|  |  |  |
| --- | --- | --- |
| Department of Electrical Engineering | Document Number: SNU/MAN/EHS/EL/003 | Version: 2.0  Effective Date: 12/08/2024 |
| Title: Digital Electronics Lab Manuals (ECE 103) | | Pages 4 |

**Instructors:** Dr. Rakesh Palisetty

Dr. Upendra Kumar Pandey

Dr. Amitabh Chatterjee

**Experiment 4**

# Verification of K-Map and code convertors along with basic understanding of Multiplexer using gates.

## PART A: Implementation of K-Map

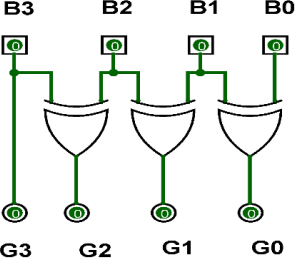
Reduce the expression and implement using NAND gate only.

F (A, B, C, D) = ∑ m (1,3,4,6,9,11,12,14)

## PART B: Verify Binary to Gray and Gray to Binary conversion using NAND gates only.

Binary Numbers is default way to store numbers, but in many applications binary numbers are difficult to use and a variation of binary numbers is needed. Gray code is an ordering of the binary numeral system such that two successive values differ in only one bit (binary digit). Gray codes are very useful in the normal sequence of binary numbers generated by the hardware that may cause an error or ambiguity during the transition from one number to the next. So, the gray code can eliminate this problem easily since only one bit changes its value during any transition between two numbers.

# Binary to Gray conversion:

* 1. The Most Significant Bit (MSB) of the gray code is always equal to the MSB of the given binary code.
  2. Other bits of the output gray code can be obtained by Ex-ORing binary code bit at that index and previous index. There are four inputs and four outputs. The input variable is defined as B3, B2, B1, B0 and the output variables are defined as G3, G2, G1, G0. From the truth table, combinational circuit is designed. Expression is defined as:

|  |  |
| --- | --- |
| **B3 = G3** |  |
| **B2** ⊕ **B3 = G2** |  |
| **B1** ⊕ **B2 = G1** |  |
| **B0** ⊕ **B1 = G0** |  |
|  | Figure-1: Binary to Gray Code Converter Circuit |
|  |  |

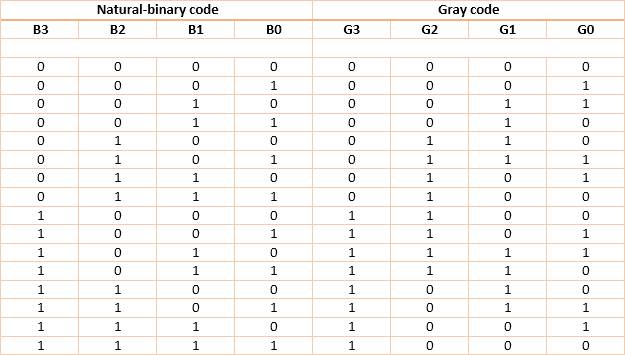


Table 1: Binary to Gray Code Converter Truth Table

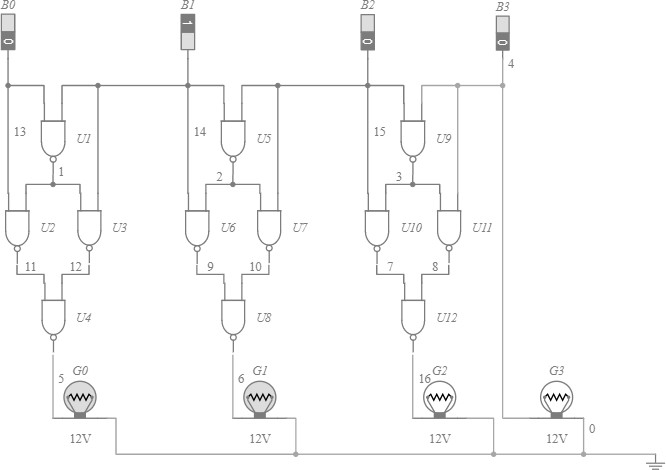


Figure.2 NAND gate realization for Binary to Gray Code Converter

## Gray to binary conversion:

1. The Most Significant Bit (MSB) of the binary code is always equal to the MSB of given binary number.
2. Other bits of the output binary code can be obtained by checking gray code bit at that index. If current gray code bit is 0, then copy previous binary code bit, else copy invert of previous binary code bit. There are four inputs and four outputs. The input variable is defined as G3, G2, G1, G0 and the output variables are defined as B3, B2, B1, B0. From the truth table, combinational circuit is designed. The expression is given as,

