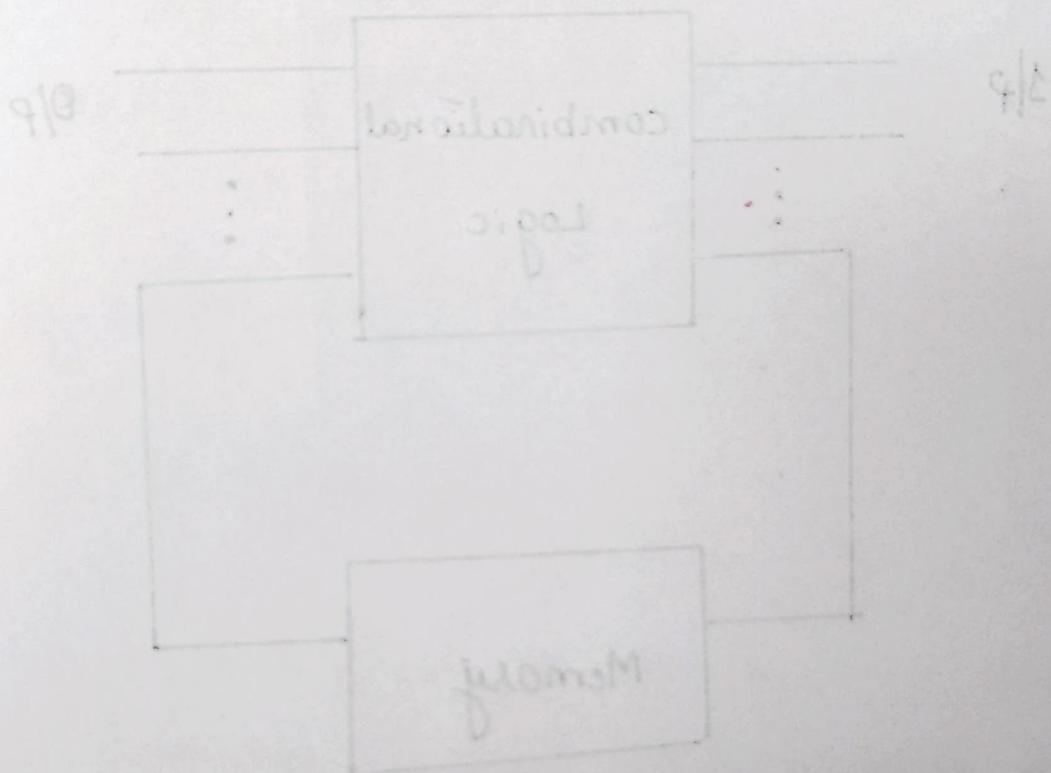


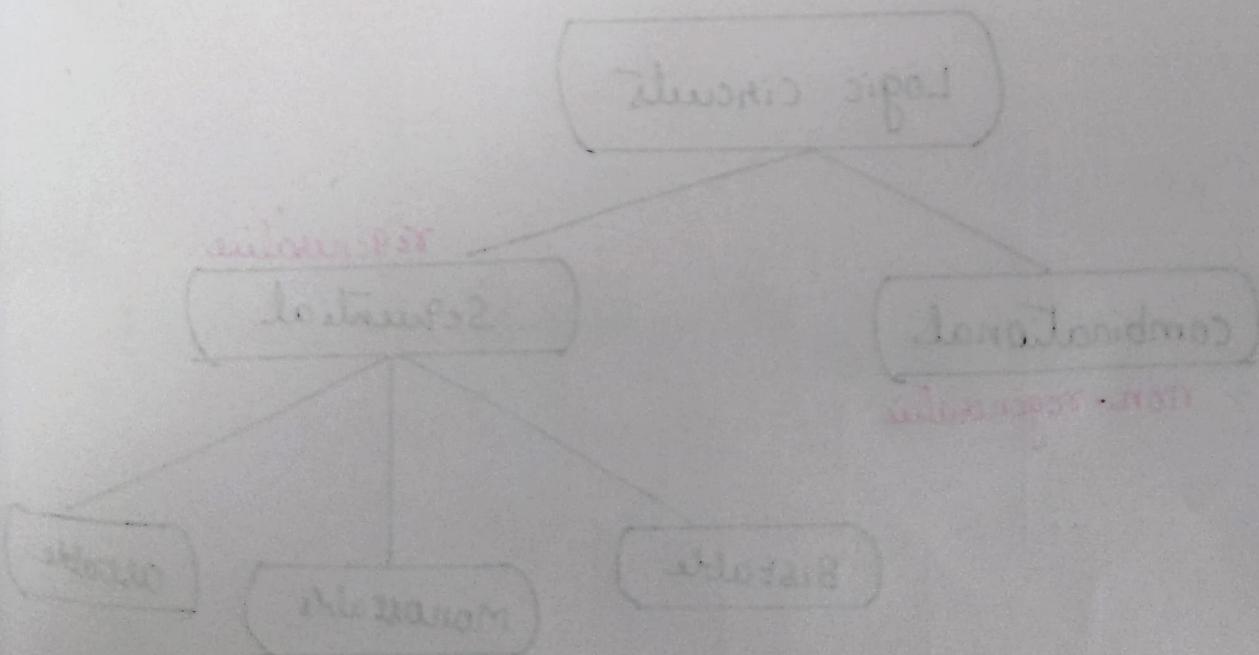
Module 4

- Block diagram :-

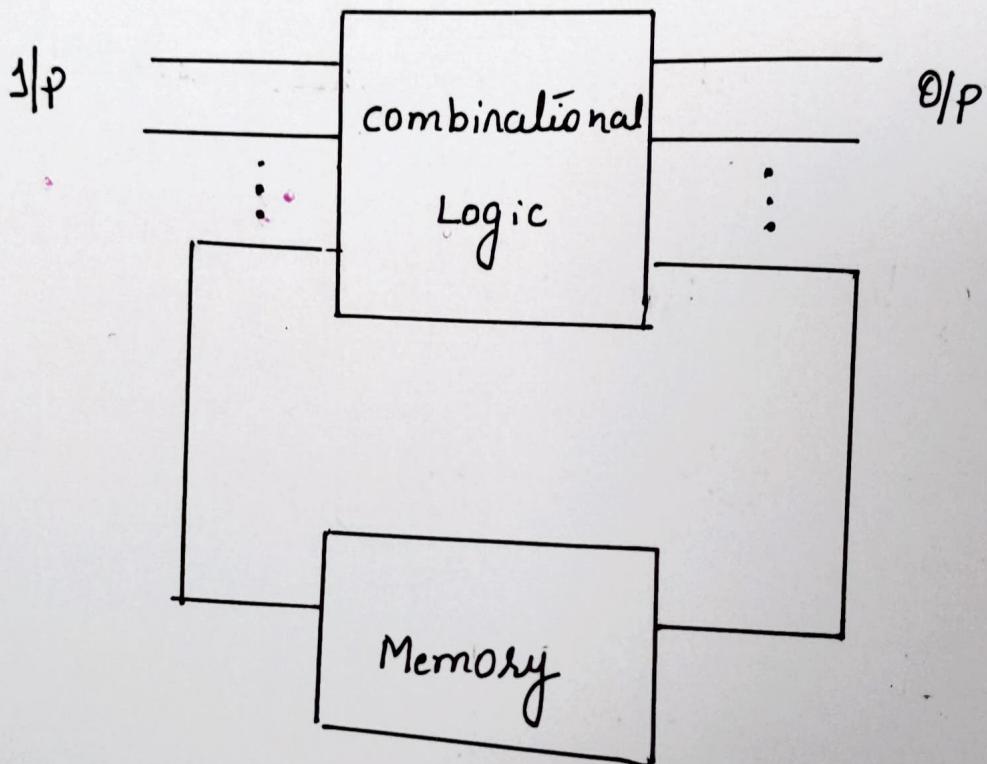
contents :-



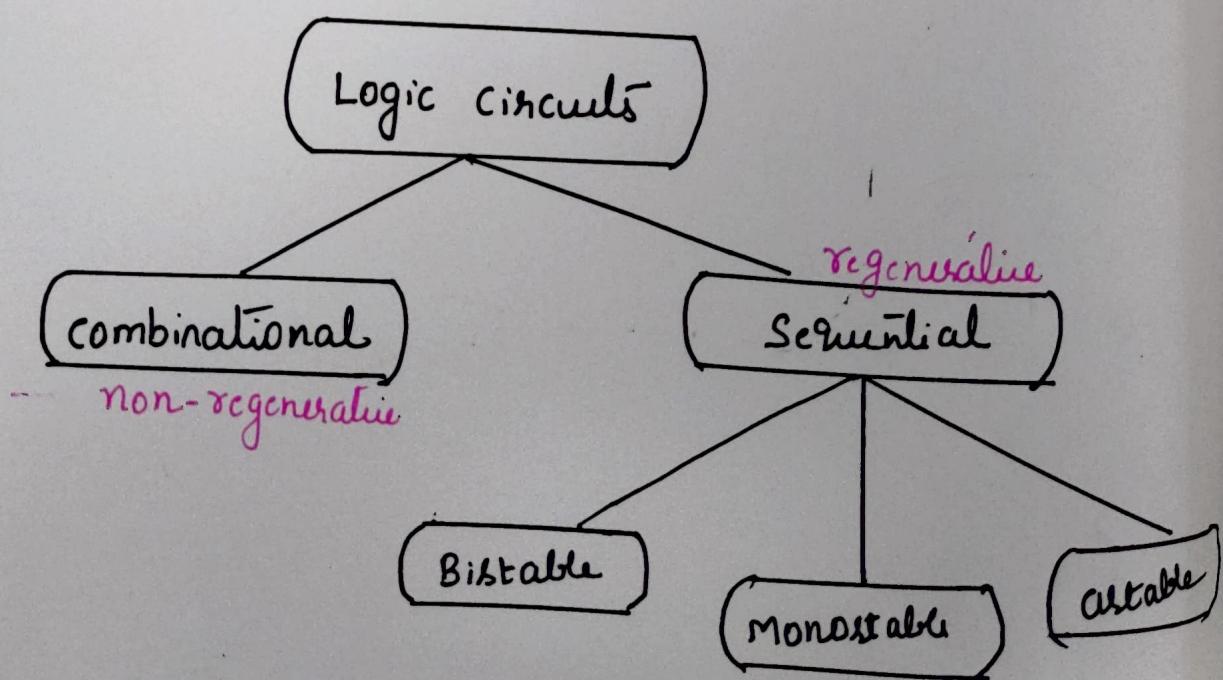
Classification of logic circuit



Block diagram :-



Classification of logic circuits

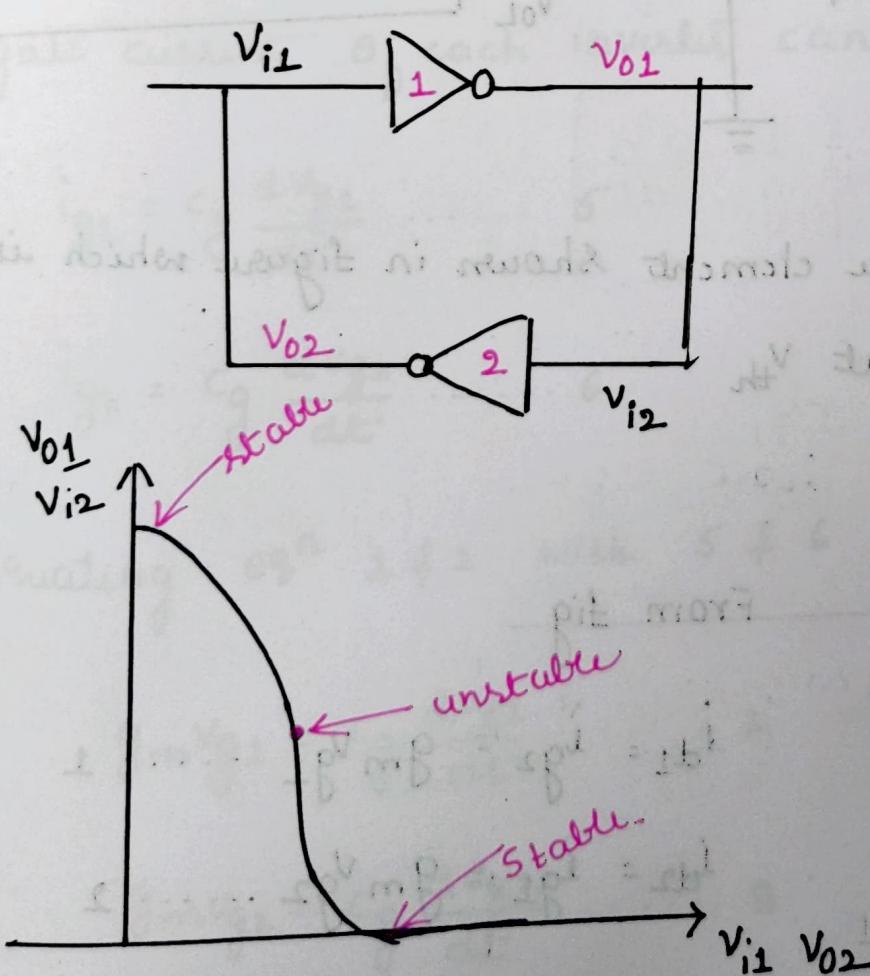


→ Bistable Circuit is a one which has two stable state. Ex: flip flop.

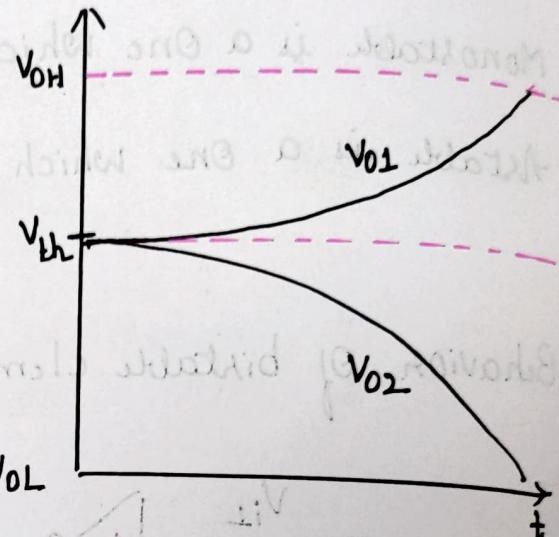
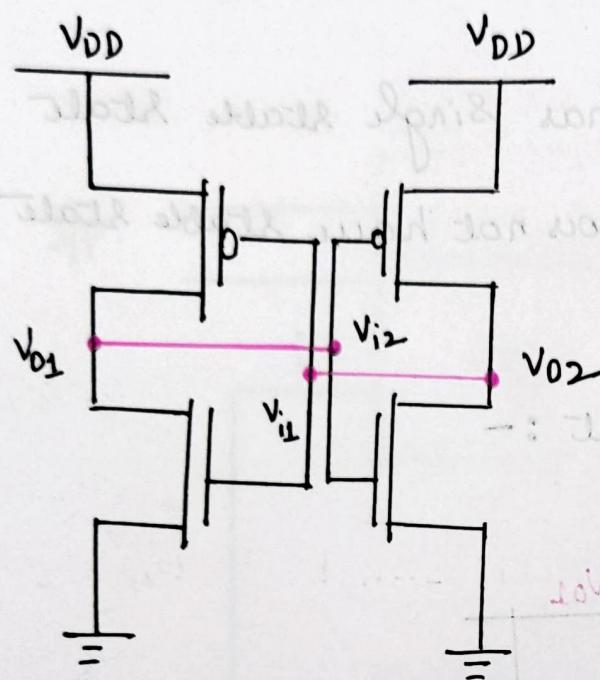
→ Monostable is a one which has single stable state

\rightarrow metastable is a one which does not have stable state

Behavior of bistable element :-



CMOS bistable element :-



→ Consider a bistable element shown in figure which is initially operating at v_{th}

→ assume $C_g > C_d$

From fig

$$i_{d1} = i_{g2} = g_m v_{g1} \dots \dots 1$$

$$i_{d2} = i_{g1} = g_m v_{g2} \dots \dots 2$$

Here g_m represents small signal transconductance

$$i_d = g_m v_{gs}$$

$$q = Cv$$

$$i = C \frac{dv}{dt}$$

→ gate voltage of both inverters can be expressed in terms of gate charge q_1 and q_2

$$V_{g1} = \frac{q_1}{C_g} \quad \dots \dots \quad 3$$

$$V_{g2} = \frac{q_2}{C_g} \quad \dots \dots \quad 4$$

→ gate current of each inverter can be written as

$$i_{g1} = C_g \frac{dV_{g1}}{dt} \quad \dots \dots \quad 5$$

$$i_{g2} = C_g \frac{dV_{g2}}{dt} \quad \dots \dots \quad 6$$

→ Equating eqⁿ 1 f 2 with 5 f 6

$$g_m V_{g1} = C_g \frac{dV_{g2}}{dt} \quad \dots \dots \quad 7$$

$$g_m V_{g2} = C_g \frac{dV_{g1}}{dt} \quad \dots \dots \quad 8$$

$$g_m V_{g1} = C_g \frac{dV_{g2}}{dt}$$

$$g_m \frac{q_1}{C_g} = i_{g2} = \frac{dq_2}{dt} \quad \dots \dots \quad 9$$

$$g_m \frac{q_2}{C_g} = i_{g1} = \frac{dq_1}{dt} \quad \dots \dots \quad 10$$

From equation 10

$$q_2 = \frac{c_g}{g_m} \frac{dq_1}{dt}$$

From equation 9

$$\frac{g_m}{c_g} q_1 = \frac{dq_2}{dt} \quad \left\{ \begin{array}{l} \text{substitute} \\ q_2 \text{ in this eqn} \end{array} \right.$$

$$\frac{g_m}{c_g} q_1 = \frac{d}{dt} \left[\frac{c_g}{g_m} \frac{dq_1}{dt} \right]$$

$$\left(\frac{g_m}{c_g} \right)^2 q_1 = \frac{d^2 q_1}{dt^2}$$

$$\therefore \frac{d^2 q_1}{dt^2} = \left(\frac{g_m}{c_g} \right)^2 q_1$$

$$\frac{d^2 q_1}{dt^2} = \frac{1}{c^2} q_1$$

where $c = \frac{c_g}{g_m}$

→ its time-domain solution is given by

$$q_1(t) = \frac{q_1(0) - c_0 q'_1(0)}{2} e^{-t/c_0} + \frac{q_1(0) + c_0 q'_1(0)}{2} e^{t/c_0}$$

$$q_1(0) = c_g v_{g1}(0) \rightarrow \text{"Initial Condition"}$$

$$v_{02}(t) = \frac{1}{2} \left[v_{02}(0) - c_0 v'_{02}(0) \right] e^{-t/c_0} + \frac{1}{2} \left[v_{02}(0) + c_0 v'_{02}(0) \right] e^{t/c_0}$$

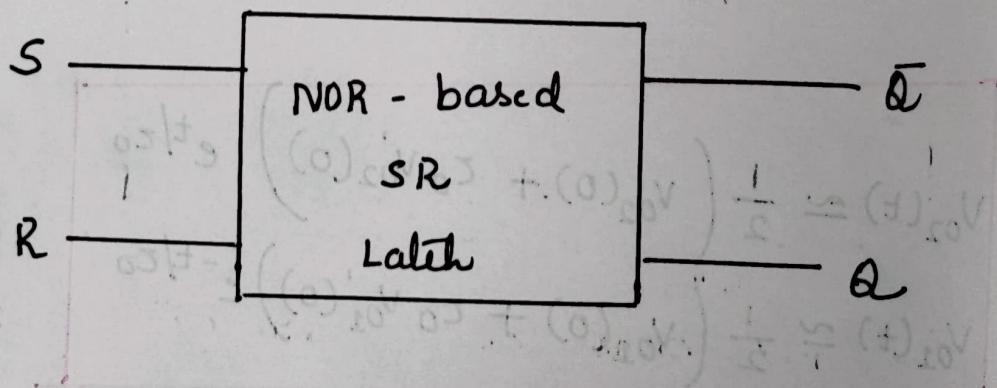
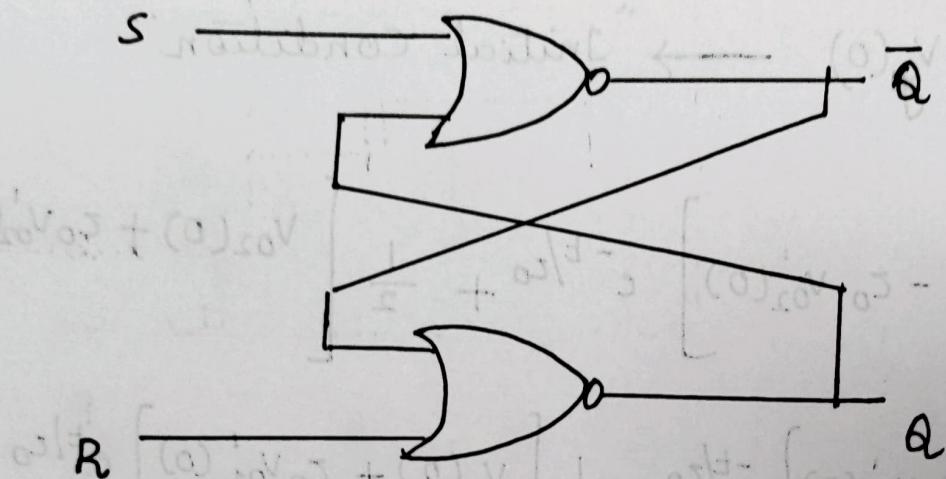
$$v_{01}(t) = \frac{1}{2} \left[v_{01}(0) - c_0 v'_{01}(0) \right] e^{-t/c_0} + \frac{1}{2} \left[v_{01}(0) + c_0 v'_{01}(0) \right] e^{t/c_0}$$

$$\boxed{v_{02}(t) \approx \frac{1}{2} \left(v_{02}(0) + c_0 v'_{02}(0) \right) e^{t/c_0}}$$

$$\boxed{v_{01}(t) \approx \frac{1}{2} \left(v_{01}(0) + c_0 v'_{01}(0) \right) e^{-t/c_0}}$$

