CO_HW4 Report

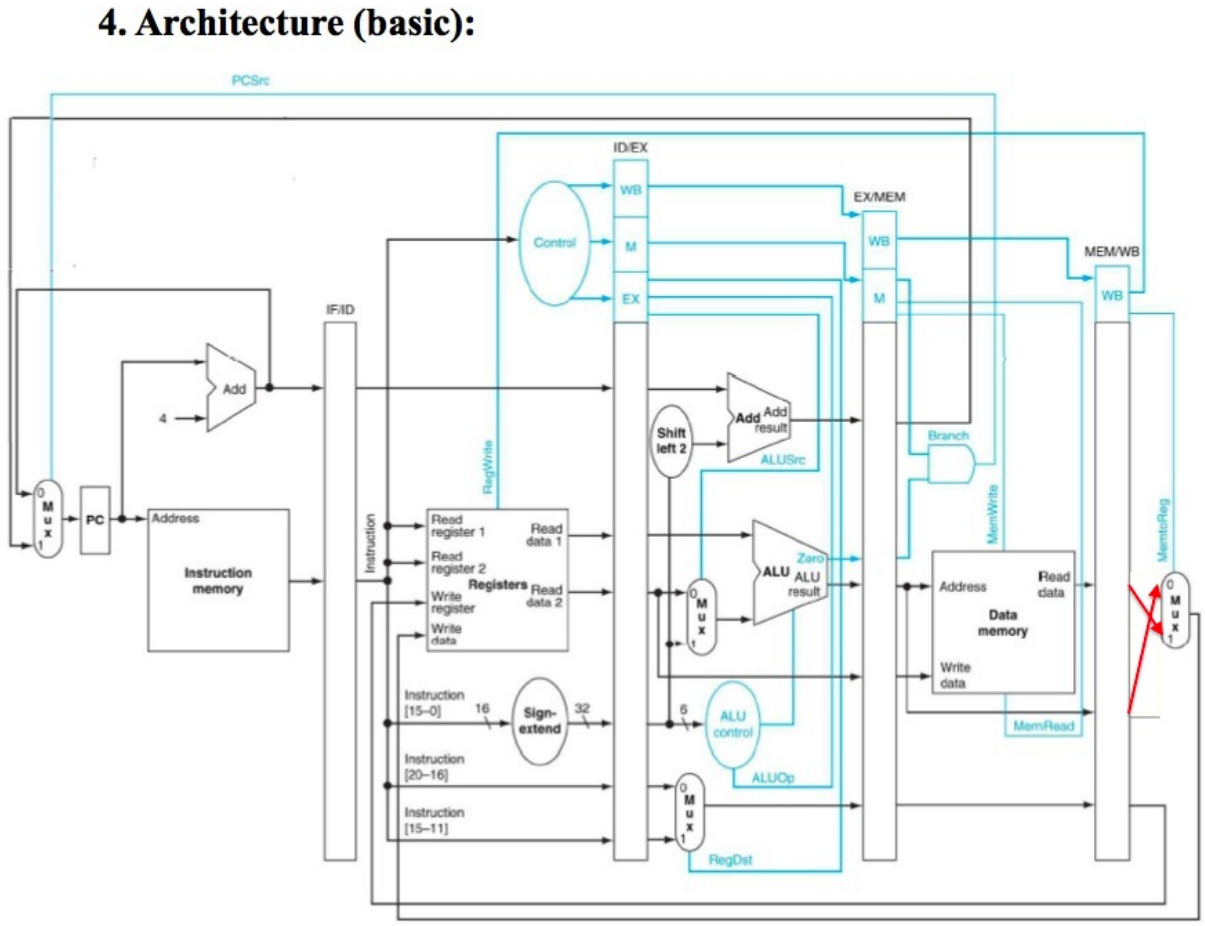
0016043 鄭順一
0016075 陳雅琳

1. Source code and the note

Compressed file handed on e3 and the comments within the Verilog source code.

2. Your architecture

A modification on the MUX of the WB stage based on the (basic part) picture in the pdf file of this homework.



3. Hardware module analysis

1. ALU.v
    a. Do the computation of the value stored in the registers.
2. Decoder.v
    a. Transform binary strings into meaningful instructions.
3. Pipe_Reg.v

    a. Registers be used in pipelined CPU.
4. Sign_Extend.v
    a. Do the signed extension of 16bits signed integer into 32bits signed integer.
5. ALU_Ctrl.v
    a. Generate the ALU command.
6. Instruction_Memory.v
    a. Initialize instruction memories.
7. ProgramCounter.v
    a. Just a Program Counter.
8. TestBench.v
    a. Load test data
    b. Display the result of loaded test data.
9. Adder.v
    a. Add two 32bits number.
10. MUX_2to1.v
    a. 2 to 1 multiplexer.
11. Reg_File.v
    a. 32 Registers stored value.
12. Data_Memory.v
    a. 32 memories stored value.
13. Pipe_CPU_1.v
    a. The structure of the main pipelined CPU.
14. Shift_Left_Two_32.v
    a. Shift a 32 bits number to left with 2 bits.

4. Finished part

We finished the basic structure of this homework.

5. Problems you met and solutions

1. After adding pipeline registers, there are so many lines and names of them.
    a. Printed out the picture of the basic structure.
    b. Wrote down the name of the line beside it.
2. Checked the code many times, but still got the wrong result of the basic part test data.
    a. Finally, the problem is on the MUX of the WB stage. The data0_i and the data1_i of it should be exchanged.

6. Division of this work

- 0016043 鄭順一
    - write code
    - code execution
    - code review & debug
    - drew some pics about the structure

    - report
- 0016075 陳雅琳
    - write code
    - code review & debug
    - drew some pics about the structure
    - report

7. Summary

Based on LAB3, this time we work on the Pipe_CPU_1.v file, try to make the program work as pipelined one. While during the execution, we found that we had wrong answer but we couldn't figure out where the problem was. It took us lots of time to debug.

Finally, we found the problem is on the MUX of the WB stage. Two inputs of the MUX should be exchanged. After solving the problem, we can run the basic part well now.

So, for next lab, it will be better for us to double check the architecture and the given files in order to avoid such stitution happening again.