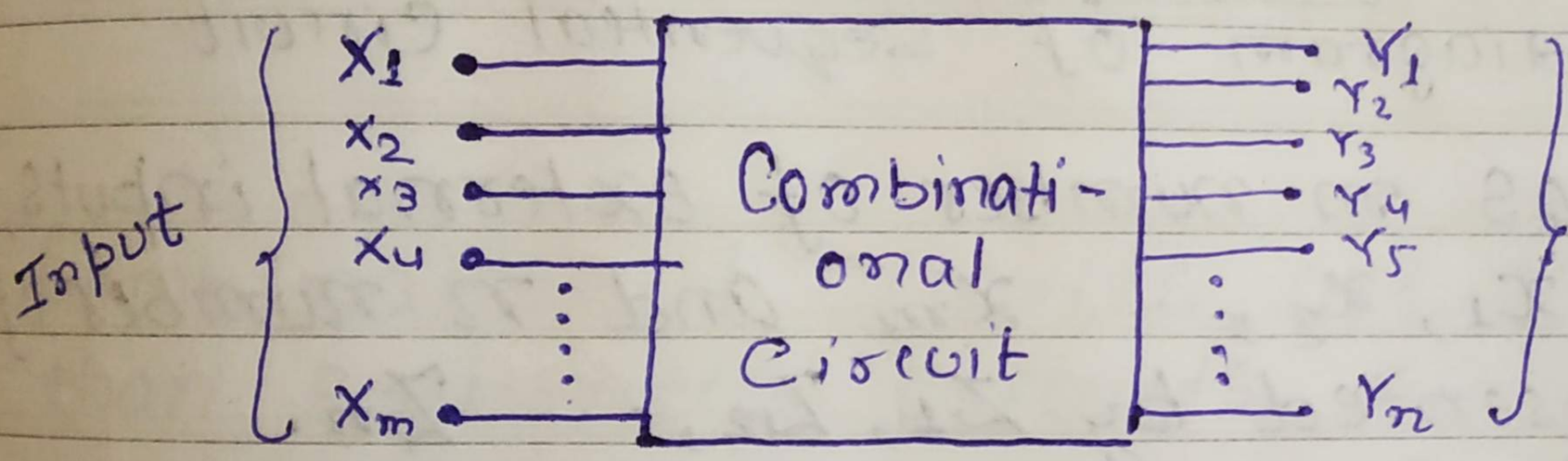
Sequential Circuit

The logic circuits whose output at any instant of time depend only on the input signals present at that time are known as combinational circuits.

The output signals of combinational circuit are not fedback to the input of the circuit.

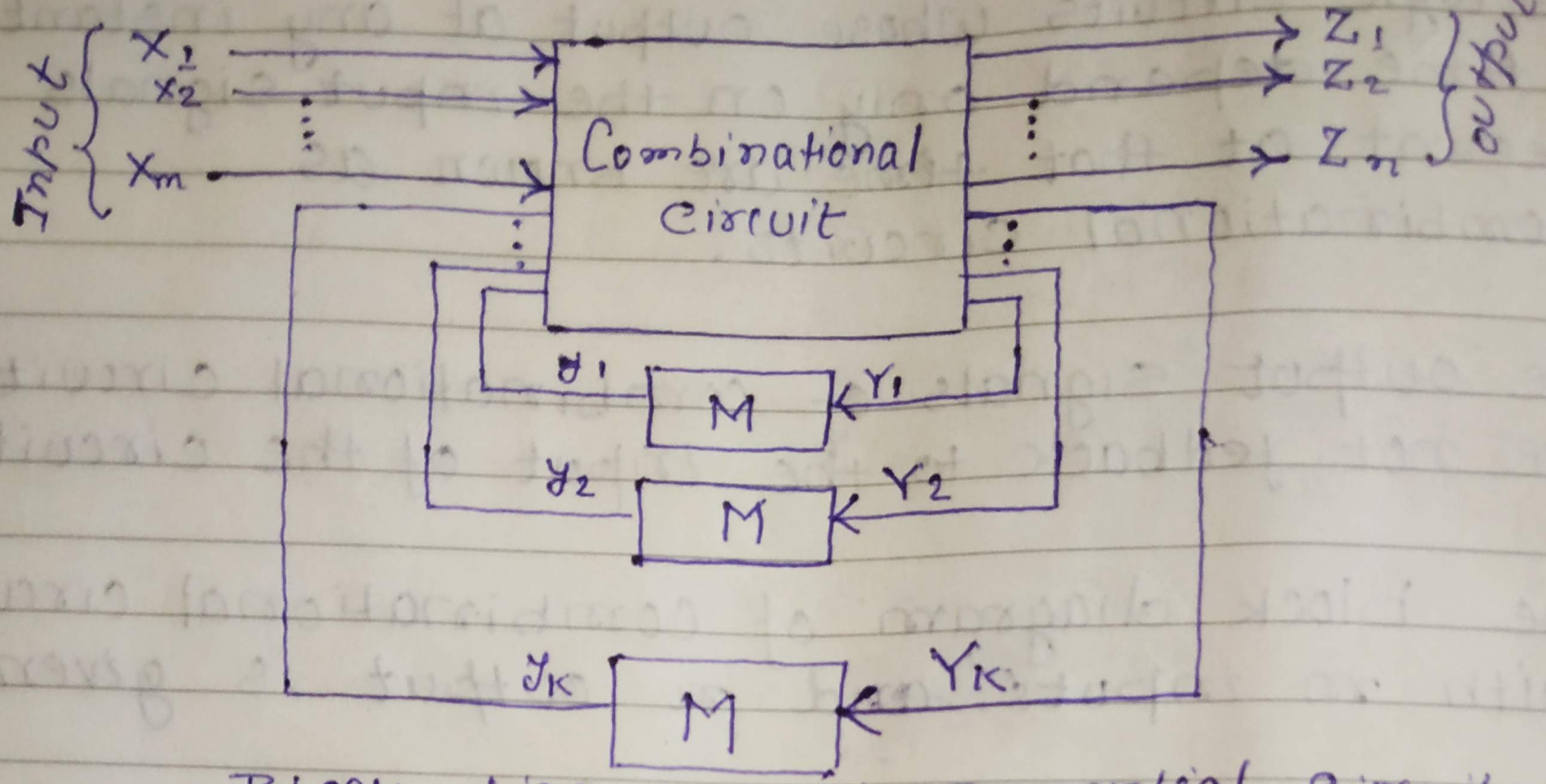
The block diagram of combinational circuit with m inputs and n output is given as



The logic circuit whose output at any instant of time depend not only on the present input but also on the past output are called sequential circuits.

An output signal is a function of the present input signals and a sequence of the past input signals i.e the past output signals.

The block diagram of a sequential circuit is given as



Block diagram of sequential circuit

The circuit has m number of external inputs, denoted by $x_1, x_2, \ldots x_m$ and n number of output, represented by $Z_1, Z_2, \ldots Z_n$.

The memory elements denoted by M are device capable of storing binary information within them.

The signal value at the output of memory elements, denoted by $y_1, y_2, \ldots y_k$ are referred to as the present state or simply the state of the sequential circuit.

The external inputs $x_1, x_2 \ldots x_m$ and the present state variables $y_1, y_2 \ldots y_k$ are applied to the combinational Circuit, which

in turn produces the outputs $Z_1, Z_2 \cdots Z_n$ and the values $Y_1, Y_2, \cdots Y_K$.

The values of Y which appear at the output of the combinational circuit at a particular time t are equal to the values of the present state variables y_s at the time (t+1). So Y_s at the input of the memory elements are referred to as next state of the sequential circuit, i.e the state, the circuit will assume next.

Combinational circuits are often faster than sequential circuits since it do not require memory element. However, modern digital computers, must have memories to function properly. Thus, Sequential circuits are of prime importance in modern digital devices.

Sequential circuits are of two types

i) Synchronous or clocked

ii) Asynchronous or unclocked.

Synchronous sequential circuit :- Synchronisation is achieved by a timing device called a master-clock generator, which generates a periodic train of clock pulses.

~~In pre~~ Clocked signal ensure the gates to transmit input signal only after arrival of the clock pulse. The rate at which the master clock generates pulses must be slow enough to permit the slowest circuit to respond. This limits the speed of all circuits.

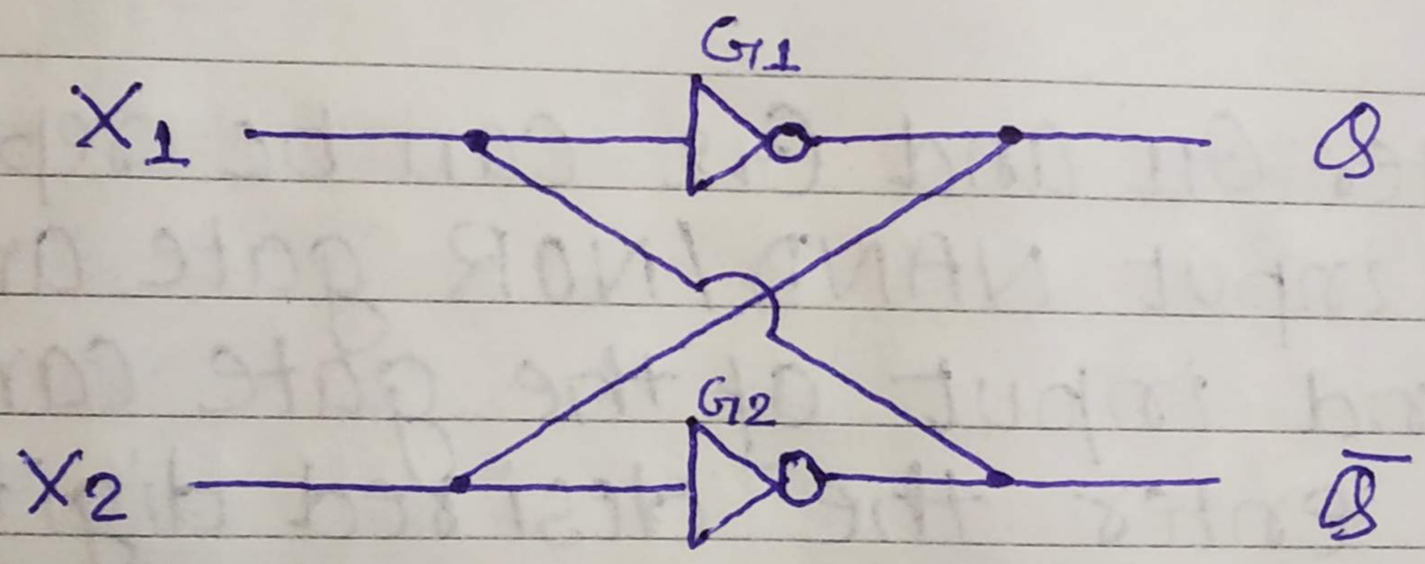
Asynchronous sequential circuit :- In this kind of circuit events can occur after one event is completed and there is no need to wait for a clock pulse.

Therefore, in general, asynchronous circuits are considerably faster than synchronous sequential circuits. However, in an asynchronous circuit, events are allowed to occur without any synchronisation. In such a case, the system becomes unstable which results in difficulties.

Latches :- The simplest kind of a sequential circuit has only two states

It is a memory cell, which is capable of storing one-bit of information, i.e logic 1 or 0. This sequential circuit is also called a latch, since one bit of information can be locked or latched.

The basic latch consists of two inverters



The output Q of inverter G1 is connected to the input X_2 of G2 and the output $\bar{Q}$ of G2 is connected to the input X_1 of G1.

Let us assume that $Q = 1$, then $\bar{Q} = 0$ and when $Q = 0$, then $\bar{Q} = 1$ as per circuit operation.

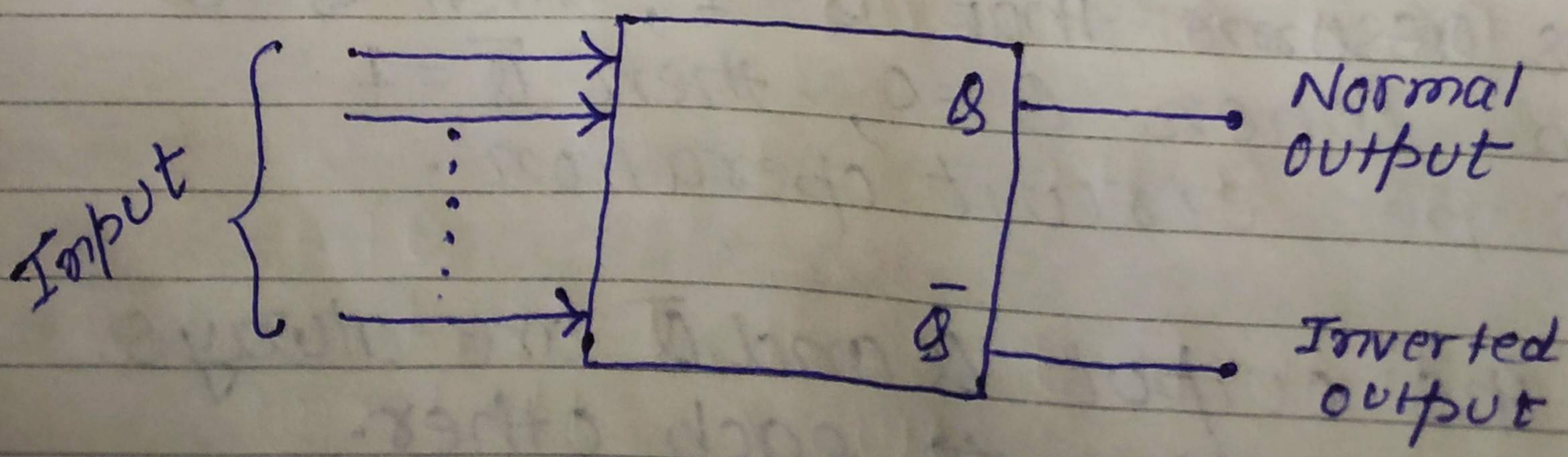
Thus the output Q and $\bar{Q}$ are always complementary to each other.

If the circuit is in state '1' or '0' at Q and $\overline{Q}$, then it continues to remain latched in the same state. This property shows that it can store one bit of digital information.

The basic latch shown in figure above has no provision to get any desired digital information stored. In fact, when the power is switched on, the circuit switches to one of the stable States and it is not possible to predict the state.

The inverter G_{I1} and G_{I2} can be replaced with two input NAND/NOR gate and the second input of the gate can be used to enter the desired digital information.

The general block of a latch with provision to enter digital data can be given as



It has one or more inputs and two outputs Q and $\overline{Q}$. The two output are complement of each other.

If Q = 0 (Reset State), then $\overline{Q} = 1$

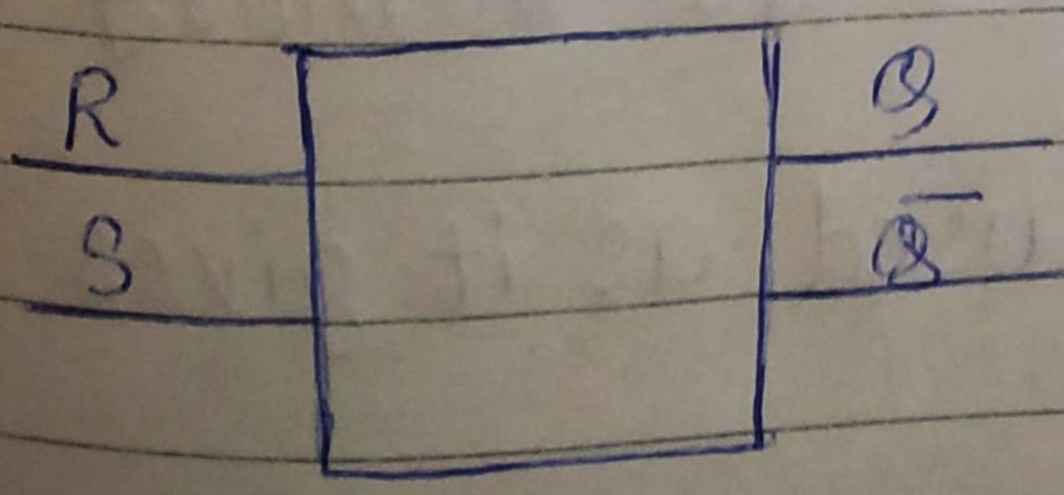
Q = 1 (Set State), then $\overline{Q} = 0$

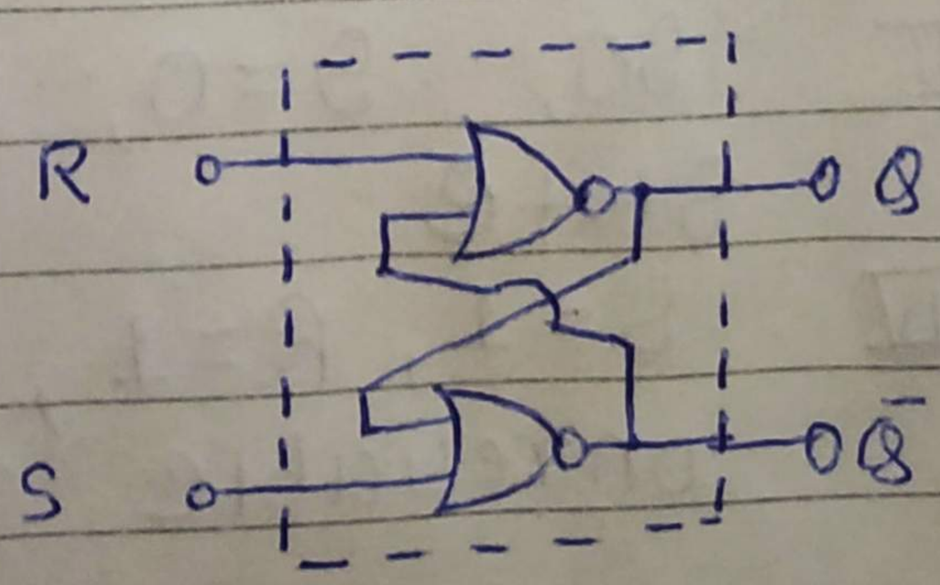
When latch output Q = '0' or '1', it will remain in the same state until one or more of the inputs are excited to effect a change in the output.

Since the latch output will remain Set/reset until the trigger pulse is given to change the state, it can be regarded as a memory device with the capability of storing one binary digit of information.

~~NAND based~~ S-R latch (Set-Reset Latc

The S-R latch has two inputs, and two outpu Q and $\overline{Q}$. The two outputs are complement t each other. The SR latch can be easily construced using two NOR gates connected back to back.





To analyse the circuit, one must remember that the output of a NOR gate is 0 if any input is '1' and the output is 1 only when all inputs are 0.

Case I $S=0$ $R=1$, then
$Q=0$, $\bar{Q}=1$

Let us put $S=0$, $R=0$ after case I, we find
$Q=0$, $\bar{Q}=1$

Case II $S=1$ $R=0$, then
$Q=1$, $\bar{Q}=0$

Let us put $S=0$, $R=0$ after case II, we find
$Q=1$ $\bar{Q}=0$

Here we see that, to make output $Q=0$, we should have $S=0$ $R=1$ and to have $Q=1$, we should have $S=1$, $R=0$

However to keep output Q at its previous state, we need to have $S=0$, $R=0$

Case III So, $S=0$, $R=0$ acts as memory state

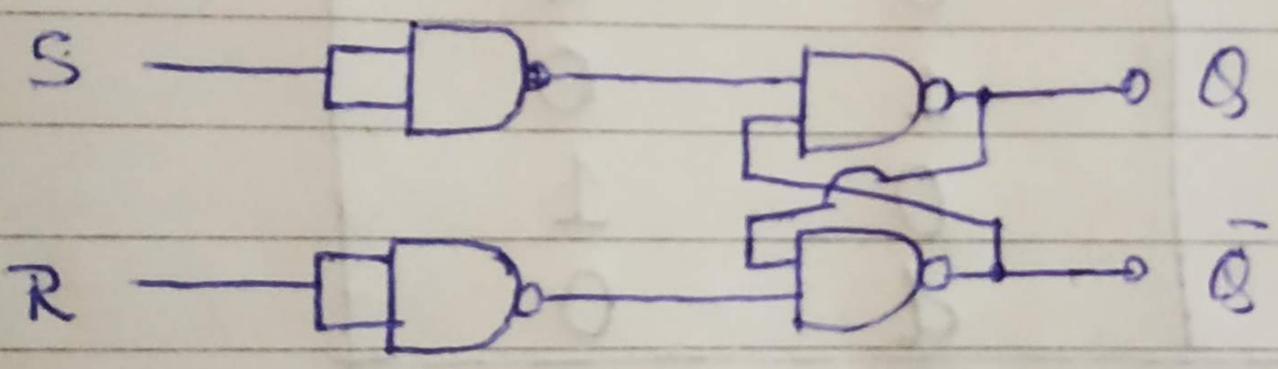
case IV $S=1$, $R=1$, is not used as it gives unreliable result.

So on the basis of the anlayis, the characteristic table of NOR-based SR latch can be written as

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Not used

Q_{n+1} = Next State
Q_n = Previous state

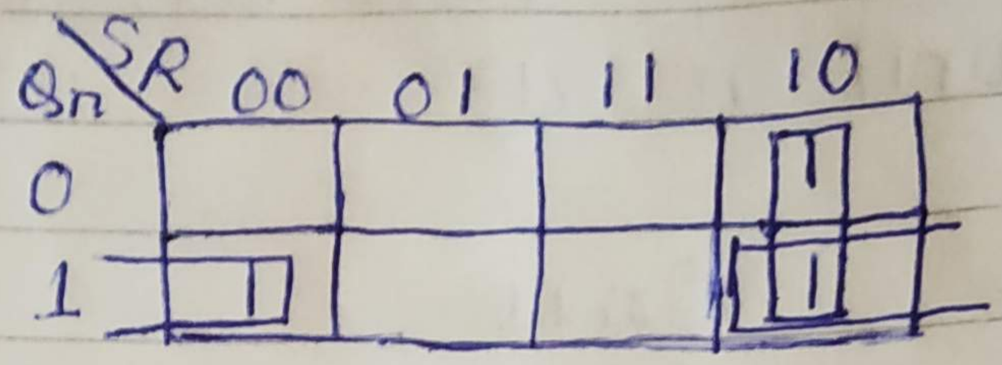
Same functionality can be achieved by SR lacth implemented by NAND gate. The circuit diagram can be given as



Based on above analysis, taking Q_n, S & R as input & Q_{n+1} as output, truth table can be written as

S	R	Q_n	Q_{n+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	?
1	1	1	?

Solving for Q_{n+1} by K-Map we get



$$Q_{n+1} = S\bar{R} + Q_n\bar{R}$$

$$Q_{n+1} = (S + Q_n)\bar{R},$$

It is the characteristic equation of SR latch

To get the ~~excitation table~~, excitation table

Q_n	Q_{n+1}	S	R
0	0	0	d
0	1	1	0
1	0	0	1
1	1	d	0

Excitation table shows us that what input S&R we should provide so that we will get the desired next state (Q_{n+1}) for a given present state (Q_n).