DESIGN AND IMPLEMENTATION OF CODE CONVERTOR

EXPT NO :

DATE :

AIM: To design and implement 4-bit

(i) Binary to gray code converter
(ii) Gray to binary code converter
(iii) BCD to excess-3 code converter
(iv) Excess-3 to BCD code converter

APPARATUS REQUIRED:

Sr. No.	COMPONENT	SPECIFICATION	QTY.
1.	X-OR GATE	IC 7486	1
2.	AND GATE	IC 7408	1
3.	OR GATE	IC 7432	1
4.	NOT GATE	IC 7404	1
5.	IC TRAINER KIT	-	1
6.	PATCH CORDS	-	35

THEORY:

The availability of a large variety of codes for the same discrete elements of information results in the use of different codes by different systems. A conversion circuit must be inserted between the two systems if each uses different codes for the same information. Thus, code converter is a circuit that makes the two systems compatible even though each uses different binary code.

The bit combination assigned to binary code to gray code. Since each code uses four bits to represent a decimal digit. There are four inputs and four outputs. Gray code is a non-weighted code.

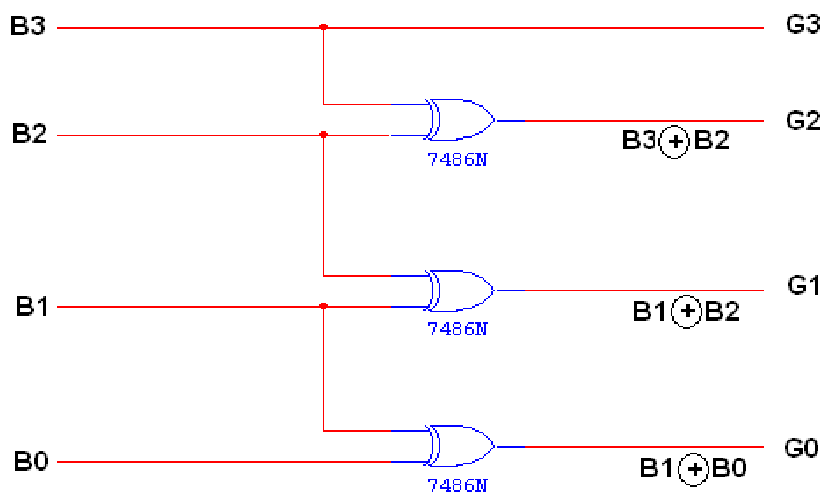
The input variable is designated as B3, B2, B1, B0, and the output variables are designated as C3, C2, C1, Co. from the truth table, the combinational circuit is designed. The Boolean functions are obtained from K-Map for each output variable.

A code converter is a circuit that makes the two systems compatible even though each uses a different binary code. To convert from binary code to Excess-3 code, the input lines must supply the bit combination of elements as specified by code and the output lines generate the corresponding bit combination of code. Each one of the four maps represents one of the four outputs of the circuit as a function of the four input variables.

A two-level logic diagram may be obtained directly from the Boolean expressions derived by the maps. These are various other possibilities for a logic diagram that implements this circuit. Now the OR gate whose output is C+D has been used to implement partially each of three outputs.

LOGIC DIAGRAM:

BINARY TO GRAY CODE CONVERTOR



K-Map for G_3:

B3B2 \ B1B0	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

$$G_3 = B_3$$

K-Map for G_2:

B3B2 \ B1B0	00	01	11	10
00				
01	1	1	1	1
11				
10	1	1	1	1

$G2 = B3 \oplus B2$

K-Map for G_1:

B3B2 \ B1B0	00	01	11	10
00			1	1
01	1	1		
11	1	1		
10			1	1

$G1 = B1 \oplus B2$

K-Map for G_0:

B3B2 \ B1B0	00	01	11	10
00		1		1
01		1		1
11		1		1
10		1		1

$$G0 = B1 \oplus B0$$

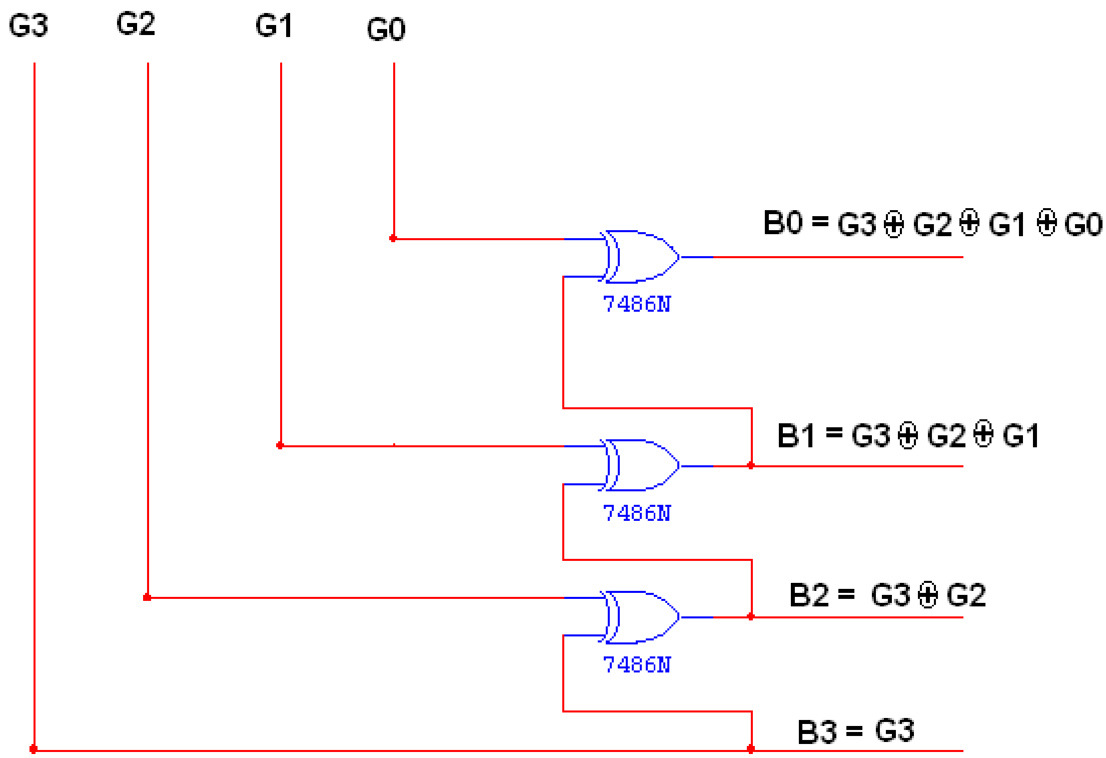
TRUTH TABLE:

| Binary input | Gray code output |

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

LOGIC DIAGRAM:

<u>GRAY CODE TO BINARY CONVERTOR</u>



K-Map for B_3:

G3G2 \ G1G0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$$B3 = G3$$

K-Map for B_2:

G3G2 \ G1G0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

$$B2 = G3 \oplus G2$$

K-Map for B_1:

G3G2 \ G1G0	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	0	0	1	1
10	1	1	0	0

$$B1 = G3 \oplus G2 \oplus G1$$

K-Map for B_0:

G3G2 \ G1G0	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

$$B0 = G3 \oplus G2 \oplus G1 \oplus G0$$

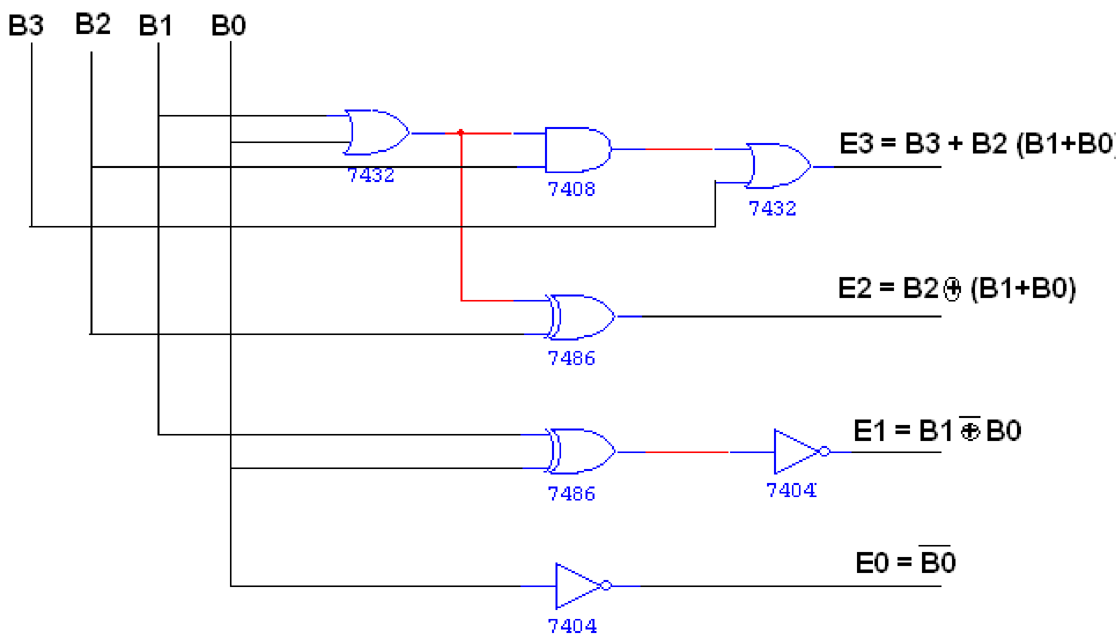
TRUTH TABLE:

| 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	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

