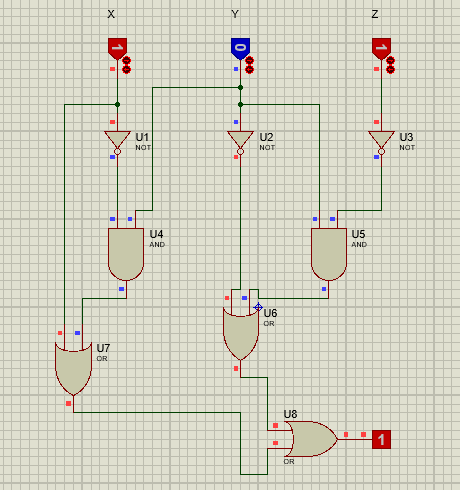
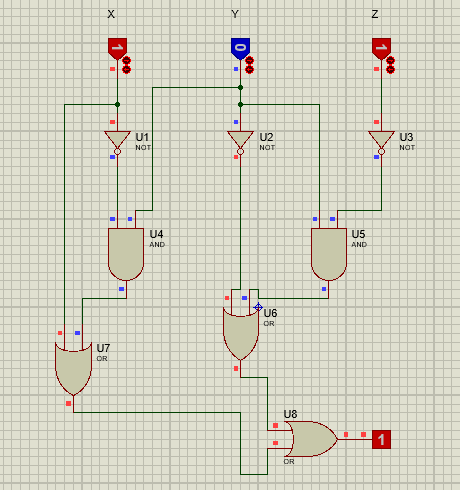
# Digital Logic Design EEE241

Lab Report



|  |  |
| --- | --- |
| **Name** | **M. Ruslan Babar** |
| **Registration Number** | **FA20-BSE-094** |
| **Class** | **BSE 2B** |
| **Instructor’s Name** | **Sir Sajid Ali Gillal** |

**True In Each Case (Tautology)**

****

**Report**

**TRUTH TABLE**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **X** | **Y** | **Z** | **Z’** | **Y’** | **X’** | **X’Y** | **YZ’** | **X+X’Y+YZ’+Y’**  1  1  1  1  1  1  1  1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

**Requirements:**

* And Gate
* NOT Gate
* OR Gate
* Wires.
* Logic States (X, Y, Z)
* Probe (Output Display)

**Design and Implementation:**

* Top to Bottom design.
* The sequence of Logical operators is:
  + Firstly, Logic States
  + Secondly NOT Gate is applied.
  + Then, Combinations of AND Gates are implemented according to the equation: (X+X’Y+YZ’+Y’).
  + At the end AND Gates are attached to multiple OR gates for Final Output.
  + Probe is used to display the Logical Output.

**Properties of OR Gate:**

* A High output (1) results if one or both the inputs to the gate are High (1).
* If neither input is high, a LOW output (0) results.

**Properties of AND Gate:**

* A High output (1) results only if all the inputs to the **AND** gate are High (1).
* If none or not all inputs to the **AND** gate is High, then it results in low Output.

**Conclusions:**

* I learn how to design a digital circuit using Proteus Software.
* I am not able to play with logic gates to implement more Logical circuits.
* The Circuit shown in first figure show the complete functioning of **AND, OR and NOT gates.** The **OR** gate results to true if any of the state is true. The **AND** gate results to true only if both the state are true. The **NOT** gate negates the state of **Input**.
* The designing factor is very important to acknowledge, for better understandings.
* The circuit implemented in figure is a tautology because it gives true in every case of input.

**END**