DEPT. OF ELECTRICAL & ELECTRONICS ENGINEERING
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, Kattankulathur – 603203.

Title of Experiment	: **Verification and interpretation of Logic Gates.**
Name of the candidate	: **Arnav shukla**
Register Number	: **RA2111050010001**
Date of Experiment	: **07/12/21**

Sl. No.	Marks Split up	Maximum marks (50)	Marks obtained
1	Pre Lab questions	5	
2	Preparation of observation	15	
3	Execution of experiment	15	
4	Calculation / Evaluation of Result	10	
5	Post Lab questions	5	
Total		**50**	

Staff Signature

PRE-LAB QUESTIONS

1. Name the different Logic Gates.

Ans. AND, OR, XOR, NOT, NAND, NOR and XNOR

2. List out the IC names for the different logic Gates.

Ans.

Quad two-input NAND gate (four NAND gates)
Quad two-input NOR gate (four NOR gates)
Hex inverter (six NOT gates)
Quad two-input AND gate (four AND gates)
Quad two-input OR gate (four OR gates)
Quad two-input XOR gate (four XOR gates)

3. What is the Boolean expression for a NOR gate?

Ans. The Boolean expression for a logic NOR gate is denoted by a plus sign, (+) with a line or Overline, (‾) over the expression to signify the NOT or logical negation of the NOR gate giving us the Boolean expression of: A+B = Q.

4. How does a NOR gate work?

Ans. The NOR gate is a digital logic gate that implements logical NOR - it behaves according to the truth table to the right. A HIGH output (1) results if both the inputs to the gate are LOW (0); if one or both input is HIGH (1), a LOW output (0) results. NOR is the result of the negation of the OR operator.

5. Expression for Ex-OR and Ex-NOR?

Ans. The logic symbols ⊕, Jpq, and ⊻ can be used to denote an XOR operation in algebraic expressions. Expand the Ex-NOR function to=A⊕B=(A.B)+(A.B) which mean we can realise this new expression.

Experiment No. **Date :**	**Verification and interpretation of truth tables for AND, OR, NOT, NAND, NOR Exclusive OR (EX-OR), Exclusive NOR (EX-NOR) Gates.**

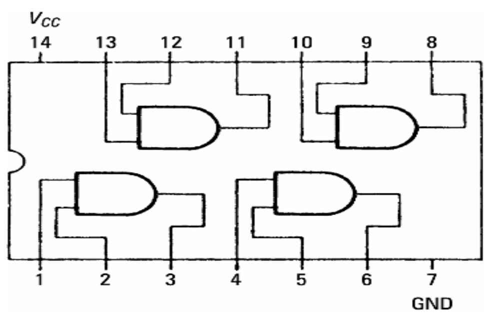
Aim: To verify the Boolean expression using logic gates.

Apparatus: Logic trainer kit, logic gates / ICs, wires.

Theory: Logic gates are electronic circuits which perform logical functions on one or more inputs to produce one output. There are seven logic gates. When all the input combinations of a logic gate are written in a series and their corresponding outputs written along them, then this input/ output combination is called **Truth Table**. The following logic gates and their working are explained.

i) AND Gate

AND gate produces an output as 1, when all its inputs are 1; otherwise the output is 0. This gate can have minimum 2 inputs but output is always one. Its output is 0 when any input is 0.



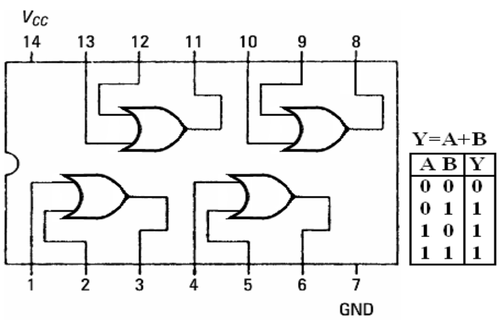
Y=A.B

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

IC 7408

ii) OR Gate

OR gate produces an output as 1, when any or all its inputs are 1; otherwise the output is 0. This gate can have minimum 2 inputs but output is always one. Its output is 0 when all input are 0.



