

## CHAPTER 4

### RESULTS AND DISCUSSION

#### 4.1 Introduction

In this chapter, all of the programs, circuits and waveforms are shown. The result for improved traffic signals controller will be detailed in this chapter. The outputs of simulation result will also be shown in this chapter.

#### 4.2 Result of Combinational Logic

The combinational logic has three parts, which are state decoder, output logic and trigger logic. Based on the block diagram of traffic light controller in previous chapter (Figure 3.1) and the interface pins shown in Figure 3.8 (Block Diagram the Combinational Logic) and also refer the Table 3.1, Table 3.2 and Table 3.3 to synthesizable Verilog Code is written and circuit gate logic was design.

##### 4.2.1 Result for State Decoder

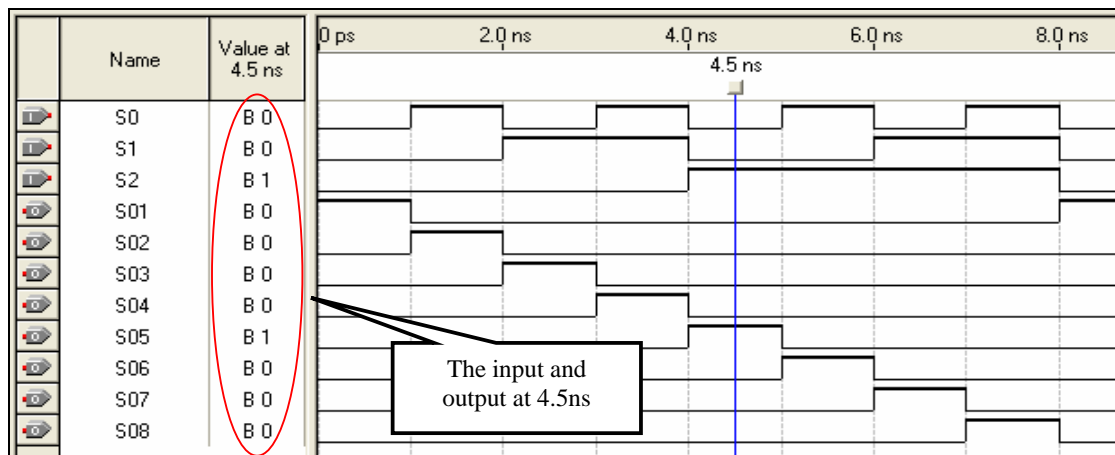
The Verilog code for “state\_decoder” is shown below and the complete timing diagram is shown in Figure 4.0.

```
Module state_decoder (S0,S1,S2,S01,S02,S03,S04,S05,S06,S07,S08);  
//Declaration port input and output  
input S0,S1,S2;      // State decoder input signals
```

```
output S01,S02,S03,S04,S05,S06,S07,S08; // State decoder outputs
```

```
assign S01 = ~S2 & ~S1 & ~S0; // State output1
assign S02 = ~S2 & ~S1 & S0; // State output2
assign S03 = ~S2 & S1 & ~S0; // State output3
assign S04 = ~S2 & S1 & S0; // State output4
assign S05 = S2 & ~S1 & ~S0; // State output5
assign S06 = S2 & ~S1 & S0; // State output6
assign S07 = S2 & S1 & ~S0; // State output7
assign S08 = S2 & S1 & S0; // State output8
```

```
endmodule // End port declaration
```



**Figure 4.0:** Diagram showing simulation result of module state\_decoder

Referring Figure 4.0, S0 through S2 is an input and S01 through S08 is a state decoder output. The outputs of state decoder are become from the Boolean expressions and the truth table for the state decoder logic is shown in Table 3.1 in previous chapter.

At the first condition, the state input S2, S1 and S0 is equal to '000' or LOW for all state input condition, the first output (S01) is equal to '1' or HIGH condition and the others output is LOW condition. The second output (S02) is

HIGH when the condition inputs are  $\overline{S_2} \overline{S_1} S_0$  or '001' ( $S_2=0, S_1=0$  and  $S_0=1$ ). The each output of state decoder has a different input condition. The third output state (S03) is HIGH when the state input is '010' ( $S_2=0, S_1=1$  and  $S_0=0$ ). The fourth state (S04) output is HIGH when the input of state is '011' and the state output of S05 is HIGH when the input is '100'. The S06, S07 and S08 are high when the state inputs are '101', '110' and '111'.

In this part, the output of the state decoder is referred to the state input. When one of state output is HIGH, the others output is LOW. At the same time, only one state output can HIGH. This decoder can determine which eight states the systems are in.

