DLD

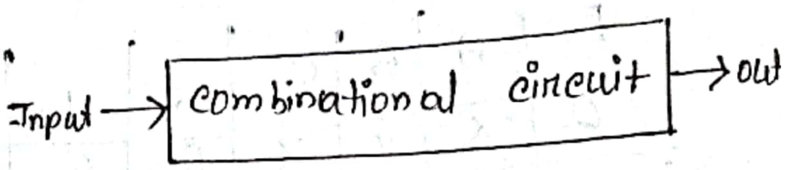
sequential circuit

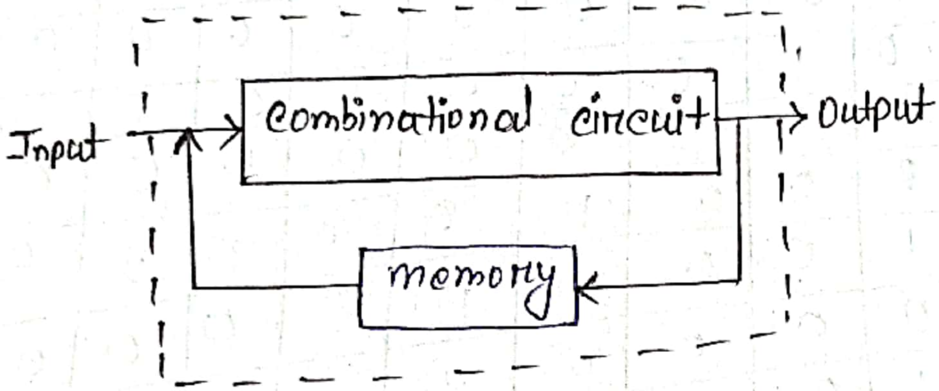
1. Basics of sequential circuit
2. sequential circuit VS combinational circuit
3. Clock and triggering by clock in sequential circuits.
4. Difference between Latch and Flip Flops
5. SR Latch using NOR gates
6. SR Latch using NAND gates
7. D Latch
8. Example on Latch

Sequential circuit

Sequential circuit: It is a combinational circuit and memory.

In ~~combinational~~ sequential circuit present output is only depends on present input and past output.





(sequential circuit)

↓

Example: Flip flop, counter, Register

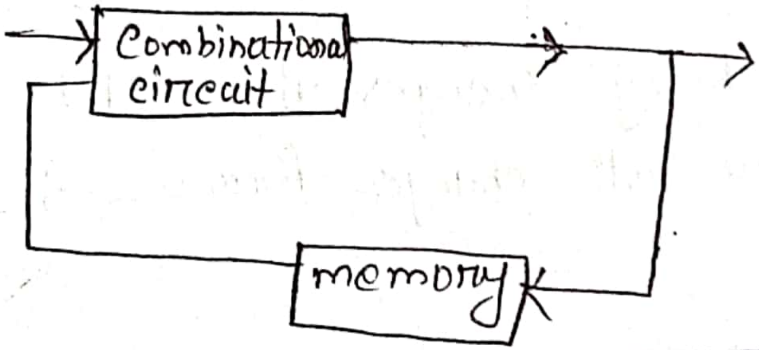
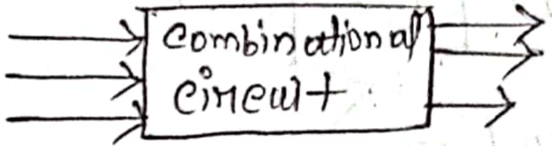
(1 bit memory store করে)

Classification of sequential circuit:

① Synchronous sequential circuit: All memory elements work with same clock pulse.

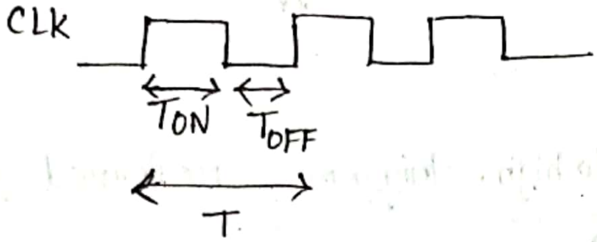
② Asynchronous sequential circuit: Memory elements are working with different clock pulse.

