ELP 831

IEC LAB – 1

Traffic Light Controller

Report

Submitted by:
Maya Khangembam
EEN212020

1 Introduction

1.1 Referenced documents

HDL_assignment.pdf

1.2 Design project name

Traffic Light Controller

1.3 People involved in the block

Maya Khangembam (EEN212020)

2 Function

2.1 Brief overview

The project is on the given assignment. It implements a Traffic Light Controller which senses the number of vehicles at the road and dynamically updates the duration of green and red lights at all roads. The outermost structure of the system is as shown in Fig 1.

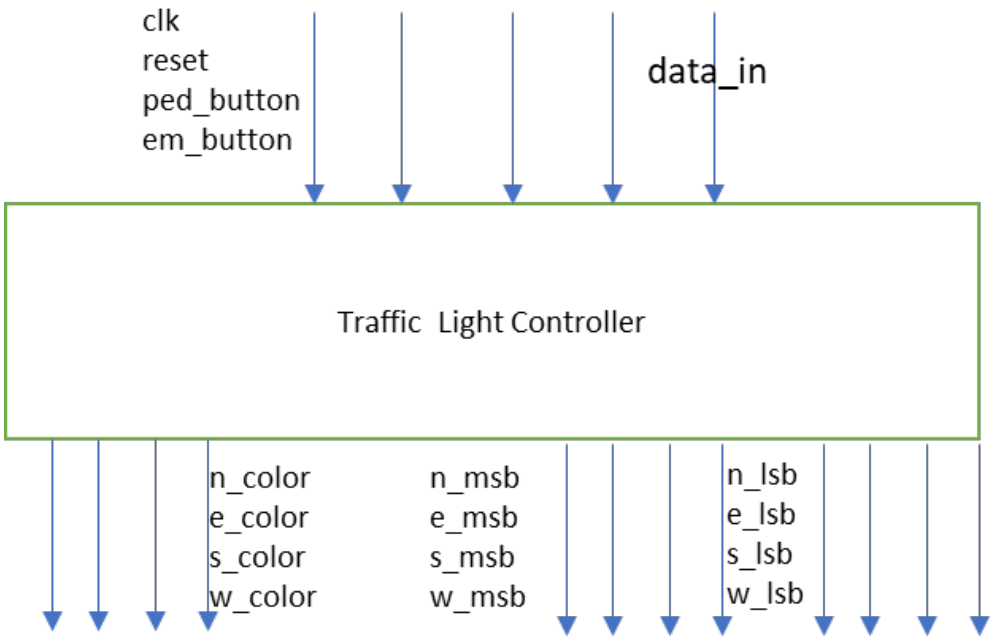

Fig 1

2.2 Interfaces

Overall			
Signal Name	**Bus Size**	**I/O**	**Description**
reset	1	input	Reset signal input
clk	1	input	Clock
data_in	8	input	Number of vehicles sensed at current cycle
ped_button	1	input	Pedestrian signal
em_button	1	input	Emergency signal
n_color	56	output	Signal color at north road
e_color	56	output	Signal color at east road
s_color	56	output	Signal color at south road
w_color	56	output	Signal color at west road
n_msb	7	output	Seven segment display of msb
n_lsb	7	output	Seven segment display of lsb
e_msb	7	output	Seven segment display of msb
e_lsb	7	output	Seven segment display of lsb
s_msb	7	output	Seven segment display of msb
s_lsb	7	output	Seven segment display of lsb
w_msb	7	output	Seven segment display of msb
w_lsb	7	output	Seven segment display of lsb

Signal Name	**Bus Size**	**I/O**	**Description**
North Sensor			
reset	1	input	Reset signal input
next_road	2	input	Next road value
data_in	8	input	Number of vehicles sensed at current cycle
avg	8	output	Average at north road over last 64 cycles
East Sensor			
reset	1	input	Reset signal input
next_road	2	input	Next road value
data_in	8	input	Number of vehicles sensed at current cycle
avg	8	output	Average at east road over last 64 cycles

