Module 4:- Combinational and Sequential Circuit Design

Module 4:- Combinational and Sequential Circuit Realization

4.1 Analysis and design of 2-I/P NAND, 2-I/P NOR and complex Boolean function realization using equivalent CMOS inverter for simultaneous switching

4.2 Basic gates and MUX realization using pass transistor and transmission gate logic.

4.3 SR Latch, JK FF, D FF, 1 Bit Shift Register realization using CMOS logic.

Realization using CMOS Inverter

For Series connected MOSFETs

$$\frac{1}{Ceq} = \frac{1}{C1} + \frac{1}{C2} + \cdots$$

$$\frac{1}{\left(\frac{W}{L}\right)eq} = \frac{1}{\left(\frac{W}{L}\right)1} + \frac{1}{\left(\frac{W}{L}\right)2} + \cdots$$

$$\left(\frac{W}{L}\right)eq = \frac{1}{\frac{1}{\left(\frac{W}{L}\right)1} + \frac{1}{\left(\frac{W}{L}\right)2} + \ \ldots}$$

$$\left(\frac{W}{L}\right)eq = \frac{1}{\sum_1^K \frac{1}{\left(\frac{W}{L}\right)}}$$

For unequal (W/L) ratio

$$\left(\frac{W}{L}\right)eq = \frac{W}{nL}$$

For equal (W/L) ratio

For Parallel connected MOSFETs

$$Ceq = C1 + C2$$

$$\left(\frac{W}{L}\right)eq = \left(\frac{W}{L}\right)1 + \left(\frac{W}{L}\right)2 + \ \ldots$$

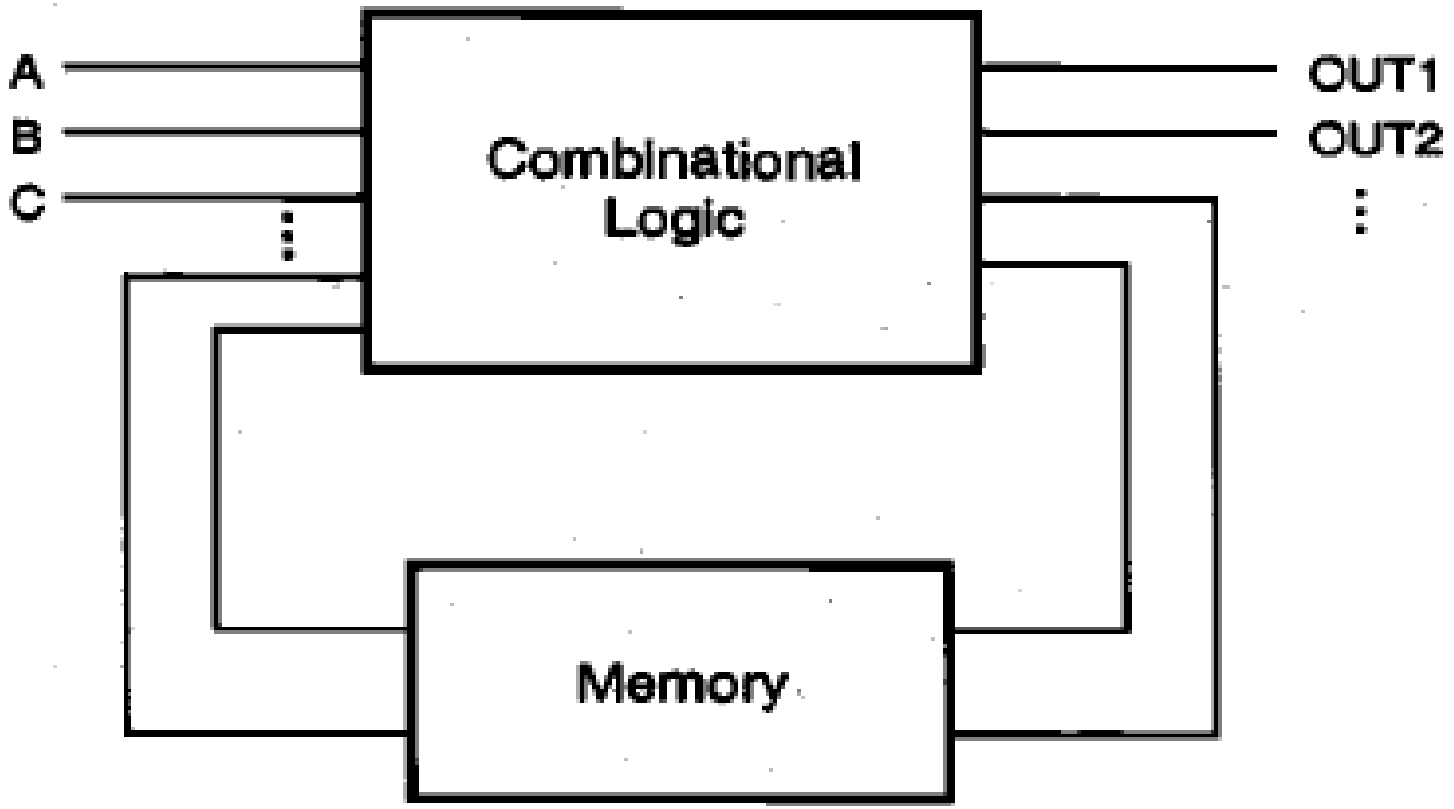
$$\left(\frac{W}{L}\right)eq = \sum_1^K \left(\frac{W}{L}\right)$$

