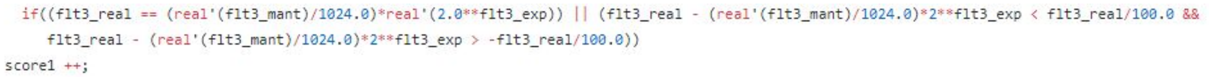
Yixin Chen A13824695
Yikuan Xia A14009432

Note: In our design, one simulation only runs one program. We passed all the test cases provided in the original test bench for each of the three programs. We also passed all the test cases provided by TA during the demo.

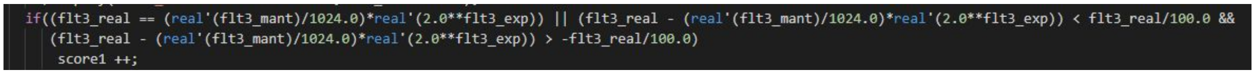
Instruction on running a program:

1. Run assembler.py to generate machine code for each of the three programs
2. Open InstROM.sv, you will see three lines of “$readmemb”. Uncomment the line that corresponds to the program you are going to run. Comment out the other two lines.
3. Open LUT.sv, you will see three blocks of lookup tables. Uncomment the block that corresponds to the program you are going to run. Comment out the other two blocks.
4. Launch Modelsim and create a project. The project location is the folder containing all the .sv files.
5. Click “add existing file” and add all the .sv files to the project
6. Compile and simulate the test bench (xxx_tb.sv) of the program you want to run.
7. If you want to simulate another program, repeat step 2, 3 and 6

Modification on test benches:

1. For each of the test benches, we changed the dummy DUT to our own TopLevel module.
2. For each of the test benches, we renamed data_mem1 to data_mem, mem_core or my_memory to core.
3. For the test bench for float_add, we changed memory space from 128-131 to 8-11, from 132-133 to 12-13, according to the data memory locations given in Lab 1 prompt.
4. For the test bench for float_add, we change the if statement that checks score1.
   The code before change is:
   

```
      if((flt3_real == (real'(flt3_mant)/1024.0)*real'(2.0**flt3_exp)) || (flt3_real - (real'(flt3_mant)/1024.0)*2**flt3_exp < flt3_real/100.0 &&
        flt3_real - (real'(flt3_mant)/1024.0)*2**flt3_exp > -flt3_real/100.0))
    score1 ++;
```

   The code after change is:
   

```
  if((flt3_real == (real'(flt3_mant)/1024.0)*real'(2.0**flt3_exp)) || (flt3_real - (real'(flt3_mant)/1024.0)*real'(2.0**flt3_exp)) < flt3_real/100.0 &&
    (flt3_real - (real'(flt3_mant)/1024.0)*real'(2.0**flt3_exp)) > -flt3_real/100.0)
     score1 ++;
```

   Our reason for making this modification is that we found that in some test cases where the result should be nonzero, the evaluation of `(real'(flt3_mant)/1024.0)*2**flt3_exp` is zero. So we deduced that this expression is problematic