Synthesis of Flip-Flops

Name - Aditya Kumar
UID-24BCS12427
SUBJECT- COMPUTERORGANIZATION AND ARCHITECTURE
SUBJECT CODE- 24CST-201
SECTION/GROUP- 712-A

Aim:

To study and synthesize various types of flip-flops such as RS, JK, D, and T flip-flops and to understand their working principles, timing behavior, and the race-around condition.

Apparatus Required:

- Logic Trainer Kit / Simulator Software
- Logic Gates (NAND, NOR, AND, OR, NOT)
- Connecting wires
- Power Supply (5V DC)
- LEDs (for output indication)
- Breadboard or Virtual Circuit Editor

Objective:

- To understand the basic concepts of flip-flops as elementary units of sequential circuits.
- To understand what race-around condition is and why it occurs in a JK flip-flop.
- To know how the race-around condition in JK flip-flop is avoided.
- To understand what problems may occur in the master-slave JK flip-flop.

- To know the need for master-slave JK flip-flop with asynchronous preset and clear.
- To examine the behavior of flip-flops for both the working module and the student-designed module.
- To perform simulation-based analysis using gate-level and behavioral flip-flops.

Recommended Learning Activities:

- Perform the experiment in two stages:
    1. On the **given encapsulated working module** (to observe behavior).
    2. On the **student-designed circuit module** (to perform detailed circuit and error analysis).
- Proceed to the advanced stage to perform design-based assignments.

Test Plan:

- In master-slave flip-flop, check when the master and slave change state relative to the clock.
- Compare design differences between master-slave D and master-slave JK flip-flops.

Assignments:

1. Design an edge-triggered D flip-flop.
2. Design a JK flip-flop with asynchronous preset and clear.
3. Design a master-slave D flip-flop.
4. Design a T flip-flop using JK flip-flop.

Theory:

In previous experiments, circuits were combinational, where output depends only on current input. Sequential circuits, however, have memory elements where output depends on both **current input and past output** .

Flip-flops are the simplest sequential circuits that store 1-bit information.

1. Basic Flip-Flop

A basic flip-flop can be constructed using two cross-coupled NAND or NOR gates.

It has **Set (S)** and **Reset (R)** inputs and outputs **Q**and **Q′** .
When S=1, Q=1 (Set); when R=1, Q=0 (Reset).

2. Clocked RS Flip-Flop

Responds only when clock = 1.

Clock	S	R	Q(next)	Description
1	0	0	Q	No Change
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	?	Invalid

3. JK Flip-Flop

An improved RS flip-flop where the invalid state is removed.

Clock	J	K	Q(next)	Description
1	0	0	Q	No change
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	Q′	Toggle

When J=K=1 and the clock pulse width is large, output toggles rapidly (race-around condition).
Master-slave JK flip-flop solves this by using two stages triggered by opposite clock phases.

4. D Flip-Flop

Derived from JK flip-flop with K = ¬J.

Transfers the D input to Q on every clock edge.

Clock	D	Q(next)	Description
↑	0	0	Reset
↑	1	1	Set

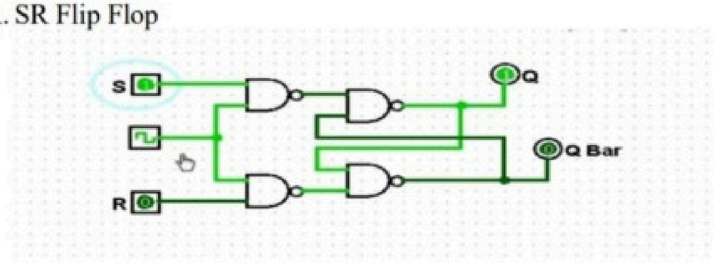
5. T Flip-Flop

Derived from JK flip-flop by connecting J=K=T.

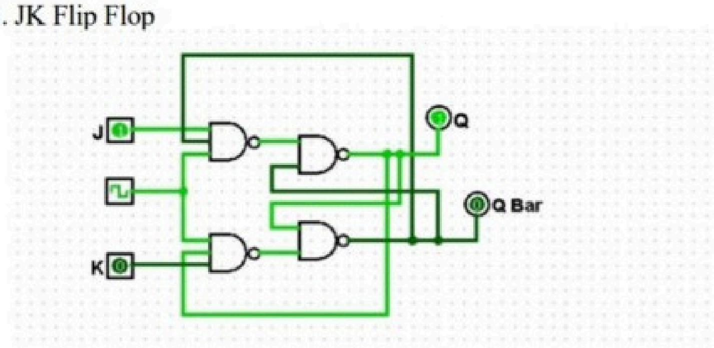
Clock	T	Q(next)	Description
↑	0	Q	No Change
↑	1	Q′	Toggle

Circuit Diagrams

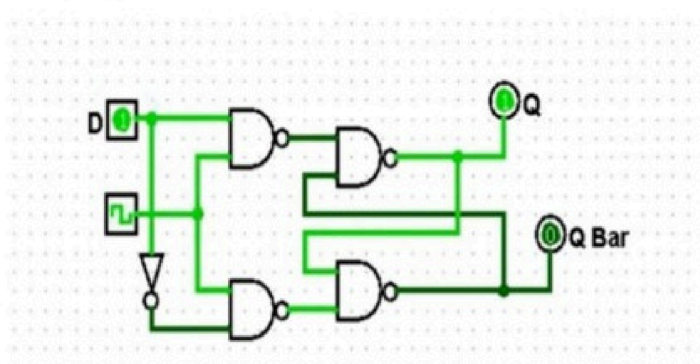
A. SR Flip Flop



B. JK Flip Flop



C. D Flip Flop



Label	**Description**
A. SR Flip-Flop	Constructed using NAND gates. Has Set (S) and Reset (R) inputs controlling Q and Q’.
B. JK Flip-Flop	Modification of SR Flip-Flop that eliminates invalid state. Inputs J and K control toggling.
C. D Flip-Flop	Derived from JK Flip-Flop; transfers input D to output Q on each clock edge.

Experiment:
General Guidelines to Use the Simulator

1. Start the simulator as directed.
2. The simulator supports **5-valued logic** (Unknown, True, False, High Impedance, Invalid).
3. Select any logic component and click on the canvas to place it.
4. Use the “Show Pin Config” button to view pin numbers.
5. Use the **Connection Tool** to link terminals (source → target).
6. Use **Bit Switch** for input toggles and **Bit Display** for output monitoring.
7. Use **Start/Stop Clock** for clock control; adjust period using “Set Clock”.
8. Click **Simulate** to start circuit execution.
9. Observe input-output behavior and waveforms using “Plot Graph”.
10. Save circuit files with .logic extension for future use.

Procedure:

1. Start the simulator and ensure 5-valued logic support is enabled.
2. Open or build circuits for RS, JK, D, and T flip-flops.
3. Place required components (flip-flops, clock, bit switches, bit displays).
4. Connect all inputs, outputs, and the clock properly.
5. Turn on **case analysis** for flip-flops requiring it.
6. Start the simulation and toggle inputs as per the objective.

7. Observe outputs, waveforms, and race-around conditions.

8. Compare results with theoretical truth tables.

Components Required:

- Logic Gates
- Clock Pulse
- Bit Switches
- Bit Displays
- Connecting Wires

Software Used:

For Windows (64-bit):

- Download simulator: coldvl64Windows.jar
- Requires Java Runtime Environment (JRE).

Set PATH:
C:\Program Files\Java\<jdk_version>\bin;
Run using:
java -jar coldvl64Windows.jar

For Linux (64-bit):

java -jar coldvl64Linux.jar

For Windows/Linux (32-bit):

java -jar coaSimulatorNew.jar

Observations:

Truth tables for each flip-flop were verified:

- RS flip-flop: Set/Reset behavior confirmed.
- JK flip-flop: Toggle and race-around observed.
- D flip-flop: Output followed input.
- T flip-flop: Output toggled with every clock edge.
- Master-Slave JK: Eliminated race-around condition.

Result:

The synthesis of various flip-flops (RS, JK, D, T, Master-Slave JK) was successfully performed and verified.
The theoretical and simulated results matched.
The race-around condition in JK flip-flop was observed and resolved using the master-slave configuration.

Conclusion:

The experiment demonstrates the synthesis and functioning of various flip-flops, the occurrence of race-around condition, and how it is eliminated. It provides a fundamental understanding of sequential circuits and memory elements used in digital logic design.