**Part 5 – Hardware Implementation and Working Processor Demo**

Feedback:

* What did you learn from this project?
  + Choosing a pipelined architecture taught us a great deal about both the concept (and implementation) of pipelining and dealing with the data, control, and structural hazards that can be introduced when designing your architecture. We also learned the value of asynchronous components in regard to dealing with potential timing issues presented by hardware. Finally, we gained experience dealing with the realities of hardware (and how they can conflict with your initial, idealized designs), especially with regard to timing and clock-to-output delays.
* What would you do differently next time?
  + There really isn’t anything major that we would do differently; we made steady progress on our project and didn’t encounter any drastic setbacks. There are a few minor design decisions (such as the formatting of machine code) that we would make differently during the early stages of design (as we now have a greater perspective on how those decisions affect the design of hardware elements).
* What is your advice to someone who is going to work on a similar project?
  + The first piece of advice we can offer is to read all of the responses given by previous semesters: you can often glean important information (especially about the specifics of the hardware being used). Additionally, if any particular approach was plagued by hardware limitations or other issues, you can be assured that someone will have something to say about it.
  + Don’t be afraid to make changes to your design; you can often save yourself a great deal of time and effort by making simple changes. For example, we found that re-arranging our machine code to always store the RP register address first (for those instructions that included it) streamlined the implementation of our datapath.
  + Though something of a cliché, the advice to start early is still relevant. This project represents a fairly major time commitment even if you design everything perfectly the first time (and believe us, you probably won’t).
  + Keeping your documentation up to date and orderly is critical. Good documentation aids not only in visualizing while implementing your design, but also aids in the debugging process. Having a quick way to look up your assembly directives and their corresponding machine code is enormously useful.
  + Pipelining, though initially daunting in appearance, is quite worth the effort to implement. Aside from the obvious (and significant) performance gains, implementing a pipelined architecture allows you to really apply the lessons learned from the class.