

CS 50 Homework 2: Floating Point Conversion in RISC-V

The IEEE floating point standard is a rich and complex design, and the best way to really understand how it represents numbers is to build utilities to manipulate floating point numbers as raw bits. Likewise, the best way to understand assembly is to convert existing C code into assembly.

In this assignment you will write a RISC-V implementation of `as_ieee_16` from the previous homework. To aid in your debugging there is a copy of the Homework 1 autograder for you to use to make sure your C code is correct, and on Sunday the 28th there will be a released copy of the official solution for `floating.c`. You can use either your own code or that released code as a starting point.

You will only turn in your `floating.s` file, and you must not change anything in the file ABOVE the line in the code. The file already contains a main function that reads the arguments passed in to main, converts them to hex, and applies the `as_ieee_16` function, which means for testing you should simply add arguments for the different magic numbers you want to test.

This assignment is due on May 6th at 2000 PDT (10:00 pm Davis time). You must work on this assignment individually.

Expected Homework Outcomes

1. Really ***understand*** the structure and complexity of IEEE Floating Point
2. Understand the process of converting C code into RISC-V assembly language programs

Hints

You are strongly encouraged to build your own copy of the RISC-V simulator. Instructions are available here:

https://docs.google.com/document/d/1gibQ88EyITDgLLui_Flv1qaE8c6ilb2NHsmJ_bTZ-Fc/edit

In particular the graphical version of the simulator allows you to more easily examine what is going on, add debugging breakpoints, and provide general visibility.