**Verilog Code**

module alu\_vr(a,b,c,s,clk,d,e);

input [15:0]a,b,c;

input [3:0]s;

input clk;

output [15:0]d;

output [31:0]e;

reg [15:0]d;

reg [31:0]e;

always@(posedge clk)

begin

case(s)

4'b0000: d=a&b; //16-bit and

4'b0001: d=a|b; //16-bit or

4'b0010: d=~(a|b); //16-bit nor

4'b0011: d=(((~a)&b)|((~b)&a)); //16-bit xor

4'b0100: d=~(((~a)&b)|((~b)&a)); //16-bit xnor

4'b0101: d=~(a&b); //16-bit nand

4'b0110: d=~a; //16-bit not

4'b0111: d=a+b; //16-bit add

4'b1000: d=a-b; //16-bit sub

4'b1001: d=a+1; //16-bit increment

4'b1010: d=b-1; //16-bit decrement

4'b1011: d=a; //16-bit input

4'b1100: e=(((~(a&b))&c)|((~c)&(a&b)))\*c;

4'b1101: e=((a&b)\*(a|b));

4'b1110: e=a\*b;

4'b1111: e=((~a)&b)\*(a&(~b));

endcase

end

endmodule