1 Introduction

1.1 Brief overview

A 4-way traffic controller is designed with four roads labelled as North, East and south and west. The signal go through the sequence North -> East -> South -> West with sequence of light in each road is Green -> Orange -> Red . It basically has six interconnected modules i.e. four Sensor unit (one for each road), one adaptation unit and one display timer unit. The display timer is our control path and the other modules are data paths. The top-level architecture is shown in Figure 1.

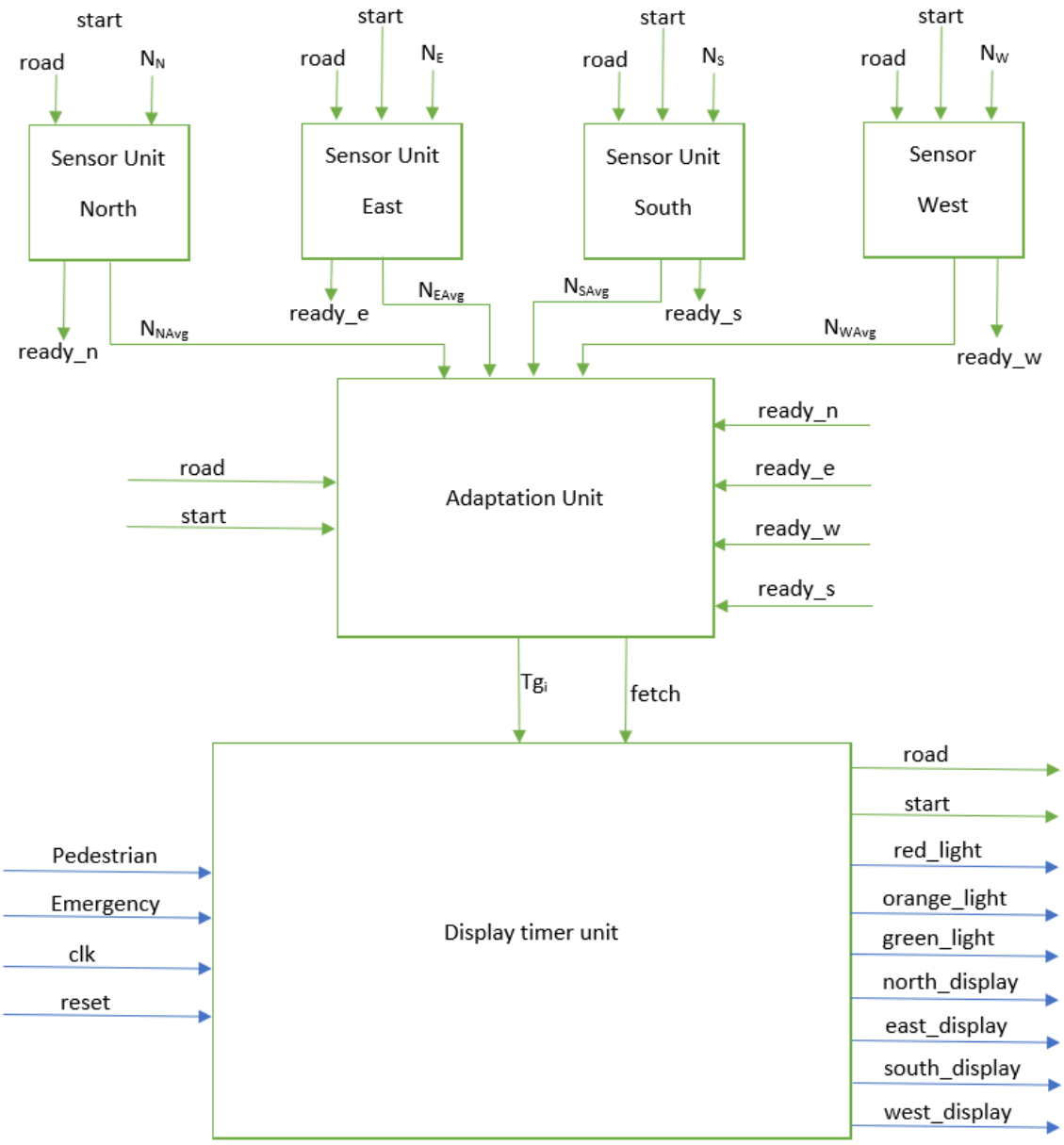

Figure 1: 4-way traffic controller architecture

1.2 Interfaces

The table shown below provides the information about the signals that are shown in Figure 1.

