Chapter 5a: Sequential Circuits

Objectives of this chapter:

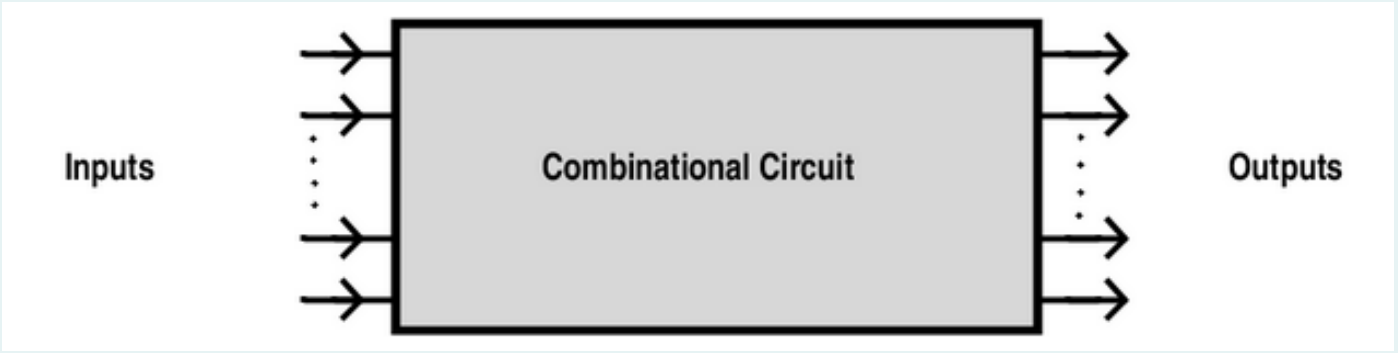
1. Understand the difference between combinatorial and sequential circuits.
2. Examine the different sequential circuits such as;
    - Flip flops
    - Registers
    - Counters

SEQUENTIAL LOGIC & COMPUTER CIRCUITS

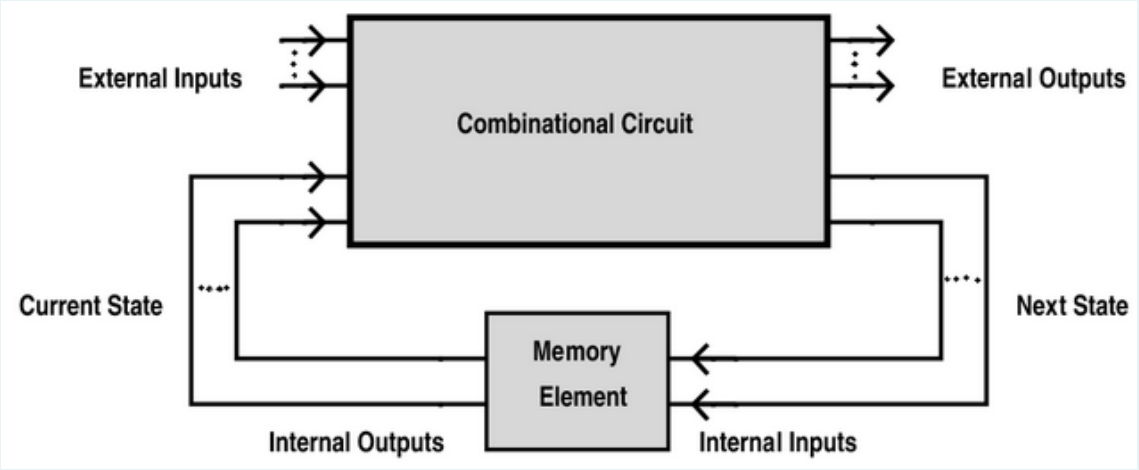
- In sequential circuits, the output depends on the current input as well as past output/outputs (current state).
- Whereas in combinatorial circuits the output depends only on the present values of the input, at any instant of time.
- Sequential circuit can be considered as combinational circuit with feedback circuit.
- Sequential circuit uses a memory element as feedback circuit in order to store past values.
- The information stored in sequential circuits represents current state of the circuit.
- The current state and current input will define output and the next state of the circuit.

SEQUENTIAL LOGIC & COMPUTER CIRCUITS

1. Combinatorial circuits



2. Sequential circuits



SEQUENTIAL LOGIC & COMPUTER CIRCUITS

- Combinatorial circuits cannot be used for storage because:
  - They have no memory
  - They do not contain feedbacks
  - They are time independent.
  - Their outputs solely depend on their current inputs.
  - They have no intrinsic timing control.

- Sequential circuits **have internal states** that can be used to store information and modify their inputs.

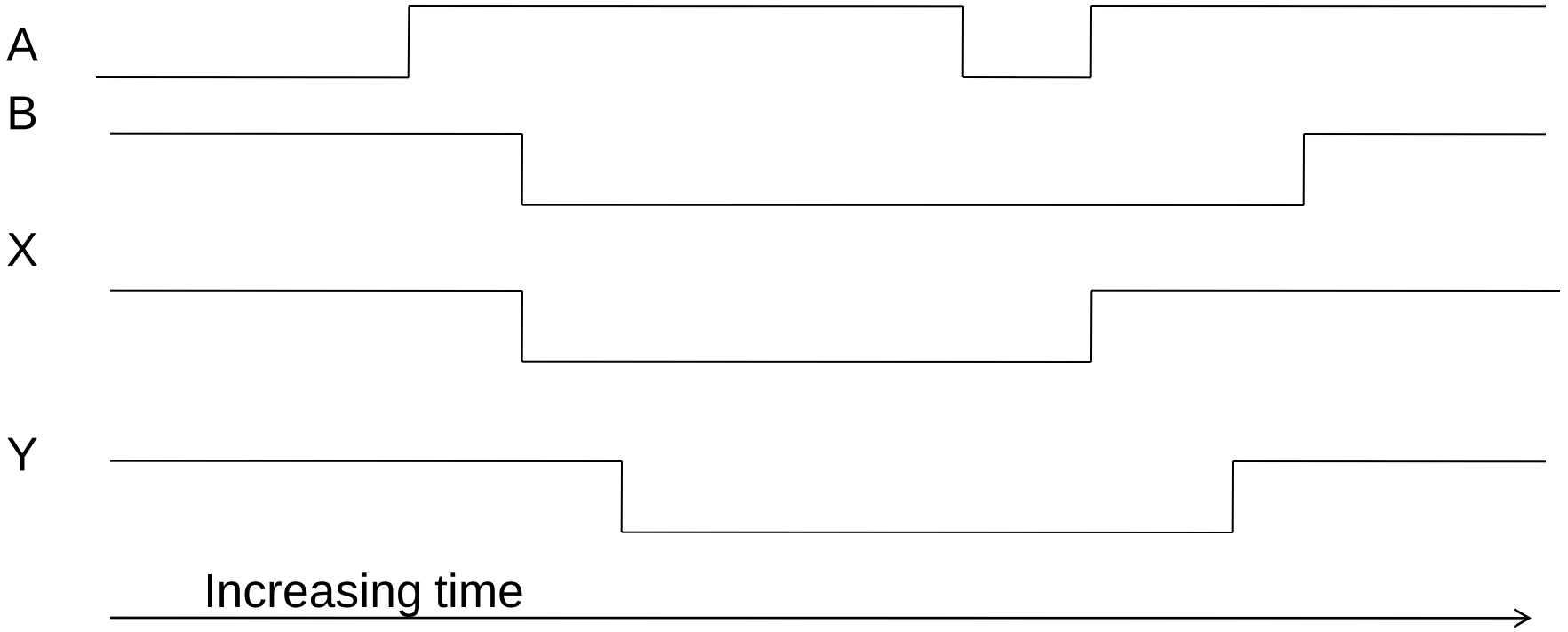
SEQUENTIAL LOGIC & COMPUTER CIRCUITS

- **Examples**
  - **A Counting Device**: Generally, we come across many counters in our daily life to count the number of objects. For example to count the number of audience entering or leaving an auditorium or to count number of vehicles in parking. In this when any person enters in to auditorium the counter increments its value depending on its present value. Similarly, it decrements its value depending on its previous and present value. So Counter retains the current state of the counter to do next operation.
  - **A memory circuit**; the input causes the contents of memory to be applied to the outputs.

SEQUENTIAL LOGIC & COMPUTER CIRCUITS

- Time in a sequential circuit takes on a significant role.
- A sequential network is defined according to its inputs and outputs over a period of time.
- In sequential networks, a truth table is not enough to show input/output relationships because it ignores time.
- The aid used in examining the time dependent aspect of a sequential network is called a **timing diagram**

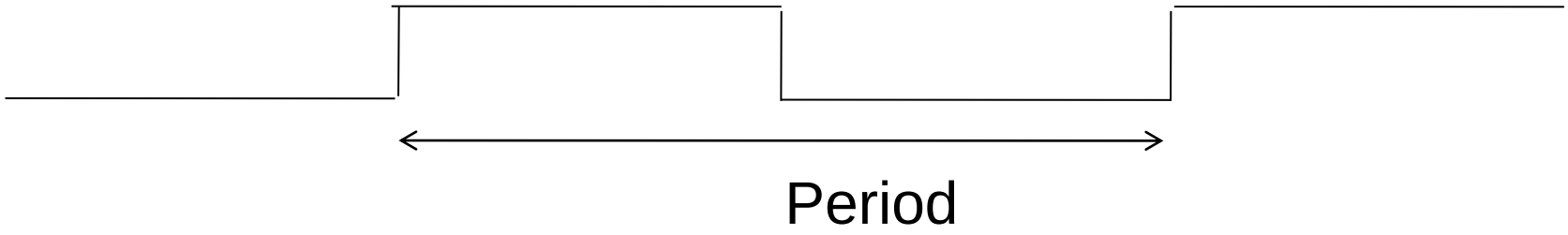
A Timing Diagram



- The higher value on the vertical state corresponds to state 1 and lower value corresponds to state 0.
- A begins in state 0, B and X begin in state 1.
- Transition of A to 1 does not change X.
- Transition of B from 1 to 0 causes X to change to 0.
- Transition of A from 1 to 0 does not affect X but the following transition of A causes X to change to 1.
- Output Y is shown to change states 25 μs after a change in X.

Timing Diagrams

- Sequential networks are synchronized by a time standard called a **clock**.
- A clock is a signal that oscillates between 1 and 0 state and is used to coordinate actions of a digital circuit.
- A clock generates an evenly spaced train of pulses.
- Time between consecutive pulses is called a **period**. The number of periods per second is called a **frequency.**



- Elementary sequential circuits fall into a class of binary electronic circuits known as **multivibrators** which may be astable, monostable or bistable

Multivibrators

- **Astable multivibrators**

  They cannot maintain a fixed state but they keep on switching back and forth between their states.

- **Monostable multivibrators**

  They can take on two states but are stable in only one of them. They can only temporarily stay in the unstable state.

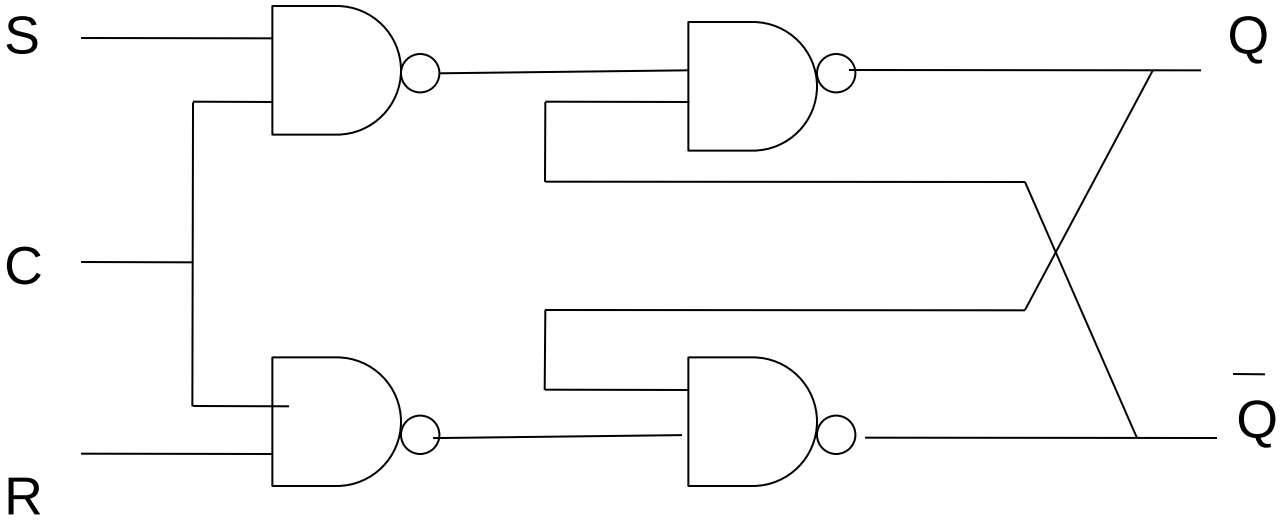
- **Bistable**

  They are stable in either of the 2 states and can therefore maintain either state indefinetely.

