Experiment / Assignment / Tutorial No. 1

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Batch: B1	Roll No.: 1711072	Experiment / assignment / tutorial No.: 1

Title: Basic Gates & Universal Gates

Objective: To study the basic gates: AND, OR, NOT and universal gates: NAND, NOR, XOR, XNOR

Expected Outcome of Experiment:

CO1: Recall basic gates and binary, octal & hexadecimal calculations and conversions.

Books/ Journals/ Websites referred:

- R. P. Jain, "Modern Digital Electronics", Tata McGraw Hill
- http://www.ee.surrey.ac.uk/Projects/Labview/gatesfunc/
- http://www.electronics-tutorials.ws/boolean/bool_6.html

Pre Lab/ Prior Concepts:

Gate is a logic circuit with one or more inputs but only one output. Gates are digital (two state) circuit because the input & output are either low or high. Gates provide high output for certain combinations of input & for other combinations the output is low. Total number of combinations for a gate is 2^n; where n is number of input.

Classification: The two types of gate are:

1. **Basic or Fundamental Gates:**

a. **NOT:** It basically inverts the input given to it. That is if input is true, then the output will be false and vice versa. It generally has a single input.

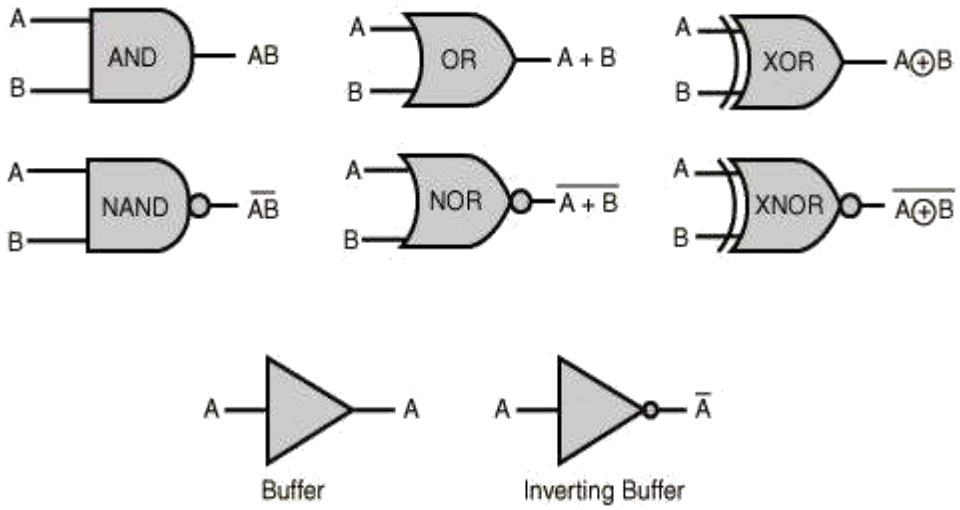
b. **OR:** It usually has two (or more) inputs, if either of the two inputs are *true* then the output is also true or else it is *false*.

c. **AND:** It usually has two (or more) inputs, if both of the two inputs are *true* then the output is also *true* or else it is *false*.

1. **Derived Gates:**

a. **NAND:** It is simply an *AND* gate with an inverted output. With its help any other logic gate can be reproduced.
b. **NOR:** It is simply an *OR* gate with an inverted output. With its help any other logic gate can be reproduced.
c. **EXOR:** If only one input out of the two is *true*, then the output is *true* else it is *false*.
d. **XNOR:** It basically inverts the output of an EXOR gate.

Symbols of gates



Type of IC	Specification
IC 7432	Quad-2 input OR gate TTL IC.
IC 7408	Quad-2 input AND gate TTL IC HEX.
IC 7404	Inverter TTL IC NOT gate.
IC 7400	Quad-2 input NAND gate TTL IC.
IC 7402	Quad-2 input NOR gate TTL IC.
IC 7486	Quad-2 input XOR gate TTL IC.

