FPGA Implementation of Freeway Traffic Control System

EE271

Priya Dilip Shinde, Student ID-01256511

Aniket Phatak, Student ID-012564994

Department of Electrical Engineering, San Jose State University

{priyadilip.shinde, aniket.phatak}@ sjsu.edu

Abstract:

This project implements traffic-light controller on a certain patch of the freeway. With the help of FPGA board this project focuses on the pedestrian as well as vehicle traffic light management to reduce congestion, increase pedestrian walk ability and increase pedestrian safety. This project is focused on a part of a freeway where the east-west corridor is a one-way street and has two intersection: one is towards south direction and other towards north direction. Each of these intersections has separate pedestrian crosswalks.

Table of Content:

Introduction:

The general idea of this project is to help the local government, citizens and pedestrians overcome congestion caused by vehicles. To overcome this problem, this project will allow management of traffic lights, which in turn solve the problem of congestion. For the sake of efficiency, we have used the Altera Cyclone II board. With a memory of 8-Mbyte SDRAM and 512 Kb SRAM, this board also has SMA connector that can be used as an external clock input. The board has many other additional features like 1 to 4 Mbyte flash , four 7-segment display, ten green and red LED's, USB port, ten switch and four push buttons. Fig 1. Shows the map where this project can be implemented.

In figure 1, the east-west corridor is a one-way street and is coordinated with the 10th street. There are total of two traffic lights in this area with a dedicated pathway for the pedestrian to cross the roads i.e. crosswalks. These two crosswalks are also in synchronization with the 10th street traffic lights. Addition to this there are two intersection with two more pedestrian crosswalks. Fig 2 shows the complete sketch of the area of concentration where this can be implemented with four pedestrian crosswalks. S1, S2, S3 and S4 from figure 2 indicated traffic lights.

Fig 1. The scenario of project implementation.

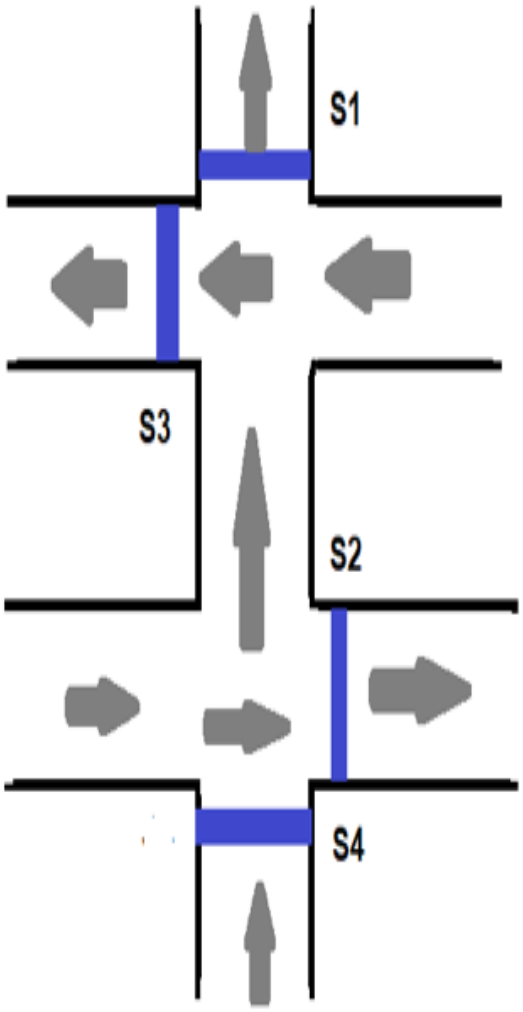
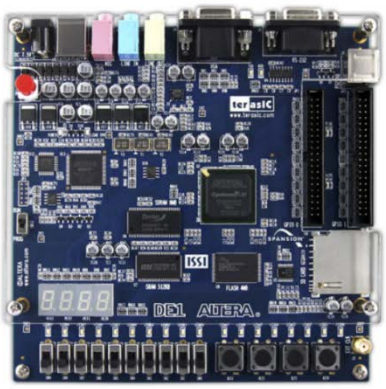


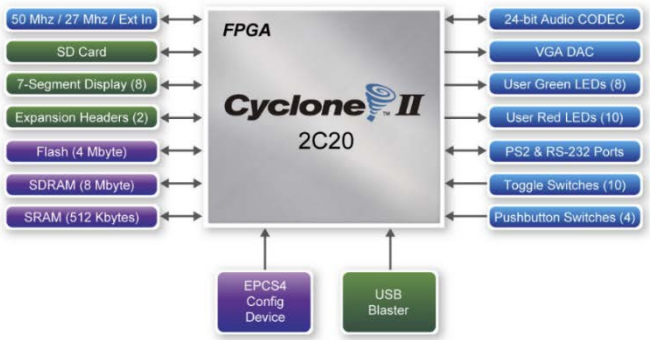
Fig 2. Complete sketch of area of concentration.

