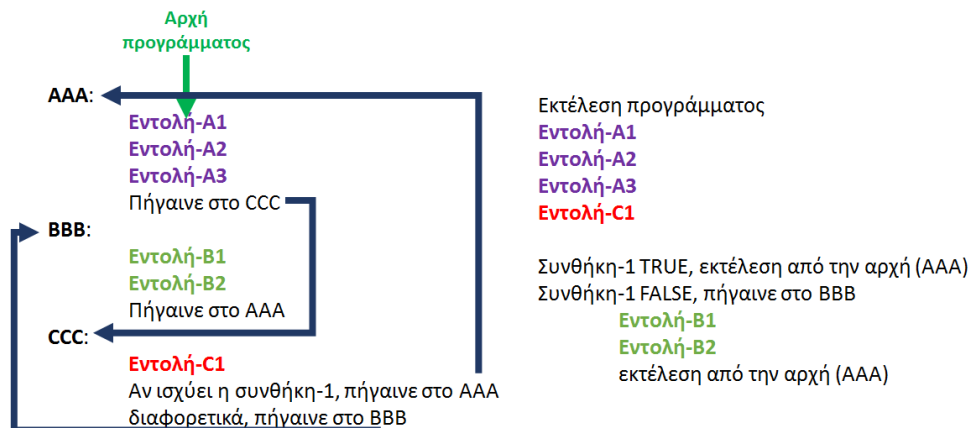


Άσκηση 4

Έλεγχος ροής εκτέλεσης – Εκτέλεση εντολών υπό συνθήκη

A. ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ

Οι εντολές άλματος κατευθύνουν τη ροή εκτέλεσης στο επιθυμητό σημείο του κώδικα. Το άλμα αυτό είναι συνήθως υπό συνθήκη και συνδυάζεται με μια εντολή σύγκρισης η οποία προηγείται. Ακολουθεί ένα γενικό παράδειγμα που δείχνει ένα σενάριο στο οποίο η εκτέλεση του κώδικα γίνεται βάσει των εντολών άλματος που χρησιμοποιούνται.



Στο παραπάνω παράδειγμα, τα σύμβολα AAA, BBB και CCC είναι ετικέτες και όχι εντολές. Επομένως, χρησιμοποιούνται απλά για να ορίζουμε συγκεκριμένα σημεία μέσα στο πρόγραμμα, προκειμένου να οδηγούμε εκεί τη ροή εκτέλεσης, όταν αυτό είναι απαραίτητο. Στη γλώσσα Assembly, χρησιμοποιούμε μια εντολή σύγκρισης (CMP) σε συνδυασμό με μια εντολή άλματος υπό συνθήκη (π.χ. je=άλμα αν προκύψει ισότητα)

Η λογική, είναι η ακόλουθη:

Σύγκριση (εντολή CMP)
Εντολή άλματος (τύπου Jump)

Η εντολή σύγκρισης, συντάσσεται ως εξής:

CMP κατ1, κατ2 ;σύγκριση περιεχομένου των καταχωρητών **κατ1** και **κατ2**

ή

CMP κατ, τιμή ;σύγκριση περιεχομένου του καταχωρητή **κατ** με την **τιμή**

Ο μικροεπεξεργαστής στην ουσία αφαιρεί το ένα όρισμα από το άλλο προκαλώντας στη συνέχεια την ενεργοποίηση κάποιων bit (ZF για την περίπτωση που το αποτέλεσμα είναι μηδέν) από τον καταχωρητή FLAGS.

Η CMP ακολουθείται από μια εντολή άλματος (αλλαγή ροής εκτέλεσης) υπό συνθήκη. Η εντολή άλματος μπορεί να περιλαμβάνει οποιοδήποτε έλεγχο, ενώ συντάσσεται με ένα όρισμα (ετικέτα) που αντιστοιχεί στο σημείο που θέλουμε να οδηγήσουμε τη ροή εκτέλεσης. Έτσι, για την αλλαγή ροής εκτέλεσης υπό συνθήκη, χρησιμοποιούμε ένα ζευγάρι των εντολών σύγκρισης και άλματος.

Παράδειγμα #1:

CMP AL,5	;Σύγκριση του AL με το 5
JE EINAI_ISO	;Αν είναι ίσο (E=Equal), τότε το πρόγραμμα συνεχίζει από το σημείο EINAI_ISO
;εντολές-F	;Εντολές που εκτελούνται αν η συνθήκη είναι ψευδής
.	;Πάντα οι εντολές που ακολουθούν μια Jump, αντιπροσωπεύουν
.	;την περίπτωση False (ψευδής)
.	
.	

Π. Παπάζογλου

JMP **termatismos**
ENAI_ISO:
;εντολές-T.

termatismos:

Παράδειγμα #2:

CMP CH,0
JE TRUE_omada_1

FALSE_omada_2:
;Ομάδα_εντολών_2
;CH≠0
JMP exit

TRUE_omada_1:
;Ομάδα_εντολών_1
;CH=0

exit:

Αν CH=0, τότε εκτελείται η ομάδα_εντολών_1 (περίπτωση TRUE), διαφορετικά η ομάδα_εντολών_2 (περίπτωση FALSE). Η εντολή άλματος JE (Jump Equal) οδηγεί τη ροή εκτέλεσης στο σημείο TRUE_omada_1 εφόσον CH=0. Διαφορετικά (περίπτωση ELSE), δεν γίνεται η αλλαγή ροής εκτέλεσης στο σημείο TRUE_omada_1 και μοιραία εκτελούνται οι εντολές που ξεκινούν από το σημείο FALSE_omada_2. Όταν εκτελεστούν αυτές οι εντολές, η ροή εκτέλεσης οδηγείται χωρίς συνθήκη στο σημείο exit έτσι ώστε να μην εκτελεστούν και οι εντολές της ενότητας TRUE_omada_1.

