module bcmc(input1,input2,select,clock,reset,out,mux,adn,sub,sl,sr,andd,orr,exxor,exxnor);

input [15:0]input1,input2;

input clock,reset;

input [2:0]select;

output [16:0]out,mux,adn,sub;

output [15:0]sl,sr,andd,orr,exxor,exxnor;

reg [15:0]a1,a2,a3,a4,a5,a6,a7,a0,b1,b2,b3,b4,b5,b6,b7,b0;

reg [16:0]out,mux;

reg [15:0]sl,sr,andd,orr,exxor,exxnor;

reg [16:0]adn,sub;

always@(\*)

begin

case (select)

3'b000 : a0<=input1;

3'b001 : a1<=input1;

3'b010 : a2<=input1;

3'b011 : a3<=input1;

3'b100 : a4<=input1;

3'b101 : a5<=input1;

3'b110 : a6<=input1;

3'b111 : a7<=input1;

default $display("select input is wrong");

endcase

end

always@(\*)

begin

case (select)

3'b000 : b0<=input2;

3'b001 : b1<=input2;

3'b010 : b2<=input2;

3'b011 : b3<=input2;

3'b100 : b4<=input2;

3'b101 : b5<=input2;

3'b110 : b6<=input2;

3'b111 : b7<=input2;

default $display("select input is wrong");

endcase

end

always@(\*)begin

adn=a0+b0;

sub=a1-b1;

sl=a2<<1;

sr=b3>>1;

andd=a4&b4;

orr=a5|b5;

exxor=a6^b6;

exxnor=a7^~b7;

end

always@(\*)

begin

case (select)

3'b000 : mux<=adn;

3'b001 : mux<=sub;

3'b010 : mux<=sl;

3'b011 : mux<=sr;

3'b100 : mux<=andd;

3'b101 : mux<=orr;

3'b110 : mux<=exxor;

3'b111 : mux<=exxnor;

default $display("select input is wrong");

endcase

end

always@(posedge clock)

begin

if (reset) begin

out<=0;

end

else begin

out<=mux;

end

end

endmodule