

## Experiment 2: Universal Gates and Applications

Time limit of experiment : 3 hours  
of Boolean algebra  
constant values of result of 3  
and logic function

Name:

Asifuzzaman, Id: 22201601

Badhon Nandi, Id: 22201601

Prithula Sikder, Id: 2220113291

Ahesan Habib, Id: 22201027

Azmani Sultana, Id: 22201949

(Signature) S.I.E.

Section 2

CSE 260

Group no: 01

IC 8208 IC 3208 IC 3208

### Objective:

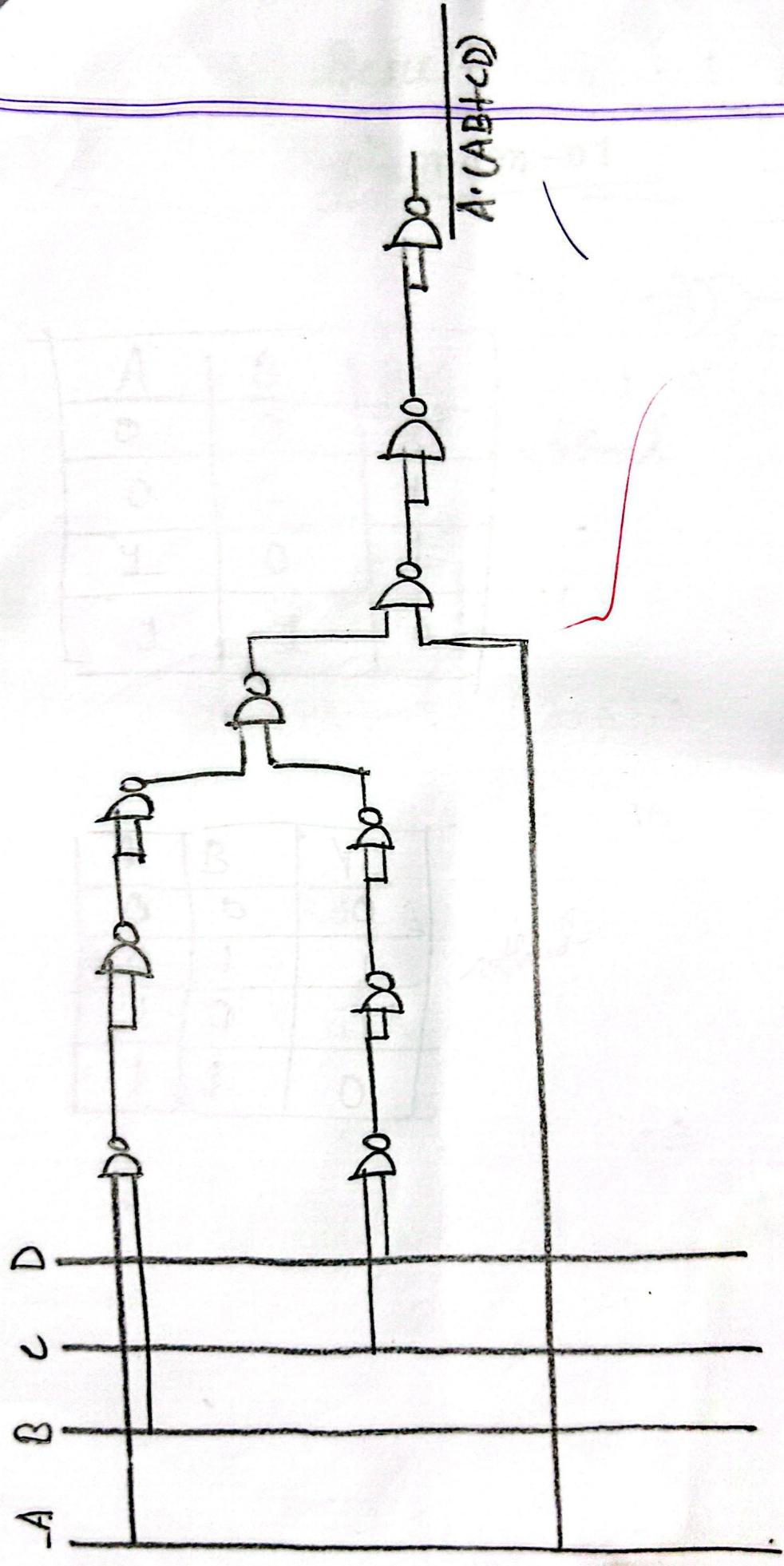
1. Learn to build practical circuit
2. To learn to solve a complex function using Boolean algebra
3. To gain deep knowledge of rules

### Equipment:

1. Breadboard
2. Jumper wire
3. IC

(Write names  
of IC)

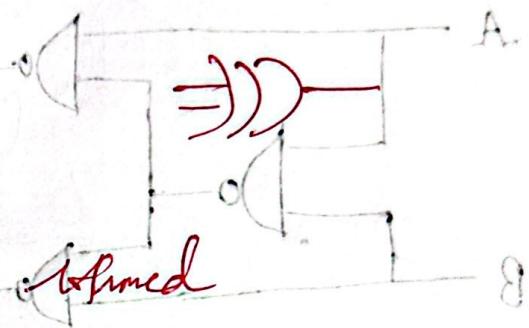
Discussion: In this experiment, we've build circuit using NAND and NOR gate. In both circuits, we get the result ~~of~~ similar to XOR gate. In 1 IC, we have 4 gates. So when ~~we~~ in second circuit, there are 5 NOR gate, so we need to use 2 IC.



