Digital Electronics

Traffic Light

Nouran Soliman

13101167

Abstract

This documentation is written to describe the solution to a traffic light problem. The problem statement requires to switch between 2 sets of traffic lights at a cross providing the features of instant switching by the officer and variable input time for the lights.

The solution circuit is designed using a 2-bit asynchronous counter implemented with 2 D flip-flops to switch between the traffic lights. For clock generation, an astable circuit is used. Pulse generation is also done using 2 monostable circuits to instantly switch the lights. The total cost of the project is 36.5 Egyptian Pounds.

The report includes an analysis of different solutions, and a detailed description of the design, transistor-level implementation, operation and cost of the project.

Table of Contents

1. Introduction 2
2. Problem Statement 2
3. Literature Review 3
   3.1 Using Comparator, Full adder and Counter 3
   3.2 Using 555 timer 3
   3.3 Using Counter 4
4. Circuit Block Diagram 5
5. Transistor-Level Implementation 7
   5.1 OR Gate 7
   5.2 AND Gate 8
   5.3 D Flip-flop 9
   5.4 Clock generator 14
   5.5 Pulse generator 16
6. System Integration 17
7. Practical Implementation 19
8. Cost Analysis 21
9. Conclusion 21
10. Future Work 21
11. References 21
12. Appendix 22

Nomenclature

LED – Light Emitting Diode

RTL – Resistor-to-Transistor Logic

TTL – Transistor-to-Transistor Logic

DIP – Dual In-line Package

1. Introduction

Traffic lights, also known as traffic signals, stop lights, traffic lamps, stop-and-go lights, robots or semaphore, are signaling devices positioned at road or junction intersections, pedestrian crossings, and other locations to control competing flows of traffic.

Traffic lights have been installed in most cities around the world to control the flow of traffic. They assign the right of way to road users by the use of lights in standard colors (Red - Amber - Green), using a universal color code (and a precise sequence, for those who are color blind). They are used at busy intersections to more evenly apportion delay to the various users.

The most common traffic lights consist of a set of three lights: red, yellow (officially amber), and green. When illuminated, the red light indicates for vehicles facing the light to stop; the amber indicates caution, either because lights are about to turn green or because lights are about to turn red; and the green light to proceed, if it is safe to do so.

There are many variations in the use and legislation of traffic lights, depending on the customs of a country and the special needs of a particular intersection. There may, for example, be special lights for pedestrians, bicycles, buses, trams, etc.; light sequences may differ; and there may be special rules, or sets of lights, for traffic turning in a particular direction. Complex intersections may use any combination of these.

The normal function of traffic lights requires a slightly more control and coordination to ensure that traffic moves as smoothly and safely as possible and that pedestrians are protected when they cross the roads. A variety of different control systems are used to accomplish this, ranging from simple clockwork mechanisms to sophisticated computerized control and coordination systems that self-adjust to minimize delay to people using the road.

2. Problem Statement

The problem statement requires designing and building a Digital Traffic Light controller in transistor logic. The traffic of cars is to be adjusted at a crossing with two sets of colored lights (red, amber and green) in order to arrange for different timings for each of the flow directions and the waiting period. The circuit must be adjustable in time. In addition, a direct interfere of the officer is important in the cases of emergencies in order to alter the direction of the traffic. In addition, pedestrian crossing access must be included.

3. Literature Review

Before implementing a certain solution, several comparisons were made to evaluate the functionality, simplicity and cost of each design in order to pick a good compromise.

3.1 Using Comparator, Full adder and Counter

Fig.1 shows the block diagram of the circuit. This circuit was intended to take the input time for each LED from a DIP switch. This time is added to the time of all the LEDs operated before this LED and is input to a comparator. The other input of the comparator is taken from a counter. When the counter reaches the required time the suitable LED is illuminated. Implementing this circuit is inconvenient as it is very huge and will cost a lot.

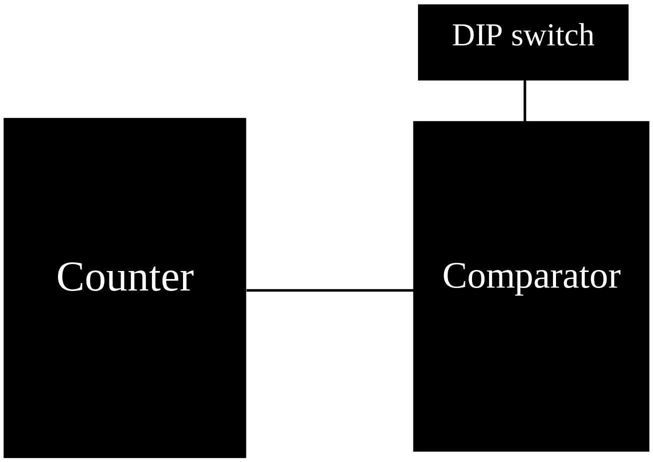


Fig. 1: Circuit of full adder and comparator

3.2 Using 555 timer

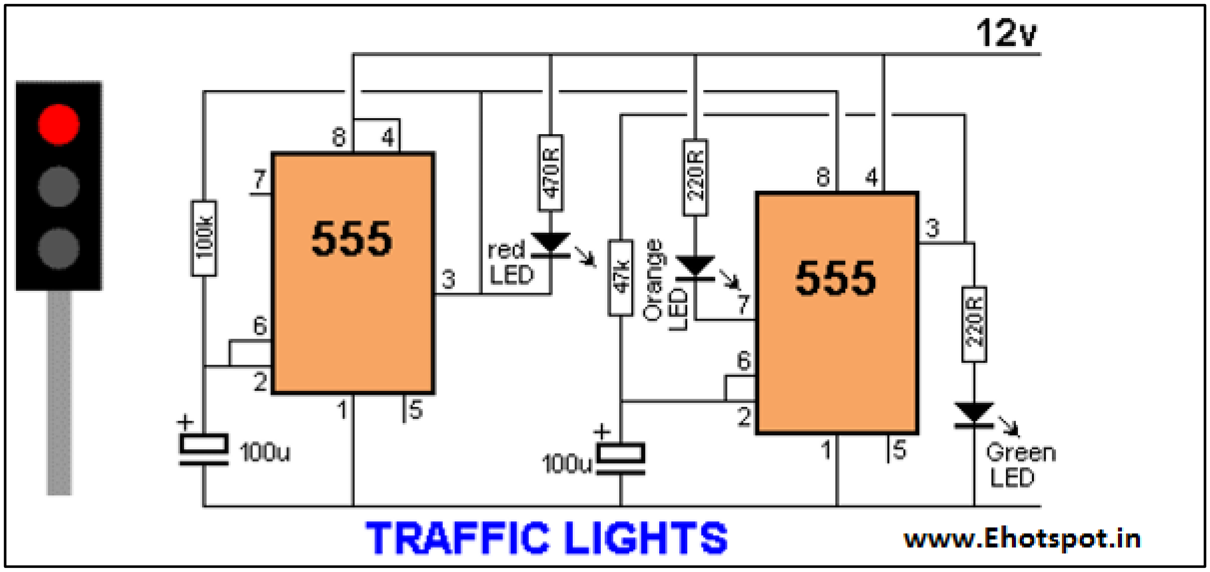


Fig. 2: Circuit of 555 timer

Both 555's are wired as oscillators in astable mode (Fig.2) and will oscillate all the time when they are turned on. But the second 555 is not turned on all the time. The first 555 turns on and the 100u is not charged. This makes output pin 3 HIGH and the red LED is not illuminated. However the output feeds the second 555 and it turns on.

Output pin 3 of the second 555 turns on the green LED and the second 100u charges to 2/3 rail voltage and causes the 555 to change states. The green LED goes off and the orange LED turns on. The second 100u starts to discharge, but the first 100u is charging via a 100k and after the orange LED has been on for a short period of time, the first 555 changes state and pin 3 goes LOW.

This turns on the red LED and turns off the second 555. The first 100u starts to discharge via the 100k and eventually it changes state to start the cycle again. The problem of this circuit is that the timing depends on the long cycle-time of the first 555 due to the 100k and the short cycle due to the 47k on the second 555. Therefore, the time of the green and amber lights must be equal to the red one.

<u>COMPONENTS USED:</u>
1. Two 555 IC.
2. Two 100uF cap.
3. One 100K resistance.
4. One 47K resistance.
5. One 470ohm resistance.
6. Two 220ohm resistance.
7. One red LED.
8. One orange LED.
9. One green LED.
10. One 6-12V power Supply.

This circuit is quite simple and cheap but does not fulfill the function required.

3.3 Using Counter

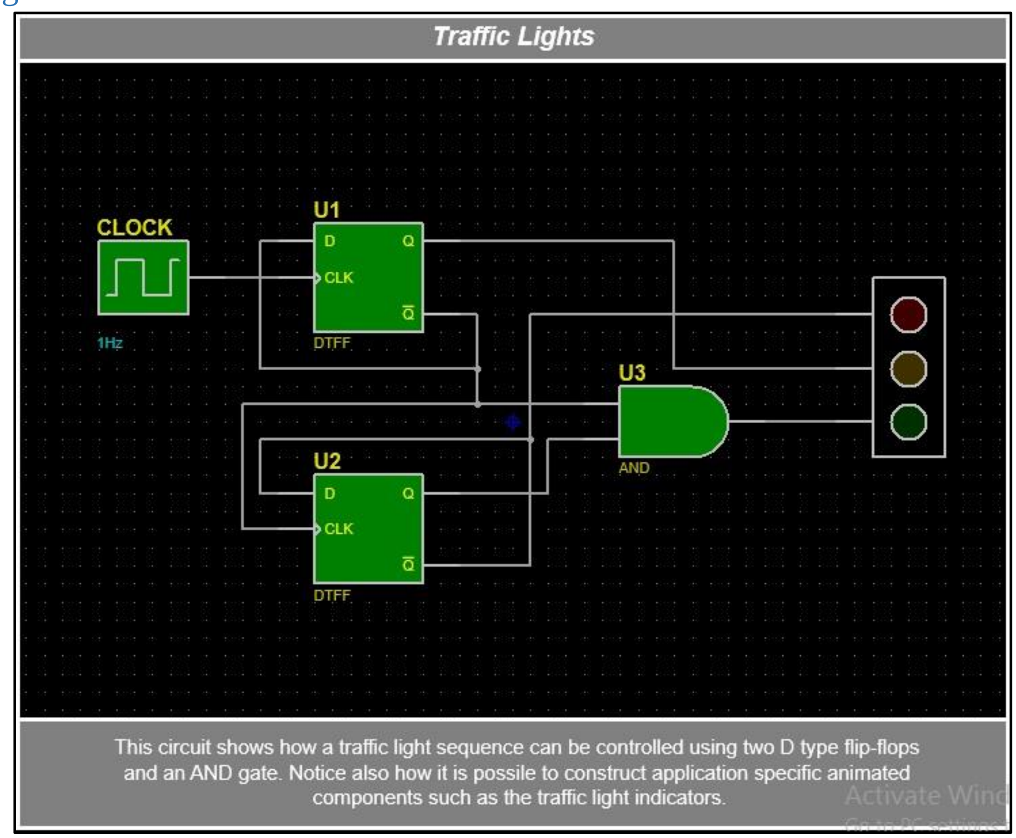


Fig. 3: Circuit using D flip-flop counter

The solution described through this documentation is based on this circuit in Fig.3. This circuit uses a 2-bit asynchronous counter implemented using 2 D Flip-Flops and an AND gate. The D flip-flops generate numbers from 0 to 3 in binary. These numbers are used to satisfy the 4 cases present in the problem. The 4 cases are: Red, Red and Yellow, Green, and then Yellow (Fig.4). The red LED should illuminate in case of 00, the green LED in case of 10 and the yellow LED in case of 01 or 11.

Let the output of flip-flop 1 = Q1 and output of flip-flop 2 = Q2.

$\therefore$ Red = Q2', Yellow = Q1, and Green = Q1'Q2.

This illuminates the LEDs in correct order for a certain time duration depending on the clock fed to flip-flop 1.

Compared to the other solutions, this circuit has relatively moderate complexity and cost, and can be easily modified to fulfill all the requirements of the problem statement.

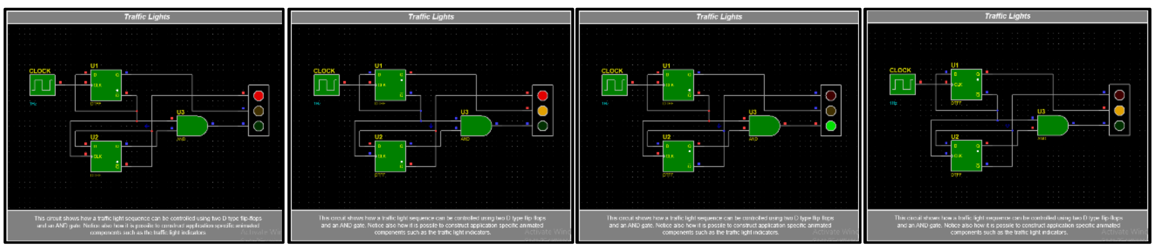


Fig. 4: Stages of D flip-flop circuit

4. Circuit Block Diagram

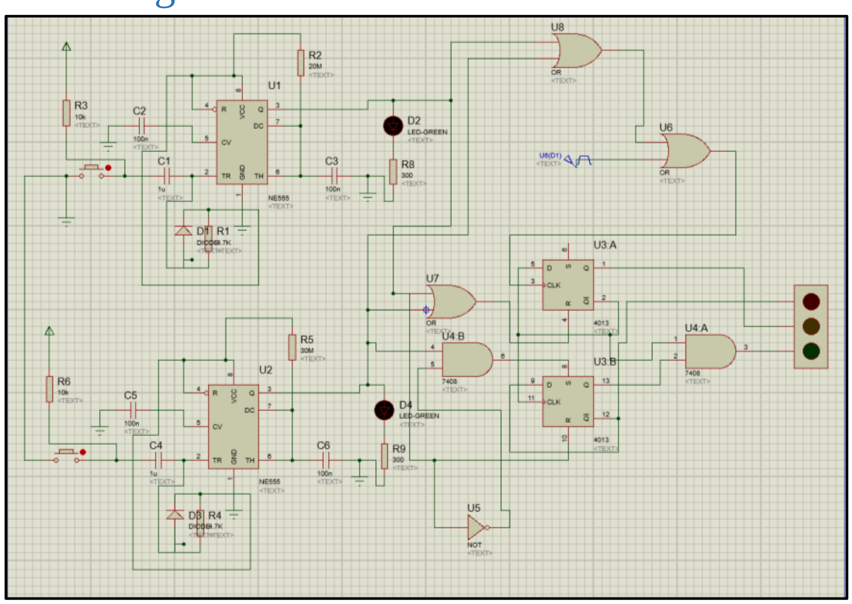


Fig. 5: Solution block diagram

Fig.5 shows the full block diagram of the circuit after doing some improvements to fulfill all of the requirements. The circuit consists of 3 OR gates, 2 AND gates, 2 D flip-flops, 1 clock generator and 2 pulse generators. The LEDs sequence operates as explained in the previous section. (Fig.6)

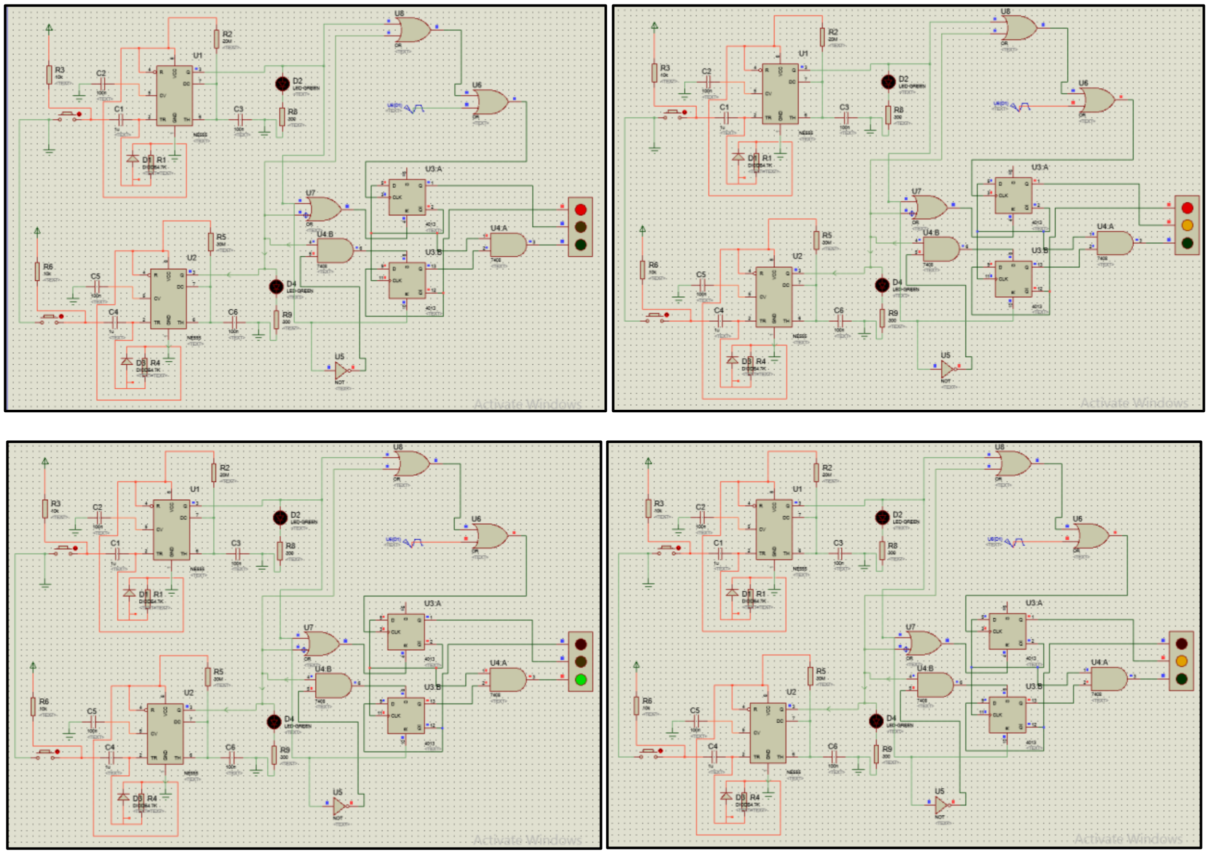


Fig. 6: Circuit stages

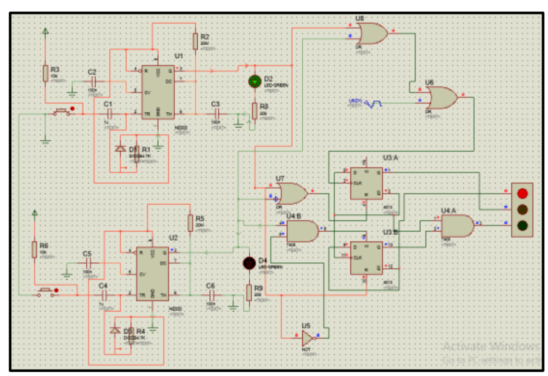


Fig. 7: Red Interrupt

In order for the officer to be able to instantly switch the traffic light, a switch is used to trigger a pulse generator, which produces 1 pulse with the duration required to turn-on the LED after which the cycles resumes its normal operation. If the switch of the Red LED is pressed, it is required that both flip-flops output 0 to light up the Red LED (Fig.7). However, if the switch of the Green LED is pressed, flip-flop 1 is required to output 0 and flip-flop 2 is required to output 1 to light up the Green LED (Fig.8). Therefore, the set of flip-flop 1 is always grounded, the reset of flip-flop 1 is connected to the OR-ing of both pulses, the set of flip-flop 2 is connected to the pulse responsible for the Green LED and the reset of flip-flop 2 is connected to the pulse responsible for the Red LED. If both switches are pressed at the same time, the set and reset of flip-flop 2 will have an input of 1. To avoid this situation, the set of the flip-flop is connected to an AND gate of inputs: Green LED pulse and inversion of Red LED pulse. This allows the Red LED to take over, however, it is illogical to operate the Red and Green LEDs at the same time. Any of the generated pulses will also disable the clock of the flip-flop through 2 OR gates so that the flip-flop keeps its last state after setting and resetting for the duration of the pulse. The first OR gate is used so that if any of the switches is pressed, the output is 1. The output of the first OR is then OR-ed with the main clock fed to the circuit. This ensures that the clock is disabled if any of the switches is pumped. This is done for one set of traffic lights and the second set is connected in an opposite order.

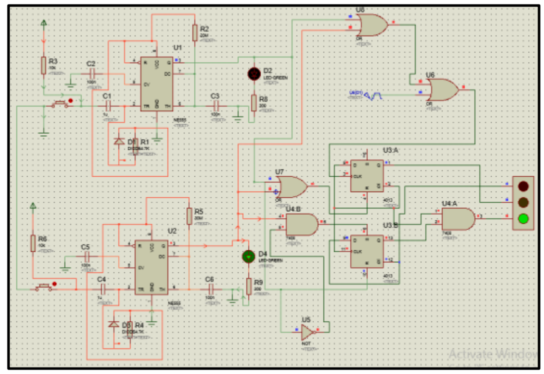


Fig. 8: Green Interrupt

5. Transistor-Level Implementation

5.1 OR Gate

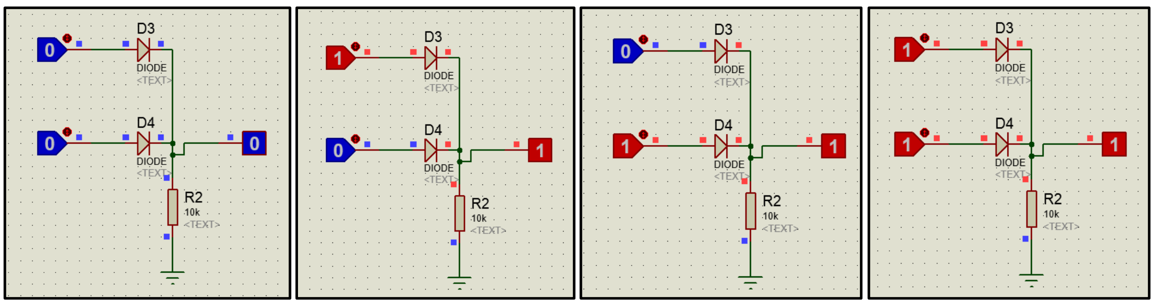


Fig. 9: RTL OR gate

At first, the OR gate was implemented using RTL. If any or both of the inputs are 0, the output is 0. If both inputs are 1, the output is 1. This matches the truth table of the OR gate (Fig.9).

This circuit works well in the simulation, however, during the practical implementation of the circuit, the RTL OR gate caused problems to the circuit and was switched in to TTL logic (Fig.10).

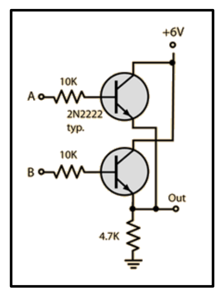


Fig. 10: TTL OR gate

5.2 AND Gate

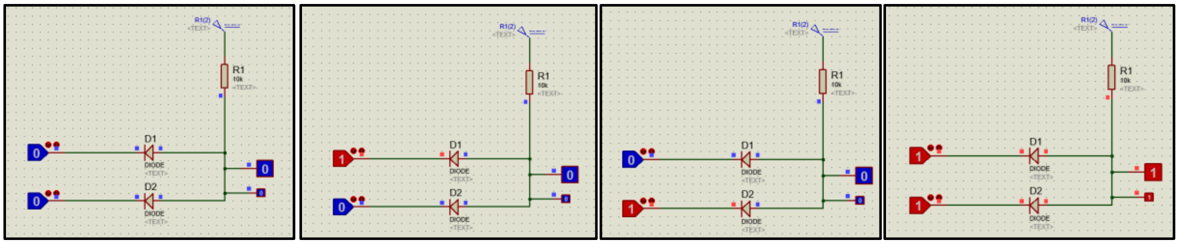


Fig. 11: RTL AND gate

At first, the AND gate was designed using RTL (Fig.11). However, when any of the inputs is 0, the diode conducts current to the component connected before the AND gate. This caused problems when implemented on Proteus simulator. Consequently, the AND gate is implemented using TTL. The output of the AND gate is 1 when both inputs are 1 and 0 otherwise (Fig.12). Only when both inputs of the transistors are 1, the output at the collector is approximately $2 * V_{CE\ saturation}$ (low level output). The third transistor is used to invert this output to high level which matches the truth table of the AND gate.

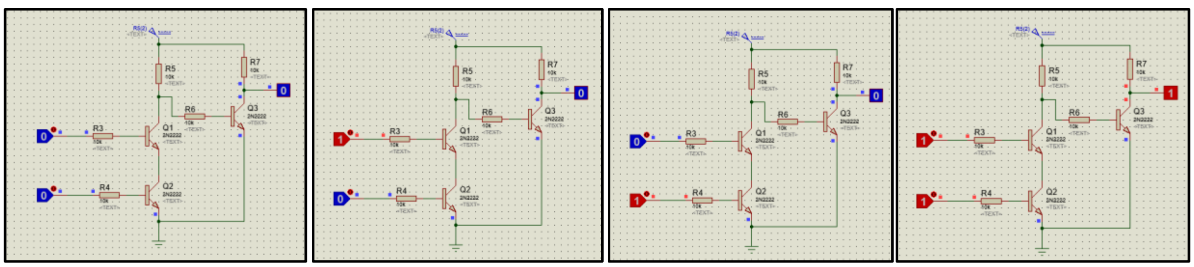


Fig. 12: TTL AND gate

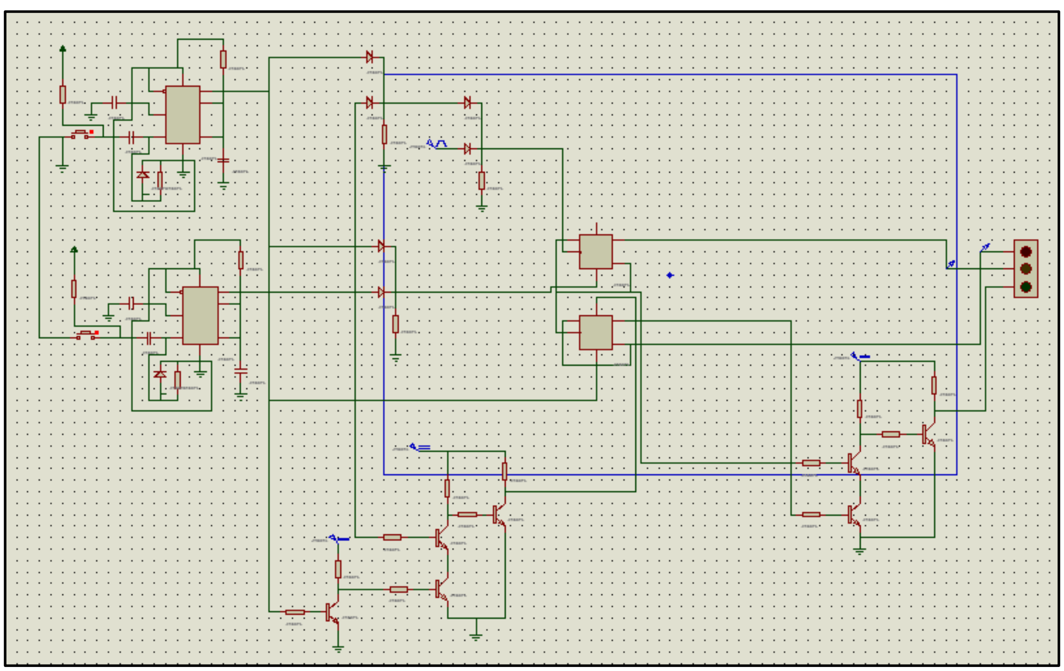


Fig. 13: Full circuit with gates

5.3 D Flip-flop

The D flip-flop block diagram is shown in Fig.14. As long as the clock input is low, changes at the D input make no difference to the outputs. Provided that the CK input is high (at logic 1), then whichever logic state is at D will appear at output Q. The set forces the output to 1 and the reset forces the output to 0. (Fig.15, 16, 17 and 18)

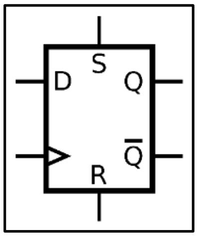


Fig. 14: D flip-flop

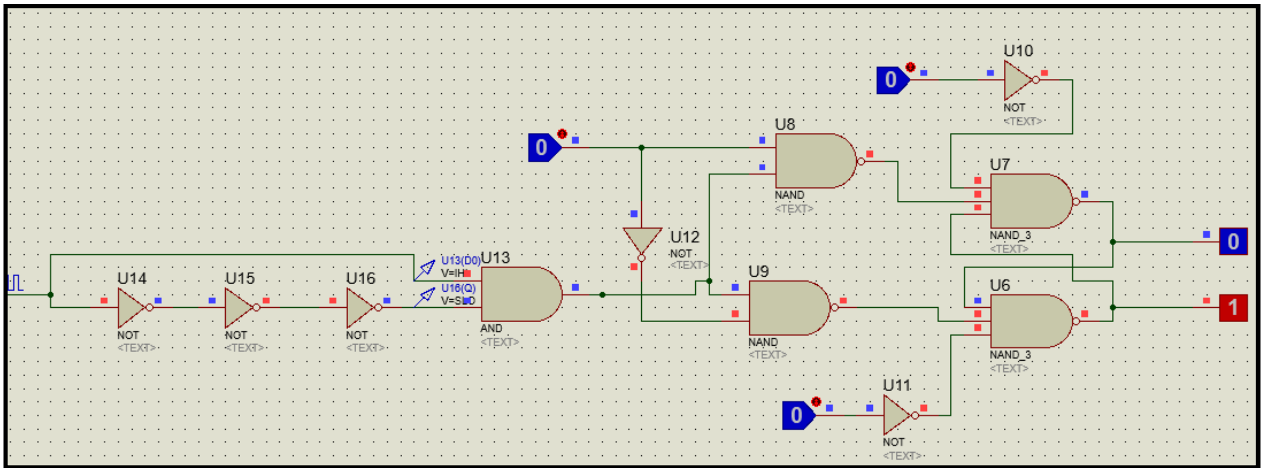


Fig. 15: D flip-flip (D=0)

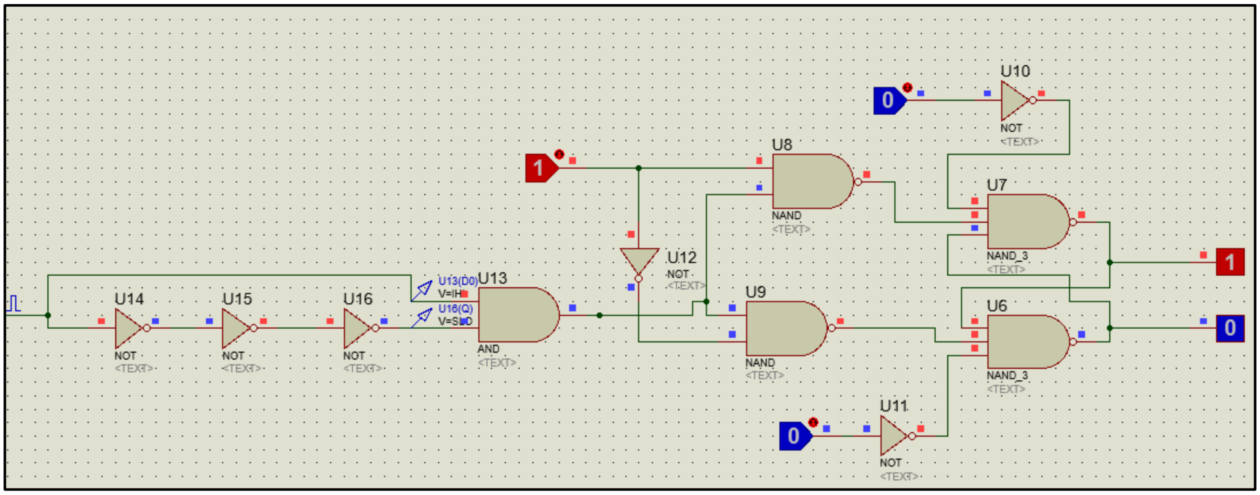


Fig. 16: D flip-flop (D=1)

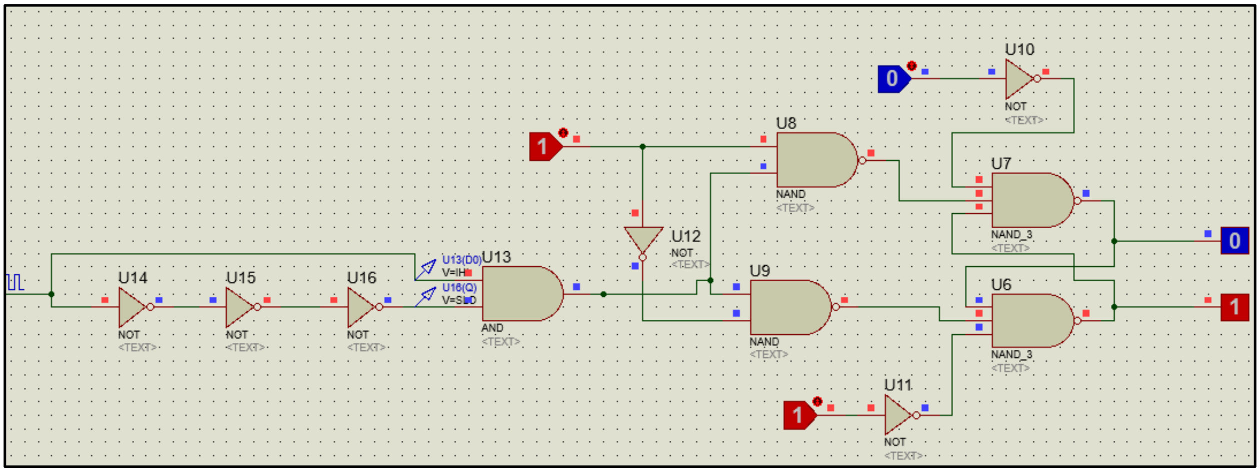


Fig. 17: D flip-flop (reset)

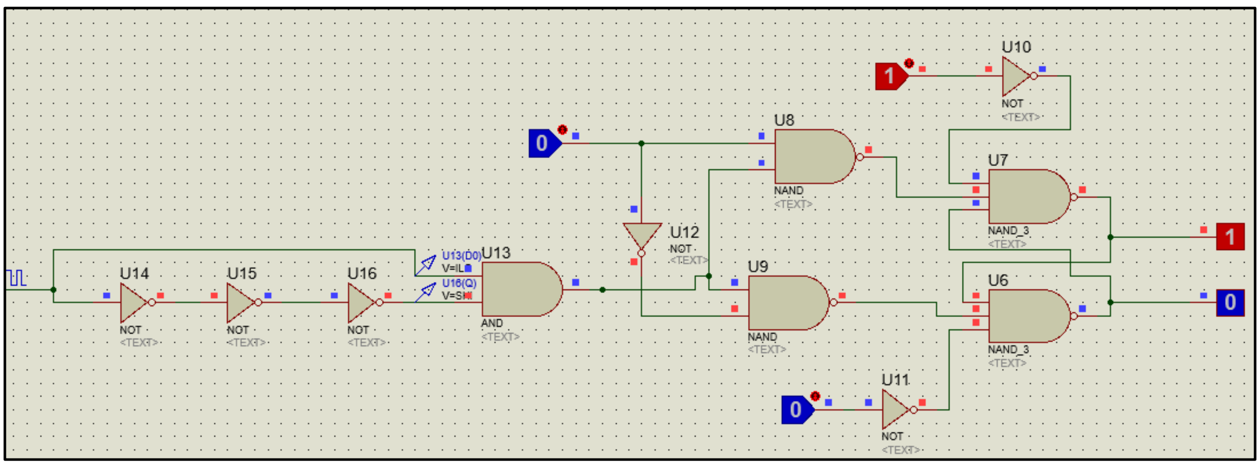


Fig. 18: D flip-flop (set)

The clock pulse applied to the flip-flop is reduced to a very narrow positive going clock pulse of only about 45ns duration, by using an AND gate and applying the clock pulse directly to input ‘a’ but delaying its arrival at input ‘b’ by passing it through 3 inverters (Fig.19). This inverts the pulse and also delays it by three propagation delays. The AND gate therefore produces logic 1 at its output only for the 45ns when both ‘a’ and ‘b’ are at logic 1 after the rising edge of the clock pulse.

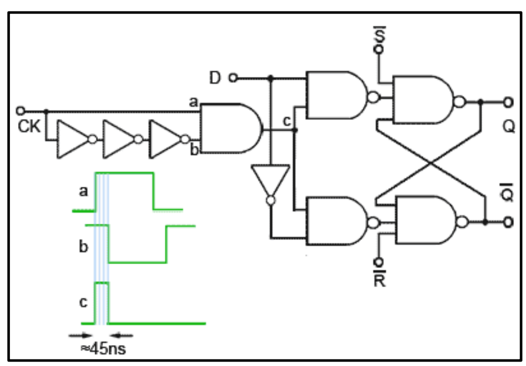


Fig. 19: pulse delay of D flip-flop

During the simulation of the flip-flop, three NOTs were not enough to produce a pulse that could drive the following gates. To solve this solution, 3 options were proposed: increase the number of NOTs, use an RC circuit to delay the signal followed by a Schmitt trigger to adjust the shape of the signal to a sharp square wave (Fig.20), or AND a monostable multivibrator with the input of the NOT. The first solution has proven to be simpler and costs less money, therefore, 5 NOTs are used instead of three (Fig.21).

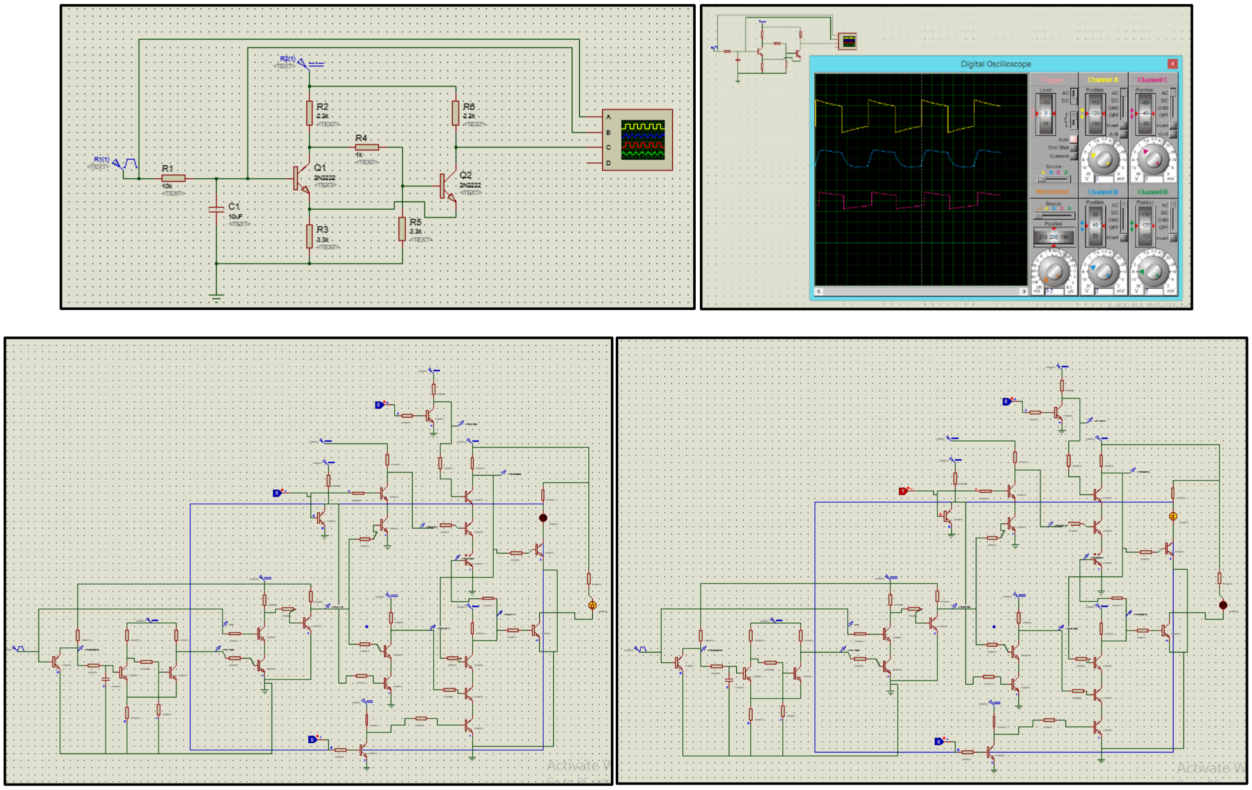


Fig. 20: D flip-flop with Schmitt trigger

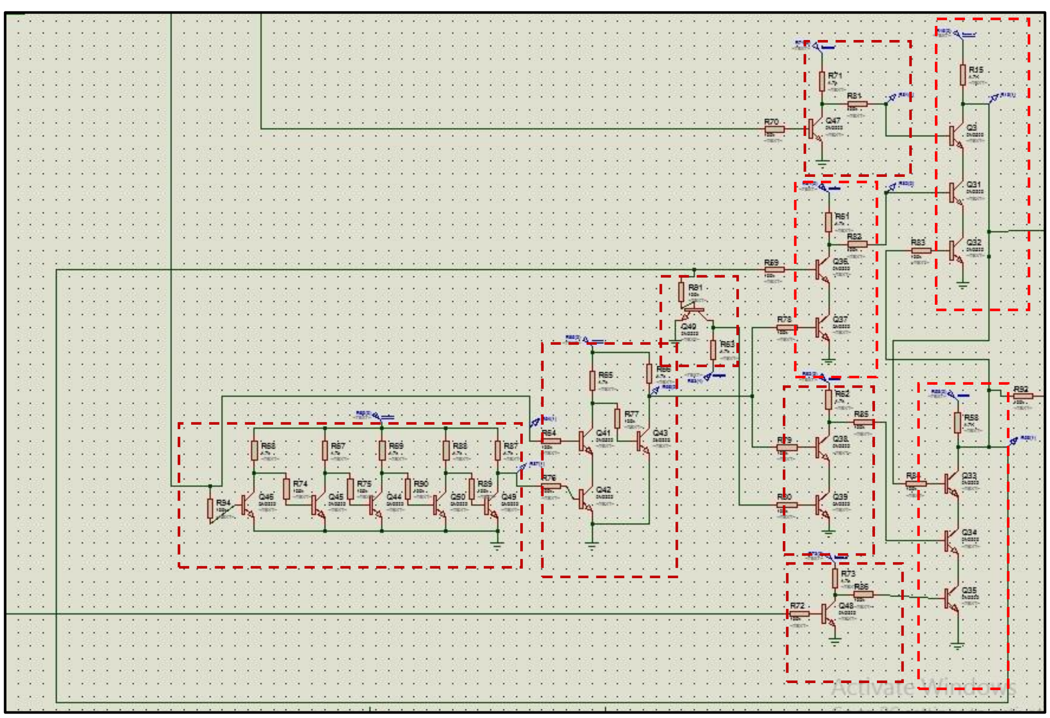


Fig. 21: D flip-flop with 5 NOTs

The simulation of the NAND gate was done first before integrating the whole flip-flop (Fig.22).

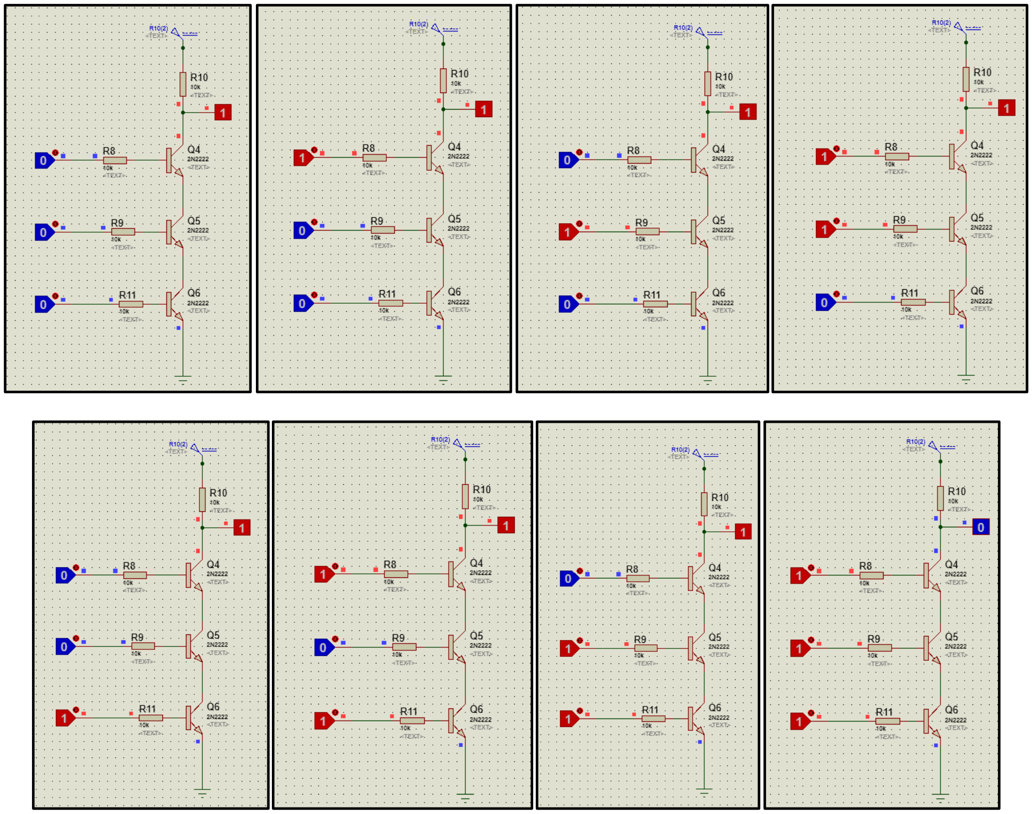


Fig. 22: NAND gate

The converted flip-flop is shown in Fig.23, 24, 25 and 26 with all of its cases.

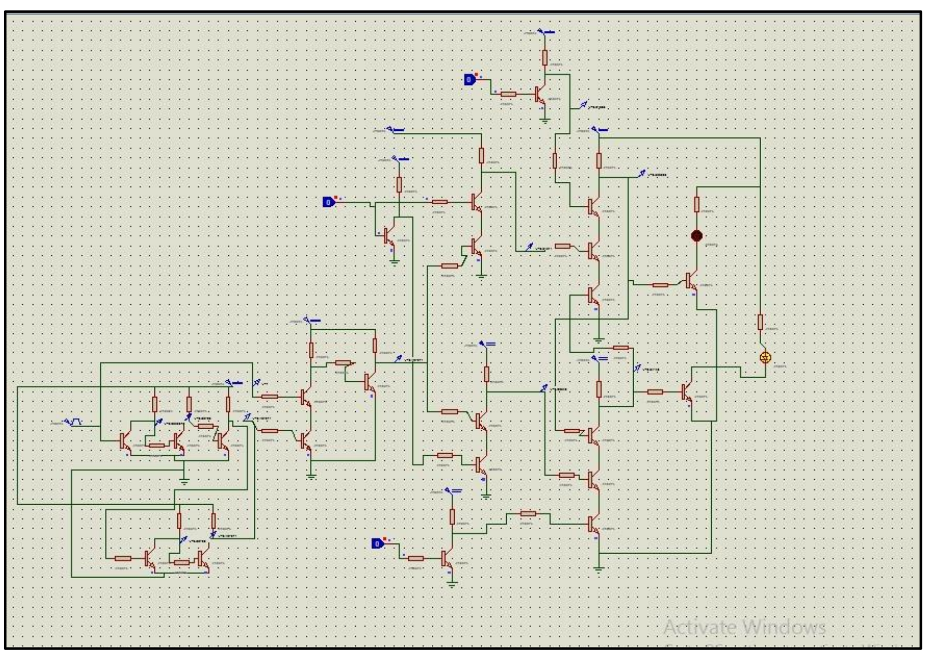


Fig. 23: D flip-flop with transistors (D=0)

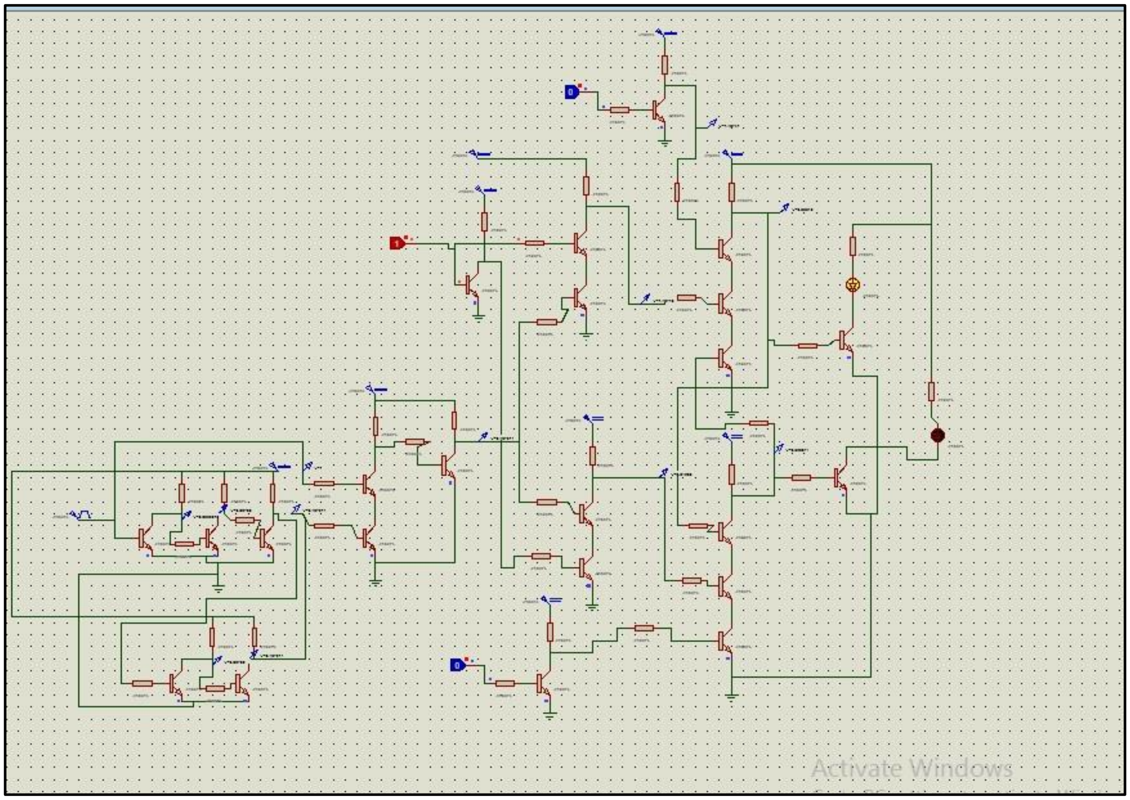


Fig. 24: D flip-flop with transistors (D=1)

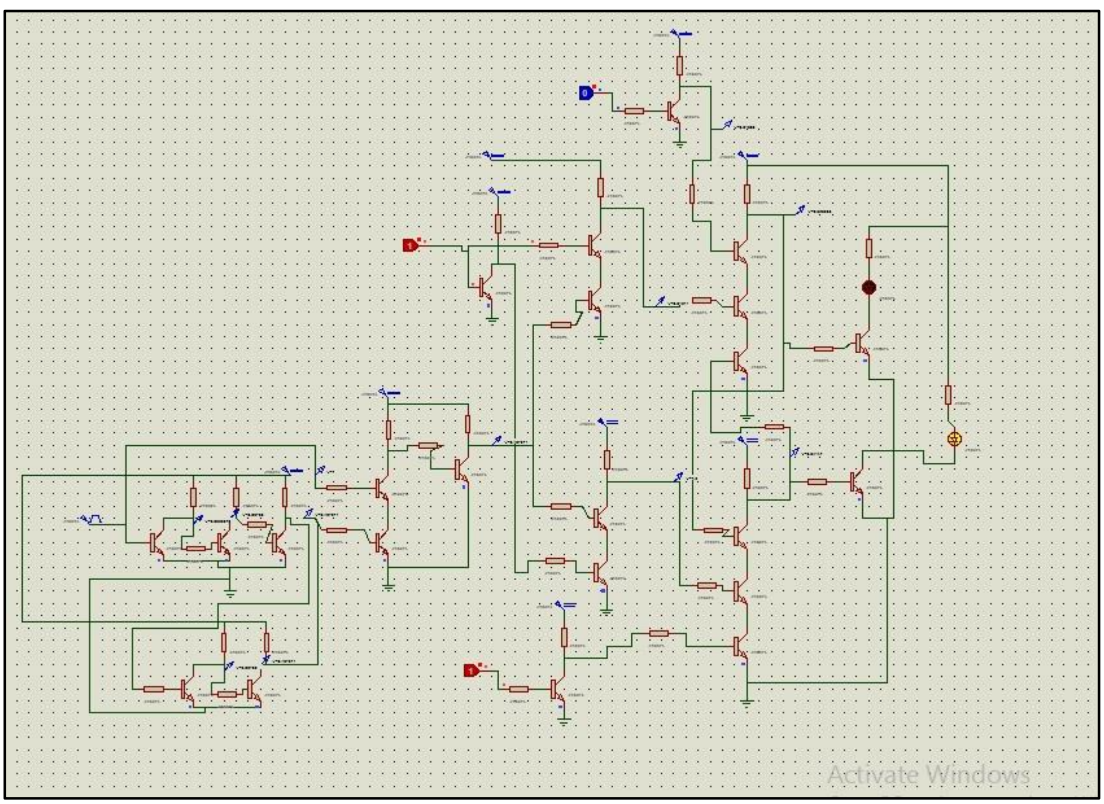


Fig. 25: D flip-flop with transistors (reset)

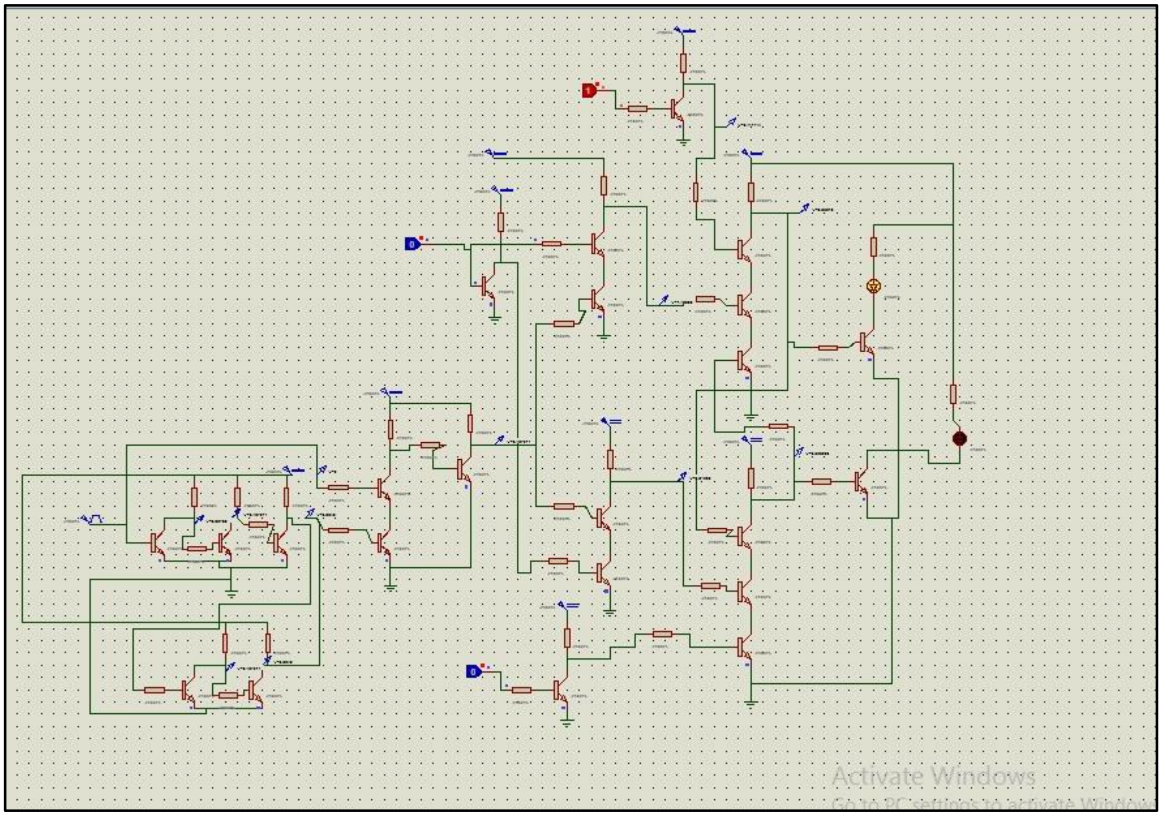


Fig. 26: D flip-flop with transistors (set)

5.4 Clock generator

Initially, the clock generator was implemented using 555 in astable mode. Fig.27 shows the astable connection. In the *astable* mode, the circuit will keep re-triggering itself, resulting in a pulse train. In this case, the capacitor charges and discharges between 1/3 V_{CC} and 2/3 V_{CC}. The 555's output is *high* while charging, and *low* while discharging (Fig.28). The capacitor charges and discharge at different rate—it has to charge through R_A and R_B, but it only discharges through R_B. Thus, the length of the output's highs and lows can be changed by adjusting these resistors. This is used to vary the time of the traffic lights through a variable resistance.

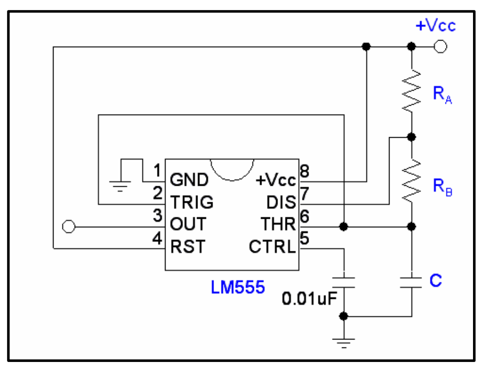


Fig. 27: Astable connection of 555 timer

The length of the high *output* is equal to:

$$t_{high} = 0.693(R_A + R_B)C$$

The length of the low *output* is equal to:

$$t_{low} = 0.693(R_B)C$$

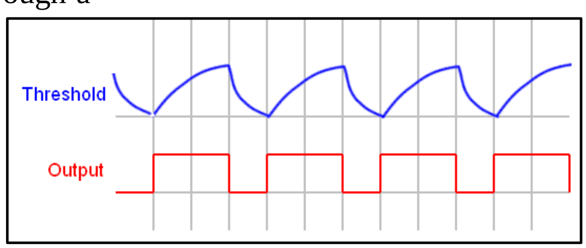


Fig. 28: Output graph

Fig.29 shows the Proteus simulation of the astable multivibrator using 555 timer.

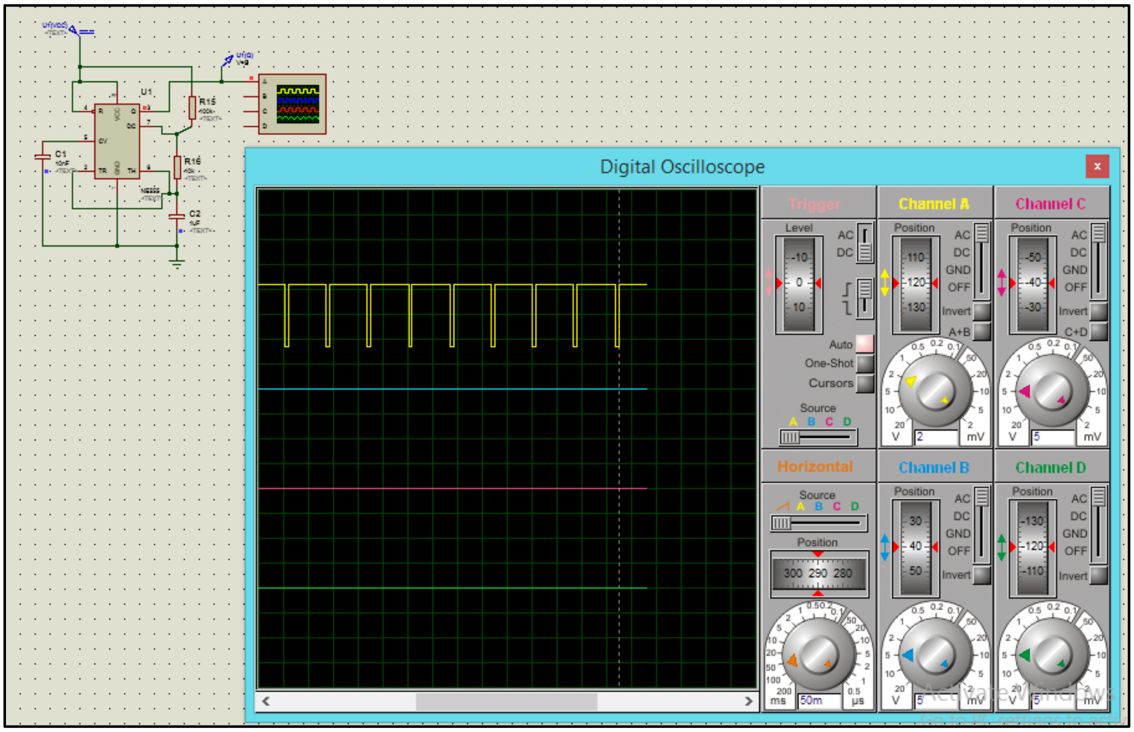


Fig. 29: Proteus simulation of astable

The transistor-level simulation of the astable multivibrator (Fig.30) does not work on the simulation programs because both transistors start at the same time. The astable multivibrator is directly implemented on a bred board.

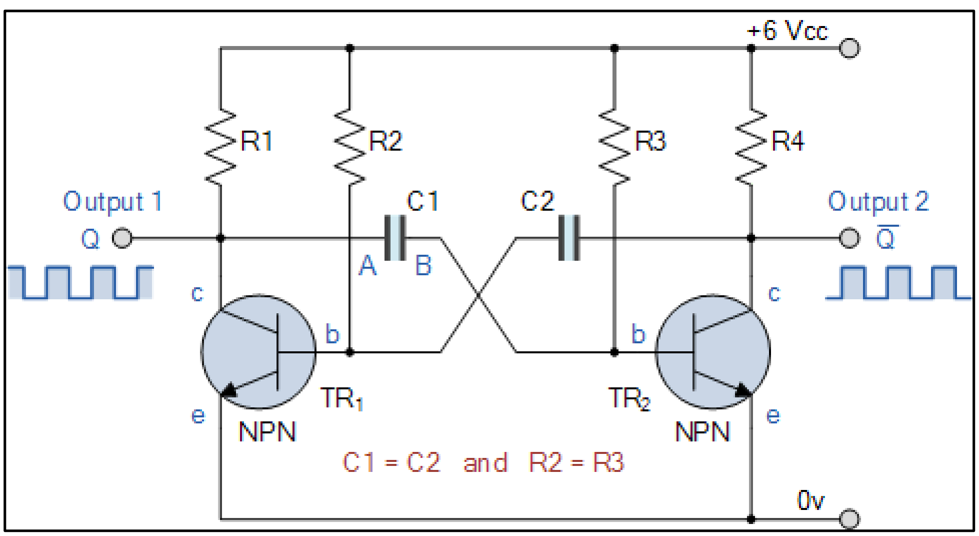


Fig. 30: Astable connection with transistors

5.5 Pulse generator

This circuit is implemented using 555 timer in the monostable mode to generate a pulse when it receives a trigger. Connections for the 555 timer in monostable mode is shown in Fig.31.

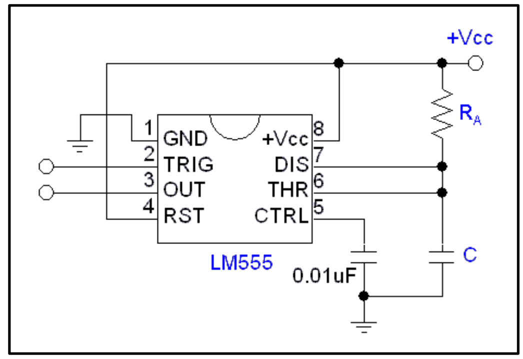


Fig. 31: Monostable connection

The circuit operates as follows:

1. In this circuit's initial condition, the capacitor *C* is held discharged through the *discharge* pin, which is grounded through the flip-flop in the timer. The *threshold* voltage is equal to the voltage across the capacitor.
2. When the *trigger* pin receives a negative trigger pulse less than 1/3 V_{CC}, the flip-flop sets the *output* to high and disconnects the *discharge* pin from the ground. This allows the capacitor to charge until the voltage across it reaches 2/3 V_{CC}, which takes about **$t=1.1R_AC$ seconds**.
3. When the *threshold* voltage reaches 2/3 V_{CC}, the flip-flop resets, connecting *discharge* to the ground and setting *output* to *low*. It is now back in the initial state, and awaits another trigger pulse. (Fig.32)

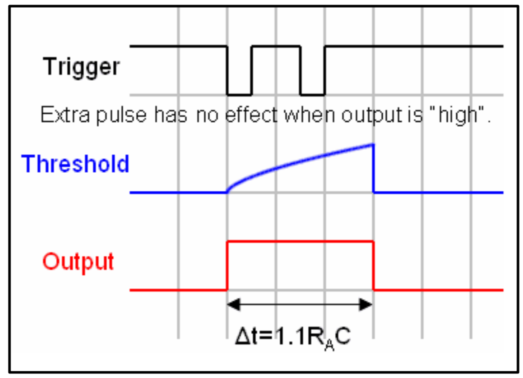


Fig. 32: Monostable output

By selecting the resistor and capacitor, the length of the output pulse can be controlled. If the trigger receives a signal while the output is still *high*, there is no effect. This feature is used to vary the time of the traffic lights after switching for emergency. Fig.33 shows the simulation of the monostable circuit on Proteus.

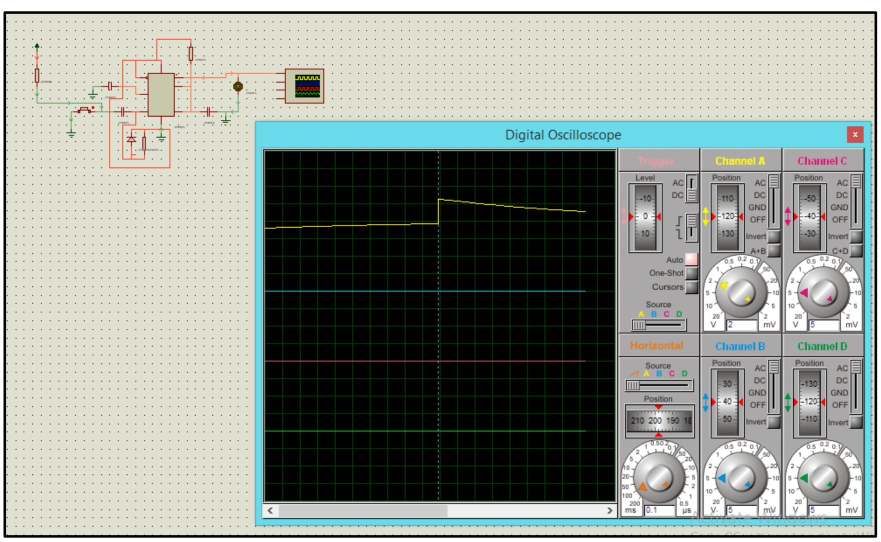


Fig. 33: Monostable on proteus

6. System Integration

6.1 Transistors biasing

During the simulation process, several values of transistor biasing resistances were tried. At first, 4.7KOhm resistances were used to bias the collector and base of each transistor. However, when several transistors were connected in a train, these values of resistances caused a decay in the signal and the collector of the transistor couldn't operate the following one. Accordingly, the value of resistance of output collector must be smaller than the value of resistance of the input base. Values of 4.7KOhm and 100KOhm are picked for the collectors and bases respectively. These values have proven to be well-functioning and fixed the problem.

6.2 Signal drop problems

After testing each module separately, the circuit was integrated all together. This produced many errors due to the effect of each circuit on the next one. This occurred when the output of a transistor braches in to many inputs or when the signal decays in a stage such as: the output of the monostable 555 timer responsible for the Red light, and the 3-input NAND gate of the Q' which feeds back the input D. To avoid this problem, 2 transistors were placed between the output and the inputs to amplify the signal without inverting it.

