

Άσκηση 5

Σύνθετοι έλεγχοι – Δομές επανάληψης

A. ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ

Για το θεωρητικό μέρος, ανατρέξτε στην άσκηση 4 (εκεί έχουν παρουσιαστεί οι τεχνικές ελέγχου)

B. ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ

1. Το πρόγραμμα που ακολουθεί, ελέγχει αν ο αριθμός που είναι αποθηκευμένος στη θέση [0002] ανήκει στο διάστημα [1Α, 40]. Αν ανήκει, τότε θα αποθηκεύεται ο αριθμός 5 στη θέση μνήμης [0005].

<u>Πρόγραμμα</u>	<u>Πρόγραμμα στον debugger</u>	<u>Ερωτήσεις</u>
mov AL,[0001]	0100 mov AL,[0002]	α) Ποιος είναι ακριβώς ο έλεγχος που υλοποιείται ;
CMP AL,1a	0103 CMP AL,1a	β) Σχεδιάστε το διάγραμμα ροής
JGE syn2	0105 JGE 0109	γ) Δοκιμάστε το πρόγραμμα για περιεχόμενο θέσης μνήμης [0002] τους αριθμούς 10, 20 και 41
jmp exit	0107 jmp 0114	*το λογισμικό αντικαθιστά το JGE με το JNL (Non Less)
syn2:	0109	
CMP AL,40	CMP AL,40	
JLE syn3	010B JLE 010F	
jmp exit	010D jmp 0114	
syn3:	010F	
mov [0005],5	mov [0005],5	
exit:	0114	

2. Το πρόγραμμα που ακολουθεί, ελέγχει αν ο αριθμός που είναι αποθηκευμένος στη θέση [0002] ανήκει στο διάστημα [1Α, 40]. Αν ανήκει, τότε θα αποθηκεύεται ο αριθμός 5 στη θέση μνήμης [0005].

<u>Πρόγραμμα</u>	<u>Πρόγραμμα στον debugger</u>	<u>Ερωτήσεις</u>
mov AL,[0001]	0100 mov AL,[0002]	α) Ποιος είναι ακριβώς ο έλεγχος που υλοποιείται ;
CMP AL,1a	0103 CMP AL,1a	β) Σχεδιάστε το διάγραμμα ροής
JL exit	0105 JL 0110	γ) Δοκιμάστε το πρόγραμμα για περιεχόμενο θέσης μνήμης [0002] τους αριθμούς 10, 20 και 41
CMP AL,40	0107 CMP AL,40	δ) Να σχολιάσετε επαρκώς τις διαφορές των δύο υλοποιήσεων-προγραμμάτων (ερωτήματα 1 και 2)
JG exit	0109 JG 0110	
mov [0005],5	010B mov [0005],5	
exit:	0110	

3. Δοκιμάστε το ακόλουθο πρόγραμμα:

<u>Πρόγραμμα</u>	<u>Πρόγραμμα στον debugger</u>	<u>Ερωτήσεις</u>
mov al,1	0100 mov al,1	α) Σχεδιάστε διάγραμμα ροής
mov bl,0	0102 mov bl,0	β) Τι υπολογίζει ;
start:	0104:	γ) Ποια είναι τα βασικά συστατικά ενός βρόχου επανάληψης ; Πώς υλοποιούνται στη γλώσσα Assembly ;
add bl,al	add bl,al	
inc al	0106 inc al	
cmp al,0a	0108 cmp al,0a	
JLE start	010A JLE 0104	

4. Δοκιμάστε το ακόλουθο πρόγραμμα:

Πρόγραμμα

```
mov al,0a
mov bl,0
start:
  add bl,al
  dec al
  JNZ start
```

Πρόγραμμα στον debugger

```
0100  mov al,0a
0102  mov bl,0
0104:
      add bl,al
0106  dec al
0108  JNZ 0104
010A
```