SR Latch:

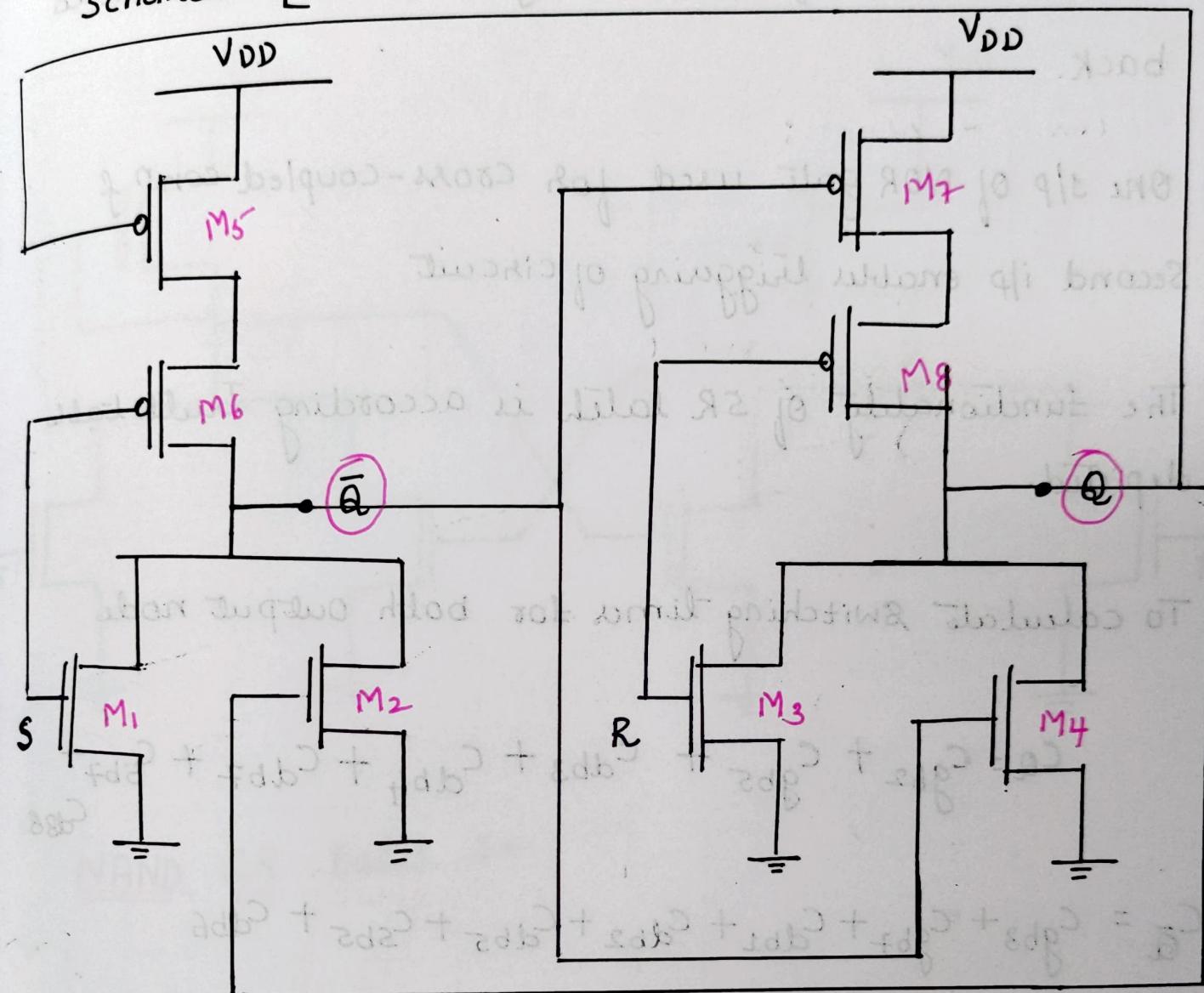
Logic diagram :-



Truth Table :-

S	R	Q	\bar{Q}	Operation
0	0	Previous state		Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	not - allowed

Schematische [CMOS]



- Bistable element consisting of two cross coupled inverters
- has two stable operating modes.
- The circuit preserve output as long as power supply is provided
- Simple inverter has no provision in applying external input
- Fig. shows simple S-R latch

- circuit consists of two NOR gates connected back-to-back.
- One input of NOR gate used for cross-coupled connection of second input enables triggering of circuit
- The functionality of SR latch is according to truth table depicted.
- To calculate switching times for both output nodes

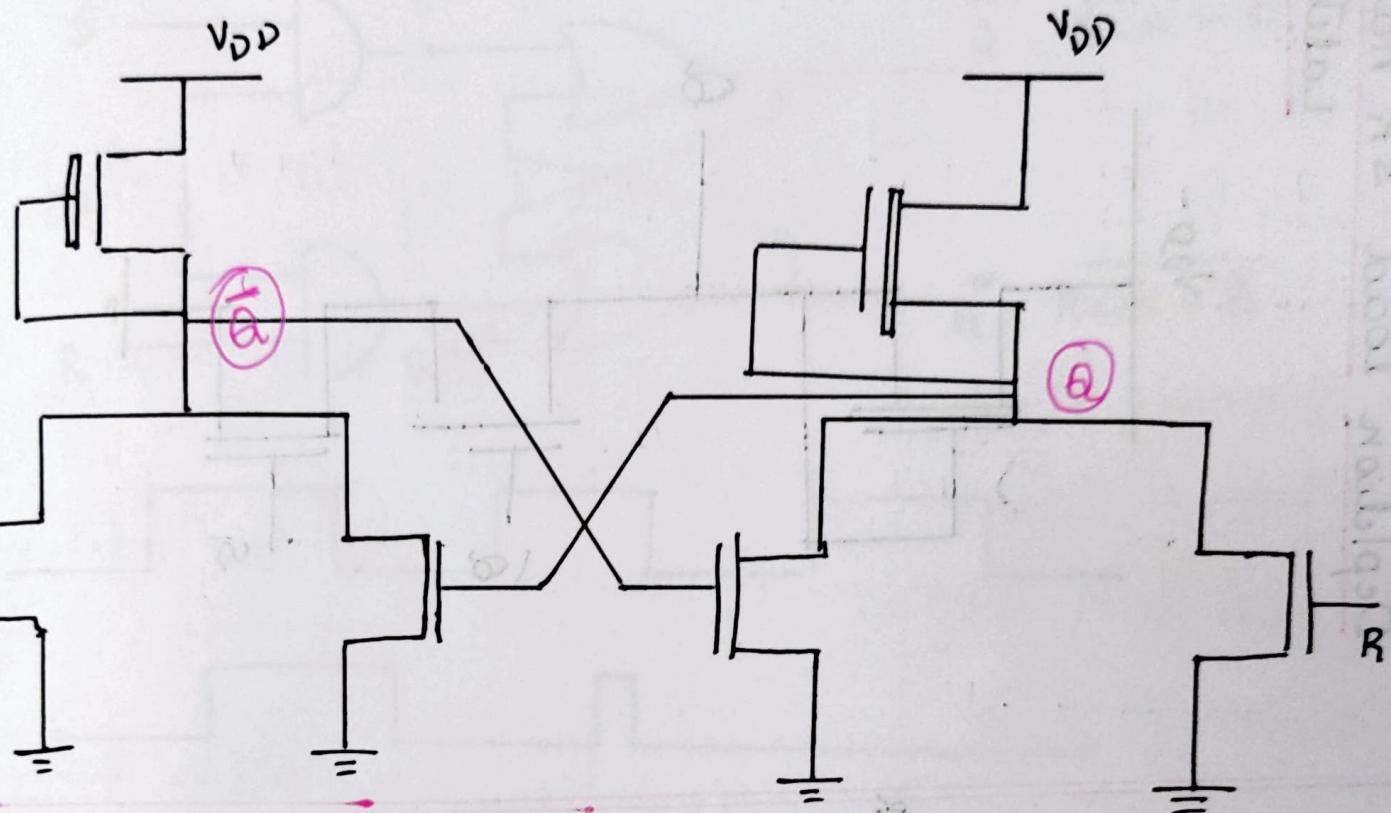
$$C_Q = C_{gb2} + C_{gb5} + C_{db3} + C_{db4} + C_{db7} + C_{sb7} + C_{db8}$$

$$C_{\bar{Q}} = C_{gb3} + C_{gb7} + C_{db1} + C_{db2} + C_{db5} + C_{sb5} + C_{db6}$$

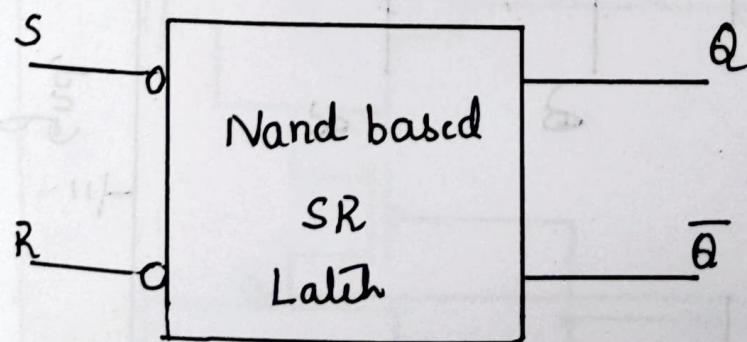
- rise-time associated with node Q is

$$T_{rise Q} (SR\text{-latch}) = T_{rise Q} (\text{NOR}_2) + T_{all \bar{Q}}^{(NOR)}$$

Depletion load nmos SR Latch



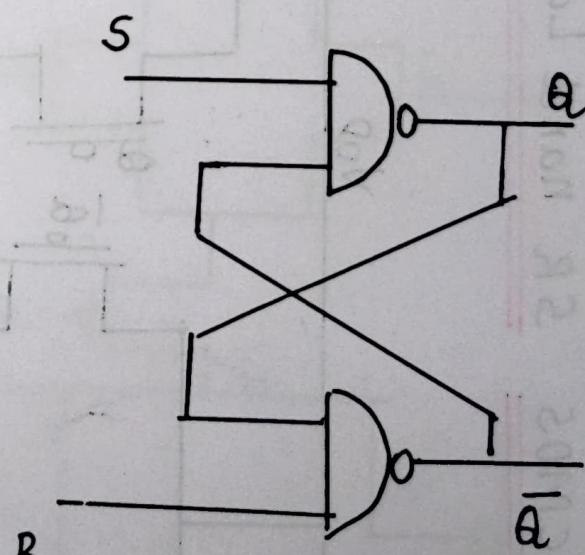
NAND SR latch :-



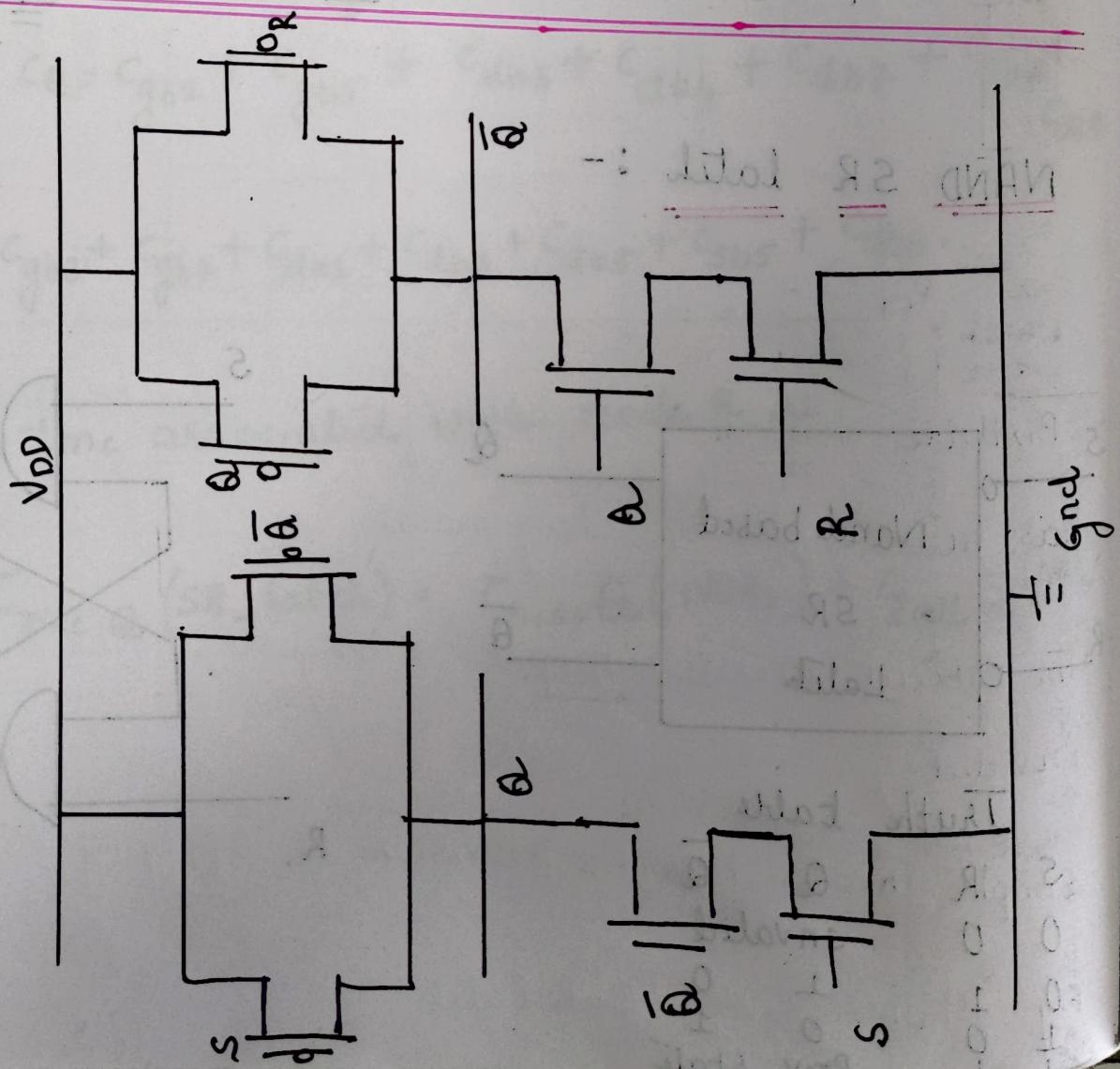
Truth table

S	R	Q	\bar{Q}
0	0	invalid	
0	1	1	0
1	0	0	1
1	1		

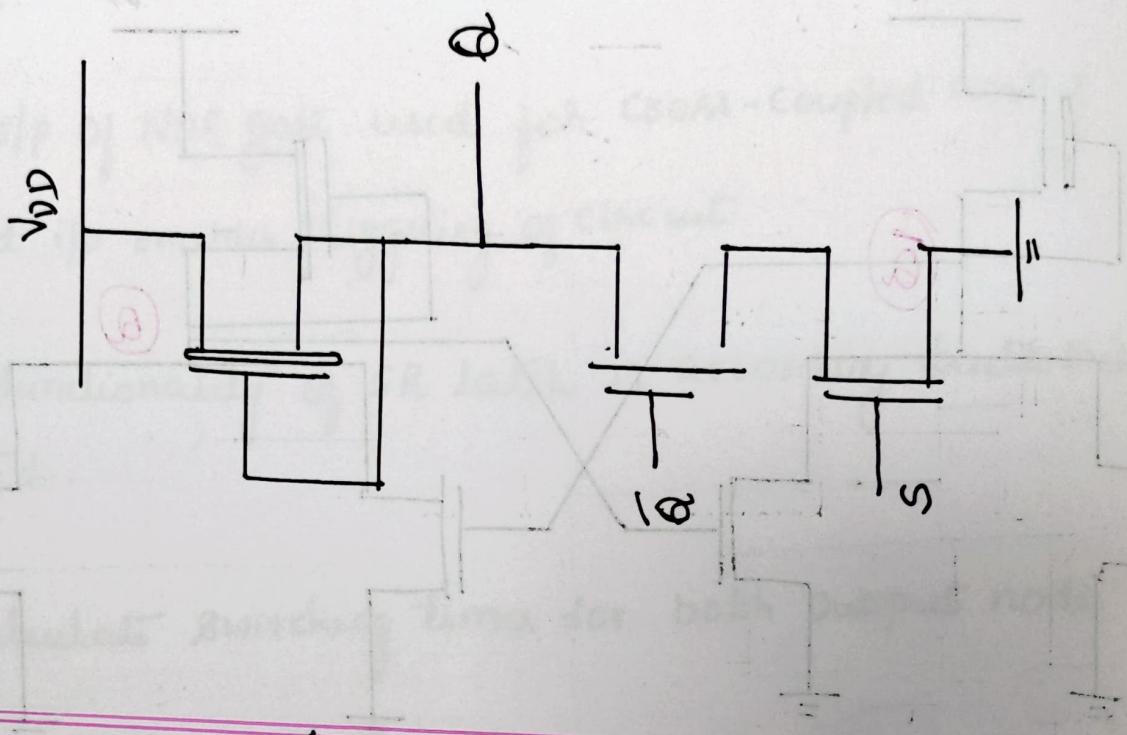
Prev. state



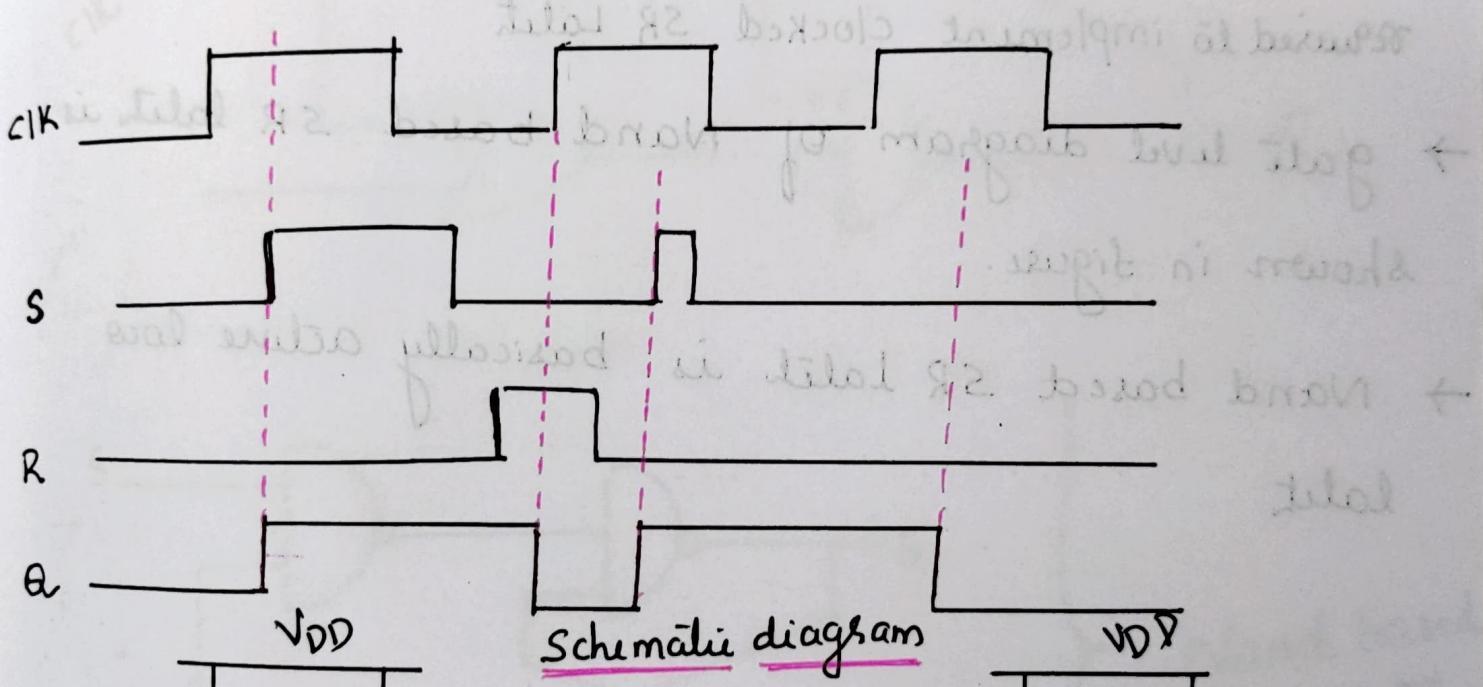
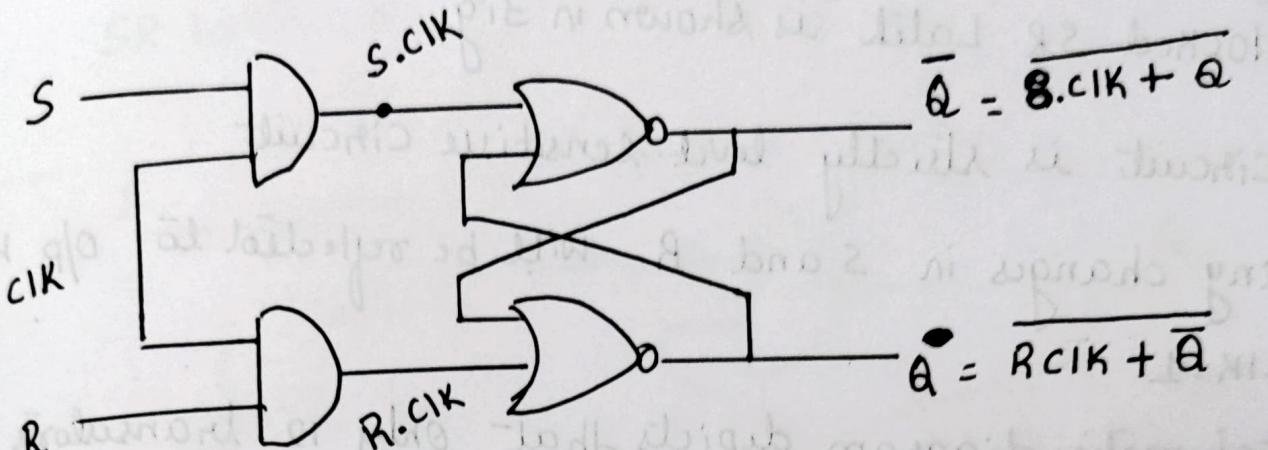
CMOS SR nand Latch



