

## Lab 3: Pipelining and Superscalar using SimpleScalar

**Due by the end of Thursday, Apr. 11, 2024**

In this lab, you will implement pipelined and superscalar configurations on **PISA ISA** and compare the two using the SimpleScalar tool set. You can do the same for **Alpha ISA**. You will also see if forwarding is being implemented or not.

You will use **sim-outorder** as the simulator to calculate IPC or CPI in various processor configurations. It is a detailed microarchitectural simulator and models timing. This tool models in detail in-order and out-of-order microprocessor with all of the bells and whistles, including branch prediction, caches, and external memory. This simulator is highly parameterized and can emulate machines of varying numbers of execution units. Refer to reference #1 to get a visual description of this simulator. You should read the references to know the 5 stages in **PISA** architecture.

Go to **u/xyz/simplesim-3.0** directory and type the following to seek help about **sim-outorder**.

```
./sim-outorder -h
```

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

1) In this part, you will see whether **sim-outorder** implements forwarding to reduce stalls due to data hazards or not.

Follow the procedure described below.

Create new directory e.g., **lab3** in the **/u/xyz/simplesim-3.0** directory. This is where you keep all the files for this lab.

b) Copy **default.cfg** configuration file from **/u/xyz/simplesim-3.0/config/** into the **lab3** directory.

c) Make a copy of the file **default.cfg** and name it e.g., **config\_a.cfg**.

d) Copy **test-math** binary file from **/u/xyz/simplesim-3.0/tests-pisa/bin.little/** into the **lab3** directory.

e) View **config\_a.cfg** file in an editor or text viewer. This file is used by **sim-outorder** simulator. The file is easy to understand. To dig into the details, refer to reference #2.

f) Modify **config\_a.cfg** file according to the following specifications.

```
Fetch queue size = 1,  
Load store queue size = 8,  
Register update unit (ruu) size = 8,    //This is the same as the number  
                                         //of reservation stations  
Issue width = 1,  
Memory ports = 1,  
Inorder = true,  
Resources all types = 1.
```

```
lab3/ alias soo="/u/xyz/simplesim-3.0/sim-outorder"

lab3/ alias pv="/u/xyz/simplescalar/simplesim-3.0/pipeview.pl"
```

```
lab3/  soo -config config_a.cfg -ptrace config_a.trc 0:1024 -redir:sim
sim_configa.out ./test-math
```

j) View the simulation results by viewing the `sim_configa.out` file.

Q1: Describe the configuration.

Q2: What is the IPC (Instructions per cycle) of the `test-math` program using this configuration?

Q3: Is the forwarding implemented? To answer this question, consider the following two instructions

View the pipeline trace `config_a.trc` by executing the following command and see if you can find the two instructions such that `ag` is in the WB stage and `ah` is in the EX stage.

lab3/ pv config\_a.trc | less

2) In this part, you will change the configuration. Every thing else remains the same as in part 1)

a) For this, make a copy of `default.cfg`, call it `config_b.cfg`. Make changes as in part 1) except that

Inorder = false

b) Run `sim-outorder` with the `config_b.cfg` file.

Now Answer the following questions:

Q4: Describe the configuration.

Q5: What is the IPC (Instructions per cycle) of the `test-math` program using this configuration?

Q6: Compared to part 1) or configuration A, what can you say about this configuration?

Q7: The commit stage CT reorders the execution of instructions, explain why is it necessary?

References

[www.cs.wisc.edu/~markhill/cs752/Fall2004/includes/ss\\_tutorial\\_fall2004.ppt](http://www.cs.wisc.edu/~markhill/cs752/Fall2004/includes/ss_tutorial_fall2004.ppt)  
<http://www.eecs.umich.edu/~taustin/papers/UWTR97-simple.pdf>