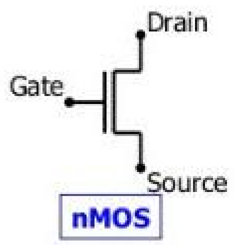
NMOS:

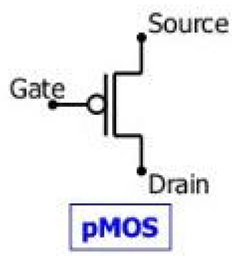
N- channel MOSFET consists of N-type Source and Drain diffused on an P-type substrate. In NMOS, the majority of carriers are electrons. When a high voltage is applied to the gate, the NMOS will conduct. Similarly, when a low voltage is applied to the gate, NMOS will not conduct. NMOS is considered to be faster than PMOS, since the carriers in NMOS, which are electrons, travel twice as fast as the holes.



NMOS Transistor

PMOS:

P- channel MOSFET consists of P-type Source and Drain diffused on an N-type substrate. The majority of carriers are holes. When a high voltage is applied to the gate, the PMOS will not conduct. When a low voltage is applied to the gate, the PMOS will conduct. The PMOS devices are more immune to noise than NMOS devices.



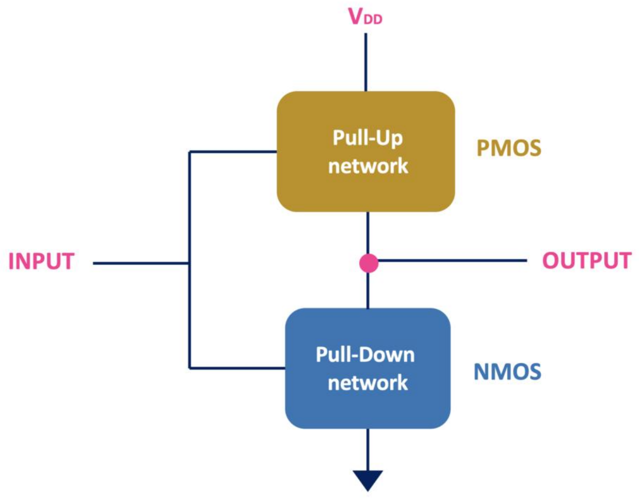
PMOS Transistor

CMOS(Complementary Metal Oxide Semiconductor):

In CMOS technology, both N-type and P-type transistors are used to design logic functions. The same signal which turns ON a transistor of one type is used to turn OFF a transistor of the other type. This characteristic allows the design of logic devices using only simple switches, without the need for a pull-up resistor.

CMOS offers relatively high speed, low power dissipation, high noise margins in both states, and will operate over a wide range of source and input voltages (provided the source voltage is fixed).

In CMOS logic gates a collection of n-type MOSFETs is arranged in a pull-down network between the output and the low voltage power supply rail (Vss or quite often ground). Instead of the load resistor of NMOS logic gates, CMOS logic gates have a collection of p-type MOSFETs in a pull-up network between the output and the higher-voltage rail (often named Vdd).



CMOS using Pull Up & Pull Down

Thus, if both a p-type and n-type transistor have their gates connected to the same input, the p-type MOSFET will be ON when the n-type MOSFET is OFF, and vice-versa. The networks are arranged such that one is ON and the other OFF for any input pattern.

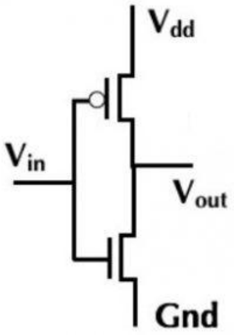
CMOS technology has been used for the following digital IC designs.

- Computer memories, CPUs
- Microprocessor designs
- Flash memory chip designing
- Used to design application-specific integrated circuits (ASICs)

