UNIT – IV

Logic Gates and other Complex Gates :–

Cmos static Logic :–

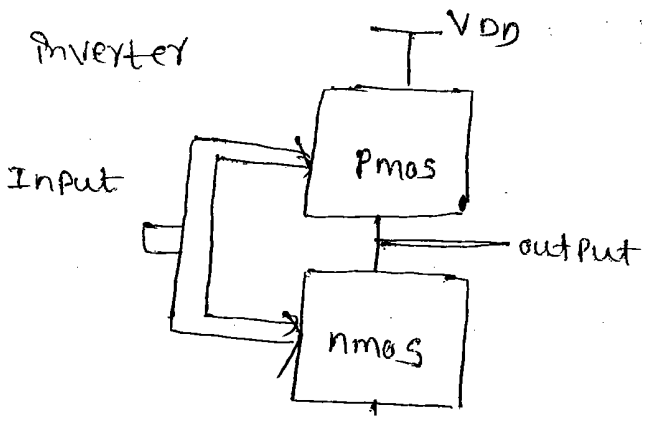
Static, fully complementary cmos gate designs using inverter, NAND and NOR gates can build more complex functions.

→ These cmos gates have good noise margins and low static power dissipation at the cost of more transistors when compared with other cmos logic designs.

→ Cmos gate have 2 tansistor nets (nmos and Pmos) whose topologies are related.

Pmos transistor net is connected between the Powersupply and logic gate output, whereas the nmos transistor topology is connated between the output and ground.

EX: Cmos inverter



The transistor network is related to the Boolean function with a straightforward design procedure:

(1) Derive the nmos transistor topology with following rules:

- Product terms in the Boolean function are implemented with series-connected nmos transistors.
- Sum terms are mapped to nmos transistors connected in parallel.

(2) The pmos transistor network has a dual or complementary topology with respect to the nmos net.

(3) Add an inverter to the output to complete the function if needed. some functions are inherently negated such as NAND, NOR etc, and donot need an inverter at the output state.

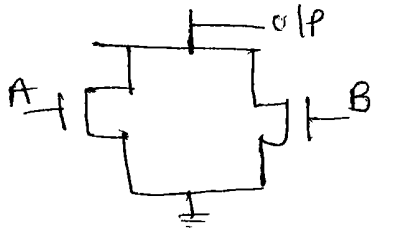
An inverter added to a NAND or NOR function produces the AND and OR function.

Examples which require inverter to fulfil the function:

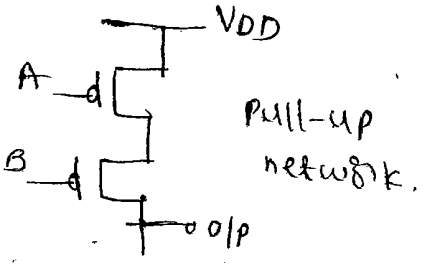
(1) OR: $F=(A+B)$

(1) nmos transistor topology:

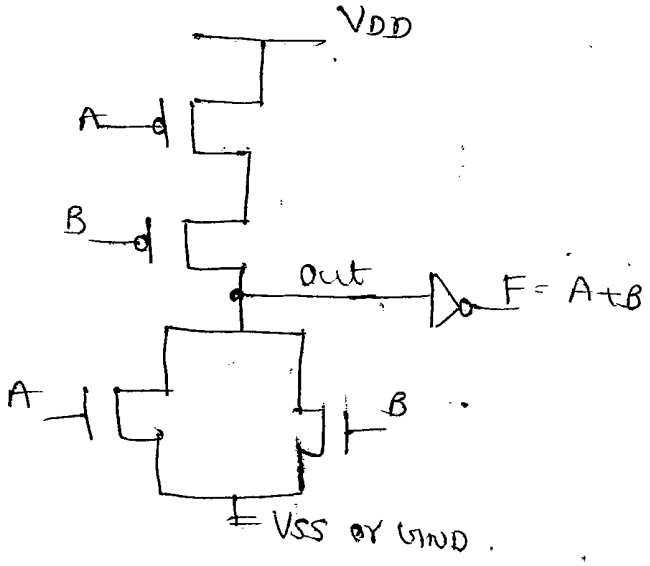
Connect 2 transistors in parallel which indicates NOR function of A | B



(2) Implement Pmos net as a dual topology to nmos net. i.e, Connect 2 Pmos transistors in series.



(3) Finally add an inverter to obtain the function, so that $F = \overline{Out}$

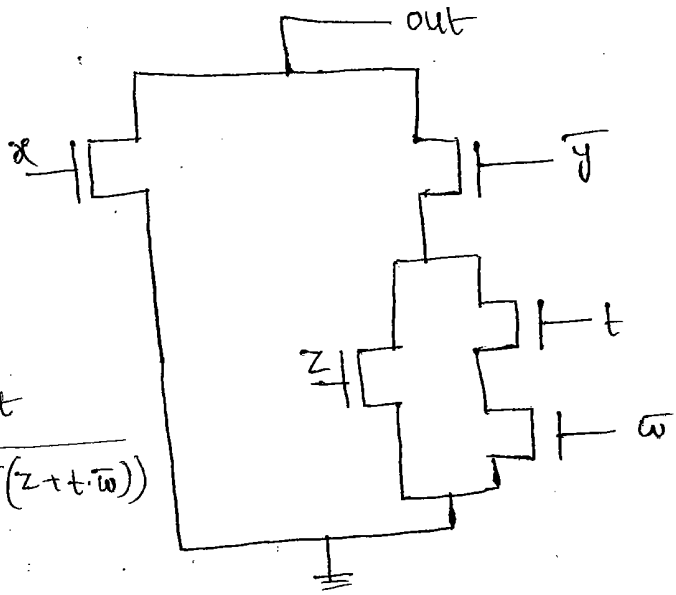


Example 2 :-

Design nmos transistor net for a Boolean function

$F = x + \{\bar{y} \cdot [z + (t \cdot \bar{w})]\}$.

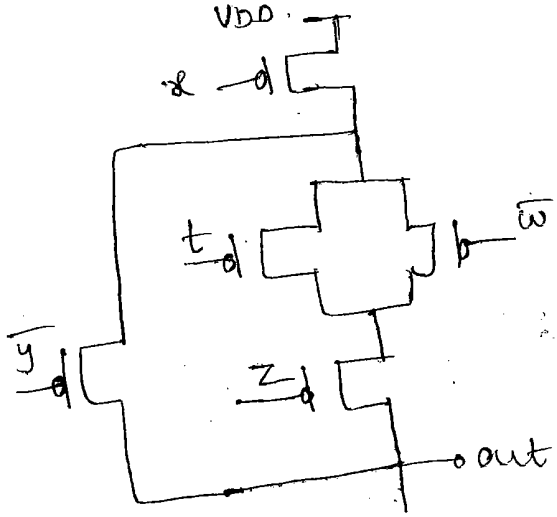
Soln: We design this gate with a top-down approach. The nmos transistor network is connected between the output and ground terminal.