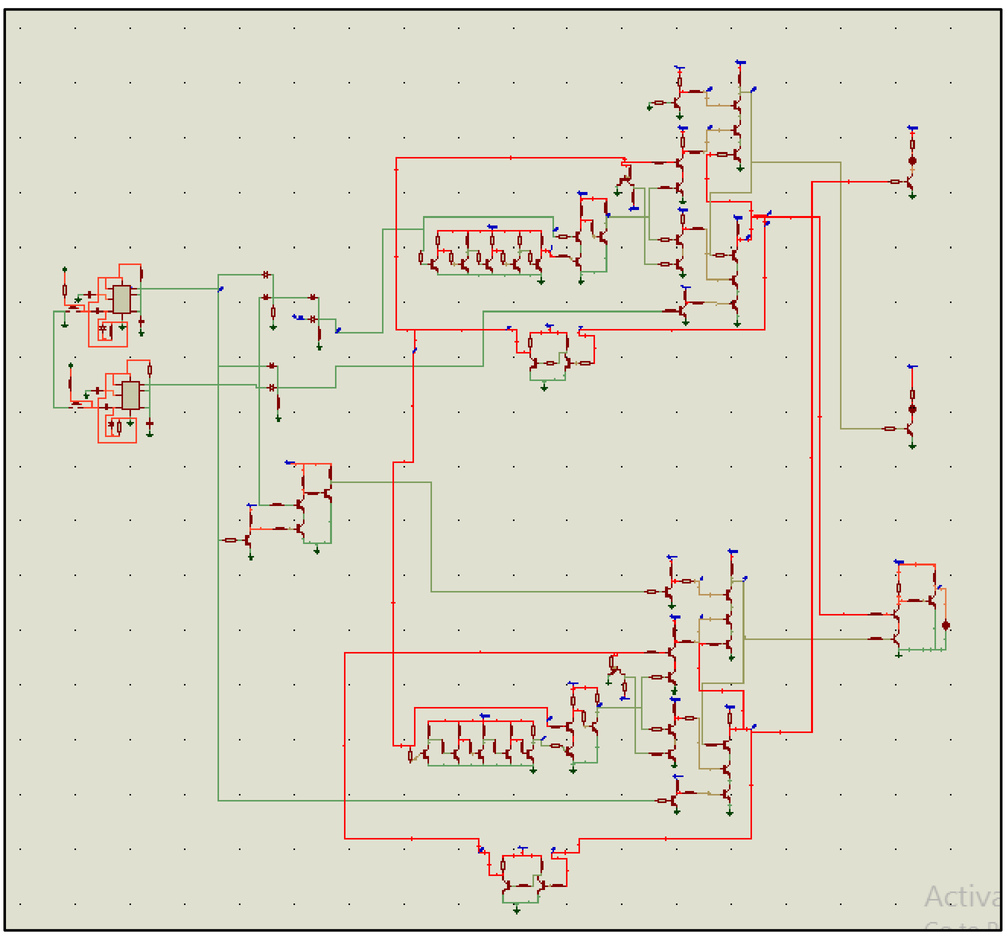


Fig. 34: full system red ON

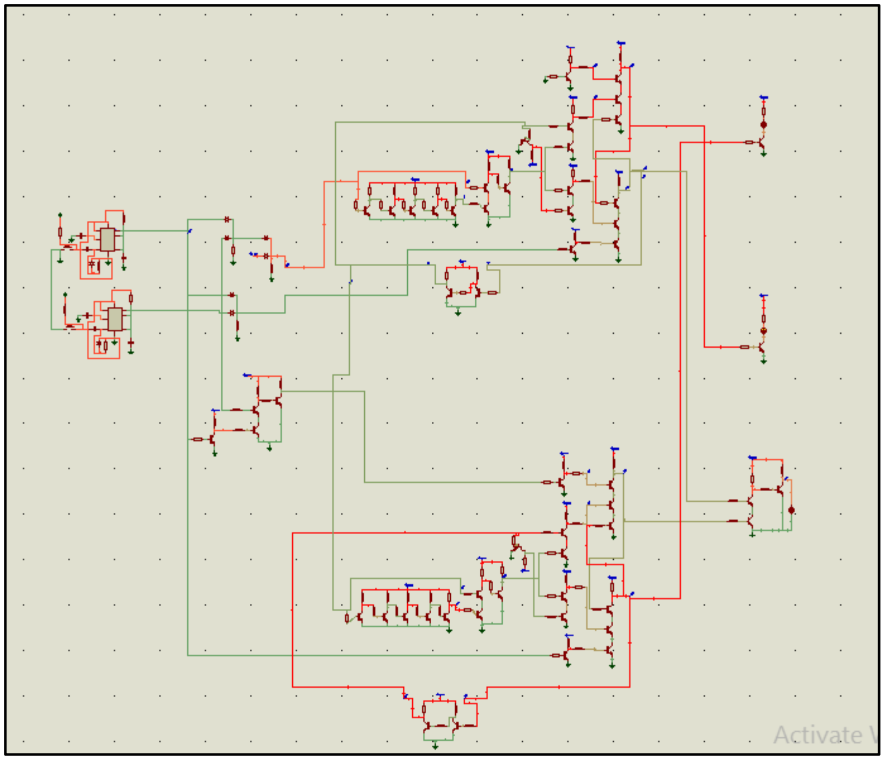


Fig. 35: full system red & yellow ON

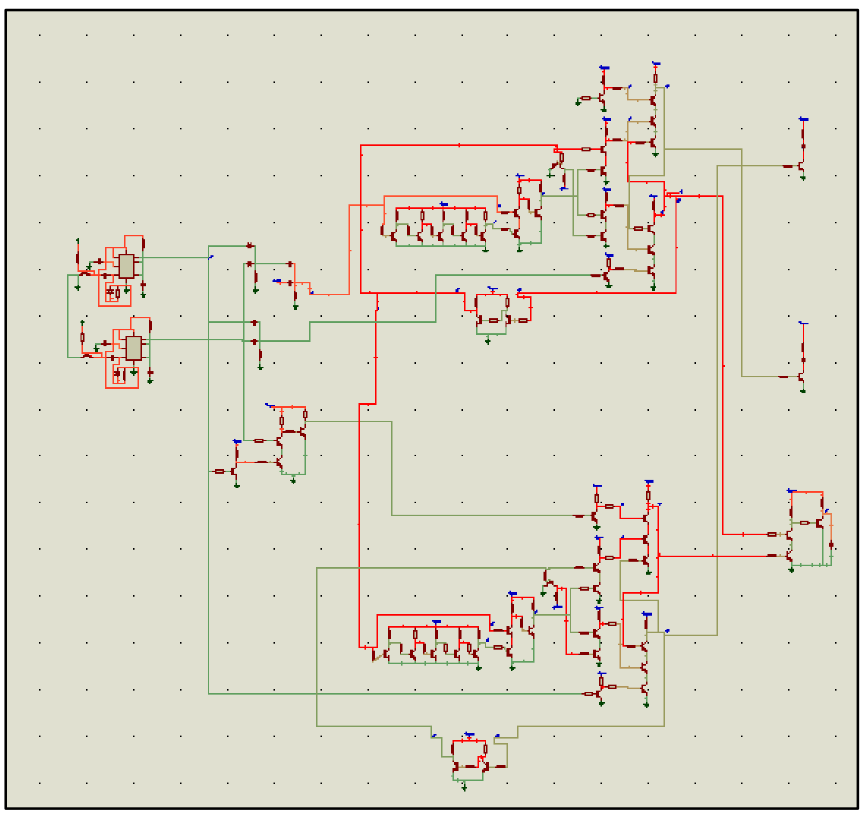


Fig. 36: full system green ON

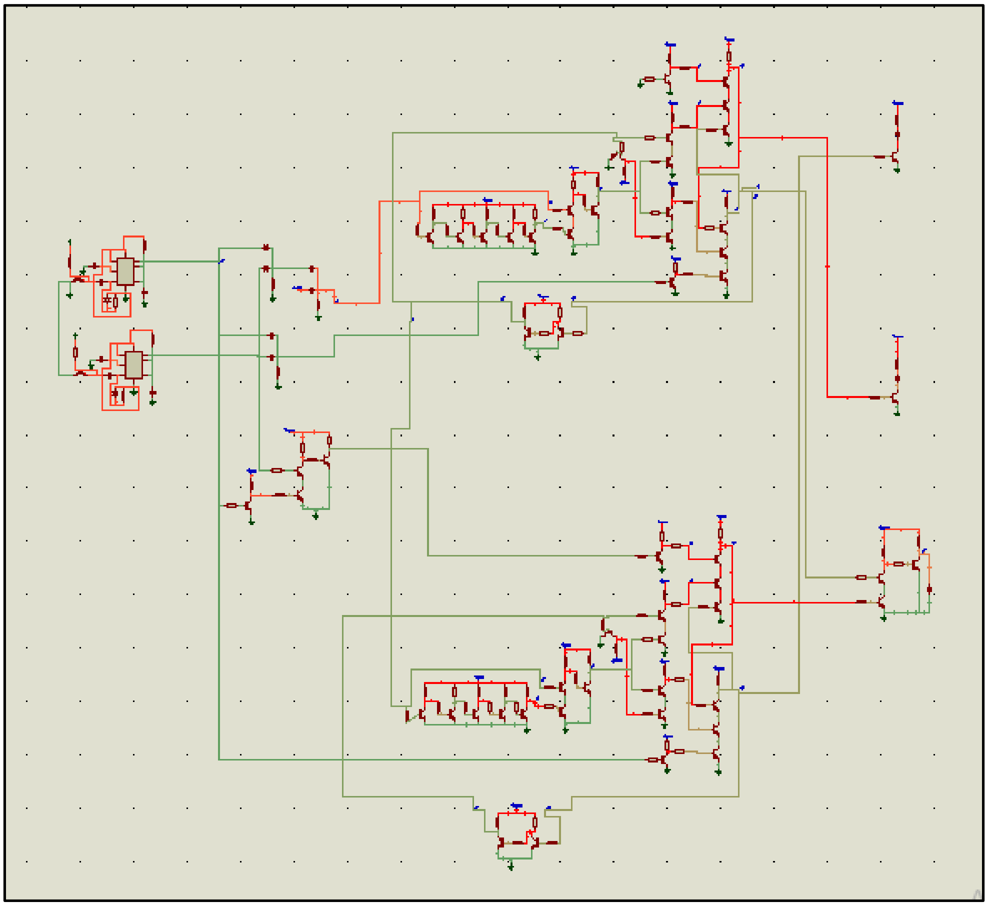


Fig. 37: full system yellow ON

7. Practical Implementation

After successfully integrating the circuit all together using Proteus simulator, a real prototype was connected on a bred board. Fig.38 shows the bred board in its final stage. A rectifier bridge is connected using 4 silicon diodes to ensure power connection safety. A 5V adapter is used to power up the circuit. The circuit is built using npn 2N2222 transistor due to its availability and low cost. Resistors of values 100KΩ and 4.7KΩ are used to bias each transistor's base and collector respectively. Resistances of 1KΩ or 270Ω are used to limit the current through the LEDs. For the astable multivibrator, a capacitor of 1uF and 2 resistances of values 1MΩ are used to produce a high time of 1.4s and low time of 0.7s. For the monostable multivibrator, a capacitor of 1uF and a resistance of 1.5MΩ are used to produce a pulse width of 1.7s. Therefore, a variable resistance in the range of Mega Ohms is required for each multivibrator (astable and monostable) to produce a noticeable time duration for testing.

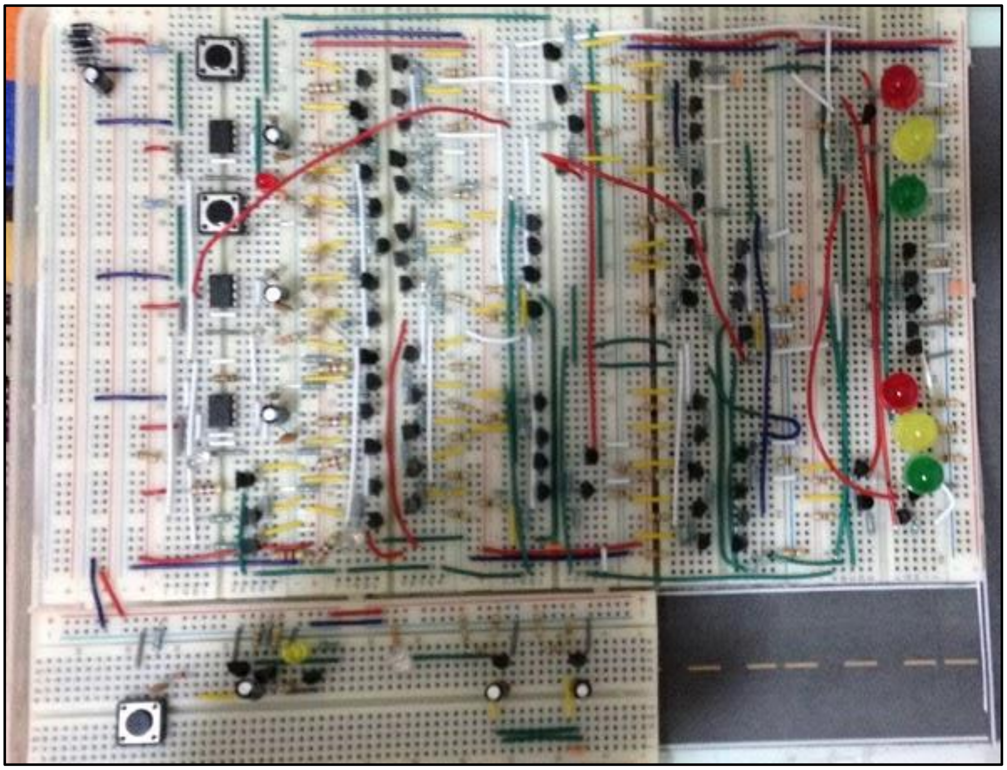


Fig. 38: practical prototype

A last-minute modification is done so that the traffic lights of the 2 sets switch more realistically. It is required that when:

- Red light of first is on, green light of second is on.
- Red and yellow lights of first are on, yellow light of second is on.
- Green light of first is on, red light of second is on.
- Yellow light of first is on, red and yellow lights of second are on.

To fulfill these conditions, the red LED of the second set is connected to the result of OR-ing of its original input and the yellow case, and the green LED is connected to the result of AND-ing of its original input and inverted yellow. The sequence of switching is shown in the Appendix.

The new expressions for the second traffic light set are:

Let the output of flip-flop 1 = Q1 and output of flip-flop 2 = Q2.

$\therefore$ Red2 = Q2.Q1'+Q2 = Q2, Yellow2 = Q1, and Green = Q1'Q2'.

8. Cost Analysis

The following table includes the cost of all the components used.

Component Name	Price/Item (LE)	Quantity	Total Price (LE)
2N2222 Transistor	0.25	60	15
LEDs	0.25	6	3
Push buttons	0.5	3	1.5
555 Timer	2	3	6
Diodes	0.1	10	1
Resistances (4.7K, 100K, 10K, 1K)	0.1	90	9
Capacitors (1u, 10n, 100n)	0.1	10	1
Total Price (LE)	36.5		

If the circuit is implemented using ready-made ICs, the following components are used:

- One 7408 (for AND gates)
- One 7432 (for OR gates)
- One 7474 (for D flip-flops)
- Three 555 timers (for clock and pulse generation)

The total estimated cost of this ready-made circuit is around 12 Egyptian pounds, which is much cheaper than the transistor-level implementation solution.

