

# Συστήματα Μικροϋπολογιστών

## 1η Σειρά ασκήσεων

Παναγιώτης Μπέλσης  
Ανδρόνικος Γιαννουσιάδης

AM: 03120874  
AM: 03120241

### Άσκηση 1

Αριστερά δίνεται το πρόγραμμα σε assembly.  
Δεξιά δίνεται το πρόγραμμα με συμβολικές διευθύνσεις.

```
        MVI C,08H
        LDA 2000H
LABEL2:
        RAL
        JC LABEL1
        DCR C
        JNZ LABEL2
LABEL1:
        MOV A,C
        CMA
        STA 3000H
        RST 1
END
```

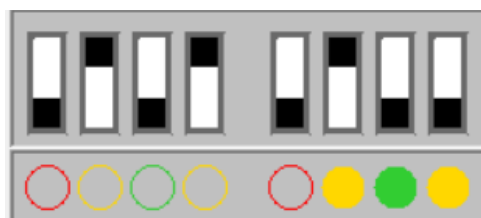
0800	0E	MVI C,08H
0801	08	
0802	3A	LDA 2000H
0803	00	
0804	20	
LABEL2:		
0805	17	RAL
0806	DA	JC LABEL1
0807	0D	
0808	08	
0809	0D	DCR C
080A	C2	JNZ LABEL2
080B	05	
080C	08	
LABEL1:		
080D	79	MOV A,C
080E	2F	CMA
080F	32	STA 3000H
0810	00	
0811	30	
0812	CF	RST 1

Η λειτουργία του προγράμματος είναι η εξής:

Ανάλογα με τη θέση του πιο αριστερού dip switch που είναι ON, εμφανίζεται στα φωτάκια LED η δυαδική αναπαράσταση του αριθμού της θέσης. Θεωρούμε πως τα φωτάκια είναι αριθμημένα από δεξιά προς τα αριστερά (1 έως 8). Τα φωτάκια είναι αντίστροφης λογικής, δηλαδή όταν είναι 0 είναι ON, ενώ όταν είναι 1 είναι OFF.

