CMPT350: Lab Demo 6

Floating point arithmetic and representation in MIPS *1%*

Due: November 2, 2022 @ 17:00

So far, we've only worked with integer values in the lab. Of course, we are not restricted to integers, and much of scientific computing requires the use of floating point values. This lab will demonstrate basic floating point arithmetic and representation in MIPS.

Laboratory Procedure:

On the course Moodle page, look for the Demo 6 assignment in the Laboratories section. Inside is a premade skeleton MIPS file, `demo6.s.` Save this somewhere handy on your machine. You may use both the command line version, `spim,` and the GUI version, QtSpim to run your program. As well, you will need a text editor to edit your programs.

Assignment Requirements:

- Read in two *single precision floating point* values using the appropriate syscalls (refer to the MIPS Syscall table for which syscalls to use for floating point input and output, and how to use them).
- With these two floating point values, perform and display the results of the following arithmetic operations:
  - Addition
  - Subtraction
  - Multiplication
  - Division
- Find the sign of the sum of the two values without branching, using bitwise instructions. (e.g. if the two inputs are 3.14 and –6.28, the sum is –3.14 and the sign is negative)

*** **NOTE:** Your code **must interpret and execute** within the **laboratory environment**. Failure to do so will result in a **mark of ZERO** for the program. ***

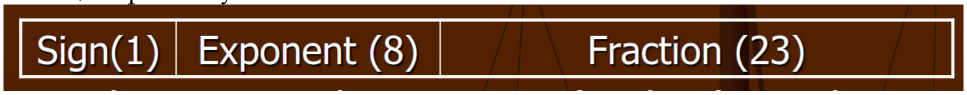
Submission:

Your programs should be named using your name and end in a .s extension. For example, BakerDemo6.s

When you feel your programs meet all of the above requirements, call the lab coordinator to demonstrate your program, and create a zip archive with your code submission and trace file(s) included. An easy way to do this is with the `zip` command. For example, if your solution file and trace file are in a folder called `baker`, then the command `zip -r baker.zip baker/` will create a zip file containing your submission files called baker.zip (Replace this with your name when creating your zip file). Upload this zip file to Moodle.

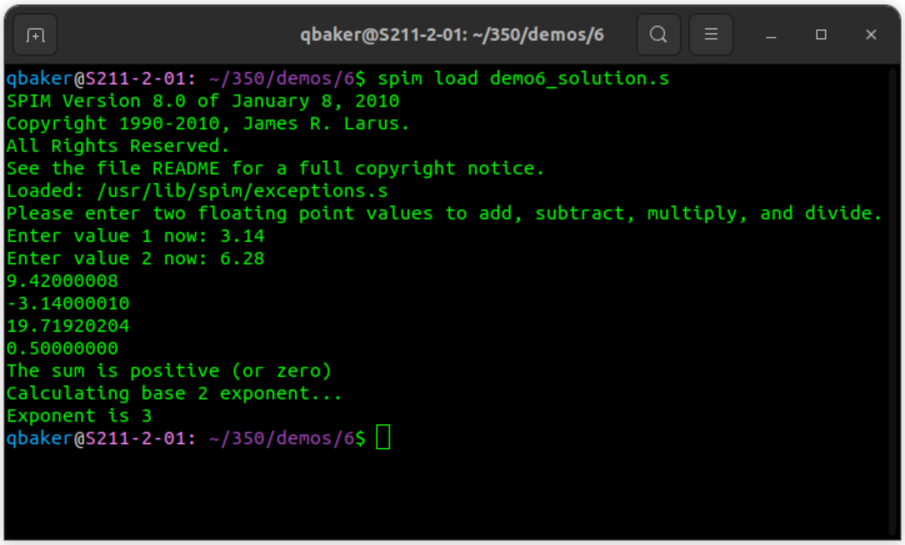
Hints:

- Recall that floating point operations take place on a separate coprocessor, the floating point co-processor (also known as C1). This processor has its own instruction set, meaning that traditional instructions (move, add, etc) won't work on floating point registers. Thankfully, many, if not all, of the common instructions have floating point equivalents
- As well, remember the floating point coprocessor is used to perform floating point arithmetic. If you need to do other work with the contents of a floating point register, you'll need to move/convert and move the floating point register contents over to the main processor into a regular register.
- Recall 32-bit floating point representation in MIPS – the 32 bits are broken down into 3 sections, in order of bit magnitude: sign, biased exponent, and mantissa, using 1, 8, and 23 bits, respectively.

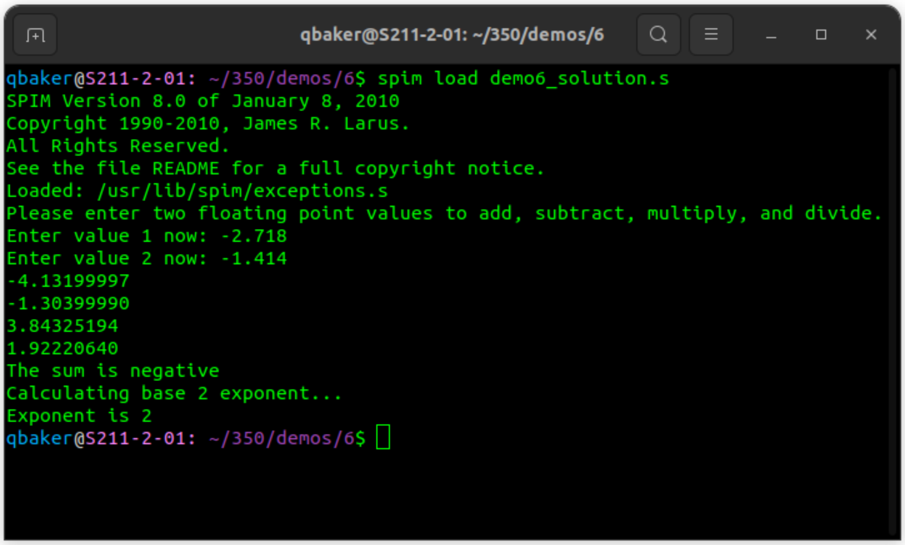


- The easiest way to check the sign using bitwise operators is with logical shift operators (`sll` and `srl`), though this can be achieved with other bitwise operators as well. Any bitwise approach is acceptable.

Example Output:



```
qbaker@S211-2-01: ~/350/demos/6$ spim load demo6_solution.s
SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: /usr/lib/spim/exceptions.s
Please enter two floating point values to add, subtract, multiply, and divide.
Enter value 1 now: 3.14
Enter value 2 now: 6.28
9.42000008
-3.14000010
19.71920204
0.50000000
The sum is positive (or zero)
Calculating base 2 exponent...
Exponent is 3
qbaker@S211-2-01: ~/350/demos/6$ 
```



```
qbaker@S211-2-01: ~/350/demos/6$ spim load demo6_solution.s
SPIM Version 8.0 of January 8, 2010
Copyright 1990-2010, James R. Larus.
All Rights Reserved.
See the file README for a full copyright notice.
Loaded: /usr/lib/spim/exceptions.s
Please enter two floating point values to add, subtract, multiply, and divide.
Enter value 1 now: -2.718
Enter value 2 now: -1.414
-4.13199997
-1.30399990
3.84325194
1.92220640
The sum is negative
Calculating base 2 exponent...
Exponent is 2
qbaker@S211-2-01: ~/350/demos/6$ 
```