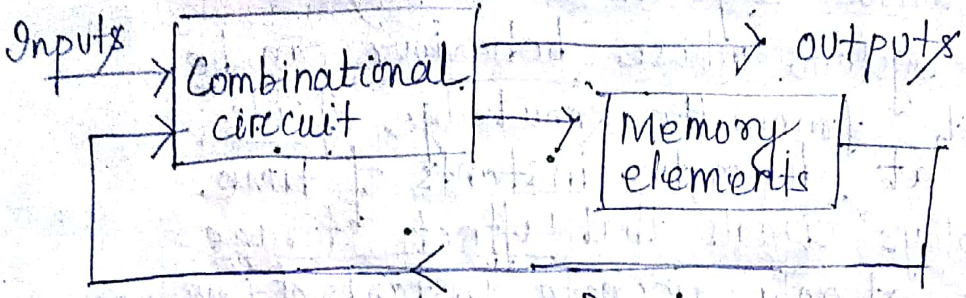
[Synchronous Sequential Logic]:-

* (Sequential Circuits):-



(block diagram)

→ It consists of a combinational circuit to which storage elements are connected to form a feedback path.

→ Storage elements are devices capable of storing binary information which stored in these elements at any given time defines "state" of sequential ckt at that time.

→ Sequential ckt recieves binary information from external inputs that together with present state of storage elements, determine binary value of outputs.

→ These external inputs also determine condition for changing the state in storage elements.

→ Block diagram states that $O/_{Ps}$ in a sequential ckt are a function not only of the inputs, but also of present state of storage elements

→ Next state of storage elements is also a function of external inputs & present state.

→ So, sequential ckt is specified by a time sequence of inputs, outputs & internal states.

→ Two types of sequential ckts :-

i) (Sequential) Synchronous ckts.

ii) Asynchronous sequential ckts

✳ (Synchronous sequential circuits) :-

→ It's a system whose behaviour can be defined from the knowledge of its signals at discrete instants of time.

→ It employs signals that affect storage elements at only discrete instants of time.

→ Syncronization achieved by a timing device called "clock generator" which provides a clock signal in form of periodic train of clock pulses.

→ It uses clock pulses to control storage elements called "clocked sequential ckts"

→ Activity within ckt & resulting updating of stored values is synchronised to occurance of clock pulses.

✳ (Asynchronous sequential circuits) :-

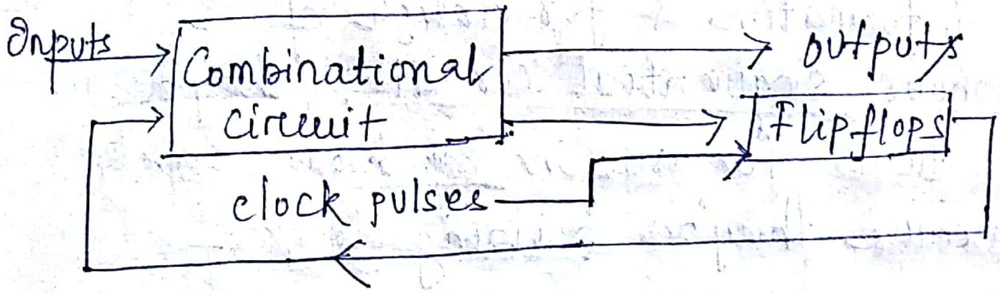
→ Behaviour of it depends upon input signals at any instant of time & the order in which inputs change.

→ storage elements used are 'time-delay devices.

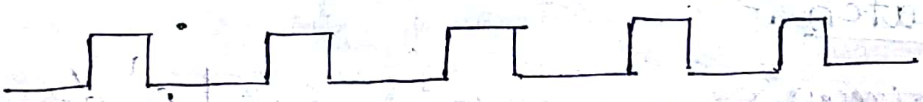
→ storage elements consists of logic gates whose propagation delay provide required storage so, it's regarded as combinational ckt with feedback.

* Storage elements (memory) used in clocked sequential circuits are called flipflops. It is a binary storage device capable of storing one bit of information.

→ In a stable state, O/p of flipflop is either '0' or '1'.



(Block diagram)



(Timing diagram of clock pulses) → Synchronizing pulses at regular interval

[Synchronous clocked Sequential circuit]

→ O/ps are formed by a combinational logic function of inputs to circuit or values stored in flipflops or both.

