CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the process and flow to design a traffic light controller will be discussed. Figure 3.0 shows the flow chart of this project.

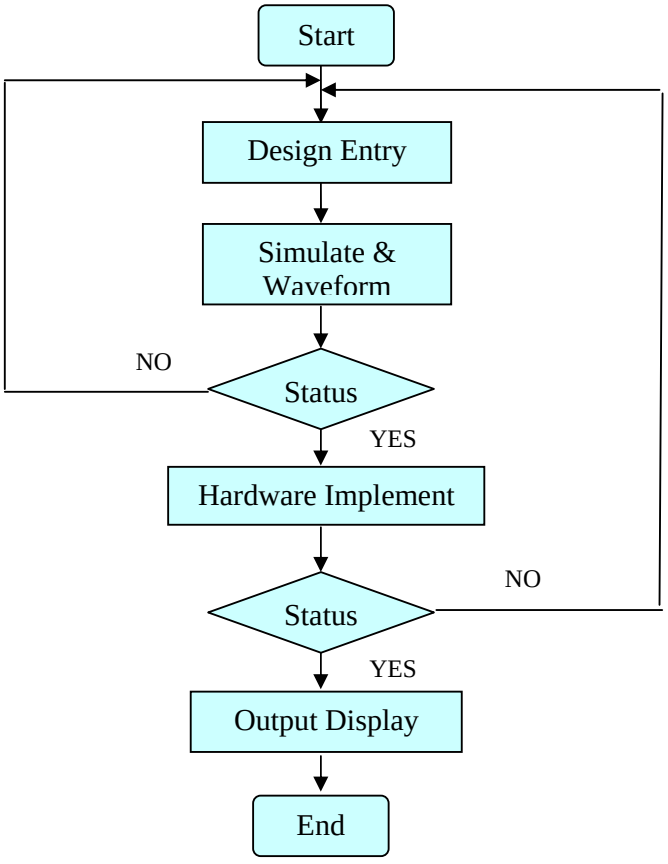

Figure 3.0: Flow Chart of traffic light controller design

This project consists of two parts, design a program using gate logic and implement this logic gate with hardware. In the fist part, QuartusII Software was chosen for designing a program and gate logic function of the traffic light controller. In this software, Block diagram/Schematic File and Verilog HDL text editor was chosen to design all logic gate of traffic light controller. After finish write a coding, continue with simulate to get a waveform and implement the gate logic in hardware.

3.2 Developing a Block Diagram and Interface Signal of Traffic Light Systems

Based on the requirements, a block diagram is created to defines the input and output signals of the traffic system controller. Table 3.0 shows a description of the functions of each interface signal.

Table 3.0: Interface Signal Description For Traffic Light Controller

Signal Name	Direction	Description	
Vs	Input	Sense a cars waiting in road	
GL1	Output	Green – Left sign	Traffic signals for Road 1
G1		Green	
Y1		Yellow	
R1		Red	
GL2	Output	Green – Left sign	Traffic signals for Road 2
G2		Green	
Y2		Yellow	
R2		Red	
GL3	Output	Green – Left sign	Traffic signals for Road 3
G3		Green	
Y3		Yellow	
R3		Red	
GL4	Output	Green – Left sign	Traffic signals for Road 4
G4		Green	
Y4		Yellow	
R4		Red	

This controller consists of three important parts, which are combinational logic, timing circuits and sequential logic. That systems control twelve different pairs of lights and

four left sign lights. Figure 3.1 is shows the input and output signals for the traffic light controller.

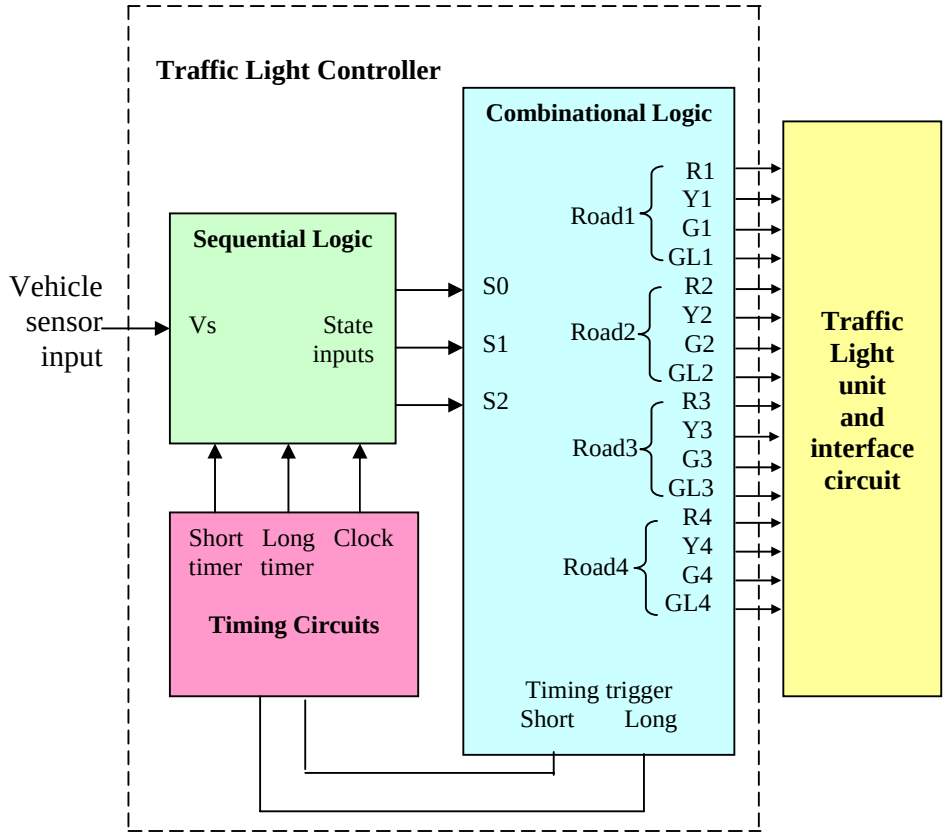

Figure 3.1: Traffic light control system block diagram

3.3 Possible Direction of Intelligent Traffic Light Controller

A state machine format is a good example for a design of an intelligent traffic control systems. A traffic control system has only certain fixed conditions to fulfill in order to control the traffic lights that control traffic flow [4].

Figure 3.2 shows an interchange junction of four roads. Traffic crossing the interchange junction needs to be regulated by a traffic light system in order to enable a smooth and safe interchange crossing for motorized vehicles.

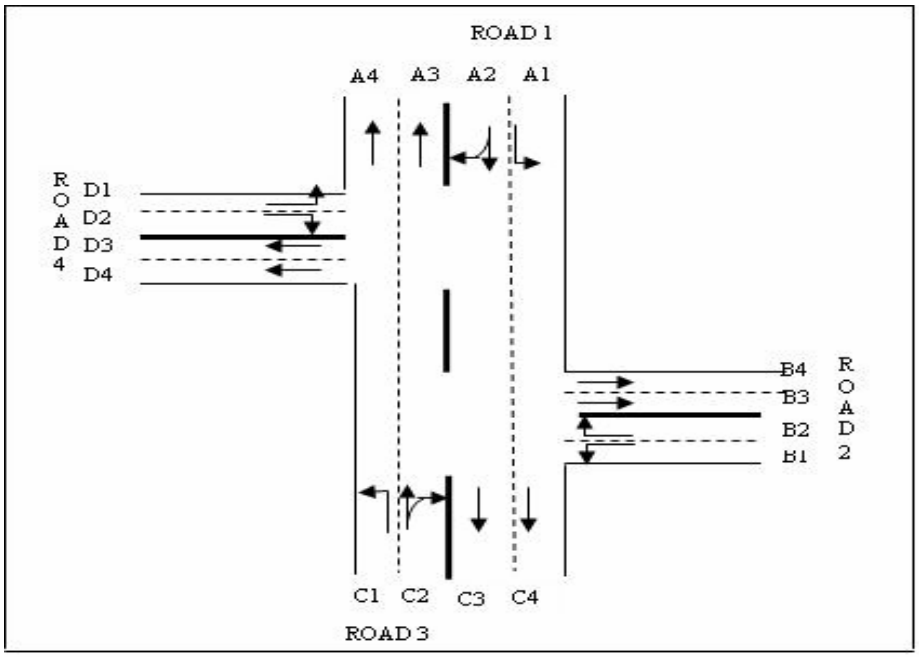

Figure 3.2: Diagram showing an interchange junction traffic light

Figure 3.2 shows all possible cars movement at the junction. By referring Figure 3.2, the interchange junction has four sets of traffic lights at Road1, Road2, Road3 and Road4. These four sets of traffic lights allow four different passes on the interchange junction. A sensor a located at lane A2 (Road1), B2 (Road2), C2 (Road3) and the last one is at D2 (Road4).

