University of Southern California
Department of Electrical Engineering
EE557 Fall 2022
Instructor: John Paul Walters
Project #3, Due: 11:59 PM., Tuesday, November 3rd
TOTAL SCORE: / 10

I. Extending Project 2

Project 3 builds on your experience gained in Project 2 configuring architectural simulators. In this project your goal is to redesign the baseline processor by changing several micro-architectural blocks, such as branch predictors, Register update units etc., to improve the performance of the baseline processor. In this project you will iteratively look for an optimal design choice for all the micro-architectural blocks by exploring the design space using simulations. Again, this task can be accomplished without any need to modify the code and instead by simply (and intelligently) changing the simulation parameters in the configuration file as you have already done in Project 2.

Unless otherwise stated, every detail in Project 3 stays the same as in Project 2. In particular, the simulator and the benchmark locations, baseline configuration, and all other project environments are identical to Project 2.

II. Project Description

In this project you are given a **MAXIMUM transistor and area budget**. Your goal is to change any combination of the following micro-architectural blocks below to achieve the overall best performance for all four benchmark programs. We will measure the overall performance as:

$$\frac{\sum_{i=1}^{4} \text{\# of committed instrucitons}_{\text{benchmark(i)}}}{\sum_{i=1}^{4} (\text{\# of cycles} \times \text{clock cycle period})_{\text{benchmark(i)}}} \text{(in MIPS)}$$

The four benchmark programs are **perl**, **compress95** (from Project 2), **anagram** and **cc1** as below in the project environments. For instance, if there are 1 million instructions committed per each benchmark; the simulation cycles of the four benchmarks are 1, 2, 3, 4 million cycles; and the clock cycle time is 1 ns, then performance is computed as follows:

$$\frac{(1+1+1+1)\text{million instructions}}{(1+2+3+4)\text{million cycles} \times 1\text{ns}} = \frac{4\text{million instructions}}{10 \times 10^{-3}\text{seconds}} = 400\text{MIPS}$$

The transistor count including every component and the area budget are given below. Your design is NOT allowed to exceed either of them. This budget will be measured by the Real Estate Estimator tool. **Designs over the budget will get 0 point.**

Transistor count: 200 million

Area: 20 mm2

You are allowed to change only the following micro-architectural blocks: (For instance, you can increase or decrease the sizes of the components, change the cache associativities, change the cache replacement policies.)

Dynamic Branch Predictor[1]
Branch Target Buffer
Size of Return Address Stack
Machine Width (issue/decode/commit per cycle)
Instruction Fetch Queue Size
Register Update Unit Size[2,3] **(must be equal or larger than 32-entry)**
Load/Store Queue Size
Number of Integer ALUs and Multiplier/Divider Units
Number of Floating-point ALUs and Multiplier/Divider Units
Number of Memory Ports
Caches (Size, Associativity, Replacement Algorithm, Block Size) [2,4]

[1] **The perfect branch predictor is not allowed.**

[2] **Remember when you change your RUU or cache structures, the number of read and write ports will be affected. So, each time you change one of those, you need to check the estimator tool for any change in number of ports, and then use CACTI to compute access time and latencies.**

[3] **The RUU size must equal or larger than 32-entry. Any number under 32 is NOT allowed**

Basic Project Steps

Here are new steps for doing this project:

First, repeat the steps 1-6 of Project 2.

1. In this step you will look at the result files generated from the SimpleScalar simulation tool and decide which one of the allowed micro-architectural blocks you want to change. Keep in mind that you cannot exceed the area and the transistor count limits specified above when you increase the structure sizes. Also, make sure that the clock cycle latency is appropriately adjusted to reflect the new structure sizes. So be clever about which structure to change and by how much. Since the SimpleScalar result file contains various block access counts, cache misses, hits etc. there is no need to change the code.
2. Once you change one or more micro-architectural parameters you redo steps 1 through 6 of Project #2, as necessary. Look at the new MIPS rating of the processor with your enhanced processor configuration. Compare it with all prior configurations. Iterate the steps till you think you have the world's best processor.
3. Finally, you will generate a report that shows how you iterated through the design space and why you made those design choices. Support your arguments with charts and compelling arguments.

III. Project Environment

Project environment is the same as that of Project 2 except that you need to copy the additional inputs in **p3files.zip** on DEN. The executables and commands to run the simulation are shown in the below table

Executable	Command
./test.bin.little/anagram	sim-outorder -config test.config -redir:sim anagram.out test.bin.little/anagram words <anagram.in
./spec95-little/cc1.ss	sim-outorder -config test.config -redir:sim cc1.out spec95-little/cc1.ss -O 1stmt.i
./spec95-little/perl.ss	sim-outorder -config test.config -redir:sim perl.out spec95-little/perl.ss perl-tests.pl
./spec95-little/compress95.ss	sim-outorder -config test.config -redir:sim compress.out spec95-little/compress95.ss <compress95.in

For all benchmarks, we will limit all our simulations to less than **50 million instructions.** We will fast-forward through first **5 million instructions**. For all benchmarks, set the **-max:isnt** and **-fastfwd** to be **50000000** and **5000000**, respectively.

IV. Project Submission

You must submit the following files in one .tar file named as LastName_FirstName_Proj3.tar by the due date to DEN. – Penalties will be applied for missing files:

1) Your sim-outorder configuration file that you believe to have the best overall MIPS for the four benchmarks. (**LastName_FirstName_Proj3.conf**)
2) The excel sheet of the Real estimator tool of your best design. (**LastName_FirstName _Proj3.xls**)
3) Your RUU Cacti input (**LastName_FirstName_RUU.cfg)**
4) **All** cache Cacti inputs (**LastName_FirstName_L1I.cfg**, **LastName_FirstName_L1D.cfg**, **LastName_FirstName_L2U.cfg**)
5) An electronic copy of your project report (**LastName_FirstName_Proj3.pdf**) that includes the followings:

1. Front page

   a) Title; b) Name: <your name>; c) your email address.

2. Section 1. Design Process

   Description and discussion of your design process – your iteration process: for example, what design progress and iteration you made to approach your final design, based on what results you observed and how that observation affect your next step of design iteration - at least a half page

3. Section 2. Intermediate Results – including a), b) and c) below for each of the 3 intermediate iterations.

a) Intermediate average MIPS rates in a graph; b) transistor count and area estimates from Real Estate Estimator in two graphs; c) cache miss rates for all caches in a table

4. Section 3. Final Design
   a) MIPS rate; b) cycle time; c) area from Real Estate Estimator; d) transistor count from Real Estate Estimator; e) cache latencies; f) cache miss rates – in a table

Please keep all of your shell scripts and simplescalar config files as they might be required to be submitted or asked to run by the TA.

V. Grading

Your final design will be evaluated based on the following criteria:

1. All necessary files as mentioned in section IV. – 2pt
2. The pdf report as mentioned in section IV. – 4pts
3. Performance – 4pts
   This part is evaluated by ranking the overall MIPS of all students.

0 point will be given for a mismatch between a reported overall MIPS and an overall MIPS from running sim-outorder with the config file submitted.

0 point will be given to designs over the transistor count and the area budget.

Like other assignments, this project must be done INDIVIDUALLY!
Similar designs will be securitized.