Dept. of CSE, Bennett University

ECSE217L – Microprocessor and Computer Architecture

Lab Assignment – 9

In this lab, you will learn to use floating point numbers in MIPS. There is separate system call to print floating point numbers.

To use the floating-point numbers on MIPS a separate chip called coprocessor 1 is used. The coprocessor1 is also called FPA (Floating Point Accelerator). Modern MIPS chips include floating point operations on the main processor chip. But the instructions sometimes act as if there were still a separate chip.

- There exist total 32 single precision (32 bit) floating point registers in MIPS.
- The floating-point registers are named as $f0, $f1, … $f31.
- Like the '$zero' register, $f0 is not any special register, it can hold any bit pattern, not just zero.
- Single precision floating point load, store, arithmetic, and other instructions work with these registers.
- The general-purpose registers cannot be used in floating-point instructions.
- Only floating-point instructions may be used with the floating-point registers.

Let us have a simple example first to print the value of 'pi'.

```
.data
        PI: .float 3.14
.text
        li $v0, 2          # syscall 2 for printing floating point number
        l.s $f12, PI       # argument to print float in $f12  ## instead of l.s you can also use lwc1
        syscall

        li $v0,10          # code 10 is for terminating the program
        syscall
```

You will observe in the example that the system call code to print a float value is 2, which has been stored in register v0. The corresponding argument will be stored in register $f12.

Now write the following programs in MIPS.

1. Write a program to print the value of 'pi' as a floating-point number.

   Example: The value of pi is: 3.14

2. Write a program to print the value of 'pi'. Now take any floating-point number as user input at run-time and print them.

Example: The value of pi is: 3.14
Enter any floating-point number: 2.5
The user-entered floating-point number is: 2.5

3. According to problem 1 and problem 2, you have two floating point numbers. Now add them and print the addition result. The addition result should be also represented in floating point numbers.

   (Hint: floating-point arithmetic operation will be done by add.s, sub.s, mul.s … etc.

   Example: The value of pi is: 3.14
   Enter any floating-point number: 2.5
   The user-entered floating-point number is: 2.5
   The Addition result is 5.6400003

4. Write a program to swap to floating-point numbers.

   (Inst: No need to take the floating-point values as user-input. You can take them as static data)

   Example: Before Swapping, 1st value: 8.32
   Before Swapping, 2nd value: 0.62
   After Swapping, 1st value: 0.62
   After Swapping, 2nd value: 8.32

Submission Instructions:

- Submit your .asm files in LMS according to the deadline. Save all the files as per the format **RollNo_Lab#_QuestionNo.asm (Example: E21CSE632_Lab09_Q2.asm)**.
- Write your Name and Roll No. as comment before starting of each program.
- Make it sure that in each program, you have mentioned enough comments regarding the explanation of program instructions.
- In the LMS please submit in your respective batch's submission portal. Submission in other batch's submission portal will not be checked.
- Write your Name and Roll No in the .m file itself (Use # to insert comment lines). Without this you will score zero for that particular question.
- Late submission will lead to penalty.
- Any form of plagiarism/copying from peer or internet sources will lead penalty.