CHAPTER-3 UNIT-2

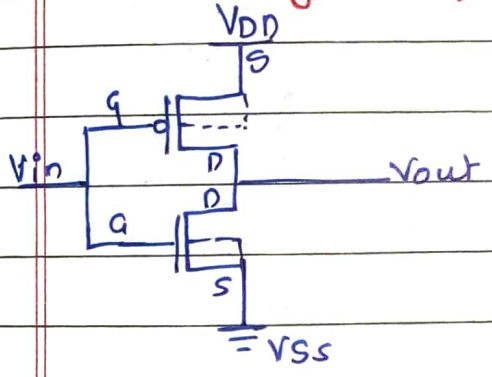
# DESIGN OF CMOS LOGIC GATES

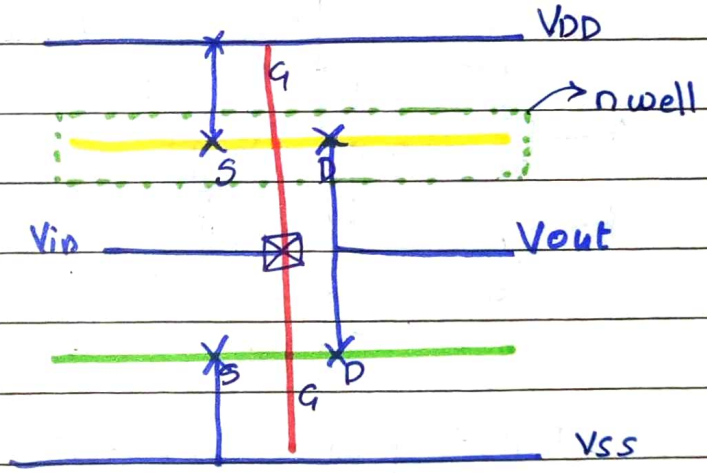
## STICK DIAGRAM:

The graphical representation of CMOS circuit

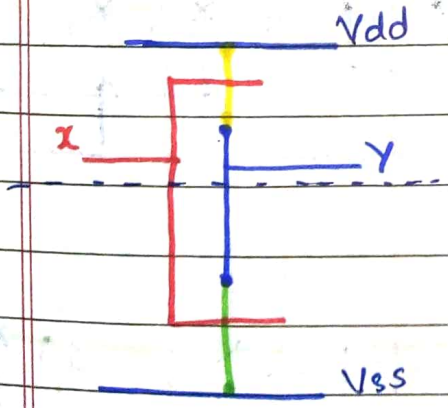
* poly: represented by red colour
* Metal1: " " blue colour
* ndiff : represented by green colour
* pdiff : represented by yellow (brown) colour
* Via contact → ⊠

Ex: Stick diagram for CMOS inverter ckt




or



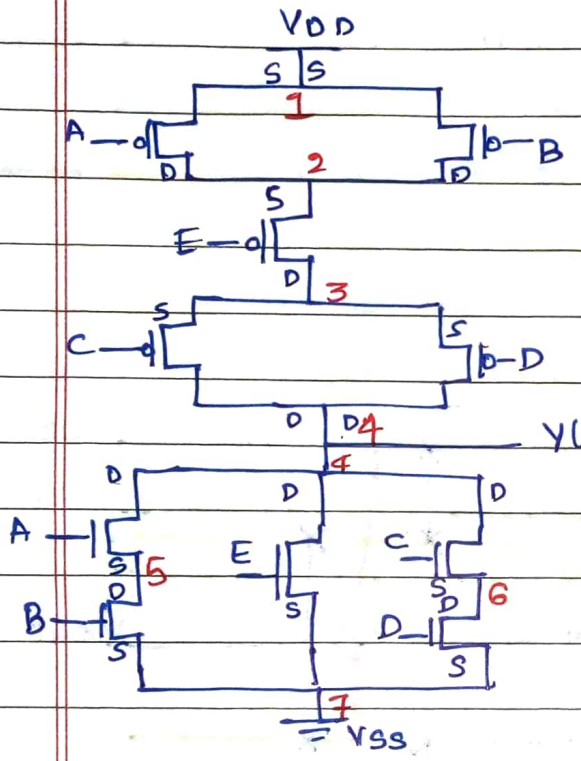

when poly cuts diffusion, we will have a transistor (PMOS & NMOS)

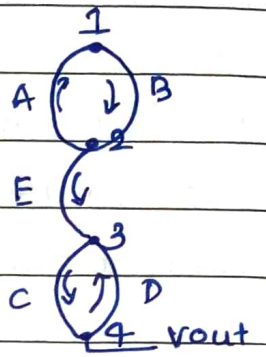
## Euler Path :-

↳ Ordering of gate

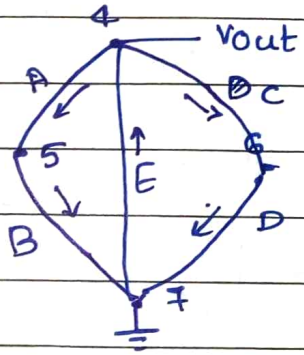
↳ To minimize the area, for routing we need ordering of gate.

Ex $Y = \overline{AB + E + CD}$



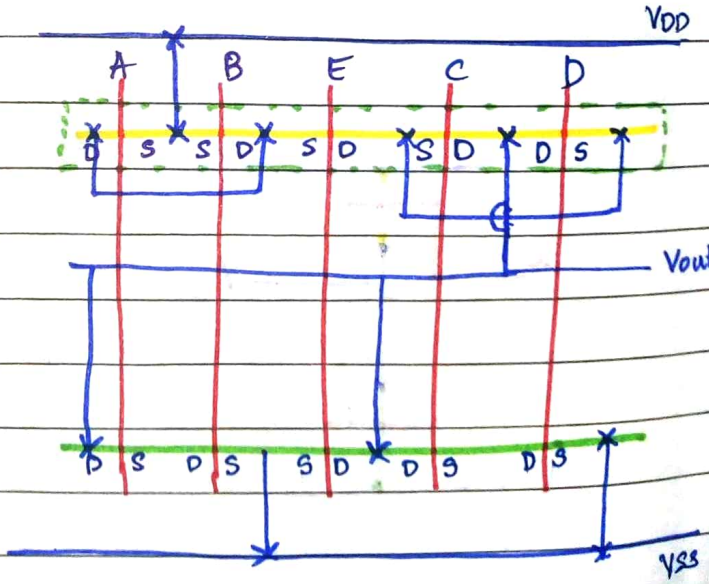




→ p-tree




n-tree

Euler's path : A B E C D



2λ : minimum

## Layout Design Rules:

* Layout design rules describe how small features can be and how closely they can be reliably packed in a particular manufacturing process.

* Lambda - based rules are necessarily conservative because they round up dimensions to an integer multiple of lambda.

* Designers often describe a process by its feature size.

* Feature size refers to min transistor length, so lambda is half the feature size.

Length of gate $L = 180nm$ for 180nm technology.

$\lambda = \frac{L}{2}$ $\therefore$ 90nm, for 180nm technology

$= 0.09\mu m$.

* Layout design rule:

→ Metal and diffusion have min width and spacing of 4λ.

→ Contacts are 2λ x 2λ & must be surrounded by 1λ on the layers above & below.

