ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Συστήματα Μικροϋπολογιστών

1η ομάδα ασκήσεων

Διδάσκων:

Δ. Σούντρης

Ομάδα:

Ειρήνη Δόντη
ΑΜ 03119839

6ο εξάμηνο

Αθήνα 2022

Περιεχόμενα:

1η ΑΣΚΗΣΗ:

Γράφουμε το πρόγραμμα σε assembly, βάσει του δοσμένου προγράμματος σε γλώσσα μηχανής:

```
ΆΣΚΗΣΗ 1η:

0800 06 MVI B,01H
0801 01
0802 3A LDA 2000H
0803 00
0804 20
0805 FE CPI 00H
0806 00
0807 CA JZ 0813H
0808 13
0809 08
080A 1F RAR
080B DA JC 0812H
080C 12
080D 08
080E 04 INR B
080F C2 JNZ 080AH
0810 0A
0811 08
0812 78 MOV A,B
0813 2F CMA
0814 32 STA 3000H
0815 00
0816 30
0817 CF RST 1
```

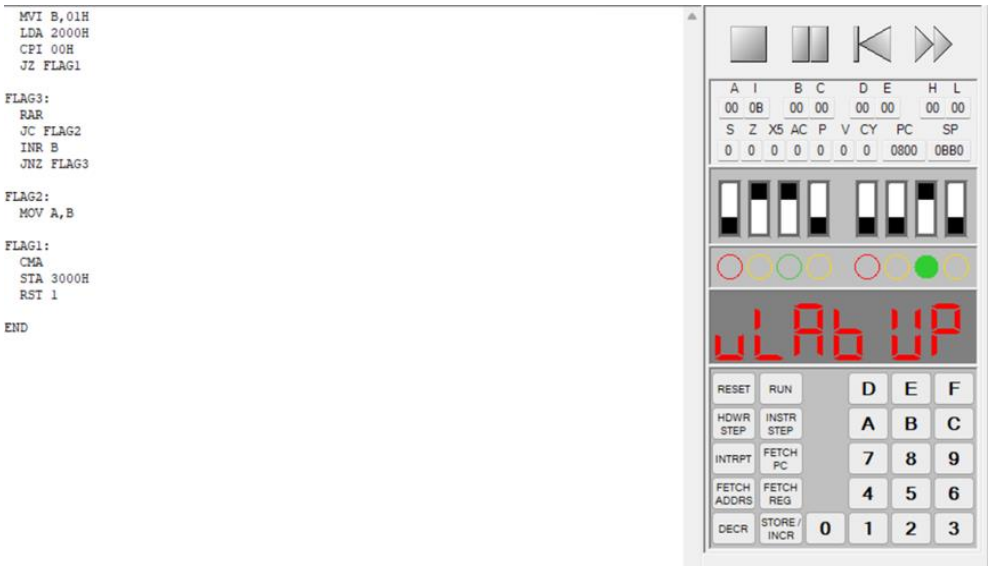
Θεωρούμε ότι το πρόγραμμα είναι φορτωμένο στη μνήμη με αρχή τη διεύθυνση 0800 και ότι οι bold κωδικοί είναι εντολές.

Εκτελούμε την προσομοίωση του προγράμματος, όπως φαίνεται παρακάτω:

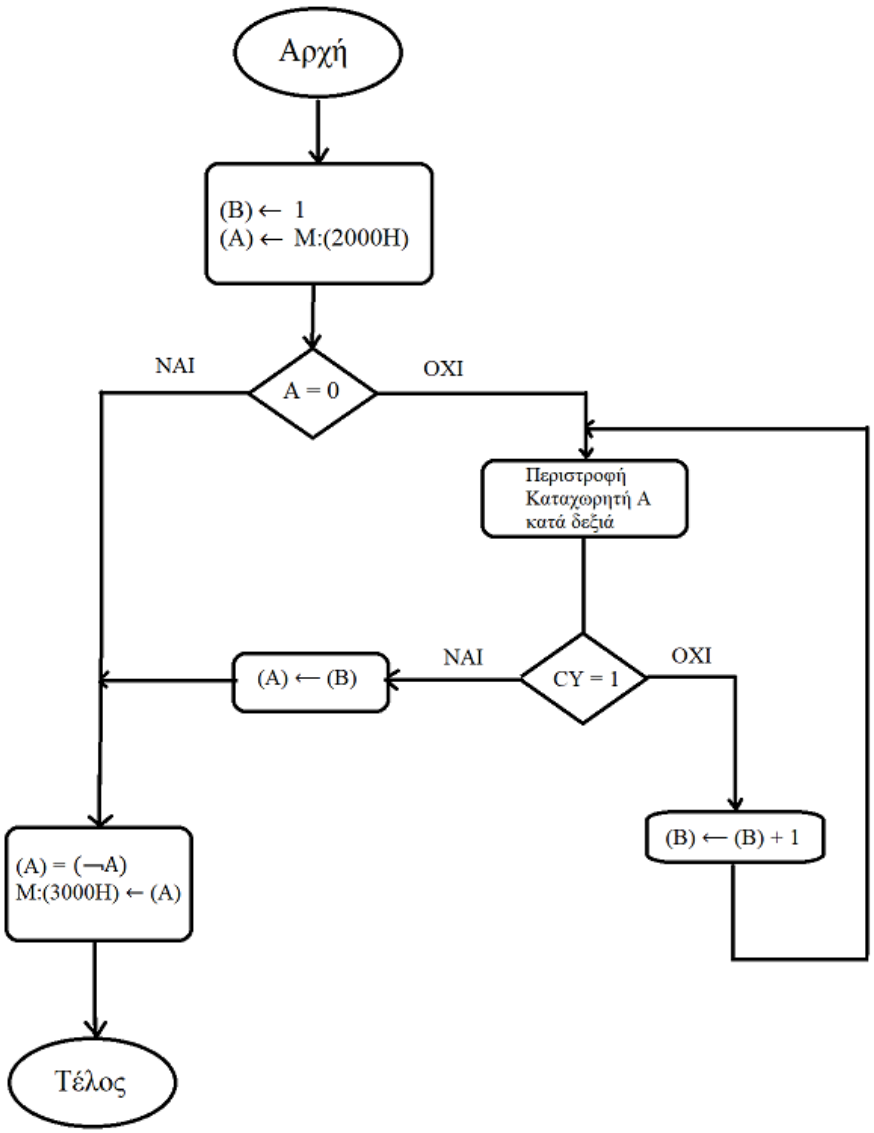
```
  MVI B,01H
  LDA 2000H
  CPI 00H
  JZ FLAG1

FLAG3:
  RAR
  JC FLAG2
  INR B
  JNZ FLAG3

FLAG2:
  MOV A,B

FLAG1:
  CMA
  STA 3000H
  RST 1

END
```

Παρατηρούμε ότι το πρόγραμμα ανιχνεύει και τυπώνει τη θέση του δεξιότερου 1 σε έναν αριθμό 8-bit που δίνεται ως είσοδος. Στην περίπτωση που δεν υπάρχουν λογικά 1 στον δοσμένο αριθμό, το πρόγραμμα τυπώνει 0.

Για το παρακάτω παράδειγμα προσομοίωσης, αν δώσουμε ως είσοδο τον δυαδικό αριθμό 01100010, τότε το πρόγραμμα θα επιστρέψει τον δυαδικό αριθμό 00000010, δηλαδή τον δεκαδικό αριθμό 2.



Σχεδιάζουμε το διάγραμμα ροής του προγράμματος:



Για να επαναλαμβάνεται χωρίς τέλος η λειτουργία του προγράμματος, θα πρέπει να προσθέσουμε, στην αρχή του προγράμματος, μία ετικέτα BEGIN και μία εντολή JMP BEGIN πριν την εντολή END. Δεν αρκεί μόνο η εντολή RST 1, η οποία κάνει μία προσωρινή παύση στο πρόγραμμα μετά από κάθε τύπωμα μίας εξόδου.

2η ΑΣΚΗΣΗ:

Αρχικά, προσθέτουμε την εντολή IN 10H, η οποία αίρει την προστασία μνήμης επιτρέποντας την πρόσβαση για αποθήκευση μεταβλητών και δεδομένων στη διαθέσιμη μνήμη RAM. Στη συνέχεια, φορτώνουμε στο ζεύγος καταχωρητών τον αριθμό 500, ώστε η DELB να προκαλεί καθυστέρηση 500 ms. Επίσης, στον καταχωρητή A, φορτώνεται η αρχική θέση του αναμμένου LED (FE στο δεκαεξαδικό) και στη συνέχεια εκτελείται μία συνεχώς επαναλαμβανόμενη διαδικασία η οποία προβλέπει τις εξής συνθήκες:

Αν τα δύο *LSB* της εισόδου είναι *01*, το αναμμένο LED μετακινείται κατά μία θέση αριστερά. Στην περίπτωση που βρίσκεται ήδη στην αριστερότερη θέση, το αναμμένο LED μετακινείται κατά μία θέση δεξιά και συνεχίζεται αυτή η διαδικασία, έως ότου το αναμμένο LED να φτάσει στη δεξιότερη θέση ή να αλλάξουν τα δύο τελευταία LSB της εισόδου.