Result

### Diagram - 01

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



Almed

$$(B+A) \cdot A = Y$$

$$S \cdot (B+A) + (S+A) \cdot A =$$

$$B \cdot A + B \cdot A =$$

A	B	Y
0	0	$(B \cdot A)$
0	1	1
1	0	$B+A$
1	1	0

$$S \cdot (B+A) =$$

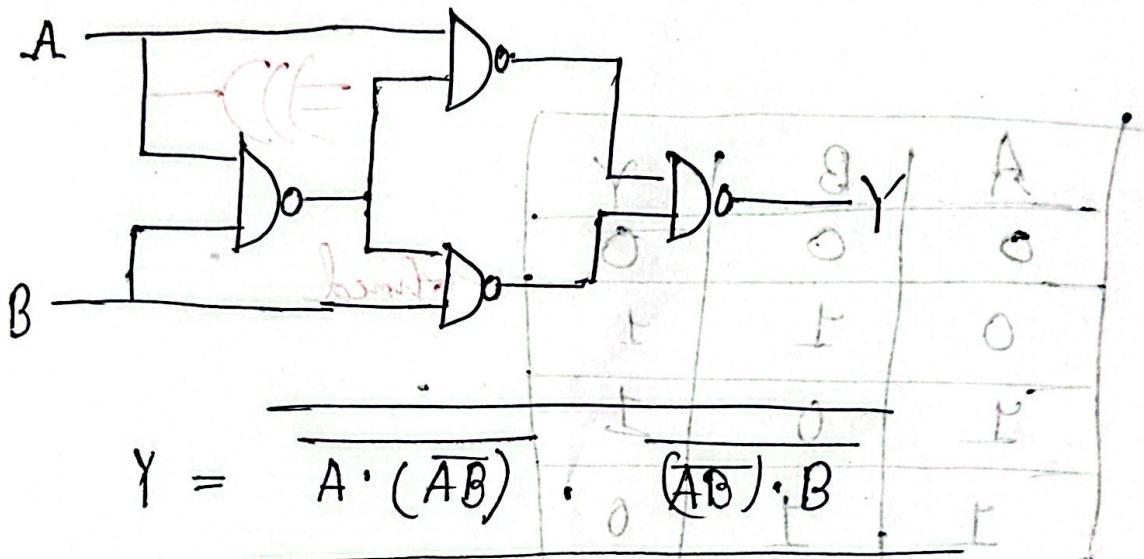
$$+ (B+A) =$$

$$B \cdot A + B \cdot A =$$

Almed

# Experimental set cep

Digital logic - 01

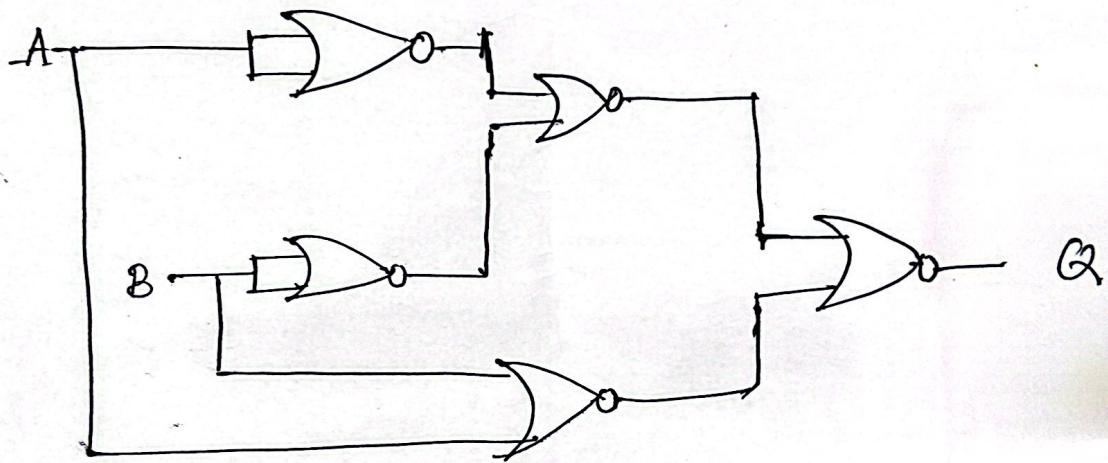


$$\begin{aligned}
 Y &= \overline{A \cdot (\overline{AB})} \cdot \overline{(\overline{AB}) \cdot B} \\
 &= \overline{A \cdot (\overline{A+B})} \cdot \overline{(\overline{A+B}) \cdot B} \\
 &= \overline{\overline{AB}} \cdot \overline{\overline{A} \cdot B} \\
 &= \overline{(\overline{A}+B)} \cdot \overline{(A+\overline{B})} \\
 &= \left( \overline{\overline{A}+B} \right) + \left( \overline{A+\overline{B}} \right) \\
 &= A \cdot \overline{B} + \overline{A} \cdot B
 \end{aligned}$$

		Y	B	A
0	0	0	0	0
1	0	1	0	1
0	1	0	1	0
1	1	1	1	1

? Name  
simplified  
logic

## Diagram - Q



$$\begin{aligned} Q &= \overline{\overline{A+B}} + \overline{(A+B)} \\ &= \overline{(A \cdot B)} + \overline{(\overline{A} \cdot \overline{B})} \\ &= (\overline{A}\overline{B}) \cdot (\overline{\overline{A} \cdot \overline{B}}) \\ &= (\overline{A}+\overline{B}) \cdot (A+B) = \overline{AB} + \overline{BA} \end{aligned}$$

? Www

10

Group 02

Name: Prithula Sikder Mayee

22201132

Badhon Nandi

222001601

Ahsan Habib

22201027

Azmani sultana

22201949

course: ESE -260

feetion: 02

Experiment name:

Design and implementation of 4-bit parallel binary adder.

