Week 8

1. Write a SV program for Instruction Execute Stage

module instr\_exec #(parameter N=32) (input logic [N-1:0]rs1\_data,rs2\_data,imm,pc,input logic alusrc,branch,input logic alu\_op,output logic [N-1:0]alu\_result,pc\_imm,output logic and\_result);

wire [N-1:0]mux\_result,imm32\_new;

wire zero;

mux\_1 g1(rs2\_data,imm,alusrc,mux\_result);

alu g2(rs1\_data,mux\_result,alu\_op,alu\_result,zero);

and\_gate g3(zero,branch,and\_result);

assign imm32\_new = imm<<1;

fulladder g8(imm32\_new,pc,pc\_imm);

endmodule

1. Write a SV program for the ALU

module alu #(parameter N=32)(input logic [N-1:0]rs1\_data,rs2\_data,input logic alu\_op, output logic [N-1:0]alu\_result,output logic zero);

always\_comb

begin

case(alu\_op)

4'b0000: alu\_result=rs1\_data&rs2\_data;

4'b0001: alu\_result=rs1\_data|rs2\_data;

4'b0010: alu\_result=rs1\_data+rs2\_data;

4'b0110: alu\_result=rs1\_data-rs2\_data;

endcase

if(alu\_result==0)

zero=1;

else

zero=0;

end

endmodule