Αν τα δύο *LSB* της εισόδου είναι *00*, τότε το αναμμένο LED μετακινείται κατά μία θέση αριστερά μέχρι να αλλάξουν τα δύο LSB της εισόδου.

Αν τα δύο *LSB* της εισόδου είναι *10 ή 11*, τότε το δεξιότερο LED παραμένει αναμμένο έως ότου αλλάξουν τα δύο τελευταία LSB της εισόδου, αποθηκεύοντας παράλληλα την προηγούμενη θέση του αναμμένου LED στην περίπτωση που αλλάξουν τα δύο τελευταία LSB της εισόδου και χρειαστεί το πρόγραμμα να ξεκινήσει από το σημείο που διακόπηκε.

Ο κώδικας του ερωτήματος που προσομοιώνεται, παρουσιάζεται παρακάτω:

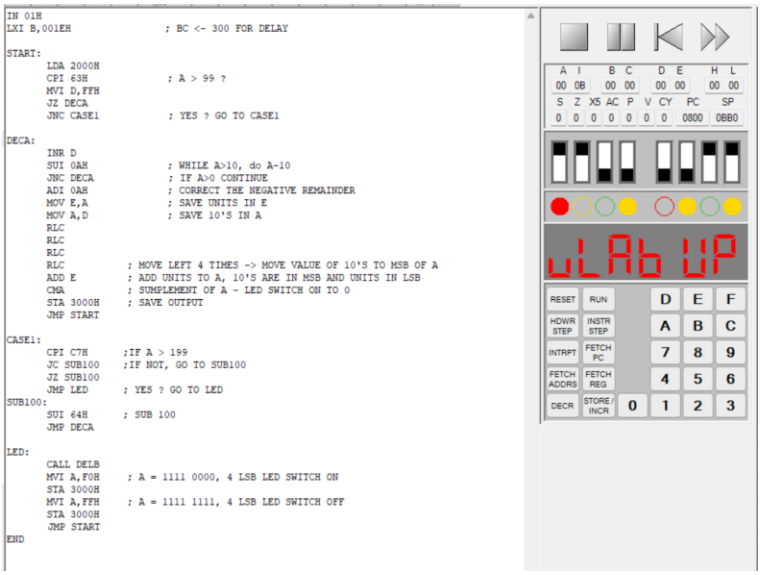
```
  IN 10H        ; NO MEMORY SAFE
  LXI B,01F4H   ; BC <- 500 FOR DELB DELAY
  MVI A,FEH     ; LED STARTING THESIS

EXIT:
  STA 3000H     ; STORE IN EXIT PORT
  MOV D,A       ; D IN PREVIOUS LED POSITION

START:
  CALL DELB     ; 0.5ms DELAY
  LDA 2000H     ; SAVE A
  ANI 03H       ; KEEP VALUES OF 2 LSBs
  CPI 01H       ; {X = (A) - 01} IF X = 0 => Z = 1
  JZ  LEFT      ; MOVE LEFT

RETURN:
  CPI 00H       ; IF {A = 00} => Z=1
  JZ LEFT_ROTATION
  MVI A,FEH     ; LED TO INITIAL STATE
  STA 3000H     ; STORE IN EXIT PORT
  JMP START     ; BEGIN

LEFT:
  MOV A,D       ; A IN PREVIOUS LED POSITION
  CPI 7FH       ; IF LED IS TO MSB
  JZ RIGHT      ; MOVE RIGHT
  RLC           ; A INDEX MOVE 1 THESIS LEFT
  JMP EXIT

RIGHT:
  MOV A,D       ; A IN PREVIOUS LED POSITION
  CPI FEH       ; IF LED IS IN LSB
  JZ LEFT       ; MOVE LEFT
  RRC           ; A INDEX MOVE 1 THESIS RIGHT
  STA 3000H     ; STORE IN EXIT PORT
  MOV D,A       ; D IN PREVIOUS LED POSITION
  CALL DELB     ; 0.5ms DELAY
  LDA 2000H     ; SAVE A
  ANI 03H       ; KEEP VALUES OF 2 LSBs
  CPI 01H       ; {X = (A) - 01} IF X = 0 => Z = 1
  JZ  RIGHT     ; MOVE RIGHT
  JMP RETURN    ; RETURN

LEFT_ROTATION:
  MOV A,D       ; A IN PREVIOUS LED POSITION
  RLC           ; A INDEX MOVE 1 THESIS LEFT
  JMP EXIT

END
```