Ερωτήσεις

- α) Σχεδιάστε διάγραμμα ροής
- β) Τι υπολογίζει ;
- γ) Γιατί δεν υπάρχει εντολή CMP, ενώ απαιτείται έλεγχος ;
- δ) Ποια προγραμματιστική δομή υλοποιείται στα ερωτήματα 3 και 4;

*το λογισμικό αντικαθιστά το JNZ με το JNE (Not Equal)

5. Δοκιμάστε το ακόλουθο πρόγραμμα:

Πρόγραμμα

```
mov al,1
mov bl,0
start:
  cmp al,0a
  JG exit
  add bl,al
  inc al
  JMP start
Exit:
```

Πρόγραμμα στον debugger

```
0100  mov al,1
0102  mov bl,0
0104
      cmp al,0a
0106  JG 010E
0108  add bl,al
010A  inc al
010C  jmp 0104
010E
```

Ερωτήσεις

- α) Σχεδιάστε διάγραμμα ροής
- β) Τι υπολογίζει ;
- γ) Ποια προγραμματιστική δομή υλοποιείται;
- δ) Να συγκρίνετε τις υλοποιήσεις των ερωτημάτων 4 και 5
- ε) Ποιες αλλαγές θα έπρεπε να γίνουν στο πρόγραμμα, αν αντικαθιστούσαμε την εντολή JG με JLE ;

6. Υλοποιήστε μια δομή επανάληψης **do-while** σε Assembly (από τα προηγούμενα ερωτήματα) και C. Να σχολιάσετε ομοιότητες και διαφορές.

7. Να υλοποιήσετε ένα πρόγραμμα για την ακόλουθη λειτουργία:

Φόρτωση του AL από τη θέση μνήμης [0005]
Αν AL<0, τότε [0002]=FF
Αν AL=0, τότε υπολογισμός του 1+2+...+10
Αν AL>0, τότε [0003]=1 και [0004]=2

Δοκιμάστε το πρόγραμμα για περιεχόμενο της θέσης [0005] τις τιμές 00, 01 και FF

Βοηθητικός πίνακας

Εντολές άλματος		
Μνημονικό	Σημασία	Συνθήκη άλματος
JE	Jump if Equal (=) <i>Αν είναι ίσο</i>	ZF=1
JZ	Jump if Zero (=) <i>Αν το αποτέλεσμα είναι μηδέν (ισότητα μέσω ελέγχου του ZF)</i>	ZF=1
JNE	Jump if Not Equal (≠) <i>Αν δεν είναι ίσο</i>	ZF=0
JNZ	Jump if Not Zero (≠) <i>Αν το αποτέλεσμα δεν είναι μηδέν (ανισότητα μέσω ελέγχου του ZF)</i>	ZF=0
JG	Jump if Greater (signed) (>) <i>Αν είναι μεγαλύτερο</i>	ZF=0 και SF=OF
JGE	Jump if Greater or Equal (signed) (>=) <i>Αν είναι μεγαλύτερο ή ίσο</i>	SF=OF
JNG	Jump if Not Greater (signed) (<=) <i>Αν δεν είναι μεγαλύτερο</i>	ZF=1 ή SF != OF
JNGE	Jump if Not Greater or Equal (signed) (<=) <i>Αν δεν είναι μεγαλύτερο ή ίσο</i>	SF != OF
JL	Jump if Less (signed) (<) <i>Αν είναι μικρότερο</i>	SF != OF
JLE	Jump if Less or Equal (signed) (<=) <i>Αν είναι μικρότερο ή ίσο</i>	ZF=1 ή SF != OF
JNL	Jump if Not Less (signed) (>=) <i>Αν δεν είναι μικρότερο</i>	SF=OF
JNLE	Jump if Not Less or Equal (signed) (>=) <i>Αν δεν είναι μικρότερο ή ίσο</i>	ZF=0 και SF=OF
JC	Jump if Carry <i>Αν έχει προκύψει κρατούμενο</i>	CF=1
JMP	Εντολή άλματος χωρίς συνθήκη <i>Χωρίς έλεγχο</i>	-