Introduction to Ic Technology:

* We know that to day Electronics is characterized by reliability, low power dissipation, extremely less size & low cost, coupled with an ability to cope easily with a high degree of sophistication and complexity.

* Electronics & in a particular, the Integrated circuit, has made possible the design of powerful and flexible processors which provide highly intelligent and adaptable devices for the user.

* IC memories have provided the Essential Elements to complement these processors & together with a wide range of logic & analog IC's, they have provided, the s/m designer with components of considerable capability & Extensive application.

* B. Shockley, walter H. Brattain & John Bardeen of Bell Telephone labratory was followed by the development of Ic.

* The generations of Ic's are:

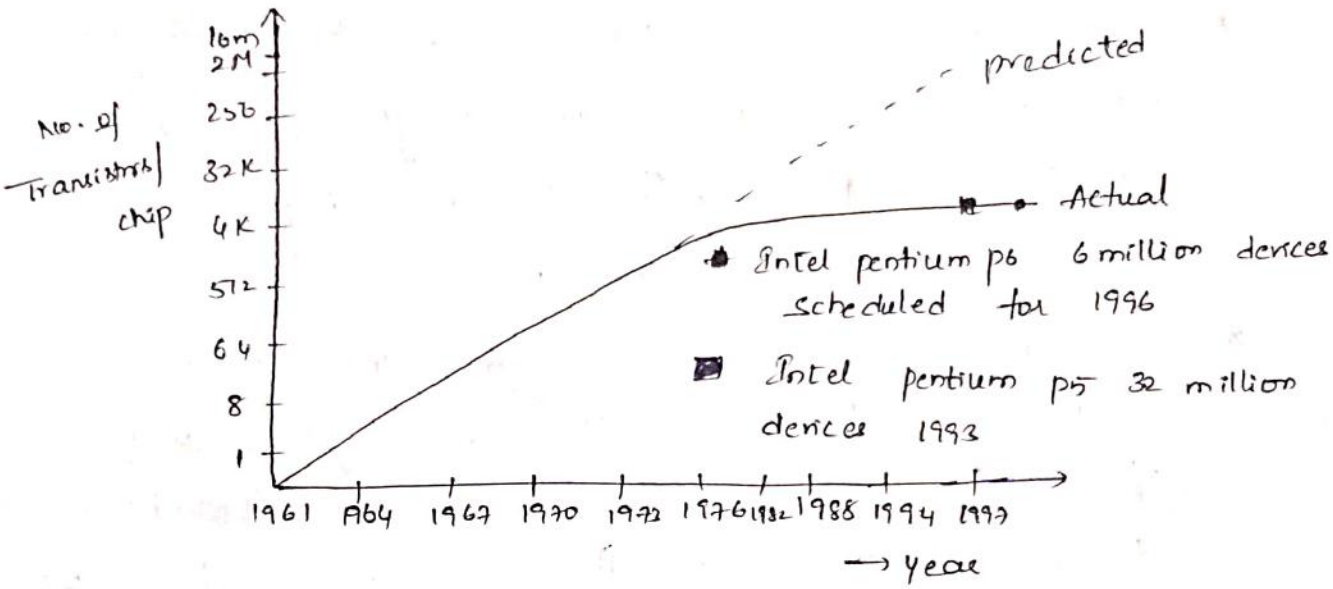
1) Small scale Integration (SSI) :- (10-100 Transistors)
2) Medium scale Integration (MSI) : (100-1000)
3) Large scale Integration (LSI) : (1k - 20k)
4) VLSI (Very large scale Integration : (>20k) = (20 to 10lakhs)

* The upcoming 5th generation ultra large scale Integration (ULSI) which is characterized by complexities in Excess of 3 million devices on a single Ic chip. (> 1crore)

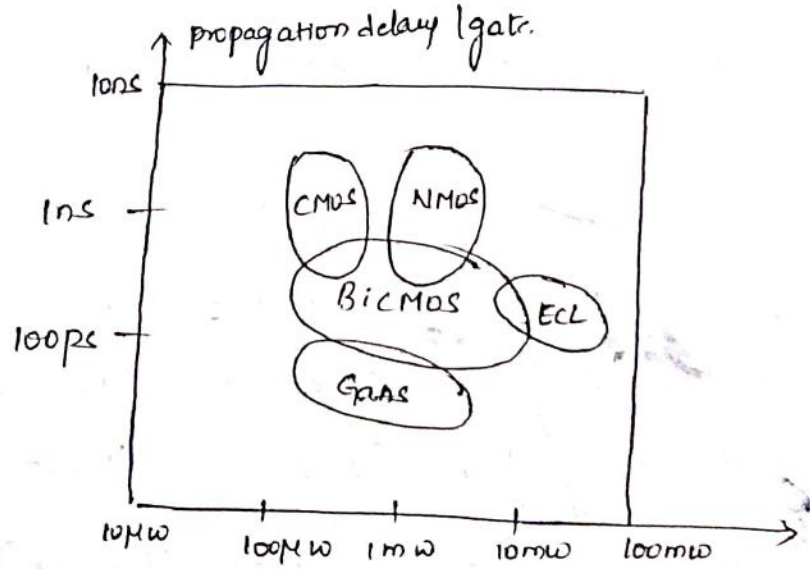
IC era:

* The potential of silicon IC's has been extremely rapid growth interms of no of Transistors being integrated in to ckts. on a single silicon chip

* In less than 3 decades, this no has risen from tens to millions. & it increases double for every 24 years. is known as moore's 1st law.



Speed / power performance of available technology:



* over the past several years, silicon cmos technology has become the dominant fabrication process for relatively high performance & cost effective VLSI circuits.

* The increase in no. of Transistors/chip is highlighted by recent products such as RISC chips in which it is possible to process some 35 million instructions per second.

* In particular the Emerging (GaAs) based technology will be most significant technology for ultra high speed logic/fast digital processors.

* VLSI systems are much smaller, and consume less power than the discrete components used to build electronic S/m's before 1960's. IC's are easier to design, manufacture and more reliable than discrete S/m's. that makes it possible to develop special purpose S/m's that are more efficient than General purpose computers for the task at hand

Advantages of VLSI:

* Size: the size of IC is small - both Transistors & wires are shrunk to micrometer size which leads to high speed & low power consumption.

* Speed: S/g's can be switched b/n logic '0' & logic '1' much quicker with in a chip. communication within a chip can occur hundreds of times faster than communication b/n chips on a printed ckt board.

* power consumption: Logic operations with in a chip also take much less power due to small size of ckts on chip.

applications of VLSI:

* used in personal Entertainment S/m's such as portable mp3 players & DVD players
* used in Digital electronics compress & decompress video

even at high definition data rates on the fly in consumer electronics

* low cost terminals for web browsing.
* used in medical electronic sim's measure bodily functions & perform complex processing algorithms. to warn about unusual conditions.

Basic steps of Ic fabrication :

The manufacturing of Integrated circuits (Ic) consists of following steps.

step 1): Wafer production:

* The wafer is round slice of semiconductor material such as silicon is preferred due to its characteristics. It is more suitable for manufacturing Ic.
* It is base or substrate to entire chip.
* 1st purified poly crystalline silicon is created from sand. then it is heated to produce molten liquid.
* A small piece of solid silicon is dipped on the molten liquid.
* Then the solid silicon is slowly pulled from melt. the liquid cools to form single crystal ingot.
* A thin round wafer of silicon is cut using wafer slicer. wafer slicer is a precise cutting machine & each slice having thickness about 0.01 to 0.025 μm.
* when wafer is sliced. the surface will be damaged. It can be smoothening by polishing. After polishing the wafer. it must throughly clean and dried.
* The wafers are cleaned using high purity low particlechemicals The silicon wafers are exposed to ultra pure oxygen.

Step2: Epitaxial growth:

* It means the growing of single silicon crystal up on original silicon substrate.
* A uniform layer of silicon dioxide is formed on the surface of water.

Step 3: Etching:

* Selectively removing unwanted material from the surface of the wafer.
* The pattern of the photo resist is transferd to the wafer by means of etching agents.
* The parts of material are protected by this etching mask

Step 4: Masking:

* To protect some area of wafer when working on another area, a process called photolithography is used.
* The process of photolithography includes masking with a photo graphic mask. and photo etching.
* A photo resist film is applied on the wafer
* The wafer is aligned to mask using photo aligner
* then it is exposed to uv light through mask
* Before that the wafer must be aligned with the mask.

Step 5: Photolithography:

Photolithography is a technique that is used to define the shape of micro machined structures on a wafer.

Step6: Doping:

* To alter the electrical character of silicon, atom with one less electron than silicon such as boron atoms with one electron

greater than silicon such as phosphorous are introduced in to area.

* The p type (boron) and N-type (phosphorous) are created to reflect their conducting characteristics.

Step 7: -Atomic diffusion:

* Diffusion is defined as the movement of Impurity atoms in semiconductor material at high temperatures
* Then the wafers are heated at temperature of about 1500-2200°F. the inert gas carries the dopant chemical. The dopant & gas is passed through the wafers and finally the dopant will get deposited on the wafer.
* This method can only be used for large areas. For small areas it will be difficult and it may not be accurate.

Step 8: Ion Implantation:

* Ion Implantation is a low temperature process by which ions of one element are accelerated in to a solid target. there by changing the physical, chemical, or electrical properties of target.
* Ion Implantation is used in semiconductor device fabrication and in metal finishing, as well as in material science research
* The Ions can alter the elemental composition of target. If they stop & remain in target.
* Ion Implantation also causes, chemical and physical changes when the ions impinge on the target at high energy.

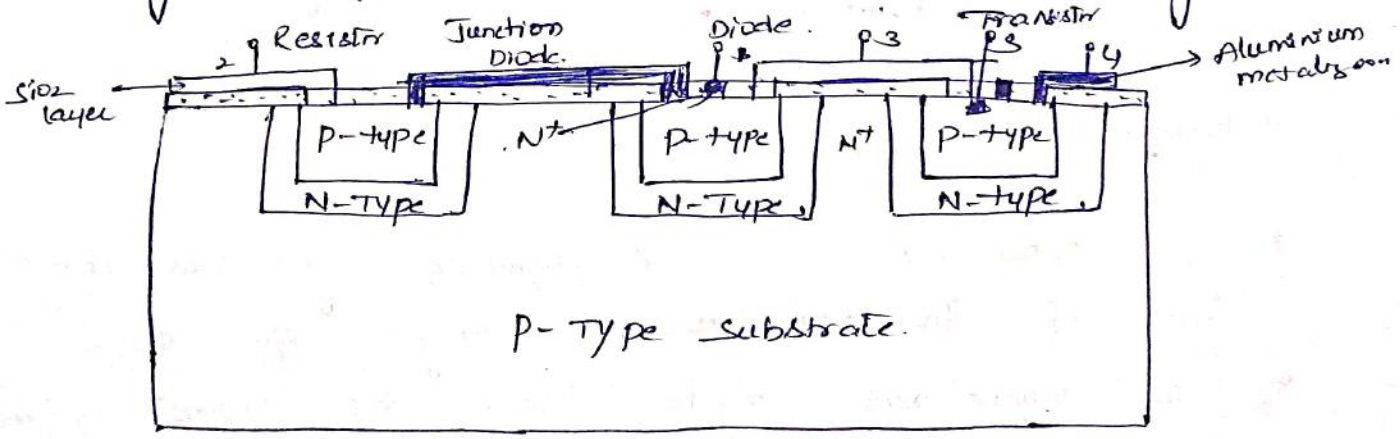
Step 9: Metallization:

* It is used to create contact with silicon and to make interconnections on chip.
* A thin layer of aluminium is deposited over the whole wafer
* Aluminium is selected because it is a good conductor, has good mechanical bond with silicon, forms low resistance contact and it can be applied and patterned with single deposition and etching process

Step10:

Assembly & packaging:

* Each of the wafer containing hundres of chips. Those chips are seperated and packaged by a method called scribing & cleaving.
* The wafer is similar to a piece of glass. A diamond saw cut the wafer in to single chips.
* The diamond tipped tool is used to cut the lines through the rectangular grid which seperates the individual chips
* Before packing, remaining chips are observed under microscope. The good chip is then mounted in to a package.



Basic MOS Transistors:

nmos Enhancement and depletion mode Transistors

* N-mos devices are formed in a p-type substrate of moderate doping level.
* The source and drain are formed by diffusing n-type impurity through suitable masks in to these areas to give the desired n-impurity concentration and give rise to depletion regions which External mainly in the more lightly doped p-region.
* Thus source and drain are isolated from one another by 2 diodes.
* connections to the source and drain are made by a deposited metal layers.
* In order to make a useful device, there must be the capability of establishing and controlling of current b/n source & drain
* This can be done in 2 ways, giving rise to Enhancement & depletion mode of Transistors

Enhancement mode MOS Transistor:

* In enhancement mode, A polysilicon gate is deposited on a layer of Insulation over the p-region b/n source & drain
* In enhancement mode device the channel is not established & the device is in non-conducting condition

$V_D = V_S = V_{GS} = 0$

* If the gate is connected to a suitable +ve voltage w.r.t source, then the electric field is established by the gate and substrate gives rise to a charge inversion region to substrate under the gate insulation and a conducting path or channel is formed b/n source & drain

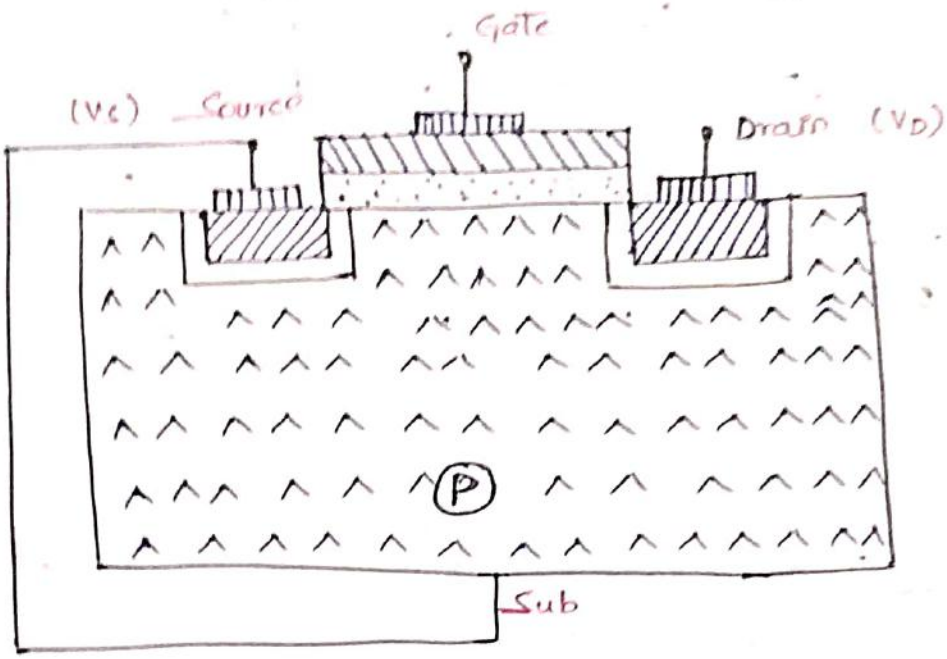
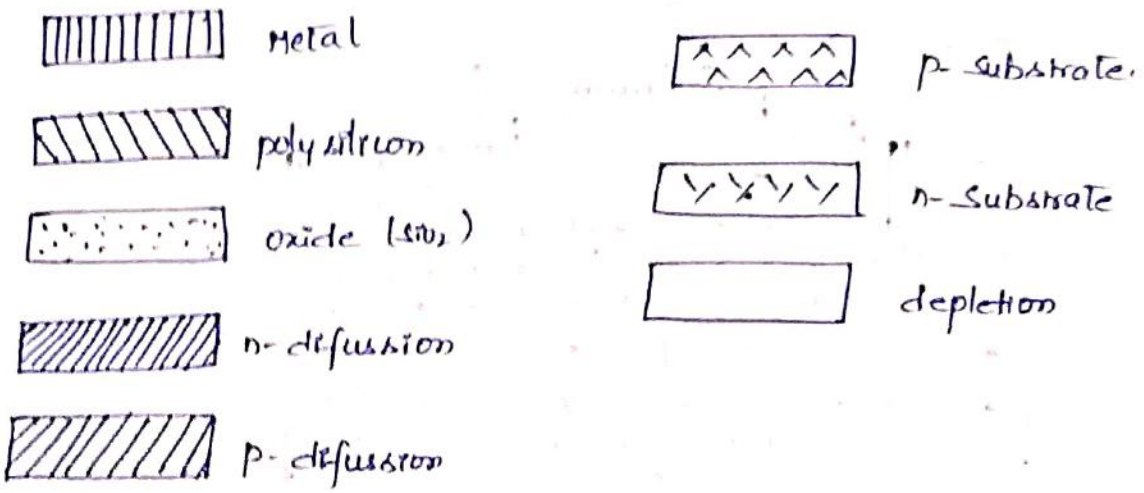


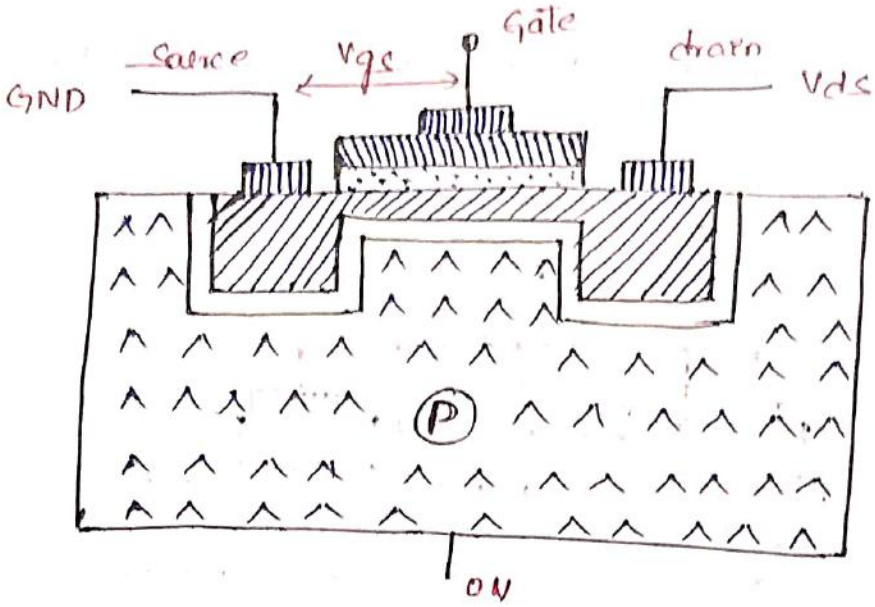
fig: NMOS Enhancement mode Transistor



To understand the mechanism of enhancement mode. we have to consider 3 conditions.

Threshold voltage (V_t): The minimum voltage applied b/n gate & source to establish channel.

Case i:

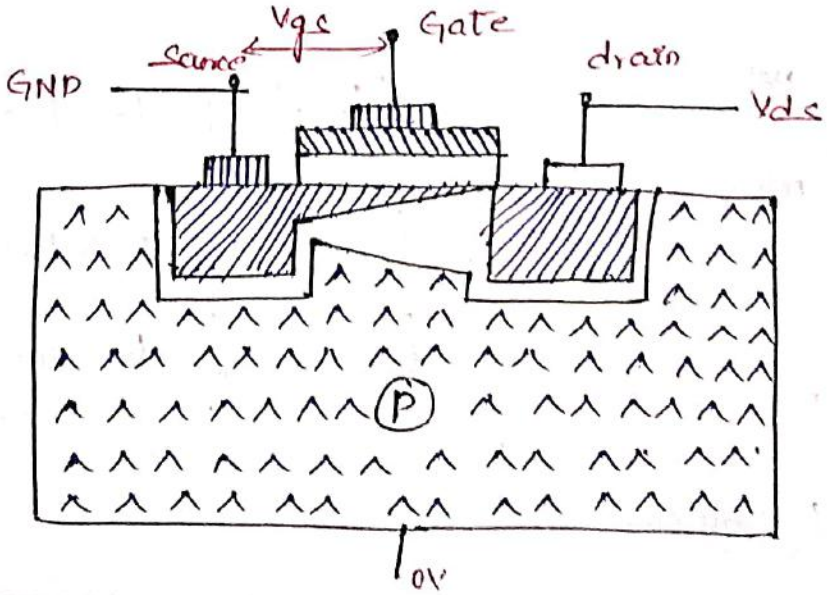


$V_{gs} > V_t$

$V_{ds} = 0V$

* by appling +ve voltage b/n gate and source. the channel is established. but no current flowing b/n source & drain ($V_{ds}=0V$)

Case (ii):



$V_{gs} > V_t$

$V_{ds} < V_{gs}$

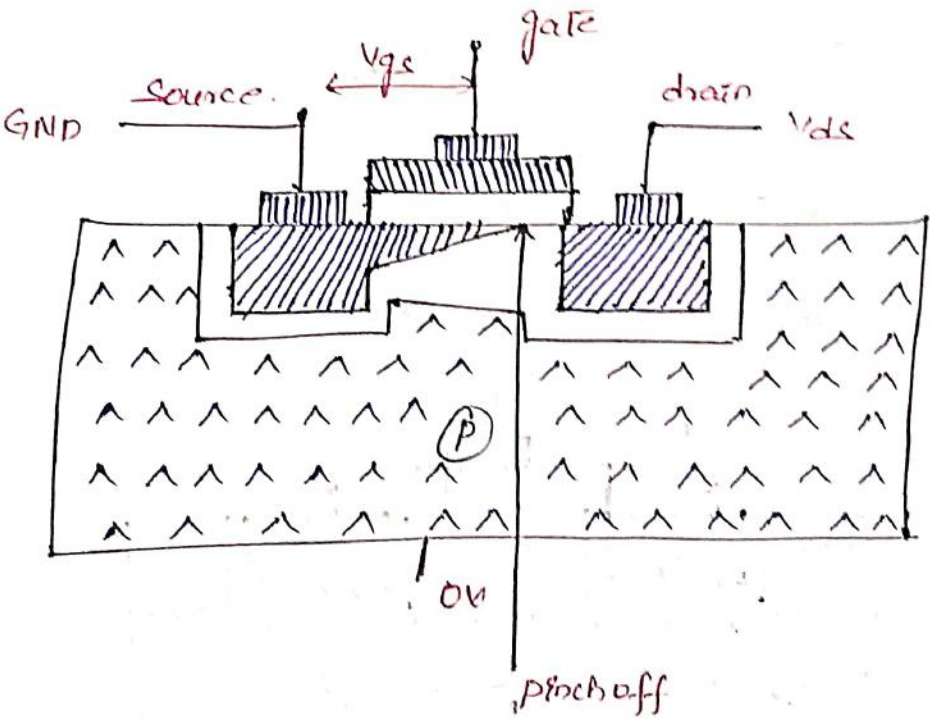
* Now consider the prevailing when current flows in the channel by applying a voltage V_{DS} b/n drain & source.

* Along the channel, a corresponding IR drop V_{DS} will be there. This results in voltage b/n gate & channel varying with distance along the channel with the voltage being a maximum V_{GS} at the source end.

* effective gate voltage $V_g = V_{GS} - V_t$ (no current flows when $V_{GS} < V_t$). there will be voltage available to invert the channel at the drain end so long as $V_{GS} - V_t \geq V_{DS}$.

* the limiting condition comes when $V_{DS} = V_{GS} - V_t$. for all voltages $V_{DS} < V_{GS} - V_t$, the device is in non-saturation region of operation which is $\boxed{V_{DS} < V_{GS} - V_t}$

Case(iii):



$V_{GS} > V_t$

$V_{DS} > V_{GS} - V_t$

* If V_{DS} is increased to a level greater than $V_{GS} - V_t$, or IR drop = $V_{GS} - V_t$ takes place over less than the whole length

of the channel so that over the part of channel near the drain, there is insufficient electric field available to give rise to an inversion layer to create the channel is called "pinchoff"

* In this case, diffusion current completes the path from source to drain. the channel to exhibit a high resistance & behaves as constant current source.

* This region is called as saturation, is characterised by almost constant current for increase in V_{ds} above $V_{dc} = V_{gs} - V_t$.

* In all the cases the channel will cease to exist & no current will flow when $V_{gs} < V_t$.

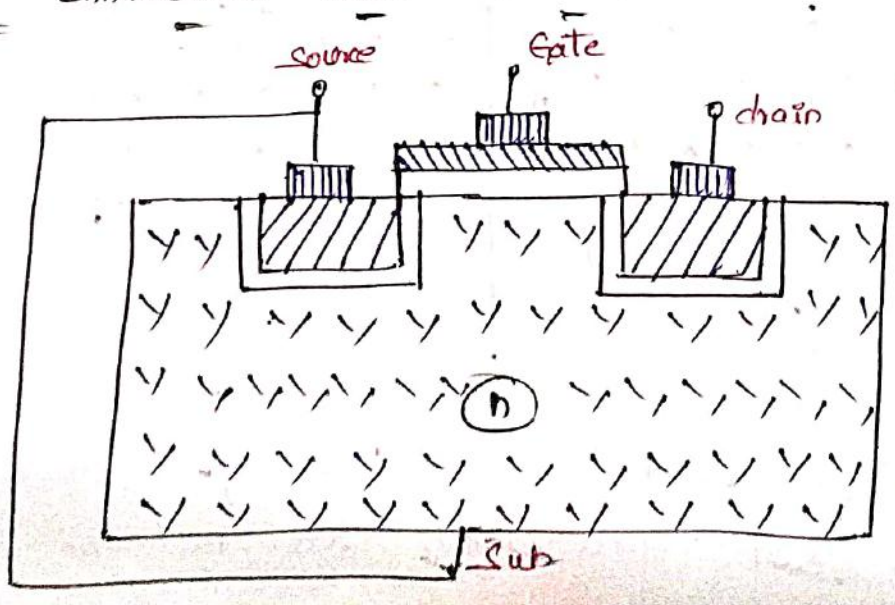
* The typical values for enhancement mode devices

$V_t = 1V$ for $V_{DD} = 5V$

(or) In general $V_t = 0.2V_{DD}$

Depletion mode MOS Transistor Action:

P MOS Enhancement mode Transistor:



* In p MOS enhancement mode of Transistor, the substrate is of n-type material & the source & drain diffusions are p-type.

* by the application of -ve voltage of suitable magnitude ($>|V_t|$) b/n gate & source will give to the formation of a channel (p-type) b/n source & drain & current may then flow, if the drain is made -ve w.r.t to source.

* In this case, current is caused by holes as applied to e⁻'s.

* The PMOS Transistors are inherently slower than NMOS since $\mu_n = 2.5\mu_p$

$\mu_n = 650\ cm^2/Vsec$

$\mu_p = 240\ cm^2/Vsec$

Depletion mode MOS Transistor action:

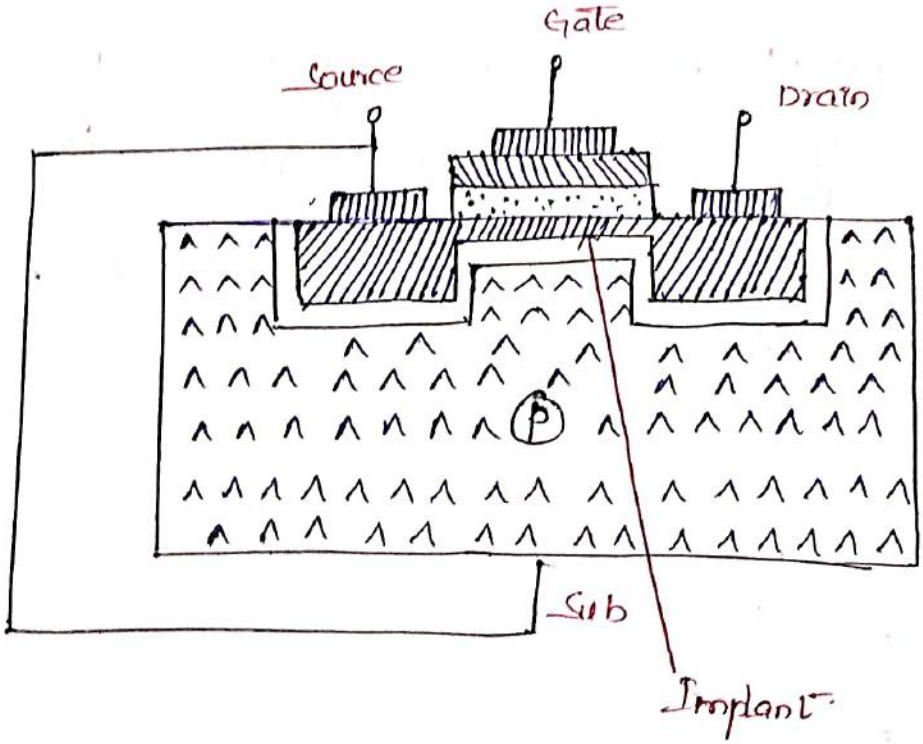


fig: N-MOS depletion mode Transistor

* N-MOS depletion mode MOS are built with p-type silicon substrates, p-MOS are built on N-type substrates. In both cases they include a thin gate oxide formed b/n source & drain regions.

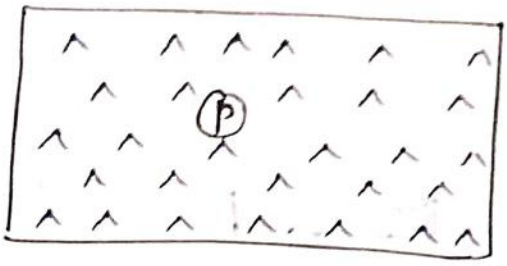
* A conductive channel is deliberately formed below the gate oxide layer & b/n the source & drain by using Ion-~~Implantation~~ Implantation. By implanting the correct ion polarity in channel region during fabrication determines the polarity threshold voltage (i.e $-V_t$ for N-channel transistor or $+V_t$ for p-channel. Transistor.

* ~~In~~ depletion mode devices are little more difficult to manufacture & their characteristics harder to control than Enhancement type. which do not require ion implantation.

* In depletion mode devices the channel is established due to implant, Even when $V_{gs}=0$ & to cause the cear a -ve voltage V_{td} must be applied b/n gate & source. $V_{td} \leq 0.5 V_{DD}$ depending on the implant & substrate bias. but threshold voltage deffernt apart.

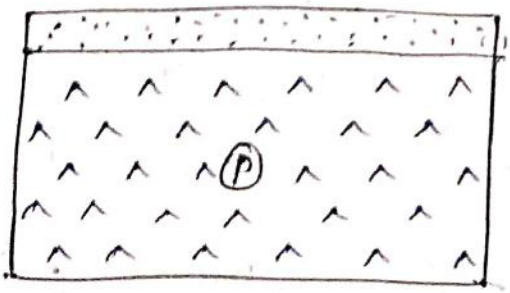
N-mos fabrication :

Step1: processing is caused out on a thin wafer cut from a single crystal of silicon of high purity in to which the required p- impurities are introduced as the crystal is grown such wafers are typically 75 to 150mm in diameter & 0.4mm thick and are dopped with boron to impurity concentration of $10^{15}/cm^3$ to $10^{16}/cm^3$. giving resistivity in the approximate range 0.25ohm cm to 2ohm cm.



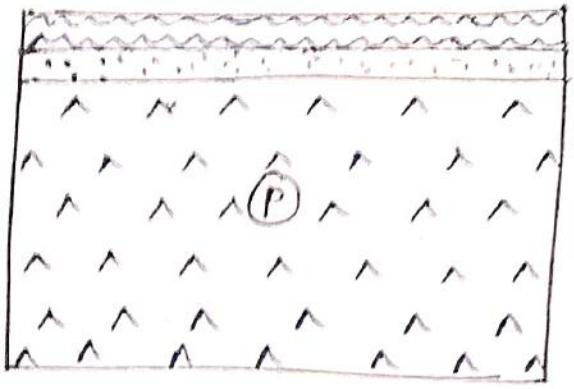
Substrate

Step2: A layer of sio2, Typically 1μm thickness is grown all over the surface of wafer to protect the surface, act as barrier to dopants during processing & provide a generally insulating substrate. on to which other layers may be deposited & patterned.



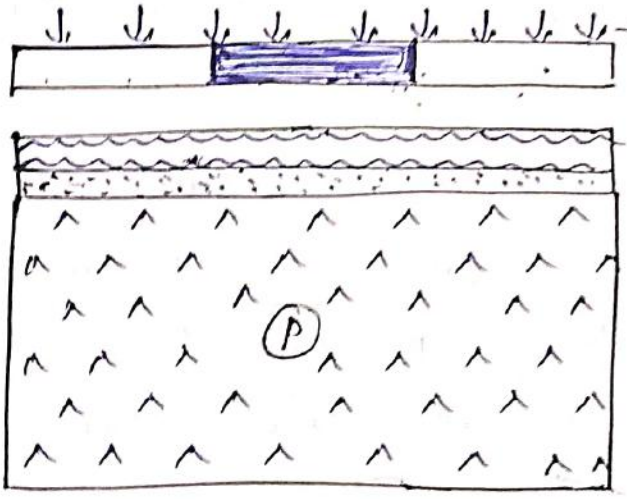
Thickness oxide (1Hm)

Step3: The surface is now covered with a photoresistive which is deposited on to the wafer &



→ photo resist (or) photofilm

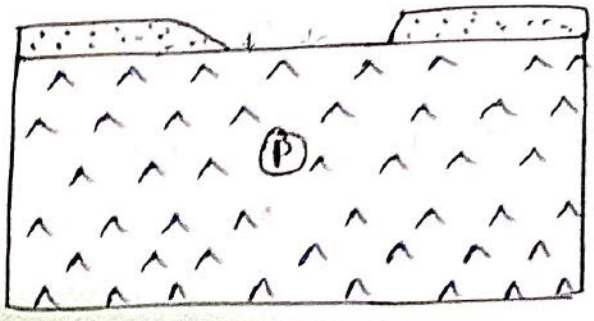
Step4: The photo resist layer is then exposed to uv light through a mask which defines those regions in to which diffusion is to take place together with transistor channels i.e. those areas exposed to uv radiation are polymerized (hardened), but that the areas required for diffusion are shielded by the mask & remain unaffected.



→ uv light

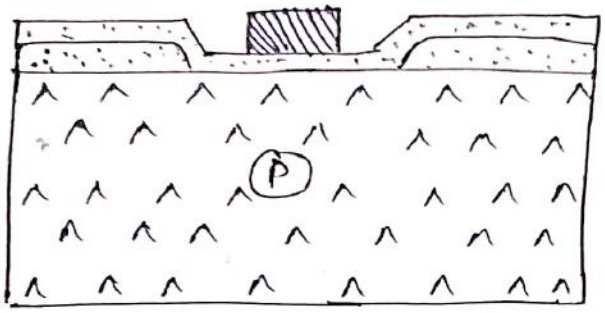
mask

Step5: These areas are subsequently readily etched away together with underlying silicon dioxide so that the wafer surface is exposed in window defined by mask



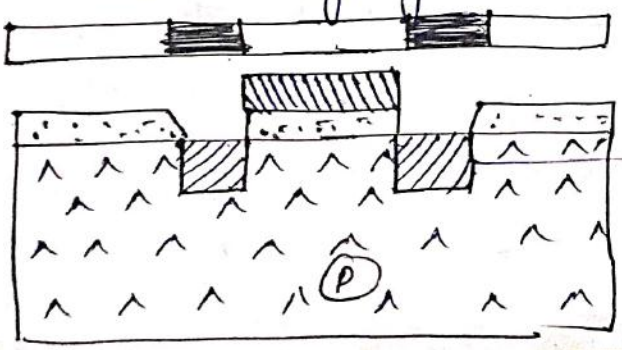
window in oxide

Step6: The remaining photoresist is removed & a thin layer of SiO_2 is grown over the entire chip surface & then polysilicon is deposited on top of this to form the gate structure. the polysilicon layer consists of heavily doped polysilicon deposited by "chemical vapour deposition (CVD)". In the fabrication of fine pattern devices precise control of thickness, impurity concentration & resistivity is necessary.



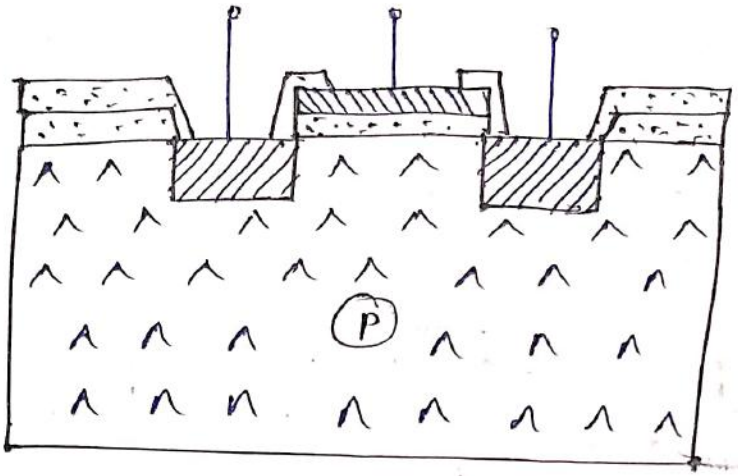
poly silicon (1-2μm) on thin oxide (800-1000°A)

Step 7: Further photoresist coating and masking allows polysilicon to be patterned & then the thin oxide is removed to expose areas to to which n-type impurities are to be diffused to form source & drain. Diffusion is achieved by heating the wafer to a high temperature & passing a gas containing desired n-type impurity (phosphorous) over the surface. Note that the polysilicon with underlying thin oxide & thick oxide acts as masks during diffusion. the process is called self aligning.



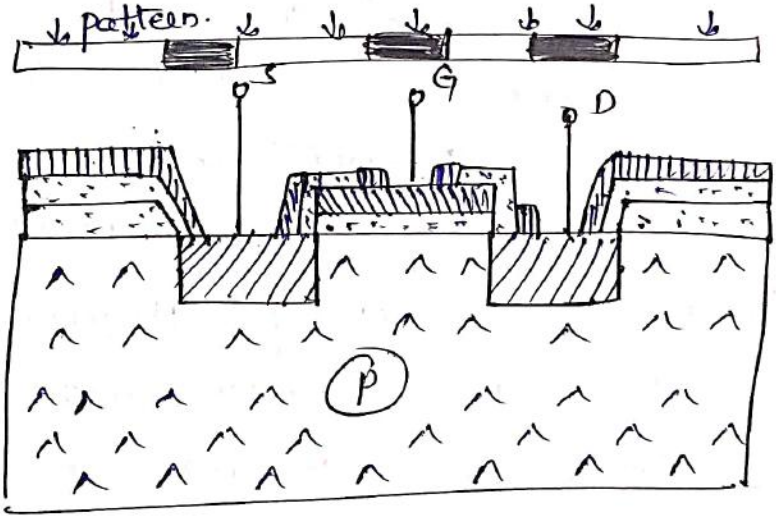
n^+ diffusion (1μm deep)

Step8: Thick oxide (SiO_2) is grown overall again & is then masked with photoresist & etched to expose selected areas of the polysilicon gate & source & drain areas where connections (i.e contact cuts) are to be made.



contact holes (cuts)

Step9: The whole chip then by metal (aluminium) deposited over its surface to a thickness typically of 1 μm. This metal layer is then masked & etched to form the required interconnection pattern.



uv light

pattern metalization (aluminium 1μm)

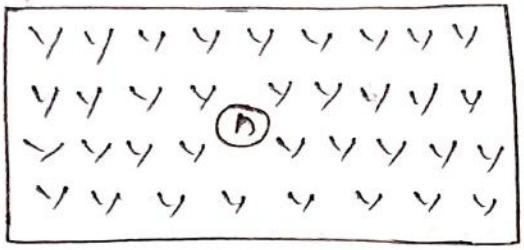
The process revolves around the formation or deposition & Patterning of 3 layers seperated by SiO_2 insulation. the layers are diffusion with in the substrate, polysilicon on oxide on the substrate & metal insulated again by oxide.

P-MOS fabrication:

Step1:

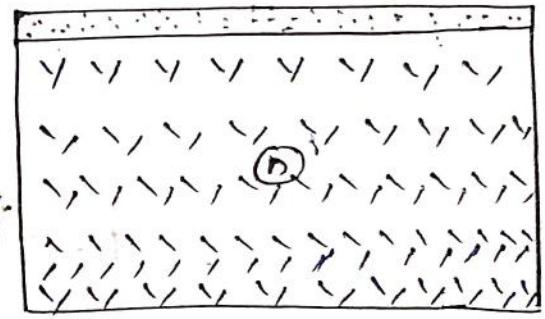
* N-type substrate should be required for p-MOS Transistor.

* Substrate has to be there should be a p-type Impurities are introduce as a crystal of pure silicon wafer surface with thickness of 0.4mm & 75-150mm diameter. when N-type materials are heated up to 1000 to 1500°C & then bubbles are formed

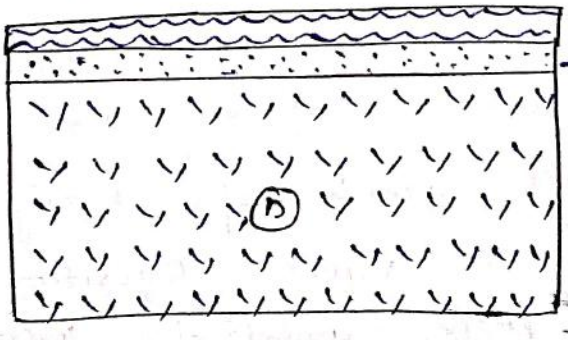


in the form of steam thus n-type Impurities are Inserted in to pure silicon material.

Step2: In this silicon dioxide layer is coated above the surface with thickness of 1μm this process is called oxidation & this layer is called oxidation layer (or) Insulator. This is used to protect the substrate from atmospheic conditions.

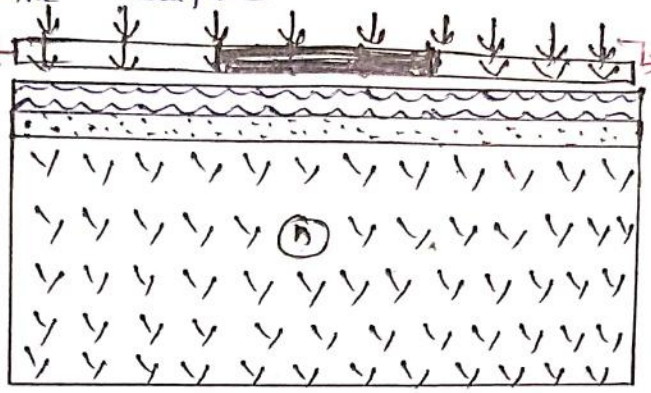


Step3: The oxidation layer is coated with thin layer of photo resist. This will be acts as mask of oxidation layer. this process is called photo lithography.

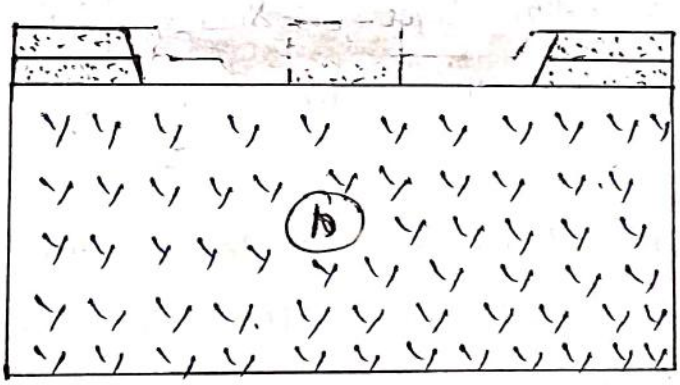


Step4:

The photo resistive layer is exposed uv light through a mask. which defines source & drain regions in to which diffusion is takes place. through etching process. so that to pass in to the surface.



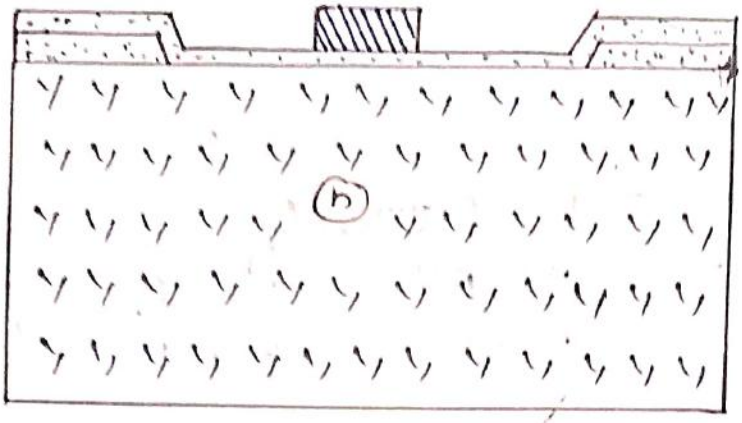
Step5: These areas are subsequently etched away together with the underlying SiO_2 so that the wafer is exposed in the window by mask. After protecting uv rays it removes the areas to create 2 regions layer.



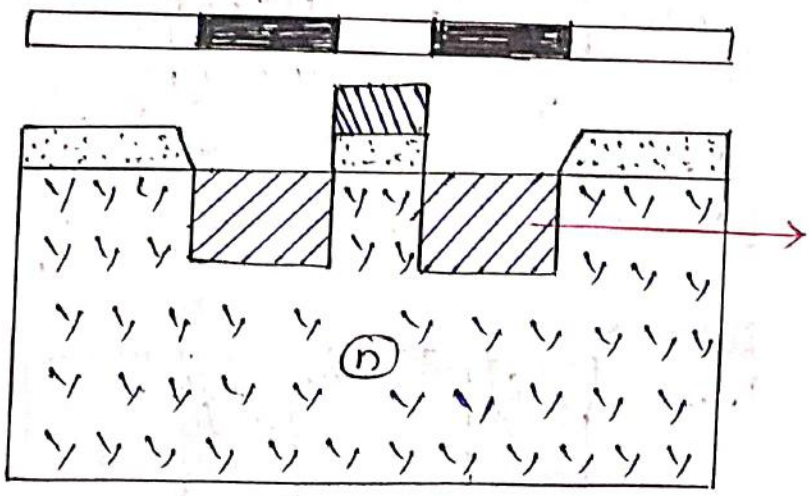
window in oxide.

Step6: The remaining photo resistor is removed & a thin layer of SiO_2 is grown over the entire chip surface & then polysilicon is deposited on top of this to form the gate. the polysilicon layer consists of heavily doped polysilicon deposited by chemical vapour deposition (CVD). In this pattern devices precise control of thickness, Impurity concentration

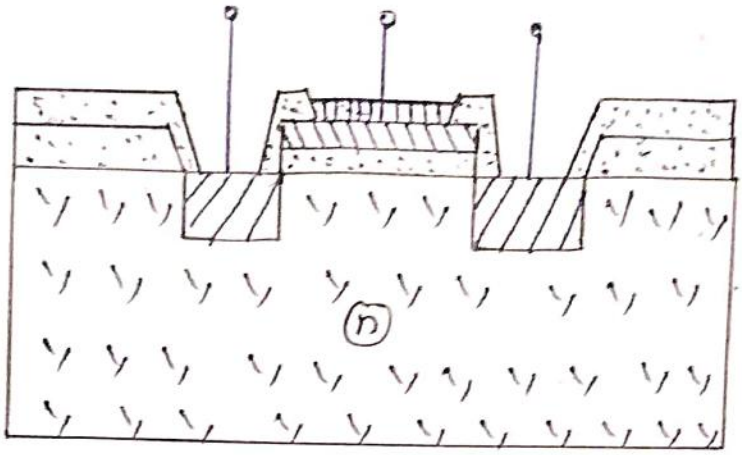
resistivity is necessary.



Step 4: Further photo resist and masking allows the polysilicon is to be patterned. & then thin oxide is removed to expose areas in to which p-type impurities are to be diffused to form drain & source. Diffusion is achieved by heating the wafer to a high temperature & passing a gas containing the desired p-type impurity (ex: boron) over the surface. Note. that the polysilicon with underlying thin oxide. & thick oxide acts as masks during diffusion. the process is self-aligning.

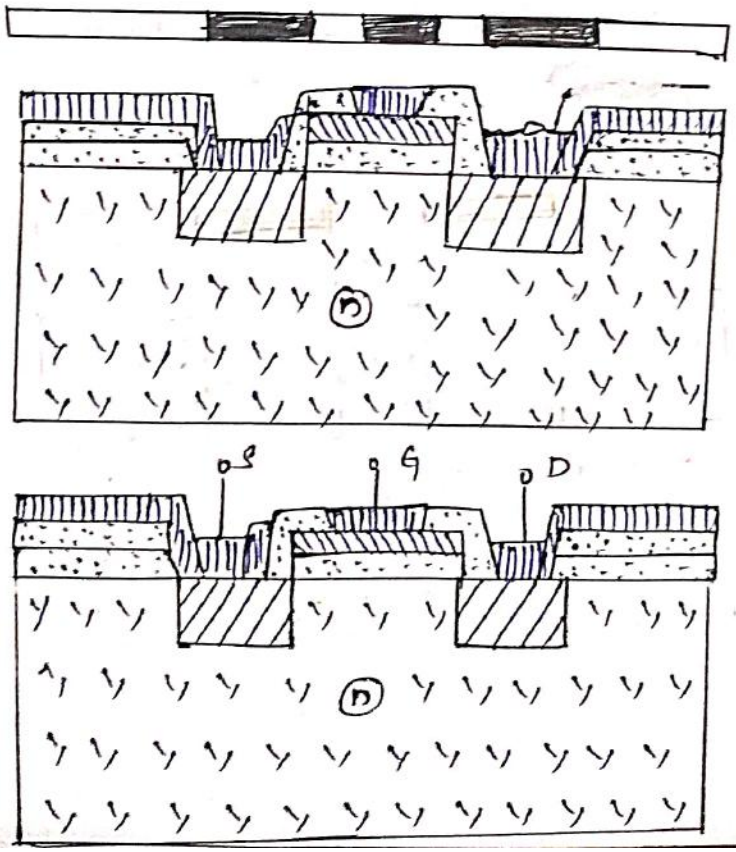


Step 8: Thick (SiO_2) oxide is grown over all again & then is masked with photo resist & etched to expose selected areas of polysilicon gate & drain & source area where connections are to be made.



contact holes (cuts)

Step 9: The whole chip then by metal (aluminium) deposited over its surface to a thickness typically of 1 μm this layer is then masked & etched to form the required interconnection pattern



Patterned metalization (aluminium 1 μm)

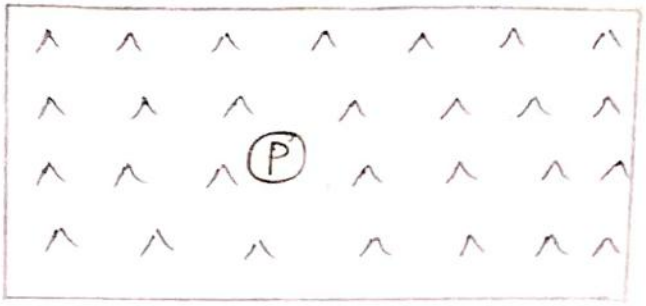
CMOS fabrication:

* CMOS is a complementary metal oxide semiconductor in which combines p-MOS and N-MOS Transistors In a complementary way ie. CMOS technology uses both NMOS & P MOS Transistors. The transistors are arranged in a structure formed by two complementary networks.
  * pull up network is complement of pulldown network.
  * pull up transistor are connected in parallel, then pulldown Transistors are connected in series (or) vice versa.
  * parallel → series, series → parallel.
* CMOS are fabricated in a various kinds of process they are :
  1) N-well process
  2) p-well process
  3) Twin tub process.
* In these 3 process. Most of the Industries they are using N-well based CMOS fabrication process because they have lot of advantages.

Basic steps Involved in N-well process:

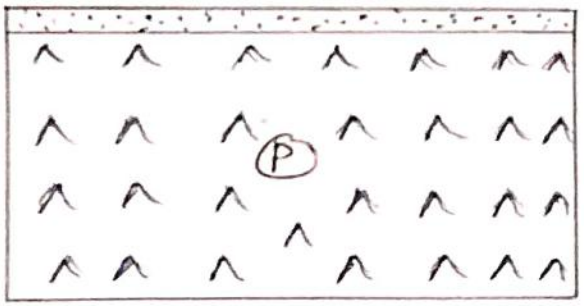
Step 1): SI substrate :

* All the Ic chips are fabricated through a SI substrate
* By adding p-type Impurities (ie boron is a trivalent Impurity) to SI substrate by a process called doping. & It converts the SI substrate in to p-substrate.



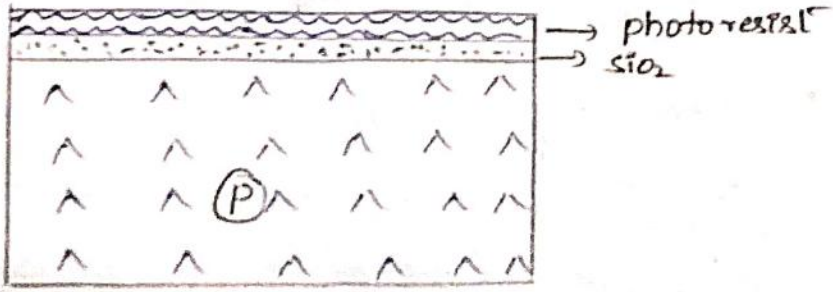
Step2 : oxidation:

oxidation is a process of converting silicon in to silicon dioxide (SiO_2). because silicon is a Insulator (ie it does not conduct). Exposing ~~to~~ high purity oxygen & hydrogen at appox. 1000°c by heating. SiO_2 is formed. & it is applied on the top of the surface to protect the substrate surface from external atmospheric conditions.

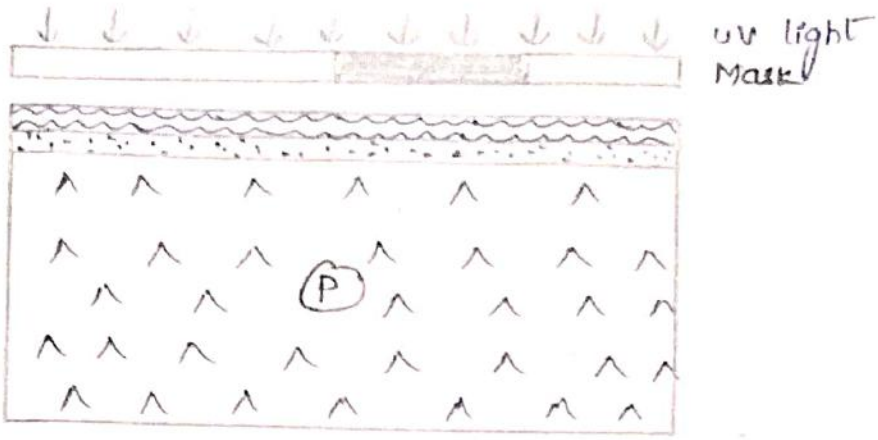


Step3: photoresist:

photo resist is a light sensitive organic polymer softens when exposed to light which is used to transfer the desired pattern.



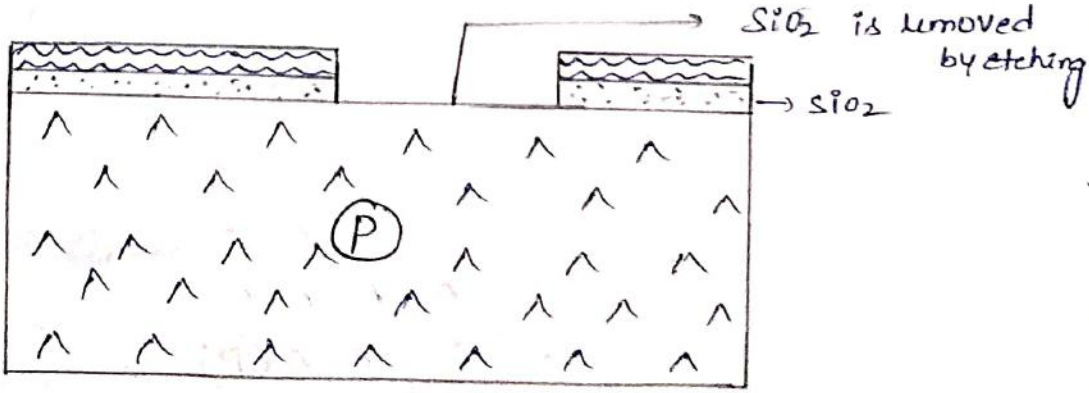
Step4 : Masking



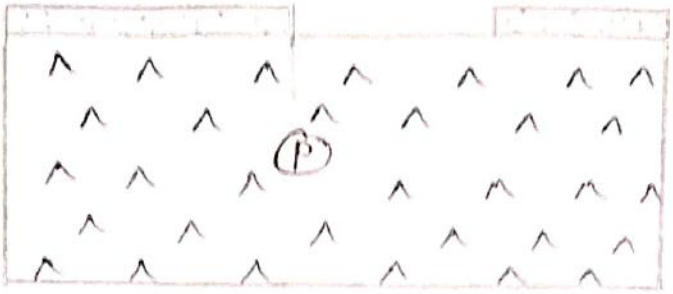
Masking is the process of removing unwanted area of SiO_2 for n-well Implantation. & this process is called photolithography. by exposing it with UV light rays. when the UV light is Expose to photoresist areas is softened & in Masked area is hardened because there is no exposure of UV light to SiO_2.

Step5 : Etching :

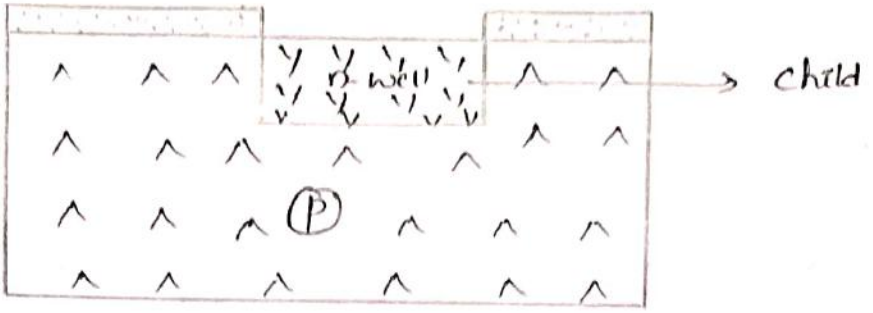
photo resist are removed by treating the wafer with acid or basic solution. SiO_2 is selectively removed from areas of wafer that are not covered by photoresist by using hydroflouric acid. SiO_2 is easily reacts with hydroflouric acid these unwanted or Hard area is removed and a window is formed.



remove remaining photoresist layer by using mixture of acids (Hydroflouric acid



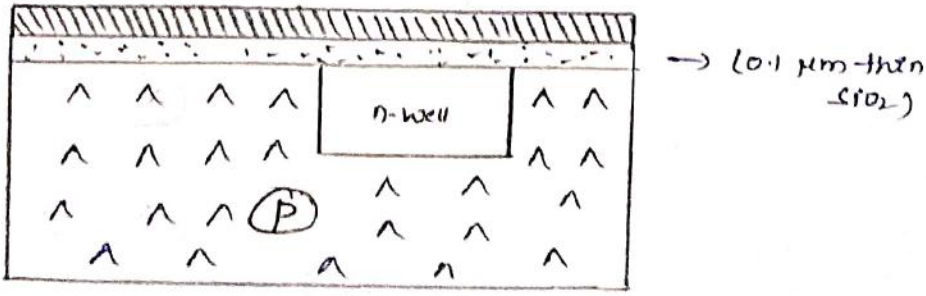
Step 6: Formation of n-well:



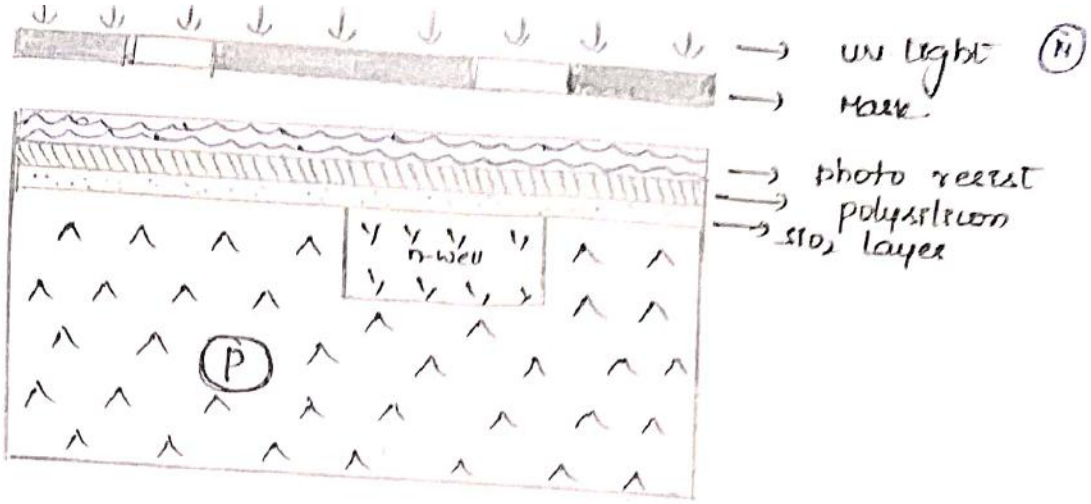
n-well is formed by dopping n-type Impurities (i phosphrous pentavalent) with diffusion or ion Implantation. Diffuse n-type Impurity to window to form n-well it is called child or local substrate. p-substrate is called Global substrate.

Step 7: removal of SiO_2:

remove remaining SiO_2 by using Hydroflouric acids. deposit thin SiO_2 layer & Deposit polysilicon for formation of Gate structure.



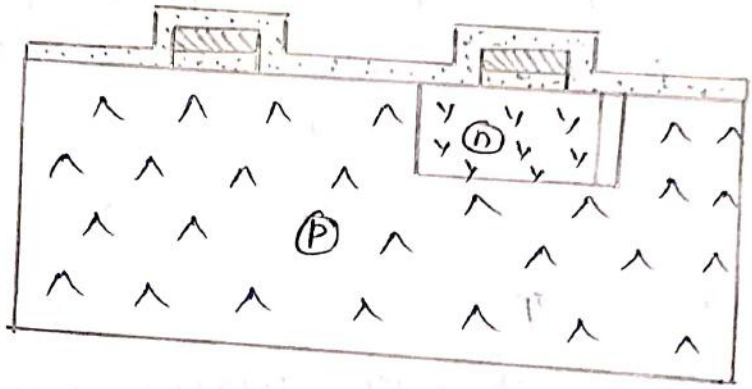
Steps:



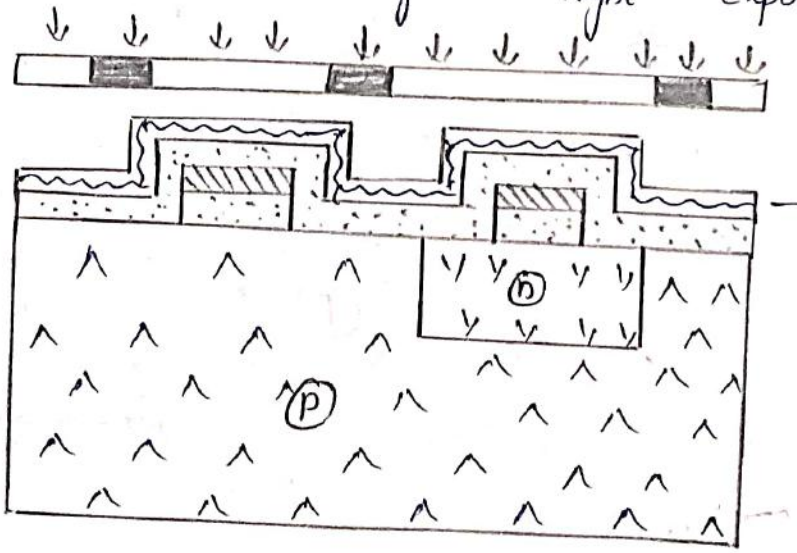
(Photoresist layer + uv light + Masking) are formed applied for the formation of NMOS & PMOS.

Step 9:

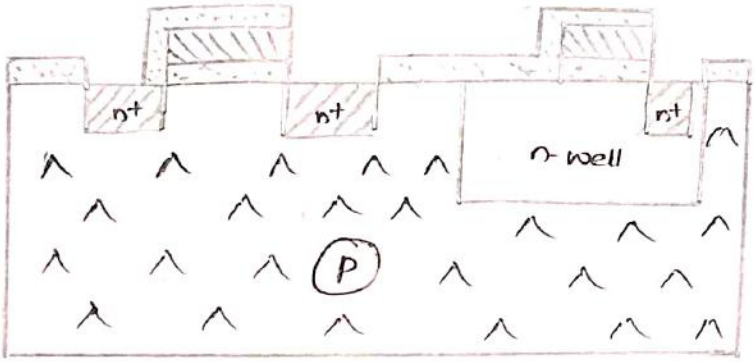
Etching of photo resist layer and remaining polysilicon & thin SiO_2 layer & cover the surface with SiO_2 layer. In order to form n^+ regions (Source & drain).



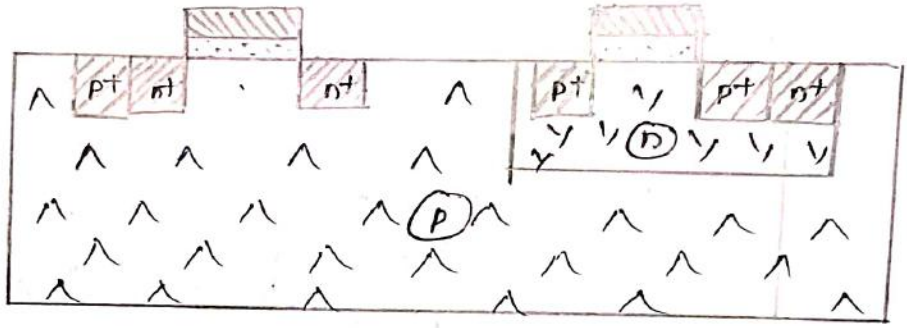
Step 10: photo resist layer + Masking + uv light exposion to form n^+ regions.



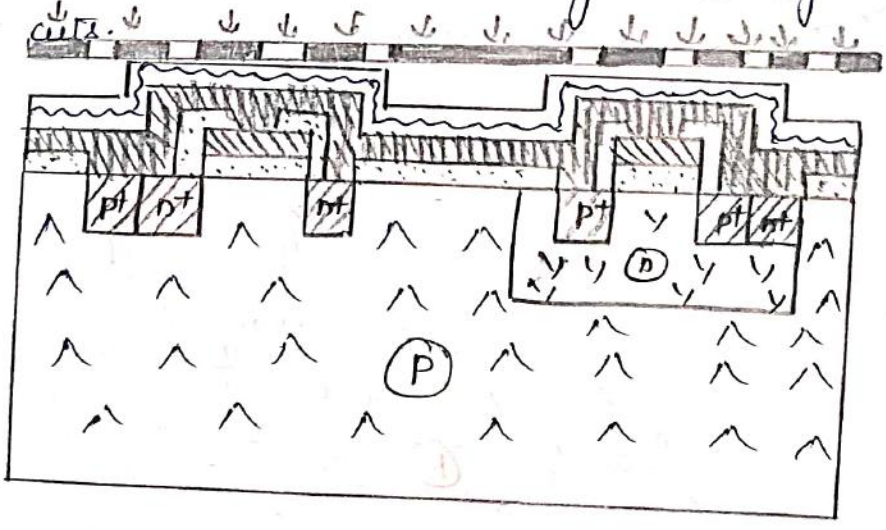
Step 11: Etch the masked regions & diffuse n-type Impurities



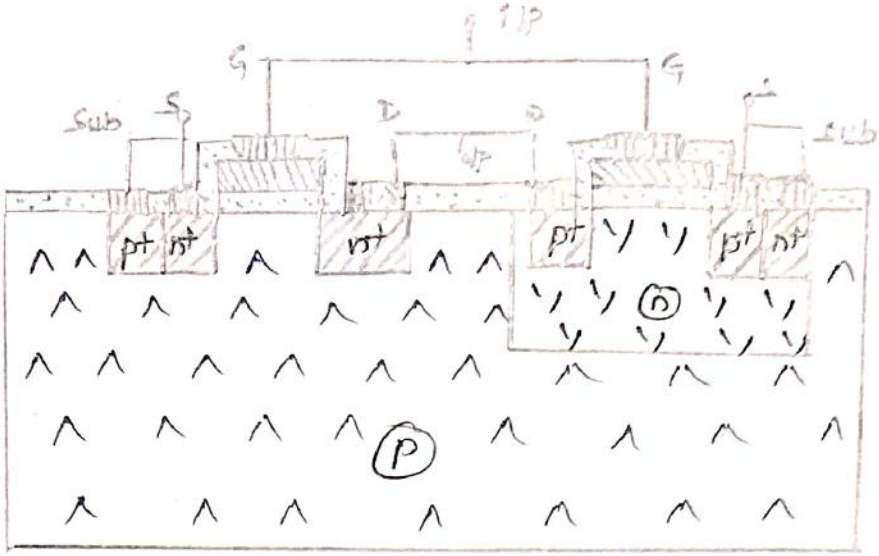
Step 12: Remove remaining SiO_2 layer. To form p^+ regions follow same. [SiO_2 layer + photo resist + Masking + UV light exposing. Send p-type Impurity gas through windows by heating P-substrate at high temperature. then p^+ regions are formed



Step 13: cover the chip with thick SiO_2 layer for contact cuts [SiO_2 + photo resist layer + Masking + UV light exposure) to form contact cuts.



Step 14:

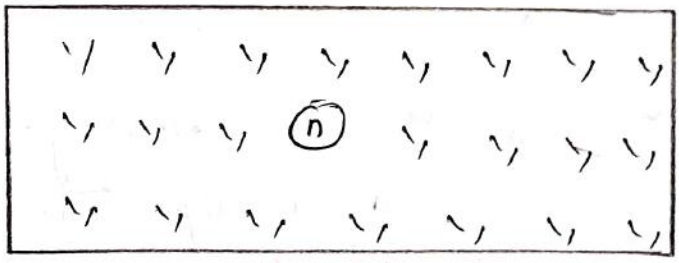


After etch the metal give connections this is n-well process.

2) P-well process:

Step 1: SI substrate:

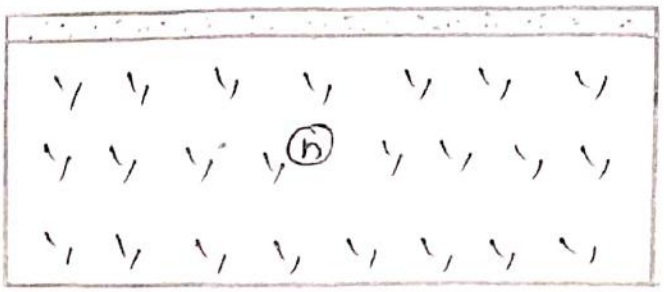
By dopping n-type Impurities (phosphorous Impurity) to SI substrate so SI substrate converted to N-Substrate.



Substrate.

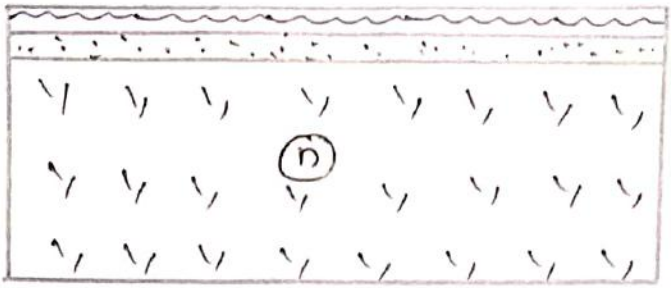
Step 2: oxidation:

exposing of high purity oxygen & hydrogen at 1000°c by heating SiO_2 is formed. & it is deposited on the top of the surface to protect the substrate from external atmospheric conditions.



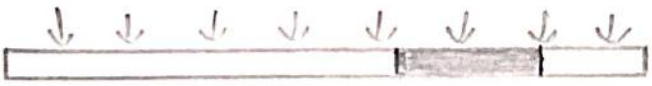
SiO_2 layer (1 μm thickness)

Step 3: photoresist layer is deposited on the top of SiO_2 which is used to transfer the desired pattern.

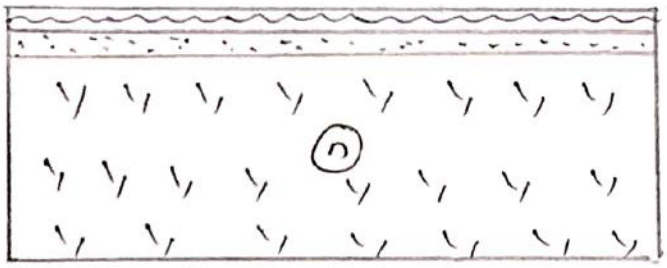


→ photo resist layer

Step 4: Masking:

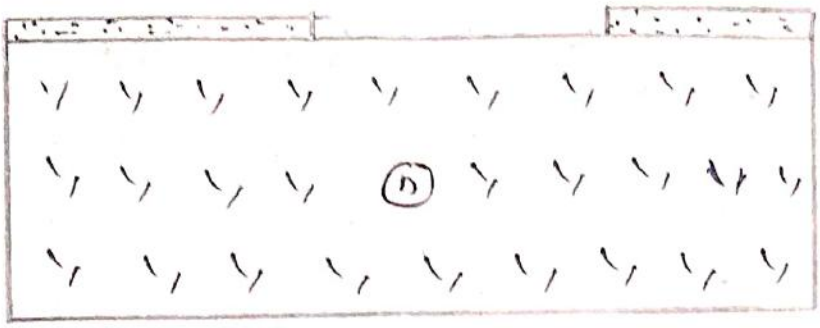


· Mask.

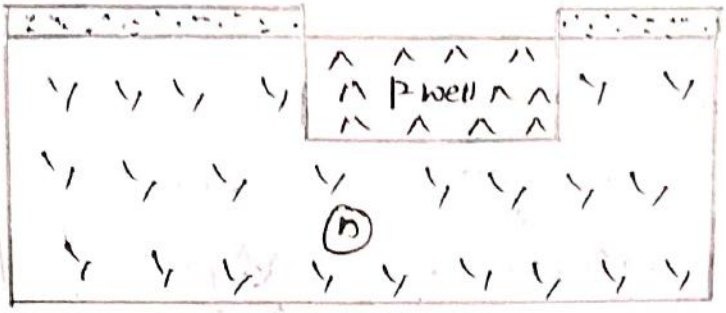


Masking is a process of removing unwanted area of SiO_2 to form p-well & then uv light is expose to photoresist layer & the masked area is softened & remaining photoresist areas are softened

Step 5: Etching: The Masked area is removed by adding hydroflouric acid. so SiO_2 is reacts with hydroflouric acid & a window is formed. & the remaining photoresist areas removed by treating the wafer with acid or base solution.

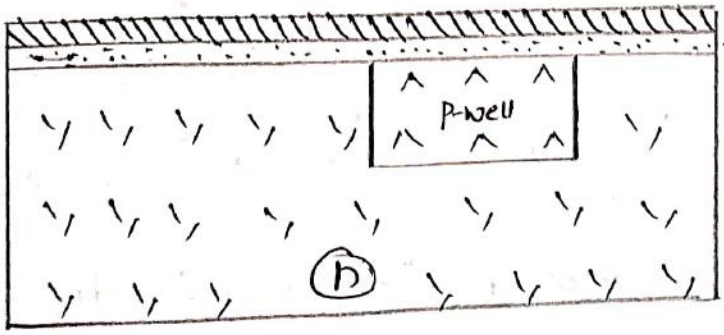


Step 6: formation of p-well :



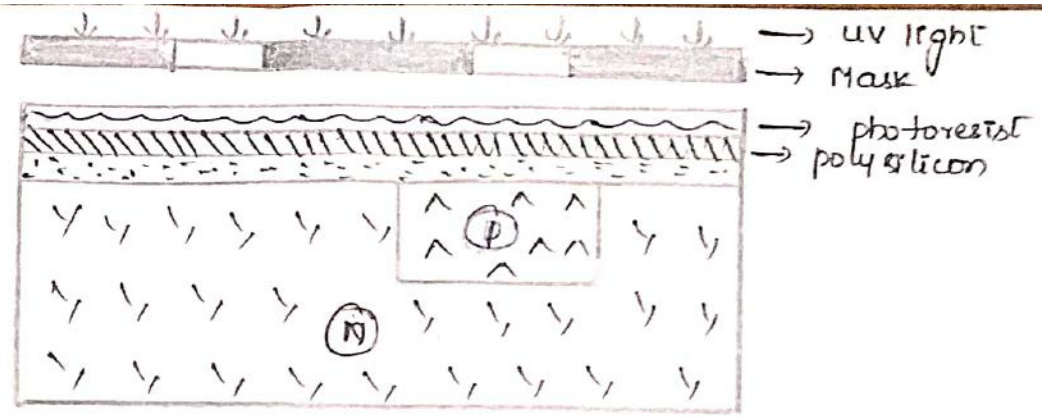
p-well is formed by doping p-type Impurities (boron) with diffusion or ion implatation process. to form p-well it is called child or local substrate. n-substrate is called Global or parent substrate.

Step 7: remove remaining SiO_2 layer using Hydroflouric acid. Deposit thin SiO_2 layer & Deposit polysilicon layer for formation of Gate structure.



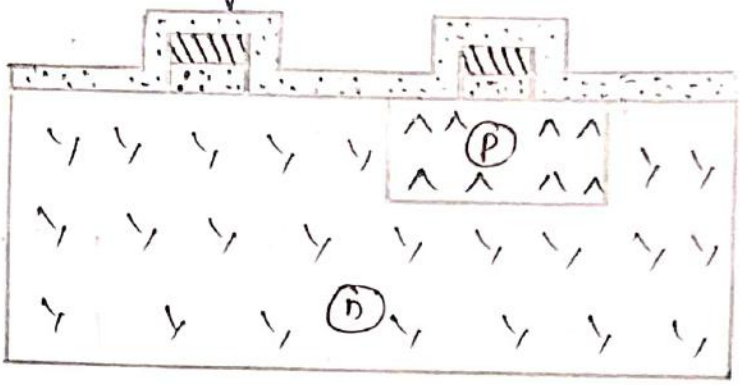
→ Thin SiO_2 layer (0.1Hm)

Step 8: (photoresist layer + uv light + Masking) for gates of NMOS & PMOS

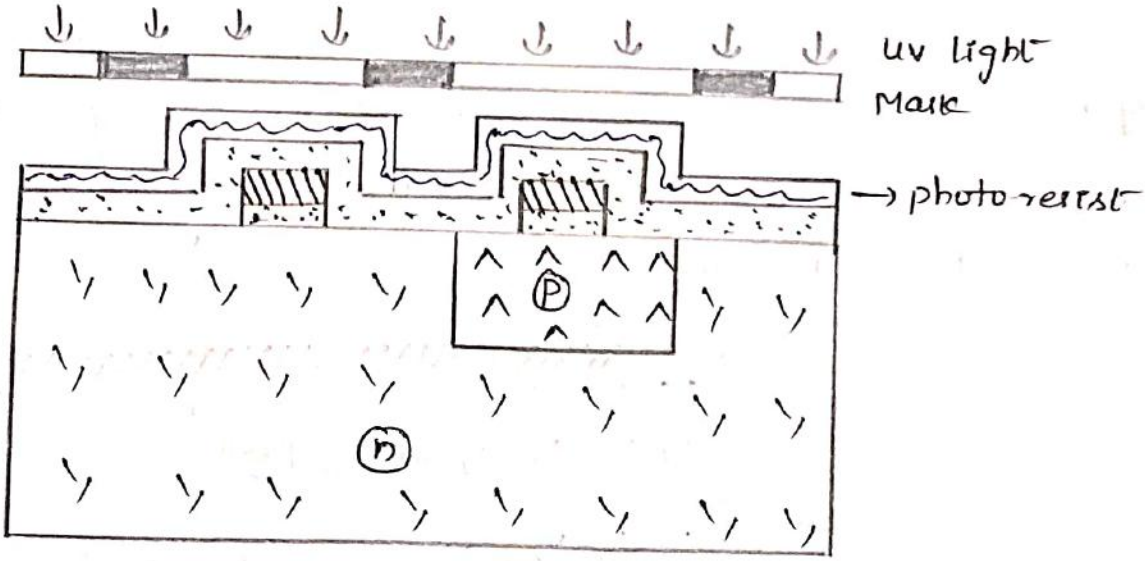


Step9:

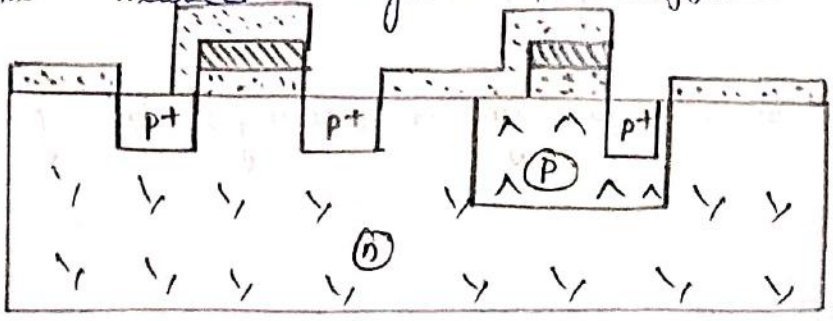
Etching of photoresist layer and remaining polysilicon & thin SiO_2 layer is used. In order to form p^+ regions (source & drain) & gate structure.



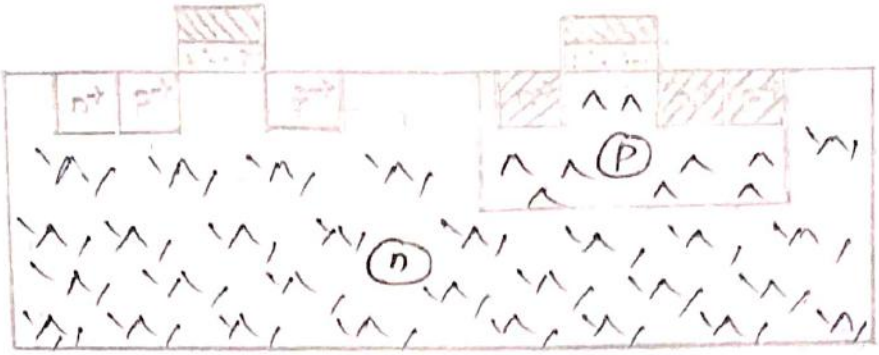
Step10: photo resist layer + Masking + uv light exposion to form p^+ regions.



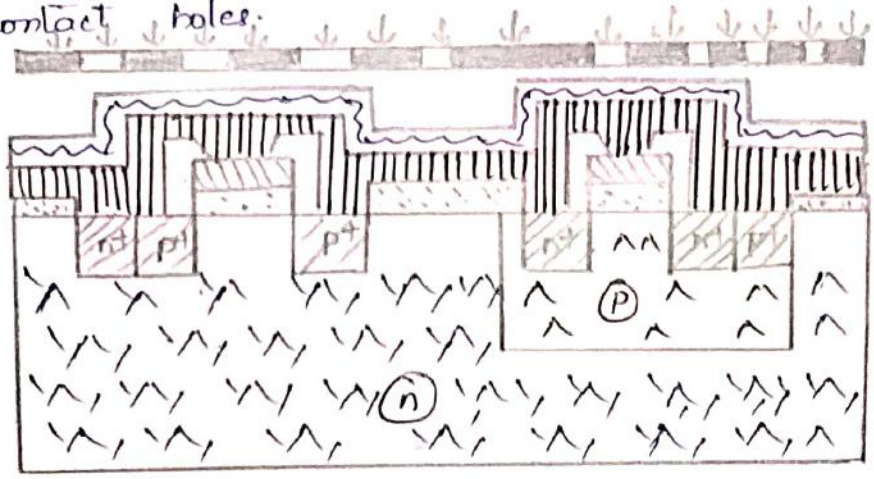
Step11: Etch the masked regions & diffuse p-type Impurities



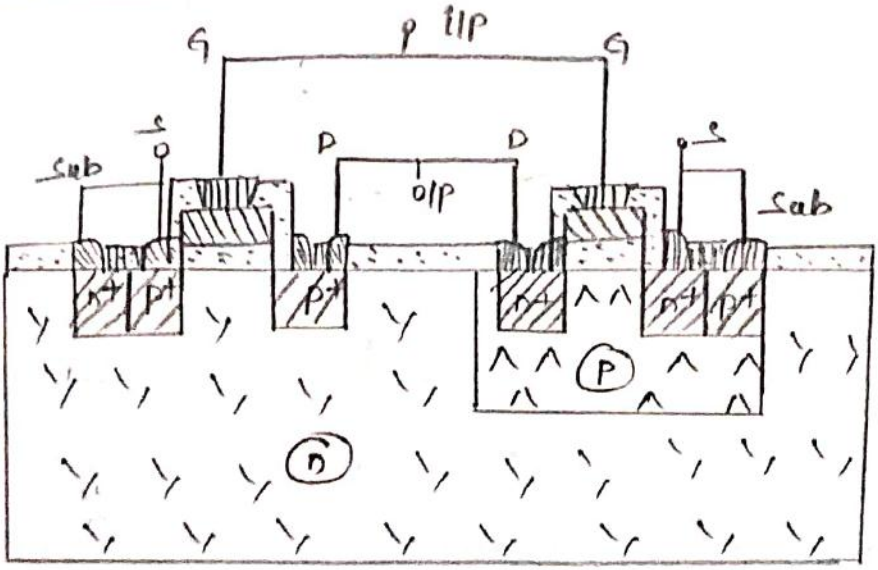
Step12: Remove remaining SiO_2 layer to form n+ regions follow same [SiO_2 layer + photo resist + Masking + uv light exposing] & pass n-type Impurity gas through window by heating n-type substrate at high temperature then n+ regions are formed.



Step13: Cover the whole chip with thick SiO_2 layer for contact cuts [SiO_2 + photoresist layer + Masking + uv light exposing] to form contact holes.



Step14:



After etching the metal it gives connections. This is p well process of CMOS

Advantages of n-well process over p-well process:

* n-well CMOS are superior to p-well because of lower substrate bias effects on Transistor threshold voltage.
* lower parasitic capacitances associated with source & drain region
* Latchup problems can be considerably reduced by using a low resistivity epitaxial p-type substrate.
* However n-well process degrades the performance of poorly performing p-type transistor.

3) twin tub process(or) twin well technology:

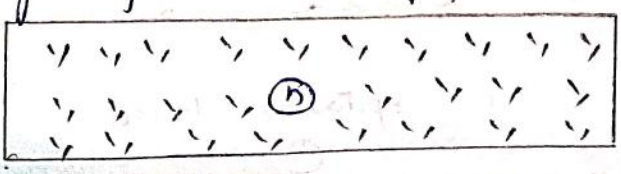
Fabrication of CMOS using Twin tub process or twin well self aligned process: silicide Material

* The N-MOS and PMOS fabrication process is having some issues are 1) Mutual coupling (or) cross talk. the reason is on p-type substrate to form N-well to have p-MOS
2) Latchup of MOS. it generates the issues regarding speed of operation of CMOS.

* to avoide (or) to reduce the issues we use twin tube fabrication process
* In twin tube process there will be two different wells N-well & p-well. In N-well PMOS fabrication is done & In p-well there will be NMOS fabrication.

Basic steps of fabrication of twin tub process:

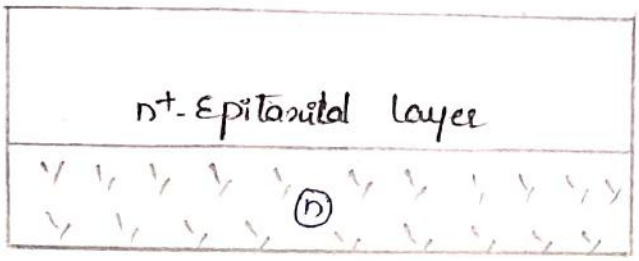
Step1: The silicon substrate is converted in to N-substrate by adding of dopping of n-type impurities to Si substrate.



* by using N-type substrate the resistivity of substrate should be higher.

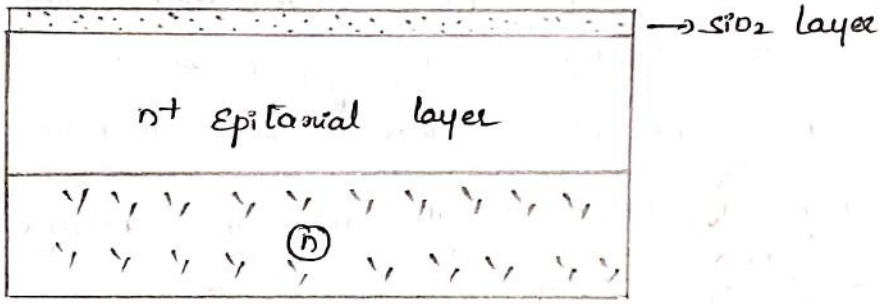
* Higher the resistivity lesser the current through substrate. & It can be done by less dopped N-type material.

Steps:



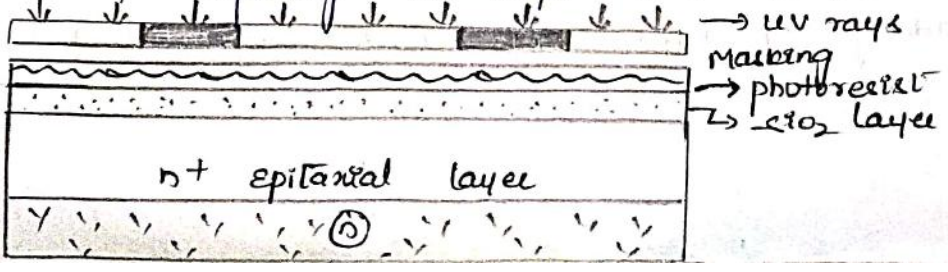
on N-type substrate we grow Epitaxial layer of n+ type material with higher concentration of doping to N-type substrate. that is having less resistance compared to N-substrate.

Steps:

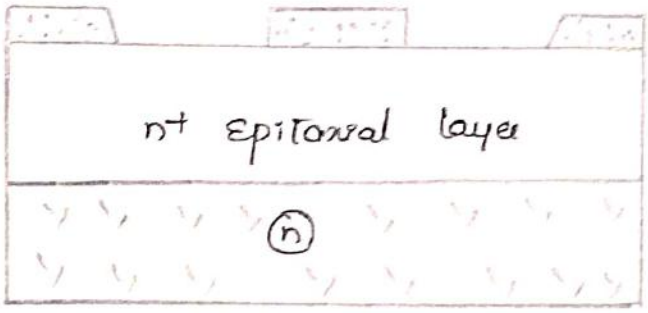


After that, substrate is subjected to oxidation and we grow SiO_2 layer. to form (or) to provide to well's i.e n-well & p-well

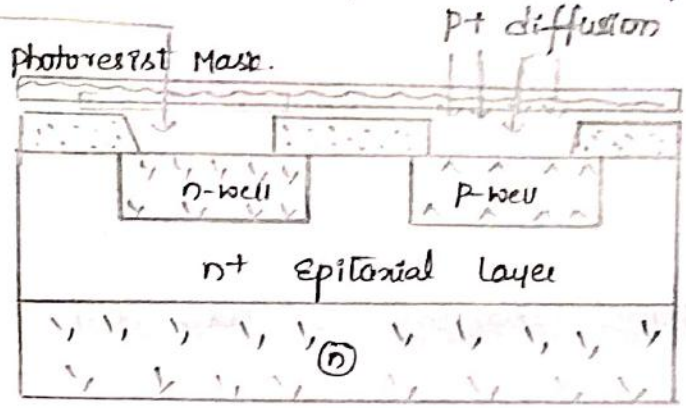
steps: to form wells. The oxidation layer (SiO_2 layer) is coated with photoresist layer by a process called photolithography. & then to form well regions in substrate Masking is done. to remove particular areas. later by using etching process the photoresistive layer is removed by exposing UV rays in to surface.



Step5: SiO_2 layer is etched using masking. two windows are formed, one for n-well & another for p-well.

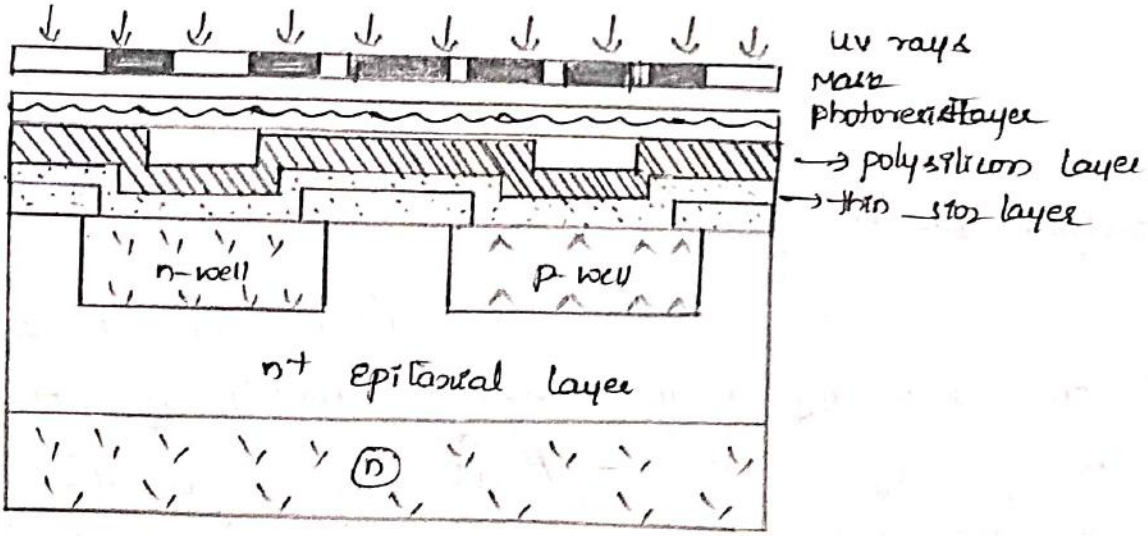


Step6: 1st window is covered by photoresist mask. then p-type impurities diffused to form p-well. with high doping concentration



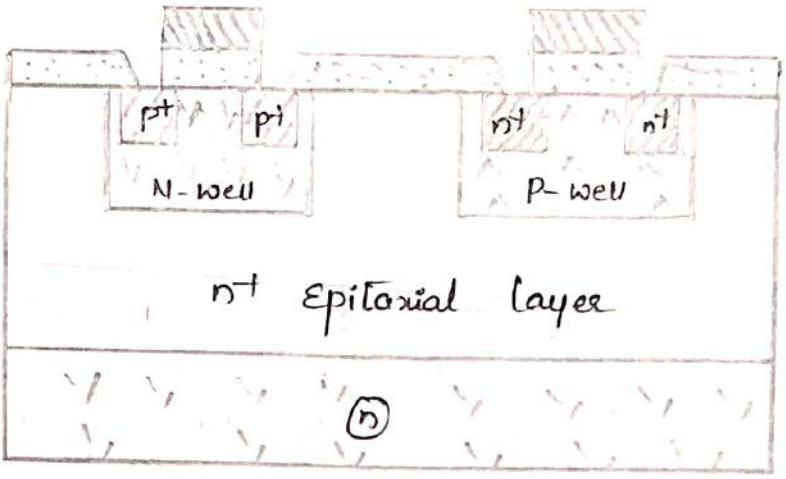
The n-type impurities ~~diffused~~ diffused to form n-well by covering 2nd window with photoresist mask.

Step7: * Grow thin SiO_2 layer by thermal oxidation for gate terminal. (CVD) Grow polysilicon layer for photolithography & pattern making. can be done.



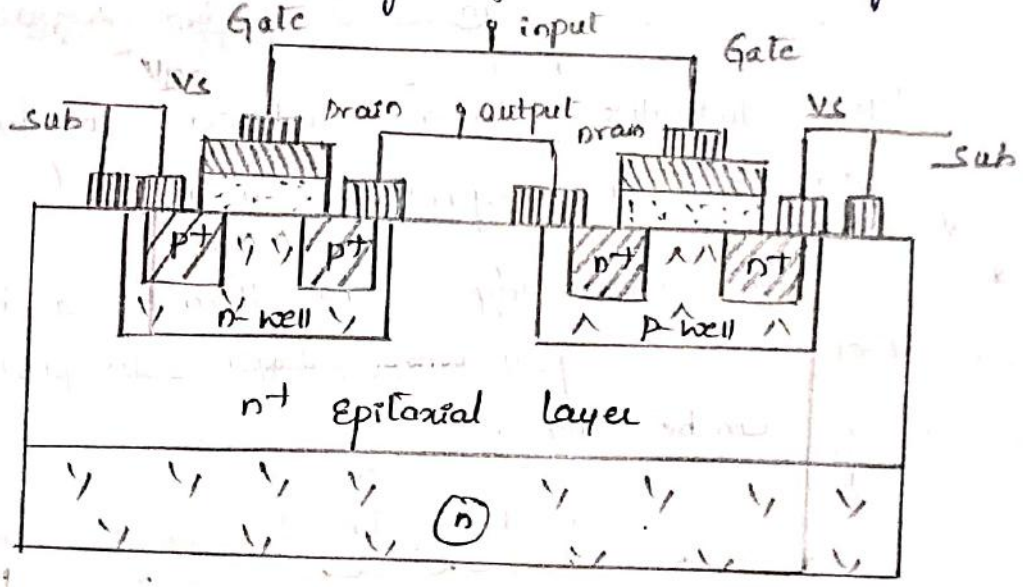
by etching process. two gate terminals are placed in p-well & n-well region. for the implantation of source & drain regions. these

terminals are dopped with n^+ diffusion Impurities in-to p-well and p-type Impurities diffused in-to N-well



Steps:

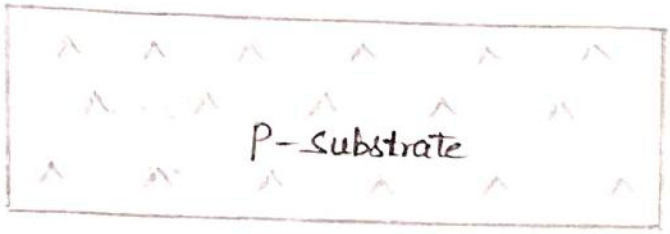
The purpose of interconnecting the terminals the metalization will takes place through a mask windows. to form contact cuts. After this. a proper shapping of device may takes place



BiCMOS process:

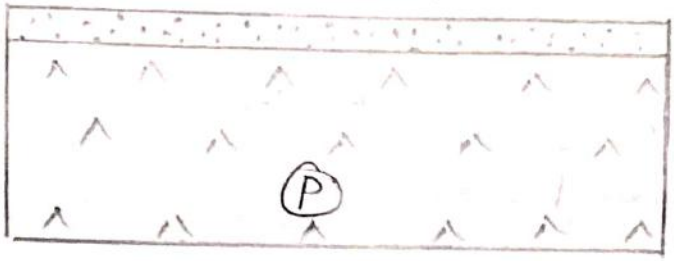
BiCMOS is a Bipolar CMOS having one N-MOS & one p-MOS & one NPN BJT. The driving capability of MOS Transistor is less because of limited current sourcing & sinking capabilities of Transistors. To drive large capacitive loads BiCMOS technology is used. As this technology combines of both bipolar & CMOS transistors in single IC. to achieve VLSI ckts with speed power density performance.

Step1: SI substrate is converted into p- substrate by lightly doping concentration p type Impurities to SI substrate

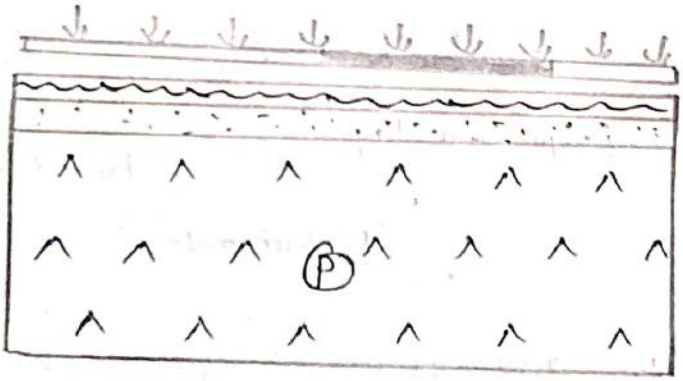


Step2: oxidation:

P-type substrate is coved with oxide layer (SiO_2). by adding high purity of oxygen & hydrogen to Si to form SiO_2.

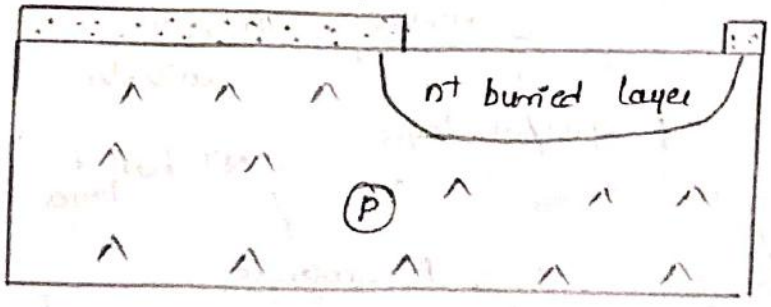


Step3:

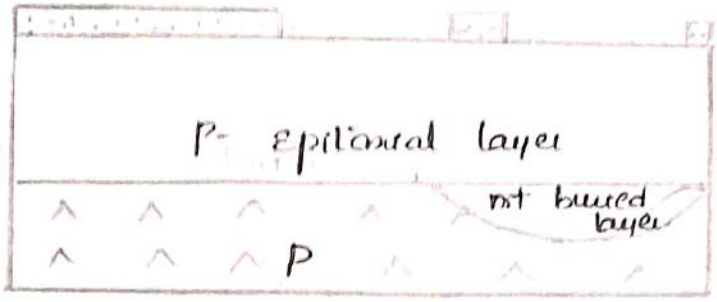


Step4: The (photoresist + SiO_2 + Masking + Etching) is done to form window to diffuse

The antimony (or) silic are diffused into p- substrate to form n^+ buried layer. by heavily dopped n-type Impurities.

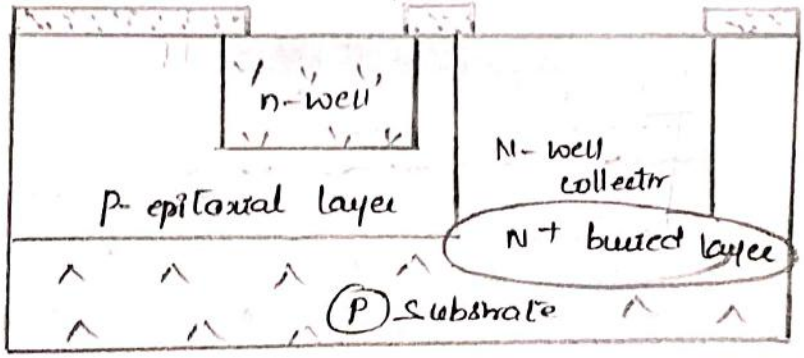


Step5: p-type epitaxy layer is grown on the entire substrate of thickness 4 μm by using Implantation at temp of 350°-400°c

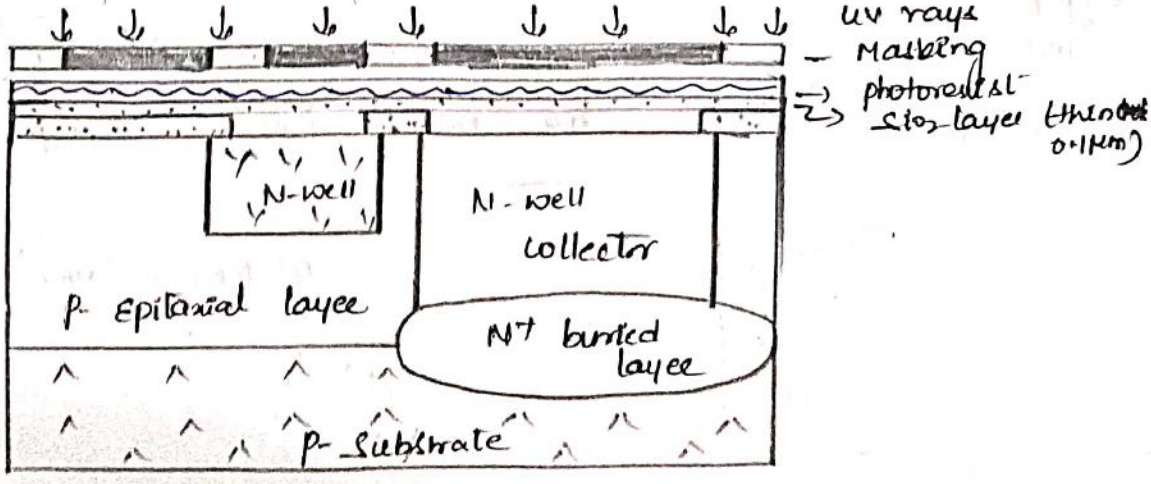


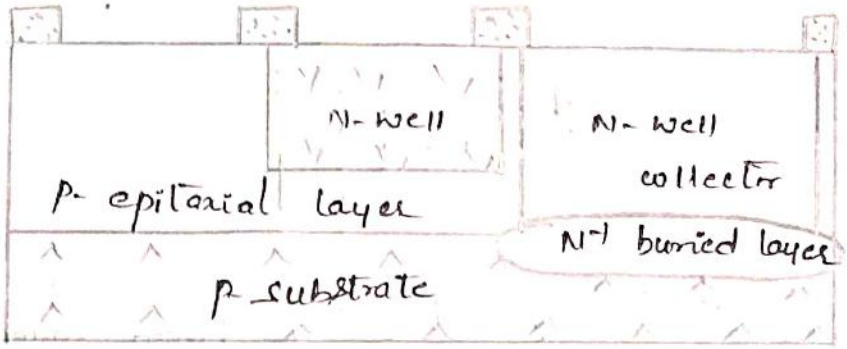
The Entire surface is covered by SiO_2 layer. photolignography, Masking, etching will be done two windows are opened through the oxide layer.

Step6: Through the two windows n-type impurities are diffused to form n-wells.

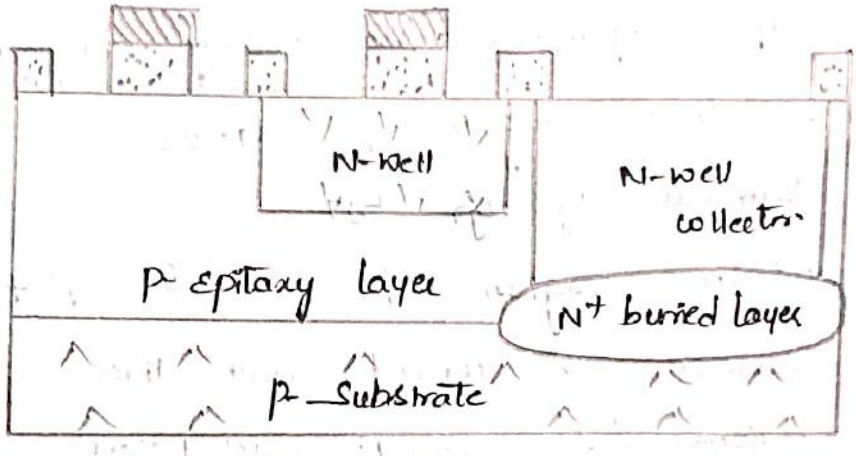


Step7: again (SiO_2 layer + photoresist layer + Masking + uv ligh exposure) can be done to form n-MOS, pMOS and NPN BJT. the three windows are formed.



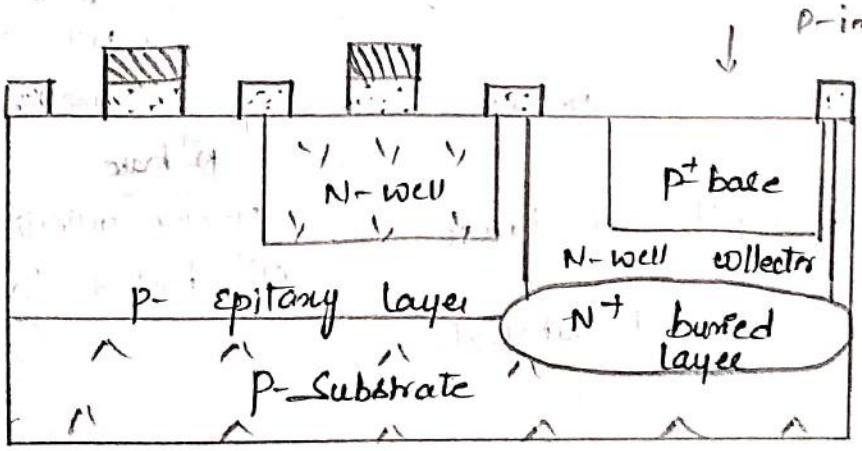


step8:



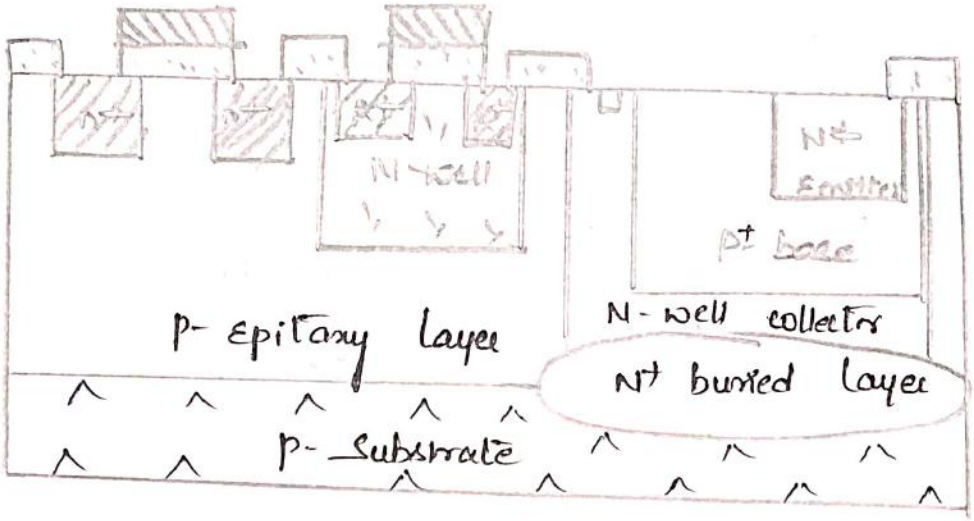
The Entire Surface is coated with thin SiO_2 & polysilicon are patterned to form the gate terminals of NMOS and PMOS (SiO_2 + photoresist layer + Masking + etching)

Step9:



Through 3rd window the p- impurities are moderately doped to form the base terminal of BJT n-well acts like collector terminal.

Step 10:



n-type impurities are heavily doped to form

1) Source & drain region of NMOS

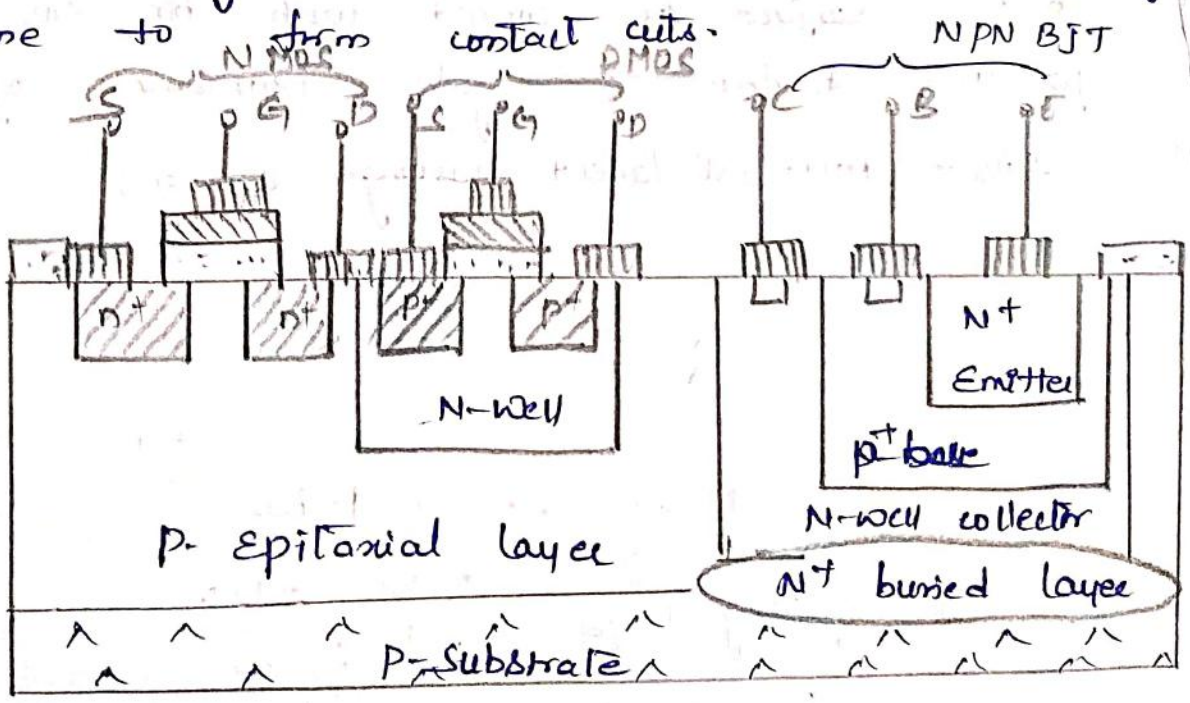
2) Emitter terminal of ~~bjt~~ BJT

3) N-well collector region for contact purpose

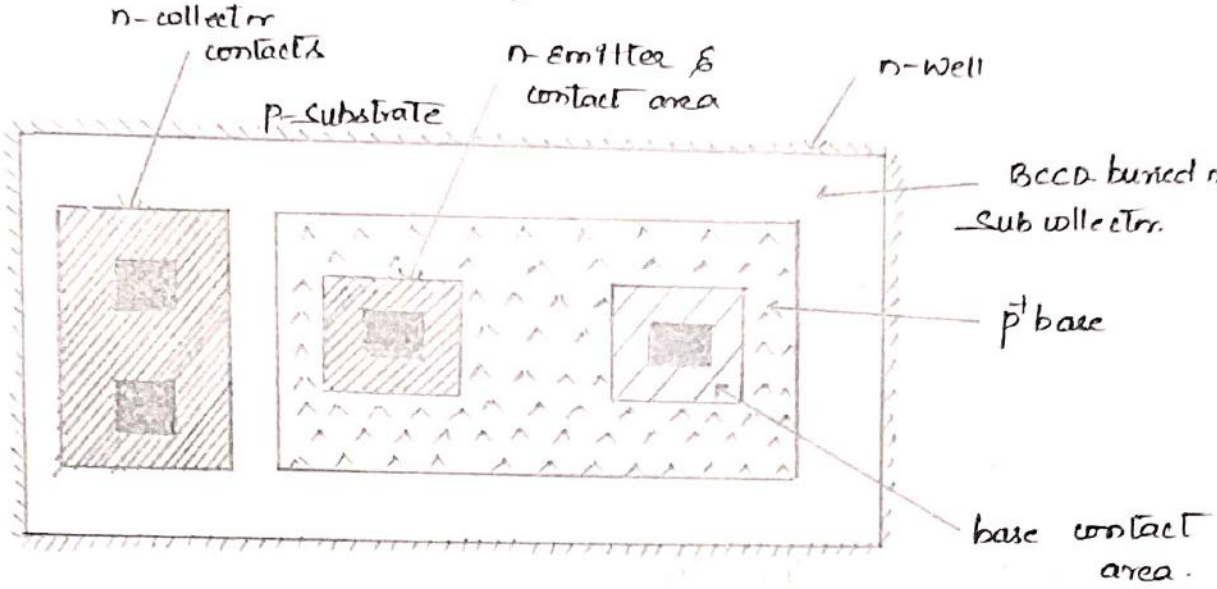
then entire surface is covered with thick SiO_2 layer to

p+ diffusion regions. (SiO_2 layer + photoresist + Mask + etching + Meta

) is done to form contact cuts.



The layout view of BiCMOS Transistor is shown in fig below. The fabrication process of BiCMOS is similar to CMOS but certain additional process steps and additional masks are considered. They are (i) p+ base region (ii) n+ collector area & (iii) buried sub collector (BCCD)



Silicon on Insulator (SOI) process:

Silicon on Insulator refers to the use of layered silicon Insulator in silicon substrate in place of conventional use of SI Substrate in semiconductor manufacturing specially Micro electronics to reduce parasitic capacitance in devices thereby Improving the performance.

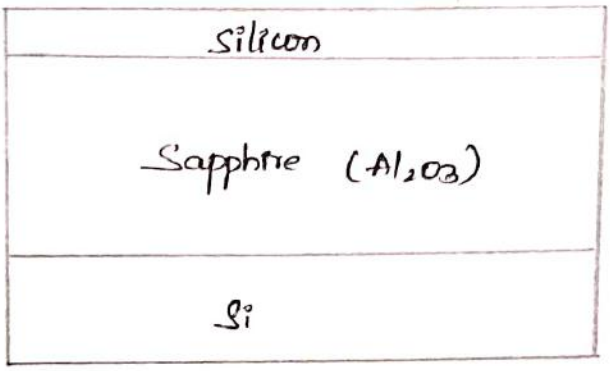
* The choice of Insulator depends on application they are two types of Insulators.
  i) Al_2O_3 (Sapphire) ii) SiO_2

1) Silicon on Sapphire Insulator (or) (SOS):

In this technology a thin layer of silicon is formed on the Sapphire surface. The thin layer of silicon is selectively doped to define different threshold voltages. Transistors. A Gate oxide is grown on top of this layer. above this

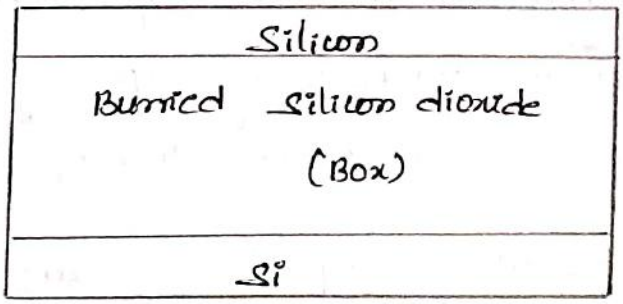
gate oxide it is covered with polysilicon gates. The p-MOS and N-MOS Transistors are formed by Ion Implantation.

* Al_2O_3 layer (or) Sapphire layer is used for high performance of devices or radio sensitive applications.



By using SiO_2 Insulator:

A silicon substrate is used and a burried oxide is grown on top of silicon substrate.



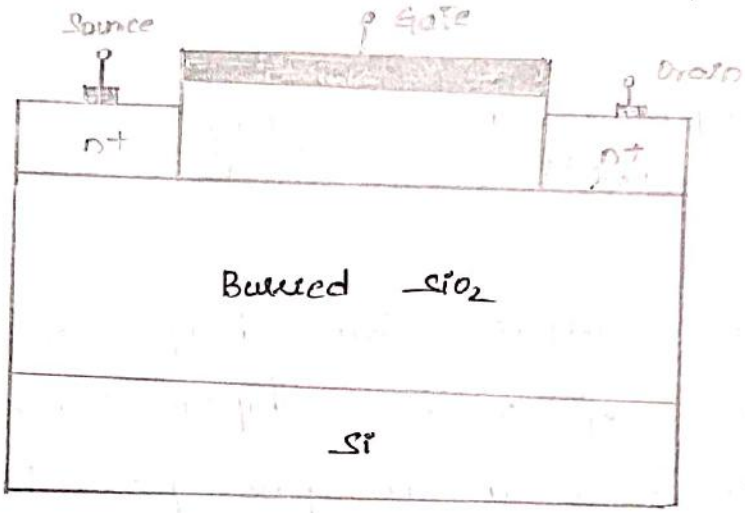
* SiO_2 layer is used for reducing the short channel effects in micro electronic devices.

There are two types of SOI process which are

1) fully depleted SOI
2) partially depleted SOI

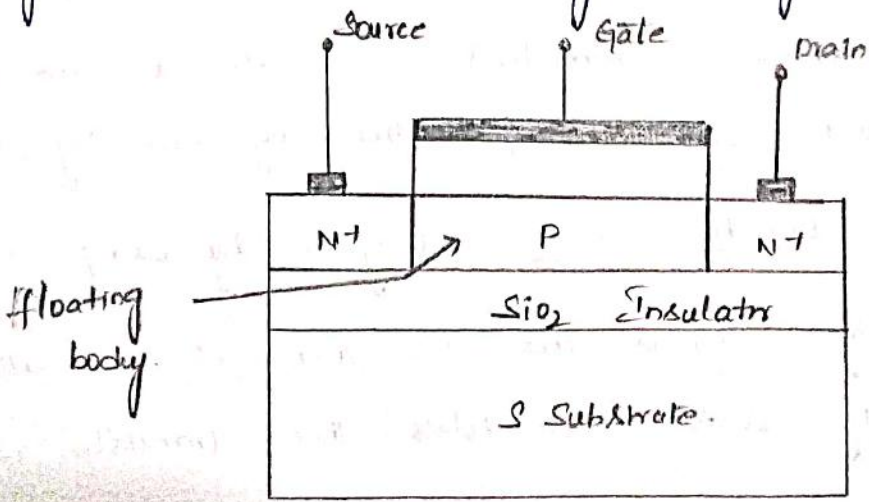
1> Fully depleted SOI :

In fully depleted SOI the body is thinner than the channel depletion width so the body charge is fixed & body voltage doesn't change. It is a process which is very difficult to manufacture because of thin body. A depletion region is empty (or) free of carriers in the body behind the gate (or) below the gate from.



2> Partially depleted SOI :

In partially depleted SOI the body is thicker than the channel depletion width, the voltage can vary depends on how much charge is present. this varying body voltage in changes the threshold voltage voltage (V_t).



Advantages :

* low parasitic capacitances due to isolation from the bulk silicon which lowers dynamic power consumption
* latch up due to the complete isolation of n & p well substrates.
* the threshold variations are smaller.
* Higher performance of equivalent Vpp it can also works at low Vpp.
* reduce temperature due to dopping.

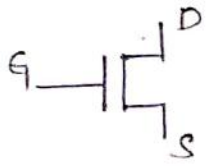
disadvantages:

* The PDSOI suffers from body effects.
* The presence of parasitic bipolar Transistor should be in off. It is known as pass gate leakage.
* Self heating (a best thermal insulating layer).

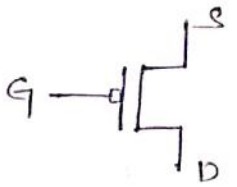
Mos Transistor switches (or) Mos Transistor as a switch:

All the Microprocessors (or) chips are build up of millions & Trillions of Transistors. & it is called as MOSFET. (Metal oxide Semiconductor field effect Transistor). Mos acts as switch (Digital switch) has to be operated by either logic 1 (or) logic '0'.

* the digital switch can be designed by using Mos Transistor.
* If the Majority carriers are e⁻s then it is called NMos
* If the Majority carriers are hole's then Transistor is called pMos



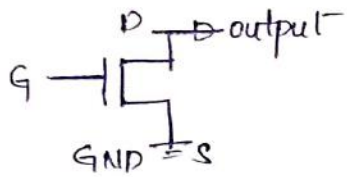
(a) N-MOS Symbol



(b) P-MOS Symbol.

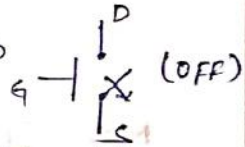
* the Digital Inverter (or) Not gate. are used to obtain same operation of digital switch. using MOS Transistors.

* NMOS (Buffer ckt):

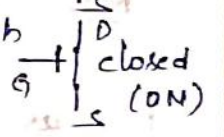


i/p(G)	o/p
0	D
1	$\overline{D}$

i) $V_G = 0$ (or) $V_{GS} = 0V$ / logic '0' then NMOS acts as open switch

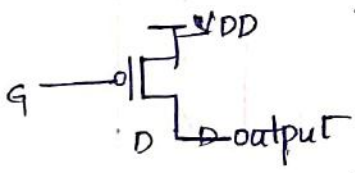


ii) If $V_G = 1$ (or) V_{GS} = logic '1' (or) high then NMOS acts closed switch

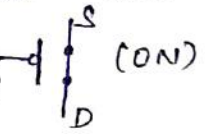


* NMOS will gives strong '0' & weak '1'.

PMOS (Inverter) ckt:

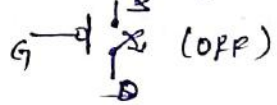


* when V_G (or) V_{GS} = 0V / logic '0' / low then PMOS acts as closed switch



i/p(G)	o/p
0	1
1	0

* when V_G (or) V_{GS} = G / logic '1' / high, then PMOS acts as open switch



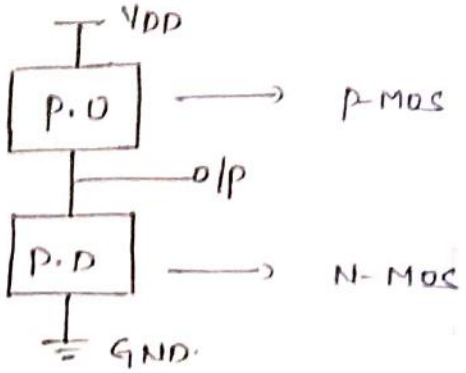
* PMOS will gives strong '1' & weak '0'.

* CMOS is a combination of PMOS & NMOS in a complementary way & it has good logic level, that has strong '0' & strong '1'.

* In CMOS the ckt is divided in to two parts.

1) pull up n/w → it is constructed using P-MOS only.

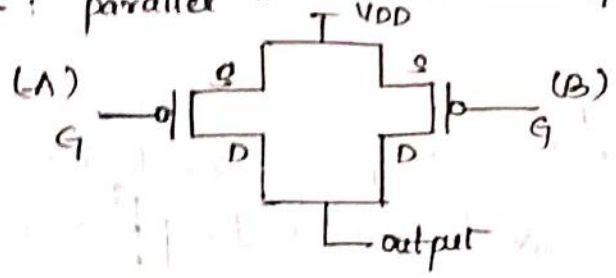
2) pull down n/w → it is constructed using N-MOS only



CMOS

MOS Transistors connected in parallel

1) PMOS : parallel connection (NAND Gate):

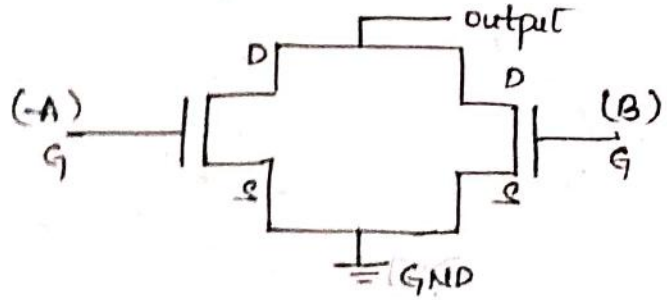


A	B	output
0	0	1
0	1	1
1	0	1
1	1	0

(Nand gate)

* If PMOS Transistors are connected in parallel then the ckt is called NAND Gate $[output = \overline{A.B}]$

2) N-MOS Transistors connected in parallel. (NOR Gate):



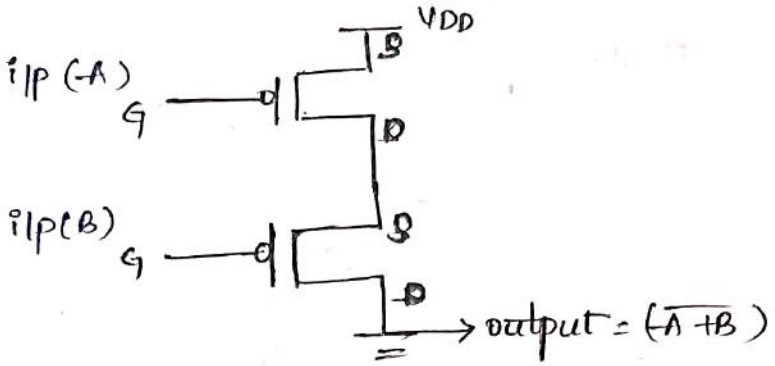
A	B	o/p
0	0	1
0	1	0
1	0	0
1	1	0

Partially o/p is at high logic level

* If All PMOS (or) 'n' number of NMOS Transistors are in parallel then the ckt is called NOR gate [output = $\overline{A+B}$]

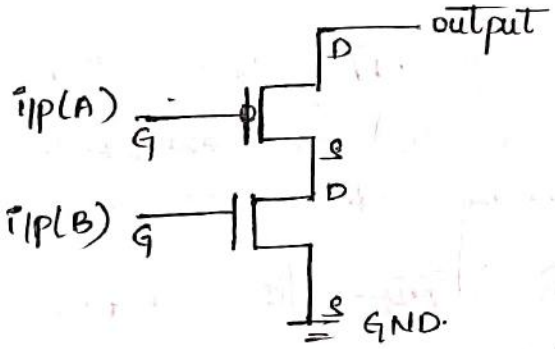
MOS Transistors connected in series:

1) PMOS Transistors connected in series (NOR Gate):



A	B	output
0	0	1
0	1	0
1	0	0
1	1	0

2) NMOS Transistors connected in series (NAND Gate):



A	B	output
0	0	1
0	1	1
1	0	1
1	1	0

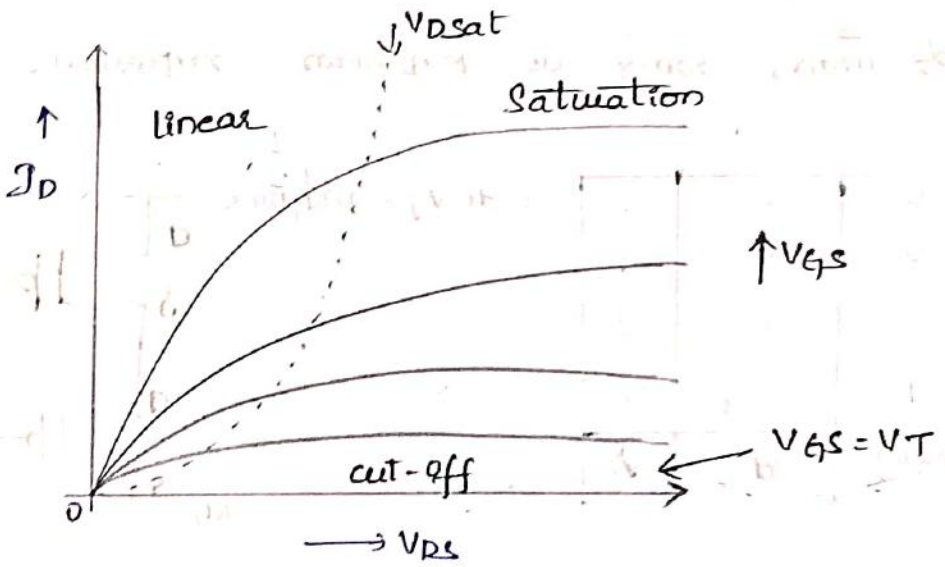
* If NMOS Transistors are connected in series then the ckt is NAND Gate.

Working of polar Transistor Resistors :

* when the $V_{GS} < V_{th}$ then it is in cut-off mode, it means that MOSFET is acting as a open load with Infinite resistance.

* when you slowly increase the gate voltage, the MOSFET slowly starts conducting by entering the linear region, when it starts developing voltage across it which we called as V_{DS}. In this region, the MOSFET acts as resistance of finite value

* Now when the MOSFET enters the saturation region, the resistance of the MOSFET is the least & is equal to the R_{on} resistor ($R_{DS\,on}$) of the MOSFET

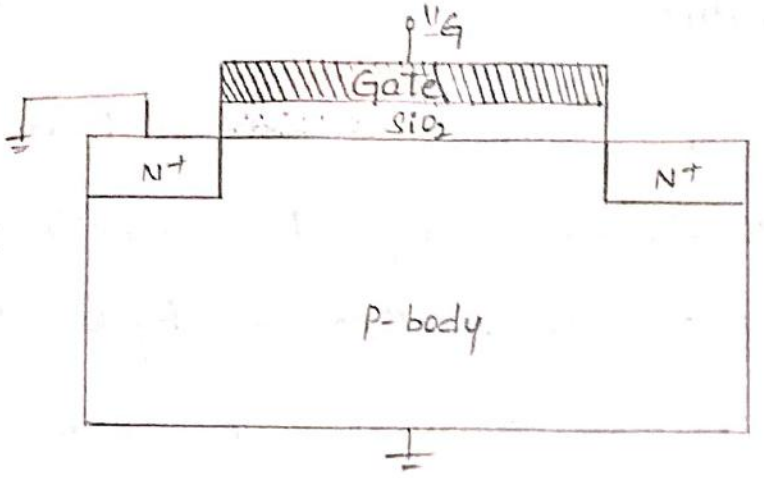


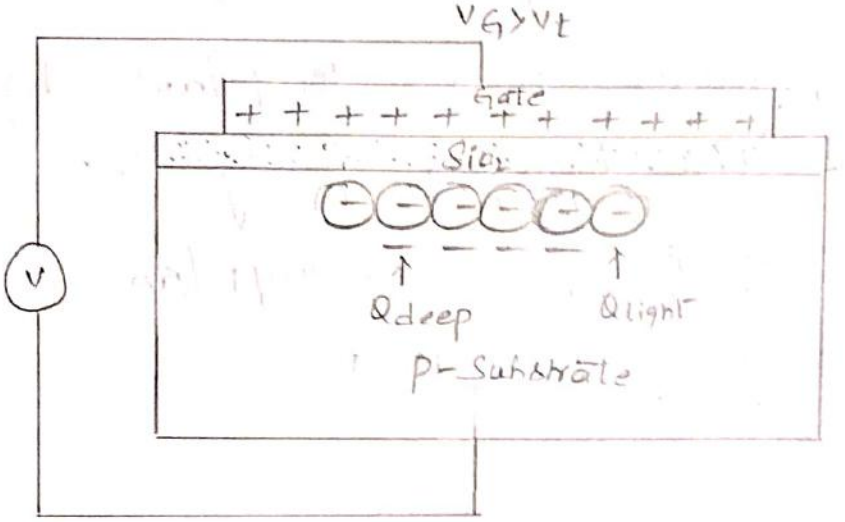
cut off
$V_{GS} < V_t$
$V_{DS} = 0$

$V_{ds} > V_{gs} - V_t$

Working polar transistor capacitors

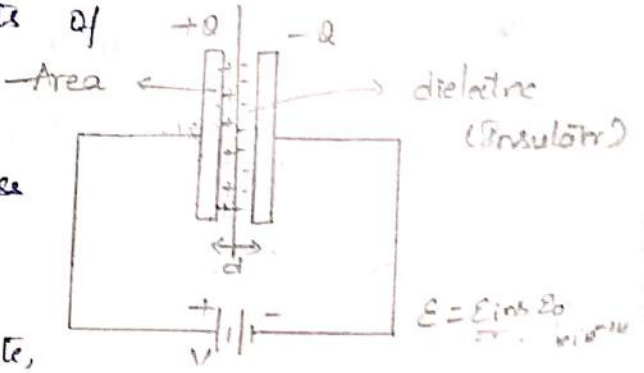
* An Mos capacitor is made of a semiconductor body or substrate, an insulator film, such as SiO_2 & a metal electrode called a gate.





* The structure looks like a parallel plate capacitor where one of the plates is the semiconductor & the other is the gate metalization. the insulator b/n the parallel plates is the oxide.

* the parallel plate capacitor consists of two parallel metallic plates of area A separated by an insulator of thickness d & dielectric constant.



* If we place a charge Q on upper plate, it attracts charges of opposite sign in the bottom plate. while repelling charges of same sign. If the bottom plate is connected to ground, the repelled charges flows to ground. so the two capacitor plates hold equal & opposite charges.

$C = \varepsilon A/d$

2) Junction capacitors:

- * It is formed when p-n junction diode is reverse biased
- * This can be formed using base - emitter & base collector junctions of an npn structure in bipolar ICs. This junction must be Maintained in reverse bias to provide desired capacitance i.e capacitance depends on reverse voltage
- * base emitter junction provides 1000pf/mm² high capacitance per unit with low breakdown voltage (~5V)
- * base - collector junction provides ~ 100pf/mm² with higher breakdown voltage (~40V).

Diffused resistors :

- * It is formed during emitter or base diffusion of npn bipolar process.
- * p⁺ base diffusion will provide moderate sheet resistivity of 100-200 Ω/m has resistance of 50-10kΩ range.
- * n⁺ emitter diffusion will provide low sheet resistivity of 2-10Ω/m has resistance of 1-100Ω range.

Basic electrical properties of MOS and BiCMOS circuits:

relation-ship between I_{ds} Vs V_{ds} of MOSFET:

* The idea of MOS Transistor develops from the use of voltage on the gate to induce channel between source and drain.

* The channel induced causes, the charge to flow from source to drain under the influence of an electric field created by voltage V_{ds} applied between drain & source.

* The charge Induced in to the channel is dependent on the gate to source voltage V_{GS}.

Hence I_{ds} is dependent on both V_{GS} & V_{DS}.

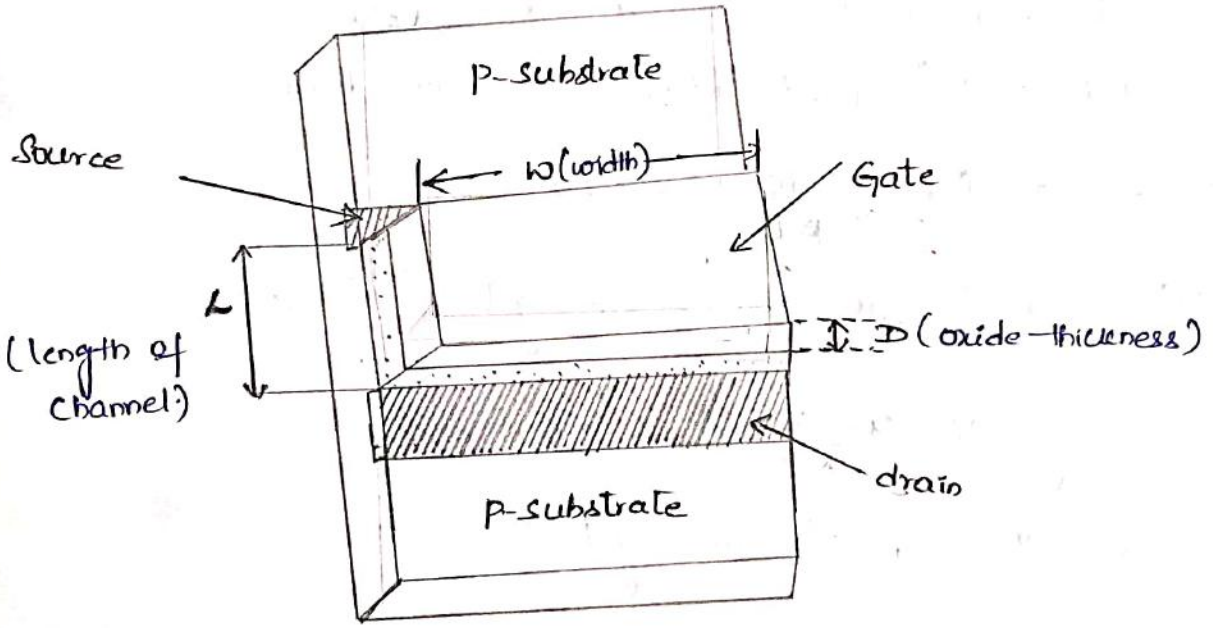


fig: N-MOS Transistor Structure:

* In this N-MOS structure, electrons drift from source to drain. at a rate proportional to the electric field b/n these regions. thus by knowing the amount of charge in

the channel and the rate at which it moves the currents can be computed.

thus

$$I_{ds} = -I_{sd} \left[\text{Charge Induced in channel } (Q_c) \right] \rightarrow (1)$$

$$I_{ds} = \frac{\text{charge Induced in the channel } (Q_c)}{\text{Electron transit time } (T)}$$

Transit time (T):

$$T_{sd} = \frac{\text{length of channel } (L)}{\text{velocity } (V)} \rightarrow (2)$$

where

velocity $V = \mu E_{ds} \rightarrow (3)$

where

μ = mobility of electron (or) hole (surface)

E_{ds} = Electric field from source to drain.

$$\text{Electric field } E_{ds} = \frac{V_{ds}}{L} \rightarrow (4)$$

Substitute eq (4) in eq (3) we get.

$$\text{velocity } V = \mu \cdot \frac{V_{ds}}{L} \rightarrow (5)$$

Sub eq (5) in eq (2) we get

$$T_{sd} = \frac{L}{\mu \cdot \frac{V_{ds}}{L}} =$$

$$\boxed{T_{sd} = \frac{L^2}{\mu \cdot V_{ds}}} \rightarrow (6)$$

Non Saturated region:

* charge Induced in channel is owed to the diffaence in voltage among gate and channel. Vgs because of the IR drop In the channel. the voltage along the length of the channel vaues. lineaily with distance.

* In Non saturated region of device, the average value along the channel is $\frac{V_{ds}}{2}$ & the Effective gate voltage.

$$V_{ds} = V_{gs} - V_t = V_{eff}$$

where V_t = threshold voltage.

The charge per unit area. = $E_g \varepsilon_{ins} \varepsilon_0$

thus induced charge $Q_c = E_g \varepsilon_{ins} \varepsilon_0 WL \rightarrow (7)$

E_g = Average eletric field from gate to channel.

ε_{ins} = Relative permitivity of Insulation b/n gate & channel.

= 4.0 for SiO_2

ε_0 = permitivity of free space.

= 8.85×10^{-14} cm^{-1}

Now

$$E_g = \frac{\left[(V_{gs}-V_t) - \frac{V_{ds}}{2}\right]}{D} \rightarrow (8)$$

where D = oxide thickness.

Substitute Eq (8) in Eq (7) we get.

$$Q_c = \frac{\varepsilon_{in} \varepsilon_0 WL}{D} \left[(V_{gs}-V_t) - \frac{V_{ds}}{2}\right] \rightarrow (9)$$

Substitute Eq (9) & Eq (6) in I_{ds}.

$$I_{ds} = \frac{\varepsilon_{ins}\varepsilon_0 W}{D}\left[(V_{gs}-V_t) - \frac{V_{ds}}{2}\right]$$

$$\frac{L}{\mu V_{ds}}$$

$$I_{ds} = \frac{\varepsilon_{ins}\varepsilon_0}{D}\mu\frac{W}{L}\left[V_{ds}(V_{gs}-V_t) - \frac{V_{ds}^2}{2}\right]$$

$$I_{ds} = K\cdot\frac{W}{L}\left[(V_{gs}-V_t)V_{ds} - \frac{V_{ds}^2}{2}\right] \longrightarrow (10)$$

where $K = \frac{\varepsilon_{ins}\varepsilon_0}{D}\mu$.

Non Saturated region $V_{ds} < V_{gs}-V_t$.

$$\beta = K\cdot\frac{W}{L}$$

$$I_{ds} = \beta\left[(V_{gs}-V_t)V_{ds} - \frac{V_{ds}^2}{2}\right] \longrightarrow (11)$$

* It is possible to model the gate as parallel plate capacitor with capacitance proportional area over thickness. If the length and width of the gate are L and W with an oxide thickness 'D'. the gate to channel capacitance is

$$C_g = \frac{\varepsilon_{ins}\varepsilon_0 WL}{D} \longrightarrow (12)$$

we also have.

$$K = \frac{\varepsilon_{ins}\varepsilon_0\mu}{D}$$

$$C_g = \frac{K\cdot WL}{\mu}$$

$$K = C_g = K = \frac{C_g\mu}{WL}$$

Substitute above value 'K' in eq (10)

$$I_{ds} = \frac{C_g \mu}{WL} \cdot \frac{W}{2} \left[(V_{gs} - V_t) V_{ds} - \frac{V_{ds}^2}{2} \right]$$

$$I_{ds} = \frac{C_g \mu}{2L} \left[(V_{gs} - V_t) V_{ds} - \frac{V_{ds}^2}{2} \right] \rightarrow (13)$$

Sometimes, it is convinient to use gate capacitance per unit area then C_g, it is given as

$$C_g = C_0 W L$$

Now Eq (13) can be written as

$$I_{ds} = \frac{C_0 W L \mu}{2L} \left[(V_{gs} - V_t) V_{ds} - \frac{V_{ds}^2}{2} \right]$$

$$\boxed{I_{ds} = \frac{C_0 W \mu}{L} \left[(V_{gs} - V_t) V_{ds} - \frac{V_{ds}^2}{2} \right]}$$ (non saturation region)

Saturated region:

For $V_{ds} > V_{d(sat)} = V_{gs} - V_t$

[Substitute $V_{ds} = V_{d(sat)}$ in eq (10)

$$I_{ds} = K \cdot \frac{W}{L} \left[(V_{gs} - V_t) V_{d(sat)} - \frac{V_{d(sat)}^2}{2} \right]]$$

$$I_{ds} = K \cdot \frac{W}{L} \left[(V_{gs} - V_t)(V_{gs} - V_t) - \frac{(V_{gs} - V_t)^2}{2} \right]$$

$$= K \cdot \frac{W}{L} \left[\frac{2(V_{gs} - V_t)^2 - (V_{gs} - V_t)^2}{2} \right]$$

$$I_{ds} = k \cdot \frac{W}{L} \left[\frac{(V_{gs} - V_t)^2}{2} \right]$$

$$I_{ds} = \frac{\beta}{2} (V_{gs} - V_t)^2$$

$$I_{ds} = \frac{C_g \mu}{2L^2} (V_{gs} - V_t)^2 \quad \text{for gate capacitance.}$$

$$C_g = C_o \cdot WL$$

$$I_{ds} = \frac{C_o \cdot WL \mu}{2L^2} (V_{gs} - V_t)^2$$

$$\boxed{I_{ds} = \frac{C_o \cdot W \mu}{2L} (V_{gs} - V_t)^2} \quad \text{(Saturation region)}$$

Threshold voltage:

* The Gate voltage (V_{gs}) which is required to produce inversion charge layer is known as threshold voltage. It is denoted by V_t ($V_{gs} > V_t$)

V_t can be achieved by following aspects.

i) Gate voltage is needed.

ii) Surface potential $\phi_s = 2E_{fp}$ for p-type semiconductor and $\phi_s = 2E_{fn}$ for n-type semiconductor.

* if $V_{GS} < V_t$, then MOSFET is in cut off region.

* To operate the MOSFET in linear and saturation region

when $V_{gs} > V_t$.

Parameters of threshold voltage in MOSFET :

there are 4 physical parameters which effects the threshold voltage of MOS structure.

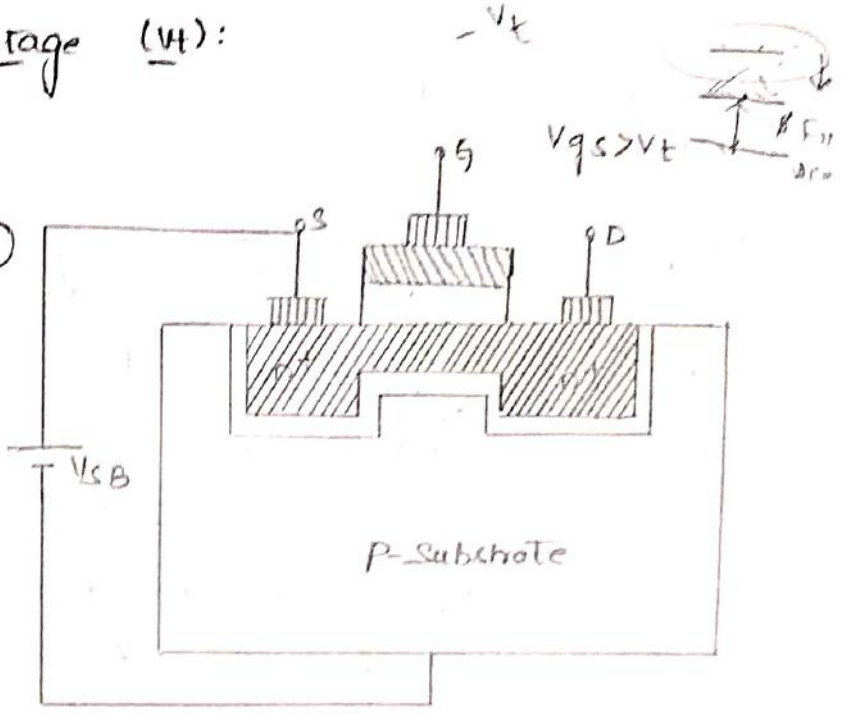
* work function difference between gate and channel. there is a different in gate & channel resulting in generation of V_t.

* The gate voltage component to change surface potential. i.e. fermi potential of substrate surface should changes from ϕ_f to $-\phi_f$ for generation of inversion layer.

* The gate voltage component to offset the depletion region charge. ie when applied gate voltage, the holes of p-substrate will move from surface to deeper inside of material. when it moves holes from it generates ion terms ions. & because of ions there will be generation of voltage called offset voltage.

* The voltage component to offset the fixed charges in the gate oxide & in the silicon oxide interface. ie when in the fabrication of MOSFET at that time due to imperfection in lattice structure there is some fixed charges that is generating some potential.

Derivation of threshold voltage (V_t):



$$V_t = \phi_{ms} \frac{Q_B - Q_{ss}}{C_0} + 2\phi_{FN} \rightarrow (1)$$

Q_B = charge per unit area in depletion layer

Q_{ss} = charge density of $Si : SiO_2$ (substrate to oxide layer)

C_0 = capacitance per unit gate area.

ϕ_{ms} = work function is the difference b/n gate and channel.

$\phi_{ms} = \phi_F$ (substrate) $- \phi_M$ (metal)

$\phi_{ms} = \phi_F$ (substrate) $- \phi_F$ (Gate polysilicon) is -ve & negligible.

to balance the -ve term we can obtain V_t

$$\text{Negative term} = -\frac{Q_{ss}}{C_0}$$

ϕ_{FN} = fermi level potential difference b/n Inverse surface and bulk si

$$Q_B = \sqrt{2\varepsilon_0 \varepsilon_{si} q_N (2\phi_{FN} + V_{SB})} \quad \text{coulombs/m}^2$$

V_{SB} = substrate bias voltage, it is -ve for N-MOS & +ve for P-MOS.

ε_{Si} = silicon permitivity = 11.7. $\quad \varepsilon_0 = 8.854 \times 10^{-12}$

$$\phi_{FN} = \frac{KT}{q} \ln \frac{N}{n_i} \text{ volts}$$

$Q_{ss} = (1.5 \text{ to } 8) \times 10^{-8}$ columbs/m^2

where N = Impurity concentrate on substrate

n_i = Instrinsic electron concentration $= 1.6 \times 10^{10}/cm^3$ at $300^\circ k$

K = Boltzmann constant = 1.4×10^{-23} Joules/°K

q = charge of electron = 1.6×10^{-19} coulomb

Problems on threshold voltage:

Consider a MOS s/m with following parameters:

thickness of oxide tox = 200 A°

ϕ_{ms} (or) ϕ_{GC} = +0.85V

$Q_{OX} = q \times 2 \times 10^{11}$ c/cm²

$N_A = 2 \times 10^{15}$ cm^{-3}

1) determine the threshold voltage V_{TO} under zero bias at room temp (300K). Note that ε_{ox} (or) $\varepsilon_o = 3.97\varepsilon_o$ $\varepsilon_{si} = 11.7\varepsilon_o$

Sol: $$V_{TO} = \phi_{ms} - 2\phi_{FN} - \frac{Q_{BO}}{C_{OX}} - \frac{Q_{OX}}{C_{OX}}$$

Fermi potential of p-type substrate

$$\phi_F = \frac{KT}{q} \ln\left(\frac{N_A}{N_A}\right) = \frac{1.38 \times 10^{-23} \times 300}{1.6 \times 10^{-19}} \ln\left(\frac{2 \times 10^{15}}{1.6 \times 10^{10}}\right)\left(\frac{1.6 \times 10^{10}}{2 \times 10^{15}}\right)$$

$$= 0.025875 \ln(7.25 \times 10^6)$$

$$= 0.025875 \times 11.736069$$

$$= -0.30367 \text{ Volts.}$$

charge density Q_{BO}

$$Q_{BO} = -\sqrt{2qN_A\varepsilon_o\varepsilon_{si}(2\phi_{FN})} = -\sqrt{2 \times 1.6 \times 10^{-19} \times 2 \times 10^{15} \times 11.7 \times 8.854 \times 10^{-14} \times 2 \times 0.30367}$$

$$= -2.0143 \times 10^{-8} \text{ c/cm}^2$$

$$C_{OX} = \frac{\varepsilon_{ox}}{t_{ox}} = \frac{3.97 \times \varepsilon_o}{200 \times 10^{-8}} = \frac{3.97 \times 8.854 \times 10^{-14}}{200 \times 10^{-8}}$$

$$= 1.757 \times 10^{-7} \text{ F/cm}^2$$

$$V_{TO} = 0.85 + 2 \times 0.30367 + \frac{2.0143 \times 10^{-8}}{1.757 \times 10^{-7}} - \frac{1.6 \times 10^{-9} \times 2 \times 10^{11}}{1.757 \times 10^{-7}}$$

$= -0.85 + 0.607 + 0.114 + 0.182$

$V_{TO} = -0.306$ volts.

2) Determine the type (p-type or n-type) and amount of channel Implant (Ni/cm^2) required to change the threshold voltage. to 0.8V.

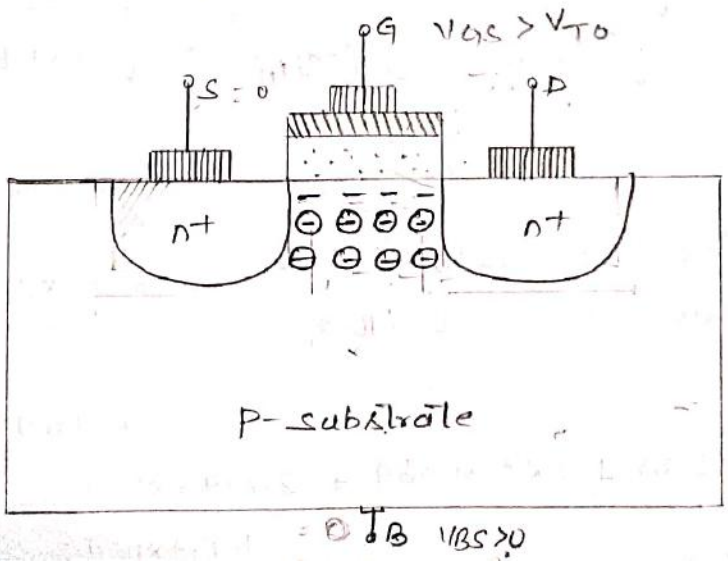
Sol: In this substrate we add Impurities of N_A (p-type) which we need to Implant.

$$\Delta V = 0.8 - V_{TO} = \frac{qNi}{Cox} = 0.8 + 0.306 = 1.106$$

$$Ni = \frac{\Delta V \times Cox}{q} = \frac{1.106 \times 1.757 \times 10^{-7}}{1.6 \times 10^{-19}} = 1.21 \times 10^{12}/cm^2$$

Body effect:

Generally MOS transistor is treated as three terminal device. But the body of the transistor is also an implicit terminal which helps to understand the characteristics of the transistor. considering the body of the MOS transistor as a terminal is known as the body effect. the potential difference between the source & body (V_{SB}) affects the threshold voltage of transistor.



$V_{SB} = 0$ V_{TH} ↑

$V_{SB} > 0$ V_{TH} ↓

$$V_{TH} = \psi_s + \frac{1}{C_{ox}} \sqrt{2\varepsilon_{si} q N_A |\psi_s|}$$

for NMOS transistor the surface potential is +ve. & PMOS surface potential -ve. so the $|\psi_s|$ is defined.

* Basically the body and source potential are zero initially. $V_S = 0$, $V_B = 0$ and it is not always connected to source.

* Assume V_{GS} = +ve voltage so -ve immobile charges are created across the substrate & have some free electrons when $V_B = 0$ V or grounded then all the work that is done to invert the channel is done by gate is called threshold voltage.

* when $V_{BS} > 0$ i.e some +ve potential is applied to V_{BS} then capacitn. across the channel. is acts like a back gate capacitn. just like capacitn btn gate & the channel. so that -ve charges are generated. so V_{TH} is decreased. [illegible].

* [illegible].

$$V_{TH} = V_{TH0} + \gamma \left(\sqrt{|\psi_s + V_{SB}|} - \sqrt{|\psi_s|} \right)$$

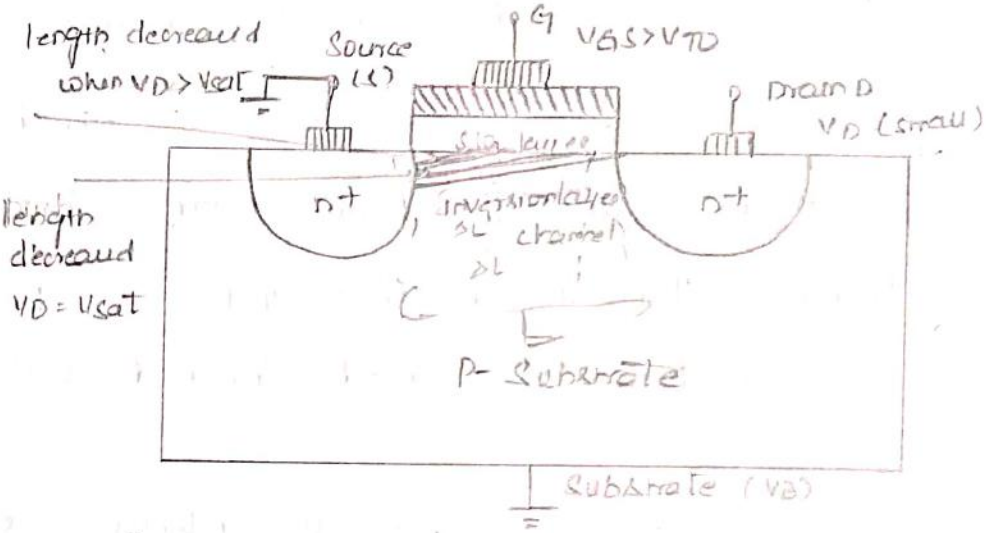
V_{TH0} = Threshold voltage at $V_{SB} = 0$ V.

γ = Empirical body effect coefficient > 0 for NMOS $\sqrt{V}$.

$V_{GS} > V_T$
$V_{DS} > V_{GS} - V_T$

channel length modulation:

* The sudden changes in channel length. is called channel length modulation. when MOSFET operates in saturation region.
* the length of channel changes w.r.t V_{DS}. If V_{DS} is increases in saturation region then the channel length decreases.
* when $V_{DS} > V_{sat}$ then the channel length decreases gradually.



* ~~Total charge density~~ is ~~decreased~~ by

I_D (or) $I_{DS} = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{TH})^2$ (long channel equation ≠ fn (VDS))

when $V_{DS} > V_{sat}$ (or) V_{DS} increases length of channel decreases to

$$L \rightarrow (L - \Delta L)$$

$$I_D \text{ (or) } I_{DS} = \frac{1}{2} \mu_n C_{ox} \frac{W}{(L - \Delta L)} (V_{GS} - V_{TH})^2$$

It can be written as

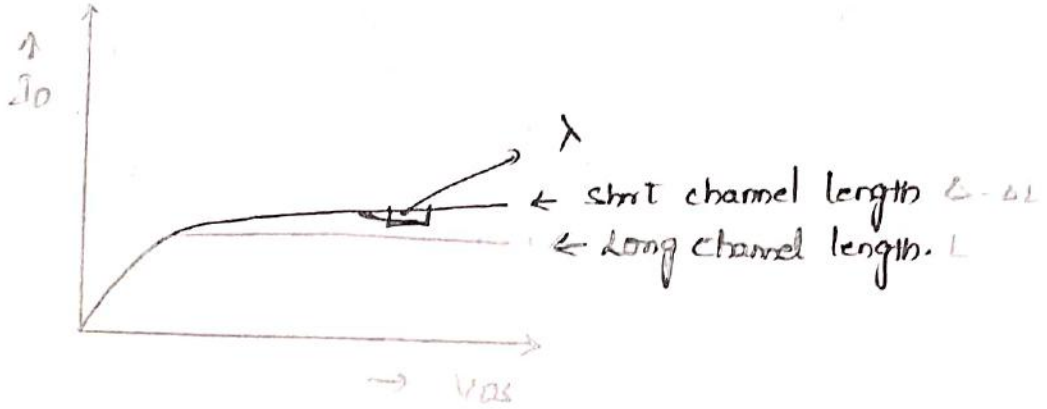
$$I_D = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{TH})^2 \left(1 + \frac{\Delta L}{L}\right)$$

~~$\frac{\Delta L}{E}$ is called~~ $\frac{\Delta L}{L} = \lambda V_{DS}$.

λ = channel length modulation parameter

$$\boxed{I_D = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{TH})^2 (1 + \lambda V_{DS})}$$

I_D Vs V_{DS} Graph for channel length modulation



MOS Transistor Transconductance (g_m):

The Transconductance will gives relationship between output current (I_{ds}) to input voltage (V_{gs})

$$g_m = \frac{\delta I_{ds}}{\delta V_{gs}}\Big|_{V_{ds} = \text{constant}} \rightarrow (1)$$

To find the expression for transconductance g_m in terms of Circuit and transistor parameter consider that the charge in channel Q_c is

$$I_{ds} = \frac{Q_c}{\tau_{sd}}$$

where τ_{sd} = electron transit time b/n source & drain

Now the change in the output current is

$$\delta I_{ds} = \frac{\delta Q_c}{\tau_{sd}} \rightarrow (2)$$

$$\text{Now } \tau_{sd} = \frac{L^2}{\mu V_{ds}} \rightarrow (3)$$

Substitute τ_{sd} (3) in Eq (2) we get

$$\delta I_{ds} = \frac{\delta Q_c}{\frac{L^2}{\mu V_{ds}}}$$

$$\delta I_{ds} = \frac{\delta Q_c \mu V_{ds}}{L^2} \rightarrow (4)$$

But change in the charge $\delta Q_c = C_g \delta V_{gs} \rightarrow (5)$

Substitute Eq (5) in Eq (4) we get

$$\delta I_{ds} = \frac{\mu V_{ds} C_g \delta V_{gs}}{L^2}$$

$$\frac{\delta I_{ds}}{\delta V_{gs}} = \frac{\mu V_{ds} C_g}{L^2} \quad \Big| \because g_m = \frac{\delta I_{ds}}{\delta V_{gs}}$$

$$\boxed{g_m = \frac{\mu V_{ds} C_g}{L^2}} \rightarrow (6)$$

At saturation region, $V_{ds} = V_{gs} - V_t \rightarrow (7)$

Substitute Eq (7) in Eq (6)

$$\boxed{g_m = \frac{\mu C_g (V_{gs} - V_t)}{L^2}} \rightarrow (8)$$

W.K.T $C_g = \frac{\varepsilon_0 \varepsilon_{ins} WL}{D} \rightarrow (9)$

Substitute eq (9) in Eq (8) we get.

$$g_m = \frac{\mu \varepsilon_0 \varepsilon_{ins} WL}{D L^2} (V_{gs} - V_t)$$

$$g_m = \frac{\mu \varepsilon_0 \varepsilon_{ins}}{D} \frac{W}{L} (V_{gs} - V_t)$$

$$g_m = K \cdot \frac{W}{L} (V_{gs} - V_t)$$

$$\boxed{g_m = \beta (V_{gs} - V_t)}$$

* It is possible to increase the transconductance g_m of MOS Transistor by measuring the width, but in this process to increase the i/p capacitance as well as area occupied of the MOS. A reduction in channel length to increase transconductance (g_m) up to the short channel effect is involved when the increasing effective gate voltage to provide the better transconductance (g_m).

MOS transistor output conductance (g_{ds}):

the output conductance g_{ds} is expressed as

$$g_{ds} = \frac{\delta I_{ds}}{\delta v_{gs}} = \lambda \cdot I_{ds} \propto \left(\frac{1}{L^2}\right)$$

Here the strong dependence on the channel length is

$$\lambda \propto \left(\frac{1}{L}\right) \text{ and}$$

$$I_{ds} \propto \left(\frac{1}{L}\right) \text{ for MOS device.}$$

It is defined as ratio of change in output current to the change in input voltages.

figure of Merit (w_0) :-

The indication of the frequency response may be obtained from the parameter w_0 where.

$$w_0 = \frac{g_m}{c_g} \longrightarrow (1)$$

the figure of merit is defined as the ratio of transconductance to the gate capacitance.

W.K.T $g_m = \frac{\mu \varepsilon_0 \varepsilon_{ins}}{D} \frac{W}{L} (V_{gs} - V_t)$

$C_g = \frac{\varepsilon_0 \varepsilon_{ins} WL}{D}$

Substituting these values g_m & C_g in eqn (1) we get

$$W_0 = \frac{\frac{\mu \varepsilon_0 \varepsilon_{ins}}{D} \cdot \frac{W}{L} (V_{gs} - V_t)}{\frac{\varepsilon_0 \varepsilon_{ins} WL}{D}}$$

$$W_0 = \frac{\mu \varepsilon_0 \varepsilon_{ins}}{D} \cdot \frac{W}{L} (V_{gs} - V_t) \times \frac{D}{\varepsilon_0 \varepsilon_{ins} \cdot WL}$$

$$\boxed{W_0 = \frac{\mu}{L^2} (V_{gs} - V_t)} \rightarrow (2)$$

at saturation region $V_{ds} = V_{gs} - V_t \rightarrow (3)$

Substitute eqn (3) in eqn (2) we get

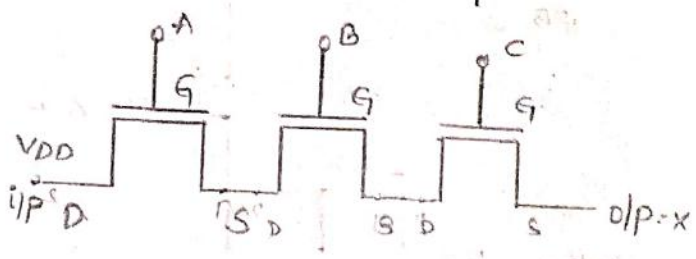
$$W_0 = \frac{\mu}{L^2} V_{ds}$$

W.K.T $\tau_{sd} = \frac{L^2}{\mu V_{ds}}$

$$\boxed{W_0 = \frac{1}{\tau_{sd}}}$$

Pass transistor:

* Pass transistor is used to reduce the power consumption in static CMOS logic. PTL which is used to reduce the no. of transistors required to implement logic gates.
* It is a series of transistor connected to denote a logic



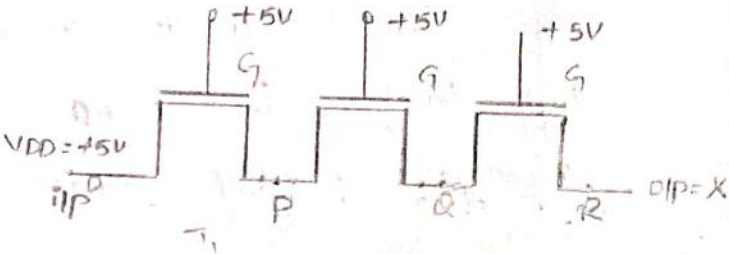
or $X = A \cdot B \cdot C$.

* when the transistor are connected to denote OR logic

~~$X = \bar{A} + \bar{B} + \bar{C}$~~ $X = A + B + C$.

Example:

1) For N-MOS pass transistor logic



i) $V_D > V_t$ Transistor is 'ON'

ii) $V_D < V_{gs} - V_t$; $V_S = V_D$

iii) $V_D > V_{gs} - V_t$; $V_S = V_{gs} - V_t$

$V_D = 5V$, $V_G = 5V$, $V_T = 1V$

For (P) output: i) $V_D > V_t$. so Transistor is ON

ii) $V_g - V_t = 4V = V_S$

$V_D = 5V$

$V_D > V_{gs} - V_t$, $V_S = V_g - V_t$

$5V > 4V$, $V_S = 5 - 1 = 4V = V_P$

For Q output: $V_D = 4V$, $V_G = 5V$

i) $V_D > V_t$; $4V > 1V$ so Transistor is ON.

ii) $V_G - V_t = 5 - 1 = 4V$.

$V_D = V_G - V_t = 4V$.

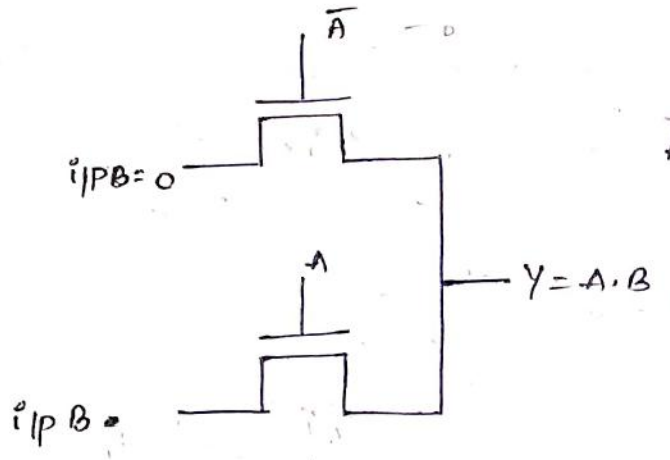
$V_Q = V_S = V_G - V_t = 4V$.

For R output: $V_G = 5V$. $V_D = 4V$ $V_T = 1V$

$V_Q = V_D = 4V$. , $V_t = 1V$.

$V_R = 4V = V_S = 5V - 1V$

2) AND logic using PTL

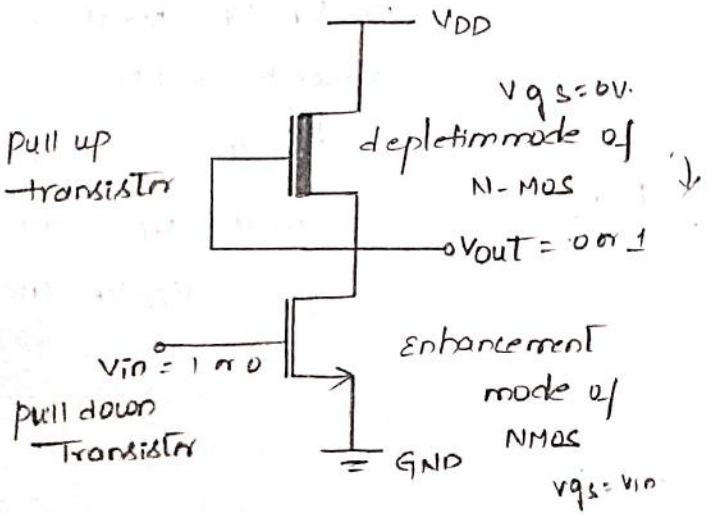


i/p

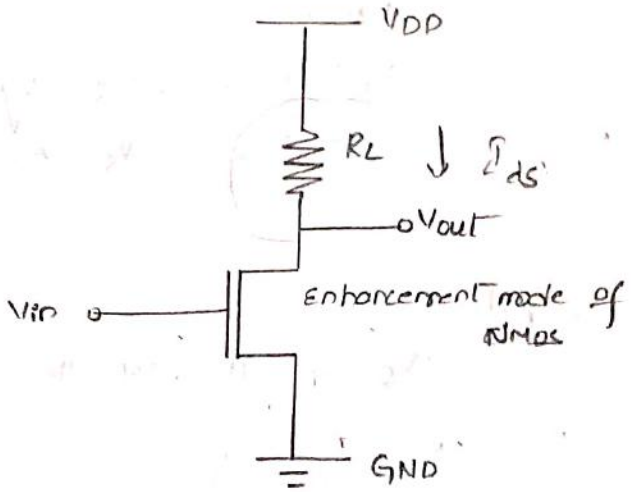
A	B	output
$\bar{A}=0$	0	0	0
	1	0
A=1	0	0
	1	1

NMOS Inverter:

* A Basic requirement for producing a complete range of logic circuit is the inverter. this is needed for restring logic level for NAND and NOR gates and for sequential as well as memory circuits. of various from the basic bias inverter ckt.
* The inverter is the basic element in digital ckt design. and to get output.



fig(b): Basic NMOS Inverter



fig(a): Resistive load of NMOS Inverter

* In fig: (a) shows the NMOS Inverter ckt with resistive load. the Enhancement Mode NMOS Source is connected to ground and the load is connected to chain and the drain is connected to +ve supply (+VDD) through some load resistor (R_L). the output is taken from chain & the i/p is applied b/n gate & ground.

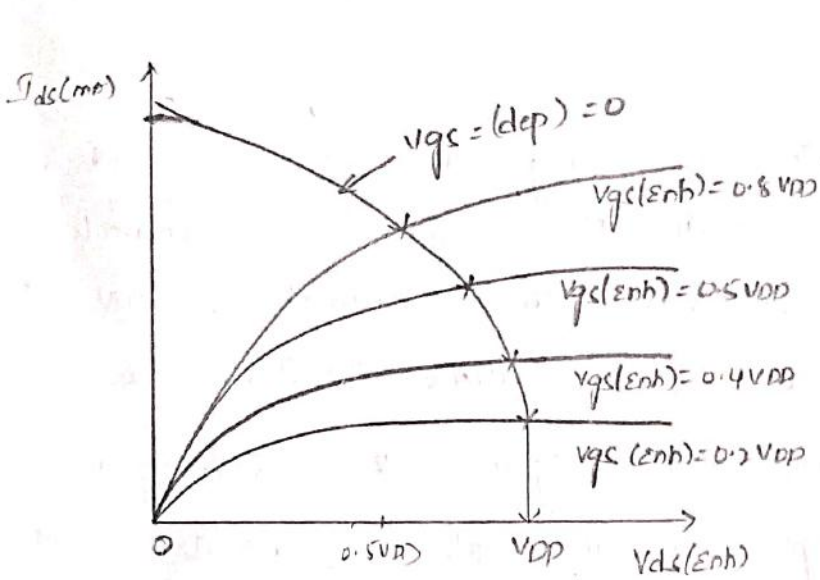
* In this the resistor is bulky in size instead of that we can use depletion mode NMOS which is connected to power supply VDD. so that the chanel is ON forever acts as a resistive Network in NMOS Inverter. as shown in figure (b). because in fig(a). the resistors are not convienently producing on the silicon substrate. even modest values occupies extensively large areas. In order to solve the problem. depletion mode transistor is used as load in fig(b).

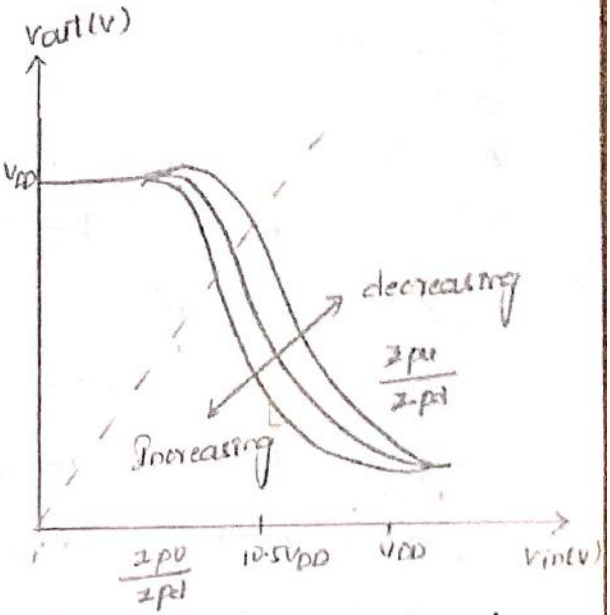
* In this depletim mode transistor $V_{gs}=0V=V_{in}$ & it acts as pull up transistor. and Enhancement mode transistor $V_{in}=V_{gs}>V_t$ & it acts as pull down transistor so that it is in ON state.

N-MOS Inverter Transfer characteristics:

The transfer characteristic is drawn by taking V_{ds} on x-axis & I_{ds} on y-axis for both Enhancement and depletion mode transistors. to obtain the Inverter transfer characteristics. for $V_{gs}=0V$ for depletion mode. and the graph shows that maximum voltage across the Enhancement mode device corresponds to minimum voltage across the depletim mode transistor.

* as $V_{in} = V_{gs} > V_t$ then current begins to flow. the output voltage V_{out} decreases & subsequently increase in V_{in} will cause pull down transistor to comeout of saturation region to resistive region.





NMOS Inverter transfer characteristics

$V_{ds}(Enh) = V_{DD} - V_{ds}(dep) = 0V$

$V_{gs}(enh) = V_{in}$

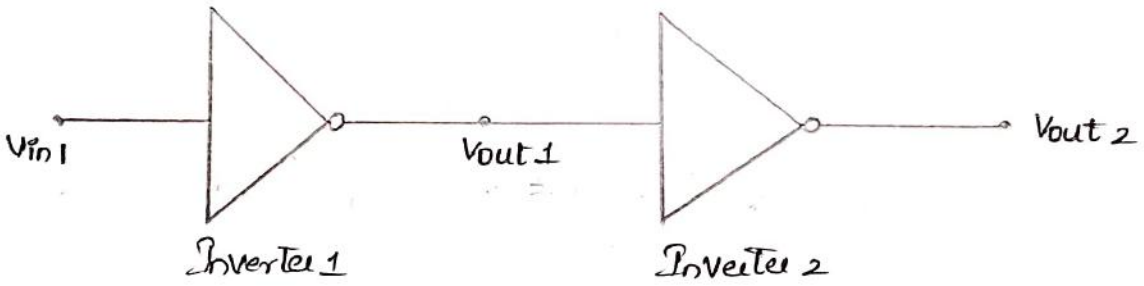
during the transition the slope of transfer characteristics determines the gain

$$Gain = \frac{\delta V_{out}}{\delta V_{in}}$$

the point at which $\boxed{V_{out} = V_{in} = 0.5VDD = V_{inv}}$ it is denoted as Inverter voltage.

Determination of pullup to pulldown ratio (Z_{pu}/Z_{pd}) for NMOS Inverter driven by another NMOS Inverter:

* consider the arrangement in which an inverter is driven from the output of another similar NMOS Inverter. consider the depletion mode transistor for which $V_{gs}=0V$ under the all conditions and further assumed that in order to cascade Inverters without degradation of levels we are aiming to meet the requirement $V_{in} = V_{out} = V_{inVerter}$ (Inverter voltage).



For equal margins around the Inverter threshold we set $V_{in} = 0.5V_{DD}$ at this point both transistors are in saturation region. then

$$I_{ds} = K \cdot \frac{W}{L} \frac{(V_{gs}-V_t)^2}{2}$$

In the depletion mode of pull up N-Mos transistor

$$I_{ds} = K \cdot \frac{W_{pu}}{L_{pu}} \frac{(0-V_t)^2}{2} \quad [\because V_{gs}=0V]$$

$$I_{ds} = K \cdot \frac{W_{pu}}{L_{pu}} \left[\frac{-V_{td}}{2}\right]^2$$

Enhancement mode of pulldown N-Mos Transistor

$$I_{ds} = K \cdot \frac{W_{pd}}{L_{pd}} \left(\frac{V_{inv} - V_t}{2} \right)^2$$

where W_{pd}, W_{pu}, L_{pu}, L_{pd} are the widths & the lengths of pullup & pull down transistors respectively. by equating above two equations because of the currents are equal.

$$K \cdot \frac{W_{pd}}{L_{pd}} \left(\frac{V_{inv} - V_t}{2} \right)^2 = K \cdot \frac{W_{pu}}{L_{pu}} \left(\frac{-V_{td}}{2} \right)^2$$

$$\frac{W_{pd}}{L_{pd}} (V_{inv} - V_t)^2 = \frac{W_{pu}}{L_{pu}} (-V_{td})^2$$

$$Z_{pu} = \frac{L_{pu}}{W_{pu}} \quad \& \quad Z_{pd} = \frac{L_{pd}}{W_{pd}}$$

$$\frac{1}{Z_{pd}} (V_{inv} - V_t)^2 = \frac{1}{Z_{pu}} (-V_{td})^2$$

$$(V_{inv} - V_t)^2 = \frac{Z_{pd}}{Z_{pu}} (-V_{td})^2$$

$$= \frac{1}{\frac{Z_{pu}}{Z_{pd}}} (-V_{td})^2$$

$$(V_{inv} - V_t) = \frac{-V_{td}}{\sqrt{\frac{Z_{pu}}{Z_{pd}}}}$$

$$V_{inv} = V_t - \frac{V_{td}}{\sqrt{\frac{Z_{pu}}{Z_{pd}}}}$$

typical value $V_t = 0.2 V_{DD}$

$V_{td} = -0.6 V_{DD}$

$V_{inv} = 0.5 V_{DD}$

$$0.5V_{DD} = 0.2V_{DD} + \frac{0.6V_{DD}}{\sqrt{\frac{Z_{pu}}{Z_{pd}}}}$$

$$0.5V_{DD} - 0.2V_{DD} = \frac{0.6V_{DD}}{\sqrt{\frac{Z_{pu}}{Z_{pd}}}}$$

$$\frac{0.3V_{DD}}{0.6V_{DD}} = \frac{1}{\sqrt{\frac{Z_{pu}}{Z_{pd}}}}$$

$$\sqrt{\frac{Z_{pu}}{Z_{pd}}} = \frac{2}{1}$$

$$\boxed{\frac{Z_{pu}}{Z_{pd}} = \frac{4}{1}}$$

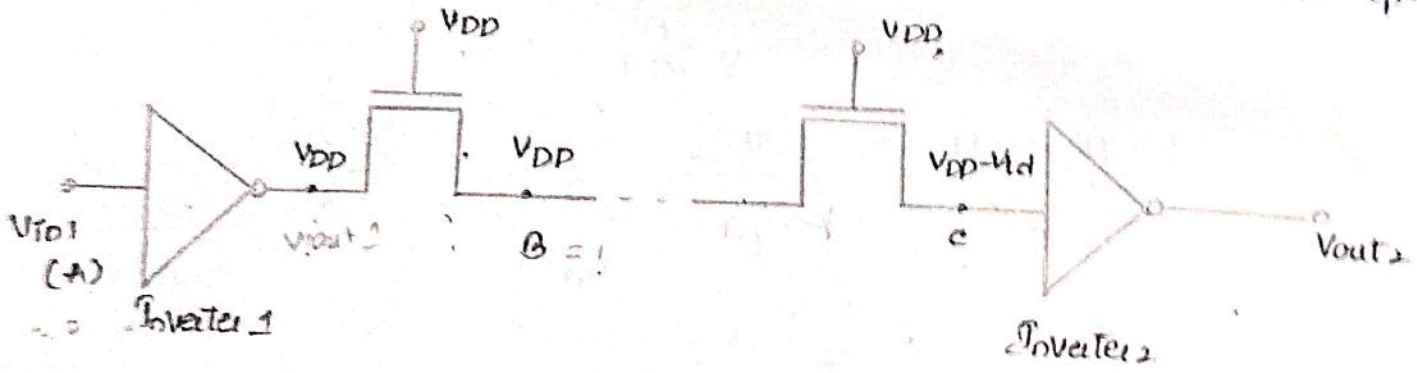
$$\boxed{Z_{pu} : Z_{pd} = 4:1}$$

Determination of pullup to pulldown ratio (Z_{pu}/Z_{pd}) for one inverter driven by another N-Mos inverter through one or more pass transistors :

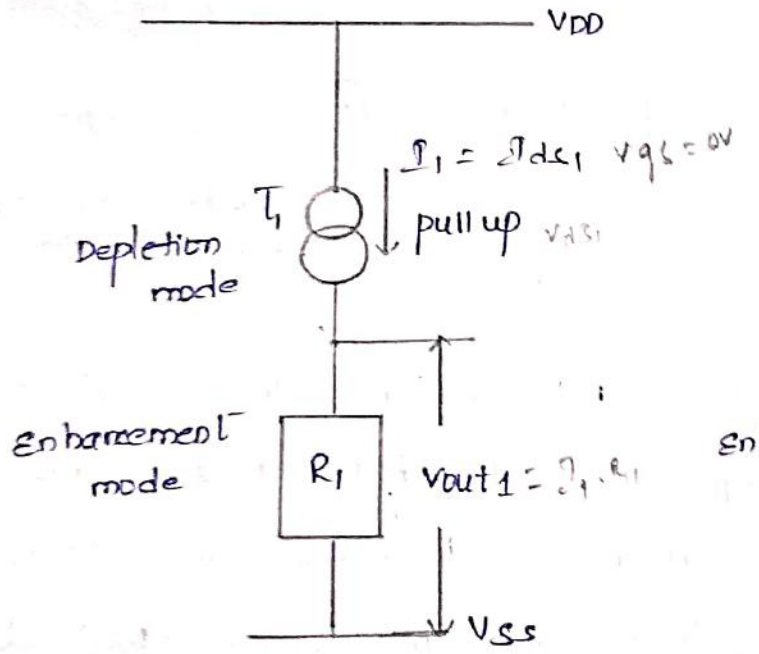
* consider the arrangement in which the input to inverter 2 comes from the o/p of inverter 1 but passes through one or more pass transistors used as switches in series. We connected the pass transistors in series will degrade the logic-1 level in to inverter 2 so that the output will not be a proper logic '0' level.

