i. Write a SV program for Instruction Execute Stage

```
module instr_exec #(parameter N=32) (input logic [N-1:0]rs1_data,rs2_data,imm,input logic
alusrc,branch,input logic alu_op,output logic [N-1:0]alu_result,output logic and_result);

wire [N-1:0]mux_result;

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);

endmodule
```

ii.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
```