NAME OF THE EXPERIMENT: PROVE NAND GATE IS UNIVERSAL GATE.

OBJECTIVE:

i. Understanding the concept of universal NAND gate.
ii. Implimenting the basic logic gate (AND, OR, NOT) Using NAND gate.
iii. Implimenting boolean function using NAND gate.
iv. Understand gate level minimization.

DISCRIPTION OF THE PROBLEM:

Universal gates are gates which can be used to implement all other gates. This is useful as manufacturers only need to produce one type of universal gate to be able to use all other gates. Universal gate is a gate which can implement any boolean function without need to use any other gate type. The NAND gate is a universal gate. In practice, this is advantageous since IC digital logic families. In fact AND gate is tipically implement as a NAND gate followed by an Inverter not the other way around.

~~PROT~~ REQUIRED COMPONENTS:

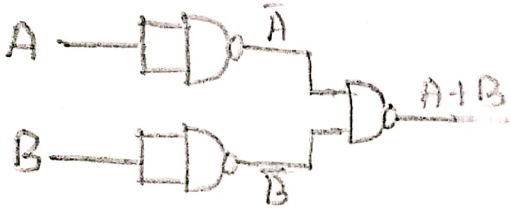
i. 7400 Ic of NAND gate
ii. Bread board
iii. Trainer meachine
iv. Wires
v. +5 Vcc power Supply

CIRCUIT DIAGRAM:

figure: NAND gate

equivalent, 

figure: Implementing NOT gate using NAND gate



equivalent 

figure: Implementing OR gate using NAND gate

equivalent, 

figure: Implementing AND gate using NAND gate

TRUTH TABLE:

Truth table of NAND gate:

A	B	Output (Y)
0	0	1
0	1	1
1	0	1
1	1	0

TRUTH table of NOT, AND & OR gates using NAND gate;

A	A	Y
0	0	1
1	1	0

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

A	B	X	Y	Z
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1

TIMING DIAGRAM:

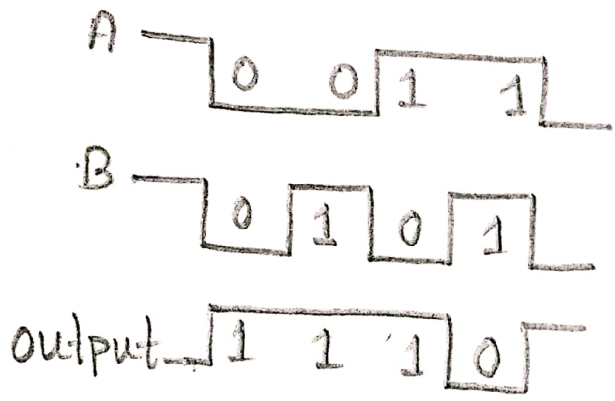


figure: timing diagram of NAND gate.

Ic CONFIGURATION:

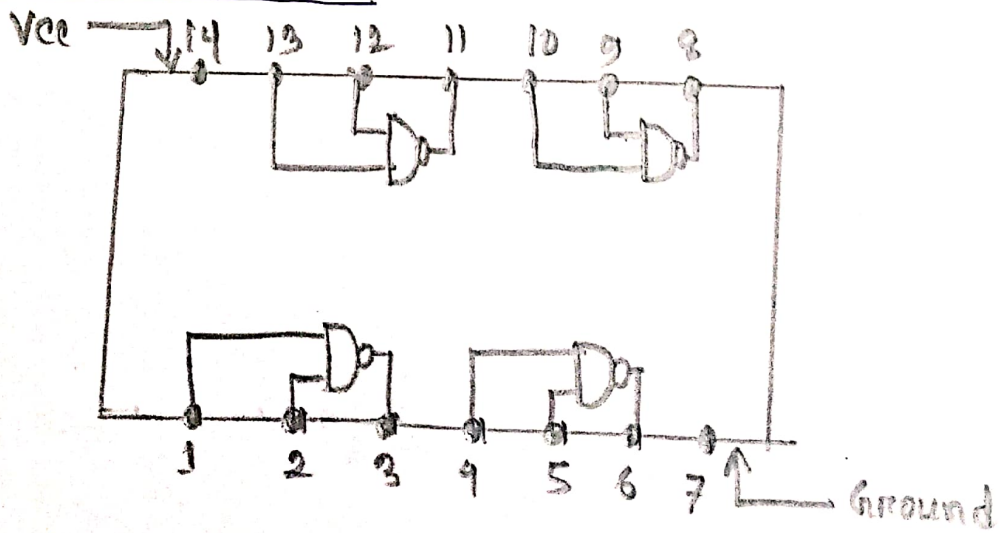


figure: Ic configuration of a NAND gate

INPUT & OUTPUT:

In digital electronics a NAND gate (NOT-AND) is a logic gate which produces an output which is false only if all its inputs are true. Thus it's output is complement to that of an AND gate. A Low (0) output results only if all the inputs to the gate are high(1). If any input is low (0) a high output results. So, we can say that NAND gate is a Universal gate (proved).

NAME OF THE EXPERIMENT: PROVE NOR GATE IS AN UNIVERSAL GATE.

OBJECTIVE:

i. Understanding the concept of universal NOR gate.
ii. Implementing the basic logic gate(AND, OR, NOT) Using NOR gate.
iii. Implementing Bolean function using NOR gate
iv. Understanding gate level minimization.

DESCRIPTION OF THE PROBLEM:

Universal gates are gates which can be used to implement all other gates. This is useful as manufacturers only needs to produce one type of universal gate to be able to use all other gates. Universal gate is a gate which can implement any boolean function without need to use any other gate type. The NOR gate is universal gate. In practice, this is advantageous since Ic digit logic family, In fact, NOR gate is typically implemented as a OR gate followed by an inverter not the other way around.

REQUIRED COMPONENTS:

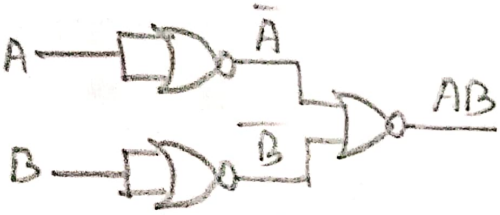
1. 7402 Ic of NOR gate
ii. Breadboard
iii. Wires
iv. Trainer Meachine

CIRCUIT DIAGRAM:

figure: NOR Gate

equivalent,

figure: Implementing OR gate using NOR gate



equivalent,

figure: Implementing AND gate using NOR gate

equivalent,

figure: Implementing NOT gate using NOR gate.

TRUTH TABLE:

Truth table for NOR gate -

A	B	Output (Y)
0	0	1
0	1	0
1	0	0
1	1	0

Truth table for AND, OR & NOT using NOR gate:

A	A	Y
0	0	1
1	1	0

NOT

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

OR

A	B	X	Y	Z
0	0	1	1	0
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

AND

TIMING DIAGRAM:

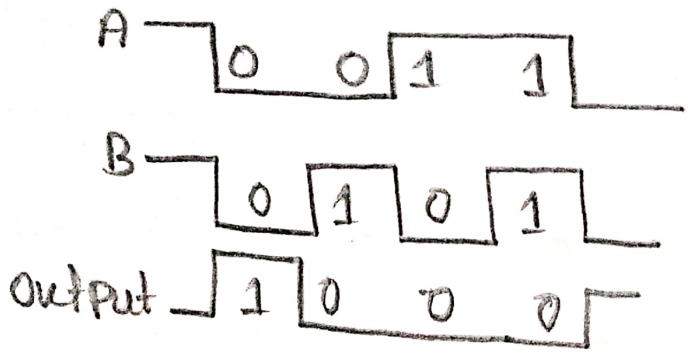


figure: timing diagram of a NOR gate.

Ic CONFIGURATION:

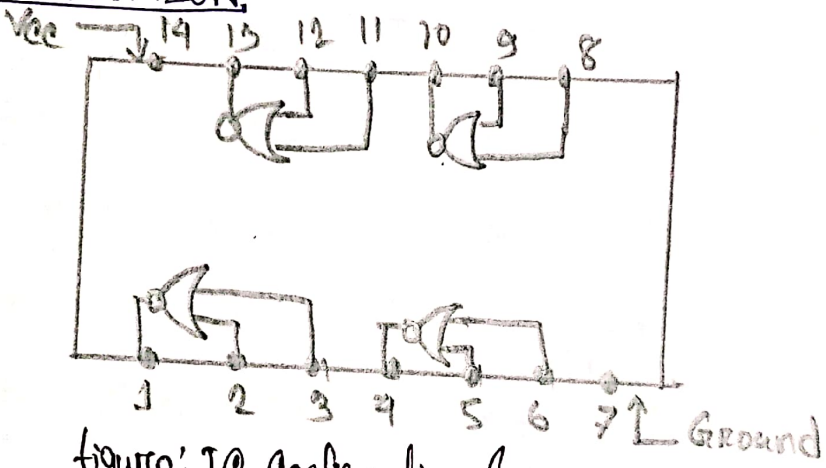


figure: Ic configuration of NOR gate.

INPUT OUTPUT:

The NOR gate is a digital logic that implements logical NOR, A HIGH output (1) results if both the inputs to the gate are low (0); if one or both inputs is HIGH (1), a low output (0) results. So we can say that NOR gat is Universal gate (proved).

NAME OF THE EXPERIMENT: DESIGNING A X-OR GATE USING BASIC GATES.

OBJECTIVE:

i. Properties of operation and basic identities of X-OR gate.
ii. Odd function and even function of an X-OR gate.
iii. Parity generation and checking of an X-OR gate.

DESCRIPTION OF THE PROBLEM:

An XOR gate or Exclusive OR gate is a digital logic gate with two or more inputs and one output that performs exclusive disjunction. The output of an X-OR gate is true only when exactly one of its input is true. If both of it's input is true then the output of X-OR gate is false. X-OR gates are used to implement binary addition in computers.

REQUIRED COMPONENTS:

i. 7404 IC of NOT gate
ii. 7408 IC of AND gate
iii. 7432 IC of OR gate.
iv. Breadboard, Digital trainer machine
v. Wires

CIRCUIT DIAGRAM:

figure: X-OR gate

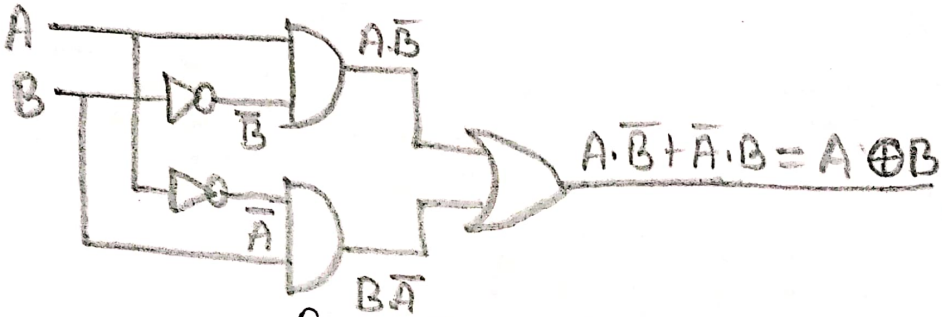


figure: Implementing XOR gate using AND, OR, NOT gate (Basic gates)

TRUTH TABLE:

A	B	$\bar{A}$	$\bar{B}$	$A\bar{B}$	$\bar{A}B$	$A \oplus B$
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	1	0	0	0	0	0

TIMING DIAGRAM:

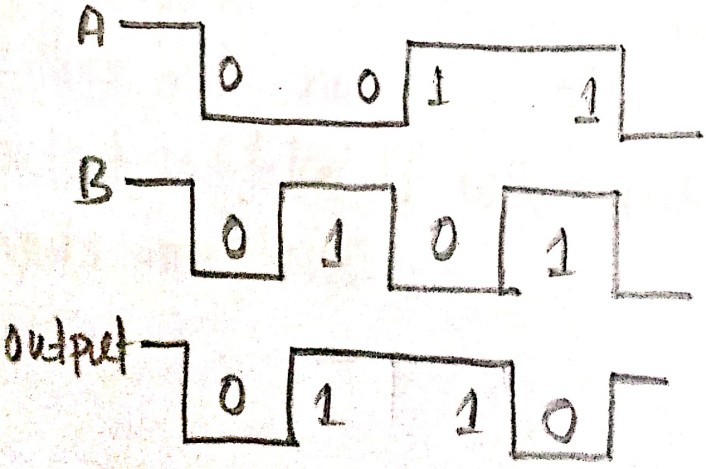


figure: timing diagram of X-OR gate.

Ic CONFIGURATION:

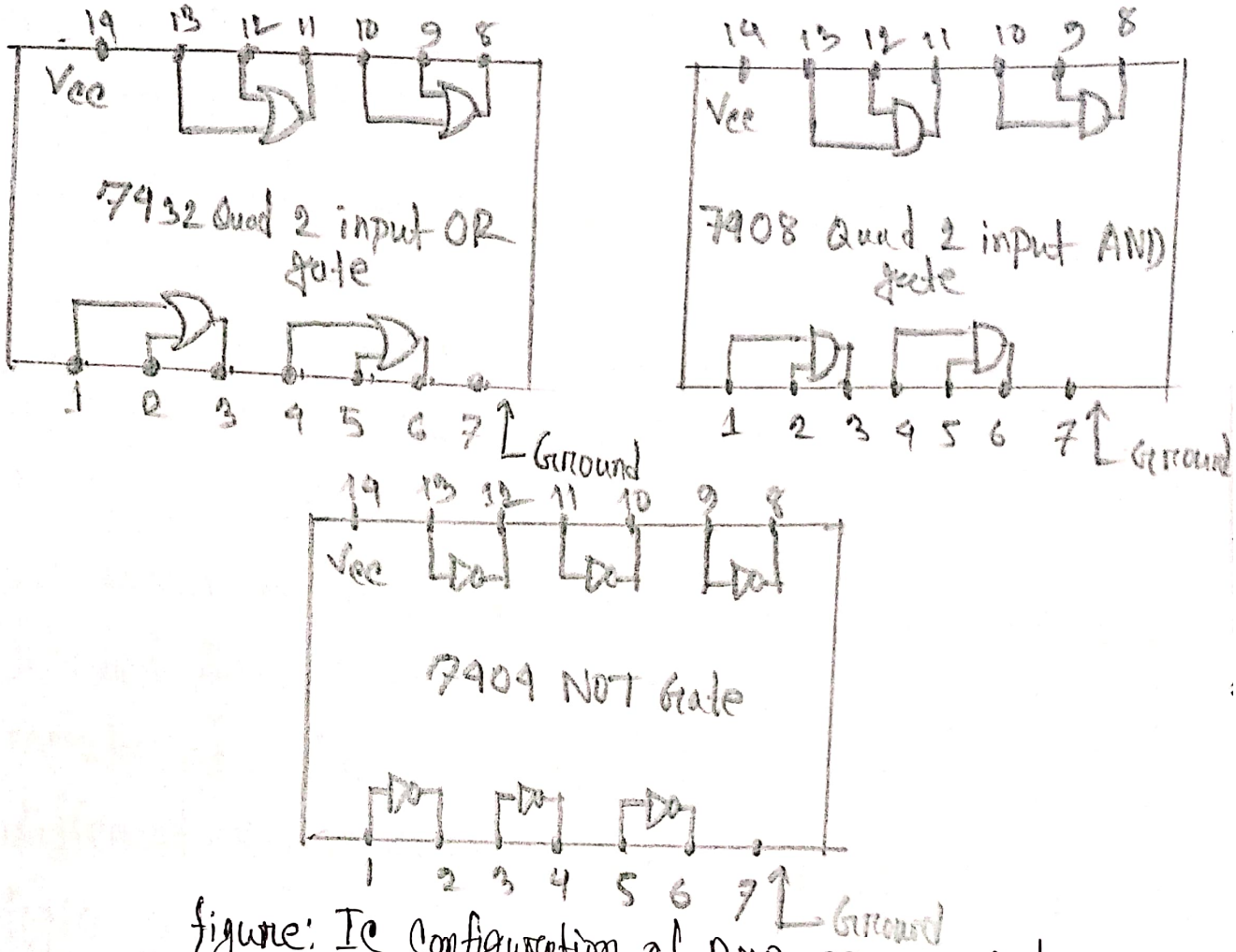


figure: Ic configuration of AND, OR, NOT gates.

INPUT OUTPUT:

The XOR (exclusive-OR) gate acts in the same way as the logical "either/or". The output is 'true' if either but not both of the inputs are 'true'. The output is 'false' if both inputs are "false" or if both inputs are "true".

NAME OF THE EXPERIMENT: DESIGNING A X-NOR GATE USING BASIC GATES.

OBJECTIVE:

i. Properties of operation and basic identities of X-NOR gate.
ii. Odd function and even function of an-X-NOR gate.
iii. To perform arithmatic and data processing operation.

DESCRIPTION OF THE PROBLEM:

The X-NOR (exclusive-NOR) gate is a combination. XOR gate followed by an inverter. It's output is "true" if the inputs are the same, and false if the inputs are different. X-NOR gate can be used to creates other logic gates.

REQUIRED COMPONENTS:

i. 7404 Ic of NOT gate
ii. 7408 Ic of AND gate
iii. 7432 Ic of OR gate
iv. Breadboard
v. Wires
vi. Digital trainer machine.

CIRCUIT DIAGRAM:

figure: X-NOR gate.

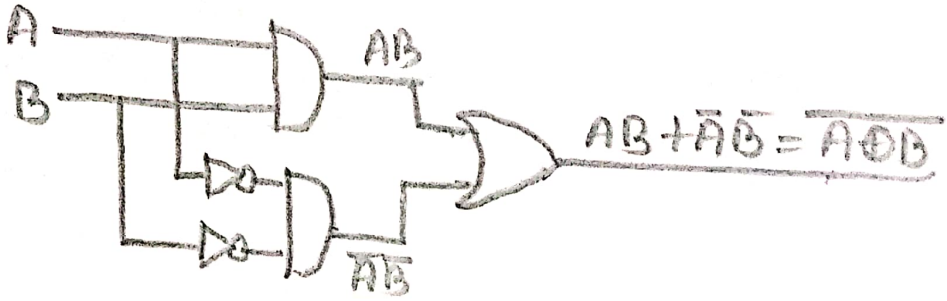


figure: Implementation of X-NOR gate using AND, OR, NOT gate.

TRUTH TABLE:

A	B	$\bar{A}$	$\bar{B}$	AB	$\bar{A}\bar{B}$	$\overline{A \oplus B}$
0	0	1	1	0	1	1
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	1	0	0	1	0	1

TIMING DIAGRAM:

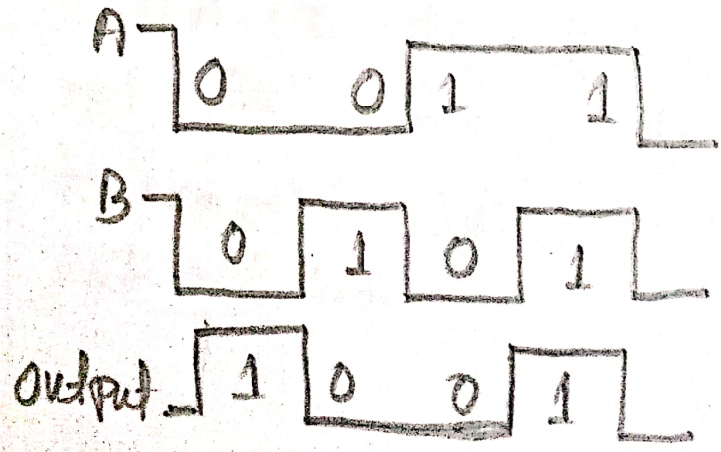


figure: timing diagram of X-NOR gate.

IC CONFIGURATION:

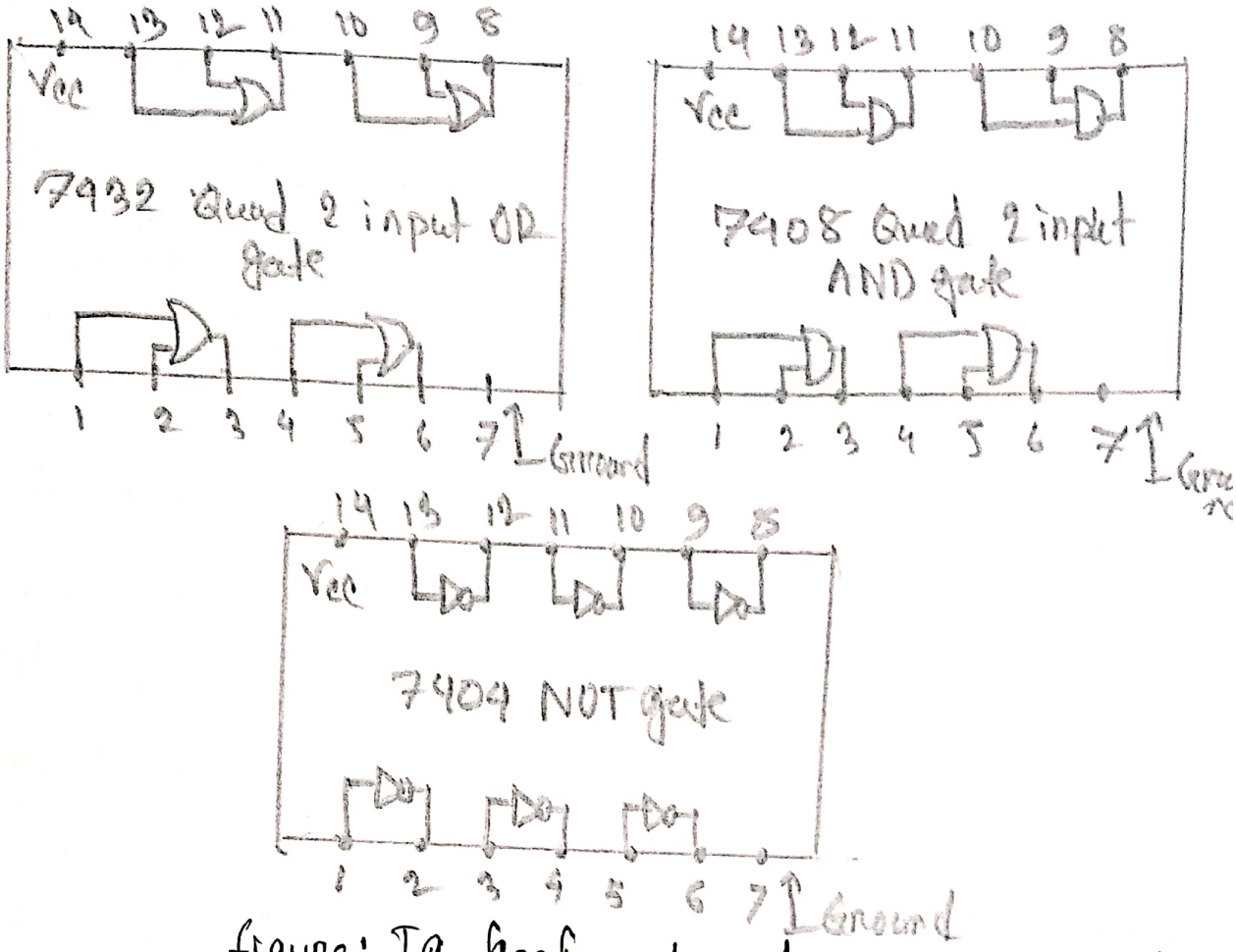


figure: Ic Configuration of AND, OR, NOT gate

INPUT OUTPUT:

If both the inputs are false (0/LOW) or both are true a false output results, XNOR represents the inequality function. i.e. the output is true if the inputs are not alike otherwise the output is false. A way to remember XNOR is "must have one or the other but not both.

NAME OF THE EXPERIMENT: DESIGNING A HALF ADDER.

OBJECTIVE:

i. To understand the properties of binary addition.
ii. To understand the half adder concept
iii. Use thruth table and boolean algebra theorems in simplyfing a circuit design.
iv. To implement half adder circuit using logic gate.

DESCRIPTION OF THE PROBLEM:

A half adder is a logical circuit that performs an addition operation on two binary digits. The half adder operation on two binary digits & produces a sum and a carry value which are both binary digits.

REQUIRED COMPONENTS:

i. 7486 X-OR gate
ii. 7408 AND gate
iii. Trainer Board
iv. Wires.

CIRCUIT DIAGRAM:

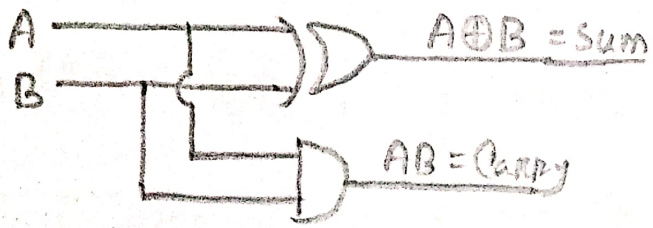


figure: Half adder logic diagram

TRUTH TABLE:

A	B	Sum=A⊕B	Carry=AB
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

TIMING DIAGRAM:

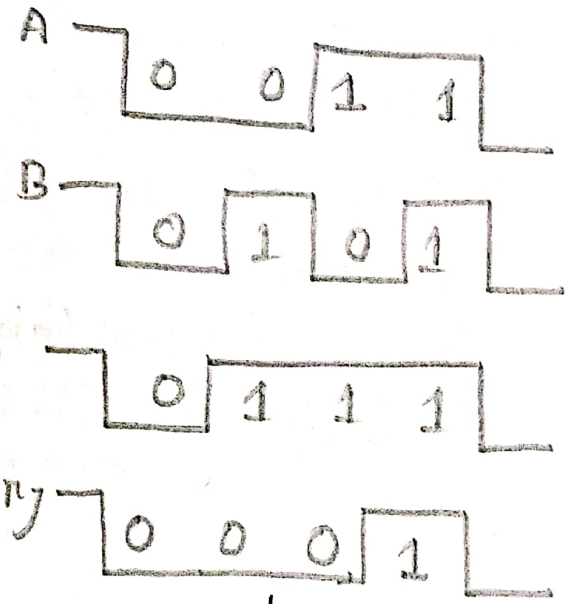


figure: timing diagram of half adder

IC CONFIGURATION:

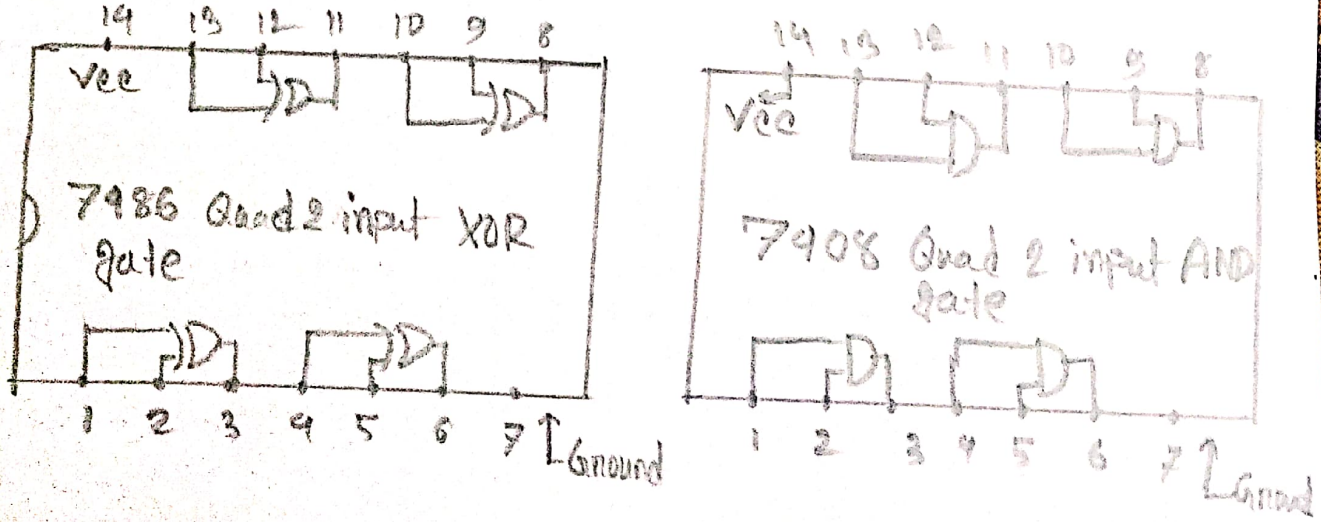


Figure: IC configuration of XOR gate & AND gate.

INPUT OUTPUT: A half adder is a type of adder, an electric circuit that performs the addition of numbers. The half adder is able to add two single binary digits and provid the output plus a carry value. It has two inputs, called A and B and two outputs Sum & Carry.

When A and B both are LOW (0) the output Sum and Carry are Low (0). When, A is low (0) and B is HIGH the Sum is HIGH (1) But Carry is LOW (0). Similarly when A is HIGH (1) and B is Low (0) the Sum is HIGH (1) and Carry is low (0). When A and B both are HIGH (1). the output Sum and Carry both are also HIGH (1).

NAME OF THE EXPERIMENT: DESINING A FULL ADDER.

OBJECTIVE:

i. To understand the principle of binary addition.
ii. To understand full adder concept.
iii. Using the truth table concept and bodean algebra theorems in simplifying a circuit design.
iv. To implement full adder circuit using logic gate.

DESCRIPTION OF THE PROBLEM:

A full adder is a logical circuit that performs an addition operation on two binary digits. The half adder produces a sum and a carry value which are both binary digits.

REQUIRED COMPONENTS:

1. 7486: Quad 2 input X-OR gate
ii. 7408: Quad 2 input AND gate
iii. 7432: Quad 2 input OR gate
iv. Digital trainer Board
v. Wires.

CIRCUIT DIAGRAM:

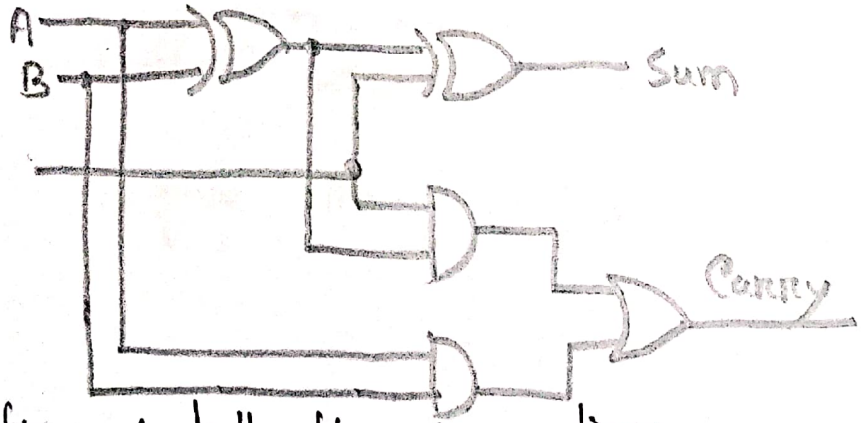


figure: full adder logic diagram.

TRUTH TABLE:

Input			Output	
A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	0	0	1	0
1	0	1	0	1
0	1	1	0	1
1	1	0	0	1
1	1	1	1	1

TIMING DIAGRAM:

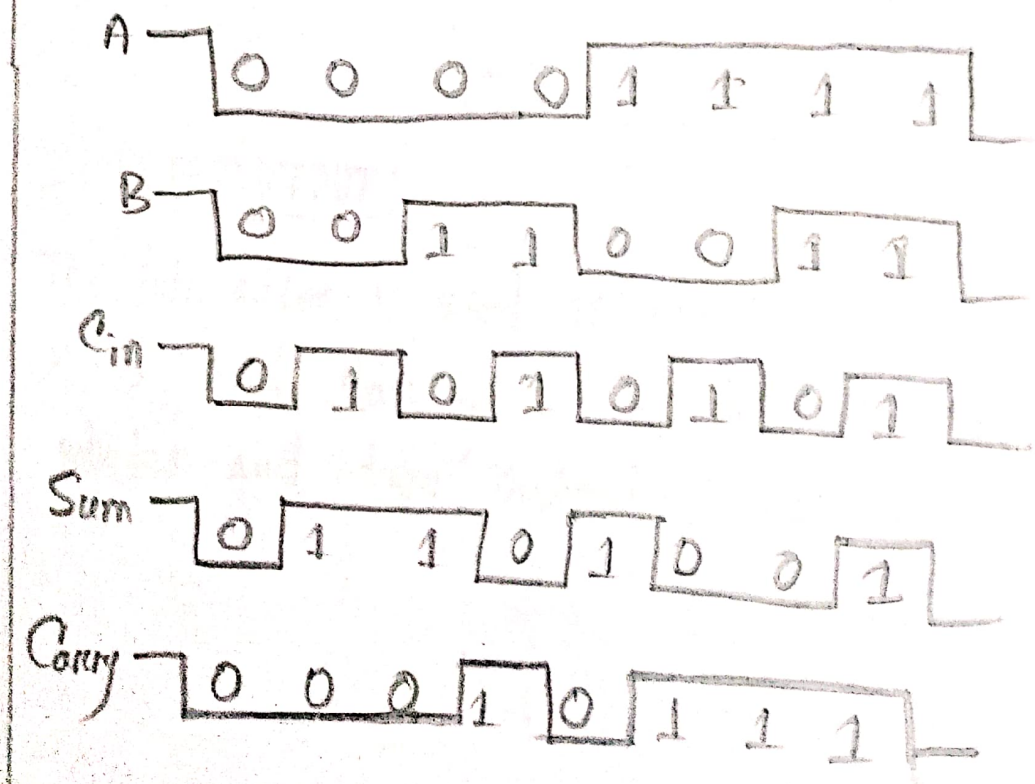


figure: Timming diagram of a full adder.

IC CONFIGURATION:

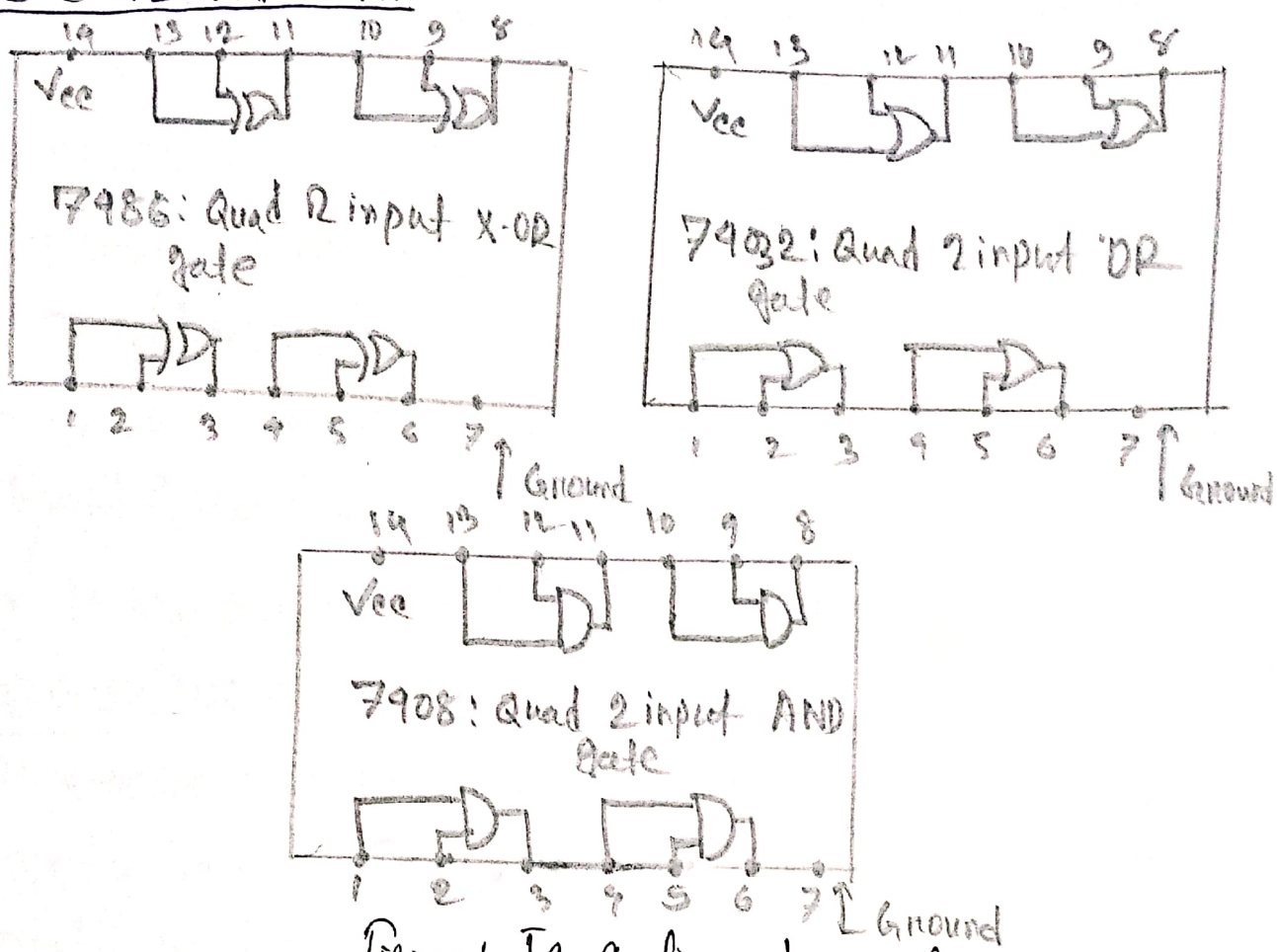


figure: IC configuration of X-OR, OR & AND gate.

INPUT-OUTPUT:

The full adder is used to add three 1-bit binary numbers A, B, and Carray C. The full adder has three input states and two output states. ie Sum & Carry.

NAME OF THE EXPERIMENT: DESIGNING AN ENCODER WITH OR GATE.

OBJECTIVE:

i. To study the operating principle of encoders.
ii. To implement the applications of encoders.
iii. Using truth table and boolean algebra theorems in symplifying the circuit design.
iv. To implement encoder using OR gate.

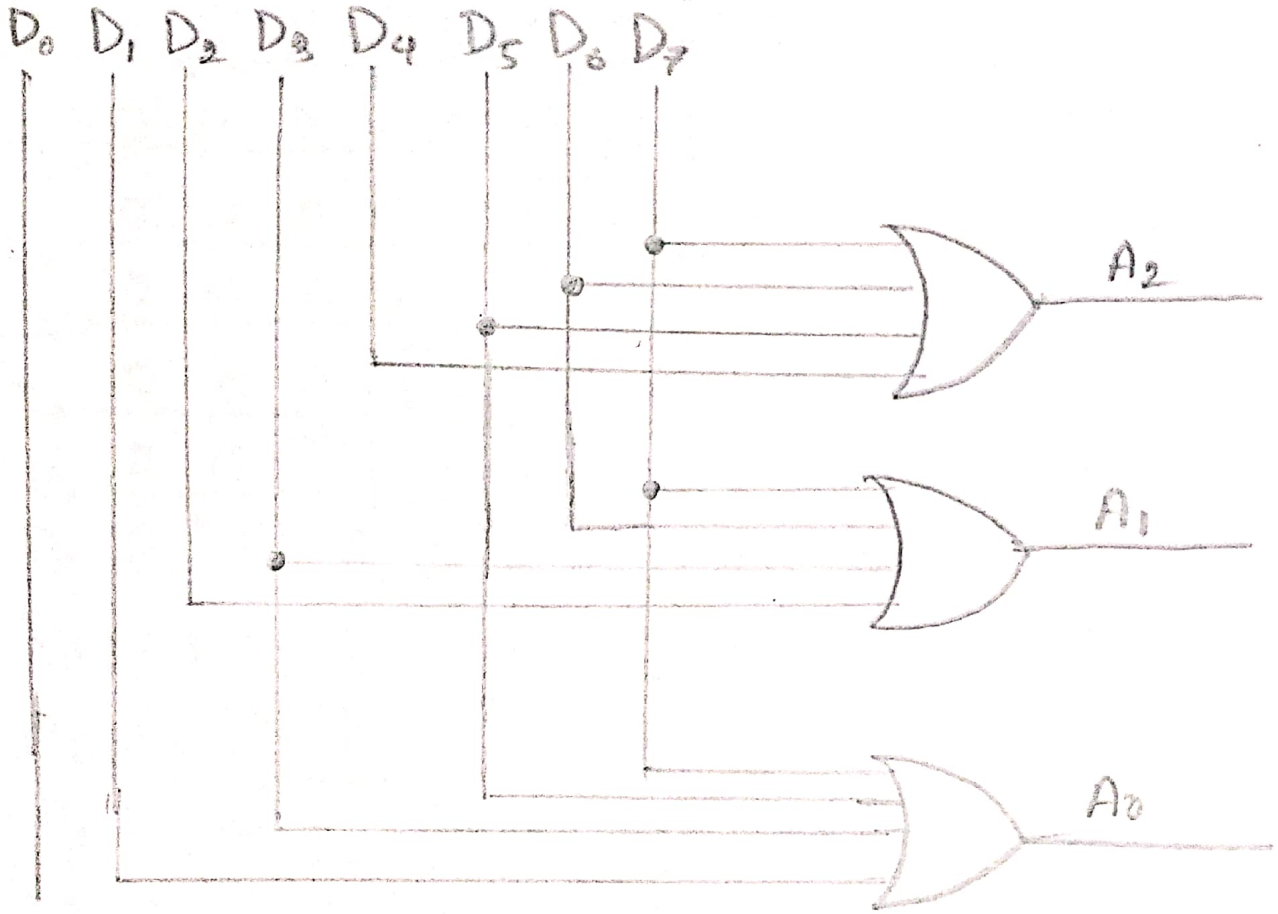
DESCRIPTION OF THE PROBLEM:

An encoder is a digital circuit that converts a set of binary inputs into a unique binary code. Encoders are commonly used in digital systems to convert a parallel set of inputs into a serial code. There are different types of encoders. In this problem we will use octal to binary encoders (8 to 3 Encoder). The 8 to 3 Encoder or octal to binary encoder consist of 8 inputs D_7 to D_0 and 3 outputs; A_2, A_1 and A_0. Each Input line corresponds to each octal digital and three output generate corresponding binary Code.

RIQUIRED COMPONENTS:

i. 7432: Quad 4 input OR gate (3 OR gates)
ii. Bread board / Digital trainer board.
iii. Wires

CIRCUIT DIAGRAM:



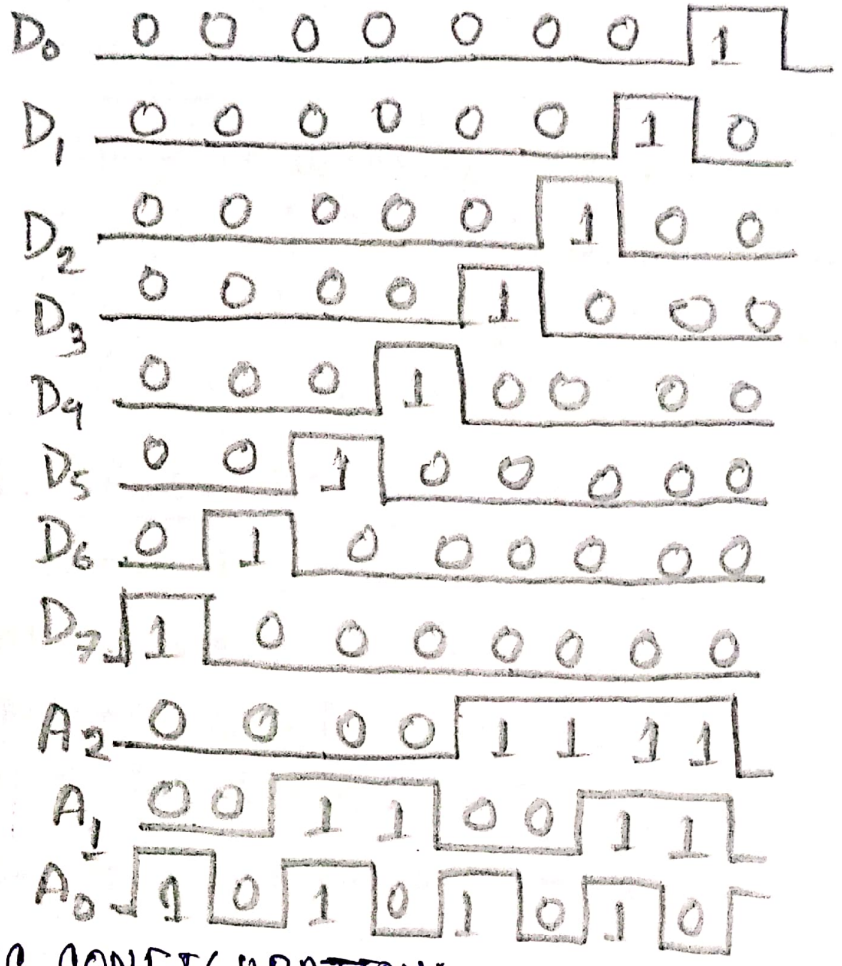
$$A_2 = D_7 + D_6 + D_5 + D_4$$
$$A_1 = D_7 + D_6 + D_3 + D_2$$
$$A_0 = D_7 + D_5 + D_3 + D_1$$

figure: Implementation of encoder using OR gates

INPUT OUTPUT: The truth table for 8 to 3 encoder is as follows:

Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

TIMING DIAGRAM:



IC CONFIGURATION:

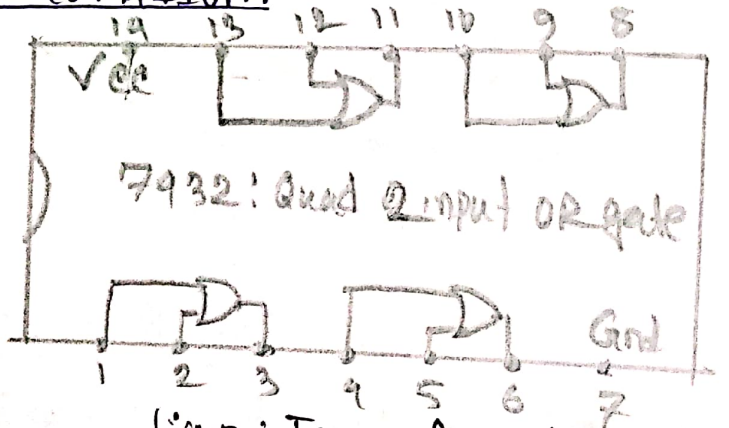


figure: Ic configuration of OR gate

INPUT OUTPUT:

The 8 to 3 line Encoder is also known as octal to Binary Encoder. In 8 to 3 line encoder, there is a total of 8 inputs. i.e. D_0, D_1, D_2, D_3, D_4, D_5, D_6, D_7 and there outputs. i.e A_0, A_1 & A_2.

NAME OF THE EXPERIMENT: DESINING A DECODER WITH NOT & AND GATE.

OBJECTIVE:

i. To study the operating principle of decoders.
ii. To implement the applications of decoders.
iii. Using truth table and boolean algebra theorems in simplifying the circuit design.

DESCRIPTION OF THE PROBLEM:

A decoder is a multiple input, multiple output logic circuit that converts coded inputs into coded outputs, where the input and output codes are different. The input code generally has fewer bits than the output code, and there is one-to-one mapping from input code words into output code words. In this problem, we will simplify 3 to 8 decoder through there are different kinds of decoders. The 3 to 8 decoder consists of 3 inputs: A, B, C & 8 outputs: D_7 to D_0

REQUIRED COMPONENTS:

i. 7411: Quad 3 input AND gate (4)
ii. 7404: NOT gate
iii. Digital trainer board
iv. Wires.

CIRCUIT DIAGRAM:

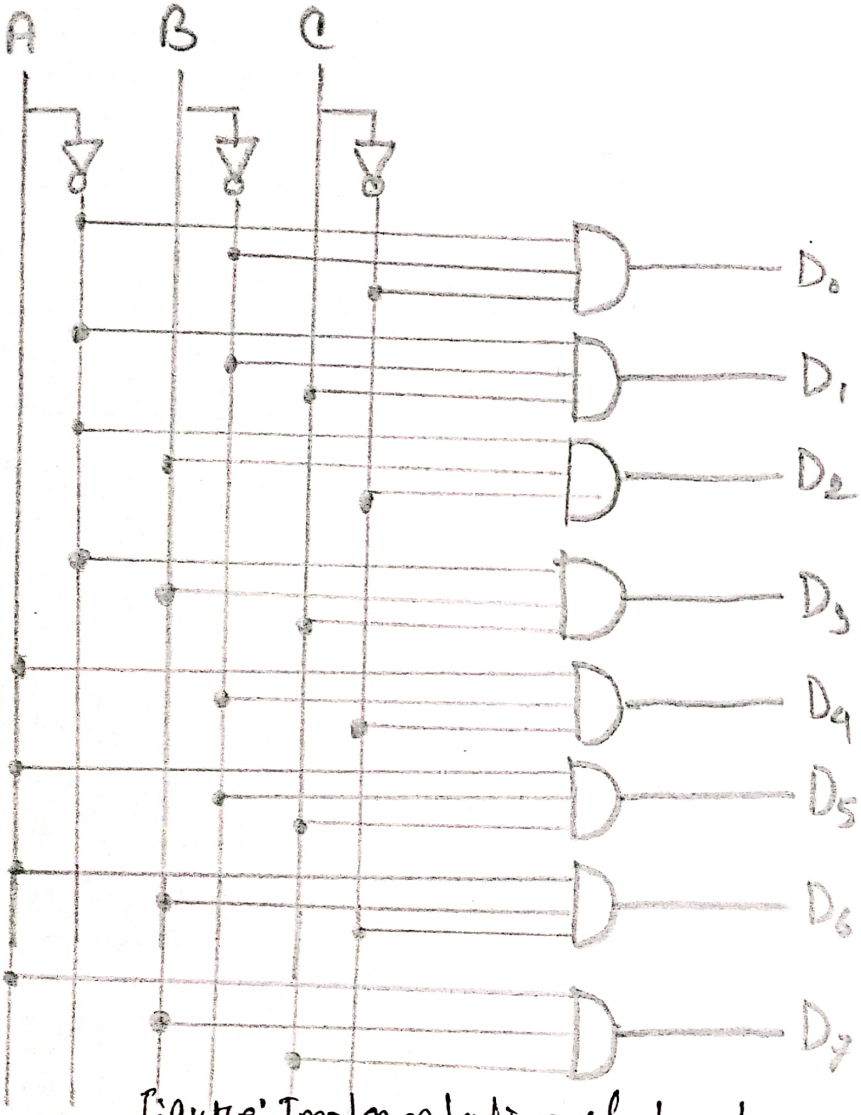


figure: Implementation of decoder

TRUTH TABLE: The truth table for the 3 to 8 decoder is as follows;

Inputs			Outputs							
A	B	C	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0