**CPU Design Project – Part 5 – Final Report**

**Jie Wen**

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

**In this project, I designed a multi-cycles MIPS CPU using VHDL and tested it in the FPGA board. From this project, I learned how the MIPS CPU works with this kind of datapath and the state machine controller. In my implementation, I use three cycles for the multi-cycles operation. But theoretically all the instructions can be implemented in two cycles. For example, if I use falling edge to access memory and at the same time eliminate most of the registers which is redundant in the datapath, in two cycles the instruction can be read into datapath and the data can be operated accordingly. In the first cycle (from rising edge of the first cycle to the rising edge of the second cycle), control unit sends the control signals related to fetch to the datapath and the memory. By the end of the first cycle the instruction will be on data bus and available to the datapath. After the rising edge of the second cycle, the instruction will be read into instruction register. At the beginning of the second cycle, the instruction in IR will be decoded and send all the control signals related to this instruction into the datapath. For the case of load data, the control signals related to read memory data and write into register file will be sent in the second cycle. The memory data will be available to the datapath in the middle of the second cycle and this data can be read into register file by the end of the second cycle. There is no conflict for this because the data is transferred sequentially in the datapath. When the data is available at the input of data path, the register file will be ready to write it into half cycle later. This mechanism can also be applied to the other instructions such as R type, store, branch and jump.**

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

**Next time, I want to separate the datapath into several stages and insert corresponding register at the end of each stage to implement the pipeline type of MIPS CPU. The control unit should also be rewritten accordingly.**

**And then, I want to design a multi-function CPU using micro-programming in control unit.**

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

**My suggestion to the other ones is that be clear about every details. How the data be transferred in the datapath and how the control unit control the data flow. Avoid hardware conflict in the data operation.**