

# ECE/CS 466 Project 1

In this part of the project, you are asked to learn how to use the popular architectural simulator, SimpleScalar, to analyze the performance impact of varying architectural parameters.

First, you need to download “simplesim-3v0e.tgz” from [www.simplescalar.com](http://www.simplescalar.com) and install the simulator on your computer. If your computer is windows-based, you need to install the simulator either under “cygwin” or a Linux-like virtual machine.

When you install the codes, please use the PISA configuration (follow the installation instruction in file “README” and use “make config-pisa”). To verify the installation, run the testing with command of “./sim-safe tests/bin.little/test-math”. You should see printed results as shown in the class.

Next, you will use SimpleScalar to run a SPEC2000 benchmark program, equake. This is a floating-point application for Seismic Wave Propagation Simulation. You can download the PISA binary code of equake (equake.ss) and the input file (equake.in) from the blackboard. It’s better to put the two files under the same directory of sim-outorder and sim-safe on your computer.

Now, you can begin the following experiments to see the performance impact of varying hardware parameters. Run the simulation and report the observed results (including the printed statistics in your report – you can only include the related ones instead of the full printout). Since running the program from beginning to end using the detailed simulator, sim-outorder, would take too long, you are asked to run only a small portion of the program. Please use the following option “-fastfwd 200000000” to fast-forward the first 200 million instructions and collect statistics on the next 300 million instructions (using the option “-max:inst 300000000”).

1. What is the performance of running the program, equake, under the default system setup (without changing any simulation parameters) using command: `./sim-outorder -fastfwd 200000000 -max:inst 300000000 equake.ss < equake.in`?
2. How much is the performance loss if the processor uses in-order execution instead of the default out-of-order execution for running the program?
3. The above experiments only perform detailed simulation on 300 million instructions. Based on the simulator running time in Question 1, estimate how long it would take to run the program on a real machine with 3GHz processor and the same IPC value as in Question 1; and then estimate how long it would take to simulate the program’s execution in details from beginning to end using the default configuration. Note: Do not run the detailed simulation from beginning to end. It may take days to finish.

4. An advantage of using simulator is that you can vary the processor parameters to see their performance bottleneck. A widely used approach is to partition the program's execution time or CPI value into three categories: CPU execution (including L1 cache access), L2 cache access and memory access. To achieve this goal, we can run experiments under three configurations: (1) the default configuration, (2) a configuration with the memory latency equal to the L2 cache latency, which basically assumes an infinitely large L2 cache, and (3) a configuration with both the memory and L2 cache latency equal to the L1 cache latency, which basically assumes an infinitely large L1 cache. The performance under these three configurations can be used to partition a program's execution into three categories mentioned above. Report your results and discuss which portion is the performance bottleneck for this program running on the simulated machine.

Note:

- Please attach the printed results or screenshots to your report.

Hint:

- You need to use "sim-outorder" in order to get any performance-related statistics such as the CPI or IPC value.
- There is no program called sim-inorder for in-order execution. But sim-outorder has an option to perform in-order execution.
- The reported "sim\_elapsed\_time" is how fast the simulator runs on your machine instead of how fast the program runs on the simulated processor.
- For question 3, you can use "sim-safe" to run the program till the end to get the total number of instructions of the program (it may take a few hours depending on the speed of your machine) and then do the estimation.