Department of Biomedical Engineering
Electronics Laboratory / Third stage
Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi
Email: hussam.jawad@uomus.edu.iq

Experiment No. 6

SR Flip Flop

1. Introduction

Flip-flops are fundamental sequential circuit elements used for storing binary states. The **SR (Set-Reset) flip-flop** can be designed using either **NAND** or **NOR** gates, each with distinct input conditions and behavior. The **NAND-based SR flip-flop** operates with active-low inputs, while the **NOR-based design** uses active-high inputs. Understanding these implementations helps in analyzing their characteristics.

1.1 Objective

To design and analyze an **SR flip-flop** using **NAND and NOR gates**, demonstrating their role in implementing bistable circuits. This experiment aims to compare both designs' behavior and examine their truth tables.

1.2 Work Environment

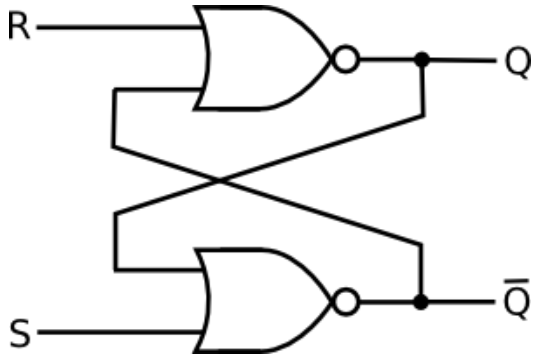
The CircuitMaker software is used to design and simulate logical circuits.

1.3 Theory

Flip-flops are essential components in digital circuits, used to store and control binary data. The **SR (Set-Reset) flip-flop** is one of the fundamental bistable circuits, having two inputs—**Set (S) and Reset (R)**—and two stable output states. It can be designed using either **NAND** or **NOR** gates, each affecting its operation differently.

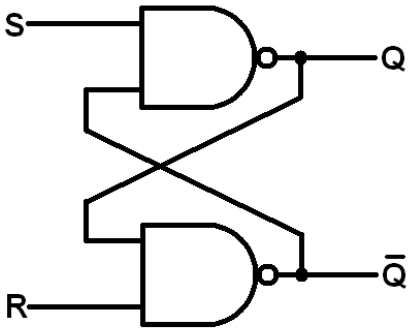
1.3.1 SR Flip-Flop Using NOR Gates

Mode of operation	Input		Output	
	S	R	Q	$\overline{Q}$
Hold	0	0	No change	
Set	0	1	1	0
Reset	1	0	0	1
Prohibited	1	1	1	1



1.3.2 SR Flip-Flop Using NAND Gates

Mode of operation	Input		Output	
	S	R	Q	$\overline{Q}$
Prohibited	0	0	1	1
Set	0	1	1	0
Reset	1	0	0	1
Hold	1	1	No change	



1.3.3 Characteristics and Applications

The SR flip-flop is the basis for more complex flip-flops (D, JK, and T) and is widely used in latches, registers, and memory circuits. The key differences between NAND and NOR implementations lie in their **input logic levels and the handling of invalid states**. Understanding these variations helps in designing stable and efficient sequential logic circuits

1.3 Experiment

Designing SR Flip-Flop Using NOR Gates

1. **Open CircuitMaker** and create a new schematic.
2. **Add Components**:
   - Two **NOR gates**.
   - Two **logic switches** (for inputs S and R).
   - Two **LEDs** (for outputs Q and $\bar{Q}$).
   - **Power source (Logic switch)**.
3. **Connect the Circuit**:
   - Connect the output of the first NOR gate to one input of the second NOR gate.
   - Connect the output of the second NOR gate to one input of the first NOR gate.
   - Connect **S** to the other input of the first NOR gate.
   - Connect **R** to the other input of the second NOR gate.
   - Connect **LEDs** to Q and $\bar{Q}$ to observe output changes.
4. **Run Simulation** and observe the circuit behavior for different **S and R** combinations.

Designing SR Flip-Flop Using NAND Gates

Open CircuitMaker and start a new schematic.

1. **Add Components**:
   - Two **NAND gates**.
   - Two **logic switches** (for inputs S and R).
   - Two **LEDs** (for outputs Q and $\bar{Q}$).
   - **Power source (Logic switch)**.

2. **Connect the Circuit**:
    - Connect the output of the first NAND gate to one input of the second NAND gate.
    - Connect the output of the second NAND gate to one input of the first NAND gate.
    - Connect $\bar{\mathbf{S}}$ (inverted S) to one input of the first NAND gate.
    - Connect $\bar{\mathbf{R}}$ (inverted R) to one input of the second NAND gate.
    - Connect LEDs to **Q and $\bar{\mathbf{Q}}$** to visualize state changes.
3. **Run Simulation** and test different input conditions.

Discussion:

- Why is the JK flip-flop preferred over the SR flip-flop?
- Compare the truth tables of an SR flip-flop implemented using NAND and NOR gates.
- Draw the circuit diagram of an SR flip-flop with a clock. What is the difference in the truth table between a clocked SR flip-flop and an SR flip-flop without a clock (using NAND gates)?
- State one function of a flip-flop.