



DEPT. of Computer Science Engineering

SRM IST, Kattankulathur – 603 203

Sub Code & Name: 18CSS201J - ANALOG AND DIGITAL ELECTRONICS

Experiment No	06
Title of Experiment	Design and implementation of Binary to gray code converters and vice versa using logic gates
Name of the candidate	RAHUL GOEL
Register Number	RA1911030010094
Date of Experiment	26/9/2020

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Oral Viva / Online Quiz	5	
2	Circuit Connection and Execution	10	
3	Verification of truth table	5	
Total		20	

Staff Signature with date

6.a. Design and implementation of Binary to gray code converters using logic gates

AIM

1. To design and implementation of Binary to gray code converters using Multisim-online software.
2. Hardware Implementation of the same with virtual Lab - IIT Bombay

SOFTWARE REQUIRED

<https://www.multisim.com/>

APPARATUS REQUIRED

S.No	Apparatus	Type	Range	Quantity
1	IC	IC 7486		1
2	LED			4
3	Switch			4
4	DC Power Source			1
5	Multisim online virtual lab IIT Bombay			
6	Wires			As Required

THEORY

The logical circuit which converts binary code to equivalent gray code is known as binary to gray code converter. The gray code is a non-weighted code. The successive gray code differs in one-bit position only that means it is a unit distance code. It is also referred as cyclic code. It is not suitable for arithmetic operations. It is the most popular of the unit distance codes. It is also a reflective code. An n-bit Gray code can be obtained by reflecting an n-1-bit code about an axis after 2^{n-1} rows, and putting the MSB of 0 above the axis and the MSB of 1 below the axis. This method uses an Ex-OR gate to perform among the binary bits. In this conversion method, take down the MSB bit of the present binary number, as the primary bit or MSB bit of the gray code number is similar to the binary number.

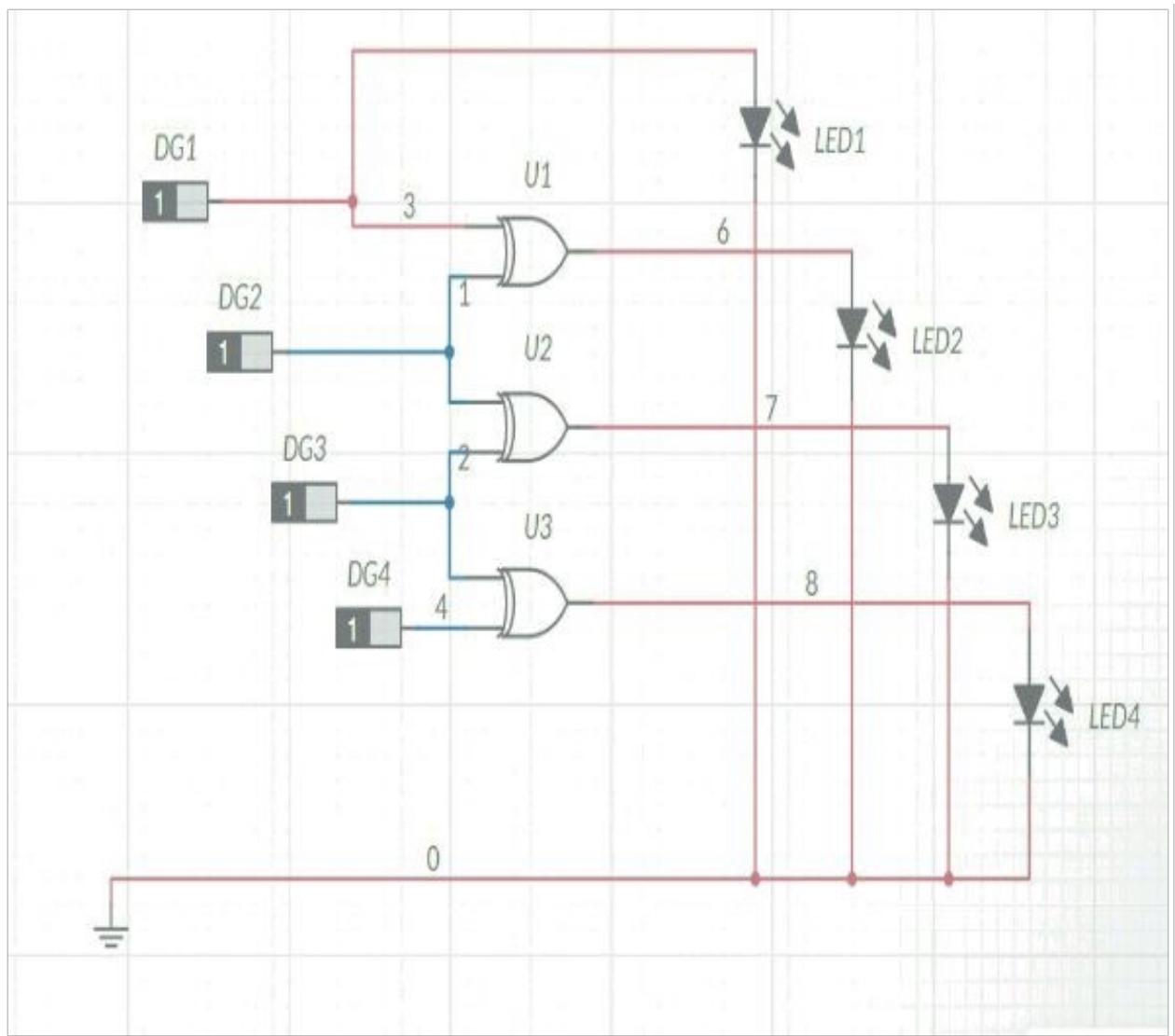
To get the straight gray coded bits for generating the corresponding gray coded digit for the given binary digits, add the primary digit or the MSB digit of binary number toward the second digit & note down the product next to the primary bit of gray code, and add the next binary bit to third bit then note down the product next to the 2nd bit of gray code. Similarly, follow this procedure until the final binary bit as well as note down the outcomes depending on EX-OR logic operation to generate the corresponding gray coded binary digit.