Figure 3.3 shows a possible situation where by the traffic lights at lane A1, A2 and D1 are GREEN while the others lanes are RED. Figure 3.4 shows a possible second situation where by the traffic lights at lane A1, B1 and B2 are GREEN while the others lanes are RED. Figure 3.5 shows a possible third situation where by the traffic lights at lanes B1, C1 and C2 are GREEN while the others are RED. Figure 3.6 shows a possible fourth or last situation where by the traffic lights at lanes C1, D1 and D2 are GREEN while the others are RED.

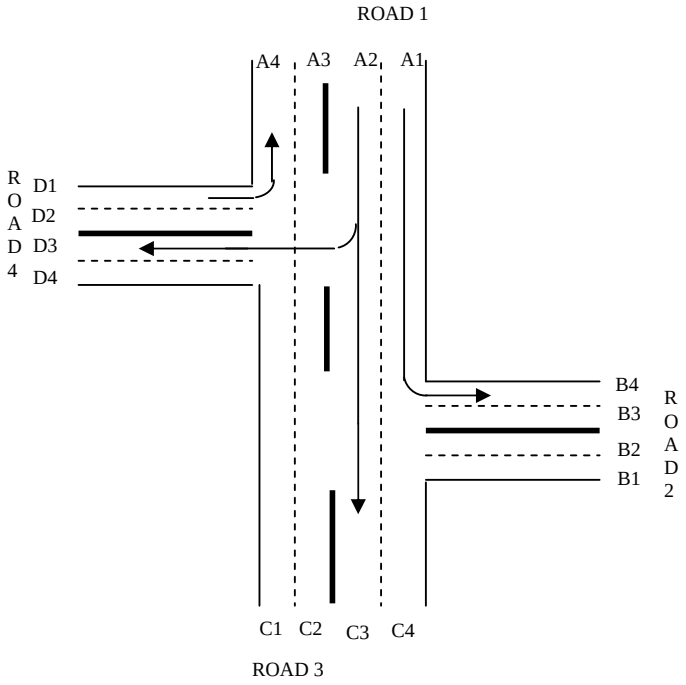

Figure 3.3: Diagram showing possible pass situation whereby lanes A1, A2, D1 is green

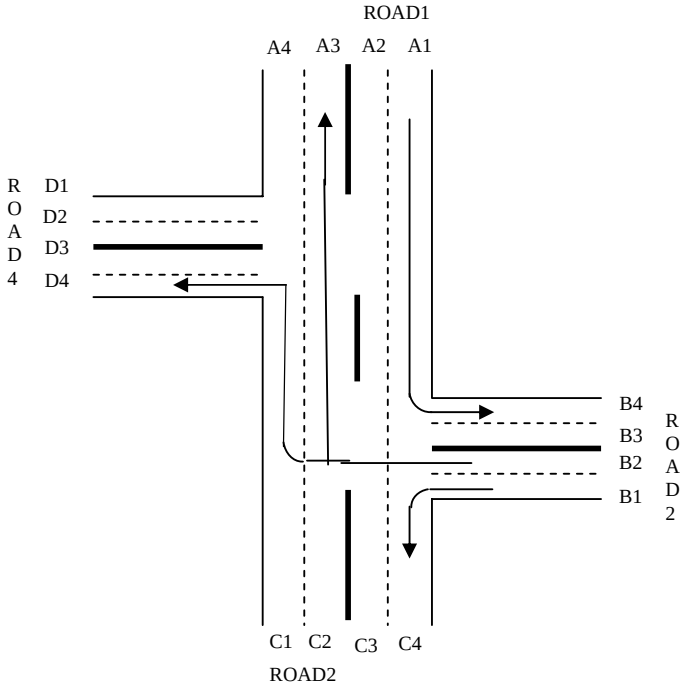

Figure 3.4: Diagram showing possible pass situation whereby lanes A1, B1, B2 is green

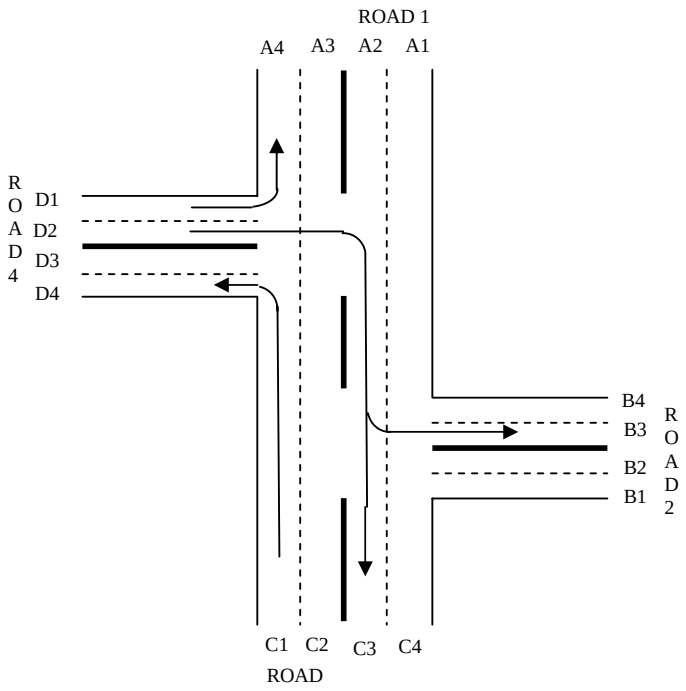

Figure 3.5: Diagram showing possible pass situation whereby lanes B1, C1, C2 is green

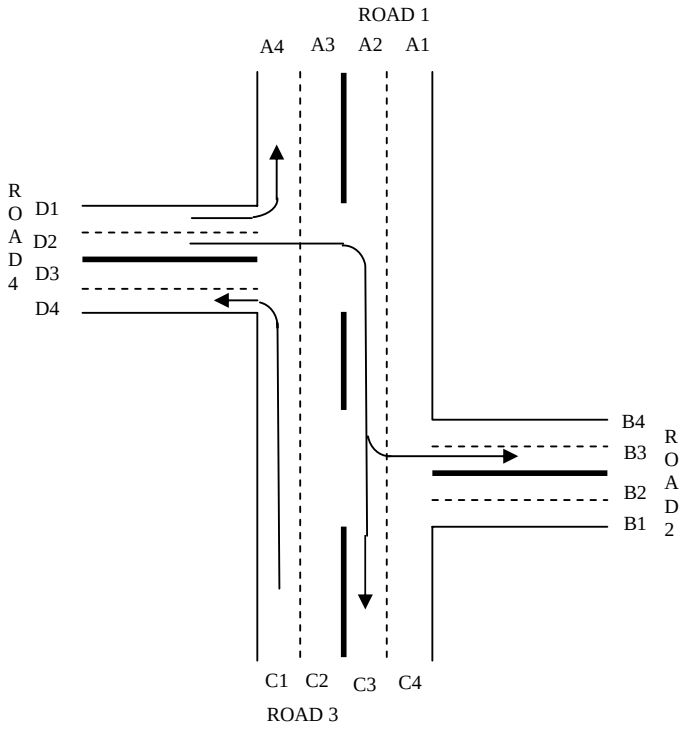

Figure 3.6: Diagram showing possible pass situation whereby lanes C1, D1, D2 is green

Figures 3.3 through 3.6 show the possible passes through the interchange junction. Apart from the four sets of traffic light at Road1, Road2, Road3 and Road4, there are present four sensors at lane A2, B2, C2 and D2. Each sensor is strategically located in each road to sense whether there are any waiting cars. For example, if there are have cars queue to turn from Road1 (as shown in Figure 3.3), the sensor A2 would sense that cars queuing at Road1. Similarly, are same for the other sensors. Sensor B2 would sense for cars queuing at Road2, C2 would sense for cars queuing at Road3 and D2 would sense for cars queuing at Road4. These four sensors together with an external timer would allow for the building block of an intelligent traffic light system.