FPGA Implementation of Freeway

Board Image

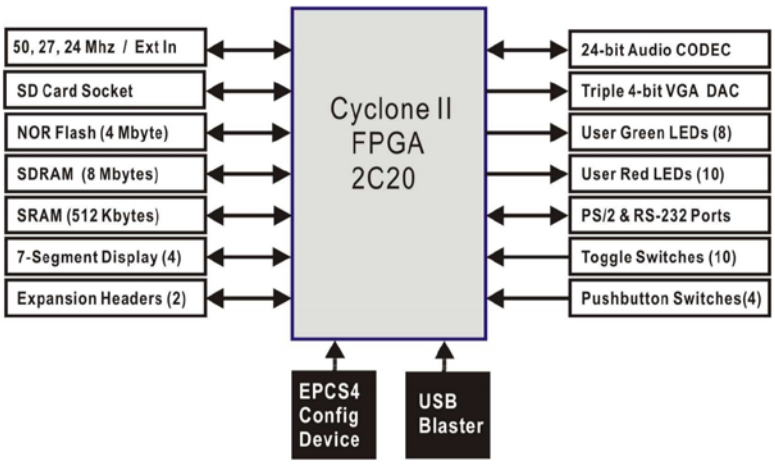


Block Diagram



Block Diagram of the DE1 Board

Figure gives the block diagram of the DE1 board. To provide maximum flexibility for the user, all connections are made through the Cyclone II FPGA device. Thus, the user can configure the FPGA to implement any system design.



Signal Name	FPGA Pin No.	Description
SW[0]	PIN_L22	Toggle Switch[0]
SW[1]	PIN_L21	Toggle Switch[1]
SW[2]	PIN_M22	Toggle Switch[2]
SW[3]	PIN_V12	Toggle Switch[3]
SW[4]	PIN_W12	Toggle Switch[4]
SW[5]	PIN_U12	Toggle Switch[5]
SW[6]	PIN_U11	Toggle Switch[6]
SW[7]	PIN_M2	Toggle Switch[7]

Signal Name	FPGA Pin No.	Description
KEY[0]	PIN_R22	Pushbutton[0]
KEY[1]	PIN_R21	Pushbutton[1]
KEY[2]	PIN_T22	Pushbutton[2]
KEY[3]	PIN_T21	Pushbutton[3]

Table 4.2. Pin assignments for the pushbutton switches.

Signal Name	FPGA Pin No.	Description
LEDR[0]	PIN_R20	LED Red[0]
LEDR[1]	PIN_R19	LED Red[1]
LEDR[2]	PIN_U19	LED Red[2]
LEDR[3]	PIN_Y19	LED Red[3]
LEDR[4]	PIN_T18	LED Red[4]
LEDR[5]	PIN_V19	LED Red[5]
LEDR[6]	PIN_Y18	LED Red[6]
LEDR[7]	PIN_U18	LED Red[7]
LEDR[8]	PIN_R18	LED Red[8]
LEDR[9]	PIN_R17	LED Red[9]
LEDG[0]	PIN_U22	LED Green[0]
LEDG[1]	PIN_U21	LED Green[1]
LEDG[2]	PIN_V22	LED Green[2]
LEDG[3]	PIN_V21	LED Green[3]
LEDG[4]	PIN_W22	LED Green[4]
LEDG[5]	PIN_W21	LED Green[5]
LEDG[6]	PIN_Y22	LED Green[6]
LEDG[7]	PIN_Y21	LED Green[7]

Table 4.3. Pin assignments for the LEDs.

