

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ**  
**ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ**  
**ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 4**

---

**1) ΑΝΑΓΝΩΣΗ ΚΑΙ ΕΓΓΡΑΦΗ ΣΤΗ ΜΝΗΜΗ**

**2) ΕΠΕΞΕΡΓΑΣΙΑ ΠΙΝΑΚΩΝ (VECTOR PROCESSING)**

Το παρακάτω πρόγραμμα πραγματοποιεί τον ορισμό τριών πινάκων A, B και C στην μνήμη και στην συνέχεια την συμπλήρωση του A σύμφωνα με την  $A[i] = B[i] + C[i]$ . Κάθε στοιχείο των πινάκων είναι μια ψηφιολέξη (1 byte) και δηλώνεται αντίστοιχα στο data segment.

```
.data
atable:      .byte 0, 0, 0, 0
btable:      .byte 10, 5, 5, 20
ctable:      .byte 10, 4, 5, 1

.text
.globl main
main:
    add $t4, $zero, $zero    # i is initialized to 0, $t4 = 0
    la $t0, atable
    la $t1, btable
    la $t2, ctable

Loop:
    add $t5, $t4, $t1        # $t5 = address of b[i]
    lw $t6, 0($t5)           # $t6 = b[i]

    add $t5, $t4, $t2        # $t5 = address of c[i]
    lw $t7, 0($t5)           # $t7 = c[i]
    add $t6, $t6, $t7        # $t6 = b[i] + c[i]

    add $t5, $t4, $t0        # $t5 = address of a[i]
    sw $t6, 0($t5)           # a[i] = b[i] + c[i]
    addi $t4, $t4, 4          # i = i + 4
    slti $t5, $t4, 4          # $t5 = 1 if $t4 < 4, i.e. i < 2
    bne $t5, $zero, Loop     # go to Loop if i <> 0

    jr $ra
    # alternate exit
exit:
    li $v0, 10
    syscall
```

Στο πλαίσιο της εργαστηριακής άσκησης θα πρέπει ο φοιτητής να μάθει τις εντολές προσπέλασης της μνήμης και των καταχωρητών (βάσει του τύπου του δεδομένου). Έτσι οι εντολές που είναι απαραίτητες να γνωρίζει ο φοιτητής είναι οι παρακάτω.

## Εντολές Load / Store

Επιτρέπεται η προσπέλαση μόνο μέσω των εντολών load και store, με την χρήση καταχωρητή

### load:

Ανάγνωση λέξης 32 bits (4 bytes) από την διεύθυνση μνήμης και αποθήκευση στον καταχωρητή

```
lw    register_destination, RAM_source
```

Ανάγνωση ψηφιολέξης 8 bits (1 byte) από την διεύθυνση μνήμης και αποθήκευση στον καταχωρητή

```
lb    register_destination, RAM_source
```

### store:

Εγγραφή λέξης 32 bits (4 bytes) στην διεύθυνση μνήμης από τον καταχωρητή

```
sw    register_source, RAM_destination
```

Εγγραφή ψηφιολέξης 8 bits (1 byte) στην διεύθυνση μνήμης από τον καταχωρητή

```
sb    register_source, RAM_destination
```

### load immediate:

Ανάγνωση μιας τιμής (value) και αποθήκευση σε καταχωρητή

```
li    register_destination, value
```

## Εντολές έμμεσης και με βάση διευθυνσιοδότησης

Επιτρέπεται η προσπέλαση μόνο μέσω εντολών load και store.

### load address:

Αντιγράφει την διεύθυνση μνήμης (όχι το περιεχόμενο) στον καταχωρητή (\$t0 στο υπόδειγμα)

```
la    $t0, var1
```

### indirect addressing:

Αντιγράφει το περιεχόμενο μιας λέξης 32 bits (4 bytes) που βρίσκεται στη διεύθυνση μνήμης (η οποία περιέχεται στον καταχωρητή \$t0 στο υπόδειγμα) στον καταχωρητή \$t2 (στο υπόδειγμα)

```
lw    $t2, ($t0)
```

Αντιγράφει το περιεχόμενο μιας λέξης 32 bits (4 bytes), η οποία βρίσκεται στον καταχωρητή \$t2 (στο υπόδειγμα), στη διεύθυνση μνήμης (η οποία περιέχεται στον καταχωρητή \$t0 στο υπόδειγμα)

**sw     \$t2, (\$t0)**

**indexed addressing:**

Αντιγράφει το περιεχόμενο μιας λέξης 32 bits (4 bytes) που βρίσκεται στη διεύθυνση μνήμης (η οποία περιέχεται στον καταχωρητή \$t0 στο υπόδειγμα) αυξημένη κατά 4 (βάση) στον καταχωρητή \$t2 (στο υπόδειγμα).

**lw     \$t2, 4(\$t0)**

Αντιγράφει το περιεχόμενο μιας λέξης 32 bits (4 bytes), η οποία βρίσκεται στον καταχωρητή \$t2 (στο υπόδειγμα), στη διεύθυνση μνήμης (η οποία περιέχεται στον καταχωρητή \$t0 στο υπόδειγμα) μειούμενη κατά 12 (η βάση στο υπόδειγμα).

**sw     \$t2, -12(\$t0)**

Αναλύστε κατά την διάρκεια του εργαστηρίου τον κώδικα και εκτελέστε τον στο QtSpim. Λαμβάνοντας υπόψη ότι ο MIPS είναι little endian, ποια η αναμενόμενη έξοδο και ποια αυτή που προέκυψε από την εκτέλεση του προγράμματος; Συμπληρώστε τον πίνακα που ακολουθεί τον κώδικα, αντικαθιστώντας στο πρόγραμμα όπου χρειάζεται.

A[]	B[]	C[]	Αναμενόμενο A[]	Προκύπτων A[]
0, 0, 0, 0, 0	10, 5, 5, 20	10, 4, 5, 1		
0, 0, 0, 0, 0	10, 252, 251, 10	10, 5, 6, 1		

**ΕΡΓΑΣΙΑ 1**

Να εξηγήσετε ποιο είναι το πρόβλημα που δημιουργείται από αυτόν τον κώδικα και πως επιλύεται.

**ΕΡΓΑΣΙΑ 2**

Να δημιουργήσετε ένα πρόγραμμα το οποίο θα διαβάζει το string textlow, το οποίο θα έχετε ορίσει στο text segment και θα περιέχει το κείμενο «This is a sample text!», και θα μετατρέπει κάθε γράμμα από πεζό (μικρό) σε κεφαλαίο. Για την τροποποίηση προτείνεται η αντιμετώπιση κάθε χαρακτήρα ως στοιχείου πίνακα και η μετατροπή ένα προς ένα σε κεφαλαίο. Προσοχή θα πρέπει να δοθεί στην περίπτωση που ο χαρακτήρας είναι ήδη κεφαλαίο γράμμα ή ειδικός χαρακτήρας, οπότε και δεν θα πρέπει να κάνει τίποτα. Δοκιμάστε το πρόγραμμα με διαφορετικά strings για να επιβεβαιώσετε την ορθή λειτουργία του. (Θεωρείστε ότι είναι γνωστό το μήκος του string)

**ΠΑΡΑΔΟΣΗ**

Έχοντας τα παραπάνω ως πληροφορία, να παραδώσετε την αναφορά όπως έχει ανακοινωθεί στο e-class.