module multiplier\_fp(

input clk,

input [15:0]a,

input [15:0]b,

output wire [15:0]out

);

reg [4:0]out\_exp;

reg [10:0]out\_man;

wire [4:0]a\_exp;

wire [4:0]b\_exp;

assign a\_exp = a[14:10] ;

assign b\_exp = b[14:10] ;

wire [10:0]a\_man;

wire [10:0]b\_man;

wire [21:0]mult\_man;

reg [31:0]in;

reg [4:0]out\_1;

integer i;

integer flag;

assign a\_man = {1'b1,a[9:0]} ;

assign b\_man = {1'b1,b[9:0]} ;

assign mult\_man = (a\_man\*b\_man);

always@(clk)

begin

if(mult\_man[20] == 1'b1)

begin

out\_man = mult\_man[20:10];

out\_exp = (a\_exp - 5'b01111) + (b\_exp - 5'b01111) + 5'b01111 ;

end

else if(mult\_man[21] == 1'b1)

begin

out\_man = mult\_man[21:11];

out\_exp = (a\_exp - 5'b01111) + (b\_exp - 5'b01111) + (1'b1 + 5'b01111);

end

else

begin

in={mult\_man,10'b0000000000};

out\_1 = 0; // default value if 'in' is all 0's

flag = 0;

for (i=31; i>=0; i=i-1)

begin

if (in[i] && flag == 0)

begin

out\_1 = i;

flag = 1;

end

end

in = in << ( 5'b11111 - out\_1 -1 );

out\_man = in[30:20];

out\_exp = ( a\_exp - 5'b01111 ) + ( b\_exp - 5'b01111 ) - ( 5'b11111 - out\_1 -1 ) + 5'b01111 ;

end

end

assign out = { (a[15]^b[15]),out\_exp,out\_man[9:0] };

endmodule