CprE 381 – Computer Organization and Assembly Level Programming

Project Part 3

[Note: This is the final and summative component for your term project. The purpose of this assignment is to put the other components of your project into context and allow you to relate the detailed implementations you have done to the higher-level concepts we have interacted with in other aspects of the course. As such, ***this report is expected to be of higher quality than your previous reports, in particular with regard to its clarity, analysis, and readability****. Please make an effort to provide context for all figures and some flow through the paper (i.e., don't just copy the report template and respond directly). You have three working processor designs: Congratulations for getting to this step!*

Disclaimer: Due to this project being due during ~~Dead~~Prep Week, there will be no extensions granted. Please turn in whatever work you have completed by the due date.]

0. **Prelab.** Review your notes, evaluations, and feedback regarding your single-cycle, software-scheduled pipeline, and hardware-scheduled pipeline. Make sure you have the three designs ready to evaluate. In particular, ensure that you have your synthesis results handy.

1. **Introduction.** Write a one paragraph summary/introduction of your term project.

2. **Benchmarking.** Now we are going to compare the performance of your three processor designs in terms of execution time. Please generate a table for each of your final single-cycle, software-scheduled pipeline, and hardware-schedule pipeline designs. The rows should correspond to your synthetic benchmark (i.e., the one with all instructions), grendel (provided with the testing framework), Bubblesort, and, for teams >4, Mergesort. The columns should be # instructions (count using MARS), total cycles to execute (count using your Modelsim simulations), CPI (using the previous two columns to calculate), maximum cycle time (from your synthesis results), and total execution time (using the appropriate previous columns). Note that the applications used to benchmark the single-cycle and hardware-scheduled pipeline applications should be identical and thus the same number of instructions, while the software-scheduled pipeline programs should be modified to work on the software-scheduled processor and thus should have more instructions. Count software-inserted NOPS as instructions. Make sure to include units and double-check that these results make sense from your first principles!

3. **Performance Analysis.** Analyze the performance of the three applications on the three processors. Explain in your own words why the performance was better on one processor versus another or why some applications may have had a smaller difference in performance between processors versus other applications. This section should reference the above performance table and describe how it was generated including any formulas you used. The section should be about five to seven substantial paragraphs.

4. **Software Optimization.** Identify and describe one software optimization (i.e., assembly-level software refactoring) that would improve the performance of software on the software-scheduled pipeline relative to the others. Provide an estimate of the performance benefit this change could have given your specific benchmarks.

5. **Hardware Optimization.** Identify and describe at least one different hardware optimization for *each* design that would improve its performance. The optimization cannot be turning it into one of the other designs. Certain optimizations can be beneficial to more than one design – chose one design on which you would apply the optimization. Briefly list the specific set of changes you would have to make to your design to accommodate each optimization (a figure would be helpful). Provide an estimate of the performance benefit each optimization could have given your specific benchmarks.

6. **It Depends.** Given the above discussion, you should now understand the interaction between the programs and your hardware designs in terms of performance. Identify or write a program that performs better on a single-cycle processor versus a hardware-scheduled pipeline and another one that performs better on the hardware-scheduled pipeline versus the software-scheduled pipeline. Describe your approach to building these programs. If one of these cases is impossible given your designs, argue *quantitatively* why that is the case.

7. **Challenges.** This term project was challenging for every group. In at least three detailed paragraphs, describe the three most critical challenges your group faced, how you resolved them, and how you could avoid them in the future.