Κανόνες:

Μια εντολή άλματος της μορφής Jxx, ελέγχει μια συνθήκη (εφόσον έχει προηγηθεί εντολή CMP) και οδηγεί τη ροή εκτέλεσης σε συγκεκριμένη ετικέτα, εφόσον η συνθήκη είναι TRUE. Διαφορετικά, η εκτέλεση συνεχίζεται από την επόμενη εντολή (συνθήκη FALSE).

Αξίζει να σημειωθεί ότι, οι ετικέτες αντιπροσωπεύουν διευθύνσεις μνήμης στις οποίες ξεκινά η εντολή στην οποία οδηγούμε τη ροή εκτέλεσης. Έτσι, κατά τη εκτέλεση του προγράμματος, οι ετικέτες μετατρέπονται αυτόματα σε διευθύνσεις.

Αργότερα, θα δούμε τη χρήση εντολής άλματος υπό συνθήκη, χωρίς εντολή σύγκρισης.

Οι ετικέτες δεν είναι εντολές και χρησιμοποιούνται όταν επιθυμούμε να οδηγήσουμε τη ροή εκτέλεσης σε συγκεκριμένο σημείο.

Ας δούμε τώρα ένα πρόγραμμα που περιλαμβάνει ετικέτες.

Το πρόγραμμα 1 δεν έχει καμία διαφορά με το πρόγραμμα 2, διότι οι ετικέτες δεν είναι εντολές και επομένως, ο μικροεπεξεργαστής τις αγνοεί. Το πρόγραμμα 3 χρησιμοποιεί την ετικέτα start, προκειμένου να οδηγηθεί εκεί η ροή εκτέλεσης μετά την ολοκλήρωση του κώδικα.

Πρόγραμμα 1
mov bl,09
mov al,01
add al,bl
mov [0000],al
mov ax,0ffffh
mov bx,1233h
and ax,bx
inc ax
inc bx
or al,bl
mov [0002],ax

Πρόγραμμα 2
mov bl,09
mov al,01
start:
add al,bl
mov [0000],al
mov ax,0ffffh
etiketa2:
mov bx,1233h
and ax,bx
inc ax
inc bx
termatismos:
or al,bl
mov [0002],ax

Πρόγραμμα 3
mov bl,09
mov al,01
start:
add al,bl
mov [0000],al
mov ax,0ffffh
etiketa2:
mov bx,1233h
and ax,bx
inc ax
inc bx
termatismos:
or al,bl
mov [0002],ax
jmp start

Ο πίνακας που ακολουθεί, περιέχει τις εντολές άλματος που μπορούμε να χρησιμοποιήσουμε ανάλογα την περίπτωση. Η επιλογή γίνεται από τον προγραμματιστή βάσει των αναγκών του προγράμματος.

Εντολές άλματος υπό συνθήκη		
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>	-

B. ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ

1. Δοκιμάστε και σχολιάστε το ακόλουθο πρόγραμμα:

Διεύθυνση	Εντολή
cs:0100	mov al,0f0
cs:0102	inc al
cs:0104	jmp 0102

2. Δοκιμάστε το ακόλουθο πρόγραμμα:

cs:0100	mov bl,5	Αν bl=06, τότε al=9, διαφορετικά al=0
cs:0102	cmp bl,06	
cs:0105	je 010bh	α) Σχολιάστε το πρόγραμμα
cs:0107	mov al,0	β) Σχεδιάστε διάγραμμα ροής
cs:0109	jmp 010dh	
cs:010B	mov al,9	

Π. Παπάζογλου

3. Δίνεται το ακόλουθο πρόγραμμα:

Πριν τη δοκιμή του προγράμματος, καταχωρήστε στο περιβάλλον του debugger τις τιμές 02 και 01 στις θέσεις μνήμης [0000] και [0001] αντίστοιχα.

cs:0100	mov al,[0000]	α) Σχολιάστε το πρόγραμμα
cs:0103	mov bl,[0001]	β) Σχεδιάστε διάγραμμα ροής
cs:0107	cmp al,bl	γ) Δοκιμάστε το πρόγραμμα για
cs:0109	jg 0112	διάφορες τιμές στις θέσεις
cs:010B	mov [0003],0	μνήμης [0000] και [00001]
cs:0110	jmp 0117	
cs:0112	mov [0003],1	

4. Ποιο έλεγχο υλοποιεί το ακόλουθο πρόγραμμα:

```
mov al,[0000]
mov bl,[0001]
cmp al,1
je syneheia
jmp exit
syneheia:
    cmp bl,2
    je ok_label
    jmp exit
ok_label:
mov [0000],00
mov [0001],00
exit:
```