□ Sequential circuit vs combinational circuit

Sequential circuit	combinational circuit
① Output depends on present input and past output	① Output depends on the present input
② Memory element is present	② Memory element is absent
③ Clock signal required	③ No clock signal is applied.
④ Flip Flop, Counters, Registers	④ Half Adder, Full Adder, Multiplexer
	

'clock' and 'triggering by clock' in sequential circuits

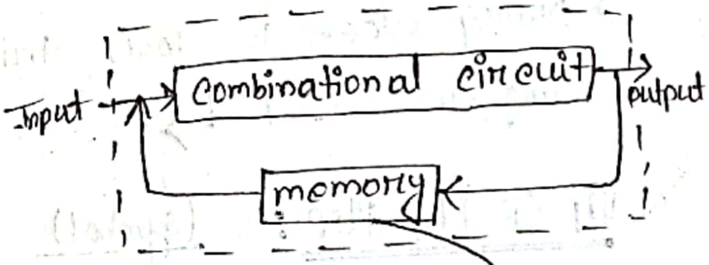
clock is a signal, used in digital circuit.



⇒ Duty cycle, $D = \frac{T_{ON}}{T} = \frac{T/2}{T} = \frac{1}{2} = 0.5$

⇒ Frequency, $F = \frac{1}{T}$

Operational speed and transition of state is defined by clock in sequential circuit



⇒ Output of a circuit is defined as state.

⇒ Output state will change with respect to clock.

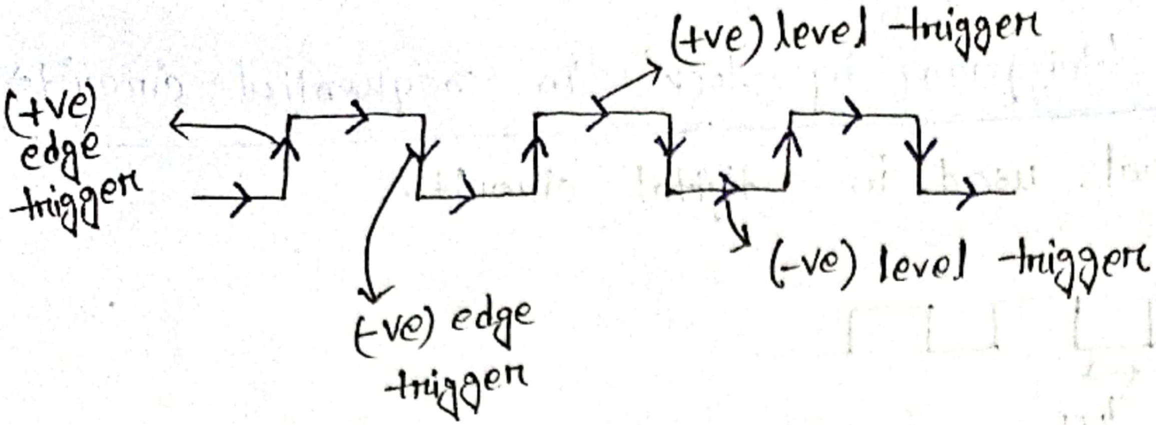
clock is used to control the operational speed and transition of state.

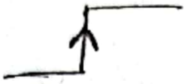
Types of clock triggering:

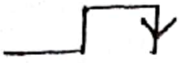
There are two types of clock triggering

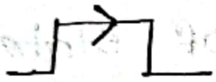
① Edge trigger clock : +ve Edge trigger
-ve Edge trigger

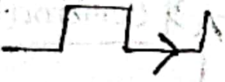
② Level trigger clock : +ve level trigger
-ve level trigger



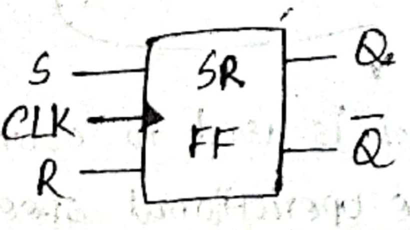
⇒ clock raise from low to high, triggere referred as +ve edge trigger. 

⇒ clock falls from high to low, trigger referred as −ve edge trigger. 

⇒ During clock is high, trigger referred as +ve level trigger. 

⇒ During clock is low, trigger referred as −ve level trigger. 