→ Polysilicon uses a width of 2λ

→ " overlaps diffusion by 2λ where a transistor is desired and has a spacing of 1λ away

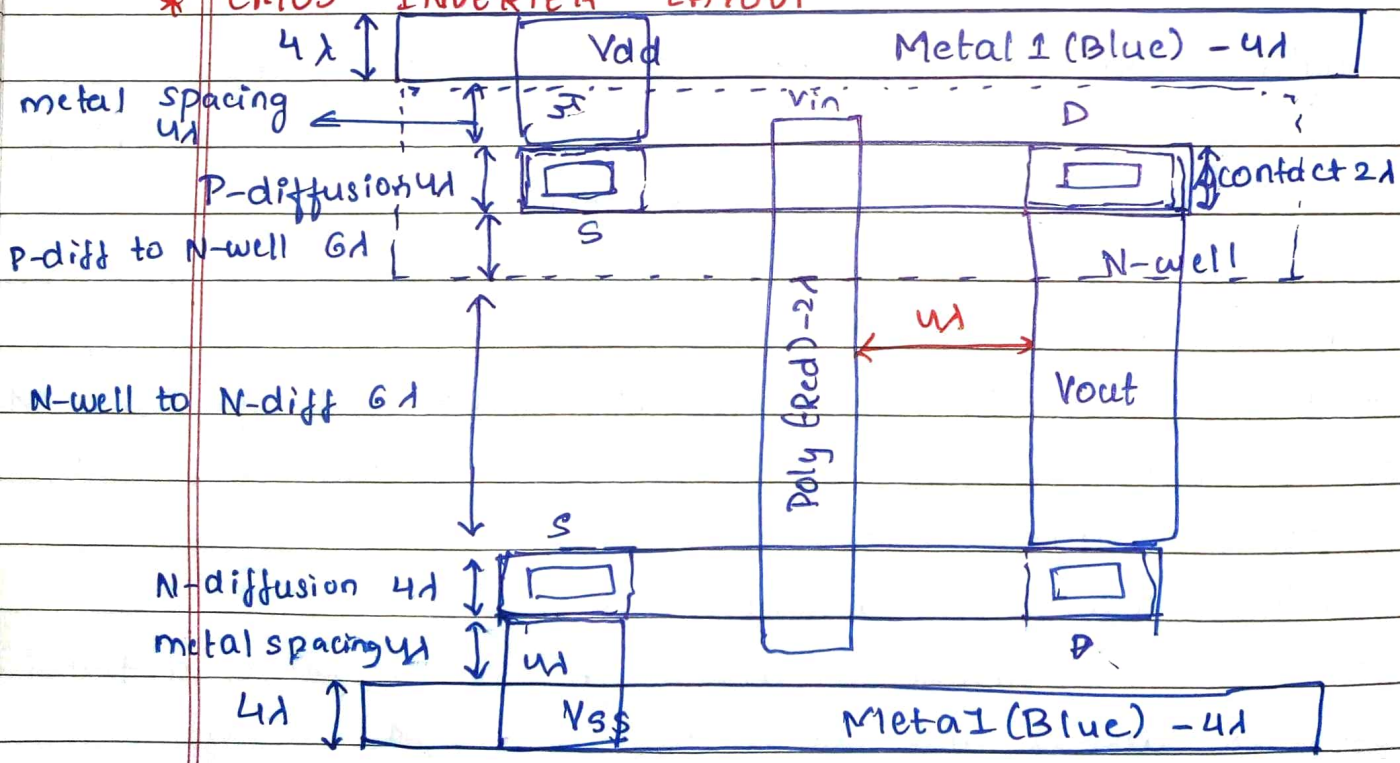
no transistor is desired.

→ Polysilicon & contacts have a spacing of 2λ from other polysilicon or contacts.

→ N-well surrounds pMOS transistors by 6λ & avoids nMOS transistors by 6λ.

* Metal1 : 4λ & metal to metal spacing is also 4λ
* Diffusion (P & n) : diffusion is of 4λ & spacing is also 4λ
* polysilicon: width is 2λ & space bt 2 polysilicon is 3λ
* Metal1 - diffusion contact → 1λ
* Metal1 - polysilicon contact → 2λ
* Metal1 - Metal2 via → 2λ

* CMOS INVERTER LAYOUT



∴ height of Inverter : $4\lambda + 4\lambda + 4\lambda + 6\lambda + 6\lambda + 4\lambda + 4\lambda + 4\lambda$

$= 36\lambda$

width of Inverter = $12\lambda = (4+4+4)\lambda$.

∴ $36\lambda \times 12\lambda$

## DRC, LVS and Circuit Extraction.

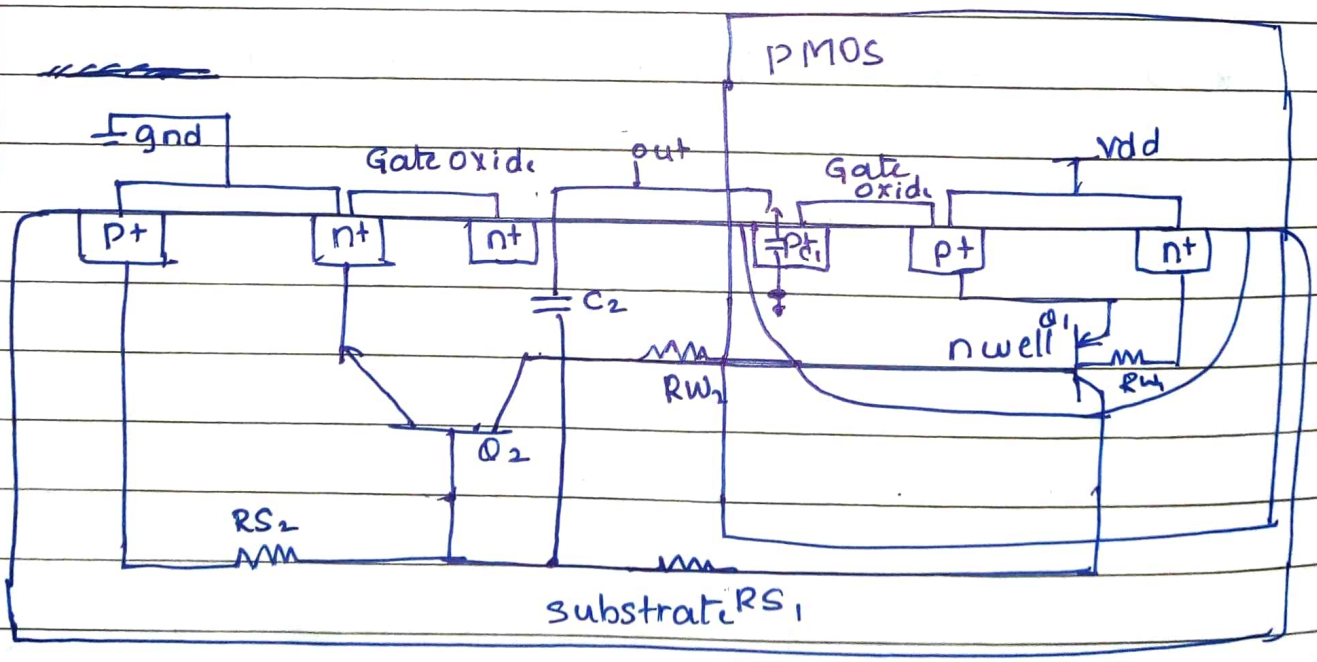
* DRC [Design Rule check]
  - ↳ It will check minimum spacing of the layout.
  - ↳ Spacing of contacts, diffusion, Poly, Metal etc.