3.4 State Diagram of Traffic Light Controller

A state diagrams graphically shows the sequence of states from one state to the next state. Before a traditional state diagram can be developed, the variables that determine how the systems sequences through its states must be defined. These variables and their symbols are listed as follows:

I. Vehicle present on side or main street = Vs
II. 20s timer (long timer) is ‘on’ = TL
III. 4s timer (short timer) is ‘on’ = TS

The use of complemented variables indicates the opposite conditions. For examples, $\overline{Vs}$ indicates that there is no vehicle on the side or main street, $\overline{TL}$ indicates the long timer if ‘off’ and $\overline{TS}$ indicates the short timer is ‘off’.

The state diagram for the traffic light output is shown in Figure 3.7. Each of eight states is labeled according to the 3-bit code sequence, as indicated by the circles. The looping arrow at each state indicates that the system remain in that state under the condition defined by the associated variable or expression. Each of the arrows going from one state to the next state indicates a state transition under the condition defined by

the associated variable or expression. Based on the state diagram in Figure 3.7, the sequential logic operation is described as follows:

I. **First state**: The code for this state is 000. The main street or Road1 is green light and the others road are red. The system remains in this state for at least 20s when the long timer is 'on' or as long as there is a vehicle on the next road ($TL + \overline{V}s$). The system goes to the next state when the timer 'off' and there is has vehicle on the next road ($\overline{TL}\ Vs$).

II. **Second state**: The code for this state is 001. The Road1 light is yellow and the others light is red. The system remains in this state for 4s when the short timer is 'on' (TS) and goes to the next state when the short timer goes 'off' ($\overline{TS}$).

III. **Third state**: The code for this state is 010. The Road1 or main street is red and the Road2 lights is green. The system remains in this state when the long timer is 'on' and there is a vehicle on the Road2 ($TL\ Vs$). The system goes to the next state when the 20s have elapsed or when there is no vehicle on the Road2, whichever comes first ($\overline{TL} + \overline{Vs}$).

IV. **Fourth state**: The code for this state is 011. The Road1 is red and the Road2 is yellow. The system remains in this state for 4s when short timer is 'on' (Ts) and goes to the next state when the short timer is 'off' ($\overline{Ts}$).

V. **Fifth state**: The code for this state is 100. The main street or Road3 is green light and the others road are red. The system remains in this state for at least 20s when the long timer is 'on' or as long as there is no vehicle on the next road ($TL + \overline{V}\,s$). The system goes to the next state when the timer 'off' and there is no vehicle on the next road. This is expressed as ($\overline{TLVs}$).

VI. **Sixth state**: The code for this state is 101. The Road3 light is yellow and the other traffic is red. The system remains in this state for 4s when the short timer is 'on' (Ts) and goes to the next state when the short timer goes 'off' ($\overline{Ts}$).

VII. **Seventh state**: The code for this state is 110. The Road3 or main street is red and the Road4 lights is green. The system remains in this state when the long timer is 'on' and there is a vehicle on the Road4 ($TL\ Vs$). The system goes to the next state when the 20s have elapsed or when there is no vehicle on the Road2, whichever comes first ($\overline{TL} + \overline{Vs}$).

VIII. **Eighth state**: The code for this state is 111. The Road3 is red and the Road4 is yellow. The system remains in this state for 4s when short timer is 'on' (Ts) and goes back to the first state when the short timer is 'off' ($\overline{Ts}$).

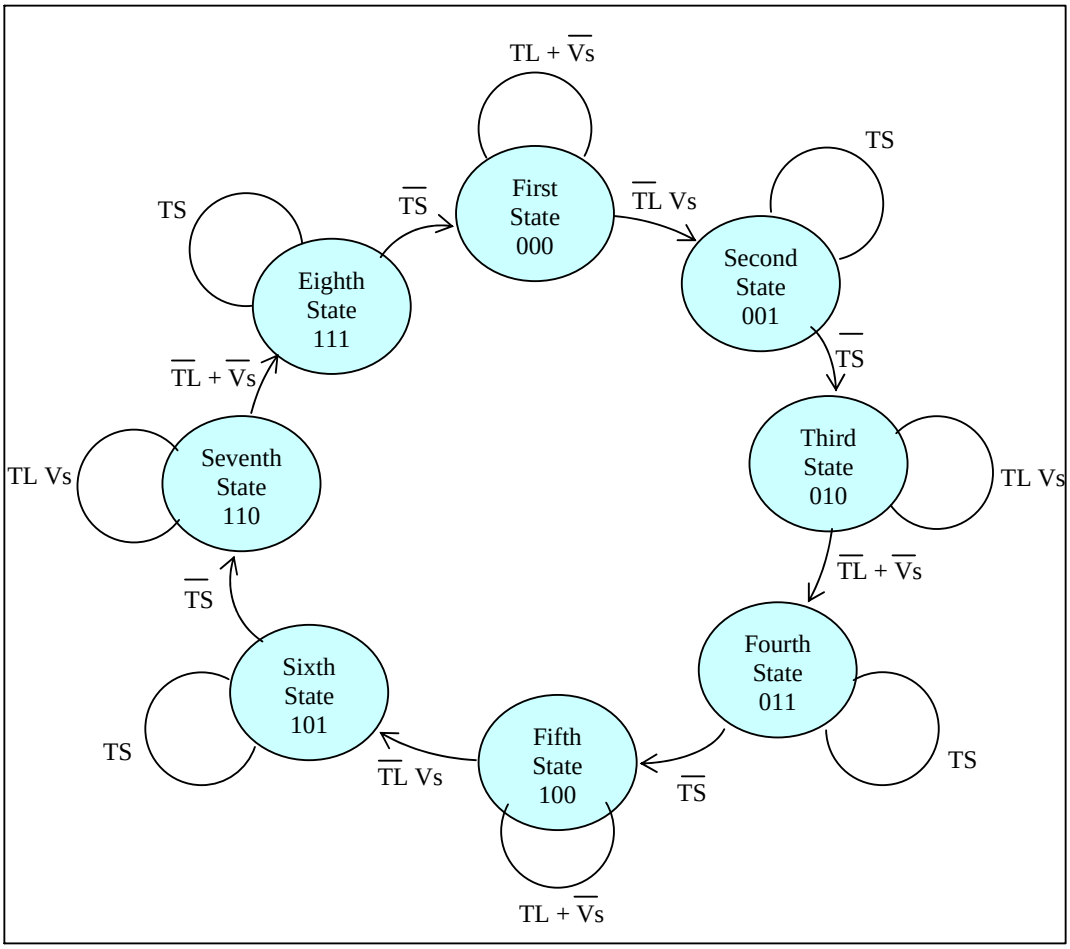

Figure 3.7: State diagram showing the 3-bit code sequence

3.5 The Combinational Logic

A block diagram for the combinational logic portion of the system is developed as the first step in designing the logic. The three functions that this logic must perform are defined as follows, and the resulting diagram with a block for each of the three

functions shown in Figure 3.8. Combinational logic design is using IC (integrated circuit) to implement the each of parts.

I. **State decoder**: Decodes the 3-bit code from the sequential logic to determine which of the eight states the system is in.
II. **Output Logic**: Uses the decoded to activate the appropriate traffic lights for the main and side street light units.
III. **Trigger Logic**: Uses to decoded states to produce signals for properly initiating (triggering) the long timer, 20s and the short timer, 4s.