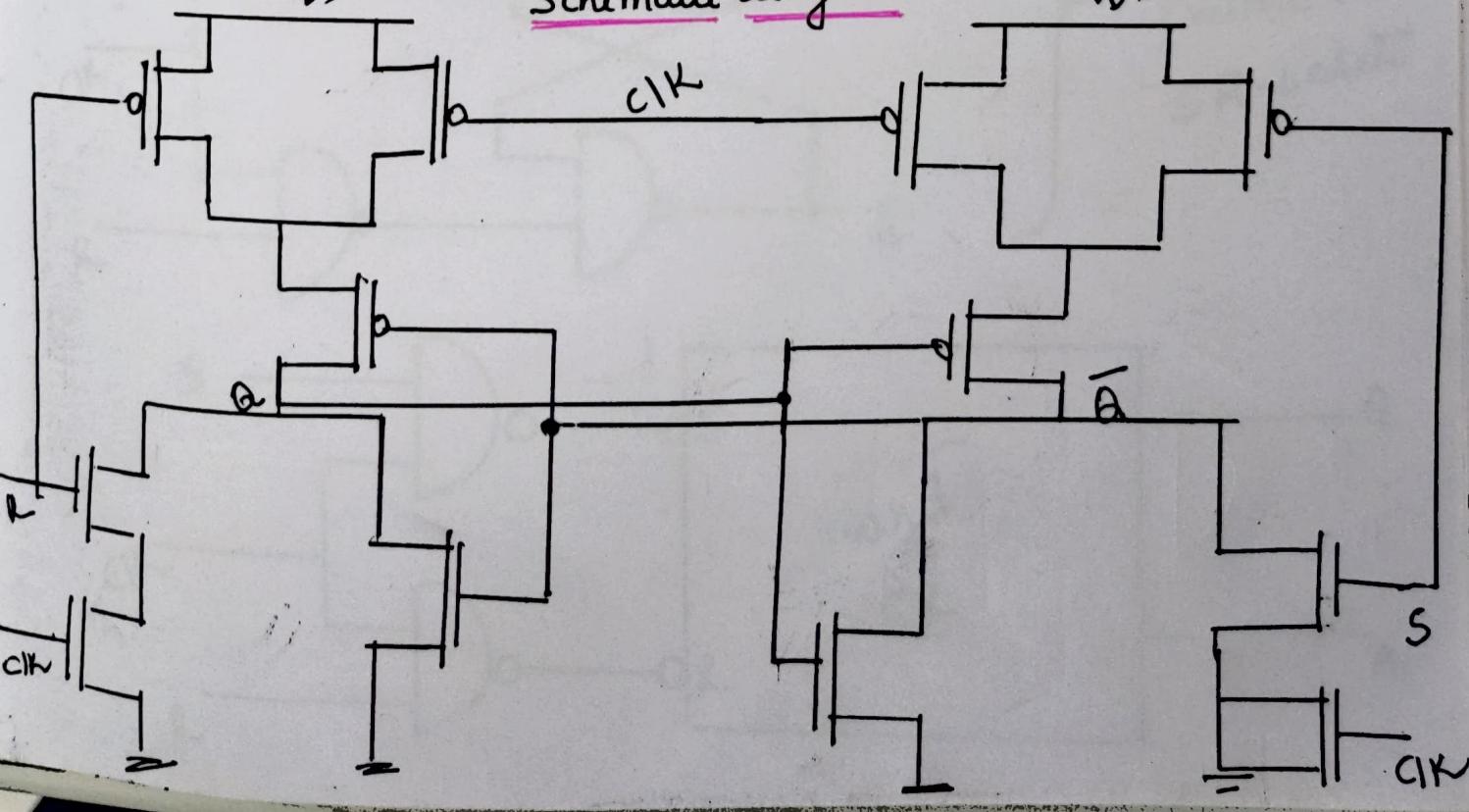
Depletion load SR nand Latch



Clocked SR Latch :-



Schematic diagram



→ Gati level diagram of CMOS Schematic of NOR based clocked SR Latch is shown in fig.

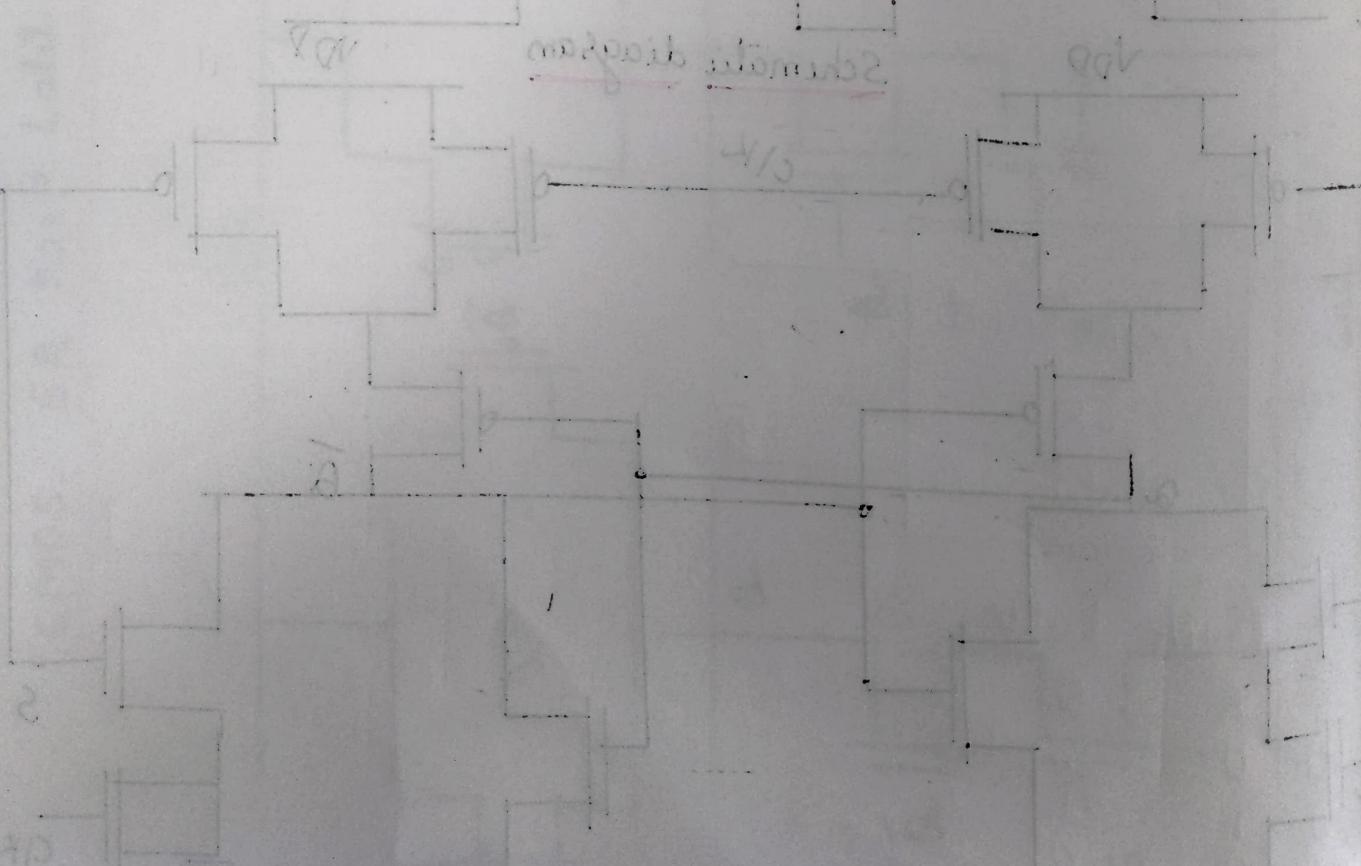
→ Circuit is strictly level sensitive circuit

→ any changes in S and R will be reflected to O/P when $Clk = 1$

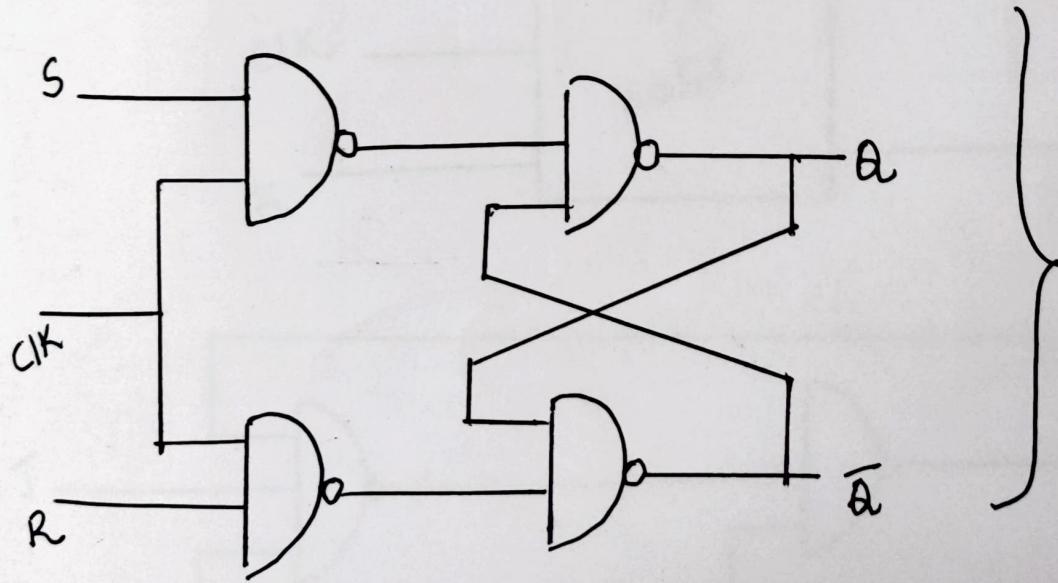
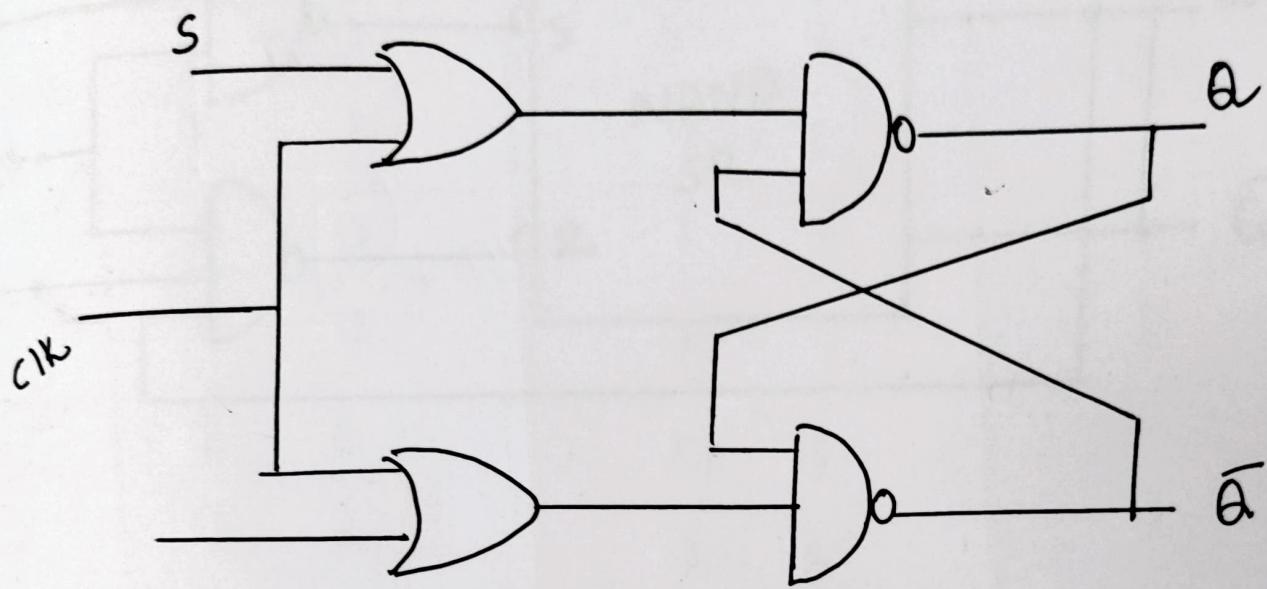
→ Schematic diagram depicts that only 12 transistors are required to implement clocked SR Latch

→ gati level diagram of Nand based SR Latch is shown in figure.

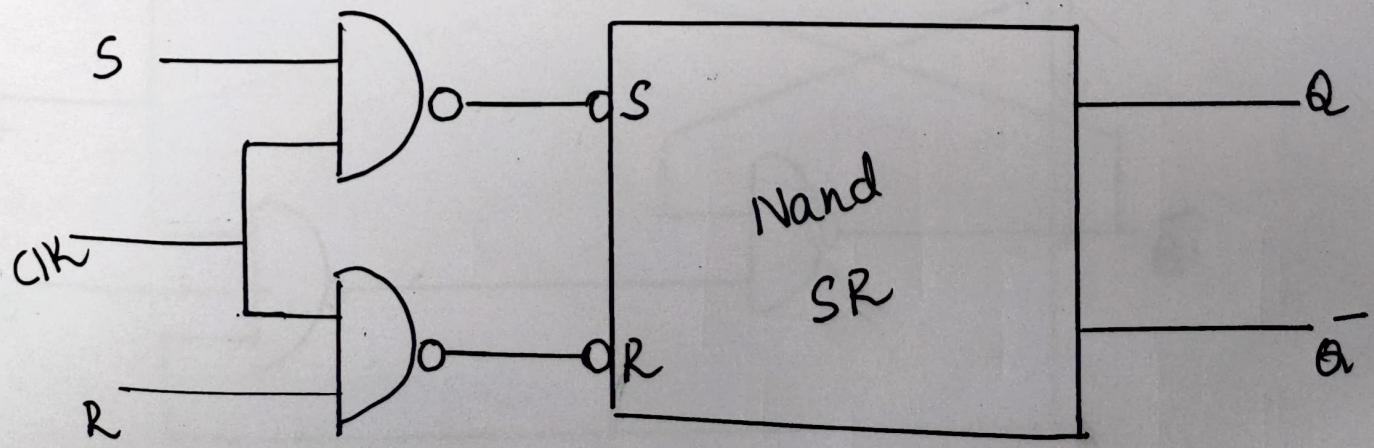
→ Nand based SR Latch is basically active low latch



gate-level schematic of clocked nand based
SR Latch :-

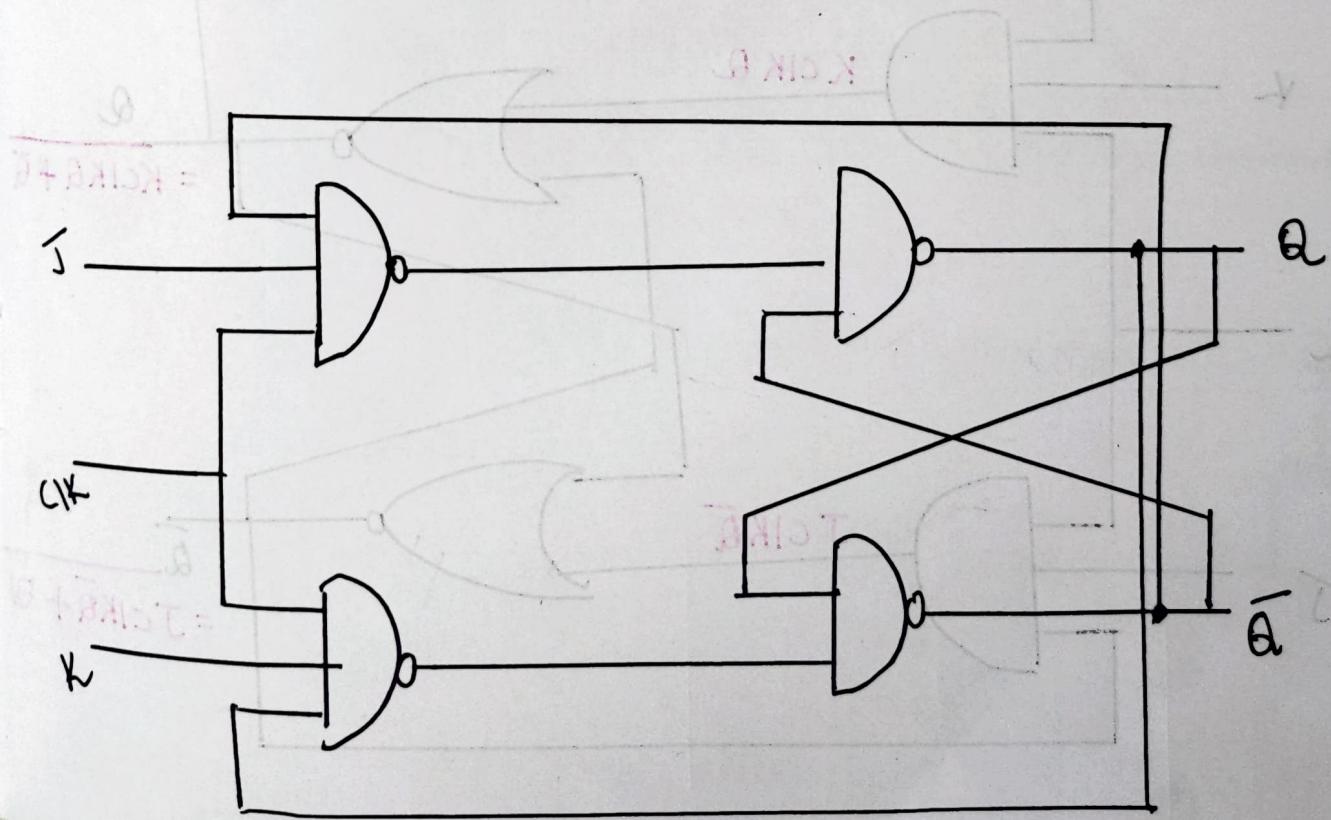
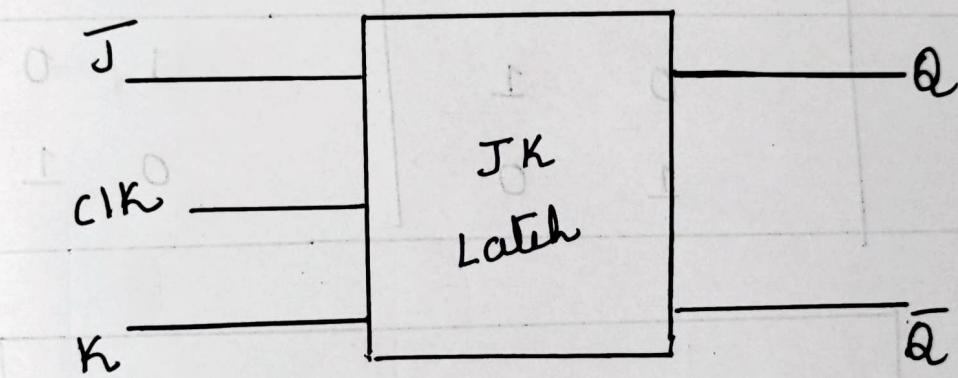
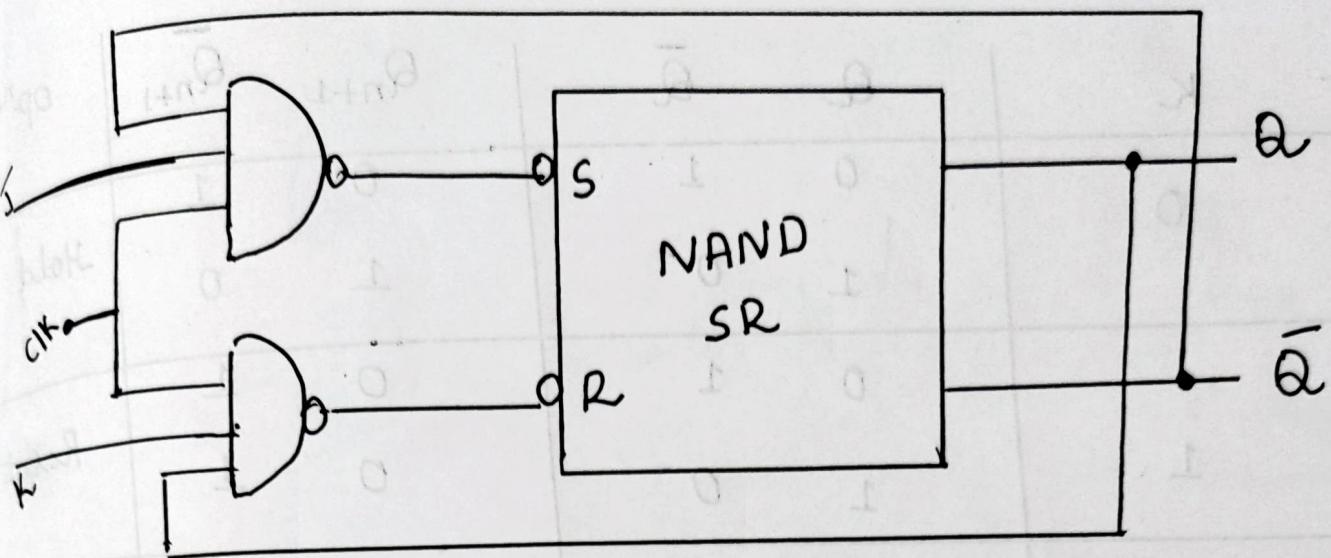