* LVS (layout Vs Schematic]
  - ↳ It check for layout vs Schematic errors.

* Circuit Extraction / AV Extracted View (RC)

# LATCH UP PROCESS & PREVENTION

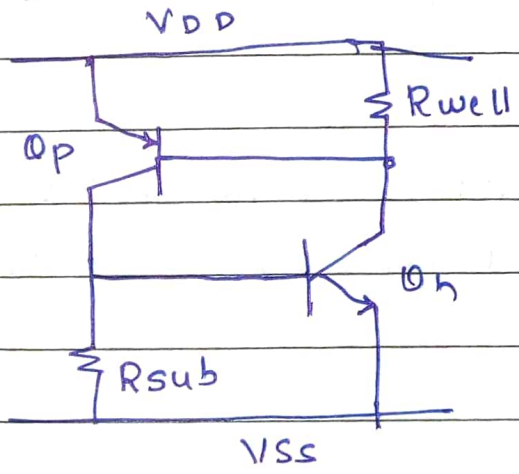
* Latch up is a condition where a low impedence path is created between the supply pin and the ground pin.
* This condition is created by a trigger but once activated, the low impedance path remains even if the trigger has been removed so the inverter output will not change upon changing the input.
* due to interaction between parasitic pnp & npn transistor
* The structure formed by these resembles a silicon controlled rectifier (SCR, usually known as thyristor, a PNPN device used in power electronic).
* These form a +ve feedback loop, Shrt ckt the power rail and ground rail, & eventually causes excessive current, & can even permanantly damage device.




↳ latchup formation in CMOS inverter.

* The shunting resistors Rwell and Rsub represent the effective resistance from the well tap to the PNP base and the substrate tap to the NPN base.

* For the circuit to latch up, several conditns must be met
① The transistor current gain product of Qn & Qp must be greater than 1 such that the structure will remain latched.
② Both emitter - base junctions of Qn & Qp must be forward biased to initiate & sustain latchup.
③ The power supply must be able to sustain the supply current drawn while latched (the holding current) & supply voltage (the holding voltage).




→ Steps to prevent latch up:

1] Reduce the beta of either or both parasitic devices. In practice this can be achieved by increasing the spacing between the devices, which increases the width

of the lateral device. However, such increased spacing reduces packing density.

② Increase well & substrate doping concentration to reduce Rwell & Rsub. For example, using retrograde doped wells.

③ Provide alternative (or better) collectors of the minority carriers. for example the use of guard rings around devices.

post test:

① Which layer is used for power and signal lines?
a) n-diffusion b) polysilicon c) metal d) p-diffusion
→ c) metal

② Which design method occupies or uses less area?
a) layer rule b) lambda rules c) micro rules d) source rule
→ c) micron rules

③ Design rules does not specify
a) Separations b) colours c) extensions d) line widths
→ b) colours

④ stick diagrams does not gives the position of placement of the element.

(5) The minimum spacing between two n-well is 8.5 μm

(6) Spacing bt & diffusion layers → 3λ.

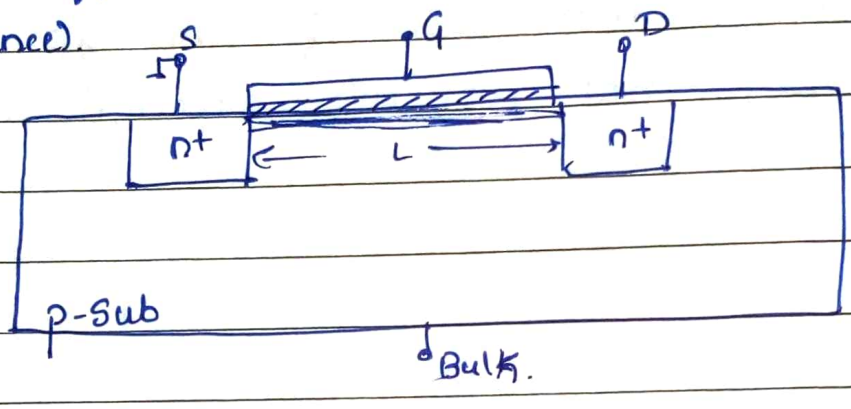
(7) Width of n-diff & p-diffusion layer should be 2λ.

(8) When two or more cuts of same type cross @ or touch each other, that represents electrical contact.

# Chapter 4: Designing Combinational Logic Networks.

## MOS Capacitance Models:

→ Normally we will have two capacitances, one gate capacitance and a diffusion capacitance (may be drain or source capacitance).




* We have capacitance between gate and polysilicon (p-substr - ckt)
  Let us say source is grounded.
* Now when gate voltage is 0, the nmos will be in cutoff. & there will be no channel
  ∴ capacitance will be bt gate and bulk & we call this capacitance as $C_G = C_{ox}(WL)$
  $L$ = length of channel, $W$ = width of device
  ∴ $C_G = C_{ox}(WL) = C_0$ → cutoff

= area of gate

* Now when we increase gate voltage, transistor will be in triode region, channel is uniform and form second plate
  Now capacitance is bt ⓐgate and source

and ② gate and drain.

We will not have gate to bulk capacitance now.

$\therefore C_g = C_{GS} + C_{GD}$

$= \frac{1}{2} C_{ox}(WL) + \frac{1}{2} C_{ox}(WL)$

$C_g \quad C_{GS} + C_{GD} = C_{ol_2} \quad (\because C_{GB} = 0)$

→ Now when transistor goes into saturation.

$V_{GS} > V_t$, $V_{DS} \geq V_{GS} - V_t$ during this, pinch off happens @ drain

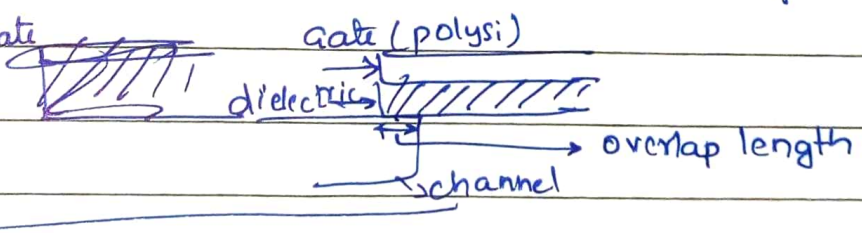
pinch off means channel becomes non existance @ drain.

that means there is no capacitance bt gate & drain. There is only capacitance between gate & source

$\therefore C_{GS} \approx \frac{2}{3} C_o \approx \frac{2}{3} C_{ox}(WL)$. $[C_{GD} = 0, C_{gb} = 0)$

~~Overall~~

After all these capacitance, we have one more capacitance @ gate which is known as overlap capacitances →

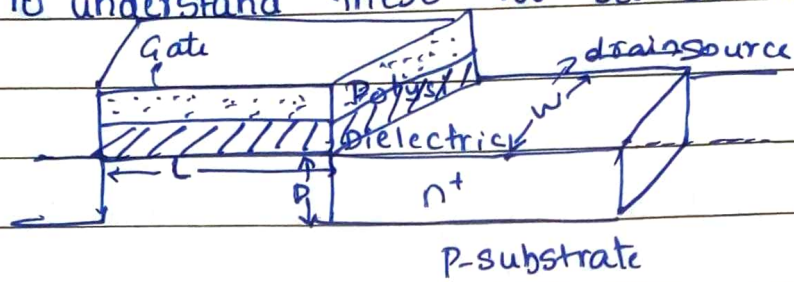



↓ when this happen in all region there will capacitances bt gate & drain and gate & source which is called as Overlap capacitance.