3η ΑΣΚΗΣΗ:

Επεκτείνουμε το 4ο πρόγραμμα που αφορά τη μετατροπή δυαδικού αριθμού των 8 bits σε δυαδική μορφή 2 ψηφίων. Εκτελούμε, όπως φαίνεται παρακάτω, τη ζητούμενη προσομοίωση με καθυστέρηση εναλλαγής LED 300 ms:

```
IN 01H
LXI B,001EH               ; BC <- 300 FOR DELAY

START:
       LDA 2000H
       CPI 63H             ; A > 99 ?
       MVI D,FFH
       JZ DECA
       JNC CASE1           ; YES ? GO TO CASE1

DECA:
       INR D
       SUI 0AH             ; WHILE A>10, do A-10
       JNC DECA            ; IF A>0 CONTINUE
       ADI 0AH             ; CORRECT THE NEGATIVE REMAINDER
       MOV E,A             ; SAVE UNITS IN E
       MOV A,D             ; SAVE 10'S IN A
       RLC
       RLC
       RLC
       RLC          ; MOVE LEFT 4 TIMES -> MOVE VALUE OF 10'S TO MSB OF A
       ADD E        ; ADD UNITS TO A, 10'S ARE IN MSB AND UNITS IN LSB
       CMA          ; SUMPLEMENT OF A - LED SWITCH ON TO 0
       STA 3000H    ; SAVE OUTPUT
       JMP START

CASE1:
       CPI C7H      ;IF A > 199
       JC SUB100    ;IF NOT, GO TO SUB100
       JZ SUB100
       JMP LED      ; YES ? GO TO LED
SUB100:
       SUI 64H      ; SUB 100
       JMP DECA

LED:
       CALL DELB
       MVI A,F0H    ; A = 1111 0000, 4 LSB LED SWITCH ON
       STA 3000H
       MVI A,FFH    ; A = 1111 1111, 4 LSB LED SWITCH OFF
       STA 3000H
       JMP START
END
```

Εκτελούμε την προσομοίωση με αριθμό εισόδου την τιμή 195 (11000011 στο δυαδικό), οπότε λαμβάνουμε την παρακάτω έξοδο:



```
IN 01H
LXI B,001EH              ; BC <- 300 FOR DELAY

STARI:
       LDA 2000H
       CPI 63H           ; A > 99 ?
       MVI D,FFH
       JZ DECA
       JNC CASE1         ; YES ? GO TO CASE1

DECA:
       INR D
       SUI 0AH           ; WHILE A>10, do A-10
       JNC DECA          ; IF A>0 CONTINUE
       ADI 0AH           ; CORRECT THE NEGATIVE REMAINDER
       MOV E,A           ; SAVE UNITS IN E
       MOV A,D           ; SAVE 10'S IN A
       RLC
       RLC
       RLC
       RLC          ; MOVE LEFT 4 TIMES -> MOVE VALUE OF 10'S TO MSB OF A
       ADD E        ; ADD UNITS TO A, 10'S ARE IN MSB AND UNITS IN LSB
       CMA          ; SUMPLEMENT OF A - LED SWITCH ON TO 0
       STA 3000H    ; SAVE OUTPUT
       JMP START

CASE1:
       CPI C7H      ;IF A > 199
       JC SUB100    ;IF NOT, GO TO SUB100
       JZ SUB100
       JMP LED      ; YES ? GO TO LED
SUB100:
       SUI 64H      ; SUB 100
       JMP DECA

LED:
       CALL DELB
       MVI A,F0H    ; A = 1111 0000, 4 LSB LED SWITCH ON
       STA 3000H
       MVI A,FFH    ; A = 1111 1111, 4 LSB LED SWITCH OFF
       STA 3000H
       JMP START
END
```