Nand based
SR-Latch



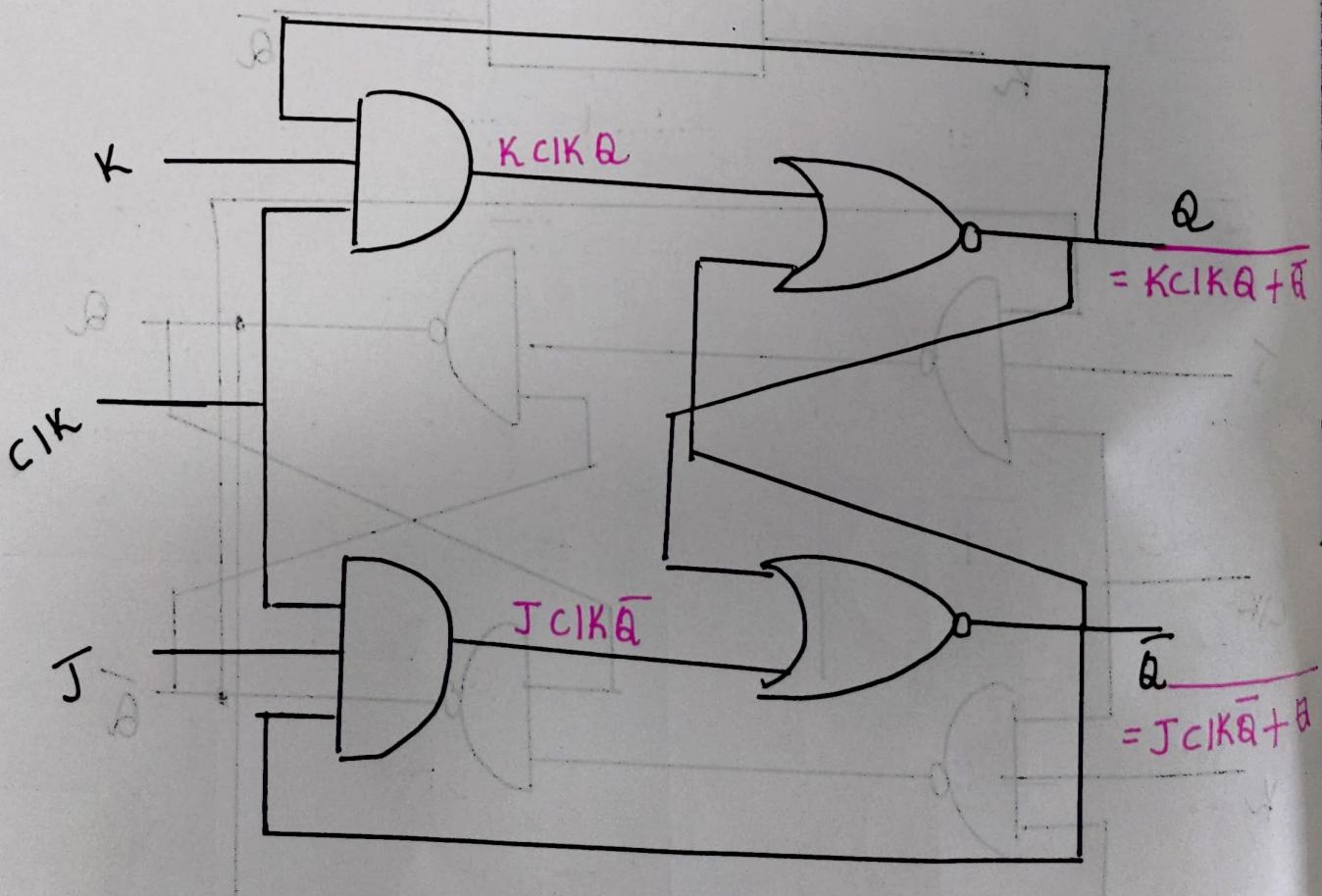
clocked JK Latch

- Ward Hunt

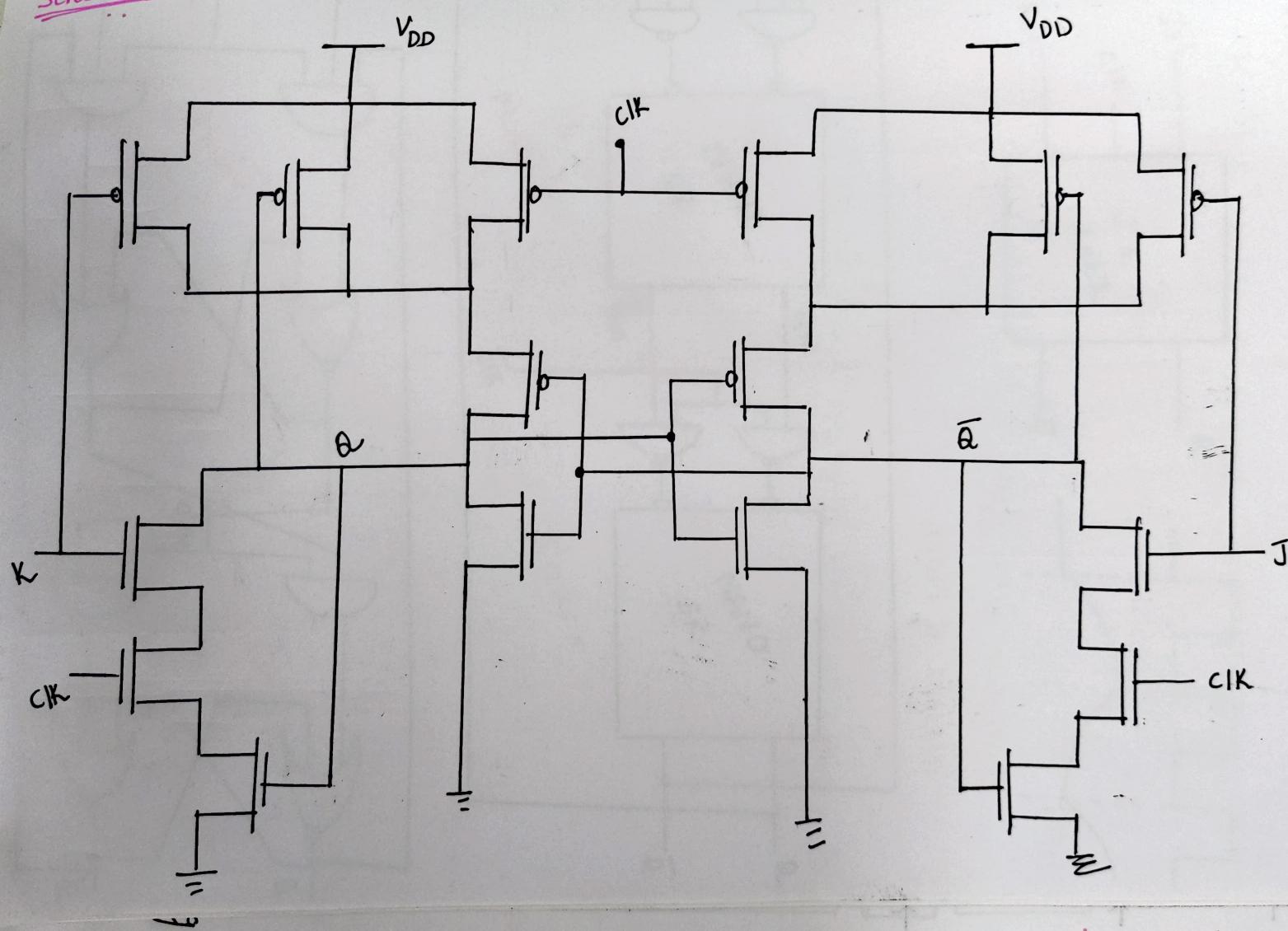


Truth Table :-

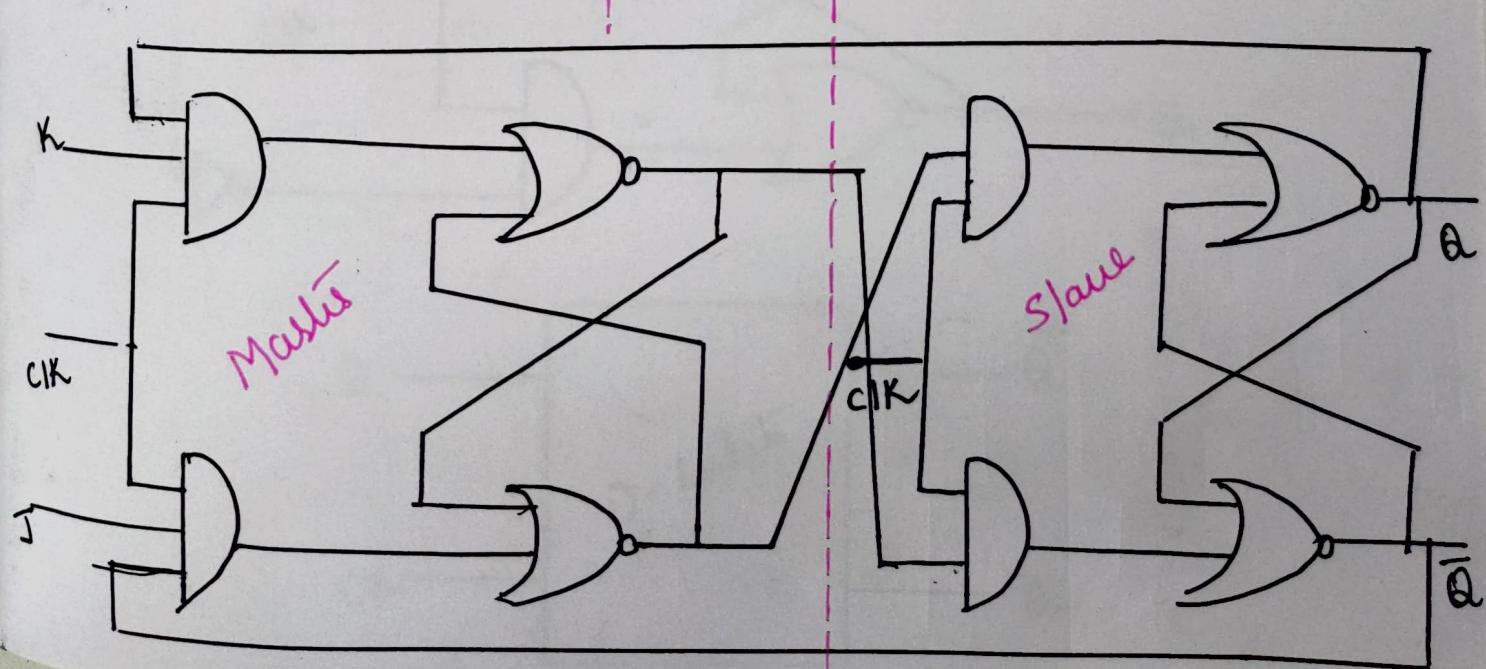
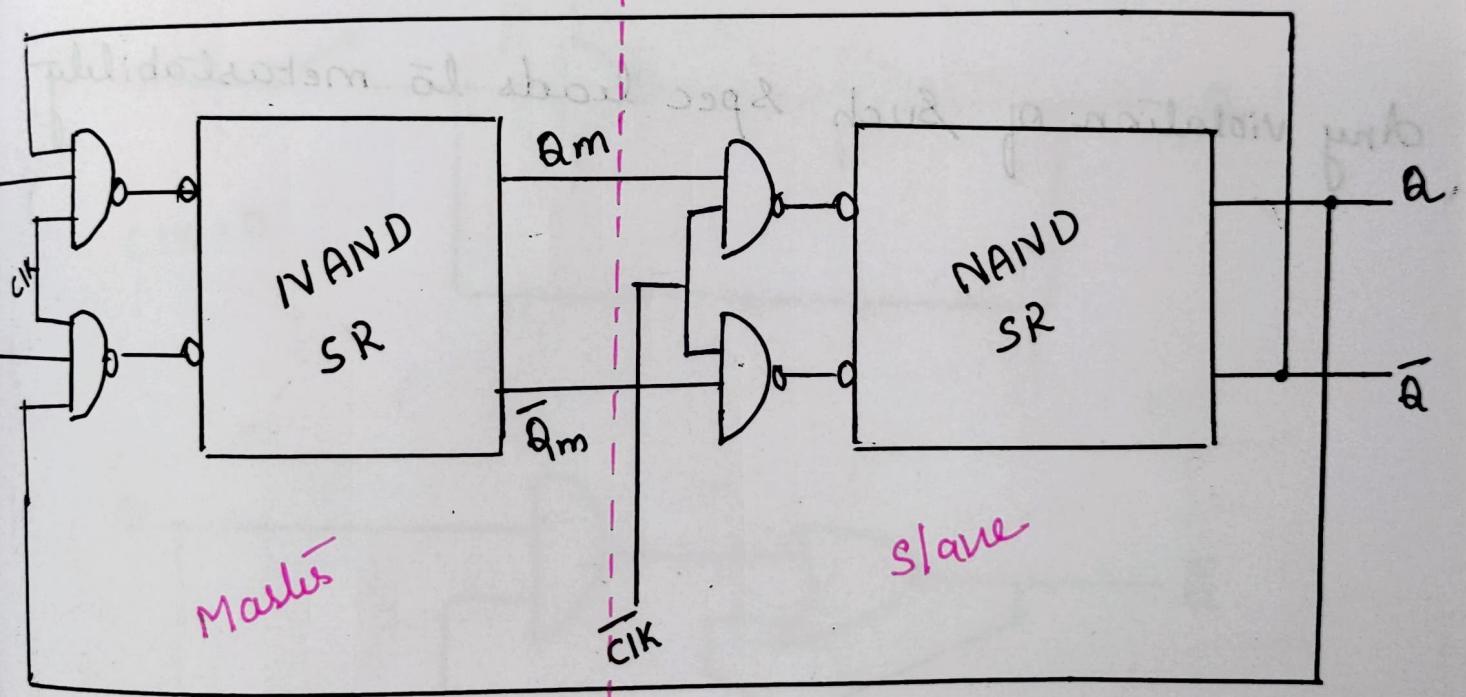
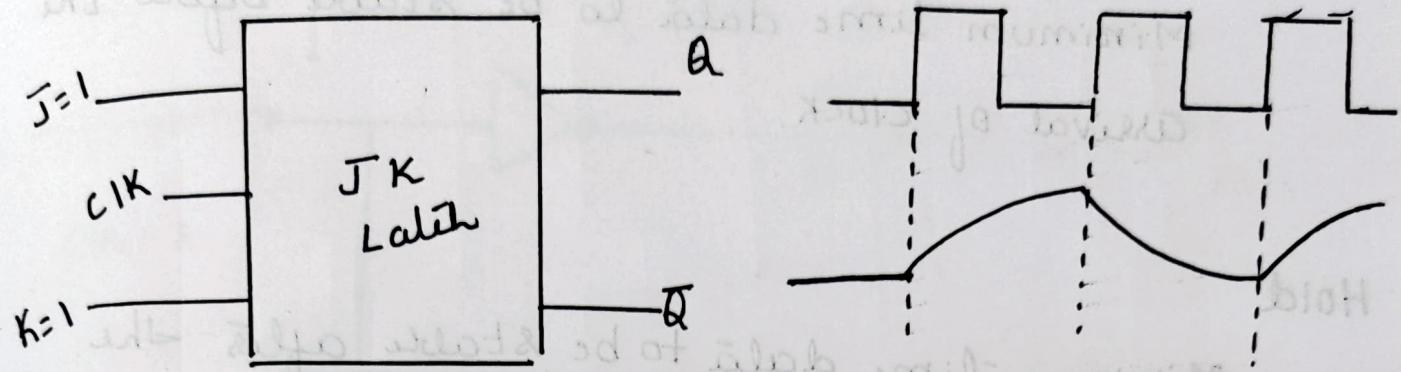
J	K	Q	\bar{Q}	Q_{n+1}	\bar{Q}_{n+1}	Op. ⁿ
0	0	0	1	0	1	Hold
		1	0			
0	1	0	1	0	1	Reset
		1	0			
1	0	0	1	1	0	Set
		1	0			
1	1	0	1	1	0	Toggle
		1	0			



Schemati



Master - slave FF



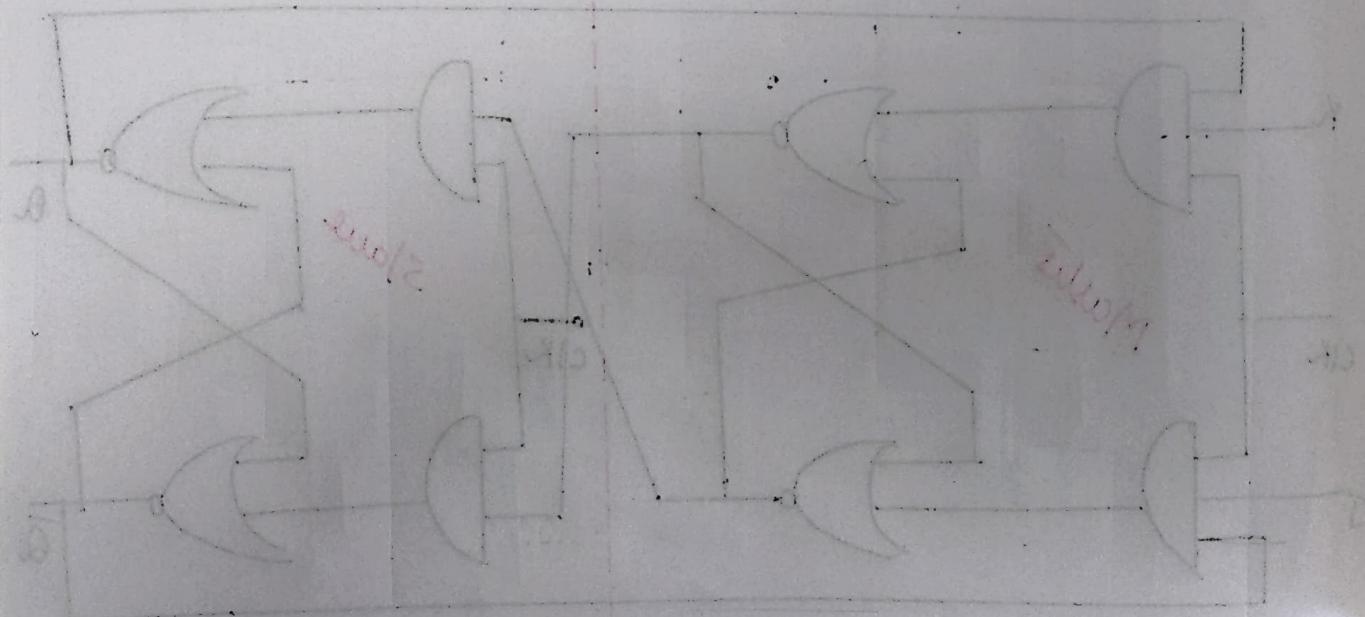
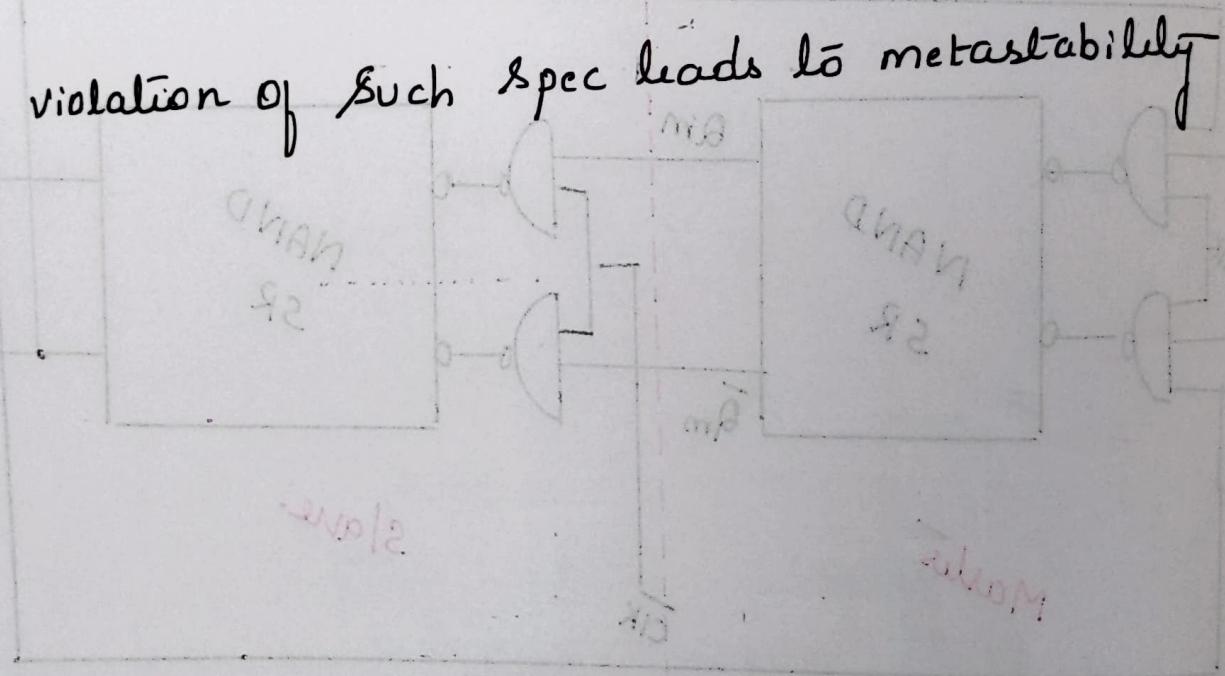
Setup =

Minimum-time data to be stable before the arrival of clock.

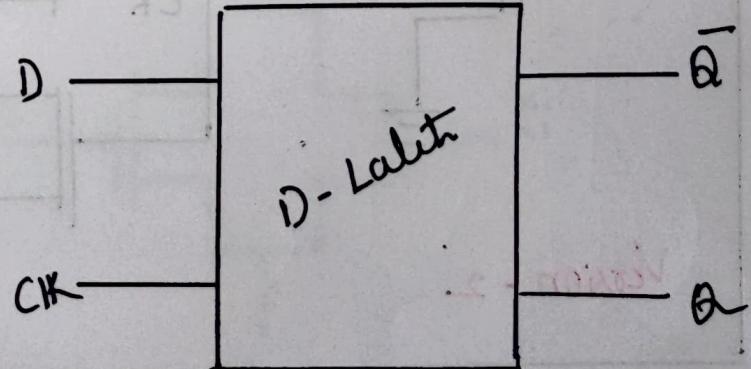
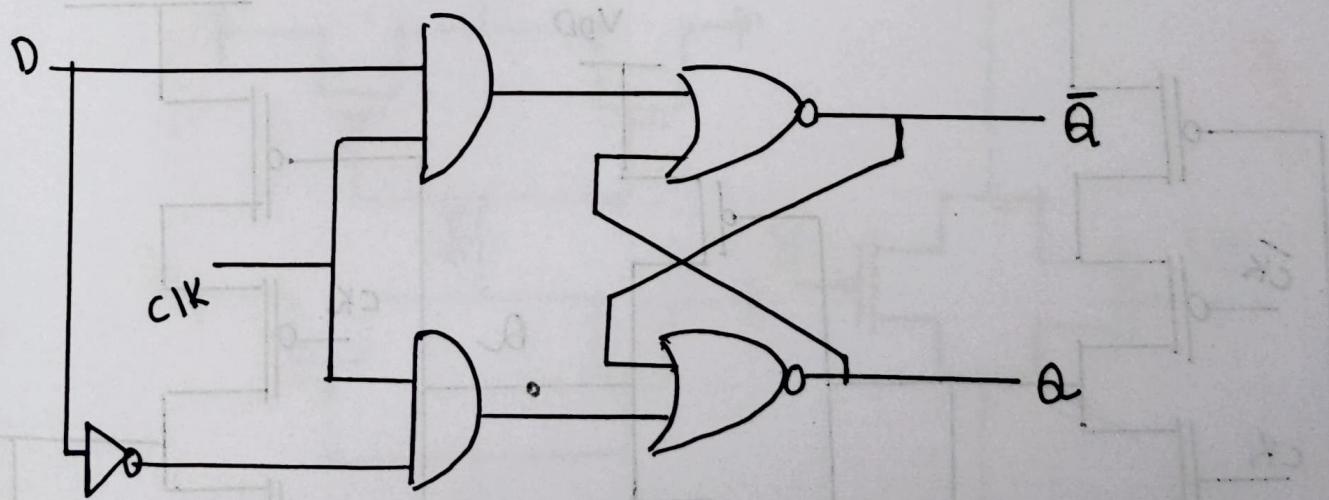
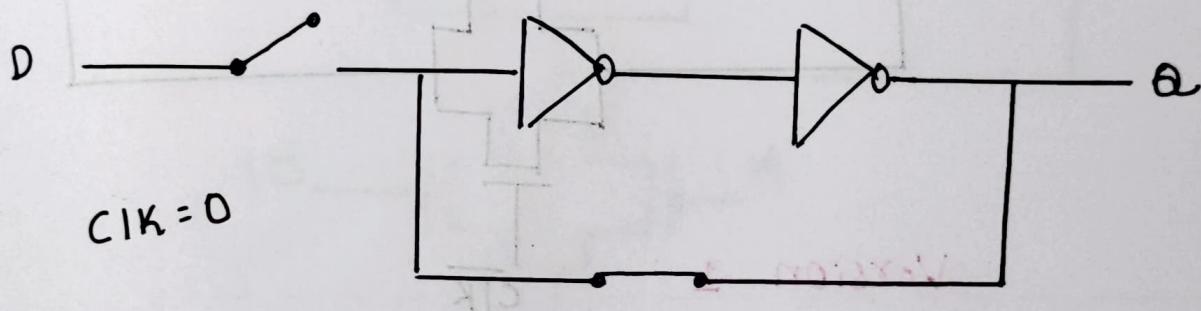
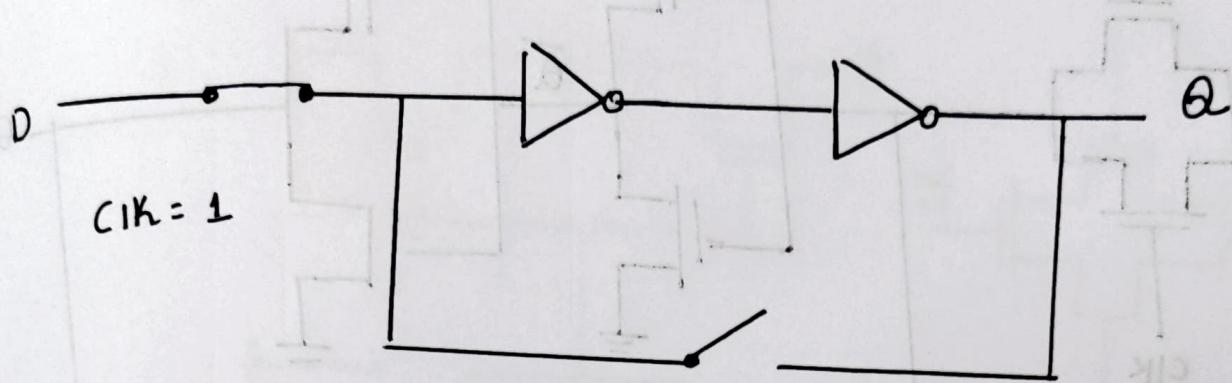
Hold

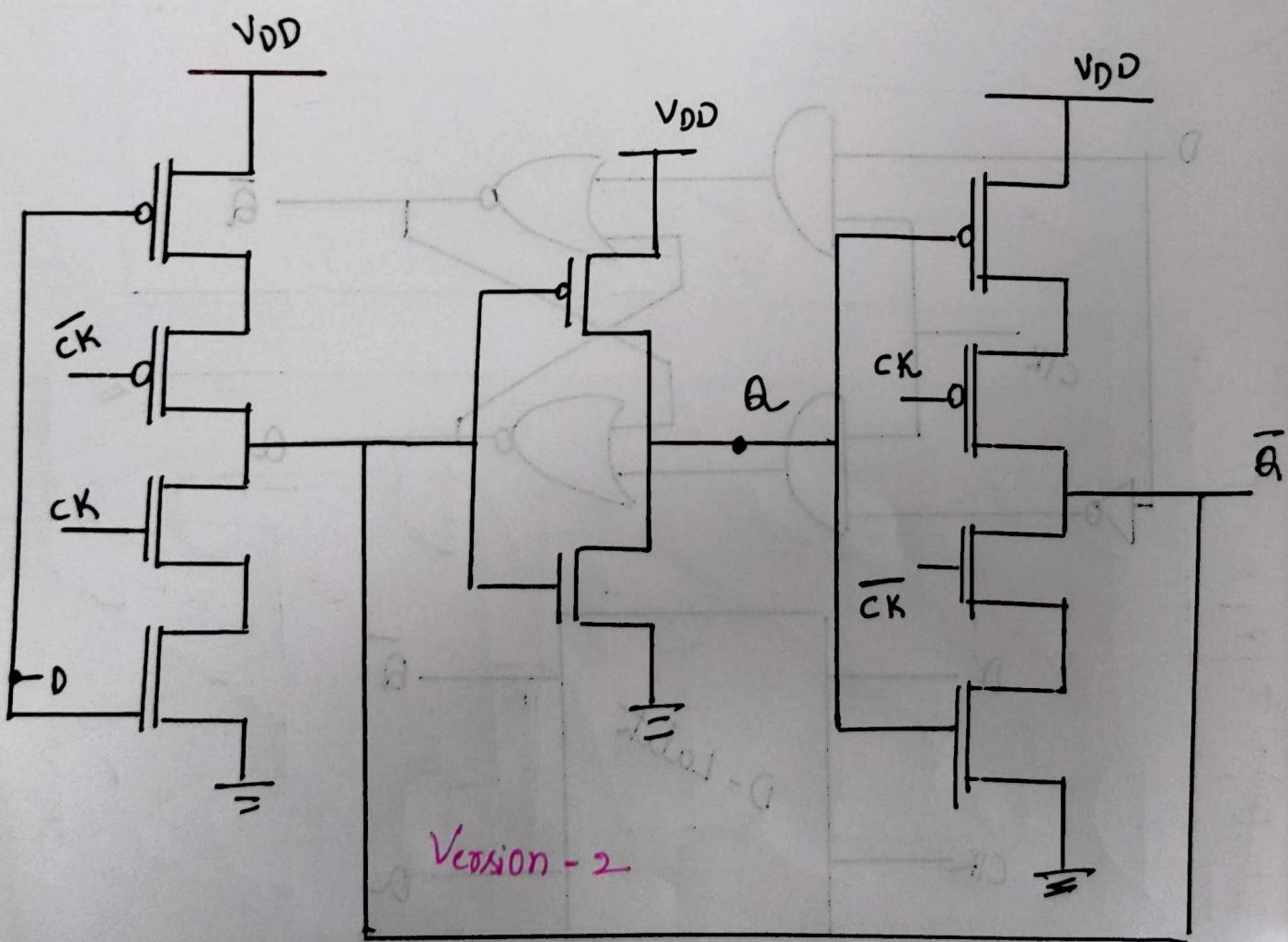
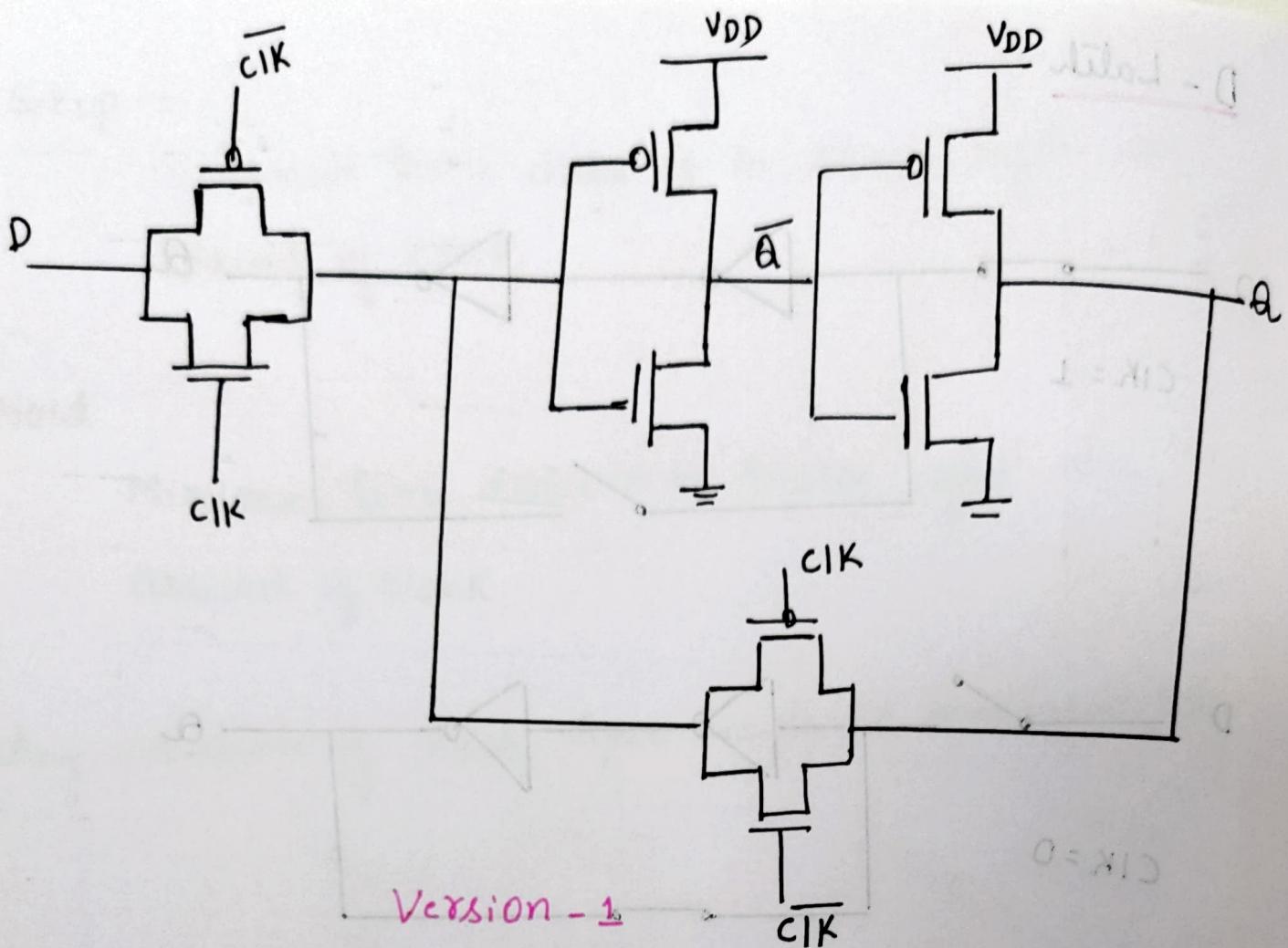
Minimum-time data to be stable after the arrival of clock.

Any violation of such spec leads to metastability

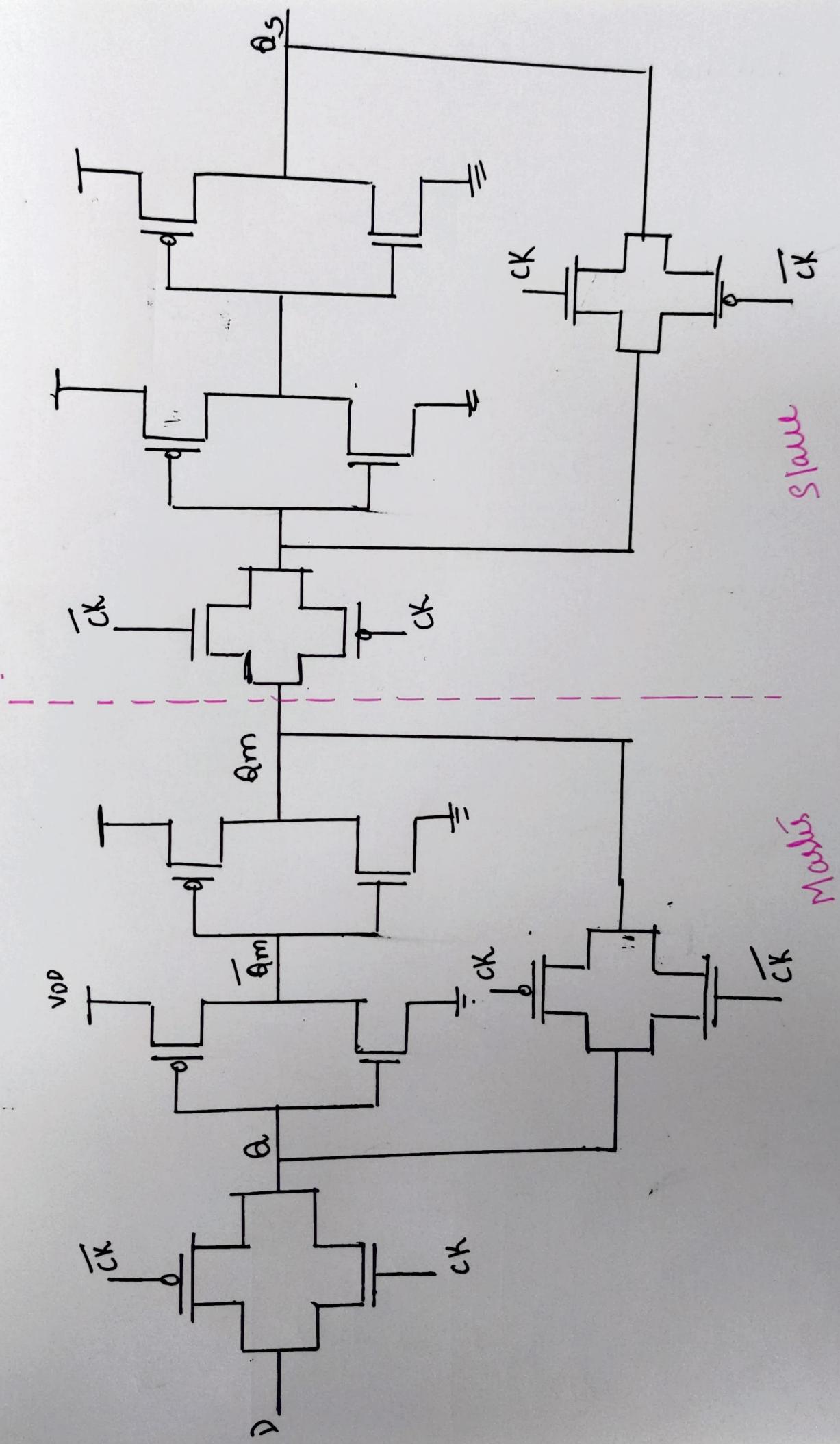


D - Latih





CMOS - V_C edge triggered



→ combinational circuit is one in which the output is a function of current inputs

→ Sequential circuit is one in which O/P depends on previous as well as current inputs.

Ex:- FSM
Pipeline.

→ Sequential circuit are usually designed with Flipflop or latches which are called memory elements

→ They hold by them are called "tokens"

→ The purpose of these elements is not really memory instead it is to enforce sequence to distinguish current token from previous token.

