Practical No 4
Implement code converters

Aim:

a) To design and implement Binary – to – Gray code converter.
b) To design and implement Gray – to – Binary code converter.
c) To design and implement Binary – to – BCD code converter.
d) To design and implement Binary – to – XS-3 code converter.

Theory:

a) **Binary – to – Gray code converter.**

Binary Code: A number system with two digits, 0 and 1. Each digit represents a power of 2.

Gray Code: A binary code where consecutive numbers differ by only one bit. This property is useful in applications like analog-to-digital converters (ADCs) to prevent spurious outputs.

Conversion Process
To convert a binary number to its equivalent Gray code, follow these steps:
The Most Significant Bit (MSB) of the Gray code is the same as the MSB of the binary number.
For the remaining bits of the Gray code, perform an XOR (exclusive OR) operation between the corresponding bit and the previous bit of the binary number.

Example: Convert binary number 1011 to Gray code.
MSB of Gray code = MSB of binary code = 1
Second bit of Gray code = XOR of first and second bits of binary code = 0 XOR 1 = 1
Third bit of Gray code = XOR of second and third bits of binary code = 1 XOR 1 = 0
Fourth bit of Gray code = XOR of third and fourth bits of binary code = 1 XOR 1 = 0
Therefore, the Gray code equivalent of 1011 is 1100.

For designing a Binary-to-Gray code converter we first draw the truth table, we assume the binary code to be of 4-bits and the corresponding gray code to be of 4-bits
There are 4-inputs and 4utputs, so we need to draw four K-maps for designing the given code converter
The inputs are $B_3B_2B_1B_0$ and the outputs are $G_3G_2G_1G_0$

Prof. Ismail H. Popatia

Truth table

Input				Output			
B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
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

We draw k-map for each output and then implement

K-map for G_3

	$\overline{B}_3.\overline{B}_2$	$\overline{B}_3.B_2$	$B_3.B_2$	$B_3.\overline{B}_2$
$\overline{B}_1.\overline{B}_0$	0	0	1	1
$\overline{B}_1.B_0$	0	0	1	1
$B_1.B_0$	0	0	1	1
$B_1.\overline{B}_0$	0	0	1	1

From the K-map we see that the equation of G_3 is

$$G_3. = B_3$$

Prof. Ismail H. Popatia

K-map for G_2

	$\overline{B}_3.\overline{B}_2$	$\overline{B}_3.B_2$	$B_3.B_2$	$B_3.\overline{B}_2$
$\overline{B}_1.\overline{B}_0$	0	1	0	1
$\overline{B}_1.B_0$	0	1	0	1
$B_1.B_0$	0	1	0	1
$B_1.\overline{B}_0$	0	1	0	1

From the K-map we see that the equation of G_2 is

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

$$G_2. = B_2 \oplus B_3$$

K-map for G_1

	$\overline{B}_3.\overline{B}_2$	$\overline{B}_3.B_2$	$B_3.B_2$	$B_3.\overline{B}_2$
$\overline{B}_1.\overline{B}_0$	0	1	1	0
$\overline{B}_1.B_0$	0	1	1	0
$B_1.B_0$	1	0	0	1
$B_1.\overline{B}_0$	1	0	0	1