LOGIC DIAGRAM:

<u>BCD TO EXCESS-3 CONVERTOR</u>



K-Map for E_3:

B3B2 \ B1B0	00	01	11	10
00				
01		1	1	1
11	X	X	X	X
10	1	1	X	X

E3 = B3 + B2(B0 + B1)

K-Map for E_2:

B3B2 \ B1B0	00	01	11	10
00		1	1	1
01	1			
11	x	x	x	x
10		1	x	x

$$E2 = B2 \oplus (B1 + B0)$$

K-Map for E_1:

B3B2 \ B1B0	00	01	11	10
00	1		1	
01	1		1	
11	x	x	x	x
10	1		x	x

$$E1 = B1 \overline{\oplus} B0$$

K-Map for E_0:

B3B2 \ B1B0	00	01	11	10
00	1			1
01	1			1
11	x	x	x	x
10	1		x	x

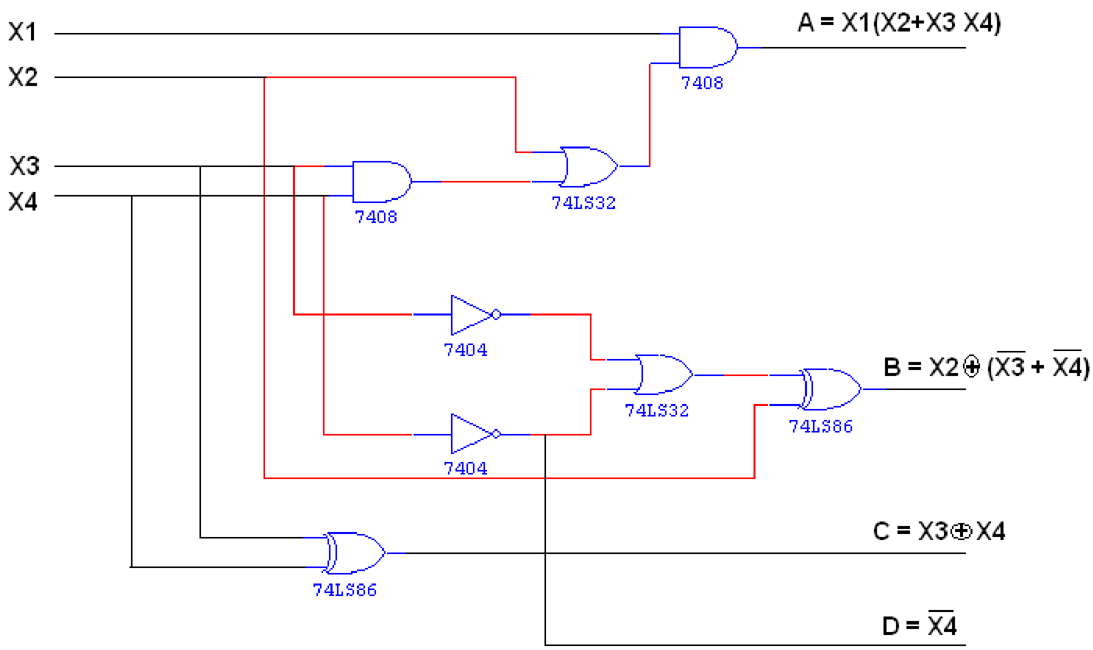
$$E0 = \overline{B0}$$

TRUTH TABLE:

BCD input				Excess – 3 output			
B3	**B2**	**B1**	**B0**	**G3**	**G2**	**G1**	**G0**
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	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

LOGIC DIAGRAM:

EXCESS-3 TO BCD CONVERTOR



K-Map for A:

X1 X2 \ X3 X4	00	01	11	10
00	X	X	0	X
01	0	0	0	0
11	1	X	X	X
10	0	0	1	0

A = X1 X2 + X3 X4 X1

K-Map for B:

X1 X2 \ X3 X4	00	01	11	10
00	X	X	0	X
01	0	0	1	0
11	0	X	X	X
10	1	1	0	1

$$B = X2 \oplus (\overline{X3} + \overline{X4})$$

K-Map for C:

X1 X2 \ X3 X4	00	01	11	10
00	X	X	0	X
01	0	1	X	1
11	0	X	X	X
10	X	1	0	1

$$C = X3 \oplus X4$$

K-Map for D:

X1 X2 \ X3 X4	00	01	11	10
00	X	X	0	X
01	1	0	0	1
11	1	X	X	X
10	1	0	0	1

$$D = \overline{X4}$$

TRUTH TABLE:

| Excess – 3 Input | BCD Output |

B3	B2	B1	B0	G3	G2	G1	G0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1

PROCEDURE:

(i) Connections were given as per circuit diagram.

(ii) Logical inputs were given as per truth table

(iii) Observe the logical output and verify with the truth tables.

RESULT: