Project report on

Traffic Light Controller for Intersection Between Highway and Farm Way

Submitted by

Sai Sharvani Rajana BT21ECE011

A report submitted for the partial fulfilment of the requirements of the course

ECL-303 Hardware Description Languages

Submission Date : 20/11/2023

Under the guidance of:

Dr. Mayank B. Thacker

Department of Electronics and Communication Engineering

भारतीय सूचना प्रौद्योगिकी संस्थान, नागपुर

Indian Institute of Information Technology, Nagpur

Table of Contents:

Chapter No.	Particular	Page No.
1	Introduction	2
	Introduction (Continuation)	3
	Finite State Machine (FSM)	4
	ESP32 board	5
2	Code	6
	Code coninuation	7
	Code continuation	8
	Code continuation	9
	Code continuation	10
	Code ending	11
	Pin planner ,references	12

Chapter 1: Introduction

Traffic Light Controller for Intersection Between Highway and Farm Way :--

Modern transportation systems require efficient and intelligent traffic management to ensure the smooth flow of vehicles and enhance road safety. In this context, the design and implementation of traffic light controllers play a crucial role in regulating vehicular movement at intersections. This project focuses on the development of a traffic light controller made for an intersection between a highway and a farm way, addressing the unique traffic dynamics of such a scenario.

Project Objective

The primary objective of this project is to create a smart and responsive traffic light system that accommodates the specific needs of an intersection involving a highway and a farm way. The controller employs a finite state machine (FSM) to control the state transitions of the traffic lights, optimizing the traffic flow and ensuring the safety of vehicles using the farm way.

System Overview

The traffic light controller interacts with various components, including a sensor placed on the farm way, a clock signal (**clk**), and a reset signal (**rst_n**). The sensor detects the presence of vehicles on the farm way, triggering state transitions in the FSM. The controller manages the lights on both the highway (**L_H**) and the farm way (**L_F**), dictating their states based on the current conditions.

Finite State Machine (FSM)

The heart of the traffic light controller is a finite state machine, which guides the system through distinct states representing different combinations of highway and farm way lights. The FSM responds to sensor inputs and employs delay counters to control the duration of each state, ensuring a smooth and efficient traffic flow.

Project Structure

The project is structured around the FSM, which transitions through states such as:

. **HGRE_FRED:** Highway Green and Farm Red

. **HYEL_FRED:** Highway Yellow and Farm Red

. **HRED_FGRE:** Highway Red and Farm Green

. **HRED_FYEL:** Highway Red and Farm Yellow

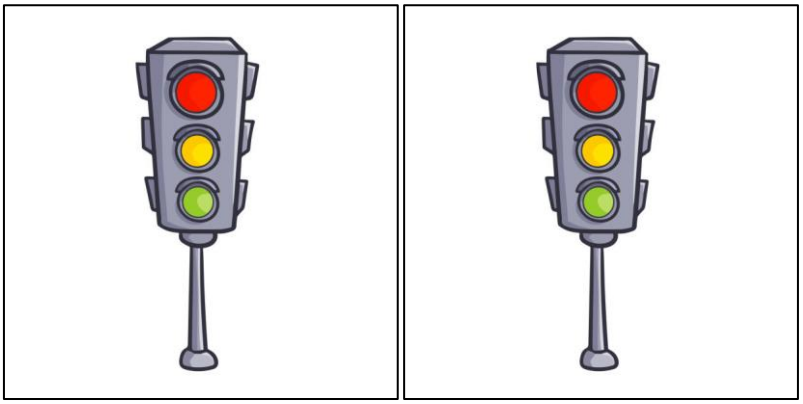


Fig :- 3.1 Traffic lights

Each state corresponds to a specific configuration of traffic lights, taking into account the presence of vehicles on the farm way and predefined delay intervals.

- If clock signal is 1 and rst_n is 0 then the process will undergo as the below flow chart Fig :- 4.1.