South Sensor			
reset	1	input	Reset signal input
next_road	2	input	Next road value
data_in	8	input	Number of vehicles sensed at current cycle
avg	8	output	Average at south road over last 64 cycles
West Sensor			
reset	1	input	Reset signal input
next_road	2	input	Next road value
data_in	8	input	Number of vehicles sensed at current cycle
avg	8	output	Average at west road over last 64 cycles
Adaptation Unit			
reset	1	input	Reset signal input
next_road	2	input	Next road value
N_n	8	input	Ni value at north road from north sensor
N_e	8	input	Ni value at east road from north sensor
N_s	8	input	Ni value at south road from north sensor
N_w	8	input	Ni value at west road from north sensor
TGn	8	output	Updated value of TGin
TGe	8	output	Updated value of TGie
TGs	8	output	Updated value of TGis
TGw	8	output	Updated value of TGiw

Signal Name	Bus Size	I/O	Description
Display Unit			
clk	1	input	Clock signal input
reset	1	input	Reset signal input
ped_button	1	input	Pedestrian signal
em_button	1	input	Emergency signal

TGi0	8	input	Green light duration at north road
TGi1	8	input	Green light duration at east road
TGi2	8	input	Green light duration at south road
TGi3	8	input	Green light duration at west road
TG0	8	output	Rounded value of TGi0
TG1	8	output	Rounded value of TGi1
TG2	8	output	Rounded value of TGi2
TG3	8	output	Rounded value of TGi3
N_color	56	output	Signal color at North road
E_color	56	output	Signal color at East road
S_color	56	output	Signal color at South road
W_color	56	output	Signal color at West road
n_count	8	output	Timer count at North road
e_count	8	output	Timer count at East road
s_count	8	output	Timer count at South road
w_count	8	output	Timer count at West road
t_count	8	output	Count of complete cycle
n_msb	7	output	Seven segment display of msb
n_lsb	7	output	Seven segment display of lsb
e_msb	7	output	Seven segment display of msb
e_lsb	7	output	Seven segment display of lsb
s_msb	7	output	Seven segment display of msb
s_lsb	7	output	Seven segment display of lsb
w_msb	7	output	Seven segment display of msb
w_lsb	7	output	Seven segment display of lsb

2.3 Architecture

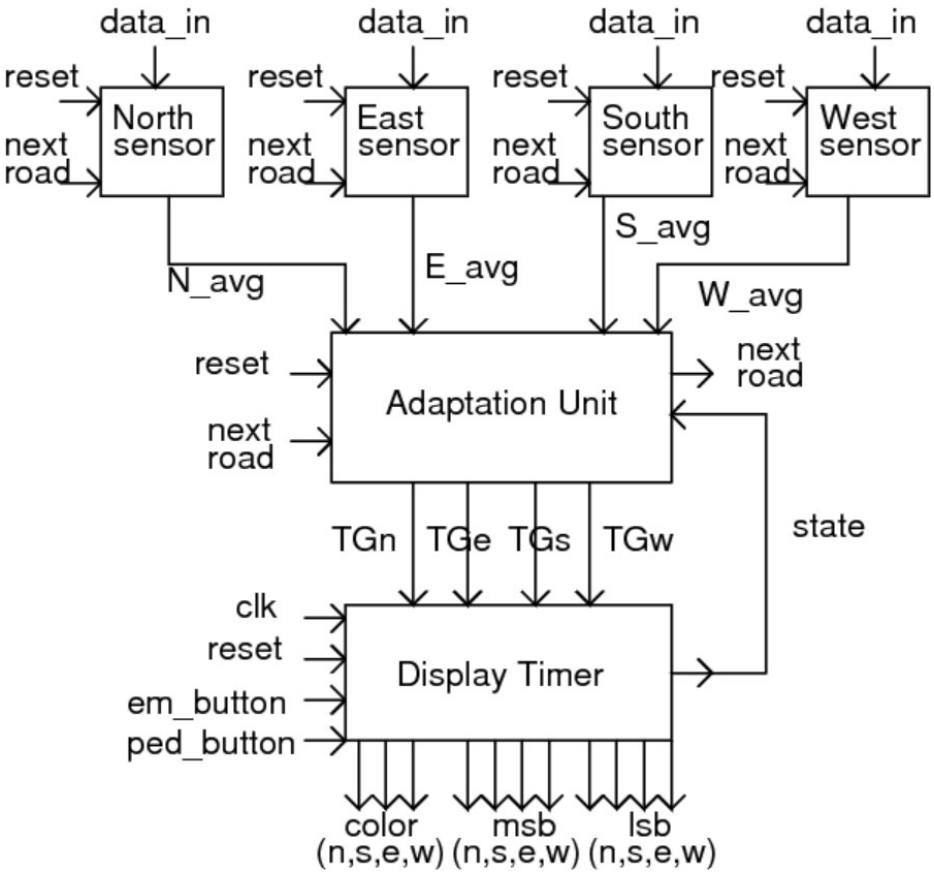

Fig 2

2.4 Detailed functional description

The project implements the working of a traffic light controller. The system consists of 3 main units – the sensors, the adaptation unit and the display unit.