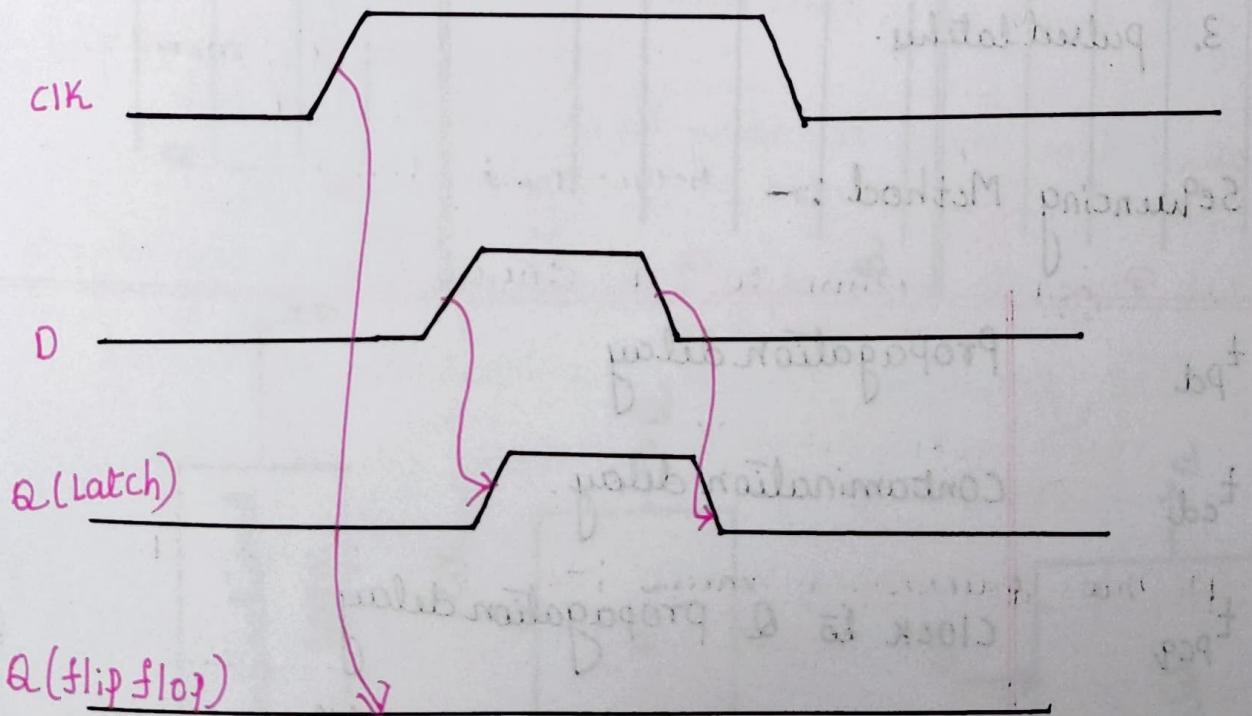
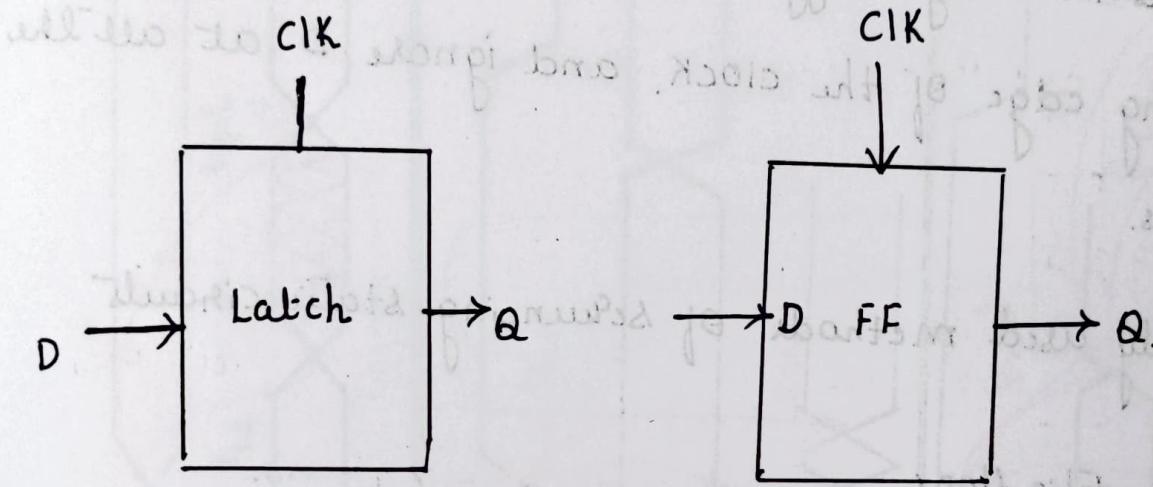
→ ∴ it is called sequencing elements.

→ With out sequencing elements the next token might catch up with previous token yields to garbage result.

→ Sequencing element delay token that arrive too early preventing them with catching up with previous token.

→ This extra delay is called Sequencing Overhead.

Latches and flip flops :-



→ Latches and flip flops are most commonly used sequencing elements

→ both have 3 terminals D, CLK, and Q

- Latch is transparent when clock is high and opaque when CLK is low.
- FF is an edge triggered device copies $D \rightarrow Q$ on rising edge of the clock. and ignore D at all the times.
- Widely used methods of sequencing static circuits

1. Flip flops
2. Two-phase transparent latches
3. pulsed latches.

Sequencing Method :-

t_{pd}

Propagation delay

t_{cd}

Contamination delay.

t_{pcq}

Clock to Q propagation delay

t_{ccq}

Clock to Q contamination delay

t_{pdq}

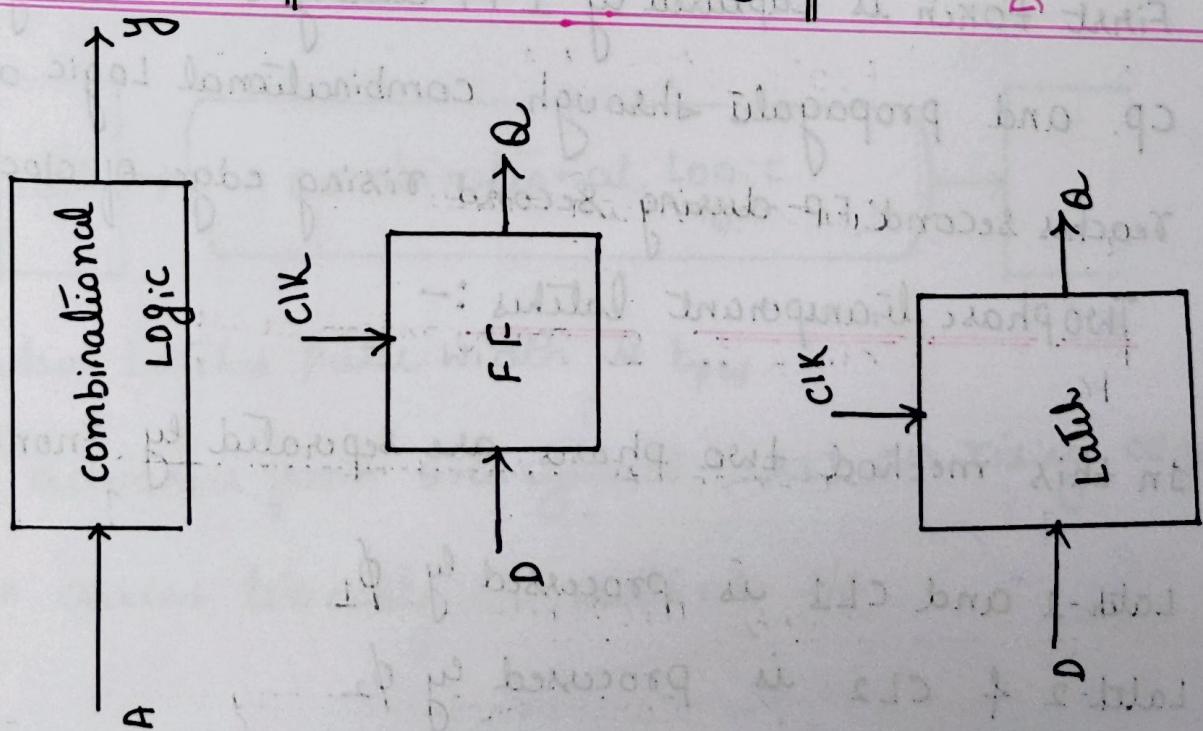
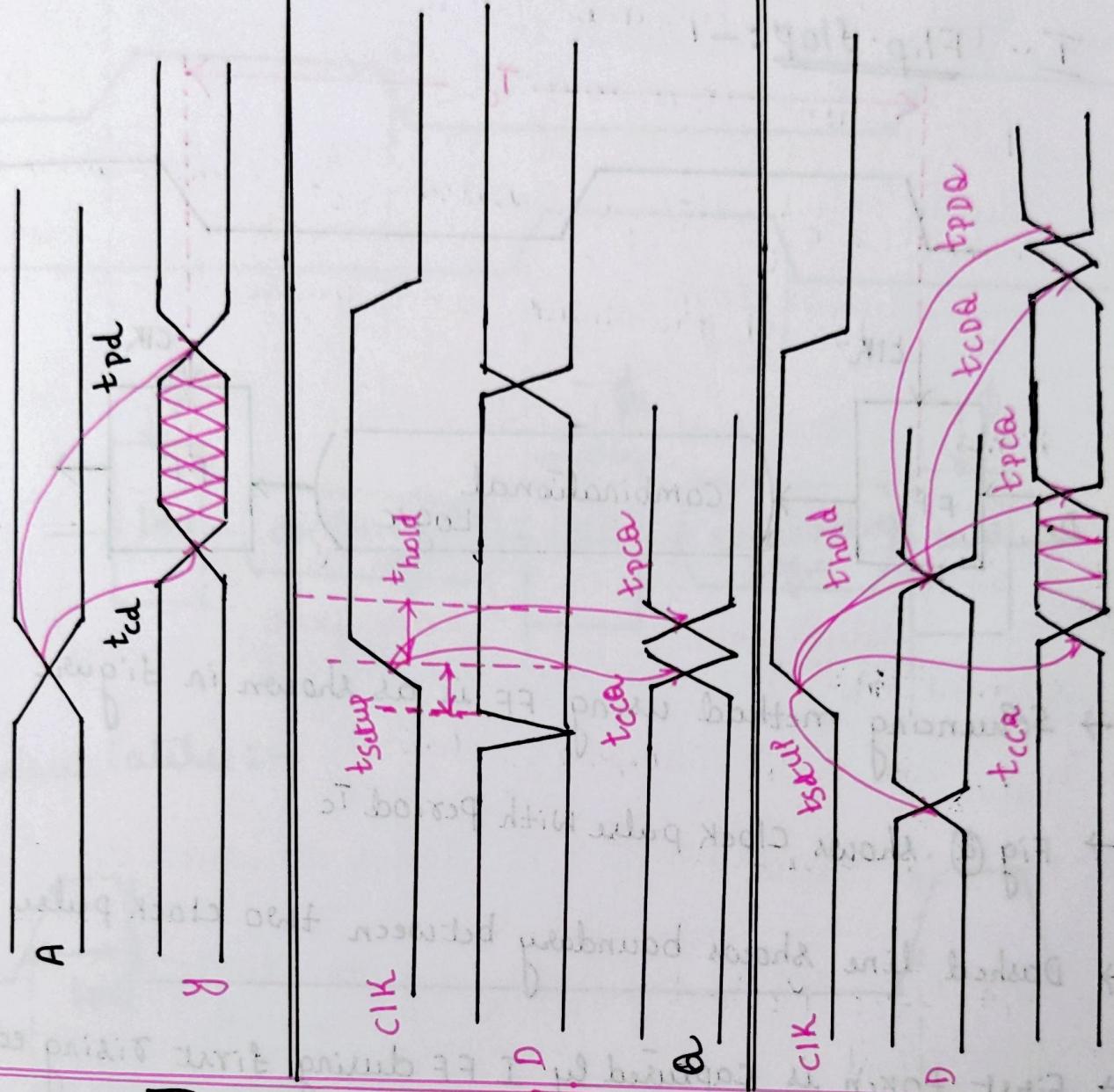
$D \rightarrow Q$ propagation delay

t_{cdq}

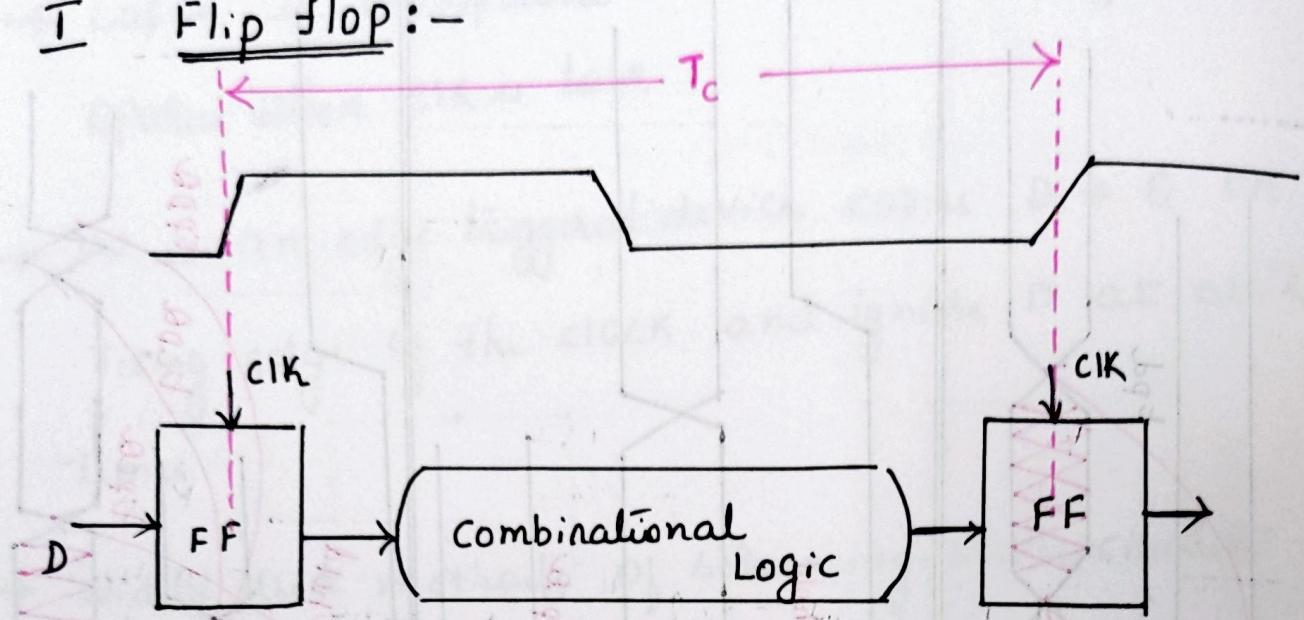
$D \rightarrow Q$ contamination delay

t_{setup} t_{hold}

Set up & hold time.



I Flip flop :-



→ Sequencing method using FF is as shown in figure

→ Fig @ shows clock pulse with period T_C

→ Dashed line shows boundary between two clock pulses

→ First token is captured by 1st FF during first rising edge of cp. and propagates through combinational logic and reaches second FF during second rising edge of clock.

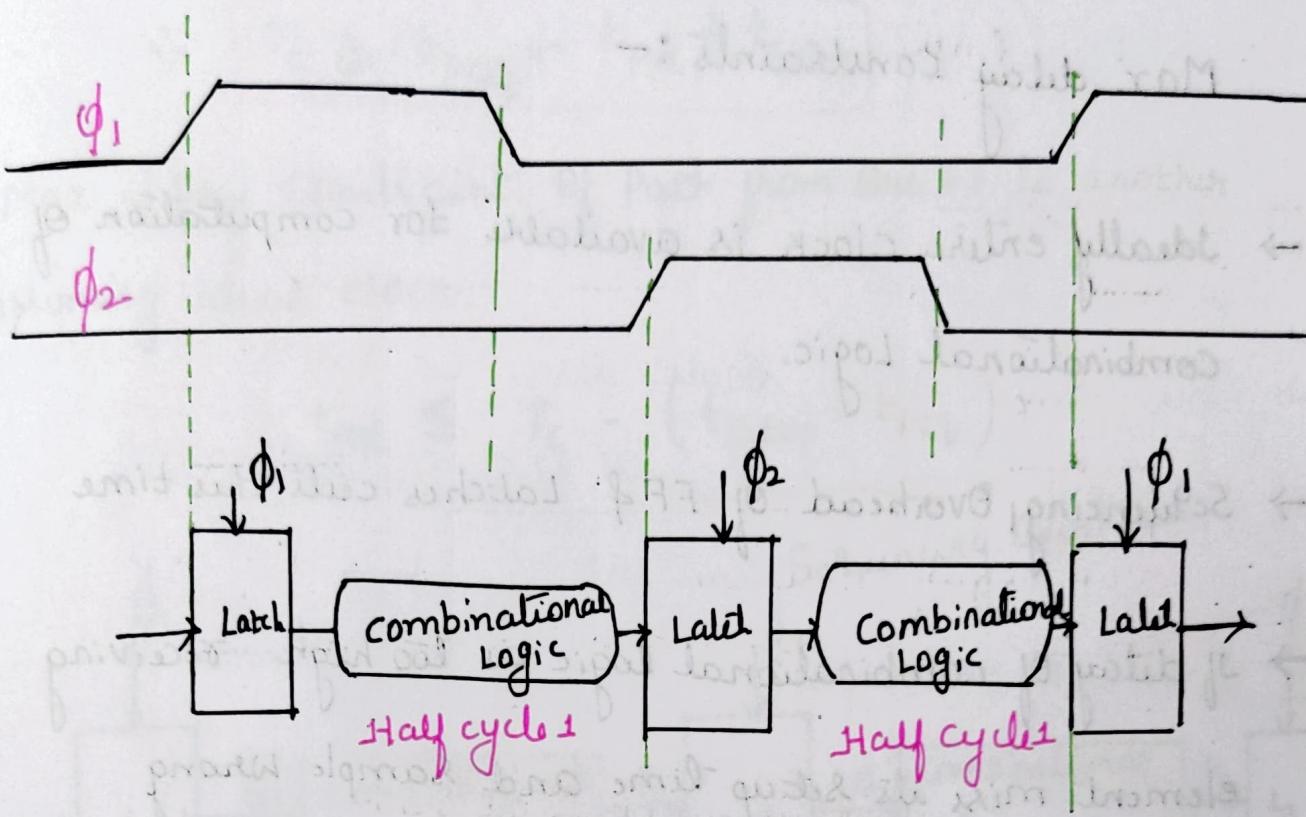
II Two phase binapentant latches :-

→ In this method two phases are separated by $t_{nonoverlap}$

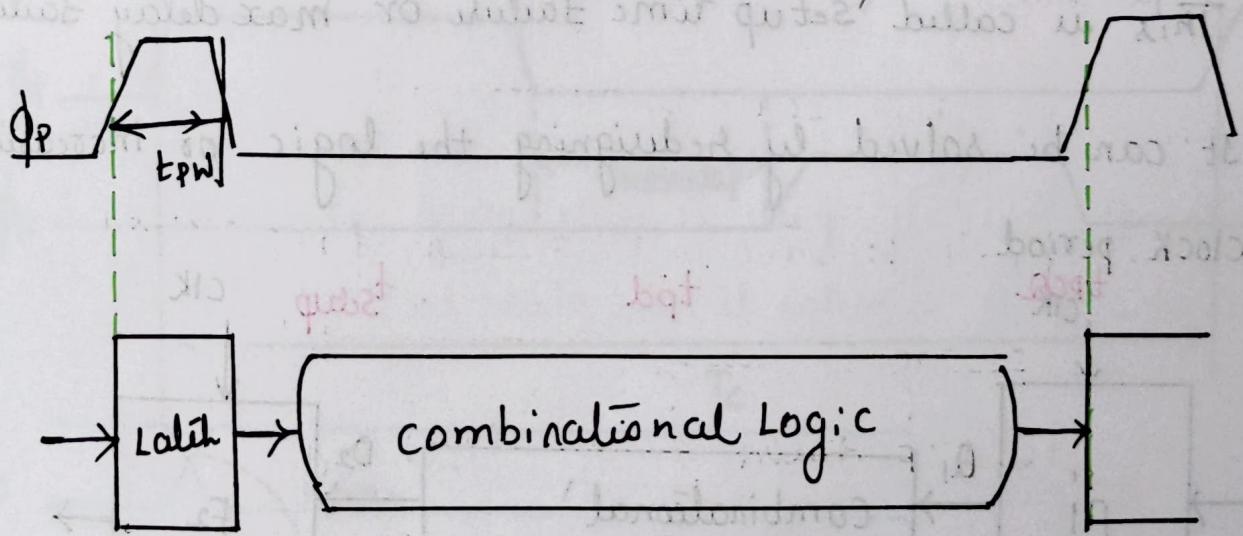
→ Latch-1 and CL1 is processed by ϕ_1

→ Latch-2 & CL2 is processed by ϕ_2

→ Data should reach latch 3 before the arrival of rising edge ϕ_1



III Pulsed Latches :-



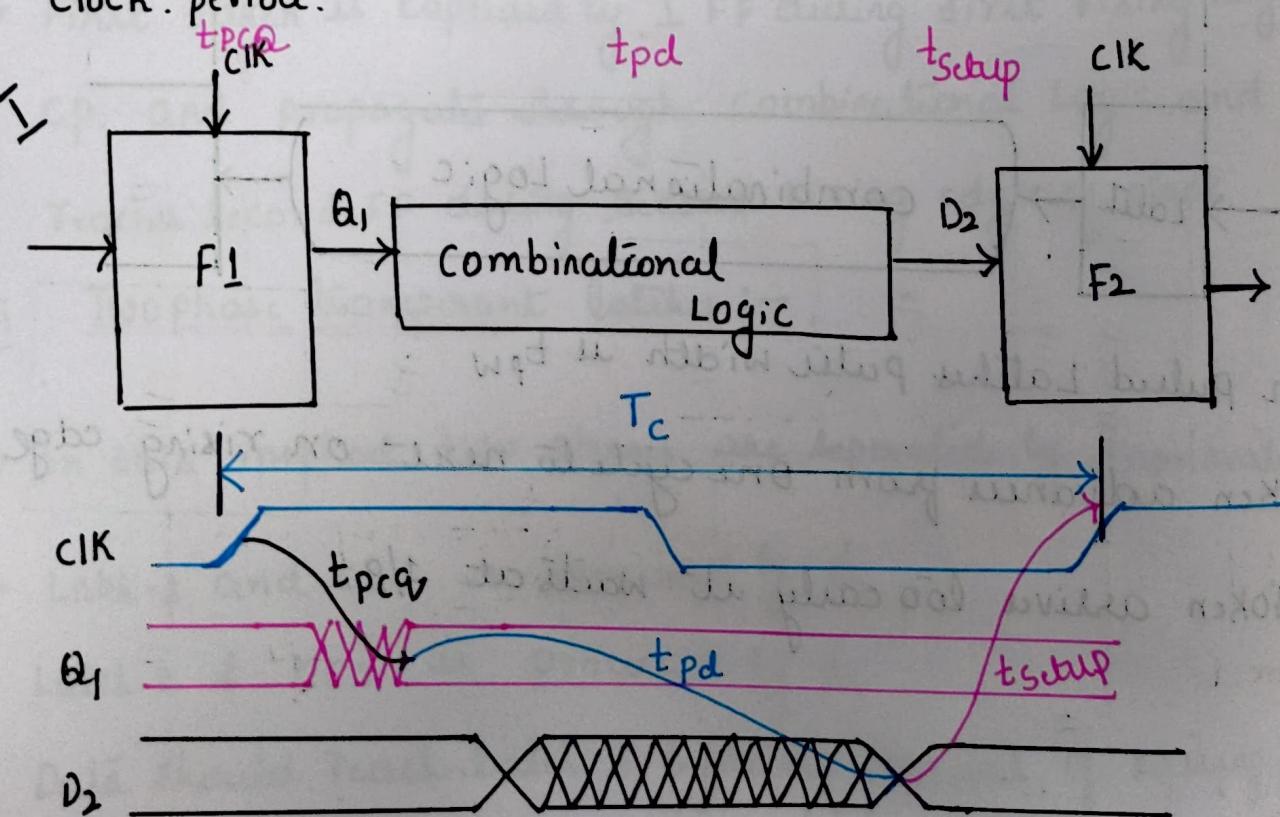
→ In pulsed Latches pulse width is t_{pw}