#### 4.2.2 Result for Output Logic

Verilog code and simulation result for output logic is shown in Figure 4.1. Referring the coding and waveform in Figure 4.1, the certain output can HIGH more than one pulse.

```

Module output_logic (S01,S02,S03,S04,S05,S06,S07,S08,G1,Y1,R1,
GL1,G2,Y2,R2,GL2,G3,Y3,R3,GL3,G4,Y4,R4,GL4 ); // Declaration of
input and output output_logic module
input S01,S02,S03,S04,S05,S06,S07,S08; // Declaration input
output G1,Y1,R1,GL1,G2,Y2,R2,GL2,G3,Y3,R3,GL3,G4,Y4,R4,GL4;
// Declaration output of output_ logic

    assign GL1 = S01+ S02 + S03+ S04; // Road1 Green (left-sign)
    assign G1 = S01; // Road1 Green
    assign Y1 = S02; // Road1 Yellow
    assign R1 = S03 + S04 + S05 +S06 + S07 + S08; // Road1 Red
    assign GL2 = S03 + S04 + S05 + S06; // Road2 Green (left-sign)
    assign G2 = S03; Road2 Green
    assign Y2 = S04; // Road2 Yellow
    assign R2 = S01 + S02 + S05 + S06 + S07 + S08; // Road2 Red

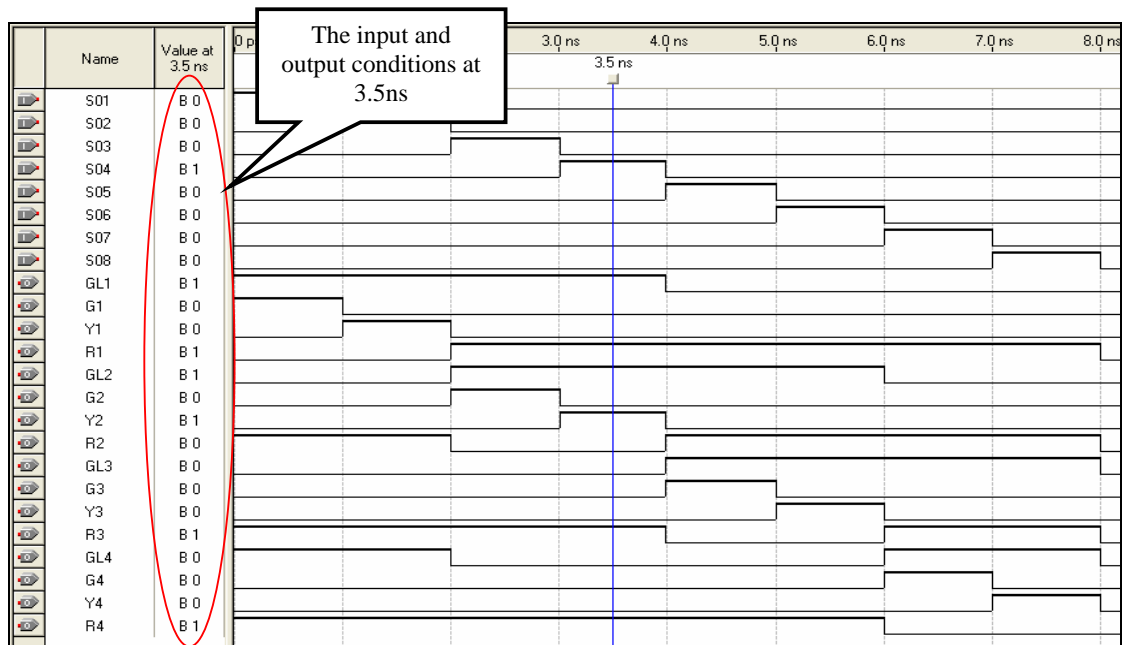
```

```

assign GL3 = S05 + S06 + S07 + S08; // Road3 Green (left-sign)
assign G3 = S05; // Road3 Green
assign Y3 = S06; // Road3 Yellow
assign R3 = S01 + S02 + S03 + S04 + S07 + S08; // Road3 Red
assign GL4 = S01 + S02 + S07 + S08; // Road4 Green (left-sign)
assign G4 = S07; // Road4 Green
assign Y4 = S08; // Road4 Yellow
assign R4 = S01 + S02 + S03 + S04 + S05 + S06; // Road4 Red

