**Computer Organization**

**HDL simulator you used (ModelSim or Xilinx): ModelSim**

**The input fields of each pipeline register:**

IF/ID register input:  
- clk\_i  
- rst\_n  
- next\_pc\_i  
- instruction\_i  
ID/EX register input:  
- clk\_i  
- rst\_n  
- next\_pc\_i  
- instruction\_i  
- data1\_i  
- data2\_i  
- signex\_i  
- zeroex\_i  
- jmp\_addr\_i  
- jump\_i  
- branch\_i  
- memwrite\_i  
- RegDst\_i  
- memread\_i  
- mem\_to\_reg\_i  
- RegWrite\_i  
- ALUSrc\_i  
- branch\_type\_i  
- ALUOp\_i

EX/MEM register input:  
- clk\_i  
- rst\_i  
- write\_reg\_i  
- data2\_i  
- last\_data\_i  
- next\_pc\_i  
- branch\_to\_PCSrc\_i  
- branch\_i  
- branch\_rslt\_i  
- jmp\_address\_i  
- jump\_i  
- memwrite\_i  
- memread\_i  
- mem\_to\_reg\_i  
- regWrite\_i

**Compared with lab4, the extra modules:**- IF\_ID\_register  
- ID\_EX\_register  
- EX\_MEM\_register  
- MEM\_WB\_register

**Problems you met and solutions:**

* Writing a pipeline CPU require 4 additional register to be added. This add to the complexity of the program. One of the many met problems in the writing process is miss to declare a wire or wrong wiring routing.  
  Solution: By designing the register in order and checking the result after every each register is attached can reduced the time to debug a program because we can be sure that the register written before should be correct after the testing.

**Summary:**

Writing or designing a big code is not an easy task. Before starting, make sure to planned out the order an problem that might arises in the process. By preventing the worse outcome to happen, we can reduce the programming time by much. One of the way to achieve this in this assignment is by testing the program every time one new register has been successfully attached. This method can reduce the location of error so that the mistake can be pinpoint easier.

Pipelining a CPU let us design a more efficient CPU with faster clock speed. However, this design have some disadvantage due to its way of calculating that could do much more than one at the same time.