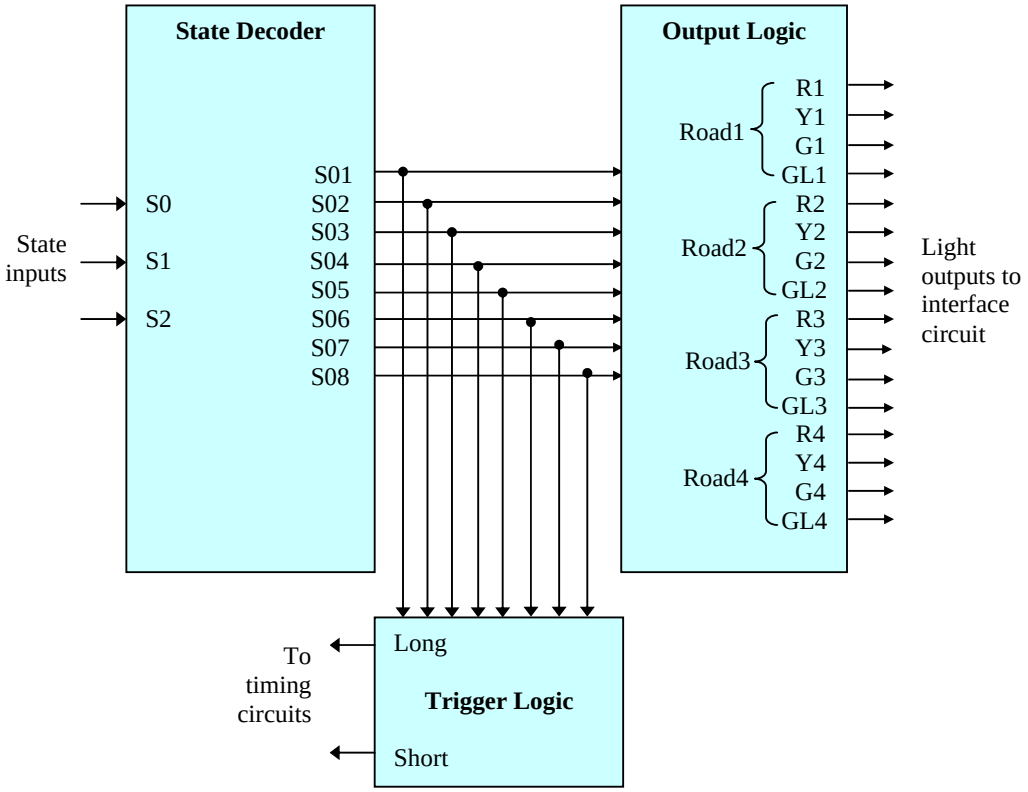

Figure 3.8: Block diagram of the combinational logic

3.5.1 Implementing the Decoder Logic

The state decoder section has three inputs (3-bit code) and an output for each of the eight states, as shown in Figure 3.8. The three bit code inputs are

designated S0, S1 and S2 and the eight state outputs are labeled S01, S02, S03, S04, S05, S06, S07 and S08. A truth table for state decoder is shown in Table 3.1 and the Boolean expressions for the state outputs are:

$$S01 = \overline{S2}\,\overline{S1}\,\overline{S0} \qquad S02 = \overline{S2}\,\overline{S1}\,S0$$

$$S03 = \overline{S2}\,S1\,\overline{S0} \qquad S04 = \overline{S2}\,S1\,S0$$

$$S05 = S2\,\overline{S1}\,\overline{S0} \qquad S06 = S2\,\overline{S1}\,S0$$

$$S07 = S2\,S1\,\overline{S0} \qquad S08 = S2\,S1\,S0$$

Table 3.1: Truth table for the state decoder logic

[Note: State outputs and light outputs are active-HIGH]

State Inputs			State Outputs							
S2	S1	S0	S01	S02	S03	S04	S05	S06	S07	S08
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

According to Boolean expressions and the truth table in Table 3.1, the gate logic can be design. The first output (S01) can high went the input logic is '000'. The steps to design gate logic for the others state outputs (S02 through S08) are also same with design the state output for the first state, but the input condition for every step was changing. The value for the input state is HIGH or LOW, refer to Boolean expressions in above. When one of the state outputs is HIGH, other state is LOW. To implement this in hardware, the needed IC (Integrated Circuits) are NOT/Inverter gate (74LS04) and 3-to-8 Line Decoder (74LS138) gate. The complete gate logic designs for state decoder are shown in Figure 3.9. Appendix A is shown the datasheet for Inverter gate and Appendix B for 3-to-8 Line Decoder gate.

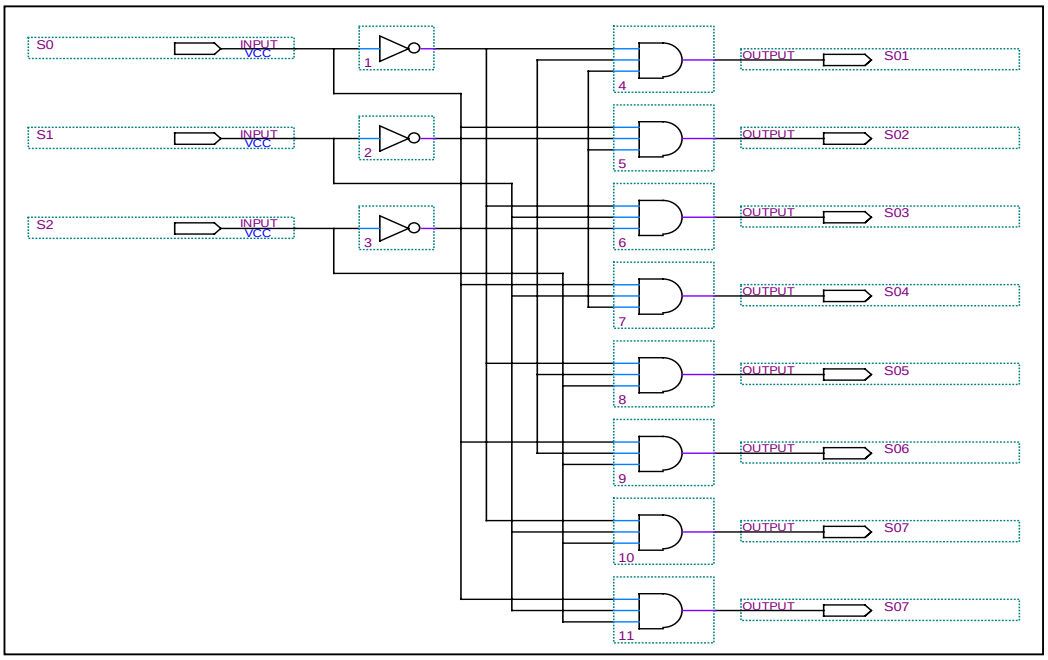

Figure 3.9: Gate logic diagram for state decoder

3.5.2 Implementing the Light Output Logic

The light output logic takes the eight state outputs and produces sixteen outputs for activating the traffic lights. The output logic must take the four active-high state outputs from the 74LS138 state decoder and produces sixteen outputs. A 74LS138 is active-low, so the inverter must add to get the outputs is active-high. These outputs are designated GL1, G1, Y1, R1 (for sign left Road1 green (lane A1), Road1 green (lane A2), Road1 yellow and Road1 red), GL2, G2, Y2, R2 (for Left-sign Road2 green (lane B1), Road2 green (lane B2), Road2 yellow and Road2 red), GL3, G3, Y3, R3 (for Left-sign Road3 green (lane C1), Road3 green (lane C2), Road3 yellow and Road3 red) and GL4, G4, Y4, R4 (for Left-sign Road4 green (lane D1), Road4 green (lane D2), Road4 yellow and Road4 red). A truth table for the output logic is shown in Table 3.2. Referring to the truth table in Table 3.2, the traffic light outputs can be expressed as

GL1 = S01 + S02 + S03 + S04

G1 = S01

Y1 = S02

R1 = S03 + S04 + S05 + S06 + S07 + S08

GL2 = S03 + S04 + S05 + S06

G2 = S03

Y2 = S04

R2 = S01 + S02 + S05 + S06 + S07 + S08

GL3 = S05 + S06 + S07 + S08

G3 = S05

Y3 = S06

R3 = S01 + S02 + S03 + S04 + S07 + S08

GL4 = S01 + S02 + S07 + S08

G4 = S07

Y4 = S08

R4 = S01 + S02 + S03 + S04 + S05 + S06

Based on expressions output and the truth table in Table 3.2, the gate logic can be design. The output logic gate is implementing in Figure 3.10. Only OR gate is used to implement this part. Refer datasheet in Appendix C for OR gate characteristic and logic diagram.

3.5.3 Implementing the Trigger Logic

The trigger logic produces two outputs. The long output produces a LOW-to-HIGH transition to trigger the 20s timing circuit (long timer) when the system goes into the first (000), third (010), fifth (100) or seventh (110) states. The short output is a LOW-to-HIGH transition that triggers the 4s timing circuit when the system goes into the second (001), fourth (011), sixth (101) or eighth (111) states. The trigger outputs are shown in Table 3.3 and the equation form are:

Long trigger = S01 + S03 + S05 + S07

Short trigger = S02 + S04 + S06 + S08

From the equation, the gate logic is shown in Figure 3.11. Only OR gates is used to implement this part. Appendix C is shown the datasheet for OR gate.