```

**endmodule**



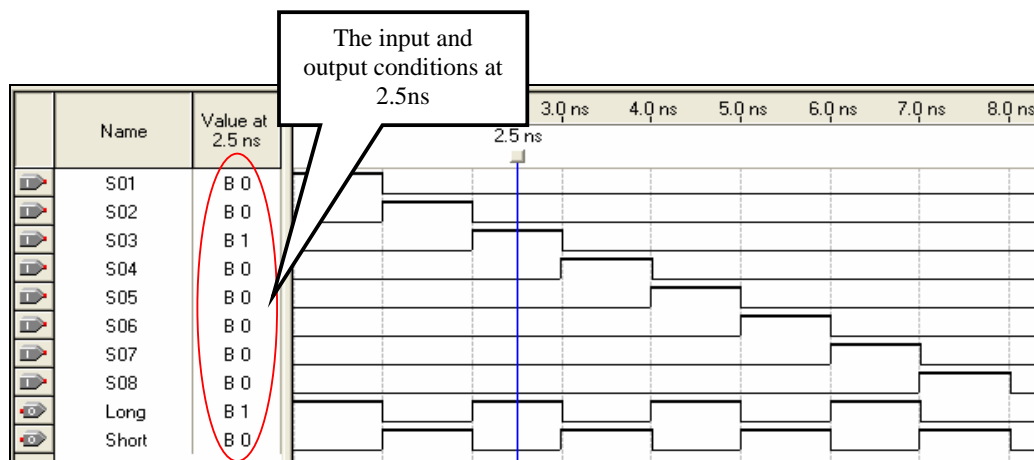
**Figure 4.1:** Diagram showing simulation result of module output\_logic

According to Figure 4.1, the inputs are labeled S01 through S08. This input is from the state decoder output. The outputs are labeled GL1, GL2, GL3 and GL4 for left-sign at Road1, Road2, Road3 and Road4. The G1 through G4 is a green lights output, Y1 through Y4 is for yellow lights and R1 through R4 is red lights. The main road is Road1 & Road3 and side street is Road2 & Road4.

The condition for left-sign (GL1, GL2, GL3 and GL4) are green is four states or four pulse. In this simulation, the one pulse is for one transition state. The outputs from state decoder are used as an input for the output logic. Traffic signals can HIGH more than one state at the same time for the certain output signals. For example, at pulse 3.5ns, yellow light is HIGH at Road2, the GL1 (Road1) is green, R3 (Road3) and R4 (Road4) also HIGH condition is Red colour. For more information, refer Figure 4.1, assume the signal lights is ‘on’ when the clock pulse is HIGH and the signal lights is ‘off’ when the clock pulse is LOW.

#### 4.2.3 Result for Trigger Logic

The trigger logic is uses the decoded states to produce signals for properly triggering the long timer and short timer. Figure 4.2 shows the simulation result for the trigger logic and a coding for timing trigger.



**Figure 4.2:** Diagram showing simulation result of module trigger\_logic

The trigger logic produces two outputs. The long output is LOW-to-HIGH transitions that trigger 20s timing circuit when the system goes into the states first (000), third (010), fifth (100) and seventh (110). The long timer is for green and red signal lights. The short output is a LOW-to-HIGH transmission that short triggers the 4s timing circuits when the system goes into the states

second (001), fourth (011), sixth (101) and eighth (111). The short timer is for yellow light signal.

```
module trigger_logic (S01,S02,S03,S04,S05,S06,S07,S08,Short,Long);  
// Declaration input and output  
input S01,S02,S03,S04,S05,S06,S07,S08; // Input declaration  
output Short,Long; // Output declaration  
  
    assign Long = (S01 + S03 + S05 + S07); // Long timer output  
    assign Short = (S02 + S04 + S06 + S08); // Short timer output  
  
endmodule
```

For example, at 2.5ns, the third state (010) is HIGH pulse, so the output is long trigger. The long or short trigger is depends the output from the state decoder. At the same time, only one output can 'on', long or short when the different input condition is set.

#### 4.2.4 Result of Complete Combinational Logic

Figure 4.3 is shown the timing diagram result for the complete combinational logic. These combinational results are combining of three parts, which are 'state\_decoder module', 'output\_logic module' and 'trigger\_logic module'. Referring Figure 4.3, S0 through S2 is an input and GL1 through R4 is an output for the complete combinational logic. A wire is uses to connect the output from 'state\_decoder module' to input of 'output\_logic module'. The output from state decoder also connected to the timing trigger. Below this is shown the complete of combinational logic code.

```
module combinational_logic ( S0,S1,S2,GL1,G1,Y1,R1,GL2,G2,Y2,R2,  
GL3, G3, Y3, R3, GL4, G4, Y4, R4, Long, Short); // Module delaration  
input S0,S1,S2; // Input declaration from state decoder  
output GL1,G1,Y1,R1,GL2,G2,Y2,R2,GL3,G3,Y3,R3,GL4,G4,Y4,R4;  
// Declaration output signals
```