Finite State Machine (FSM)

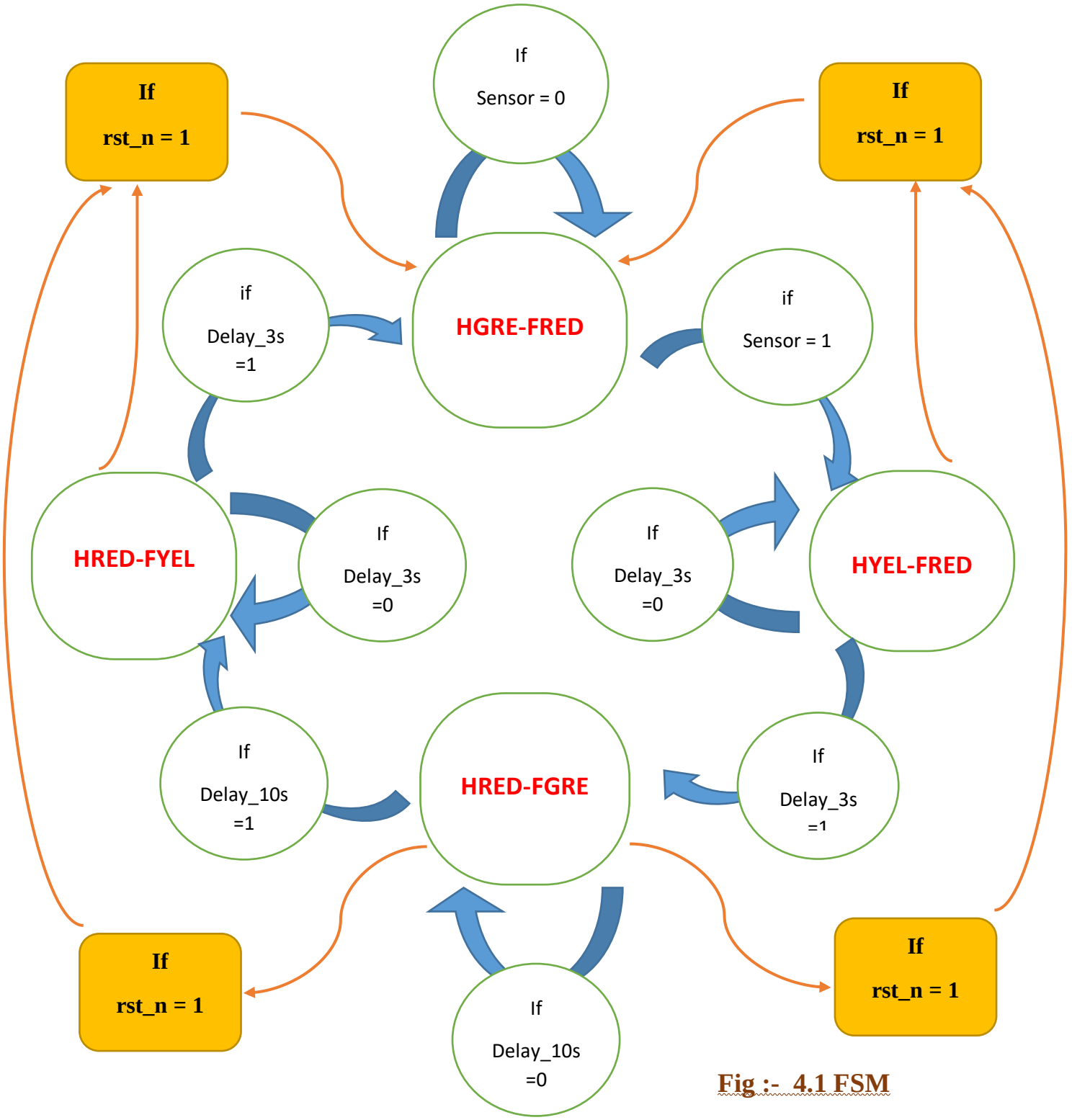


Fig :- 4.1 FSM

- If rst_n is 1 then the present state will jump to **HGRE-FRED**

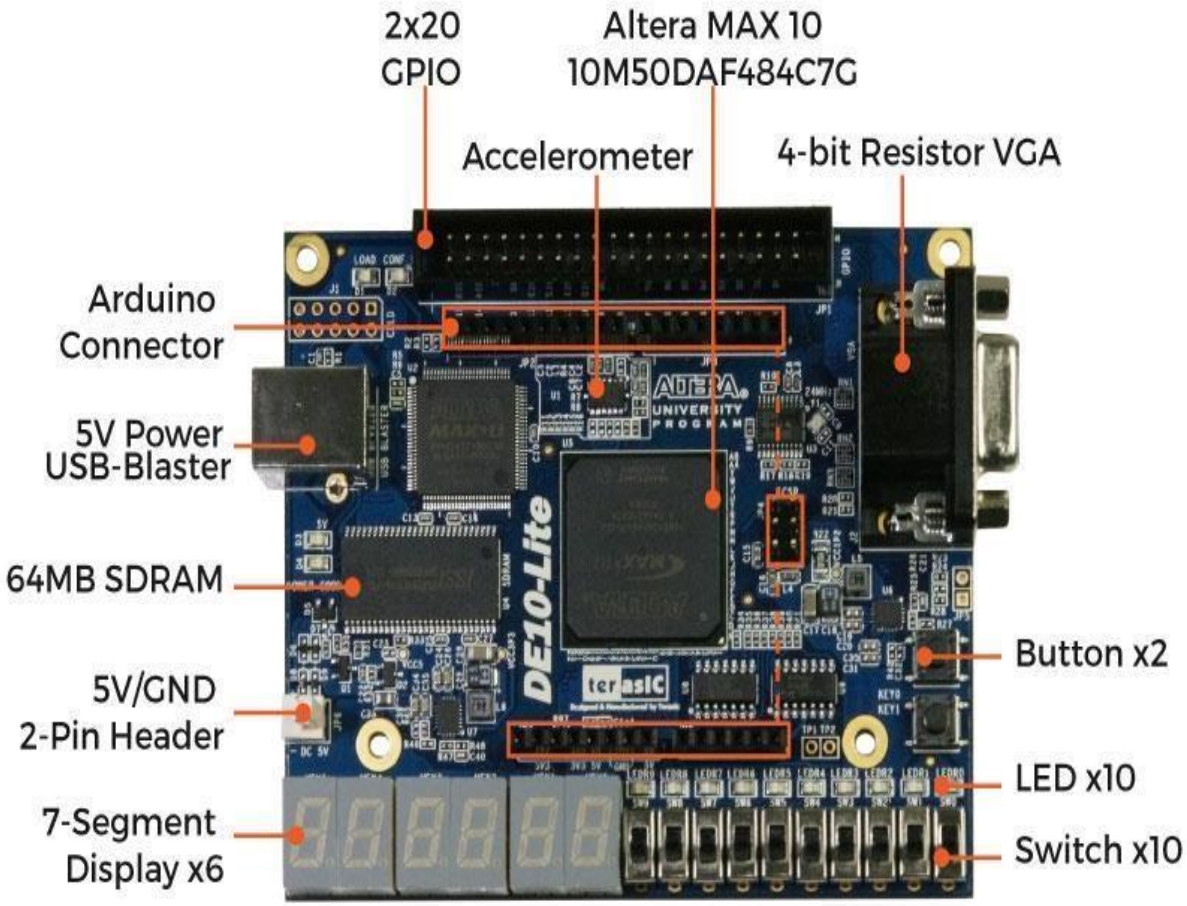


Fig. 5.1: ESP32 board

Chapter 2: Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
                -- Traffic ligh system for a intersection between highway and farm way
                -- There is a sensor on the farm way side, when there are vehicles,
                -- Traffic light turns to YELLOW, then GREEN to let the vehicles cross the highway
                -- Otherwise, always green light on Highway and Red light on farm way
entity traffic_light_controller is
 port ( sensor  : in STD_LOGIC; -- Sensor
       clk  : in STD_LOGIC;    -- clock
       rst_n: in STD_LOGIC;    -- reset active low
       L_H  : out STD_LOGIC_VECTOR(2 downto 0); -- light outputs of high way
       L_F  : out STD_LOGIC_VECTOR(2 downto 0) -- light outputs of farm way
                                   --RED_YELLOW_GREEN
     );
end traffic_light_controller;

