PES UNIVERSITY

Digital Design and Computer Organisation Laboratory (UE23CS251A)

3rd Semester, Academic Year 2024

MINI PROJECT

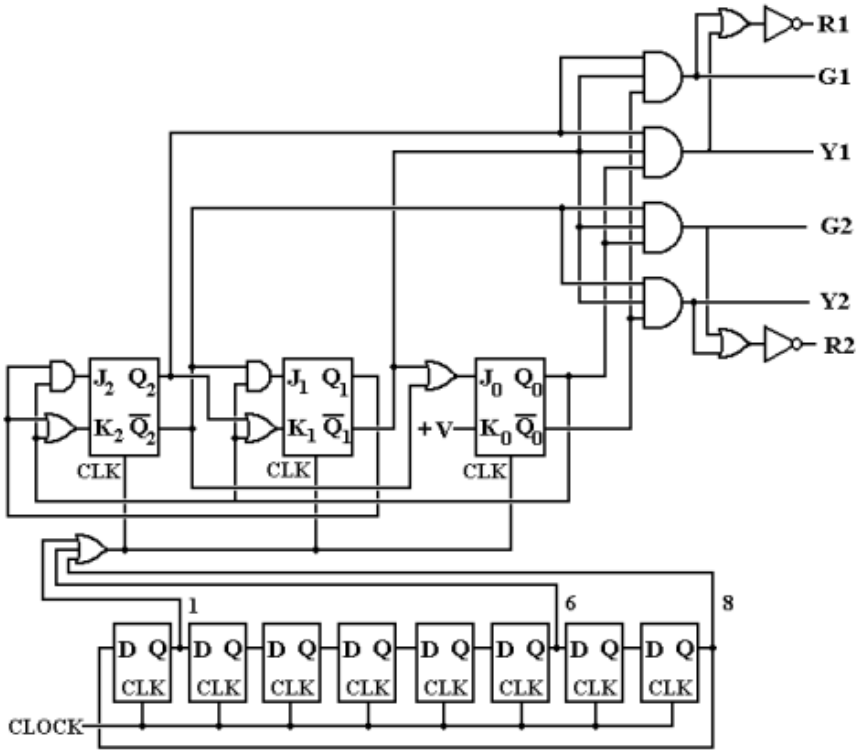
Date: 11-11-2024 Section: H

Name	SRN
Pranav Rajesh Narayan	PES1UG23CS435
Praneet Tenkila	PES1UG23CS439
Pranshu Malhotra	PES1UG23CS443
Dhanush M	PES1UG23CS920

Problem statement: Implement the control logic for the traffic lights, handle timing constraints, and simulate the system's behavior.

PES UNIVERSITY

Digital circuit:



Truth table:

	Alias	$Q_2Q_1Q_0$	R1	G1	Y1	R2	G2	Y2
0	RR	0 0 0	1	0	0	1	0	0
1	RG	0 0 1	1	0	0	0	1	0
2	RY	0 1 0	1	0	0	0	0	1
3	RR	0 1 1	1	0	0	1	0	0
4	GR	1 0 0	0	1	0	1	0	0
5	YR	1 0 1	0	0	1	1	0	0
6	RR	1 1 0	1	0	0	1	0	0
7	RR	1 1 1	1	0	0	1	0	0

Here are the output equations

$G1 = Q_2 \bullet Q_1' \bullet Q_0'$ $\quad G2 = Q_2' \bullet Q_1' \bullet Q_0$

$Y1 = Q_2 \bullet Q_1' \bullet Q_0$ $\quad Y2 = Q_2' \bullet Q_1 \bullet Q_0'$

$R1 = (G1 + Y1)'$ $\quad R2 = (G2 + Y2)'$

Verilog module:

raffic_light_controller.v

```
module jk_flipflop (
    input wire j, k, clk, reset,
    output reg q
);
    always @(posedge clk or posedge reset) begin    // on positive edge or reset high
        if (reset) q <= 0;
        else if (j && ~k) q <= 1;   // sets to 1
        else if (~j && k) q <= 0;   // sets to 0
        else if (j && k) q <= ~q;   // toggles
    end
endmodule
```

```
module traffic_light_controller (
    input wire clk, reset,
    output wire R1, Y1, G1,   // North-South lights
    output wire R2, Y2, G2    // East-West lights
);

    wire Q2, Q1, Q0;

    // Flip-flop inputs based on derived equations
    wire J2 = Q1 & Q0;
    wire K2 = Q1 | Q0;
    wire J1 = ~Q2 & Q0;
    wire K1 = Q2 | Q0;
    wire J0 = ~Q2 | ~Q1;
    wire K0 = 1;

    // Instantiate JK flip-flops
    jk_flipflop ff2 (J2, K2, clk, reset, Q2);
    jk_flipflop ff1 (J1, K1, clk, reset, Q1);
    jk_flipflop ff0 (J0, K0, clk, reset, Q0);

    // Output equations for lights
    assign G1 = Q2 & ~Q1 & ~Q0;
    assign Y1 = Q2 & ~Q1 & Q0;
    assign G2 = ~Q2 & ~Q1 & Q0;
    assign Y2 = ~Q2 & Q1 & ~Q0;

    assign R1 = ~(G1 | Y1);     // to ensure that when other lights are on red is off
    assign R2 = ~(G2 | Y2);

endmodule
```

Test bench:

```
module testbench;
    reg clk, reset;
    wire R1, Y1, G1, R2, Y2, G2;

    traffic_light_controller uut (
        .clk(clk),
        .reset(reset),
        .R1(R1), .Y1(Y1), .G1(G1),
        .R2(R2), .Y2(Y2), .G2(G2)
    );

    // Clock generation
    initial begin
        clk = 0;
        forever #5 clk = ~clk;
    end

    // Test sequence
    initial begin
        $dumpfile("traffic_light_controller.vcd");
        $dumpvars(0, testbench);
        reset = 1;
        #10 reset = 0;
        #100 $finish;
    end

    initial begin
        $monitor("Time=%0t R1=%b Y1=%b G1=%b | R2=%b Y2=%b G2=%b",
                 $time, R1, Y1, G1, R2, Y2, G2);
    end
endmodule
```

Output:

```
C:\iverilog\bin>vvp traffic
VCD info: dumpfile traffic_light_controller.vcd opened for output.
Time=0 R1=1 Y1=0 G1=0 | R2=1 Y2=0 G2=0
Time=15 R1=1 Y1=0 G1=0 | R2=0 Y2=0 G2=1
Time=25 R1=1 Y1=0 G1=0 | R2=0 Y2=1 G2=0
Time=35 R1=1 Y1=0 G1=0 | R2=1 Y2=0 G2=0
Time=45 R1=0 Y1=0 G1=1 | R2=1 Y2=0 G2=0
Time=55 R1=0 Y1=1 G1=0 | R2=1 Y2=0 G2=0
Time=65 R1=1 Y1=0 G1=0 | R2=1 Y2=0 G2=0
Time=75 R1=1 Y1=0 G1=0 | R2=0 Y2=0 G2=1
Time=85 R1=1 Y1=0 G1=0 | R2=0 Y2=1 G2=0
Time=95 R1=1 Y1=0 G1=0 | R2=1 Y2=0 G2=0
Time=105 R1=0 Y1=0 G1=1 | R2=1 Y2=0 G2=0
```

GTK wave simulation: