Computer Architecture Project Part-5

1. What did you learn from this project?

From this project, we learnt how to design a RISC computer from the instruction set till the FPGA implementation of the system. We had a good experience in VHDL programming and using Modelsim and Quartus tools and programming in Altera FPGA boards.

1. What would you do differently next time?

We have moved the branch and jump data path components to the second stage and included a comparator to decide the branch conditions. This will decrease the no-op instructions to be used after the branch instruction to eliminate hazards by reducing the number of cycles taken to execute a branch or a jump to 2 cycles. We did not include the hazard detection unit because we did not want to complicate the programming part. We concentrated on making the system work rather than the performance. If we had more time we would have included the hazard detection unit in our pipeline.

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

Firstly, be very careful with the multiplexer controls and the inputs. You might swap the two inputs for the multiplexer controlling the destination registers or the input to the ALU. Secondly, make sure the register file read is out of the write process. That will allow read and write operation to be executed simultaneously. You should be able to do that because in a pipeline, the destination register is written in the 5th cycle and at the same time some other instruction will be decoded. Also, the register write control should come from 5th stage as it is the control derived from the previous instruction and not the current instruction being decoded in the 2nd stage.

At begging we found we were storing 0000 into data memory for sw instruction all the time. Then we realized that it may be a timing issue because we didn’t pass the data through pipeline register between stage 3 and 4 but directly sending data to data memory. Later we pass the data through the pipeline register into the data memory, this problem is fixed. Now we can correctly store data into desired memory location.