The sensor unit consists of 4 sensor modules, one for each road. The four sensor modules are **north_sensor, east_sensor, west_sensor**, and **south_sensor**. Each sensor takes in the value of the next road and new vehicle count at the corresponding road and outputs an average of the number of vehicles at that road over the past 64 cycles. It employs one shift register to store and update values over the 64 cycles and another to perform the division operation to obtain the average.

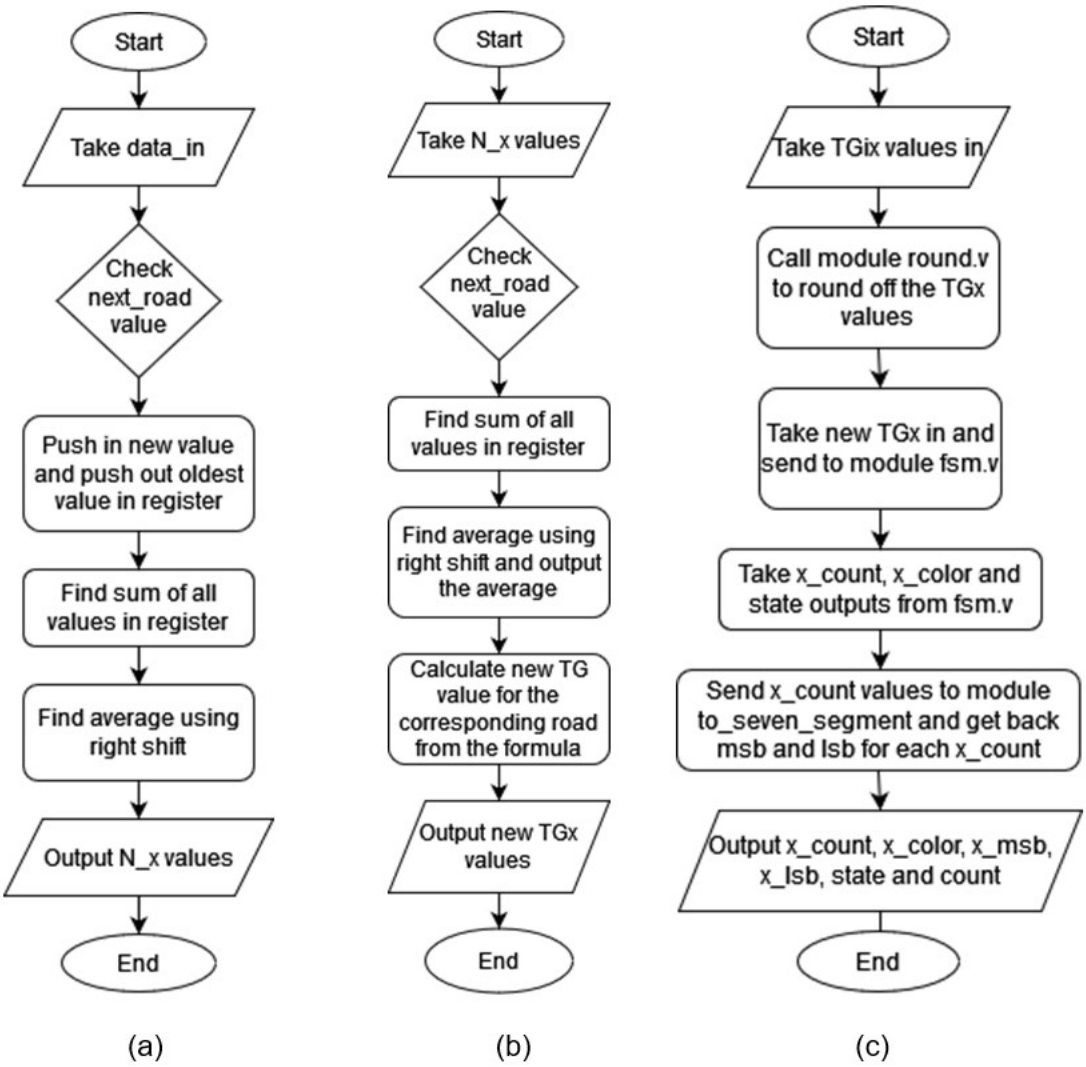

Fig 3. The flowcharts represent the working of (a) Sensor unit, (b) Adaptation unit and (c) Display unit

