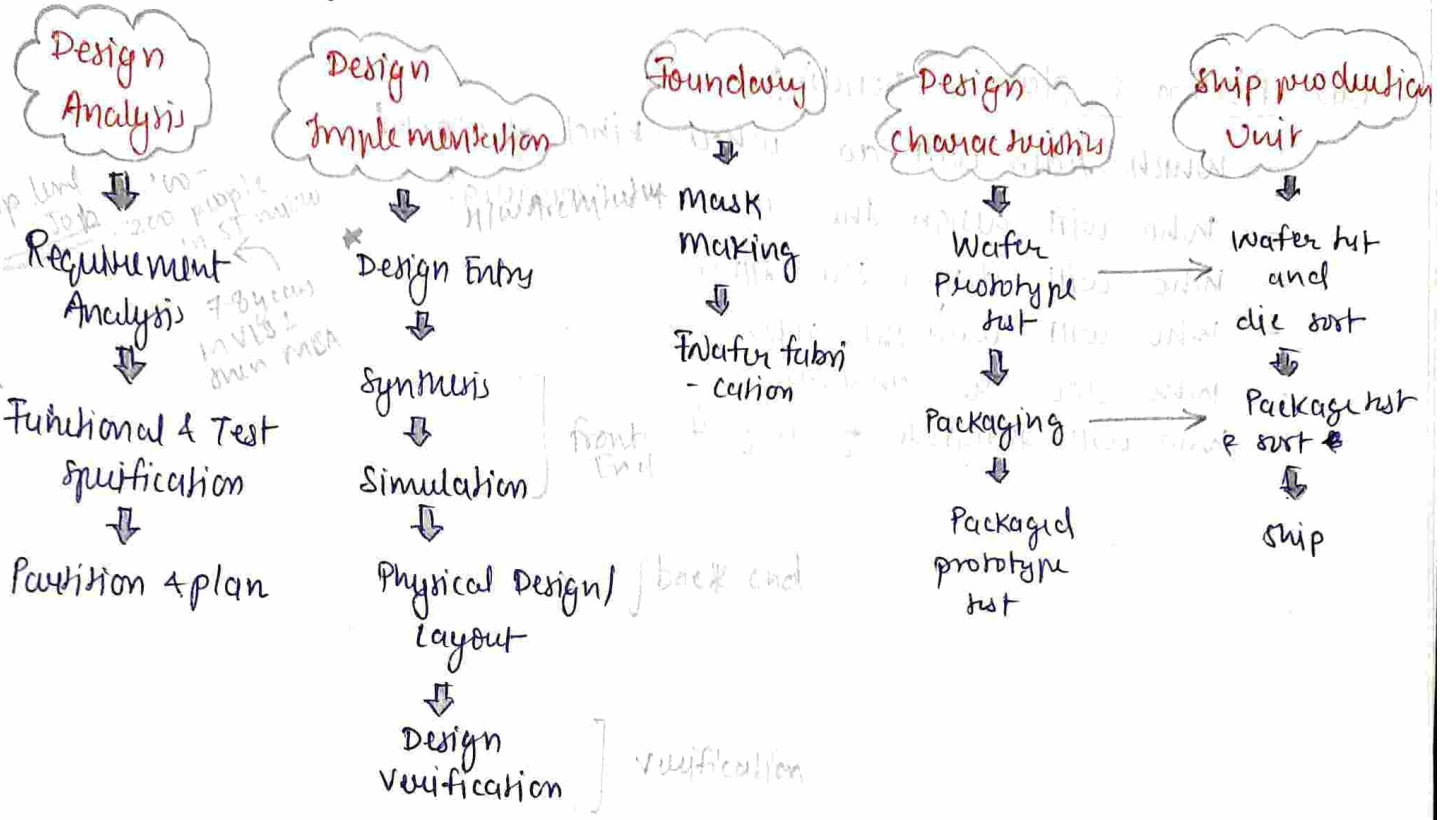
1. Introduction to VLSI Design

Video-1
Lecture 2

SOC Design steps



(1) Design Analysis:

(i) Requirement Analysis :- Here the market Analyst will look for the customers is at which price point they are ready to pay for the product.

(ii) Functional & Test spec :- While making the face recognition of the iphone we have to decide its func^n's ie, it should be infrared, it should recognize the face, it should unlock the screen, it should unlock the screen if 80% or above face matches.

- Along with the functionality, you have to decide how we are going to test the functionality.
- We will be taking 10,000 samples from the camera & if any sample matches with the i/p face, it will unlock the screen.

"functional specification" and "Test Specification"

** If the functional specification changes at any time of the design flow, then we have go back to the functional & test specs again. and start over again.

(iii) Partion & plan : Deciding
- Which team will do what kind of work?
- Who will design the layout for camera?
- Who will design the H/w.
- Who will write the software
- Who does the analysis
- Who will integrate all of it.

⇒ All these process can be done by 10 people.

- These kinds of job are provided in headquarter by :- (1) very close to customer
  - TI - US
  - micron - US
  - Samsung - Korea
  - ST microelectronics - Italy & France

② Design Implementation

(i) Design Entry :- Here we have to make a design H/w which will do the calculation to ensure that face recognition happens accurately.

(ii) Synthesis :- Here we decide like what components are required in here like adder, ~~counter etc~~. multiplier, comparator etc. . And I synthesis it & make the connections.

(iii) Simulation :- Here we simulate our ckt & tell whether the o/p is correct or not.

* Here we can emulate the ckt using FPGA etc.

Simulation → Emulation
(CAD) (FPGA)

* Aging simulation model.
* Monte carlo

(iv) Physical design & layout : • Tape out

Hard core implementation.

(v) Design verification : Verifying the complitly disigned ckt. then we tape out the design & send it to foundary.

DRC and all.

③ Foundary :

(i) mask making
(ii) Wafer fabrication

* design is send to mask shop & then the masks are send for the wafer fabrication.

- Depending on the technology, there can be 39, 28, 47, 60 masks that the design would go through.

④ Design characterization :

(i) Wafer prototype test
(ii) Packaging
(iii) Packaged prototype test

- We have 12 inch diameter wafer with our, which can have billions of dies (ie, IC used in our phone). We test if all the dies are working fine or not.
- ~~To testing methods~~
- while packaging, we are connecting wires from the pins of the package to the pads on your die.
- there could be some error in the connection above, therefore we do Packaged prototype test.

Q. What is meant by design characterization?

→ If I have designed a ckt at 1.2 GHz then I have to check after its fabrication that for silicon at what clock speed is it working 1.1GHz or 1.25 GHz.

PPA ⇒ power - performance - Area.

⑤ Ship production Units

(i) Wafer test & die sort

(ii) Package test & sort

(iii) Ship

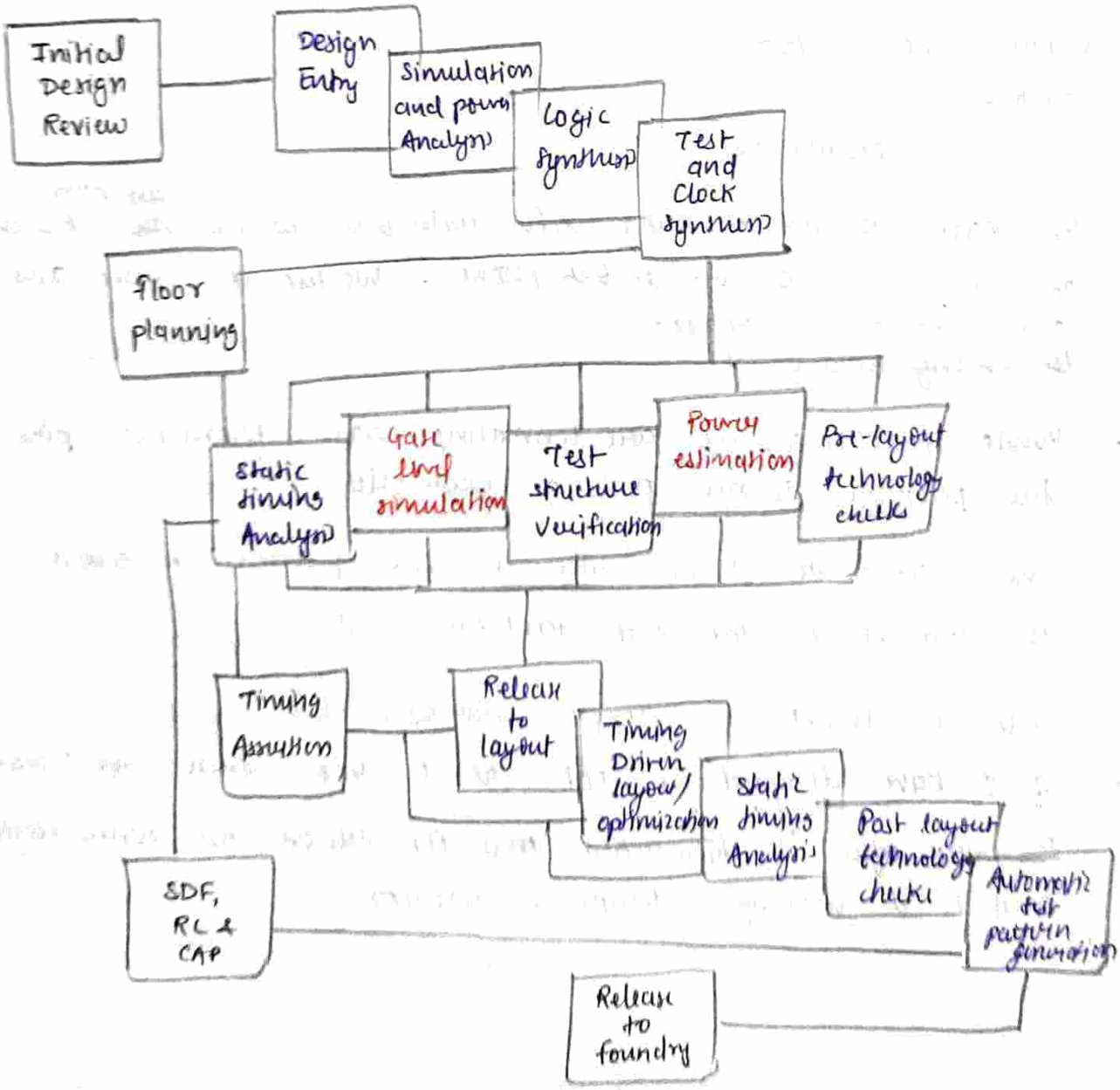
- Testing the wafer & calculating its frequency & then sorting the die on the basis of its freq & performans

Ship is just an Eg.

- Die sort means when the dies are in wafer.

  Package sort means when the dies are packaged into their ICs.

Design Implementation :-



① Initial Design Review :- Where you say I need this kind of a processor, I need such kind of memory, interfaces, DDR and so on.

② Design Entry :- RTL design, where you write a verilog code.

③ Simulation and power analysis :- Now you check whether after simulation that power calculation is meeting your needs.

④ * Logic synthesis :- Converting an RTL verilog code to netlist to a gate level netlist. (ie, we have design a adder logic in our design entry (RTL) now we make the (design the) logic behind that adder.

⑤ Test & clock synthesis :-

- Suppose you have 1 million FF and every FF needs a clock signal to run. so can we use single clock to run all the FF ? No.
- We will use buffers b/w the clocks & FFs, such as 10 buffers will drive the 10 FFs. So this part of inserting clock buffers. such that none of the clock pins nodes has a load more than 100pf
- If you have 2 FF & both are supposed to be run sequentially ∴ equal no. of buffers must be put in path of both the FF. This is called as clock synthesis.

- Stm 32
- atmega

⑥ Floor planning :-

Here we decide where we are going to put memory, where the processor is going to be placed, so on.

Chip production units

(i) Wafer test & die sort

(ii) Package test & sort

(iii) Ship

(7) Static timing Analysis (STA)

- Here we are going to analyze the timing behaviour of the ckt.
- Checking whether the ckt is meeting the setup & hold time conditions.

(8) Gate level simulation

- When we have converted the A+B logic to Adder ckt then simulating the gate level netlist is done at this level.

(9) Test structure verification

- Here we test the components individually.

(10) Power consumption

- From the real gates available here we estimate the power consumption.

(11) Pre-layout checks

— We can check whether is there any single node that is driving from two inputs.

— or is there any node which is floating.

— or is there any load at particular gate which it can't bear to hold.

* After pnr checks you can generate your netlist.

(12) Timing assertion

(13) Relcan to layout

- After deciding where we are going to place all of our components we can now start placing it in the chip.

(14) Timing driven layout optimization :

- If you know that output of this AND gate is going into this OR gate. Therefore we can put this AND & OR gates together, this is layout optimization.

(15) STA in the layout :-

- Because in layout we ~~can~~ ~~do~~ know how long the wire is & the ~~o~~ actual delay of the ckt must be known.

** Every polysilicon wire needs to be connected to source or drain because the charge due to plasma on the polysilicon must be sunk into the substrate.

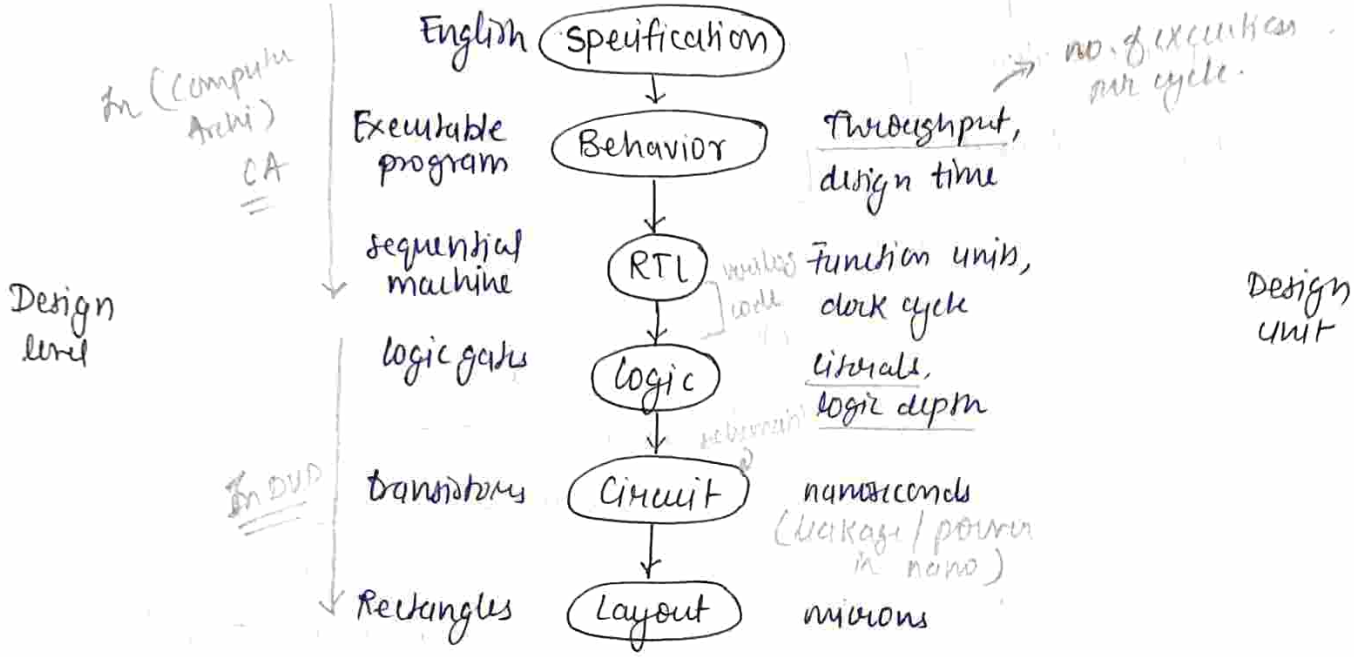
class
01-09-2022

- finer granularity

- STA → worst case analysis of ckt w.r.t time.
- Vector based analysis.
- SDF → standard delay format

Lec-3 Library Design

1. Design Abstraction



→ .lib file

→ Zodiac

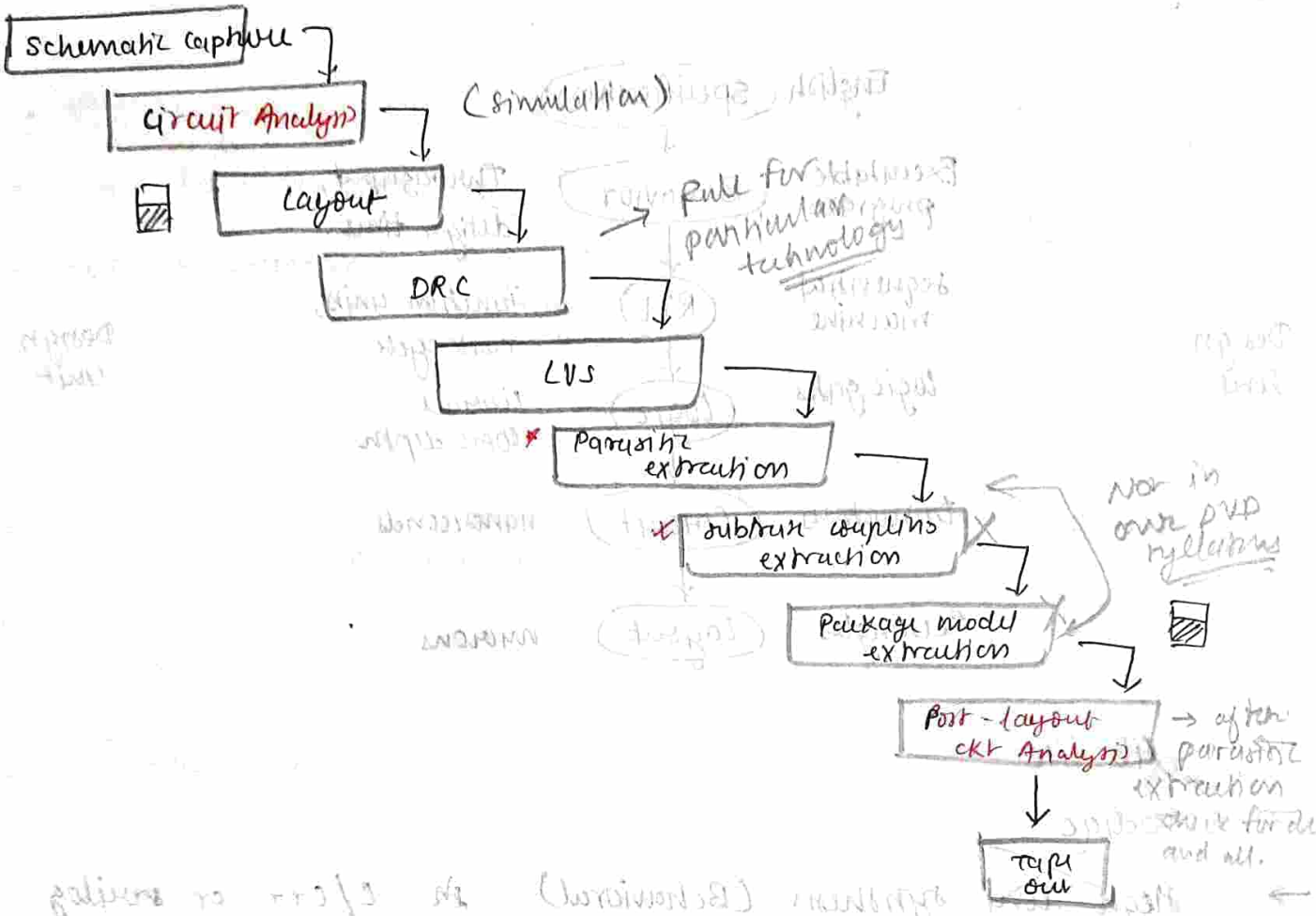
→ High level synthesis (Behavioral) in C/C++ or verilog then go to RTL phase by doing high level synthesis tool.

→ Layout can be automated but it may take more area as compared to manual layout.

→ Usually we do verification after every step of design abstraction.

2. Library Design Flow

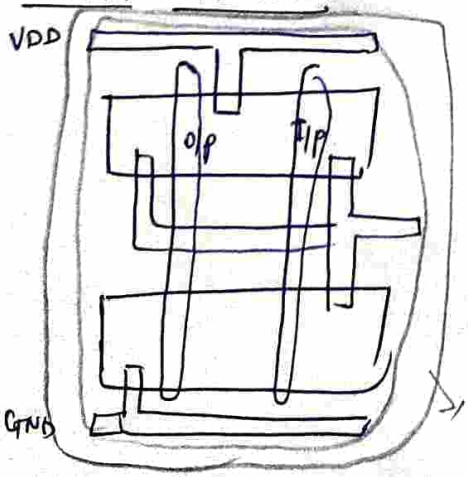
2. Library Design flow :-



→ suppose we make a NAND gate layout then we have to maintain some minimum spacing b/w the wires, min area must be full filled & this conditions are DRC.

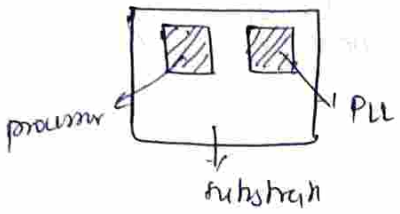
→ LVS is used to check whether the ~~take~~ obtained layout matches with the designed schematic or not.

Parasitic extraction :-



- metal will have some inbuilt R & C inside
- As we can see that output is connected to the metal hence some capacitance effect will be there (due to metal wire) & I want to extract that (find the value of capacitance) capacitance & ~~so~~ how it is effect on my circuit
- We will do all the capacitance & resistance extraction. (means finding more values)

- **substrate coupling** →

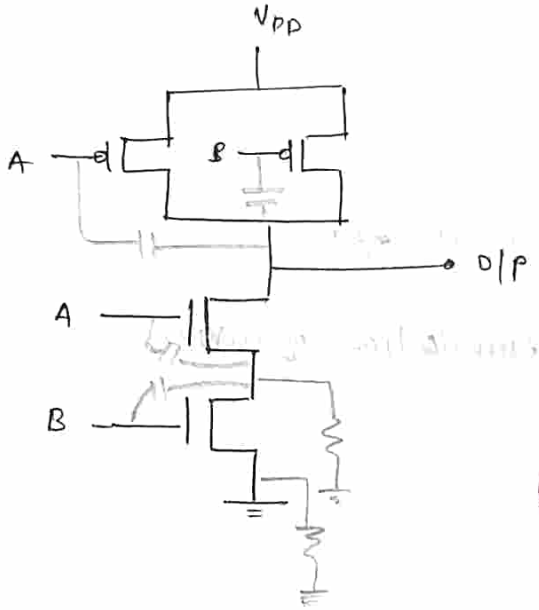


Since processor & PLL are on same substrate, then the noise due to processor will effect the PLL. So there should not be any substrate noise.

- It will be like source to body capacitance.
- In DVD we will not discuss this
- It can be removed by the guarded ring.

---

* If there is any RC in the ckt, it adds delay to the ckt.



* Post-layout simulation will have all the capacitance & then we simulate.

* We have to see whether the C & R present due to metal connection effecting our ckt or not.

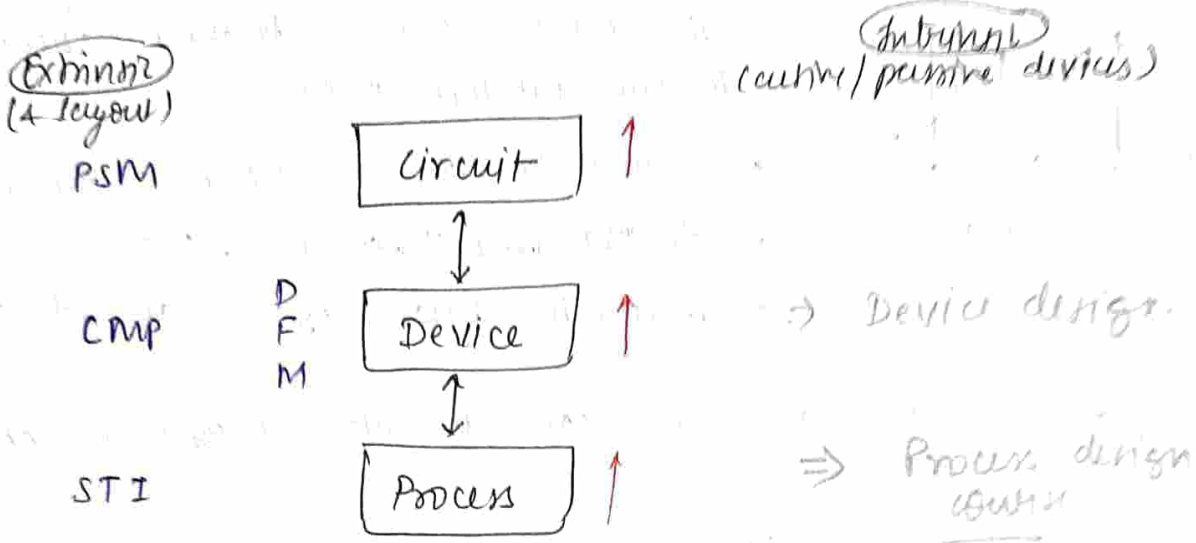
(*) When you increase the size of your PMOS & NMOS then it can flow more current then this extra current can discharge the charges in RC in less time hence the delay will be reduced.

What is

(*) ~~Increase~~ the significance of doing parasitic extraction?

→ By doing parasitic extraction in the metal wire we get to know about the R&C value & finally get the delay in the ckt now we can increase the delay by adding more RC (inckt) or we can decrease the delay by increasing the size of PMOS & NMOS.

3. Before ckt Analysis - model Development



- TCAD tools → Silvaco

* Before ckt design - Device design

- Silvaco → physics based simulation of devices

⇒ - Library design
- soc design
- IP design

* Key factors for Tremendous Growth of microelectronics

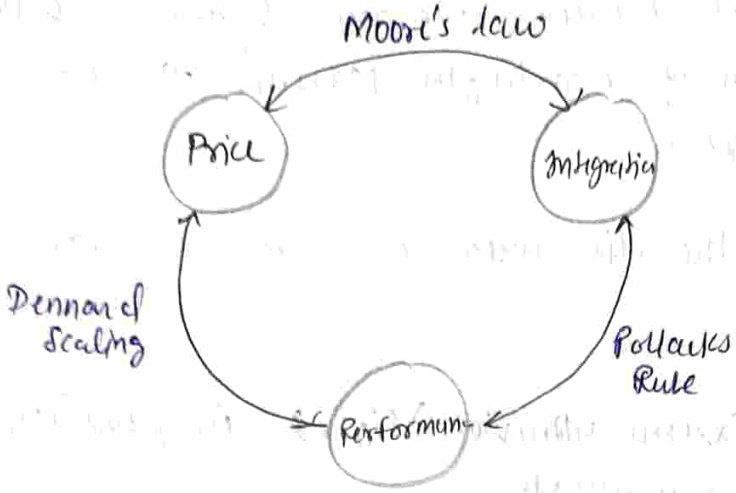
Moore's law : Denard scaling : Pollacks Rule

— Every 18 months the no. of transistor that you can pack gets double in per cm^2 area.

- Every 18 months the area will be reduced to design the same ckt.

- So we can add more ckt on the same die. i.e we can add more functionality.

- Denard scaling : Area ↓ ⇒ price ↓
- Pollacks Rule : Integration ⇒ performance ↑



Moore : time ↑, integration improves, area ↓

Dennard : area ↓, price ↓, performance ↑

Pollacks : ~~cost~~ integration improves, area ↓, performance ↑

- In 1994 125 Rs for 1 call for 1 min (Eg.)
  Now the value of money is degraded in VLSI industry.
- Moore law is just an observation.
- Dennard gives the scaling rule for reducing the transistor size.

(+) till 20nm price is reducing
after that price stops reducing.

* In actual, the fabrication ecosystem is not able to handle finer geometries than 28 nm. Hence total no. of mask increases.
* physical upto 45nm are designed only & after that when we say like 10nm it means not physical 10nm dimension but it some what 20-20% better than its previous version like 32nm.

=> physical size is not changing after 45nm but some how the performace of the ckt is improving by other things.

* for finer geometries, the cost of making mask increased exponential is from 180nm to 28nm

(vi) It is getting difficult because we are trying to design 10nm using the light of wavelength 193nm. Therefore we have to do lots of optics.

* when you reduce the die area the yield of correct die increases.

* We use EUV (Extreme ultraviolet) light for photolithography as it has 13nm wavelength.

→ It will reduce the no. of mask you require.

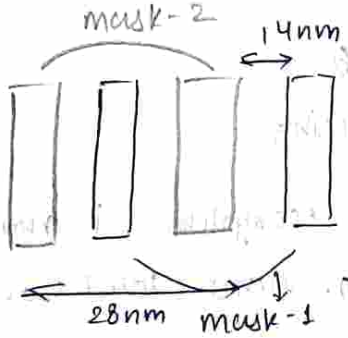
- EUV is made by ASML company (only one)
(Stock of ASML is getting higher & higher).

wavelength like X-ray

EUV > 100 W/hr

⇒ We are also adding some additional content in our chips which we can't scale. Like Analog ckt, high temp sensors in automotive.

** RET → Resolution enhancement techniques



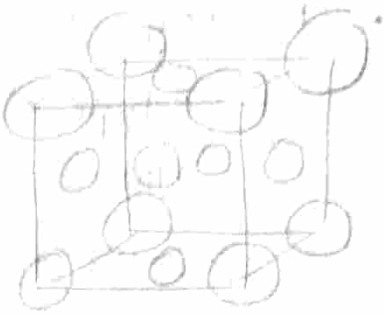
⇒ no. of mask increases
∴ price for chip ↑.

⇓ Dual patterning.

2.5D design → image sensor + processor together.

Lec-4 **Fabrication** [Re-watch the video]

Crystal structure :



- Diff. planes of same material, say diagonal plane are dissimilar
  - diff. density of atoms
  - diff. topology of atoms

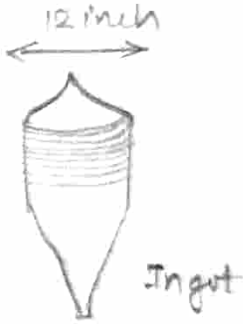
* anisotropic → different properties in different direction.

- Different EF on both planes
  - diff. electrical properties

- Plane of silicon wafer decides its behaviour
  - has to be accounted for when wafers are being made.

How such required plane can be fabricated?

(i) Crystal from sand



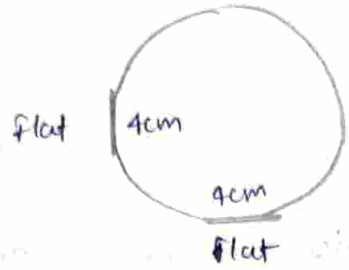
- silicon crystal (seed) dipped into molten silicon.
- Gradually pulled out.
- molten silicon bonds to crystal
- solidifies according to crystal's orientation
- speed determines diameter of crystal - called ingot.
- size of diameter can be : 6", 12", 8".

(ii) Ingot to Wafer

- slicing
- Grinding
  - To make circular & of desired size.
- Lapping
  - To make of desired thickness
- Polishing
  - To remove top layer that got dislocated.

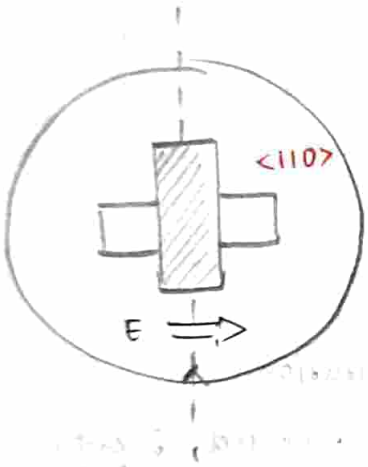
- substrate doping
- Primary & secondary flat → used to determine the plane.
  - P-type <111> substrate is used - so no secondary flat.

- Why notch > [110]
- flat vs Notch ?
- what is meant by rotation?



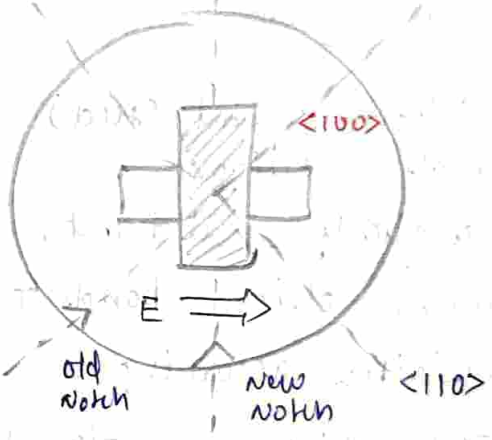
n-type wafer - 2 flat
p-type wafer - 1 flat

(iii) Rotated substrate (wafer)



"standard substrate"

* Current is flowing in <110> direction



"Rotated substrate"

* current is flowing in <100> direction

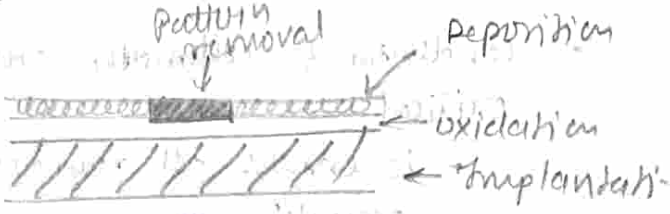
* After rotation, mobility of holes are improved. ie, to get the same current, we can have smaller device.
* hole mobility is increased significantly after 65nm tech.
  * p:n = 2.
* e^- mobility remains same & you could save more area.

* Because of the change in effective mass of carrier, mobility is changing.

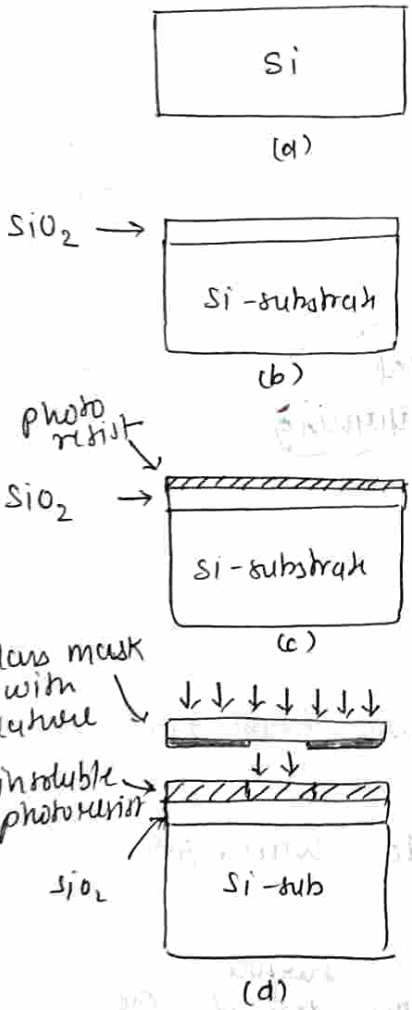
(iv) Process in Si Manufacturing

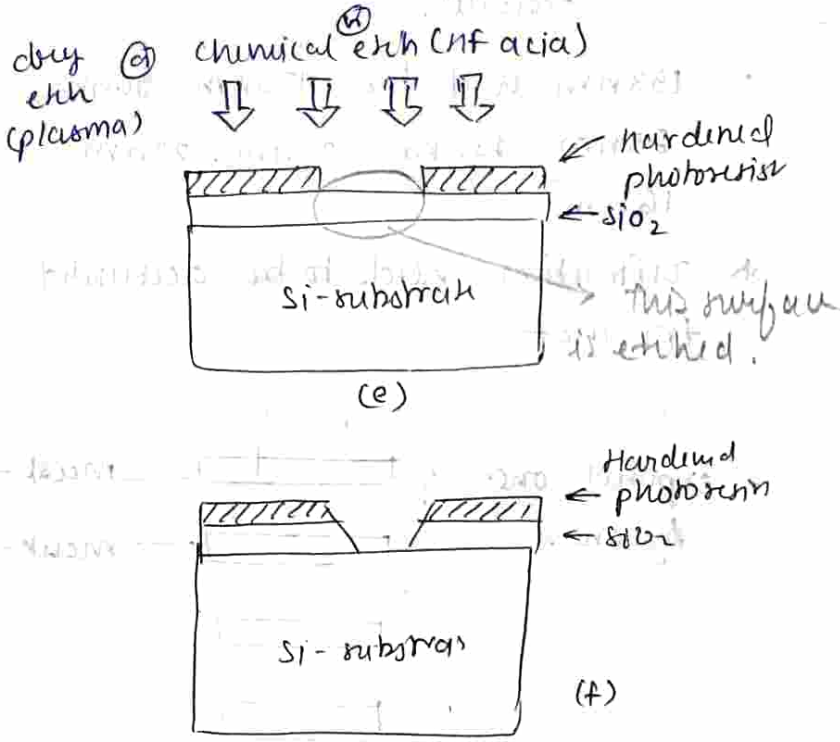
- Implantation
- Oxidation
- Deposition photo resist
- Patterned removal

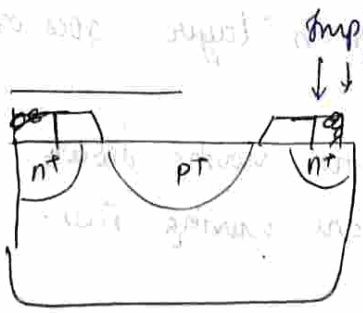
* Repeated operation of these steps for diff. time durations results in a functional chip:



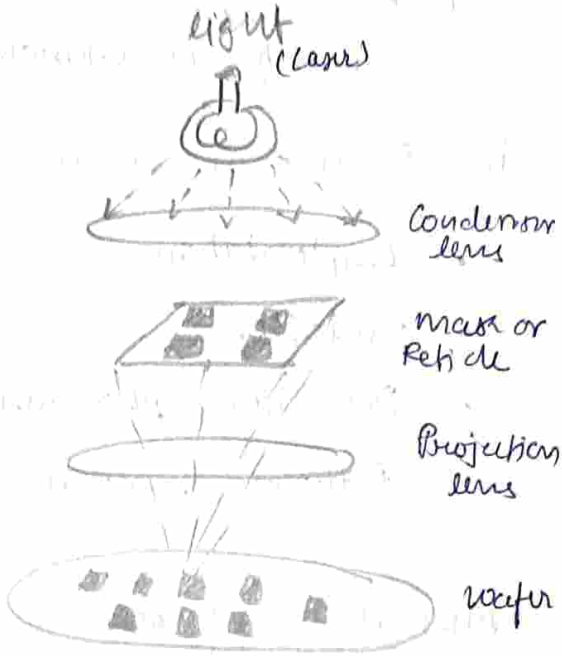
Photolithography ⇒







Photolithography Technique

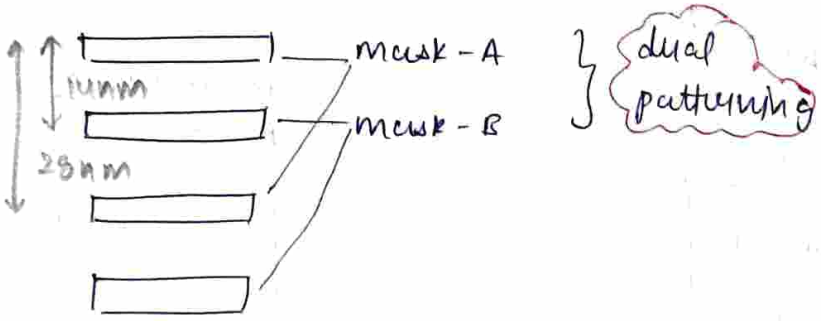


- Pattern made on a mask
  - Positive mask ✓
  - Negative mask ✓
- Photoresist expond to light through the mask
- condensor & collimator are critical
  - to ensure all rays are parallel.
- 193nm used for 130nm, 90nm, 65nm, 45nm, 32nm, 22nm, 16nm.

* Diffraction needs to be accounted for that

dummy poly !

- expond one by one.



* Tubelight generate a spectrum of light with large range of wavelength.

* — But laser will generate light of single wavelength.

* All the things like immersion layer goes on the ~~top~~ surface of the Si.

{ heavy hole — if holes are moving slowly
  light hole — if holes are moving fast.

193 nm - the reality

- Diffraction for present range of feature sizes.
- Loss of accuracy & precision
- +/- 5nm is easily in scope of error
- Optical proximity correction
  - to reduce ill-effects of diffraction
  - to diffraction to enhance resolution
- Change in medium of light to change effective wavelength
  - So it is 193i then days
- Increasing lithography cost in every technology
  - increase focus on "first time right"
- EUV is necessary for price scaling at lower than 20nm.

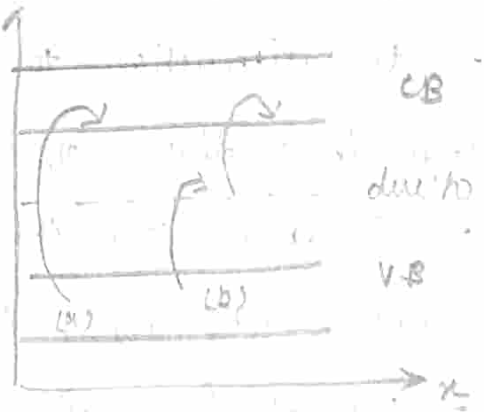
* During the fabrication of resistor, the actual metal contact is less than as it was supposed to be. Less area means same current will flow from the less area hence resistance will increase.

Process in Silicon Manufacturing

- Oxidation
  - Gate dielectric
  - Inter metal dielectric
  - Masking
- Deposition
  - Gate
  - Interconnects (metals)
- Patterned removal
  - Etching
- Implantation
  - Dopant to modify electrical properties of device.

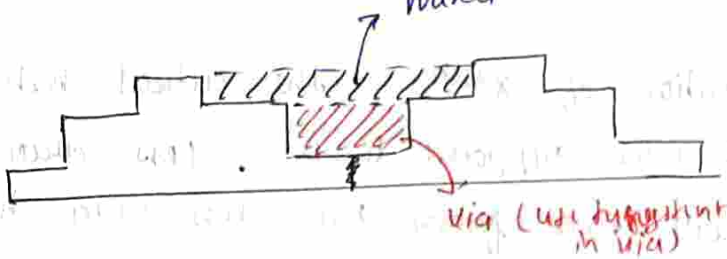
CMP → chemo-mechanical polishing.

* — Copper is more ductile than Aluminium
— Copper has less resistance
— Nowadays Cu is used in IC fabrication.
earlier it was Al.
— When copper, gold & silver when they interact with Si, they create energy states deep inside the band gap



— But the contact from silicon is by Tungsten because Cu can't come in contact with Si.

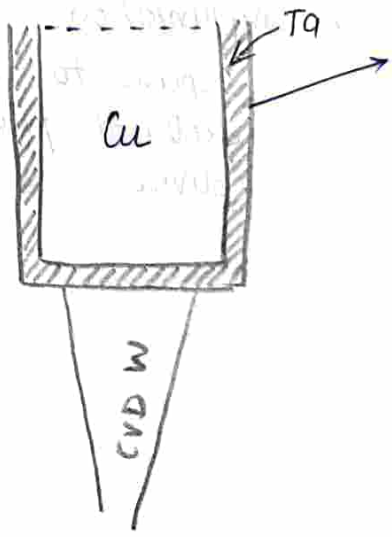
⇒ Dual Damascene process



* Before putting the metal (Cu), we do the etching multiple times.

Because of Cu we can now go to 10-12 layers of metal.

Copper wires



ALD — atomic layer deposition (atom-by-atom)

- For 65 nm technology, the barrier width is say 5nm.
- But for finer geometries if we keep the barrier width same then the effective resistance ↑ significantly.
- So for advanced technologies we are continously looking for advanced barrier materials.

Variability

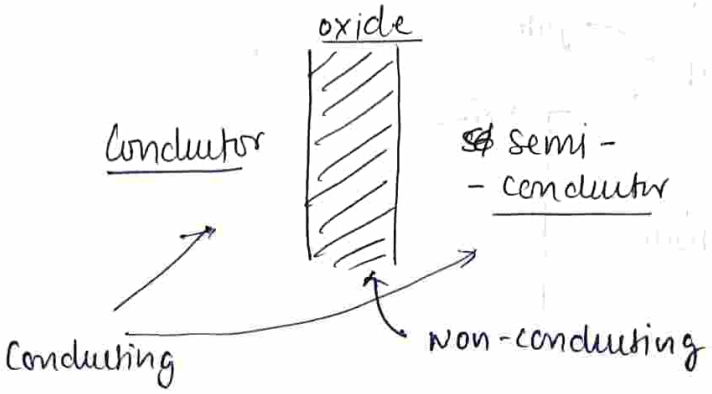
- Device operating in range of nanometer.
- 65nm technology node
  - Gate oxide only 1.2nm (12Å) thick
    - * only 2-3 molecule of SiO_2
    - * Additional molecule leads to 25% change in performance
  - 60nm gate length (LP process)
    - * etching for few additional milliseconds may result in reduction in length by 5nm.
    - * change in performance (speed) by 8%
    - * change in leakage by more than 30%
- 22 nm technology node?
  - Gate oxide replaced by high K-dielectric & metal gate (work-fun^n management)
  - 18nm gate length (LP process)
    - * sensitivity to variation increases further.

* Clean Room
- Liquid cooling
- Self Aligned polysilicon.

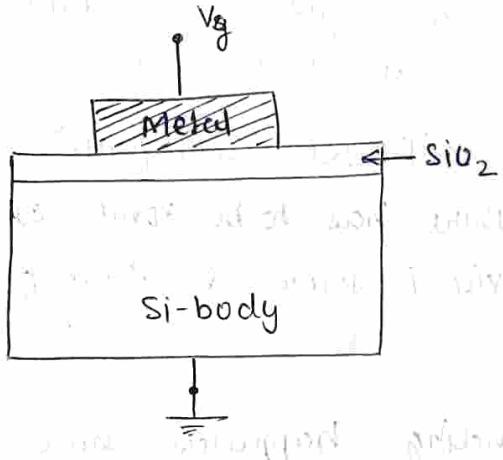
Lec-5

2. VLSI Devices

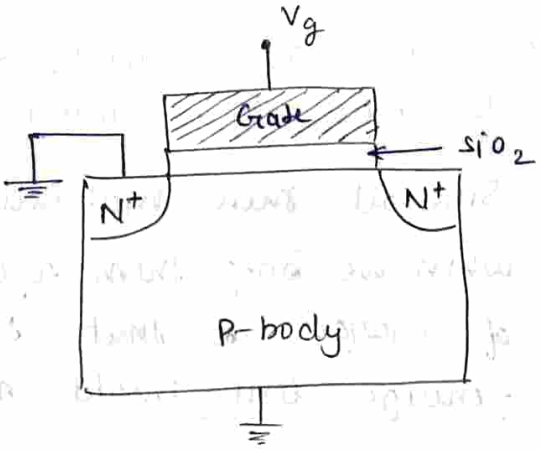
Metal Oxide S/C (MOS)



⇒ Capacitance



"MOS capacitor"



"MOS transistor"

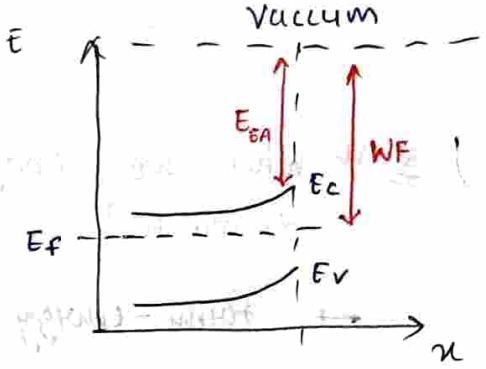
Electron affinity (Work func^n): It is the energy required by the e^- to escape from the material.

Electron affinity: Energy required by e^- to move from CB to vaccum. (E_{EA})

- it doesn't change with doping
- it is material constant (it don't change)

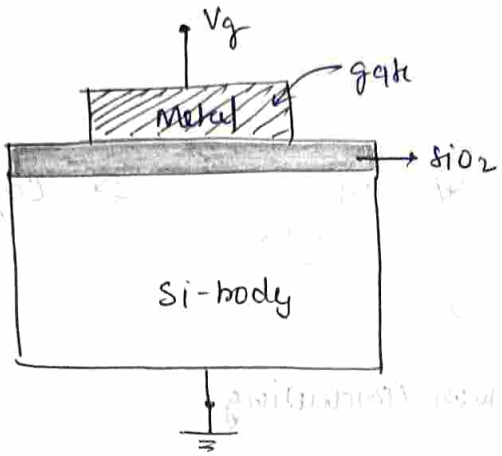
Work func^n (W_F): Energy required by e^- to move from fermi energy level to vacuum

- it changes wrt doping
- it depends on surface termination (Crystal face, surface chemistry).



"energy band diagram of MOSCAP"

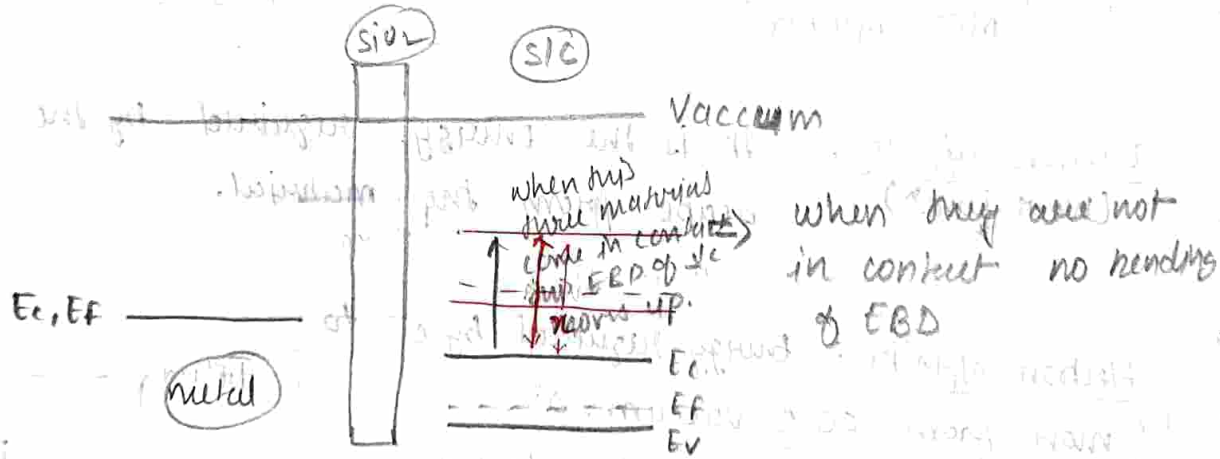
MOS Capacitors



Q. When we bring metal, oxide & s/c together in contact with each other, what happens at their interface?

→ Since all these materials have different work-fun^n, so when we bring them together there has to be some arrangement of charges so that the device is stable & their fermi-energy level should match. (Why?)

* So to match that band bending happens here.



Q. * When we bring this materials together then what needs to match?

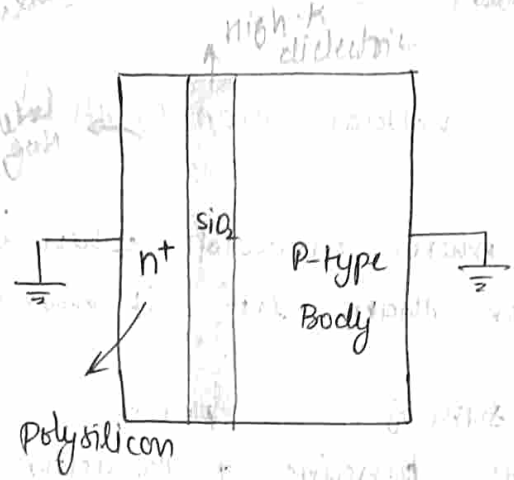
→ fermi-energy level

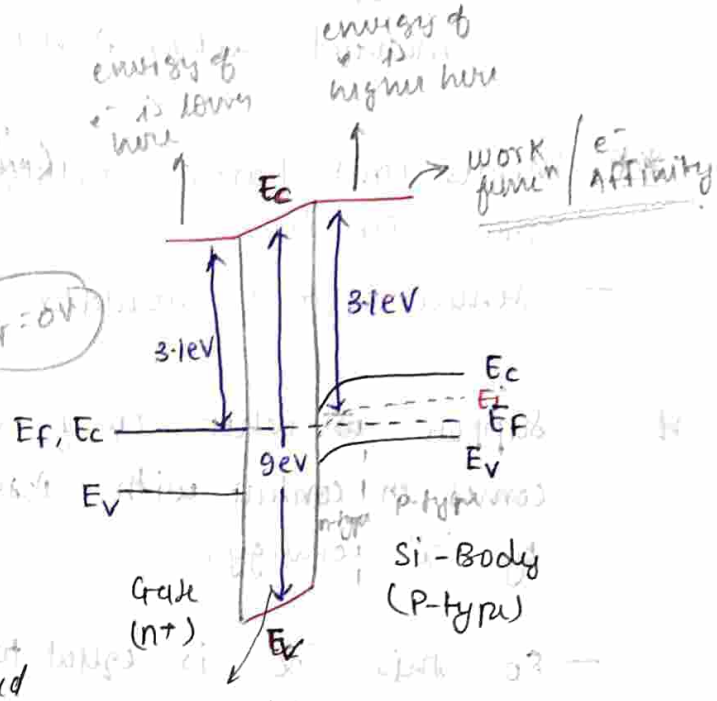
* At equilibrium, E_F for all becomes same. then no e^- will flow

* When we talk about capacitance b/w two metals & capacitance b/w metal - oxide - s/c then both are very different.

— because in first case, when diff. materials come in contact then fermi - level of materials needs to match & for that Band - bending occurs

MOSCAP - EBD





Q. What will happen if n^+ is replaced with p^+.

- n^+ - polysilicon : In this region, the amount of doping is very high, so that in poly - region E_F & E_c both are at same level.

* on increasing the doping in n-type material the fermi level moves toward CB.

$$\uparrow E_{fn} = E_c - kT \ln\left(\frac{N_c}{N_D}\right)$$

→ position of E_F for n-type s/c.

N_D ↑ : no. of dopants in n-type material.

⇨ Fermi - Energy level :- (E_F)

— It tell us the prob. of finding an e^- &

— It tells that 50% of e's are above E_F & 50% of e's are below E_F.

at $T > 0K$

— At $T = 0K$, e's doesn't lie above E_F.

Q. What does the fermi-level in b/w VB & CB, represents?

→ (i) E_f near VB represents, it is a p-type material & prob. of finding an e^- in VB is higher.

(ii) E_f is near CB, it means a n-type material & prob. of finding an e^- in CB is higher

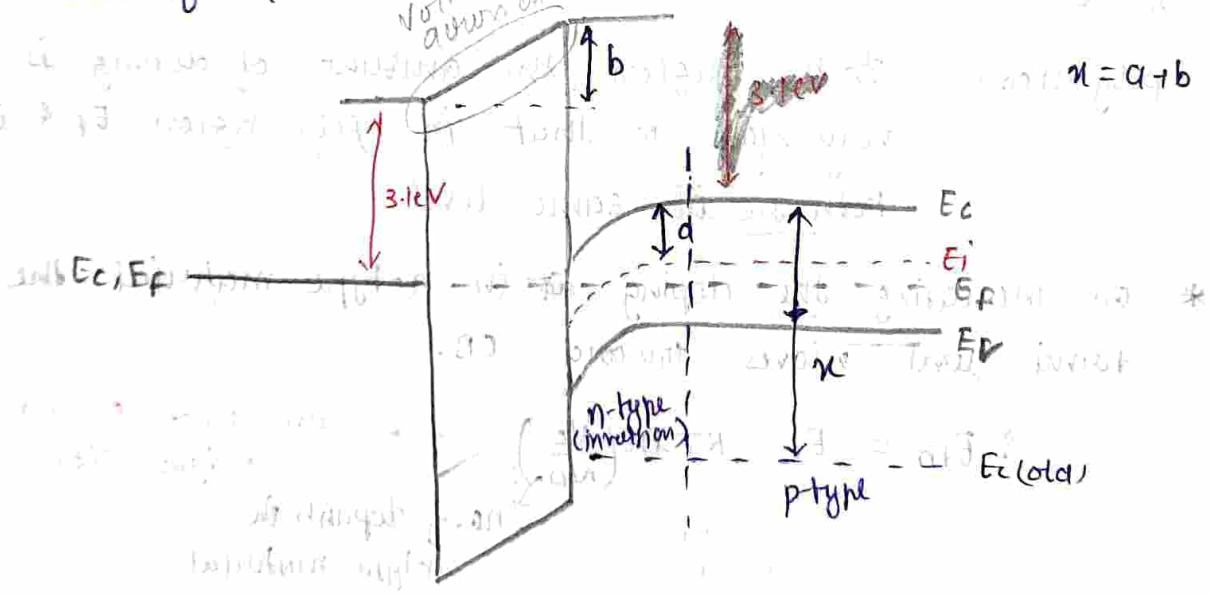
(iii) If E_f is in b/w VB & CB, then it means a intrinsic material, which is not good for conduction.

** Metals can't have a voltage drop across it as the current starts flowing.

– Semiconductor & insulator have voltage drop in them.

* Suppose when CB of s/c moves upward when it comes in contact with insulator than let it moved by 'x' energy.

– So this 'x' is equal to the sum of EBD of bending of insulator & EBD of s/c.



$x = a + b$

* Both the bending is occured due to potential difference b/w the materials.

* It moved upward in such a way that workfunc^n = 3.1eV (remains const) & E_f matches with metal.

** Why both polysilicon & Body have 3.1eV of work fun^n?

→ Because poly is also a kind of Si hence both have same work fun^n & hence we can easily change the work fun^n of the material.

→ Therefore now we use high-K dielectric material, whose workfun^n can be easily changed & V_T of PMOS & NMOS can be varied independently.

* The slope of bending EBD in oxide region, decrease with the increase in thickness of oxide-layer.
[for thinner oxide-layer it will lead to tunneling.]

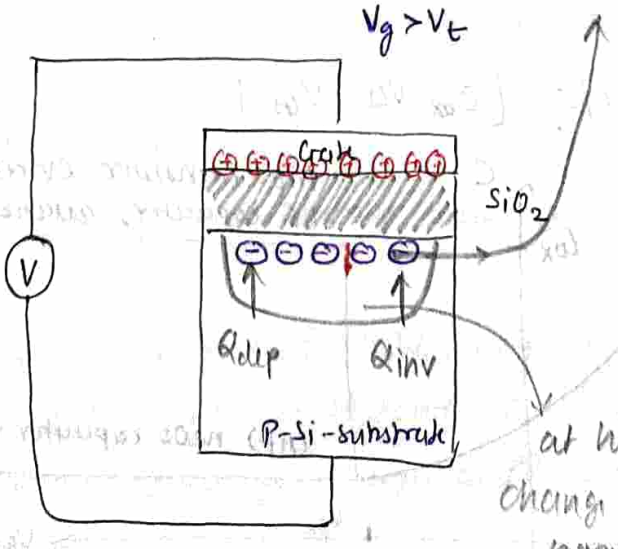
* If we replace dielectric material with high-K dielectric material then Band Bending will also decrease & Bending will change]
Bending is fun^n of dielectric const.

Q. Why we use high-K dielectric? (21:30 – 22:30)

Q. What is the significance of work fun^n if e^- can conduct from CB?

→ Band Bending determines the threshold voltage.

MOSCAP – Characterization :

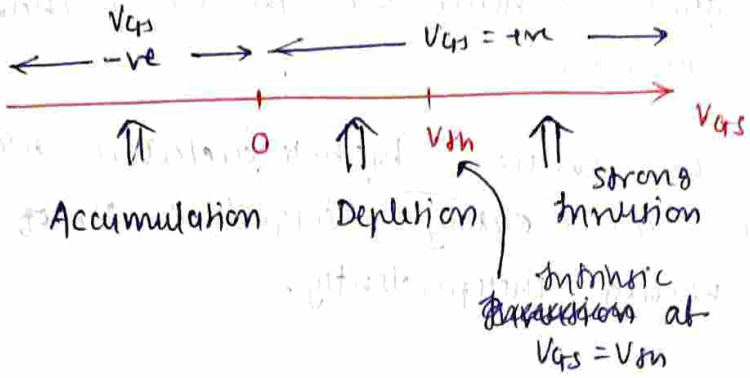


at low freq. change in charge happens over here

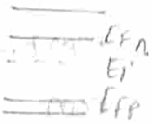
- When we apply & vary voltage V, we observe the change in charge, capacitance & flow of current.
(i.e. wrt V_g)

at high freq. change in charge happens over here.

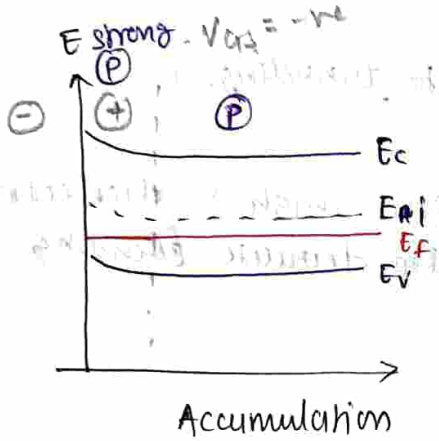
MOSCAP - Region of Operation

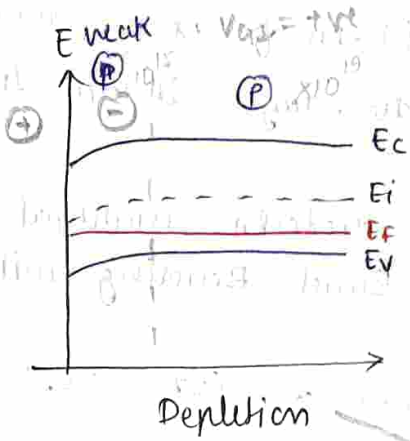


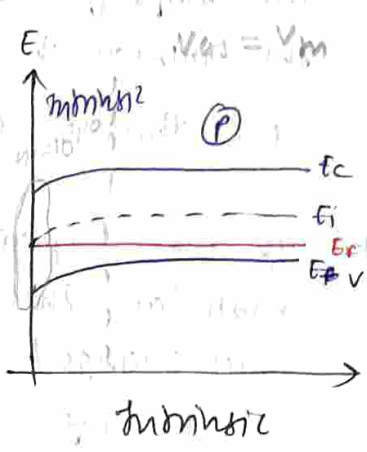
for NMOS



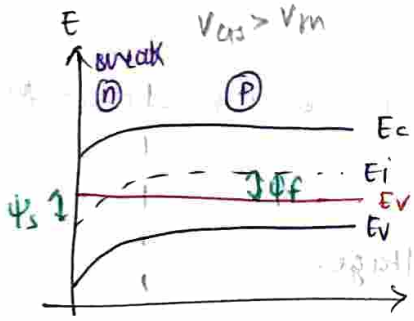
For NMOS :

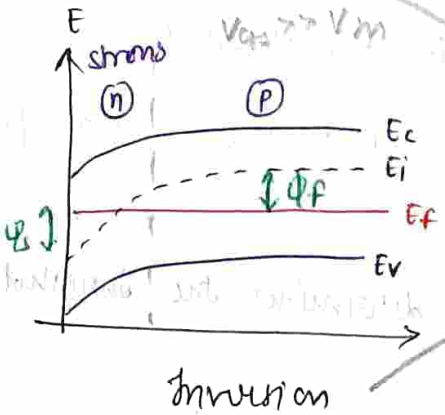






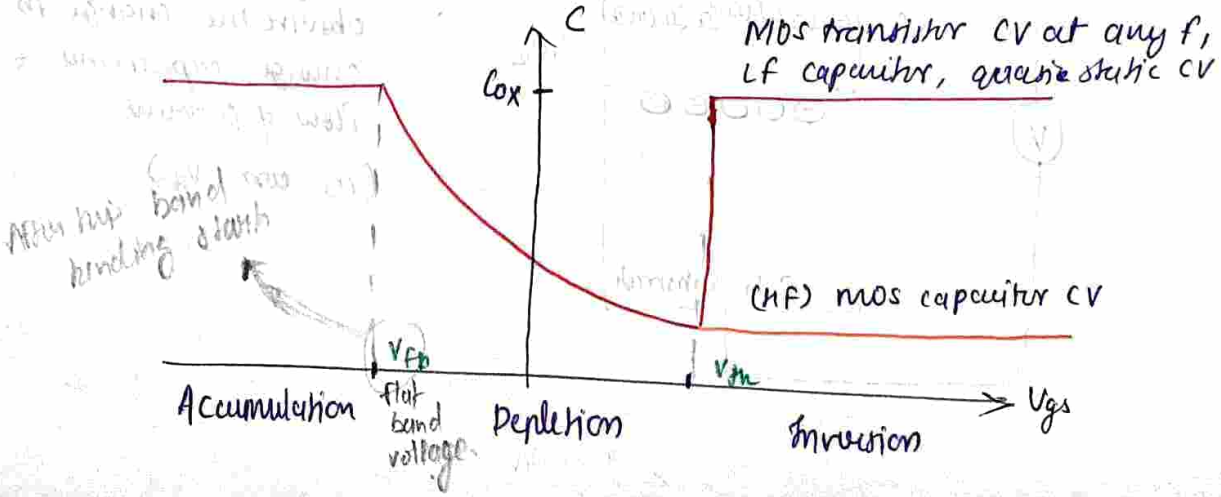
it has immobile ion charges





it has mobile e's + charge ions

Imp **MOSCAP - Transfer c/s** [C vs V_{GS}]



At low freq.

In accumulation region, capacitance is high because the change in charge is happening across the oxide only. (i.e. -ve charge) $V\uparrow$ (in -ve), $Q\uparrow$ (in -ve), $\Rightarrow$ $C \rightarrow$ constant

- As $V_{gs}\uparrow$, in depletion region the depletion layer starts to form hence the capacitance starts decreasing. (due to depletion region effective length of insulator is increased hence $C\downarrow$)
  — Depletion layer means, there is no available charges. Therefore any change in the charge would happen at the boundary of depletion layer. $V\uparrow$, $Q\uparrow$ (very small), $d_{eff}\uparrow$ $\Rightarrow C \rightarrow \downarrow$

$Q = CV$, $C = \frac{Q}{V}$

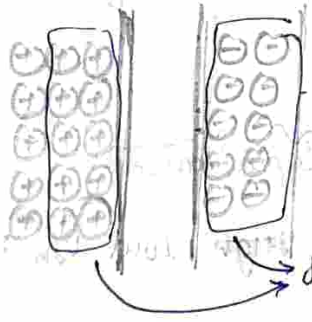
- When inversion layer forms, the capacitance ↑ which ~~occurs~~ due to the addition of additional charges at the interface of silicon & SiO_2.

$V\uparrow$, $Q\uparrow$, t_{eff} = const, $C\uparrow$ (low voltage)

$V\uparrow$, Q (don't ↑ much because of low frequency.

$\Rightarrow$ Above curve (Red curve) is for low frequency.

- After inversion layer is formed, when we increase the V_{gs} further then value of capacitance remains same because on increasing V_{gs} the charge near gate is only increasing & there is no change in the charge amount near si-SiO_2 interface.



$C_{ox} = \frac{Q_{ox}}{V_g}$

then the capacitance will be due to this only.

At high freq

- At very high freq. inversion layer doesn't have any influence on capacitance because it can't change any charges in the inversion layer wrt to change in gate voltage. $V\uparrow$,
  — means inversion is formed here, only depletion is here & depletion width can't go beyond a limit hence for high freq. also capacitance values gets saturated.

** for MOSFET (MOS Transistor), the value of capacitance will remains ~~same~~ exact as that for the LF capacitance of MOSCAP.

* The band bending starts after the applied flat Band voltage (V_{fb}). ~~It is when the +ve value of~~

Q. Why do we use high -K dielectric?

** $\uparrow C = \frac{\varepsilon_0 A}{d \downarrow}$ ($C \uparrow$),

as $d = t_{ox} \downarrow$, "tunneling occurs" → problem.

$\varepsilon \; C \uparrow$

∴ we keep t_{ox} → high & ε_0 → high-K value

$C \uparrow$, coupling ≡ Q (inv) ↑

* if cap. ↑, then coupling of charges ↑, hence more charge on inversion will form for low value of gate voltage.

(slow-slow) SS → $V_{th} \uparrow$

(~~fast~~-fast) ff → low V_{th} → inv. layer.

(typical-typical) tt →

$Q = CV_g$

E_c, E_f ——— Poly.

E_v ———

$\downarrow V_g = \frac{Q}{C \uparrow}$

Q. Why we need high C_g capacitance?

→ At the end we want low V_{th}:

and, $V_{th} = \phi_{MS} + 2\phi_f + \frac{Q_{dep}}{C_{ox}}$

$\phi_{ms} \rightarrow \phi_m - \phi_s$

$\phi_f = \frac{KT}{q} \ln\left(\frac{N_{sub}}{n_i}\right)$

$Q_{dep} = \sqrt{4q\varepsilon_{si}|\phi_f| N_{sub}}$.

C_{ox} → SiO_2 (oxide layer) capacitance ⇒ for $t_{ox} = 20\text{Å}$

$C_{ox} = 17.25\ \text{fF}/\mu\text{m}^2$

Lec-6

MOSFET

Linear Region $\Rightarrow$ $V_{DS} \leq V_{GS} - V_{tn}$

$$I_D = K_n \frac{W}{L} \left[(V_{GS} - V_{tn}) V_{DS} - \frac{V_{DS}^2}{2} \right]$$

where, $K_n = \mu_n C_{ox} = \frac{\mu_n \varepsilon_{ox}}{t_{ox}}$ → process transconductance parameter

Saturation mode $\Rightarrow$ $V_{DS} \geq V_{GS} - V_{tn}$

$$I_D = \frac{K_n}{2} \frac{W}{L} (V_{GS} - V_{tn})^2 (1 + \lambda V_{DS})$$

↑ channel length modulation.

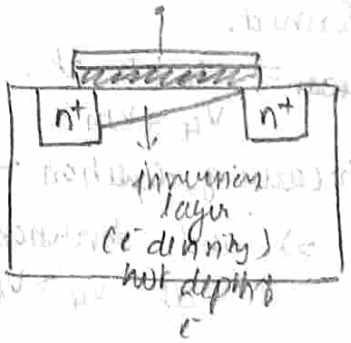
Q. For $V_{DS} = V_{GS} - V_{tn}$, what will be the region of operation?
→ linear.

Q.* ? In saturation region, will the charges in inversion layer also stops increasing as we increase V_{DS} at const. V_{GS}.

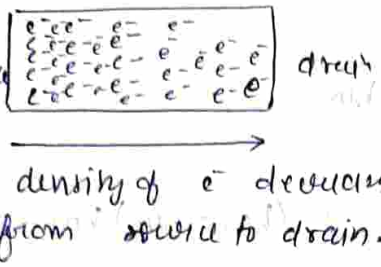
Q. ? For $V_{GS} > V_{tn}$, will the charge in inversion layer change or remains same.

Q.* What is the depth of inversion layer?
→ Inversion is always at the surface, there is no physical depth in inversion layer

Q. ? What is the prob. of finding an e^- in inversion layer?



Q. How to visualize the pinch-off in the sheet of paper (area)?



Q. Why current I_D doesn't become zero after pinch off as there is no e^- near the drain region?

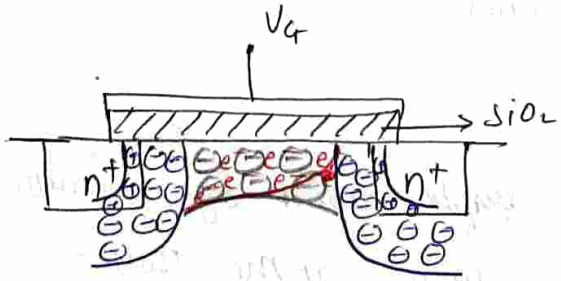
** When you increase V_{DS}, the depletion region b/w n^+-source/drain and substrate will increase as it forms a diode b/w them called a parasitic.

– But the $W_{d,max}$ for depletion region at the ~~depletion~~ interface of SiO_2 - substrate will not increase.

Q. ~~Depletion region~~ Inversion Layer b/w SiO_2 - substrate, there is no width/depth
Depletion region b/w n^+ - substrate, its width ↑
am I right?

"Depletion Region"
"Inversion Layer"

~~Velocity saturation~~



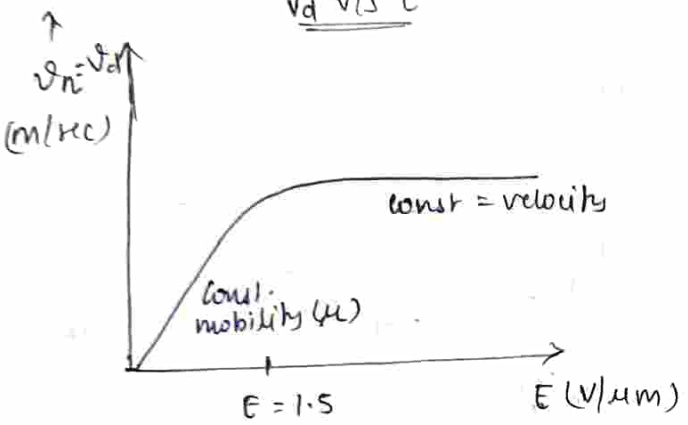
As V_G↑, depletion region b/w SiO_2 - substrate is also formed.

~~drain~~ ⇒ $I_D \neq 0$ at $V_G < V_{th}$ (subthreshold conduction)
because of depletion region b/w n^+-p-type
⇒ and inversion layer is formed at $V_G > V_{th}$.

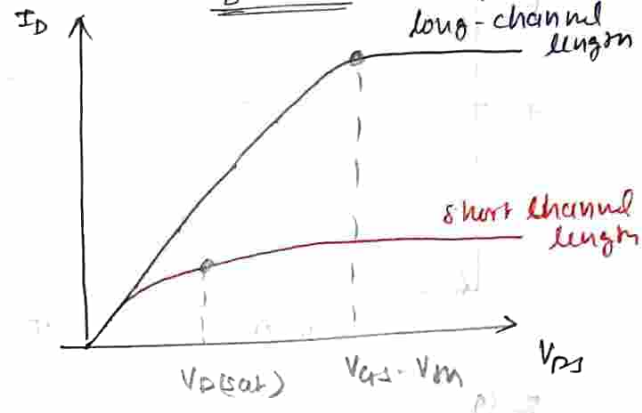
Velocity Saturation

- As $V_{DS}\uparrow$, velocity of $e^-\uparrow$.
  - and pinch-off occurs as we can't observe the e^- (as they are moving very fast at high V_{DS}) near the Drain region.
- But after some V_{DS}, the v_n starts saturating because of the collision of e^- as they are moving at high speed.

v_d v/s E



I_D v/s V_{DS}

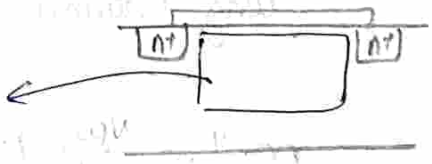


- for long-channel → current saturates at $V_{DS} = V_{GS} - V_{th}$
- for short-channel → current saturates at $V_{DS} = V_{DS(sat)}$

$$v_d = \frac{-eE}{m^*}\tau \quad , \quad \mu = \frac{-e\tau}{m^*}$$

$$\Rightarrow \boxed{\mu = \frac{v_d}{E}}$$

Q. What is the short-channel exactly?



→ when this region no longer remains as rectangular as it has to be then we start to observe short channel effect.

- when depth of depletion region (How?) increases too much as compared to the length of channel then we start observing short-channel effect.

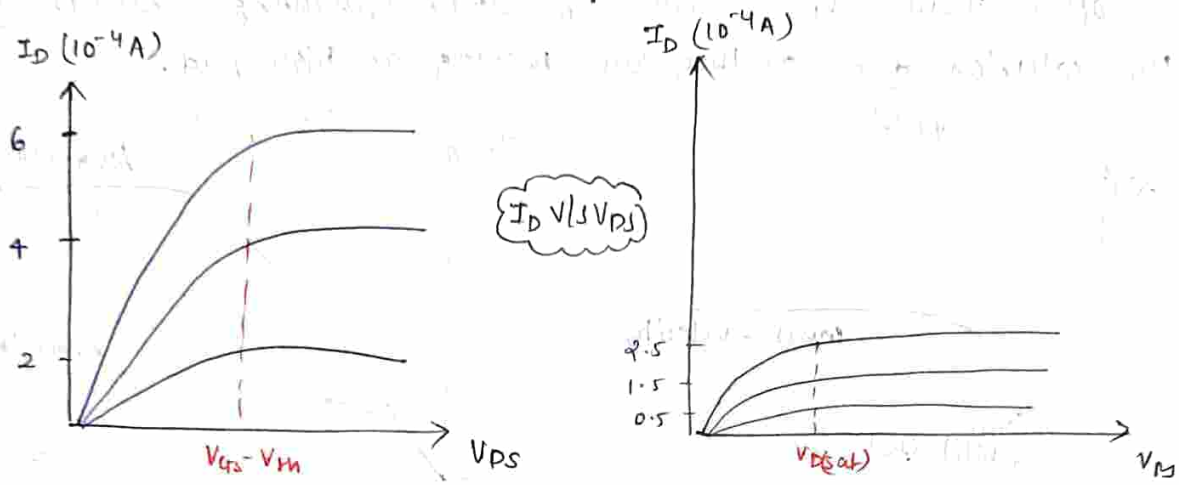
** for long-channel you can dope the substrate more & the depletion width will remain the same (How?) & will not increase

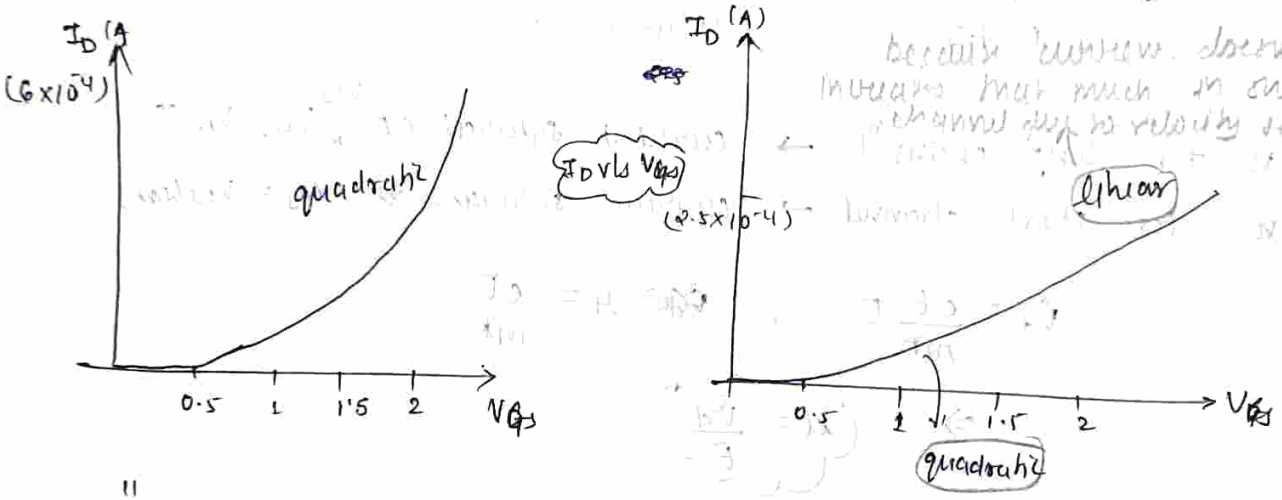
- But for short-channel we can't dope the substrate.

halo implant
super halo implant

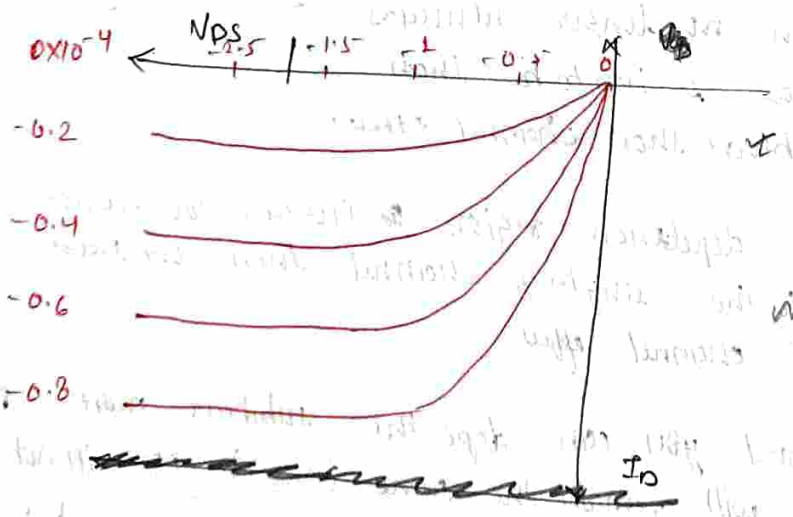
* for short-channel you will reach the saturation at lower value of ext EF.

— while talking about short-channel we only discuss $V_{DS(sat)}$ & not $V_{GS} - V_{TH}$.
pinch off





Because current doesn't increase that much in short channel due to velocity saturation



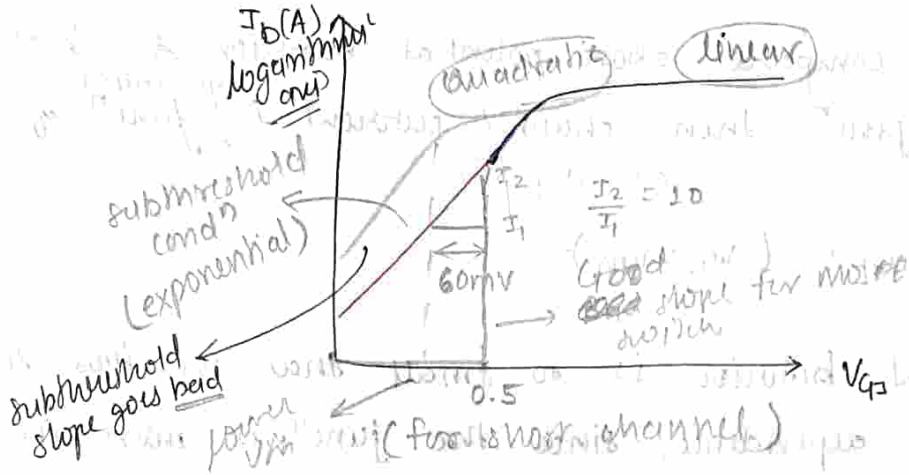
* long-channel is used for high gain in opamp of length more than 150nm.

Imp for interview

Subthreshold Conduction (doubt)

Neamen → Pg - 444
Pedavi → Pg. 24

- Current from source to drain when $V_{GS} < V_m$.



* we can't derive capacitance with the subthreshold cond^n
* current is very low.

- In reality, $V_{GS} \leq V_m$, $I_D \neq 0A$, but I_D is finite & I_D depends on V_{GS} exponentially as

$$I_D = I_0 e^{\frac{V_{GS}}{\eta KT}}$$

$\eta \rightarrow$ non-ideality factor.

- V_{GS} must decrease by approx 80mV for one decade change in I_D.

Eg. $I_{D1} = 1\mu A$ at $V_{GS} = V_m$

$\therefore I_{D2} = 18mA$ at $V_{GS} = 0V$

* due to subthreshold cond^n significant amount of power dissipats.

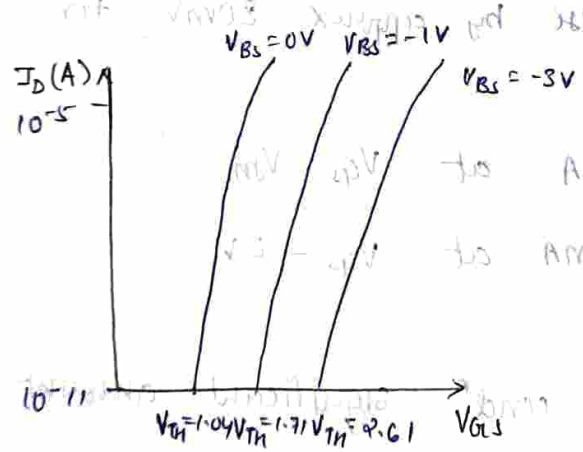
Subthreshold slope →

S is ΔV_{GS} for $I_{D2}/I_{D1} = 10$

$$S = \eta \left(\frac{KT}{q}\right) \ln(10)$$

} Typical value for S: 60 ... 100 mV/decade

- Accumulation & weak inversion case: ($V_{GS} < V_m$)
  - To generate current, e^- must overcome the potential barrier b/w n^+ source & channel.
  - when we compare above potential barrier & barrier b/w p-n junction, then channel current is exponential function of V_{GS}.
- In strong inversion case: ($V_{GS} > V_m$)
  - the potential barrier is so small that we loose the exponential dependence, since the junction is more like an ohmic contact.
- Below graph shows the exponential behavior of the subthreshold current of several body-to-source voltages.
  - And each curve has different V_m values.



* Ideally, a change in gate voltage (V_G) of 60mV produces a change in drain current (I_D) as $I_{D2} = 10 I_{D1}$

Eg: $V_m = 0.3$ (for low voltage operation)

$V_{GS} = 0V$

Drain current decreases by a factor of $10^{0.3/80mV}$

$\approx 10^{3.75}$

$= 5.62 \times 10^3$.

- When we bring M-O-Si together, to make E_f equal, a potential drop is created due to which bending of curve occurs.

* All the transistor behaviour is at the surface of S/C.

- If you decrease the device size, then applied voltage must be decreased. (i.e. we use polysilicon)

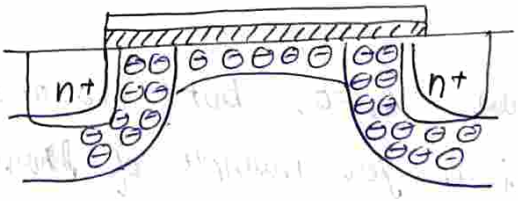
Eg. 2 for $I_{D_1} = 1\mu A$, $V_{GS} = V_{th}$

& when, $V_{GS} = 0V$ (off), $I_{D_2} = 18nA$

Note: If MOS device conducts at $V_{GS} < V_{th}$, then how we define threshold voltage (V_{th})?

— On logarithmic scale, the weak & strong inversion cls and consider their inversion voltages as the threshold volt.

for $V_{GS} < V_{th}$



Q.** What happens when subthreshold slopes goes bad (means here lower subthreshold current). What should be done to avoid this?

→ We want subthreshold slope, such that within 60mV you should be able to decrease current by one decade (↓ 90 ... 100mV)

i.e. $\Delta V_{GS} = 60mV \Rightarrow \frac{I_2}{I_1} = 10$

Q. MOSFET is off at $V_{GS} < V_{TH}$. But because of subthreshold condⁿ how will it be in off state.

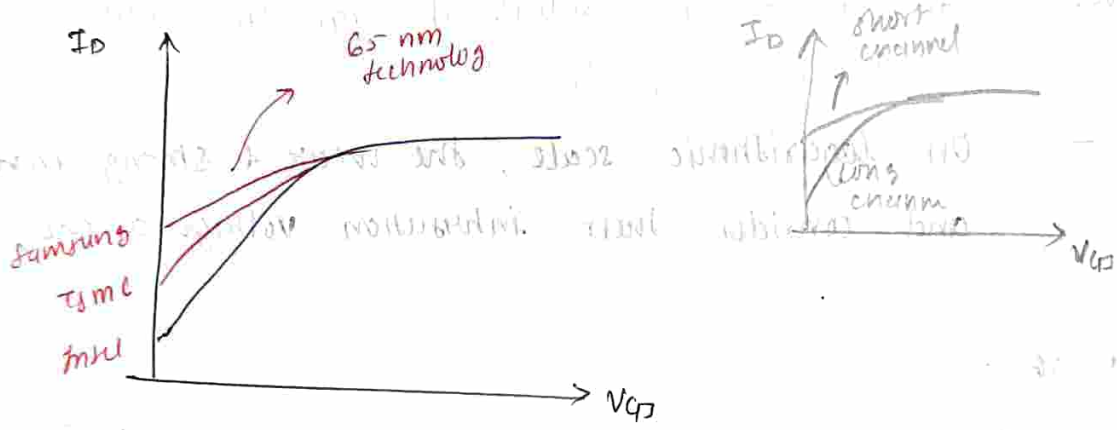
- sub. threshold → used for high gain in Analog cmos
- Normal condⁿ → for switching.

Note: → Leakage increases as you decrease the V_T

→ Leakage also increases when technology is bad & V_{TH} = same. (short channel effect, punch through)

* A technology has ds called subthreshold slope, for diff. technology V_{TH} will change

* When you look at device, you don't only look at the V_{TH} but also at the subthreshold leakage.



- Leakage is '0' when $V_{GS} = 0$, but increases to its full value when V_{GS} is a few multiple of thermal voltage V_T (eg. $V_{GS} > 50mV$)

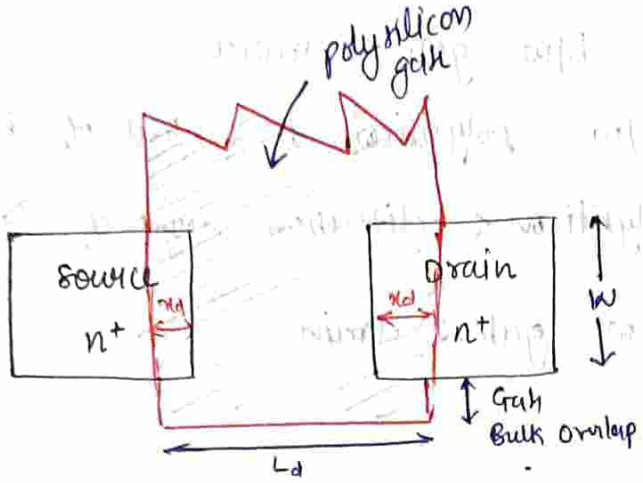
Q. what is subthreshold condⁿ current dependance on V_{DS}?

→ Good slope → sharp o/s (sharp slope of I_D vs V_{GS})] before V_{TH}.
Bad slope → exponential & graph of I_D vs V_{GS}

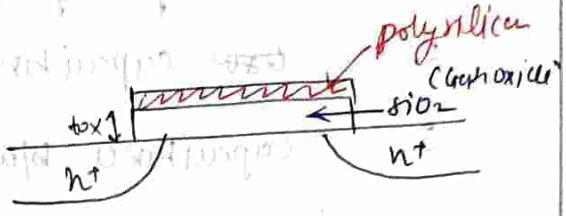
Lec-7

MOSFET Capacitance

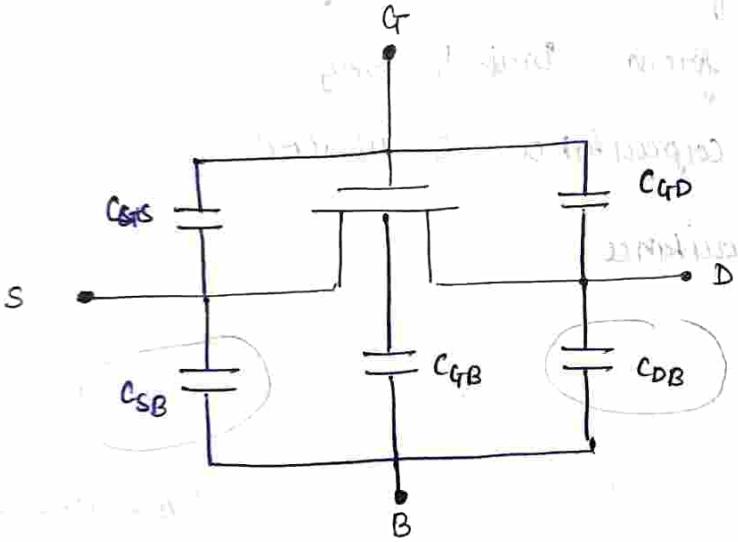
Rabaey - Pg. 109



"Top view"



"cross view"

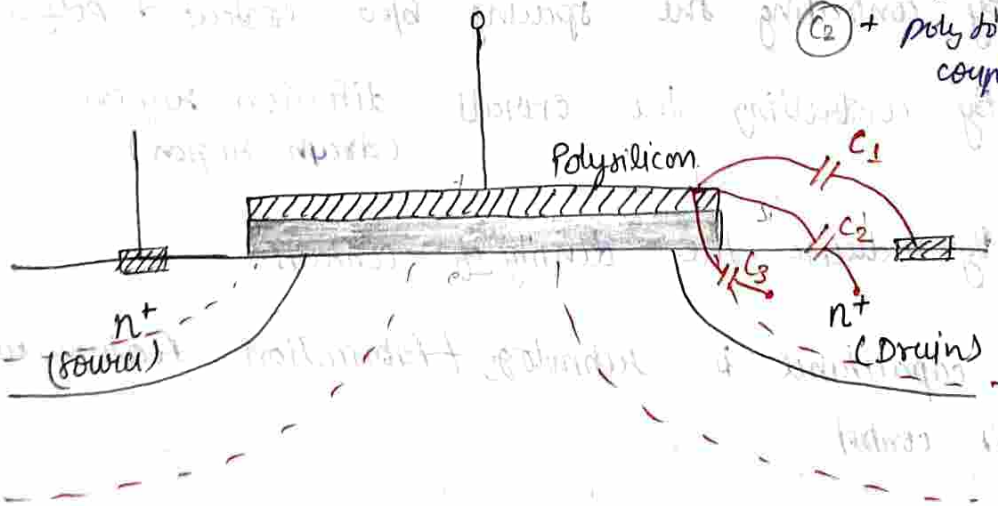


C_{GD} → overlap capacitance of G&D.