→ New value is stored (flipflop is updated) when a pulse of clock signal occurs.

→ State of flipflops can change (updated) only during a clock pulse transition.

Storage Elements : Latches :-

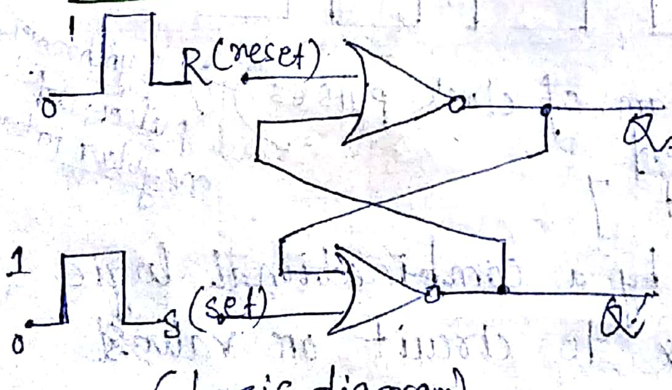
→ " " that operate with signal levels (rathere than signal transitions) are latches, but those controlled by a clock transition are flipflops.

→ Latches are said to be level sensitive devices, but flipflops are edge sensitive "

→ Latches are the basic circuits from which all flipflops are constructed.

→ Although latches are useful for storing binary information & for design of asynchronous sequential circuits, they're not practical for use in 'synchronous sequential ckts. Because they're building blocks of flipflops.

S-R Latch :-



(Logic diagram)

S	R		Q	Q'	
1	0	→	1	0	(set)
0	0	→	1	0	(after S=1,
0	1	→	0	1	(reset)
0	0	→	0	1	(after S=0, R
1	1	→	0	0	(forbidden)

(Function Table)

[S-R Latch with NOR gates]

→ SR latch is a ckt with two cross-coupled NOR gates or NAND gates, & two inputs labeled 'S' for set & 'R' for reset.

→ In above fig, latch has two useful states. When o/p (Q = 1) & (Q' = 0), latch is said to be in 'set state', but when (Q = 0) & (Q' = 1), it's in reset state.

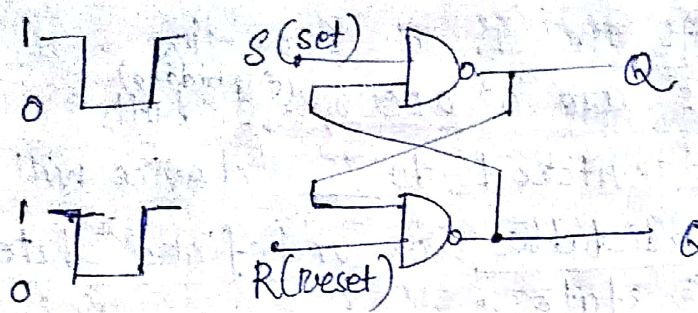
→ When both inputs are '1' at same time then both o/ps are '0' occurs (forbidden) & both inputs are if switched to '0', device will enter an unpredictable or 'undefined' state or 'metastable' state.

→ 1st condition (S=1, R=0) is action must be taken by input 'S' to bring ckt to 'set' state. Removing active i/p from 'S' leaves ckt in same state. (S=0, R=0)
(Q=1, Q'=0)

→ After both inputs return to '0', it's then possible to shift to 'reset' state by applying '1' to 'R' input. The '1' can then be removed from 'R' where ckt remains in reset state. (S=0, R=0)
(Q=0, Q'=1)

→ So, when both inputs 'S' & 'R' are equal to '0', the latch can be in either 'set' or 'reset' state, depending on which input was most recently a '1'.

→ If a '1' is applied to both 'S' & 'R' i/ps of latch, both outputs go to '0'. which produces an 'undefined next state' (forbidden) because the state results from input transitions in which they return to '0'. It violates ~~to~~ of o/ps be complement of each other.