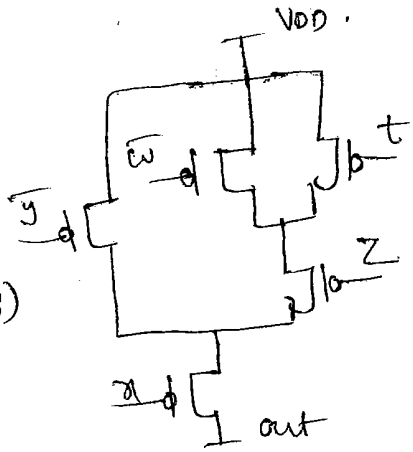
nmos net

for $F = \overline{x + (\bar{y}(z + t \cdot \bar{w}))}$

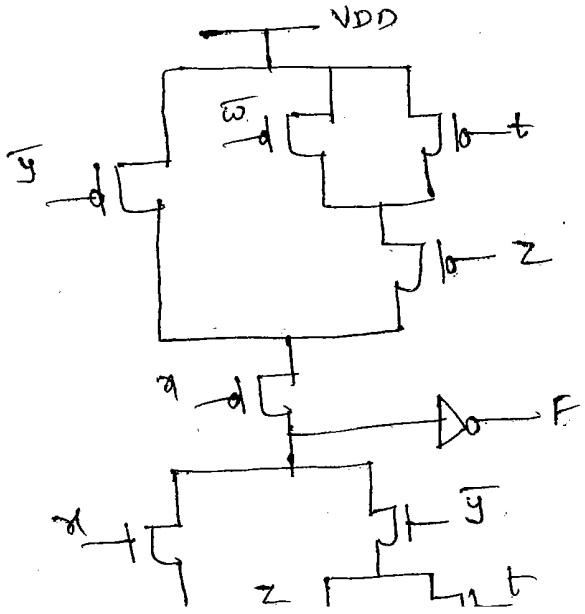
Pmos net



(or)



→ Then connect both pmos net & nmos net, then connect an inverter to its output.



Driving large capacitive loads :-

The problem of driving comparatively capacitive loads arises when signals must be propagated from the chip to off chip destinations.

Generally, off-chip capacitances may be several orders higher than on-chip C_g values.

Ex: C_L denotes off chip load then

$$C_L \geq 10^4 C_g \text{ (typically)}$$

→ Capacitances of this order must be driven through low-resistances, otherwise excessively long delays will occur.

Cascaded inverters as drivers :-

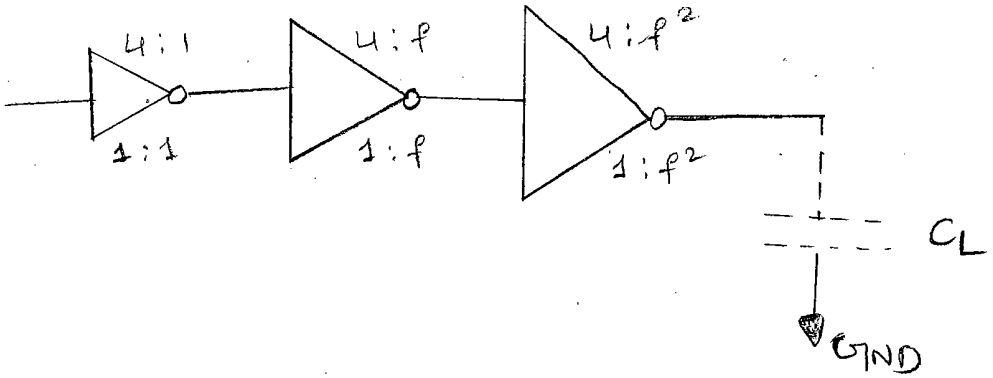
Inverters intended to drive large capacitive loads must therefore present low pull-up & pull-down resistance.

Low resistance values for Z_{pd} and Z_{pu} imply low L:w ratio's.

→ channel must be made very wide to reduce resistance value and in consequence, an inverter to meet this need occupies a larger area.

→ moreover, because of large L:w ratio and since length L cannot be reduced below the minimum feature size, the gate region area L x w becomes significant and a comparatively large capacitance is presented at the input, which in turn slows down the rate of change of voltage which can take place at the input.

Remedy: Use N cascade inverters, each one of which is larger than the preceding stage by a width factor f.



Driving Large capacitive loads

As the width factor increases, the capacitive load presented at the inverter input increases and area occupied increases also. Equally clearly, the rate at which the width increases (ie, value of f) will influence the number N of stages which must be cascaded to drive a particular value of C_L.

With large f, N decreases but delay per stage increases.

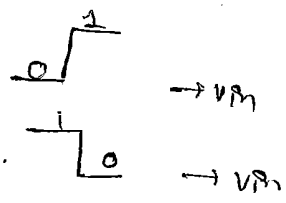
For 4:1 nmos inverters

delay per stage = fT for ΔV_{in}

or = 4fT for ∇V_{in}

ΔV_{in} = indicates logic 0 to 1 transition.

∇V_{in} = " logic 1 to 0 " ΔV_{in}.



∴ Total delay per nmos pair = 5fT.

Ily treatment yields delay per cmos pair = 7fT.

Let $y = \frac{C_L}{\square C_g} = f^N$

so that the choice of f and N are interdependent.

We need to determine the value of f which will minimize delay for a given value of y & from the definition of y

$$\ln(y) = N \ln(f)$$

i.e, $N = \frac{\ln(y)}{\ln(f)}$

Thus for N even

total delay = $\frac{N}{2} 5fT = 2.5 N f T$ (nmos)

or = $\frac{N}{2} 7fT = 3.5 NfT$ (cmos)

In all cases,

delay $\propto NfT = \frac{\ln(y)}{\cdot} fT$

total delay minimized if f assumes the value e (base of natural logarithms).

i.e, each stage should be $\simeq$ 2.7* times wider than its predecessor.

assuming that $f = e$, we have

$N = \ln(y)$ & overall delay t_d.

N even: $t_d = 2.5 \varepsilon_{NT}$ (nmos)

$t_d = 3.5 \varepsilon_{NT}$ (cmos)

N odd: $t_d = [2.5(N-1)+1] \varepsilon_T$ (nmos)

$t_d = [3.5(N-1)+2] \varepsilon_T$ (cmos)

} for ΔV_{in}

or $t_d = [2.5(N-1)+4] \varepsilon_T$ (nmos)

$t_d = [3.5(N-1)+5] \varepsilon_T$ (cmos)

} for ∇V_{in}

Super buffers :-

The asymmetry of the conventional inverter is clearly undesirable, and gives rise to significant delay problem when an inverter is used to drive more significant capacitive loady.

→ A common approach used in nmos technology to alleviate the effect is to make use of supper buffers.