(C1) + poly to contact coupling

(C2) + poly to diffusion layer coupling.

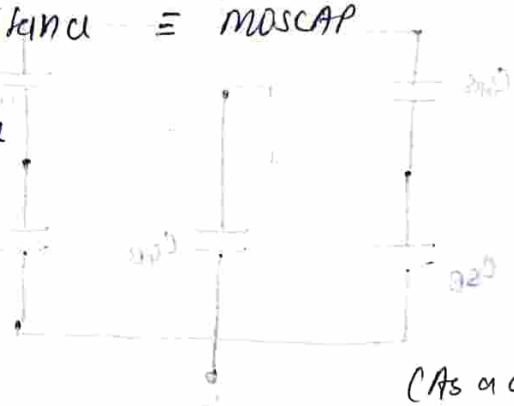
(C3)



Body.

Types of capacitance available are:-

1. Overlap capacitance b/w gate & source
2. capacitance b/w polysilicon & contact of Drain.
3. capacitance b/w polysilicon & n^+ diffusion layer of Drain
4. Overlap cap. b/w gate & drain
5. —
6. —
7. Junc^n capacitance from source to Body C_{SB}
8. Junc^n capacitance from Drain to Body C_{DB}
9. Gate to channel capacitance ≡ MOSCAP
10. Gate to Body capacitance



(As a designer)

Q. How do we reduce gate to drain cap. C_{GD} ?

→ (i) By controlling the spacing b/w contact & poly. (They are placed at minPRE)

(ii) By controlling the overall diffusion region (drain region)

(iii) By reducing the density of contacts.

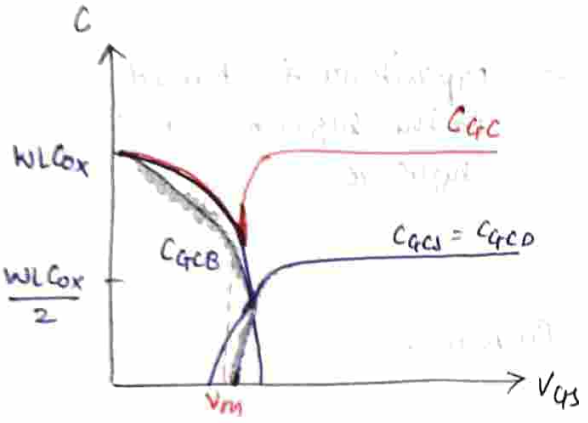
* overlap capacitance ∄ technology / fabrication feature, which you can't control

Q. Why do we have multiple contacts over drain region?

→

MOSFET - Gate capacitance

$$C_{gate} = \frac{\varepsilon_{ox}}{t_{ox}} WL \qquad C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}}$$



C_{GC} → cap. b/w gate & channel

during depletion region:
- C_{GCB} → cap. b/w gate - channel - body

After inversion layer:
- C_{GCD} → cap. b/w gate - channel - drain
- C_{GCS} → cap. b/w gate - channel - source

fig.1 capacitance as a func^n of V_{GS} (with $V_{DS}=0$)

- During depletion region, width of depletion region ↑ as we ↑ V_{GS} hence C_{GCB} ↓.
- During inversion region, a capacitance is formed b/w G-C-S & G-C-D because of inversion layer formation.

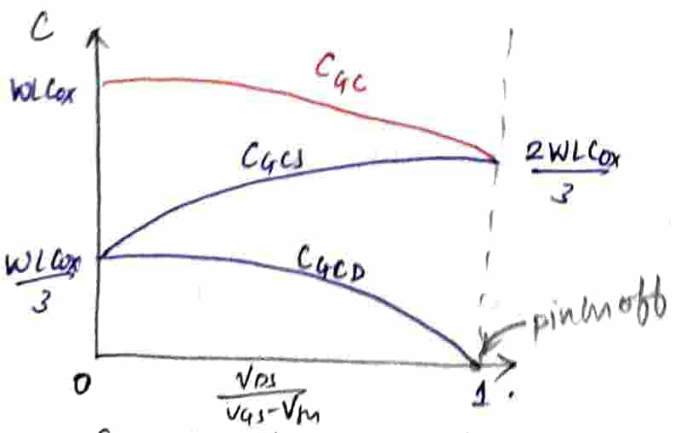
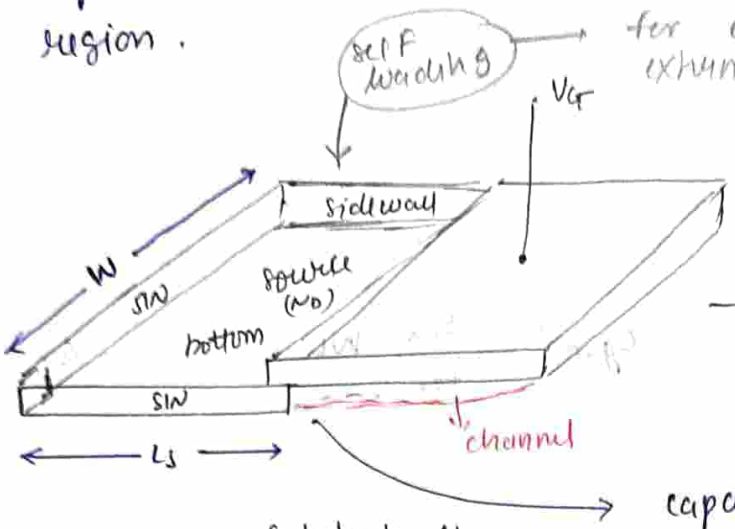


fig.2 capacitance as a func^n of degree of saturation

- C_{GCS} ↑ & C_{GCD} ↓ because ~~pinchoff has occured & drain is not connected to the channel~~ on ↑ V_{DS}, it is going towards pinch off & after some time pinch off occurs & there will be no C_{GCD} capacitance.

MOSFET - Diffusion capacitance

- Diffusion capacitance is formed whenever there is ~~diffusion~~ depletion region.

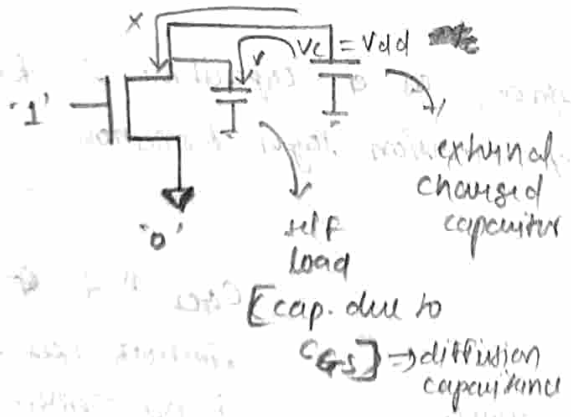


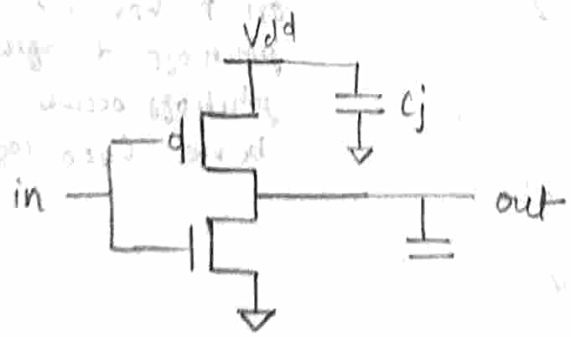
$$C_{diff} = C_{bottom} + C_{sw}$$

$$= C_j \times Area + C_{jsw} \times Perimeter$$

$$= C_j L_S W + C_{jsw}(2L_S + W)$$

⇒ Diff capacitance acts as self load when you want to discharge external capacitance.





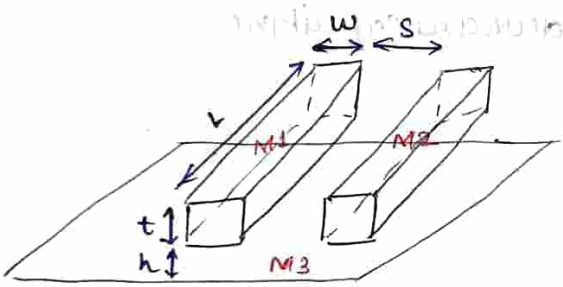
VLSI Devices - Wires

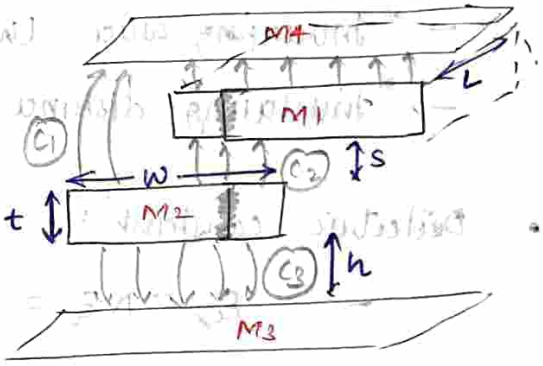
- For making interconnects in integrated ckts we use wires & wires are generally of Cu or Al.
  - Nowadays Cu is mostly used.

VDF

1. Wire Geometry and Capacitance

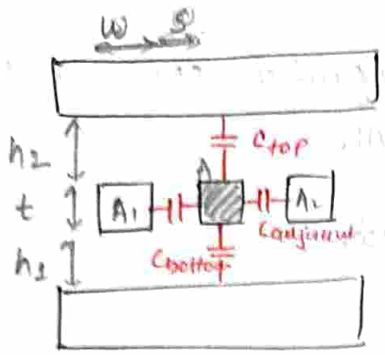
- Pitch = w + s
- Aspect ratio : $AR = \frac{t}{w}$ (Ratio of thickness & width)
  - → Old process had $AR << 1$
  - → Modern process have $AR \simeq 2$

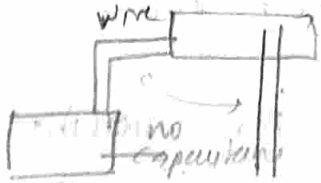




- Between diff. metal layers there is oxide layer of SiO_2. because of which diff. metal layers shows capacitance to each other.
- Different metal layers are connected with via when we want to transfer data b/w them hence no capacitance is formed.
- But if M1 → carries clk pulse
  M2 → carries data signal

  so we don't connect them with vias hence a capacitance is observed b/w them because of SiO_2 b/w them.





$[C_{total} = C_{top} + C_{adj} + C_{bot}]$

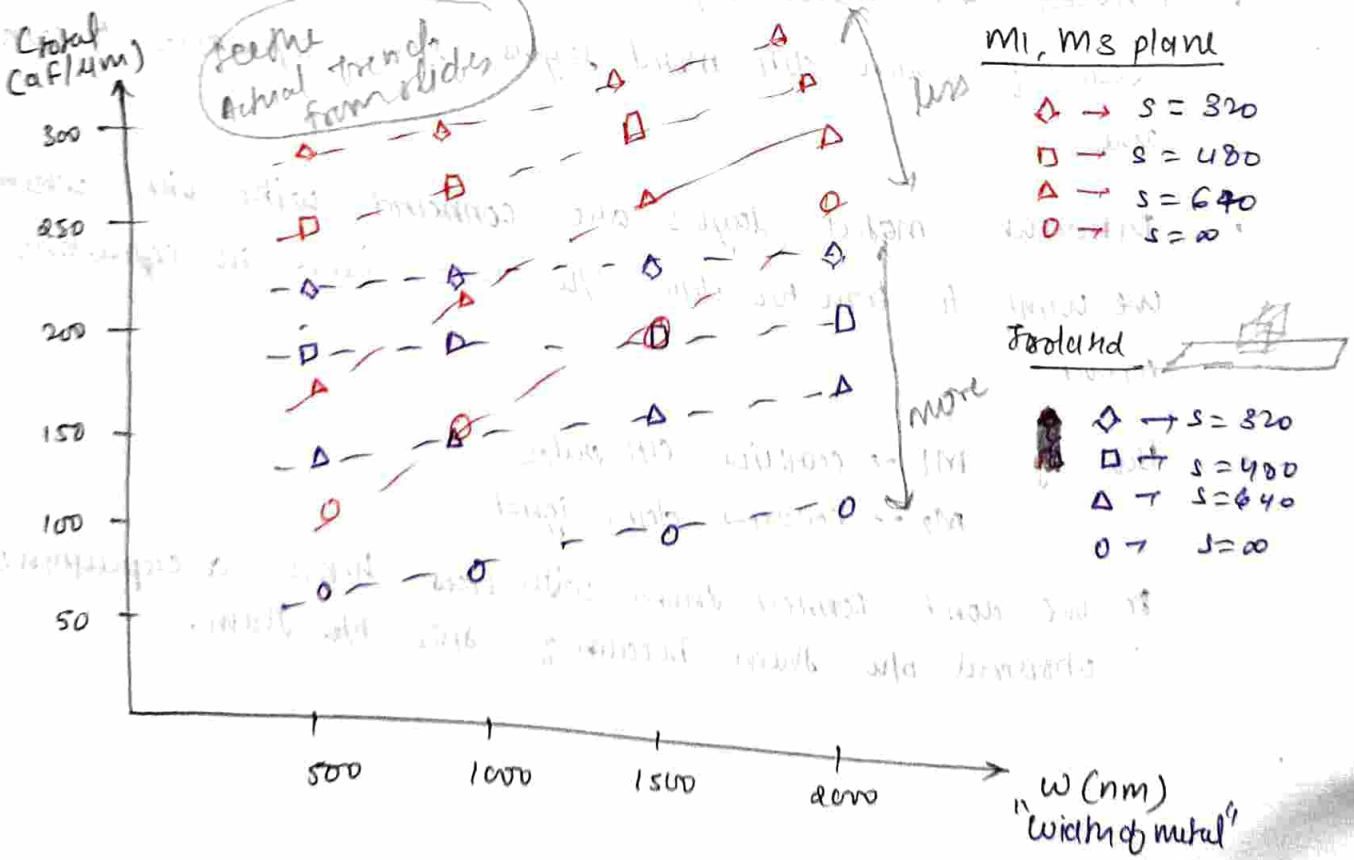
Q. How to decrease the cap of shaded box in Middle

→ (i) Increase distance of A_1 & A_2 and A_1 & A_3

(ii) By not connecting the wire over A, so there will be no effect of capacitance.

Capacitance Trend:

- Parallel plate equation : $C = \frac{\varepsilon_0 A}{d}$
  - Wires are not parallel but they obey trend.
  - * Increasing area (W, t) increases capacitance $C \propto W, t$
  - * Increasing distance (S, h) decreases capacitance $C \propto \frac{1}{S, h}$
- Dielectric constant :
  - $\varepsilon_{ox} = K\varepsilon_0 = \varepsilon_r \varepsilon_0$
  - $\varepsilon_0 = 8.85 \times 10^{-14}$ F/cm
  - $K = \varepsilon_{Si} = 3.9$ (for SiO_2)



(a) ~~For each 'w' → width of wire~~,

(b) ~~For isolated metal :-~~

(*) Typical diffn wire has $\approx 0.2\,fF/\mu m$

(~~Compared to 1-2 fF/μm of~~ gate capacitance)

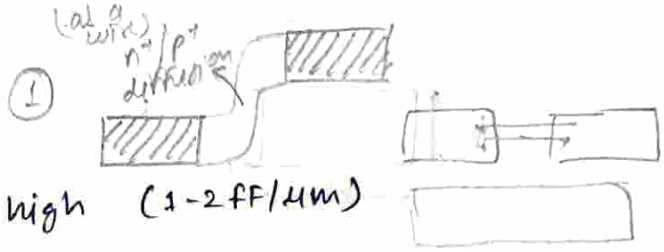
(i) For isolated metal, for constant 'w' → width of wire, when we increase 's' → distance b/w adjacent metal, then change in capacitance is very high.

— But for M1, M3 plane the change is low because of adjacent metal capacitance may have.

(ii) ~~For isolated metal~~, w → increases, s → same
then increase in capacitance is more in M1, M3 plane as compared to isolated plane.

(iii) for small value of w, the capacitance don't have much of difference in case of both M1, M2 plane & isolated plane.

2. Diffusion & Polysilicon

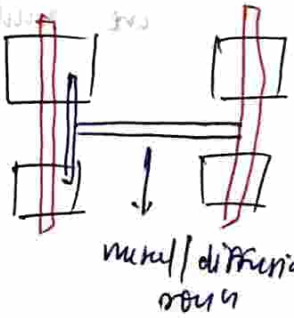


(1-2 fF/μm)

- Diffusion capacitance is very high
  - similar to gate capacitance
  - Diffusion also have high resistance
  - Avoid using diffusion runners for wires
    [We don't use diffusion to make more capacitance]
  - (?) But if we use diffusion then we put lots of contacts over it, put lots of wire over it, metal over it so that resistance reduces

Eg. ** Suppose we want to connect two inverters & we have two option first through metal route & another through diffusion route.

— we prefer metal route because it provides less capacitance but sometimes metal is not available for data transmission ∴ we use diffusion runners. (Active regions which can transmit data)

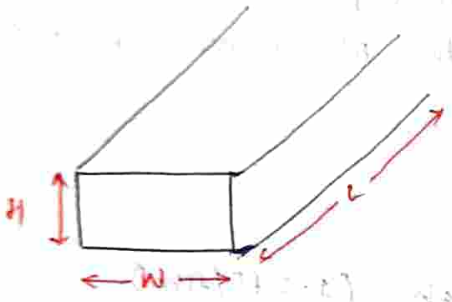


- Types of wires:
  - metal wires
  - Polysilicon wire (for very short distance)
  - diffusion routes

Q: What is active region, for transmitting signals?

→ Active region ≡ OD ≡ diffusion region
≡ Region of source & drain ≡ Green region

3. Wire : Resistance



$$R = \frac{\rho L}{HW}$$

↳ sheet resistance (R_0) [Ω/square]

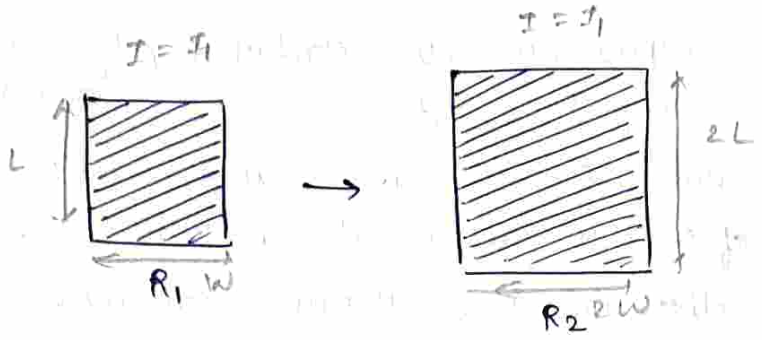
* As a designer we don't have control over the ~~hgt~~ height H of the wire, it is a technology feature.

* $\frac{\rho}{H}$ → constant ∴ total resistance depends on $\frac{L}{W}$.

and $\frac{L}{W}$ says that the no. of squares that we can make in direction of flow of current

* Unit of sheet resistance $R_0 = \frac{\rho}{H}$ → Ω/square

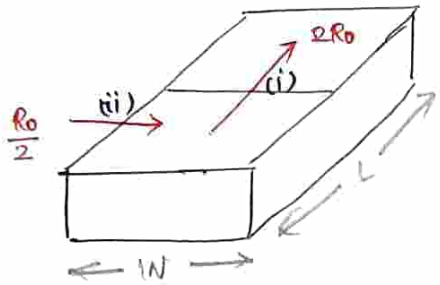
∴ For calculating the ~~no.~~ resistance of material we multiple sheet resistance with no. of squares.



Unit of sheet resistance (R_0) → ohm/square

(or) $\Omega/\square$

- I will simply multiply the no. of sheets/squares in the direction of flow of current with the sheet resistance to get total resistance in the direction of current.



fixed

(i) $R = R_0 \frac{L}{W} = R_0 . (2)$

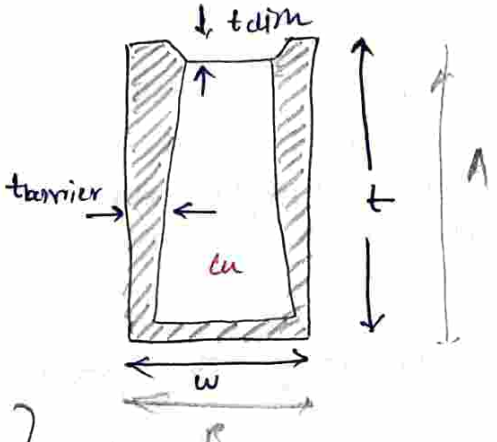
$= 2R_0$

(ii) $R = R_0 \left(\frac{W}{L}\right) = R_0 \left(\frac{1}{2}\right)$

$= \frac{R_0}{2}$

Cu-wire the barrier

- Cu-wire diffusion barrier has high resistance
- Cu is also prone to dishing during polishing
- Effective resistance is higher



$$\left\{ R = \underbrace{\frac{\rho}{(t - t_{dish} - t_{barrier})}}_{A} \cdot \underbrace{\frac{l}{(w - 2t_{barrier})}}_{B} \right\}$$

Q. For decreasing the capacitance b/w contact & poly, we can make our contact away from poly (as d↑, c↓). But for DRC it says that for OD also - it has to be covered by the minimum of outside the contact, therefore we shift the OD by which diffusion ↑ & diffusion capacitance comes into the picture. So is there a trade of we can't do this?

— By decreasing the diffusion ~~region~~ area overall capacitance decreases because the cap. due to poly - contact doesnot get affected as it is always butted at the min - DRC distance. hence not affecting the cap due to poly - contact.

** Jab poly ke upper se metal jata hai to capacitance add hota hai but uske karan phele se hi bahut zyada capacitance tha kaafi tha.

Q. Elmore delay numericals on Mid-Sem & End-Sem.

→ solve from Weste. & Question Bank.

Q. For decreasing the capacitance b/w contact & poly, we can make our contact away from poly (as d↑, c↓). But for DRC it says that for OD also - it has to be covered by the minimum of outside the contact, therefore we shift the OD by which diffusion ↑ & diffusion capacitance comes into the picture. So is there a trade of we can't do this?

— By decreasing the diffusion ~~region~~ area overall capacitance decreases because the cap. due to poly - contact doesn't get affected as it is always butted at the min - DRC distance. hence not affecting the cap due to poly - contact.

** Jab poly ke upper se metal jata hai to capacitance add hota hai but uske karan phele se hi bahut zyada capacitance tha hua karta tha.

Q. Elmore delay numericals on Mid-Sem & End-Sem
— solve from Weste. & Question Bank.

Q. For decreasing the capacitance b/w contact & poly, we can make our contact away from poly (as d↑, C↓). But for DRC it says that for OD also - it has to be covered by the minimum of outside the contact, therefore we shift the OD by which diffusion ↑ & diffusion capacitance comes into the picture. So is there a trade of we can't do this?

— By decreasing the diffusion ~~region~~ area overall capacitance decreases because the cap. due to poly - contact doesn't get affected as it is always butted at the min - DRC distance. hence not affecting the cap due to poly - contact.

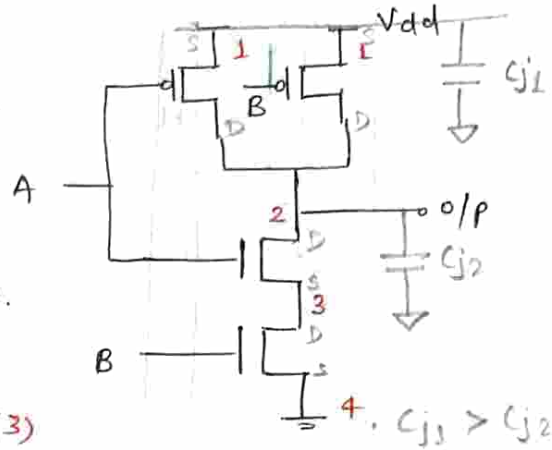
** Jab poly ke upper se metal jata hai to capacitance add hota hai but uske karan phle se hi bahut zyada capacitance tha kuch kha tha.

Q. Elmore delay numericals on Mid-Sem & End-Sem
→ solve from Weste. & Question Bank.

Lec-8

Standard cell layout Design

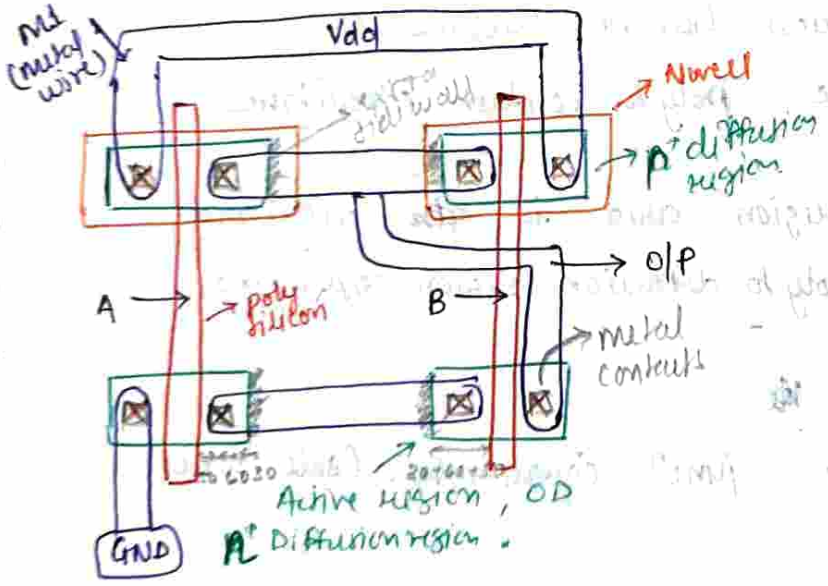
Layout of NAND gate



- for pmos, one of the diffusion region ~~is~~ from of both pmos are connected to Vdd. (1)
- And other diffusion region of both pmos are connected to o/p. (2)
- for nmos, one of the diffusion (3) region is ~~is~~ shorted (or directly connected)
  - and one of nmos diff. is connected to o/p. (2)
  - and another of nmos is connected to GND. (4)

$C_{j1} > C_{j2} \Rightarrow (b)$

$C_{j1} < C_{j2} \Rightarrow (a)$



* If we make NAND gate using this method then overall capacitance will increase

Total diffusion area = 280 um

(i) No. of metal contacts = 8

- ~~met~~poly to contact capacitance is due to this 8 metal contacts.

(ii) Total 8 source & drain capacitance

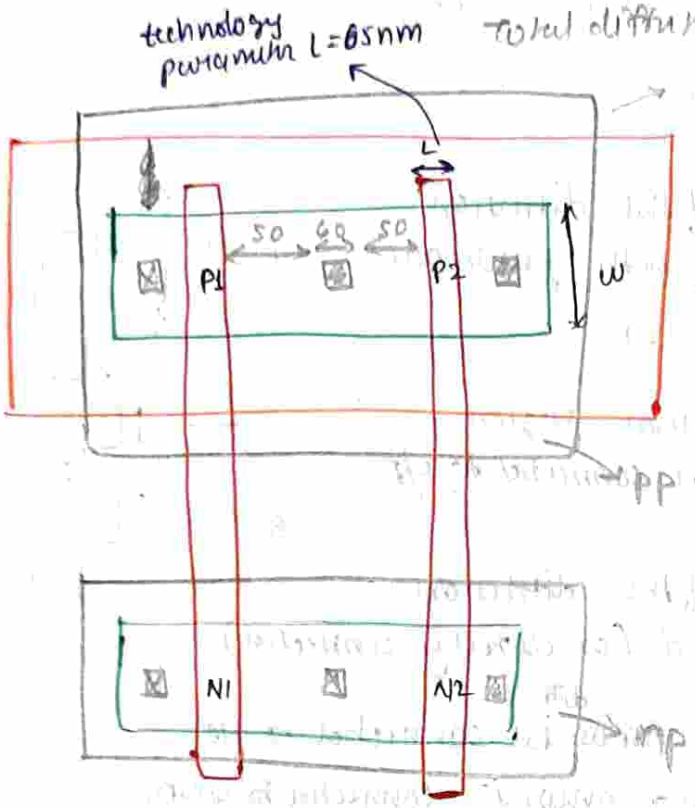
∴ Overall area for the active (diffusion region is increased

- capacitance due to ~~junction depletion region is also increased.~~ metal contact & diffusion region is increased.

(iii) Here it has extra side walls because of which junc^n capacitance b/w diffusion region & substrate increase.

* Any of the poly region can be drain any can be ~~drain~~ source.

- Therefore we use the below standard layout for NAND gate



w/o metal wire connection

Total diffusion length = 50+60+50 = 160

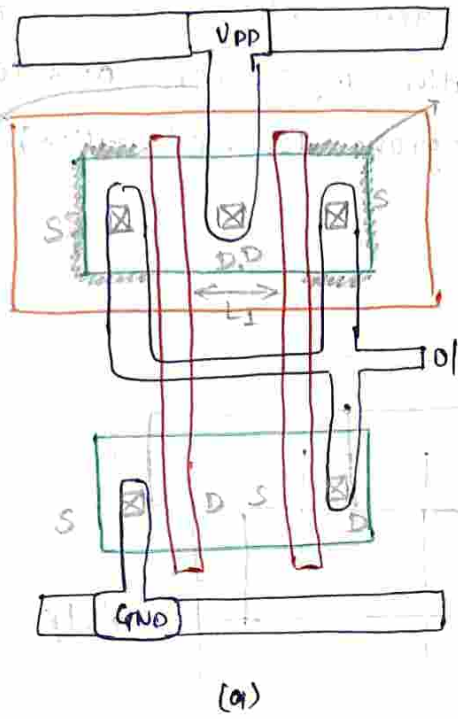
(i) Here no. of metal contacts has be reduces to 6.
- Hence decreasing the poly to contact capacitances.

(ii) Also the diffusion region area is decreased
- Decreasing the poly to diffusion region capacitances

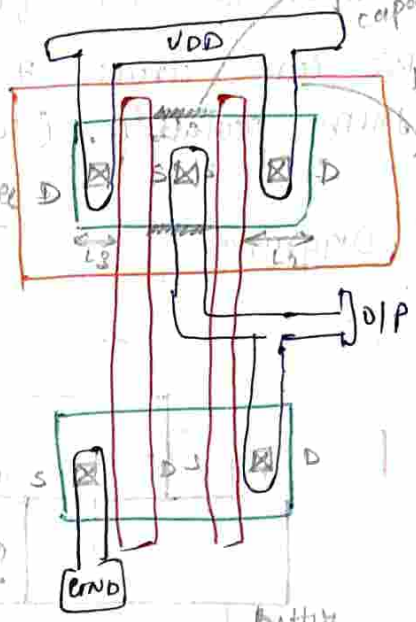
(iii) No extra side walls
- Hence decreases the junc^n capacitances. (side wall capacitances)

Q. (2) In the above layout which one should be o/p & which one should be Vdd for pmos. (Tell all the metal connections).

→ Electrically there is no diffusion but we have to see it wot capacitance



(a)



(b) Better

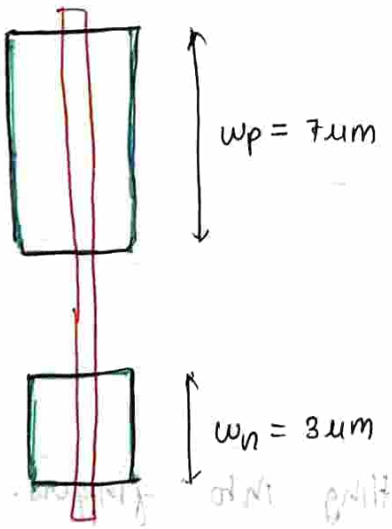
- If we go by fig. (a) then more metals will be used in the o/p resulting in the increase in capacitance at the o/p.

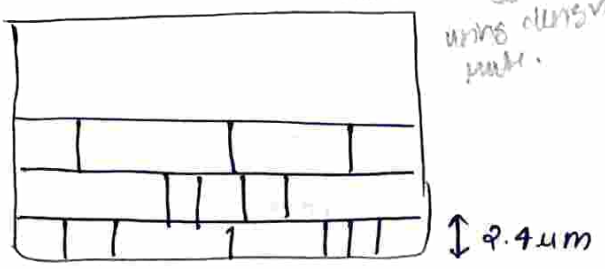
Part-2

* While drawing the each layer in our layout we have to follow design rules that me (i) minimum ~~con~~ width of contact, metal & poly (ii) min area of OD, Nwell, PP, NP must be satisfied (iii) spacing b/w the contact & poly should be more.

Q. Why we use NTAP & PTAP ?

Q. How to decide the height of standard cell using design rule.





"standard cell layout"

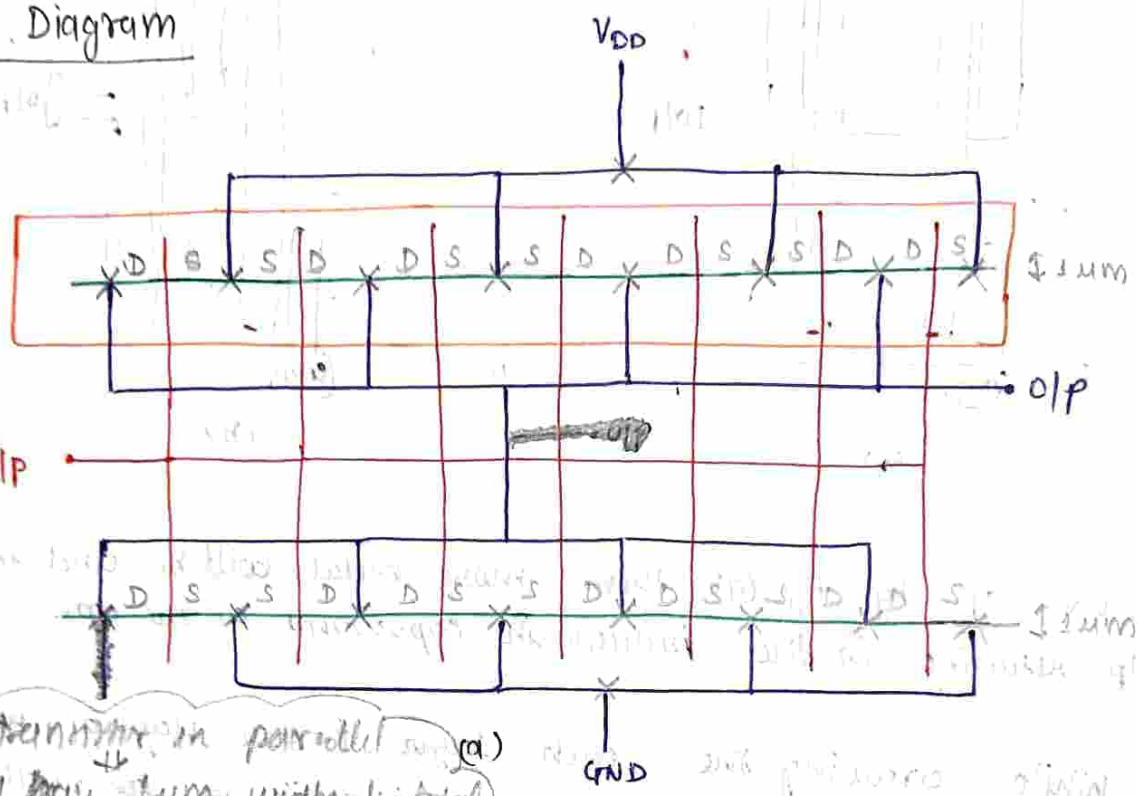
Q. $W_p = 7\mu m$ & $W_n = 3\mu m$ in our cmos but the height constraint ~~for~~ is $2.4\mu m$ then how do we draw this cmos under $2.4\mu m$?

→ explanation is shown in stick diagram.

Q. How to draw the pmos of $w_p = 7\mu m$ into a height of $1\mu m$.

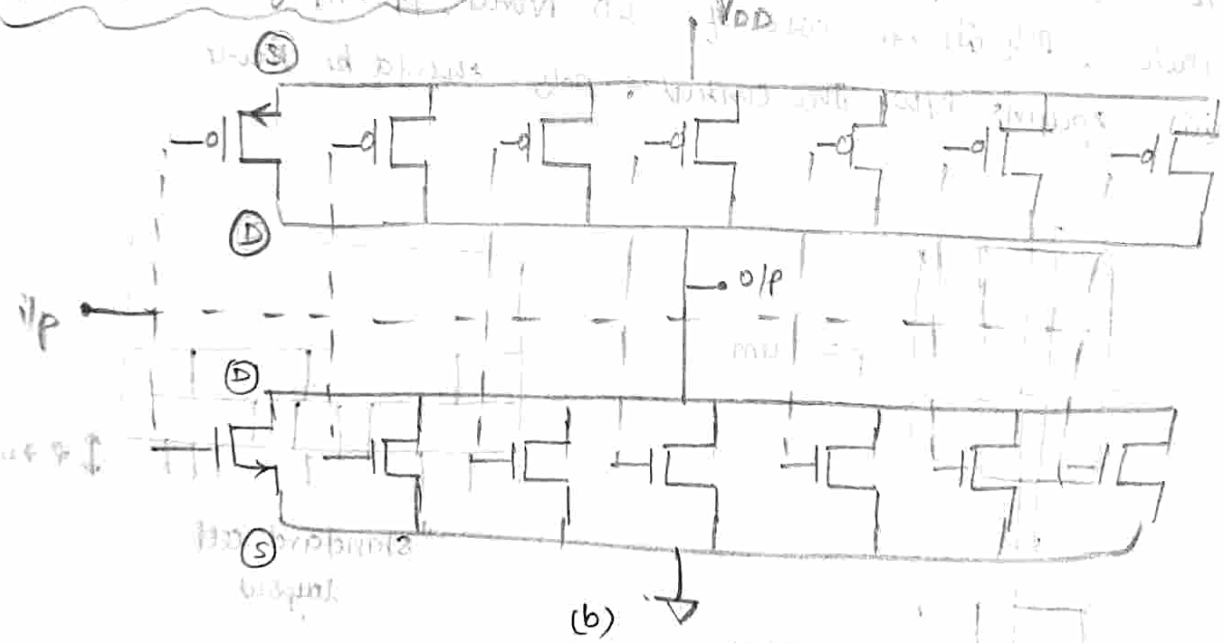
→ We can draw 7 pmos of width $w_p = 1\mu m$ and connect them parallelly. (see the stick diagram for explanation)

Stick Diagram



(a)

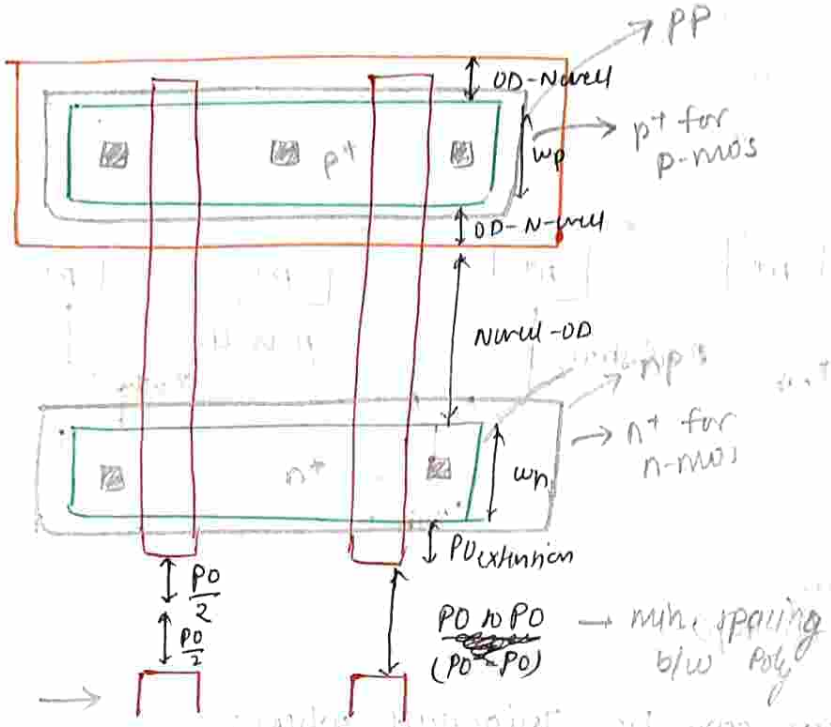
when transistors in parallel are drawn, all have 1um width but have a width of 7um.



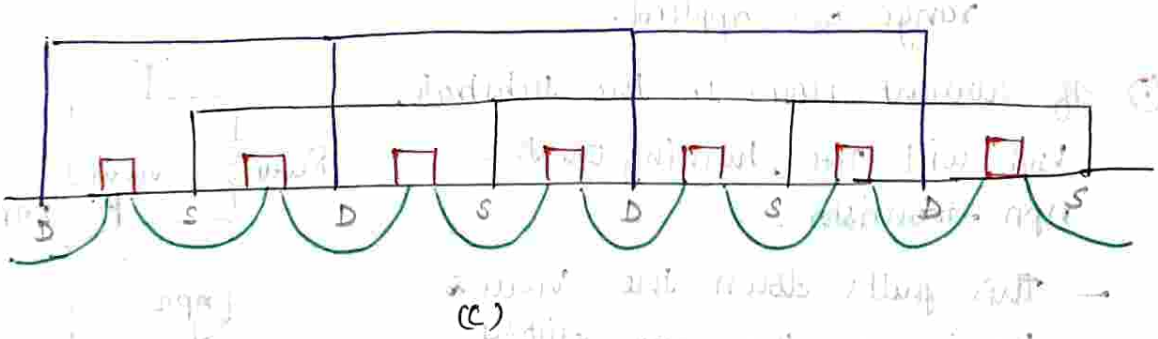
(b)

- Above process is called splitting into fingers.
- When we do the fingers then area increases because we have to constraint the height.

Q. How calculate the height of given CMOS.



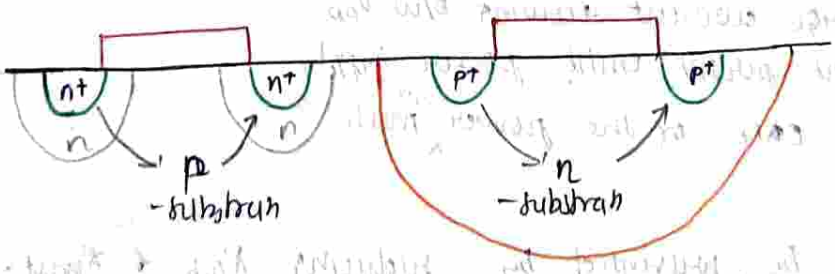
Height = (OD-Nwell) + (wp) + (OD-NW) + (NW-OD)
+ (wn) + (PO$_{ext}$) + (PO-PO)



(c)

Part-3

Q. What is latch up? And how do we ~~also~~ avoid them?

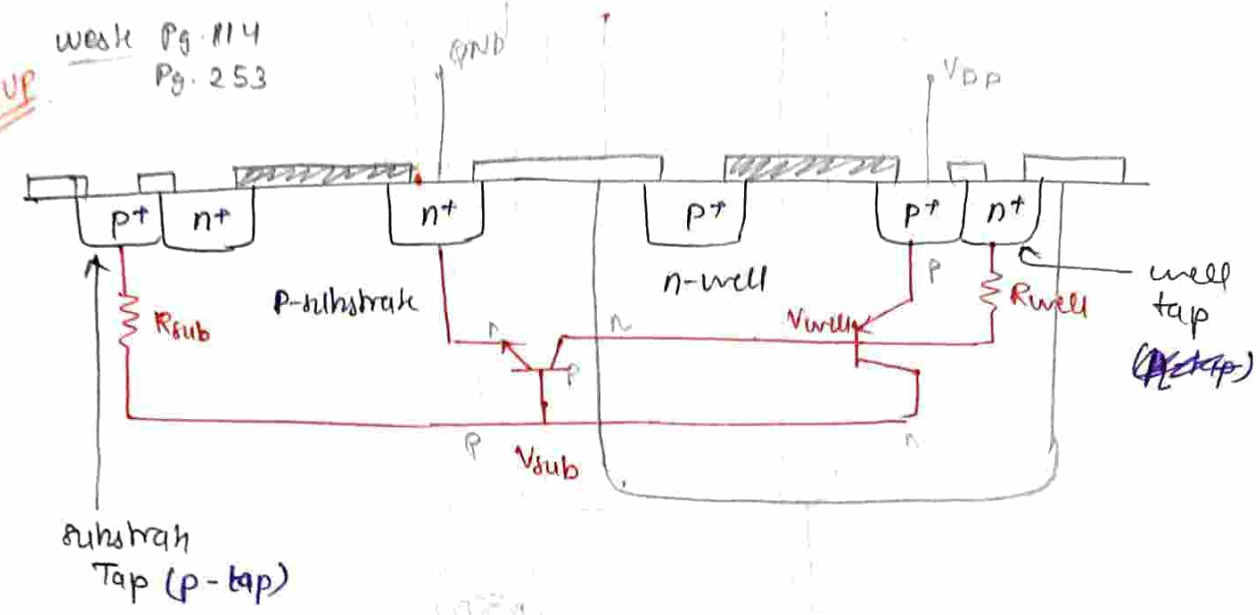


- npn+ & pnp+ kind of BJT is observed here when resistance of substrate is high.

- So a ckt is made with a +ve f/B of npn & pnp xsistors. [Latch-Up]

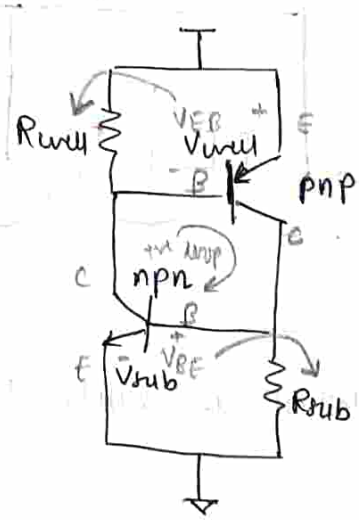
Latch UP

week Pg 114
Pg. 253



- Latch up can be triggered when:
  (i) when transient current flows through the substrate during normal chip power up.
  (ii) when external voltages outside the normal operating range are applied.

⊙ If current flows in the substrate, Vsub will rise, turning ON the npn-transistor.



- This pulls down the Vwell & turning ON the pnp-xsistors.
- The pnp-xsistor in turn raises Vsub, initializing the +ve f/B loop with large current flowing b/w VDD & GND that persist until power supply is turned OFF or the power wire melts.

- * Latchup can be prevented by reducing Rsub & Rwell.
- ** A conservative approach is to place 'tap' adjacent to every source connected to VDD & GND.

The following guidelines :-

(i) Every well should have atleast one tap.
— because the n-well sheet resistance can be several KΩ per square, it is necessary to ground the well thoroughly by providing a sufficient no of well taps.
— This will prevent excessive voltage drops due to well currents.

(ii) All substrate & well taps should connect directly to the appropriate supply in metal.

(iii) A tap should be placed for every 5-10 transistors, or more often in 'space' areas.

(iv) nmos xistor should be clustered together near GND & pmos xistor should be clustered together near V_{DD}.

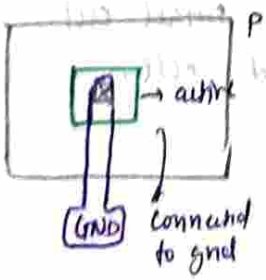
Q. What is latch up & how do we avoid them? (Conclusion)
PTAP & NTAP are used for?

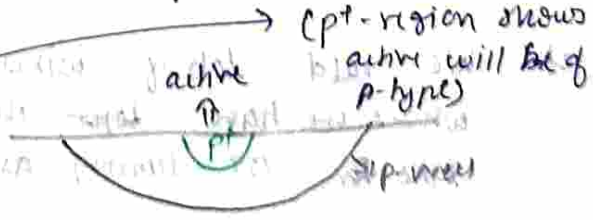
→ • +ve FIB loop due to npn & pnp xistor.
• It can be avoided by reducing the resistances
• multiple PTAP & NTAPs (substrate & well taps) are used to reduce R_{sub} & R_{well}.

* After LVS we do substrate coupling extraction in cadence.
- LVS will show error if you don't put PTAP & N1-well in layout
tap → substrate connection, → contacts are sometimes else

→ 'Vias' in layout are different from 'taps'.

Q. How do you ~~connect~~ make a 'tap', if you want to make the substrate connection for p-well. (for p-MOS).



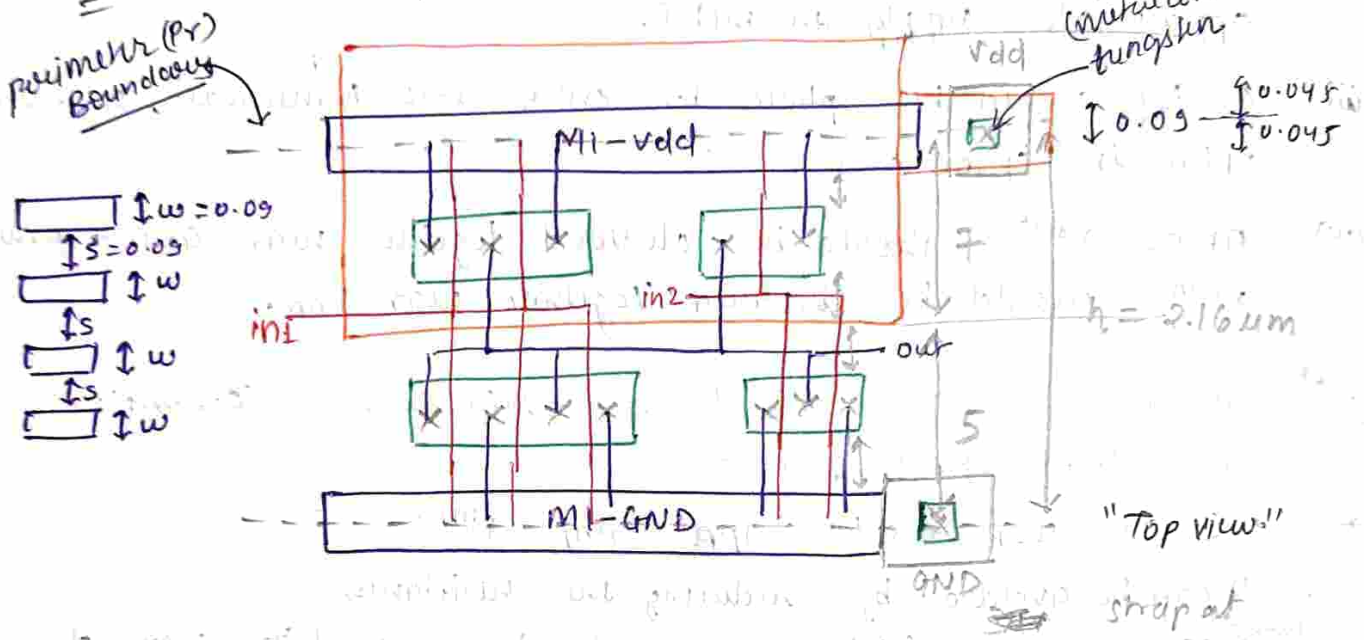


* There is no depletion region b/w them

** n+ & p+ regions shows us what kind of doping goes there they are not actually the layers, they are the fabricated. (why?)

Part-4

Q. How the templates of the standard cells are made?



Q. How do we decide the height of 'h'.

→ The minimum width of M1-metal = 0.09 µm
The min. spacing b/w M1 & M1 = 0.09m
DRC rule

Hence if we can use using 12T (12 track) library where T = 0.18 µm, then we can place only 12 metal layer (M2, M3, ... & i.e., other metals) above our cell with the min. spacing b/w them

h ⇒ 12T = 12 × 0.18 = 2.16 µm

It We need lots of wires to run across the whole chip where we have millions of small cell hence we take 12T library as standard cell.

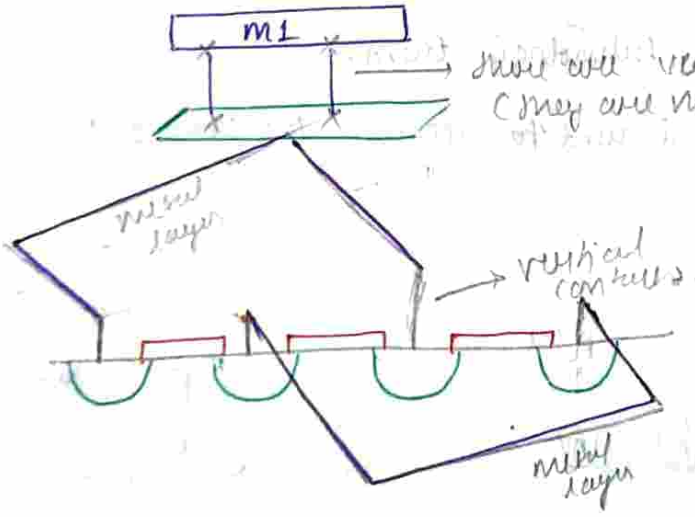
- We only take half DRC of the metal layers above & below the cell.
  - As metals layers (VDD, GND) are shared with other cell above it & below it.
  - cell above & below will be rotated 180° for connection of VDD & GND.

- DRC is only b/w M1-M1, PO-PO.
  - we can pass M1 above the PO or vice-versa.

- Metal contacts are tungstens,
  - via is copper from dual damascene process

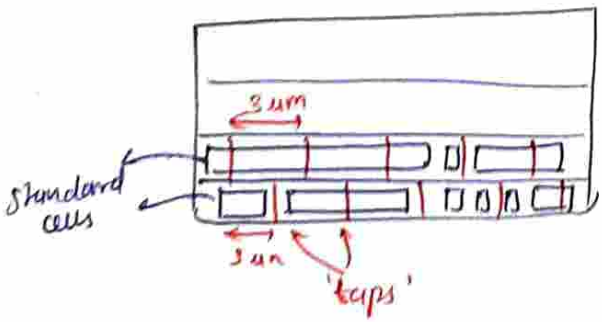
Q. When we make any vertical connection b/w metal M1 & active region, as M1 is running above active region. Is that by via?

— No it is not a via.



* minimum spacing b/w the taps should be ~~0.8~~ 3µm.

Q. How do we place the 'taps' in our chip?



* During placement of the standard cell in our chip we also put the 'taps' with minimum spacing.
  - So no need to make tap for every cell.

* For no-active region the tap should be farmed from 3µm (min. DRC) distance away.

Q. If we have many no. of cells b/w, so do we have to share the n-well for all the stand. cells as taps are ~~used~~ used at 3µm distance only.

→ Yes, the n-well will also be from right to left.

- Therefore it is advised to make np & pp vertically parallel to each other.

* 'taps' are also called as 'straps'.

Q Why we have DRCs?

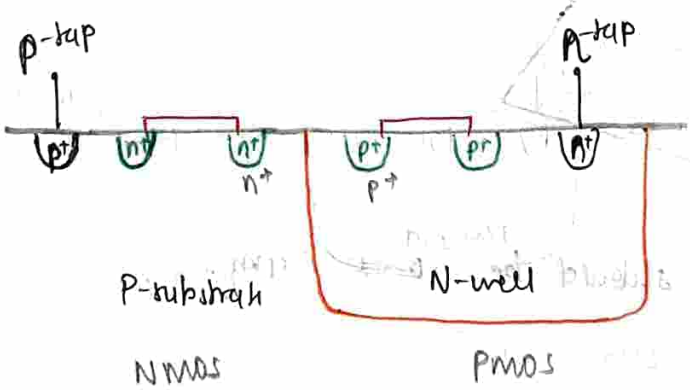
→ Agar hum min DRC se kam width like hai to technology wale logo ko koi dikkat nhi hai, dikkat is liye hogi kuki functionality of the device may change.

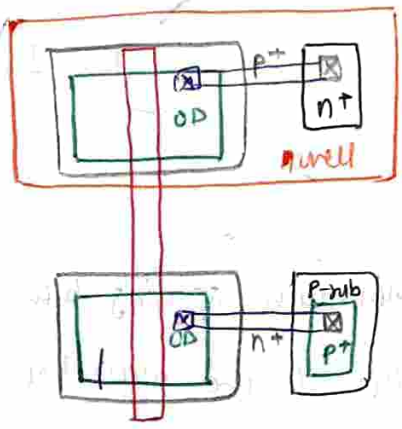
- Agar min ~~width~~ spacing se kam space denge to do wires or two things may get short circuited.

"Sky-water"

- DRC is defined by technology team.

* V_T implant layer is used to decrease the value of V_T.





Part-2

4. Challanges in Cu-wire:

- Diffusion barrier in Cu-wire has high resistance.
- Cu is also affected by dishing during polishing.

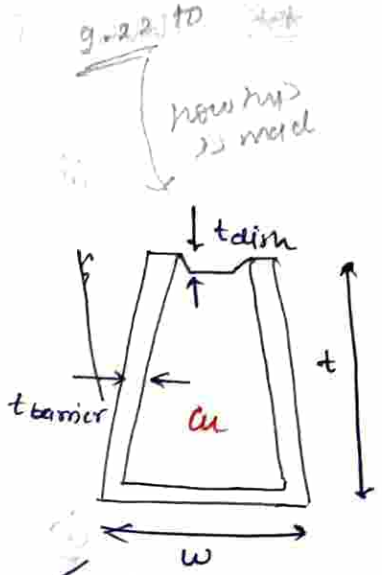
VDF

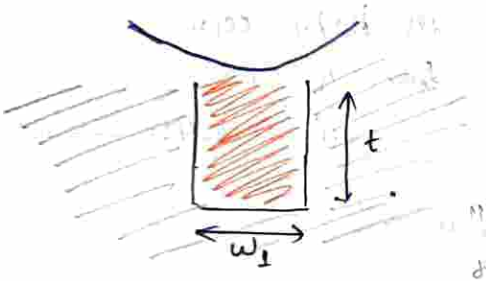
Q. What is dishing in Cu-wire?

→ polishing → flattening out the surface

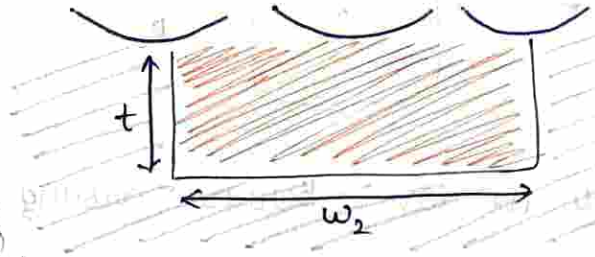
- Due to this polishing a mechanical force is also applied which presses the Cu-wire & don't affect the Barrier.

→ this changing the thickness of Cu is called dishing.





"cross section view"



for less width 'w1' less dishing is observed

for high width 'w2' more dishing is observed.

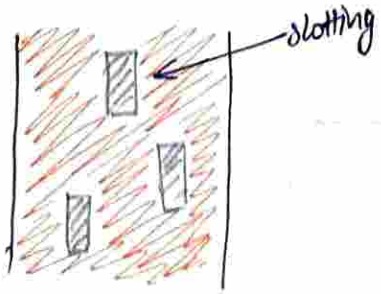
but becomes this. due to dishing

Q. What is the problem due to dishing?

→ We wanted less resistance hence we increase the width 'w' but due to dishing the effective area is reduced hence increasing the Resistance

Q. How to decreasing this dishing?

→



"Top view"

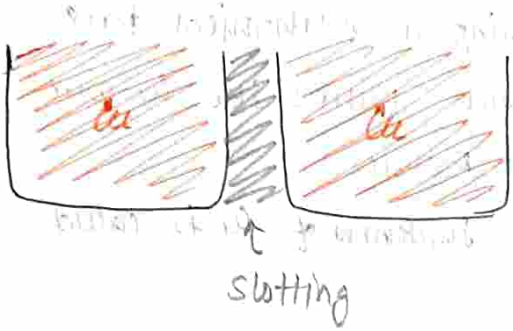
- if we can somehow increase the mechanical strength of Cu, then dishing can be reduced.
- — Therefore we use slot in b/w the Cu-wire, this slots are of SiO_2.
- — * This also reduces the Cu-area, ∴ R↑ but it is better than reducing the width of Cu.

(i) Reduce dishing by decreasing width but it will increase resistance more.

(ii) Reduce dishing by slotting. It will also increase the resistance but not that much.

*** There is DRC to all Cu-wire, slot, barrier etc.

(ii)



(i)

Less dishing occurs in both cases, but $R_1 < R_2$

∴ (ii) is better than (i)

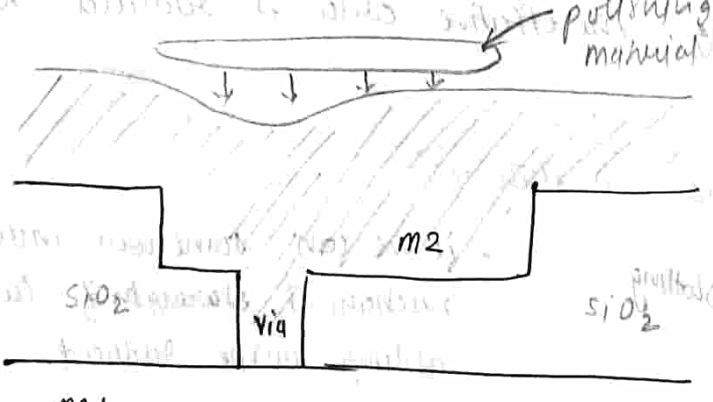
Q. Why do we have barrier around Cu-wire?

~~RC Delay~~

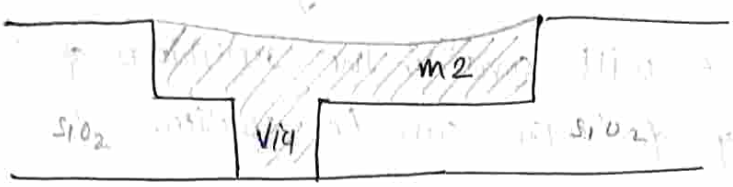
Wire Pg 220 (6.3.1) (6.3.5) (6.4.2) 435

Wells Crosstalk (6.3.3) (6.4.3)

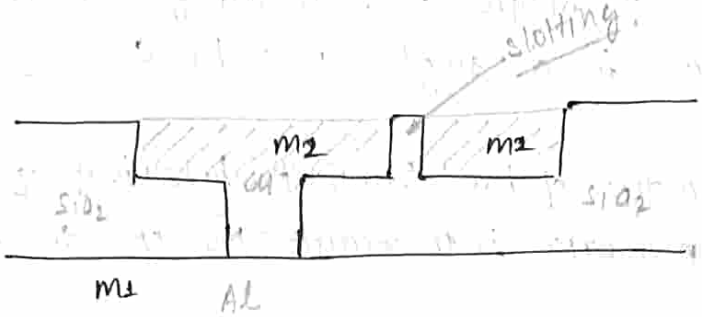
Robust Pg. 149 (4.4.2) (4.4.3) (4.4.4)



dual damascene



⇒ Resistance ↑. (bad)



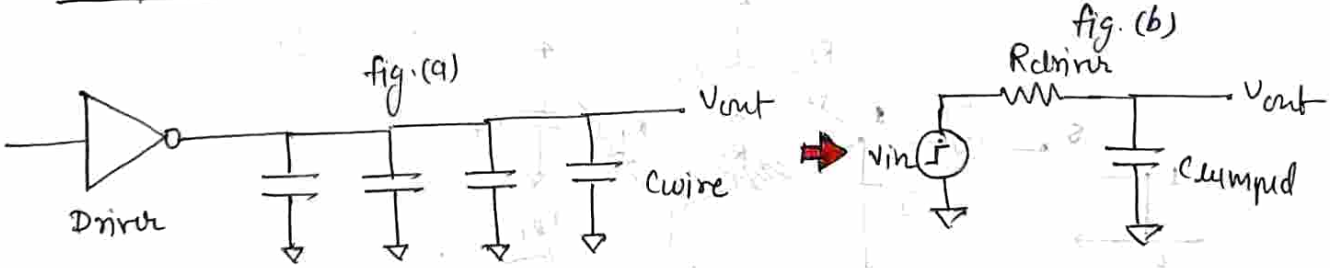
⇒ Resistance ↓

- We use low-K dielectric b/w metal (for low capacitance)
- We use high-K dielectric in gate insulator region.

* miller coupling.

5. RC Delay

(i) Lumped Model "Lumped capacitor Model"



- In this model we only consider the capacitive component of wire, and to lump the distributed capacitance into a single capacitor.
  - In this model the wire still represents an equipotential region, and the wire itself doesnot introduce any delay.
- Distributed v/s lumped cap. model of wire [fig.(a) & (b)]

  $C_{lumped} = L \times C_{wire}$

  $L \rightarrow$ length of wire

  $C_{wire} \rightarrow$ cap. per unit length

  * The driver is modeled as a voltage source & a source resistance = R_{driver}.

(ii) Lumped RC model [Elmore Delay] Pubaru 4.4.23

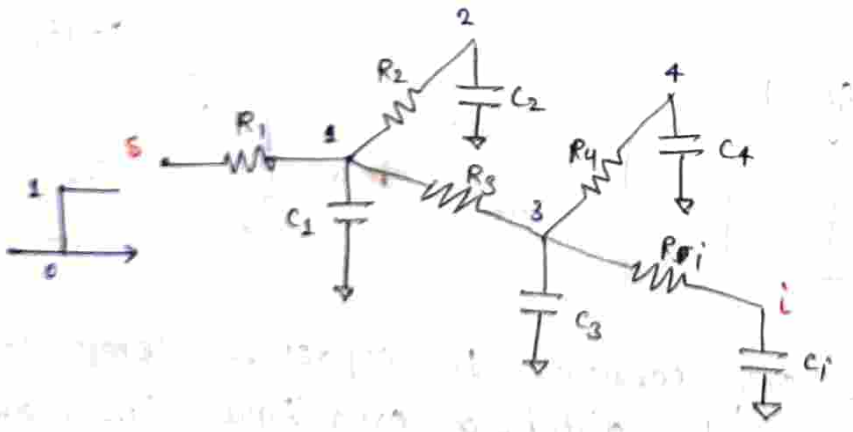
- In this model, we will consider the resistance of wire also.
  - Hence a delay of wire can be studied.

- In this model, we lump the total wire resistance of each wire segment into a single R & similarly combines the global capacitance in a single capacitor C.
  - A common study of the transient behaviour of complex system-wire networks is to reduce the ckt to an RC n/w.

- A large complex RC n/w can be solved by two methods:
  (i) Elmore Delay method
  (ii) Distributed RC delay method

Elmore Delay : [only good for short length wire]

Consider a less complex RC n/w for understanding Elmore delay,



Properties of above RC-tree:

(i) The n/w has single i/p node.
(ii) All the cap are b/w a node & the ground
(iii) The n/w doesnot contain any resistive loop (which makes it a tree).

- The total Resistance along path 's to 'i' is, path Resistance (R_{ii})
  - The path resistance b/w the source node 's' & node 4' is,

$$R_{44} = R_1 + R_3 + R_4$$

- the resistance shared among the path from the root node 's' to node 'k' & 'i', is shared Path Resistance (R_{ik})

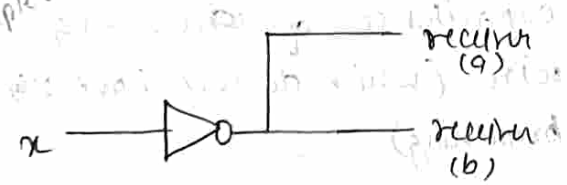
* Assume 'N' nodes of the N/w is initially discharged to GND, and that a step i/p is applied at node 's' at t=0.
  - The Elmore delay at node 'i' is,

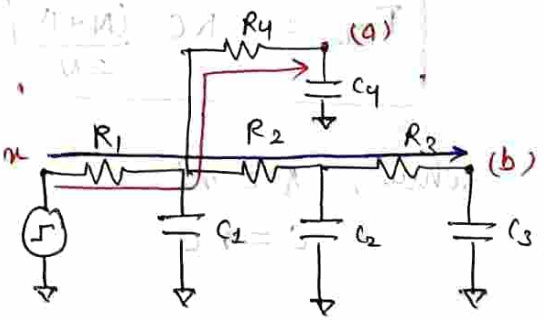
$$\tau_{Di} = \sum_{K=i}^{N} C_K R_{iK}$$

shared path R_{iK}

$$\therefore \tau_{Di} = R_1 C_1 + R_1 C_2 + (R_1 + R_3) C_3 + (R_1 + R_3) C_4 + \cdots + (R_1 + R_3 + R_i) C_i$$

Eg: Find Elmore Delay from i/p x to each receiver (a) & (b)





→

$$\tau_{D(a)} = R_1 C_1 + R_1 C_2 + R_1 C_3 + (R_1 + R_4) C_4$$

$$\tau_{D(b)} = R_1 C_1 + (R_1 + R_2) C_2 + (R_1 + R_2 + R_3) C_3 + (R_1 \cdots) C_4$$

Example : Time constant of Resistive-capacitive wire

- Time constant → RC delay.

A wire of total length 'L' is partitioned into N-identical segments, each with a length of L/N. The resistance & capacitance of each segment are hence given by $\frac{rL}{N}$ & $\frac{cL}{N}$ resp.

Compute the dominant time constant of the wire.

→

$$\tau_{DN} = \frac{rL}{N}\cdot\frac{cL}{N} + \frac{2rL}{N}\cdot\frac{cL}{N} + \frac{3rL}{N}\cdot\frac{cL}{N} + \cdots + \frac{Nrl}{N}\cdot\frac{cL}{N}$$

$$= \left(\frac{L}{N}\right)^2 [rc + 2rc + 3rc + \cdots + Nrc]$$

$$= (rc)\left(\frac{L}{N}\right)^2 [1 + 2 + 3 + \cdots + N]$$

$$= rc\,\frac{L^2}{N^2}\cdot\frac{N(N+1)}{2}$$

$$= rc\,L^2\,\frac{(N+1)}{2N} = (rL)(cL)\,\frac{(N+1)}{2N}$$

$$\boxed{\tau_{DN} = RC\,\frac{(N+1)}{2N}}$$

→ Total lumped resistance & capacitance of a long wire (which doesn't have any branching)

where, $R = rL$

$C = rC$

If for very large value of N, this model asymptotically approaches the distributed rc line, which gets simplified to

$$\tau_{DN} = \frac{RC}{2} = \frac{rcL^2}{2}$$ → "delay for distributed rc"

$\tau_{DN} = RC$ → "delay for lumped rc"

Conclusion :

1. The delay of the wire is a quadratic func^n of its length.

$$\tau_{DN} \propto L^2$$

2. The delay of the distributed rc-line is half of the delay that would have been predicted by RC model.

$$\tau_{distribution} = \frac{1}{2} \tau_{lumped}$$

* In a switch model, the transistors are replaced by their equivalent, linearized on-resistance. Then delay is calculated by RC model.

(iii) Distributed rc-line

- We will not use this method for our delay calculation.
- The step-response of lumped & distributed RC N/w

Voltage range	Lumped RC N/w	Distributed RC N/w
0-50% (t_p)	0.69RC	0.38RC
0-63% (τ)	RC	0.5RC
10% - 90% (t_r)	2.2RC	0.9RC
0-90%	2.3RC	1.0RC

Que: When should be use lumped RC model & lumped capacitive model.

→ (i) RC delay should be considered when

$$t_{PRC} >> t_{pgate}$$

t_{PRC} ↓ propagation delay for RC model

t_{pgate} ↓ driving gate delay

$L_{crit} \Rightarrow \sqrt{\frac{t_{pgate}}{0.38RC}}$

L_{crit} depends on
- sizing of driving gate
- Choosing interconnect material.

(ii) RC delay should be considered when rise (fall) time at the i/p is smaller than RC, the rise (fall) time of the line.

$$t_{rise} < RC$$

- with R & C the total resistance & cap. of wire

* when this condⁿ is not meet, the change in signal is slower than the propagation delay of the wire, and a lumped capacitive model suffices.

Note: (i) 3-segment π-model is more accurate (is upto 30%)

(ii) But we commonly use single-segment π-model

* This π-model is nothing but the lumped RC model. & lumped RC model is intermediate b/w lumped cap. & distributed RC.

Q. How does simulator works?

→ Simulator works on matrix.
Matrix size increases with more no. of elements or nodes.

∴ • Simulation time is less for π-model than distributed RC. [lumped RC]
• Simulation time is more for π-3 model than π-1 model.

Q. * "Contacts" & "vias" have completely different resistance profile than the "wires" (why?)

— If you want to decrease the resistance of wire you increase its area.

— If you want to decrease the resistance of via you don't have any control. Therefore you make the multiple resistances in parallel.

(*) Cost of fabricating via is more than metals. (why?)

DFM → Design for manufacturability
⇒ which checks the no. of contacts in our design.

∴ if increasing the no. of vias not affecting the area of the chip then we can do that.

Q. * How many contacts we can put in our design or should?

→ If we instantiate a pMOS (pcell) from the cadence of width $w_p = 10\mu m$ then it would have a numbers of contacts then it is good or bad?

✓ Increasing the no. of contact ↓ resistance.

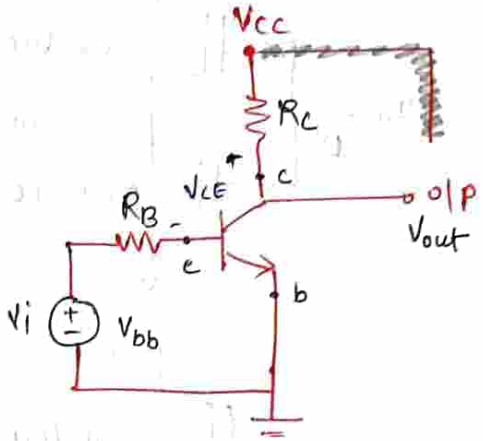
✓ Increasing the no. of contacts ↑ gate-contact capacitance

⇒ So we keep the no. of contacts such that our requirement is filled & both g. cap. & Resistance is at good value.

6. Crosstalk effect : Weste Pg. 222 (6.3.3)

3. Design of an Inverter

BJT Inverter



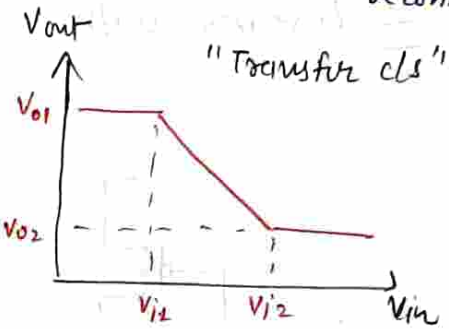
- When the i/p voltage is low, the xsistor doesnot conduct. there is no current through R_C, and the collector voltage (o/p) is pulled upto V_{cc}.
- When the i/p voltage is increased, the xsistor begin to conduct, the voltage drop across R_C starts increasing, and the o/p voltage falls.

finally,

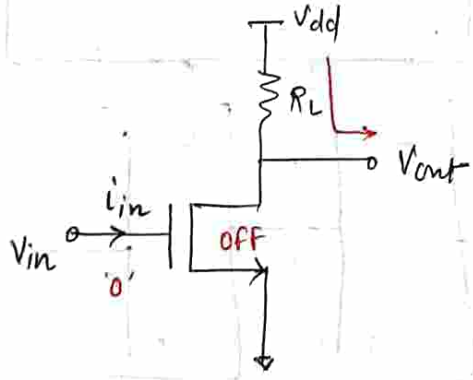
- When the i/p voltage is high enough to drive the xsistor into saturation, there is small fixed drop (V_{CE} = 0.1 to 0.2V) across the xsistor, & o/p voltage saturates to its low value.

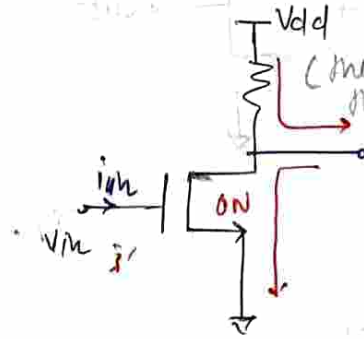
* when V_{bb} is less
∴ $V_{eb} < 0.7V$
⇒ o/p = high

* when V_{bb} is high
∴ I_c is high &
o/p = low (GND)
is connected



MOS Inverter

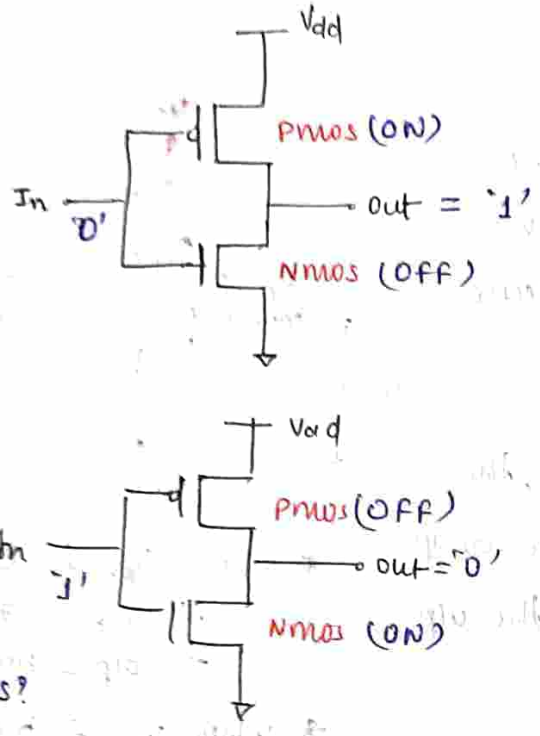




"short ckt power"

- if V_{in} = 0, MOSFET is off
∴ V_{out} = Vdd '1'
- if V_{in} = 1, MOSFET is ON
∴ V_{out} = GND '0'

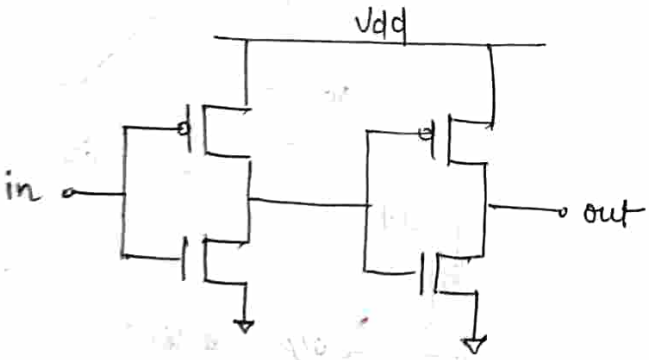
Cmos Inverter



why cmos?

- At steady state, no power consumption.
- Better noise margin

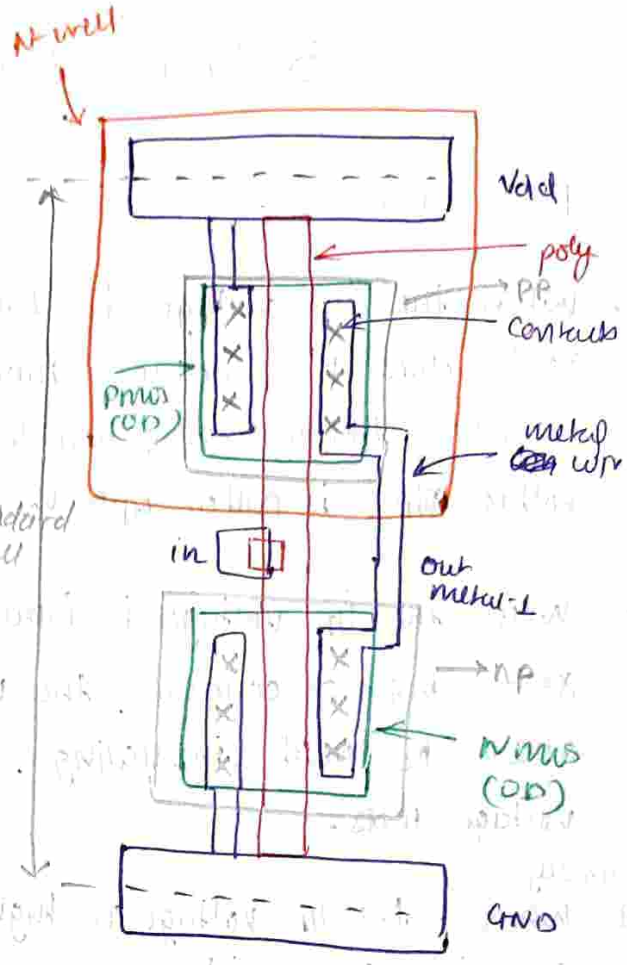
2 Inverter with 'straps' (taps)



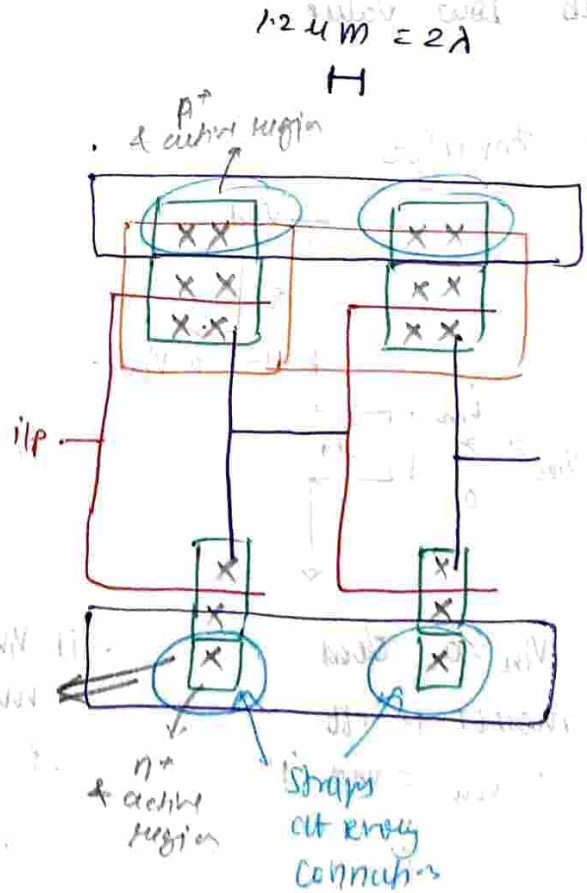
"Buffer"

- We can also use this straps later at the required place also.

* Straps are required only at 32um distance.



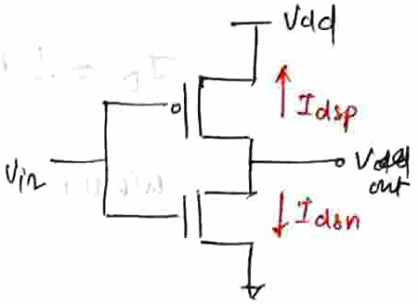
"without taps" (straps)



DC Response

- DC Response → If my i/p is held constant at a particular voltage level at what voltage my O/P be held at.

  for V_{out} v/s V_{in} (V_{in} for gate)



- for CMOS: when $V_{in} = 0$, $V_{out} = V_{dd}$
  when $V_{in} = V_{dd}$, $V_{out} = 0$
- In b/w, the V_{out} depends on transistor size & current.
- By KCL, $|I_{dsn}| = |I_{dsp}|$
- We could solve equations but graphical solution give more insight.

Noise → DC response
Delay → transient response

MOSFET: Region of operation

NMOS

Cutoff

$V_{gsn} < V_{tn}$

$V_{in} < V_{tn}$

Linear

$V_{gsn} > V_{tn}$

$V_{in} > V_{tn}$

$V_{dsn} < V_{gsn} - V_{tn}$

$V_{out} < V_{in} - V_{tn}$

$$\begin{bmatrix} V_{gsn} = V_{in} \\ V_{dsn} = V_{out} \end{bmatrix}$$

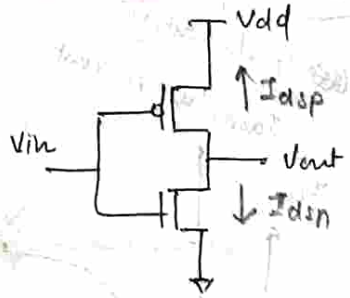
Saturation

$V_{gsn} > V_{tn}$

$V_{in} > V_{tn}$

$V_{dsn} > V_{gsn} - V_{tn}$

$V_{out} > V_{in} - V_{tn}$



PMOS

Cutoff

$V_{gsp} > V_{tp}$

$V_{in} > V_{DD} + V_{tp}$

Linear

$V_{gsp} < V_{tp}$

$V_{in} < V_{DD} + V_{tp}$

$V_{dsp} > V_{gsp} - V_{tp}$

$V_{out} > V_{in} - V_{tp}$

Saturation

$V_{gsp} < V_{tp}$

$V_{in} < V_{DD} + V_{tp}$

$V_{dsp} < V_{gsp} - V_{tp}$

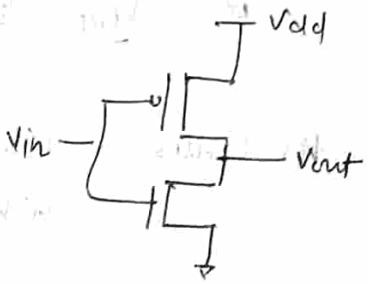
$V_{out} < V_{in} - V_{tp}$

$$\begin{bmatrix} V_{gsp} = V_{in} - V_{DD} \\ V_{dsp} = V_{out} - V_{DD} \end{bmatrix}$$

NMOS - PMOS : I-V c/s

- We are assuming that $\beta_n = \beta_p$ by making the pmos wider than the nmos.

$$I_D = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{tn})^2$$ in saturation



where, $\beta_n = \mu_n C_{ox} \frac{W_n}{L_n}$

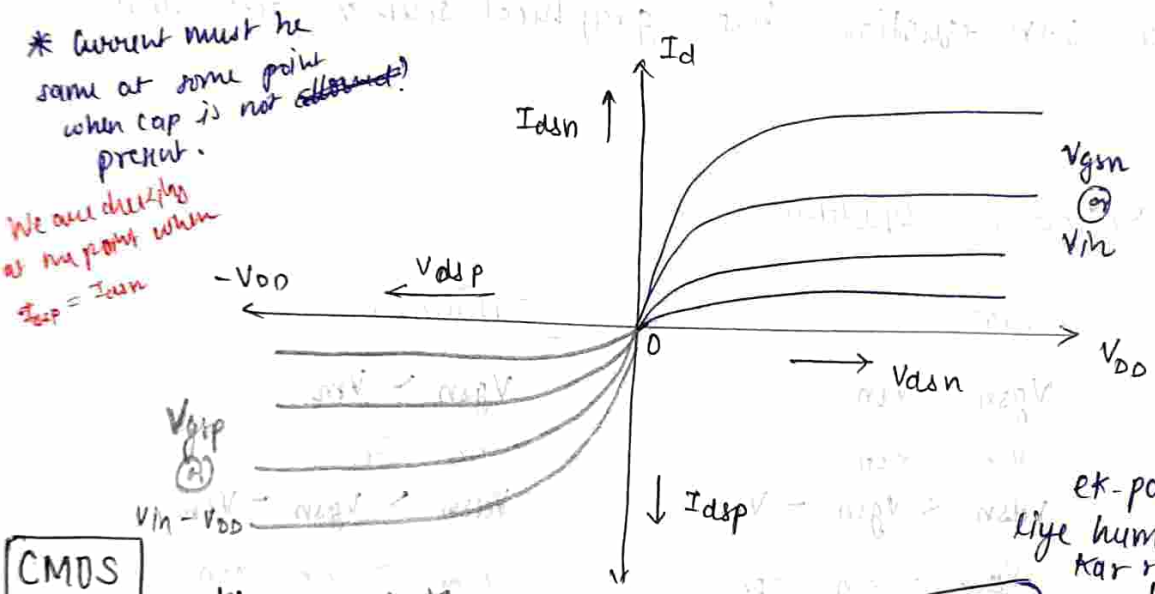
$\beta_p = \mu_p C_{ox} \frac{W_p}{L_p}$

since, $\mu_n \neq \mu_p$ or $\mu_n > \mu_p$

∴ To make $\beta_n = \beta_p$ we make wider pmos ∴ $W_p > W_n$.

* Current must be same at some point when cap is not present.

We are checking at the point when $I_{dsp} = I_{dsn}$



CMOS

ek particular i/p ke liye hum o/p ko check kar rhe hai.

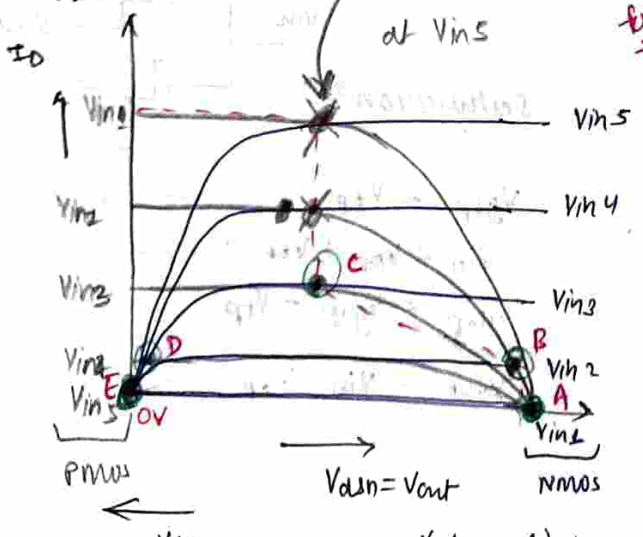
⇓

1. Vout vs Vin

I_D vs Vout, Vm

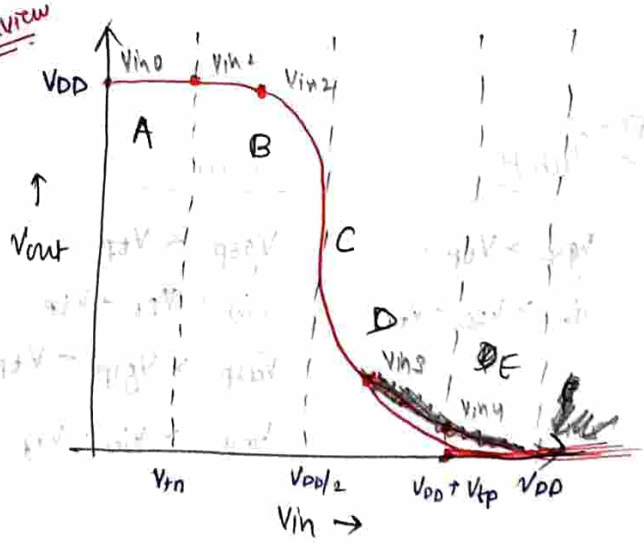
I_{dsn}, I_{dsp} vs Vout

Vout at this point $I_{dsn} = |I_{dsp}|$ at Vin 5



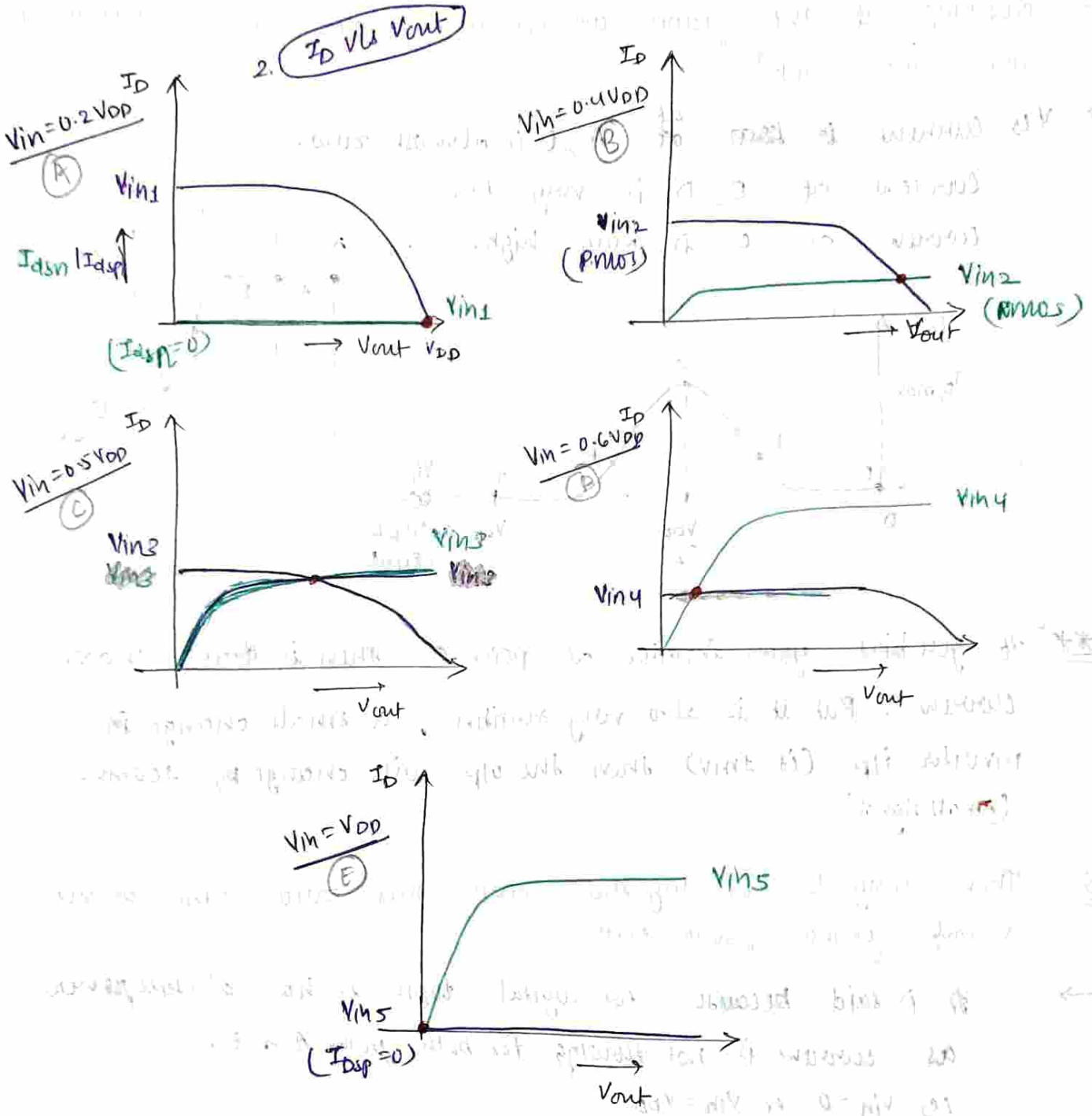
"load line Analysis"

Imp for interview



"DC transfer curve of CMOS"

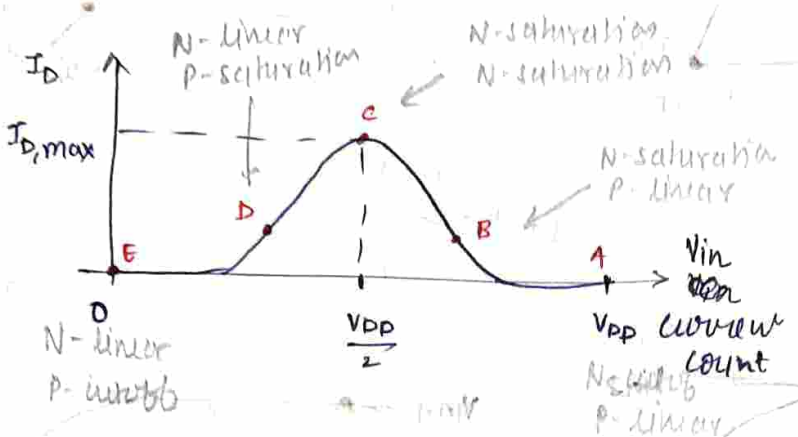
2. I_D Vs V_{out}

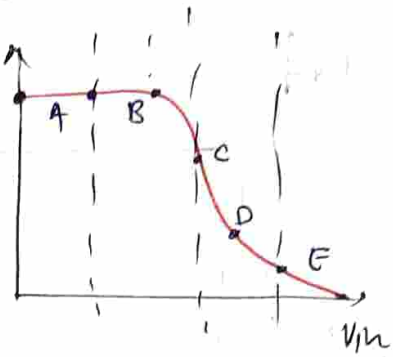


Region	nMOS	pMOS
$V_{in}=0.2V_{DD}$ (A)	Cutoff	Linear
$V_{in}=0.4V_{DD}$ (B)	Saturation	Linear
$V_{in}=0.5V_{DD}$ (C)	Saturation	saturation
$V_{in}=0.6V_{DD}$ (D)	Linear	saturation
$V_{in}=V_{DD}$ (E)	Linear	cutoff

Q. Looking at the graph can we also predict the current (I_D) from the graph?

→ Yes, current at A, E is almost zero.
current at B, D is very low
current at C is very high.