PROCEDURE

1. Open Multisim. Sign in.
2. Open a new circuit file
3. Select the components
 - Go to digital, choose digital constants

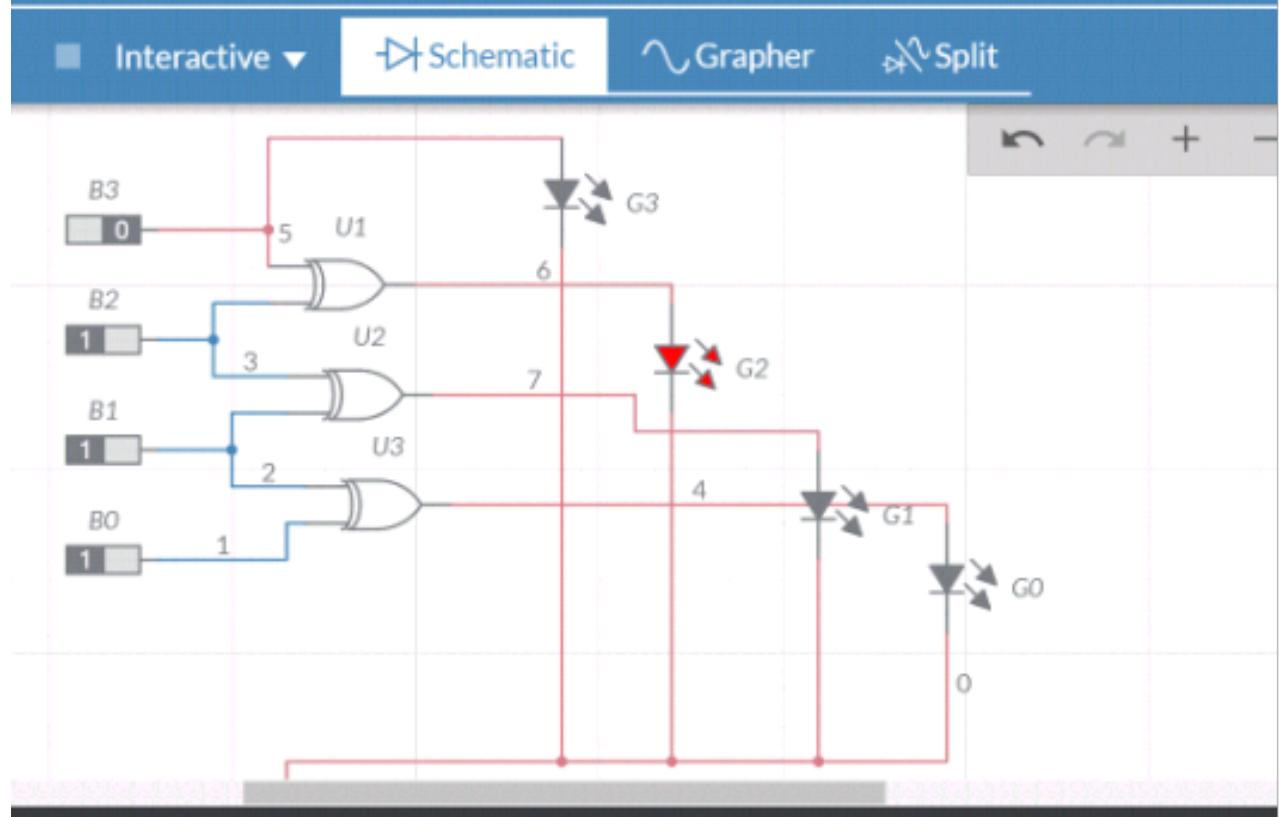
- Go to digital, choose XOR 2 input gate
 - Go to indicator, choose LED bulbs
 - Go to schematic connectors, choose ground
4. Duplicate the components and connect them as per circuit diagram.
 5. From analysis and annotation choose digital probe and place them where the output bits are to be seen
 6. Run the simulation and verify the output
 7. To change the input values, vary the values of high and low on the digital constants and verify the complete truth table illustrated below.

CIRCUIT DIAGRAM:



MY DIAGRAM:

*Binary to grayscale



TRUTH TABLE

BINARY				GRAY CODE			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

$$G3 = \sum(8,9,10,11,12,13,14,15)$$

$$G2 = \sum(4,5,6,7,8,9,10,11)$$

		B1B0	00	01	11	10
		B3B2	0	1	3	2
		00	0	1	3	2
		01	4	5	7	6
		11	1 ¹²	1 ¹³	1 ¹⁵	1 ¹⁴
		10	1 ⁸	1 ⁹	1 ¹¹	1 ¹⁰

		B1B0	00	01	11	10
		B3B2	0	1	3	2
		00	0	1	3	2
		01	1 ⁴	1 ⁵	1 ⁷	1 ⁶
		11	12	13	15	14
		10	1 ⁸	1 ⁹	1 ¹¹	1 ¹⁰

$$G3 = B3$$

$$G2 = \overline{B3}B2 + B3\overline{B2}$$

$$G2 = B3 \oplus B2$$

$$G1 = \sum(2,3,4,5,10,11,12,13)$$

$$G0 = \sum(1,2,3,5,6,9,10,13,14)$$

B1B0	00	01	11	10
B3B2	00	01	11	10
00	0	1	1 ³	1 ²
01	1 ⁴	1 ⁵	7	6
11	1 ¹²	1 ¹³	15	14
10	8	9	11 ¹¹	10 ¹⁰

$$G_1 = B_2 \bar{B}_1 + \bar{B}_2 B_1$$

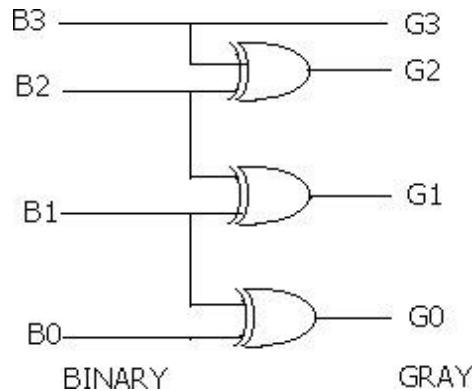
$$G_1 = B_1 \oplus B_2$$

B1B0	00	01	11	10
B3B2	00	01	11	10
00	0	1 ¹	3	1 ²
01	4	1 ⁵	7	1 ⁶
11	12	1 ³	15	1 ⁴
10	8	1 ⁹	11	1 ¹⁰

$$G_0 = \bar{B}_1 B_0 + B_1 \bar{B}_0$$

$$G_0 = B_1 \oplus B_0$$

Binary to Gray code converter Using XOR Gates Only



$$G_3 = B_3$$

$$G_2 = \bar{B}_3 B_2 + B_3 \bar{B}_2$$

$$G_2 = B_3 \oplus B_2$$

$$G_1 = B_2 \bar{B}_1 + \bar{B}_2 B_1$$

$$G_1 = B_1 \oplus B_2$$

$$G_0 = \bar{B}_1 B_0 + B_1 \bar{B}_0$$

$$G_0 = B_1 \oplus B_0$$

6.b. Design and implementation of Gray to Binary code converters using logic gates AIM

To design and implementation of Gray to Binary code converters using Multisim.