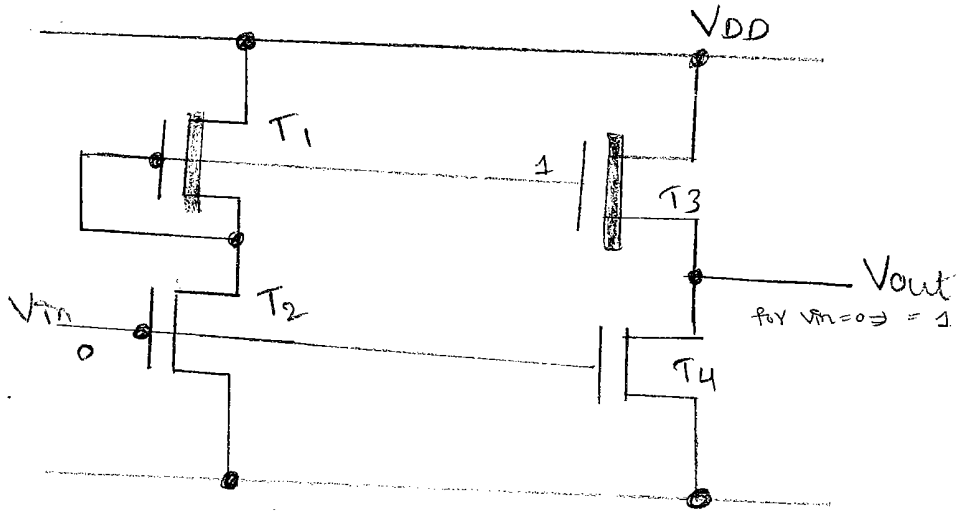


fig: Inverting type nmos super buffer

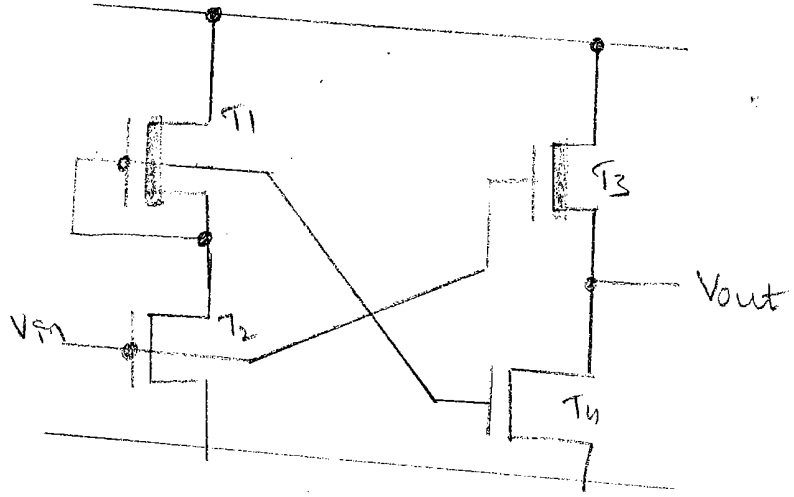


fig: Non-Inverting type nmos super buffer

Inverting type;-

Considering a positive going logic transition Vin at the input, (logic1), it will be seen that the inverter formed by T_1 and T_2 is turned on & thus the gate of T_3 is pulled down toward 0V with a small delay. Thus T_3 is cut off while T_4 (the gate of which is also connected

to Vin) is turned on and output is pulled down quickly.

When Vin drops to 0V,

then the gate of T_3 is allowed to rise quickly to V_{DD}. Thus as T_4 is also turned off by Vin, T_3 is made to conduct with V_{DD} on its gate i.e, with twice the average Voltage that would apply if the gate was tied to the source as in the conventional nmos inverter.

Now, since $I_{ds} \propto V_{gs}$ then doubling the effective V_{gs} will increase the current and thus reduce the delay in charging any capacitance on the o/p, so that mile symmetrical transitions are achieved. $I = C \cdot \frac{dv}{dt}$

Non-Inverting:-

Vin = 0V.

then T_2 open & T_1 Conducts & it will turn on T_4 with V_{DD}. Then T_3 is nonconducting and T_4 is connected to o/p. Hence we get o/p = 0V through T_4.

When Vin = logic 1.

then T_2 ON

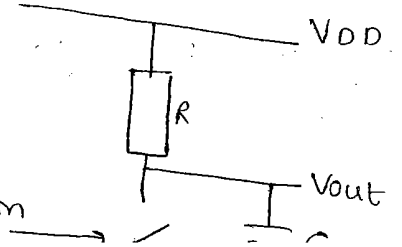
Bicmos drivers :-

The availabilty of bipolar transistors in Bicmos technology presents the possibility of using bipolar transistor drivers of the output stage of inverter and logic gate circuits.

Bipolar transistors have an exponential dependence of the output current I_C on the input base to emitter voltage V_{be}. This means that the device can be operated with much smaller input voltage swing than Mos transistors and still switch relatively large currents.

Important thing to consider is the possible effect of temperature T on the required input voltage V_{be}. Although V_{be} is logarithmically dependent on base width W_B, doping level N_A, & mobility μ_n & collector current I_C it is only linearly dependent on T.

The switching performance of a transistor driving a capacitive load may be visualized initially from the simple model.



The time necessary to change the output voltage by an amount that is equal to the input change is given by $\Delta t = \frac{C_L}{g_m}$

g_m = device transconductance

→ The time Δt necessary to change the output voltage Vout by an amount equal to the input voltage Vin is given by $\Delta t = \frac{C_L}{g_m}$

g_m — transconductance of bipolar transistor.

→ Transconductance of bipolar transistor relatively high, hence the value of Δt is small.

A more exacting appraisal of the bipolar transistor delay reveals that it comprises 2 main components.

(1) T_{in} — initial time necessary to charge the base-emitter junction of the npn transistor. Typically, for Bicmos Transistor-based driver we are considering T_{in} is in the region of 2ns.

→ Similarly, consideration of a cmos transistor driver in the same Bicmos Technology would reveal a figure of 1ns for T_{in}, this being the time taken to charge the input gate capacitance.

⇒ Another significant parameter contributing to delay is the collector resistance (R_C) of bipolar Transistor.

→ High value of R_C will mean a long propagation delay through a transistor when charging a capacitive load.

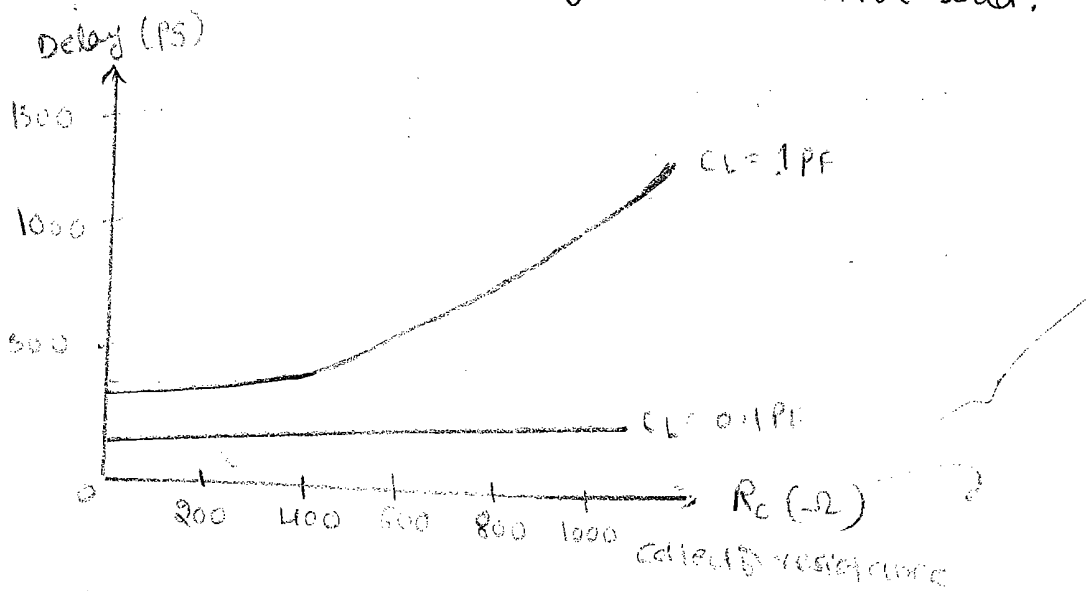


Fig: Gate delay as a function of collector resistance

→ The reason for including the buried subcollector region in the Bicmos process is to keep R_C as low as possible.

→ Bicmos fabrication processes produce reasonably good bipolar transistors – high g_m, high β, high h_{fe} & Low R_C – without compromising or overelaborating the basic cmos process.

→ The availability of bipolar transistors in logic gate and driver/buffer design provides a great deal of scope and freedom for VLSI designer.

(2) T_L – the time taken to charge the output load capacitance C_L & it will be noted that this time is less for the bipolar driver by a factor of h_{fe}, where h_{fe} is bipolar transistor gain.

