Code Converters
(Binary to Gray & Gray to Binary)

Experiment No.05

Aim: Design a combinational circuit with four inputs and four outputs that converts a four bit 'Gray Code' in to the equivalent four bit 'Binary Number'. Implement the circuit with XOR gates, only. Connect the circuit to four switches and four indicator lamps, and test it for proper operation.

The conversion from one binary code to another is common in digital systems. In this experiment, you will design and construct 'Combinational Circuit Converters'.

Binary Numbers: The binary number system is a 'Base-2 System'. It consists of only two digits, e.g., 0 & 1. The position of 0 and 1 indicates the weight within the number. The weight of each successively higher position to the left is an increasing power of two.

Gray Code: This non-weighted code belongs to a class of codes called 'Minimum-Change Code' in which only one bit in the code group changes when moving from one step to the next.

Activity-1. Binary to Gray Code Converter

The block diagram of a 4-bit binary-to-gray code converter is shown in fig. below.
It has four bit-binary inputs $B(B_3B_2B_1B_0)$ and four-bit gray code outputs $G(G_3G_2G_1G_0)$.

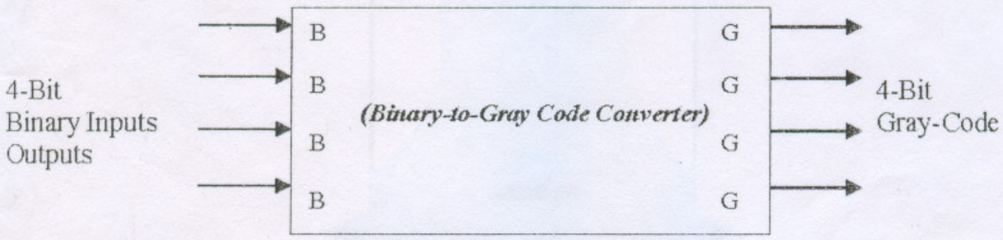


Fig. Block diagram of 4-bit binary-to-gray code converter

Design: To design and Binary-to-Gray code converter, follow the steps given below.

Step.1. Truth-Table

Step.2. Logical Expressions

From the truth-table, write the logical expressions for the gray code outputs.

$G_3=\Sigma_m$ ....

$G_2=\Sigma_m$ ....

$G_1=\Sigma_m$ ....

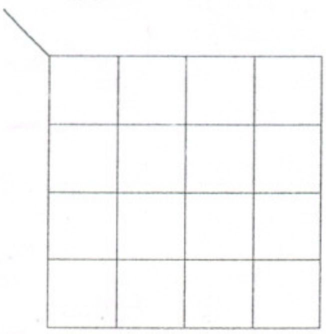
$G_0=\Sigma_m$ ....

Truth-Table

Binary Inputs				*Gray Code Outputs*			
B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

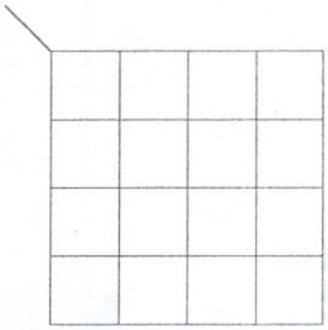
Step.3. Simplification of Logical Expressions
Simplify the logical expressions, using K-Map.

K-Map for G_3



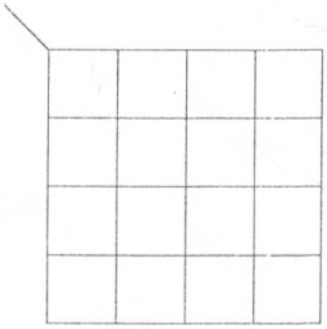
Hence, G_3= ....

K-Map for G_2



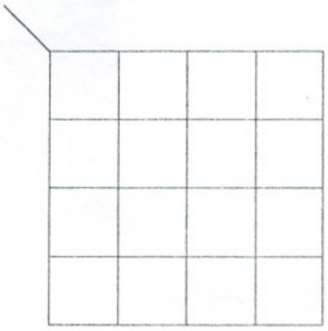
Hence, G_2= ....

K-Map for G_1



Hence, G_1= ....

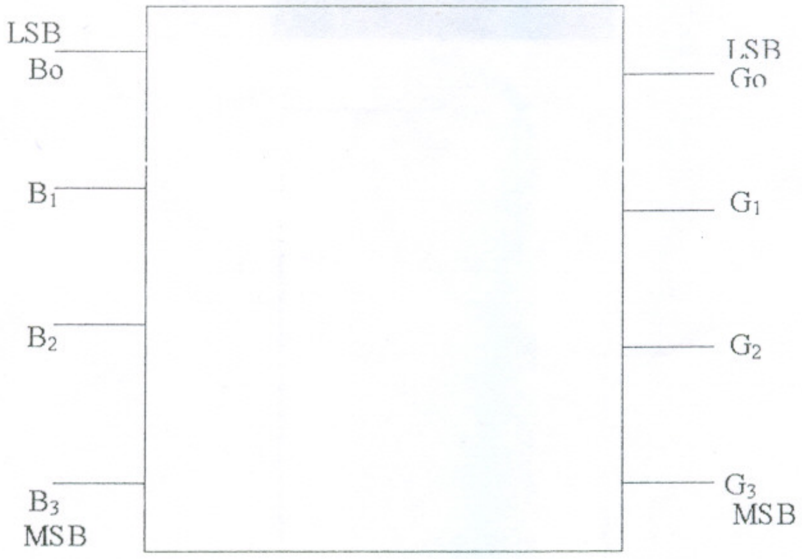
K-Map for G_0



Hence, G_0= ....

Step.4. Logic Diagram

Implement the logical expressions you find (in step3) after simplifying the K-maps, using only three XOR gates. Verify the truth-table for binary-to-gray code converter, you made in *step.1*.



Note: Do not disconnect this circuitry you implemented in the bread-board.

Write: How many ICs of XOR gate, you have used to implement this code converter. ....

Activity-2: Gray Code-to-Binary Converter

The block diagram of and 4-bit gray code-to-binary converter is shown below. It has four bit-gray code inputs G($G3G_2G_1G_0$) and four-bit binary outputs B($B_3B_2B_1B_0$).

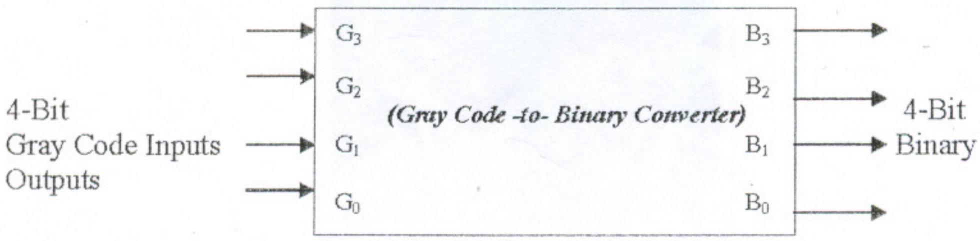


Fig. Block diagram of 4-bit gray code-to-binary converter

Design: To design a Gray code-to-Binary converter, follow the steps given below.

Step.1. Truth-Table

Gray Code Inputs				*Binary Outputs*			
G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

Step.2. Logical Expressions

From the truth-table, write the logical expressions for the binary outputs.

$B_3 = \Sigma_m$ ....

$B_2 = \Sigma_m$ ....

$B_1 = \Sigma_m$ ....

$B_0 = \Sigma_m$ ....

Step.3. Simplification of Logical Expressions

Using K-Map, simplify the above logical expressions.