→ We have diffusion capacitance also w.r.t to drain to bulk and source to bulk.

→ Diffusion capacitance is divided into bottom wall and ~~top~~ (side) -wall capacitance.

To understand these we see 3D models of mos

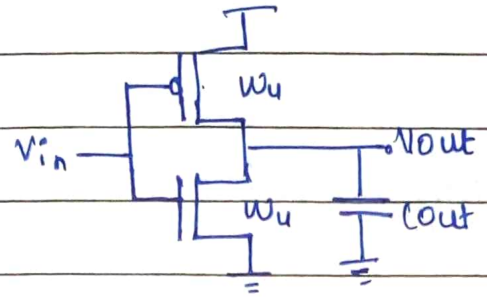



∴ Net capacitance is intrinsic capacitance and parasitic (difussion) capacitances.

# CMOS GATE DELAYS:

## INVERTER:

→ Unit Inverter is a Inverter w has 1 pmos & 1 nmos with sizes $W_{pu} = W_{pn} = W_u$ = unit width




$C_{out} = C_{FET,u} + C_L$

$t_{r0}$ = Zero load rise time = $2.2 R_{pu} C_{FET,u}$

$t_{r0} = \alpha_{pu} C_{FET,u}$

$C_{FET,u}$ is capacitance @ drain of inverter w is made up of 1 drain of pmos & nmos.

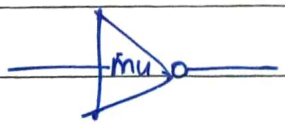
$\therefore C_{FET,u} = C_{Dp} + C_{Dn}$

$= 2C_{Du}$ (as sizes are same)

Node rise time $t_r = t_{r0} + \alpha_{pu} C_L$

→ fall time $t_f = t_{f0} + \alpha_{nu} C_L$

If I scale the Inverter, that means if I change the widths of devices



Now when I scale by factor m

$t_{ro,m} = 2.2 R_{p,m} C_{FET,m}$

$C_{FET,m} = m . C_{FET,u}$

$R_{p,m} = \frac{R_{p,u}}{m}$

as we know $R = \frac{1}{\beta (V_{DD} - V_t)}$

$\therefore t_{ro,m} = 2.2 \frac{R_{p,u}}{m} , C_{FET,u}$

$t_{rD} = t_{ro,m} = 2.2 R_{p,u} C_{FET}.$

$t_{r,m} = t_{ro} + \alpha_{p,m} C_L$

where $\alpha_{p,m} = 2.2 R_{p,m} = 2.2 \frac{R_{p,u}}{m}$

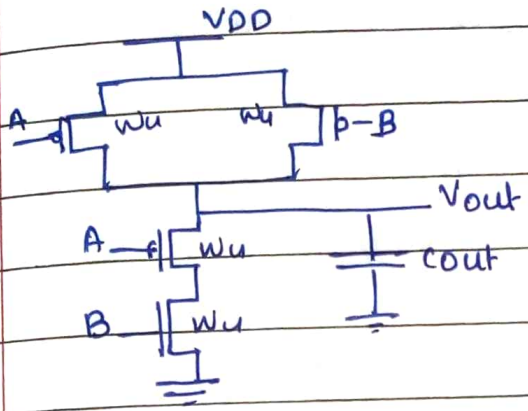
$\therefore t_{r,m} = t_{ro} + \frac{\alpha_{pu}}{m} C_L$

Similarly $t_{f,m} = t_{fo} + \frac{\alpha_{nu}}{m} C_L$

# CMOS Gate delays : NAND/NOR.

## Nand:

Unit nand gate: → Width of PMOS & NMOS will be same as that of unit inverter.




$C_{FET} = 3C_{DU}$

$C_f = \frac{2}{2} \times 3C_{DU} \Rightarrow 2C_{DU} = C_{FET,u}$

$C_{FETu_{Nand}} = \frac{3}{2} C_{FETu//}$

$t_{ro}' = 2.2 R_{pu} \frac{3}{2} C_{FETu}$

$t_{ro}' = \frac{3}{2} t_{ro}$ ↳ Zero load rise delay for unit inverter.

$t_r = \frac{3}{2} t_{ro} + \alpha_{pu} C_L$ ↳ same as unit inverter

$t_{rN} = \left(\frac{N+1}{2}\right) t_{ro} + \alpha_{pu} C_L$

↓ N inputs

when scaled by 'm' times

$t_{rN,m} = \left(\frac{N+1}{2}\right) t_{ro} + \frac{\alpha_{pu}}{m} C_L$

fall time $t_{fo}' = 2.2 [2R_{nu}] \left(\frac{3}{2} C_{FET,u}\right)$

$t_{fo}' = 2.2 R_{nu} 3 C_{FET,u}$

$t_{fo}' = 3t_{fo}$

$t_{fo_N}' = \frac{N(N+1)}{2} t_{fo}$

Overall fall delay $t_{fN} = \frac{N(N+1)}{2} t_{fo} + N\alpha_{nu} C_L$

when scaled up by 'm' times

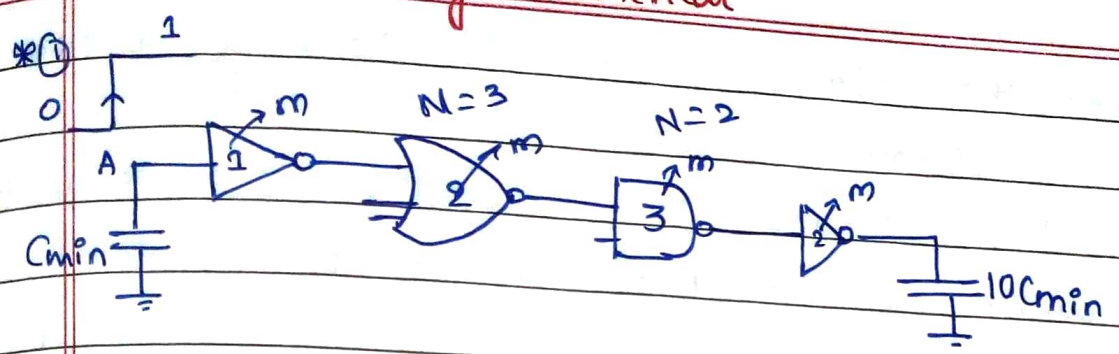
$t_{fNm} = \frac{N(N+1)}{2} t_{fo} + \frac{N\alpha_{nu}}{m} C_L //$.

* NOR Gate

$t_{fNm} = \left(\frac{N+1}{2}\right) t_{fo} + \frac{\alpha_{nu}}{m} C_{L//}$

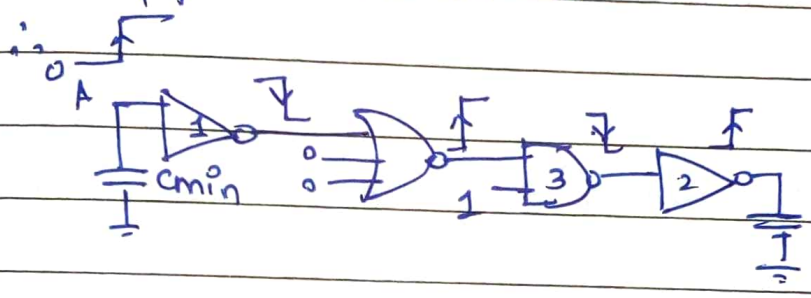
$t_{rNm} = \frac{N(N+1)}{2} t_{ro} + \frac{N}{m} \alpha_{pu} C_{L//}$

# CMOS GATE delay: Numerical



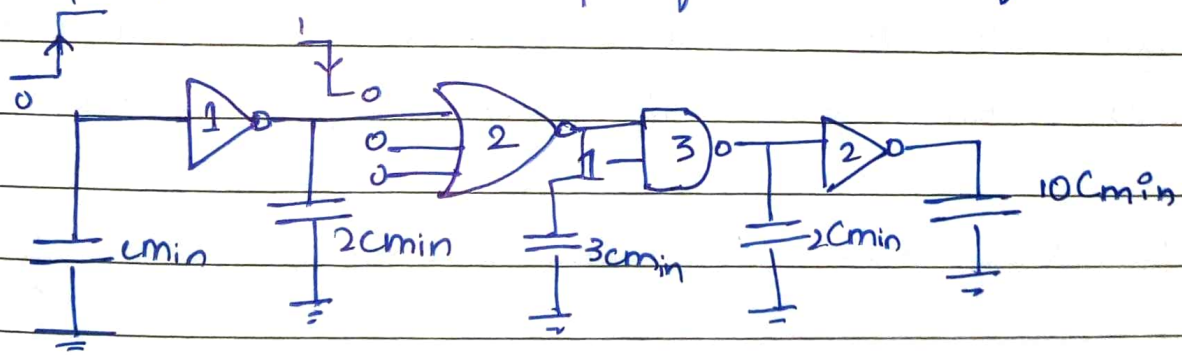

Estimate worst case delay.

→ To have transition for a nor we should have other ilps fixed 0 & for nand gate to have transition, other ilp should be 1

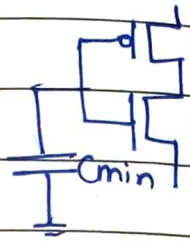



If input A changes from 0 to 1 then olp of inverter changes from 1 to 0 ∴ We need to find fall time for the first inverter.

To find fall time, we have to figure out the capacitance @ olp of inverter. (fan out capacitance)



Now this o/p of inverter is feeding a inv. ckt
w is 2times scaled
that means it is feeding 1 nmos w is twice width & one pmos which twice width



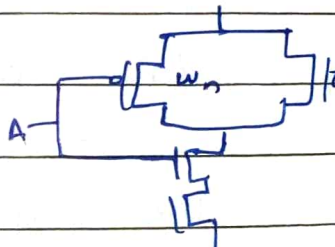

i/p capacitance of nor gate = o/p cap of inverter
= 2Cmin
similarly i/p cap of nand gate = 3Cmin
& o i/p cap of last inverter = 2Cmin

Cmin = Cpu + Cpu
= 2Cpu
↓ If scaled m units it becom mCmin

W.K.T Cout = (CFET) + Ch
↳ taken care in α

→ now Cl for Inverter = 2Cmin
" " nor = 3Cmin
" " nand = 2cmin
" " Inverter2 = 10Cmin

similarly for nand gate



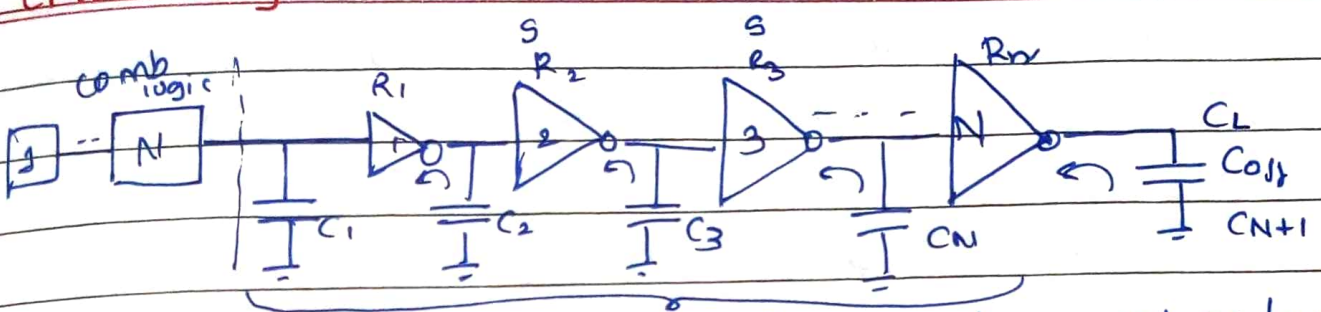

→ fall time for inverter, Rise time of 3 i/p nor gate, fall time for 2 i/p nand gate, rise time of inverter

↓

∴ time delay $t_d = \left[t_{fo} + \alpha_{nu}(2C_{min})\right] + \left[6t_{ro} + \left(\frac{3}{2}\right)\alpha_{pu}(3C_{min})\right]$

$+ \left[3t_{fo} + \frac{2}{3}\alpha_{nu}(2C_{min})\right] + \left[t_{ro} + \frac{\alpha_{pu}}{2}(10C_{min})\right]$

# CMOS delay Minimization Inverter Cascade.




Inverter Cascade ω act as buffer

→ Let us find delays @ each inverter

Overall delay will be sum of all time constants

$$T_d = T_1 + T_2 \cdots T_N$$

$$T_d = R_1C_2 + R_2C_3 + R_3C_4 \cdots R_NC_{N+1}$$

Suppose if i choose uniform scaling factors

→ S is scaling factor.

when we scale 2 inverter by S

then $C_2 = SC_1$ & similarly

$$C_3 = SC_2 = S^2C_1$$

$$\therefore R_1C_2 = \underline{\underline{R_1SC_1}}, \quad R_2C_3 = \frac{R_1}{S}S^2C_1 = \underline{\underline{SR_1C_1}}$$

$$\therefore R_1C_2 = R_2C_3 = R_3C_4 = SR_1C_1$$

∴ delay of each stage is same if we have uniform scaling.

$$\boxed{\therefore T_d = NSR_1C_1}$$

if $R_1C_1 = T_r$ = ref inverter

$$\boxed{\therefore T_d = NST_r}$$ → time delay for overall inverter chain.

Now let us think decide about how many stages should we have or the value of N

$C_L = S\,C_{N-1}$ , $= S^2 C_{N-2} = \cdots$

$\therefore C_L = S^N C_1$

$\Rightarrow \boxed{N = \dfrac{\ln(C_L/C_1)}{\ln(S)}}$

Now we have to find value of N for w the delay is minimum

$\therefore$ derivate w.r.t S & equate to 0

WKT $T_d = NST_r$

$T_d = \dfrac{\ln(C_L/C_1)}{\ln(S)} S T_r$

$\ln(S) - 1 = 0$

$\ln(S) = 1$

$\boxed{S = e} = 2.713$

This says if we inverter chain & if we scale them by factor of e then we will have minimum delay.