$$\left(\frac{W}{L}\right)eq = \frac{nW}{L}$$

2-I/P NAND and NOR equivalent CMOS Inverter

Note :- Refer class notebook for numerical

Sequential Logic Circuit



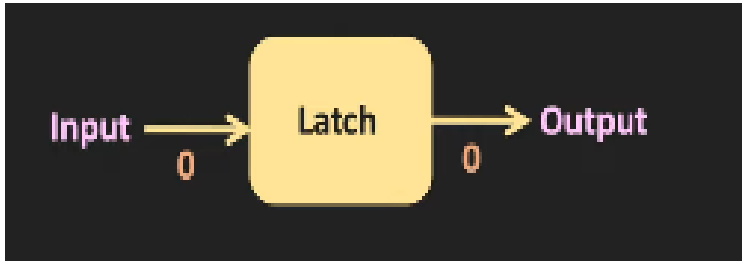
Memory

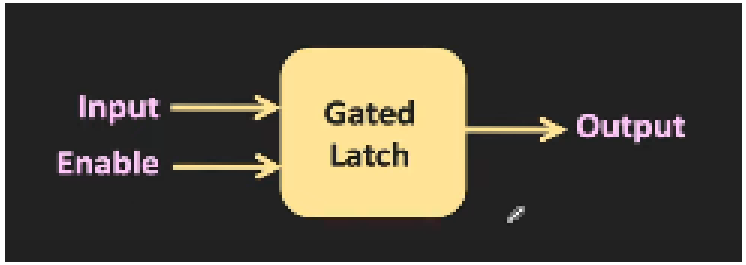
Latch

Flip-flop

Transparent latch

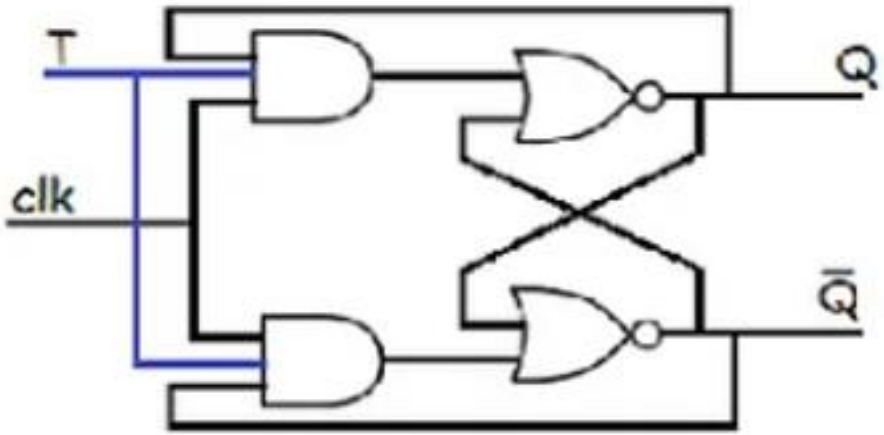
Gated latch

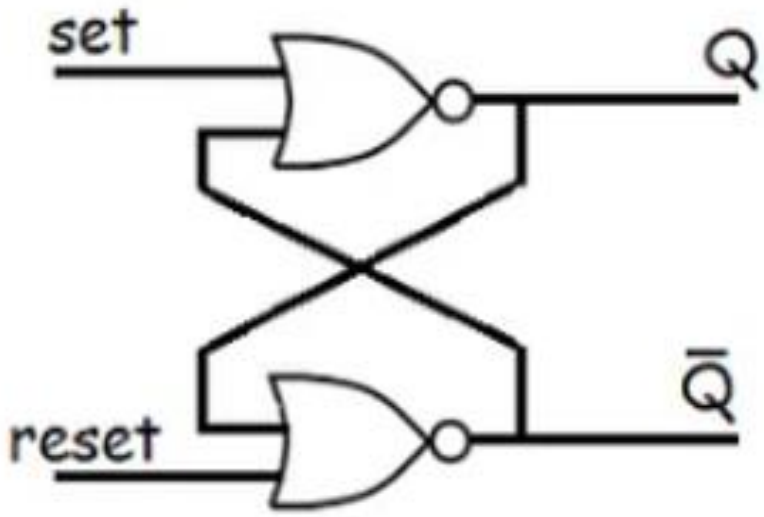




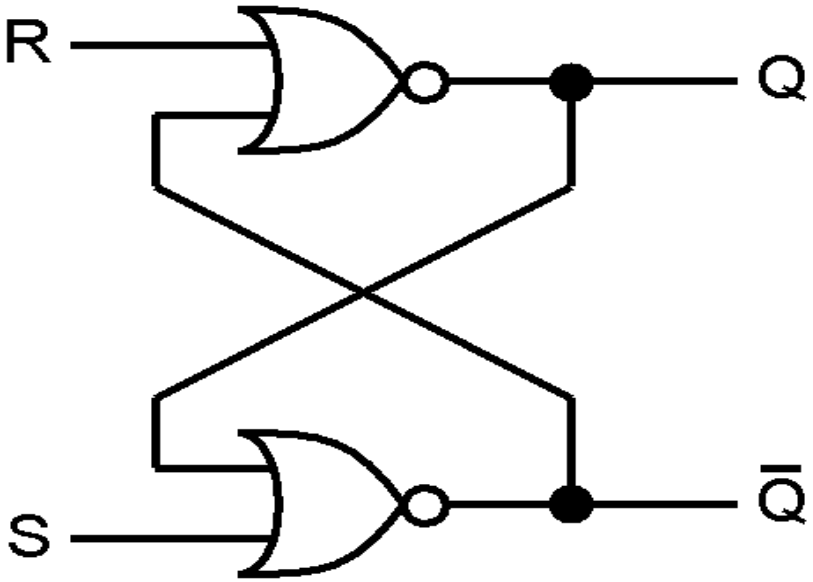
Difference between Flip flop and Latch

Flip flop	Latch
It requires Clock to function	It does not requires Clock
It is Edge triggered device	It is Level triggered device
It checks the i/p continuously but changes the o/p only when triggered by clock	It checks the i/p continuously and changes the o/p immediately w.r.t. i/p
It is made up of latches and logic gates	It is made up of logic gates only