** If you bias your inverter at point C, then it draws max. current. But it is also very sensitive, a small change in inverter i/p (is 1mV) then the o/p will change by 100mV. (small signal)

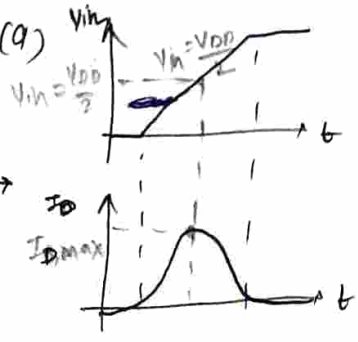
Q. Then why do we say that cmos has zero static power & only dynamic power exists.

→ It is said because for digital logic it has '0' static power as current is not flowing for both point A & E.
ie, $V_{in} = 0$ or $V_{in} = V_{DD}$.

Also for analaog it has some static power when small signal voltage is (small change in i/p voltage) is applied when operating in region C (sat-nmos & sat-pmos)

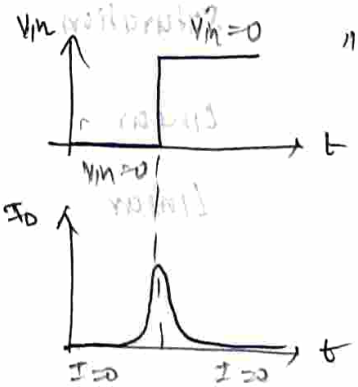
Q. What would be the curve of I_D vrt time if V_{in} vrs time is given.

(a)



small signal "Analog"

(b)



* So, no current is drawn for digital signal hence, static power = 0

And, for small signal (V_{in}) $I_D \neq 0$, ∴ it has some static power.

* For very less rise and fall time of V_{in} (b) we don't pay so much of attention in region C, we directly move from E to 1.

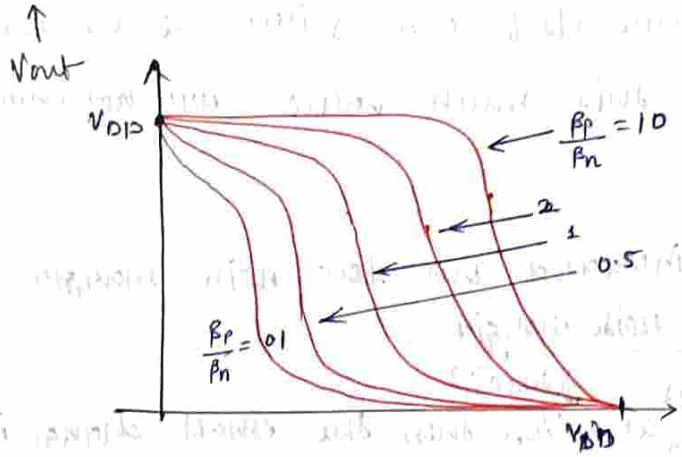
Part-2 **β - Ratio** [Weste pg 90 (2.5.2)]

Q What would happen if β_n is greater than β_p?

→ the transfer cls of V_{out} & V_{in} (ie, cmos) will move towards left.

for $\beta_n > \beta_p$ → curve shift left

$\beta_n < \beta_p$ → curve shift right



β ⇒ 2.5 → 2] opt
rise time ↓
fall time ↓

- If $\beta_p / \beta_n \neq 1$, switching point (ie, region 'c') will move from $V_{DD}/2$ → called skewed gate
- Other gates will also have similar kind of V_{out} vs V_{in} cls not same because their β-ratios can also be different.

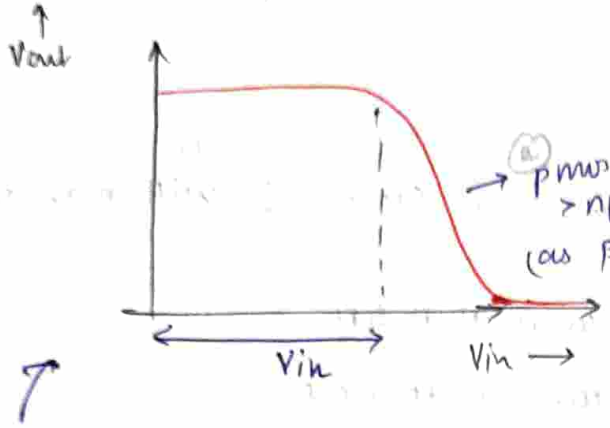
⇒ The method that we have used to draw the transfer cls of cmos inverter, we can also use this method to draw cls of other gate also. [for inverting gates only]

– for non-inverting gates the transfer cls may change.

Q. As a designer, what is the relevance of the change in the β-value?

→ By changing the β-value you change the noise margin of your inverter.

- At times I don't want my inverter to switch.



If this is the transfer c/s of our xsistor.

(2) means saturation current of nmos with the earlier current of pmos.

(1) If this is the transfer c/s of our xsistor & let the Vin be change by this much value will not change the o/p value

(2)

- means we have increased our low-noise margin & reduced the high noise margin.

- means if my i/p is at V_{DD} then the small change in Vin my o/p will go do it. (how?)

Q. 8:42 Now $\beta_p > \beta_n$ then why is it shifting towards right?

→ If $\beta_p > \beta_n$ then the saturation current of nmos is much less than the linear current of pmos.

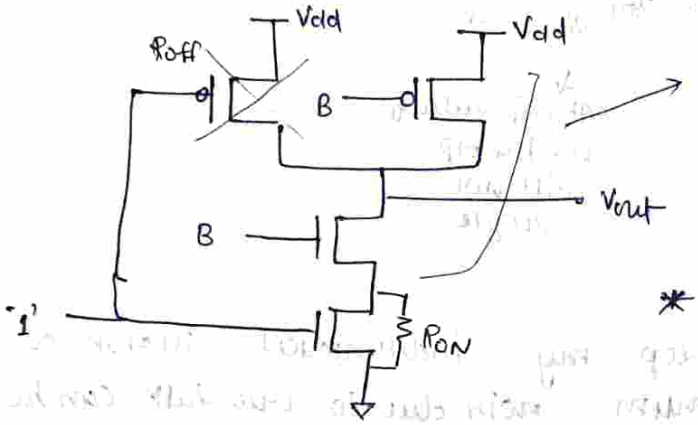
$$I_{dsn(sat)} < I_{dsp(linear)}$$

So even the nmos is trying to sink as much current as it can, the linear current of pmos is so high that it kind of doesn't allow the o/p cap. to discharge.

Q. What is pull up & pull down N/W?

Q. How the other gates will collapse into the inverter c/s?

→ Let us assume a NAND gate



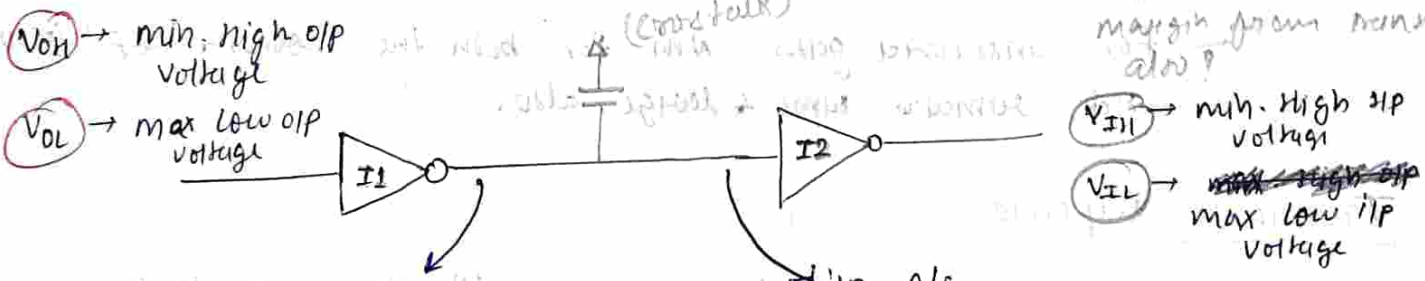
Therefore only this ckt remains and this now acts as like NAND gate is collapsed into an inverter.

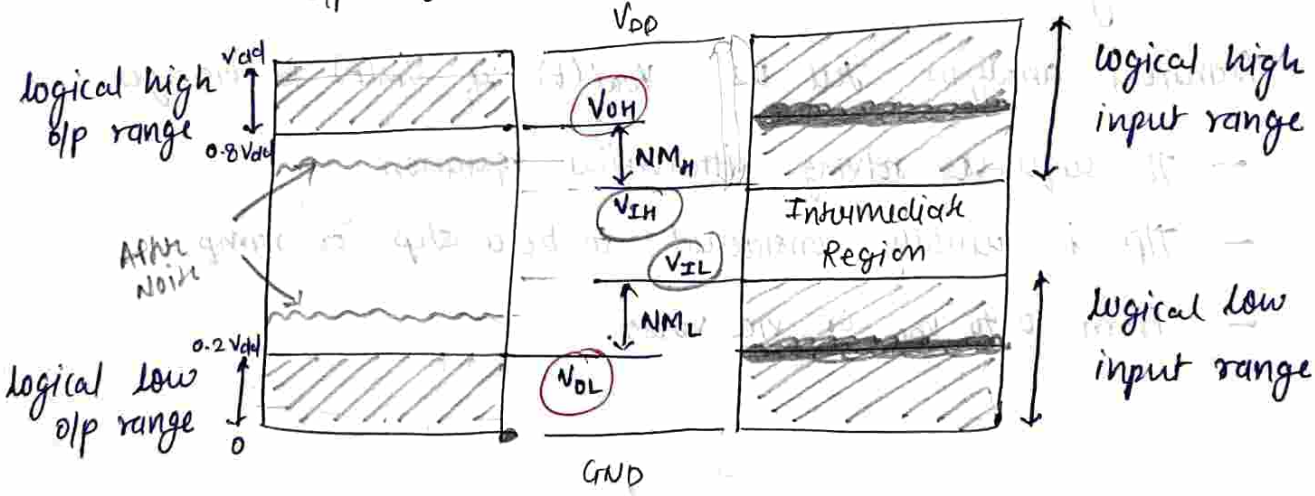
* So our main job is to select those i/p in other gates to get the c/s of inverter.

Part-3

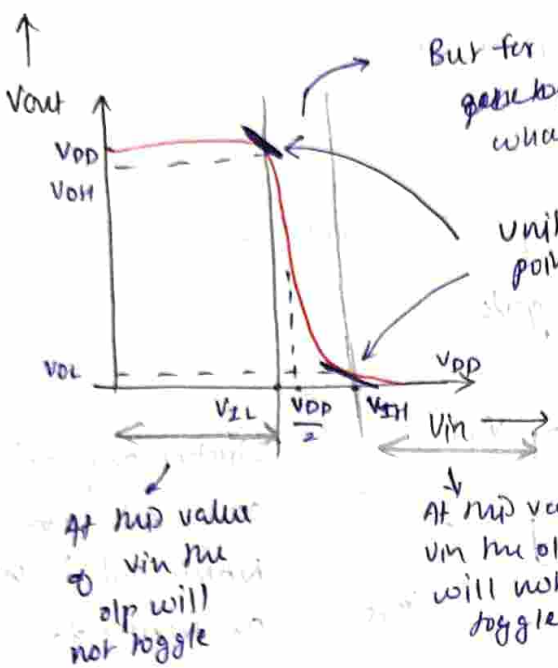
Noise Margin (Week-Pg. 94)

Why DC analysis, if we get noise margin from transient also?





- As the o/p from I1 goes out due to some crosstalk noise is added to the o/p voltage.
- So the noise margin (NM_H & NM_L) should be such that the voltage coming from I1 should should be easily read by the i/p of the I2 inverter.



- So we will try to keep my intermidiate region as small as possible so maximum noise due to crosstalk can be easily handled.
  - for skewed gate if NM of one side increases then NM of other side decreases
  - for unskewed gates NM for both the lower & higher side remains same & large also.

Transient Response

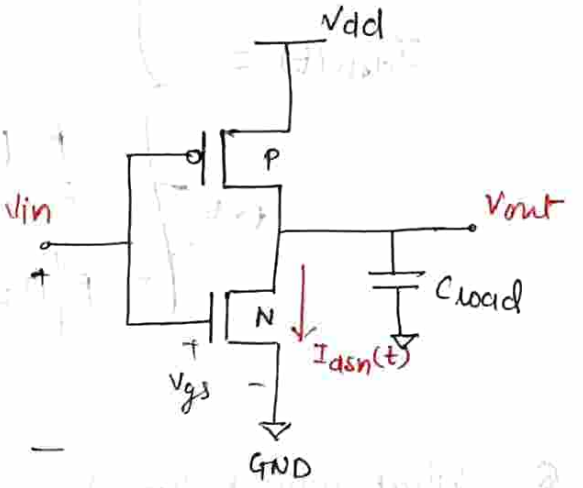
Week Pg. No. 3 (4-2)

(i) DC analysis tells us about Vout if Vin → constant

(ii) Transient analysis tell us $V_{out}(t)$ if $V_{in}(t)$ changes
  - It requires solving differential equation
  - I/p is usually considered to be a step or ramp
  - From 0 to V_{DD} or vice-versa.

1. Delay Models :

Inverter step response :



- $V_{in}(t) = \begin{cases} 0 & , \ t < t_0 \\ V_{DD} & , \ t \geq t_0 \end{cases}$

$= u(t - t_0) V_{DD}$

~~$V_{out}(t < t_0)$~~

- $V_{out}(t) = \begin{cases} V_{DD} & ; \ t < t_0 \\ ? & ; \ t \geq t_0 \end{cases}$

$V_{gs} = V_{in}$

$\therefore \quad \frac{dV_{out}(t)}{dt} = -\frac{I_{dsn}(t)}{C_{load}}$

$t < t_0$: charging of cap.

[NMOS - ~~OFF~~ OFF , PMOS - ~~OFF~~ ON]

$$I_{dsn} = \begin{cases} 0 & , \ V_{gs} < V_{tn} \\ \frac{\beta}{2}(V_{gs} - V_{tn})^2 & , \ V_{DS} > V_{GS} - V_{tn} \\ \beta\left[(V_{gs} - V_{th})V_{DS} - \frac{V_{DS}^2}{2}\right] & , \ V_{DS} < V_{GS} - V_{tn} \end{cases} \quad \text{---- (1)}$$

$V_{gs} = 0V$ ✓

$V_{out} = V_{DD}$

$V_{DS} = V_{DD}$ ✓

$t \geq t_0$ discharging of cap.

[NMOS - ON , PMOS - OFF] → when $V_{GS} = V_{DD}$ & $V_{DS} = V_{out}$

initially at $t < t_0$, $V_{DS} = V_{DD} > V_{GS} - V_{th}$

$\therefore$ NMOS is in saturation region.

when $t \geq t_0$, $V_{DS} = V_{out}$ falls below $V_{DD} - V_{th}$

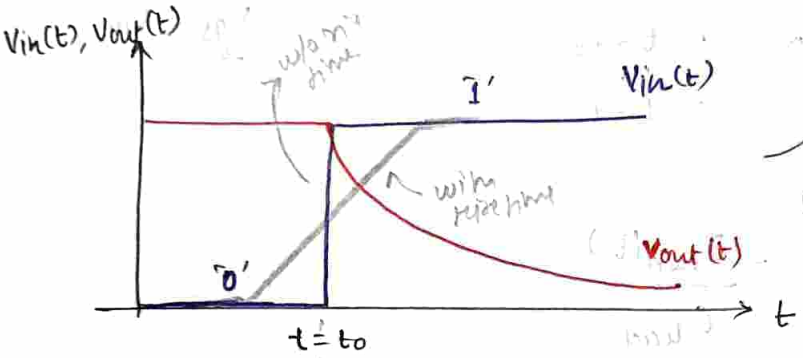
$\therefore$ NMOS comes in linear region.

** $t \geq t_0$ in eqn. (2).

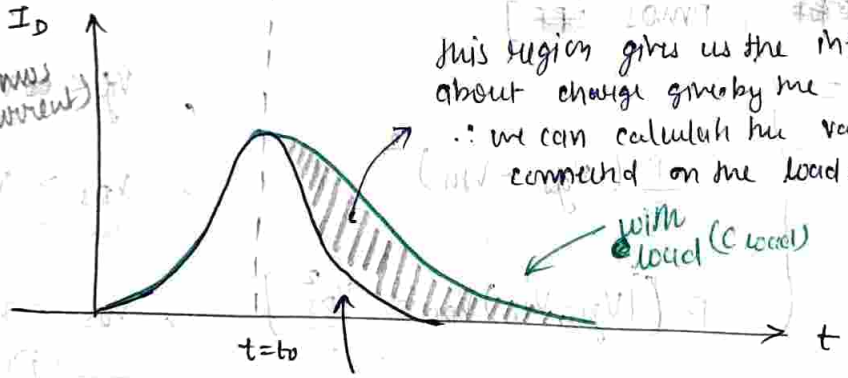
$$I_{dsn}(t) = \begin{cases} 0 & , \quad t < t_0 \\ \dfrac{\beta}{2}(V_{DD} - V_{tn})^2 & , \quad V_{out} > V_{DD} - V_{tn} \\ \beta\left[(V_{DD} - V_{tn})V_{out} - \dfrac{V_{out}^2}{2}\right] & , \quad V_{out} < V_{DD} - V_{tn} \end{cases} \qquad t \geq t_0$$

Q: What will be the impact of C_{load} on current waveform?

→

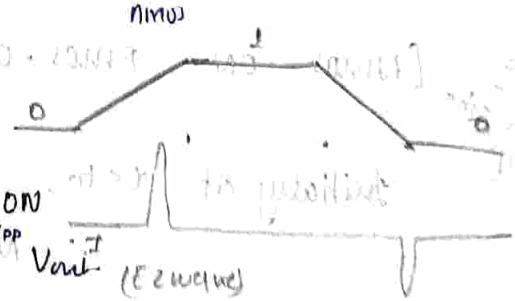


Mid-sem



without C_{load} : $I_{dsn} = I_{dsp}$

with C_{load} : $I_{dsn} \neq I_{dsp}$ (i ≠ 0)



w/o load

* when $V_{in} = 0V$ '0' NMOS - OFF PMOS - ON

∴ I_{dsp} flows from V_{DD} to V_{out}

$V_{in} = 1V$ '1' NMOS - ON , PMOS - OFF

∴ I_{dsn} flows from V_{out} to Gnd

with load

* $V_{in} = 0' \Rightarrow$ NMOS - OFF, P-MOS - ON :

$\therefore$ I_{dsp} flows from V_{DD} to C_{load} & charge the cap.

$V_{in} = 1'$, NMOS - ON, PMOS - Off :

$\therefore$ I_{dsn} flows from C_{load} to GND

$\because$ due to RC delay the current I_{dsn} takes more time discharge the cap.

Delay Definitions

(i) $t_{pdr} = t_{PLH}$ → Rise propagation delay

- When o/p switches from low to high, the time difference b/w the instant when o/p is at 50% of its max value & when the i/p is at 50% of its max value.

(ii) $t_{pdf} = t_{PHL}$ → fall propagation delay

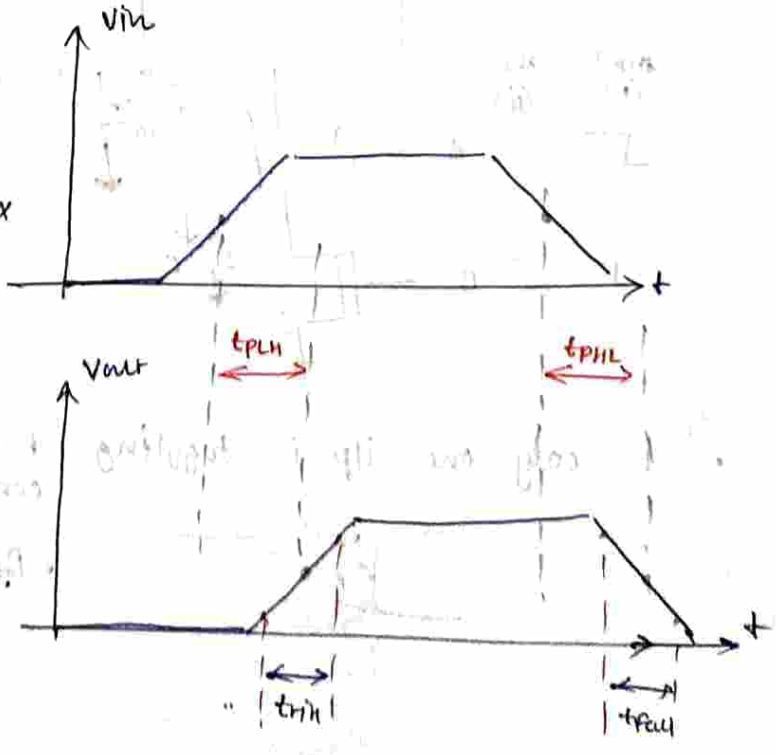
- When o/p switches from high to low, the time difference b/w the instant when o/p is ....

(iii) t_{pd} → average propagation delay

$$t_{pd} = (t_{pdr} + t_{pdf})/2$$

Buffer

$V_{out} = V_{in}$



(iv) Rise time → (t_r)

- The time taken by the signal to rise from 10% of the max value to 90% of max value.

(v) fall time → (t_f)

- The time taken by the signal to fall from 90% of max value to 10% of the max value.

(vi) t_{cdr} → Rising contamination delay

- minimum time from input rising to o/p rising at 50% of max value of o/p.

(vii) t_{cdf} → falling contamination delay

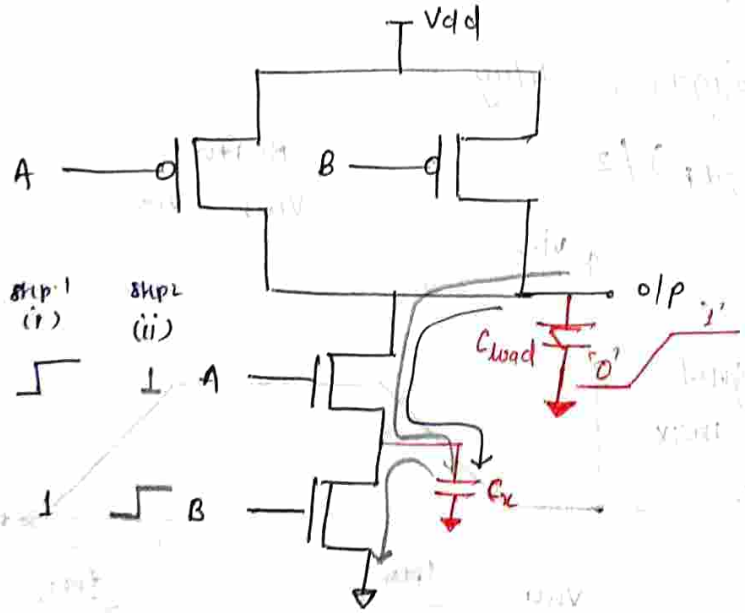
- minimum time from i/p falling/rising to o/p falling at 50% of max value of o/p.

(viii) t_{cd} → average contamination delay

$$t_{cd} = (t_{cdr} + t_{cdf}) / 2$$

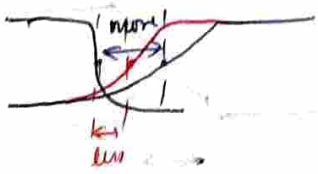
Note: When i/p changes, the o/p will change its old value for atleast the contamination delay & take on its new value in at most the propagation delay.

Eg: Understanding t_{cd} with example of NAND gate



A	B	$\overline{AB}$	
0	0	1	(2)
0	1	1	(2)
1	0	1	(1)
1	1	0	

- if only one i/p is toggling & other is constant then contamination delay is more.



- But when both A & B are toggling together then contamination delay is less.

Q: Why contamination delay is important?

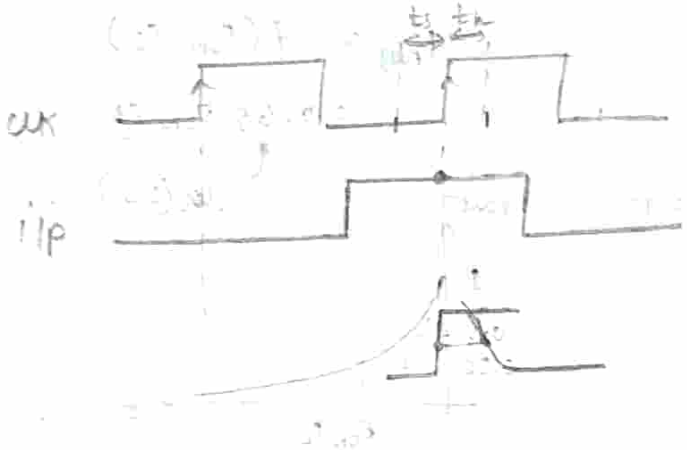
→ Contamination delay is important to validate hold time.

* Hold time error will appear if you don't consider contamination delay.

- Hold time is characterized by fastest delay. (t_{cd})
- Setup time is characterized by ~~slowest~~ slowest delay. (t_{pd})

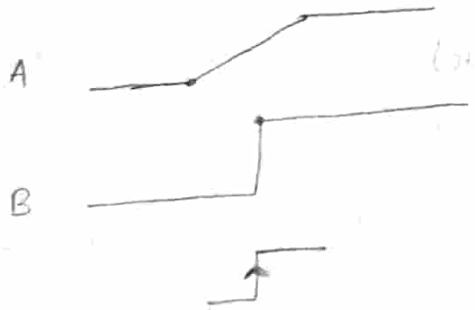
Setup time - The min. amount of time before the clock's active edge that the data must be stable for it to be latched correctly.

Hold time - The min. amount of time after the clock's active edge during which data must be stable.



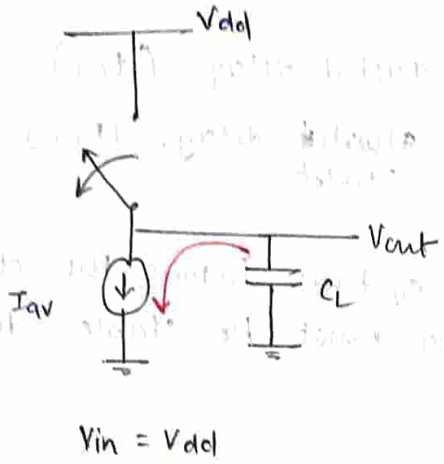
o/p := $\overline{i/p}$

- pipelining
- throughput



Delay models:

1. CMOS inverter propagation delay approach - 1

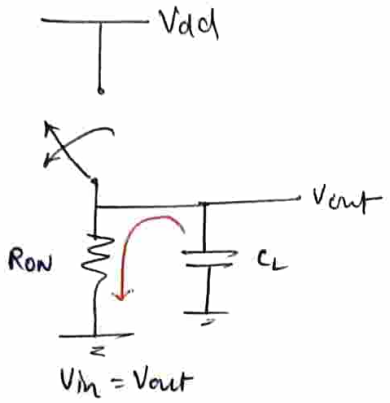


$$t_{PHL} = \frac{C_L V_{swing}/2}{I_{av}}$$

$$\approx \frac{C_L}{K_n V_{DD}}$$

$V_{in} = V_{dd}$

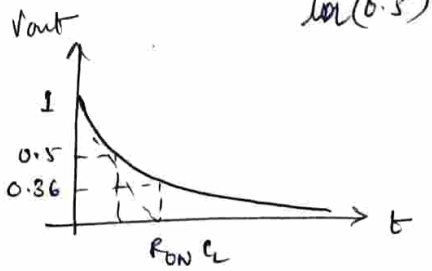
2. CMOS inverter propagation delay approach - 2 [simpler]



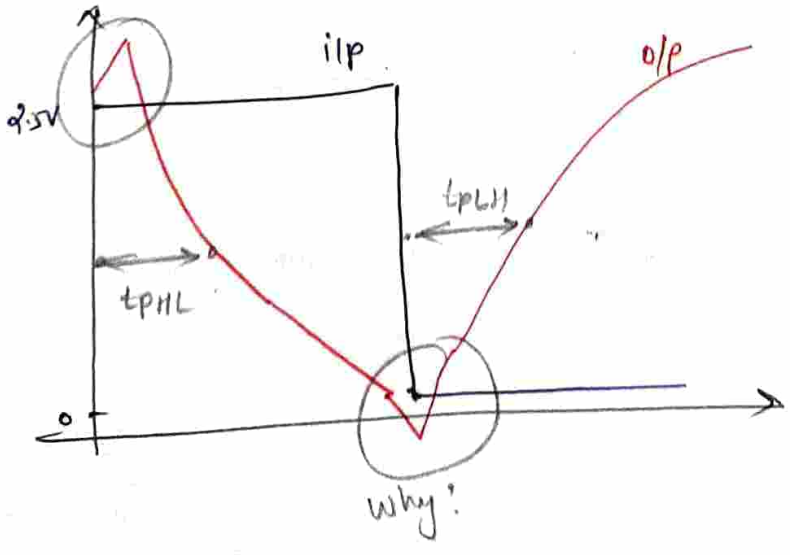
$$t_{PHL} = f(R_{ON} \cdot C_L)$$

$$= 0.69\, R_{ON} C_L$$

$\uparrow$ $\ln(0.5)$



Transient Response (of CMOS inverter)



Q. Why the peak in V_{out} is observed?

→ Due to miller capacitance

⇒ As the gate i/p goes from 'zero' to 'one' there is coupling with source & drain

- the source is gnd or vdd, which doesn't change much
- but drain is influenced by the change in i/p gate voltage

hence a little spike is observed.

or ⇒

there is diffusion layer here already but when the device is ~~turned~~ turned off this charge has to go somewhere.

- then half of the charge goes to drain region & half goes to the ~~drain~~ source region.

Also,

$$t_p = 0.69\, C_L \left[\frac{R_{eqn} + R_{eqp}}{2}\right]$$

where, $R_{eq} = \frac{1}{V_{DD}/2} \int_{V_{DD}/2}^{V_{DD}} \frac{V}{I_{Dsat}(1+\lambda V)}\, dV$

$$= \frac{3}{4} \frac{V_{DD}}{I_{Dsat}} \left(1 - \frac{7}{9}\lambda V_{DD}\right)$$

with, $I_{Dsat} = K' \frac{W}{L}\left[(V_{DD} - V_T)V_{Dsat} - \frac{V_{Dsat}^2}{2}\right]$

Q. How to reduce the glitches ??

→ Add extra cap. in parallel to the diffusion cap.

$\because \Delta Q = C\Delta V$

on $\uparrow C$, $\Delta V \downarrow$

$\therefore$ glitches ↓

Q. Why we don't want glitches ?

→ Glitches may lead to breakdown.

How to keep the delay less ?

1. Keep load capacitance small.
2. Keep R_{eq} small
3. Increase V_{DD}, R_{eq} ↓, delay ↓

but at V_{DD} near subthreshold region, delays ↑ exponential

1. By keep the capacitance small :-

- By ↓ C_L.
- By reducing the self-loading capacitance.
  — capacitance due to the g drain & source diffusion

2. By reducing the R_{ON} of the device (Nmos or Pmos)

** on increasing the width of the device (w_p & w_n)

(i) diff. cap ↑ , self-loading cap ↑

$\therefore$ C_{load} (overall) ↑.

(ii) $R \propto \frac{1}{w}$ $\therefore$ w ↑, R ↓

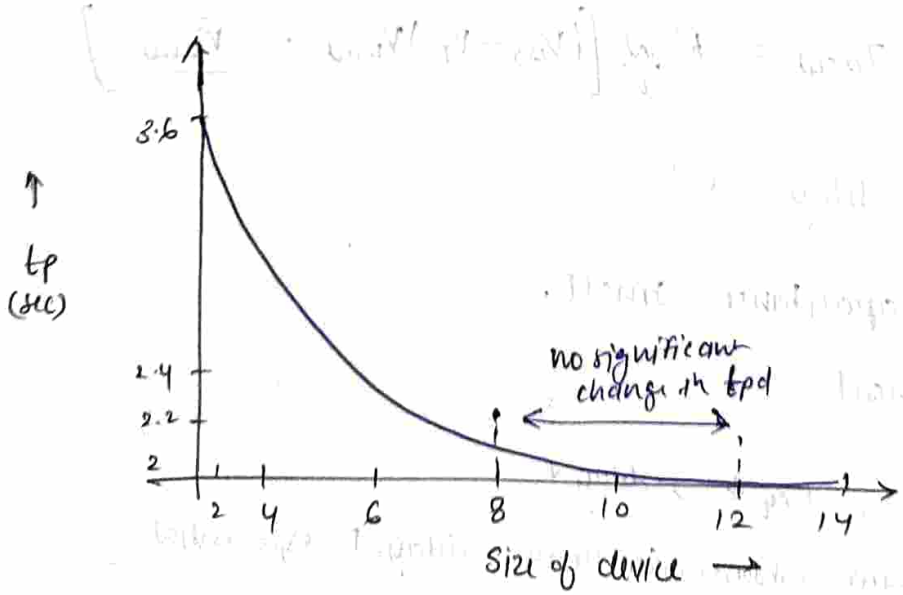
⇒ But effectively on ↑ w, t_p ↓

± But on further increasing the value of w, the change in w is not significant.

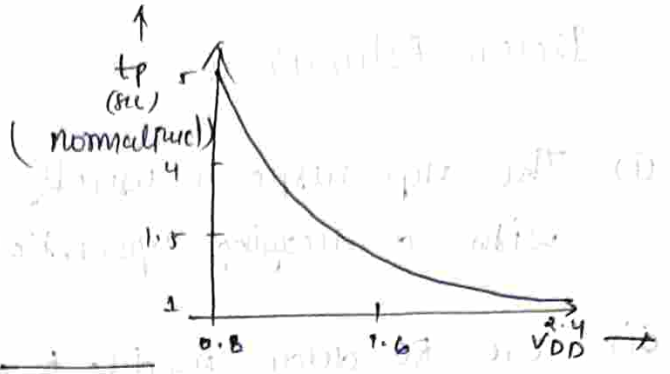
Q. How large the logic gate should be to achieve least delay? (Ans?)

3. By increasing the V_{DD}. R_{eq} ↓ } from formula on previous page.

on ↑ V_{DD}, $I_{D,sat}$ ↑, ⇒ R_{eq} ↓ }



Delay as a func^n of V_{DD}.



$$t_{PHL} = 0.69 \frac{3}{4} \frac{C_L V_{DD}}{I_{Dsat_n}}$$

$$= 0.52 \frac{C_L V_{DD}}{\left(\frac{W}{L}\right)_n K_n' V_{DSat_n} \left(V_{DD} - V_{TH} - \frac{V_{DSAT_n}}{2}\right)}$$

Q. What are ways of reducing rise & fall time?

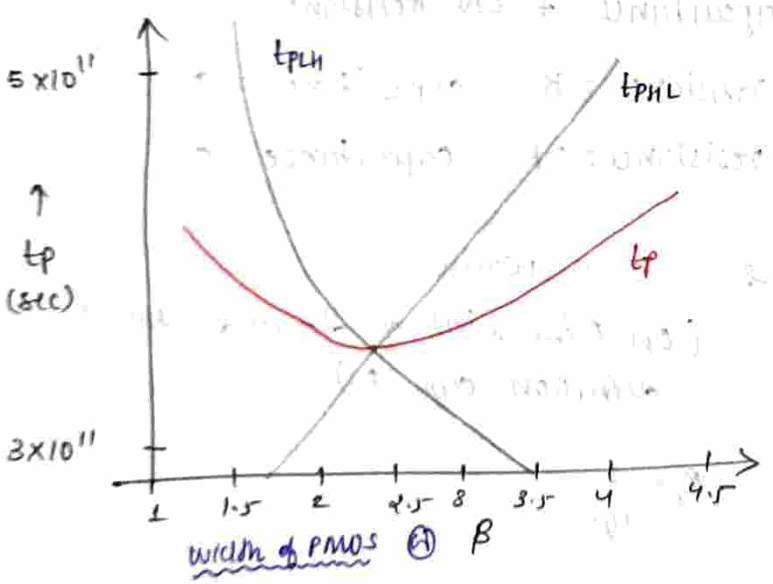
→ • rise delay & fall delay (t_{pd}) are for calculating delays.

• rise & fall time are only calculating powers.

* the way we are reducing t_{pd} we can reduce t_r & t_f also.

— load cap ↑ ⇒ t_r & t_f ↓.

NMOS - PMOS : β ratios



Q. On changing $\frac{W_p}{W_n}$
• how cap changes
• how delay changes

Q. for ↑ cap how $\frac{W_p}{W_n}$ will change

$\frac{W_p}{W_n}$ → β ratio of PMOS & NMOS

① as β ↑, size of PMOS ↑, more current will flow & its less time to charge the cap ∴ t_{PLH} ↓.

② as β ↑, size of PMOS ↑ but NMOS size ↓ its less more time is required to discharge the cap. ∴ t_{PHL} ↑.

It β-ratio $\left(\frac{W_p}{W_n}\right)$ should lie b/w 1.5 to 2.6

(for 65 nm technology)

③ Overall propagation delay first ↓ as PMOS size ↑ & time taking to charge the cap ↓ & after further ↑ in PMOS size due to significant effect of self loading t_p starts ↑ as time taking to charge the load cap increases.

Delay Estimations

Weste
Pg. 146 (4.3)
Pg 155 (4.4)

(i) The step response usually looks like a 1st order RC response with a decaying exponential.

(ii) Use RC delay models to estimate delay
- → C = total capacitance on o/p node
- → Use effective resistance R
- → So that $t_{pd} = RC$

(iii) characterize switches by finding their effective R.
- — Depends on avg. current as gate switches

(iv) Too inaccurate to predict current at any given time
- — But good enough to predict RC delay.

RC Delay model

- RC equivalent ckt of MOSFET is:

(i) Ideal switch + capacitance + ON Resistance

(ii) One nMOS has, resistance = R, capacitance = C

(iii) One pMOS has, resistance = 2R, capacitance = C

* $\downarrow R \propto \frac{1}{W \uparrow}$ & $C \propto$ width
(on ↑ the width of drain & source diffusion cap. ↑)

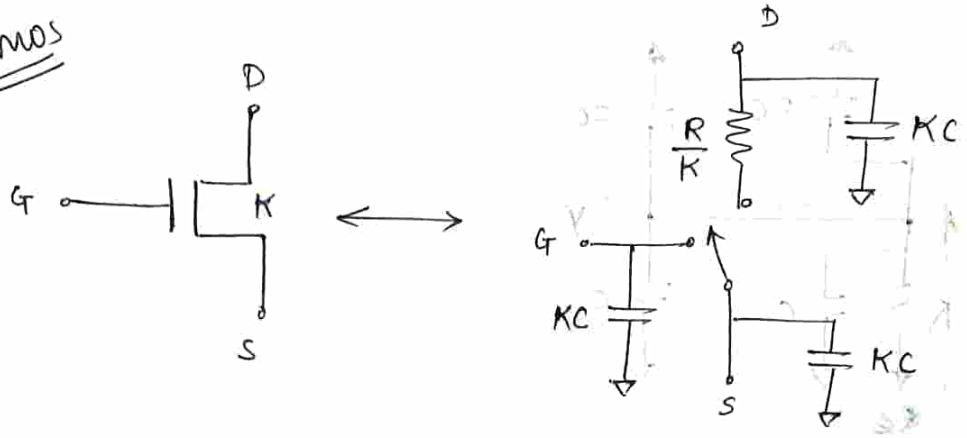
* $R_n \propto \frac{1}{\mu_n}$ & $R_p \propto \frac{1}{\mu_p}$

& let $\mu_n = 2\mu_p$ $(\mu_n > \mu_p)$
$(\because R_n < R_p)$

if $R_n = R$

$\therefore R_p = 2R$

NMOS

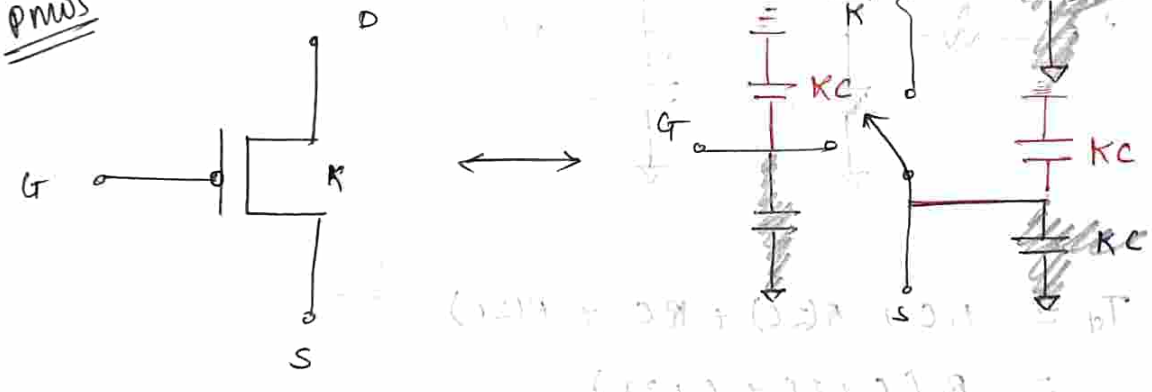


where K represents the K-times the width of the unit device size

if $w_{n(old)} = w$

$\therefore w_{n(new)} = K \cdot w$

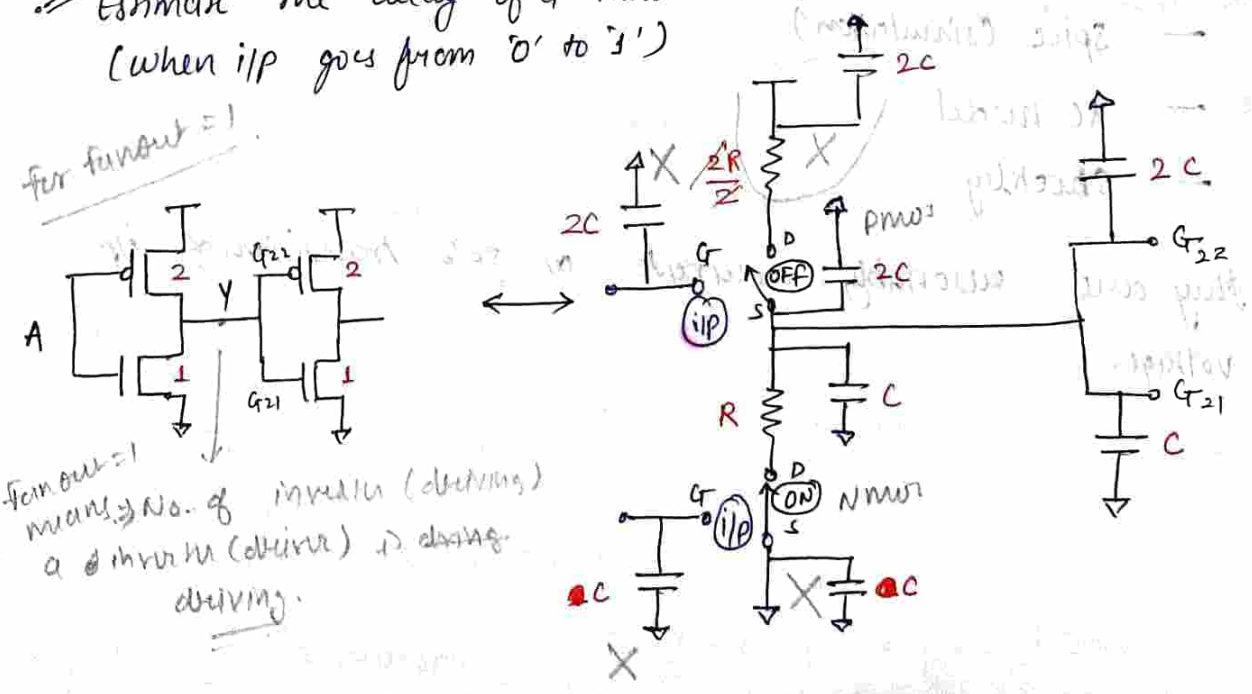
PMOS



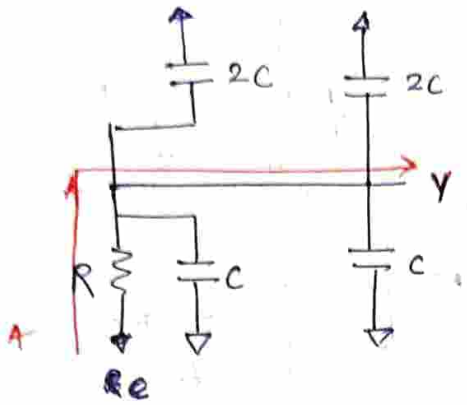
* So make the inverter with same delay for both NMOS & PMOS we take $w_p = 2w_n$ (i.e, K=2 for PMOS) (& K=1 for NMOS) (or) $\frac{K_p}{K_n} = 2$.

Inverter Delay estimation

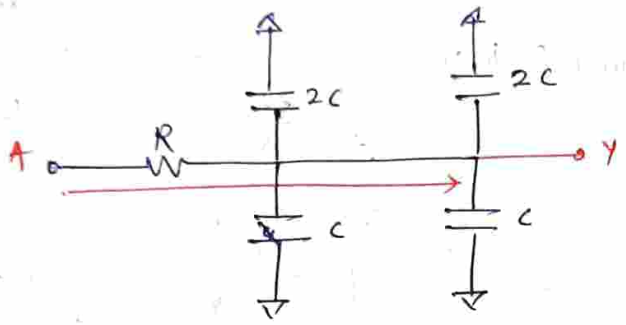
Ex: Estimate the delay of a fanout-1 inverter? (when i/p goes from '0' to '1')



→



Now we have to calculate delay for path AY



$$T_d = RC + R(2C) + RC + R(2C)$$
$$= R(C + 2C + C + 2C)$$
$$= R(6C)$$
$$= 6RC$$

//

** When try to calculate the delay in the path A to Y we can use multiple Delay models.

- Spice (simulation)
- RC model
- Shockley

They are reasonably accurate at 50% transition of i/p voltage.

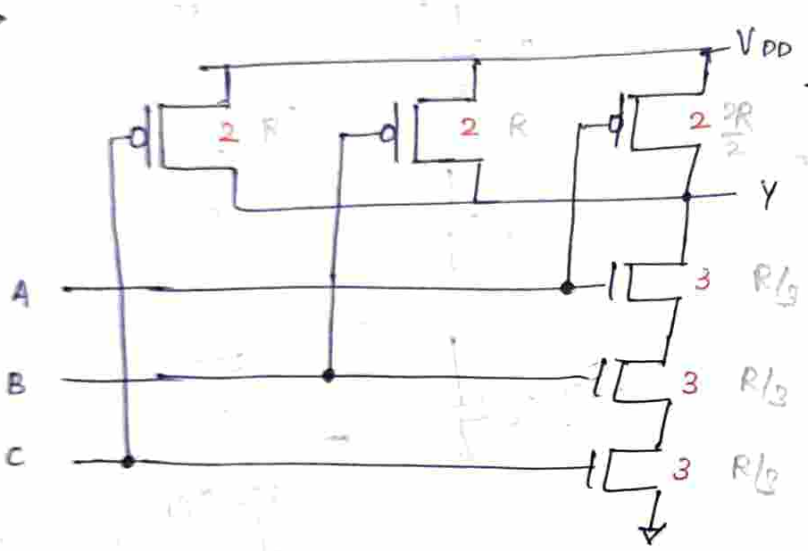
Example : "3-i/p NAND gate"

Q. Sketch a 3-i/p NAND with transistor width chosen to achieve effective rise & fall resistances equal to a unit inverter (R). Annotate the gate with its gate & diffusion capacitance. Assume all diffusion nodes are connected. Also sketch equivalent ckt for: (Also calculate delay)

(i) falling o/p transition (Both case & worst case) → discharging

(ii) Rising o/p transition (Both case & worst case) → charging

→



A	B	C	$\overline{ABC}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Q. How to take the value of size of PMOS & NMOS (K_p & K_n)?

→ The size of PMOS & NMOS should be such that the (charging) ≥ (discharging)

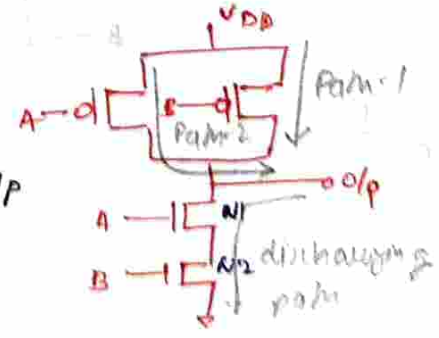
R_{ON} for NMOS = R_{ON} for PMOS = R_{ON} (for the worst case)

(a) I_D for NMOS = I_D for PMOS ~~=~~ of inverter

(i) If we want to charge a C load at o/p 'Y' then we can charge it through any path-1 or path-2 (for the worst case) i.e, not considering both path-1 & path-2 together

∴ Both the path should have Resistance = R.

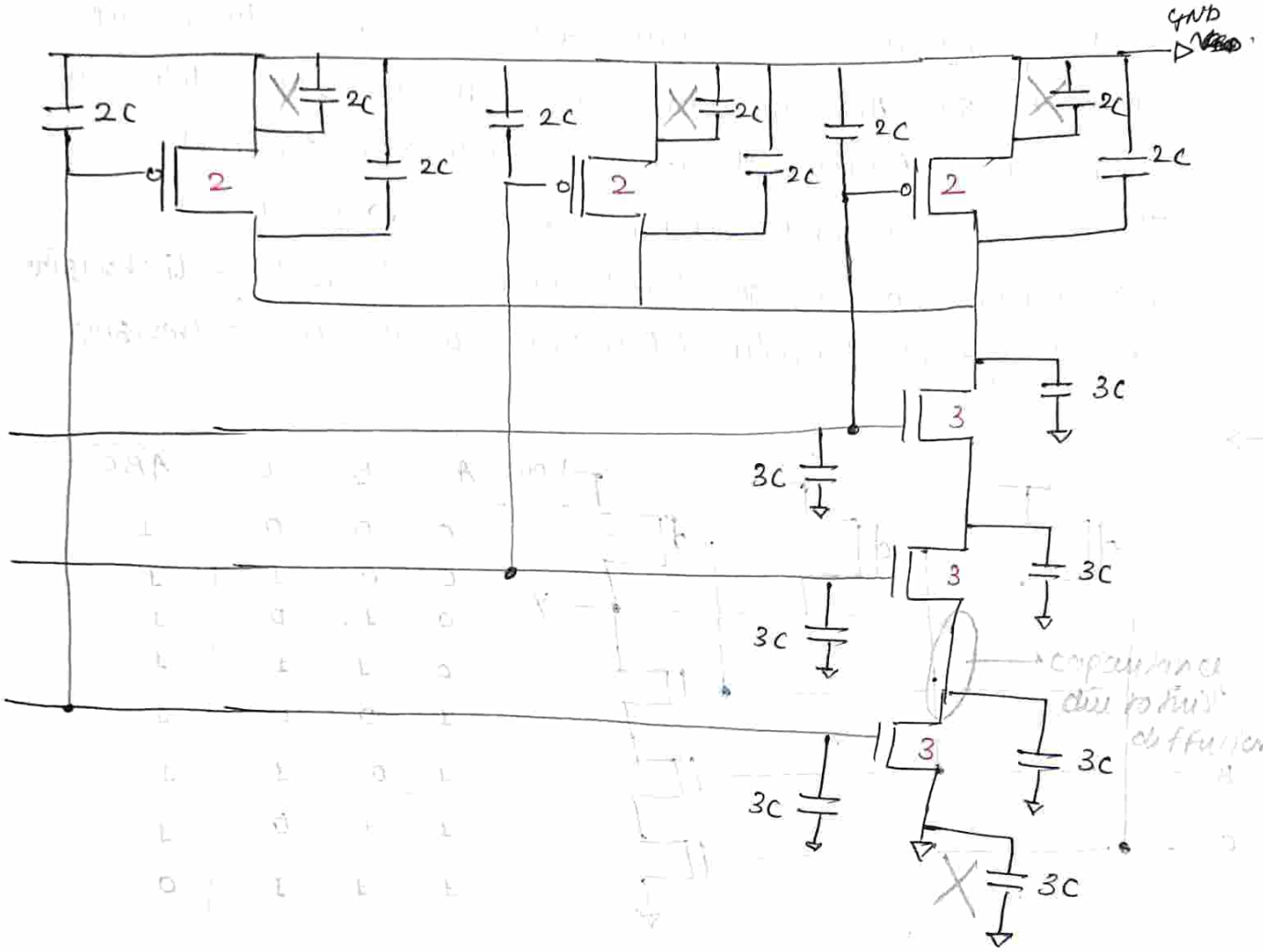
∴ $K_p = 2$

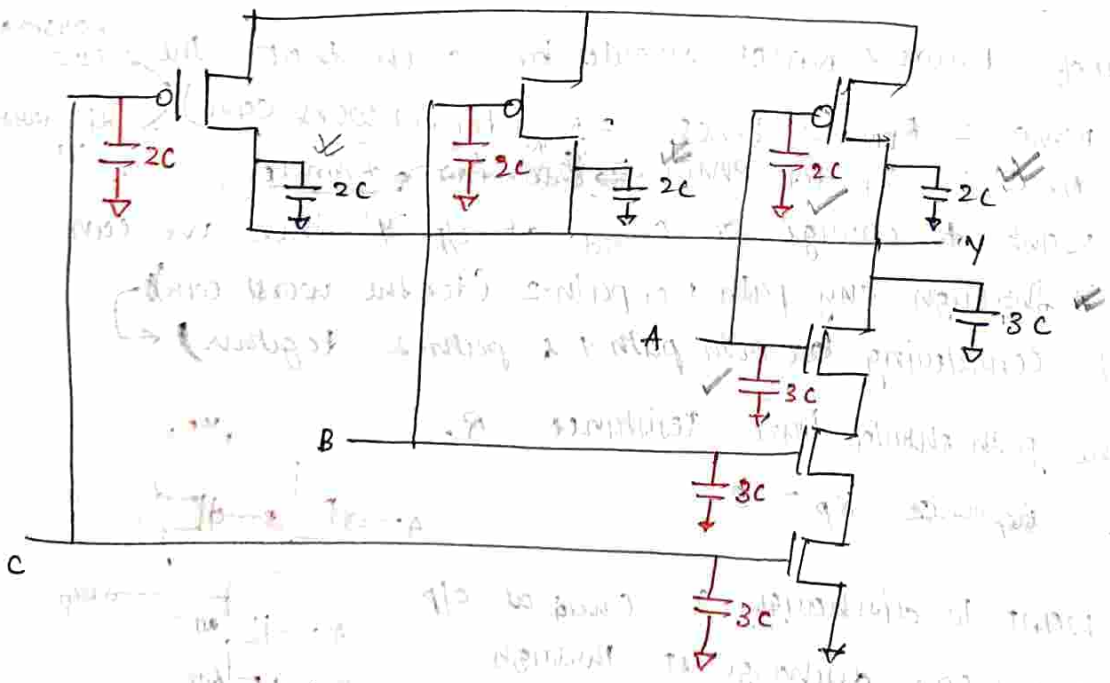


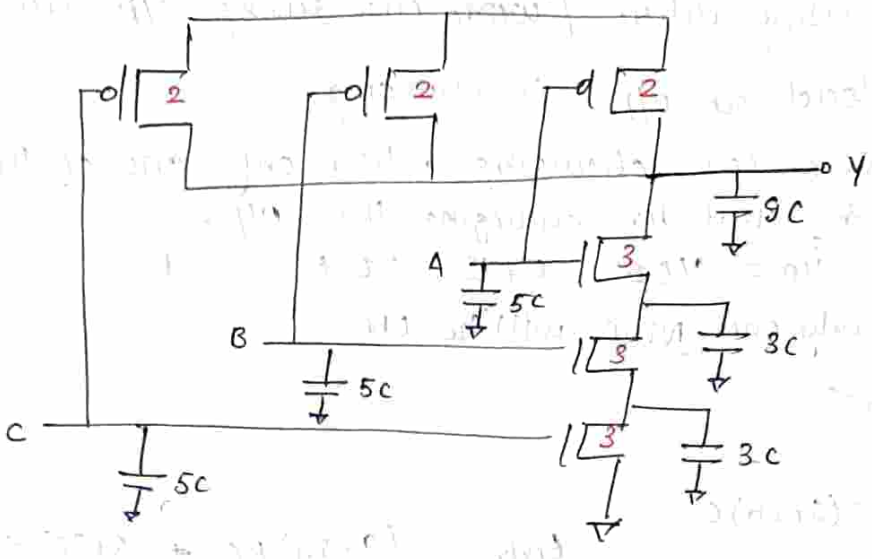
(ii) If we want to discharge a C load at o/p 'Y' then we can discharge it through discharging path, for that resistance of N1 should be R/2 & for N2 $\frac{R}{2}$

∴ $R_{discharging\ path} = \frac{R}{2} + \frac{R}{2} = R$ [as they are in series.

- Annotate the 3-ilp NAND gate with gate & diffusion (drain & source) capacitance.

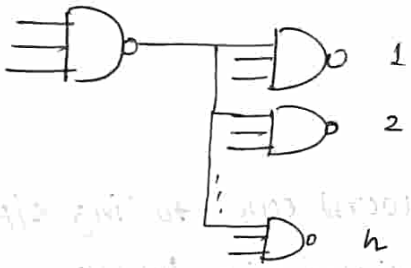


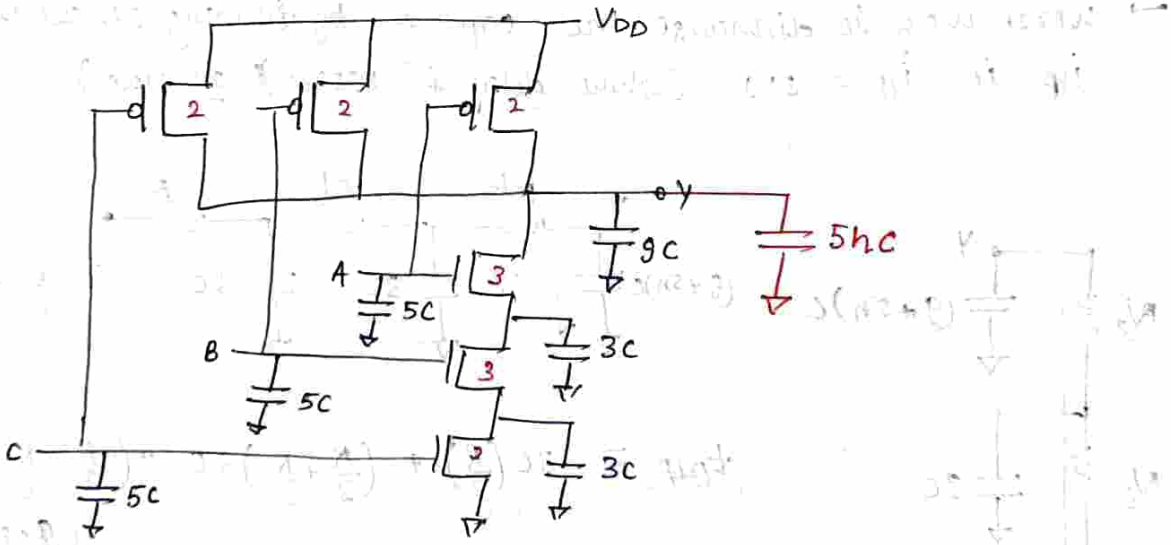




(ii) Worst case rising delay

* Suppose this 3-i/p NAND gate is driving 'n' identical 3-i/p NAND gates (fanout = n)





* Now we have to calculate the different kind of delays for the o/p to charge & discharge.

(i) Worst-case rising delay [worst case rising o/p transition]

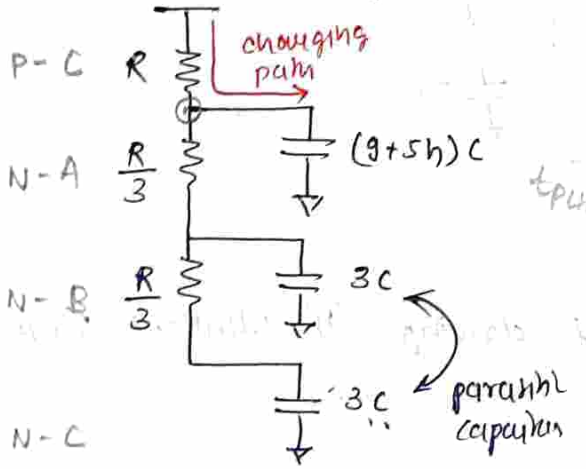
→ when load at o/p is charging.

→ it is worst case charging when only one of the PMOS is ON & and for charging the o/p.

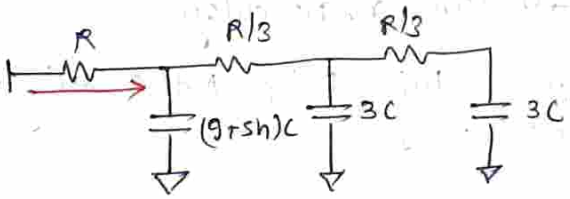
ie, i/p = 110 .. ~~[illegible]~~

So only one NMOS will be off

111 → 110 (charge everything)



ignoring the branching



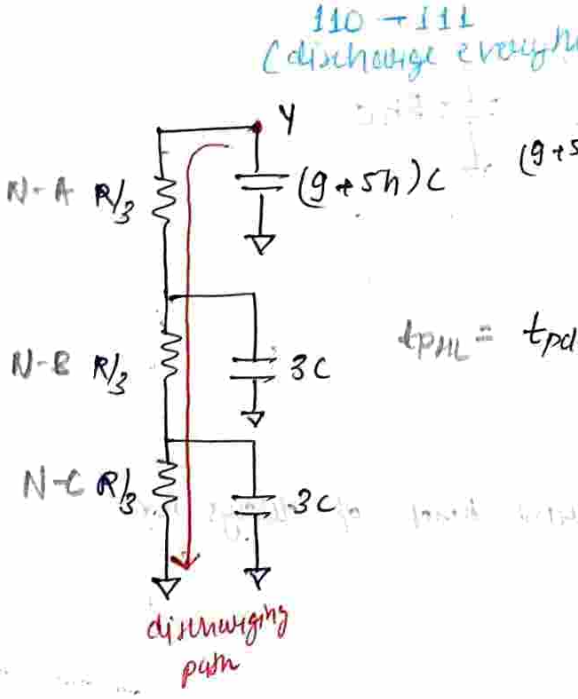
$t_{pLH} = t_{pdr} = (9+5h)RC + \frac{R(3C) + R(3C)}{(15+5h)RC}$

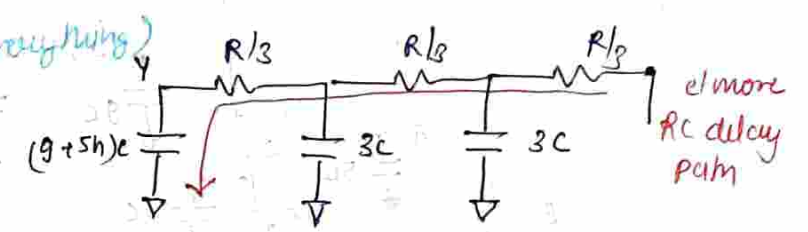
(ii) Worst case falling delay [worst case falling o/p transition]

→ when load at o/p is discharging through NMOS

→ worst way to discharge the cap is by turning ON all the i/p ie, i/p = 111 (where delay is worst or max)

110 → 111 (discharge everything)





$t_{pHL} = t_{pdf} = 3C\left(\frac{R}{3}\right) + \left(\frac{R}{3}+\frac{R}{3}\right)3C + \left(\frac{R}{3}+\frac{R}{3}+\frac{R}{3}\right)(9+5h)C$

$= RC + 2RC + (9+5h)RC$

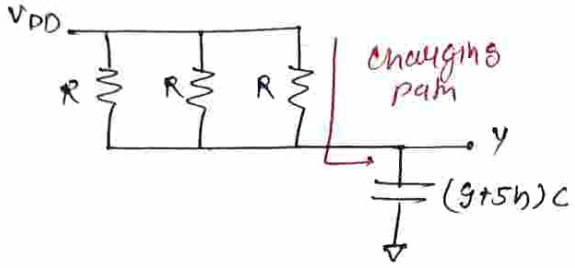
$= (12+5h)RC$

(iii) Best case rising delay

→ the best case will be when all the PMOS together charge the cap. at o/p.

ie, i/p = 000

when only o/p cap will be charge (or) least capacitor needs to be charged



111 → 000

sab discharge ho gya

only o/p is charging

$t_{cdr} = \frac{R}{3} \times (9+5h)C$

$= \left(\frac{9+5h}{3}\right) RC$

$\{ t_{cdr} < t_{pdr} \}$

$\{ t_{cdf} < t_{pdf} \}$

(iv) Best case falling delay

→ the best case for discharging will be when

$i/p = \overset{ABC}{\text{B}}001 \rightarrow \text{B}011 \rightarrow 111$

→ where we are discharging the 3C of N3 (N-C) to GND then we discharge the 3C of N2 (N-B) to GND then we discharge the (9+5h)C of N1 (N-A) to GND.

for $\overset{ABC}{i/p = 001}$: delay = RC

$i/p = 011$: delay = 2RC

$i/p = 111$: (9+5h) RC = t_{cdf}

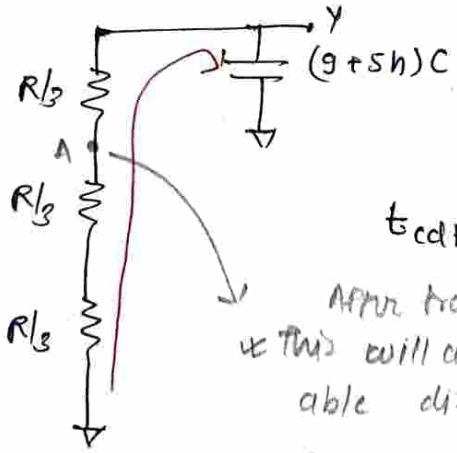
} when i/ps are given one by one.

* The best case is that the bottom two nMOS transistors are already ON when top one (N-1/NA) turns ON.
(N-B, N-C) [i/p transition 011 → 111]

011 → 111

↓ other cap are discharged except for o/p capacitance

↓ Now only o/p is discharging



$t_{cdf} = (9+5h)CR$

After transition of 011 → 111

* This will act as GND & hence 3C will not be able discharge complete (9+5h)C

* for 100 C at 'A' then we can say charged (9+5h)C will discharge

$\therefore t_{cdf} = (9+5h)\frac{C}{R}$

2. Logical Efforts

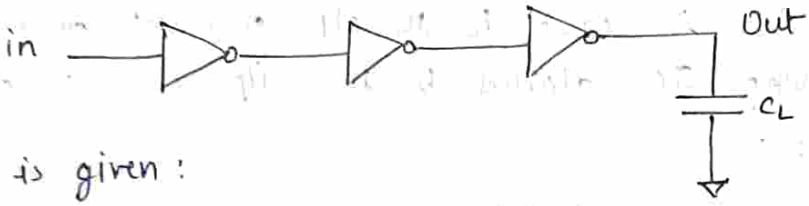
- Delay
- ~~Size~~ sizing
- No. of stages

Part-1

- Previously we have calculated the ~~g~~ delay when size of the MOSFETs are given.
- Now, we should be able size the inverter for the desired delay.

* The concept will help us to size the devices & chains of ~~devices~~ gates.

Inverter chain :



If C_L is given :

Q. * — How many stages are needed to minimize the delay?

Q. — How to size the inverters?

→ may need some additional constraints

Delay in logic gates (CMOS inverter)

- First of all we have to express delays in process-independent unit. [Delay should be same for different technologies like 45nm, 90nm, 180nm]

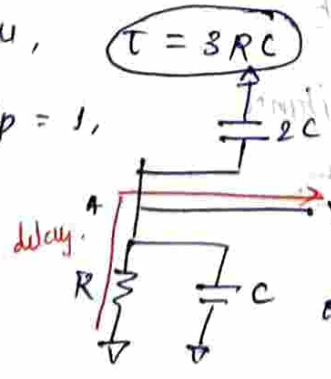
$$\therefore d = \frac{d_{abs}}{\tau}$$

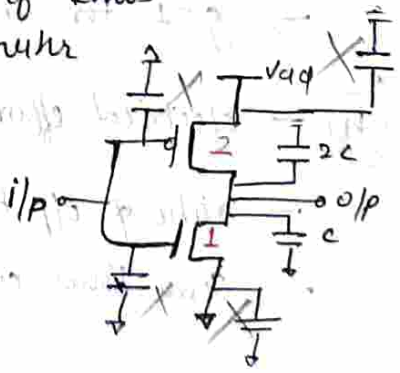
d_{abs} → absolute delay

τ → delay of cmos inverter

where, $\tau = 3RC$

for i/p = 1,





delay = $RC + R(2C)$

$= 3RC$

$\tau = 3RC$ — delay of cmos inverter w/o any fanout.

$\approx$ 3psec in 65nm process

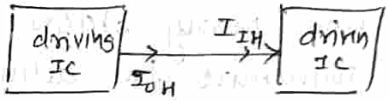
$\approx$ 60ps in 0.6μm process

$\tau \rightarrow$ (=3RC) ** if the inverter is ideal & has no self loading then the delay will be one fanout.

Fanout → (h) * It is the number of cmos logic i/p that can be driven by one cmos logic o/p.

Therefore, fanout is equal to the o/p current of the driving IC divided by the i/p current of driven ICs:

$$\text{fanout} = \frac{I_{OH}}{I_{IH}} \text{ (or) } \frac{I_{OL}}{I_{IL}}$$



Delay in logic gates:

- Delay has two components

$$d = f + p \Rightarrow \boxed{d = gh + p}$$

(* it's how big is the o/p.)

- $f \rightarrow$ effort delay $= gh$ (stage effort)
  - * (g) → logical effort (relative to inverter)
    - measures relative ability of gate to deliver current.
    - $g = 1$ for inverter.
  - * (h) → electrical effort $= \frac{C_{out}}{C_{in}}$
    - ratio of o/p to i/p capacitance
    - Some times called fanout.

- (P) → parasitic delay
  - — Represents delay of gate driving no load.
    (delay of gate w/o any load)

∴ $p = 3RC$ for inverter

(n) $d_{ans} = 3RC$ & $\tau = 3RC$] $d = 1$ for an inverter.

"1 unit delay"

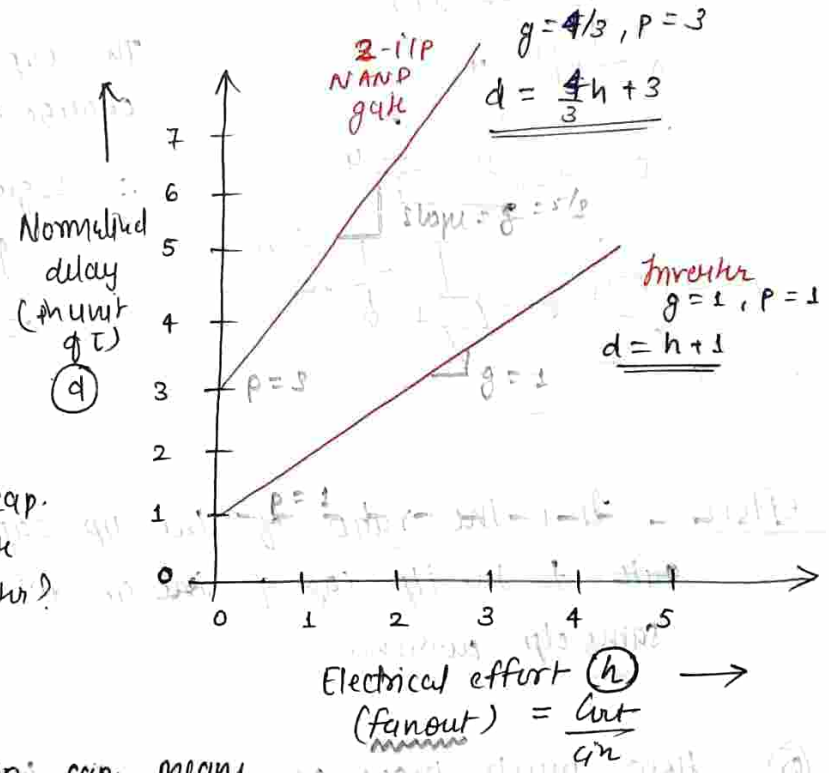
Delay plot :

Q. How do I apply force to a 2-i/p NAND gate for example? So that it gives o/p that we desire.

→ By changing its i/p capacitance.

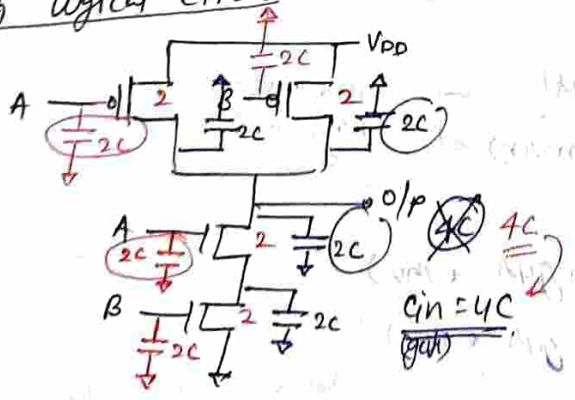
Q. Can you see the relation b/w i/p cap. of two i/p NAND gate & i/p cap. of inverter?

→ Yes.



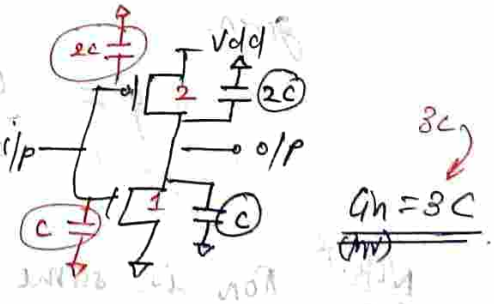
* By changing the i/p cap. means, when apply the i/p signal it will charge the gate – body capacitance (ie, gate capacitance) or i/p capacitance.

Computing Logical effort



What is the cap. need to charge for this 2-i/p NAND gate?

→ 4C



What is the cap needed to charge cmos inverter?

→ $C_{in} = 3C$ (inv)

$g = \frac{4C}{3C} = \frac{4}{3}$

Therefore,

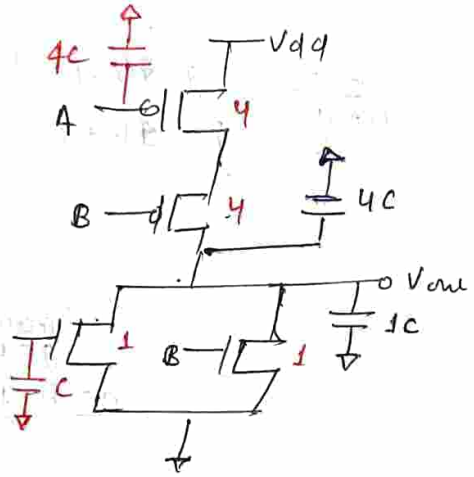
logical effort → how much more effort do I need to put to charge cap. in any gate as compared to the inverter.

$$\therefore g = \frac{C_{logic}}{C_{inverter}} =$$

Q. Logic effort of NOR-2 ?

$4C + C = 5C$

$C_{in} = 5C$



The cap. that we need to charge for NOR-2 is 5C.

∴ logical effort

$$g = \frac{5C}{3C} = \frac{5}{3}$$

Logical effort: It is the ratio of the i/p capacitance of the gate to the i/p cap. of an inverter delivering the same o/p current. (Done same power also)

(or) how much more effort do I have to put to charge the i/p cap. of gate as compared to inverter, to get the same response as the inverter would have given.

$$g = \frac{C_{in}(gate)}{C_{in}(inverter)}$$ ← gate & inverter have total size

Note: R_{ON} is same (for gate & inv)

I_D is same (for gate & inv)

⇒ Both will done same power as $V_{DS} = I_D \cdot R_{ON}$.

(sizing of mos)

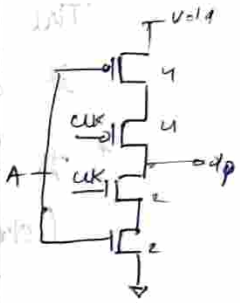
** * These sizings are done to ensure that, we will have same equivalent resistance both for rise & fall time. And not for the delay we have achieved. (wrong)

* Lower the logical effort, lower the delay because $\downarrow d = gh + f$.

Logical Effort for common gates

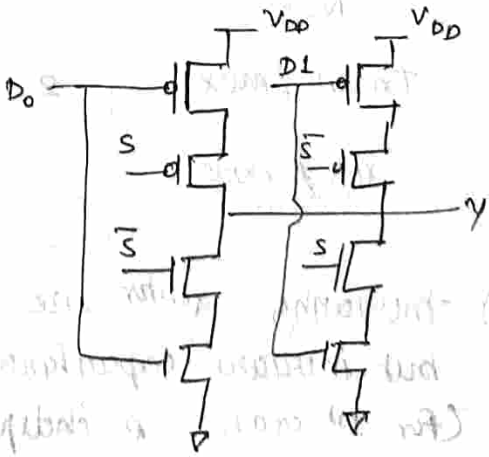
Q. Does size of gate (w_p, w_n or k) affect the logical effort or not.

Gate Type	Number of inputs				
	1	2	3	4	n
Inverter	1				
NAND		4/3	5/3	6/3	(n+2)/3
NOR		5/3	7/3	9/3	(2n+1)/3
Tristate / mux	2	2	2	2	2
XOR, XNOR		4,4	6,12,6	8,16,16,8	



Logical Efforts

- For complex gates like XOR, XNOR it can have different logical effort for different i/p combination
  - * Logical effort is strong func^n of no. of i/ps.
- Multiplex built from ganged tristates, has logical effort of 2. independent of no. of i/ps.
  - ~~This means~~ But parasitic delay with multiplexer size.
  - * Therefore it is recommended to use tree of 4-i/p mux. which is fastest. (i.e, least delay)



Q. How can you benifit from that some gate have higher logical efforts and some has less for XOR & XNOR gate?

→ • We can put the ip which don't switch much at larger logic efforts. (It will save power)

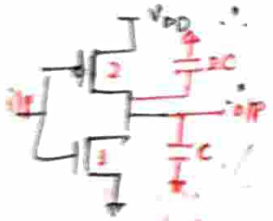
• The gate ip which has lesser logical effort you will apply signal which arrives late.

Parasitic delay of common gate :

• ✓ The parasitic delay of a logic gate, is the delay of the logic gate when it drives zero load. (delay due to self load only)

• ✓ It can be calculate by counting only diffusion capacitance (drain + source) on the o/p node.

Eg: Inverter has 3-ip of diffusion capacitance on the o/p.



∴ parasitic delay = 3RC = d_abs (P_abs)

Normalized parasitic delay, $d = \frac{d_{abs}}{\tau} = \frac{3RC}{3RC} = 1$

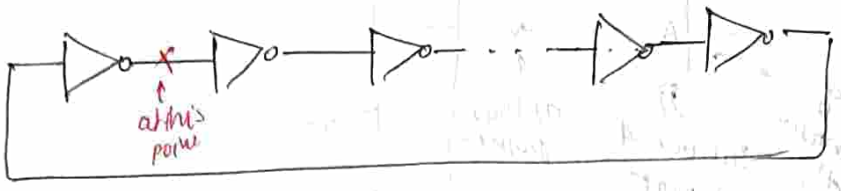
∵ $\tau = 3RC$ for inverter $\Leftrightarrow$ $P = \frac{P_{abs}}{\tau}$

Gate type	Number of inputs				
	1	2	3	4	n
Inverter	1				
NAND		2	3	4	n
NOR		2	3	4	n
Tristate / MUX	2	4	6	8	2n
XOR / XNOR		4	6	8	

Normalized parasitic delay (P)

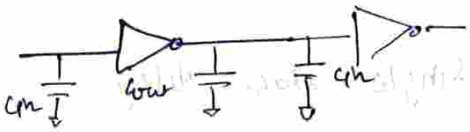
(*) Increasing transistor size reduces resistance but increases capacitance, so parasitic delay (for 1st order) is independent of logic gate size.

Example : Calculate the frequency of N-stage ring oscillator (where N → odd number)



→ For single stage,

(i) logical effort, $g = 1$.



(ii) Electrical effort (how many gates are connected to one inverter) = 1
(ie, fanout)

$h = 1$

(or) $h = \frac{C_{out}}{C_{in}} = 1$ (as both are same capacitance)

(iii) parasitic delay, for inverter = 1

$\therefore p = 1$

(iv) stage delay, $d = gh + p$

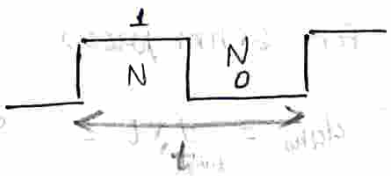
$d = 1 * 1 + 1 = 2$

$\underline{d = 2}$.

Logical effort
* For any inverter at any stage, how much effort does the previous stage needs to apply in comparision to an inverter to get the same o/p as an inverter would give.

For N-stage

(v) Frequency of N-stage,



waveform at the i/p of oscillator.

total delay = $2N \times d$ → we have to go through the N-stages twice & each stage has delay = d.

$= 2N \times 2$

$= 4N$

$\therefore$ Freq. = $\frac{1}{4N}$

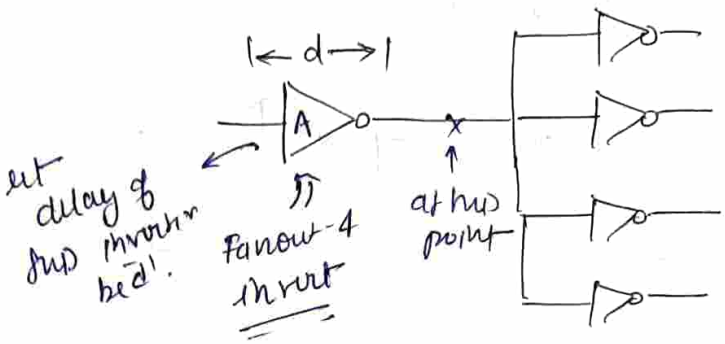
$200 = \frac{1}{2 \times 31 \times D} \Rightarrow D = \frac{1}{400 \times 31}$

(*) $N = 31$ stages of ring oscillator in 0.6μm technology have freq. of ≈ 200 MHz. [$d_{abs} = d_{total\ delay} \times \tau$] for $\tau = 60ps$

$\Rightarrow d_{abs} = 4N \times \tau = 4 \times 31 \times 60ps = 7.2$ nsec

$f_{abs} = \frac{1}{d_{abs}} = 138.9$ MHz

Example : Estimate the delay of a fanout-of-4 (FO4) inverter.



→ Single stage delay :

(i) logical effort , $g = 1$

(ii) Electrical effort , $h = 4$ (fanout)

(iii) Parasitic delay , $p = 1$

(iv) stage delay , $d = gh + p = 4 + 1$

$d = 5$

* The FO4 delay is about (i) 300ps in 0.6 µm process

→ $[\tau = 60psec] = 3RC$ ⟶ 1 unit delay

for 0.6 µm process $[\tau = 60psec] = 3RC$

(i) $d_{abs} = d \times \tau = 5 \times 60psec$

$= 300\ psec$

(ii) for 65nm process $(\tau = 3psec) = 3RC$

$d_{abs} = d \times \tau = 5 \times 3\ psec$

$= 15\ psec$

Note : We will only look at driver 'A' when we are calculating parasitic delay.

— Because parasitic delay doesn't depends on load.

— It only depends on self-load (cap.) of driver (logic) gate.

Part-3

Multistage logic N/W

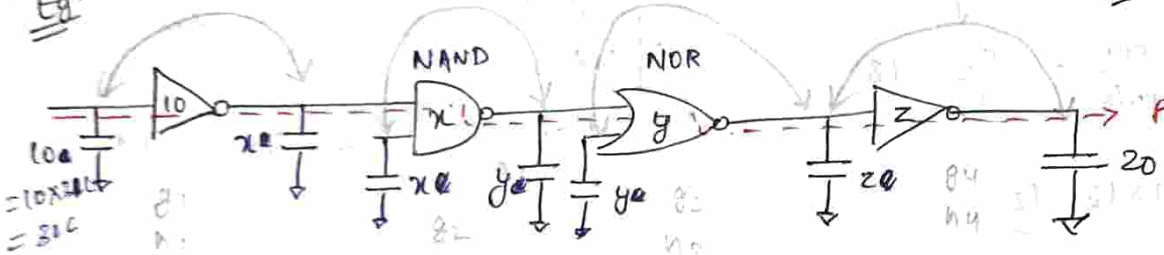
- logical effort generalizes to multistage N/W

(i) ① Path logical effort ⇒ $G = \Pi g_i$

(ii) ② Path Electrical effort $H = \frac{C_{out-path}}{C_{in-path}}$

multistage logic effort (iii) Path Effort $F = \Pi f_i = \Pi g_i h_i$

Eg:



Q. How do we find the value of size is x, y, z?

- All the capacitances drawn are i/p capacitances of logic gates.

$g_1 = \frac{C_{in(inverter)-given}}{C_{in(inverter)-known\ of\ same\ size}} = \frac{10\times3RC}{10\times3RC} = 1$ ← for calculating logical effort of any gate size 'x' we compare it with inverter of size x.

NAND, $g_2 = \frac{C_{in\ (NAND)\ of\ size\ 'x'}}{C_{in\ (inverter)\ of\ size\ 'x'}} = \frac{x\times4C}{x\times3C} = \frac{4}{3}$

$g_3 = \frac{(2n+1)}{3} = \frac{5}{3}$ ← NOR

$g_4 = 1$

$G = g_1 g_2 g_3 g_4$

$= 1 \times \frac{4}{3} \times \frac{5}{5} \times 1$

$= \frac{20}{27}$

*** Logical effort is independent of size. (How?)

* Electrical effort depends on size

$h_1 = \frac{C_{out\,1}}{C_{in\,1}} = \frac{xc}{10c} = \frac{x}{10}$

$h_2 = \frac{C_{out\,2}}{C_{in\,2}} = \frac{yc}{xc} = \frac{y}{x}$

$h_3 = \frac{C_{out\,3}}{C_{in\,3}} = \frac{zc}{yc} = \frac{z}{y}$

$h_4 = \frac{C_{out\,4}}{C_{in\,4}} = \frac{20c}{zc} = \frac{20}{z}$

(m1) $H = \Pi h_i$

$H = h_1 h_2 h_3 h_4$

$= \frac{xyz \times 20}{10\ xyz} = \frac{20}{10} = 2$

(m2) $H = \frac{C_{out-path}}{C_{in-path}} = \frac{20c}{10c} = 2$

Q: Can we write $F = GH$?

→ NO (now?) → due to path Branching.

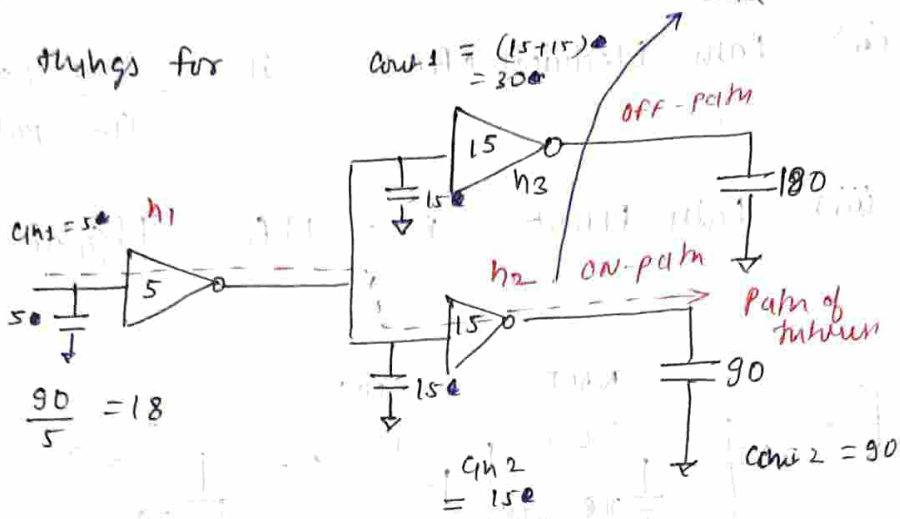
$F \neq GH$ but $F = GBH$

Paths that Branch

we are not considering the self-load here.

- We have to find things for below path only.

Eg:



(i) $G = g_1 g_2$

$= 1 \cdot 1 = 1$

$H = \frac{C_{out\text{-}path}}{C_{in\text{-}path}} = \frac{90}{5} = 18$

$GH = 1 \times 18 = 18$

(ii) $h_1 = \frac{C_{out}}{C_{in2}} = \frac{(15\epsilon + 15)}{5} = 6$

h_1 = fanout = 2 only when the size of all the inverters are same.

