**Project Part 5**

**Xin Jin**

**Pratibha Kona**

**What did you learn from this project?**

This project built up a connection between textbook and practical experience in computer architecture engineering. We learnt how instruction set, datapath and control unit works in the classroom. This project provided a great opportunity to acquire an in-depth understanding of that knowledge. We designed a pipeline CPU in this project and we believe the greatest difference between pipeline and single-cycle/multi-cycle is that data hazard has to be considered into pipeline architecture. The difficulty in implementing data hazard unit is the clock assignment in the architecture need to be carefully considered. Besides those, we also had a better understanding of programming tools, such as Modelsim and Quartus and Altera FPGA.

**What would you do differently next time?**

We designed a good datapath in the beginning, so we did not need to spend a lot time in redefining datapath later. The problem we came across is to implement it in hardware, we used Verilog in this project and we could not include the run time editable memory in our final design, there was a module in the memory generated by Quartus, which was instantiated in the memory but was not defined. Instead we used register file kind of memory to replace the Quartus generated memory. We also had problems in executing the program in FPGA. SO starting the implementation in hardware in earlier stages of the design and using the software to get rid of the problems encountered in hardware is the solution. Next time, we would check if there are some aspects that we need to pay attention, in using FPGA.

**What is your advice to someone who is going to work on a similar project?**

A good datapath is a needed to save the trouble you might run into, in later design stages. So, all the data transition and control signals need to be considered thoroughly in earlier stages. A program that can simulate does not mean it is synthesizable, so always pay attention to the syntax you use and check it whether it is synthesizable in hardware. Data hazard detection will complicate the design and need careful work in clock assignment. We used negedge clock for all registers and combinational logic in hazard detection unit and it works fine.