SR Latch using NOR gates

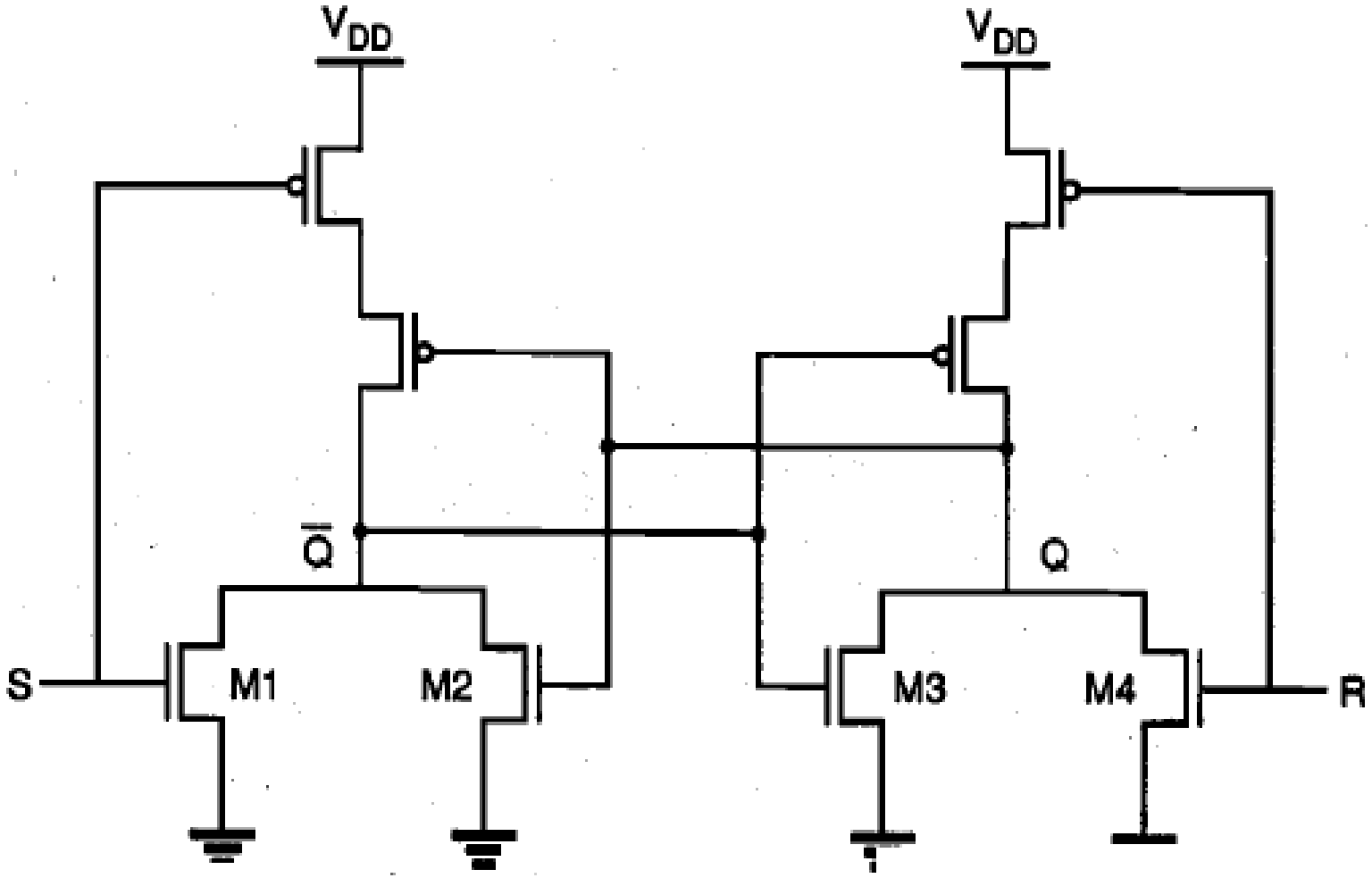


S	R	Q	Q’
0	0	No Change (hold)	
0	1	0	1
1	0	1	0
1	1	Invalid	

$$Q = \overline{R + \bar{Q}}$$

$$\bar{Q} = \overline{S + Q}$$

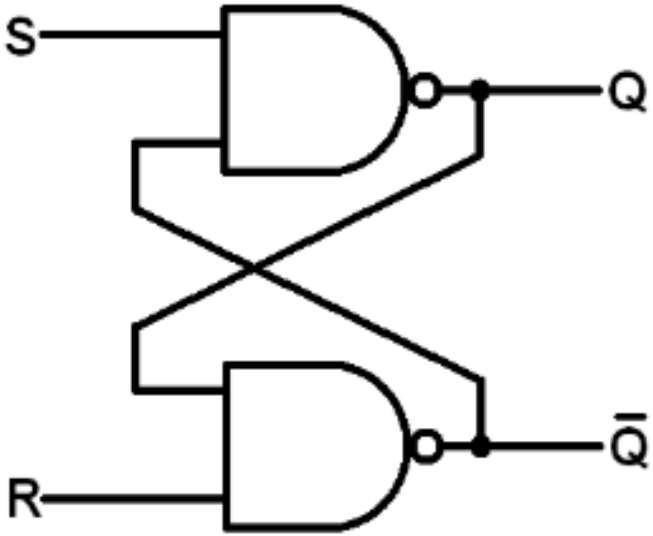
CMOS NOR based SR Latch



$$\overline{Q} = \overline{S + Q}$$

$$Q = \overline{R + \overline{Q}}$$

SR Latch using NAND gates

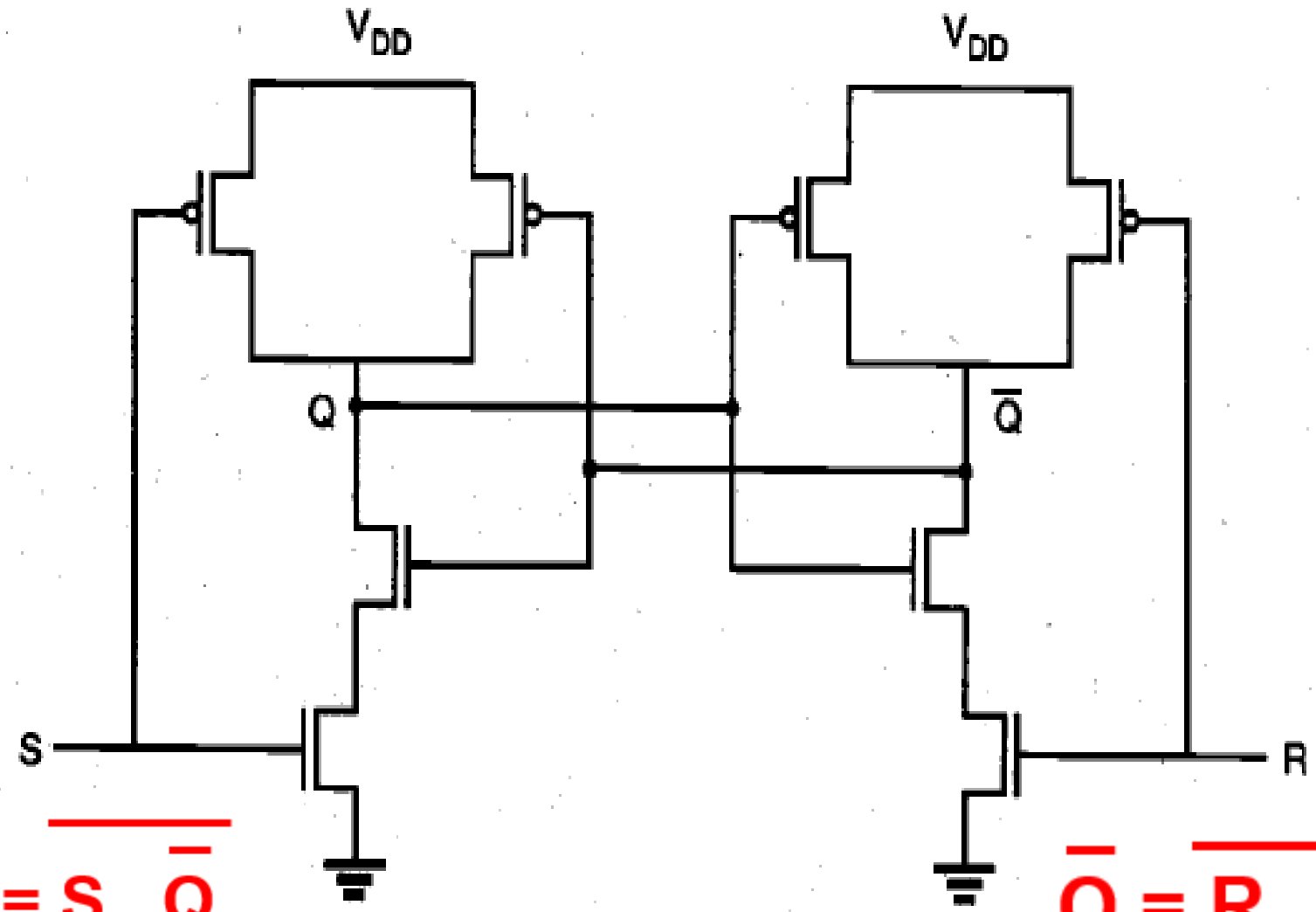


S	R	Q	Q’
0	0	Invalid	
0	1	1	0
1	0	0	1
1	1	No Change (hold)	

$$Q = \overline{S \cdot \bar{Q}}$$

$$\bar{Q} = \overline{R \cdot Q}$$

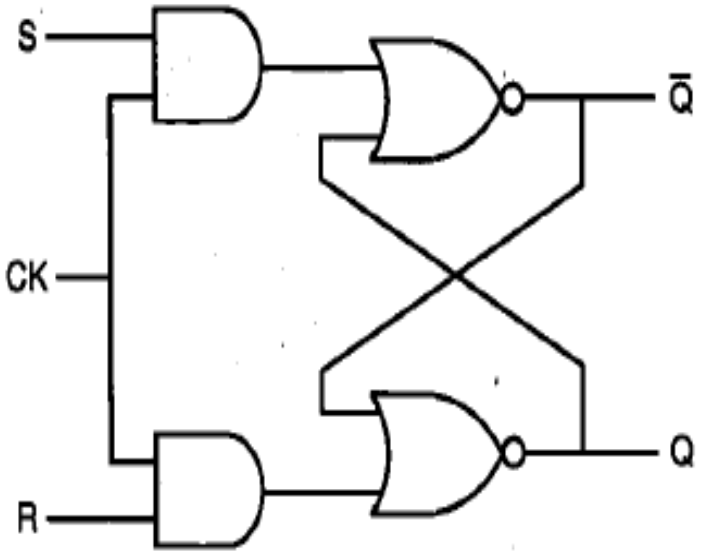
CMOS NAND based SR Latch



$$Q = \overline{S \cdot \bar{Q}}$$

$$\bar{Q} = \overline{R \cdot Q}$$

Clocked SR Latch using NOR gates

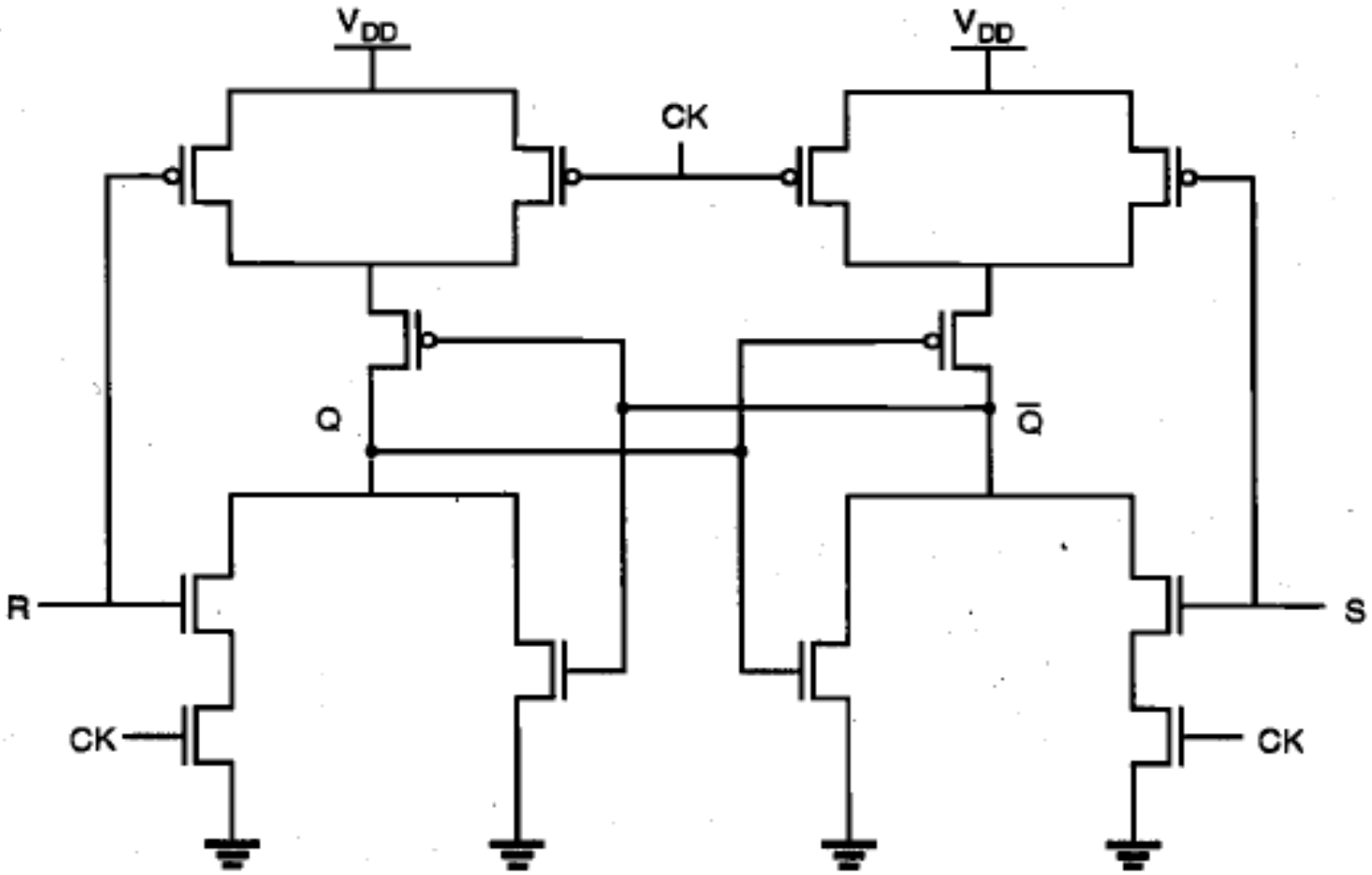


Clk	S	R	Q	Q'
0	X	X	No Change (hold)	
1	0	0	No Change (hold)	
1	0	1	0	1
1	1	0	1	0
1	1	1	Invalid	