⊞ SR Flip flop : (symbol)



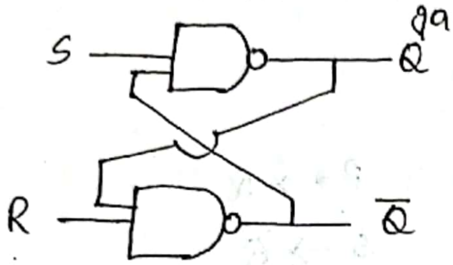
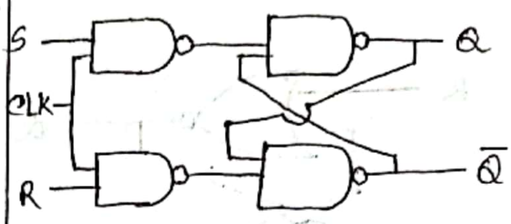
CLK ⟶▷ (+ve) Edge trigger

CLK ⟶○▷ (-ve) Edge trigger

CLK —— (+ve) Level trigger

CLK ——○ (-ve) level trigger

Latch and flip-flop differences:

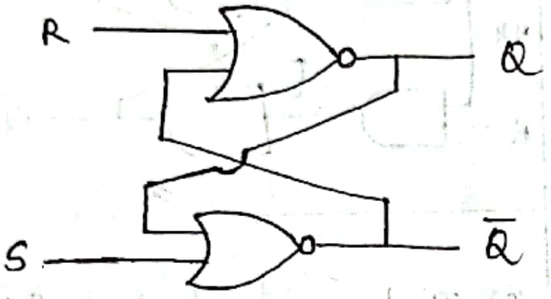
Latch	Flip-flop
① Made by logic gates only. / It is Structured based on logic gates blocks.  	① Made by logic gates and Latch. / It is structured based on blocks of latch and logic gates. 
② It doesn't have clock in its internal circuit.	② It has a clock with its internal circuit.
③ It is level trigger.	③ It is edge trigger.
④ It cannot be used as register.	④ It can be used as register.
⑤ It is Asynchronous circuit.	⑤ It is Synchronous circuit.
⑥ It requires less power.	⑥ It requires more power.
⑦ It is faster.	⑦ It is slower.
⑧ Designing is less complex.	⑧ Desigining is more complex.
⑨ It is Sensitive to input signal only.	⑨ It is sensitive to input and clock signal.

SR Latch using NOR gates:

① Circuit of SR Latch using NOR gate
② Working of SR Latch using NOR gate
③ Truth table of SR Latch using NOR gate

S = set
R = Reset

Circuit:

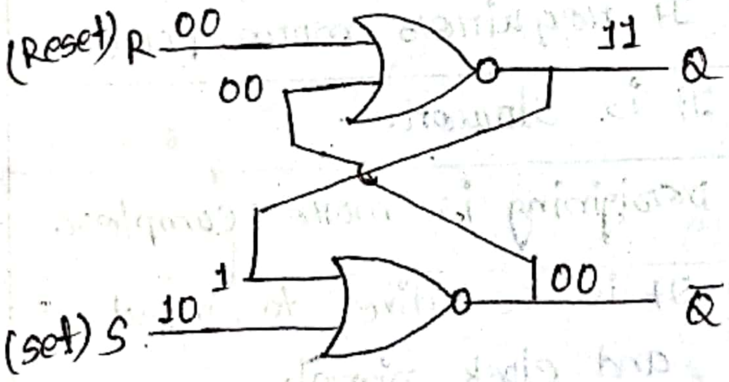


R → Q
S → $\overline{Q}$

Q and $\overline{Q}$ should be complement of each other for Proper operation of SR Latch
If 'Q = Q'' then that's an invalid state.

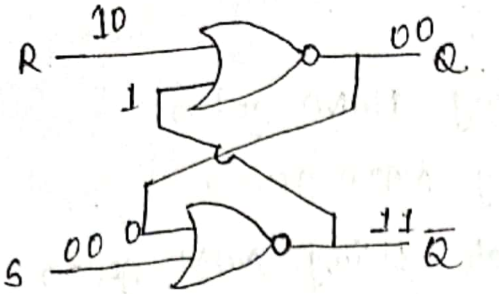
Working:

Nor এর ক্ষেত্রে Reset আগে



S = 1, R = 0, Q = 1, and $\overline{Q}$ = 0

S = 0, R = 0, Q = 1 and $\overline{Q}$ = 0 — It acts like memory.

