Pak-Austria Fachhochschule: Institute of Applied Sciences & Technology, Haripur, Pakistan

School of Computing Science

Programming Fundamentals Lab

Lab # 11

Submitted By: Haider Nasir

Registration # B24S0308AI088

Submitted to: Engr. Raghib

Date of submission: 4/25/2025

Pak-Austria Fachhochschule: Institute of Applied Sciences & Technology, Haripur, Pakistan

School of Computing Science

Lab 11:

Implementing Flip-Flops and Shift Register on Electronic Workbench (EWB).

Objectives :

To understand the fundamental working principles of different types of flip-flops (SR, JK, D, T)
To learn how to implement and simulate basic flip-flop circuits
To analyze the behavior of flip-flops through state tables
To verify the operation of shift registers

Tools/Equipment Requirement:

PC or Laptop
Electronic Workbench

Theoretical Explanation:

Flip-Flops:

Flip-flops are fundamental sequential logic circuits that can store binary information (0 or 1). Unlike combinational logic circuits that simply process inputs to produce outputs, flip-flops have "memory" - they can maintain their output state even after input signals change.

Flip-flops are edge-triggered devices, meaning they change state based on clock signal transitions (rising or falling edge). They serve as basic memory elements in digital circuits and are used to build registers, counters, memory units, and other sequential circuits.

Types:

SR Flip-Flop (Set-Reset)

D Flip-Flop (Data or Delay)

JK Flip-Flop

Shift Register:

A shift register is a sequential digital circuit that can store and shift binary data. It consists of a series of flip-flops connected in a chain, where the output of one flip-flop becomes the input to the next. Shift registers are used for data storage, data transfer, data manipulation, and in various applications including serial-to-parallel and parallel-to-serial conversion.

A universal shift register, as shown in the circuit/logic diagrams is an integrated logic circuit that can transfer data in different modes. The different modes of operation are as listed in a table below:

S0	S1	Operating Mode
0	0	Locked (No change)
0	1	Shift-Left
1	0	Shift-Right
1	1	Parallel Loading

The synchronous operation of the device is determined by the mode select inputs (S0, S1). As shown above in the table, data can be entered and shifted from left to right (A0 to A1 to A2, etc.) or, right to left (A3 to A2 to AQ1, etc.) or parallel data can be entered, loading all 4 bits of the register simultaneously. When both S0 and S1 are LOW,existing data is retained in a locked/hold ("do nothing") mode. The first and last stages provideD-type serial data inputs (Slrt, Sllft) to allow multistage shift right or shift left data transfers without interfering with parallel load operation. The four parallel data inputs (I0 to I3) are D-type inputs. Data appearing on the I0 to I3 inputs, when S0 and S1 are HIGH, is transferred to the A3 to A0 outputs respectively, following the next LOW to-HIGH transition of the clock.

Procedure:

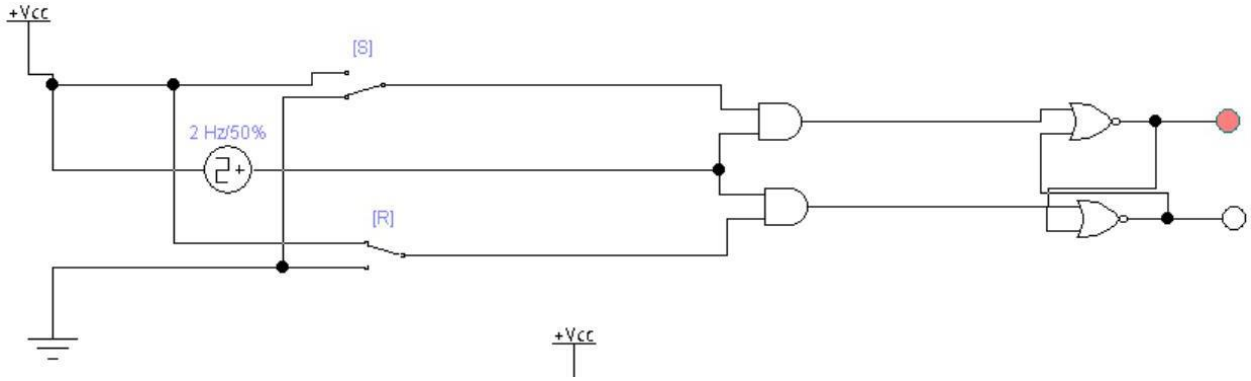
Place necessary components from the component library onto the workspace:
Logic gates (AND, OR, NOT, NAND, NOR gates)
Switches or digital sources for inputs
Clock generator (for synchronous operation)
LEDs or output displays for visualization
Power supply and ground connections
Connect the power supply (+Vcc) to appropriate components and establish ground connections.
Set up the basic flip-flop structure by cross-coupling appropriate gates:
Connect outputs of gates to inputs of other gates to create feedback paths
Add input control lines for state changes
Connect clock signals to synchronize operations