$$Q = \overline{(R.Clk) + \bar{Q}}$$

$$\bar{Q} = \overline{(S.Clk) + Q}$$

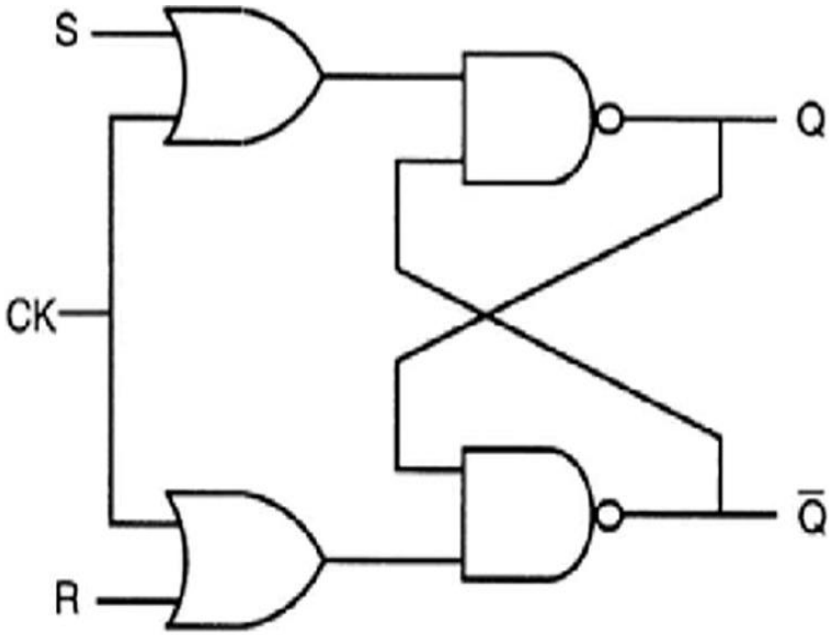
CMOS Clocked SR Latch using NOR gates



$$Q = \overline{(R.Clk) + \overline{Q}}$$

$$\overline{Q} = \overline{(S.Clk) + Q}$$

Clocked SR Latch using NAND gates

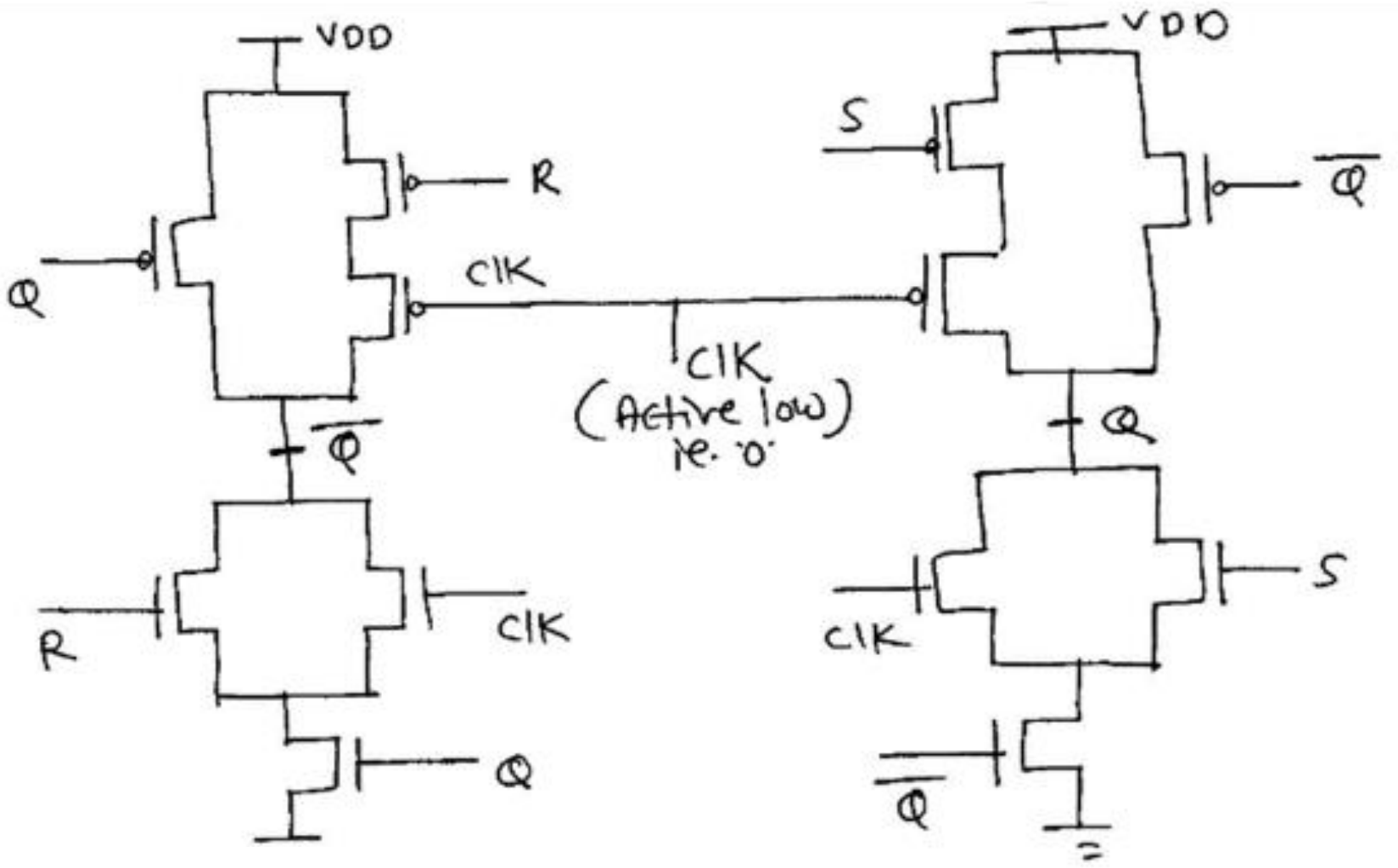


Clk	S	R	Q	Q'
1	X	X	No Change (hold)	
0	0	0	Invalid	
0	0	1	1	0
0	1	0	0	1
0	1	1	No Change (hold)	

$$Q = \overline{(S+Clk).\bar{Q}}$$

$$\bar{Q} = \overline{(R+Clk).Q}$$

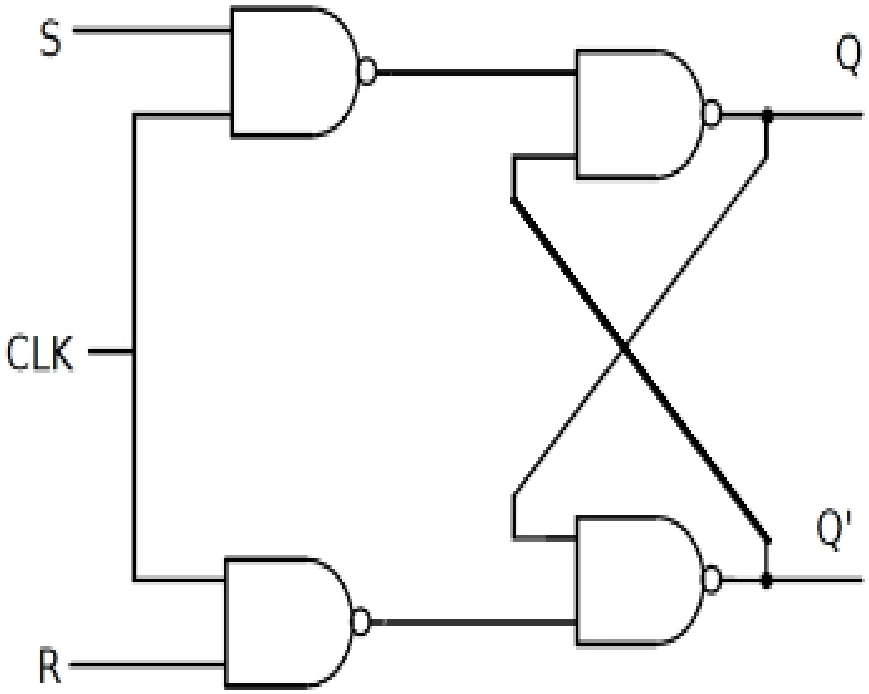
CMOS Clocked SR Latch using NAND gates



$$Q = \overline{(S+Clk).\bar{Q}}$$

$$\bar{Q} = \overline{(R+Clk).Q}$$

SR flip flop using NAND gates

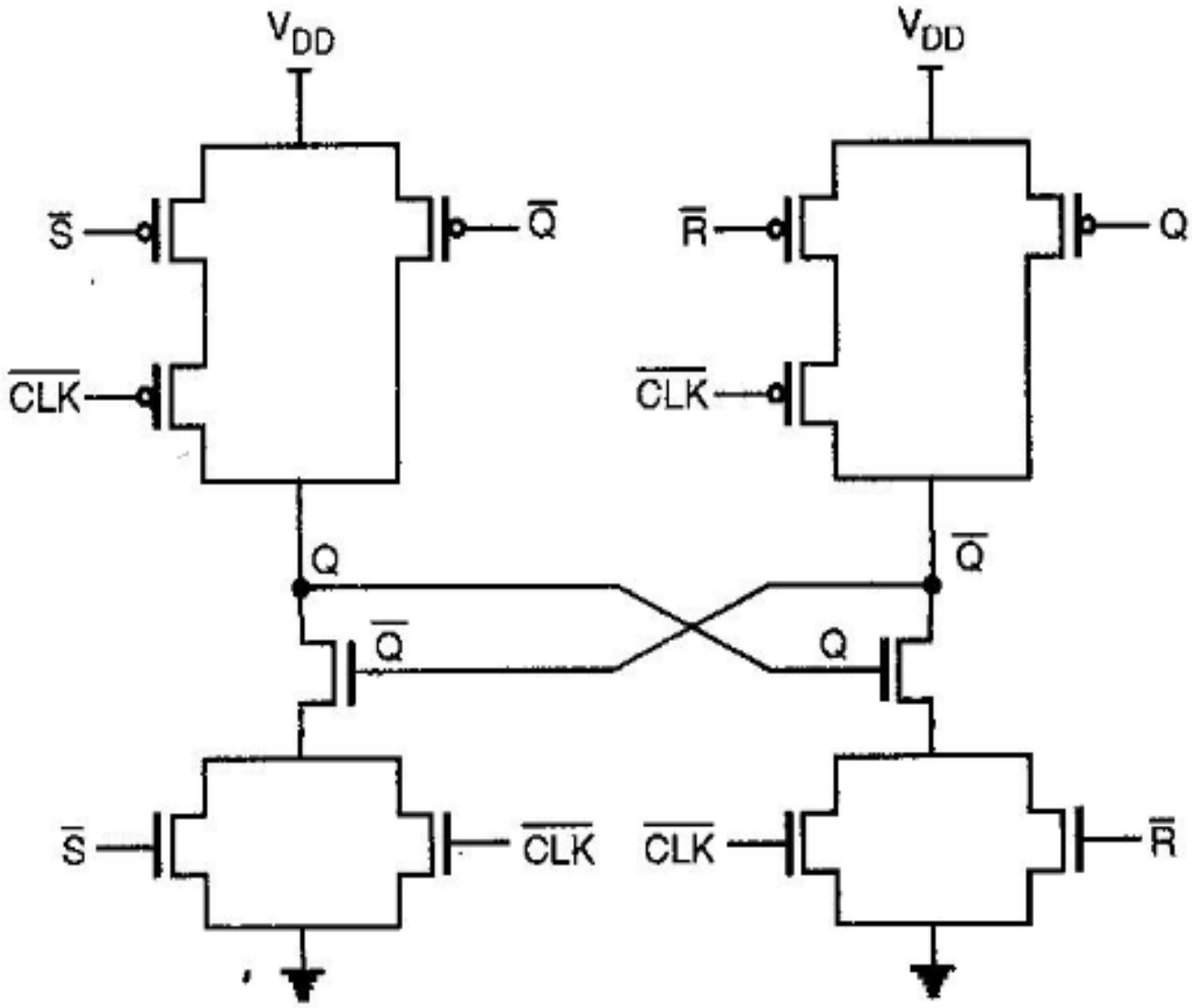


Clk	S	R	Q	Q’
1	0	0	No Change (hold)	
1	0	1	0	1
1	1	0	1	0
1	1	1	Invalid	

From Gate ckt dia, we have

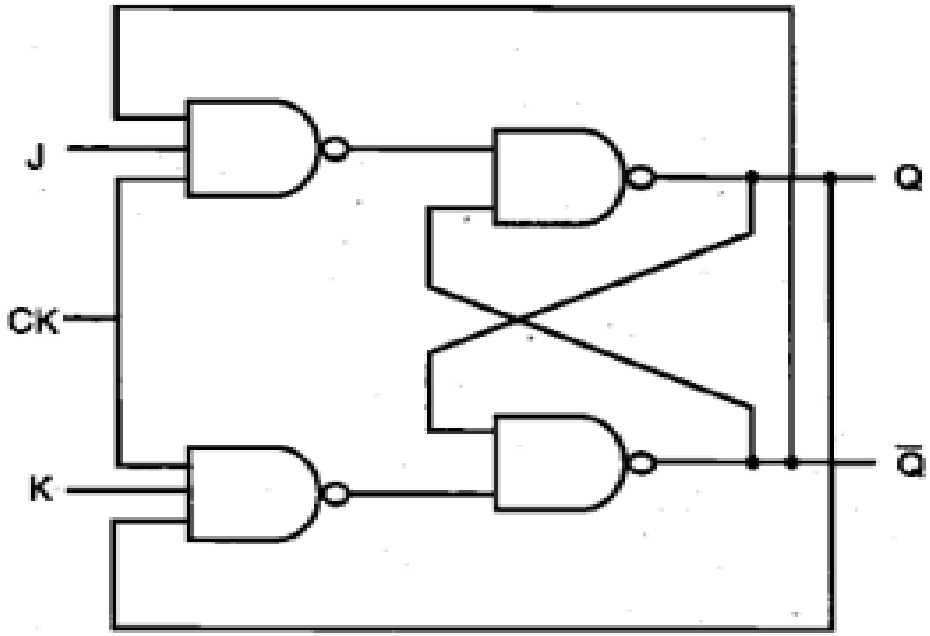
$$Q = \overline{(\overline{S \cdot CLK}) \cdot \overline{Q}} = \overline{(\overline{S} + \overline{CLK}) \cdot \overline{Q}}$$

CMOS Implementation of SR flip flop



JK flip flop

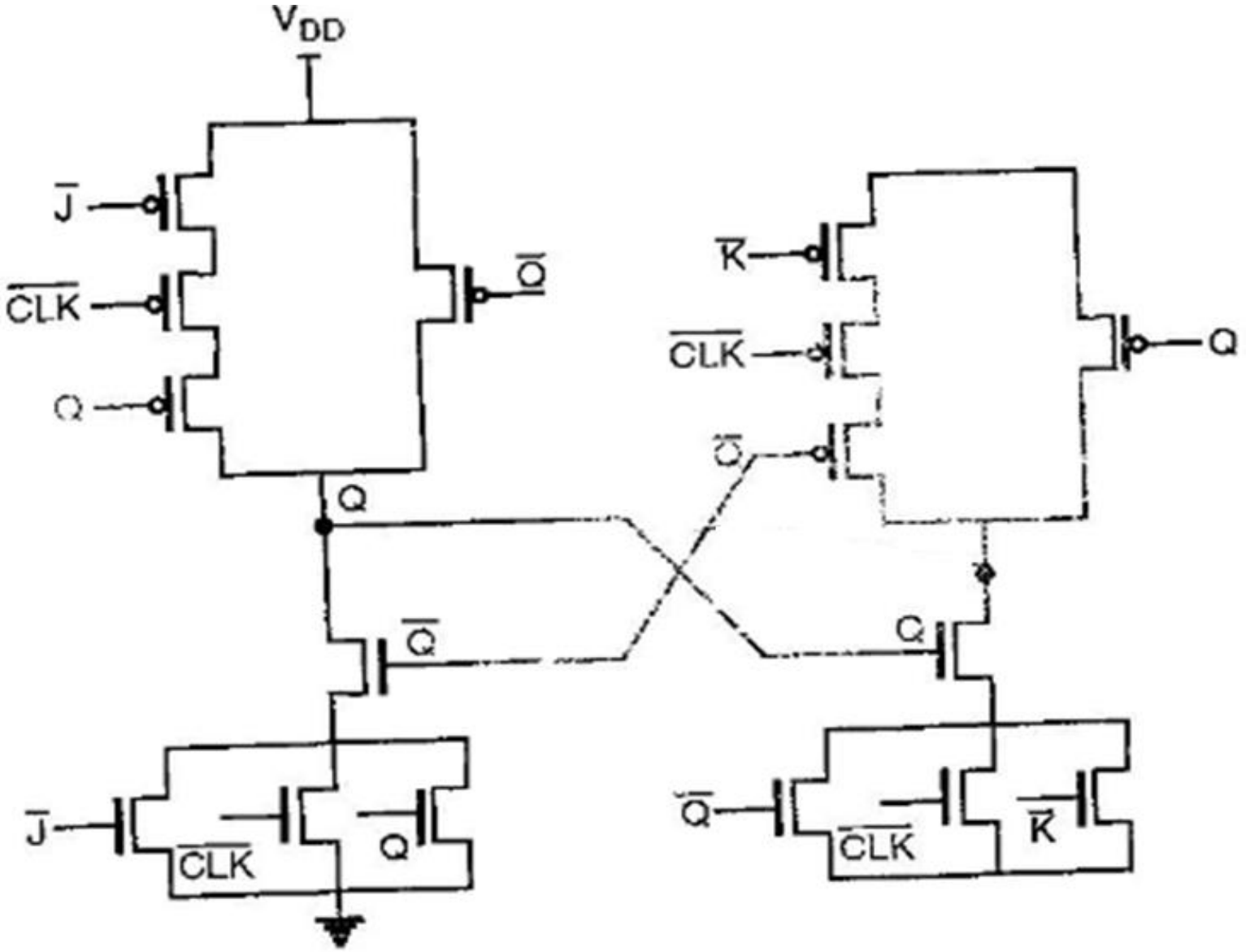
The problem in SR flip flop is when both the inputs are high at the same time , Not allowed conditions comes, hence to overcome this , two feedback lines are added from the o/p to i/p as shown below, JK flip flop is form.



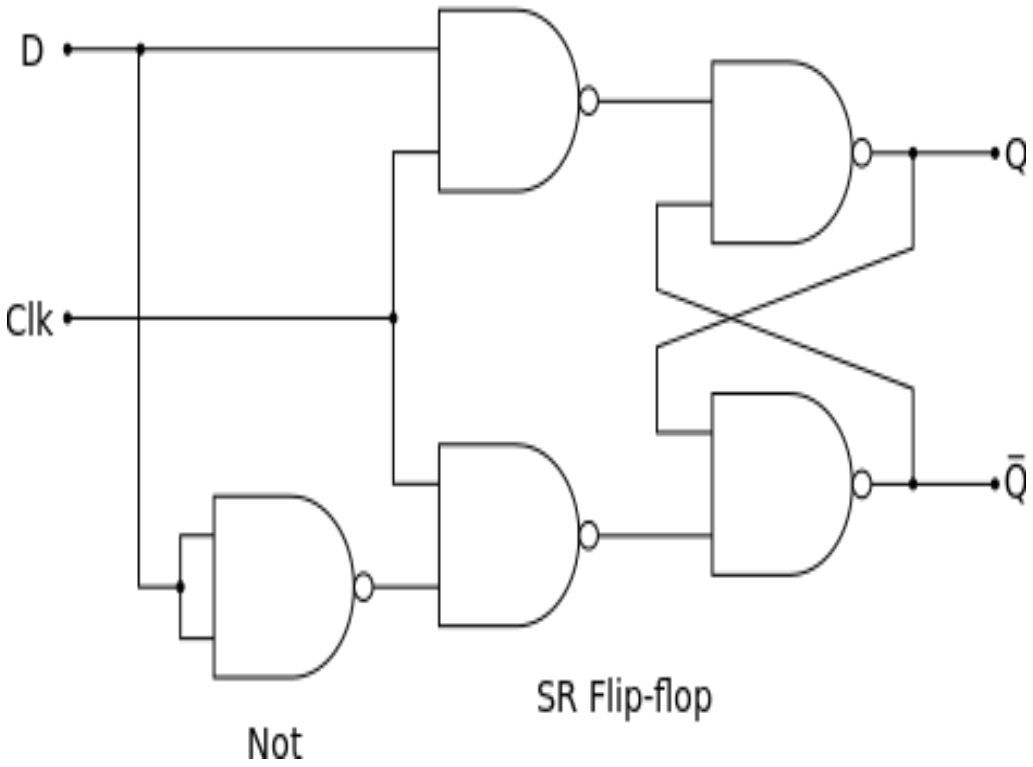
Clk	J	K	Q	Q'
1	0	0	No Change (hold)	
1	0	1	0	1
1	1	0	1	0
1	1	1	Toggle	

$$Q = \overline{(\overline{J \cdot Clk \cdot \overline{Q}}) \cdot \overline{Q}} = \overline{(\overline{J} + \overline{Clk} + Q) \cdot \overline{Q}}$$

CMOS Implementation of JK flip flop



D flip flop



Clk	D	Q	Q’
0	X	Hold	
1	0	0	1
1	1	1	0

$$Q = \overline{\overline{(D.Clk)} \cdot \overline{Q}}$$

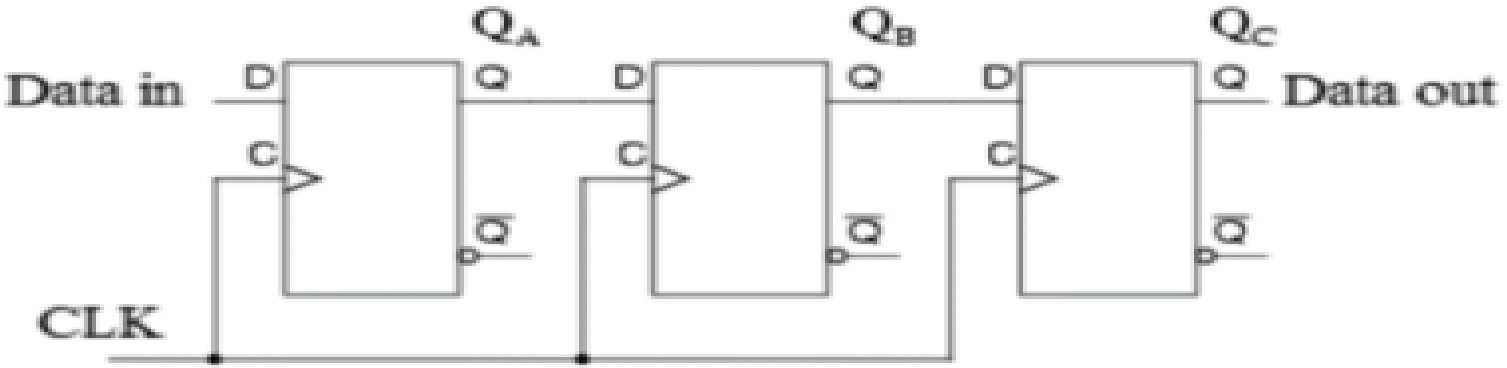
$$= \overline{(\overline{D}+\overline{Clk}).\overline{Q}}$$