SOFTWARE REQUIRED

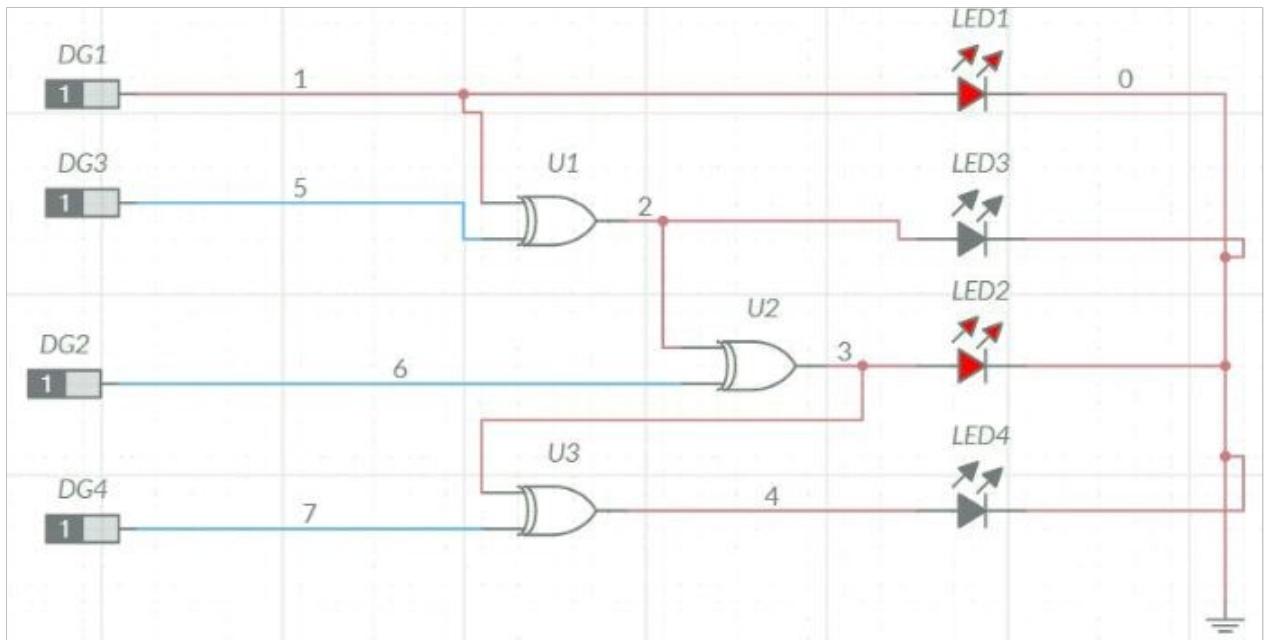
<https://www.multisim.com/>

PROCEDURE

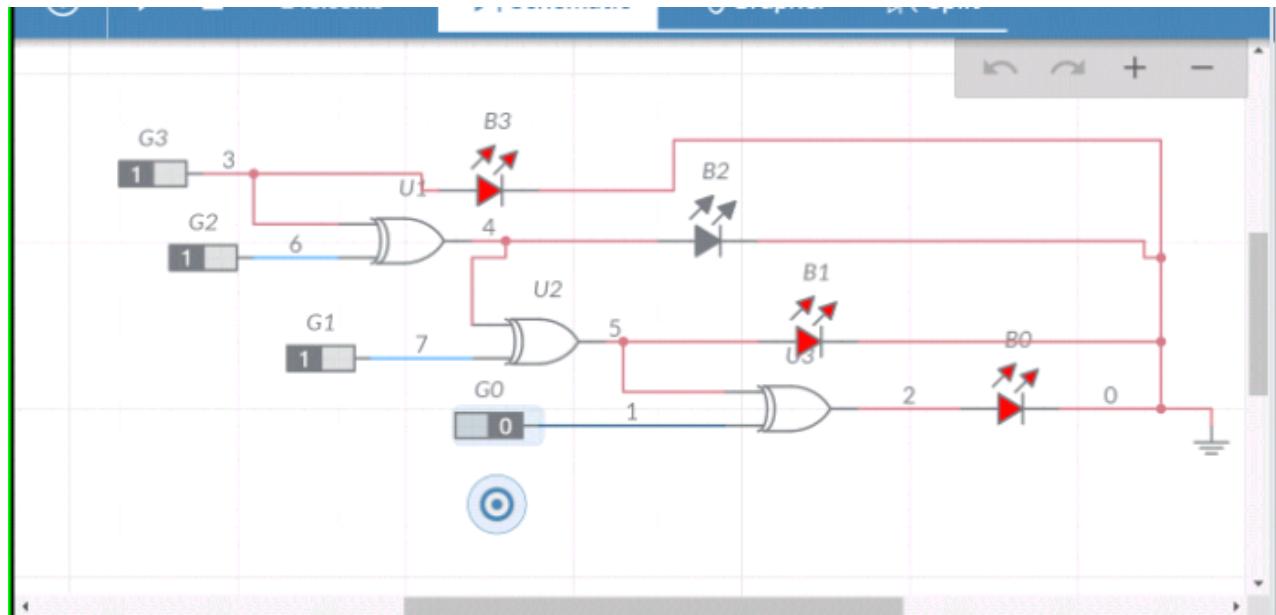
1. Open Multisim. Sign in.
2. Open a new circuit file
3. Select the components
 - Go to digital, choose digital constants
 - Go to digital, choose XOR 2 input gate
 - Go to indicator, choose LED bulbs
 - Go to schematic connectors, choose ground
4. Duplicate the components and connect them as per circuit diagram.
5. From analysis and annotation choose digital probe and place them where the output bits are to be seen
6. Run the simulation and verify the output

7. To change the input values, vary the values of high and low on the digital constants and verify the complete truth table illustrated below.