$\boxed{N = \dfrac{\ln(C_L/C_1)}{1} = \ln(C_L/C_1)}$

## Numerical :

→ $C_L = 10pF$, $C_1 = 20ff$, $\beta_1 = 200\mu A/V^2$

Find Design inverter chain find N & s.

→ $N = \ln\left(\frac{C_L}{C_1}\right) = \ln\left(\frac{10p}{20ff}\right) = \ln(500) = 6.21 \approx 6$

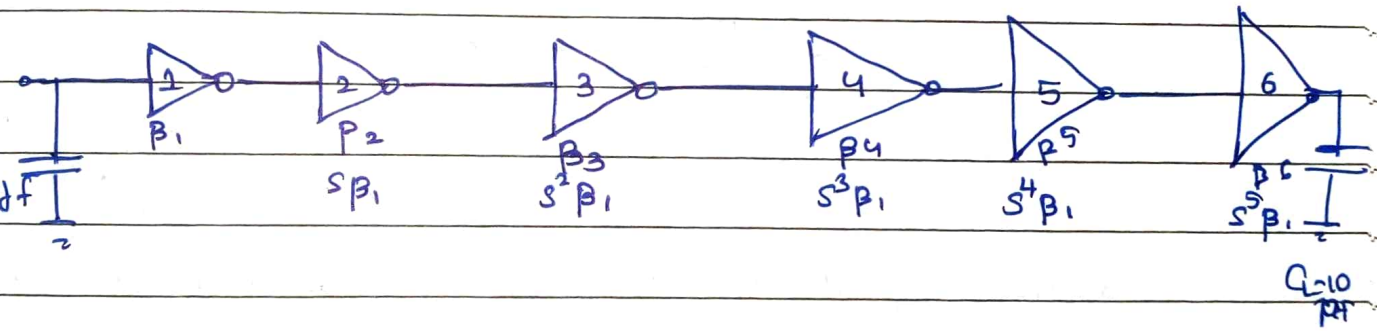
But N should be a integer

∴ N should be ~~buffer~~ even as it is a buffer

∴ $\boxed{N = 6}$

$S = \left(\frac{C_L}{C_1}\right)^{1/N} = 2.82$

∴ Inverter Cascade is



# CMOS LOGICAL EFFORT

→ If we have more possibilities for doing a logical chain. To choose best one we use or we find logical effort..

→ Overall delay $d = f + p$

where $f$ = stage effort / effort delay

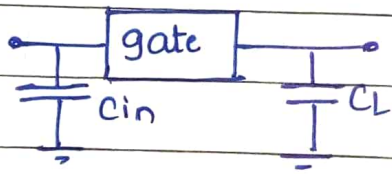
$p$ = parasitic delay

→ Stage effort $f = gh$

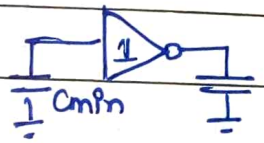
where $g$ = logical effort

$h$ = electrical effort

→ logical effort : defined as




ref inverter unit




logical effort of any gate is defined as ratio of input capacitance of that gate to input capacitance of ref unit inverter.

$$\therefore g = \text{logical effort} = \frac{C_{in}}{C_{min}}$$

→ Electrical effort : is ratio of o/p capacitance of that logic gate to of i/p capacitance of that gate

$$h = \frac{C_L}{C_{in}}$$

→ G = path logical effort = product of logical effort of each logic gates.

If I have N logic gates then

$$G = \prod_{i=1}^{N} g_i$$

→ H = path electrical effort = product of individual electrical effort

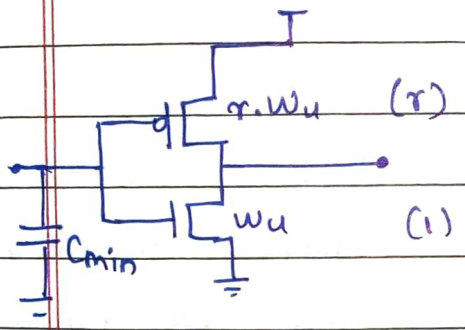
$$\therefore H = \prod_{i=1}^{N} h_i$$

→ ∴ Path effort = F = GH

↳ $\beta_p = \beta_n$

(1) Logical effort of ref Inverter (Symmetric inverter)

If $\beta_p = \beta_n$ then $r.W_{PU}$ & $W_u$



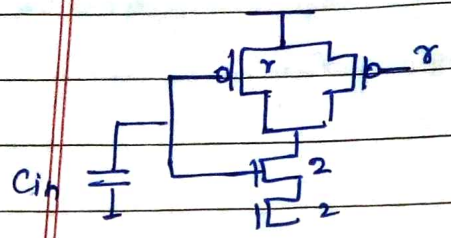

$$C_{min} = C_{Gn} + C_{Gp}$$
$$= C_{ox} W_u L + C_{ox}(r W_u) L$$
$$= \frac{C_{ox} W_u L}{C_{Gu}} (1+r)$$

$$\therefore C_{min} = C_{Gu}(1+r)$$

logical effor $g_{NOT} = \frac{C_{in}}{C_{min}} = \frac{C_{min}}{C_{min}} = 1$

r = mobility ratio

## logical effort for Nand gate.




Now $C_{in} = C_{au}(2+r)$

$$\therefore g = \frac{C_{in}}{C_{min}} = \frac{C_{au}(2+r)}{C_{au}(1+r)}$$

$$\therefore \boxed{g_{NAND} = \frac{2+r}{1+r}} \rightarrow 2 \text{ i/p nand gate}$$

→ for n input nand gate

$$\boxed{g_{NAND_n} = \frac{N+r}{1+r}}$$

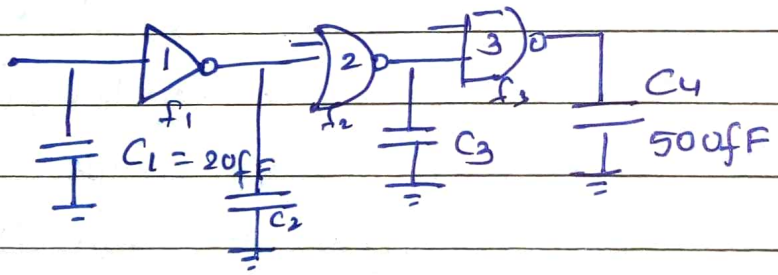
## logical effort for NOR Gate.

$$\boxed{g_{NOR,N} = \frac{1+Nr}{1+r}}$$

## NUMERICAL:

$r = 2.5$

(1)




Find out sizing this logical devices so that we have minimum delay across chain.

→ $F = GH$

$f_1 = f_2 = f_3 = \hat{f}$

$F = f_1 \cdot f_2 \cdot f_3 = (\hat{f})^3$

$\therefore \hat{f} = F^{1/3}$ $\quad (F^{1/N})$

$$F = (g_1 g_2 g_3)(h_1 h_2 h_3)$$

$$= \left[(1)\left(\frac{1+2\gamma}{1+\gamma}\right)\left(\frac{2+\gamma}{2+\gamma}\right)\right]\left[\left(\frac{C_2}{C_1}\right)\left(\frac{C_3}{C_2}\right)\left(\frac{C_4}{C_3}\right)\right]$$

$\boxed{F = 55}$ $\Rightarrow \therefore \hat{f} = (55)^{1/3} = 3.8$

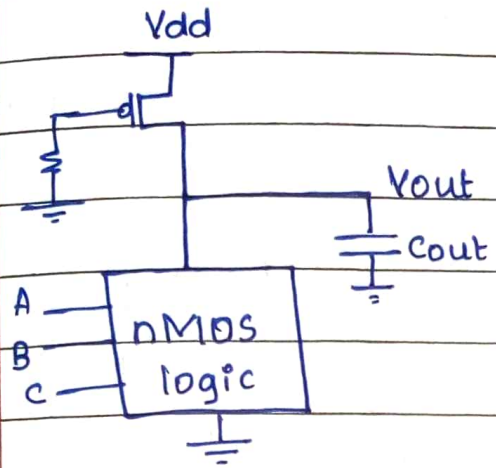
$f_1 = g_1 h_1 \Rightarrow C_2 = 76.35fF$

$f_2 = g_2 h_2 \Rightarrow C_3 = 169.5fF$

$f_3 = g_3 h_3 \Rightarrow C_4 = 500fF$

# * Pseudo nMOS & clocked CMOS logic circuits:

## • Pseudo nMOS



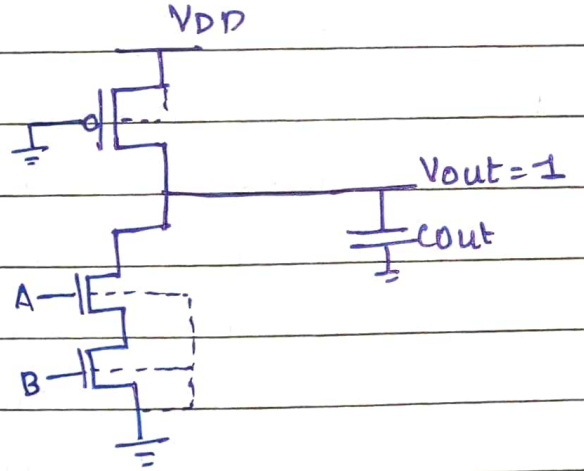

pMOS ckt gets reduced

-Disadvantage: Ratioing of the logics : we need to manage strength of nmos & pmos.

pMOS $\left(\frac{W}{L}\right)_p \gg \left(\frac{W}{L}\right)_n$ to compete with nMOS logic structure

since pMOS gate is grounded hence Vout = 1

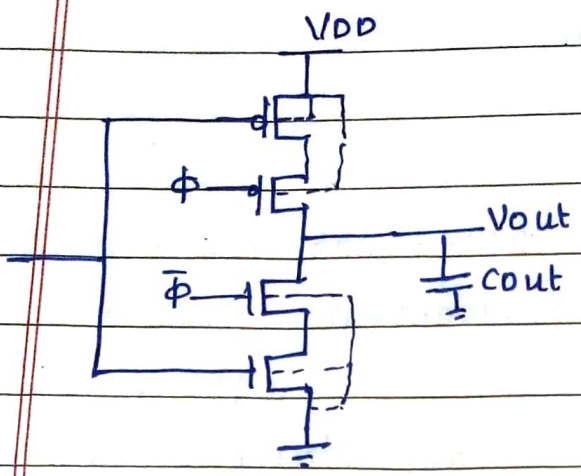
Example • NAND Gate = $\overline{A \cdot B}$




if A = B = 1

then Cout is able to discharge.

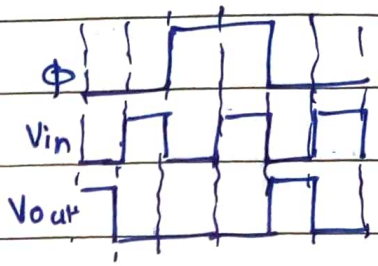
## * Clocked CMOS Ckt: Inverter logic




← $C^2MOS$ Inverter

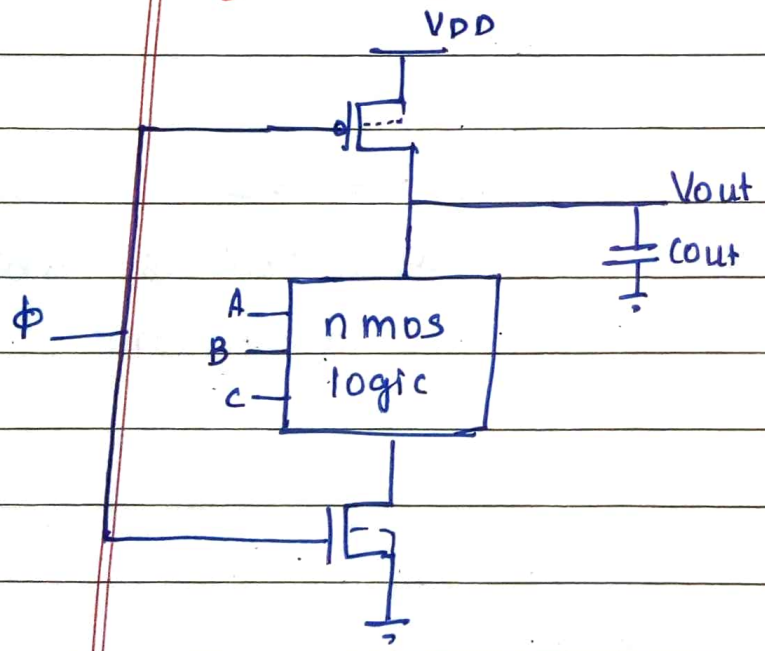
$\phi$ = clock

$\phi$ & $\bar{\phi}$ are non-overlapping clock




When $\phi = 0 \Rightarrow \bar{\phi} = 1$ only then inverter works/operates.

## * Dynamic & Domino Logic Circuits.




$\phi = 0$ → pre charge state

$\Rightarrow$ Vout = VDD

$\phi = 1$ → Evaluate state

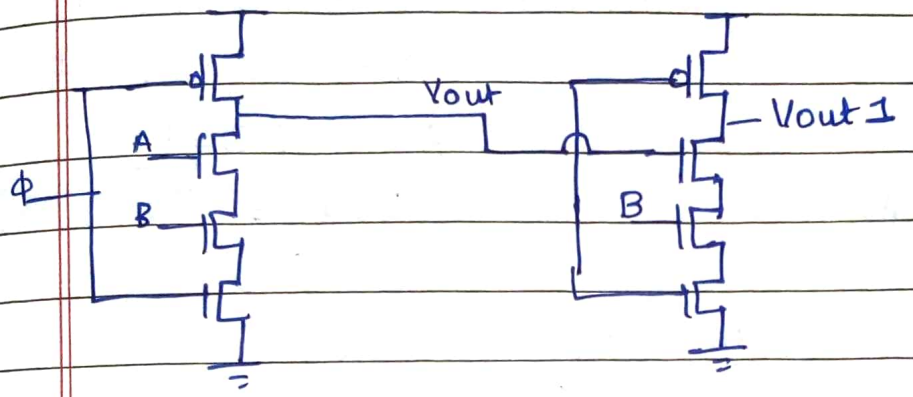
$\Rightarrow$ nmos logic gets evaluated. ie whatever is the logic of nmos logic Cout can discharge or hold the value of Vout of '0' or '1' depending on the nmos logic used.