The adaptation unit is named as **Adaptation_unit**. It takes in the average vehicle count at each road from the sensors and the value of the next road and outputs the updated green light duration value for the corresponding road. The formula used is as follows:

$$N\_average = \frac{N\_n + N\_e + N\_s + N\_w}{4}$$

$$TGx = TGx + b \times (N\_x - N\_average)$$

Tcyc = total duration of a cycle = **200**
TGx = duration of green signal for road x, for x in {N, E, S, W}
TRx = duration of red signal for road x, for x in {N, E, S, W}
TO = duration of orange signal for each road = **10**

The Display unit consists of 5 modules:

- **Display (Top Module):** connects and calls all other modules. Takes TGx values, emergency signal and pedestrian signal and outputs color and count at each road, current state, count at current road and seven segment values for the msb and lsb for the count at each road.

- **fsm_time:** Takes in the different TGx values as well as emergency and pedestrian signals and outputs the color and count at each road as well as the current state and count at current road

- **rounder:** Takes in a value and rounds off and returns the value

- **to_seven_segment:** Takes value and divides it into it's msb and lsb and sends them to the seven_segment module

- **seven_segment:** Takes value and outputs seven segment values

The working of the entire traffic light system is as follows:

- **Regular Operation:** Under regular operation, the system operates as a finite state machine and has a fixed cycle of states as defined in the state table below. Each green state lasts for TGx value where x is N, E, S, W for the north, east, south, and west roads respectively. Each orange state lasts for TO count (TO = 10). The duration of the red states depends on the formula:

$$TRx = Tcycle - TGx - TO$$

$$Tcycle = TGN + TGE + TGS + TGW + 4 \times TO$$

The Tcycle value taken in the implemented system is 200.

State	North	East	South	West
S0	Green	Red	Red	Red
S1	Orange	Red	Red	Red
S2	Red	Green	Red	Red
S3	Red	Orange	Red	Red
S4	Red	Red	Green	Red
S5	Red	Red	Orange	Red
S6	Red	Red	Red	Green
S7	Red	Red	Red	Orange

Fig 4

The states cycle from S0 to S7 and back to S0.

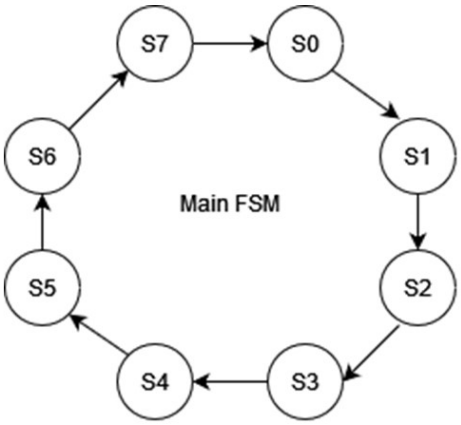

Fig 5

- **Emergency:** When the emergency button is pressed, if the current road is under a green state, then it is terminated immediately and the current road enters an orange state. If the current road is under orange state, the orange state is allowed to continue till the end. After the orange stage ends, all roads are forced to enter a red state for 10s duration. At the end of this all-red state, the system resumes to the main FSM again, starting with the next road in a green state. The flowchart below explains the working of the system when the emergency button is pressed.