CIRCUIT DIAGRAM



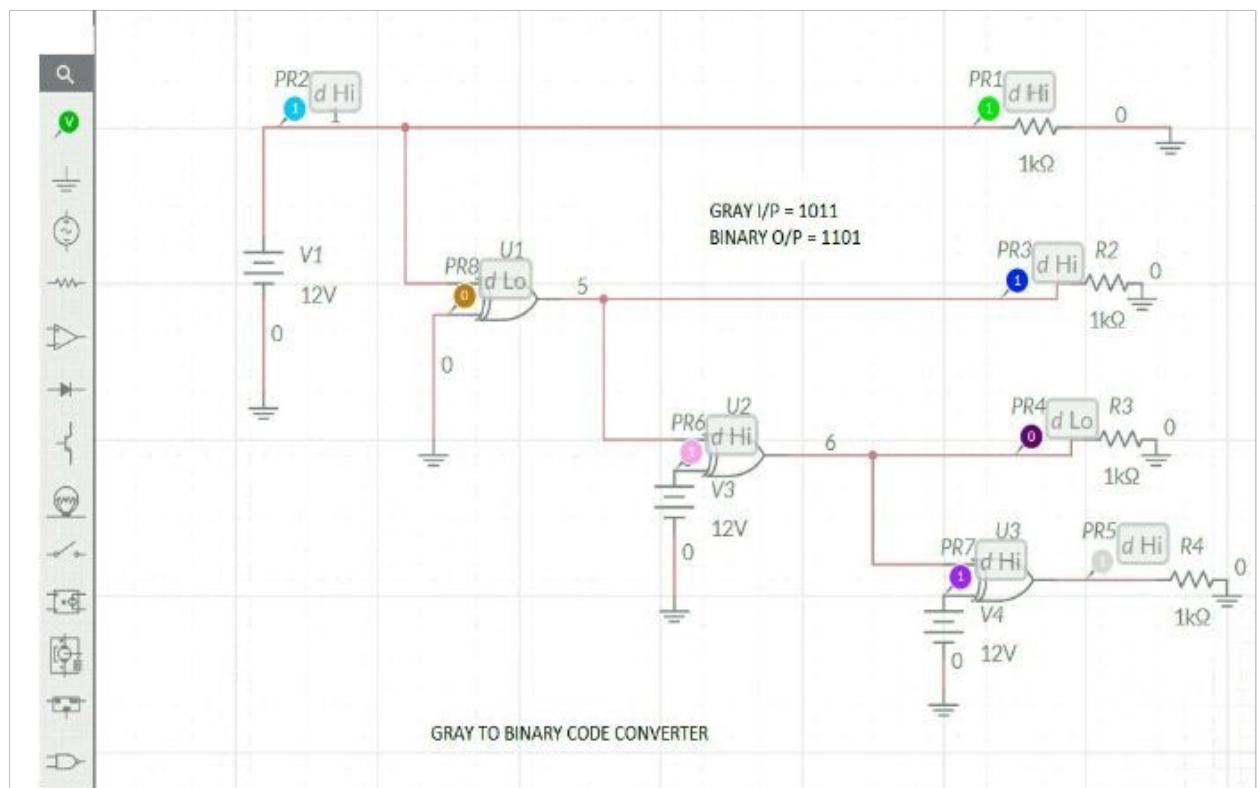
MY DIAGRAM:



ALTERNATE WAY PROCEDURE:

1. Open Multisim.
2. Select file navigation menu → New file
3. Select the components
 - a. Go to source choose DC source and give its value as 5V

- b. Go to digital choose XOR 2 input gate
 - c. Go to passive choose resistor and give its value as 100Ω .
 - d. Go to schematic connectors and choose ground
4. Duplicate the components and connect them as per circuit diagram.
5. From analysis and annotation choose digital probe and place them where the output bits are to be seen.
6. Run the simulation and verify the output
7. To change the input values connect it to the dc source or connect it to ground and verify the complete truth table given below.



TRUTH TABLE

GRAY CODE				BINARY CODE			
G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

K MAP FOR B3

G3G2		G1G0			
		00	01	11	10
G1G0	00	0	1	1 ³	1 ²
	01	4	5	1 ⁷	1 ⁶
	11	12	13	1 ⁵	1 ⁴
	10	8	9	1 ¹¹	1 ¹⁰

$$B3 = G3$$

K MAP FOR B2

G3G2		G1G0			
		00	01	11	10
G1G0	00	0	1 ¹	3	1 ²
	01	4	1 ⁵	7	1 ⁶
	11	12	1 ³	15	1 ⁴
	10	8	1 ⁹	11	1 ¹⁰

$$B2 = \overline{G3}G2 + G3\overline{G2}$$

$$B2 = G3 \oplus G2$$

K MAP FOR B1

		G3G2	00	01	11	10
		G1G0	00	01	11	10
G1G0	00	0	1	1	1	2
		4	5	7	6	
G1G0	01	12	3	15	14	
		18	19	11	10	

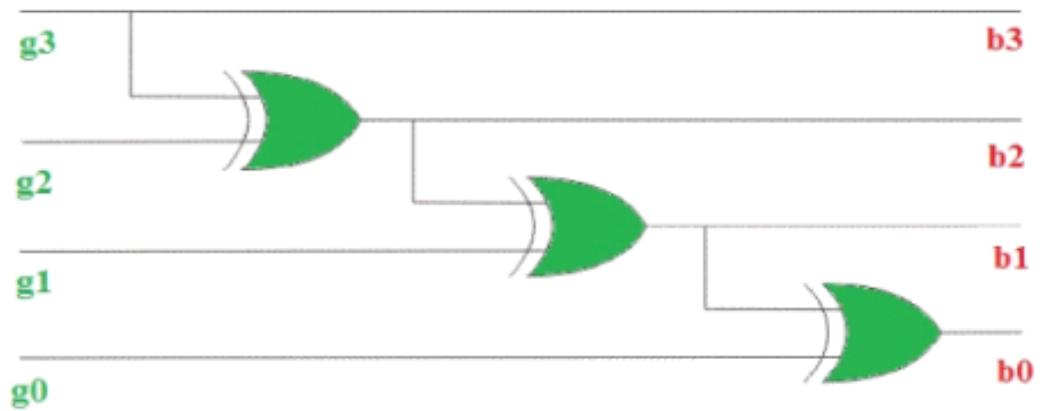
$$\begin{aligned}
 B1 &= \overline{G_1}G_0G_3 + \overline{G_1}G_0\overline{G_3} + G_1G_0G_3 + G_1\overline{G_0}\overline{G_3} \\
 &= \overline{G_1}(G_0G_3 + \overline{G_0}G_3) + G_1(G_0G_3 + \overline{G_0}G_3) \\
 &= G_1(G_0 \oplus G_3) + G_1(\overline{G_0} \oplus G_3) \\
 B1 &= G_3 \oplus G_2 \oplus G_1
 \end{aligned}$$

K MAP FOR B0

		G3G2	00	01	11	10
		G1G0	00	01	11	10
G1G0	00	0	1	3	1	2
		1	5	17	16	
G1G0	01	12	3	15	14	
		8	9	11	10	

$$\begin{aligned}
 B0 &= \overline{G_1}\overline{G_0}G_3G_2 + G_1\overline{G_0}G_2G_3 + G_0G_3G_2 + G_1G_0\overline{G_3} + \overline{G_1}G_0G_3 + \overline{G_1}G_3G_2 \\
 B0 &= G_0 \oplus G_1 \oplus G_2 \oplus G_3
 \end{aligned}$$

Gray to Binary code converter Using XOR Gates Only



6.C. Hardware Implementation of Code Converters Using Virtual Lab – IIT Bombay

Introduction to Virtual Lab – IIT Bombay

The objective of VLabsDev is to involve the community to re-think on the best practices regarding pedagogy, storyboards, lab manuals, documentation and the technologies for building a high-quality simulator.

The main activities of the VLabsDev portal currently are content creation through community, hosting the community created content, the source codes, hackathons for code and content development, events for instructors for pedagogy and storyboard design.

The academic community of students and instructors from Engineering, Science and Polytechnic institutes are the potential stakeholders as well as the beneficiaries. The objective is to provide quality Virtual Lab experiences for education.

The data collected through major universities in India indicates that, there are over 500 Labs which need to be created; (only ~120 labs of the MHRD Virtual Labs project are available for use). Therefore, there exists a huge gap in the syllabus which has to be filled by the community itself.

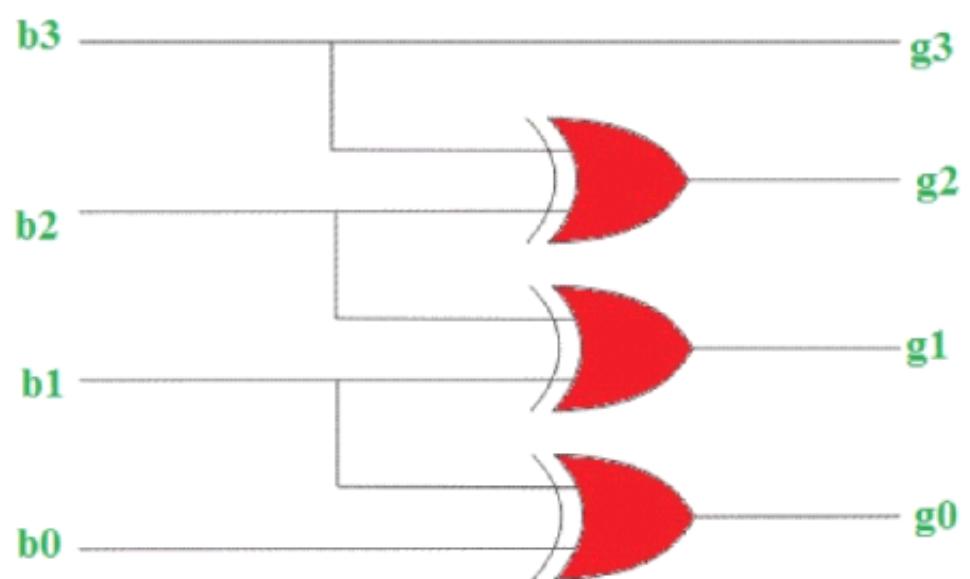
AIM

Hardware Implementation of the binary to gray code converter with virtual Lab - IIT Bombay

SOFTWARE REQUIRED virtual Lab - IIT Bombay [http://vlabs.iitb.ac.in/vlabsdev/vlab_bootcamp/
bootcamp/cool_developers/labs/exp2/BCD.html](http://vlabs.iitb.ac.in/vlabsdev/vlab_bootcamp/bootcamp/cool_developers/labs/exp2/BCD.html)

THEORY

The conversion of Binary to Gray code can be done by using a logic circuit. The gray code is a non-weighted code because there is no particular weight is assigned for the position of the bit. A n-bit code can be attained by reproducing a n-1-bit code on an axis subsequent to the rows of 2^{n-1} , as well as placing the most significant bit of 0 over the axis with the most significant bit of 1 beneath the axis. The step by step Gray code generation is shown below.

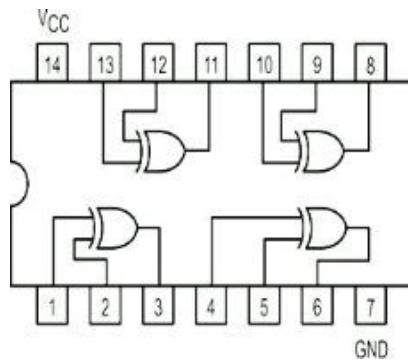


TRUTH TABLE

BINARY				GRAY CODE			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

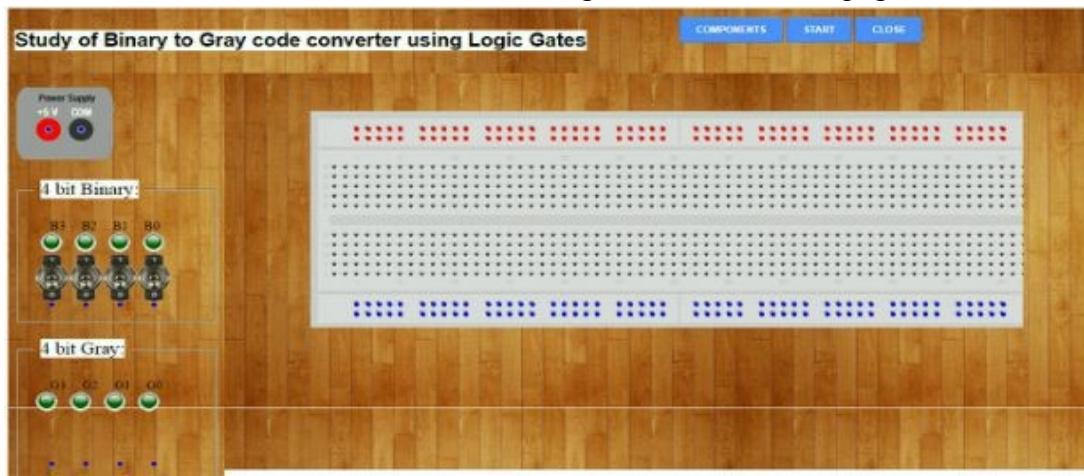
PROCEDURE

1. Choose the IC to use in the experiment – place the component on the breadboard.
2. Connect the inputs of the IC to the logic sources (A, B) and its output to the logic indicators (S, C) – make the wire connections between the various components.
3. Apply various input combinations and observe output for each one – use the DIP switches to apply various input combinations and note the LED indicators.
4. Verify the truth table for each input/output combination.
5. Exit the experiment when done – use the “Close” button to exit.
6. Build the Binary to Gray Circuit in the breadboard.
 - ✓ Use the below pin diagram for circuit connection in breadboard.

