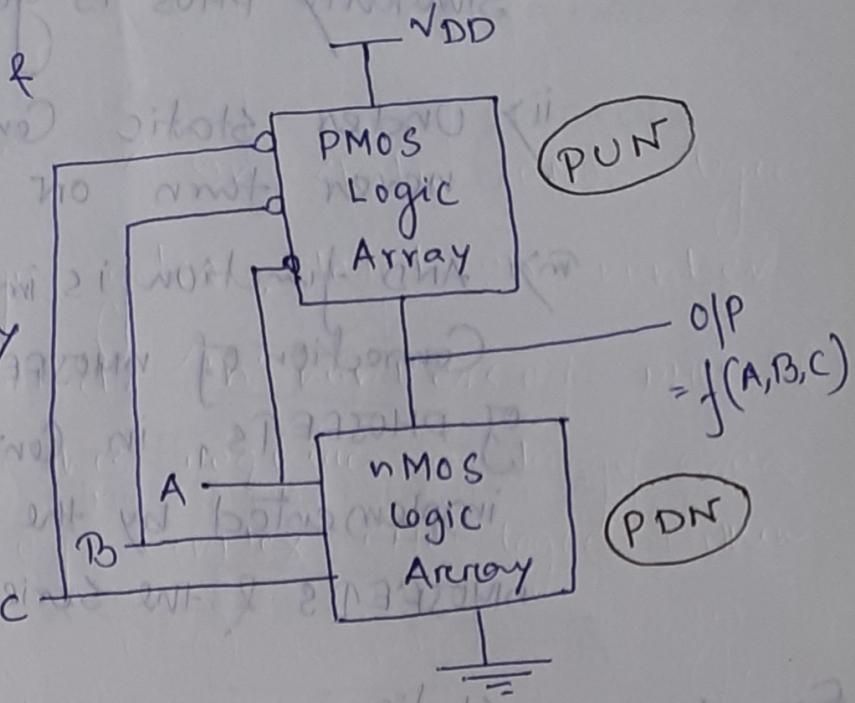


Static Combinational Digital logic Design

(4)

In a static logic circuit output node has a low resistance path either to V_{DD} or ground.

The pull up network (PUN) & pull down network (PDN) are operated by input voltages in a complementary fashion.



[general structure of static CMOS logic structure]

When pull down network (PDN) conducts, the output node establishes a low resistance path with ground \langle PDN transistors are ON & PUN transistors are OFF \rangle resulting in logic '0' output.

When pull up network (PUN) conducts, the output node establishes a low resistance path with V_{DD} \langle PUN transistors are ON & PDN transistors are OFF \rangle resulting in logic '1' output.

There is never a path through ON transistors from logic '1' to '0' supplies for any combination of inputs. \rightarrow which provides low static power dissipation in

Static CMOS logic.