Combined Effect of T_{in} & T_L :-

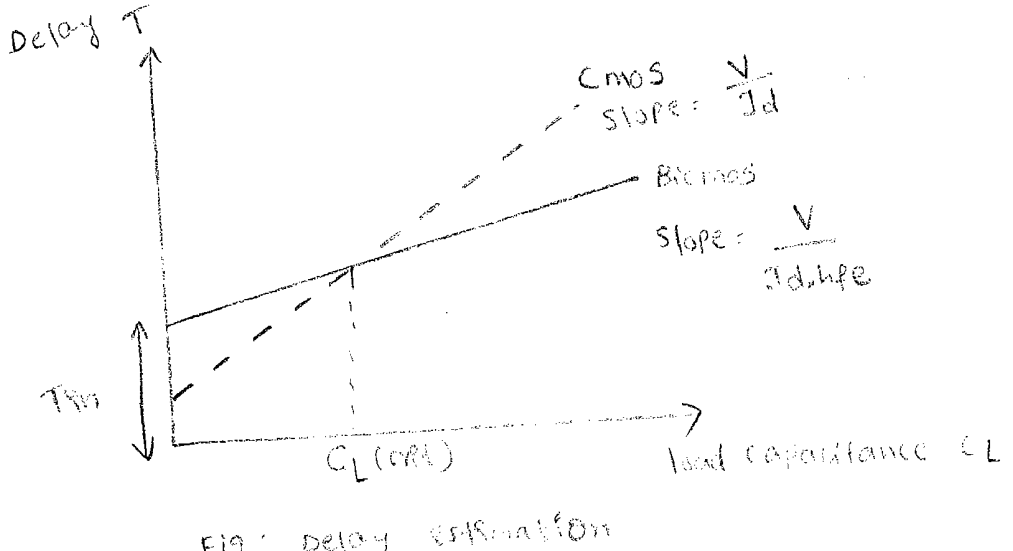


Fig: Delay estimation

Delay of Bicmos inverter can be described by

$$T = T_{in} + \left(\frac{V}{I_d}\right)\left(\frac{1}{h_{fe}}\right) C_L$$

where T_{in} – time to charge up base/emitter junction

h_{fe} = Transistor current gain (CE)

Hence, delay for Bicmos inverter is reduced by a factor of h_{fe} compared with a cmos inverter.

$C_L(crit)$: The value of load capacitance below which the Bicmos driver is slower than a comparable cmos driver.

Propagation Delays :-

Cascaded pass Transistors :-

A degree of freedom offered by Mos technology is the use of pass transistors as series / parallel switches in logic arrays. Quite frequently, therefore, logic signals must pass through a number of pass transistors in series. A chain of four such transistors is shown in fig. 5.29 (a)

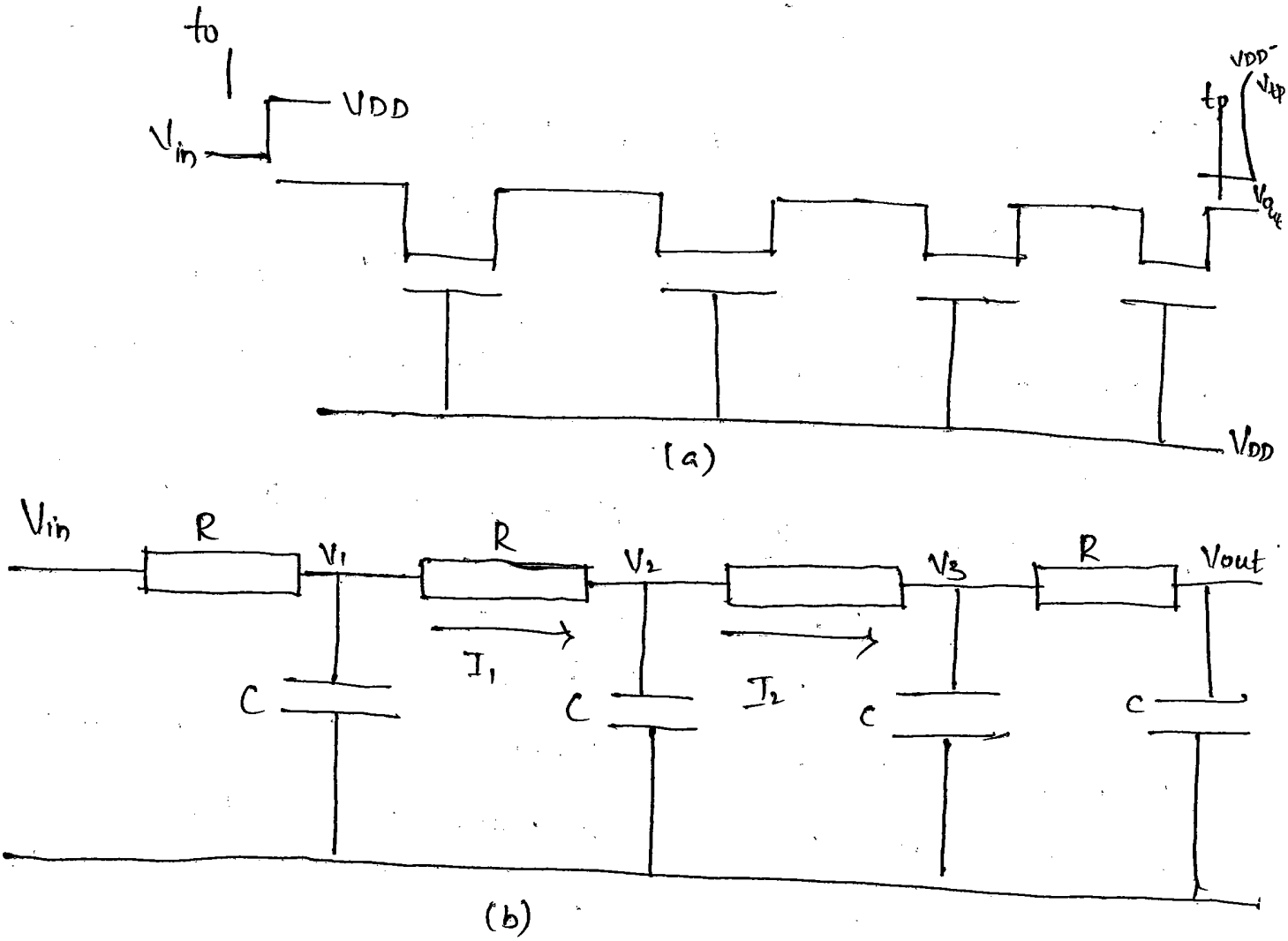


fig : 5.29 (a) and (b) propagation delays in pass transistor chain

in which all gates [illegible]

which would be the case for a signal to be propagated to the output. The circuit thus formed may be modelled as in fig. 5.29 (b) and it is then possible to evaluate the delay through the network.

$$\frac{c\,dv_2}{dt} = (I_1 - I_2) = \frac{(V_1 - V_2) - (V_2 - V_3)}{R}$$

In the limit as the number of sections in such a network becames large, this expression reduces to

$$Rc\frac{dv}{dt} = \frac{d^2v}{dx^2}$$

Where,

R = Resistance per unit length

c = Capacitance per unit length

x = distance along network from input

The propagation time t_p for a signal to propagate a distance x is such that $t_p \propto x^2$

The analysis can be simplified if all Rs and Cs are lumped together, then

$$R_{total} = nrR_s$$

$$C_{total} = nc\square c_g.$$

Where r gives the relative resistance per section in terms of R_s & c gives the relative capacitance per section in terms of $\square c_g$. Then, it may be shown that over all delay t_d for n sections - is given by

$$t_d = n^2rc(\tau).$$

Thus, the over all delay increases rapidly as n increases & in practice no more than four pass transistors should be normally connected in [illegible] however this number can be exceeded if a buffer is [illegible]

5.4.2 Design of long polysilicon wires :

Long polysilicon wires also contribute distributed series R and C as was the case for cascaded pass transistors and in consequence signal propagation is slowed down. This would also be the case for wires in diffusion where the value of C may be quite high, and for this reason the designer is discouraged from running signals in diffusion except over very short distances.

for long polysilicon runs, the use of buffers is recommended In general, the use of buffers to drive long polysilicon runs the use of buffers is recommended. In general, the use of buffers to drive long polysilicon runs have two desirable effects. first the signal propagation is speeded up & , Second there is a reduction is Sensitivity to noise.

