Lab Experiment # 12

Design and Implementation of Code Convertor

Objective

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

Parts required:-

Sl.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

Equipment required:-

- Trainer/ proto board
- Wire cutter
- Patch Cords
- Voltmeter

Theory:

The availability of 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 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.

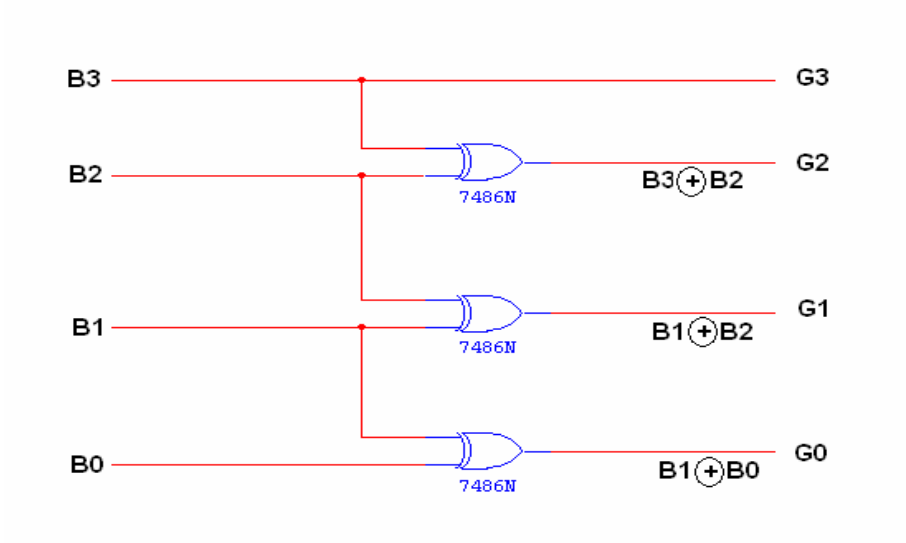
The input variable are designated as B3, B2, B1, B0 and the output variables are designated as C3, C2, C1, Co. from the truth table, combinational circuit is designed. The Boolean functions are obtained from K-Map for each output variable.

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

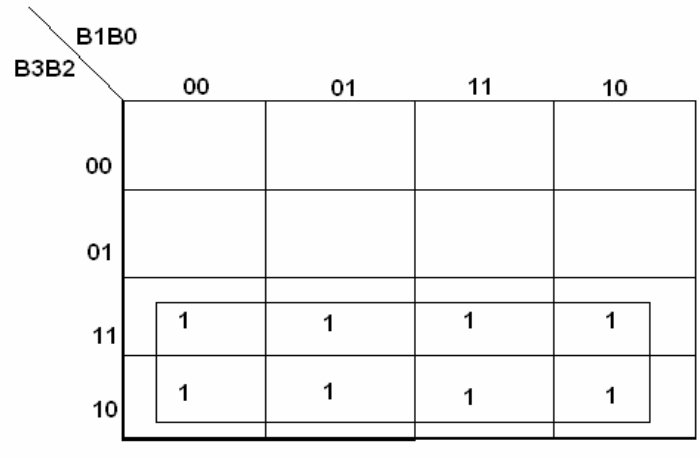
function of the four input variables.

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

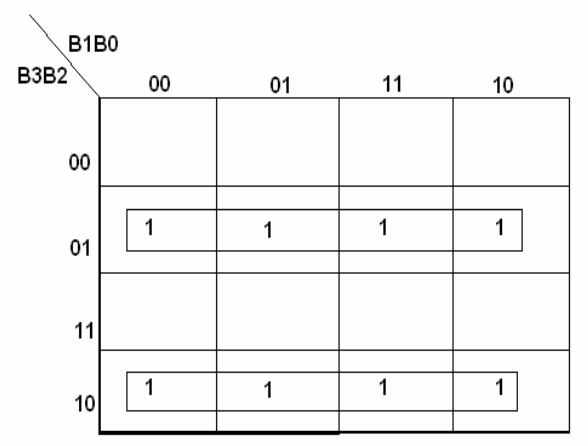


K-Map for G3:



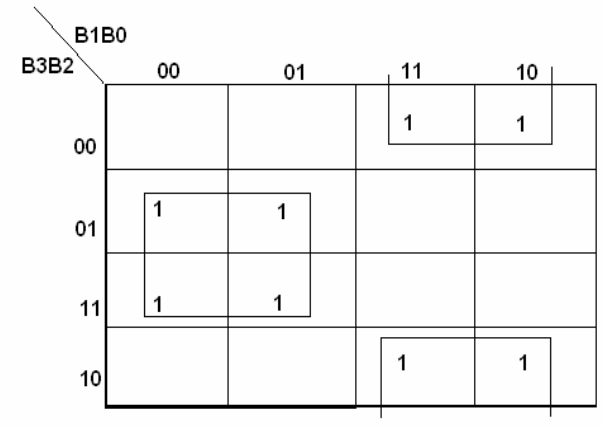
$G_3 = B_3$

K-Map for G2:



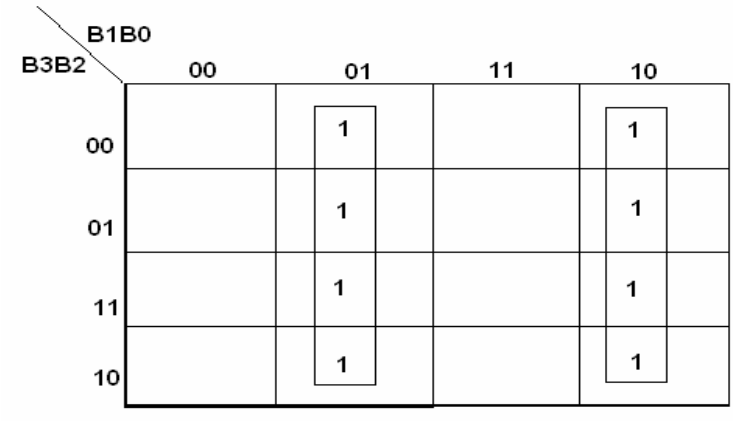
G2 = B3⊕B2

K-Map for G_1:



$G1 = B1 \oplus B2$

K-Map for G_0:



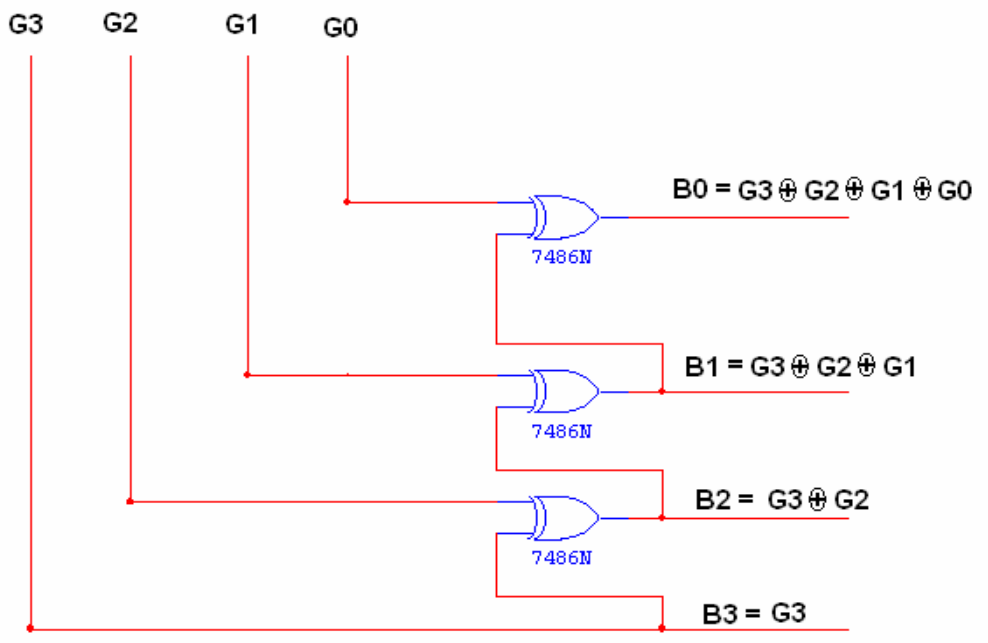
$G0 = B1 \oplus B0$

Truth Table:

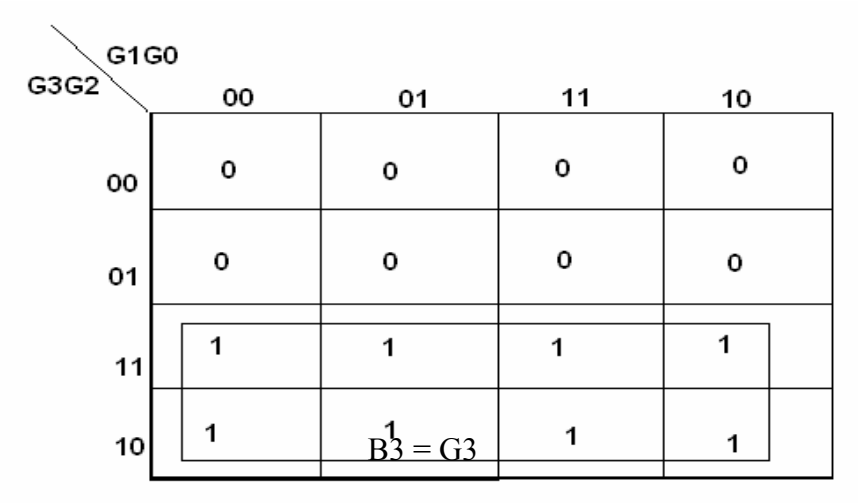
Binary input				Gray code output			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Logic Diagram:

Gray Code To Binary Convertor

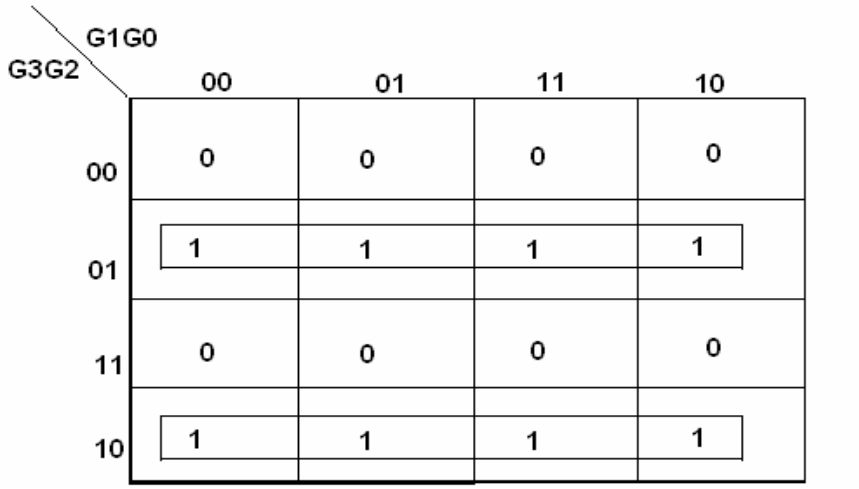


K-Map for B_3:



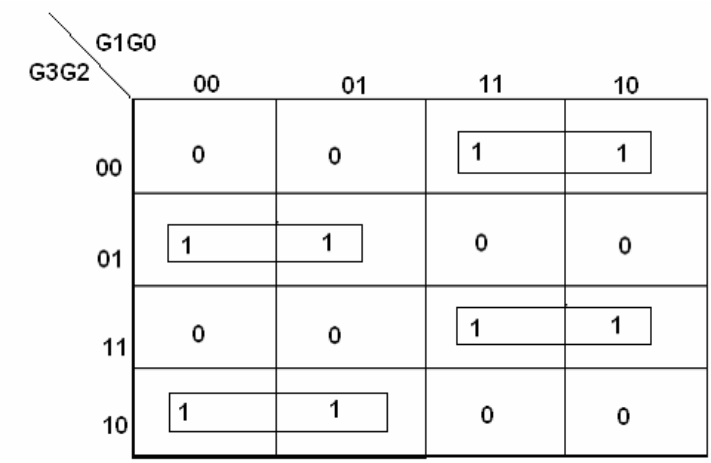
$B3 = G3$

K-Map for B_2:



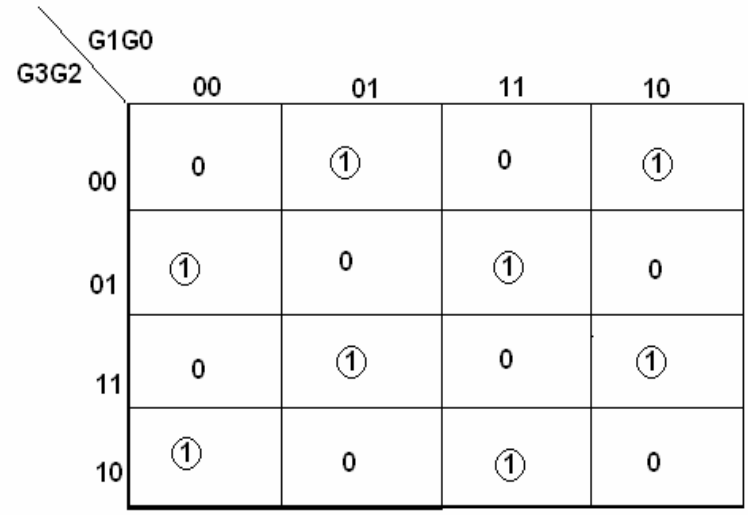
$B2 = G3 \oplus G2$

K-Map for B_1:



$B1 = G3 \oplus G2 \oplus G1$

K-Map for B_0:



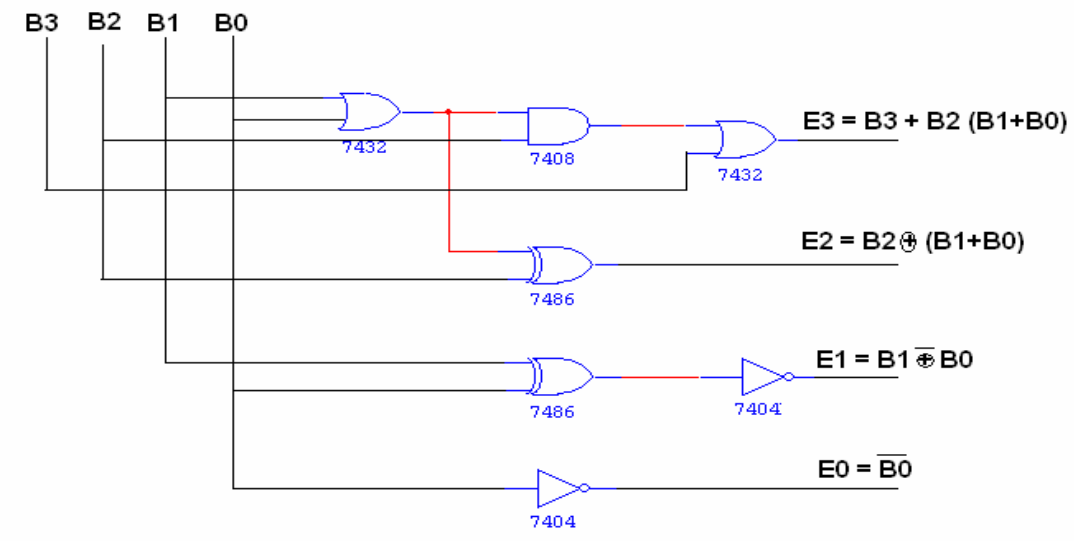
$B0 = G3 \oplus G2 \oplus G1 \oplus G0$

Truth Table:

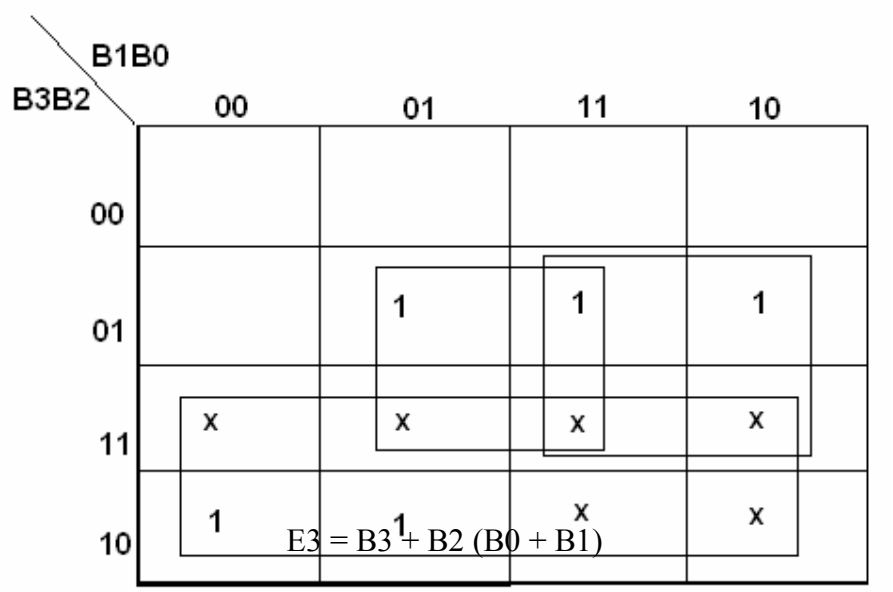
Gray Code				Binary Code			
G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

Logic Diagram:

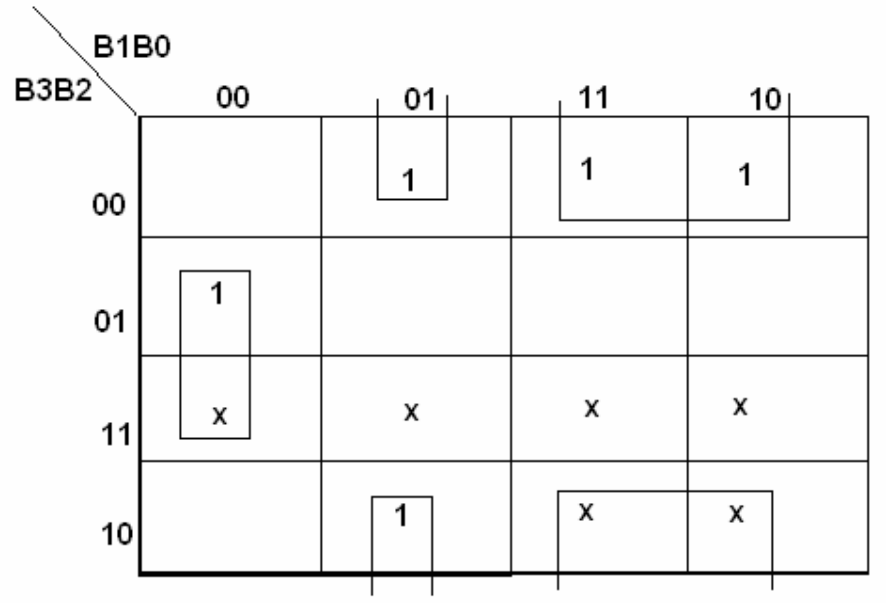
Bcd To Excess-3 Convertor



K-Map for E3:



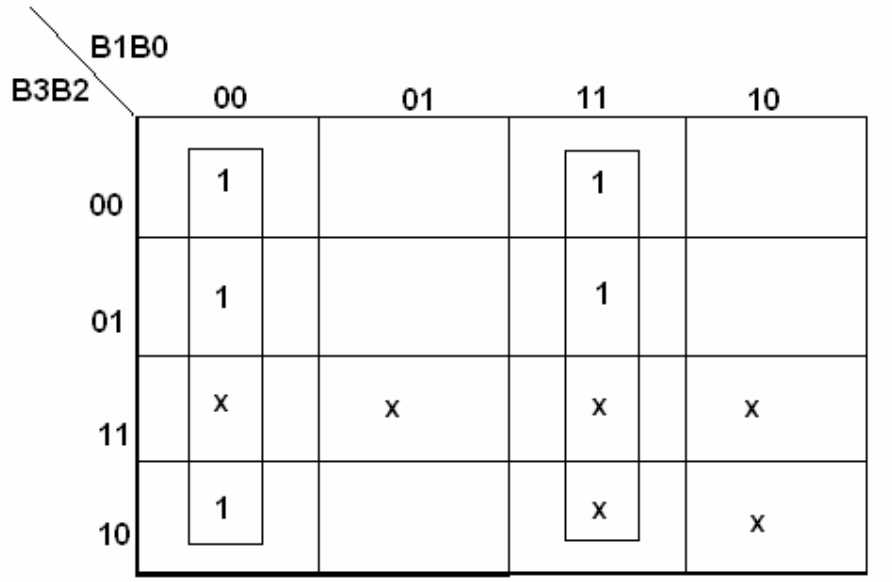
E3 = B3 + B2 (B0 + B1)



E2 = B2 ⊕ (B1 + B0)

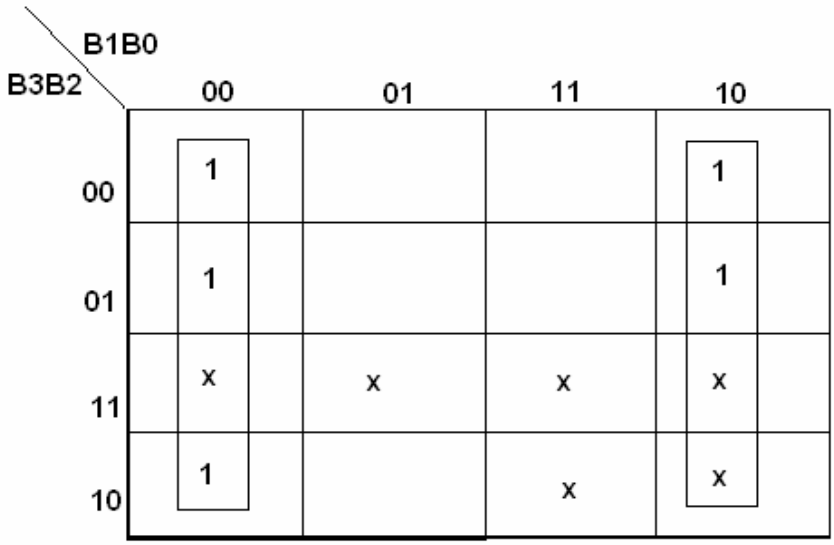
K-Map for E2:

K-Map for E_1:



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

K-Map for E_0:



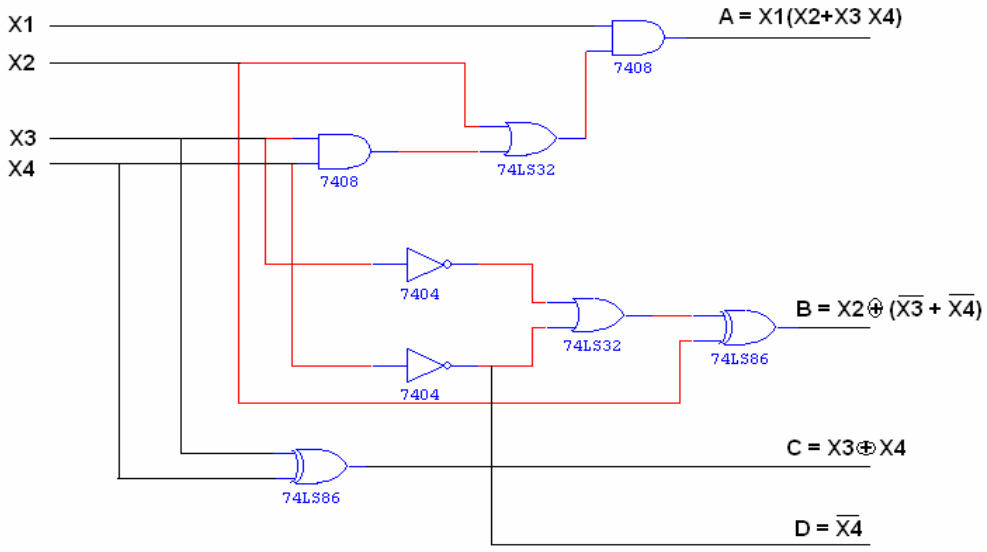
$E0 = \overline{B0}$

Truth Table:

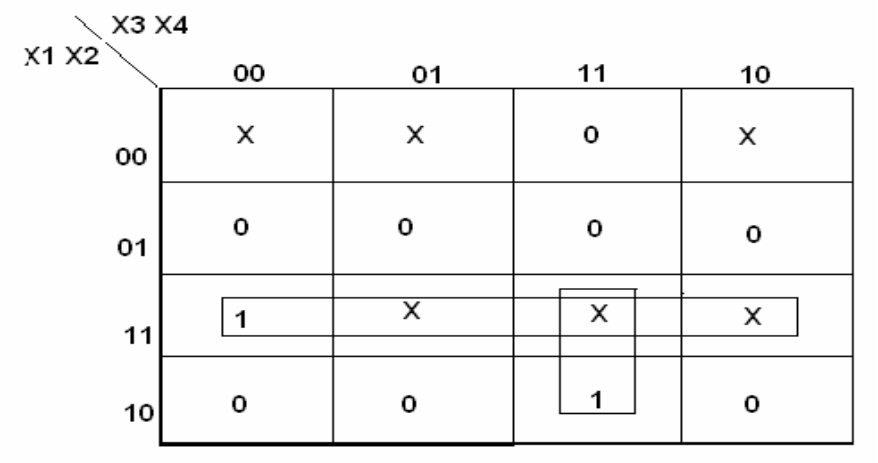
BCD input				Excess – 3 output			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1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

Logic Diagram:

Excess-3 To Bcd Convertor

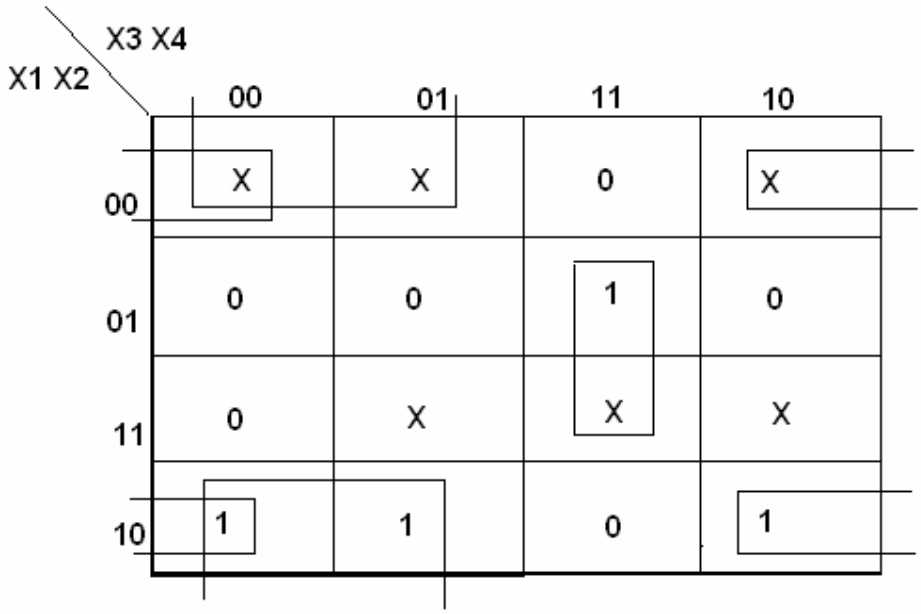


K-Map for A:



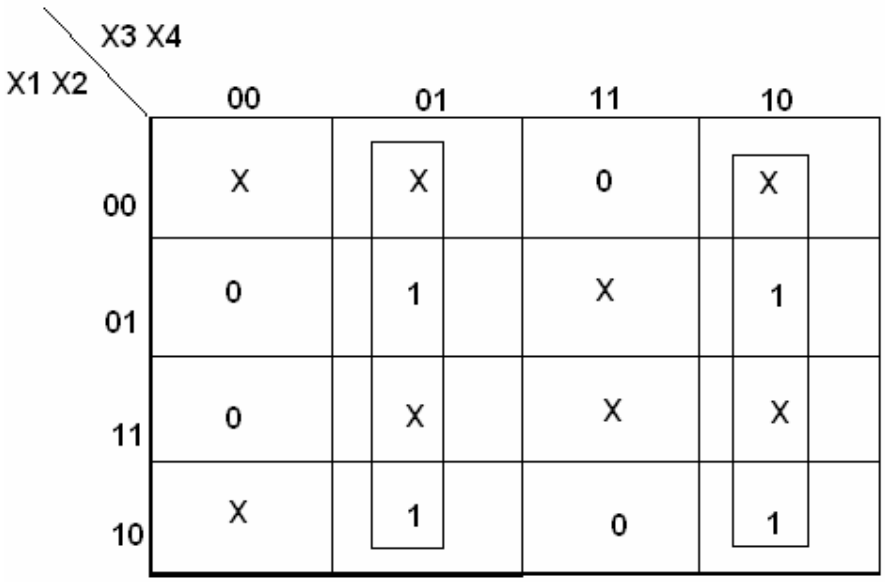
A = X1 X2 + X3 X4 X

K-Map for B:



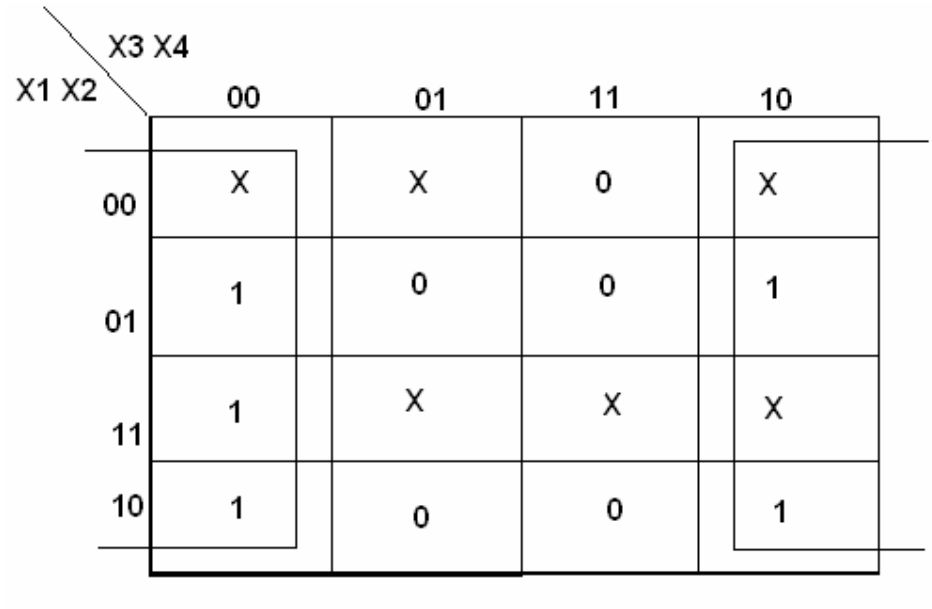
$B = X2 \oplus (\overline{X3} + \overline{X4})$

K-Map for C:



$C = X3 \oplus X4$

K-Map for D:



$D = \overline{X4}$

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) verify the gates
(ii) Connect the proper power supply
(iii) Connections were given as per circuit diagram.
(iv) Logical inputs were given as per truth table
(v) Observe the logical output and verify with the truth tables.