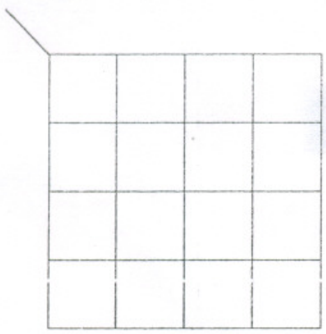
K-Map for B_3

Hence, $B_3=$ ....

K-Map for B_2

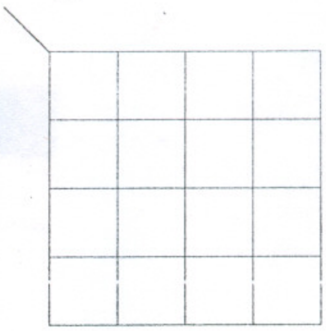
Hence, $B_2=$ ....

K-Map for B_1



Hence, B_1= ....

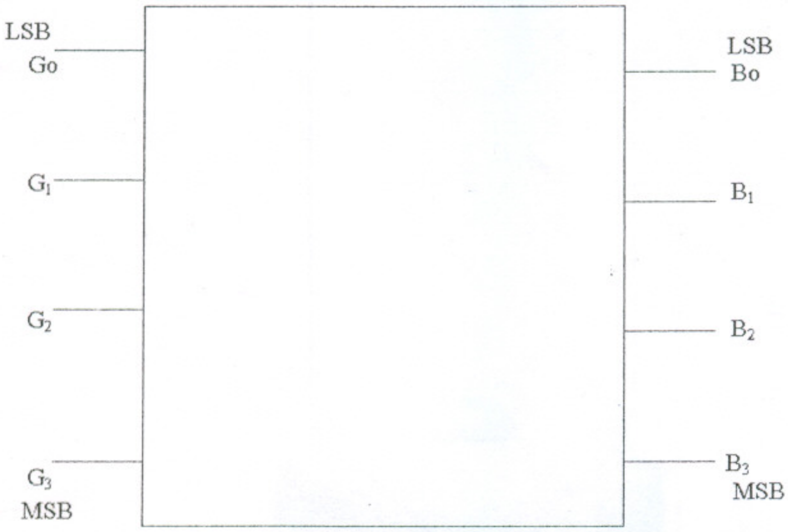
K-Map for B_0



Hence, B_0= ....

Step.4. Logic Diagram

Implement the logical expressions you find (in step3) after simplifying the K-maps, using only three XOR gates. Verify the truth-table for gray code-to-binary converter, you made in *step.1.*



Note: *Do not disconnect the circuitry you implemented in the bread-board.*

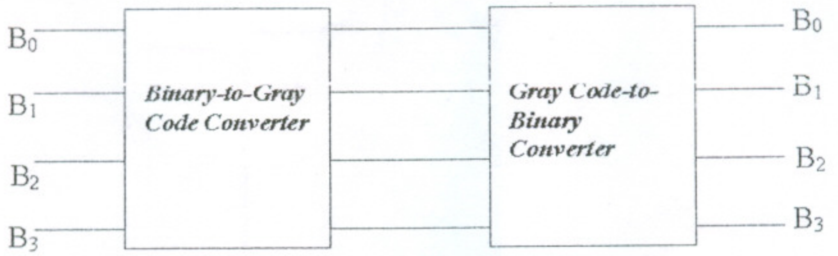
Write: How many ICs of XOR gate, you have used to implement this code converter. ....

Activity-3 Verification of Code Converter

Procedure

Follow the steps given below.

Step.1. Short, corresponding G outputs and G inputs of the two circuitries, as shown in the fig. below.



Step.2. Apply, 4-bit binary input $B(B_3B_2B_1B_0)$ to the Binary-to-Gray Code converter.

Step.3. Observe, the output displayed at the output of the Gray-to-Binary converter, whether they both are same or not. If not, recheck or redesign (if necessary) the circuitry/circuitries.

Give Comments

....
....
....
....
....
....
....
....