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 without Cache	**Pipeline with Cache**	**Multicore**
Estimated Synthesis Frequency	??		
Instructions per Cycle (IPC)	??ns		
Single Instruction Latency			
Total Utilized Combinational Functions			
Utilized Dedicated Logic Registers			
Speedup from Sequential to Parallel			

Table 1: Synthesis Results

Text for your results

4 Conclusion

Text for your conclusion

5 Contributions and Collaboration Strategies

Placeholder for contribution section intro

5.1 Individual Contributions

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.