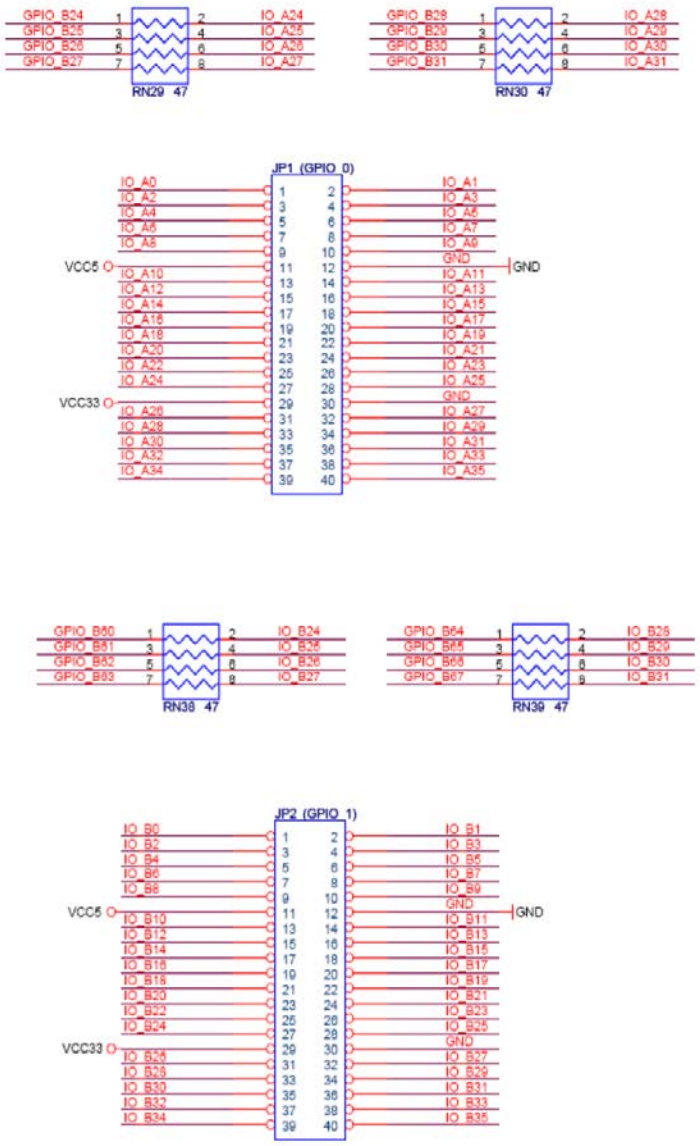


Figure 4.10. Schematic diagram of the expansion headers.

Signal Name	FPGA Pin No.	Description
GPIO_0[0]	PIN_A13	GPIO Connection 0[0]

GPIO_0[1]	PIN_B13	GPIO Connection 0[1]
GPIO_0[2]	PIN_A14	GPIO Connection 0[2]
GPIO_0[3]	PIN_B14	GPIO Connection 0[3]
GPIO_0[4]	PIN_A15	GPIO Connection 0[4]
GPIO_0[5]	PIN_B15	GPIO Connection 0[5]
GPIO_0[6]	PIN_A16	GPIO Connection 0[6]
GPIO_0[7]	PIN_B16	GPIO Connection 0[7]
GPIO_0[8]	PIN_A17	GPIO Connection 0[8]
GPIO_0[9]	PIN_B17	GPIO Connection 0[9]
GPIO_0[10]	PIN_A18	GPIO Connection 0[10]
GPIO_0[11]	PIN_B18	GPIO Connection 0[11]
GPIO_0[12]	PIN_A19	GPIO Connection 0[12]
GPIO_0[13]	PIN_B19	GPIO Connection 0[13]
GPIO_0[14]	PIN_A20	GPIO Connection 0[14]
GPIO_0[15]	PIN_B20	GPIO Connection 0[15]
GPIO_0[16]	PIN_C21	GPIO Connection 0[16]
GPIO_0[17]	PIN_C22	GPIO Connection 0[17]
GPIO_0[18]	PIN_D21	GPIO Connection 0[18]
GPIO_0[19]	PIN_D22	GPIO Connection 0[19]
GPIO_0[20]	PIN_E21	GPIO Connection 0[20]
GPIO_0[21]	PIN_E22	GPIO Connection 0[21]
GPIO_0[22]	PIN_F21	GPIO Connection 0[22]
GPIO_0[23]	PIN_F22	GPIO Connection 0[23]
GPIO_0[24]	PIN_G21	GPIO Connection 0[24]
GPIO_0[25]	PIN_G22	GPIO Connection 0[25]
GPIO_0[26]	PIN_J21	GPIO Connection 0[26]
GPIO_0[27]	PIN_J22	GPIO Connection 0[27]
GPIO_0[28]	PIN_K21	GPIO Connection 0[28]
GPIO_0[29]	PIN_K22	GPIO Connection 0[29]
GPIO_0[30]	PIN_J19	GPIO Connection 0[30]
GPIO_0[31]	PIN_J20	GPIO Connection 0[31]
GPIO_0[32]	PIN_J18	GPIO Connection 0[32]
GPIO_0[33]	PIN_K20	GPIO Connection 0[33]
GPIO_0[34]	PIN_L19	GPIO Connection 0[34]
GPIO_0[35]	PIN_L18	GPIO Connection 0[35]
GPIO_1[0]	PIN_H12	GPIO Connection 1[0]
GPIO_1[1]	PIN_H13	GPIO Connection 1[1]