From the K-map we see that the equation of G_1 is

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

$$G_1. = B_1 \oplus B_2$$

Prof. Ismail H. Popatia

K-map for G_0

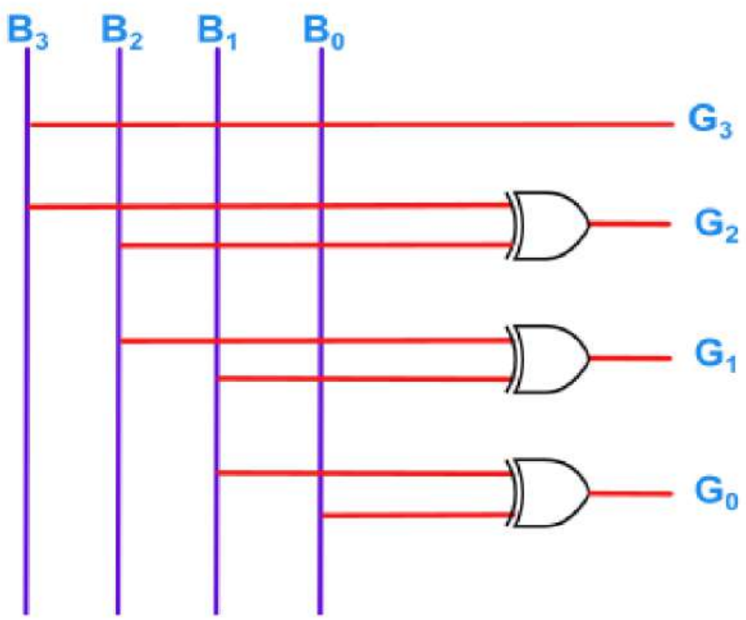
	$\overline{B}_3.\overline{B}_2$	$\overline{B}_3.B_2$	$B_3.B_2$	$B_3.\overline{B}_2$
$\overline{B}_1.\overline{B}_0$	0	0	0	0
$\overline{B}_1.B_0$	1	1	1	1
$B_1.B_0$	0	0	0	0
$B_1.\overline{B}_0$	1	1	1	1

From the K-map we see that the equation of G_0 is

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

$$G_0. = B_0 \oplus B_1$$

<u>Circuit Diagram</u> : Hence the circuit diagram for Binary-to-gray code converter is



Prof Ismail H. Popatia

b) **Gray – to – Binary code converter.**

Truth table

Input				Output			
G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
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

We draw k-map for each output and then implement

K-map for B_3

	$\overline{G_3}.\overline{G_2}$	$\overline{G_3}.G_2$	$G_3.G_2$	$G_3.\overline{G_2}$
$\overline{G_1}.\overline{G_0}$	0	0	1	1
$\overline{G_1}.G_0$	0	0	1	1
$G_1.G_0$	0	0	1	1
$G_1.\overline{G_0}$	0	0	1	1

From the K-map we see that the equation of B_3 is

$$B_3. = G_3$$

Prof. Ismail Popatia

K-map for B_2

	$\overline{G_3}.\overline{G_2}$	$\overline{G_3}.G_2$	$G_3.G_2$	$G_3.\overline{G_2}$
$\overline{G_1}.\overline{G_0}$	0	1	0	1
$\overline{G_1}.G_0$	0	1	0	1
$G_1.G_0$	0	1	0	1
$G_1.\overline{G_0}$	0	1	0	1

From the K-map we see that the equation of B_2 is

$$B_2. = \overline{G_2}.G_3 + G_2.\overline{G_3}$$

$$B_2 = G_2 \oplus G_3$$

K-map for B_1

	$\overline{G_3}.\overline{G_2}$	$\overline{G_3}.G_2$	$G_3.G_2$	$G_3.\overline{G_2}$
$\overline{G_1}.\overline{G_0}$	0	1	0	1
$\overline{G_1}.G_0$	0	1	0	1
$G_1.G_0$	1	0	1	0
$G_1.\overline{G_0}$	1	0	1	0

From the K-map we see that the equation of B_1 is

$$B_1 = G_1(\overline{G_2}\overline{G_3} + G_2G_3) + \overline{G_1}(\overline{G_2}G_3 + G_2\overline{G_3})$$

$$B_1 = G_1(G_2 \odot G_3) + G_1(G_2 \oplus G_3)$$

$$B_1 = G_1 \oplus G_2 \oplus G_3$$

Prof. Ismail H. Popatia

K-map for B_0

	$\overline{G_3}.\overline{G_2}$	$\overline{G_3}.G_2$	$G_3.G_2$	$G_3.\overline{G_2}$
$\overline{G_1}.\overline{G_0}$	0	1	0	1
$\overline{G_1}.G_0$	1	0	1	0
$G_1.G_0$	0	1	0	1
$G_1.\overline{G_0}$	1	0	1	0

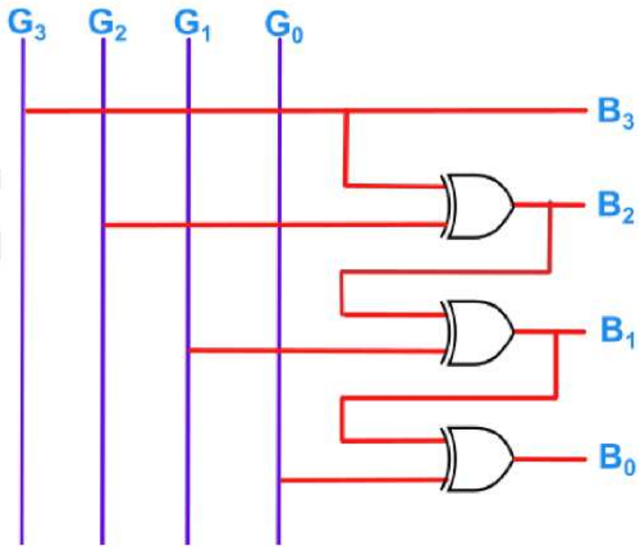
From the K-map we see that the equation of B_0 is

$$B_0 = G_0\overline{G_1}\,\overline{G_2}\,\overline{G_3} + \overline{G_0}G_1\overline{G_2}\,\overline{G_3} + \overline{G_0}\,\overline{G_1}G_2\overline{G_3} + G_0G_1G_2\overline{G_3}$$
$$+ G_0\overline{G_1}G_2G_3 + \overline{G_0}G_1G_2G_3 + \overline{G_0}\,\overline{G_1}\,\overline{G_2}G_3 + G_0G_1\overline{G_2}G_3$$

The above expression can be reduced to

$$B_0 = G_0 \oplus G_1 \oplus G_2 \oplus G_3$$

Circuit Diagram : Hence the circuit diagram for Gray-to-Binary code converter is



Prof Ismail H. Popatia

c) **Binary – to – BCD code converter.**

In this Binary-to-BCD converter, we are working with a 4-bit binary input, which can represent decimal values from 0 to 15. The BCD output is represented using 5 bits, where each digit of the decimal number is encoded in binary form.

Truth Table Construction:

- The truth table is the foundation of the conversion process. It lists all possible combinations of the 4 binary input bits ($B_3B_2B_1B_0$) and their corresponding BCD outputs ($D_4D_3D_2D_1D_0$).
- The 4-bit binary input represents values ranging from 0000 (0 in decimal) to 1111 (15 in decimal).
- For each binary input, the corresponding BCD output is written in a 5-bit format.

Truth table:

Input				Output				
B_3	B_2	B_1	B_0	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0	1
0	1	1	0	0	0	1	1	0
0	1	1	1	0	0	1	1	1
1	0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	0	1
1	0	1	0	1	0	0	0	0
1	0	1	1	1	0	0	0	1
1	1	0	0	1	0	0	1	0
1	1	0	1	1	0	0	1	1
1	1	1	0	1	0	1	0	0
1	1	1	1	1	0	1	0	1

Karnaugh Maps (K-Maps):

To minimize the logic required for the Binary-to-BCD conversion, K-maps are used for each of the 5 BCD output bits ($D_4D_3D_2D_1D_0$)

Each K-map is a visual representation of the truth table, where the input variables are mapped onto a grid. The goal is to group the '1's in the K-map to simplify the Boolean expressions for each output bit.

