*Jerry Kurtin Lab 6 Individual Lab Report*

*Team Members:*[Ariela Mitrani](mailto:amitrani@tamu.edu)*,* [Garrett Guerrero](mailto:gguerrero@tamu.edu)*,* [Jade Davis](mailto:jadealexie12@tamu.edu)

*Individual Contributions*

I was responsible for building the execute stage in the processor, I contributed to combining each stage, I wrote the assembly test cases, and I co-wrote the extra credit matrix multiplication script. The contribution I am most proud of is designing our system to communicate between stages. I brainstormed a simple method, sending a 1-clock pulse when the section is finished, and waiting dormant until receiving a similar pulse. This method proved to be successful when integrating the entire CPU. I could’ve simplified my logic in the execute stage, for I added some unnecessary computation that cluttered the project.

*Pros and Cons of Team-Based Assignment*

This project was very well-suited to a team-based approach, for there was a clear division of roles, yet each subsection of work required crucial communication between members. This was a powerful learning experience, and our team worked very well together. My team found that it was difficult to gauge the difficulty of each section of work, but that can be expected from any large project, and it only caused minor difficulties.

*Teammate’s Superior Design and Potential Improvements*

Garrett’s register file was very well-designed, and it integrated seamlessly into the rest of the project. He creatively supported 2 simultaneous read/write instructions, preventing us from worrying about timing delays. Garrett could’ve named the i/o to his blocks more clearly. Jade created simple and powerful combinational logic to guarantee that the fetch stage sent its continue pulse as soon as its data was ready, removing a lot of extraneous clock cycles. There was a small error in naming the i/o of a sub-circuit that would’ve removed confusion when integrating the circuits. Ariela created a simple, yet genius delay block that was utilized across the project to guarantee that data was ready for downstream computations. Ariela’s memory section could’ve been sped up by continuing immediately when not writing to memory.