Computer Architecture LAB1 Report

B09901089 黃柏穎

1. Adder.v
   This module just takes 2 inputs and add them together, and is used in IF stage to add PC and constant4 and ID stage to add the PC and immediate from branch instruction.
2. ALU_Control.v
   This module decodes the 10 bits function and aluop signal, generating the AluCtrl signal.
3. ALU.v
   This module takes AluCtrl and two parameters, then apply the arithmetic operation corresponding AluCtrl on the parameters as output
4. AND.v
   This module just output the and operation of two inputs. I implement this as a module is just for visualize.
5. Comparator.v
   This module simply output whether two inputs are identical. I implement this as a module is just for visualize.
6. Control.v
   This module takes the [6:0] bits if instruction, and decode the input to determine ALUSrc_o,RegWrite_o,MemtoReg_o ,MemRead_o,MemWrite_o,Branch_o,ALUOp_o, control signals.
7. CPU.v
   This module takes inputs clk, rst and start; stitches all designed module together and by a lot of wires.
8. EXMEM.v
   This module is one of the pipeline registers, only forwards the value from EXstage to MEM stage at the rising edge of clk.
9. Forwarding.v
   This module checks EXRS1addr, EXRS2addr, MEMRegWrite, MEMRDaddr, WBRegWrite and WBRDaddr to determine whether MEM hazard and EX hazard happened, then generate corresponding Forward signal to the two MUX2 in EX stage.
10. HDU.v
    This module detects control hazard, and sends stall, nop signal and disables the pc update when detected.
11. IDEX.v
    This module is one of the pipeline registers, only forwards the value from ID stage

to EX stage at the rising edge of clk.

12. IFID.v

    This module is one of the pipeline registers, only forwards the value from IF stage to ID stage at the rising edge of clk.

13. MEMWB.v

    This module is one of the pipeline registers, only forwards the value from MEM stage to WB stage at the rising edge of clk.

14. MUX1.v

    This module is a 2-to-1 mux, determines the output by a 1 bit ctrl signal.

15. MUX2.v

    This module is a 3-to-1 mux, determines the output by a 2 bit ctrl signal.

16. Sign_Extend.v

    In this module, we implement the Imm_Gen part in CPU, and because the immediate part differs from instructions, so we first need to extract the immediate parts correctly by examining the function7 and funct3 part, then apply signed extend to 32-bit immediate.