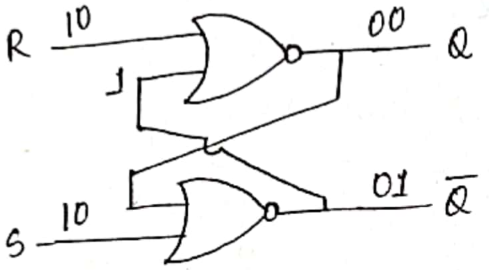


$S = 0, R = 1, Q = 0$ and $\bar{Q} = 1$

$S = 0, R = 0, Q = 0$ and $\bar{Q} = 1$

↓

It acts like memory



$S = 1, R = 1, Q = 0$ and $\bar{Q} = 0$

Invalid case / condition violation

$S = 0, R = 0, Q = 0$ and $\bar{Q} = 1$ } Undefined case
or, $Q = 1$ and $\bar{Q} = 0$

Truth table:

S	R	Q	$\bar{Q}$
0	0	Memory	
0	1	0	1
1	0	1	0
1	1	Invalid	

⊛ Values of Q and $\bar{Q}$ will always be complements of each other. Otherwise it's an invalid case.

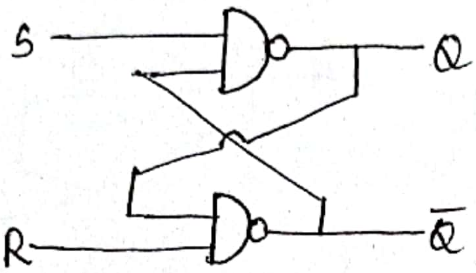
SR Latch using NAND gates:

① Circuit of SR Latch using NAND gates

② Working of SR Latch using NAND gates

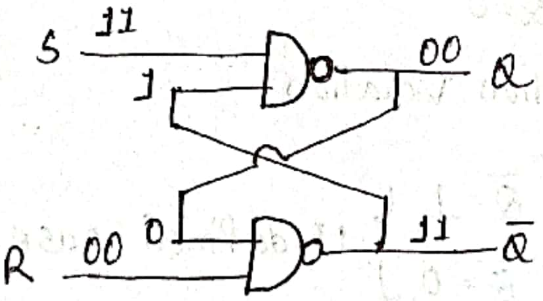
③ Truth table of SR Latch using NAND gates

Circuit:



Nand এর ক্ষেত্রে set আগে বসবে।

Working:

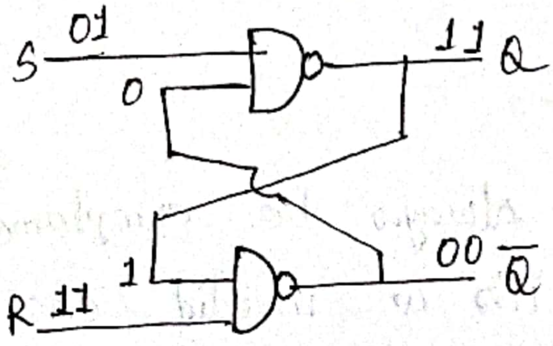


$S = 1$, $R = 0$, $Q = 0$ and $\bar{Q} = 1$

$S = 1$, $R = 1$, $Q = 0$, and $\bar{Q} = 1$

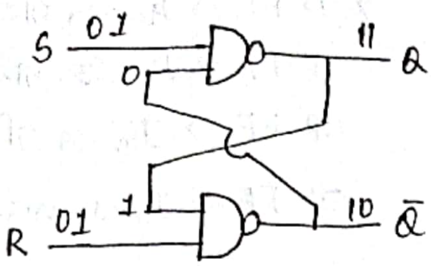
↓

Acts as memory



$S = 0, R = 1, Q = 1$ and $\bar{Q} = 0$

$S = 1, R = 1, Q = 1$ and $\bar{Q} = 0$



$S = 0, R = 0, Q = 1$ and $\bar{Q} = 1$

↓

Invalid case / condition violation

$S = 1, R = 1, Q = 1$ and $\bar{Q} = 0$ or $Q = 0$ and $\bar{Q} = 1$ } undefined

Truth Table

S	R	Q	$\bar{Q}$
0	0	Invalid	
0	1	1	0
1	0	0	1
1	1	Memory	

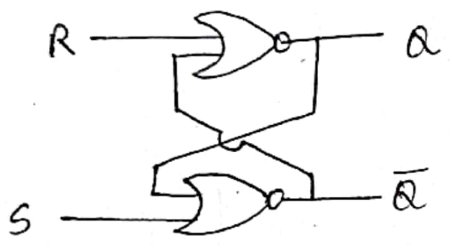
- In NAND gate, output will always be 1 if one of the inputs is zero.

