# Assembly Language Programming

## Lab 1: Introduction to MIPS Assembly Programming

Purpose

Learn to develop assembly language programs.

Method

Translate a simple program from a higher-level language into assembly language.

Files to use

What to Hand In

Completed oddeven.asm, swap.asm

1. Write a MIPS program with comments that loads a random integer from memory (use the **.word** assembler directive) and determines whether the number is odd or even. If the number is odd write a ‘1’ to register 15 ($t7) if the number is even write a ‘0’ to register 15.

1. Write a MIPS program with comments that swaps the contents of the two consecutive memory locations with the two other consecutive locations.

Example : (2000) : 45 (3000) : 78

(2004) : 67 (3004) : 90

before execution

(2000) : 78 (3000) : 45

(2004) : 90 (3004) : 67

after execution

Use labels in your program instead of absolute addresses as below

first\_loc .word 45,67

sec\_loc .word 78,90

To load the addresses to registers, use the pseudo instruction la (load address) as below :

la $1, first\_loc # address first\_loc is loaded to register 1