32

GPIO_1[2]	PIN_H14	GPIO Connection 1[2]
GPIO_1[3]	PIN_G15	GPIO Connection 1[3]
GPIO_1[4]	PIN_E14	GPIO Connection 1[4]
GPIO_1[5]	PIN_E15	GPIO Connection 1[5]
GPIO_1[6]	PIN_F15	GPIO Connection 1[6]
GPIO_1[7]	PIN_G16	GPIO Connection 1[7]
GPIO_1[8]	PIN_F12	GPIO Connection 1[8]
GPIO_1[9]	PIN_F13	GPIO Connection 1[9]
GPIO_1[10]	PIN_C14	GPIO Connection 1[10]
GPIO_1[11]	PIN_D14	GPIO Connection 1[11]
GPIO_1[12]	PIN_D15	GPIO Connection 1[12]
GPIO_1[13]	PIN_D16	GPIO Connection 1[13]
GPIO_1[14]	PIN_C17	GPIO Connection 1[14]
GPIO_1[15]	PIN_C18	GPIO Connection 1[15]
GPIO_1[16]	PIN_C19	GPIO Connection 1[16]
GPIO_1[17]	PIN_C20	GPIO Connection 1[17]
GPIO_1[18]	PIN_D19	GPIO Connection 1[18]
GPIO_1[19]	PIN_D20	GPIO Connection 1[19]
GPIO_1[20]	PIN_E20	GPIO Connection 1[20]
GPIO_1[21]	PIN_F20	GPIO Connection 1[21]
GPIO_1[22]	PIN_E19	GPIO Connection 1[22]
GPIO_1[23]	PIN_E18	GPIO Connection 1[23]
GPIO_1[24]	PIN_G20	GPIO Connection 1[24]
GPIO_1[25]	PIN_G18	GPIO Connection 1[25]
GPIO_1[26]	PIN_G17	GPIO Connection 1[26]
GPIO_1[27]	PIN_H17	GPIO Connection 1[27]
GPIO_1[28]	PIN_J15	GPIO Connection 1[28]
GPIO_1[29]	PIN_H18	GPIO Connection 1[29]
GPIO_1[30]	PIN_N22	GPIO Connection 1[30]
GPIO_1[31]	PIN_N21	GPIO Connection 1[31]
GPIO_1[32]	PIN_P15	GPIO Connection 1[32]
GPIO_1[33]	PIN_N15	GPIO Connection 1[33]
GPIO_1[34]	PIN_P17	GPIO Connection 1[34]
GPIO_1[35]	PIN_P18	GPIO Connection 1[35]

Table 4.7. Pin assignments for the expansion headers.

Proposed Design:

The proposed design specifies the adequate amount of ‘stop’ time as well as the ‘go’ time for the vehicles. This time determines the amount of traffic flow from one traffic light to another and this time also allows pedestrians to walk from one street point to another street point using the crosswalk signal indicator. For this project, we have determined that an 8 sec is adequate time for the vehicles to move (green) and avoid congestion. With this in mind, we have allocated 2 sec for vehicles to get a warning that they need to stop (yellow) and another 10 sec for a complete stop for that particular signal (red). As in figure 2, the east west corridor has two signals i.e. S1 and S2. The proposed design is developed in such a way that when S1 gets a red signal, S2 will also get a red signal in precise 10 sec. This avoids congestion in the streets and the traffic is quite manageable even if there is a congestion. Initially, all the streetlights will be ‘red’ indicating the default state or the state zero. As we proceed ahead, we will get different states, which are shown, in figure 3. Each state, if not met with the criteria or condition will yield the same result i.e. its own state.