<table>
<tr><th>Signal Name</th><th>Signal Type</th><th>Description</th></tr>
<tr><td>Pedestrian</td><td>Input</td><td>This port will be connected to pedestrian button</td></tr>
<tr><td>Emergency</td><td>Input</td><td>This port will be connected to emergency button</td></tr>
<tr><td>clk</td><td>Input</td><td>This port will be connected to clock input</td></tr>
<tr><td>reset</td><td>Input</td><td>This port will be connected to reset button</td></tr>
<tr><td>N_n (N_N)</td><td>Input</td><td>This is the port that will get number of vehicles in north road from sensor</td></tr>
<tr><td>N_e (N_E)</td><td>Input</td><td>This is the port that will get number of vehicles in east road from sensor</td></tr>
<tr><td>N_s (N_S)</td><td>Input</td><td>This is the port that will get number of vehicles in south road from sensor</td></tr>
<tr><td>N_w (N_W)</td><td>Input</td><td>This is the port that will get number of vehicles in west road from sensor</td></tr>
<tr><td>red_light</td><td>Output</td><td>It is a 4-bit output port containing the status of red light in all four road.
<table>
<tr><td>Condition</td><td>Bit3 (East)</td><td>Bit0 (South)</td><td>Bit1 (East)</td><td>Bit0 (North)</td></tr>
<tr><td>All red</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>All red except North</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>All red except East</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>All red except South</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>All red except West</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>orange_light</td><td>Output</td><td>It is a 4-bit output port containing the status of orange light in all four road.
<table>
<tr><td>Condition</td><td>Bit3 (East)</td><td>Bit0 (South)</td><td>Bit1 (East)</td><td>Bit0 (North)</td></tr>
<tr><td>No orange</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>North orange</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>East orange</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>South orange</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>West orange</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>green_light</td><td>Output</td><td>It is a 4-bit output port containing the status of green light in all four road and it’s table will look similar to orange_light</td></tr>
<tr><td>north_display</td><td>Output</td><td>It is 12-bit output port for two 7-segnment display in north road</td></tr>
<tr><td>east_display</td><td>Output</td><td>It is 12-bit output port for two 7-segnment display in east road</td></tr>
<tr><td>south_display</td><td>Output</td><td>It is 12-bit output port for two 7-segnment display in south road</td></tr>
<tr><td>West_display</td><td>Output</td><td>It is 12-bit output port for two 7-segnment display in west road</td></tr>
</table>

<table>
<tr><td>road</td><td>Control signal</td><td>It is a used to select the sensor unit of a particular road.
<table><tr><td>Road</td><td>Condition</td></tr><tr><td>00</td><td>North road sensor is selected</td></tr><tr><td>01</td><td>East road sensor is selected</td></tr><tr><td>10</td><td>South road sensor is selected</td></tr><tr><td>11</td><td>West road sensor is selected</td></tr></table></td></tr>
<tr><td>start</td><td>Control signal</td><td>It is used to start the sensor and adaptation unit</td></tr>
<tr><td>N_navg (N_{NAvg})</td><td>Data signal</td><td>It is used to send the average number of vehicles to the adaption unit for north road</td></tr>
<tr><td>N_navg (N_{EAvg})</td><td>Data signal</td><td>It is used to send the average number of vehicles to the adaption unit for east road</td></tr>
<tr><td>N_navg (N_{SAvg})</td><td>Data signal</td><td>It is used to send the average number of vehicles to the adaption unit for south road</td></tr>
<tr><td>N_navg (N_{WAvg})</td><td>Data signal</td><td>It is used to send the average number of vehicles to the adaption unit for west road</td></tr>
<tr><td>ready_n</td><td>Data signal</td><td>It tells the adaptation unit that the average value is calculated for north road</td></tr>
<tr><td>ready_e</td><td>Data signal</td><td>It tells the adaptation unit that the average value is calculated for east road</td></tr>
<tr><td>ready_w</td><td>Data signal</td><td>It tells the adaptation unit that the average value is calculated for west road</td></tr>
<tr><td>ready_s</td><td>Data signal</td><td>It tells the adaptation unit that the average value is calculated for south road</td></tr>
<tr><td>T_gi (T_{gi})</td><td>Data signal</td><td>It send the green time for the selected road to display timer module</td></tr>
<tr><td>fetch</td><td>Data signal</td><td>It tells the display timer module that green time for a particular road is calculated.</td></tr>
</table>

2 Functional description

In this section we will discuss all the modules in detailed manner

2.1 Display timer unit

- Under no emergency, pedestrian and reset signal, display timer unit will cycle from state S0 to S7 and then back to S0. State is changed when the time for the current state is over.
- When emergency signal is given between any green signal, then that state will be changed followed by the orange light in that same road and after that all the light will go red for 10 second and then the system will go to the state next to the state just before the all red state.
- When pedestrian signal is pressed between any orange signal, then all lights red will go red after the time for current orange light is over and all red state will be active for only 10 seconds and after that the system will go to the state next to state just before the all red state.
- When the reset is pressed then all the lights will go red instantly and it will stay red for 10 seconds and after that the system will start the normal function from S0 state. This same condition is also our all red condition
- There are four different counter used for denoting the time of the lights for each road