The reason why noise may be a problem with slowly rising signals may be deduced by Considering fig. 5.30. In the diagram, the slow rise-time of the signal at the input of the inverter (to which the signal emerging from the long polysilicon line is connected) means that the input voltage spends a relatively long time in the vicinity of V_{inv} so that small disturbances due to noise will switch the inverter state between '0' and '1' as shown at the output point.

Thus it is essential that the long polysilicon wires be driven by suitable buffers to guard against the effects of noise and to stop up the rise time of propagated signal

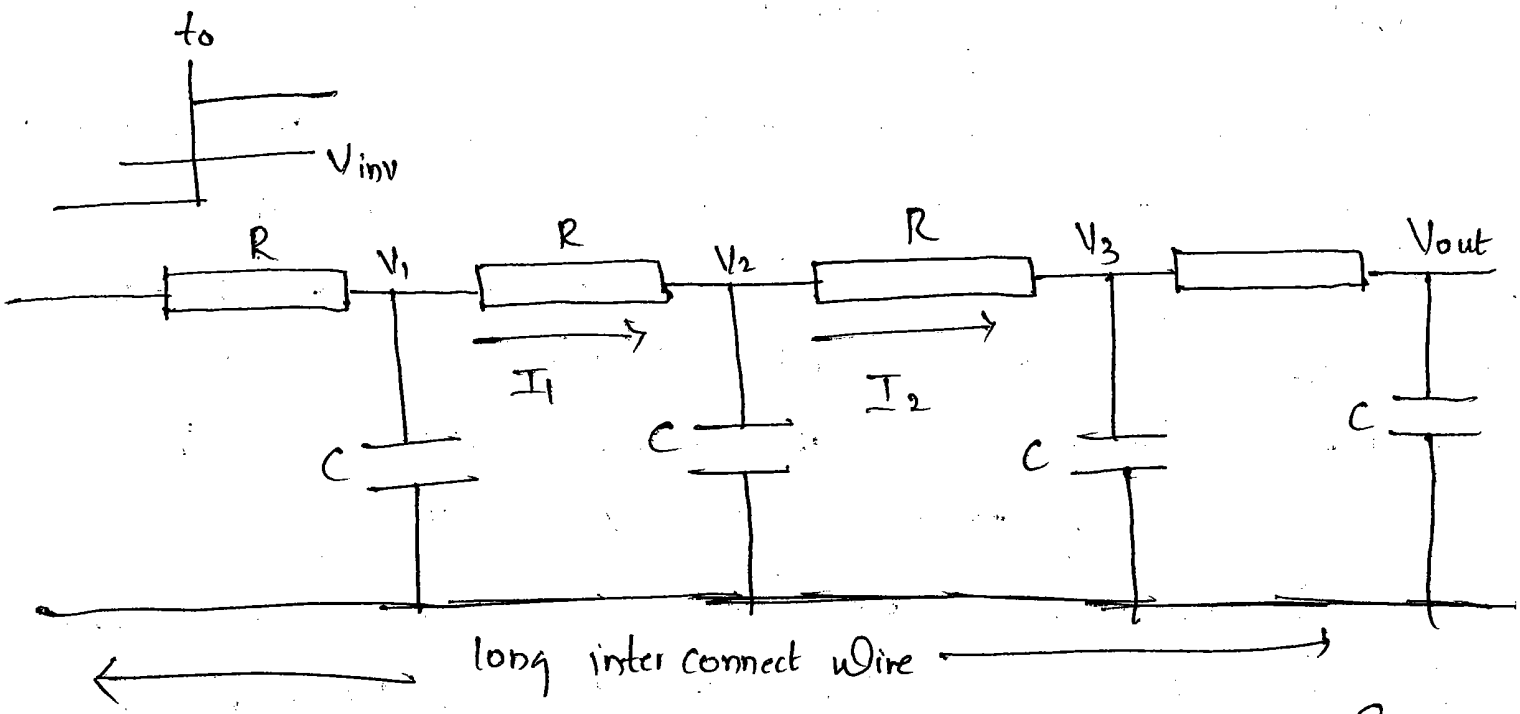


Fig: 5-30 Possible effects of delays in Polysilicon wires

Wiring capacitances:-

There are significant sources of capacitance which contribute to the overall wiring capacitance. Three such sources are

1. Fringing fields
2. Interlayer capacitances.
3. Peripheral capacitance.

Fringing fields:-

Capacitance due to fringing field effects can be a major component of the overall capacitance of interconnect wires.

→ For fine line metallization, the value of fringing field capacitance (C_{ff}) can be of the same order of that of the area capacitance.

$$C_{ff} = \epsilon_{sio_2}\epsilon_0 l\left[\frac{\pi}{\ln\left\{1+\frac{2d}{t}\left(1+\sqrt{\left(1+\frac{t}{d}\right)}\right)\right\}} - \frac{t}{4d}\right]$$

l – wire length

t – thickness of wire

d – wire to substrate seperation.

Then, total wire capacitance, $C_w = C_{area} + C_{ff}$

Interlayer capacitances:-

→ Obviously, the parallel plate effects are present between one layer and another.

→ For example, some thought on the matter will confirm the fact that, for a given area,

metal to polysilicon capacitance > metal to substrate capacitance.

→ The reason for not taking such effects into account for simple calculations is that the effects occur only when layers cross or when one layer underlies another, & therefore interlayer capacitance is highly dependent on Layout.

→ However for regular structures it is readily calculated & contributes significantly to the accuracy of circuit modeling and delay calculation.

Peripheral capacitance:-

→ The source and drain n-diffusion regions form junctions with the p-substrate or p-well at well-defined and uniform depths.

Similarly, for P-diffusion regions in n-substrate or n-wells.

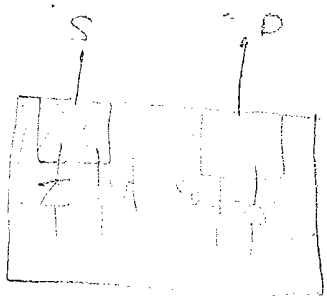
→ For diffusion regions, each diode thus formed has associated with it a perpheral capacitance in pF (picofarady) per unit length.

→ This can be considerably greater than the area capacitance of the diffusion region to substrate.

→ The smaller the source or drain area, the greater becomes the relative value of the peripheral capacitance.

In order to calculate the total diffusion capacitance we must add the contributions of area and peripheral components,

$$C_{total} = C_{area} + C_{periph.}$$



Typical values for diffusion capacitance:

Diffusion capacitance	Typical values		
	5 lim	2 lim	1.2 lim
Area c (C_{area})	1.0×10^{-4} pF/lim^2	1.75×10^{-4} pF/lim^2	3.75×10^{-4} pF/lim^2
Periphery capacitance (C_{periph})	8.0×10^{-4} pF/lim^2	negligible	negligible

Switch Logic :-

To build switches from MOS Transistors one way is "Transmission Gate", built from parallel n-type and p-type transistors.

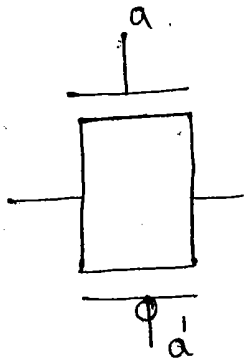


Fig: Complementary Transmission gate

→ This switch is built from both types of Transistors so that it transmits logic 0 and 1 from drain to source equally well.