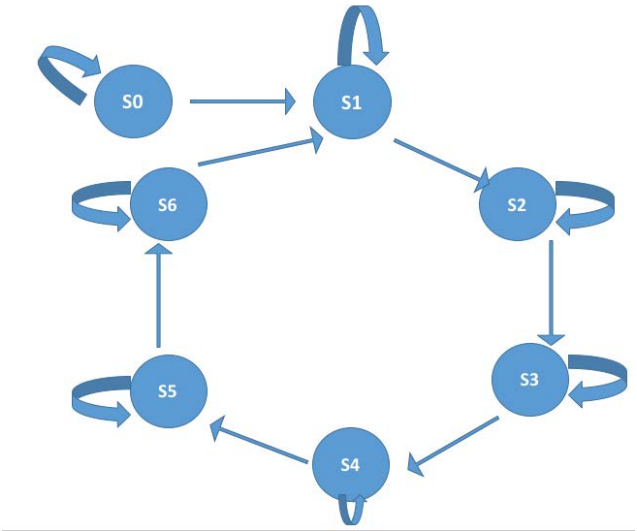
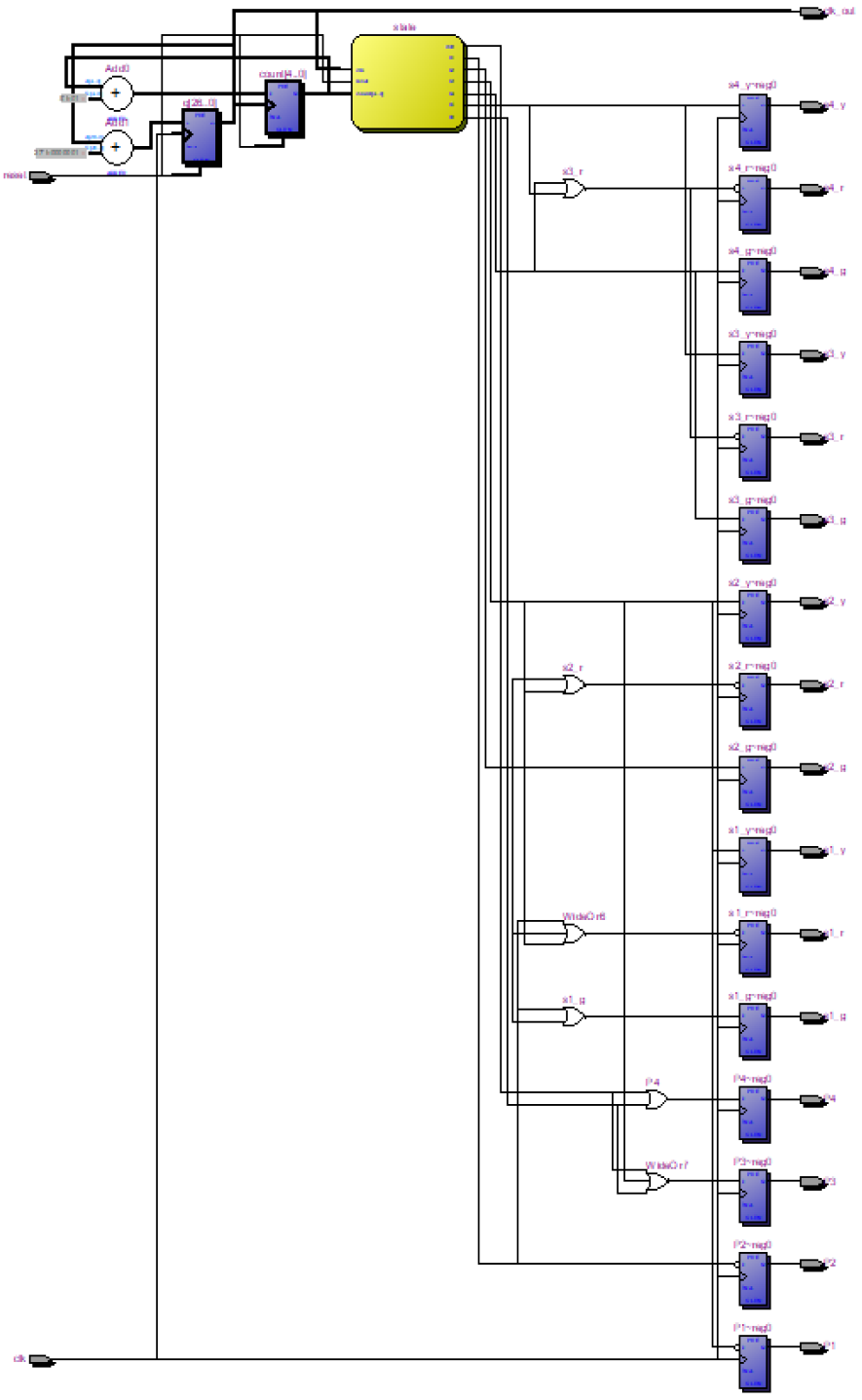