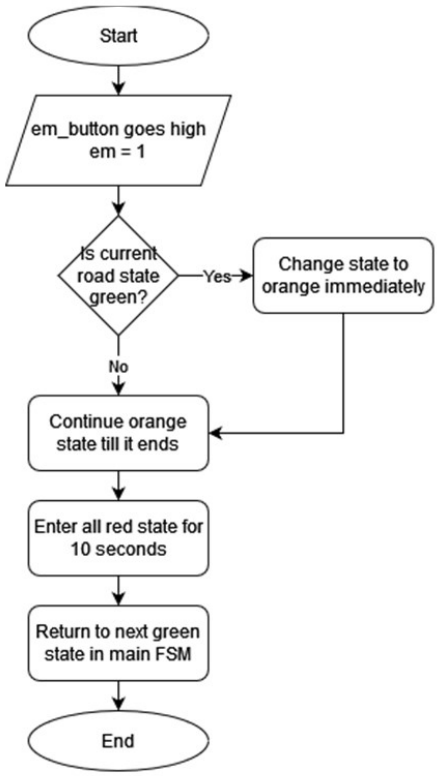

Fig 6

- **Pedestrian:** When the pedestrian button is pressed, the operation is same as under the emergency case, except that instead of immediately entering orange state, the green state is allowed to complete its duration after which the orange and all-red states follow. The flowchart for the same has been included.

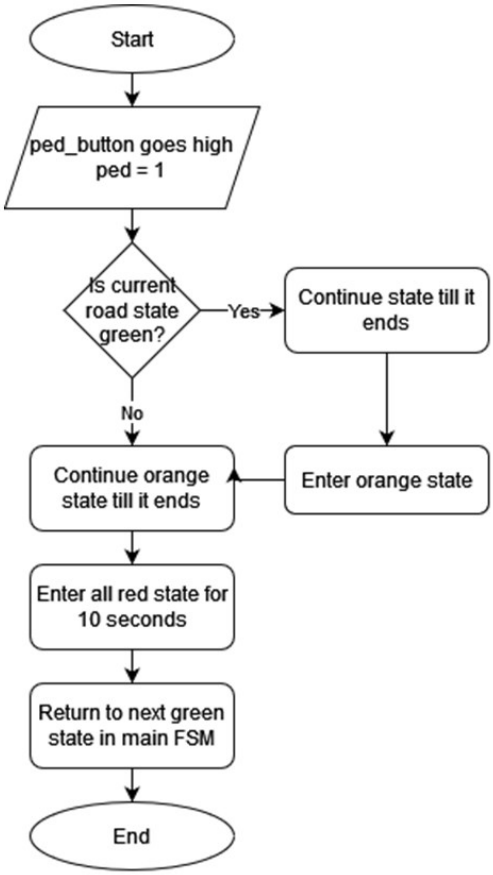

Fig 7

4 Verification Strategy

4.1 Objectives

The objective is to check whether the system works as intended under different cases such as normal operation, when emergency button is pressed, when the pedestrian button is pressed and when both the emergency and pedestrian buttons are pressed at the same time.

4.2 Tools and Version

Tool used is Xilinx Vivado 2020.1.

4.3 Checking mechanisms

Checking of functionality has been done manually by verifying the waveforms produced in Vivado to see if the outputs are as desired.

5 Functional Checklist

Unit name	Status	Comments
Sensors	OK	4 sensor modules - north_senso:r.v - east_sensor.v - south_sensor.v - west_sensor.v
Adaptation Unit	OK	One module – Adaptation_Unit.v
Display Unit	OK	4 modules: - Display_Timer.v – calls the other modules in this unit - fsm.v - round.v - to_seven_seg - seven_seg
Main Unit	OK	Main_Traffic_Module.v – calls all the other units

6 Testbench

There are in total 4 testbenches:

Testbench	Function	DUT
Testbench.v	Tests the normal operation of the entire traffic controller system	Traffic_Main_Module.v

em_ped_tb.v	Tests the operation of the entire traffic controller system when both emergency and pedestrian button are high	Traffic_Main_Module.v
Display_tb.v	Tests operation of the Display unit under normal operation, only emergency button high and only pedestrian button high	Display_Timer
Adaptation_tb.v	Tests operation of the Adaptation unit with the sensor unit	adaptation_sensor_unit.v

7 Tests Specification

Module: Traffic_Main_Module.v

Tested cases in the overall traffic controller system are:

- **Normal FSM operation:** To check if the system works as intended under normal operation, i.e, when both emergency and pedestrian buttons are low.

```
initial begin
    clk = 0;
    em_button = 0;
    ped_button = 0;
    reset = 0;#2
    data_in = 2;#2
    reset = 1; #50
    data_in = 10; #50
    data_in = 0; #50
    data_in = 5; #50
    data_in = 20; #30
```