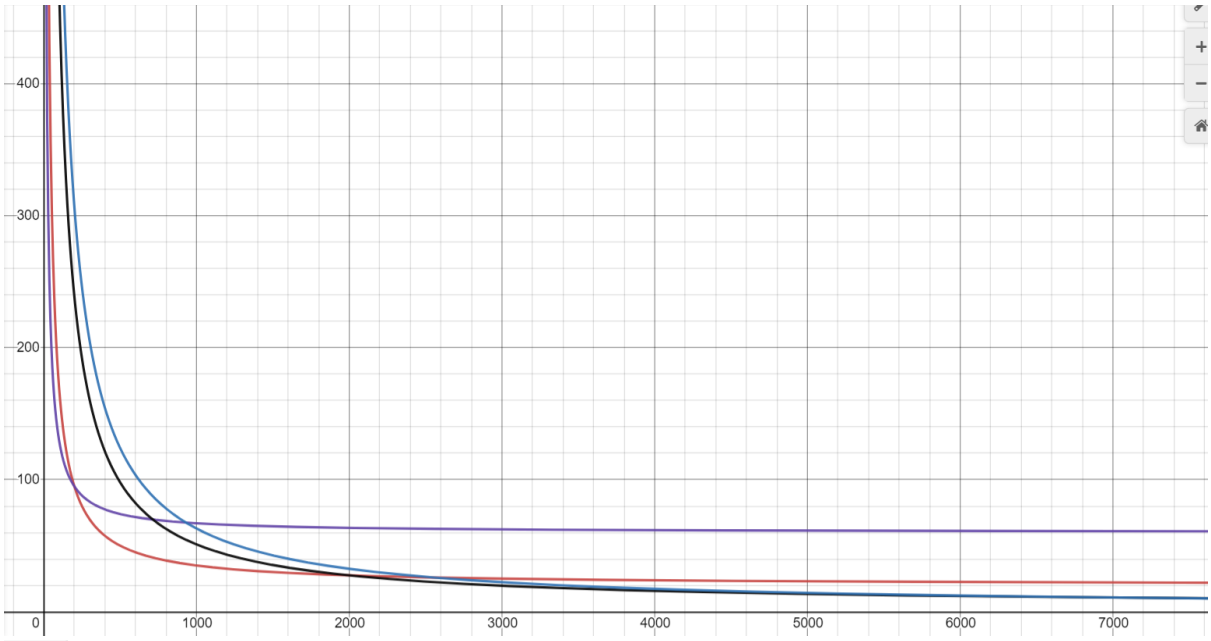

Παρατηρούμε ότι ο αριθμός που εμφανίστηκε στην έξοδο, είναι ο αριθμός 95, καθώς δημιουργούνται στην έξοδο 2^0 + 2^2 = 5 μονάδες και 2^0 + 2^3 = 9 δεκάδες.

Εκτελούμε την προσομοίωση με αριθμό εισόδου την τιμή 250 (11111010 στο δυαδικό) και παρατηρούμε ότι τα 4 LSB αναβοσβήνουν συνεχώς. Οπότε, η προσομοίωσή μας παράγει τα σωστά αποτελέσματα.

4η ΑΣΚΗΣΗ:

Οι σχέσεις και οι αντίστοιχες καμπύλες (σχεδιασμένες σε octave) κόστους ανά τεμάχιο για τις 4 τεχνολογίες είναι οι παρακάτω.

$$x_1 = \frac{15000 + (10 + 10)x}{x}$$
$$x_2 = \frac{7000 + (50 + 10)x}{x}$$
$$x_3 = \frac{47000 + (2 + 2)x}{x}$$
$$x_4 = \frac{61000 + (1 + 1)x}{x}$$