K-map for D_4

	$\overline{B_3}.\overline{B_2}$	$\overline{B_3}.B_2$	$B_3.B_2$	$B_3.\overline{B_2}$
$\overline{B_1}.\overline{B_0}$	0	0	1	0
$\overline{B_1}.B_0$	0	0	1	0
$B_1.B_0$	0	0	1	1
$B_1.\overline{B_0}$	0	0	1	1

The equation of the k-map is

$$D_4 = B_2B_3 + B_1B_3$$

K-map for D_3

	$\overline{B_3}.\overline{B_2}$	$\overline{B_3}.B_2$	$B_3.B_2$	$B_3.\overline{B_2}$
$\overline{B_1}.\overline{B_0}$	0	0	0	1
$\overline{B_1}.B_0$	0	0	0	1
$B_1.B_0$	0	0	0	0
$B_1.\overline{B_0}$	0	0	0	0

The equation of the k-map is

$$D_3 = \overline{B_1}\,\overline{B_2}B_3$$

K-map for D_2

	$\overline{B_3}.\overline{B_2}$	$\overline{B_3}.B_2$	$B_3.B_2$	$B_3.\overline{B_2}$
$\overline{B_1}.\overline{B_0}$	0	1	0	0
$\overline{B_1}.B_0$	0	1	0	0
$B_1.B_0$	0	1	1	0
$B_1.\overline{B_0}$	0	1	1	0

The equation of the k-map is

$$D_2 = B_1B_2 + B_2\overline{B_3}$$

Prof. Ismail Popatia

K-map for D_1

	$\overline{B_3}.\overline{B_2}$	$\overline{B_3}.B_2$	$B_3.B_2$	$B_3.\overline{B_2}$
$\overline{B_1}.\overline{B_0}$	0	0	1	0
$\overline{B_1}.B_0$	0	0	1	0
$B_1.B_0$	1	1	0	0
$B_1.\overline{B_0}$	1	1	0	0

The equation of the k-map is

$$D_1 = B_1\overline{B_3} + \overline{B_1}B_2B_3$$

K-map for D_0

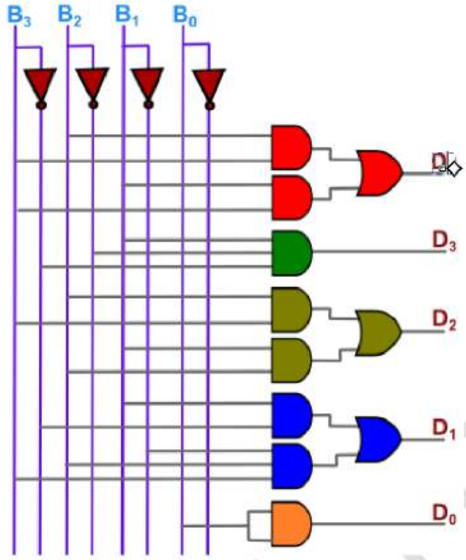
	$\overline{B_3}.\overline{B_2}$	$\overline{B_3}.B_2$	$B_3.B_2$	$B_3.\overline{B_2}$
$\overline{B_1}.\overline{B_0}$	0	0	0	0
$\overline{B_1}.B_0$	1	1	1	1
$B_1.B_0$	1	1	1	1
$B_1.\overline{B_0}$	0	0	0	0

The equation of the k-map is

$$D_0 = B_0$$

Prof. Ismail H. Popatia

Circuit Diagram : Hence the circuit diagram for Binary – to – BCD code converter is



d) **Binary – to – Excess-3 code converter.**

The 4-bit binary to 4-bit Excess-3 converter is a digital circuit that converts a 4-bit binary number into its corresponding Excess-3 code. Excess-3 (XS-3) is a non-weighted code used to express decimal digits. It is derived by adding 3 to the binary representation of each decimal digit.

Truth table:

Input				Output			
B_3	B_2	B_1	B_0	E_3	E_2	E_1	E_0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	1	1	1	1
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

In the above truth table the binary values 1101, 1110, and 1111 represent decimal values 13, 14, and 15. When adding 3 to these values, the results go beyond a single digit in Excess-3, so they don't directly translate into a 4-bit code.
In practice, for inputs 1101 to 1111, the output should indicate an invalid code, or we can truncate or mask them as per specific design requirements. If this is a BCD-based application, these values would be out of range and usually flagged as invalid or unused
Hence these values in the truth table are represented as don't care conditions (denoted by X)