* the cct condition at point 'A' are connected to '0V' [∵ logic '0'] at point 'B' its VDD [logic '1'] but the voltage in to inverter 2 at point 'c' is reduced from VDD by threshold voltage [$V_{DD} - V_{tp}$] of the pass transistor. We consider this

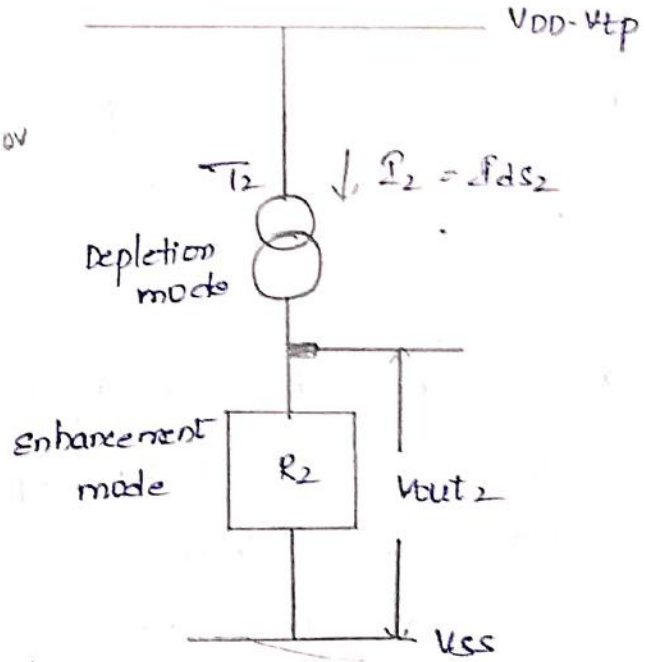
loss as V_{tp} so the i/p of the Inverter 2 is $V_{DD} - V_{tp}$.



Consider the Inverter-1 with the i/p is equals to V_{DD}. If the i/p is at V_{DD} then the pull down transistor T_2 is conducting but with low voltage across it therefore it is indicated as a resistive region. represented by R_1. the pull up transistor T_1 in saturation and represented as current source.



fig(a): Inverter 1 with i/p = V_{DD}



fig(b): Inverter 2 with input = $V_{DD}-V_{tp}$

For Enhancement mode of pull down device or transistor.

$$I_{ds} = K \cdot \frac{W_{pd1}}{L_{pd1}} \left[(V_{DD} - V_t) V_{ds1} - \frac{V_{ds1}^2}{2} \right]$$

$$I_{ds} = K \cdot \frac{W_{pd1}}{L_{pd1}} \left[(V_{DD} - V_t) - \frac{V_{ds1}}{2} \right] V_{ds1}$$

$$R_1 = \frac{V_{ds1}}{I_{ds}} = \frac{1}{k} \cdot \frac{L_{pd1}}{W_{pd1}} \left[\frac{1}{(V_{DD} - V_t) - \frac{V_{ds1}}{2}} \right]$$

For depletion mode pull-up Transistors.

$$I_{ds1} = I_1 = k \cdot \frac{W_{pu1}}{L_{pu1}} \left[\frac{(V_{gs} - V_{td})^2}{2} \right] \quad ; V_{gs} = 0$$

$$I_1 = k \cdot \frac{W_{pu1}}{L_{pu1}} \left[\frac{(-V_{td})^2}{2} \right]$$

W.K.T $V_{out1} = I_1 \cdot R_1$

$$V_{out1} = k \cdot \frac{W_{pu1}}{L_{pu1}} \left[\frac{(-V_{td})^2}{2} \right] \cdot \frac{1}{k} \cdot \frac{L_{pd1}}{W_{pd1}} \left[\frac{1}{V_{DD} - V_t - \frac{V_{ds1}}{2}} \right]$$

If V_{ds1} is small then $\frac{V_{ds1}}{2}$ is very small, so we can neglect

$$V_{out1} = \frac{L_{pd1}}{L_{pu1}} \left[\frac{1}{V_{DD} - V_t} \right] \cdot \frac{1}{\frac{L_{pu1}}{W_{pu1}}} \left[\frac{(-V_{td})^2}{2} \right]$$

$$V_{out1} = \frac{L_{pd1}}{L_{pu1}} \left[\frac{(-V_{td})^2}{V_{DD} - V_t} \right]^{1/2}$$

For Enhancement mode of pull down Transistors with input $V_{DD} - V_{tp}$ in figure (b).

$$R_2 = \frac{1}{k} \cdot \frac{L_{pd2}}{W_{pd2}} \left[\frac{1}{V_{DD} - V_{tp} - V_t} \right]$$

For depletion mode pull-up transistors

$$I_{ds2} = I_2 = k \cdot \frac{W_{pu2}}{L_{pu2}} \left[\frac{(-V_{td})^2}{2} \right]$$

$$V_{out2} = I_2 \cdot R_2$$

$$V_{out} = K \cdot \frac{W_{pu_2}}{L_{pu_2}} \frac{(-V_{td})^2}{2} \cdot \frac{1}{K} \cdot \frac{L_{pd_2}}{W_{pd_2}} \left[\frac{1}{V_{DD}-V_{tp}-V_t}\right]$$

$$= \frac{1}{Z_{pu_2}} \frac{(-V_{td})^2}{2} \cdot Z_{pd_2} \left[\frac{1}{V_{DD}-V_{tp}-V_t}\right]$$

$$= \frac{Z_{pd_2}}{Z_{pu_2}} \left[\frac{(-V_{td})^2}{V_{DD}-V_{tp}-V_t}\right]^{1/2}$$

$\therefore V_{out_1} = V_{out_2}$

$$I_1 R_1 = I_2 R_2$$

$$\frac{Z_{pd_1}}{Z_{pu_1}} \left[\frac{(-V_{td})^2}{V_{DD}-V_{tp}}\right]^{1/2} = \frac{Z_{pd_2}}{Z_{pu_2}} \left[\frac{(-V_{td})^2}{V_{DD}-V_{tp}-V_t}\right]^{1/2}$$

$$\frac{Z_{pd_1}}{Z_{pu_1}} \left[\frac{1}{V_{DD}-V_t}\right] = \frac{Z_{pd_2}}{Z_{pu_2}} \left[\frac{1}{V_{DD}-V_{tp}-V_t}\right]$$

$$\frac{Z_{pd_1}}{Z_{pu_1}} = \frac{Z_{pd_2}}{Z_{pu_2}} \left[\frac{V_{DD}-V_t}{V_{DD}-V_{tp}-V_t}\right]$$

typical values of $V_t = 0.2V_{DD}$, $V_{tp} = 0.3V_{DD}$

$$\frac{Z_{pu_2}}{Z_{pd_2}} = \frac{Z_{pu_1}}{Z_{pd_1}} \left[\frac{V_{DD}-V_t}{V_{DD}-V_{tp}-V_t}\right]$$

by applying the values of V_t & V_{tp}. we get

$$\frac{Z_{pu_2}}{Z_{pd_2}} = \frac{Z_{pu_1}}{Z_{pd_1}} \left[\frac{V_{DD}-0.2V_{DD}}{V_{DD}-0.3V_{DD}-0.2V_{DD}}\right]$$

$$\frac{Z_{pu_2}}{Z_{pd_2}} = \frac{Z_{pu_1}}{Z_{pd_1}} \left[\frac{0.8V_{DD}}{0.5V_{DD}}\right]$$

$$\frac{z_{pu_2}}{z_{pd_2}} = \frac{2}{1}\left[\frac{z_{pu_1}}{z_{pd_1}}\right]$$

$$\frac{z_{pu_2}}{z_{pd_2}} = \frac{2}{1}\left[\frac{4}{1}\right]$$

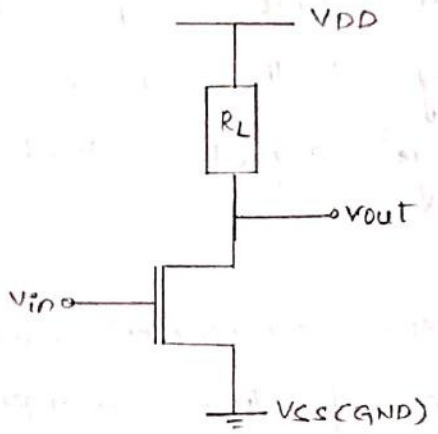
$$\boxed{\frac{z_{pu_2}}{z_{pd_2}} = \frac{8}{1}}$$

One N-MOS inverter is driven to the another N-MOS Inverter through one or more pass transistors & the ratio of $\frac{z_{pu_2}}{z_{pd_2}}$ is 8:1

Alternative forms of the pull-ups:

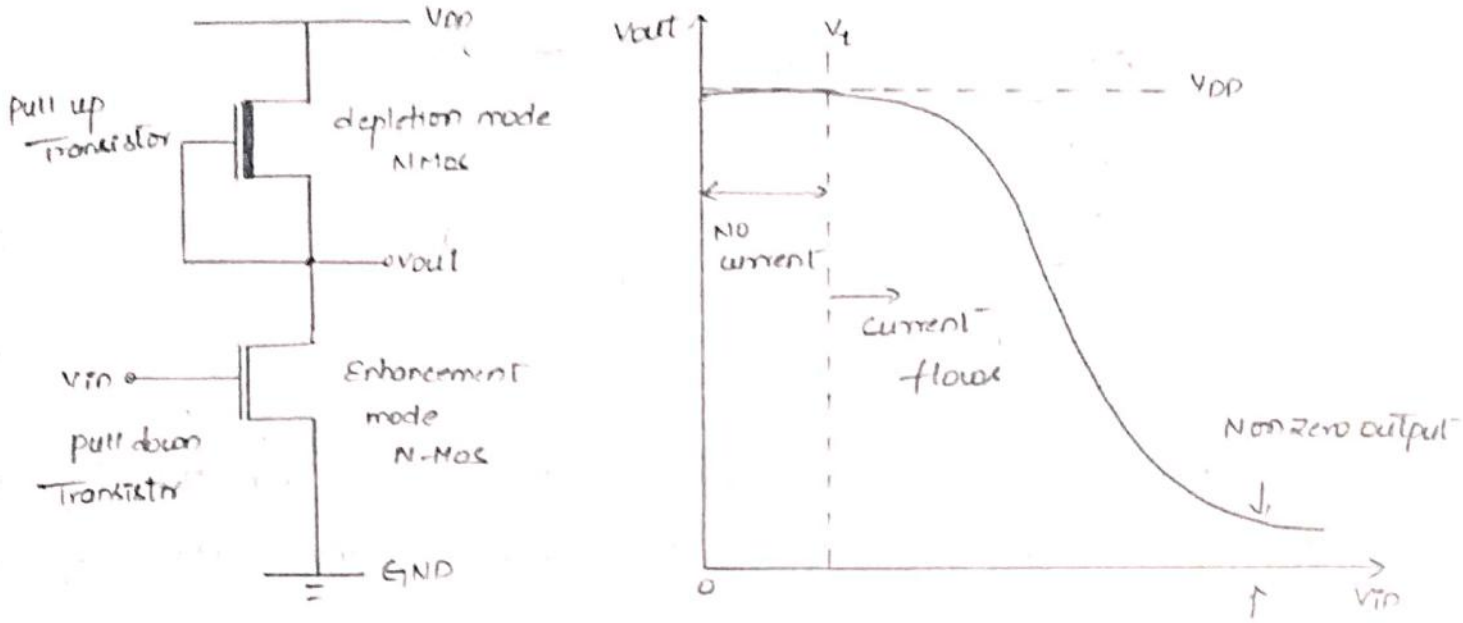
In N-MOS inverter ckt, we have assumed that the inverter ckt has a depletion mode pull transistor as its load. there are atleast four possible arrangements.

1) Load Resistance R_L as load:



This arrangement consists of a load resistor as pullup But it is not widely used because of the large space requirements of resistors produced to silicon substrate.

2> N-MOS depletion mode transistor as pull up :



This arrangement consists of a depletion mode transistor as pull-up. In this type of arrangement we observe.

(a) Dissipation is high, since rail to rail current flows when Vin = logical '1'.

(b) Switching of output from '1' to '0' begins when Vin exceeds V_t of pull down device.

(c) When switching the output from 1 to 0, the pull-up device is non saturated initially & this presents lower resistance through which to charge capacitive loads.

3> NMOS Enhancement mode pull up:

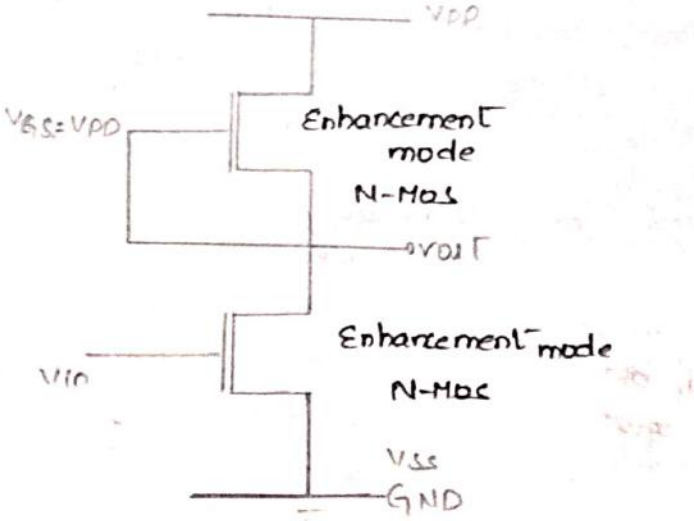
This arrangement consists of a NMOS enhancement mode transistor as pull-up. The important features of this arrangements are.

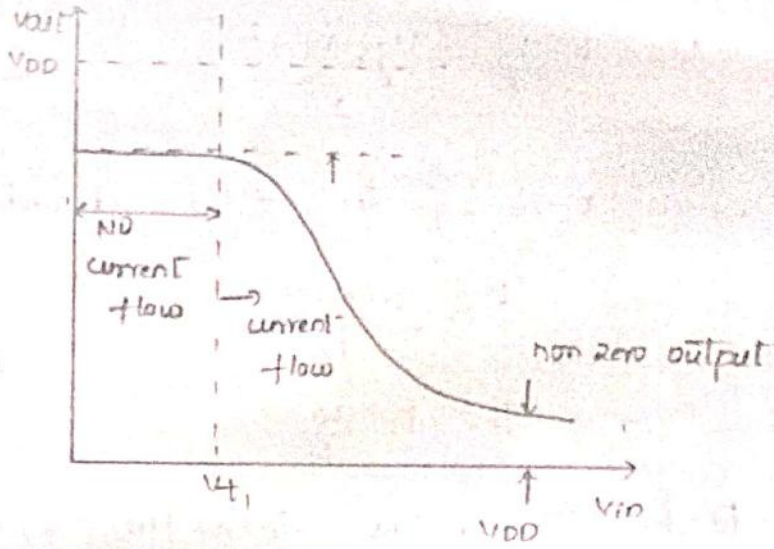
(a) Dissipation is high since current flows when Vin = logical '1'.

(b) Vout can never reach VDD (logic '1') if $V_{GG} = V_{DD}$ as is normally the case.

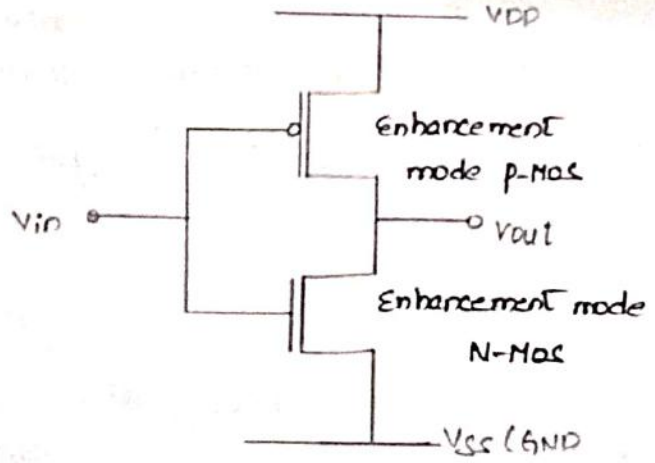
(c) V_{GG} may be derived from a switching source, for example one phase of a clock, so that dissipation can be greatly reduced

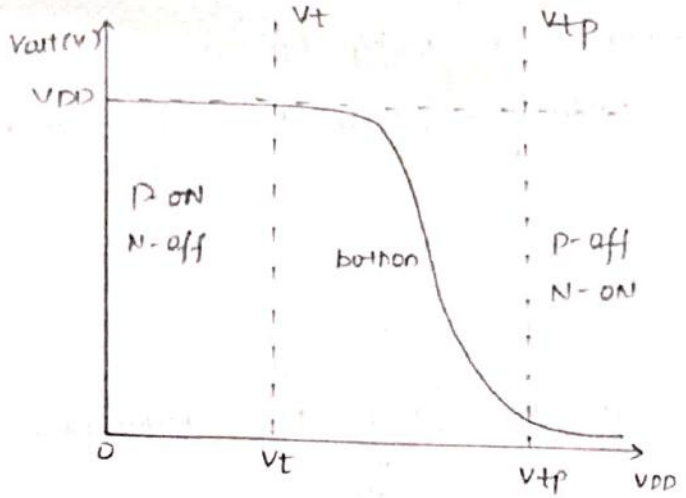
(d) If V_{GG} is higher than V_{DD} then an extra supply rail is required.





4) complementary device pull-up:





This arrangement consists of a CMOS arrangement as pull-up The salient features of this arrangement are.

(a) No current flows either for logical 0 or for logical '1' inputs.

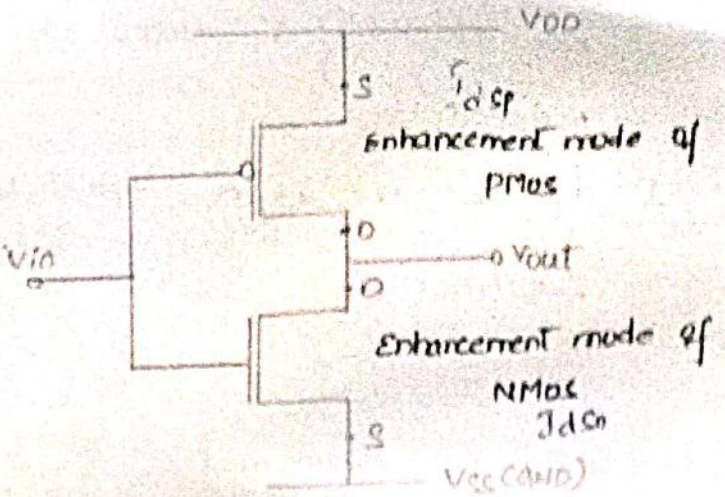
(b) Full logical '1' & '0' are presented at the output.

(c) For devices of similar dimensions of p-MOS ~~ab~~ is slower than the N-MOS devices.

CMOS Inverter – Analysis and design:

The Inverter in resistive Region

$$I_{ds} = K \cdot \frac{W}{L}\left[(V_{gs}-V_t)V_{ds} - \frac{V_{ds}^2}{2}\right]$$



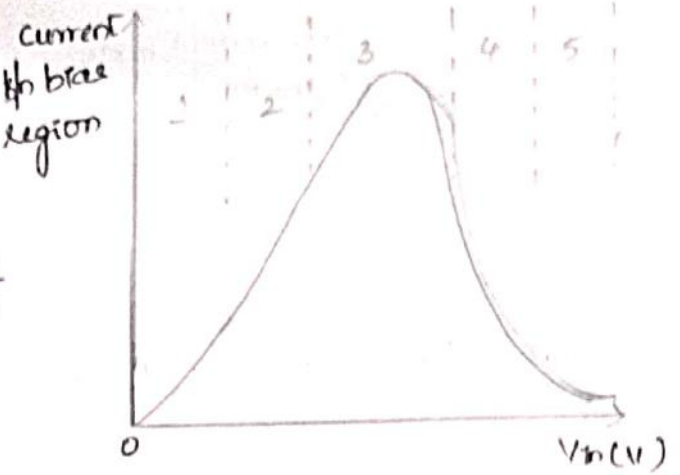
The Inverter in saturation region

$$I_{ds} = K \cdot \frac{W}{L}\frac{(V_{gs}-V_t)^2}{2}$$

where K is factor it is dependent on technology.

$$K = \frac{\varepsilon_0 \varepsilon_{ins} \mu}{D}$$

$$\beta = K \cdot \frac{W}{L}$$



β for NMOS $\quad \beta_n = \frac{\varepsilon_0 \varepsilon_{ins} \mu_n}{D} \cdot \frac{W_n}{L_n}$

β for PMOS $\quad \beta_p = \frac{\varepsilon_0 \varepsilon_{ins} \mu_p}{D} \cdot \frac{W_p}{L_p}$

where W_n, L_n, W_p, L_p are widths & lengths of the NMOS & P-MOS respectively. μ_n & μ_p are mobilities of NMOS & PMOS respectively.

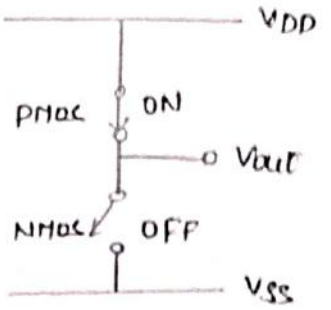
Region 1:

when Vin = logic 0

P-MOS is fully ON & NMOS is fully OFF

Vout = VDD

when Vout = VDD then there is no current flow from rail to rail.



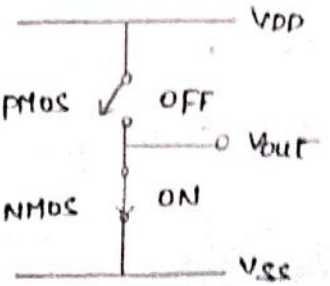
Region 5:

when Vin = logic 1

PMOS is fully OFF & NMOS is fully ON

Vout = GND (logic 0)

when Vout = GND (logic 0) then there is no current flowing from rail to rail.



Region 2 :

In the region 2, the i/p voltage increases, the level of i/p voltage is exceeds the threshold voltage of NMOS, NMOS conducts & it provides the maximum voltage across it, but minimum voltage across p-MOS transistor is in unsaturated resistive region. & NMOS is in saturation region & acts as current source.

* A small current now flows through the inverter from VDD to Vss.

Region 4: It is similar to region 2 but with the roles of PMOS & NMOS are reversed. How ever the i/p voltage is increases & it excess the threshold voltage of PMOS transistor then its conducts, & it is in saturation region. & NMOS is in unsaturated region. So minimum (or) small current flows through the inverter from VDD to Vss, there is a voltage drop

Region 3:

The input voltage is in between threshold voltage of NMOS as well as PMOS transistor [Vtn & Vtp]. In this region both transistors are in saturation region then this condition is

$$I_{dsp} = -I_{dsn}$$

$$I_{dsp} = \frac{\beta_p}{2} (V_{in} - V_{DD} - V_{tp})^2$$

$$I_{dsn} = \frac{\beta_n}{2} (V_{in} - V_{tn})^2$$

$$\frac{\beta_p}{2} [V_{in} - V_{DD} - V_{tp}]^2 = -\frac{\beta_n}{2} [V_{in} - V_{tn}]^2$$

$$[V_{in} - V_{DD} - V_{tp}]^2 = -\frac{\beta_n}{\beta_p} [V_{in} - V_{tn}]^2$$

Taking square root on both sides

$$(V_{in} - V_{DD} - V_{tp}) = -\sqrt{\frac{\beta_n}{\beta_p}}\,(V_{in} - V_{tn})$$

$$V_{in} - V_{DD} - V_{tp} = -\sqrt{\frac{\beta_n}{\beta_p}}\,V_{in} + \sqrt{\frac{\beta_n}{\beta_p}}\,V_{tn}$$

$$V_{in} + \sqrt{\frac{\beta_n}{\beta_p}}\,V_{in} = \sqrt{\frac{\beta_n}{\beta_p}}\,V_{tn} + V_{tp} + V_{DD}.$$

$$V_{in}\left(\sqrt{\frac{\beta_n}{\beta_p}} + 1\right) = V_{DD} + V_{tp} + V_{tn}\sqrt{\frac{\beta_n}{\beta_p}}$$

$$V_{in} = \frac{V_{DD} + V_{tp} + V_{tn}\sqrt{\frac{\beta_n}{\beta_p}}}{\left(\sqrt{\frac{\beta_n}{\beta_p}} + 1\right)}$$

when typical values $\beta_n = \beta_p$ & $V_{tp} = -V_{tn}$

$$V_{in} = \frac{V_{DD} + \cancel{-V_{tn}} \cancel{- V_{tn}}}{+1}$$

$$V_{in} = \frac{V_{DD}}{2}$$

$$\boxed{V_{in} = 0.5\,V_{DD}}$$

At minimum i/p voltage

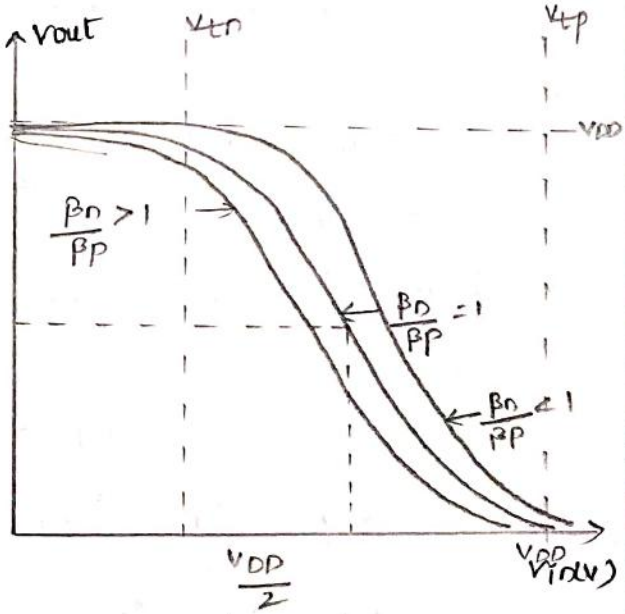
$$\beta_n = \beta_p$$

$$\frac{\varepsilon_0 \varepsilon_{ins} \cancel{\mu_n}}{\cancel{D}} \cdot \frac{W_n}{L_n} = \frac{\varepsilon_0 \varepsilon_{ins} \cancel{\mu_p}}{\cancel{D}} \cdot \frac{W_p}{L_p}$$

$$\mu_n \cdot \frac{W_n}{L_n} = \mu_p \cdot \frac{W_p}{L_p}$$

$$\frac{W_p}{L_p} = \frac{\mu_n}{\mu_p} \cdot \frac{W_n}{L_n}$$

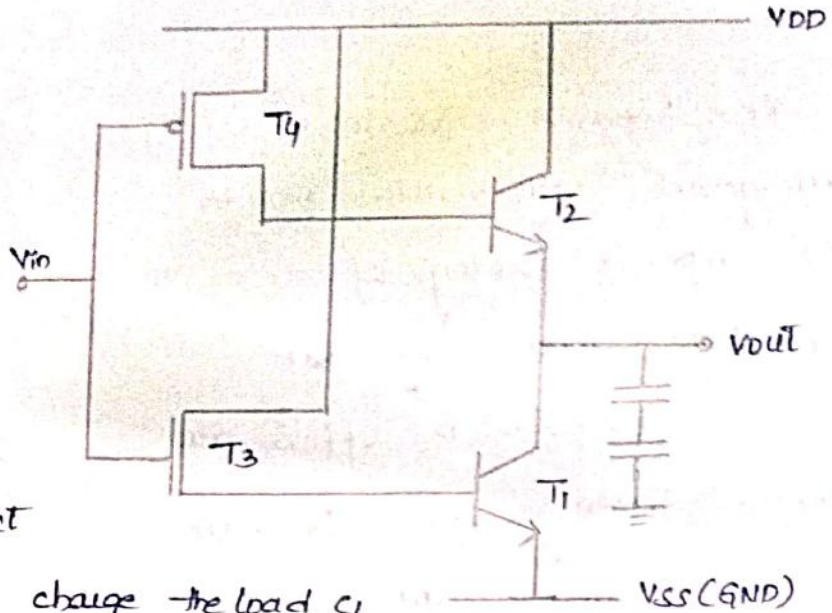
$$\frac{W_p}{L_p} \simeq 2.5\,\frac{W_n}{L_n}$$



The dimensions of NMOS transistor is greater than PMOS transistor.

BICMOS Inverter:

It consists of two Bipolar transistors T_1 & T_2 along with one NMOS Transistor T_3 & one PMOS Transistor T_4 both are Enhancement mode Transistors



Characteristics:

Case(I):

Initially Vin = logic '0'.

In this case. T_3 is OFF, so T_1 will be non conducting & T_4 is ON & supplies current to base of Transistor T_2 which will conduct & acts as current source to charge the load C_L to +5V.

$V_{out} = V_{DD} - V_{BE}$

Case(II): Vin = logic 1

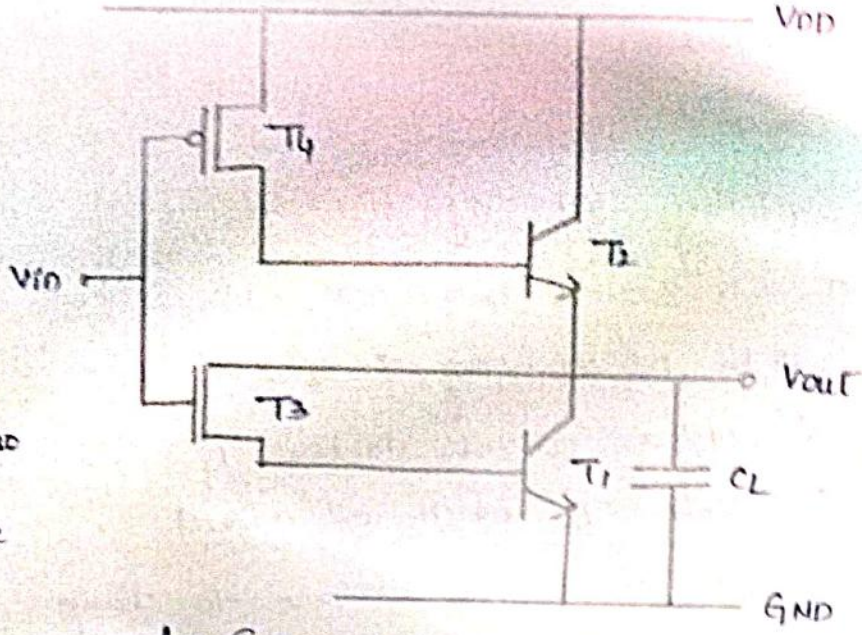
T_4 is OFF, So T_2 is non conducting & T_3 is ON & supply current to base of T_1 which conducts & acts as a current sink. to the load. C_L discharging towards 0V (GND).

Case(III):

T_1 & T_2 will present low impedance when turned on into saturation & load capacitor C_L will be charged or discharged rapidly.

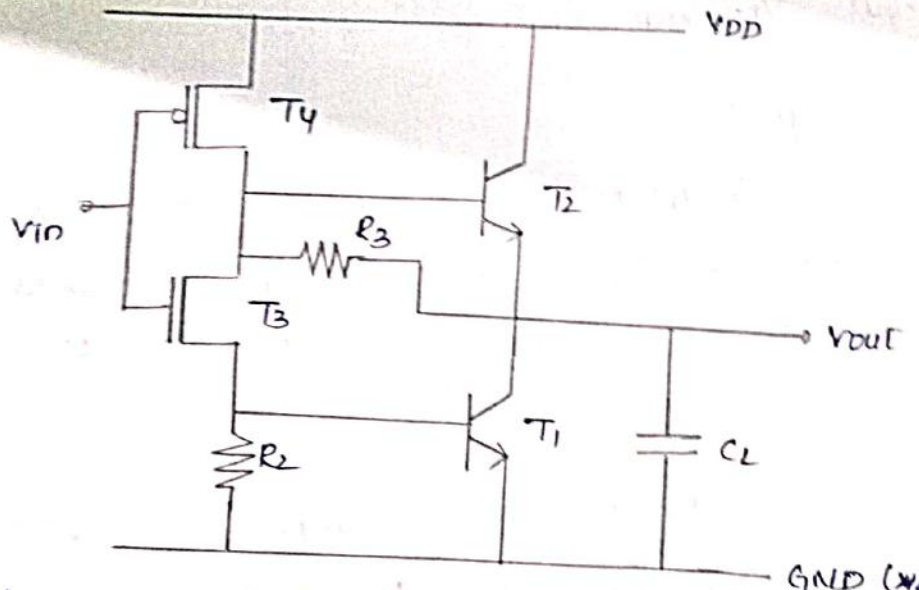
Here the presence of dc path from V_{DD} to ground through T_3 & T_1 as shown in figure is not a good arrangement to complement. Since there will be a significant static current flow whenever logic '1' = Vin there is a problem in that there is no discharging path for current from the base of either BJT

When it is turned off this will slow down the action of ckt. In the given below ckt T_3 & T_1 is eliminated but the voltage swing is not reduced since the o/p cannot fall below V_{BE} of T_1

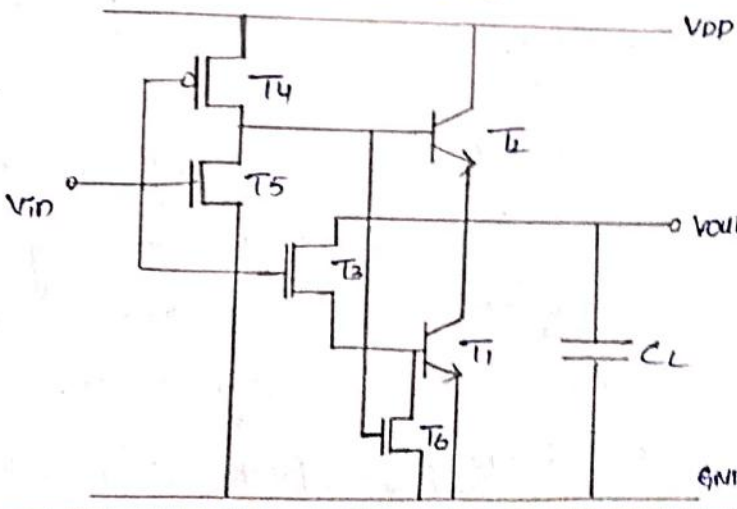


* An Improved Inverter arrangement with resistors provide the improved swing of o/p voltage when each Bipolar Transistor is off & also provide discharge path for base current during turn off.

fig: Improved version of BiCMOS Inverter

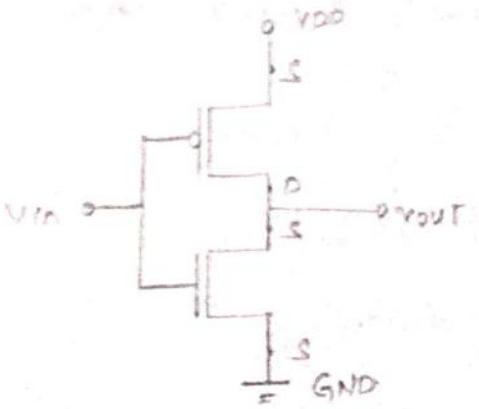


The provision of onchip resistor values are not available. this is the drawback. & by placing the T_5 & T_6 Transistors we can overcome the drawback & is shown in figure.

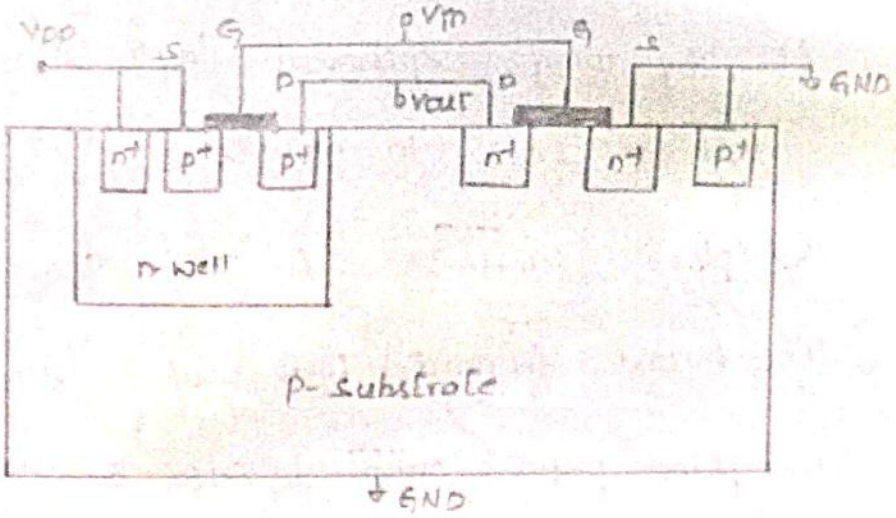


Latch up in CMOS:

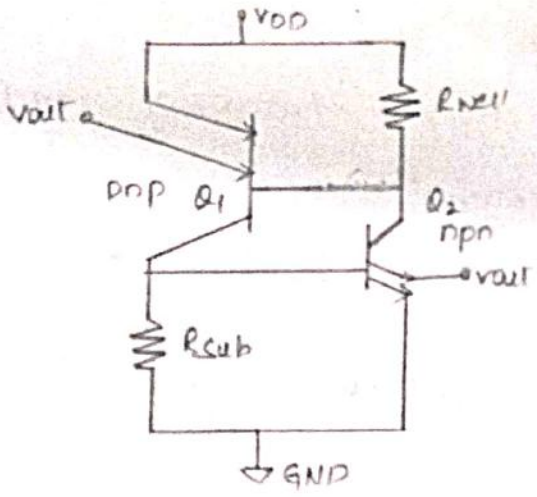
Latch up is defined as the generation of low impedance path in CMOS states in b/n V_{DD} (supply) & ground (Vss). so high amount of current is driven by V_{DD} (supply) toward ground and that may damage CMOS ckt.



Basic CMOS Inverter



Fabrication of CMOS Inverter



There are two conditions which we need to take care for latch up condition.

Condition 1 : If V_{in} or $V_{out} > V_{DD}$

the emitter base junction of Q_1 is ON. (forward biased) and the current flow in base Q_2 which triggers npn transistor to ON. So current is flow from VDD to GND.

condition 2: If V_{in} or $V_{out} < GND$

the emitter base junction of Q_2 is forward biased & is in ON. (npn transistor). the base current of pnp transistor Q_1 is dragged to turn ON. So the current flow from V_{DD} through Q_1 & Q_2 to GND.

* Latch up means if we have to turn ON Q_1 & Q_2 from V_{DD} to GND after we cannot turn it off by giving i/p supply. So base

now the control will not be there with input.

Latch up prevention steps:

1) By adding gold impurities in substrate we can minimize latch up.

2) By using epitaxial layer at substrate which reduces Rsub to minimize latch up.

3) place substrate & well as close as possible.

4) Avoid froward bias of source & drain.

5) place PMOS close to VDD & NMOS close to GND.

Problems on Ids Vs VDS relation ship :

1) find gm & rds for an n-channel transistor with Vgs = 1.2V, Vtn = 0.8V, $\frac{W}{L} = 10$, $\mu_n C_{ox} = 92 \mu A/V^2$ & Vds = Veff + 0.5V. the output impedance constant = $95.3 \times 10^{-3}/V$

Sol: $\lambda = 95.3 \times 10^{-3}/V$, Vds = Veff + 0.5V

where Veff = Vgs - Vtn

= 1.2V - 0.8V

Veff = 0.4V

Vds = 0.4V + 0.5V

Vds = 0.9V

0.9V > 0.4V. $\Rightarrow$ (∵ Vds > Veff)

So Transistor is in saturation region

$$I_d = \frac{\mu_n C_{ox}}{2} \frac{W}{L} (V_{gs} - V_{tn})^2$$

Transconductance, $g_m = \frac{C_g \mu}{L^2} \cdot V_{ds}$.

$C_g = C_{ox} WL$ (C_{ox} < unit capacitance)

$$g_m = \frac{\mu_n C_{ox} WL}{L^2} \cdot V_{ds}$$

$$= \mu_n C_{ox} \frac{W}{L} \cdot V_{ds}$$

$$= 92 \times 10^{-6} \times 0.9 \times 10$$

$$\boxed{g_m = 8.28 \times 10^{-4} \text{ } \sigma}$$

drain to source resistance, $r_{ds} = \frac{1}{\lambda I_{ds}}$

$$I_{ds} = \frac{\mu_n C_{ox}}{2} \cdot \frac{W}{L} (V_{gs} - V_{tn})^2$$

$$= \frac{92 \times 10^{-6}}{2} \times 10 \ (0.4)^2$$

$$I_{ds} = 7.36 \times 10^{-5} A$$

$$r_{ds} = \frac{1}{95.3 \times 10^{-3} \times 7.36 \times 10^{-5}}$$

$$\boxed{r_{ds} = 142.57 \text{ K}\Omega}$$

2) $V_{tn} = 0.8V$, $\lambda = 0.04V^{-1}$, $k_n' = 20\mu A/V^2$, $\frac{W}{L} = 20$, $V_D = 4V$, $V_G = 5V$, $V_S = 2V$ in MOSFET. find drain current I_D.

Sol: $V_{GS} = V_G - V_S = 5 - 2 = 3V$, $V_{DS} = V_D - V_S = 4 - 2 = 2V$

the relation b/n V_{GS} & V_{tn} ($V_{GS} > V_{tn}$) MOSFET is in linear or saturation region.

$V_{GS} - V_{tn} = 3V - 0.8 = 2.2V > V_{DS}$.

$2.2V > 2V \Rightarrow V_{GS} - V_{tn} > V_{DS}$ so, MOSFET is in linear region.

$$I_D = \frac{k_n}{2} \left(2 (V_{GS} - V_{tn}) V_{DS} - V_{DS}^2 \right) \qquad k_n = k_n' \times \frac{W}{L}$$

$$= \frac{1}{2} \times 20 \times 10^{-6} \times 20 \left[2 (2.2) 2 - 2^2 \right]$$

$I_D = 9.6 \times 10^{-4} = 0.96\ mA$

If MOSFET is in saturation region

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

λ = channel length modulation coefficient.

3) $\frac{W}{L} = \frac{10}{0.55}$, $k'_n = 110\mu A/V^2$, $\lambda = 0.05\ V^{-1}$, $V_{GS}=2V$, $V_{DS}=2V$, $V_{tn}=0.7V$ in MOSFET. find drain current I_D.

Sol: $V_{GS} > V_{tn}$ (linear or saturation region)

$V_{GS}-V_{tn} = 2V-0.7V = 1.3V$ $\quad V_{DS} = 2V$

$V_{GS}-V_{tn} < V_{DS} \Rightarrow 1.3V < 2V$ (if MOSFET is in saturation region then)

$$I_D = \frac{k_n}{2}(V_{GS}-V_{tn})^2 (1+\lambda V_{DS}) \quad \left[\because k_n = k'_n \cdot \frac{W}{L}\right]$$

$$= \frac{1}{2} \times 110\times10^{-6} \times \frac{10}{0.55} \times (1.3)^2 (1+0.05\times2)$$

$I_D = 1.859 \times 10^{-3}\ A = 1.859\ mA$

Unit-II

VLSI circuit Design processes

VLSI Design flow:

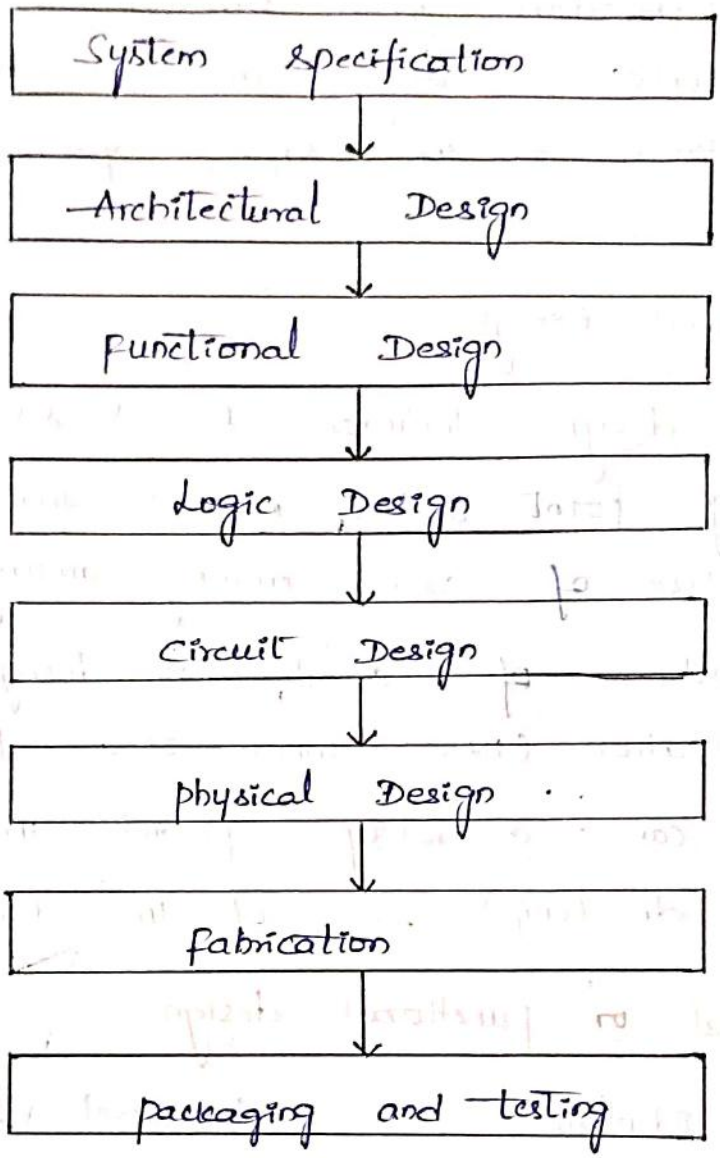


figure: VLSI Design flow flowchart

A typical design cycle may be represented by the flow chart as shown in figure.

1) System specification:

* The first step is system specification, is a high level representation of the s/m. The factors to be considered in this process includes performance functionality

and physical dimensions (size of the die (chip)). The fabrication technology and design techniques are involved & considered.

* The specification of a s/m is a compromise b/n market requirements, technology and economical viability. The specifications for the size, speed, power and functionality of VLSI.

2) Architectural Design:

* This design includes RISC verses CISC, number of ALU's, floating point units, numbers and structures of pipelines and size of cache memory. among others.

* The outcome of Architectural design is a micro Architectural specification (MAS) which is a textual (English) description that can accurately predict the performance, power and die (chip) size of the design.

3) Behavioral or functional design:

* This identifies the interconnect requirements b/n the units the area, power and other parameters of each unit is estimated. The behavioral aspects of the s/m are considered without implementation of specific information. ie. to specify the behaviour interms of input, output and timing of each unit. without specifying its internal structure.

* Functional or behavioral design provides quick. Emulation of the system and allows fast debugging of full system.

4) Logic Design:

* In this step the control flow, word widths, register allocation, arithmetic operations and logic operations of the design that represent the functional design are derived and tested.

* This description is called register Transfer level (RTL) description. RTL is expressed in Hardware description language. (HDL). This description can be used in simulation and verification. & its consists of boolean functions & timing information. which defines the behavioral description of design.

5) Circuit Design:

* The purpose of ckt design is to develop a ckt representation based on logic design. The boolean Expressions are converted in to a ckt representation by taking in to consideration the speed & power requirements of original design.

* The ckt design is usually expressed in detailed ckt diagram (cells, macros, gates, Transistors) and inter-connection b/n these elements. This representation is also called netlist. Tools used to manually enter such description are called ~~used~~ schematic capture tools. In many cases, netlist can be automatically created from logic (RTL) description by using logic synthesis tools.

6) physical design:

* In this step the ckt representation (netlist) is converted in to a geometric representation. this

Geometric representation of a cct is called a layout.

* Layout is created by converting each logic component into Geometric representation. (shapes in multiple layers) which perform the intended logic function of corresponding components.
* physical design is a very complex process & therefore it is usualy broken down into various sub steps. Various verification & validation checks are performed on the layout
* In many cases. layout can be generated directly from netlist by layout synthesis tools.

7) Fabrication:

* After layout & verification, the design is ready for fabrication. since layout data is typically sent to fabrication on a tape, the event of release of data is called tapeout. layout data is converted into photo lithographic masks. Masks identify spaces on wafer where certain materials need to be deposited, diffused or removed. during each step one mask is used. to complete the fabrication process.
* A large wafer is 20cm (8 inches) in diameter & can be used to produce hundreds of chips., depending of the size of chip.

8) Packaging, testing and Debugging:

* finally the wafer is fabricated & diced into individual chips. Each chip is then packaged & tested to ensure that it meets all the design specifications

and its functions properly. chips used in printed ckt boards (PCB) are packed in dual in line package (DIP), Pin Grid Array (PGA), Ball Grid Array (BGA), & Quad flat package (QFP)

MOS Layers:

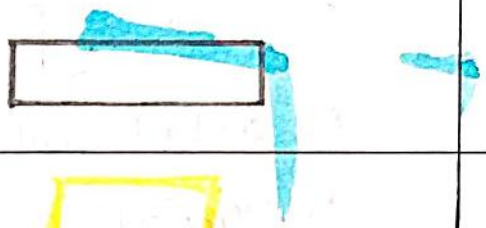
MOS circuits are formed on ~~some~~ basic layers:

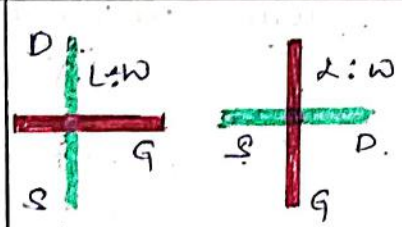
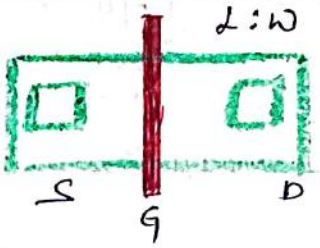
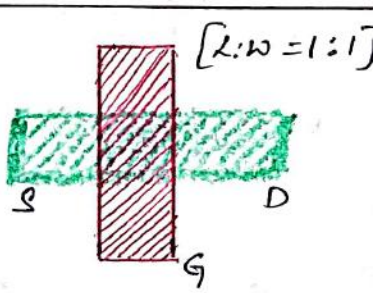
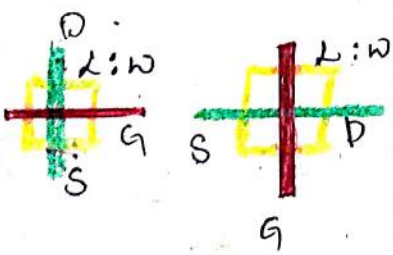
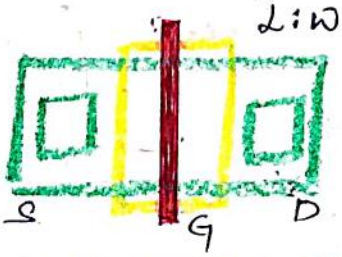
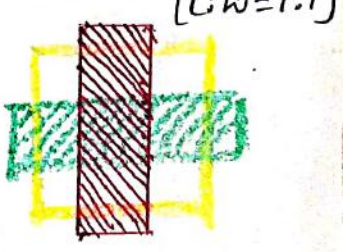
1) N- Diffusion → N-MOS → Green
2) p- diffusion → p-MOS → Yellow
3) polysilicon → Red
4) Metal → Metal 1 → Sky blue
   → Metal 2 → Dark blue
5) Contact cuts → Black
6) Implant → Yellow
7) Demarcation line → Gray
8) Burried contact → Brown

The fabrication process consists of several steps involving deposition and diffusion of various materials on the wafer

Stick Diagrams:

Encodings for NMOS process:

colour	Stick Encoding	Layers	Mask Layout Encoding	CP layer
Green		n+ diffusion Thindx		ND
RED		polysilicon		NP
Blue		Implant Metal 1		NM
BLACK		contact cut		NC
GRAY	NOT Applicable	overglass		NG
nMOS only Yellow		Implant		NI
nMOS only Brown		Buried contact		NB

Feature	Feature (Stick)	Feature (Symbol)	Feature (Mask)
n-type Enhancement mode transistor Transistor length to width ratio		 L:W should be shown	
n-type depletion mode transistor nMOS only			

* the above figure shows the way of representing different layers in stick diagram notation and mask layout using nmos style.

* stick diagram shows the layer information's in all types of layers. through color codes. (or monochrome encoding). and it acts as an Interface between symbolic circuit and the actual layout.

Transistor level modelling :-

Generally stick diagrams are used for even complicated circuits, we have to define the layout design ~~by using~~ with the help of stick diagrams.

NMOS Inverter :

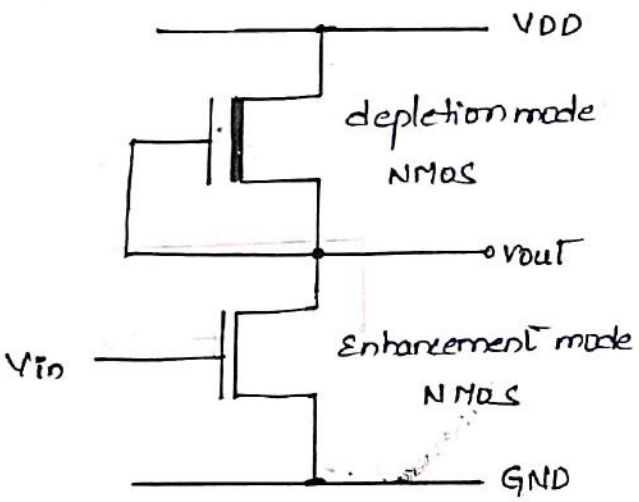


fig: NMOS Inverter circuit

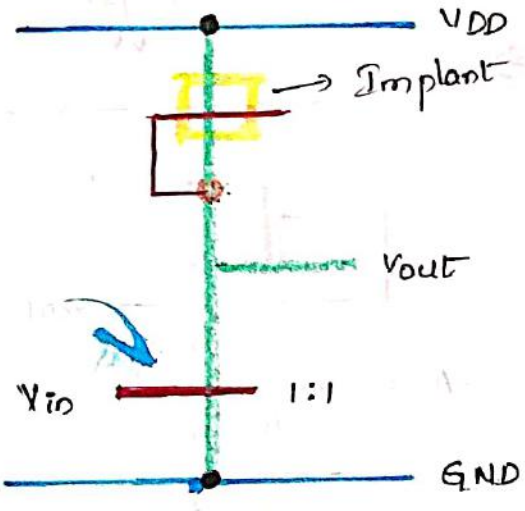


fig: Stick diagram of NMOS Inverter

Examples :

1) Draw the stick diagram of two input NOR gate in N-MOS technology.

Sol:

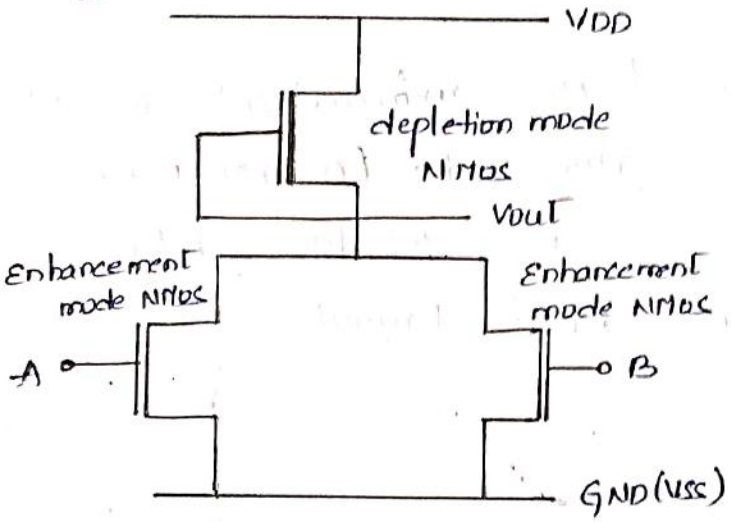


fig: Circuit diagram

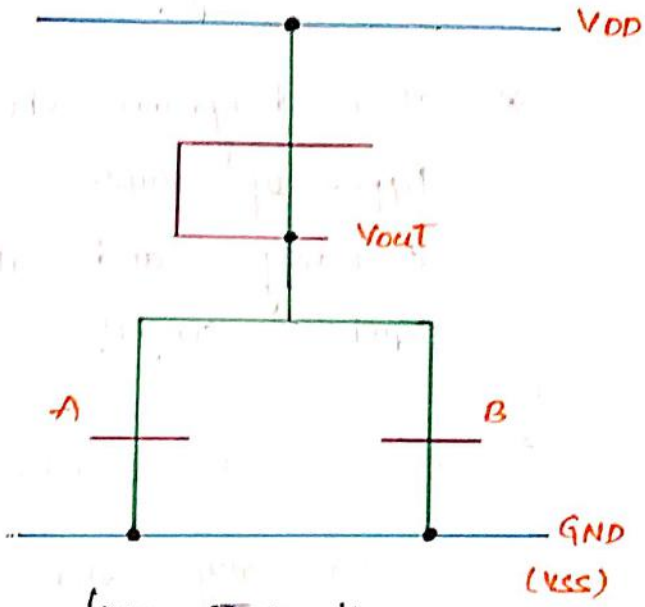
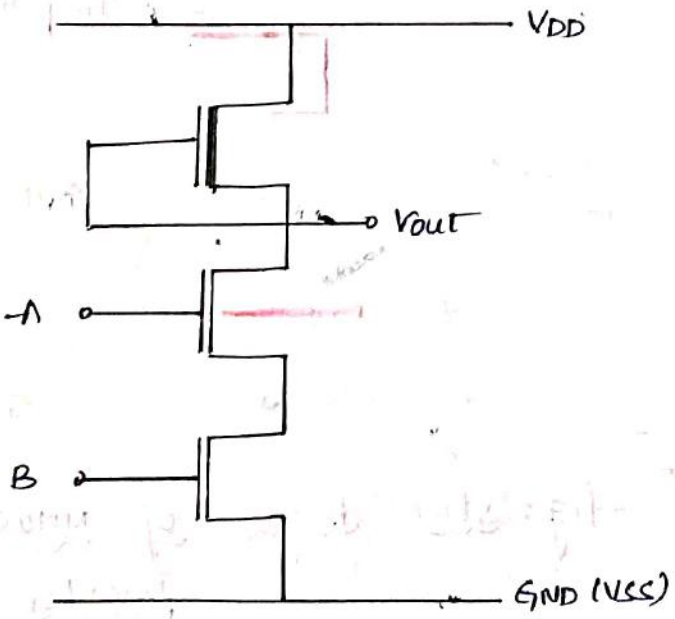


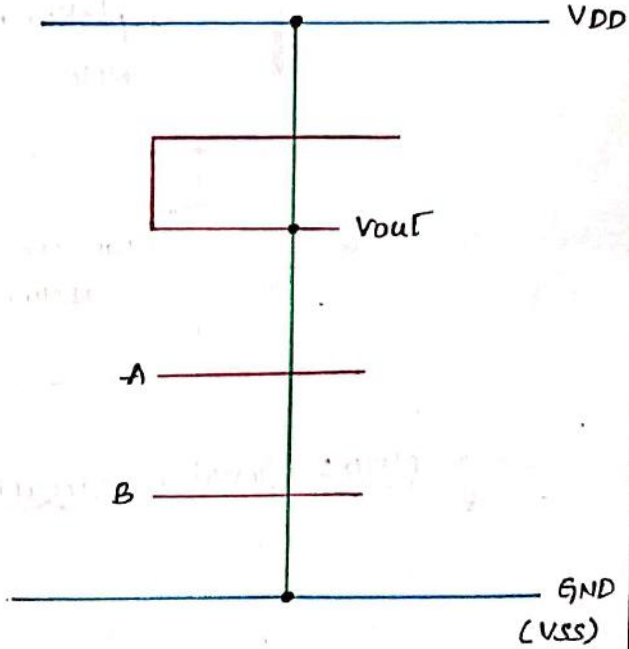
fig: Stick diagram

$V_{out} = \overline{A+B}$

2) Draw the stick diagram for two input NAND gate in N-MOS technology.

Sol:





3) Draw the stick diagram for CMOS Inverter.

Sol:

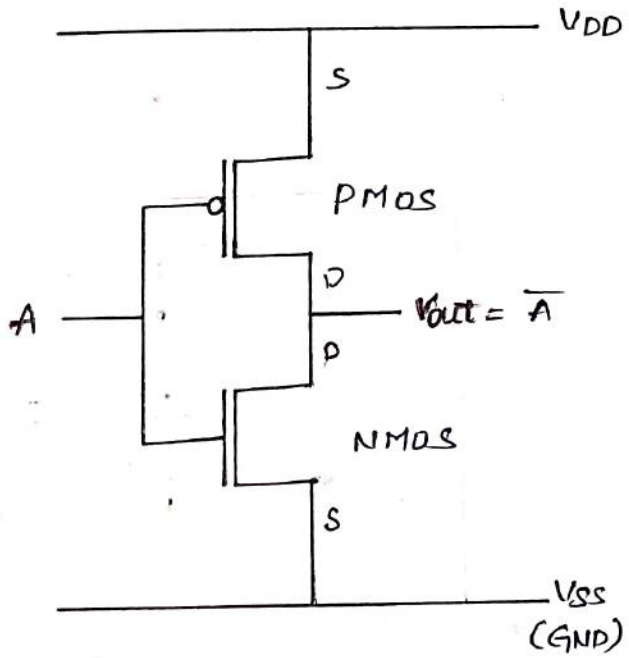


fig: CMOS Inverter ckt diagram

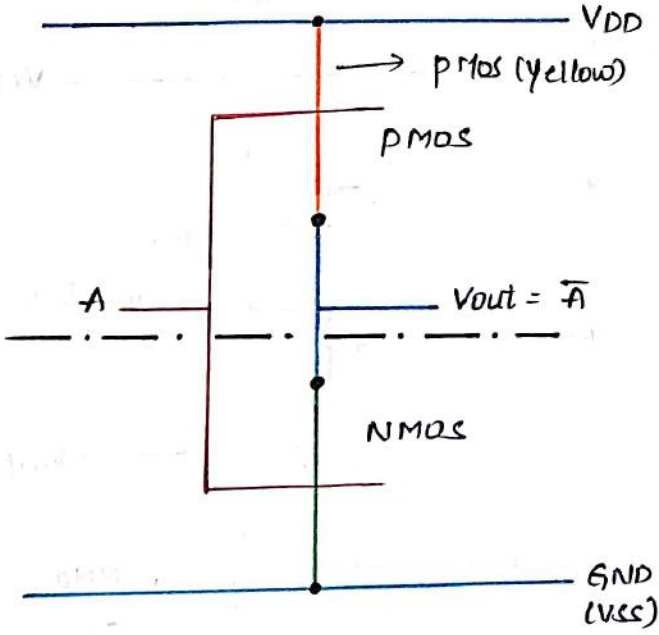


fig: stick diagram for CMOS Inverter

4) Draw the stick diagram for two input NAND Gate.

Sol:

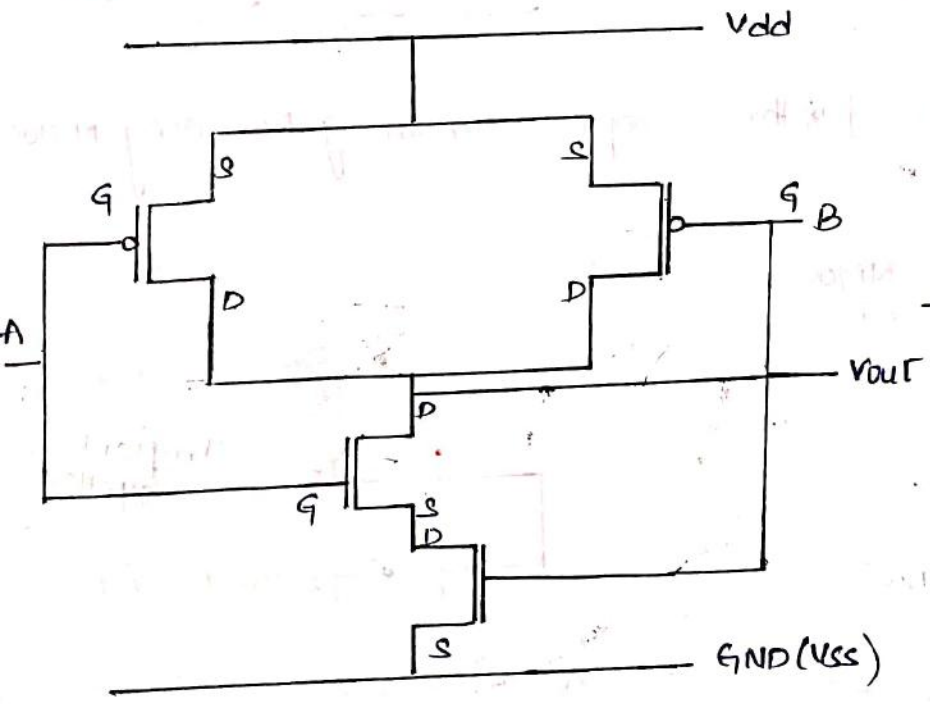
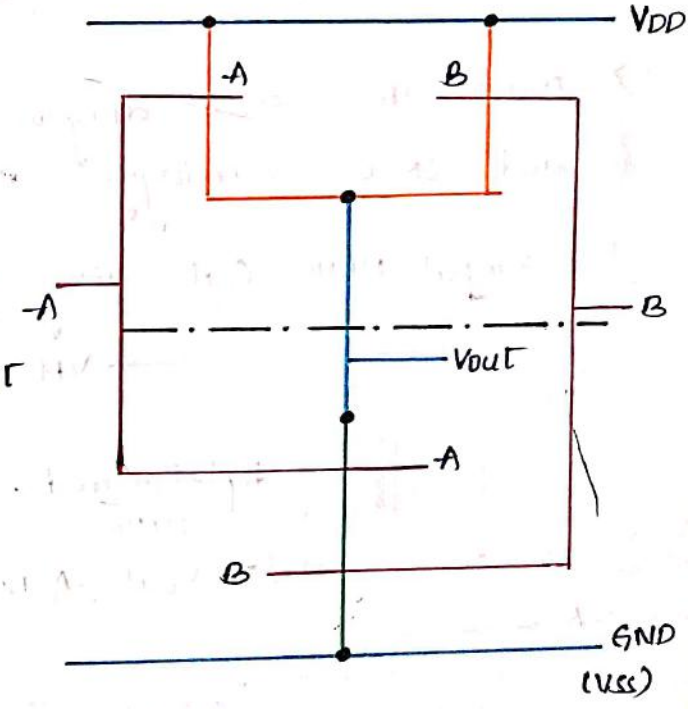


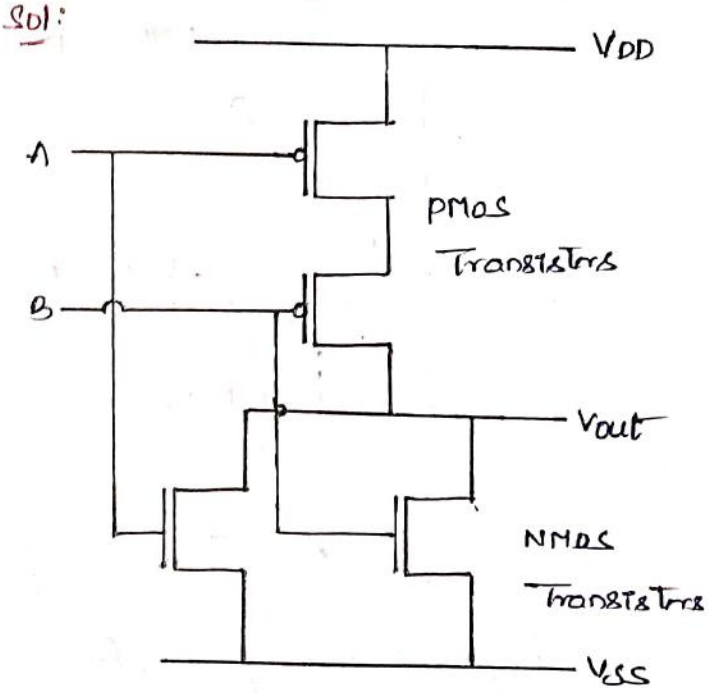
figure: Circuit for two input NAND Gate (CMOS AND Gate)



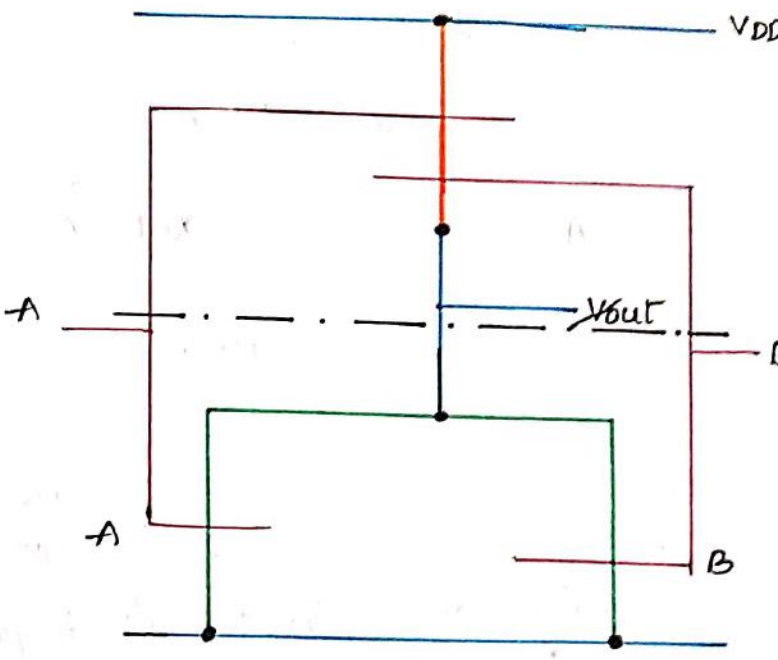
Stick diagram for CMOS NAND Gate

5) Draw the stick diagram for two input NOR gate using CMOS technology.

Sol:



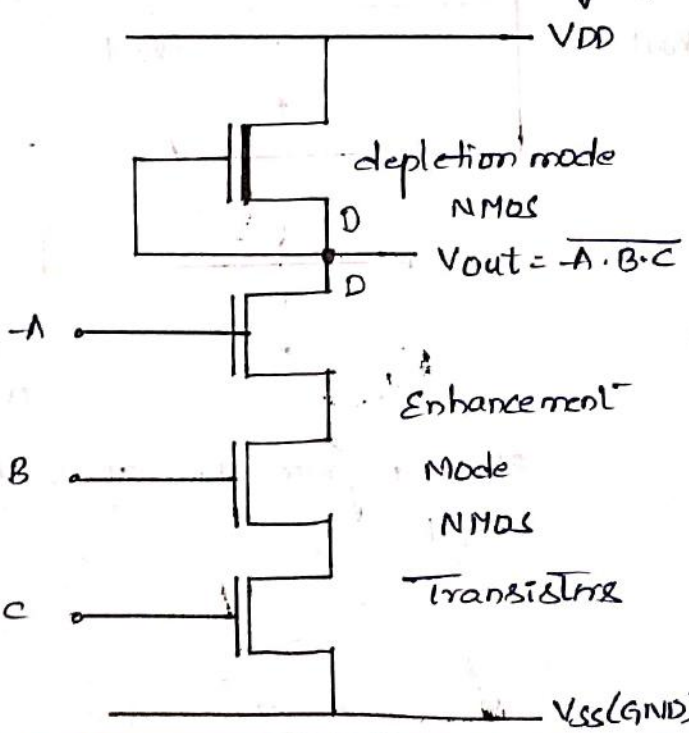
fig(a): CKT diagram for 2 i/p NOR Gate using CMOS Technology

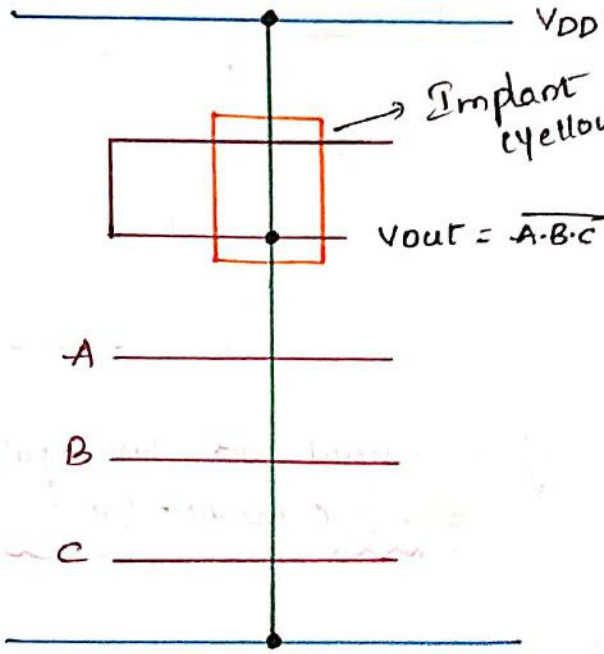


fig(b): Stick diagram for 2 i/p NOR Gate

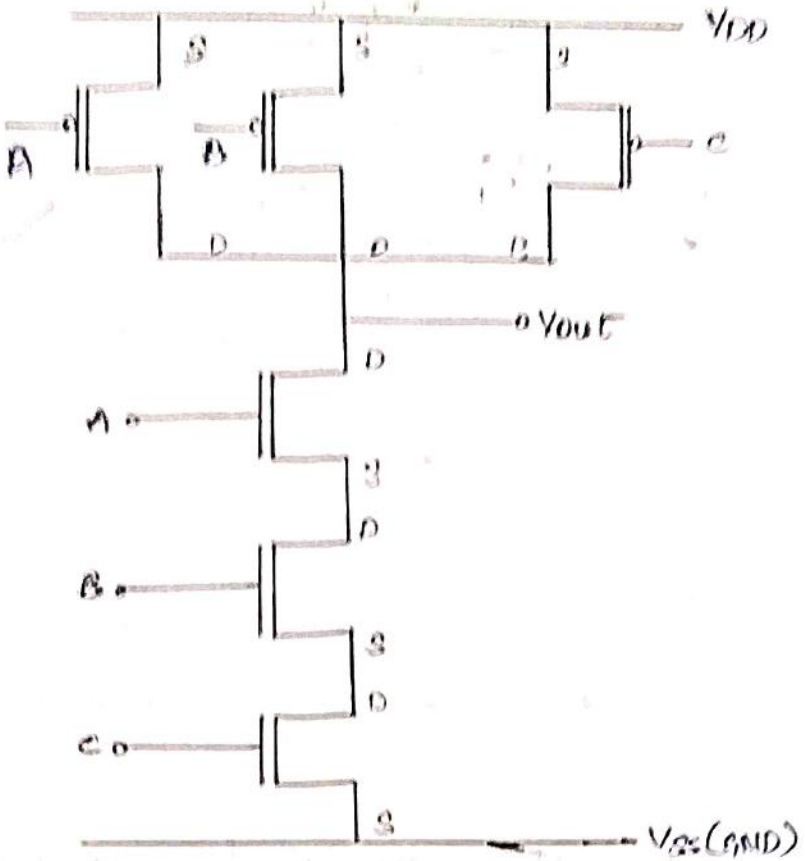
6) Draw the stick diagram for three input NAND gate using N-M and CMOS technology:

Sol: 3 input NAND Gate using NMOS:

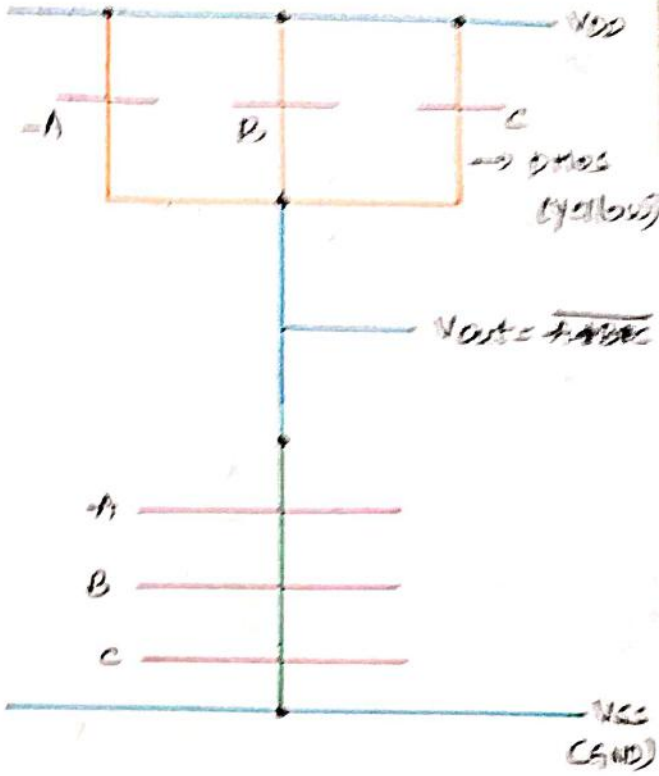




3 input NAND gate using CMOS:



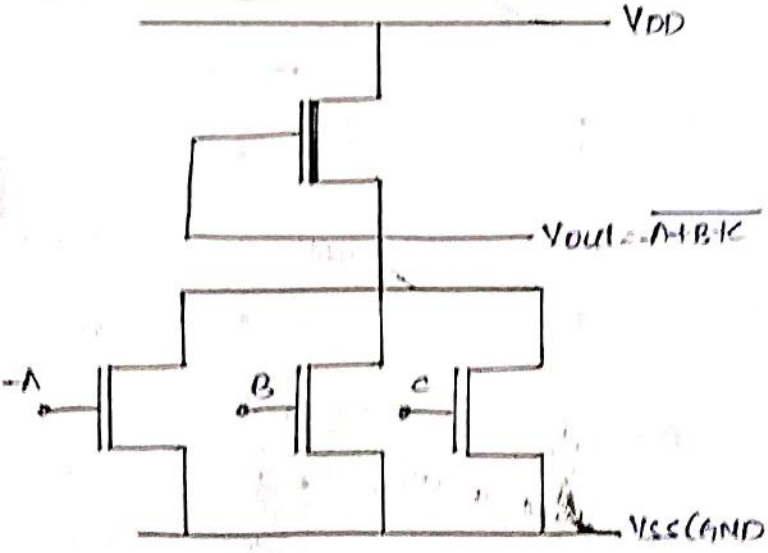
fig(a) ckt diagram of 3 I/p NOR Gate



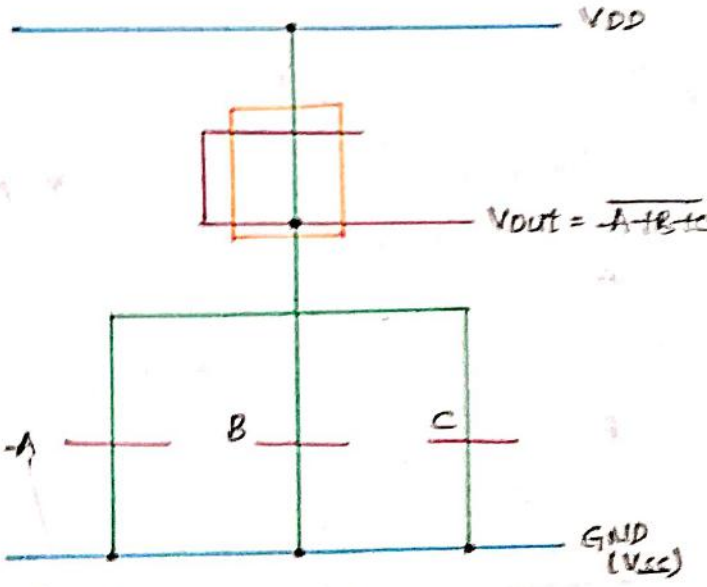
fig(b): Stick diagram

7) Draw the stick diagram for 3 input NOR Gate in N-MOS and CMOS technology:

Sol) For NMOS:

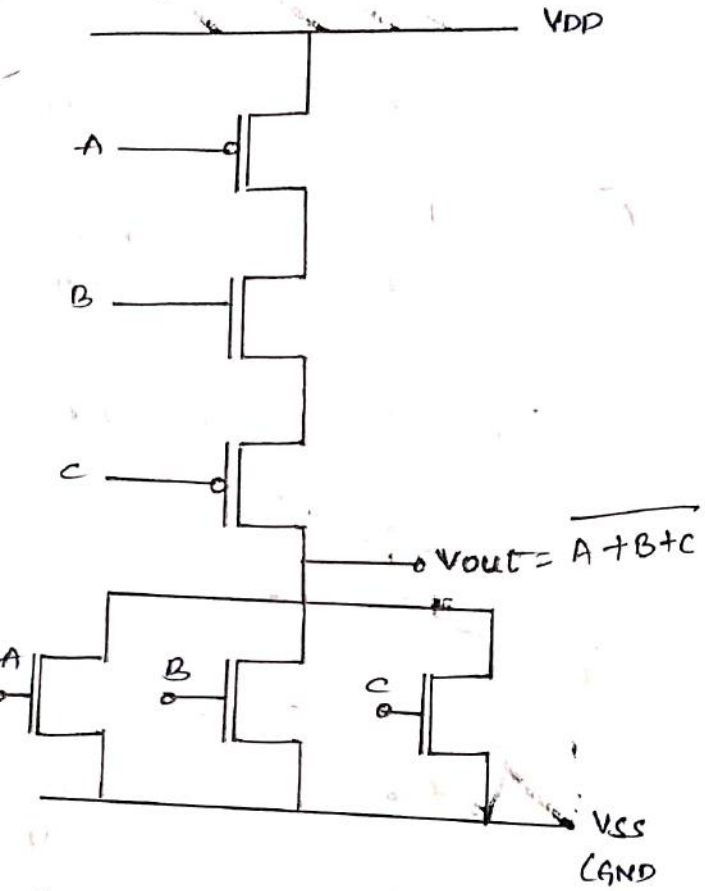


fig(a): ckt diagram of 3 I/p NOR Gate

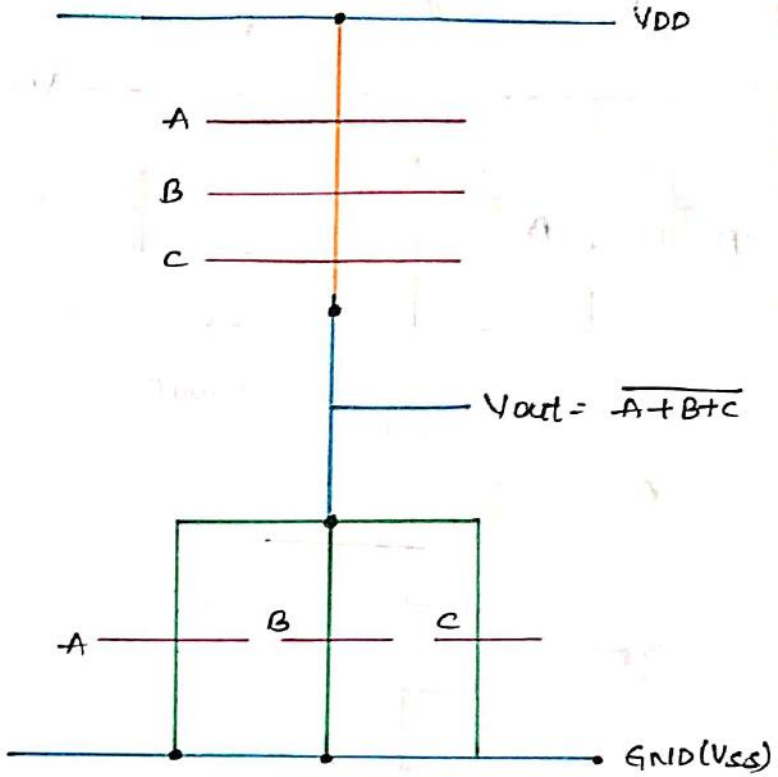


fig(b): Stick diagram

For CMOS:



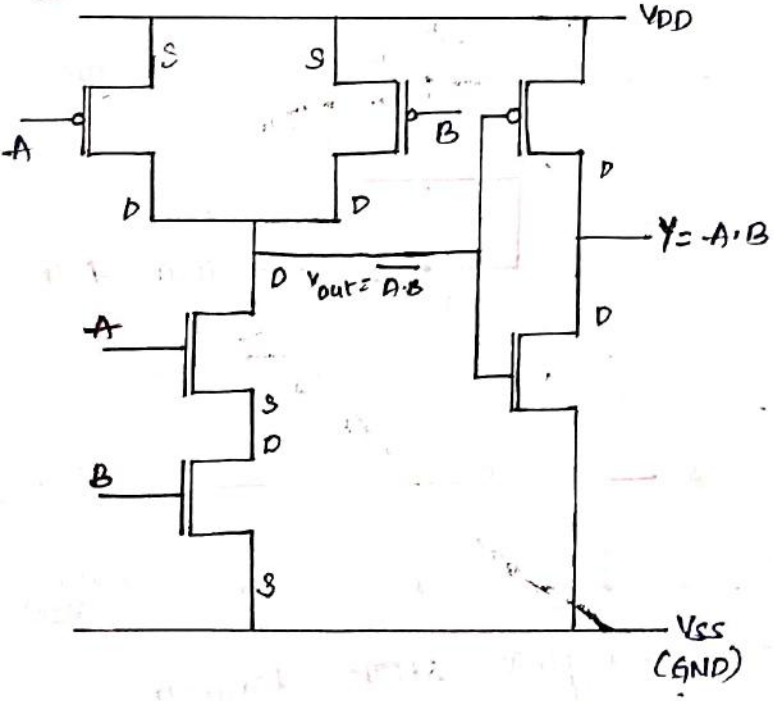
fig(a): 3 i/p NOR gate using CMOS ckt

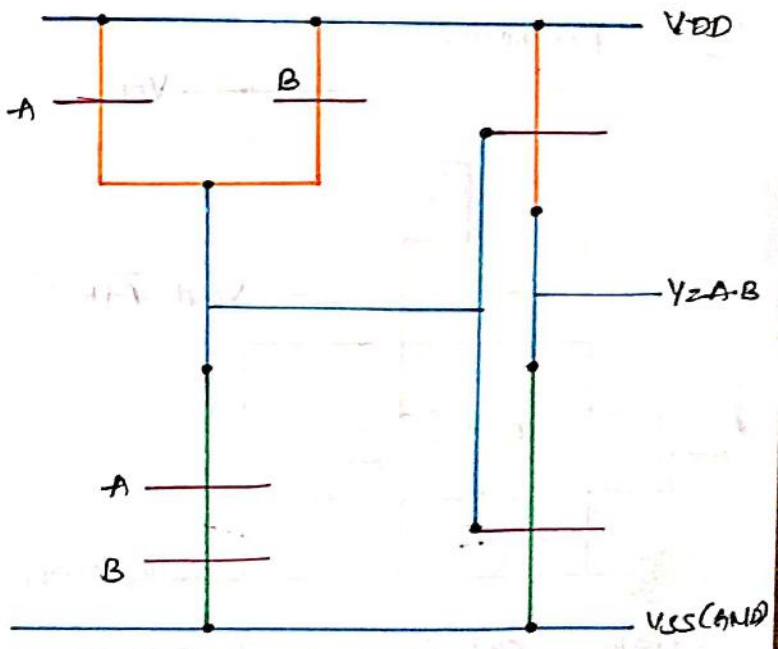


fig(b): Stick diagram for 3 i/p NOR Gate using CMOS

8) Draw the Stick diagram for 2 input AND gate for CMOS?

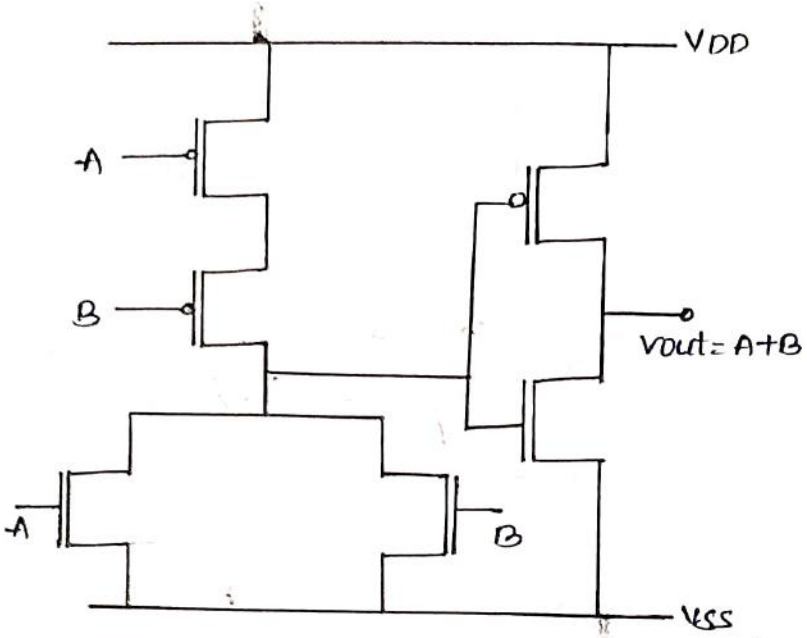
Sol:





9) Draw the stick diagram for 2 input OR Gate for CMOS?

Sol:



fig(a): ckt diagram 2 i/p OR Gate for CMOS

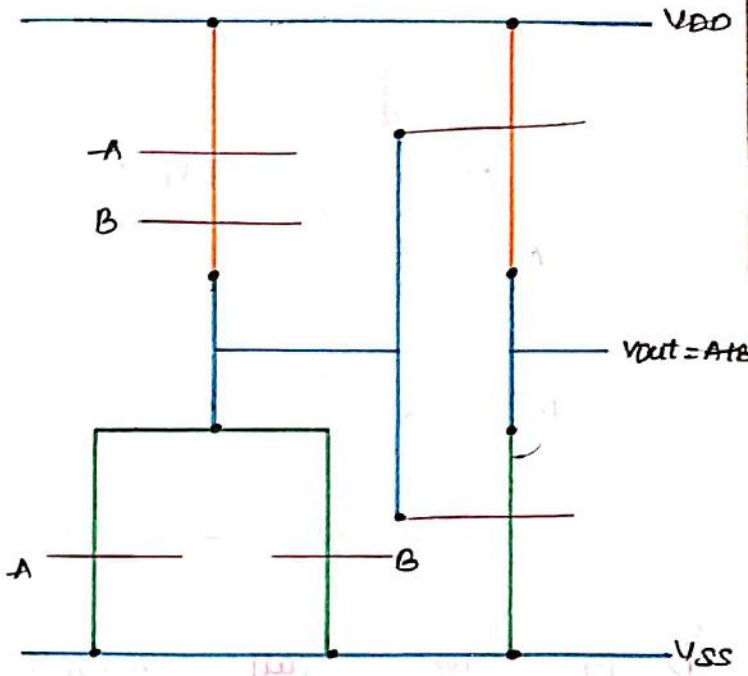


fig (b): Stick diagram

10) Draw the stick diagram for 2 input NAND & NOR gates using PMOS Technology?

