SAVEETHA ENGINEERING COLLEGE AUTONOMOUS
AFFILIATED TO ANNA UNIVERSITY
TNEA CODE 1216

Skill Assessment

Subject Code	Subject Name
19EC526	Verilog HDL

Name of the Faculty	**Dr. Arunkumar K**
Name of the Student	**Balaji K**
Register Number	**212223060028**
Department	**ECE**
Slot Name	**4X1-26**
Submitted Date	
Marks Obtained	

1. Problem Statement

1.1 Real-World Context

Modern smart cities aim to reduce energy consumption and improve the life span of public infrastructure. One major contributor to unnecessary power usage is street lighting, where lamps remain fully ON throughout the night even when there is no traffic. This results in wasted electricity, increased maintenance costs, and higher carbon emissions.

To solve this, smart street-lighting systems automatically dim or turn off LEDs when no vehicles are present and brighten them when motion is detected. This adaptive lighting significantly reduces energy usage while maintaining road safety.

1.2 Project Objective

The objective of this project is to design a **Smart Streetlight Controller using Verilog**, which:

- Turns **OFF** lights during daytime.
- Keeps lights in **DIM mode** at night when no vehicles are detected.
- Switches to **BRIGHT mode** when a vehicle is detected.
- Uses RGB output (R, G, B) to visually demonstrate the three lighting states.

1.3 Key Modules in the Design

Sensor Input (veh_detect, day)

Simulates motion/vehicle presence and daylight detection using switches.

FSM Controller

A finite state machine controls:

- OFF state
- DIM state
- BRIGHT state

PWM Simulation

RGB outputs demonstrate brightness levels using colors:

- **Red (R)=1** → OFF

- **Green (G)=1** → DIM
- **Blue (B)=1** → BRIGHT

Timer/Clock Control

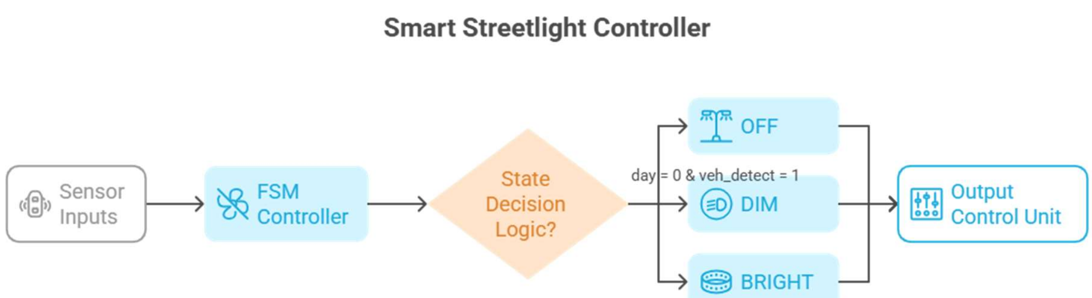
Clock drives FSM transitions.

1.4 Relevant Verilog Concepts Used

- **Finite State Machine (FSM):** State registers, next-state logic.
- **Sequential Logic:** State transitions on positive clock edge.
- **Combinational Logic:** Output logic for RGB LED.
- **Testbench Simulation:** Validates behavior using timing delays.

2. Design and Methodology

2.1 Block Diagram



2.2 Functional Description

> OFF State

- **Activated during daytime.**
- **Lights remain OFF (R=1).**

> DIM State

- **Activated during night if no vehicle is detected.**
- **Low brightness simulated using G=1.**

> BRIGHT State

- **Activated at night when a vehicle is detected.**

- **Maximum brightness simulated using B=1.**

> State Transitions

- **day = 1 → always OFF**
- **day = 0 & veh_detect = 0 → DIM**
- **day = 0 & veh_detect = 1 → BRIGHT**

3. Verilog Coding & Implementation

Inputs

• clk:

Provides the timing signal for FSM state transitions.
(Clock input)

• rst:

Resets the system back to the OFF state.
(Push button reset)

• day:

Indicates whether it is daytime.
When day = 1, the streetlight turns OFF.
(Switch or light sensor input)

• veh_detect:

Indicates the presence of a vehicle at night.
When veh_detect = 1, the light switches to BRIGHT mode.
(Motion/vehicle sensor)