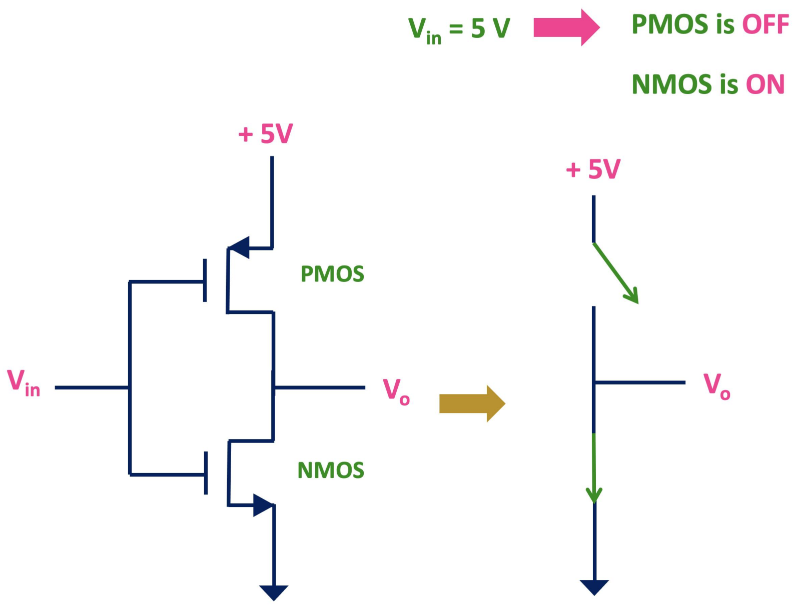
CMOS Inverter:

The inverter circuit as shown in the figure below. It consists of PMOS and NMOS FET. The input A serves as the gate voltage for both transistors.

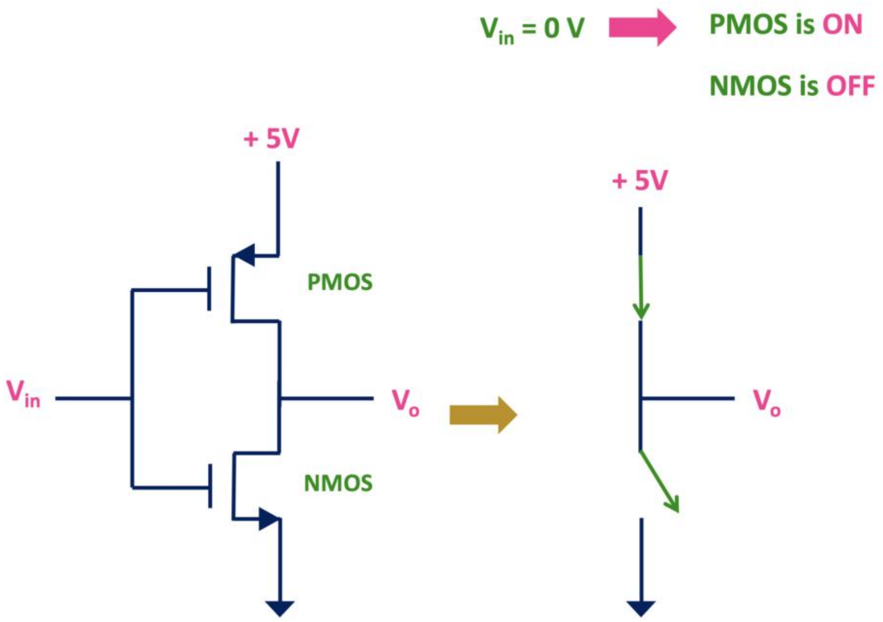
The NMOS transistor has input from Vss (ground) and the PMOS transistor has input from Vdd. The terminal Y is output. When a high voltage (~ Vdd) is given at input terminal (A) of the inverter, the PMOS becomes an open circuit, and NMOS switched OFF so the output will be pulled down to Vss.



CMOS Inverter



When a low-level voltage (<Vdd, ~0v) applied to the inverter, the NMOS switched OFF and PMOS switched ON. So the output becomes Vdd or the circuit is pulled up to Vdd.

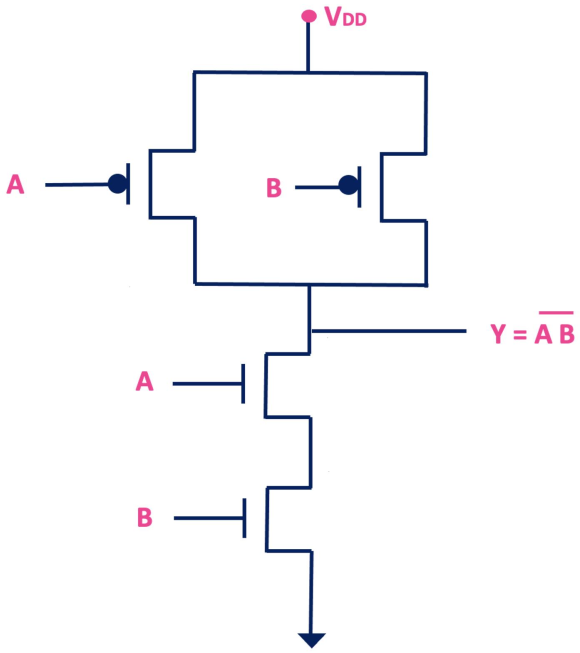


INPUT	LOGIC INPUT	OUTPUT	LOGIC OUTPUT
0v	0	Vdd	1
Vdd	1	0v	0

CMOS NAND Gate:

The below figure shows a 2-input Complementary MOS NAND gate. It consists of two series NMOS transistors between Y and Ground and two parallel PMOS transistors between Y and VDD.

If either input A or B is logic 0, at least one of the NMOS transistors will be OFF, breaking the path from Y to Ground. But at least one of the pMOS transistors will be ON, creating a path from Y to VDD.



Truth Table

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Two Input NAND Gate

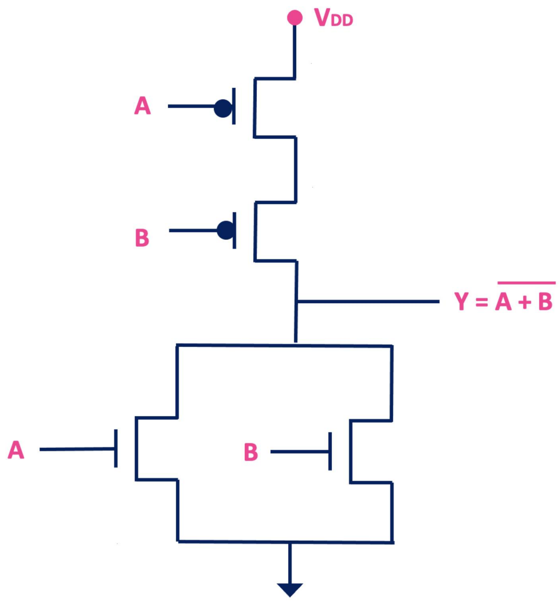
Hence, the output Y will be high. If both inputs are high, both of the nMOS transistors will be ON and both of the pMOS transistors will be OFF. Hence, the output will be logic low. The truth table of the NAND logic gate given in the below table.

A	B	Pull-Down Network	Pull-up Network	OUTPUT Y
0	0	OFF	ON	1
0	1	OFF	ON	1
1	0	OFF	ON	1
1	1	ON	OFF	0

CMOS NOR Gate:

A 2-input NOR gate is shown in the figure below. The NMOS transistors are in parallel to pull the output low when either input is high. The PMOS transistors are in series to pull the output high when both inputs are low, as given in the below table. The output is never left floating.

NOR Gate



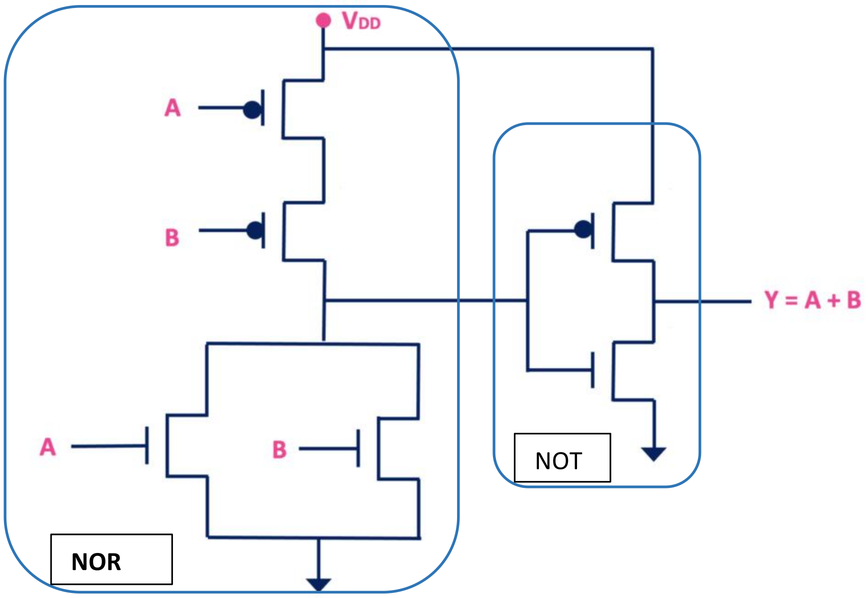
Truth Table

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

CMOS OR Gate:

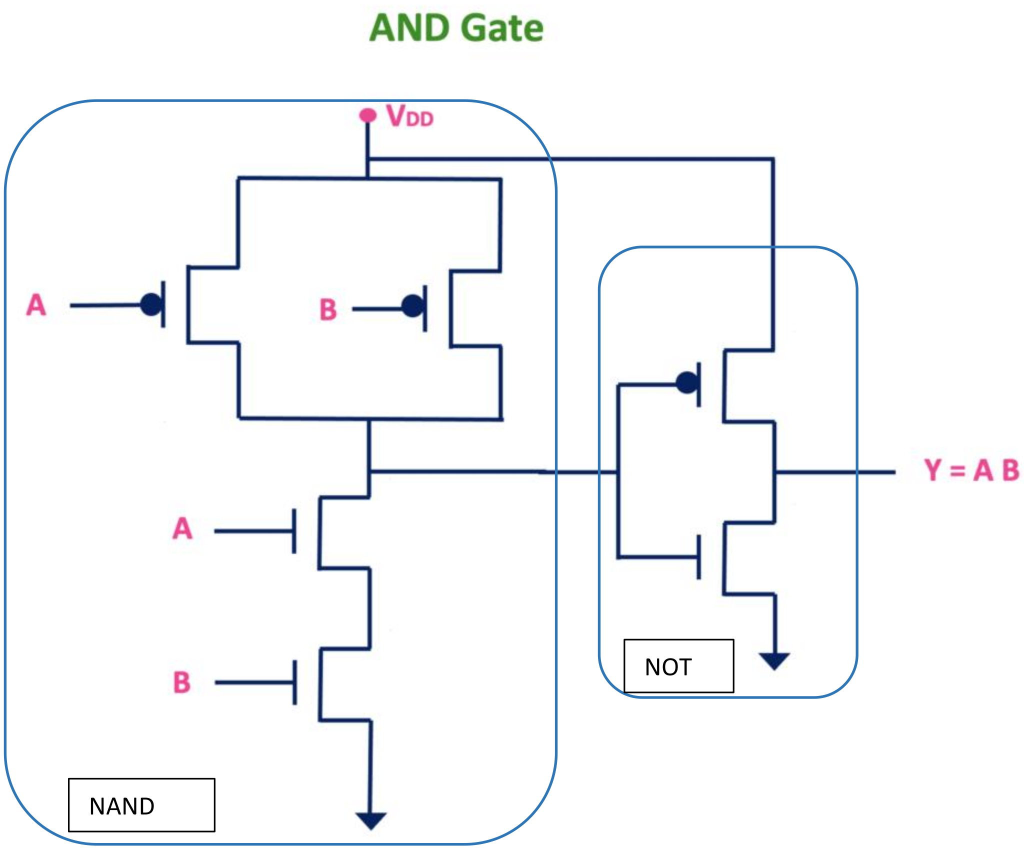
To implement the OR gate, just add the inverter at the output of the NOR gate. The CMOS OR gate is shown below.

OR Gate



CMOS AND Gate :