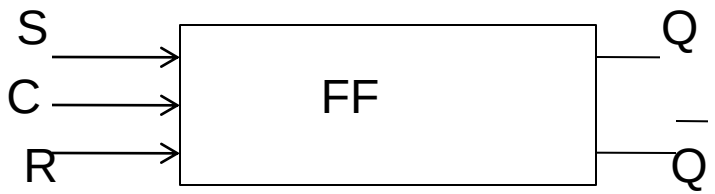
FLIP FLOPS

- They are bistable devices that are used in sequential networks.
- The most common flip-flops are the R-S, J-K, T, and the D flip flop.

THE R-S FLIP FLOP



S	R	Q+	$\overline{Q}$+
0	0	Q^-	$\overline{Q}^-$
0	1	0	1
1	0	1	0
1	1	-	-



THE R-S FLIP -FLOP

- It has 3 inputs S(Set), R (Reset) and C (a clock input) which synchronises the action of the flip-flop with its surrounding

- The two outputs (Q and $\overline{Q}$) are always in opposite states from each other.
- Most significant changes occur when there is a clock transition.
- If the clock input is constant, the outputs will follow the changes in the inputs at all times.
- When the clock is in 0 state Both R and S inputs have no effect on the state of the flip-flop. The network is then stable.

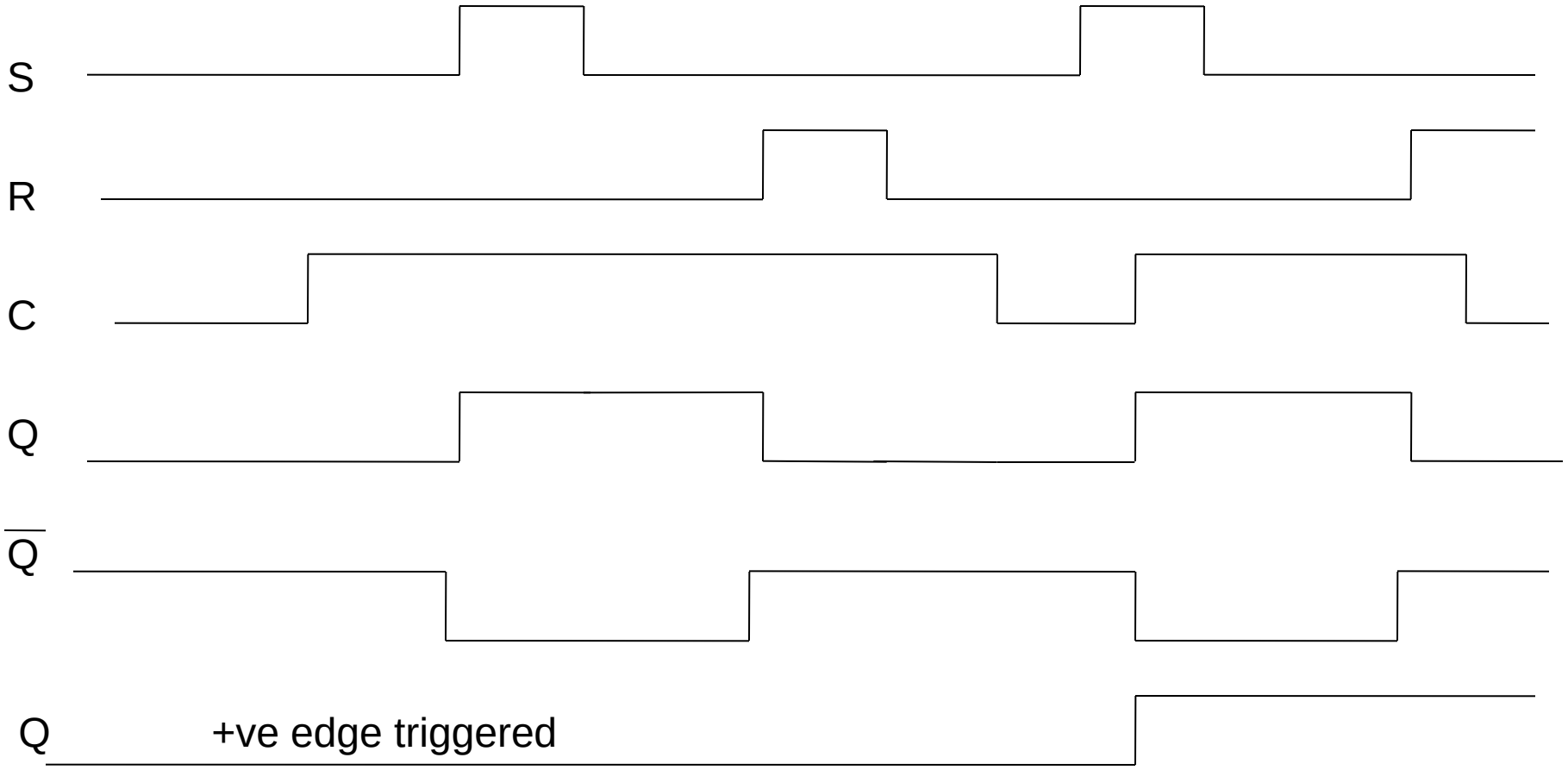
  In this state, if Q = 1, $\overline{Q}$ is = 0 and Q is maintained at 1. If $\overline{Q}$ is 1, Q = 0 and $\overline{Q}$ is maintained at 1.
- If the clock is raised to 1 the network will not change if R = S = 0