Sol: 2 input NAND gate of PMOS:

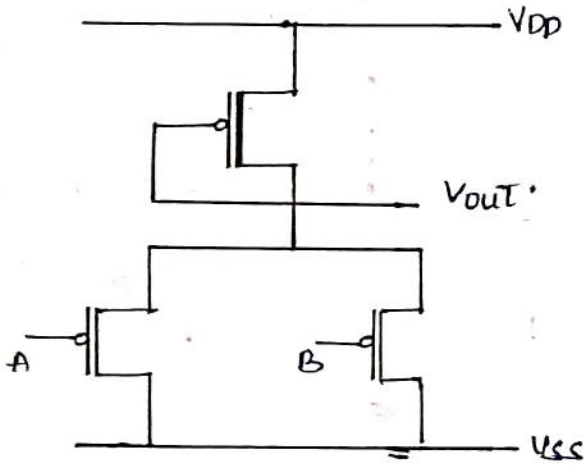


fig (a): ckt diagram

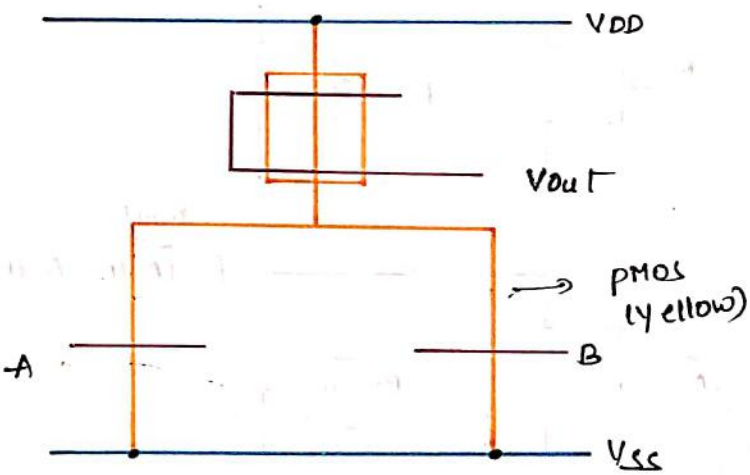
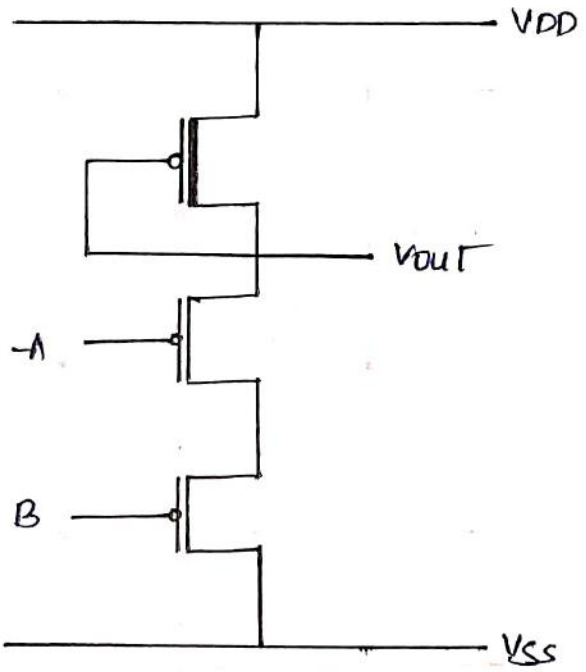


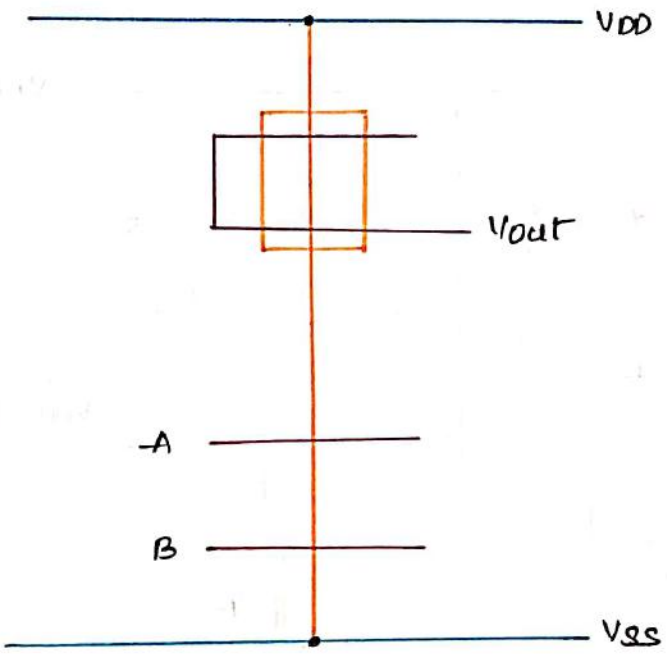
fig (b): Stick diagram

P-MOS - Yellow polysilicon - Red

Metal 1 - Blue Implant - Yellow

2 i/p NOR gate of PMOS:

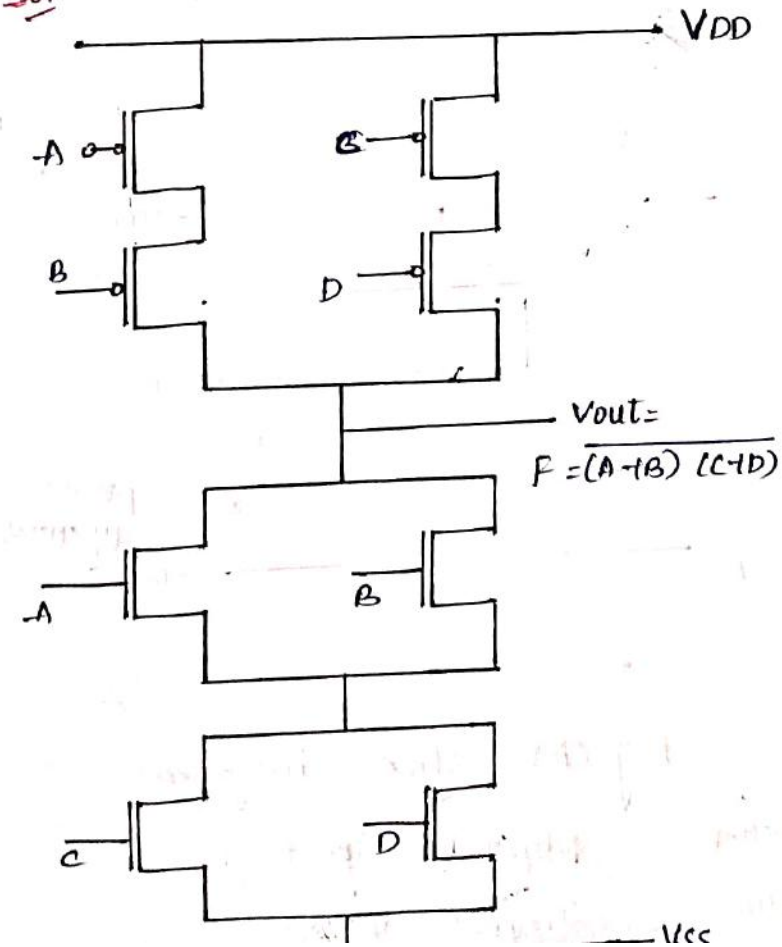


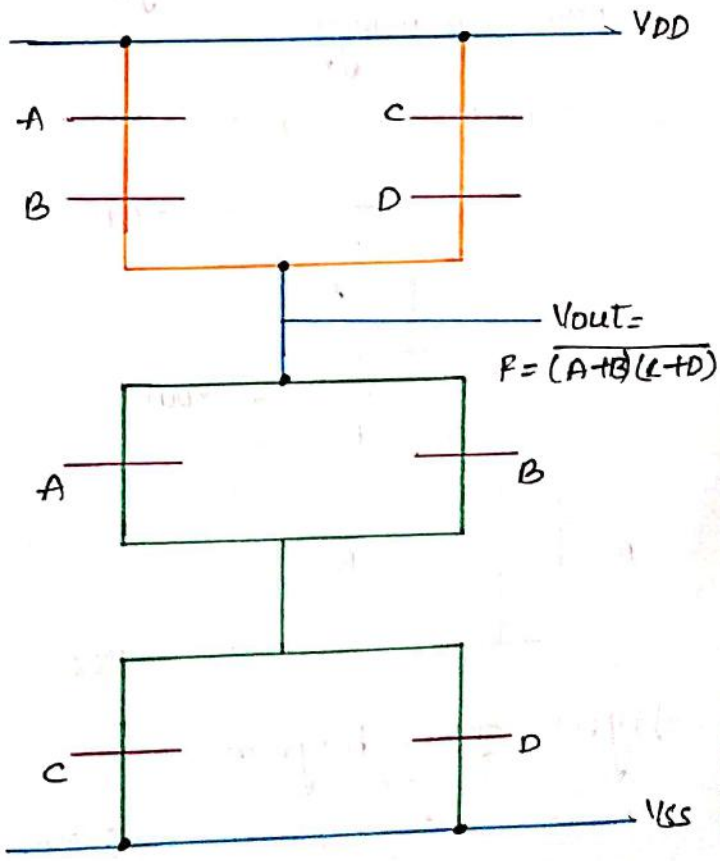


11) Draw the circuit diagram and stick diagram for the following expression in CMOS Technology

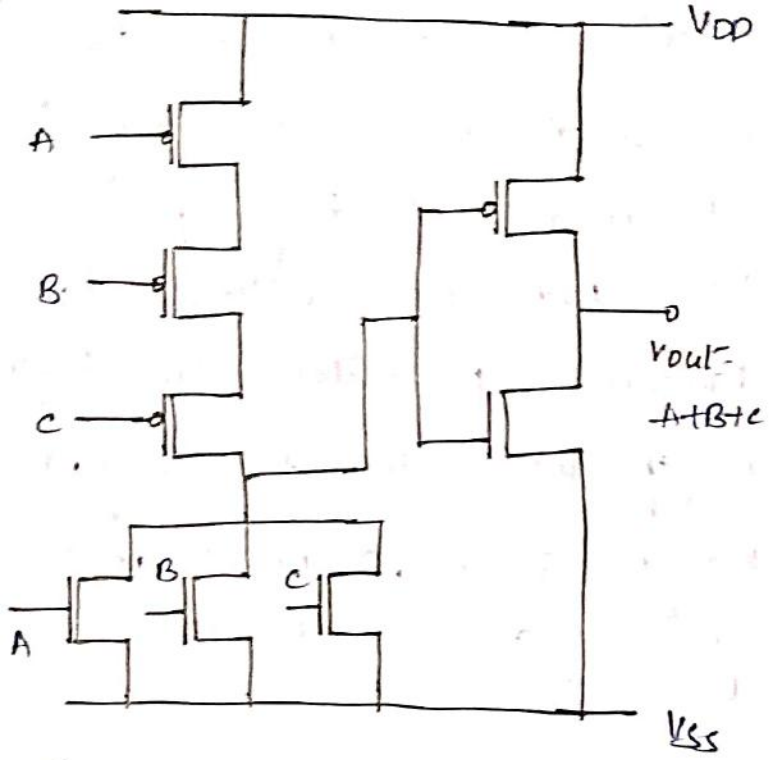
i) $F = \overline{(A+B)(C+D)}$

Sol:

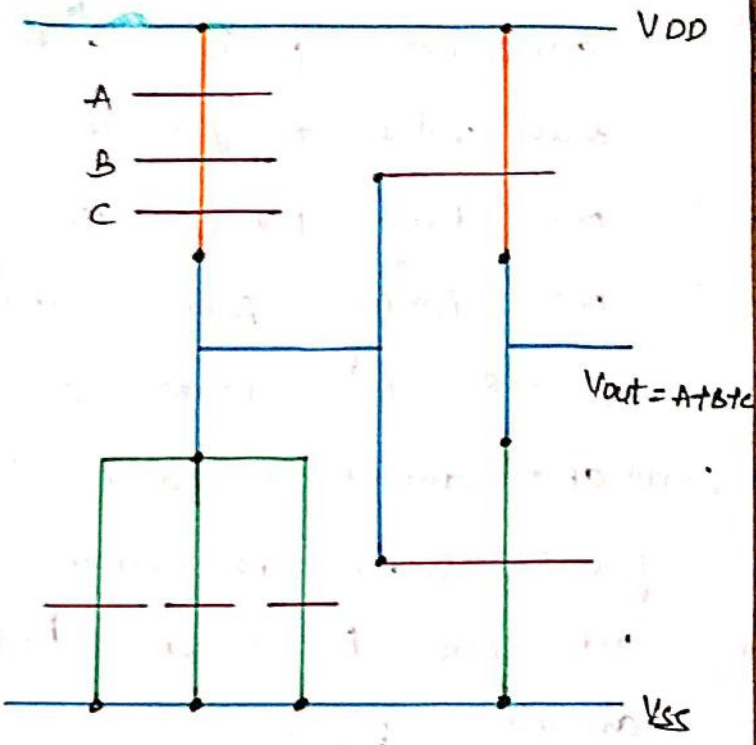




ii) $F = A+B+C$



fig(a): Ckt diagram for 3 i/p CMOS OR Gate using.



fig(b): Stick diagram for CMOS OR Gate.

Design Rules and Layouts:

* Design rules allow the translation of circuit design concepts, usually in stick diagrams (or) symbolic form in to actual Geometry in silicon.

* Layout design is a schematic of the integrated ckt (IC) which describes the exact placement of the components for fabrication.

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

* Layout design rules are used generally specified for Industrial design that are generally specified in microns in terms of micrometers.

* This makes migrating from one process to more advanced process. and difficult because. not all the scale rules are not the same it is if we are going for one design process the next model of design process not having same similarities. both are different process in terms of different parameters like length & width and dimensions and technologies where we are using. In general cases is moving through 360nm, 250nm, 180 nm, 150nm and 90nm based on technolog & scalable design rules are also changes.

* In order to bring unifirmity, Mead & conway populaised lambda - based design rules based on single parameter.

* Lambda characterises the resolution of the process. & is generally half of minimum channel length

* The channel length is the distance b/n drain & source which is set by a minimum width of polysilicon wire.

* When the channel length increases the size of the transistrs increases or viceversa.

* the potential density advantage of micron rules is sacrificed for simplicity and scalability of lambda rules.

* The transistor dimensions are always specified by width / length ratio.

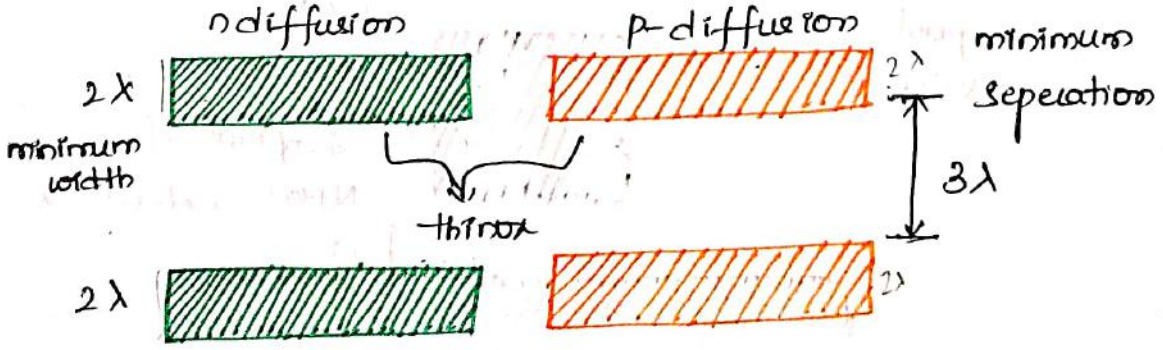
* In digital S/m's, the transistors are chosen to have minimum possible length because short channel Transistors, the size reduces and speed should be faster & consume less power.

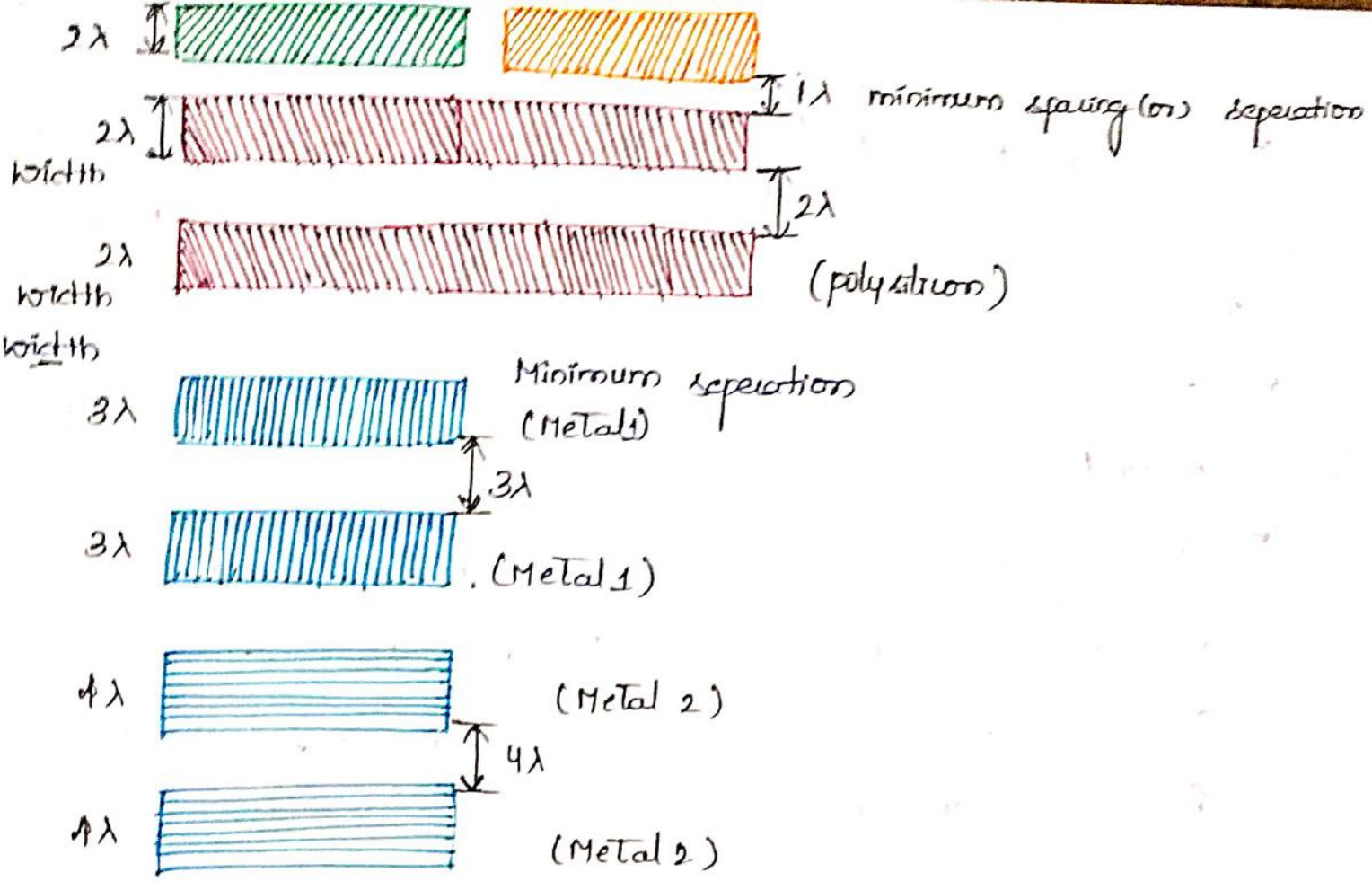
There are three different design rules we have. they are.

1) Lambda based design rules (or) scalable design rules
2) 2μm design rules
3) 1.2μm design rules.

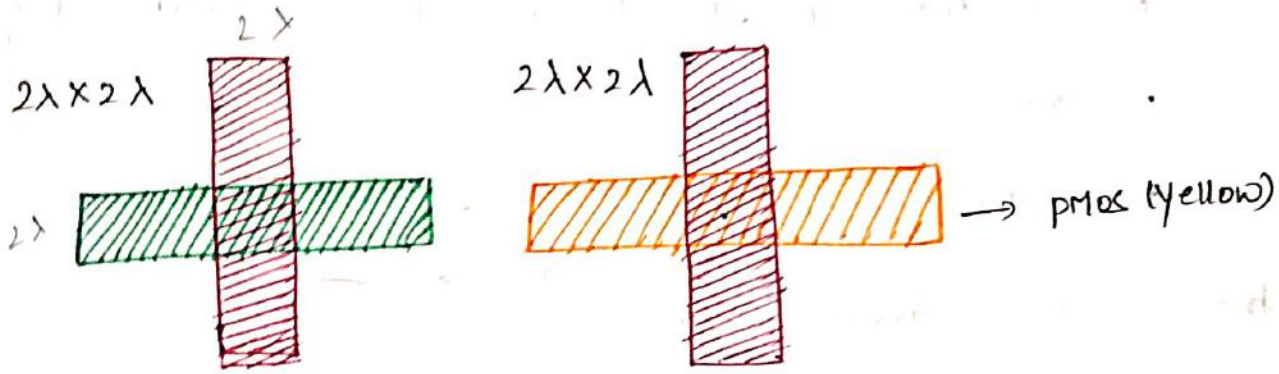
Lambda based design rule:

Lambda based design rule λ show a particular value with feature size and λ allocated a value.



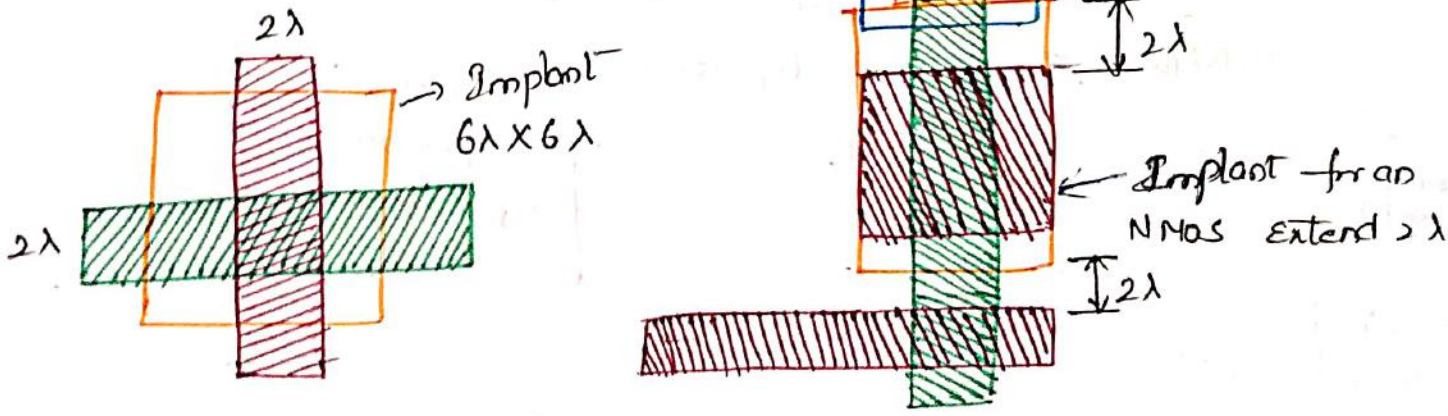


Lambda rules for NMOS:

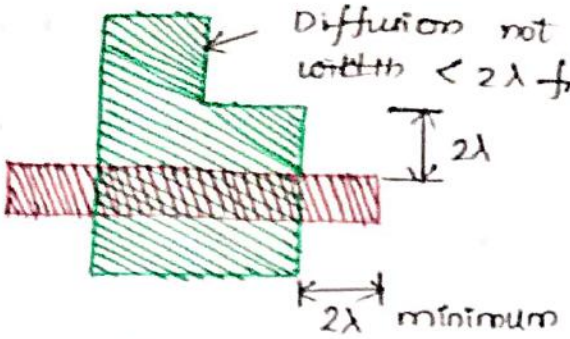


nmos Enhancement

pMos Enhancement



NMOS depletion



* In this approach, all rules are defined in terms of single parameter λ. The rules are so chosen that a design can be easily ported over a cross-section of Industrial process, making the layout portable.
* When devices shrink, layouts need not be completely redesigned. All features can be measured in integral multiples of 'λ'.
* By choosing values of λ, all dimensions set at a scalable layout.

Summary:

1) Metal 1: minimum width = 3λ, minimum Seperation = 3λ
2) Metal 2: minimum width = 4λ, minimum Seperation = 4λ
3) polysilicon: minimum width = 2λ, minimum poly to polysilicon Seperation = 2λ
4) P & n diffusions: min width = 2λ, min Seperation b/n same diffusions = 2λ
5) Tubs: 10λ wide

Minimum Seperation b/n tub & Source/drain = 5λ

Tub-tie: p-tub-tie: 2λ×2λ cut, 4λ×4λ metal, 4λ×4λ p^+ diffusion.

the same for n-tub-tie

construction rules of Transistor (or) Transistor related design rules:

1) Transistors : width = 2λ
length = 2λ

2) poly silicon extends 2λ beyond active region.

3) Diffusion extends 2λ

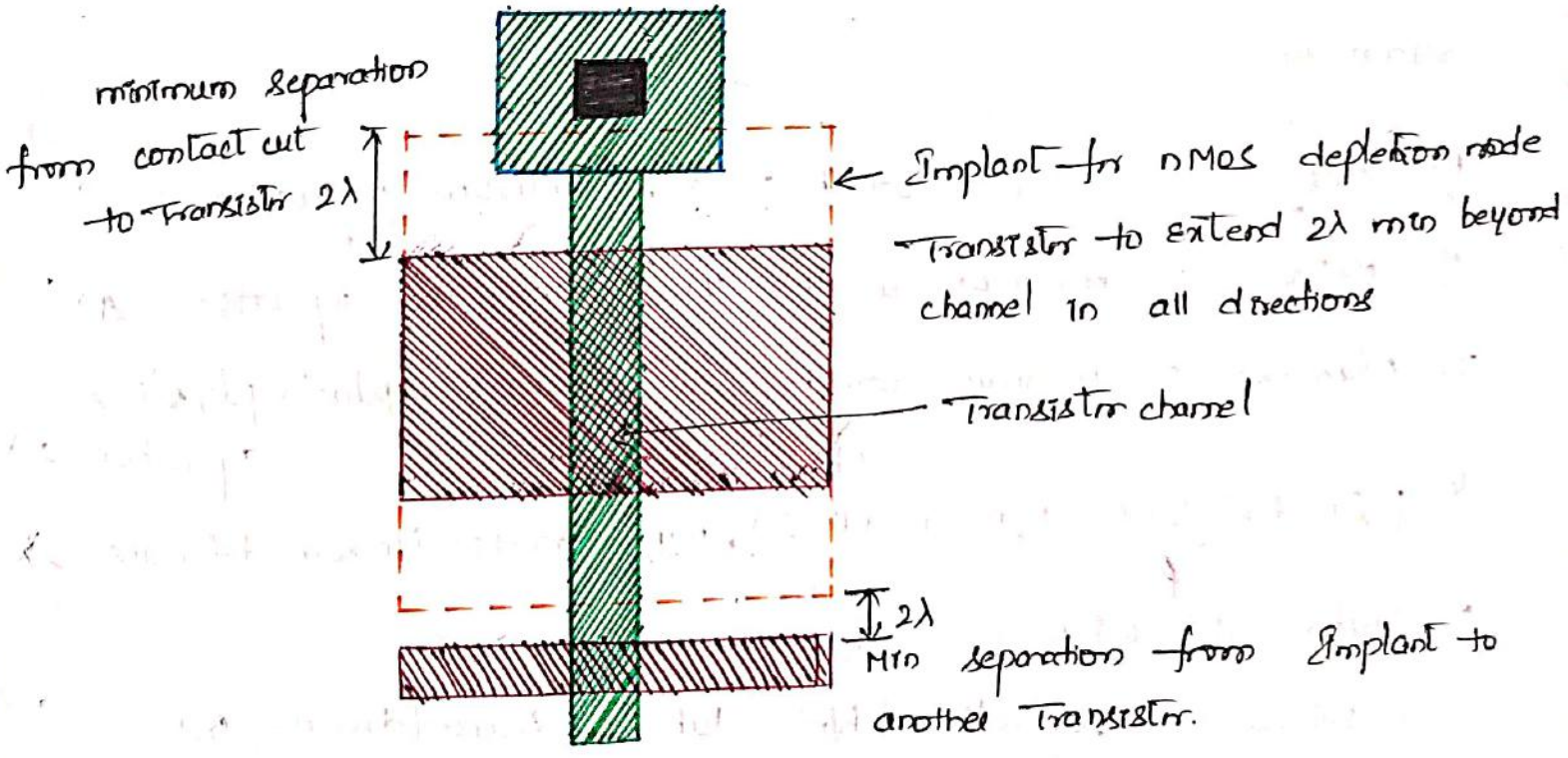
4) Active region must be atleast 1λ from poly-metal via 2λ from another Transistor.

Vias:

* Material on both layers to be connected must extend beyond SiO_2 cut and cut must be 2λ in all directions.

* cuts: $2\lambda \times 2\lambda$

* via size = $4\lambda \times 4\lambda$



Extensions & Seperations of Transistors.

Contact cuts:

3 ways to make contacts between poly & diffusions in NMOS circuits.

i) poly to metal then metal to diffusion.

ii) buried contact (poly to diffusion)

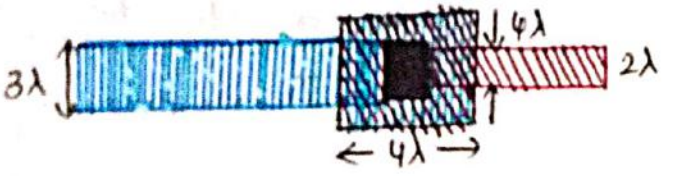
iii) butting contact (poly to diffusion using metal).

* The 2λ × 2λ contact cut indicates an area in which the oxide is to be removed down to the underlying polysilicon or diffusion surface.
* When deposition of metal layer takes place the metal is deposited through contact cuts areas onto underlying area so that contact is made b/n the layers.

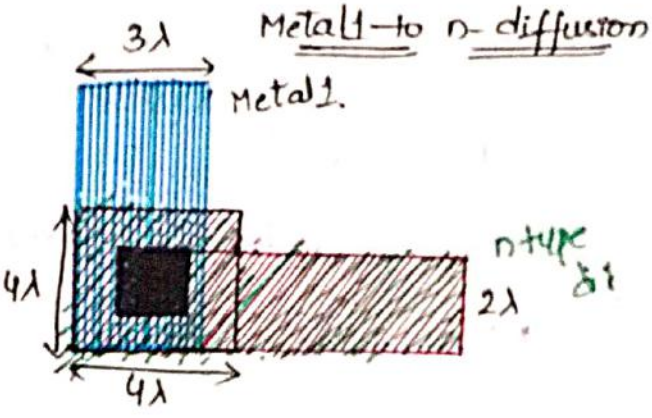
Examples:

1) Metal1 to polysilicon or to diffusion

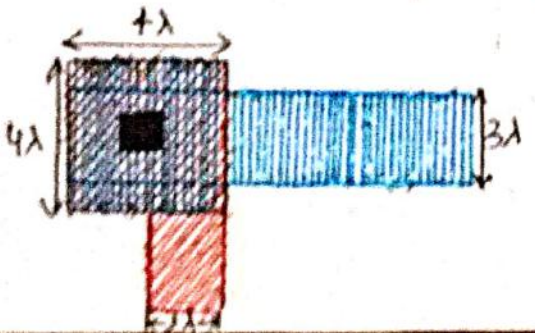
Metal1 to poly

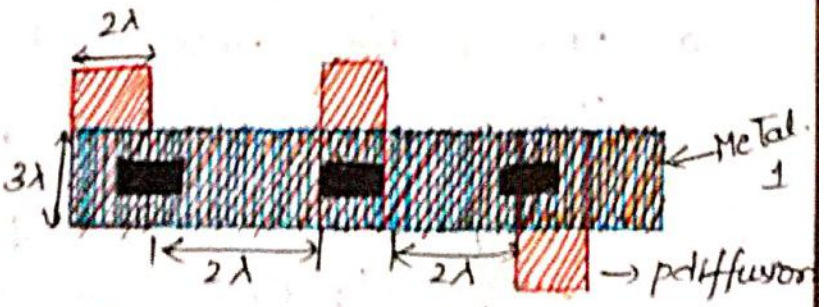


Metal1 to n-diffusion



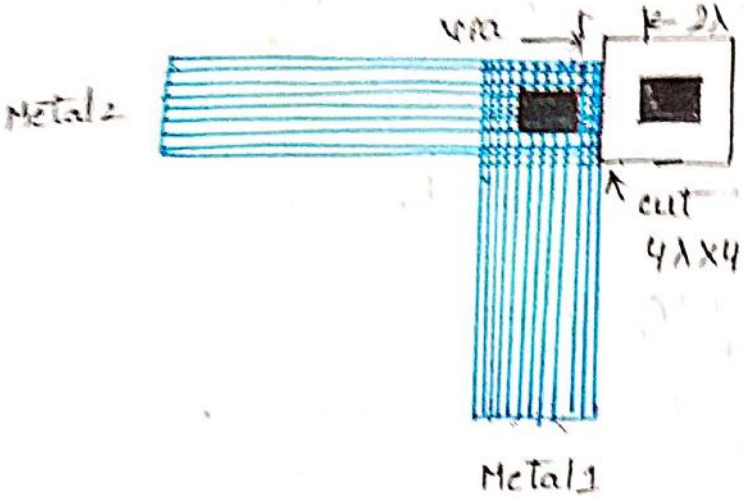
Metal1 to p-diffusion:





Min separation, Multiple cuts

2) via (contact from metal 2 to metal 1)



4λ×4λ area of overlap with 2λ×2λ via at center.

* contact cuts are also known as via cuts.
* 4λ×4λ size.
* contact cut types:
  - i) n/p diffusions to polysilicon
  - ii) polysilicon to Metal
  - iii) n/p diffusions to metal 1
  - iv) metal 1 to metal 2

CMOS Lambda based design rules:

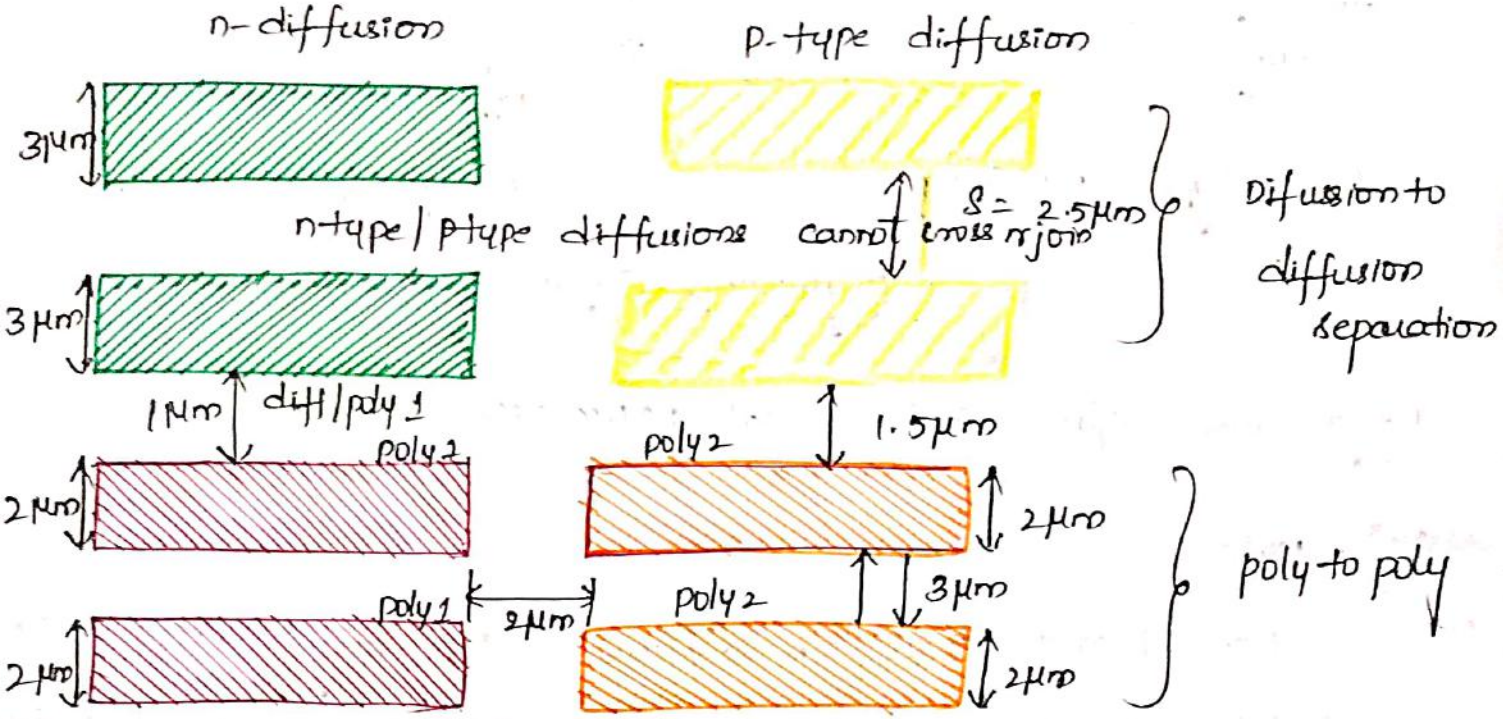
The rules of n-well (pMOS Transistor), p-wires and special substrate contacts are added to the existing nMOS rules.

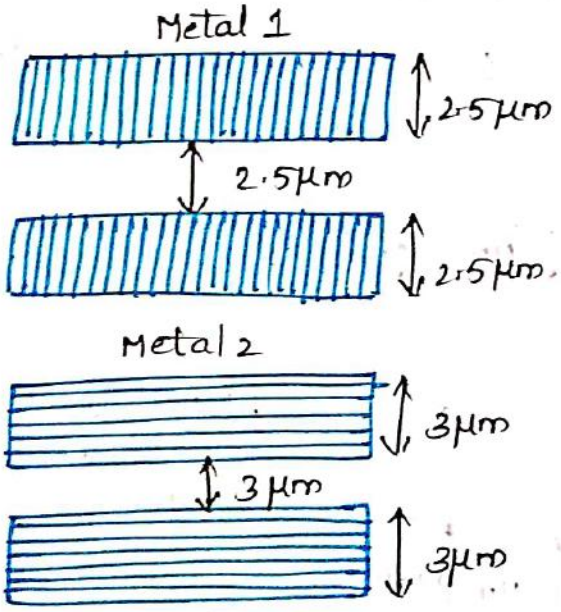
2μm CMOS Design rules:

* 2μm double metal, double poly cMOS rules

→ n-well : brown
poly 1 : red
poly 2 : Green
n-diffusion : Green
p-diffusion : yellow
} for CMOS

buried n^+ sub collector : pale green
p-base : pink
} BICMOS





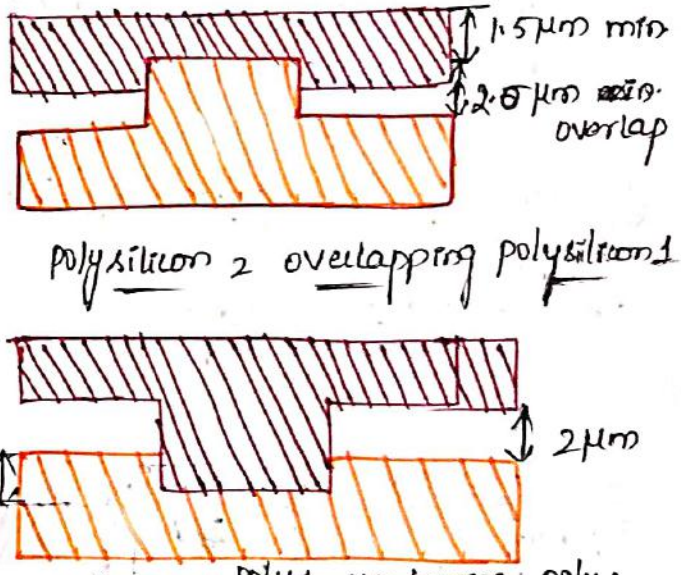
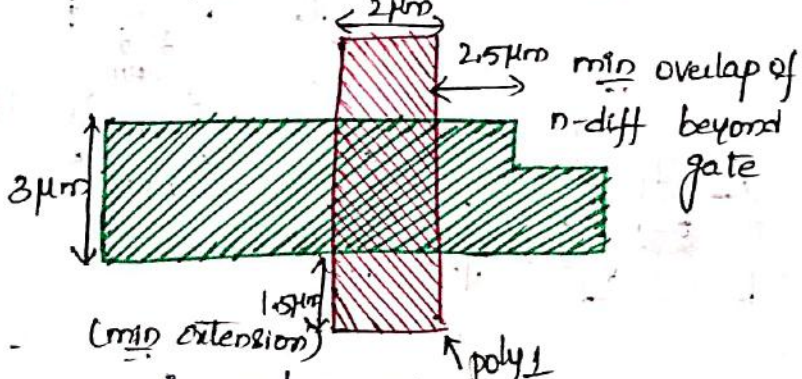


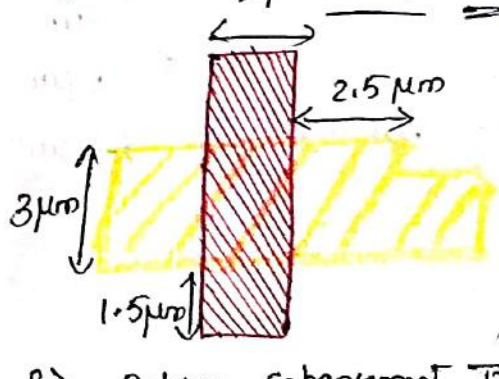
fig: Design rules for wires (2μm CMOS figures)

Design rules for transistors (CMOS):

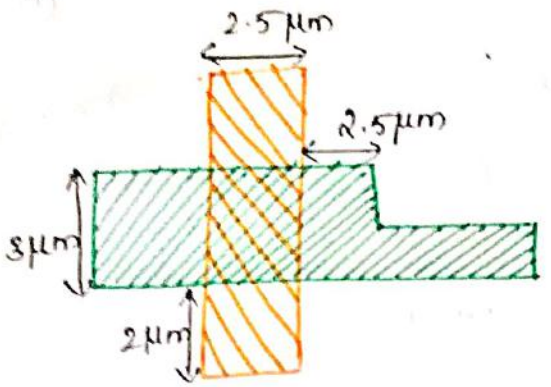
poly 1 overlapping poly 2 for polysilicon 1 transistors



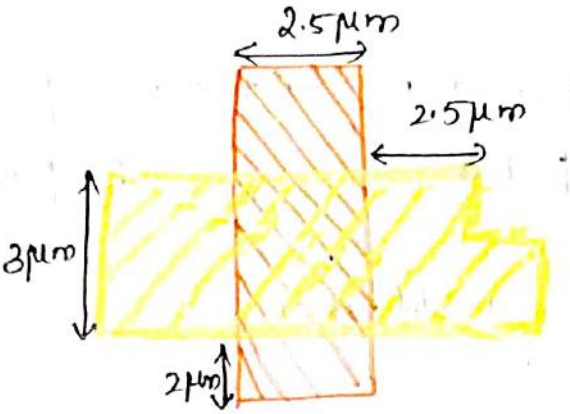
1> n-type enhancement



2> p-type enhancement Tran



(i) n-type transistor



(ii) p-type transistor

fig : polysilicon 2 Transistors

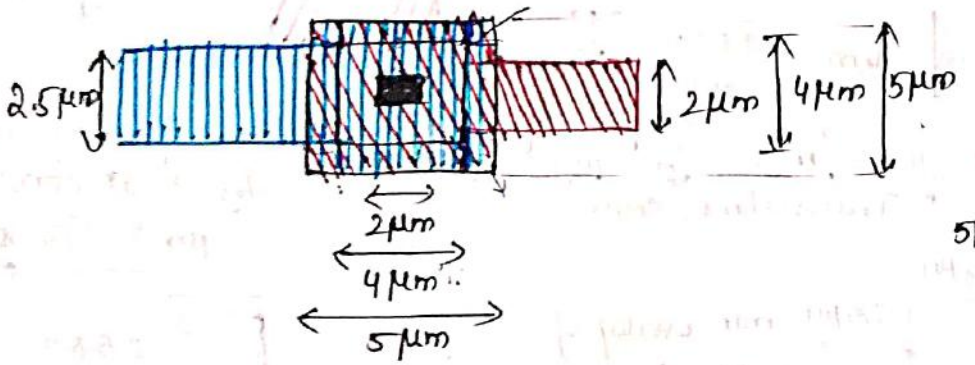
Design rules for contact cuts (2μm CMOS):

(a) Note:

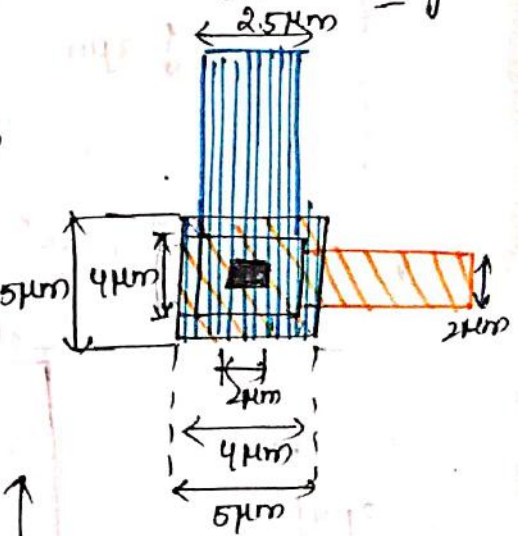
For p-well CMOS, n-diffusion can only exist inside & p-diffusion wires outside p-well. For n-well CMOS, p-diffusion wires can only exist inside & n-diff wires outside n-well.

Design Rules for contact cuts (2μm CMOS):

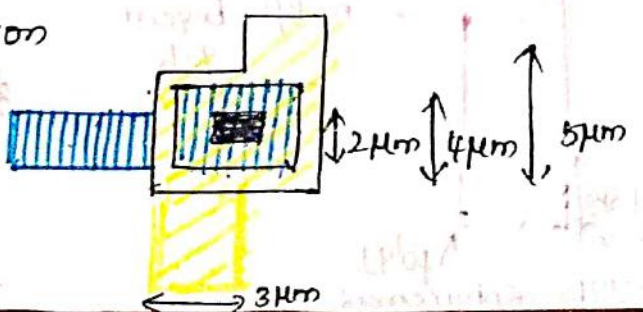
(a) Metal 1 to poly 1



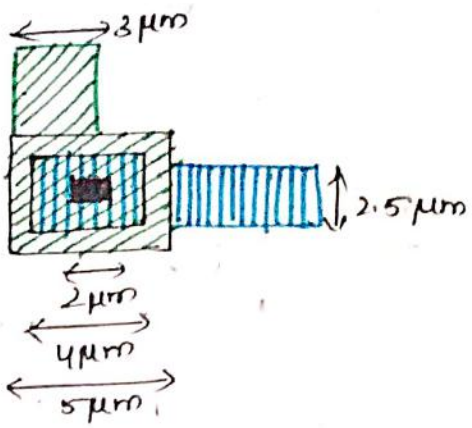
(b) Metal 1 to poly 2



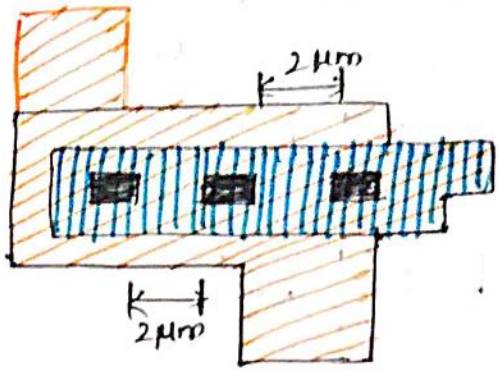
(c) Metal 1 to p diffusion



(d) Metal 1 to n diff

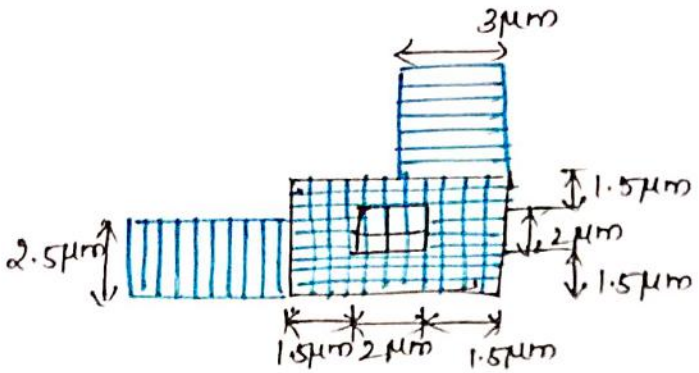


(e) Multiple contact cuts

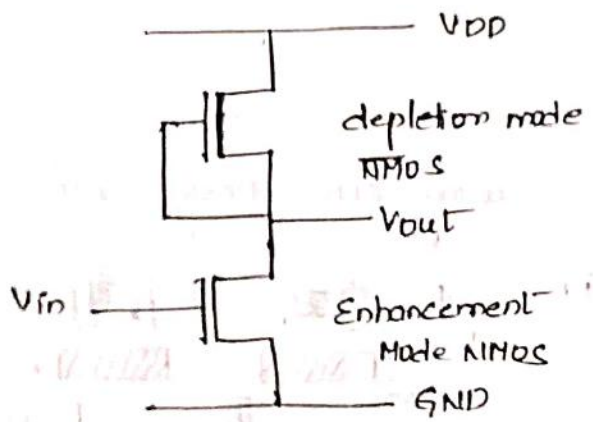


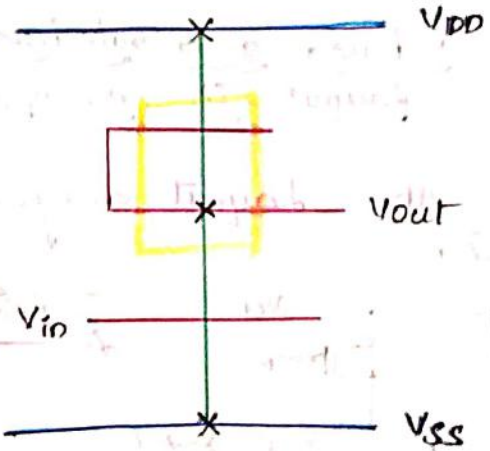
minimum spacing b/n contact cuts = 2μm

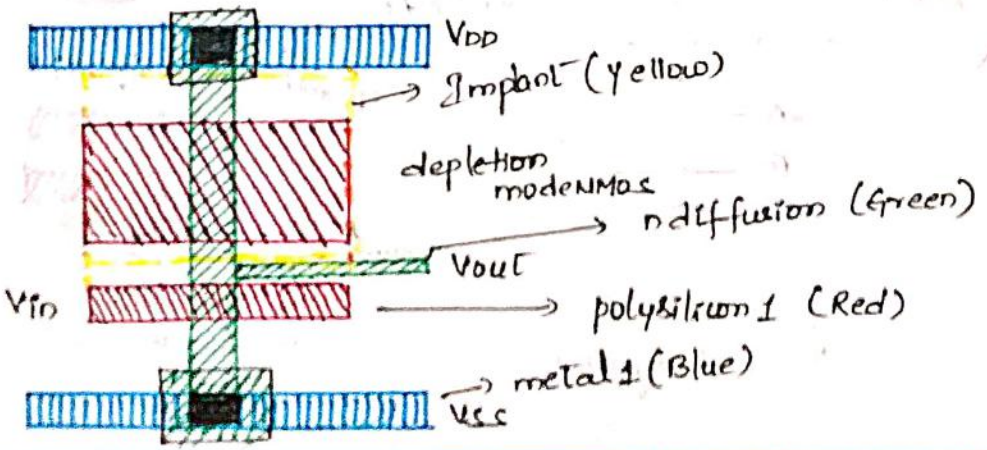
(f) Via metal_1 / metal_2



NMOS Layout diagrams for NMOS Inverter gate:







Layout diagram for CMOS Inverter:

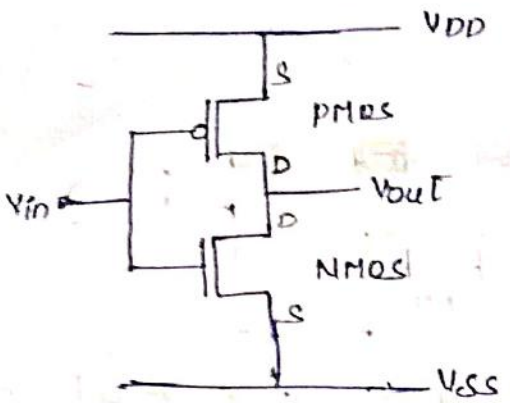


fig: CKT diagram for CMOS Inverter

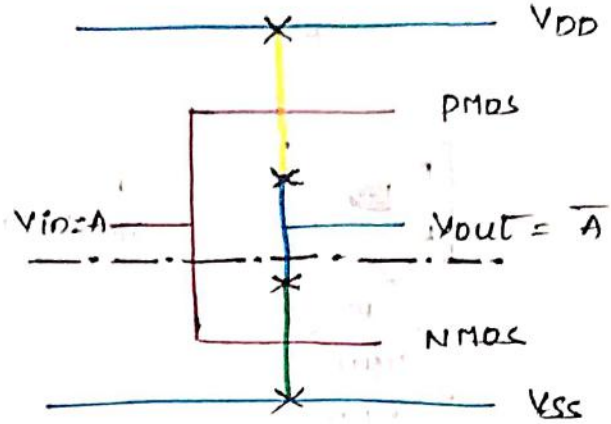


fig: stick diagram for CMOS Inverter.

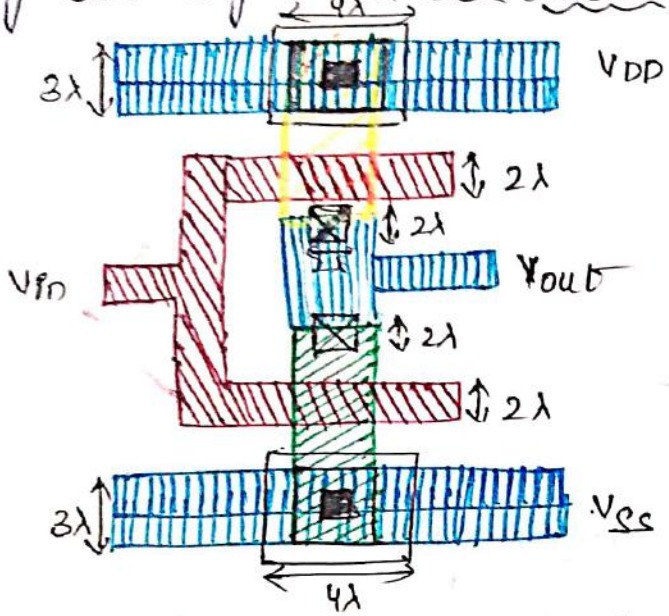


fig: λ based design rule based Layout for CMOS Inverter

1) Draw the Layout diagram of NAND-Gate using CMOS Design rule.

Sol: $Y = \overline{A \cdot B}$

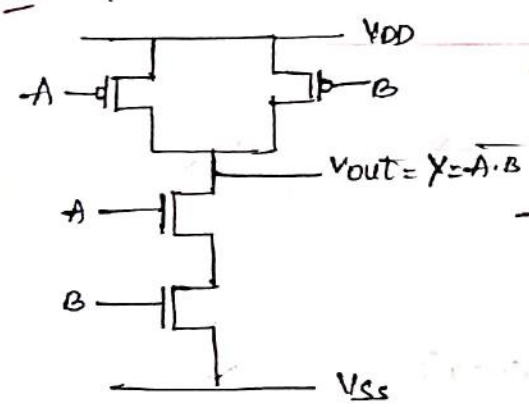


fig: schematic(or) CKT diagram

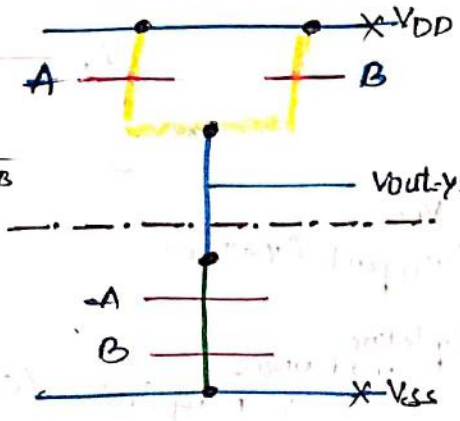


fig: stick diagram

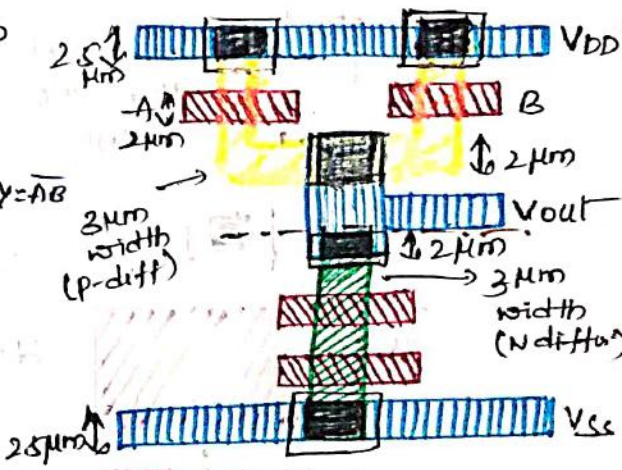
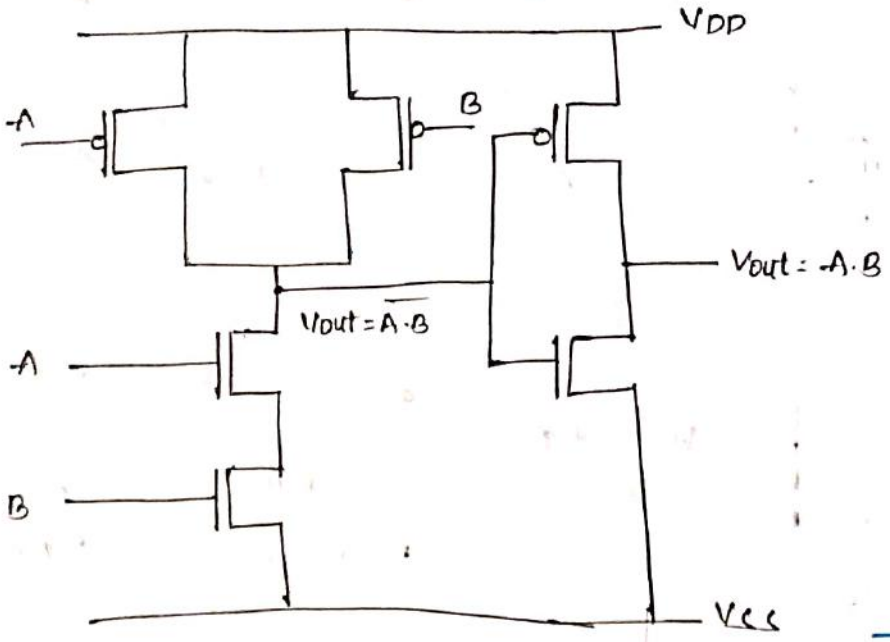
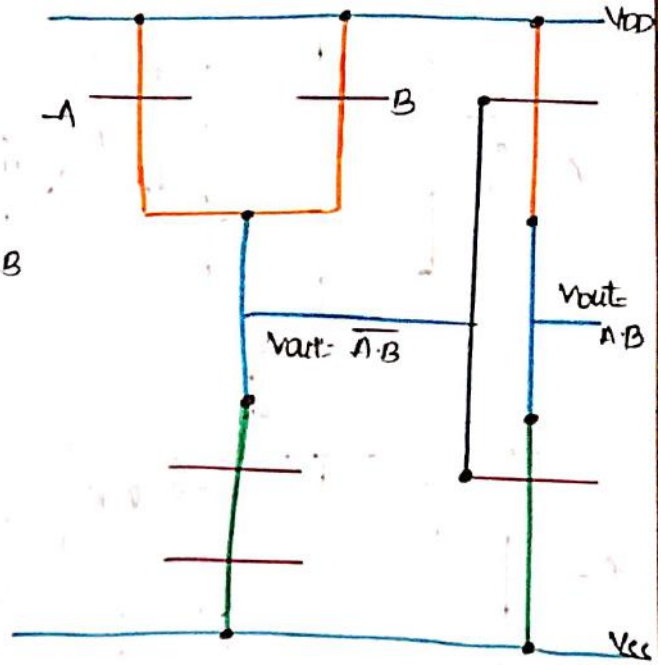


fig: Layout diagram

Layout diagram for CMOS AND gate:



fig(a): ckt diagram for 2ip CMOS AND Gate



fig(b): Stick diagram for CMOS 2ipAND gate

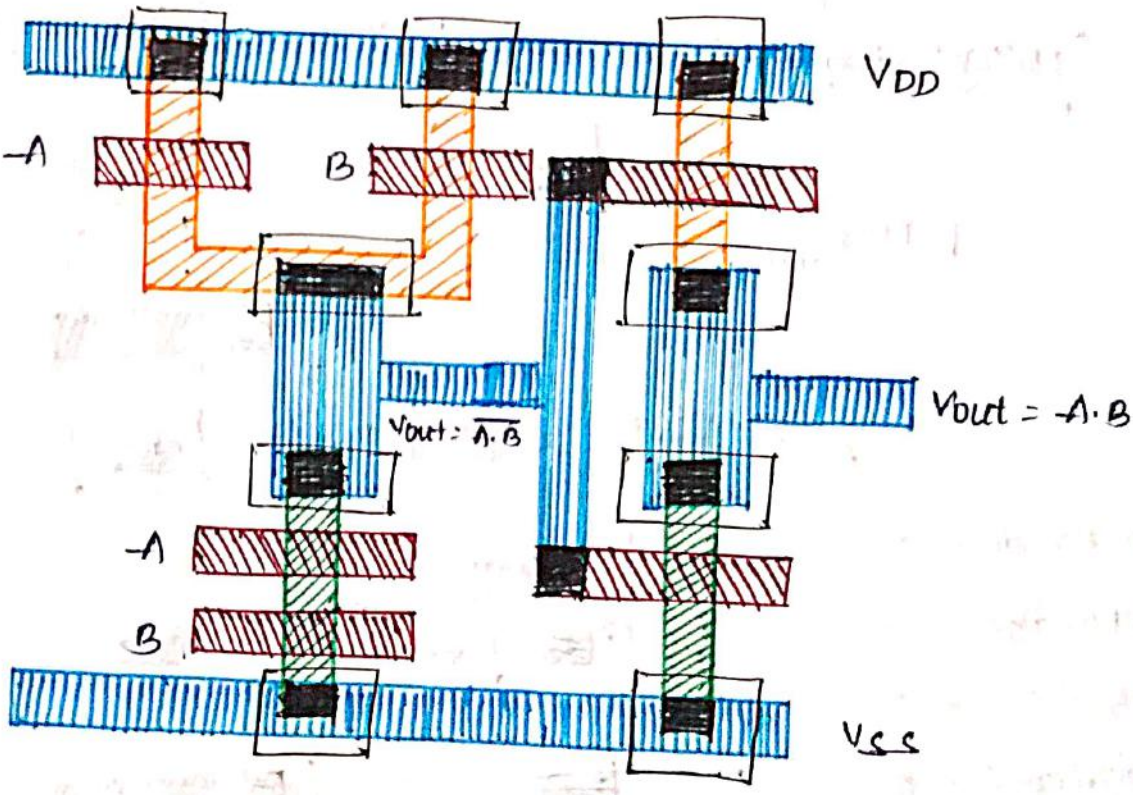
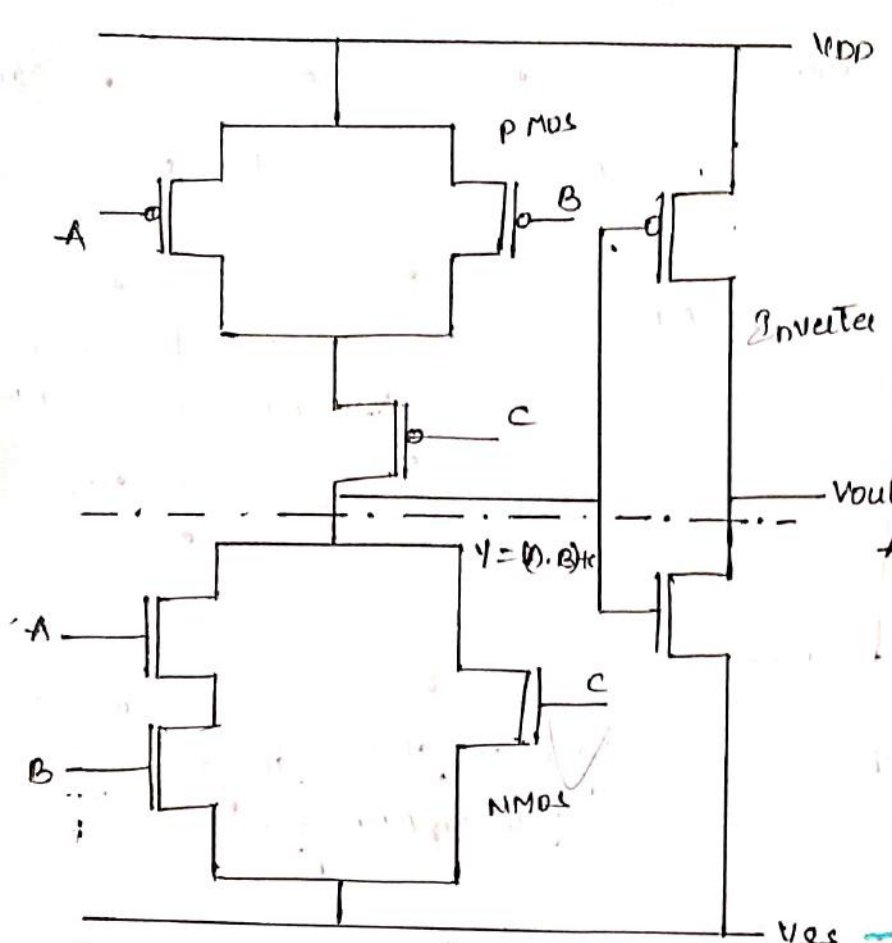
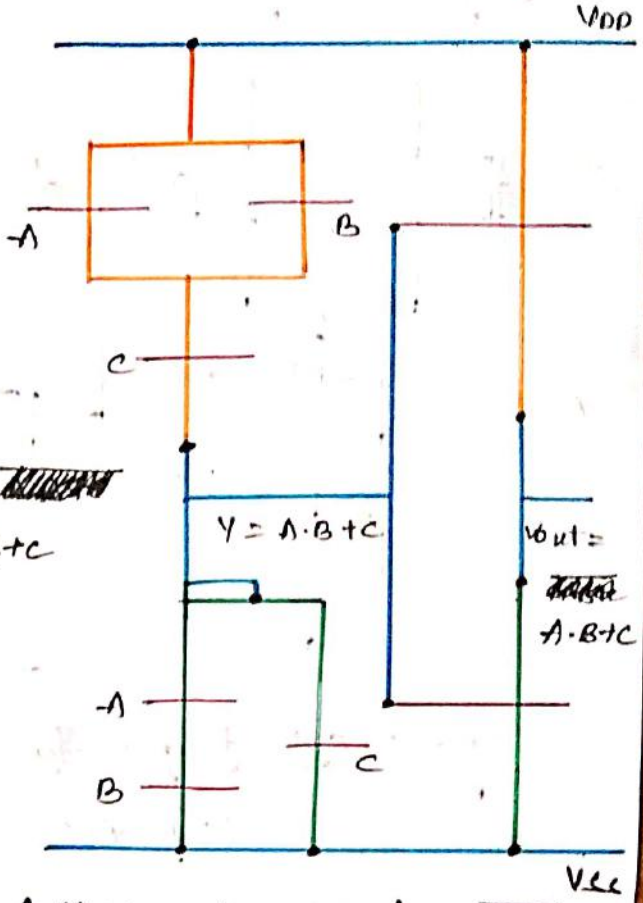


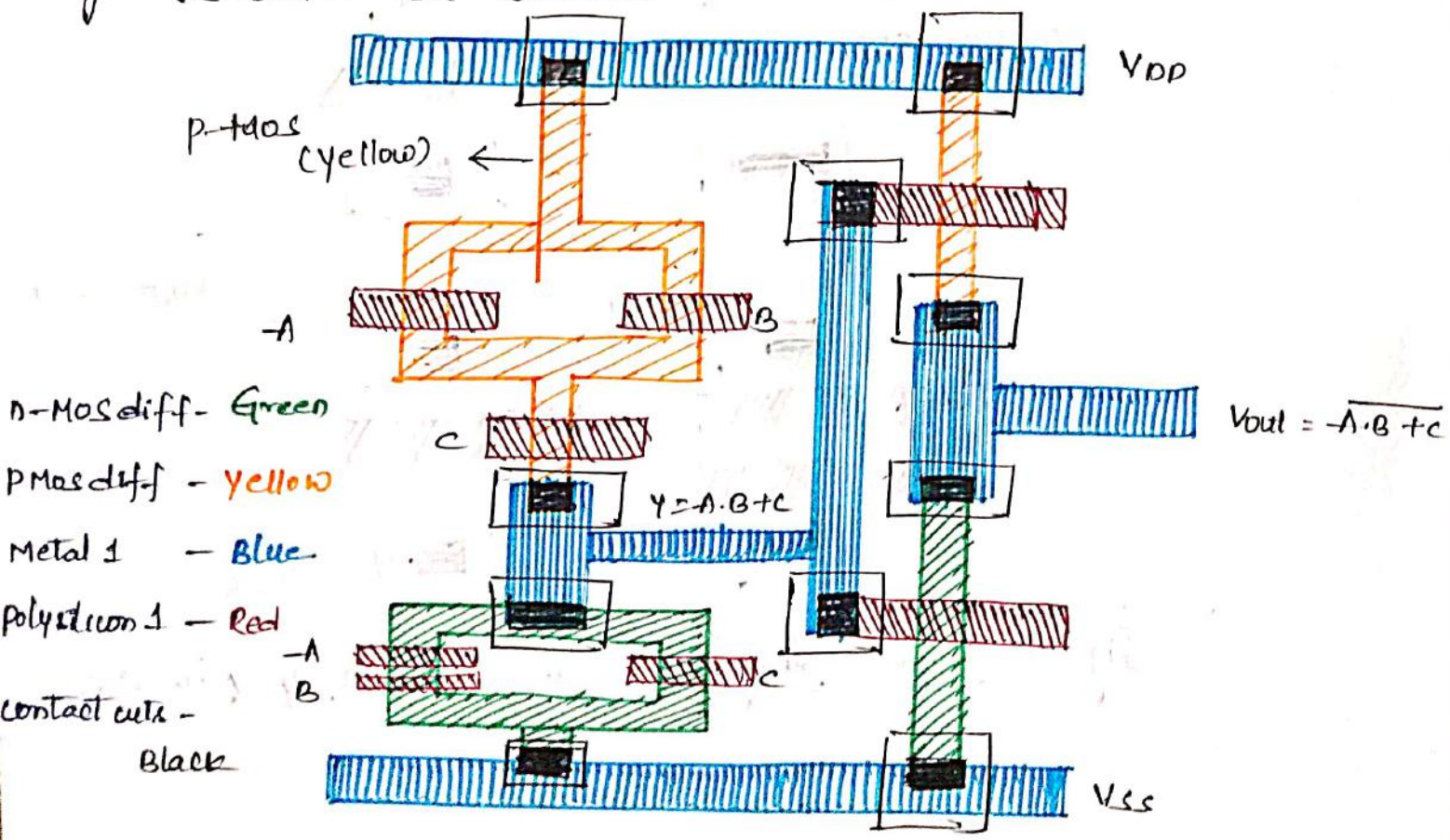
fig c: Layout diagram for CMOS 2ip AND Gate

1) Draw the stick diagram and layout diagram for $Y = \overline{(AB+C)}$





fig(a): Ckt diagram for $Y = \overline{A \cdot B + C}$ — fig(b): stick diagram for $Y = \overline{AB+C}$



fig(c): Layout diagram for $Y = \overline{A \cdot B + C}$

Scaling of MOS circuits:

* Adjustment of the dimensions of an electronic device while maintaining the electrical properties of the device either larger or smaller than the unscaled device.

* Scale the devices and wires down, make the chips faster, functionality, intelligence memory and faster, make more chips per wafer increased yield, make the end user happy giving more or less. & the cost will be low.

* Impact of scaling is characterized in terms of several Indicators.

1) minimum feature size
2) Number of gates on one chip.
3) power dissipation.
4) maximum operational frequency.
5) Disc size.
6) production cost.

* This can be Improved by shrinking the dimensions of transistors and interconnections. shrinking the separation b/n Transistors and wires. Adjusting doping levels and supply voltages.

Technology scaling:

* reduce the gate delay by 30%. (increase operating frequency by 43%.) double transistor density.

* Reduce Energy per transition by 65%. (50% power saving at 43% increase in frequency), Die size used to increase by 14% per generation.

* The technology generation spans 2-3 years.

Scaling models:

* Ideal model - dimensions and voltage scale together by the same scale factor fixed voltage scaling.

* most common model until recently - only the dimensions scale, voltage remains constant called General scaling

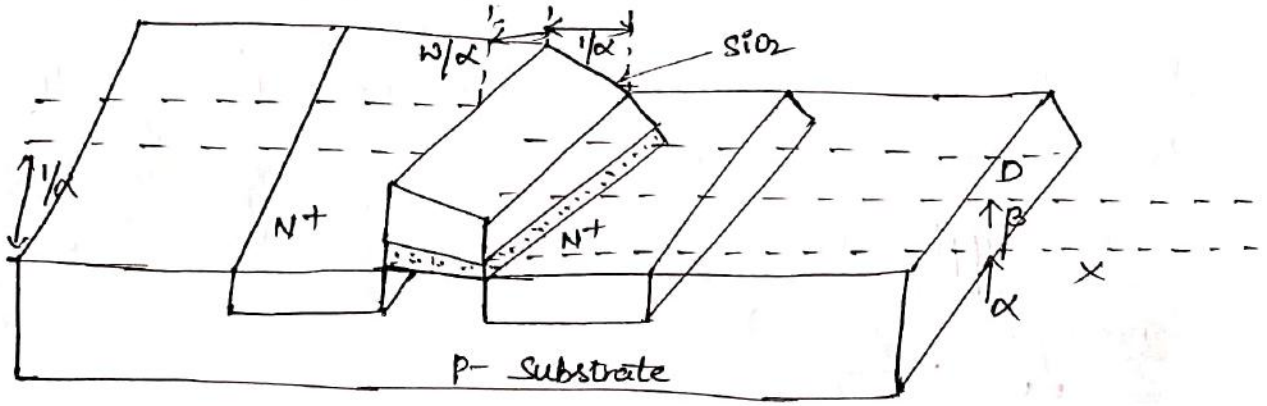
* Most realistic today's situation - voltages and dimensions scale with different factors.

Scaling factors for device parameters:

Device scaling modelled in Terms of generic scaling factors:

1) Scaling factor for supply voltage V_{DD} and Gate oxide thickness D.

2) linear dimensions both horizontal & vertical dimensions.



1) Gate area [Ag]:-

$A_g = W \times L$

width is scaled by $\frac{1}{\alpha}$, length is scaled by $\frac{1}{\alpha}$

$A_g = \frac{1}{\alpha} \cdot \frac{1}{\alpha}$

$A_g = \frac{1}{\alpha^2}$

2) Gate oxide capacitance per unit area C_0 (or) C_{ox}:-

$C_0 = \frac{\varepsilon_{ox}}{D}$

where ε_{ox} = Relative permittivity of the gate oxide, is scaled by '1'

D = Diameter of oxide, is scaled by $\frac{1}{\beta}$

$C_0 = \frac{1}{\frac{1}{\beta}} = \beta$

C_0 (or) C_{ox} is scaled by β

3) Gate capacitance (C_g):-

$C_g = C_0 \cdot W \cdot L$

W.k.T $C_0 = \beta \cdot L = \frac{1}{\alpha}$

$C_g = \beta \cdot \frac{1}{\alpha} \cdot \frac{1}{\alpha}$

$C_g = \frac{\beta}{\alpha^2}$

C_g is scaled by $\frac{\beta}{\alpha^2}$

4) parasitic capacitance (C_x):

* C_x is directly proportional to $\frac{A_x}{d}$

where A_x = area of depletion region around drain is scaled by $\frac{1}{\alpha^2}$

d = depletion width present in source & drained Scaled by $1/\alpha$

$$C_x = \frac{1}{\frac{\alpha^2}{\alpha}} = \frac{1}{\alpha}$$

C_x is scaled by $\frac{1}{\alpha}$

5) carrier density in channel (Q_{ON}):

$$Q_{ON} = C_o . V_{gs}$$

$$= \beta . \frac{1}{\beta}$$

$$Q_{ON} = 1$$

Q_{ON} is scaled by 1

6) ON channel Resistance (R_{ON}):

$$R_{ON} = \frac{1}{Q_{ON} . \mu} \left(\frac{L}{W}\right)$$

μ is carrier mobility is constant and scaled by '1'.

$$R_{ON} = \frac{1}{|x|} = \frac{1}{\frac{\alpha}{\alpha}} = 1$$

R_{ON} is scaled by '1'.

7) Gate delay (T_d):

T_d is proportional to $R_{ON} C_g$

$$T_d \propto 1 . \frac{\beta}{\alpha^2}$$

$$T_d \propto \frac{\beta}{\alpha^2}$$

8) Maximum operating frequency (f_o):-

$$f_o = \frac{W}{L} \frac{\mu . C_o V_{DD}}{C_g}$$

$$= \frac{\frac{1}{\alpha}}{\frac{1}{\alpha}} . \frac{1 . \beta . \frac{1}{\beta}}{\beta / \alpha^2}$$

$$f_0 = \frac{\alpha^2}{\beta}$$

'f_0' is scaled by $\frac{\alpha^2}{\beta}$

9) Saturation current (I_{dss}):

$$I_{dss} = \frac{C_{OM}}{2} \cdot \frac{W}{L} [V_{gs} - V_t]^2$$

$$= \frac{\beta \cdot 1}{2} \cdot \frac{\frac{1}{\alpha}}{\frac{1}{\alpha}} \left[\frac{1}{\beta}\right]^2$$

$$I_{dss} = \frac{\beta}{2} \cdot \frac{1}{\beta^2}$$

$$I_{dss} = \frac{1}{\beta}$$ ['2' constant is scaled by '1']

$\therefore$ I_{dss} is scaled by "$\frac{1}{\beta}$"

10) Current density (J):

$$J = \frac{I_{dss}}{A} = \frac{\frac{1}{\beta}}{1/\alpha^2} = \frac{\alpha^2}{\beta}$$

J is scaled by $\frac{\alpha^2}{\beta}$

11) Switching Energy per gate (E_g):

$$E_g = \frac{C_g}{2} \cdot V_{dd}^2$$

$$= \frac{\beta}{\frac{\alpha^2}{1}} \cdot \frac{1}{\beta^2}$$

$$E_g = \frac{1}{\alpha^2 \beta}$$

'E_g' is scaled by $\frac{1}{\alpha^2 \beta}$

12) Power dissipation per gate (P_g):

$$P_g = P_{gs} + P_{ds}$$

P_{gs} = static component

$$= \frac{V_{pp}^2}{R_{ON}}$$

$$= \frac{1}{\beta^2}$$

$$P_{gs} = \frac{1}{\beta^2}$$

P_{ds} = dynamic component

$P_{ds} \Rightarrow E_g \cdot f_o = \frac{1}{\alpha^2\beta} \cdot \frac{\alpha^2}{\beta}$

$P_g =$

$$P_{ds} = E_g \cdot f_o = \frac{1}{\alpha^2\beta} \cdot \frac{\alpha^2}{\beta}$$

$$P_{ds} = \frac{1}{\beta^2}$$

$$P_g = \frac{1}{\beta^2}$$

P_g is scaled by $\frac{1}{\beta^2}$

13) Power dissipation per unit area (P_a):-

$$P_a = \frac{P_g}{A_g} = \frac{\frac{1}{\beta^2}}{\frac{1}{\alpha^2}} = \frac{\alpha^2}{\beta^2}$$

P_a is scaled by $\frac{\alpha^2}{\beta^2}$

14) Power speed product (P_T):

$$P_T = P_g \times T_d = \frac{1}{\beta^2} \times \frac{\beta}{\alpha^2} = \frac{1}{\alpha^2\beta}$$

'P_T' is scaled by $\frac{1}{\alpha^2\beta}$

Note:

1) All dimensions are scaled by $\frac{1}{\alpha}$

2) Inputs, outputs and diameter of SiO_2 are scaled by i/p

3) constants are scaled by '1'.

Limitations of scaling:

Definition of scaling:

Varying (or) Adjusting the dimensions of MOS device along with the electrical properties is said to be scaling the spacing between source to drain is going to decreased then the channel length is decreased

where $\boxed{\therefore \text{ channel length } (L) = 2d}$

d = width of depletion layer

$$\text{Delay } [\tau] = \frac{\text{length of the channel}}{\text{drift velocity}}$$

$$\text{then } \tau = \frac{L}{V_{drift}}$$

where $L = 2d$

$$V = \mu \varepsilon$$

where $\varepsilon = \varepsilon_0 \varepsilon_{ins}$

$$\text{hence } \tau = \frac{2d}{\mu \varepsilon}$$

$$\text{hence } \boxed{\tau \propto d}$$

In this scaling process to vary the channel length by using channel length modulation.

Advantages of scaling:

* The channel length is reduced
* Switching speed is increased [ON & OFF actions]
* less power consumption.
* Drift velocity increases
* Area decreases.

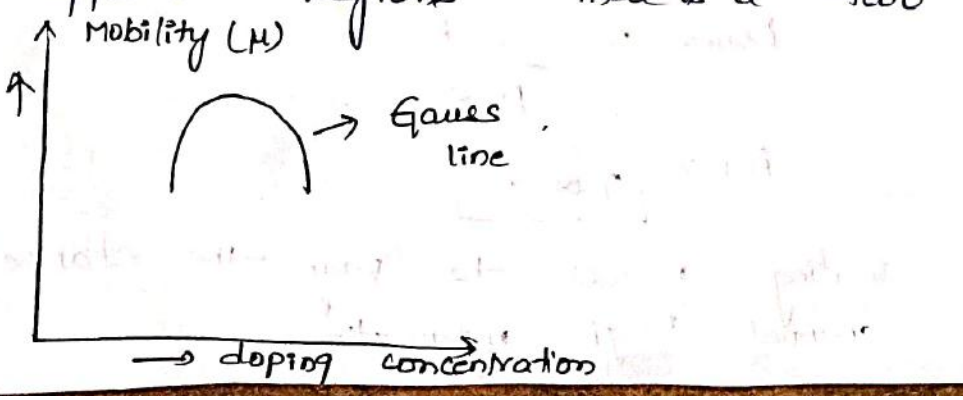
Disadvantages of scaling:

1) Noise effect (or) short channel effect:

* In this scaling process by using the channel length modulation, the channel length is minimum. So threshold voltage V_T of Mos device is reduced automatically. small noise signals are presented. In the i/p signal then the switching pattern will damage i.e. ON & OFF action.
* To overcome this problem we go for doping can be done differently in different regions of FET.

2) The FET having correctly doped the following characteristics presented in between the doping concentration Vs mobility by observing the characteristics at different regions so, we go for doping in diffusion regions there is a two possible ways.

Surface region	Region 1
Sub threshold region	Region 2
Sub Sub threshold region	Region 3



Limitations of scaling may cause a problem which prevents further miniaturization.

1) Substrate doping:

The built in (junction) potential V_B is small compared with V_{BD}

* As the channel length of MOS transistor is reduced, the depletion region width must also be scaledown

$$\text{depletion width } d = \sqrt{\frac{2\varepsilon_{si}\varepsilon V}{N_A q}}$$

N_A = substrate level doping, q = charge.

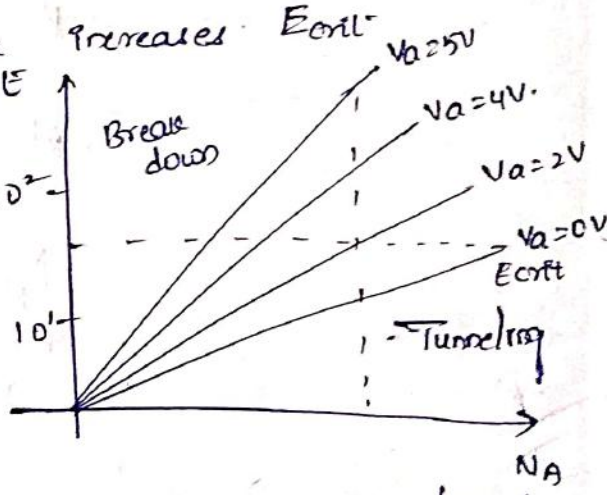
$V = V_a + V_B$ ↓ $\max(V_{DD})$, $V_B = \frac{kT}{q} \ln\left[\frac{N_A N_D}{n_i^2}\right]$

* When depletion width (d) is decreased, N_A increases which effects built in potential V_B increases.

2) Depletion width:

As depletion width decreases, N_A increases which increases built in potential V_B which further increases E_{crit}

$$d = \frac{\varepsilon_{si}\varepsilon_0 (E'_{crit})}{N_A \cdot q}$$



3) Limits of miniaturization:

- The minimum size of transistor effects the channel length. 'L'. 'L' can be decreased as long as there is no punch through to the depletion region around.

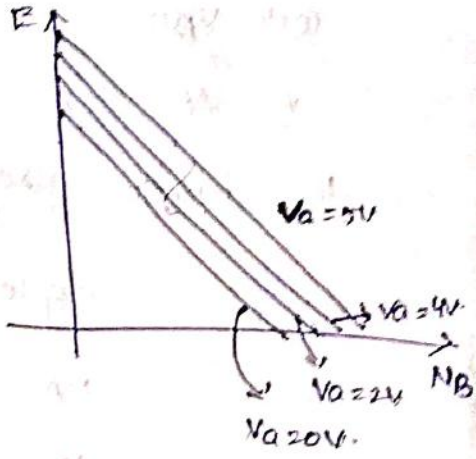
Source should not come closer to that around the drain. So L must be atleast $2d$ from meeting.

$$L = 2d$$

from semiconductor electronics $V_{drift} = \mu E$

$$t = \frac{L}{V_{drift}} = \frac{2d}{\mu E}$$

depletion width $d = \sqrt{\frac{2\varepsilon_{si}\varepsilon_0 V}{q N_B}}$



4) Interconnections & contact resistance:

Since width, thickness and spacing are scaled by $1/\alpha$, cross sectional area must be scaled by $1/\alpha^2$. Thus R is increased by α and I is scaled by $1/\alpha$. So IR drop remains constant. Thus driving capability and noise margins are degraded.

* The

$$L' \rightarrow L/\alpha.$$

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

$$R' = \frac{\rho L'}{A'} = \frac{\rho \cdot L/\alpha}{A/\alpha^2} = \alpha \cdot \frac{\rho L}{A}$$

$$\boxed{R' = \alpha R} \rightarrow \text{contact resistance.}$$

$$V = IR' = \frac{I}{\alpha} \cdot \alpha R \Rightarrow V \text{ is constant.}$$

$$\Rightarrow R = V/I$$

* (* In region-2 and region-3 · If does not have proper characteristics then we go for doping with boron the channel length will be decreased but the spacing b/n Source and drain also reduces for special requirements we can there should be in bonding (or) meeting. while taking the dimensions like length, width, thickness diffusion width correctly.) *

5) Limitation due to subthreshold currents:

$$I_{sub} \propto (V_{gs}-V_t)\frac{q}{KT}$$

whenver $\frac{V_{gs}-V_t}{KT}$ ratio decreases, sub-threshold currents. then it might damage the device.

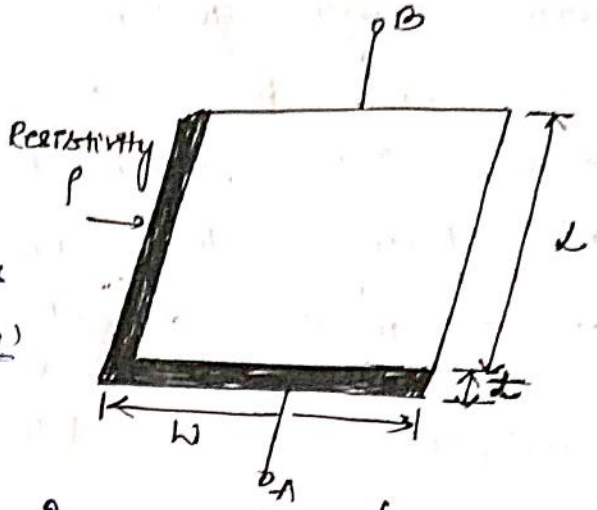
$$\text{Max}\{E_{max}\} = \frac{2\{V_a+V_B\}}{d}$$

(E_{max}) maximum electric field is depends on (V_B) built in potential & depletion width.

Basic circuit concepts

Sheet Resistance (R_s):

Consider a uniform slab of conducting material of resistivity ρ, width w, thickness t & length between faces $= L$



* Resistance R_{AB} between 2 opposite faces

$$R_{AB} = \frac{\rho L}{A}\,\Omega = \frac{\rho L}{t \cdot w}\,\Omega = \left(\frac{\rho}{t}\right) \cdot \frac{L}{w} = R_s \left(\frac{L}{w}\right)\Omega$$

where R_s = sheet resistance ohm per square

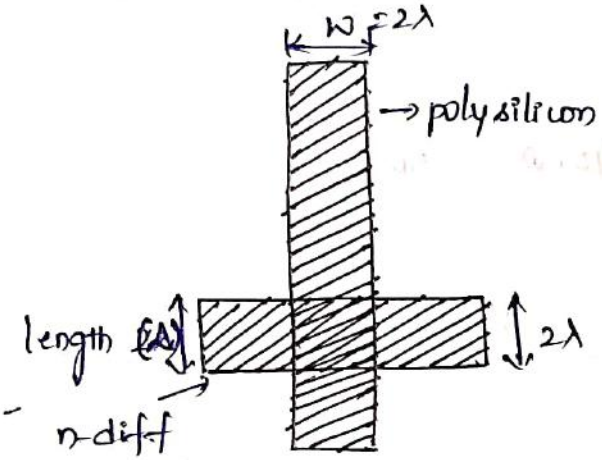
$= \frac{\rho}{t}$ ohm per square depends only on 'ρ' & 't'

* R_s is completely independent of area of square. R_{AB} is directly proportional to length & inversely proportional to width of channel.
* If $L = w$, then $R_{AB} = R_s$ i.e. the resistance of a channel which is a perfect square is a constant & is called sheet resistance
* The typical sheet resistance values for MOS layers. 5μm ($L = 2\lambda = 5\mu m$) are as follows.

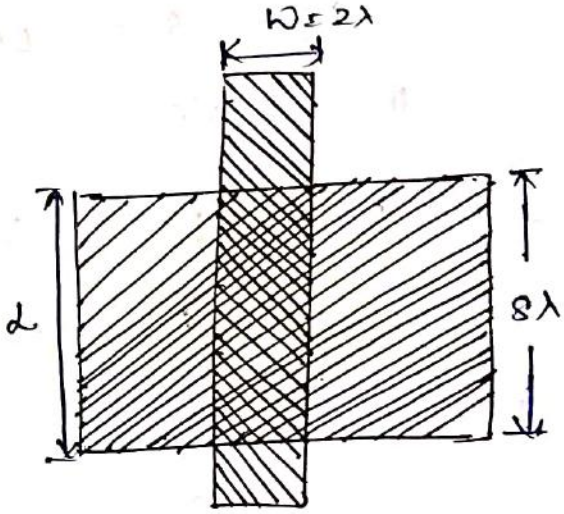
Layers	5 μm	orbit 2μm	orbit 1.2μm
Metal	0.03	0.04	0.04
Diffusions	10 → 50	20 → 45	20 → 45
Silicide	2 → 4	→	→
polysilicon	15 → 100	15 → 30	15 → 30.

layers	5μm	orbit 2μm	orbit 1.2μm
N- transistor channel	10^4	2×10^4	2×10^4
P- channel transistor	2.5×10^4	4.5×10^4	4.5×10^4

Sheet resistance concept is applied to Mos Transistors and Inverter



(a) N- channel Transistor with without Implant $L = 2\lambda$



(b) P channel Transistor. with $L = 8\lambda$ with Implant

* A simple N-Mos Transistor with channel length $L=2\lambda$ and channel width $W=2\lambda$. therefore the square channel resistance is

$$R = Z \cdot R_S \qquad Z = L/W = \frac{2\lambda}{2\lambda} = 1$$

R = 1 square $\times R_S$ ohm per square.

$$\boxed{R = 1\times10^4 \text{ ohms.} = 10\,K\Omega}$$

* Then we take P channel Mos transistor with the length $L = 8\lambda$ & $W = 2\lambda$ (with Implant) then we can Estimate the resistance.

$$R = Z \cdot R_S \Rightarrow R = R_S \cdot \frac{L}{W} = R_S \cdot \frac{8\lambda}{2\lambda} = 4$$

$$R = 4R_S = 4\,(2.5\times10^4)\,\Omega$$

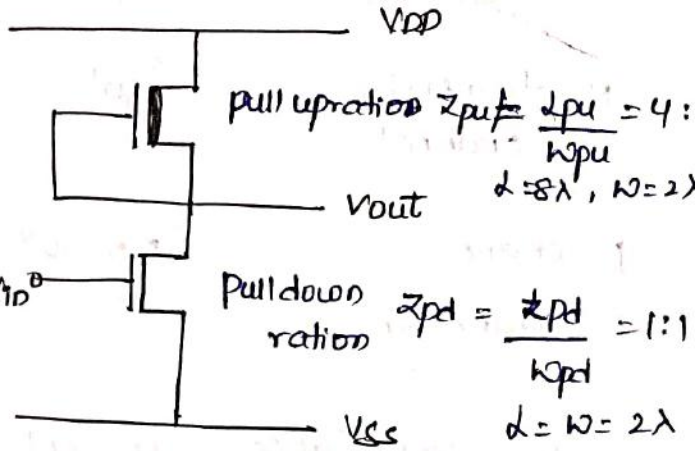
$$\boxed{R = 100\,K\Omega}$$

c) NMOS Inverter.

$$R_{pu} = R_s \cdot \frac{l}{w} = 10^4 \times \frac{4}{1}$$

$$= 4(10^4) = 40K\Omega$$

$$R_{pd} = R_s \cdot \frac{l}{w} = 1(10^4) = 10K\Omega$$



when $V_{in} = 5V$, pull down transistor is ON.

R_{ON} = static ON resistance from V_{DD} to V_{SS} through the Inverter.

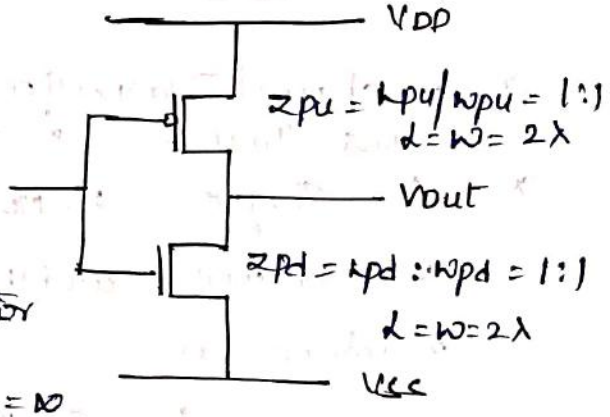
$$R_{ON} = R_{pd} + R_{pu} = 10K\Omega + 40K\Omega = 50K\Omega$$

$$\boxed{R_{ON} = 50K\Omega}$$

4) CMOS Inverter:-

$$R_{pd} = R_s \cdot \frac{l}{w} = 10^4(1) = 10K\Omega$$

$$R_{pu} = R_s \cdot \frac{w}{l} = 2.5 \times 10^4 = 25K\Omega$$



In static condition, one of the transistor is OFF, so resistance of Inverter $R_{ON} = \infty$

R_{ON} = static resistance from V_{DD} to $V_{SS} = \infty$

(Ideal condition)

If both transistors are ON for a small duration (non ideal)

$$R_{ON} = R_{pu} + R_{pd} = 25K\Omega + 10K\Omega$$

Area capacitance for layer:

$$C = \frac{\varepsilon_0 \varepsilon_{ins}\, \omega L}{D} \quad \text{Paradays}$$

$$C = \frac{\varepsilon_0 \varepsilon_{ins}\, A}{D} \quad \text{Paradays}$$

$$\left[\because \varepsilon_{ins} = 4.0 = \text{relative permittivity}, \quad \because \varepsilon_0 = 8.85 \times 10^{-4}\, pf/cm\right]$$

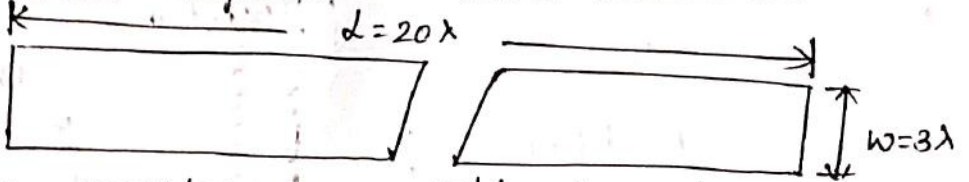
$$C\,[pf \times 10^{-4}/\mu m^2] = \frac{\varepsilon_0 \varepsilon_{ins}}{D} \frac{F}{cm^2} \times \frac{10^2 pf}{1} \frac{x\, cm^2}{10^8\, \mu m^2}$$

typical values for capacitance

Value in $pf \times 10^{-4}/\mu m^2$ (Relative values in bracket)

Capacitance	5 μm	orbit 2 μm	orbit 1.2 μm
Gate to channel	4(1.0)	8(1.0)	16(1.0)
Diffusions	1(0.25)	1.75(0.22)	3.75(0.23)
Polysilicon to substrate	0.4(0.1)	0.6(0.75)	0.6(0.038)
Metal1 to substrate	0.3(0075)	0.33(004)	0.33(0.02)
Metal2 to substrate	0.2(0.05)	0.17(0.02)	0.17(001)
metal2 to metal1	0.4(0.1)	0.5(0.6)	0.5(0.03)
Metal2 to polysilicon	0.2(0.075)	0.3(0.038)	0.3(0.018)

Some area capacitance value calculations:



assume considering a metal1 whose length $L = 20\lambda$, and $W = 8\lambda$ which is capacitance value undertaken by ratio feature area.

$$\text{relative area} = \frac{\text{area of interest}}{\text{area of standard}}$$

$$\text{relative area} = \frac{20\lambda \times 3\lambda}{2\lambda \times 2\lambda} = \frac{60\lambda^2}{4\lambda^2} = 15$$

Consider the area in metal 1

Capacitance of substrate = relative area x relative c value.

$C_g = 15 \times 0.075 \square C_g$

$= 1.125 \square C_g$

Consider the same area in polysilicon

Capacitance of polysilicon = Relative area x Relative C value.

Capacitance of substrate = $15 \times 0.1 \square C_g = 1.5 \square C_g$.

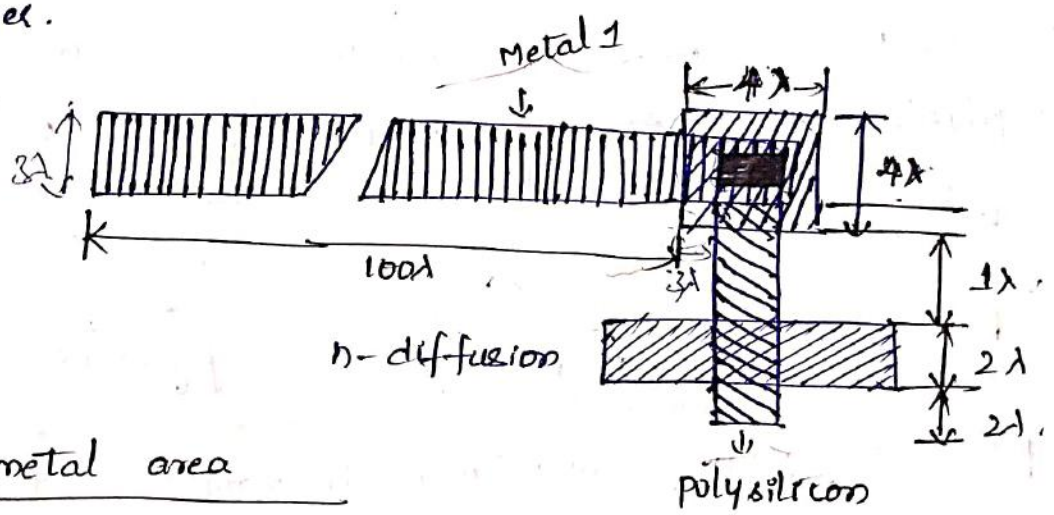
Consider the same area in n-type diffusion.

Capacitance of n-diffusion = relative area x relative C value

Capacitance of Substrate = $15 \times 0.25 \square C_g = 3.75 \square C_g$

Routing Capacitance:

Area of capacitance associated with structure more than one layer.



Ratio = $\frac{\text{metal area}}{\text{Standard gate area.}}$

$= \frac{100\lambda \times 3\lambda}{2\lambda \times 2\lambda} = \frac{100\lambda \times 3\lambda}{4\lambda^2} = \frac{300\lambda^2}{4\lambda^2} = 75$

Metal capacitance $C_m = 75 \times 0.075 = 5.625 \square C_g$

polysilicon area = $4\lambda \times 4\lambda + 3\lambda \times 2\lambda = 16\lambda^2 + 6\lambda^2 = 22\lambda^2$

polysilicon capacitance $\boxed{C_p = \frac{22}{4} \times 0.1 = 0.55 \square C_g}$

Gate capacitance for transistor $\boxed{C_g = 1 \square C_g}$

Total capacitance $C_T = C_m + C_p + C_g =$

$= (5.625 + 0.55 + 1) \square C_g$

$\boxed{C_T = 7.20 \square C_g}$

* Now the width of the metal is increased from 3λ to 4λ then we can estimate the metal 1 capacitance then

Area of the metal = $100\lambda \times 4\lambda$

Relative area of metal = $\frac{\text{Actual area of metal 1}}{\text{Standard area}}$

$= \frac{100\lambda \times 4\lambda}{2\lambda \times 2\lambda} = \frac{400\lambda^2}{4\lambda^2} = 100$

Capacitance for metal 1 C_M = Relative area of Metal 1 × relative capacitance for metal.

$C_M = 100 \times 0.075 \Delta C_g$

$C_M = 7.5 \Delta C_g$

In this case the metal capacitance is increased from 5.6 to $7.5 \Delta C_g$ the overall capacitance increased to

$C_T = 7.5 + 0.55 + 1 \Rightarrow \boxed{C_T = 9.05 \square C_g}$

Analytic Inverter delays:-

Consider a N-MOS inverter with the basic ratio of 4:1 in order to achieve the 4:1 Zpu to Zpd ratio. Rpu will be 4*Rpd as contributed by the minimum sized Transistor then

$$Rpu = 4 * Rs = 4 \times 10K\Omega$$

$$Rpu = 40K\Omega$$

$$Rpd = 1 * Rs = 1 \times 10K\Omega$$

$$Rpd = 10K\Omega$$

The total ON resistance for N-MOS Inverter is RON

$$R_{ON} = Rpu + Rpd$$

$$= (40+10)\ K\Omega$$

$$\boxed{R_{ON} = 50K\Omega}$$

If we consider a pair of cascade Inverters then the delay over the pair will be constant irrespective of the logical level transition of the input we can assume $\tau = 0.3nsec$. and making no extra allowances for wiring capacitances we have the overall delay through a pair of similar N-MOS Inverters is

$$Td = \left(1 + \frac{Zpu}{Zpd}\right)\tau$$

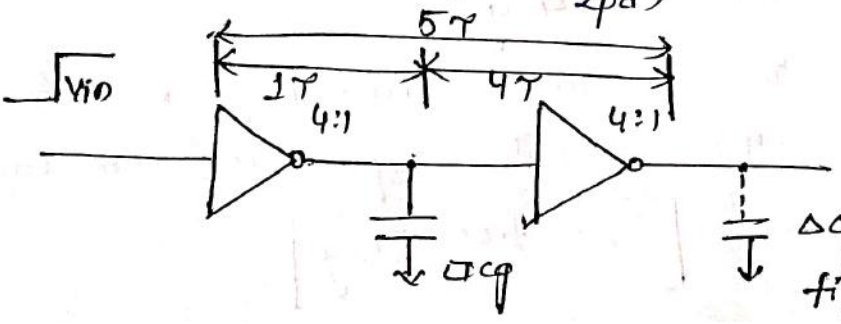


fig: NMOS Inverter delay model

$T_d = 1\tau + 4\tau$

$\boxed{T_d = 5\tau}$

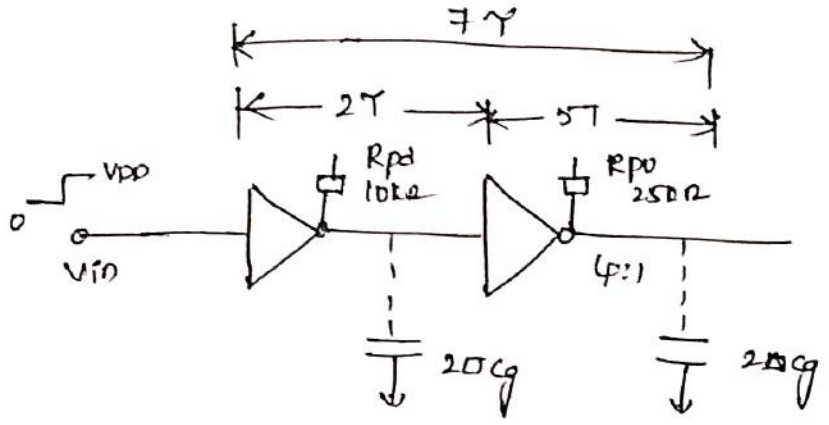


fig: CMOS Inverter delay model

Similar to the CMOS Inverter the overall delay

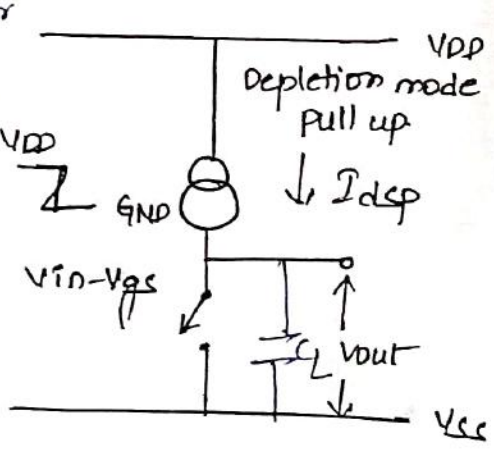
$T_d = 2\tau + 5\tau$

$\boxed{T_d = 7\tau}$

The gate capacitance will be doubled ($2C_g$) as compared with the N-MOS Inverter.

Raise time Estimation:

We can assume that the p-MOS Transistor is in saturation for the entire charging period of load capacitor C_L. the ckt may be shown in figure.



The saturation current drawn for p-MOS Transistor is given by

$$I_{dsp} = \frac{\beta_p}{2}\left[V_{gs} - |V_{tp}|\right]^2 \text{ then}$$

$$V_{out} = \frac{I_{dsp} \cdot t}{C_L}$$

$$t = \frac{V_{out} \cdot C_L}{I_{dsp}}$$

$$t = \frac{V_{out} \cdot C_L}{\frac{\beta_p}{2}\left[V_{gs} - |V_{tp}|\right]^2}$$

$$t = \frac{2\, V_{out} \cdot C_L}{\beta_p \left[V_{gs} - |V_{tp}|\right]^2}$$

Assume that $V_{gs} = V_{DD}$, $V_{tp} = -0.2 V_{DD}$

then $t = t_r$

$$t_r = \frac{2\, V_{DD} \cdot C_L}{\beta_p \left[V_{DD} - 0.2 V_{DD}\right]^2}$$

$$= \frac{2\, V_{DD} \cdot C_L}{\beta_p \left[0.8 V_{DD}\right]^2}$$

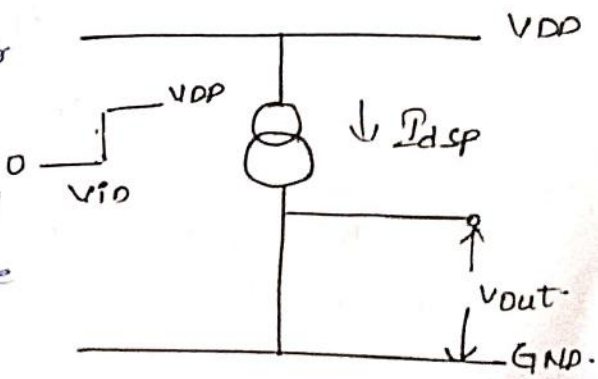
$$= \frac{2\, V_{DD} \cdot C_L}{\beta_p \left(0.64 V_{DD}^2\right)}$$

$$= \frac{2 C_L}{\beta_p\, 0.64\, V_{DD}}$$

$$\boxed{t_r = \frac{3 C_L}{\beta_p \cdot V_{DD}}}$$

2) Fall time estimation (T_f):

It is similar the load capacitor (discharging condition) will connected to NMOS transistor. the ckt model is shown in figure making the assumptions we may write the fall time



$$T_f = \frac{3CL}{\beta_n V_{DD}}$$

By using the expressions raise time & fall time we denoted as

$$\frac{T_r}{T_f} = \frac{\beta_n}{\beta_p}$$

W.K.T

$$\beta_n = \frac{\varepsilon_0 \varepsilon_{ins}}{D} \cdot \mu_n \cdot \frac{W_n}{L_n}$$

$$\beta_p = \frac{\varepsilon_0 \varepsilon_{ins}}{D} \cdot \mu_p \cdot \frac{W_p}{L_p}$$

$$\mu_n = 2.5 \mu_p$$

$$\frac{\mu_n}{\mu_p} = 2.5$$

$$\beta_n = 2.5 \beta_p$$

$$\frac{T_r}{T_f} = \frac{2.5 \beta_p}{\beta_p}$$

$$\frac{T_r}{T_f} = 2.5$$

$$\boxed{T_r = 2.5 T_f}$$

* The raise time and fall time $[T_r, T_f]$ are proportional to the $\frac{1}{V_{DD}}$
* The raise time and fall time are proportional to C_L (load capacitor.)
* The $T_r = 2.5 T_f$, for equal N-MOS & P-MOS Transistor, Geometrical

driving large capacitive loads:

There is a problem of driving comparitively large capacitive loads arises when the signal must be propagated ON chip to off chip dimensions. Generally typical off chip capacitance may be several order higher than ON chip capacitance value. It is denoted as $\square C_g$.

for example. if the off chip load is denoted as C_L then $C_L \geq 10^4 \square C_g$

For an inverter implementation the drive large capacitive load must be present low. pull up to the pull down resistance otherwise excessively large. delays will occurs for an MOS circuits the low length to width ratio in otherwords. the channel must be made very wide. to reduce the resistance value it is not possible because. of the technology limitations. then the width is increased to several factors to achieve the low resistance.

* Here to increase, the width, the area of the MOS device is increases. To overcome this problem to use the N-cascade inverter each one of which is largee than the preceeding stage by width factor of 'f' as shown in figure.

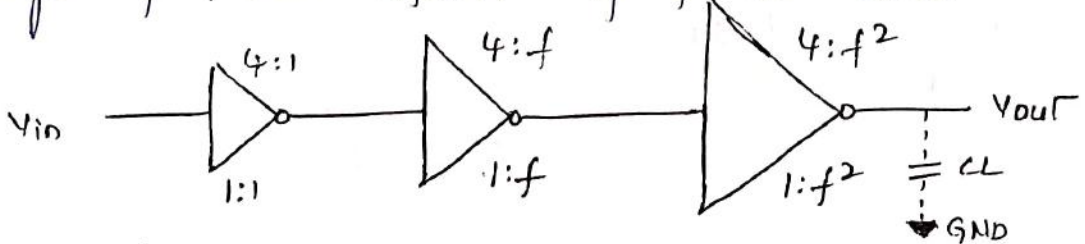


fig: Cascading of Inverters as delays drivers

* In this configuration the width factor is increases So, the capacitive load present at the inverter i/p increases & area occupied is also increases, the rate at which width

Inverter will must be cascaded. to drive a particular value of C_L for an optimum solution with large 'f', 'N' increases but delay per stage increases.

for 4:1 N-MOS Inverter delay per stage $= 1\tau$ for ΔV_{in}

$= 4.f\tau$ for ∇V_{in}

where ΔV_{in} = indicates logic '0' to logic '1' transition

∇V_{in} = indicates logic '1' to logic '0' transition.

The total delay per N-MOS path = $5.f\tau$

total delay per CMOS path = $7.f\tau$

let $Y = \frac{C_L}{\sum C_g} = f^N$ where f, N are inter dependent.

$Y = f^N$

$\ln(Y) = \ln(f^N)$

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

$$\boxed{N = \frac{\ln(Y)}{\ln(f)}}$$

If 'N' is even, then the total delay is

$$\text{total delay} = \frac{N}{2} 5.f\tau$$

$$= 2.5Nf\tau \quad [\text{NMOS Inverter}]$$

$$\text{total delay} = \frac{N}{2} 7.f\tau$$

$$= 3.57f\tau \quad [\text{CMOS Inverter}]$$

for all possible cases:

$$\boxed{\text{delay} \propto Nf\tau = \frac{\ln(Y)}{\ln(f)} \cdot f\tau}$$

Now we can assume $f = e$

Number of stages $N = \frac{\ln(V)}{\ln(f)} = \frac{\ln(y)}{\ln(e)}$

$$\boxed{N = \ln(y)}$$

Overall delay t_d

If N is even $t_d = 2.5\,e\,N\,\tau$ [NMOS]

(or)

If N is even $t_d = 3.5\,e\,N\,\tau$ [CMOS]

N is odd, $t_d = [2.5(N-1)+1]\,e\,\tau$ (NMOS) } for ΔV_{io}

$t_d = [3.5(N-1)+2]\,e\,\tau$ [CMOS] }

$t_d = [2.5(N-1)+4]\,e\,\tau$ [NMOS] } for ∇V_{io}

$t_d = [3.5(N-1)+5]\,e\,\tau$ [CMOS] }

Super buffers:

* The asymmetry of the conventional Inverter is clearly undesirable, and gives rise to significant delay problems when an inverter is used to drive more significant capacitive loads.

* A common approach used in nMos technology to alleviate this effect is to make use of Super buffers.

* An inverting type as shown in below fig (a). considering a +ve going logic transition Vin at input, it will be seen that the inverter formed by T_1 & T_2 is turned 'ON'. & thus gate of T_3 is pulled down toward '0' V with a small delay. thus, T_3 is cutoff while T_4 (The gate of which is also connected to Vin) is turned on and the output is pulled down quickly. So Vout = 0V.

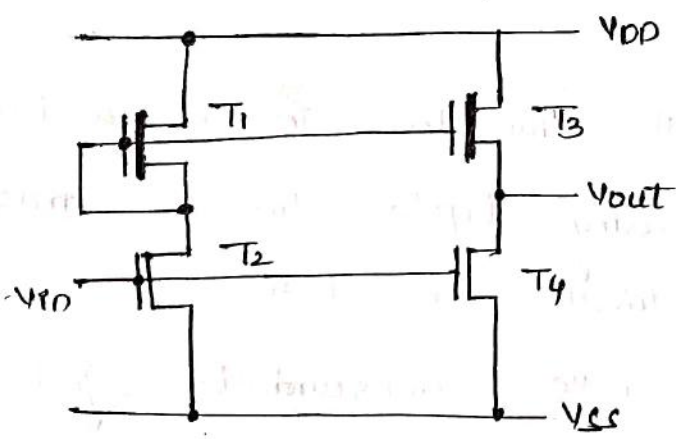


fig (a): Inverting type nMos Super Buffer

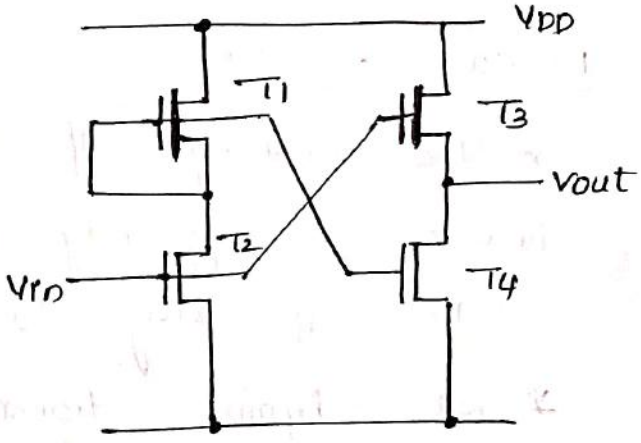


fig (b): non Inverting type NMos Super buffer

* Now consider the opposite transition: when Vin drops to 0V then the gate of T_3 is allowed to rise quickly to VDD thus T_4 is also turned off by Vin, T_3 is made to conduct

with VDD on its gate, thus with hence the average voltage would apply if the gate was tied to the source as in the conventional nMos Inverter. Now since $I_{ds} \propto V_{gs}$ then doubling the effective V_{gs} will increase the current & thus reduce the delay in charging any capacitance on the output, so that more symmetrical transitions are achieved. So $V_{out} = V_{DD}$.

* the corresponding non inverting nMos super buffer ckt as shown in above fig (b). the structures when realized in 5μm technology are capable of driving loads of 2pF with 5 nsec rise time.

* other nMos arrangements such as those based on the native transistor & known as native super buffers may be used but such process are not readily available. to the designer and are mentioned here only.

BICMOS drivers:

* The availability of bipolar transistors in BiCMos technology presents the possibity of using bipolar transistor drivers as the o/p stage of Inverter and logic gate ckts.

* W.r.T bipolar transistors have transconductance gm & current area (-I/A) characteristics that are greatly superior to those Mos devices. this Indicates high current driving capabilities for small areas of silicon

* In bipolar CE transistor devices, there will be a increase in output current (I_C) with smaller change in voltage (V_{BE}). than Mos Transistors

So thus bipolar transistors have a much better switching performance, primarily as a result of smaller i/p voltage swings. only a small amount of charge must be moved during switching.

* It is possible effect of temperature T on the required i/p voltage V_{BE} & is logarithmically dependent on base width W_B, doping level N_A, Electron mobility μ_n & collector current I_C. is linearly dependent on T. as the temperature differences across a chip will not be sufficient to cause more than few 'mV' of difference in V_{BE} between any two bipolar transistors.

* The switching performance of a transistor driving a capacitive load may be visualized. It may be shown that time Δt necessary to change the o/p voltage V_{out} by an amount Equal to i/p voltage V_{in} is given by

$$\Delta t = \frac{C_L}{g_m}$$

where
g_m = device transconductance.

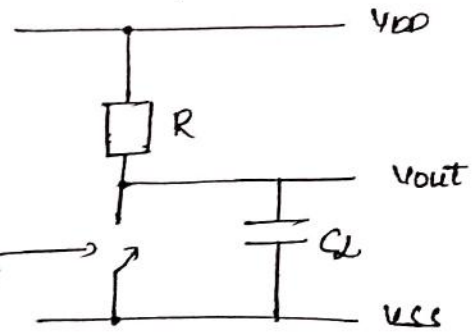


fig: driving ability of bipolar transistor.

* Since, the bipolar transistor has a relatively high transconductance, the value of Δt is small.

* Bipolar Transistor Delay consists of two main timing components T_{in} & T_L

* T_{in}: Δt is an initial time required to charge the base Emitter (V_{BE}) junction of npn transistor. in CE configuration.

* T_L: It is the time taken to charge the load capacitor.

for BICMOS driven T_{in} = 2nsec.

for CMOS Transistor, driven for Same BICMOS technology 1nsec.

* To change the input gate capacitance.
* For gallium Arsinide T_{in} - 50-100 psec.
* T_L is less for bipolar by a factor of h_{fe} (common Emitter current gain).

The delay of BICMOS transistor is

$$T = T_{in} + (V/I_d) \cdot 1/h_{fe} \cdot C_L$$

Gate Level Design

Introduction:

The gate level modelling is used for implementing lowest level modules in design like a full adder, multiplexer etc.,. The module (Integrated circuit) is implemented in terms of logic gates and interconnections between these gates.

* Boolean algebra is used to represent logical (combinational logic) functions of digital circuits. A combinational logic Expression is a mathematical formula which is to be interpreted using the laws of boolean Algebra.

* A logic gate is an idealized or physical device implementing a boolean function, that is it performs a logical operation on one or more logic inputs and produces a single logic output.

* logic gates are primarily implemented using diodes or transistors acting as electronic switches. but can also be constructed using electromagnetic relays (relay logic), fluid logic Pneumatic logic, optics, molecules or even mechanical elements.

* Simplest form of electronic logic is diode logic. This allows AND & OR gates to be built, but not inverters.

* The Simplest family of logic gates using bipolar Transistors is called resistor-Transistor logic (RTL). Unlike diode logic gates, RTL gates can be cascaded indefinitely to produce more complex logic functions. For higher speeds, the resistors used in RTL were replaced by

diodes, leading to diode Transistor logic (DTL)

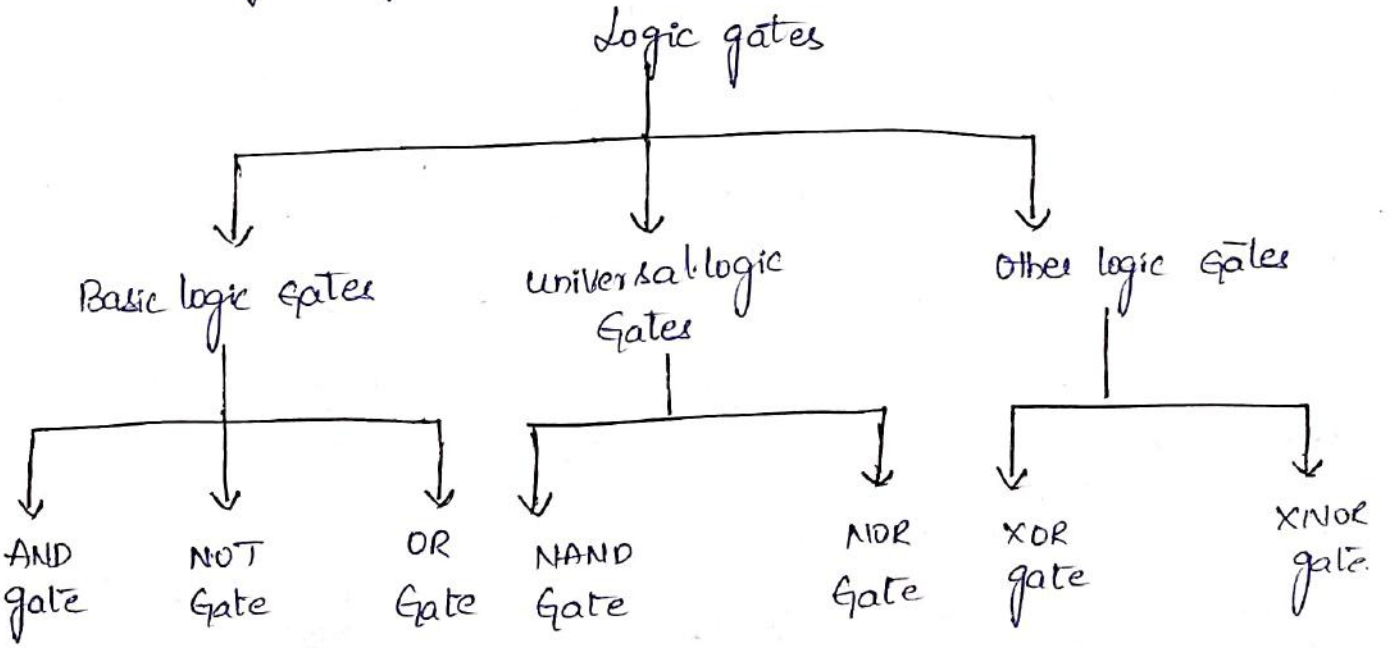
* Transistor - Transistor logic (TTL) then supplanted DTL with the observation that one transistor could do the job of two diodes even more quickly.

* In virtually every type of contemporary chip implementation of digital systems, the bipolar transistors have been replaced by complementary field effect transistors (MOSFETs) to reduce size and power consumption still further, thereby resulting in complementary metal oxide semiconductor (CMOS) Logic. that can be described with boolean logic

Logic Gates:

* Logic gates are the basic building blocks of any digital ckt

* Logic gates are classified as

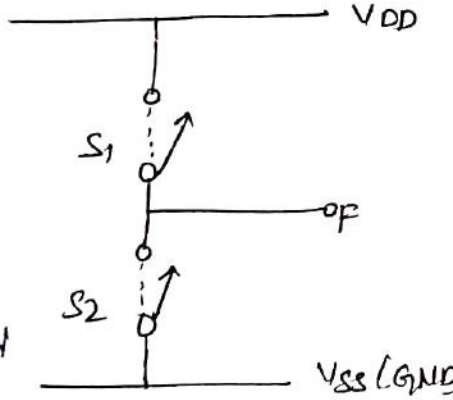


* Logic gates are designed by using CMOS gate design. CMOS static gates have two transistors & pMOS &

NMOS whose topologies are related.

CMOS Logic Gates and other complex gates:

General Logic Gates: Any boolean logic function (F) has two possible values Either logic '0' or logic '1'. For some of i/p combinations, F=1, & for all other i/p combinations, F=0,



* The switch S_1 is closed and S_2 is open for input combinations that produces F=1
* The switch S_1 is open and S_2 is closed for input combinations that produces F=1
* The switches S_1 & S_2 are open for input combinations that produces F=0. ↑
* thus o/p (F) is Either connected to VDD or ground. So the requirement of digital logic design is Implement the pullup switch (S_1) & pull down switch (S_2).

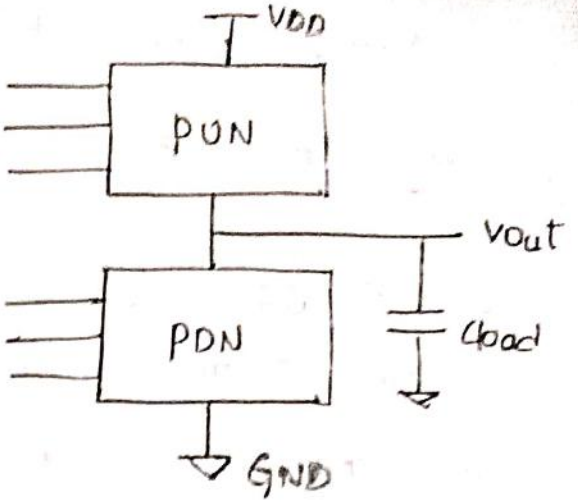
CMOS static logic:

A generalized CMOS logic ckt consists of two transistor networks NMOS & PMOS. The PMOS is connected b/n VDD & logic gate output called as pull up network. and the NMOS transistor n/w is connected between the output & ground called pull down network. Depending on the applied input logic, the PUN connects the O/p node to VDD & PDN connects the o/p node to ground.

* Design the pull down network (PDN) by realizing

→ AND (product) terms using series-connected NMOSFETS.

→ OR (Sum) terms using paralled connected NMOSFETS.



* Design pull-up network (PUN) by realizing

→ AND (product) terms using paralel connected PMOSFETS.

→ OR (Sum) terms using series connected PMOSFETS.

* Add an inverter to the output to complement the functions such as NAND, NOR gates do not need an inverter at the output Terminals.

CMOS Inverter:

A CMOS Inverter is the simplest logic act that uses one NMOS and one PMOS Transistor. the NMOS is used in PDN & PMOS is used in PUN as shown in figure.

Working operation:

1) when Vin = logic '1' then NMOS Transistor is ON & PMOS is off. Thus Vout is pulled down to ground Vout = logic '0' since it is connected to ground.

2) when Vin = logic '0', then NMOS is off & PMOS is ON. Thus Vout = logic '1' which is pulled to VDD. Since it is connected to source via PMOS.

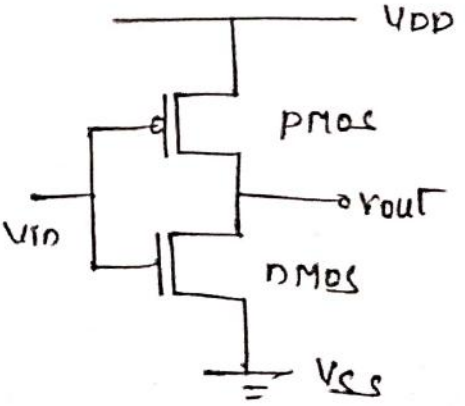
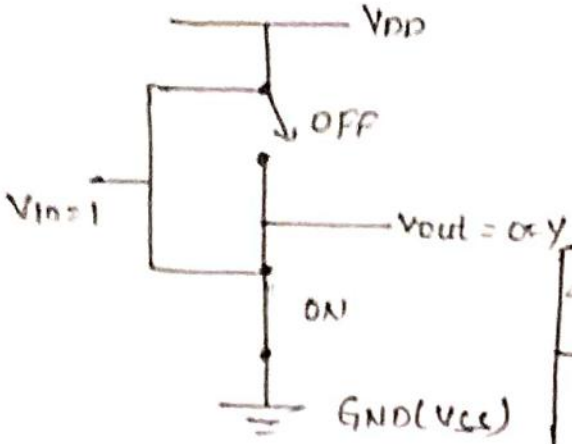
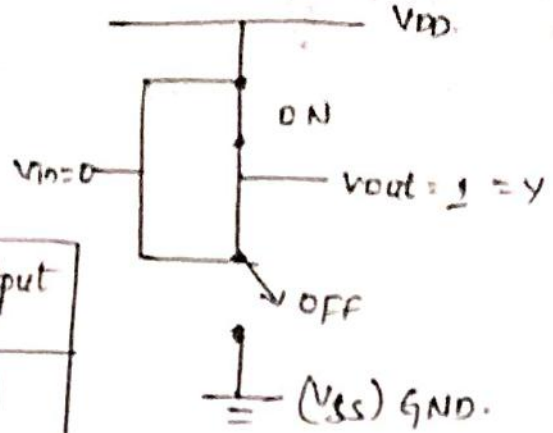


fig: CMOS Inverter



fig(a) : $V_{in} = 1$

Input A	output y
0	1
1	0



fig(b) : $V_{in} = 0$

CMOS NAND gate:

* The two input NAND function is expressed by $Y = A \cdot B$ & its complement $Y = \overline{A \cdot B}$

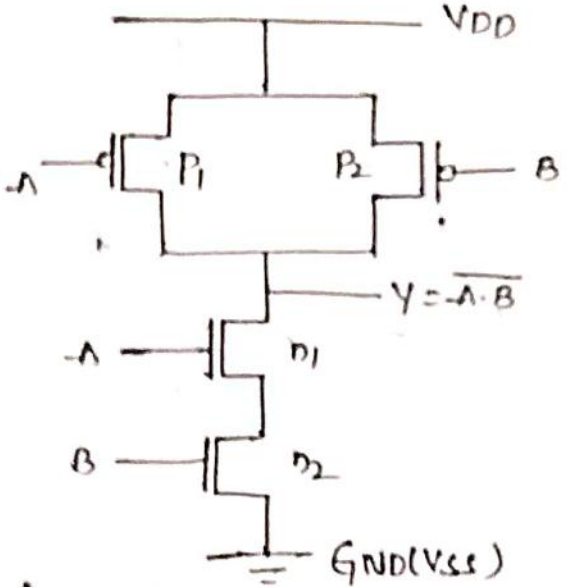


fig: 2 i/p CMOS NAND gate

A	B	O/P $Y = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

Truth Table

* here in the pull up network two PMOS transistors are connected in parallel and in pull down network two NMOS transistors are connected in series which realizes 2-input NAND gate.

Working operation:

1) when ever atleast one of the inputs is low, the corresponding PMOS is ON & N-MOS transistor is OFF so the output Vout = logic '1' (High).

2) conversely, if both inputs A = B = 1 (High), then both PMOS are is OFF and NMOS is ON. & the output Vout = Y = Logic '0'

CMOS NOR gate:

* The two input NOR function is expressed by $Y = \overline{A+B}$
* The PMOS transistors are connected in series in pull up network and NMOS transistors are connected in parallel acts as pull down network. as shown in figure.

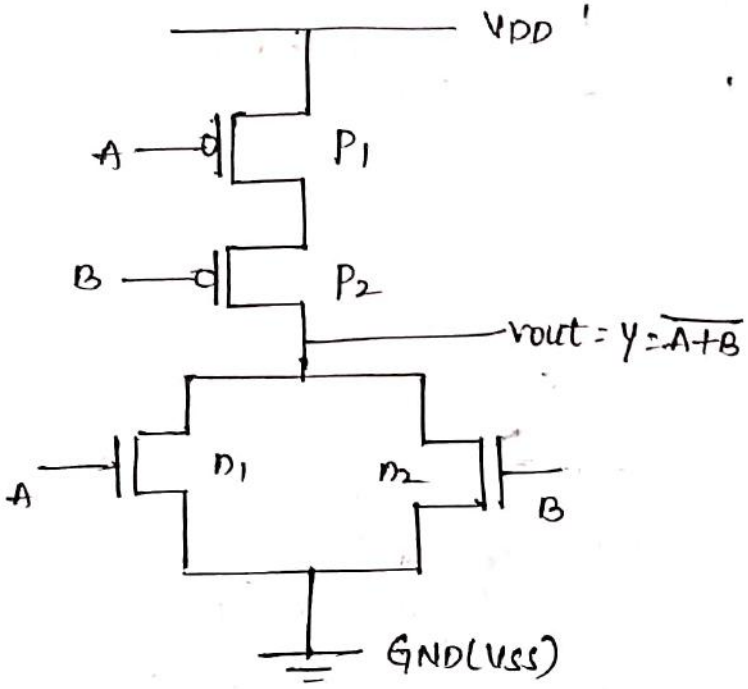


fig: 2 input CMOS NOR Gate

A	B	$Y = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Truth Table

Working Operation:

1) when ever both inputs A = B = 0 then PMOS transistors is ON and NMOS is off so the output (Y) = logic '1' (High).

2) when A = 0 & B = 1 (or) A = 1 & B = 0 then the corresponding PMOS transistors is ON & NMOS is OFF so output Y = logic (Low)

3) when A = 1 & B = 0 then corresponding PMOS is OFF & NMOS i.e P_1 is OFF & P_2 is ON & NMOS transistors N_1 is ON & N_2 is OFF. So the output Y = logic '0' (Low).

4) when both inputs A = B = 1, PMOS transistor is OFF & NMOS is ON so O/P Y = logic '0' (Low).

complex gates in CMOS Logic:

A complex logic gate is one that implements a function that can provide the basic NOT, AND & OR operation but integrates them in to a single circuit. CMOS is ideally suited for creating gates that have logic equations by exhibiting the following.

1) AND-OR Invert form (AOI)

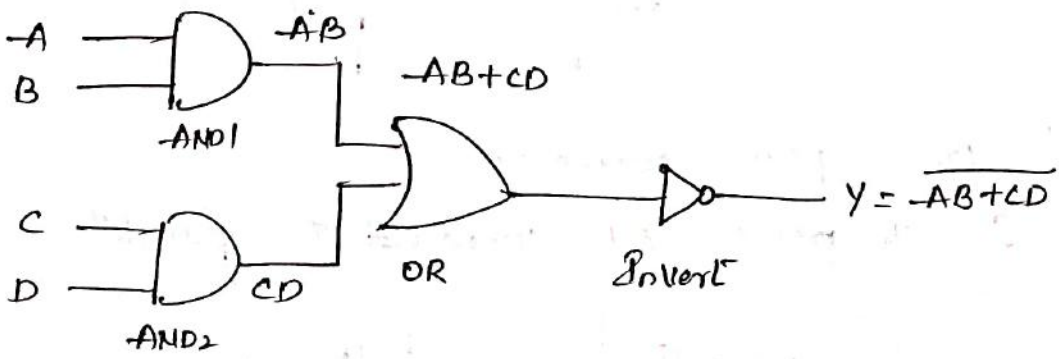
2) OR-AND Invert (OAI) form

→* An AOI logic equation is equivalent to a complemented sop form, while an OAI Equation is equivalent to a complemented pos form. In CMOS, output always produces NOT operation acting on i/p variable.

1) AOI Logic function (OR) Design of XOR gate using CMOS Logic:

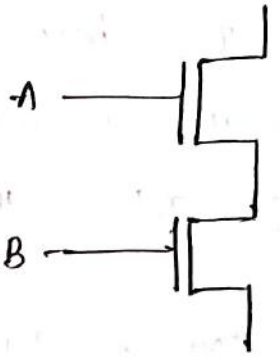
AND-OR Invert logic function (AOI) implements Equation in the order AND, OR, NOT. whose o/p function is

$Y = \overline{AB + CD}$

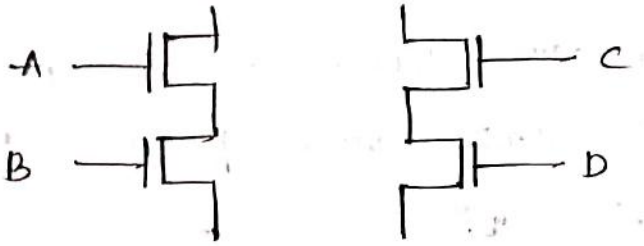


CMOS Implementation for Y

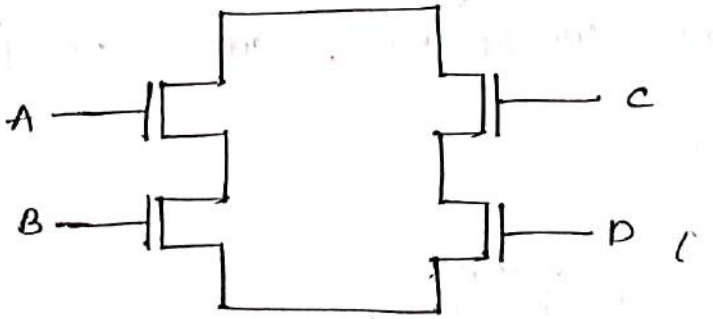
Step1: Draw AB (AND) function first by connecting 2 NMOS Transistors in series



Step2: Draw CD (AND) Implementation by using 2 NMOS Transistors in series.

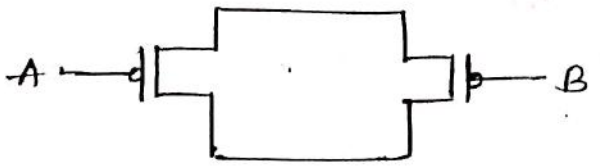


Step3: $Y = AB + CD$, In this function AB and CD are added for addition, we have to draw parallel connection. So AB series connected in parallel with CD as shown in fig.



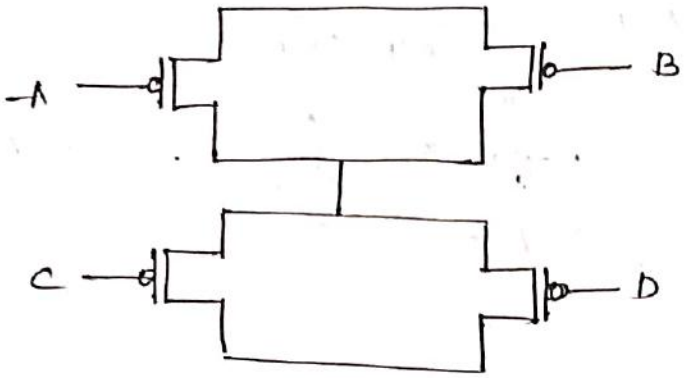
Step4: Draw PMOS connection.

I. In PMOS AB connected in parallel.

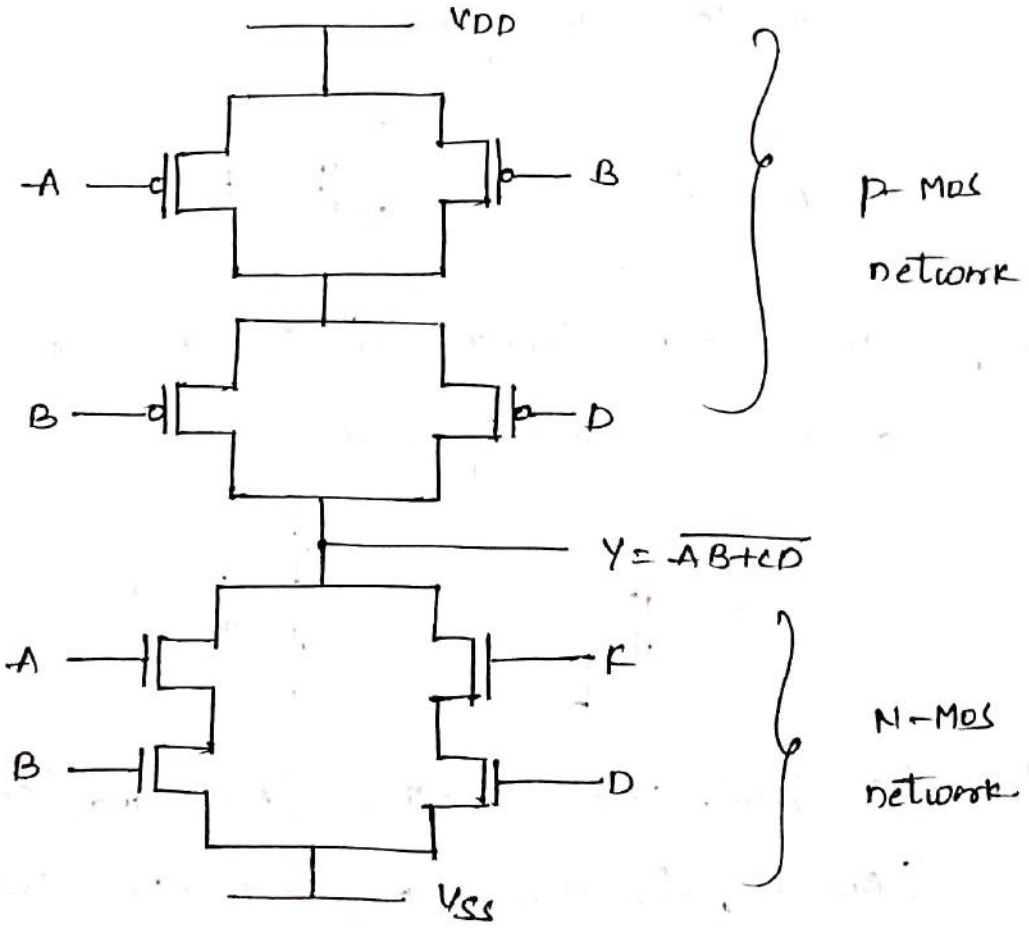


ii. In pMOS CD should be connected in parallel

iii. In pMOS AB and CD networks should be connected in series. for addition.



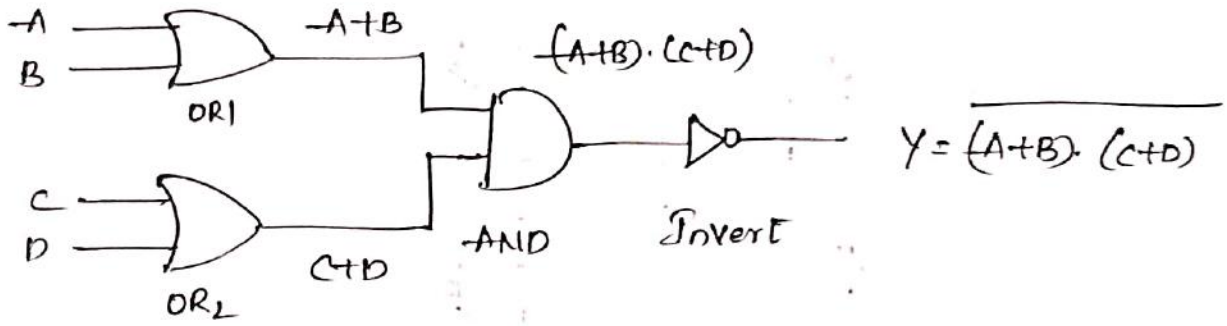
Step 5: Take output at the point in between nMOS and pMOS networks.



2) OAI Logic function (or) Design of XNOR gate using CMOS Logic :

OR-AND Invert logic function (AOI) Implements operation in the order OR, -AND, NOT. Whose output function is

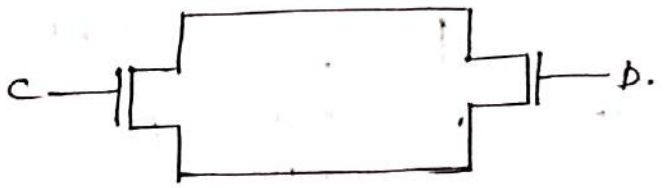
$$Y = \overline{(A+B)\cdot(C+D)}$$



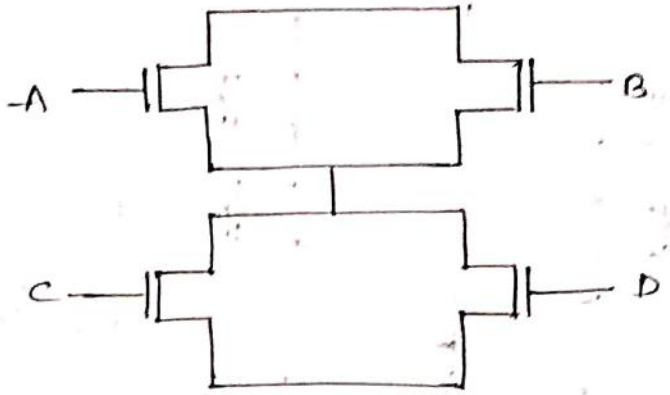
CMOS Implementation for Y :

Step1: Draw A+B function first by connecting 2 NMOS Transistors in parallel.

Step2: Draw C+D function first by connecting 2 NMOS Transistors in parallel.

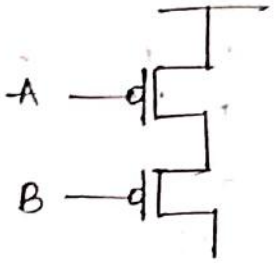


Step3: Y = (A+B)·(C+D), In this (A+B) and (C+D) are Multiplied (Anded), for this we have to draw series connection So (A+B) and CD are connected in series.

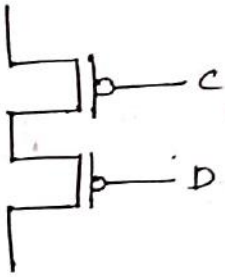


Step4: Draw pMOS connection:

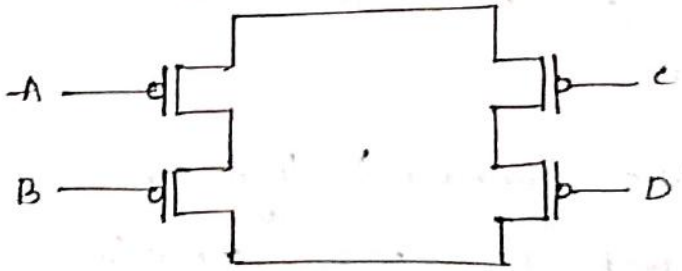
I. In pMOS (A+B) are connected in series



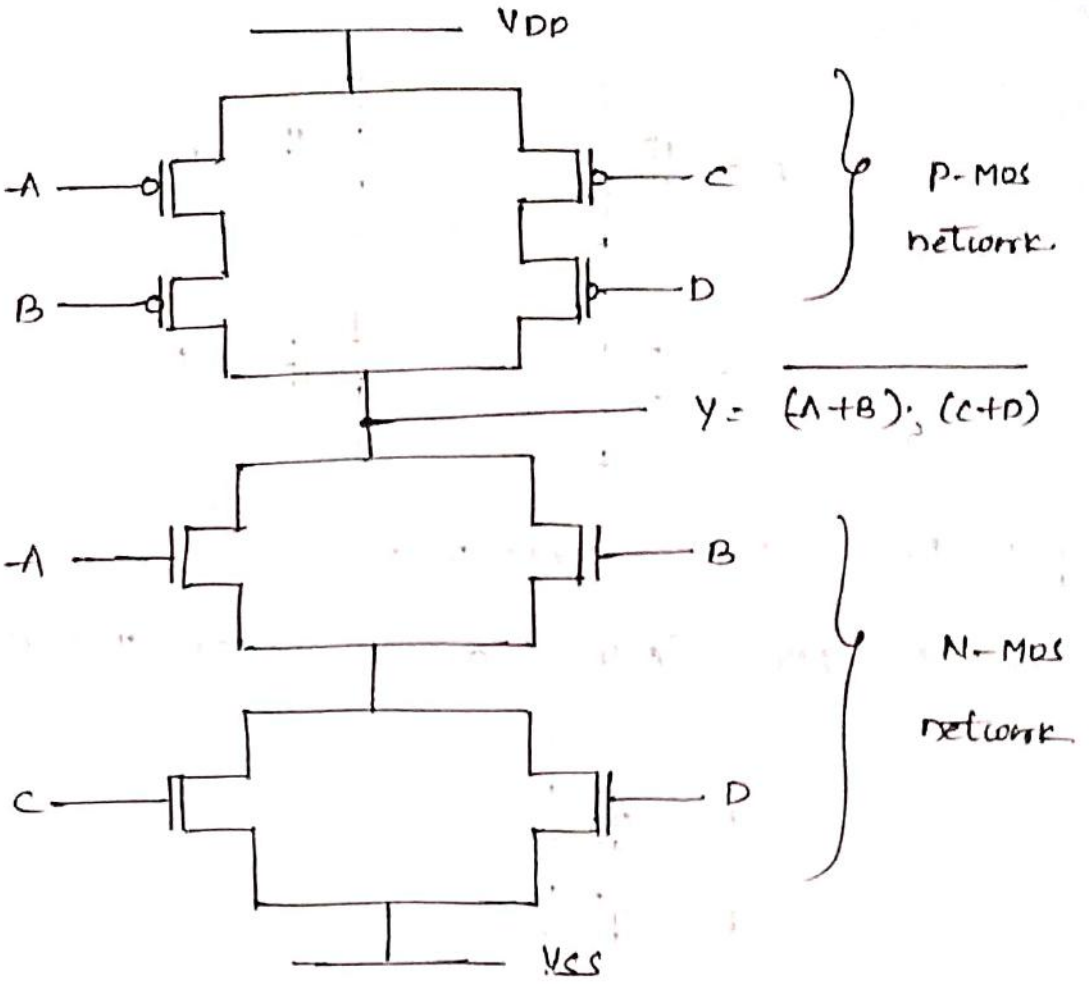
II. In pMOS (C+D) are connected in series



III. In pMOS (A+B) and (C+D) networks should be connected in parallel for Multiplication.



Step5: Take output at the point in between nMOS & pMOS networks



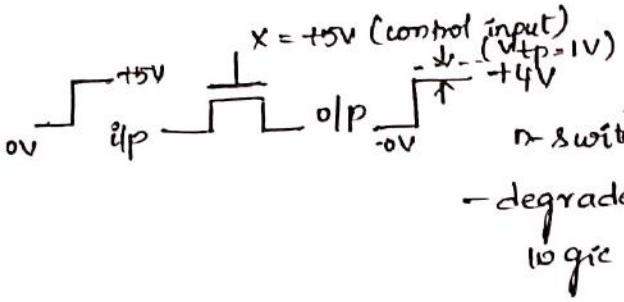
Switch Logic:

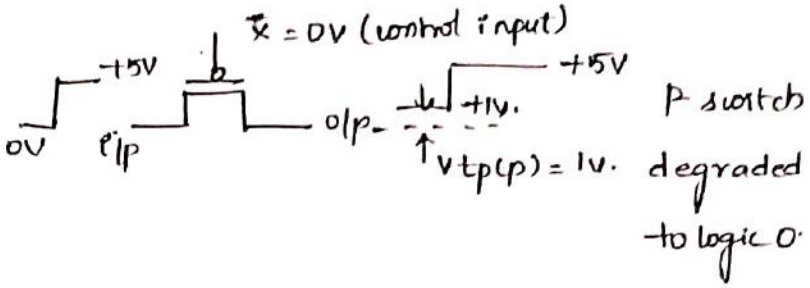
* Switch logic is mainly based on pass transistor or Transmission gate.
* It is fast for small arrays and takes no static current from the supply, V_{DD}. Hence power dissipation of such arrays is small since current only flows on switching.
* Switch (pass transistor) logic is analogous to logic arrays based on relay contacts where no path through each switch is isolated from the logic levels activating the switch.

Pass Transistor:

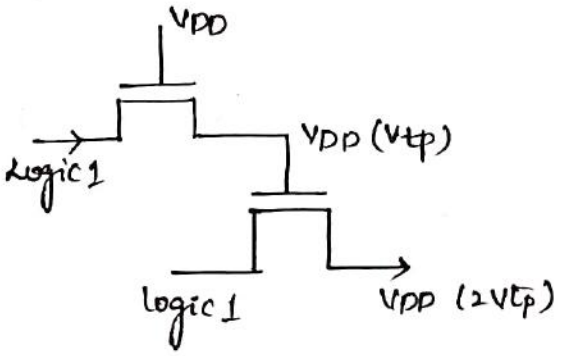
1) This logic uses Transistors as switches to carry logic signals from node to node instead of connecting output nodes directly to V_{DD} or ground (GND).

2) If a single transistor is a switch between two nodes, then voltage degradation equal to V_t (threshold voltage) in high or low level depends up on nMOS or pMOS logic





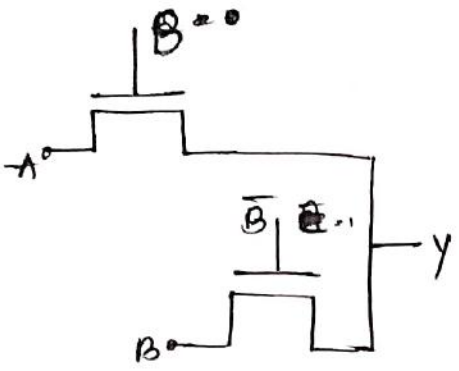
3) When using nMOS switch logic no pass transistor gate input may be driven through one or more pass transistors as shown in figure.



loss of logic 1 if the gate of a pass transistor is driven from another pass-transistor.

4) Since the signal output of pass transistor T_1 does not reach a full logic 1 by threshold voltage effects signal is degraded by below a true logic 1, this degraded voltage would not permit the output of T_2 to reach an acceptable logic 1 level.

Example: Implementation of 2i/p AND gate using pass Transistor logic.



A	B	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

Advantages:

They have topological simplicity

1) Requires minimum Geometry.
2) Do not dissipate stand by power, since they do not have a path from supply to ground.

Disadvantages:

1) Degradation in the voltage levels due to undesirable threshold voltage effects.
2) Never drive a pass transistor with the output of another pass transistor.

Transmission Gate:

1) It is an electronic element, good non-mechanical relay built with CMOS technology.
2) It is made by parallel combination of an NMOS & PMOS transistors with the input gate of one transistor being complementary to the input at the gate of the other as shown in figure.

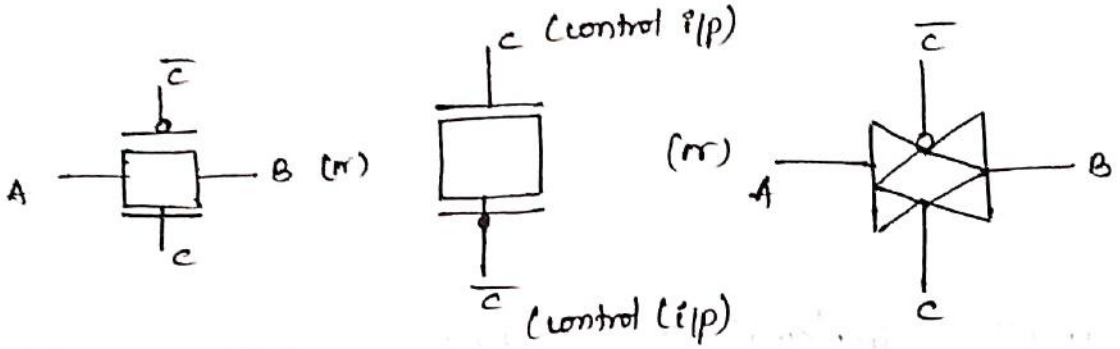
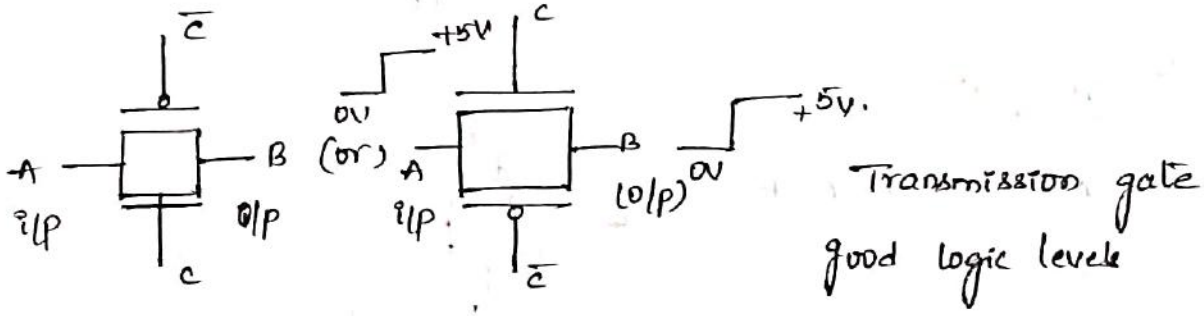


fig: Symbol for Transmission gate.

3) thus current can flow through this element in either direction.

4) Depending on whether or not there is a voltage on the gate, the connection between the input and output is either low resistance or high resistance, respectively $R_{on} = 100\,\Omega$ & $R_{off} > 5M\Omega$.



Transmission gate good logic levels

operation:

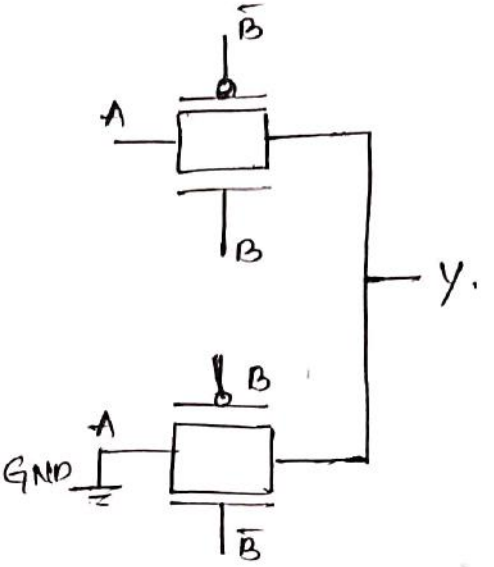
* When the gate input to nMOS transistor is '0' (control i/p c=0) & the complementary '1' is gate input to pMOS ($\bar{c} = 1$), thus both are turned off so i/p & o/p are not equal ($A \neq B$).
* When gate i/p to NMOS is 1 ($c=1$) & its complementary '0' is the gate input to the pMOS, both are turned on and pass any signal '1' (or) '0' at input equal to the output without any degradation ($A = B$).

* The use of transmission gates eliminates the undesirable threshold voltage effects. which gives rise to loss of logic levels in pass transistors.

Example: Circuit implementation using Transmission using TG (logic gates)

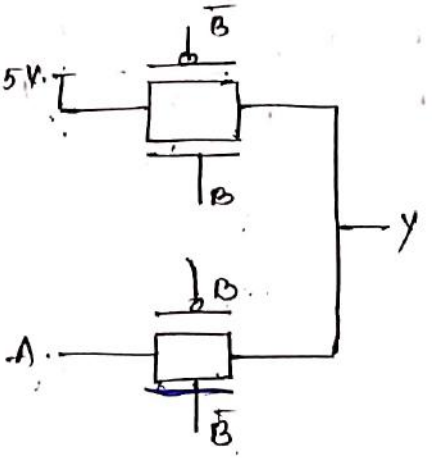
1) AND Gate:

A	B	Y = A·B
0	0	0
0	1	0
1	0	0
1	1	1



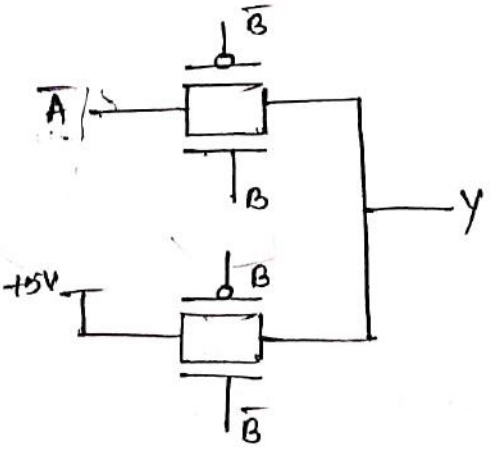
2) OR Gate:

A	B	Y = A+B
0	0	0
0	1	1
1	0	1
1	1	1



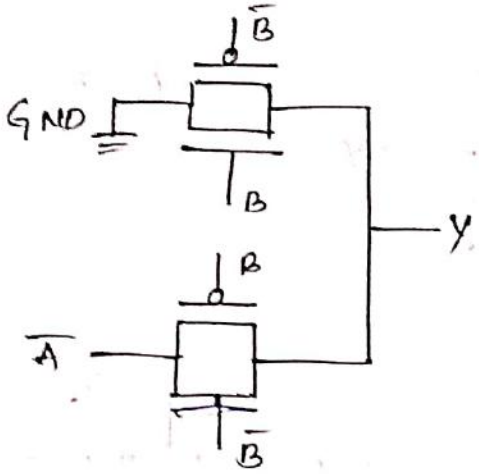
3) NAND Gate:

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



4) NOR Gate:

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



Example-2: $F = \bar{A}B + BC + AC$ draw the Transmission Gate circuit for the boolean function.

A	B	C	AB	BC	AC	F	
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	
0	1	0	0	0	0	0	C
0	1	1	0	1	0	1	
1	0	0	0	0	0	0	C
1	0	1	0	0	1	1	
1	1	0	1	0	0	1	1
1	1	1	1	1	1	1	

$\Rightarrow$

A	B	F
0	0	0
0	1	C
1	0	C
1	1	1

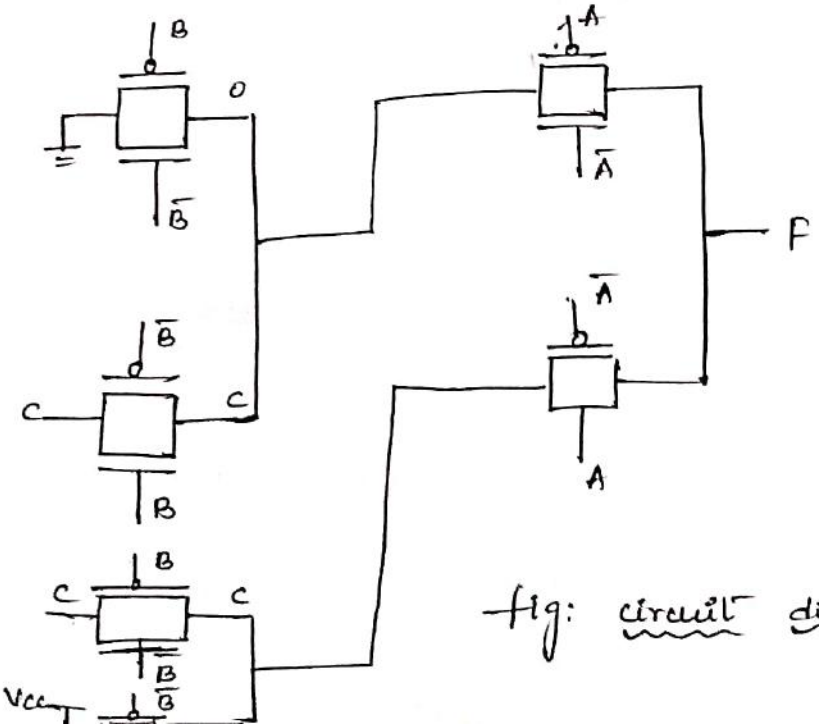
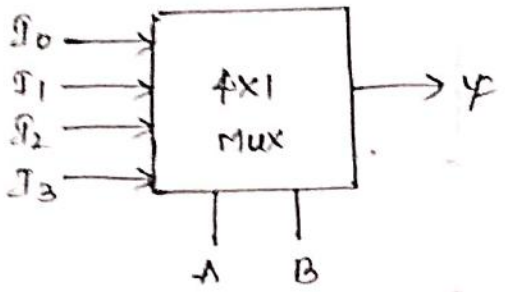


fig: circuit diagram for $F = AB + BC + AC$ using Transmission Gate

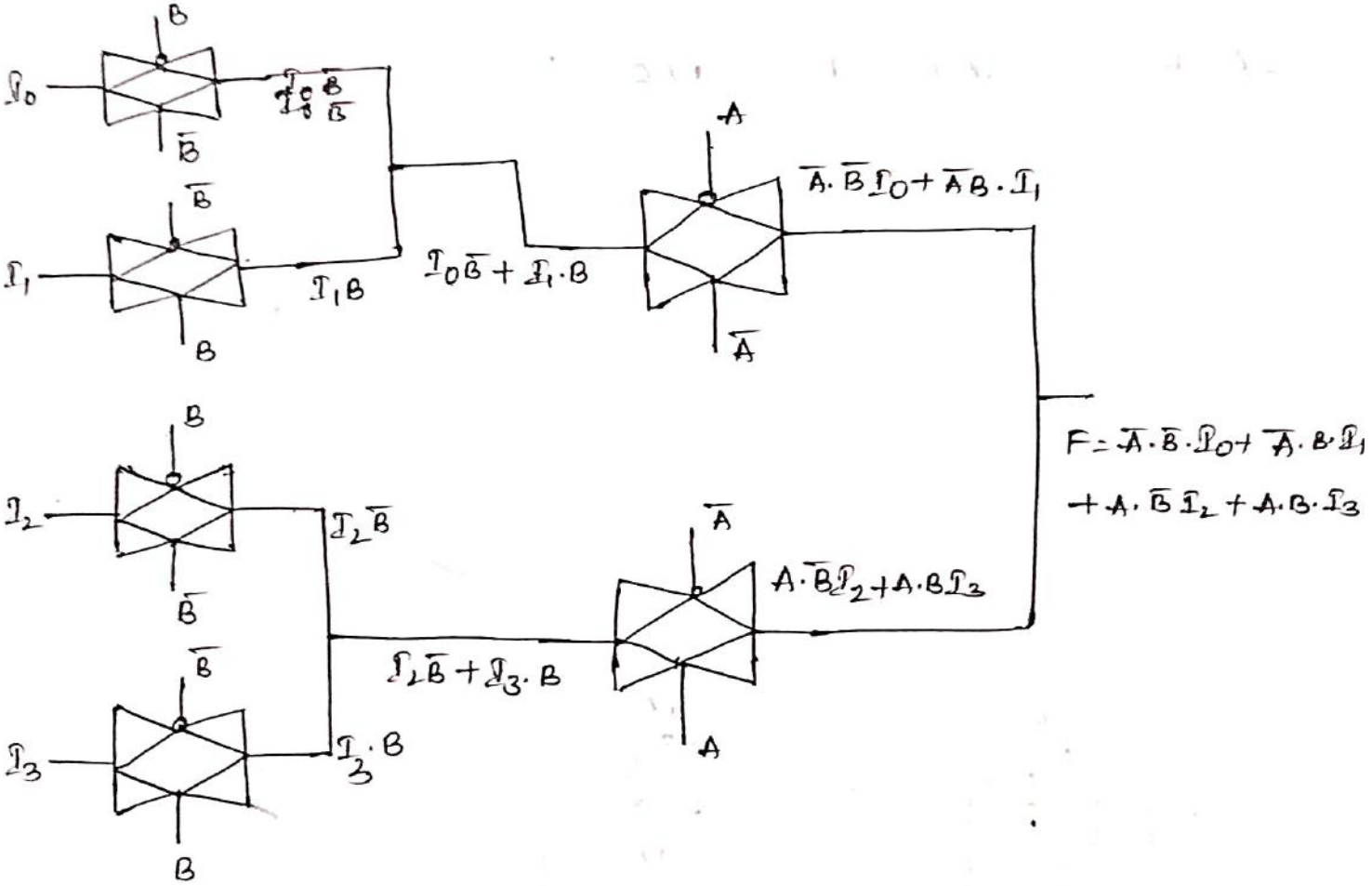
4x1 Multiplexer Implementation using CMOS Transmission Gate.



A	B	F
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Boolean function can be determined from the truth table

$$F = \bar{A}\cdot\bar{B}\cdot I_0 + \bar{A}\cdot B\cdot I_1 + A\cdot\bar{B}\,I_2 + A\cdot B\,I_3$$



Advantages:

1) Transmission gates eliminates the signal degration in the output logic levels.
2) Transmission gate consists of two transistors in parallel and except near the positive and negative rails.

Disadvantages:

1) Transmission Gate requires more area than nMos pass Circuits.

2) Transmission gate requires complemented control signals.

-Alternative Gate circuits:

CMOS suffers from increased area and correspondingly increased capacitance and delay, as the logic gates become more complex. For this reason, designers developed -Alternate gate circuits that can be used to supplement the complementary type circuits. These forms are not intended to replace CMOS but rather to be used in special applications. for special purposes.

-Alternate gate circuits are classified in to

1) pseudo nMos logic.
2) Dynamic CMOS logic.
3) CMOS Domino Logic
4) clocked CMOS Logic
5) n-p CMOS Logic.

1) PSEUDO NMOS Logic:

Pseudo nMOS logic is one of the alternate gate ckt that is used as a supplement for the CMOS logic circuits. In the pseudo logic, the pullup n/w (PUN) is realized by a single pMos transistor. The gate Terminal of pMos

Transistor is connected to the ground. It remains permanently in the ON state. Depending on the input combinations, output goes low through the PDN.

* Here only nMOS logic (Q_n) is driven by i/p voltage while the gate of p-transistor (Q_p) is connected to ground or substrate and Q_p acts as an active load for Q_n. Except for the load device, the pseudo n-MOS gate circuit is identical to the pulldown network (PDN) of the CMOS gate.

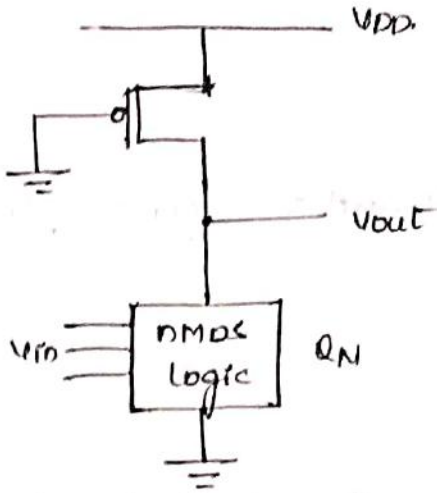
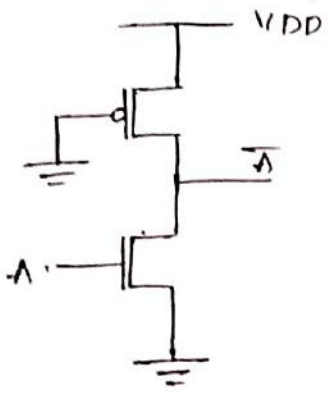


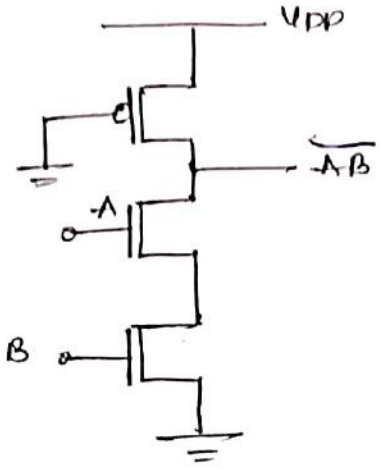
fig: General building block of logic ckts that follows.

Pseudo nMOS logic

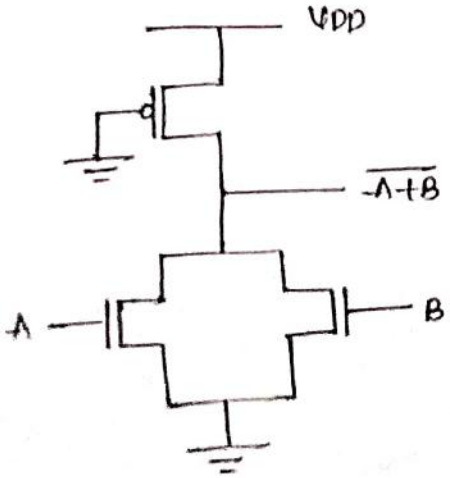
The realization of logic circuits using pseudo nMOS logic as shown in figure.



(a) pseudo nMOS Inverter



(b) pseudo nMOS NAND.



Advantages:

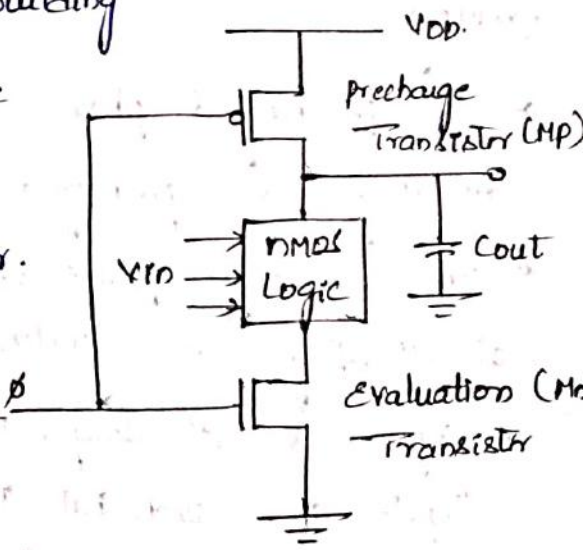
1) Uses less number of transistors as compared to CMOS logic.
2) Geometrical area and delay gets reduced as it requires less transistors.
3) Low power dissipation.

Disadvantages:

1) The main drawback of using pseudo nMOS gate instead of a CMOS gate is that the always on pMOS load conducts a steady current when the output voltage is lower than V_{DD}.

Dynamic CMOS Logic:

A dynamic CMOS logic uses charge storage and clocking properties of MOS transistors to implement logic operations. below figure shows the building block of dynamic CMOS logic. Here the clock ϕ drives nMOS evaluation transistor and pMOS precharge transistor. A logic is implemented using an nFET array connected between output node and ground.



* The gate (clock ϕ) defines two phases, evaluation and precharge phase during each clock cycle.

Working:

* When clock $\phi = 0$, the ckt is in precharge phase with PMOS (Mp) ON & the Evaluation nMOS (MN) off. This Establishing a conducting path between VDD & output allowing cout to charge to a voltage vout = VDD. Mp is often called precharge FET.
* When clock $\phi = 1$ the ckt is in Evaluation phase with PMOS device Mp off & Evaluation nMOS (MN) ON. If the block acts like a closed switch cout can discharge through logic array & MN. So vout = VDD, this is logically an output of F = 1. Charge leakage eventually drops output to Vout = 0V which could be an incorrect logic value.

The logic formation is formed by 3 series connected FET's

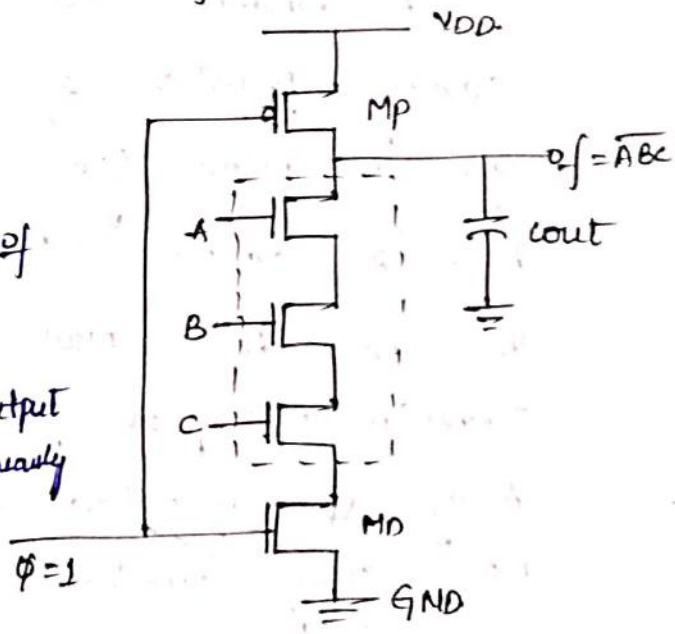


fig: 3 input NAND Gate

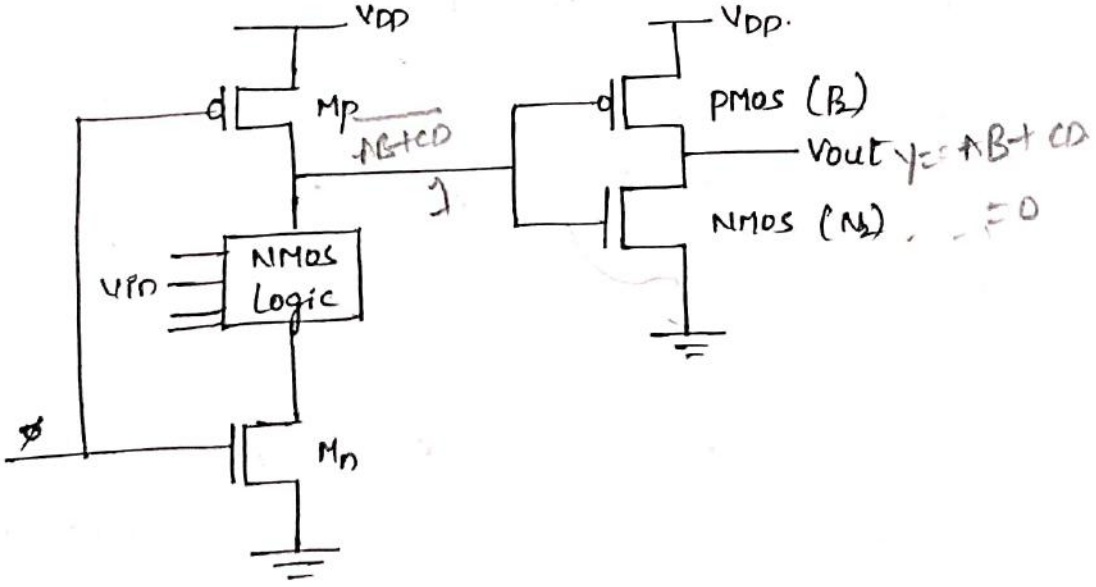
Working:

* In precharged phase ($\phi = 0$), output of all stages are precharged to logic high.
* In evaluation phase ($\phi = 1$), the output of all stages are evaluated simultaneously
* Suppose in 1st stage, the output Inputs are such that the output is logic low after the Evaluation
* In second stage, the o/p of 1st stage is one input and so on. Such that o/p of 1st discharges to logic low. then the Evaluated o/p of 1st stage can never make the o/p of second stage logic high. Since Evaluation happens simultaneously. the o/p cannot be charge to logic high

CMOS Domino Logic:

Standard CMOS logic gates need a PMOS and an NMOS Transistor for each logic input. The PMOS Transistors require a greater area than the nMOS Transistors carrying the same current. So, a large chip area is necessary to perform complex logic operations. The package density in CMOS is improved if a dynamic logic ckt, called the domino CMOS logic circuit is used.

Domino CMOS logic is slightly modified version of the dynamic CMOS logic circuit. In this case, a static Inverter is connected at the output of each dynamic CMOS logic block. The addition of the Inverter solves the problem of cascading of dynamic CMOS logic circuits.



A domino (logic) CMOS AND-OR gate that realizes the function $Y = AB + CD$ is depicted in above figure. The left hand part of circuit containing M_p, M_n, T_1, T_2, T_3 & T_4 form and AND-OR-Inverter gate (AOI). right side It derives the static CMOS formed by N_2 and P_2.

* The domino gate is activated by single phase clock ϕ applied to NMOS (Mn) & pMOS (Mp) Transistors. The load on AOI part of ckts is the parasitic load capacitance.

Working:

* When $\phi = 0$, Mp is ON and Mn is OFF, so that no current flows in the AND-OR paths of AOI. The capacitor C_L is charged to V_{DD} through Mp since the latter is ON. The i/p to the Inverter is high & drives the output voltage Vout = logic '0'.

* When $\phi = 1$, Mp is OFF & Mn is ON, If either (or both) A&B or C&D is at logic-1, C_L discharges through either T_1, T_2 & Mn (or) T_3, T_4 and Mp. So, the Inverter Input is driven to logic '0'. & hence the output voltage Vout = logic'1 The boolean expression for o/p voltage is $Y = AB + CD$.

Note: Logic input can change only when $\phi = 0$, & NO changes of i/ps are permitted when $\phi = 1$ since a discharge path may occur.

Advantages:

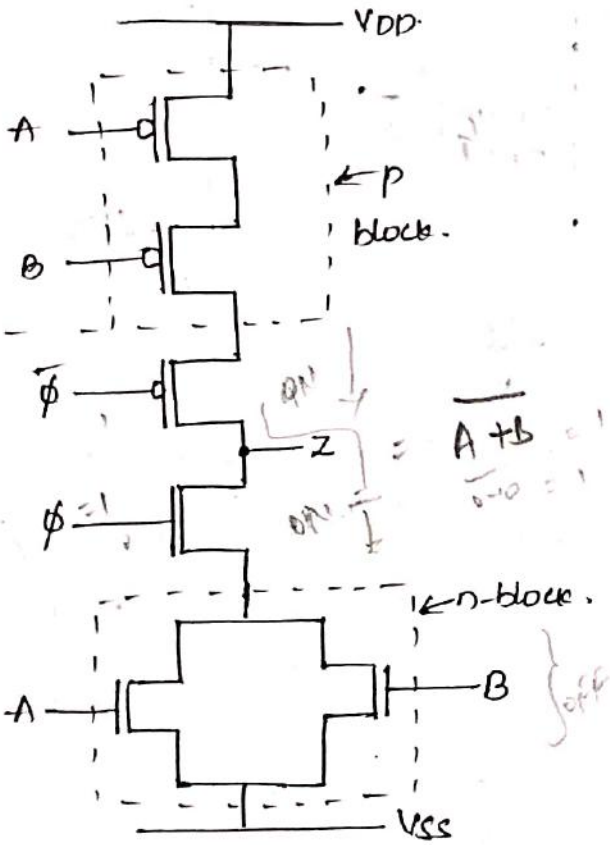
1) Smaller areas compared to conventional CMOS logic
2) parasitic capacitances are smaller so, that higher operating speeds are possible.
3) operation is free of glitches since each gate can make one transition

disadvantages:

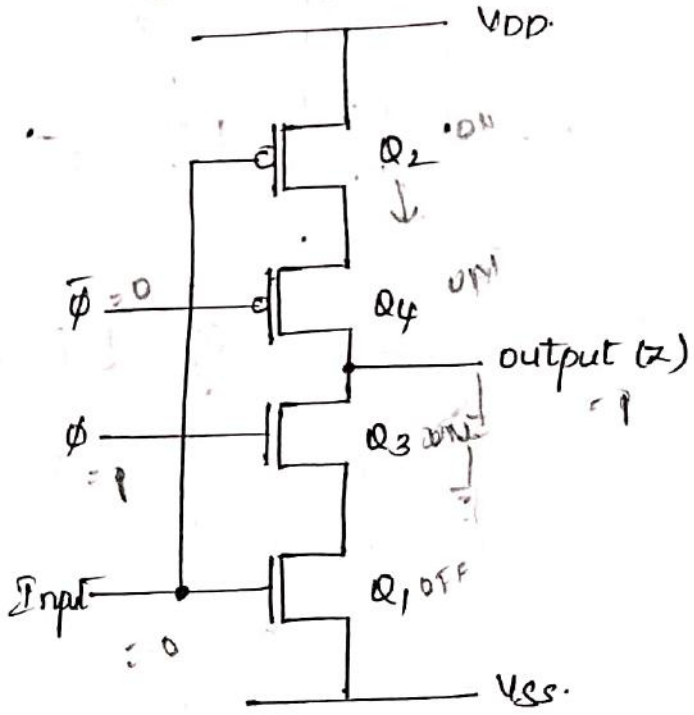
1) Non Inverting structures are possible because of presence of Inverting buffer.
2) charge distribution may be a problem.

clocked CMOS Logic (C²MOS Logic):

The clocked CMOS logic is also reffered as C²MOS logic. the below figure shows the general arrangement of C²MOS logic. A pullup p-block & a complementary n-block pull down structure represent p&n Transistors respectively. & are used as Implement clocked CMOS logic shown in figure. However, the logic in this case is connected to the output only during the ON period of clock. The slower rise times & fall times can be expected due to owing of Extra Transistors in series with the output.



fig(a): Implemented CMOS logic



fig(b): clocked Inverter ckt.

Working:

* When $\phi = 1$, the ckt acts as Inverter, because Transistors Q_3 & Q_4 are 'ON' It is said to be in Evaluation mode. Therefore the output z changes its previous value.

-* When $\phi = 0$, the circuit is in hold state (or) Mode, because Transistor Q_3 & Q_4 are off. It is said to be in "precharge mode". Therefore the output '2' remains its previous value.