Now we use the k-maps for the designing part

K-map for E_3

	$\overline{B_3}.\overline{B_2}$	$\overline{B_3}.B_2$	$B_3.B_2$	$B_3.\overline{B_2}$
$\overline{B_1}.\overline{B_0}$	0	0	1	1
$\overline{B_1}.B_0$	0	0	X	1
$B_1.B_0$	0	1	X	1
$B_1.\overline{B_0}$	0	1	X	1

The equation of the k-map is

$$E_3 = B_3 + B_1B_2$$

K-map for E_2

	$\overline{B_3}.\overline{B_2}$	$\overline{B_3}.B_2$	$B_3.B_2$	$B_3.\overline{B_2}$
$\overline{B_1}.\overline{B_0}$	0	1	1	0
$\overline{B_1}.B_0$	1	0	X	1
$B_1.B_0$	1	0	X	1
$B_1.\overline{B_0}$	1	0	X	1

The equation of the k-map is

$$E_2 = B_0\overline{B_2} + B_1\overline{B_2} + B_0\overline{B_1}B_2$$

Prof. Ismail H. Popatia

K-map for E_1

	$\overline{B}_3.\overline{B}_2$	$\overline{B}_3.B_2$	$B_3.B_2$	$B_3.\overline{B}_2$
$\overline{B}_1.\overline{B}_0$	1	1	1	1
$\overline{B}_1.B_0$	0	0	X	0
$B_1.B_0$	1	1	X	1
$B_1.\overline{B}_0$	0	0	X	0

The equation of the k-map is

$$E_1 = \overline{B}_0\overline{B}_1 + B_0B_1$$

$$E_1 = B_0 \odot B_1$$

K-map for E_0

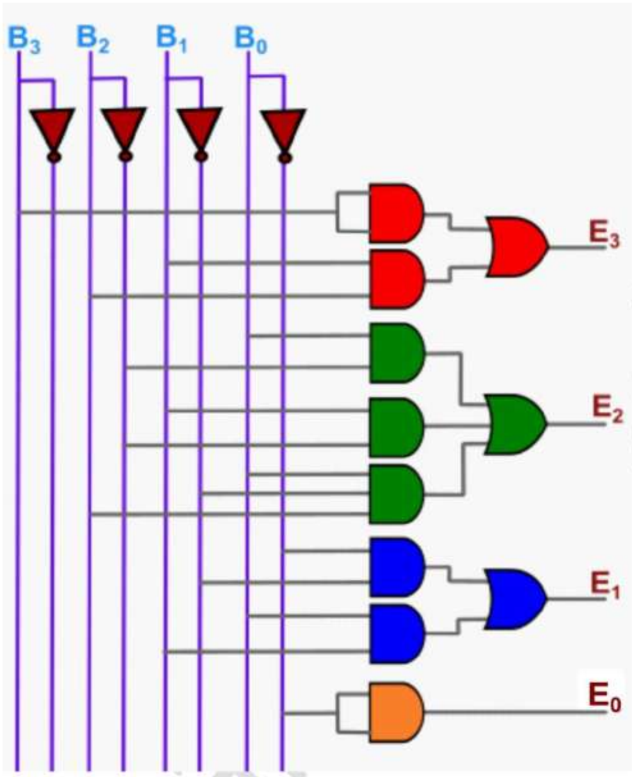
	$\overline{B}_3.\overline{B}_2$	$\overline{B}_3.B_2$	$B_3.B_2$	$B_3.\overline{B}_2$
$\overline{B}_1.\overline{B}_0$	1	1	1	1
$\overline{B}_1.B_0$	0	0	X	0
$B_1.B_0$	0	0	X	0
$B_1.\overline{B}_0$	1	1	X	1

The equation of the k-map is

$$E_0 = \overline{B}_0$$

Prof. Ismail H. Popatia

Circuit Diagram : Hence the circuit diagram for Binary – to – XS3 code converter is



Result: Hence the following code-converters were designed and implemented

a) Binary – to – Gray code converter.
b) Gray – to – Binary code converter.
c) Binary – to – BCD code converter.
d) Binary – to – XS-3 code converter.

For video demonstration of the given practical click on the link below or scan the QR-code

https://youtu.be/Q-Y4fl4nwb8