SEQUENTIAL LOGIC CIRCUITS (FLIP-FLOP)

Difference between Combinational and Sequential Logic Circuits

1. Difference – Definition

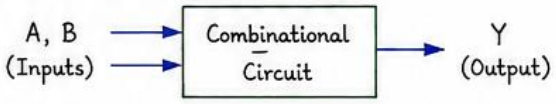
Combinational and Sequential logic circuits are the two main types of digital logic circuits. They differ mainly in terms of memory, output dependency and feedback.

Key Points :

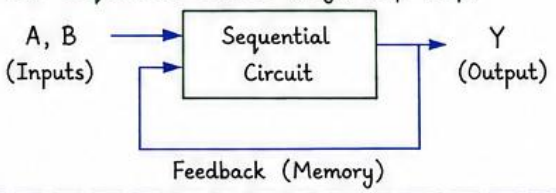
- Combinational circuits have no memory.
- Sequential circuits have memory.
- Output of combinational circuits depends only on present inputs.
- Output of sequential circuits depends on present inputs and past state.
- Sequential circuits use feedback.
- Flip-Flops are the basic building blocks of sequential circuits.

Example :

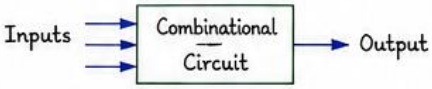
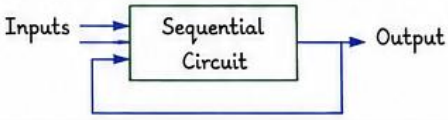
(i) Combinational Circuit (e.g., Adder)



(ii) Sequential Circuit (e.g., Flip-Flop)



2. Difference – Comparison Table

S. No.	Parameter	Combinational Logic Circuits	Sequential Logic Circuits
1.	Definition	Logic circuits whose output depends only on the present inputs.	Logic circuits whose output depends on present inputs as well as past state.
2.	Memory	No memory.	Has memory.
3.	Output Dependency	Output depends only on present inputs.	Output depends on present inputs and past state.
4.	Feedback	No feedback.	Uses feedback.
5.	Building Blocks	Logic gates (AND, OR, NOT, etc.) are used.	Flip-Flops (SR, JK, D, T) are used.
6.	Examples	Adder, Subtractor, Multiplexer, Decoder, Encoder, etc.	Counters, Registers, Shift Registers, Memory, etc.
7.	Clock Signal	Not required.	Usually required (for synchronized operation).
8.	Complexity	Generally less complex.	Generally more complex.
9.	Applications	Used in arithmetic and data processing circuits.	Used in control systems, memory devices, sequence generation, etc.
10.	State	No state.	Has one or more states.
3.	Boolean Expression	$Y = f$ (Present Inputs)	$Y = f$ (Present Inputs, Past State)
4.	Diagram		
5.	Description	The output is determined only by the current combination of inputs.	The output is determined by the current inputs and the previous state of the circuit.

★ **Note :** Combinational circuits are memoryless, while sequential circuits have memory and their behavior depends on time (past state).

SEQUENTIAL LOGIC CIRCUITS (FLIP-FLOP)

What is Flip-Flop? and Types of Flip-Flop

1. What is Flip-Flop? - Definition

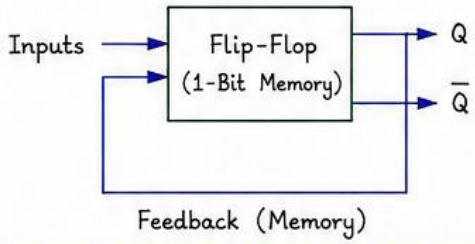
A Flip-Flop is a basic memory element in sequential logic circuits that can store one bit of information. It has two stable states: SET (1) and RESET (0). Flip-Flops use feedback to store information and their output depends on present input and past state.

Key Points :

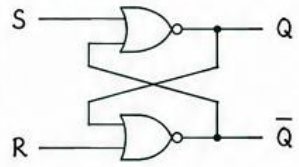
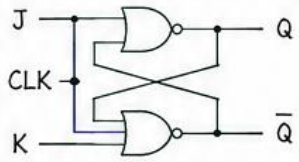
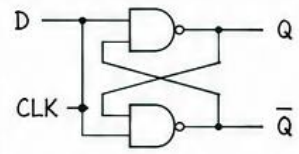
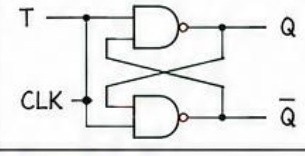
- Flip-Flop is a bistable circuit.
- It stores 1 bit of binary information (0 or 1).
- It has two outputs: Q and $\overline{Q}$ (complementary).
- The output remains stable until changed by input conditions.
- It is the basic building block of registers, counters and memory.

Example :

Flip-Flop as 1-Bit Memory



2. Types of Flip-Flop

S. No.	Type	Characteristic	Symbol	Inputs	Applications
1.	SR Flip-Flop (Set-Reset Flip-Flop)	• Simplest flip-flop. • Two inputs: S (Set), R (Reset). • Invalid condition when S = R = 1.		S, R	• Basic memory element. • Control circuits.
2.	JK Flip-Flop (J-K Flip-Flop)	• Improved SR flip-flop. • No invalid condition. • When J = K = 1, it toggles.		J, K, CLK	• Counters. • Registers. • Frequency dividers.
3.	D Flip-Flop (Data or Delay Flip-Flop)	• Has one input D (Data). • Output follows input after clock edge. • No invalid condition.		D, CLK	• Data storage. • Shift registers. • Pipelines.
4.	T Flip-Flop (Toggle Flip-Flop)	• Has one input T (Toggle). • When T = 0, No change. • When T = 1, Output toggles.		T, CLK	• Counters. • Frequency dividers. • Timing circuits.
5.	Output	Each flip-flop has two outputs: Q (Normal Output) and $\overline{Q}$ (Complement Output)			
6.	Note	All flip-flops are edge triggered (in practical circuits) using a Clock (CLK) signal to change state synchronously.			

★ Note : Flip-Flops are the heart of sequential circuits and are used to design registers, counters, memory units and digital systems.

SEQUENTIAL LOGIC CIRCUITS (FLIP-FLOP)

What is Latch? and Types of Latches

1. What is Latch? – Definition

A Latch is a basic memory element in sequential logic circuits that can store one bit of information. It is a level-sensitive bistable circuit, which means its output changes as long as the Enable (or Clock) signal is active.

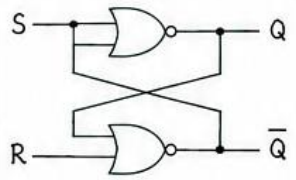
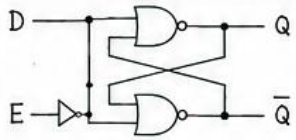
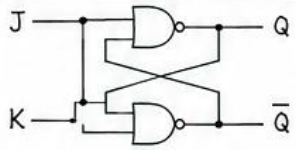
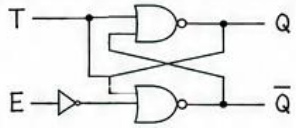
Key Points :

- Latch stores 1 bit of binary information (0 or 1).
- It has two stable states and two outputs: Q and $\overline{Q}$ (complementary).
- The output depends on input as long as enable signal is active.
- It is used in registers, buffers and data storage.
- Latch is level-sensitive, whereas Flip-Flop is edge-triggered.

Example :

Latch as 1-Bit Memory

2. Types of Latches

S. No.	Type	Characteristic	Logic Symbol	Inputs	Applications
1.	SR Latch (Set-Reset Latch)	• Most basic latch. • Two inputs: S (Set), R (Reset). • When S = 1 → Set (Q = 1) When R = 1 → Reset (Q = 0) • When S = R = 0 → No change. • S = R = 1 → Invalid.		S, R	• Basic memory storage. • Control circuits. • Switch debouncing.
2.	D Latch (Data Latch)	• Has one input D (Data) and an Enable (E). • When E = 1 → Q follows D. • When E = 0 → Output remains unchanged.		D, E	• Data storage. • Buffers. • Used in registers.
3.	JK Latch (Improved SR Latch)	• Improved version of SR latch. • Two inputs: J, K. • When J = K = 1 → Output toggles. • No invalid condition.		J, K	• Counters. • Registers. • Frequency dividers.
4.	T Latch (Toggle Latch)	• Has one input T (Toggle) and Enable (E). • When T = 0 → No change. • When T = 1 → Output toggles.		T, E	• Counters. • Frequency dividers. • Timing circuits.
5.	Output	All latches have two outputs: Q (Normal Output) and $\overline{Q}$ (Complement Output)			
6.	Note	Latch is level-sensitive. Its output changes as long as the Enable signal is active. It is usually used when data must be stored temporarily.			

★ **Note :** Latch is a fundamental building block of sequential circuits and is used in designing registers, counters and memory systems.

SEQUENTIAL LOGIC CIRCUITS (FLIP-FLOP)

Construction, Working, Circuit Diagram of RS Flip-Flop

1. Construction of RS Flip-Flop

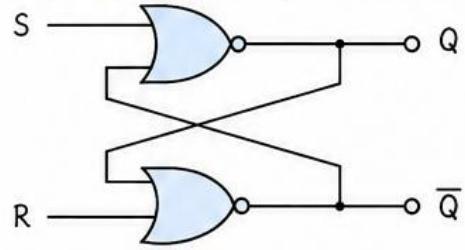
RS Flip-Flop (Reset-Set Flip-Flop) is the simplest flip-flop, constructed using two cross-coupled NOR gates (active-high) or two cross-coupled NAND gates (active-low). It has two inputs Set (S) and Reset (R), and two outputs Q and $\overline{Q}$.

Key Points :

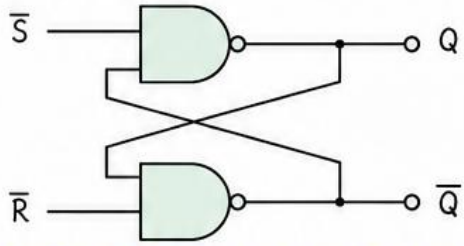
- RS Flip-Flop is constructed using two cross-coupled logic gates.
- It can be implemented using NOR gates or NAND gates.
- It has two inputs: Set (S) and Reset (R).
- It has two outputs: Q and $\overline{Q}$ (complement of Q).
- It is the basic building block for other flip-flops.

Circuit Diagram :

(i) RS Flip-Flop using NOR gates (active-high)



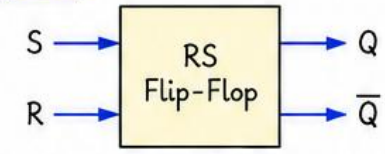
(ii) RS Flip-Flop using NAND gates (active-low)



2. Working of RS Flip-Flop

RS Flip-Flop has two inputs, Set (S) and Reset (R).
When S = 1, the flip-flop is set and Q becomes 1.
When R = 1, the flip-flop is reset and Q becomes 0.
When both inputs are 0, the previous state is maintained.
In case of NOR gate implementation, S = R = 1 is an invalid condition.

Symbol :



3. Truth Table

For NOR Gate Implementation (Active-High)		Output		Operation
S	R	Q (Next State)	$\overline{Q}$	
0	0	No change (Previous state)	No change (Previous state)	Hold (No change)
0	1	0	1	Reset
1	0	1	0	Set
1	1	Invalid condition	Invalid condition	Not allowed

For NAND Gate Implementation (Active-Low)		Output		Operation
$\overline{S}$	$\overline{R}$	Q (Next State)	$\overline{Q}$	
1	1	No change (Previous state)	No change (Previous state)	Hold (No change)
1	0	0	1	Reset
0	1	1	0	Set
0	0	Invalid condition	Invalid condition	Not allowed

4. Operation Summary

Operation	Input Condition (Active-High NOR)	Result	Description
Set	S = 1, R = 0	Q = 1, $\overline{Q}$ = 0	Sets the flip-flop (Q becomes 1).
Reset	S = 0, R = 1	Q = 0, $\overline{Q}$ = 1	Resets the flip-flop (Q becomes 0).
Hold	S = 0, R = 0	No change	Maintains the previous state.
Invalid	S = 1, R = 1	Invalid	Not allowed (both outputs become 0).

☆ **Note :** RS Flip-Flop is a basic memory element and forms the foundation for more advanced flip-flops like JK, D and T flip-flops.

SEQUENTIAL LOGIC CIRCUITS (FLIP-FLOP)

Construction, Working, Circuit Diagram of JK Flip-Flop

1. Construction of JK Flip-Flop

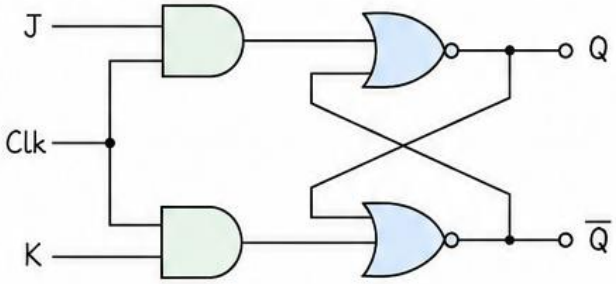
JK Flip-Flop is constructed by modifying the SR Flip-Flop to overcome the invalid condition (S = R = 1). It is implemented using either NOR gates (active-high) or NAND gates (active-low) along with a clock input.

Key Points :

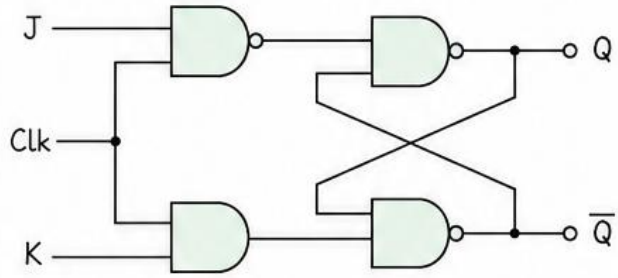
- JK Flip-Flop is an advanced form of SR Flip-Flop.
- It has two inputs: J and K, one clock input and two outputs Q and $\overline{Q}$.
- It eliminates the invalid condition of SR Flip-Flop.
- It can perform Set, Reset, Hold and Toggle operations.
- It is widely used in counters and registers.

Circuit Diagram :

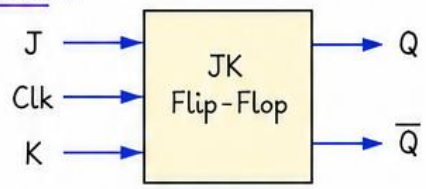
JK Flip-Flop using NOR gates (active-high)



JK Flip-Flop using NAND gates (active-low)



Symbol :



2. Working of JK Flip-Flop

JK Flip-Flop has three inputs: J, K and Clock. Its operation depends on the combination of J and K at the active clock edge. It removes the invalid condition of SR Flip-Flop. When J = K = 1, the output toggles (changes its state) at every clock pulse.

3. Truth Table

J	K	Q (Present State)	Q (Next State)	Operation
0	0	Q	Q	No Change (Hold)
0	1	Q	0	Reset
1	0	Q	1	Set
1	1	Q	$\overline{Q}$	Toggle

4. Operation Summary

J	K	Operation	Description
0	0	Hold	No change in the output (Q remains the same).
0	1	Reset	Output becomes 0.
1	0	Set	Output becomes 1.
1	1	Toggle	Output changes to its complement (Q becomes $\overline{Q}$).

★ **Note :** JK Flip-Flop is widely used in synchronous counters, frequency dividers and other sequential circuits.

SEQUENTIAL LOGIC CIRCUITS (FLIP-FLOP)

Construction, Working, Circuit Diagram of T Flip-Flop

1. Construction of T Flip-Flop

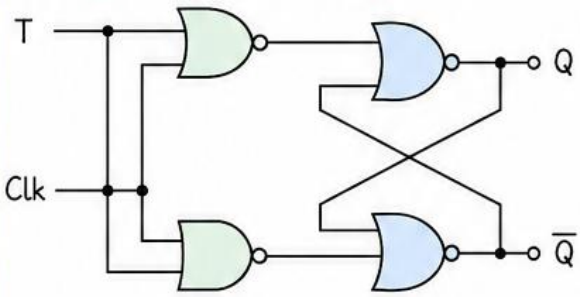
T Flip-Flop is constructed by modifying JK Flip-Flop by connecting J and K inputs together and using a single input T (Toggle). It can be implemented using either NOR gates (active-high) or NAND gates (active-low) along with a clock input.

Key Points :

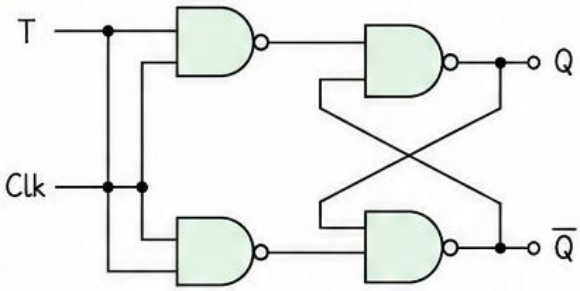
- T Flip-Flop is a modification of JK Flip-Flop.
- It has one input T and one clock input.
- It has two outputs Q and $\overline{Q}$ (complement of Q).
- When T = 0, it holds the previous state.
- When T = 1, it toggles the state (changes).
- It is widely used in counters and frequency dividers.

Circuit Diagram :

(i) Using NOR gates (active-high)



(ii) Using NAND gates (active-low)



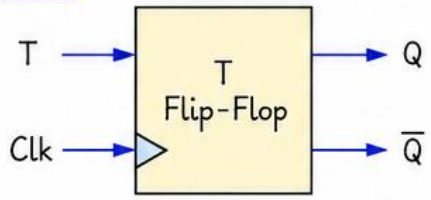
(All gates are NAND gates – active low)

2. Working of T Flip-Flop

T Flip-Flop has two inputs: T and Clock.
Its operation depends on the value of T at the active clock edge.

- When T = 0, the output state remains unchanged (Hold).
- When T = 1, the output state toggles (changes to its complement).
- Thus, it works as a Toggle flip-flop.

Symbol :



3. Truth Table

T	Q (Present State)	Q (Next State)	Operation
0	0	0	No Change (Hold)
0	1	1	No Change (Hold)
1	0	1	Toggle
1	1	0	Toggle

4. Operation Summary

T	Operation	Result	Description
0	Hold	$Q(t+1) = Q(t)$	Output remains the same.
1	Toggle	$Q(t+1) = \overline{Q(t)}$	Output changes to its complement.
X	No Change	No Change	When clock is not active.

★ **Note :** T Flip-Flop is mainly used in binary counters, frequency dividers and digital systems where toggling operation is required.

SEQUENTIAL LOGIC CIRCUITS (FLIP-FLOP)

Construction, Working, Circuit Diagram of D Flip-Flop

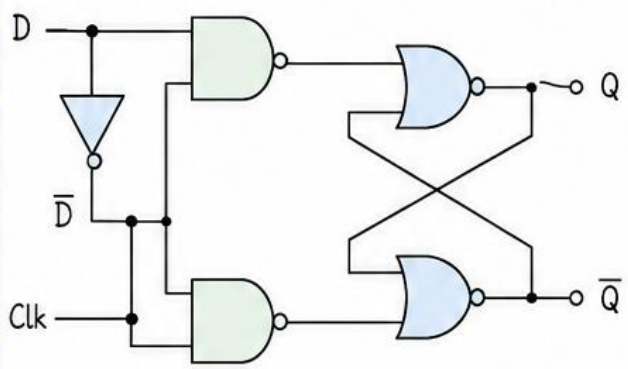
1. Construction of D Flip-Flop

D Flip-Flop is constructed from a gated SR Flip-Flop by connecting the input D to the Set input (S) and the complement of D ($\overline{D}$) to the Reset input (R) using an inverter.

Key Points :

- D Flip-Flop is a modification of SR Flip-Flop.
- It has one data input (D) and one clock input.
- It has two outputs Q and $\overline{Q}$ (complement of Q).
- It stores one bit of data.
- It changes the output only on the active clock edge.
- It is widely used in registers and shift registers.

Circuit Diagram :



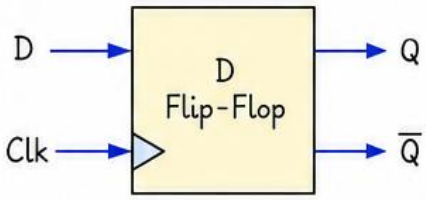
(All gates are NAND gates – active low)

2. Working of D Flip-Flop

D Flip-Flop copies the value of D at the time of the active clock edge and transfers it to the output Q.

- When D = 1, the output Q becomes 1.
- When D = 0, the output Q becomes 0.
- The output remains unchanged between clock pulses.
- It eliminates the invalid condition of SR Flip-Flop.
- The output Q follows the input D.

Symbol :



3. Truth Table

D	Q (Present State)	Q (Next State)	Operation
0	0	0	Reset
0	1	0	Reset
1	0	1	Set
1	1	1	Set

4. Operation Summary

D	Operation	Result	Description
0	Reset	Q = 0, $\overline{Q}$ = 1	The output is reset to 0.
1	Set	Q = 1, $\overline{Q}$ = 0	The output is set to 1.
X	No Change	Q(t+1) = Q(t)	Output remains the same between clock pulses.

★ **Note :** D Flip-Flop is a basic memory element and is mainly used in registers, shift registers and data storage applications.

SEQUENTIAL LOGIC CIRCUITS (FLIP-FLOP)

Master-Slave JK Flip-Flop

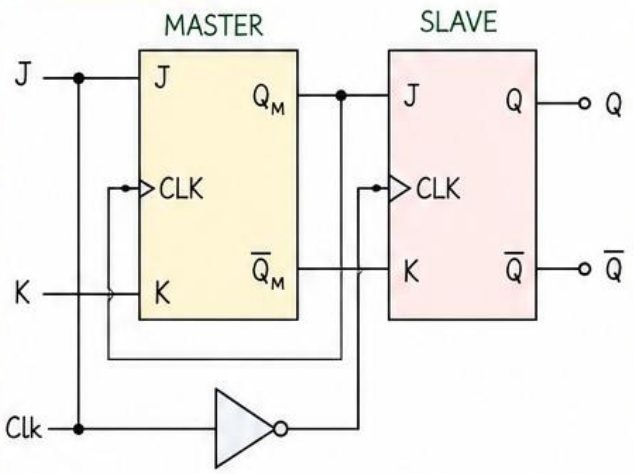
1. Construction of Master-Slave JK Flip-Flop

Master-Slave JK Flip-Flop is constructed by connecting two JK Flip-Flops in cascade. The first flip-flop acts as the Master and the second acts as the Slave. The clock input of the slave is complemented (inverted) so that when the master is enabled, the slave is disabled and vice-versa.

Key Points :

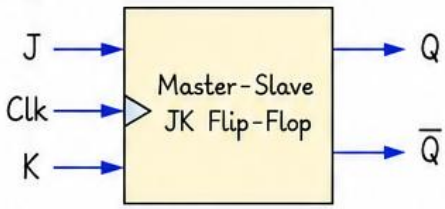
- It is formed by cascading two JK Flip-Flops.
- The Master receives the clock (Clk) input.
- The Slave receives the inverted clock ($\overline{Clk}$) input.
- Data is accepted by the Master when Clk = 1.
- Data is transferred to the Slave when Clk = 0.
- Output changes only on the falling edge of the clock (negative edge-triggered).
- It eliminates race-around condition.

Circuit Diagram :



(Edge Triggered – Negative Edge Triggered)

Symbol :



2. Working of Master-Slave JK Flip-Flop

When Clk = 1, the Master is enabled and accepts the inputs J and K, while the Slave is disabled and holds the previous state.

When Clk = 0, the Master is disabled and the Slave is enabled. The data stored in the Master is transferred to the Slave and appears at the output.

Hence, the output changes only on the falling edge (1 → 0 transition) of the clock.

3. Truth Table

J	K	During Clk = 1 (Master Active)		During Clk = 0 (Slave Active)		Operation
		Q_M (Next State) (Master Output)	Q_S (Output) (Slave)	Q_M	Q_S (Next State) (Output)	
0	0	Q (No Change)	Q (No Change)	Q (No Change)	Q (No Change)	No Change (Hold)
0	1	0	Q (Hold)	0	0	Reset
1	0	1	Q (Hold)	1	1	Set
1	1	$\overline{Q}$ (Toggle)	Q (Hold)	$\overline{Q}$	$\overline{Q}$	Toggle

Note : Q_M → Output of Master, Q_S → Output of Slave (Final Output).

4. Operation Summary

Clk	Master Flip-Flop	Slave Flip-Flop	Output (Q)	Remarks
Clk = 1 (High)	Active (Accepts J, K)	Disabled (Holds)	No Change	Master stores data
Clk = 0 (Low)	Disabled (Holds)	Active (Transfers)	Changes according to JK	Output updates
Falling Edge (1 → 0)	Data transfer	Output updated	Updated	Negative edge triggered

★ Note : Master-Slave JK Flip-Flop changes its output only on the falling edge of the clock. It eliminates race-around condition and is widely used in counters, registers and sequential circuits.

SEQUENTIAL LOGIC CIRCUITS (FLIP-FLOP)

Positive and Negative Edge Triggered Flip-Flop

1. Introduction

Edge triggered flip-flops change their output only at the instant of the active clock edge. There are two types :

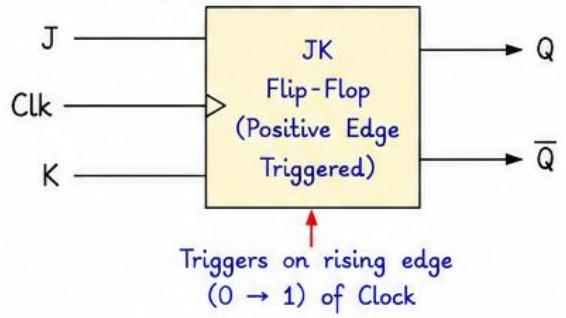
★ Positive Edge Triggered Flip-Flop (P-FF)
★ Negative Edge Triggered Flip-Flop (N-FF)

Key Points :

- In edge triggered flip-flops, data is transferred only at the clock edge (rising or falling).
- Positive edge triggered flip-flop responds to the rising edge (0 → 1) of the clock.
- Negative edge triggered flip-flop responds to the falling edge (1 → 0) of the clock.
- These flip-flops eliminate race-around condition.
- Widely used in registers, counters and synchronous sequential circuits.

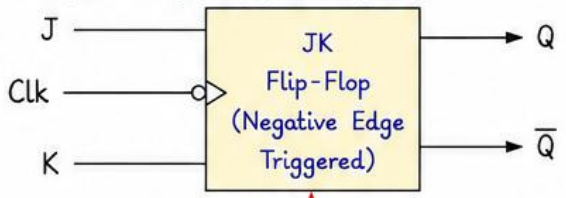
2. Circuit Diagrams

(i) Positive Edge Triggered Flip-Flop



Triggers on rising edge (0 → 1) of Clock

(ii) Negative Edge Triggered Flip-Flop



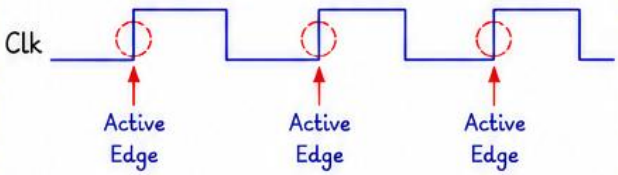
Triggers on falling edge (1 → 0) of Clock

