```
`timescale 1ns / 1ps

module ALU(A,B,ALU_SEL,OUT,imm,alu_immed,alu_op);

input [31:0] A;                         // main register
input [31:0] B;                         // second register
input [31:0] imm;                       // immediate value;
input alu_immed,alu_op;                 // control signals
input [2:0] ALU_SEL;                    // alu selector
output [31:0] OUT;                      // ALU output

wire [31:0] ALU_reg;
reg [31:0] ALU_output;

assign ALU_reg = alu_immed ? imm : B;           // select between 2nd register or imm
value
assign OUT = alu_op ? ALU_output : 32'hZZZZ;    // tri-state alu_output
////////////////////// ALU OPERATION ///////////////
always @(*)
begin
case(ALU_SEL)
3'b000: ALU_output = A + ALU_reg;
3'b001: ALU_output = A - ALU_reg;
3'b010: ALU_output = A >> ALU_reg;
3'b011: ALU_output = !(A & ALU_reg); /// need to be changed
3'b100: ALU_output = A ^ ALU_reg;
3'b101: ALU_output = A << ALU_reg;
3'b110: ALU_output = (A | ALU_reg);
3'b111: ALU_output = (A & ALU_reg);
default: ALU_output = 32'h0000;
endcase
end
endmodule
```