Objectives

Through this experiment we have implemented ~~x-or, & and, or gate by which we have seen how the addition and subtraction is done for multivalued bits~~ for bits 1101.

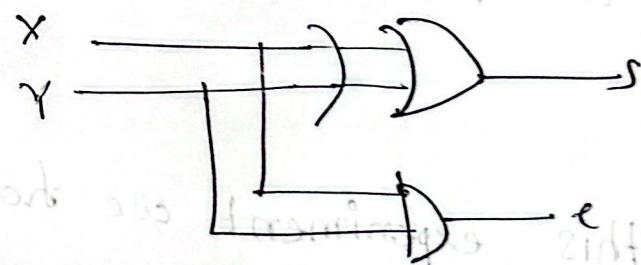
Required components and Equipment:

1. IC (7486-XOR, 7483(4 bit parallel, 7408- and 7432-OR)

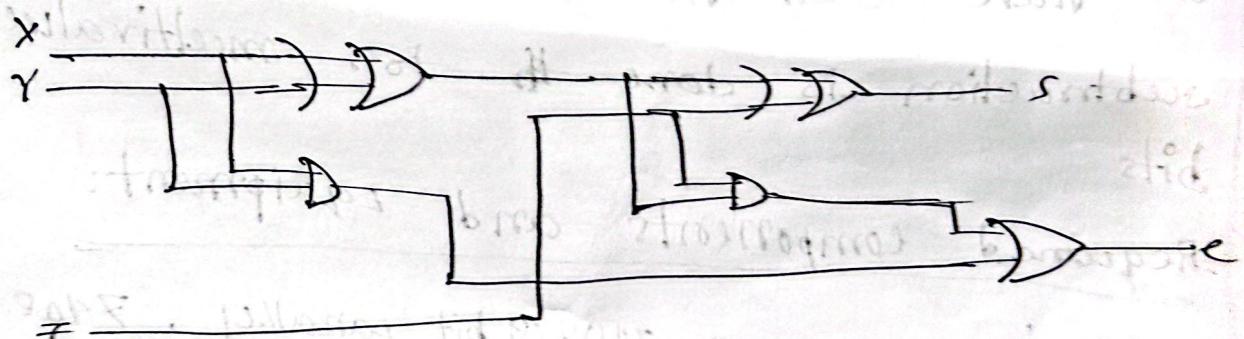
2. jumper wire, bread board,

## Experimental setup:

half adder



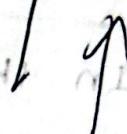
full adder



## Half adder

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1

1010000000000000



(n65)

Parallel

## Full adder

X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Ahd  
10101

Ahd  
10101

Ahd  
10101

## Discussion:

Lab 10

In this lab, we've experienced addition through half adder, full adder and parallel adder. We've learned the necessity of half adder which performs binary addition of two

single bit numbers, producing a sum and a carry output and full adder adds three 1-bit binary numbers (including a carry-in bit) and produces a sum and a carry-out bit, and also known about parallel adder which performs binary addition of multiple bit numbers by adding all corresponding bits simultaneously. We faced problem when we did full adder and parallel adder because of the connection of breadboard and wires.

Group-1

Name: Prithula Sikder Trajya

22201132

Badhon Nandi, 22201601

Ahsan Habib; 22201022

Azmani Scettana; 22201949

Section: 02

Course: CSE-260

2

Experiment name: Design a circuit that outputs 2's complement of a 3 bit number using encoder & decoder.

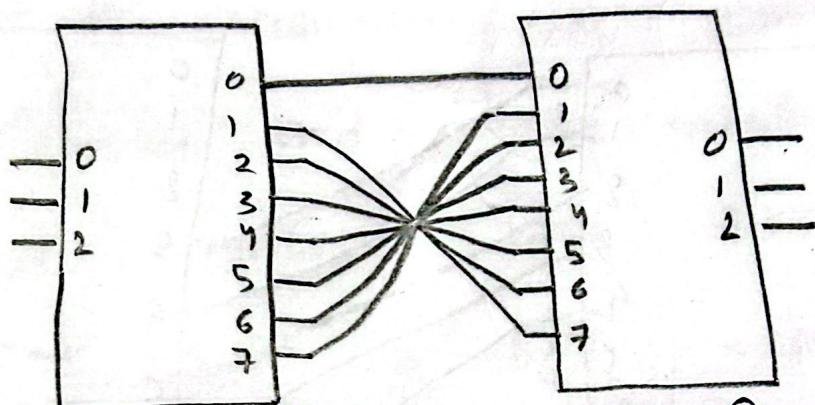
Objective: To understand and apply the concept of 2's complement for binary numbers and to design a digital circuit using encoder and decoder for computing 2's complement.

