# Experiment 4

Electronics Lab

## (A) Verify the NAND and NOR gates as universal logic gates

## Aim

To implement the logic functions i.e. AND, OR, NOT, Ex-OR, Ex- NOR and a logical expression with the help of NAND and NOR universal gates respectively.

## Components and Apparatus

| **Components** | **Quantity** |
|---|---|
| IC 7400 (NAND gate) | 1 |
| IC 7402 (NOR gate) | 1 |
| Digital trainer kit | 1 |
| Wires | As required |
| Probes | As required |

Table 1: List of components and apparatus required for the experiment

## Integrated Circuits (IC) Pin Diagrams

This experiment utilizes the IC 7400 (NAND gate) and the IC 7402 (NOR gate). Below are the pin diagrams for both ICs, essential for understanding their functionalities and ensuring correct circuit assembly.

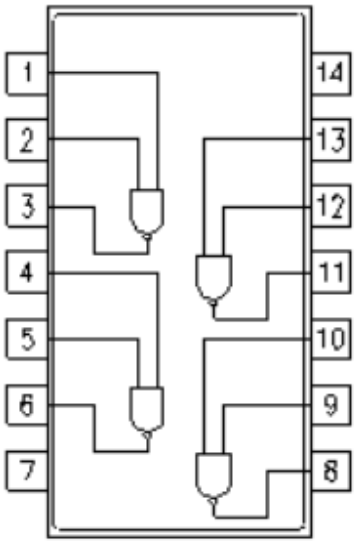


Figure 1: Pin diagram of IC 7400 (NAND gate)

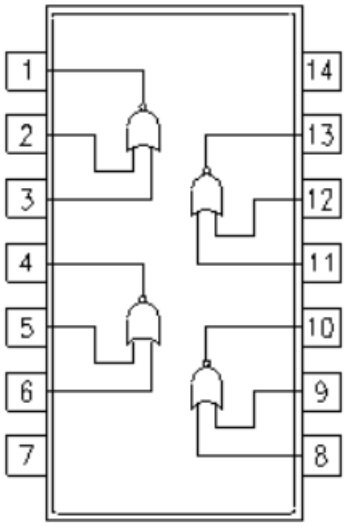


Figure 2: Pin diagram of IC 7402 (NOR gate)

These diagrams detail the functionality of each pin on the ICs, guiding proper integration into the breadboard. Familiarize yourself with these diagrams to avoid wiring errors during the experiment setup.

# Nand gate as Universal gate

NAND gate is actually a combination of two logic gates i.e. AND gate followed by NOT gate. So its output is the complement of the output of an AND gate. This gate can have a minimum of two inputs. By using only NAND gates, we can realize all logic functions: AND, OR, NOT, Ex-OR, Ex-NOR, NOR. So this gate is also called a universal gate.

## 1.1) NAND gates as OR gate

From DeMorgan's theorems:

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

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

So, give the inverted inputs to a NAND gate, obtain OR operation at output.

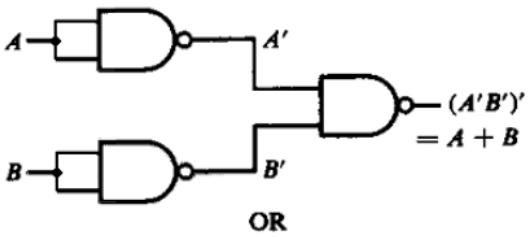


Figure 3: NAND gates as OR gate

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

Figure 4: Truth table of OR

## 1.2) NAND gates as AND gate

A NAND produces the complement of an AND gate. So, if the output of a NAND gate is inverted, the overall output will be that of an AND gate.

$$Y = ((A \cdot B)')'$$

$$Y = (A \cdot B)$$

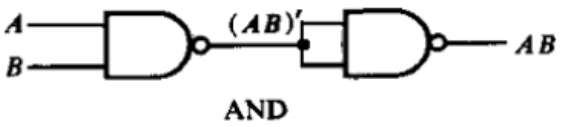


Figure 5: NAND gates as AND gate

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

Figure 6: Truth table of AND

## 1.3) NAND gates as Ex-OR gate

The output of a two-input Ex-OR gate is shown by: $Y = A'B + AB'$. This can be achieved with the logic diagram shown on the left side.

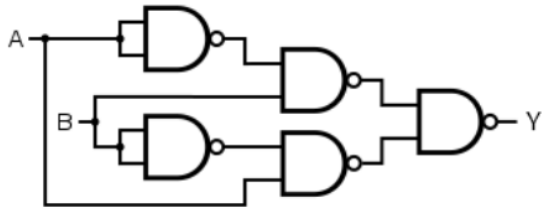


Figure 7: NAND gates as Ex-OR gate

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

Figure 8: Truth table of Ex-OR

## 1.4) NAND gates as Ex-NOR gate

Ex-NOR gate is actually Ex-OR gate followed by NOT gate. So give the output of Ex-OR gate to a NOT gate, the overall output is that of an Ex-NOR gate.

$$Y = AB + A'B'$$

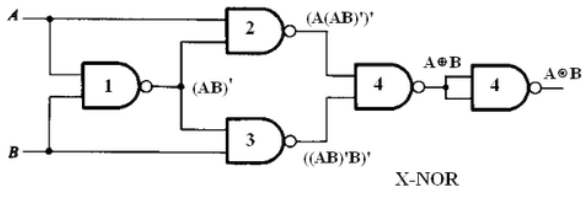


Figure 9: NAND gates as Ex-NOR gate

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

Figure 10: Truth table of Ex-NOR

# Nor gate as Universal Gate

NOR gate is actually a combination of two logic gates: OR gate followed by NOT gate. So its output is the complement of the output of an OR gate. This gate can have a minimum of two inputs, and the output is always one. By using only NOR gates, we can realize all logic functions: AND, OR, NOT, Ex-OR, Ex-NOR, NAND. So this gate is also called a universal gate.

## 2.1) NOR gates as OR gate

A NOR produces the complement of an OR gate. So, if the output of a NOR gate is inverted, the overall output will be that of an OR gate.

$$Y = ((A + B)')'$$

$$Y = (A + B)$$

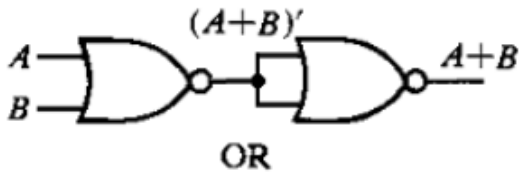


Figure 11: NOR gates as OR gate

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

Figure 12: Truth table of OR

## 2.2) NOR gates as AND gate

From DeMorgan's theorems:

$$(A + B)' = A'B'$$

$$(A' + B')' = A''B'' = AB$$

So, give the inverted inputs to a NOR gate to obtain the AND operation at output.

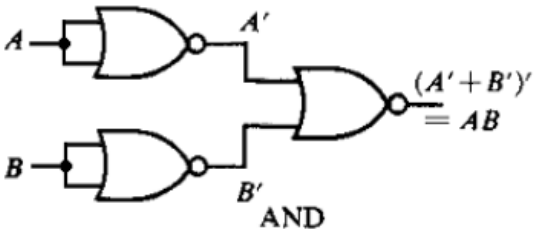


Figure 13: NOR gates as AND gate

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

Figure 14: Truth table of AND

## 2.3) NOR gates as Ex-OR gate

Ex-OR gate is actually an Ex-NOR gate followed by a NOT gate. So, give the output of an Ex-NOR gate to a NOT gate, the overall output is that of an Ex-OR gate.

$$Y = A'B + AB'$$

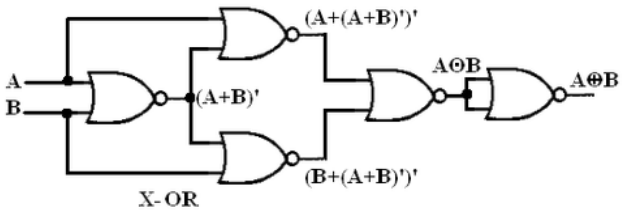


Figure 15: NOR gates as Ex-OR gate

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

Figure 16: Truth table of Ex-OR

### 2.4) NOR gates as Ex-NOR gate

The output of a two-input Ex-NOR gate is shown by: $Y = AB + A'B'$. This can be achieved with the logic diagram shown on the left side.

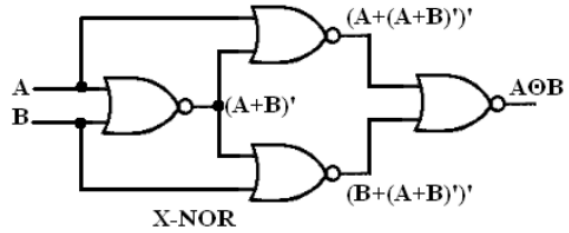


Figure 17: NOR gates as Ex-NOR gate

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

Figure 18: Truth table of Ex-NOR

## (B) Design and verification of the truth tables of Half adder circuit

## Aim

This experiment aims to design a Half Adder circuit using basic logic gates and to verify its truth table, demonstrating the circuit's ability to add two single-bit binary numbers.

# Components and Apparatus

| **Components** | **Quantity** |
|---|---|
| IC 7408 (AND gate) | 1 |
| IC 7486 (XOR gate) | 1 |
| Power supply | 1 |
| LEDs | 2 |
| Resistors (220 Ω) | 2 |
| Connecting wires | As required |
| Breadboard | 1 |

Table 2: List of components and apparatus required for the experiment

# Integrated Circuits (IC) Pin Diagrams

This experiment utilizes the IC 7408 (AND gate) and the IC 7486 (XOR gate). Below are the pin diagrams for both ICs.

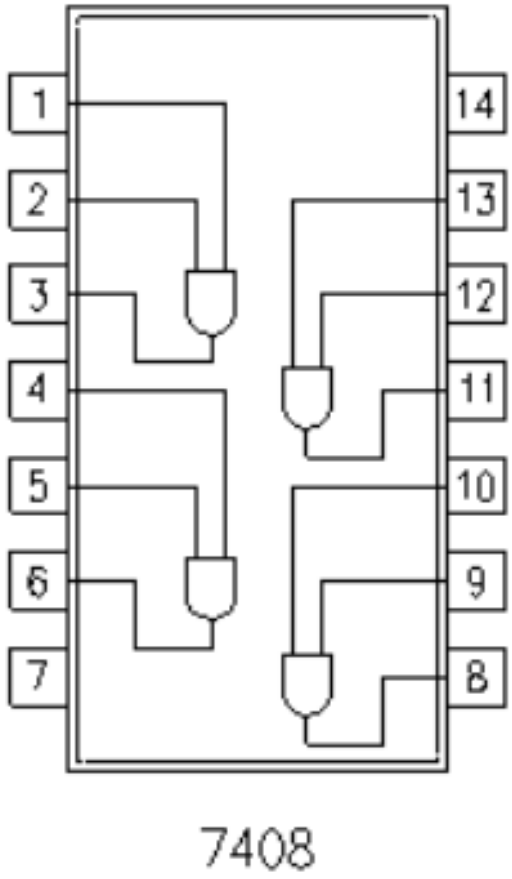


Figure 19: Pin diagram of IC 7408 (AND gate)

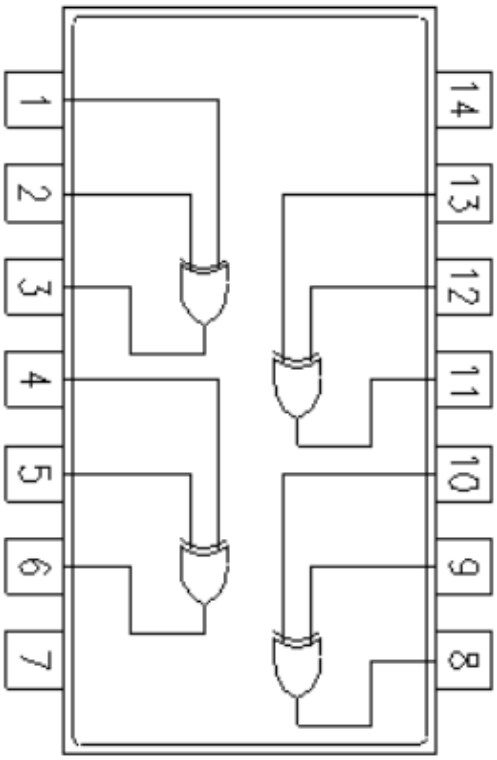


Figure 20: Pin diagram of IC 7486 (XOR gate)

These diagrams indicate the functionality of each pin on the IC, enabling the correct setup of the Half Adder circuit. Ensure that you refer to these diagrams while inserting the ICs into the breadboard to avoid any wiring mistakes.

# Theory

A Half Adder is a combinational circuit that performs the addition of two bits. It has two inputs, A and B, representing the bits to be added, and two outputs, Sum (S) and Carry (C). The Sum represents the least significant bit of the addition, while the Carry represents an overflow into the next significant bit.

The logical expressions for the outputs are given by:

$$S = A \oplus B$$

$$C = A \cdot B$$

where $\oplus$ represents the XOR operation and $\cdot$ represents the AND operation.

Functional Table of Half Adder:

| A | B | Sum | Carry |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Table 3: Functional Table of Half Adder

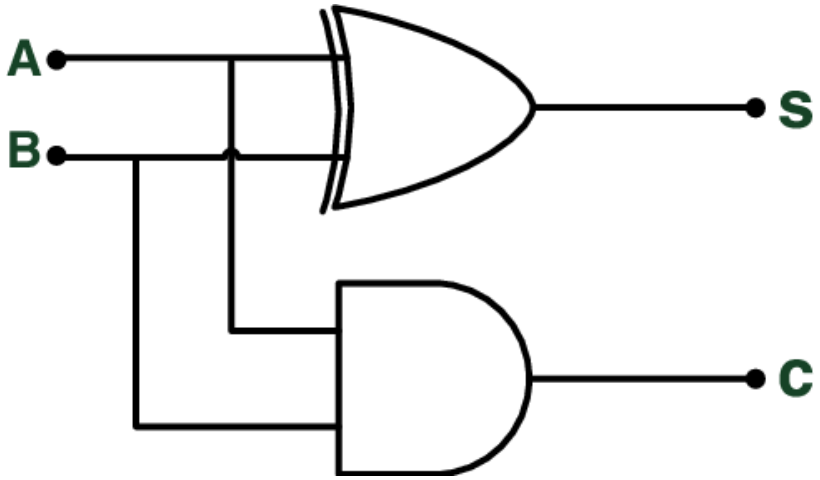


Figure 21: Block diagram of a Half Adder

# Procedure

1. Place the breadboard gently on the observation table.
2. Fix the IC between the half shadow line of the breadboard to avoid voltage shortage.
3. Connect the main voltage source (Vcc) to the last pin of the IC and the ground of IC to the ground terminal on the digital lab kit.
4. Provide inputs at the gate of the ICs using connecting wires according to the IC pin configuration.
5. Connect output pins to the LEDs on the digital lab kit.
6. Switch on the power supply.
7. Observe the LED: if it glows, the output is true (1); if it doesn't, the output is false (0).
8. Verify the truth table based on the LED outputs.

# Precautions

- Ensure all ICs are functional before starting the experiment.
- Make all connections tight and secure.
- Always connect ground before Vcc.

- Use appropriate wires for different types of circuits.
- Turn off the kit before changing connections.
- After completing the experiment, switch off the apparatus.

## Conclusion

Discuss the operation of the Half Adder circuit based on the observed truth table and confirm whether the experimental results align with the theoretical expectations.

### (C) Verification of JK Flip-Flop and SR Flip-Flop

## Aim

To understand and verify the functioning of JK and SR Flip-Flops by constructing their circuits on a digital trainer kit and observing their behavior through different input combinations.

## Components and Apparatus

| **Components** | **Quantity** |
|---|---|
| IC 7476 (JK Flip-Flop) | 1 |
| IC 7474 (SR Flip-Flop) | 1 |
| Digital trainer kit | 1 |
| Clock pulse generator | 1 |
| LEDs | As required |
| Resistors (220 Ω) | As required |
| Connecting wires | As required |
| Breadboard | 1 |

Table 4: List of components and apparatus required for the verification of JK and SR Flip-Flops.

## Integrated Circuits (IC) Pin Diagrams

This section will introduce the pin diagrams for IC 7476 (JK Flip-Flop) and IC 7474 (SR Flip-Flop), crucial for understanding their pin configurations and ensuring correct connections during circuit assembly.

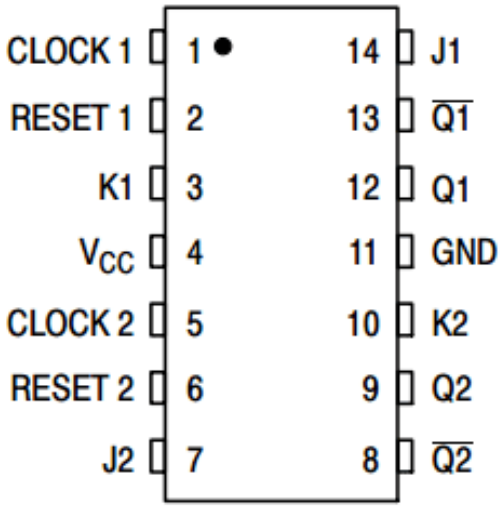


Figure 22: Pin diagram of IC 7476 (JK Flip-Flop)

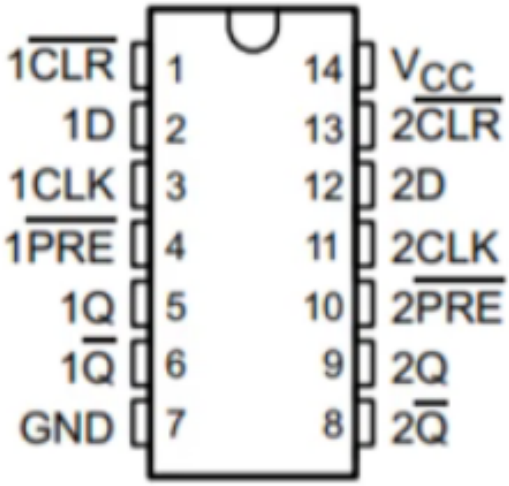


Figure 23: Pin diagram of IC 7474 (SR Flip-Flop)

# Theory

Flip-Flops are fundamental building blocks in digital electronics systems used for storage and transfer of digital data. They can store a single bit of data and are based on bistable states, meaning they have two stable states which represent binary 0 and 1.

## JK Flip-Flop

JK Flip-Flop is known for its versatility. The JK inputs can overcome the limitations of the SR Flip-Flop (i.e., the undefined state). It functions similarly to the SR Flip-Flop but with the JK input handling the previously undefined state by toggling the output.

### Truth Table

| J | K | Q(next) |
|---|---|---|
| 0 | 0 | Q(previous) |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | $\overline{Q(previous)}$ |

Table 5: Truth table of a JK Flip-Flop

### Equations

The operation of a JK Flip-Flop can be described by the following equations:

$$Q(next) = J \cdot \overline{Q(previous)} + \overline{K} \cdot Q(previous)$$

$$\overline{Q(next)} = \overline{J} \cdot Q(previous) + K \cdot \overline{Q(previous)}$$

## SR Flip-Flop

SR Flip-Flop (Set-Reset Flip-Flop) operates with two inputs, Set (S) and Reset (R), and provides basic memory storage operation. It has two stable states, making it a bistable. Its functionality is determined by the combination of inputs applied.

### Truth Table

| S | R | Q(next) |
|---|---|---|
| 0 | 0 | Q(previous) |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | Undefined |

Table 6: Truth table of an SR Flip-Flop

### Equations

The operation of an SR Flip-Flop can be described by the following equations:

$$Q(next) = S + R \cdot \overline{Q(previous)}$$

$$\overline{Q(next)} = R + S \cdot Q(previous)$$

# Procedure

1. Start by placing the breadboard on your work area and carefully inserting the ICs in the middle section to avoid any short circuits.
2. Connect the Vcc and Ground to the appropriate pins of the ICs as per the pin diagram.
3. For the JK Flip-Flop: Connect the J and K inputs to the switch outputs, the clock input to the clock pulse generator, and the Q and Q' outputs to LEDs through resistors.
4. For the SR Flip-Flop: Connect the S and R inputs to switch outputs, the clock input to the clock pulse generator, and the Q and Q' outputs to LEDs through resistors.
5. Apply different combinations of inputs for both JK and SR Flip-Flops by using the switches and observe the changes in the LED states.
6. Document the output states for each input combination to verify the truth tables of both Flip-Flops.

# Conclusion

Through this experiment, the fundamental operational principles of the JK and SR Flip-Flops should be clearly understood. By constructing the circuits and verifying their behavior against the theoretical truth tables, the experiment demonstrates how Flip-Flops serve as essential elements in digital memory and logic circuits.