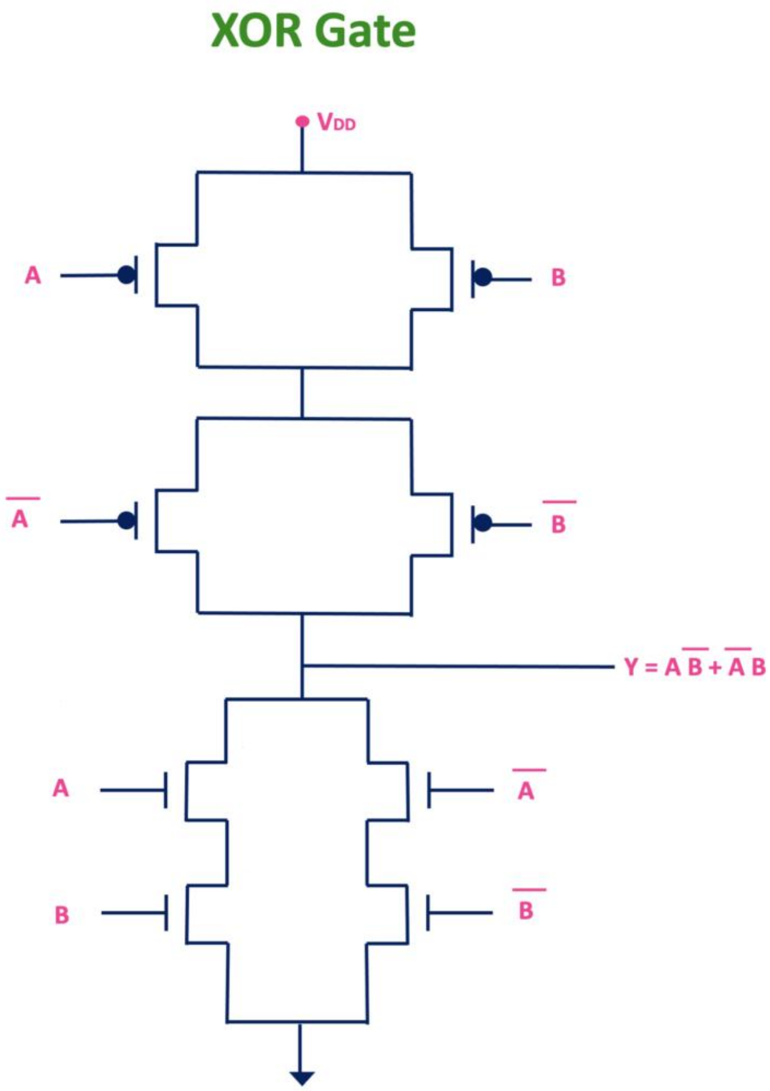
Similarly, by connecting the inverter at the output of the NAND gate, we can implement AND gate. The CMOS AND gate is shown below.



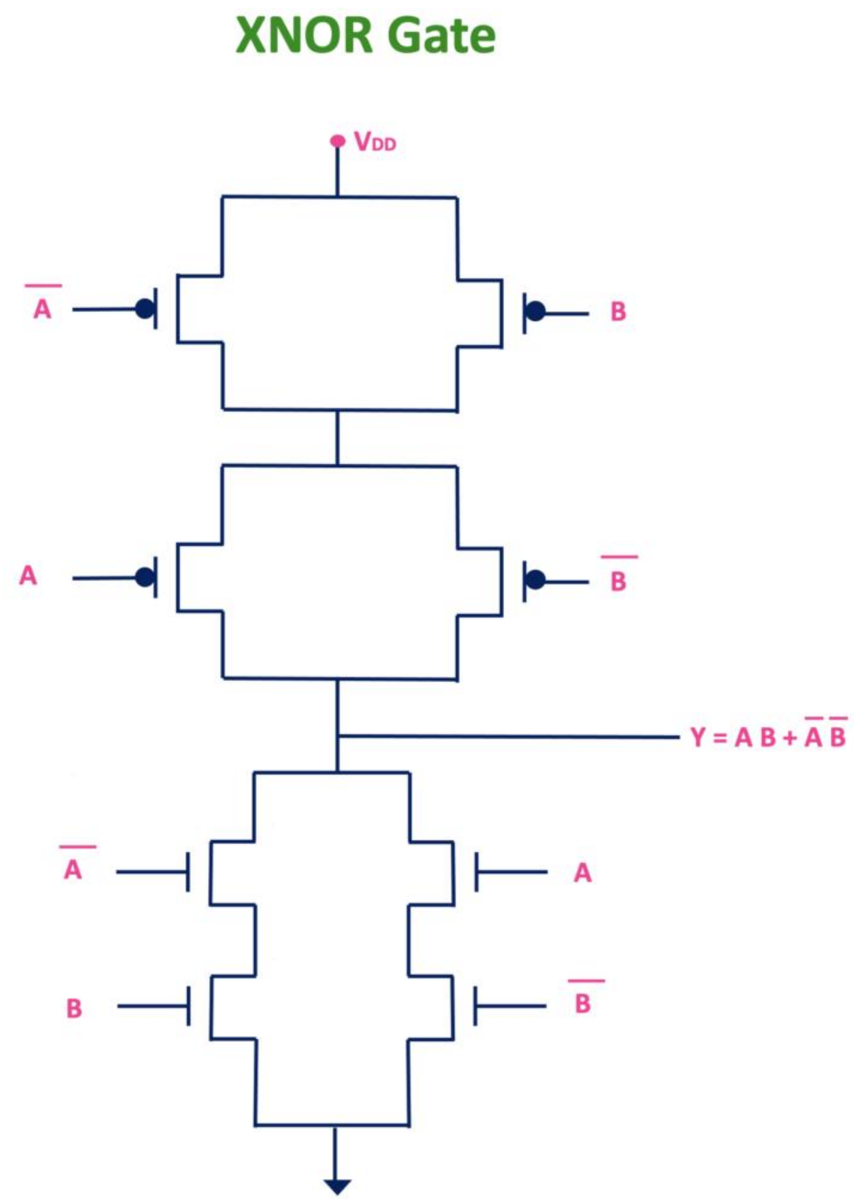
Implementation of XOR Gate and XNOR using CMOS Logic :