(Logic diagram)

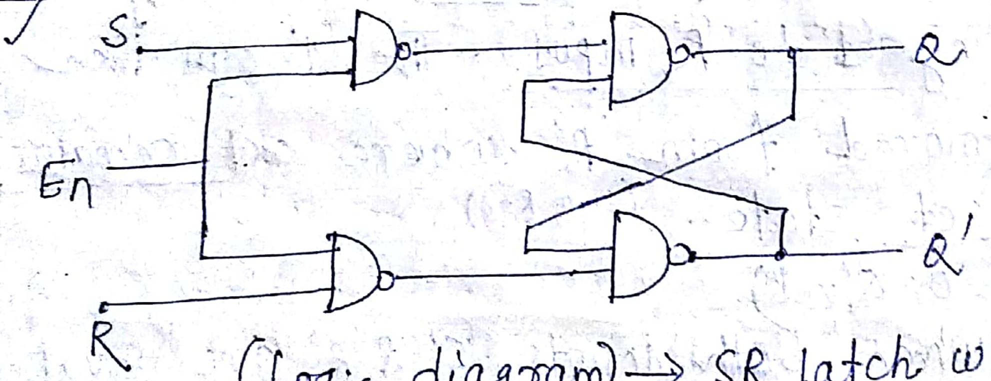
S	R	Q	Q'	
1	0 →	0	1	(Reset
1	1	0	1	(after S=
0	1	1	0	(Set) R=
1	1	1	0	(after S=
0	0	1	1	(forbidd

(Function Table)

[S-R Latch with NAND gates]

→ Input signals for NAND require complement of those values used for NOR latch. Because NAND latch requires a '0' signal to change its state, it's refered as an S'R' latch.

→



(Logic diagram) → SR latch with control input

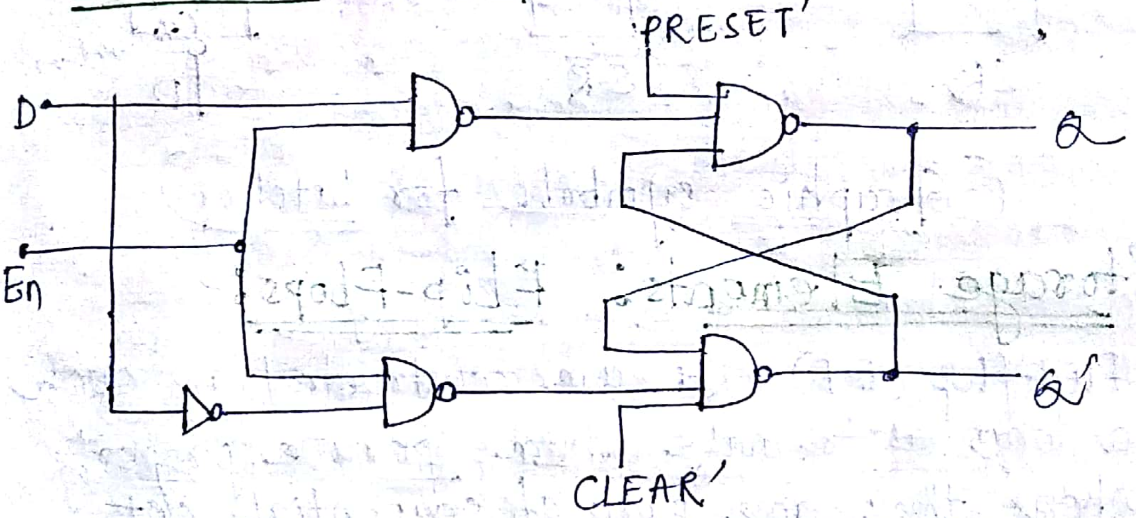
En	S	R	Next state of Q
0	X	X	→ No change
1	0	0	→ No change
1	0	1	→ Q = 0; reset state
1	1	0	→ Q = 1; set state
1	1	1	→ Indeterminate

(Function table)

→ Operation of basic SR latch can be modified by providing an additional input signal that determine (control) when state of latch can be changed.

→ Fig. shows basic SR latch & two additional NAND gates. Control input (En) acts as an enable signal for other two inputs.

D Latch (Transparent Latch)



(Logic diagram)

En	D	Next state of Q
0	X	→ No change
1	0	→ Q = 0 (reset state)
1	1	→ Q = 1 (set state)