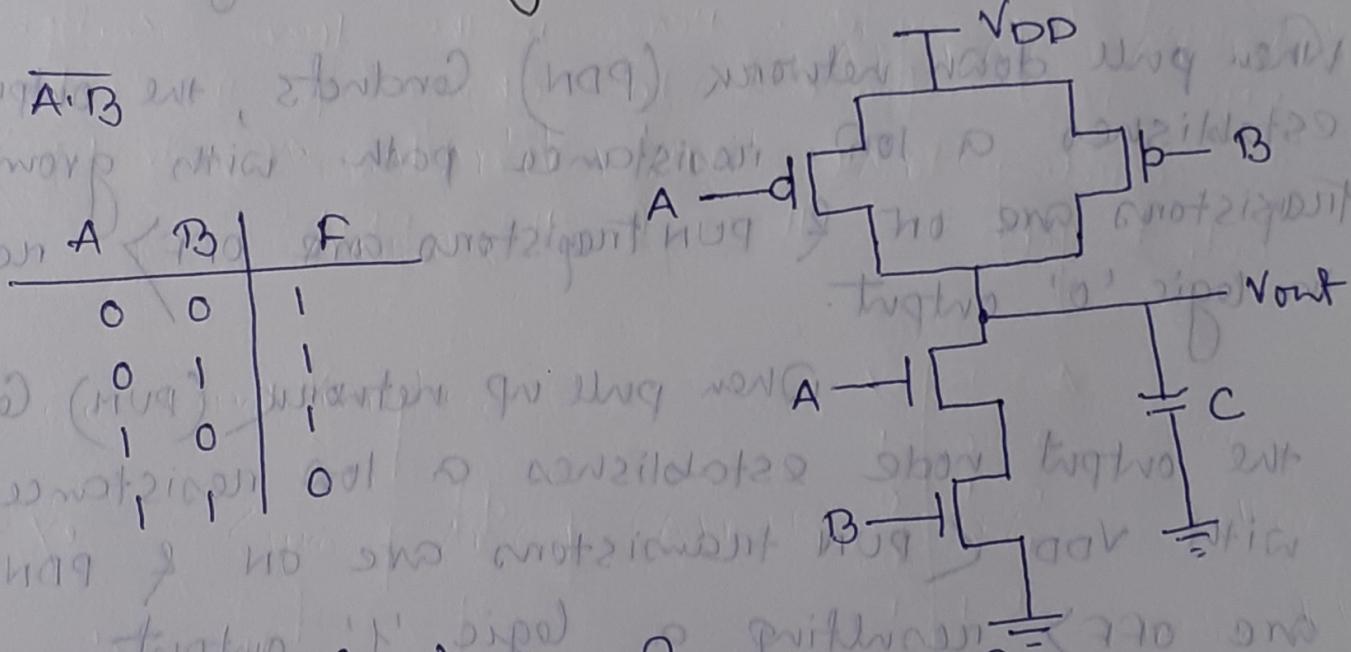
N.TB \Rightarrow

i) nMOS transistors is good for passing logic '0' < without signal degradation >;
Similarly PMOS is good for passing logic '1'.

ii) Under static Condition PUN & PDN never turn ON Simultaneously.
iii) AND function is implemented by the series Connection of nMOSFET & the parallel Connection of PMOSFETs, in contrast OR function is implemented by the parallel Connection of nMOSFETs & the series Connection of PMOSFET's.

Example:- Static

i) CMOS NAND gate.



The output voltage (V_{out}) = 0V; if and only if both the two input voltages are high. $\langle A = B = V_{DD} \rangle$

ii) static CMOS XOR gate

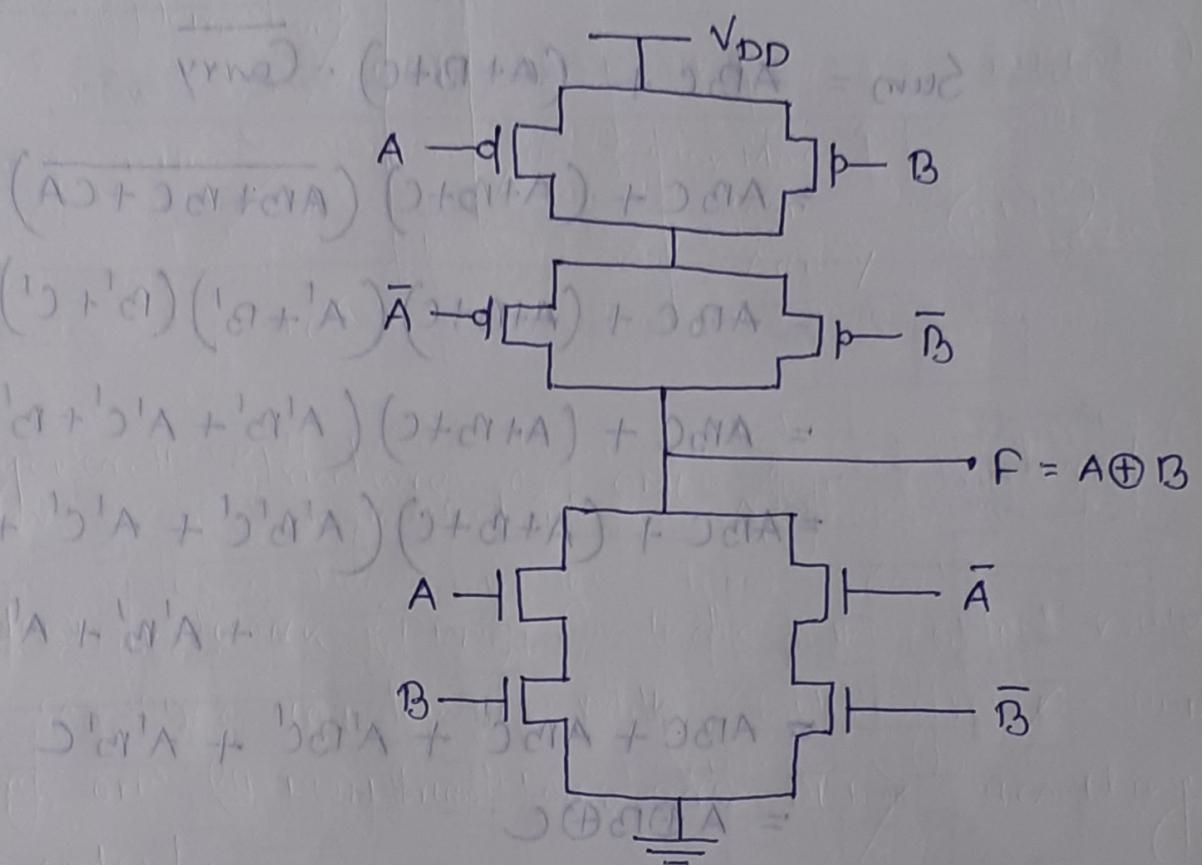
\Rightarrow

$$F = A \oplus B = AB' + A'B$$

To realize XOR gate in CMOS logic, we have to realize the function $\Rightarrow F' = \overline{A \oplus B} = \overline{AB' + A'B}$

$$= AB + A'B'$$

< as (CMOS) always generate inverted logic .



with {

⇒ Full adder <1-bit> circuit using static CMOS logic.

$$\Rightarrow \text{Sum} = A \oplus B \oplus C = ABC + AB'C' + A'B'C + A'B'C'$$

$$= ABC + (A+B+C) \cdot \overline{\text{Carry}}$$

$$\& \text{Carry} = AB + BC + CA$$

$$= AB + C(A+B)$$

Here,

$$\text{Sum} = ABC + (A+B+C) \cdot \overline{\text{Carry}}$$

$$= ABC + (A+B+C)(\overline{AB+BC+CA})$$

$$= ABC + (A+B+C)(A'+B')(B'+C')(C'+A')$$

$$= ABC + (A+B+C)(A'B'+A'C'+B'C'+B'C)(C'+A')$$

$$= ABC + (A+B+C)(A'B'C' + A'C' + B'C' + B'C + A'B' + A'B'C')$$

$$+ A'B' + A'C' + A'B' + A'B'C'$$

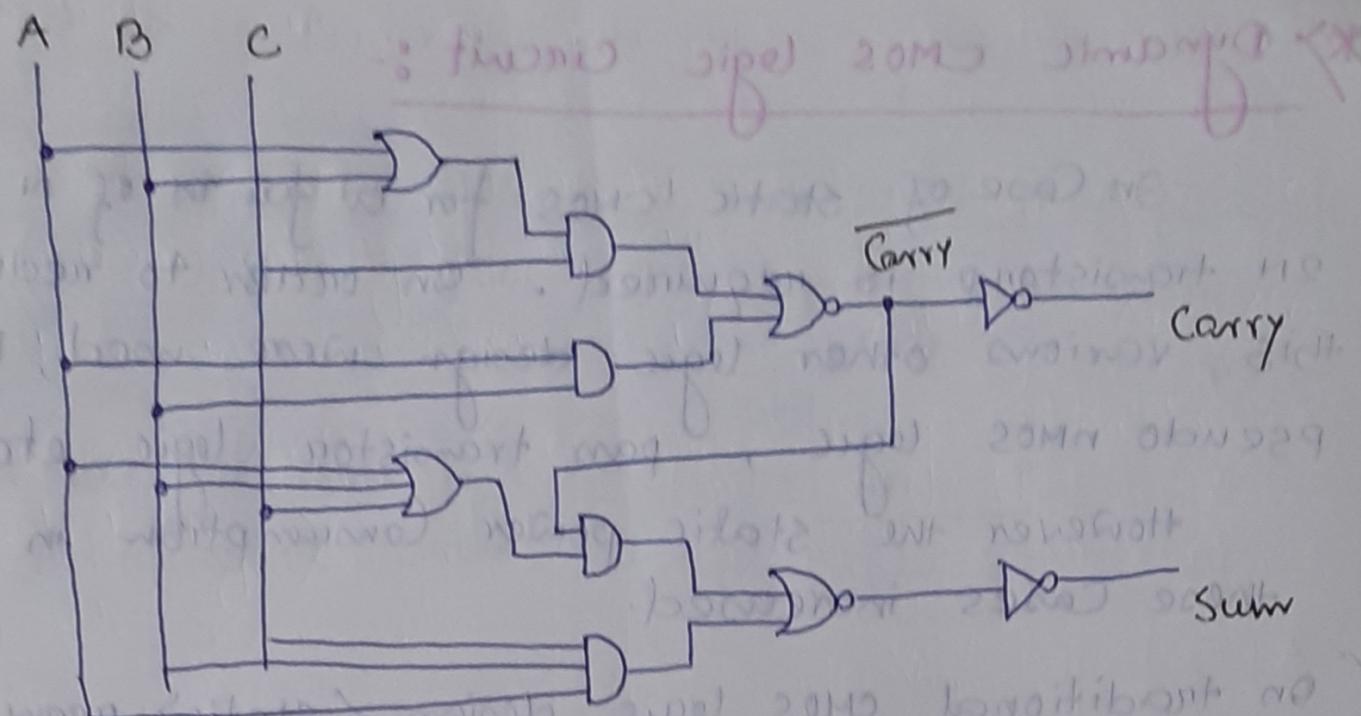
$$= ABC + AB'C' + A'B'C' + A'B'C$$

$$= A \oplus B \oplus C$$

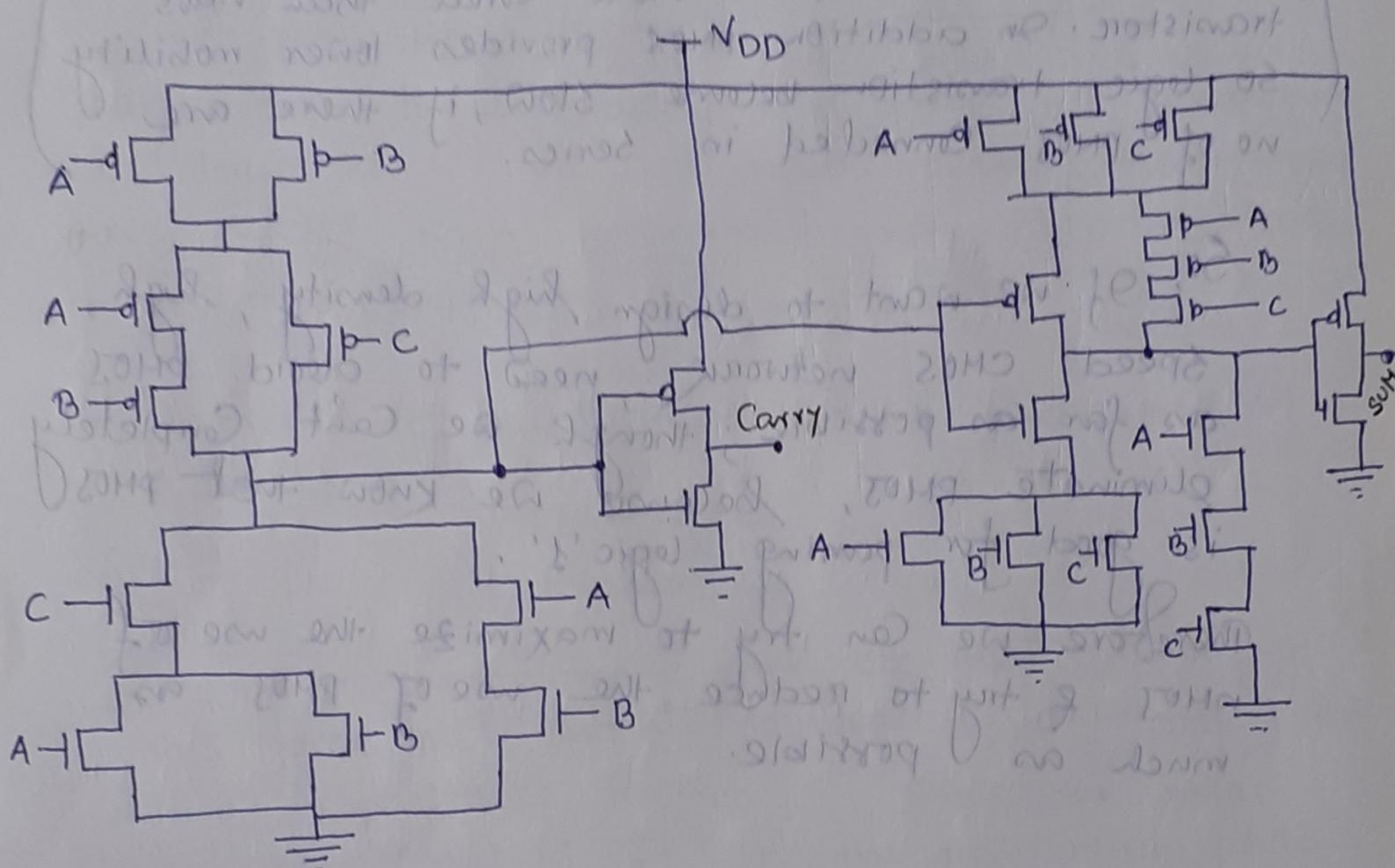
So, we can write,

$$\text{Sum} = ABC + (A+B+C) \cdot \overline{\text{Carry}}$$

{ for
better
implementation }



Logic gate diagram of 1-bit full adder



Total 28 no of transistors (both nMOS & pMOS) is required to implement 1-bit full adder circuit using static CMOS logic.