D-P CMOS logic:

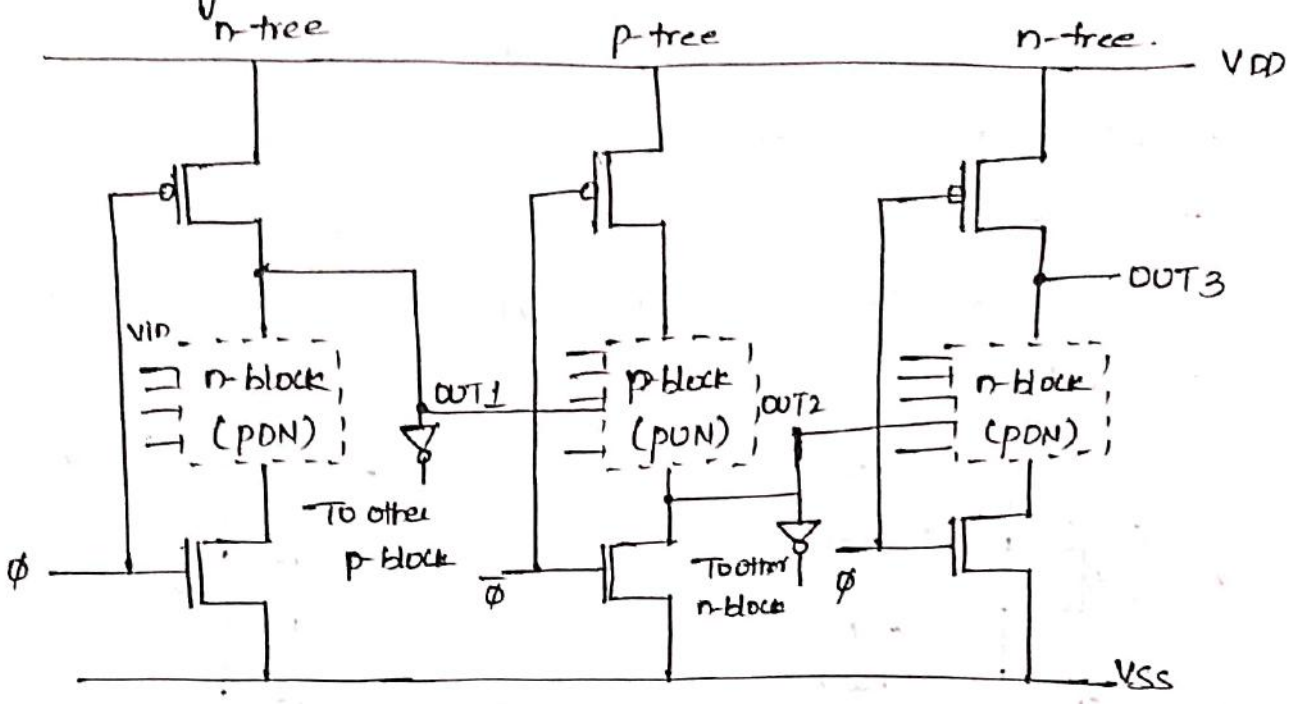


fig: D-P CMOS logic.

* In this logic the actual logic blocks are alternatively 'n' & 'p' in a cascaded structure. The clock ϕ & $\bar{\phi}$ are used alternatively to feed the precharge & evaluate Transistors. However, the functions of top & bottom Transistors are also alternate b/n precharge & evaluate Transistors

Working:

-* During the precharge phase $\phi = 0$, the output of n-tree gate, out1, out3 are charged to VDD, while the output of p-tree gate OUT2 is precharged to 0V. Since the n-tree gate connects pMOS pull-up devices, the PUN of p-tree is turned off at that time.

* During the Evaluation phase $\phi = 1$, the outputs (OUT_1, OUT_3) of the n-tree gate can only make a 1-0 Transition. conditionally turning on some transistors in p-tree. This ensures that no accidental discharge of OUT_2 can occur.

* Similaly n-tree blocks can follow p-tree gates without any problems, because the inputs to n-gates are precharged to '0'.

Disadvantages:

* Here the p-tree blocks are slower than the n-tree modules, due to the lower current drive of p-MOS Transistors in the logic network.

Physical Design

In Integrated circuit design, physical design is a step in the standard design cycle which follows after the circuit design. At this step, circuit representations of components (devices & Interconnections) of the design are converted into Geometric representation of shapes, which when manufactured in the corresponding layers of materials, will Ensure the required function of the components. This Geometric representation is called Integrated circuit layout. This step is usually split in to several sub steps, which include both design & verification & validation of the layout.

* In Modern days Integrated circuit (Ic) design is split up in to. Front end design using HDL's, & verification and Back End design or physical design.

* The physical design is the manufacturing process or Fabrication process that is done in wafer fabrication Houses. Fab houses fabricate designs on to silicon dies which are then packed in to Ic's.

* Each of the phases mentioned above has design flow associated with them. physical design flow uses the technology libraries that are provided by fabrication houses. these technology files provide Information Regarding the type of silicon wafer used, the standard cells used, the layout rules (like DRC in VLSI) etc.,

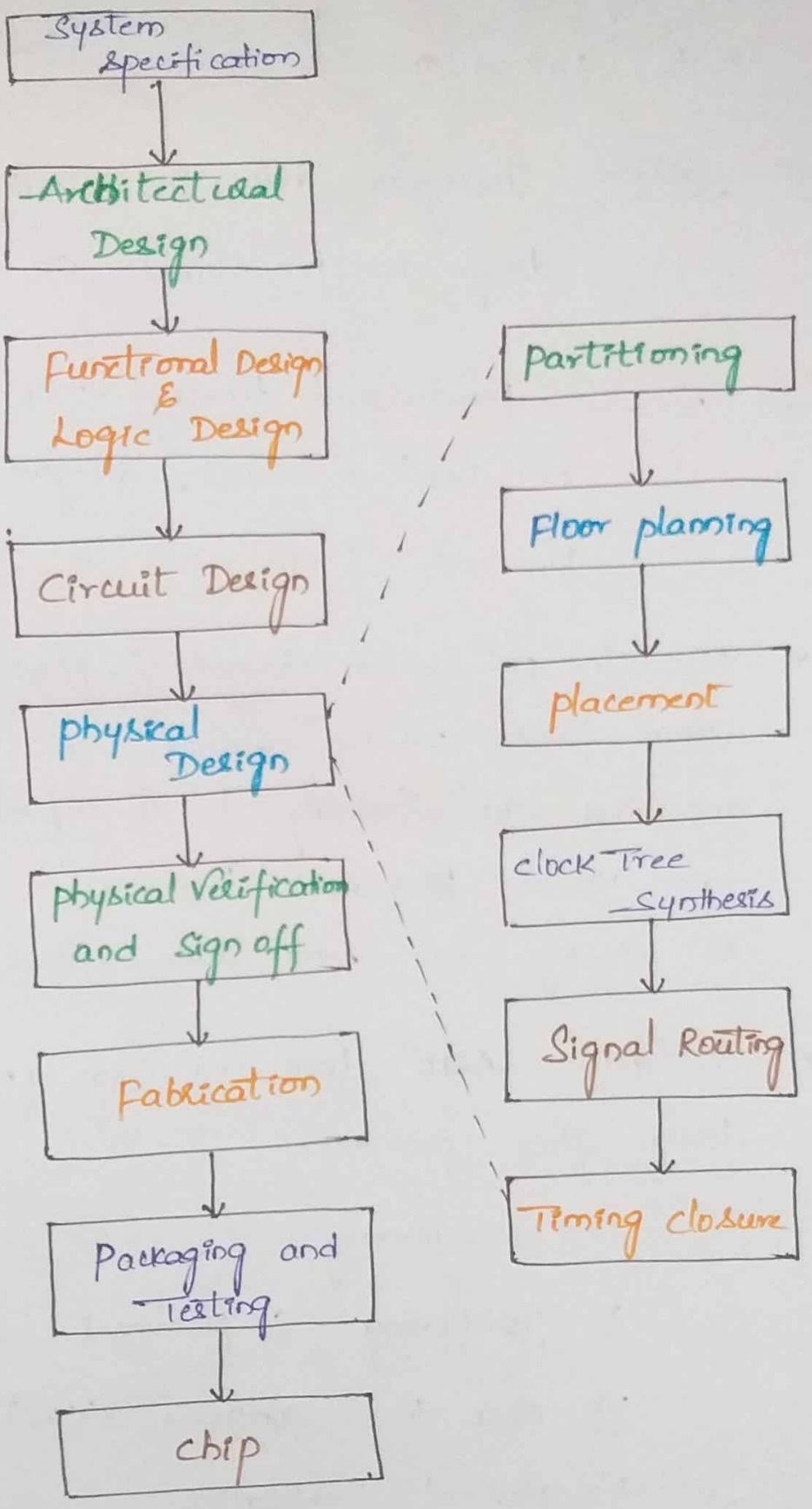


figure: physical design steps within the Ic Design flow

* Typically, the Ic physical design is categorized into full custom & semi-custom Design.

Full-custom: Designer has full flexibility on the layout design, no predefined cells are used.

Semi-custom: predesigned library cells (preferably tested with DFM) are used, designer has flexibility in placement of the cells & routing.

* ASIC for full custom design & FPGA for semi-custom design flows can be refered. The reason being that one has the flexibility to design/modify design blocks from Vendor provided libraries in ASIC. This flexibility is missing for semi-custom flows like FPGA (Eg. Altera

* A typical ASIC back End flow (or) ASIC physical design flow contains.

1) floor planning
2) partitioning & placement.
3) clock tree synthesis (CTS) Routing
4) physical Verification.

Floor planning:

* Floor planning is chip level layout design
* Floor planning is the process of placing blocks in the chip area, there by determining the routing area between them.
* Floor plan determines the size of die & creates wire tracks for placement of standard cells. It creates power & Ground connections. It is also determines the Input/output pad placement Information.

* A good floor planning should meet the following constrains.

1) minimize the total chip area.
2) make routing phase Easy (routable).
3) Improve the performance by reducing signal delays.

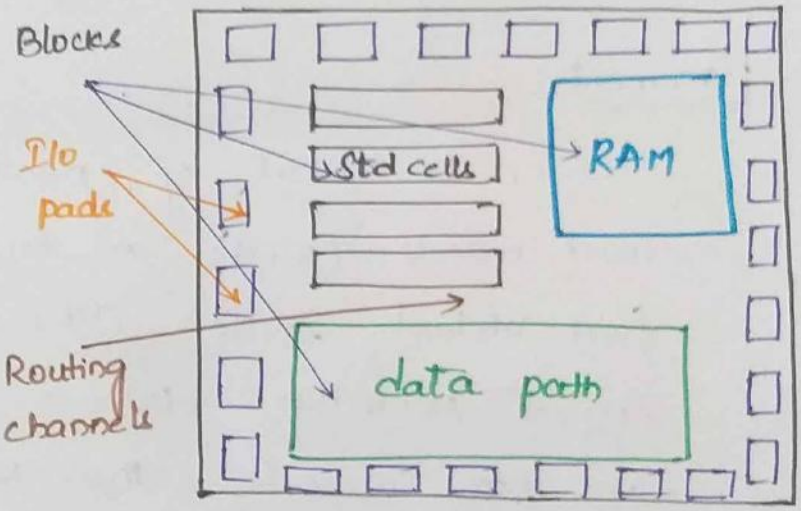


figure: floor planning.

* The Goals of floor planning are.

1) Arrange the blocks on a chip.
2) Decide the location of the I/O pads.
3) Decide the location & number of power pads.
4) Decide the type of power distribution.
5) Decide the location & type of clock distribution

partitioning :

Partitioning is a process of dividing the chip in to small blocks. This is done mainly to seperate different functional blocks & also to make placement & routing Easier. partition can be done in the RTL design phase when the designer engineer partitions the entire design in to sub blocks. & then proceeds to design Each module. these modules are linked together in the main module called Top level module. This kind of partitioning is commonly reffered to as logical partitioning

placement :

Before the start of placement optimization all wire load models (WLM) are removed. placement uses RC values from virtual route (VR) to calculate time. VR is the shortest Manhattan distance between two pins. VR RC's are more accurate than WLM RC's.

Placement is performed in 4 optimization phases.

1) preplacement optimization: It optimizes the netlist before placement, It can also downsize the cells.

2) Inplacement optimization: It reoptimizes the logic based VR. This can perform cell sizing, cell moving, cell bypassing, net splitting, gate duplication, buffer Insertion, area recovery, optimization performs Iteration of setup fixing, Incremental timing & congestion driven placement.

3) post placement optimization: before clock Tree Synthesis: (CTS) perform netlist optimization with ideal clocks. It can fix setup, hold, max trans/ cap violations. It can do placement optimization based on global routing. It reduces High fanout Net (HFN) Synthesis.

4) post placement optimization after CTS: It optimizes timing with propagated clock. It tries to preserve clock skew.

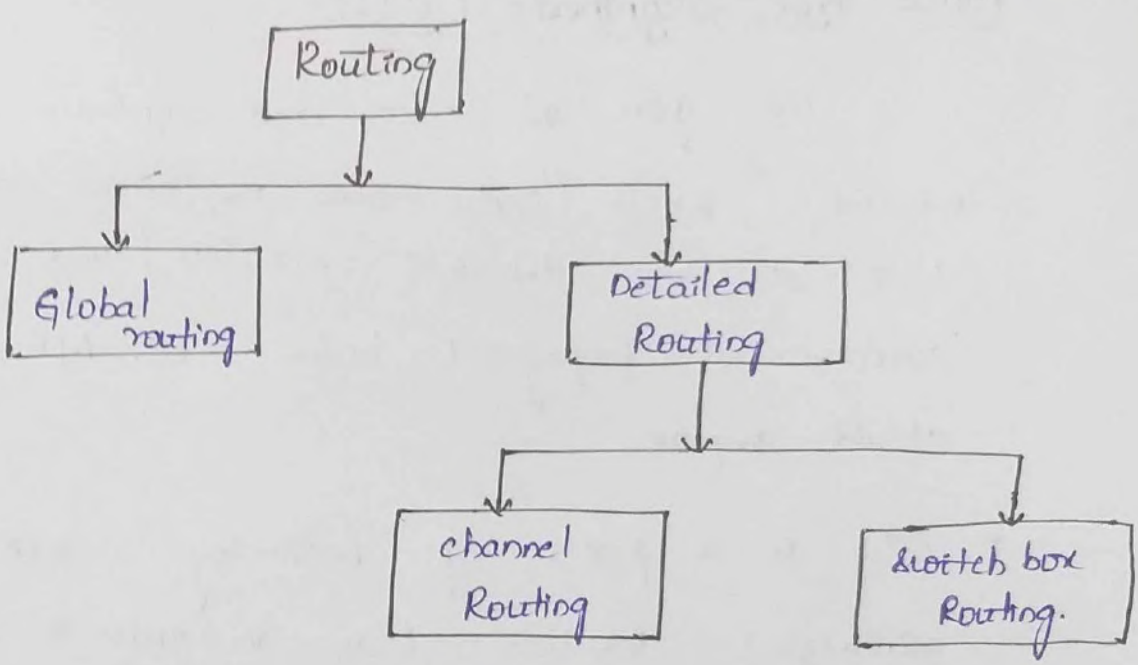
clock Tree Synthesis (CTS):

The goal of 'clock Tree Synthesis (CTS) is to minimize skew [skew means Difference in clock arival time at two different registers] and Insertion delay. clock is not propagated before CTS. After CTS hold slack should Improve.

* CTS is a process of balancing clock skew and maximizing Insertion delay in order to meet timing, power & other requirements
* It is a process of distributing clock signal to clock pins based on physical Information.

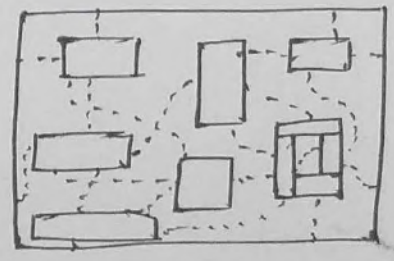
Routing:

* The physical connections are made by routing.
* Routing is to provide the connections between the blocks.
* It is to locate a set of wires in routing space so as to connect all the nets in the netlist.
* The objective of routing is to minimize total wire length.
* There are types of approaches are used in routing.



1) Global Routing:

Generate a loose route for each net assign a list of routing region to each net without specifying the actual layout of wires.



2> Detailed Routing:

To find the actual Geometry layout of each net with in the assigned routing regions

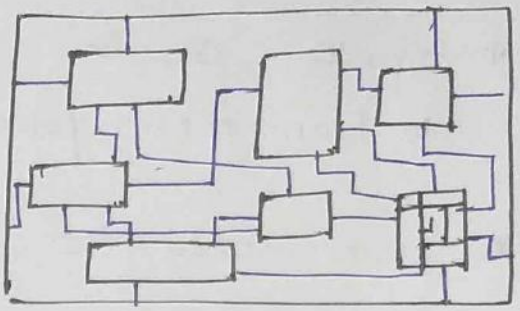
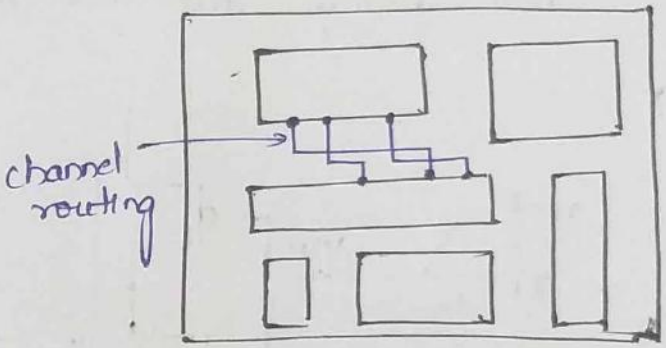


fig: Detailed Routing

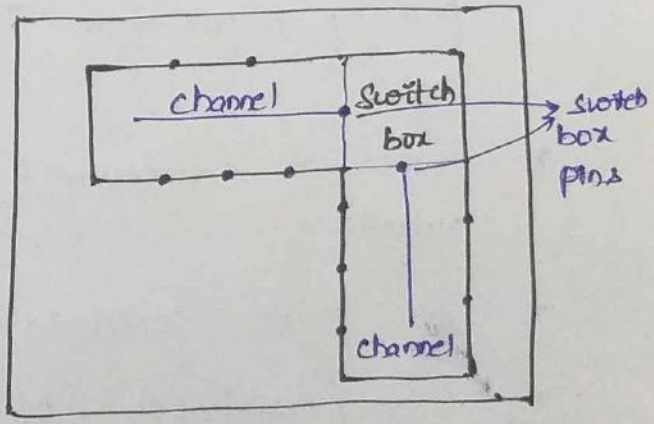
i) Channel Routing:

Channel routing is a special case of the routing problem in which wires are connected within the routing channels.



ii) Switch box routing:

Switch box routing is harder than channel routing because we can't expand the switch box to make the room for more wires.



Power delay Estimation:

* Analyzing the power consumption of an inverter provides an alternate care in to performance of a logic gate.
* circuits can be made to go faster by causing them to burn more power.
* power consumption always come at the cost of heat which must be dissipated out of the chip.
* To analyze an Inverter with capacitor connected to its outputs.
* To analyze the power consumption we must consider both pullup and pull down operation.

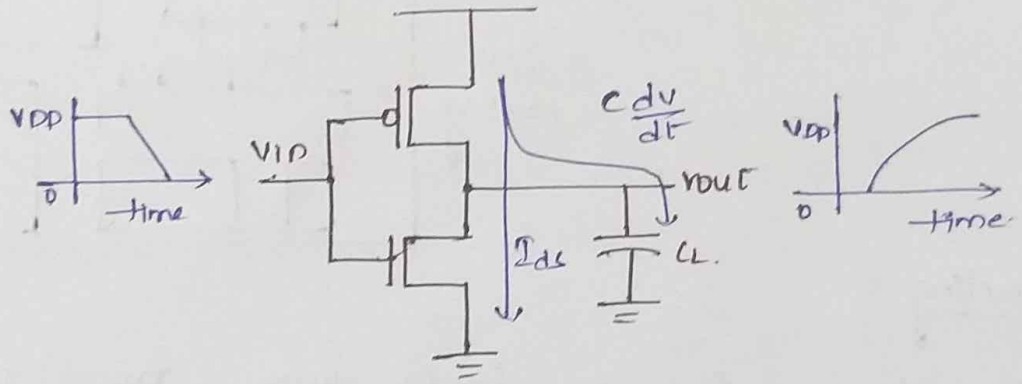


fig: Inverter ckt

* The C_L is charged or discharged is depends upon the pull up & pull down Transistors & only one of Transistors is 'ON'.
* power is consumed when the gate is driven these outputs.
* The power consumed by the Inverter is independent of sizes / resistances of pull up & pull down Transistor. but power consumption depends on size of capacitive loads at outputs.

* the energy required to drive the Inverter output high is calculated in two ways.

1) current through the load capacitance C_L

2) current through pull up transistor & represented as effective resistance R_p.

the current through the C_L is represented as

$$I_{CL}(t) = \frac{V_{DD} - V_{SS}}{R_p} e^{-(t/R_pC_L)} \rightarrow (1)$$

the voltage across the capacitor is

$$V_{CL}(t) = (V_{DD} - V_{SS}) \left[1 - e^{-(t/R_pC_L)}\right] \rightarrow (2)$$

Energy required to charge the capacitor.

$$E_C = \int_0^\infty I_{CL}(t)\, V_{CL}(t)\, dt \longrightarrow (3)$$

Substitute eqns (1) & (2) in (3) we get.

$$E_C = \int_0^\infty \frac{V_{DD} - V_{SS}}{R_p} e^{-(t/R_pC_L)} \cdot (V_{DD} - V_{SS}) \left[1 - e^{-(t/R_pC_L)}\right]$$

$$E_C = \int_0^\infty \frac{(V_{DD} - V_{SS})^2}{R_p} e^{-(t/R_pC_L)} \cdot \left[1 - e^{-(t/R_pC_L)}\right]$$

$$E_C = \frac{1}{2} C_L (V_{DD} - V_{SS})^2 \longrightarrow (4)$$

the current through the pull up transistor is

$$I_p(t) = I_{CL}(t). \longrightarrow (5)$$

the voltage across pull up Transistor is

$$V_p(t) = V. e^{-(t/R_pC_L)} \longrightarrow (6)$$

Energy required to charge capacitor as computed from a resistor (R_p) point of view

$$E_R = \int_0^{\infty} i_p(t)\, V_p(t)\, dt \rightarrow (7)$$

Substitute Eq (5) & (6) in (7) we get

$$E_R = \frac{1}{2} C_L (V_{DD}-$$

$$E_R = \int_0^{\infty} \left(\frac{V_{DD}-V_{SS}}{R_p}\right) e^{-(t/R_pC_L)}\, V \cdot e^{-(t/R_pC_L)}$$

$$E_R = \frac{1}{2} C_L (V_{DD}-V_{SS})^2 \rightarrow (8)$$

* Energy consumed in discharging the capacitor can also be calculated in the same way. the discharging energy consumption is equal to charging power consumption

$$E_C = \frac{1}{2} C_L (V_{DD}-V_{SS})^2 \rightarrow (9)$$

* A single cycle requires the capacitor to both charge & discharge so the total energy consumption is from Eq (4) & (9) we get.

$$E_T = E_R + E_C = C_L (V_{DD}-V_{SS})^2$$

Unit-4 SUBSYSTEM DESIGN

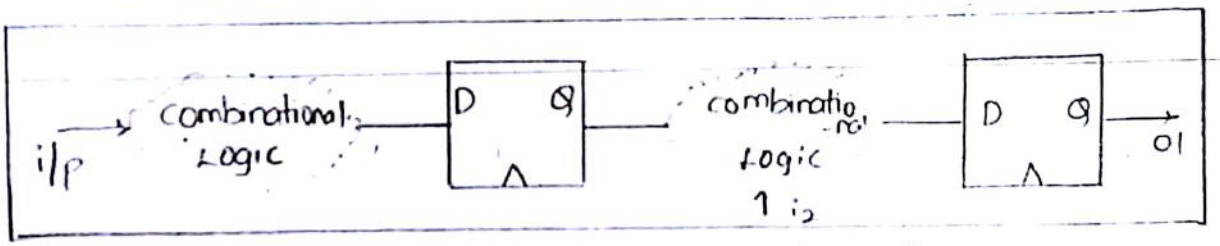
Part-A

Subsystem Design Principles:- ①

Pipelining

→ Pipelining is a relatively simple method for reducing the clock period of long combinational operations.

→ pipelining allows large combinational functions to be broken up into pieces whose delays are in balance with the rest of the system components.



structure of pipelined system.

→ Pipelining entails introducing a rank of memory elements along a cut through the combinational logic

→ We usually want all the outputs to appear on the same clock cycle, which implies that the cut must divide the inputs and outputs into disjoint sets.

→ The pipelined system still computes the same combinational function but that function requires several cycles to be computed.

Scanned by CamScanner

→ The ② number of cycles between the presentation of an input value and the appearance of its associated output is the latency of the pipeline.

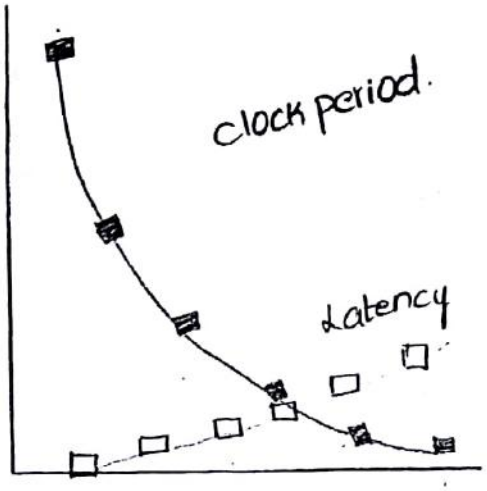


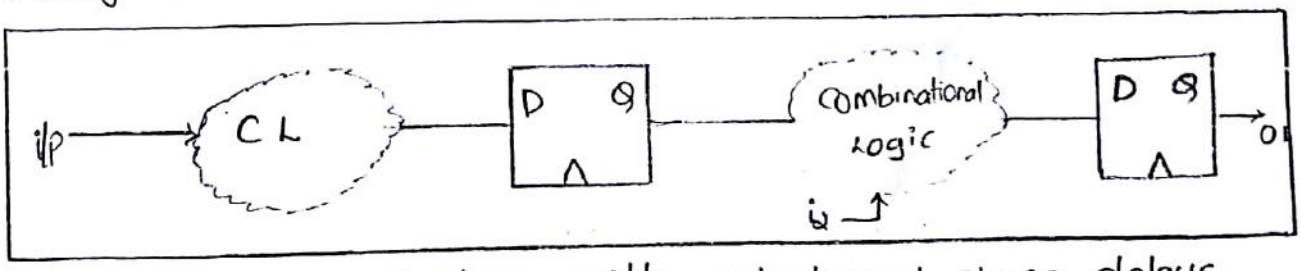
Fig:- clock period and latency as a function of pipeline depth.

→ Pipeling cause a great reduce in clock period, but heavy decreased gains.

→ Latency increases linearly.

→ The total delay through the Pipelined system is slightly longer than the combinational delay.

→ When pipelining, memory elements can be placed along any cut, the best placement balances delays through the combinational logic.



Pipeline with unbalanced stage delays.

→ if memory elements are placed so that some delay paths are much longer than others.

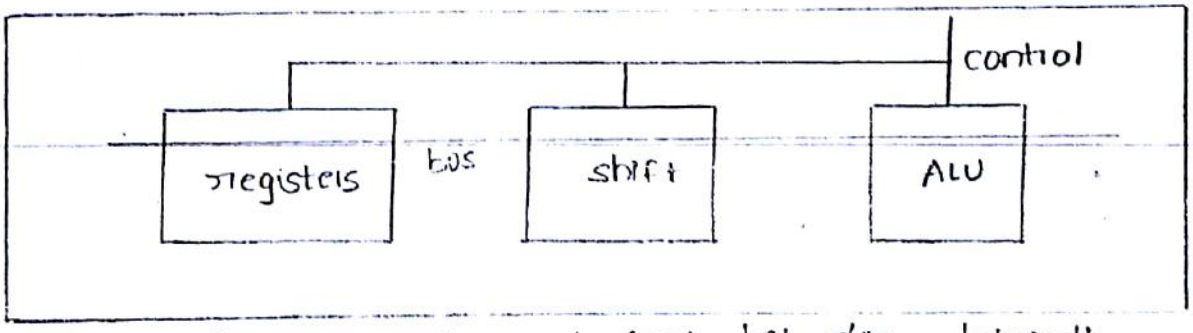
→ Perfect pipeling balances the delay between ranks of memory elements.

Scanned by CamScanner

Paths:-

A datapath is both logical and physical structure, it is built from components which perform typical data operations, such as addition and it has a layout structure which takes advantage of the regular logical design of the data operators.

→ Datapath include several types of components: registers (memory elements) store data, adders and ALUs perform arithmetic, shifters perform bit operations, counters may be used for program counters.



Structure of a typical bit-slice datapath.

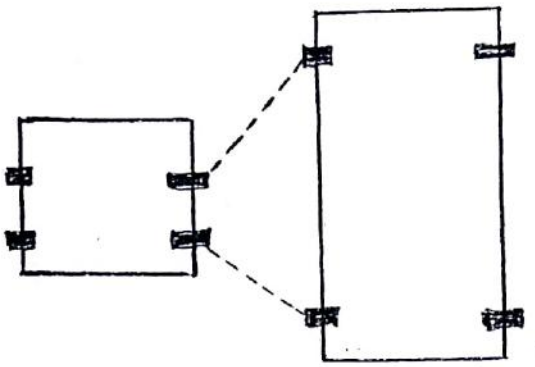
→ Data is often passed between components on one or mae busses, or common connections; the number of buses determines the maximum number of data transfers on a clock cycle and is a primary design parameter of data paths.

→ A bit slice is a one-bit version of the complete datapath and the n-bit data path is constructed by replicating the bit-slice.

→ data flows horizontally through the bit-slice along point-to-point connections or buses, while control

Scanned by CamScanner

Signals flow Vertically.



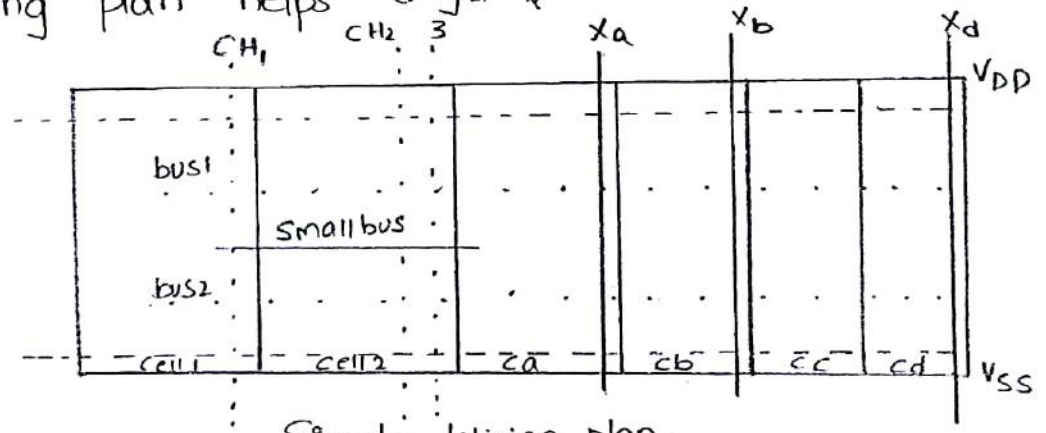
A butting cells may require moving pins & streching.

→ cells often need to be stretched beyond their natural heights to make connections with other cells.

→ The data path's byout design also requires careful consideration of layer assignment.

→ With a process that provides two levels of metal, metal-1 is used for horizontal wires and metal-2 for Vertical Wires.

→ Wiring plan helps organize the wires required.



Simple Wiring plan.

→ TWO ckt design problems unique to data path design are registers and buses.

→ The ckt choosen for the register depends on the number of registers required. If only a few registers are needed, a standard flip-flop is a perfectly reasonable choice.

→ If many registers are required, area can be saved by using an n-port static RAM, which includes one row enable, and one pair of bit lines for each port:

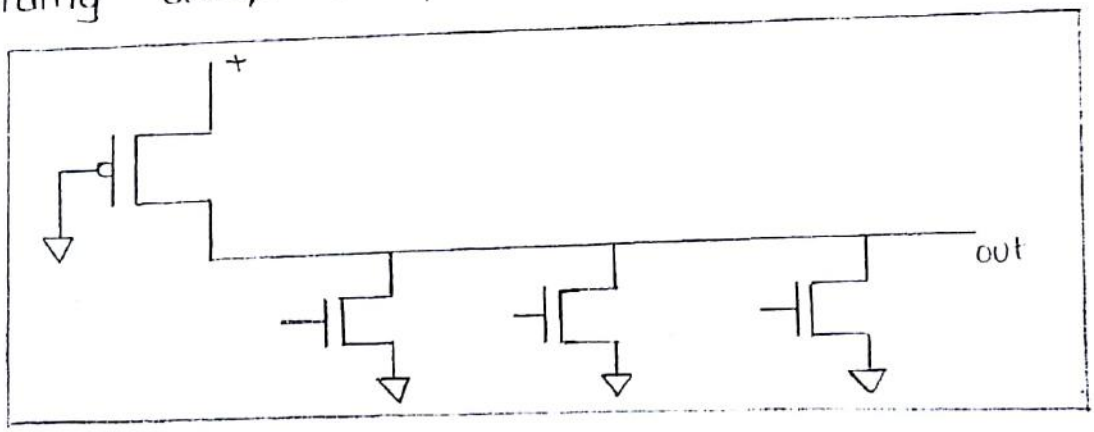
Scanned by CamScanner

→ Although an individual bit/word can allow only one read or write operation at a time, the (5) RAM array can support the simultaneous. independent reading or writing of two words simply by setting the select lines of the two words high at the same time.

→ Data paths are almost always made with busses that provide fewer connections at much less cost in area.

→ The system probably doesn't need all data path components to talk to each other simultaneously, connections can be shared.

→ But these shared connections often require special ckts that take up a small amount of space while providing adequate speed of communication.



Pseudo nmos bus Circuit

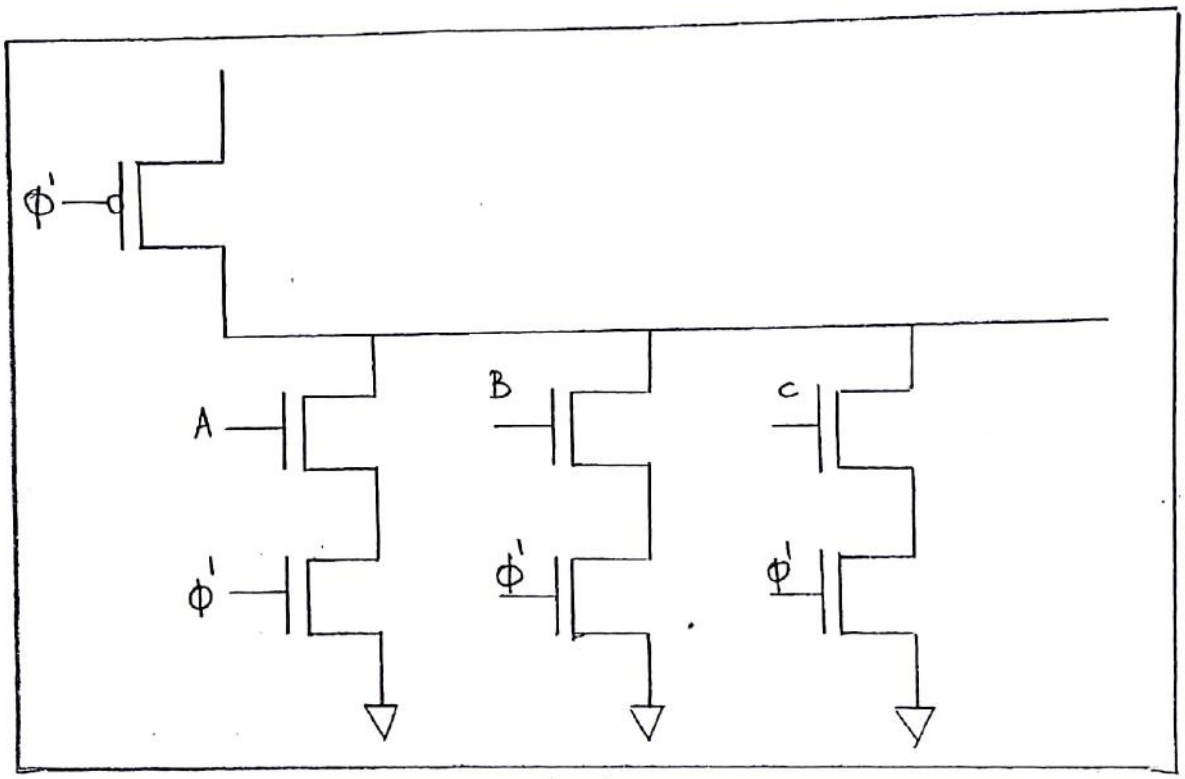
→ ckt design for a bus is as a distributed NOR gate:

→ The common wire forms the NOR gate's output, while pull downs at the sources select the source and set the NOR gate's o/p.

Scanned by CamScanner

→ Pseudo-nmos shown in the above and precharged shown in the below fig. The trade off's are also similar.

→ the pseudo-nmos bus is slow but does not require a seperate precharge phase.

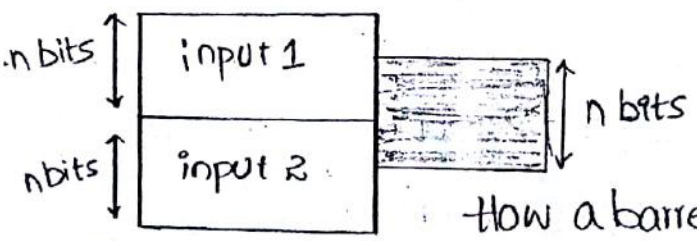


A precharged bus ckt.

Combinational Shifters:-

→ A shifter is most useful for arithmetic operations since shifting is equivalent to multiplication by powers of two.

→ Simplest shifter is the shift register - which shift by one position per clock cycle.



How a barrel shifter performs shifts & rotates

Scanned by CamScanner

→ A barrel shifter can perform n-bit shifter in a single combinational function and it has a very efficient layout. Length of shift is determined by position of selected column.

→ It can rotate and extend signs as well.

→ The barrel shifter accepts 2n data bits and n control signals and produces n o/p bits.

→ It shifts by transmitting an n-bit slice of the 2n data bits to the o/p.

→ The position of the transmitted slice is determined by the control bits, the exact operation is determined by the values placed at the data input.

→ A barrel shifter with n output bits is built from a 2n Vertical by n horizontal array of cells, each of which has a single transistor & a few-wires.

→ The core of the cell is a transmission gate built from a single n-type transistor

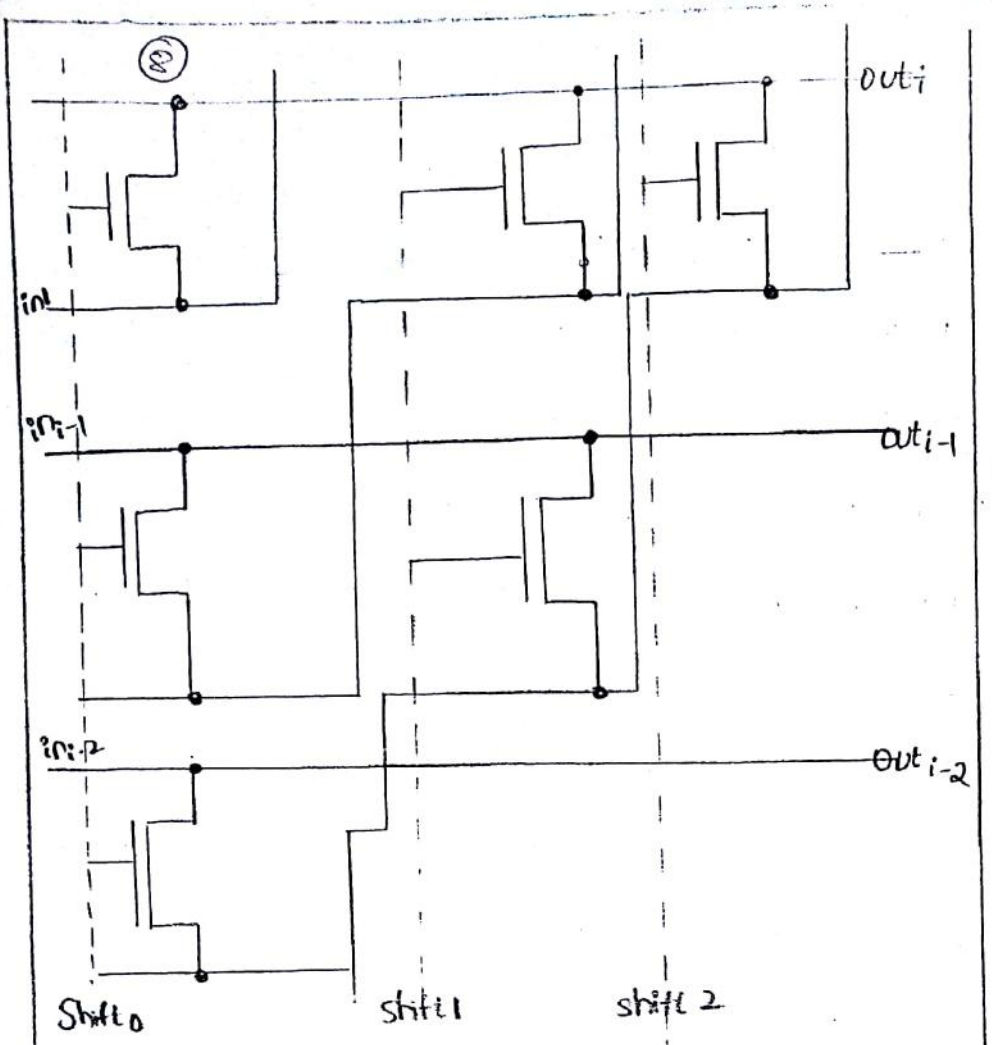
→ The control lines runs vertically, i/p data run diagonally & horizontal o/p wires.

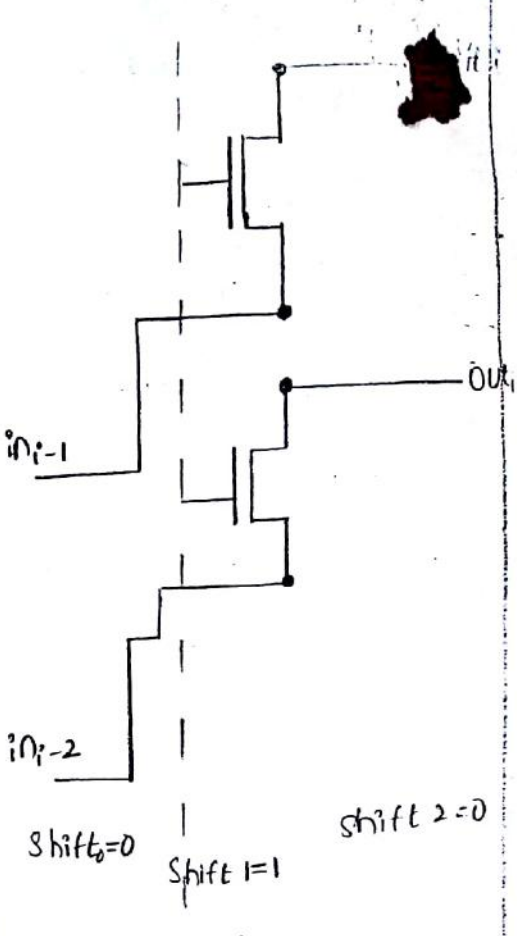
→ When a column is turned on, all inputs are shunted to the o/p's.

→ The control line values are set so that exactly one is 1, which turns all the transmission gates in a single column. Delay of barrel shifter is largely determined by parasitic capacitances on wires.

→ the transmission gates connect the diagonal input wires to the horizontal o/p wires.

Scanned by CamScanner





Operation

structure

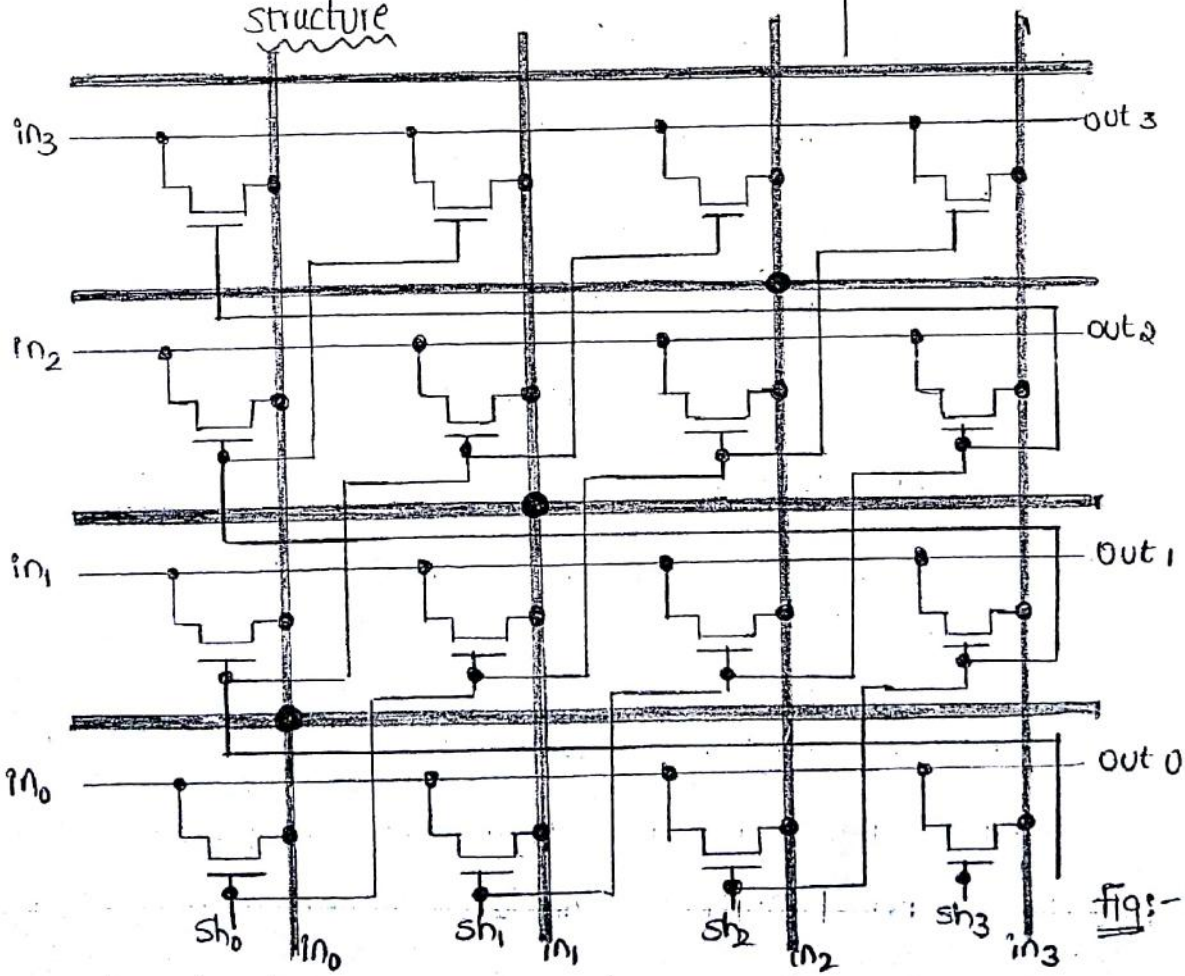


fig:- 4X4 barrel shifter

sh0 -

sh1 -

-sh2 -

sh3 -

Scanned by CamScanner

Adders:-

→ The adder is a digital circuit & the basic adder is a fulladder.

→ It computes a one-bit sum and carry from two addends and a carry-in.

→ Equations for a full adder

$$S_i = a_i \oplus b_i \oplus c_i$$

$$C_{i+1} = a_i b_i + a_i c_i + b_i c_i$$

S_i → sum at the i^{th} stage

C_{i+1} → carry out of the i^{th} stage

→ the n bit adder is built from n one-bit full adders is known as ripple carry adder.

→ the addition is not complete until the $n-1^{th}$ adder has computed its S_{n-1} output; that result depends on q i/p and so on down the line.

→ The ripple carry adder is area efficient and easy to design but it is slow when n is larger.

→ Speeding up the adder requires speeding up the carry chain.

→ The carry-look ahead adder is one way to speed up the carry computation.

→ The carry look ahead adder breaks the carry computation into two steps with two intermediate values.

→ the adder inputs are a_i's and b_i's in which p(propagate) & g (Generate) are computed.

Scanned by CamScanner

$$P_i = a_i + b_i$$

$$G_i = a_i \cdot b_i$$

If $G_i = 1$ there is definitely a carry out of the i^{th} bit of the sum a carry is generated.

If $P_i = 1$ then carry from $i-1^{th}$ bit is propagated to the next bit.

$$\therefore S_i = C_i \oplus P_i + G_i$$

$$C_{i+1} = G_i \oplus P_i C_i$$

On recursive expanding.

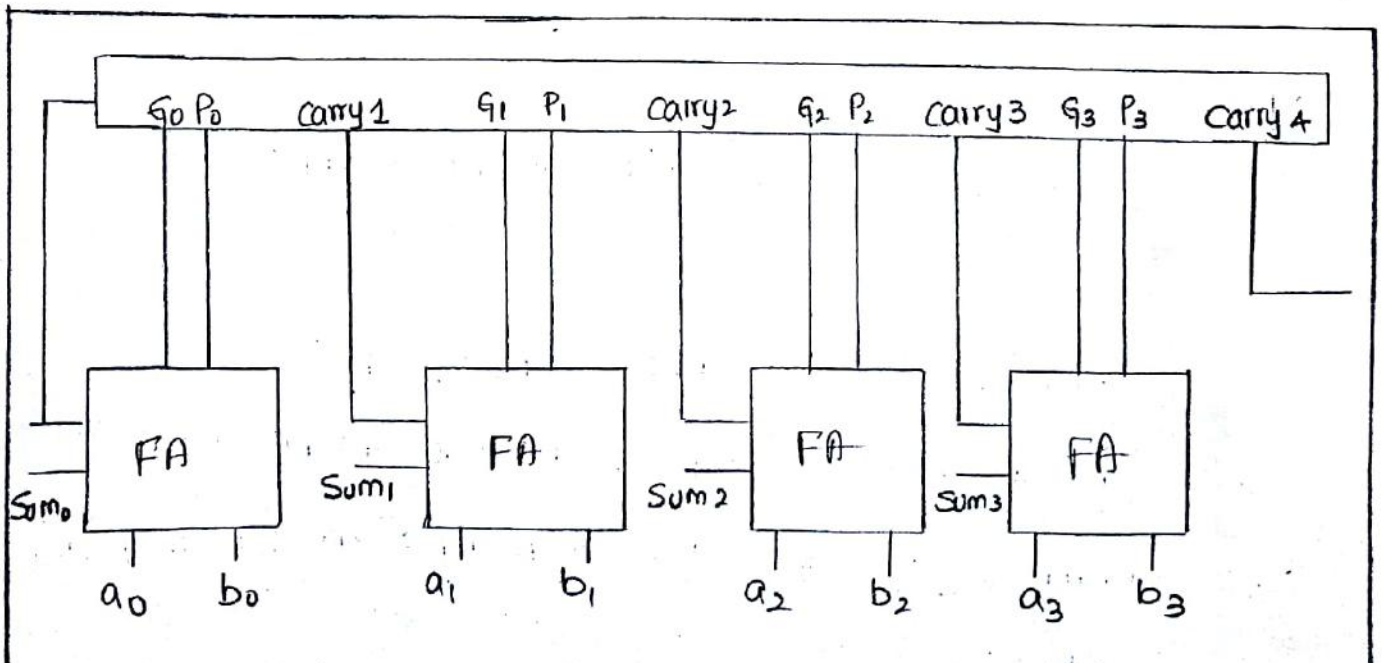
$$C_{i+1} = G_i + P_i (G_{i-1} + P_{i-1} \cdot C_{i-1})$$

$$= G_i + P_i G_{i-1} + P_i P_{i-1} (G_{i-2} + P_{i-2} \cdot C_{i-2}).$$

$$= G_i + P_i G_{i-1} + P_i P_{i-1} G_{i-2} + P_i P_{i-1} P_{i-2} C_{i-2}.$$

→ There is a limit beyond which the larger gates are slower than chains of smaller gates, typically 4 levels of carry can be usually expanded.

→ A depth -4 carry look ahead unit is



structure of a carry look ahead adder.

Scanned by CamScanner

→ The unit takes P & G values from its four associated adders and computes four carry values. Each carry o/p is computed by its own logic.

→ The carry-look ahead units can be recursively connected to form a tree. each unit generates own P & G values which are used to feed the carry-look ahead unit at the next level of the tree.

→ A simpler scheme is to connect the carry-in's and carry-outs of the units in a ripple chain.

→ The wiring of or the carry-look ahead tree is hard to design and area consuming

ALU's

Design of an ALU Subsystem

* Having designed the shifter, we may now turn our attention to another subsystem of the 4-bit data path. A convenient and appropriate choice is the ALU.

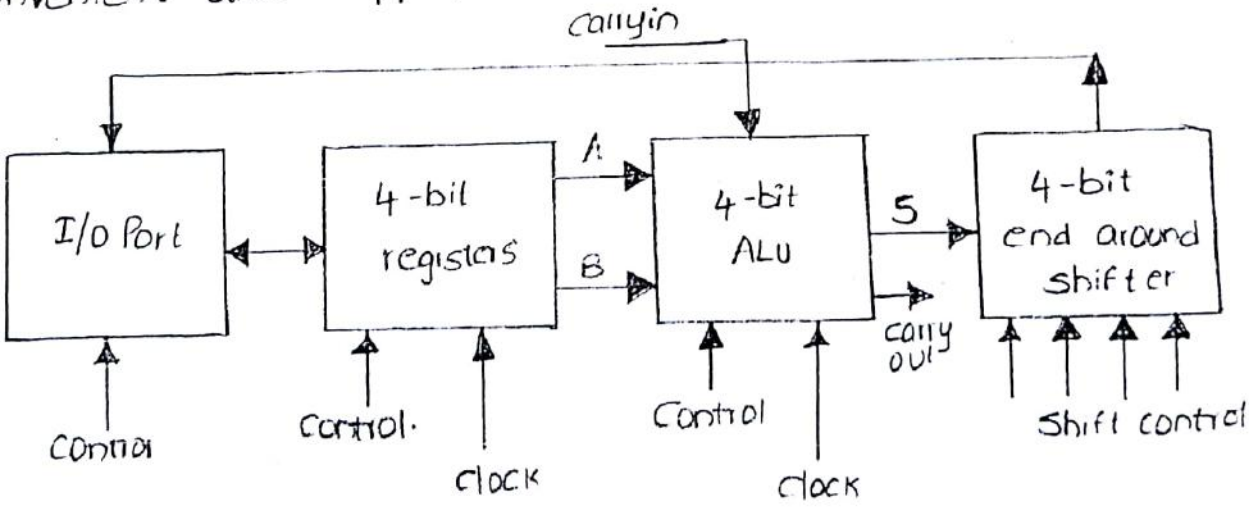


Fig:- 4-bit data path for processor (block diagram)

* The hearth of the ALU is a 4-bit adder circuit and it is

Scanned by CamScanner

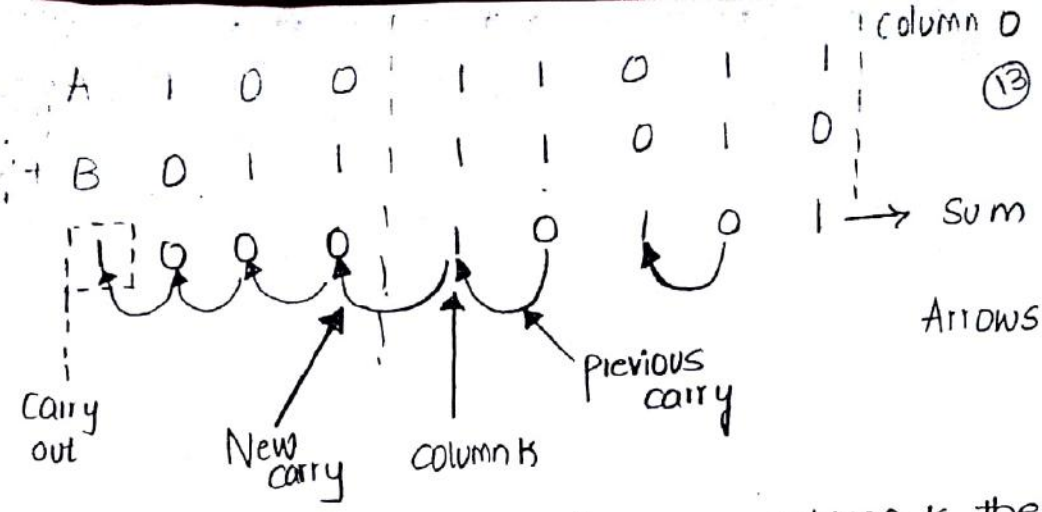
this which we will actually design, indicating later how it may be readily adapted to subtract and perform logical operations.

* A 4-bit adder must take the sum of two 4-bit numbers, & it will be seen we have assumed that all 4-bit quantities are presented in parallel form and that the shifter ckt has been designed to accept and shift a 4-bit parallel sum from the ALU.
* Let us now specify that the sum is to be stored in parallel at the output of the adder from where it may be fed through the shifter and back to the register Array.
* Thus, a single 4-bit data bus is needed from the adder to the shifter and another 4-bit bus is required from shifted o/p to register array.
* As far as the i/p to the adder is concerned, the 2 4-bit parallel numbers to be added are to be presented in parallel on two 4-bit buses.
* Clock signals are required by the ALU as shown.
* The shifter is unclocked but must be connected to four shift control lines. It is also necessary to provide a "carry out" sig from the adder and, in the general case, to provide for a possible "carry in" signal, as indicated in above fig.

Design of a 4-bit Adder

⇒ Inorder to derive the requirements for an n-bit adder, Let us first consider the addition of two binary numbers A+B as follows:

Scanned by CamScanner



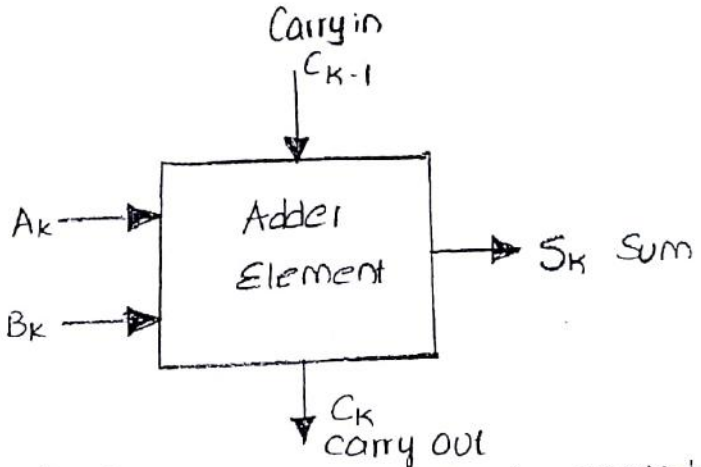
(13)

Arrows indicate 'carry 1'

* It will be seen that for any column k there will be 3 i/p's the corresponding bits of the i/p numbers, A_k and B_k, and the 'previous carry' - Carry in (C_{k-1}). It will also be seen that there are 2 o/p's, the sum (S_k) & a new carry (C_k).

A standard adder element

⇒ A 1-bit adder element may now be represented as in fig. Note that any number of such elements may be cascaded to form any size of adder and that the element is quite general.



n such elements would be cascaded to form an n-bit adder

Fig:- Adder element

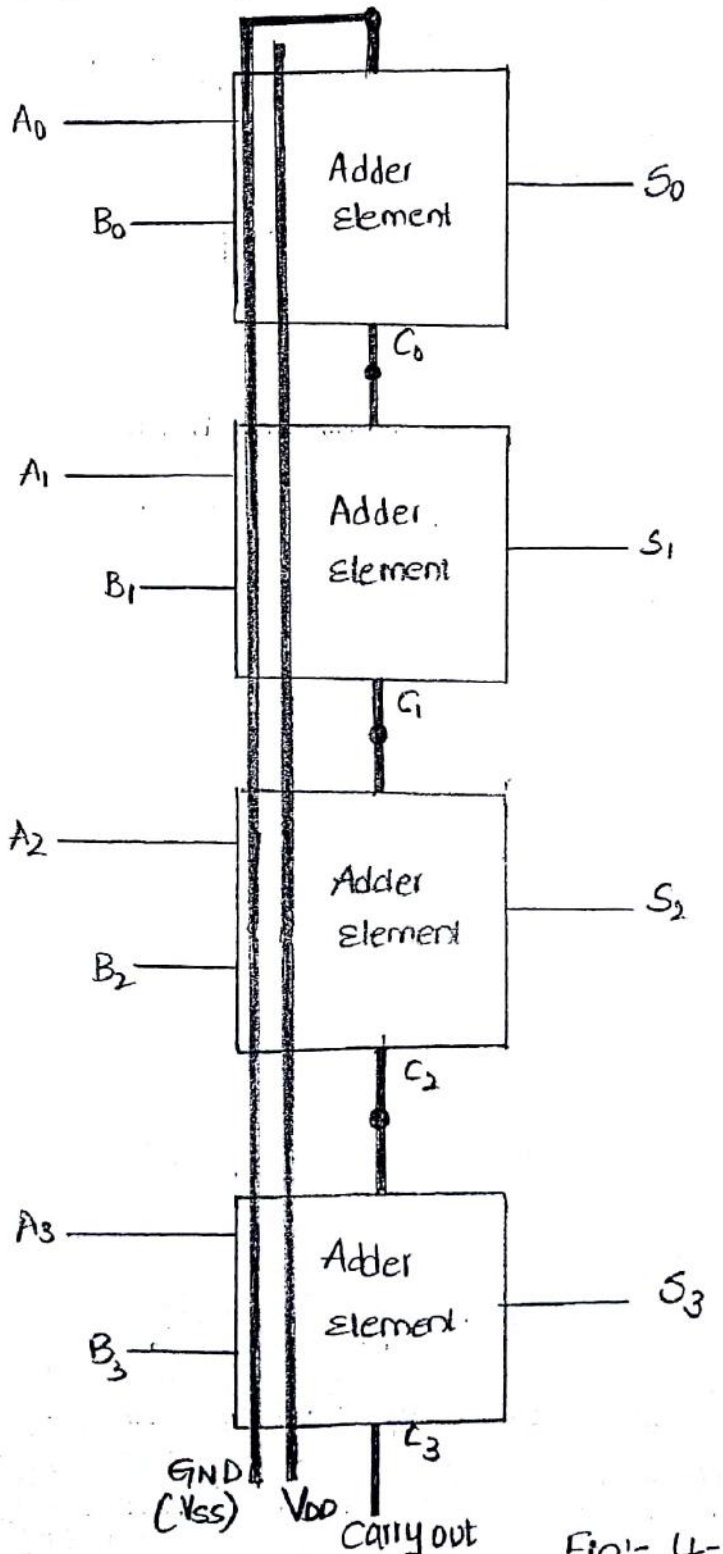
* Note also that this standard adder element may itself be composed from a number of replicated subcells. Regularity and generality must be aimed at in all levels of architecture.

Scanned by CamScanner

4-bit Adder

⇒ 4-bit adder is then formed by cascading four adder elements as indicated in fig:- and an initial assessment of the minimum floor plan area requirement follows from 4-bit adder bounding box.

* This is the second subsystem of floor plan, the first being the barrel shifter.



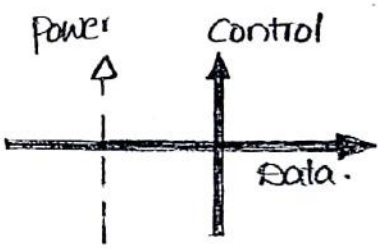


Fig:- 4-bit adder

Scanned by CamScanner

MULTIPLIERS

A study of computer arithmetic processes will reveal that the most common requirements are for addition and subtraction, but that there is also a significant need for a multiplication capability. Although division is obviously useful, it is a much less common requirement and will not be dealt with in this text.

The serial-parallel Multiplier:-

* This multiplier is the simplest one, the multiplication being considered as a succession of additions.

* Multipliers basic algorithm for multiplication is.

* one digit of the multiplier by the full multiplicand is multiplied. Add the result, shifted by the proper number of bits to the partial product.

* Single digit of binary multiplication of two bits is performed by the AND function.

```
      0 1 1 0  multiplicand
    x 1 0 0 1
    ---------
      0 1 1 0
  + 0 0 0 0
  ---------
    0 0 1 1 0
+ 0 0 0 0
---------
  0 0 0 1 1 0
+ 0 1 1 0
---------
  0 1 1 0 1 1 0
```


Scanned by CamScanner

* One simple way to implement basic multiplication algorithm is serial - parallel multiplier.

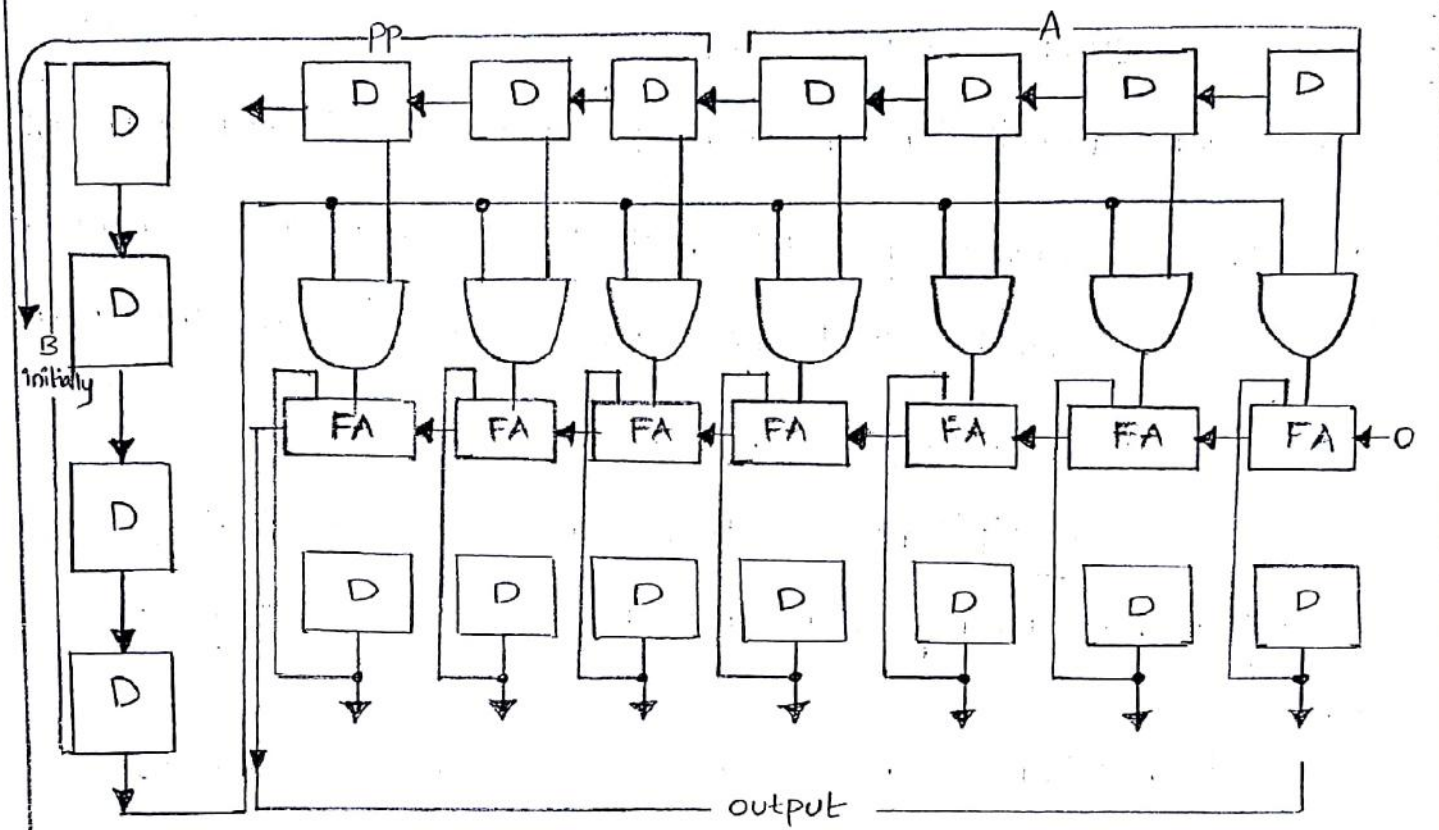


Fig: Arrangement of a 4-bit Serial-Parallel multiplier

* Note that D indicates a D flip-flop simple and FA indicates a full adder - or adder bit slice.

* Number A is entered in the right-most 4-bits of the top row of D flipflops which are connected to three further D flip-flops to form a 7-bit shift register to allow the multiplication of number A.

* In some cases, it may be easier to right shift the contents of the Accumulator rather than left shifting A. This approach can be used to eliminate the least significant bits of the product if so desired.

Scanned by CamScanner

* A further reduction in hardware can result from noting that the three most significant bits of the partial product are set to zero initially, and are used only one bag one as the shifting of A proceeds.

* The structure under discussion here is suited only to positive or unsigned operands. If the operands are negative and twos complement encoded then:

1. The most significant bit of B will have a negative weight and so subtraction must be performed as last step

2. the most significant bit of A must be replicated since operand A must be expanded to 2N bits.

The Braun Array.

*A relatively simple form of parallel adder is the Braun array. All partial products $A \cdot b_K$ are computed in parallel, then collected through a cascaded array of carry save adders. At the bottom of the array, an adder is used to convat the carry save form to the required form of output.

* Completion time is fixed by the depth of the array, and by the carry propagation characteristics of the adder.

* Notice that this multiplier is suited only to positive operands. Negative operands can be handled.

For example: Baugh-Wooley multiplier.

Scanned by CamScanner

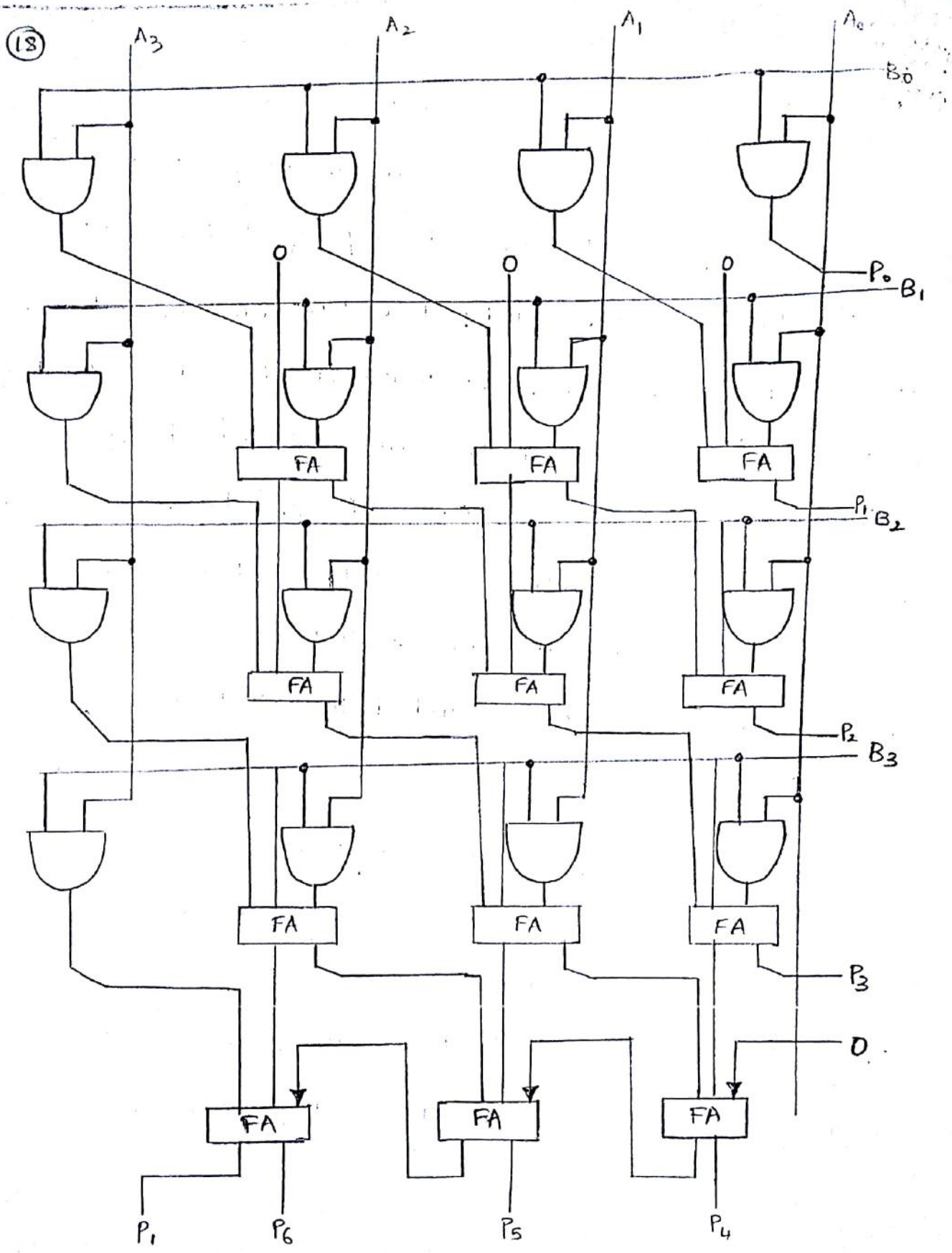


Fig:- A 4-bit Braun Multiplier

Scanned by CamScanner

Multiplication of 2's complement signed numbers

Baugh - Wooley multiplier: (A)

* It maximizes the regularity of the multiplier logic & allows all the partial products to have positive sign bits.

* The multiplier X can be written in binary as.

$$X = x_{n-1} 2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i$$

Where $n \rightarrow$ no. of bits in representation

$y \rightarrow$ multiplicand

$P \rightarrow$ product.

$$P = P_{2n-1} 2^{2n-2} + \sum_{i=0}^{2n-2} P_i 2^i$$

* If some of the product has negative sign then

$$P = \left[x_{n-1} y_{n-1} 2^{2n-2} + \sum^{n-2} \sum^{n-2} x_i y_i 2^{i+j} \right] - \left[\sum^{n-2} (x_{n-1} y_i + y_{n-1} x_i) 2^{n-1+i} \right]$$

* To move the negative-signed partial products to the last steps & to add the negation of the Partial product rather than subtract.

The final form:

$$P = 2^{n-1} \left[-2^n + 2^{n-1} + \bar{x}_{n-1} 2^{n-1} + x_{n-1} + \sum_{i=0}^{n-2} x_{n-1} \bar{y}_i 2^i \right]$$

* Each partial product is formed with AND functions & the partial products are all added together.

Scanned by CamScanner

The Modified Booth's Algorithm:

* Another Approach which avoids having many idle cells in a cellular multiplier as well as reducing the number of cycles compared with the serial-parallel multiplier is the use of the so-called modified Booth's algorithm.

* In principle, the modified algorithm requires rewriting the multiplicand in such a way that half the bits are 0.

* Booth's algorithm takes advantage of the fact that adder-subtractor is nearly as fast and small as a simple adder.

* The most common form of Booth's algorithm looks a 3 bits of multiplier at a time to perform two stages of the multiplication.

* considering 2's complement of multiplier y.

$$y = (-2)^n y_n + 2^{n-1} y_{n-1} + 2^{n-2} y_{n-2} + \cdots\cdots$$

$$\boxed{2^a = 2^{a+1} - 2^a}$$

$$\therefore \quad y = 2^n (y_{n-1} - y_n) + 2^{n-1} (y_{n-2} - y_{n-1}) + 2^{n-2} (y_{n-3} - y_{n-2})$$

Extracting 2 terms

$$2^n (y_{n-1} - y_n) + 2^{n-1} (y_{n-2} - y_{n-1}).$$

* the right hand term can be used to add x to.

Scanned by CamScanner

Partial products while the left hand term can add 2x.

* If $y_{n-1} = y_n$, the left hand term does not contribute to the partial product. By picking 3 bits of y at a time, we can determine while to add or Subtract x or 2x. to the partial product.

* Each 3 bit Value. overlaps with its neighbours by 1 bit.

Actions during Booth multiplication:-

$y_i\ y_{i-1}\ y_{i-2}$	increment.
0 0 0	0
0 0 1	x
0 1 0	x
0 1 1	2x
1 0 0	-2x
1 0 1	-2x
1 1 0	- x
1 1 1	0

for eg: $x = 011001\ (25_{10})$, $y = 101110\ (-18_{10})$,

P_i = i^{th} partial product

at start $P_0 = 0000\ 0000\ 000$ (two 6 bit nos give an 11 bit result).

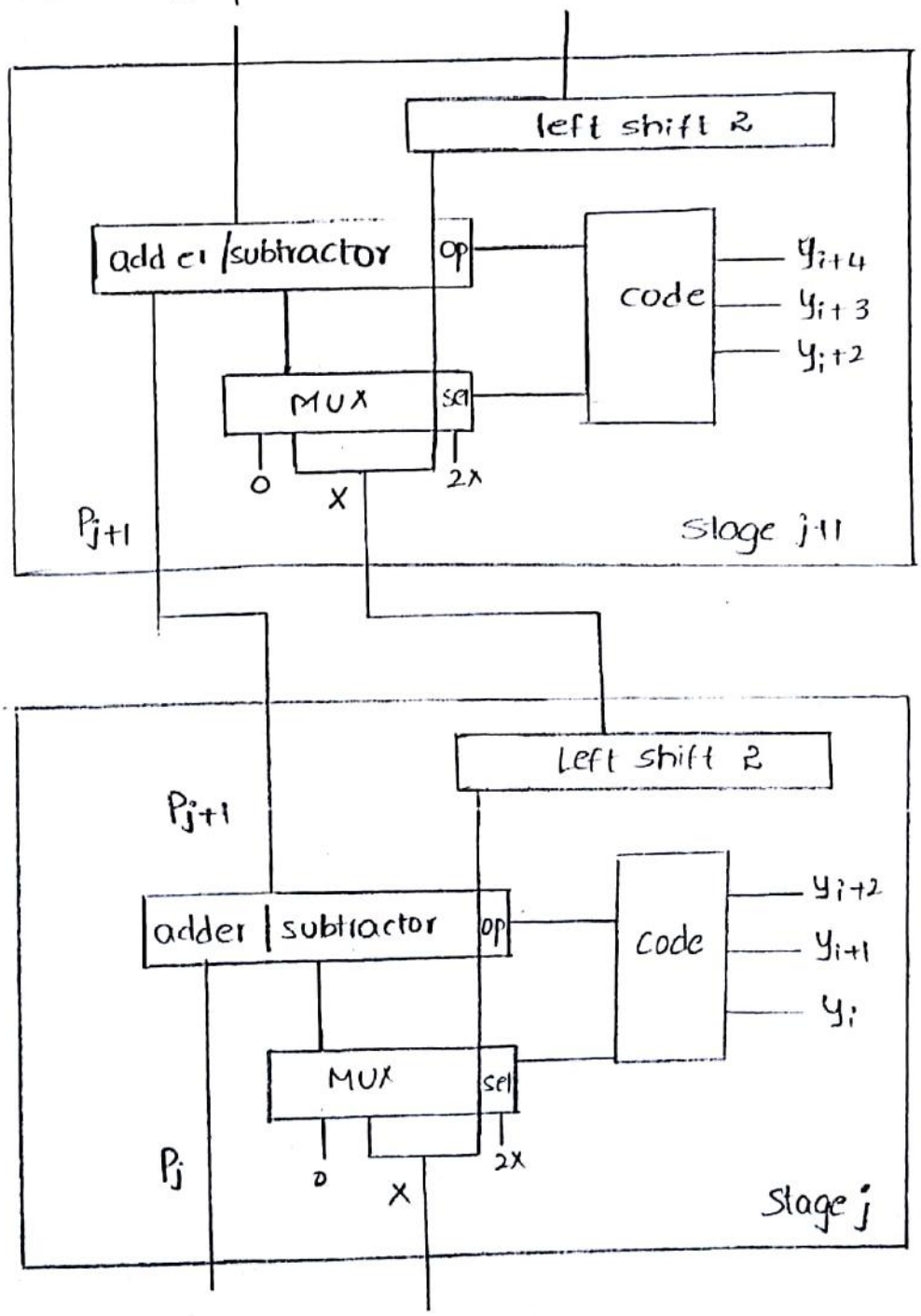
1. $y_1 y_0 y_{-1} = 100$, so $P_1 = P_0 - (10.011001) = 11111001110$

Scanned by CamScanner

2. $y_3 y_2 y_1 = 111$, so $P_2 = P_1 + 0 = 1111100111 0$.

3. $y_5 y_4 y_3 = 101$, so $P_3 = P_2 - 0110010000 = 1100011110$.

In decimal $y_1 y_0 y_{-1}$ contribute -2×1, $y_3 y_2 y_1$ contribute 0.4, $y_5 y_4 y_3$ contribute $-x.16$ giving a total of $-18\ x$. Since the multiplier is -18 the result is correct.



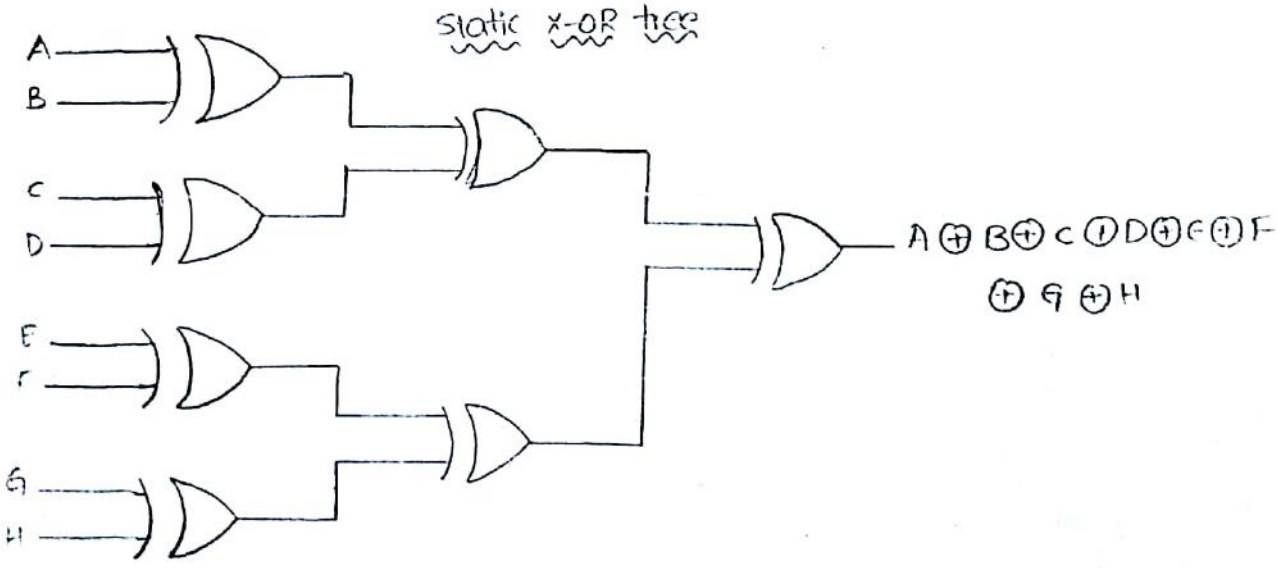
Scanned by CamScanner

Parity Generators

* Parity Generation is a function related to binary addition.
* parity generator detectors whether the no of ones in an input is even or odd.
* Parity generators are most widely used to generate the parity of 16 bit or 32 bit word The function is given by.

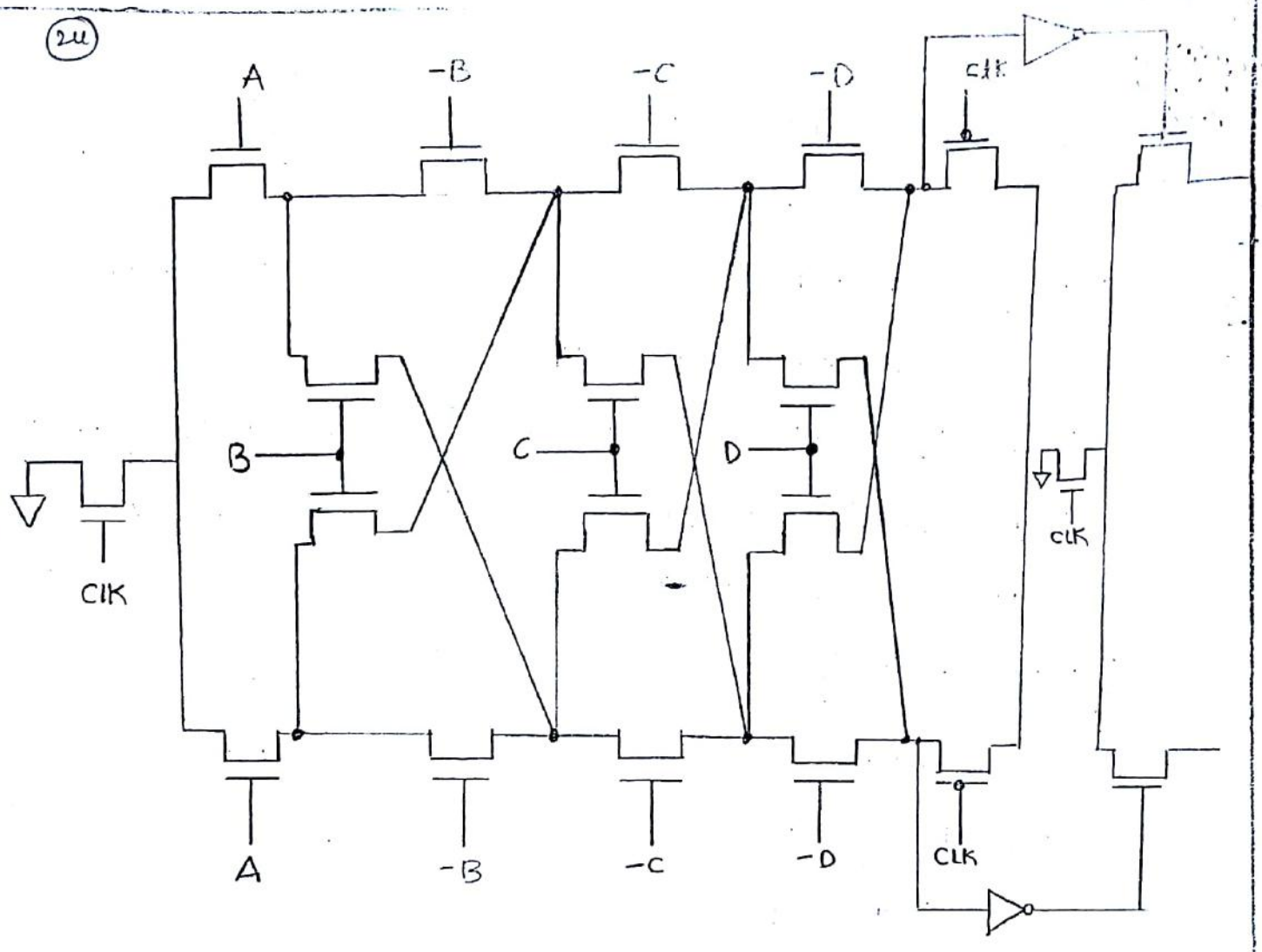
$$\text{Parity} = A_0 \oplus A_1 \oplus A_2 \oplus A_3 \cdots \oplus A_n.$$

* The circuits for the generation of parity bits & checking the parity of a given word can be designed using gates.
* the conventional implementation of parity generator.



* It may be implemented as a column with a tree routing channel computing X-OR gates on a data pattern.
* A dynamic version of parity generator is.

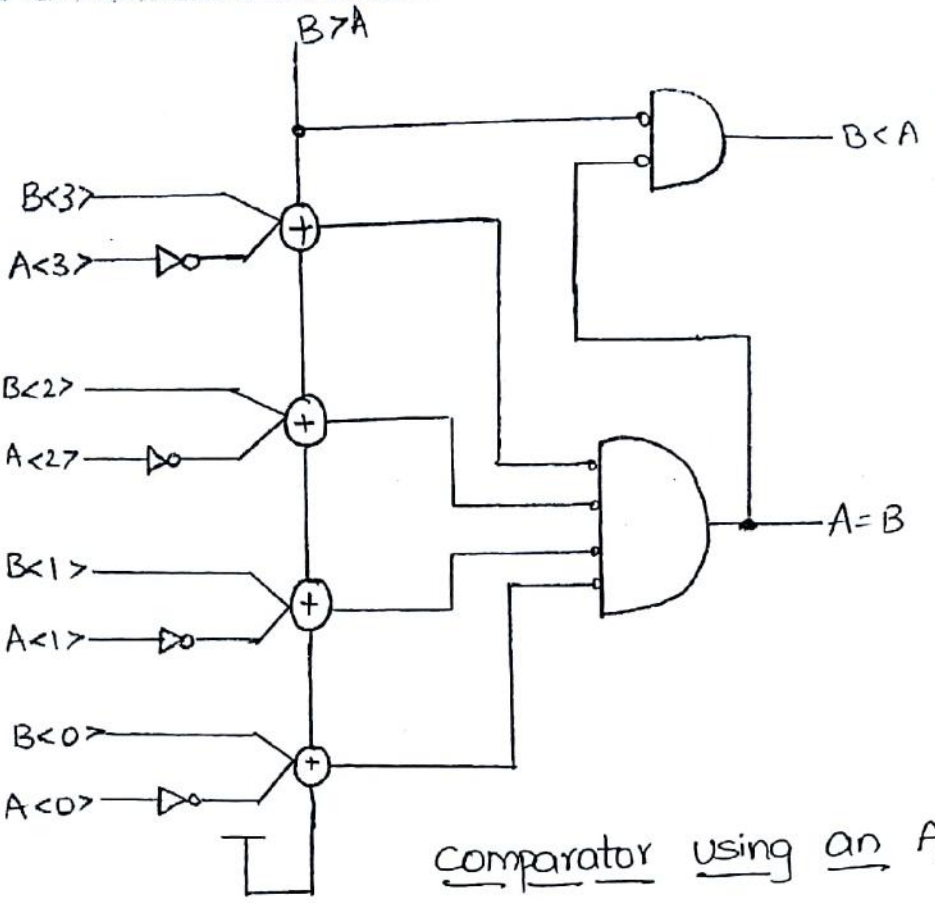
Scanned by CamScanner



* A 32 bit parity function is performed by cascading a no. of dynamic Versions.

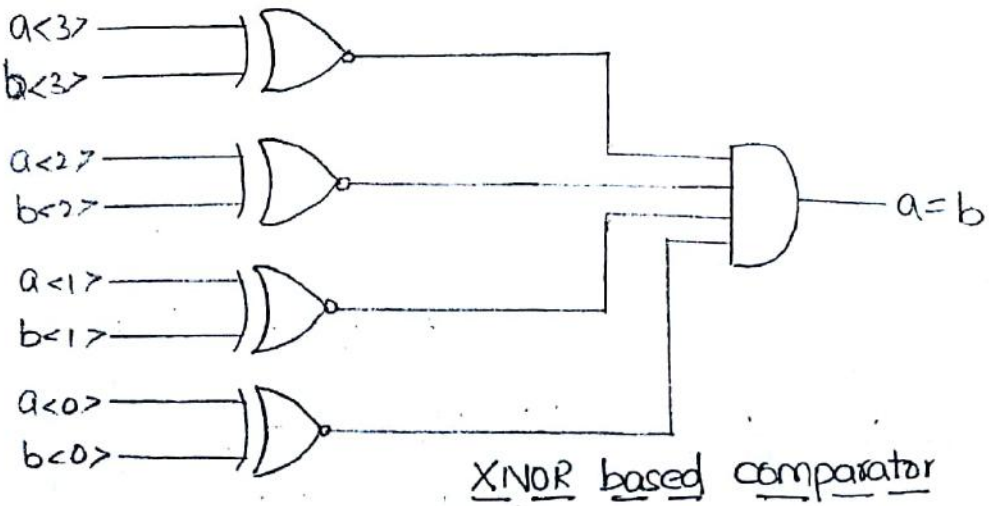
Static 4 input X-OR parity generator.

Scanned by CamScanner



comparator using an Adder

* A NOR gate provides then A=B signal while the final carry output provides the B>A signal.
* A<B or A≤B may be generated by logical combinations of these signals.
* Whenever Equality comparision is required, XNOR gates and AND gate & all that is required is given below.

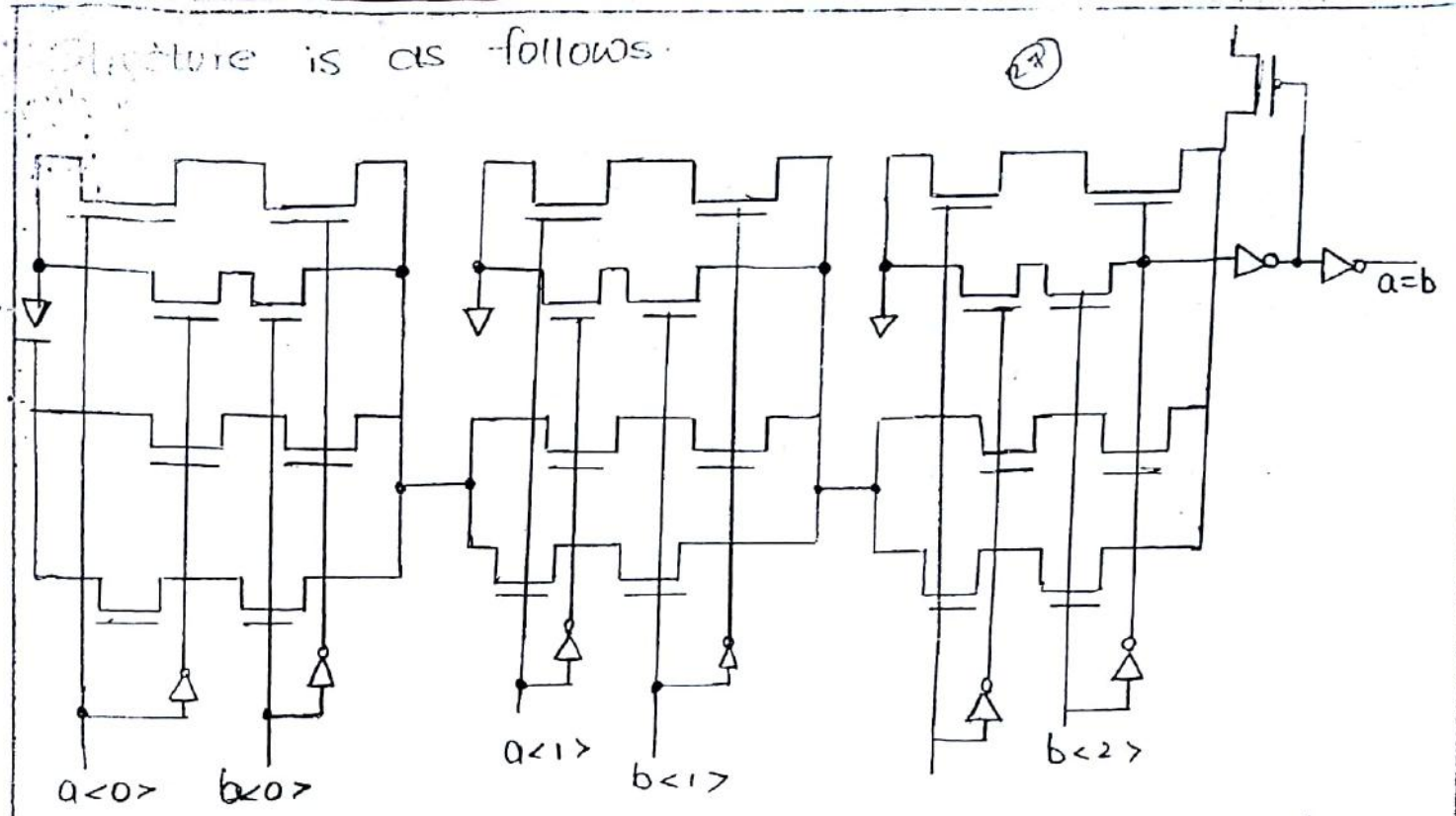


XNOR based comparator

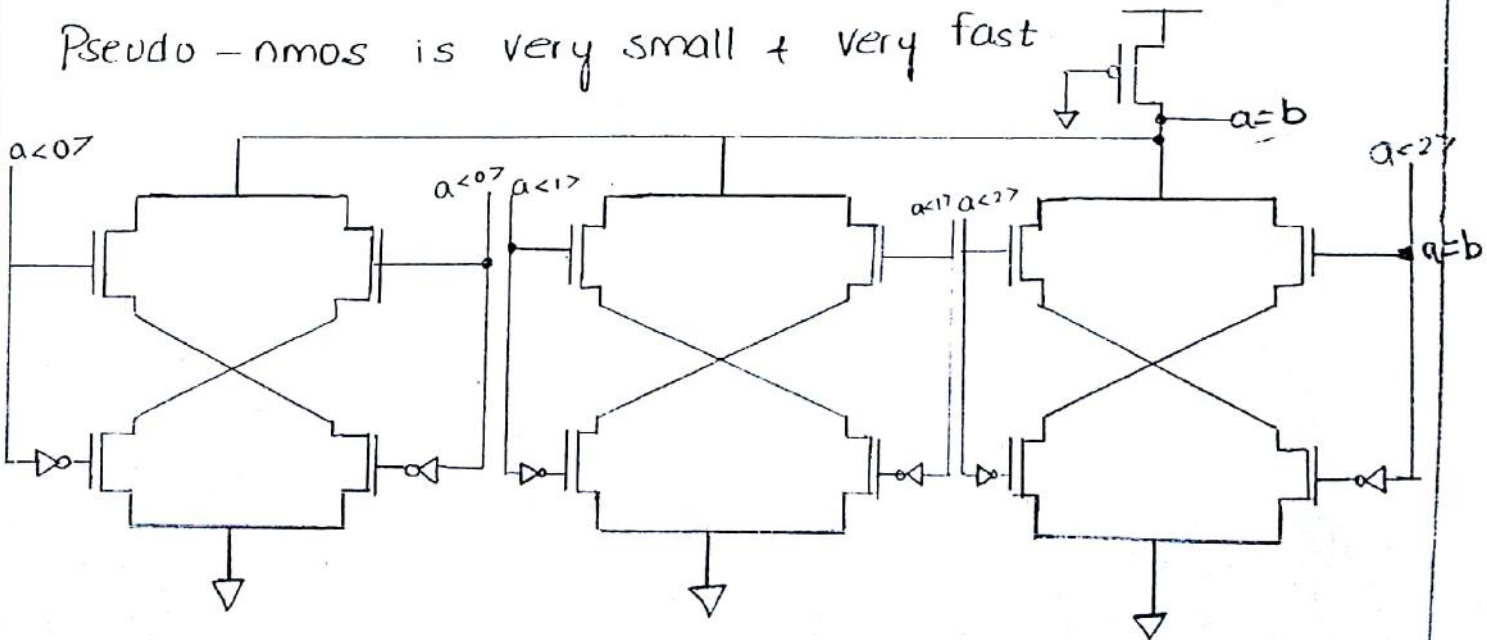
* Rather than a gate implementation, a pass-gate logic

Scanned by CamScanner

Structure is as follows.



* Single polarity transmission gates have been used as appropriate low-power circuit & complementary transmission gates may also be used.
* The DC current drawn by this structure is slow for long comparators.
* D.C current drawn by merged XNOR/NOR gate using Pseudo-nmos is very small & very fast

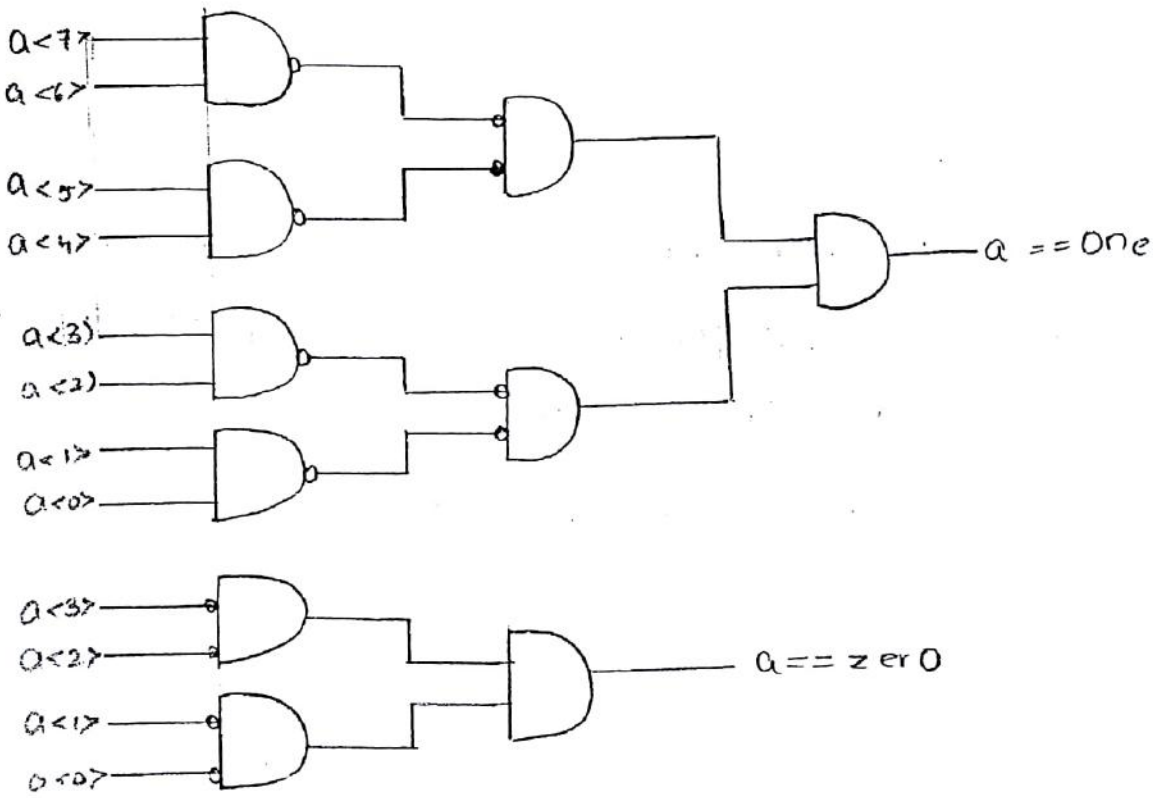


Scanned by CamScanner

Zero / One Detector

Large fan-in OR gates or large fan-in AND gates are required for detecting all zeros or ones on wide words. One and zero-detector circuit built with a tree of AND gates is given below.

One & Zero Detect Circuit Tree



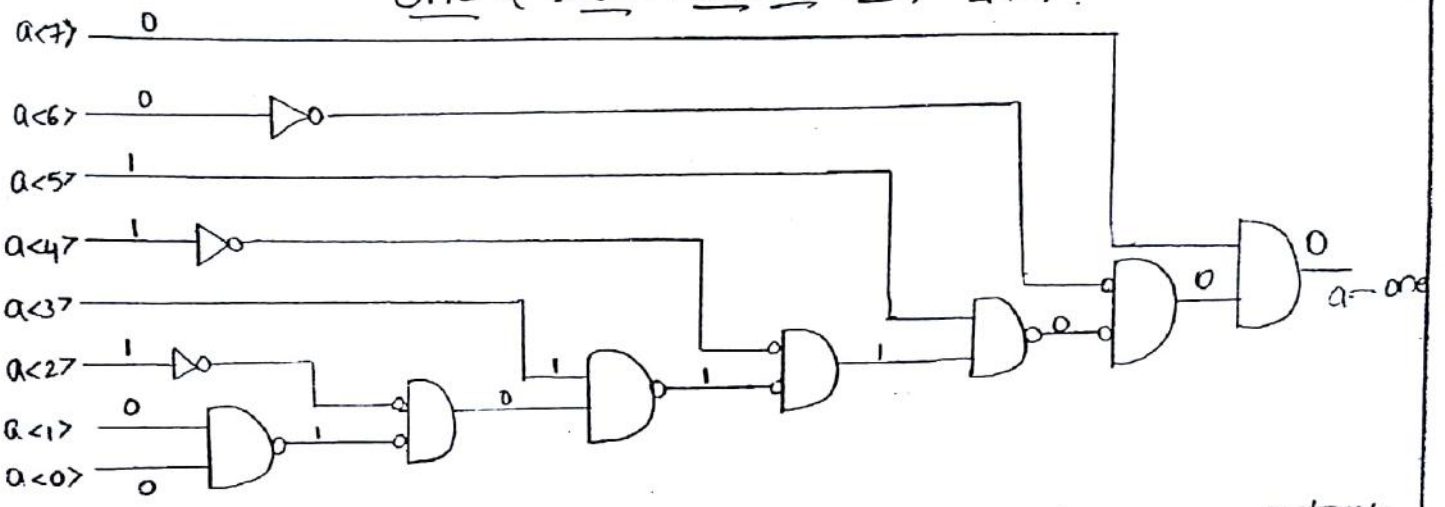
* The delay to the o/p is $\propto \log N$
* Where, N = The bit width of the word.
* If the word being checked has a natural skew in the output the designer might consider mimicking the adder delay in zero or one detector is given below. figure (One & zero detect ckt ripple)

Scanned by CamScanner

* Here, alternately NAND and NOR gates have been used. path has Log N stages. the path logical effort is

$$G_{and}(N) = \left(\frac{4}{3}\right)^{\log_2 N} = N^{\log_2 \frac{4}{3}} = N^{0.415}.$$

One & Zero detect ckt Ripple



* In this case delay from the last changing output to zero or one detect is a constant one gate delay.

* A small and fast zero|one detection ckt for word widths of less than 32 bits is the pseudo nMOS NOR gate

* At large word widths, self-loading may require the Pseudo-nMOS gate to be split in to 8 or 16 bit chunks.

Counters

* Binary counters are used to cycle through a sequence of binary numbers

* Binary counters are classified into 2 types.

1. Asynchronous Counters.
2. Synchronous Counters.

1. Asynchronous Counters:-

A Counter whose outputs can change at varying

Scanned by CamScanner

times w.r.to the clock edge is known as asynchronous Counter.

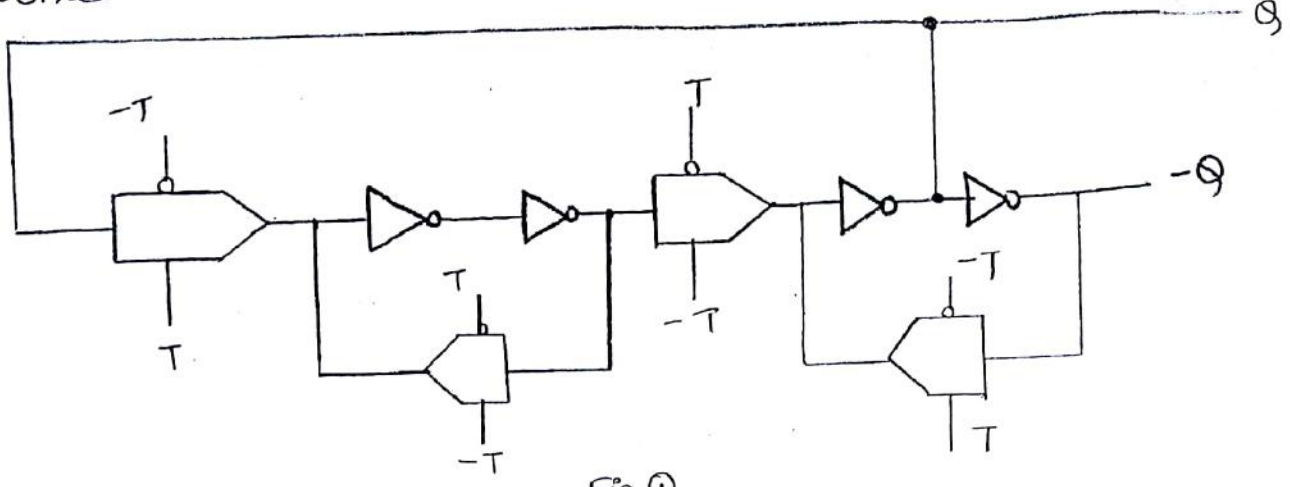


Fig ①

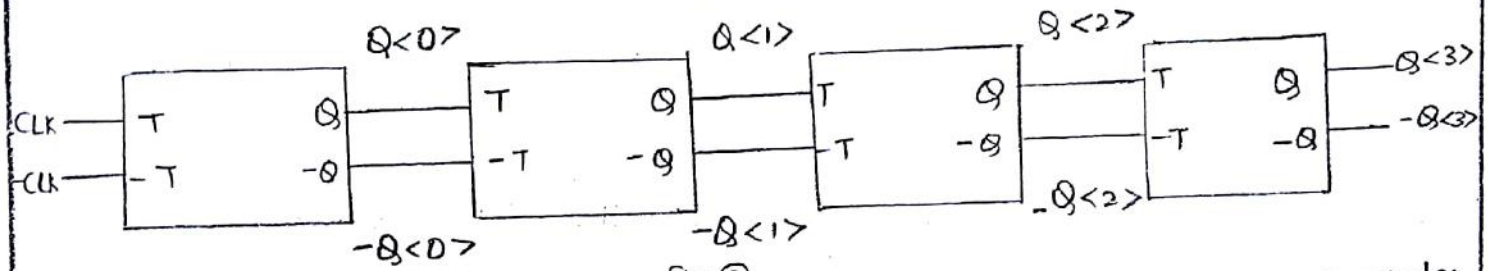


fig ②

* The asynchronous counter is based on the toggle register. The toggle register 'T' is a single counter stage and is in fig ①.
* fig ② is the cascaded stage of asynchronous counter.
* In case asynchronous counter of clocking of each stage is carried out by the previous counter stage.
* Therefore the time it takes the last counter stage to settle can be quite large for a long counter chain.
* this counter has no reset signal, hence making it Extremely difficult to test.

Synchronous Counter:

* The counting speed of synchronous counter is determined by the ripple-carry time from the least significant bit to the

Scanned by CamScanner

* The carry look-ahead techniques are widely used to improve the counting speed of Synchronous Counter. (3)

* If only an incrementor is required, the adder ckt degener-ates in to a synchronous counter stage, comprising an XOR gate, an AND gate, a D-register as shown in fig (2).

* Fig (1) shows the general synchronous up/down counter using adders and registers.

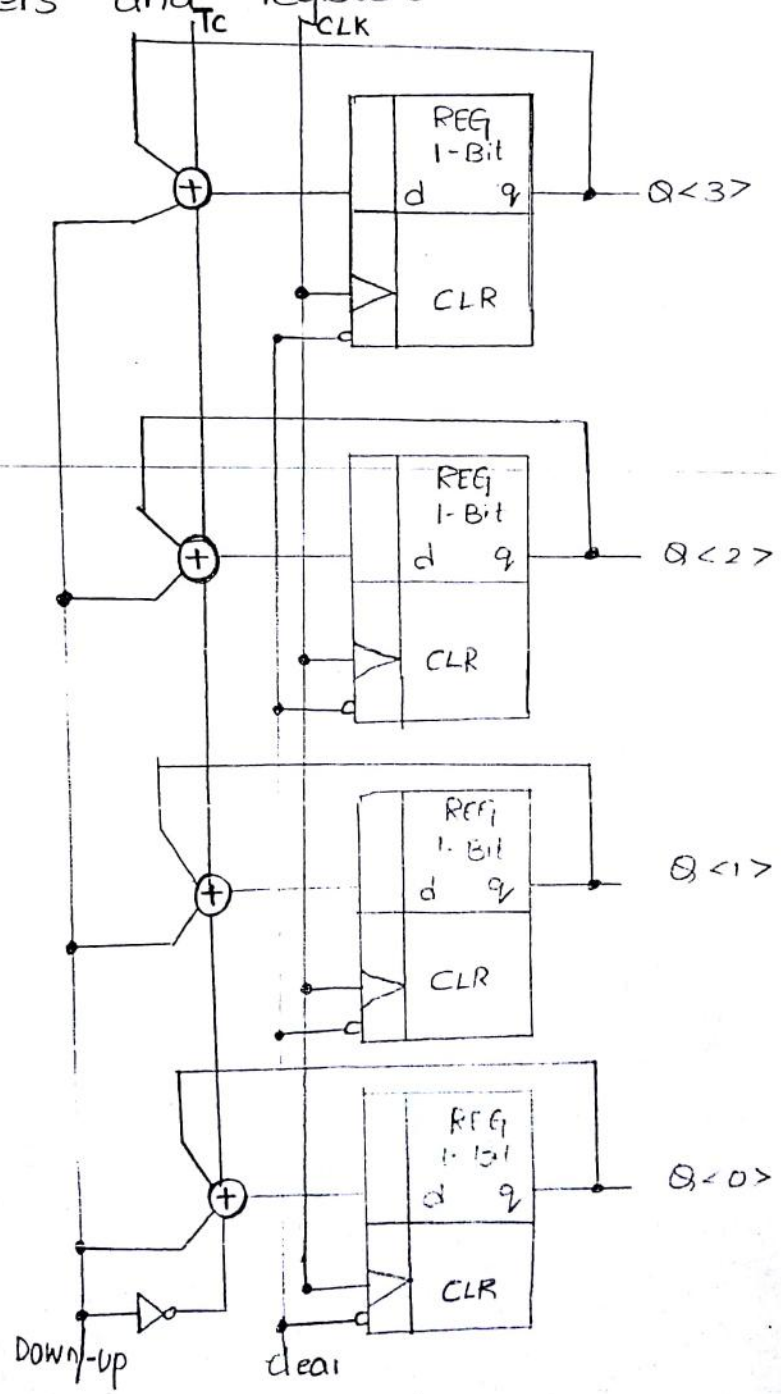


Fig (1):- Synchronous up/Down Counter Using Adders + Registers

Scanned by CamScanner

(3)

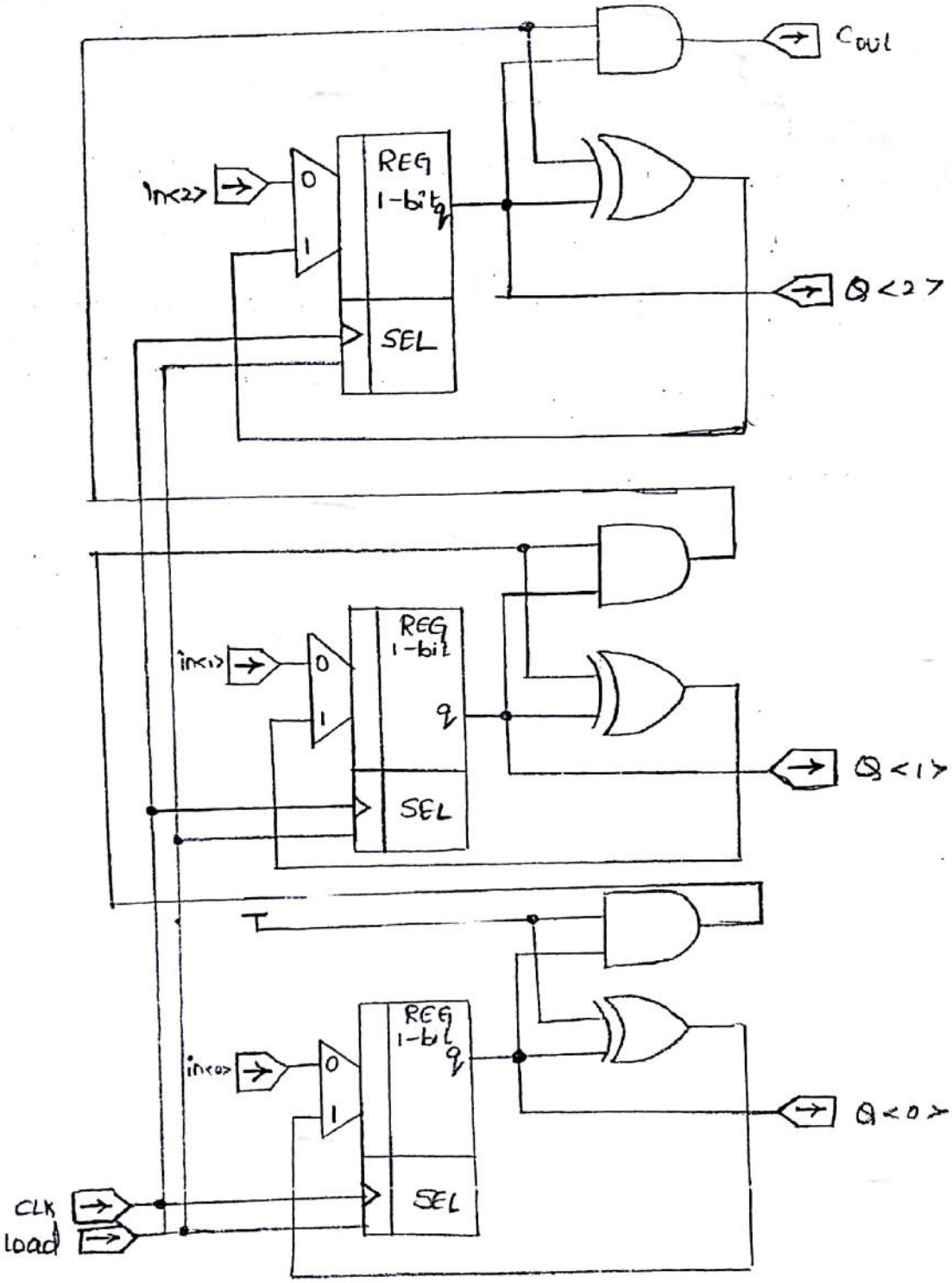


Fig (2): Incrementer

* A multiplexer on the D input of the Register allows a value to be loaded in to the register for initialization. An XOR implemented with multiplexer provides the counter structure as shown in fig (3).
* the multiplexer on the register D-input provides the XOR f'n and a reset register allows initialization.
* Depending upon the carry input value the multiplexer selects

Scanned by CamScanner

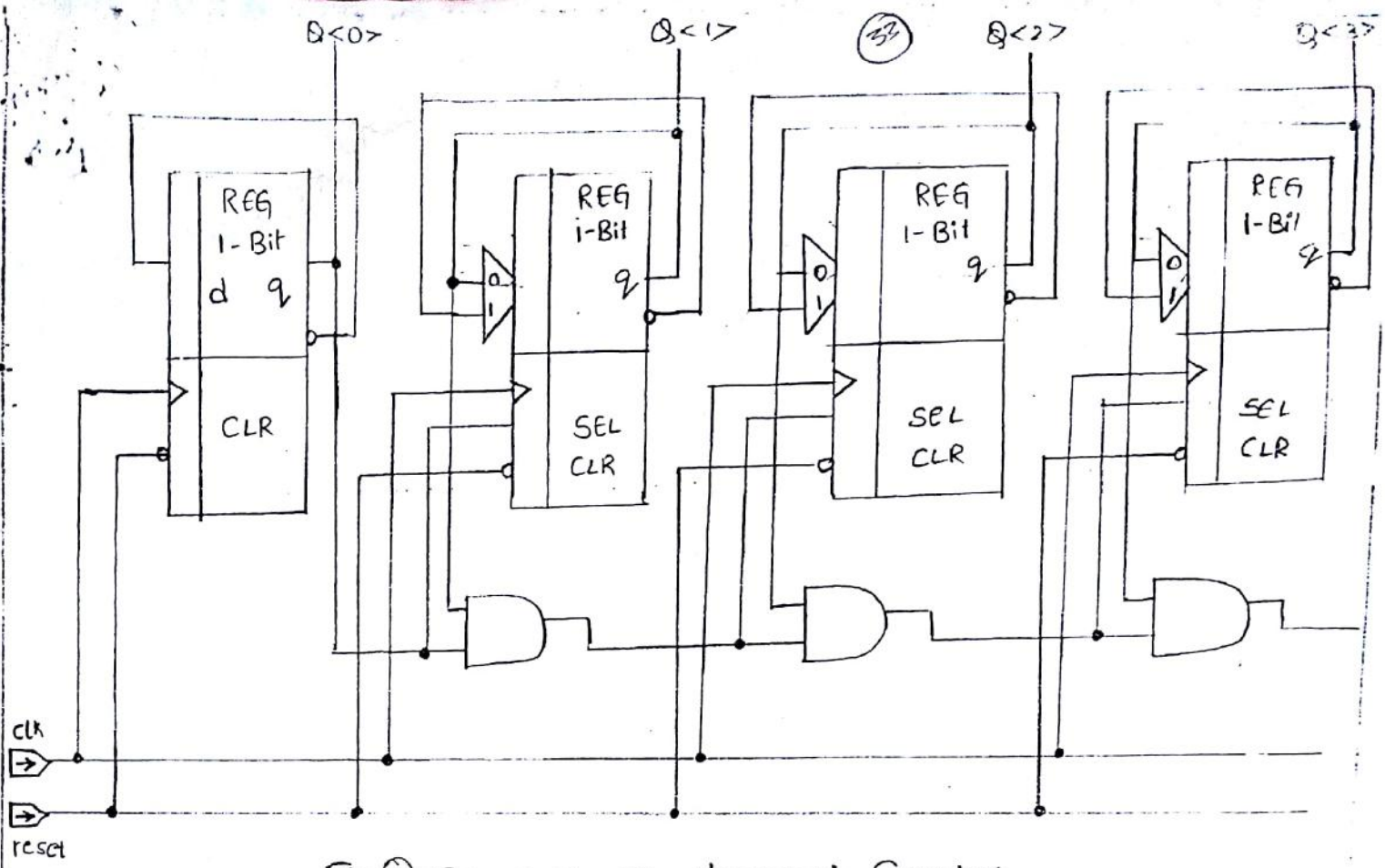
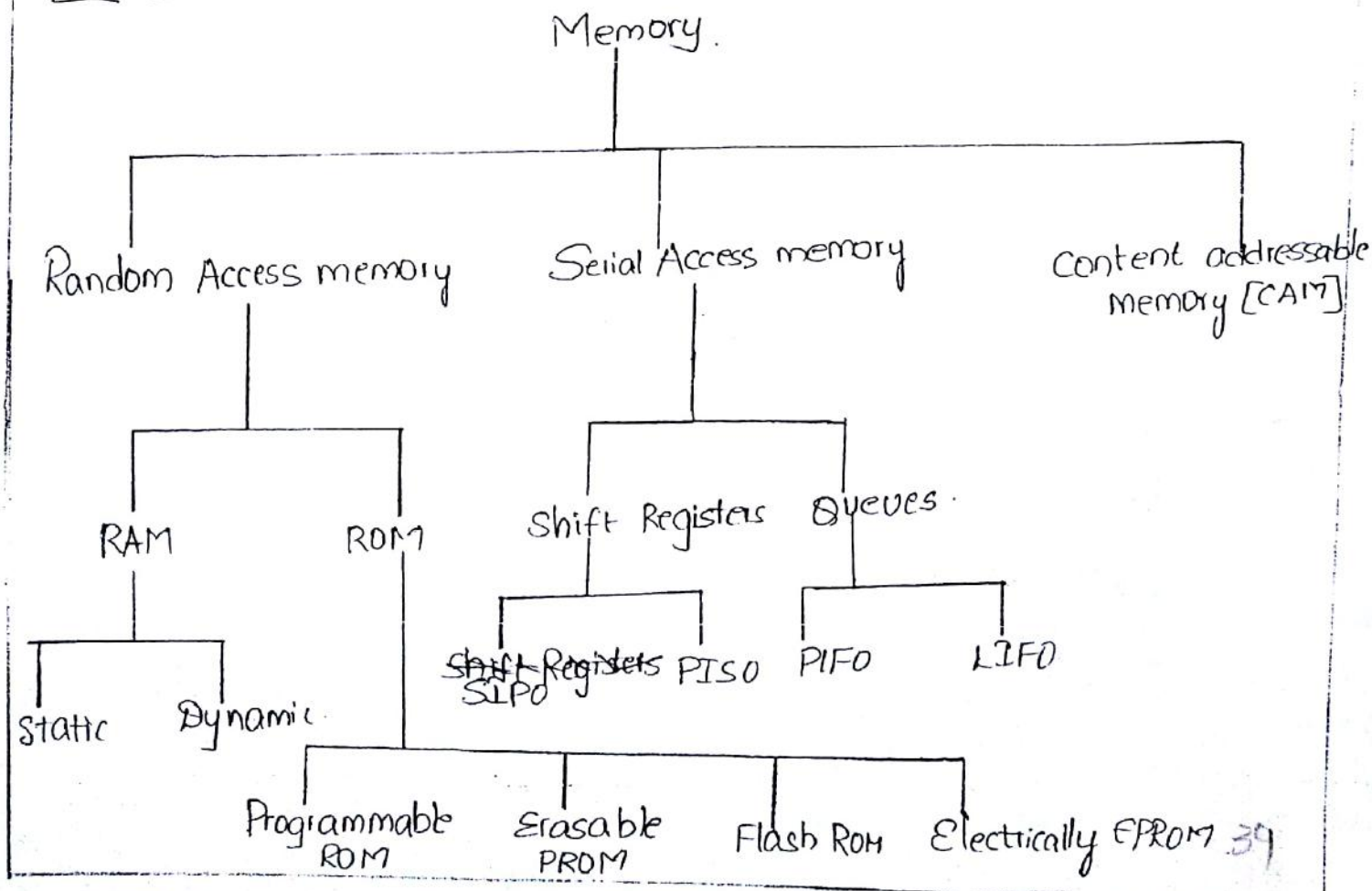


Fig (13): Compact Synchronous Counter

High Density Memory elements:-

Classification



Scanned by CamScanner

Categorizing Memory types:-

Volatile Structured Memories are : Static RAM (S RAM)
Dynamic RAM (D RAM)

* It requires power be applied continuously or else the memory will lose its contents.

Non volatile memories : EPROM
EEPROM
Flash
ROM.

→ Keeps their content even when power removed.

→ Both RAM, ROM are Random Access type. RAM is a Sequential ckt. ROM is a combinational ckt.

Static RAM	Dynamic RAM.
1. In static RAM information Stored in the form of voltage	1. In Dynamic RAM information Stored in the form of charge.
2. Static RAM Constructed Using BJT's	2. Dynamic RAM constructed Using MOSFET's
3. It occupies more space	3. It occupies less space (Here 6 bits can store instead of 1bit in S.R).
4. Power Consumption is more	4. power Consumption is less
5. No Refresh ckt is required	5. Refresh ckt is required for every 2msec.
6. package density poor.	6. Good package density
7. It is faster than Dynamic RAM	7. It is slower.

Scanned by CamScanner

ROM: The ROM is very often used to implement a complex combinational ckt in one IC package and thus eliminate all interconnecting wires.

Types of ROMS
- PROM
- EPROM
- EEPROM | E^2PROM | EAROM

Programmable Read only Memory:-

It is also called as masked ROM. This allows the user to store data. The blowing of fuses is done according to the truth table. If the hardware procedure for programming PROM is irreversible i.e; if once program, the pattern is permanent and can't be altered to change the bit pattern, the entire unit must be discarded. The fuses are made by using Nichrome / polycrystaline material.

EPROM (Erasable PROM)

Erasable programmable Read only memory. Here the information is stored using FAMOS technology (i.e Using floating gate Avalanche injection MOS) Here the programming is electrically erasing is by high intensity UV light, but the disadvantage is a single /bit/ group of bits not possible to erase & Erasing time is also more.

EEPROM / EAROM

Electrically Erasable PROM / Electrically alterable ROM here the programming & Erasing both electrically. Here Nitrable Metal Oxide Semiconductors is used.

Scanned by CamScanner

Advantage:- A single bit | group of bits can erased th... erasing time is few sec.

Disadvantage: It is costly.

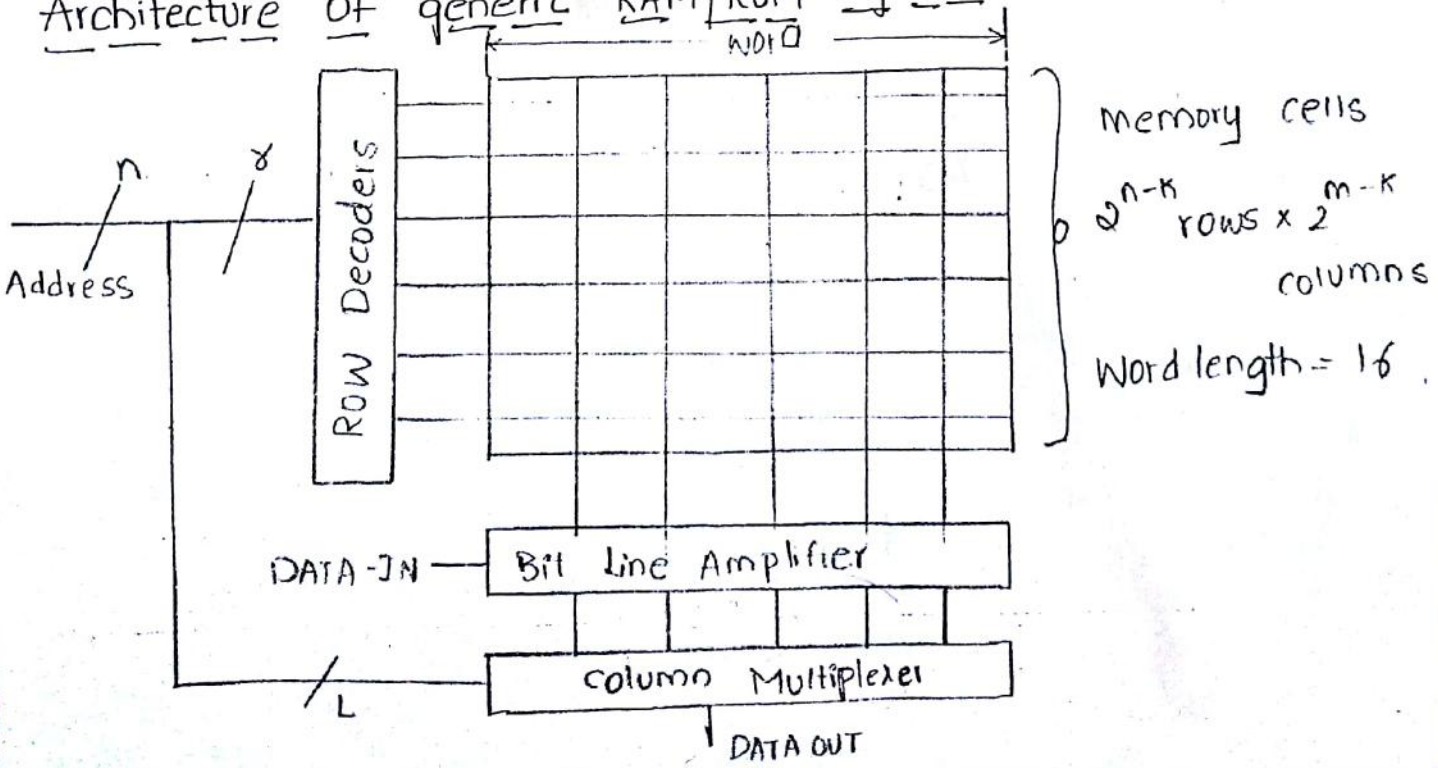
Flash: Flash also require a more complicated process to implement.

→ Flash is a type of programmable memory that is block or page addressable.

ROM:-

1. Another memory type that allows the memory data to remain when power is removed in the ROM.
2. The most common embeded memory in the chip is the "hidden Mask programmable' ROM in which the memory array contents are installed during manufacturing process.
3. Once the data is installed with in the ROM memory it can be altered w/o reprocessing the silicon.

Architecture of generic RAM | ROM System:-



Scanned by CamScanner

what is parity bit:

The parity generating technique is one of the most widely used error detection techniques for the data transmission. In digital systems, when binary data is transmitted and processed , data may be subjected to noise so that such noise can alter 0s (of data bits) to 1s and 1s to 0s.

Hence, **parity bit** is added to the word containing data in order to make number of 1s either even or odd.Thus it is used to detect errors , during the transmission of binary data .The message containing the data bits along with parity bit is transmitted from transmitter node to receiver node.

At the receiving end, the number of 1s in the message is counted and if it doesn't match with the transmitted one, then it means there is an error in the data.

parity generator and checker:

A parity generator is a combinational logic circuit that generates the parity bit in the transmitter. On the other hand, a circuit that checks the parity in the receiver is called parity checker. A combined circuit or devices of parity generators and parity checkers are commonly used in digital systems to detect the single bit errors in the transmitted data word.

The sum of the data bits and parity bits can be even or odd . In even parity, the added parity bit will make the total number of 1s an even amount whereas in odd parity the added parity bit will make the total number of 1s odd amount.

The basic principle involved in the implementation of parity circuits is that sum of odd number of 1s is always 1 and sum of even number of 1s is always zero. Such error detecting and correction can be implemented by using Ex-OR gates (since Ex-OR gate produce zero output when there are even number of inputs).

To produce two bits sum, one Ex-OR gate is sufficient whereas for adding three bits two Ex-OR gates are required as shown in below figure.

Parity Generator:

It is combinational circuit that accepts an n-1 bit stream data and generates the additional bit that is to be transmitted with the bit stream. This additional or extra bit is termed as a parity bit.

In **even parity** bit scheme, the parity bit is '**0**' if there are **even number of 1s** in the data stream and the parity bit is '**1**' if there are **odd number of 1s** in the data stream.

In **odd parity** bit scheme, the parity bit is '**1**' if there are **even number of 1**s in the data stream and the parity bit is '**0**' if there are **odd number of 1s** in the data stream. Let us discuss both even and odd parity generators.

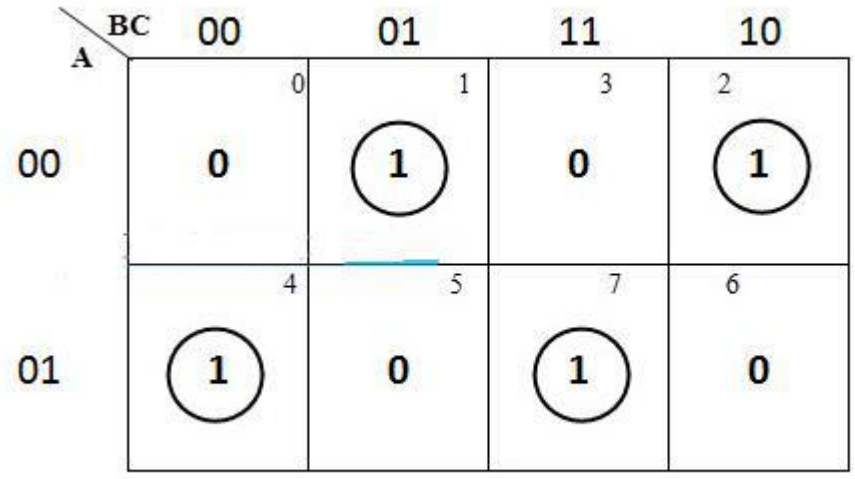
Even Parity Generator

Let us assume that a 3-bit message is to be transmitted with an even parity bit. Let the three inputs A, B and C are applied to the circuits and output bit is the parity bit P. The total number of 1s must be even, to generate the even parity bit P.

The figure below shows the truth table of even parity generator in which 1 is placed as parity bit in order to make all 1s as even when the number of 1s in the truth table is odd.

3-bit message			Even parity bit generator (P)
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

The K-map simplification for 3-bit message even parity generator is



From the above truth table, the simplified expression of the parity bit can be written as

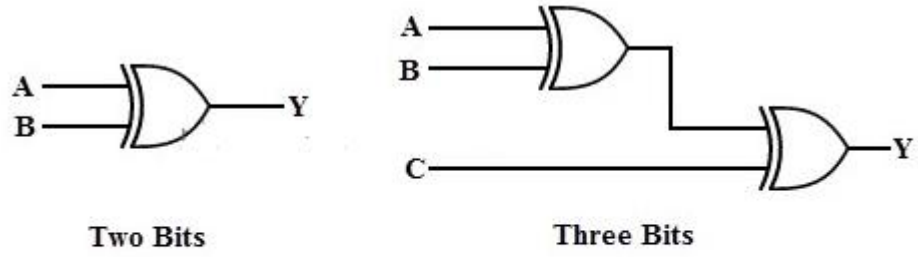
$$P = \bar{A}\,\bar{B}\,C + \bar{A}\,B\,\bar{C} + A\,\bar{B}\,\bar{C} + A\,B\,C$$

$$= \bar{A}\,(\bar{B}\,C + B\,\bar{C}) + A\,(\bar{B}\,\bar{C} + B\,C)$$

$$= \bar{A}\,(B \oplus C) + A\,(\overline{B \oplus C})$$

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

The above expression can be implemented by using two Ex-OR gates. The logic diagram of even parity generator with two Ex – OR gates is shown below. The three bit message along with the parity generated by this circuit which is transmitted to the receiving end where parity checker circuit checks whether any error is present or not.



Two Bits Three Bits

Parity Generator:

It is combinational circuit that accepts an n-1 bit stream data and generates the additional bit that is to be transmitted with the bit stream. This additional or extra bit is termed as a parity bit.

In **even parity** bit scheme, the parity bit is '**0**' if there are **even number of 1s** in the data stream and the parity bit is '**1**' if there are **odd number of 1s** in the data stream.

In **odd parity** bit scheme, the parity bit is '**1**' if there are **even number of 1**s in the data stream and the parity bit is '**0**' if there are **odd number of 1s** in the data stream. Let us discuss both even and odd parity generators.

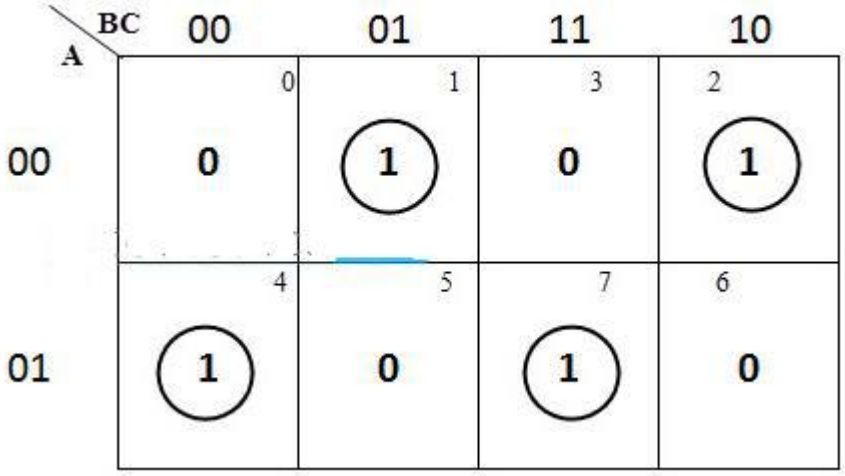
Even Parity Generator

Let us assume that a 3-bit message is to be transmitted with an even parity bit. Let the three inputs A, B and C are applied to the circuits and output bit is the parity bit P. The total number of 1s must be even, to generate the even parity bit P.

The figure below shows the truth table of even parity generator in which 1 is placed as parity bit in order to make all 1s as even when the number of 1s in the truth table is odd.

3-bit message			Even parity bit generator (P)
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

The K-map simplification for 3-bit message even parity generator is



From the above truth table, the simplified expression of the parity bit can be written as

$$P = \bar{A}\,\bar{B}\,C + \bar{A}\,B\,\bar{C} + A\,\bar{B}\,\bar{C} + A\,B\,C$$

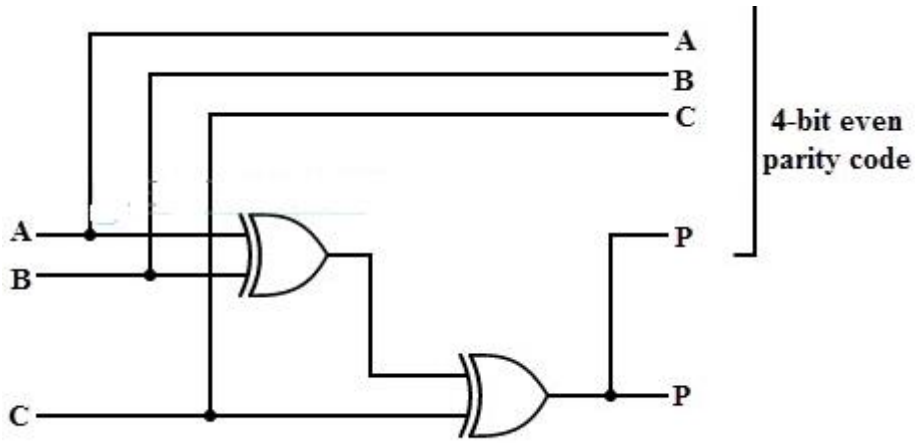
$$= \bar{A}\,(\bar{B}\,C + B\,\bar{C}) + A\,(\bar{B}\,\bar{C} + B\,C)$$

$$= \bar{A}\,(B \oplus C) + A\,(\overline{B \oplus C})$$

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

The above expression can be implemented by using two Ex-OR gates. The logic diagram of even parity generator with two Ex – OR gates is shown below. The three bit message along with the parity generated by this circuit which is transmitted to the receiving end where parity checker circuit checks whether any error is present or not.

To generate the even parity bit for a 4-bit data, three Ex-OR gates are required to add the 4-bits and their sum will be the parity bit.



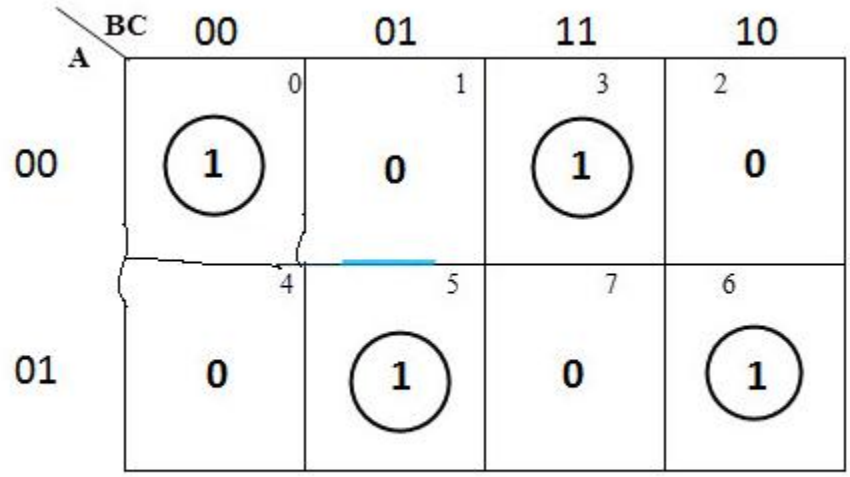
Odd Parity Generator

Let us consider that the 3-bit data is to be transmitted with an odd parity bit. The three inputs are A, B and C and P is the output parity bit. The total number of bits must be odd in order to generate the odd parity bit.

In the given truth table below, 1 is placed in the parity bit in order to make the total number of bits odd when the total number of 1s in the truth table is even.

3-bit message			Odd parity bit generator (P)
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

The truth table of the odd parity generator can be simplified by using K-map as

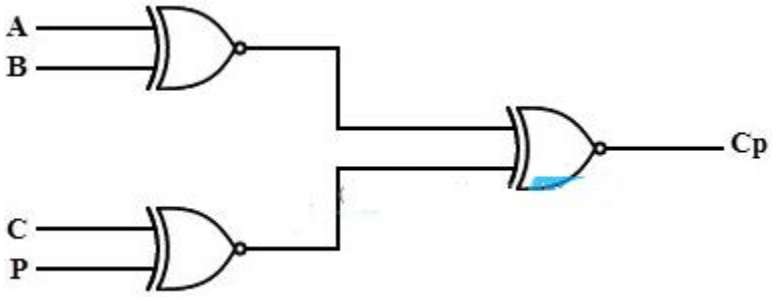


The output parity bit expression for this generator circuit is obtained as

$$P = A \oplus B \text{ Ex-NOR } C$$

The above Boolean expression can be implemented by using one Ex-OR gate and one Ex-NOR gate in order to design a 3-bit odd parity generator.

The logic circuit of this generator is shown in below figure , in which . two inputs are applied at one Ex-OR gate, and this Ex-OR output and third input is applied to the Ex-NOR gate , to produce the odd parity bit. It is also possible to design this circuit by using two Ex-OR gates and one NOT gate.



Parity Check

It is a logic circuit that checks for possible errors in the transmission. This circuit can be an even parity checker or odd parity checker depending on the type of parity generated at the transmission end. When this circuit is used as even parity checker, the number of input bits must always be even.

When a parity error occurs, the ‘sum even’ output goes low and ‘sum odd’ output goes high. If this logic circuit is used as an odd parity checker, the number of input bits should be odd, but if an error occurs the ‘sum odd’ output goes low and ‘sum even’ output goes high.

Even Parity Checker

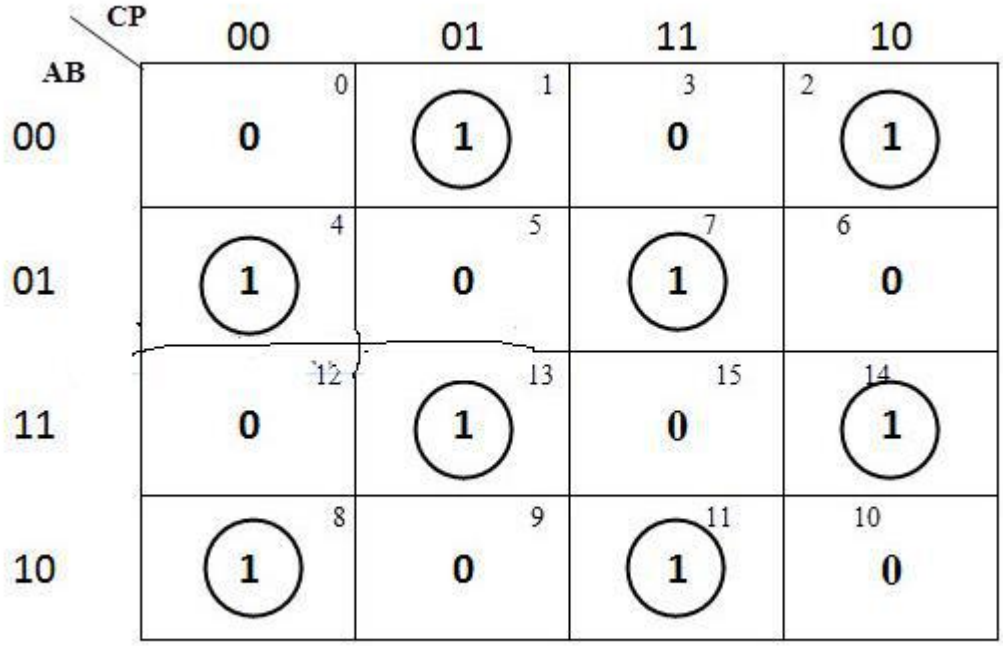
Consider that three input message along with even parity bit is generated at the transmitting end. These 4 bits are applied as input to the parity checker circuit which checks the possibility of error on the data. Since the data is transmitted with even parity, four bits received at circuit must have an even number of 1s.

If any error occurs, the received message consists of odd number of 1s. The output of the parity checker is denoted by PEC (parity error check).

The below table shows the truth table for the even parity checker in which PEC = 1 if the error occurs, i.e., the four bits received have odd number of 1s and PEC = 0 if no error occurs, i.e., if the 4-bit message has even number of 1s.

4-bit received message				Parity error check C_p
A	B	C	P	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

The above truth table can be simplified using K-map as shown below.



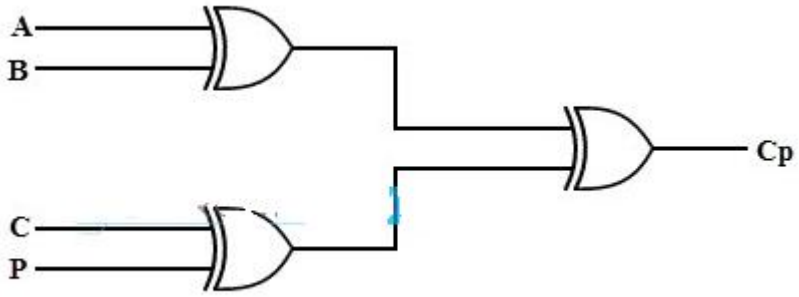
$$PEC = \bar{A}\,\bar{B}\,(\bar{C}\,D + C\,\bar{D}) + \bar{A}\,B\,(\bar{C}\,\bar{D} + C\,D) + A\,B\,(\bar{C}\,D + C\,\bar{D}) + A\,\bar{B}\,(\bar{C}\,\bar{D} + C\,D)$$

$$= \bar{A}\,\bar{B}\,(C \oplus D) + \bar{A}\,B\,(\overline{C \oplus D}) + A\,B\,(C \oplus D) + A\,\bar{B}\,(\overline{C \oplus D})$$

$$= (\bar{A}\,\bar{B} + A\,B)\,(C \oplus D) + (\bar{A}\,B + A\,\bar{B})\,(\overline{C \oplus D})$$

$$= (A \oplus B) \oplus (C \oplus D)$$

The above logic expression for the even parity checker can be implemented by using three Ex-OR gates as shown in figure. If the received message consists of five bits, then one more Ex-OR gate is required for the even parity checking.



Odd Parity Checker

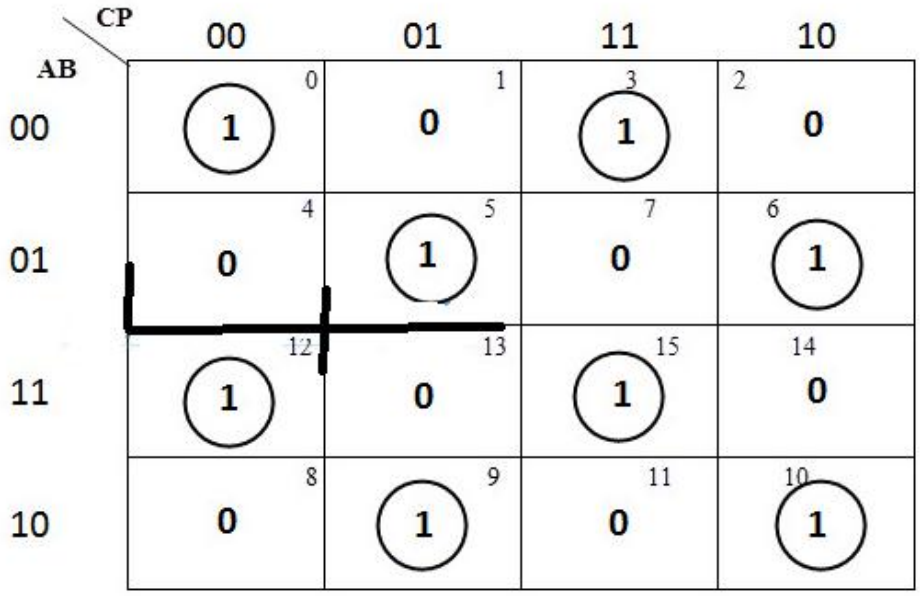
Consider that a three bit message along with odd parity bit is transmitted at the transmitting end. Odd parity checker circuit receives these 4 bits and checks whether any error are present in the data.

If the total number of 1s in the data is odd, then it indicates no error, whereas if the total number of 1s is even then it indicates the error since the data is transmitted with odd parity at transmitting end.

The below figure shows the truth table for odd parity generator where **PEC =1** if the 4-bit message received consists of **even number of 1s** (hence the error occurred) and **PEC= 0** if the message contains **odd number of 1s** (that means no error).

4-bit received message				Parity error check C_p
A	B	C	P	
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

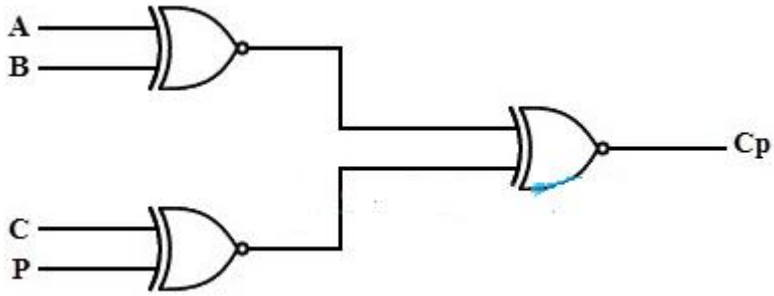
The expression for the PEC in the above truth table can be simplified by K-map as shown below.



After simplification, the final expression for the PEC is obtained as

PEC = (A Ex-NOR B) Ex-NOR (C Ex-NOR D)

The expression for the odd parity checker can be designed by using three Ex-NOR gates as shown below.

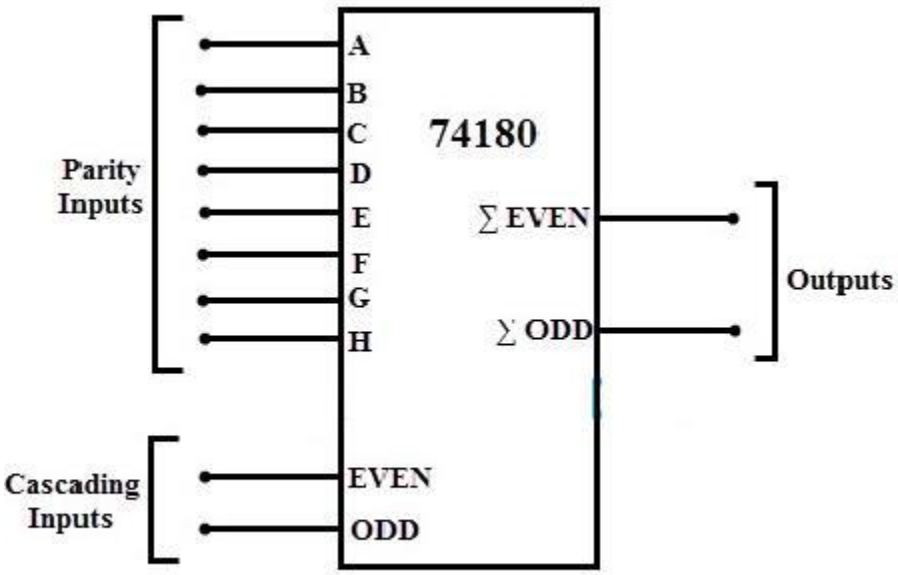


Parity Generator/Checker ICs:

There are different types of parity generator /checker ICs are available with different input configurations such as 5-bit, 4-bit, 9-bit, 12-bit, etc. A most commonly used and standard type of parity generator/checker IC is 74180.

It is a 9-bit parity generator or checker used to detect errors in high speed data transmission or data retrieval systems. The figure below shows the pin diagram of 74180 IC.

This IC can be used to generate a 9-bit odd or even parity code or it can be used to check for odd or even parity in a 9-bit code (8 data bits and one parity bit).



This IC consists of eight parity inputs from A through H and two cascading inputs. There are two outputs even sum and odd sum. In implementing generator or checker circuits, unused parity bits must be tied to logic zero and the cascading inputs must not be equal.

Digital Comparator:

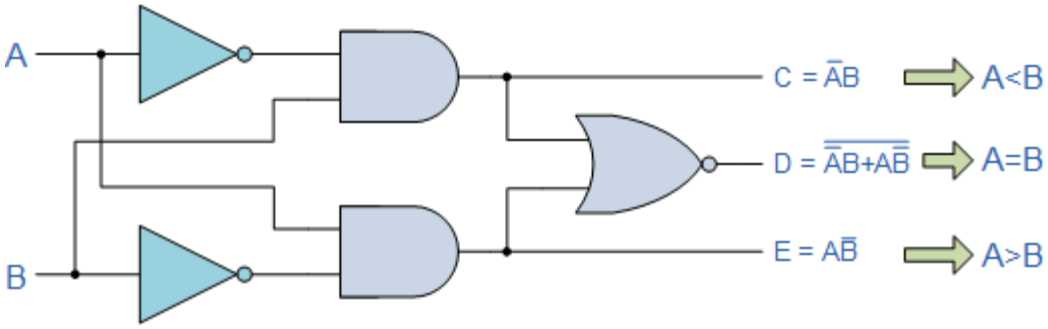


Fig.Digital Comparator

Another common and very useful combinational logic circuit is that of the **Digital Comparator** circuit. Digital or Binary Comparators are made up from standard AND, NOR and NOT gates that compare the digital signals present at their input terminals and produce an output depending upon the condition of those inputs.

For example, along with being able to add and subtract binary numbers we need to be able to compare them and determine whether the value of input A is greater than, smaller than or equal to the value at input B etc. The digital comparator accomplishes this using several logic gates that operate on the principles of *Boolean Algebra*. There are two main types of **Digital Comparator** available and these are.

- 1. Identity Comparator – an *Identity Comparator* is a digital comparator that has only one output terminal for when A = B either "HIGH" A = B = 1 or "LOW" A = B = 0
- 2. Magnitude Comparator – a *Magnitude Comparator* is a digital comparator which has three output terminals, one each for equality, A = B greater than, A > B and less than A < B

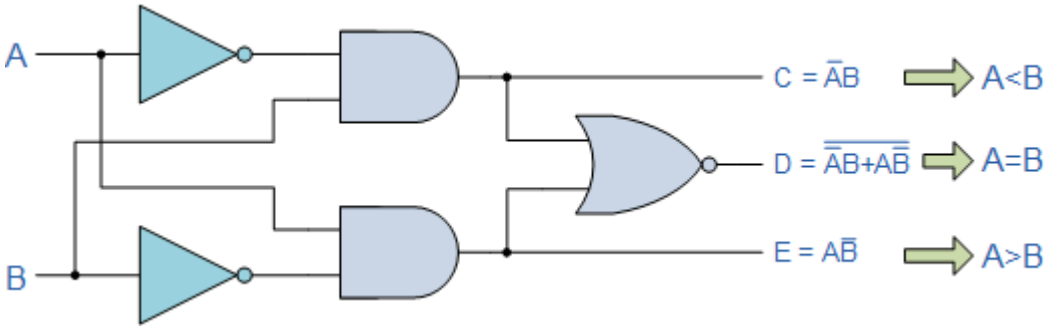
The purpose of a **Digital Comparator** is to compare a set of variables or unknown numbers, for example A (A1, A2, A3, …. An, etc) against that of a constant or unknown value such as B (B1, B2, B3, …. Bn, etc) and produce an output condition or flag depending upon the result of the comparison. For example, a magnitude comparator of two 1-bits, (A and B) inputs would produce the following three output conditions when compared to each other.

$$A > B, \quad A = B, \quad A < B$$

Which means: A is greater than B, A is equal to B, and A is less than B

This is useful if we want to compare two variables and want to produce an output when any of the above three conditions are achieved. For example, produce an output from a counter when a certain count number is reached. Consider the simple 1-bit comparator below.

1-bit Digital Comparator Circuit



Then the operation of a 1-bit digital comparator is given in the following Truth Table.

Digital Comparator Truth Table

Inputs		Outputs		
B	A	A > B	A = B	A < B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

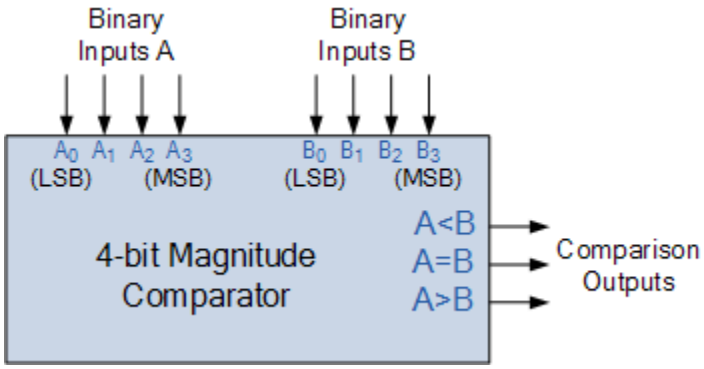
You may notice two distinct features about the comparator from the above truth table. Firstly, the circuit does not distinguish between either two "0" or two "1"'s as an output A = B is produced when they are both equal, either A = B = "0" or A = B = "1". Secondly, the output condition for A = B resembles that of a commonly available logic gate, the Exclusive-NOR or Ex-NOR function (equivalence) on each of the n-bits giving: Q = A ⊕ B

Digital comparators actually use Exclusive-NOR gates within their design for comparing their respective pairs of bits. When we are comparing two binary or BCD values or variables against each other, we are comparing the "magnitude" of these values, a logic "0" against a logic "1" which is where the term **Magnitude Comparator** comes from.

As well as comparing individual bits, we can design larger bit comparators by cascading together n of these and produce a n-bit comparator just as we did for the n-bit adder in the previous tutorial. Multi-bit comparators can be constructed to compare whole binary or BCD words to produce an output if one word is larger, equal to or less than the other.

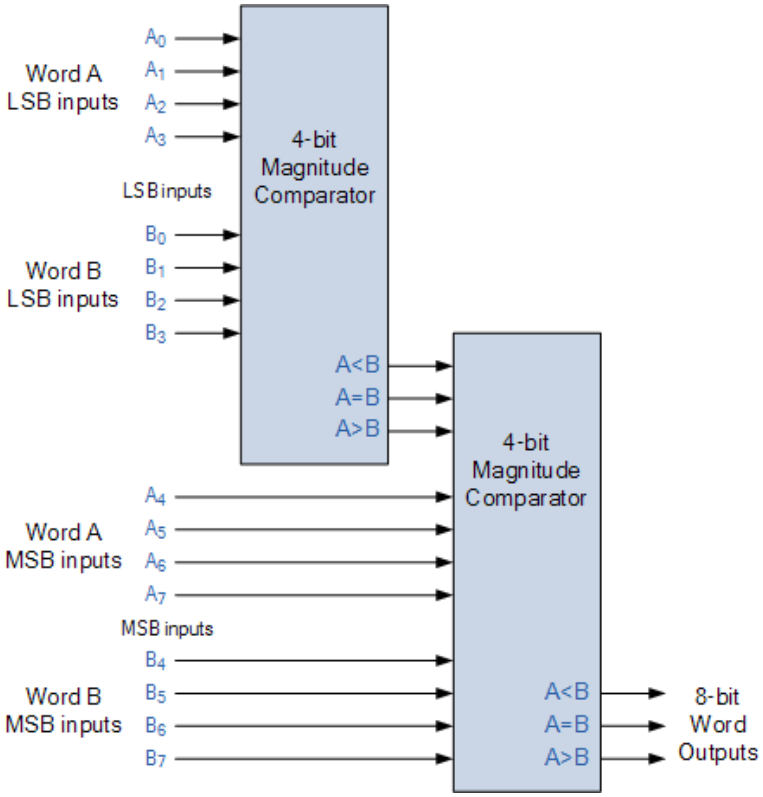
A very good example of this is the 4-bit **Magnitude Comparator**. Here, two 4-bit words ("nibbles") are compared to each other to produce the relevant output with one word connected to inputs A and the other to be compared against connected to input B as shown below.

4-bit Magnitude Comparator



Some commercially available digital comparators such as the TTL 74LS85 or CMOS 4063 4-bit magnitude comparator have additional input terminals that allow more individual comparators to be "cascaded" together to compare words larger than 4-bits with magnitude comparators of "n"-bits being produced. These cascading inputs are connected directly to the corresponding outputs of the previous comparator as shown to compare 8, 16 or even 32-bit words.

8-bit Word Comparator:



When comparing large binary or BCD numbers like the example above, to save time the comparator starts by comparing the highest-order bit (MSB) first. If equality exists, A = B then it compares the next lowest bit and so on until it reaches the lowest-order bit, (LSB). If equality still exists then the two numbers are defined as being equal.

If inequality is found, either A > B or A < B the relationship between the two numbers is determined and the comparison between any additional lower order bits stops. **Digital Comparator** are used widely in Analogue-to-Digital converters, (ADC) and Arithmetic Logic Units, (ALU) to perform a variety of arithmetic operations.

UNIT IV

SUBSYSTEM DESIGN

Objectives: At the end of this unit we will be able to understand

- Design consideration, problem and solution
- Design processes
- Basic digital processor structure
- Data path
- Bus Architecture
- Design 4 – bit shifter
- Design of ALU subsystem
- 4 – bit Adder

General Considerations:

- Lower unit cost
- Higher reliability
- Lower power dissipation, lower weight and lower volume
- Better performance
- Enhanced repeatability
- Possibility of reduced design/development periods

Some Problems :

1. How to design complex systems in a reasonable time & with reasonable effort.
2. The nature of architectures best suited to take full advantage of VLSI and the technology
3. The testability of large/complex systems once implemented on silicon

Some Solution

Problem 1 & 3 are greatly reduced if two aspects of standard practices are accepted.

1. a) Top-down design approach with adequate CAD toolsto do the job
   b) Partitioning the system sensibly
   c) Aiming for simple interconnections
   d) High regularity within subsystem
   e) Generate and then verify each section of the design
2. Devote significant portion of total chip area to test and diagnostic facility
3. Select architectures that allow design objectives and high regularity in realization

Illustration of design processes

1. Structured design begins with the concept of hierarchy
2. It is possible to divide any complex function into less complex sub functions that is up to leaf cells

3. Process is known as top-down design
4. As a systems complexity increases, its organization changes as different factors become relevant to its creation
5. Coupling can be used as a measure of how much sub models interact
6. It is crucial that components interacting with high frequency be physically proximate, since one may pay severe penalties for long, high-bandwidth interconnects
7. Concurrency should be exploited – it is desirable that all gates on the chip do useful work most of the time
8. Because technology changes so fast, the adaptation to a new process must occur in a short time.

Hence representing a design several approaches are possible. They are:

- Conventional circuit symbols
- Logic symbols
- Stick diagram
- Any mixture of logic symbols and stick diagram thatis convenient at a stage
- Mask layouts
- Architectural block diagrams and floor plans

General arrangements of a 4 – bit arithmetic processor

The basic architecture of **digital processor structure** is as shown below in figure

Here the design of datapath is only considered.

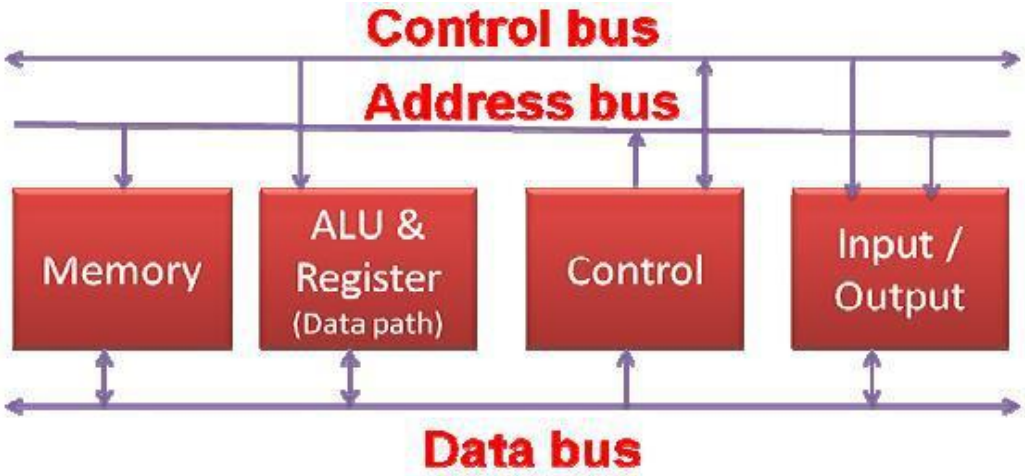
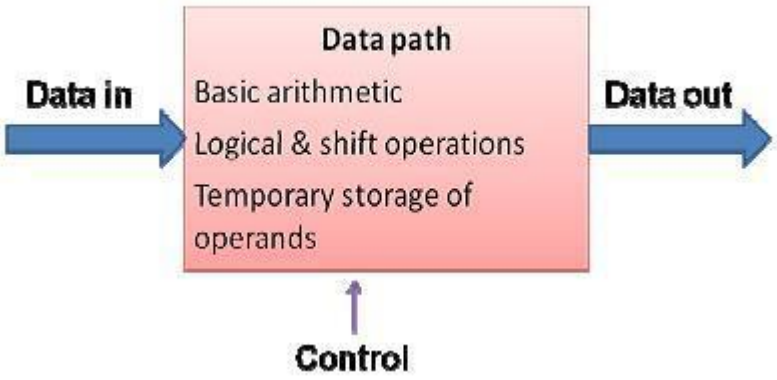


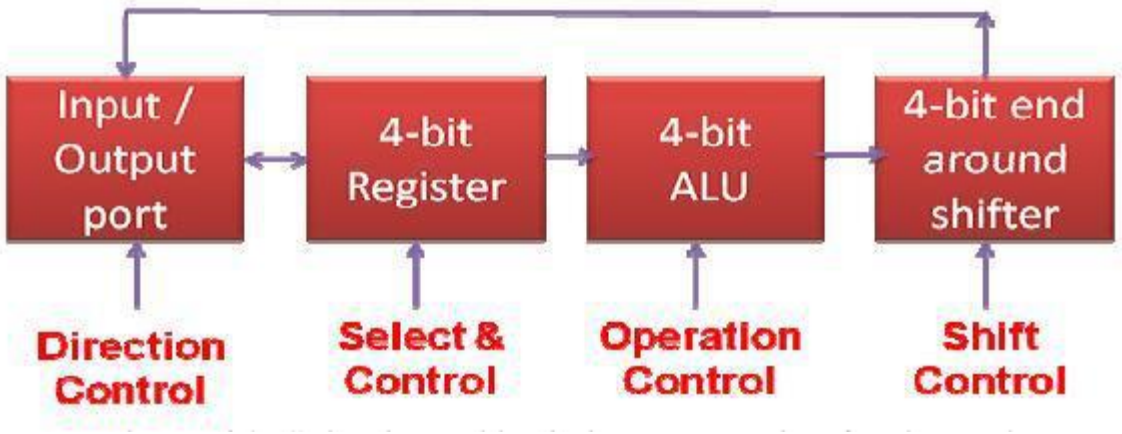
Figure 6.1: Basic digital processor structure

Datapath is as shown below in figure 6.2. It is seen that the structure comprises of a unit which processes data applied at one port andpresents its output at a second port. Alternatively, the two data ports may be combined as a single bidirectional port if storage facilities exist in the datapath. Control over the functions to be performed is effected by control signals as shown



Communication strategy for the datapath

Datapath can be decomposed into blocks showing the main subunits as in figure 3. In doing so it is useful to anticipate a possible floor plan to show the planned relative decomposition of the subunits on the chip and hence on the mask layouts.



Subunits and basic interconnection for datapath

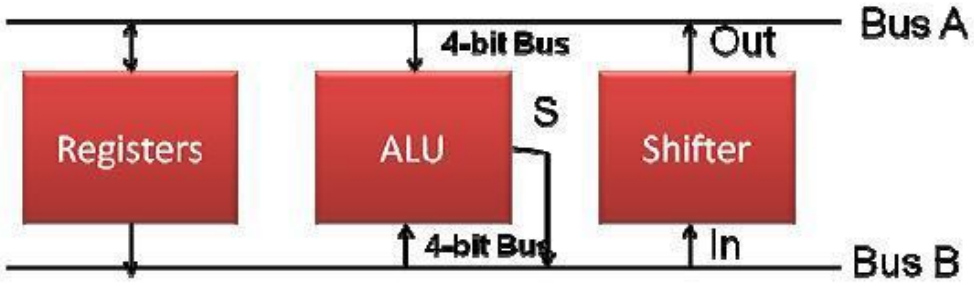
Nature of the bus architecture linking the subunits is discussed below. Some of the possibilities are:

One bus architecture:

Sequence:

1. 1st operand from registers to ALU. Operand is stored there.
2. 2nd operand from register to ALU and added.
3. Result is passed through shifter and stored in the register

Two bus architecture:

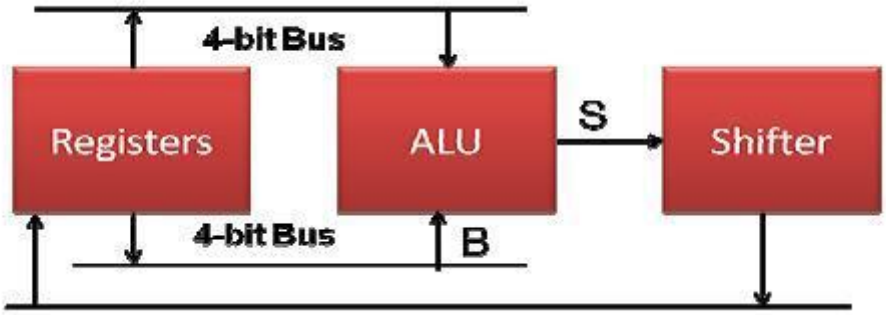


Two bus architecture

Sequence:

1. Two operands (A & B) are sent from register(s) to ALU & are operated upon, result S in ALU.
2. Result is passed through the shifter & stored in registers.

Three bus architecture:

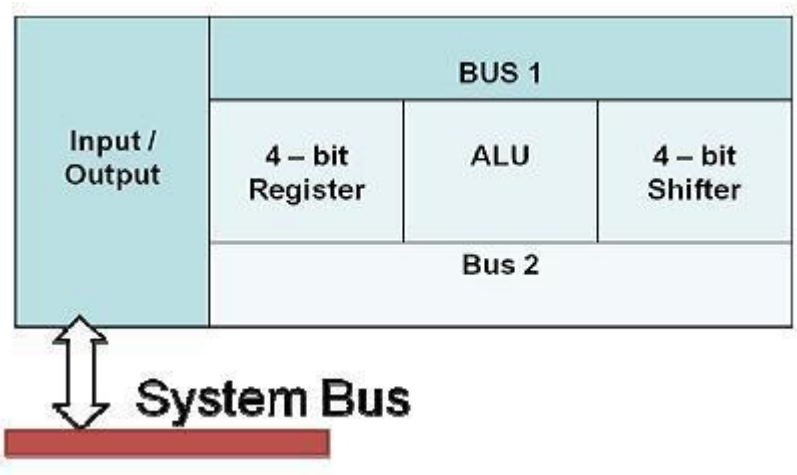


Three bus architecture

Sequence:

Two operands (A & B) are sent from registers, operated upon, and shifted result (S) returned to another register, all in same clock period.

In pursuing this design exercise, it was decided toimplement the structure with a 2 – bus architecture. A tentative floor plan of theproposed design which includes some form of interface to the parent system data bus is shown in figure



Tentative floor plan for 4 – bit datapath

The proposed processor will be seen to comprise a register array in which 4-bit numbers can be stored, either from an I/O port or from the output of the ALU via a shifter. Numbers from the register array can be fedin pairs to the ALU to be added (or subtracted) and the result can be shifted or not. The data connections between the I/O port, ALU, and shifter must be in the form of 4-bitbuses. Also, each of the blocks must be suitably connected to control lines so that its function may be defined for any of a range of possible operations.

During the design process, and in particular when defining the interconnection strategy and designing the stick diagrams, care must be taken in allocating the layers to the various data or control paths. Points to be noted:

Metal can cross poly or diffusion Poly crossing diffusion form a transistor Whenever lines touch on the same level an interconnection is formed Simple contacts can be used to join diffusion or poly to metal.

Buried contacts or a butting contacts can be used to join diffusion and poly Each layer has particular electrical properties which must be taken into account For CMOS layouts, p-and n-diffusion wires must not directly join each other Nor may they cross either a p-well or an n-well boundary

Design of a 4-bit shifter

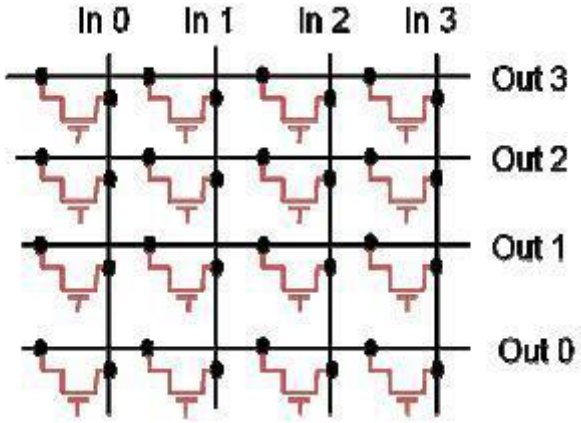
Any general purpose n-bit shifter should be able toshift incoming data by up to n – 1 place in a right-shift or left-shift direction. Further specifying that all shifts should be on an end-around basis, so that any bit shifted out at one end of a data word will be shifted in at the other end of the word, then the problem of right shift or left shift is greatly eased. It can be analyzed that for a 4-bit word, that a 1-bit shift right is equivalent to a 3-bit shift left and a 2-bit shift right is equivalent to a 2-bit left etc. Hence, the design of either shift right or left can be done. Here thedesign is of shift right by 0, 1, 2, or 3 places.

The shifter must have:

- input from a four line parallel data bus
- four output lines for the shifted data
- means of transferring input data to output lines with any shift from 0 to 3 bits

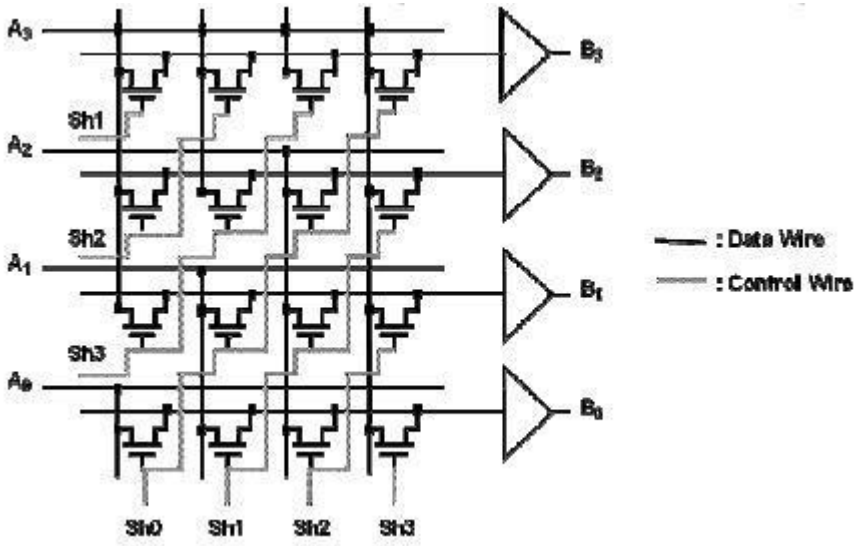
Consider a direct MOS switch implementation of a 4 X 4 crossbar switches shown in figure 6.8. The arrangement is general and may be expanded to accommodate n-bit inputs/outputs. In this arrangement any input can be connected to any or all the outputs.

Furthermore, 16 control signals (sw00 – sw15), one for each transistor switch, must be provided to drive the crossbar switch, and such complexity is highly undesirable.



4 X 4 crossbar switch

An adaptation of this arrangement recognizes the fact that we couple the switch gates together in groups of four and also form four separate groups corresponding to shifts of zero, one, two and three bits. The resulting arrangement is known as a barrel shifter and a 4 X 4 barrel shifter circuit diagram is as shown in the figure



barrel shifter

The interbus switches have their gate inputs connected in a staircase fashion in groups of four and there are now four shift control inputs which must be mutually exclusive in the active state. CMOS transmission gates may be used in place of the simple pass transistor switches if appropriate. Barrel shifter connects the input lines representing a word to a group of output lines withthe required shift determined by its control inputs (sh0, sh1, sh2, sh3). Control inputsalso determine the direction of the shift.

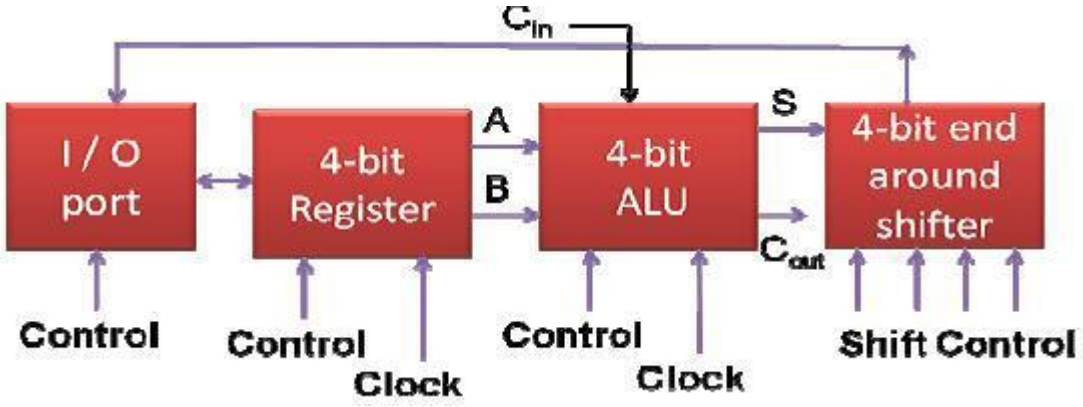
If input word has n – bits and shifts from 0 to n-1bit positions are to be implemented.

To summaries the design steps

- Set out the specifications
- Partition the architecture into subsystems
- Set a tentative floor plan
- Determine the interconnects
- Choose layers for the bus & control lines
- Conceive a regular architecture
- Develop stick diagram
- Produce mask layouts for standard cell
- Cascade & replicate standard cells as required to complete the design

Design of an ALU subsystem

Having designed the shifter, we shall design another subsystem of the 4-bit data path. An appropriate choice is ALU as shown in the figure below.



4-bit data path for processor

The heart of the ALU is a 4-bit adder circuit. A 4-bit adder must take sum of two 4-bit numbers, and there is an assumption that all 4-bit quantities are presented in parallel form and that the shifter circuit is designed to accept and shift a 4-bit parallel sum from the ALU. The sum is to be stored in parallel at theoutput of the adder from where it is fed through the shifter and back to the register array. Therefore, a single 4-bit data bus is needed from the adder to the shifter and another 4-bit bus is required from the shifted output back to the register

array. Hence, for an adder two 4-bit parallel numbers are fed on two 4-bit buses. The clock signal is also required to the adder, during which the inputs are given and sum is generated. The shifter is unclocked but must be connected to four shift control lines.

Design of a 4-bit adder:

The truth table of binary adder is as shown in table

Inputs			Outputs	
A_k	B_k	C_{k-1}	S_k	C_k
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

As seen from the table any column k there will be three inputs namely A_k , B_k as present input number and C_{k-1} as the previous carry. It can also be seen that there are two outputs sum S_k and carry C_k.

From the table one form of the equation is:

Sum $S_k = H_k C_{k-1}' + H_k' C_{k-1}$

New carry $C_k = A_k B_k + H_k c_{k-1}$

Where

Half sum $H_k = A_k' B_k + A_k B_k'$

Adder element requirements

Table 6.1 reveals that the adder requirement may be stated as:

If $A_k = B_k$ then $S_k = C_{k-1}$

Else $S_k = C_{k-1}'$

And for the carry C_k

If $A_k = B_k$ then $C_k = A_k = B_k$

Else $C_k = C_{k-1}$

Thus the standard adder element for 1-bit is as shown in the figure 6.11.

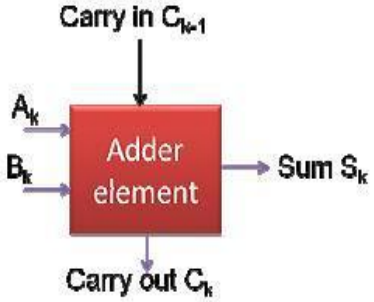


Figure 6.11: Adder element

Implementing ALU functions with an adder:

An ALU must be able to add and subtract two binary numbers, perform logical operations such as And, Or and Equality (Ex-or) functions. Subtraction can be performed by taking 2's complement of the negative number and perform the further addition. It is desirable to keep the architecture as simple as possible, and also see that the adder performs the logical operations also. Hence let us examine the possibility.

The adder equations are:

Sum $S_k = H_k C_{k-1}' + H_k' C_{k-1}$

New carry $C_k = A_k B_k + H_k C_{k-1}$

Where

Half sum $H_k = A_k' B_k + A_k B_k'$

Let us consider the sum output, if the previous carry is at logical 0, then

$S_k = H_k . 1 + H_k' . 0$

$S_k = H_k = A_k' B_k + A_k B_k'$ – An Ex-or operation

Now, if C_{k-1} is logically 1, then

$S_k = H_k . 0 + H_k' . 1$

$S_k = H_k'$ – An Ex-Nor operation

Next, consider the carry output of each element, first C_{k-1} is held at logical 0, then

$C_k = A_kB_k + H_k . 0$

$C_k = A_kB_k$ - An And operation

Now if C_{k-1} is at logical 1, then

$C_k = A_kB_k + H_k . 1$

On solving $C_k = A_k + B_k$ - An Or operation

The adder element implementing both the arithmetic and logical functions can be implemented as shown in the figure 6.12.

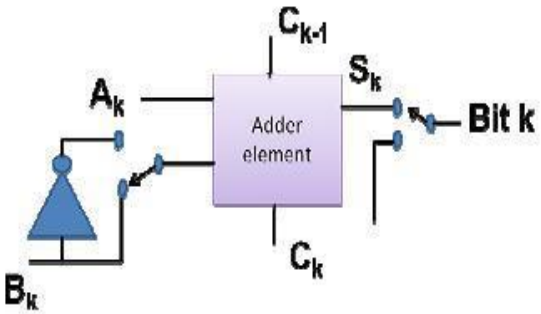


Figure 6.12: 1-bit adder element

The above can be cascaded to form 4-bit ALU.

A further consideration of adders

Generation:

This principle of generation allows the system to take advantage of the occurrences "a_k=b_k". In both cases (a_k=1 or a_k=0) the carry bit will be known.

Propagation:

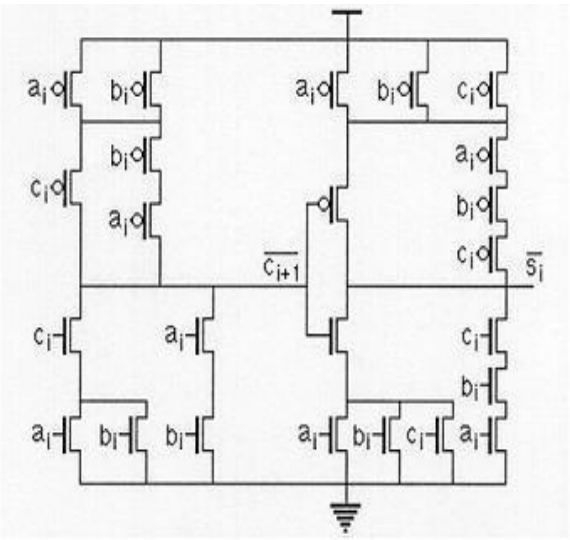
If we are able to localize a chain of bits $a_k\ a_{k+1}...a_{k+p}$ and $b_k\ b_{k+1}...b_{k+p}$ for which a_k not equal to b_k for k in [k,k+p], then the output carry bit of this chain will be equal to the input carry bit of the chain.

These remarks constitute the principle of generation and propagation used to speed the addition of two numbers.

All adders which use this principle calculate in a first stage.

$p_k = a_k$ XOR b_k

$g_k = a_k\ b_k$



Manchester carry – chain

This implementation can be very performant (20 transistors) depending on the way the XOR function is built. The carry propagation of the carry is controlled by the output of the XOR gate. The generation of the carry is directly made by the function at the bottom. When both input signals are 1, then the inverse output carry is 0.

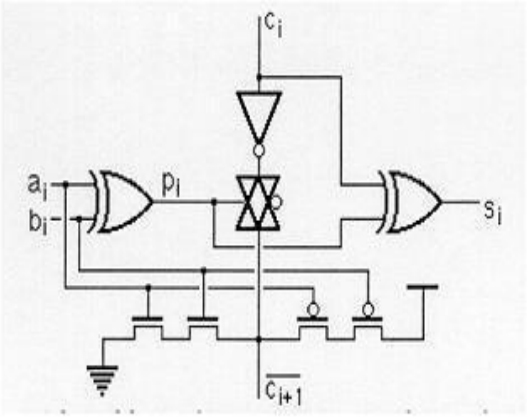


Figure-6.12: An adder with propagation signal controlling the pass-gate

In the schematic of Figure 6.12, the carry passes through a complete transmission gate. If the carry path is precharged to VDD, the transmission gate is then reduced to a simple NMOS transistor. In the same way the PMOS transistors of the carry generation is removed. One gets a Manchester cell.

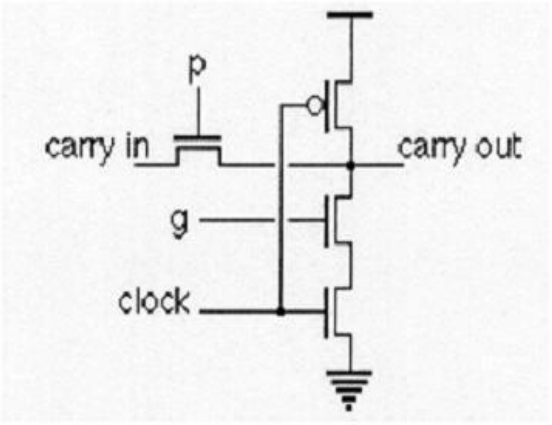


Figure-6.13: The Manchester cell

The Manchester cell is very fast, but a large set of such cascaded cells would be slow. This is due to the distributed RC effect and the body effect making the propagation time grow with the square of the number of cells. Practically, an inverter is added every four cells, like in Figure 6.14.

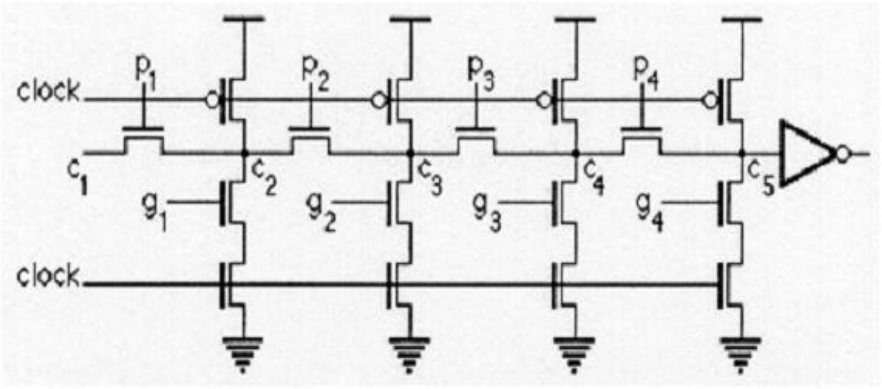


Figure-6.14**:** The Manchester carry cell

Adder Enhancement techniques

The operands of addition are the addend and the augend. The addend is added to the augend to form the sum. In most computers, the augmented operand (the augend) is replaced by the sum, whereas the addend is unchanged. High speed adders are not only for addition but also for subtraction, multiplication and division. The speed of a digital processor depends heavily on the speed of adders. The adders add vectors of bits and the principal problem is to speed- up the carry signal. A traditional and non optimized four bit adder can be made by the use of the generic one-bit adder cell connected one to the other. It is the ripple carry adder. In this case, the sum resulting at each stage need to wait for the incoming carry signal to perform the sum operation. The carry propagation can be speed-up in two ways. The first –and most obvious– way is to use a faster logic circuit technology. The second way is to generate carries by means of forecasting logic that does not rely on the carry signal being rippled from stage to stage of the adder.

The Carry-Skip Adder

Depending on the position at which a carry signal has been generated, the propagation time can be variable. In the best case, when there is no carry generation, the addition time will only take into account the time to propagate the carry signal. Figure 6.15 is an example illustrating a carry signal generated twice, with the input carry being equal to 0. In this case three simultaneous carry propagations occur. The longest is the second, which takes 7 cell delays (it starts at the 4th position and ends at the 11th position). So the addition time of these two numbers with this 16-bits Ripple Carry Adder is 7.k + k', where k is the delay cell and k' is the time needed to compute the 11th sum bit using the 11th carry-in.

With a Ripple Carry Adder, if the input bits Ai and Bi are different for all position i, then the carry signal is propagated at all positions (thus never generated), and the addition is completed when the carry signal has propagated through the whole adder. In this case, the Ripple Carry Adder is as slow as it is large. Actually, Ripple Carry Adders are fast only for some configurations of the input words, where carry signals are generated at some positions.

Carry Skip Adders take advantage both of the generation or the propagation of the carry signal. They are divided into blocks, where a special circuit detects quickly if all the bits to be added are different (Pi = 1 in all the block). The signal produced by this circuit will be called block propagation signal. If the carry is propagated at all positions in the block, then the carry signal entering into the block can directly bypass it and so be transmitted through a multiplexer to the next block. As soon as the carry signal is transmitted to a block, it starts to propagate through the block, as if it had been generated at the beginning of the block. Figure 6.16 shows the structure of a 24-bits Carry Skip Adder, divided into 4 blocks.

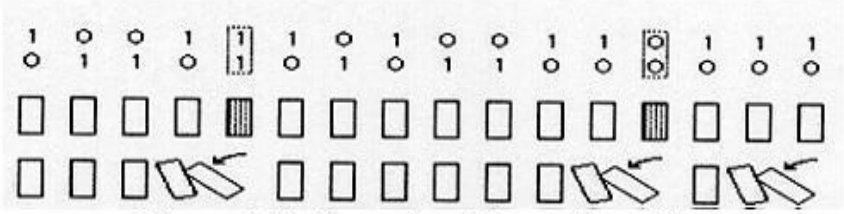


Figure 6.15: Example of Carry skip adder

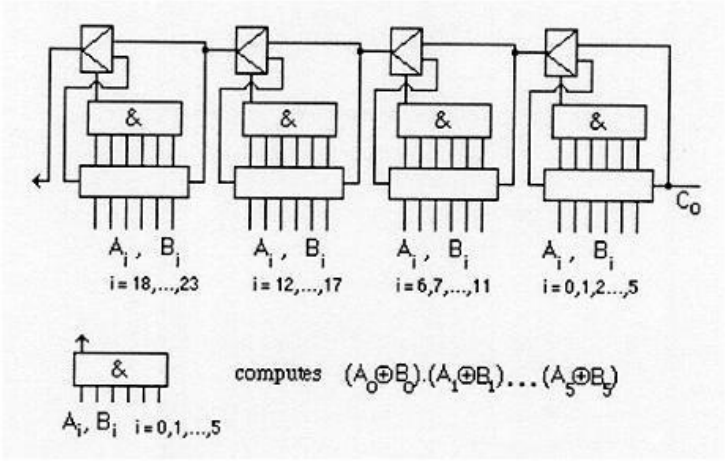


Figure-6.16: Block diagram of a carry skip adder

Optimization of the carry skip adder

It becomes now obvious that there exist a trade-off between the speed and the size of the blocks. In this part we analyze the division of the adder into blocks of equal size. Let us denote k1 the time needed by the carry signal to propagate through an adder cell, and k2 the time it needs to skip over one block. Suppose the N-bit Carry Skip Adder is divided into M blocks, and each block contains P adder cells. The actual addition time of a Ripple Carry Adder depends on the configuration of the input words. The completion time may be small but it also may reach the worst case, when all adder cells propagate the carry signal. In the same way, we must evaluate the worst carry propagation time for the Carry Skip Adder. The worst case of carry propagation is depicted in Figure 6.17.

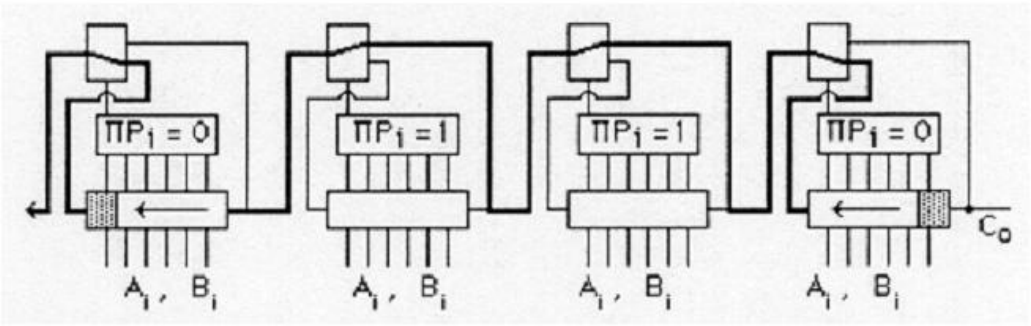


Figure-6.17: Worst case carry propagation for Carry Skip adder

The configuration of the input words is such that a carry signal is generated at the beginning of the first block. Then this carry signal is propagated by all the succeeding adder cells but the last which generates another carry signal. In the first and the last block the block propagation signal is equal to 0, so the entering carry signal is not transmitted to the next block. Consequently, in the first block, the last adder cells must wait for the carry signal, which comes from the first cell of the first block. When going out of the first

block, the carry signal is distributed to the 2nd, 3rd and last block, where it propagates. In these blocks, the carry signals propagate almost simultaneously (we must account for the multiplexer delays). Any other situation leads to a better case. Suppose for instance that the 2nd block does not propagate the carry signal (its block propagation signal is equal to zero), then it means that a carry signal is generated inside. This carry signal starts to propagate as soon as the input bits are settled. In other words, at the beginning of the addition, there exist two sources for the carry signals. The paths of these carry signals are shorter than the carry path of the worst case. Let us formalize that the total adder is made of N adder cells. It contains M blocks of P adder cells. The total of adder cells is then

$$N=M.P$$

The time T needed by the carry signal to propagate through P adder cells is

$$T=k_1.P$$

The time T' needed by the carry signal to skip through M adder blocks is

$$T'=k_2.M$$

The problem to solve is to minimize the worst case delay which is:

$$T_{worstcase} = 2 \cdot P \cdot k_1 + (M-2) \cdot k_2$$

$$T_{worstcase} = 2 \cdot \frac{N}{M} \cdot k_1 + (M-2) \cdot k_2$$

The Carry-Select Adder

This type of adder is not as fast as the Carry Look Ahead (CLA) presented in a next section. However, despite its bigger amount of hardware needed, it has an interesting design concept. The Carry Select principle requires two identical parallel adders that are partitioned into four-bit groups. Each group consists of the same design as that shown on Figure 6.18. The group generates a group carry. In the carry select adder, two sums are generated simultaneously. One sum assumes that the carry in is equal to one as the other assumes that the carry in is equal to zero. So that the predicted group carry is used to select one of the two sums.

It can be seen that the group carries logic increases rapidly when more high- order groups are added to the total adder length. This complexity can be decreased, with a subsequent increase in the delay, by partitioning a long adder into sections, with four groups per section, similar to the CLA adder.

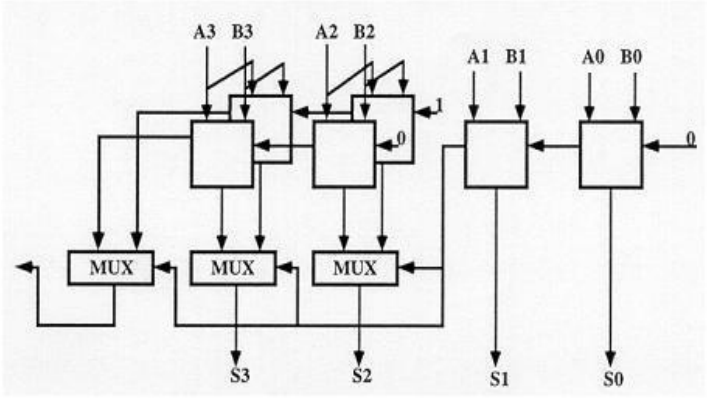


Figure-6.18: The Carry Select adder

Optimization of the carry select adder

- Computational time

$$T = K_1 n$$

- Dividing the adder into blocks with 2 parallel paths

$$T = K_1 n/2 + K_2$$

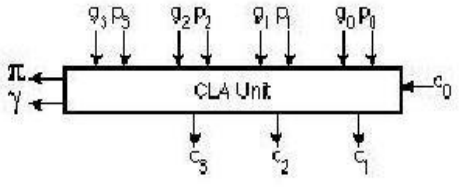
- For a n-bit adder of M-blocks and each block contains P adder cells in series $T = PK_1 + (M - 1) K_2$; $n = M.P$ minimum value for T is when $M=\sqrt{(K_1 n / K_1)}$

The Carry Look-Ahead Adder

The limitation in the sequential method of forming carries, especially in the Ripple Carry adder arises from specifying c_i as a specific function of c_{i-1}. It is possible to express a carry as a function of all the preceding low order carry by using the recursivity of the carry function. With the following expression a considerable increase in speed can be realized.

$$C_i = G_i + G_{i-2} P_{i-1} + G_{i-3} P_{i-2} P_{i-1} + \ldots \quad \ldots + G_0 P_1 P_2 \ldots P_{i-1} + C_0 P_0 P_1 P_2 \ldots P_{i-1}$$

Usually the size and complexity for a big adder using this equation is not affordable. That is why the equation is used in a modular way by making groups of carry (usually four bits). Such a unit generates then a group carry which give the right predicted information to the next block giving time to the sum units to perform their calculation.



$$\pi = p_0 p_1 p_2 p_3$$

$$\gamma = g_3 + p_3 g_2 + p_3 p_2 \cdot g_1 + p_3 p_2 p_1 \cdot g_0$$

$$c_4 = \gamma + \pi . c_0$$

Figure-6.19: The Carry Generation unit performing the Carry group computation

Such unit can be implemented in various ways, according to the allowed level of abstraction. In a CMOS process, 17 transistors are able to guarantee the static function (Figure 6.20). However this design requires a careful sizing of the transistors put in series.

The same design is available with less transistors in a dynamic logic design. The sizing is still an important issue, but the number of transistors is reduced (Figure 6.21).

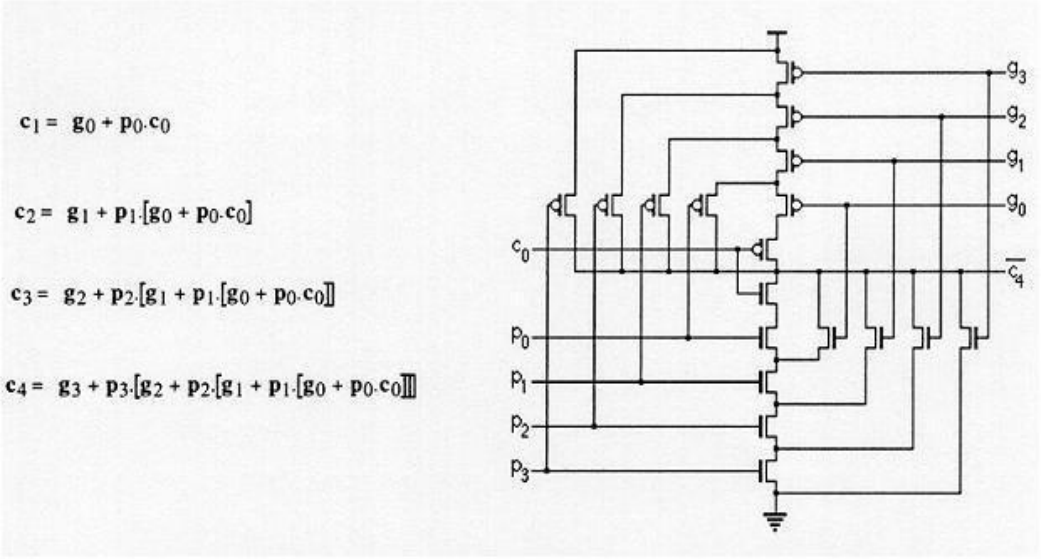


Figure-6.20: Static implementation of the 4-bit carry lookahead chain

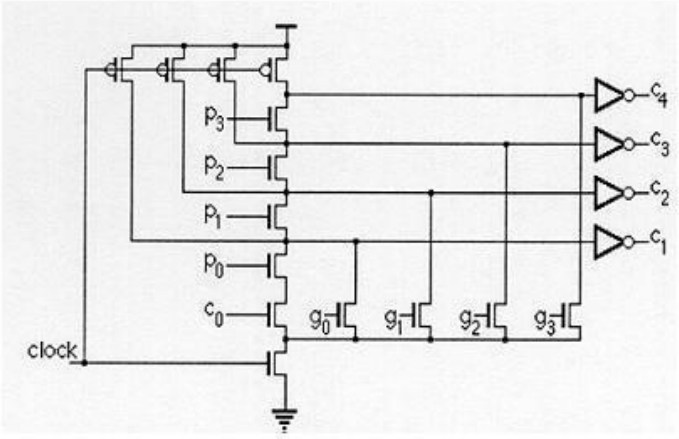


Figure-6.21: Dynamic implementation of the 4-bit carry lookahead chain

Figure 6.22 shows the implementation of 16-bit CLA adder.

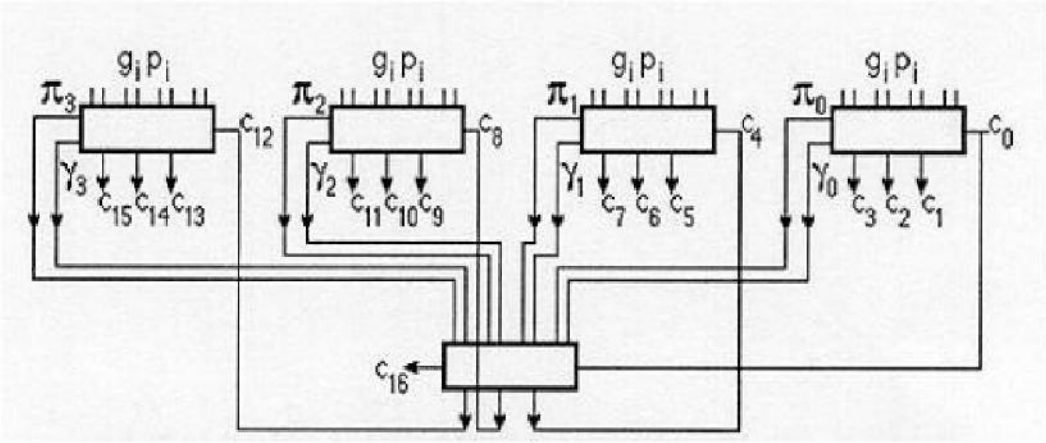


Figure-6.22: Implementation of a 16-bit CLA adder

Multipliers

Introduction

Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Each step of the addition generates a partial product. In most computers, the operands usually contain the same number of bits. When the operands are interpreted as integers, the product is generally twice the length of the operands in order to preserve the information content. This repeated addition method that is suggested by the arithmetic definition is slow that it is almost always replaced by an algorithm that makes use of positional number representation.

It is possible to decompose multipliers in two parts. The first part is dedicated to the generation of partial products, and the second one collects and adds them. As for adders, it is possible to enhance the intrinsic performances of multipliers. Acting in the generation part, the Booth (or modified Booth) algorithm is often used because it reduces the number of partial products. The collection of the partial products can then be made using a regular array, a Wallace tree or a binary tree

Serial-Parallel Multiplier

This multiplier is the simplest one, the multiplication is considered as a succession of additions.

$$\text{if } A = (a_n\, a_{n-1} \ldots\ldots a_0) \text{ and } B = (b_n\, b_{n-1} \ldots\ldots b_0)$$

The product A.B is expressed as :

$$A.B = A.2^n.b_n + A.2^{n-1}.b_{n-1} + \ldots + A.2_0.b^0$$

The structure of Figure 6.23 is suited only for positive operands. If the operands are negative and coded in 2's complement:

1. The most significant bit of B has a negative weight, so a subtraction has to be performed at the last step.

2. Operand $A.2^k$ must be written on 2N bits, so the most significant bit of A must be duplicated. It may be easier to shift the content of the accumulator to the right instead of shifting A to the left.

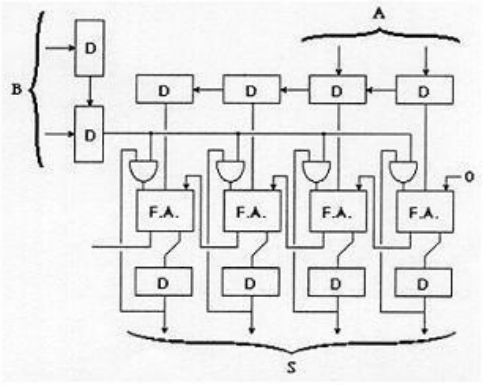


Figure-6.23: Serial-Parallel multiplier

Braun Parallel Multiplier

The simplest parallel multiplier is the Braun array. All the partial products A.bk are computed in parallel, and then collected through a cascade of Carry Save Adders. At the bottom of the array, the output of the array is noted in Carry Save, so an additional adder converts it (by the mean of carry propagation) into the classical notation (Figure 6.24). The completion time is limited by the depth of the carry save array, and by the carry propagation in the adder. Note that this multiplier is only suited for positive operands. Negative operands may be multiplied using a Baugh-Wooley multiplier.

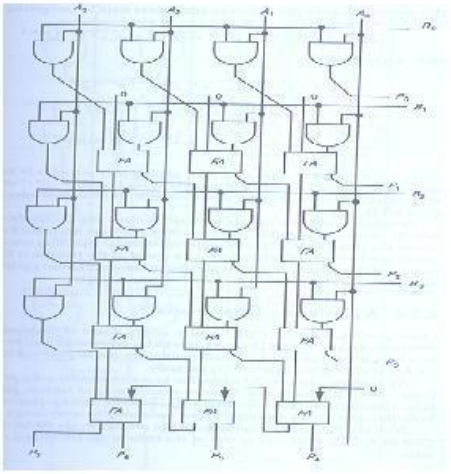


Figure 6.24: A 4-bit Braun Array

Baugh-Wooley Multiplier

This technique has been developed in order to design regular multipliers, suited for 2's-complement numbers.

Let us consider 2 numbers A and B:

$$A = (a_{n-1} \dots a_0) = -a_{n-1} \cdot 2^{n-1} + \sum_{0}^{n-2} a_i \cdot 2^i$$

$$B = (b_{n-1} \dots b_0) = -b_{n-1} \cdot 2^{n-1} + \sum_{0}^{n-2} b_i \cdot 2^i$$

The product A.B is given by the following equation:

$$A \cdot B = a_{n-1} \cdot b_{n-1} \cdot 2^{2n-2} + \sum_{0}^{n-2} \sum_{0}^{n-2} a_i \cdot b_i \cdot 2^{i+j} - a_{n-1} \sum_{0}^{n-2} b_i \cdot 2^{n+i-1} - b_{n-1} \sum_{0}^{n-2} a_i \cdot 2^{n+i-1}$$

We see that subtraction cells must be used. In order to use only adder cells, the negative terms may be rewritten as:

$$-a_{n-1} \sum_{0}^{n-2} b_i \cdot 2^{i+n-1} = a_{n-1} \cdot \left(-2^{2n-2} + 2^{n-1} + \sum_{0}^{n-2} \overline{b_i} \cdot 2^{i+n-1} \right)$$

By this way, A.B becomes:

$$A \cdot B = a_{n-1} \cdot b_{n-1} \cdot 2^{2n-2} + \sum_{0}^{n-2} \sum_{0}^{n-2} a_i \cdot b_j \cdot 2^{i+j}$$

$$+ b_{n-1} \left[-2^{2n-2} + 2^{n-1} + \sum_{0}^{n-2} \overline{a_i} \cdot 2^{i+n-1} \right]$$

$$+ a_{n-1} \left[-2^{2n-2} + 2^{n-1} + \sum_{0}^{n-2} \overline{b_i} \cdot 2^{i+n-1} \right]$$

The final equation is:

$$A.B = -2^{2n-1} + (\overline{a_{n-1}} + \overline{b_{n-1}} + a_{n-1}.b_{n-1}).2^{2n-2}$$

$$+\sum_{0}^{n-2}\sum_{0}^{n-2} a_i.b_j.2^{i+j} + (a_{n-1} + b_{n-1}).2^{n-1}$$

$$+\sum_{0}^{n-2} b_{n-1}.\overline{a_i}.2^{i+n-1} + \sum_{0}^{n-2} a_{n-1}.\overline{b_i}.2^{i+n-1}$$

because:

$$-(b_{n-1} + a_{n-1}).2^{2n-2} = -2^{2n-1} + (\overline{a_{n-1}} + \overline{b_{n-1}}).2^{2n-2}$$

A and B are n-bits operands, so their product is a 2n-bits number. Consequently, the most significant weight is 2n-1, and the first term -2^{2n-1} is taken into account by adding a 1 in the most significant cell of the multiplier. The implementation is shown in figure 6.25.

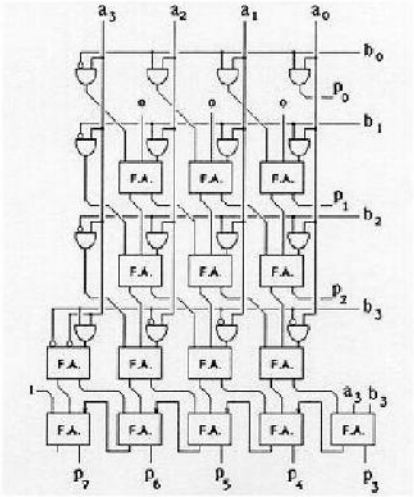


Figure-6.25: A 4-bit Baugh-Wooley Multiplier

Booth Algorithm

This algorithm is a powerful direct algorithm for signed-number multiplication. It generates a 2n-bit product and treats both positive and negative numbers uniformly. The idea is to reduce the number of additions to perform. Booth algorithm allows in the best case n/2 additions whereas modified Booth algorithm allows always n/2 additions.

Let us consider a string of k consecutive 1s in a multiplier:

..., i+k, i+k-1, i+k-2 , ..., i, i-1, ...
..., 0 , 1 , 1 , ..., 1, 0, ...

where there is k consecutive 1s.

By using the following property of binary strings:

$$2^{i+k}-2^{i}=2^{i+k-1}+2^{i+k-2}+...+2^{i+1}+2^{i}$$

the k consecutive 1s can be replaced by the following string

..., i+k+1, i+k, i+k-1, i+k-2, ..., i+1, i , i-1 , ...
..., 0 , 1 , 0 , 0 , ..., 0 , -1 , 0 , ...
k-1 consecutive 0s Addition Subtraction

In fact, the modified Booth algorithm converts a signed number from the standard 2's-complement radix into a number system where the digits are in the set {-1,0,1}. In this number system, any number may be written in several forms, so the system is called redundant.

The coding table for the modified Booth algorithm is given in Table 1. The algorithm scans strings composed of three digits. Depending on the value of the string, a certain operation will be performed.

A possible implementation of the Booth encoder is given on Figure 6.26.

Table-1: Modified Booth coding table

	BIT			M is
2^1	2^0	2^{-1}	OPERATION	multiplied
Y_{i+1}	Y_i	Y_{i-1}		by
0	0	0	add zero (no string)	+0
0	0	1	add multipleic (end of string)	+X
0	1	0	add multiplic. (a string)	+X
0	1	1	add twice the mul. (end of string)	+2X
1	0	0	sub. twice the m. (beg. of string)	-2X
1	0	1	sub. the m. (-2X and +X)	-X
1	1	0	sub . the m. (beg. of string)	-X
1	1	1	sub. zero (center of string)	-0

.

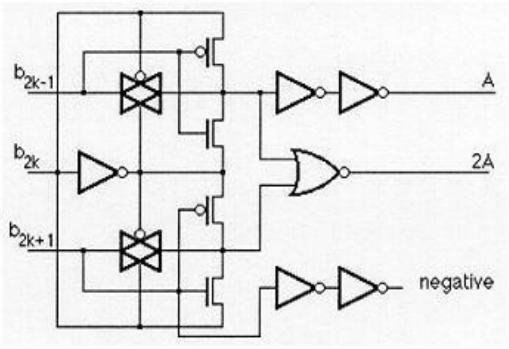


Figure-6.26: Booth encoder cell

To summarize the operation:

- Grouping multiplier bits into pairs
  - Orthogonal idea to the Booth recoding
  - Reduces the num of partial products to half
  - If Booth recoding not used ➔ have to be able to multiply by 3 (hard: shift+add)
- Applying the grouping idea to Booth ➔ Modified Booth Recoding (Encoding)
  - We already got rid of sequences of 1's ➔ no multiplication by 3
  - Just negate, shift once or twice

Wallace Trees

For this purpose, Wallace trees were introduced. The addition time grows like the logarithm of the bit number. The simplest Wallace tree is the adder cell. More generally, an n-inputs Wallace tree is an n-input operator and log2(n) outputs, such that the value of the output word is equal to the number of "1" in the input word. The input bits and the least significant bit of the output have the same weight (Figure 6.27). An important property of Wallace trees is that they may be constructed using adder cells. Furthermore, the number of adder cells needed grows like the logarithm log2(n) of the number n of input bits. Consequently, Wallace trees are useful whenever a large number of operands are to add, like in multipliers. In a Braun or Baugh-Wooley multiplier with a Ripple Carry Adder, the completion time of the multiplication is proportional to twice the number n of bits. If the collection of the partial products is made through Wallace trees, the time for getting the result in a carry save notation should be proportional to log2(n).

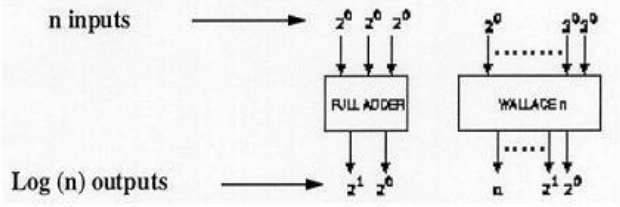


Figure-6.27: Wallace cells made of adders

Figure 6.28 represents a 7-inputs adder: for each weight, Wallace trees are used until there remain only two bits of each weight, as to add them using a classical 2-inputs adder. When taking into account the regularity of the interconnections, Wallace trees are the most irregular.

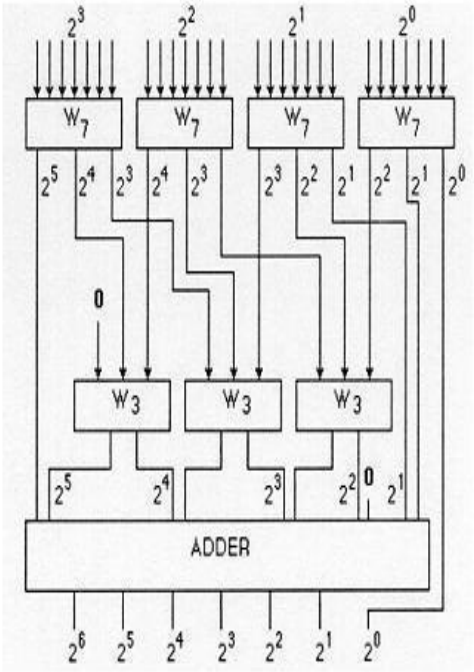


Figure-6.28: A 7-inputs Wallace tree

To summarize the operation:

The Wallace tree has three steps:

- Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, yielding n^2 results.
- Reduce the number of partial products to two by layers of full and half adders.
- Group the wires in two numbers, and add them with a conventional adder.

The second phase works as follows.

- Take any three wires with the same weights and input them into a full adder.
- The result will be an output wire of the same weight and an output wire with a higher weight for each three input wires.
- If there are two wires of the same weight left, input them into a half adder.
- If there is just one wire left, connect it to the next layer.

VHDL SYNTHESIS:

* VHDL is acronym as V-VHSIC-Very High speed Integrated circuit
HDL - Hardware Description language.

* VHDL is developed to describe the complex digital circuits such as Field programmable gate arrays and integrated circuits.

* It is a 'C' like language that supports wide range of description styles (Structural description, Data flow description, Behavioral description and mixed description).

* The synthesis tool will infer a hardware architecture from the VHDL model.

* Synthesis is the process of Converting RTL (synthesizable VHDL Code) to technology specific gate level net list (includes nets, sequential, and combinational cells and their connectivity).

* Synthesis is nothing but TOM - Translate, optimize, map.

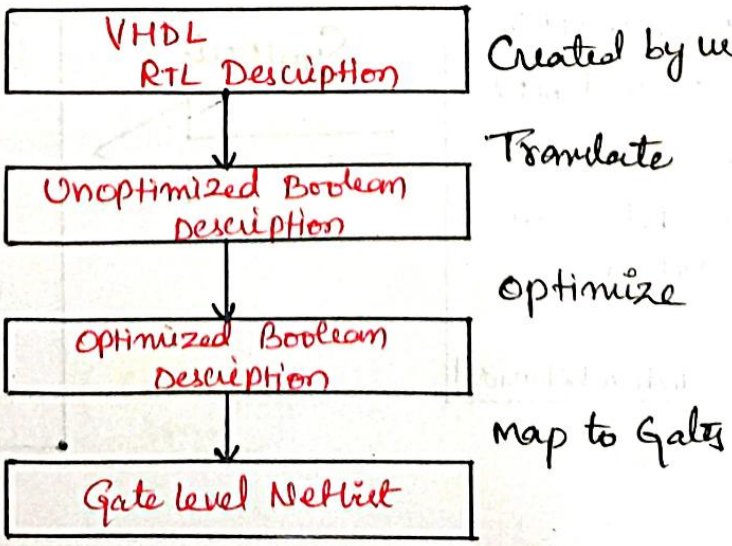


Fig: Synthesis process

Scanned by Scanner Go

VHDL RTL Description:

* RTL is an Acronym of Register Transfer level. This implies that your VHDL code describes how data is transferred as it is passed from register to register.

Unoptimized Boolean Description:

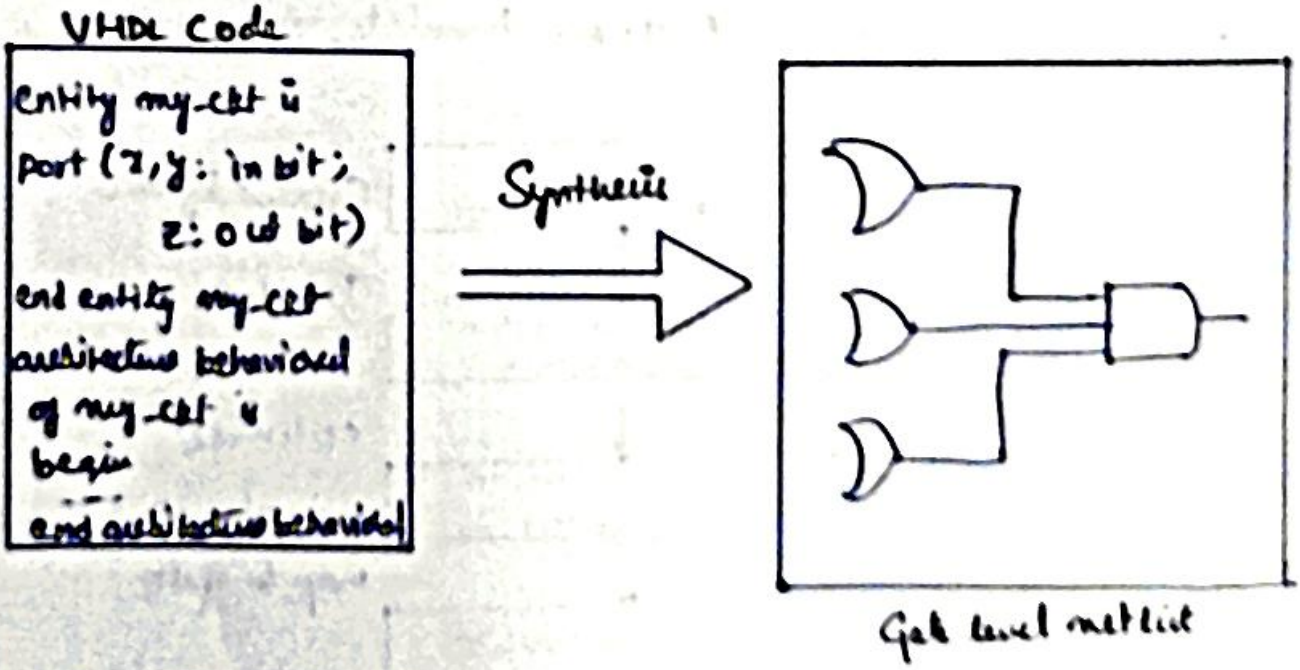
* It is defined that the following Boolean Description is not in a optimized manner, so it takes more amount of gate to construct a circuit.

Optimized Boolean Description:

* The Boolean description will be optimized so it takes less amount of gate to construct a circuit.

Gate level netlist:

* Synthesis is the process of converting RTL to technology specific gate level netlist (includes net, sequential and combinational cells and their connectivity).



Scanned By Scanner Go

SYNTHESIS PROCESS:

* The inputs to the synthesis process are RTL, VHDL description, Circuit constraints and attributes for the design and technology library.
* The synthesis process, as shown in the diagram, produces optimized gate level netlist and schematic of the design

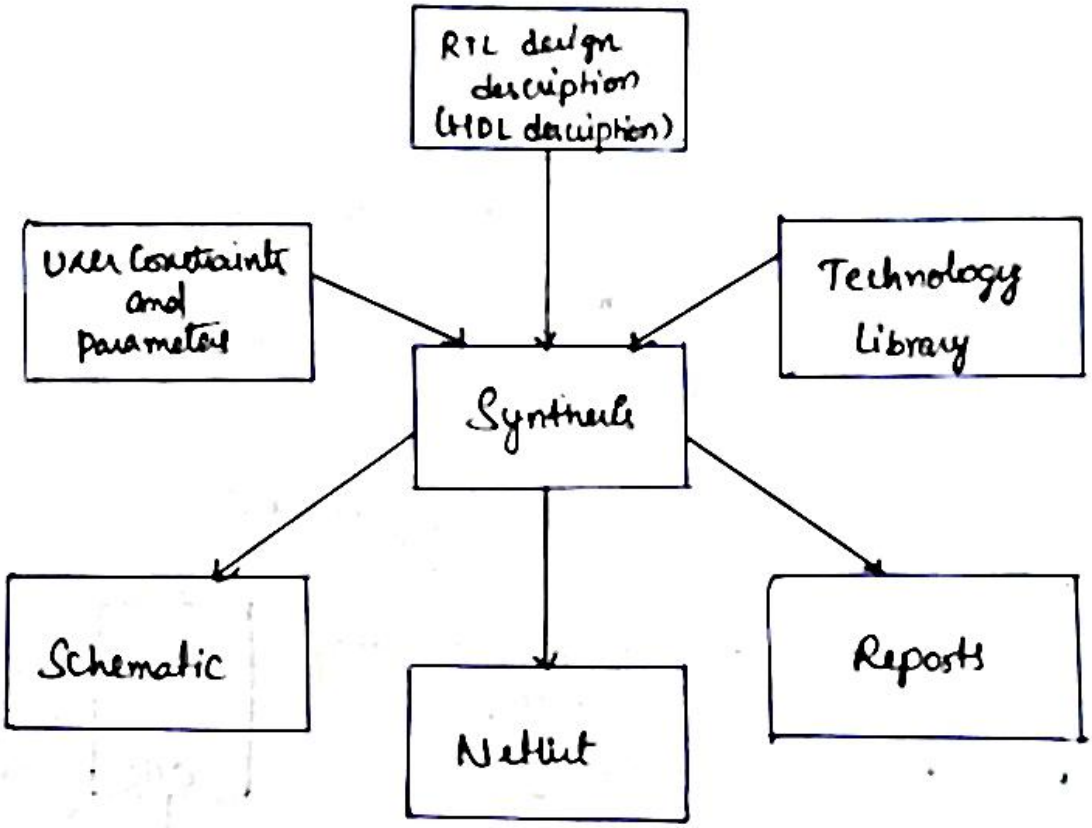
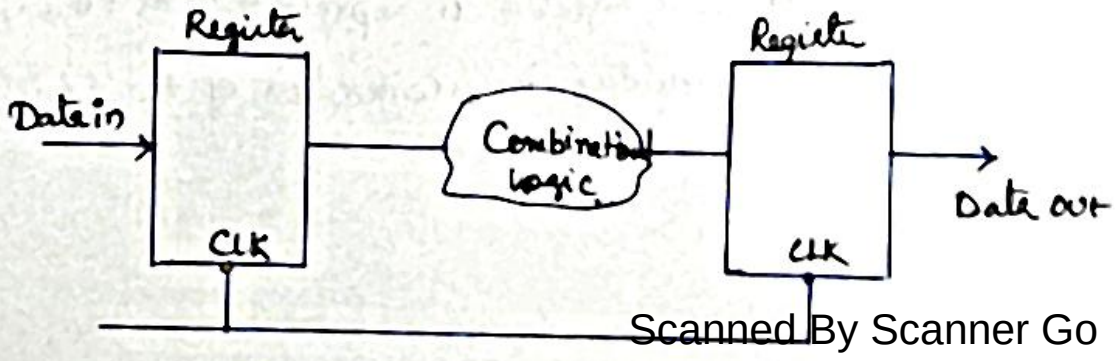


FIG: SYNTHESIS PROCESS

1. Register Transfer level description:

* It is characterized by a style that specifies all of the registers in a design, and combinational logic in between.
* This can be shown with Register and Cloud diagram.



Scanned By Scanner Go

* The Registers are described either explicitly through Component Instantiation or implicitly through interference.

* The Combinational logic is described by logical equations, sequential control statements, Subprograms or with Concurrent statements which are represented by the cloud objects in between the Registers.

CONSTRAINTS:-

* Constraints are used to control the o/p of the optimization and mapping process.

* The constraints available in synthesis tools today include area, timing, power and testability constraints.

* A block diagram of a design with some possible Constraints is as shown.

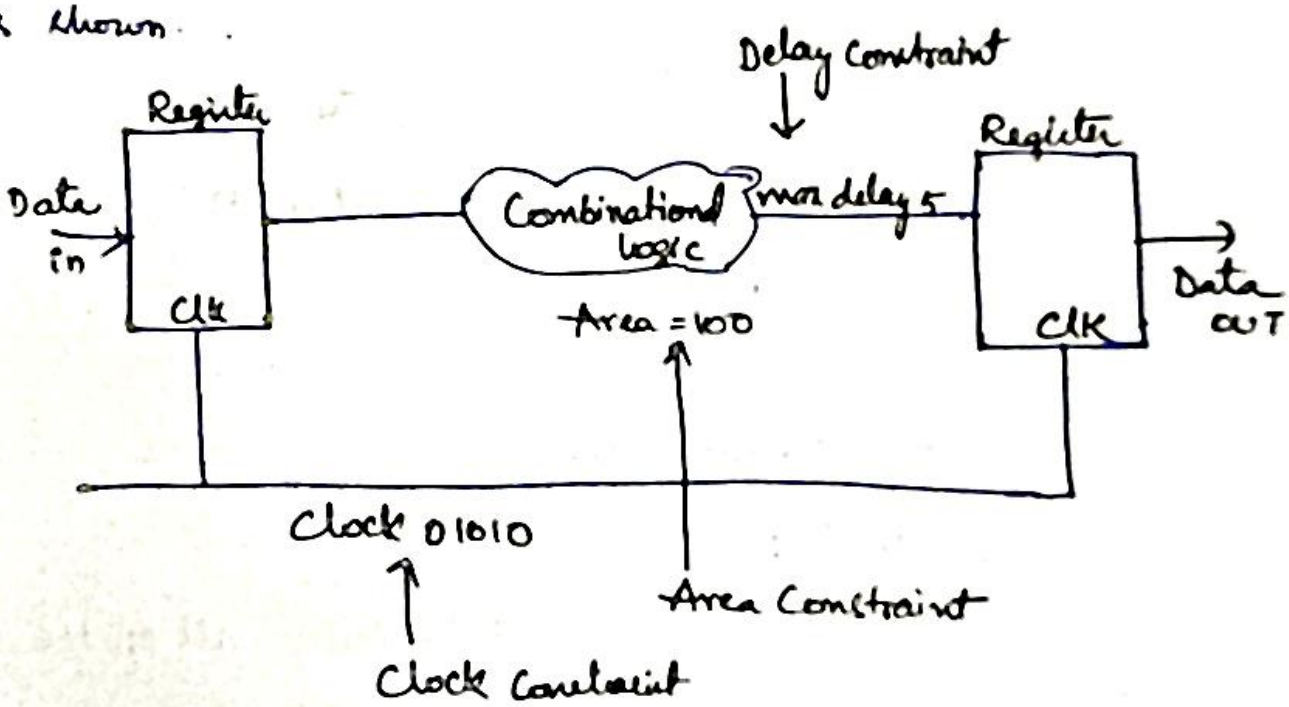


Fig: Register and cloud diagram with Constraints.

* The Combinational logic between Registers is represented as clouds, with wires going in and out representing Interconnection of the registers.

Scanned By Scanner Go

TIMING CONSTRAINTS:

* Time constraints specify the latest time that a signal can occur.
* Typical uses of timing constraints are to specify maximum delays for particular paths in the design.
* The timing constraint guides the optimization and mapping to produce a Netlist which meets the timing constraint.

CLOCK CONSTRAINT:

* One of the method to constrain a design is to add required time constraint to every flipflop input with the value of a clock cycle.
* The resulting design could be optimized to meet the one clock cycle timing constraint.
* A clock constraint effectively adds an i/p required-time constraint to every flipflop data i/p.

Technology Libraries:

* Technology libraries holds all the information necessary for a synthesis tool to create a netlist for a design based on the desired logical behaviour and constraints on the design.
* Technology libraries contain all the information that allows the synthesis process to make the correct choices to build a design.

Scanned By Scanner Go

* Some tools are required for the process of synthesis. The tools are, 1. Xilinx
2. Cadence

CIRCUIT DESIGN FLOW:

* In this design flow, synthesis is the process of creating a gate level description of the blocks that are described behaviorally in VHDL and preparing the complete design for the place and route process in a physical device.

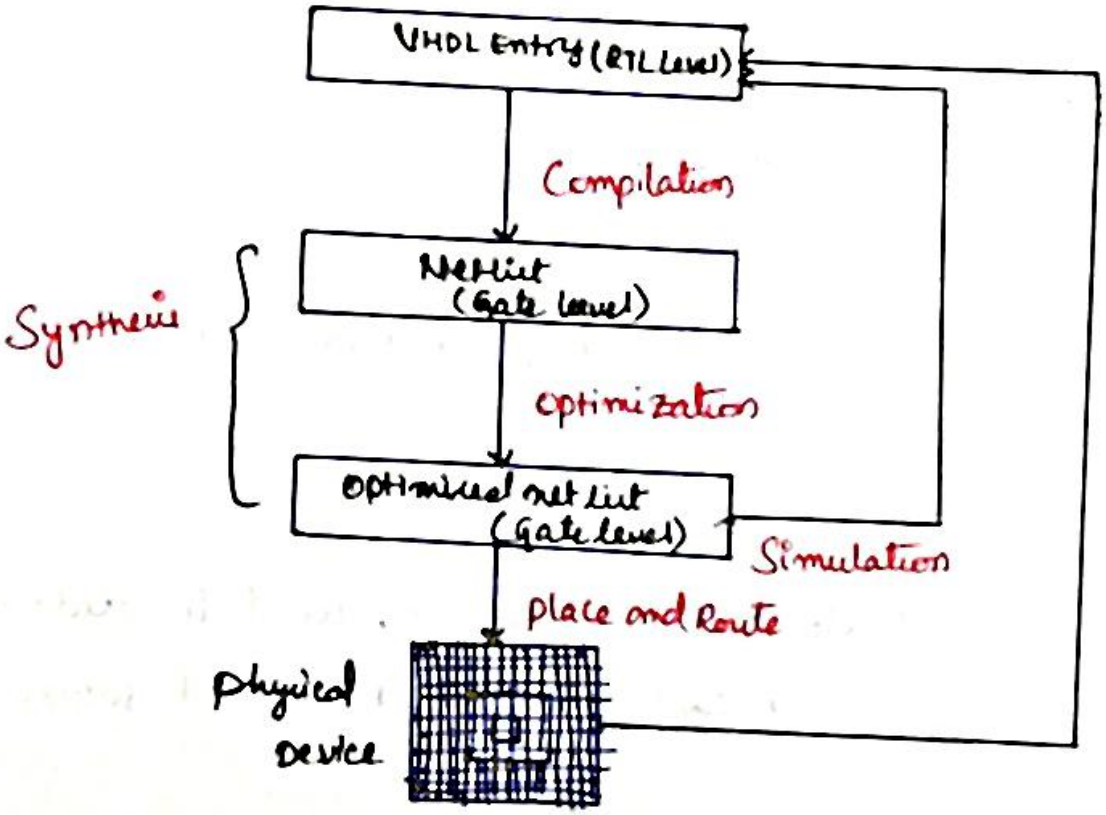


FIG: CIRCUIT DESIGN FLOW

RTL:

* It implies that your VHDL code describes how data is transformed as it is passed from register to register.

Scanned By Scanner Go

Compilation:

* Compilation is the process of Reading in source code and analyzing the source code for syntax and semantic errors.

Gate level netlist:

* Synthesis is the process of converting RTL to technology specific gate level netlist.

Optimization:

* Optimization is the process of finding an equivalent representation of the specified logic circuit.
* Generally the circuit is constrained to minimum chip area meeting a minimum delay.

Simulation:

* Simulation is the process of giving input and verifying the functionality of Circuit.

Place and Route:

* A place and Route tool is software, used to automatically map & fit synthesized logic to a target physical devices.

Scanned By Scanner Go

CIRCUIT SYNTHESIS:

* The designer describes the design at a high level by using RTL Constructs. The designer spends time in functional verification to ensure that the RTL description spends time in functions correctly. After the functionality is verified, RTL description is I/p to the logic synthesis tool.
* The steps followed in synthesis process are given below.

1. **TRANSLATION :-**

* The RTL description is converted by the logic synthesis tool to an unoptimized, intermediate, internal Representation. This process is known as translation.
* Translation is not user controllable. It is relatively simple and uses techniques of HDL Constructs Interpretation
* Interpretation is a process which converts all Conditional & sequential and Concurrent statements to boolean equivalent format.
* Translator understands the basic primitives and operators in the HDL description.
* Design Constraints such as area, timing and power are not Considered. At this point, the logic synthesis tool does a simple allocation of internal resources.

2. **BOOLEAN OPTIMIZATION:**

* The optimization process takes an unoptimized boolean description and converts it to an optimized boolean description.

Scanned By Scanner Go

* The optimization process uses a number of algorithms and rules to convert the unoptimized boolean description to an optimized one.

* Optimization is the process which decreases the area & increases the speed of a design.

3 FLATTENING:

* The process of converting unoptimized boolean description to a PLA format (low level description) is known as flattening, because it creates a flat signal representation of only two levels - an AND level and an OR level.

* An example of a boolean description is shown here

Original equation

a = b and c;

b = x or (y and z);

c = q or w;

* The flattening process removes these intermediate nodes to produce a completely flat design, with no intermediate nodes.

* After removing intermediate variables;

a = (x and q) or (q and y and z) or
(w and x) or (w and y and z);

* This second example is boolean equivalent of the first, but it has no intermediate nodes.

Scanned By Scanner Go

4. FACTORING:

* Factoring is the process of adding intermediate terms to add structure to a description. It is exactly opposite to flattening process.

* See following example before factoring

x = a and b or a and d

y = z & b or d;

* After factoring the common terms (b or d), is factored out to a separate intermediate node. The results are as shown below.

x = a and q;

y = z or q;

q = b or d;

* Factoring generally produces a better design but can be very design dependent.

* After the design has been optimized at the boolean level, it can be mapped to the gate functions in a technology library.

5. Mapping to Gate:

* The mapping process takes the logically optimized boolean description created by the optimization step and uses the logical and timing information from a technology library to build a netlist.

* This netlist is targeted to the user needs for area and speed. There are a number of possible netlists that are functionally same but vary widely in speed and area.

Scanned By Scanner Go

* The mapping process takes as input the optimized boolean description, the technology library and the user constraints and generates an optimized netlist built entirely from cells in the technology library.

* During mapping process, cells are inserted while implement the boolean function from the optimized boolean description. These cells are then locally optimized to meet speed and area requirements.

SIMULATION:

* Simulation is the fundamental and essential part of the design process for any electronic based product.

* Simulation is the process of applying stimuli (test i/p) to design under test over same duration of time and producing corresponding response from the design under test.

* The software used for simulation is known as "Simulator".

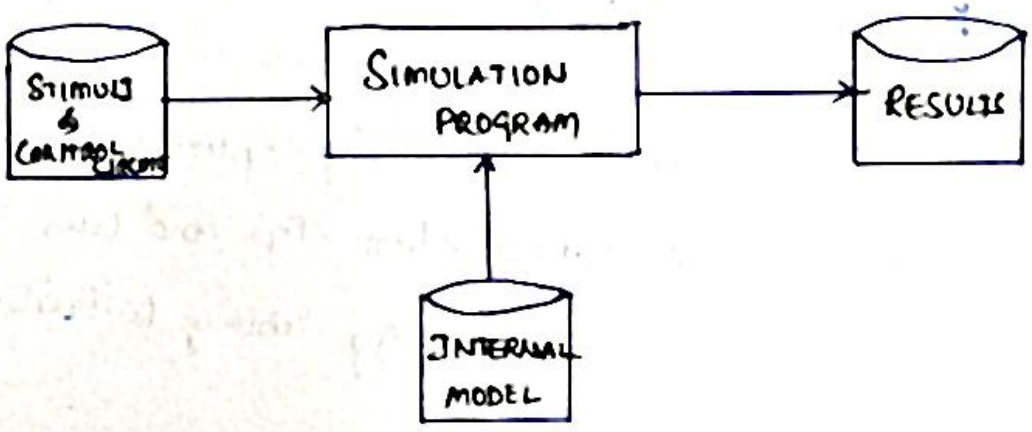


FIG: SIMULATION PROCESS

* The verification is done by comparing the results obtained by simulation with the expected results provided by the specification.

Scanned By Scanner Go

TYPES OF SIMULATION

* The level of simulation corresponds to the level of modeling employed to represent the simulated system. There, we have following types

1. Register level simulation
2. Behavioral simulation
3. Functional simulation
4. Gate-level simulation
5. Transistor level simulation
6. Switch level simulation
7. Mixed level simulation
8. Static timing analysis
9. Timing simulation

1. Register level simulation: RTL level simulation allows verifying or simulating a description for systems modeled entirely in RTL or as an interconnection of components modeled in RTL.
2. Behavioral simulation: Behavioral simulation models large ~~spec~~ pieces of a system as black boxes with inputs and output. It only specifies behavior of system in terms of its input-output relationship
3. Functional simulation: Functional simulation is the process of identifying logic errors in design for systems modeled as an interconnection of primitive functional blocks. It verifies only the functionality of the design.
4. Gate-level simulation: Gate level simulation can be used to check the timing performance after synthesis using gate level netlist. In a gate level simulation, a logic gate is treated as black box modeled by a function ~~where~~ where inputs are I/P signal

Scanned By Scanner Go

5. Transistor level simulation: This type of simulation is based on the behavioral model of transistor. This is complex and time consuming form of simulation. A transistor level simulation requires models of transistor describing their non-linear and current characteristics.

6. Switch level simulation: In this type of simulation, models of transistors are taken as switches. Switch level simulation predicts accurate timing than gate level simulation. This is possible without the ability to use logic cell delays as parameters of the models.

7. Mixed level simulation: This type of simulation permits different parts of design simulation to use different simulation modes.

8. Static timing analysis: A static timing analysis is a point to point delay analysis of a design networks.

9. Timing simulation: The purpose of this simulation is to check the dynamic timing behaviour of any design of target technology. Timing simulation is normally performed after synthesis and place and route using back annotation information.

Scanned By Scanner Go

LAYOUT:

* Once a network is available, it is then converted into layout. Layout is prepared by Software.

* There are two main phases in it.

1. Placement.
2. Routing.

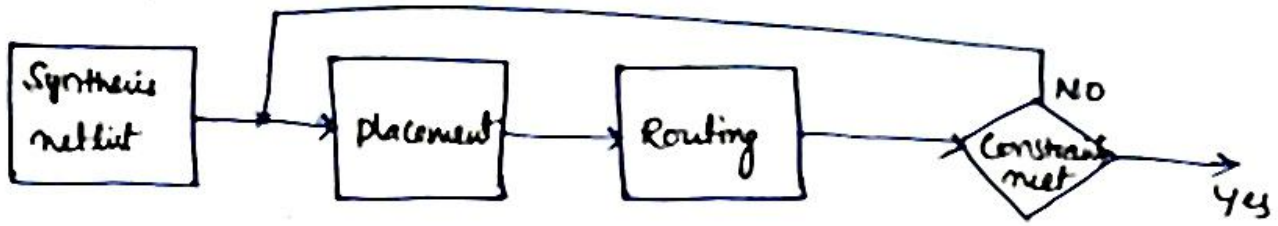


FIG: PLACE AND ROUTE PROCESS

PLACEMENT:

* Placement is task of placing modules such that to minimize area.

* The synthesis netlist is input to the placement process. The placement process analyzes all the macros used in the design and their connectivity to try to determine an optimal solution placement for the macros.

* Typical target devices have areas of the chip where logical functions are placed and areas where interconnect signals are routed to connect the logical functions.

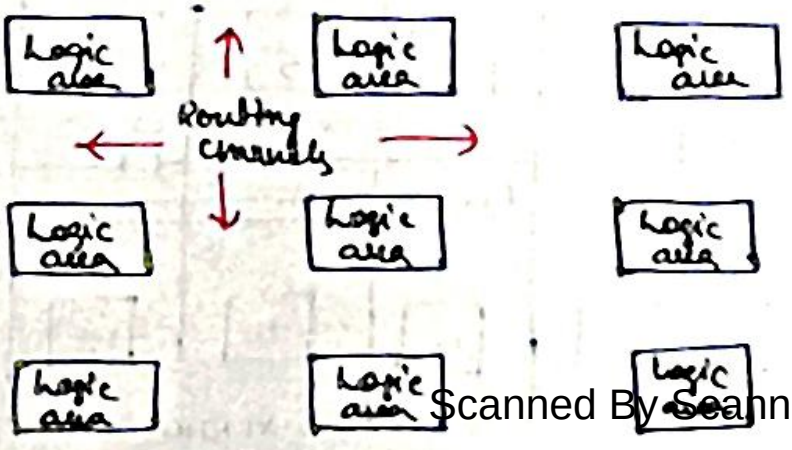


FIG: FPGA CHIP ARCHITECTURE.

Scanned By Scanner Go

* The device is split into a number of logic area with routing channels which surround the logic areas.

* By enabling the proper sets of pass transistor gates, signal interconnections between logic gates can be formed as shown.

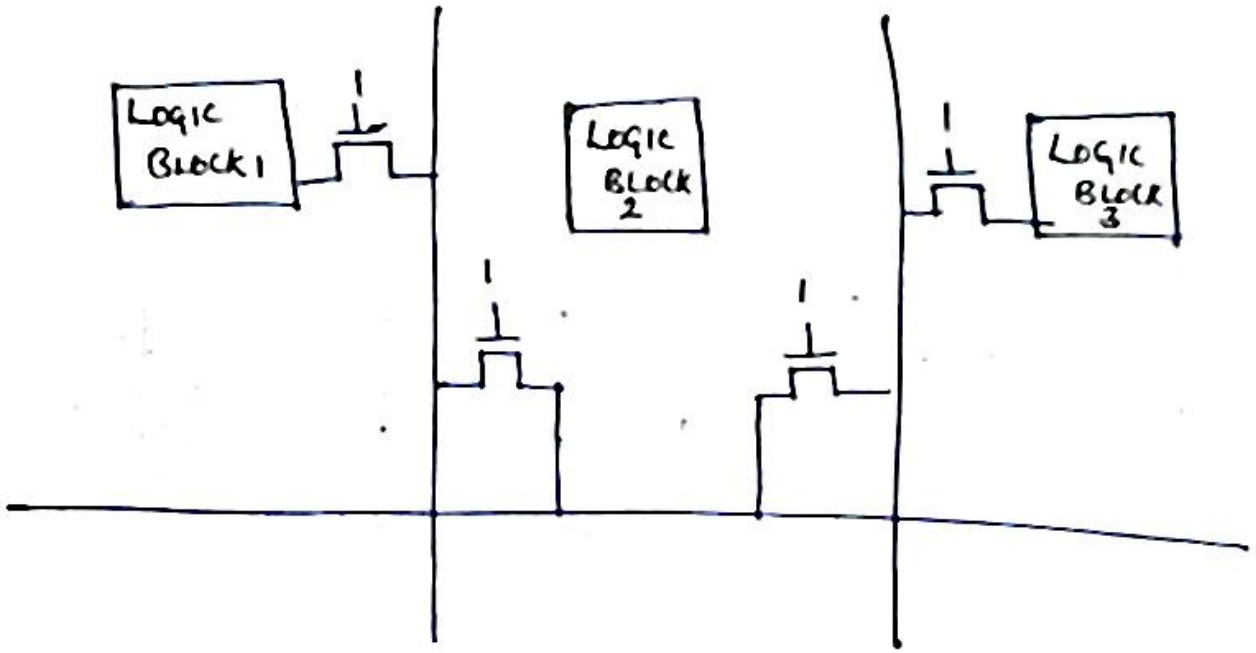


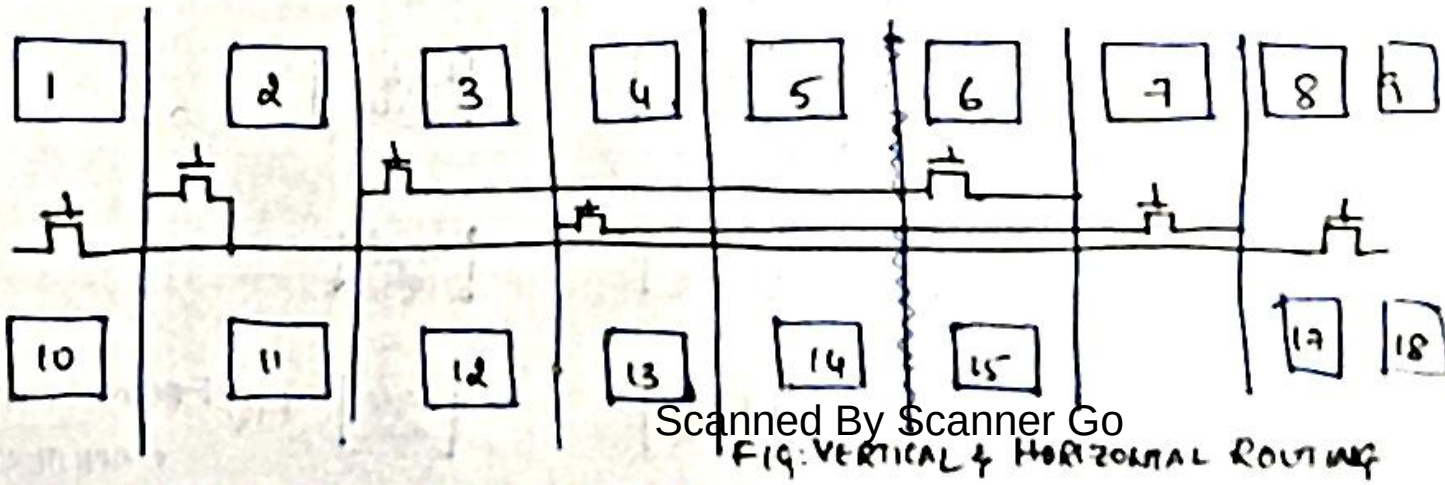
FIG: LOGIC BLOCK INTERCONNECTION

* To make a connection from logic block 1 to logic block 3, all the switches shown need to be enabled with a logic value 1.

ROUTING:

* A Router takes a module placement and a list of connections and connects the modules and wires.

* The routing channels contain vertical and horizontal lines the horizontal wires connect devices within a row, while the vertical lines allow connections across rows.

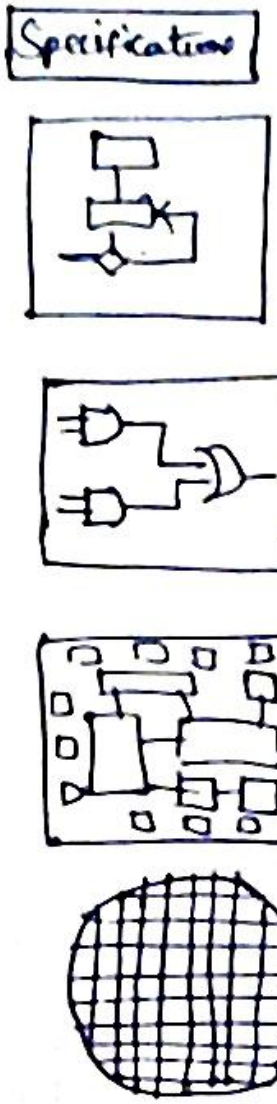


Scanned By Scanner Go

FIG: VERTICAL & HORIZONTAL ROUTING

DESIGN CAPTURE TOOLS

* Design entry tool encapsulates a circuit description. These tools capture a design and prepare it for simulation.



* The following tools represent the design capture tools.

1. Editors
2. VHDL / Verilog
3. System verilog / SystemC
4. State charts
5. FSM Capture.

Scanned By Scanner Go

1. HDL DESIGN :-

* Majd drawback of traditional design method is manual method of translation of design description into group of logic equations.

* This monotonous procedure can be entirely eliminated with HDL design as it allows the use of FSM for sequential systems and truth table for Combinational Circuits.

* Thus any design description can automatically Converted into HDL code that can be implemented by using synthesis tools.

* HDLs are used for to design two kinds of systems.

i. Integrated Circuits

ii. Programmable logic devices.

2. Schematic Design:

* Schematic Design provides a means to draw and connect Components. An icon is defined for a Collection of Components within a module. The Icon can be used in another module.

* Schematic Editors are available with various features like.

1. Creating, Selecting and deleting parts by pointing.

2. Changing the graphic view by panning, Zooming.

3. Floorplanning:

* Arranging blocks of layout within a chip to minimize area or maximize speed is called Floorplanning.

Scanned By Scanner Go

Rules for Floorplanning

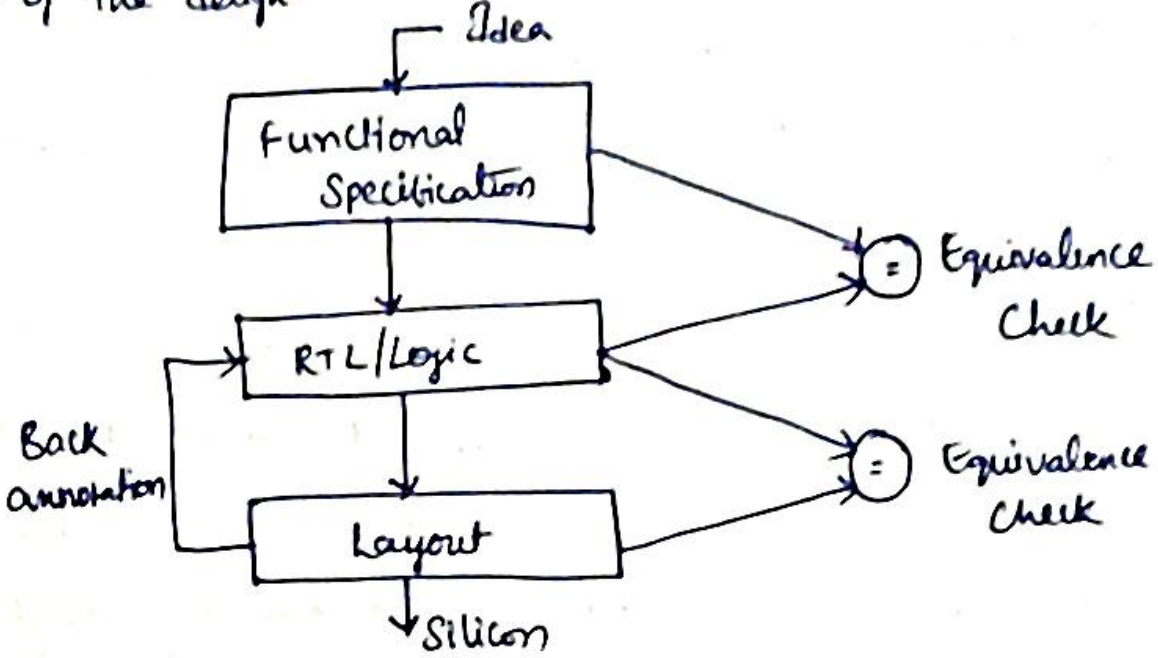
1. Make a block for small components: Small components create a lot of problems in floorplanning. Small components require extra effort in placing the components which disrupts the flow of wires across the chip. Therefore small components are combined to form a large block.
2. Developing a wiring plan: Different directions are used for different layers. A plan is to be sketched on graph paper before assigning layers to the net.
3. Designing Simple wiring plan: Wiring plan with conveniently placed wires is designed. When a block is to be moved then move pin locations to simplify routing technique.
4. Designing a planar wiring: When all the nets are routed in the plane without crossing it is called Planar wiring. First design a floorplan with all important signals for planar wiring then less critical signals can be added to it.
5. Wiring for power and clock signals: power and clock signals require a separate wiring plane which can be verified for design considerations.

Design Verification Tools:

* Design verification tools, which verifies correctness of design.
* These tools analyze design and find problems. It is responsible for quality of design.
* There are two major tool sets for verification:
  1. Functional (logic)
  2. Timing verification

Scanned By Scanner Go

* Functional verification verify the logical behavior of a design based on design entry.
* Timing verification verify the real behavia of the circuit with circuit delays.
* It is always important to verify equivalence between different levels of the design.



* The functionality of cmos chip is to be verified Certain set of verification tools are used for testing functional specification. Following tools are popular for design verification

1. Simulation
    a) Circuit-level simulation
    b) Timing simulation
    c) logic-level simulation
    d) Switch level simulation
    e) Mixed-mode simulation
2. Timing verifiers
3. Network Isomorphism
4. Netlist Comparison

Scanned By Scanner Go

5 Layout Extraction

6. Back-annotation

7. Design Rule Verification

Schematic Rule Check (SRC):

* In cell based design a schematic rule checker (SRC) is used to verify the schematics i.e., Schematic Rule violation. The violation of rule may be indicated in terms of warning & errors.
* SRC warning is indicated because of following probabilities
  i. Floating wire segments.
  ii. Open Connection.
  iii. Higher fanout.
* SRC error is indicated because of following probabilities.
  i. Undefined i/p's / open inputs.
  ii. Unmatched bus Connections.
  iii. Different I/O pins.
  iv. multiple drivers connection to single line.

Design Rule Check (DRC):

* The mask database provides interface between the semiconductor and chip designer. Two important requirements for this interface are
  1. Specified geometric design.
  2. Inter-relationships of the mask.
* The test for above two requirements are carried out by a CAD tool called Design Rule Checker (DRC).
* Two different categories of DRC programs are used, these are

Scanned By Scanner Go

1. Polygonal Check
2. Raster Scan Check

* Polygonal check are popularly used by semiconductor industry for defining connected areas for the layers of interest.

* Raster scan design rule checks involves comparatively simple computing as compared to polygonal rule checks.

* The raster scan check paces small bitters over a rasterized image of the integrated circuit.

Scanned By Scanner Go

TESTING & TESTABLITY

* A failure is said to have occured in a circuit & system if it deviates from its specified behavior.

FAULT:

* Representation of defect affecting physical condition that cause circuit to fail to perform in required manner is defined as 'Fault'

Detected fault: A fault for which a valid test vector has been generated

Test vector: An input vector for the circuit under test that cause the presence of fault observable at the primary output.

Undected fault: A fault for which a test vector has not been generated

Fault Coverage: Fault Coverage is defined as the percentage of fault that can be detected by the applied Test vector. Fault coverage gives a measure of goodness of a test program.

*/. Fault Coverage = Number of fault detected / total nodes in the circuit

Permanent fault: A fault that is occured due to breaking of a Component & wire is called as permanent fault.

Scanned By Scanner Go

Transient faults: The fault which is occured due to some power supply fluctuations.

Intermediate faults: This kind of fault is occured due to the Component misbehavior sometimes this turns into permanent fault

Fault detection: The task of determining whether a fault is present & not is called fault detection

FAULT MODELLING:

* In general the effect of a fault is represented by means of a model, which represent the change the fault produces in circuit signals.
* These models are used to identify the physical defect within the Circuit & a system.
* The classification of fault models

1. Single Stuck-at faults.
2. multiple stuck-at-faults.
3. Stuck-open fault models.
4. Stuck-short fault models.
5. Bridging fault models
6. Delay fault models

1. SINGLE STUCK AT FAULTS:

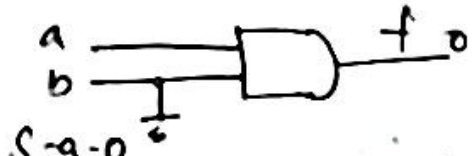
* Single stuck-at faults have two faults per line and they are
1. S-a-1 (Stuck at 1)
2. S-a-0 (Stuck at 0)

Scanned By Scanner Go

* Three properties define a single Stuck-at-fault

1. only one line is faulty.
2. The faulty line is permanently set to 0 or 1.
3. The fault can be at an i/p or o/p of a gate

* When the signal line always present at logical '0' is referred as Stuck-at-0 fault.



Stuck-at-0 fault

* When the signal line always present at logical '1' is referred as Stuck-at-1 fault

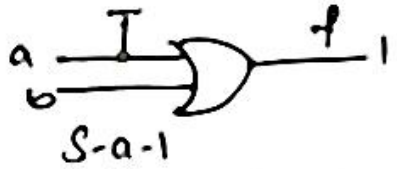


FIG: Stuck-at-1 fault.

d. STUCK-OPEN FAULT MODELS:

* MOS transistor is considered an ideal switch and two types of faults are modelled.

* Stuck open: A single transistor is permanently stuck in the open state.

* Stuck short: A single transistor is permanently shorted irrespective of its gate voltage.

* Detection of a Stuck-open fault requires two vectors.

Scanned By Scanner Go

STUCK-SHORT FAULT MODELS:

* In Stuck short fault models, a single transistor is permanently shorted irrespective of its gate voltage.
* Detection of a stuck-short fault requires the measurement of quiescent current (IDDQ).

BRIDGING FAULTS:

* A bridging fault occurs when two leads in a logic network are connected accidentally and "wired logic" is performed at the connection.
* Depending on whether positive & negative logic is being used the faults have the effect, respectively of ANDing & ORing the signals involved.

POSITIVE LOGIC

Equivalent faulty circuit

NEGATIVE LOGIC

Equivalent faulty circuit

Scanned By Scanner Go

Fault Simulation

* Fault simulation is the methodology used for testing a design after introducing a fault intentionally.
* There is a program that applies a set of test vectors to the primary inputs (PIs) that program is known as test program.
* The time required to apply a test vector & pattern, to check the response of the primary outputs (PO's), and verify the o/p response against the expected o/p response, is known as test-cycle time

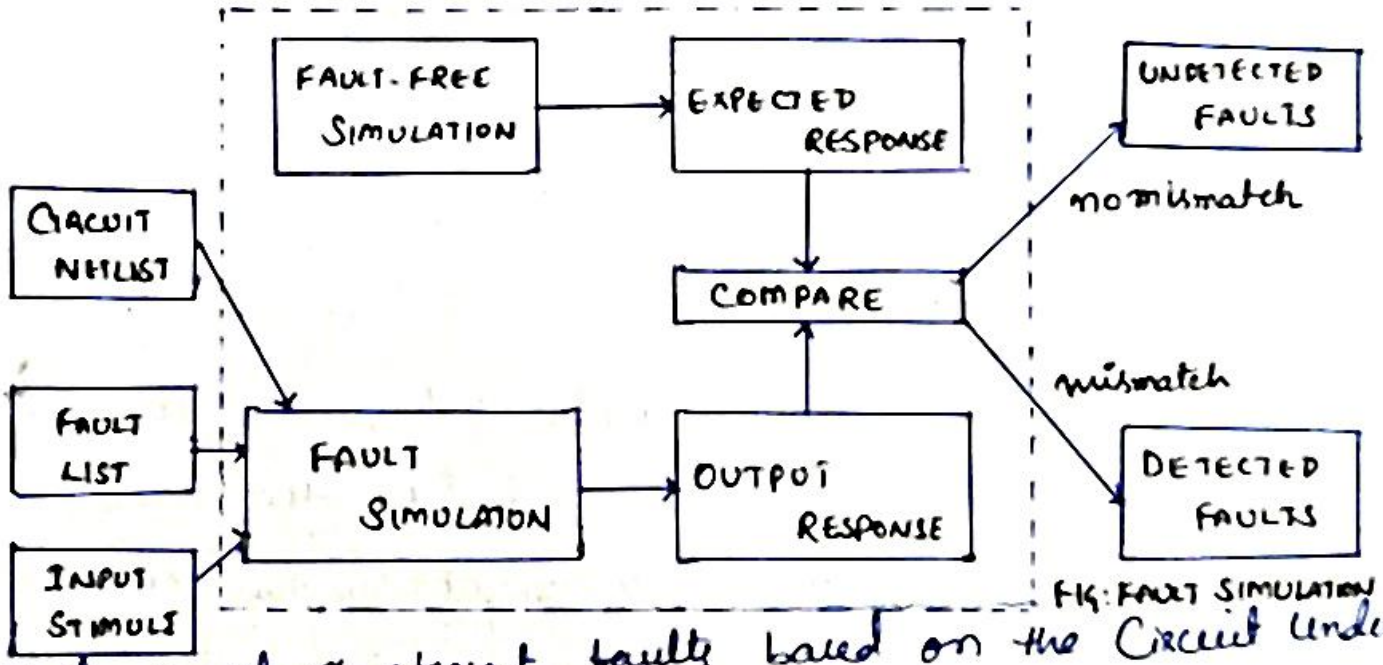


Fig: FAULT SIMULATION

* First, a set of target faults based on the Circuit Under test (CUT) is enumerated
* The i/p stimuli are applied to the CUT and the o/p responses are compared with the expected fault-free response to determine whether the circuit is faulty.
* For fault simulation, the CUT is typically synthesized down to a gate level design

Scanned By Scanner Go

* Fault simulation also provides valuable information on portions of the design that need further design verification.
* Because design verification vectors are often used as functional vectors during manufacturing test.
* Fault simulation can measure the quality of the test program as it can verify if the test program can catch all the known faults that are introduced.
* It measures the fault coverage, which is defined as the ratio of the detected faults to the detable fault.
* Fault simulation can be classified as,
  1) Deterministic fault simulation.
  2) Non-deterministic fault simulation.

1. Deterministic fault simulation:

* In deterministic fault simulation technique, a set of test vectors are used to simulate a circuit and catch the faults.
* But if all the faults are not caught by the test vectors, they are modified and the fault simulation is repeated.
* There are mainly three types of deterministic fault simulation are there.

1. Serial Fault Simulation: In this process, two copies of circuit are tested. The first copy is a good circuit and the second is generated from the first by inserting faults into it to make a faulty circuit. In this process, each faulty circuit is simulated at a time. As it is done one after another, the process is immensely slow

Scanned By Scanner Go

Parallel fault simulation : In the parallel fault simulation, the faulty circuit are simulated simultaneously. This process is very fast as compared to serial fault simulation.

Concurrent Fault simulation : In this Concurrent fault simulation, the whole circuit is not simulated, but only a part is simulated where the fault is introduced. In this process, many faults are simulated at the same time allowing a very fast simulation

2. NON-DETERMINISTIC FAULT SIMULATION:

* In the Non-deterministic fault simulation, instead of testing every fault, a subset or sample of the fault is tested and extrapolate the fault coverage from the sample tested.

TEST GENERATION :-

* The goal of test generation is to find an efficient set of test vectors that detects all faults considered for that circuit.
* To test a circuit with n i/p's and m o/p's, a set of test i/p patterns is applied to the circuit Under Test (CUT) and its responses are compared to the good response of fault-free circuit
* Each i/p pattern is called a test vector. In order to completely test a circuit, many test patterns are required.

Exhaustive Testing: It is difficult to know how many test vectors are needed to guarantee a satisfactory reject rate. If the CUT is an n-i/p combinational logic circuit, we can apply all 2^n possible i/p patterns for testing Stuck-at-faults; this approach is called Exhaustive testing

Scanned By Scanner Go

Functional Testing: In this testing, every entry in the truth table for the Combinational logic Circuit is tested to determine whether it produces the Correct response. In practice, functional testing is Considered by many designers and test engineers to be testing the CUT as thoroughly as possible in a system like mode of operation.

STRUCTURAL TESTING: The approach of structural testing is to select specific test patterns based on Circuit structural information and a set of fault models. Structural testing saves time and improves test efficiency, as the total number of test patterns is decreased because the test vectors target specific faults that would result from defects in the manufactured circuit.

$$\text{Fault Coverage} = \frac{\text{Number of detected faults}}{\text{Total number of faults}}$$

DESIGN STRATEGIES FOR TESTING:

CONTROLLABILITY: The ability to apply the i/p test vectors to the primary inputs of a circuit to set up appropriate logic value (logic 0 or logic 1) is known as Controllability.

* for example, in the presence of fault, the primary I/p has to set logic 1 for stuck-at-0 fault is known as 1-Controllability

* Controlability is important while accessing the degree of difficulty of testing a particular SIL in a circuit.

OBSERVABILITY:- The ability to observe the response of a fault on internal node through primary o/p's of a circuit is known as Observability If the logic gate of this node can reliably observed, this node is regarded as observable

Scanned By Scanner Go

* Whether a circuit node is stuck 1 & 0 is only testable if that node is both controllable and observable.
* These two major factors Controllability and observability, play a vital role to test the circuit for Stuck-at-fault model.
* Controllability and observability can be achieved by the functionality of the combinational circuit and selection of appropriate test vectors.

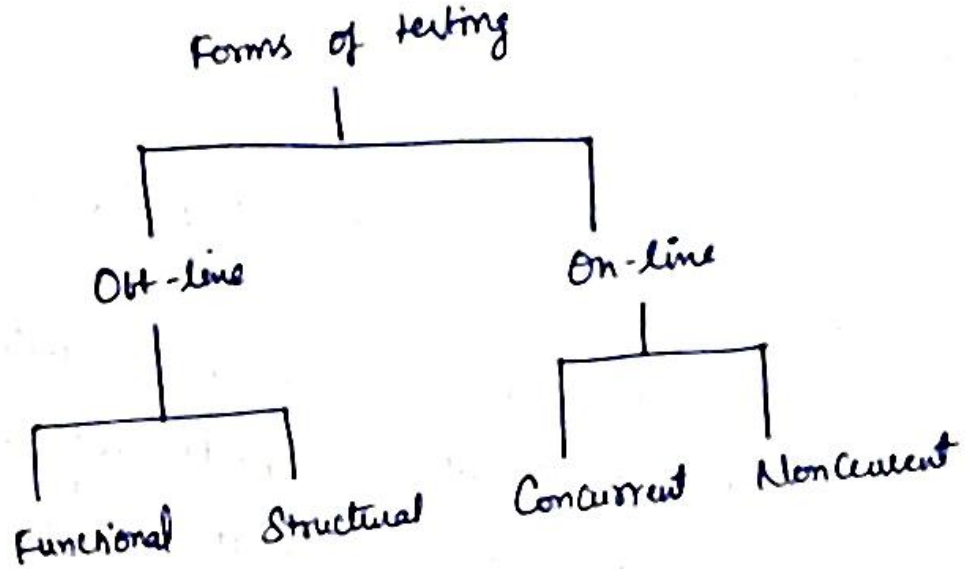
DESIGN FOR TESTABILITY (DFT):

* During the chip development process, the designers must not only guarantee the chip's functionality, but they also must ensure its testability for volume production.
* The extra effort that designers incorporate into the development process for this purpose is called Design for Testability, & DFT
* Design for testability is an essential part of any production chip since the IC manufacturing process is inherently defective.
* If a bad part, & malfunctioning chip is delivered to a customer uses this part to build a system and eventually sell this system to end users, the resultant damage could be significant.
* DFT techniques generally fall into one of the following three categories.
  1. Adhoc DFT techniques
  2. Level-Sensitive Scan design (LSSD) & Scan design
  3. Built-in self test (BIST).

Scanned By Scanner Go

BUILT-IN-SELF-TEST (BIST) :-

* Built-in self test BIST, is the technique of designing additional hardware and software features into integrated circuits to allow them to perform self-testing i.e., testing of their own operation (functionality, parametrically or both) using their own circuitry thereby reducing dependence on an external automated test equipment (ATE)



OFF-LINE BIST :

* Off line BIST deals with testing a system when it is not carrying out its normal functions. Systems, boards and chips can be tested in this mode.

* Off-line BIST testing is carried out using on-chip or on-board test-pattern generators (TPG's) and o/p response analyzers (ORA's).

* Off-line testing does not detect errors in real time.

Functional off line BIST deals with execution of a test based on a functional description of the CUT and often employs a functional, or high level, fault model.

Scanned By Scanner Go

Structural offline BIST: deals with the evaluation of a test based on the structure of the CUT.

* Off-line BIST architectures at the chip and board level can be classified according to the following criteria:

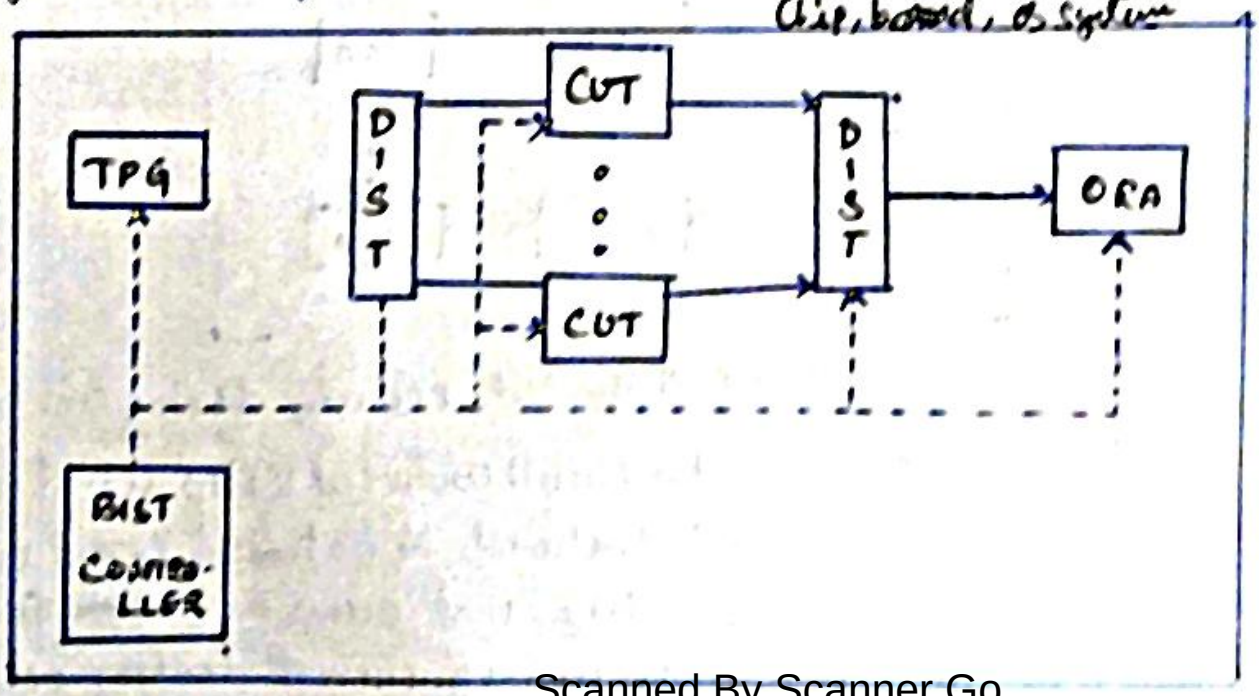
1. Centralized Vs. distributed BIST circuitry.
2. Embedded Vs. Separate BIST elements

* BIST architectures consist of several key elements, namely

1. Test-pattern generator
2. Output-response analyzer.
3. The circuit under test.
4. A distribution system (DIST) for transmitting data from TPG's to CUTs and from CUTs to ORA's.
5. BIST Controller for controlling the BIST circuitry and CUT during self-test.

CENTRALIZED BIST ARCHITECTURE:

* The general form of a centralized BIST architecture is as shown.



Scanned By Scanner Go

* Here, several CUT's share TPG and ORA circuitry. This leads to reduced overhead but increased test time.

* During testing, the BIST Controller may carry out one or more of the following functions.

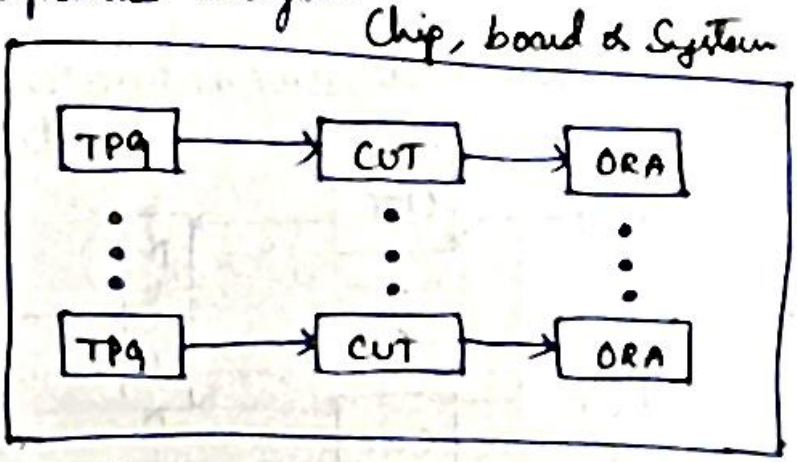
1. Single-step the CUTs through some test sequence.
2. Inhibit s/m clocks and Control test clocks.
3. Communicate with other test Controllers, possibly using test buses.

DISTRIBUTED BUS ARCHITECTURE:

* The distributed BIST architecture is as shown in the figure. Here each CUT is associated with its own TPG and ORA circuitry. This leads to more overhead but less test time and usually more accurate diagnosis.

* In BIST architecture two types of test vectors are generated. These are:

1. Linear Feedback shift register (LFSR)
2. Signature Analyser.



ONLINE BIST: In online BIST, test occurs during normal functional operating conditions i.e, the circuit under test (CUT) is not placed into a test mode where normal functional operation is locked out. Concurrent on-line BIST is a form of testing that occurs simultaneously with normal functional operation. In non-concurrent testing occurs while s/m is in Idle state.

Scanned By Scanner Go