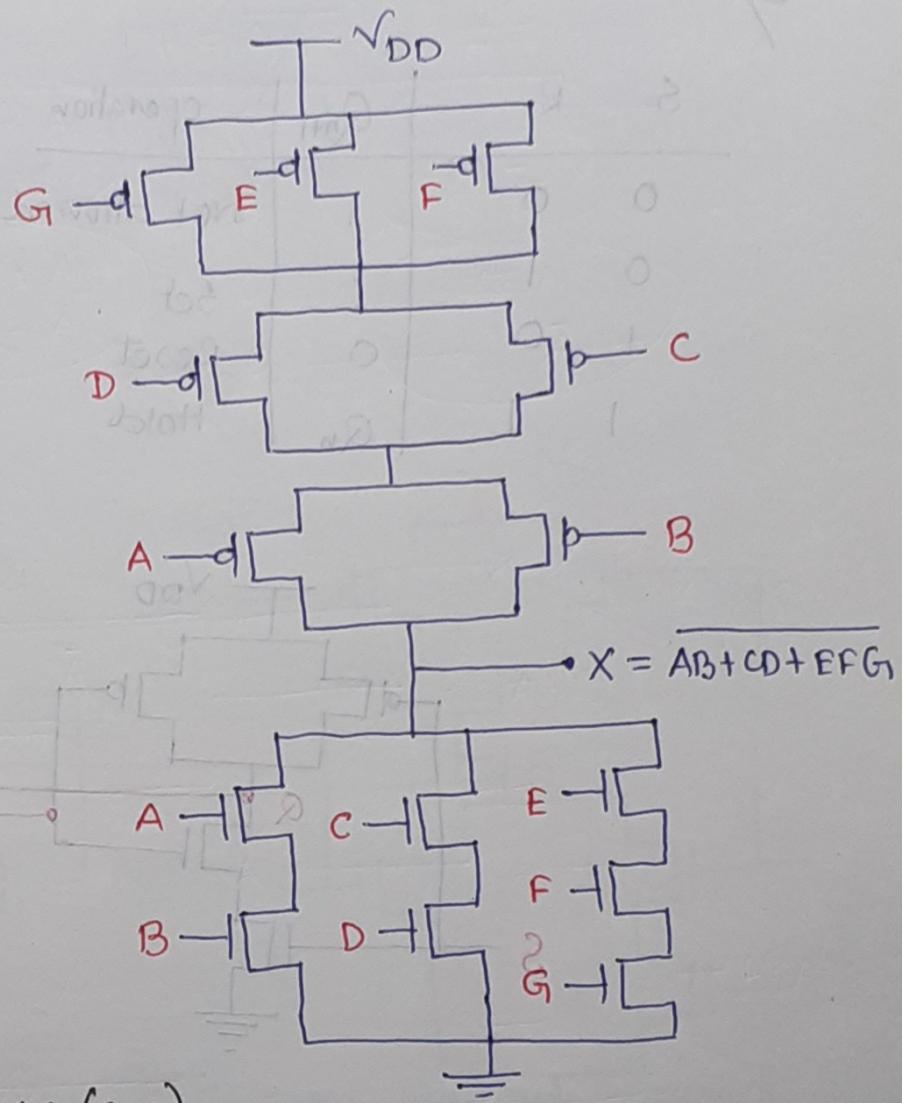


i) Implement & design using static CMOS logic. $X = \overline{AB} + CD + EFG$

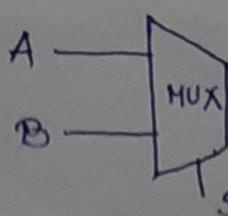
\Rightarrow



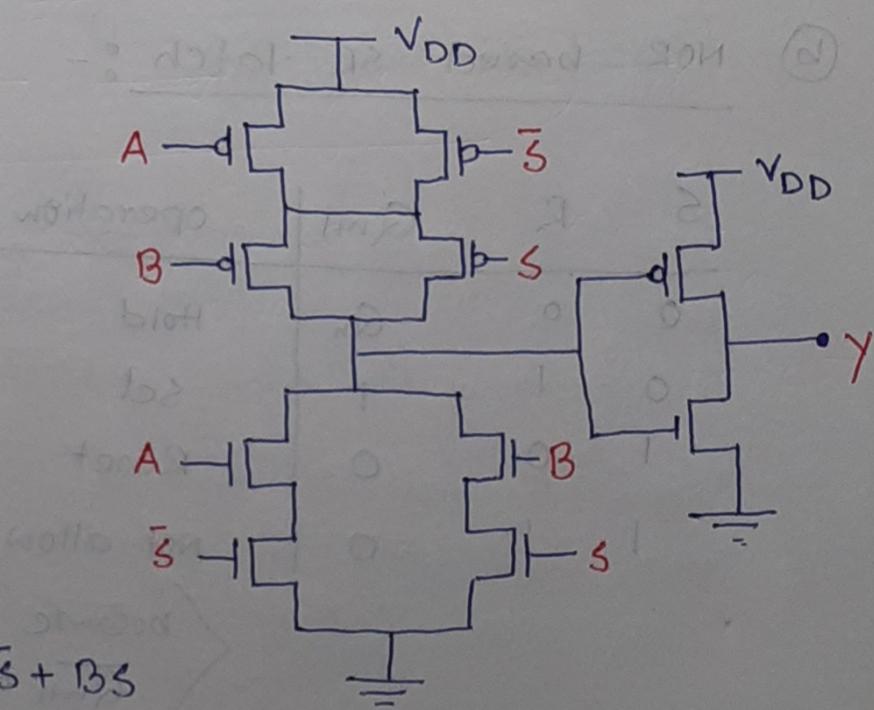
ii) Implement & Design MUX (2:1) using static CMOS logic.

\Rightarrow

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



$$y = A\bar{s} + Bs$$

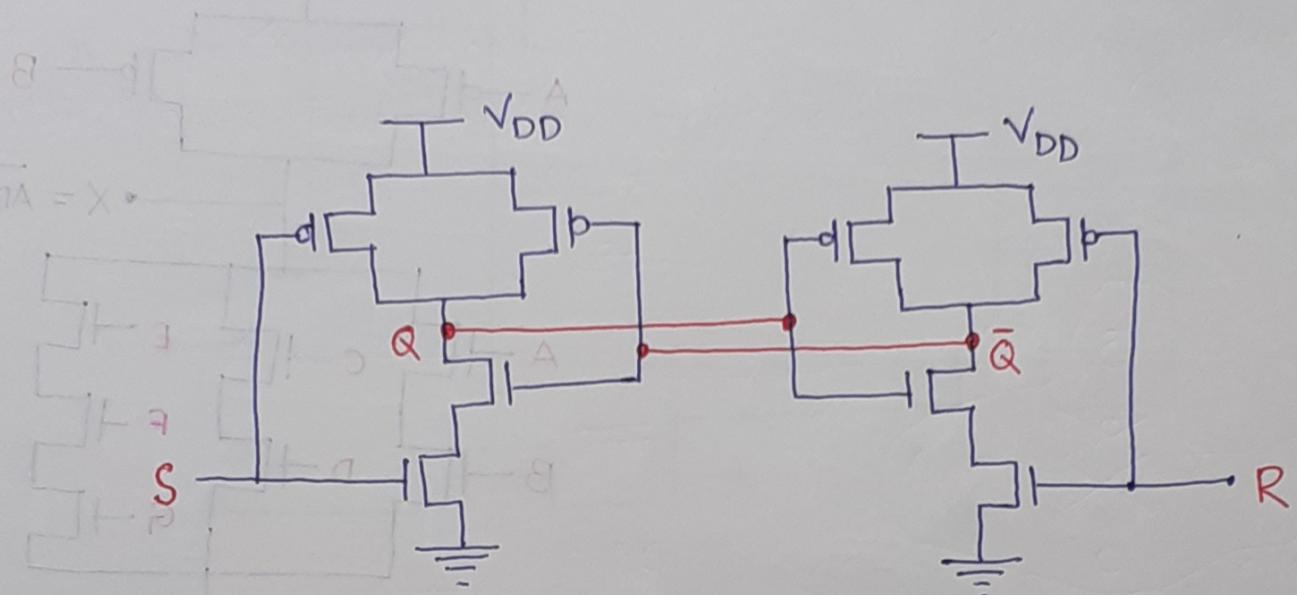
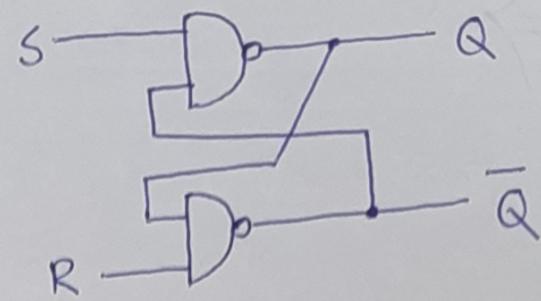


Static CMOS realization of NAND based SR

① latch circuit.



S	R	Q_{n+1}	operation
0	0	1	not allowed
0	1	1	Set
1	0	0	Reset
1	1	Q_n	Hold

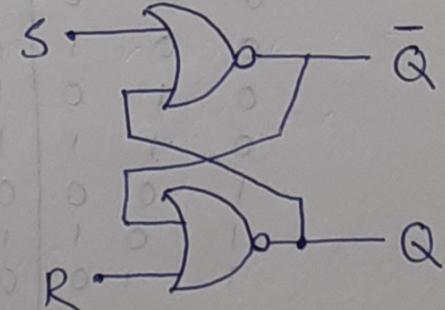


[NAND based SR latch]

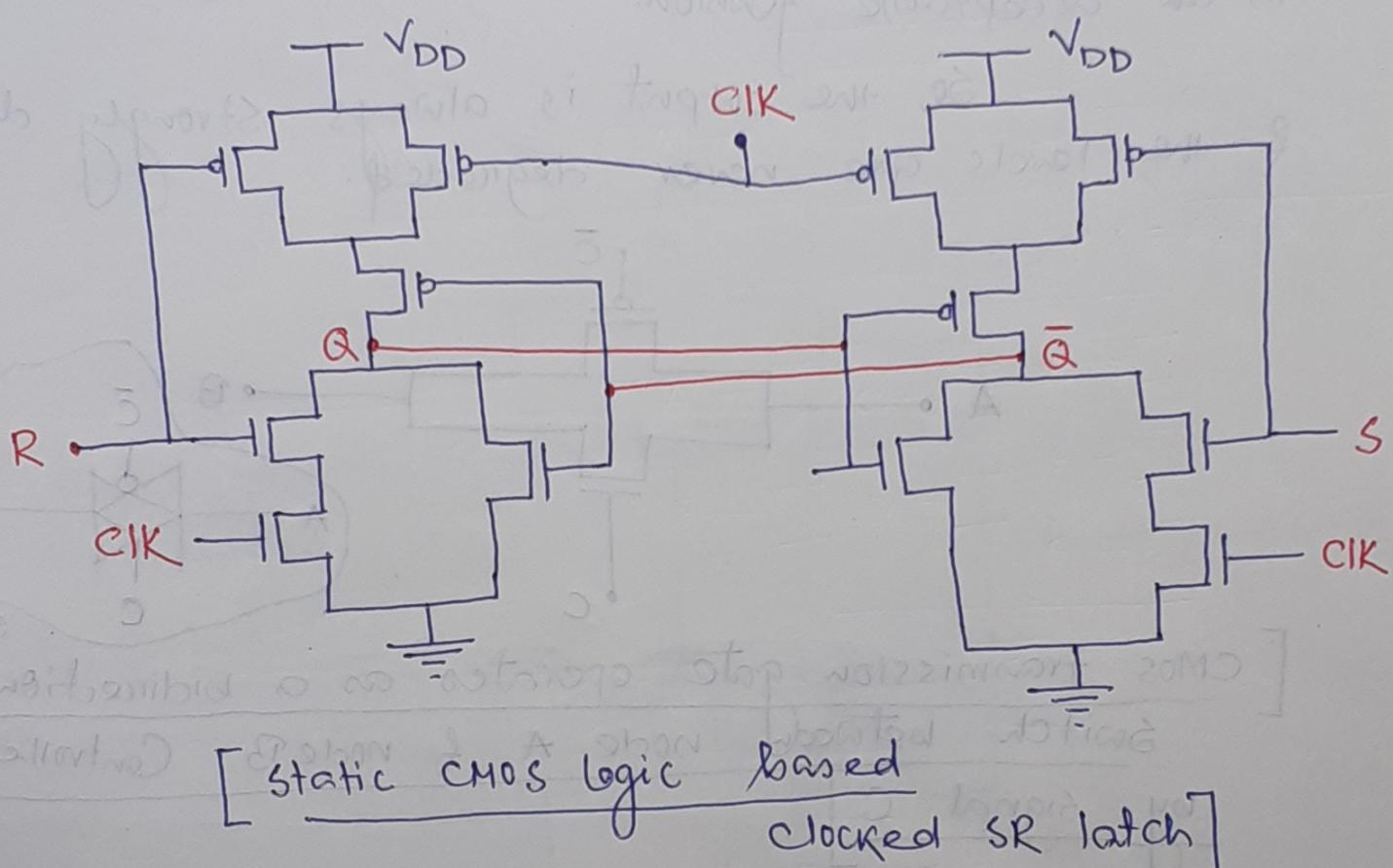
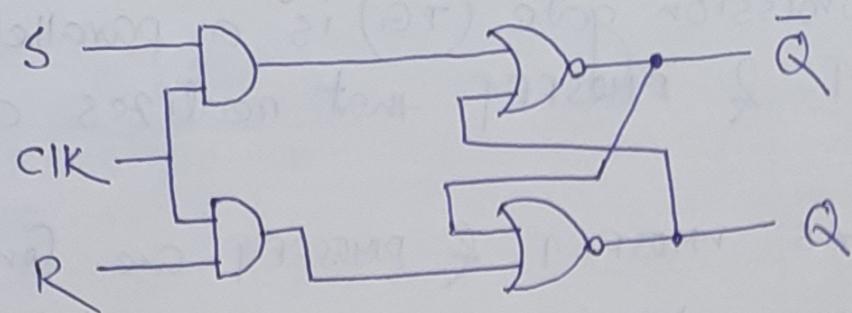
② NOR based SR latch :-

S	R	Q_{n+1}	operation
0	0	Q_n	Hold
0	1	1	Set
1	0	0	Reset
1	1	0	NOT allowed

because
 $\overline{Q_{n+1}}$ also = 0



iv) NOR based clocked SR latch \Rightarrow

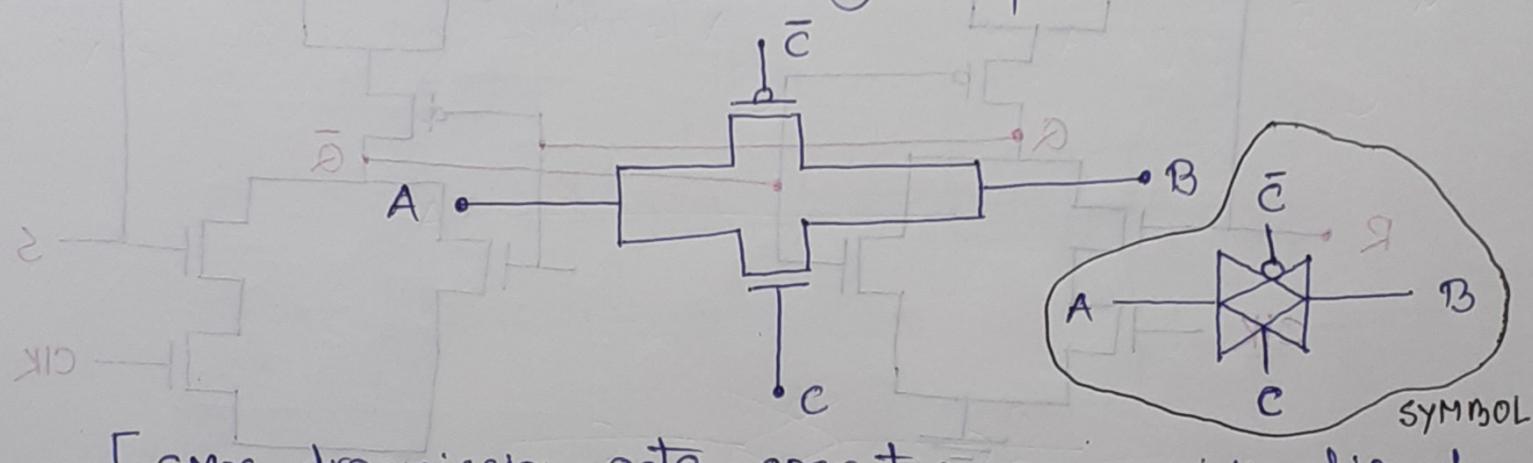


*> Introduction to Transmission gate (TG):-

CMOS transmission gate (TG) is a parallel connection of nMOSFET & pMOSFET that realizes a simple switch.

AS nMOSFET & pMOSFET are connected in parallel; nMOS transistor passes logic '0' & pMOS transistor passes logic '1' without degradation & in an acceptable fashion.

So, the output is always strongly driven & the levels are never degraded.



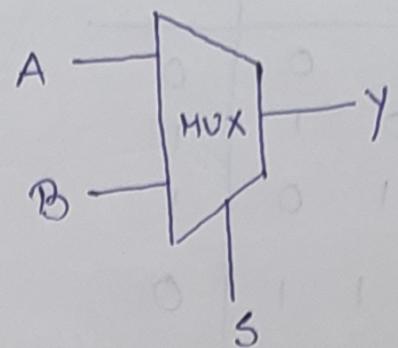
[CMOS transmission gate operates as a bidirectional switch between node A & node B controlled by signal C]

- i) When signal 'c' is high, \bar{C} is low \rightarrow thus both nMOS & pMOS transistors are ON; nodes A & B are short circuited; so the input logic is transferred to the output.
- ii) When signal 'c' is low, \bar{C} is high; thus both nMOS & pMOS transistors are OFF. Nodes A & B are open circuited \rightarrow called high impedance state.

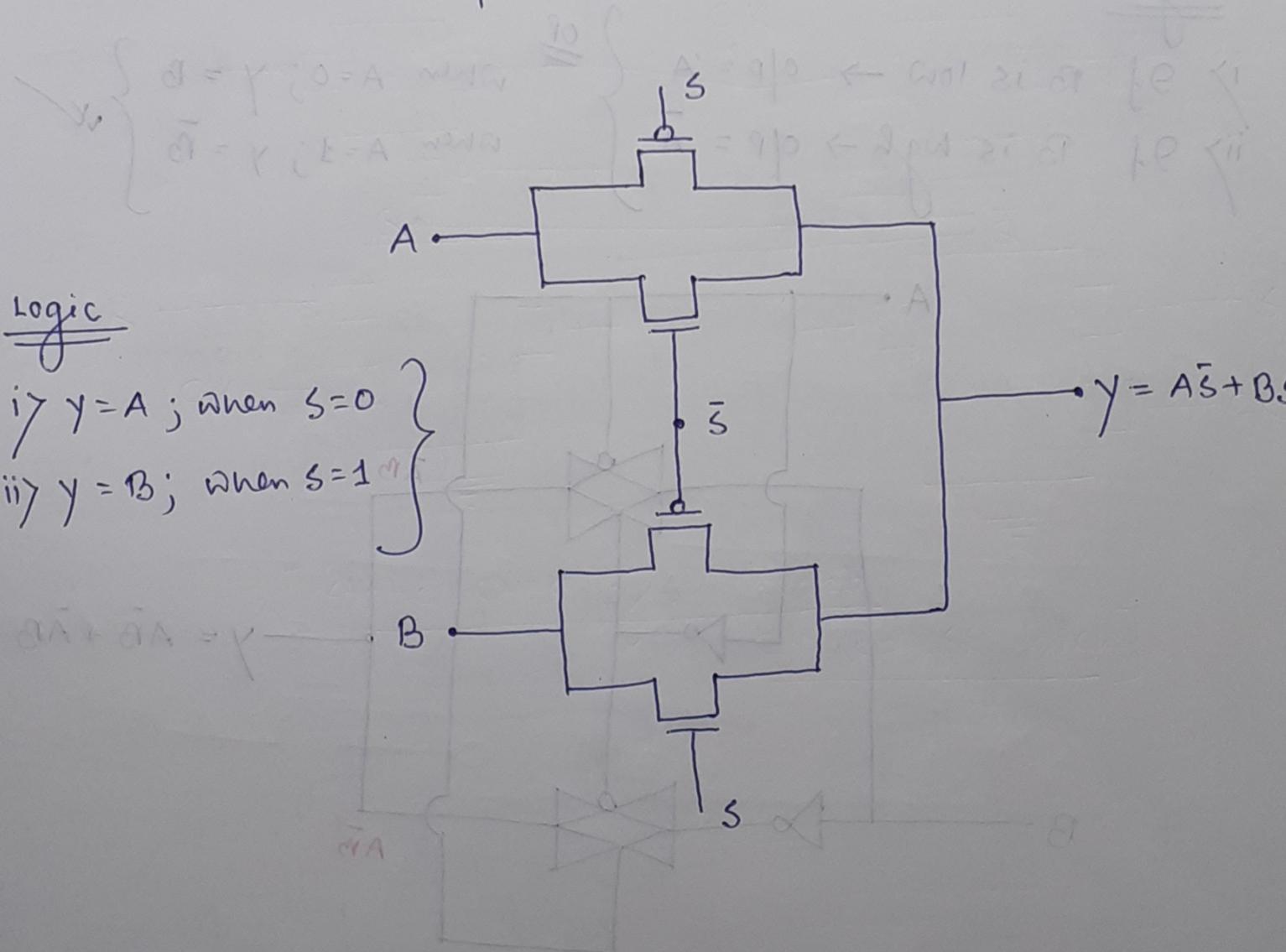
* Implementation of 2:1 MUX using CMOS TG.

\Rightarrow

S	A	B	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

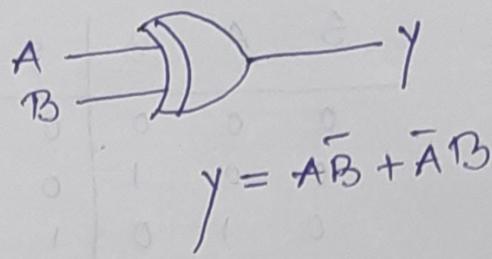


$$y = A\bar{S} + BS$$



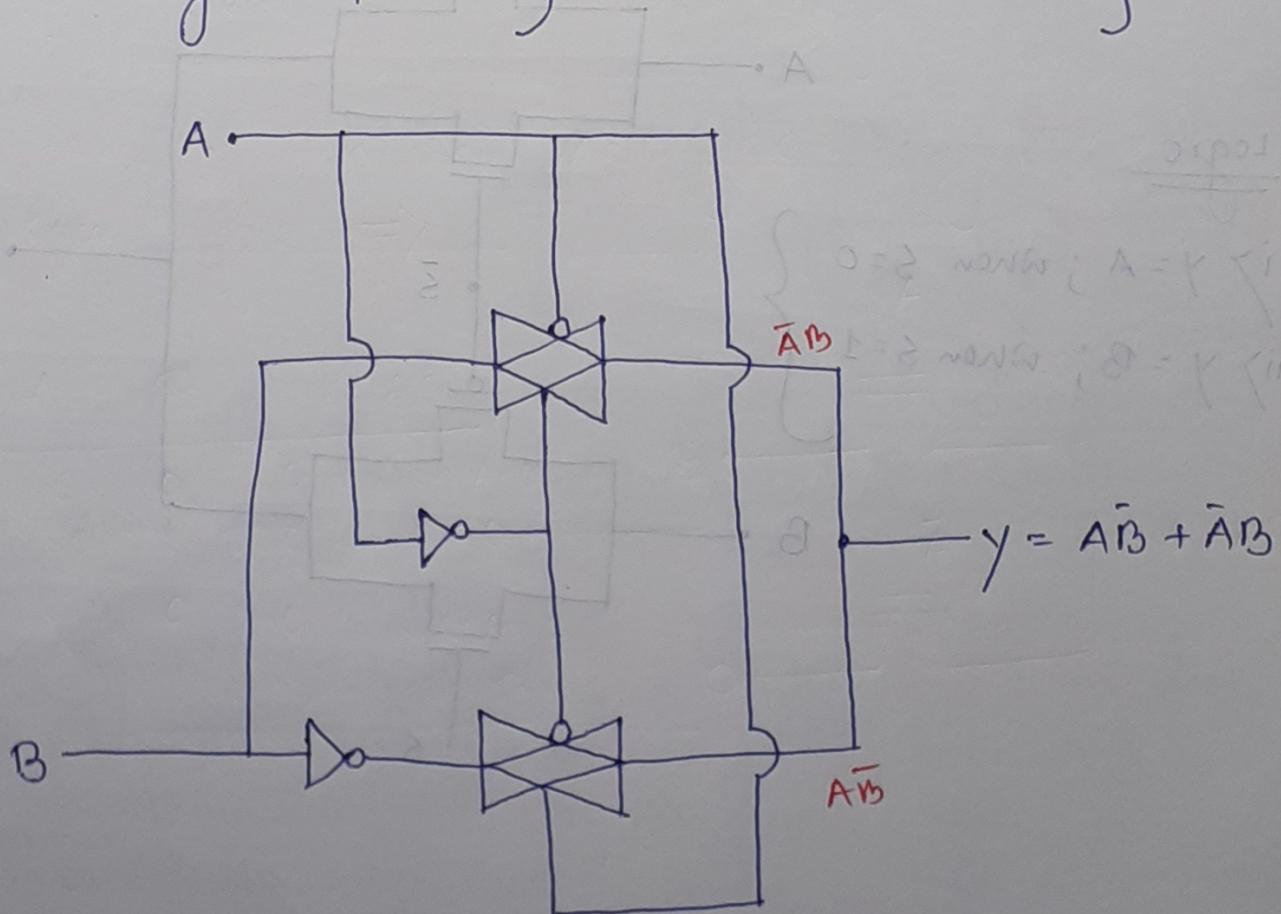
* Implement XOR gate using CMOS TG \Rightarrow

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



Logic

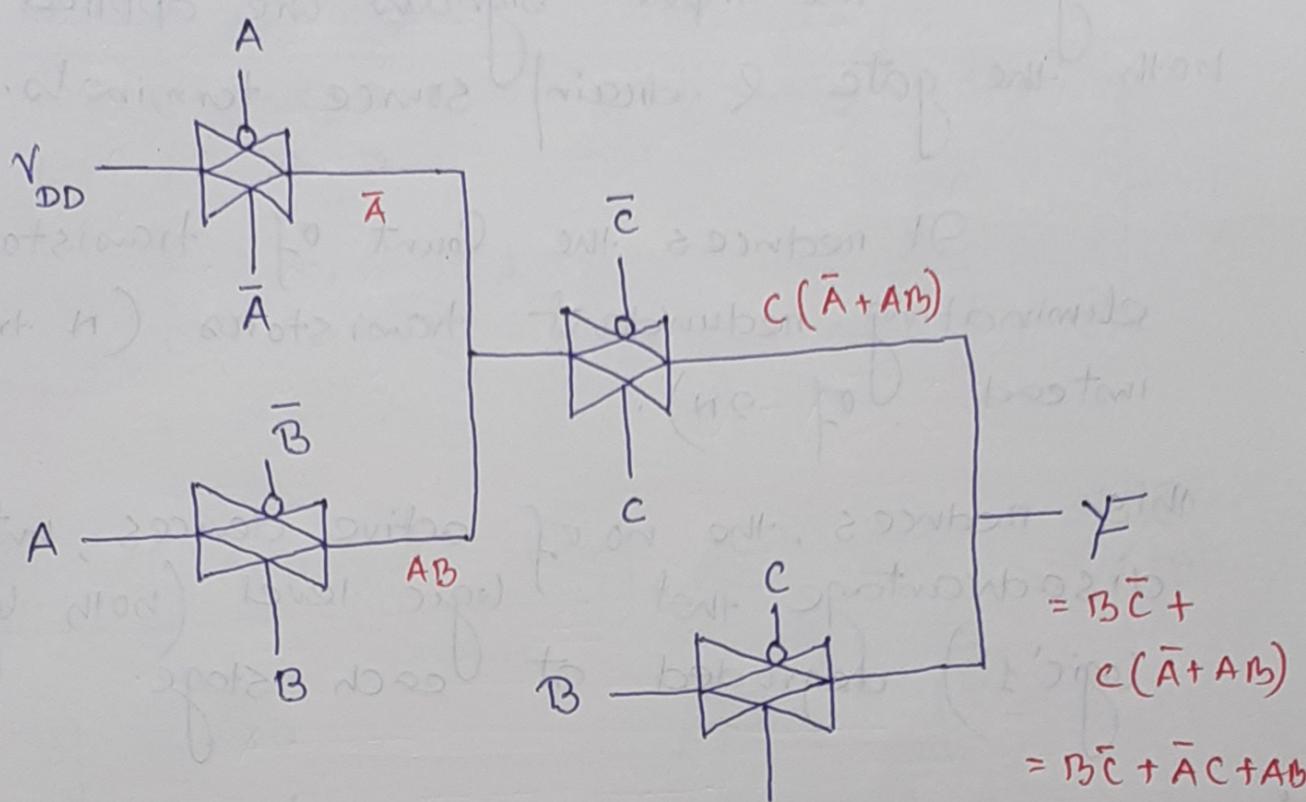
- i) If B is low \rightarrow O/P = A
 - ii) If B is high \rightarrow O/P = \bar{A}
- } OR
- when A=0; Y = B
when A=1; Y = \bar{B}
- } ✓



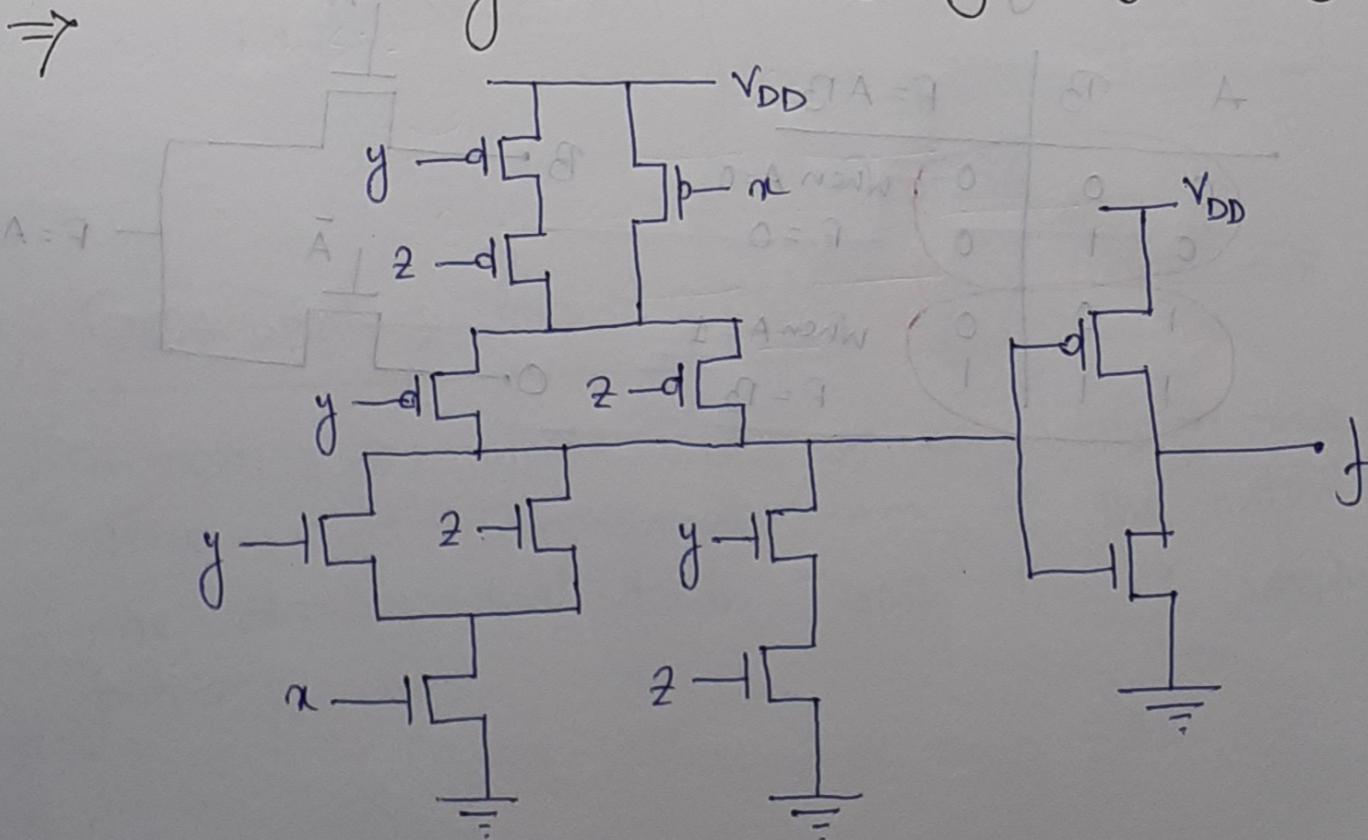
* Design & Implement boolean function, \rightarrow

$$f = ABC + \bar{A}C + B\bar{C}$$

\Rightarrow 3t buildings are adopted



* Implement the following function $f = xy + xz + yz$ using static CMOS logic



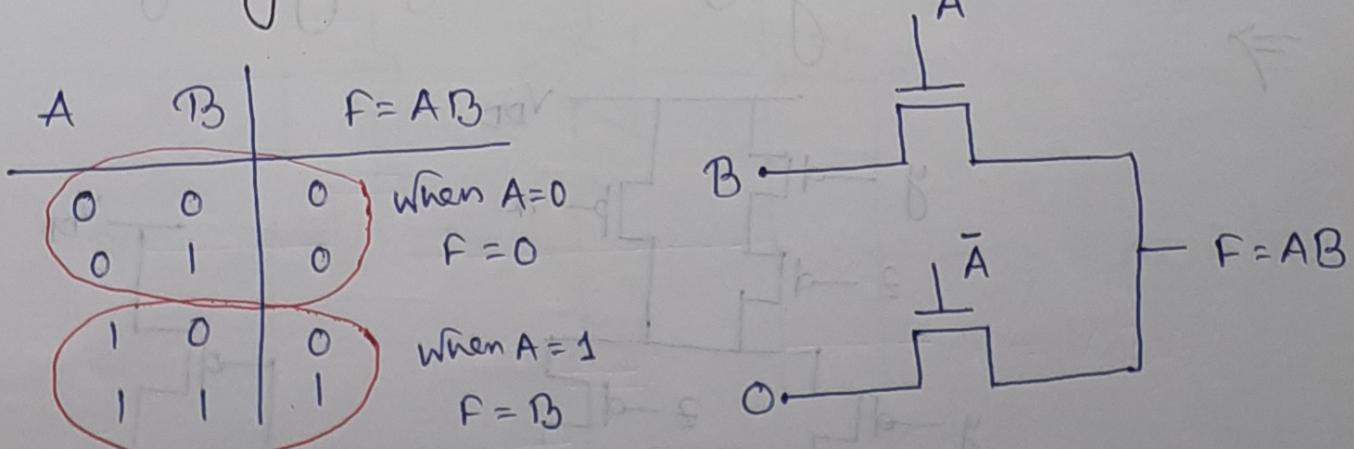
*pass transistor logic \rightarrow In pass transistor

logic, only nMOS transistors are used to design the logic. The input signals are applied to both the gate & drain/source terminals.

It reduces the count of transistors by eliminating redundant transistors (N transistors instead of $2N$).

This reduces the no of active devices, but the disadvantage is that logic level (both logic '0' & logic '1') degraded at each stage.

i) Realization of two input AND gate using pass transistor logic \Rightarrow



ii) Design of XOR gate using PTL \rightarrow

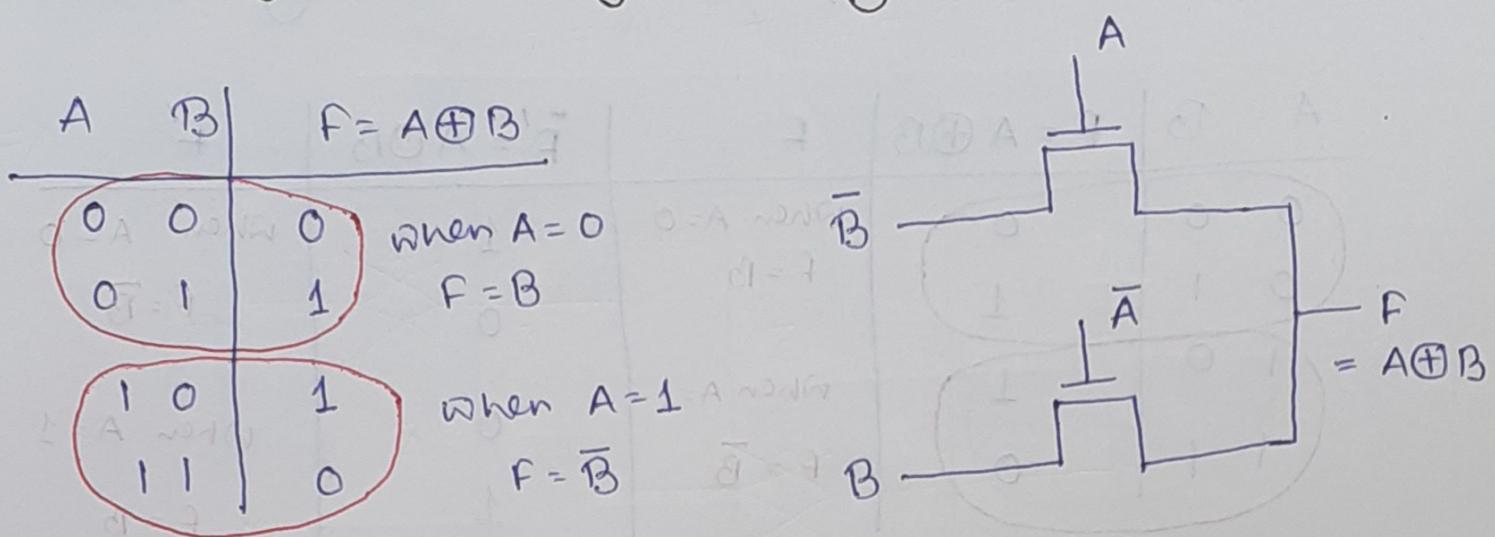
A	B	$F = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

when $A=0$

$$F=B$$

when $A=1$

$$F=\bar{B}$$



*) Design of AND gate using PTL \rightarrow different style

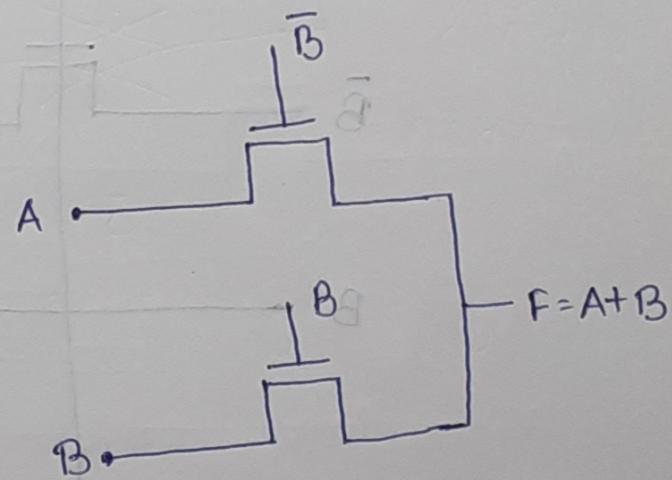
A	B	$F = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

when $B=0$

$$F=A$$

when $B=1$

$$F=B$$

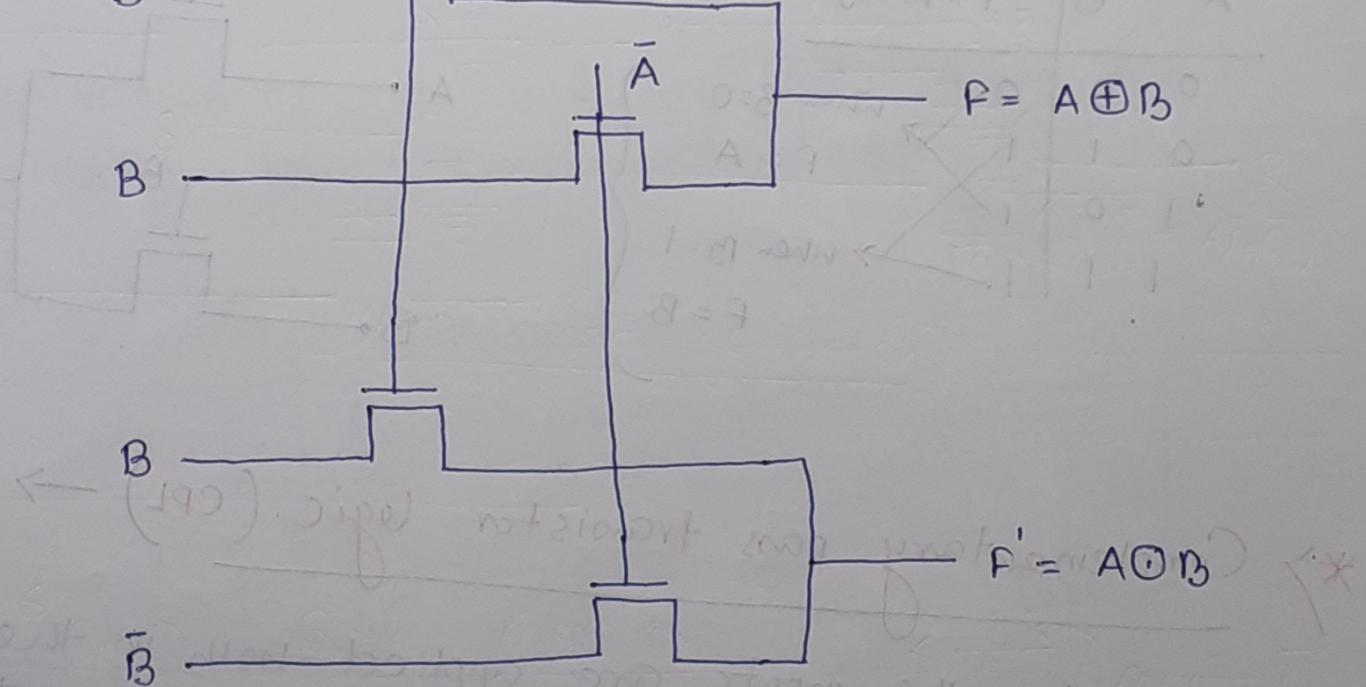
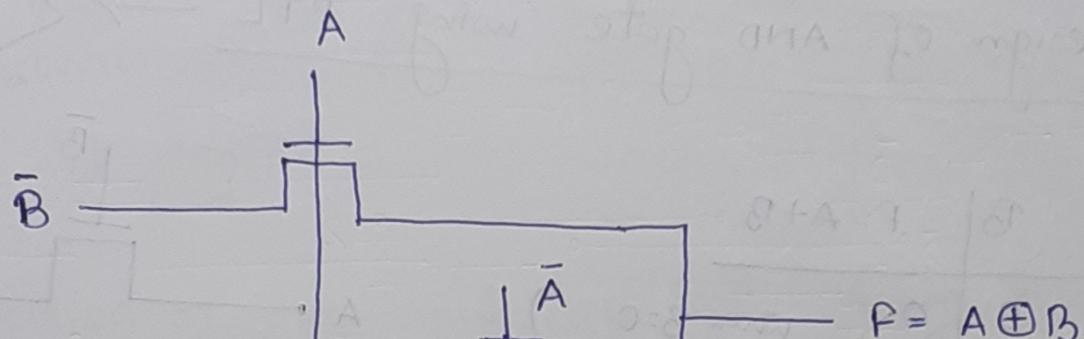


*) Complementary pass transistor logic (CPL) \rightarrow

In CPL, the inputs are applied both in the true and complement form & the outputs are also evaluated in both true & complement form.

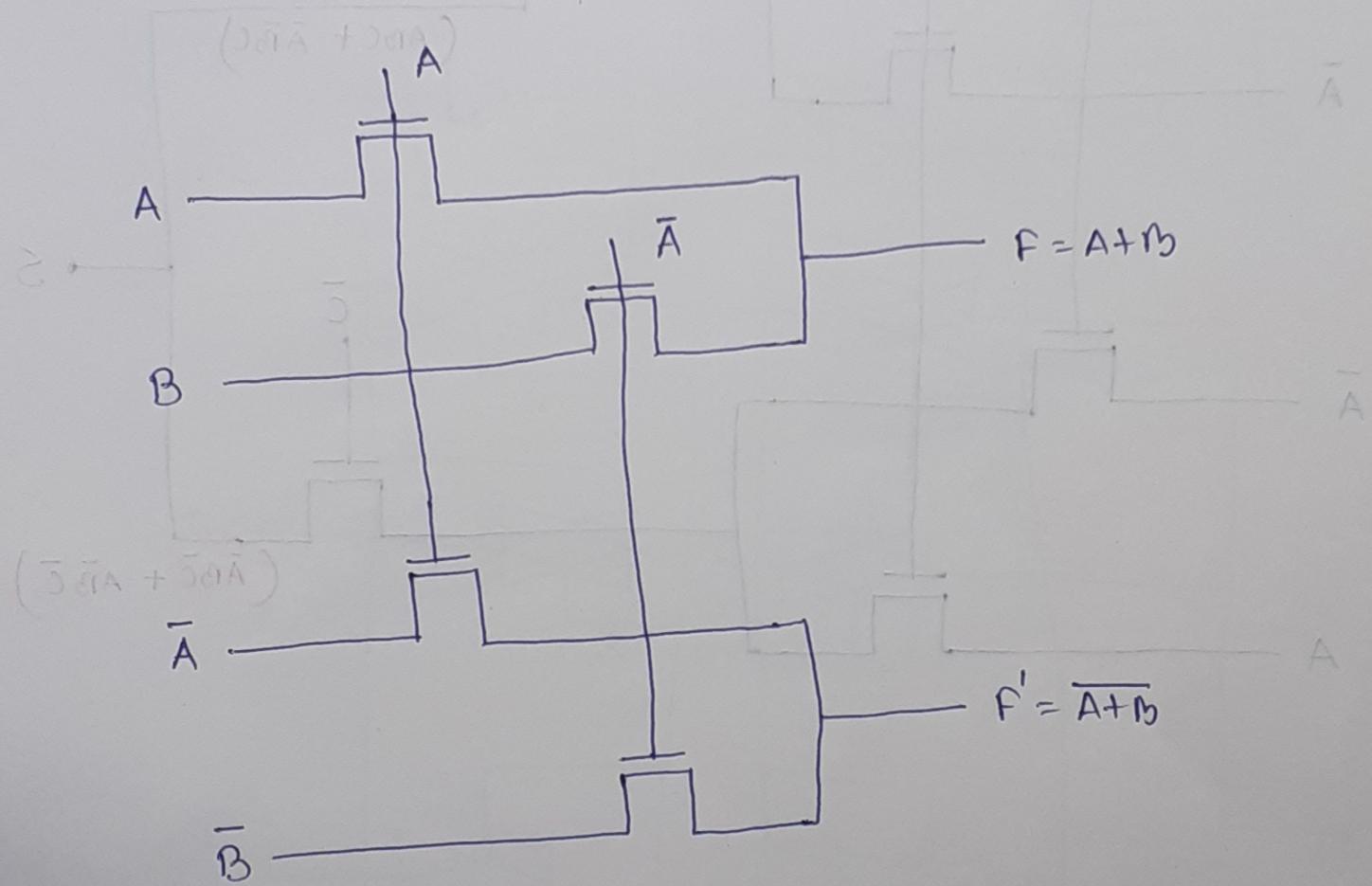
→ Realization of XOR / XNOR function using CPL →

A	B	$F = A \oplus B$	F	$F' = A \odot B$	f'
0	0	0	When $A=0$ $F=B$	1	When $A=0$ $F=\bar{B}$
0	1	1		0	
1	0	1	When $A=1$ $F=\bar{B}$	0	When $A=1$ $F=B$
1	1	0		1	