*> Dynamic CMOS logic circuit :-

In case of static CMOS for a fan in of N , $2N$ transistors is required. In order to reduce this, various other logic design were used, like pseudo nMOS logic, pass transistor logic etc.

However the static power consumption in these cases increased.

{ On traditional CMOS logic design (static) requires same no of PMOS transistors as nMOS transistors. PMOS transistors occupies more area than nMOS transistors. In addition PMOS provides lower mobility so logic transition become slow, if there are no of PMOS connected in series. }

So, If we want to design high density, high speed CMOS network, need to avoid PMOS as far as possible; though we can't completely eliminate PMOS, because we know that PMOS is good for passing logic '1'.

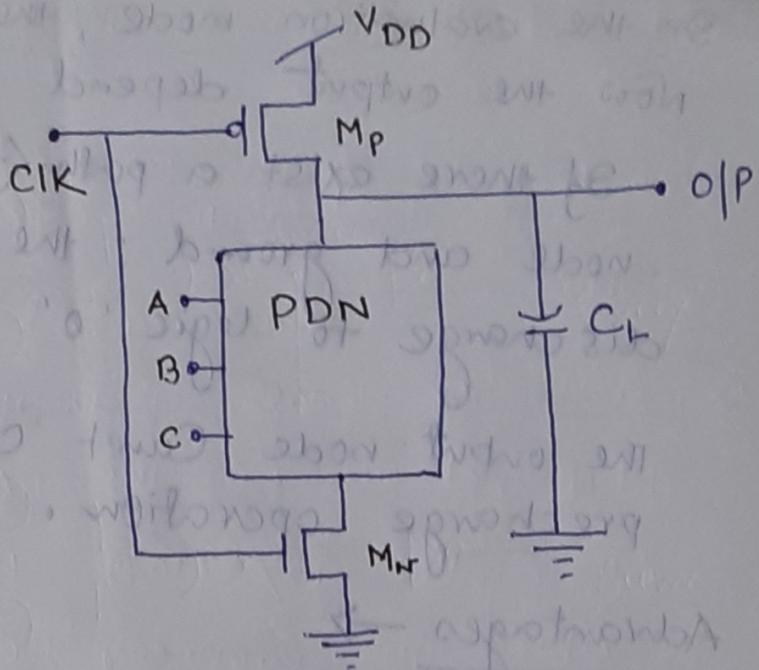
Therefore we can try to maximize the use of nMOS & try to reduce the use of PMOS as much as possible.

∴ (DMOS & other effect) contribute to an 80% reduction in the total power consumption of by using signal, the other side with two

In dynamic CMOS logic design, no

of transistors requires
here are $(N+2)$ as

Compared to $(2N)$ in
the static CMOS circuit.



In Dynamic logic, O/P

logic stored as charge

<with parasitic capacitor>

[Basic structure of Dynamic CMOS logic]

Charges tend to leak away with time. To maintain the logic level, the CLK needs periodic refreshment, thus 'CLK' with minimum frequency is essential for internal operation.

The operation of this circuit can be explained in two modes.

i) precharge

ii) Evaluation

In the precharge mode, the CLK input is at logic '0' this force the O/p to logic '1', charging the load capacitance to V_{DD}. Since the nMOS transistor (M_N) is off, so the pull down path is disabled.

So there is no static consumption, as there is no direct path between supply & ground.

In the evaluation mode, the CK input is at logic 1.

Now the output depend on the PDN block.

If there exist a path (through PDN) between op^p node and ground; the load capacitor (C_L) will discharge to logic '0' else remains at logic 1.

The output node can't charge until the next precharge operation.

Advantages →

i) The no of transistors requires here are $(N+2)$ Compare to $(2N)$ in the static CMOS logic ckt.

ii) Static power loss is negligible.

iii) Large noise margin.

