**Department of Electrical Engineering**

|  |  |
| --- | --- |
| **Faculty Member: Miss Rafia** | **Dated: 11-3-25** |
| **Semester: 2nd** | **Section: ds-2a** |

*Group No.:*

**EE-241: Digital Logic Design**

**Lab 07: Design a display system of a rolling dice**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **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 Team Work** | **Total marks Obtained** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **25 Marks** |
| **AILYA ZAINAB** | **523506** |  |  |  |  |  |  |
| **IMAN NAEEM** | **525378** |  |  |  |  |  |  |
| **LAIBA NASIR** | **510419** |  |  |  |  |  |  |
| **LUQMAN SHEHZAD** | **507599** |  |  |  |  |  |  |

## Lab7: Design a system that display from 1 to 6. It displays the result on a dice. The dice has seven lights

This Lab Activity has been designed to practice the use of basic gates for designing a system

* Simplification of Combinational Circuits
* Design and Implementation of a design a system to display dice values.
* Values-Segment Decoder for Selected Digit Display
* There are related questions at the end of this activity. Give complete answers.

Use diagrams if needed for clarity.

**PreLab Task (2 marks)**

Design a system that display from 1 to 6 (ONLY)i.e. it shows no output in case of illegal input like 0 and 7. It displays the result on a dice. The dice has seven LEDs a,b,c,d,e,f,g, placed in H shape pattern as shown on the diagram below

**a**

**b**

**c**

**f**

**e**

**g**

**d**

A 1 for each segment ( a, b, c, d, e, f, g) indicates that it is lit (on); and a 0 that it is off. The arrangement is the six numbers on a dice are shown below; where the darken circles depicts LED is ON. (**Make sure to switch off all the lights in Don’t care case**)

7 bit output

3 bit Input

**6**

**5**

**4**

**3**

**2**

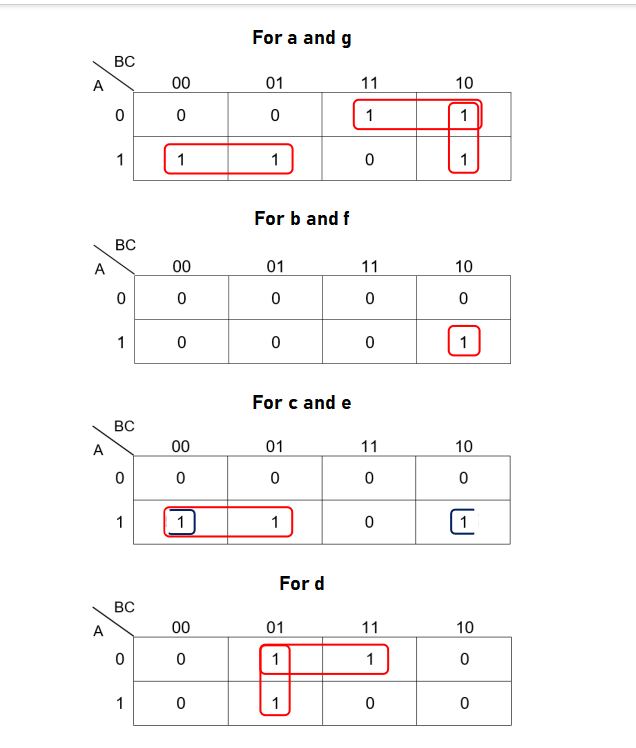
**1**

* Design a driver that produces the seven LED (a, b, c, d, e, f, g) to drive the display.

1. Complete the following table. (**Make sure to switch off all the lights in Don’t care case**) (First 3 inputs are filled for guidance)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Inputs (Binary)** | | | **Outputs (7 LEDs on Dice Display)** | | | | | | |
| **#m** | **A** | **B** | **C** | **a** | **b** | **c** | **d** | **e** | **f** | **g** |
| **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| **2** | **0** | **1** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **1** |
| **3** | **0** | **1** | **1** | **1** | **0** | **0** | **1** | **0** | **0** | **1** |
| **4** | **1** | **0** | **0** | **1** | **0** | **1** | **0** | **1** | **0** | **1** |
| **5** | **1** | **0** | **1** | **1** | **0** | **1** | **1** | **1** | **0** | **1** |
| **6** | **1** | **1** | **0** | **1** | **1** | **1** | **0** | **1** | **1** | **1** |
| **7** | **1** | **1** | **1** | **0** | **1** | **0** | **0** | **0** | **0** | **0** |

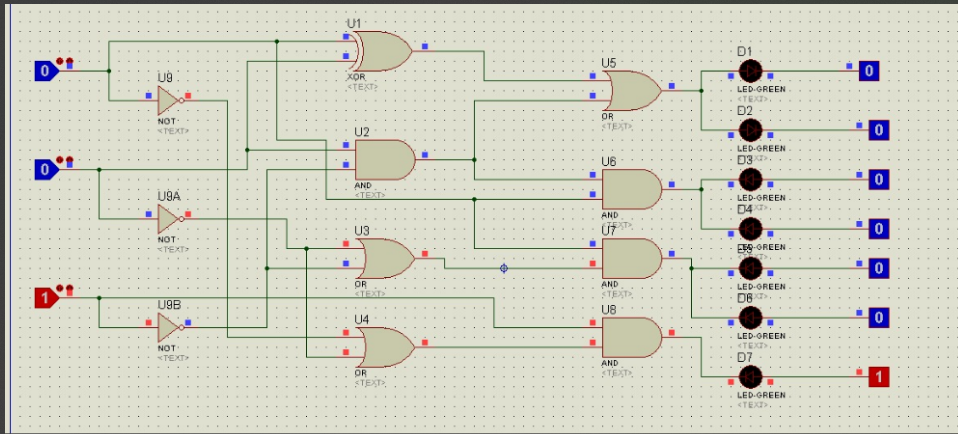
* Write minimum possible functions to realize outputs (Either using k-mapping/ or minimization of Boolean function) (**Make sure to switch off all the lights in Don’t care case**). Show and get verified the minimized Boolean Function expressions to Lab Engineer before implementation.

Hint: You will get 7 output expressions. Output expression for ‘a’ and ‘g’ will be same. Output expression for ‘b’, and ’f’ will be same. Output expression for ‘c’ , and ‘e’ will be same.  
 

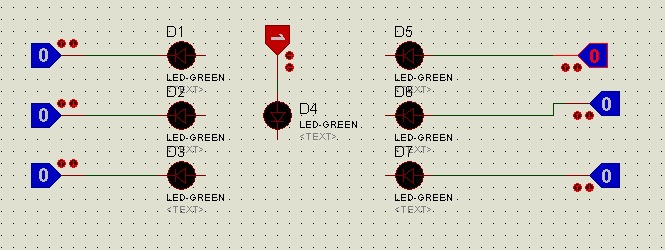
**Simplified Expressions**

|  |  |  |
| --- | --- | --- |
| **Variable** | | **Expression** |
| **a** | **g** | **(A ⊕ B) + BC’** |
| **b** | **f** | **A(BC’)** |
| **c** | **e** | **A(B’+C’)** |
| **d** | **-** | **C(A’+B’)** |

* Draw the complete logic circuit diagram of the system from simplified equations (Proteus)



DICE FORM OUTPUT:

