K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)
Department of Computer Engineering

Course Name:	**Digital Design Laboratory**	**Semester:**	**III**
Date of Performance:	**17 / 07 / 25**	**Batch No:**	**B-2**
Faculty Name:		**Roll No:**	**16010124107**
Faculty Sign & Date:		**Grade/Marks:**	**___/25**

Experiment No: 1
Title: Study of Basic Gates and Universal Gates

Aim and Objective of the Experiment:

Understand Basic Logic Gates and Universal Gates

COs to be achieved:

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

Tools used:

Trainer kits

Theory:

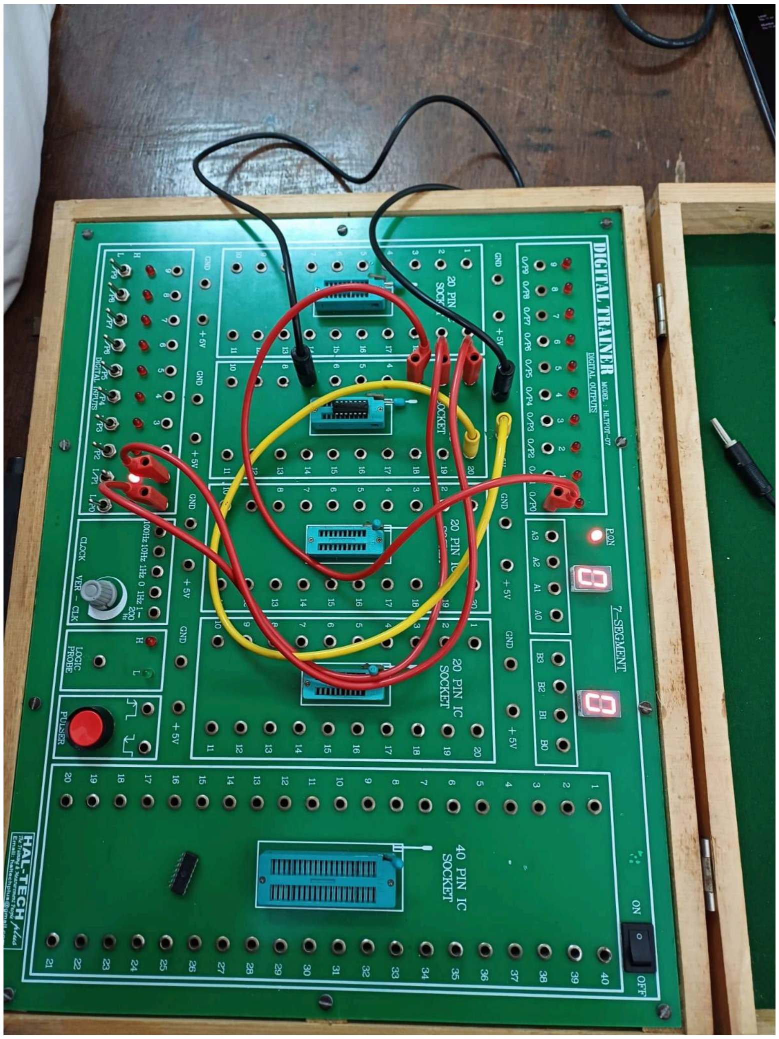
Logic gates are electronic circuits that perform logical operations on one or more input signals to produce an output signal based on a set of logical rules. Logic gates can be classified into the following categories:

1. Basic Gates:
   a. AND Gate: The AND gate produces a high output (1) only when all of its inputs are high (1).
   b. OR Gate: The OR gate produces a high output (1) if any of its inputs is high (1).
   c. NOT Gate (Inverter): The NOT gate produces the logical complement of its input. It takes a single input and produces the opposite value as the output.
2. Derived Gates:
   a. NAND Gate: The NAND gate is a combination of an AND gate followed by a NOT gate. It produces the inverse of the AND gate's output. It outputs a low (0) only when all of its inputs are high (1).
   b. NOR Gate: The NOR gate is a combination of an OR gate followed by a NOT gate. It produces the inverse of the OR gate's output. It outputs a high (1) only when all of its inputs are low (0).
   c. XOR Gate (Exclusive OR): The XOR gate produces a high output (1) when the number of high inputs is odd. It outputs a low (0) when the number of high inputs is even.
   d. XNOR Gate (Exclusive NOR): The XNOR gate produces a high output (1) when the number of high inputs is even. It outputs a low (0) when the number of high inputs is odd.
3. Universal Gates:

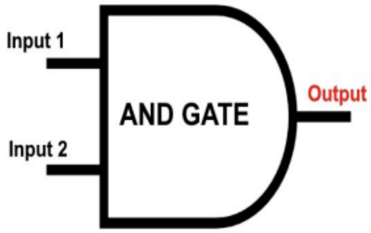
NAND and NOR gates are considered universal gates because any logic function can be implemented using only NAND gates or only NOR gates. This means that with a sufficient number of NAND or NOR gates, you can create circuits that can perform any logical operation.