Disadvantages →

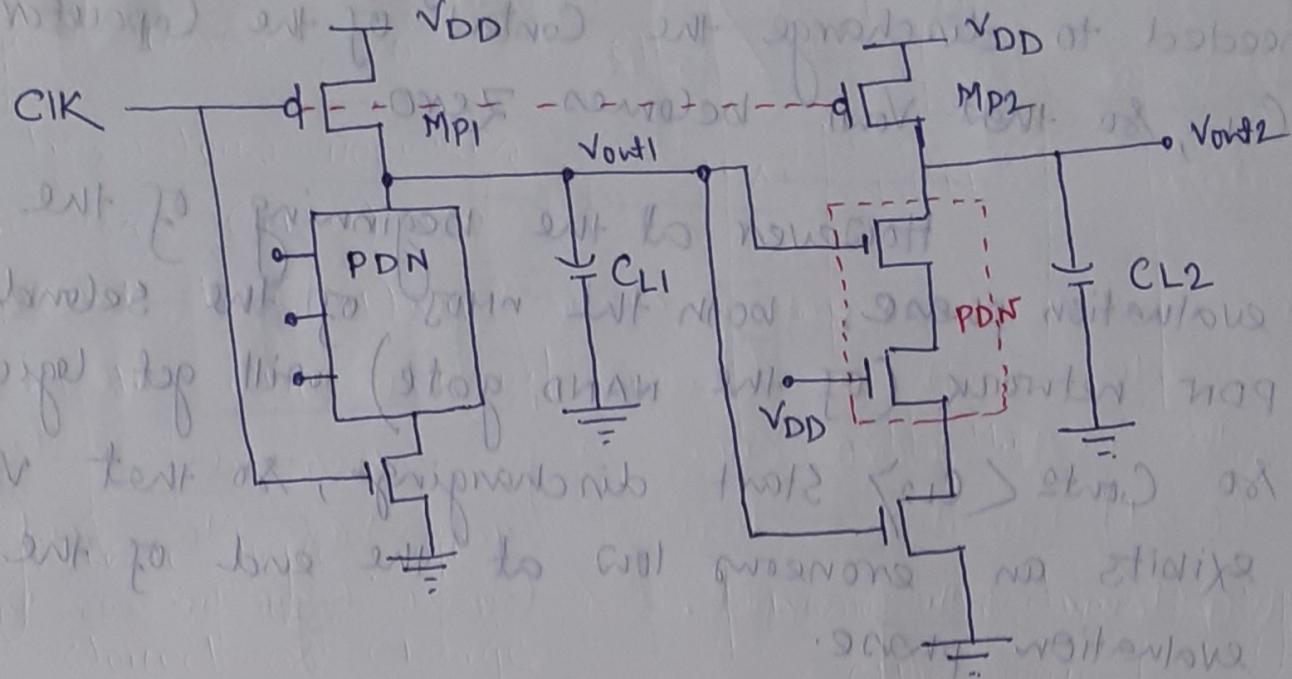
i) The penalty paid in such ckt is that the CK must run everywhere to each such block.

ii) The major problem is that, the output node is at V_{DD} till the end of the precharge node.

Now, if the CK in the block arrives earlier compared to the CK in this block, so PDN in this block takes a longer time to evaluate its output, then the next block will start to evaluate using this erroneous value.

This ii point disadvantages can be eliminated by using Domino CMOS logic design.

* problems in Cascading dynamic logic →



Hence, output of the first dynamic stage is driving the second dynamic stage, which is assumed a NAND gate for simplicity.

During the precharge phase C_{L1} & C_{L2} will be charged by V_{DD} through M_{P1} & M_{P2} respectively. The input variables of the first stage are assumed to be such that V_{out1} will drop to logic '0' during the evaluation phase.

Obviously for the particular input combination applied, the correct output should be $V_{out1} = 0V$ & $V_{out2} = 1V_{DD}$. But, what happens however is something different.

As the evaluation phase begins, C_{L1} begins to discharge, so that V_{out1} drops to its correct logic level after a certain time delay.

The output V_{out1} was high just at the beginning of the evaluation phase. Some finite time is needed to discharge the content of the capacitor C_{L1} , so that V_{out1} becomes zero.