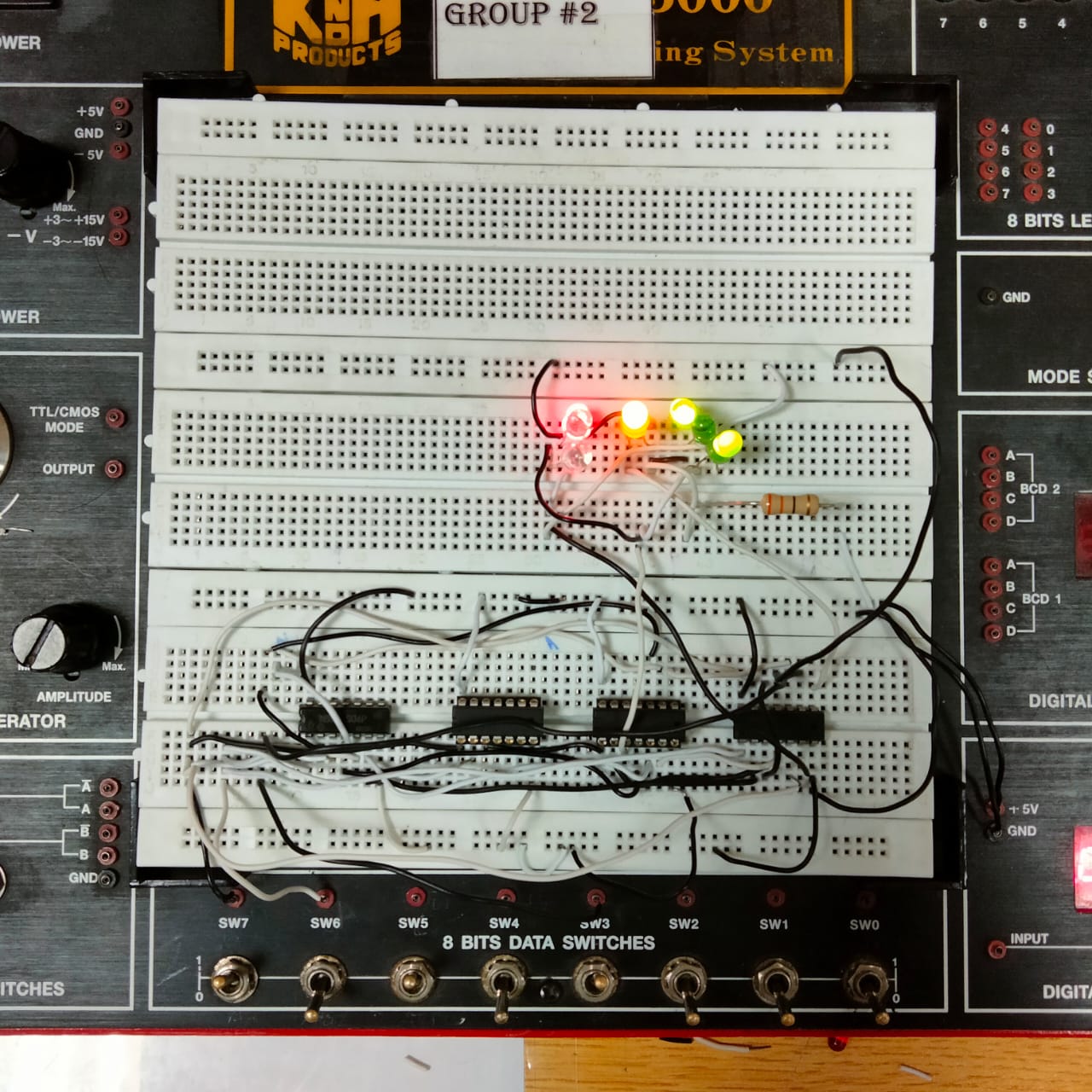
****

**Lab Task 1 (5 marks)**

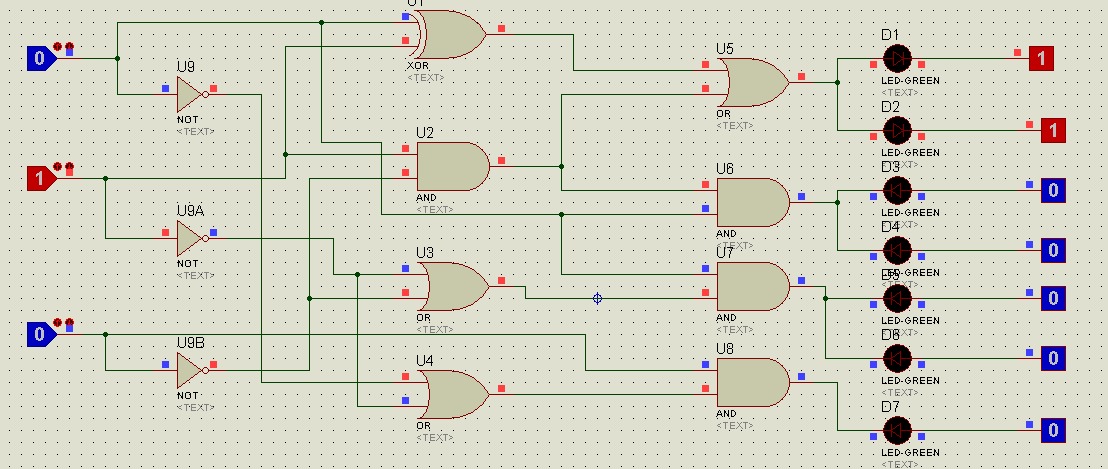
* Implement the designed logic circuit on hardware. Utilize your creativity to make the Dice display **model**

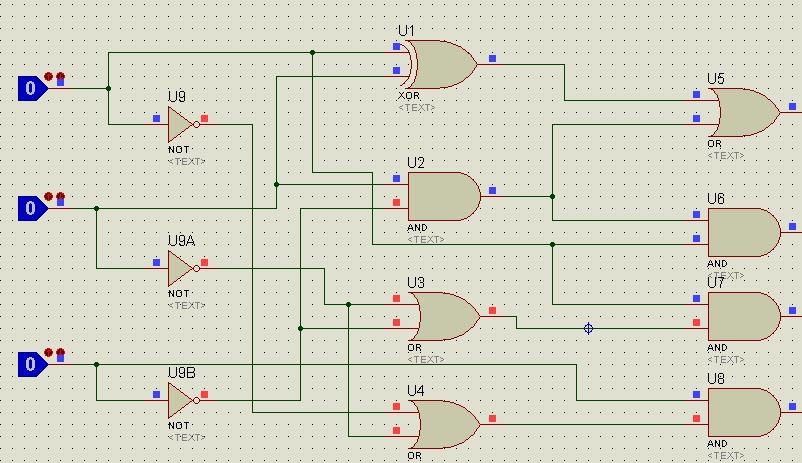
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Inputs (Binary)** | | | **Outputs (7 LEDs on Dice Display)** | | | | | | |
| **#m** | **A** | **B** | **C** | **a** | **b** | **c** | **d** | **e** | **f** | **g** |
| **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| **2** | **0** | **1** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **1** |
| **3** | **0** | **1** | **1** | **1** | **0** | **0** | **1** | **0** | **0** | **1** |
| **4** | **1** | **0** | **0** | **1** | **0** | **1** | **0** | **1** | **0** | **1** |
| **5** | **1** | **0** | **1** | **1** | **0** | **1** | **1** | **1** | **0** | **1** |
| **6** | **1** | **1** | **0** | **1** | **1** | **1** | **0** | **1** | **1** | **1** |
| **7** | **1** | **1** | **1** | **0** | **1** | **0** | **0** | **0** | **0** | **0** |

