* Digital Electronics and Logic Design (DELD) - Practical Number - 23

Name:- Kaustubh Shrikant Kabra
Class:- Second Year engineering
Div:- A Roll Number:-
Batch:-
Department:- Computer Engineering Department.
College:- AISSMS's IOIT.

Title:-
Design and Implement code converters a) Binary to Gray
b) BCD to Excess-3.

Aim:-
To Design and understand the code converters:- a) Binary to Gray
b) BCD to Excess-3

Theory:-
a) Binary Number-
A number represented by the digits '0' and '1' is called binary number. Example:- i) 001
ii) 100
iii) 011

b) Gray Code-
In gray code, bit patterns for two consecutive numbers differ in only one bit position. It's application is it is use in which the normal sequence of binary number generated by the hardware may produce an error during transition from one number to another.
Example:- 011 (Binary) $\longrightarrow$ 010 (Gray)

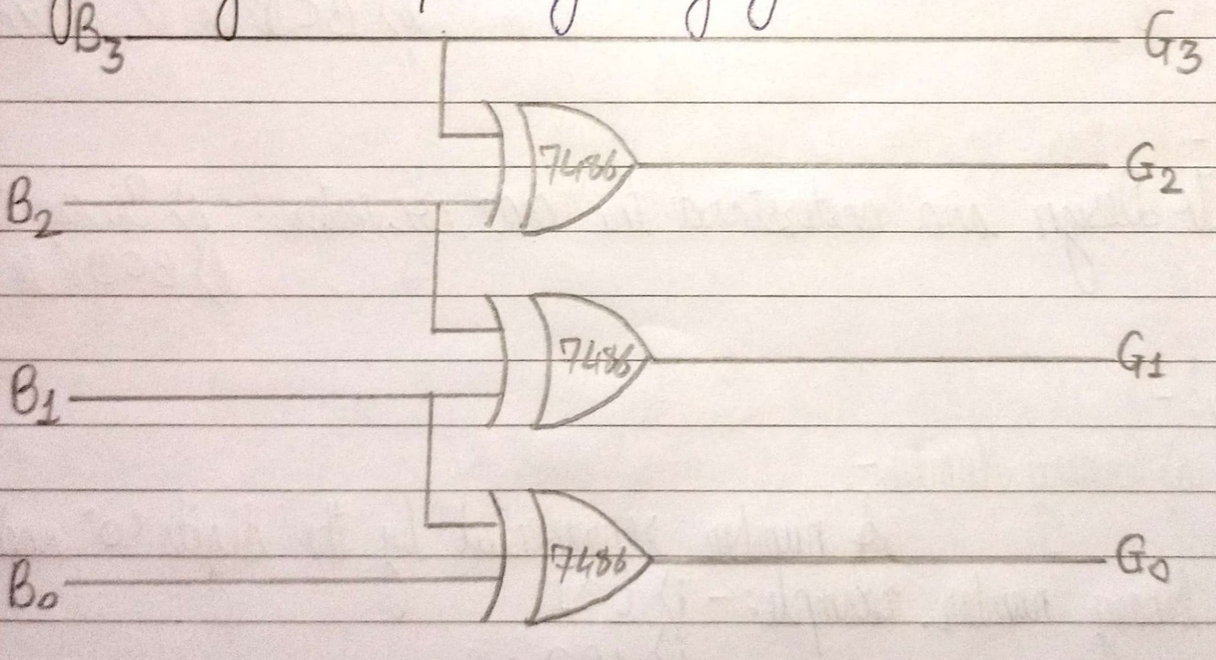
c) Binary Coded Decimal Number (BCD) -

BCD is a class of encoding of a decimal number where each digit is represented by a fixed number of bits, usually four or eight. Four bit BCD can be represent 16 numbers i.e. from 0(0000) to 15(1111).

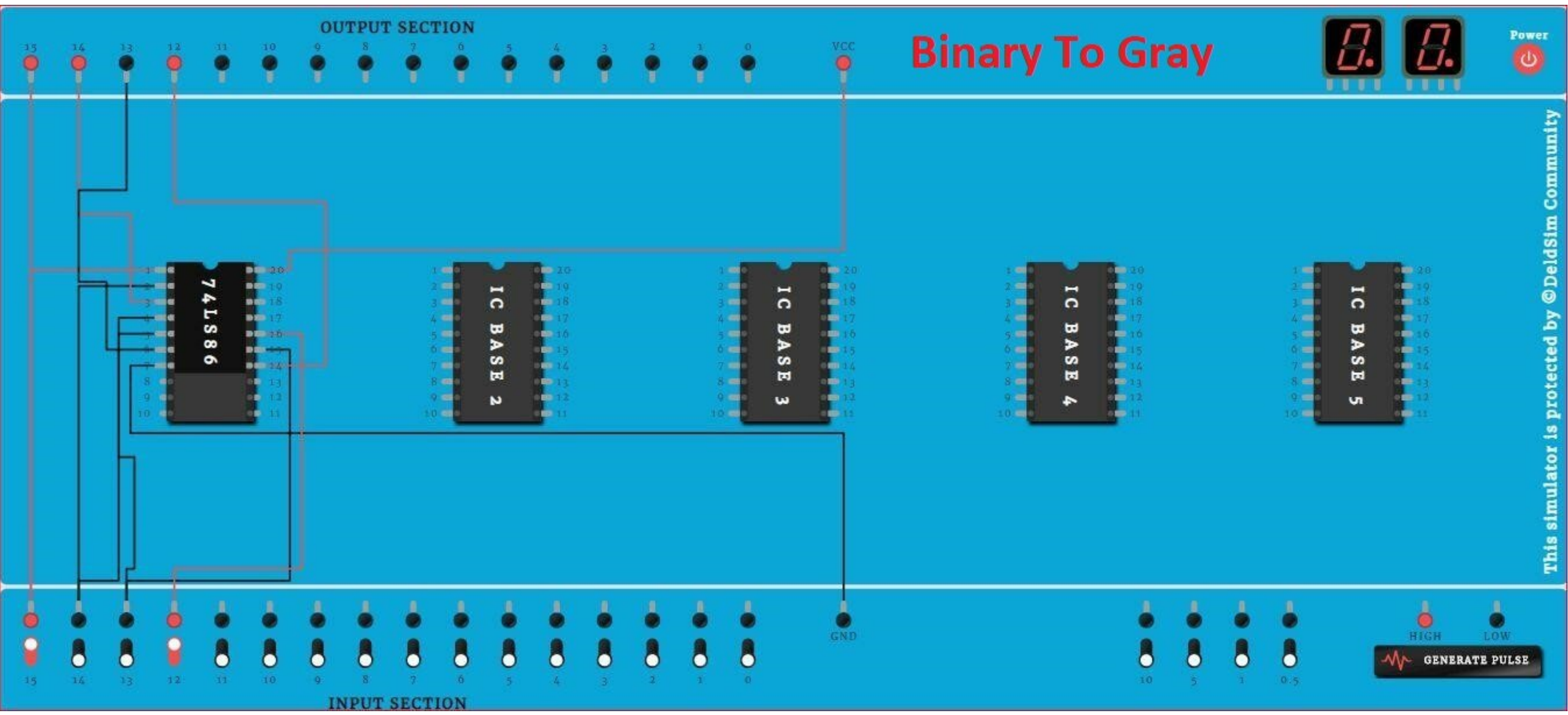
d) Excess - 3 ->

It is a non-weighted code used to express code to express decimal number. It is particularly significant for arithmetic operations as it overcomes shortcoming encounters.

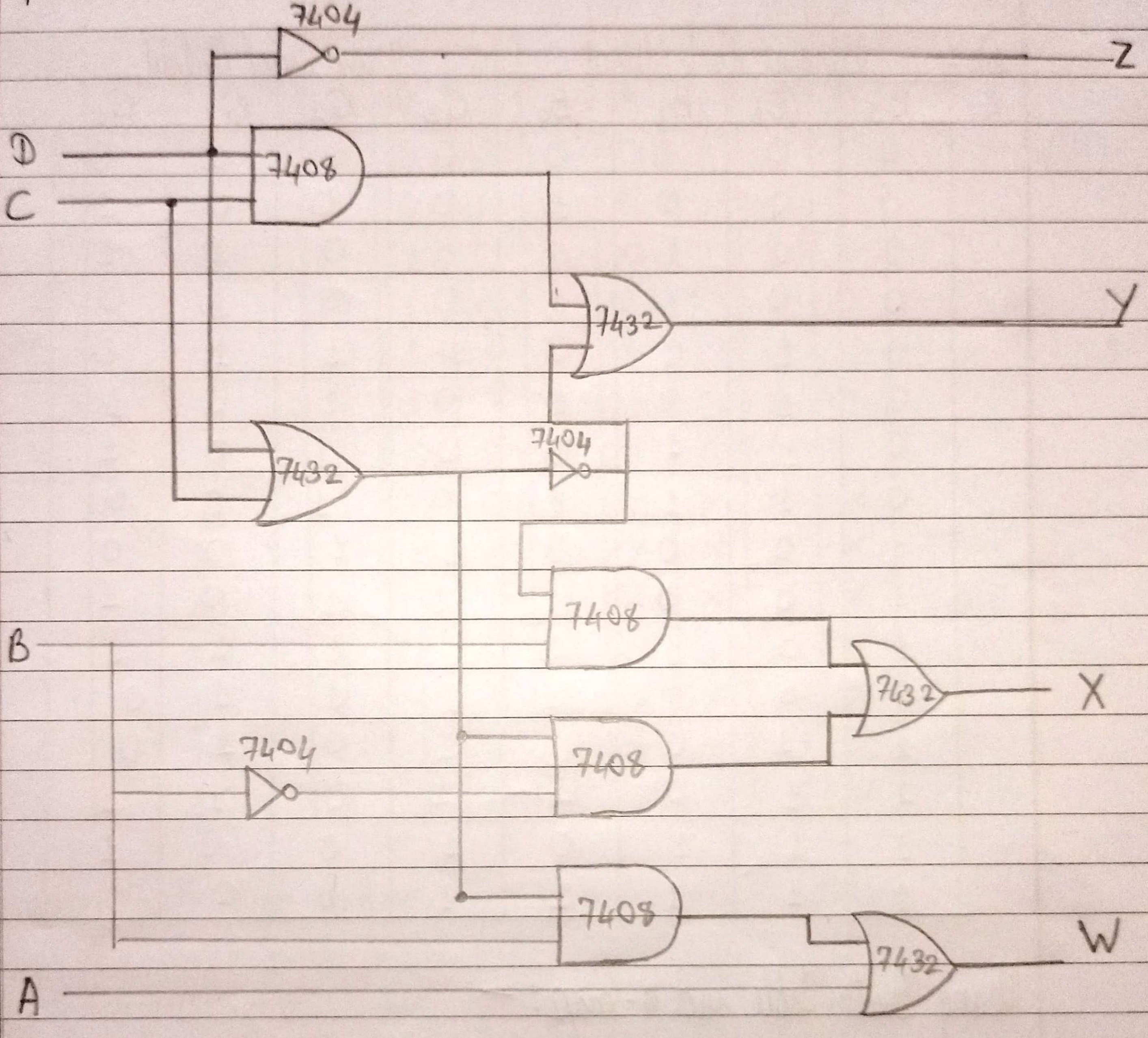
Logic Diagram: -> a) Binary to Gray: -



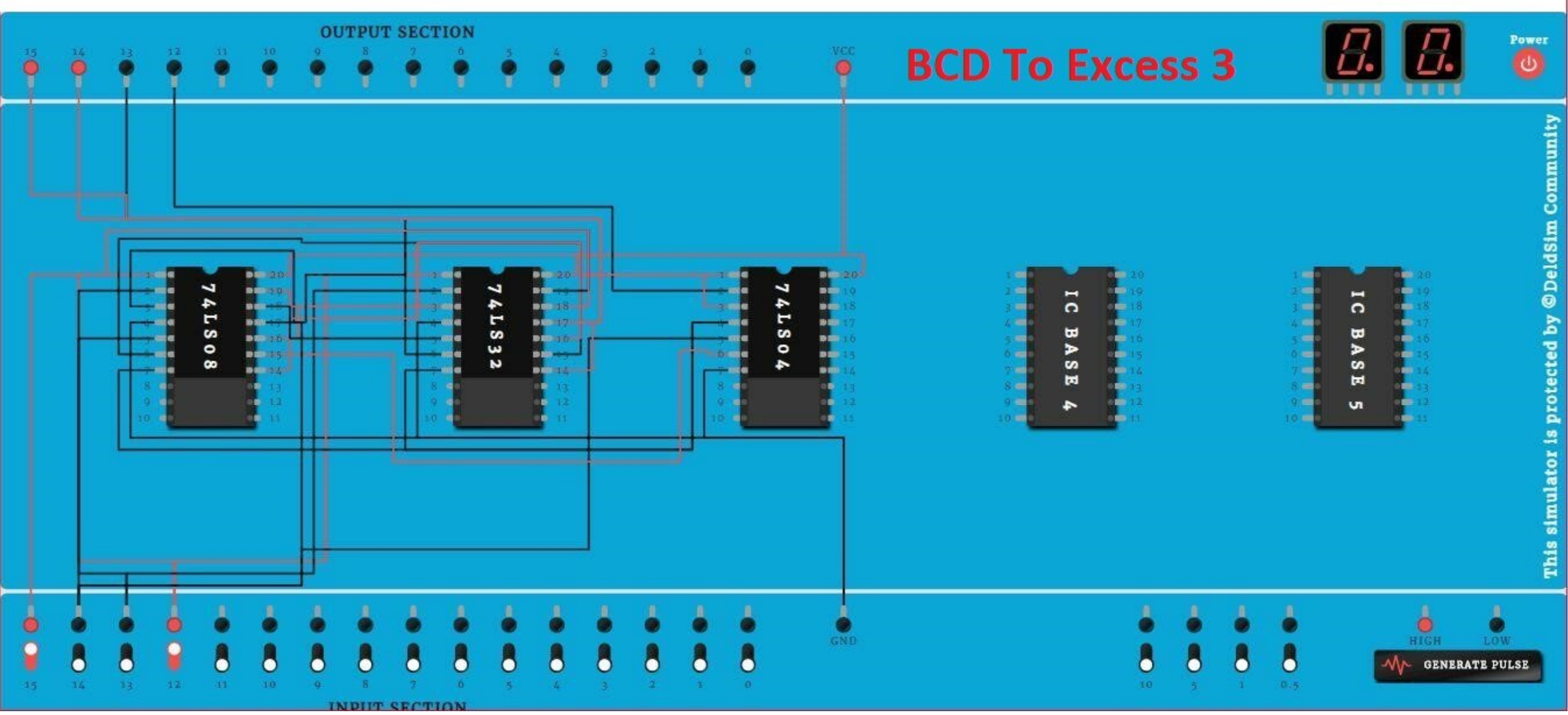
IC 7486 -> 2-Input XOR gate.


OUTPUT SECTION
Binary To Gray
Power
VCC
74LS86
IC BASE 2
IC BASE 3
IC BASE 4
IC BASE 5
This simulator is protected by ©DeldSim Community
GND
HIGH
LOW
GENERATE PULSE
INPUT SECTION

b) BCD to Excess 3:-



IC 7404 → NOT gate.
IC 7408 → 2-Input AND gate
IC 7432 → 2-Input OR gate.


OUTPUT SECTION
VCC
BCD To Excess 3
Power
74LS08
74LS32
74LS04
IC BASE 4
IC BASE 5
This simulator is protected by ©DeldSim Community
GND
HIGH
LOW
GENERATE PULSE
INPUT SECTION

Truth Table:-

a)→

Binary Code/Input				Gray Code/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

Using Truth Table and K-map:-

$G_3 = B_3$

$G_2 = B_2 \oplus B_3$ / B_2 XOR B_3

$G_1 = B_1 \oplus B_2$ / B_1 XOR B_3.

$G_0 = B_0 \oplus B_1$ / B_0 XOR B_1.

b) →

Decimal	BCD				Excess-3			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	X	X	X	X
10	1	0	1	0	X	X	X	X
11	1	0	1	1	X	X	X	X
12	1	1	0	0	X	X	X	X
13	1	1	0	1	X	X	X	X
14	1	1	1	0	X	X	X	X
15	1	1	1	1	X	X	X	X

Using Truth table and K-map: –

$W = A + BC + BD$

$X = B'C + B'D + BC'D'$

$Y = CD + C'D'$

$Z = D'$

Outcome:-

From the experiment we learnt to design and analyze binary to gray and BCD to Excess-3 code convertor.

Conclusion:-

Hence, we have designed and implemented a) binary to gray and b) BCD to Excess-3.