Required components and equipments

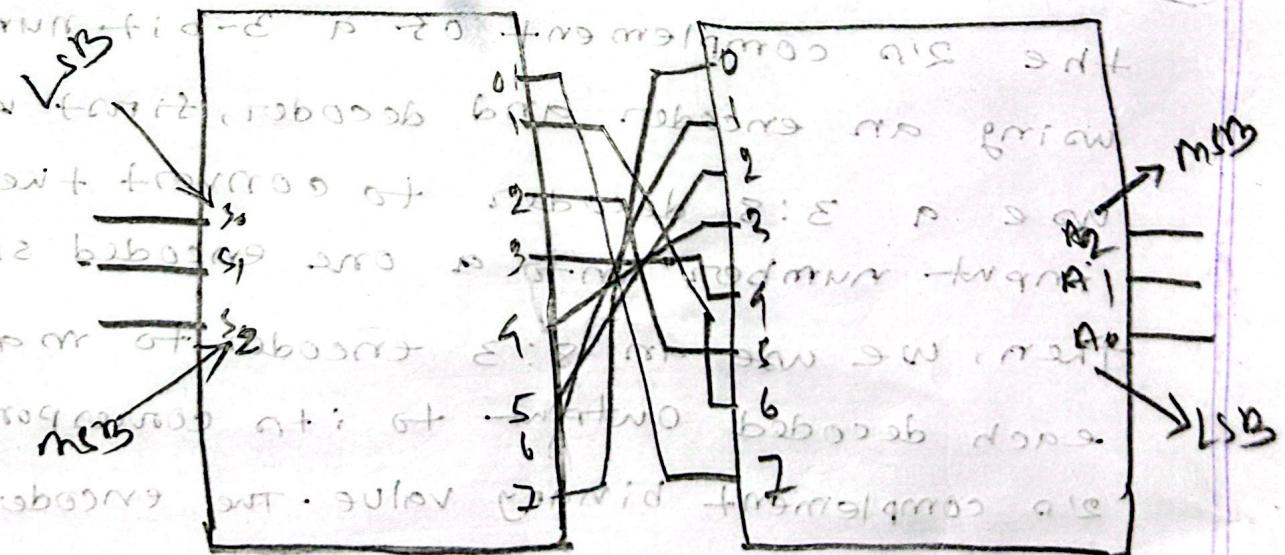
1. IC : 7415138 [decoder]; 7415148 [encoder]
2. Bread board
3. Jumper wire.

Experimental setup

## Experimental setup

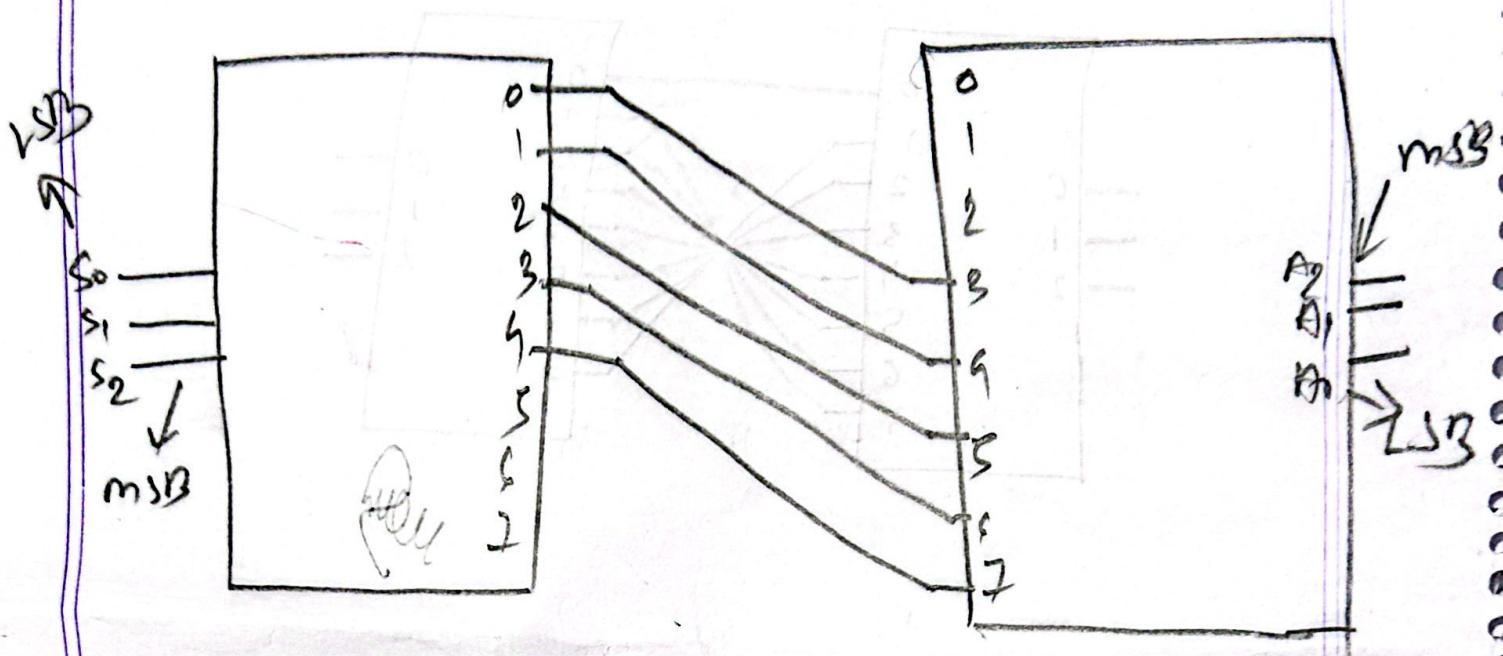


## Result:



I's complement for 3 bit input

## ~~Auto 2 Information~~ BCD to excess-3 converter



Discuss: To design a circuit that outputs the 2's complement of a 3-bit number using an encoder and decoder, first we use a 3:8 decoder to convert the input number into a one encoded signal. Then, we use an 8:3 encoder to map each decoded output to its corresponding 2's complement binary value. The encoder output gives the final 2's complement result; ~~the experiment demonstrates basic data encoding/decoding principles in digital logic design.~~

B. Here, when we are trying to connect the BCD to excess-03 code converter using  $3 \times 8$  decoder and  $8 \times 3$  encoder, if we connect convert upto BCD-4, because the Encoder is  $8 \times 3$ , means it has input from 0,7. As  $4+3=7$  is maxm for  $8 \times 3$  deode encoder. Rather if we used  $4:6 \times 4$  encoder instead, we would not have faced this issue.

Experiment name:

Implementation of 4-bit Magnitude  
Comparators.

Objective:

We have designed a 4-bit magnitude comparators circuit using not, X-nor, and OR, OR gate and proper truth table

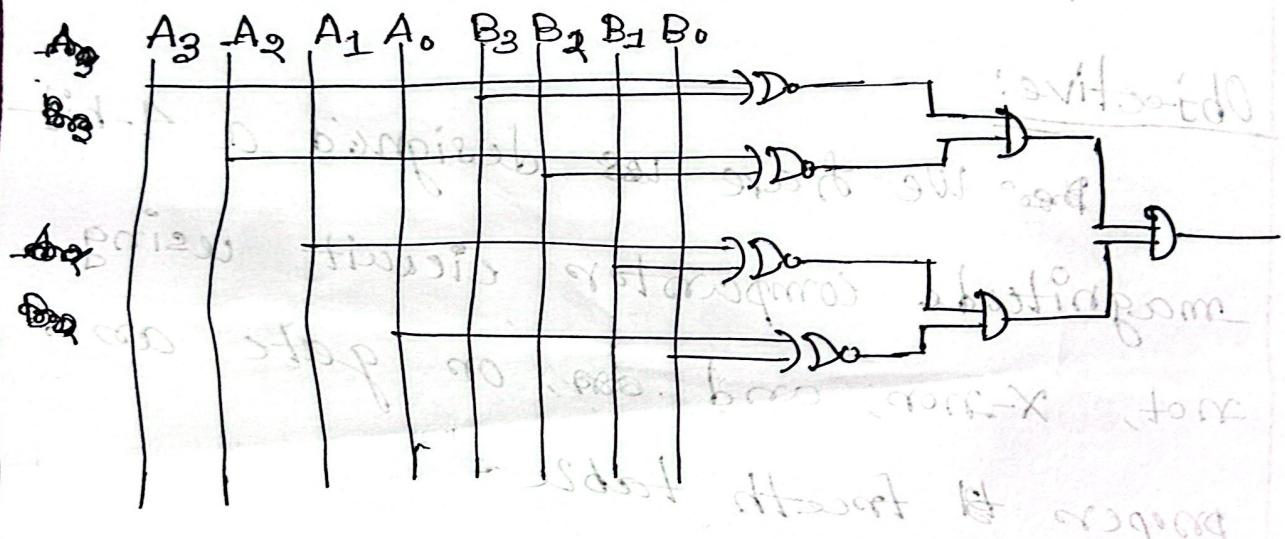
Required Components and Equipment's

1. IC(X108, X132, X109, 4077)  
AND OR NOT XNOR

2. Bread-board.
3. Jumper-wire.

## Experimental setup:

a)  $A = B$

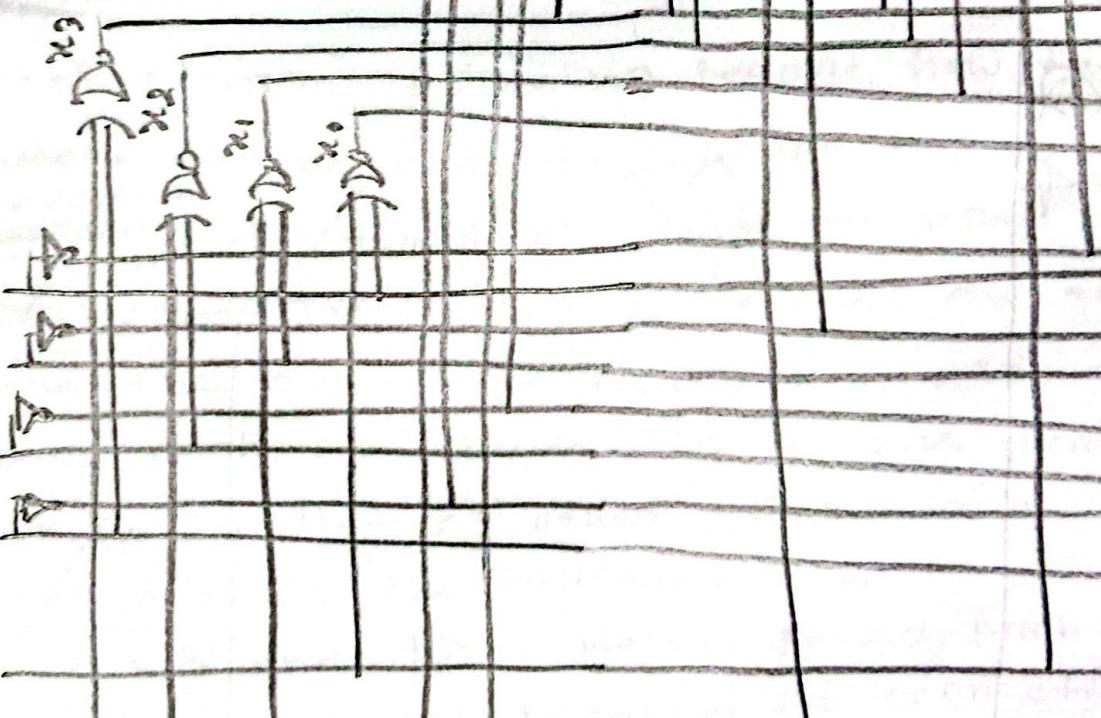


b)  $A \neq B$  gives 0 from the program because

(POP, POP, EST, 8018) L . L  
 ASK TON 90 40 V  
 - based - be set .  
 start - acquire S

A > B:

	$A_3$	$A_2$	$A_1$	$A_0$	$B_3$	$B_2$	$B_1$	$B_0$