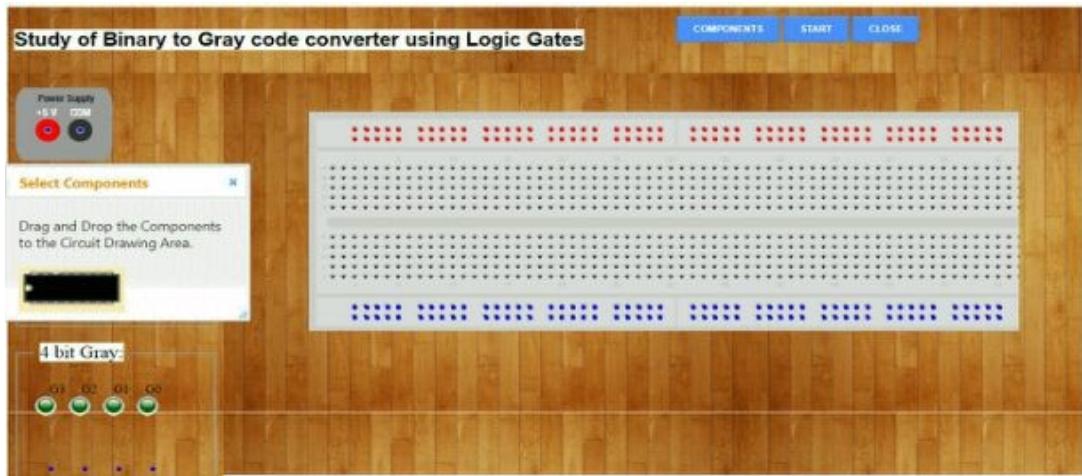


DETAILED PROCEDURE

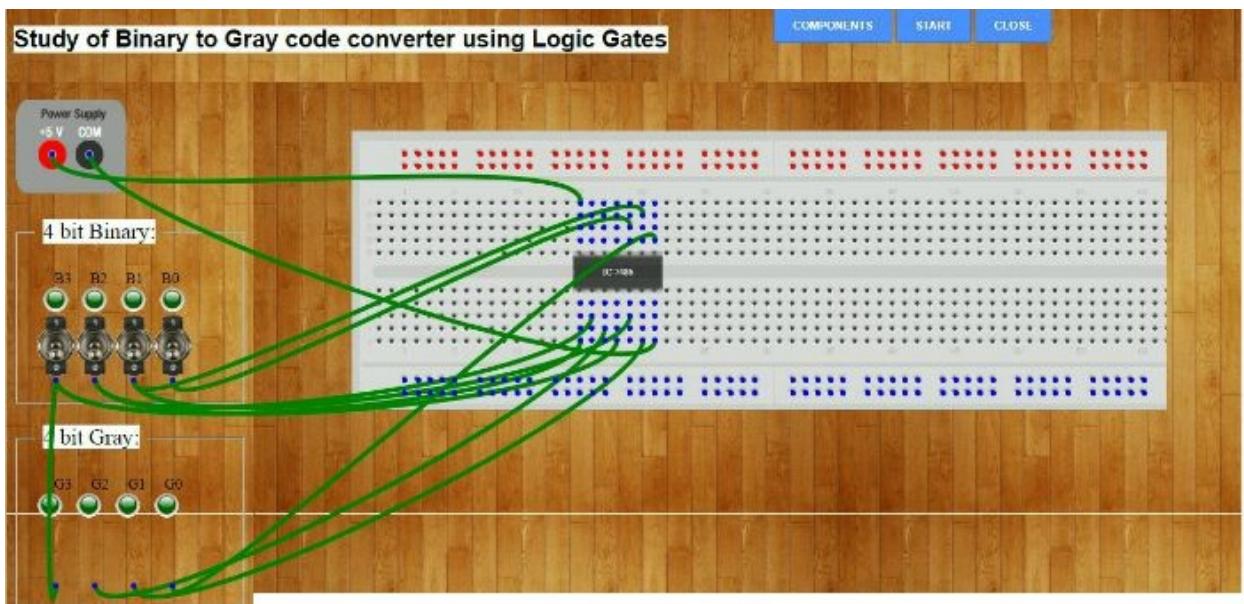
1. Search <http://vlabs.iitb.ac.in/vlabs-dev/> in the google.
2. Select “Approved labs” under “Explore Labs” tab in the VlabsDev page.
3. Click the “Digital Electronics” under Electronics and Communications.
4. Click ‘Explore’ the Digital Electronics page.
5. Select the experiment “To implement Binary to Gray code converter using basic and universal gates”
6. Click Simulation and then click ‘here’ to go to the simulation page as shown below.



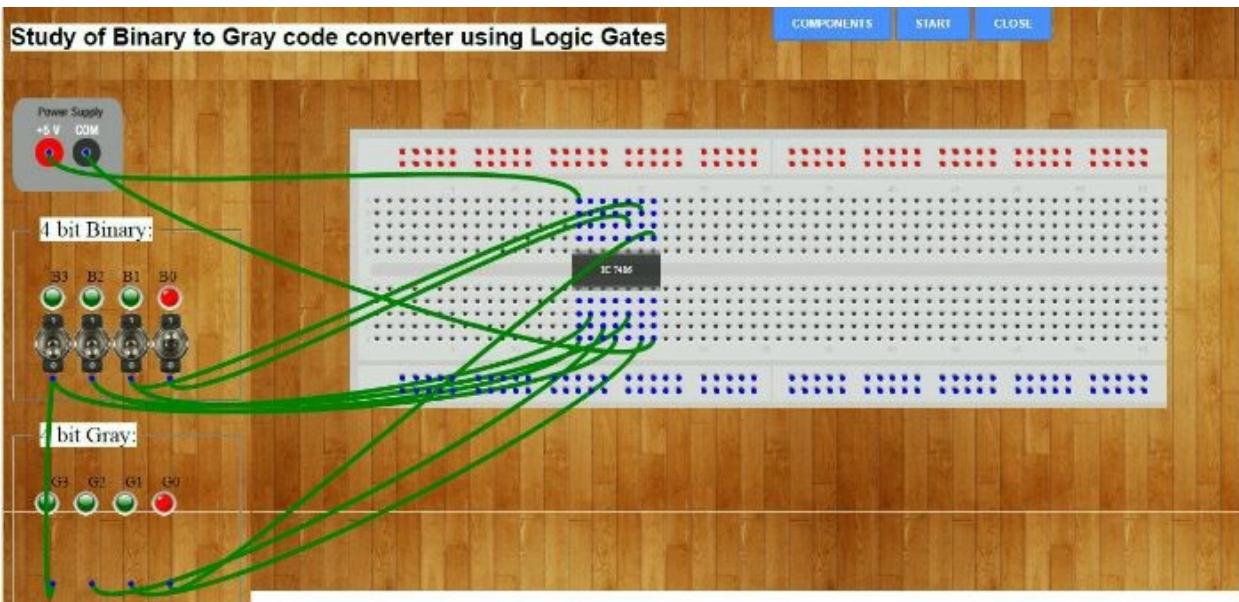
7. Click the ‘Components’ to get the required IC for the experiment as shown below.



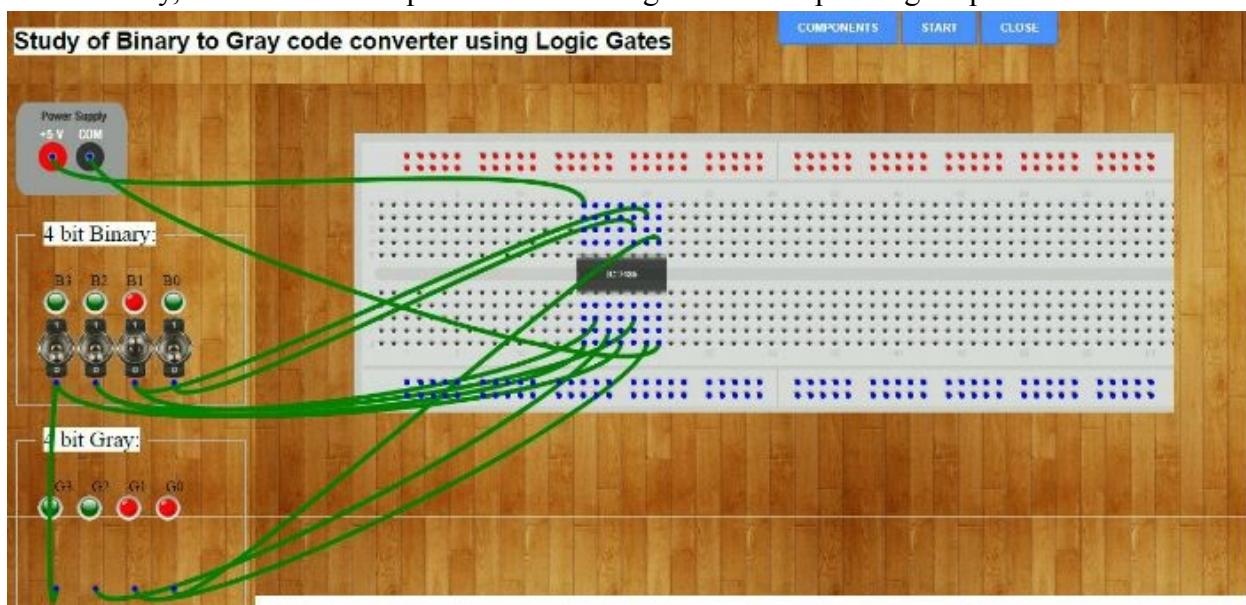
8. Drag and drop the IC in the bread board for making the connections and make the circuit as shown below using the Binary to gray code converter circuit for four inputs.



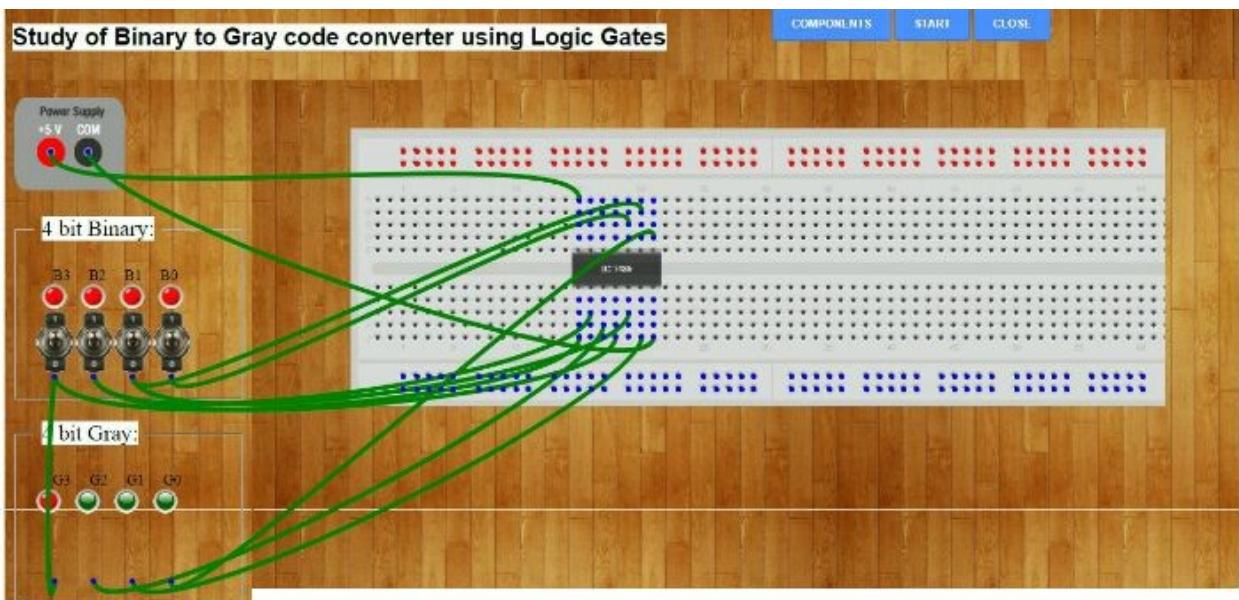
9. After making the circuit, click 'start' to verify the output for the input. For all zero inputs (B3B2B1B0), the output is all zero which is shown through output LEDs (G3G2G1G0) in the above picture. Here Green color represents zero and the Red color represents one.



