F. Y. B. Tech Academic Year 2021-22

Trimester:I **Subject:** Basics of Electrical and Electronics Engineering

Name: Krishnaraj Thadesar **Division 9**

Roll No:109054 **Batch I3**

Experiment No: 6

Name of the Experiment: Design and implementation of full adder using basic and universal gates.

Performed on: 4th February 2022

Submitted on: 4th February 2022

Aim: Design and implementation of Full Adder using basic and universal gates.

Prerequisite:

- Theory of digital logic circuits

Objectives:

- Identify pins of digital logic gates ICs
- Implement Full Adder circuit with basic and universal gates

Components and equipment required:

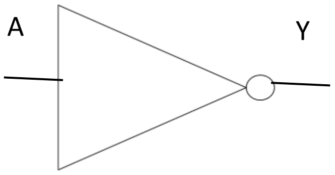
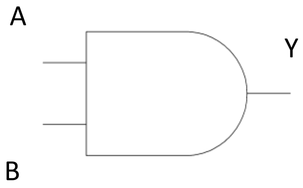
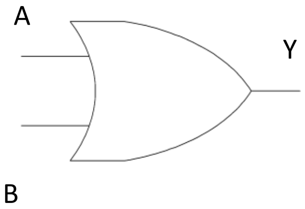
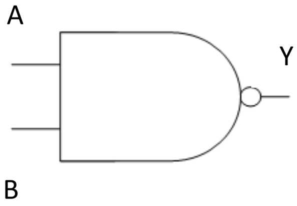
Power supply, digital board, digital logic gate ICs: OR (IC 7432), AND (IC 7408), NOT (IC 7404), NOR (IC 7402), NAND (IC 7400), EX-OR (IC 7486), connecting wires etc.

Theory:

Logic Gates:

Logic Gates are used to implement Boolean Logic. There are six types of basic logic gates that are used in digital systems. It is a device which has the ability to produce one output level with the combinations of input levels. Table 5.1 demonstrates description, symbol and truth table of the logic gates namely AND, OR, NOT, NAND, NOR and EXOR.

Table 5.1 Symbol and truth table of the logic gates

<table>
<tr><th>GATE</th><th>Description</th><th>Symbol</th><th>Truth Table</th></tr>
<tr><td><b>NOT</b></td><td><b>Inverts the given input A</b></td><td>
</td><td><table><tr><th>A</th><th>Y = A’</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td></tr>
<tr><td><b>AND</b></td><td><b>Logical AND of two inputs A and B</b></td><td>
</td><td><table><tr><th>A</th><th>B</th><th>Y = A*B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td><b>OR</b></td><td><b>Logical OR of two inputs A and B</b></td><td>
</td><td><table><tr><th>A</th><th>B</th><th>Y = A+B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td><b>NAND</b></td><td><b>Complement of logical AND of two inputs A and B</b></td><td>
</td><td><table><tr><th>A</th><th>B</th><th>Y = (A.B)’</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
</table>

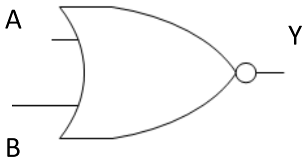
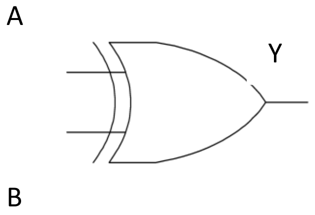
<table>
<tr><th>GATE</th><th>Description</th><th>Symbol</th><th>Truth Table</th></tr>
<tr><td><b>NOR</b></td><td><b>Complement of Logical OR of two inputs A and B</b></td><td> </td><td><table>
<tr><th>A</th><th>B</th><th>Y = (A+B)’
(A+B)’</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table></td></tr>
<tr><td><b>EXOR</b></td><td><b>Exclusive OR of two inputs A and B</b></td><td> </td><td><table>
<tr><th>A</th><th>B</th><th>Y = A□B</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table></td></tr>
</table>

Figure 5.1 indicates pin diagram of IC 7404 which is NOT gate. Fig. 5.2 indicates pin diagram of IC 7400 which is quad two input NAND gate. Pin diagrams of OR gate IC 7432, AND gate IC 7408 and EX-OR gate IC 7486 are similar to that of NAND gate IC 7400. Fig. 5.3 indicates pin diagram of IC 7404 hex inverter (NOT) gate.

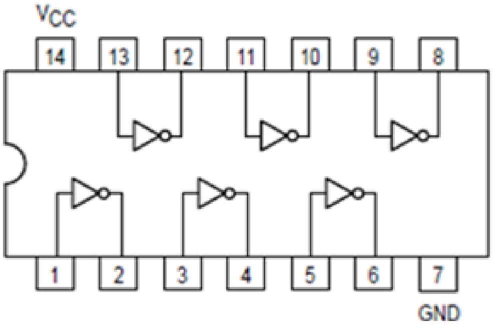


Fig 5.1 Pin diagram of IC 7404 NOT gate

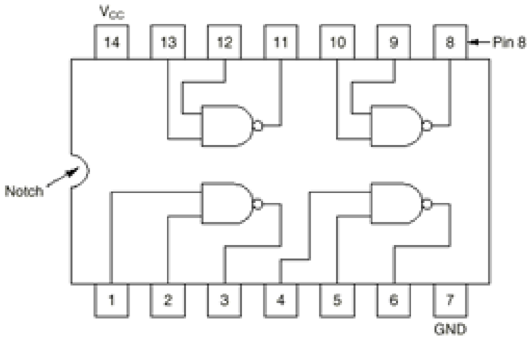


Fig 5.2 Pin diagram of IC 7400 NAND gate

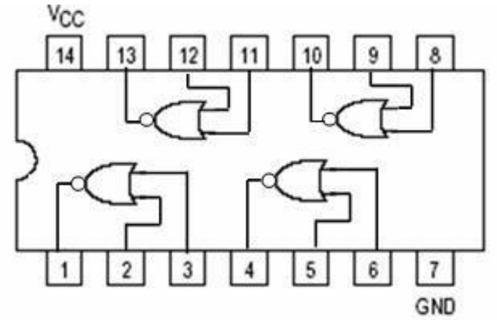


Fig 5.3 Pin diagram of IC 7402 NOR gate

Adder:

An Adder is a device that can add two binary digits. It is a type of digital circuit that performs the addition of two binary numbers. Adders are used in various applications like binary code decimal, address decoding, table index calculation etc.