- The subscripts Q^- and $\overline{Q}^-$ indicate outputs just before the clock becomes 1 and the subscripts Q^+ and $\overline{Q}^+$ show outputs just after the clock becomes 1

Flip Flops

- All clocked flip flops that react to their inputs anytime C = 1 are called **latches**.
- The R-S flip-flop is called a latch because it uses the clock inputs to determine whether or not the inputs will be recognised.
- If a flip-flop changes only at the very beginning (or the very end) of a clock pulse it is called an **edge triggered** flipflop. They change state only when there is a 0 to 1 transition at C (+ve edge triggered) or a 1 to 0 transition at C (-ve edge triggered).
- A change from 0 to 1 is a +ve transition and the **+ve transition of a clock pulse is called the leading edge.**
- A change from 1 to 0 is a negative transition and the negative transition of a clock is called a **trailing edge**.

Example



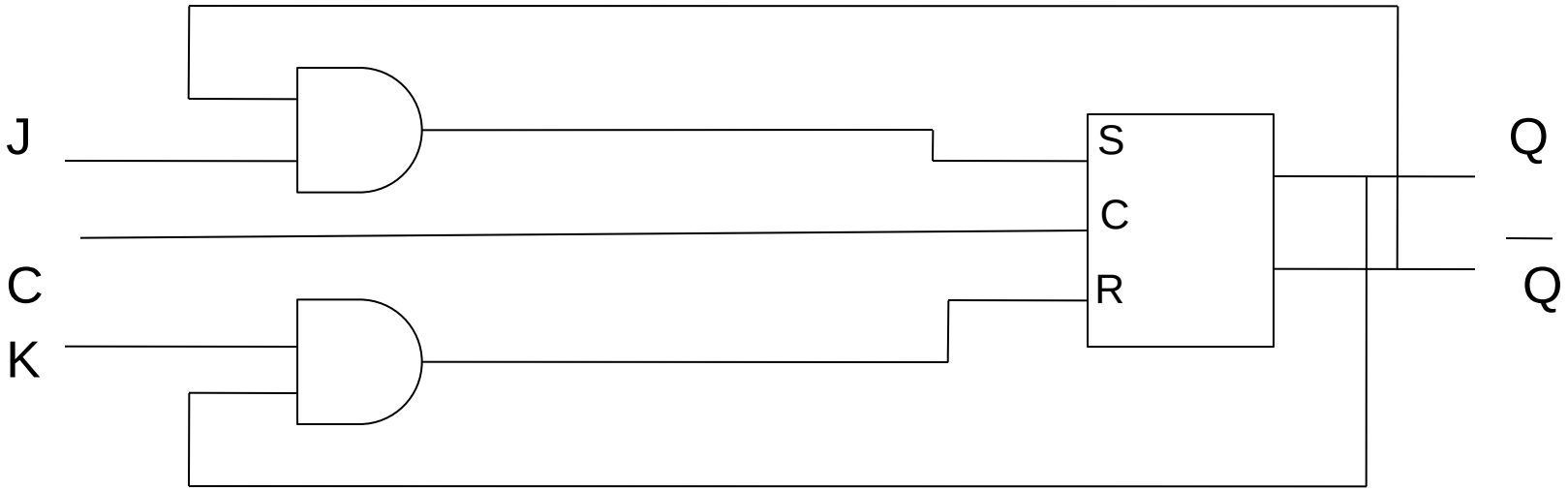
The J-K Flip Flop

- It is an R-S flip flop that has been modified by feeding the outputs back and ANDYING them with the inputs.

- It has the same behaviour like the R-S flip flop except that the C = J = K = 1 combination is meaningful and the results in the output states is reversed.

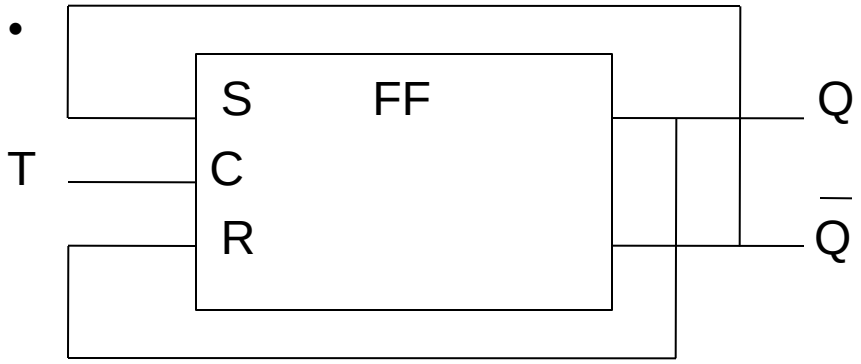
- The J-K flip –flop is constructed from an edge triggered R-S flip-flop otherwise the C=J=K=1 state would be unstable.

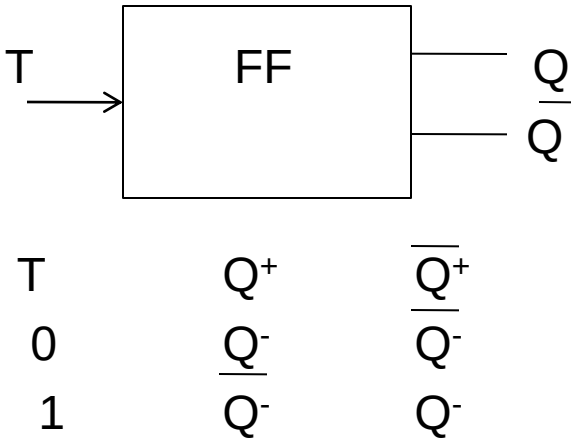
J-K Flip Flop



J	K	Q^+	$\overline{Q}^+$
0	0	Q^-	$\overline{Q}^-$
0	1	0	1
1	0	1	0
1	1	$\overline{Q}^-$	Q^-

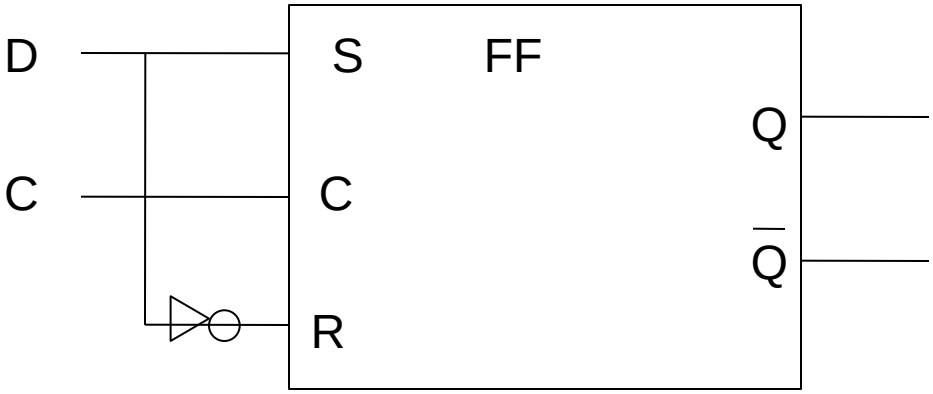
T Flip-Flop

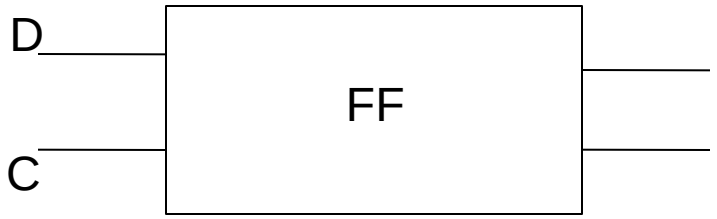




- It has only one input.
- Its output states are reversed each time the input is pulsed.
- It is used in the design of counters.
- A T flip flop can be obtained from a J-K flip flop by permanently applying 1's to the J and K inputs. The R-S flip flop must be edge triggered, otherwise the network would be unstable.

D FLIP FLOP



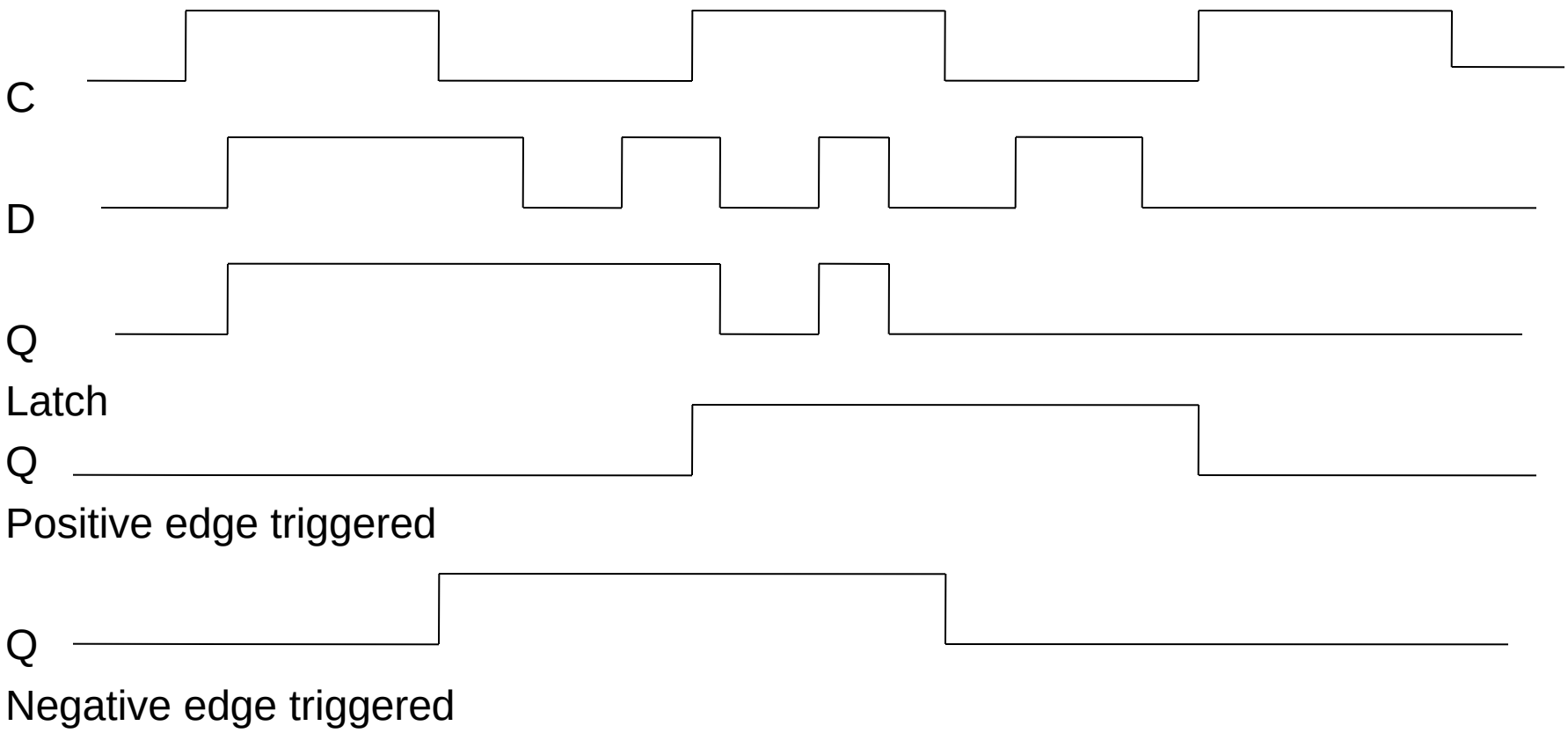


D	Q+	$\overline{Q}$+
0	0	1
1	1	0

- It has 2 inputs, a clock input and an input labelled D such that the Q output is equal to the D input whenever the clock input is set to 1; otherwise it is not affected by the D input.

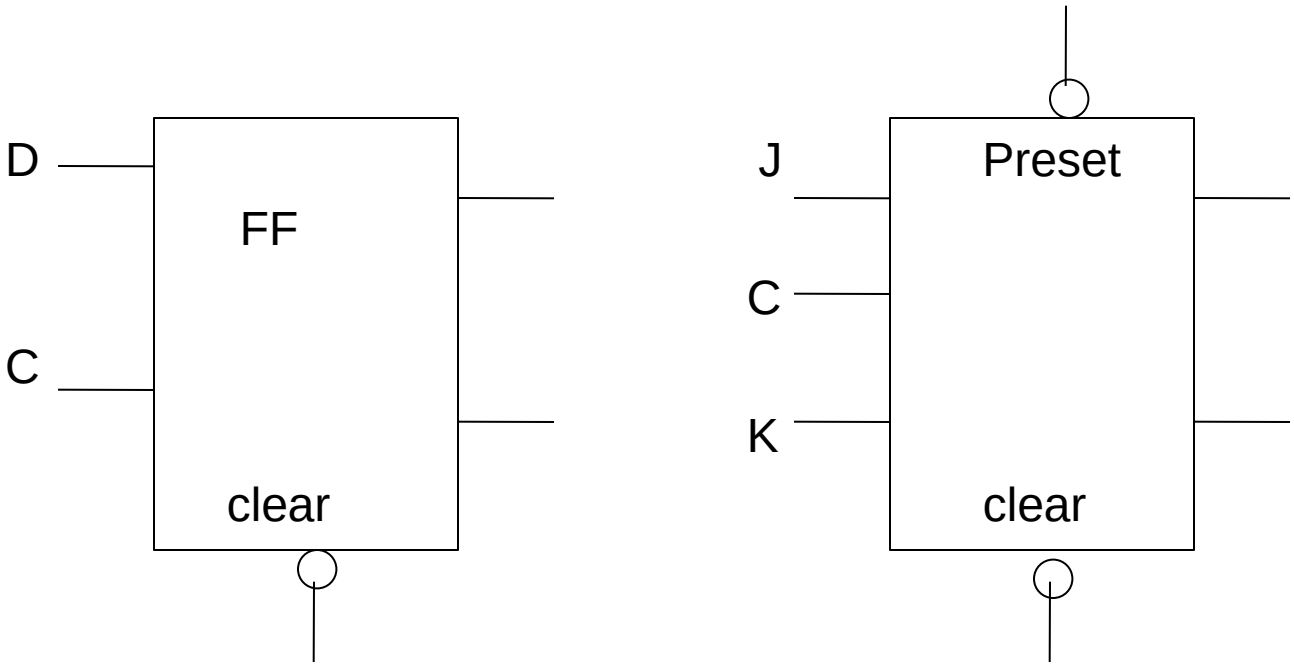
- It is used in constructing registers . It is easily constructed from the R-S flip flop by letting the D input be S input and connecting R to D through an inverter.

EXAMPLE


C
D
Q
Latch
Q
Positive edge triggered
Q
Negative edge triggered

Clear and Preset Inputs

- Flip flops can clear or set the Q output irrespective of the state of the other inputs. The clear input clears Q and the Preset input sets Q



Try this out!!!

1. The Timing diagrams below show inputs for the J-K flip-flop. Give corresponding Q outputs assuming that it is:

(i)An Ordinary Flip-flop (a latch)

(ii) A positively edge triggered flip-flop

(iii) A negatively edge triggered flip-flop

•Assume in all cases that the initial state of Q is 0.