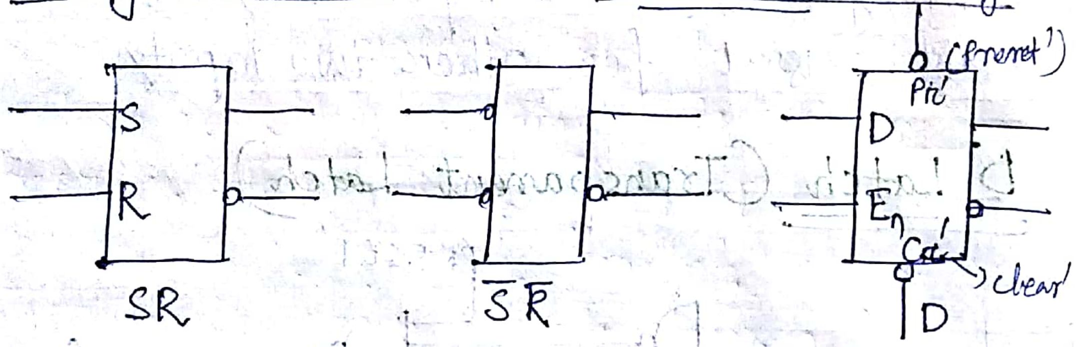
[Function table]

→ One way to eliminate undesirable condition of "indeterminate" state in SR latch is to ensure that inputs S & R are never equal to '1' at same time. This is done in D-latch.

→ This latch has only two inputs:- D(data) and En(enable).

→ 'D' input goes directly to 'S' input & it's complement is applied to 'R' input.

→ 'D' latch recieves that designation from its ability to hold data in its internal storage.



(Graphic symbols for Latches)

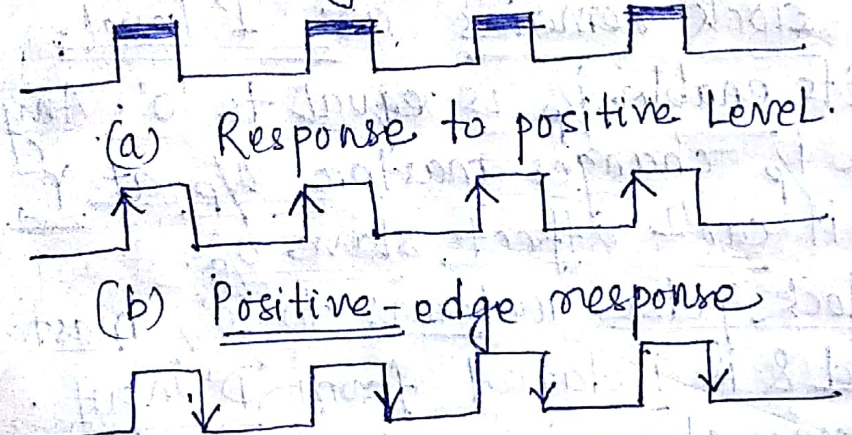
Storage Elements: Flip-Flops:-

→ Flip-flop (F.F) ckts are constructed in such a way as to make them operate properly when they are part of sequential ckt. that employ a common clock.

→ Key to proper operation of a flip-flop is to trigger it only during a signal transition. This can be accomplished by eliminating the feedback path that is inherent in operation of sequential ckt using latches.

→ As shown in fig., positive transition is defined as 'positive edge' & negative transiti as 'negative " '.

→ There're two ways that a latch can be modified to form a flipflop. One way is to employ two Latches in a special configuration that isolates o/p of flipflop & prevents it from being affected while i/p to flipflop is changing. Another way to produce a " that triggers only during a signal transition of synchronized signal (clock) & is disabled during rest of clock pulse.



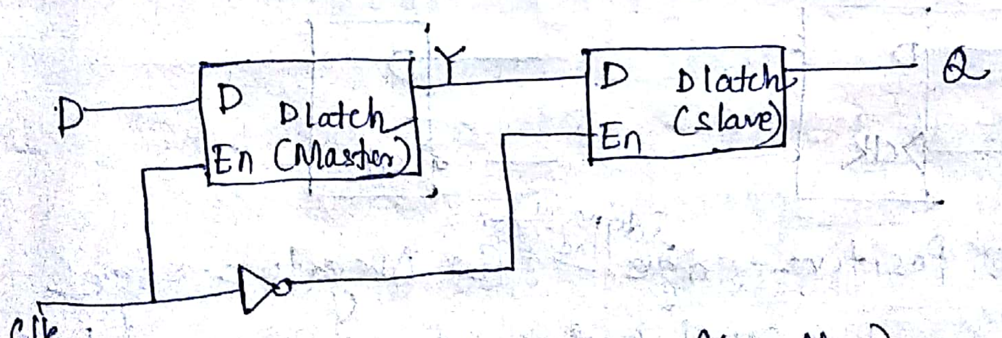
(a) Response to positive Level.