	1	1	1	1	1	1	1	1



## Result

A	B	A=B	A>B	B>A
0000	0000	1	0	0
0010	0001	0	1	0
1000	0010	0	1	0
1000	1001	0	0	1

## Discussion

From the experiment, we've learnt how the ~~if~~ experiment can be used to compare two four bit numbers. The two 4-bit numbers are  $A = A_3 A_2 A_1 A_0$  and  $B = B_3 B_2 B_1 B_0$  where  $A_3$  and  $B_3$  are the most significant bits. It compares each of these bits in one number with bits in the other number and produces one of the following outputs as  $A=B$ ,  $A < B$  and  $A > B$ . We've faced problems doing the experiment, because the connections of our breadboard has some errors. Overall we've learnt to compare two four bit words.

$$A > B = A_3 B_3 + x_3 A_2 B_2 + x_3 x_2 A_1 B_1$$

~~A < B~~

$$+ x_3 x_2 x_1 A_0 B_0$$

$$A < B = A_3 B_3 + x_3 A_2 B_2 + x_3 x_2 A_1 B_1$$

$$+ x_3 x_2 x_1 A_0 B_0$$

Experiment name: Applications of Kmap method

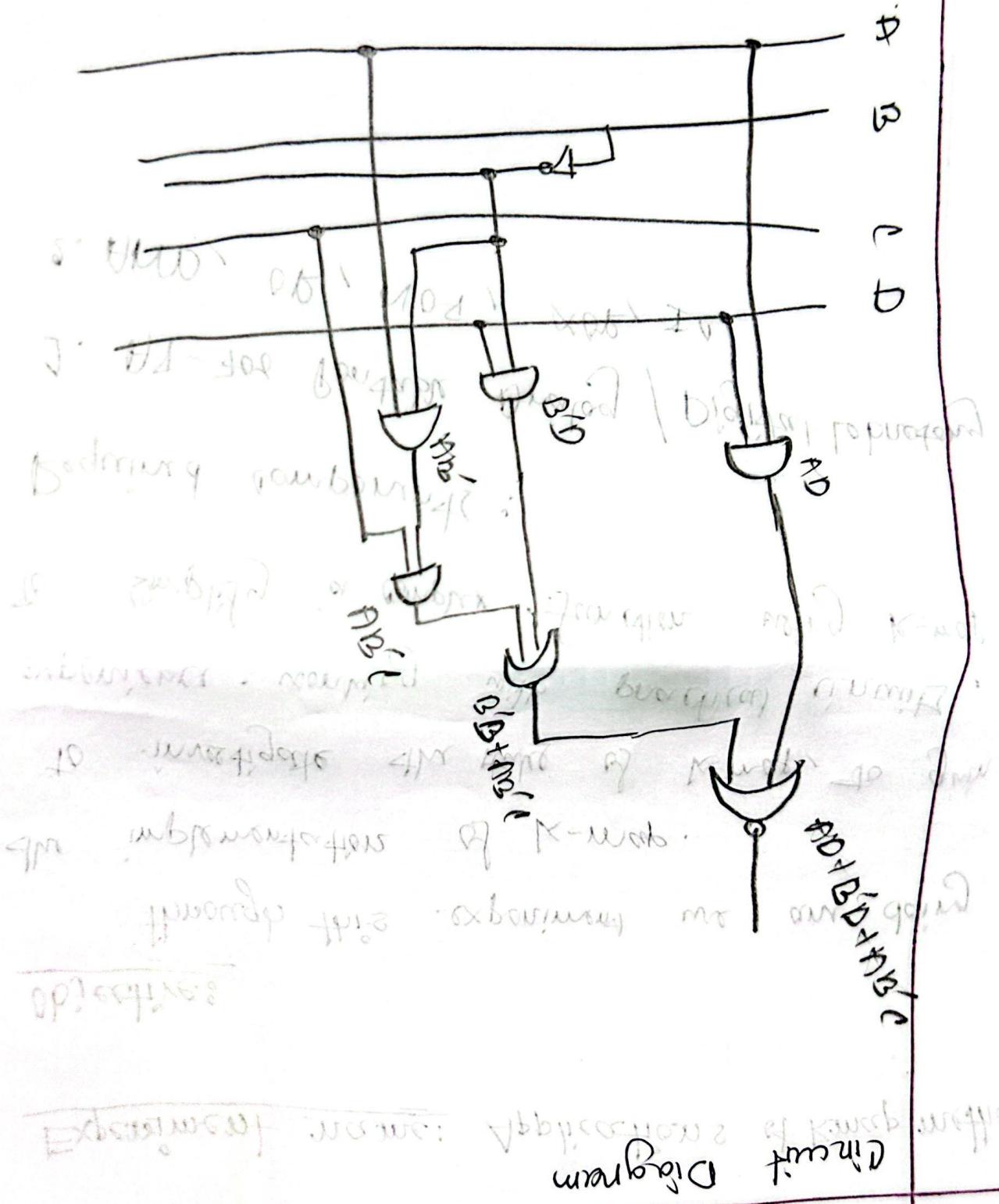
Objectives:

through this experiment we are doing the implementation of K-map.