Outputs

• R:

Controls the Red LED.
Used to show **OFF state** on RGB LED.
(RGB LED – Red channel)

• G:

Controls the Green LED.
Used to show **DIM state** on RGB LED.
(RGB LED – Green channel)

• **B:**

Controls the Blue LED.
Used to show **BRIGHT state** on RGB LED.
(RGB LED – Blue channel)

3.1 RTL Code

```
`timescale 1ns / 1ps

module street_light_project(

  input clk,

  input rst,

  input day,

  input veh_detect,

  output reg R,

  output reg G,

  output reg B

);


  parameter off = 2'b00;

  parameter dim = 2'b01;

  parameter bright = 2'b10;


  reg [1:0] current_state, next_state;


  always @(posedge clk or posedge rst) begin

    if (rst)
```

```
      current_state <= off;
    else
      current_state <= next_state;
  end

  always @(*) begin
    case (current_state)
      off:    next_state = day ? off : dim;
      dim:    next_state = day ? off : (veh_detect ? bright : dim);
      bright: next_state = day ? off : (!veh_detect ? dim : bright);
      default: next_state = off;
    endcase
  end

  always @(*) begin
    case (current_state)
      off:    begin R = 1; G = 0; B = 0; end
      dim:    begin R = 0; G = 1; B = 0; end
      bright: begin R = 0; G = 0; B = 1; end
      default: begin R = 0; G = 0; B = 0; end
    endcase
  end

endmodule
```

3.2 Testbench Code

```
module tb_street_light_project;
```

```
  reg clk, rst, day, veh_detect;
  wire R, G, B;

  street_light_project uut(clk, rst, day, veh_detect, R, G, B);

  always #5 clk = ~clk;

  initial begin
    clk = 0; rst = 1; day = 1; veh_detect = 0;
    #20 rst = 0;

    #100 day = 0;
    #200 veh_detect = 1;
    #200 veh_detect = 0;
    #200 day = 1;

    #200 $finish;
  end

endmodule
```

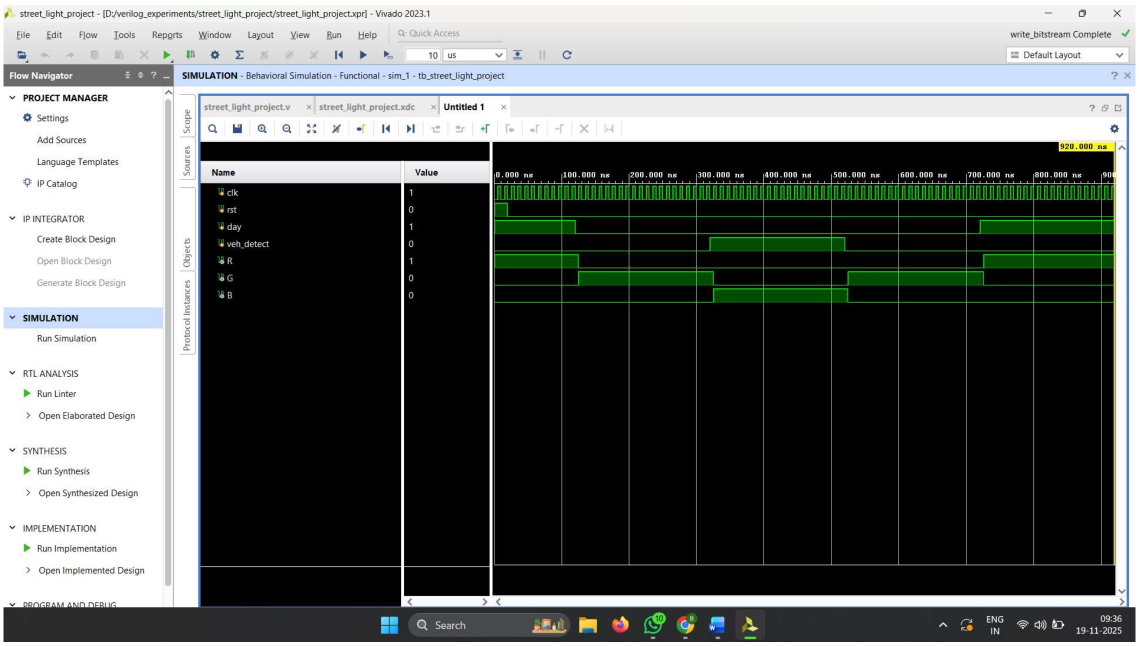
3.3 Constraint File

```
set_property -dict {PACKAGE_PIN F14 IOSTANDARD LVCMOS33} [get_ports {clk}]
set_property -dict {PACKAGE_PIN U1 IOSTANDARD LVCMOS33} [get_ports {rst}]
set_property -dict {PACKAGE_PIN V2 IOSTANDARD LVCMOS33} [get_ports {day}]
set_property -dict {PACKAGE_PIN U2 IOSTANDARD LVCMOS33} [get_ports {veh_detect}]
```