Similarly, the implementation of XOR and XNOR gate is shown below.

XOR Gate:

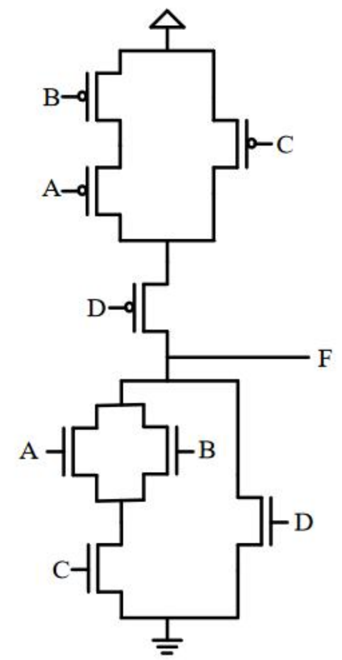


XNOR Gate :

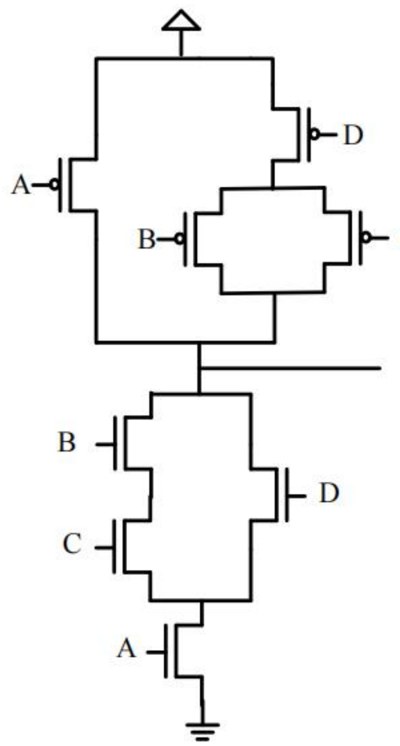

XNOR Gate
VDD
$\overline{A}$
B
A
$\overline{B}$
$Y = AB + \overline{A}\,\overline{B}$
$\overline{A}$
A
B
$\overline{B}$

DESIGN OF GENERAL BOOLEAN CIRCUITS USING CMOS TECHNOLOGY:

1. Please draw the minimum CMOS transistor network that implements the functionality of Boolean equation F= ((A+B) C + D)'. You can assume both the original and complemented versions of each literal are available as gate inputs.

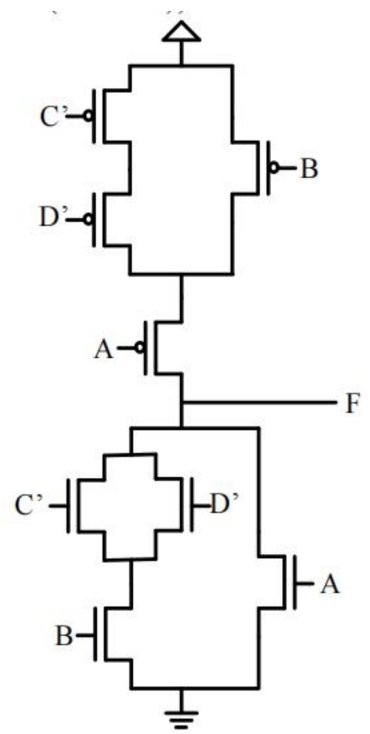


2. Please draw the minimum CMOS transistor network that implements the functionality of Boolean equation F= (A (B C + D))'. You can assume both the original and complemented versions of each literal are available as gate inputs.