(b) Positive-edge response

(c) Negative-edge response

[clock response in Latch fig (a) & flipflop fig (b) & (c)]

Edge-Triggered D-Flip-Flop:-



(Master-slave D flip-flop)

→ First latch is called 'master' & second latch is 'slave'

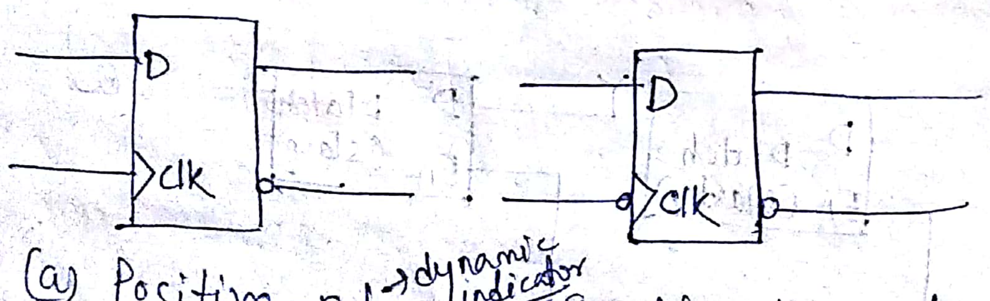
→ ckt samples 'D' input & changes its o/p 'Q' only at negative edge of synchronizing or controlling clock.

→ When clock is '0', O/p of inverter is '1'. The slave latch is enabled, & its O/p 'Q' is equal to master output 'Y'. The master latch is disabled, because (clk = 0).

→ When input pulse changes to logic-1 lev data from external 'D' input are transferred to master. The slave is disable as long as clock remains at '1' level. because its enable i/p is equal to '0'. Any change in i/p changes master O/p at Y because it can't affect slave O/p.

→ When clock pulse returns to '0', master is disabled & is isolated from 'D' input. At same time, slave is enabled & value of 'Y' is transferred to O/p of flipflops at 'Q'.

→ So, a change in O/p of F.F. can be triggered only by & during transition of clock from '1' to '0'.



(a) Positive-edge →dynamic indicator (b) Negative-edge

[Graphic symbol for edge-triggered 'D' F.F]

→ "Timing" of response of F.F to i/p data & to clock must be considered for edge-triggered F.F.

→ There's minimum time called "set-up time" during which 'D' i/p must be maintained at constant value prior to occurence of clock transition.

→ Minimum time called 'hold time" during which 'D' i/p must not change after application of positive transition (0 to 1) of clock.

→ "propagation delay time" of F.F is interval between trigger edge & stabilization of o/p to a new state.

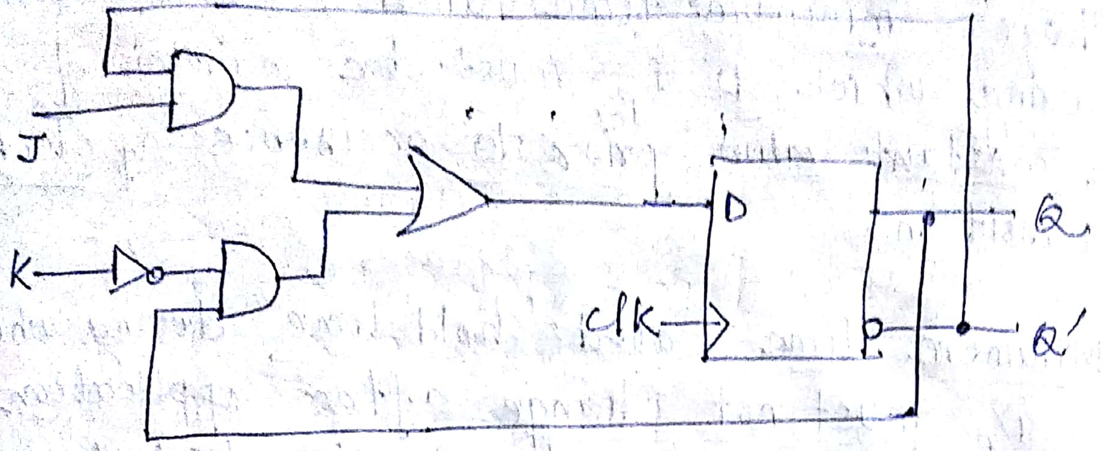
Other FlipFlops :-

→ Each flipflop is constructed from an interconnection of gates.

