module sqr(clk,start,round,in\_a,out\_z,nan\_o,inf\_o,zero\_o,ine\_o);  
  
input clk;  
input start;  
input [31:0] in\_a;  
input [1:0] round;  
output [31:0] out\_z;  
output reg nan\_o,inf\_o,zero\_o,ine\_o;  
  
reg [23:0] man\_i;  
reg [23:0] man\_o;  
reg [7:0] exp\_i,exp\_o;  
reg sign\_i,sign\_o;  
reg guard,round\_bit,sticky;  
reg [49:0] pro;  
reg s\_round;  
reg [31:0] z;  
reg [3:0] state;  
  
always @(posedge clk)  
begin  
 if(start)  
 begin

state<=0;  
 nan\_o<=0;  
 inf\_o<=0;  
 zero\_o<=0;  
 ine\_o<=0;  
 end  
   
 case(state)  
 0:begin  
 sign\_i<=in\_a[31];   
 man\_i<={1'b1,in\_a[22:0]} ;  
 exp\_i<=in\_a[30:23]-127;  
 state<=1;  
 end  
  
 1:begin  
 if ((exp\_i == 128 && man\_i != 0) )

begin

z[31] <= 1'b1;

z[30:23] <= 8'b11111111;

z[22] <= 1'b1;

z[21:0] <=1'b0;

state <= 4'bz;

nan\_o<=1;

end   
 else if (exp\_i == 8'b10000000)   
 begin  
 z[31] <= sign\_i;  
 z[30:23] <= 8'b11111111;  
 z[22:0] <= 23'b0;  
 state <= 4'bz;   
 inf\_o<=1;  
 end  
 else if (($signed(exp\_i) == -127) && (man\_i == 0))   
 begin   
 z <= 32'b0;  
 state <= 4'bz;  
 zero\_o<=1;  
 end   
   
 state<=2;   
 end   
  
 2:begin   
 sign\_o = sign\_i^sign\_i ;  
 exp\_o = exp\_i + exp\_i + 1;  
 pro = man\_i \* man\_i \* 4;   
 man\_o = pro[49:26];  
 guard = pro[25];  
 round\_bit = pro[24];  
 sticky = (pro[23:0] != 0);  
 state=3;  
 end  
 3:begin  
 if (man\_o[23] == 0)   
 begin  
 exp\_o <= exp\_o - 1;  
 man\_o <= man\_o << 1;  
 man\_o[0] <= guard;  
 guard <= round\_bit;  
 round\_bit <= 0;  
 end  
 else  
   
 //round to nearest even  
 if(round==0)   
 s\_round<=(guard & round\_bit | sticky);   
 //round up  
 else if(round==1)   
 s\_round<=((guard | round\_bit| sticky) & (~sign\_o));  
 //round down  
 else if(round==2)   
 s\_round<=((guard | round\_bit | sticky) & sign\_o);  
 //round to zero or truncate  
 else   
 s\_round<=0;  
 if(s\_round)   
 man\_o<=man\_o+1;  
 state<=4;  
 end   
 4:begin  
 z[22 : 0] <= man\_o[22:0];  
 z[30 : 23] <= exp\_o[7:0] + 127;  
 z[31] <= sign\_o;  
 if ($signed(exp\_o) > 8'b01111111)   
 begin  
 z[22 : 0] <= 23'b0;  
 z[30 : 23] <= 8'b11111111;  
 z[31] <= sign\_o;  
 end  
 end  
 endcase  
end   
  
 assign out\_z=z;  
endmodule

**TESTBENCH:**

module post\_tb;

reg clk;

reg start;

reg [1:0] round;

reg [31:0] in\_a;

wire [31:0] out\_z;

post mu(clk,start,round,out\_z);

initial

begin

clk=1'b0;

forever #5 clk=~clk;

end

initial

begin

start=1;

round=0;

#10 start=0;

#150 $stop;

end

endmodule