IIT ROORKEE MOTORSPORTS

Project Report

Simulate a continuously looping Traffic Light using 555 Timer and Design a PCB for the circuit

Submitted By:

PRIYANGSI DEY

2ND YEAR ELECTRICAL

JUL 2021 – AUG 2021

Mentor:

ADITYA SHARMA, 3RD YEAR ELECTRICAL

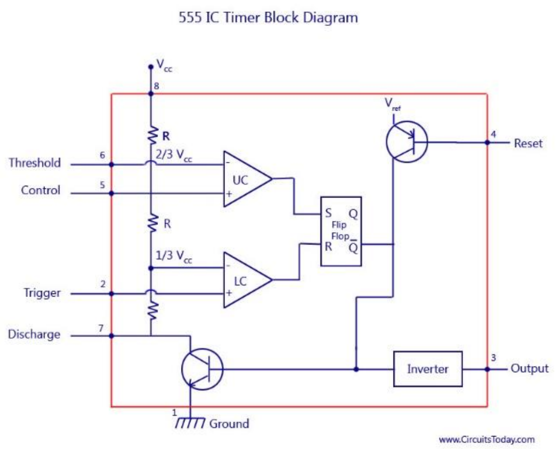
ROSHAN RAJA SAMUEL A., 4TH YEAR ELECTRICAL

Contents:

1. Introduction-

Traffic Lights are used to control the vehicular traffic. In today's times, everyone has different types of vehicles resulting in rise to the numbers of vehicles. That's why traffic lights are mandatory to avoid unnecessary traffic jams and accidents. There are three lights in the traffic signal, having different message for the drivers. Red light asks the driver to stop at the intersection, green light gives the driver free license to drive through the intersection whereas the yellow light alerts the driver to wait if the next light is red one or get ready to go if the green light is next.
One of the most important components in this project is 555 timer. We will need it's monostable mode of operation in this circuit.

1.1. Working Principle Of 555 Timer-



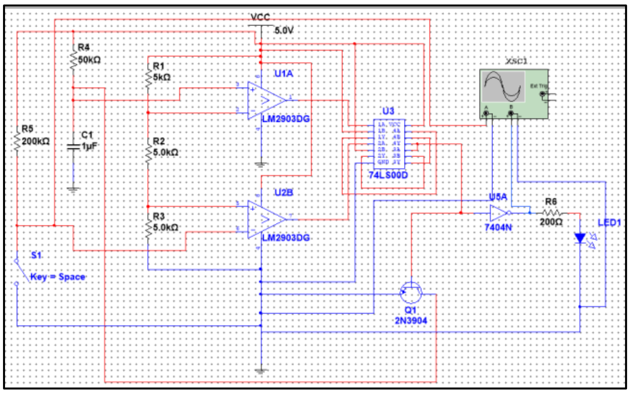
The internal resistors act as a voltage divider network, providing (2/3)Vcc at the non-inverting terminal of the upper comparator and (1/3)Vcc at the inverting terminal of the lower comparator. In most applications, the control input is not used, so that the control voltage equals +(2/3) Vcc. Upper comparator has a threshold input (pin 6) and a control input (pin 5). Output of the upper comparator is applied to set (S) input of the flip-flop. Whenever the threshold voltage exceeds the control voltage, the upper comparator will set the flip-flop and its output is high.
A high output from the flip-flop when given to the base of the discharge transistor saturates it and thus discharges the transistor that is connected externally to the discharge pin 7. The complementary signal out of the flip-flop goes to pin 3, the output. The output available at pin 3 is low. These conditions will prevail until lower comparator triggers the flip-flop. Even if the voltage at the threshold input falls below (2/3) V_{CC}, that is upper comparator cannot cause the flip-flop to change again. It means that the upper comparator can only force the flip-flop's output high.
To change the output of flip-flop to low, the voltage at the trigger input must fall below + (1/3) Vcc. When this occurs, lower comparator triggers the flip-flop, forcing its output low. The low output from the flip-flop turns the discharge transistor off and forces the power amplifier to output a high. These conditions will continue independent of the voltage on the trigger input. Lower comparator can only cause the flip-flop to output low.
From the above discussion, it is concluded that for the having low output from the timer 555, the voltage on the threshold input must exceed the control voltage or + (2/3) Vcc. This also turns the discharge transistor on. To force the output from the timer high, the voltage on the trigger input must drop below +(1/3) Vcc. This turns the discharge transistor off.
A voltage may be applied to the control input to change the levels at which the switching occurs. When not in use, a capacitor should be connected between pin 5 and ground to prevent noise coupled onto this pin from causing false triggering.
Connecting the reset (pin 4) to a logic low will place a high on the output of flip-flop. The discharge transistor will go on and the power amplifier will output a low. This condition will continue until reset is taken high. This allows the resetting of the circuit's operation. When not in use, reset should be tied to +Vcc.

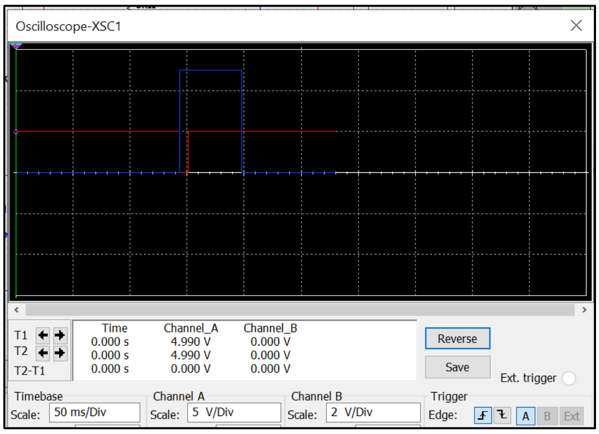
2. Objective-

To understand and simulate internal circuitry of 555 timer IC by utilising basic digital/analog components like Flip-Flops, Comparators, etc. And then simulating a continuously looping RYG traffic light on multisim using 555 timers. Finally designing a PCB on Altium for the above traffic light circuit.

3. 555 Timer Simulation-

First we went through the videos and the material provided in the project document and started 555 timer simulation in multisim. We studied about flip flop and how it is being used in the 555 timer. My first attempt didn't give the desired output since the output stage was connected to Q rather than notQ. After correcting the connection, the timer simulation was completed by giving it a trigger using a simple switch. But it had to be manually operated so there was a problem. If the time of the switch being in the on state exceeds the time capacitor takes to charge upto *(2/3)*Vcc*, then the capacitor for some time will get charged a bit higher which will lead the threshold value go lower and hence the inputs to the flip flop would become 1 this will make the output invalid and hence oscillations occur. So basically a system was required so that the trigger should be there for a very small amount of time, this will lead the output to go 1 hence the led glows and remains glowing till the capacitor charges to (2/3)*Vcc.

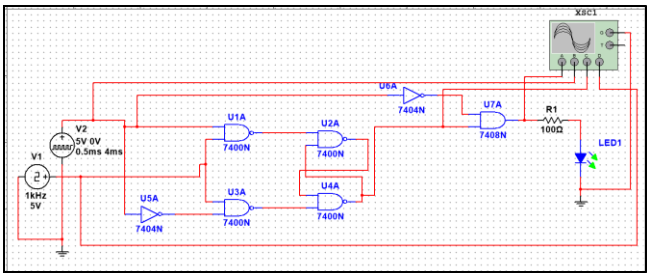




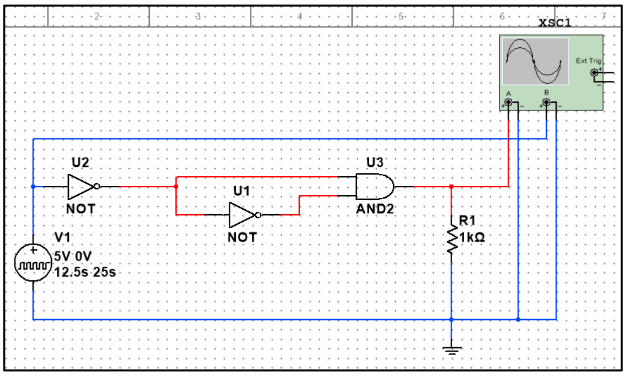
Hence, we started reading about the square wave edge detector which would act as a trigger for the timer so that we will not have to operate it manually.

4. Square wave edge detector-

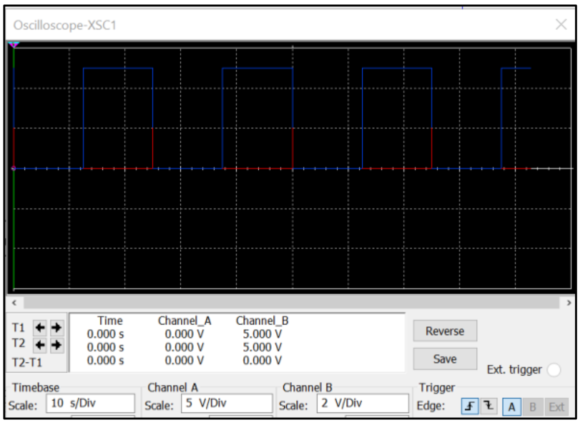
Learnt to use the Timer wizard which helps to create a timer IC in multisim where one can input the desired parameters. Eventually this is what was supposed to be used since we had to put 3 timers(and not just 1) for switching three lights- red, yellow and green. Now there was a need for a trigger which would switch it on and off. For this we read about square wave edge detector which would detect whenever the input goes from high to low(negative edge detector). In my first attempt I used a D-flip flop to create a delay of 1 clock cycle. The not of the input signal went as one of the inputs of the flip flop and the other input was the signal only. The output of this had a time delay of one clock signal. Now this output and the output of the not gate were given as input to AND gate and the final output was a spike whenever the input went from high to low. But this circuit required an external clock signal and moreover this also needed so many gates, so this circuit was scrapped-



A better circuit was to simply use 2 not gates and an AND gate. The not gates were enough to produce the required delay-



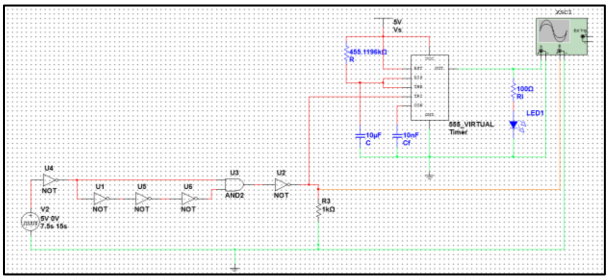
This was the final Falling edge detector used.

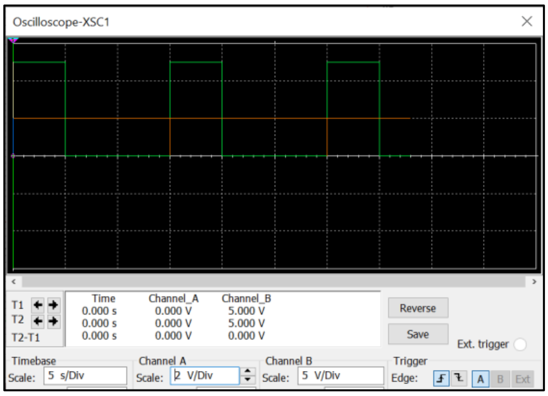


Output

5. Using the edge detector as a trigger-

Now this detector had to be used as a trigger to the timer. Whenever the trigger goes low the output is high and the led glows so initially we need the trigger to be high so that led is off and then the trigger should go low for a very short period of time which would make the output high and will remain so until the capacitor charges to (2/3)VCC. If we see, the output of the edge detector is such that it is initially low and becomes high(spike) whenever the input falls low from high but what we want is that initially the trigger should be high therefore the output of the detector has to be high so an extra not gate was required to be added. However, when this was given as a trigger to the timer, unexpectedly the output never went high! Further speculations and discussions led to the fact that more not gates have to be used to increase the delay so three not gates were added in series instead of one-





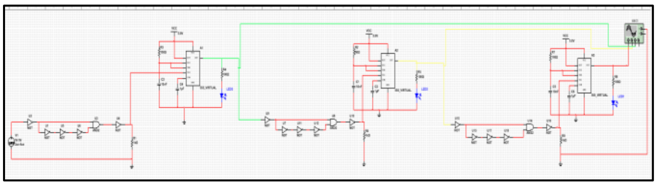
Now I wanted the led to glow for 5secs so I had to set the values of R and C such that the time for the capacitor to charge upto (2/3)VCC is equal to 5secs.

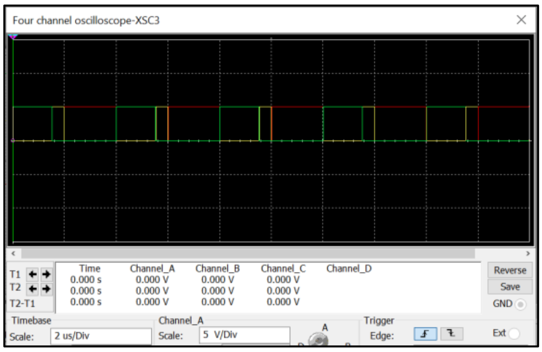
$Vc(t) = VCC*(1-e^{-t/RC})$ => $(2/3)VCC = VCC*(1-e^{-t/RC})$ => $e^{-t/RC} = 1/3$ => $t = R*C*ln(3) = 5secs$

If I set capacitance to 10 uF then I get R = 455.12 kohm

6. Simulating RYG LEDs sequentially-

Then finally three timers and three edge detectors were used altogether to operate it as a sequential circuit to simulate three leds - red, yellow and green(traffic light) – in sequence. Whenever the output of first timer goes low i.e led stops glowing, the 2nd detector detects the same and makes the 2nd timer's trigger to go low and hence the ouput high and the led glows. Same follows for the 3rd timer. But there was a problem- the simulation was so slow that we couldn't observe the changes properly. So we had to reduce the time, the capacitor takes to charge, to microseconds in order to view the simulation output. *And for this we reduced the capacitance to 10nF and the resistors – 150ohms, 50ohms and 195ohms. And the respective time for which the leds glowed- 1.65us, 0.55us and 2.14us*. Since we used a clock input, we had to ensure that the sum of the their time of glowing should be approximately equal to the time period of the clock signal so that every time the signal goes low till then all the three leds must have glowed only once. My first attempt-





6.1. Removing the Clock input signal-

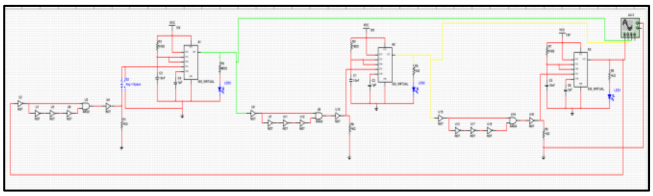
Now, as shown above, we used a clock input to operate the circuit, but this will again require a separate IC altogether and make the circuit even more complicated. So we were asked if somehow we could use only a switch so that once the switch is pushed the three leds glow in sequence in a loop and no clock input is used. Now this could be done if the output of the third timer was connected to the input of the 1st detector and by using a push button which would make the trigger low for a very short while and the circuit will continue working in sequence. Now we could set the time according to our will. But still the simulation was very slow, if a time of 10s, 2s and 10s was given. So we just scaled it down to microseconds so that the output can be seen properly.

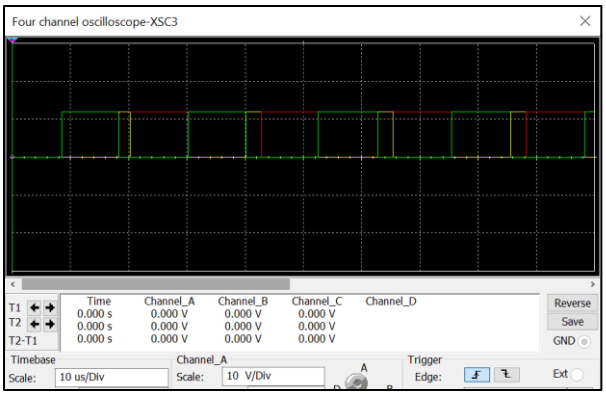
Final $R1*C1*ln(3) = 10s$ where $C1 = 10uF$ and therefore $R1 = 910kohms$.

$R2*C2*ln(3) = 2s$ where $C2 = 10uF$ and therefore $R2 = 182kohms$.

$R3*C3*ln(3) = 10s$ where $C3 = 10uF$ and therefore $R3 = 910kohms$.

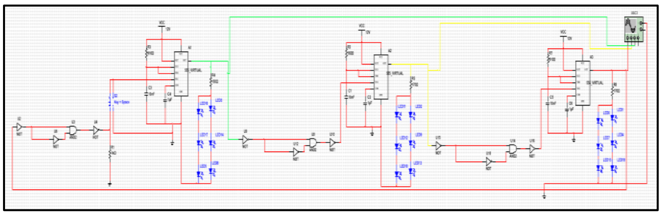
But for the simulation, Resistor values were scaled down to ohms and capacitances to nF so that the time of glowing becomes 10us for green led, 2us for yellow and 10us for red-



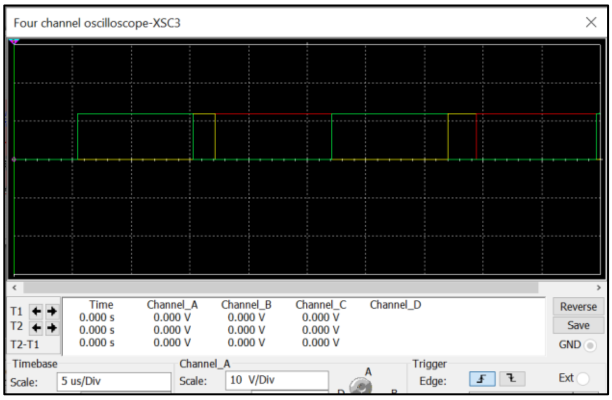


6.2. Using single NOT gate and adding multiple LEDs-

Now when checking if some of the components could be reduced, we tried to use one not gate rather than 3 in series(to create delay). Just thought to try and surprisingly it worked this time! Earlier when only one timer was there, one not gate didn't give the desired output but here it did! So we just deleted the extra not gates. Secondly I was told that the two pull down resistors were not required(the ones connected with timer2 and timer3's trigger) since the output of these detectors were directly going as input to the timers. The pull down resistors were not needed, it was only needed in the first case because when the switch goes to on position the detector part would become a floating component without pull down resistor. We do not want such components. Now we had to increase the number of leds per colour and accordingly choose the resistor in series so that enough current flows through the leds to turn them on. On current of the leds chosen was near 20mA. We decided to use 6 leds of each colour, 3 in series and then the 2 sets in parallel. This gave 18.8mA current through the green leds and around 37.5mA current through the resistor. To set the power to less than 0.25 W(so that 1206 footprint could work for it) I chose 150 ohm. But for red and yellow leds the situation is a bit different because voltage drop across green led is around 2.12V whereas across red and yellow leds it is around 1.8V. Therefore the voltage across the resistance increases and hence the power too so to reduce the power I increased the resistance to 170 ohms and now the current through the leds is 19.2mA which can turn the led on. So the final circuit-



Final Traffic Light circuit



Output waveform

7. Working on ALTIUM-

7.1. Creating Schematic Library components-

First of all we started work on schematic library. For that our first job was to create component symbols to be used in the schematic. I created two resistors- one 0805 and the other 1206. I chose 0805 for the resistor through which the capacitor gets charged. It had a high enough power rating which was more than enough for this resistance(didn't chose 0603 because they are smaller and difficult to solder so if 0805 works well for the circuit then why not go for it). Same was used for the pull down resistor as well. 1206 resistor was for the resistor connected to the leds because their power rating is 0.25W and I needed the same for the circuit-

Power through the resistor connected to green leds -> $I^2R = (37.5)^2 * 150 = 211mW$

Power through the resistor connected to red and yellow leds -> $I^2R = (19.2*2)^2 * 170 = 250mW$

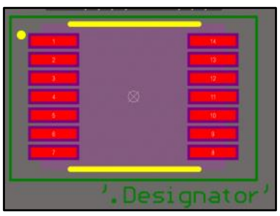
After this our next component was LED which I chose based on it's voltage level and maximum current that can pass through it. Next was the capacitor which I chose 0805 because it could operate at 25V maximum and our supply is 12V so this was perfect! I also saw the maximum and minimum requirements of these components. Next was the 555 timer IC. For this I needed an IC that could operate at 12V smoothly and the maximum input and output currents should also satisfy the circuit. My circuit needs the timer to output around 40mA current so I chose an IC which has 200mA as maximum output current. Next was AND gate IC. I chose an IC which had 3 circuits inside so one single IC was required in the whole circuit. Next was NOT gate IC. For this I used one IC having 6 circuits inside and three single circuit NOT ICs. These were again chosen on the basis of the voltage supply they can operate at. All of them have a supply range of around 16-18V. All the ICs chosen were CMOS. The main advantage of CMOS logic family is *their extremely low power consumption*. This is because there is no direct conducting path from Vcc to ground in either of input conditions. So there is practically zero power dissipation in STATIC conditions.

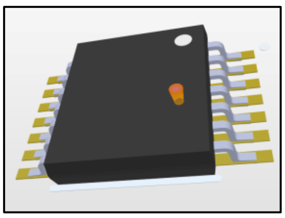
Next was the push button. I couldn't find the appropriate switch in digikey so I took it from SnapEDA. Last was the header which I added much later in the library.

7.2. Creating Footprints-

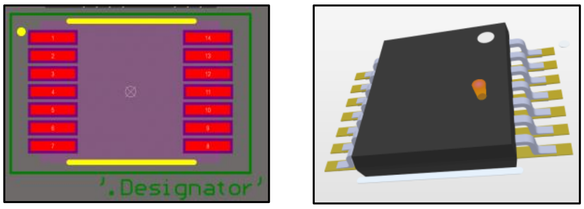
Then we started our work on the footprints of the components. First of all, for some components we used the wrong dimensions resulting in a component that couldn't be soldered. Actually we used the dimensions of the part itself rather than the pad dimensions, so later we corrected them. Depending upon the data and gap lengths given in the datasheet of the various components we created all the footprints. Next task was to add all these footprints to their respective components.

6 NOT IC-

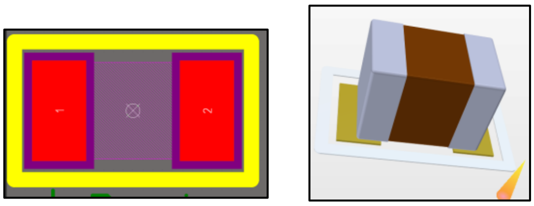




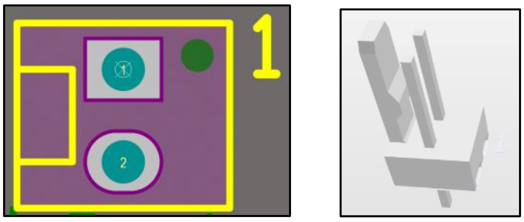
AND IC-



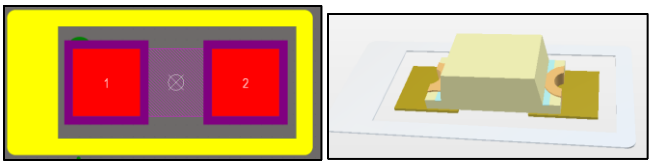
Capacitor-



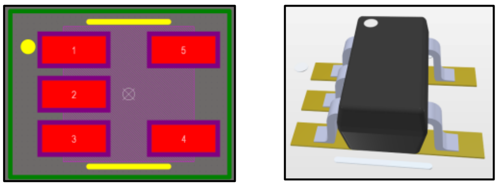
Header-



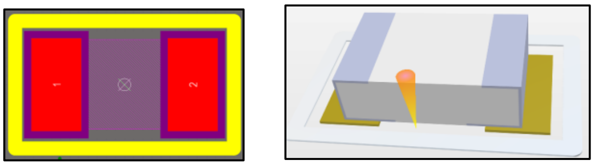
LED-



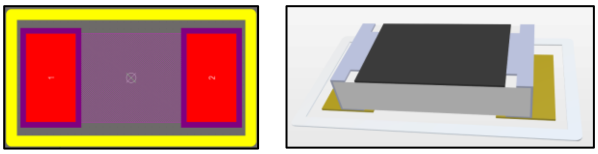
NOT IC-



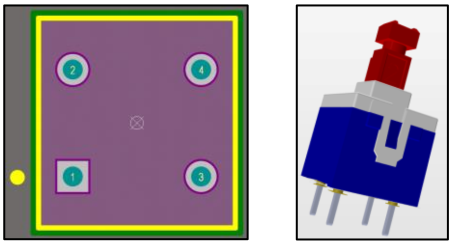
Resistor0805-



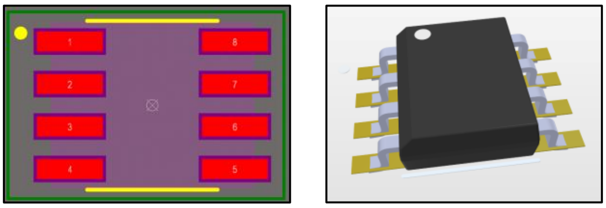
Resistor1206-



Switch-

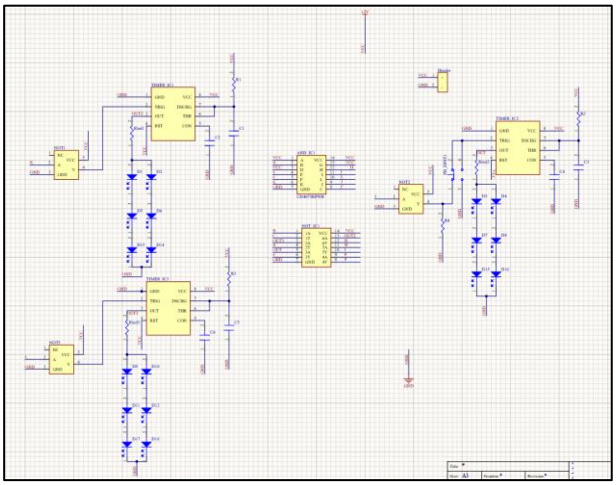


555 Timer IC-



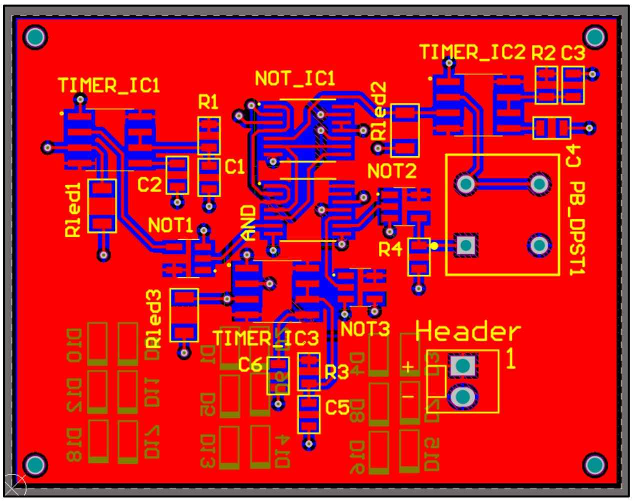
7.3. <u>Creating Schematic-</u>

Finally we started our work on the schematic. In my first attempt I connected all the components by simply using wires so the circuit was a bit messy. In my second attempt I used Net labels for making the connections, making the circuit easy to understand-

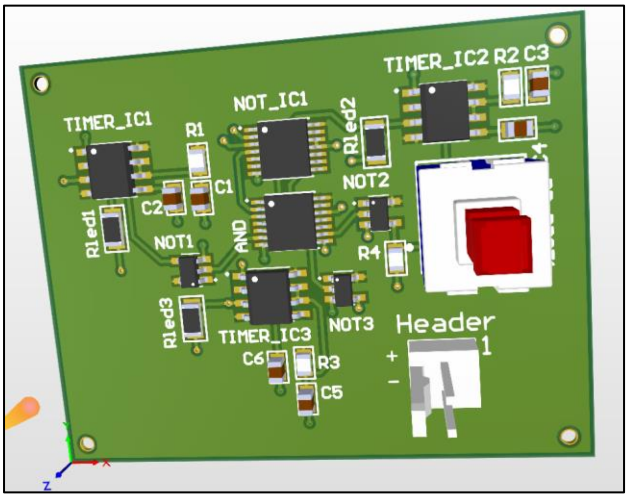


7.4. <u>PCB Routing-</u>

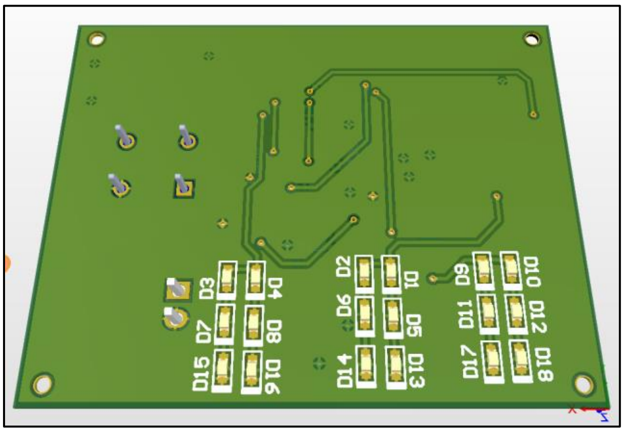
Now our task was to update the PCB from the schematic. All the components were updated on to the PCB with their connections shown. My first few routings went wrong because the connections were too long and moreover I was unable to complete various other connections. Then I used the bottom layer to make the connections and positioned the components correctly so that they can be brought nearer and the size of the PCB can be reduced. I used vias to make connections using bottom layer. Moreover to make the routing even more simpler I used two polygon pours- one at the bottom layer for ground and the other at the top layer for VCC(supply). The top polygon pour gave me unrouted constraint error when I put it for the 1st time. This was because islands were created and there was no way to connect those. To correct this I first thought to use vias and connect those unconnected islands but later when I brought the components nearer and used the bottom layer as well, this error was gone! Finally I completed the routing and connected the LEDs at the back of the PCB so that it looks good. Thus, after many re-routings PCB was completed successfully!-



Final Routing



PCB Front


PCB Back

8. Conclusions-

So basically we learnt:

- Simulating 555 Timer IC and understood its internal circuitry.
- Using it as a sequential control circuit to generate a continuously looping RYG LED circuit which functions like a real-life traffic light.
- Adjusting the ON and OFF Timings of the lights.
- Choosing proper components that would work best in our circuit, for eg- choosing the footprints of resistors and capacitors.
- Designing PCB- how to use various layers and how to position the components in order to reduce the size of the PCB and make connections efficient.

PCB Dimensions:

- Horizontal – 49.784mm
- Vertical – 38.608mm
- Area – 1922.061 sq. mm
- Components Area – 593.043 sq. mm

9. Acknowledgements-

This project was conducted by IIT Roorkee Motorsports of Indian Institute of Technology, Roorkee. The project was conducted under the guidance of Aditya Sharma (IIIrd Year, EE) and Roshan Raja Samuel A. (IVth Year, EE). I would like to sincerely thank them for their time and useful advice.

10. References-

- 555 Timer working – https://www.youtube.com/watch?v=i0SNb__dkYI https://www.youtube.com/watch?v=oZzjmAbyyIQ https://www.youtube.com/watch?v=qfWIjb48mjE
- Monostable operation- https://www.youtube.com/watch?v=ypV6gdIJJU4
- Robert's Altium Playlist- https://www.youtube.com/watch?v=KpgTud1iQ-4&list=PLXvLToQzgzdfKKQn2wmpuSXz6sROQmO6R