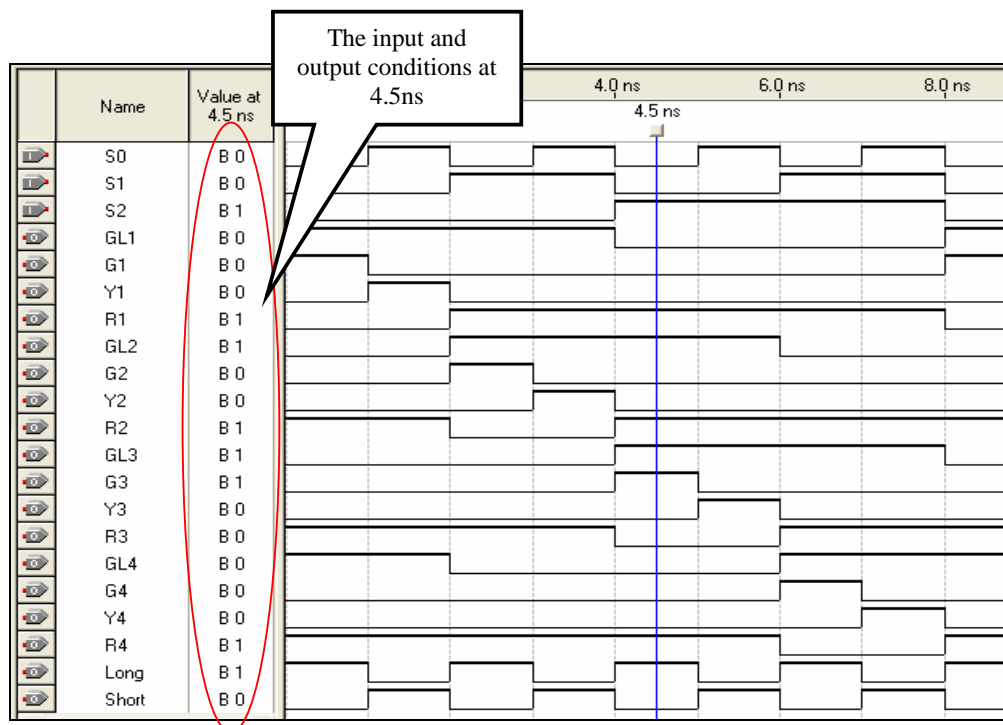
```

output Long, Short; // Output timing
wire w1,w2,w3,w4,w5,w6,w7,w8; // to connect the module

state_decoder N1 (S0,S1,S2,w1,w2,w3,w4,w5,w6,w7,w8);
// State decoder module
output_logic N2 (w1,w2,w3,w4,w5,w6,w7,w8,G1,Y1,R1,GL1,G2,Y2,
R2, GL2, G3, Y3, R3, GL3, G4, Y4, R4, GL4); // Output logic module
trigger_logic N3(w1,w2,w3,w4,w5,w6,w7,w8,Short,Long);
// Trigger module for long and short timer

endmodule

```



**Figure 4.3:** Diagram showing simulation result of module combinational\_logic

Referring Figure 4.3, at 4.5ns, the input conditions is  $100(\overline{S2} \overline{S1} S0)$ , the Long timer, R1 (Red light for road1), GL2 (Green for sign-left at road2), G3 (Green light for road3) and R3 (Red light for road4) is a HIGH condition. That

means the traffic signals and long trigger is 'on' on this time. The condition is 'on' when the pulse is HIGH ('1') and the condition is 'off' when the pulse is LOW ('0'). A result simulation must be same with the table in previous chapter (Table 3.2 and Table 3.3).

### 4.3 Result of Sequential Logic

The sequential logic consists of two parts, input logic and three D-flipflop to implement the 3-bit counter. The output from input logic is providing the D input to the flipflop and the counter is clocked by the 10 kHz clock from oscillator. The output from D-flipflop goes to the input logic.