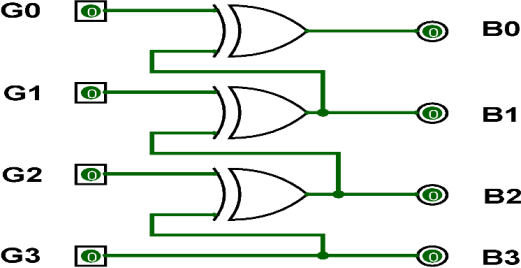


Figure-3: Gray to Binary Code Converter Circuit

**G0** ⊕ **G1** ⊕ **G2** ⊕ **G3 = B0 G1** ⊕ **G2** ⊕ **G3 = B1**

**G2** ⊕ **G3 = B2 G3 =** B3

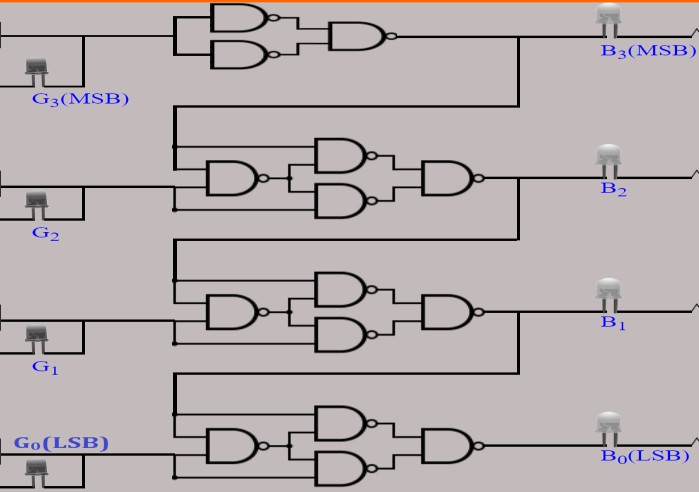


Figure 4. NAND gate realization for Gray Code to binary Converter

## PART C: Multiplexers using Basic Logic Gates

A digital multiplexer is similar to a multi-position switch with many inputs and only one output. It has control inputs to select a particular input. It is also called as a channel selector and abbreviated as Mux.

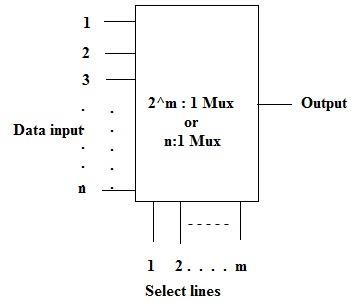
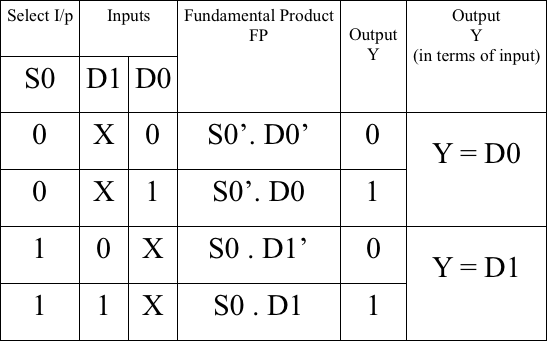


Figure 5. Block diagram of multiplexer

|  |  |  |  |
| --- | --- | --- | --- |
| **No. of select lines (m)** | **No. of inputs (2^m)** | **No. of Outputs Y** | **Multiplexer Type** |
| 1 | 2 | 1 | 2:1 |
| 2 | 4 | 1 | 4:1 |
| 3 | 8 | 1 | 8:1 |
| 4 | 16 | 1 | 16:1 |
|  |  |  |  |
|  |  |  |  |

It is easier to build multiplexers using gates (small scale integration -SSI ICs) for a few select lines. But as the number of select lines increases, use of medium scale integration MSI ICs becomes the best choice. TTL IC 74LS150 is a 16:1 Mux. In this experiment design of smaller size Muxes are considered. A 2:1(read as 2 as to 1) multiplexer can be designed using: a. basic logic gates b. universal gates (NAND & NOR).

**Example 1**: Design of 2:1 Mux using basic logic gates A simple 2:1 Mux will have 2 input lines D0 & D1 and one select line S0 and a single output Y. The select line can take a value either 0 or 1: a. If S0 takes a value 0, the input D0 is selected and the output Y = D0. b. If S0 takes a value 1, the input D1 is selected and the output Y = D1.

* 1. **Prepare the function table of the 2:1 Mux**
  2. **Formulate the expression for output Y by considering only those FPs for which the output is 1. Y = S0’. D0 + S0. D1**

**3**

c. **Draw the logic diagram for the expression:**

A diagram of a logic board

Description automatically generated

**d.Verify the truth table.**

## Requirements for above all parts:

* + - AND Gate IC: 74HC08
    - NOT Gate IC: 74HC04
    - NAND Gate IC: 74HC00
    - OR Gate IC: 74HC32

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**4**