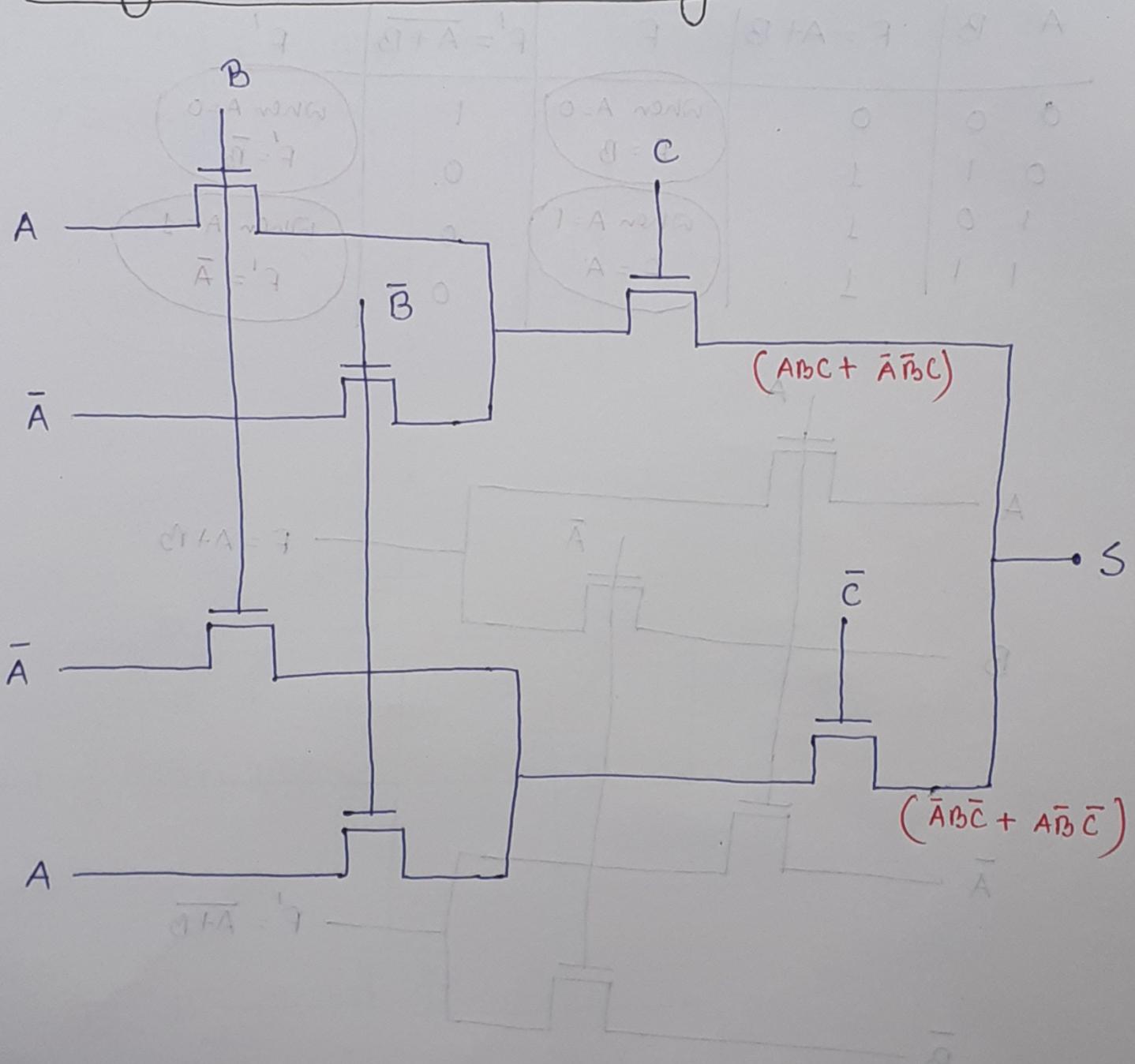


ii) Design of OR/NOR gate using CPL \rightarrow

A	B	$F = A + B$	F	$F' = \overline{A + B}$	F'
0	0	0	When $A=0$ $F=B$	1	When $A=0$ $F'=B$
0	1	1		0	
1	0	1	When $A=1$ $F=A$	0	When $A=1$ $F'=\bar{A}$
1	1	1		0	

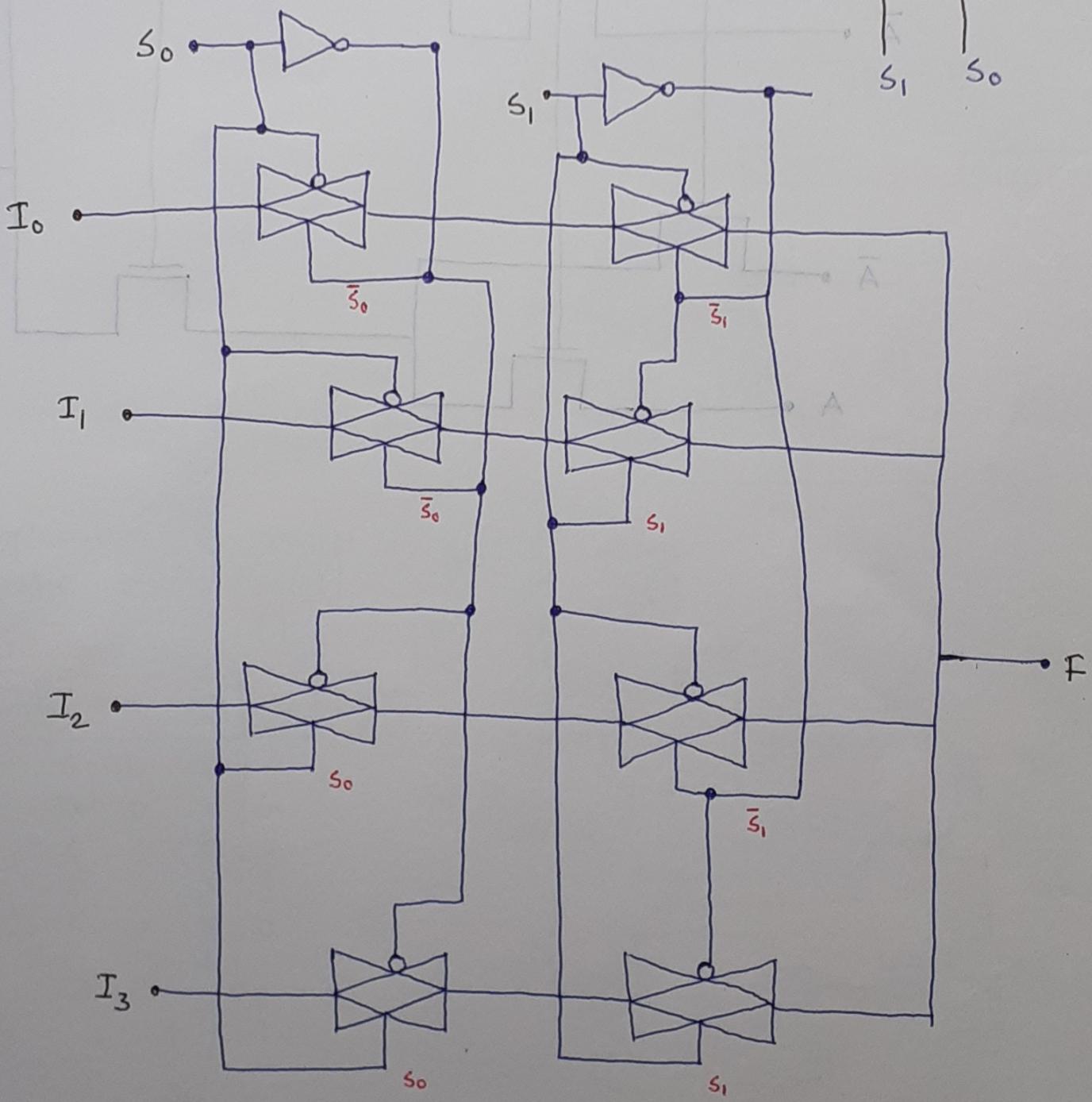
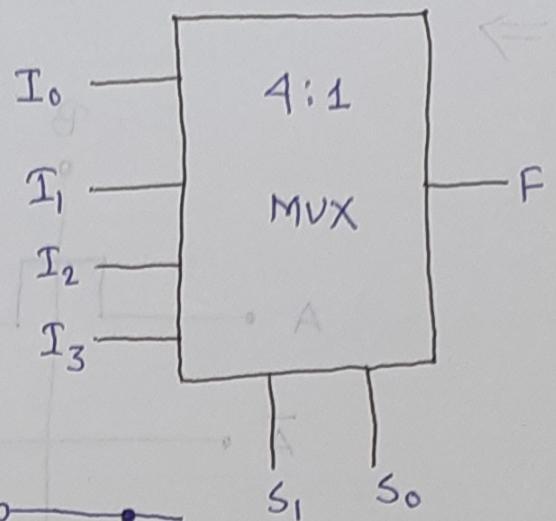


Implement the function $S = ABC + A\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$
 using nMOS pass transistor logic.



Design of 4:1 MUX using TG

$$F = \bar{S}_0 \bar{S}_1 I_0 + \bar{S}_0 S_1 I_1 + S_0 \bar{S}_1 I_2 + S_0 S_1 I_3$$



*> Implement the function $S = ABC + A\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$
 which gives the sum of two inputs with a carry bit
 using NMOS pass transistor logic.

=>