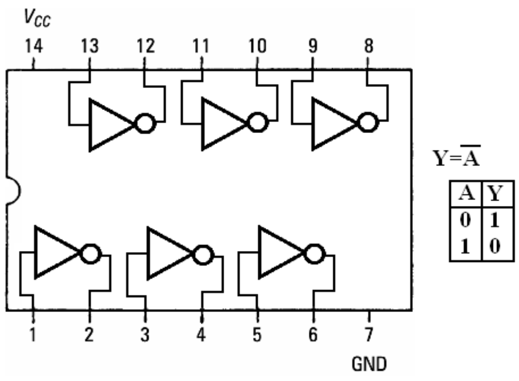
Y=A+B

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

IC 7432

iii) NOT Gate

NOT gate produces the complement of its input. This gate is also called an INVERTER. It always has one input and one output. Its output is 0 when input is 1 and output is 1 when input is 0.



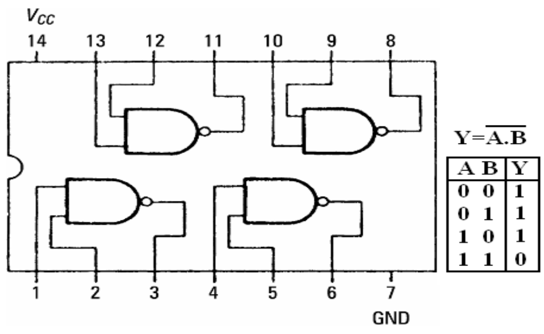
$Y=\overline{A}$

A	Y
0	1
1	0

IC 7404

iv) NAND Gate

NAND gate is actually a series of AND gate with NOT gate. If we connect the output of an AND gate to the input of a NOT gate, this combination will work as NOT-AND or NAND gate. Its output is 1 when any or all inputs are 0, otherwise output is 1.



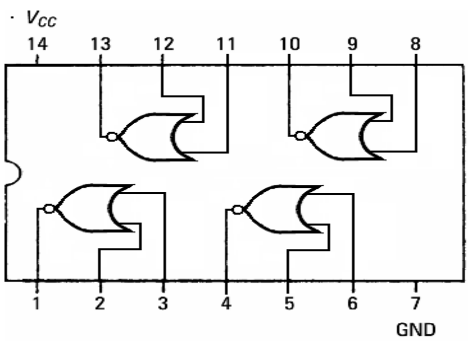
$Y=\overline{A.B}$

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

IC 7400

v) NOR Gate

NOR gate is actually a series of OR gate with NOT gate. If we connect the output of an OR gate to the input of a NOT gate, this combination will work as NOT-OR or NOR gate. Its output is 0 when any or all inputs are 1, otherwise output is 1.



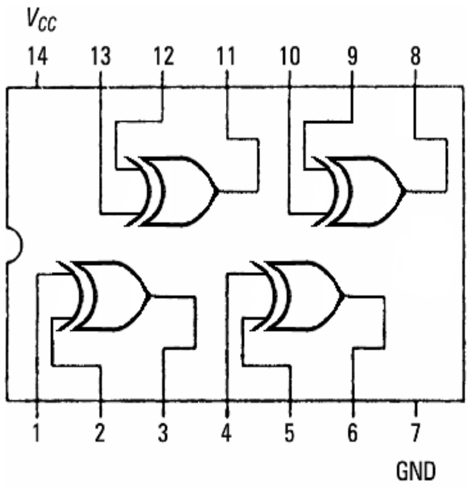
$Y=\overline{A+B}$

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

IC 7402

vi) Exclusive OR (X-OR) Gate

X-OR gate produces an output as 1, when number of 1's at its inputs is **odd**, otherwise output is 0. It has two inputs and one output.



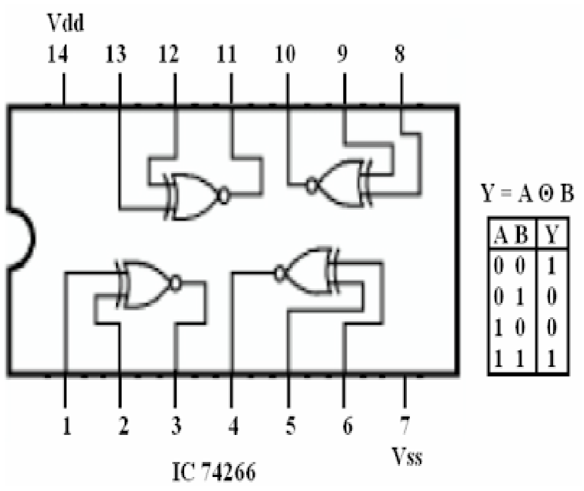
$Y=A\oplus B$

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

IC 7486

vii) Exclusive NOR (X-NOR) Gate

X-NOR gate produces an output as 1, when number of 1's at its inputs is **not odd**, otherwise output is 0. It has two inputs and one output.

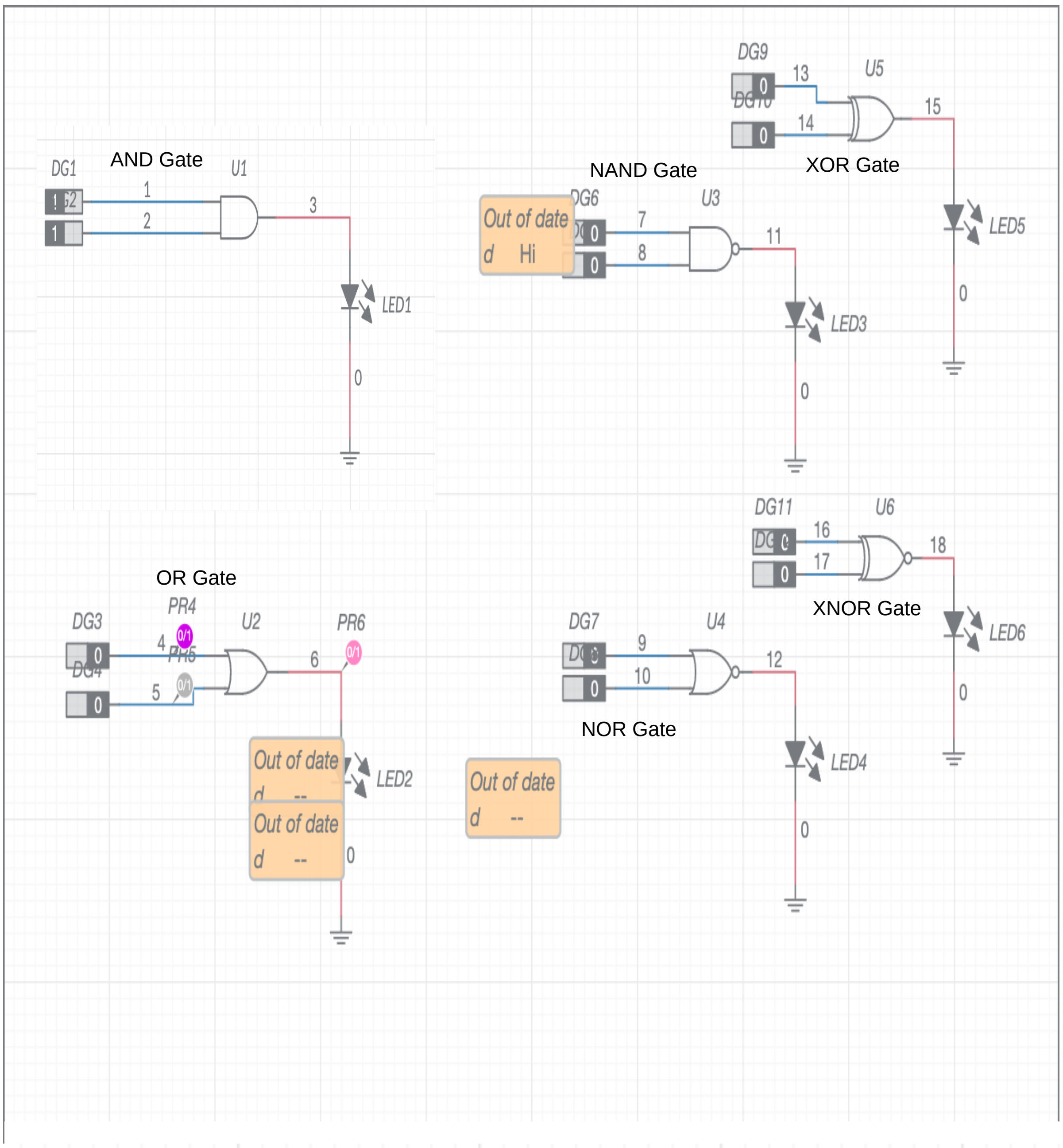


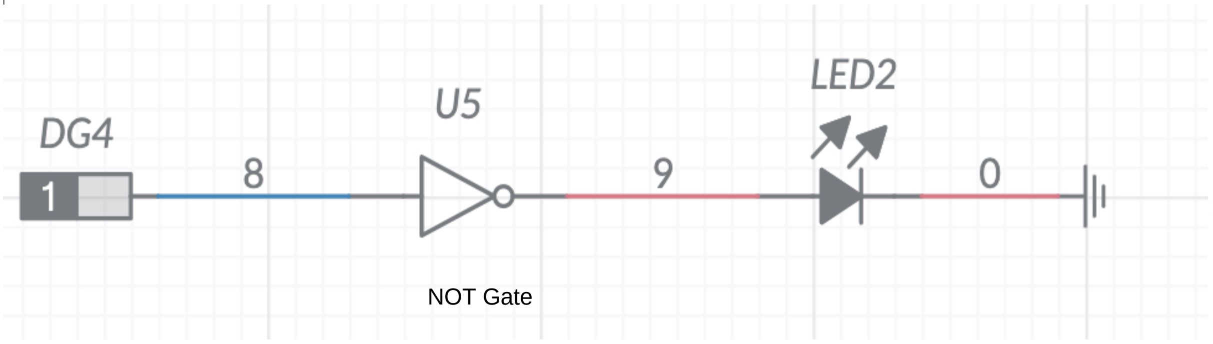
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Procedure:

1. Connect the trainer kit to ac power supply.
2. Connect the inputs of any one logic gate to the logic sources and its output to the logic indicator.
3. Apply various input combinations and observe output for each one.
4. Verify the truth table for each input/ output combination.
5. Repeat the process for all other logic gates.
6. Switch off the ac power supply.

Simulation Circuit for all the Gates:



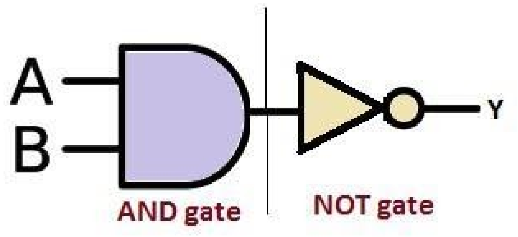


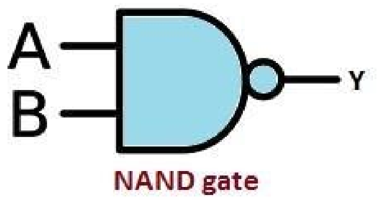
POST-LAB QUESTIONS

1. Name the universal Gates?
Ans. A universal gate is a gate which can implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates.

2. Deduce the logic of AND gate using NAND and NOR?
Ans. A universal gate is a logic gate which can implement any Boolean function without the need to use any other type of logic gate. The NOR gate and NAND gate are universal gates. This means that you can create any logical Boolean expression using only NOR gates or only NAND gates.

3. What is the symbol of NAND gate?