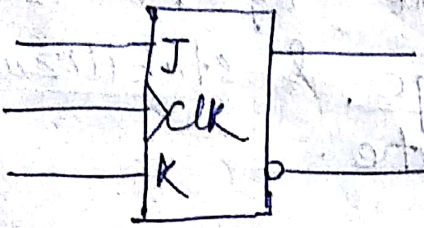
→ Most economical & efficient flipflop constructed is edge-triggered D-flipflop, because it requires smallest number of gates.

→ There're three operations that can be performed with flipflop :- a) Set it to '1', b) Reset it to '0', c) or complement its o/p'

→ Synchronized by a clock signal, "J-K flipflop has two inputs & performs all three operations.



(a) (ckt diagram)



(b) (Graphic symbol)

("JK" Flip-flop)

→ Ckt diagram of a JK flipflop constructed with a D flip flop & gates is shown.

→ 'J' input sets flipflop to '1', the 'K' i/p resets it to '0' & when both i/p's are enabled, o/p is complemented.

→ This can be verified applying 'D' input.

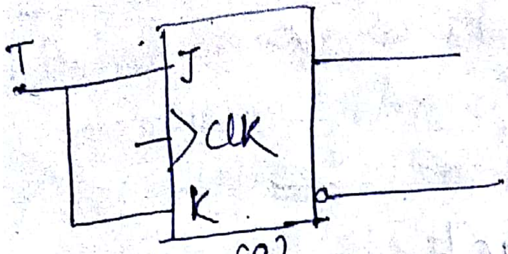
$\boxed{D = JQ' + K'Q}$ —— ①

→ When (J=1), & (K=0), D = Q' + Q = 1, so next clock edge sets o/p to '1'.

→ When (J=0), (K=1), then (D = 0), so next clock edge resets o/p to '0'.

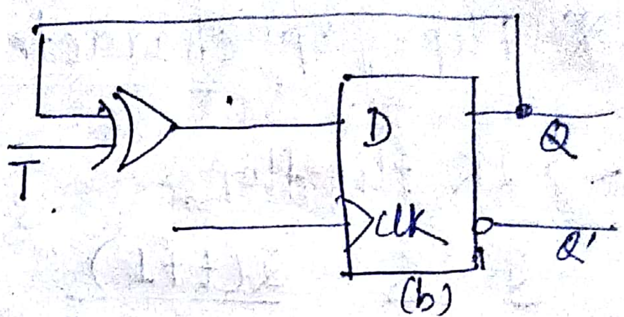
→ When both (J=K=1), then (D = Q'), the next clock edge complements the o/p.

→ When both (J=K=0), then (D = Q), the clock edge leaves o/p unchanged.



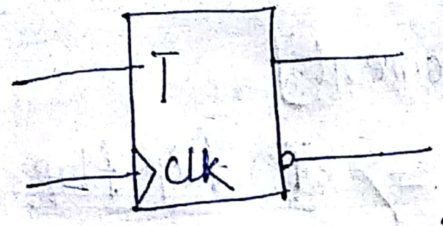
(a)

(From JK Flip flop)



(b)

(From D flip-flop)



(c). (Graphic symbol)

[T flip-flop]

→ T (Toggle) flip-flop is a complementing f-f & can be obtained from JK f-f when i/ps 'J' & 'K' are tied together as shown in fig(a)

→ When T = 0, (J = K = 0), a clock edge doesn't change o/p. When (T = 1), (J = K = 1), a clock edge complements the o/p, useful for designing binary counters.

→ T f-f can be constructed with D f-f & an EX-OR gate as shown in fig(b) Expression for D i/p is

$$\boxed{D = T \oplus Q = TQ' + T'Q} \quad \text{—— ②}$$

→ When T = 0, (D = Q) & there's no change in o/p. When (T = 1) ⇒ (D = Q') & the o/p complements.

* Flip-flop characteristic Tables :-

⇒ JK flip-flop

J	K	Q(t+1)
0	0	→ Q(t) → [No change]
0	1	→ 0 → [Reset)
1	0	→ 1 → [set]
1	1	→ Q'(t) → [Complement]

⇒ D flip-flop

D	Q(t+1)
0 →	0 → (Reset)
1 →	1 → (Set)

⇒ T flip-flop

T	Q(t+1)
0 →	Q(t) [No chang
1 →	Q'(t) [complem

Characteristic Tables :-

→ It defines logical properties of a f-f by describing its operation in tabular form.