→ when we put a V_{DD} or V_{SS} at the drain, we get V_{DD} or V_{SS} at the source.

→ But it requires 2 transistors and their associated tubs; equally damning, it requires both true and complement forms of the gate signal.

An alternative to build switches from MOS Transistors is the "n-type switch" — a solitary n-type transistor.

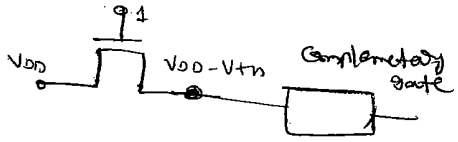
→ It requires only one transistor and one gate signal,

→ but it is not as forgiving electrically.

→ It transmits a logic 0 well, but when V_{DD} is applied to the drain, the voltage at the source is $V_{DD} - V_{th}$.

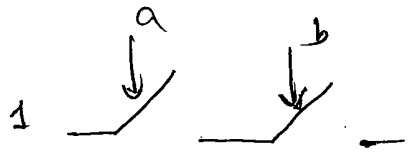
→ When switch logic drives gate logic, n-type switches can cause electrical problems.

→ An n-type switch driving a complementary gate causes the complementary gate to run slower when switch input is 1. Since the n-type pulldown current is weaker when a lower gate voltage is applied, the complementary gate's pulldown will not suck current off the output capacitance as fast.

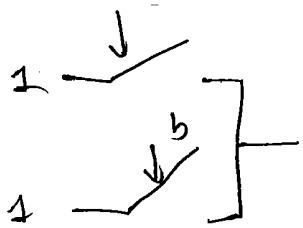


→ A pseudo-nmos is driven by n-type switch, disaster may occur. A pseudo-nmos gate's ratioed transistors depend on logic 0 and 1 inputs to occur within a prescribed voltage range.

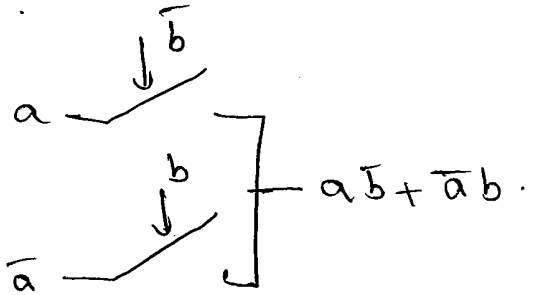
→ If the n-type switch doesn't turn on the pseudo-nmos pulldown strongly enough, the pulldown may not divert enough current from the pull-up to force the output to a logic 0, even if we wait forever.



Pseudo-AND



Pseudo - OR



switch n/w with non-constant source inputs.

Several important alternative CMOS gate topologies. Each has important uses in chip design. But it is important to remember that they all have their limitations and caveats. particular care must be taken when mixing logic gates designed with different circuit topologies to ensure that one's output meets the requirments of the next's inputs

i) Pseudo-nMOS Logic:

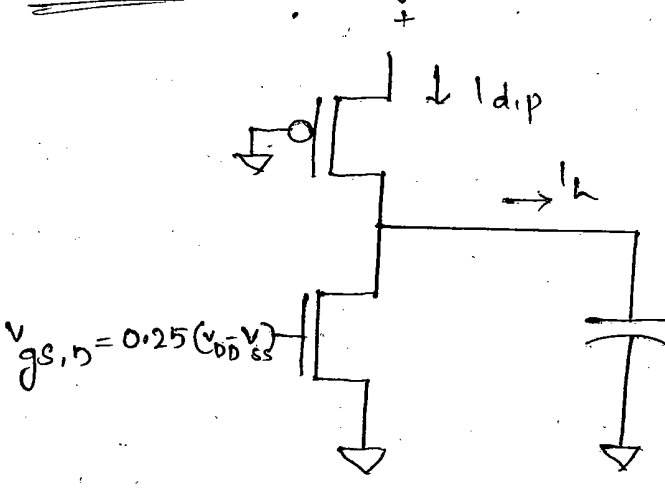


Fig: Pseudo nmos NOR gate

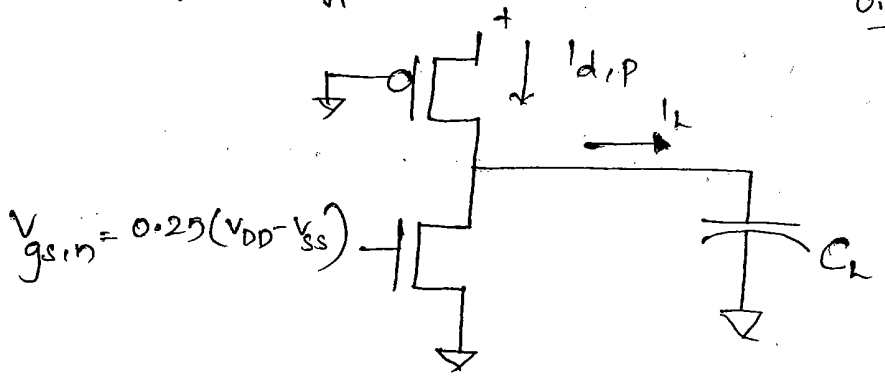
The simplest non-standard gate topology is pseudo-nMOS, so called because it mimics the design of an nMOS logic gate. The pulldocon network of the gate is the same as for a fully complementary gate. The pullup network is replaced by a single p-type transistor whose gate is connected to V_{ss}, leaving the transistor permanently on. The p-type transistor is used as a resistor. When the gate's inputs are ab=00, both n-type transistores are off and the p-type

transistor pulls the gate's output up to V_{DD}. When the input is 1, both the p-type and n-type transistor are on and both are fighting to determine the gate's output voltage.

We need to determine the realationship between the W/L ratio of the pullup & and the pulldowns which provide reasonable output voltage for the gate. For simplicity, assume that only one of the pulldown transistors is on; then the gate circuit's output voltage depends on the ratio of the effective resistance of the pullup and the operating pulldown. The high output voltage of the gate is V_{DD}, but the output low voltage V_{OL} will be some voltage above V_{SS}. The chosen V_{OL} must be low enough to activate the next logic gate in the chain. For pseudo-nMOS gate which feed static or pseudo-nMOS gate, a value of $V_{OL} = 0.15(V_{DD} - V_{SS})$ is a reasonable value, though others could be chosen. To find the transistor sizes which give reasonable output voltages, we must consider the simultaneous operation of the pullup and pulldown. When the gate's output has just switched to a logic 0, the n-type pulldown is in saturation with $V_{gs,n} = V_{in}$. The p-type pullup is in it's linear region: it's $V_{gs,p} = V_{DD} - V_{SS}$ and it's $V_{ds,p} = V_{out} - (V_{DD} - V_{SS})$. We need to find V_{out} in terms of the W/Ls of the pullup and pulldown. To solve this problem, we set the currents through the saturated pulldown and the linear pullup to be equal.

$I_{D,n} = \frac{1}{2} k'_n \frac{W_n}{L_n} [2(V_{gs,n} - V_{tn}) V_{ds,n} - V_{ds,n}^2] = \frac{1}{2} k'_p \frac{W_p}{L_p} [2(V_{gs,p} - V_{tp}) V_{ds,p} - V_{ds,p}^2]$ — (1)

The pulldown network must exhibit this effective resistance in the worst case combination of inputs. Therefore, if the network contains series pulldowns, they must be made larger to provide the required effective resistance.



Tech 0.5 Lm: Sub, $V_{dd} = 3.3V$,

$V_{gs,n} = V_{DD} - V_{SS}$ in (1)

we find that

$$\frac{W_p/L_p}{W_n/L_n} \cong 3.9$$

As shown in figure so long as the pulldown drain current is significantely less than the pullup drain current, there will be enough current to charge the output capacitance and bring the gate output to the desired level.