#### 4.3.1 Result for D-flipflop module

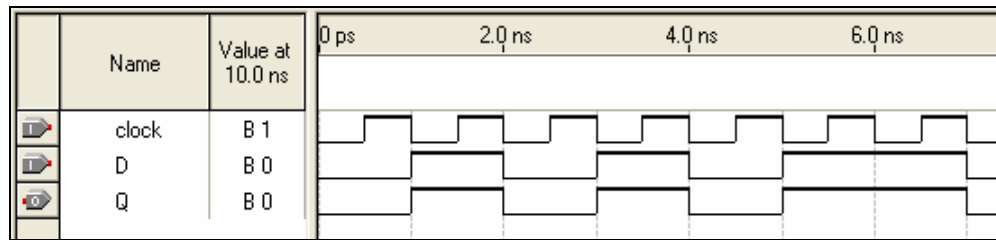
Figure 4.4 shows the simulation result for D-flipflop. The output for D-flipflop is equal to the input. The input is equal to output when the pulse sees the nedgedge clock.

```
module D_flipflop (clock, D, Q); // Module declaration – input & output
input D; // Input of D flip-flop
input clock; // Clock input
output Q; // Output of D flip-flop
reg Q; // Register output

always @ (negedge clock) // high went see nededge clock
    Q <=D;

endmodule
```





**Figure 4.4:** Diagram showing simulation result of module D-flipflop

#### 4.3.2 Result for Input Logic

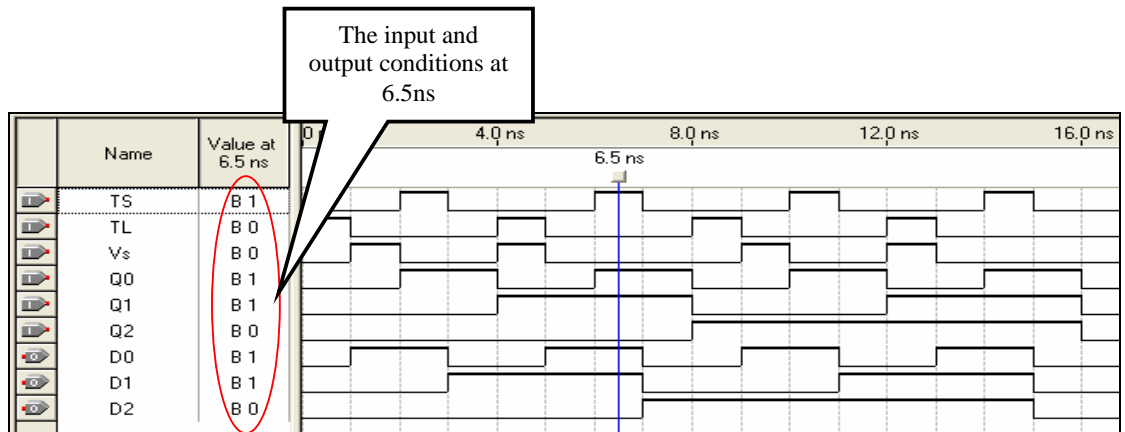
Simulation result for input logic is shown in Figure 4.5. The input logic is expressed from the state machine. State machine diagrams are shown in Figure 3.8 and also refer the truth table in Table 3.3. According to Figure 4.5, TS, TL, Vs, Q0, Q1 and Q2 are an input for the input logic module and D0, D1 and D2 is an output. TS and TL are the output from timing circuits and Q0 through Q2 is from Dual flip-flop. The output from D-flip-flop goes into the Q0 through Q2 input.

```

module input_logic (TS,TL,Vs,Q0,Q1,Q2,D0,D1,D2);
//Module declaration
input TS,TL,Vs,Q0,Q1,Q2; // input declaration
output D0,D1,D2; // output declaration

assign D0
    = ((Q0 & TS) | (~Q1 & ~Q0 & ~TL & Vs) | (Q1 & ~Q0 & ~TL) |
      (Q1 & ~Q0 & ~Vs)); // output1
assign D1
    = ((Q0 & (Q1 ~^ TS)) | (Q1 & ~Q0 & TL & Vs) | (Q1 & ~Q0 &
      ~TL) | (Q1 & ~Q0 & ~Vs));
assign D2
    = ((Q0 & ~TS) & (Q2 ^ Q1) | (Q2 & Q0 & TS) | (Q2 & ~Q0) &
      (Q1^ TL) | (Q2 & ~Q0 & ~Vs) | (Q2 & ~Q0 & Vs) & (Q1 ~^
      TL));
endmodule

```



**Figure 4.5:** Diagram showing simulation result of module input\_logic

Referring Figure 4.5, at 6.5ns the short timer is ‘on’, the long timer is ‘off’, no vehicle detection and the state for Q0 through Q2 is ‘011’. The output D0 and D1 is HIGH went timer short is ‘on’ and the input Q1 and Q0 also high.