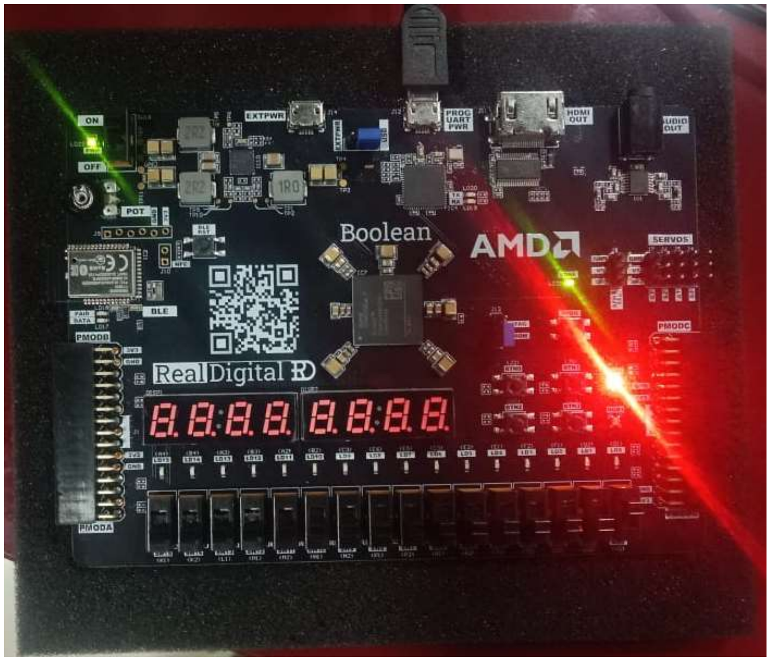
```
set_property -dict {PACKAGE_PIN V6 IOSTANDARD LVCMOS33} [get_ports {R}]

set_property -dict {PACKAGE_PIN V4 IOSTANDARD LVCMOS33} [get_ports {G}]

set_property -dict {PACKAGE_PIN U6 IOSTANDARD LVCMOS33} [get_ports {B}]
```