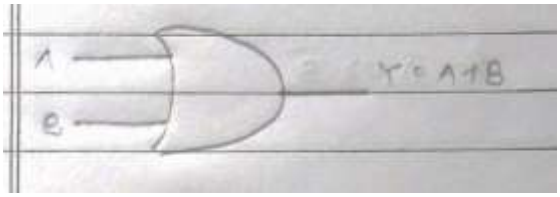
Implementation Details:

Basic Gates

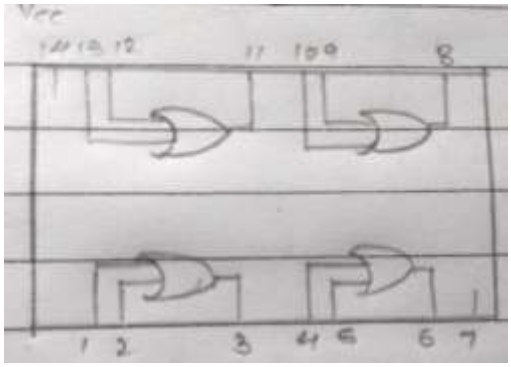
1. **OR gate:** The OR gate has two or more inputs but only 1 output. If any or all the inputs are high, the output is high. If all the inputs are low, the output is low.

$$Y = A+B$$

Symbol for OR Gate



Pin diagram for IC 7432



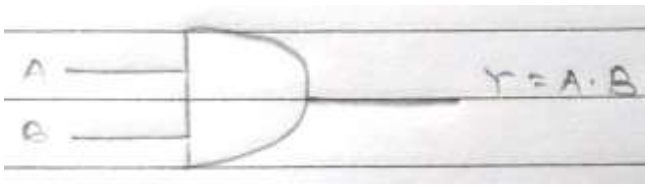
The truth table for OR operations are:

I/P		O/P
A	B	y = A+B
0	0	0
0	1	1
1	0	1
1	1	1

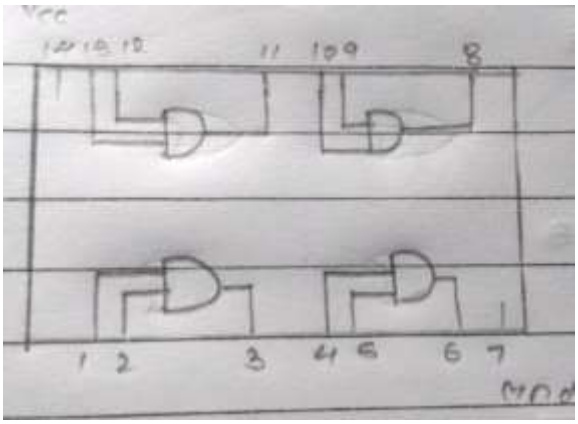
2. **AND gate:** The AND gate has two or more inputs but only one output. If any or all inputs are high then output is also high

$$Y= A.B$$

Symbol for AND gate



Pin Diagram For IC 7408



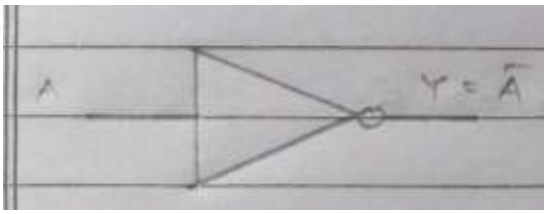
The truth table for AND operations are:

I/P		O/P
A	B	Y=A·B
0	0	0
0	1	0
1	0	0
1	1	1

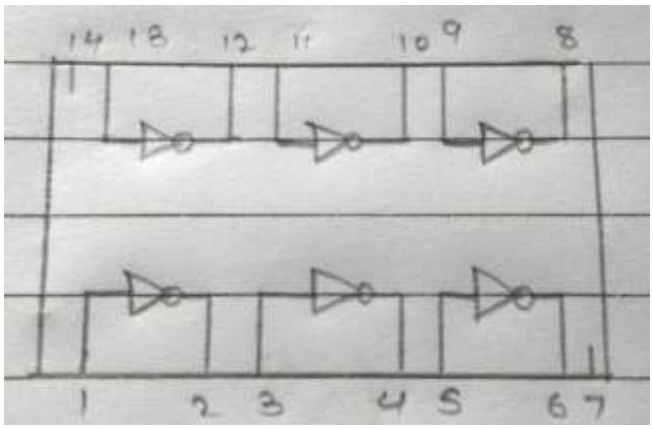
3. **NOT gate:** The Not gate is a gate with only one input and one output. The output is always in opposite state of an input. A NOT gate is also called as Inverter because it performs inversion.