9. Conclusion

This report presented a simple solution for a traffic light system with a low cost and good functionality.

10. Future Work

In this circuit, the input time duration of all the LEDs is varied by a single variable resistance, so, all the LEDs must have equal switching times unless the officer presses the interrupt button for a certain color. As a future work, it is intended to modify the circuit to enable the variation of the ON-duration of each LED separately.

11. References

[1] http://myelectronicshub.blogspot.com.eg/2014/07/making-traffic-lights-using-555-ic.html

[2] http://hades.mech.northwestern.edu/index.php/555_Timer

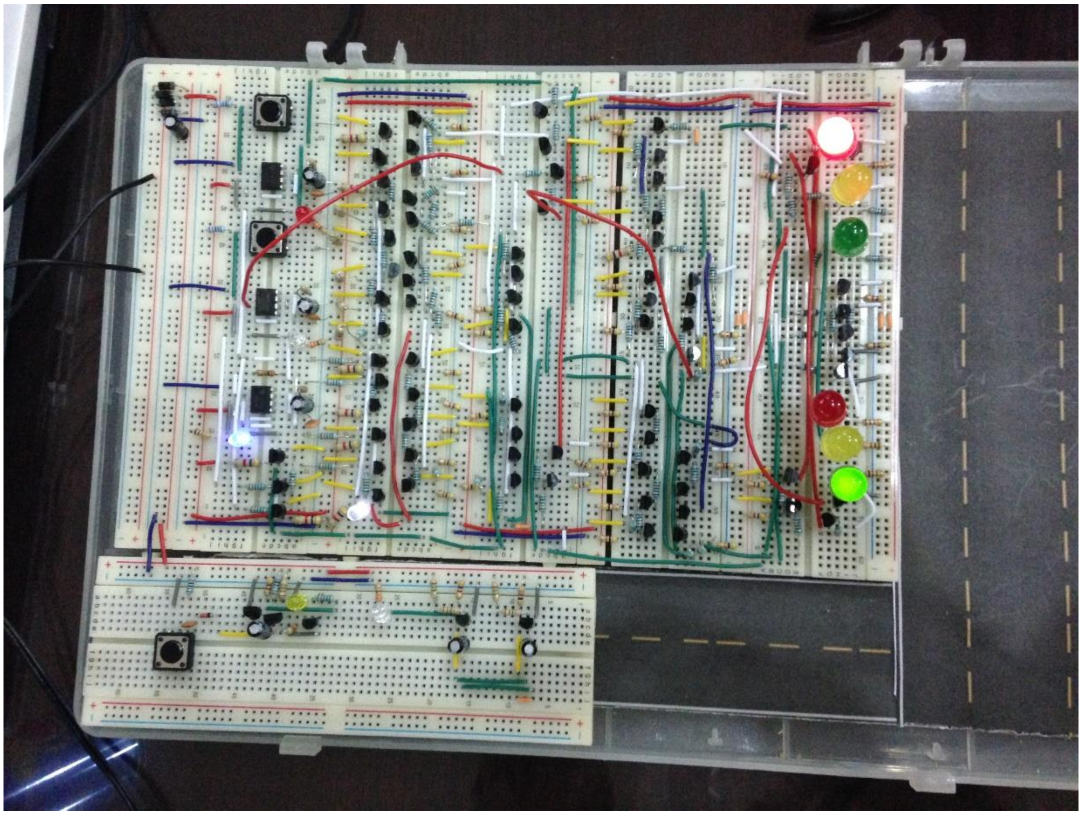
[3] https://en.wikipedia.org/wiki/Flip-flop_(electronics)

[4] http://www.learnabout-electronics.org/Digital/dig53.php

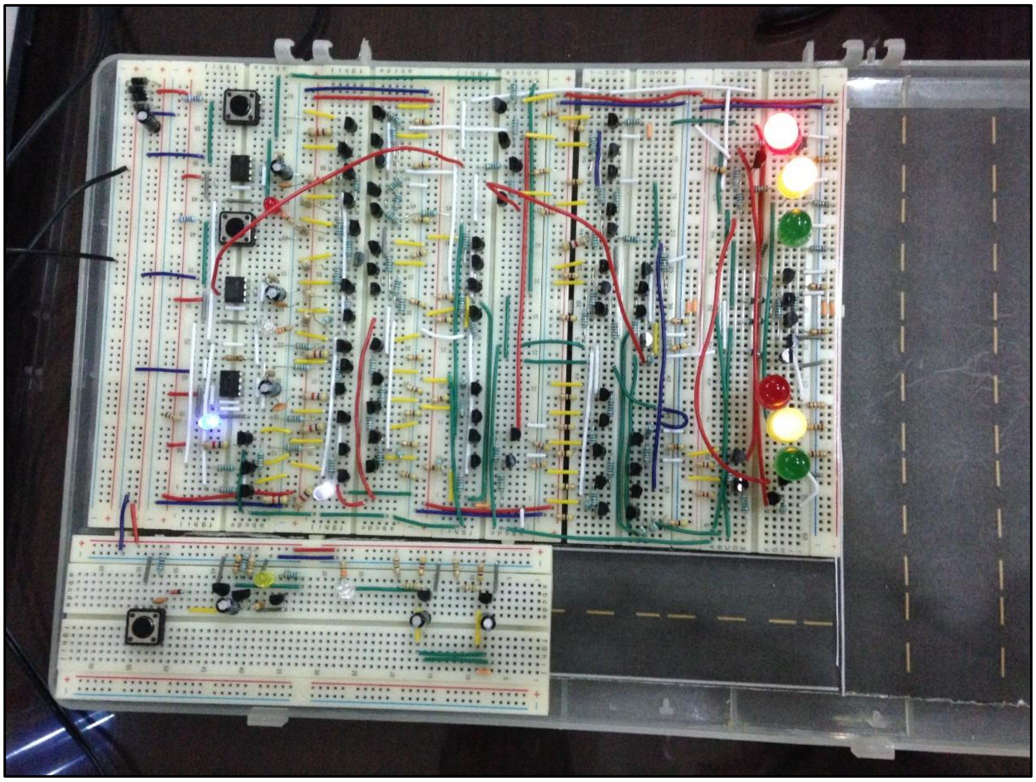
[5] http://www.electricaltechnology.org/2014/10/traffic-light-control-electronic-project.html

[6] http://www.radio-electronics.com/info/circuits/transistor/circuit-configurations.php

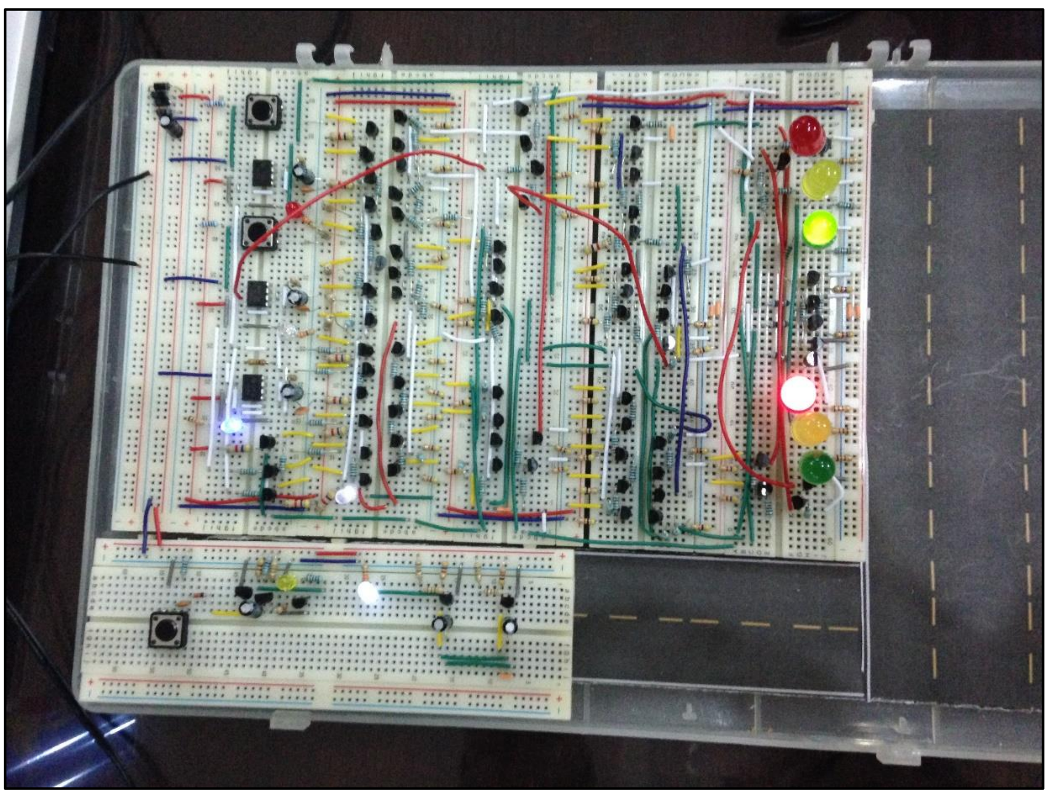
12. Appendix



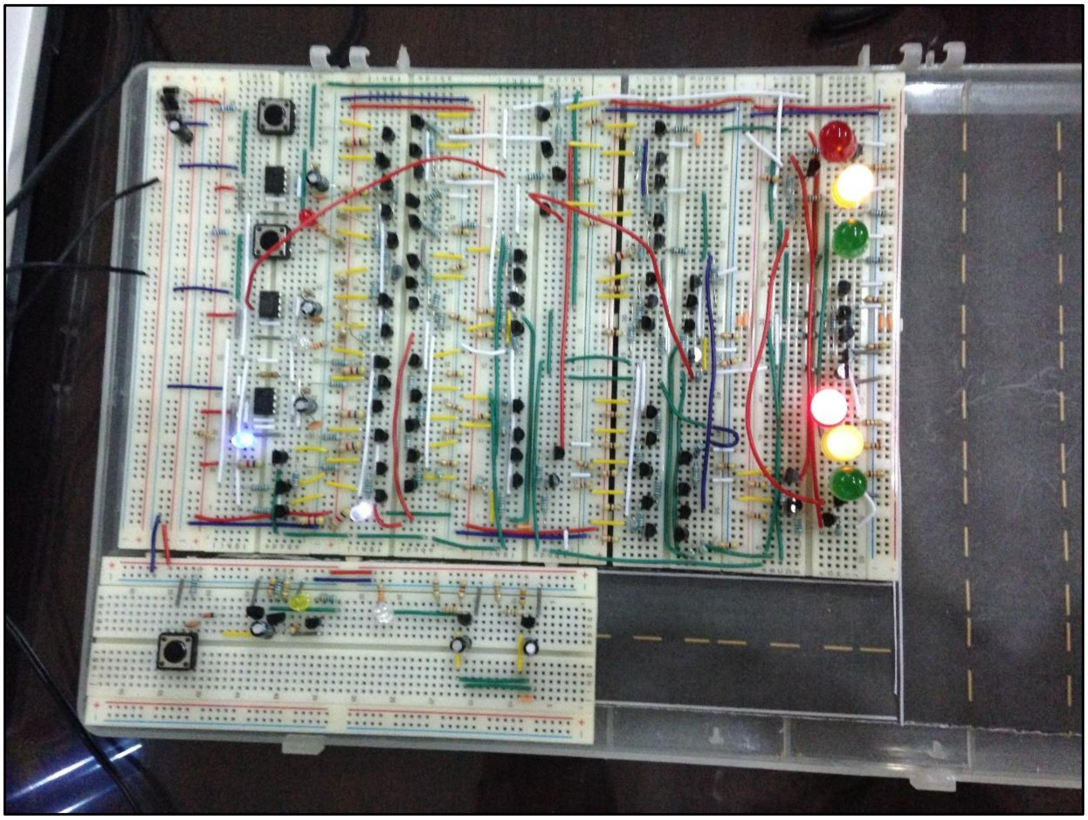
First: Red on, Second: Green on



First: Red and yellow on, Second: Yellow on



First: Green on, Second: Red on



First: Yellow on, Second: Red and yellow