#### 4.3.3 Result of Complete Sequential Logic

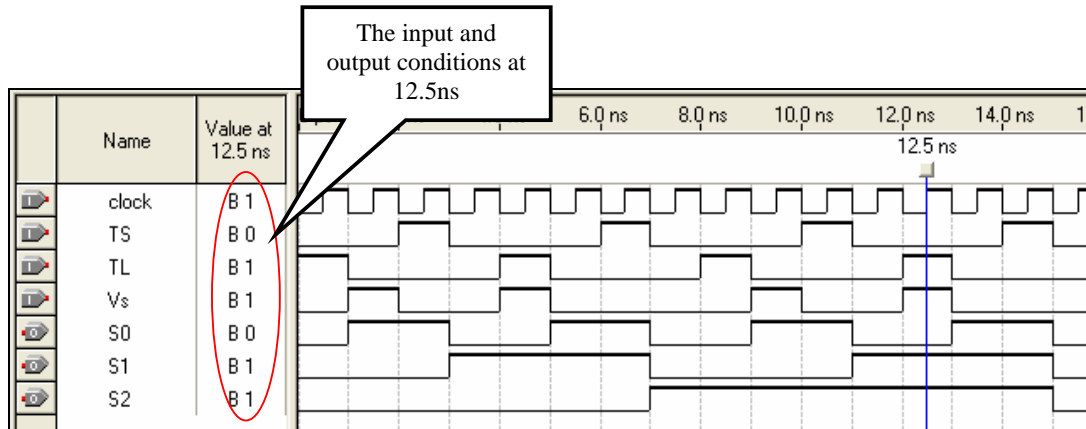
The result simulation complete sequence logic is shown in Figure 4.6. The output of this part goes to the state decoder and the coding for the sequential logic is shown below:

```

module sequential_logic (clock,TS,TL,Vs,S0,S1,S2 );
// Module declaration
input clock; // clock input
input TS,TL,Vs; // declaration sequential logic inputs
output S0,S1,S2; // declaration sequential logic outputs
wire w1,w2,w3; // connection of module

input_logic (TS, TL, Vs, S0, S1, S2, w1, w2, w3); // input logic module
D_flipflop D1 (clock, w1, S0); // first bit counter module
D_flipflop D2 (clock, w2, S1); // second bit counter module
D_flipflop D3 (clock, w3, S2); // third bit counter module
endmodule

```



**Figure 4.6:** Diagram showing simulation result of module sequential\_logic

According to Figure 4.6 above, clock, TS (Short timer), TS (long timer), and Vs (Vehicle sensor) labeled as input condition. The S0 through S2 is labeled as output condition. If no vehicle at road the time long is stay in this state for 20s against. If sensor detect have vehicle in other road, the time long is ‘off’ and goes to the next state. For example, refer Figure 4.6, when time long is ‘on’ and has vehicle, the state goes to the next state and the n the time long is ‘off’.

#### 4.3.4 The combination of Combinational Logic and Sequence Logic

The simulation result for combination of combinational logic and sequence logic are shown in Figure 4.7 and the coding is shown below:

```

module sequence_combine (clock,TS,TL,Vs,GL1,G1,Y1,R1,GL2,G2,
Y2,R2,GL3,G3,Y3,R3,GL4,G4,Y4,R4,Long,Short);
// Module declaration
input clock; // input clock
input TS, TL, Vs; // input module declaration
output GL1,G1,Y1,R1,GL2,G2,Y2,R2,GL3,G3,Y3,R3,GL4,G4,Y4,R4,
Long,Short; // output module declaration
wire S0,S1,S2; // jumper to connect a modules

sequential_logic (clock,TS,TL,Vs,S0,S1,S2); // sequential logic module