**Hardware Implementation**

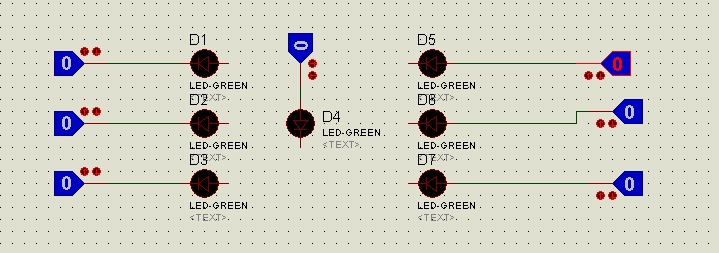


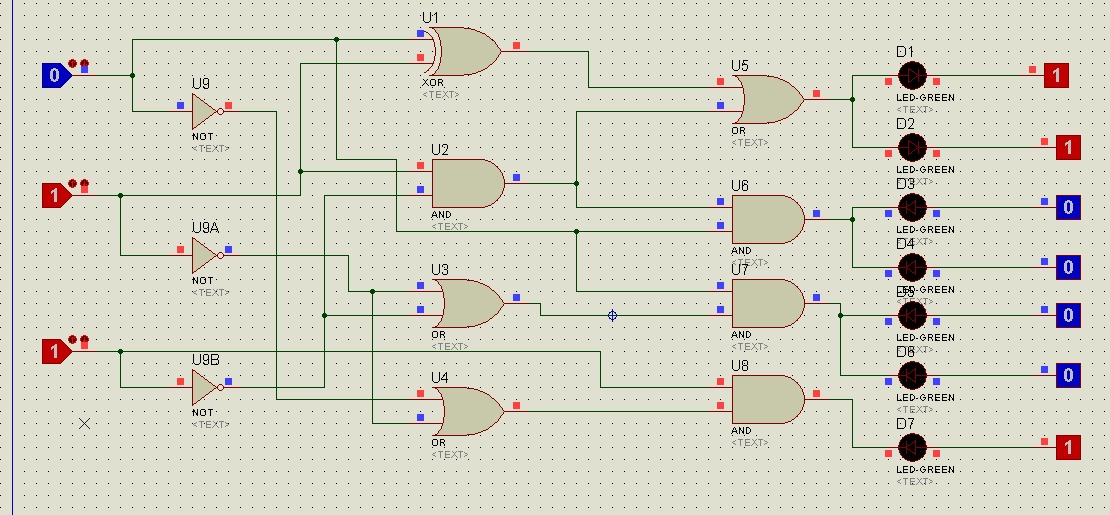
