module en\_out(clk,rst,scl\_i\_data\_en,scl\_cfg\_mode,scl\_cfg\_rsz,scl\_o\_data\_en,en\_ff0,en\_ff1,en\_ff2,en\_ff3);

input clk,rst;

input scl\_i\_data\_en,scl\_cfg\_mode,scl\_cfg\_rsz;

output scl\_o\_data\_en;

reg scl\_o\_data\_en;

reg [1:0]q;

output en\_ff0,en\_ff1,en\_ff2,en\_ff3;

reg en\_ff0,en\_ff1,en\_ff2,en\_ff3,en\_ff4,en\_ff5;

always @(posedge clk or negedge rst)

begin

if(!rst)

begin

en\_ff0 <= 0;

en\_ff1 <= 0;

en\_ff2 <= 0;

en\_ff3 <= 0;

en\_ff4 <= 0;

en\_ff5 <= 0;

end

else

begin

en\_ff0 <= scl\_i\_data\_en;

en\_ff1 <= en\_ff0;

en\_ff2 <= en\_ff1;

en\_ff3 <= en\_ff2;

en\_ff4 <= en\_ff3;

en\_ff5 <= en\_ff4;

end

end

always @(posedge clk or negedge rst)

begin

if(!rst)

begin

q <= 2'b0;

end

else if(en\_ff5==1)

begin

q <= q+2'b1;

end

else

begin

q <= 0;

end

end

always @(posedge clk or negedge rst)

begin

if(!rst)

begin

scl\_o\_data\_en <= 0;

end

else if(scl\_cfg\_mode==0)

begin

scl\_o\_data\_en <= en\_ff5;

end

else if(scl\_cfg\_rsz==0 && (q==0 || q==2))

begin

scl\_o\_data\_en <= en\_ff5;

end

else if(scl\_cfg\_rsz==1 && q==0)

begin

scl\_o\_data\_en <= en\_ff5;

end

else begin

scl\_o\_data\_en <= 0;

end

end

endmodule