**Department of Electrical Engineering**

**Faculty Member:** Arshad Nazir **Dated:** 08/10/2021

**Semester:** 3rd **Section:** BEE 12C

**Group No.: 7**

**EE-221: Digital Logic Design**

Lab 3: Design of Simple Practical Circuits

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **PLO4/CLO4** | **PLO4/CLO4** | **PLO5/CLO5** | **PLO8/CLO6** | **PLO9/CLO7** |  |
| **Name** | **Reg. No** | **Viva / Lab Performance** | **Analysis of data in Lab Report** | **Modern Tool Usage** | **Ethics and Safety** | **Individual and Teamwork** | **Total marks Obtained** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **25 Marks** |
| **Muhammad Adil Azeem** | 342459 |  |  |  |  |  |  |
| **Muhammad Umer** | 345834 |  |  |  |  |  |  |
| **Raahim Ahmad Waqar** | 342287 |  |  |  |  |  |  |
| **Saad Bakhtiar** | 341150 |  |  |  |  |  |  |
| **Syed Aun Ali Kazmi** | 342384 |  |  |  |  |  |  |

## Lab 2: Design of Simple Practical Circuits



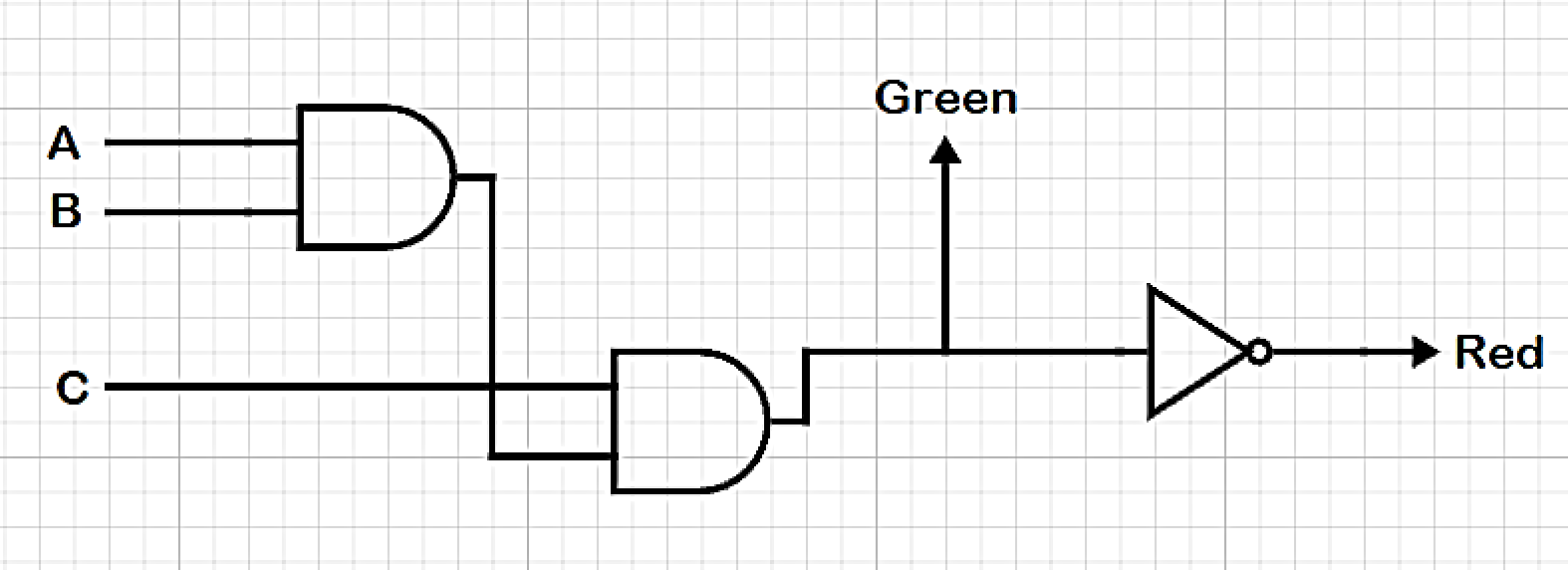
**Lab Instructions**

* This lab activity comprises three parts, namely Pre-lab, Lab tasks, and Post-Lab Viva session.
* The lab report will be uploaded on LMS three days before the scheduled lab date. The students will get a hard copy of the lab report, complete the Pre-lab task before coming to the lab and deposit it with the teacher/lab engineer for necessary evaluation. Alternately each group uploads a completed lab report on LMS for grading.
* The students failing to submit Pre-Lab will not be allowed to perform Lab work.
* The students will start lab task and demonstrate design steps separately for stepwise evaluation (course instructor/lab engineer will sign each step after ascertaining functional verification)
* Remember that a neat logic diagram with pins numbered coupled with a nicely patched circuit will simplify trouble-shooting.
* After the lab, students are expected to unwire the circuit and deposit back components before leaving.
* The students will complete a lab task and submit a complete report to the Lab Engineer before leaving the labs. The Verilog tutorial part is non-printable and for reference only.
* There are related questions at the end of this activity. Give complete answers.

**Pre-Lab Tasks:**

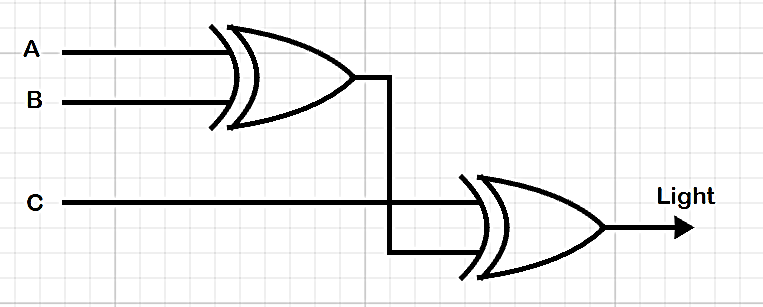
1. Design the practical circuits given in Task 1 and Task 2 by giving the truth table of the problems and then giving the circuit (logic diagram) for the designs

**Task 1**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | | | **Output** | |
| **Left** | **Right** | **Nose** | **RED LED** | **GREEN LED** |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |

**Task 2**



|  |  |  |  |
| --- | --- | --- | --- |
| **zInput** | | | **Output** |
| **S1** | **S2** | **S3** | **Light** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

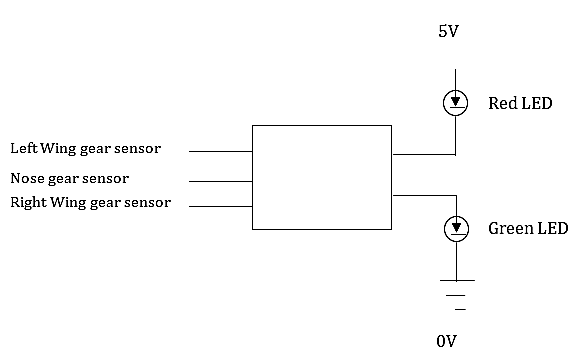
1. Provide only Truth Table of Task 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Speed Limit | Speed Range | Overspeeding | Dangerous Overspeeding |
| ≤ 45 | 00 | 00 | **0** | **0** |
| 00 | 01 | **1** | **0** |
| 00 | 10 | **1** | **1** |
| 00 | 11 | **1** | **1** |
| > 45 and < 55 | 01 | 00 | **0** | **0** |
| 01 | 01 | **0** | **0** |
| 01 | 10 | **1** | **0** |
| 01 | 11 | **1** | **1** |
| > 55 and < 65 | 10 | 00 | **0** | **0** |
| 10 | 01 | **0** | **0** |
| 10 | 10 | **0** | **0** |
| 10 | 11 | **1** | **1** |
| ≥ 65 | 11 | 00 | **X** | **X** |
| 11 | 01 | **X** | **X** |
| 11 | 10 | **X** | **X** |
| 11 | 11 | **X** | **1** |

**Lab Tasks:**

1. **Lab Task 1:**

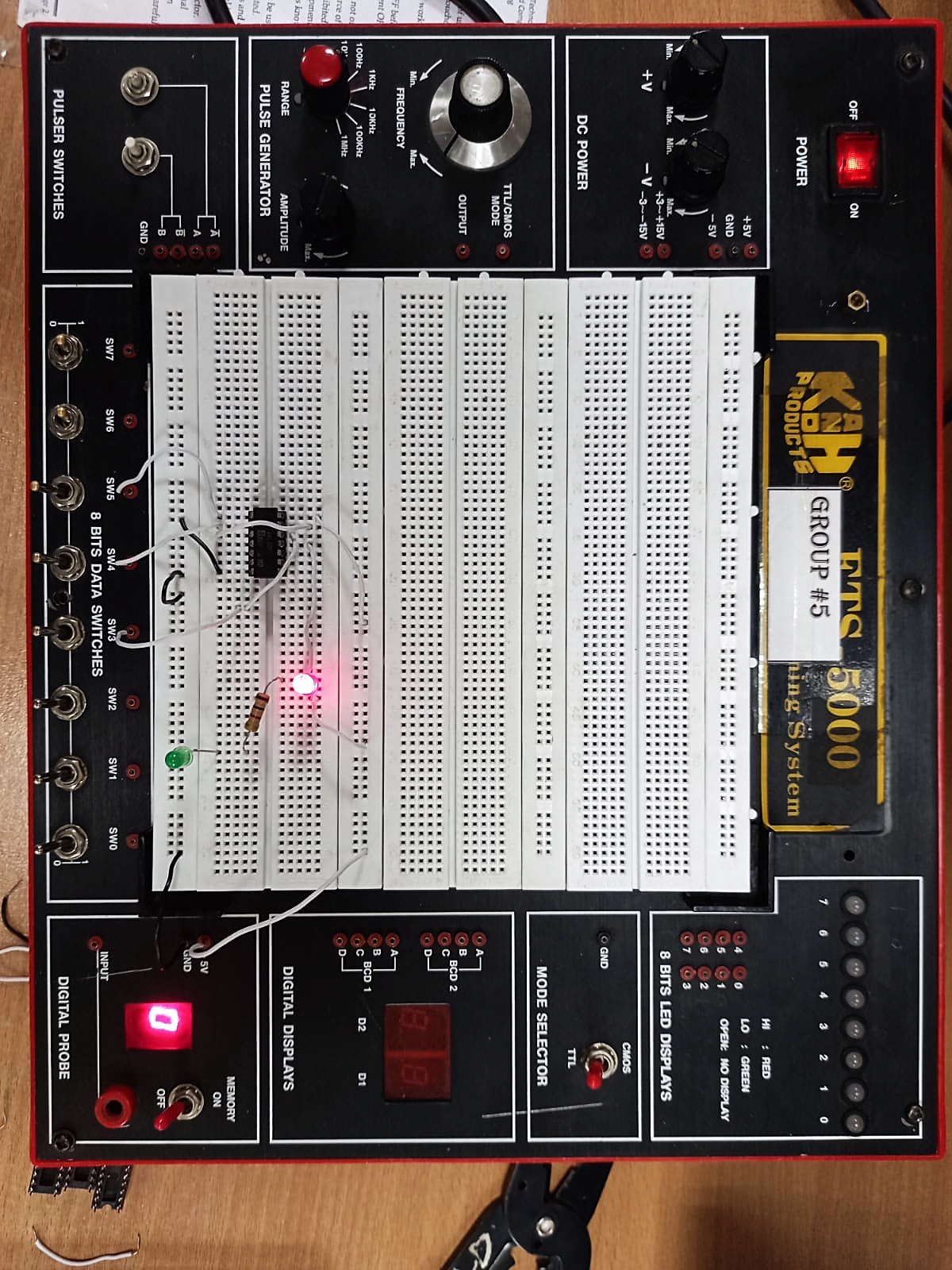
As part of an aircraft’s functional monitoring system, a circuit is required to indicate the status of the landing gear prior to landing. A green LED (Light Emitting Diode) display turns on if all three gears are properly extended when the “gear down” switch has been activated in preparation for landing. A red LED display turns on if any of the gear fails to extent properly prior to landing. When landing gear is extended, its sensor produces a HIGH voltage. When a landing gear is retracted its sensor produces a LOW voltage. Write the truth table for the aircraft landing system. Implement with basic logic gates.



**Proteus Simulation**



**Hardware Implementation**



The above hardware implementation performed in accordance with the Truth Table filled in the Task 1 (**part a**) of the Pre-Lab section of this document; that the Green LED only lit up when all of the switches (playing the role of gears here) were turned ON.

**Verilog Simulation**

module task1 (red, green, left, nose, right);

    input left, nose, right;

    output red, green;

    wire w1, w2;

    and a1(w1, left, nose);

    and a2(w2, w1, right);

    assign {green} = w2;

    not n1(red, w2);

endmodule

module testbench ();

    reg LEFT, RIGHT, NOSE;

    wire RED, GREEN;

    task1 t1(RED, GREEN, LEFT, RIGHT, NOSE);

    initial begin

        #100ps LEFT = 0; RIGHT = 0; NOSE = 0;

        #100ps LEFT = 0; RIGHT = 0; NOSE = 1;

        #100ps LEFT = 0; RIGHT = 1; NOSE = 0;

        #100ps LEFT = 0; RIGHT = 1; NOSE = 1;

        #100ps LEFT = 1; RIGHT = 0; NOSE = 0;

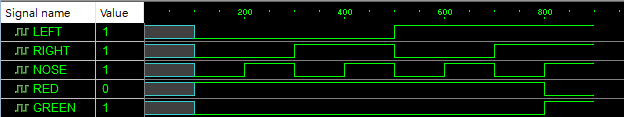
        #100ps LEFT = 1; RIGHT = 0; NOSE = 1;

        #100ps LEFT = 1; RIGHT = 1; NOSE = 0;

        #100ps LEFT = 1; RIGHT = 1; NOSE = 1;

    end

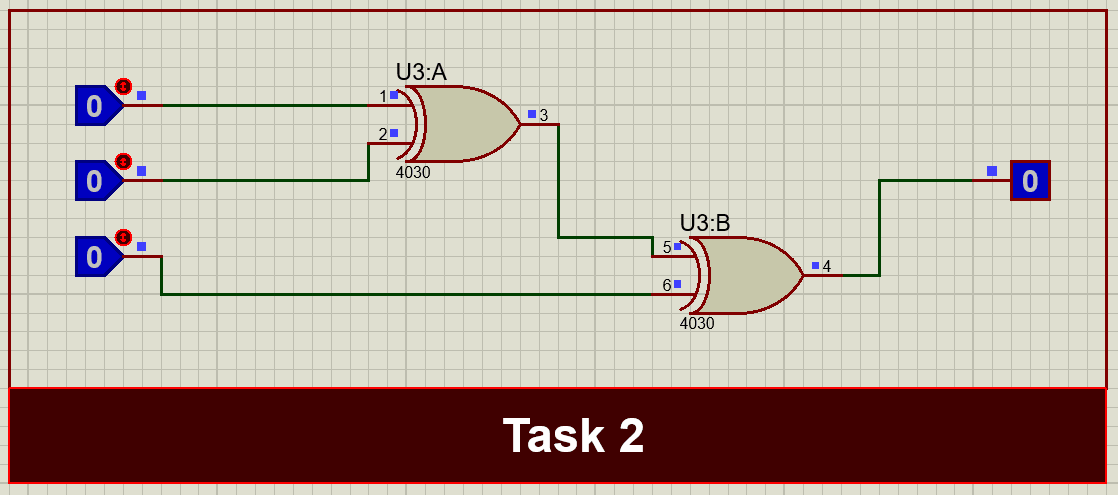
endmodule



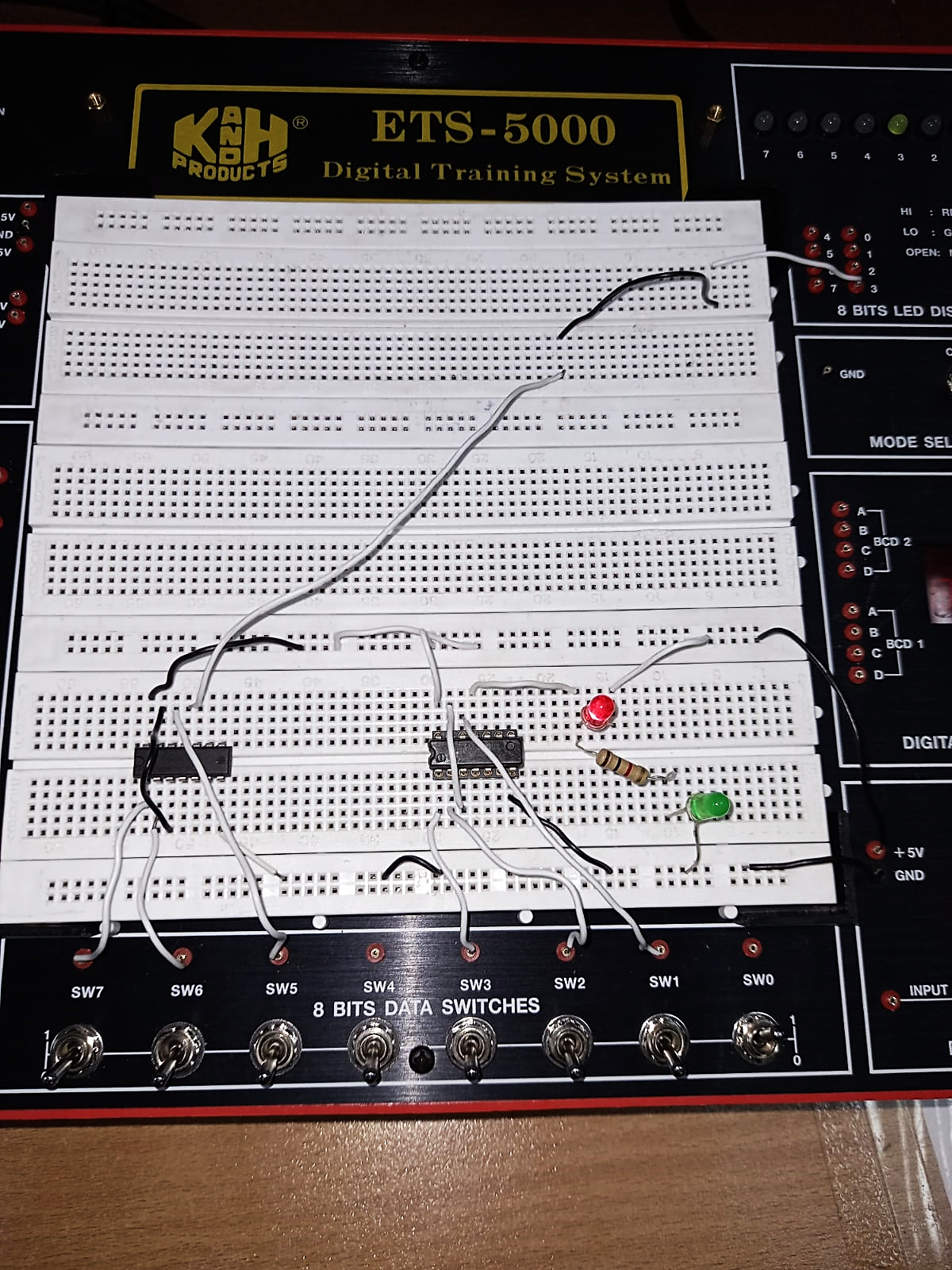
1. **Lab Task 2:**

Design a three-way switch for lighting a bulb. Suppose the Switch A is installed at the entrance of the room, switch B is installed to the bedside table, Switch C is installed outside the room. Now three way switching can facilitate the switching ON/OFF from any location. The mechanism is such that if a single switch is on the bulb should be ON. Now if the second switch is turned on the bulb if switched on earlier is switched off. If two switches are already on and third switch is turned on then it should switch on the bulb. Implement with basic gates.

**Proteus Simulation**



**Hardware Implementation**



Similar to Task 1’s hardware implementation, it performed in accordance with the Truth Table filled in the Task 1 (**part b**) of the Pre-Lab section of this document. The output was only HIGH (1) when either all the switches were ON, or individual switches were ON with other two being OFF; essentially, performing as a three-input XOR gate.

**Verilog Simulation**

module task2 (light, a, b, c);

    input a, b, c;

    output light;

    wire w1;

    xor(w1, a, b);

    xor(light, w1, c);

endmodule

module testbench2 ();

    reg A, B, C;

    wire LIGHT;

    task2 t2(LIGHT, A, B, C);

    initial begin

          #100ps A = 0; B = 0; C = 0;

          #100ps A = 0; B = 0; C = 1;

          #100ps A = 0; B = 1; C = 0;

          #100ps A = 0; B = 1; C = 1;

          #100ps A = 1; B = 0; C = 0;

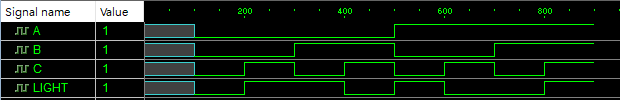
          #100ps A = 1; B = 0; C = 1;

          #100ps A = 1; B = 1; C = 0;

          #100ps A = 1; B = 1; C = 1;

    end

endmodule



1. **Lab Task 3:**

The system is a speed warning device. It receives, on two lines, an indication of the speed limit on the highway. There are three possible values 45, 55, or 65 MPH. It receives from the automobile, on two other lines, an indication of speed of the vehicle. There are four possible values under 45, between 46 and 55, between 56 and 65, and over 65 MPH. It produces two outputs. The first f, indicates whether the car is going above the speed limit. The second g, indicates that the car is driving at “dangerous speed” – defined as either over 65 MPH or more than 10 MPH above the speed limit. Show how each of the inputs and outputs are coded (in terms of binary values) and complete the truth table for the system.

**Coding Convention**

|  |  |
| --- | --- |
| **Key** | **Interval** |
| 00 | ≤ 45 |
| 01 | > 45 and < 55 |
| 10 | > 55 and < 65 |
| 11 | ≥ 65 |

**Truth Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Speed Limit** | **Speed Range** | **Overspeeding** | **Dangerous Overspeeding** |
| **≤ 45** | 00 | 00 | **0** | **0** |
| 00 | 01 | **1** | **0** |
| 00 | 10 | **1** | **1** |
| 00 | 11 | **1** | **1** |
| **> 45 and < 55** | 01 | 00 | **0** | **0** |
| 01 | 01 | **0** | **0** |
| 01 | 10 | **1** | **0** |
| 01 | 11 | **1** | **1** |
| **> 55 and < 65** | 10 | 00 | **0** | **0** |
| 10 | 01 | **0** | **0** |
| 10 | 10 | **0** | **0** |
| 10 | 11 | **1** | **1** |
| **≥ 65** | 11 | 00 | **X** | **X** |
| 11 | 01 | **X** | **X** |
| 11 | 10 | **X** | **X** |
| 11 | 11 | **X** | **1** |

**Observations/Comments:**

We have deployed the concepts learned in the previous labs to implement simple practical circuits and with it, stated such a circuit’s attributes; namely, the *Truth Table, Proteus Modeling and Gate-level code* of the respective circuit in an Active HDL software. We have also confirmed/verified the circuit’s output as it conforms to the truth table laid out in the Pre – lab and that it serves the purpose for what it was designed definitively.