1 $\frac{(15000 + 20x)}{x} \{x \geq 0\}$

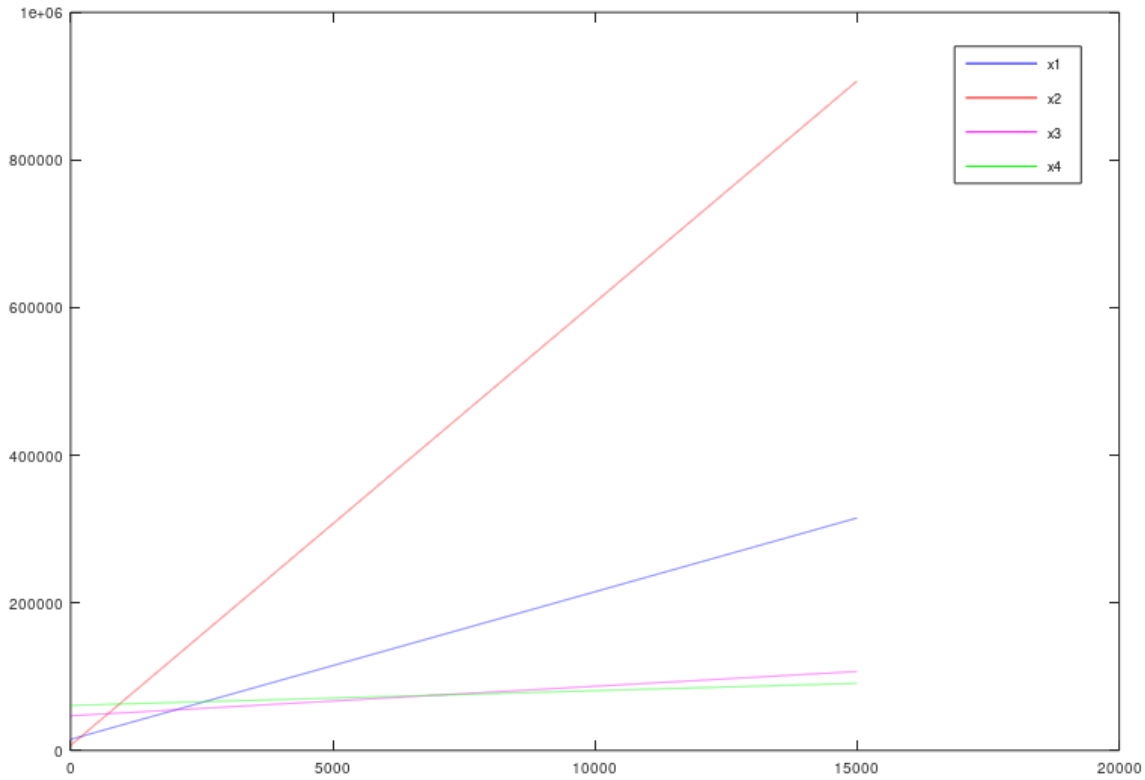
2 $\frac{(7000 + 60x)}{x} \{x \geq 0\}$

3 $\frac{(47000 + 4x)}{x} \{x \geq 0\}$

4 $\frac{(61000 + 2x)}{x} \{x \geq 0\}$

Όπου x είναι ο συνολικός αριθμός των τεμαχίων.

Για ευκολία, παρακάτω απεικονίζουμε και το συνολικό κόστος της κάθε μιας τεχνολογίας.



Βρίσκουμε κάποια από τα σημεία τομής των 4 καμπυλών:

- Καμπύλες x1, x2 σημείο τομής x = 200
- Καμπύλες x1, x3 σημείο τομής x = 2000
- Καμπύλες x3, x4 σημείο τομής x = 7000

Διακρίνουμε τις 4 περιοχές που είναι πιο συμφέρουσες για την κάθε μια τεχνολογία:

- 0 - 200 τεμάχια: **2η** τεχνολογία
- 200 - 2000 τεμάχια: **1η** τεχνολογία
- 2000 - 7000 τεμάχια: **3η** τεχνολογία
- 7000 και περισσότερα τεμάχια : **4η** τεχνολογία

Για να εξαφανιστεί η επιλογή της 1ης τεχνολογίας θέλουμε στο σημείο τομής της 1ης και της 3ης καμπύλης, δηλαδή στα 2000 τεμάχια, η 2η τεχνολογία να είναι πιο φθηνή από την 1η. Συνεπώς, πρέπει η τιμή των I.C στην τεχνολογία των FPGAs να είναι το πολύ 14€. Η τιμή αυτή προκύπτει από την σχέση x2 < x1, για γνωστό αριθμό τεμαχίων 2000 και άγνωστο το κόστος των I.C ανά τεμάχιο στην 2η τεχνολογία.
Μπορούμε να παρατηρήσουμε ότι για 2000 τεμάχια και κόστος I.C 14€ ανά τεμάχιο, το συνολικό κόστος της 2ης τεχνολογίας είναι 55000€, δηλαδή ισούται με το κόστος της 1ης και της 3ης τεχνολογίας για τα 2000 τεμάχια (σημείο τομής).