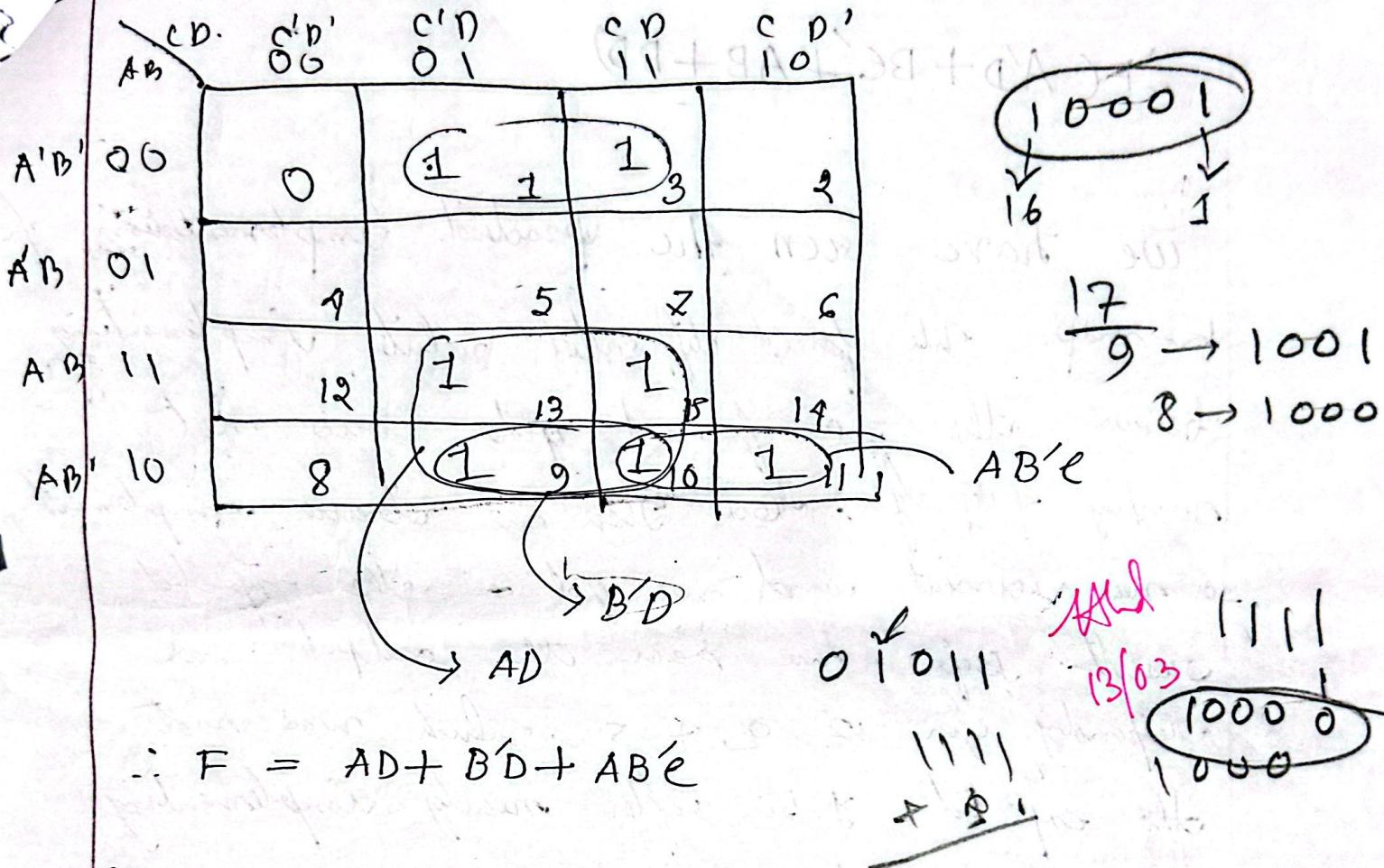
To investigate the rules of Kmap, To gain experience working with practical circuits,  
To Simplify a complex function using Kmap

Required Components:

1. AT-708 Portable Analog / Digital Laboratory
2. AND, OR, NOT, XOR, IC



Discussion:



$F(A, B, CD) = \sum(1, 4, 15) + d(3, 5, 7, 12, 13, 14)$

	CD	C'D'	C'D	CD'
A'B	00	00	11	10
A'B'00	0	1	X	2
A'B'01	1	X	X	6
A'B'11	X	X	1	X
A'B'10	8	9	11	10

$\rightarrow A'D$   
 $\rightarrow BC'$   
 $\rightarrow AB$   
 $\rightarrow BD$

$$\therefore F(A'D + BD' + AB + BD)$$

### Discussion:

We have seen the practical implementation of K-map. We faced difficulties while implementing because the IC of not gate was not working properly. Then we have complement a not circuit and we got the expected result. Before we saw the output was responding for 2, 4, 5 which was not the expected. Later after nearly implementing circuit we get output for  $\Sigma C_1, 3, 9, 10, 11, 13, 15$ .