$h_2 = \frac{C_{out2}}{C_{in2}} = \frac{90\epsilon}{15\epsilon} = 6$

(i) $= \frac{15 \times 3RC + 15 \times 3RC}{5 \times 3RC} = \frac{15 + 15}{5} = 6$

$\therefore$ F (path effort) $= g_1 h_1 g_2 h_2 =$

$= 1 \times 6 \times 1 \times 6 = 36$

$= 2 \times 18$

$= 2GH$

Q: How does this '2' come? → This is due to branching. These two things can be said as branches but if two branches were of different sizes we would not be able to put it as branches.

if they are same sizes.

Branching Effort [like in Elmore delay]

- It accounts for branching b/w stages in path

$b = \frac{C_{on\text{-}path} + C_{off\text{-}path}}{C_{on\text{-}path}}$

Here we are also considering the delay of branches which are not in our own path as we do in Elmore delay.

$\therefore B = \prod b_i$

$\therefore \prod h_i = BH$

$\therefore$ path effort: $F = GBH$

This is only valid when all the branches are of same size.

Multistage Delay

(i) (4) Path Effort Delay : $D_F = \Sigma f_i$ $\qquad d = gh + p$

(ii) (5) Path Parasitic Delay : $P = \Sigma p_i$

(iii) Path Delay : $D = \Sigma d_i = D_F + P$

$= \Sigma (f_i + p_i)$

* The product of stage effort ($f = gh$) is path effort (F).

(3) $F = \Pi f_i$

* The sum of stage effort ($f_i = g_i h_i$) is path effort delay (D_F).

(2) $D_F = \Sigma f_i$ $\qquad f_i \rightarrow$ stage delay

Designing Fast Circuit → Minimum path Delay

- The sum of a set of numbers whose product is constant is minimized by choosing all the no.s to be equal.
  - The path delay is minimum when each stage bears the same effort.
  - If path has 'N' stages & each bears the same effort, that efforts must be

$$\hat{f} = g_i h_i = F^{1/N}$$

- The min. path delay (D) of a N-stage path, with path effort(F) path parasitic delay (P) is

$\because \quad D = D_F + P$

$= \Sigma f_i + P$

$= \Sigma g_i h_i + P \quad = \Sigma \hat{f} + P$

$D = N \cdot f^{1/N} + P$

N → No. of stages; $f^{1/N}$ → stage effort of each stage.

* The key result of logical effort is that the minimum delay of the path can be estimated knowing only
  - no. of stages
  - path effort
  - parasitic delay

  w/o the need to assign transistor sizes.

* Above method is better than simulation, because in simulation delay depends on sizes. And you will never know the size for min. delay. (?)

Gate Sizes

- How wide should the gates be for least delay?

→ ∴ optimal stage effor,

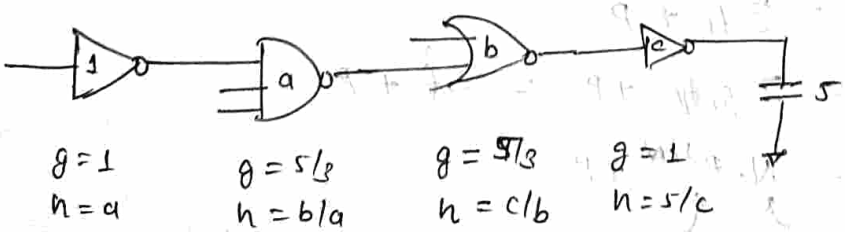
$$F^{1/N} = \hat{f} = gh = g \cdot \frac{C_{out}}{C_{in}}$$ ← for a stage

$$\Rightarrow C_{in_i} = \frac{g_i C_{out_i}}{\hat{f}}$$ ← when o/p capacitance of the path is given. effort & appropriate stage delay is also known.

- Starting with the load at the end of the path, work backwards applying the capacitance transformation to determine the size of each stage.

  — Check the calculation by verifying that the size of the initial stage matches the specification.

Example: 1 What will be the size of each gate for least delay?



$g = 1$	$g = 5/3$	$g = 5/3$	$g = 1$
$h = a$	$h = b/a$	$h = c/b$	$h = 5/c$

→ (i) Effective fanout, $H = h_1 h_2 h_3 h_4 = a \times \frac{b}{a} \times \frac{c}{b} \times \frac{5}{c} = 5$

(ii) $G = g_1 g_2 g_3 g_4 = 1 \times \frac{5}{3} \times \frac{5}{3} \times 1 = \frac{25}{9}$

(iii) $F = BGH = 1 \times 5 \times \frac{25}{9} = \frac{125}{9}$ → ($B = 1$) since only one branch

(iv) $\hat{f} = F^{1/N} = \left(\frac{125}{9}\right)^{1/4} = 1.9305$ → optimal stage effort

Therefore,

$$\hat{f} = g \cdot \frac{C_{out}}{C_{in}}$$

$$C_{in} = \frac{g \cdot C_{out}}{\hat{f}}$$

(i) for NOT gate,

$$C_{in_4} = \frac{g_4 \cdot C_{out}}{\hat{f}} = \frac{1 \times 5}{1.93} = 2.59$$

$\therefore C = 2.59$

(ii) for NOR-2

$$C_{in_3} = \frac{g_3 \times C_{out(NOT)}}{\hat{f}} = \frac{5/3 \times c}{1.93} = \frac{\frac{5}{3} \times 2.59}{1.93}$$

$= 2.236$

$\therefore b = 2.23$

(iii) for NAND-3

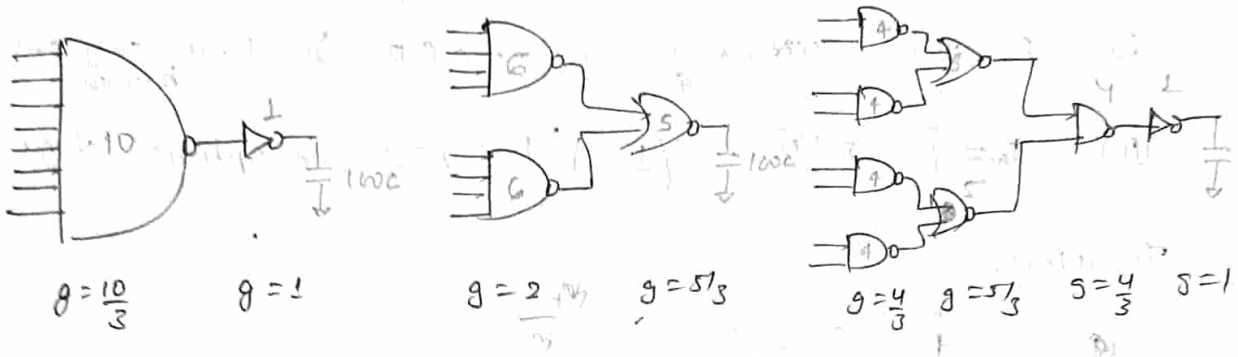
$$C_{in_2} = \frac{g_2 \times C_{out(NAND)}}{\hat{f}} = \frac{\frac{5}{3} \times b}{1.93} = \frac{\frac{5}{3} \times 2.23}{1.93}$$

$C_{in_2} = 1.9257$

$\therefore a = 1.93$

** The back hand calculation (logical effort) is just an estimation & not that accurate as we are not considering parasitic delays. & simulation will tell you the exact delay but we can't simulation everytime the complex circuits hence we do the logical effort.

Example: 2 Which of the following 8-i/p AND gate has least delay & also calculate the area for each cases.



$g = \frac{10}{3}$ $g = 1$ $g = 2$ $g = 5/3$ $g = \frac{4}{3}$ $g = 5/3$ $s = \frac{4}{3}$ $s = 1$

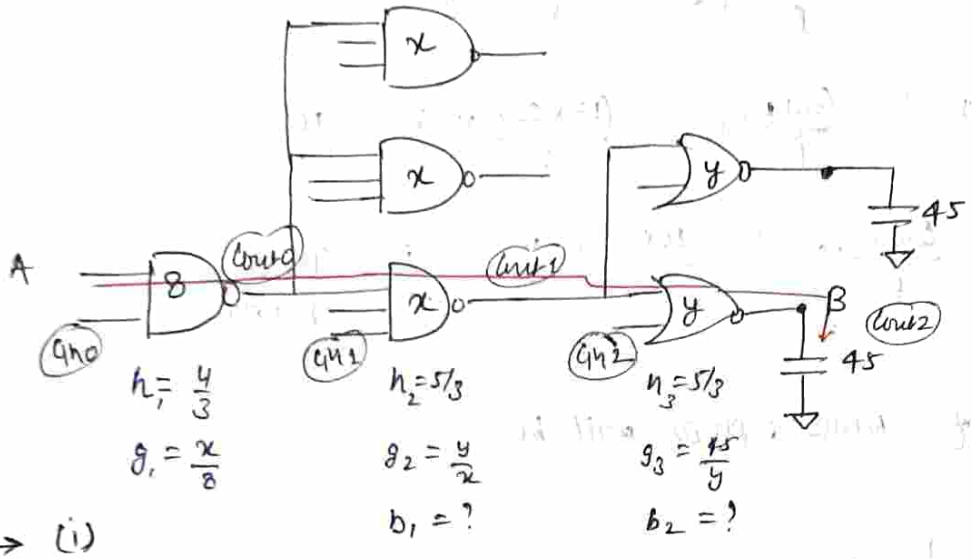
(a) (b) (c)

No. of i/p = Parasitic delay

(*)

Example : 3 (i) Calculate the least delay from A to B.

(ii) Select sizes x + y for atleast delay from A to B.



⇒ 3-stages in path A to B..

→ (i)

(i) Logical Effort : $G = g_1 g_2 g_3 = \frac{4}{3} \cdot \frac{5}{3} \cdot \frac{5}{3} = \frac{100}{27}$

(ii) Electrical Effort : $H = \frac{\text{Cout-path}}{\text{in-path}} = \frac{45}{8}$

(iii) Branching effort : $B = \Pi b_i$

$$b_i = \frac{\text{Con-path} + \text{Coff path}}{\text{Con-path}}$$

$$b_1 = \frac{x + 2x}{x} = 3 \quad , \quad b_2 = \frac{y + y}{y} = 2$$

$$\therefore \quad B = b_1 b_2 = 3 \times 2 = 6$$

(iv) Path Effort : $F = GBH = \frac{100}{27} \times 6 \times \frac{45}{8} = 125$

(v) Best stage effort : $\hat{f} = (F)^{1/N} = (F)^{1/3} = (125)^{1/3} = 5$

(vi) Parasitic delay : $P = 2 + 3 + 2 = 7$

(vii) Delay : $D = N\hat{f} + P = 3 \times 5 + 7 = 22$

$$\frac{(22)}{5} = 4.4 \ (!)$$

Delay for $\Rightarrow \frac{D}{\hat{f}} = 4.4$

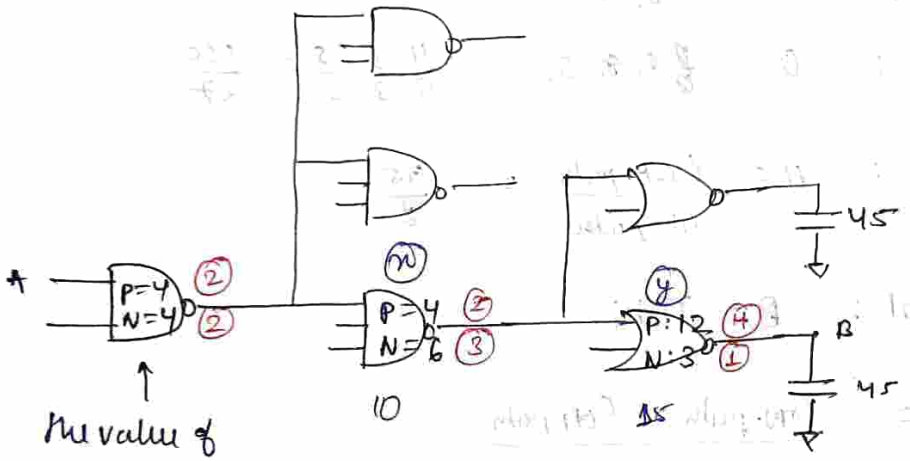
(ii) Now work backward for sizing :-

$\hat{f} = g \cdot h$

$\hat{f} = g \cdot \frac{C_{out}}{C_{in}}$

- $y = C_{in_2} = \frac{C_{out_2}}{\hat{f}} \cdot g_3 = \frac{45}{5} \times \frac{5}{3} = 15$
- $x = C_{in_1} = \frac{C_{out_1}}{\hat{f}} \cdot g_2 = \frac{(15 \times 2)}{5} \times \frac{5}{3} = 10$
- $C_{in_0} = \frac{C_{out_0}}{\hat{f}} \cdot g_1 = \frac{10 \times 3}{5} \times \frac{4}{3} = 8$ Hence proved

Therefore the size of NMOS & PMOS will be



The value of x - multiple to upsize are written here.

Conclusion → Using logical effort you have got the
- delay estimation
- area estimation

Best Number of Stages → cost delay & Area both.

- How many stages should a path use?
  - Increasing the no. of stages first delay ↓ then delay ↑.
  - Increasing the no. of stages area always ↑.

* $D = NF^{1/N} + P$

$= N F^{1/N} + N$

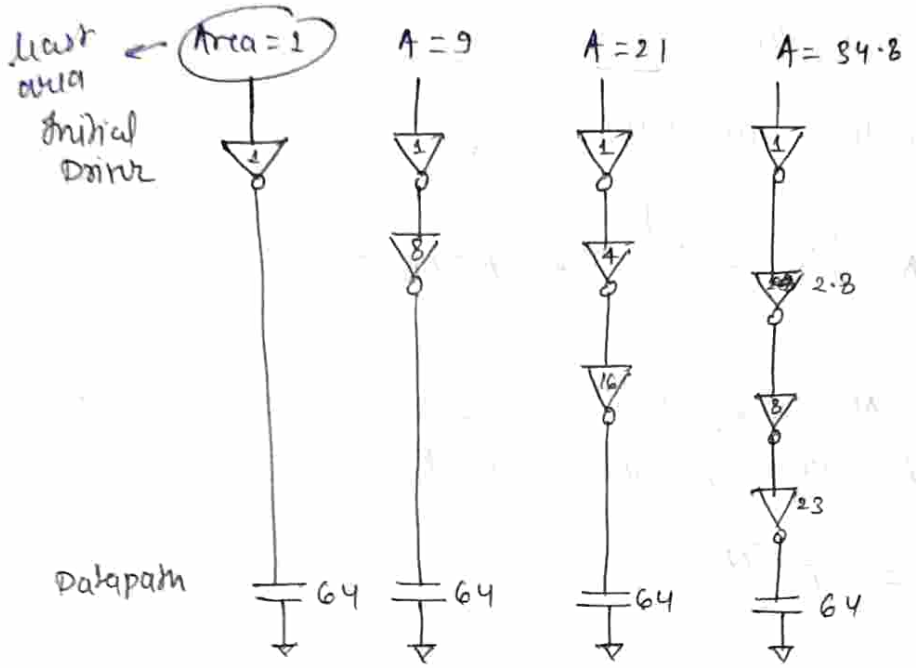
$= N(64)^{1/N} + N$ → since we are only using the NOT inverter hence each inverter will have parasitic delay = 1 only.

∴ No. of stages = to parasitic delay.

$F = GBH$

$= 1 \times 1 \times \frac{64}{1} = 64$

Eg: Drive 64-bit datapath with unit inverter



(i) $\hat{f} = (64)^{1/N}$ ✓

(ii) $D = N\hat{f} + N$ ✓
$= N(\hat{f}+1)$

N :	1	2	3	4
$\hat{f}$:	64	8	4	2.8
D :	65	18	15	15.3

Best in terms of both area & delay

fastest

because parasitic delay ↑.

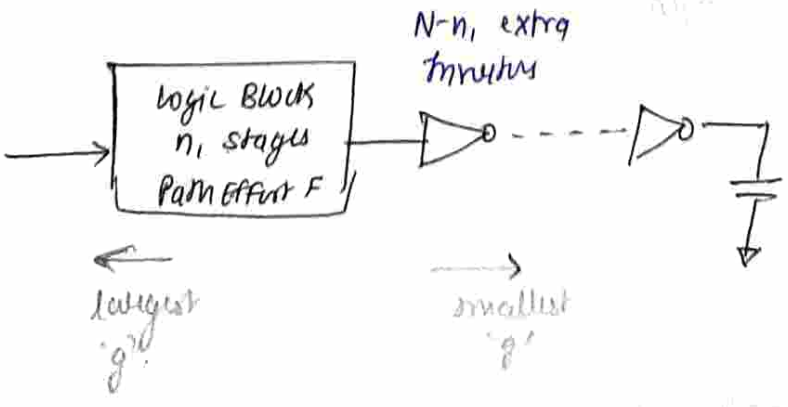
$\left(D = \frac{\text{Work}}{A \uparrow \text{(Area)}}\right)$

doubt

Note: (i) Gates with fewer i/ps are better than gates with many i/ps.

(ii) It is believed that the fastest ckt are those that use the fewest stages of logic. (which is not true always)
— off-course, the gate delay actually depends on the Electrical effort, so sometimes using fewer stage results in more delay.

(iii) In general you can add extra inverters to the end of the path for decreasing the effort w/o changing the func^n of original combinational block. (Why?)



- The extra inverters don't change the path logical effort but do add parasitic delay.
  - The delay of the new path is

$$D = N F^{1/N} + \sum_{i=1}^{n_L} p_i + (N - n_L) p_{inv}$$

- Differentiating wrt N & equating it to '0' allows us to solve for the best no. of stages, which we call $\hat{N}$.

$$\therefore \quad \rho = F^{1/N}$$

Also,

$$\frac{\partial D}{\partial N} = -F^{1/N} \ln F^{1/N} + F^{1/N} + p_{inv} = 0$$

$$\Rightarrow \quad p_{inv} + \rho (1 - \ln(\rho)) = 0$$

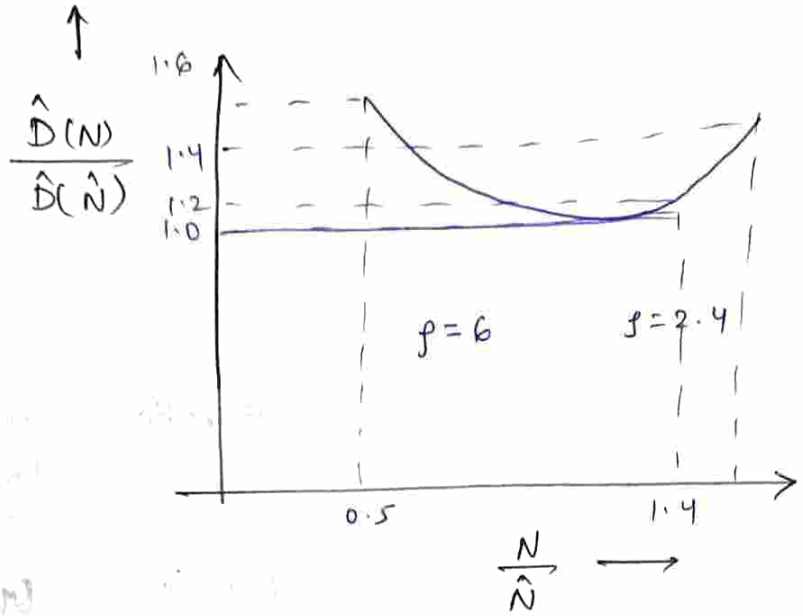
✓

it has no closed form solution

* Parasitic delay means each inverter is some-what more costly to add.
* As a result it is better to use fewer stages, or equivalently a higher stage effort than e.
* Solving numerically, when $p_{inv} = 1$, we find $\rho = 3.59$

** The path achieves best delay by using,

$$\hat{N} = \log_\rho F \text{ stages}$$



(13) (41)

Logical effort & PV

$T = 2(2n+1).T_b$

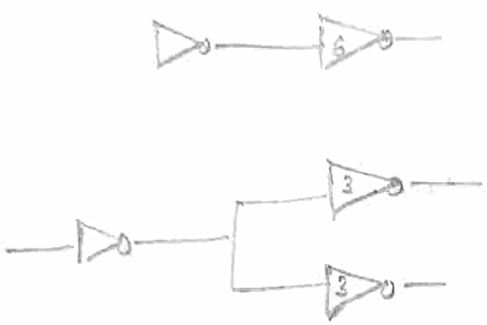
$\frac{1}{T} = f = \frac{1}{2(2n+1)\times T_b}$

$10\,mhz = \frac{1}{(2n+1)T_b}$

$T_b(2n+1) = 10^{-7}$

$T_b \cdot N = 10^{-7}$

if $T_b = 2$



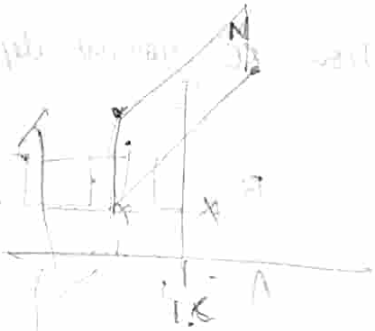
① INV - NOR4 - INV

Delay = ?

Stages = 3

② NOR4 - INV - INV

③ NAND4 - INV - INV - INV



⇒ V_T & L_{eff} are independent for Large channel device.
V_T & L_{eff} are dependent for small channel device.

⇒ TT → FF

Use inverter which changes both NMOS & PMOS.

* Uncorrelated variations

we get $I_p \neq I_n$

- for fast corners:

push all corners, voltages, temperature (V_{DD}) together.

- for calculating delay & ~~max~~ min freq.

we have calculate for SS.

- Functional corners (FFF)
  - Latches (used more in our ckt)
- Characterization corners (FF)

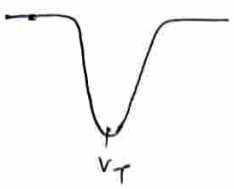
β-ratio : $\frac{I_{ON}\text{-pmos}}{I_{DN}} = \frac{I_{DP}}{I_{DN}}$

β-ratio $= \frac{\mu_p C_{ox_p}}{\mu_n C_{ox}}$

Q. What is V_T?

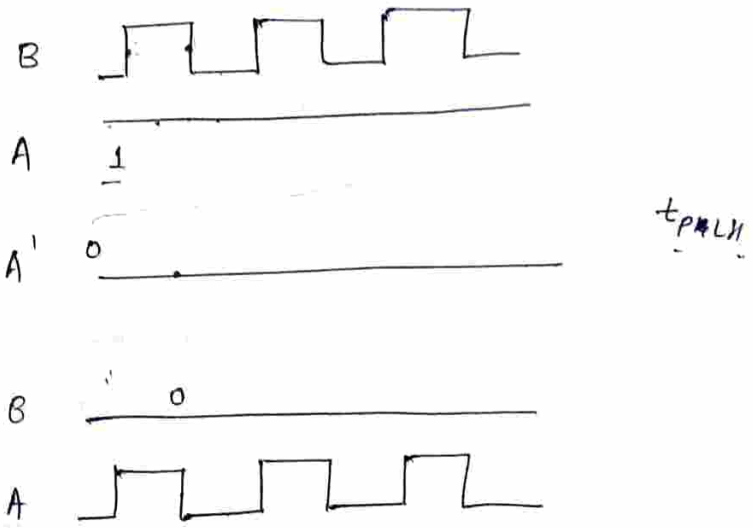
→ The value of V_{in} at which $I_D = 1\,\mu A$.

Cap. Analog



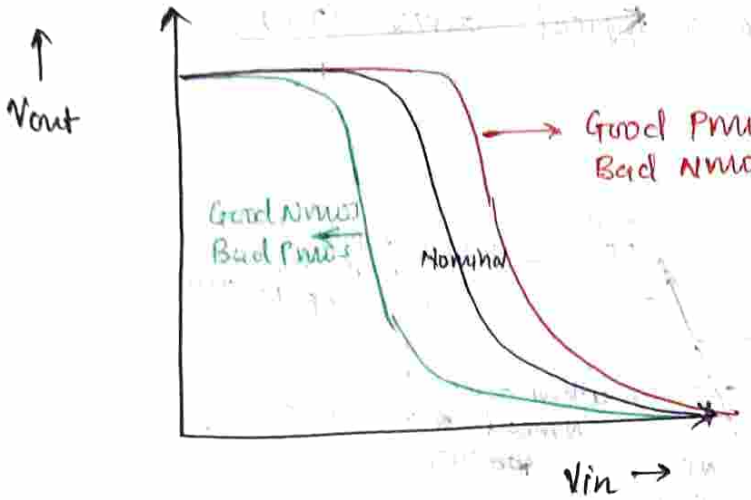
Q. How RB current depends on applied voltage?

DFM:-



3. Process, Voltage & Temperature (PVT)

Impact of process variation :



	NMOS	PMOS
A	cutoff	linear
B	sat.	linear
C	sat	sat
D	linear	sat
E	linear	cutoff

- When curve is shifting towards right then it means that PMOS is driving more current as compared to NMOS.
  i.e, PMOS is fast (Good) & NMOS is slow (Bad).
  — Since PMOS is driving more current then capacitor will be charging for more time & curve shifts Right.

Q: How do we have a good PMOS & Bad NMOS ?

— • Good PMOS → Fast PMOS → more current will be drawn (It can be due to ~~error~~ bigger size of PMOS or higher doping)

* β-ratio of model $\Rightarrow \frac{\beta_P}{\beta_n} = \frac{\mu_p C_{ox}}{\mu_n C_{ox}} = \frac{K_P}{K_n}$

* β-ratio $\Rightarrow \frac{I_{ON(PMOS)}}{I_{OFF(PMOS)}}$

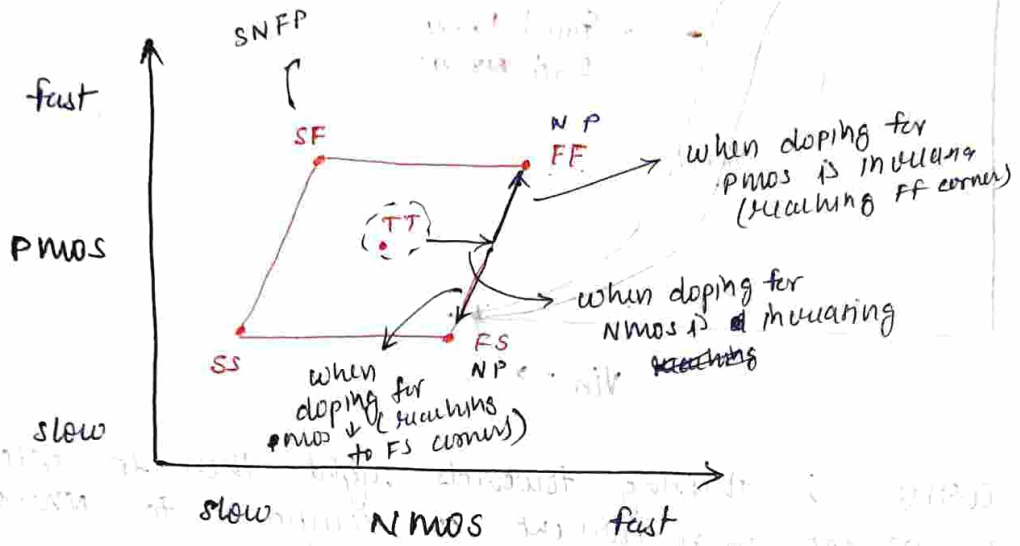
  if β-ratio > 1 $\Rightarrow$ it is ON for more time & OFF for less time.
  of MOSFET

** As we can see that sometimes PMOS & NMOS can have different driving strength depending on how ~~it~~ they have been manufactured.
— It is possible that due to manufacturing differences any of one of either PMOS & NMOS can be slow or fast. And this case called process skew.

Process Variations :

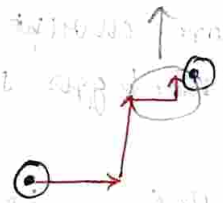
- Variation in process is due to the following things:
  (i) L_{eff} (ii) V_t (iii) t_{ox} of PMOS & NMOS

- So we targeted to make a typical device (TT).

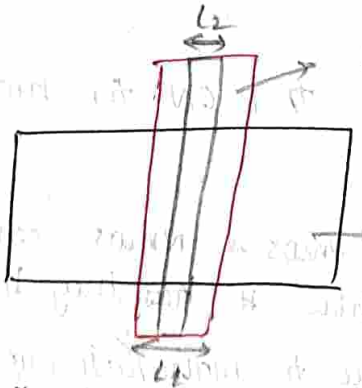


✓ We started from TT.

✱ (why?) If we say that etching of poly (when we are defining the gates) that also happens for few more milliseconds more so both NMOS & PMOS become little faster.



— when poly is etched more then length (L_{eff}) is shorter & $\uparrow I_D \propto \frac{1}{L_{eff}} \downarrow$ ⇒ more current will flow & NMOS becomes fast.



"Self Aligned Gate"

Q. What is the process of manufacturing?

→ (i) First we will have the substrate & then we will have the N-well.

substrate → SiO_2 → etching area for n-well → oxide

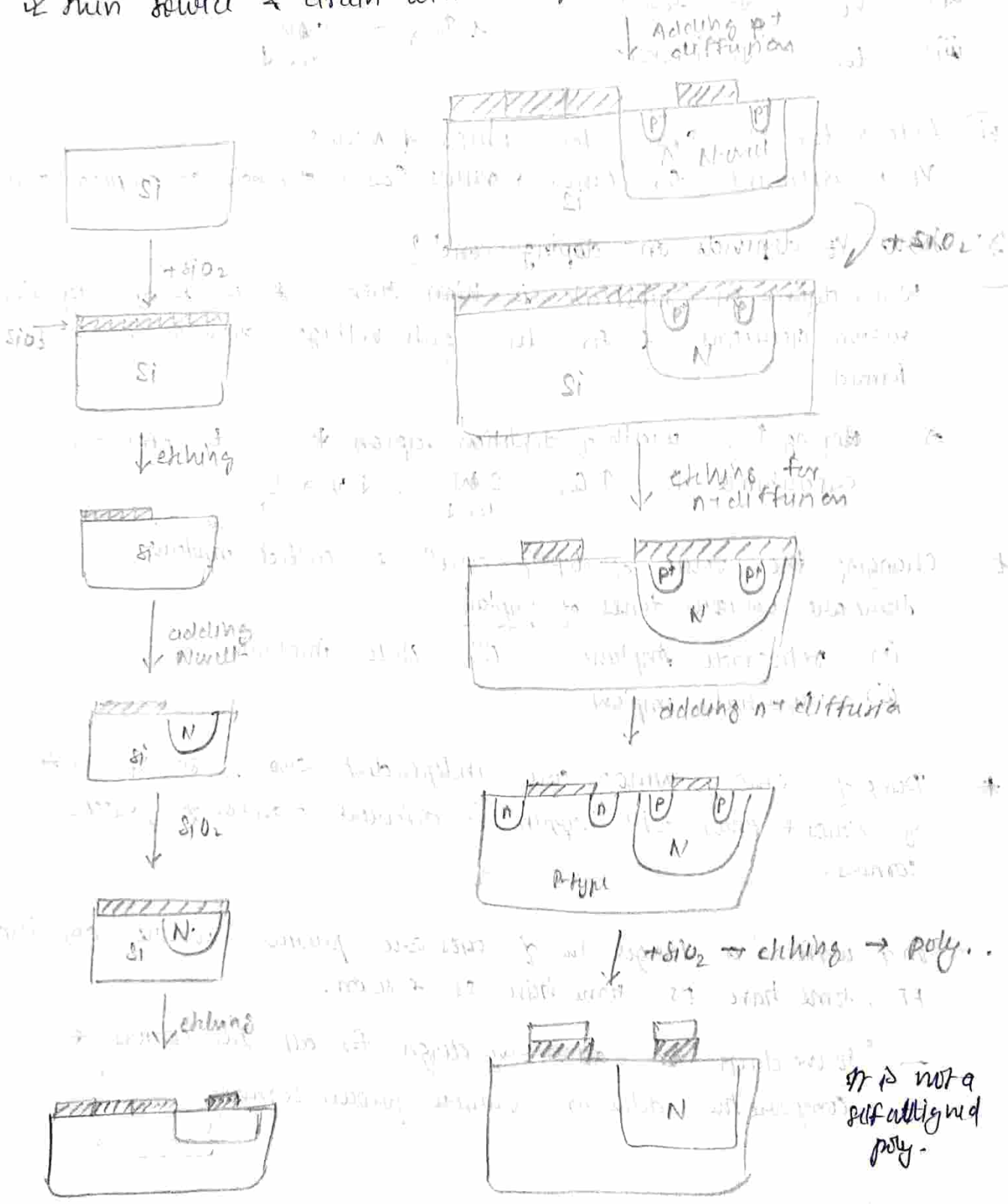
N-well → SiO_2 → etching area for poly

→ SiO_2 → poly (gate is formed after etching) → etched area for diffusion area.

→ diffusion → etched.

[Visualize all this]

* Then source & drain will always be in contact with each other.



- In the previous method of fabrication there is a possibility of poly-mismatch portion.

— But when poly is added before the n^+ & p^+ diffusion then poly is self alligned & no mismatch is observed

For a fast device:

(i) L_{eff} : ↓ (short) $\quad I_D = \mu_n C_{ox} \frac{W}{L}\left[(V_{GS}-V_t)V_{DS} - \frac{V_{DS}^2}{2}\right]$

(ii) V_t : ↓ (low)

(iii) t_{ox} : ↓ (thin) $\quad \& \uparrow C_{ox} = \frac{\varepsilon A W}{t_{ox}\downarrow}$

(*) L_{eff} & t_{ox} is same for PMOS & NMOS
V_t is different for PMOS & NMOS (as it depends on doping conc^n)

Q: How V_t depends on doping conc^n?

— When doping in substrate is high then width of depletion region decreases & for less gate voltage depletion region is formed

⇒ doping ↑, width of depletion region ↓, t_{ox} (eff) for capacitance ↓, $\uparrow C_{ox} = \frac{\varepsilon A W}{t_{ox}\downarrow}$, $\downarrow V \propto \frac{1}{C\uparrow}$

(*) Changing the value of doping conc^n is called implant:
There are various types of implant.

(i) retrograde implant (iii) halo implants
(ii) super halo implant

(C V_t doesn't only depends on amount of doping, it depends on depth of doping & many other factors)

* Doing of PMOS & NMOS are independent step, so V_t shift of NMOS & PMOS will happen in different direction of process corners.

- In a wafer a larger no. of dies are present where some have FF, some have FS, some have SS & so on.

— * So we check our design for all the corners & compare the delay for various process corners.

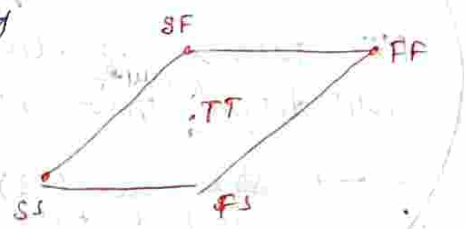
- In actual the dies (or ICs) will lie anywhere in b/w these process corners.

Q. On what other factors the delay will change?

→ (i) process corners (due to V_T, t_{ox}, L_{eff})
(ii) Temperature
(iii) Voltage of operation (V_{dd})
(iv) Input slope
(v) width of the mosfets (NMOS & PMOS)

* Manufactured IC will not have as bad as you designed it for.
& it will not be as good as you designed it for.

- As a designer we can't check at every point inside the process corners
∴ we only check at the corners & then we generate an analysis.



Environment Variations:

- Mobility (μ_n) variation wrt temperature.

$\mu_n \propto \frac{T^{3/2}}{N_D}$ & $\mu_n \propto \frac{1}{T^{3/2}}$

fast (at lower temp.) due to ionized impurity scattering

slow (at higher temp.) due to phonon scattering

- (now?)
At low temp & high voltage we get fast device
At high temp & low Vdd voltage we get slow device

- V_{DD} & Temp. also vary wrt time & space.

Corner	Voltage	Temp
(F)	1.98	0C
($)	1.8	70C
(S)	1.62	125C

Ans: At high temp. $\mu_n \downarrow$ & due to low voltage less carriers are generated. Hence slow device.

— At low temp. $\mu_n \uparrow$ & due to high voltage more carriers are generated. Hence fast device.

Process corners:

⊙ Process corners describe worst case variations
— If design works in all corners, it will probably work for any variation.

Que. Describe corner with four letters (T, F, S):

(a) ~~Process Slow Speed~~ → that can be described by more 4 variations

PVT means process, voltage & Temperature for which you will design & verify your ckt.

→ ~~Slow - Slow (SS)~~ Fast - Fast (FF), low temp & high voltage

- Nmos speed
- Pmos speed
- Voltage
- Temperature

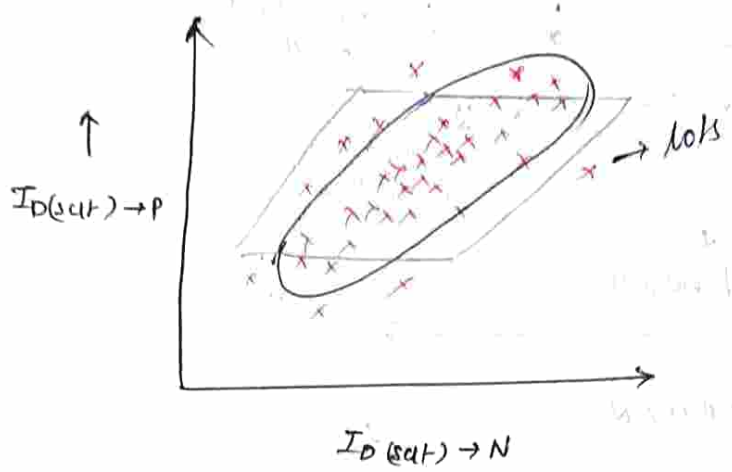
How?

- Contamination delay will be (?) worst ⇒ max in
  — FF, low temp & high voltage
- Propagation delay will be worst in
  — SS, high temp & low voltage

- Some critical simulation corner includes: (for worst case)

Purpose	Nmos	Pmos	V_{DD}	Temp
Cycle time	S	S	S	S
Power	F	F	F	F
Subthreshold leakage	F	F	F	S

Model is a fitting exercise:



① A large no of wafers a manufactured simultaneously & each wafer is called a lot.

② When we characterize a particular lot that what it is FF, FS or SF then it is called lot variation

③ And suppose we pick 100 x-sistors inside of a lot, and when we characterize them then it is called device variation.

- In ~~reality~~ reality we don't get rombus we get ellipse. In ellipse you define your process corners.
- It is possible that in the dots outside the ~~ellipse~~ ellipse you will get some extra yield loss.
- These small dots are the different process corners at which our design or device may operate.

Conclusion:- (i) Our CKT (or device) can be fabricated at any of this (x) ie, lots. (called lot variation)

(ii) When we take a single ~~wafer~~ wafer which is ~~fabricated~~ fabricated at a particular lot then we do the "monte carlo simulation" to validate a design with device variation like SS, SF, FF, FS.

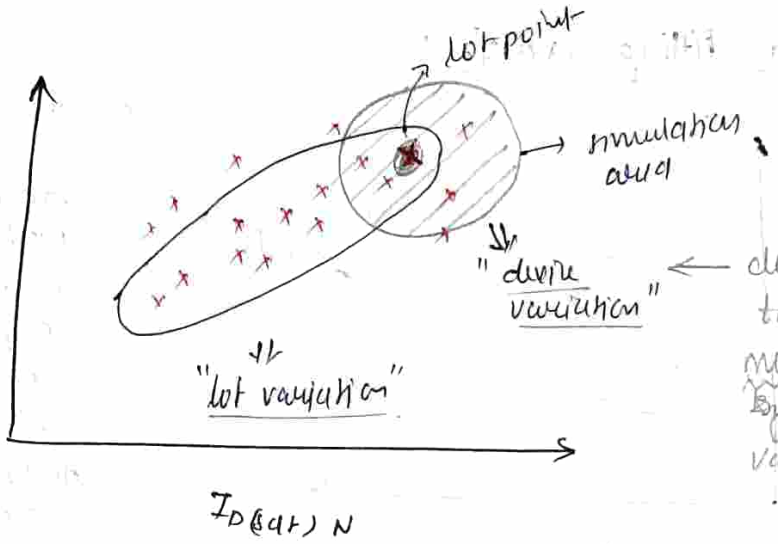
Q. What is Monte carlo Analysis?

— It is the analysis in which you will vary different parameters which are known to exhibit variation.

— parameters that exhibit variations are (common parameters):

- Doping (V_t)
- L_{eff}
- t_{ox}

] this variation can be pulled by putting variation corners SS, SF, FS, FF.



← done in functional test (by doing monte-carlo simulation) by considering device variations.

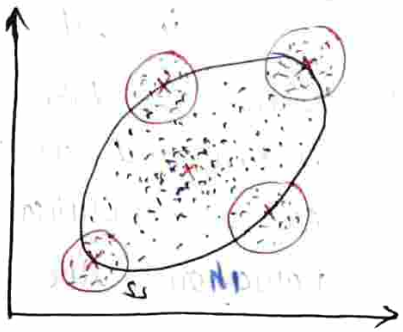
- When you do the monte-carlo simulation around the selected lot then you get the big shaded area around the lot point
  - It is the sample of transistors that could be manufactured on the particular lot.

④ • When we measure current in all the transistors inside the lot, and we list down all their current values & say that my current distribution in this lot & this distribution is fed into simulator & we run monte-carlo simulation.

- X_1 : PMOS → $V_{tss} + \Delta V_{t1}$
  X_2 : PMOS → $V_{tss} + \Delta V_{t2}$
  X_3 : NMOS → $V_{tss} + \Delta V_{t3}$
  X_4 : NMOS → $V_{tss} + \Delta V_{t4}$

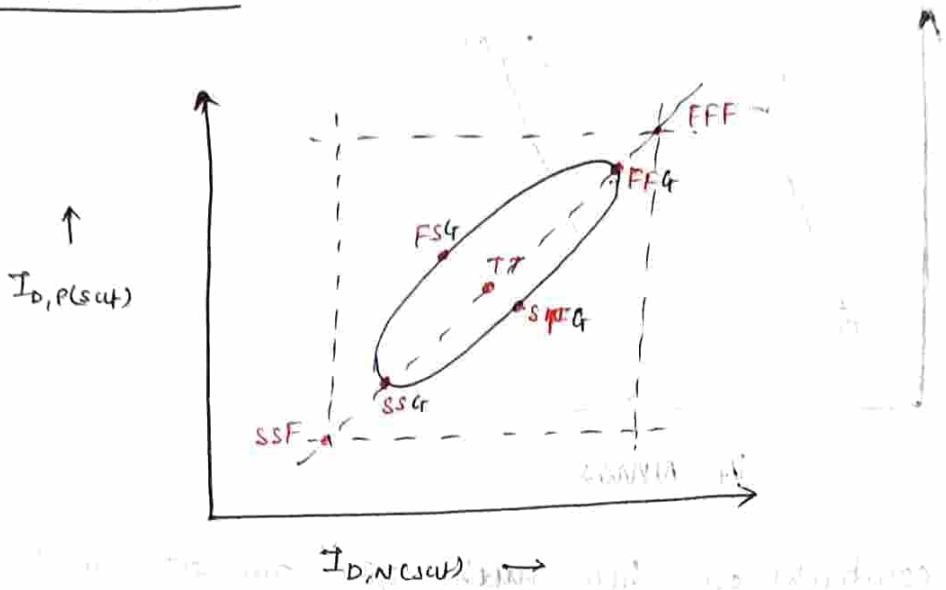
After selecting V_{tss}, we added ΔV_t this is called monte-carlo simulation

this is for monte-carlo variation



- characterization → You have a design (on silicon or layout data) & you measure current, leakage, delay is called characterization
- Verification (only on CAD) → You put this parametric variation like SS, SF, FS, FF & you put monte-carlo variation & then you verify your device will work fine or not
- Validation → studying the variations in silicon & you validate whether it is meeting your lot specification or not.

Functional models :



- Some foundaries provide extra models called SSF & FFF for verifying the device.

⇒ .mc command for mote carlo simulation in eldo.

Process Compensation : (Area of research)

- Suppose a device is designed fast but here leakage is also very high & delays are much less

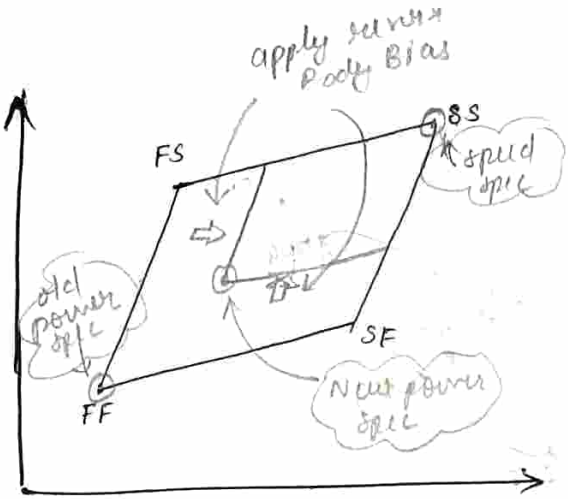
Eg. You designed something to operate at 2 GHz, but after manufacturing it is operating at 3 GHz. But we can't operate it at 3 GHz because it can have resonance issue, inductive effect and so on.

- So how to operate the device at 2 GHz which is fast & leakage is also very high?

→ We can try to reduce the leakage. For that we can do multiple things :

① Apply Body bias (reverse body bias), so device more towards slow & V_t starts to increase (& leakage ↓) (doubt)
"Adaptive Body Bias"

② Lower the Vdd from 1.2V to 1.15V,
At 1.2V V_{DD}, operating freq is 3 GHz
At 1.15V V_{DD}, operating power decreases
"Adaptive Voltage scaling"



- There are constraint on how much you can do reverse body bias, because tunneling can occur.

* In FDSOI (fully depleted silicon on insulator), there is no junction in source & drain region there is a buried oxide
  — Therefore there is no concept of junction leakage.

Power Dissipation

Q. Where does the power go in CMOS?

- (i) Dynamic power consumption
  - during charging & discharging of capacitor
- (ii) Short ckt currents
  - Short ckt path b/w supply rails during switching
  - when short ckt path exists b/w Vdd & ground b/w the supply rails during switching operations
- (iii) Leakage
  - Leaking diode & resistors

⇨ Power is drawn from a voltage source attached to the V_{DD} pins of a chip.

(i) Instantaneous Power : $P(t) = I(t)V(t)$

- Designer must be interested in instantaneous power as it gives us the estimate of what kind of current is flowing & with that we can design our power grid. The current should not be that high that my power grid collapses

(ii) Energy : $E = \int_0^T P(t)\,dt$

- As a designer we must know the energy consumed for one calculation, so we can calculate the total energy required by a calculator to do 1 billion calculation and we place the battery according to that.

(iii) Average power : $P_{avg} = \frac{E}{T} = \frac{1}{T}\int_0^T P(t)\,dt$

- When you quote that your bulb consumes 10watt then it is avg. power so this data is available for the user to understand & so the designer must define it.

* As a designer you are interested in all three of them.

Q: On what factor / parameter I (current) depends on? for cmos

→ (i) Resistance of wire

(ii) V_{DD}

(iii) Resistance of channel or $\frac{W}{L}$ or μ_n or V_T

(iv) slew → next will tell for how long short ckt current will flow

(v) load

(vi) charge = $\int i(t)dt$

(vii) Frequency → No. of times the cmos inverter will toggle decides the current through inverter at a particular instant.

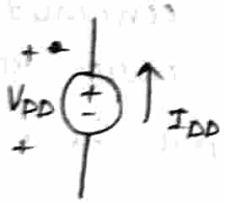
$I = \frac{dQ}{dt}$ → charge consumed per unit time

* Here we are talking about no. of times the toggling is happening, so freq. has very imp role to play & associated with it is called activity factor.

* Instantaneous power depends on freq. of operation, if you want to limit your instantaneous power you have to reduce the freq. of operation

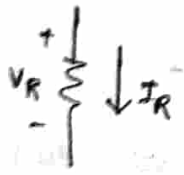
Power in ckt elements

(i)



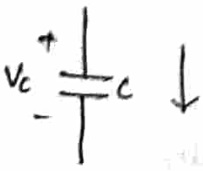
$P_{VDD}(t) = I_{DD}(t) \cdot V_{DD}$

(ii)



$P_R(t) = \frac{V_R^2(t)}{R} = I_R^2(t) \cdot R$

(iii)



$E_c = \int_0^\infty I(t)V(t)\,dt = \int_0^\infty C\frac{dV}{dt} \cdot V(t)\,dt$

$E_c = C\int_0^{V_c} V(t)\,dV = \frac{1}{2}CV^2$ — (1)

Sources of Power Dissipation

- $P_{total} = P_{dynamic} + P_{static}$

(i) Dynamic power : $P_{dynamic} = P_{switching} + P_{short\ ckt}$

- switching load capacitance
- short ckt current (when device gets on & Vdd connects to GND).

(ii) Static power : $P_{static} = (I_{sub} + I_{gate} + I_{junc^n}) V_{DD}$

I_{sub} → submicrometer leakage; I_{gate} → Gate leakage; I_{junc^n} → Junction leakage

While going through the dynamic & static power, we will consider a chip having 1 Billion transistors:

- 50M logic transistors
  - Avg width = 12λ
  - Activity factor = 0.1

  } In DVD, course we will talk about this transistors only.

- 950M memory transistors
  - Avg width = 4λ (small)
  - Activity factor = 0.02

  W_p or W_n = 4λ; λ → minimum width of transistor

- 1.0V, 65nm process
- C = 1fF/μm (gate) + 0.8fF/μm (diffusion)

— Above all are assumptions

1. Energy delivered by power supply:

$$E_{Vdd} = \int_0^\infty i(t) \cdot V_{supply}(t)\, dt = \int_0^{V_0 = V_{DD}} C_L \frac{dV_0}{dt} \cdot V_{DD}\, dt = C_L V_{DD}^2$$

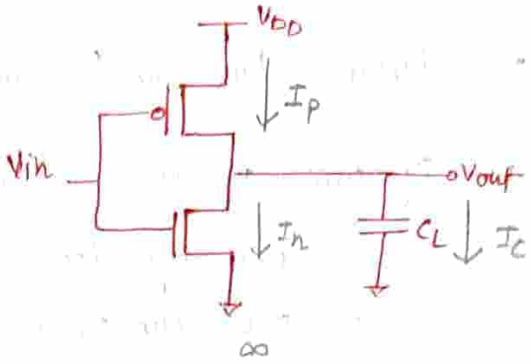
2. Energy consumed by capacitor:

$$E_C = \int_0^\infty V_0(t) \cdot i(t)\, dt = \int_0^\infty V_0 \cdot \frac{C_L dV_0}{dt} \cdot dt = \int_0^{V_{DD}} C_L V_0\, dV_0$$

$$= \frac{1}{2} C_L V_{DD}^2$$

1. Dynamic Power : ⟶ Power because of charging & discharging

(i) When the gate o/p rises:



- Energy stored in capacitor is

$$E_c = \frac{1}{2} C_L V_{DD}^2 \quad (\text{from } ①)$$

- Energy drawn from the supply is

$$E_{rdd} = \int_0^\infty I_c(t) V_{DD}\, dt = \int_0^\infty C_L \frac{dV}{dt} \cdot V_{DD}\, dt = \int_0^\infty C_L V_{DD}\, dV$$

(I_c = $C_L \frac{dV}{dt}$)

$$= C_L V_{DD} \int_0^{V_{DD}} dV = C_L V_{DD}^2$$

- Half the energy from V_{DD} is dissipated in the pmos transistor as heat, and other half is stored in capacitor.

(*) ① Energy dissipated is func^n of Resistance (R_{eq}) of NMOS & PMOS hence it is not dependent on design parameter such as width of itself.

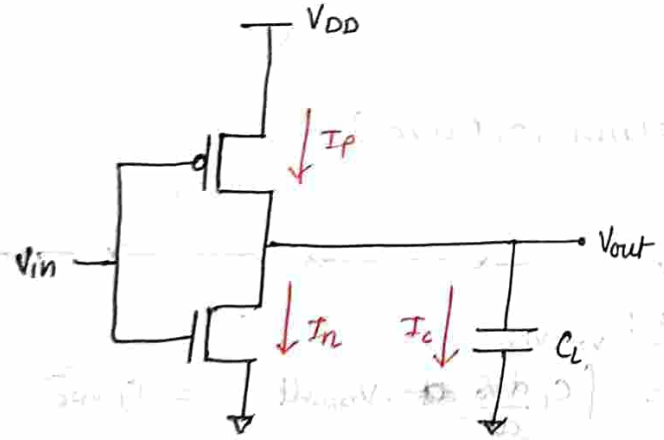
② But depends on W because of its own parasitic capacitance as load.

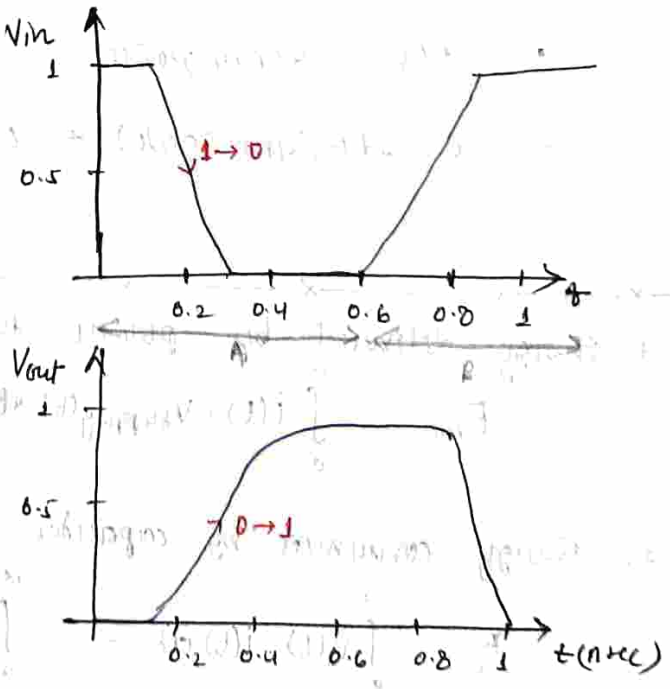
(ii) When gate o/p falls - dissipated as heat in NMOS

- Energy $\frac{1}{2} C_L V_{DD}^2$ gets dissipated as heat in NMOS

Energy in one capacitor $E_c = \int_0^{V_{DD}} v_o \cdot i(t)\, dt = \int_0^{V_{DD}} v_o \cdot C_L\, dv_o = \frac{1}{2} C_L V_{DD}^2$

Switching waveforms :





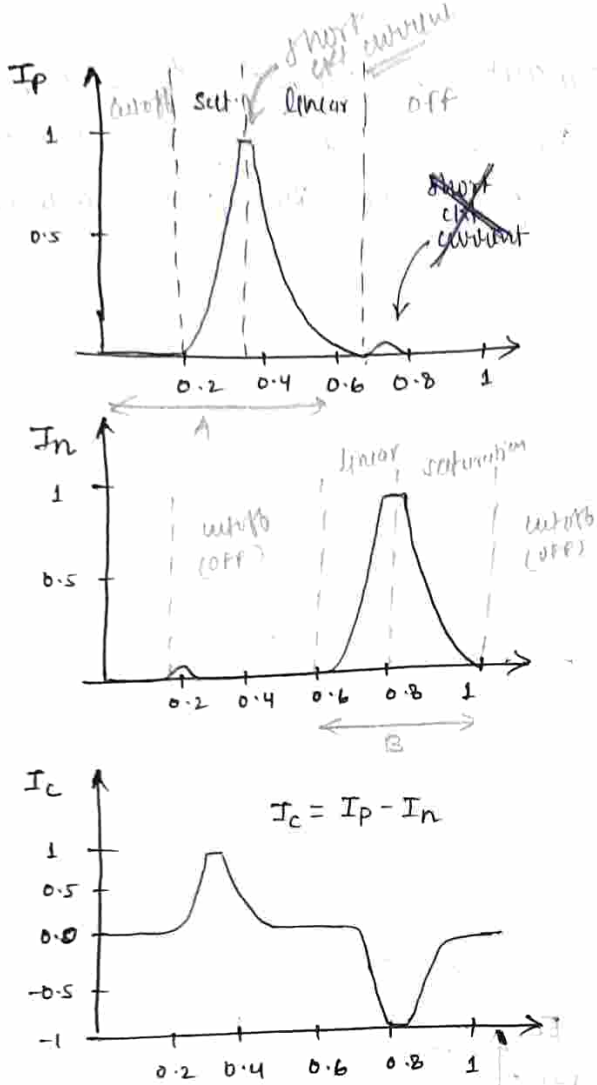
- When the i/p switches from 1 to 0, the PMOS transistor turns ON & charges the load to V_{DD}.

— Energy stored in capacitor

$$E_C = \frac{1}{2} C_L V_{DD}^2$$

$$V_{out} = \frac{1}{C} \int I_c\, dt$$

$$\because I_c(t) = C_L \frac{dV_{out}}{dt}$$



- But the power supply energy from ~~[illegible]~~ $= C_L V_{DD}^2$
- Power dissipated in the PMOS transistor ~~[illegible]~~ is $= \frac{1}{2} C_L V_{DD}^2$

** The power dissipated depends only on the load capacitance, is it true? not on the size of the transistor or the speed at which gate switches.

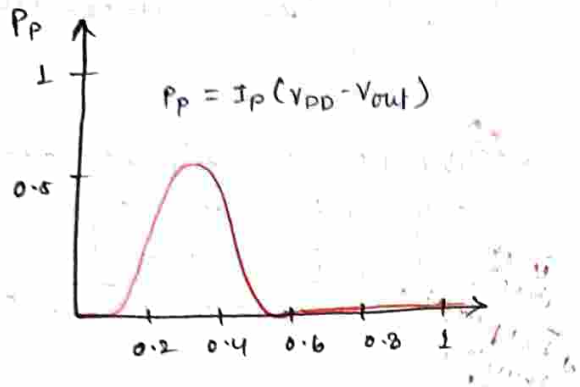
do the power consumption & clock?

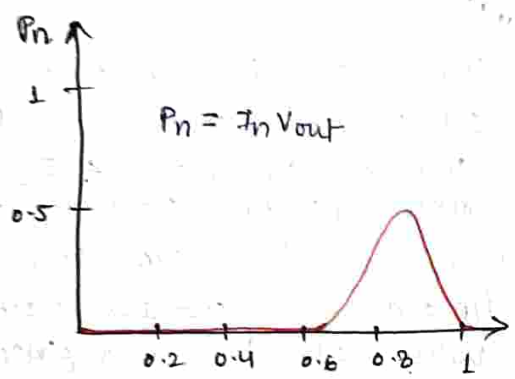
- When i/p is transiting from 1 to 0 then current is flowing through PMOS only & $I_n = 0A$, since capacitor is charging & I_p ↑.
- But when transition stops & i/p = '0' then current is at peak value. capacitor is charging completely.
- And when i/p = '0' for sometime & cap. is completely charging. then current will flow through PMOS but value of current I_p starts decreasing. & after complete charging $I_p = 0A$.
- Now when i/p transition from 0 to 1, during the transition, NMOS is ON & the current through NMOS increases as it will be in linear region.
- After some time NMOS enters into the saturation region & current starts decreasing & when cap discharges completely then $I_n = 0A$

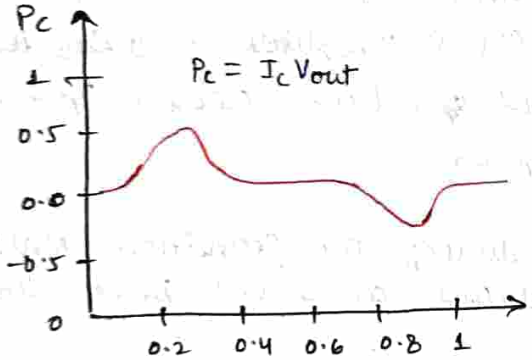
* A short ckt current will be flowing when both NMOS & PMOS are off & no current is supposed to flow.

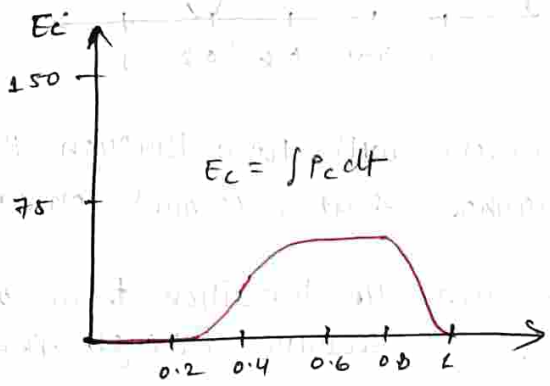
— when Vin goes from '0' to '1', the PMOS starts to turn off, however, there is a small blip of current while the partially ON PMOS fights against the NMOS. This is called Short-ckt current.

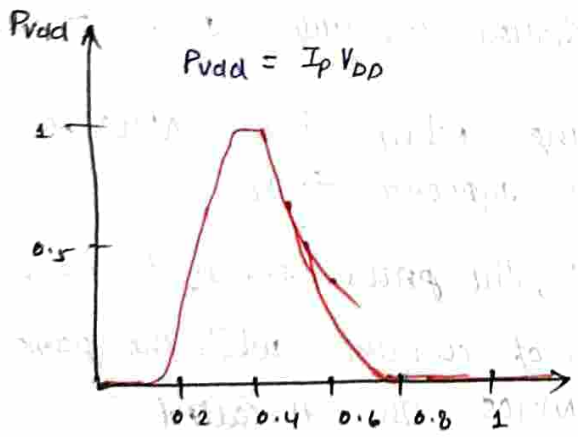
* The energy stored on the capacitor is pumped to GND. The short-ckt current consumes an almost imperceptibly small 2.7 fJ of additional energy from V_{DD} during the transition.

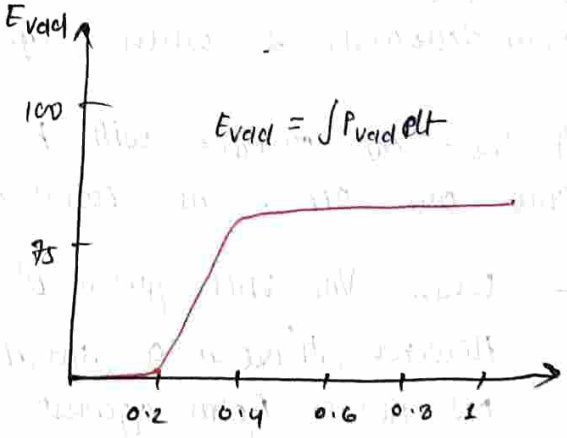












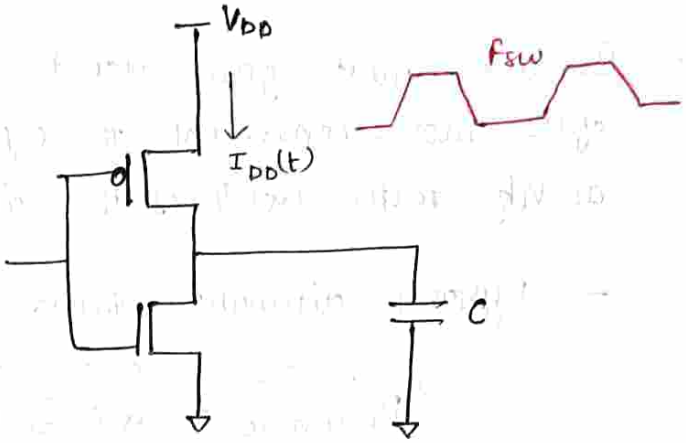
Switching Power :

$$P_{switching} = \frac{1}{T}\int_0^T I_{DD}(t)\cdot V_{DD}\,dt$$

$$= \frac{V_{DD}}{T}\int_0^T I_{DD}(t)\,dt$$

$$= \frac{V_{DD}}{T}\left[T f_{sw} C V_{DD}\right]$$

$$= C V_{DD}^2 f_{sw}$$



(a) $P_{switching} = \frac{E}{T} = \frac{T f_{sw} C V_{DD}^2}{T} = C V_{DD}^2 f_{sw}$

** when gate switches at some average freq. f_{sw}. Over time period T, then load will be charged & discharged Tf_{sw} times.

* Since, Energy drawn from the supply

$$E_{vdd} = C_L V_{DD}^2$$

if gate switches at freq f_{sw} over interval T.

$$\therefore E_{vdd} = T f_{sw} C_L V_{DD}^2$$

$\therefore$ switching power will be

dynamic power $\leftarrow$ $P_{switching} = \frac{E_{vdd}}{T} = C_L V_{DD}^2 f_{sw}$

f_{sw} → no. of times the ckt switches or toggles in a time period T

** $\frac{1}{2} C V_{DD}^2$ when ckt rises, $\frac{1}{2} C V_{DD}^2$ when ckt falls

$\Rightarrow$ [per cycle $C V_{DD}^2$] × no. of times the ckt will toggle

$\Rightarrow$ $C_L V_{DD}^2 \times f_{sw}$

Activity Factor :

- Because most gates do not switch every clock cycle, it is often more convenient to express switching freq f_{sw} as an activity factor (α) times the clock freq. f.

— Dynamic dissipation power may be written in :

$$P_{switching} = \alpha C V_{DD}^2 f$$ ← clock freq.

where, $f_{sw} = \alpha f$

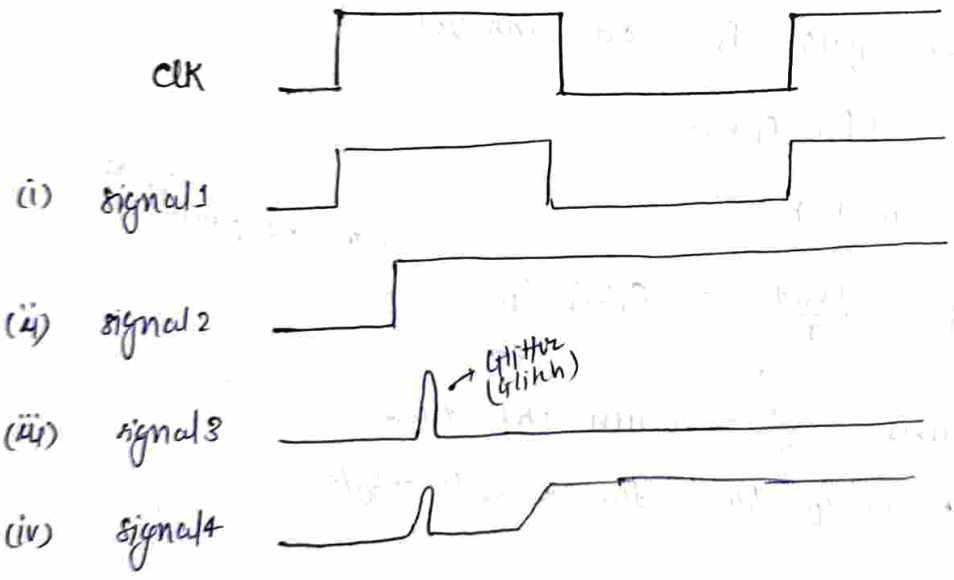
$\alpha \rightarrow$ activity factor → No. of times any node in the ckt is switching per clk cycle.

(i) * If signal is a clock, $\alpha = 1$.

(ii) If the signal switches one per cycle of clock, $\alpha = 1/2$

(iii) If the signal switches twice per cycle of clock, $\alpha = 2/2 = 1$

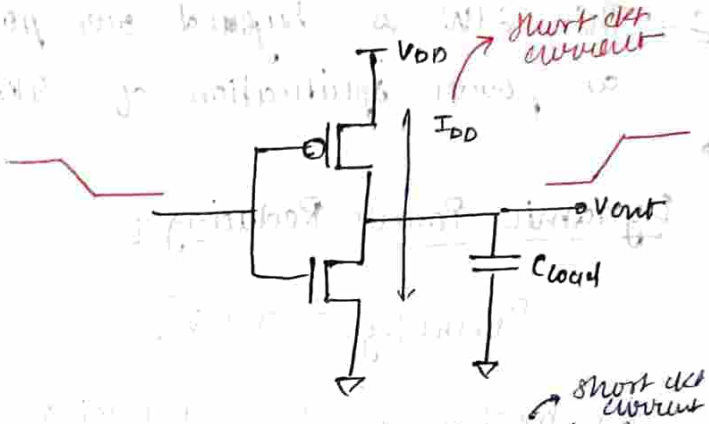
(iv) If the signal switches thrice per cycle of clock, $\alpha = \frac{3}{2}$



* for 1 glitch, $\alpha = 3/2$
2 glitch, $\alpha = 5/2$
} in signal 4

* We don't want glitches in our system because
(a) It is a signal integrity issue
(b)* Power issue [more power dissipation]

Short circuit current

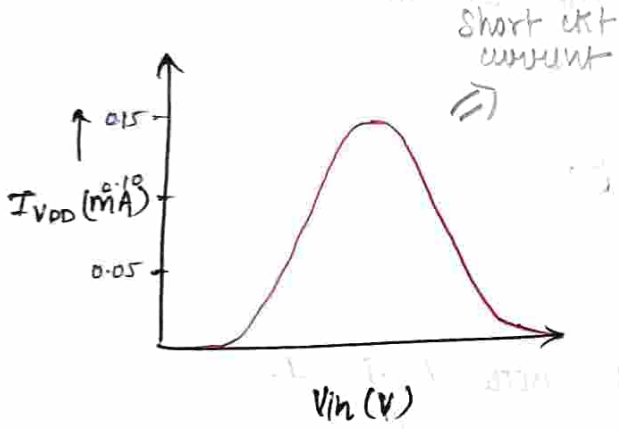


- When xistor switch, both NMOS & PMOS networks may be momentarily ON at once.
  — It leads to a blip of "short ckt" current

Short ckt current (very less)

* because some current may flow from V_{DD} to GND + some current may flow from V_{DD} to C_{load} + C_{load} to GND

- * < 10% of dynamic power if rise/fall time of i/p & o/p are comparable. [rise/fall time ①, short ckt current ②]
- We generally ignore this component while doing the calculation but consider it while doing the simulation.



Dynamic power for 1B xistor chip :

how much width I consider with λ, λ = 25nm here

(i) $C_{logic} = (50 \times 10^6)(12\lambda)\left(\frac{0.025\mu m}{\lambda}\right)\left(\frac{1.8fF}{\mu m}\right) = 27nF$

$= 50 \times 10^6 \times 12 \times 0.025 \times 1.8\,fF$

(ii) $C_{memory} = (950 \times 10^6)(4\lambda)\left(\frac{0.025\mu m}{\lambda}\right)\left(\frac{1.8fF}{\mu m}\right) = 171nF$

(iii) $P_{dynamic} = (0.1\,C_{logic} + 0.02\,C_{memory})(1.0)^2(1GHz) = 6.1W$

↓ 'α' for logic; ↓ α for memory (Activity factor)

* while taking the value of 'α' we have considered that this kind of glitches can be there.

Q. This 6.1W is beyond our power budget we can only operate at power specification of 4W. So what can be done?

Dynamic Power Reducing:

$$P_{switching} = \alpha C V_{DD}^2 f$$

- for minimizing the switching power:
  - (i) Decrease Activity factor per ckt (how to do that?)
    - by gating the device, I will put NAND at some places & enables only the part of the ckt & not other parts of the ckt, so if I reduce the activity factor of the ckt, it reduces power.
  - (ii) Reduce the capacitances, at least the capacitances that toggle for that we can use smaller devices.
  - (iii) operate at lower voltage
  - (iv) operate at lower frequency.

Activity Factor Optimization

- Let P_i be the probability that node i is 1.

  $\therefore \quad \overline{P_i} = 1 - P_i$

  And,

  $$\alpha = P_i * \overline{P_i}$$

- Completely random data has $P = \frac{1}{2} = 0.5$ &

  $\therefore \quad \alpha = P_i * \overline{P_i}$

  $= \frac{1}{2} \times \frac{1}{2} = \frac{1}{4} = 0.25$

- Data is often not completely random
  - Eg. Upper bits of 64-bits words representing bank account balance are usually 0.

- Data propagating through ANDs & ORs has lower activity factor
  - it depends on design, but typically, $\alpha \approx 0.1$.

Q. Why do AND & OR gate have lower activity factor?

* Switching probability:

- what is prob of o/p of AND2 being 1?
  - o/p of AND2 is 1 when both A & B is 1.

$$\therefore P = P_A \cdot P_B = \frac{1}{4}$$

(or)

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

$\therefore P = \frac{1}{4}$

- Similarly for NOR2, it is o/p is 1 when any of the i/p is 1 (or) 1 - when both the i/p is 0.

$$\therefore P = 1 - \bar{P}_A \bar{P}_B = 1 - \frac{1}{4} = \frac{3}{4}$$

if $P_A = P_B = P_C = 1/2$

Gate	P_Y	
AND 2	$P_A P_B$	1/4
AND 3	$P_A P_B P_C$	1/8
OR2	$1 - \bar{P}_A \bar{P}_B$	3/4
NAND2	$1 - P_A P_B$	3/4
NOR2	$\bar{P}_A \bar{P}_B$	1/4
XOR2	$P_A \bar{P}_B + \bar{P}_A P_B$	2/4 = 1/2

$$\alpha_{NOR2} = P \times \bar{P} = \frac{3}{4} \times \frac{1}{4} = \frac{3}{16} \text{ (max)}$$

$$\alpha_{AND2} = P \times \bar{P} = \frac{1}{4} \times \frac{3}{4} = \frac{3}{16} \text{ (max)}$$

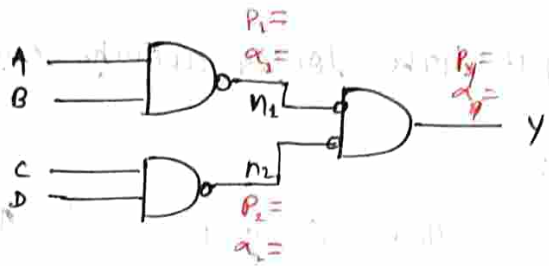
$$\alpha_{XOR2} = P \times \bar{P} = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

$$\alpha_{NAND2} = \frac{1}{4} \times \frac{3}{4} = \frac{3}{16}$$

$$\alpha_{NOR2} = \frac{3}{4} \times \frac{1}{4} = \frac{3}{16}$$

Example: A 4-i/p AND is build out of two levels of gates. Estimate the activity factor at each node if the i/p have $P = 0.5$.

$P = 0.5$ means prob. of A being 1 is $\frac{1}{2}$, prob of B being 1 is $\frac{1}{2}$.



- P_1 = Prob. of o/p of NAND gate being 1

$$P_1 = \frac{3}{4}, \quad \bar{P_1} = \frac{1}{4}$$

$$\alpha_1 = \frac{3}{4} \cdot \frac{1}{4} = \frac{3}{16}$$

A B	$\overline{AB}$
0 0	1
0 1	1
1 0	1
1 1	0

- P_2 → prob. of o/p of NAND gate being 1

$$P_2 = \frac{3}{4}, \quad \bar{P_2} = \frac{1}{4}$$

$$\alpha_2 = \frac{3}{16}$$

- P_y → prob. of o/p of Gate being 1

$$P_y = \frac{1}{16}, \quad \bar{P_y} = \frac{15}{16}$$

$$\alpha_y = \frac{15}{256}$$

R S	$Y = \bar{R} \cdot \bar{S} = \overline{R+S}$
0 0	1
0 1	0
1 0	0
1 1	0

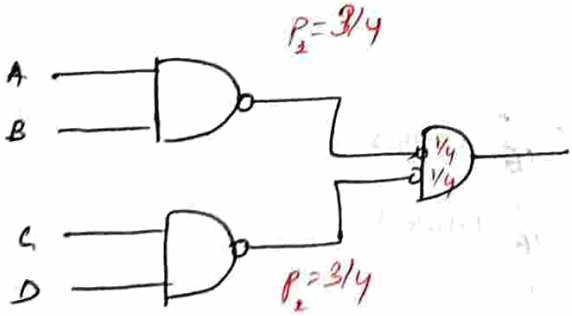
$Y = \overline{AB} \cdot \overline{CD}$

$= AB + CD$

$Y = ABCD$

$Y = 1$ for $ABCD = 1$

OR



$P_y = \frac{1}{4} \cdot \frac{1}{4}$ (for AND gate)

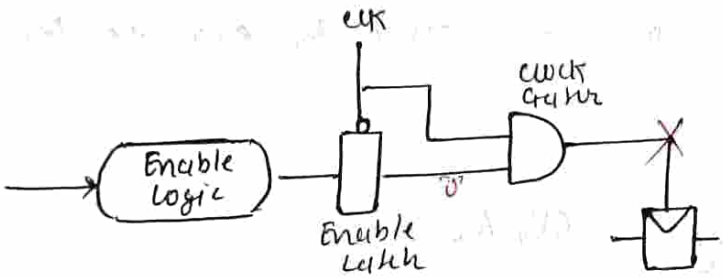
$= \frac{1}{16}$

Reducing Dynamic power

1. Clock Gating : It reduces the activity factor.

- The best way to reduce the activity factor is to turn off the clock to register in unused blocks.
- We can put an enable logic, such that when enable logic is 0 then o/p will not toggle. and if o/p does not toggle, we save power

- Saves clock activity ($\alpha = 1$)
- Eliminates all switching activity in the block
- Requires determining if block will be used.



2. Gate capacitance
   - Fewer stages of logic
   - Small gate sizes

3. Wire capacitance
   - Good floorplanning to keep communication blocks close to each other
   - Drive long wires with inverters or buffers rather than complex gates

* Things that we have used to reduce logical effort can be used to reduce the ~~cost~~ dynamic power also.

Dynamic Voltage & Frequency scaling

- while using the cell-phone and just browsing the internet mobile will start operating at 200MHz instead of 2GHz i.e. 10 times dynamic power consumption is reduced. called Dynamic Frequency scaling.

- If we are operating at 200MHz for past 1 minute then CPU will also lower the voltage to reduce the power consumption. So we use 0.7V supply instead of 1.2V because 200MHz can be easily operated by 0.7V (as an eg.) [Dynamic Voltage scaling]

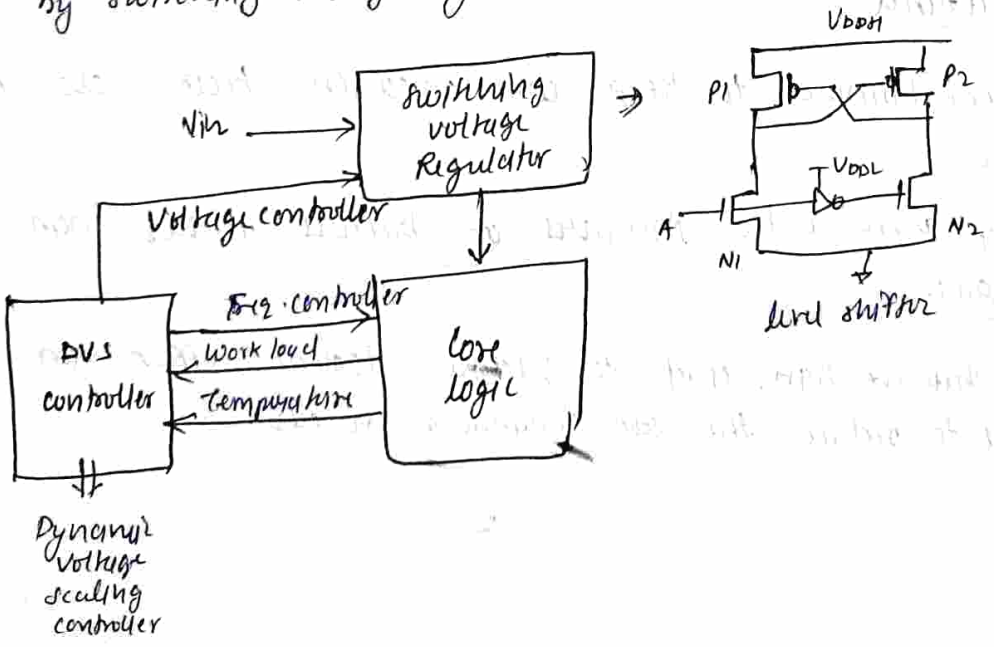
— At low voltages & 200MHz my delay will increase but at this time we don't need that much speed so we are going to save the power by reducing the voltage of operation.

$$P_{switching} = C_L V_{DD}^2 f_{sw}$$
$$= C_L V_{DD}^2 f_{sw}$$

So we reduced power consumption cubically.

Q: How to reduce the voltage?

→ by switching voltage regulator with the help of DVS controller.



* First reduce the freq. then voltage
* First increase the voltage then freq.
  - means at whatever freq you are operating, the voltage must be able to do at least that particular next freq.

Quick Refresh

Dynamic Power Consumption
- switching power
- short ckt power

short-ckt power
- Good Transition slope

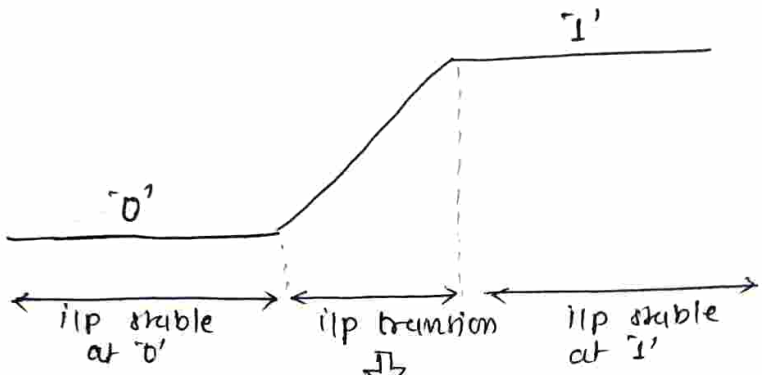
Reducing switching power
$\propto C V_{DD}^2 f$
- Activity factor
  - fully Random Data, AND/NAND Gate
  - Clock gating
  - Dynamic glitches
- capacitance reduction
- Dynamic voltage & freq. scaling

[max delay : SS 1.08V 125°C (Slow)
  min delay : FF 1.32V -25°C (Fast)

[max voltage :

2. Static Power :

- When there is no change in input means everything is stable (no transition is occuring) even then some power is getting consumed that is called static power (because some current is still flowing).
- In CMOS inverter, in a stable state there is no on current neither from the NMOS nor from the PMOS because only one of the NMOS or PMOS will be ON, not both will be ON.



when i/p is stable at '0' then PMOS is ON & NMOS is OFF hence at this time capacitor is completely charged & there is no direct path from V_{DD} to ground. But still some current flows is called static power

during this transition both NMOS & PMOS flow some current & the power calculated is dynamic power

Revisit 1B transistor chip for static power

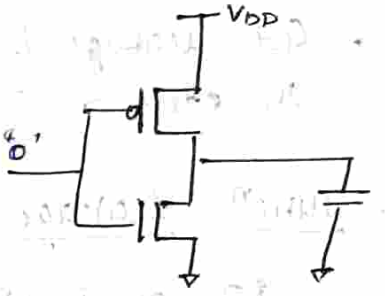
- Subthreshold leakage
  - (i) Normal V_t : 100nA/mm
  - (ii) High V_t : 10nA/mm
  - * All memories use High V_t 95% of logic gate use High V_t
- Gate leakage : 5nA/mm
- Junc^n leakage : Negligible

Note: When i/p = 0, and NMOS is off then still some current flows in NMOS i.e, NMOS is leaking means leakage current will flow through NMOS.

— * In static cond^n, the PMOS will not govern the amount of current flowing through it, the NMOS will govern the amount of current flowing through PMOS also.

- The overall current that can flow from the PMOS is limited by the current that will flow from the NMOS (why?)
- PMOS can't supply more current than the NMOS can sink.

— The capacitor is charged at V_{DD} + cap. can't source current from itself. So if NMOS wants to sink any current & PMOS is ready to give so much of current will the current come from the capacitor or PMOS?
— from PMOS



Static power calculation for 1B xnd order

- $W_{normal\text{-}V_t} = (50\times10^6)(12\lambda)\left(\frac{0.025\,\mu m}{\lambda}\right)(0.05) = 0.75\times10^6\,\mu m$
- $W_{high\text{-}V_t\,(logic)} = (50\times10^6)(12\lambda)\left(\frac{0.025\,\mu m}{\lambda}\right)(0.95)$

  $W_{high\text{-}V_t\,(memory)} = (950\times10^6)(4\lambda)\left(\frac{0.025\,\mu m}{\lambda}\right)(1)$

  $\therefore W_{high\text{-}V_t} = W_{high\text{-}V_t\,(logic)} + W_{high\text{-}V_t\,(memory)}$

  $= 109.25\times10^6\,\mu m$

- $I_{sub} = \dfrac{\left[W_{normal\text{-}V_t}\times\frac{100nA}{\mu m} + W_{high\text{-}V_t}\times\frac{10nA}{\mu m}\right]}{2}$ → divide by '2' because at right time either of PMOS or NMOS will be ON & half of half will flow through any of them.

  $= 584mA$

- $I_{gate} = \dfrac{\left[W_{normal\text{-}V_t} + W_{high\text{-}V_t}\right]\times\frac{5nA}{\mu m}}{2} = 2.75mA$

- $P_{static} = (584mA + 2.75mA)(1V) = 859\,mW$

Sources of Leakage

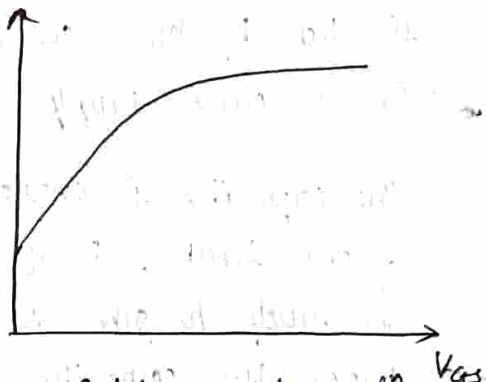
1. **Subthreshold conduction :**

- Transistor also conduct current when $V_{GS} \leq V_{th}$ & this conduction of current is called subthreshold cond^n.

* Transistor can't abruptly turn ON or OFF.

2. **Gate leakage :**

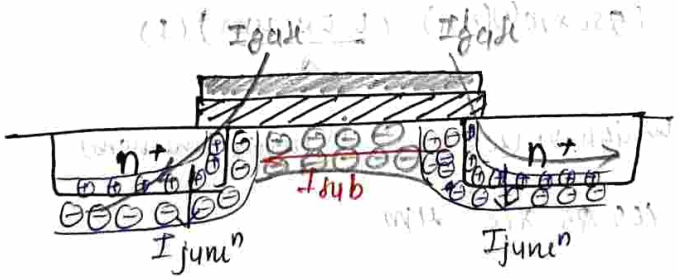
- Tunneling through ultra-thin gate dielectric
- Physical thickness of dielectric is of order 1nm, 2nm etc.
- Gate leakage is reduced by not reducing subthreshold cond^n the t_{ox} & using high K-dielectric. Dominant source in contemporary transistors



3. **Junc^n leakage :**

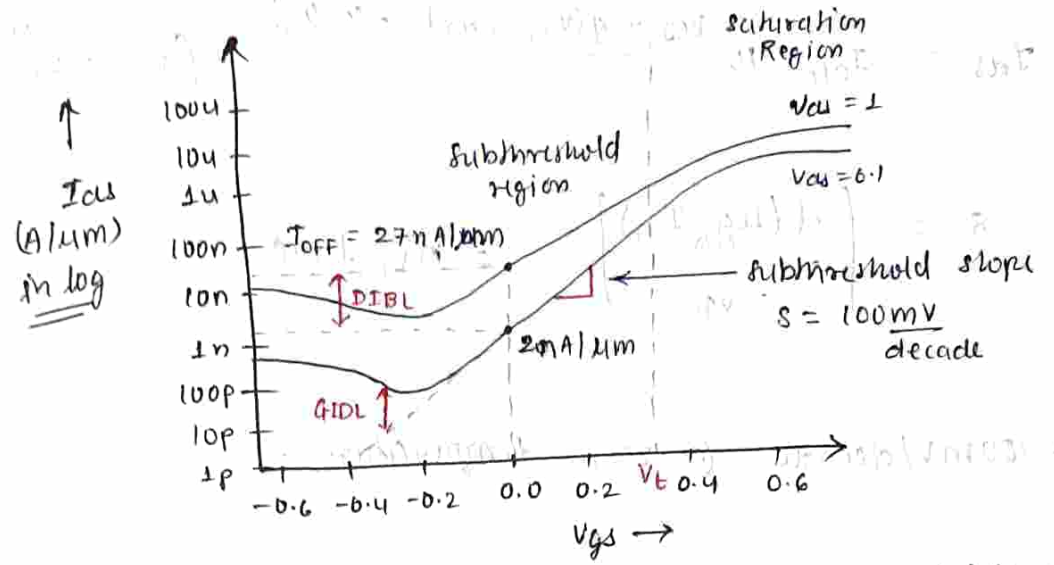
- RB pn-junc^n diode current b/w source-body & drain-body.

$$I_{OFF} = \frac{W}{L} I_0 e^{\left(\frac{V_{GS}-V_{th}}{n V_T}\right)} \left(1 - e^{-V_{DS}/\phi_t}\right)$$



* There is some subthreshold cond^n when current flow ~~from~~ between drain to source where ~~gate~~ n-o-s has a depletion region & np-junc^n also has depletion region.

① Subthreshold Leakage



* In a good transistor, the subthreshold slope 'S' lies in range of 60 mV/decade & 65 mV/decade.

Q. Why lower value of 'S' is better?

→ When we say subthreshold slope, S = 65mV/decade. ~~Below Vt~~ It means we get $\frac{1}{10}$th the current ~~below~~ when we go below V_t by 65mV.

- So if we get $\frac{1}{10}$th current at lower value of voltage below V_t then it is good.

* Subthreshold leakage means source to drain path. for e⁻ flow.
* There, it is also possible that w/o depletion region current can also flow.
* Subthreshold depends on V_t. (if V_t ↓, subthreshold current ↑)
* Subthreshold current also depends on body bias voltage ie, the relationship b/w source & body.
* Subthreshold depends on temperature, by the virtue of the component $\frac{KT}{q}$.

- Subthreshold leakage exponential with V_{gs}.

$$I_{ds} = I_{ds0} \cdot e^{\left(\frac{V_{gs} - V_{to} + \eta V_{ds} - K_y V_{sb}}{n V_T}\right)} \left(1 - e^{-\frac{V_{ds}}{V_T}}\right)$$

n → process dependent
typically : 1.3 – 1.7

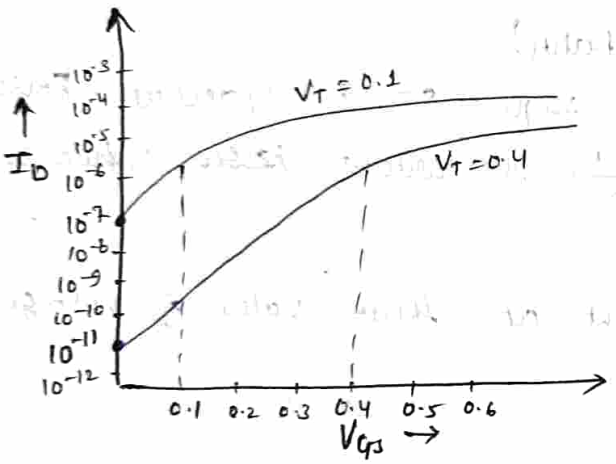
- Rewriting relative to I_{off} on log-scale

$$I_{ds} = I_{off}\, 10^{\frac{V_{gs} + \eta(V_{as} - V_{dd}) - K_V V_{sb}}{S}} \cdot \left(1 - e^{\frac{-V_{ds}}{V_T}}\right)$$

$$S = \left[\frac{d(\log_{10} I_{ds})}{d V_{gs}}\right]^{-1} = n V_T \ln 10$$

∴ $S \approx 100\,mV/decade$ @ room temperature

Subthreshold leakage wrt V_t



Typical value in 65nm tech.
for $\eta = 0.1$, $K_V = 0.1$
& $S = 100\,mV/decade$

(i) $I_{off} = 100\,nA/\mu m$ @ $V_t = 0.3V$

(ii) $I_{off} = 10\,nA/\mu m$ @ $V_t = 0.4V$

(iii) $I_{off} = 1\,nA/\mu m$ @ $V_t = 0.5V$

(*) As we decrease V_t (threshold voltage), subthreshold leakage current increases.

1. Drain-Induced Barrier lowering (DIBL)

- EF from drain affects the channel
- More pronounced in small transistors where the drain is closer to the channel.
- Drain voltage also affects V_t

$$V_t' = V_t - \eta V_{ds}$$

- High drain voltage causes current to increase.
  — more effective for small channel device.

* We can also get the subthreshold leakage w/o applying any gate voltage (or -ve V_{GS}) i.e. w/o any depletion region in the m-o-s region.

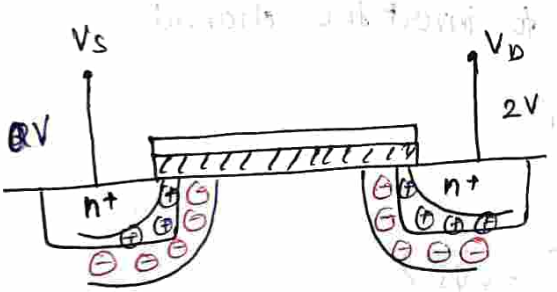
— By applying the drain voltage.