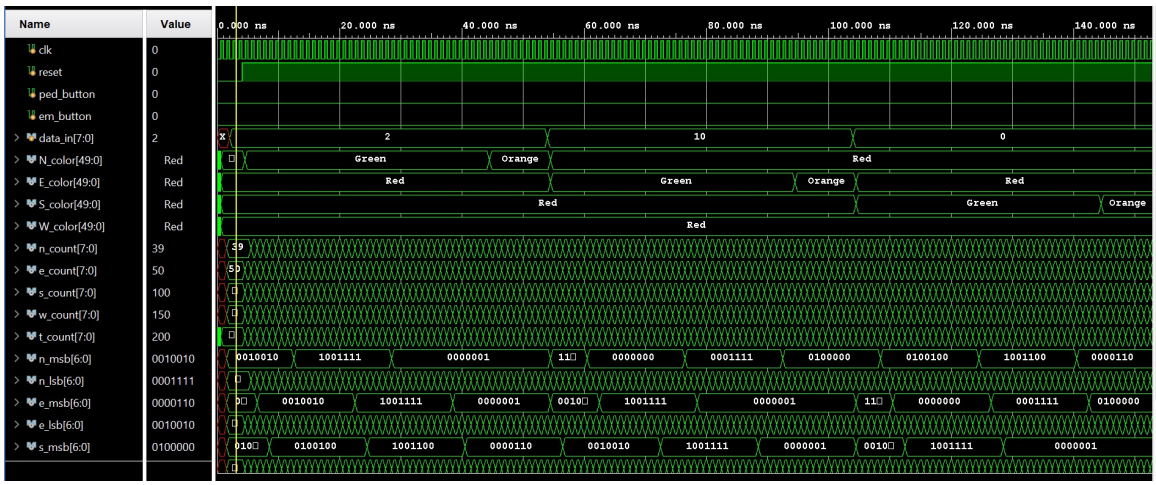

Fig 8

- **Emergency button pressed:** When only emergency button goes high.

  The emergency button goes high for one clock cycle (clock period being 1 here).

```
em_button = 1;#1
em_button = 0;#100
```

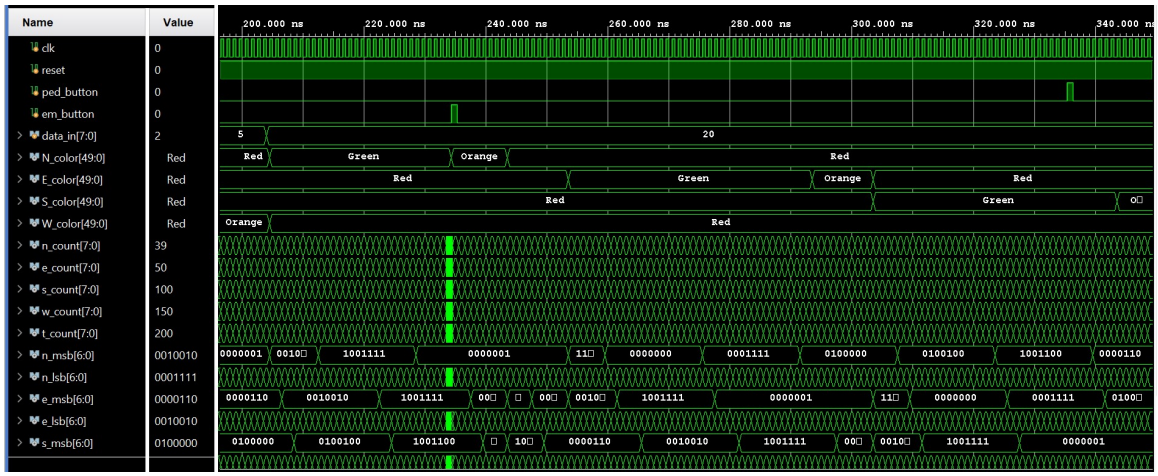

Fig 9

- **Pedestrian button pressed:** When only pedestrian button goes high.