$D_3$	$D_2$	$D_1$	$D_0$	$C_1$	$C_3$	$C_2$	$C_0$
01	11	X	X	X	X	X	X
10	00	X	X	X	X	X	X
00	00	X	X	X	X	X	X
11	11	X	X	X	X	X	X
01	11	X	X	X	X	X	X
11	00	X	X	X	X	X	X
00	10	X	X	X	X	X	X

# Op: Parity Generator and checker

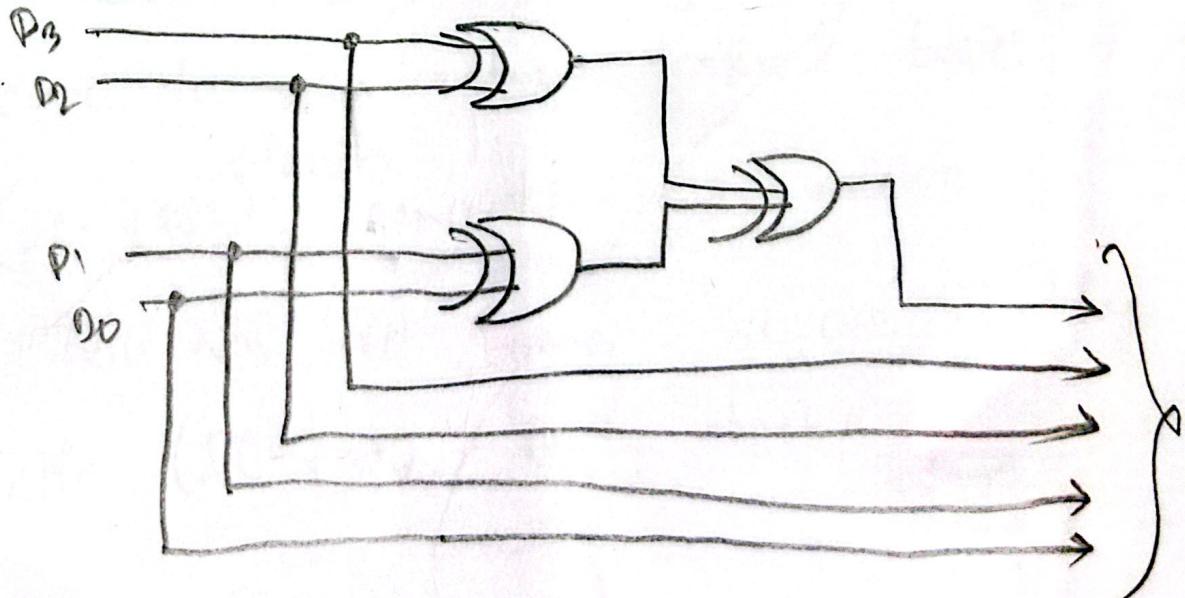
## Objective:

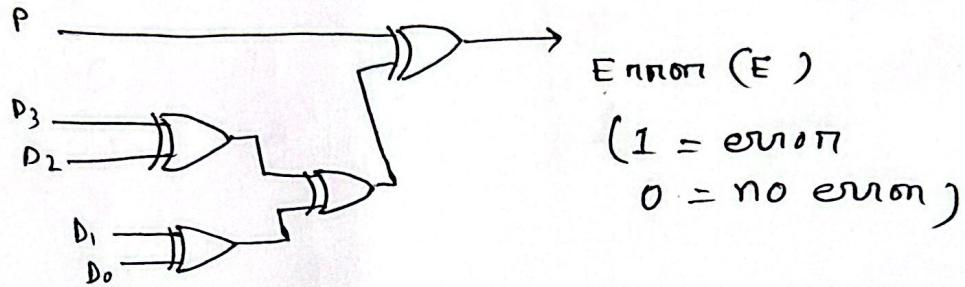
1. Understanding the mechanism of a parity generator & checker.
2. Designing a parity generator & checker with xor gates (IC-7486)

## Equipment's:

1. AT-708 Portable Analog / Digital Laboratory
2. 7486 - IC

## Even parity Generator:





Even Parity checker

### Discussion :

1. It is used to check the errors between receiver and sender.
2. During this experiment due to misconnection one of our (IC-7486) got shorted.
3. Because of this experiment we came to understand the basics of a Parity generator and checker (IC-7486).
4. If there was no Error we can identify it with (IC-7486) Parity checker.

$D_3$	$D_2$	$D_1$	$D_0$	Parity
0	1	1	1	1
1	0	0	1	0
0	0	0	0	0
0	1	0	0	1

Ahsan  
06/03

P	$D_3$	$D_2$	$D_1$	$D_0$	Parity
0	1	0	1	0	0
1	1	1	1	0	0
1	1	1	1	1	1
1	0	0	0	0	1

Ahsan  
06/03

### Discussion :

while sending bits from senders to receivers, sometimes the bits are altered. For detecting this errors we have used <sup>Even</sup> parity generators and checkers. ~~an even~~

In even parity, we get 0 for even numbers 1, we get 1 for odd numbers 1's.

In parity checkers if the output doesn't match with the parity bit, we get 1 and if it matched then we get 0.

Due to ~~wrong~~ wrong connection, the IC wasn't working properly. That's the difficulty we have faced during this experiment.