The material covered in this lecture will be as follows:

⇒Sequential Circuits.

⇒Basic flip-flops (Latches).

⇒SR latch with control input.

⇒D latch

After finishing this lecture, you should be able to:

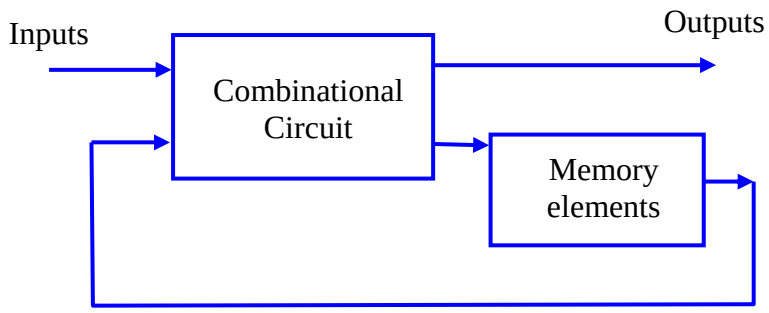
⇒ Differentiate between combinational and sequential circuits.

⇒ Understand the basic operation of flip flops.

⇒ Appreciate the function of the basic flip flop as a memory element.

## Sequential Circuits

The digital circuits studied so far were combinational circuits. Sequential circuits are very important class of digital circuits. They differ from combinational circuits by having memory elements as shown in the following block diagram.



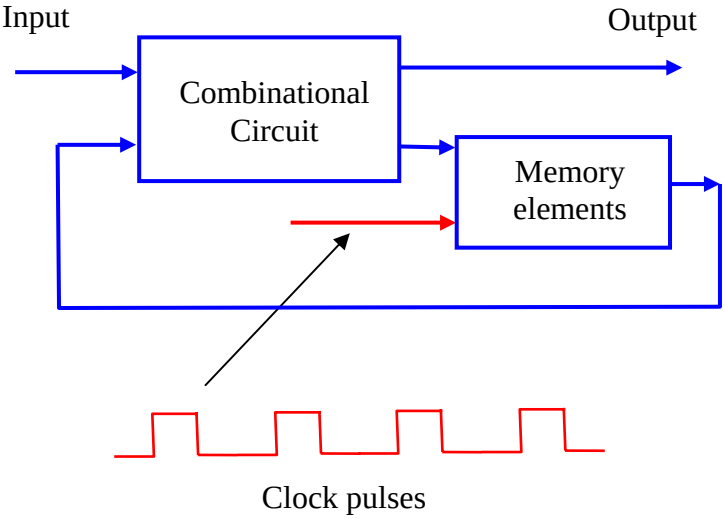

- The external outputs of the sequential circuit are functions of the external inputs and the present state of the memory elements.
- The binary information stored in the memory elements at any given time defines the state of the sequential circuit.

There are two types of sequential circuits depending on the timing of their signals:

- Synchronous Sequential Circuits – behaviour is defined from the knowledge of signals at discrete instants of time.
- Asynchronous Sequential circuits – behaviour depends on the order of change of input signals and can be affected at any instant of time.

Synchronous sequential circuits that use clock pulses at the input of the memory elements are called clocked sequential circuits. These are the most common circuits and are the type studied in

this course. A block diagram of such circuits is shown




next.

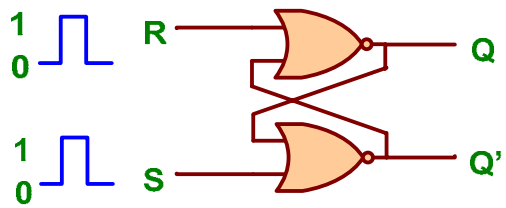
## Basic Flip-Flops

These are also called latches. They are the basic building blocks of more practical types of flip-flops. The basic flip-flop can be constructed from a pair of NOR gates or a pair of NAND gates as described below.

The basic SR flip-flop (latch) using two NOR gates is shown below with its truth table that describes its behaviour.

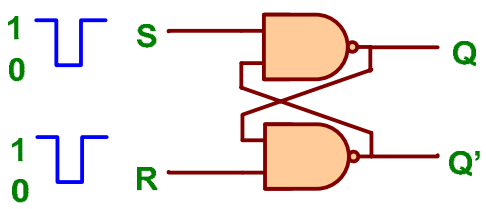
## Basic SR Flip – Flops

1. Using NOR gates:



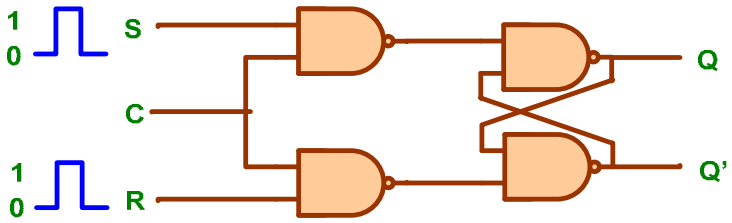

| S | R | Q | Q' | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | After (0,1) |
| 0 | 0 | 1 | 0 | After (1,0) |
| 0 | 1 | 0 | 1 | |
| 1 | 0 | 1 | 0 | |
| 1 | 1 | 0 | 0 | Unstable |

2. Using NAND gates:




| S | R | Q | Q' | |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | Unstable |
| 0 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 1 | |
| 1 | 1 | 1 | 0 | After (1,0) |
| 1 | 1 | 0 | 0 | After (0,1) |

## SR Flip-Flop with Control Input




The characteristic table shows the changes of the output for different values of the S and R inputs when the control input is set to 1. When C is 0, then no change for the outputs Q and Q'. When C becomes 1 then the new Q will depend on the values of S and R. This is shown in the following characteristic table.

| S | R | $Q_t$ | $Q_{t+1}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | Indeterminate |
| 1 | 1 | 1 | Indeterminate |

This characteristic table can also be shown in a compact form,

For C=1: →

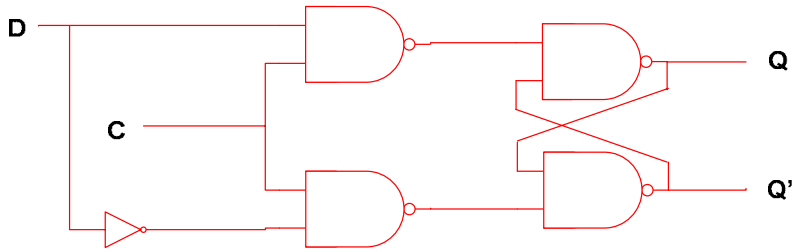
| S | R | $Q_{t+1}$ | |
|---|---|---|---|
| 0 | 0 | $Q_t$ | No change |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |
| 1 | 1 | 0 | Indeterminate |

Where $Q_t$ is the present state of Q,

And $Q_{t+1}$ is the next state of Q.

## D Latch

To eliminate the undesirable condition of the indeterminate state in the SR latch is to ensure that the S and R are never equal to 1 at the same time. This is achieved in the D latch, by having one input D to replace the S input, and connect D' instead of R. This is shown in the following logic diagram of the D latch.




When C=1:

| D | $Q_{t+1}$ | |
|---|---|---|
| 0 | 0 | Reset |
| 1 | 1 | Set |