mono to 3o
30     RRC          ;right shift 2o b
31     ORA D        ; A = 0 Y1 0 C0 C3 C2 C1 C0
32     ANI 0FH      ; A = 0 0 0 0 C3 C2 C1 C0
33     CMA          ;antistrofh gia leds
34     STA 3000H
35     JMP START
36     END

```

## Άσκηση (v)

Τέλος έχουμε μια τεχνοοικονομική μελέτη με θέμα την κατασκευή μια φορητή ηλεκτρονική συσκευή με χρήση 3 τεχνολογιών.

Τεχνολογίες:

- Διακριτά στοιχεία και ολοκληρωμένα κυκλώματα σε πλακέτα

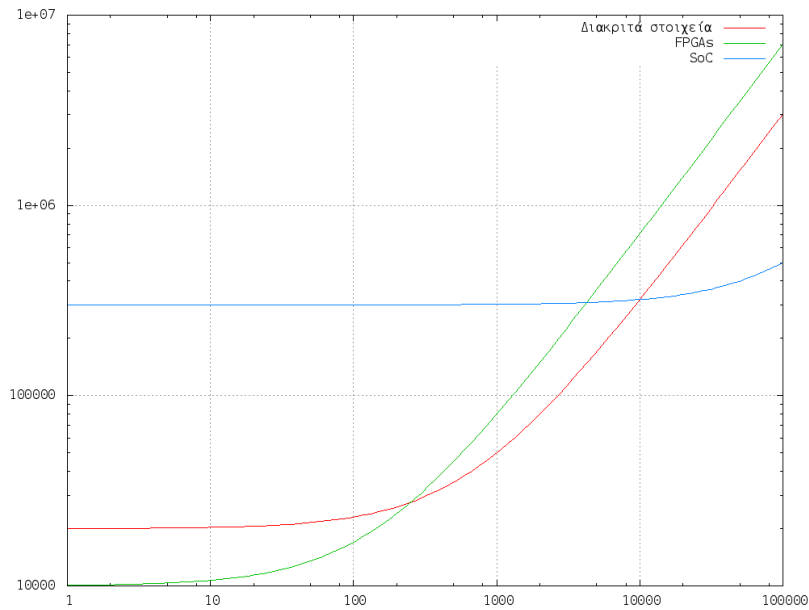
$$Cost = 20000 + (15 + 15) \times x = 20000 + 30 \times x \quad (1)$$

- FPGAs και λίγα περιφεριακά σε μικρότερη πλακέτα

$$Cost = 10000 + (60 + 10) \times x = 10000 + 70 \times x \quad (2)$$

- Σχεδίαση SoC για πολύ μικρή πλακέτα

$$Cost = 300000 + (1 + 1) \times x = 300000 + 2 \times x \quad (3)$$



Σχήμα 1: Καμπύλες ανά τεμάχιο (λογαριθμική κλίμακα)

Για να είναι μη συμφέρουσα η επιλογή των FPGA θα πρέπει η τιμή των ολοκληρωμένων για τα FPGA να ανέρχεται στα 20€.

Επαναλαμβάνοντας τους υπολογισμούς συνυπολογίζοντας στο κόστος και άλλους συντελεστές για κάθε τεχνολογία (π.χ. κόστος μπαταρίας, αυτονομία, κατανάλωση, μέγεθος) έχουμε τα ακόλουθα. Τεχνολογίες:

- Διακριτά στοιχεία και ολοκληρωμένα κυκλώματα σε πλακέτα

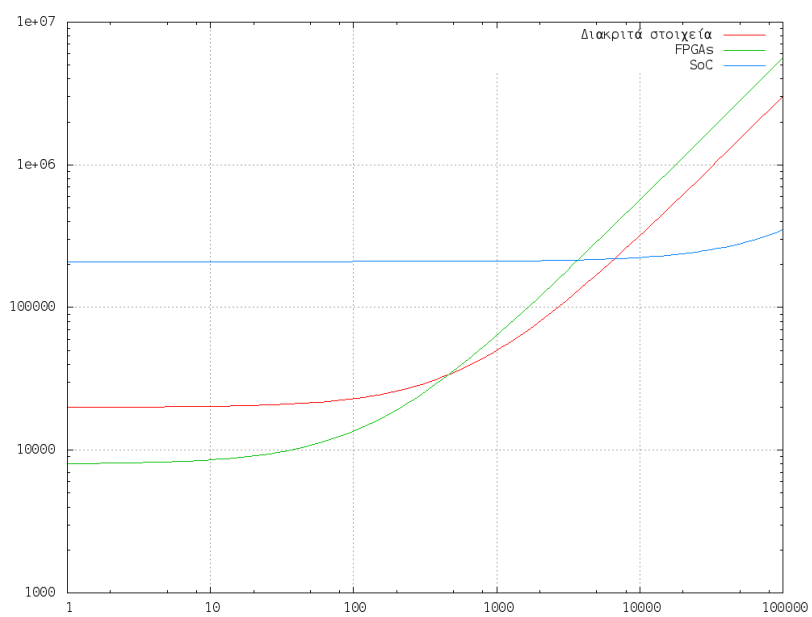
$$Cost = 20000 + (15 + 15) \times x = 20000 + 30 \times x \quad (4)$$

- FPGAs και λίγα περιφεριακά σε μικρότερη πλακέτα

$$Cost = 0.8 \times (10000 + (60 + 10) \times x) = \times(10000 + 70 \times x) \quad (5)$$

- Σχεδίαση SoC για πολύ μικρή πλακέτα

$$Cost = 0.7 \times (300000 + (1 + 1) \times x) = 0.7 \times (300000 + 2 \times x) \quad (6)$$



Σχήμα 2: Καμπύλες ανά τεμάχιο (λογαριθμική κλίμακα)