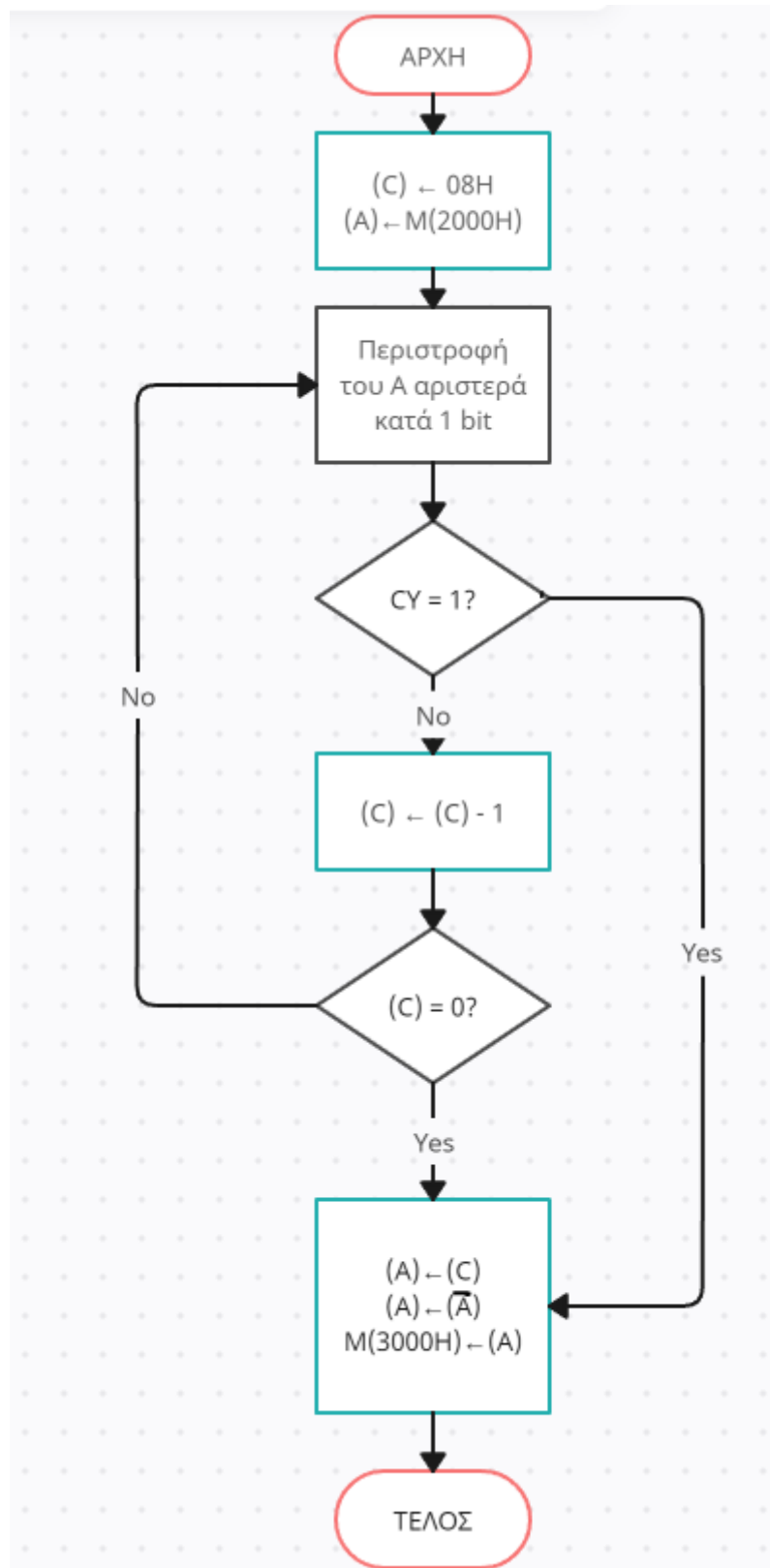
**Παράδειγμα εκτέλεσης:**

8   7   6   5   4   3   2   1



(το αριστερότερο ανοιχτό dip switch είναι το 7ο. Άρα στα φωτάκια ανάβει ο αριθμός 7 σε δυαδική αναπαράσταση.)

**Διάγραμμα Ροής :**



Για να έχουμε συνεχή λειτουργία του προγράμματος αντικαθιστούμε στον κώδικα την εντολή RST 1, με μια εντολή JMP που μας πηγαίνει ξανά στην αρχή του προγράμματος. Με αυτό τον τρόπο το πρόγραμμα θα επαναλαμβάνεται χωρίς τέλος.

Έπειτα από αυτή την αλλαγή το πρόγραμμα έγινε ως εξής :

```
Start:
    MVI C,08H
    LDA 2000H
LABEL2:
    RAL
    JC LABEL1
    DCR C
    JNZ LABEL2
LABEL1:
    MOV A,C
    CMA
    STA 3000H
    JMP Start
END
```

## Άσκηση 2

```
IN 10H
LXI B,01F4H ; delay=500ms
MVI E,FEH ; E=11111110

CHECK:
LDA 2000H ; load dip switches to A
CALL DELB ; using delay of 500ms
RRC ; rotate right to check CY
JNC CHECK ; if LSB dip switch is OFF

RLC ; gurnaw se arxikh katastash
RLC ; check MSB (fora kinhshs)
JC RIGHT ; if CY=1 -> deksioastrofh kinhsh LED
; alliws -> aristerostrofh kinhsh LED

LEFT:
MOV A,E ; prohgoumeno state tw n LED
STA 3000H
RLC ; kinhsh mia thesi aristera
MOV E,A ; krataw kainouria katastasi sto E
JMP CHECK

RIGHT:
MOV A,E ; prohgoumeno state tw n LED
STA 3000H
RRC ; kinhsh mia thesi deksia
MOV E,A ; krataw kainouria katastasi sto E
JMP CHECK

END
```

### Άσκηση 3

```
LXI B,01F4H
CHECK:
    LDA 2000H
    CPI C8H      ;elegxos gia >=200
    JNC GTR200
    CPI 64H      ;elegxos gia >=100
    JNC GTR100
    MVI E,FFH    ;E=-1

DECA:
    INR E        ;E=0
    SUI 0AH      ;kane A-10 mexri A<0
    JNC DECA

    ADI 0AH      ;diorthosi tou A gia na vroume tis monades
    MOV D,A      ;D = MONADES

    MOV A,E      ;A = DEKADES
    RLC          ;metakinoume tis dekaDES sto MSB
    RLC
    RLC
    RLC
    ADD D        ;pleon A = Dekades Monades
    CMA         ;ta LED einai antistrofhs logikhs
    STA 3000H    ;fortwnoume thn timh sta LED
    JMP CHECK

GTR200:
    MVI A,0FH    ;A = 0000 1111
    STA 3000H    ;4 MSB LED = ON
    CALL DELB
    MVI A,FFH    ;A = 0000 0000
    STA 3000H    ;OFF
    CALL DELB
    JMP CHECK    ; check an allakse kapoios diakoptis

GTR100:
    MVI A,F0H    ;A = 1111 0000
    STA 3000H    ;4 LSB LED = ON
    CALL DELB
    MVI A,FFH    ;A = 0000 0000
    STA 3000H    ;OFF
    CALL DELB
    JMP CHECK    ; check an allakse kapoios diakoptis

END
```

## Άσκηση 4

Υπόδειξη : Κόστος= Αρχικό + (Κόστος-ICs + Κόστος-κατασκευής)\*Πλήθος τεμαχίων

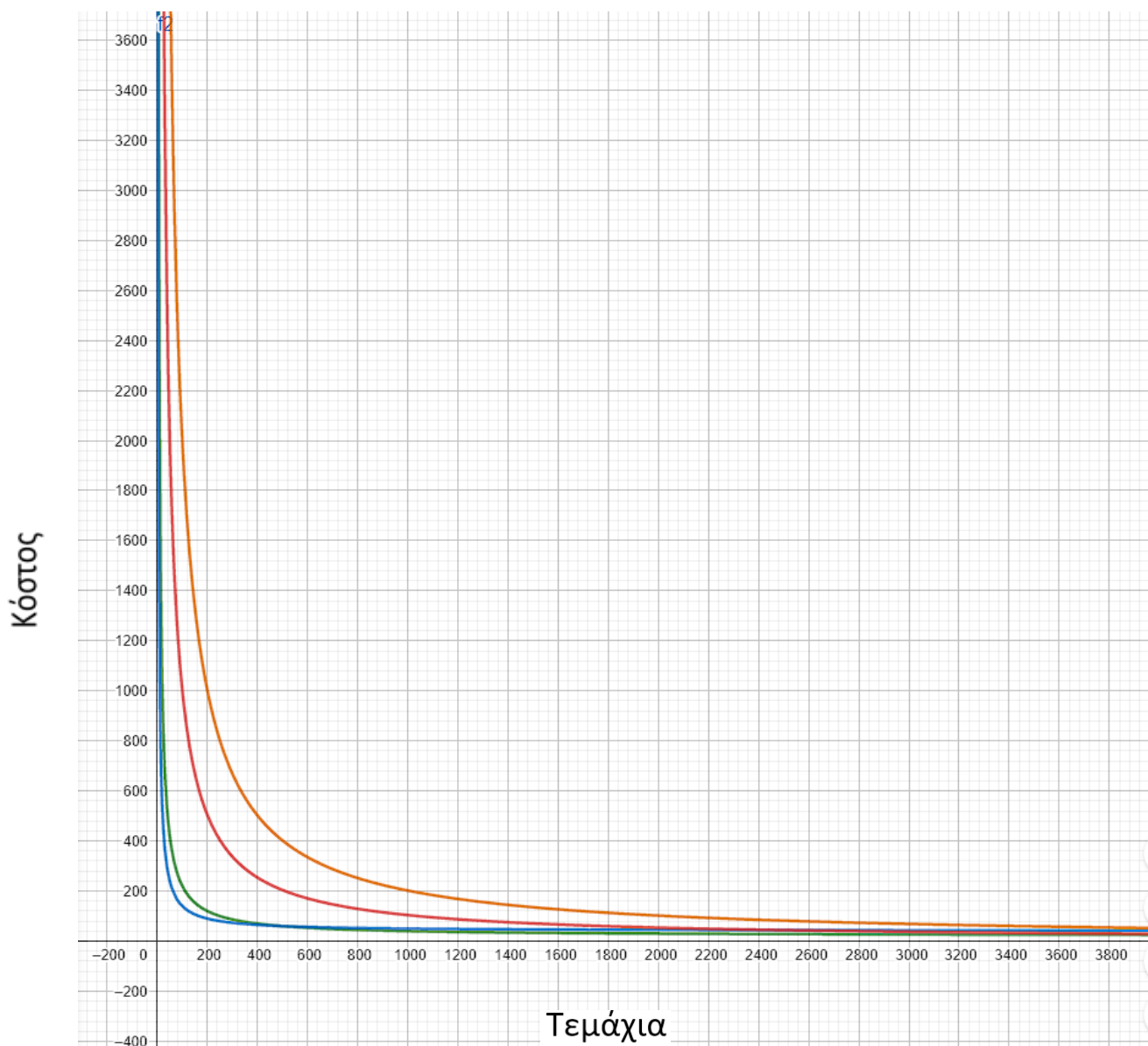
Με χρήση της παραπάνω σχέσης υπολογίσαμε :

$$1) f_1(x) = \frac{20.000}{x} + 20$$

$$2) f_2(x) = \frac{10.000}{x} + 40$$

$$3) f_3(x) = \frac{100.000}{x} + 4$$

$$4) f_4(x) = \frac{200.000}{x} + 2$$



Παίρνοντας τις παραπάνω συναρτήσεις σε ζεύγη στοχεύουμε στην εύρεση των σημείων τομής των καμπυλών του παραπάνω διαγράμματος.

$$\mathbf{1-2:} f_1(x) = f_2(x) \Rightarrow x = 500$$

$$\mathbf{1-3:} f_1(x) = f_3(x) \Rightarrow x = 5000$$

$$\mathbf{1-4:} f_1(x) = f_4(x) \Rightarrow x = 10000$$

$$\mathbf{2-3:} f_2(x) = f_3(x) \Rightarrow x = 2500$$

$$\mathbf{2-4:} f_2(x) = f_4(x) \Rightarrow x = 5000$$

$$\mathbf{3-4:} f_3(x) = f_4(x) \Rightarrow x = 50000$$

Τελικά, οι περιοχές τεμαχίων που συμφέρουν πιο πολύ για κάθε τεχνολογία είναι:

$$\mathbf{(1) I.C.:} 500 < x < 5000$$

$$\mathbf{(2) FPGA:} 0 < x < 500$$

$$\mathbf{(3) SoC-1:} 5000 < x < 50000$$

$$\mathbf{(4) SoC-2:} 50000 < x$$

Οι τεχνολογίες με το μεγαλύτερο κόστος σχεδίασης (SoC-1, SoC-2) είναι πιο κερδοφόρες για υψηλότερους αριθμούς τεμαχίων.

Προκειμένου να εξαφανιστεί η επιλογή της πρώτης τεχνολογίας θα πρέπει, για το νέο κόστος ( $K_{new}$ ) ανά I.C. της τεχνολογίας των FPGAs, να ισχύει:

$$f_2'(x) = \frac{10000 + (10 + K_{new})x}{x} < \frac{20000 + 20x}{x} = f_1(x) \Rightarrow$$

$$K_{new} < \frac{10000 + 10x}{x}$$

Για  $x \rightarrow \infty$  θα πρέπει να ισχύει  $K_{new} < 10$ . Άρα, αρκεί  $K_{new} \leq 10$ . Άρα, οποιαδήποτε τιμή από 0 έως 10€ για το νέο κόστος  $\kappa$  αρκεί για να αποκλειστεί η πρώτη τεχνολογία.