Add input switches to control the flip-flop behavior:
Connect switches to the appropriate input terminals
Configure switch states (normally open/closed as required)
Implement any additional logic required for specific functionality:
Add inverters for complementary signals
Add AND gates for clock synchronization
Create feedback paths as needed for toggle functionality
Connect output indicators (LEDs) to visualize the state:
Connect Q output to one LED
Connect $\bar{Q}$ (complementary output) to another LED

Lab Tasks:

SR FLIP FLOP USING LOGIC GATES

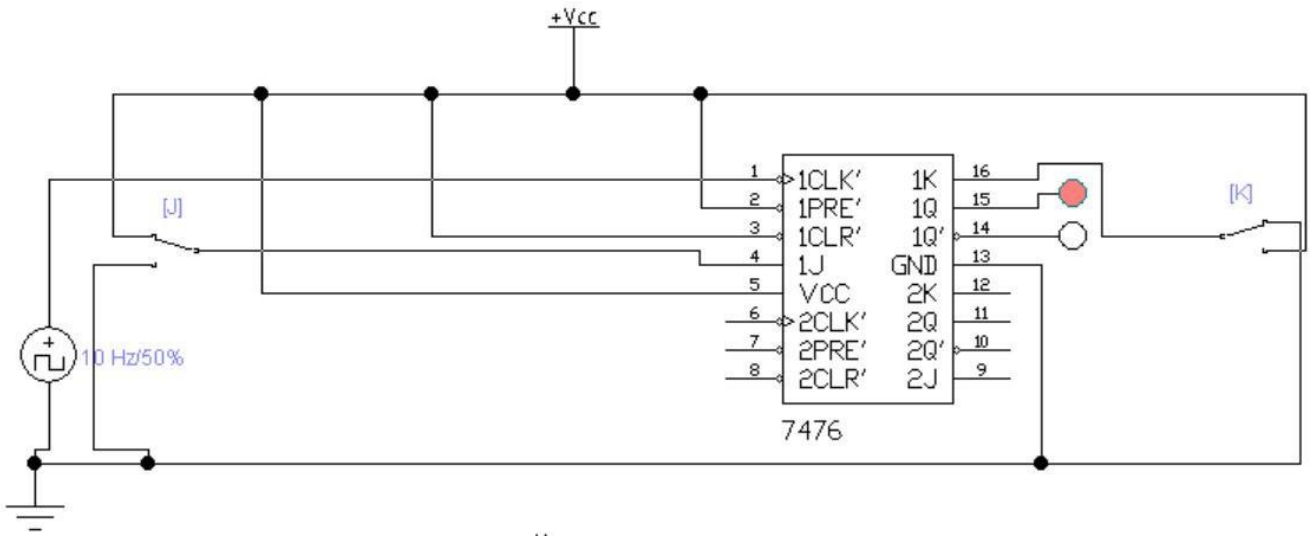
CIRCUIT DIAGRAM



TRUTH TABLE

S	R	Q_{n+1}
0	0	Hold
0	1	0
1	0	1
1	1	invalid

JK FLIP FLOP USING IC'S

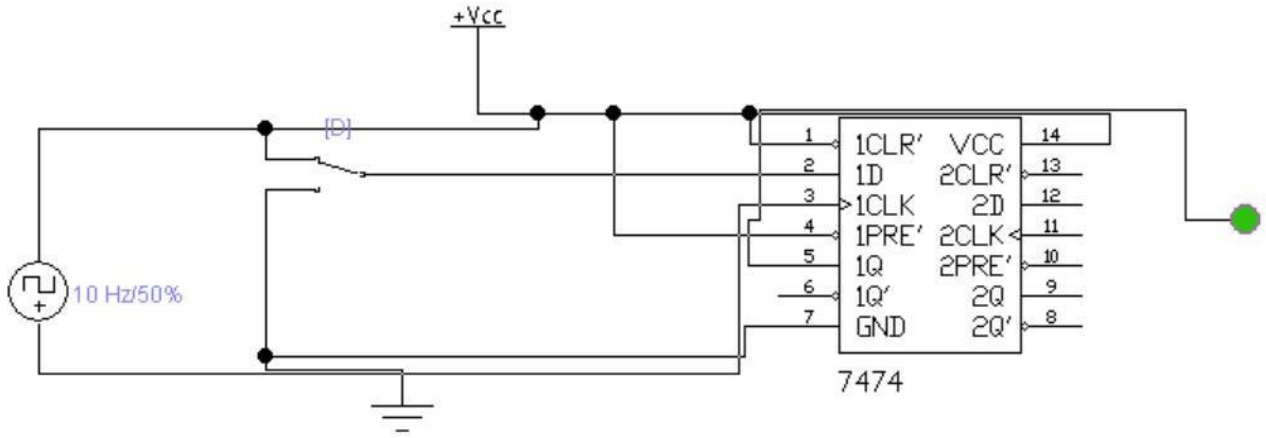


TRUTH TABLE

S	R	Q_{n+1}
0	**0**	**Hold**
0	**1**	**0**