  The pedestrian button goes high for one clock cycle (clock period being 1 here).

```
ped_button = 1;#1
ped_button = 0; #100
```

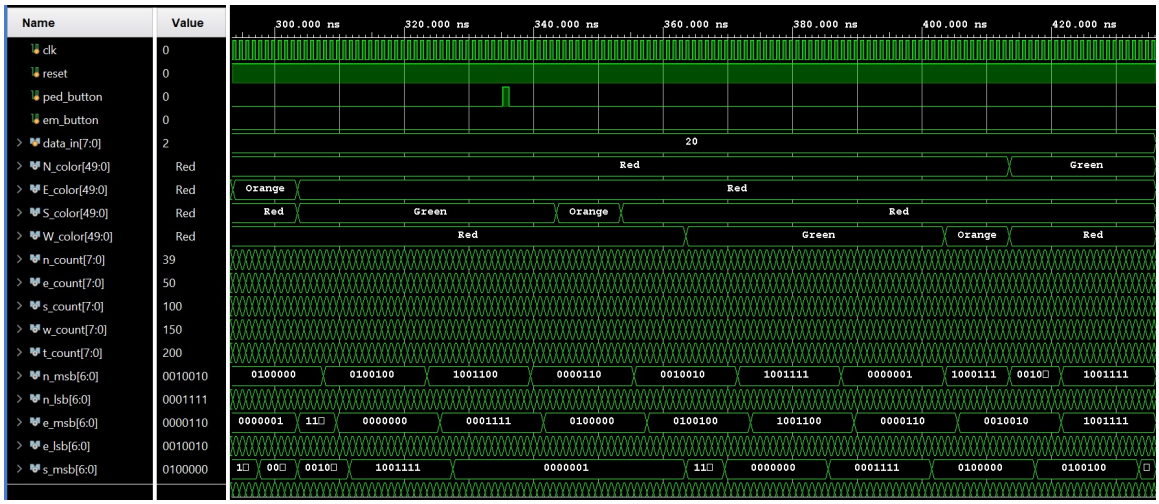

Fig 10

- **When both emergency and pedestrian are pressed at the same time:** When both buttons go high at the same time, the priority is given to emergency state. The same can be seen in the waveform included.

```
em_button = 1;
ped_button = 1;#1
em_button = 0;
ped_button = 0; #100
```

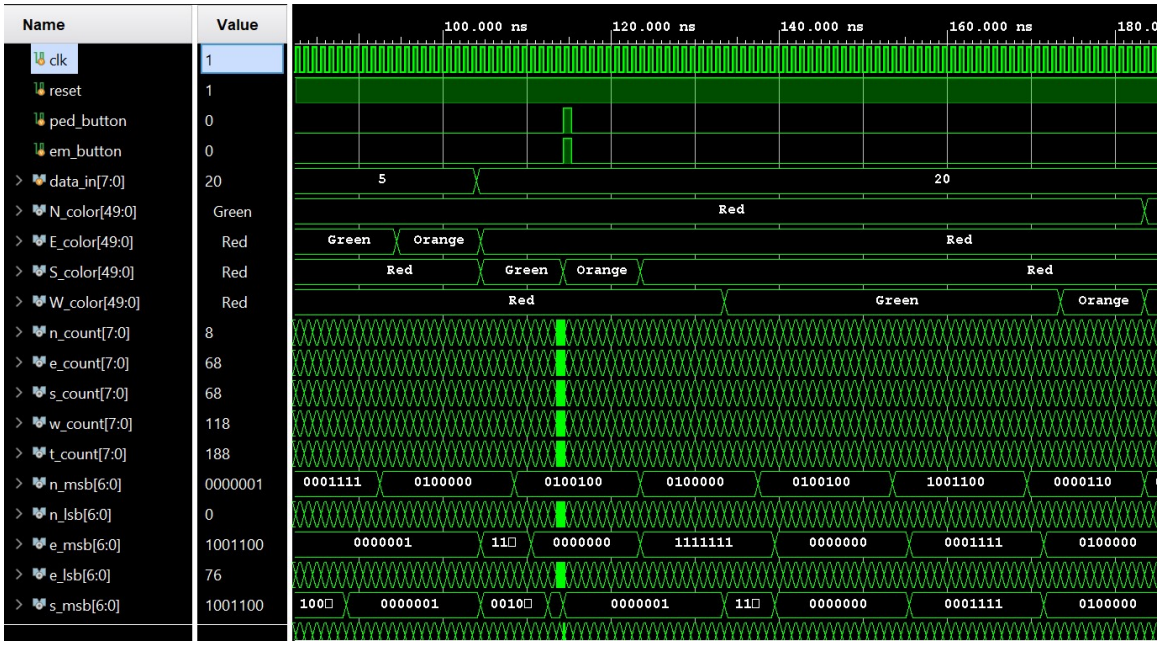

Fig 11

8 Bugs known at submission date

None. The system is working as intended under all tested cases.