Subthreshold leakage ⇒ when device is off or $V_G < V_t$ for NMOS or no depletion region is present even though some e^- flows from source to drain, when we apply V_{DS}.

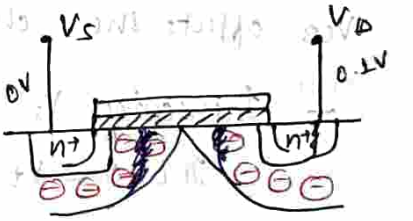
(ii) Both ON current & off current depends on V_{DS}.

Short-channel effect: As technology is scaling we are using shorter length devices. & this devices starts to behave differently as compared to standard set of eqn then it is called short-channel effect.

Q. Why this happens in short-channel device?

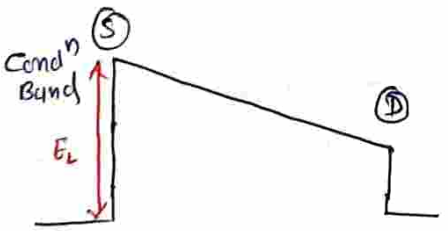


long channel



short channel

* for short channel as we reduced drain, energy barrier is reduced. (on ↑ V_D, Energy ↓)



e^- require high energy to move from source to drain

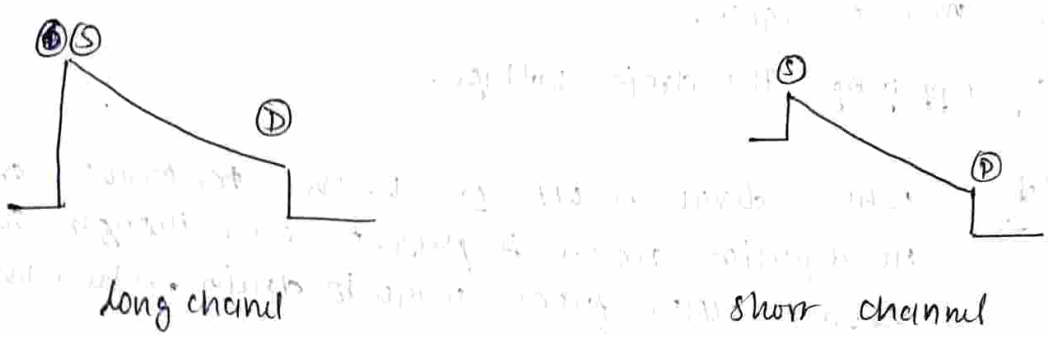
$E_L > E_S$

e^- require less energy to move from drain to source

* $E_S < E_L$ because at less V_D we can observe the subthreshold current.

Q) For short channel, how source is moving upwards as we apply V_d ?

→



Long channel Short channel

Because of the short channel, the depletion region b/w source & body and drain & body starts to overlap & hence we see such kind of effect in energy difference.

2. Body effect : (doubt)

- As you change Body voltage, V_t will change, causing the change in leakage.
- Body is the fourth terminal of transistor
- V_{SB} affects the charge required to invert the channel
  - Increasing V_s or decreasing V_b increases V_t

as, $$V_t = V_{to} + \gamma \left(\sqrt{\phi_s + V_{SB}} - \sqrt{\phi_s}\right)$$

- ϕ_s → surface potential at threshold

$$\phi_s = 2 v_T \ln\left(\frac{N_A}{n_i}\right)$$

↓

— Depends on doping level N_A
— Intrinsic carrier conc^n n_i

- Body effect coefficient

$$\gamma = \frac{t_{ox}}{\varepsilon_{ox}} \sqrt{2q \varepsilon_{si} N_A} = \frac{\sqrt{2q \varepsilon_{si} N_A}}{C_{ox}}$$

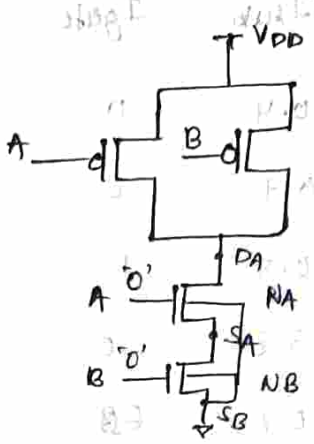
- For small source-to-body voltage, treat as linear

$$V_t = V_{to} + K_\gamma V_{SB}$$

where, $K_\gamma = \frac{\gamma}{2\sqrt{\phi_s}} = \frac{\sqrt{\frac{2\varepsilon_{si} N_A}{V_T \ln\left(\frac{N_A}{n_i}\right)}}}{2 C_{ox}}$

Stack effect:

- At A=1, B=0 $\Rightarrow S_A = V_{DD}$
- At A=0, B=0 $\Rightarrow S_A$ = Any floating value



Leakage through 2-stack reduces 10 times

↳ At A=0, B=0 means both N_A & N_B are off and drain of N_A (D_A) is at V_{DD}, so this stack of N_A & N_B will leak.

↳ However, S_A settles to a point, that each transistor has the same current.

— If ↓ voltage at S_A is small then N_A will see much smaller DIBL effect & will leak less

— As voltage at S_A rises, V_{gs} for NA becomes -ve, reducing its leakage

* Hence we would expect that the series transistor leak less.

↳ At A=0, B=0, Voltage at S_A is floating hence due to body effect $V_{SB} \neq 0$ hence $V_t \uparrow \Rightarrow$ leakage ↓.

Q. ? If we connect source to Body directly, then we will never observe body effect. Then why don't we connect Body to source directly.

→ See through the physical structure of the device & layout.

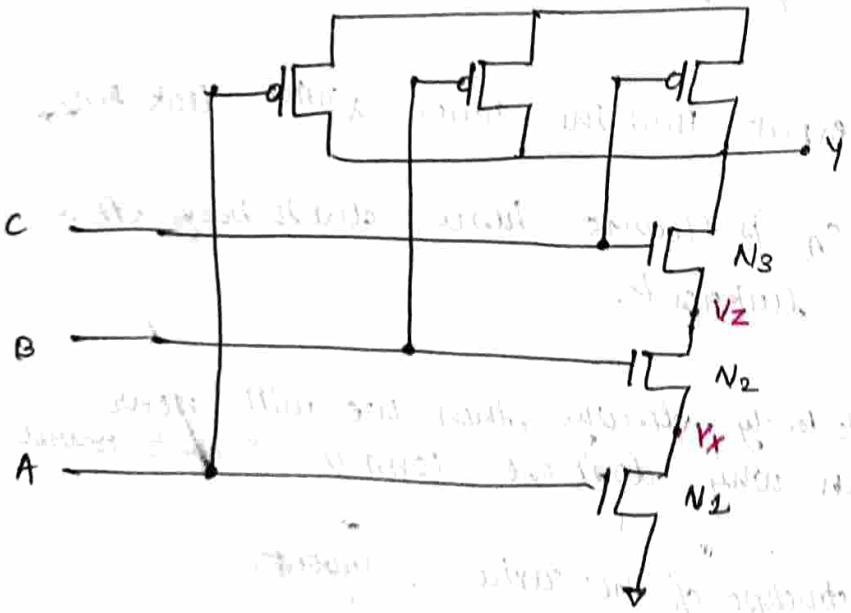
NAND3 Leakage Example :

- 100nm process

$I_{gn} = 6.3nA$ $I_{gp} = 0A$

$I_{offn} = 5.63nA$ $I_{offp} = 9.8nA$

Input state (ABC)	I_{sub}	I_{gate}	I_{total}	V_x	V_z
000	0.4	0	0.4	Stack effect	Stack effect
001	0.7	0	0.7	Stack effect	$V_{DD} - V_t$
010	0.7	1.3	2.0	intermediate (floating value)	intermediate
011	3.8	0	3.8	$V_{DD} - V_t$	$V_{DD} - V_t$
100	0.7	6.3	7.0	0	Stack effect
101	3.8	6.3	10.1	0	$V_{DD} - V_t$
110	5.6	12.6	18.2	0	0
111	28	18.9	46.9	0	0



Subthreshold Leakage control

- Leakage & Delay trade off :
  - Aim for low leakage in sleep (inactive) & low delay in active mode

- To Reduce leakage :
  (i) Increase V_t : multiple V_t
  - Use low V_t only in critical circuits
  (ii) Increase V_s : stack effect
  - I/P vector control in sleep mode
  (iii) Decrease V_b
  - Reverse Body bias in sleep mode
  - Forward Body bias in active mode

Q. Difference b/w sleep mode & clock gating.

→ Sleep mode involves much more than just enable signal on the clock gating path.

(2) Gate Leakage :

- Carrier tunnel through very thin gate oxides
- Exponentially sensitive to t_{ox} & V_{DD}

$$I_{gate} = WA \left(\frac{V_{DD}}{t_{ox}}\right)^2 e^{\frac{-B t_{ox}}{V_{DD}}}$$

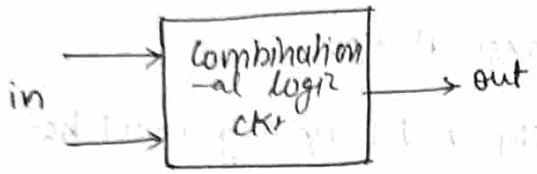
① V_{DD} ↓, I_{gate} ↑ & ② t_{ox} ↓, I_{gate} ↑

- A & B are technology constants
- Greater for e^-s
- * So NMOS gate leaks more.

- Negligible for older processes ($t_{ox} > 20$ Å)
- Critically important at 65nm technology & below 10.5 Å t_{ox}.

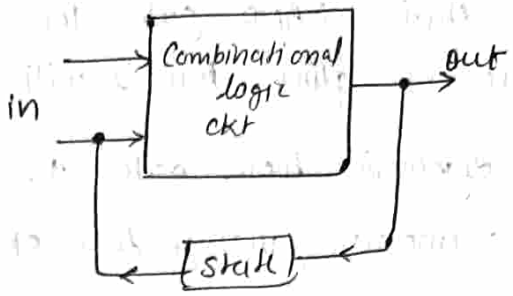
Combinational Circuit Design

Combinational v/s sequential logic



Combinational

Output = f(In)



Sequential

Output = f(In, Previous In)

storage ckt
SRAM → Latch
DRAM → capacitor

Static CMOS v/s Dynamic

Static CMOS

- At every point in time (except during switching transients) each o/p is connected to either V_{DD} or V_{SS} via a low resistive path.
- Outputs of the gates are driven at all times - to the value of Boolean func^n implemented by the ckt

Dynamic logic

- * Rely on temporary storage of signal values on the capacitance of high impedance ckt node
- Advantage → Area gets reduced
- Disadvantage → we get some race around cond^n & cascading conc^n
  - Resemblence of clock

Eg. In static memory, you will have 6T transistors memory cell & in dynamic memory, you will have capacitors in which you store charge

* So, in dyamic logic for some combinational state where the o/p is not driven by anything else but is dependent on some charge stored in capacitor.

- Now, if due to crosstalk or leakage that charge gets dissipated then there is no way to recover that charge during the cycle & data is lost.

* In static NAND gate, for any i/p glitch the o/p will also have some glitch but it will not change it's o/p.

In dynamic logic gate, for any glitch in the i/p you will have a permanent voltage loss at the o/p.

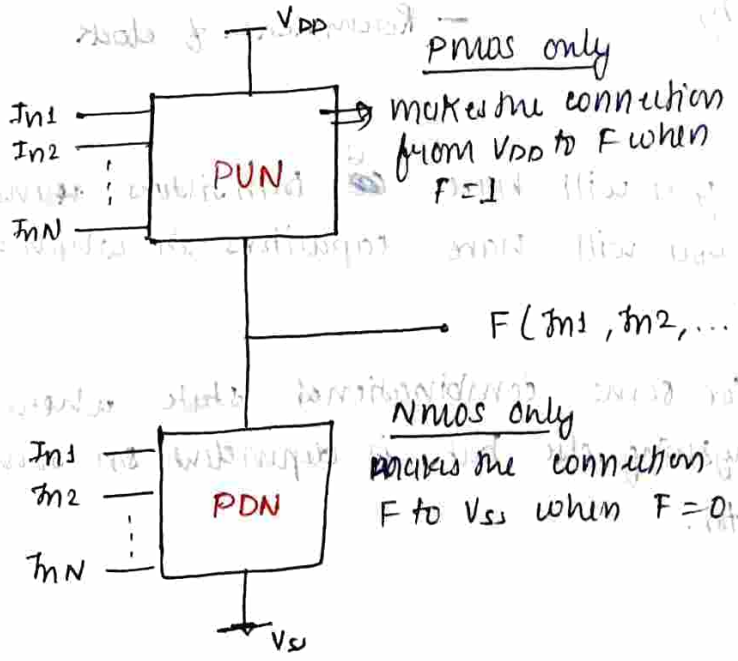
* For designing AOI domino logic →
  - You have to look at the various failure modes of dynamic logic
  - You have to take care of the capacitances
  - You have to check the noise immunity of your gate

Q. DRAM cell is not a dynamic logic then why dynamic is there?

→ It is called dynamic RAM cell & not dynamic logic

* dynamic because it is storing something on a capacitor & you need to refresh it you need to ensure that some leakage or whatever has happened it is kind of recovered on regular basis.

1. Static Complementary MOS [Static cmos] (Rabaey Pg. 226)



PMOS only
makes the connection from V_{DD} to F when F=1

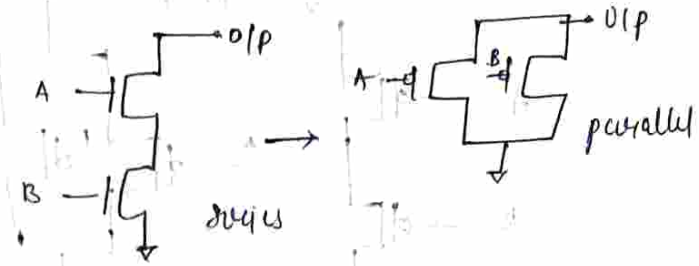
NMOS only
makes the connection from F to V_{SS} when F=0.

- The func^n of PUN n/w is to provide a common connection b/w the o/p & V_{DD} anytime the o/p of logic gate is meant to be 1 (based on i/ps).

* PUN & PDN are dual of each other

Eg: For NAND gate

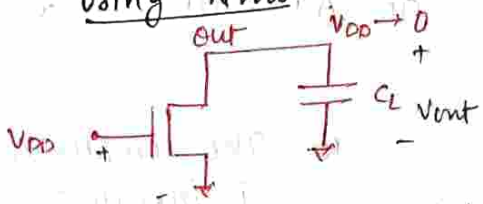
if PDN is, then PUN is.



Threshold Drop

Q: Why PDN is constructed using NMOS & PUN using PMOS?

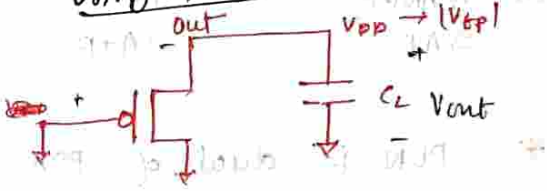
PDN ⇒ Using NMOS



at i/p = V_{DD}, NMOS = ON always

as $V_{GS} = V_{DD} - 0 = V_{DD}$ always

Using PMOS



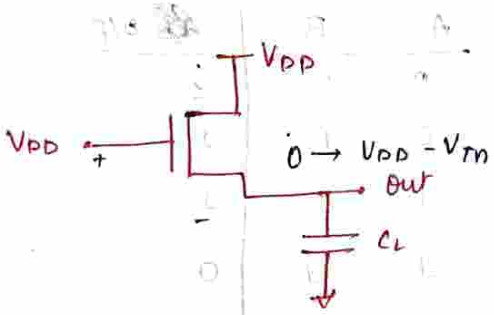
at i/p = 0, $|V_{GS}| = V_{DD} - V_{out}$

as $V_{GS} = -|V_{tp}|$

then PMOS → OFF & C_L will not completely discharge.

as $|V_{GS}| < |V_{tp}|$

PUN ⇒ Using NMOS



$V_{GS} = V_{DD} - V_{out}$

∴ NMOS ON till $V_{GS} > V_{tn}$

at $V_{out} = V_{DD} - V_{tn}$

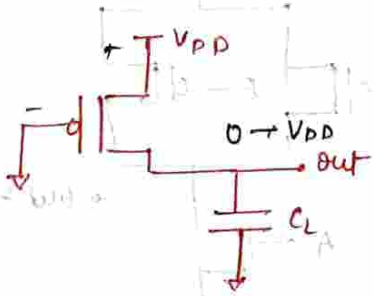
$V_{GS} = V_{DD} - (V_{DD} - V_{tn})$

$V_{GS} = V_{tn}$

∴ NMOS becomes off

& No charging will occur.

Using PMOS

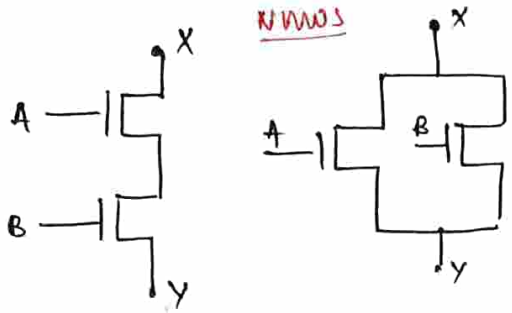


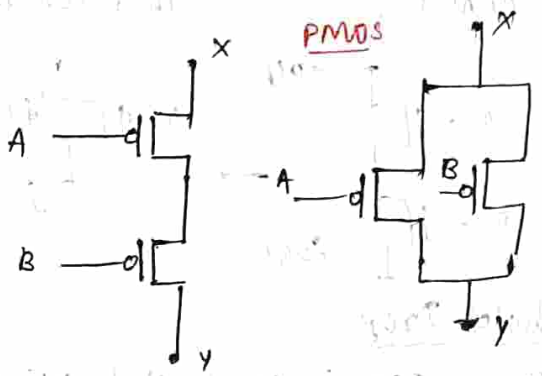
$V_{SG} = V_{DD}$ ∴ PMOS ⇒ ON (always)

∴ C_L charge to V_{DD}

* NMOS transistor pass a "strong" → 0 but "weak" → 1
* PMOS transistor pass a "weak" → 0 but "strong" → 1

Transistor in series & Parallel





① $Y = X$
ckt A AND B
$\Rightarrow AB$

② $Y = X$
ckt A OR B
$\Rightarrow A+B$

③ $Y = X$
ckt $\overline{A} \cdot \overline{B} = \overline{A+B}$

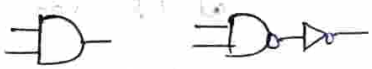
④ $Y = X$
ckt $\overline{A} + \overline{B} = \overline{AB}$

* PUN is dual of PDN

① + ④ i.e., $AB \xrightarrow{\text{dual}} \overline{A+B}$ (NAND)

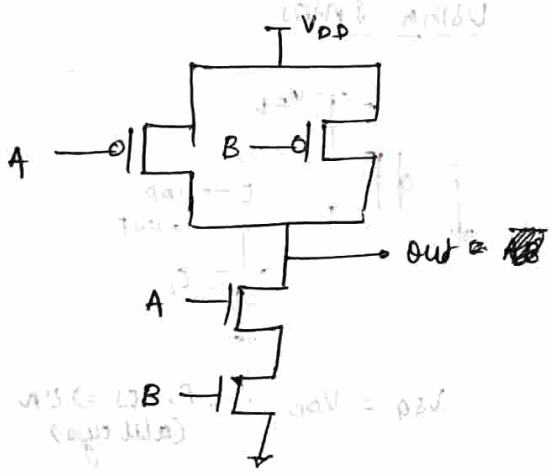
② & ③ & $A+B \xrightarrow{\text{dual}} \overline{AB+B}$ (NOR)

* The complementary gate is inverting



AND = NAND + INV

Example : NAND Gate



(NAND)

A	B	O/P
0	0	1
0	1	1
1	0	1
1	1	0

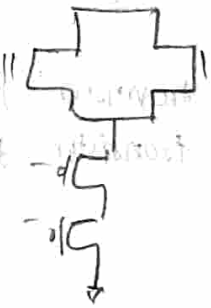
(OR)

A	B	O/P
0	0	0
0	1	1
1	0	1
1	1	1

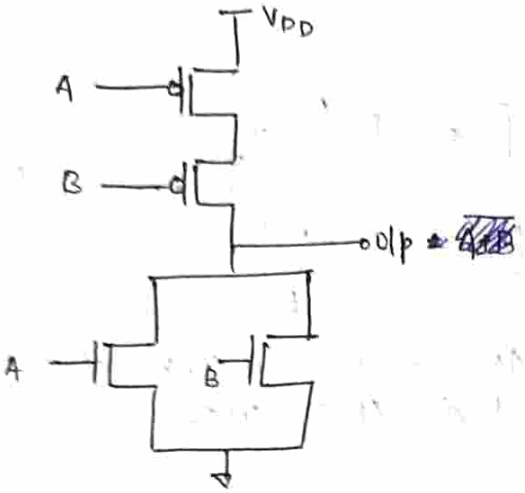
PDN : $G = AB$ → cond^n to GND

PUN : $F = \overline{A} + \overline{B}$ → cond^n to V_{DD}

$\overline{G(in1, in2, \ldots inN)} = F(\overline{in1}, \overline{in2}, \ldots \overline{inN})$



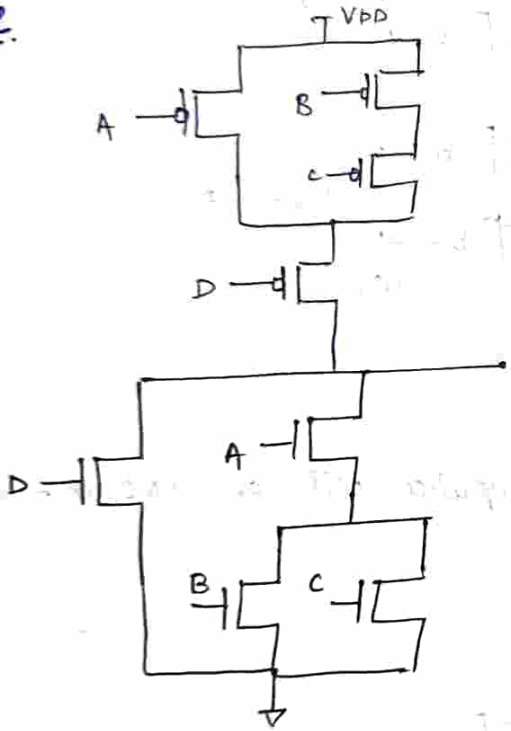
Example : NOR Gate



A	B	o/p
0	0	1
0	1	0
1	0	0
1	1	0

$\therefore o/p = \overline{A+B}$

Example : Complex CMOS Gate



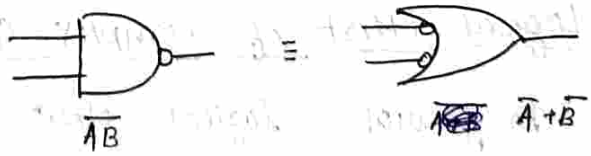
* while constructing the complex CMOS gate from given expression
  (i) Draw PDN for expression which is complement
  (ii) make the dual of PDN in PUN using PMOS.

$\Rightarrow o/p = \overline{A(B+C)+D}$

Bubble Pushing : (Weste Pg. 325)

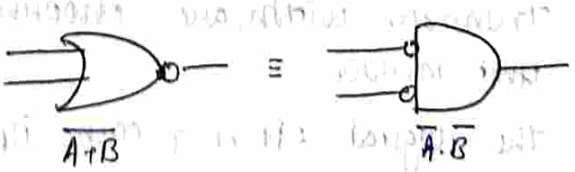
(i) A NAND is equivalent to an OR of inverted i/p.

$\overline{A \cdot B} = \overline{A} + \overline{B}$



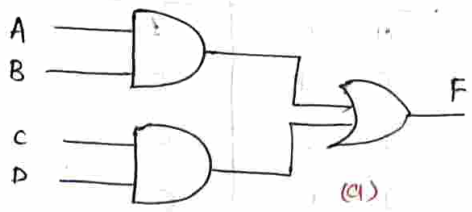
(ii) A NOR is equivalent to an AND of inverted i/p.

$\overline{A+B} = \overline{A} \cdot \overline{B}$

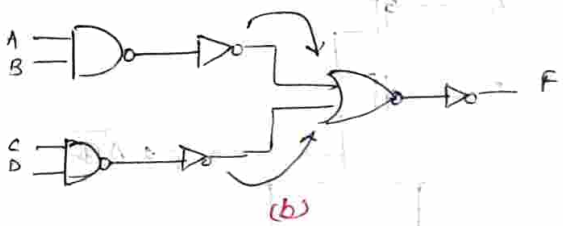


Que- Design a ckt to compute $F = AB + CD$ using NANDs & NORs

→



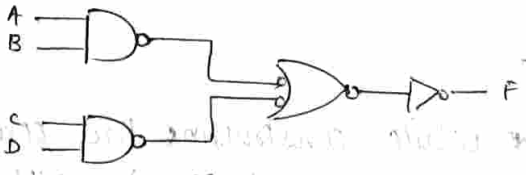
(a)



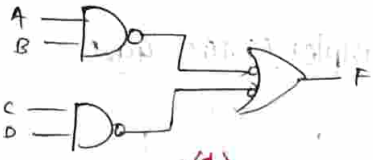
(b)

CKT consists of two AND & one OR gate

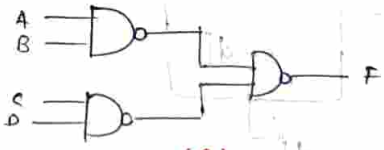
ANDs & ORs are converted to basic CMOS stages



(c)



(d)

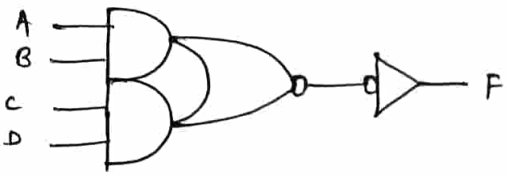


(e)

Bubble pushing is used to simplify the logic to three NANDs

Complex Gates (Weste Pg. 329)

- The func^n $F = AB + CD$ can be computed with an AND-OR-INVERT-22 (AOI22) gate and an inverter.



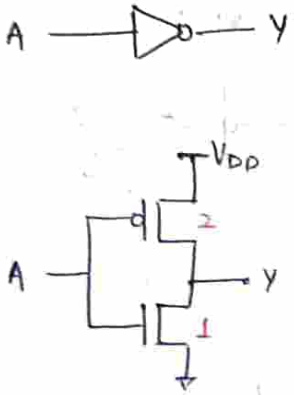
1. Logical Effort of Complex Gates

- In general, logical effort of compound gates can be different for different inputs
- Transistor widths are chosen to give the same drive as a unit inverter
- The logical effort of each i/p is the ratio of the i/p cap of that i/p to the i/p cap of inverter.

- The parasitic delay is roughly estimated from the total diffusion capacitance on the o/p node by summing the sizes of the transistors attached to the o/p.

Unit Inverter

$Y = \overline{A}$

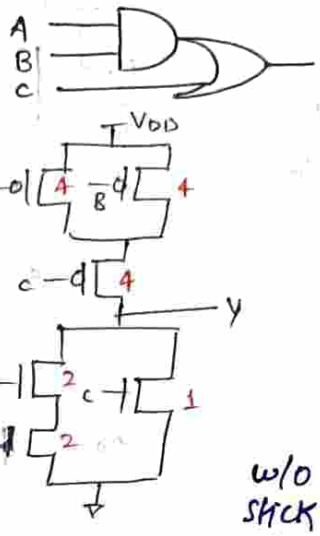


$g_A = 3/3$

$P = 3/3$

AOI 21

$Y = \overline{AB + C}$



$g_A = 6/3$

$g_B = 6/3$

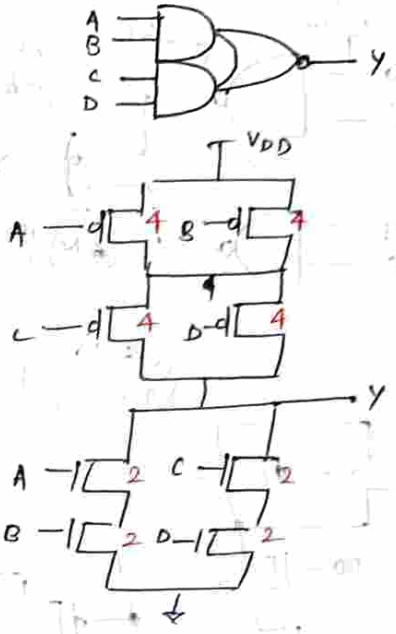
$g_C = 5/3$

$P = 7/3$

w/o stick Diagram

AOI 22

$Y = \overline{A \cdot B + CD}$



$g_A = 6/3$, $g_B = 6/3$

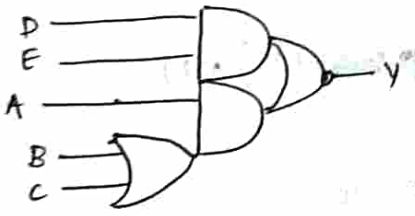
$g_C = 6/3$, $g_D = 6/3$

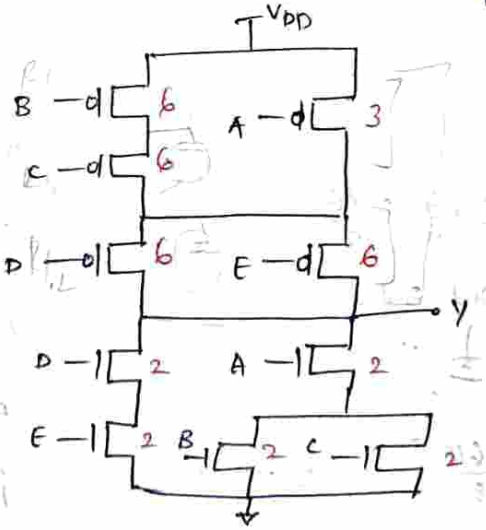
$P = 12/3$

while Designing $R_{PUN} = R_{PDN}$ Area should be less.

Complex AOI

$Y = \overline{A(B+C) + DE}$





$g_A = 5/3$

$g_B = 8/3$

$g_C = 8/3$

$g_D = 8/3$

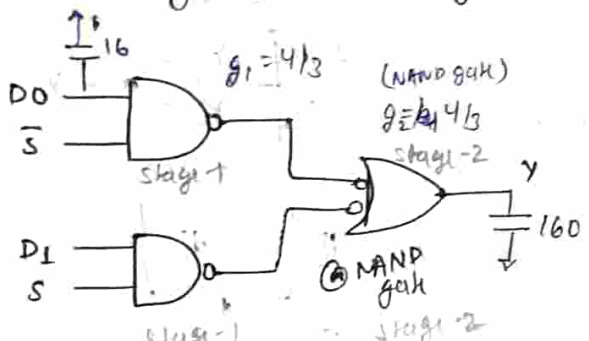
$g_E = 8/3$

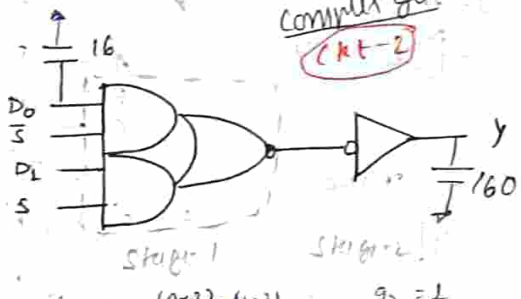
$P = 16/3$

* For AOI 21 gate, the logical effort is lower for OR terminal (C) than for the AND terminals (A, B).

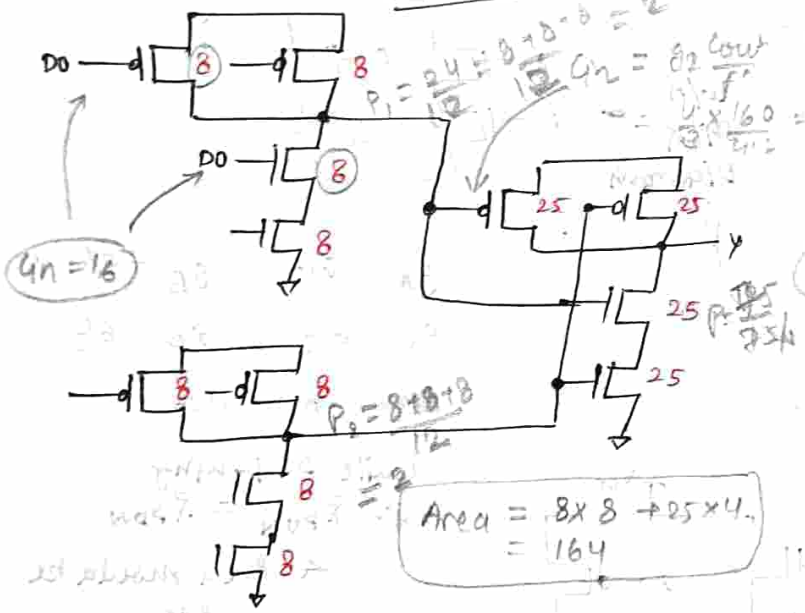
2. Delay & Gate size of complex gates

- The multiplexer has a maximum i/p capacitance of 16 units on each i/p. It must drive a load of 160 units. Estimate the delay of two designs

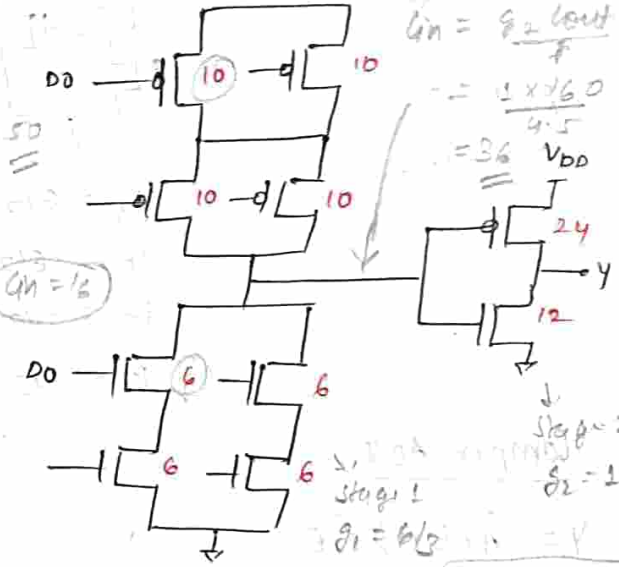




(ckt-1) Normal gate



Area $= 8 \times 8 + 25 \times 4$
$= 164$



Area $= 10 \times 4 + 6 \times 4 + 36 = 100$

$G = g_1 g_2 = \frac{4}{3} \times \frac{4}{3} = \frac{16}{9}$

$B = 1$ (∵fanout = 1)

$H = \frac{160}{16} = 10$

$F = GBH = \frac{160}{9}$

$\hat{f} = (F)^{1/2} = 4.2$

$P = 2 + 2 = 4$ (No of i/ps are 2 at stage-1 hence no. of i/ps are 2 at stage-2)

$D = N\hat{f} + P = 2(4.2) + 4$

$D = 12.4$

$\Rightarrow D_{abs} = 12.4\tau$

for i/p D0

$G = \left(\frac{6}{3}\right) \times (1)$

$B = 1$

$H = \frac{160}{16} = 10$

$F = GBH = \frac{60}{3} = 20$

$\hat{f} = (F)^{1/2} = 4.5$

$P = 4 + 1 = 5$

$D = N\hat{f} + P$

$= 2 \times (4.5) + 5$

$D = 14$

$D_{abs} = 14\tau$

* The delay for complex gate is more than the delay of normal gate. But area of complex gate is less as compared to normal gate

* OR lesser the i/p capacitance at the o/p stage lesser the area of the ckt.

[Compare delay & Area of both the ckts] → you can also see in ckt for wiring in ckt.

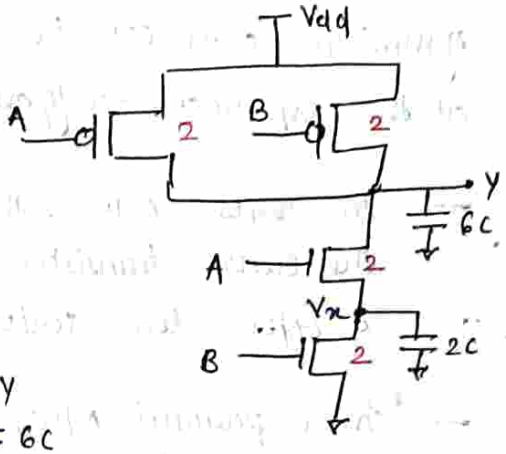
⇒ (*) since the area is lesser, the parasitic will also be lesser & hence in the practical you will see the delay of complex gate is less.

Q. If my ckt has diff. g values for different i/ps then for calculation of the delay which i/p should I consider?

→ For calculation of prop. delay, take worst g. (i.e high g)
For calculation of cont. delay, take best g. (i.e low g)

Input Ordering Delay Effect :

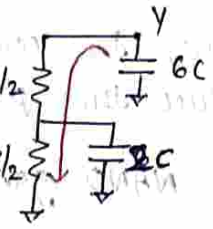
- Complex gates like AOI21, are asymmetrical, that one i/p sees less capacitance than another.
- NAND & NOR gates symmetrical but actually have slightly different logical effort & parasitic delay for the different inputs.
- Consider a ckt of 2-i/p NAND gate annotated with diffusion parasitic.



— Consider the falling o/p transition occuring when one i/p is held at '1' & other varies from '0' to '1'.

(i) let A = 1, B = 0 ↓, $V_x = V_{DD} - V_t$
then A = 1, B = 1

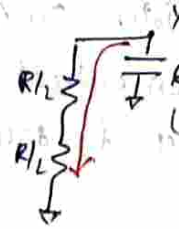
Elmore delay = $6C(R) + 2C(\frac{R}{2})$ (worst case delay)
= 7RC
= $\frac{4}{3}(3RC) = 4.33\tau$



(2C kobhi discharge karna pdega)

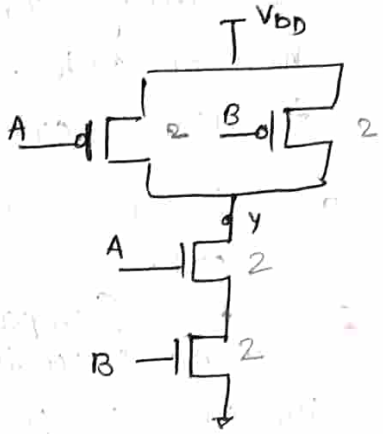
(ii) let A = 0 ↓, B = 1, $V_x = 0$
then A = 1, B = 1.

Elmor delay = $6C(R) = 6RC$
(but case delay) = $3(3RC) = 2\tau$



(2C charge hi nhi hua to discharge bhi nhi karna pdega)

- outer i/p → i/p closer to supply rail (eg. B)
  inner i/p → i/p closer to the o/p (eg. A)



* Parasitic delay is smallest when inner i/p switches last because the intermediate nodes have already been discharged.
  - Signal arriving at the inner gate i/p must arrive late, if any one i/p is coming late

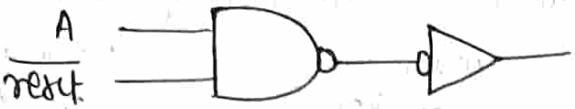
* The inner i/p has a slightly higher logical effort because the intermediate node x tends to rise & cause -ve f/b when inner i/p turns ON.

- Which ever i/p provides shortest delay to the o/p should be designed to be arriving last in your path.
  - here delay is smallest for i/p A ~~=0 & 1 when B=1 already~~ means A should arrive last.

Asymmetric Gates : (Write Pg. 332)

- when one i/p is far less critical than another, even nominally symmetric gates can be made asymmetric to favour the late i/p at the expense of early one.
  - In series N/w, it involves connecting the early i/p to the outer transistor & making the transistor wider so that it offers less series resistance when the critical i/p arrives
  - In a parallel N/w, the early i/p is connected to a narrower transistor to reduce the parasitic capacitances.

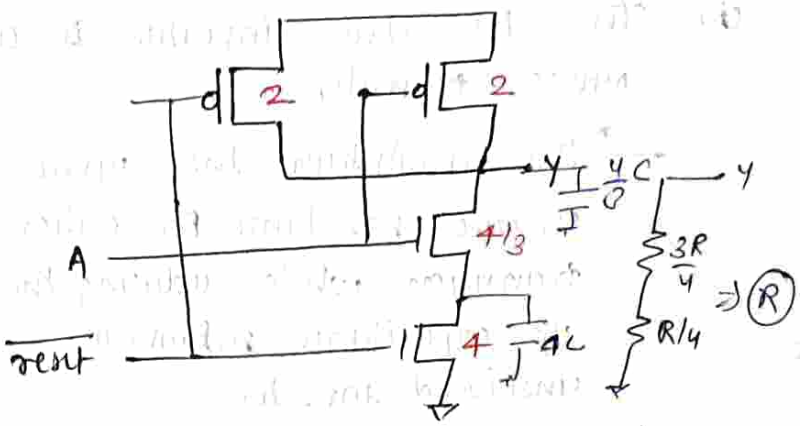
Eg. suppose i/p A of NAND gate is most critical
- use smaller transistor on A (less capacitance)
- Boost size of non-critical path
- so total resistance is same



$g_A = \frac{\frac{4}{3}+2}{3} = \frac{10}{9} = 1.01 \downarrow$

$g_B = \frac{4+2}{3} = 2 \uparrow$

$g_{total} = g_A + g_B = \frac{28}{9}$



* but total logical effort ↑.
* logical effor for critical path (A→Y) ↓.

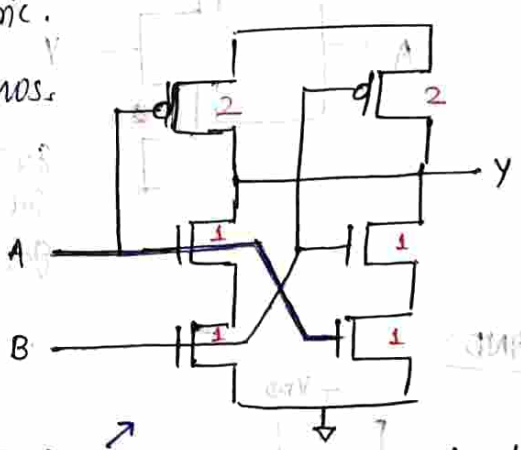
(*) This reduces the delay but area increases

- The improvement in logical effort of i/p A comes at the cost of much higher effort on the reset i/p.

* We will only use this asymmetric gate when we have complete understanding of critical path.

Symmetric gates

- Input can be made perfectly symmetric.
- In symmetric gate you have two NMOS. each of same size but your both the i/ps are coming randomly at the same time. So I get equal delay from A to Y & B to Y.
  - — 1st of A is close to o/p
    1st of B is close to o/p.



This is used when your i/p data A & B is completely random.

Skewed Gates → This will be used in Pulse width in sequential ckt.

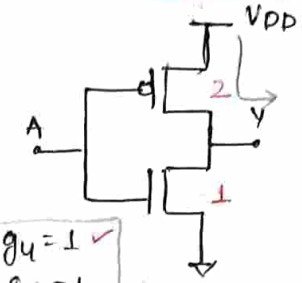
- HI - skew gate to favour the rising o/p transition.
  LO - skew gate to favour the falling o/p transition.
  - — * This favouring can be done by decreasing the size of the non-critical transistor.
- g_u → logical effort for rising (up) transition of o/p.
  g_d → logical effort for Falling (down) transition of o/p.

(*) And these are the ratio of the i/p capacitances of the skewed gate to the i/p capacitances of an unskewed inverter with equal drive for the transition.

(i) The HI-skew inverter is constructed by downsizing the NMOS transistor [O/P => 0 → 1]

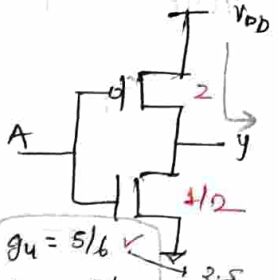
— This inverter gives the same effective resistance for critical transition while reducing the i/p capacitance relative to unskewed inverter.

unskewed



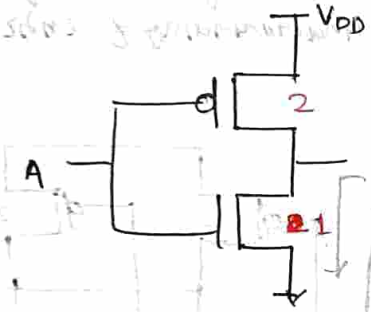
$g_u = 1$ ✓
$g_d = 1$
$g_{avg} = 1$

HI-skewed



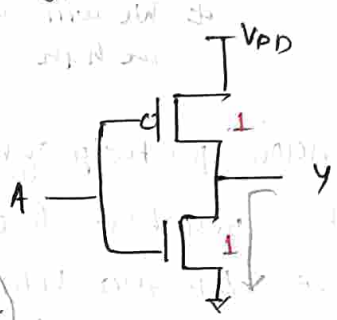
$g_u = 5/6$ ✓
$g_d = 5/3 = \frac{2.5}{3}$
$g_{avg} = 5/4$
$= g_u \times 2$

(ii) The LO-skew inverter is constructed by downsizing the PMOS transistor & keeping the NMOS of same size so that effective resistance for critical path remains same & reducing the i/p capacitance hence logical effort decreases.



unskewed

$g_u = 1$
$g_d = 1$ ✓
$g_{avg} = 1$

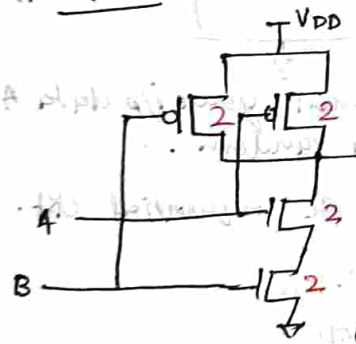


LO-skewed

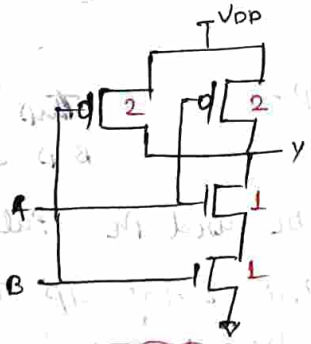
$g_u = 4/3 = g_d \times 2$
$g_d = 2/3$ ✓
$g_{avg} = \frac{6/3}{2} = 1$

$g_{avg} = \frac{g_u + g_d}{2}$

NAND2

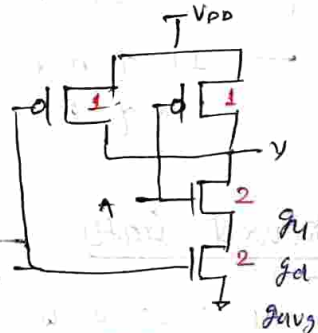


$g_u = 4/3$
$g_d = 4/3$
$g_{avg} = 4/3$



HI-skew

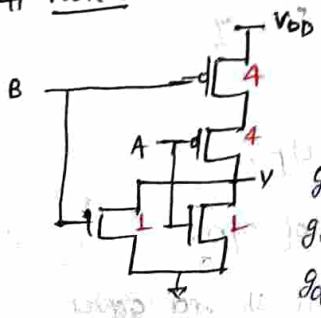
$g_u = 3/3 = 1$
$g_d = g_u \times 2 = 2$
$g_{avg} = 3/2$



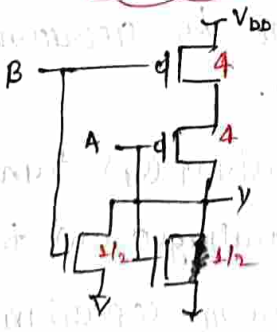
$g_u = g_d \times 2 = 2$
$g_d = 3/3 = 1$
$g_{avg} = 3/2$

LO-skew

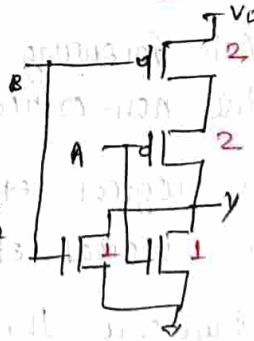
NOR2



$g_u = 5/3$
$g_d = 5/3$
$g_{avg} = 5/3$



$g_u = 4.5/3 = 9/6 = 3/2$
$g_d = g_u \times 2 = 3$
$g_{avg} = 9/4$



LO-skew

Since 2 times more effort will be required for o/p.

$g_u = g_d \times 2 = 2$
$g_d = 3/3 = 1$
$g_{avg} = 3/2$

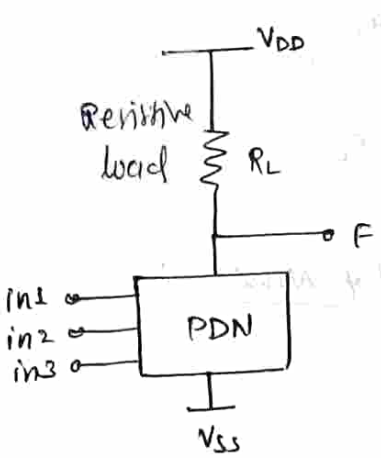
* Skew gates are used only when one transition is imp.
* Skewed gates work particularly well with dynamic ckt.

Ratioed Logic : [Rabaey Pg. 249]

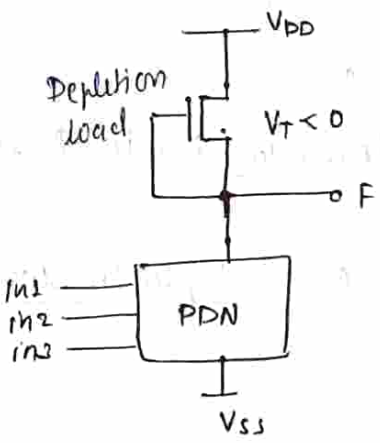
- In static CMOS logic design, for 'N' no. of inputs we have to use atleast '2N' no. of devices and on that 'N' devices are PMOS which is larger in size

Goal : To reduce the number of devices over complementary CMOS

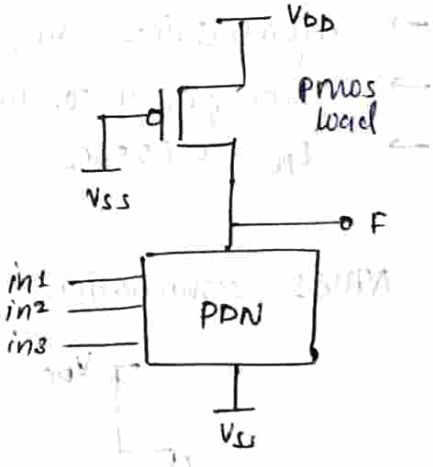
→ for that we can use resistive load, depletion load NMOS and grounded PMOS (pseudo NMOS) in place of PUN.



Resistive load



Depletion load NMOS



Pseudo NMOS

- But the proble with this kind of load is
  - noise margin ↓ (bad)
  - propagation delay ↓ (good) (How?) ⇒ because net load C ↓.
  - power dissipation ↓ (Leakage ↑) (bad). Cap CL ↓ ⇒ Dy. power ↓) good
  - output will never go to exact zero
  - a continous static current from VDD to o/p.

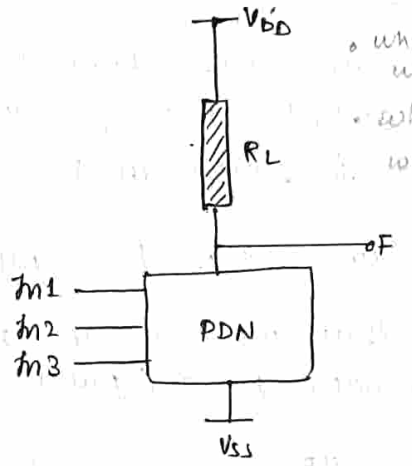
Q. What if increase the resistance of our load such that we get clear '0'?

→ If we do so then,
- current through load will be very less & charging (rise time) for o/p increases.
- (?) High resistive load increases the area also

(i) NMOS in PDN + resistive load :

- $V_{OH} = V_{DD}$
  - ↳ min High o/p voltage
- $\uparrow V_{OL} = \frac{R_{PN}}{R_{PN} + R_L \downarrow}$ → large pull up
  - ↳ max Low o/p voltage

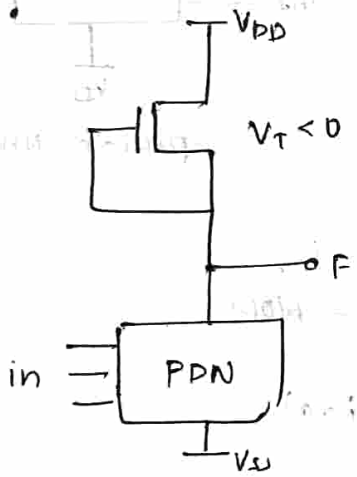
V_{OL} ⇒ must go to '0' for better switching. (∵ Noise margin ↑)



- what happens when $R_L \uparrow$?
- what happens when $R_L \downarrow$?

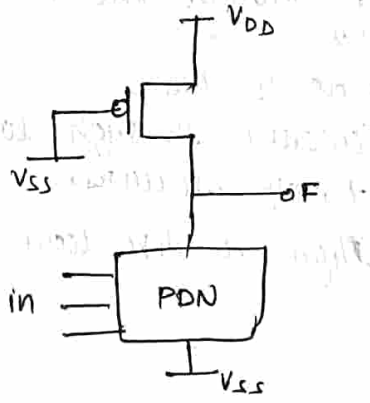
→ Asymmetrical response
→ Static power consumption ↑.
→ $t_{PL} = 0.69 R_L C_L$ ⇒ $t_{Prop\ delay}$ ↑ with ↑ in R_L.

(ii) NMOS combination in PDN + Depletion load of NMOS :



(iii) NMOS comb. in PDN + Pseudo-NMOS

- Computing the DC transfer ch of the pseudo-NMOS proceeds along paths similar to those used for its complimentary CMOS counterpart.
- The value of V_{OL} is obtained by equating the current through driver & load devices for $V_{in} = V_{DD}$.



- At this operating point, it is reasonable to assume that the NMOS device resides in linear mode (since o/p should ideally be close to 0V), while PMOS load is saturated.

$$K_n\left[(V_{DD}-V_{tn})V_{OL}-\frac{V_{OL}^2}{2}\right]=\frac{K_p}{2}(V_{DD}-|V_{tp}|)^2$$

assume, $V_T=V_{tn}=V_{tp}$

$$\therefore \quad V_{OL}=(V_{DD}-V_T)\left[1-\sqrt{1-\frac{K_p}{K_n}}\right] \quad \text{"from slides"}$$

$$V_{OL}\approx\frac{K_p(-V_{DD}-V_{TP})\cdot V_{DSAT}}{K_n(V_{DD}-V_{Tn})} \quad \text{"from Pabaey"}$$

$$V_{OL}\approx\frac{\mu_p W_p}{\mu_n W_n}\cdot V_{DSAT}$$

- In order to make V_{OL} as small as possible, the PMOS device must be sized much smaller than the NMOS PDN devices. [size of PMOS < size of NMOS] => take 'o' ja sake
- This will create a -ve impact on the propagation delay for charging up the o/p node since current provided by the PMOS device is limited. (Rising time degrades)

* A major disadvantage of the pseudo-NMOS gate is the static power dissipated when the o/p is low through the direct current path that exists b/w V_{DD} & GND. (short ckt current ↑)

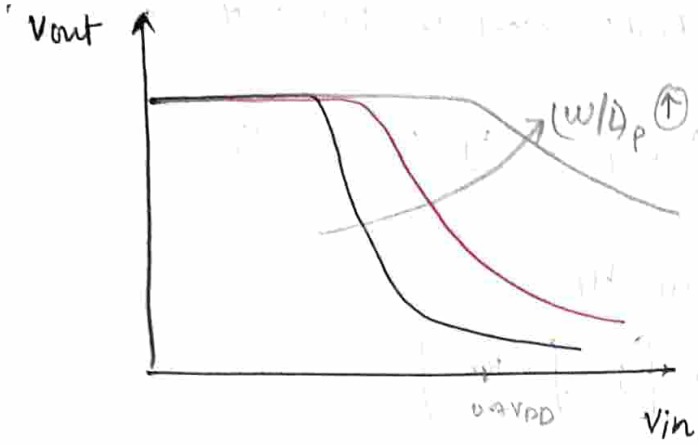
imp

** A larger pull-up device (R_L↓)
- improves the performance (good) → rise time↓
- increases static power dissipation (Bad) ∝ size
- lower Noise margin (ie V_{OL}↑) (Bad)
  ↳ Aache se 'o' nhi bana rha hai.

∴ To decrease V_{OL}, small PMOS is required.

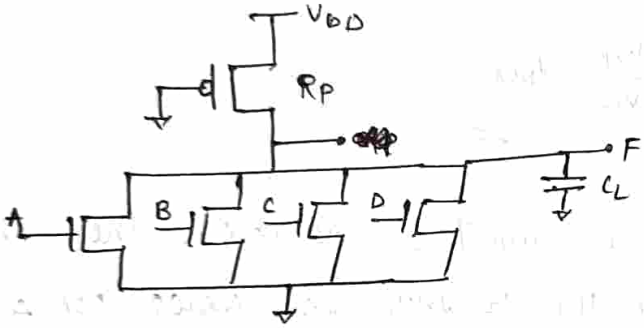
But small pull-up device, (R_L↑)
- increases rise time (Bad)
- decreases static power (Good) ∝ size
- V_{OL}↓ (Good) (Noise margin↑)

Pseudo NMOS – Voltage Transfer c/s



pmos size ↑ ⇒ $(\frac{R_p}{R_n})$ ↑
⇒ V_{OL} ↑ ⇒ NM ↓

pseudo pmos :



on ↓ R_p, V_{OL} ↑

(i) $R_p = 10K\Omega$, $R_n = 2K\Omega$

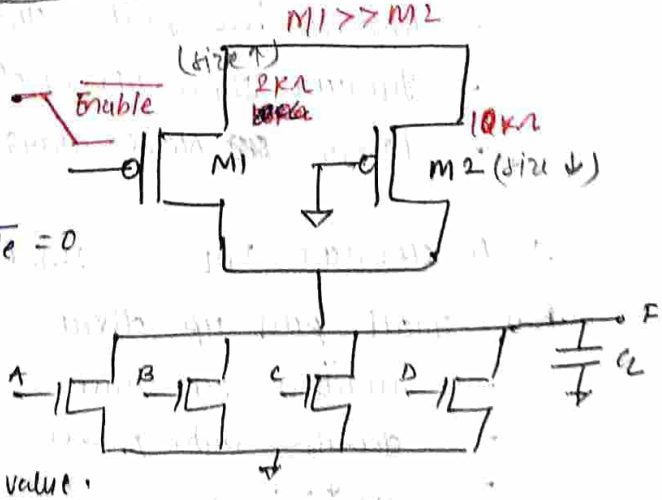
$V_{OL} = \frac{0.5}{10.5K\Omega} V_{DD} = \frac{1}{21} V_{DD} = 5\% V_{DD}$

(ii) $R_p = 2K\Omega$, $R_n = 2K\Omega$

$V_{OL} = \frac{0.5K\Omega}{2.5K\Omega} V_{DD} = 20\% V_{DD}$

(iv) Improved load : Imp

- Previously, on ↓ R_p, V_{OL} ↑ & delay also ↓, so we how to overcome this
  - Add less resistive (2KΩ) pmos in parallel to the already present (10KΩ) pmos, so that overall resistance will be equal to the resistance of low resistive pmos. (static power dissipation ↑)
  - During the evaluation only nmos added pmos will be ON & effective Resistance is 2K so our ckt becomes fast.
  - And due to nmos enable signal the M1 becomes ON at enable = 0 so o/p gets to V_{DD} from low resistive path. (∴ rise delay ↓)
  - And from the high resistive M2 the device becomes gets to V_{OL} = low value.

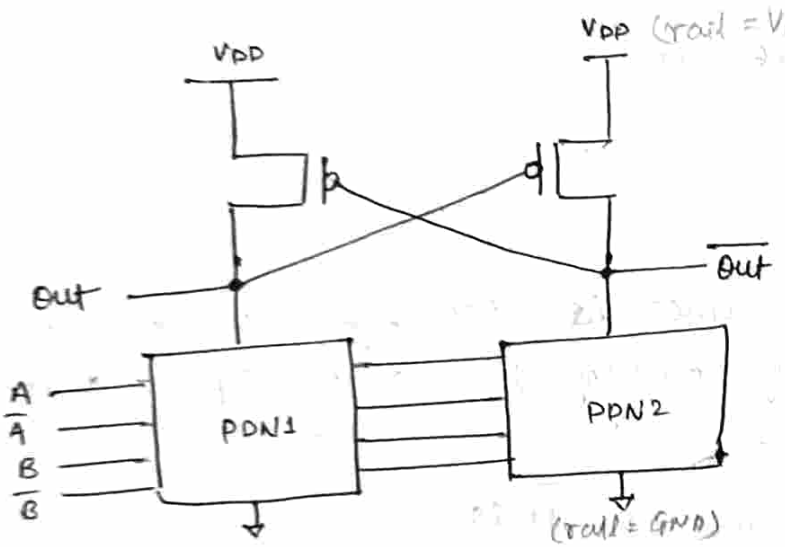


- $\overline{Enable} = 1$, charging path = m2 (highly resistive) hence V_{OL} ↓.
- $\overline{Enable} = 0$, charging path = m1 (low resistance) hence delay for rise ↓.

⇒ for this load all the things are achieved.

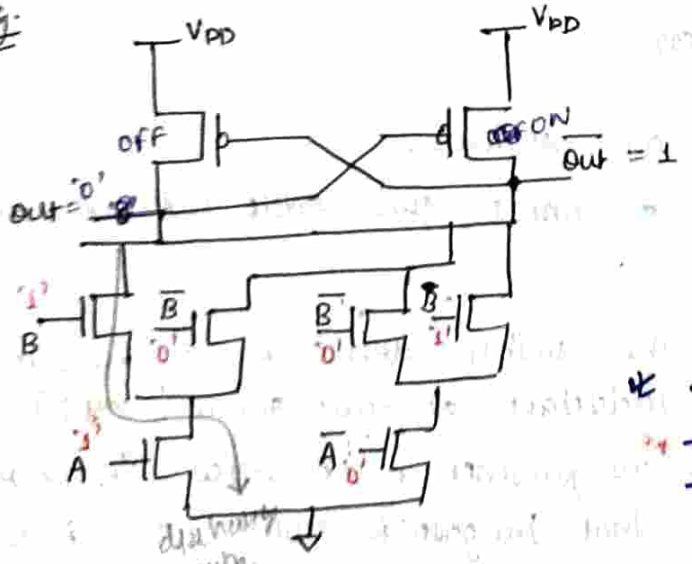
(v) DCVSL : "Differential cascode Voltage switching logic"

- Using this logic style we completely eliminates static currents & provides rail-to-rail swing. Such a gate combines
  - differential logic
  - +ve f/b → it ensures that load device is turned off when not needed. (Static power dissipation ↓)



- The pull down N/w PDN1 & PDN2 are NMOS devices & are mutually exclusive
  - when PDN1 conducts PDN2 is off & vice versa.

Eg:



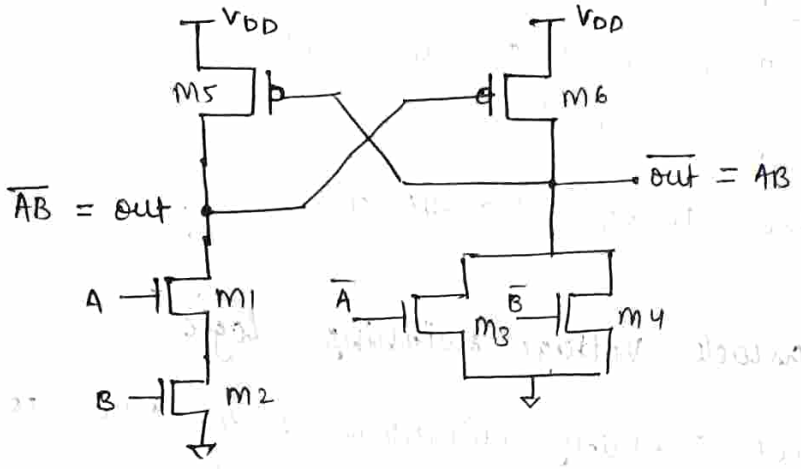
(?). we will size the PMOS accordingly to get the best rise delay = fall delay.

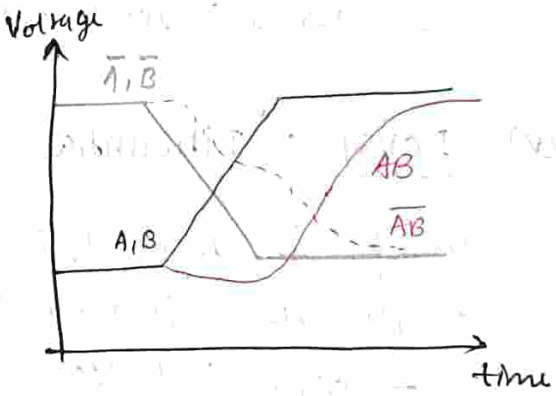
- this i/p combination is

* sizing will decided by considering the
  - Area (W)
  - Rise time = fall time

B XOR-XNOR gate

Example : AND & NAND gate





- A = 1, B = 1, out = 0, $\overline{out}$ = 1,
  ↓ ↗ ↓
  M6 → ON, M5 → off

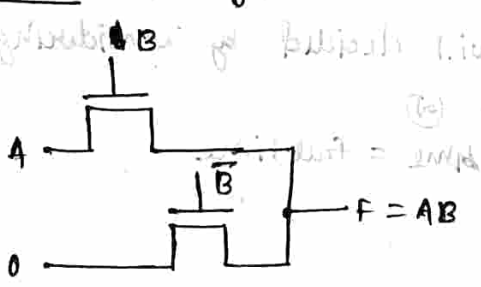
- A = 0, B = 1, $\overline{out}$ = 0, out = 1
  ↓ ↗ ↓
  M5 → ON, M6 → off

Pass Transistor Logic : (Rabaey PG. 259)

- Alternative to complementary cmos is PTL logic, which uses less number of transistors by allowing the primary ilps to drive gate, source, drain terminals.
  - PTL will only NMOS and no PMOS
  - No static power consumption because of no PMOS no direct V_{DD} to GND connection

- In PTL, NMOSs are used as switches.
  - (why?) You make me switch n/w of nmos then add buffer to it.

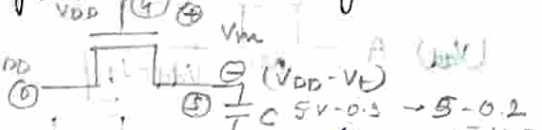
Example : AND gate



- The switch driven by $\overline{B}$ seems to be reluctant at first glance. (why?)
  - ** Its presence is essential to ensure that the gate is static, that is that a low impedance path exists to supply rails in all cases & specially when B is low.

* Now we can implement AB (AND gate) with only 4 transistors (including the inverter to input B). ~~using~~ CMOS require 6 transistor to implement AND gate.

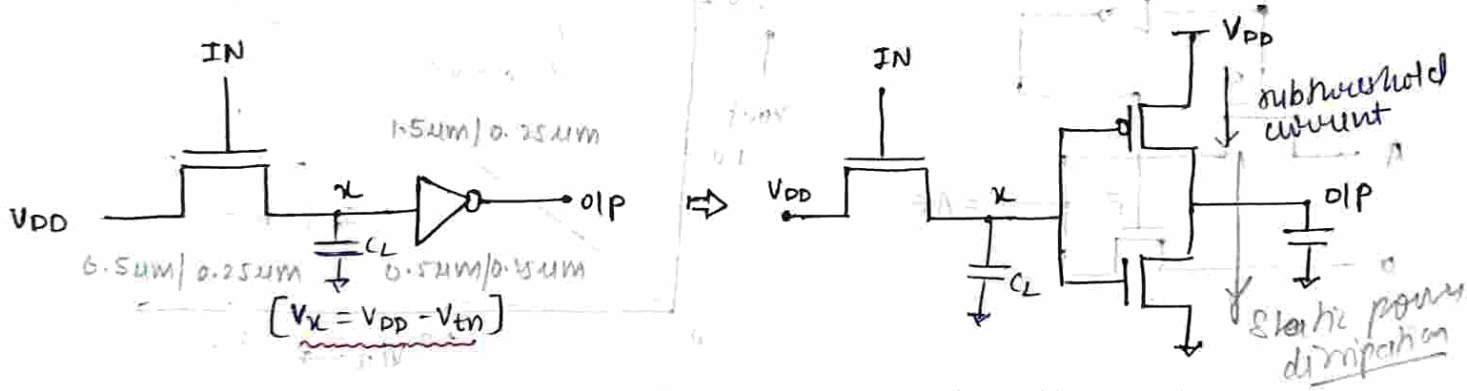
Note: An NMOS is effective at passing '0' but poor at a node to V_{DD}. When the pass transistor pull a node high, the o/p only charges upto $V_{DD} - V_{tn}$.



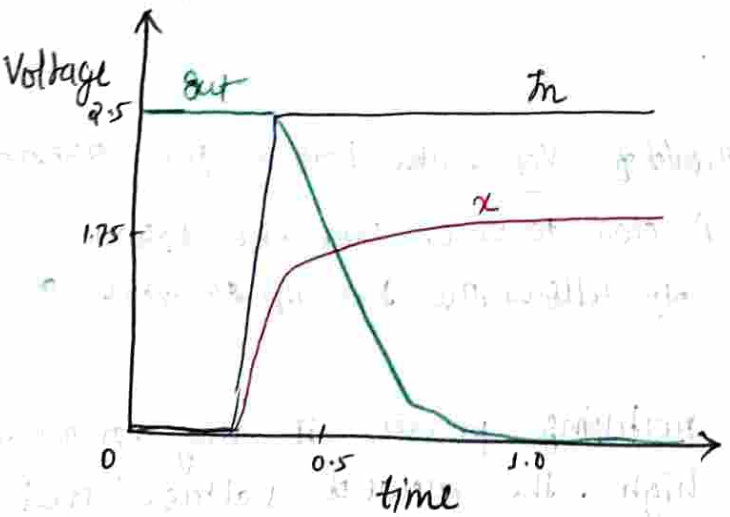
* In PTL you can have AND functionality & NAND functionality also. So it is no longer limited to inverted o/p.

* To improve the voltage swing at the o/p of the PTL, we added a inverter so that we can get proper V_{DD}.

- Voltage swing for PTL



. The drive (gate to source voltage) reduces significantly as the o/p approaches $V_{DD} - V_{tn}$ & the current available to charge up node x reduces drastically.

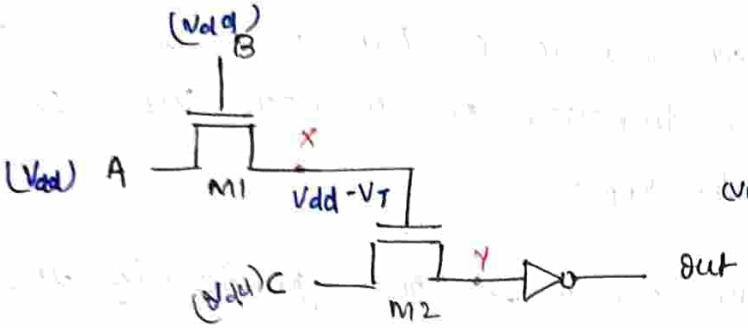


** Now the PMOS is not fully off so the pmos current (subthreshold current) will not be zero. (static current)
- NMOS is completely ON.
  ↳ it is in saturation.

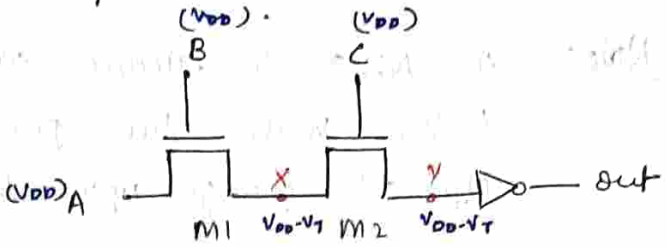
* In PTL there should not be any static current, but here we are observing static current.

* V_x doesn't pulls up to 2.5V but 2.5V - V_{tn}
- Threshold voltage loss causes static power consumption.

- NMOS has higher threshold than PMOS due to body effect.
- PTD/P (Drain/source) terminal should not drive other gate terminal to avoid threshold drops.



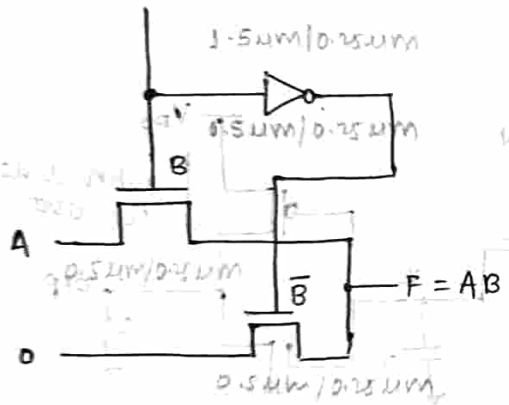
swing on $Y = V_{DD} - V_{tn1} - V_{tn2}$

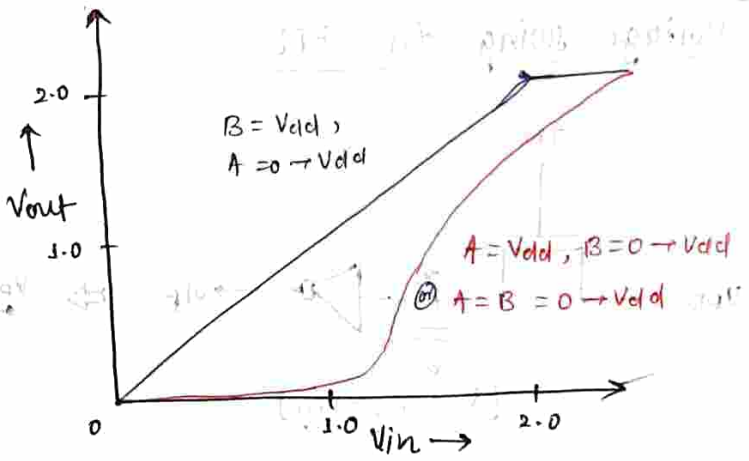


swing on $Y = V_{DD} - V_{tn1}$

doubt

VTC of pass transistor: AND gate





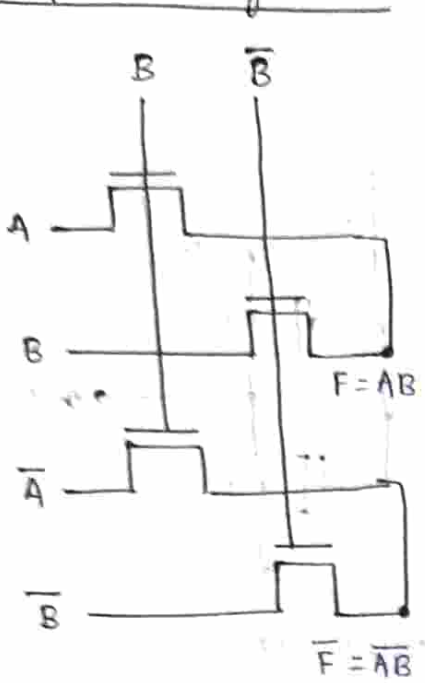
(i) When B = Vdd, top pass transistor → ON
bottom pass transistor → off.

The o/p just follows the i/p A until the i/p is high enough to turn off the top pass transistor (i.e, reaches $V_{DD} - V_{tn}$).

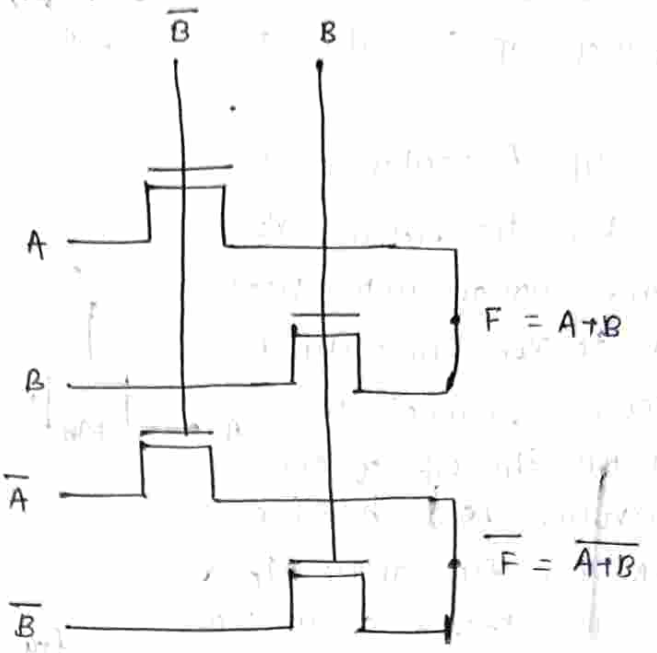
(ii) When A = Vdd, B = 0 → 1.
Since the inverter has the threshold of $\frac{V_{DD}}{2}$, the bottom pass transistor is turned ON till then the o/p is close to zero. Once the bottom pass transistor turns ~~off~~ OFF, the o/p follows the the i/p ~~B~~ minus a threshold drop.

* While the ckt exhibits lower switching power, it may consumes static power when the o/p is high - the reduced voltage level may be insufficient to turn OFF the PMOS transistor of subsequent cross inverter.

Complementary PTL & Differential PTL



AND / NAND



OR / NOR

* CPL (complementary PTL) or DPL (Differential PTL) like single ended PTL suffers from static power dissipation & reduced noise margin, since the high i/p to the signal-restoring inverter only charges upto $V_{DD} - V_{tn}$.

— Robust & Efficient PTL design are given below

Robust & Efficient PTL

- Unfortunately, differential PTL, like single ended PTL suffers from static power dissipation & reduced noise margin, since the high i/p to the signal restoring inverter only charges upto $V_{DD} - V_{Tn}$.

1. NMOS only logic : Level Restoring Transistor

- The common solution to the voltage drop problem is the use of level restorer, which is a single PMOS configured in a feedback path.

↳ it solves the problem of
— static power dissipation (it removes it)
— it improves the Rail to Rail swing (i.e. noise margin)

- Assume that node X is at 0V (out is at V_{DD} and the M_r is turned OFF) with, $B = V_{DD}$ & $A = 0$.

(i) If i/p A makes a 0 to V_{DD} transition, Mn only charges upto node X to $V_{DD} - V_{tn}$. This is however, enough to switch the o/p of the inverter low, turning ON the f/B device M_r & pulling node X all the way to V_{DD}.

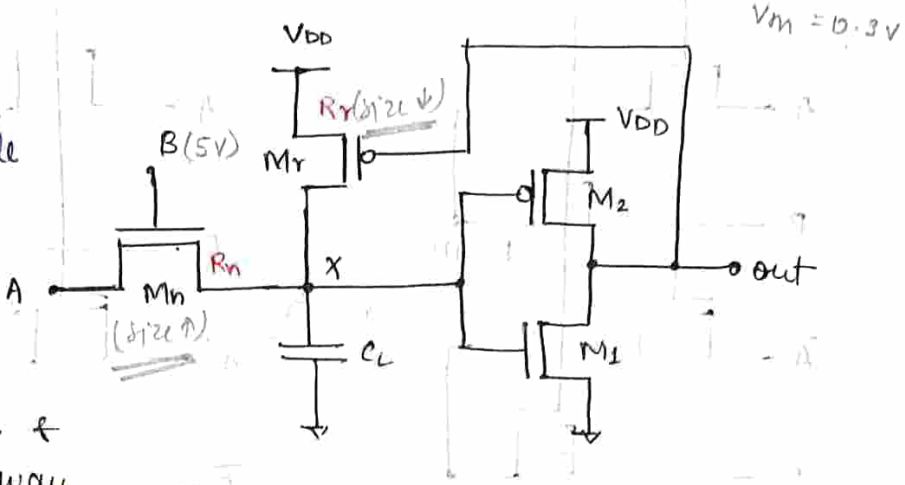


fig: Level restoring ckt

— This eliminates any static power dissipation in the inverter. Furthermore, no static current path can exist through the level restorer & pass-transistor, since all the restorers are active when A is high.

Problem with Restorer →

(ii) (A) The problem arises during the transition of node X from high to low. The pass transistor N/W attempts to pull down node X, while the level restorer pulls now X to V_{DD}. Therefore, the pull-down device must be stronger than the pull-up device in order to switch node X and the output.
(from 1 to 0) (so more resistance)

— Soln We have to take the size of transistors in such a way that the voltage at node X drops below the threshold of the inverter ; $V_M = f(R_1, R_2)$;

* ↳ This condn is sufficient to guarantee a switching of the o/p voltage V_{out} to V_{DD} & a turning off the level-restoring transistor.

⇒ size ↑, R ↓, V ↓, etc

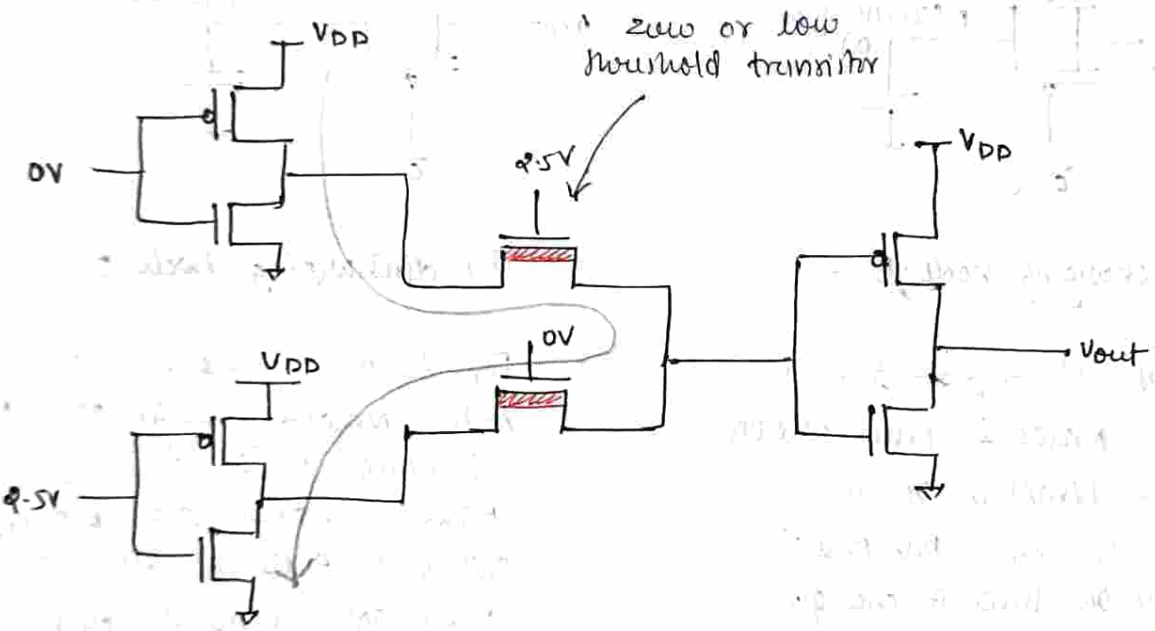
(Mr) decreasing the size of M_r, R_r ↑, hence V(C_L) ↓.

* Imp. point is that sizing of M_r is critical for DC functionality, not just performance.

(Mn) Size of M_n should be high.

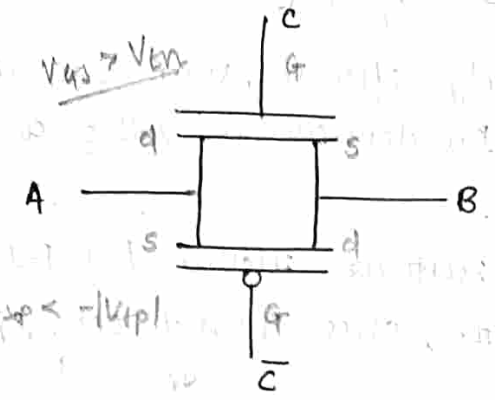
- Another problem is on adding restoring device increase in capacitance at the internal node x, slowing the gate down.
  - The rise time of the gate is largely affected, since the level restoring transistor M_r fights the decrease in voltage at node x before being switched off.
  - On the other hand, the level restorer reduces the fall time, since the PMOS transistor, once turned ON, speeds the pull up action.

(ii) Multiple threshold transistor

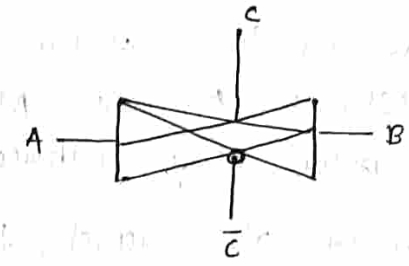


- Using "zero threshold" device for the NMOS pass-transistor eliminates most of threshold drop & passes a signal close to V_{DD}.
- The use of zero threshold transistor, can be dangerous due to subthreshold current that can flow through the pass transistor, even if V_{GS} is slightly below V_T.
  - While sub leakage paths are not critical when the device is switching, they do pose a significant energy-overhead when the ckt is in the idle state.

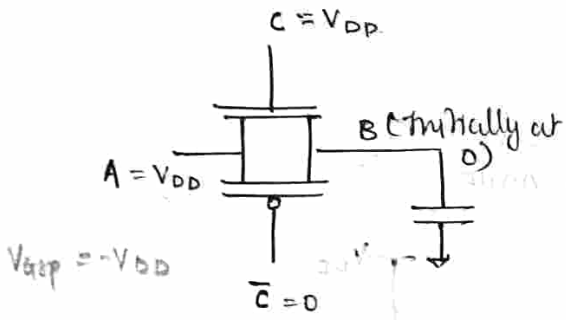
(iii) Transmission Gate Logic



"circuit"

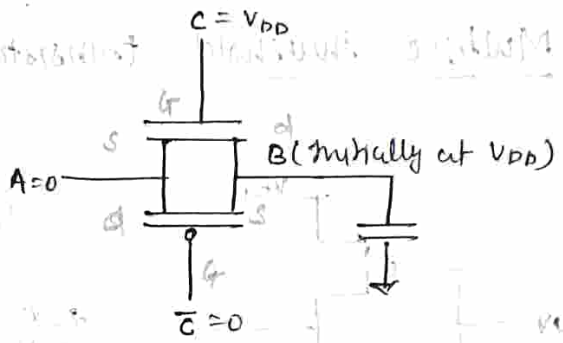


"symbol"



(a) charging node B

- For A = 1 & C = 1 & $\bar{C}$ = 0 both NMOS & PMOS are ON & B = 1. NMOS is ON still $V_B = V_{DD} - V_{tn}$ but PMOS is still ON hence B charges to V_{DD}.



(b) discharging node B

- For A = 0 & C = 1 & $\bar{C}$ = 0 both NMOS & PMOS are ON B discharges to '0'. Initially PMOS pulls down the B to V_{tp} after that PMOS gets off but NMOS is still ON & hence it discharges the B to zero.

#

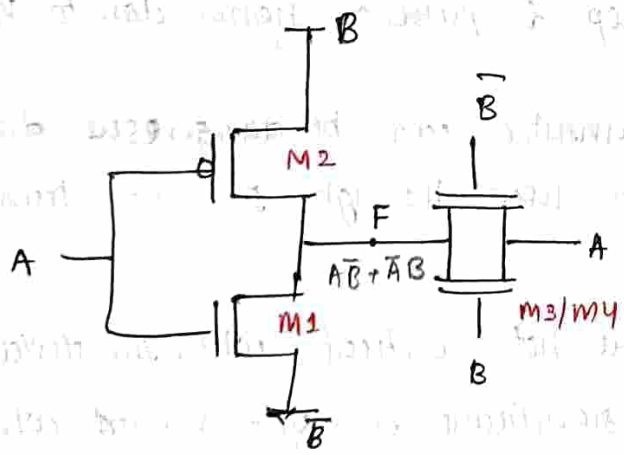


fig. Transmission gate XOR

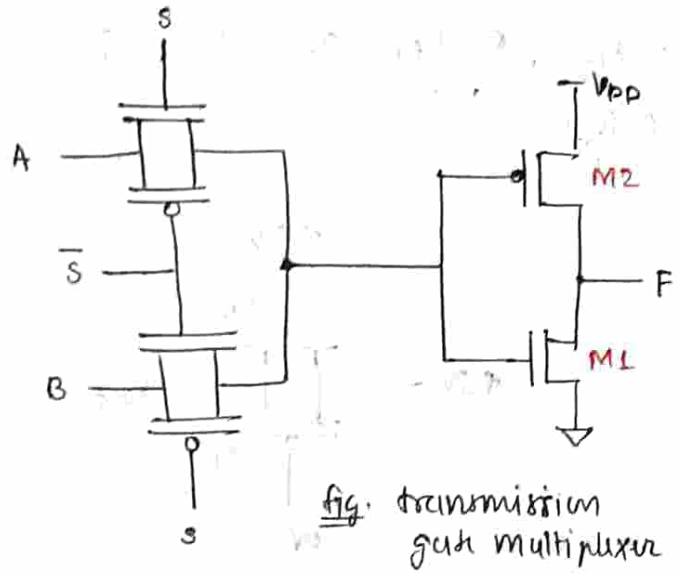


fig. transmission gate multiplexer

Delay in Transmission gate

- There is a series resistance associated with the transmission gate.
- The effective resistance of the switch is modelled as a parallel connection of the resistance R_n & R_p of the NMOS & PMOS devices.

— For low-to-high ~~transition~~ transition of o/p, for low value of V_{out}, the NMOS is in saturation & resistance is

$$R_n = \frac{V_{DD} - V_{out}}{I_N} = \frac{V_{DD} - V_{out}}{K_n'\left(\frac{W}{L}\right)_N\left[(V_{DD} - V_{out} - V_{Tn})V_{DSAT} - \frac{V_{DSAT}^2}{2}\right]}$$

$$R_n \approx \frac{V_{DD} - V_{out}}{K_n (V_{DD} - V_{out} - V_{Tn}) \cdot V_{DSAT}}$$

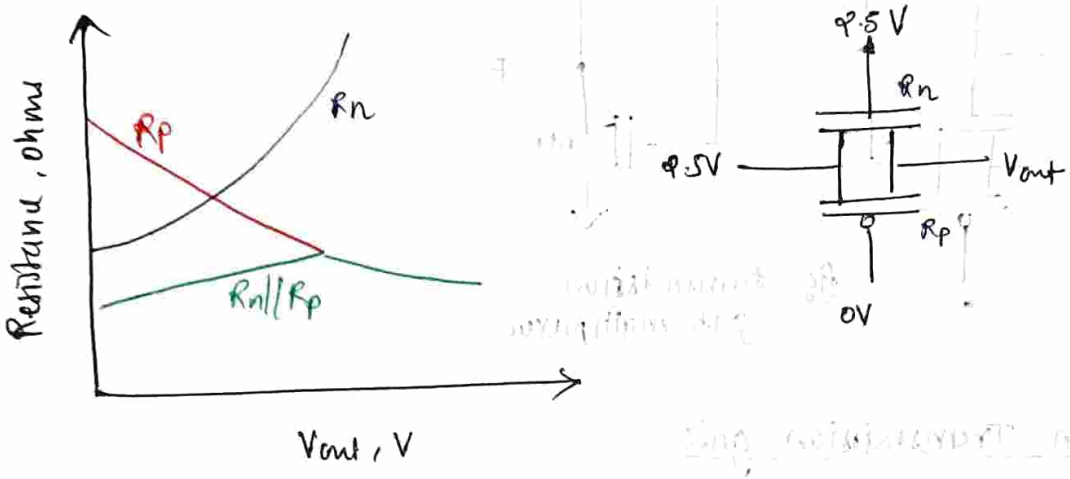
* The resistance goes up as V_{out} increases, R_n approaches ∞ as V_{out} reaches $V_{DD} - V_{Tn}$ then NMOS turn off

— Similarly, for PMOS transistor, when V_{out} is small, the PMOS is saturated but it enters the linear mode of operation ~~for~~ when V_{out} goes to V_{DD} then R_P is

$$R_P = \frac{V_{DD} - V_{out}}{I_P} = \frac{V_{DD} - V_{out}}{K_P\left[(-V_{DD} - V_{TP})(V_{out} - V_{DD}) - \left(\frac{V_{out} - V_{DD}}{2}\right)^2\right]}$$

$$R_P \approx \frac{1}{K_P(V_{DD} - |V_{TP}|)}$$

* The simulated value of $R_{eq} = R_n // R_p$ as func^n of Vout when Vout moves from 0 to 1.