→ Token advances from one cycle to next on rising edge

→ If token arrives too early it waits at $1/p$

Max delay constraints :-

- Ideally entire clock is available for computation of combinational logic.
- Sequencing overhead of FF & Latches cut this time.
- If delay of combinational logic is too high receiving element misses its setup-time and samples wrong value.
- This is called setup time failure or max delay failure.
- It can be solved by redesigning the logic or increasing clock period.

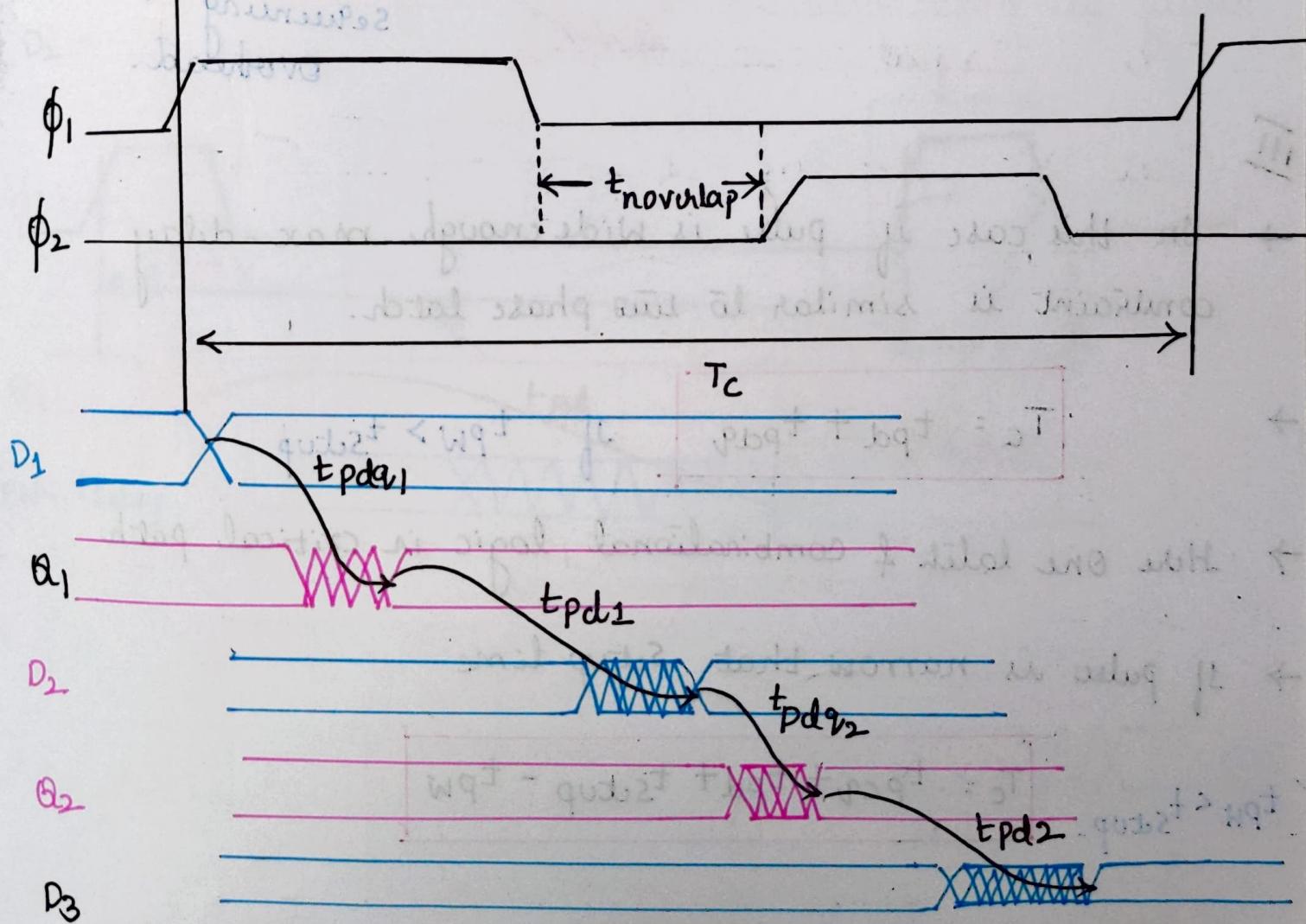
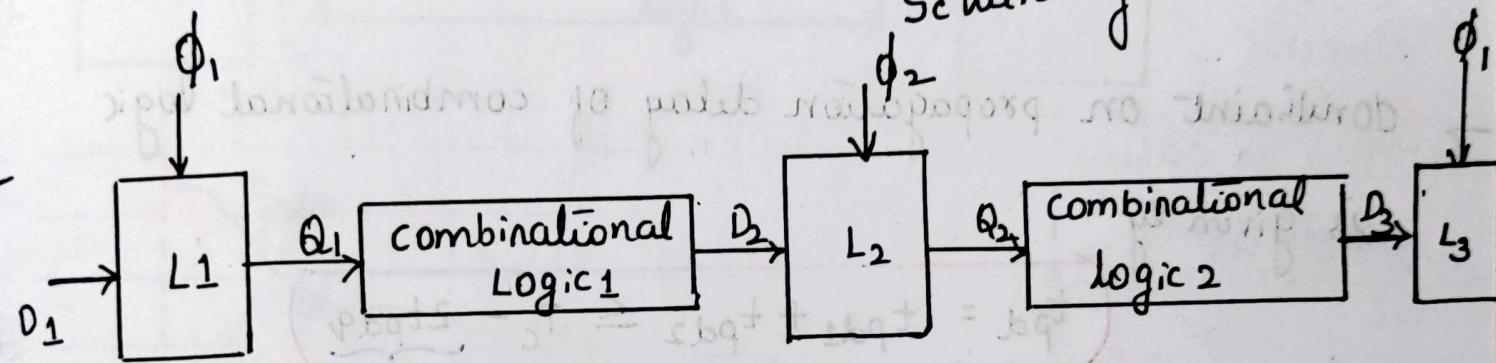


$$\therefore T_c \geq t_{pcq} + t_{pd} + t_{\text{setup}}$$

→ Max delay constraint of Path from one FF to another assuming ideal clock.

$$t_{pd} \leq T_c - (t_{\text{setup}} + t_{pcq})$$

Sequencing Overhead.



$$T_c \geq t_{pdq_1} + t_{pd_1} + t_{pdq_2} + t_{pd_2}$$

- Sequencing method using two phase transparent latch method is shown in figure.
- Here data at D_3 could arrive as late as setup time before falling edge of ϕ_1 and still captured correctly.
- constraint on propagation delay of combinational logic is given by

$$t_{pd} = t_{pd_1} + t_{pd_2} \leq T_c - \underline{2t_{pdq}}$$

sequencing
overhead

III

- In this case if pulse is wide enough max-delay constraint is similar to one phase latch.

$$\rightarrow T_c = t_{pd} + t_{pdq} \quad \text{if } t_{pw} > t_{setup}$$

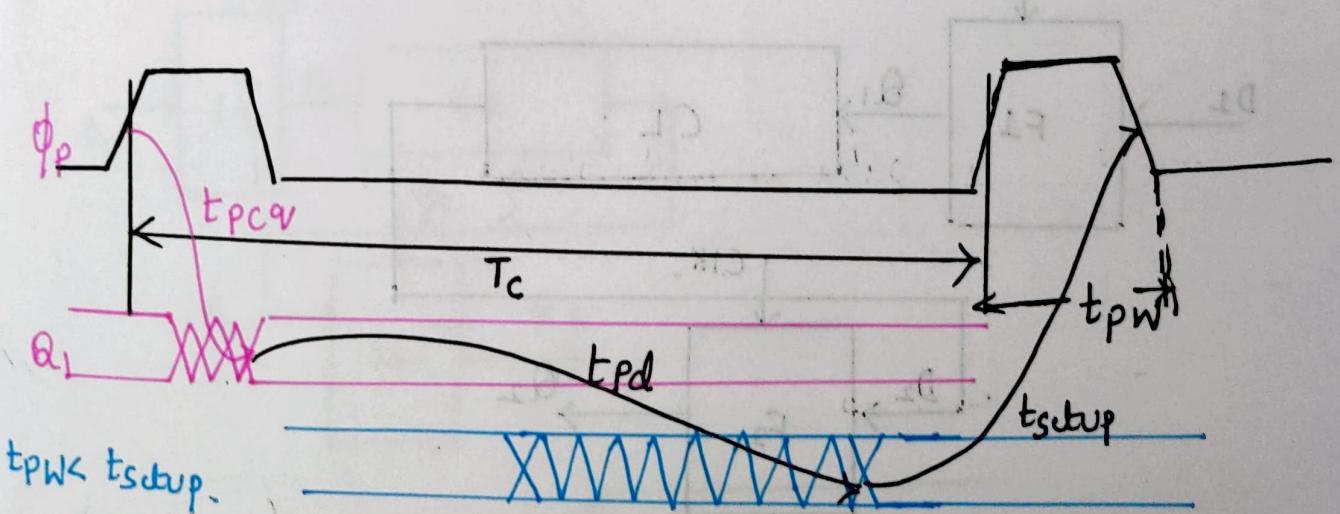
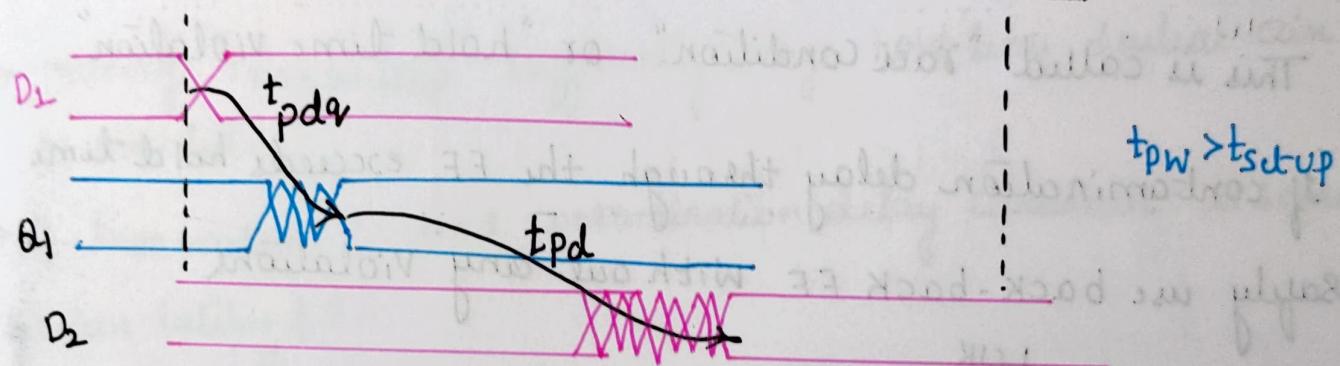
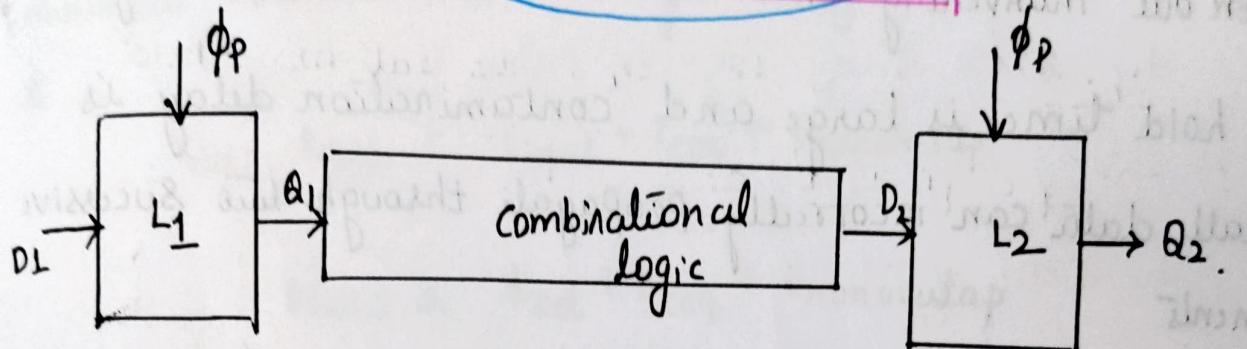
- Here one latch & combinational logic in critical path
- If pulse is narrow than setup time

$$t_{pw} < t_{setup} \quad T_c = t_{pdq} + t_{pd} + t_{setup} - t_{pw}$$

$$t_{pd} < T_c - t_{pdq} \rightarrow t_{pw} > t_{setup}$$

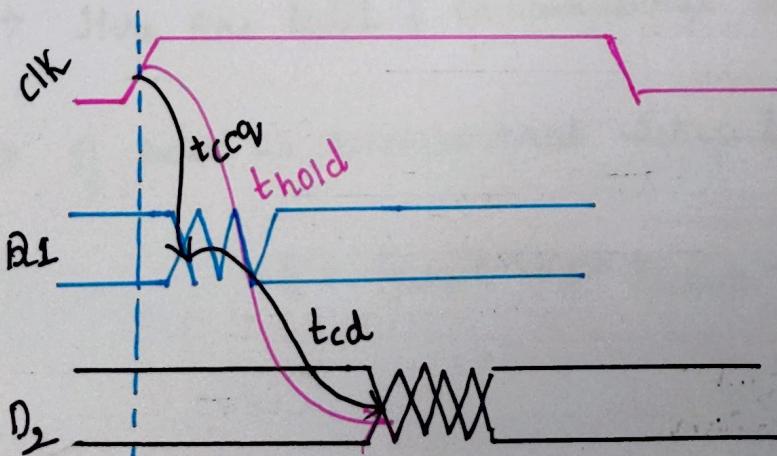
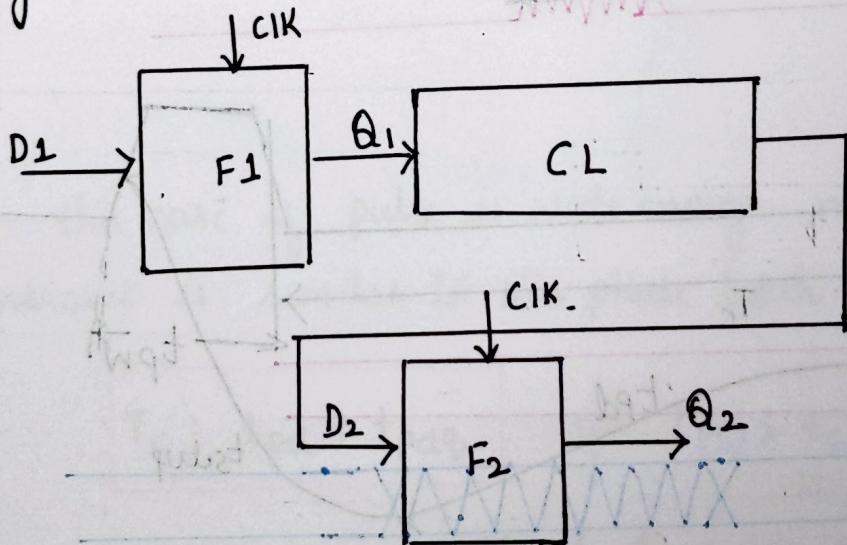
scrubbing overhead.

$$t_{pd} < T_c - t_{pcq} - t_{setup} + t_{pw} \rightarrow t_{pw} < t_{setup}$$



Min. delay Constraint :-

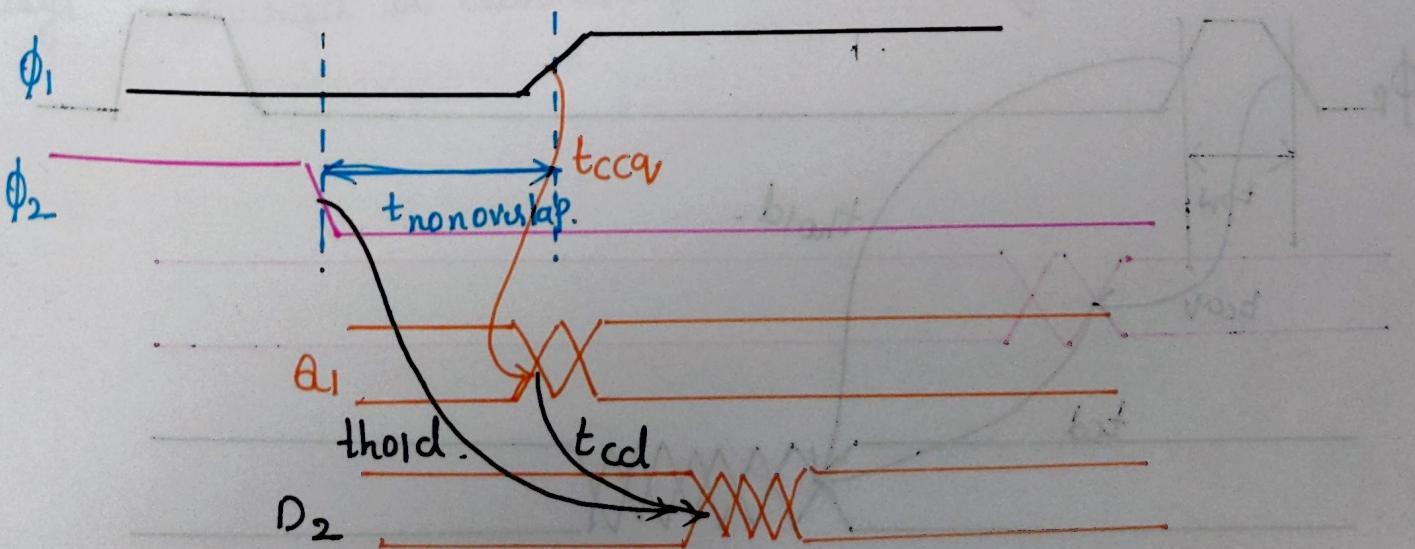
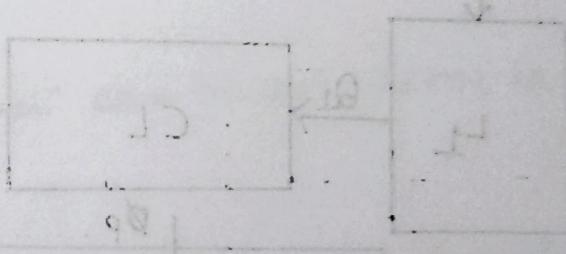
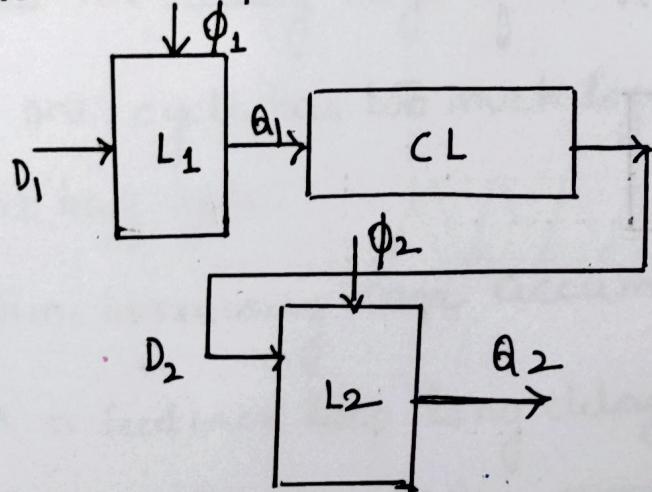
- Ideally sequencing elements can be placed back to back with out intervening C.L. and still function correctly
- If hold time is large and contamination delay is small data can incorrectly propagate through two successive elements
- This is called "race condition" or "hold time violation"
- If contamination delay through the FF exceeds hold time safely use back-back FF without any violations.



