

**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** |  |
| **Register Number** |  |
| **Date of Experiment** |  |

**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**

# 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/**](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**

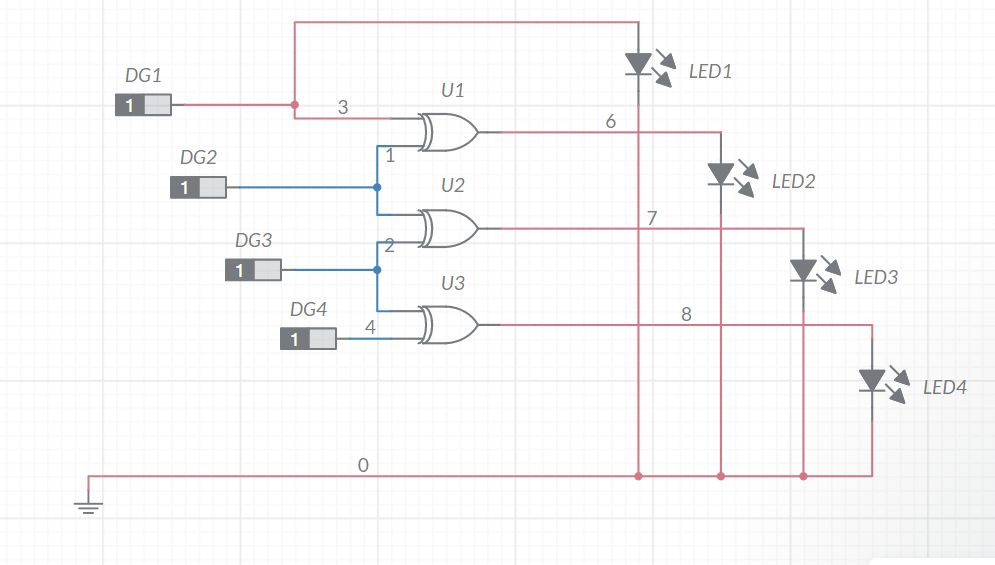
The logical circuit which converts binary code to equivalent gray code is known as binary to gray code converter. The gray code is a non-weighted code. The successive gray code differs in one-bit position only that means it is a unit distance code. It is also referred as cyclic code. It is not suitable 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 2n-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](https://www.elprocus.com/basic-logic-gates-with-truth-tables/) to generate the corresponding gray coded binary digit.

## PROCEDURE

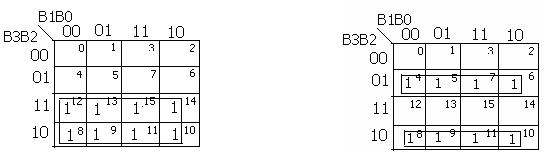
1. Open Multisim. Sign in.
2. Open a new circuit file
3. Select the components
   * Go to digital, choose digital constants
   * Go to digital, choose XOR 2 input gate
   * Go to indicator, choose LED bulbs
   * Go to schematic connectors, choose ground
4. Duplicate the components and connect them as 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:

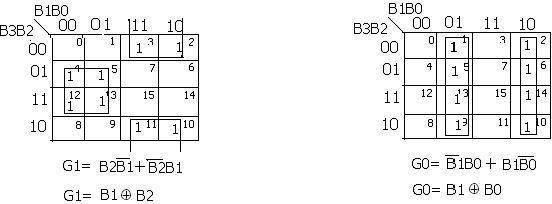
****

## TRUTH TABLE

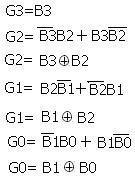
| **BINARY** | | | | **GRAY CODE** | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **B3** | **B2** | **B1** | **B0** | **G3** | **G2** | **G1** | **G0** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0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

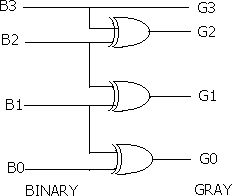
G3 = ∑(8,9,10,11,12,13,14,15) G2 = ∑(4,5,6,7,8,9,10,11)



G1= ∑(2,3,4,5,10.,11,12,13) G0 = ∑(1,2,3,5,6,9,10,13,14)

**Binary to Gray code converter Using XOR Gates Only**





# 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/**](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

## 

## RESULT

The design and implementation of Binary to gray code converters and Vice Versa using online Multisim is simulated and verified.