→ They define the next state [Q(t+1)] as a function of inputs & present state [Q(t)].

Characteristic Equations :-

-> Logical properties of a f-f described in characteristic table, can be expressed algebraically with a charactristic eq^n.

$$\boxed{Q(t+1) = D} \text{ --- } ①$$

$$\boxed{Q(t+1) = JQ' + K'Q} \text{ --- } ②$$

$$\boxed{Q(t+1) = T \oplus Q = TQ' + T'Q} \text{ --- } ③$$

State Equations:- (Transition Equation)

→ Behaviour of a clocked sequential ckt can be described algebrically by means of state eqⁿ.

→ It specifies the next state as a function of present state and inputs.

State Table:- (Transition Table):-

→ Time sequence of inputs, outputs & flipflops states can be enumerated in a state table.

Let

$$\left.\begin{aligned} &[A(t+1) = Ax + Bx] \\ &[B(t+1) = A'x] \\ &[y = Ax' + Bx'] \end{aligned}\right\} \text{(State eq}^{n}\text{s)}$$

State Table:-

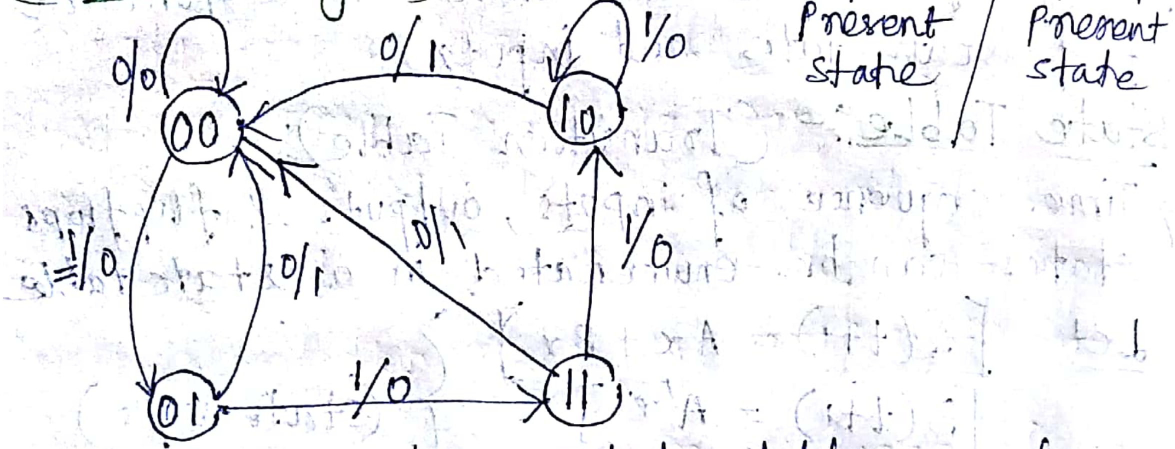
Present state		Input		Next state		O/P
A	B	x		A [A(t+1)]	B [B(t+1)]	y
0	0	0	→	0	0	0
0	0	1	→	0	1	0
0	1	0	→	0	0	1
0	1	1	→	1	1	0
1	0	0	→	0	0	1
1	0	1	→	1	0	0
1	1	0	→	0	0	1
1	1	1	→	1	0	0

* (Second form of state table):-

Present state			Next state				output	
			(x=0)		(x=1)		(x=0)	(x=1)
A	B		A	B	A	B	y	y
0	0	→	0	0	0	1	0	0
0	1	→	0	0	1	1	1	0
1	0	→	0	0	1	0	1	0
1	1	→	0	0	1	0	1	0

→ A sequential ckt with 'm' flipflops & 'n' i/ps needs 2^{m+n} rows in state table.

(State Diagram):-



→ information in a state table can be represented graphically in form of state diagram (pictorial view of state transition)

→ A state is represented by a circle & transition bet^n states are indicated by (clock triggered) directed lines connecting circles.

→ Binary no. inside each circle identifies state of flipflops.

→ ex:- directed line from state '00' to 01 is labeled 1/0, meaning when sequential ckt is in present state '00' & i/p is '1', the o/p is '0'. After next clock cycle, ckt goes to next state, 01.