```

```

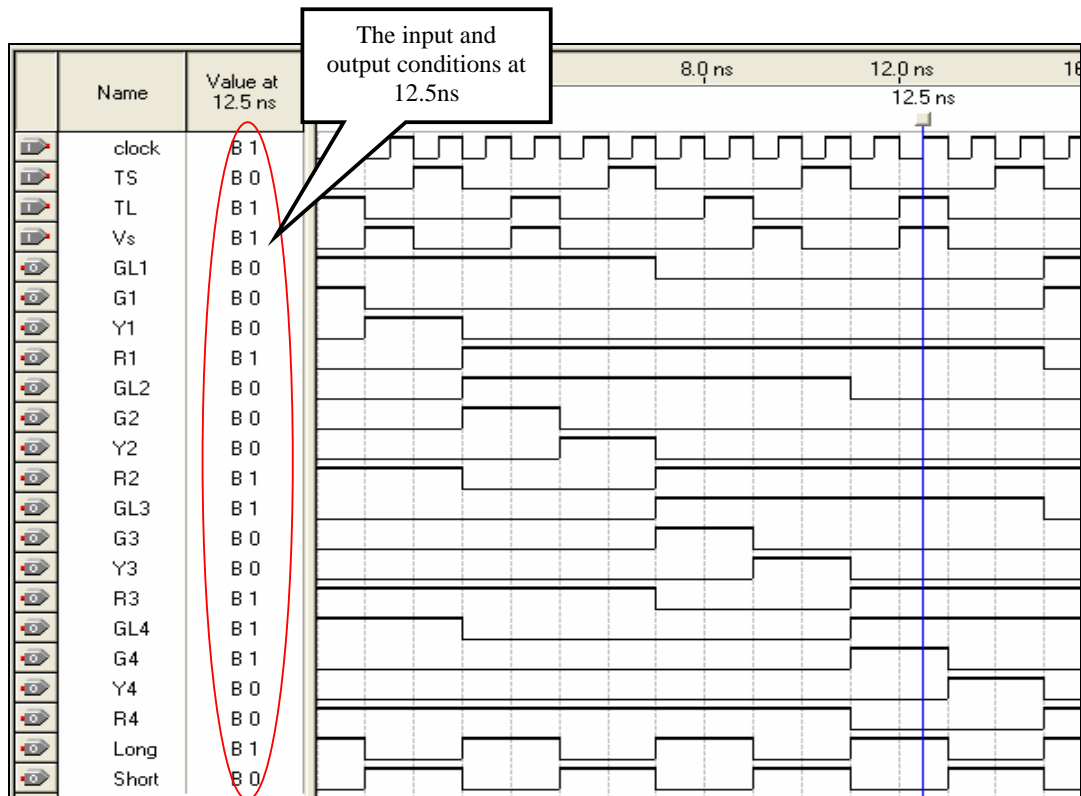
combinational_logic (S0,S1,S2,GL1,G1,Y1,R1,GL2,G2,Y2,R2,GL3,
G3,Y3,R3,GL4,G4,Y4,R4,Long,Short); // combinational logic module

```

```

endmodule

```

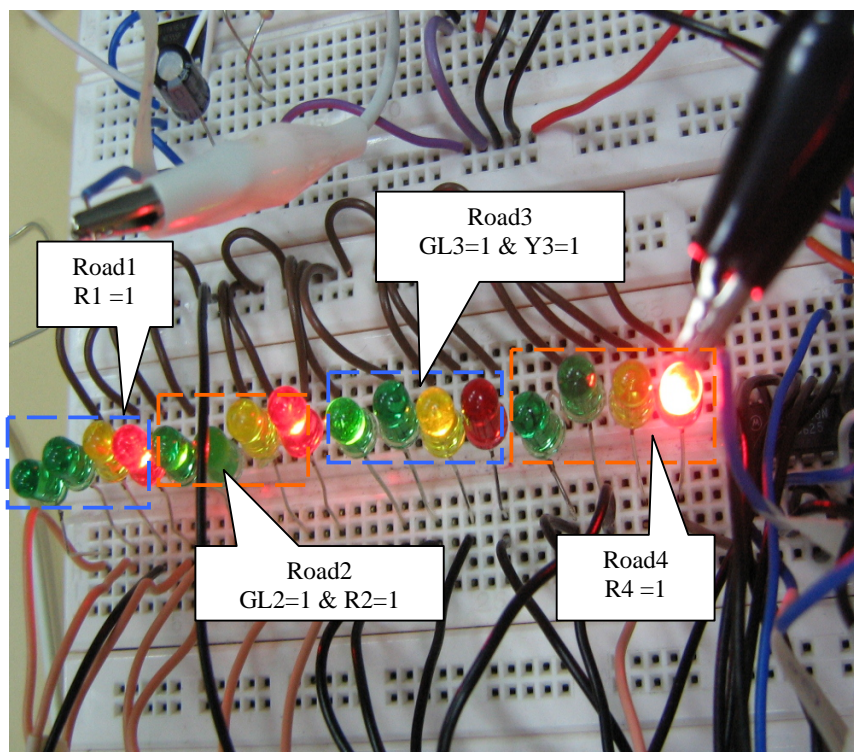


**Figure 4.7:** Combination of combinational logic and sequential logic

Based on Figure 4.7, the inputs are labeled as clock, TS, TL and Vs. The outputs are labeled as GL1 through GL4 for sign-left is green, G1 through G4 for the green light at second lanes, Y1 through Y4 for yellow light and R1 through R4 for red light. At 12.5ns, the timer long is 'on' and have vehicle in road, the output or signal light at Road1 and Road2 is red signal light. The signal Road3 is green for left-sign (GL3) and the signal at Road3 (G2) is red, the left-sign (GL4) and G3 for the Road4 are green. The three light signals will be green at the same time and the others lights are red.