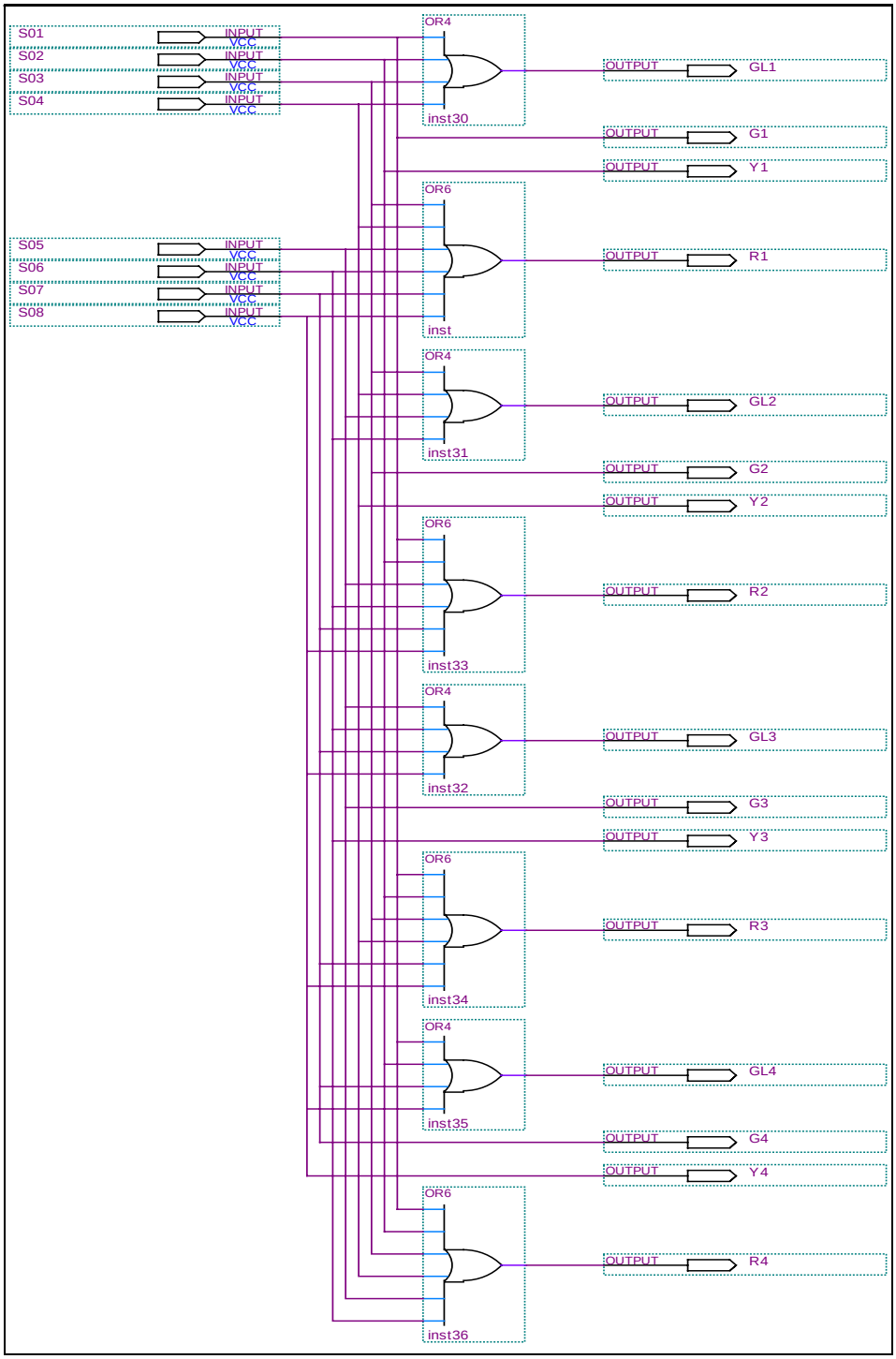

Figure 3.10: The gate logic diagram for Output Logic

Table 3.2: Truth table for the output logic

State Outputs								Light Outputs														
S01	S02	S03	S04	S05	S06	S07	S08	GL1	G1	Y1	R1	GL2	G2	Y2	R2	GL3	G3	Y3	R3	GL4	G4	Y4	R4
1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	1	0	0	1
0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	1	1	0	0	1
0	0	1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1
0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	1	1	0	0	0	0	0	1
0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1	1	0	1	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	1	1	1	0	0
0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0	1	1	0	1	0

Table 3.3: Truth table for the trigger logic

State Outputs								Trigger Outputs	
S01	S02	S03	S04	S05	S06	S07	S08	Long	Short
1	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1
0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	1	0	1