4. Output and Result

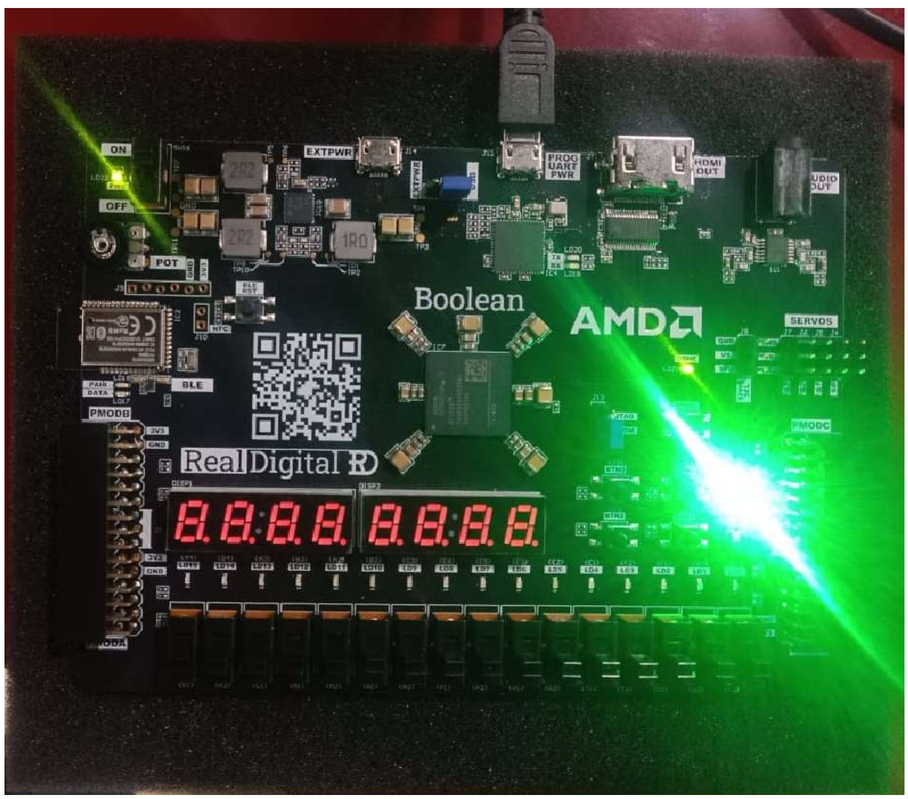
4.1. Testbench Output



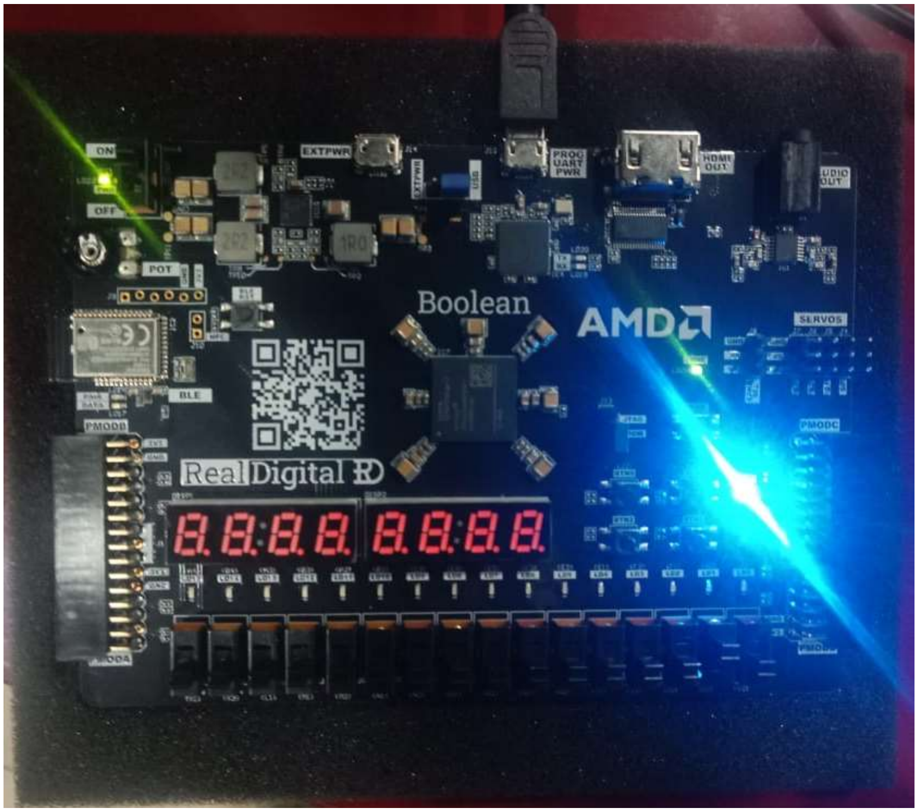
4.2. Spartan-7 Boolean FPGA Implementation Output



a) In day time



b) In night time but no vehicle detects



c)In night time as well as vehicle detects

4.3. Result

The Smart Streetlight Controller was successfully implemented using Verilog HDL. The FSM-based design demonstrated correct transitions between OFF, DIM, and BRIGHT states depending on:

- Day/night condition
- Vehicle presence

RGB outputs accurately represented each state in simulation.

This design is scalable and can be extended to include:

- Real PWM brightness control
- Multiple streetlights
- Ultrasonic/IR sensor inputs
- IoT-based monitoring