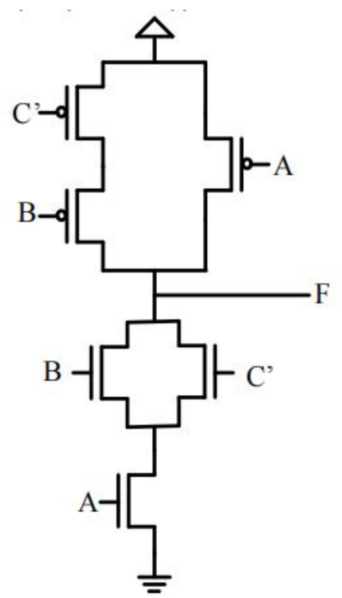
3. Please draw the minimum CMOS transistor network that implements the functionality of Boolean equation F= (A +(B' + CD)')'. You can assume both the original and complemented versions of each literal are available as gate inputs.

So, F= (A +(B' + CD)')' = (A + B(CD)')' = (A + B (C' + D'))'



4. Please draw the minimum CMOS transistor network that implements the functionality of Boolean equation F= (A' + B'C). You can assume both the original and complemented versions of each literal are available as gate inputs.

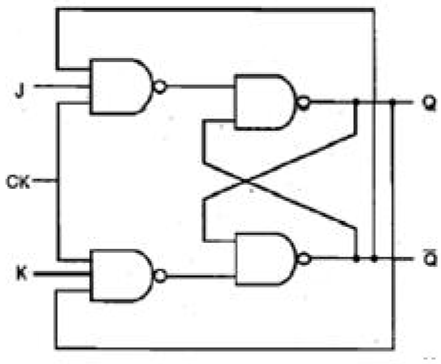
So, F = (A' + B'C) = ((A' + B'C)')'= (A (B'C)')' = (A (B + C'))'



JK FF using CMOS logic:

The problem in clocked SR latch, in which when both inputs S and R are activated at the same time, not allowed condition comes, hence to overcome this, two feedback lines are added from the output to input as shown below, and resulting circuit is called as JK latch.

Clk	J	K	Q_{n+1}	$\overline{Q_{n+1}}$
0	X	X	Q_n	$\overline{Q_n}$
1	0	0	Q_n	$\overline{Q_n}$
1	0	1	0	1
1	1	0	1	0
1	1	1	Race around	



Consider any one output. From gate level diagram we can write,

$$Q = \overline{\overline{J . Clk . \bar{Q}}\ \bar{Q}}$$

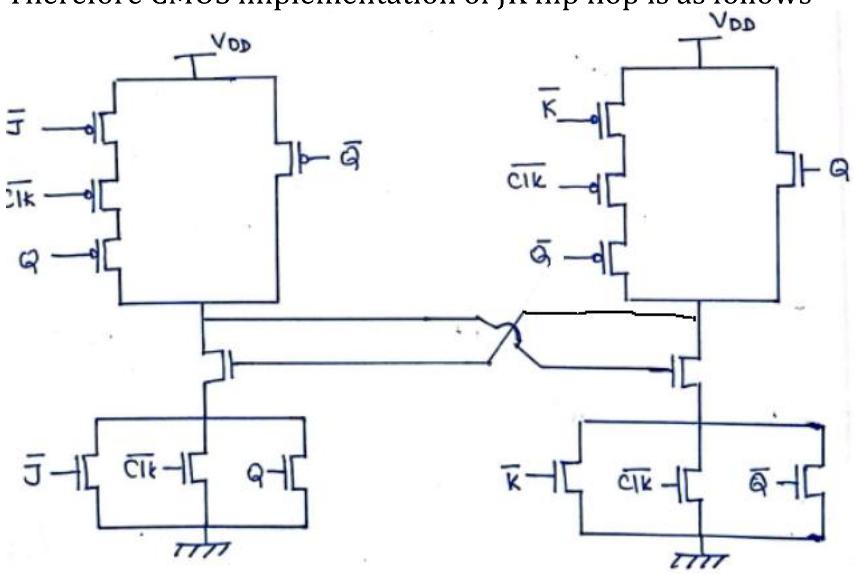
$$Q = \overline{(\bar{J} + \overline{Clk} + Q) . \bar{Q}} \qquad \text{..... De – Morgan's law}$$