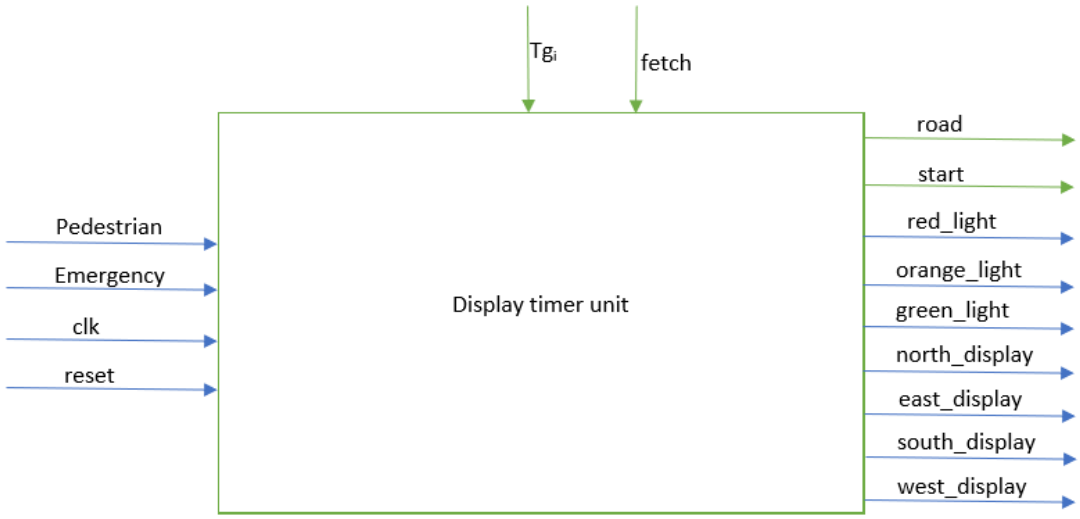

Figure 2: Block diagram of display timer unit

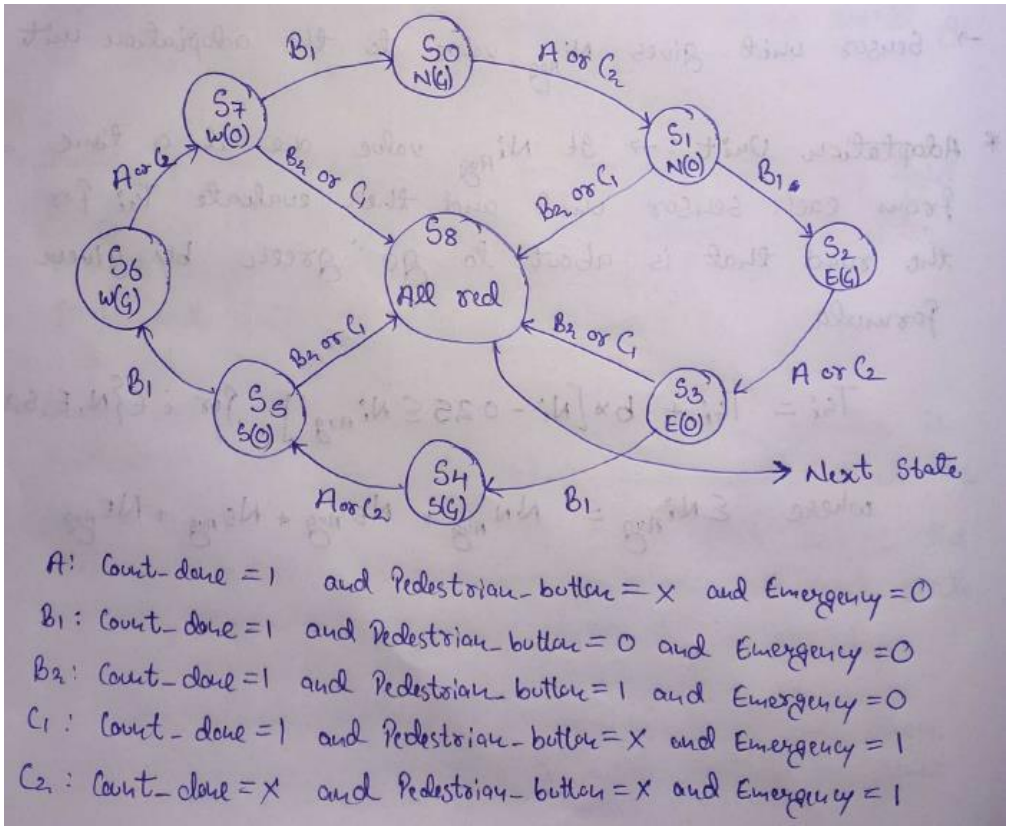

Figure 3: FSM of Display timer unit

State Table:

State	Value	Description
S0	0000	Represent North green
S1	0001	Represent North orange
S2	0010	Represent East green
S3	0011	Represent East orange
S4	0100	Represent South green
S5	0101	Represent South orange
S6	0110	Represent West green
S7	0111	Represent West orange
S8	1000	Represent All red condition

Module used in this unit:

1. **Clock Divider module:** This module is used to lower the frequency of input clock which is assumped to be 100MHz and brought down to 1Hz with the help of this module so that it is easier to create the delay having the magnitudes in seconds for the light and 7-segment display.

```
module clock_divider(clk,divided_clk);
input wire clk; //100Mhz
output reg divided_clk = 0; //1Hz
localparam count_value=49999999;
integer count=0;

always @(posedge clk)
begin
if (count==count_value)
    begin
    count<=0;
    divided_clk<=~divided_clk;
    end
else
    begin
    count<=count+1;
    divided_clk<=divided_clk;
    end
end

endmodule
```

Figure 4: Clock divider module

2. **Binary to BCD module:** This module is used to convert the countdown for lights of each road to BCD value from a binary value so that it can be easily decoded for a seven segment display.

```
module binary_to_bcd(bcd_out,binary_in);
input [7:0] binary_in;
output reg [11:0] bcd_out;
reg [3:0] i;

always @(binary_in)
begin
    bcd_out=0;
    for(i=0;i<8;i=i+1)
    begin
        bcd_out={bcd_out[10:0],binary_in[7-i]};

        if(i<7 && bcd_out[3:0]>4)
        bcd_out[3:0]=bcd_out[3:0]+3;
        if(i<7 && bcd_out[7:4]>4)
        bcd_out[7:4]=bcd_out[7:4]+3;
        if(i<7 && bcd_out[11:8]>4)
        bcd_out[11:8]=bcd_out[11:8]+3;
    end
end

endmodule
```

Figure 5: Binary to BCD module

3. **BCD to 7-segment module**: This module is used to display the count in 7-segment display of each road from the equivalent BCD value of the count corresponding to that road.

```
module bcd_to_7segment(display_out,bcd_in,clk);
input clk;
input [3:0] bcd_in;
output reg [6:0] display_out; //abcdefg

always @(posedge clk)
begin
    case(bcd_in)
    4'b0000: display_out = 7'b1111110;
    4'b0001: display_out = 7'b0110000;
    4'b0010: display_out = 7'b1101101;
    4'b0011: display_out = 7'b1111001;
    4'b0100: display_out = 7'b0110011;
    4'b0101: display_out = 7'b1011011;
    4'b0110: display_out = 7'b1011111;
    4'b0111: display_out = 7'b1110000;
    4'b1000: display_out = 7'b1111111;
    4'b1001: display_out = 7'b1111011;
    default: display_out = 7'b1111110;
    endcase
end
endmodule
```

Figure 6: BCD to 7-segment module

3 Testbench

Project Summary | display_timer.v | bcd_to_7segment.v | binary_to_bcd.v | test_bench.v

C:/Users/preea/OneDrive/Documents/Vivado/Traffic_light_controller/Traffic_light_controller.srcs/sim_1/new/test_bench.v

```
17  // Revision 0.01 - File Created
18  // Additional Comments:
19  //
20  //////////////////////////////////////////////////////////////////////////////////
21
22
23  module test_bench();
24  reg clk,reset,emg,ped;
25  wire [3:0] red, orange, green;
26  wire [11:0] display_north,display_east,display_south,display_west;
27
28  display_timer abc(red,orange,green,display_north,display_east,display_south,display_west,ped,emg,clk,reset);
29
30  initial
31  begin
32  clk=1'b0;
33  reset=1'b0;
34  emg=0;
35  ped=0;
36  #150 reset=1'b1;
37  #1 reset=1'b0;
38  #14 emg=1'b1;
39  #1 emg=1'b0;
40  #112 ped=1'b1;
41  #1 ped=1'b0;
42  #200 $finish;
43
44  end
45
46  always #0.5  clk=~clk;
47  endmodule
48
```

4 Bugs known at submission date

- Time adaptation part of the display timer unit is not been executed as it was not properly implemented and hence the timing for lights are already defined in the code and it stays same throughout the simulation
- Clock Divider module is working properly (checked separately) but it is still not used while simulation because the simulation becomes too slow when we clock divider is used but still it can be implement at the end when all the things are perfect.