architecture traffic_light of traffic_light_controller is
signal counter_1s: std_logic_vector(27 downto 0):= x"0000000";
signal delay_count:std_logic_vector(3 downto 0):= x"0";
signal delay_10s, delay_3s_F,delay_3s_H, R_L_ENABLE, Y_L1_ENABLE,Y_L2_ENABLE: std_logic:='0';
signal clk_1s_enable: std_logic; -- 1s clock enable
```

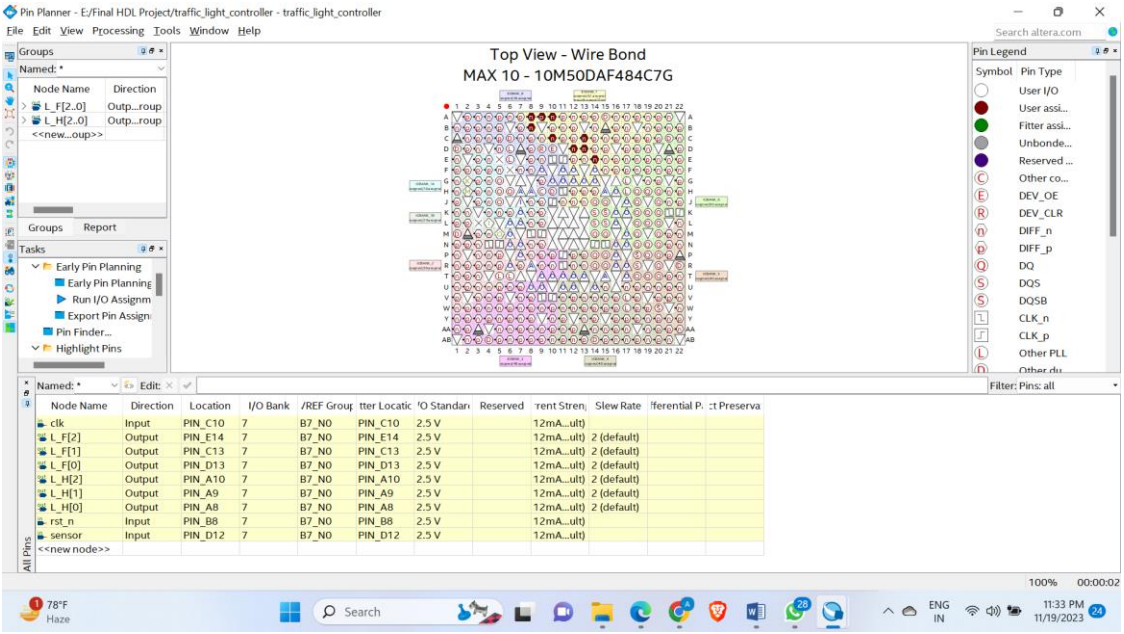
```
type FSM_States is (HGRE_FRED, HYEL_FRED, HRED_FGRE,
HRED_FYEL);
                -- HGRE_FRED : Highway green and farm red
                -- HYEL_FRED : Highway yellow and farm red
                -- HRED_FGRE : Highway red and farm green
                -- HRED_FYEL : Highway red and farm yellow
signal current_state, next_state: FSM_States;
begin
                -- next state FSM sequential logic
process(clk,rst_n)
begin
if(rst_n='0') then
 current_state <= HGRE_FRED;
elsif(rising_edge(clk)) then
 current_state <= next_state;
end if;
end process;

    -- FSM combinational logic
process(current_state,sensor,delay_3s_F,delay_3s_H,delay_10s)
begin
case current_state is
when HGRE_FRED =>       -- When Green light on Highway and Red
light on Farm way
 R_L_ENABLE <= '0';     -- disable RED light delay counting
 Y_L1_ENABLE <= '0';    -- disable YELLOW light Highway delay
counting
 Y_L2_ENABLE <= '0';    -- disable YELLOW light Farmway delay
counting
 L_H <= "001";  -- Green light on Highway
 L_F <= "100";  -- Red light on Farm way
```

```
 if(sensor = '1') then  -- if vehicle is detected on farm way by sensors
  next_state <= HYEL_FRED;

  -- High way turns to Yellow light
 else
  next_state <= HGRE_FRED;