Fig 3. State diagram

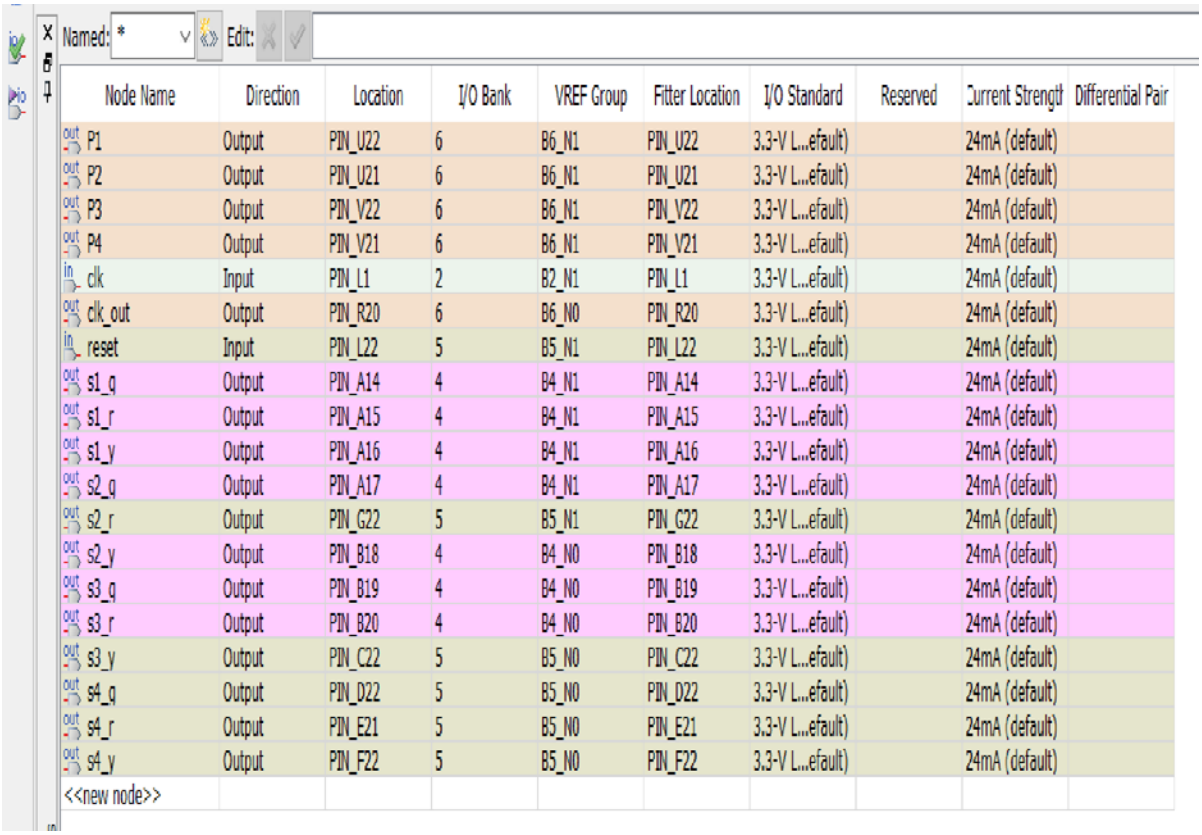
As shown in the figure there are total of seven states, which comprise of traffic light management. The proposed design also focus on the pedestrian pathways. Whenever S1 and S2 are ‘green’ (see fig. 2) we are allowing the pedestrian signal to go on i.e. ‘walk’. This condition will be on or around 10 sec and after that, when S1 and S2 are ‘red it will turn to ‘stop’. Similarly, when S3 and S4 are ‘red’ the pedestrian signal will be for ‘walk’ and will turn to ‘stop’ when the signals for S3 and S4 are ‘green’. Signal S4 is allowed to go green after state S1 goes green, then yellow and then it turns to red with the specific time delay. Then signal S2 and S3 turns green as soon as S1 and S4 turns red.

RTL View:

The following represents the RTL schematic of the entire system



Pin Assignments:



Named: * Edit:

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
P1	Output	PIN_U22	6	B6_N1	PIN_U22	3.3-V L...efault)		24mA (default)	
P2	Output	PIN_U21	6	B6_N1	PIN_U21	3.3-V L...efault)		24mA (default)	
P3	Output	PIN_V22	6	B6_N1	PIN_V22	3.3-V L...efault)		24mA (default)	
P4	Output	PIN_V21	6	B6_N1	PIN_V21	3.3-V L...efault)		24mA (default)	
clk	Input	PIN_L1	2	B2_N1	PIN_L1	3.3-V L...efault)		24mA (default)	
clk_out	Output	PIN_R20	6	B6_N0	PIN_R20	3.3-V L...efault)		24mA (default)	
reset	Input	PIN_L22	5	B5_N1	PIN_L22	3.3-V L...efault)		24mA (default)	
s1_g	Output	PIN_A14	4	B4_N1	PIN_A14	3.3-V L...efault)		24mA (default)	
s1_r	Output	PIN_A15	4	B4_N1	PIN_A15	3.3-V L...efault)		24mA (default)	
s1_y	Output	PIN_A16	4	B4_N1	PIN_A16	3.3-V L...efault)		24mA (default)	
s2_g	Output	PIN_A17	4	B4_N1	PIN_A17	3.3-V L...efault)		24mA (default)	
s2_r	Output	PIN_G22	5	B5_N1	PIN_G22	3.3-V L...efault)		24mA (default)	
s2_y	Output	PIN_B18	4	B4_N0	PIN_B18	3.3-V L...efault)		24mA (default)	
s3_g	Output	PIN_B19	4	B4_N0	PIN_B19	3.3-V L...efault)		24mA (default)	
s3_r	Output	PIN_B20	4	B4_N0	PIN_B20	3.3-V L...efault)		24mA (default)	
s3_y	Output	PIN_C22	5	B5_N0	PIN_C22	3.3-V L...efault)		24mA (default)	
s4_g	Output	PIN_D22	5	B5_N0	PIN_D22	3.3-V L...efault)		24mA (default)	
s4_r	Output	PIN_E21	5	B5_N0	PIN_E21	3.3-V L...efault)		24mA (default)	
s4_y	Output	PIN_F22	5	B5_N0	PIN_F22	3.3-V L...efault)		24mA (default)	
<<new node>>									

Implementation:

The logical implementation of this project is shown in appendix, which shows how the code goes from one state to another following the state diagram. The first part of the code focuses on whether the condition needs a reset or no. According to that, the count remains the same or changes. Rest of code sheds light on how the states are been shifted from one state to another. Each state is called to execute and is been distinguished from other states with the 'count' variable. The variable 'next state' determines which state will be followed by the previous state.

