



Εργαστήριο 1

Εαρινό Εξάμηνο 2010-2011

Στόχοι του εργαστηρίου

- Γνωριμία με το περιβάλλον του MARS
- Δομή ενός προγράμματος σε MIPS
- Κατανόηση απλών εντολών MIPS
- Απλή δομή επανάληψης

Ο προσομοιωτής MARS

Προγράμματα γραμμένα σε γλώσσα Assembly του MIPS μπορεί να τα δοκιμάσει κανείς και να παρακολουθήσει πώς τρέχουν χρησιμοποιώντας τον "προσομοιωτή" MARS (MIPS Assembler and Runtime Simulator).

Οι προσομοιωτές είναι προγράμματα υπολογιστή που προσπαθούν να συμπεριφέρονται όσο πιο παρόμοια γίνεται, από ορισμένες απόψεις, με ένα φυσικό σύστημα. Εν προκειμένω, ο SPIM συμπεριφέρεται σαν ένα τσιπάκι MIPS από την άποψη των περιεχομένων των καταχωρητών και της μνήμης μετά την εκτέλεση κάθε εντολής.

Μπορείτε να κατεβάσετε τον MARS εδώ:

http://courses.missouristate.edu/KenVollmar/MARS/MARS_4_1_Jan_2011/Mars_4_1.jar

Επίσης, μπορείτε να βρείτε ένα tutorial για τον MARS εδώ:

<http://courses.missouristate.edu/KenVollmar/MARS/CCSC-CP%20material/MARS%20Tutorial.doc>

Τις εντολές assembly του επεξεργαστή MIPS μπορείτε να τις βρείτε στο αντίστοιχο instruction set το οποίο βρίσκεται στο site του μαθήματος.

Άσκηση 1

α) Αντικείμενο της παρούσας άσκησης είναι να εξοικειωθείτε με τη χρήση του MARS (και με τη γλώσσα Assembly του MIPS).

Για τον σκοπό αυτό, αντιγράψτε το παρακάτω πρόγραμμα στο IDE του MARS, μελετήστε το και προσομοιώστε το.

```
# This is a test lab to get used to the MARS simulator

.text
.globl main          # label "main" must be global

main:
add  $s0,$0,$0       # Reset register $s0
addi $t0,$0,1        # Put the value 1 in register $t0
add  $s1,$t0,$s0      # $s1 = $t0 + $s0
addi $s1,$s1,-6       # $s1 = $s1 - 6
addi $s0,$s0,2        # $s0 = $s0 + 2
sub  $t4,$s1,$s0      # $t4 = $s1 - $s0

# print the integer value of $t4
move $a0, $t4
li $v0,1
syscall

#exit program
li $v0,10
syscall
```

- Το κομμάτι κάθε γραμμής μετά το # είναι σχόλια.
- Οι γραμμές μετά το .text είναι εκτελέσιμος κώδικας (σε αντίθεση με αριθμητικά δεδομένα στη μνήμη, που εδώ δεν έχουμε).
- Η γραμμή .globl main λέει στον MARS να βάλει την ετικέτα (label) main στον πίνακα καθολικών (global) συμβόλων.
- Η γραμμή main: ορίζει την ετικέτα main σαν τη διεύθυνση μνήμης όπου θα τοποθετηθεί αυτό που ακολουθεί ακριβώς μετά, στην περίπτωσή μας η πρώτη εντολή (add) του προγράμματός μας.

β) Επαναλάβετε την διαδικασία για τον παρακάτω κώδικα ο οποίος περιέχει δομή επανάληψης (loop)

```
# This is a test lab to get used to the MARS simulator

.text
.globl main                # label "main" must be global
main:
    add $t0,$0,$0          # Reset register $t0 (Counter)
    addi $t1,$0,10         # Put value 10 in register $t1
                           # (Number Of Loops)
loop:
    addi $t0,$t0,1         # $t0=$t0 + 1
    bne $t0,$t1,loop       # if($t0!=$t1) go to "loop"

    #exit program
    li $v0,10
    syscall
```

Άσκηση 2

Για το επόμενο εργαστήριο αυτό που έχετε να κάνετε είναι να μελετήσετε τους 2 κώδικες καθώς και το πώς μπορείτε να τους προσομοιώσετε στον MIPS με την χρήση του MARS. Θα πρέπει να γράψετε ένα πρόγραμμα το οποίο, βασιζόμενο στα δύο παραδείγματα που δώσαμε, θα πρέπει να εκτυπώνει όλους τους ζυγούς αριθμούς από το 2 έως και το 20.

Χρησιμοποιείτε την εντολή εκτύπωσης *syscall* όπως στο πρώτο παράδειγμα.