Implementation Details

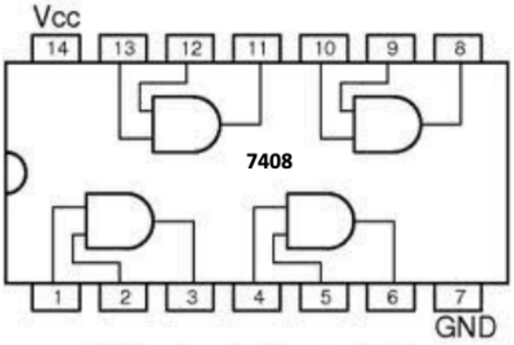
1. AND Gate: Y = A*B



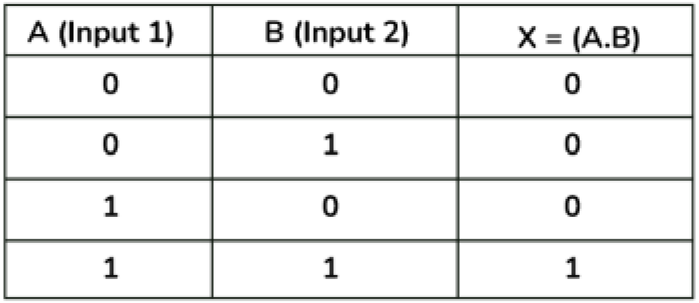
Symbol



Pin Diagram

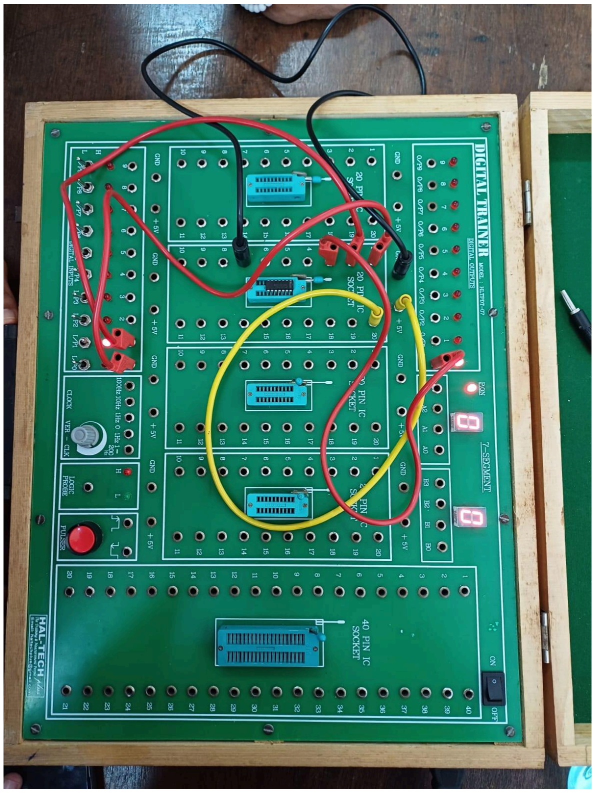


Truth Table:

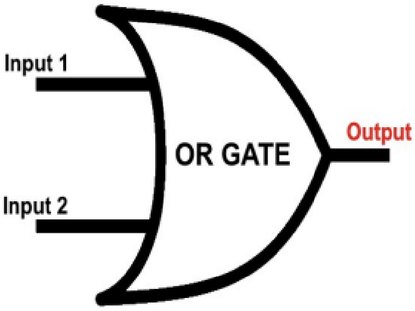


A (Input 1)	B (Input 2)	X = (A.B)
0	0	0
0	1	0
1	0	0
1	1	1

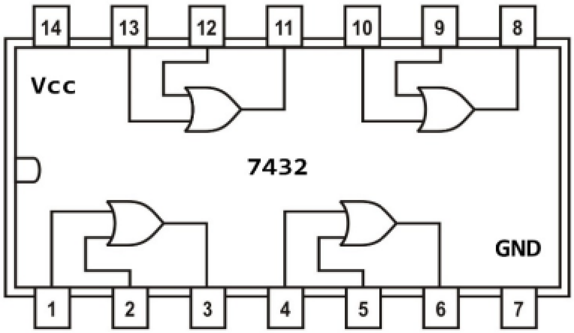
2. OR Gate: Y = A+B



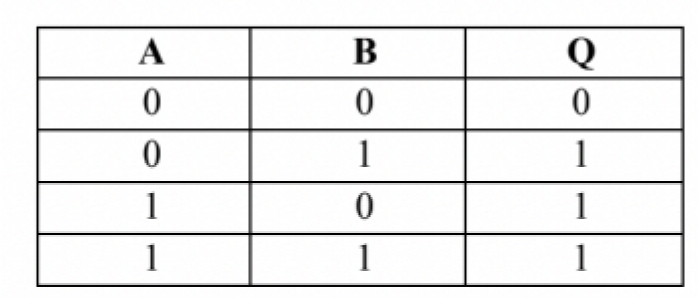
Symbol



Pin Diagram

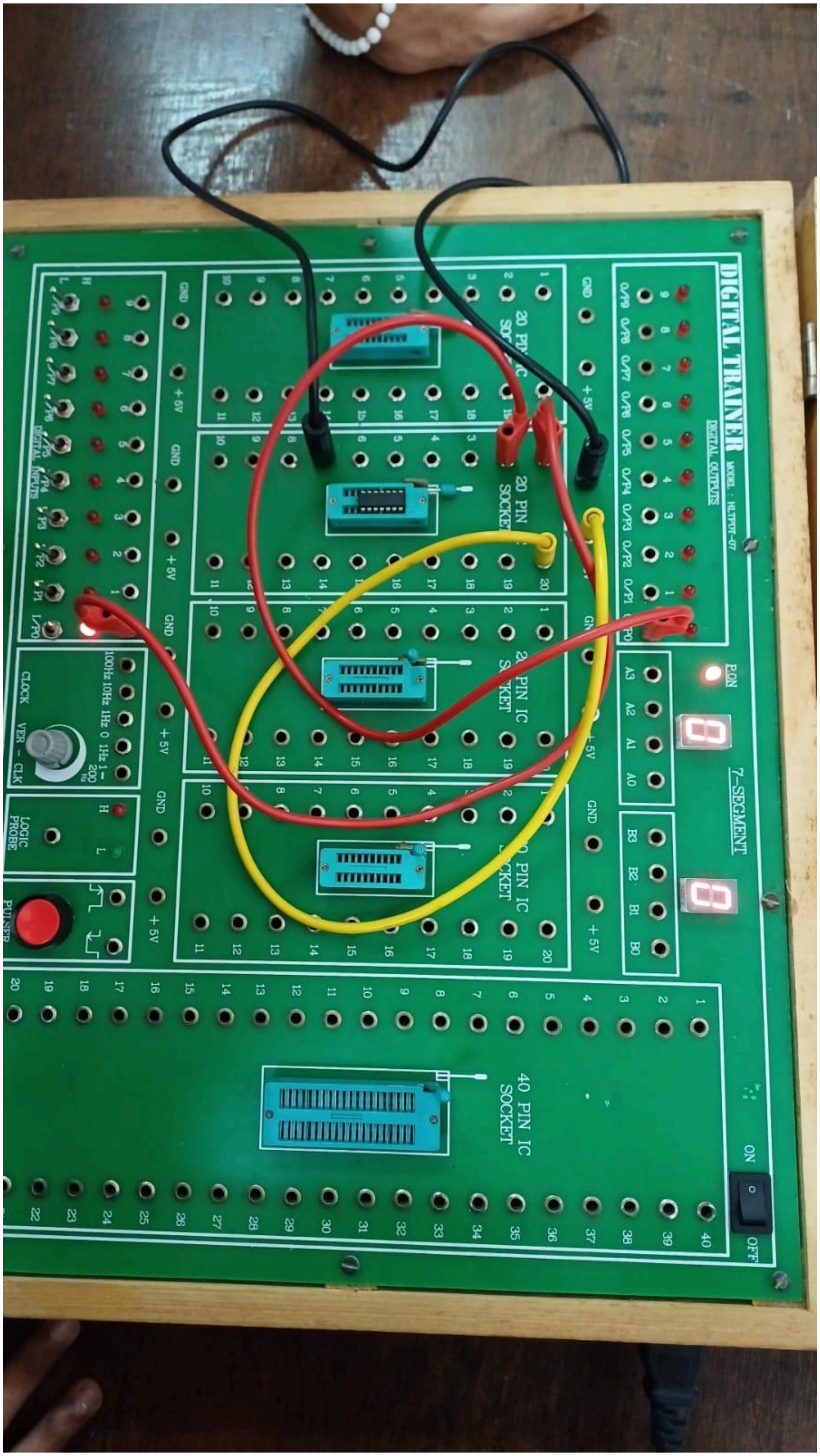


Truth Table:

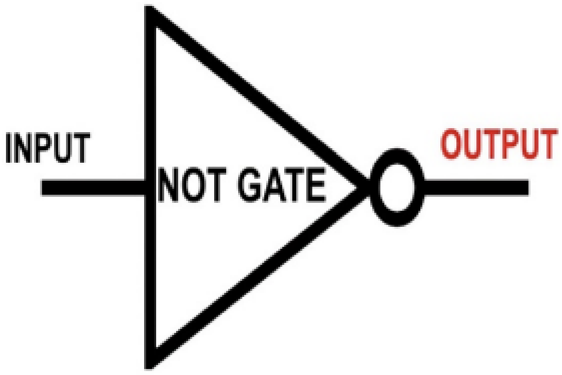


A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

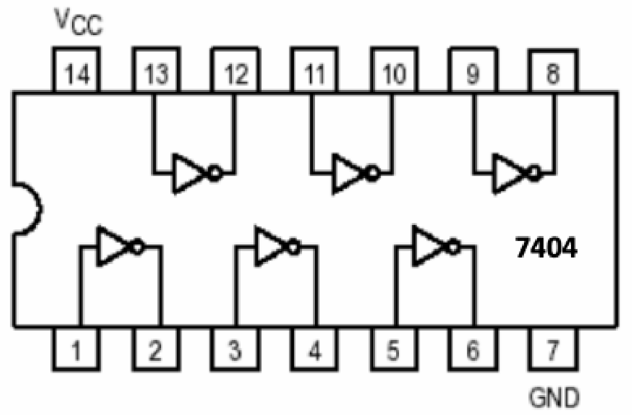
3. NOT Gate: Y = A’


DIGITAL TRAINER
MODEL : HLTPDT-07
DIGITAL OUTPUTS
DIGITAL INPUTS
20 PIN IC SOCKET
20 PIN SOCKET
7-SEGMENT
CLOCK
VER - CLK
LOGIC PROBE
PULSER
40 PIN IC SOCKET
GND
+5V
ON
OFF

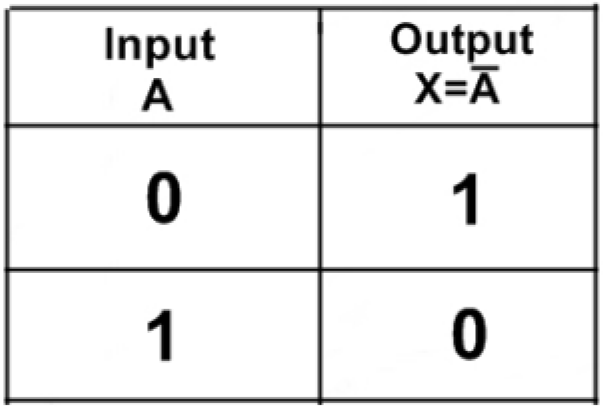
Symbol



Pin Diagram

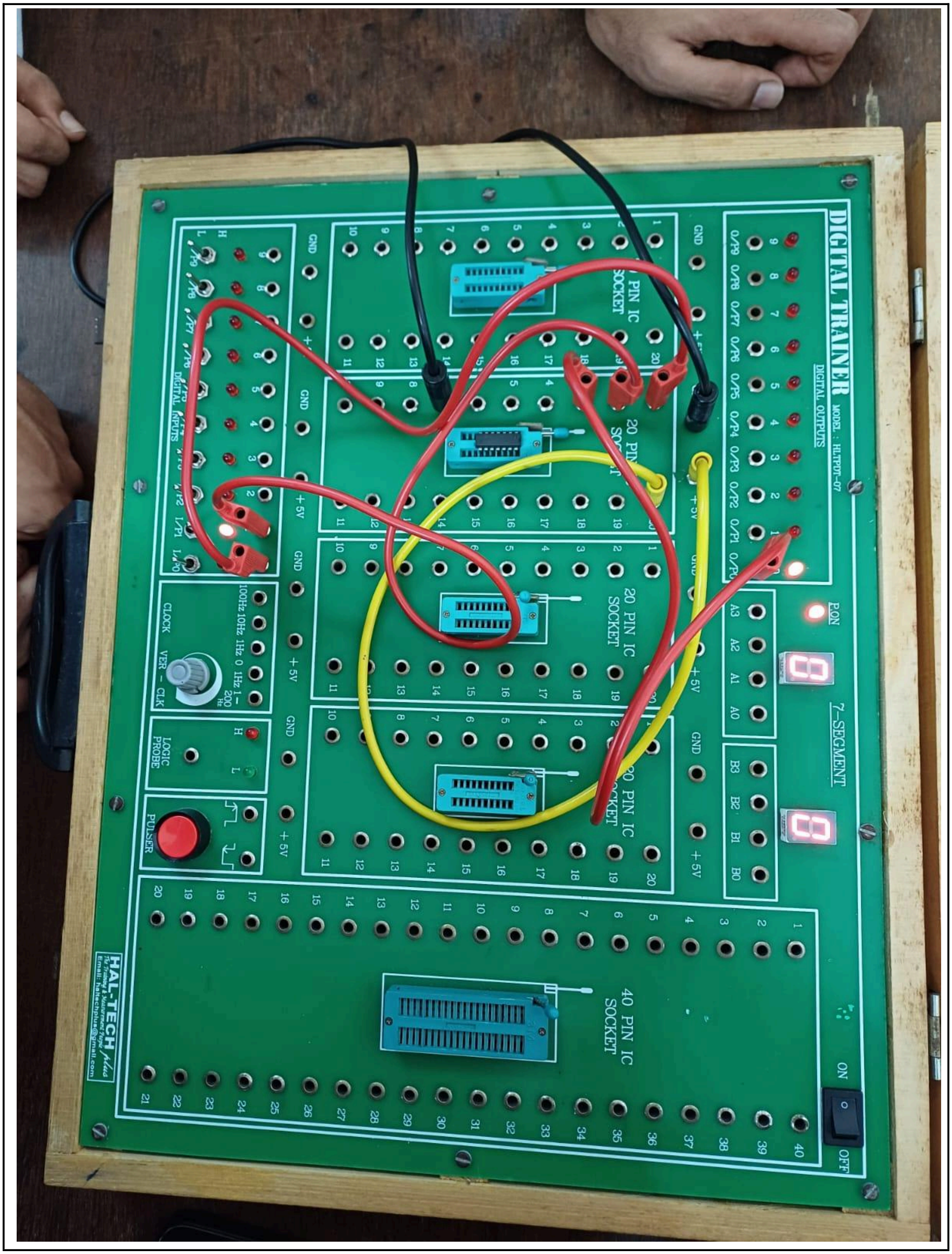


Truth Table:

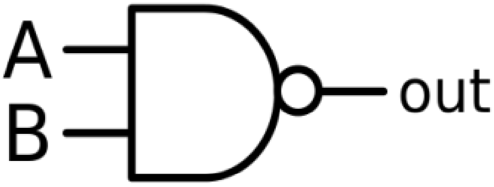


Input A	Output $X=\overline{A}$
0	1
1	0

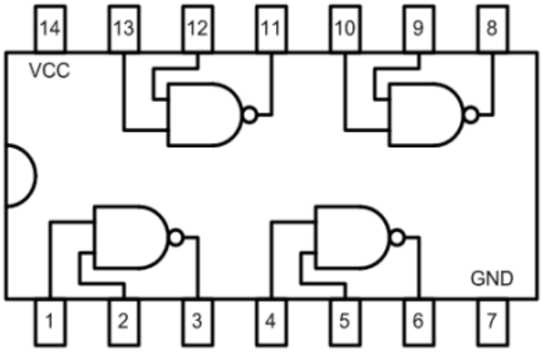
4. NAND Gate: Y = (A*B)’


DIGITAL TRAINER
MODEL : HLTPDT-07
DIGITAL OUTPUTS
DIGITAL INPUTS
20 PIN IC SOCKET
40 PIN IC SOCKET
7-SEGMENT
CLOCK
LOGIC PROBE
PULSER
GND
+5V
PON
ON
OFF
HAL-TECH

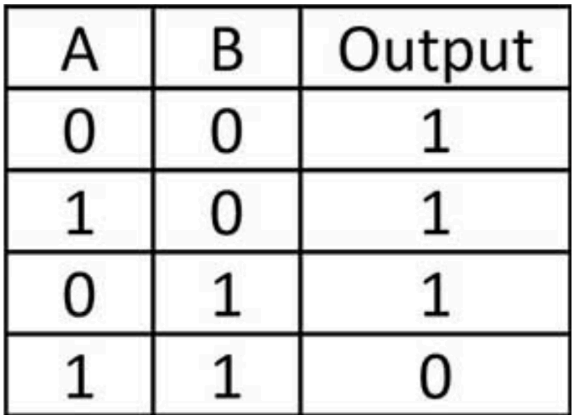
Symbol



Pin Diagram

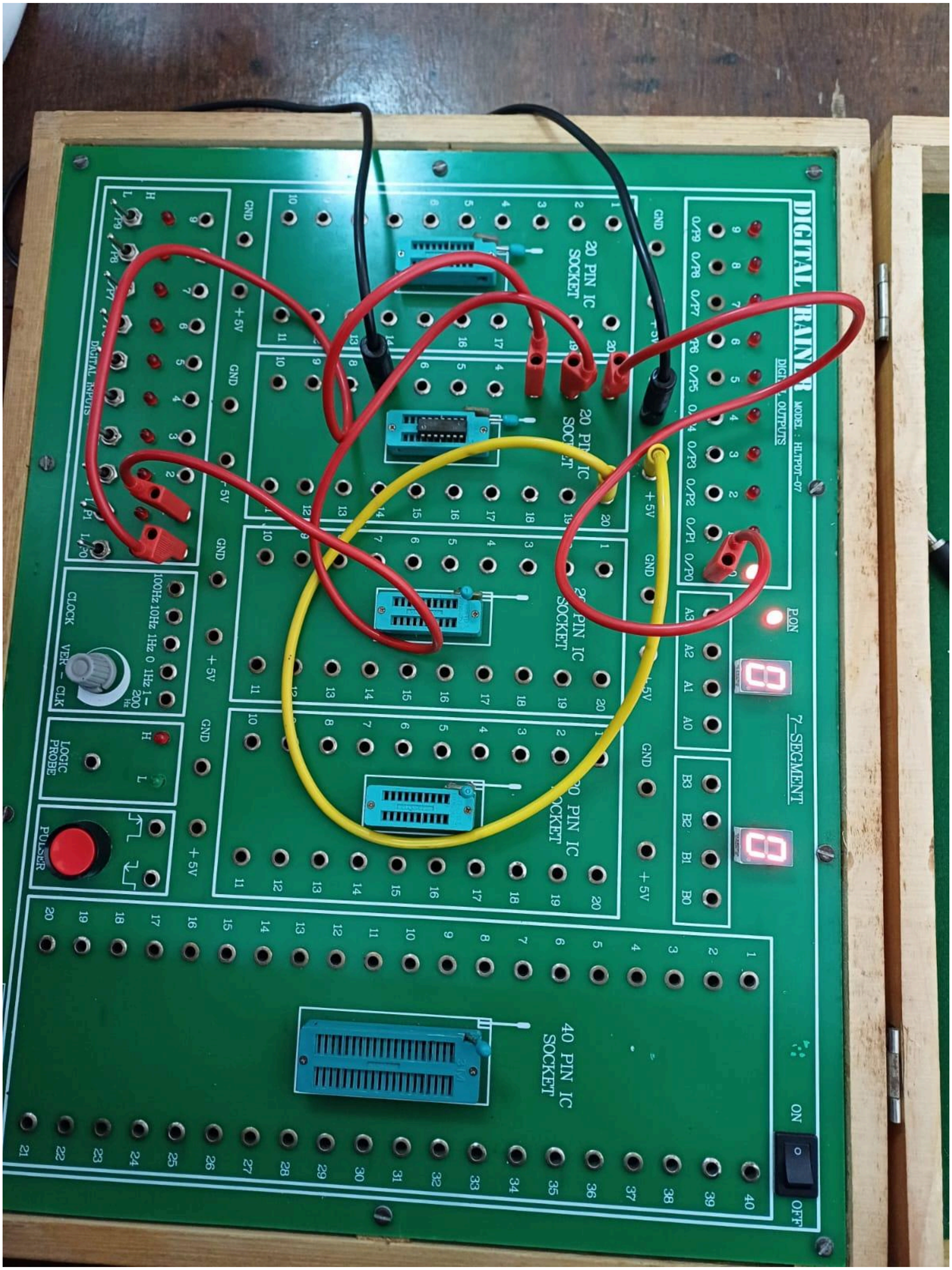


Truth Table:

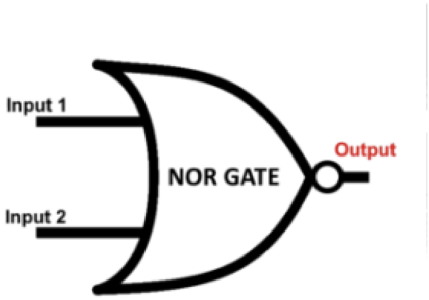


A	B	Output
0	0	1
1	0	1
0	1	1
1	1	0

5. NOR Gate: Y = (A+B)’

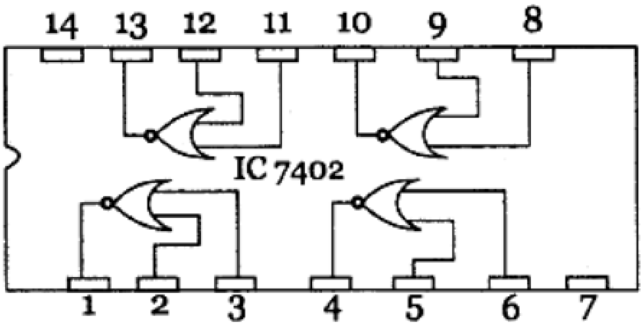

DIGITAL TRAINER
MODEL : HLTPOT-07
DIGITAL INPUTS
DIGITAL OUTPUTS
20 PIN IC SOCKET
40 PIN IC SOCKET
7-SEGMENT
CLOCK
LOGIC PROBE
PULSER
PON
ON
OFF

Symbol

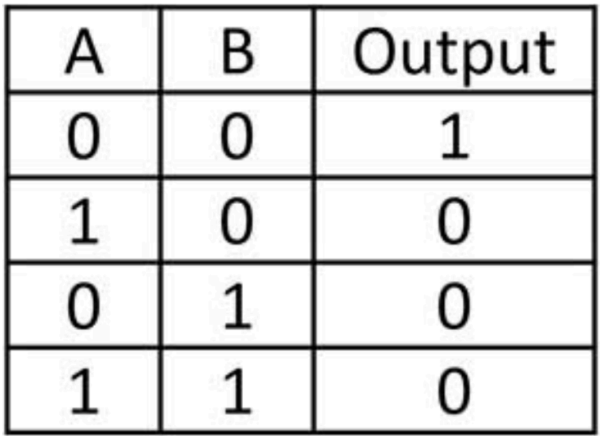


Pin Diagram

t



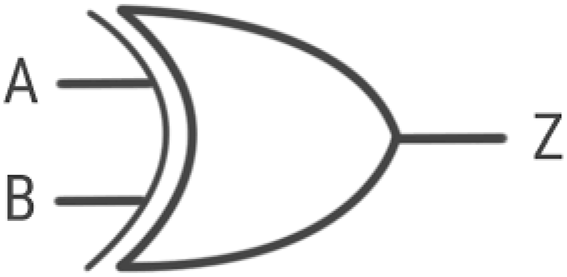
Truth Table:



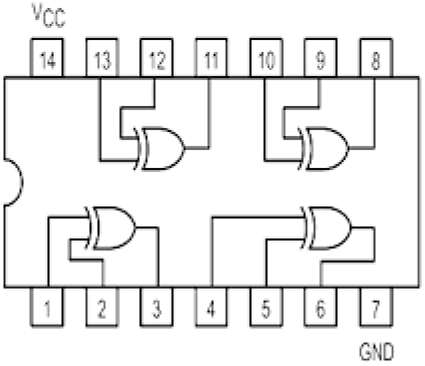
A	B	Output
0	0	1
1	0	0
0	1	0
1	1	0

6. XOR Gate: Y =

Symbol



Pin Diagram

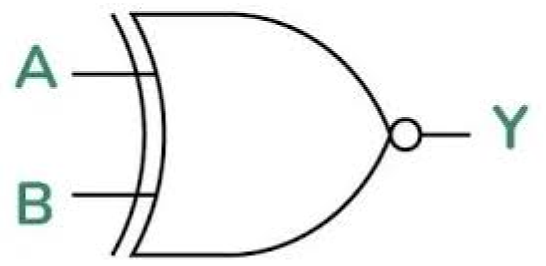


Truth Table:

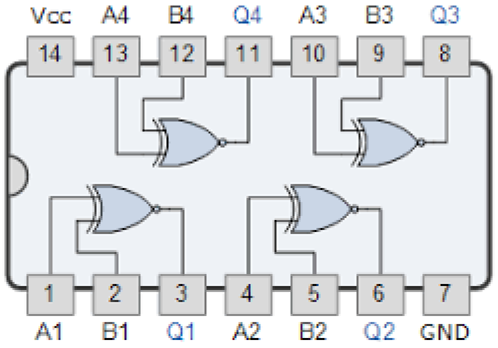
A	B	A **XOR** B
0	0	0
0	1	1
1	0	1
1	1	0

7. XNOR Gate: Y =

Symbol



Pin Diagram

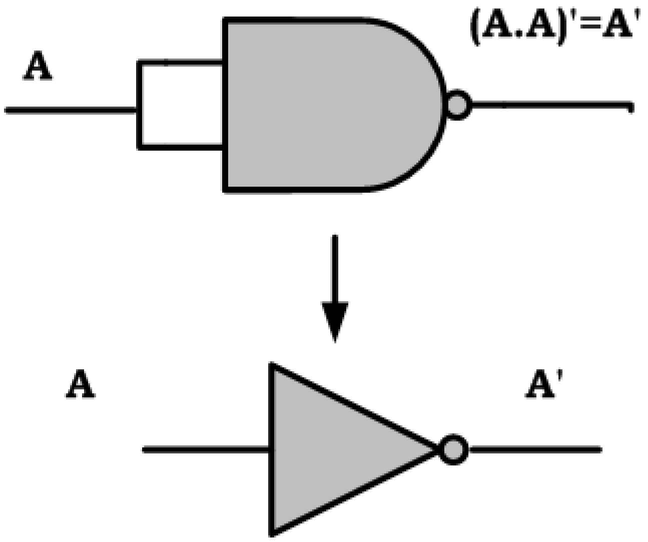


Truth Table:

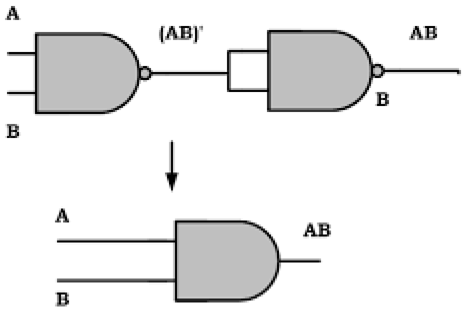
A	B	A **XNOR** B
0	0	1
0	1	0
1	0	0
1	1	1