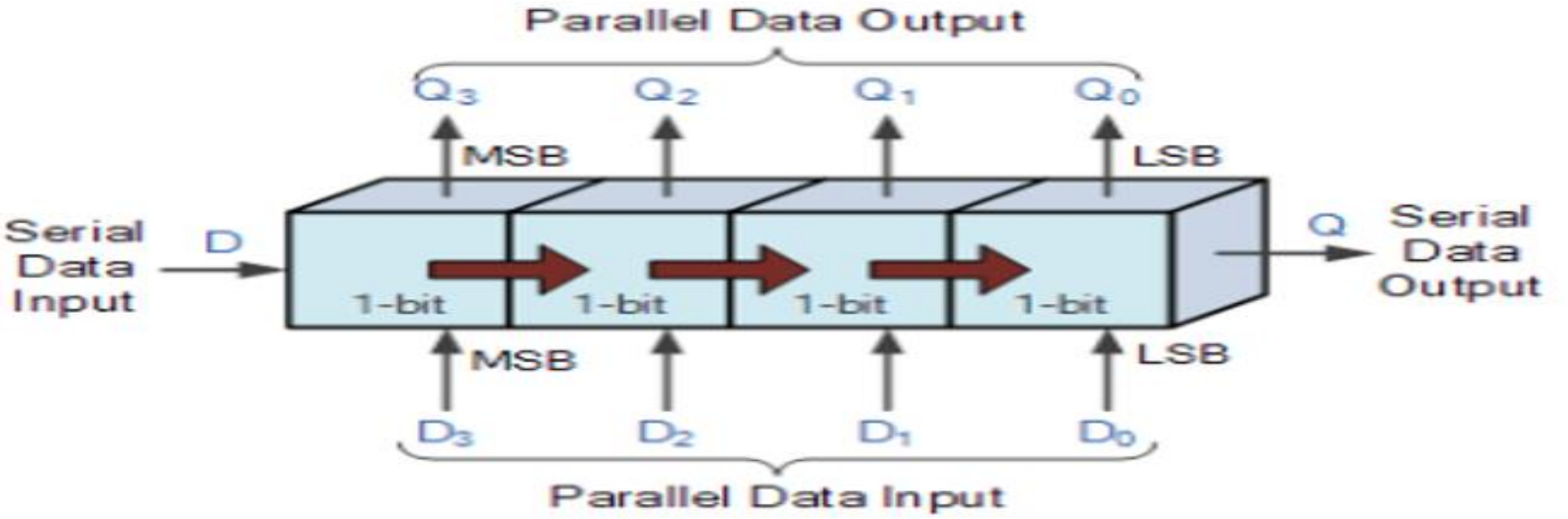
$$\overline{Q} = \overline{\overline{\overline{(D.Clk)} \cdot Q}}$$

$$= \overline{(D+\overline{Clk}).Q}$$

Shift Register

- Registers are chain of flip flops connected so that data can be shifted in one direction at each clock pulse.





Shift Register

Static shift register

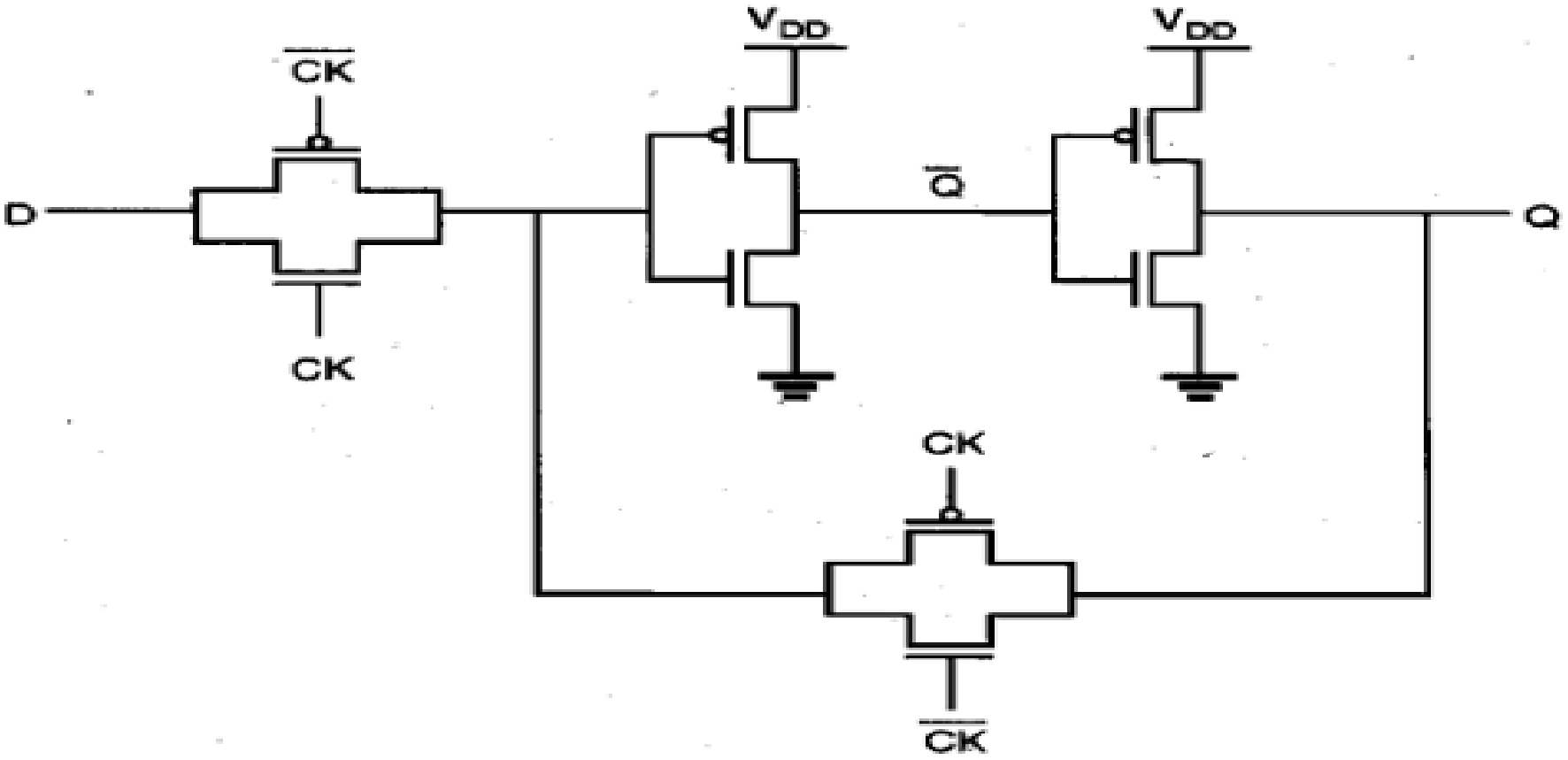
- Uses flip flops
- Stores information for indefinite period of time

Dynamic shift register

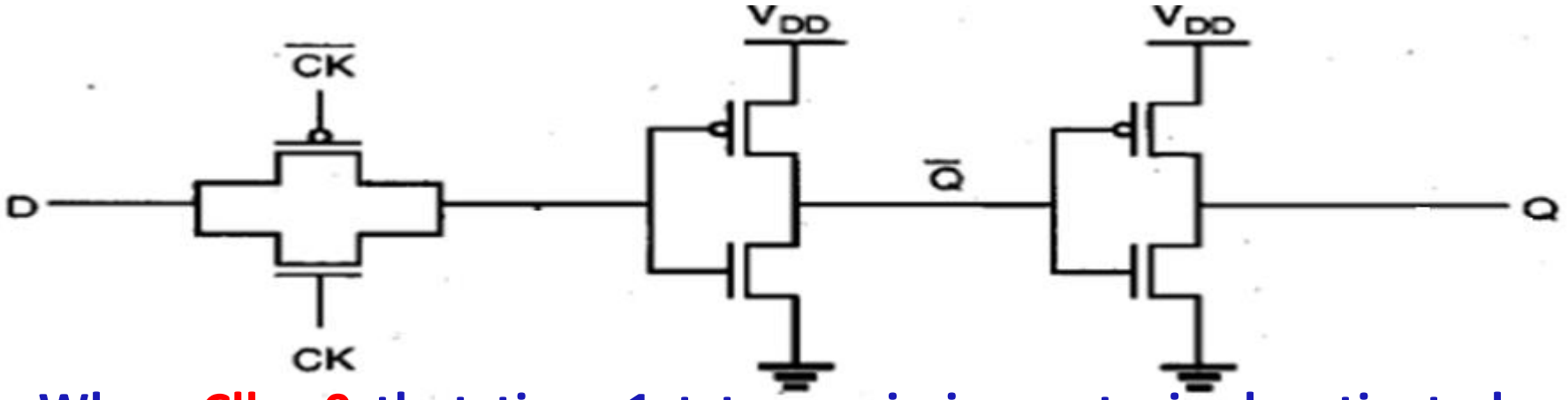
- Uses MOS + Capacitors
- Required refresh cycles

