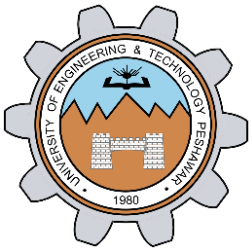
LAB # 10



Spring 2021

CSE-308L Digital System Design Lab

Submitted by: **Hurair Mohammad**

Registration No. : **18PWCSE1657**

Class Section: **B**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: _____

Submitted to:

Engr. Madiha Sher

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

MODULE

```
module traffic_light(light_highway, light_farm, C, clk, rst_n); parameter
HGRE_FRED=2'b00,HYEL_FRED =
3'b001,HRED_FGRE=3'b010,HRED_FYEL=3'b100; input C,clk,rst_n;
output reg[2:0] light_highway, light_farm; reg[27:0] count=0,count_delay=0;
reg delay10s=0,
delay3s1=0,delay3s2=0,RED_count_en=0,YELLOW_count_en1=0,YELLOW_
count_e n2=0; wire clk_enable; reg[1:0] state, next_state; always @(posedge
clk or negedge rst_n) begin if(~rst_n)  state <= 3'b001; else
 state <= next_state;  end always @(*) begin case(state)
HGRE_FRED: begin // Green on highway and red on farm way
 RED_count_en=0;
 YELLOW_count_en1=0;  YELLOW_count_en2=0;  light_highway = 3'b001;
light_farm = 3'b100;  if(C) next_state = HYEL_FRED;   else next_state
=HGRE_FRED;
end
HYEL_FRED: begin   light_highway = 3'b010;   light_farm = 3'b100;
RED_count_en=0;
 YELLOW_count_en1=1;  YELLOW_count_en2=0;
  if(delay3s1) next_state = HRED_FGRE;   else next_state = HYEL_FRED;
end
HRED_FGRE: begin  light_highway = 3'b100;  light_farm = 3'b001;
 RED_count_en=1;
 YELLOW_count_en1=0;  YELLOW_count_en2=0;
 if(delay10s) next_state = HRED_FYEL;
 // red in 10s then turn to yello -> green again for high way  else next_state
=HRED_FGRE;
end
HRED_FYEL:begin// red on highway and yellow on farm way  light_highway =
3'b100;  light_farm = 3'b010;  RED_count_en=0;
 YELLOW_count_en1=0;  YELLOW_count_en2=1;
 if(delay3s2) next_state = HGRE_FRED;  else next_state =HRED_FYEL; end
default: next_state = HGRE_FRED; endcase end
always @(posedge clk) begin
if(clk_enable==1) begin
 if(RED_count_en||YELLOW_count_en1||YELLOW_count_en2)
  count_delay <=count_delay + 1;   if((count_delay == 9)&&RED_count_en)
begin   delay10s=1;   delay3s1=0;   delay3s2=0;   count_delay<=0;   end
  else if((count_delay == 2)&&YELLOW_count_en1)   begin   delay10s=0;
delay3s1=1;   delay3s2=0;   count_delay<=0;   end
  else if((count_delay == 2)&&YELLOW_count_en2)   begin   delay10s=0;
delay3s1=0;   delay3s2=1;   count_delay<=0;   end   else   begin
   delay10s=0;   delay3s1=0;   delay3s2=0;   end   end end
always @(posedge clk) begin
 count <=count + 1;  if(count == 50000000)  if(count == 3)   count <= 0; end
```

```
 assign clk_enable = count==3 ? 1: 0;
endmodule


//////////////////////////////////////////////////////////////////////////////////////////////////////////
net "clk" loc = V10 | iostandard = LVCMOS33 | PERIOD = 50MHZ; net "rst_n"
loc = M18 | iostandard = LVCMOS33 | DRIVE = 8 | SLEW = FAST | PULLUP;
net "C" loc = F17 | iostandard = LVCMOS33 | DRIVE = 8 | SLEW = FAST |
PULLUP;
net "light_highway[0]" loc = P15 | iostandard = LVCMOS33 | DRIVE = 8 |
SLEW = FAST;
net "light_highway[1]" loc = P16 | iostandard = LVCMOS33 | DRIVE = 8 |
SLEW = FAST;
net "light_highway[2]" loc = N15 | iostandard = LVCMOS33 | DRIVE = 8 |
SLEW = FAST;
net "light_farm[0]" loc = N16 | iostandard = LVCMOS33 | DRIVE = 8 | SLEW =
FAST;
net "light_farm[1]" loc = U17 | iostandard = LVCMOS33 | DRIVE = 8 | SLEW =
FAST;
net "light_farm[2]" loc = U18 | iostandard = LVCMOS33 | DRIVE = 8 | SLEW =
FAST;
```