Implementation Using NAND Gate

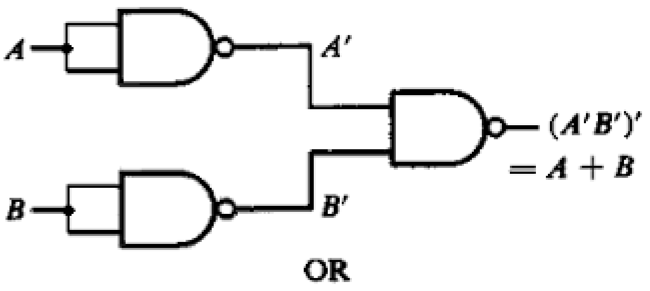
NOT GATE



AND GATE

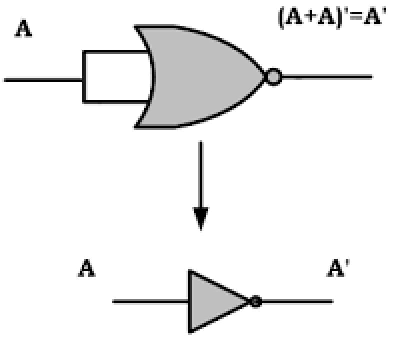


OR GATE

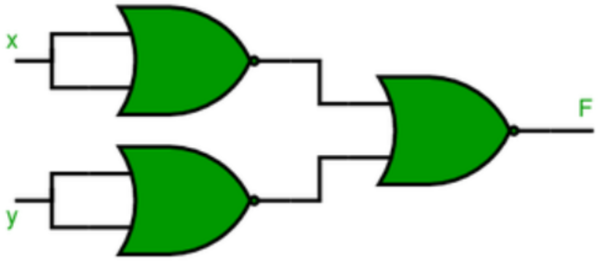


Implementation Using NOR Gate

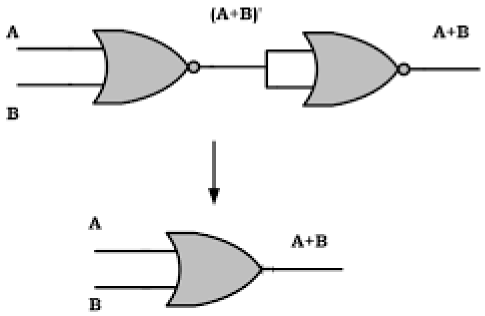
NOT GATE



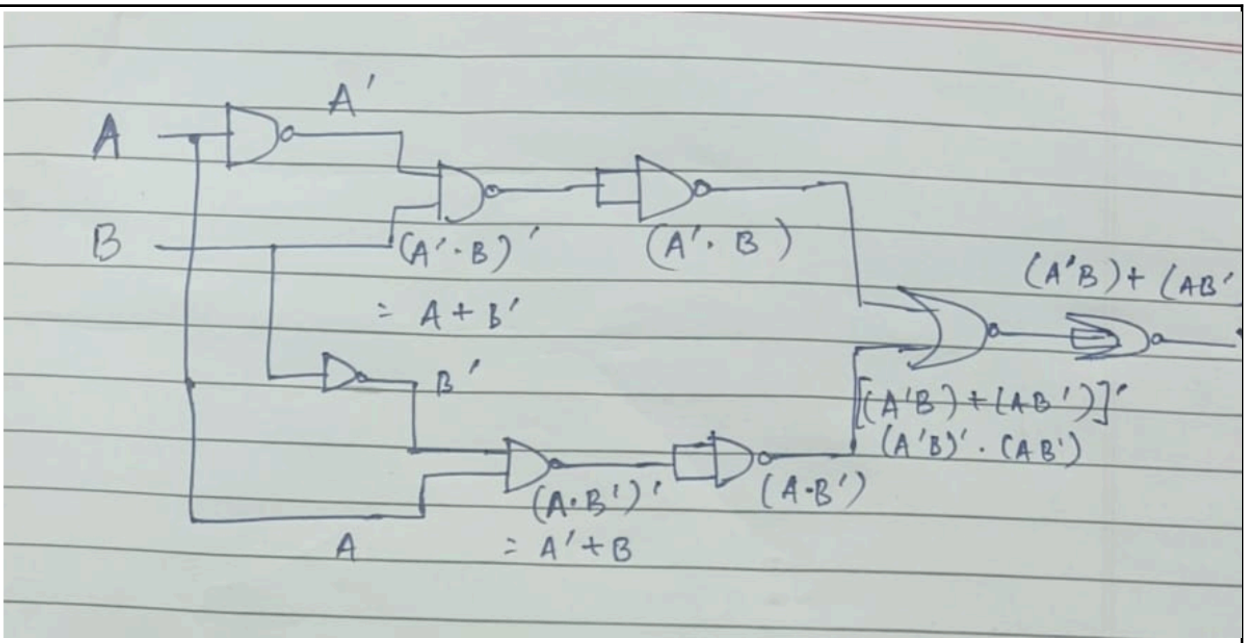
AND GATE



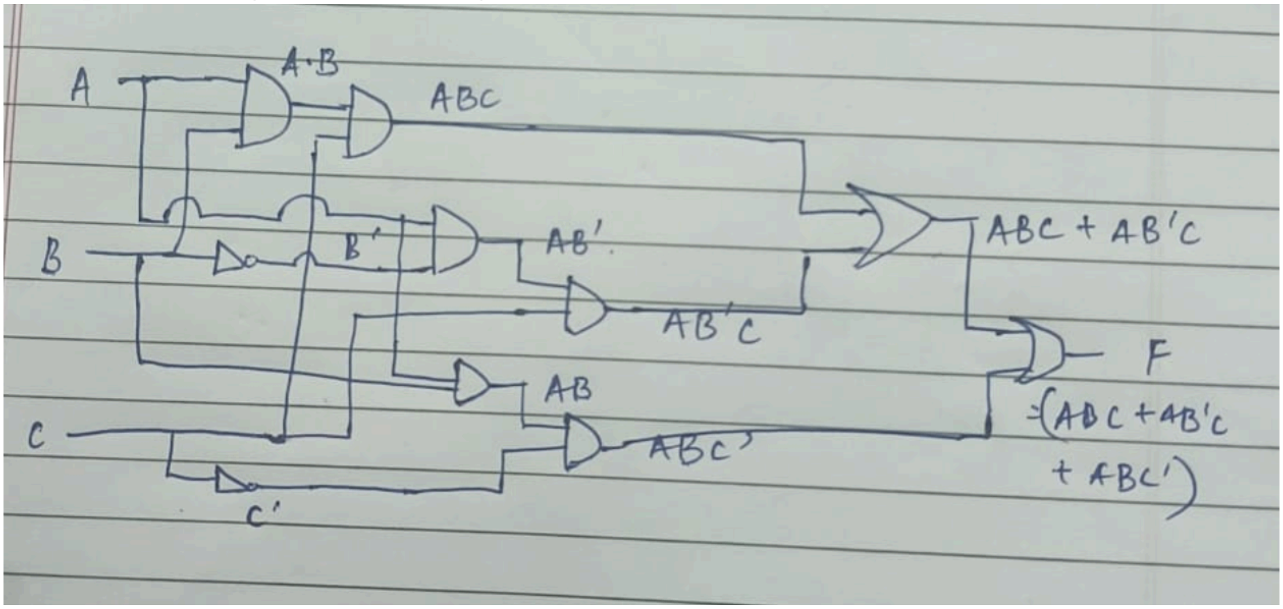
OR GATE



Post Lab Subjective/Objective type Questions:
1. Implement the Boolean function using NAND gates and NOR gates F=A’B + AB’



2. Implement using combination of gates F = ABC + AB’C + ABC’



Conclusion:

The NAND and NOR gates are called universal gates. They can be implemented and combined together to form other gates like NOT, AND, OR, etc. by implementing DeMorgan’s Theorem. The basic gates are just AND, OR, and NOT. The gates made by combining these are called derived gates: XOR, XNOR, NAND, and NOR are all derived gates.

Signature of faculty in-charge with Date: