

Lab Report: 2

Name: Atharv Bhatt

Roll number: 2023101089

Group number: 3

Experiment 2: Basic Logic Gates

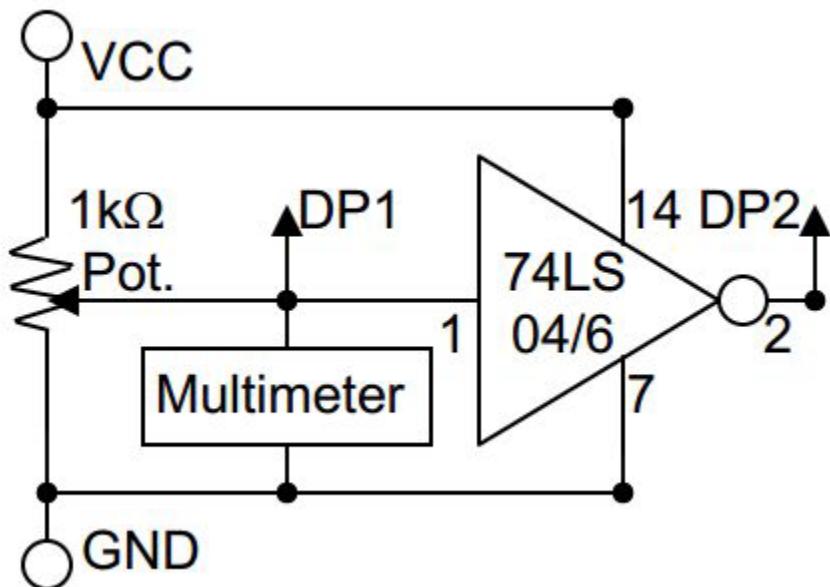
Part A: Logic Levels

Aim/Objective: The objective of this experiment is to understand binary logic levels within the specified 0-5V range and compare the measured voltages with the standard binary logic level specifications.

Electronic Components Used

1. Breadboard
2. Potentiometer
3. Multimeter
4. TTL 74LSxx Series Logic IC (as shown in Fig. 2.1)
5. $0.1\mu F$ Ceramic Capacitor
6. Power Supply

Reference Circuit



Procedure

1. Set up the circuit as shown in Fig. 2.1 on the breadboard.
2. Rotate the potentiometer shaft to one end, ensuring that the multimeter initially reads 0V. This configuration causes LG1 (Green LED 1) and LR2 (Red LED 2) to illuminate.
3. Gradually rotate the potentiometer shaft towards the other end while tabulating the transitions in LG1, LR2, LG2 (Green LED 2), and LR1 (Red LED 1). Simultaneously, record the corresponding multimeter readings.
4. Repeat this process until LR1 and LG2 are fully illuminated.
5. Compare the recorded voltages with the standard binary logic level specifications within the 0-5V range:
 - Low-Level Output Voltage (VOL): $0 \leq VOL \leq 0.4V$
 - Low-Level Input Voltage (VIL): $0 \leq VIL \leq 0.8V$
 - High-Level Input Voltage (VIH): $2.0 \leq VIH \leq 5.0V$
 - High-Level Output Voltage (VOH): $2.4 \leq VOH \leq 5.0V$

Observations

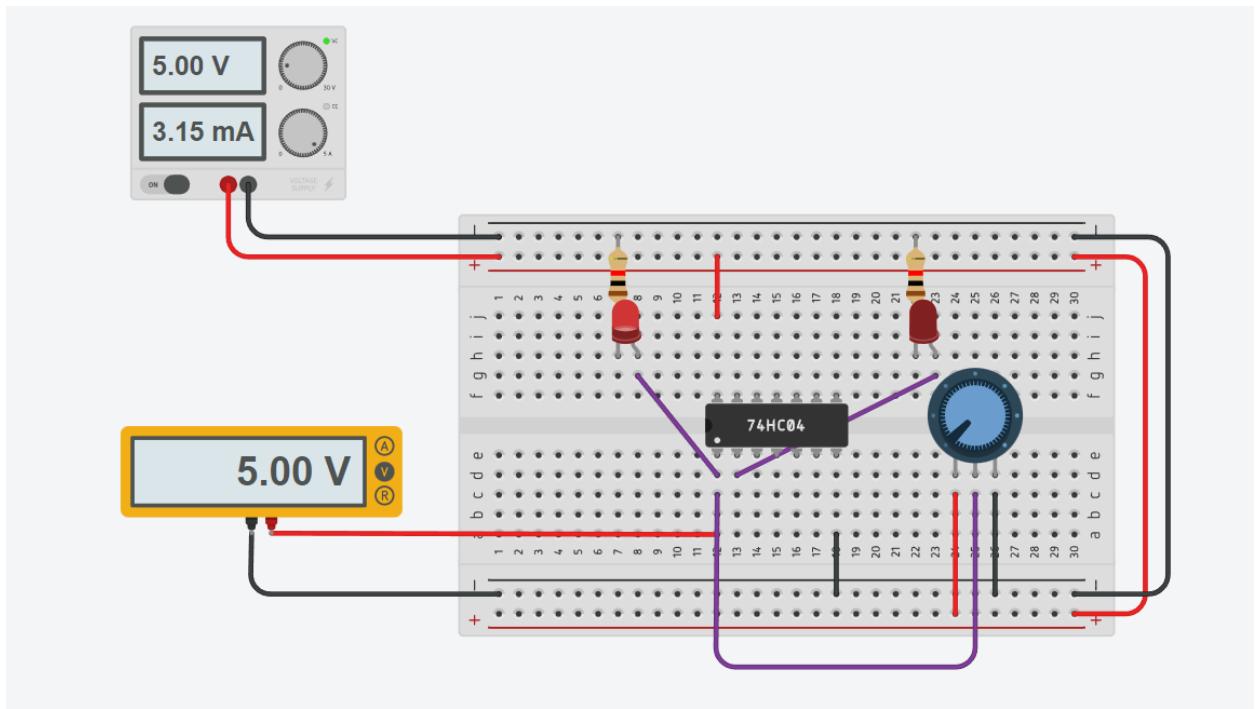
Voltage Readings(V)