3. Working

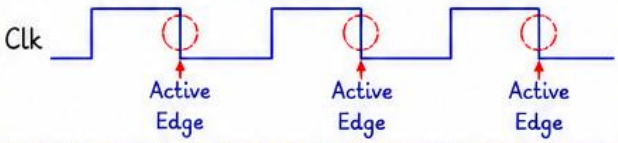
Positive Edge Triggered Flip-Flop (P-FF)	• When the clock has a low to high transition (0 → 1), the flip-flop gets triggered. • Inputs J and K are sampled at the rising edge. • The output Q changes according to J and K only at the rising edge. • Between two rising edges, output remains unchanged.
Negative Edge Triggered Flip-Flop (N-FF)	• When the clock has a high to low transition (1 → 0), the flip-flop gets triggered. • Inputs J and K are sampled at the falling edge. • The output Q changes according to J and K only at the falling edge. • Between two falling edges, output remains unchanged.

4. Clock Waveforms

(i) Positive Edge Triggered (Triggers on Rising Edge)



(ii) Negative Edge Triggered (Triggers on Falling Edge)



5. Truth Table (Applicable to both Positive and Negative Edge Triggered Flip-Flop)

J	K	Q (Present State)	Q (Next State)	Operation
0	0	Q	Q	No Change (Hold)
0	1	Q	0	Reset
1	0	Q	1	Set
1	1	Q	$\overline{Q}$	Toggle

6. Operation Summary

Type	Active Edge	Clock Transition	When Triggered	Output Change
Positive Edge Triggered (P-FF)	Rising Edge	0 → 1	At 0 → 1 transition	Changes according to J and K
Negative Edge Triggered (N-FF)	Falling Edge	1 → 0	At 1 → 0 transition	Changes according to J and K

★ **Note :** Edge triggered flip-flops provide precise timing control and are essential for the design of synchronous sequential circuits.

SEQUENTIAL LOGIC CIRCUITS (FLIP-FLOP)

Excitation / Transition Table of all Flip-Flops

1. Introduction

Excitation table (or Transition table) shows the input requirements for a flip-flop to make a transition from Present State (Q) to Next State (Q^+).

Key Points :

- It helps to determine the inputs needed for a desired state transition.
- 'X' indicates Don't Care condition.
- It is widely used in the design and simplification of sequential circuits.

Example :

Q = Present State

Q^+ = Next State

Inputs = Inputs required to achieve $Q \rightarrow Q^+$ transition

2. Excitation / Transition Tables of Various Flip-Flops

(i) SR Flip-Flop

Present State (Q)	Next State (Q^+)	Inputs		Operation
		S	R	
0	0	0	X	No Change (Hold)
0	1	1	0	Set
1	0	0	1	Reset
1	1	X	0	No Change (Hold)
0	0	0	0	Invalid
1	1	1	1	Invalid

(ii) JK Flip-Flop

Present State (Q)	Next State (Q^+)	Inputs		Operation
		J	K	
0	0	0	X	No Change (Hold)
0	1	1	X	Set
1	0	X	1	Reset
1	1	1	1	Toggle

(iii) D Flip-Flop

Present State (Q)	Next State (Q^+)	Input D	Operation
0	0	0	Reset
0	1	1	Set
1	0	0	Reset
1	1	1	Set

(iv) T Flip-Flop

Present State (Q)	Next State (Q^+)	Input T	Operation
0	0	0	No Change (Hold)
0	1	1	Toggle
1	0	1	Toggle
1	1	0	No Change (Hold)

3. Summary of All Flip-Flops

Flip-Flop	Number of Inputs	Inputs	No. of States	Special Feature	Used In
SR Flip-Flop	2	S, R	2	Not allowed when S = R = 1 (Invalid condition)	Basic memory element
JK Flip-Flop	2	J, K	2	No invalid state, when J = K = 1 it toggles	Counters, Registers
D Flip-Flop	1	D	2	No invalid state, Output follows input (D)	Registers, Shift Registers, Data storage
T Flip-Flop	1	T	2	T = 1 ⇒ Toggle, T = 0 ⇒ Hold	Counters, Frequency dividers

★ **Note :** Excitation tables are very useful in designing sequential circuits, state machines, counters, and in minimizing the logic required for state transitions.