1	**0**	**1**
1	**1**	**TOGGLE**

DESIGN D FLIP FLOP USING IC'S

CIRCUIT DIAGRAM

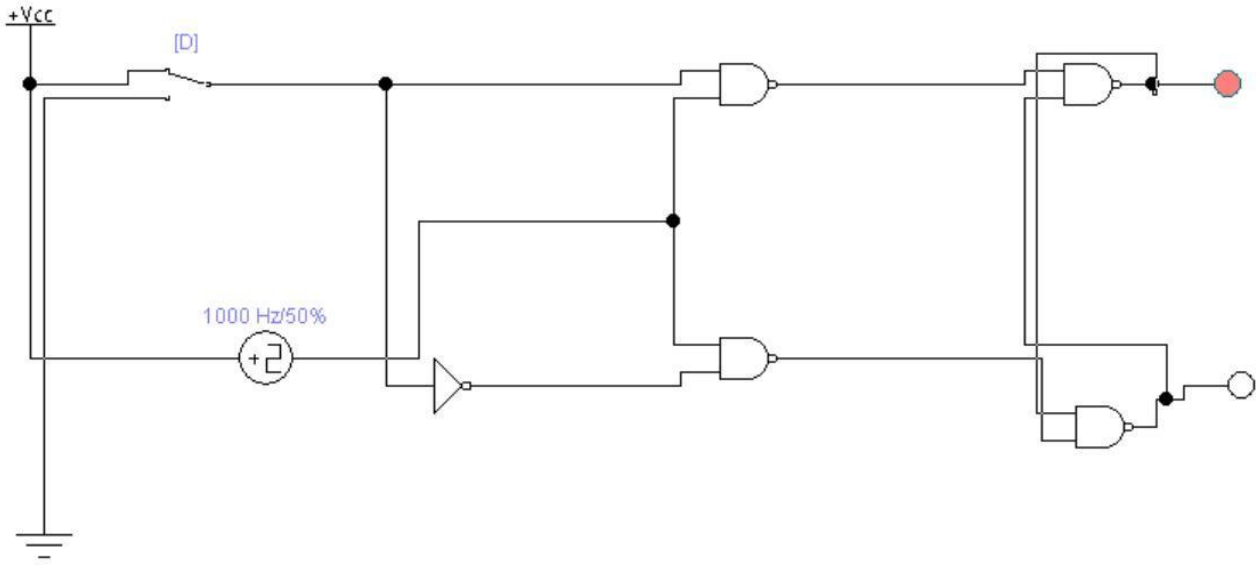


TRUTH TABLE

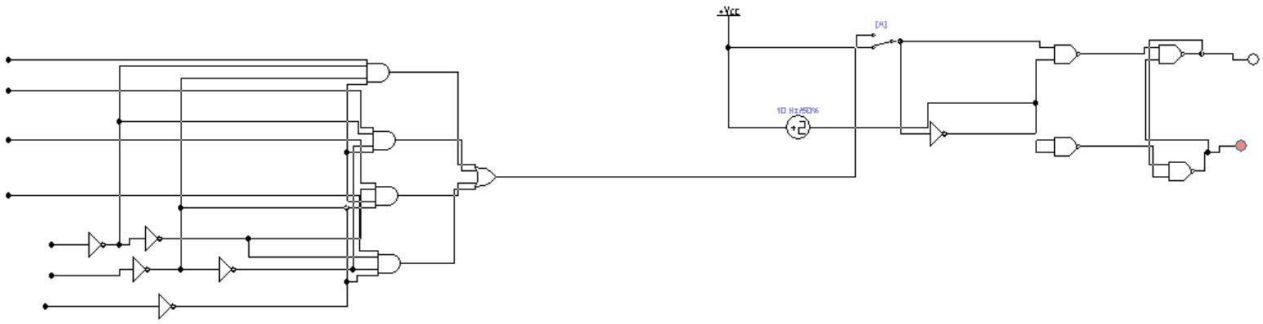
D	Q
0	0
1	1

DESIGN D FLIP FLOP USING LOGIC GATES

CIRCUIT DIAGRAM



Universal Shift Register:



Conclusion:

In this lab we successfully constructed various flip-flop configurations and shift register designs using basic logic gates, verifying their functionality through simulation. The lab demonstrated how these circuits store and manipulate binary information, respond to clock signals, and maintain state information. By implementing these fundamental building blocks of digital systems, we gained practical understanding of memory elements that form the basis for counters, registers, and more complex digital components.