$$t_{hold} < t_{ccq} + t_{cd}$$

$$t_{cd} \geq t_{hold} - t_{ccq}$$

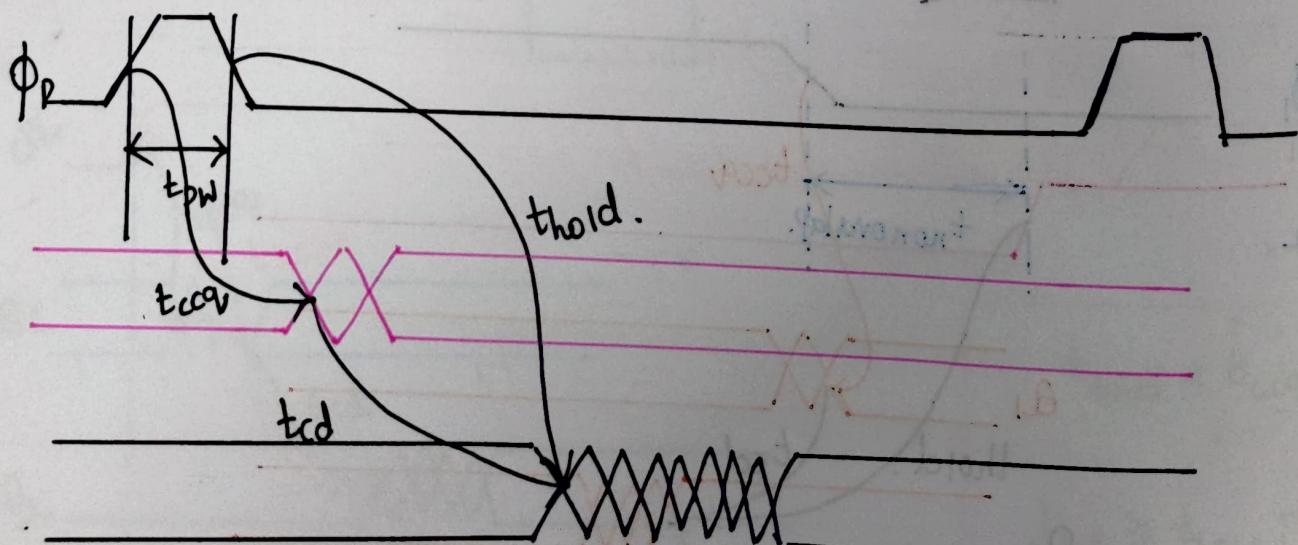
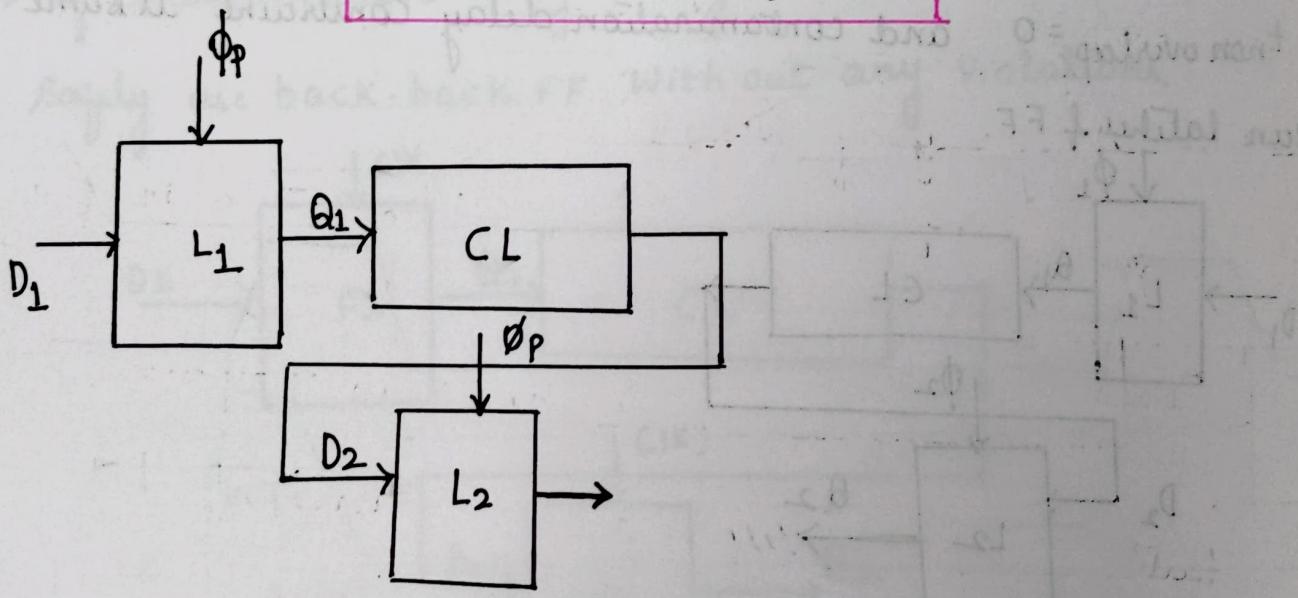
- II
- Figure shows min-delay-limiting constraint on a path from one transverse-latch to next
 - minimum contamination delay through each phase of logic is
- $$t_{cd1}, t_{cd2} \geq t_{hold} - t_{ccq} - t_{nonoverlap}$$
- $$t_{hold} \leq t_{cd} + t_{ccq} + t_{nonoverlap}$$
- making $t_{nonoverlap}$ sufficiently large hold-time value can be avoided.
 - If $t_{nonoverlap} = 0$ and contamination delay constraint is same between latches & FF.



III

- Fig. shows min-delay timing constraints on a path from one pulsed latch to next
- Data departs on rising edge of pulse but must hold until falling edge of pulse
- ∴ Pulse width effectively increases hold time of Pulsed Latch

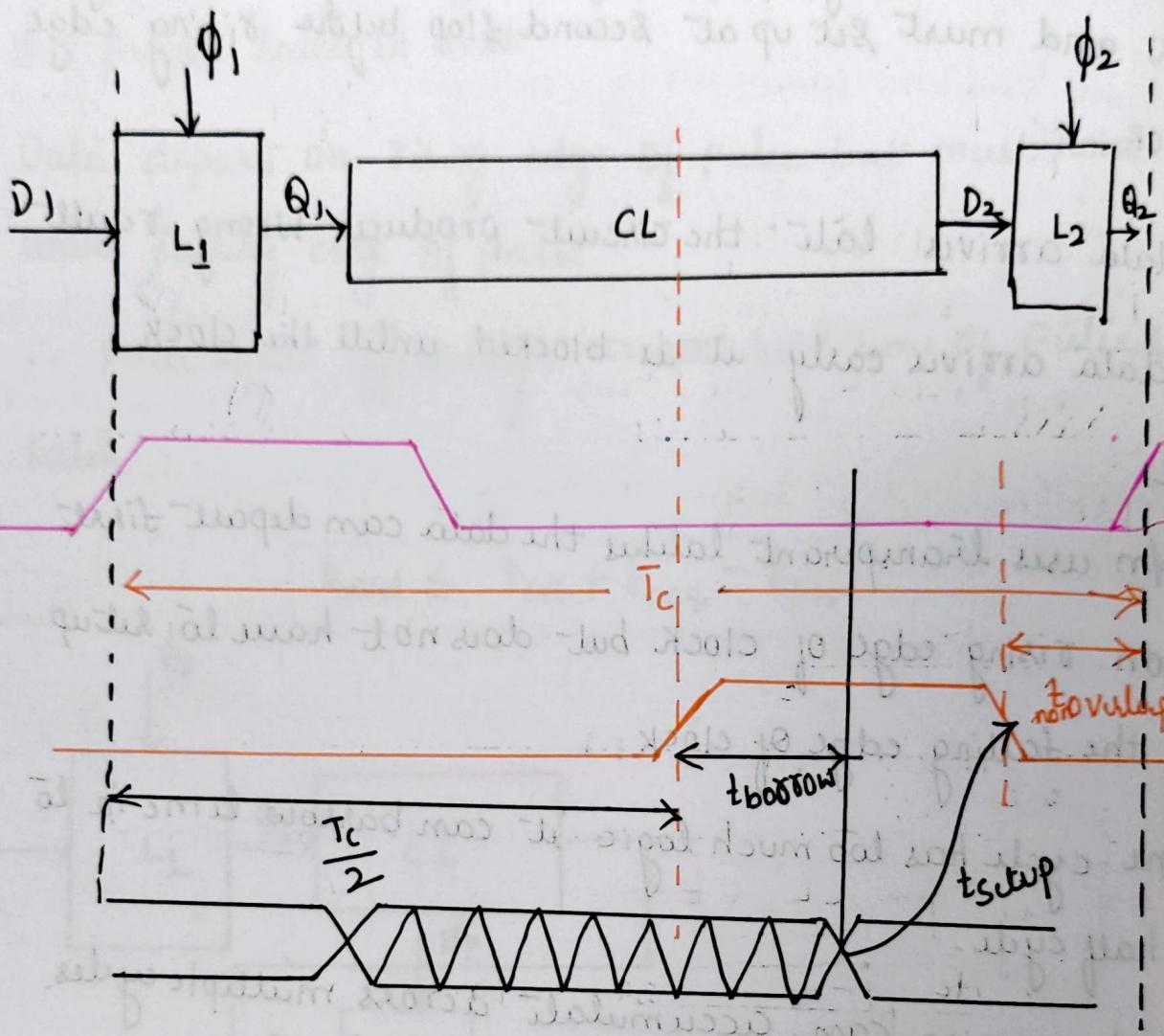
$$t_{hold} \leq t_{cd} + t_{ccq} - t_{pw}$$



Time borrowing:-

- In a S/m using FF data departs on first rising edge of CLK and must set up at second stop before rising edge of clock
- If data arrives late the circuit produces wrong result
- If data arrives early it is blocked until the clock edge.
- If S/m uses transparent latches the data can depart first latch on rising edge of clock but does not have to setup until the falling edge of clock.
- If one cycle has too much logic it can borrow time in to next half cycle.
- Time borrowing can accumulate across multiple cycles.
- In a feedback loop long delays must be balanced by short delays ∴ Overall it can complete in same cycle

$$t_{\text{borrow}} = \frac{T_c}{2} - (t_{\text{setup}} + t_{\text{nonoverlap}})$$

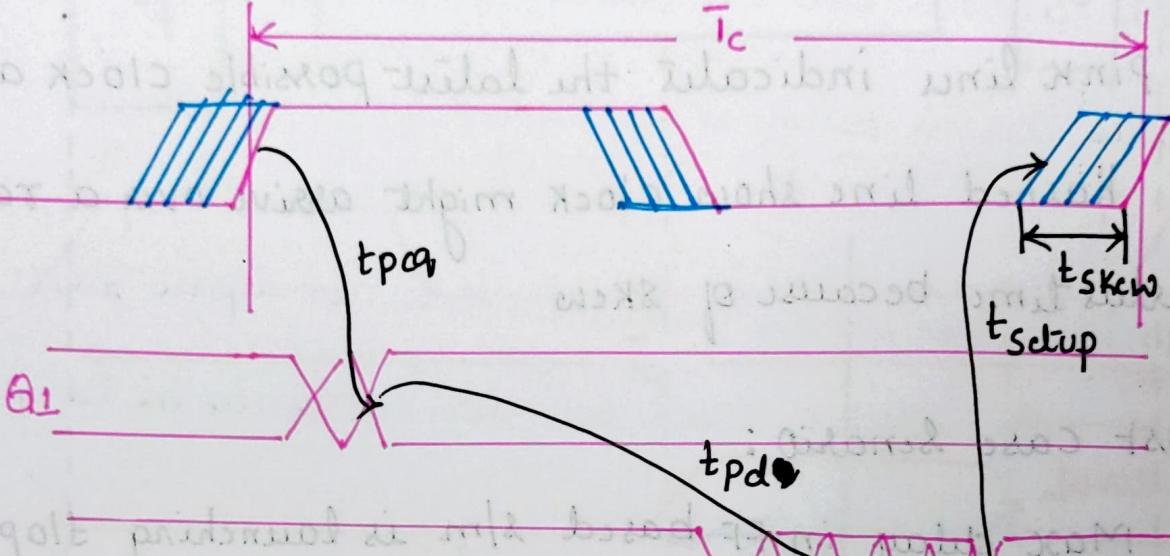
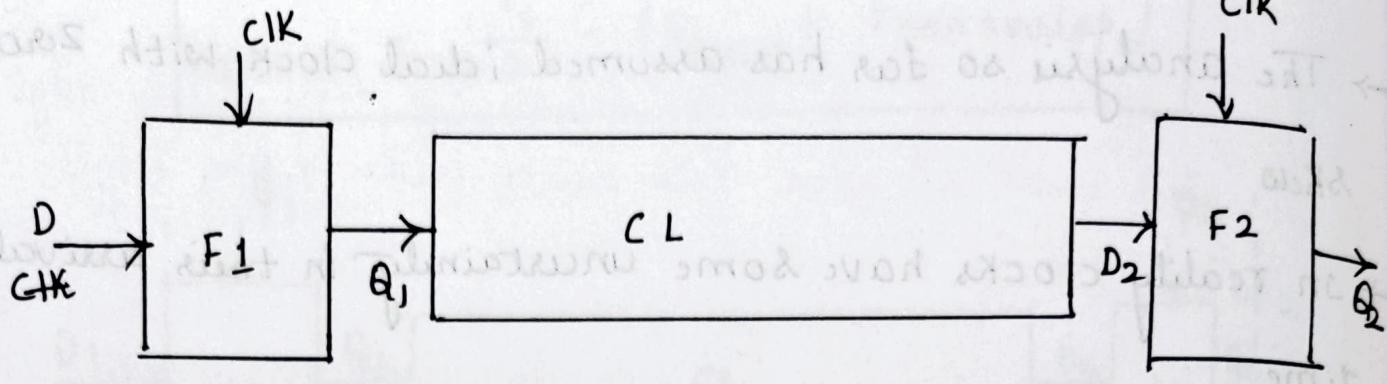


$$\frac{T_c}{2} = t_{\text{borrow}} + t_{\text{setup}} + t_{\text{nonoverlap}}.$$

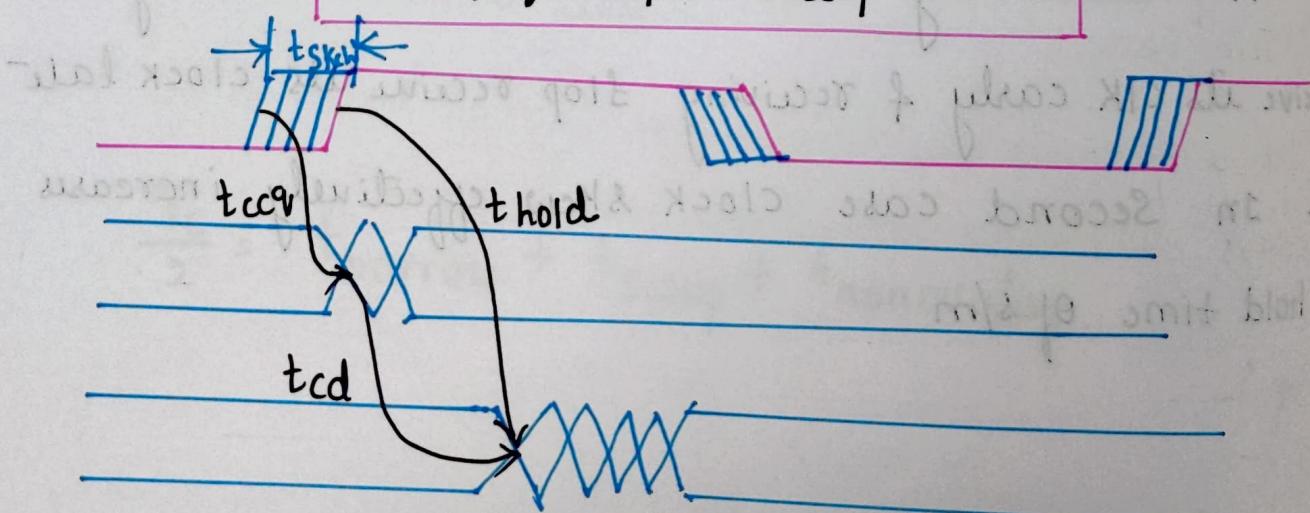
CLOCK-SKew

- The analysis so far has assumed ideal clock with zero skew
- In reality clocks have some uncertainty in their arrival time
- The pink line indicates the latest possible clock arrival
- The dashed line shows clock might arrive over a range of earlier time because of skew
- Worst Case Scenario:
 - i. Max delay in FF based s/m is launching flop receiving its CLK late & receiving flop receive its clock early.
 - ii. Min delay in FF based s/m is launching flop receive its CLK early & receiving flop receive its clock late
- In Second case clock skew effectively increases hold time of s/m

$$\text{Setup} - \text{Hold} + \text{Setup} \geq \text{Hold}$$



$$T_c = t_{pq} + t_{pd} + t_{setup} + t_{skew}$$



$$t_{hold} \leq t_{cq} + t_{cd} - t_{skew}$$

→ If a S/m uses transparent latches clock skew doesn't degrade performance

→ ∴ transparent-latch based S/m are skew tolerant S/m.

$$t_{pd} \leq T_c - \underbrace{(2t_{pdq})}_{\text{sequencing overhead}}$$

$$t_{cd} > t_{hold} - t_{ccq} - t_{nonoverlap} + t_{skew}$$

$$t_{borrow} \leq \frac{T_c}{2} - [t_{setup} + t_{nonover} + t_{skew}]$$

→ Pulsed latches can tolerate amount of skew proportional to PW

→ If PW is wide enough skew will not increase sequencing overhead.

→ If pulse is narrow skew can degrade performance

$$T_c = t_{pd} + t_{pdq}$$

$$T_c = t_{pd} + t_{pcq} + t_{setup} - t_{pw} + t_{skew}$$

$$t_{hold} \leq t_{cd} + t_{ccq} - t_{pw} - t_{skew}$$

$$t_{borrow} \leq t_{pw} - (t_{setup} + t_{skew})$$

	T_c	t_{hold}	t_{borrow}	$\{ \begin{matrix} T_c \\ t_{hold} \end{matrix} \} \\ t_{skew}$
I Flip Flops	$T_c = t_{pcq} + t_{pd} + t_{setup}$	$t_{hold} \leq t_{ccq} + t_{cd}$	-	$T_c = t_{pcq} + t_{pd} + t_{setup}$ $+ t_{skew}$
II Two-phase Latch	$T_c = t_{pdg} + t_{pd} + t_{pdg} + t_{pd}$	$t_{hold} \leq t_{cd} + t_{ccq} + t_{nonoverlap}$	$t_{borrow} < \frac{T_c}{2}$ $- (t_{setup} + t_{nonoverlap})$ $t_{borrow} < \frac{T_c}{2} -$ $(t_{setup} + t_{nonoverlap} + t_{skew})$	$T_c = \text{no change}$ $t_{hold} < t_{ccq} + t_{cd} + t_{nonoverlap} + t_{skew}$
III $t_{pw} > t_{setup}$ i) $T_c = t_{pat} + t_{pdg}$	$T_c = t_{pat} + t_{pdg}$			$T_c = t_{pd} + t_{pdg}$
ii) $t_{pw} < t_{setup}$	$T_c = t_{pcq} + t_{pd} + t_{setup} - t_{pw}$	$t_{hold} < t_{cd} + t_{ccq} - t_{pw}$	$t_{borrow} < t_{pw} - t_{setup}$ $t_{borrow} < t_{pw} - (t_{setup} + t_{skew})$	$T_c = t_{pcq} + t_{pd} + t_{setup} - t_{pw} + t_{skew}$ $t_{hold} < t_{cd} + t_{ccq} - t_{pw} - t_{skew}$