

# Project #1

## CDA-4205 Computer Architecture

### Due June 28, 2017

#### Notes:

1. All labs should be done and submitted individually
2. Show all steps to get full points
3. Submit report electronically in canvas

#### Introduction to ISA using SimpleScalar

In this exercise, you will run some Benchmarks to find out the distribution of instructions they execute. You will use **sim-profile** simulator available in the SimpleScalar toolset for this purpose. In the first part of this project, you will use SimpleScalar configured for ALPHA ISA while in the second part, you will use both ALPHA and PISA configuration. Note that PISA is similar to MIPS ISA whereas ALPHA is another type of RISC ISA.

[Note: Results of simulation may vary over multiple runs and among students. This fact is stated at [www.simplescalar.com](http://www.simplescalar.com) . You should see FAQ section at this website. But the results will be consistent which means they will make sense.]

The SimpleScalar tool set is a system software infrastructure used to build modeling applications for program performance analysis, detailed microarchitectural modeling, and hardware-software coverification. Using the SimpleScalar tools, users can build modeling applications that simulate real programs running on a range of modern processors and systems. The tool set includes sample simulators ranging from a fast functional simulator to a detailed, dynamically scheduled processor model that supports non-blocking caches, speculative execution, and state-of-the-art branch prediction.

SimpleScalar simulators can emulate the Alpha, PISA, ARM, and x86 instruction sets. SimpleScalar was created by Todd Austin. Development began while he was a Ph.D. student at the University of Wisconsin in Madison. Early versions of the tool set included contributions by Doug Burger and Guri Sohi. Today, SimpleScalar is developed and supported by SimpleScalar LLC.

#### Prerequisites for the installation

1. Create an account on using usf netid/password on the research computing at USF (CIRCE) server using the following link: <https://www.rc.usf.edu/signup/account.php>
2. Download the SimpleScalar Simulator (simplesim-3v0e.tgz) using the following link: <http://www.simplescalar.com/>
3. Create a directory named SimpleScalar under your home folder on CIRCE server and place/copy the simplesim-3v0e.tgz file.

(Note: You can either remote login to CIRCE server using any remote login client or use ssh to login using terminal(Unix/linux/Mac), putty(windows))

#### Installation Instructions:

1. Using the terminal go to the SimpleScalar directory on the CIRCE server.
2. Make sure the simplesim-3v0e.tar file exists in the folder.
3. Unzip the file using following command  
**tar -xvf simplesim-3v0e.tgz**  
This will create a folder named simplesim-3.0 in the current directory.
4. Enter the simplesim-3.0 directory and execute the following three commands to configure the SimpleScalar simulator as ALPHA simulator.

**make config-alpha**

**make**

5. To configure SimpleScalar as PISA Simulator, execute the following three commands while you are in the simplesim-3.0 directory

**make clean**

**make config-pisa**

**make**

To switch back to the ALPHA configuration, execute the following three commands

**make clean**

**make config-alpha**

**make**

Now that you have installed and configured the SimpleScalar simulator, you are ready to work with it.

### Problem-1

Quick fact about sim-profile simulator: It is a dynamic instruction profiler which means it collects information about instructions as they are executed by the simulator. Note that it is a functional simulator and not a detailed timing simulator as sim-outorder. All the simulators including sim-profile are available in the home/simplesim-3.0 directory.

Go to home/simplesim-3.0 directory and type the following to seek help about sim-profile.

**home/simplesim-3.0\$ ./sim-profile -h**

Help can also be invoked just by typing simulator name without any arguments.

Go through help. Now use this help information to execute the following four benchmarks available in this tar file. Download the benchmarks from canvas. Unzip and Untar this file in a directory e.g. /home/benchmark directory.

The information on how to execute these benchmarks using sim-safe is available in the README file which is also in the home/benchmark directory. Replace sim-safe with sim-profile and add information using sim-profile help to fill out the following table (**hint: use the -iclass flag**)

- 1) anagram: a program for finding anagrams for a phrase, based on a dictionary.
- 2) compress: (SPEC) compresses and decompresses a file in memory.
- 3) go: Artificial intelligence; plays the game of go against itself.
- 4) gcc: (SPEC) limited version of gcc.

#### (a) Fill following table (20 points)

| Benchmark      | Total # of instructions | Load % | Store % | Uncond branch % | Cond branch % | Integer Computation % | Floating pt Computation % |
|----------------|-------------------------|--------|---------|-----------------|---------------|-----------------------|---------------------------|
| anagram.alpha  |                         |        |         |                 |               |                       |                           |
| go.alpha       |                         |        |         |                 |               |                       |                           |
| compress.alpha |                         |        |         |                 |               |                       |                           |
| gcc.alpha      |                         |        |         |                 |               |                       |                           |

Now answer the following questions for each individual benchmark executed above (10x3=30 points)

- b) Is the benchmark memory intensive or computation intensive?
- c) Is the benchmark mainly using integer or floating point computations?
- d) What % of the instructions executed are conditional branches? Given this %, how many instructions on average does the processor execute between each pair of conditional branch instructions (do not include the conditional branch instructions)

## Problem-2

In this part, you will compare the PISA and Alpha ISA by executing the same benchmarks on the two configurations.

Since the configuration from part 1 is still Alpha, execute the following benchmarks available in the home/simplesim-3.0/tests-alpha/bin/ directory using sim-profile and record the results in the following table.

- 1) test-math: performs various math computations mostly in integer and displays their result.
- 2) test-fmath: performs various math functions mostly in integer.
- 3) test-llong: performs computations in long format.
- 4) test-printf: displays various print statements.

(a) Fill following table (20 points)

| ALPHA Benchmark | Total # of instructions | Load % | Store % | Uncond branch % | Cond branch % | Integer Computation % | Floating pt Computation % |
|-----------------|-------------------------|--------|---------|-----------------|---------------|-----------------------|---------------------------|
| test-math       |                         |        |         |                 |               |                       |                           |
| test-fmath      |                         |        |         |                 |               |                       |                           |
| test-llong      |                         |        |         |                 |               |                       |                           |
| test-printf     |                         |        |         |                 |               |                       |                           |

Now change configuration to PISA by typing the following three commands in a sequence.

```
home/simplesim-3.0$./make clean
home/simplesim-3.0$./make config-pisa
home/simplesim-3.0$./make
```

To change configuration back to Alpha, execute the same command sequence except that in the second command replace config-pisa by config-alpha.

(b) Now execute the following benchmarks available in the home/simplesim-3.0/tests-pisa/bin.little/ directory using sim-profile and record the results in the following table. (20 points)

| PISA Benchmark | Total # of instructions | Load % | Store % | Uncond branch % | Cond branch % | Integer Computation % | Floating pt Computation % |
|----------------|-------------------------|--------|---------|-----------------|---------------|-----------------------|---------------------------|
| test-math      |                         |        |         |                 |               |                       |                           |
| test-fmath     |                         |        |         |                 |               |                       |                           |
| test-llong     |                         |        |         |                 |               |                       |                           |
| test-printf    |                         |        |         |                 |               |                       |                           |

**(c) Now compare the two ISAs using a plot (a Histogram is preferred). Use MATLAB or EXCEL to plot the histogram. What can you conclude about the two ISAs from the Histogram. (10 points)**