$$Y= \bar{A}$$

Symbol for NOT gate



Pin Diagram For IC 7404



The truth table for NOT operations is:

I/P	O/P
A	$y = \bar{A}$
0	1
1	0

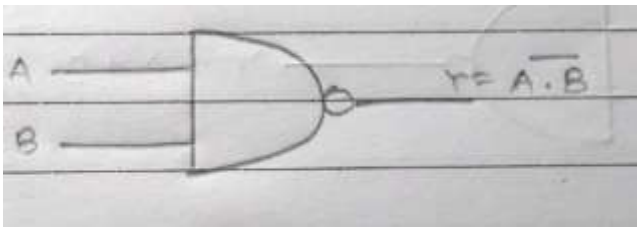
Derived Gates/Universal Gates

- NAND gate
- NOR gate
- EX-OR gate
- EX-NOR gate

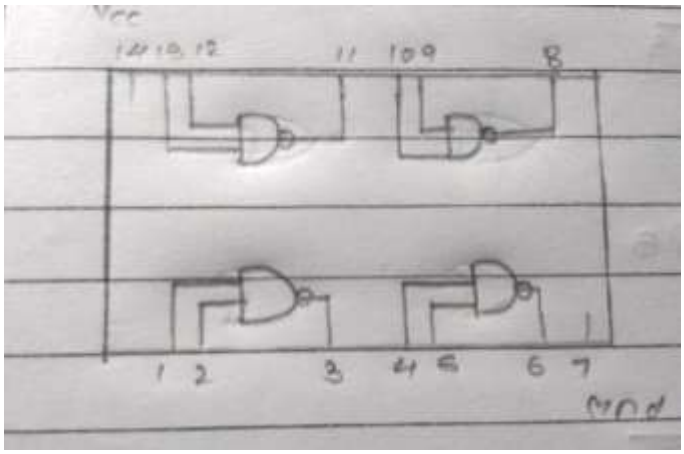
1. **NAND gate:** This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if any of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

$$Y = \overline{A.B}$$

Symbol



Pin Diagram for IC 7400



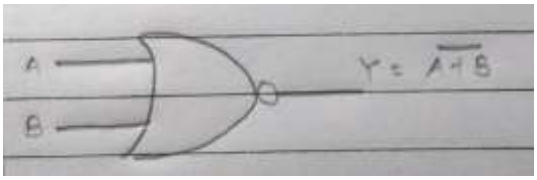
The truth table for NAND operations is:

I/P		O/P
A	B	$y=\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

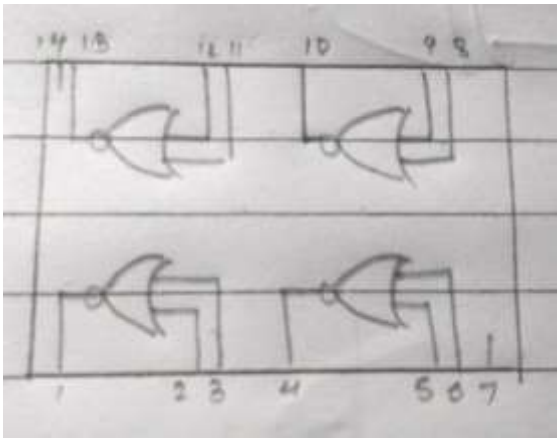
2. **NOR gate:** This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if any of the inputs are high. The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

$$Y= \overline{A + B}$$

Symbol for NOR gate



Pin Diagram For IC 7402



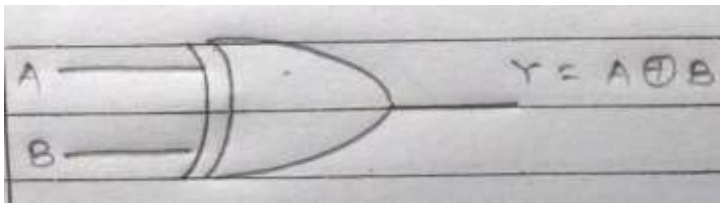
The truth table for NOR operations are:

I/P		O/P
A	B	$Y=\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

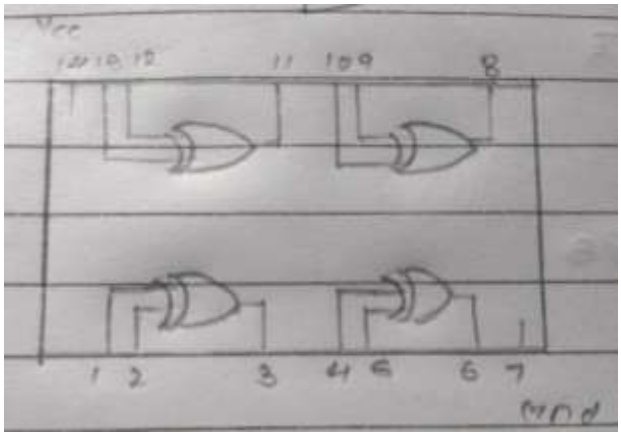
3. **EX-OR gate**: The 'Exclusive-OR' gate is a circuit which will give a high output if either, but not both, of its two inputs are high. An encircled plus sign (⊕) is used to show the EX-OR operation.

$$Y = \overline{A \oplus B}$$

Symbol for Ex-OR gate



Pin Diagram For IC 7486



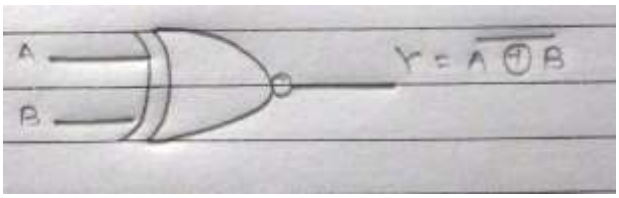
The truth table for XOR operations is:

I/P		O/P
A	B	y=A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

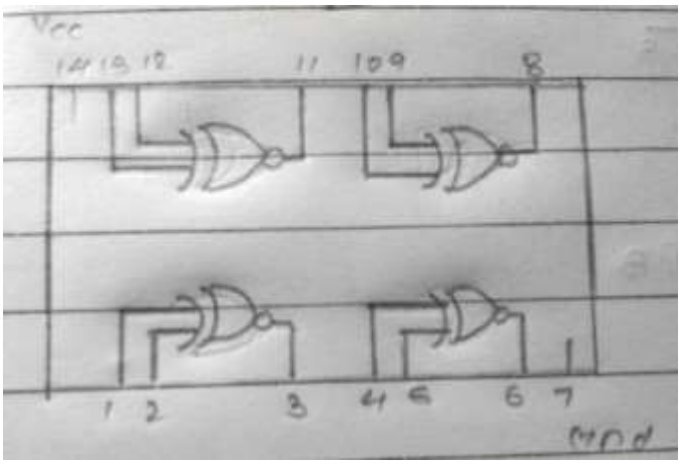
4. **EX-NOR gate**: The 'Exclusive-NOR' gate circuit does the opposite to the EOR gate. It will give a low output if either, but not both, of its two inputs are high. The symbol is an EXOR gate with a small circle on the output. The small circle represents inversion

$$Y=\overline{A \oplus B}$$

Symbol for Ex-NOR gate



Pin Configuration for IC

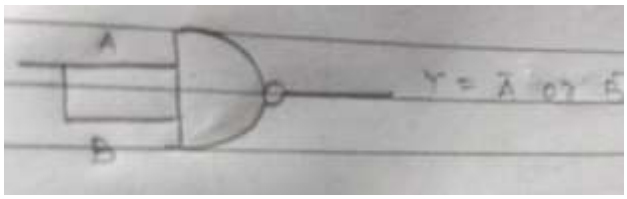


The truth table for XNOR operations is:

I/P		O/P
A	B	$y=\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

Implementation Using NAND Gate

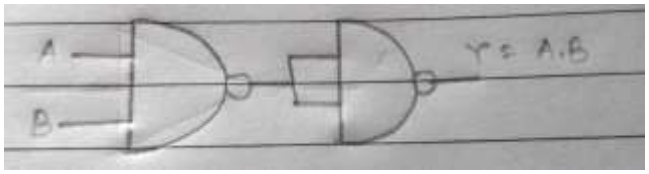
NOT GATE



STEPS

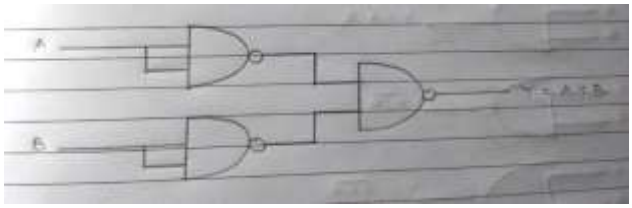
Here, we shot both the inputs of NAND, now using its truth table, we come to know that when both the inputs are same, the output is the inverse of it.

AND GATE



Here, we use a normal NAND Gate, and using the complement's complement property we apply a NOT made using NAND over it to get the respective AND Gate.

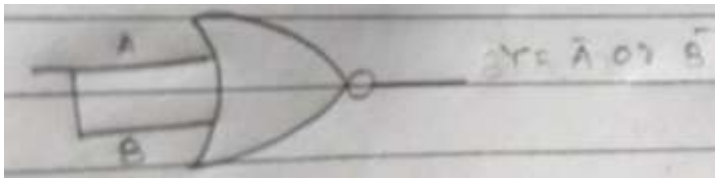
OR GATE



Here, we use DeMorgans Law, where the complement of multiplication (mathematically) of inverted input gives the OR output. We do so, by using two NOT gates for inverting inputs to NAND Gate.

IMPLEMENTATION USING NOR GATE

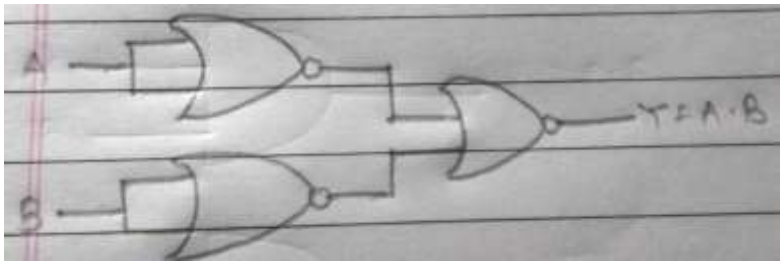
NOT GATE



STEPS

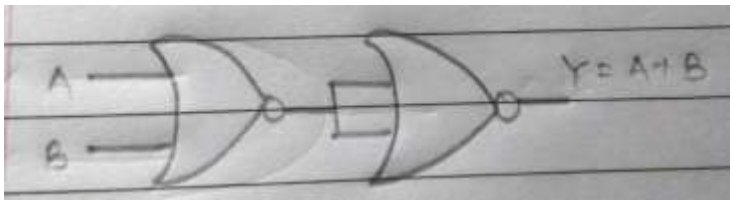
Here, we use similar Logic to that of NAND Gate implementation, where on shorting both the inputs and referring to the truth table for NOR the required output of NOT Gate is obtained.

AND GATE



Here, we use DeMorgans Law, wherethe complement of summation of inverted input gives the AND output. We do so, by using two NOT gates for inverting inputs to NOR Gate.