For input K, assume

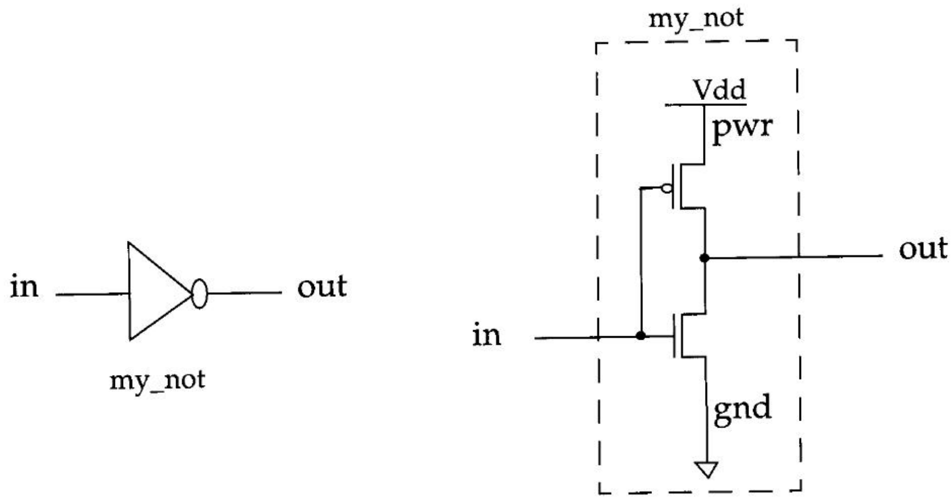
$$\bar{Q} = \overline{\overline{K . Clk . Q}\ Q}$$

$$\bar{Q} = \overline{(\bar{K} + \overline{Clk} + \bar{Q}) . Q} \qquad \text{..... De – Morgan's law}$$

Therefore CMOS implementation of JK flip flop is as follows-



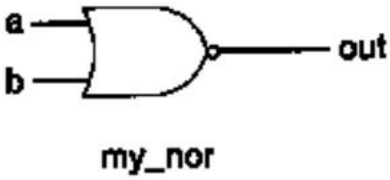
CMOS INVERTER VERILOG CODE (USING SWITCH LEVEL MODELING):

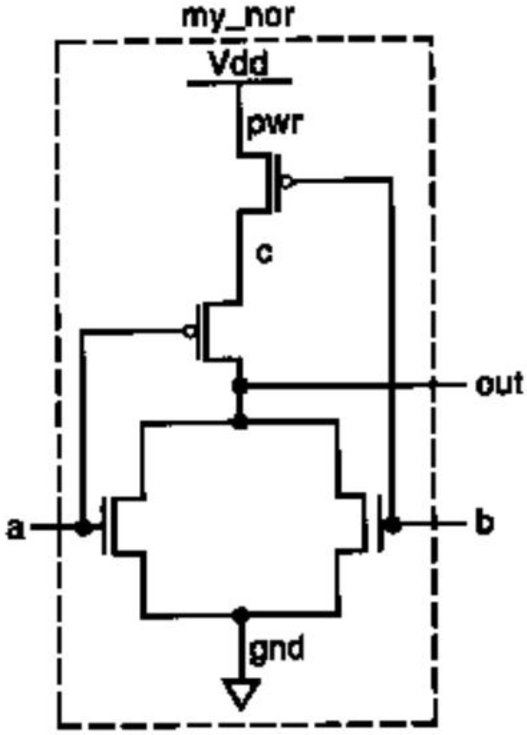



```
module my_not(out, in);

output out;
input in;

//declare power and ground
supply1 pwr;
supply0 gnd;

//instantiate nmos and pmos switches
pmos  (out, pwr, in);
nmos  (out, gnd, in);

endmodule
```

NOTE: TEST BENCH CONCEPT IS SAME AS DISCUSSED EARLIER

CMOS NOR LOGIC VERILOG CODE (USING SWITCH LEVEL MODELING):






```
module my_nor(out, a, b);

output out;
input a, b;

//internal wires
wire c;

//set up power and ground lines
supply1 pwr;     //pwr is connected to Vdd (power supply)
supply0 gnd ;    //gnd is connected to Vss(ground)

//instantiate pmos  switches
pmos  (c, pwr, b);
pmos  (out, c, a);

//instantiate nmos switches

nmos  (out, gnd, a);
nmos  (out, gnd, b);

endmodule
```

NOTE: TEST BENCH CONCEPT IS SAME AS DISCUSSED EARLIER