Ans. The symbol of the NAND gate is represented as a combination of AND gate and NOT gate. The Boolean expression is given as Y=A.B.





4. How many NAND gates are required to make an OR gate?
Ans. Three NAND gates are used to make an OR gate.

5. How many NOR gates are required to implement a NAND gate?
Ans. To make a NOR gate perform the NAND function, we must invert all inputs to the NOR gate as well as the NOR gate's output. For a two-input gate, this requires three more NOR gates connected as inverters.

DEPT. OF ELECTRICAL & ELECTRONICS ENGINEERING
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, Kattankulathur – 603203.

Title of Experiment	: **Reduction of Boolean expression using K-map**
Name of the candidate	: **Arnav shukla**
Register Number	: **RA2111050010001**
Date of Experiment	: 07/12/2021

Sl. No.	Marks Split up	Maximum marks (50)	Marks obtained
1	Pre Lab questions	5	
2	Preparation of observation	15	
3	Execution of experiment	15	
4	Calculation / Evaluation of Result	10	
5	Post Lab questions	5	
Total		**50**	

Staff Signature

PRE LAB QUESTIONS:

1. How many Cells are in 4 and 5 Variable K- Map.
Ans. The number of cells in 4 variable K-map is 16 and in 5-variable K-map, we have 32 Cells.

2. What do you mean by don't care condition in K-map or truth table?
Ans. One of the very significant and useful concepts in simplifying the output expression using K-Map is the concept of "Don't Cares". The "Don't Care" conditions allow us to replace the empty cell of a K-Map to form a grouping of the variables which is larger than that of forming groups without don't cares.

3. Write the Distributive property of Boolean Algebra.
Ans. Distributive Law states that the multiplication of two variables and adding the result with a variable will result in the same value as multiplication of addition of the variable with individual variables. For example: A + BC = (A + B) (A + C)

4. Write down the De Morgan law.
Ans. De Morgan's Law states that the complement of the union of two sets is the intersection of their complements and the complement of the intersection of two sets is the union of their complements. These are mentioned after the great mathematician De Morgan. This law can be expressed as $(A \cup B)' = A' \cap B'$

5. State the difference between SOP and POS.
Ans. : : The SOP (Sum of Product) and POS (Product of Sum) are the methods for deducing a particular logic function. The prior difference between the SOP and POS is that the SOP contains the OR of the multiple product terms. Conversely, POS produces a logical expression comprised of the AND of the multiple OR terms.

Experiment No. Date :	**Reduction of Logic Expression using Karnaugh map (K- Map)**

Aim: To simply and verify the Boolean expression using K-map.

Apparatus: Logic trainer kit, logic gates / ICs, wires.

Theory:

Karnaugh maps: Karnaugh maps or K-maps for short, provide another means of simplifying and optimizing logical expressions. This is a graphical technique that utilizes a sum of product (SOP) form. SOP forms combine terms that have been ANDed together that then get ORed together. This format lends itself to the use of De Morgan's law which allows the final result to be built with only NAND gates. The K-map is best used with logical functions with four or less input variables. One of the advantages of using K-maps for reduction is that it is easier to see when a circuit has been fully simplified. Another advantage is that using K-maps leads to a more structured process for minimization. In order to use a K-map, the truth table for a logical expression is transferred to a K-map grid. The grid for two, three, and four input expressions are provided in the tables below. Each cell corresponds to one row in a truth table or one given state in the logical expression. The order of the items in the grid is not random at all; they are set so that any adjacent cell differs in value by the change in only one variable. Because of this, items can be grouped together easily in rectangular blocks of two, four, and eight to find the minimal number of groupings that can cover the entire expression. Note that diagonal cells require that the value of more than two inputs change, and that they also do not form rectangles.

	A'B' 00	A'B 01	AB 11	AB' 10
C' 0				
C 1				

	A' 0	A 1
B' 0		
B 1		

Figure 1. Three variables K Map **Figure 2. Two variables K- Map**

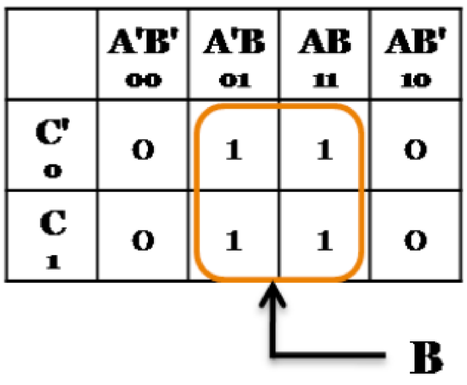
Given expression

F(C,A,B) = CAB + C'AB + CA'B + C'A'B

Simplification Using Boolean Properties

CAB + C'AB + CA'B + C'A'B = AB(C + C') + A'B(C + C') Distributive Property
= AB + A'B C + C' is always true
= (A + A')B Distributive Property
= B A + A' is always true

Simplification using K- Map



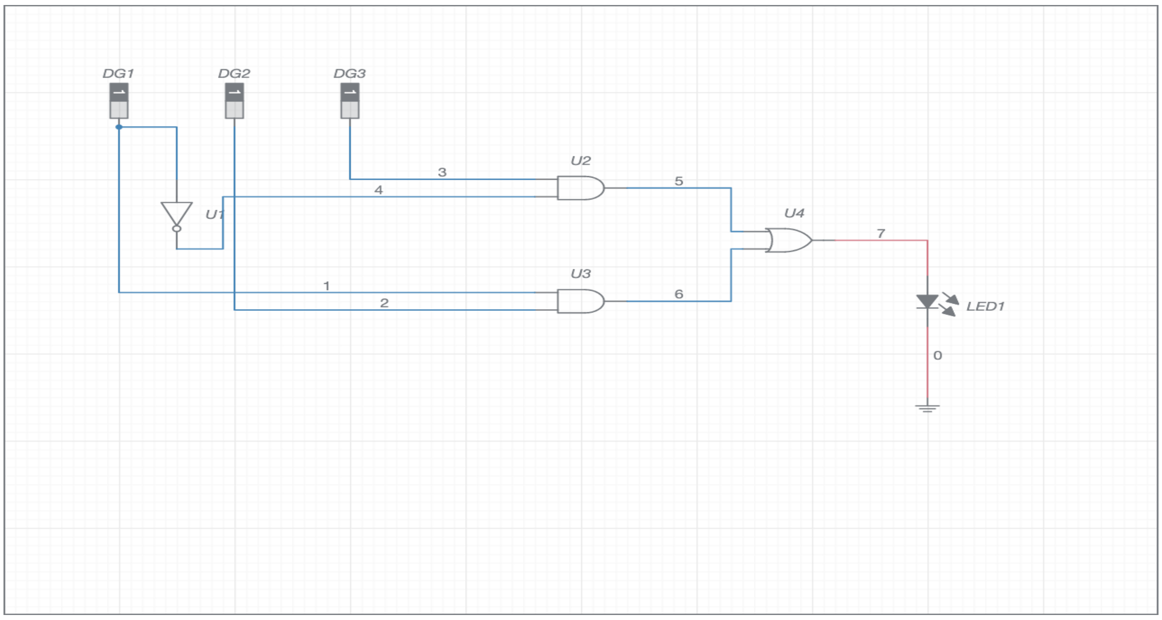
	A'B' 00	A'B 01	AB 11	AB' 10
C' 0	0	1	1	0
C 1	0	1	1	0

<u>**Procedure:**</u>

1. Connect the trainer kit to ac power supply.
2. Connect the circuit based on the given logic functions to be simplified.

3. Connect the inputs of first stage to logic sources and output of the last gate to logic indicator.
4. Apply various input combinations and observe output for each one.
5. Verify the output before and after reducing the expression.
6. Switch off the ac power supply.

Simulation Output:



Result: Hence, the given expression output has been verified both via reducing the expression using K-Map and via E-Circuit

Post-lab questions

1. Simply the expression F=AB+AB'

Ans: F = AB + AB'
= A (B+B')
=A (Because B + B' = 1)

2. Name the different reduction techniques

Ans. Boolean Laws: Distributive, Communicative, De Morgan's'; and K-Map

3. Give the merits and demerits of K-map

Ans. *Advantages of K-Map:*

- Minimizes Boolean expressions without the need using various Boolean theorems & computations.
- Minimizes number of Logical gates used.

Disadvantages of K-Map:

- It is not suitable for computer reduction.
- It is not suitable when the number of variables involved exceed four.
- Care must be taken to field in every cell with the relevant entry, such as a 0, 1 (or) don't care terms.

4. What are differences between K-map and Quine McCluskey?
Ans. Karnaugh map (K-map) and Quine-McCluskey (QM) methods are well known methods to simplify Boolean expression. K-map method becomes complex beyond five variable Boolean expression. Quine-McCluskey method is computer-based technique for minimization of Boolean function and it is faster than K-map method.

5. Give steps for reducing two variable expression using K-map?
Ans. Ans: Following are the steps for reducing two variable expressions:
Step 1- First, let’s construct the truth table for the given equation.
Step 2- We put 1 at the output terms given in equation.
Step 3- We can create 2 groups by following the rules for grouping, one is by combining (X’, Y) and (X’, Y’) terms and the other is by combining (X, Y’) and (X’, Y’) terms. Here the lower right cell is used in both groups.
Step 4- After grouping the variables, the next step is determining the minimized expression. By reducing each group, we obtain a conjunction of the minimized expression such as by taking out the common terms from two groups, i.e., X’ and Y’. So, the reduced equation will be X’ +Y’.