OR GATE



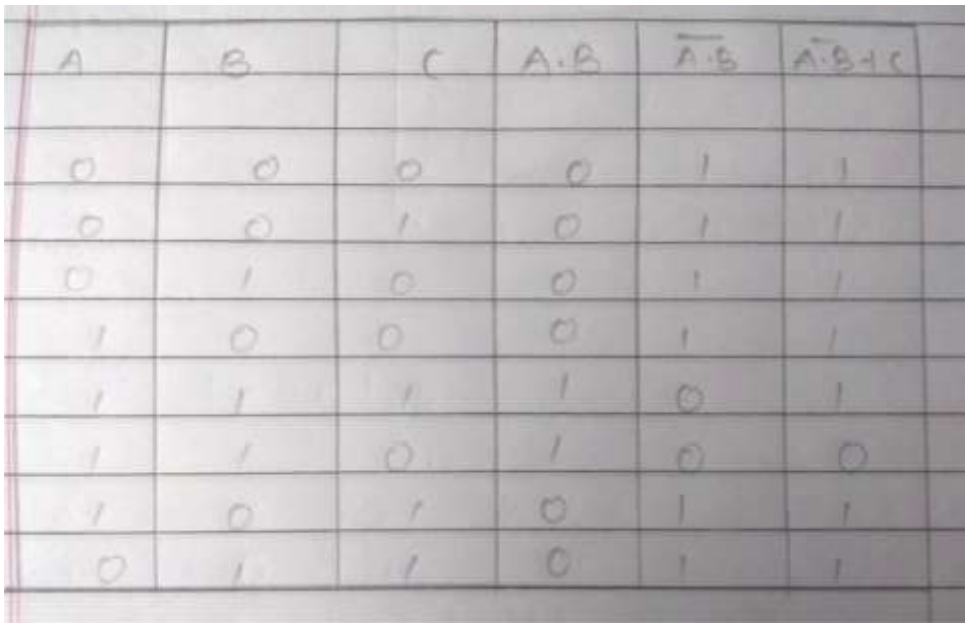
Here, we use a normal NOR Gate, and using the complement's complement property we apply a NOT made using NOR over it to get the respective OR Gate.

Conclusion:

All the Logic Gates have been implemented along with the Implementation of basic Gates using NOR and NAND Gates and hence the aim of the experiment has been achieved successfully.

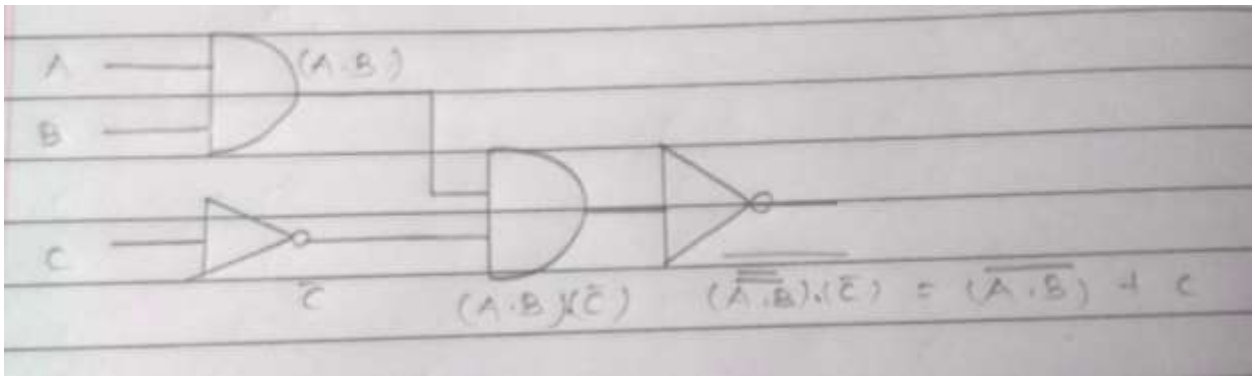
Post Lab Descriptive Questions

1. Verify the expression (A·B)' + C by:
   a) Using NAND Gate directly. (TRUTH TABLE)



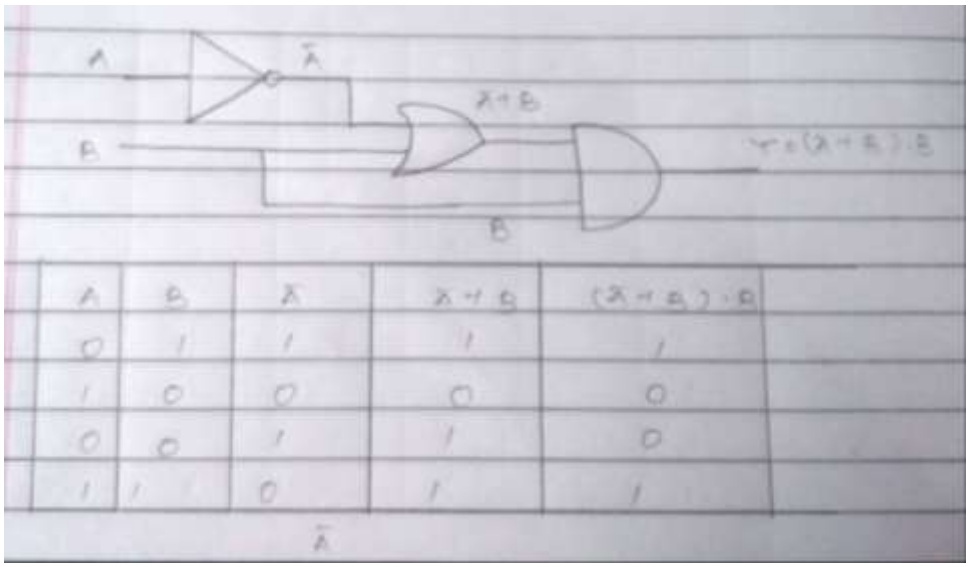
A	B	C	A·B	$\overline{A \cdot B}$	$\overline{A \cdot B} + C$
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	0	1	1
1	0	0	0	1	1
1	1	1	1	0	1
1	1	0	1	0	0
1	0	1	0	1	1
0	1	1	0	1	1

   b) Using AND & NOT gate consecutively.
   (LOGIC GATE COMBINATION DIAGRAM)

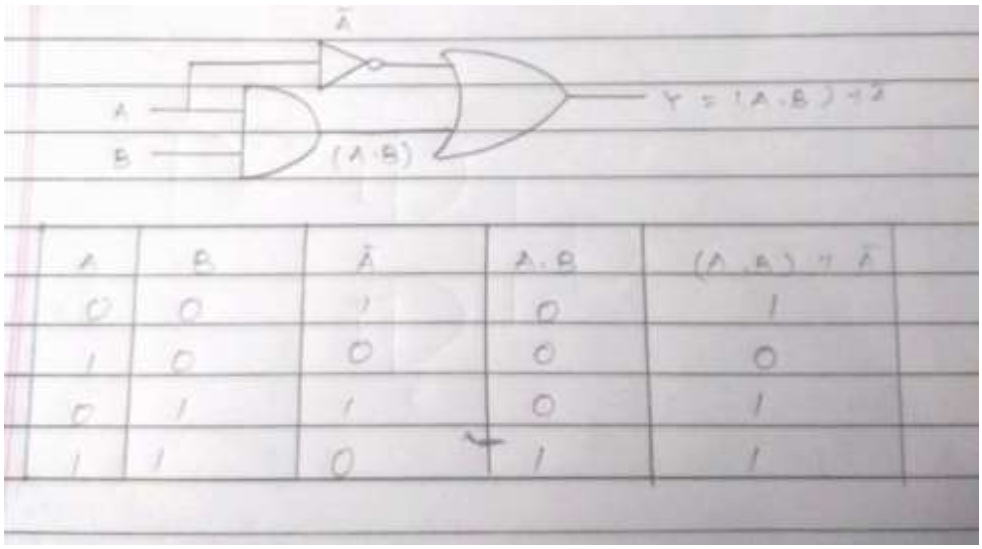


2. Implement the following expressions using combination of gates:
   a) (A'+B)·B
   b) (A·B)+A'
   c) A· (B·B')
   d) (A'⊕B)·A

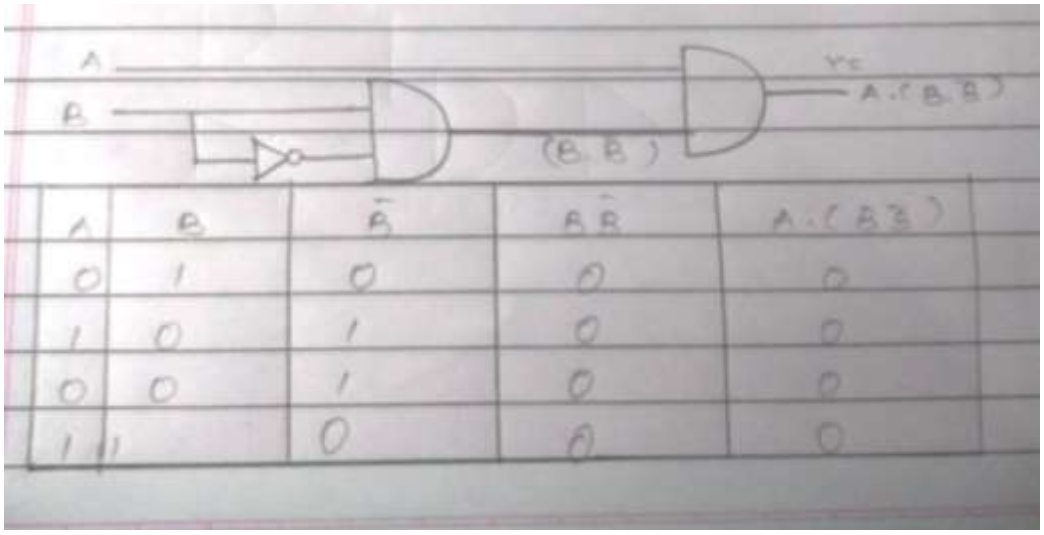
a.



b.

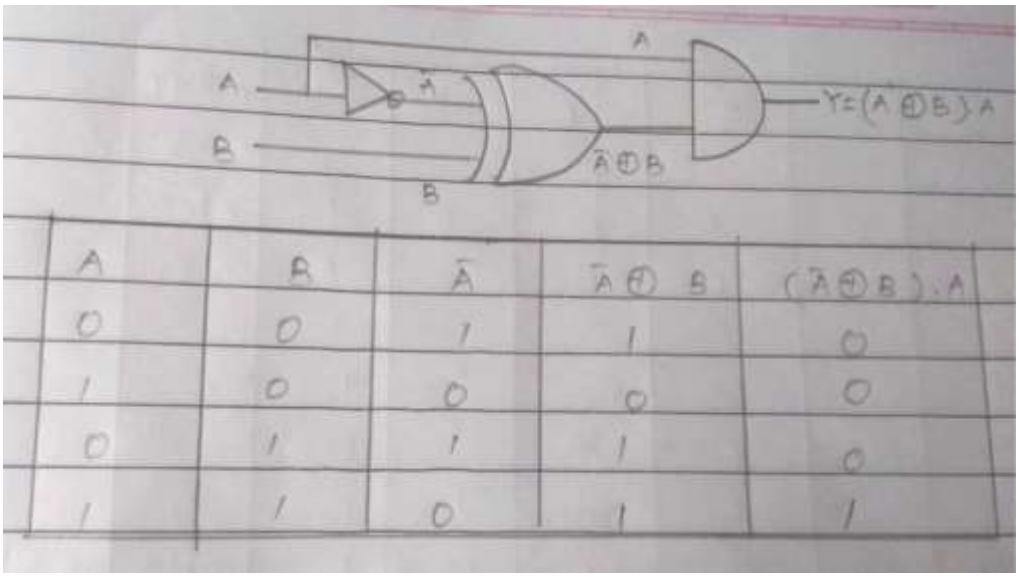


c.



A	B	$\bar{B}$	B $\bar{B}$	A.(B $\bar{B}$)
0	1	0	0	0
1	0	1	0	0
0	0	1	0	0
1	1	0	0	0

d.



A	B	$\bar{A}$	$\bar{A}$ ⊕ B	($\bar{A}$ ⊕ B).A
0	0	1	1	0
1	0	0	0	0
0	1	1	1	0
1	1	0	1	1