ECE 437L Final Report

Brian Rieder (mg267)

Pooja Kale (mg273)

TA: Nick Pfister

9 December 2016

1 Executive Summary

Text for your executive summary

2 Processor Design

Text for your multicore design

Text for your pipeline design

Text for your cache design

3 Results

	Pipeline w/o Cache	**Pipeline w/ Cache**	**Multicore**
Estimated Synthesis Frequency	45.09 MHz	38.01 MHz	20.37 MHz
Instructions per Cycle (IPC)	0.078	0.230	
Single Instruction Latency			
Total Utilized Combinational Functions	3,274	6,446	13,800
Utilized Dedicated Logic Registers	1,720	4,175	8,208
Speedup from Sequential to Parallel	1.000		

Table 1: Synthesis Results

Text for your results

4 Conclusion

Text for your conclusion

5 Contributions and Collaboration Strategies

The entirety of the processor design in all of its stages was done in tandem between Pooja and Brian. Early on in the process, the design was formulated together with all principles of the design being taken into mutual consideration - all design decisions (for instance, the design of the state machines for DCACHE and the coherence controller) were made together before any development began.

5.1 Individual Contributions

During development, tasks were partitioned between Pooja and Brian in order to ease the development process. As such, there were two primary initiatives in this phase of development, divided up as follows:

5.1.1 Cache Design

For cache development, there were two primary products: the instruction cache and the data cache. Both were constructed as a team with the following contributions:

- **Instruction Cache** - The instruction cache was developed by Brian and a testbench was written and used to ensure functionality by Pooja. The instruction cache, in an entirely anomalous case compared to the rest of the project, required no debugging in order to work properly.

- **Data Cache** - The data cache state machine was developed together in order to ensure understanding between the two teammates. After the cache design was discussed, Brian began to add it to the data cache itself and Pooja constructed the testbench to ensure that the data cache itself would work. At this point, the cache appeared to work properly, but when it was inserted into the greater system new bugs began to appear. The diagnosis process at this point was done together and the changes were discussed and added until the design itself was complete.

5.1.2 Multicore Design

Placeholder for individual contributions

5.2 Collaboration Strategies

Account information was shared between team members at the beginning of the collaboration stages and SSH keys were set up between the two accounts for ease of user-switching, but each team member did their development within their respective accounts and contributed to GitHub using their own account. Because of this Git usage and presence on GitHub, all contributions are documented in a quantitative manner within the repository.

5.2.1 Git VCS Usage

Just as in the midterm, in order to compartmentalize development of features and to keep both contributors on the same page, an industry strategy of progressive-stability branching was employed and all development was done within respective branches and "work silos" - a master branch was maintained with the most recent functioning design, "singlecycle", "pipeline", "caches", and "multicore" branches were maintained, and each feature were developed on an independent branch for that feature (some examples being "branchpredict" and "forwarding" in the earlier stages of development). While this required more overhead in terms of planning and maintenance, the preservation of states and releases within the development process enabled ease of rollbacks, identification of changes, and the ability to develop without worrying about destruction of a prior working state.