module trafficcontrol(

input peak,

input clk,

input rst,

input sen1,

input sen2,

output [2:0] t1,

output [2:0] t2,

output [2:0] t3,

output [2:0] t4,

output [2:0] t5,

output [2:0] t6

);

reg [2:0] t1;

reg [2:0] t2;

reg [2:0] t3;

reg [2:0] t4;

reg [2:0] t5;

reg [2:0] t6;

reg [2:0] state;

reg [4:0] count;

parameter s0=3'b000;

parameter s1=3'b001;

parameter s2=3'b010;

parameter s3=3'b011;

parameter s4=3'b100;

parameter s5=3'b101;

parameter s6=3'b110;

parameter s7=3'b111;

parameter sec32=5'd32;

parameter sec16=5'd16;

parameter sec8=5'd8;

parameter sec5=5'd5;

parameter sec3=5'd3;

always @ (posedge clk or negedge rst)

begin

if (~rst)

begin

state<=s0;

count<=0;

end

else

case (state)

s0:if(count<sec3)

begin

state<=s0;

count<=count+1;

end

else

begin

state<=s1;

count<=0;

end

s1:if (peak==1)

begin

if(count<sec32)

begin

state<=s1;

count<=count+1;

end

else if (sen1==1)

begin

state<=s2;

count<=0;

end

else if(sen2==1)

begin

state<=s5;

count<=0;

end

else

begin

state<=s1;

count<=0;

end

end

else

begin

if(count<sec16)

begin

state<=s1;

count<=count+1;

end

else if(sen1==1)

begin

state<=s2;

count<=0;

end

else if(sen2==1)

begin

state<=s5;

count<=0;

end

else

begin

state<=s1;

count<=0;

end

end

s2:if(count<sec5)

begin

state<=s2;

count<=count+1;

end

else

begin

state<=s3;

count<=0;

end

s3:if(peak==1)

begin

if(count<sec16)

begin

state<=s3;

count<=count+1;

end

else if(sen2==1)

begin

state<=s4;

count<=0;

end

else

begin

state<=s1;

count<=0;

end

end

else

begin

if(count<sec8)

begin

state<=s3;

count<=count+1;

end

else if(sen2==1)

begin

state<=s4;

count<=0;

end

else

begin

state<=s1;

count<=0;

end

end

s4:if(count<sec5)

begin

state<=s4;

count<=count+1;

end

else

begin

state<=s6;

count<=0;

end

s5:if(count<sec5)

begin

state<=s5;

count<=count+1;

end

else

begin

state<=s6;

count<=0;

end

s6:if(peak==1)

begin

if(count<sec16)

begin

state<=s6;

count<=count+1;

end

else

begin

state<=s7;

count<=0;

end

end

else

begin

if(count<sec8)

begin

state<=s6;

count<=count+1;

end

else

begin

state<=s7;

count<=0;

end

end

s7:if(count<sec5)

begin

state<=s7;

count<=count+1;

end

else

begin

state<=s1;

count<=0;

end

default:state<=s0;

endcase

end

always@(\*)

begin

case(state)

s0:begin

t1=3'b000;

t2=3'b000;

t3=3'b000;

t4=3'b000;

t5=3'b000;

t6=3'b000;

end

s1:begin

t1=3'b001;

t2=3'b001;

t3=3'b100;

t4=3'b001;

t5=3'b100;

t6=3'b001;

end

s2:begin

t1=3'b010;

t2=3'b010;

t3=3'b100;

t4=3'b010;

t5=3'b100;

t6=3'b001;

end

s3:begin

t1=3'b100;

t2=3'b100;

t3=3'b001;

t4=3'b100;

t5=3'b100;

t6=3'b001;

end

s4:begin

t1=3'b100;

t2=3'b100;

t3=3'b010;

t4=3'b100;

t5=3'b100;

t6=3'b010;

end

s5:begin

t1=3'b010;

t2=3'b001;

t3=3'b100;

t4=3'b010;

t5=3'b100;

t6=3'b010;

end

s6:begin

t1=3'b001;

t2=3'b001;

t3=3'b100;

t4=3'b100;

t5=3'b001;

t6=3'b100;

end

s7:begin

t1=3'b100;

t2=3'b001;

t3=3'b100;

t4=3'b100;

t5=3'b010;

t6=3'b100;

end

endcase

end

endmodule