The ratio of the pullup & pulldown sizes also ensures that the times for $0 \to 1$ & $1 \to 0$ transitions are asymmetric. Since the pullup transistor has about three times the effective resistance of the pulldown, the $0 \to 1$ transition occures much more slowely then the $1 - 0$ transition and dominates the gate's delay. The long pullup time makes the pseudo-nMOS gate slower than the static complementary gate.

The main advantage of pseudo-nMOS gate is

the small size of the pullup network, both in terms of number of devices and wiring complexity. The pullup network of a static complementary gate can be large for a complex function. The input signals do not have to be routed to the pullup, as in a static complementary gate. The pseudo-nMOS gate is used for circuits where the size and wiring complexity of the pullup network are major concerns but speed and power are less important.

ii) DCVS logic (Differential Cascode Voltage Switch logic)

→ DCVS logic is a static logic family that has a very different structure.

→ It uses a latch structure for the pullup which both eliminates static power consumption and provides true and complement outputs

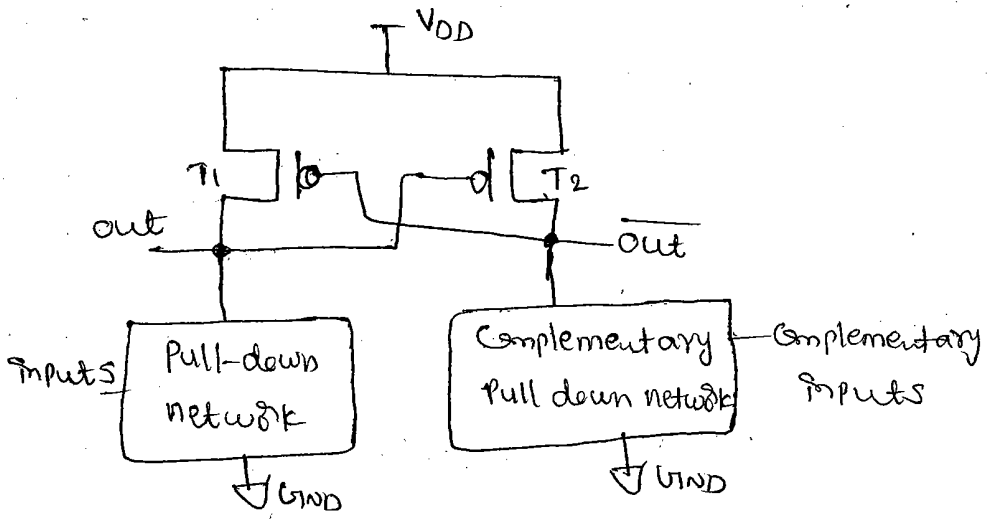
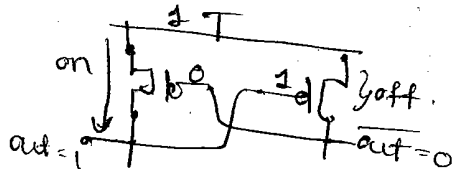


Fig: Structure of a DCVS gate.

→ There are 2 pull-down networks which are duals of each other.

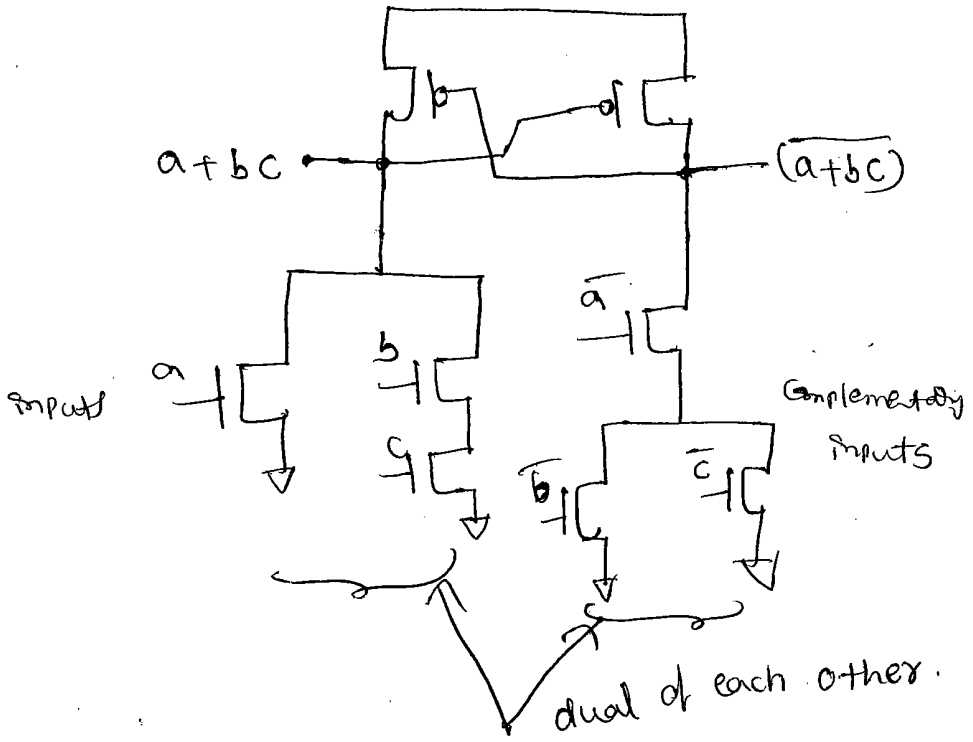
→ Each pull-down network has a single p-type pull-up, but the pull-ups are cross coupled. Exactly one of the pulldown n/w will create a path to ground when the gate's inputs change, causing the output nodes to switch to the required values.

→ The cross-coupling of the pull-ups helps to speedup the transition, Ex, If the complementary n/w forms a path to ground, the $\overline{out}$ goes toward Vss, which turns on the true output's pullup (T_1), raising the True output, which in turn lowers the gate voltage on the complementary output's pullup (T_2).



→ This gate consumes no DC power, since neither side of gate will ever have both its pullup and pull down network at once.

DCVSL gate, which computes $(a+bc)$ on one output & $\overline{(a+bc)} = \bar{a}\bar{b}+\bar{a}\bar{c}$ on its other output.



$= \overline{a+bc}$

$= \overline{(a+b)(a+c)}$

(demorgan's law $\overline{ab} = \bar{a}+\bar{b}$)

$= \overline{a+b} + \overline{a+c}$

$= (\bar{a}\bar{b}) + (\bar{a}\bar{c})$

$= \bar{a}(\bar{b}+\bar{c})$

$\bar{a}$ scaled

iii) Domino Logic :-

Precharged circuits offer both low area and higher speed than static complementary gates. Precharged gates introduce functional complexity because they must be operated in 2 distinct phases, requiring introduction of a clock signal.

The canonical precharged logic gate circuit is the domino circuit.