Simulation result:

The simulation result show how and when a certain signal will go on and off in a simulated environment. If you see figure 7 we can see how the waveforms alter with the change from one state to another. At exactly three, there is a long delay, which lasts until the end of 10.

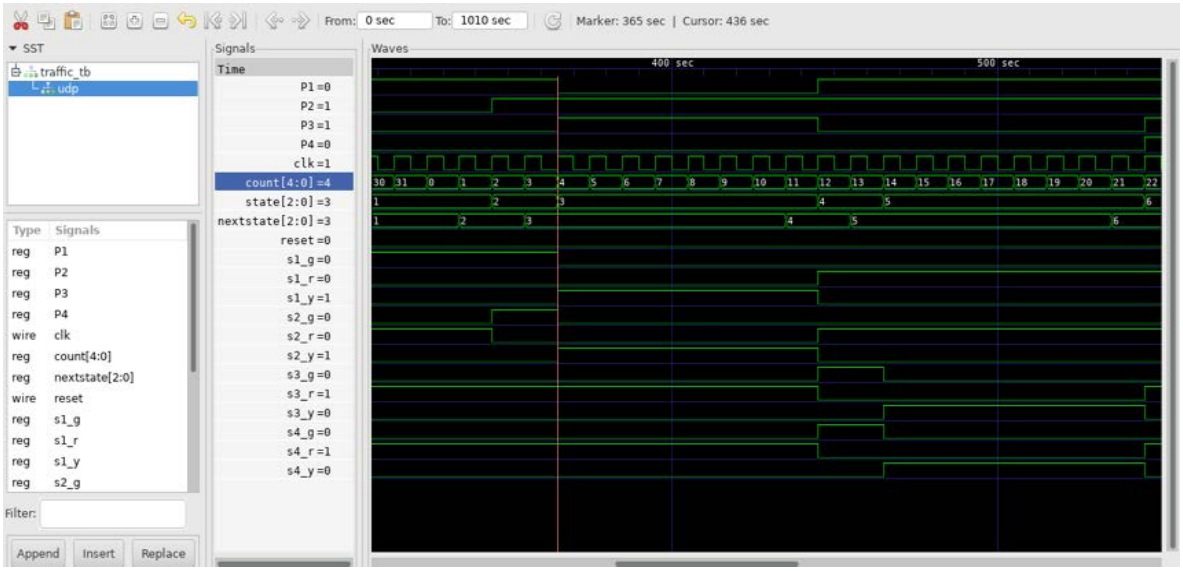


Fig. 4 Simulation result

Completion Result:

Figure 5. show the completion of the project and its working result. It shows specific details regarding the implementation.