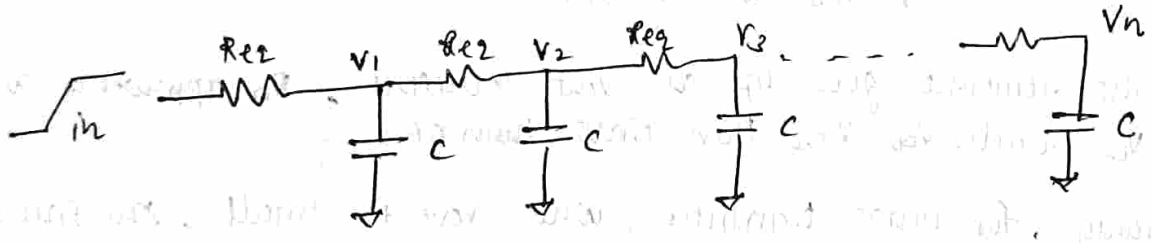


- The N/w of transmission gate are replaced by their equivalent resistance R_{eq}.

  — The delay of a N/w 'n' transmission gate in sequence can be estimated using elmore delay

$$t_p = 0.69 \sum_{K=0}^{n} CR_{eq} \cdot K = 0.69\, C R_{eq} \frac{n(n+1)}{2}$$

* $t_p \propto n^2$ → as no. of switches (ie, n) ↑ prob. delay ↑.



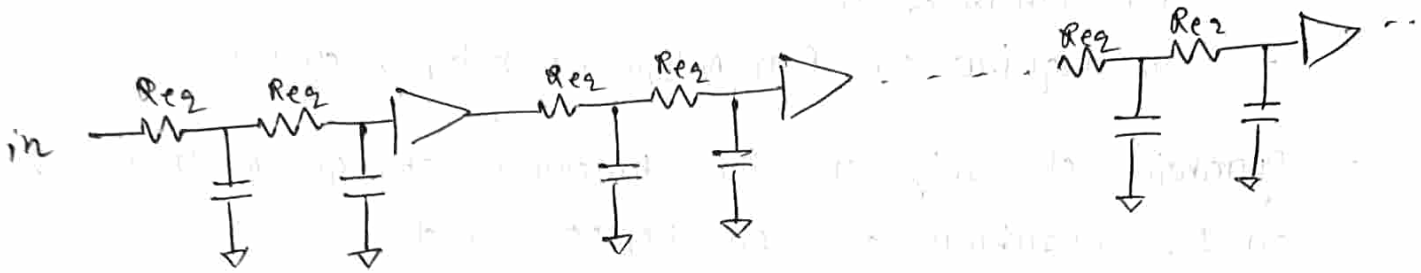
- The most common approaches for dealing with the long delay is to break the chain & by inserting buffer every 'm' switches. then delay becomes

$$t_p = 0.69 \left[\frac{n}{m} CR_{eq} \frac{m(m+1)}{2} \right] + \left(\frac{n}{m} - 1 \right) t_{buf}$$

↓ prop. delay of each buffer.

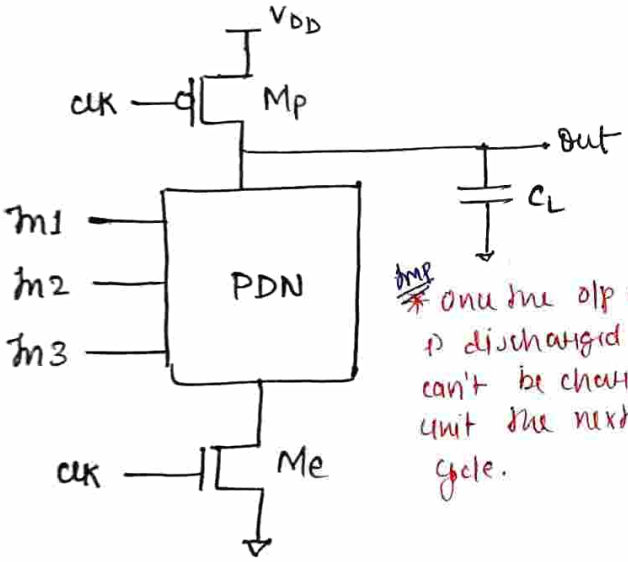
* The optimal no. of switches b/w buffers can be found by setting the derivative $\frac{\partial t_p}{\partial m} = 0$ to 0.

$$\therefore \quad m_{optimal} = 1.7 \sqrt{\frac{t_{p\,buf}}{C R_{eq}}}$$

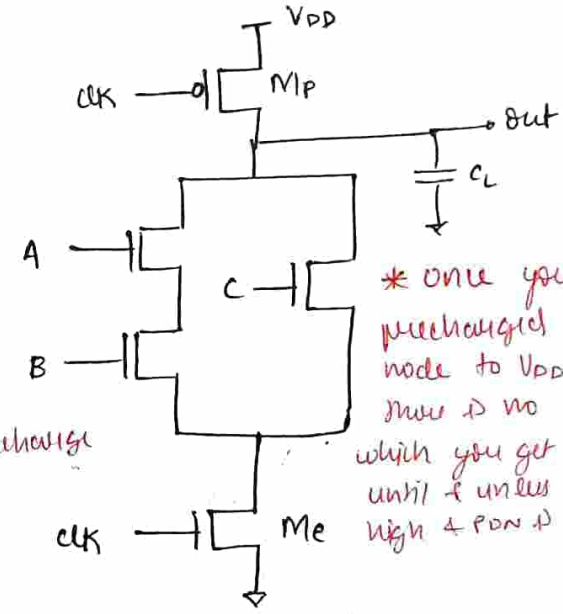


2. Dynamic logic

- In static ckt at every point in time (except when switching) the o/p is connected to either GND or VDD via a low resistance path.
  - For fan-in of n
  - ∴ it requires 2n (n N-type + n P-type) devices
- Dynamic ckt rely on the temporary storage of signal values on the capacitance of high impedance nodes.
  - It requires n+2 (n+1 N-type + 1 P-type) devices.



* once the o/p node is discharged, it can't be charged until the next precharge cycle.



* once you have precharged the o/p node to VDD then there is no way in which you get it discharg until & unless clock is high & PDN is activated.

Two phase operation:

(i) Precharge (clk = 0)

∴ Mp → ON, Me → OFF } Out = '1'

(ii) Evaluate (clk = 1)

∴ Mp → OFF, Me → ON, & $\overline{AB+C} = 0$ } Out = '0' "Discharging"

$$Out = \overline{clk} + (\overline{A \cdot B + C})\,clk$$

$$= \overline{clk} + (\overline{XX\ldots X})\,clk$$

"charging"

↓ Boolean expression

Concept

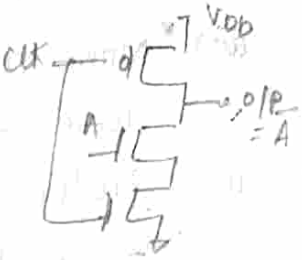
* When During evaluation, when $clk = 1$ but $\overline{AB+C} \neq 0$ (or) $\overline{AB+C} = 1$ } worst case

'worst case'

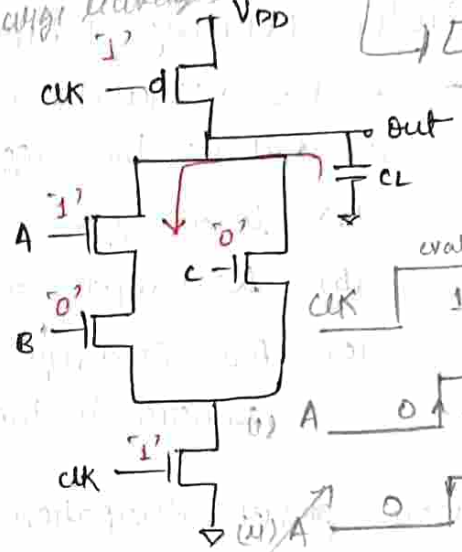
(i) Let $A = 1, B = 0, C = 0$

∴ o/p is kept floating or it will be in high impedance state

∴ Therefore it is called dynamic logic.



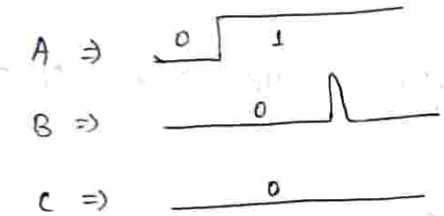
- So there is some charge sharing (charge leakage) happening b/w these ~~two~~ node 'out' & 'A' & we can't recover that charge.

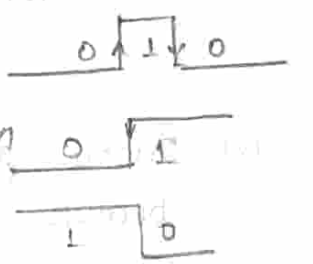


- There is no path to ground

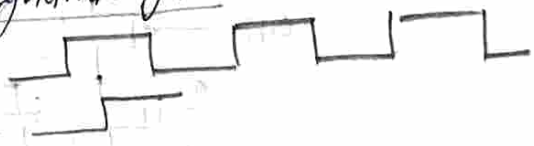
(ii) $A = 1, B = 0, C = 0$

and 'B' has a glitch





- Then out will discharge through this path A,B & clk and there is no way to recover this charge (until my clk goes to '0')
- The glitches are not acceptable in dynamic gates



Conditions on Output

(i) Once the o/p of a dynamic gate is discharged, it can't be charged again until the next precharge operation.

(ii) Input to the gate can make atmost one transition during evaluation. and that must be a rising transition (0 to 1).

(iii) Output can be in high impedance state during & after the evaluation (PDN off), state is stored on C_L.

floating value

- we can use level restorer to recover that charge.

(iv) Output can only go from '1' to '0'.

(precharge) (evaluate)

Properties of Dynamic Gates

(i) Logic fun^n is implemented by PDN only.

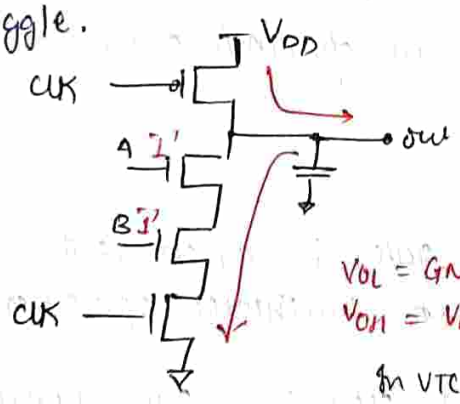
(ii) It has full swing outputs ($V_{OL} = GND$ & $V_{OH} = V_{DD}$)
— we will get full '0' & full '1'. → Because of clocks in dynamic gates

(iii) * Non-ratioed ~~logic~~ – sizing of the devices doesnot affect the logic levels. [But in static CMOS sizing decides the functionality]
↳ I think this is wrong.

(iv) Faster switching speeds
— As we reduces the no. of transistors that an i/p goes & hence we reduces the logical effort & path delay reduces.
(a) load cap. ↓ as i/p (cap. (C_{in})) ↓
(b) load cap ↓ as smaller o/p loading (C_{out})
(c) No short ckt current (I_{sc}), so all the current provided by the PDN goes into discharging C_L.

(v) Overall power dissipation usually higher than static CMOS. because

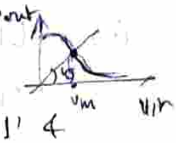
— In static CMOS, the current will flow only when i/p toggle.



— But in Dynamic logic, the current will flow even when i/p doesnot toggle & clk is changing.

* Hence frequent charging & discharging is done due to clk switching ∴ more power dissipation.

switching threshold
$V_M = V_{IH} = V_{IL} = V_{th}$



In VTC, above which the value is read as '1' & below it the value is read as '0'.

(why?)

- No static current path ever exist b/w V_{DD} & GND (∵ No static power loss) including no short ckt current.
- But large dynamic power is there
- No glitching is allowed
- higher transition probabilities
- extra load on clk should be more

Q. Why no static power dissipation?
→ No static power means no charge leakage will occur.
→ because will never be at const. '0' or '1' as o/p cap. may be sharing its charge to the PDN NMOS transistors.

(vi) PDN starts to work as soon as the i/p signal exceeds V_{tn} so, V_M, V_{IH}, V_{IL} equal to V_{tn}.

— low noise margin

(v) Dynamic logic gates required precharge / evaluate clocks.

Q. Can we put clk before the PDN & PDN after clk (i.e clk near the o/p)? What will be the failure?

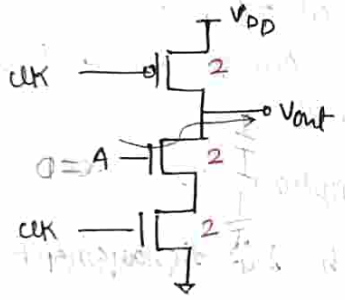
→ It is not usually done, but we can do it

— It as soon as clk toggles we see the noise in o/p.

— As soon as the clk comes the other things have already been evaluated

Q What will be the logical effort of dynamic logic inverter?

→



** In static CMOS, there is no extra load on 'clk', clk only goes to FF

In dynamic logic, clk goes to every combination block.

∴ load on clk is very high

for clock, $\alpha = 1$ (activity factor)

That is also the reason that the dynamic power of dynamic logic system is very high

** In evaluation cycle, the i/p A, B & C will start switching only when the i/p voltage is large than the gate voltage of the individual transistors.

— So you have to wait till the i/p voltage to the PDN crosses the threshold voltage.

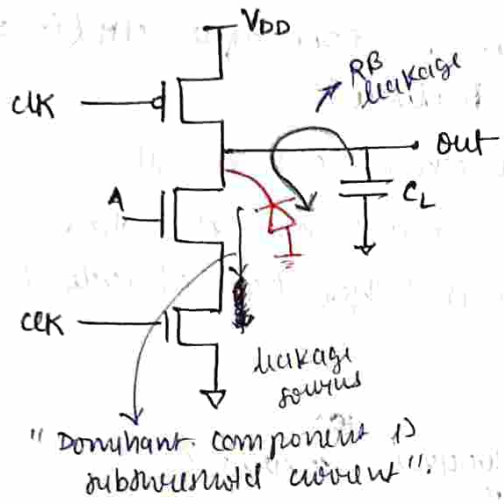
∴ {switching threshold = $V_M = V_{tn}$}

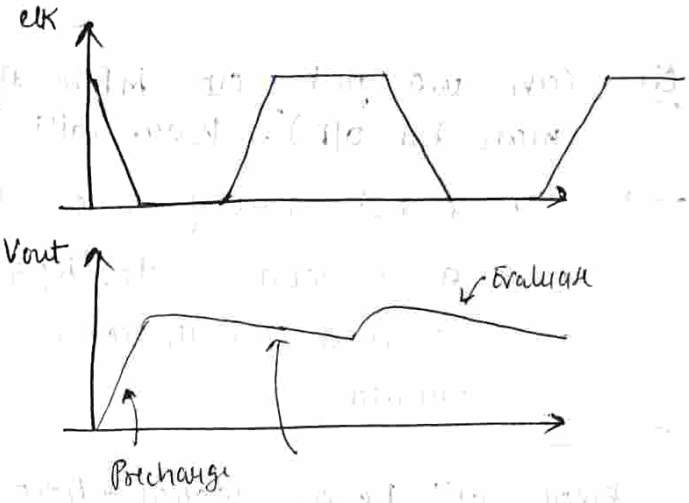
Then NM_L rather than going to GND it goes to V_{tn} (∴ low NM_L) it doesn't go more than V_{tn}

Issues in Dynamic Design

1. Charge Leakage

– Due to floating o/p, there is some charge leakage.





- After precharge, when $V_{out} = V_{DD}$, when clk = 1 & A = 0 there is some charge leakage from 'out' to A. And to restore that charge we are giving the clk = 1' at frequent rate.

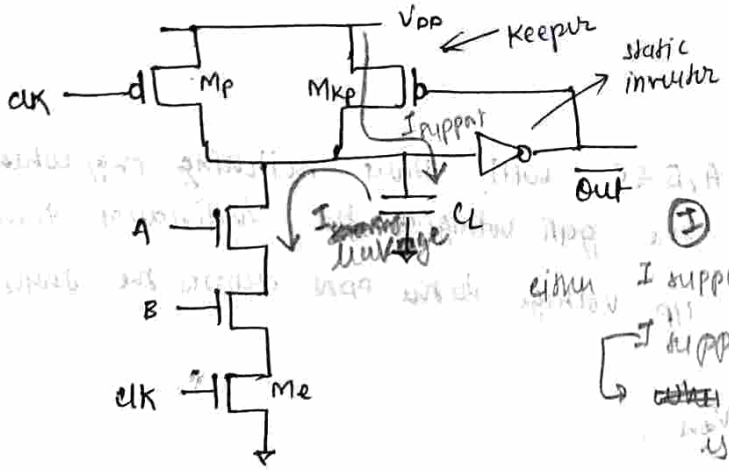
– But this leads to extra power consumption.

Q. Can we also reduce the period for which this discharging is happening?

→ A period of clk in combinational logic is determined by the overall evaluation time of that combinational logic. So you can't change the time period of clk.

(*) ~~evaluation~~ time period of clk will be determined by the critical path.

Solution: Same approach as level restorer for PTL.



(1) either $I_{support} = 10 I_{leakage}$ (empirical)

$I_{support} = I_{leakage}$ (perfect)

↳ the size for which miss-ege matches is the size of keeper.

(II)

- Sizing of keeper gate must be dependent on the (1) size of inverter & the (2) load capacitance at Out.

— Our main work is to get the output from O/p of inverter only.

- why? We have to use very small keeper, it will restore the charge at C_L even if it has very high resistance.

— so that keeper is fully ON & PMOS is in saturation

— saturation current of PMOS V/s off current of NMOS

* In project, we use small sized keeper & where $W \leq 0.135\,\mu m$ &/or $L > 0.6\,\mu m$

- Here keeper gate is a F/B gate, where we can use ratioed logic.

Q. what size should we take of keeper?

→ It depends on of your device

let PVT ⇒ FS 125°C at high voltage

fast NMOS, slow PMOS

since PMOS is slow so the PMOS keeper should still be able to compentate for the charge loss through the NMOS which are fast. Temp. is also very high, so leakage is also high.

Q. Why don't I prefer FF 125C 1.32V over FS 25C 1.32V ?

→ Because hum will also make the PMOS keeper very fast, it will anyway offer more current

* keeper size should be tested at FS, 1.32, 125C F-NMOS & S-PMOS at high voltage & high temperature, the leakage is max. & drive strength of PMOS will also reduce.

Q. • sizing of keeper gate is very important for designing AOI/OAI. How do we decide it?

→ hum ko again point me keeper ka size nikalna hai, jab keeper PMOS jo hai vo slow ho, high temp & high voltage pe operate kar rha ho.

(III)

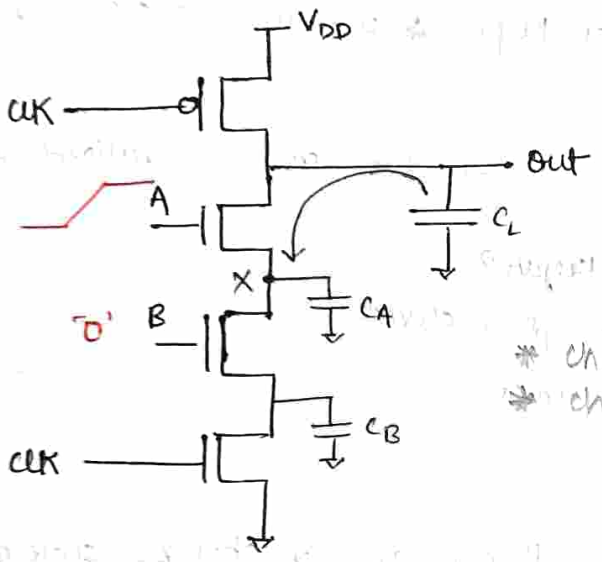
Part-3

* When more are gates cascaded one after the other, or if gate is cascaded after a dynamic gate then that gate has to come after the inverter, it can't come before the inverter.
(To avoid backgate coupling).

* There is no time period for signal (i/p) A & B.
- There is timing constraint b/w clock, A & B i.e. A & B should be settled before the clock arrives (similar to setup)

2. Charge Sharing

- Initially both A=B=0 (precharging)
- When i/p A goes from '0' to '1' & B=0 then charge gets redistributed from C_L to C_A leading to reduced robustness. (C_L decreases, o/p value)



* charge leakage when A=0 (OFF)
* charge sharing when A=1 (ON)

(i) If $\Delta V_{out} < V_{tn}$ → Final value of V_x equals $V_{DD} - V_{Tn}(V_x)$

From charge conservation,

$$C_L V_{DD} = C_L V_{out}(t) + C_A [V_{DD} - V_{Tn}(V_x)]$$

(or)

$$C_L V_{DD} = C_L V_{out}(t) + C_A V_{DD} - C_A V_{Tn}(V_x)$$

$$-[C_A (V_{DD} - V_{Tn}(V_x))] = C_L (V_{out}(t) - V_{DD})$$

$$V_{out}(t) - V_{DD} = -\frac{C_A}{C_L} [V_{DD} - V_{Tn}(V_x)]$$

where,

$$\Delta V_{out} = V_{out}(t) - V_{DD}$$

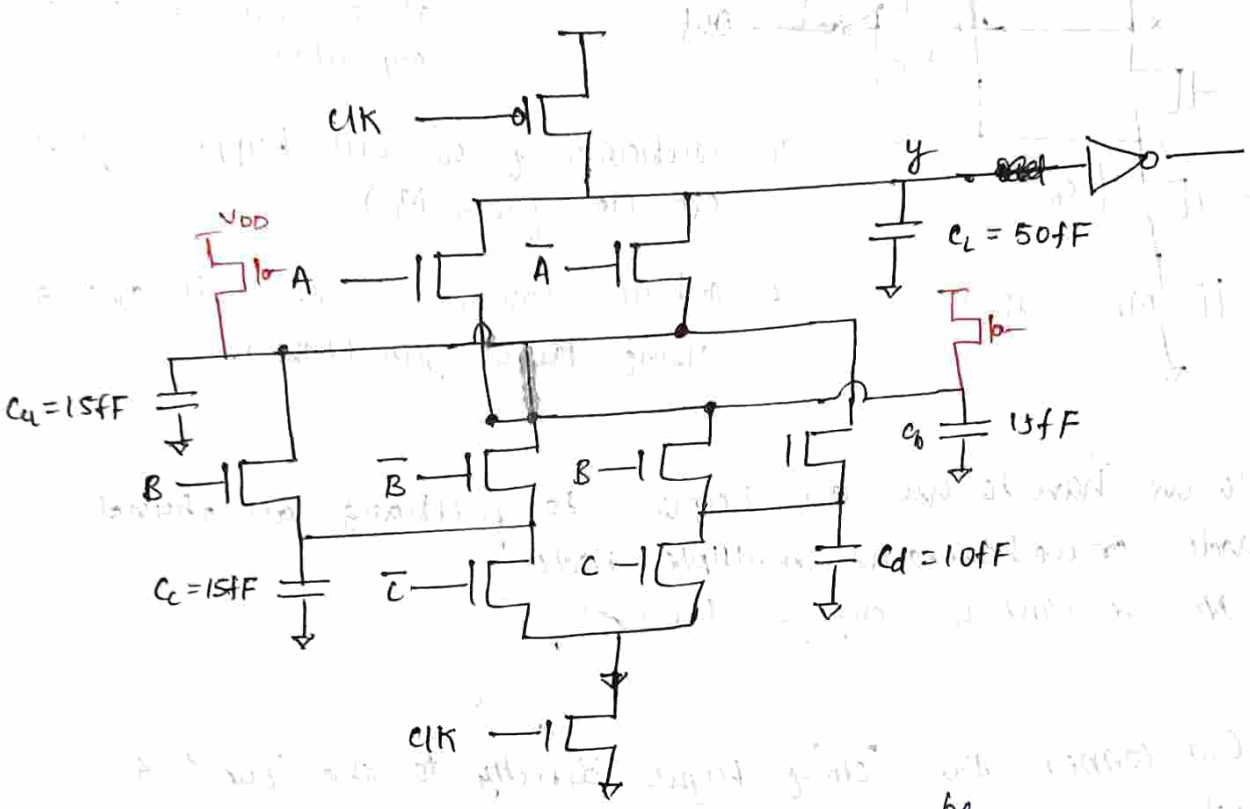
$$\Delta V_{out} = -\frac{C_A}{C_L} [V_{DD} - V_{Tn}(V_x)]$$

& $V_X = V_{DD} - V_{tn}$ =

(ii) If $\Delta V_{out} > V_{tn}$ → V_{out} becomes equal to V_x

$$\therefore \quad \Delta V_{out} = -V_{DD}\left(\frac{C_A}{C_A + C_L}\right)$$

Example : Charge sharing example



- Here the total charge from C_L will be distributed b/w all the capacitances C_a, C_b, C_c, C_d & C_L such that the total voltage across the output capacitance becomes nearly $\frac{V_{DD}}{2}$.

Q. What can be done to solve this charge sharing problem?

→ We can precharge all the capacitance (i.e C_a, C_b, C_c, C_d) using the keeper gate.

- We will not put the inverters at all the keepers but we only put inverter near the O/p.

Solution: * Precharge all the internal nodes using a clock-driven transistor (at the cost of increased area & power).

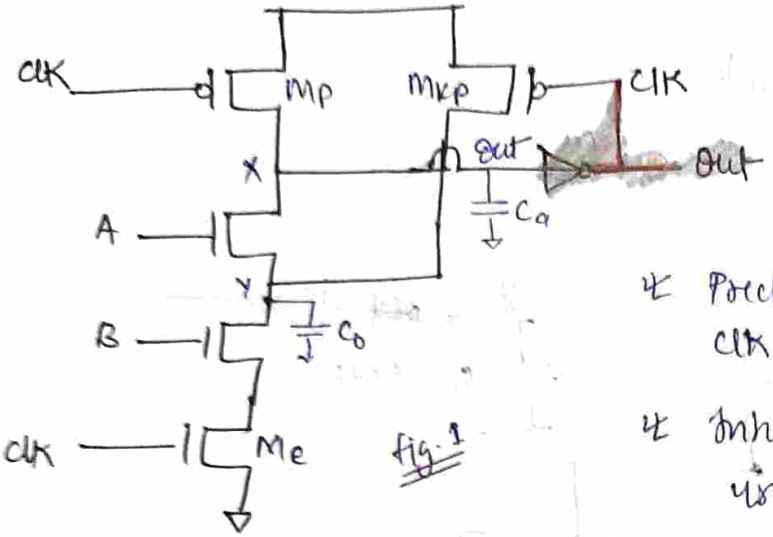


fig. 1

(i) At clk = 0, node X = '1'

(ii) If node X = 1, ⇒ Out = 0

∴ Mkp = ON hence it will also charge the internal capacitance

* Precharge of 'C_a' will happen from clk (ie, from Mp)

* Internal capacitance 'C_b' will charge using keeper gate (Mkp).

Q. Do we have to use 1-keeper to precharge all internal node or we have to use multiple nodes?

→ No we can't use only one keeper

Q. Can connect the 'clk' of keeper directly to the 'out' & when we can't. Which is better way?

→

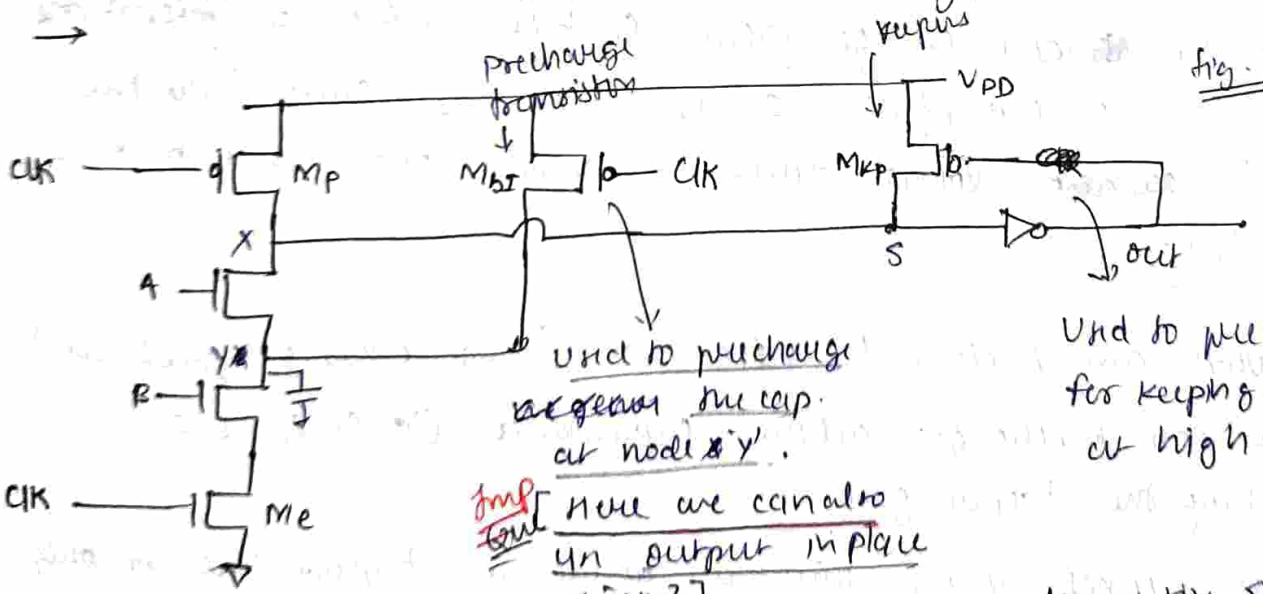


fig. 2

Used to precharge the cap. at node 'Y'.

[Here we can also use output in place of 'CLK']

Used to precharge or for keeping the node 'S' at high.

keeper is < Precharge transistors

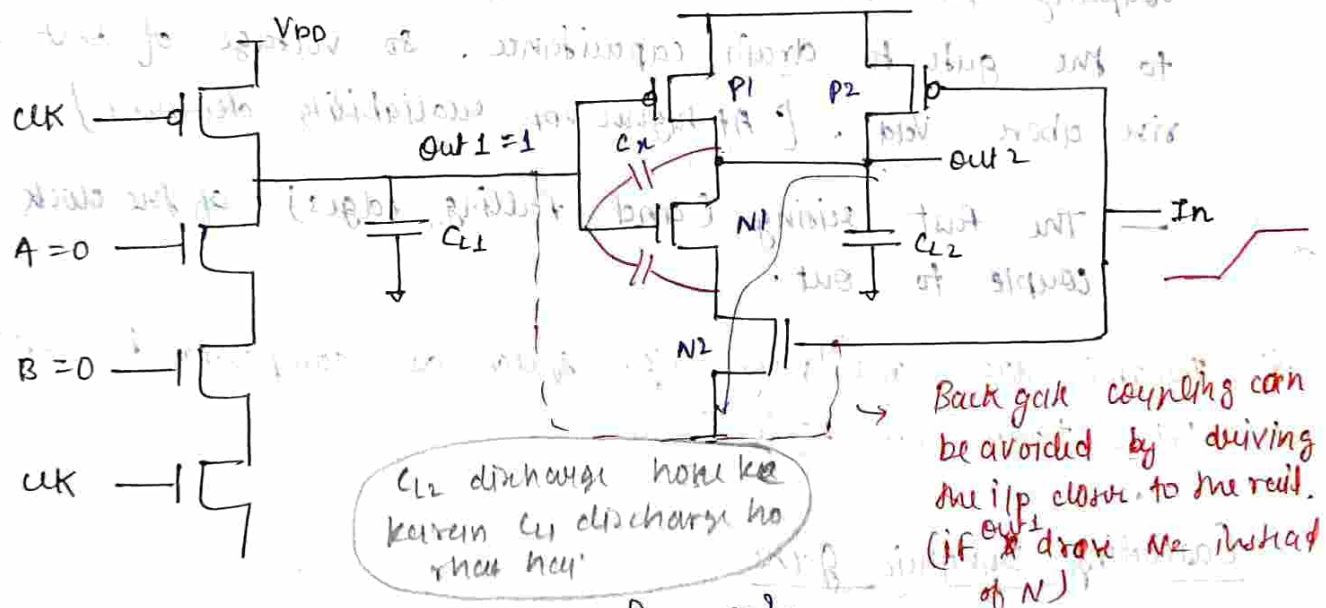
but we don't use this because at A = 0, B = 1 a direct path from V_{DD} to ground is made.

Q. Imp Why we keep 'clk' on the bottom of PDN?

→ As we know, slower signal must be closer to the o/p. In evaluation phase, the o/p will discharge when A & B will go to '1'. So what is coming after 'clk' has gone to 1, → A & B. Therefore 'clk' needs to be in bottom.

3. Backgate Coupling → To avoid this we always add static cmos at the o/p of dynamic cmos.

- You could couple static gates directly at the o/p of the dynamic gates w/o the inverter because static gate can take both kind of transitions (0 → 1) & (1 → 0) (so what kind of transition that dynamic gate can take?)
- But when you couple Dynamic gate right after the out1 then backgate coupling occurs.



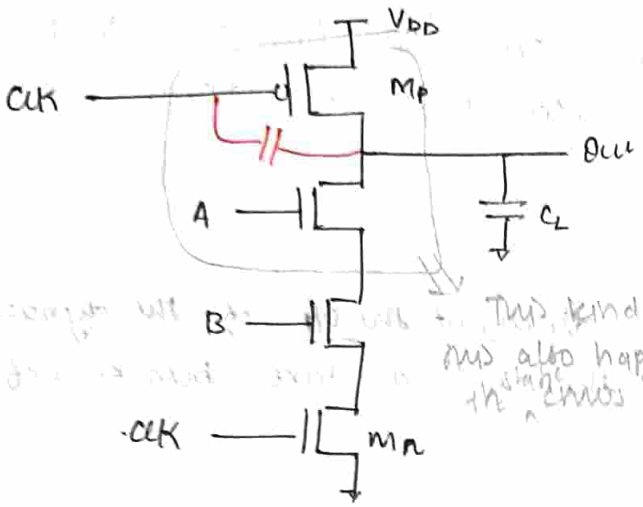
Back gate coupling can be avoided by driving the i/p closer to the rail. (if out1 drive N2 instead of N)

- In = 0 → 1 & Out1 = 1; [P1 off] &
  ⇓
  then C_{L2} at out2 will discharge through N1 & N2 [P2 ON]
- When out2 discharge, due to miller coupling out1 will also see some discharge happening called Back Gate Coupling. (why?)

* out2 becomes 0, ~~then Cx act as~~ and coupling capacitance is b/w out2 & out1. Therefore the ~~miller capacitance~~ out1 also discharges from the miller capacitance (C_M) path.

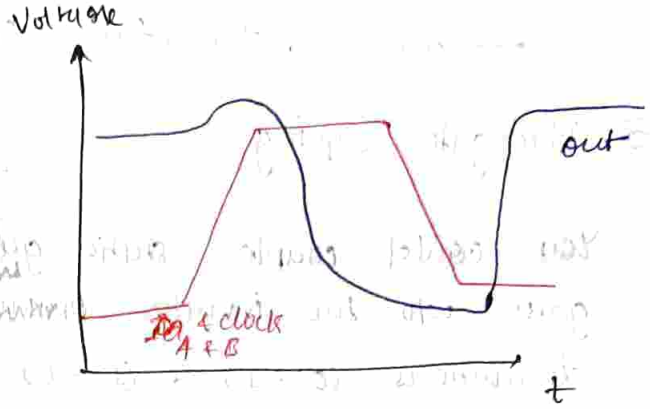
4. Clock Feedthrough

→ This can't be avoided.



This kind of thing also happens in the CMOS inverter.

- when clock comes there some feedthrough b/w out & clk.

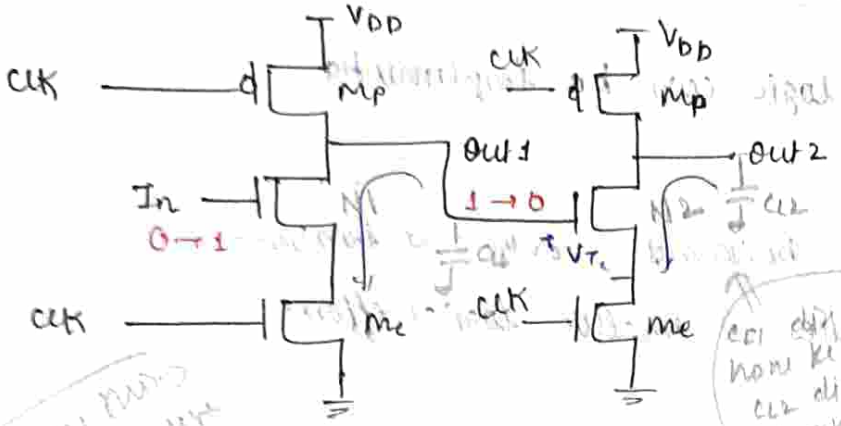


- coupling b/w out & clk i/p of the precharge device due to the gate to drain capacitance. So voltage of Out can rise above Vdd. [At higher VDD reliability decreases]
  - The fast rising (and falling edges) of the clock couple to out.

* Dynamic logic are very high speed as compared to static CMOS, PTL, transmission logic.

Cascading Dynamic Gate (Rabaey 6.3.4 Pg- 283)

- During the precharge phase (i.e. CLK=0) the o/p of both inverters are 1 (i.e. VDD).
- Assume that the primary I/P "In" makes a 0 → 1 transition.

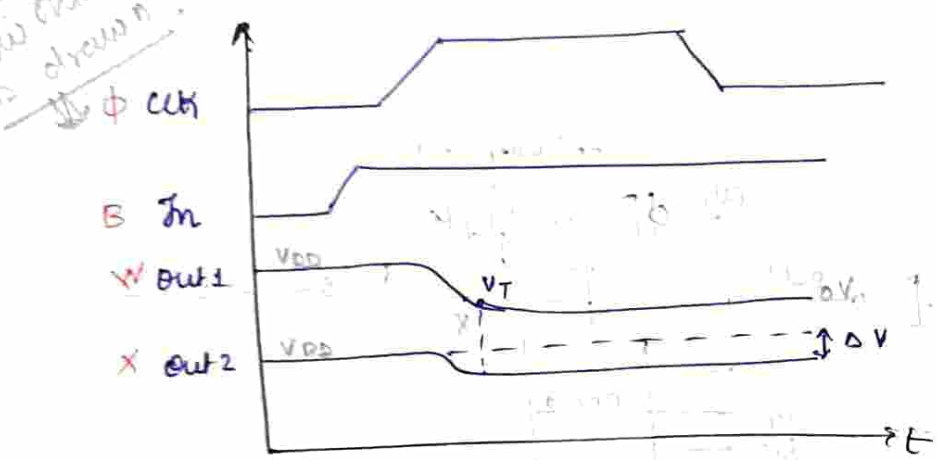


(i) clk = 0, out 1 = out 2 = 1

(ii) In : 0 → 1, & clk = 1

∴ out 1 discharges & out 2 also discharges

(iii) out 2 discharges till out 1 = V_T (this should not happen) (i.e, N2 is ON) and out 1 discharges to 0.

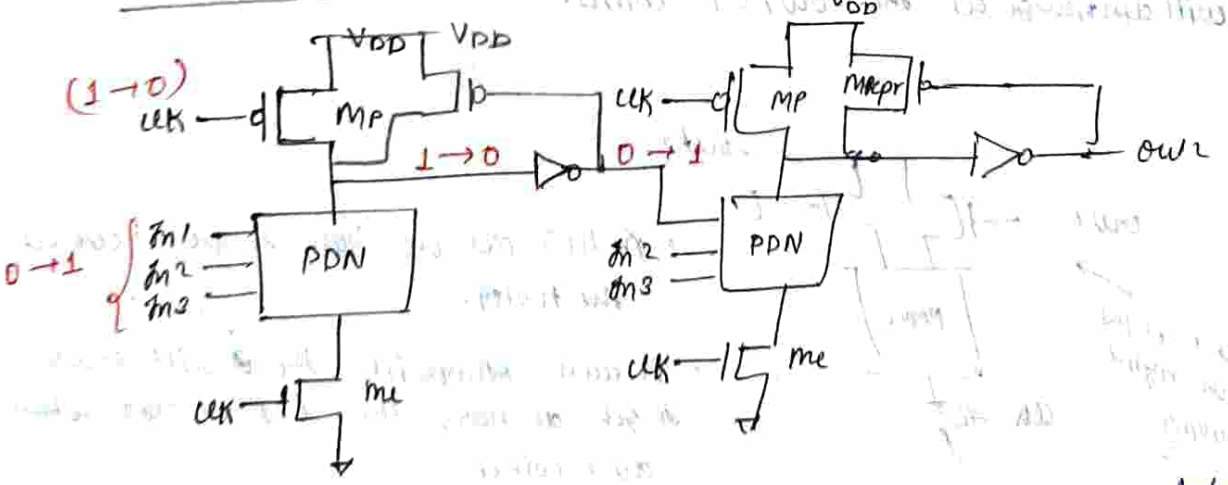


* Cascading problem arises because the o/p of each gate (i.e input of next stage are precharged to 1)

∴ Therefore we use other ways to cascade dynamic gates.

① Domino logic :

- At i/p of dynamic logic : 0 → 1 transition is safe but 1 → 0 transition is not safe. (why?)



* This is called domino because ckt's evaluates one after the other (stage by stage).

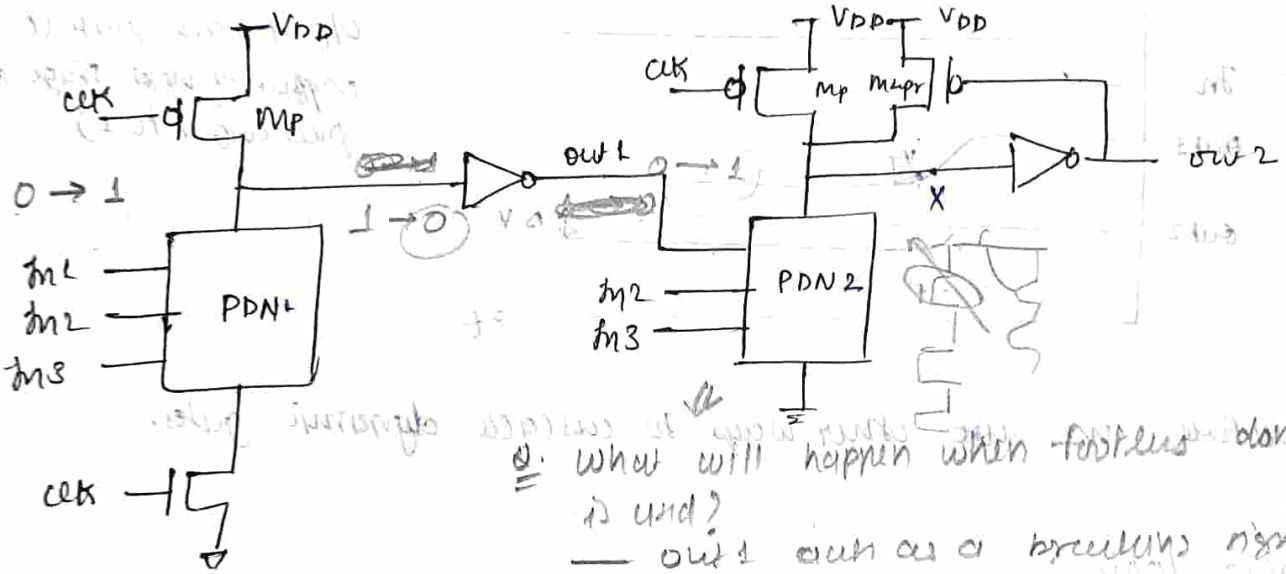
Properties of Domino logic :

(i) Only non-inverting logic can be implemented

(ii) Very high speed

- static inputs can be skewed, only 0→1 transition
- Hp capacitance reduces → smaller logical effort.

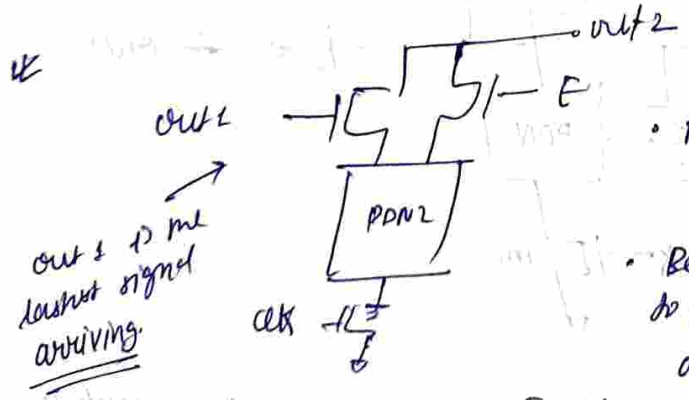
(1.2) Footless Domino



Q. What will happen when footless domino is used?

— out 1 can act as a precharge signal to Next ckt.

- As out 1 = 1 then only the node X will be discharged through PDN2 (which doesn't require any clk), X will discharge as the out 1 = 1 comes.



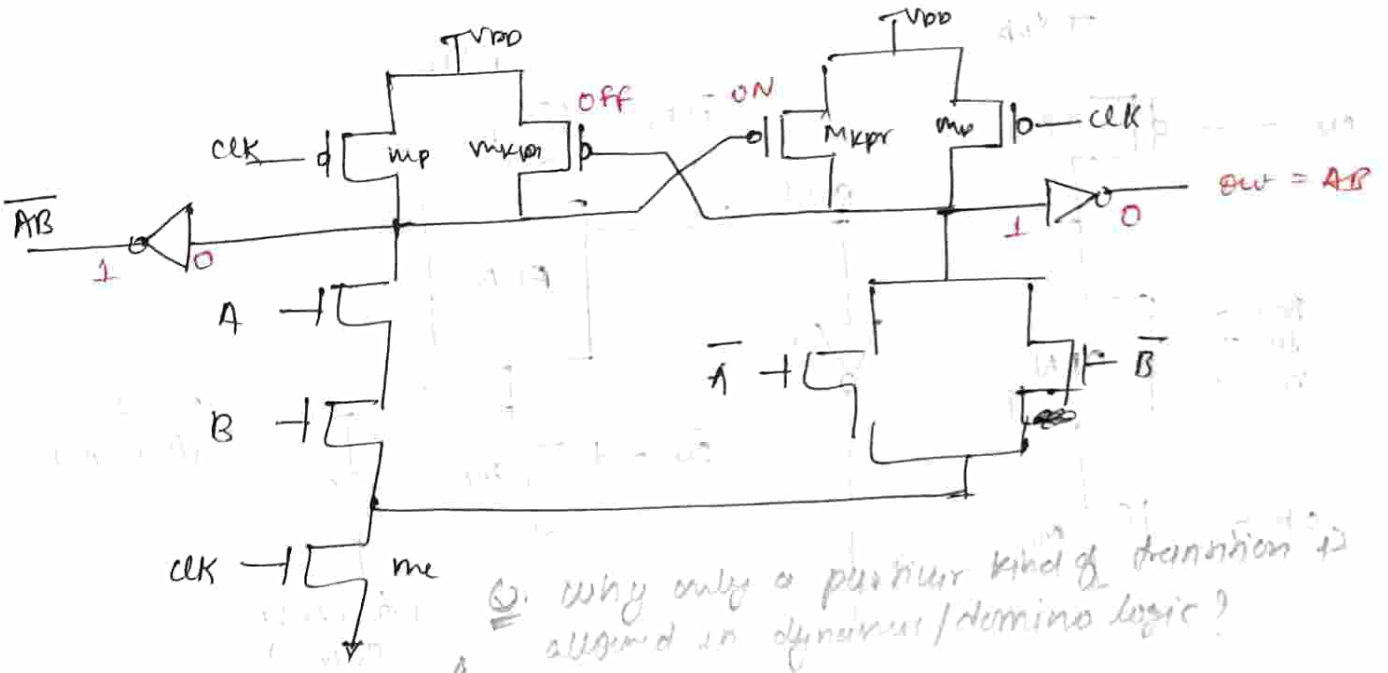
out 1 is the latest signal arriving.

- In hp can we have to use clk at the footer.
- Because otherwise out 2 will always to get discharge the E part also. when out 1 comes.

Q. Problem with footless domino?

— only footless NMOS is allowed for parallel NMOS connection it will not work.

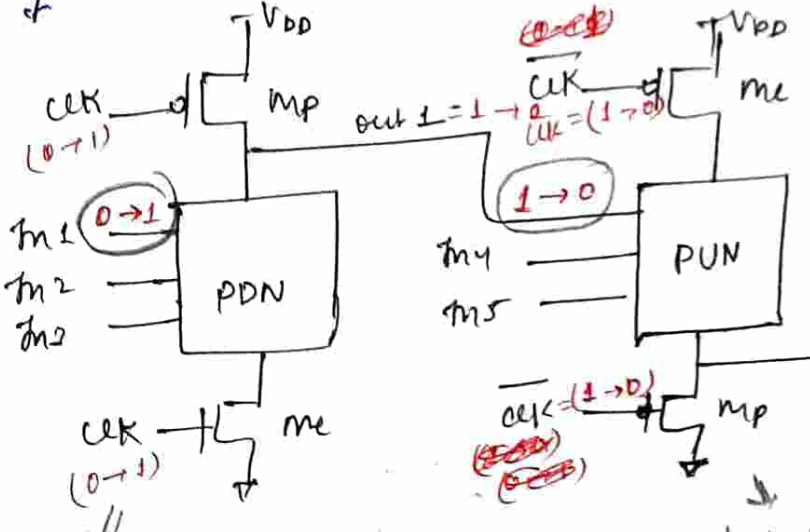
② Differential (Dual Rail) Domino



Q. Why only a particular kind of transition is allowed in dynamic / domino logic?

③ np - CMOS

- When PDN is driving the PDN then we can only allow one 1 transition (why!) and to get such transition one ~~gate~~ is allowed at the out 1 inverter in b/w the cascading.
- But when PUN is ~~also allowed then~~ connected then 1 to 0 transition are only allowed, so you don't need such inverter in b/w the dynamic gate.

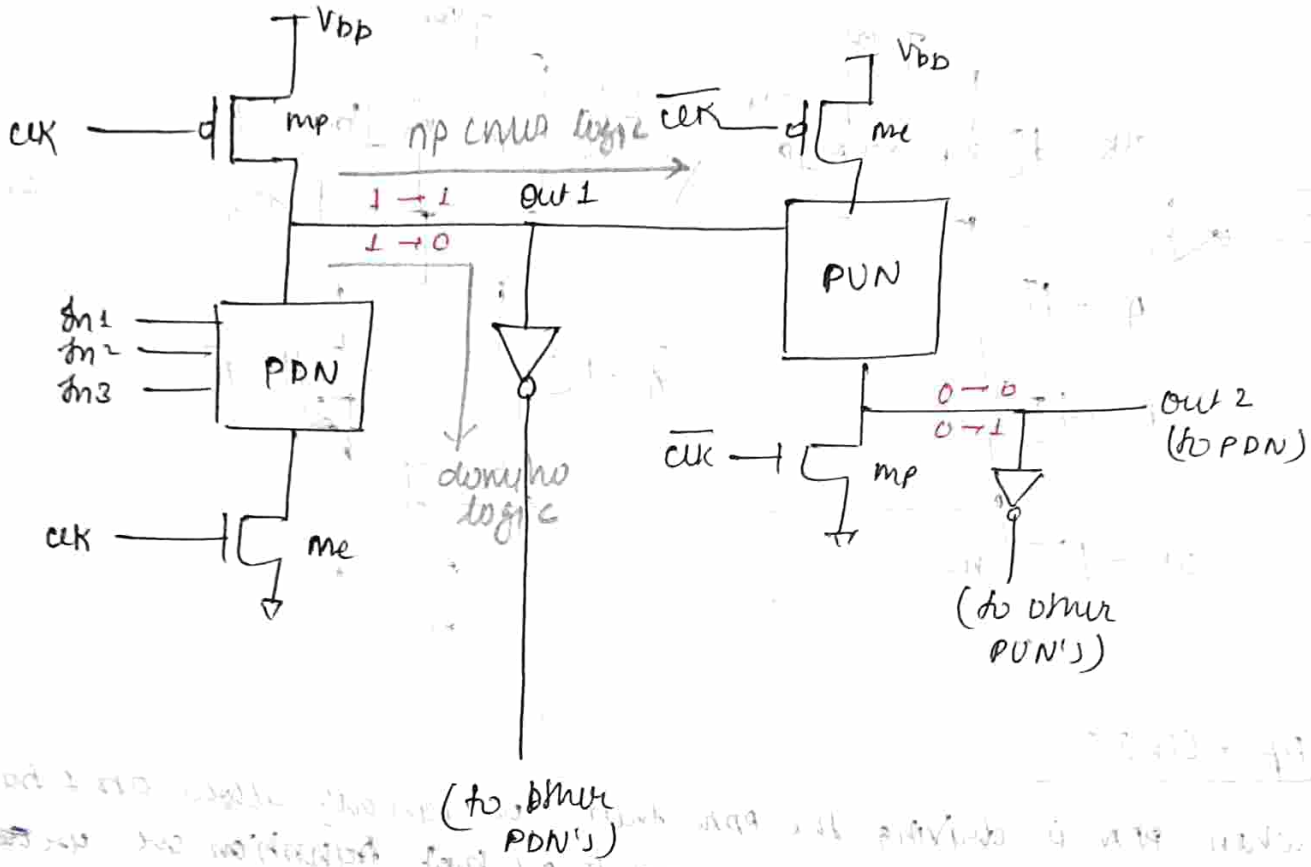


Why?
- Only 0 → 1 transition allowed at I/Ps of PDN
- Only 1 → 0 transition allowed at I/Ps of PUN.

Q. What will happen when In1 transition from 1 → 0?

Q. What will happen when out 1 transition from 0 → 1.

(4) NORA logic:



- Using PUN increases area but it also decreases delay caused by the inverter b/w the dynamic logics

Sequential Circuit

Combinational ckt → Output only depends on current input.

Sequential logic → Output depends on current i/p + past o/p. (or past i/p)

- As we need something to hold our previous state, present state, or future state, for that we use state or tokens.

Eg: FSM, pipelines.

* In FSM we use sequential elements to hold the state.
  In pipelines we use sequential elements.

Q: What does exactly token means?

→ Token is something eg when state of signal is zero then only I will proceed forward.

* Only when my previous set of results are there then only I will start my second operation.
  - In Asynchronous ckt → we call it tokens
  - In synchronous ckt → we call it states
    - ↳ where in FFs when clock comes it says okay sir go to next stage.

* It can have a special o/p (Token) that your ckt would give, which says that okay now I am ready for the next i/p.

Q. What is meant by pipeline?

→ Basically to make the o/p of ckt faster, we separate two blocks of ckt & put flip-flops in b/w so that the o/p of previous one is not able to interact with the i/p of the next stage until I send the clock signal.

Q. So even if the delays of all the transitions of i/p for both the ckt for stage 1 & stage 2 are similar, even then to make the ckt faster, ~~should~~ do we put FF or not.

→ Yes, even then also we put the FFs.
because the functionality will not be affected even after removing the FF but by adding FF it makes the ckt faster. That's why we use it.

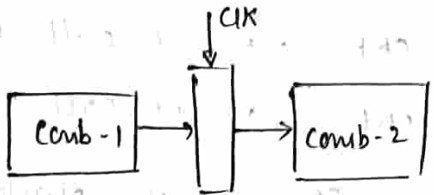
Q. There are two purpose of using this sequencing ckt

→ 2 marks
(i) To ensure sequencing, ie. when all the i/ps have come then only the o/p should be calculated.
→ earlier i/p should not get mixed with the present i/p.
(ii) To improve overall system throughput.
→ To make the system faster.

* We need to keep this sequencing elements to play safe for worst cases.

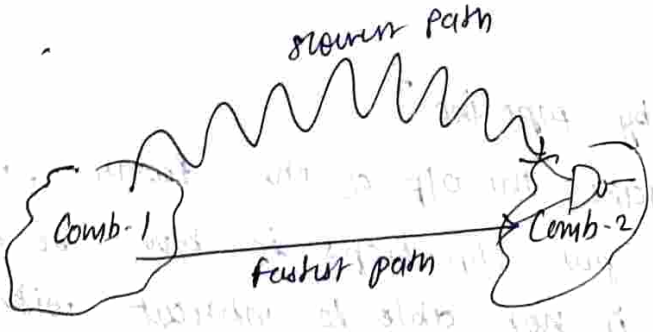
Q. Why we do pipelining?
→ (1) To make ckt faster
(2) To improve throughput
(3) To care about the slow coming signal
(4) To talked with spurious o/p

Note:



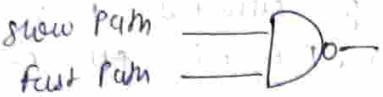
① What error happened in the previous cycle ~~data~~

② will define what will happen in the next cycle



* if faster path & slower path would never interfere then we are safe.

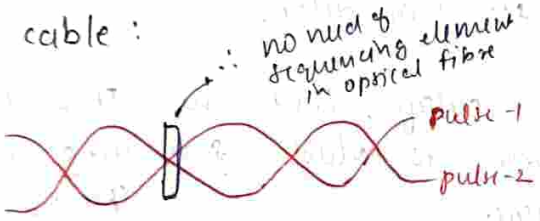
⇒ There are two i/ps, one comes very fast & other comes slow then in intermediate state, when this i/p has reached but the other one has not reached I could get glitches @ we get spurious transitions.



∴ some spurious state will appear
which will lead to
→ more power consumption
→ Extra toggling
→ possible corruption of state machine

* If we can't ensure that flow of information will always be synchronized we use what is called as tokens @ FF.

Eg. Fibre-optic cable:

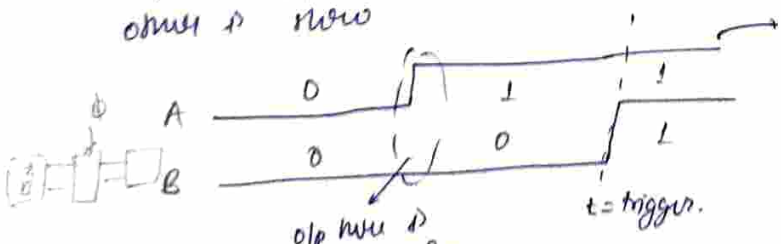


both will travel with speed of light

∴ it is only the dispersion b/w the pulses that differs. what is the minimum time b/w two pulses.

* But in terms of our hardware @ our ckt we can't always ensure that the slow path would be able to catup with fast path.
∴ We need FF or latches to delay the fast signals

Q. What is meant by spurious o/p? Because coherence we get
• we get the o/p acc. to our functionality?
→ Suppose I want to add two bits & one is fast & other is slow



∴ we need to delay this signal

Note: If tokens moved through pipeline at constant speed, then no sequencing element would be necessary.

Sequencing Overhead :

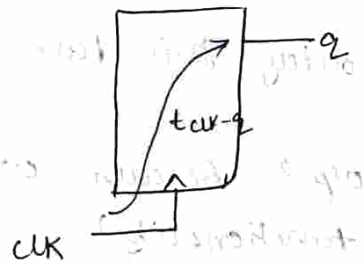
- Def: When we use this sequencing elements in our path (ie, we are adding FF & latches) & they will inevitably lead to additional delay even on the slower paths. (why?)

  ↳ When you know you will have some slow path & fast path in your system, but you exactly don't know which path is fast & which is slow.

  ∴ We put FF on all the paths, then the slower path will also see some delay.

* This additional delay, that this FF or latch will take for slower path is called sequencing overhead (or) clocking overhead. (better)

- This kind of overhead also occurs in token system. In token system (ie, in asynchronous system) it is called as clocking overhead.



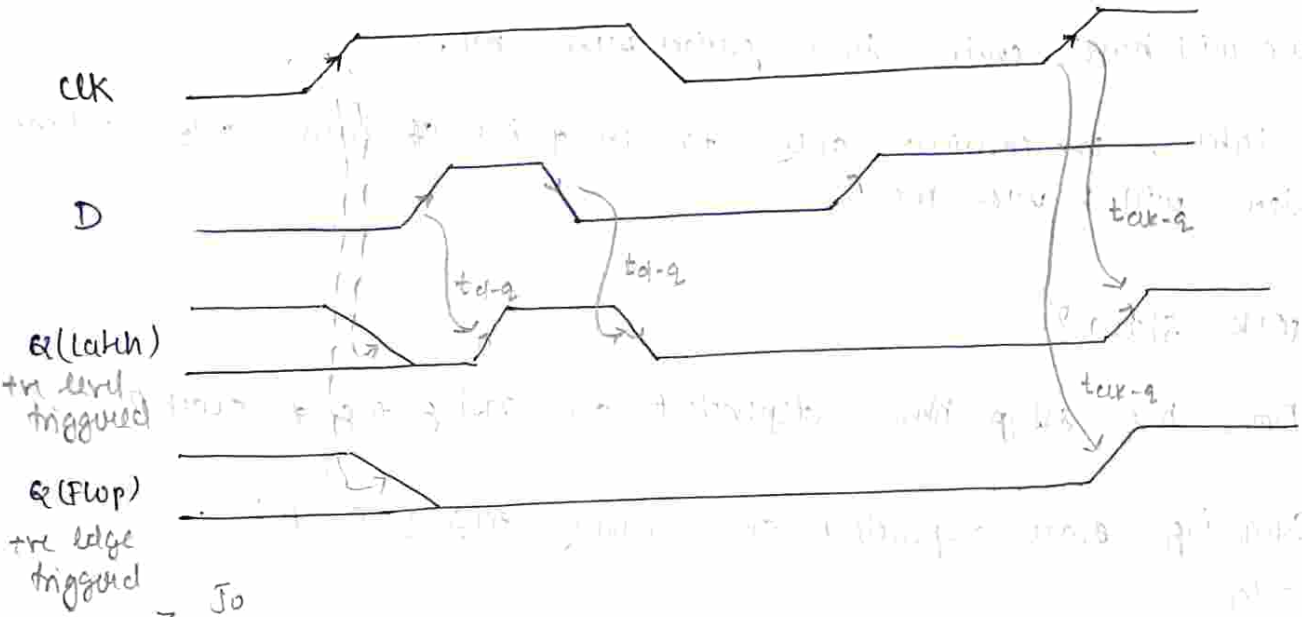
* The sequencing overhead will be the FF delay ie. clk to Q delay of the FF.

→ Because that is the extra delay that your logic has to travel.

States	token
synchronous	Asynchronous
sequencing overhead	clocking overhead

Sequencing Elements

1. Latch → level sensitive (transparent Latch, D-Latch)
2. Flip-Flop → edge triggered (master-slave FF, D-FF, D-register)



So

* Latch can have both t_{d-q} delay & t_{clk-q} delay
  - if d comes late (i.e, after the clk) then t_{d-q} delay will be there.
  - if clk comes late (i.e, after the D) then t_{clk-q} delay will be there.

* FF will only have t_{clk-q} delay.

Note: (i) Only on the clk edge FF will be transparent and other than that it ~~is opaque~~ will hold its previous value

(ii) Only on the +ve level of the clock latch will be transparent & other than that it ~~is opaque~~. will hold its previous value.

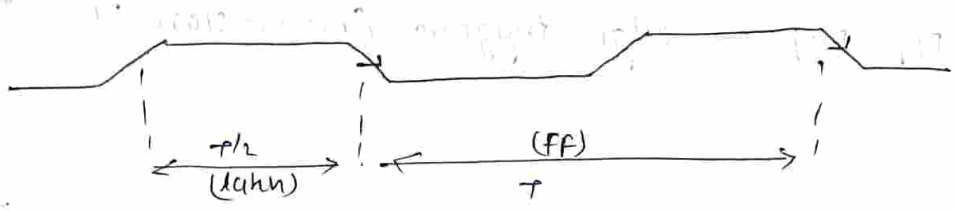
* FF will transit the data whatever is there on the rising edge. (for +ve edge triggered FF).

Que: Why we prefer FF over the latch?

→ ① Transparent window for FF is very low ∴ it ~~will not~~ is less likely that it will transmit any glitch.
But latch will also transmit the glitch.

Q. Why we prefer FF over latch when we study STA?

* (2) For FF, we consider sampling at -ve edge of the clock then



we will have entire time period with me.

* For latch, we consider only +ve level of the clk then only T/2 time period will be with me.

(3) To avoid Race cond^n in latch.

Q. clock skew?

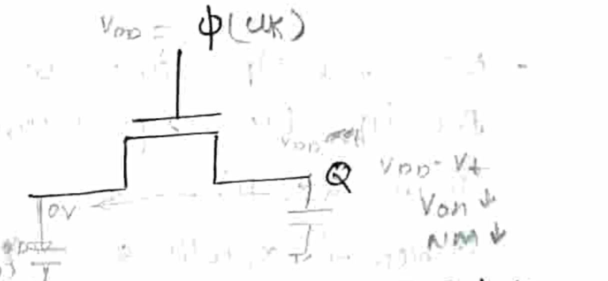
Q. Does the setup time dependent on rising edge of clock?

Q. Sampling time dependent on rising edge of clock?

→ Yes.

1. Latch Design :

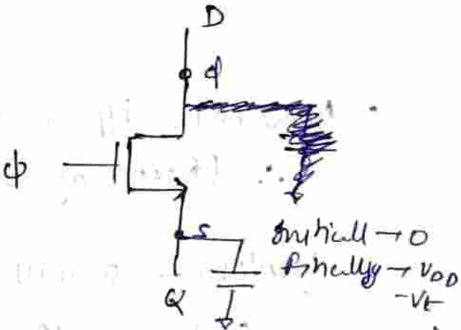
(1) Pass Transistor Latch :-



Pros :- · tiny
· low clock load

~~(a) Q will have floating value for clk = 0.~~

Cons :- (i) ① V_T drop at Q, the o/p doesnot swing from 0 to V_{DD}.



Assume, D as (drain) & Q as (source)

o/p being able to drive i/p

if D = V_{DD}, & φ = logic-1

when Q becomes $V_{DD} - V_t$

· Back driving ∴ $V_{GT} = \phi - (V_{DD} - V_t)$

· Non-restoring $= V_{DD} - V_{DD} + V_t = V_t$

∴ Q can only go to $V_{DD} - V_t$

Signal can flow in both directions.

③ Back-gate driving (Q will change the i/p D, when D=0)

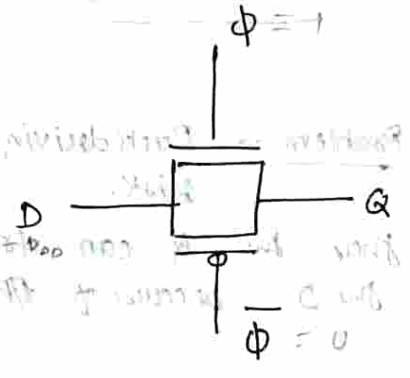
④ Non-restoring. (when clk = 0, charge leakage occurs)

(ii) ② The o/p is dynamic ⓐ the o/p floats when the latch is opaque.

(iii) D drives the diffusion i/p of a pass transistor directly, leading to potential noise issue & making delay harder to model with static timing analyzer.

(iv) The state node is exposed, so noise on the o/p can corrupt the state

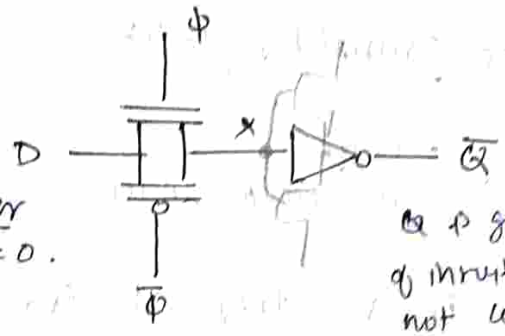
(2) Transmission Gate Latch



- Now it has no V_T drop
- But other problem remains same
- Extra problem is, it requires inverted clock also.

(3) Cascading with Inverter (Inverting Buffer)

- Now the o/p can't back drive the i/p (i.e., signal can't flow from Q to D).

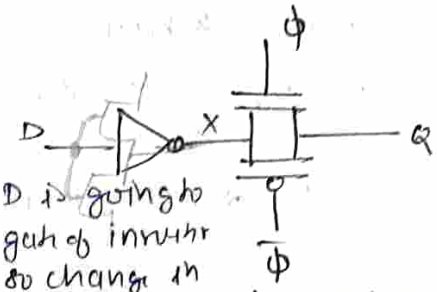


Q is going to drain of inverter ∴ it is not connected to diffusion i/p.

Problem → X will go on floating or dynamic in nature when both Φ = 0.

- Now the i/p is no longer driving the diffusion.
  ∴ (Restoring o/p)

Problem → Q will be floating or dynamic in nature when both Φ = 0.



D is going to gate of inverter so change in Q would not transmit on to D.

We can fix only either of two problems:-

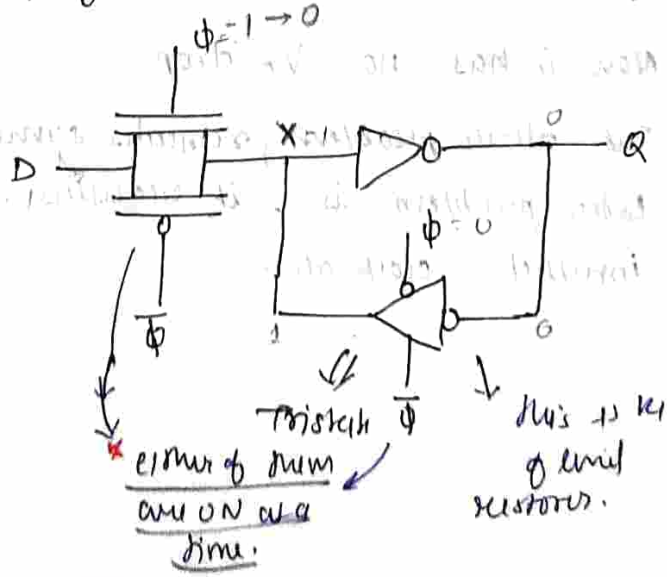
(i) Output noise sensitivity or

(ii) Diffusion i/p

(4) Tristate F/B → This will solve the problem of floating value of X.

Problem → Back driving risk.

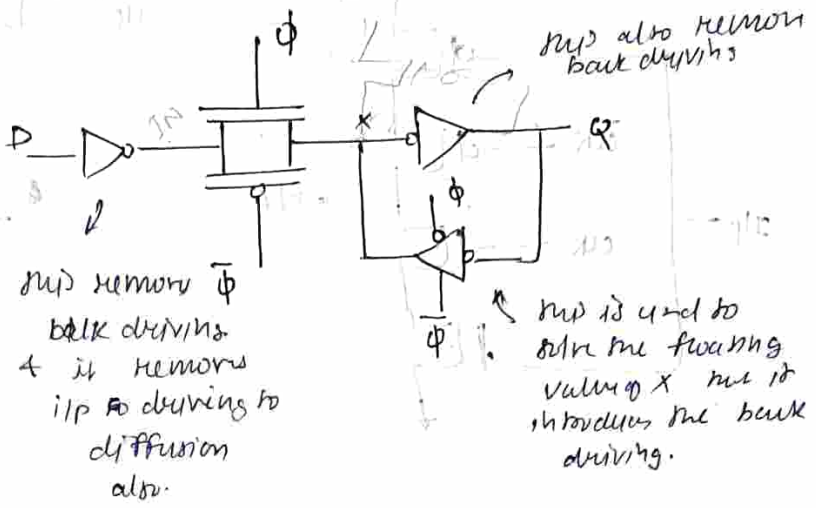
That the Q can drive the D because of F/B path.



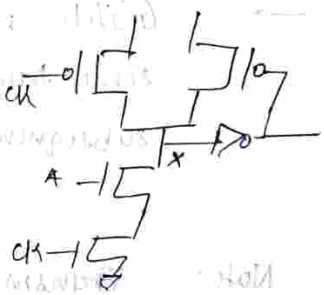
* static latches are now essential because of leakage

(5.) Buffered i/p with f/B at o/p → To remove the risk of back driving.

Problem: There can be some noise on Q that will impact X.



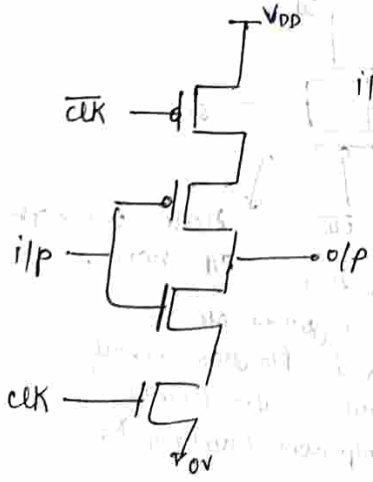
* We put pmos in dynamic logic as level restorer to keep X at 1.
* But here we put tristate in place as level restorer so to keep X at both 1 & 0 at required time

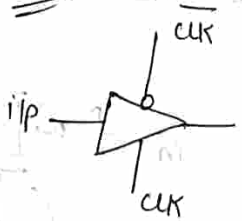


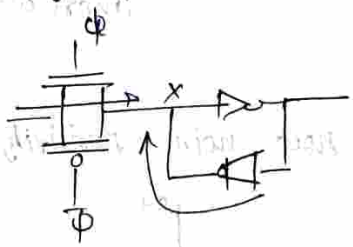
Q. Why did we put tristate & not a simple inverter?

→ if tristate replaced by inverter then both D will be driving X & inverter will also be driving X hence it would be difficult to study them.

Tri-state inverter : (1) bad

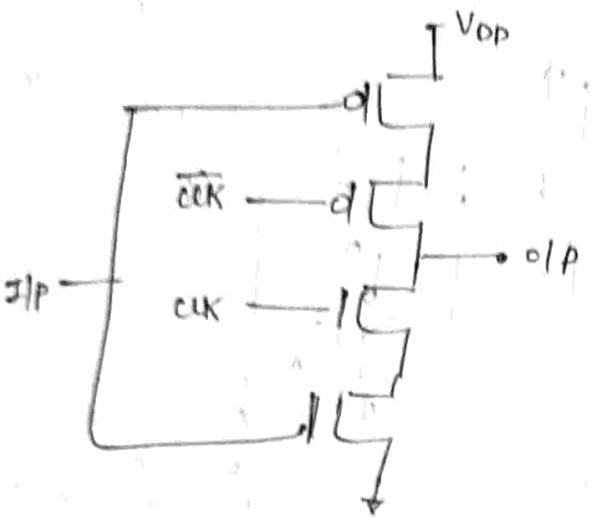


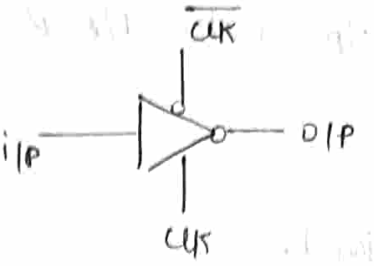




clk	i/p	o/p
0	0	Z
0	1	Z
1	0	1
1	1	0

Tristate inverter (2) good



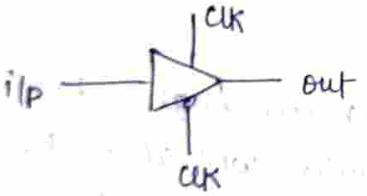


CLK	I/P	O/P
0	0	Z
0	1	Z
1	0	1
1	1	0

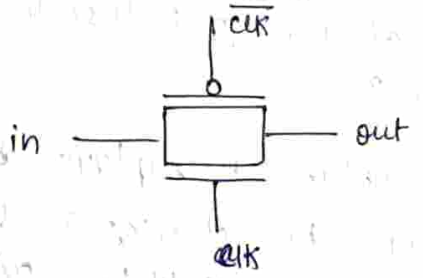
Q. What is the difference b/w the dip & the glitch?

→ Glitch means that there is something rise or fall i.e, some transition on the signal is so large that the subsequent stage samples it as it as a change in input.

Note: Transmission gate is advanced version of tristate ~~buffer~~



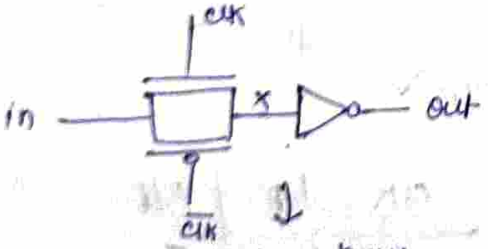
(more area)



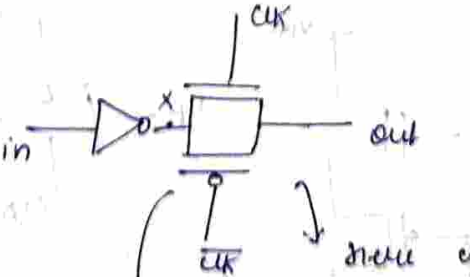
(less area)

How noise sensitivity has been reduced?

→



here there is no o/p noise because out is driven by an inverter

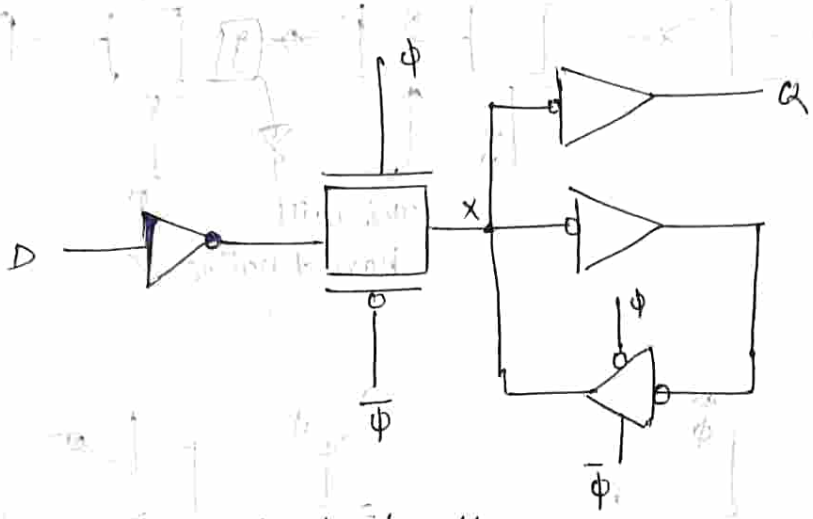


when the transmission gate is off ∴ node x is floating as well as out is also floating ∴ o/p noise sensitivity ↑.

here we have o/p noise.

(6) Modified Buffered i/p Latch with f/b :

- In Buffered i/p latch, the noise on Q can impact the X.
- But in modified ckt the noise at X can be reduced by,

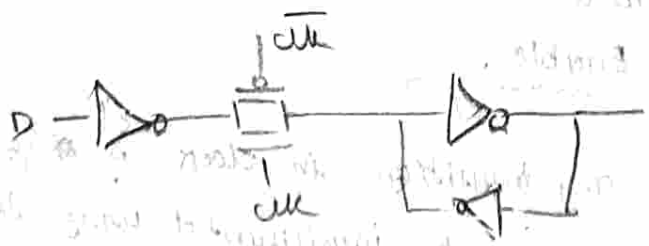


- It is widely used in standard cells,
  * Very robust (most important)
    - Power large
    - Power slow (1.5 - 2 FO4 delay)
    - High clock loading

Q. Are we taking clk (ϕ) & $\overline{clk}$ ($\bar{\phi}$) as independent signals or $\overline{clk}$ is o/p of inverter with i/p as clk.

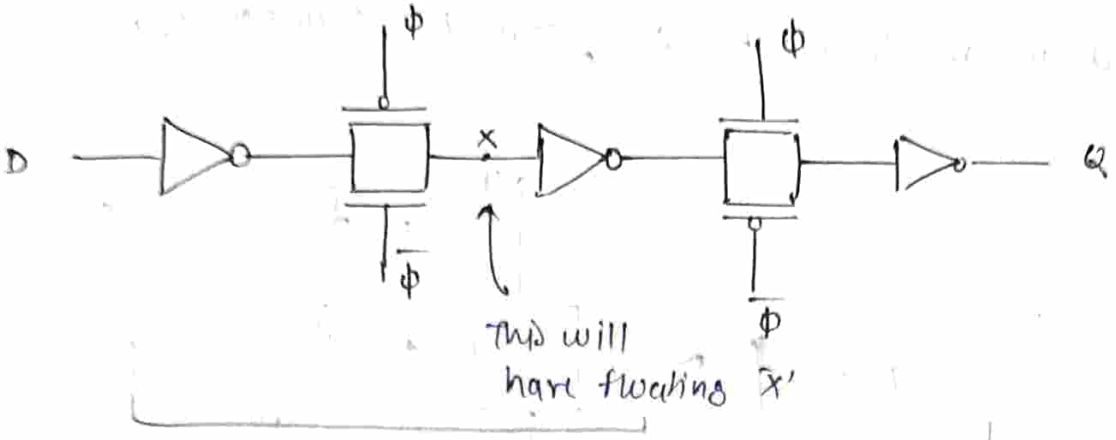
→ We will generate clk from PLL (clock generator) & pass that clk through the inverter to get $\overline{clk}$, where we will see some delay in $\overline{clk}$ wrt clk.

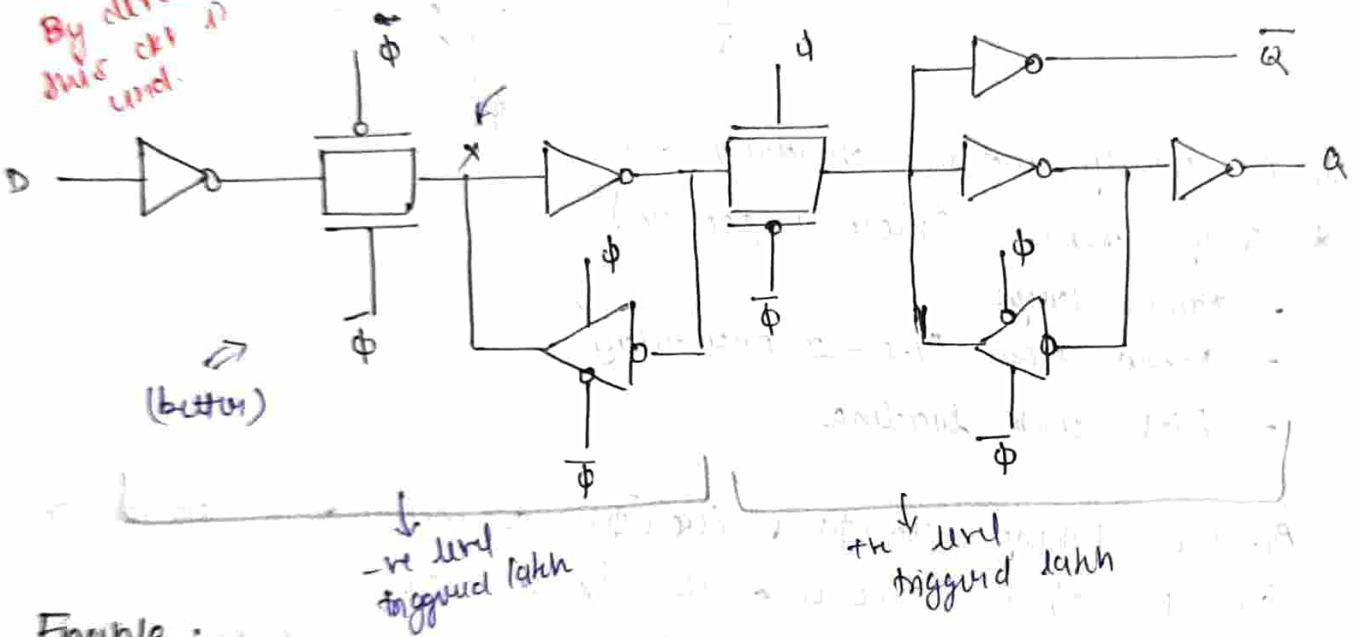
∴ There will always be delay b/w clk & $\overline{clk}$



2. Flip-Flop Design : Remainy

- FF is built as pair of back-to-back latches.





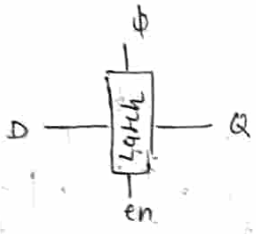
Enable :

- To reduce dynamic power we will gate some part of the ckt so that there is no clock toggling there hence we save toggling power of the signal
- clock gating is enabled in FF & latches by using the signal called Enable.
- When Enable = 0, any transition in clock is ignored; And this above system can be implemented using the
  (i) MUX
  (ii) AND & NAND gate

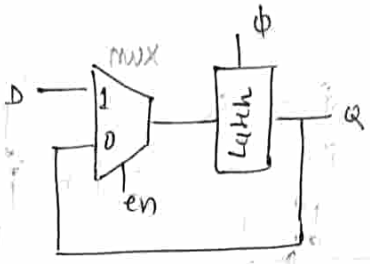
* When enable becomes zero then ~~clock~~ out becomes independent of clk.

- MUX : increases latch D-Q delay ($t_{d-q}\uparrow$)
- Clock gating : increases Enable(en) setup time, skew

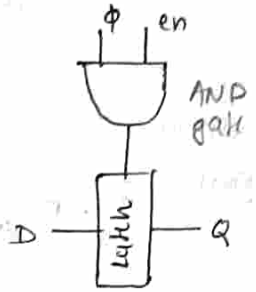
Symbol

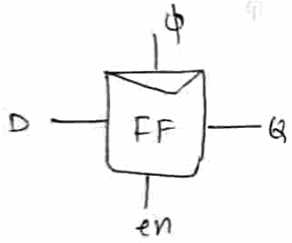


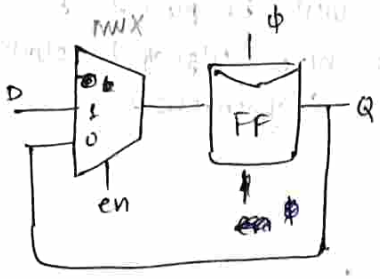
MUX design

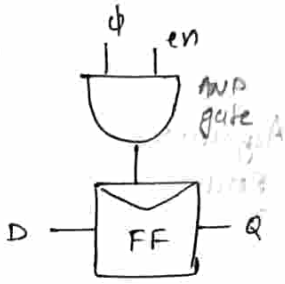


Clock gating design









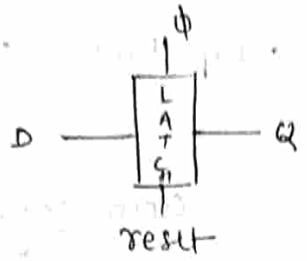
* Now signal & clock should come, there is some constraint.
  (i) Constraint b/w D & clk(ϕ)
  (ii) constraint b/w en & clk(ϕ)
  (iii) No constraint b/w D & en.

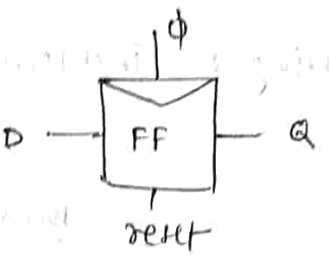
* when en = 0, Q will hold its value independent of clk(ϕ).
  when en = 1, Q will hold D with some delay like
  $t_{comb} + t_{clk-Q}$.

Reset :

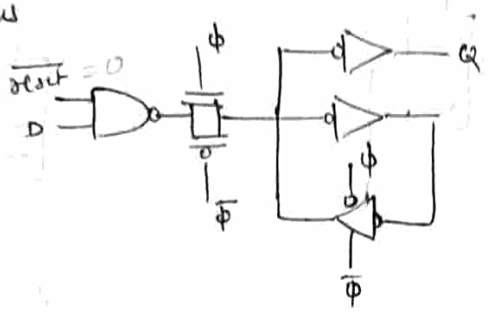
* When reset is asserted then o/p is pushed to zero/low.

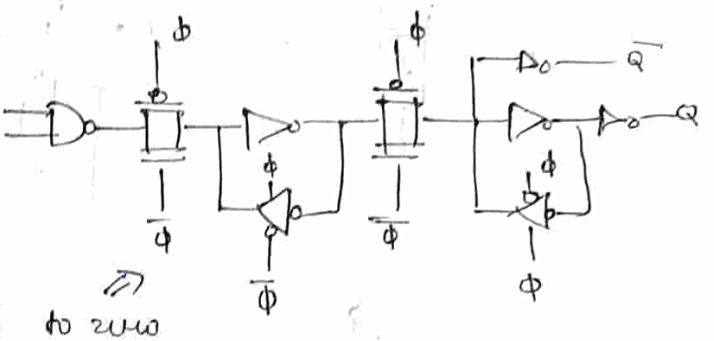
Symbol





Synchronous Reset

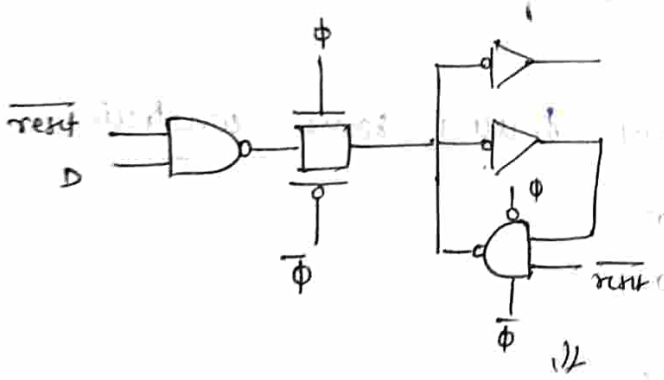


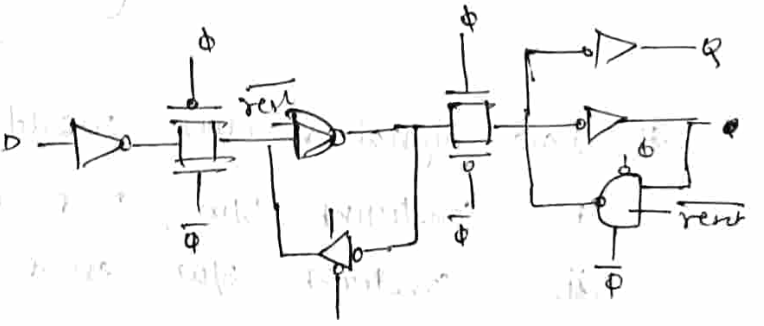


(o/p will be pushed to zero at next edge of the clock)

∴ synchronous

Asynchronous Reset





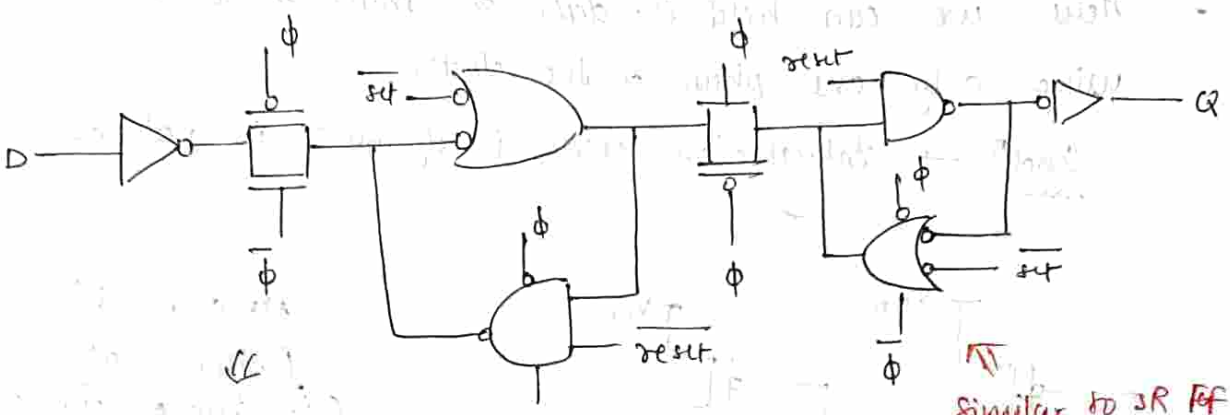
Asynchronous reset.

Q. How they are working & how they are going to get reset?
working?

Set / Reset : remarks

(i) Set forces the output to high value when Enable = 1.

(ii) FF with asynchronous set & reset:



They are generally for asynchronous machine.

Similar to SR FF in digital ckt.

Q. Why we need set & Reset?

→ (i) When power cuts we want to set all the states to '0' ∴ reset is used.

(ii) For memory, some are expected to be at '1' & some are expected to be at '0', ∴ we need both set & reset.

➡ As we have seen that, there is possibility of overlap b/w clk & $\overline{clk}$ which can lead to back gate driving, then we added Inverter in FIB to avoid back gate driving.

* Also we can use TSPC, we can avoid all the problems occured in transmission gate latch & FF.