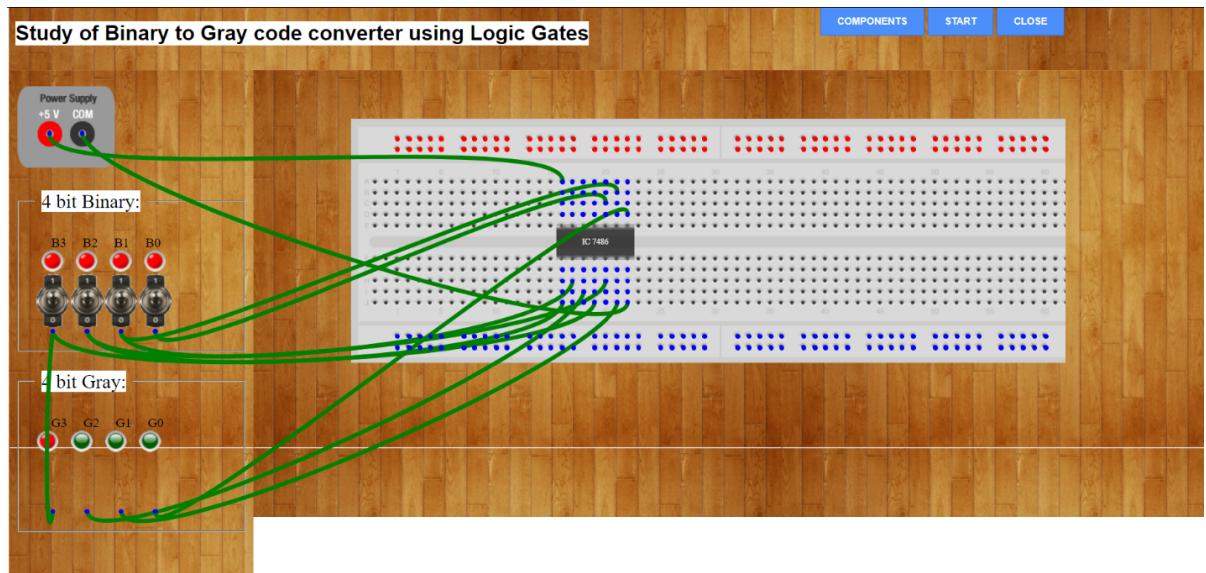
10. To verify the output for next set of inputs (0001), the input LED B0 is made one (RED) by pressing the corresponding switch and other LEDs are kept zero (GREEN). Now press 'start' button to get the output (0001) as shown in the above picture.
11. Similarly, the third set of inputs is checked to get the corresponding output as shown below.



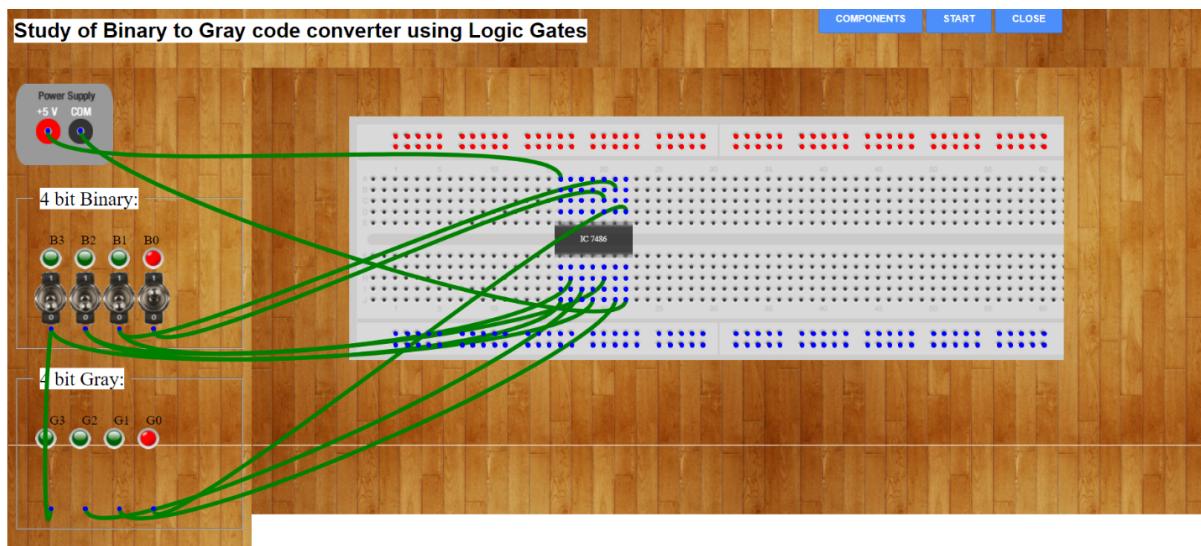
12. Outputs of the binary to gray code converter are verified for the remaining inputs using the truth table.
13. For the last set of input (1111), the output is 1000 which is verified through below connection diagram.



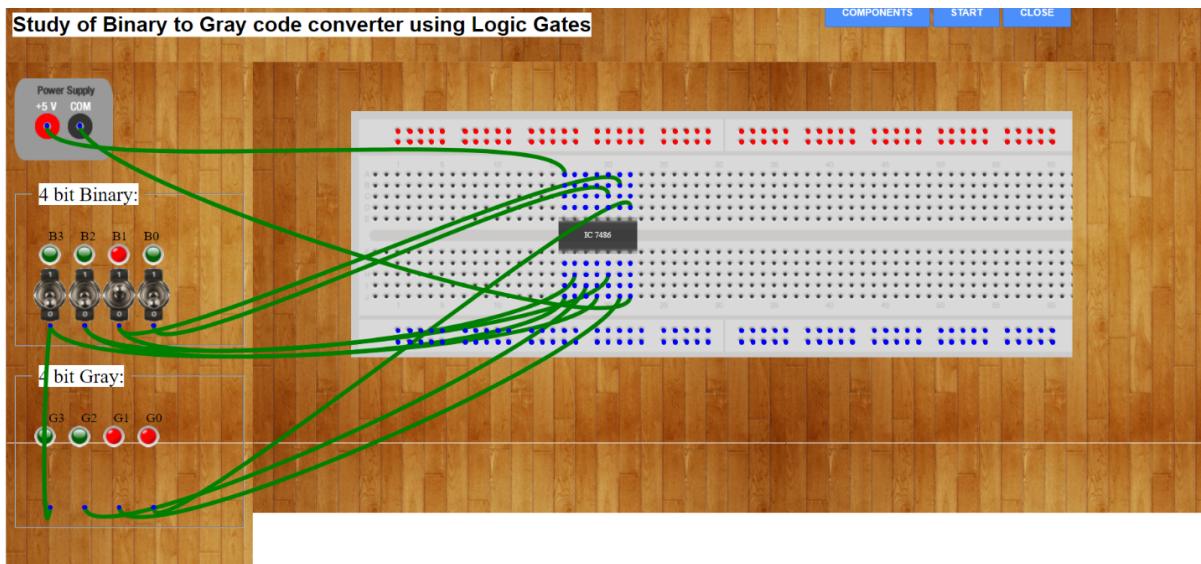
14. After verifying the outputs for their inputs, press ‘close’ button to end the simulation.



OUTPUT 1



OUTPUT 2



OUTPUT 3

RESULT

The design and implementation of Binary to gray code converters and Vice Versa using online Multisim and design and implementation of Binary to gray code converters using virtual Lab - IIT Bombay was verified.