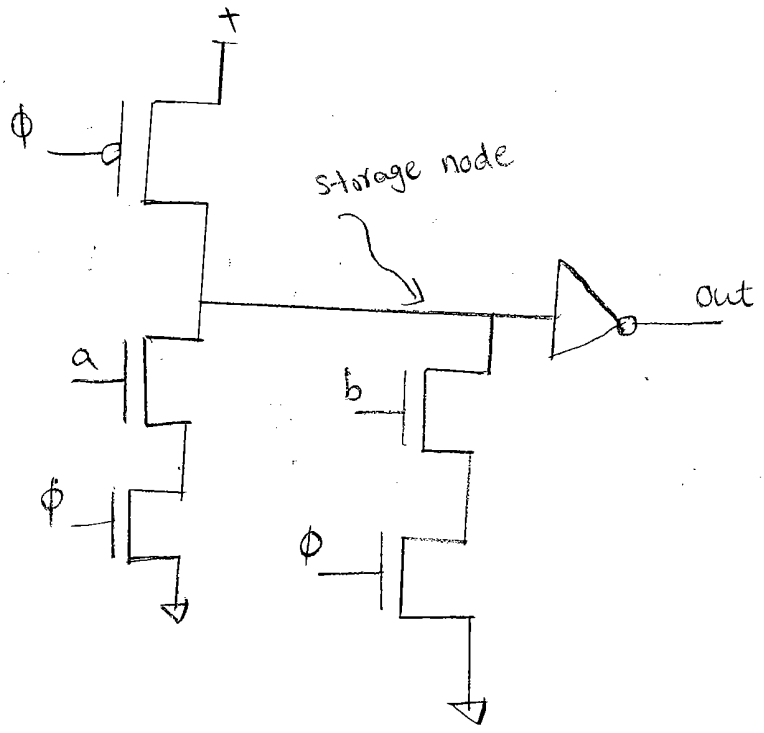


fig: domino OR gate

→ The gate works in 2 phases, first to precharge the storage node, then to selectively discharge it. The 2 phases are controlled by the clock signal ϕ.

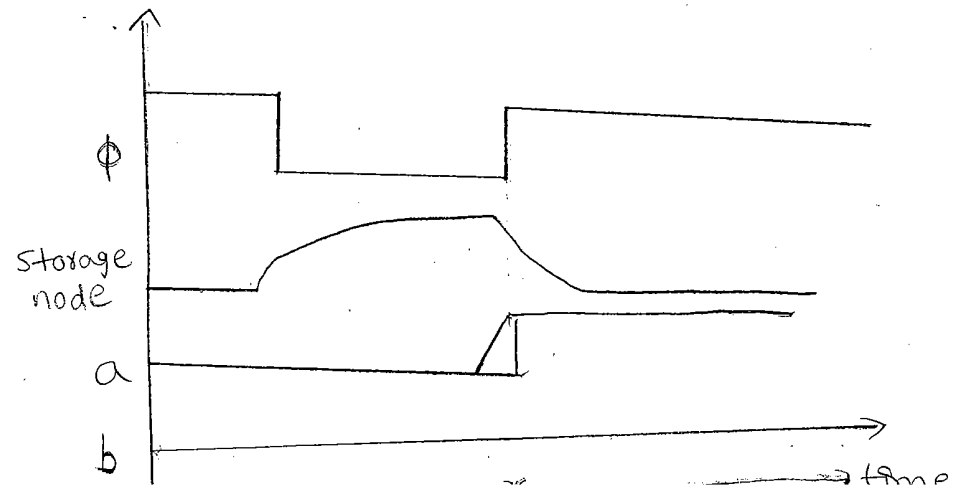
Precharge: When ϕ goes low, the p-type transistor starts charging the precharge capacitance. The pull down transistors controlled by the clock keep that precharge node from being drained. The length of the $\phi=0$ phase is adjusted to ensure that the storage node is charged to a solid logic 1.

Evaluate: when ϕ goes high, precharging stops (P-mos off) and the evaluation phase begins (n type Pulldown ON).

→ The logic inputs a and b can now assume their desired value of 0 or 1.

→ The input signals must monotonically rise – if an input goes from 0 to 1 and back to 0.

→ If the inputs create a conducting path through the pulldown network, the precharge capacitance is discharged forcing its value to 0 and the gate's o/p is 1 (through the inverter).



→ If neither a nor b is 1, then the storage node would be left charged at logic 1 and the gate's output would be 0.

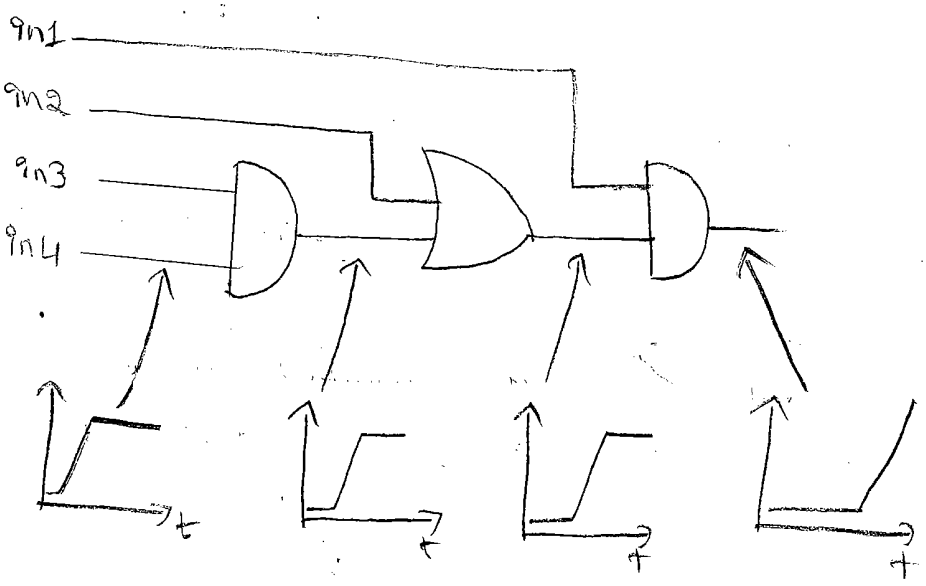


Fig: successive evaluations in a domino logic network

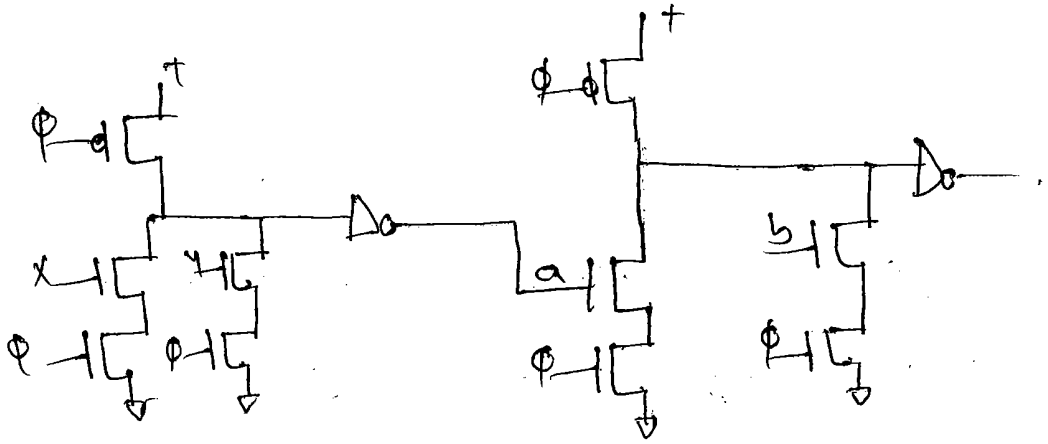
The figure illustrates the phenomeon which gave the domino gate its name. Since each gate is precharged to a low output level before evaluation, the change at the primary inputs ripple through the domino network from one end to another.

→ signals at the far end of the network change last, with each change to a gate output causing a change to the next output. This sequential evaluation resembles a string of falling dominos.

Need of Inverter at the Output of the domino gate:

Reason 1: logical operation and circuit behaviour.

If output of one domino gate is fed into an input of another domino gate, then during precharge phase, if the inverter were not present, the intermediate signal would rise to 1, violating the requirement that all inputs to the second gate be '0' during precharging.



Reason 2: To increase the reliability of the gate.