→ From fig., from state 00, o/p is '0' as long as i/p stays at 1.

→ First '0' input after a string of 1's gives an o/p of '1' & transfers ckt back to initial state 00.

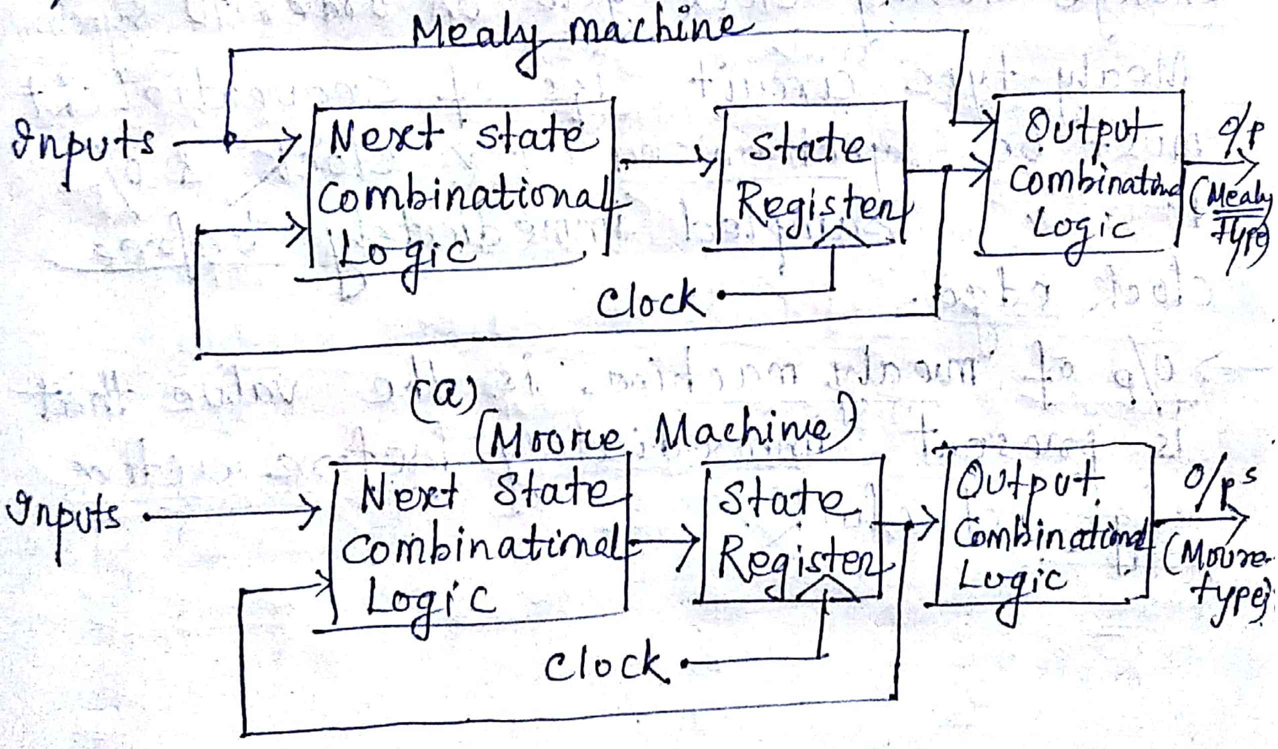
Mealy & Moore Models of Finite State Machine :-

→ General model of a sequential ckt has inputs, outputs & internal states.

→ Two models of sequential ckts :

i) Mealy model

ii) Moore model.



[Block diagrams of Mealy & Moore state Machine]

→ They differ only in the way the o/p is generated.

→ In 'Mealy model', O/P(y) is a function of both present state (A&B) & Input (x). In 'Moore model', O/p is a function of only present state

→ i) Mealy "Finite State Machine" (FSM) / Mealy Mach

ii) Moore FSM / Moore Machine

→ In 'Moore model' O/ps of sequential ckt are synchronized with clock, because they depend only on flip flop O/ps that are synchronized with clock.

→ In 'Mealy model', O/ps may change if inputs change during clock cycle. In order to synchroni Mealy-type circuit, I/ps of sequential ckt must be synchronized with clock & O/ps " " sampled immediately before clock edge.

→ O/p of mealy machine is the value that is present immediately before active edge of clock.