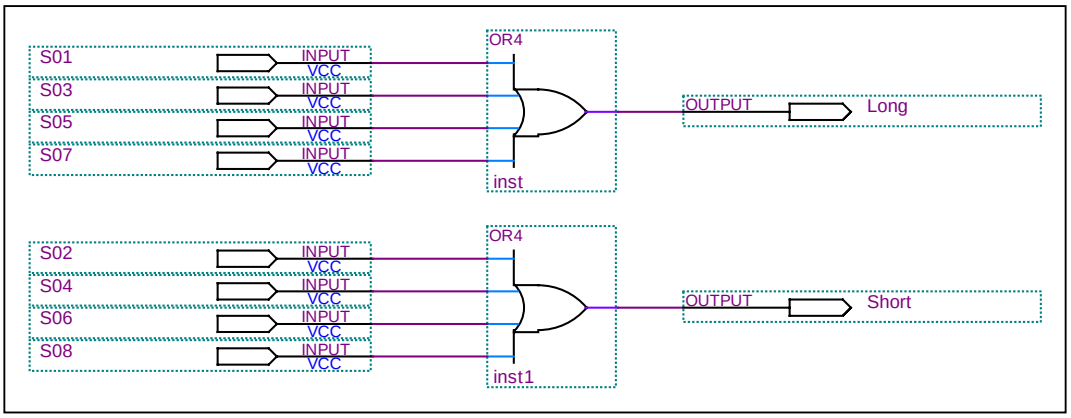

Figure 3.11: The gate logic for timing trigger

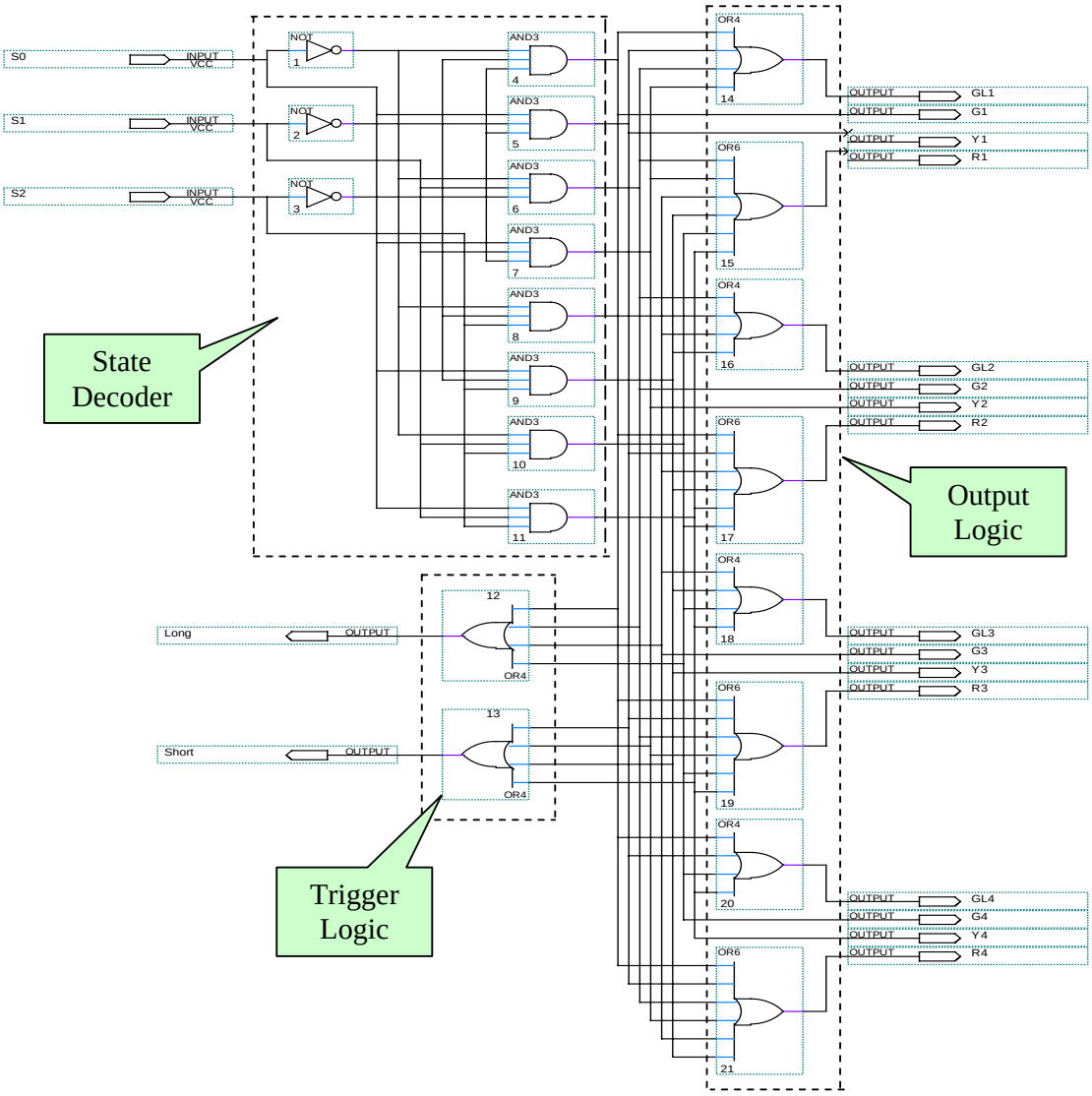

Figure 3.12: The complete combinational logic gate

The complete combinational logic is shown in Figure 3.12. This combinational logic consists of three parts, which are state decoder, output logic and trigger logic to decode the short and long timer. After complete design gate logic, continued with the gate circuit. Figure 3.13 is shown the circuit connection using IC for combinational logic.

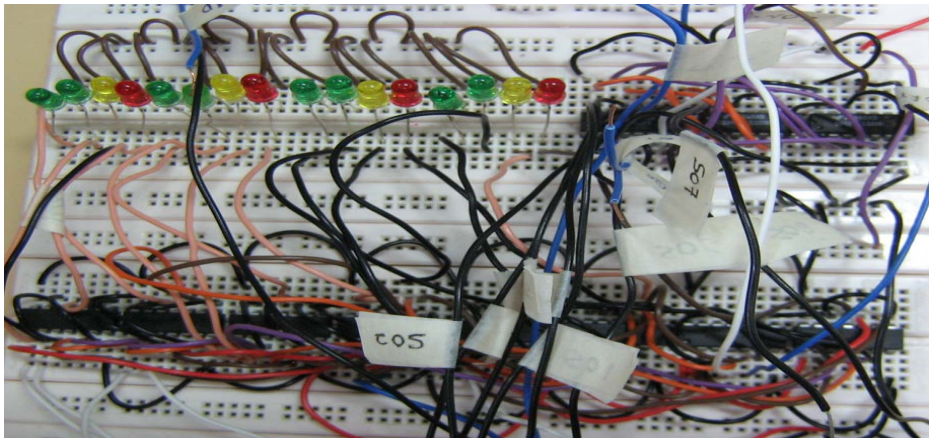

Figure 3.13: Combinational logic gate circuit

3.6 Timing Circuits Requirement

The timing circuits consist of three parts, the long timer, short timer and the clock oscillators shown in Figure 3.14. The long timer and short timer are implemented with one-shots, and the clock oscillator is implemented with a 555 timer connected in the astable mode. The datasheet for Timer 555 IC is shown in Appendix D.

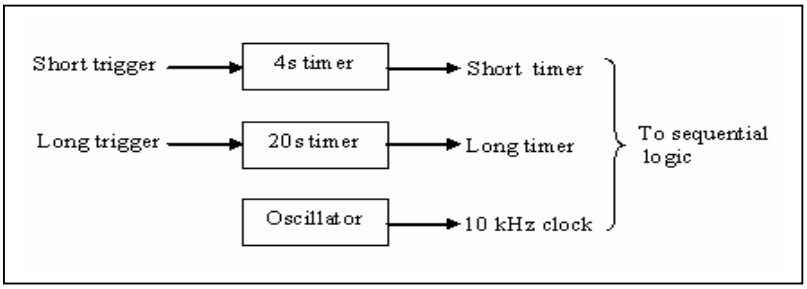

Figure 3.14: Block diagram of the timing circuits

Before construct this circuits, a value for each resistance and capacitance must calculate to mention the value for resistance and capacitance are related with the one-shots on the board that will produce a 4s positive pulse, a 20s positive pulse and a 10 kHz pulse waveform. The calculation for a value capacitance and resistance a shown below:

$Tw = 1.1 \times R \times C$ Apply $C = 100uF$

Long timer = 20s

$$R = \frac{Tw}{1.1 \times C} = \frac{20}{1.1 \times 100 \quad uF} = 182 \quad k\Omega$$

Short timer = 4s

$$R = \frac{Tw}{1.1 \times C} = \frac{4}{1.1 \times 100 \quad uF} = 36 \quad k\Omega$$

Figure 3.15 below is shown the circuit for timing requirements. It circuit consists of 3 parts, which are long timer, short timer and 10 kHz oscillator waveform.

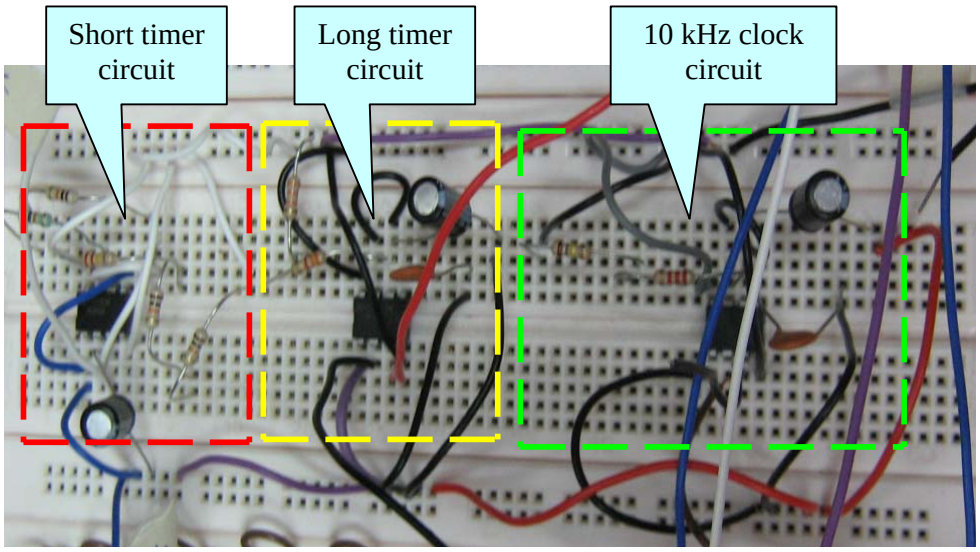

Figure 3.15: Timing circuit

3.7 Sequential Logic

The sequential logic controls the sequencing of the traffic lights based on inputs from the timing circuits and the vehicle sensor. The sequential logic consists of 3-bit code counter and associated input logic as shown in Figure 3.16.

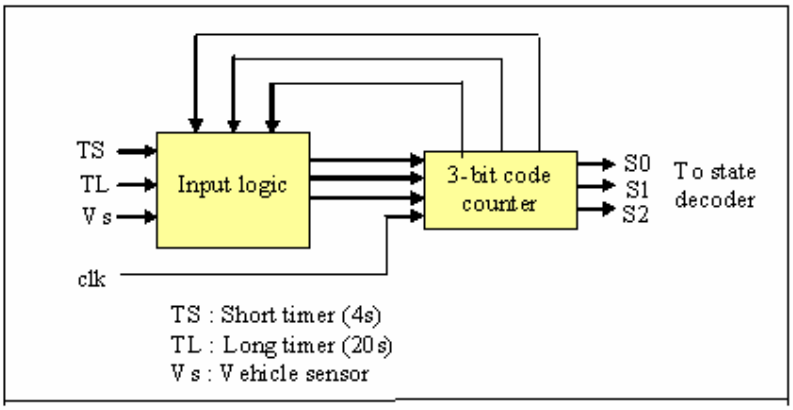

Figure 3.16: Block diagram of the sequential logic

The counter produces a sequence of eight states. Transitions from one to the next are determined by the 4s timer, the 20s timer and the vehicle sensor input. The clock for the counter is the 10 kHz signal produced by the oscillator in timing circuits.