**PROTEUS:**

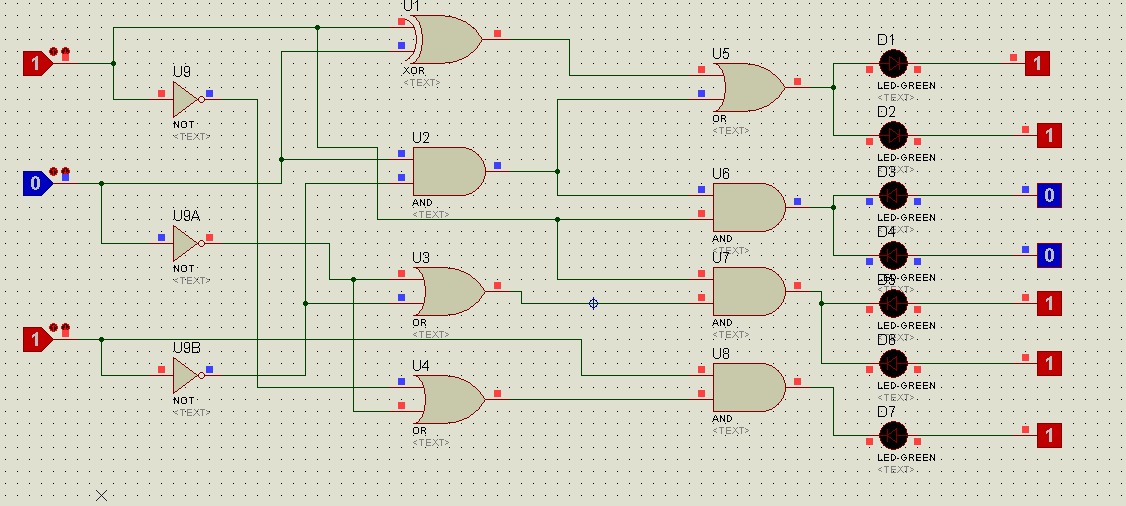


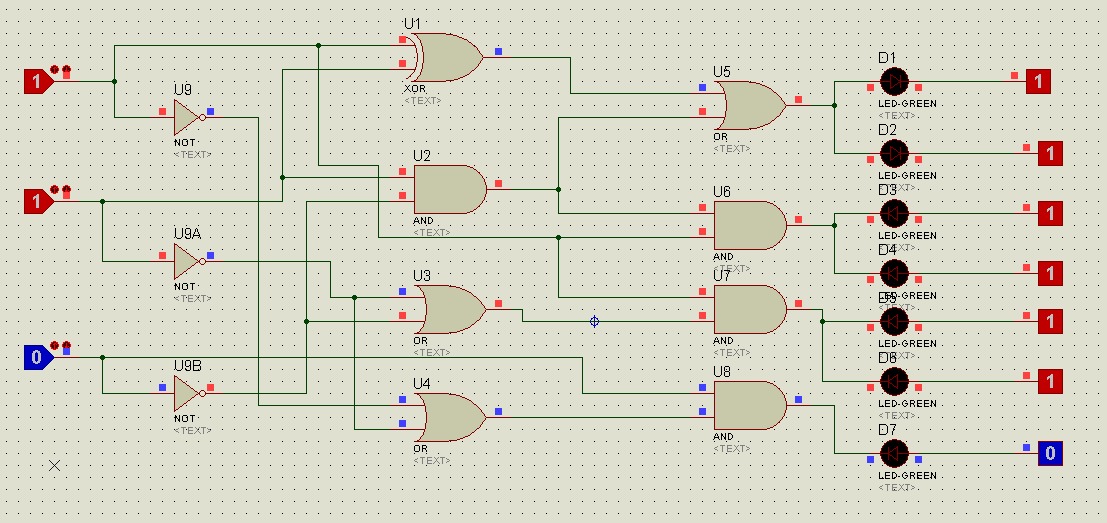


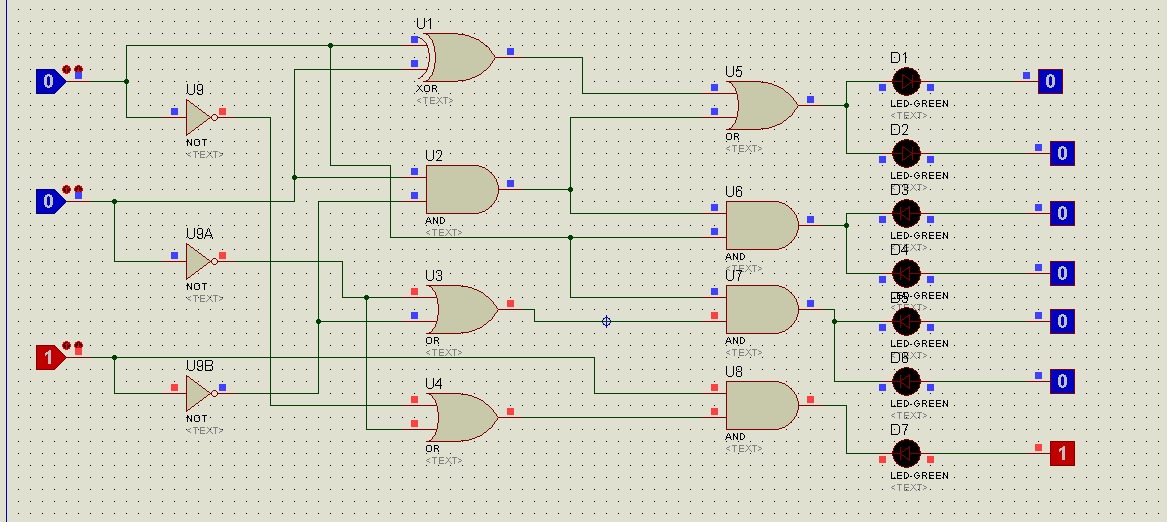
OUTPUT:

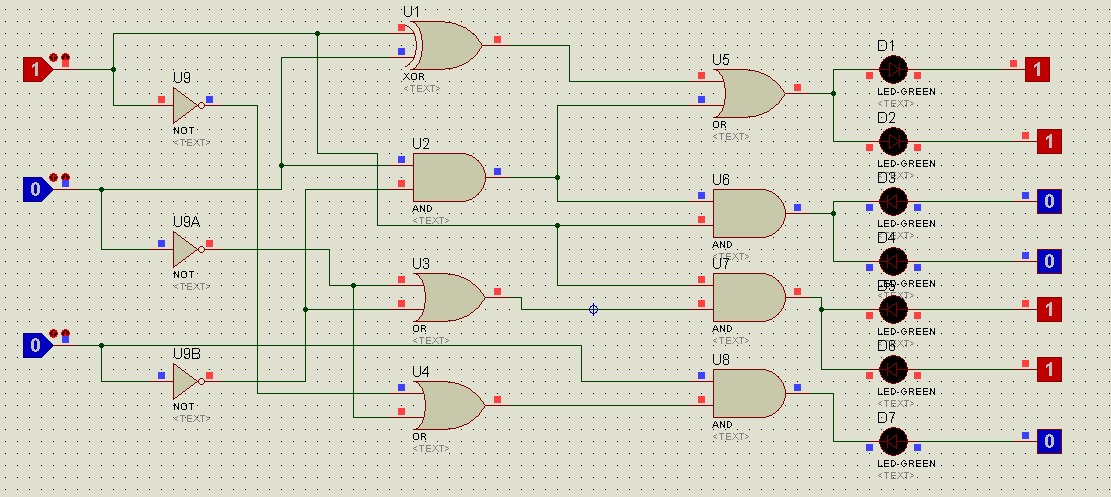












**Lab Task 2 (3 marks)**

* Write Verilog code to realize the design using dataflow model. Design also test bench to check the valid outputs. Include all the snap shots and Verilog code in the report.

**CODE:**

module task1(a, b, c, d, e, f, g, A, B, C);

input A, B, C;

output a, b, c, d, e, f, g;

assign a = (~A & B) | (A & ~B) | (B & ~C);

assign g = (~A & B) | (A & ~B) | (B & ~C);

assign b = A & (B & ~C);

assign f = A & (B & ~C);

assign c = A & (~B | ~C);

assign e = A & (~B | ~C);

assign d = C & (~A | ~B);

endmodule

module testbench;

reg A, B, C;

wire a, b, c, d, e, f, g;

task1 t1(a, b, c, d, e, f, g, A, B, C);

initial

begin

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

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

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

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

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

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

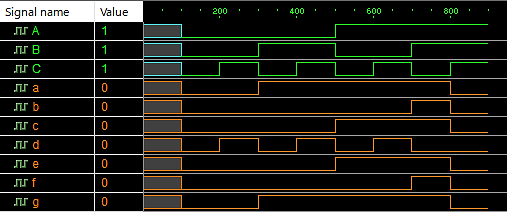
#100 A = 1; B = 1; C = 0;

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

end

Endmodule

**WAVE FORM:**



**Observations/Comments:**

In this lab we implemented the logic to design a display system of a rolling dice using:

1. Basic gates (AND, NOT and OR)
2. Using XOR

We implemented the LEDs (placed in an H - shape) at the receier end of the circuit and wrote the Verilog code for the circuit and verified the outputs.