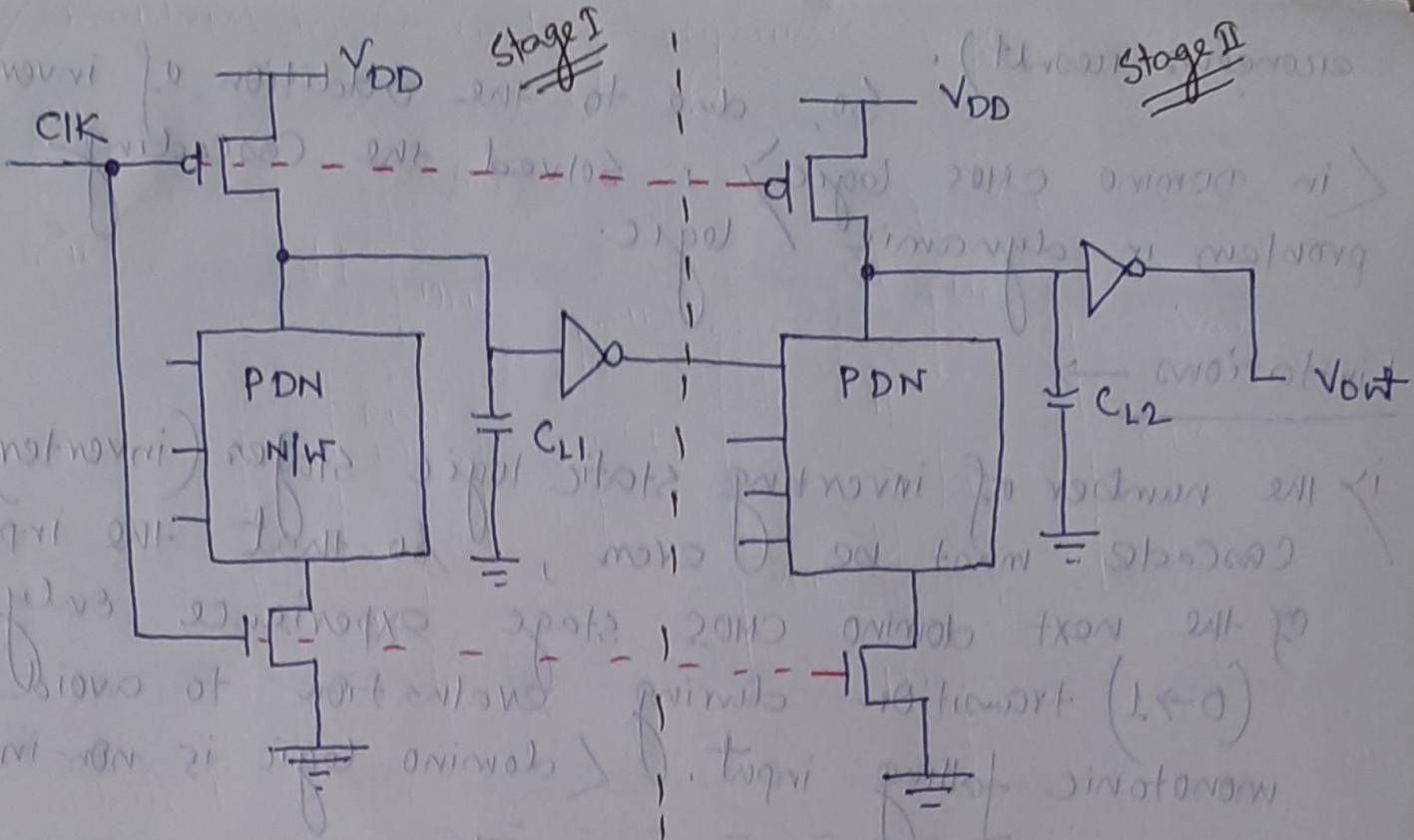
However at the beginning of the evaluation phase, both the NMOS of the second PDN network (of the NAND gate) will get logic '1'; so $C_{out2} < C_{L2}$ start discharging, so that V_{out2} exhibits an erroneous low at the end of the evaluation phase.

Once the stored charge are lost, the recovery or correction is not possible in dynamic logic.

*Domino CMOS logic → Domino CMOS logic is a slightly modified version of Dynamic CMOS logic circuit.

In this case, a static inverter is connected at the output of each dynamic CMOS logic circuits.

The addition of the inverter solves the problem of cascading of dynamic CMOS logic circuits.



Cascaded domino CMOS logic circuit

In the precharge phase ($\text{CLK}=0$), the output of the dynamic CMOS logic circuits are pre-charged to logic high & the output of the static inverter become logic low.

In the evaluation phase ($\text{CLK}=1$), the output of the dynamic CMOS logic circuit can either go to logic low or remain at logic high.

On the Domino CMOS logic, in addition of the CMOS inverter, after pre-charge phase, the output of every inverter become logic '0'. So, during the evaluation phase, the inverter output can make only one transition ($0 \rightarrow 1$).

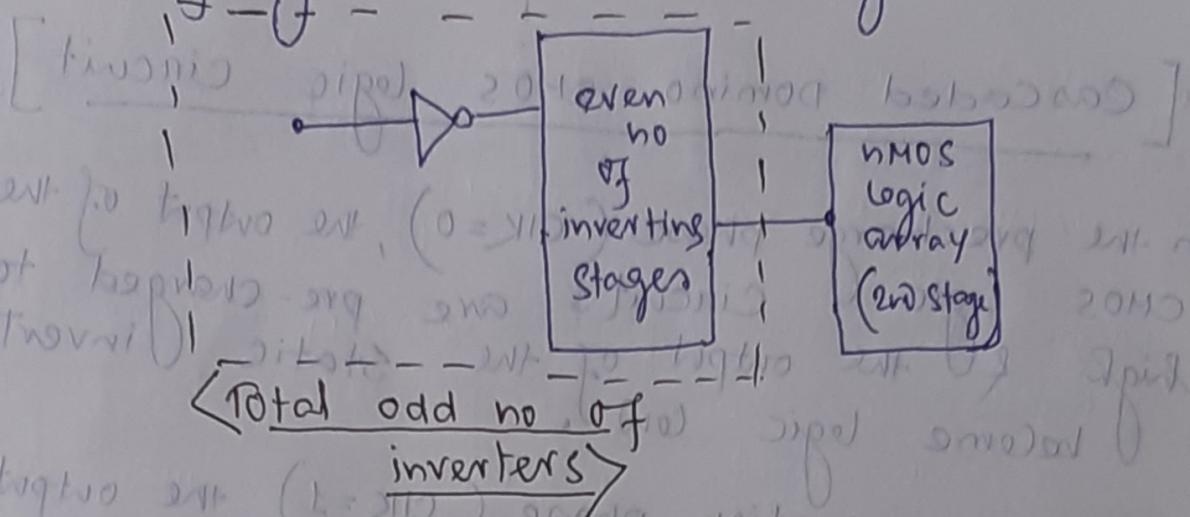
Therefore, in the precharge phase output stage of the second dynamic stage never degrades its logic level (which may produce

erroneous result).

So, due to the addition of inverter in domino CMOS logic, solved the cascading problem in dynamic logic.

Limitations →

- i) The number of inverting static logic stages (inverter) in cascade must be even, so that the inputs of the next domino CMOS stage experience only $(0 \rightarrow 1)$ transition during evaluation to avoid monotonic falling input. *< domino logic is non-inverting>*



- ii) Domino gates are inherently non-inverting. If inversion is required, it must be carried out at the beginning of the domino logic chain.

- iii) Output voltage decay due to leakage effects; To prevent the output node from decaying, it needs to be refreshed periodically.

features →

- i) parasitic capacitances are smaller, so that higher operating speeds are possible.
- ii) operation is free of glitches as each gate

can make only one transition.

- ∴ only non-inverting structures are possible because of the presence of inverting buffer.
iv) charge distribution may be a problem.

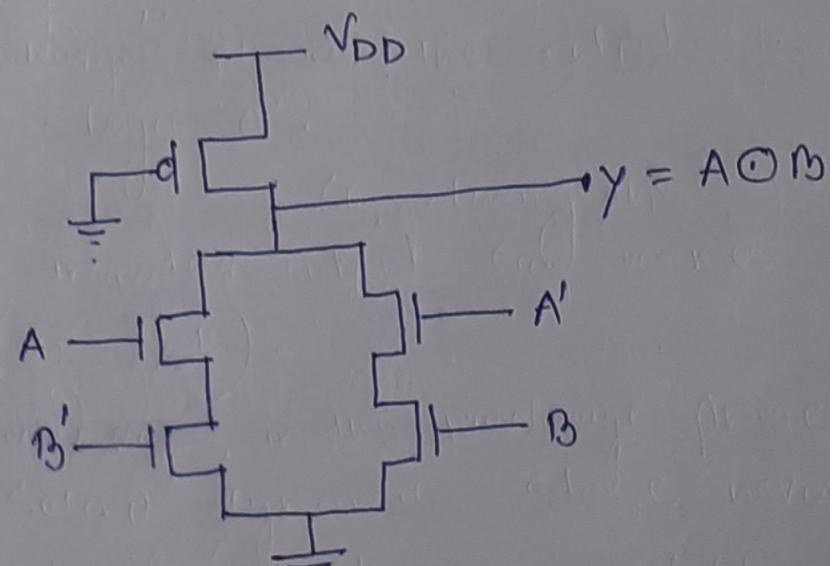
*> PSEUDO n-MOS gates → In static CMOS logic

design, for n-input logic gates, total $2n$ number of transistors are needed. \langle 'n' numbers of PMOS & 'n' numbers of NMOS \rangle . This requires a large area for the implementation of Complex CMOS logic within an high density IC.

In pseudo NMOS logic design ($n+1$) number of transistors are required for implementing n-input logic gate. Here in the load section one PMOS is sufficient to charge up the output node & according to the input variables \langle Combination \rangle output goes low through the PDN.

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

$$y = AB + A'B' \\ = A \oplus B$$



[PSEUDO NMOS REALISATION]