D latch : D latch does not have clock input and D-flip-flop has clock input.

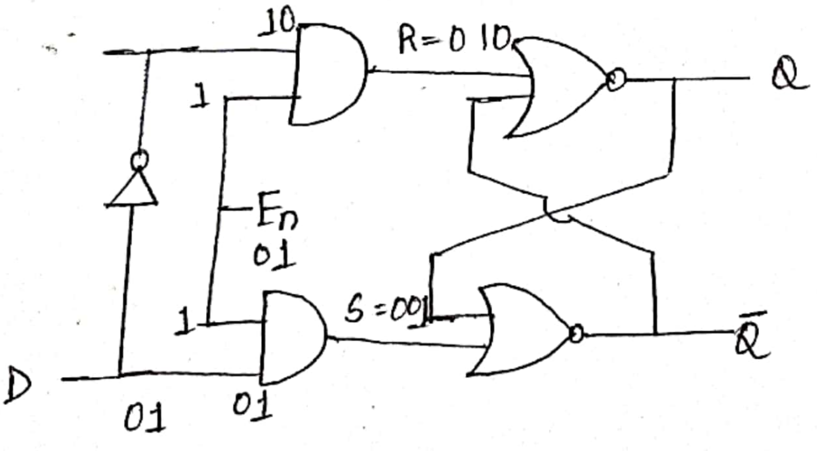
output = Input when Enable = 1

output = memory when En = 0

SR Latch using nor gate truth table



S	R	Q	$\bar{Q}$
0	0	memory	
0	1	0	1
1	0	1	0
1	1	Invalid	



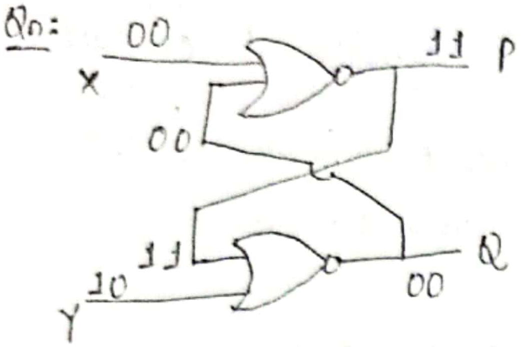
□ En = 0, D = X, Q = mem

□ En = 1, D = 0 , Q = 0
D = 1 , Q = 1

En	D	Q
0	X	mem
1	0	0
1	1	1

Example on Latch

Qn:

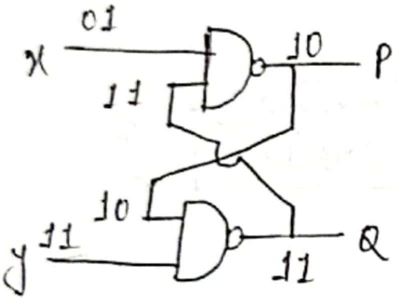


If input xy changes from 01 to 00 then output PQ will change from ? to ?.

when xy is 01 PQ will be 10

If input xy changes from 01 to 00 then output PQ will changes from 10 to 10.

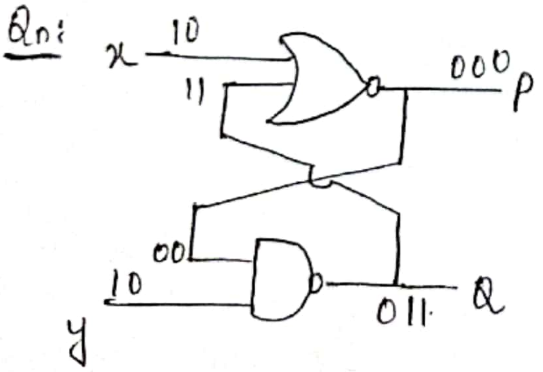
Qn:



If input xy changes from 01 to 11 then PQ will changes from __ to __.

If input xy changes from 01 to 11 then PQ will changes from 11 to 01

Qn:



If input xy changes from 11 to 00 then PQ will changes from __ to __

[Initially PQ is 00]

If input xy changes from 11 to 00 then PQ will changes from 01 to 01.