1 bit Static shift register using Transmission gates

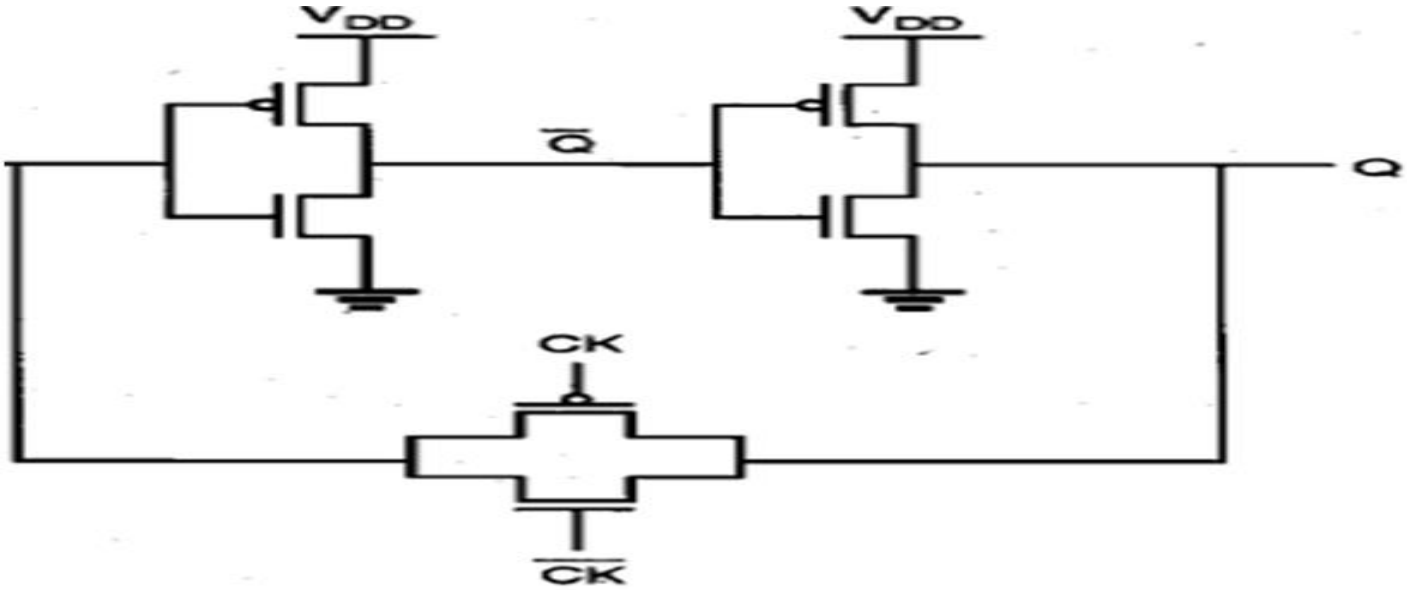
- **Static shift registers are composed of flip flops and are capable of storing the information within them for indefinite period of time.**



- When Clk= 1 that time 1st transmission gate is activated and 2nd transmission gate is deactivated and data (D) is passed through both the inverters and reaches to the output as shown below.

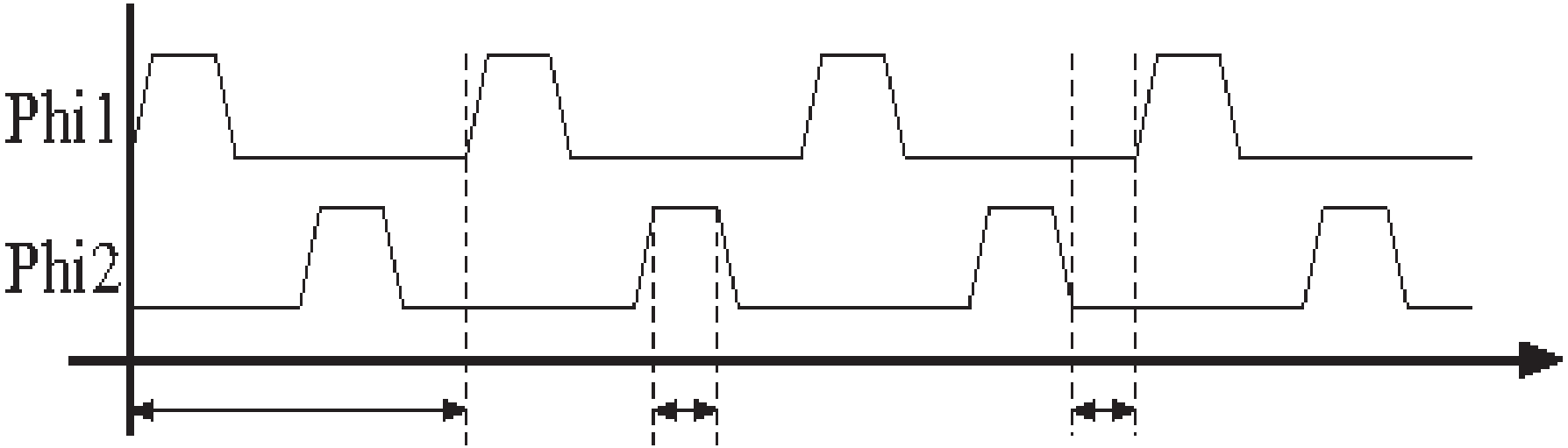


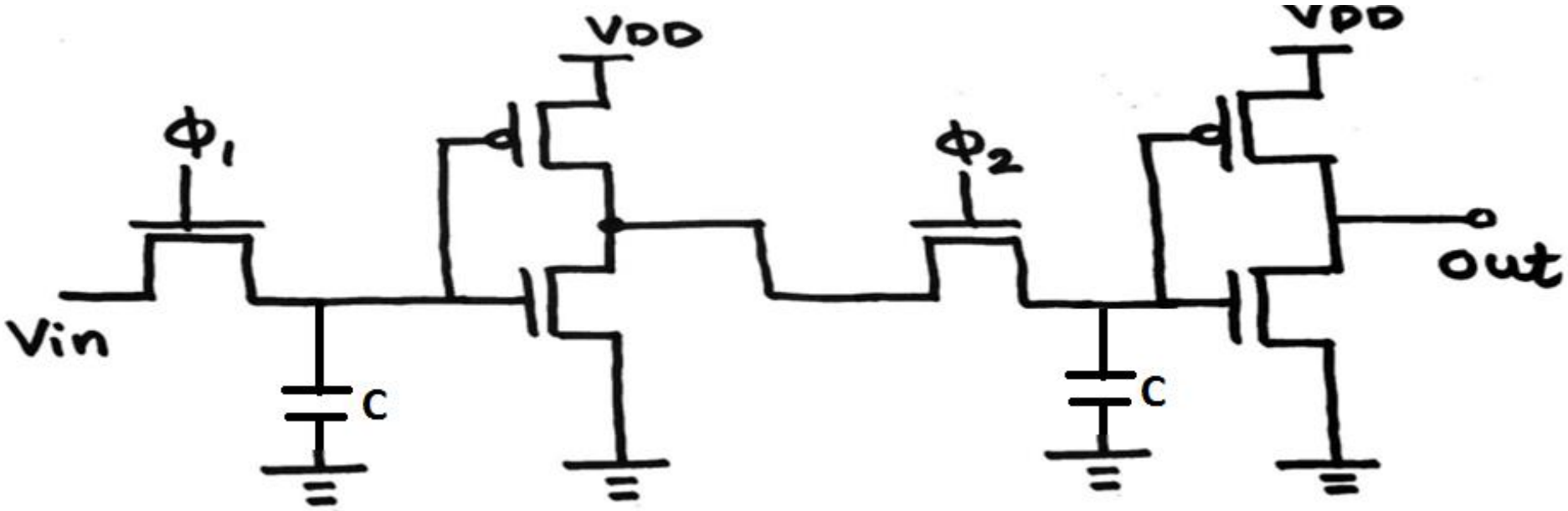
- When Clk= 0 that time 1st transmission gate is deactivated and 2nd transmission gate is activated and data (D) is hold as it is in the latch.



1-bit Dynamic Shift Register

- Dynamic shift registers are formed by a combination of MOS transistors and Capacitors.
- It consist of two Non-overlapping clock signals (ϕ1 & ϕ2)





- When ϕ1= 1 that time ϕ2 = 0 , hence transistor T1 is ON and T2 is OFF.
- Input data Vin passes through a CMOS inverter and produces complement output. (ie. If Vin=0 the output of 1st inverter is 1 and vice-versa)
- Now, ϕ1= 0 that time ϕ2 = 1, hence transistor T1 is OFF and T2 is ON and hence data passes through the second inverter to produce the complement o/p.
- Hence, at the o/p we get original Vin data .

Difference between Static & Dynamic Shift Register

Feature	Static shift register	Dynamic shift register
Storage element	Flip-flops	MOS + Capacitors
Data retention	As long as power is on	Needs refreshing (clock)
Power consumption	Higher	Lower
Need of clock pulse	1 clock pulse per bit	2 clock pulses per bit
Reliability	More reliable	Less reliable
Hardware complexity	More transistors	Fewer transistors
Common use	Counters	Memory chips


Thank you!