  -- Otherwise, remains GREEN ON highway and RED on Farm way
 end if;
when HYEL_FRED => -- When Yellow light on Highway and Red light on Farm way
 L_H <= "010";    -- Yellow light on Highway
 L_F <= "100";    -- Red light on Farm way
 R_L_ENABLE <= '0';   -- disable RED light delay counting
 Y_L1_ENABLE <= '1';  -- enable YELLOW light Highway delay counting
 Y_L2_ENABLE <= '0';  -- disable YELLOW light Farmway delay counting

 if(delay_3s_H='1') then
  -- if Yellow light delay counts to 3s,
  -- turn Highway to RED,
  -- Farm way to green light
  next_state <= HRED_FGRE;
 else
  next_state <= HYEL_FRED;
  -- Remains Yellow on highway and Red on Farm way
  -- if Yellow light not yet in 3s
 end if;
when HRED_FGRE => -- When Yellow light on Highway and Red light on Farm way
```

```
 L_H <= "100";   -- RED light on Highway
 L_F <= "001";   -- GREEN light on Farm way
 R_L_ENABLE <= '1';-- enable RED light delay counting
 Y_L1_ENABLE <= '0';-- disable YELLOW light Highway delay
counting
 Y_L2_ENABLE <= '0';-- disable YELLOW light Farmway delay
counting
 if(delay_10s='1') then
 -- if RED light on highway is 10s, Farm way turns to Yellow
  next_state <= HRED_FYEL;
 else
  next_state <= HRED_FGRE;
  -- Remains if delay counts for RED light on highway not enough 10s
 end if;
when HRED_FYEL =>
 L_H <= "100";-- RED light on Highway
 L_F <= "010";-- Yellow light on Farm way
 R_L_ENABLE <= '0'; -- disable RED light delay counting
 Y_L1_ENABLE <= '0';-- disable YELLOW light Highway delay
counting
 Y_L2_ENABLE <= '1';-- enable YELLOW light Farmway delay
counting
 if(delay_3s_F='1') then
 -- if delay for Yellow light is 3s,
 -- turn highway to GREEN light
 -- Farm way to RED Light
 next_state <= HGRE_FRED;
 else
 next_state <= HRED_FYEL;
 -- if not enough 3s, remain the same state
 end if;
```

```
when others => next_state <= HGRE_FRED; -- Green on highway, red on farm way
end case;
end process;
 -- Delay counts for Yellow and RED light
process(clk)
begin
if(rising_edge(clk)) then
if(clk_1s_enable='1') then
 if(R_L_ENABLE='1' or Y_L1_ENABLE='1' or Y_L2_ENABLE='1') then
  delay_count <= delay_count + x"1";
  if((delay_count = x"9") and R_L_ENABLE ='1') then
   delay_10s <= '1';
   delay_3s_H <= '0';
   delay_3s_F <= '0';
   delay_count <= x"0";
  elsif((delay_count = x"2") and Y_L1_ENABLE= '1') then
   delay_10s <= '0';
   delay_3s_H <= '1';
   delay_3s_F <= '0';
   delay_count <= x"0";
  elsif((delay_count = x"2") and Y_L2_ENABLE= '1') then
   delay_10s <= '0';
   delay_3s_H <= '0';
   delay_3s_F <= '1';
   delay_count <= x"0";
  else
   delay_10s <= '0';
   delay_3s_H <= '0';
   delay_3s_F <= '0';
```

```
  end if;
 end if;
 end if;
end if;
end process;
 --creating 1second delay using 50MHZ clock frequency device
process(clk)
begin
if(rising_edge(clk)) then
 counter_1s <= counter_1s + x"0000001";
 if(counter_1s >= x"2FAF080") then --here,2FAF080 IS HEX conversion
of 50*10^6HZ
                                --here for every second,clock rises by 2FAF080
times
  counter_1s <= x"0000000";
 end if;
end if;
end process;
clk_1s_enable <= '1' when counter_1s = x"2FAF080" else '0';
end traffic_light;
```



Here , coming to the pin planner : -

- clk : - PIN_C10
- rst_n : - PIN_B8
- sensor:- PIN_D12
- L_H [0] : - PIN_A8
  L_H [1] : - PIN_A9
  L_H [2] : - PIN_A10
- L_F [0] : - PIN_D13
  L_F [1] : - PIN_C13
  L_F [2] : - PIN_E14

Here the traffic lights are assigned according to the above positions .

References:

1. VHDL Programming by Example
   Douglas L. Perry Fourth Edition
2. FPGA4student.com