Difference between combinational and sequential circuit

Combinational circuits are defined as the time independent circuits which do not depends upon previous inputs to generate any output are termed as combinational circuits. **Sequential circuits** are those which are dependent on clock cycles and depends on present as well as past inputs to generate any output.

Combinational Circuit –

1. In this output depends only **upon present input.**
2. Speed is fast.
3. It is designed easy.
4. There is no feedback between input and output.
5. This is time independent.
6. Elementary building blocks: Logic gates
7. Used for arithmetic as well as Boolean operations.
8. Combinational circuits don't have capability to store any state.
9. As combinational circuits don't have clock, they don't require triggering.
10. These circuits do not have any memory element.
11. It is easy to use and handle.

Examples – Encoder, Decoder, Multiplexer, Demultiplexer

Block Diagram –

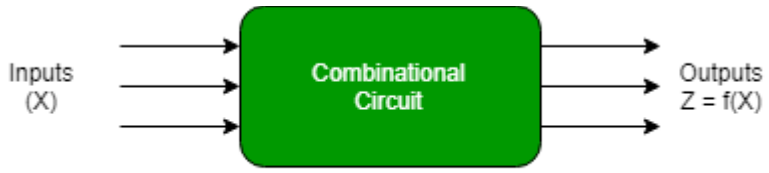


Figure: Combinational Circuits

Sequential Circuit

1. In this output depends upon present as well as past input.
2. Speed is slow.
3. It is designed tough as compared to combinational circuits.
4. There exists a feedback path between input and output.
5. This is time dependent.
6. Elementary building blocks: Flip-flops
7. Mainly used for storing data.
8. Sequential circuits have capability to store any state or to retain earlier state.
9. As sequential circuits are clock dependent, they need triggering.
10. These circuits have memory element.
11. It is not easy to use and handle.

Examples – Flip-flops, Counters

Block Diagram

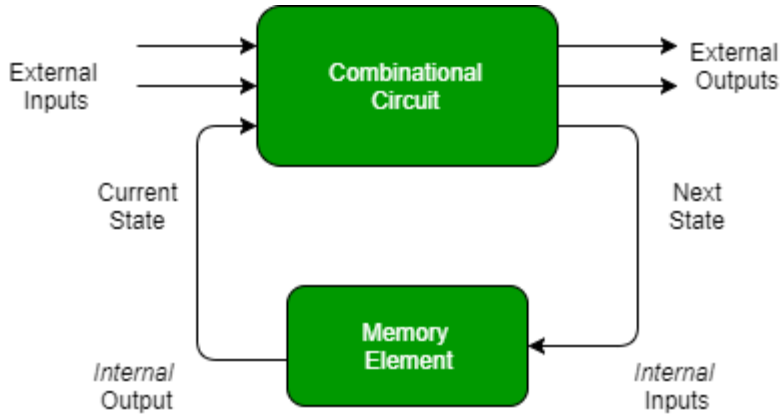


Figure: Sequential Circuit

Basic Flip-Flop Circuit

A flip-flop circuit can be constructed from two NAND gates or two NOR gates. These flip-flops are shown in Figure 2 and Figure 3. Each flip-flop has two outputs, *Q* and *Q'*, and two inputs, *set* and *reset*. This type of flip-flop is referred to as an *SR flip-flop* or *SR latch*. The flip-flop in Figure 2 has two useful states. When Q=1 and Q'=0, it is in the *set state* (or 1-state). When Q=0 and Q'=1, it is in the *clear state* (or 0-state).

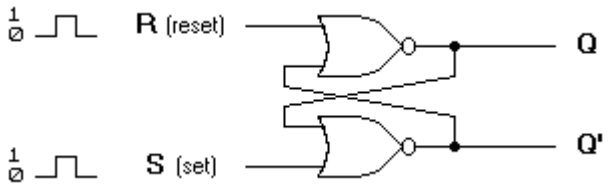
The outputs Q and Q' are complements of each other and are referred to as the normal and complement outputs, respectively. The binary state of the flip-flop is taken to be the value of the normal output. When a 1 is applied to both the set and reset inputs of the flip-flop in Figure 2, both Q and Q' outputs go to 0. This condition violates the fact that both outputs are complements of each other. In normal operation this condition must be avoided by making sure that 1's are not applied to both inputs simultaneously.

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after S=1, R=0)
0	1	0	1	
0	0	0	1	(after S=0, R=1)
1	1	0	0	

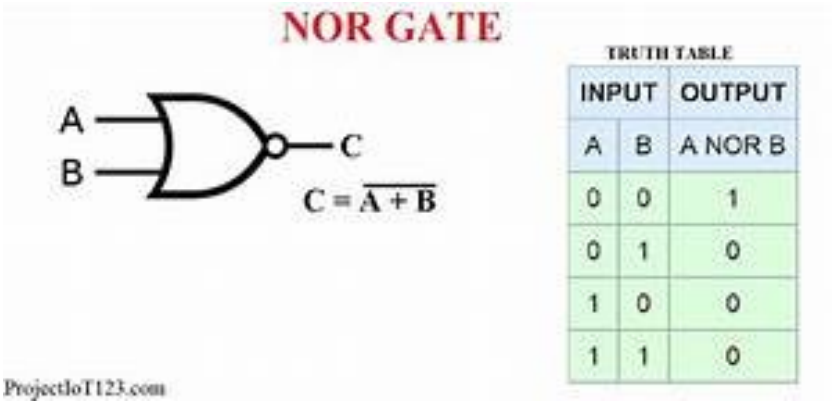
(b) Truth table for S-R LATCH

Figure 2. Basic flip-flop circuit with NOR gates

The basic storage element is called LATCH (0 or 1), If we can store, I bit (0 or 1) so we can store 10011 for example.



(a) Logic diagram



Remember Set means Q=1 and Reset means Q=0

Starting case is S&R are low means 0

1) Because the Q is 1 so LATCH can store 1
2) Apply pulse to input R to reset
3) S & R at (1,1) cannot be determined (not valid)

S	R	Q	Q’
0	0	1	0
		0	1
	1	0	1
1	0	1	0
1	1	0	0

ACTIVE HIGH SR LATCH: means normal condition for S and R are low, high pulse to input S to make output Q at 1 and high pulse to input R to Reset Q to 0.

S-R Latch, Using NAND Gates

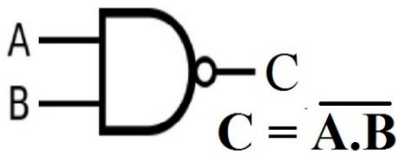
S	R	Q	Q'
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0	1

NAND gates on ACTIVE LOW SR Latch Normally High

So, in order to Set or Reset only by low R&S inputs.

1 0

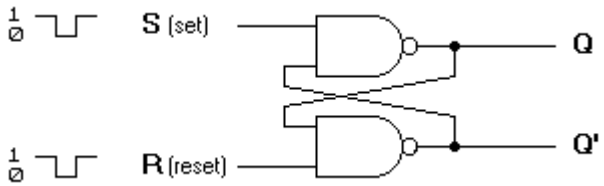
NAND GATE



Truth Table

INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

ProjectIoT123.com



(a) Logic diagram

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after S=1, R=0)
0	1	1	0	
1	1	1	0	(after S=0, R=1)
0	0	1	1	

(b) Truth table

Figure 3. Basic flip-flop circuit with NAND gates

The NAND basic flip-flop circuit in Figure 3(a) operates with inputs normally at 1 unless the state of the flip-flop has to be changed. A 0 applied momentarily to the set input causes Q to go to 1 and Q' to go to 0, putting the flip-flop in the set state. When both inputs go to 0, both outputs go to 1. This condition should be avoided in normal operation.

Clocked SR Flip-Flop (Gated)

In the field of electronics, a gated latch is a latch that has a third input that must be **active** in order for the SET and RESET inputs to take effect. This third input is sometimes **called ENABLE** because it enables the operation of the SET and RESET inputs. The ENABLE input can be connected to a simple switch.

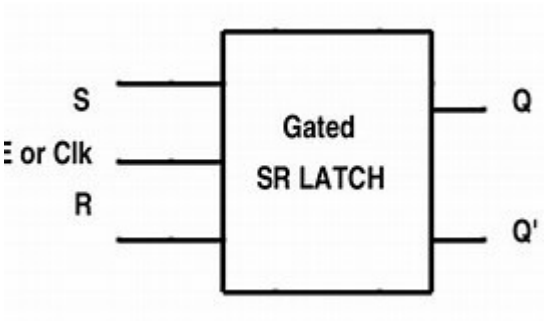
Active High means needs 1 for S to have Q=1 or High

1 For R to have Q=1 or High

Can be built by either another pair of AND OR NAND gates .

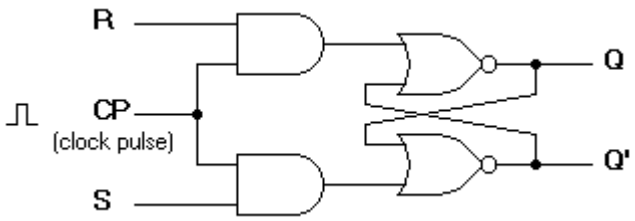
Gated SR Latch has an additional input E

Which must be high before the latch will respond



The clocked SR flip-flop shown in Figure 4 consists of a basic NOR flip-flop and two AND gates. The outputs of the two AND gates remain at 0 as long as the clock pulse (or CP) is 0, regardless of the S and R input values.

When the clock pulse goes to 1, information from the S and R inputs passes through to the basic flip-flop. With both S=1 and R=1, the occurrence of a clock pulse causes both outputs to momentarily go to 0. When the pulse is removed, the state of the flip-flop is indeterminate, i.e.., either state may result, depending on whether the set or reset input of the flip-flop remains a 1 longer than the transition to 0 at the end of the pulse.



a) Logic diagram

Q	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	indeterminate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	indeterminate

(b) Truth table

Figure 4. Clocked SR flip-flop