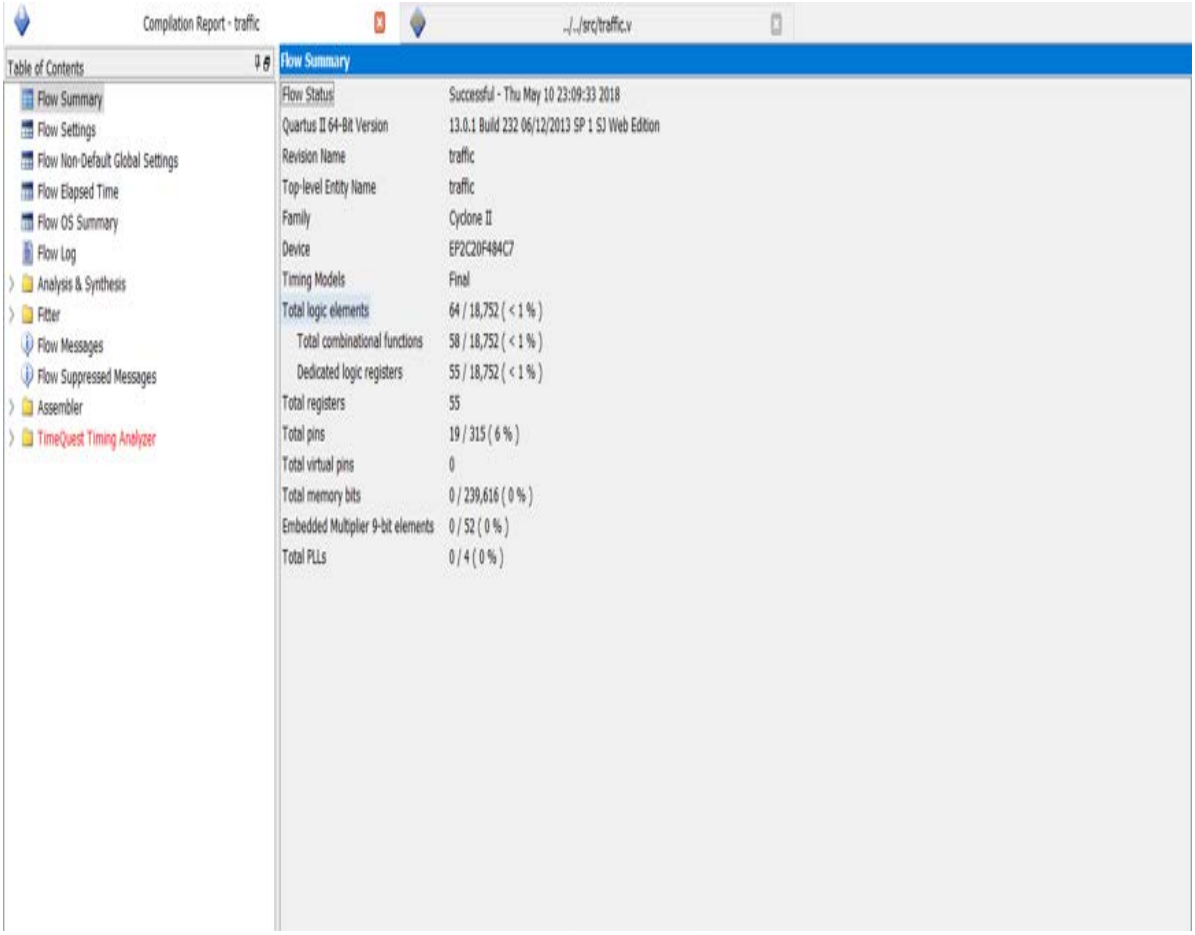


Fig 5. Completion result

Conclusion:

With the increasing rate of traffic condition and reduction of pedestrian safety, this project surely shows how proper management can be used to avoid congestion and increase pedestrian safety. The FPGA board has helped to increase the overall efficiency and has helped to create a stable system. Each state of the system are been carefully designed in such a way that equal and proportional time is given for each signal, either pedestrian or traffic, to turn ‘green’ and then ‘red’. In conclusion, if we implement this system on the proposed area, we can surely get less congestion on vehicles and increase in vehicle mobility.

Appendix

Test bench:

```
//traffic light submission

`include "traffic.v"

module trafficlight_tb();

reg clk, reset;

wire s1_g, s1_r, s1_y, s2_g, s2_r, s2_y, s3_g, s3_r, s3_y, s4_g, s4_r, s4_y,    P1_r, P1_g, P2_r, P2_g, P3_r, P3_g, P4_r, P4_g;

traffic udp(.clk(clk), .reset(reset), .s1_g(s1_g), .s1_r(s1_r), .s1_y(s1_y), .s2_g(s2_g), .s2_r(s2_r), .s2_y(s2_y), .s3_g(s3_g), .s3_r(s3_r), .s3_y(s3_y),

                .s4_g(s4_g), .s4_r(s4_r), .s4_y(s4_y), .P1_r(P1_r), .P1_g(P1_g), .P2_r(P2_r), .P2_g(P2_g), .P3_r(P3_r), .P3_g(P3_g), .P4_r(P4_r), .P4_g(P4_g));


always #10 clk = ~clk;

initial begin

    $dumpfile("traffic.vcd");

    $dumpvars(0);

   end


initial begin

$monitor("time=%g clk=%b reset=%b s1_g=%b s1_r=%b s1_y=%b s2_g=%b s2_r=%b s2_y=%b s3_g=%b s3_r=%b s3_y=%b s4_g=%b s4r=%b s4_y=%b P1_r=%b, P1_g=%b, P2_r=%b, P2_g=%b, P3_r=%b, P3_g=%b, P4_r=%b, P4_g=%b",clk,reset,s1_g, s1_r,s1_y,s2_g,s2_r, s2_y,s3_g, s3_r, s3_y,s4_g,s4_r,s4_y, P1_r, P1_g, P2_r, P2_g, P3_r, P3_g, P4_r, P4_g, $time);

        reset = 1'b0;

    #10 reset = 1'b1;

    #10 reset = 1'b0;

    #50 $finish;

end

endmodule
```

References:

- https://www.altera.com/solutions/partners/partner-profile/terasic-inc-/board/altera-de1-board.html#overview
- https://www.altera.com/content/dam/altera-www/global/en_US/portal/dsn/42/doc-us-dsnbk-42-4904342209-de1-usermanual.pdf
- http://www.asic-world.com/verilog/veritut.html
- https://www.altera.com/support/support-resources/design-examples/design-software/verilog/ver_statem.html
- http://testbench.in/TB_08_CLOCK_GENERATOR.html

https://www.altera.com/en_US/pdfs/literature/wp/wp-01003.pdf