CS224 – Fall 2017 - Lab #3

Doing Single Precision Floating Point Operations by Integer Arithmetic and Bit Manipulation MIPS Instructions

Dates:

Section 1:	Wed, Nov. 1st	13:40-17:30 in EA-Z04
Section 2:	Fri, Nov. 3rd	13:40-17:30 in EA-Z04
Section 3:	Wed, Nov. 1st	08:40-12:30 in EA-Z04
Section 4:	Thu, Nov. 2nd	13:40-17:30 in EA-Z04

Purpose: In this lab you will study floating point number representation and will write a program to do floating point multiplication or floating point addition, using the MIPS *integer* arithmetic and bit manipulation instructions, and using *integer* registers for the calculation.

Summary

Part 1 (30 points): Preliminary Report/Preliminary Design Report: Floating point conversion and utilities (Due date of this part is the same for all)

Part 2 & 3 (70 points): Floating Point Utilities

DUE DATE OF PART 1: SAME FOR ALL SECTIONS Dear students please bring and drop your preliminary work into the box provided in front of the lab before 3:59 pm on Tuesday October 31st. No late submission!

LAB WORK SUBMISSION TIMING: You have to show your lab work to your TA by **12:15** in the morning lab and by **17:15** in the afternoon lab. Note that you cannot wait for the last moment to do this. If you wait for the last moment and show your work after the deadline time 20 points will be taken off.

If we suspect that there is cheating we will send the work with the names of the students to the university disciplinary committee.

Part 1. Preliminary Work / Preliminary Design Report
(30 points, contains sections 1 to 3 & each section 10 points)

You have to provide a neat presentation prepared by Word or a word processor with similar output quality (for some Latex is the science fiction editor for undergraduates, you may challenge yourself by trying it). At the top of the paper on left provide the following information and staple all papers. In this part provide the program listings with proper identification (please make sure that this info is there for proper grading of your work, otherwise some points will be taken off).

CS224
Section No.: Enter your section no. here
Fall 2017
Lab No.
Your Full Name/Bilkent ID

You should prepare the following work with a word processor (e.g. Microsoft Word).

1. **[10 points]** Choose a positive floating point number of your choice between 100 and 200 with a fractional part and express it in single precision IEEE 754 floating point format and also in double precision floating point format. Give your answer in hexadecimal. Show your work in detail. [Note: Your chosen number must not be the same as any other student's number!]

2. **[10 points]** Choose a negative floating point number of your choice between -300 and -400 with a fractional part and express it in single precision IEEE 754 floating point format and also in double precision floating point format. Give your answer in hexadecimal. Show your work in detail. [Note: Your chosen number must not be the same as any other student's number!]

3. **[10 points]** Consider the following code segment, where the address of main is 0x44444444

```
    .text
main:
    bge     $t0,$t1,Next
    add     $t1,$t2,$t3
    b       main
    add     $t1,$t2,$t3
    j       main
Next:
```

   By hand, generate the object code for bge, b, and j instructions in hexadecimal notation, showing both the address and the value of each MIPS machine instruction. To receive credit, you must do the calculations and show your work in detail. Remember that pseudo-instructions may expand to more than one real MIPS instruction.

Notes:

- The above 4 items (cover page plus the 3 individual parts) should be submitted in 2 formats: as a hard-copy, printed out and stapled in the upper-left corner, and as an electronic document (in .doc or .pdf format), to be submitted online to Unilica Assignments.
- You may store the messages in .data segment.
- Make sure you initialize everything - do not assume that memory is initialized to 0.
- The utility programs that you will write must be callable from the main program by the names given in bold. *They must obey the MIPS convention for register usage, and thus will have no unexpected side-effects (we will test your functions in our context so if there is a problem with preserved registers your program can potentially execute incorrectly).* Your MIPS code should be well-commented (meaning most lines will have explanatory comments), it will use white space and indenting for easy readability, and each routine will have an explanatory header at the top, naming the registers and storage used and what they contain, and describing the algorithms used. Together these 3 characteristics make for self-documenting code; this style of code can be easily understood by anyone.

Part 2. Main Program and User I/O (25 points)

For this lab, you will need to write the MIPS program according to the following definitions:

1. **[5 points] Program structure**: Your program must be well-structured with proper functions/procedures. It should use software engineering principles such as modularity, hierarchy, abstraction, reuse, etc to organize the design and its implementation.

