

Indian Institute of Technology, Delhi

FALL, 2012

COMPUTER ARCHITECTURE

Homework 2

(DEADLINE: October 12th, 11:59 PM)

NOTE: All answers need to be brief and to the point.
Please make any assumptions that you deem to be reasonable.
Total Marks: 50

Policies:

1. If you are late by n days and m minutes, then we will assume that you are $n + 1$ days late. There is a 10 mark penalty per day.
2. Do your homework **individually**.
3. There is a penalty of 15 marks if your file is in the wrong format.

Submission:

- For IIT Delhi : All submissions will be done through Moodle. <https://jaijaivanti.cse.iitd.ernet.in>
- For IIT Ropar : Mail the submission to csl211.ropar@gmail.com
- **Delhi and Ropar:** Submit a single (.s) file whose name follows the format <entry_number>.s. For example, if your entry number is (cs101090), then your file name should be : cs101090.s

Operand	Register	Format
FP Operand 1	r7	IEEE 754 format
Operator	r8	Can be [+ or - or *]. ASCII Format
FP Operand 2	r9	IEEE 754 format

Table 1: List of Populated Registers

;

1. You need to implement floating point addition, multiplication, and subtraction in ARM assembly. Please find the file `sample.s` attached. The `sample.s` assembly file reads a file called `input.txt` (do not change the file name). The file `input.txt` contains a single line with the following format.

<FP Operand 1> Operator <FP Operand 2>

The Operator can be one of the following : (+, -, *). Some examples are shown below. (One example in each separate line)

```
2 + 3
2e-19 * 3.08e-12
-2.34e-18 * -2e12
```

The file `sample.s` has functionality to read the `input.txt` file, and parse its operands. You need to go directly to the line (`### INSERT YOUR CODE HERE`) in the `process` function.

We have already populated three registers for you (See Table 1).

You need to do the appropriate computation, and put the result in register `r0` (IEEE 754 format). Here are the rules.

- Do not touch any existing code.
- You need to insert your code between the lines (`### INSERT YOUR CODE HERE`) and (`### YOUR CODE ENDS HERE`).
- You can also write extra functions to modularize your code and call them from within the (`###`) regions.
- You are not allowed to use any external function, or library call. **USE ONLY STANDARD ARM INSTRUCTIONS.**
- Do not use any built in floating point support in ARM. This includes dedicated floating point registers, floating point operations, or any other operation/entity that is aware of the FP nature of operands. You need to use standard integer operations (adds, multiplies, and shifts).
- You need not consider denormal numbers.
- For IIT Delhi: The program should run using `arm-elf-run` on `palasi`.
(`/misc/research/teaching/srsarangi/gnuarm-4.0.2/bin/arm-elf-run`)
- For IIT Ropar: The program should run using `arm-elf-run` on your server.
- For rounding, use the nearest (even) approach.

You need to do the computation accurately, perform rounding if required, and consider all the cases. (50 marks)