reg [31:0] counter;

reg [2:0] numb;

reg clkd;

always @(posedge clk) begin

if (reset) begin

counter <= 32'b0;

numb <= 3'b000;

seg <= 8'b11111110;

szero <= 7'b0000111;

clkd<=1'b0;

end

else

counter <= counter +1'b1;

if (counter >= 50000000) begin

clkd <= !clkd;

counter <= 32'b0;

// numb <= numb + 1'b1;

// counter <= 3'b000;

end

end

always @(posedge clkd) begin

if (numb == 3'b111) begin

numb <= 3'b000;

end

else

numb <= numb +1'b1;

end

always @(numb) begin

case (numb)

3'b000 : szero = 7'b0000001;

3'b001 : szero = 7'b1001111;

3'b010 : szero = 7'b0010010;

3'b011 : szero = 7'b0000110;

3'b100 : szero = 7'b1001100;

3'b101 : szero = 7'b0100100;

3'b110 : szero = 7'b0100000;

3'b111 : szero = 7'b0001111;

default: szero = 7'b1111111;

endcase

end

always @(\*) begin

case (numb)

3'b000 : led = sensor[0];

3'b001 : led = sensor[1];

3'b010 : led = sensor[2];

3'b011 : led = sensor[3];

3'b100 : led = sensor[4];

3'b101 : led = sensor[5];

3'b110 : led = sensor[6];

3'b111 : led = sensor[7];

Endcase

end