Voltage Type	Theoretical Value (Volts)		Actual Value (Volts)	
	Lower	Upper	Lower	Higher
V _{OL}	0.00	0.40	0.00	0.35
V _{OH}	2.40	5.00	1.80	4.6
V _{IL}	0.00	0.80	0.00	0.76
V _{IH}	2.00	5.00	1.80	5.00

Note: Both the LEDs are OFF at the approximate range of 1.4 volts - 1.7 volts.

Tinkercad

https://www.tinkercad.com/things/8UJkvuvSWKh-logic-levels/editel?sharecode=5FOnaRi0t6NPsu6REHI-ZMe1gc0oGsfWDKLxHRoek_A



Conclusion

In this experiment, we observe and tabulate the transitions of LED states and corresponding multimeter readings as we adjust the potentiometer shaft. These transitions are crucial in understanding binary logic levels within the 0-5V range. Upon comparing the recorded voltages with the standard binary logic level specifications, we find that the circuit is same as the expected logic level ranges.

Part B: Gate Identification

Aim/Objective

The objective of this experiment is to identify the logic functions of different ICs, each consisting of four AND / OR / NAND / NOR / XOR gates. We will identify this through a step-by-step process involving connecting the ICs, applying binary input values, and tabulating the gate outputs to create truth tables.

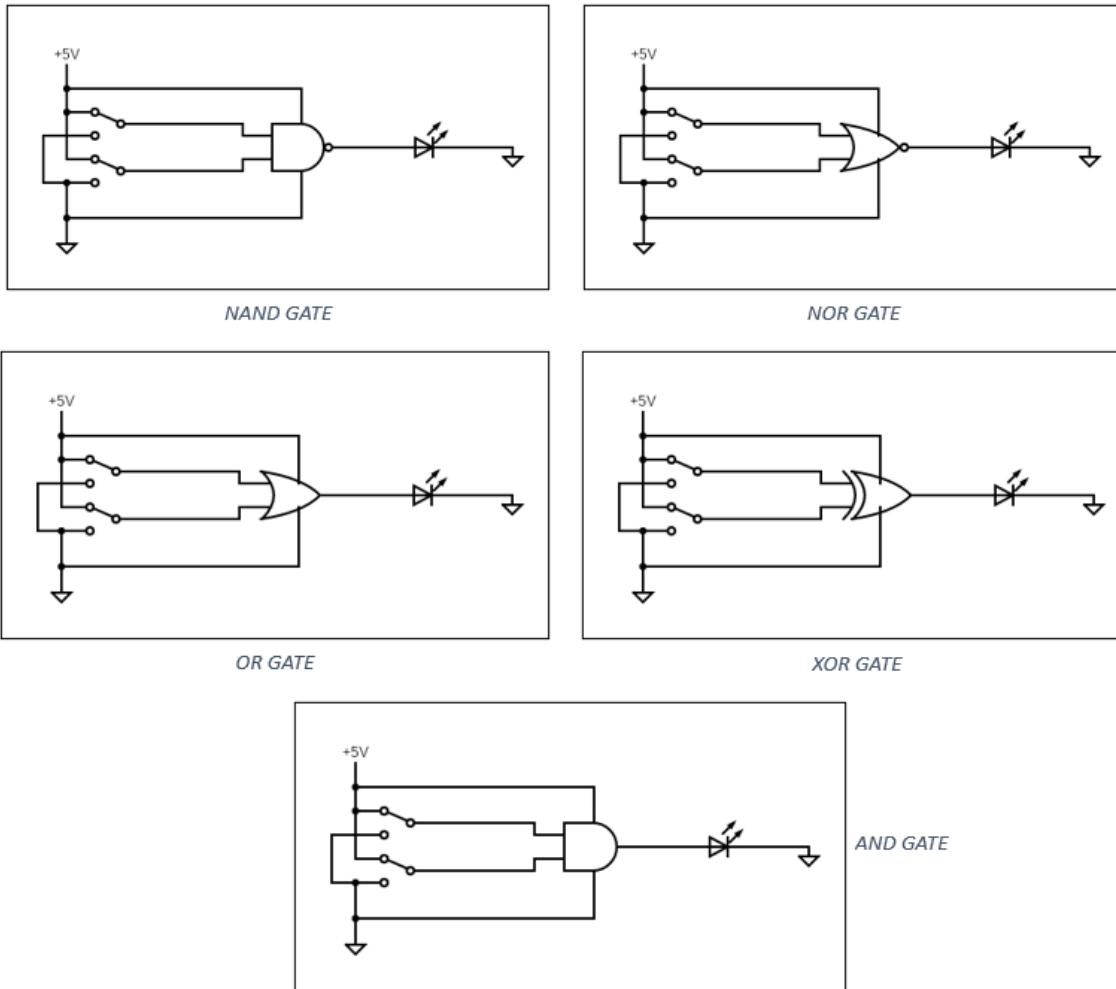
Electronic Components Used

1. Breadboard
2. TTL 74LSxx ICs (4)
3. CMOS CD40xx ICs (2)
4. RED and BLACK Wires
5. Digital Test Kit (Input Switches and Display Points)

Procedure

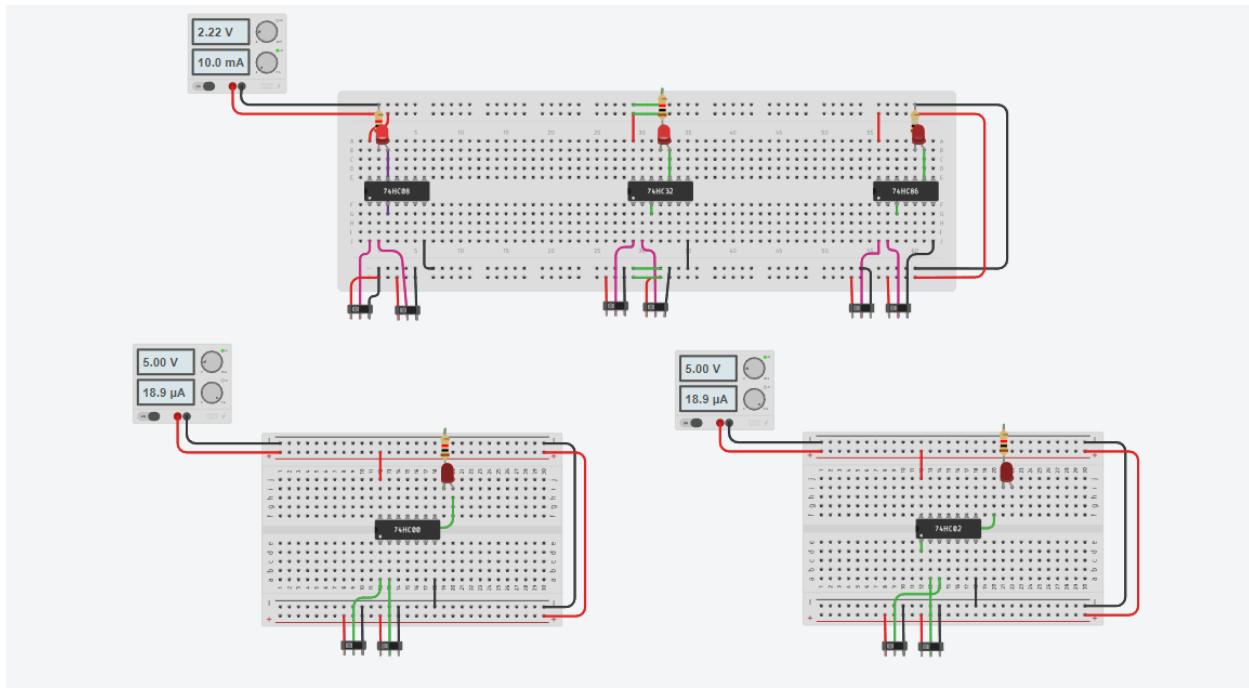
1. Connect the VCC and GND pins of the IC to the VCC and GND lines on the breadboard, using RED and BLACK wires, respectively.
2. Connect the two input pins of any one gate in the IC to two of the IP1-IP12 input switches, and connect the corresponding output pin of the IC to one of the DP1-DP8 display points provided in the Test Kit.
3. Apply the four possible combinations of binary values to the gate inputs one by one using the input switches. Tabulate the corresponding values of the gate output as observed on the LED display to create the truth table for the gate.
4. Verify that all four gates in the IC are identical by repeating steps 2 and 3 for the other three gates in the same IC.
5. Repeat steps 2-4 for each of the remaining ICs in both the TTL 74LSxx and CMOS CD40xx families.

Reference Circuits



Tinkercad

https://www.tinkercad.com/things/0fNf18Nmews-cool-bruticus-leelo/editel?sharecode=mUPn3ff2SDJnTMBw_qxZy0YrZA_KgWqCGAEB5iAgITw



Gates Found:

My Circuits

<https://drive.google.com/file/d/1fiPOcNb78CoOt7b5Ic7oM1T96T4k69B0/view?usp=sharing>

<https://drive.google.com/file/d/1S2dvTbNDRbPhaFMgbh7K2EqTUoENCJ2E/view?usp=sharing>

<https://drive.google.com/file/d/1yltcLRS6ZDqxSLOX1kQy2WWbDzU51nLQ/view?usp=sharing>

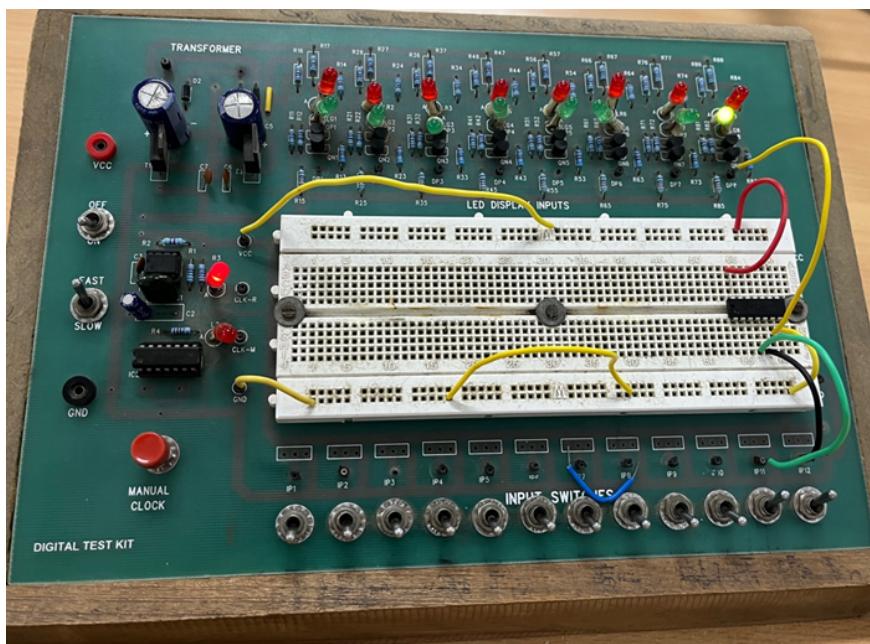
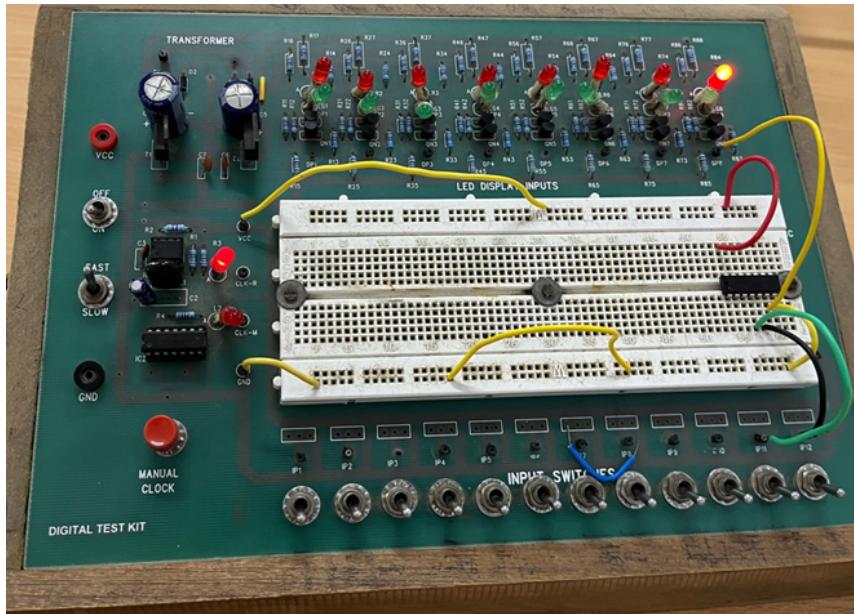
https://drive.google.com/file/d/1IF8ev3ZDE4i65kfa8KUQACwn_T8tbU76/view?usp=sharing

<https://drive.google.com/file/d/1MMoFAK0KB9MW-6bvM-G0LO9JNPoylpBH/view?usp=sharing>

<https://drive.google.com/file/d/1PFG7ot83USs5BzsZ0dgKht-eG9Dks6BE/view?usp=sharing>

Observation

OR Gate

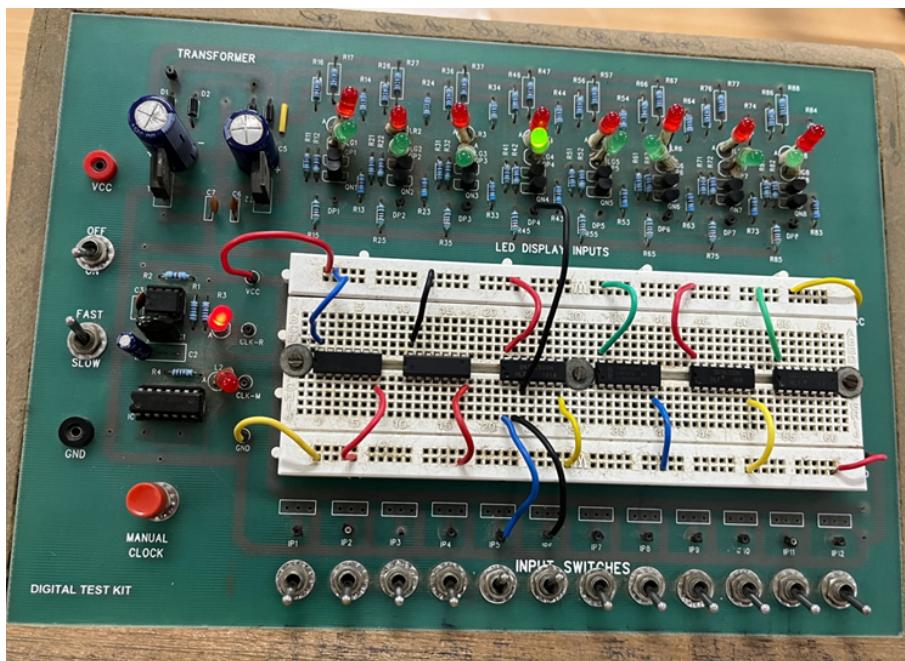
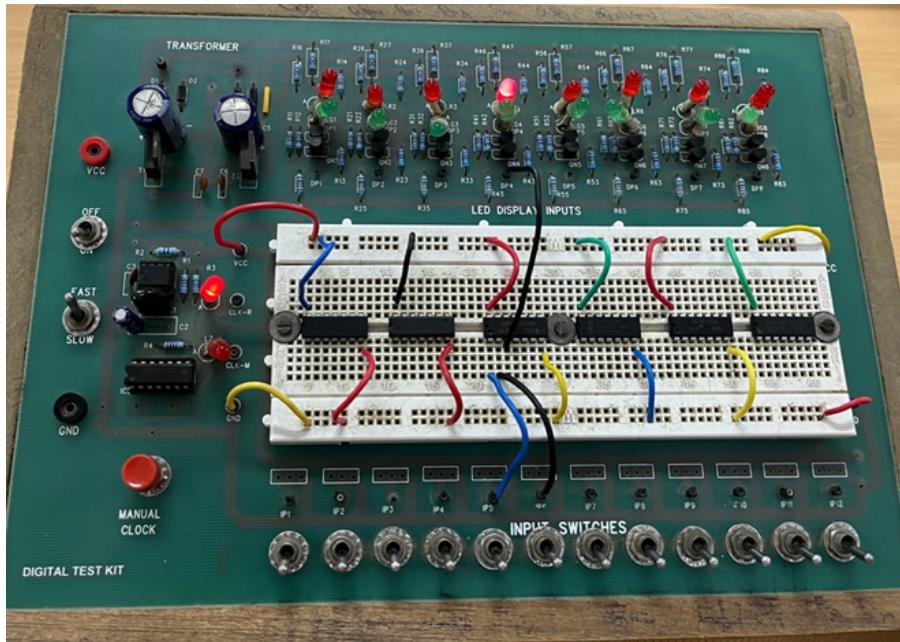


Observation Table

A	B	A OR B
0	0	0
0	1	1

1	0	1
1	1	1

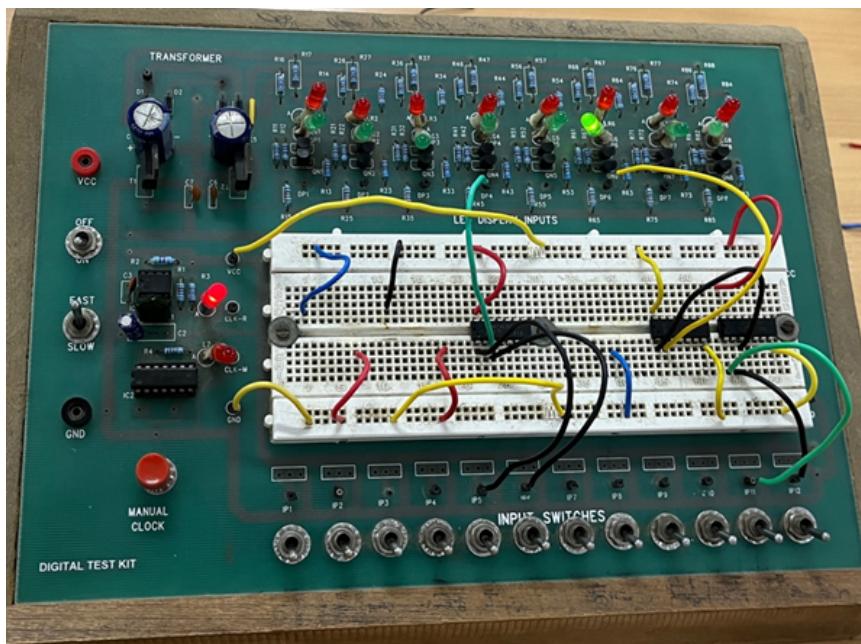
AND Gate

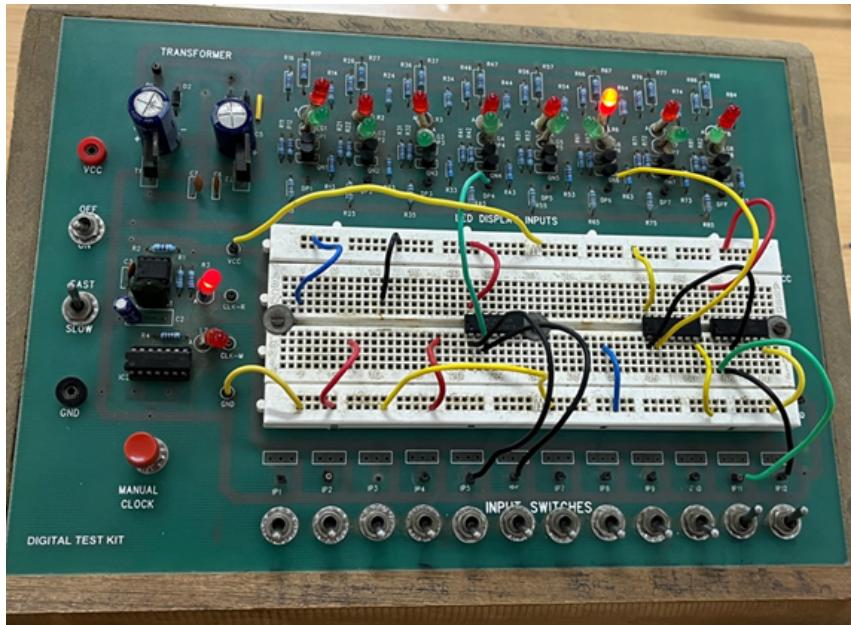


Observation Table

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

NOR Gate





Observation Table

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

NAND Gate

Observation Table

A	B	A NAND B
0	0	1
0	1	1
1	0	1

1	1	0
---	---	---

XOR Gate (74LSXX)

Observation Table

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

XOR Gate (CD40XX)

Observation Table

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

Conclusion

In this experiment, we have successfully identified the logic functions of different ICs, each containing four AND / OR / NAND / NOR / XOR gates. By connecting the ICs to the breadboard, applying binary input values, and tabulating the gate outputs to create truth tables,

we determined the logic functions of each gate within the ICs. Also, we have verified that all four gates in each IC are identical, meaning that the ICs function consistently.

Part C: Verification of De Morgan's Theorems

Aim/Objective

The objective of this experiment is to verify De Morgan's Theorems which state that $(A + B)' = A' \cdot B'$ and $(A \cdot B)' = A' + B'$.

Electronic Components Used

1. Breadboard
2. NAND Gates (2)
3. AND Gate
4. OR Gate
5. Digital Test Kit (Input Switches and Display Points)

Procedure

Verification of $(A + B)' = A' \cdot B'$

1. Set up a circuit consisting of two NAND gates and one AND gate to perform the function $Y = A' \cdot B'$, using a NAND gate with its two inputs connected together to perform the NOT function.
2. Apply all possible combinations of binary values for A and B to the circuit, following the procedure as in Part A, and record the corresponding output values.
3. Tabulate the truth table for this circuit.

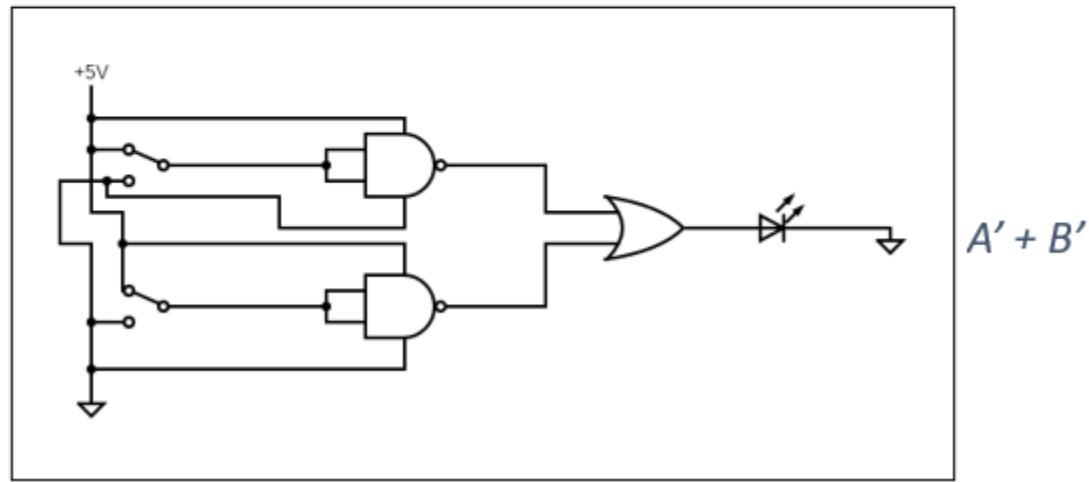
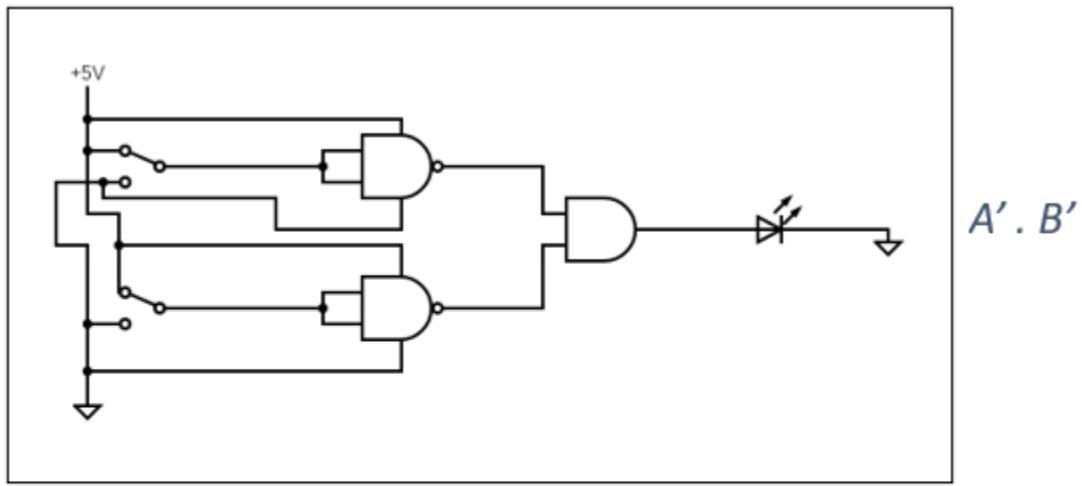
Truth Table

A	B	A'	B'	$(A+B)$	$(A+B)'$	$A' \cdot B'$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

Verification of $(A \cdot B)' = A' + B'$

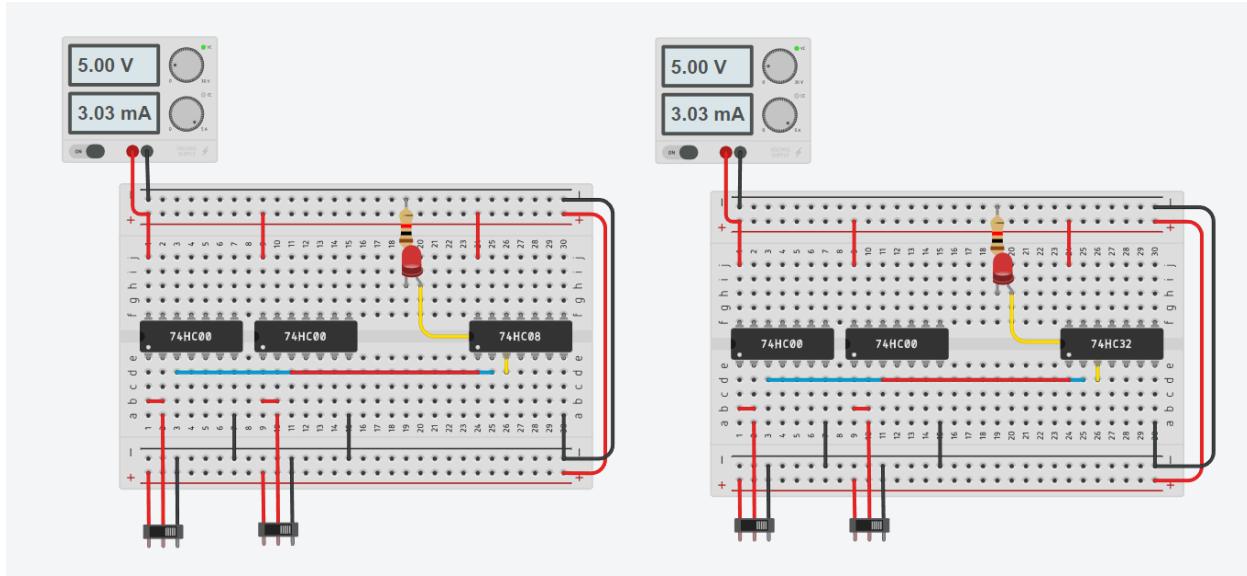
1. Repeat steps 1 and 2 of the previous verification, but this time use an OR gate instead of an AND gate to perform the function $Y = A' + B'$.
2. Apply all possible combinations of binary values for A and B to the circuit, record the corresponding output values, and tabulate the truth table for this circuit.

Reference Circuits



Tinkercad

https://www.tinkercad.com/things/iVpGwH43WV4/edit#?returnTo=%2Fclassrooms%2F65dBoBuNJYW%2Factivities%2Fgh3VY6VRVeR&sharecode=_RA-obkFLPQnOEFLCUVjtsqTL7sBYD4SgkNOSJ_pZdi



Truth Table

A	B	A'	B'	(A+B)	(A • B)	A' + B'
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

Conclusion

In this experiment, we verified De Morgan's Theorems through practical circuit implementations.

For the first theorem, $(A + B)' = A' \cdot B'$, we set up a circuit using NAND and AND gates and confirmed that the truth table matches that of a NOR gate, verifying the theorem.

For the second theorem, $(A \cdot B)' = A' + B'$, we repeated the process with an OR gate instead of an AND gate and again confirmed that the truth table matches that of a NOR gate, verifying this theorem as well.

Part D: Binary Full Adder

Aim/Objective

The objective of this experiment is to design and implement a binary Full Adder using Half Adders and verify its operation through a truth table.

Electronic Components Used

1. XOR Gates (2)
2. AND Gates (2)
3. Breadboard
4. Input Switches (3)
5. LED Displays (5)

Procedure

Half Adder Implementation

1. Set up a circuit for a Half Adder using an XOR gate and an AND gate.
2. Connect input A to one input switch and input B to another input switch.
3. Observe the outputs S1 and C1 on two LED displays.
4. Apply all possible combinations of binary values for A and B through the input switches.
5. Tabulate the values of S1 and C1 for each input combination and verify the operation of the Half Adder.

Full Adder Implementation

1. Set up another Half Adder circuit using additional XOR and AND gates from the same ICs used in step 1.
2. Connect input C and the S1 output generated by the first Half Adder as its inputs to generate the final SUM output and the C2 output.

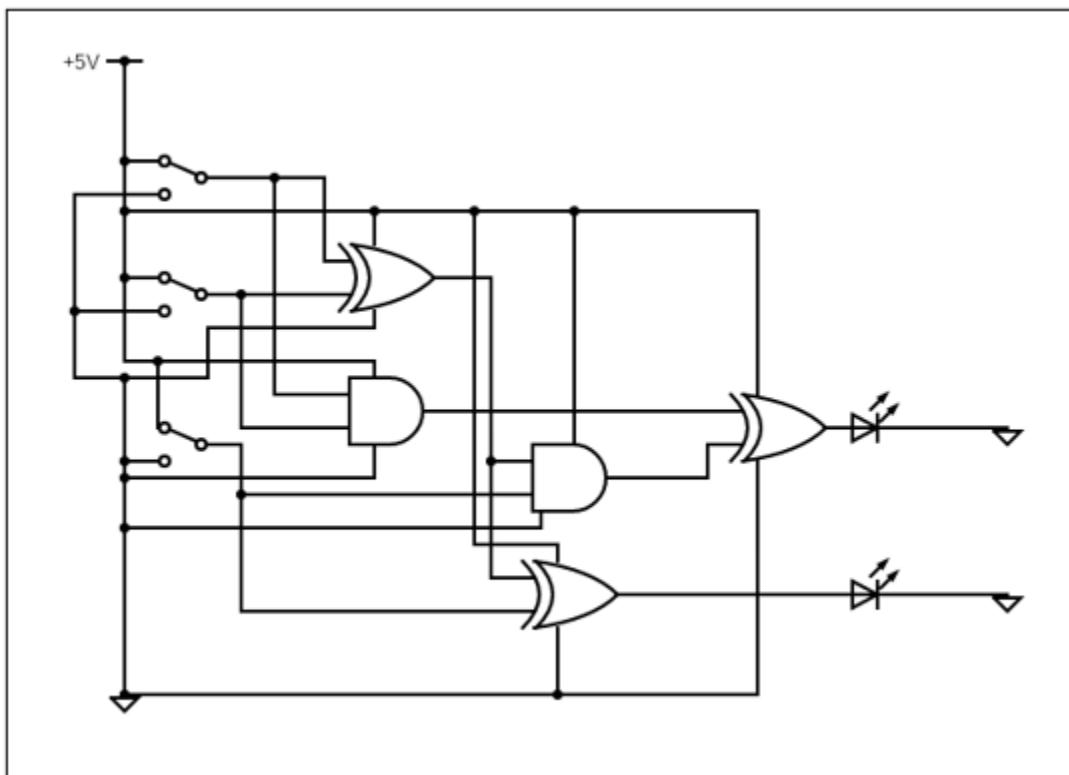
CARRY Output Logic Design

1. Generate the final CARRY output from the intermediate carry outputs C1 and C2.
2. Utilize the unused gates in the XOR and AND ICs deployed so far to achieve the logic for CARRY.

Truth Table Verification

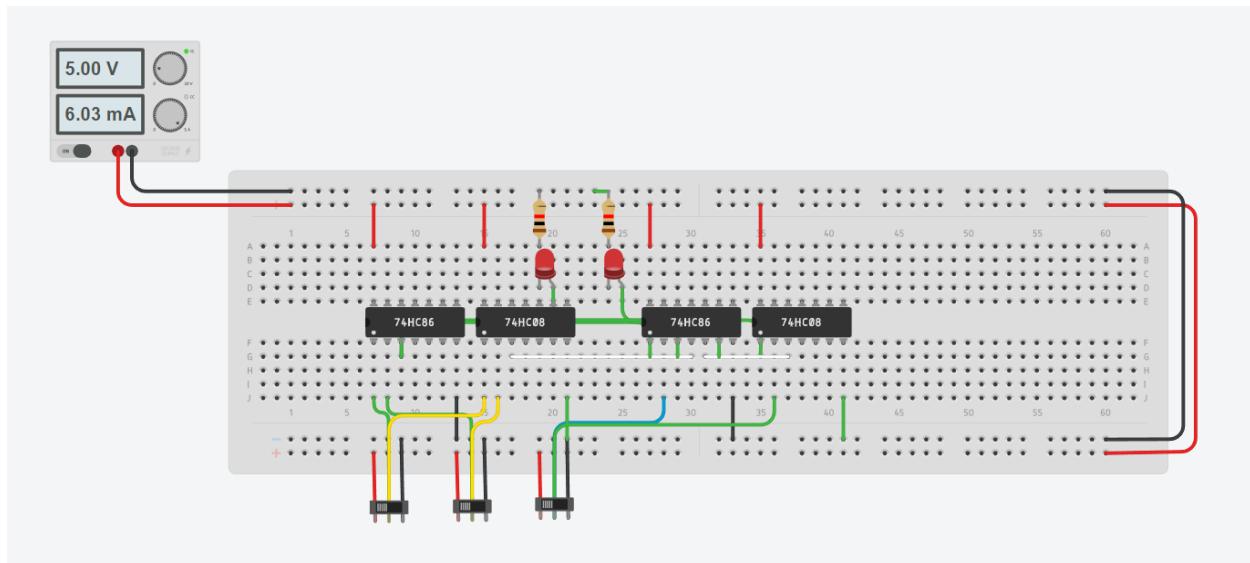
1. Verify the truth table experimentally by applying binary inputs A, B, and C through three input switches.
2. Display the outputs S1, C1, C2, SUM, and CARRY on the LED displays for all input combinations.
3. Tabulate the observed values of S1, C1, C2, SUM, and CARRY.
4. Compare the observed values with the expected values based on the Full Adder truth table.

Reference Circuits



Tinkercad

<https://www.tinkercad.com/things/48CmVvpIAV1/editel?returnTo=%2Fclassrooms%2F65dBoBuNJYW%2Factivities%2Fgh3VY6VRVeR&sharecode=p8jV8yc9yKEoVAd67h-BiBRjfTNDKLTQFpO3KsA3rsw>



TRUTH TABLE

A	B	C	SUM1	C1	SUM	C2	CARRY
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	1	0	0	1	1
1	0	0	1	0	1	0	0
1	0	1	1	0	0	1	1

1	1	0	0	1	0	0	1
1	1	1	0	1	1	0	1

Observed Results

SUM	CARRY
0	0
1	0
1	0
0	1
1	0
0	1
0	1
1	1

Conclusion

In this experiment, we have successfully designed and implemented a binary Full Adder using Half Adders composed of XOR and AND gates. We verified the operation of the Half Adder through experimental observations, and the Full Adder was constructed based on the combination of two Half Adders. We also designed the logic for the final CARRY output using the same type of gates (XOR and AND) to ensure a complete realization of the Full Adder without the need for a third IC. We experimentally verified the truth table for the Full Adder by applying inputs A, B, and C and observed the corresponding outputs S1, C1, C2, SUM, and CARRY. The observed values match the expected values based on the truth table, confirming the proper functioning of the binary Full Adder circuit.