2. **[5 points] Code readability/understandability**: Your code must use line comments, spacing/indentation, and headers, so that it is *self-documenting*. Headers should precede code blocks, explaining what the block does, and the storage resources it will use. Line comments should be regular and frequent. Spacing/indentation must be used to enhance readability. In summary, your self-documenting code should be completely understandable to the reader.

3. **[15 points] User I/O:** Your program's input-output should be easy for the user to understand, messages should be clear, the choices obvious and not confusing, sequences of messages designed to enhance understanding, etc. You should spend time designing a good user interface (UI) for your program. This is worth 10% of the lab grade.

   Program sequence and function:
   a. Get the two inputs.
   b. If an input number is equal to zero, stop execution.
   c. Output the result of the operation (**given by the TA**) that your program computed.
   d. Then use the actual MIPS floating point instruction to calculate the result, and output it.
   e. Go to step a.

Note: step d is the only place in your program where a floating point arithmetic instruction is allowed.

Note: Result rounding, overflow and underflow detection are optional, for extra credit. Rounding (according to the IEEE standard "round to nearest even"), conversion of overflow results to infinity and conversion of underflow results to de-normalized numbers will earn some extra points, for those who do these.

Part 3. Special Floating Point Operation (35 points)

A. It will be given on the board by the TA

B. Programming Hints

1. The single-precision floating point number 0 is represented by 0x0000000000 (32 0s, like integer 0).

2. For accessing floating point data as an integer, and for I/O, consider the following code segment.

```
        .data
num1:   .word 0

        .text
        …
        li  $v0,6          # Read single prec. floating point no.
        syscall             # input is returned in register $f0.

        s.s   $f0,num1      # Save input float number to memory.
   # Now you can manipulate the floating point data
   # like an integer data by accessing the memory location num1.

        mfc1  $t0,$f0
   # Using mfc1 you can directly Move from floating
   # point register $f0 to an integer register as shown above.

        movf.s $f12,$f0    # move number in $f0 to reg. $f12
        li  $v0,2          # Print the contents of register $f12.
        syscall
```

3. Floating Point Arithmetic

   Please refer to your class notes, to the textbook Chapter 5 material about floating point, and also look at the slides provided in the Unilica course web site. Needless to say, the Web has a wealth of information about floating point arithmetic with examples.

It is strongly suggested that you develop your algorithm by hand, and test it by hand with example numbers, before you attempt to program it in MIPS.

Part 4. Submit your code for MOSS similarity testing

Submit your MIPS codes for similarity testing to the Unilica > Assignment specific for your section. You will upload one file: **name_surname_SecNo_MIPS.txt** created in the relevant parts. Be sure that the file contains exactly and only the codes which are specifically detailed Part 1 to Part 3, including Part 1 programs (your paper submission for preliminary work must match MOSS submission). Check the specifications! *Even if you didn't finish, or didn't get the MIPS codes working, you must submit your code to the Unilica Assignment for similarity checking.* Your codes will be compared against all the other codes in the class, by the MOSS program, to determine how similar it is (as an indication of plagiarism). So be sure that the code you submit is code that you actually wrote yourself ! All students must upload their code to Unilica > Assignment while the TA watches. Submissions made without the TA observing will be deleted, resulting in a lab score of 0.

Cleanup

1) After saving any files that you might want to have in the future to your own storage device, erase all the files you created from the computer in the lab.
2) When applicable put back all the hardware, boards, wires, tools, etc where they came from.
3) Clean up your lab desk, to leave it completely clean and ready for the next person who will come.

LAB POLICIES

1. You can do the lab only in your section. Missing your section time and doing in another day is not allowed.
2. Students will earn their own individual lab grade. The questions asked by the TA will have an effect on your individual lab score.
3. Lab score will be reduced to 0 if the code is not submitted for similarity testing, or if it is plagiarized. MOSS-testing will be done, to determine similarity rates. Trivial changes to code will not hide plagiarism from MOSS—the algorithm is quite sophisticated and powerful. Please also note that obviously you should not use any program available on the web, or in a book, etc. since MOSS will find it. The use of the ideas we discussed in the classroom is not a problem.
4. You must be in lab, working on the lab, from the time lab starts until your work is finished and you leave.
5. No cell phone usage during lab.
6. Internet usage is permitted only to lab-related technical sites.
7. For labs that involve hardware for design you will always use the same board provided to you by the lab engineer.