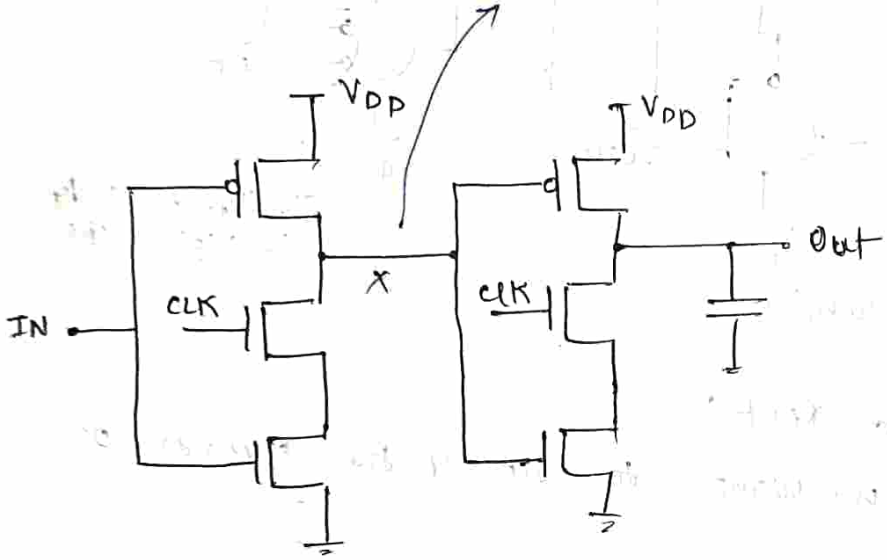
Q. How, back driving occurs when, clk & $\overline{clk}$ are overlapping?

3. TSPC [True Single Phase Clock]

Rem:

- Using this FF design we can avoid all the problems like back driving o/p.
- Here we can hold the data or make it as a latch by using only one phase of the clock.

Cond^n → Intermediate node is dynamic in nature.



(Armor for CLK)

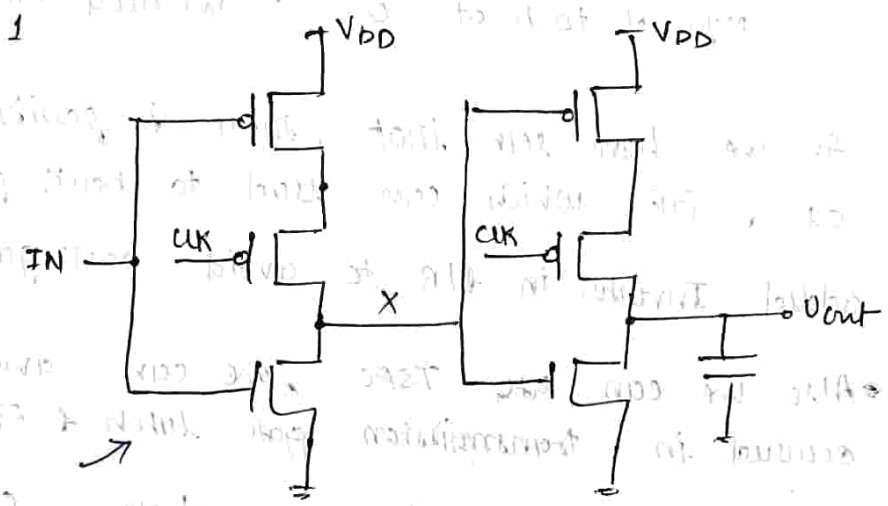
Positive latch (i.e. transparent when CLK = 1)

for CLK = 1,
$X = \overline{IN}$
& $Out = \overline{X} = IN$

Problem ↑

Node 'X' will be floating for CLK = 0 & IN = 1

CLK	IN	X	Out
0	0	1	Z
0	1	Z	?
1	0	1	0
1	1	0	1



Node 'X' will be floating for CLK = 1 & IN = 0.

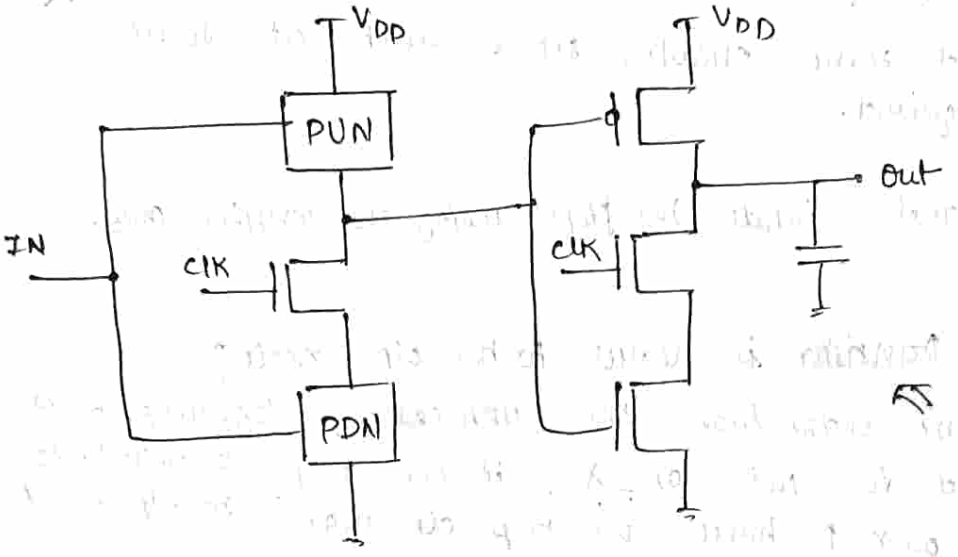
CLK	IN	X	Out
0	0	1	0
0	1	0	1
1	0	Z	?
1	1	0	1

Negative latch (armor for CLK)

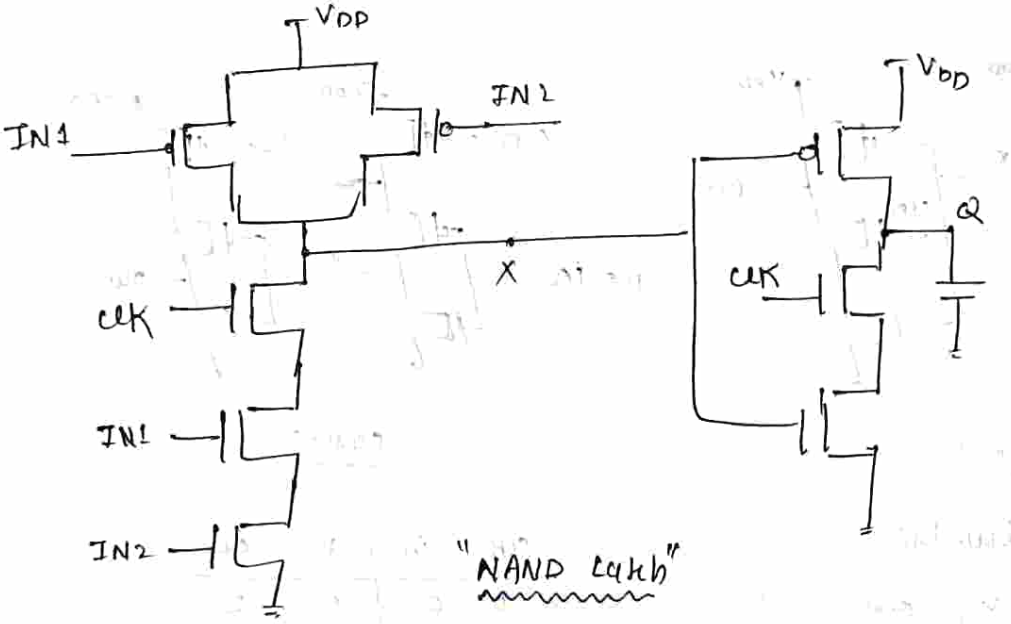
(i.e. trans-parent when CLK = 0)

for CLK = 0,
$X = \overline{IN}$
& $Out = \overline{X} = IN$

Including logic in TSPC



⇐ logic inside the latch.



"NAND latch"

when, IN1 & IN2 = 1 & CLK = 0 then node 'X' will be floating.

CLK	IN1	IN2	X	Out (output of NAND gate)
0	1	1	Z	?
0	1/0	0/1	1	0
0	0	0	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

Q. By default what design do we use?

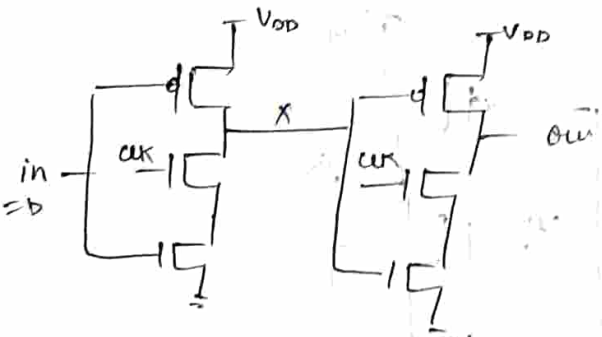
→ The static design with so many extra transistors w/o the enable pin w/reset & w/o set pin.

* You can add some enable, set & reset at some places if required.

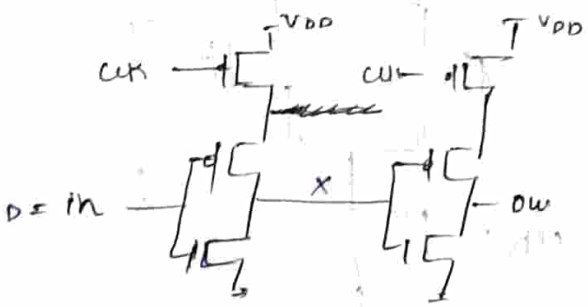
* $\overline{CLK}$ is generated inside the flop using the inverter only.

Q. Why clk i/p transistor is nearer to the o/p node?

→ Because we don't want that the unnecessary toggling of D should not lead to noise on X, if we keep D nearer to o/p then noise on X ↑ hence we keep clk nearer to o/p & not the D.



case-1



case-2

We know about flip.

clk	in	X	out
0	0	1	Z
0	1	Z	?
1	0	1	0
1	1	0	1

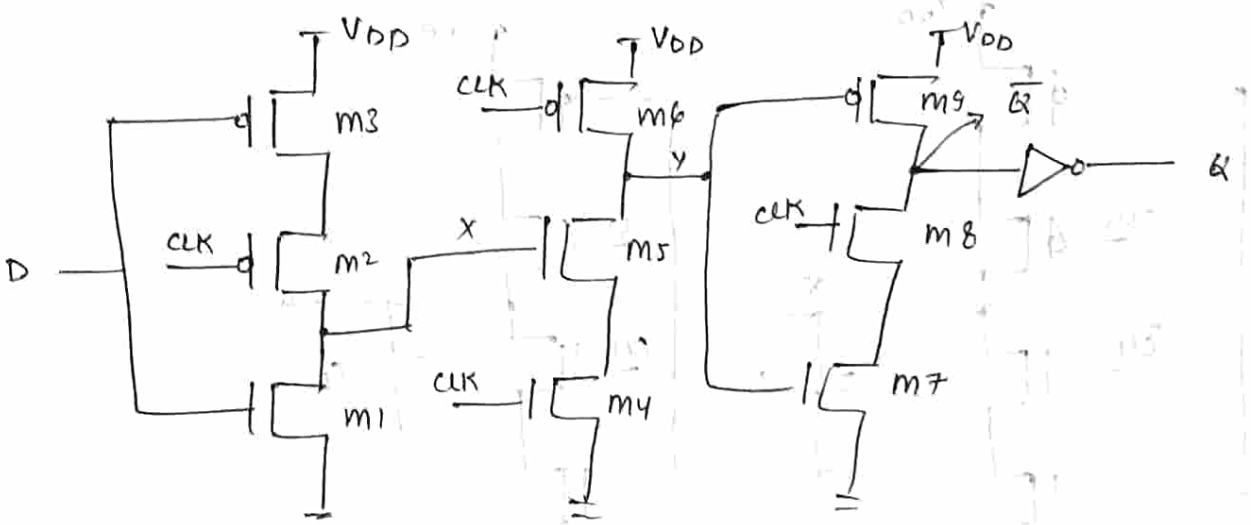
when clk = 1,
out = in
∴ latch with level triggered clk.

clk	in	X	out
0	0	1	Z
0	1	0	1
1	0	Z	?
1	1	0	Z

this is not acting as any transparent ckt.
(∴ this is wrong design)

* keeper can be added to make circuit pseudo static.

TSPC - Flip Flop :



CLK	D	X	Y	Q̄	Q
0	0	1	1	Z	?
0	1	0	1	Z	?
1	0	Z	?	?	?
1	1	0	Z		

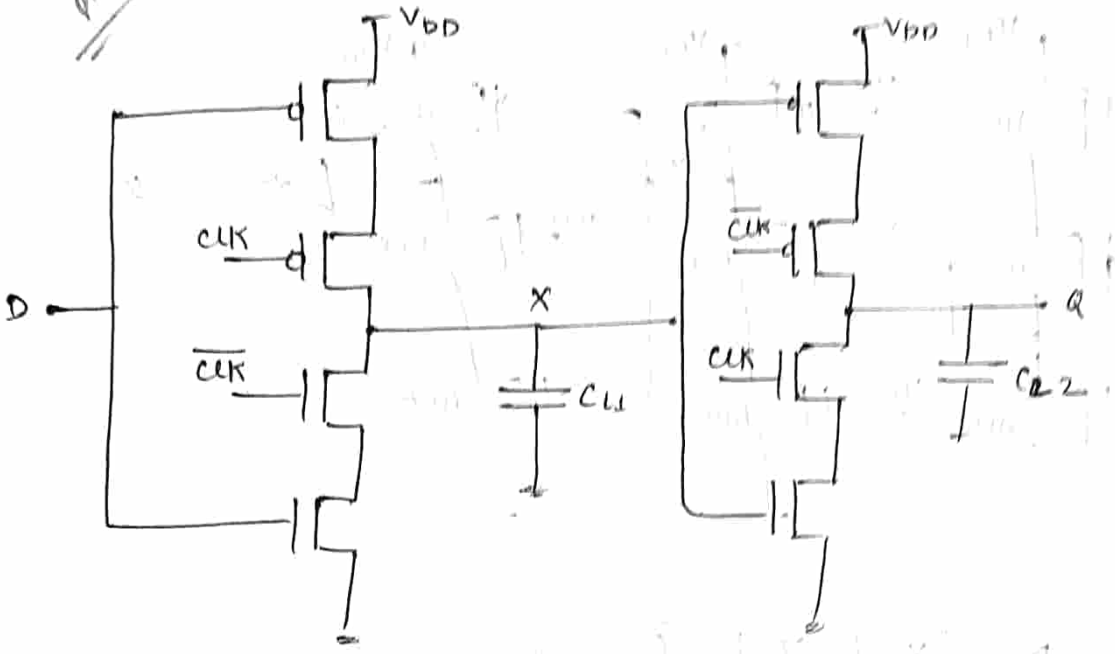
Q. Can we put CLK upside ? near V_{DD}. (CLK on top & 'D' on CLK)

→ Since D FF is edge triggered ∴ only at edge of CLK D will impact the node X.

So if D is closer to the X then even when edge is not triggered then also it can impact the o/p, w/o triggering of the CLK.

4. C²MOS

Keeper can be added to the ckt to make it pseudo-static (i.e. X will not float).

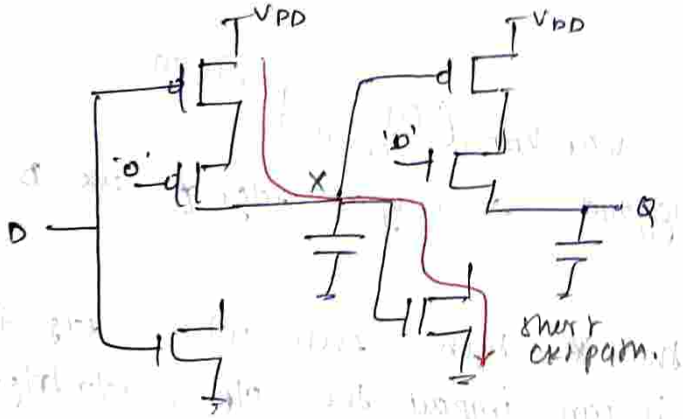


master stage

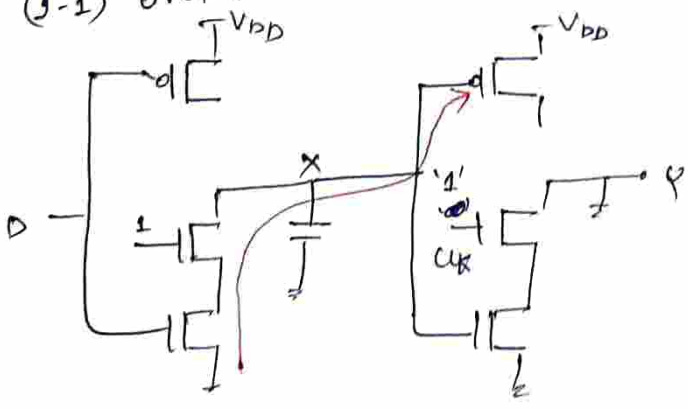
slave stage

Transitivity & clock overlap:

(i) (0-0) overlap



(ii) (1-1) overlap.



Sequential Circuit - Timing

Sampling Edge: The edge of the clock at which the data gets sampled is called the sampling edge.

- Sampling edge for a FF is:
  - For +ve edge triggered FF, it is +ve edge } where data is sampled.
  - For -ve edge triggered FF, it is -ve edge }
- Sampling edge for a Latch is:
  - For +ve level triggered Latch, -ve edge is where data is sampled. (Why?)

(+) because it starts to become opaque after +ve level.

  - For -ve level triggered Latch, +ve edge is where data is sampled.

Note: As we already know that there are some delays b/w i/p & o/p called propagation or contamination delay.

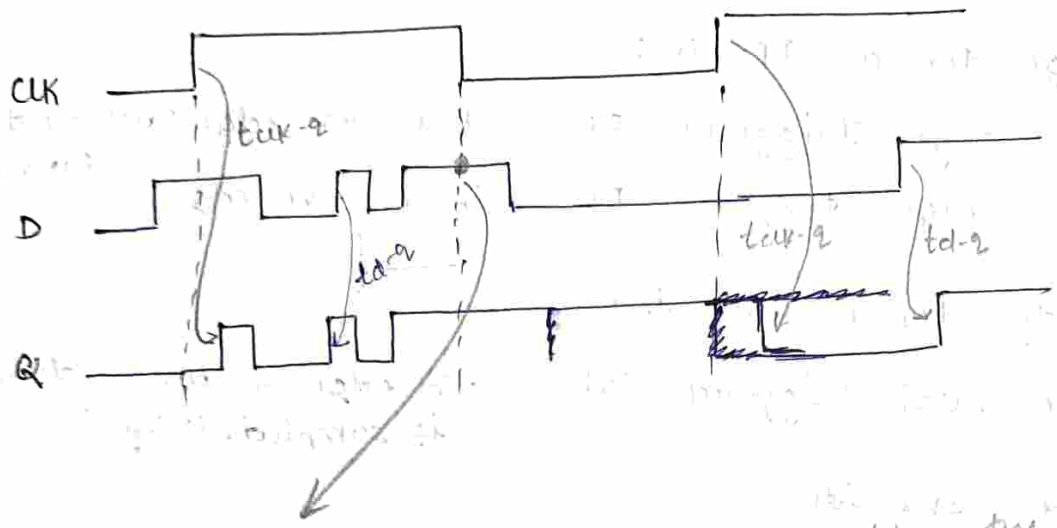
- But FF & Latch have some additional feature called sampling edge & with respect to this there are delays associated to it.

* Delays wrt i/p (D) & o/p (Q) are:
  (i) propagation
  (ii) Contamination

* Timings associated with sampling edge are:
  (i) Hold time
  (ii) Setup time

Q. Why +ve level latch ~~dea~~ sample at -ve edge of clock?

→ Because it became opaque after the +ve level of the clock ie. at the -ve edge of the clock.



∴ At opaque duration it is sampling the value of D which it has observed (the value of D) during the -ve edge of clk.

Latching → fasten (a door or gate) with a Latch.

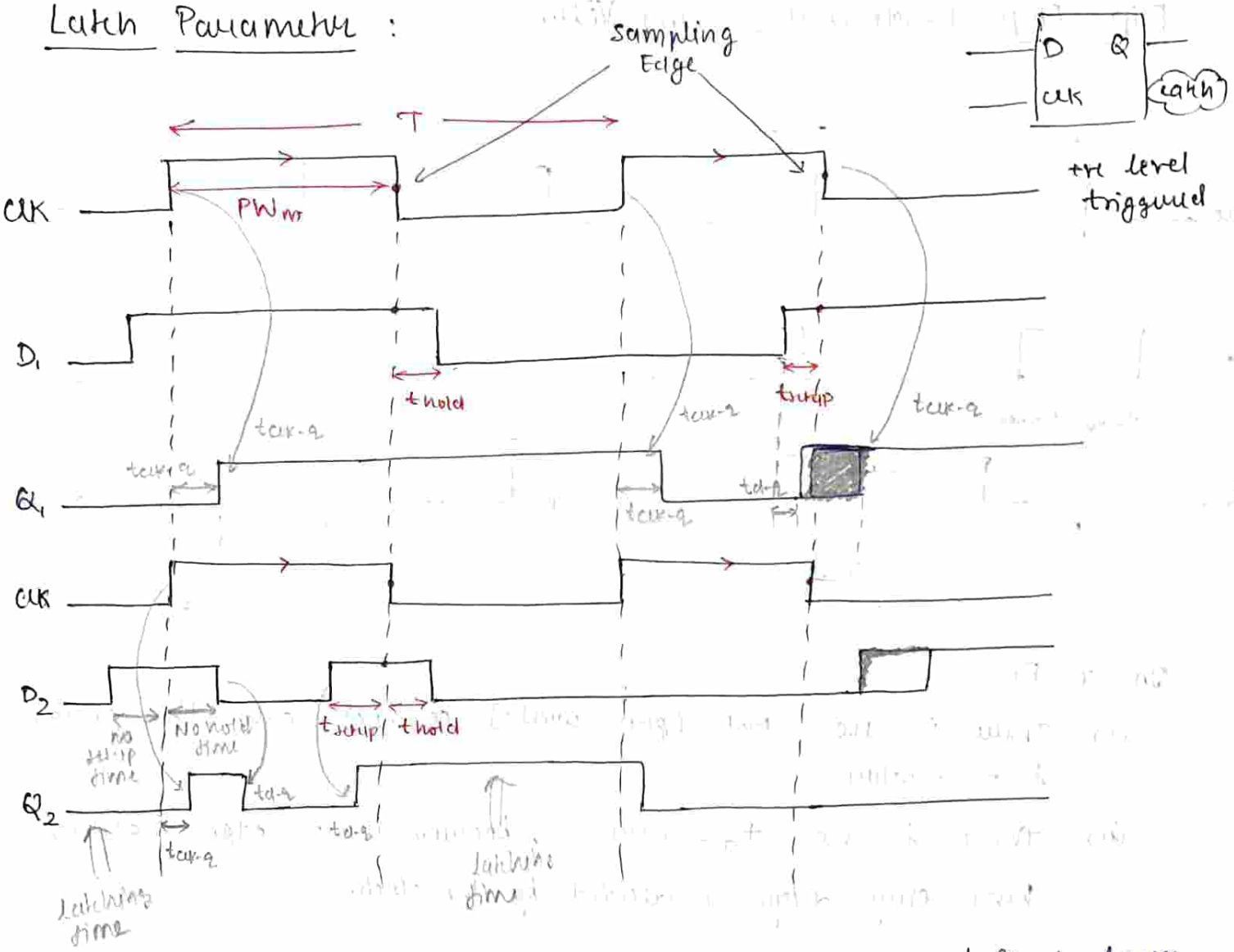
* We will only consider the setup & hold time at the falling edge of the clock for a +ve level triggered Latch.

∴ we are considering that edge from where latching action starts that is the sampling edge.

Setup time :- (t_{setup}) It is the time before the sampling edge for which the i/p should be stable.
[it is b/w two different sampling edge]

Hold time :- (t_{hold}) It is the time after the sampling edge for which the i/p should be stable.
[it is b/w same sampling edge]

Latch Parameter :



* setup time (t_{setup}) & hold time (t_{hold}) are defined from the sampling edge of the clock.

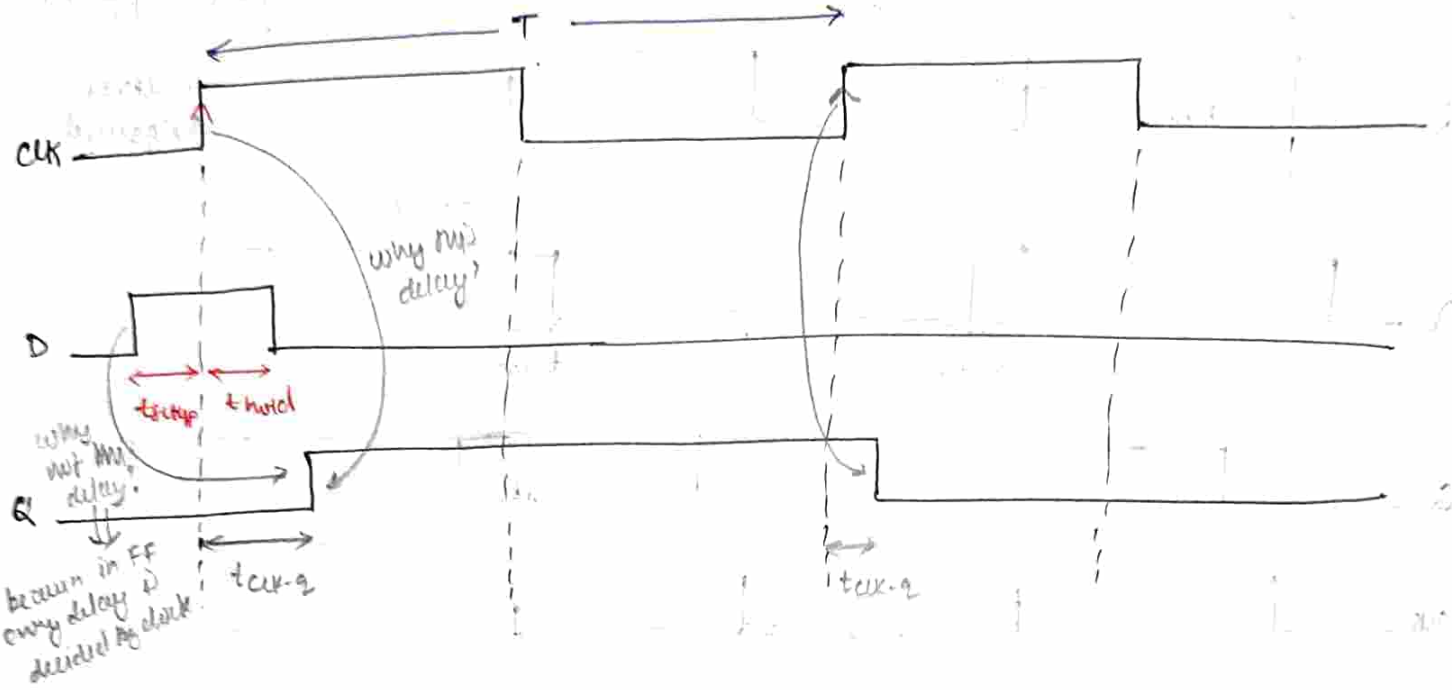
* We will only consider the falling edge of the +ve level triggered for calculating t_{setup} & t_{hold}.

Double edge triggered FF → They are both edge of the clock for sampling so they are essentially doing half the pulse period for evaluation

* Pulsed Latch? → A ~~clock~~ pulse generator that behave as FF rather than a clock.

Note: * For -ve level of the clock, the value of Q_1 will ~~be~~ hold its ~~equal~~ ~~to the~~ value that it has sampled ~~at~~ at the sampling edge.

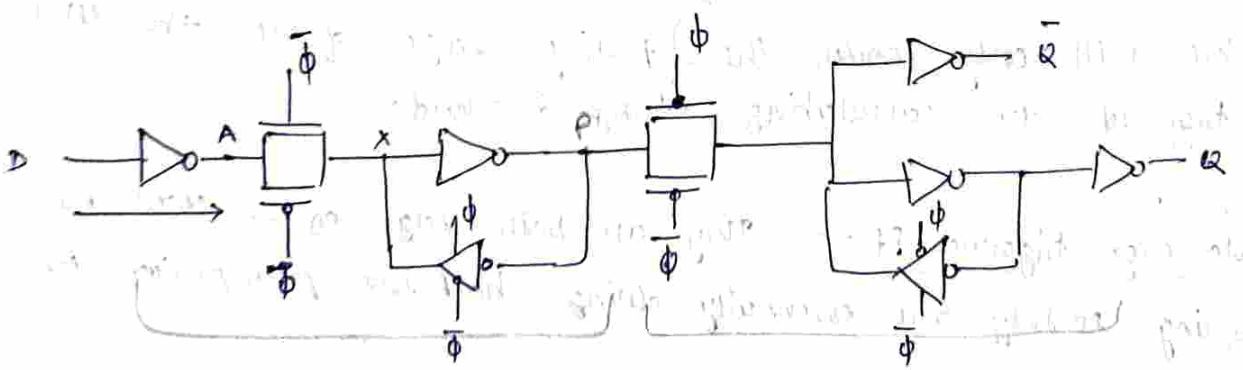
Flip-Flop Parameters : "Imp Video" (Video - 68)



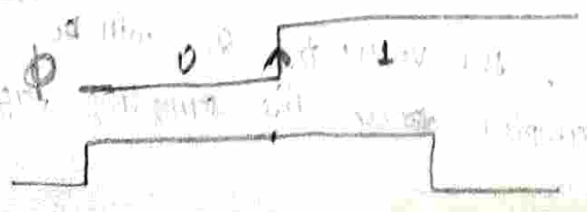
- In a FF :-
  - (i) There is no PW (pulse width) concept, only time period that matters
  - (ii) There is no t_{d-q} delay, because it is edge triggered, hence every delay is decided by the clock.

Q. Why here we are taking only clk-q delay & not d-q delay?

→ Let's take an example of FF,



when $\phi = 0$, then D can only go upto A.P.
when $\phi = 1$, then ϕ can go to Q hence what is triggering the Q?
→ ϕ i.e. clk is triggering the Q hence clk-q delay is considered.



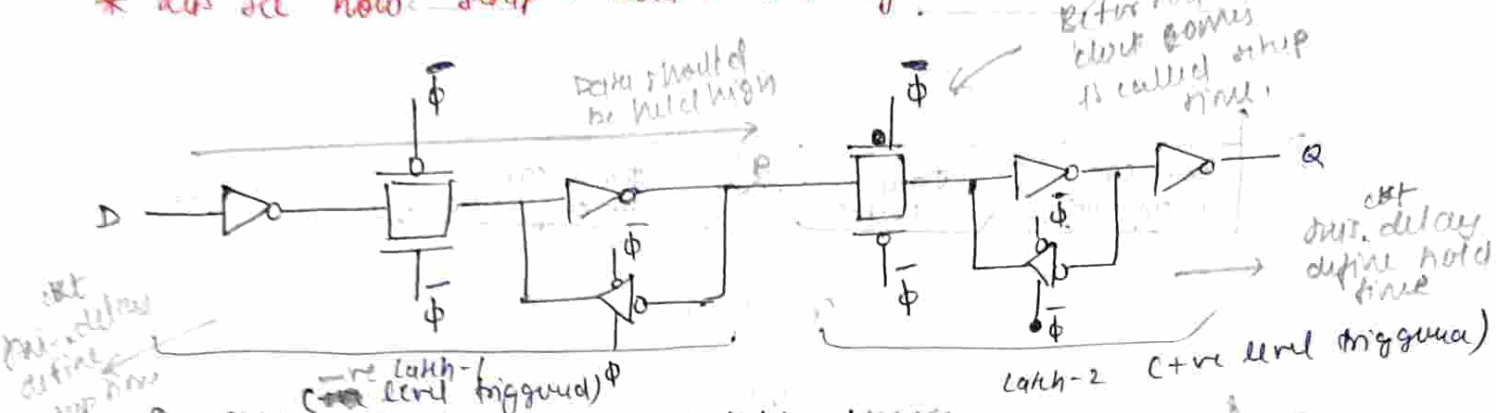
Q Can your t_{clk-q} delay be less than hold time?

good question.

can $t_{clk-q} < t_{hold}$

→ It depends on the data path.

* Let's see how setup & hold time is defined.



~~For the falling edge of the clk 1st latch becomes~~ (clk = 0)

- For the rising edge of the clk the transmission gate-1 becomes transparent and D is written into the 1st Latch.
- when clk = 1, Latch-1 becomes opaque & Latch-2 becomes transparent (so it can hold the sampled data from Latch-1) so the data from Latch-2 comes to Q.

** D should be held high until & unless my first part is totally opaque is called the setup time.

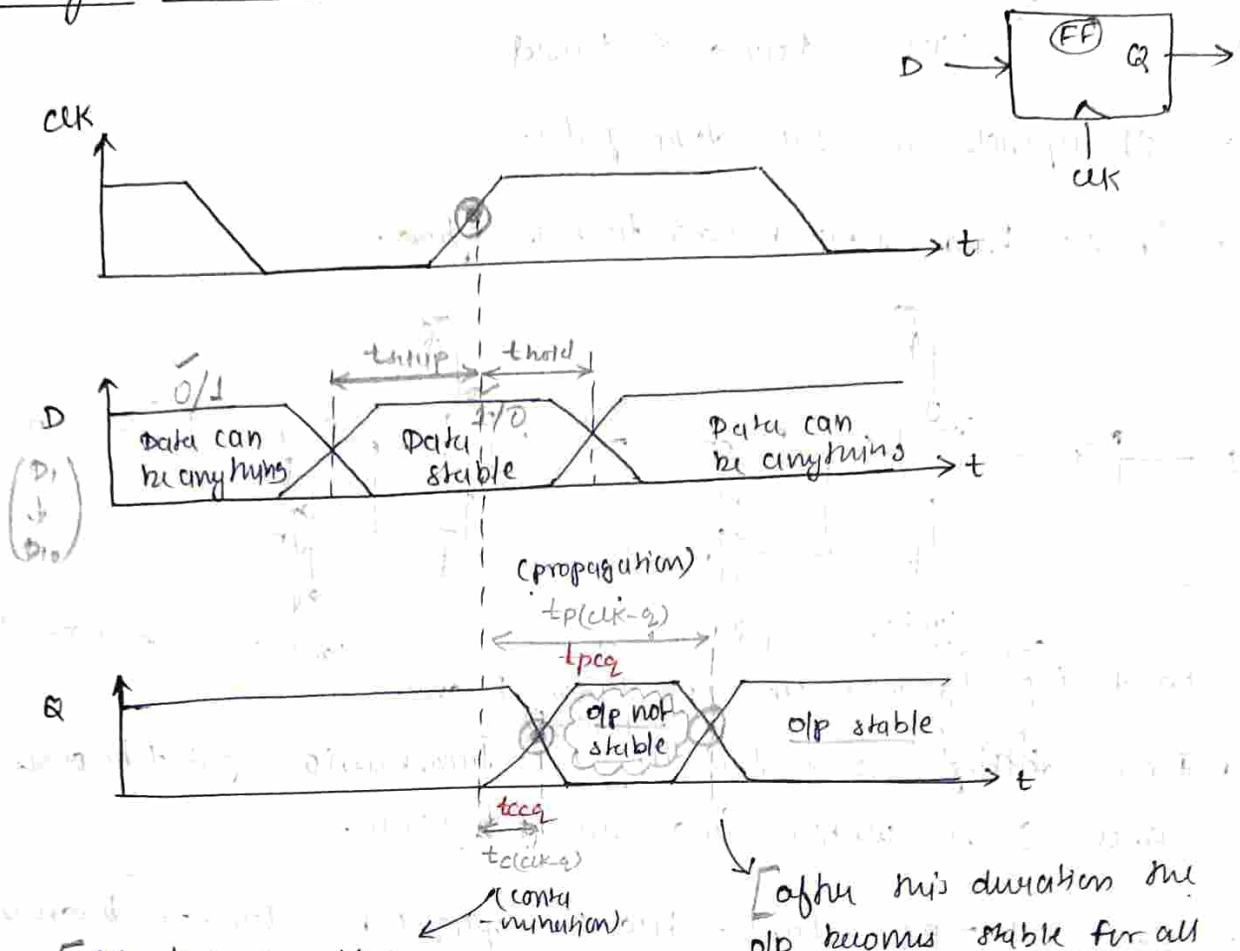
* the clk-q delay will totally depend on the second part of FF.

- * You define setup & hold time based on what is the impact of bringing D closer to clock edge on t_{clk-q} delay.

Q. How setup & hold time are related & different from propagation delay?

→ * stick to the definition of setup & hold time & according to that look at the ckt how much can many delays that can be added to the setup & hold timings.

Timing Definitions:



Eg. For 8-bit Register, there are 8-FF, the o/p will start to vary after time of t_{clk-q} (t_{ccq}) but only after t_{pcq} all the bits have been toggled and no further changes are observed.

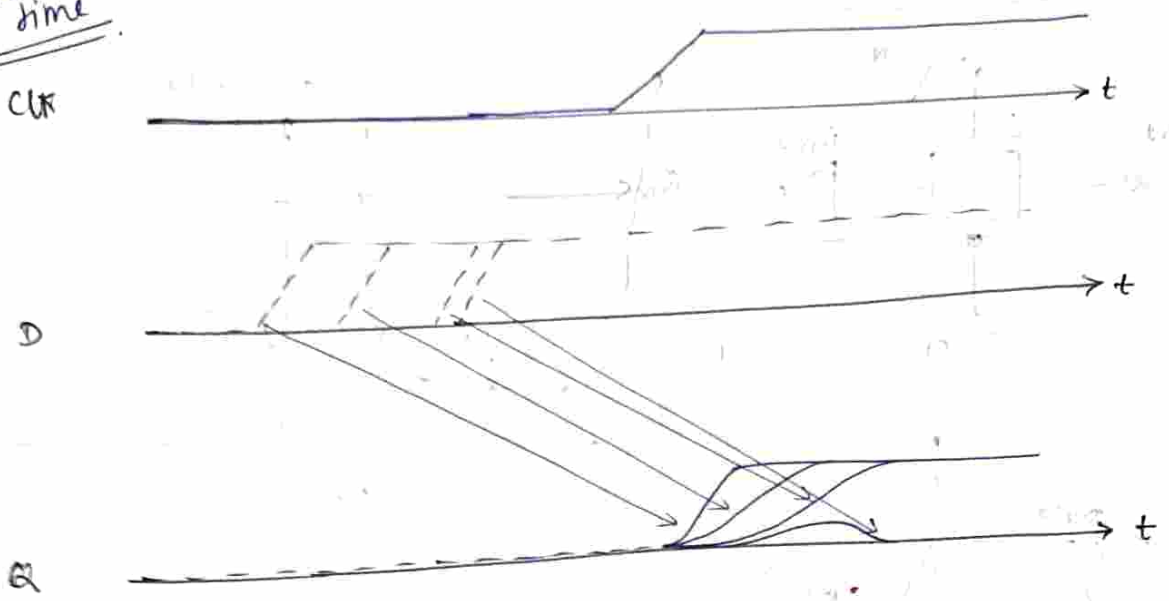
- After contamination delay → o/p starts to vary (change)
- After propagation delay → o/p settles to whatever changed value it was assuming.

Characterizing setup / hold Time:

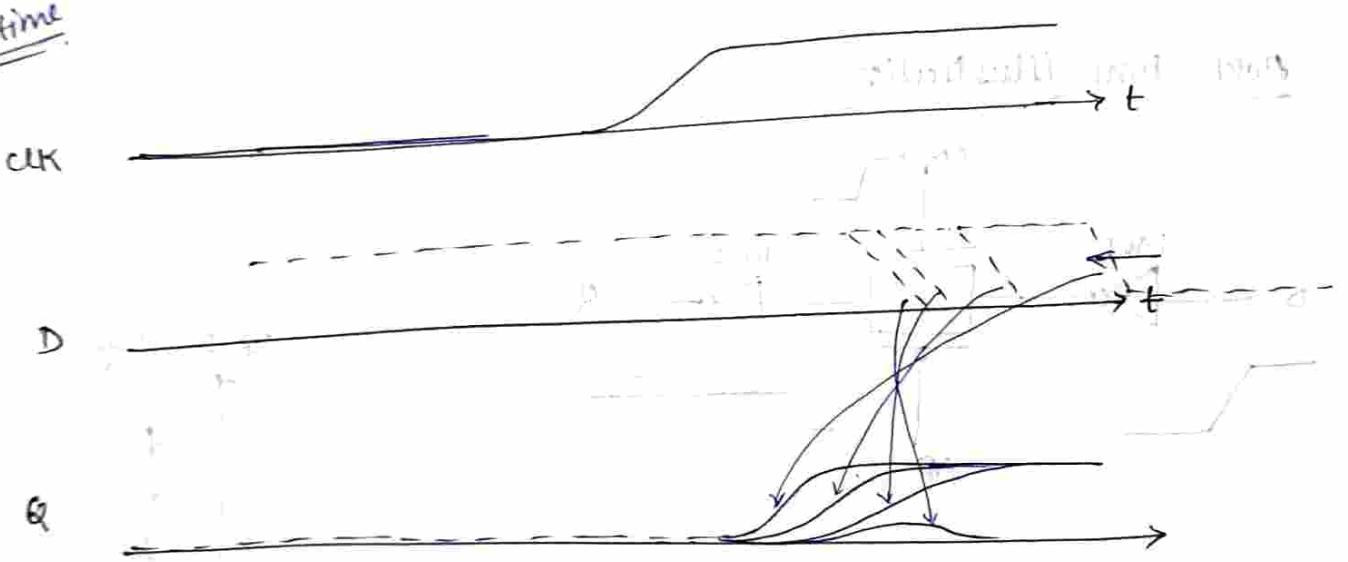
- As I bring my data close to the clk-edge then the o/p delay or clk-q delay starts increasing because ↑ in the o/p slope.

Q. Why does the o/p slope increase on bringing D closer to clk edge?

For setup time

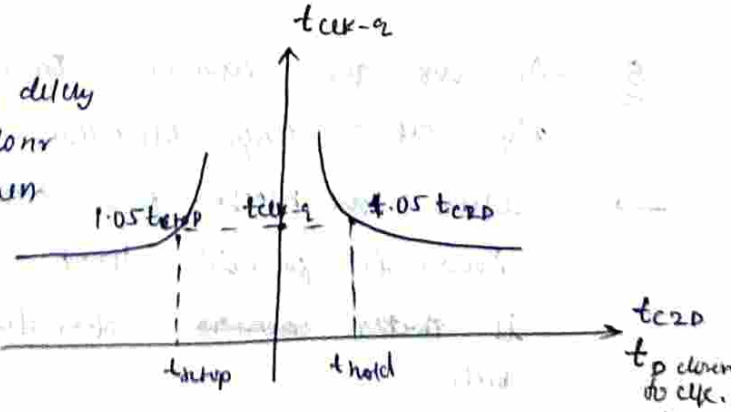


For hold time

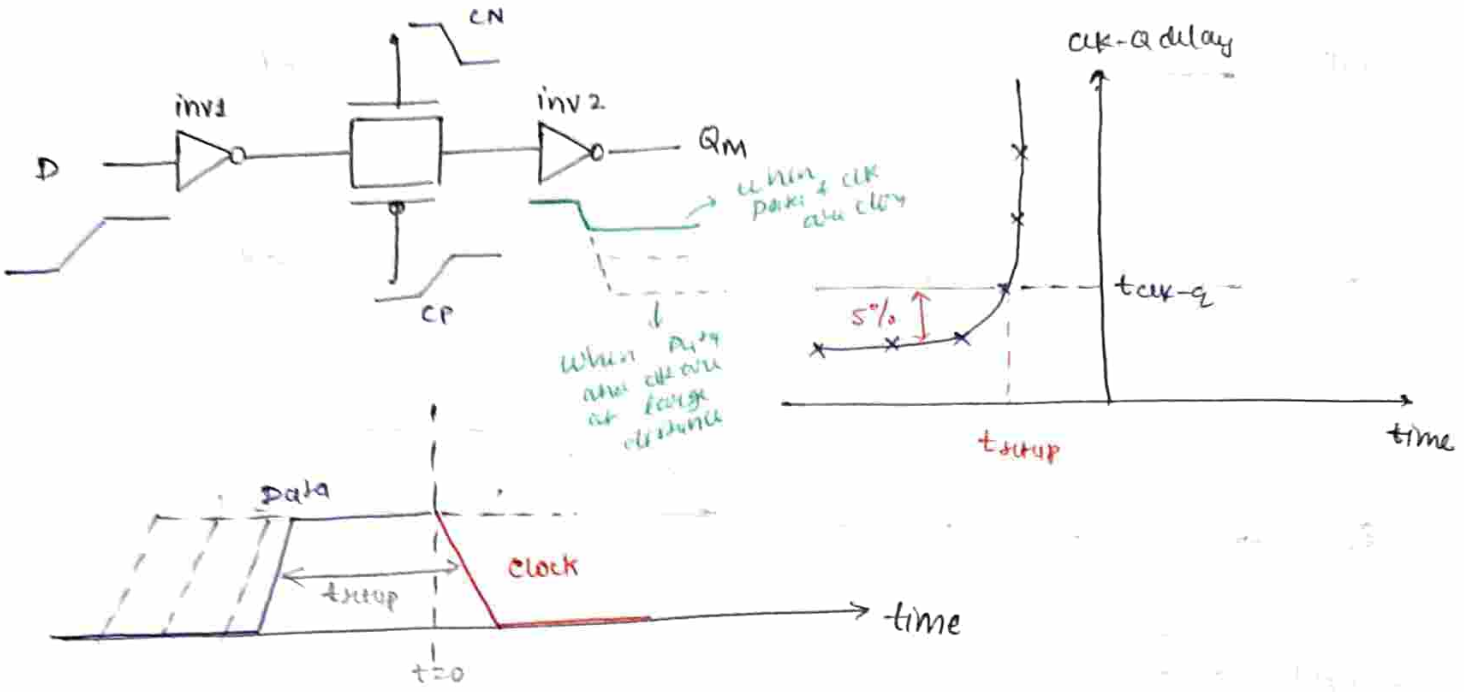


- As the edge of data comes closer to the clk-edge the clk-q delay starts increasing.
  — This is valid for both setup time & hold time.

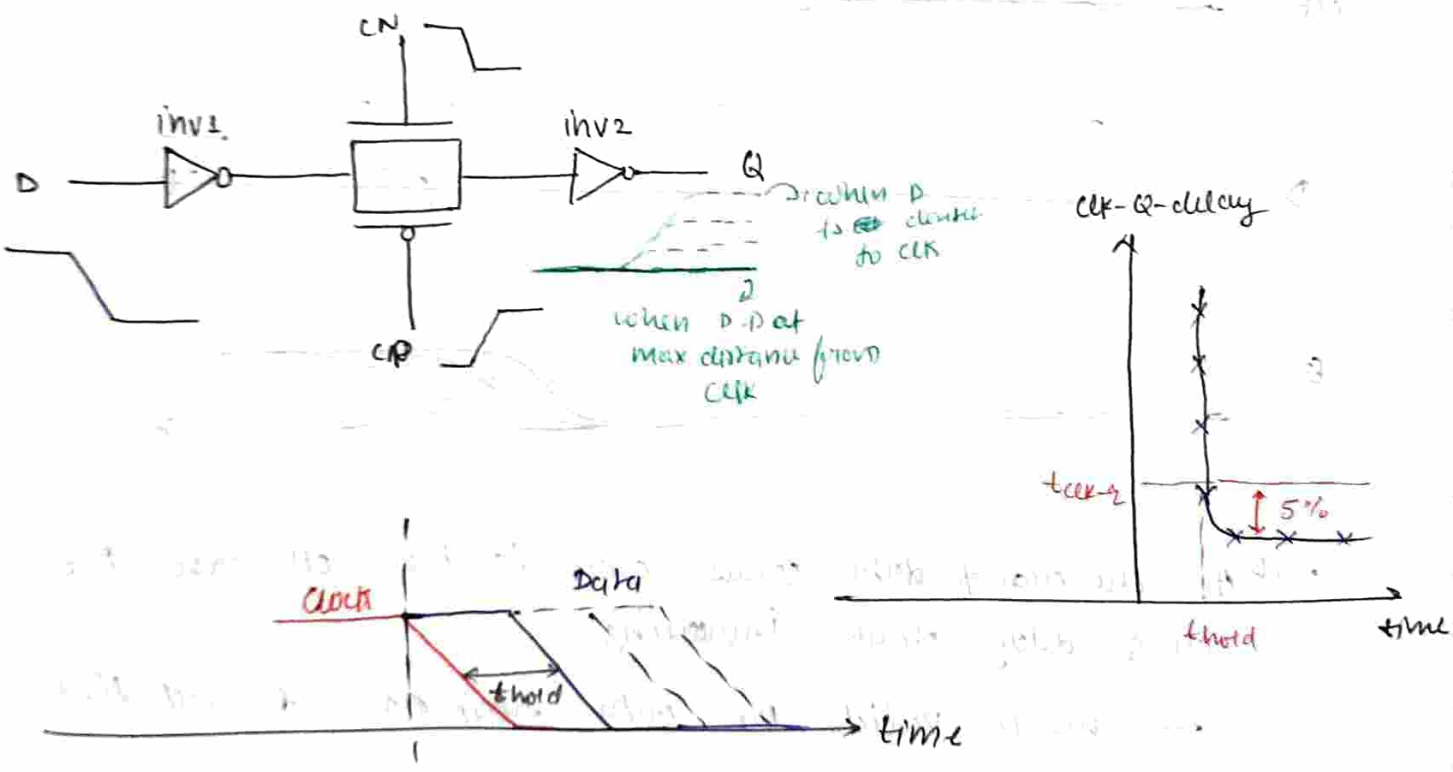
* When edge of D is far away from edge of clk then there is some t_{clk-q} delay which is least. Then on bringing D closer to edge of clk we see ↑ in t_{clk-q} when t_{clk-q} becomes 1.05 times its least value then at that time diff. b/w edge of D to edge of clk is called t_{setup}.



Setup-time illustration



Hold-time illustration

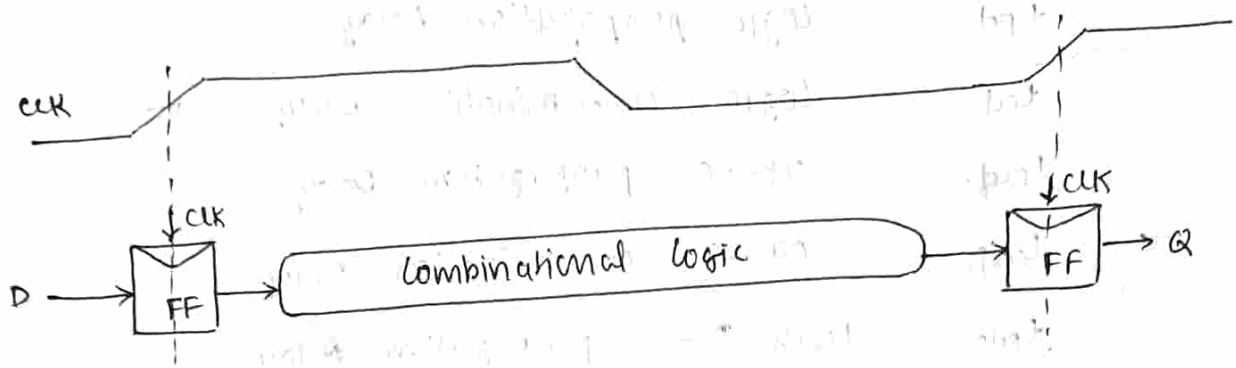


Q. As we are bringing Data edge closer to the clock edge then why clk-q delay increases?

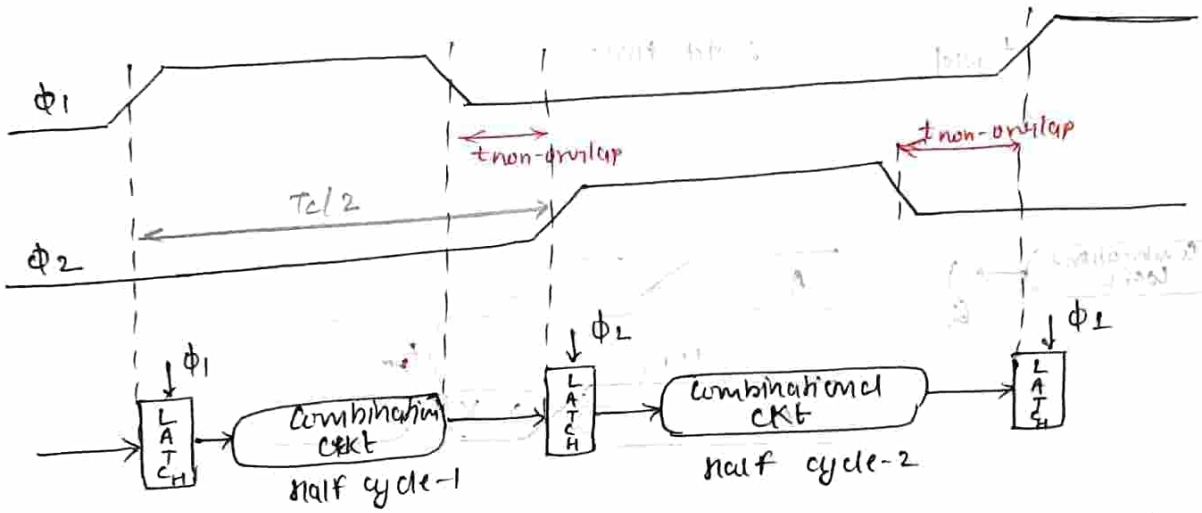
→ When we make our data to be stable for less time then there is possibility that some disturbance may go to Q hence it ~~decreases~~ ~~increases~~ degrades the slope of the transition on second side.

Sequencing Methods :

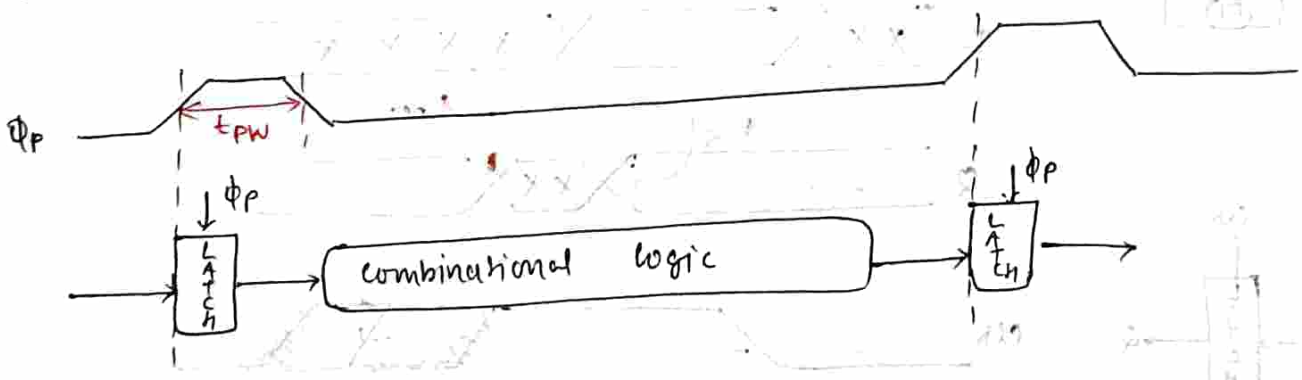
1. Flip - Flops



2. Two - phase Transparent Latch

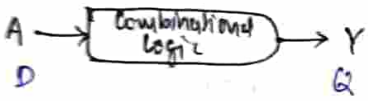


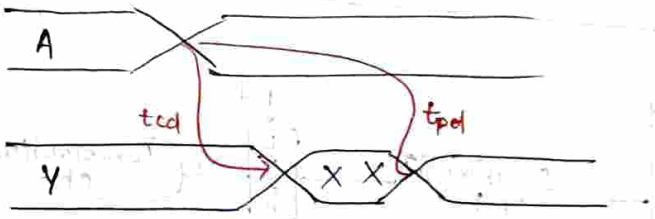
3. Pulsed Latches

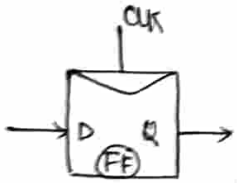


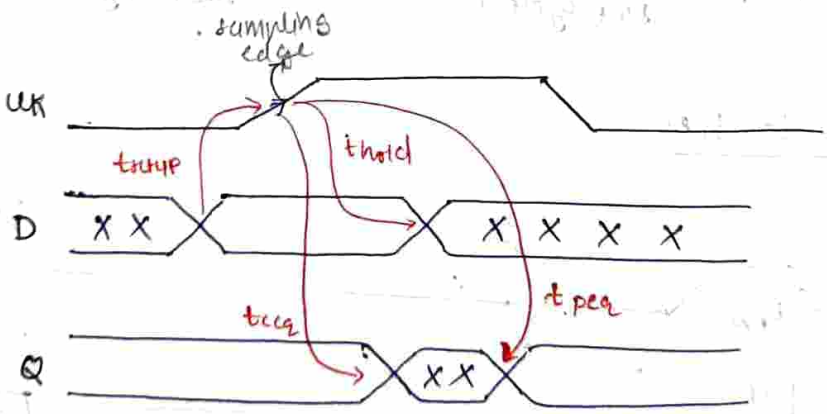
Timing Diagram

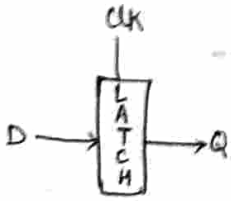
t_{pd}	Logic propagation Delay
t_{cd}	Logic contamination Delay
t_{pcq}	CLK → Q propagation Delay
t_{ccq}	CLK → Q contamination Delay
t_{pdq}	Latch D → Q propagation Delay
t_{cdq}	Latch D → Q contamination Delay
t_{setup}	Setup time
t_{hold}	Hold time

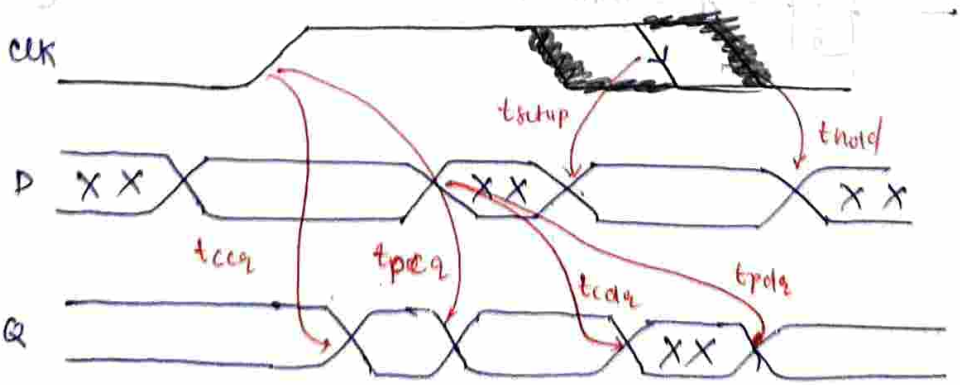








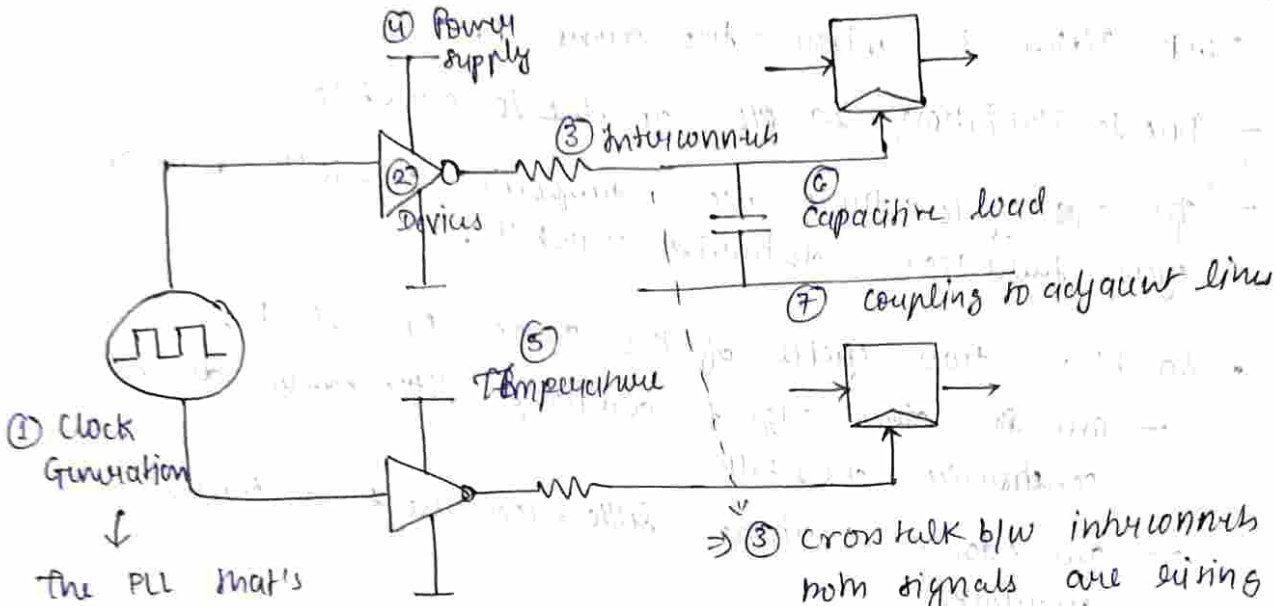




Timing Constraints

Clock Uncertainties :

· All of these are sources of uncertainty.



↓

The PLL that's generating the clock could have some jitter or differential delays

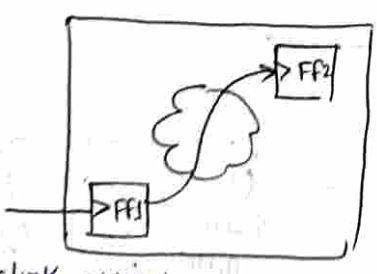
⇒ ③ Cross talk b/w interconnects when both signals are going different delays and one of them is taking a different delay [miller capacitance]

* One clock may experience delay via one path & other clk is experiencing delay through another path.
  — This is called clock uncertainty.
  · skew is because of the different routing for different clk path.
  · Jitter is because of temporal variations.

Clock Non-idealities :

1. Clock Skew :

· Spatial variation in temporary equivalent clock edges;
[deterministic + random = t_{SK}]



clk then arrives here at $t_0 + \delta$

If clock arrives here at t_0

* Clock skew is b/w 2 different ffs.
  — we are talking about spatial variations.

2. Clock Jitter :

- Temporal variation in consecutive edges of the clock signal; [modulation + random noise]

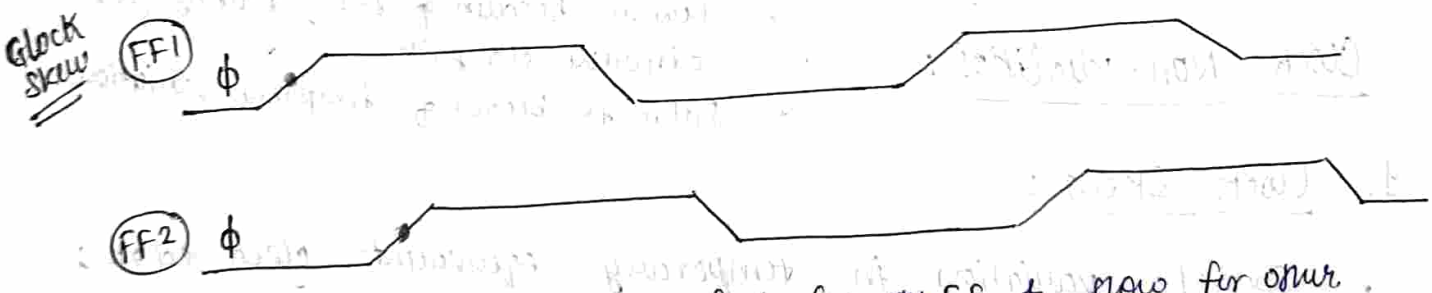
* clock Jitter ⇒ within the same ff.

- Due to variation in PLL or due to crosstalk.
- In one cycle there was supportive crosstalk & in one cycle there was destructive crosstalk.
- So b/w two cycles of the same FF itself,
  - one ~~⇒~~ ~~clock~~ edge ⇒ arriving little early due to constructive cross talk.
  - one edge ⇒ arriving little late due to destructive crosstalk.

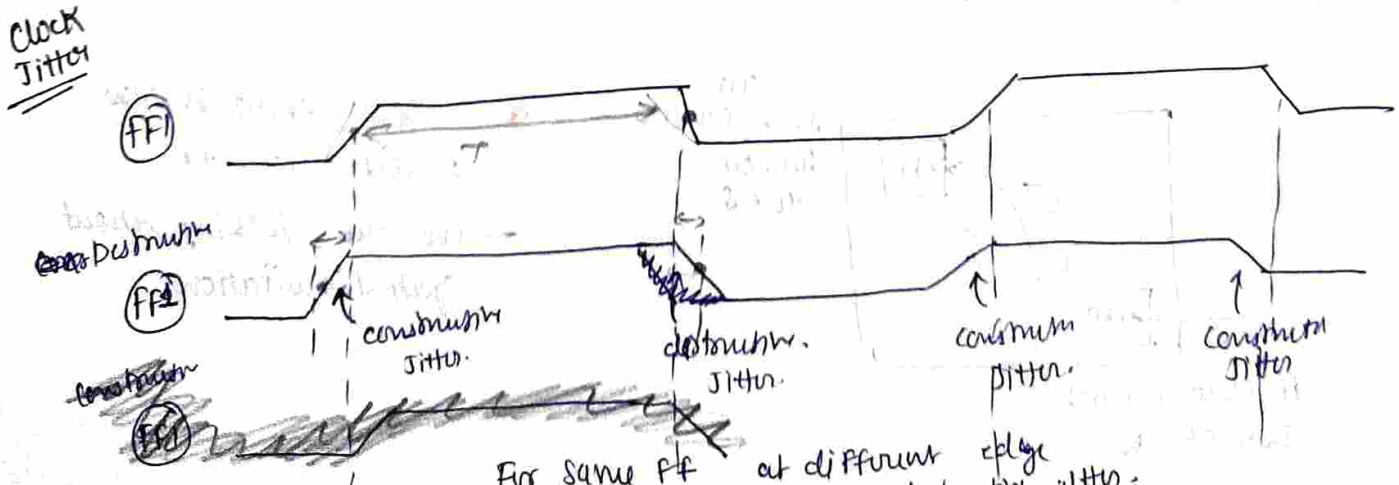
⇒ And this ⇒ called as jitter.

3. Variation of pulse width:

- Important for level sensitive clocking.



Same clock (ϕ) ⇒ fast for one FF & slow for other.



For same ff at different edge we observe constructive & destructive jitter.

- Skew is because of fixed + temporal variations
  Jitter is only because of temporal variation.

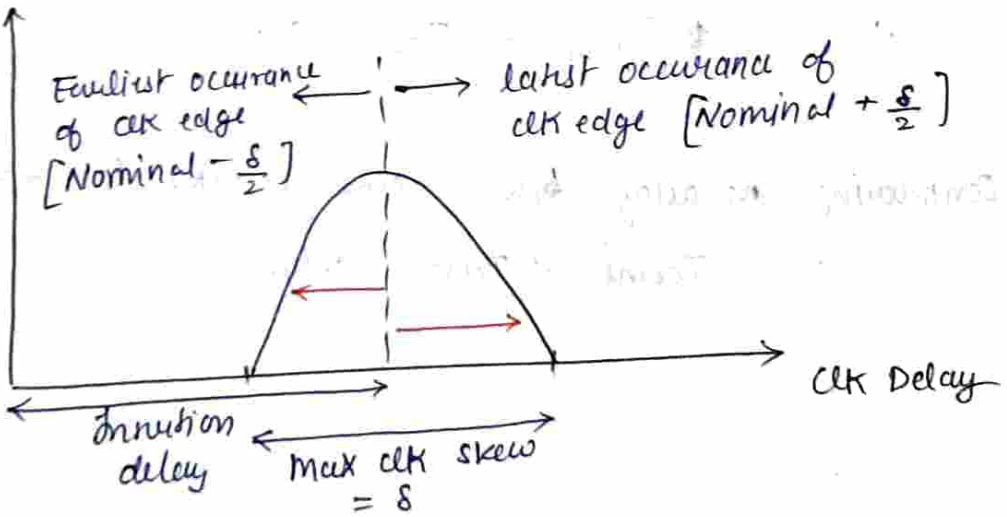
Temporal variations → Variation ~~due to~~ which are not fixed uncertain ; random.

- Temperature variation, power & device aging lead to ~~be~~ both jitter + skew ~~easier~~ clock.
- skew clock is also due to routing (ie for different path)
- skew clock is deterministic + jitter is indeterministic.
- ∴ Therefore for designing the ckt we will ~~look to~~ ~~the~~ leave the margin.

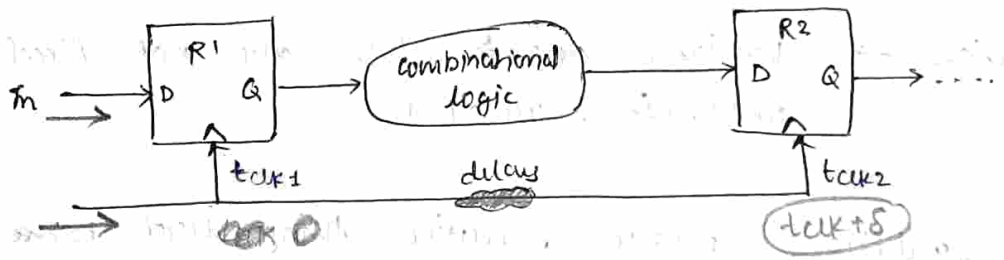
Q: A clk as i/p to the FF then there is some delay b/w the normal clk & inverted clk. Are hire the clock skew?

→ That delay will come in setup + hold time, because inverted clock is generated inside the FF itself.

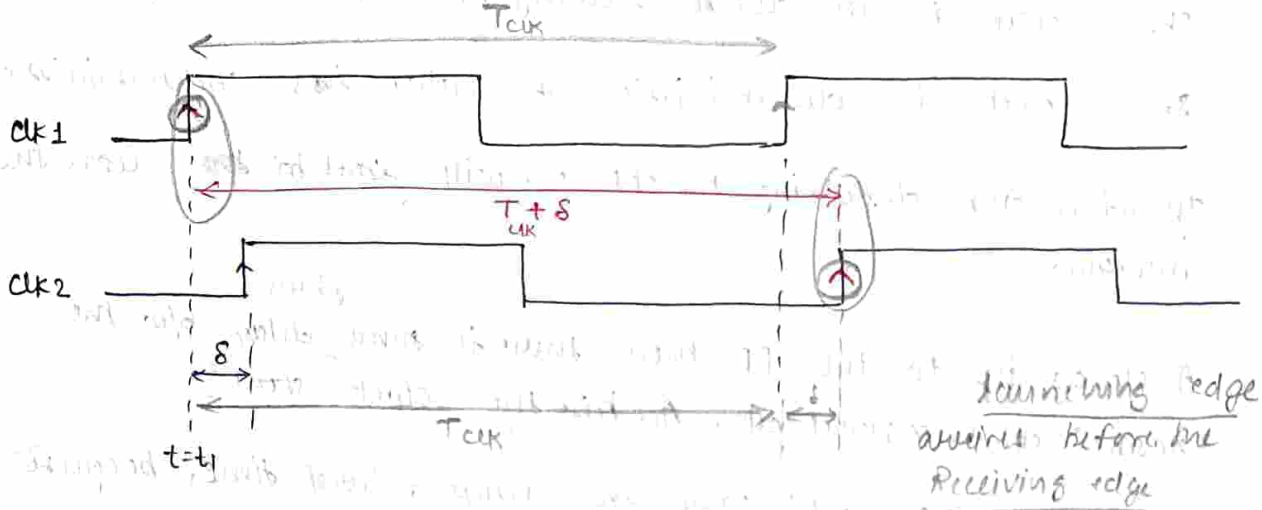
Clock skew :



(i) Positive skew



• If clk-1 arrives at $t = t_1$ at R1, then clk2 arrives at ~~t = ...~~ $t = t_1 + \delta$ at R_2.



• * clk1 ka +ve edge $t = t_1$ pe FF1 me pahucha
clk2 ka +ve edge $t = t_1 + (T_{clk} + \delta)$ pe FF2 me pahuchega

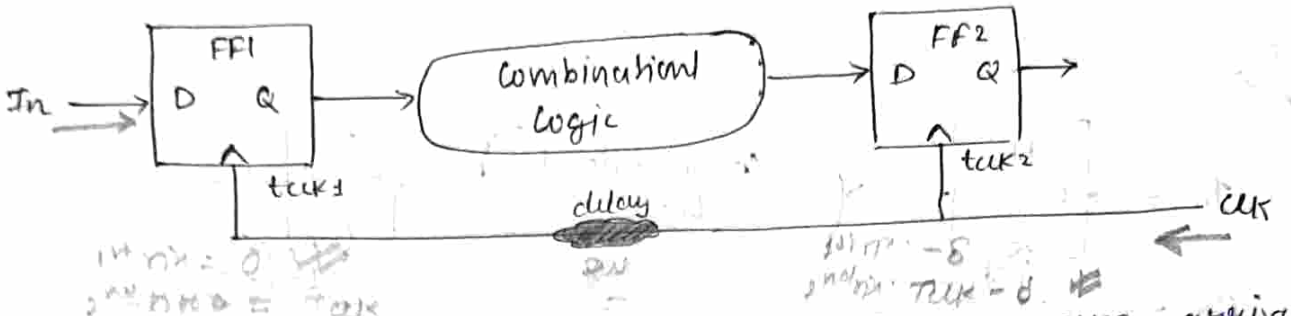
* Therefore we can place bigger combinational path b/w FF1 & FF2 whose delay should be less than $T_{clk} + \delta$.

$\therefore$ $T_{comb} < T_{clk} + \delta$

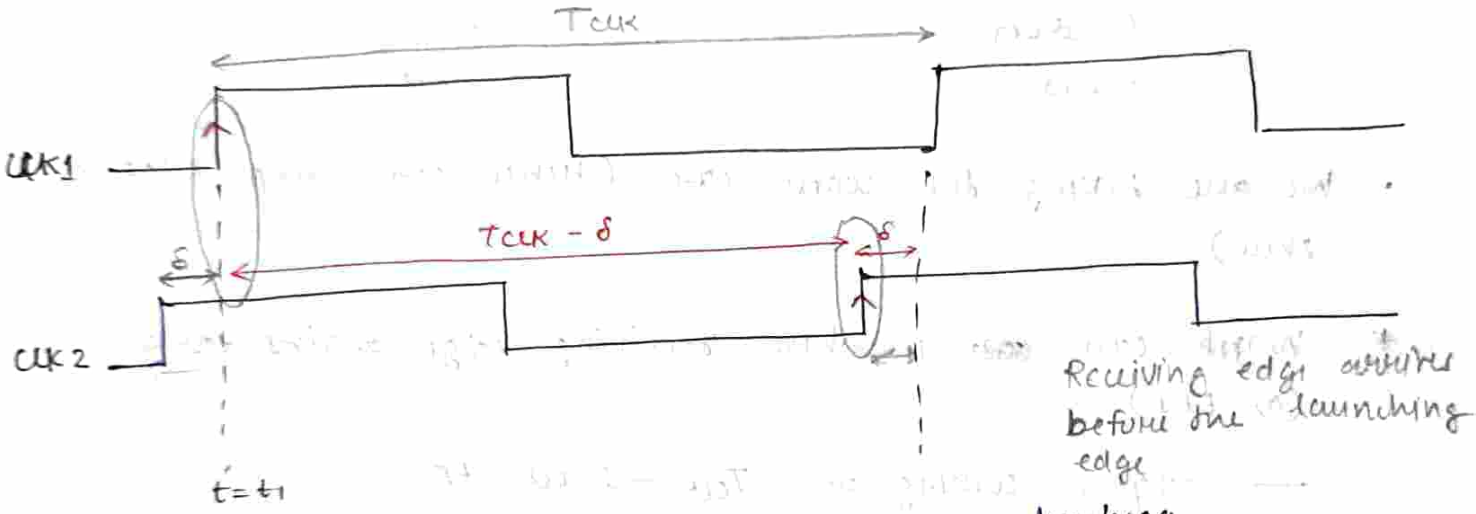
* Considering no delay b/w clk1 & clk2. (ie. $\delta = 0$)

$\therefore$ $T_{comb} < T_{clk}$ only.

(ii) Negative skew



- If clk1 arrives at $t = t_1$ at FF1 then clk2 arrives at $t = t_1 - \delta$ at FF2.



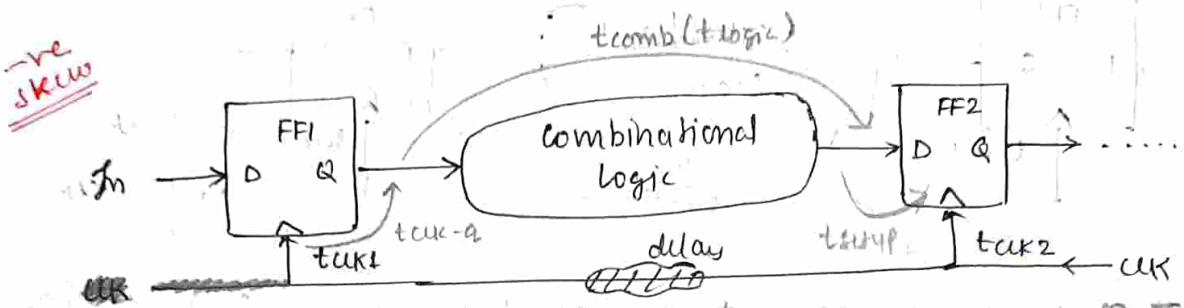
- clk1 ka 1st edge $t = t_1$ pe FF1 me pahuchega.
  clk2 ka 1st edge $t = t_1 + (T_{clk} - \delta)$ pe FF2 me pahuchega.

* Therefore we have to plan combinational logic & delay such that

$$T_{comb} < T_{clk} - \delta$$

Timing Constraint : [Considering only the worst case]

i) minimum cycle time :



① $T_{clk} - \delta$ ② T_{clk}

$t_{c-q,pd1}$
$t_{c-q,cd1}$
t_{su1}
t_{hold1}

$t_{logic,pd}$
$t_{logic,cd}$

$t_{c-q,pd2}$
$t_{c-q,cd2}$
t_{su2}
t_{hold2}

① T_{clk} ② $T_{clk} + \delta$

- We are taking the worst case (Hence considering Negative skew).

(*) Worst case ↑ when receiving edge arrives early (at FF1)

→ edge ↑ coming at $T_{clk} - \delta$ at FF1.

∴ this should be greater than worst case path delay.

∴ $T_{clk} - \delta \geq t_{c-q,pd} + t_{setup_2} + t_{logic}$

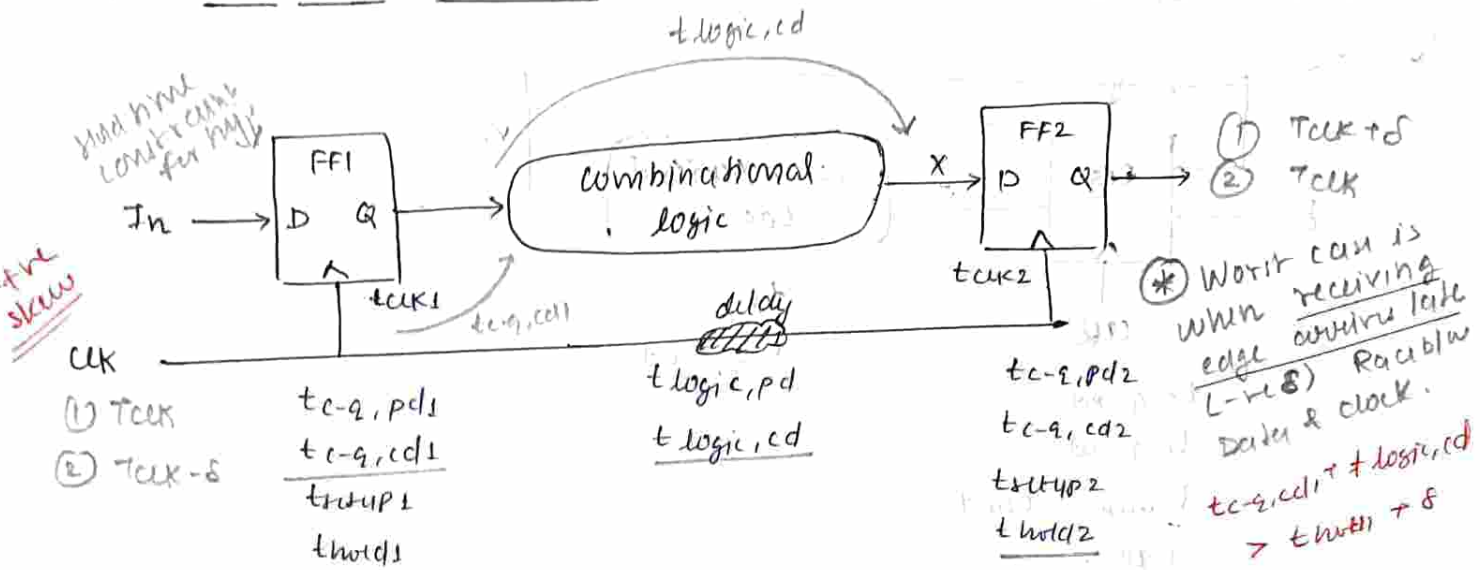
-ve skew [$T_{clk} \geq t_{setup_2} + t_{c-q,pd} + t_{logic} + \delta$]

max delay to reach FF2 & delay of clock ↑ also considered

+ve skew [$T_{clk} + \delta \geq t_{c-q,pd} + t_{logic} + t_{setup_2}$]

(ii) Hold time Constraint

* [$t_{path} + \delta > t_{hold}$]

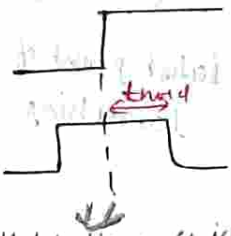


. ** To avoid hold time violation, we need to ensure that when the next clock edge comes (i.e. when clk2 comes) after that clko my earlier ilp must have remained stable for the duration of t_{hold} time.

∴ * the max. delay from 'In' to 'x' has to be greater than t_{hold} ie. [max delay se bhi jayada time ke liye usko hold karna hai dekhte ko] clk

$$t_{c-q,cd1} + t_{logic,cd} > t_{hold1} + \delta$$

(Inside) (Inside) (Inside D)



Hold time se jayada duration tak data stable rhna chahiye after clk edge comes.

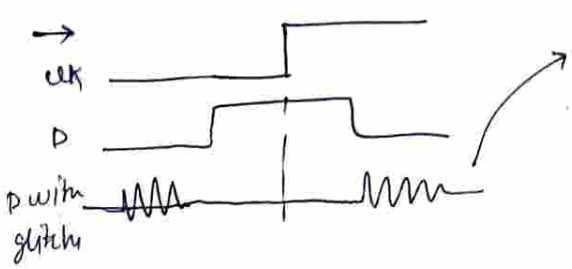
Note: (i) * For min cycle time, we are considering clock edges which are T_{clk} time apart

(ii) * For hold time constraint, we are considering same clock edges.

* FF1 me data tab tak hold karna hai jab tak hold time khatam nhi ho jata.

$$t_{cq,cd1} + t_{logic,cd} > t_{hold1}$$

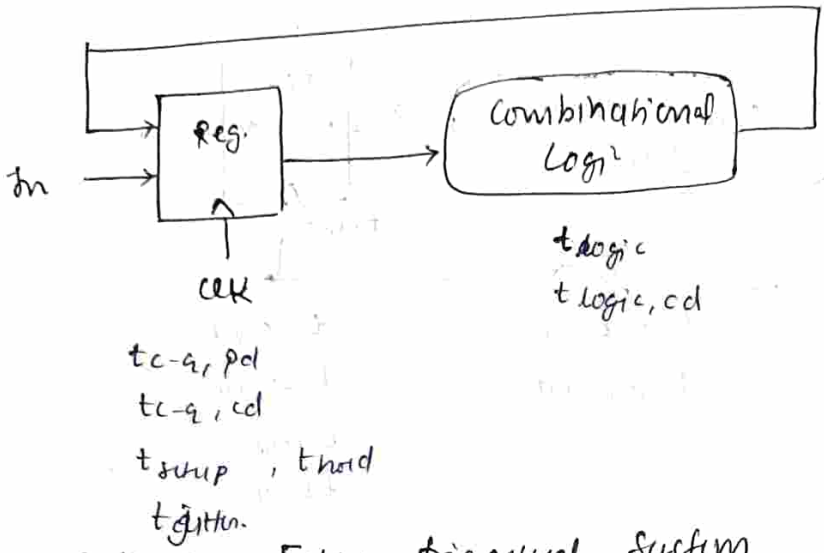
Q. Why we are taking contamination delay?



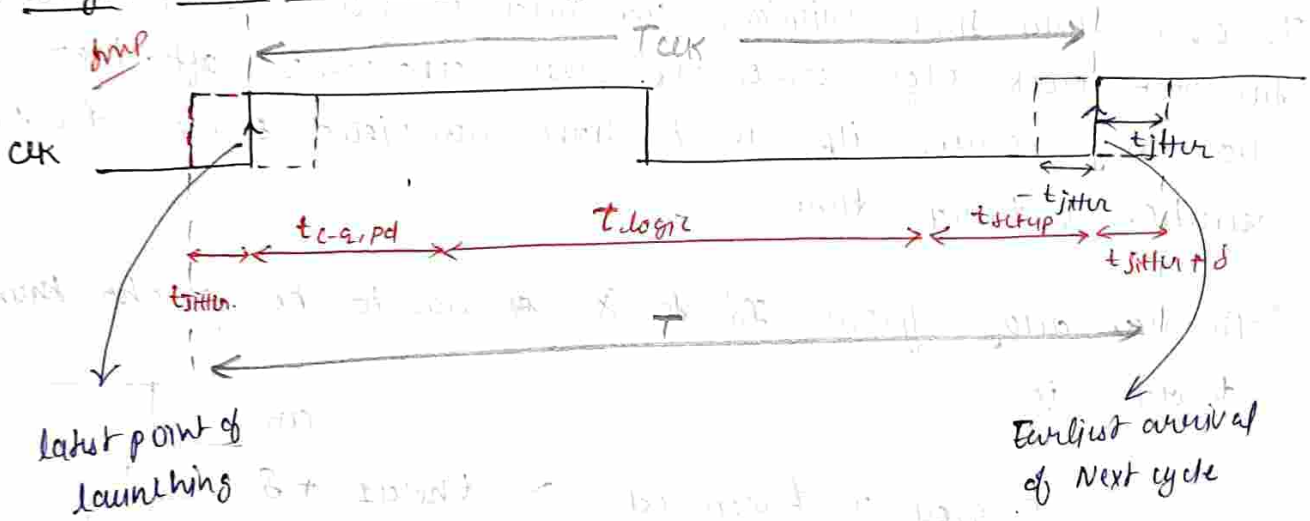
→ The first glitch should happen after the hold time

→ The first glitch should not arrive

Impact of Jitter



(i) Longest Path in Edge triggered System



(i) If launching edge is late & receiving edge is early, the data will not be too late if:

~~$t_{clk} + t_{logic} + t_{setup}$~~

$$\left[T - t_{ji,1} - t_{ji,2} - \delta > t_{c-q,pd} + t_{logic} + t_{setup}\right]$$

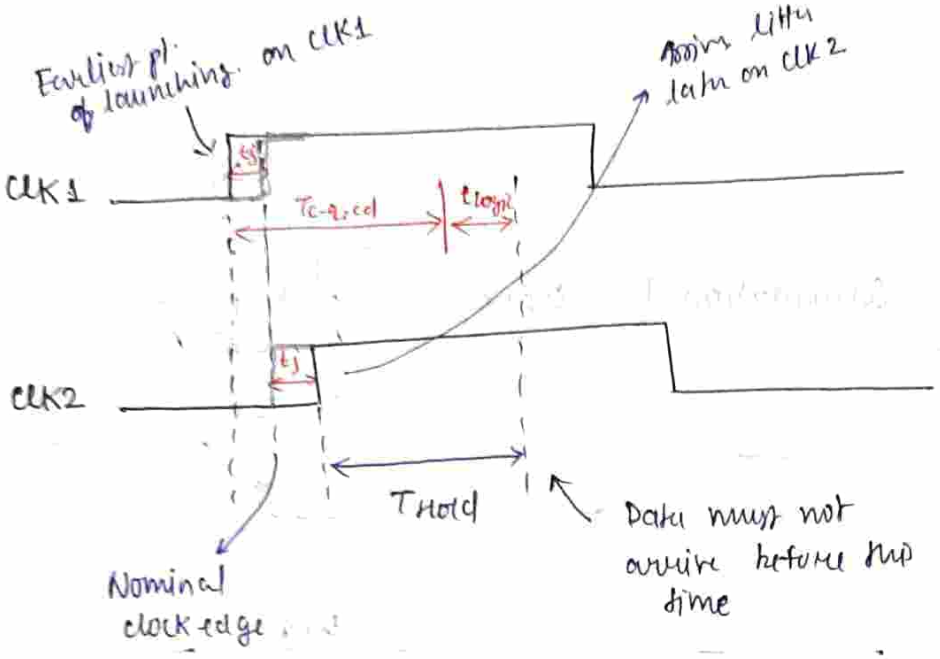
∴ minimum cycle is defined by the maximum delay through logic,

$$\left[T > t_{c-q,pd} + t_{logic} + t_{setup} + \delta + 2t_{Jitter}.\right]$$

↓

* skew can be either +ve or -ve.