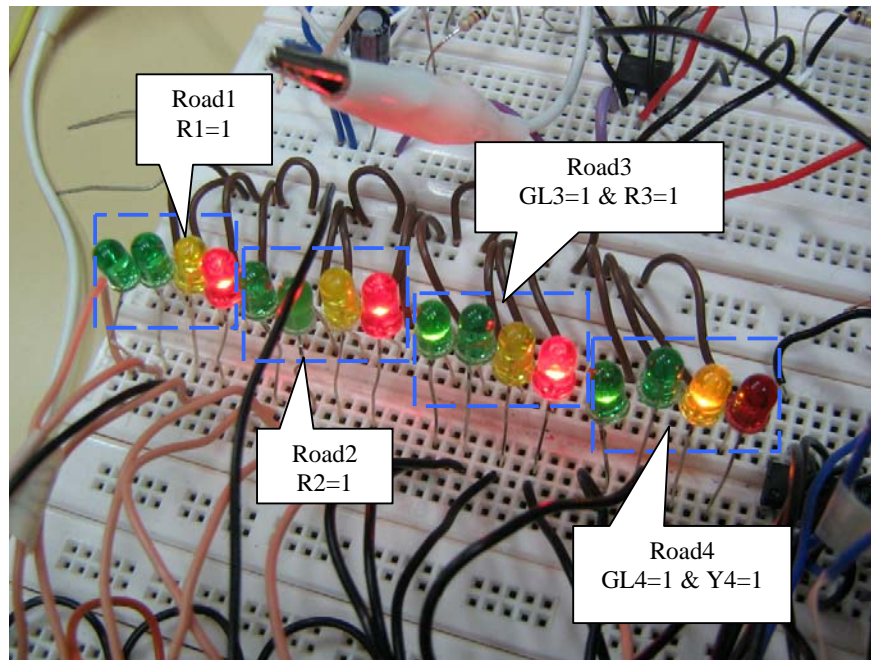
#### 4.4 Complete Combination of Traffic Light Controller

After finish construct all part of circuits, the output for traffic controller can represents. The examples traffic light signals are shown in Figure 4.8 and Figure 4.9. This is result after implement in hardware. By referring Figure 4.8, the traffic signal represents green colour at Road2 and Road3 for the left sign. The Road1 and Road4 are red and the Road3 at lane C2 is yellow. At this time, the timer short is 'on' for 4s at Road3. The Road2 and Road3 are 'on' for long timer.



**Figure 4.8:** Signal light function for state 6

According to Figure 4.8, the signal is green at Road3 and Road4 for left sign the signals at Road1, Road2 and Road3 are red colour. At this time, the short timer is 'on' at Road4. The long timer is 'on' at Road3 and Road4.



**Figure 4.9:** Signal light function for state 8

#### 4.5 Discussion

The outputs for traffic controller are dependent on the vehicle sensor at the road which will determine the long and short timer *on*. The maximum time for a long timer green is 40s and a short timer is 20s. The long and short timer is control by the setting time. The circuit for the timer short and long was design. For long timer, the one pulse input that can triggered for 20s and for the short timer one pulse that can triggered for 4s. Before construct the timing circuit, the value of capacitance and resistance must be calculate. LDR (Light Dependent Resistance) Sensor is used in this project. The LDR Sensor as a detector to detect the vehicle. It's most important device that converts the variations to their equivalent electrical pulses. The photo resistor or LDR is the same detectors. For demonstration, the LDR Sensor was chosen but for the real application, this device is not suitable. For real application, magnetic detector is used to detect are cars waiting.

In hardware, the wire connection gate logic is very important. Testing part by part is easy to detect the error or problem occurs in this project. The gate logic can

implement using IC (Integrated Circuit). The certain part it's not easy to get an output same with the theoretically. The environment, equipment and device will be able effect a result during a testing.

The step or process to finish this project is difficult. Before start a design a program, the all information and expected output must be recorded and the interface input and output also should be know in detail. Design a block diagram for the interface input and output should be an easy was to see design the system.

After combine all the parts of circuits, more problems are occur. The certain parts are not easy to handle especially the timing and the sequence circuits. Overall of this project is ok, but at certain condition the traffic signals not function properly. This is because the output from the timing circuits always changing. The environment, equipment and device can be causing the output result.