The diagram in Figure 3.17 further defines the sequential logic. The counter consists of three D flip-flop for which the D inputs come from the input logic. The input logic has six input variables: Q0, Q1, Q2, TS, TL and Vs. From the state diagram, a next-state table can be developed as shown in Table 3.4. The input conditions for TL, TS and Vs for each present-state/next-state combination are listed in the table. Referring Table 3.4, the logic conditions required for each flip-flop to go the 1, state can be determined. For examples Q0 goes from 0 to 1 when the present-state is '000' and the input condition is $\overline{TL}\ Vs$, as indicated on the Table 3.4. D0 must be a 1 to make Q0 go to a 1 or to remain a 1 on the next clock pulse. Logic expressions for D0, D1 and D2 to be a 1 can be written from Tables 3.4. The expressions are:

$$D0 = \overline{Q2}\,\overline{Q1}\,\overline{Q0}\,\overline{TL}\,Vs + \overline{Q2}\,\overline{Q1}\,Q0\,TS + \overline{Q2}\,Q1\,\overline{Q0}\,(\overline{TL} + \overline{Vs}) + \overline{Q2}\,Q1\,Q0\,TS +$$
$$Q2\,\overline{Q1}\,\overline{Q0}\,\overline{TL}\,Vs + Q2\,\overline{Q1}\,Q0\,Ts + Q2\,Q1\,\overline{Q0}\ (\overline{TL} + \overline{Vs}) + Q2\,Q1\,Q0\,TS$$
$$= \overline{Q1}\ \overline{Q0}\,\overline{TL}\,Vs\,(\overline{Q2} + Q2) + \overline{Q1}\,Q0\,TS(\overline{Q2} + Q2) + Q1\,\overline{Q0}\ (\overline{TL} + \overline{Vs})(\overline{Q2} + Q2) +$$
$$Q1\,Q0\,TS\ (\overline{Q2} + Q2)$$
$$= \overline{Q1}\ \overline{Q0}\,\overline{TL}\,Vs + \overline{Q1}\,Q0\,TS + Q1\,\overline{Q0}\,\overline{Vs} + Q1\,\overline{Q0}\,\overline{TL} + Q1\,Q0\,TS$$
$$= Q0\,TS + \overline{Q1}\ \overline{Q0}\,\overline{TL}\,Vs + Q1\,\overline{Q0}\,\overline{Vs} + Q1\,\overline{Q0}\,\overline{TL}$$

$$D1 = \overline{Q2}\,\overline{Q1}\ Q0\,\overline{TS} + \overline{Q2}\,Q1\,\overline{Q0}\,TL\,Vs + \overline{Q2}\,Q1\,\overline{Q0}\,(\overline{TL} + \overline{Vs}) + \overline{Q2}\,Q1\,Q0\,TS +$$
$$Q2\,\overline{Q1}\,Q0\,\overline{TS} + Q2\,Q1\,\overline{Q0}\,TL\,Vs + Q2\,Q1\,\overline{Q0}\ (\overline{TL} + \overline{Vs}) + Q2\,Q1\,Q0\,TS$$
$$= \overline{Q1}\,Q0\,\overline{TS}\,(\overline{Q2} + Q2) + Q1\,\overline{Q0}\,TL\,Vs\,(\overline{Q2} + Q2) + Q1\,\overline{Q0}\ (\overline{TL} + \overline{Vs})(\overline{Q2} + Q2) +$$
$$Q1\,Q0\,TS\ (\overline{Q2} + Q2)$$
$$= \overline{Q1}\,Q0\,\overline{TS} + Q1\,\overline{Q0}\,TL\,Vs + Q1\,\overline{Q0}\ \overline{TL} + Q1\,\overline{Q0}\,\overline{Vs} + Q1\,Q0\,TS$$
$$= Q0\,(\overline{Q1}\,\overline{TS} + Q1\,TS) + Q1\,\overline{Q0}\,TL\,Vs + Q1\,\overline{Q0}\ \overline{TL} + Q1\,\overline{Q0}\,\overline{Vs}$$
$$= Q0\,(\overline{Q1 \oplus TS}) + Q1\,\overline{Q0}\,TL\,Vs + Q1\,\overline{Q0}\ \overline{TL} + Q1\,\overline{Q0}\,\overline{Vs}$$

$$D2 = \overline{Q2}\,Q1\,Q0\,\overline{TS} + Q2\,\overline{Q1}\,\overline{Q0}\ (TL + \overline{Vs}) + Q2\,\overline{Q1}\,\overline{Q0}\,\overline{TL}\,Vs + Q2\,\overline{Q1}\,Q0\,TS +$$
$$Q2\,\overline{Q1}\,Q0\,\overline{TS} + Q2\,Q1\,\overline{Q0}\,TL\,Vs + Q2\,Q1\,\overline{Q0}\ (\overline{TL} + \overline{Vs}) + Q2\,Q1\,Q0\,TS$$
$$= Q0\,\overline{TS}\,(\overline{Q2}\,Q1 + Q2\,\overline{Q1}) + Q2\,\overline{Q1}\,\overline{Q0}\,TL + Q2\,\overline{Q1}\,\overline{Q0}\,\overline{Vs} + Q2\,\overline{Q1}\,\overline{Q0}\,\overline{TL}\,Vs +$$
$$Q2\,Q0\,TS(\overline{Q1} + Q1) + Q2\,Q1\,\overline{Q0}\,\overline{TL} + Q2\,Q1\,\overline{Q0}\,\overline{Vs} + Q2\,Q1\,\overline{Q0}\,TL\,Vs$$
$$= Q0\,\overline{TS}\,(Q2 \oplus Q1) + Q2\,\overline{Q0}\,(\overline{Q1}\,TL + Q1\,\overline{TL}) + Q2\,Q0\,TS(\overline{Q1} + Q1) +$$
$$Q2\,\overline{Q0}\,\overline{Vs}\,(\overline{Q1} + Q1) + Q2\,\overline{Q0}\,Vs\,(\overline{Q1}\,\overline{TL} + Q1\,TL)$$
$$= Q0\,\overline{TS}\,(Q2 \oplus Q1) + Q2\,\overline{Q0}\,(Q1 \oplus TL) + Q2\,Q0\,TS +$$
$$Q2\,\overline{Q0}\,\overline{Vs} + Q2\,\overline{Q0}\,Vs\,(\overline{Q1 \oplus TL})$$

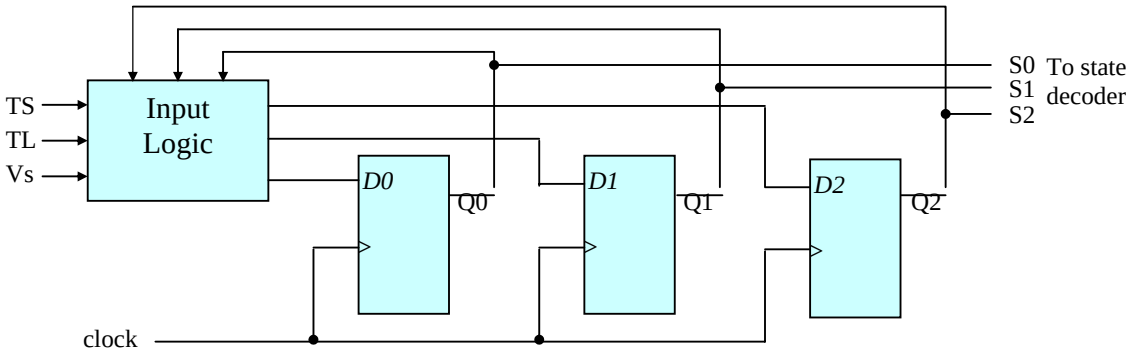

Figure 3.17: Sequence logic diagram

Table 3.4: Next-state table for the sequential logic functions

Present State			Next State			Input	Flip-flop Inputs		
Q2	Q1	Q0	Q2	Q1	Q0	Conditions	D2	D1	D0
0	0	0	0	0	0	$TL + \overline{Vs}$	0	0	0
0	0	0	0	0	1	$\overline{TL}\ Vs$	0	0	1
0	0	1	0	0	1	TS	0	0	1
0	0	1	0	1	0	$\overline{TS}$	0	1	0
0	1	0	0	1	0	$TL\ Vs$	0	1	0
0	1	0	0	1	1	$\overline{TL} + \overline{Vs}$	0	1	1
0	1	1	0	1	1	TS	0	1	1
0	1	1	1	0	0	$\overline{TS}$	1	0	0
1	0	0	1	0	0	$TL + \overline{Vs}$	1	0	0
1	0	0	1	0	1	$\overline{TL}\ Vs$	1	0	1
1	0	1	1	0	1	TS	1	0	1
1	0	1	1	1	0	$\overline{TS}$	1	1	0
1	1	0	1	1	0	$TL\ Vs$	1	1	0
1	1	0	1	1	1	$\overline{TL} + \overline{Vs}$	1	1	1
1	1	1	1	1	1	TS	1	1	1
1	1	1	0	0	0	$\overline{TS}$	0	0	0

Referring the Boolean expressions and the truth table in Table 3.4, the gate logic diagram for sequential circuit is design. The input and output is depend the equation. The combining gate logic using IC for 3-bit counter and input logic are shown in Figure 3.18. The 2-input AND gate (74LS08), 74F11 Triple 3-Input AND gate, NOT/Inverter gate (74LS04), Exclusive-OR gate (74LS86), OR (74LS32) and Dual D-type Positive

Edge Triggered Flip-flop gate (74LS74) are used to implement the traffic light signal. The data sheets for 2-input AND gate is shown in Appendix E and for Triple 3-input AND gate is shown in appendix F. Datasheets for Ex-OR gate refer Appendix G, Inverter gate refer Appendix A, OR gate referred Appendix C and Dual Flip-flop refer Appendix H. The complete gate circuit traffic light controller is shown in Figure 3.19. The gate logic diagram for sequential logic is shown in Figure 3.20.

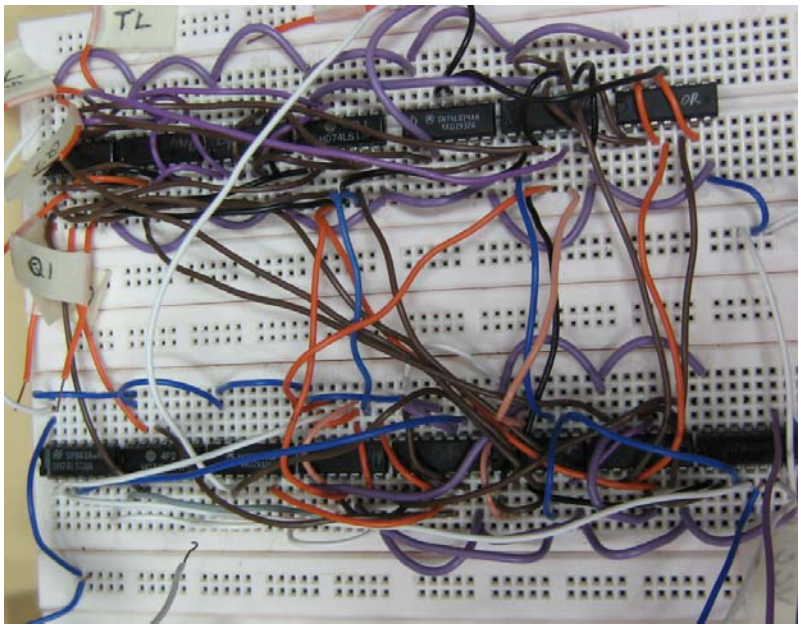

Figure 3.18: Sequential logic gate circuit

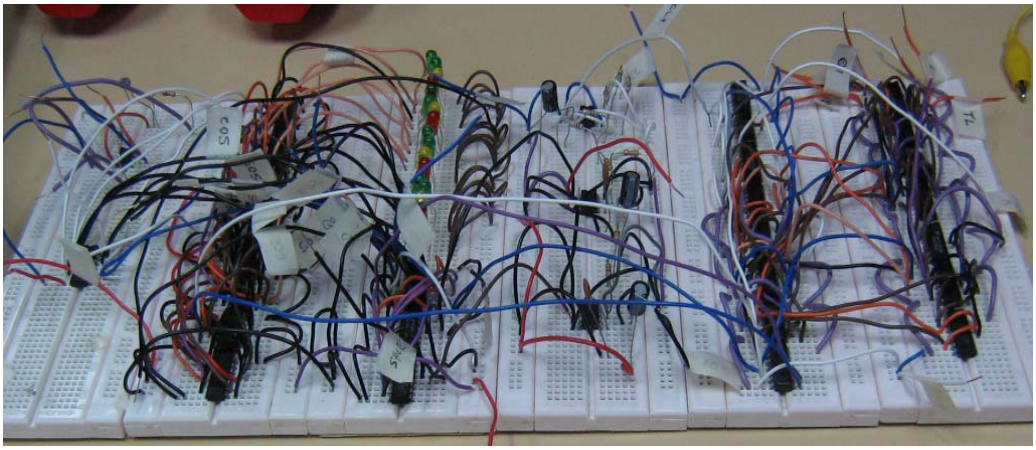

Figure 3.19: The complete circuit combination

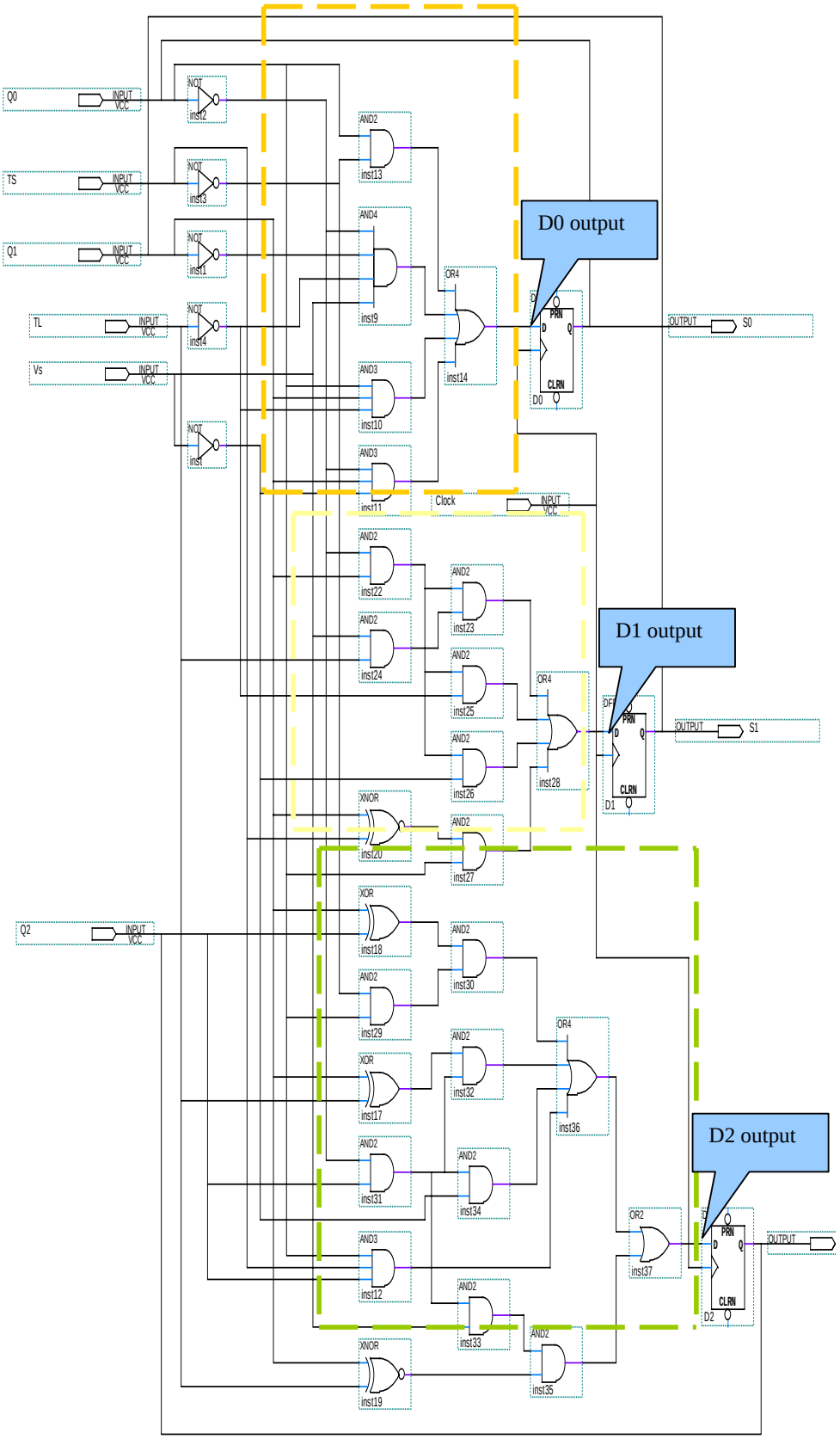

Figure 3.20: The sequential logic gate