(ii) Shortest Path



- If launching edge is early & receiving edge is late:

Data must not arrive before t_{hold} + jitter.

$$t_{c-q,cd} + t_{logic} - t_{j1} > t_{hold} + t_{j1+2} + \delta \quad \therefore$$

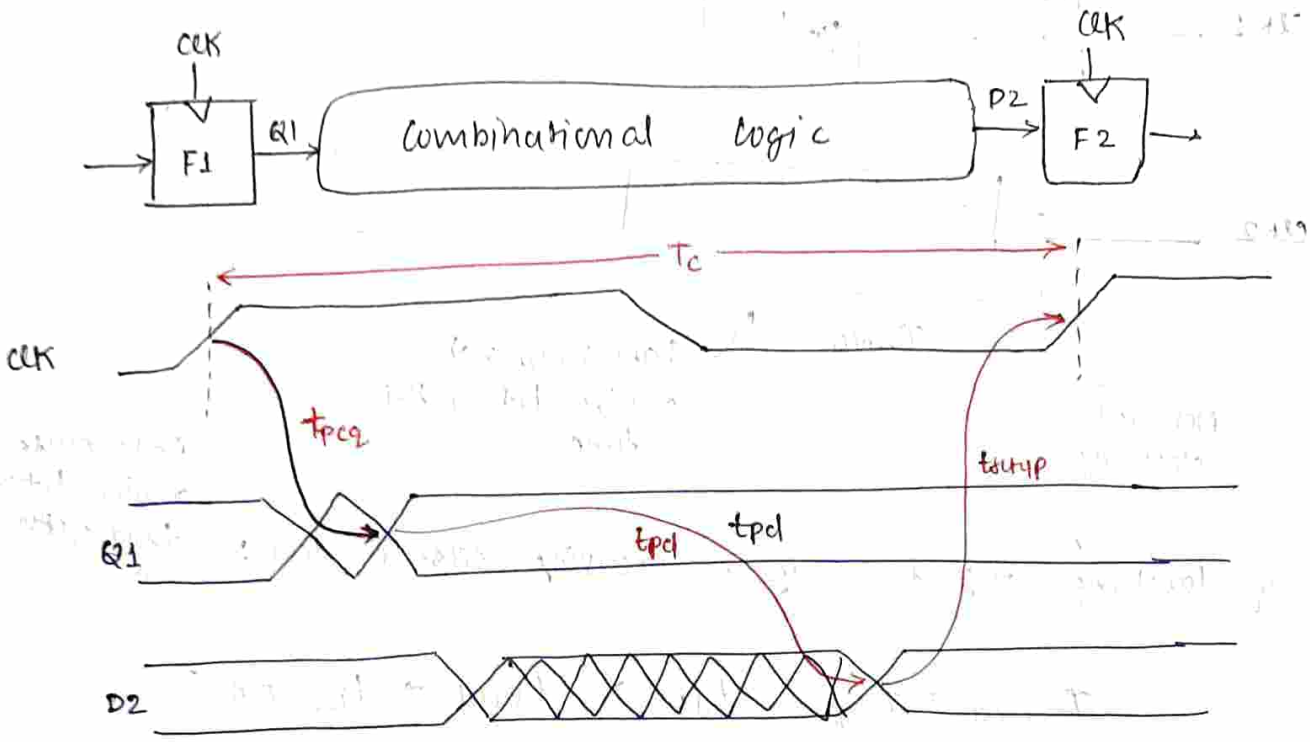
$\therefore$ Minimum logic delay,

$$t_{c-q,cd} + t_{logic} - 2t_{j1} - \delta > t_{hold}$$

$\Rightarrow$

Maximum Delay : [Neglecting skew & Jitter]

1. Flip-Flops :



$$t_{pd} \leq T_c - [(t_{setup} + t_{pcq}) + t_{skew}]$$

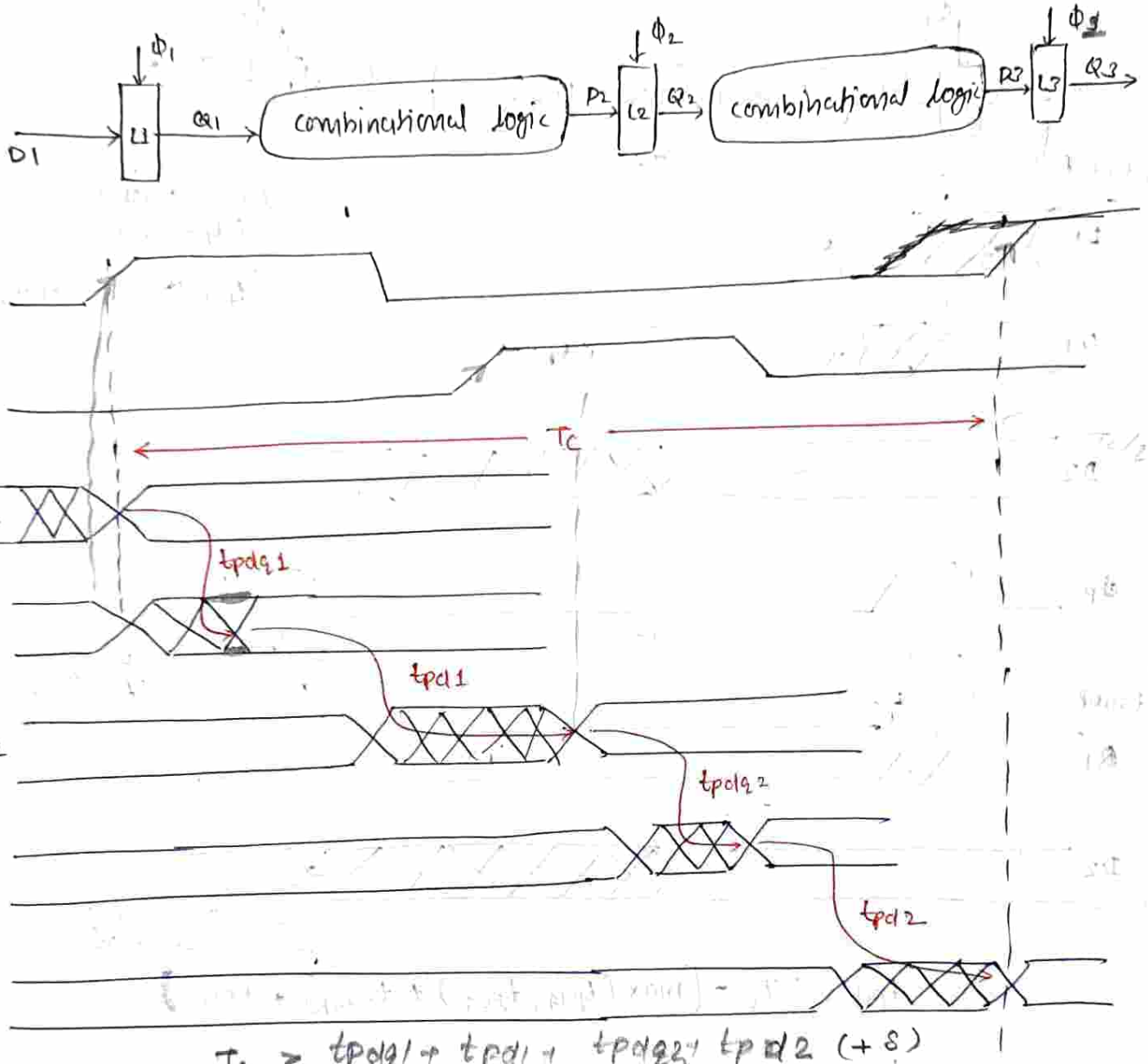
↓

sequencing overhead.

- If FFs are not inserted in the combinational path then only t_{pd} will be the total delay.
- But now due to FF we have t_{setup} & t_{c-q} delay introduced in the path.

$$T_c \geq t_{pcq} + t_{pd} + t_{setup} + \delta$$

2. ✱ Two-phased latches



$$T_c \geq t_{pdq1} + t_{pd1} + t_{pdq2} + t_{pd2} \; (+ \delta)$$

$$t_{pd} = t_{pd1} + t_{pd2} \leq T_c - (t_{pdq1} + t_{pdq2})$$

$$t_{pd1} + t_{pd2} \leq T_c - (2\, t_{pdq}) \quad \leftarrow \text{sequencing overhead}$$

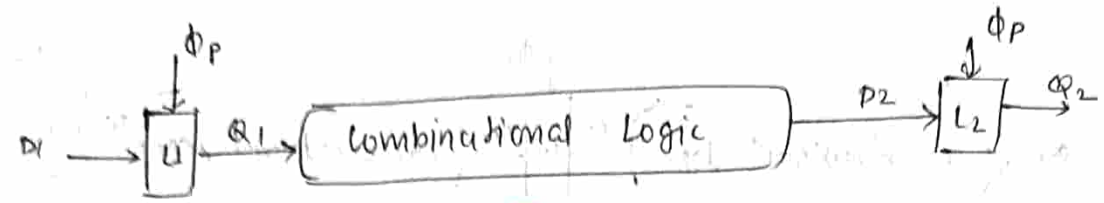
$\frac{T_c}{2}$ - 4ns

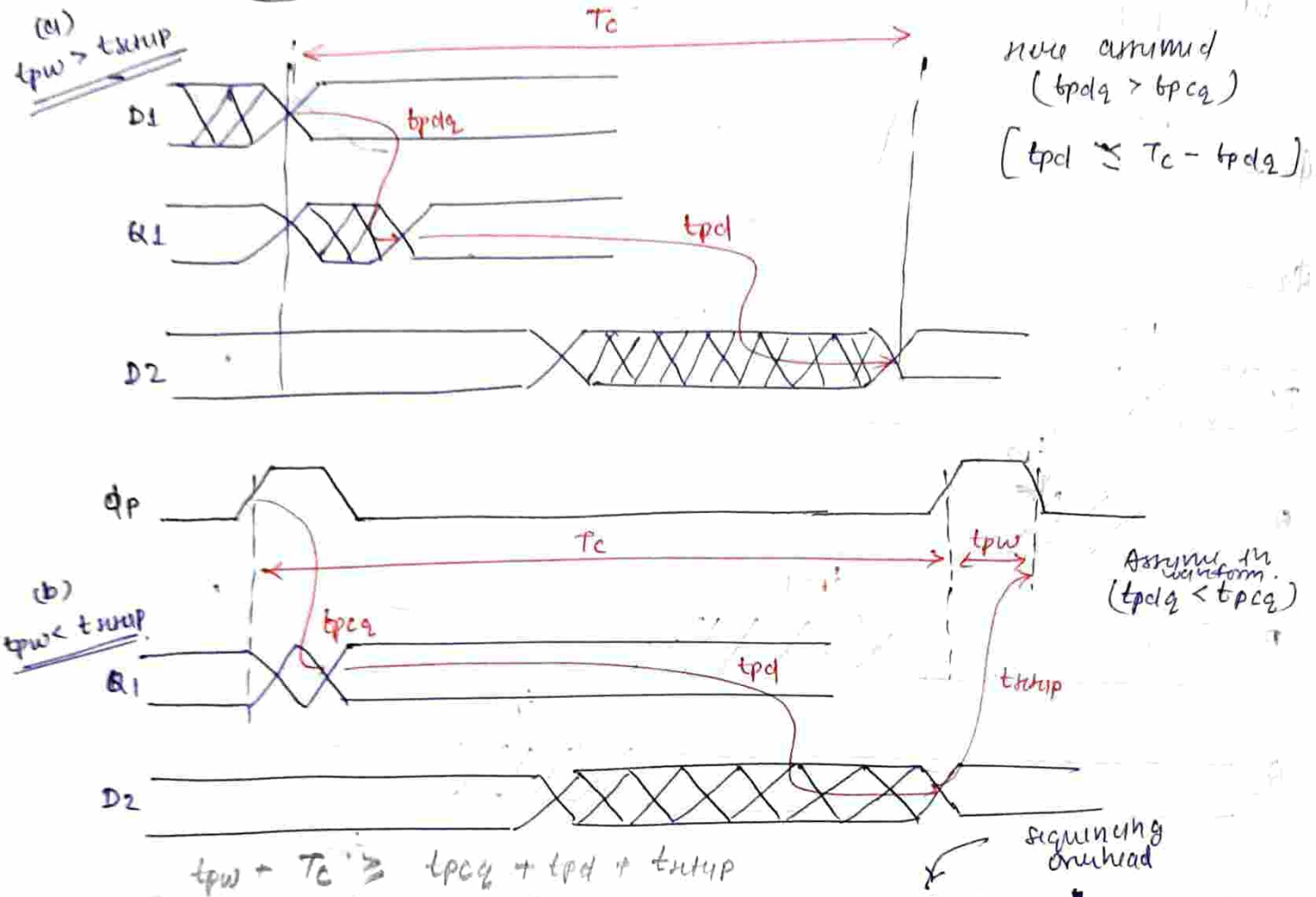
✱ We are not considering the setup time here because setup time is taken at sampling edge and for the level latch it is the falling edge

Q. Its total delay be more longer if we consider the setup time also?

→ It will be discussed in time borrowing.

3. Pulsed Latches





$t_{pw} + T_c \geq t_{pcq} + t_{pd} + t_{setup}$

sequencing overhead

$$t_{pd} \leq T_c - [\max(t_{pdq}, t_{pcq}) + t_{setup}] + t_{pw}$$

$$t_{pd} \leq T_c + t_{pw} - [\max(t_{pdq}, t_{pcq}) + t_{setup}]$$

- When, $t_{pw} >> t_{setup}$ then we ignore the setup time.
- when, $t_{pw} << t_{setup}$ then we can't ignore the setup time. and we have to consider setup time in max delay.
- For a latch maximum of D-q delay or clk-q delay is considered.

Q. ? * When $t_{pw} > t_{setup}$ then why are not we considering the setup time ?

Q: Why we are subtracting the t_{pw} from the sequencing overhead.?

→ In a latch you launched your data from rising edge & sample it at falling edge because setup time is from falling edge.

- But time period is from rising-to-rising edge of clk.
so the t_{pw} has to be reduced from the sequencing overhead.

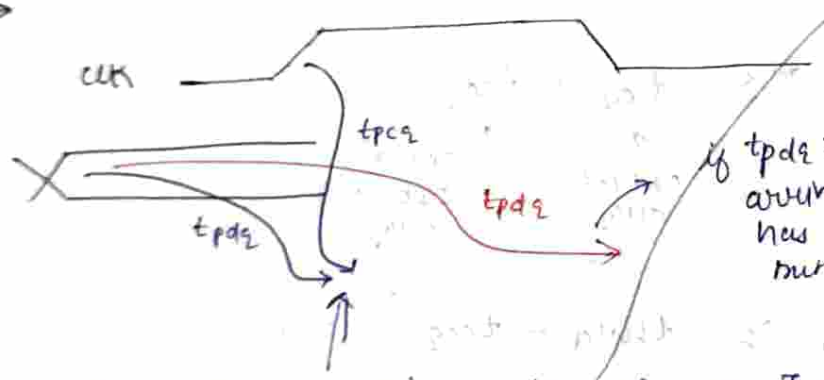
* $T_c + t_{pw} > t_{pd} + \max[t_{pdq}, t_{pcq}] + t_{setup}$

* Timing constraint window is $T_c + t_{pw}$
But my time period is only T_c.

* case(i) & (ii) both same $e2^n$.
* max (t_{pdq} & t_{pcq}) will be taken.

Que * Why max. of t_{pdq} & t_{pcq}?

→



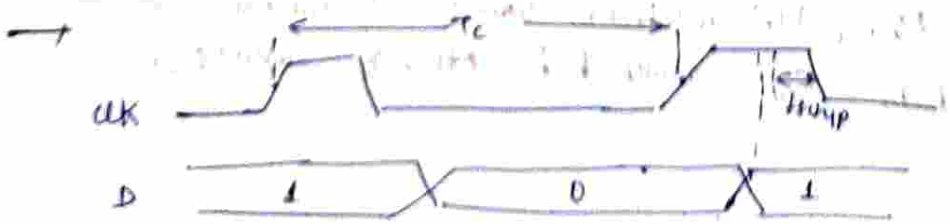
Earliest Q will arrive after T_{pcq} or T_{pdq}

If $t_{pdq} > t_{pcq}$ then data will arrive after t_{pdq} (80 clk edge has come after ~~tp~~ data ~~edge~~ toggling but even though t_{pdq} is the defining delay.

(*) For this can the toggling & would be due to clk edge but the delay would be due to t_{pdq} because. ($t_{pdq} > t_{pcq}$).

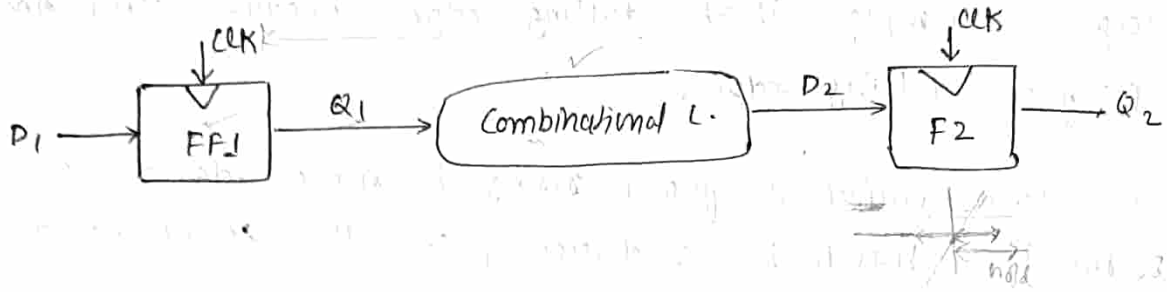
Q If pulse width (t_{pw}) is large enough & data is coming at the middle of the PW & setup time is not violating so the difference b/w the clock edge has come & data is toggled, then wouldn't that difference be taken in our timing relation?

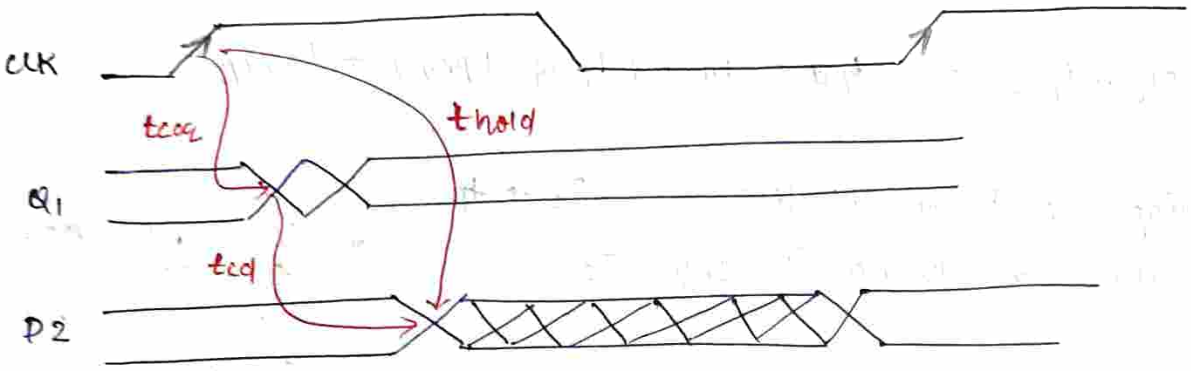
→



Minimum Delay ? :-

1. Flip-Flop





- D2 should hold its value for 'thold' time.
- And $t_{hold} \leq t_{cd} + t_{ccq}$

(t_{hold}: (why -ve sign?) ; t_{cd}: critical delay ; t_{ccq}: critical clock to q delay ; considering +ve skew)

$$\therefore \{ t_{cd} \geq t_{hold} - t_{ccq} \} + \delta$$

(δ → t_{skew})

for 2 and lakh

2). The ϕ_1 triggered o/p should reach L_2 after hold time requirement has meet. (ϕ → Q_2 or P_2)

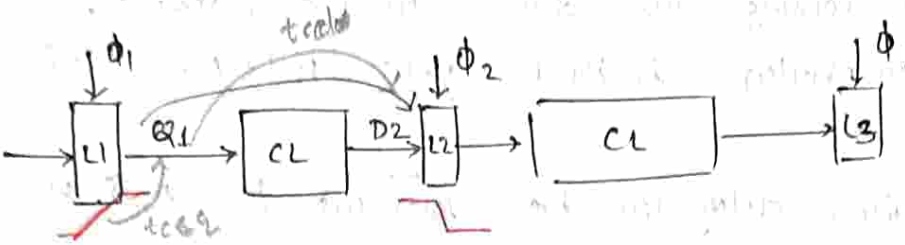
— And there is some non-overlapping window b/w ϕ_1 & ϕ_2.

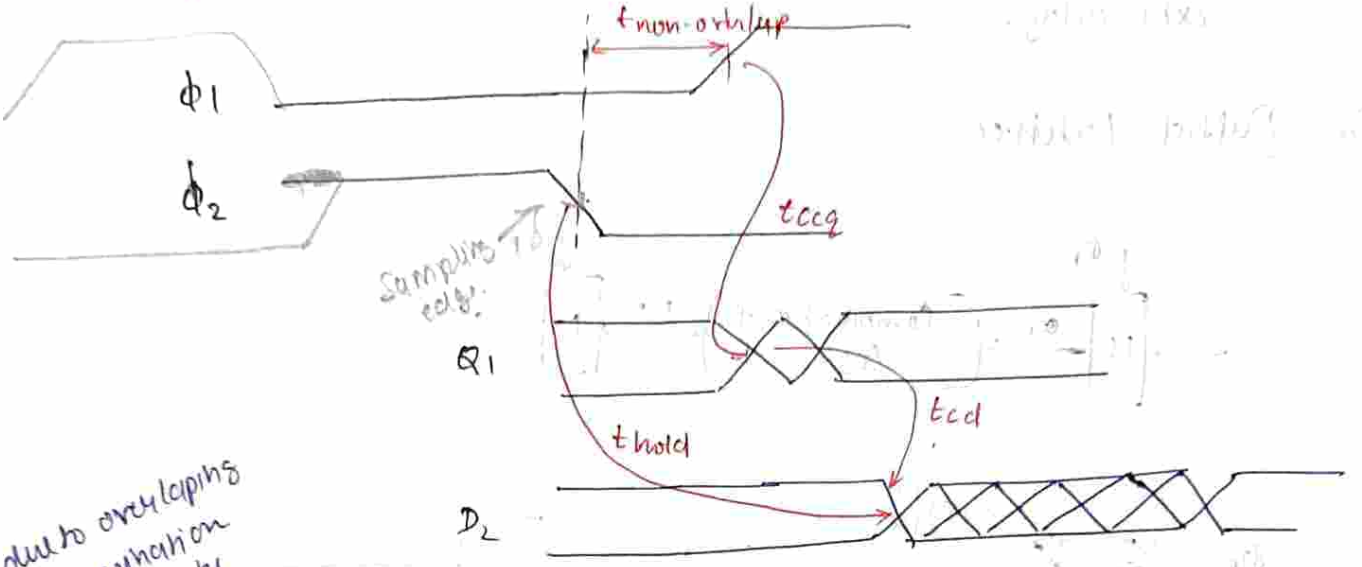
$$t_{cd} \geq t_{hold} - t_{ccq} - t_{non\text{-}overlap} + \delta$$

Note: For taking skew, Add 'δ' in 'thold', kyki hum chahte hai ki thold kiski minimum value se bhi minimum rahe [$t_{hold} < t_{path} - \delta$ would be the effect of skewing].

2. Two-phase Latches

VV Imp





due to overlapping contamination delay can be ~~small~~ smaller

$$t_{ccd1}, t_{ccd2} \geq t_{hold} - t_{ccq} - t_{non\text{-}overlap}$$

(Q_1)

- We want ϕ_1 triggered ~~@~~ o/p should reach L_2 after hold time requirement has met.

Q. Why ϕ_1 rising edge ↑ after ϕ_2 falling edge? How will I consider the hold time requirement of L_2?

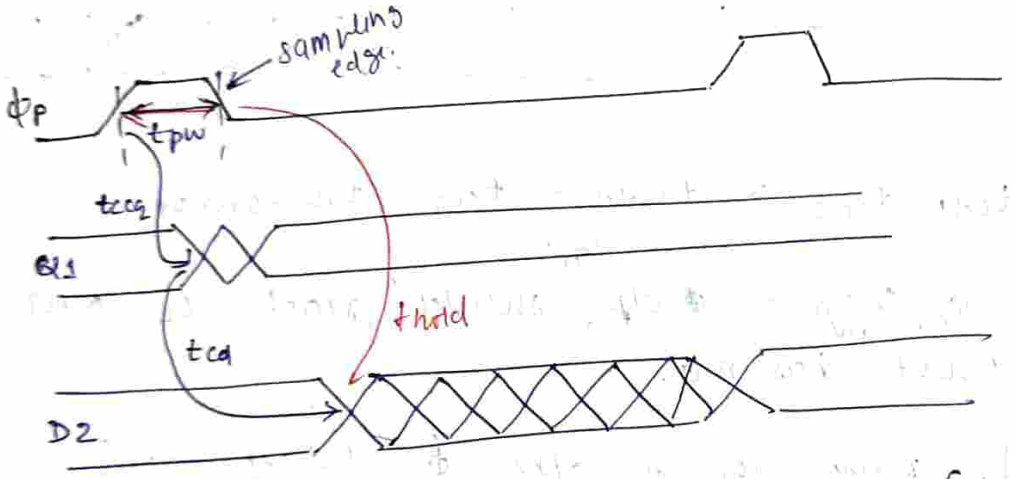
→ when L_2 sampling edge is closed, now we want i/p of L_2 to be stable and for that there should not be any rising edge of ϕ_1 (which would change the value of i/p of L_2).

Therefore now late the rising edge of ϕ_1 is coming, we are ~~in~~ insured on that thing.

After sampling edge of ϕ_2 it will stop sampling but it should ~~be~~ we have to hold the data for some time to meet the hold time requirement.

* You can't change non-overlap time period of the clk.
  - We may be loosing data due to non-overlapping.
* Benefit of non-overlap is that your hold time failure will not happen.
  - contamination delay can be reduced to min. non-overlap extra delay.

3. Pulsed Latches



(on ↑ tpw, tcd ↑)

$$t_{cd} \geq t_{hold} - t_{ccq} + t_{pw} + \delta$$

- For entire duration of pulse, no change in i/p over here.
- Data is triggered from +ve edge of ϕ_P.
  And hold time is defined from -ve edge of ϕ_P.

* We characterize setup & hold time at the original clock of the system.
  → Here we have pulsed clock in our system

Q: What is the requirement for L_2?

→ Data should be stable for t_{hold} duration after the falling edge of the clk.

- And o/p is triggered at rising edge of my clk at L_1 →

$$\therefore \quad t_{ccq} + t_{cd} \geq t_{hold} + t_{pw}$$

because t_{hold} is coming from the falling edge of ϕ_P.
[as falling edge of ϕ_P is sampling edge]

* Here the system setup-time is more relaxed & hold time very tight, so they very risky because you can't recover hold time constraint
— therefore they are not very commonly used.

Q: Since we decrease t_{pw} to get better hold-time requirement. Then how much t_{pw} can be?

→ Impulse & system behaves as FF.

Q: Does $t_{pw} > t_{ccq}$? Is it necessary?

→ There is no connection b/w them

Time Borrowing (Weste Pg: 387)

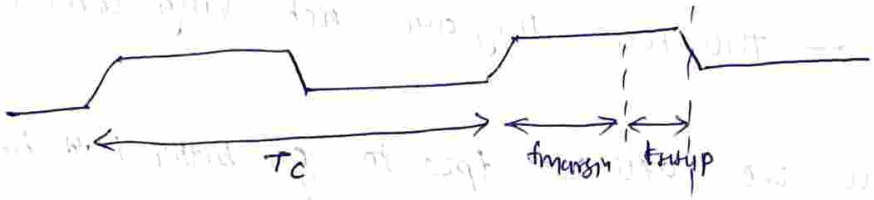
In an FF based system:

- Data launches on the rising edge
- must setup before the next rising edge

  - If it arrives late, system fails
  - If it arrives early, time is wasted.

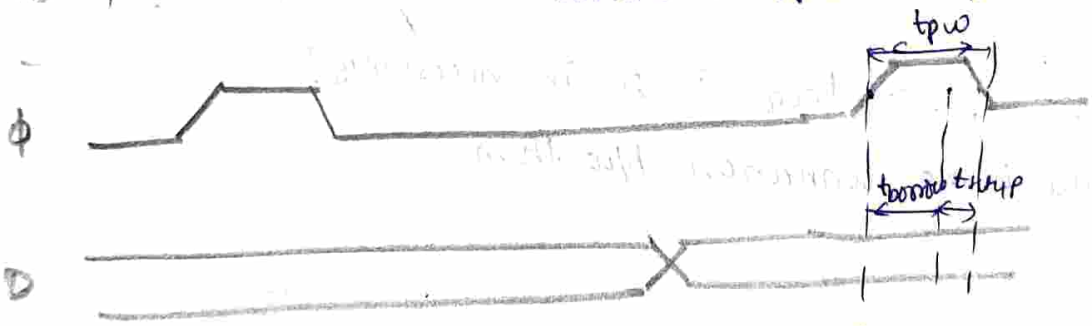
In latch based system:

- Data can pass through while it is transparent it does not limit itself for any rising edge.
- long cycle of logic can borrow time into next.



(i) two-phase latch $t_{borrow} \leq \frac{T_c}{2} - (t_{setup} + t_{non\text{-}overlap})$

(ii) pulse latch $t_{borrow} \leq t_{pw} - t_{setup}$

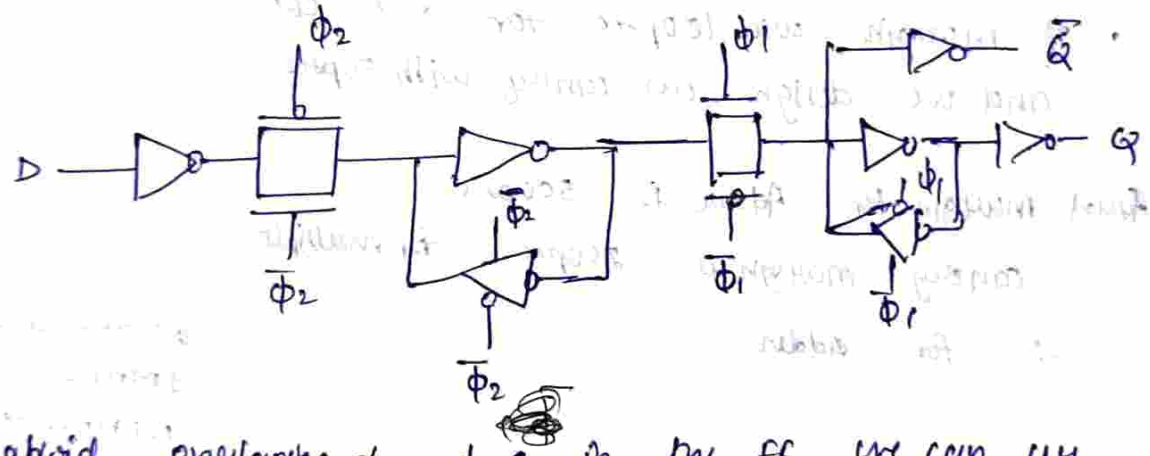


$t_{borrow} \leq t_{pw} - t_{setup}$.

* If setup time is getting violated you reduce the clock speed. (increase T_{clk})
* If hold times are violated, chips for any speed. (ie, you can't improve this by changing T_{clk} you have to change your ckt) → But with two phase latches you can avoid hold time violations.
  - To get working chip is the most imp requirement

** A way to guarantee hold time is to use 2-phase latches with big non-overlap time

Safe Flip-Flop:



- To avoid overlaping of ϕ & $\bar{\phi}$ in the FF we can use two different ϕ_1 & ϕ_2 within the same FF.
- When we design the FF with non-overlapping clocks
  - it is robust in the presence of hold time failure because there is no overlap
  - It will slow downs the system, because it gets added to the setup requirement.

Adaptive Scaling:

~~Designers include~~

- If a silicon is manufactured at typical lot.
  - then we have margin for SS lot for the setup time.
  - then we also have margin for FF lot for the hold time.

* your designed system can operate at 600MHz but you can operate at 450 MHz while taking the margin.

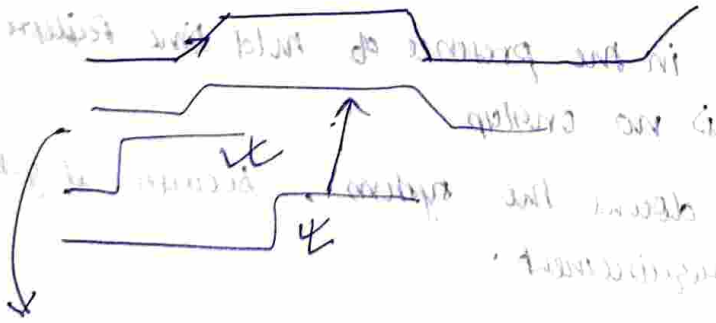
Eg. of Adder and Multiplier. (11:50 min)

- If margin was 100psec for our clk and we design our canary with 50psec

Actual margin for Adder is 500psec
canary margin is 200psec for multiplier

∴ for adder

#Canary
#Razor



0.9 MHz
1 MHz.
1.1 MHz →
1.05 → FF

Memory Design

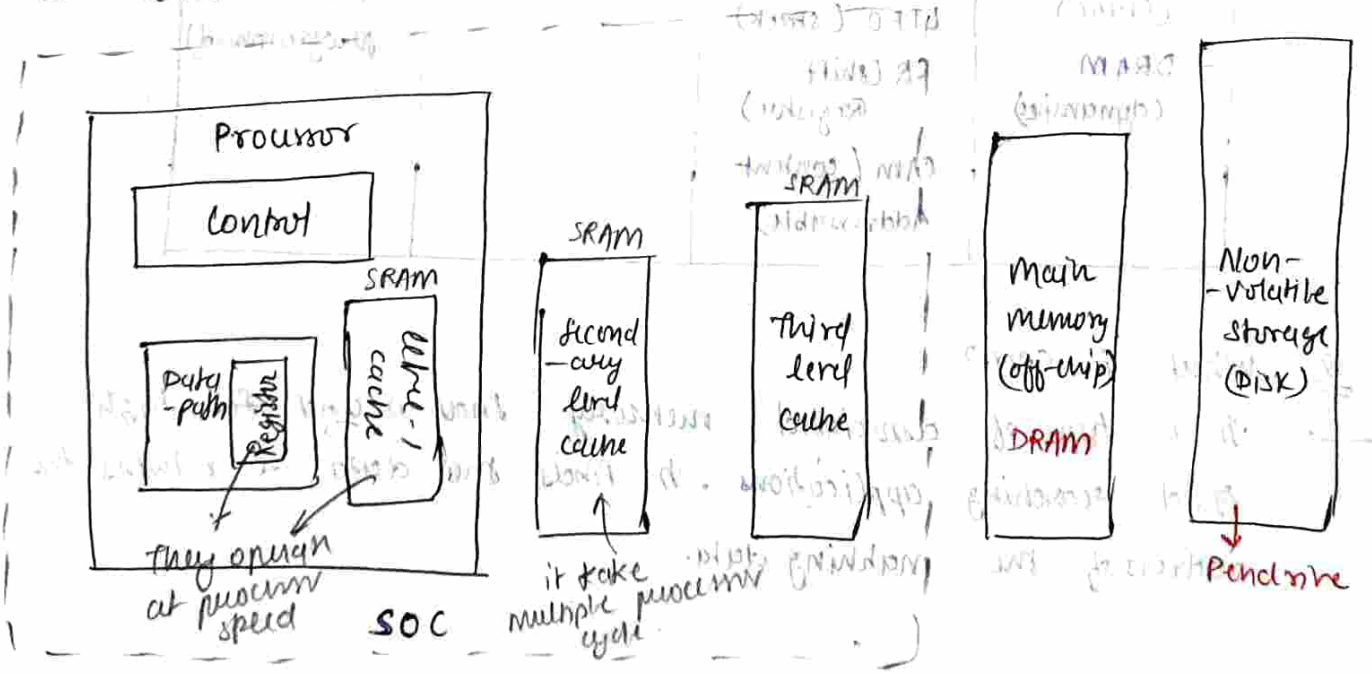
- Memory are high density storage element.
- Memories are minimalistic storage elements which provides high density. As we take FF, register or latch most involves large no. of transistors even though they provide storage functionality but they don't provide high package density. Hence we use memory models. [FF takes more area as compared to standard - 6T memory cell]
- Memory cover 70-80% of chip area in advanced SOCs.

Q: Why so much of area is given to memories?

→ Because in typical computer architecture, memories are there at various levels.

Memory Hierarchy: * memory management

* Typical system has multiple level of storage systems.
  1. Very fast registers in the Data path.
  2. Processor speed driven L1-cache
  3. Slower but dense on-chip secondary (& tertiary) memories.
  4. Off-chip, but on-board storage with latency of 100s of cycle.
  5. Hard disk



* As we move away from processor they become slower.

Classification of Memories

- Read-Write memory (RW memory) → memory in which you can read the data from memory & write data to memory.
- Read only memory (ROM) → memory in which data can't be written & we can only read data from it.
- Non-Volatile RW memory (NVRWM) → Memory that can retain stored information even after power is removed.
- Volatile memory → memory that don't retain stored information when power is removed.
- Random Access memory (RAM) → RAM is a computer's short term memory, where the data that the processor is currently using is stored.

RW memory		NVRWM	ROM
Random Access	Non-Random Access	EPROM EEPROM FLASH	Mask Programmed PROM (Fuse programmed)
- SRAM (static) - DRAM (dynamic)	FIFO (queue) LIFO (stack) SR (Shift Register) CAM (content Addressable)		

Volatile

Q. What is CAM?

→ It is type of associated memory that is used for high speed searching applications. It finds the data & returns the address of the matching data.

- Typically in memory, you give address location & you would like to find out what is written on particular address.

* In CAM, depending on the data, you get a different address to send your data.

  → CAM are tightly coupled to processors, so that when particular instruction is given, the follow-up instructions are brought up quickly based on ~~the~~ ~~content~~ by doing the quick content search, as to which location is this particular instruction stored & then I pull out remaining instructions from that region quickly.

* EEPROMs are used in Boot sequence.
  — Also used in setup Box of Airtel, Tatasky.

* Flash are used in memory card, cell-phones.
  - All solid state Devices (SSD) are flash.
    (Storage)

* ROM are used in Boot sequence.
  → Once it is programmed you can't change it.

Feature comparison in memory Type

- speed
  ON-chip memories typically SRAMs, whenever we talk about L1-cache, L2-cache, L3-cache. They are very fast but they are not very dense.
  — i.e, 6-transistors per SRAM.

Q. What is 6T-SRAM cells?
→ It has two-transistors to control bit-lines and two back to back coupled inverters.

Density

- In SRAM we use 6-transistors to save 1-bit of data.
- DRAMs are one-transistor & 1-capacitor.
- Flash is effectively less than one transistor per bit of data.
  - On one transistor you would be storing two bits or 3-bits of data. Therefore flash are much more dense.

Memory Type	SRAM	DRAM	Flash
✓ Speed	Very fast	Fast	Very slow
✓ Density	low	High	Very high
✓ Endurance	Better	Better	Poor
Power *	low	High	Very low
✓ Refresh	No	Yes	No
Retention	Volatile	Volatile	Non-volatile
Scaleable	Good	Bad	Good
✓ Mechanism *	Bi-stable Latch	Capacitor	FN Tunneling ± HCI ⓐ Hot carrier injection

Endurance

- Endurance → No. of times you can read & write on a particular memory

(i) In SRAM, it is 6-transistors fully static CMOS logic & regular voltage operation & Endurance is maximum.

(ii) In DRAM, it has capacitor which can leakage the data but even though endurance is very High

(iii) In Flash we use very high voltages & therefore endurance goes bad. (No. of read & write are very less)

Refresh

- Refresh → Restoring the data that was stored on the cell.
  - ** If you don't restore the data on regular basis then you end up loosing data. Hence you have to refresh.
- DRAMs store data on capacitor (floating cap.) hence data can leak over the time. Therefore we need to refresh the content of DRAM.

Q. How we can store two - or - three bits on a 1-transistor in a Flash?

→ We have floating gate in flash.

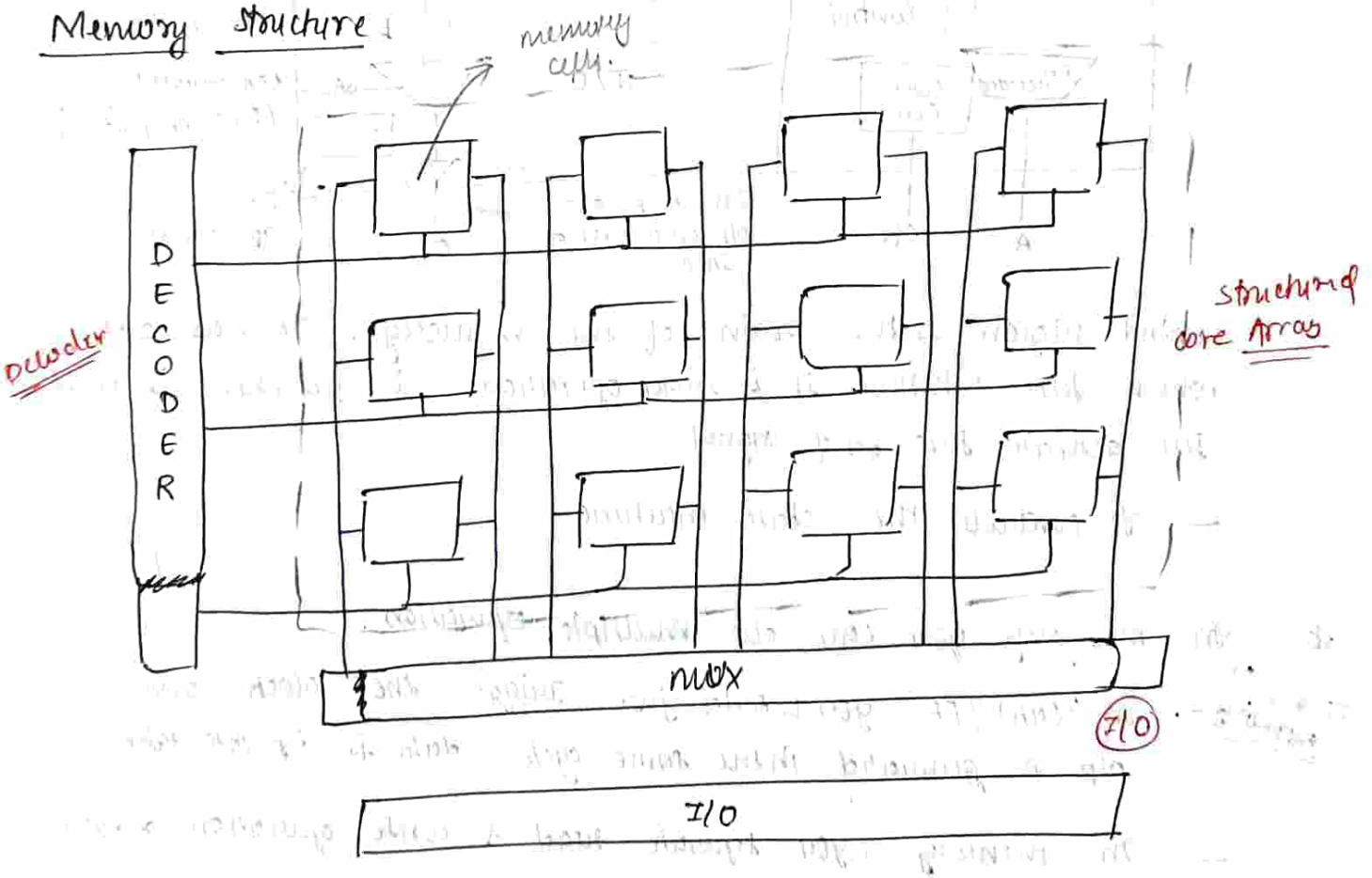
In Flash, we use special kind of device, where you have [gate-1 ↔ silicon-dioxide ↔ gate-2 ↔ SiO_2 ↔ substrate]

↳ Just like FGMOS → floating gate mos:
↳ in which you can trap charges & hence V_t would change and then you would be able to change say what is the kind of data stored here.

Q. Why we need to refresh in DRAM?

→ If we don't refresh then we have kept 2 transistor & 1 cap (1 kept) bus will become zero.

Memory Structure :

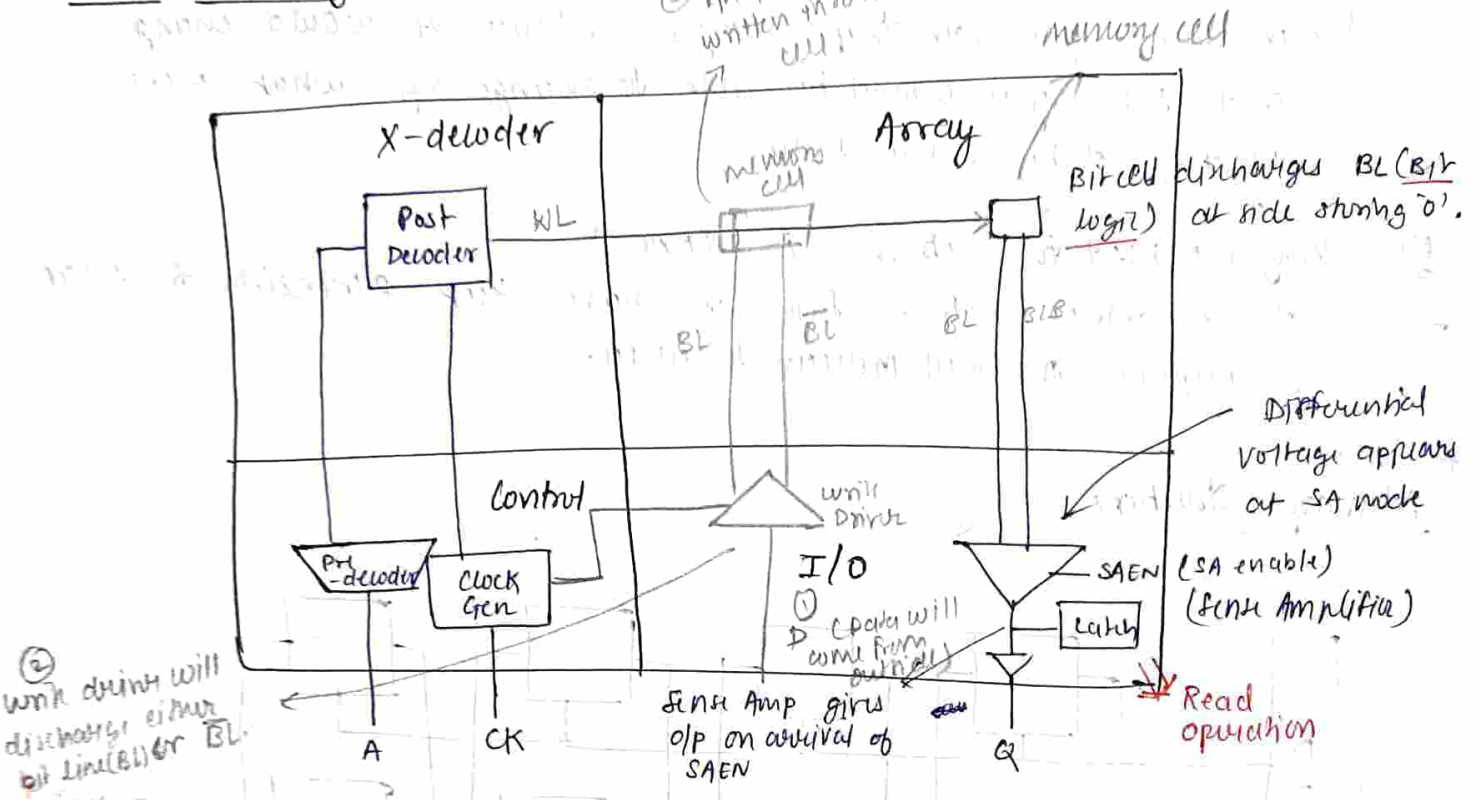


* These r cluster of storage elements which are closely coupled not only through word line but also through bit line, most time remains constant

+ Principle

- There is some decoding architecture that remains constant
- There is I/O which have ckt to buffer the o/p ~~and~~ remains same
- Some control region in memory remains constant.

Basic Memory Architecture :



- Control region is the brain of the memory. It has ckt which tells whether it is read operation, if yes then it will tell actual ~~the~~ set of signal.
  - It controls the state machine.

* In memory you can do multiple operation.

FF VLS MEMORY

- In Latch/FF, you would just trigger the clock and o/p is generated in the same cycle data is i/p into latch.
- In memory, you separate read & write operation occurs cycle.
  - read operation
    - * for ~~write~~ we use diff set of i/ps
      For write operation we use diff. set of i/ps

* There is many states in which the memory ckt can be and that is controlled by control unit.

Control Region : It has clock generator and address pre-decoding.

- When the address comes they are pre-decoded and when the clock comes internal clock is generated.
- They together go to the post decoder inside the X-decoder.
- The post decoder will generate word line.
  - It will select a particular row & selected the memory cells stored in that particular row will discharge bit lines.
- In SRAM, we are showing differential pair of bit lines (BL).
  - bit line (BL) & $\overline{BL}$
- In SRAM also we can have single ended i.e 1-bit line per memory cell
- In PROM, DRAM, flash memories, you would typically one-bit line per memory cell.

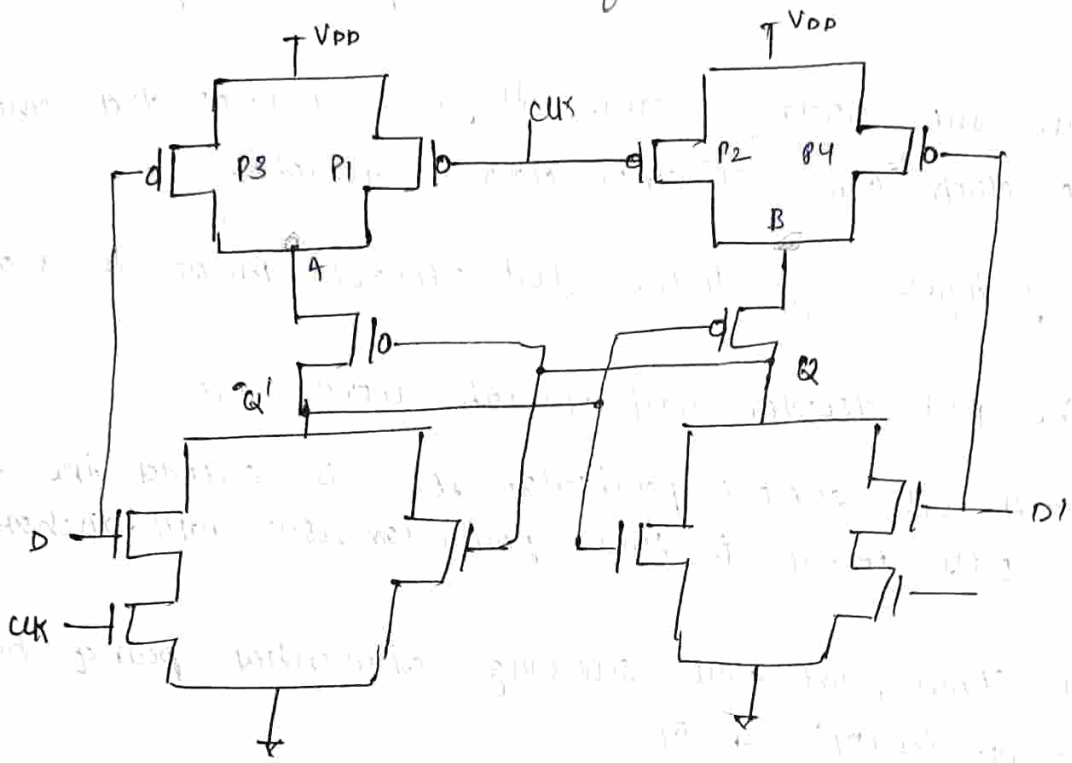
Q. Why we don't have 2-Bit lines in DRAM?

→ If we have to route 1 metal v/s 2-metal which is denser? using 1-metal will be denser. As we want to keep density very high hence we don't use 2-Bit lines in DRAM.

* We you select a memory cell some bit line would discharge and that discharge you would measure using the sense Amplifier

D-FF Implementation

This D-ff will be used to select only one word in memory block.



- Total = 14 transistors = 12 + 2 (for generating D')

Q. Out of this 14 Transistors, there are transistors which are not really needed or there are few transistors which could actually be shared across memory cells

→ For example, consider the case where word line is not selected, where the clock is not coming (i.e. clk=0) means P_1 & P_2 are ON. Therefore V_{DD} comes directly over node A & B.

— means pmos P1, P2, P3, P4 are not required 99% of the time, because for memory clock would come rarely.

- Suppose you have 1024 words in memory, even if you access memory in every cycle, then also every word is accessed in every 1 out of 1024 times.

— So more than 99% of times P1, P2, P3, P4 are not getting selected (then pmos are simply waste)

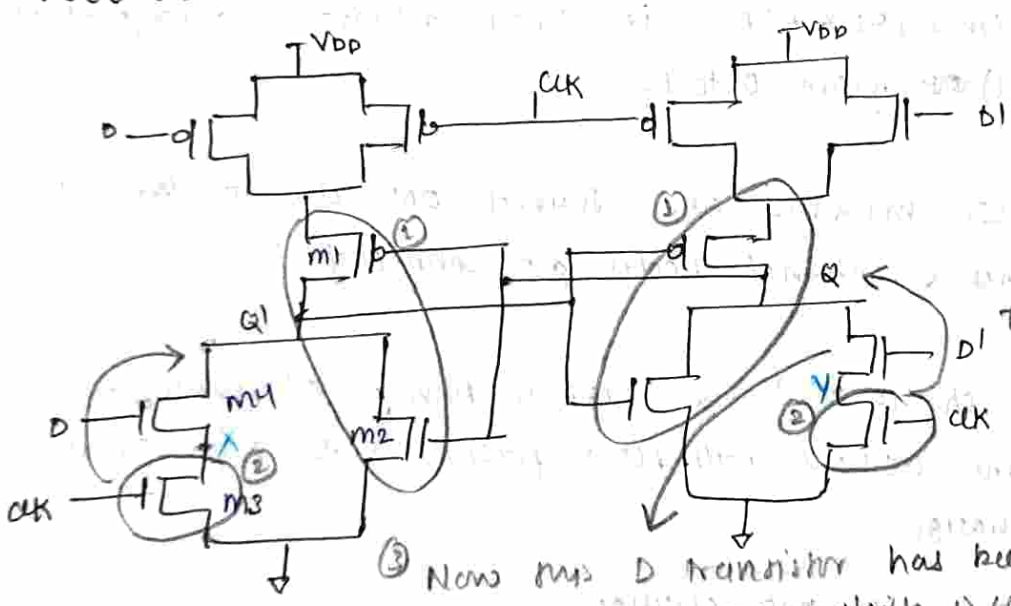
Q. What will be the alternative way to implement these 1's cell where P_1, P_2, P_3, P_4 are needed?

→ Instead of using full CMOS static ckt we can use pseudo NMOS.

* But the challange in designing pseudo NMOS is "sizing".
∴ we have size our MOSFET very carefully.

* When data is read from one memory cell then cell is ON for that cell & off for all other memory cell, therefore PMOSs of other memory cell are not used for that time.

Cell operation : (Memory cell operation)

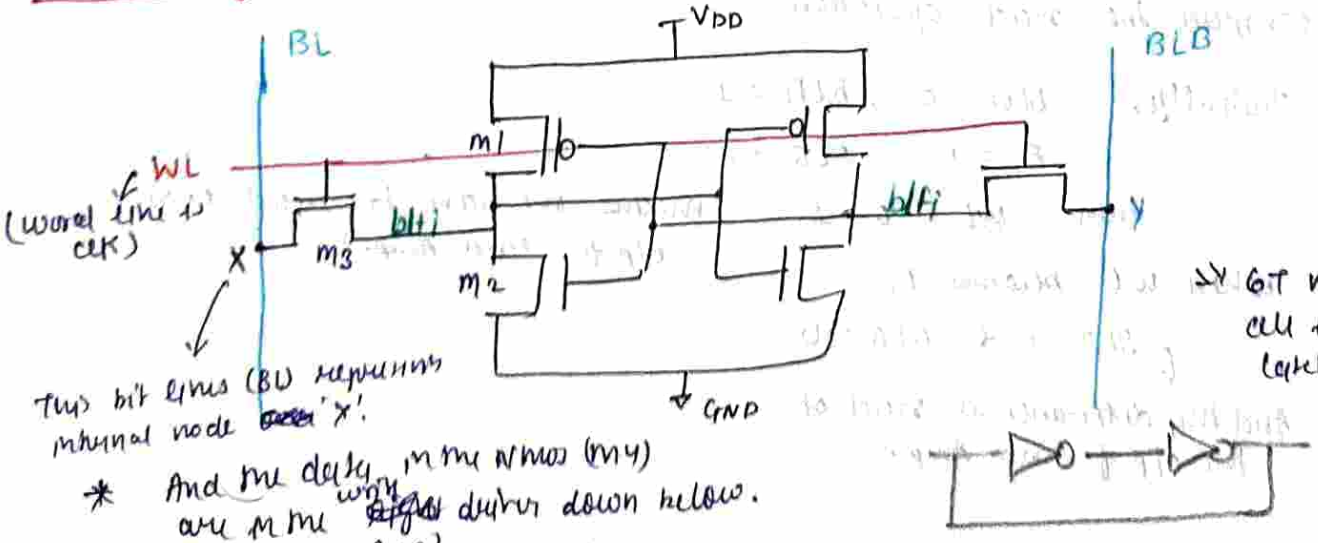


- Using latch we are replacing 14 transistor millions of times.
- Using memory cell we required only 6 transistor millions of times.

This CLK transistor is taken up, now CLK is called word line (WL).

③ Now this D transistor has been removed & put in a common area which is called the I/O region.
— This D & D' transistor is put in the write/Read driver now it is in the (Sense Amplifier)

6T Memory cell

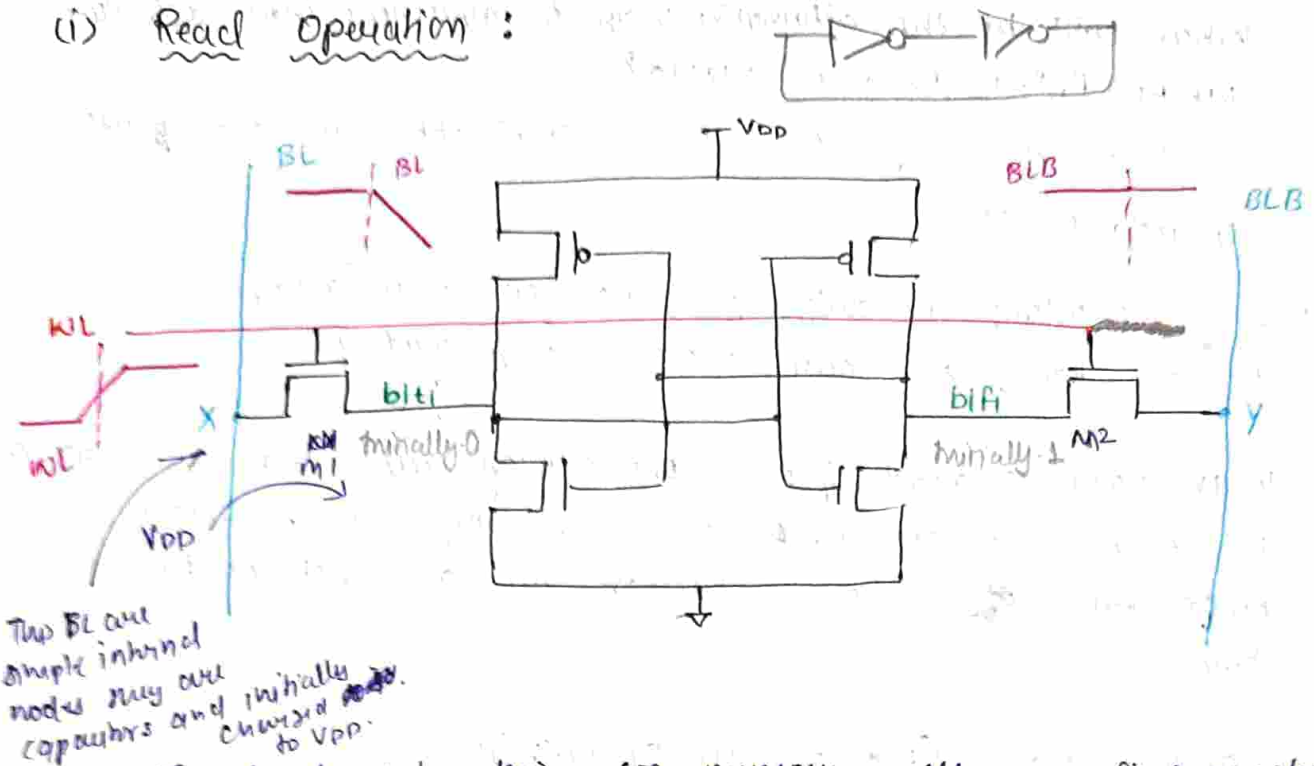


This bit lines (BL) represents internal node 'X'.

* And the data in the NMOS (M4) are in the write driver down below. (In Sense Amp)

→ 6T memory cell is minimized latch

(i) Read Operation :



- To read into 6T memory cell, we first precharge both the bit lines (BL & BLB) ie. BL = 1 & BLB = 1 and then take the word line (WL) from 0 to 1.

- When NMOS M1 & M2 are turned ON due to WL = 1. Then bit lines & internal nodes get connected.

- Node X is storing '1' & blti is having '0' initially. Therefore some current will flow from X to blti. Then BL will discharge

  – But BLB will not change

- This differential (b/w BL & BLB) will be sensed at the sense Amplifier to complete the read operation

* Initially, blti = 0 , blfi = 1

BL = 1 , BLB = 1

Then WL = 0 → 1 (means we have to read 1 at o/p of sense Amp.)

When WL becomes 1,

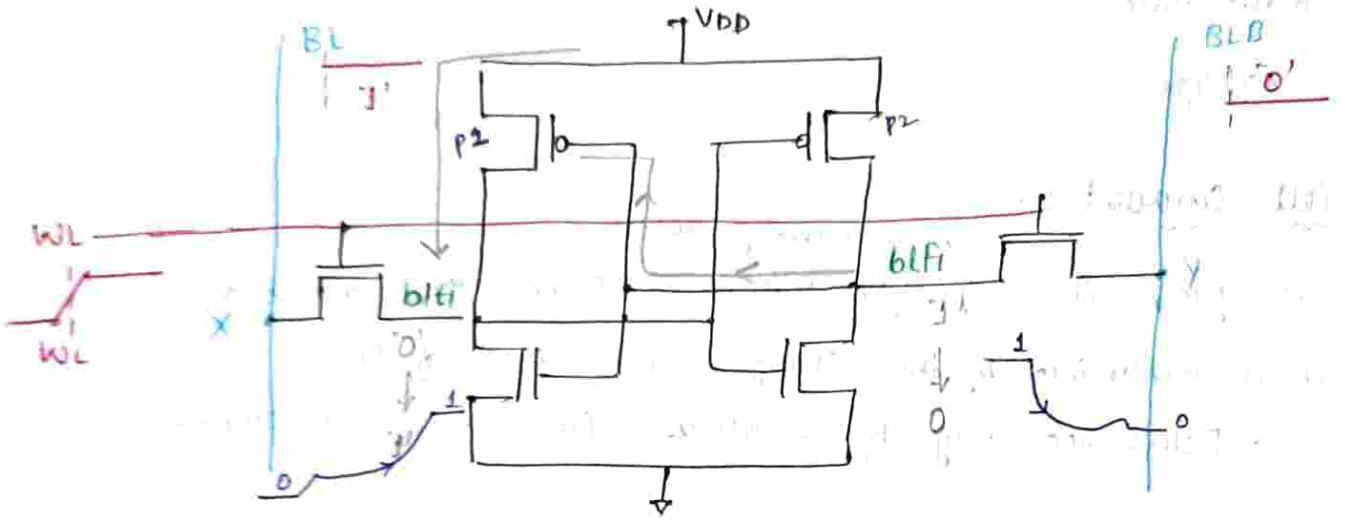
BLB = 1 & BLB = 0

And this difference is read at the o/p of sense Amp.

Note: The side on which (blti = 0) '0' was stored. That particular side, the bit line would discharge.

* If BL discharges, '0' is read at the o/p.] or vice versa
If BLB discharges, '1' is read at the o/p.

(ii) Write Operation:



- To write the data into the memory cell, we have to change
⇒ blti from '0' → '1' & blfi from '1' → '0'

— For that what would you write on the BL & BLB.

Ans - We put BL = 1 & BLB = 0

- * The data that we want at blti - should be put in BL.
The data that we want at blfi - should be put in BLB.

- The node ~~of~~ blfi discharges fast and reaches '0' and on other side node ~~of~~ blti will only charge ~~a little~~ little and it will completely ~~then~~ gets charged when blfi becomes '0' & P1 ~~near~~ turns ON then only blti becomes completely 1.

* Writing '0' ~~on~~ '1' would take more time than writing '0'. (on blfi)
(on blti)

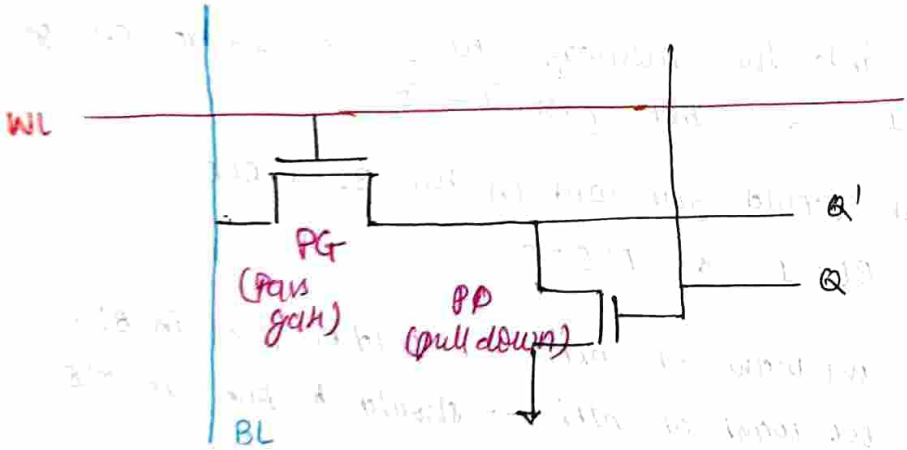
— Because to write a '1' you have to write a ~~zero~~ '0' (on blfi) & then only '1' will be written (on blti)

Bitcell - Figure of Merits

1. Cell Current → It tells how quickly my bit line will discharge. (It controls Read speed)
2. Bit line leakage → Power & Read speed.
3. Cell stability
4. Retention voltage
5. Write margin
6. Write time
7. Leakage

1. Cell Current :-

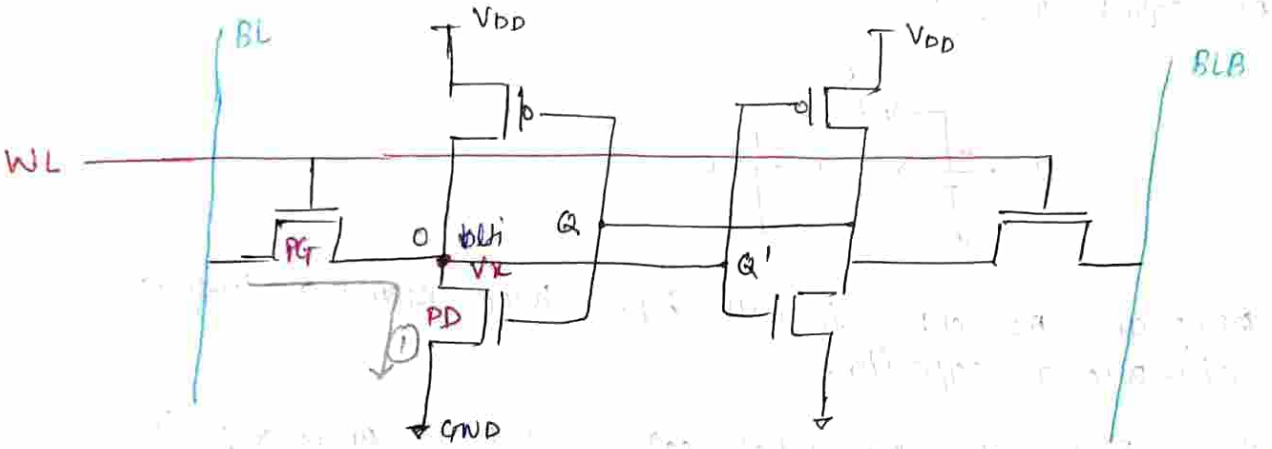
- It defines the speed of memory access at which Bit line can discharge
- It is determined by the sizing of the stack of PMOS.
  - Determined only by 2 devices PG & PD of side storing '0'



- Bigger the PG & PD, the Bigger the current would be.
  - Hence leakage also increases, more the capacitance of BL also.
  - So we can't keep on increasing the size.

2. Cell Stability

- It is the most important specification of a storage element.
- Challange is highest during read operation. because during read operation you expose your latch (your storage element) to external noise.



- When we start the read operation, current starts to flow like ①.
- Initially blti was storing '0', but to ensure the current flows from 'blti' to 'GND', the value of "blti" goes higher than '0' (let Vx).
- This Vx, turn ON PMOS OFF + NMOS ON on RHS and which makes the system unstable and risk starts to appear.
  - hence cell stability starts to degrade
- Therefore, we try to ensure that the Vx is as low as possible. For that we get another sizing constraint:
  - PG must be much more resistive than the PD (pull down) (pass gate)

$$PG\ (size) < PD\ (size)$$

Q. What will happen if PD is more resistive than PG?

→ Then node 'blti' @ Q' will rise above $\frac{V_{DD}}{2}$ and it will kind of flip the content of memory cell. And we don't want that.

* When we remove the PMOSs then to do the memory cell operation we change the size of other transistors.

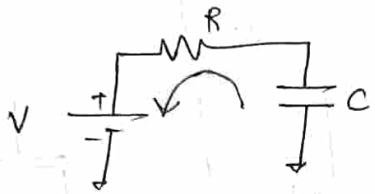
Positive F/B → Data flip → ERROR

3. Write Margin → metric of speed, writability.

- * When BL goes to '0', then node blt will discharge.
- Bit line is a RC ckt and you can't discharge a capacitor to '0' in finite time.

Q: How much time is required to discharge a capacitor (through RC path) to '0'?

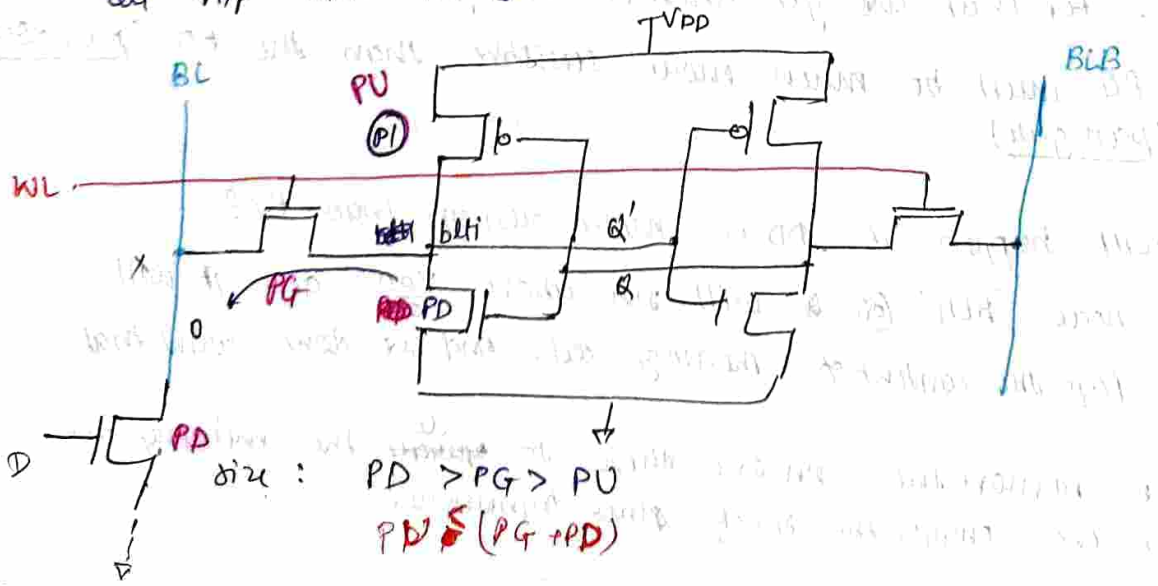
→



- Being an RC ckt it will take huge amount of time to discharge a capacitor.
- It is first order Differential eq^n, therefore practically it takes infinite amount of time to discharge. But we don't have that amount of time to read & write a memory.

(*) Since we can't discharge the capacitor to '0' then what is the higher level of BL at which I will still be able to write into the memory cell? → called write margin

- Now with this write margin we don't have to discharge the cap. to '0'. We have to discharge it to the point where it ensures memory cell flip with WL @ V_{DD}.



size: PD > PG > PU

PD ≤ (PG + PD)

* Highest level of BL to ensure cell flip with WL at V_{DD}.
* OR Lowest level of $\overline{WL}$ to ensure cell flip with BL at gnd.

Q: What is meant by -ve write margin?

→ We take value of Bit line voltage lower than 0V. so that it can write into the memory cell.

* Negative Bit line write assist scheme

* When you want to write into the memory, that the pull up PMOS (PU) should not be able to drive & should drain lesser current than what this stack of PG & PD can sink.

— series of PG & PD should be stronger than the pull up. (PU)

$(PG + PD) > PU$

. series of PG (pass gate) & write driver (PD) should be stronger than PU.

. PD should be resistive to give higher bump on other side

. * So, write ability of the cell is inverse to its stability.

4. Leakage

. All off cell in bitcell leak.

. Even ON devices have junction leakage.

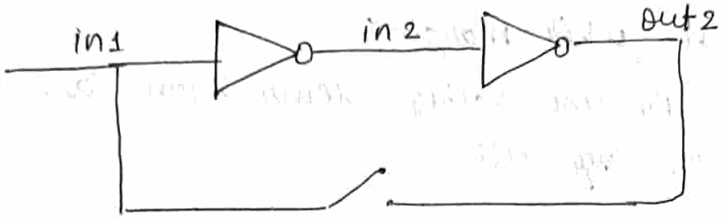
. we should have small PG, PD & PU for low leakage

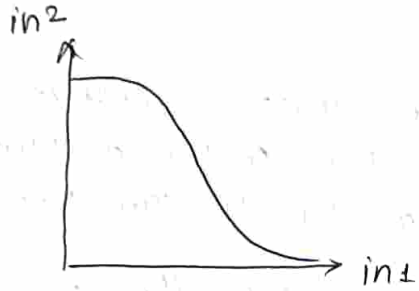
. Long channel.

— If min. length of Transistor is x then in memory cell length used will be 10x or 20x. or 1.2x or 1.5x.

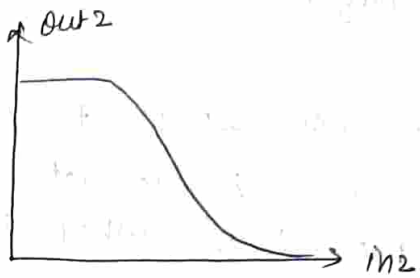
. Thick gate oxide, less doping on source/drain.

Butterfly curve with cell stability





As in1 ↑ in2 ↓



As in2 ↑ out2 ↓

- In memory cell there two inverters are connected back to back.
- Therefore we close the switch & don't give any i/p in1 it replaced by out2 in a F/B.

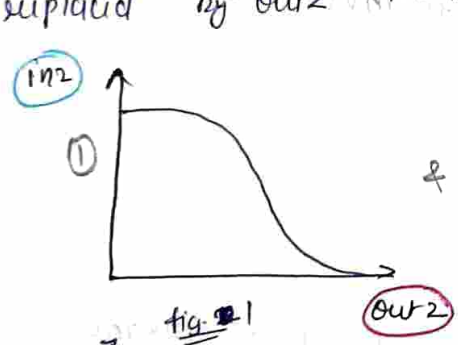


fig. 1

&

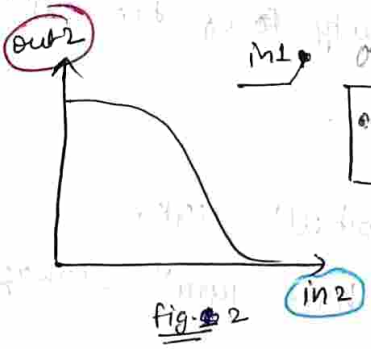
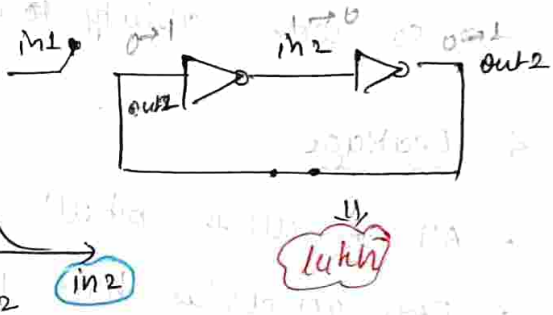


fig. 2



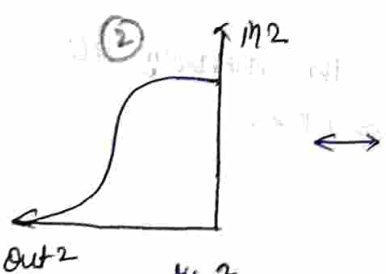


fig. 3

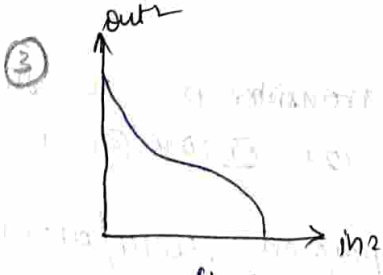
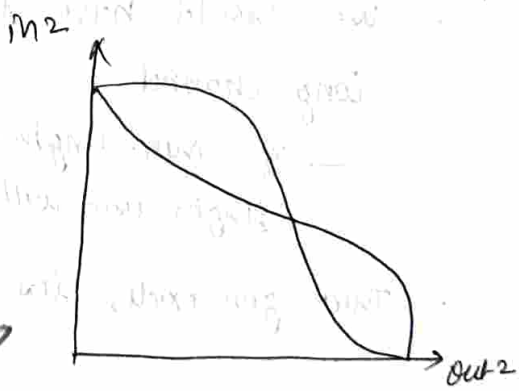


fig. 4

fig. 3 ⊕ fig. 4



Butterfly curve

* Transfer func^n of the latch is Butterfly curve.

"Transfer func^n of the latch"

- Now the important thing is, how much noise we can put in the latch ie. the cell stability.
  — The noise for which latch is immune to.

Noise injection

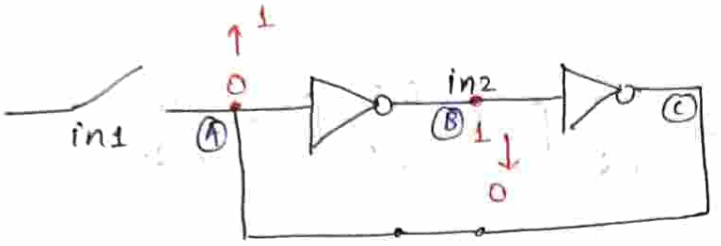


fig (a)

- If I am storing '0' at node (A) & '1' at node (B) then worst case noise will be injected when '0' goes to '1' on node (A) & '1' goes to '0' on node (B). (Worst case noise pair).
- This latch can be considered as the long chain of inverters storing 010101. (Just for sake of understanding)

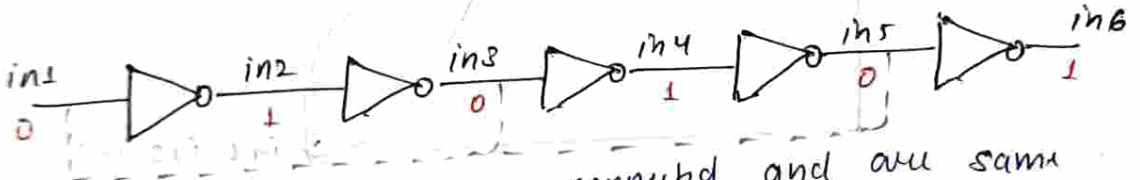


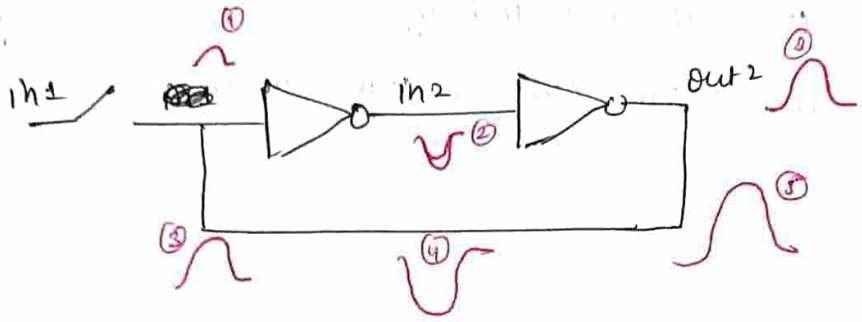
fig.(a)

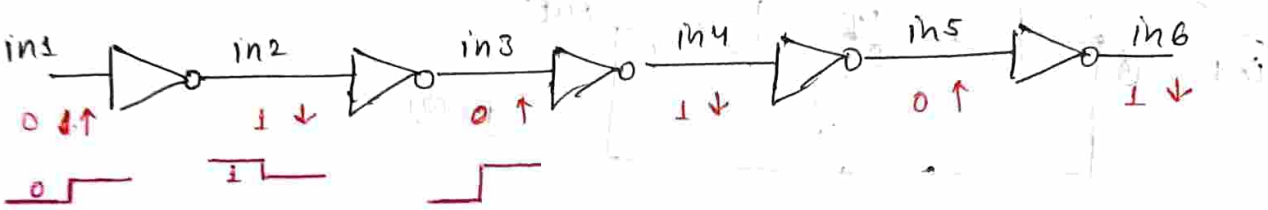
where in1, in3, in5 are connected and are same

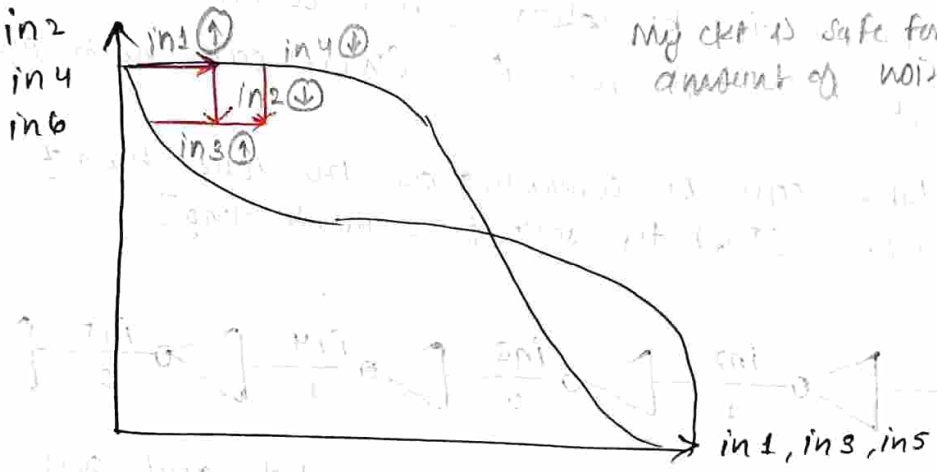
- If small noise is given at node (A) then this noise gets increased when it goes to node (B) then node (C) will have higher noise and again this noise is fed to node (A).
  — Because latch amplifying action.
- So to overlap the Big noise at node (C) & small noise at node (A) we use the fig.(b). Just for sake of understanding.

Butterfly curve with noise injection [& Noise Margin]

For high noise margin



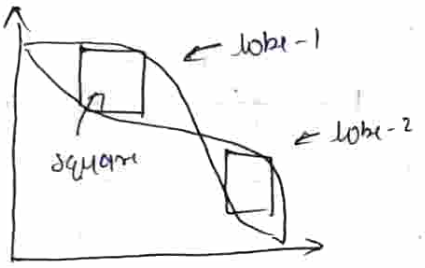


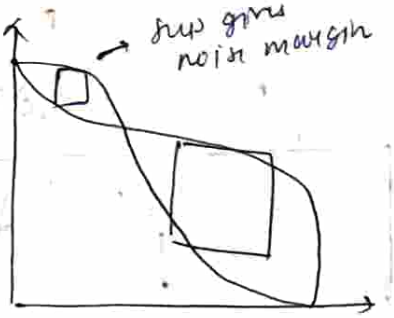


My ckt is safe for max amount of noise

For max noise margin

* The largest square that we can fit into both lobes gives us the noise margin.





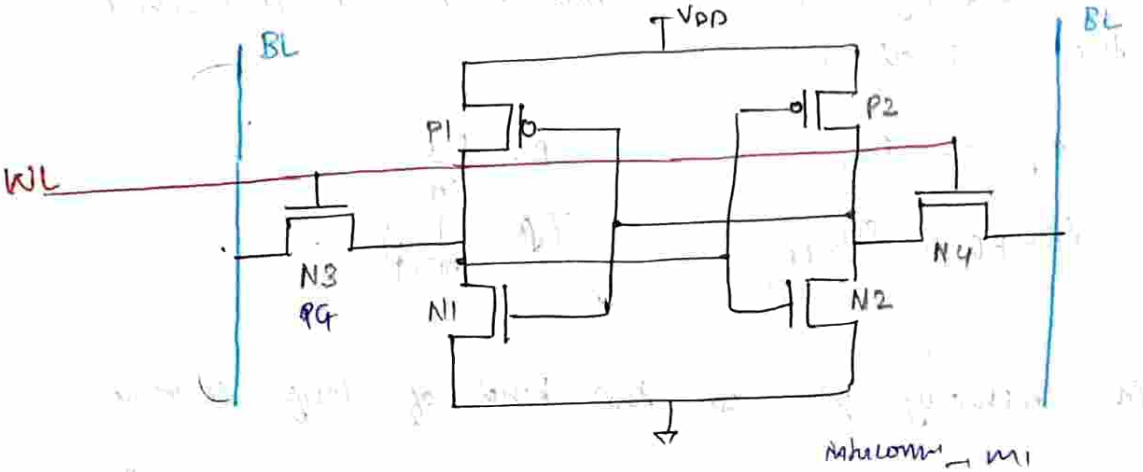
* Stability of memory cell can be measured by ~~the~~ fitting the largest square in the smaller lobe of the butterfly curve.

Bit-cell Layout

- In standard cell layout, we have to keep a distance of $\frac{1}{2}$ DRC from the PR boundary.

Q. Can we remove $\frac{1}{2}$ DRC in contacts?

* In memories you want to share as much around the cell boundary as possible, because you have a structured array. You know that your memory cell is abutted with another memory cell only.

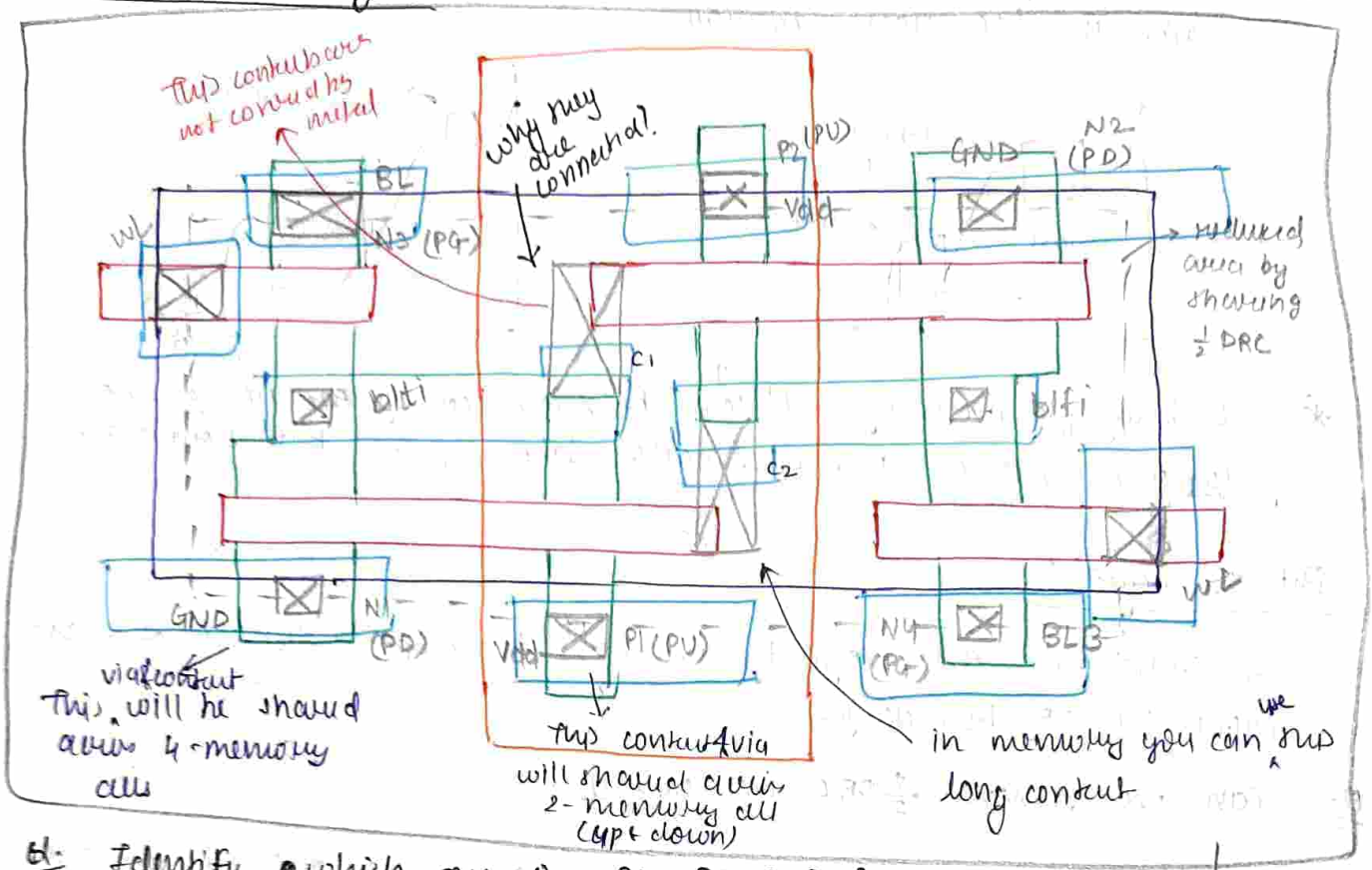


0.15×2
$0.3\ \mu m^2$

$R = 5\ \Omega/(\mu m)^2$
$= 1.5\ \Omega$

Metal1 → M1
BL → M2
WL, VDD, GND → M3

6T- SRAM Layout



Q. Identify which one is PU, PD & PG?

→ · The one in the NWell is PU (P1 & P2)

· ~~Inverter~~ ~~pdg~~ is

· For cell stability pass gate has to be more resistive than the pull down

$$R_{PG} > R_{PD} \qquad R = \rho \frac{l}{A}$$

$$\therefore \text{size}_{PG} < \text{size}_{PD} \qquad \left(R \propto \frac{1}{\text{size}}\right)$$

* Only in memory you use this kind of long contact.

Q. In layout, if you make long contact then what do you see?

→ DRC error comes.

→ Because there is minimum contact to contact spacing ~~for~~ for DRC. Hence it is increasing the area of layout.

* In memory, instead of putting two contacts (to decrease the area of layout) [ie, putting one contact on poly & one contact on active and maintaining the DRC b/w them].

- We Actually kind of merge two contacts and form a long contact. And this long contact is allowed to be used inside the memory.

Q. Why this rectangular contact only allowed in memory?

→ In Memory this is the only configuration where you allow this long contacts.

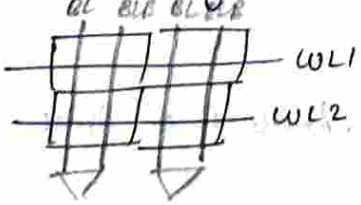
* The DRC is because optics don't allow to fabricate two things.
  - In optics you can predefine patterns then you can make the diffraction plate in such a way that you make the structure well.
  - So by using RET (Resolution enhancement technique), we allow long contacts only inside the SRAM cells.

* This rectangular contact will be used in the fixed array only and also in the fixed pattern only. So we will have a diffraction pattern for which I have to optimize my mask making process.

- Then you put some metal layers above the contact to connect it to BL, BLB, VDD, GND, WL

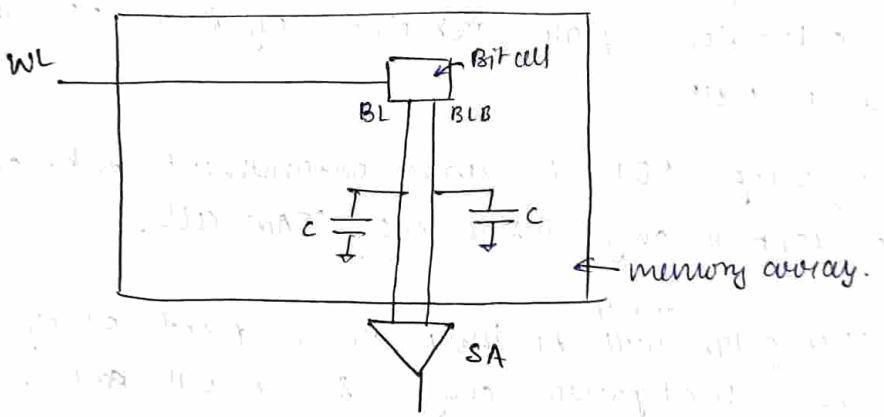
- (Vias & contacts) coming in the corner are shared across 4-memory cells & (vias & contacts) coming in the upside & downside are shared across 2-memory cell.

* Eg. of 2X2 array (ie 4 memory cells shared across PR boundary)



Sense Amplifier (Why?)

- Bit line capacitance becomes significant for large array.
  - If each cell contributes 2fF.
    ∴ for 256 cells
    ⇒ 512fF + wire capacitance
  - pull-down resistance is about 15K
    ∴ $\tau = \frac{RC\Delta V}{V_{dd}}$
  - Bit line delay = 7.5nsec [assuming $\Delta V = V_{dd}$]
    (ideally $\tau < 1$nsec)



- WL will come & select a Bit cell and that Bit cell will discharge the BL then we get the o/p.
- But this BL has very huge capacitance.
- If we want to discharge this BL which is already charged to V_{dd}. Therefore to read the data we have to discharge the BL upto $\frac{V_{DD}}{2}$ so that it can read the 0).
  - means we have to discharge atleast $\frac{CV_{DD}}{2}$ charge from the BL to read '0' at the o/p.
- And depending on the cell current, there is a time it will take to discharge particular bit line.

* Solution: We can't easily change R, C or V_{DD}, but we can change ΔV.

$$\tau = \frac{RC\Delta V}{V_{DD}}$$

$\Delta V \rightarrow$ smaller sensed voltage

- It can reliably sense ΔV as small as $< 50mV$

- If we don't discharge it to $\frac{V_{DD}}{2}$ saves time by only discharging the BL upto $\frac{V_{DD}}{8}$ @ $\frac{V_{DD}}{10}$ (to get $\tau \lesssim 1nsec$).

The lower I discharge, the faster I can be.

- So we put an amplifier down in the array, which would amplify $\frac{V_{DD}}{12}$ discharge into ~~$\frac{V_{DD}}{2}$~~ full V_{DD} discharge at the o/p.

  * Therefore we use Sense Amplifier.