Disadvantage:

→ Cascading of dynamic logic

→ Charge sharing

① Cascading issue of dynamic CMOS logic:




Initially A = 1, B = 0

Vout = 1 [Φ = 0 precharge]

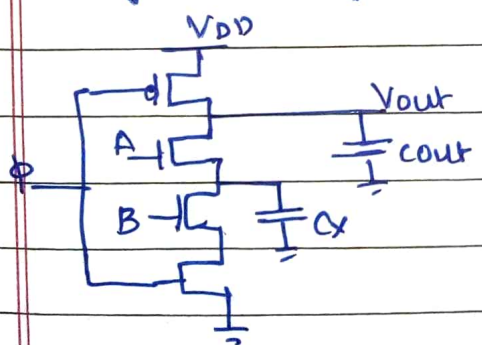
Vout = 1 [Φ = 1 evaluate state]

↳ as B = 0

A = 1, B = 1 [Evaluate State]

Should be Vout = 0 ⇒ But Vout = 1 → propagates to next logic

② Charge sharing



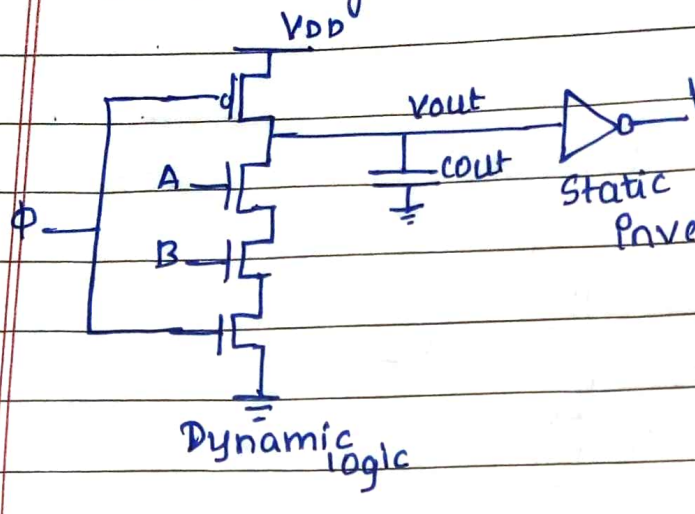

Φ = 0 Vout = VDD

⇒ A = 1 & B = 1

Cout discharges & charges Cx

hence Vout ≠ 0

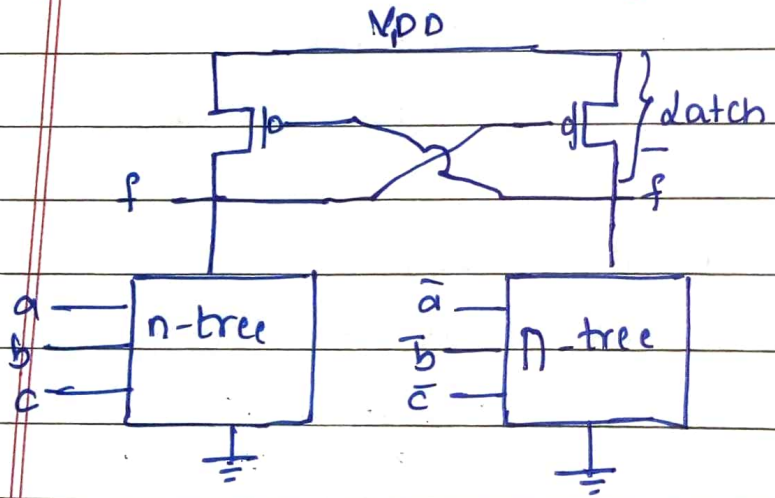
* To overcome the problem of dynamic logic
→ Domino logic circuit = dynamic logic + static inverter.



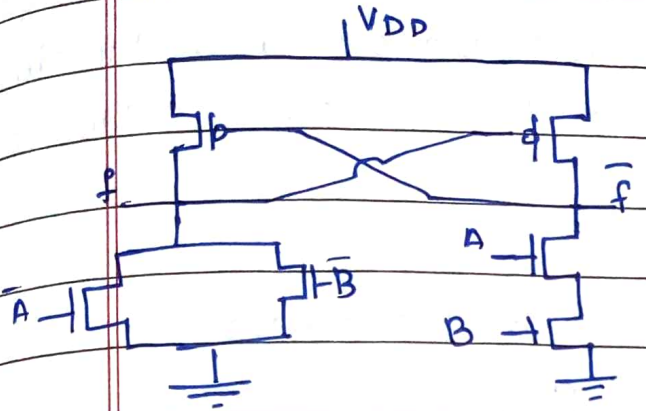

ϕ = 0 precharge state
Vout = VDD = 1
⇒ A = 1 & B = 0, Vout = 1
⇒ A = 1 & B = 1
Vout = 1 ↳ [Evaluation state]

* Dual Rail logic Networks:

P CVSL → Cascade Voltage Set logic



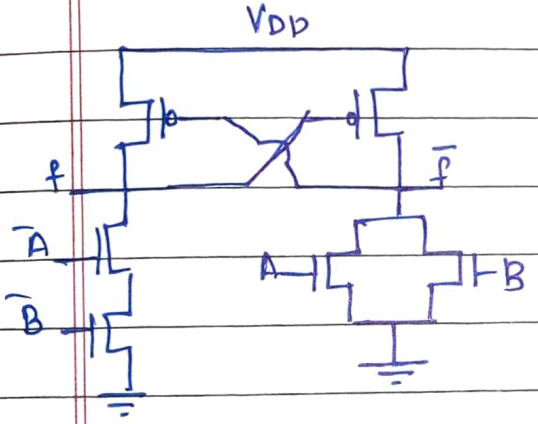
- AND/NAND




| A | B | AND | NAND | f | $\bar{f}$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 |

Reduces the area size

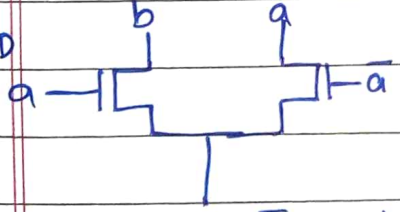
- OR/NOR




| A | B | OR | NOR | f | $\bar{f}$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |

ii) CPM → Complementary Past logic

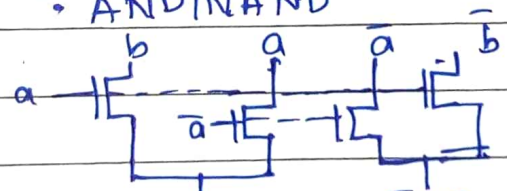
- AND




$ab + a\bar{a} = ab$

[AND]

- AND/NAND




$ab + aa = ab$

$\bar{a} + \bar{b} = \overline{ab}$

$\bar{a}\bar{a} + a\bar{b}$

$= \bar{a} + ab$

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

$= \bar{a} + \bar{b} = \overline{ab}$

Disadvantage of using nMOS as a part of transistor is that complete logic is not transferred @ the o/p.

To overcome we use inverter @ o/p we holds the complete logic