Rose-Hulman Institute of Technology
Computer Science and Software Engineering
Prof. Archana Chidanandan

Homework 2
(Assembly Language and Machine Language)
Maximum points : 40 points

Directions
This assignment is due Friday, Dec. 19th for Sections 1 and 2. Submit your solutions on a separate sheet of paper.

Learning Objectives
In the process of completing this homework assignment, students will develop their abilities to

- Translate assembly language instructions into machine language
- Interpret a sequence of bits and determine what it represents.
- Determine addressing in branches and jumps.
- Interpret instruction formats and fields.
- Implement algorithms procedural abstraction in assembly language.

Problems
1. [6 points] In Homework 1, you wrote a program that could calculate the product of two numbers. Re-write the program, so that the main program reads the two values from memory, uses procedure "Product" to determine the product of the two numbers, and writes the product into a location in memory. You must also write the procedure "Product". Again, you may NOT use the MIPS `mult, multu, mul, mulo or muluo` instructions. You MUST use a loop structure. The program must follow the MIPS register conventions (page 138, Figure 3.11).

2. [8 points, 4 each] In the in-class exercise, we looked at a procedure "Move" (p04-2.asm), that uses a position component and a velocity component to determine a new position which lies within the specified boundaries. The procedure (the name has been changed to "MoveObject") is provided in the file, hmwk02-2.asm.

a) Write a procedure "MoveBall" that calls the "MoveObject" procedure to update the "x" and "y" positions of the ball.
b) Write a main program that initializes the x and y positions and velocities of the ball and then calls the procedure "MoveBall" repeatedly to update the "x" and "y" positions.

The program must follow the MIPS register conventions (page 138, Figure 3.11). You may not modify the "MoveObject" procedure.

3. [6 pts, 2 each] Bits have no inherent meaning. However, given the rules to interpret them, a sequence of bits can represent different values and have meaning. Given the bit pattern, 0x8dad0000, what does it represent, assuming that it is

a) an unsigned integer?
b) a 2's complement number?
c) a MIPS instruction?

Hint: Use a calculator where appropriate.

Rose-Hulman Institute of Technology
Computer Science and Software Engineering
Prof. Archana Chidanandan

4. [9 pts, 3 each] For the following MIPS assembly language instructions, list the machine language fields and their binary and hexadecimal values:

```
a) ori    $2, $0, 10
b) jal    main   # main is at address 0x00400020 and the current instruction is in
                 # the same page as "main"
c) add    $t7, $t2, $0
```

5. [4 pts] Assume that the MIPS instruction beq $t0, $s0, Label is located at address 0x0400 5678, and that Label is located at address 0x0400 1234. What will the binary value of the address field be? *Hint*: Remember that the offset is relative to the instruction following the branch, and that all branch targets must be word aligned.

6. [4 pts] Assume that the MIPS instruction j Label is located at address 0x0400 5678, and that Label is located at address 0x0400 1234. What will the binary value of the target address field be? *Hint*: Remember that all branch targets must be
word aligned.

7. [3 pts] What sequence of MIPS instructions starting at address 0x0400 5678 could be used to branch to address 0x4400 1234?