**CS225 Computer Organization and Assembly Language**

**Assignment # 1 (FALL 2014)**

**Due Date: 12-Nov-14 start of the class**

* **Due date and time will be strictly followed**
* **There will be a quiz based on this assignment + topics covered in the class on 12-Nov-2014 (come prepared)**

1. Write instructions for executing A = ( B+ C) – D in stack architecture, accumulator architecture and general purpose register architecture (register – to - register)
2. Define the R-format instruction:
   1. What are the various fields?
   2. What is the basic lay-out?
   3. What does each field represent?
   4. Are all of the bits used / needed for each field?
   5. Given an example of an R-format instruction in MIPS assembler and the corresponding machine code
3. Define the I-format instruction:
   1. What are the various fields?
   2. What is the basic lay-out?
   3. What does each field represent?
   4. Why this instruction format needed as opposed to the R-format?
   5. Give an example of an I-format instruction in MIPS assembler and the corresponding machine code
4. What is the range of addresses for conditional branches in the MIPS ISA? Explain
5. Convert the following C statement into MIPS 32 assembly language. Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively.

**f = g - A[B[4]];**

Note: we are not yet familiar with multiply instruction.

1. Write the following code sequence into MIPS assembler

**x = x + y + z – q;**

Assume that x, y, z, q are stored in registers $s1 - $s4

1. In MIPS assembly, write an assembly language version of the following C code segment:



At the beginning of this code segment, the only values in registers are the base address of array A and B in registers $a0 and $a1. Avoid using multiplication instructions – they are unnecessary

1. Use the register and memory values in the table below for part a and b. Assume a 32-bit machine. Assume each of the following question starts from the table values; that is, DO NOT use value changes from one question as propagating into the future parts of the question



1. Give the values of R1, R2 and R3 after this instruction **add R3, R2, R1**

1. What values will be in R1 and R3 after this instruction is executed: **load R3, 12 (R1)**
2. What values will be in the registers after this instruction is executed : **addi R2,R3, 16**