reg [31:0] counter;

reg [2:0] numb;

reg [6:0] szero;

reg tick;

reg [6:0] led;

reg clkd;

always @(posedge clk) begin

if (reset) begin

counter <= 32'b0;

numb <= 3'b000;

clkd<=1'b0;

end

else

counter <= counter +1'b1;

if (counter >= 50000000) begin

clkd <= !clkd;

counter <= 32'b0;

end

end

always @(posedge clkd) begin

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)

0:begin

if (sensor[0] == 1'b0) begin

led = 7'b0000001;

end

else led = 7'b1001111;

end

1:begin

if (sensor[1] == 1'b0) begin

led = 7'b0000001;

end

else led = 7'b1001111;

end

2:begin

if (sensor[2] == 1'b0) begin

led = 7'b0000001;

end

else led = 7'b1001111;

end

3:begin

if (sensor[3] == 1'b0) begin

led = 7'b0000001;

end

else led = 7'b1001111;

end

4:begin

if (sensor[4] == 1'b0) begin

led = 7'b0000001;

end

else led = 7'b1001111;

end

5:begin

if (sensor[5] == 1'b0) begin

led = 7'b0000001;

end

else led = 7'b1001111;

end

6:begin

if (sensor[6] == 1'b0) begin

led = 7'b0000001;

end

else led = 7'b1001111;

end

7:begin

if (sensor[7] == 1'b0) begin

led = 7'b0000001;

end

else led = 7'b1001111;

end

endcase

end

always @(\*) begin

case(tick)

1:begin

seg = 8'b11111110;

sseg = led;

end

2:begin

seg = 8'b11111101;

sseg = szero;

end

endcase

end