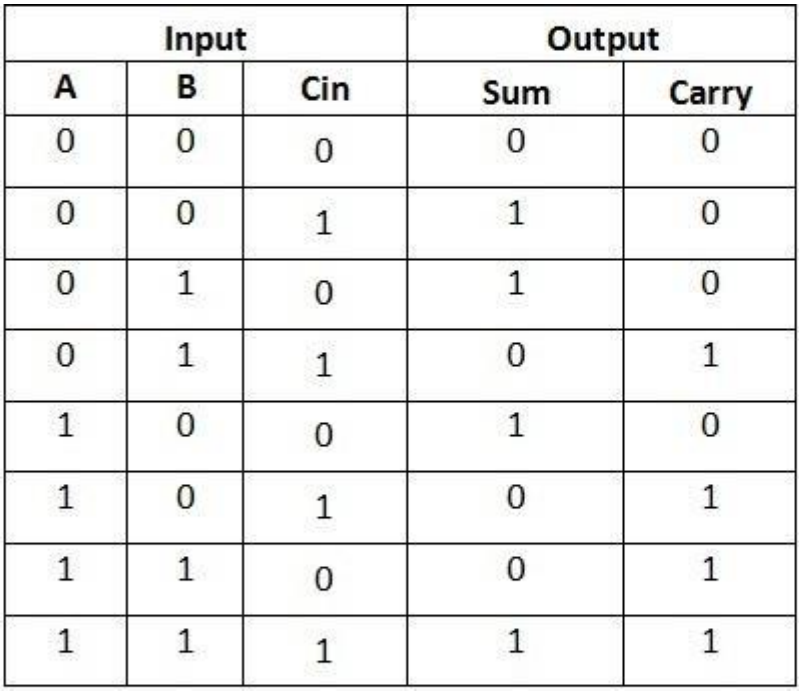
There are two types of adders, half adder and full adder. Half adder is used to add two binary digits and a full adder is used to add two binary digits with third binary digit which represents carry input. The detailed explanation of the full adder is given below.

Full Adder:

A full adder circuit is an arithmetic circuit block that can be used to add three bits to produce a SUM and a CARRY output. Such a building block is needed in order to add binary numbers with a large number of bits. The full adder circuit overcomes the limitation of the half-adder, which can be used to add two bits only

There are three inputs and two outputs to a full adder as shown in Fig. 5.4. Inputs are named as A, B and Cin, outputs are named as Sum (S) and Carry (C). Truth table for the full adder is shown in Table 5.2.

Table 5.2 Truth Table of Full adder



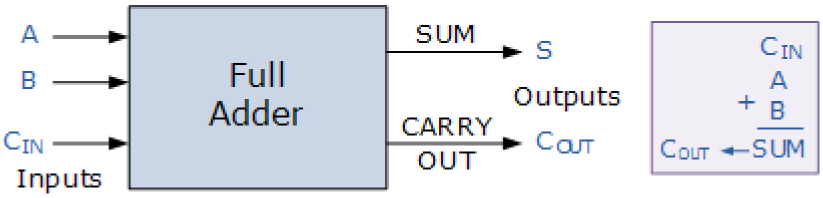
Input			Output	
A	B	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Using Karnaugh-maps (K-map), Boolean expressions derived for sum and carry are given by Eq. 5.1 & Eq. 5.2.

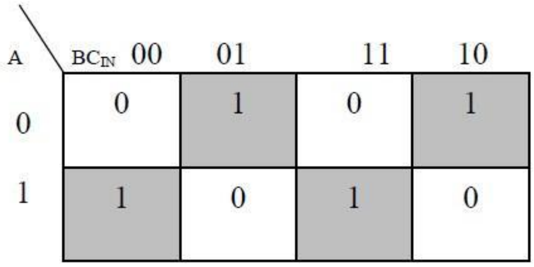
SUM = A¯B¯Cin + A¯BC¯in +A B¯C¯in + A B Cin = A ⊕ B ⊕ Cin (5.1)

CARRY-OUT = A B + A Cin + B Cin = A.B + Cin (A ⊕ B) (5.2)

Full adder schematic drawn, based on the sum and carry expressions is shown in Fig. 5.5.

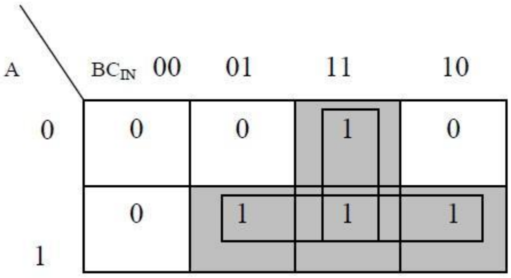


Sum = S= A̅ B̅ Cin + A̅ B C̅ in +A B̅ C̅ in + A B Cin = A ⊕ B ⊕ Cin



A \ BC_{IN}	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Carry Out = C $_{\textbf{out =}}$ A B + A Cin + B Cin



A \ BC_{IN}	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Fig: 5.4 Full adder block, K-map for sum and carry

After optimization we get

SUM = A XOR B XOR Cin = A $\oplus$ B $\oplus$ Cin

CARRY-OUT = A AND B OR Cin (A XOR B) = A.B + Cin (A $\oplus$ B)

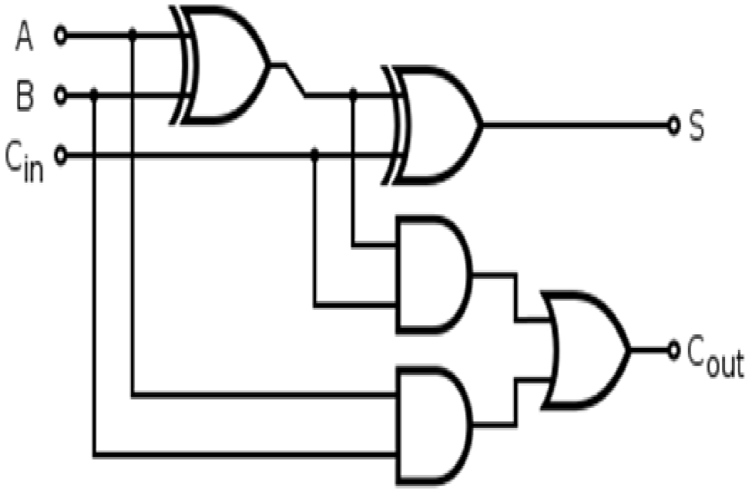


Fig. 5.5 Full adder Schematic

Procedure:

1. Identify the ICs and verify the truth table of all given ICs as per the entries in Table 5.1.
2. Build full adder circuit as shown in Fig. 5.5 on the digital board.
3. Make the connections and apply the voltage.
4. Verify the results of half adder as per the truth table entries in Table 5.2.

Conclusion:

The Full Adder circuit was made in Tinkercad and its function and working were understood in detail. The working of all basic as well as universal logic gates were also understood. ICs stopped functioning above a certain Vcc.

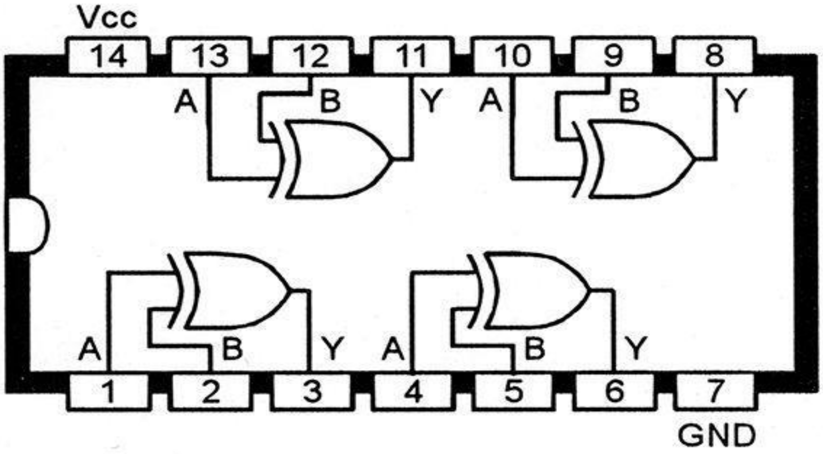
Post Lab Questions:

1. Implement the basic logic gates using Universal gates.
2. Implement NOR gate using NAND gates, NAND gate using NOR gates.
3. What is XOR gate?
4. Implement full adder using NAND gates only. .
5. State and prove Dorgan’s theorem.
6. Explain the operation of following gates:
   a) NOT b) AND c) OR d) NAND e) NOR

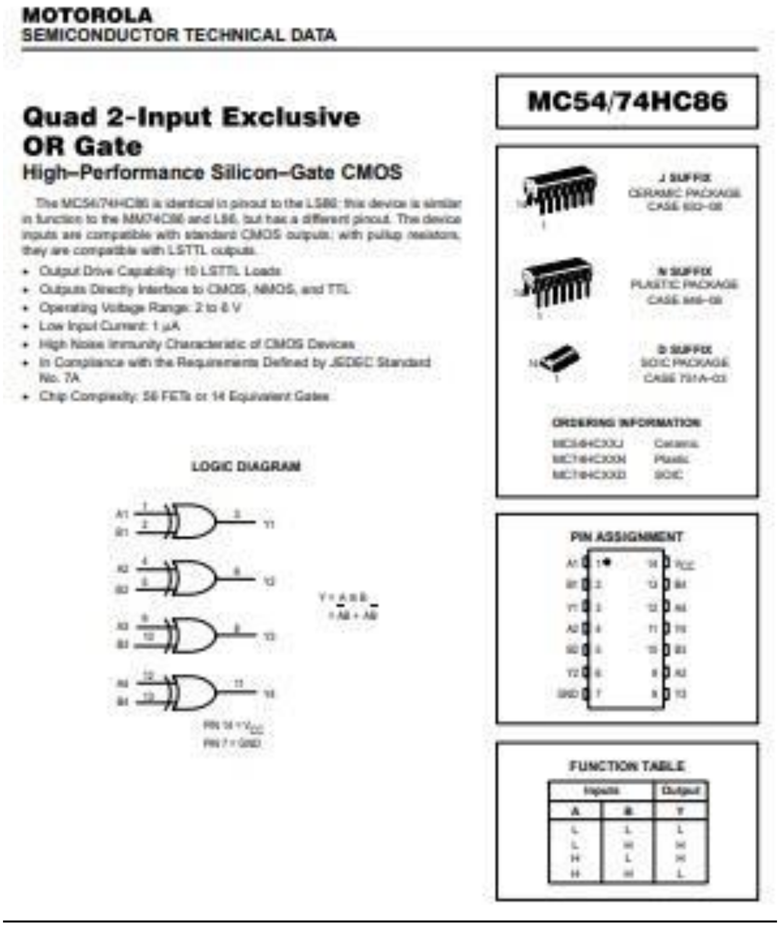
Additional links for more information:

- http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/
- https://www.electronicshub.org/half-adder-and-full-adder-circuits/

EX- OR Gate(IC 7486)



VCC: 5 V



MOTOROLA
SEMICONDUCTOR TECHNICAL DATA

MC54/74HC86

Quad 2-Input Exclusive OR Gate
High–Performance Silicon–Gate CMOS

The MC54/74HC86 is identical in pinout to the LS86: this device is similar in function to the MM74C86 and L86, but has a different pinout. The device inputs are compatible with standard CMOS outputs; with pullup resistors, they are compatible with LSTTL outputs.

- Output Drive Capability: 10 LSTTL Loads
- Outputs Directly Interface to CMOS, NMOS, and TTL
- Operating Voltage Range: 2 to 6 V
- Low Input Current: 1 μA
- High Noise Immunity Characteristic of CMOS Devices
- In Compliance with the Requirements Defined by JEDEC Standard No. 7A
- Chip Complexity: 56 FETs or 14 Equivalent Gates

J SUFFIX
CERAMIC PACKAGE
CASE 632–08

N SUFFIX
PLASTIC PACKAGE
CASE 646–06

D SUFFIX
SOIC PACKAGE
CASE 751A–03

ORDERING INFORMATION

MC54HCXXJ	Ceramic
MC74HCXXN	Plastic
MC74HCXXD	SOIC

LOGIC DIAGRAM

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

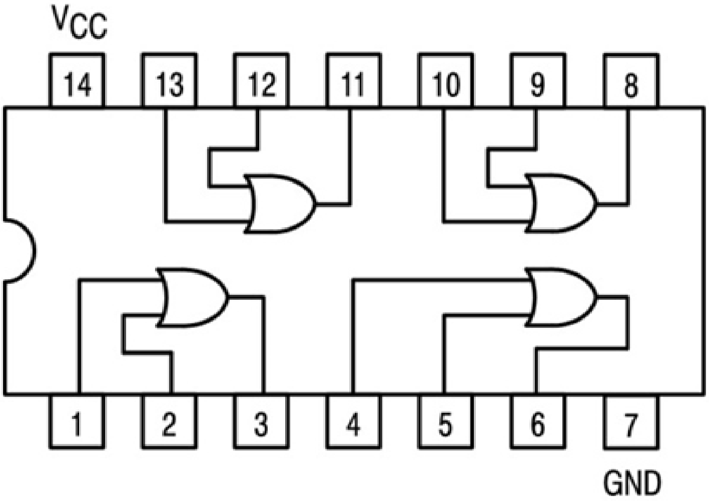
PIN 14 = V_{CC}
PIN 7 = GND

PIN ASSIGNMENT

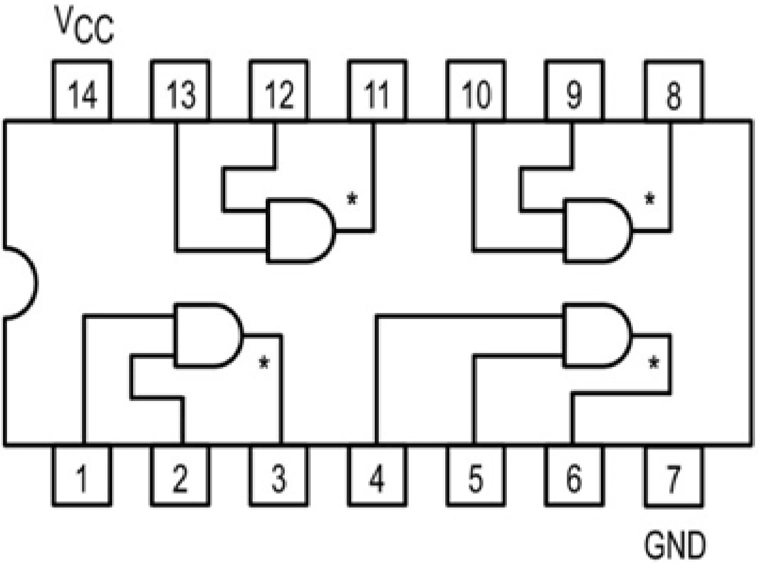
FUNCTION TABLE

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L

OR Gate(IC 7432)



AND Gate(IC 7408)



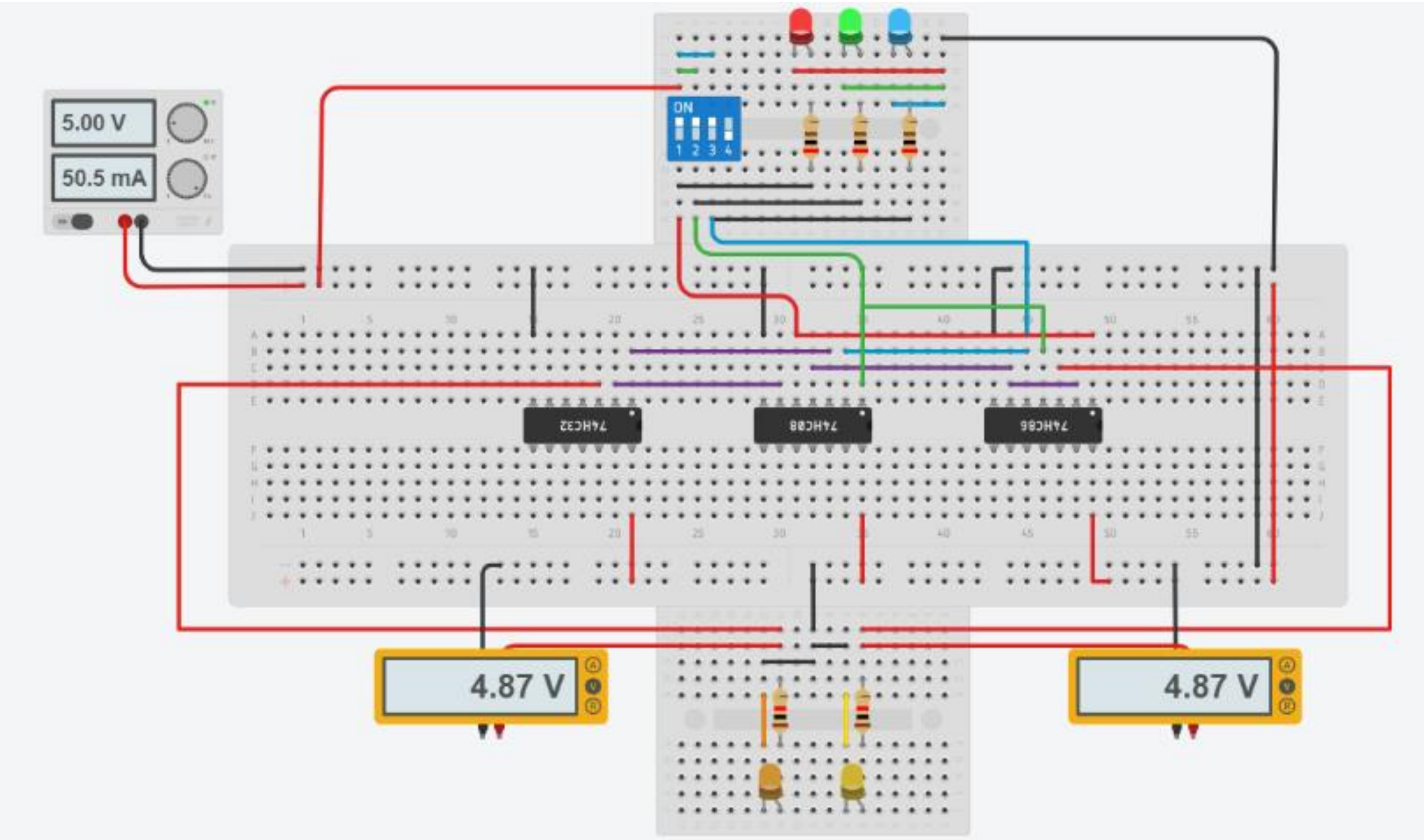
Observation Table

Inputs			Outputs			
Red	Green	Blue	Orange	Multimeter Voltage Reading	Yellow	Multimeter Voltage Reading
0	0	0	0	0 V	0	0 V
0	0	1	1	4.87 V	0	0 V
0	1	0	1	4.87 V	0	0 V
0	1	1	0	0 V	1	4.87 V
1	0	0	1	4.87 V	0	0 V
1	0	1	0	0 V	1	4.87 V
1	1	0	0	0 V	1	4.87 V
1	1	1	1	4.87 V	1	4.87 V

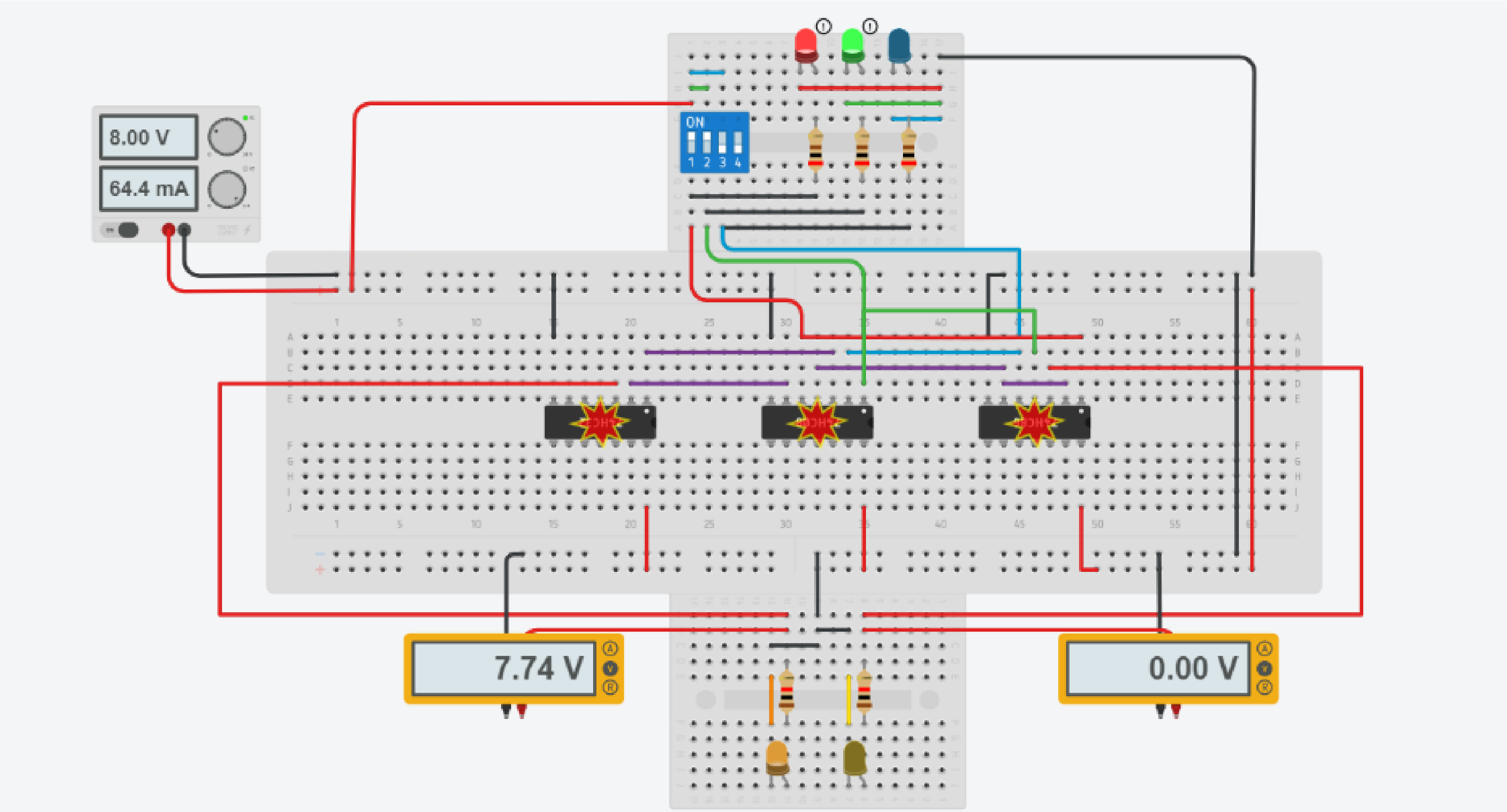
Component List

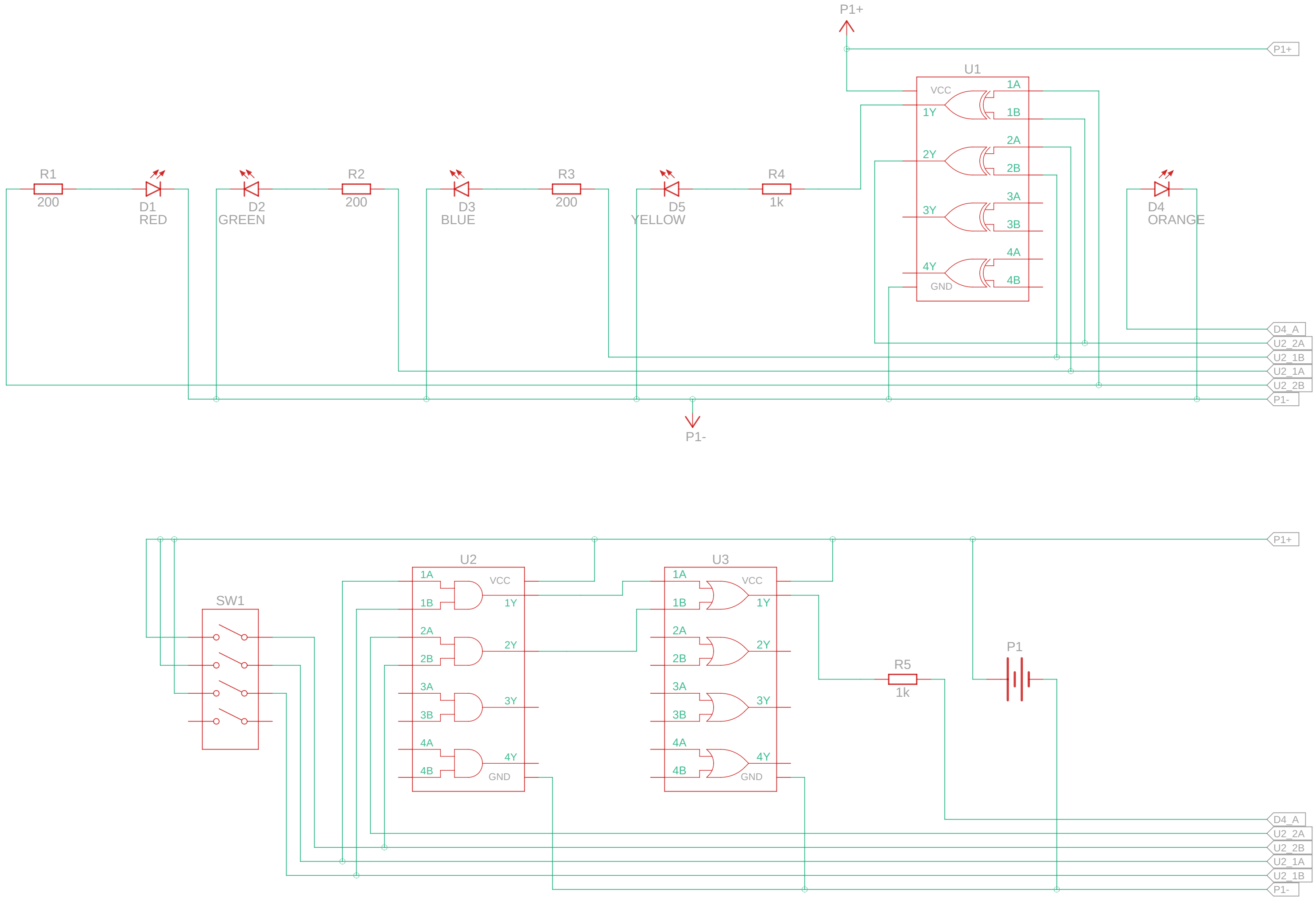
Name	Quantity	Component
P1	1	8 , 5 Power Supply
SW1	1	DIP Switch SPST x 4
D1	1	Red LED
D2	1	Green LED
D3	1	Blue LED
R1, R2, R3	3	200 Ω Resistor
D4	1	Orange LED
D5	1	Yellow LED
R4, R5	2	1 kΩ Resistor
U1	1	Quad XOR gate
U2	1	Quad AND gate
U3	1	Quad OR gate
Meter1, Meter2	2	Voltage Multimeter

Tinkercad Circuit for Full Adder



Full Adder when V_{cc} is Too High





Made with Tinkercad®

Title: E6. Full Adder

Date: 04/02/2022, 10:37:34

Sheet: 1/1

Krishnaraj PT.
~~x~~ 109054 · (I3)

4|2|2022

BEEE EXPERIMENT-6

Full Adder

(*) Post lab questions

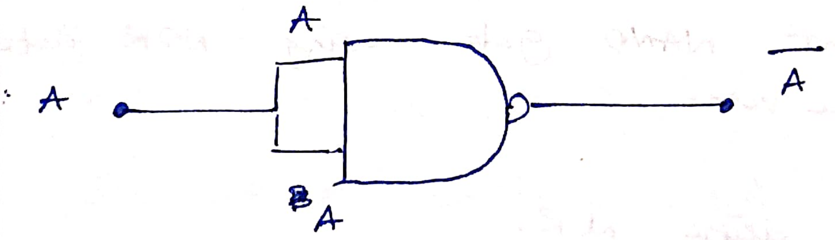
(Q-1.) Implement the basic logic gates using universal gates.

→ Universal gates = NAND and NOR.

Basic gates = AND, OR, NOT.

(A) NOT Gate :

Inp	Out
A	Y
0	1
1	0



NAND gate.

Fig.2. logical NOT

(B) AND gate.

Inputs		Out
A	B	Y
1	0	0
0	1	0
1	1	1
0	0	0

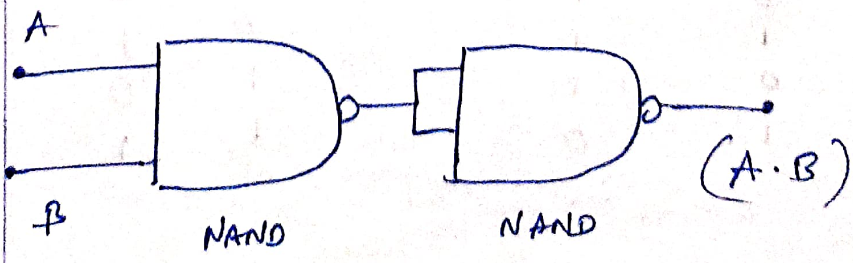
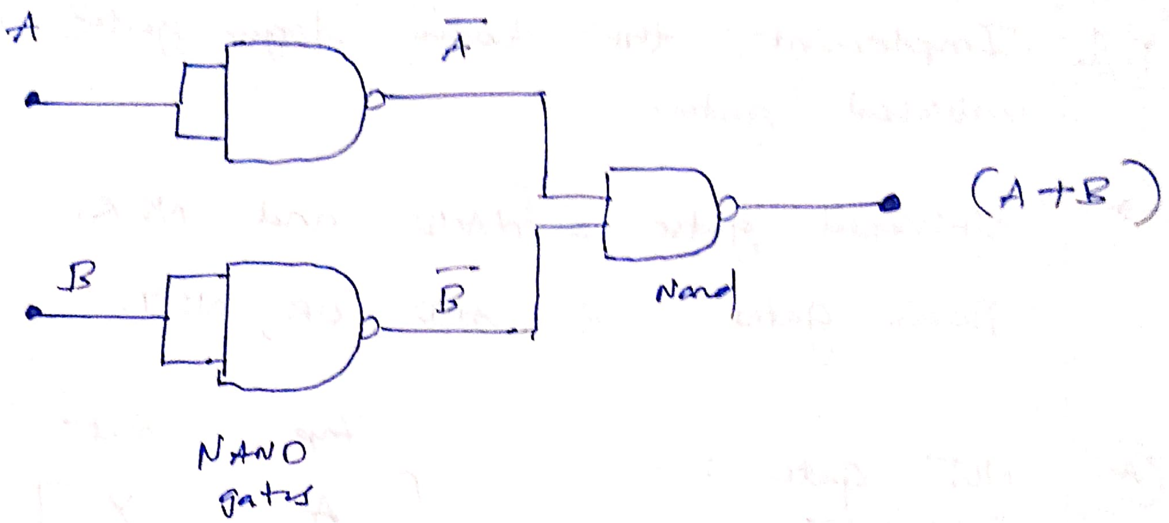


Fig. 2 - logical AND.

Truth Table

③ OR gate

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



(Fig. (3) → Logical or.)

Q②. Implement NAND gate using NOR gate and vice versa.

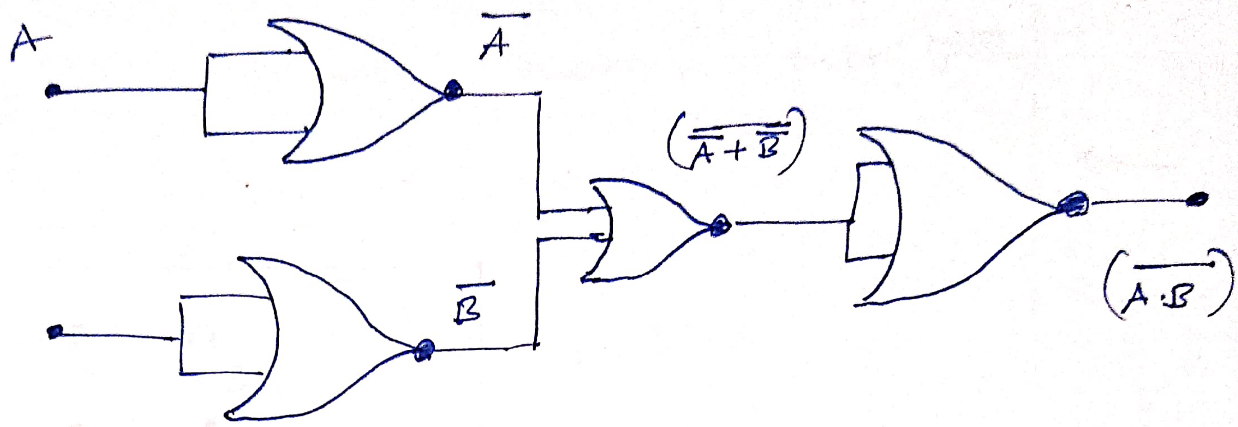
→ NAND using NOR.

Nand Truth Table

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

NOR Truth Table

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



→ Fig - 4, NAND using NOR.

④ NOR using ~~Nand~~ NAND gates

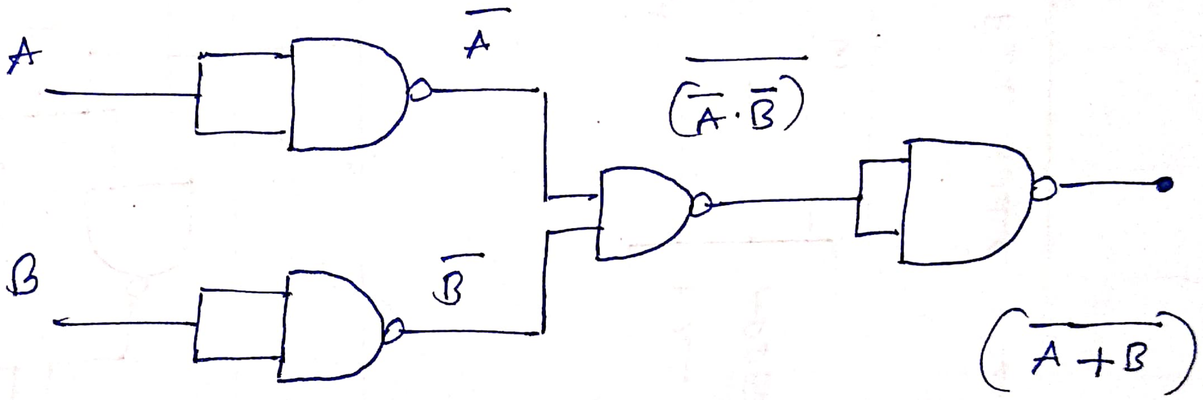
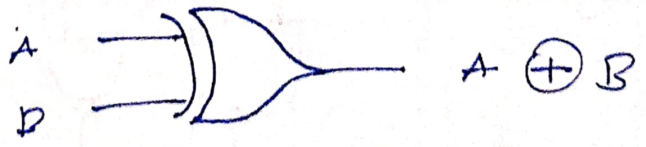


Fig. 5 NOR using NAND

Q. ③ What is XOR gate ?,

→ EX-OR or Exclusive OR ~~as~~ are special gates. They are constructed by combining other logic gates.



Truth Table

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

is equivalent to

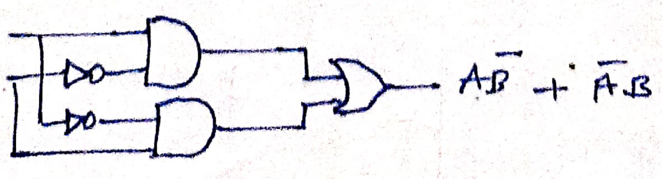


Fig. 6 : XOR Gates

Q(4). Implement full adder using NAND gates only.

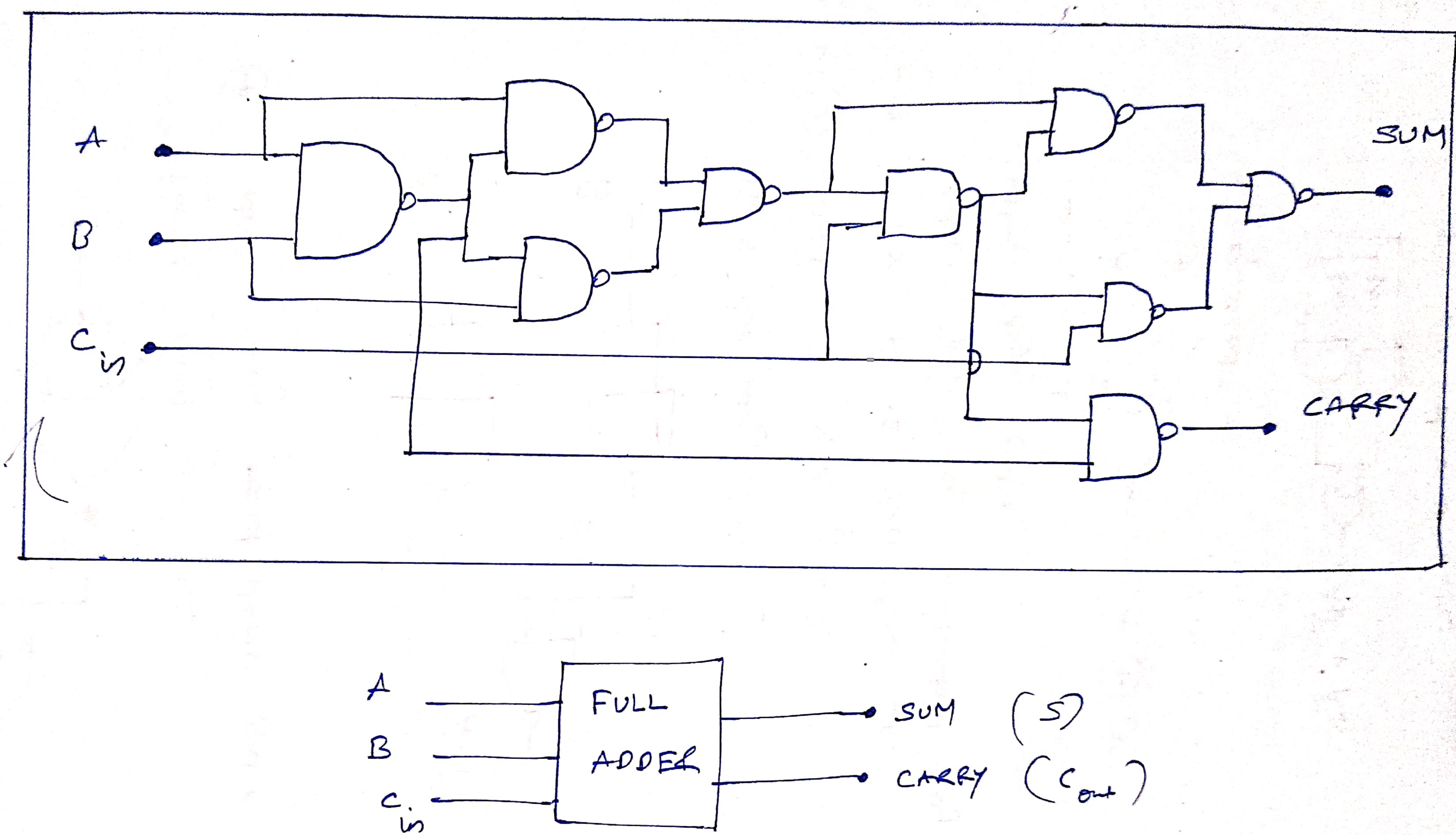


Fig : 7. Full Adder using NAND Gates only.

Q (5). State and Prove De' Morgan's Thorem.

→ De Morgan's Thorem states that

(1) $\overline{A \cdot B} = \overline{A} + \overline{B}$

(2) $\overline{A + B} = \overline{A} \cdot \overline{B}$

→ Truth Table for (1)

(=)

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

→ Truth Table for (2)

(=)

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

as Truth Tables Matchd, they are ~~eq~~ equal, RHS = LHS and De Morgan's Thorem is proven.

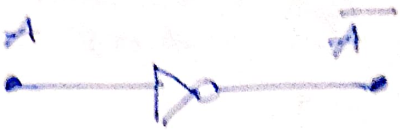
Q.6. Explain operation of the following gates.

(1) NOT :

It is used to invert the input signal.

~~A~~ ~~0~~

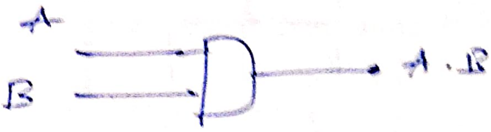
B	Y
1	0
0	1



(2) AND :

It is logical Multiplication. It gives output (A . B) only when A and B are on.

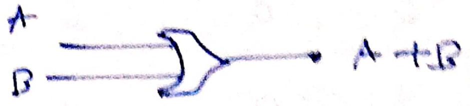
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



(3) OR : logical addition : It gives ON only if A or B are ON.

A + B .

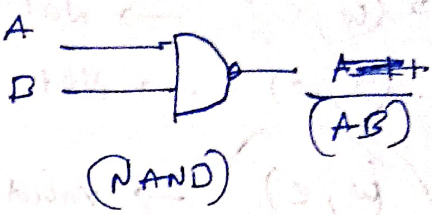
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



(4) NAND gate : These are universal gates

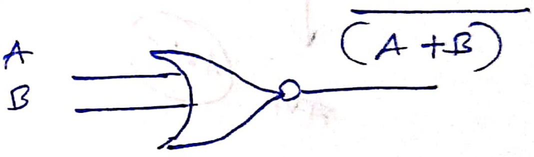
~~They~~ Nand gates is the inverted output of AND gate. → $\overline{(A \cdot B)}$

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



(5) NOR gate : It is the inverted output of OR gate → $\overline{(A+B)}$

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


A
B
$\overline{(A+B)}$