

UNIT-I

IC FABRICATION

Introduction to VLSI Technology:

The invention of the transistor by William B. Shockley, Walter H. Brattain and John Bardeen of Bell Telephone Laboratories drastically changed the electronics industry and paved the way for the development of the Integrated Circuit (IC) technology. The first IC was designed by Jack Kilby at Texas Instruments at the beginning of 1960 and since that time there have already been four generations of ICs .Viz SSI (small scale integration), MSI (medium scale integration), LSI (large scale integration), and VLSI (very large scale integration). Now we are ready to see the emergence of the fifth generation, ULSI (ultra large scale integration) which is characterized by complexities in excess of 3 million devices on a single IC chip. Further miniaturization is still to come and more revolutionary advances in the application of this technology must inevitably occur.

Over the past several years, Silicon CMOS technology has become the dominant fabrication process for relatively high performance and cost effective VLSI circuits. The revolutionary nature of this development is understood by the rapid growth in which the number of transistors integrated in circuits on a single chip.

METAL-OXIDE-SEMICONDUCTOR (MOS) AND RELATED VLSI TECHNOLOGY:

The MOS technology is considered as one of the very important and promising technologies in the VLSI design process. The circuit designs are realized based on PMOS, NMOS, CMOS and BiCMOS devices.

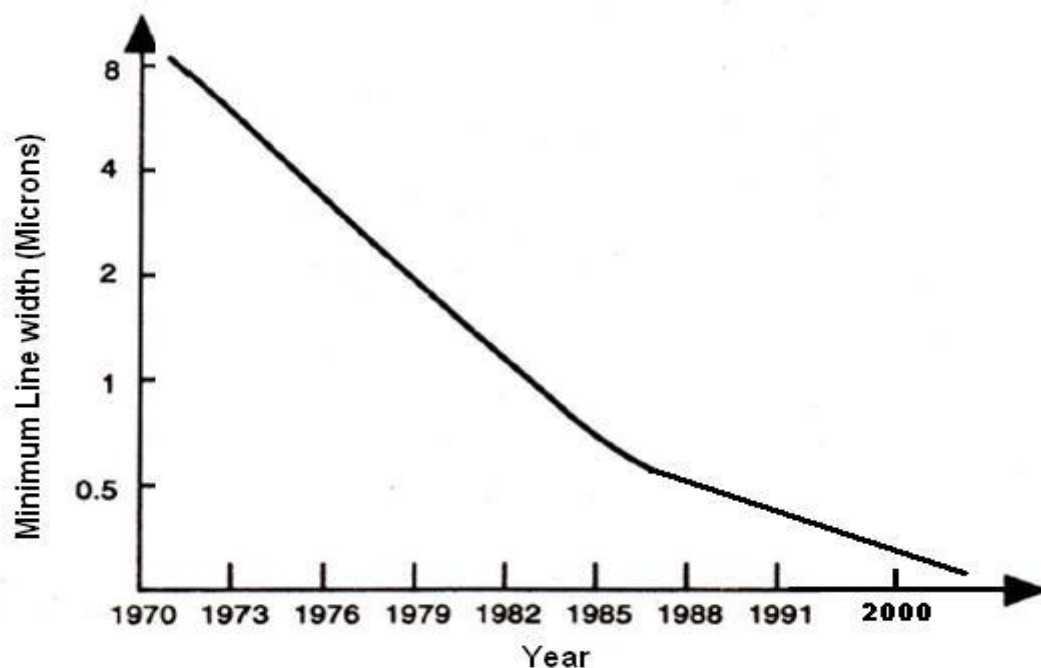
The PMOS devices are based on the p-channel MOS transistors. Specifically, the PMOS channel is part of a n-type substrate lying between two heavily doped p+ wells beneath the source and drain electrodes. Generally speaking, a PMOS transistor is only constructed in consort with an NMOS transistor.

The NMOS technology and design processes provide an excellent background for other technologies. In particular, some familiarity with NMOS allows a relatively easy transition to CMOS technology and design.

The techniques employed in NMOS technology for logic design are similar to GaAs technology.. Therefore, understanding the basics of NMOS design will help in the layout of GaAs circuits

In addition to VLSI technology, the VLSI design processes also provides a new degree of freedom for designers which helps for the significant developments. With the rapid advances in technology the size of the ICs is shrinking and the integration density is increasing.

The minimum line width of commercial products over the years is shown in the graph below.



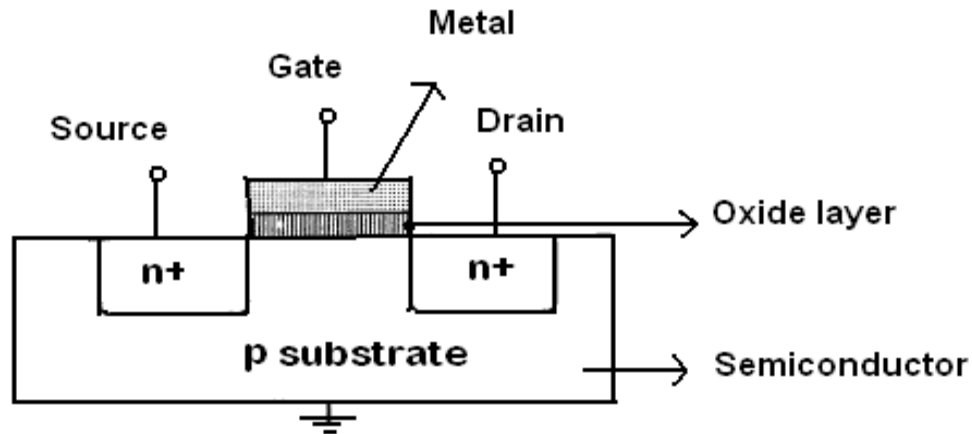
The graph shows a significant decrease in the size of the chip in recent years which implicitly indicates the advancements in the VLSI technology.

BASIC MOS TRANSISTORS:

The MOS Transistor means, Metal-Oxide-Semiconductor Field Effect Transistor which is the most basic element in the design of a large scale integrated circuits(IC).

These transistors are formed as a "sandwich" consisting of a semiconductor layer, usually a slice, or wafer, from a single crystal of silicon; a layer of silicon dioxide (the oxide) and a layer of metal. These layers are patterned in a manner which permits transistors to be formed in

the semiconductor material (the ``substrate''); a diagram showing a MOSFET is shown below in Figure .



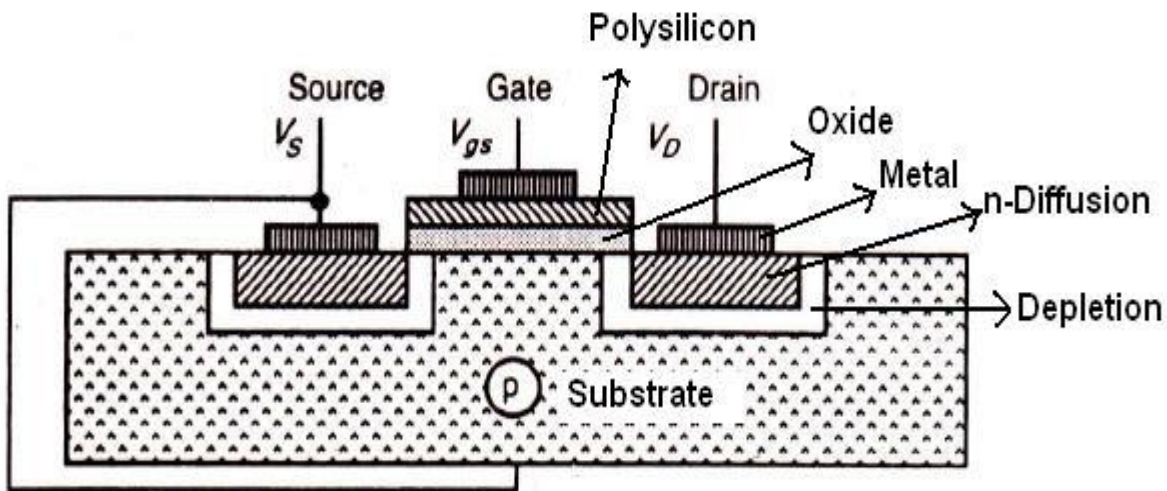
Silicon dioxide is a very good insulator, so a very thin layer, typically only a few hundred molecules thick, is used. In fact , the transistors which are used do not use metal for their gate regions, but instead use polycrystalline silicon (poly). Polysilicon gate FET's have replaced virtually all of the older devices using metal gates in large scale integrated circuits. (Both metal and polysilicon FET's are sometimes referred to as IGFET's (insulated gate field effect transistors), since the silicon dioxide under the gate is an insulator.

MOS Transistors are classified as n-MOS, p-MOS and c-MOS Transistors based on the fabrication:

NMOS devices are formed in a p-type substrate of moderate doping level. The source and drain regions are formed by diffusing n- type impurities through suitable masks into these areas to give the desired n-impurity concentration and give rise to depletion regions which extend mainly in the more lightly doped p-region . Thus, source and drain are isolated from one another by two diodes. Connections to the source and drain are made by a deposited metal layer. In order to make a useful device, there must be the capability for establishing and controlling a current between source and drain, and .this is commonly achieved in one of two ways, giving rise to the enhancement mode and depletion mode transistors.

Enhancement Mode Transistors:

In an enhancement mode device a polysilicon gate is deposited on a layer of insulation over the region between source and drain. In the diagram below channel is not established and the device is in a non-conducting condition, i.e., $V_D = V_S = V_{GS} = 0$. If this gate is connected to a suitable positive voltage with respect to the source, then the electric field established between the gate and the substrate gives rise to a charge inversion region in the substrate under the gate insulation and a conducting path or channel is formed between source and drain.

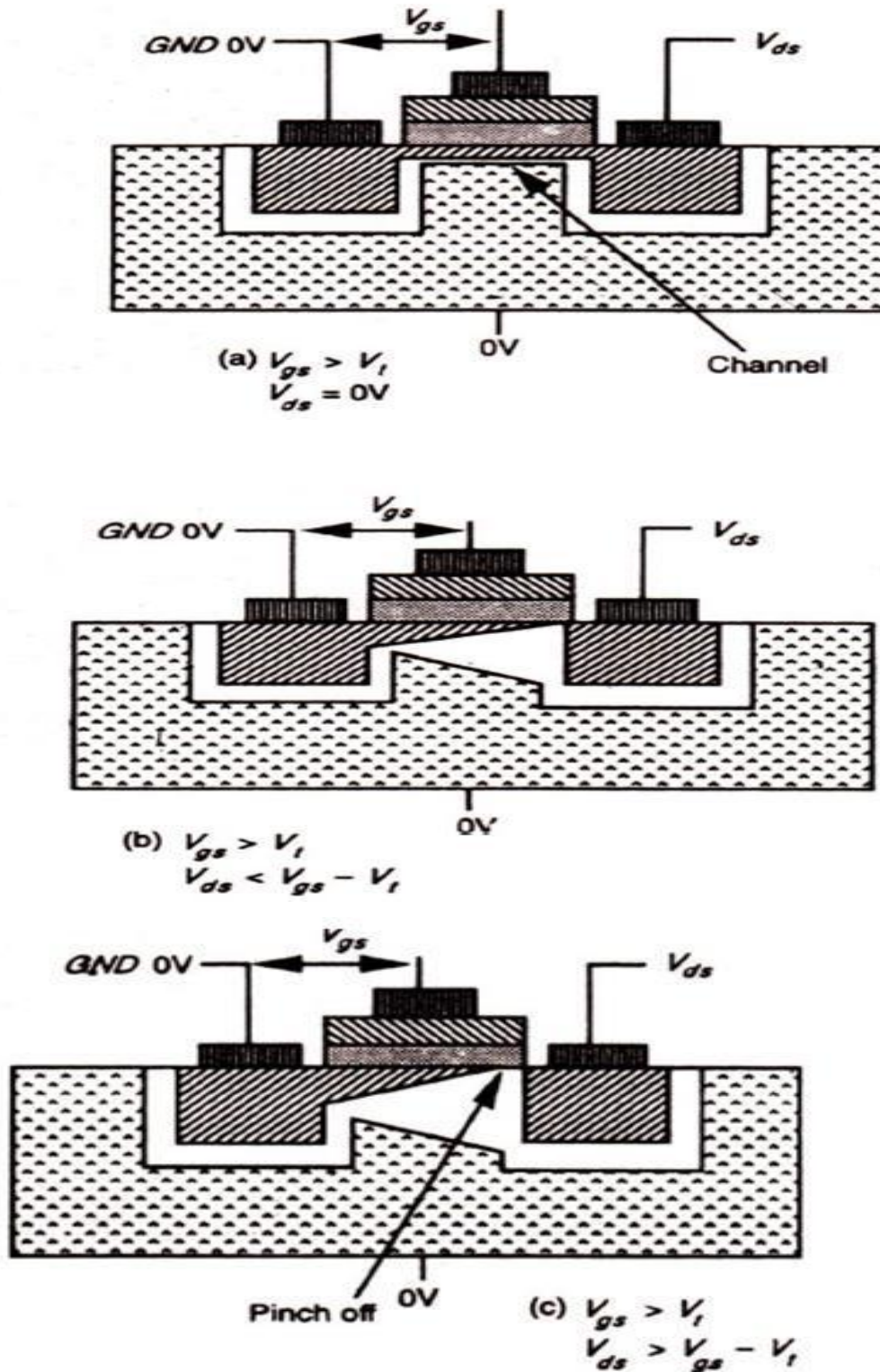


ENHANCEMENT MODE TRANSISTOR ACTION :

To understand the enhancement mechanism, let us consider the enhancement mode device. In order to establish the channel, a minimum voltage level called threshold voltage (V_t) must be established between gate and source. Fig. (a) Shows the existing situation where a channel is established but no current flowing between source and drain ($V_{ds} = 0$).

Let us now consider the conditions when current flows in the channel by applying a voltage V_{ds} between drain and source. The IR drop = V_{ds} along the channel. This develops a voltage between gate and channel varying with distance along the channel with the voltage being a maximum of V_{gs} at the source end. Since the effective gate voltage is $V_g = V_{gs} - V_t$, (no current flows when $V_{gs} < V_t$) there will be voltage available to invert the channel at the drain end so long as $V_{gs} - V_t \sim V_{ds}$. The limiting condition comes when $V_{ds} = V_{gs} - V_t$. For all

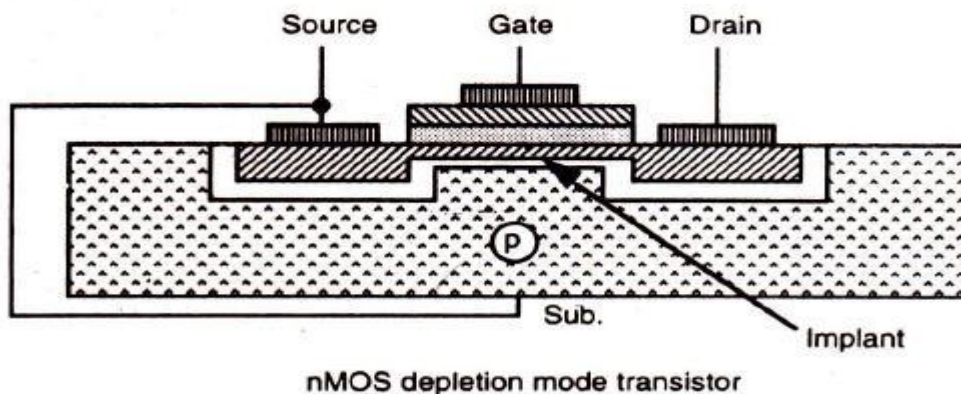
voltages $V_{ds} < V_{gs} - V_t$, the device is in the non-saturated region of operation which is the condition shown in Fig. (b) below.



Let us now consider the situation when V_{ds} is increased to a level greater than $V_{gs} - V_t$. In this case, an IR drop equal to $V_{gs} - V_t$ occurs over less than the whole length of the channel such that, near the drain, there is insufficient electric field available to give rise to an inversion layer to create the channel. The channel is, therefore, 'pinched off' as shown in Fig. (c). Diffusion current completes the path from source to drain in this case, causing the channel to exhibit a high resistance and behave as a constant current source. This region, known as saturation, is characterized by almost constant current for increase of V_{ds} above $V_{ds} = V_{gs} - V_t$. In all cases, the channel will cease to exist and no current will flow when $V_{gs} < V_t$. Typically, for enhancement mode devices, $V_t = 1$ volt for $V_{DD} = 5$ V or, in general terms, $V_t = 0.2 V_{DD}$.

DEPLETION MODE TRANSISTOR ACTION:

N-MOS Depletion mode MOSFET's are built with P-type silicon substrates, and P-channel versions are built on N-type substrates. In both cases they include a thin gate oxide formed between the source and drain regions. A conductive channel is deliberately formed below the gate oxide layer and between the source and drain by using ion-implantation. By implanting the correct ion polarity in the channel region during fabrication determines the polarity of the threshold voltage (i.e. $-V_t$ for an N channel transistor, or $+V_t$ for an P-channel transistor). The actual concentration of ions in the substrate-to-channel region is used to adjust the threshold voltage (V_t) to the desired value. Depletion-mode devices are a little more difficult to manufacture and their characteristics harder to control than enhancement types, which do not require ion implantation. In depletion mode devices the channel is established, due to the implant, even when $V_{gs} = 0$, and to cause the channel to cease a negative voltage V_{td} must be applied between gate and source.



V_{td} is typically $< -0.8 V_{DD}$, depending on the implant and substrate bias, but, threshold voltage differences apart, the action is similar to that of the enhancement mode transistor.

NMOS FABRICATION

The following description explains the basic steps used in the process of fabrication.

- (a) The fabrication process starts with the oxidation of the silicon substrate. It is shown in the Figure 1.9 (a).
- (b) A relatively thick silicon dioxide layer, also called field oxide, is created on the surface of the substrate. This is shown in the Figure 1.9 (b).
- (c) Then, the field oxide is selectively etched to expose the silicon surface on which the MOS transistor will be created. This is indicated in the Figure 1.9 (c).
- (d) This is followed by covering the surface of substrate with a thin, high-quality oxide layer, which will eventually form the gate oxide of the MOS transistor as illustrated in Figure 1.9 (d).
- (e) On top of the thin oxide, a layer of polysilicon (polycrystalline silicon) is deposited as is shown in the Figure 1.9 (e). Polysilicon is used both as gate electrode material for MOS transistors and also as an interconnect medium in silicon integrated circuits. Un-doped polysilicon has relatively high resistivity. The resistivity of polysilicon can be reduced, however, by doping it with impurity atoms.
- (f) After deposition, the polysilicon layer is patterned and etched to form the interconnects and the MOS transistor gates. This is shown in Figure 1.9 (f).
- (g) The thin gate oxide not covered by polysilicon is also etched along, which exposes the bare silicon surface on which the source and drain junctions are to be formed (Figure 1.9 (g)).
- (h) The entire silicon surface is then doped with high concentration of impurities, either through diffusion or ion implantation (in this case with donor atoms to produce n-type doping). Diffusion is achieved by heating the wafer to a high temperature and passing the gas containing desired impurities over the surface. Figure 1.9 (h) shows that the doping penetrates the exposed areas on the silicon surface, ultimately creating two n-type regions (source and drain junctions) in the p-type substrate. The impurity doping also penetrates the polysilicon on the surface, reducing its resistivity.

- (i) Once the source and drain regions are completed, the entire surface is again covered with an insulating layer of silicon dioxide, as shown in Figure 1.9 (i).(j) The insulating oxide layer is then patterned in order to provide contact windows for the drain and source junctions, as illustrated in Figure 1.9 (j).

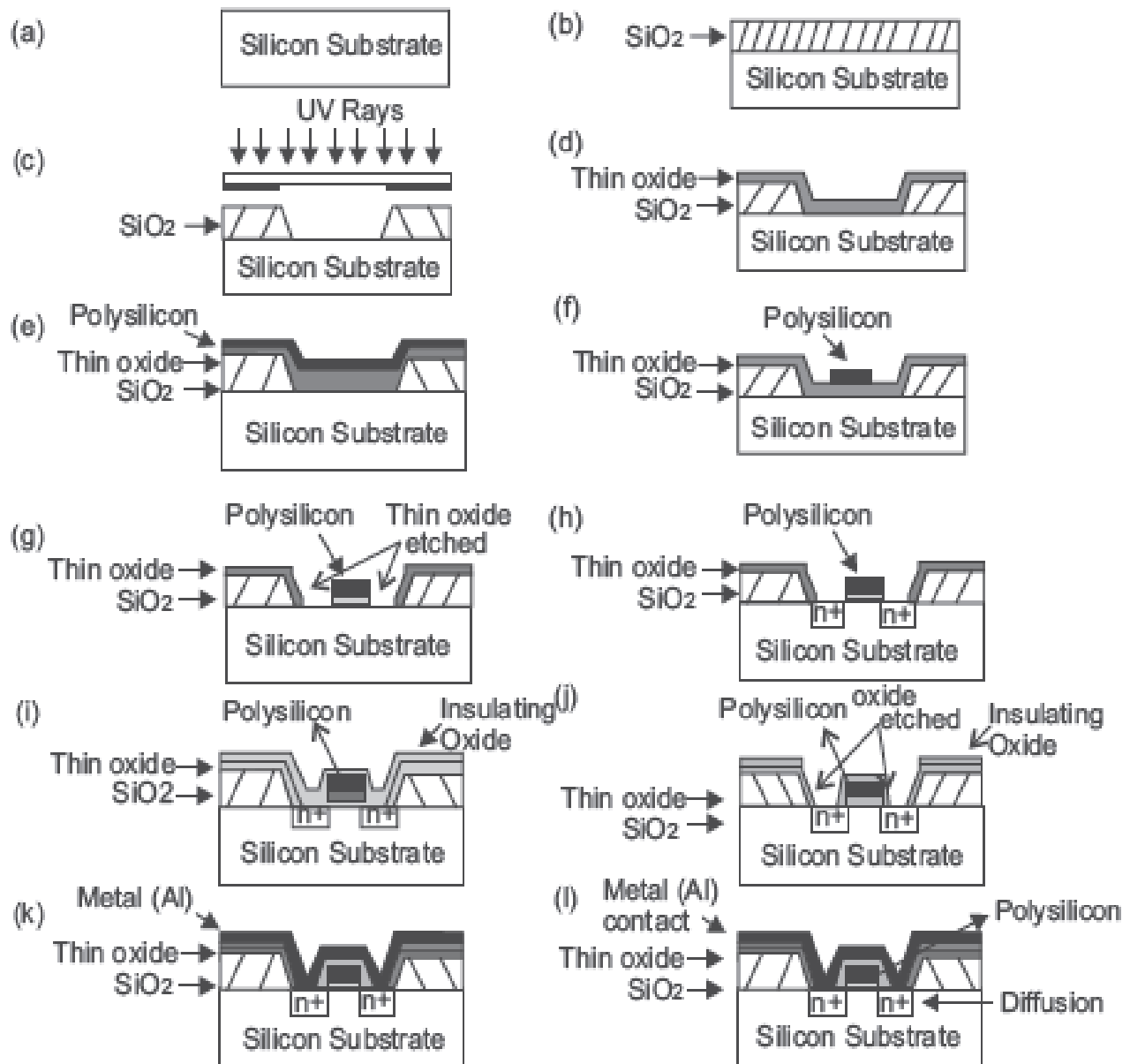


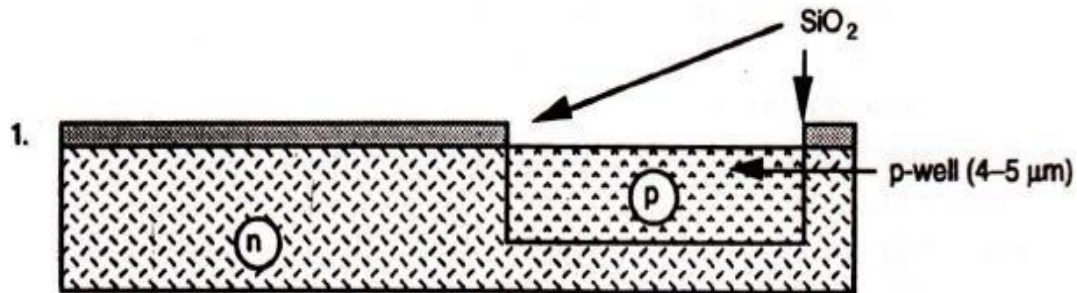
Figure 1.9: Fabrication Process of NMOS Device

CMOS FABRICATION :

CMOS fabrication is performed based on various methods, including the p-well, the n-well, the twin-tub, and the silicon-on-insulator processes. Among these methods the p-well process is widely used in practice and the n-well process is also popular, particularly as it is an easy retrofit to existing NMOS lines.

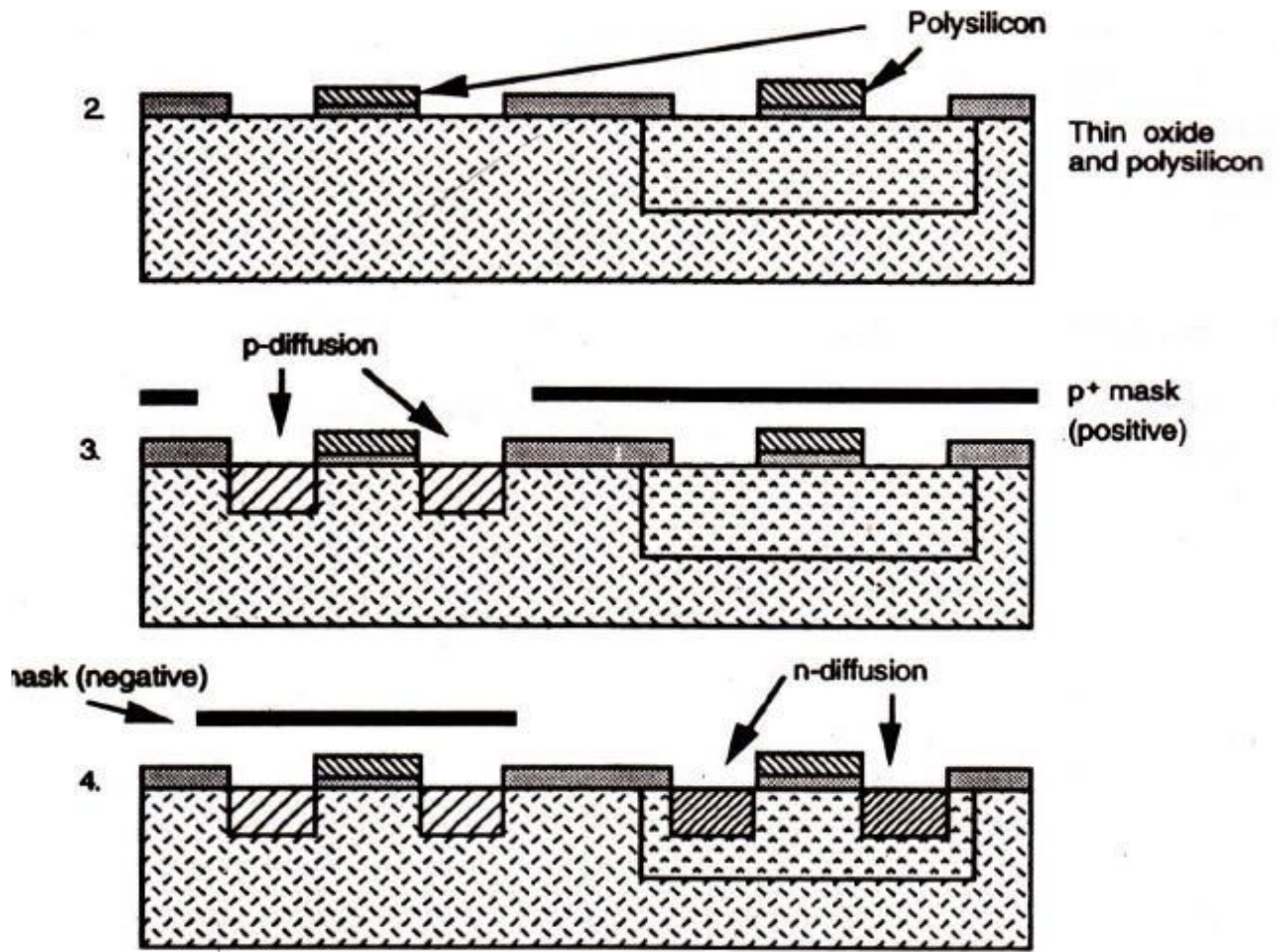
(i) The p-well Process:

The p-well structure consists of an n-type substrate in which p-devices may be formed by suitable masking and diffusion and, in order to accommodate n-type devices, a deep p-well is diffused into the n-type substrate as shown in the Fig. below.



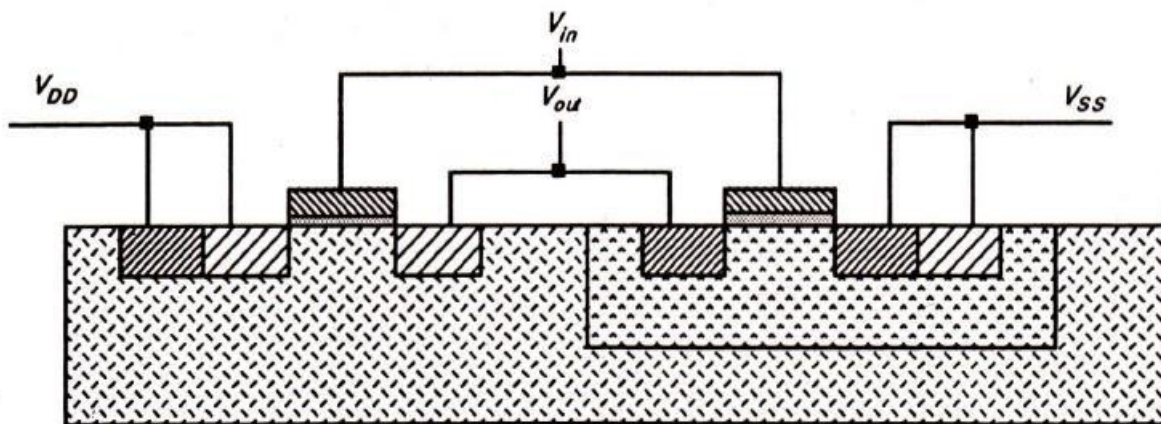
This diffusion should be carried out with special care since the p-well doping concentration and depth will affect the threshold voltages as well as the breakdown voltages of the n-transistors. To achieve low threshold voltages (0.6 to 1.0 V) either deep-well diffusion or high-well resistivity is required. However, deep wells require larger spacing between the n- and p-type transistors and wires due to lateral diffusion and therefore a larger chip area. The p-wells act as substrates for the n-devices within the parent n-substrate, and, the two areas are electrically isolated.

Except this in all other respects- like masking, patterning, and diffusion-the process is similar to NMOS fabrication.



P-well fabrication process(Figs 1,2,3 & 4)

The diagram below shows the CMOS p-well inverter showing V_{DD} and V_{SS} substrate connections



The n-well Process :

Though the p-well process is widely used in C-MOS fabrication the n-well fabrication is also very popular because of the lower substrate bias effects on transistor threshold voltage and also lower parasitic capacitances associated with source and drain regions.

The typical n-well fabrication steps are shown in the diagram below.

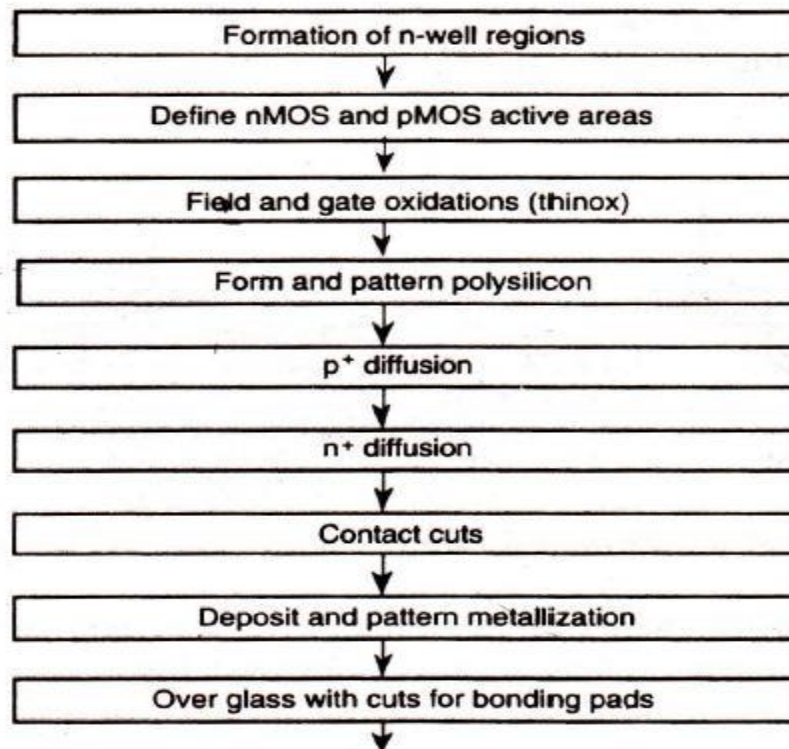
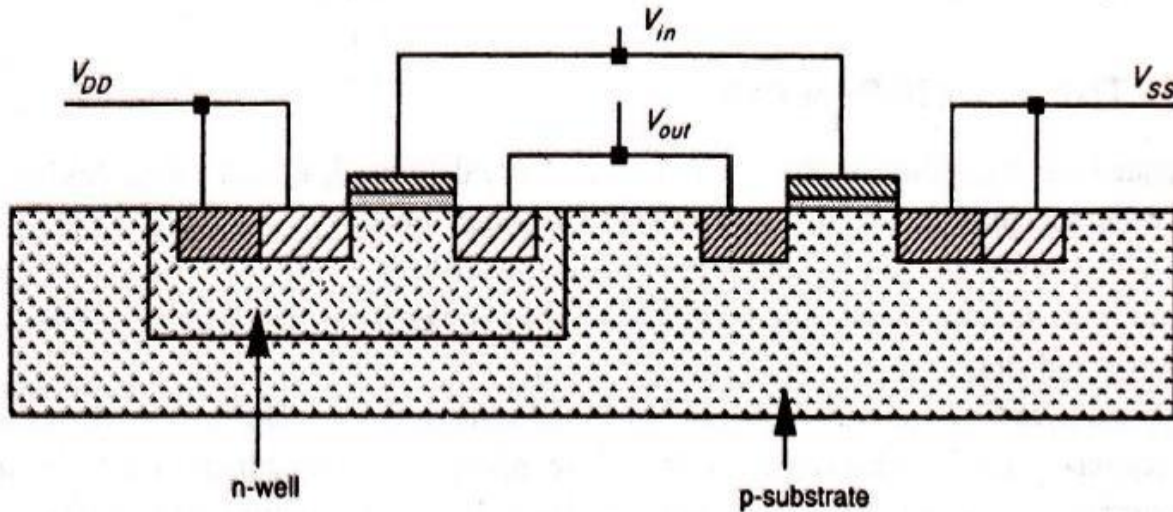


Fig. n-well fabrication steps

The first mask defines the n-well regions. This is followed by a low dose phosphorus implant driven in by a high temperature diffusion step to form the n-wells. The well depth is optimized to ensure against-substrate top+ diffusion breakdown without compromising then-well to n+ mask separation. The next steps are to define the devices and diffusion paths, grow field oxide, deposit and pattern the poly silicon, carry out the diffusions, make contact cuts, and finally metalize as before. It will be seen that an n+ mask and its complement may be used to define the n- and p-diffusion regions respectively. These same masks also include the V_{DD} and V_{SS} contacts (respectively). It should be noted that, alternatively, we could have used a p+ mask and its complement since the n + and p + masks are generally complementary. The diagram below shows the Cross-sectional view of n-well CMOS Inverter.



Due to the differences in charge carrier mobilities, the n-well process creates non-optimum p-channel characteristics. However, in many CMOS designs (such as domino-logic and dynamic logic structures), this is relatively unimportant since they contain a preponderance of n-channel devices. Thus then-channel transistors are mainly those used to form logic elements, providing speed and high density of elements.

However, a factor of the n-well process is that the performance of the already poorly performing p-transistor is even further degraded. Modern process lines have come to grips with these problems, and good device performance may be achieved for both p-well and n-well fabrication.

BICMOS Technology:

A Bi-CMOS circuit of both bipolar junction transistors and MOS transistors on a single substrate. The driving capability of MOS transistors is less because of limited current sourcing and sinking capabilities of the transistors. To drive large capacitive loads Bi-CMOS technology is used. As this technology combines Bipolar and CMOS transistors in a single integrated circuit, it has the advantages of both bipolar and CMOS transistors. Bi-CMOS is able to achieve VLSI circuits with speed-power-density performance previously not possible with either technology individually. The diagram given below shows the cross section of the Bi-CMOS process which uses an NPN transistor

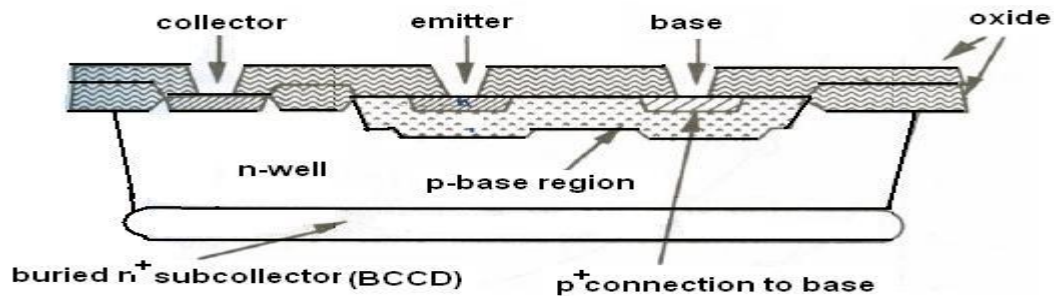
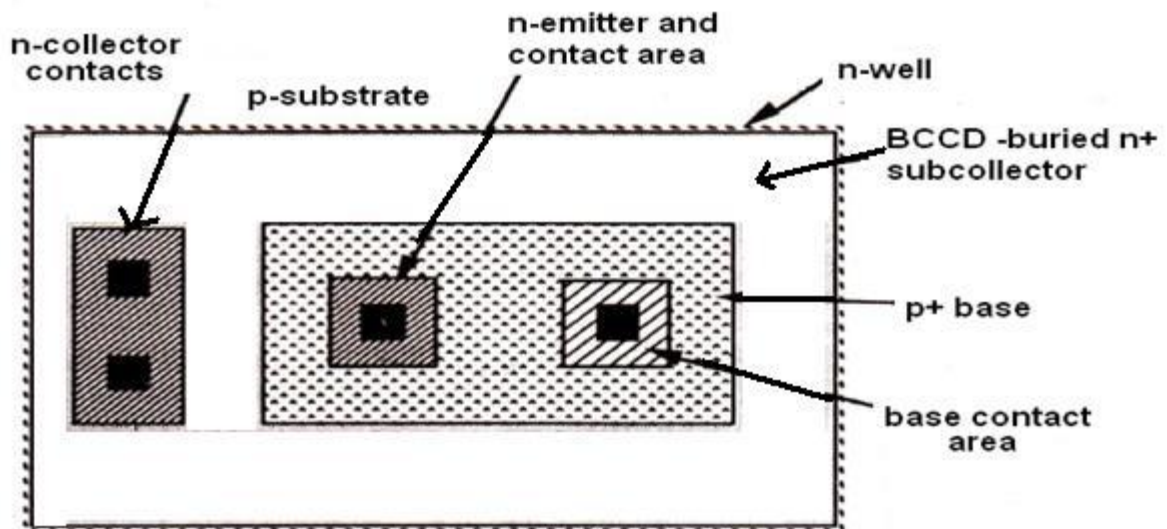


Fig. Cross section of Bi-CMOS process

The lay-out view of Bic-MOS transistor is shown in the figure below. The fabrication of Bi-CMOS is similar to CMOS but with certain additional process steps and additional masks are considered. They are (i) the p⁺ base region; (ii) n⁺ collector area; and (iii) the buried sub collector (BCCD).

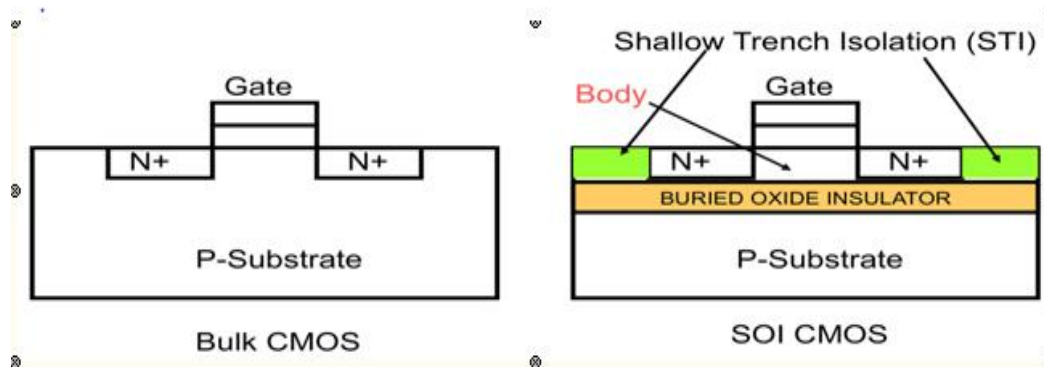


Silicon on Insulator (SOI) Technology:

Scaling of Bulk CMOS technology leads to issues such as

- Sub-threshold leakage power
- Parasitic device capacitances affecting the performance

SOI Technology has been around and used in other devices like IGFET's



Most of the process tooling is the same for Bulk and SOI

SOI devices have 5 terminals – Gate, Drain, Source, Substrate, Body

Types of SOI Devices:

1. Fully Depleted (FD) devices:

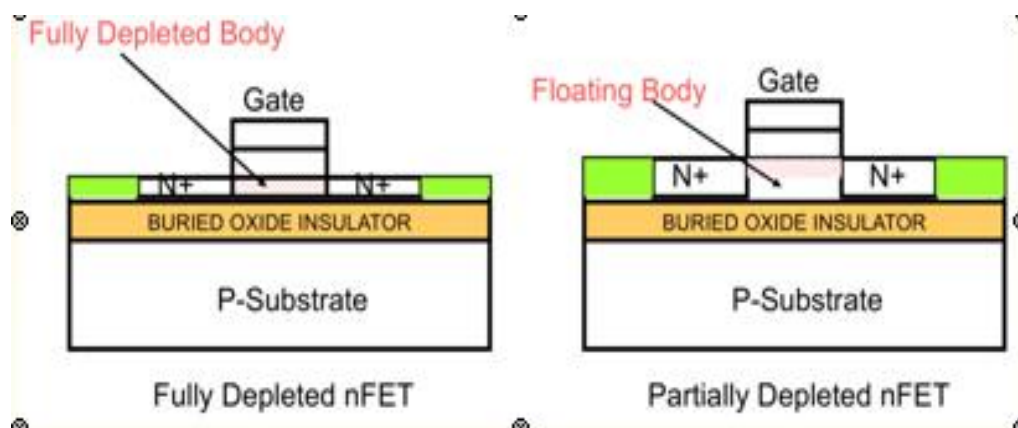
Body is completely depleted under normal bias conditions

The device behaves similar to a bulk device

Requires a very thin film of Silicon so that the body can fully deplete.

2. Partially Depleted (PD) devices:

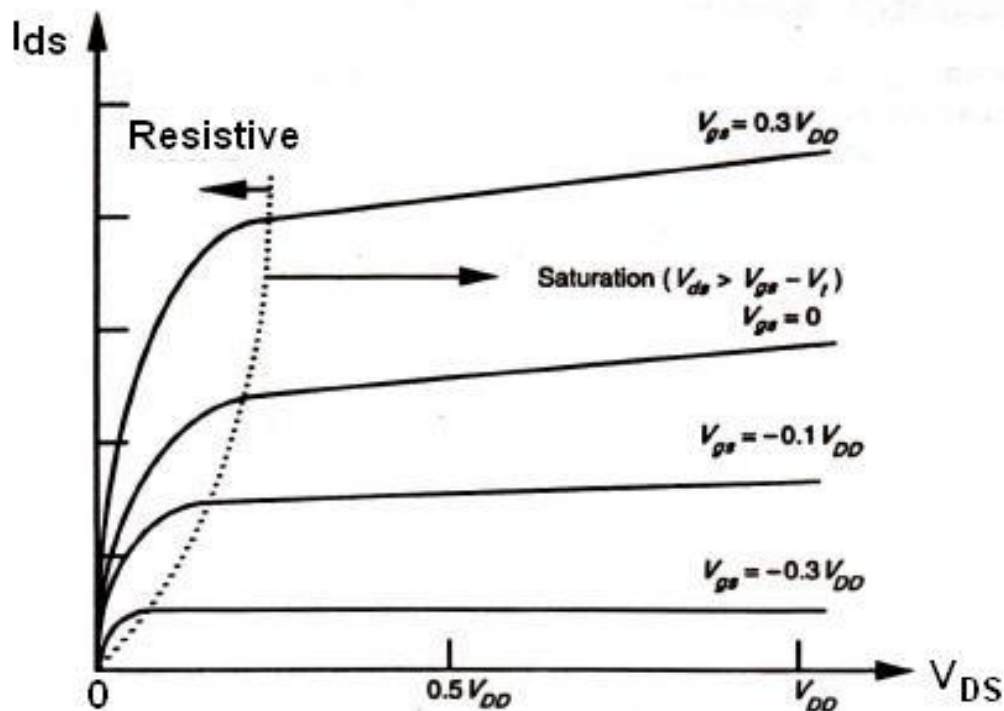
- Body is not completely depleted under normal bias conditions.
- Difficult to manufacture FD devices due to thin Si layer requirement



Basic Electrical Properties of MOS and BiCMOS Circuits:

I_{DS} - V_{DS} characteristics of MOS Transistor:

The graph below shows the I_D Vs V_{DS} characteristics of an n- MOS transistor for several values of V_{GS} . It is clear that there are two conduction states when the device is ON. The saturated state and the non-saturated state. The saturated curve is the flat portion and defines the saturation region. For $V_{gs} < V_{DS} + V_{th}$, the NMOS device is conducting and I_D is independent of V_{DS} . For $V_{gs} > V_{DS} + V_{th}$, the transistor is in the non-saturation region and the curve is a half parabola. When the transistor is OFF ($V_{gs} < V_{th}$), then I_D is zero for any V_{DS} value.

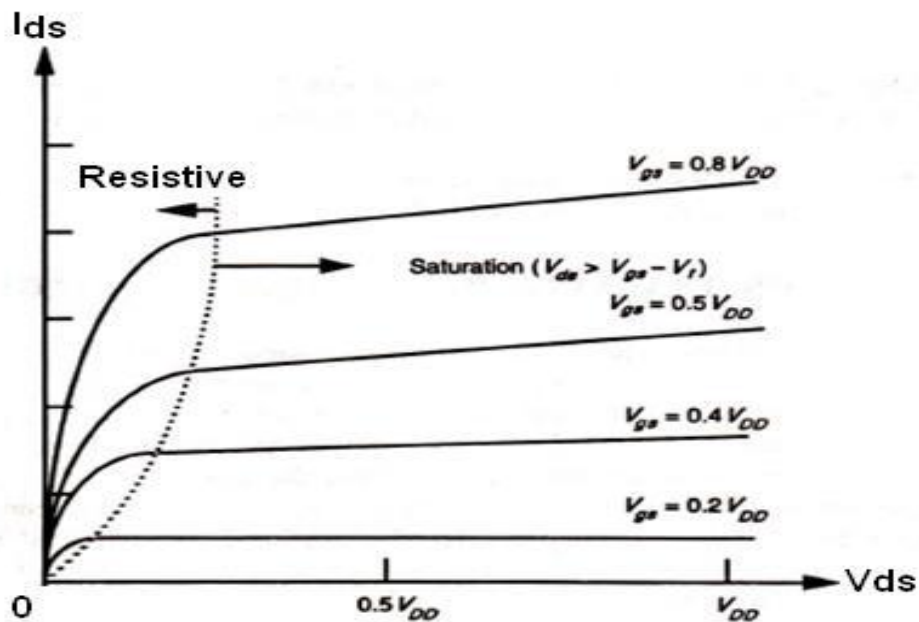


(a) Depletion mode device

The boundary of the saturation/non-saturation bias states is a point seen for each curve in the graph as the intersection of the straight line of the saturated region with the quadratic curve of the non-saturated region. This intersection point occurs at the channel pinch off voltage called V_{DSAT} . The diamond symbol marks the pinch-off voltage V_{DSAT} for each value of V_{GS} . V_{DSAT} is defined as the minimum drain-source voltage that is required to keep the transistor in saturation

for a given V_{GS} . In the non-saturated state, the drain current initially increases almost linearly from the origin before bending in a parabolic response. Thus the name ohmic or linear for the non-saturated region.

The drain current in saturation is virtually independent of V_{DS} and the transistor acts as a current source. This is because there is no carrier inversion at the drain region of the channel. Carriers are pulled into the high electric field of the drain/substrate pn junction and ejected out of the drain terminal.



(b). Enhance mode device

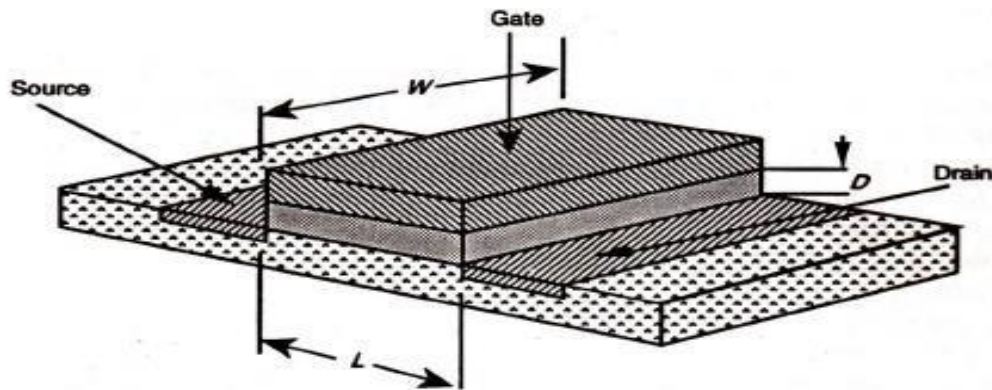
Drain-to-Source Current I_{DS} versus Voltage V_{DS} Relationships:

The working of a MOS transistor is based on the principle that the use of a voltage on the gate induce a charge in the channel between source and drain, which may then be caused to move from source to drain under the influence of an electric field created by voltage V_{ds} applied between drain and source. Since the charge induced is dependent on the gate to source voltage V_{gs} then I_{ds} is dependent on both V_{gs} and V_{ds} .

Let us consider the diagram below in which electrons will flow source to drain .So, the drain current is given by

$$I_{ds} = -I_{sd} = \frac{\text{Charge induced in channel (Qc)}}{\text{Electron transit time}(\tau)}$$

Where the transit time is given by $\tau_{sd} = \frac{\text{Length of the channel}}{\text{Velocity (v)}}$



But velocity $v = \mu E_{ds}$

Where μ = electron or hole mobility and E_{ds} = Electric field

Also , $E_{ds} = V_{ds}/L$

So, $v = \mu \cdot V_{ds}/L$

And $\tau_{ds} = L^2 / \mu \cdot V_{ds}$

The typical values of μ at room temperature are given below.

$$\mu_n \approx 650 \text{ cm}^2/\text{V sec (surface)}$$

$$\mu_p \approx 240 \text{ cm}^2/\text{V sec (surface)}$$

The Non-saturated Region:

Let us consider the I_d vs V_d relationships in the non-saturated region. The charge induced in the channel due to the voltage difference between the gate and the channel, V_{gs} (assuming substrate connected to source). The voltage along the channel varies linearly with distance X from the source due to the IR drop in the channel. In the non-saturated state the average value is $V_{ds}/2$. Also the effective gate voltage $V_g = V_{gs} - V_t$ where V_t is the threshold voltage needed to invert the charge under the gate and establish the channel.

Hence the induced charge is $Q_c = E_g \epsilon_{ins} \epsilon_0 w \cdot L$

Where

E_g = average electric field gate to channel

ϵ_{ins} = relative permittivity of insulation between gate and channel

ϵ_0 = permittivity of free space.

So, we can write that

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

Here D is the thickness of the oxide layer. Thus

$$Q_c = \frac{WL\epsilon_{ins}\epsilon_0}{D} \left((V_{gs} - V_t) - \frac{V_{ds}}{2} \right)$$

So, by combining the above two equations, we get

$$I_{ds} = \frac{\epsilon_{ins}\epsilon_0\mu}{D} \frac{W}{L} \left((V_{gs} - V_t) - \frac{V_{ds}}{2} \right) V_{ds}$$

Or the above equation can be written as

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

In the non-saturated or resistive region where $V_{ds} < V_{gs} - V_t$ and

$$K = \frac{\epsilon_{ins}\epsilon_0\mu}{D}$$

Generally, a constant β is defined as

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

So that, the expression for drain –source current will become

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

The gate /channel capacitance is

$$C_g = \frac{\epsilon_{ins} \epsilon_0 W L}{D} \text{ (parallel plate)}$$

Hence we can write another alternative form for the drain current as

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

Some time it is also convenient to use gate –capacitance per unit area, C_g

So, the drain current is

$$I_{ds} = C_{0\mu} \frac{W}{L} \left((V_{gs} - V_t) V_{ds} - \frac{V_{ds}^2}{2} \right)$$

This is the relation between drain current and drain-source voltage in non-saturated region.

The Saturated Region

Saturation begins when $V_{ds} = V_{gs} - V_t$, since at this point the IR drop in the channel equals the effective gate to channel voltage at the drain and we may assume that the current remains fairly constant as V_{ds} increases further. Thus

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

Or we can also write that

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

Or it can also be written as

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

Or

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

The expressions derived above for I_{ds} hold for both enhancement and depletion mode devices. Here the threshold voltage for the NMOS depletion mode device (denoted as V_{td}) is negative.

MOS Transistor Threshold Voltage (V_t or V_{th}) :

The gate structure of a MOS transistor consists, of charges stored in the dielectric layers and in the surface to surface interfaces as well as in the substrate itself. Switching an enhancement mode MOS transistor from the off to the on state consists in applying sufficient gate voltage to neutralize these charges and enable the underlying silicon to undergo an inversion due to the electric field from the gate. Switching a depletion mode NMOS transistor from the on to the off state consists in applying enough voltage to the gate to add to the stored charge and invert the 'n' implant region to 'p'.

The threshold voltage V_t may be expressed as:

$$V_t = \phi_{ms} + \frac{Q_D - Q_{SS}}{C_0} + 2\phi_{fn}$$

Where Q_D = the charge per unit area in the depletion layer below the oxide

Q_{SS} = charge density at Si: SiO_2 interface

C_0 = Capacitance per unit area.

ϕ_{ms} = work function difference between gate and Si

ϕ_{fn} = Fermi level potential between inverted surface and bulk Si

For polynomial gate and silicon substrate, the value of ϕ_{ms} is negative but negligible and the magnitude and sign of V_t are thus determined by balancing the other terms in the equation.

To evaluate the V_t the other terms are determined as below.

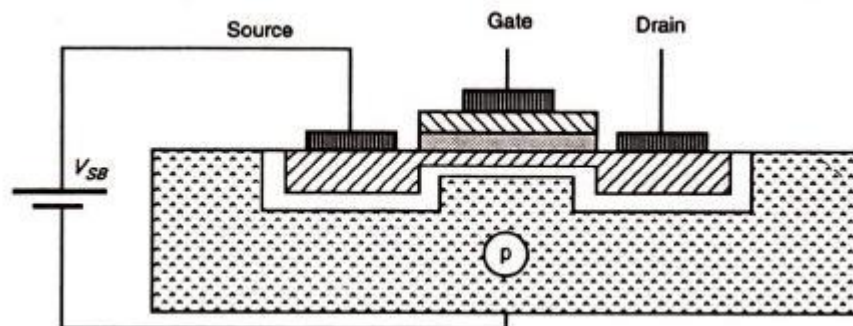
$$Q_B = \sqrt{2\epsilon_0\epsilon_{Si}qN(2\phi_{fN} + V_{SB})} \text{ coulomb/m}^2$$

$$\phi_{fN} = \frac{kT}{q} \ln \frac{N}{n_i} \text{ volts}$$

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

Body Effect :

Generally while studying the MOS transistors it is treated as a 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 and the body (V_{sb}) affects the threshold voltage of the transistor. In many situations, this Body Effect is relatively insignificant, so we can (unless **otherwise** stated) ignore the Body Effect. But it is not always insignificant, in some cases it can have a tremendous impact on MOSFET circuit performance.



Body effect - NMOS device

Increasing V_{sb} causes the channel to be depleted of charge carriers and thus the threshold voltage is raised. Change in V_t is given by $\delta v_t = \gamma \cdot (V_{sb})^{1/2}$ where γ is a constant which depends on substrate doping so that the more lightly doped the substrate, the smaller will be the body effect

The threshold voltage can be written as

$$V_t = V_t(0) + \left(\frac{D}{\epsilon_{ins} \epsilon_0} \right) \sqrt{2 \epsilon_0 \epsilon_{si} q N_s} \cdot (V_{SB})^{1/2}$$

Where $V_t(0)$ is the threshold voltage for $V_{sd} = 0$

For n-MOS depletion mode transistors, the body voltage values at different V_{DD} voltages are given below.

$$V_{SB} = 0 \text{ V ; } V_{sd} = -0.7V_{DD} \text{ (} = -3.5 \text{ V for } V_{DD} = +5\text{V)}$$

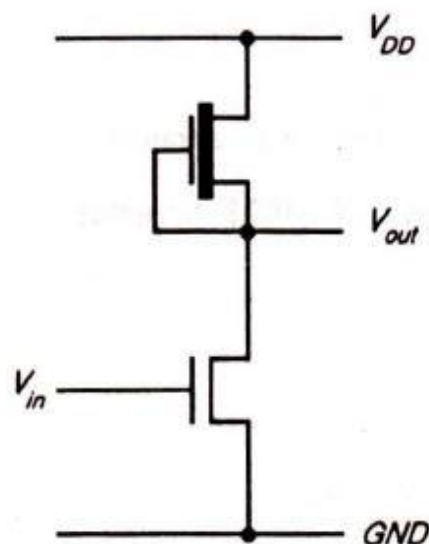
$$V_{SB} = 5 \text{ V ; } V_{sd} = -0.6V_{DD} \text{ (} = -3.0 \text{ V for } V_{DD} = +5\text{V)}$$

The NMOS INVERTER :

An inverter circuit is a very important circuit for producing a complete range of logic circuits. This is needed for restoring logic levels, for NAND and NOR gates, and for sequential and memory circuits of various forms .

A simple inverter circuit can be constructed using a transistor with source connected to ground and a load resistor of R_L connected from the drain to the positive supply rail V_{DD} . The output is taken from the drain and the input applied between gate and ground .

But, during the fabrication resistors are not conveniently produced on the silicon substrate and even small values of resistors occupy excessively large areas .Hence some other form of load resistance is used. A more convenient way to solve this problem is to use a depletion mode transistor as the load, as shown in Fig. Below.

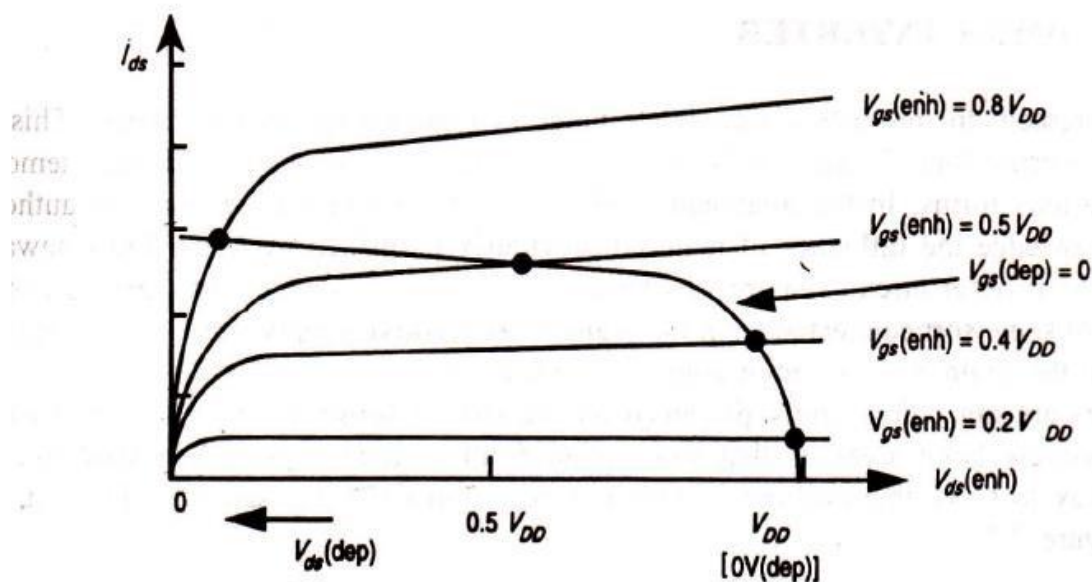


The salient features of the n-MOS inverter are

- For the depletion mode transistor, the gate is connected to the source so it is always on .
- In this configuration the depletion mode device is called the pull-up (P.U) and the enhancement mode device the pull-down (P.D) transistor.
- With no current drawn from the output, the currents I_{ds} for both transistors must be equal.

NMOS Inverter transfer characteristic:

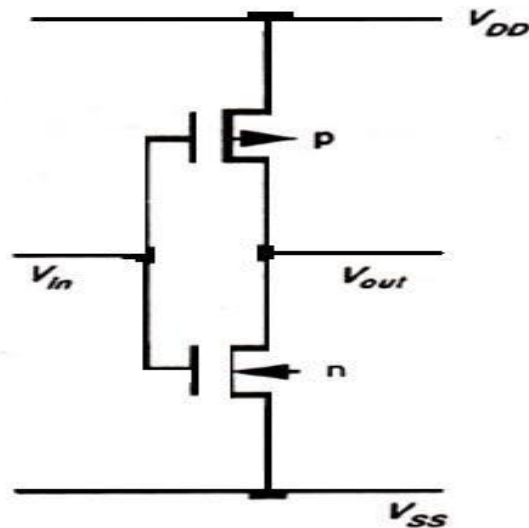
The transfer characteristic is drawn by taking V_{ds} on x-axis and I_{ds} on Y-axis for both enhancement and depletion mode transistors. So, to obtain the inverter transfer characteristic for $V_{gs} = 0$ depletion mode characteristic curve is superimposed on the family of curves for the enhancement mode device and from the graph it can be seen that , maximum voltage across the enhancement mode device corresponds to minimum voltage across the depletion mode transistor.



From the graph it is clear that as $V_{in}(=V_{gs} \text{ p.d. Transistor})$ exceeds the Pulldown threshold voltage current begins to flow. The output voltage V_{out} thus decreases and the subsequent increases in V_{in} will cause the Pull down transistor to come out of saturation and become resistive.

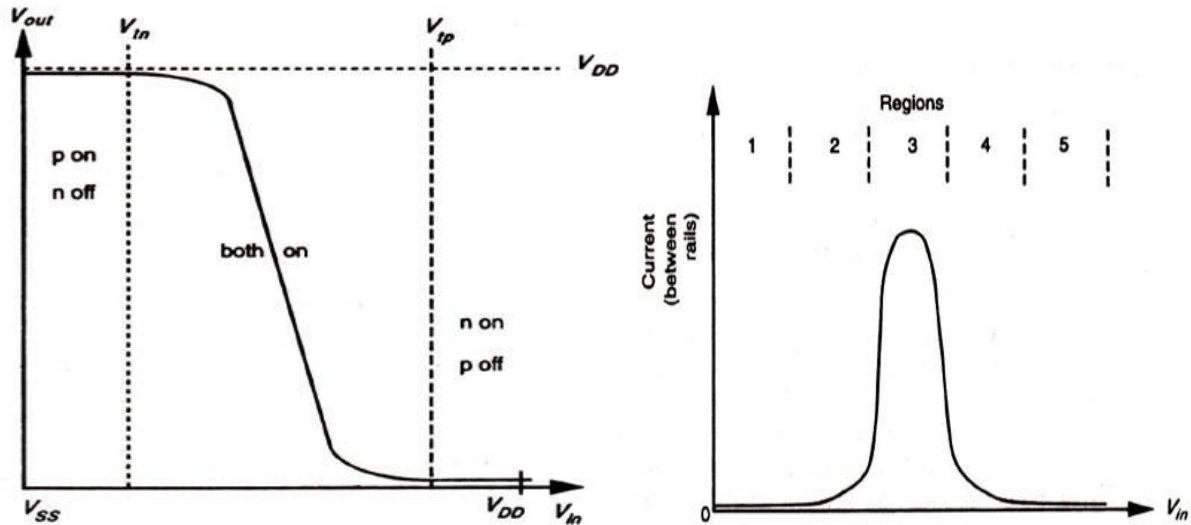
CMOS Inverter :

The inverter is the very important part of all digital designs. Once its operation and properties are clearly understood, Complex structures like NAND gates, adders, multipliers, and microprocessors can also be easily done. The electrical behavior of these complex circuits can be almost completely derived by extrapolating the results obtained for inverters. As shown in the diagram below the CMOS transistor is designed using p-MOS and n-MOS transistors.



In the inverter circuit ,if the input is high .the lower n-MOS device closes to discharge the capacitive load .Similarly ,if the input is low, the top p-MOS device is turned on to charge the capacitive load .At no time both the devices are on ,which prevents the DC current flowing from positive power supply to ground. Qualitatively this circuit acts like the switching circuit, since the p-channel transistor has exactly the opposite characteristics of the n-channel transistor. In the transition region both transistors are saturated and the circuit operates with a large voltage gain. The C-MOS transfer characteristic is shown in the below graph.

Considering the static conditions first, it may be Seen that in region 1 for which $V_{i.} =$ logic 0, we have the p-transistor fully turned on while the n-transistor is fully turned off. Thus no current flows through the inverter and the output is directly connected to V_{DD} through the p-transistor.



Hence the output voltage is logic 1. In region 5, $V_{in} = \text{logic 1}$ and the n-transistor is fully on while the p-transistor is fully off. So, no current flows and a logic 0 appears at the output.

In region 2 the input voltage has increased to a level which just exceeds the threshold voltage of the n-transistor. The n-transistor conducts and has a large voltage between source and drain; so it is in saturation. The p-transistor is also conducting but with only a small voltage across it, it operates in the unsaturated resistive region. A small current now flows through the inverter from V_{DD} to V_{SS} . If we wish to analyze the behavior in this region, we equate the p-device resistive region current with the n-device saturation current and thus obtain the voltage and current relationships.

Region 4 is similar to region 2 but with the roles of the p- and n-transistors reversed. However, the current magnitudes in regions 2 and 4 are small and most of the energy consumed in switching from one state to the other is due to the larger current which flows in region 3.

Region 3 is the region in which the inverter exhibits gain and in which both transistors are in saturation.

The currents in each device must be the same, since the transistors are in series. So, we can write that

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

where

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

and

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

Since both transistors are in saturation, they act as current sources so that the equivalent circuit in this region is two current sources in series between V_{DD} and V_{SS} with the output voltage coming from their common point. The region is inherently unstable in consequence and the changeover from one logic level to the other is rapid.

Determination of Pull-up to Pull-Down Ratio ($Z_{p.u}/Z_{p.d}$) For an NMOS Inverter driven by another NMOS Inverter :

Let us consider the arrangement shown in Fig.(a). In which an inverter is driven from the output of another similar inverter. Consider the depletion mode transistor for which $V_{gs} = 0$ under all conditions, and also assume that in order to cascade inverters without degradation the condition

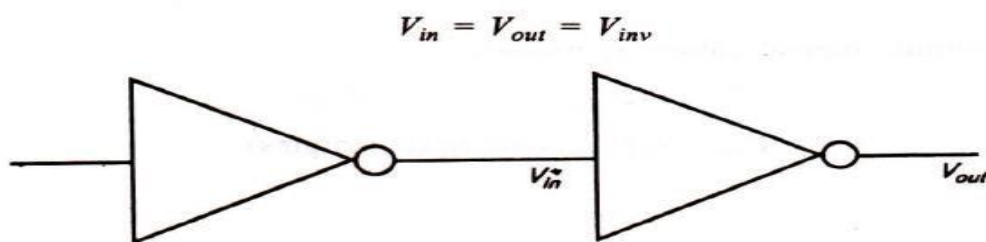


Fig.(a). Inverter driven by another inverter.

For equal margins around the inverter threshold, we set $V_{inv} = 0.5V_{DD}$. At this point both transistors are in saturation and we can write that

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

In the depletion mode $I_{ds} = K \frac{W_{p.u.}}{L_{p.u.}} \frac{(-V_{td})^2}{2}$ since $V_{gs} = 0$

and in the enhancement mode

$$I_{ds} = K \frac{W_{p.d.}}{L_{p.d.}} \frac{(V_{inv} - V_t)^2}{2} \text{ since } V_{gs} = V_{inv}$$

Equating (since currents are the same) we have

$$\frac{W_{p.d.}}{L_{p.d.}} (V_{inv} - V_t)^2 = \frac{W_{p.u.}}{L_{p.u.}} (-V_{td})^2$$

Where $W_{p.d.}$, $L_{p.d.}$, $W_{p.u.}$ And $L_{p.u.}$ are the widths and lengths of the pull-down and pull-up transistors respectively.

So, we can write that

$$Z_{p.d.} = \frac{L_{p.d.}}{W_{p.d.}}; Z_{p.u.} = \frac{L_{p.u.}}{W_{p.u.}}$$

we have

$$\frac{1}{Z_{p.d.}} (V_{inv} - V_t)^2 = \frac{1}{Z_{p.u.}} (-V_{td})^2$$

whence

$$V_{inv} = V_t - \frac{V_{td}}{\sqrt{Z_{p.u.}/Z_{p.d.}}}$$

The typical, values for V_t , V_{inv} and V_{td} are

$$V_t = 0.2V_{DD}; V_{td} = -0.6V_{DD}$$

$$V_{inv} = 0.5V_{DD} \text{ (for equal margins)}$$

Substituting these values in the above equation ,we get

$$0.5 = 0.2 + \frac{0.6}{\sqrt{Z_{p.u.}/Z_{p.d.}}}$$

Here

$$\sqrt{Z_{p.u.}/Z_{p.d.}} = 2$$

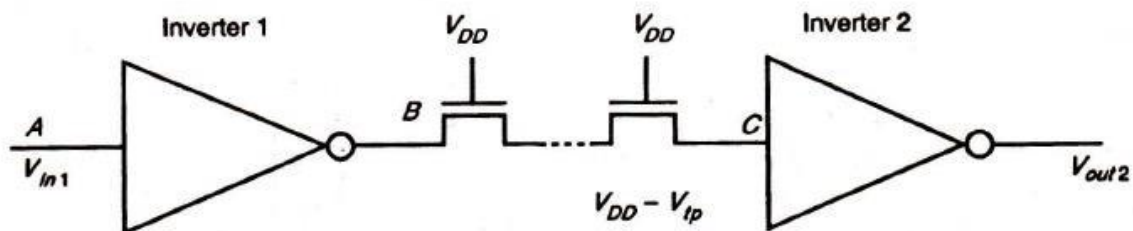
So,we get

$$Z_{p.u.}/Z_{p.d.} = 4/1$$

This is the ratio for pull-up to pull down ratio for an inverter directly driven by another inverter.

Pull -Up to Pull-Down ratio for an NMOS Inverter driven through one or more Pass Transistors

Let us consider an arrangement in which the input to inverter 2 comes from the output of inverter 1 but passes through one or more NMOS transistors as shown in Fig. Below (These transistors are called pass transistors).



The connection of pass transistors in series will degrade the logic 1 level / into inverter 2 so that the output will not be a proper logic 0 level. The critical condition is , when point A is at 0 volts and B is thus at \$V_{DD}\$. But the voltage into inverter 2at point C is now reduced from \$V_{DD}\$ by the threshold voltage of the series pass transistor. With all pass transistor gates connected to \$V_{DD}\$ there is a loss of \$V_{tp}\$, however many are connected in series, since no static current flows

through them and there can be no voltage drop in the channels. Therefore, the input voltage to inverter 2 is

$$V_{in2} = V_{DD} - V_{tp}$$

Where V_{tp} = threshold voltage for a pass transistor.

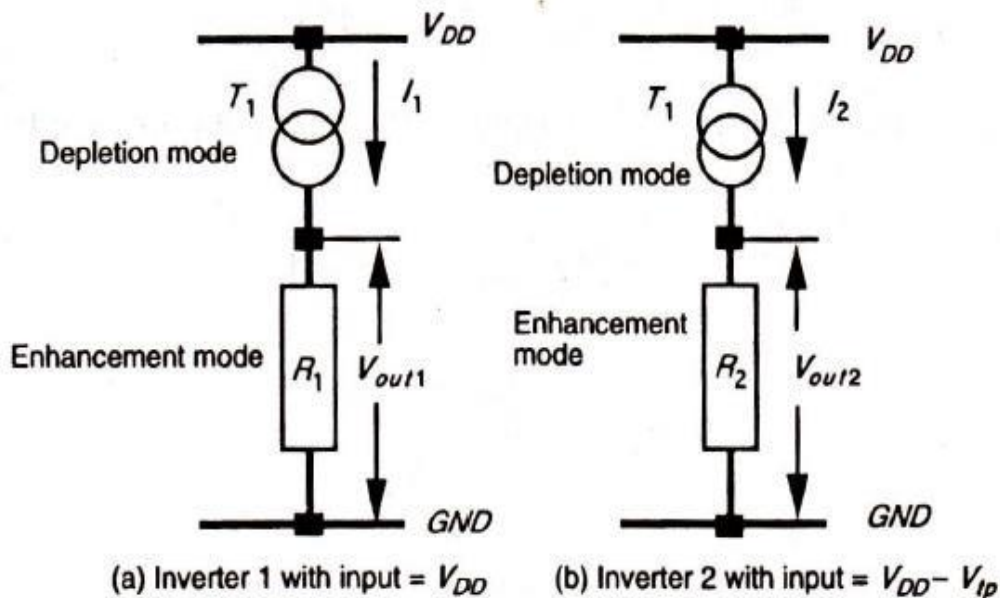
Let us consider the inverter 1 shown in Fig.(a) with input = V_{DD} . If the input is at V_{DD} , then the pull-down transistor T2 is conducting but with a low voltage across it; therefore, it is in its resistive region represented by R_1 in Fig.(a) below. Meanwhile, the pull up transistor T1 is in saturation and is represented as a current source.

For the pull down transistor

$$R_1 = \frac{V_{ds1}}{I_{ds}} = \frac{1}{K} \frac{L_{p.d.1}}{W_{p.d.1}} \left(\frac{1}{V_{DD} - V_t - \frac{V_{ds1}}{2}} \right)$$

$$I_{ds} = K \frac{W_{p.d.1}}{L_{p.d.1}} \left((V_{DD} - V_t) V_{ds1} - \frac{V_{ds1}^2}{2} \right)$$

Since V_{ds} is small, $V_{ds}/2$ can be neglected in the above expression.



So,

$$R_1 \doteq \frac{1}{K} Z_{p.d.1} \left(\frac{1}{V_{DD} - V_t} \right)$$

Now, for depletion mode pull-up transistor in saturation with $V_{gs} = 0$

$$I_1 = I_{ds} = K \frac{W_{p.u.1}}{L_{p.u.1}} \frac{(-V_{td})^2}{2}$$

The product

$$I_1 R_1 = V_{out1}$$

So,

$$V_{out1} = I_1 R_1 = \frac{Z_{p.d.1}}{Z_{p.u.1}} \left(\frac{1}{V_{DD} - V_t} \right) \frac{(V_{td})^2}{2}$$

Let us now consider the inverter 2 Fig.b .when input = $V_{DD} - V_{tp}$.

$$R_2 \doteq \frac{1}{K} Z_{p.d.2} \frac{1}{((V_{DD} - V_{tp}) - V_t)}$$

$$I_2 = K \frac{1}{Z_{p.u.2}} \frac{(-V_{td})^2}{2}$$

Whence,

$$V_{out2} = I_2 R_2 = \frac{Z_{p.d.2}}{Z_{p.u.2}} \left(\frac{1}{V_{DD} - V_{tp} - V_t} \right) \frac{(-V_{td})^2}{2}$$

If inverter 2 is to have the same output voltage under these conditions then $V_{out1} = V_{out2}$. That is

$$I_1 R_1 = I_2 R_2$$

therefore

$$\frac{Z_{p.u.2}}{Z_{p.d.2}} = \frac{Z_{p.u.1}}{Z_{p.d.1}} \frac{(V_{DD} - V_t)}{(V_{DD} - V_{tp} - V_t)}$$

Considering the typical values

$$V_t = 0.2V_{DD}$$

$$V_{tp} = 0.3V_{DD}^*$$

$$\frac{Z_{p.u.2}}{Z_{p.d.2}} = \frac{Z_{p.u.1}}{Z_{p.d.1}} \frac{0.8}{0.2}$$

Therefore

$$\frac{Z_{p.u.2}}{Z_{p.d.2}} \div 2 = \frac{Z_{p.u.1}}{Z_{p.d.1}} = \frac{8}{1}$$

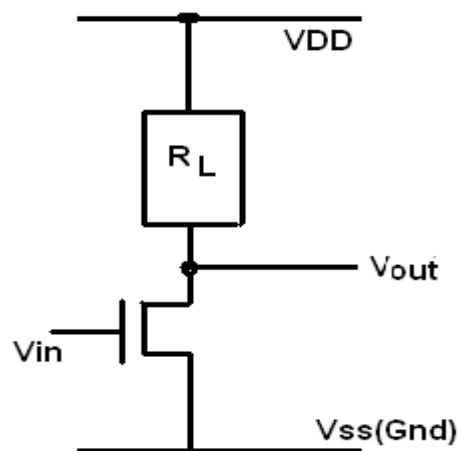
From the above theory it is clear that, for an n-MOS transistor

- (i). An inverter driven directly from the output of another should have a $Z_{p.u.}/Z_{p.d.}$ Ratio Of $\geq 4/1$.
- (ii). An inverter driven through one or more pass transistors should have a $Z_{p.u.}/Z_{p.d.}$ ratio of $\geq 8/1$

ALTERNATIVE FORMS OF PULL –UP:

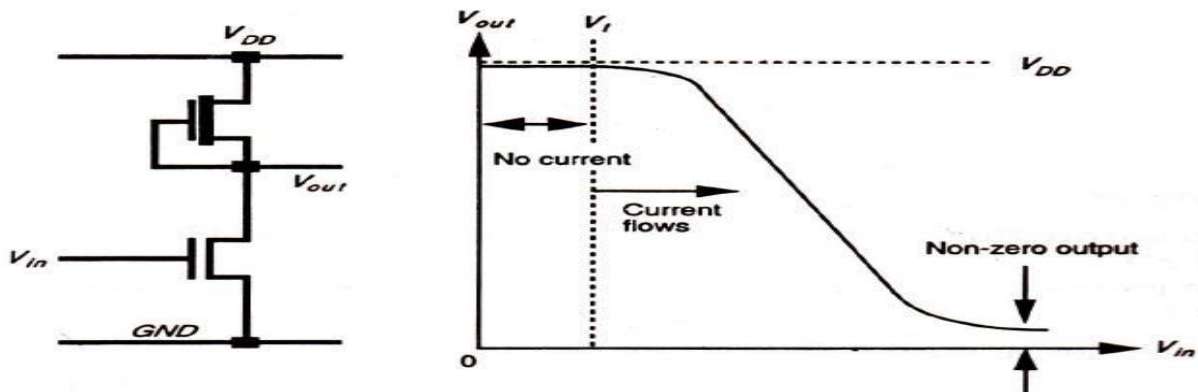
Generally the inverter circuit will have a depletion mode pull-up transistor as its load. But there are also other configurations. Let us consider four such arrangements.

(i) **Load resistance R_L** : This arrangement consists of a load resistor as a pull-up as shown in the diagram below. But it is not widely used because of the large space requirements of resistors produced in a silicon substrate.



2. NMOS depletion mode transistor pull-up : This arrangement consists of a depletion mode transistor as pull-up. The arrangement and the transfer characteristic are shown below. In this type of arrangement we observe

(a) Dissipation is high , since rail to rail current flows when $V_{in} = \text{logical 1}$.



NMOS depletion mode transistor pull-up and transfer characteristic

(b) Switching of output from 1 to 0 begins when V_{in} exceeds V_t of pull-down device.

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

3. NMOS enhancement mode pull-up : This arrangement consists of a n-MOS enhancement mode transistor as pull-up. The arrangement and the transfer characteristic are shown below.

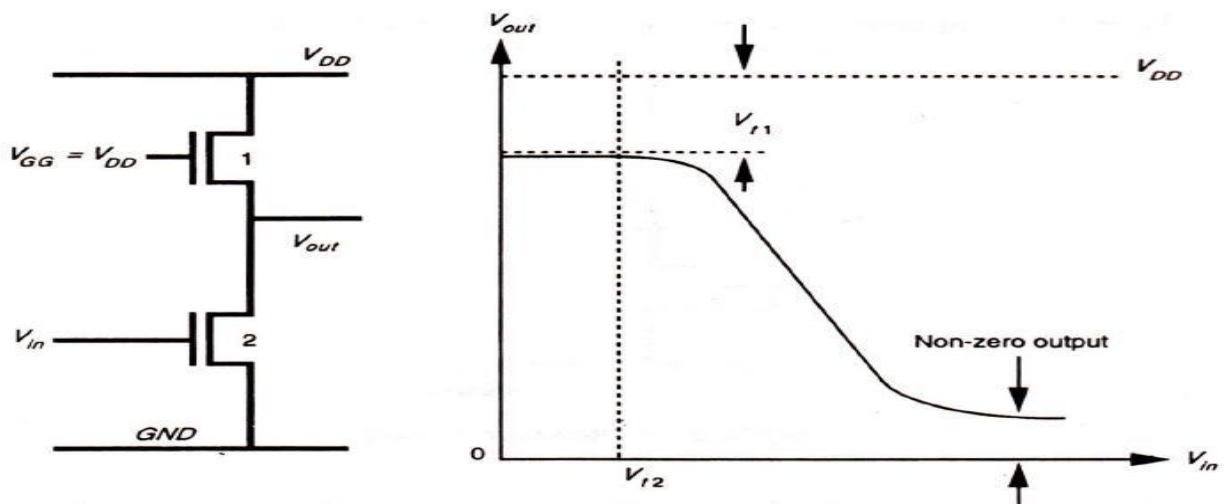
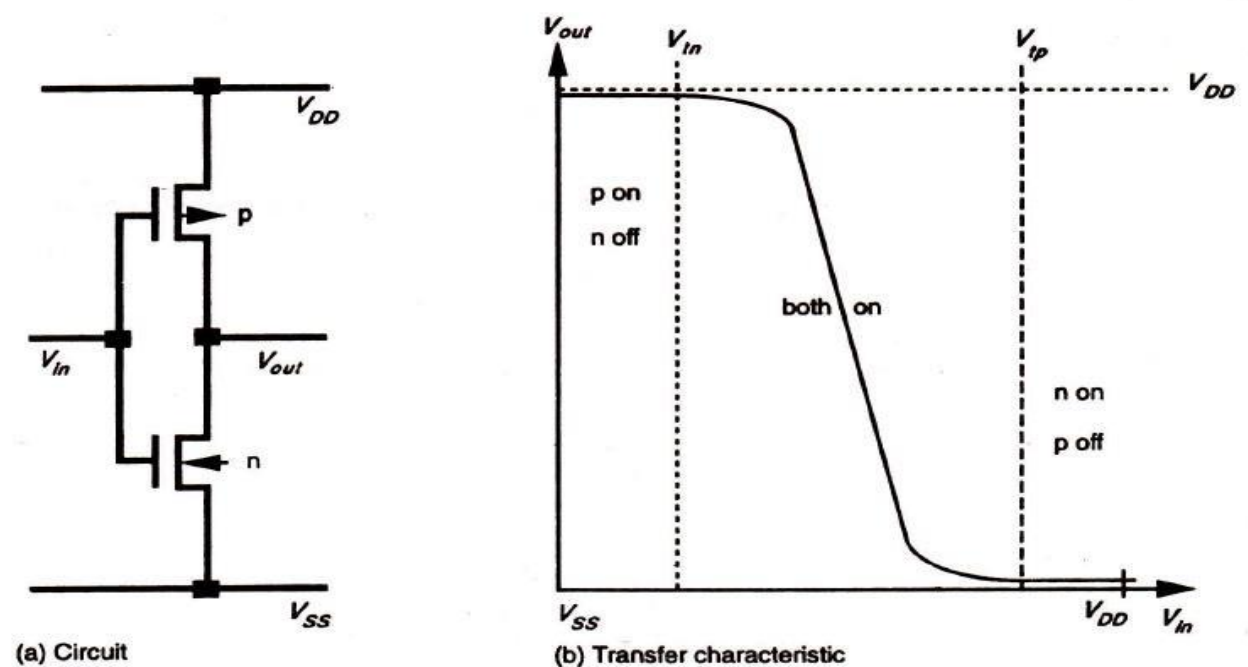


Fig: (a) NMOS enhancement mode pull-up (b) Transfer Characteristic Graph

The important features of this arrangement are

- (a) Dissipation is high since current flows when $V_{in} = \text{logical 1}$ (V_{GG} is returned to V_{DD}).
- (b) V_{out} can never reach V_{DD} (logical 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 transistor pull-up (CMOS) : This arrangement consists of a C-MOS arrangement as pull-up. The arrangement and the transfer characteristic are shown below

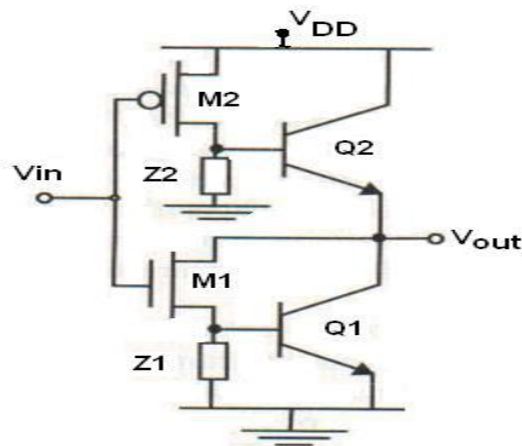


The salient features of this arrangement are:

- (a) No current flows either for logical 0 or for logical 1 inputs.
- (b) Full logical 1 and 0 levels are presented at the output.
- (c) For devices of similar dimensions the p-channel is slower than the n-channel device.

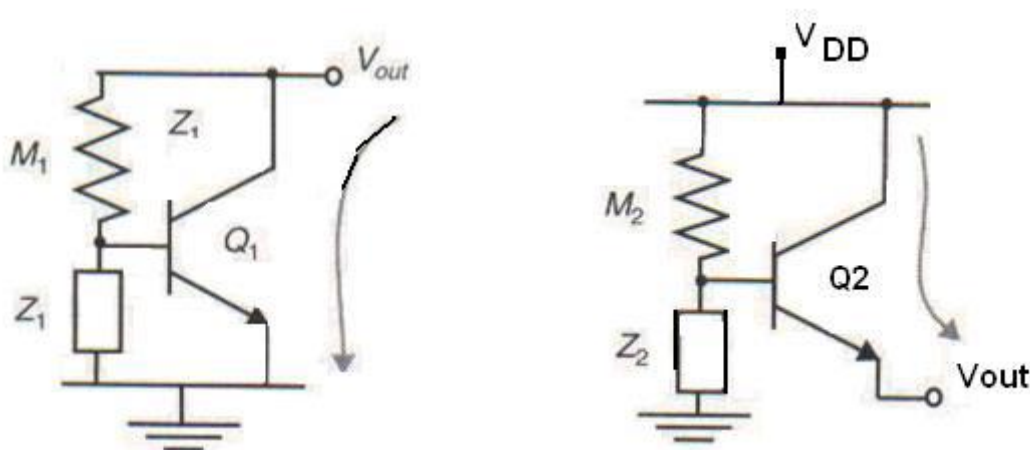
THE BiCMOS INVERTER :

A BiCMOS inverter, consists of a PMOS and NMOS transistor (M2 and M1), two NPN bipolar junction transistors,(Q2 and Q1), and two impedances which act as loads(Z2 and Z1) as shown in the circuit below.



When input, V_{in} , is high (V_{DD}), the NMOS transistor (M1), turns on, causing Q1 to conduct, while M2 and Q2 are off, as shown in figure (b) . Hence , a low (GND) voltage is translated to the output V_{out} . On the other hand, when the input is low, the M2 and Q2 turns on, while m1 and Q1 turns off, resulting to a high output level at the output as shown in Fig.(b).

In steady-state operation, Q1 and Q2 never turns on or off simultaneously, resulting to a lower power consumption. This leads to a push-pull bipolar output stage. Transistors m1 and M2, on the other hand, works as a phase-splitter, which results to a higher input impedance.



The impedances $Z2$ and $Z1$ are used to bias the base-emitter junction of the bipolar transistor and to ensure that base charge is removed when the transistors turn off. For example

when the input voltage makes a high-to-low transition, M1 turns off first. To turn off Q1, the base charge must be removed, which can be achieved by Z1. With this effect, transition time reduces. However, there exists a short time when both Q1 and Q2 are on, making a direct path from the supply (V_{DD}) to the ground. This results to a current spike that is large and has a detrimental effect on both the noise and power consumption, which makes the turning off of the bipolar transistor fast .

Comparison of BiCMOS and C-MOS technologies:

The BiCMOS gates perform in the same manner as the CMOS inverter in terms of power consumption, because both gates display almost no static power consumption.

When comparing BiCMOS and CMOS in driving small capacitive loads, their performance are comparable, however, making BiCMOS consume more power than CMOS. On the other hand, driving larger capacitive loads makes BiCMOS in the advantage of consuming less power than

CMOS, because the construction of CMOS inverter chains are needed to drive large capacitance loads, which is not needed in BiCMOS.

The BiCMOS inverter exhibits a substantial speed advantage over CMOS inverters, especially when driving large capacitive loads. This is due to the bipolar transistor's capability of effectively multiplying its current.

For very low capacitive loads, the CMOS gate is faster than its BiCMOS counterpart due to small values of C_{int} . This makes BiCMOS ineffective when it comes to the implementation of internal gates for logic structures such as ALU's, where associated load capacitances are small.

BiCMOS devices have speed degradation in the low supply voltage region and also BiCMOS is having greater manufacturing complexity than CMOS.

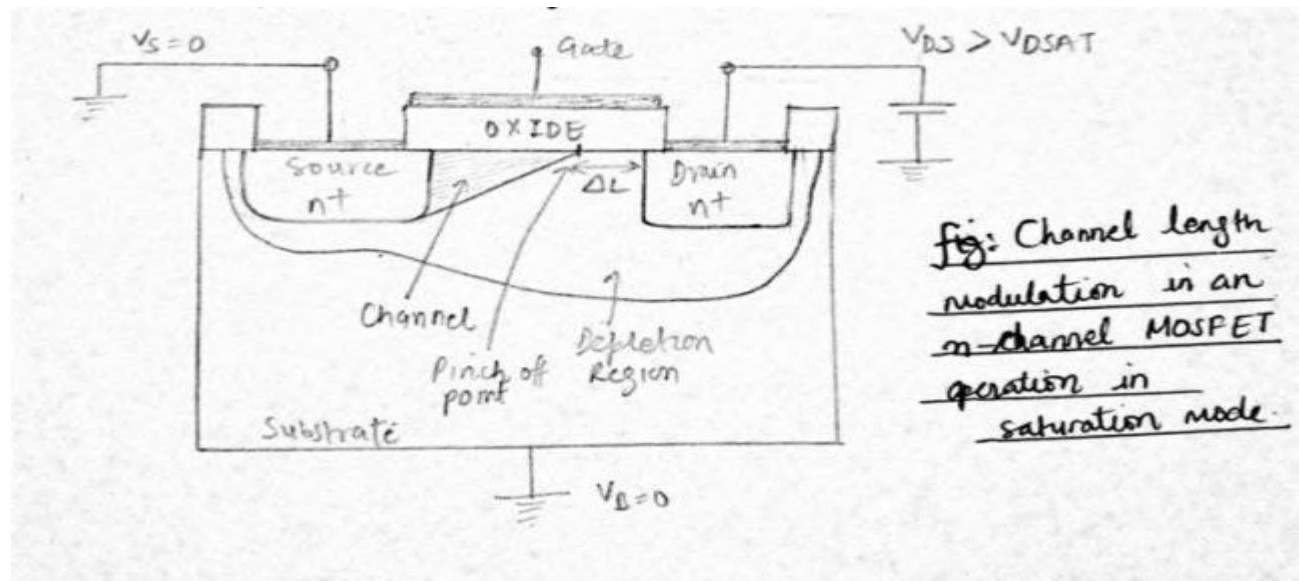
Channel Length Modulation:

Ideal Case: In the saturation region, I_{DS} becomes independent of V_{DS} i.e. in the saturation region channel is pinched off at the drain end and a further increase in V_{DS} has no effect on the channel's shape.

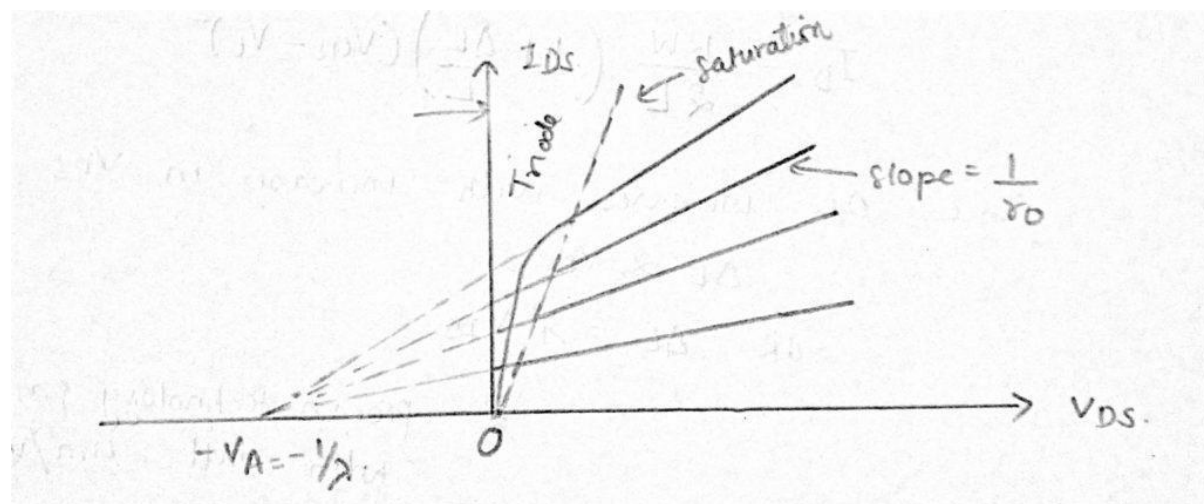
Practical Case: Increase in V_{DS} does affect the channel. In the saturation region, when V_{DS} increases, the channel pinch-off point is moved slightly away from the drain, towards the source as the drain electron field "pushes" it back. The reverse bias depletion region widens and the effective channel length decreases by an amount of ΔL for an increase in V_{DS} .

Thus the channel no longer “touches” the drain and acquires an asymmetrical shape that is thinner at the drain end. This phenomenon is known as channel length modulation.

channel length modulation can be defined as the change or reduction in length of the channel (L) due to increase in the drain to source voltage (V_{DS}) in the saturation region. In large devices, this effect is negligible but for shorter devices $\Delta L/L$ becomes important. Also in the saturation region due to channel length modulation, I_{DS} increases with increase in V_{DS} and also increases with the decrease in channel length L .



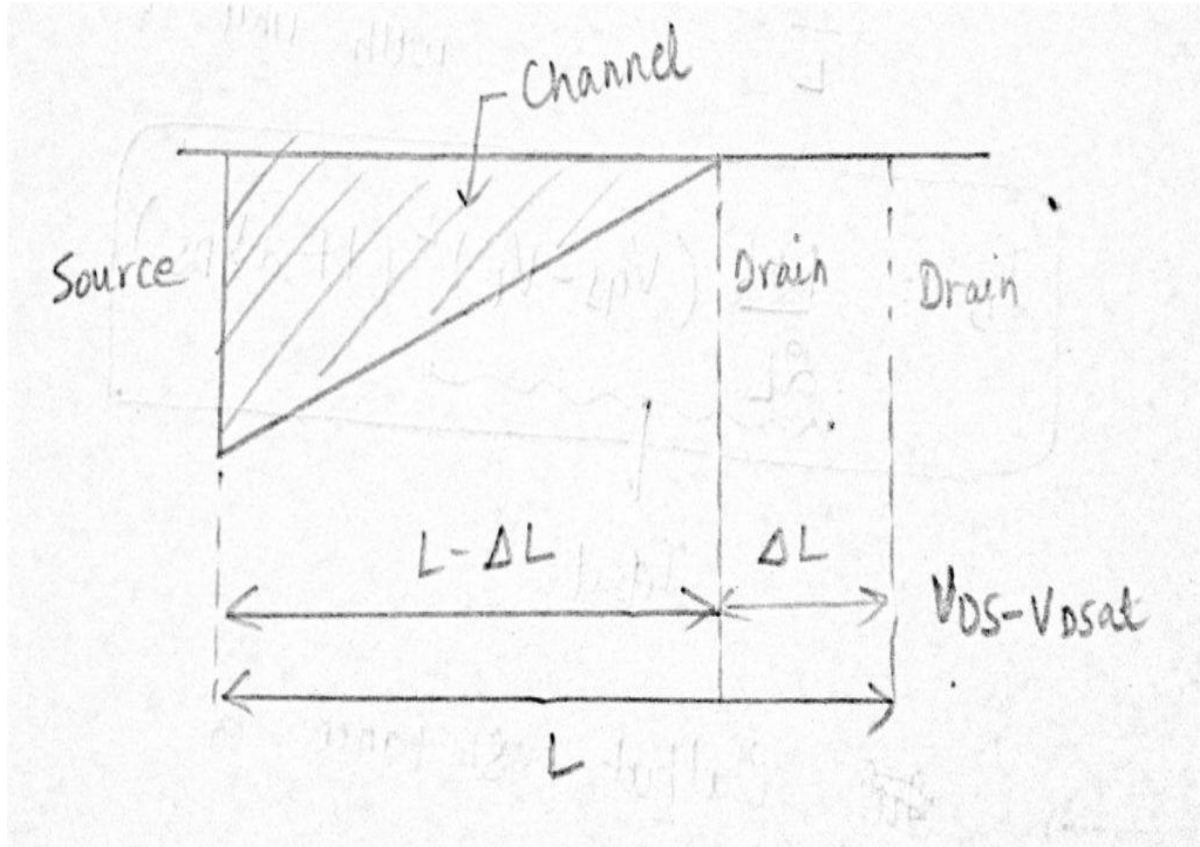
The voltage-current curve is no longer flat in this region.



The drain current with channel length modulation is given by:

$$I_{DS} = I_D = I_{Dsat}(1 + \lambda V_{DS})$$

DERIVATION:



To account for the dependence of I_D on V_{DS} in the saturation region, replace L by $L - \Delta L$. We know that in the saturation region, drain to source current ($I_{DS} = I_D$) is given by:

$$I_D = \frac{kW}{2L}(V_{GS} - V_t)^2$$

$$I_D = \left(\frac{k}{2}\right) \left(\frac{W}{L - \Delta L}\right) (V_{GS} - V_t)^2$$

$$I_D = \left(\frac{k}{2L}\right) \left(\frac{W}{1 - \frac{\Delta L}{L}}\right) (V_{GS} - V_t)^2$$

Assuming $\frac{\Delta L}{L} < 1$

$$I_D = \left(\frac{kW}{2L}\right) \left(1 + \frac{\Delta L}{L}\right) (V_{GS} - V_t)^2$$

Since ΔL increases with increase in V_{DS}

$$\Delta L \propto V_{DS}$$

OR

$$\Delta L = \lambda' V_{DS}$$

where, λ' = process technology parameter with unit $\mu\text{m}/V$.

$$I_D = \left(\frac{kW}{2L}\right) \left(1 + \frac{\lambda' V_{DS}}{L}\right) (V_{GS} - V_t)^2$$

therefore,

$$I_{DS} = I_D = I_{Dsat}(1 + \lambda V_{DS})$$

where,

$\frac{\lambda'}{L} = \lambda$ = process technology parameter with unit V^{-1}

$$I_{Dsat} = \left(\frac{kW}{2L}\right) (V_{GS} - V_t)^2$$

MOS Transistor Transconductance (g_m):

The Transconductance will give 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 change in channel Q_c is

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

where τ_{sd} = electron transit time b/w 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

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

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

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

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

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

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

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

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

Substitute eq (7) in eq (6)

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

$$\text{W.K.T } C_g = \frac{\epsilon_0 \epsilon_{ins} WL}{D} \rightarrow (9)$$

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

$$g_m = \frac{\mu \epsilon_0 \epsilon_{ins} WL}{D L^2} (V_{gs} - V_t)$$

$$g_m = \frac{\mu \epsilon_0 \epsilon_{ins}}{D} \frac{W}{L} (V_{gs} - V_t)$$

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

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

* It is possible to increase the transconductance g_m of MOS transistor by increasing the width, but in this process to increase the ip 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

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

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 (ω_0):

The indication of the frequency response may be obtained from the parameter ω_0 where

$$\omega_0 = \frac{g_m}{C_g} \rightarrow \textcircled{1}$$

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

$$I_{D0} = \frac{\mu}{L^2} V_{DS}$$

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

$$I_{D0} = \frac{1}{\tau_{sd}}$$

W.E.T $g_m = \frac{\mu \epsilon_0 \epsilon_{ins}}{D} \frac{W}{L} (V_{GS} - V_t)$

$$C_g = \frac{\epsilon_0 \epsilon_{ins} WL}{D}$$

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

$$I_{D0} = \frac{\mu \epsilon_0 \epsilon_{ins}}{D} \cdot \frac{W}{L} (V_{GS} - V_t) \cdot \frac{\epsilon_0 \epsilon_{ins} WL}{D}$$

$$I_{D0} = \frac{\mu \epsilon_0 \epsilon_{ins}}{D} \cdot \frac{W}{L} (V_{GS} - V_t) \times \frac{D}{\epsilon_0 \epsilon_{ins} \cdot WL}$$

$$I_{D0} = \frac{\mu}{L^2} (V_{GS} - V_t) \rightarrow (2)$$

at saturation region $V_{DS} = V_{GS} - V_t \rightarrow (3)$

substitute eq (3) in eq (2) we get

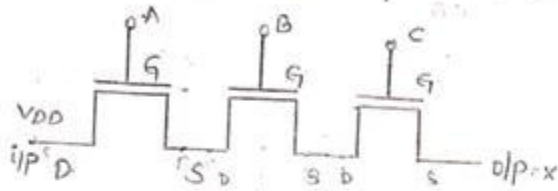
$$W_D = \frac{\mu}{L^2} V_{ds}$$

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

$$W_D = \frac{1}{\tau_{sd}}$$

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



∴ $X = -A \cdot B \cdot C$

- * when the transistor are connected to denote OR logic

∴ ~~$X = -A + B + C$~~ $X = A + B + C$

Example:

→ For N-MOS pass transistor logic

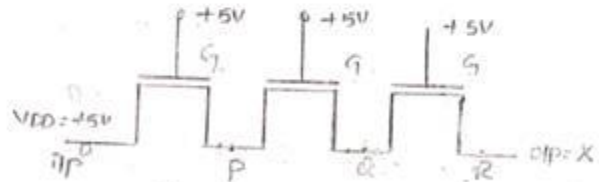
i) $V_D > V_t$ Transistor is 'ON'

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

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

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

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



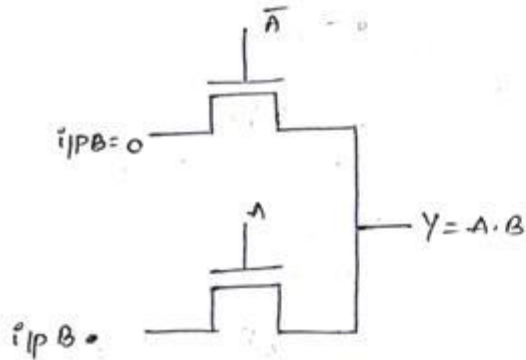
$V_D = 5V, V_G = 5V, V_t = 1V$

ii) $V_G = V_L = 4V = V_S$
 $V_D = 5V$
 $V_D > V_{GS} - V_T$, $V_S = V_G - V_T$
 $5V > 4V$, $V_S = 5 - 1 = 4V = V_P$

i) $V_D > V_L$; $4V > 1V$ so
 Transistor is ON.
 ii) $V_G - V_T = 5 - 1 = 4V$.
 $V_D = V_G - V_L = 4V$.
 $V_Q = V_S = V_G - V_T = 4V$.

Pr R output: $V_G = 5V$, $V_D = 4V$, $V_T = 1V$
 $V_Q = V_D = 4V$, $V_L = 1V$.
 $V_R = 4V = V_S = 5V - 1V$

2) AND logic using PTL



Truth Table for AND logic using PTL:

i/p		output
A	B	
0	0	0
0	1	0
1	0	0
1	1	1

Basic Circuit Concepts

CAPACITANCE, RESISTANCE ESTIMATIONS:

Sheet Resistance R_s and its concepts to MOS:

The sheet resistance is a measure of resistance of thin films that have a uniform thickness. It is commonly used to characterize materials made by semiconductor doping, metal deposition, resistive paste printing, and glass coating.

Example of these processes are: doped semiconductor regions (eg: silicon or polysilicon) and resistors.

Sheet resistance is applicable to two-dimensional systems where the thin film is considered to be a two-dimensional entity. It is analogous to resistivity as used in three-dimensional systems. When the term sheet resistance is used, the current must be flowing along the plane of the sheet, not perpendicular to it.

Model:

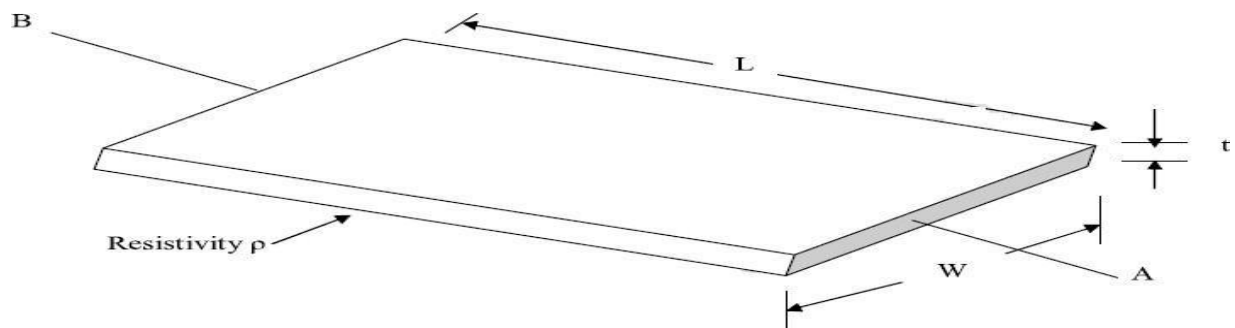
Consider a uniform slab of conducting material of resistivity ρ , of width W , thickness t , and length between faces L as shown below:

$$R_{AB} = \frac{\rho L}{tW} \quad \text{ohm}$$

Where A = cross section area.

$$\text{Thus } R_{AB} = \frac{\rho L}{tW} \quad \text{ohm.}$$

When $L = W$, i.e. a square resistive material, then



$$R_{AB} = \frac{\rho}{t} = R_s$$

Where R_s = ohm per square or sheet resistance.

Thus $R_s = \frac{\rho}{t}$ ohm per square.

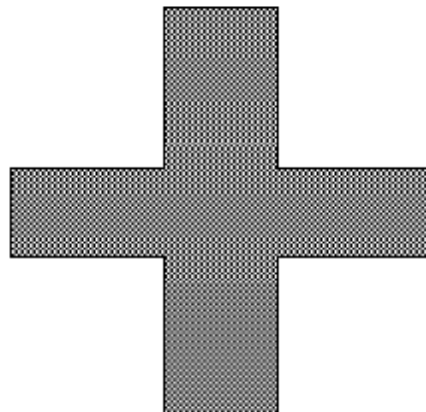
It is completely independent of the area of the square.

Typical sheet resistance R_s of MOS layers

Layer	R _s ohm per square		
	5μm	Orbit	1.2μm
Metal	0.03	0.04	0.04
Diffusion	10 → 50	20 → 45	20 → 45
Silicide	2 → 4	-	-
Polysilicon	15 → 100	15 → 30	15 → 30
n-transistor channel	10 ⁴	2 X 10 ⁴	2 X 10 ⁴
p-transistor channel	2.5 X 10 ⁴	4.5 X 10 ⁴	4.5 X 10 ⁴

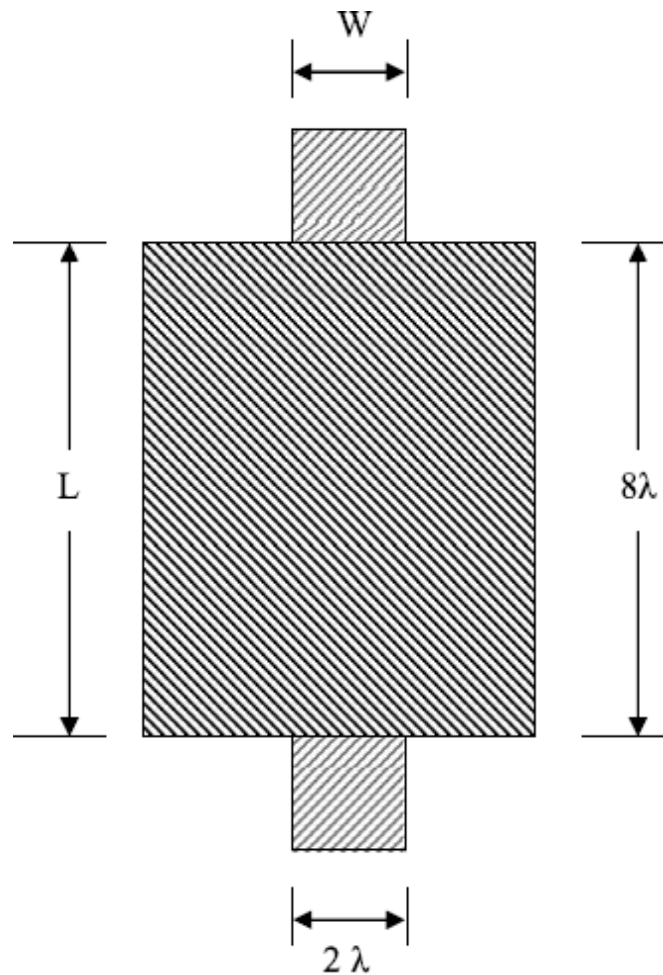
SHEET RESISTANCE CONCEPT APPLIED TO MOS TRANSISTORS AND INVERTERS

The simple n-type pass transistor has a channel length $L = 2\lambda$ and a channel width $W = 2\lambda$. The channel is square



$$R = \text{square} \times R_s \frac{\text{Ohm}}{\text{square}} = R_s = 10^4 \text{ ohm.}$$

The length to width ratio, denoted by Z is 1:1 in this case. Consider one more structure as in diagram below.



$$L = 8\lambda \text{ and } W = 2\lambda$$

$$Z = \frac{L}{W} = 4$$

$$\text{Channel resistance } R = Z R_s = 4 \times 10^4 \text{ Ohm.}$$

This channel can be taken as four $2\lambda \times 2\lambda$ squares in series.

Calculation of ON Resistance of a Simple Inverter

Consider the simple nMOS inverter in Fig.

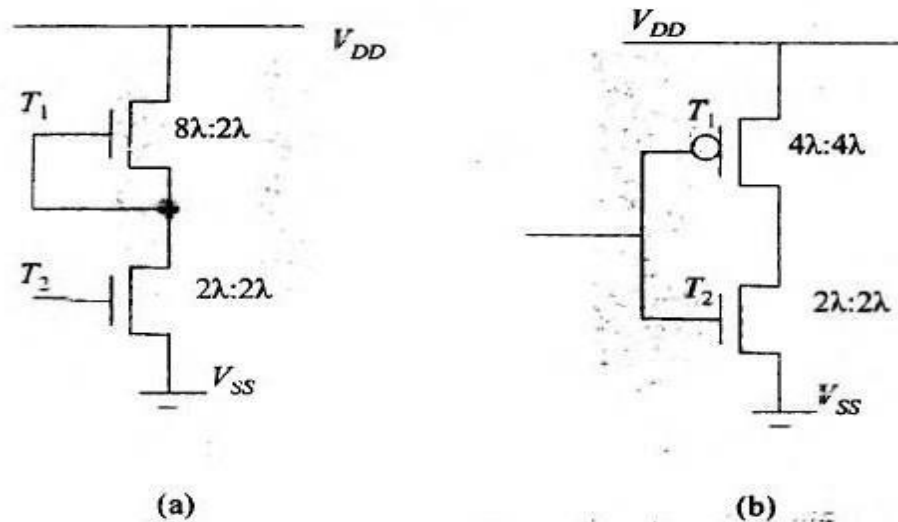


Fig. (a) NMOS Inverter (b) CMOS Inverter resistance calculations

- For the pull-up transistor (depletion mode MOSFET) the $L:W$ value is 4:1, hence the value of Z is 4. $R_{on} = 4$ and value of on resistance is $4R_s$, i.e., $4 \times 10^4 = 40 \text{ k}\Omega$.
- Similarly, for the pull down transistor (enhancement mode MOSFET) the $L:W$ value is 1:1 hence the value of Z is 1. $R_{on} = 1$ and value of resistance is $1R_s$, i.e., $1 \times 10^4 = 10 \text{ k}\Omega$.
- $Z_{p,u}$ to $Z_{p,d} = 4:1$ hence the ON resistance between V_{DD} and V_{SS} is the total series resistance, i.e., $40 \text{ k}\Omega + 10 \text{ k}\Omega = 50 \text{ k}\Omega$.

Consider the simple CMOS inverter in Fig.

- For the pull-up transistor (p-enhancement mode MOSFET) the $L:W$ value is 1:1, hence, the value of Z is 4. $R_{on} = 4$ and value of on resistance is $4 R_s$, i.e., $1 \times 25 \times 10^4 = 25 \text{ k}\Omega$ (from the table value of R_s for p-channel transistor is $2.5 \times 10^4 \text{ ohm/square}$).
- Similarly, for the pull down transistor (n-enhancement mode MOSFET) the $L:W$ value is 1:1 hence the value of Z is 1. $R_{on} = 1$ and value of resistance is $1 R_s$, i.e., $1 \times 10^4 = 10 \text{ k}\Omega$.
- In this case, there is no static resistance between V_{DD} and V_{SS} since at any point of time only one transistor is ON, but not both.
- When $V_{in} = 1$, the ON Resistance is $10 \text{ k}\Omega$, when $V_{in} = 0$ the ON Resistance is $25 \text{ k}\Omega$.

MOS Device Capacitances:

Area Capacitances calculations:

From the concept of the transistors, we studied, it is apparent that as gate is separated from the channel by gate oxide an insulating layer, it has capacitance. Similarly, different interconnects run on the chip and each layer is separated by silicon dioxide.

Area capacitance can be calculated as $C = \frac{\epsilon_o \epsilon_{ins} A}{D}$ farads

Where

D = Thickness of silicon dioxide

A = Area of plates

ϵ_{ins} = Relative permittivity of SiO₂ = 4.0

ϵ_o = 8.85 X 10⁻¹⁴ F/cm (permittivity of free space)

The layer area capacitance is in pF/ μm^2 (where μm = micron = 10⁻⁶ meter)

Typical values of area capacitance are given below in Fig. :

Capacitance	Value in pF $\times 10^{-4}/\mu\text{m}^2$ (Relative values in brackets).					
	5 μm		2 μm		1.2 μm	
Gate to channel	4	(1.0)	8	(1.0)	16	(1.0)
Diffusion (active)	1	(0.25)	1.75	(0.22)	3.75	(0.23)
Polysilicon* to substrate	0.4	(0.1)	0.6	(0.075)	0.6	(0.038)
Metal 1 to substrate	0.3	(0.075)	0.33	(0.04)	0.33	(0.02)
Metal 2 to substrate	0.2	(0.05)	0.17	(0.02)	0.17	(0.01)
Metal 2 to metal 1	0.4	(0.1)	0.5	(0.06)	0.5	(0.03)
Metal 2 to polysilicon	0.3	(0.075)	0.3	(0.038)	0.3	(0.018)

Standard unit of capacitance:

A standard unit is employed that can be used in calculations. The unit is denoted as C_g and is defined as the gate-to-channel capacitance of a MOS transistor having $W = L =$ feature size, that is a 'standard' or 'feature size' square.

C_g may be evaluated for any MOS process.

For example, for 5 μm MOS circuits

Area/standard square = 5 μm X 5 μm = 25 μm^2

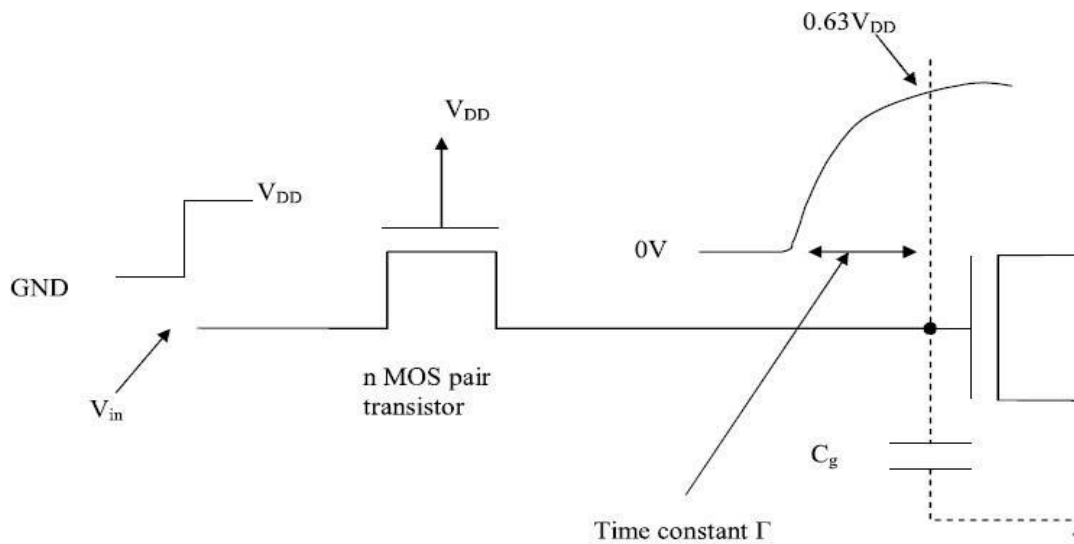
Capacitance value = $4 \times 10^{-4} \text{ pF}/\mu\text{m}^2$
 Thus standard value of $C_g = 25 \mu\text{m}^2 \times 4 \times 10^{-4} \text{ pF}/\mu\text{m}^2$
 = 0.01 pF

For 2 μm MOS circuits $C_g = 0.0032 \text{ pF}$ and for 1.2 μm MOS circuits $C_g = 0.0023 \text{ pF}$

Calculation of Delay unit τ :

The delay unit Γ is the product of 1 R_s and 1 C_g

$$\Gamma = (1 R_s (\text{n-channel}) \times 1 C_g) \text{ seconds}$$



For 5 μm technology
 $\Gamma = 10^4 \text{ ohm} \times 0.01 \text{ pF}$
 = 0.1 n sec

For 2 μm technology
 $\Gamma = 2 \times 10^4 \text{ ohm} \times 0.0032 \text{ pF}$
 = 0.064 n sec

For 1.2 μm (orbit) technology
 $\Gamma = 2 \times 10^4 \text{ ohm} \times 0.0023 \text{ pF}$
 = 0.046 n sec

Practically $\Gamma = 0.2$ to 0.3 n sec for a 5 μm technology because of circuit wiring and parasitic capacitances taken into account.

$$\tau \approx \tau_{sd} = \frac{L^2}{\mu_n V_{ds}} = \frac{25 \mu\text{m}^2 V \text{ sec}}{650 \text{ cm}^2 \cdot 3V} \times \frac{10^9 \text{ n sec cm}^2}{10^8 \mu\text{m}^2}$$

$$= 0.13 \text{ n sec}$$

V_{ds} varies as C_g charges from 0 volts to 63% of V_{DD} in period Γ . Transit time and time constant Γ can be used interchangeably.

Routing Capacitance(Wiring Capacitance):

we considered the area capacitances associated with the layers to substrate and from gate to channel. However, there are other significant sources of capacitance which contribute to the overall wiring capacitance. Three such sources are discussed below.

- Fringing Fields
- Interlayer Capacitances
- Peripheral Capacitance

Fringing Fields:

Capacitance due to fringing field effects can be a major component of the overall capacitance of interconnect wires. For fine line metallization, the value of fringing field capacitance (C_{ff}) can be of the same order as that of the area capacitance. Thus, C_{ff} should be taken into account if accurate prediction of performance is needed.

$$C_{ff} = \epsilon_{\text{SiO}_2} \epsilon_0 l \left[\frac{\pi}{\ln \left\{ 1 + \frac{2d}{t} \left(1 + \sqrt{1 + \frac{t}{d}} \right) \right\}} - \frac{t}{4d} \right]$$

where

l = wire length

t = thickness of wire

d = wire to substrate separation

Then, total wire capacitance

$$C_w = C_{area} + C_{ff}$$

Interlayer Capacitances:

Quite obviously the parallel plate effects are present between one layer and another. For example; some thought on the matter will confirm the fact that, for a given area, metal to polysilicon capacitance must be higher than metal to substrate. The reason for not taking such effects into account for simple calculations is that the effects occur only where layers cross or when one layer underlies another, and in consequence interlayer capacitance is highly dependent on layout. However, for regular structures it is readily calculated and contributes significantly to the accuracy of circuit modeling and delay calculation.

Peripheral Capacitance:

The source and drain n-diffusion regions (n-active regions for Orbit processes) form junctions with the p-substrate or p-well at well-defined and uniform depths; similarly for p-diffusion (p-active) regions in n-substrates or n-wells. For diffusion regions, each diode thus formed has associated with it a peripheral (side-wall) capacitance in picofarads per unit length which, in total, can be considerably greater than the area capacitance of the diffusion region to substrate; the smaller the source or drain area, the greater becomes the relative value of the peripheral capacitance.

For Orbit processes, the n-active and p-active regions are formed by impurity implant at the surface of the silicon and thus, having negligible depth, they have negligible peripheral capacitance.

However, for n- and p-regions formed by a diffusion process, the peripheral capacitance is important and becomes particularly so as we shrink the device dimensions. In order to calculate the total diffusion capacitance we must add the contributions of area and peripheral components

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

Analytic Inverter Delays:

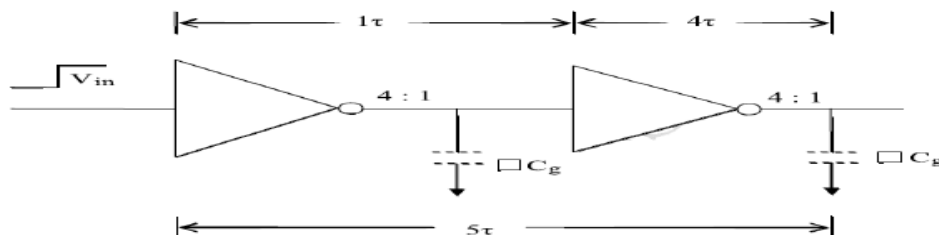
Consider 4 : 1 ratio nMOS inverter. To get 4 : 1 Z_{pu} to Z_{pd} ratio, R_{pu} will be 4 R_{pd}

$$R_{pu} = 4 R_s = 40k\Omega$$

$$\text{Meanwhile } R_{pd} = 1R_s = 10k\Omega$$

Consider a pair of cascaded inverters, the delay over the pair is constant. This is observed in diagram below:

NMOS inverter pair delay:



Assuming $\tau = 0.3$ nsec, over all delay = $\tau + 4\tau = 5\tau$.

The general equation is $\tau_d = \left(1 + \frac{Z_{p,u}}{Z_{p,d}}\right) \tau$

Consider CMOS inverter, the nmos rule does not apply. The gate capacitance is

CMOS inverter pair delay:

When considering CMOS inverters, the nMOS ratio rule no longer applies, but we must allow for the natural (R_s) asymmetry of the usually equal size pull-up p-transistors and the n-type pull-down transistors. Figure 5.21 shows the theoretical delay associated with a pair of minimum size (both n- and p-transistors) lambda-based inverters. Note that the gate capacitance ($=2 \square C_g$) is double that of the comparable nMOS inverter since the input to a CMOS inverter is connected to both transistor gates. Note also the allowance made for the differing channel resistances.

The asymmetry of resistance values can be eliminated by increasing the width of the p-device channel by a factor of two or three, but it should be noted that the gate input capacitance of the p-transistor is also increased by the same factor. This, to some extent, offsets the speed-up due to the drop in resistance, but there is a small net gain since the wiring capacitance will be the same.

GATE LEVEL DESIGN 17

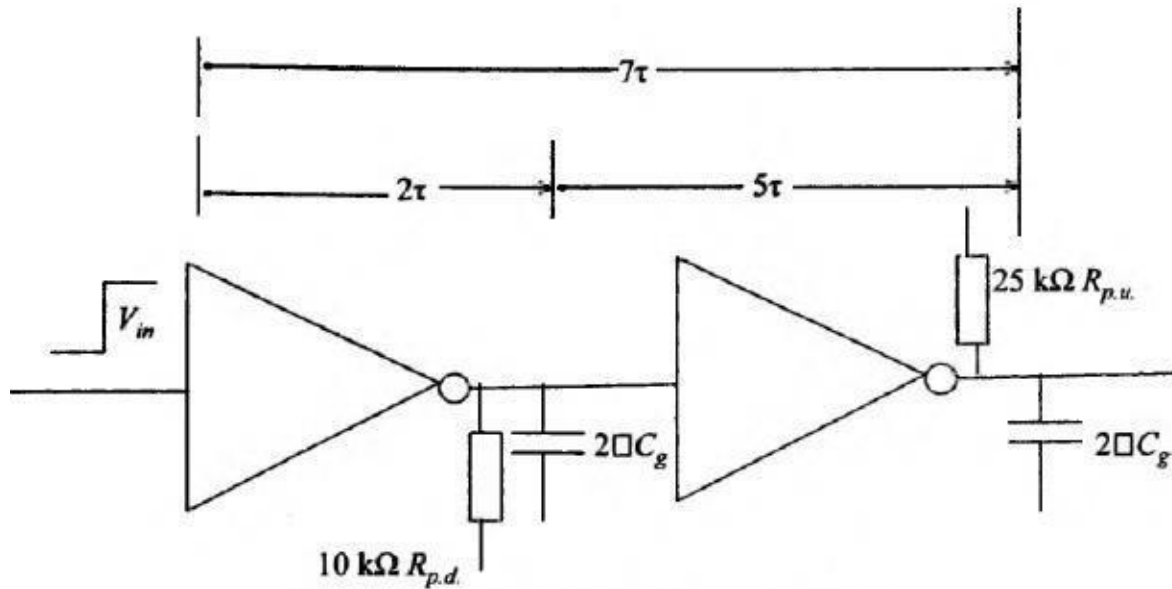


Fig. 5.21 Minimum size CMOS inverter pair delay.

Driving large Capacitive Loads:

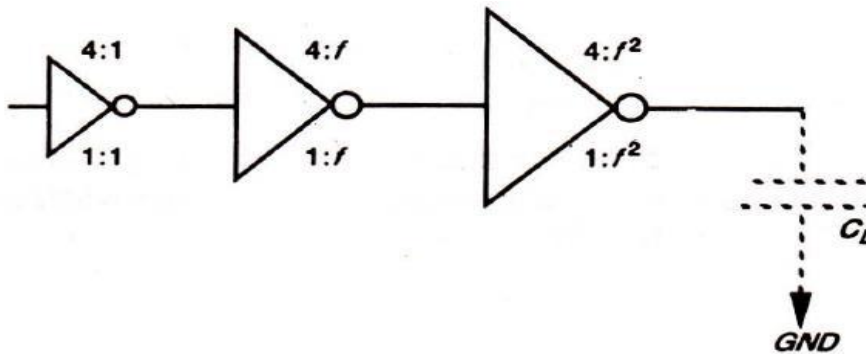
The problem of driving comparatively large capacitive loads arises when signals must be propagated from the chip to off chip destinations . . . Generally, typical off chip capacitances may be several orders higher than on chip C_g values.

Cascaded Inverters as Drivers:

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

Obviously, for MOS circuits, low resistance values for $Z_{p.d.}$ and $Z_{p.u.}$ imply low $L:W$ ratios; in other words, channels must be made very wide to reduce resistance value and, in consequence, an inverter to meet this need occupies a large area. Moreover, because of the large $L:W$ ratio and since length L cannot be reduced below the minimum feature size, the gate region area $L \times W$ becomes significant and a comparatively large capacitance is presented at the input, which in turn slows down the rates of change of voltage which can take place at the input.

The remedy is to use N cascaded inverters, each one of which is larger than the preceding stage by a width factor f as shown in Figure below



Thus, for N even

$$\text{total delay} = \frac{N}{2} 5f\tau = 2.5 Nf\tau \text{ (nMOS)}$$

$$\text{or} = \frac{N}{2} 7f\tau = 3.5 Nf\tau \text{ (CMOS)}$$

and overall delay t_d

$$N \text{ even: } t_d = 2.5eN \tau \text{ (nMOS)}$$

$$\text{or } t_d = 3.5eN \tau \text{ (CMOS)}$$

$$N \text{ odd: } t_d = [2.5(N - 1) + 1]e\tau \text{ (nMOS)}$$

$$\text{or } t_d = [3.5(N - 1) + 2]e\tau \text{ (CMOS)}$$

} for ΔV_{in}

or

$$t_d = [2.5(N - 1) + 4]e\tau \text{ (nMOS)}$$

$$\text{or } t_d = [3.5(N - 1) + 5]e\tau \text{ (CMOS)}$$

} for ∇V_{in}

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 as in Figures.

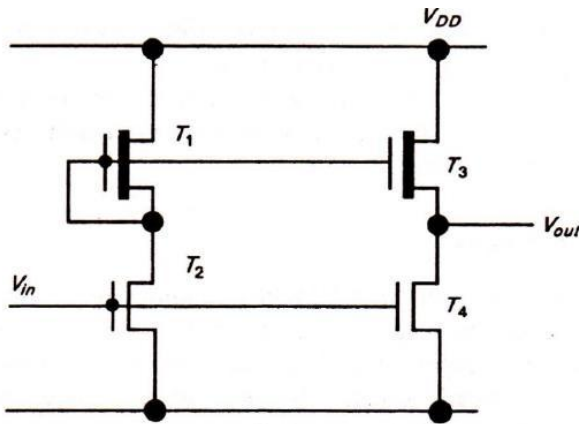


FIGURE 4.12 Inverting type nMOS super buffer.

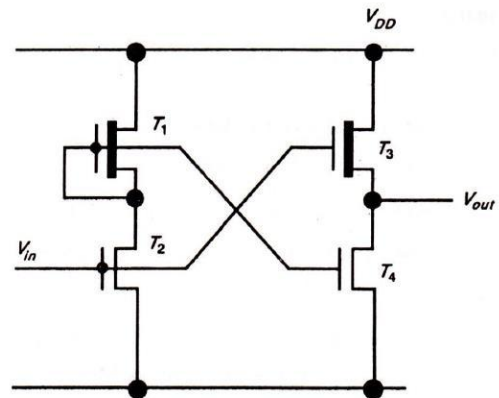
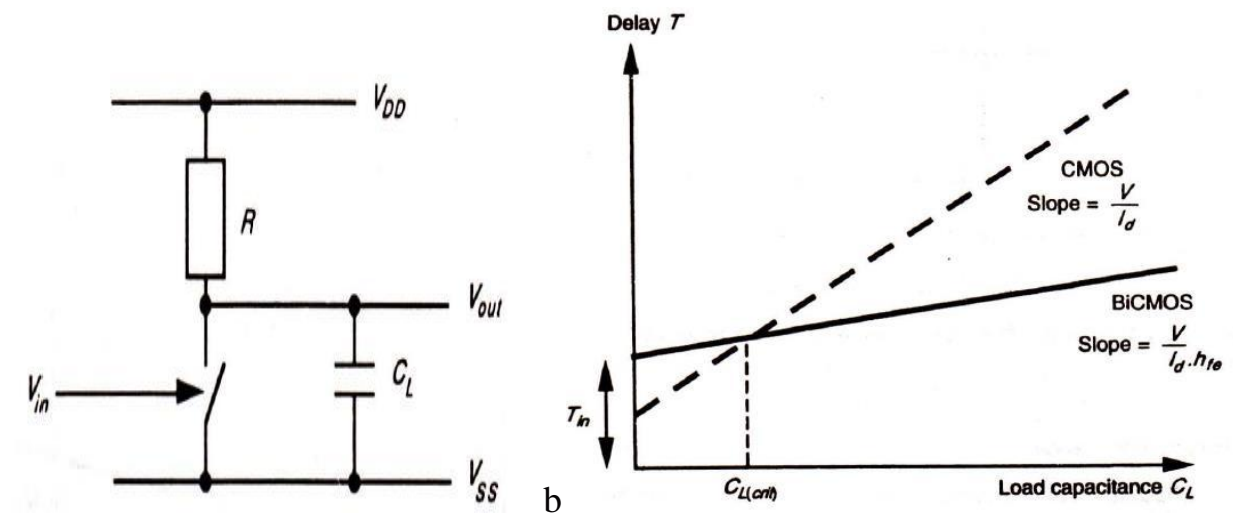


FIGURE 4.13 Non-inverting type nMOS super buffer.

BICMOS Drivers:

The availability of bipolar transistors in BiCMOS technology presents the possibility of using bipolar transistor drivers as the output stage of inverter and logic gate circuits. We have already seen earlier that bipolar transistors have transconductance gm and current/area characteristics that are greatly superior to those of MOS devices. This indicates high current drive capabilities for small areas in silicon. The switching performance of a transistor driving a capacitive load may be visualized initially from the simple model.



Delay of BiCMOS inverter can be described by

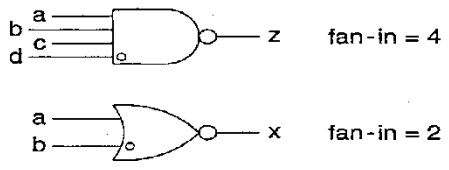
$$T = T_{in} + (V/I_d) (1/h_{fe}) CL$$

Fan in and Fan out:

- Fan-In = Number of inputs to a logic gate
 - 4 input NAND has a FI = 4
 - 2 input NOR has a FI = 2, etc. (See Fig. a below.)
- Fan-Out (FO)= Number of gate inputs which are driven by a particular gate output
 - FO = 4 in Fig. b below shows an output wire feeding an input on four different logic gates
- The circuit delay of a gate is a function of both the Fan-In and the Fan-Out.

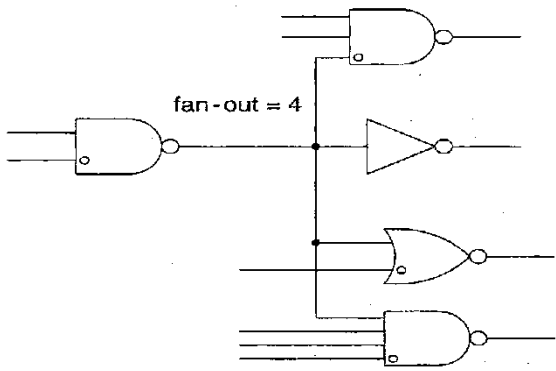
$$\begin{aligned} \text{Ex. } m\text{-input NAND: } t_{dr} &= (R_p/n)(mnC_d + C_r + kC_g) \\ &= t_{\text{internal-r}} + k t_{\text{output-r}} \end{aligned}$$

where n = width multiplier, m = fan-in, k = fan-out, R_p = resistance of min inverter P Tx, C_g = gate capacitance, C_d = source/drain capacitance, C_r = routing (wiring) capacitance.



(a)

Note: The open circle adjacent to a logic gate input denotes the series transistor closest to the output.



(b)

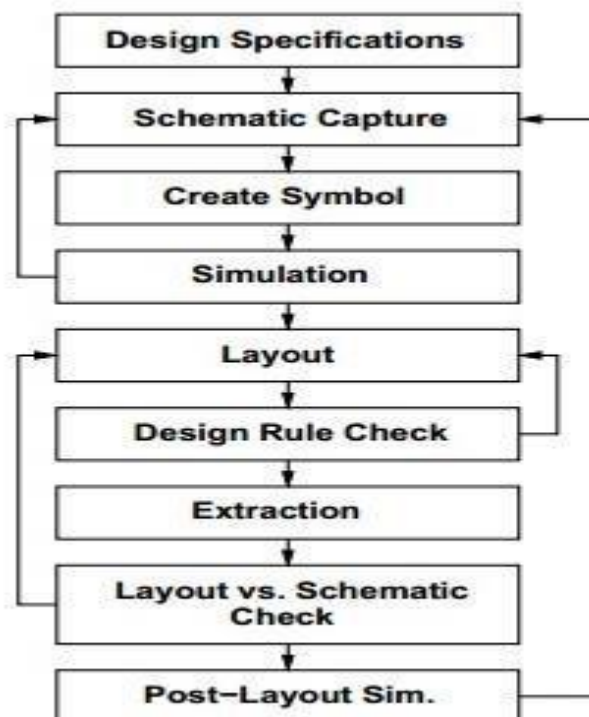
UNIT III

VLSI CIRCUIT DESIGN PROCESSES

VLSI Design Flow:

The VLSI IC circuits design flow is shown in the figure below. The various levels of design are numbered and the blocks show processes in the design flow.

Specifications comes first, they describe abstractly, the functionality, interface, and the architecture of the digital IC circuit to be designed.



Behavioral description is then created to analyze the design in terms of functionality, performance, compliance to given standards, and other specifications.

RTL description is done using HDLs. This RTL description is simulated to test functionality. From here onwards we need the help of EDA tools.

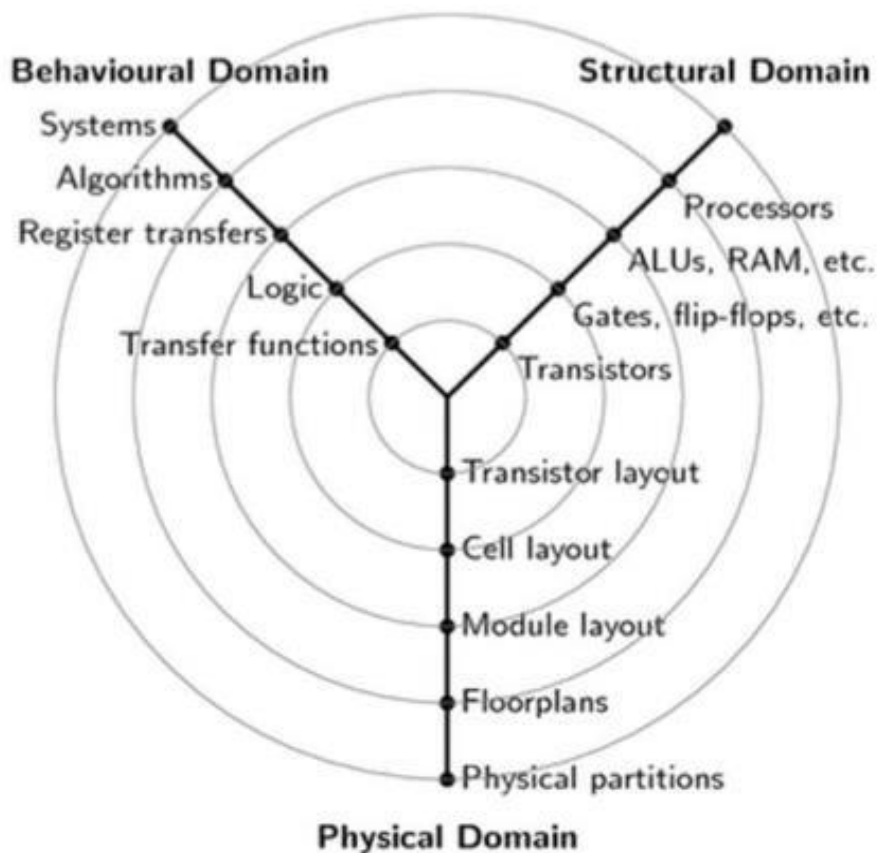
RTL description is then converted to a gate-level netlist using logic synthesis tools. A gatelevel netlist is a description of the circuit in terms of gates and connections between them, which are made in such a way that they meet the timing, power and area specifications.

Finally, a physical layout is made, which will be verified and then sent to fabrication.

Y CHART:

- The Gajski-Kuhn Y-chart is a model, which captures the considerations in designing semiconductor devices.
- The three domains of the Gajski-Kuhn Y-chart are on radial axes. Each of the domains can be divided into levels of abstraction, using concentric rings.
- At the top level (outer ring), we consider the architecture of the chip; at the lower levels (inner rings), we successively refine the design into finer detailed implementation –

- Creating a structural description from a behavioral one is achieved through the processes of high-level synthesis or logical synthesis.
- Creating a physical description from a structural one is achieved through layout synthesis.



Gajski-Kuhn Y-chart

MOS Layers:

MOS circuits are formed on four basic layers. they are:

- N-diffusion
- P-diffusion
- Poly silicon
- Metal

These layers are isolated by one another by thick or thin silicon dioxide insulating layers. Thin oxide mask region includes n-diffusion / p-diffusion and transistor channel.

Stick diagrams:

Stick diagrams may be used to convey layer information through the use of a color code.

For example:

- N-diffusion -green

- █ poly -- red
- █ Blue -- metal
- █ yellow --implant
- █ Black --contact area

Encodings for NMOS process:

COLOR	STICK ENCODING	LAYERS	MASK LAYOUT ENCODING	CIF LAYER
GREEN		n-diffusion (n ⁺ active) Thinox*		ND
RED		Poly silicon		NP
BLUE		Metal 1		NM
BLACK		Contact out		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				
n-type depletion mode transistor nMOS only				

Figure 1: NMOS encodings

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

Figure1 shows when a n-transistor is formed: a transistor is formed when a green line (n+ diffusion) crosses a red line (poly) completely. Figure also shows how a depletion mode transistor is represented in the stick format.

Encodings for MOS process:





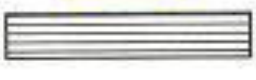



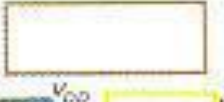

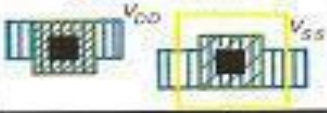

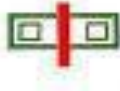


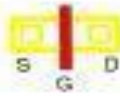
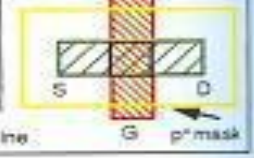
COLOR	STICK ENCODING	LAYERS	MASK LAYOUT ENCODING	CIF LAYER
GREEN	Encoding as in Color plate 1(a)	n-diffusion (n ⁺ active) Thin _{ox} *	Encoding as in Color plate 1(a)	CAA or CNA
RED		Poly _{silicon}		CPF
BLUE		Metal 1		CMF
BLACK		Contact out		CC
GRAY		Overglass		COG
YELLOW (STICK)		p-diffusion (p ⁺ active)		CAA or CPA
YELLOW	Not shown on diagram	p ⁺ mask		CPP
DARK BLUE OR PURPLE		Metal 2		CMS
BLACK		VIA		CVA
BROWN		p-well		CPW
BLACK		V _{DD} or V _{SS} contact		CC
FEATURE	FEATURE (STICK)	FEATURE (SYMBOL)	FEATURE (MASK)	
n-type enhancement mode transistor (as in Color plate 1(a)) Transistor length to width ratio L:W may be shown.				
p-type enhancement mode transistor				
Note: p-type transistors are placed above and n-type below the demarcation line.				

Figure 2: CMOS encodings

figure 2 shows when a n-transistor is formed: a transistor is formed when a green line (n+ diffusion) crosses a red line (poly) completely.

Figure 2 also shows when a p-transistor is formed: a transistor is formed when a yellow line (p+ diffusion) crosses a red line (poly) completely.

Encoding for BJT and MOSFETs:

COLOR	STICK ENCODING	LAYERS	MASK LAYOUT ENCODING	CIF LAYER
ORANGE	MONOCHROME	Polysilicon 2	MONOCHROME	CPS
SEE COLOR PLATE 1(c)		Bipolar npn transistor	see Figure 3-13(f)	Not applicable
PINK	Not separately encoded	p-base of bipolar npn transistor		CBA
PALE GREEN	Not separately encoded	Buried collector of bipolar npn transistor		CCA
FEATURE	FEATURE (STICK) (MONOCHROME)	FEATURE (SYMBOL) (MONOCHROME)	FEATURE (MASK) (MONOCHROME)	
<i>n</i> -type enhancement poly 2 transistor Transistor length to width ratio L:W may be shown.				
<i>p</i> -type enhancement poly 2 transistor Note: <i>p</i> -type transistors are placed above and <i>n</i> -type transistors below the demarcation line.				
<i>npn</i> bipolar transistor			See Figure 3-13(f) and Color plate 6	

Figure 3: Bi CMOS encodings

There are several layers in an nMOS chip:

Paths of metal (usually aluminum) a further thick layer of silicon dioxide with contact cuts through the silicon dioxide where connections are required.

The three layers carrying paths can be considered as independent conductors that only interact where polysilicon crosses diffusion to form a transistor. These tracks can be drawn as stick diagrams with

- Diffusion in green
- Polysilicon in red
- Metal in blue

Using black to indicate contacts between layers and yellow to mark regions of implant in the channels of depletion mode transistors. With CMOS there are two types of diffusion: n-type is drawn in green and p-type in brown. These are on the same layers in the chip and must not meet. In fact, the method of fabrication required that they be kept relatively far apart. Modern CMOS processes usually support more than one layer of metal. Two are common and three or more are often available. Actually, these conventions for colors are not universal; in particular, industrial (rather than academic) systems tend to use red for diffusion and green for polysilicon. Moreover, a shortage of colored pens normally means that both types of diffusion in CMOS are colored green and the polarity indicated by drawing a circle round p-type transistors or simply inferred from the context. Colorings for multiple layers of metal are even less standard.

There are three ways that an NMOS inverter might be drawn:

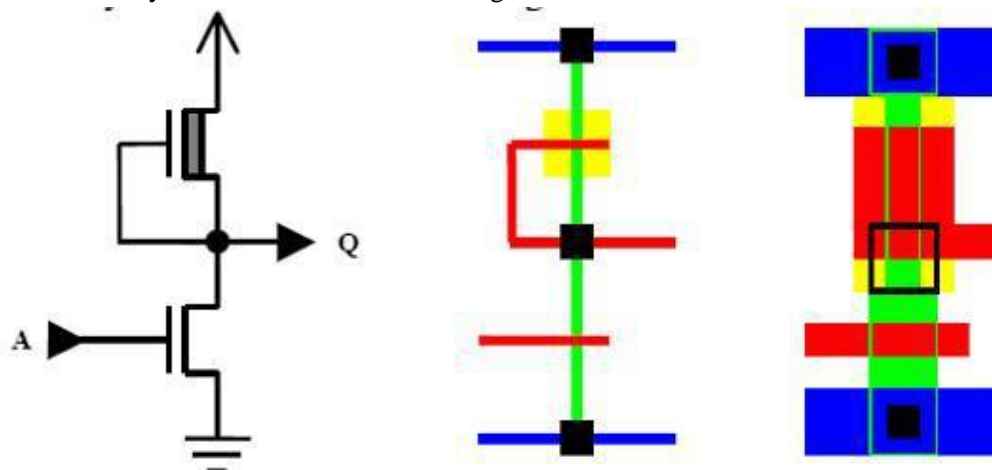


Figure 4: nMOS depletion load inverter

Figure 4 shows schematic, stick diagram and corresponding layout of nMOS depletion Load inverter.

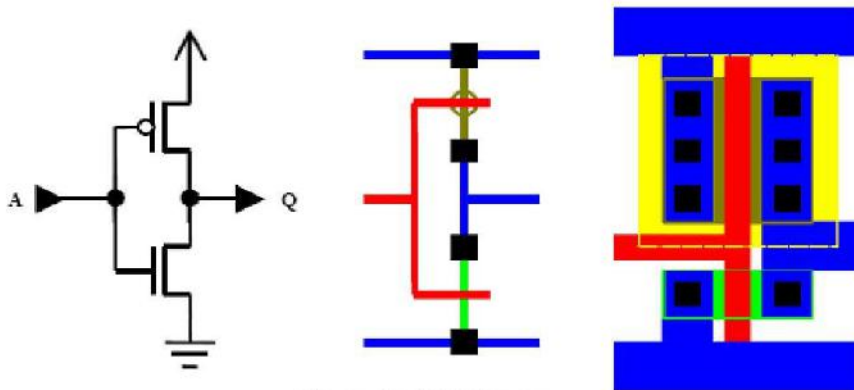


Figure 5: CMOS inverter

Figure 5 shows the schematic, stick diagram and corresponding layout of CMOS inverter

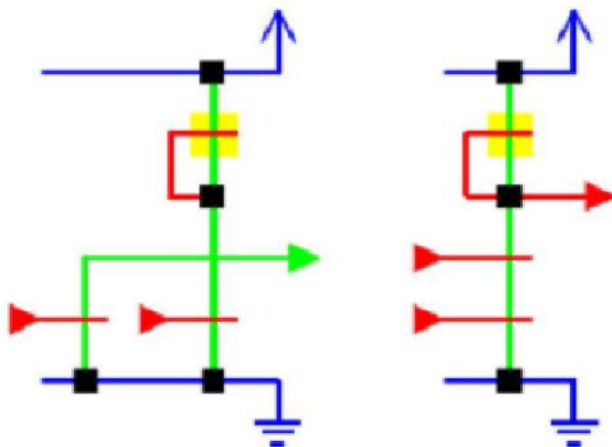


Figure 6: nMOS depletion load NAND and NOR stick diagram

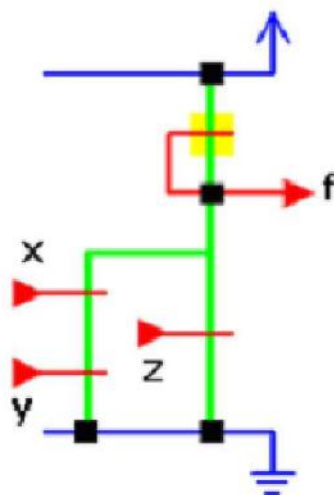


Figure 7: stick diagram of a given function f.

Figure 7 shows the stick diagram nMOS implementation of the function $f = [(xy) + z]$

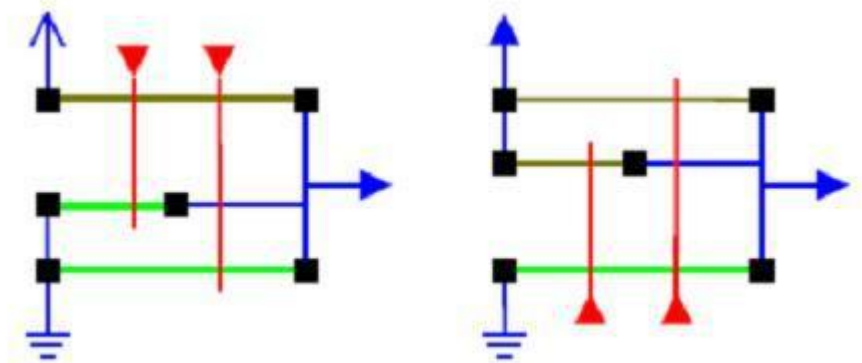


Figure 8: stick diagram of CMOS NAND and NOR

Figure 8 shows the stick diagram CMOS NOR and NAND, where we can see that the p diffusion line never touched the n diffusion directly, it is always joined using a blue color metal line.

NMOS and CMOS Design style:

In the NMOS style of representing the sticks for the circuit, we use only NMOS transistor, in CMOS we need to differentiate n and p transistor, that is usually by the color or in monochrome diagrams we will have a demarcation line. Above the demarcation line are the p transistors and below the demarcation line are the n transistors.

Following stick shows CMOS circuit example in monochrome where we utilize the demarcation line.

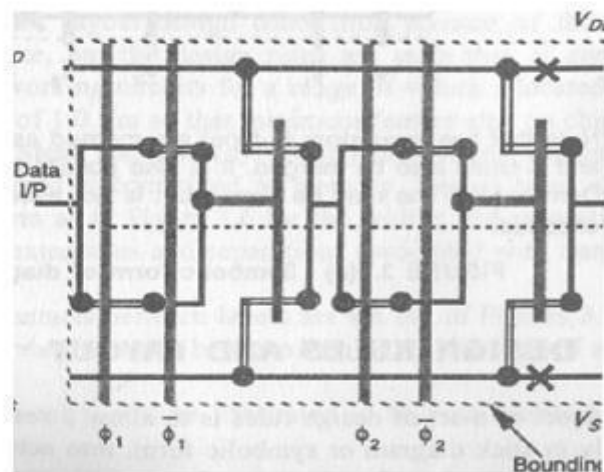


Figure 9: stick diagram of dynamic shift register in CMOS style

Figure 9 shows the stick diagram of dynamic shift register using CMOS style. Here the output of the TG is connected as the input to the inverter and the same chain continues depending the number of bits.

Design Rules:

Design rules include width rules and spacing rules. Mead and Conway developed a set of simplified scalable λ -based design rules, which are valid for a range of fabrication technologies. In these rules, the minimum feature size of a technology is characterized as 2λ . All width and spacing rules are specified in terms of the parameter λ . Suppose we have design rules that call for a minimum width of 2λ , and a minimum spacing of 3λ . If

we select a 2 μm technology (i.e., $\lambda = 1\ \mu\text{m}$), the above rules are translated to a minimum width of 2 μm and a minimum spacing of 3 μm . On the other hand, if a 1 μm technology (i.e., $\lambda = 0.5\ \mu\text{m}$) is selected, then the same width and spacing rules are now specified as 1 μm and 1.5 μm , respectively.

Figure 10 shows the design rule n diffusion, p diffusion, poly, metal1 and metal 2. The n and p diffusion lines is having a minimum width of 2λ and a minimum spacing of 3λ . Similarly we are showing for other layers

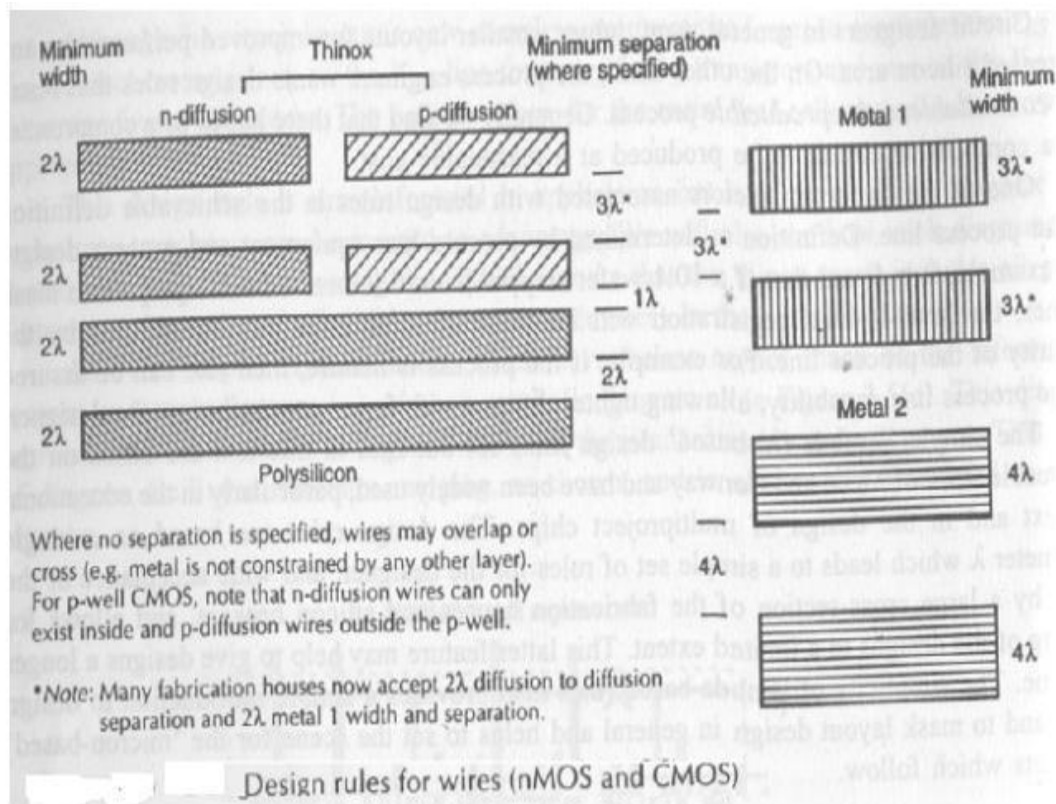


Figure 10: Design rules for the diffusion layers and metal layers

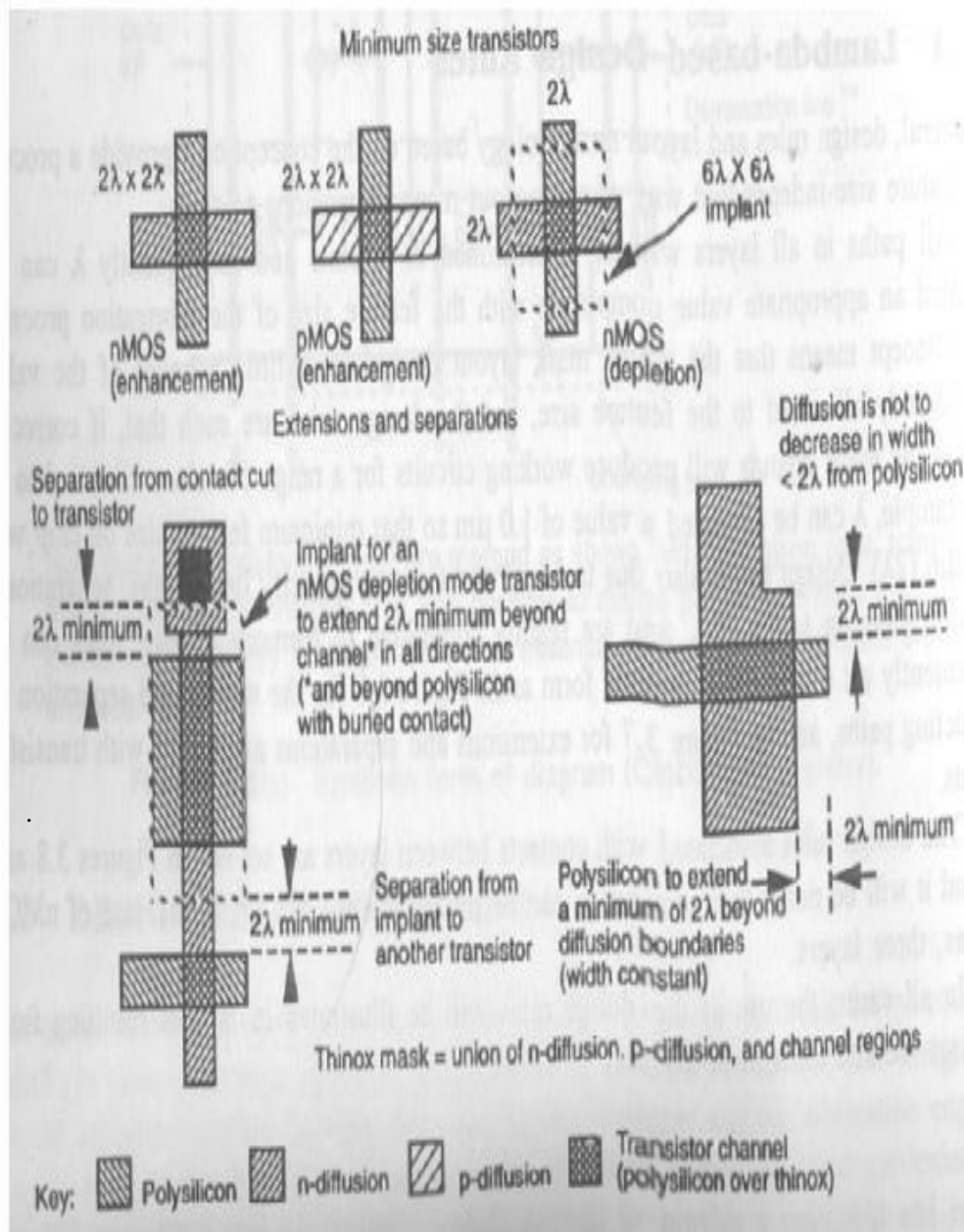


Figure 11: Design rules for transistors and gate over hang distance

Figure shows the design rule for the transistor, and it also shows that the poly should extend for a minimum of 2λ beyond the diffusion boundaries. (gate over hang distance)

What is Via?

It is used to connect higher level metals from metal1 connection. The cross section and layout view given figure 13 explain via in a better way

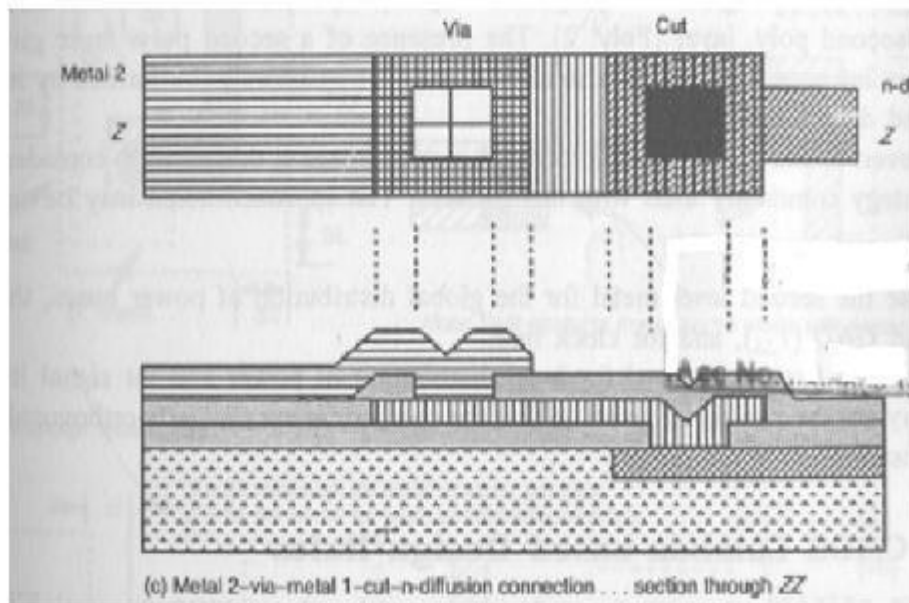


Figure 12: cross section showing the contact cut and via

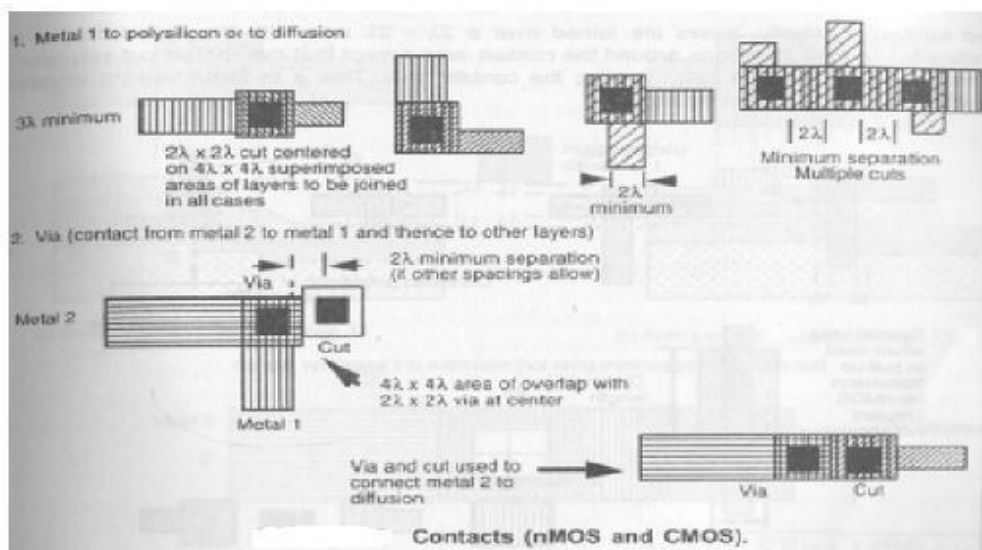


Figure 13: Design rules for contact cuts and vias

Figure shows the design rules for contact cuts and Vias. The design rule for contact is minimum $2\lambda \times 2\lambda$ and same is applicable for a Via.

Buried contact: The contact cut is made down each layer to be joined and it is shown in figure 1

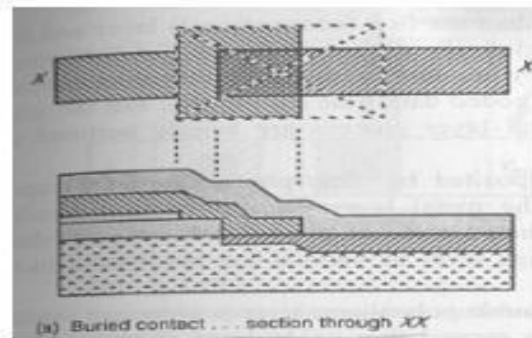


Figure 14: Buried contact

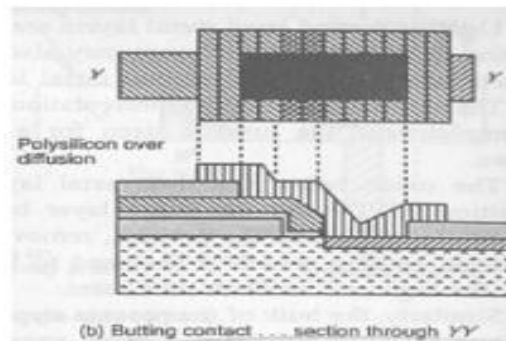


Figure 15: Butting contact

Butting contact: The layers are butted together in such a way the two contact cuts become contiguous. We can better understand this with the diagram.

CMOS LAMBDA BASED DESIGN RULES:

Till now we have studied the design rules wrt only NMOS, what are the rules to be followed if we have the both p and n transistor on the same chip will be made clear with the diagram. Figure 16 shows the rules to be followed in CMOS well processes to accommodate both n and p transistors.

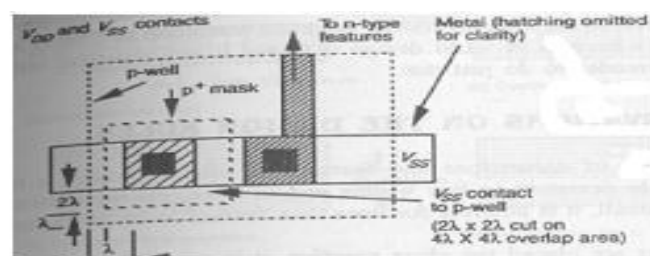


Figure 16: CMOS design rules

Orbit 2µm CMOS process:

In this process all the spacing between each layers and dimensions will be in terms micrometer. The 2µm here represents the feature size. All the design rules what ever we have seen will not have lambda instead it will have the actual dimension in micrometer.

In one way lambda based design rules are better compared micrometer based design rules, that is lambda based rules are feature size independent.

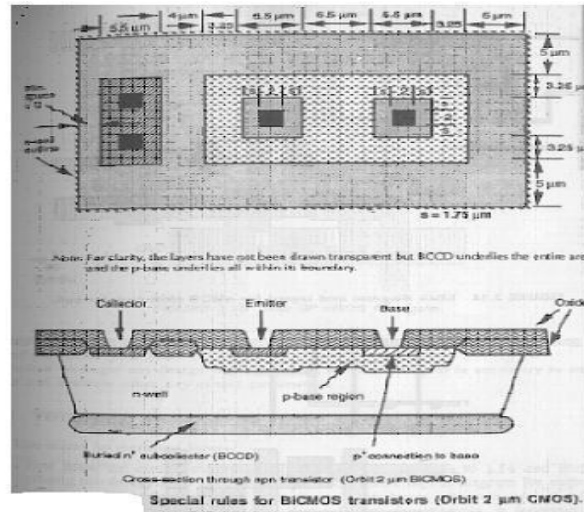


Figure 17: BiCMOS design rules

Figure 17 shows the design rule for BiCMOS process using orbit 2µm process.

The following is the example stick and layout for 2way selector with enable (2:1 MUX)

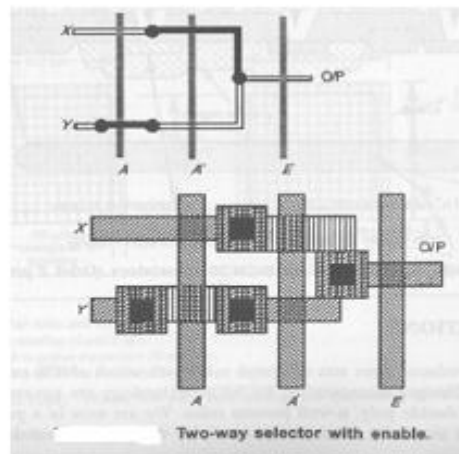


Figure 18: Two way selector stick and layout

Scaling of MOS Circuits:

Why Scaling?...

Scale the devices and wires down, Make the chips 'fatter' – functionality, intelligence, memory – and – faster, Make more chips per wafer – increased yield, Make the end user Happy by giving more for less and therefore, make MORE MONEY!!

Impact of scaling is characterized in terms of several indicators:

- Minimum feature size
- Number of gates on one chip
- Power dissipation
- Maximum operational frequency
- Die size
- Production cost

Many of the FoMs can be improved by shrinking the dimensions of transistors and interconnections. Shrinking the separation between features – transistors and wires Adjusting doping levels and supply voltages.

Technology Scaling :

- Goals of scaling the dimensions by 30%:
- Reduce gate delay by 30% (increase operating frequency by 43%) Double transistor density
- Reduce energy per transition by 65% (50% power savings @ 43% increase in frequency) Die size used to increase by 14% per generation
- Technology generation spans 2-3 years

Figure1 to Figure 5 illustrates the technology scaling in terms of minimum feature size, transistor count, propagation delay, power dissipation and density and technology generations.

Scaling Models

- Full Scaling (Constant Electrical Field)
- Ideal model – dimensions and voltage scale together by the same scale factor Fixed Voltage Scaling
- Most common model until recently – only the dimensions scale, voltages remain constant General Scaling

- Most realistic for today's situation – voltages and dimensions scale with different factors

Scaling Factors for Device Parameters

Device scaling modeled in terms of generic scaling factors: $1/\alpha$ and $1/\beta$

- $1/\beta$: scaling factor for supply voltage V_{DD} , and gate oxide thickness D
- $1/\alpha$: linear dimensions both horizontal and vertical dimensions

Why is the scaling factor for gate oxide thickness different from other linear horizontal and vertical dimensions? Consider the cross section of the device as in Figure 6, various parameters derived are as follows.

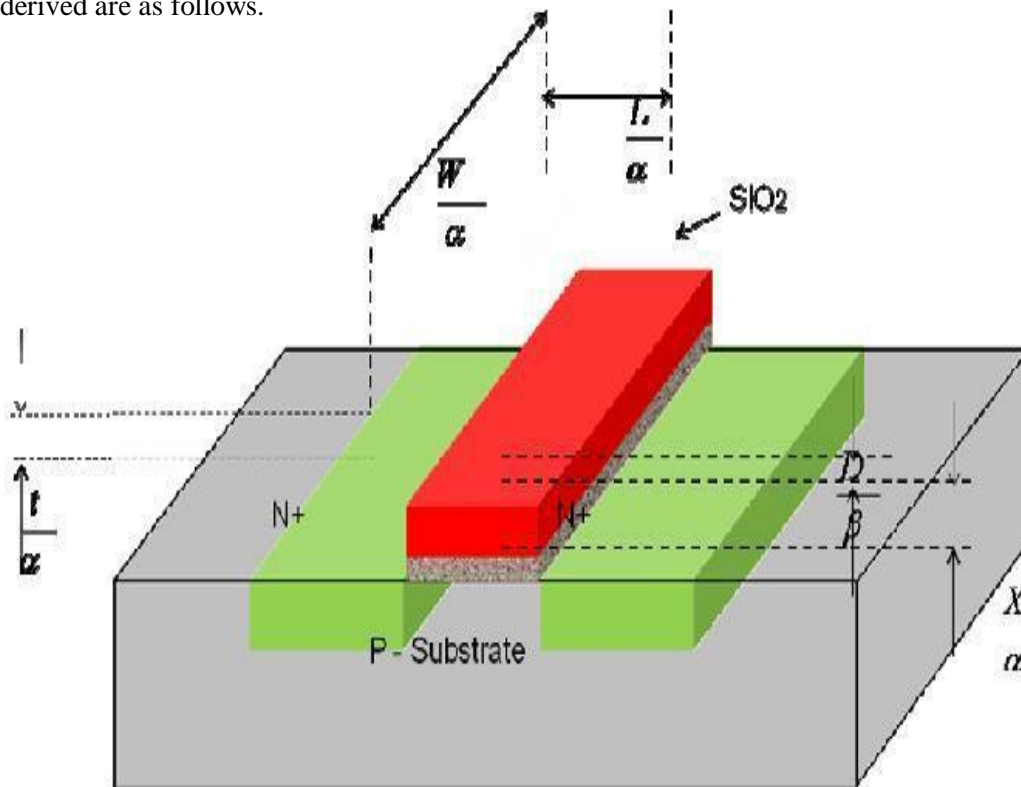


Figure-6: Technology generation

- Gate area A_g

$$A_g = L * W$$

Where L: Channel length and W: Channel width and both are scaled by $1/\alpha$

Thus A_g is scaled up by $1/\alpha^2$

- Gate capacitance per unit area C_o or C_{ox}

$$C_{ox} = \epsilon_{ox}/D$$

Where ϵ_{ox} is permittivity of gate oxide (thin-ox) = $\epsilon_{ins}\epsilon_0$ and D is the gate oxide thickness scaled by $1/\beta$

$$\text{Thus } C_{ox} \text{ is scaled up by } \left(\frac{1}{\beta}\right) = \beta$$

- Gate capacitance C_g $C_g = C_o * L * W$

Thus C_g is scaled up by $\beta * 1/\alpha^2 = \beta/\alpha^2$

- Parasitic capacitance C_x

C_x is proportional to A_x/d

where d is the depletion width around source or drain and scaled by $1/\alpha$

A_x is the area of the depletion region around source or drain, scaled by $(1/\alpha^2)$.

Thus C_x is scaled up by $\{1/(1/\alpha)\} * (1/\alpha^2) = 1/\alpha$

- Carrier density in channel Q_{on}

$$Q_{on} = C_o * V_{gs}$$

where Q_{on} is the average charge per unit area in the 'on' state.

C_o is scaled by β and V_{gs} is scaled by $1/\beta$

Thus Q_{on} is scaled by 1

- Channel Resistance R_{on}

$$R_{on} = \frac{L}{W} * \frac{1}{Q_{on} * \mu}$$

Where μ = channel carrier mobility and assumed constant

Thus R_{on} is scaled by 1.

- Gate delay T_d

T_d is proportional to $R_{on} * C_g$

$$T_d \text{ is scaled by } \frac{1}{\alpha^2} * \beta = \frac{\beta}{\alpha^2}$$

- Maximum operating frequency f_o

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

f_o is inversely proportional to delay T_d and is scaled by

$$\beta * \left(\frac{1}{\beta^2} \right) = \frac{1}{\beta}$$

- Saturation current I_{dss}

$$I_{dss} = \frac{C_o \mu}{2} * \frac{W}{L} * (V_{gs} - V_t)^2$$

Both V_{gs} and V_t are scaled by $(1/\beta)$. Therefore, I_{dss} is scaled by $\frac{1}{\left(\frac{\beta}{\alpha^2}\right)} = \frac{\alpha^2}{\beta}$

- Current density J

Current density, $J = \frac{I_{dss}}{A}$ where A is cross sectional area of the Channel in the "on" state which is scaled by $(1/\alpha^2)$.

So, J is scaled by

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

- Switching energy per gate E_g

$$E_g = \frac{1}{2} C_g V_{DD}^2$$

So E_g is scaled by

$$\frac{\beta}{\alpha^2} * \left(\frac{1}{\beta^2} \right) = \frac{1}{\alpha^2 \beta}$$

- Power dissipation per gate P_g

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

P_g comprises of two components: static component P_{gs} and dynamic component P_{gd} :

Where, the static power component is given by: $P_{gs} = \frac{V_{DD}^2}{R_{on}}$

And the dynamic component by: $P_{gd} = E_g f_o$

Since V_{DD} scales by $(1/\beta)$ and R_{on} scales by 1, P_{gs} scales by $(1/\beta^2)$.

Since E_g scales by $(1/\alpha^2 \beta)$ and f_o by (α_2/β) , P_{gd} also scales by $(1/\beta^2)$. Therefore, P_g scales by $(1/\beta^2)$.

- Power dissipation per unit area P_a

$$P_a = \frac{P_g}{A_g} = \frac{\left(\frac{1}{\beta^2}\right)}{\left(\frac{1}{\alpha^2}\right)} = \frac{\alpha^2}{\beta^2}$$

- Power – speed product P_T

$$P_T = P_g * T_d = \frac{1}{\beta^2} \left(\frac{\beta}{\alpha^2}\right) = \frac{1}{\alpha^2 \beta}$$

Limitations of Scaling:

Effects, as a result of scaling down- which eventually become severe enough to prevent further miniaturization.

- Substrate doping
- Depletion width
- Limits of miniaturization
- Limits of interconnect and contact resistance
- Limits due to sub threshold currents
- Limits on logic levels and supply voltage due to noise
- Limits due to current density

GATE LEVEL DESIGN

Introduction:

The module (integrated circuit) is implemented in terms of logic gates and interconnections between these gates. Designer should know the gate-level diagram of the design. In general, gate-level modeling is used for implementing lowest level modules in a design like, full-adder, multiplexers, etc.

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. Now the goal of logic design or optimization is to find a network of logic gates that together compute the combinational logic function we want.

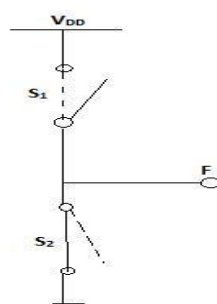
For example, given the expression $a+b$, we can compute its truth value for any given values of a and b , and also we can evaluate relationships such as $a+b = c$. but logic design is difficult for many reasons:

- We may not have a logic gate for every possible function, or even for every function of n inputs.
- Not all gate networks that compute a given function are alike—networks may differ greatly in their area and speed.
- Thus combinational logic expressions are the specification, Logic gate networks are the implementation, Area, delay, and power are the costs.
- 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), fluidic logic, pneumatic logic, optics, molecules, or even mechanical elements.
- With amplification, logic gates can be cascaded in the same way that Boolean functions can be composed, allowing the construction of a physical model of all of Boolean logic.
- simplest form of electronic logic is diode logic. This allows AND and OR gates to be built, but not inverters, and so is an incomplete form of logic. Further, without some kind of amplification it is not possible to have such basic logic operations cascaded as required for more complex logic functions.
- To build a functionally complete logic system, relays, valves (vacuum tubes), or transistors can be used.
- 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. These gates were used in early integrated circuits. For higher speed, 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, using only half the space.
- In virtually every type of contemporary chip implementation of digital systems, the bipolar transistors have been replaced by 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.

CMOS LOGIC GATES AND OTHER COMPLEX GATES:

General logic circuit Any Boolean logic function (F) has two possible values, either logic 0 or logic 1. For some of the input combinations, $F = 1$ and for all other input combinations, $F = 0$. So in general, any

Boolean logic function can be realized using a structure as shown in figure.

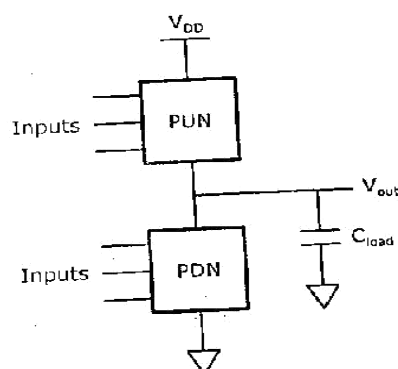


- The switch S_1 is closed and switch S_2 is open for input combinations that produces $F = 1$.
- The switch S_1 is open and switch S_2 is closed for input combinations that produces $F = 1$.
- The switch S_1 is open and switch S_2 is open for input combinations that produces $F = 0$.

Thus the output (F) is either connected to V_{DD} or the ground, where the logic 0 is represented by the ground and the logic 1 is represented by V_{DD} . So the requirement of digital logic design is to implement the pull-up switch(S_1) and the pull-down switch(S_2).

CMOS static logic

A generalized CMOS logic circuit consists of two transistor nets NMOS and MOS. The NMOS transistor net is connected between the power supply and the logic gate output called as pull-up network , Whereas the NMOS transistor net is connected between the output and ground called as pull-down network. Depending on the applied input logic, the PUN connects the output node to V_{DD} and PDN connects the output node to the ground.



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

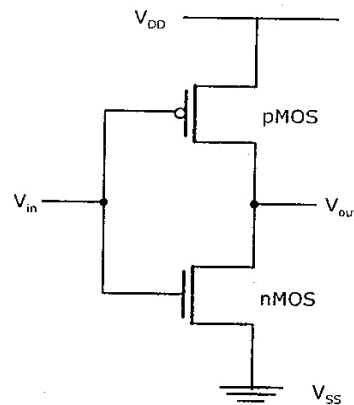
- Design the pull down network (PDN) by realizing,
 - AND(product) terms using series-connected NMOSFETs.
 - OR (sum) terms using parallel-connected NMOSFETS.
- Design the pull-up network by realizing,

AND(product) terms using parallel-connected NMOSFETs. OR (sum) terms using series connected NMOSFETs.

- Add an inverter to the output to complement the function. Some functions are inherently negated, such as NAND,NOR gates do not need an inverter at the output terminal.

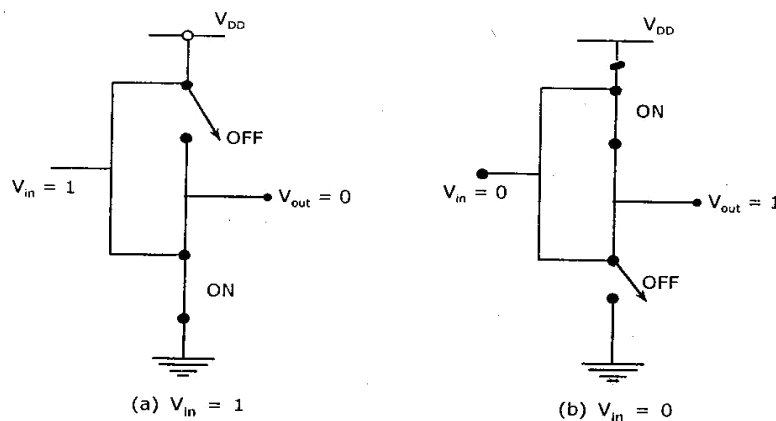
CMOS inverter:

A CMOS inverter is the simplest logic circuit that uses one PMOS and one NMOS transistor. The NMOS is used in PDN and the PMOS is used in the PUN as shown in figure.



Working operation:

- 1) When the input V_{in} is logic HIGH, then the NMOS transistor is ON and the PMOS transistor is OFF. Thus the output Y is pulled down to ground (logic 0) since it is connected to ground but not to source V_{DD} .
- 2) When the input V_{in} is logic LOW, then NMOS transistor is OFF and the PMOS transistor is ON, Thus the output Y is pulled up to V_{DD} (logic 1) since it is connected to source via PMOS but not to ground.



CMOS NAND gate:

The two input NAND function is expressed by $Y=A.B$

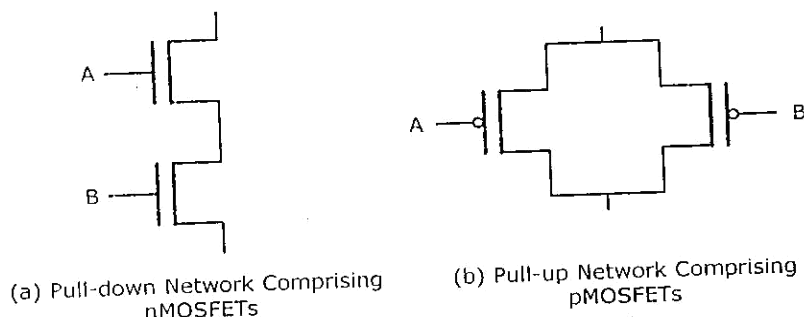
Step 1 Take complement of Y $Y=$

$$A.B = A.B$$

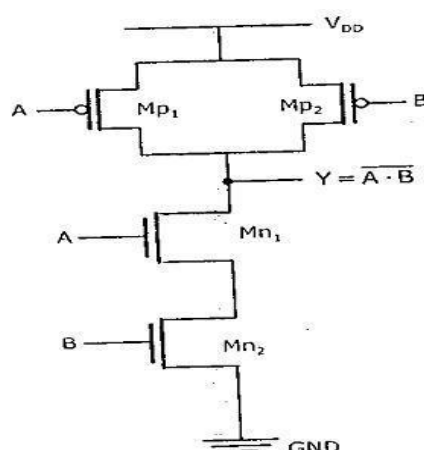
Step 2 Design the PDN

In this case, there is only one AND term, so there will be two nMOSFETs in series as shown in figure.

Step 3 Design the PUN. In PUN there will be two pMOSFETs in parallel, as shown in figure



Finally join the PUN and PDN as shown in figure which realizes two –input NAND gate. Note that we have realized y , rather than Y because the inversion is automatically provided by the nature of the CMOS circuit operation,



Working operation

- 1) Whenever at least one of the inputs is LOW, the corresponding MOS transistor will conduct while the corresponding NMOS transistor will turn OFF. Subsequently, the output voltage will be HIGH.
- 2) Conversely, if both inputs are simultaneously HIGH, then both PMOS transistors will turn OFF, and the output voltage will be pulled LOW by the two conducting PMOS transistors.

CMOS NOR gate

The two input NOR function is expressed by $Y=A+B$

Step 1 Take complement of Y

$$Y = A+B = \overline{\overline{A+B}}$$

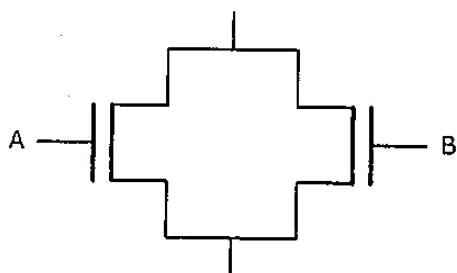
Step 2 Design the PDN

In this case, there is only one OR term, so there will be two nMOSFETs connected in parallel, as shown in

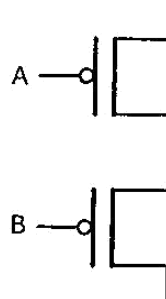
figure.

Step 3 Design the PUN

In PUN there will be two pMOSFETs in series, as shown in figure



(a) Pull-down Network Comprising nMOSFETs

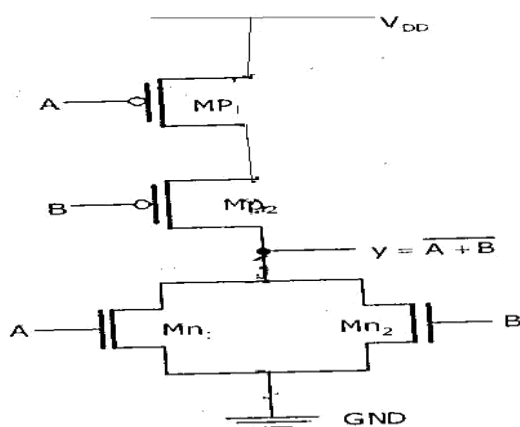


(b) Pull-up Network Comprising pMOSFETs

Finally join the PUN and PDN as shown in figure which realizes two –input NAND gate. Note that we have realized y , rather than Y because the inversion is automatically provided by the nature of the CMOS circuit operation,

Working operation:

- 1) Whenever at least one of the inputs is LOW, the corresponding PMOS transistor will conduct while the corresponding NMOS transistor will turn OFF. Subsequently, the output voltage will be HIGH.
- 2) Conversely, if both inputs are simultaneously HIGH, then both PMOS transistors will turn OFF, and the output voltage will be pulled LOW by the two conducting NMOS transistors.



COMPLEX GATES IN CMOS LOGIC:

A complex logic gate is one that implements a function that can provide the basic NOT, AND and OR operation but integrates them into a single circuit. CMOS is ideally suited for creating gates that have logic equations by exhibiting the following,

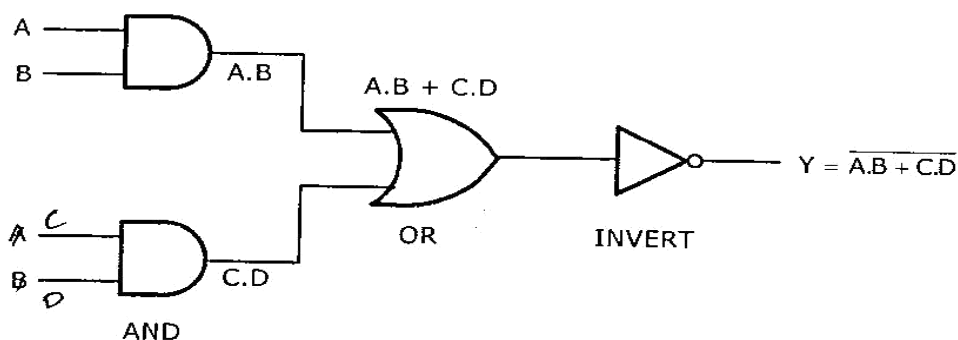
- 1) AND-OR-INVERT - AOI form
- 2) OR-AND-INVERT - OAI form

An AOI logic equation is equivalent to a complemented SOP form, while an OAI equation is equivalent to a complemented POS structure. In CMOS, output always produces NOT operation acting on input variable.

1) AOI Logic Function (OR) Design of XOR gate using CMOS logic:

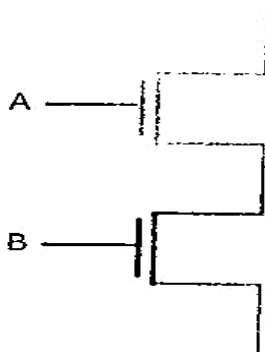
AND-OR-INVERT logic function(AOI) implements operation in the order AND,OR,NOT. For example ,

let us consider the function $Y = \overline{AB + CD}$ i.e., $Y = \text{NOT}((A \text{ AND } B) \text{ OR } (C \text{ AND } D))$ The AOI logic gate implementation for Y

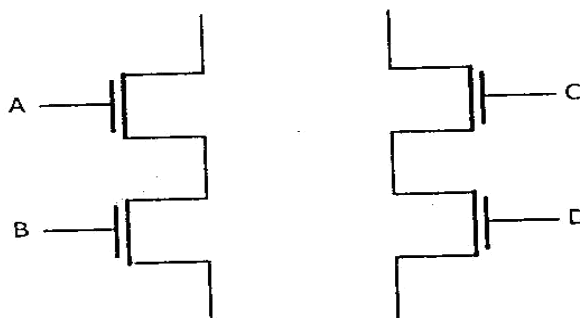


CMOS implementation for Y:

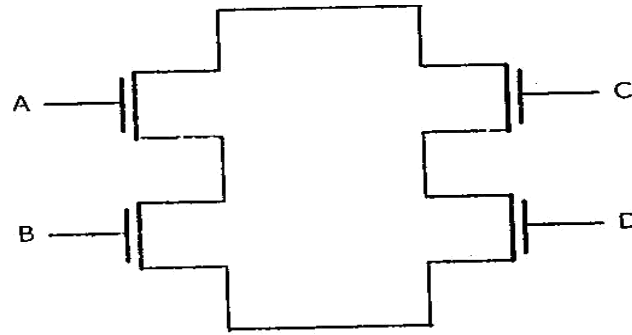
Step 1: Draw A.B (AND) function first by connecting 2 nMOS transistors in series.



Step 2: Draw C.D implementation, by using 2 nMOS transistors in series.

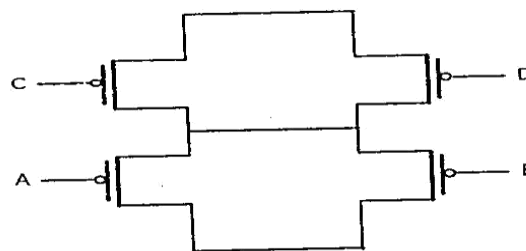


Step 3: $Y = A.B + C.D$, In this function A.B and C.D are added, for addition, we have to draw parallel connection. So, A.B series connected in parallel with C.D as shown in figure.

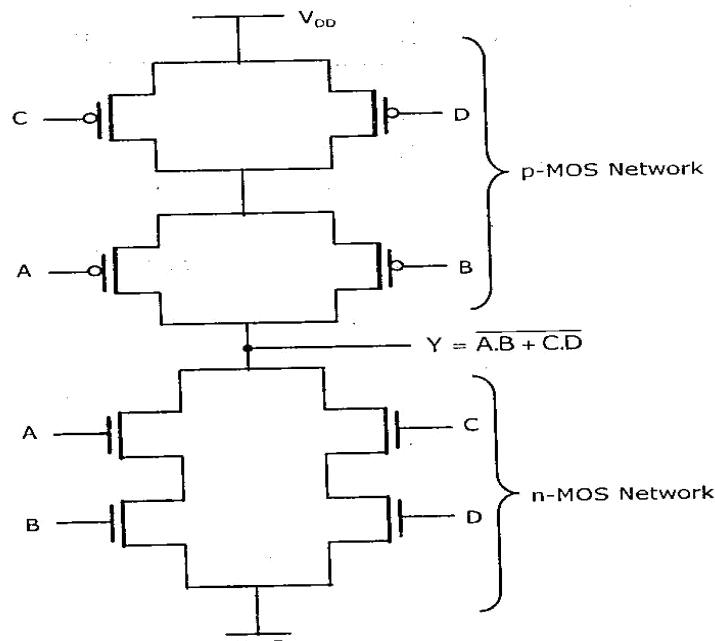


Step 4: Draw pMOS connection,

- I. In nMOS A,B connected in series. So, in pMOS side, A.B should be connected in parallel.
- II. In nMOS C,D connected in series. So, in pMOS side, C.D should be connected in parallel.
- III. A.B and C.D networks are connected in parallel in nMOS side. So, in pMOS side, A.B and C.D networks should be connected in series.
- IV. In pMOS multiplication should be drawn in parallel, then addition should be drawn in series as shown in figure.



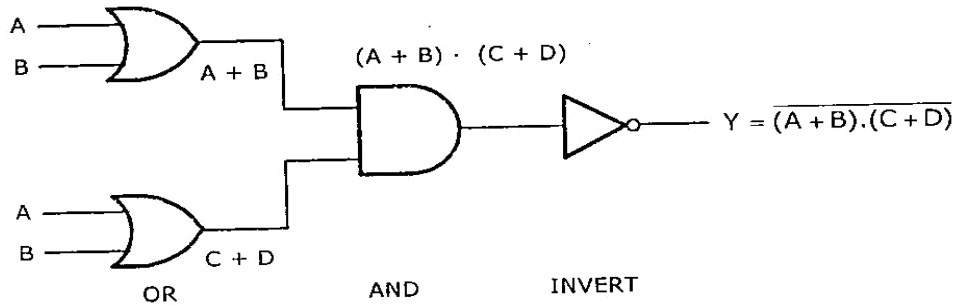
Step 5: Take output at the point in between NMOS and PMOS networks.



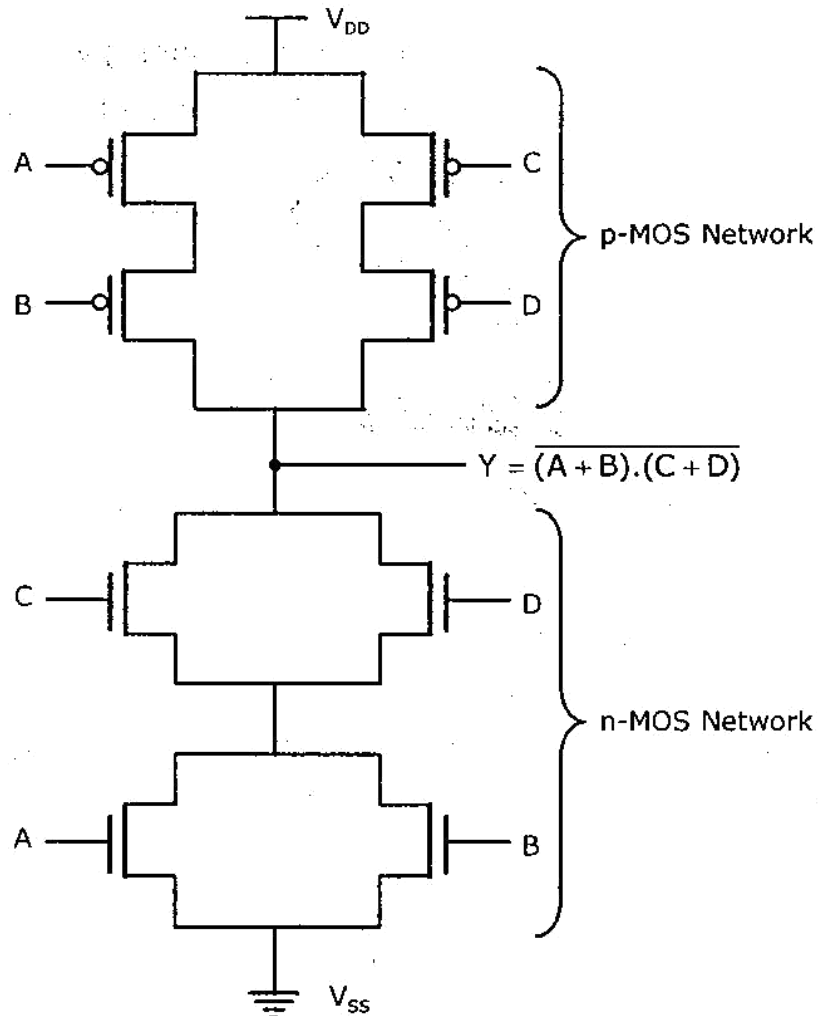
2) OAI Logic Function (OR) Design of XNOR gate using CMOS logic:

OR-AND-INVERT logic function(AOI) implements operation in the order OR,AND,NOT. For example ,

let us consider the function $Y = (A+B).(C+D)$ i.e., $Y = \overline{\overline{(A + B) \cdot (C + D)}}$ The OAI logic gate implementation for Y



CMOS implementation for Y:



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 in path through each switch is isolated from the logic levels activating the switch.

PASS TRANSISTOR:

- 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).
- If a single transistor is a switch between two nodes, then voltage degradation. equal to V_t (threshold voltage) for high or low level depends up on NMOS or PMOS logic.
- When using NMOS switch logic no pass transistor gate input may be driven through one or more pass transistors as shown in figure.
- Since the signal out of pass transistor T1 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 T2 to reach an acceptable logic 1 level.

Advantages:

They have topological simplicity.

- Requires minimum geometry.
- Do not dissipate standby power, since they do not have a path from supply to ground.

Disadvantages:

- Degradation in the voltage levels due to undesirable threshold voltage effects.
- Never drive a pass transistor with the output of another pass transistor.

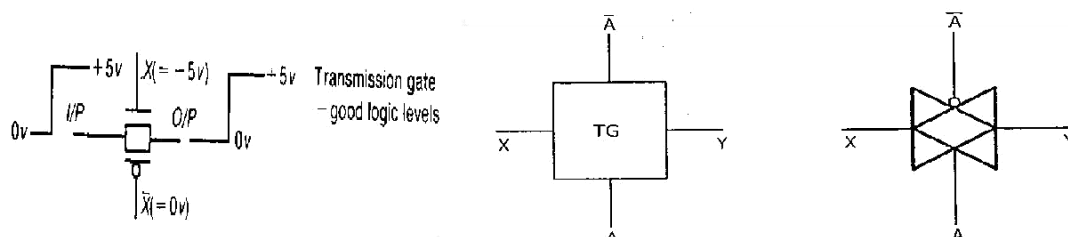
TRANSMISSION GATE:

1. It is an electronic element, good non-mechanical relay built with CMOS technology.
2. It is made by parallel combination of an nMOS and pMOS transistors with the input at gate of one transistor being complementary to the input at the gate of the other as shown in figure.
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
5. the input and output is either low resistance or high-resistance, respectively $R_{on} = 100\Omega$ and $R_{off} > 5 M\Omega$.

Operation:

- When the gate input to the NMOS transistor is '0' and the complementary '1' is gate input to the PMOS, thus both are turned off.
- When gate input to the NMOS is '1' and its complementary '0' is the gate input to the PMOS, both are turned on and passes any signal '1' and '0' equally without any degradation.

- The use of transmission gates eliminates the undesirable threshold voltage effects which give rise to loss of logic levels in pass-transistors as shown in figure.



Advantages:

- Transmission gates eliminates the signal degradation in the output logic levels.
- Transmission gate consists of two transistors in parallel and except near the positive and negative rails.

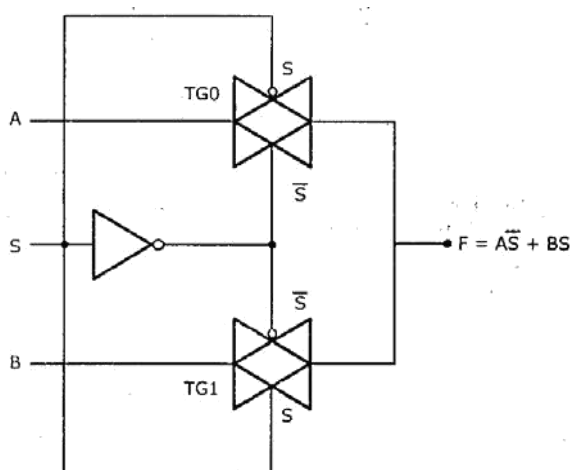
Disadvantages:

- Transmission gate requires more area than NMOS pass circuitry.
- Transmission gate requires complemented control signals.

“ Transmission gate logic can be used to design multiplexers(selector functions)”.

DESIGN A 2-INPUT MULTIPLEXER USING CMOS TRANSMISSION GATES:

Figure shows a 2-input multiplexer circuit using CMOS transmission gate.



If the control input S is low, the TG0 conducts and the output F is equal to A . On the other hand, if the control input S is high the TG1 conducts and the output F is equal to B .

ALTERNATIVE GATE CIRCUITS:

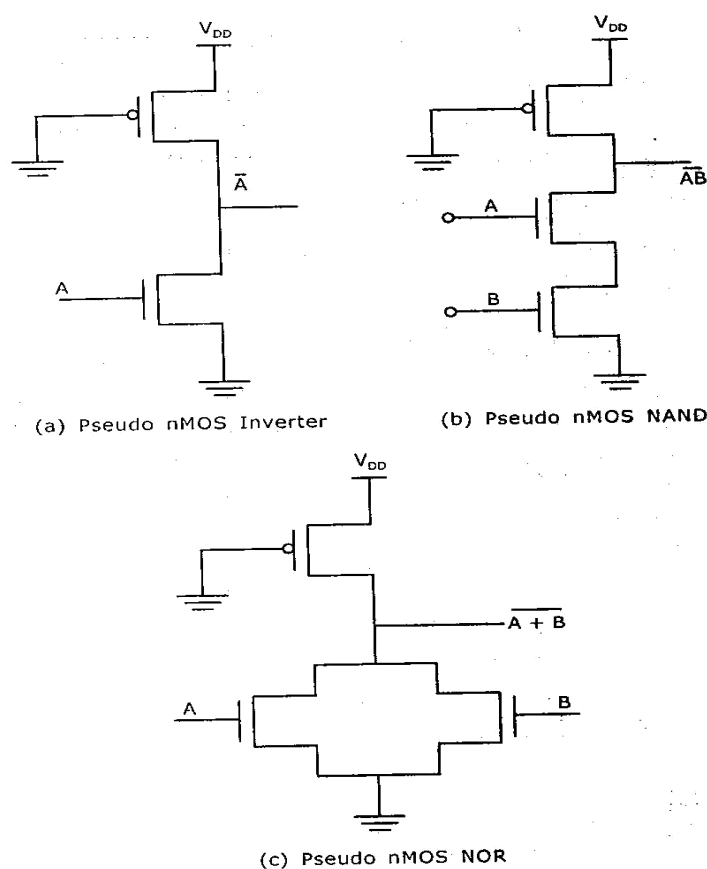
CMOS suffers from increased area and correspondingly increased capacitance and delay, as the logic gates become more complex. For this reason, designers developed circuits (Alternate gate circuits) that can be used to supplement the complementary type circuits. These forms are not intended to replace CMOS but rather to be used in special applications for special purposes.

1. PSEUDO NMOS Logic:

Pseudo NMOS logic is one type of alternate gate circuit that is used as a supplement for the complementary MOS logic circuits. In the pseudo-NMOS logic, the pull up network (PUN) is realized by a single PMOS transistor. The gate terminal of the 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. Figure shows the general building block of logic circuits that follows pseudo NMOS logic.

Here, only the NMOS logic (Q_n) is driven by the input 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-NMOS gate circuit is identical to the pull-down network (PDN) of the complementary CMOS gate.

The realization of logic circuits using pseudo-NMOS logic is as shown in figure.



Advantages:

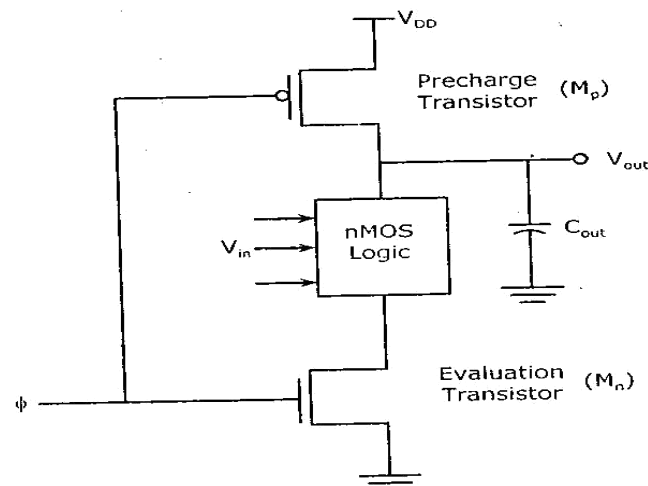
- 1) Uses less number of transistors as compared to CMOS logic.
- 2) Geometrical area and delay gets reduced as it requires less transistors.
- 3) Low power dissipation.

Disadvantages:

- 1) The main drawback of using a 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} .
- 2) Layout problems are critical.

DYNAMIC CMOS LOGIC

A dynamic CMOS logic uses charge storage and clocking properties of MOS transistors to implement logic operations. Figure shows the basic 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.

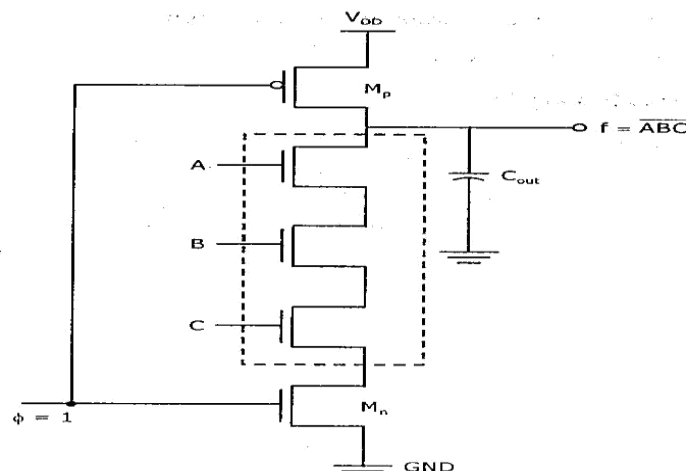


The gate (clock ϕ) defines two phases, evaluation and precharge phase during each clock cycle.

Working:

- When clock $\phi = 0$ the circuit is in precharge phase with the pMOS device M_p ON and the evaluation nMOS M_n OFF. This establishes a conducting path between V_{DD} and the output allowing C_{out} to charge to a voltage $V_{out} = V_{DD}$. M_p is often called the precharge FET.
- When clock $\phi = 1$ the circuit is in evaluation phase with the pMOS device M_p OFF and the evaluation nMOS M_n ON. If the logic block acts like a closed switch the C_{out} can discharge through logic array and M_n , this gives a final result of $V_{out} = V_{DD}$, logically this is an output of $F = 1$. Charge leakage eventually drops the output to $V_{out} = 0$ V which could be an incorrect logic value.

The logic formation is formed by three series connected FETs (3-input NAND gate) is shown in figure.



The dynamic CMOS logic circuit has a serious problem when they are cascaded. In the precharged phase ($\phi = 0$), output of all the stages are pre-charged to logic high. In the evaluation phase ($\phi = 1$), the output of all stages are evaluated simultaneously. Suppose in the first stage, the inputs are such that the

output is logic low after the evaluation. In the second stage, the output of the first stage is one input and there are other inputs. If the other inputs of the second stage are such that output of it discharges to logic low, then the evaluated output of the first stage can never make the output of the second stage logic high. This is because, by the time the first stage is being evaluated, output of the second stage is discharged, since evaluation happens simultaneously. Remember that the output cannot be charged to logic high in the evaluation phase ($\phi = 1$, PMOSFET in PUN is OFF), it can only be retained in the logic high depending on the inputs.

Advantages

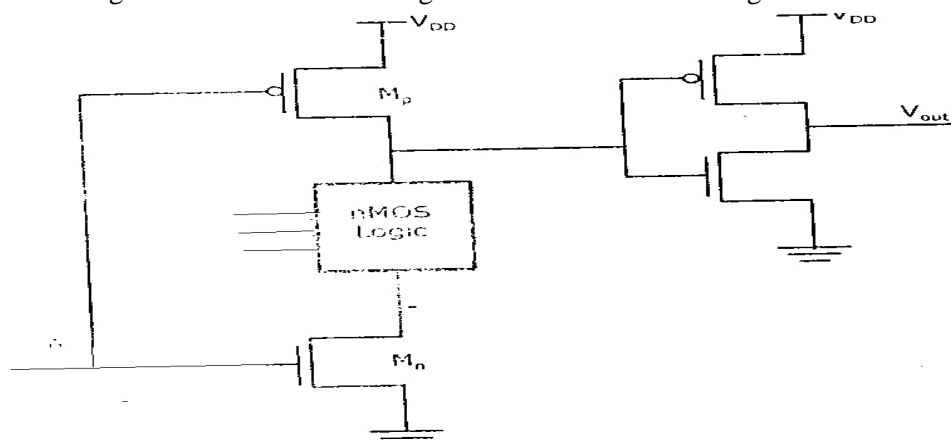
- 1) Low power dissipation.
- 2) Large noise margin.
- 3) Small area due to less number of transistors.\

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 circuit, 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.

The circuit diagram of domino CMOS logic structures as shown in figure as follows



A domino CMOS AND-OR gate that realizes the function $y = AB + CD$ is depicted in figure . The left hand part of the circuit containing M_n , M_p , T_1, T_2, T_3 , and T_4 forms an AND-OR-INVERTER (AOI) gate. It derives the static CMOS inverter formed by N_2 and P_2 in the right-hand part of the circuit. The domino gate is activated by the single phase clock ϕ applied to the NMOS (M_n) and the PMOS (M_p) transistors. The load on the AOI part of the circuit is the parasitic load capacitance.

Working:

- When $\phi = 0$, M_p is ON and M_n is OFF, so that no current flows in the AND-OR paths of the AOI. The capacitor C_L is charged to V_{DD} through M_p since the latter is ON. The input to the inverter is high, and drives the output voltage V_0 to logic-0.

- When $\phi = 1$, M_p is turned OFF and M_n is turned ON. If either (or both) A and B or C and D is at logic-1, C_L discharges through either T_2, T_1 and M_n or T_3, T_4 and M_p . So, the inverter input is driven to logic-0 and hence the output voltage V_0 to logic-1. The Boolean expression for the output voltage is $Y = AB + CD$.

Note : Logic input can change only when $\phi = 0$. No changes of the inputs are permitted when $\phi = 1$ since a discharge path may occur.

Advantages:

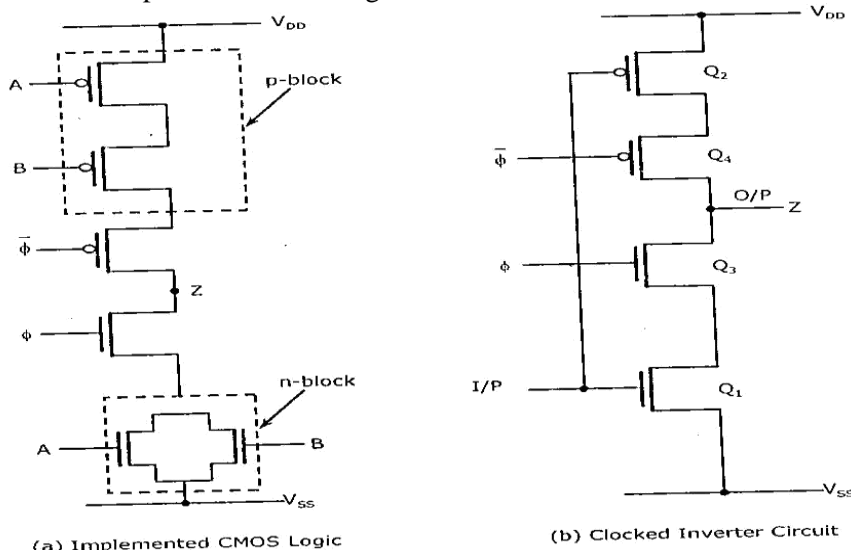
- 1) Smaller areas compared to conventional CMOS logic.
- 2) Parasitic capacitances are smaller so that higher operating speeds are possible.
- 3) Operation is free of glitches since each gate can make one transition.

Disadvantages:

- 1) Non inverting structures are possible because of the presence of inverting buffer.
- 2) Charge distribution may be a problem.

CLOCKED CMOS LOGIC:

The clocked CMOS logic is also referred as C^2 MOS logic. Figure shows the general arrangement of a clocked CMOS (C^2 MOS) logic. A pull-up p-block and a complementary n-block pull-down structure represent p and n-transistors respectively and 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 the clock. Figure shows a clocked inverter circuit which is also belongs to clocked CMOS logic family. The slower rise times and fall times can be expected due to owing of extra transistors in series with the output.

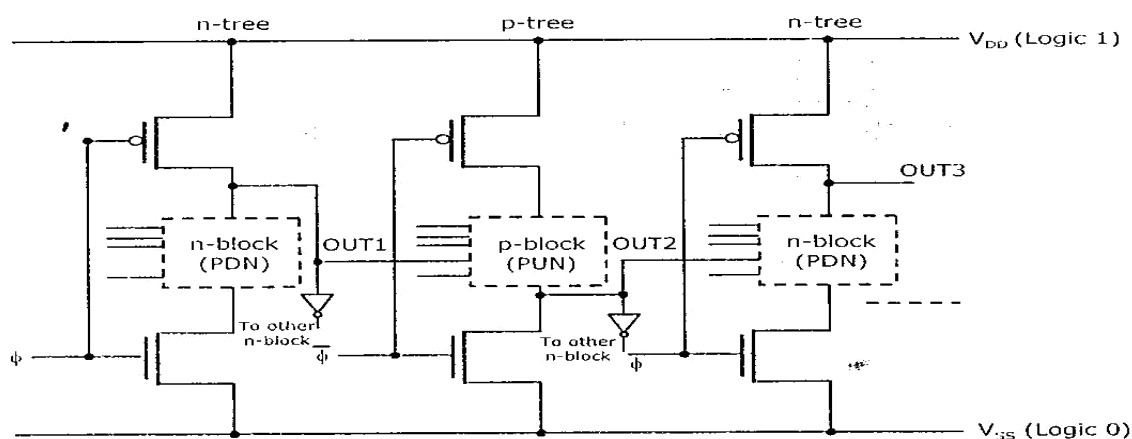


Working:

- When $\phi = 1$ the circuit acts an inverter, because transistors Q_3 and Q_4 are 'ON'. It is said to be in the "evaluation mode". Therefore the output Z changes its previous value.
- When $\phi = 0$ the circuit is in hold mode, because transistors Q_3 and Q_4 becomes 'OFF'. It is said to be in the "precharge mode". Therefore the output Z remains its previous value.

n-p CMOS LOGIC

Figure shows the another variation of basic dynamic logic arrangement of CMOS logic called as n-p CMOS logic. In this, logic the actual logic blocks are alternatively 'n' and 'p' in a cascaded structure. The clock ϕ and ϕ' are used alternatively to feed the precharge and evaluate transistors. However, the functions of top and bottom transistors are also alternate between precharge and evaluate transistors.



Working:

- During the pre charge phase $\phi = 0$, the output of the n-tree gate, OUT 1 OUT3, are charged to V_{DD} , while the output of the p-tree gate OUT2 is pre discharged to 0V. Since the n-tree gate connects PMOS pull-up devices, the PUN of the p-tree is turned off at that time.
- During the evaluation phase $\phi = 1$, the outputs (OUT1,OUT3) of the n-tree gate can only make a 1-0 transition, conditionally turning on some transistors in the p-tree. This ensures that no accidental discharge of OUT 2 can occur.
- Similarly n-tree blocks can follow p-tree gates without any problems, because the inputs to the n-gate are pre charged to 0.

Disadvantages:

Here, the p-tree blocks are slower than the n-tree modules, due to the lower current drive of the PMOS transistors in the logic network.

UNIT-IV

PHYSICAL DESIGN:

INTRODUCTION:

The transformation of a circuit description into a geometric description, is known as a layout. A layout consists of a set of planar geometric shapes in several layers.

The process of converting the specifications of an electrical circuit into a layout is called the Physical design.

Due to the large number of components and the fine details required by the fabrication process, the physical design is not practically possible without the help of computers. As a result, almost all phases of physical design extensively use computer-aided design (CAD) tools and many phases are either partially or fully automated. This automation of the physical design process has increased the level of integration, reduced the turnaround time, and enhanced chip performance.

There are various CAD tools available in market and each of them have their own strengths and weaknesses. The Electronic Design Automation (EDA) companies like Cadence, Synopsys, Magma, and Mentor Graphics provide these CAD tools.

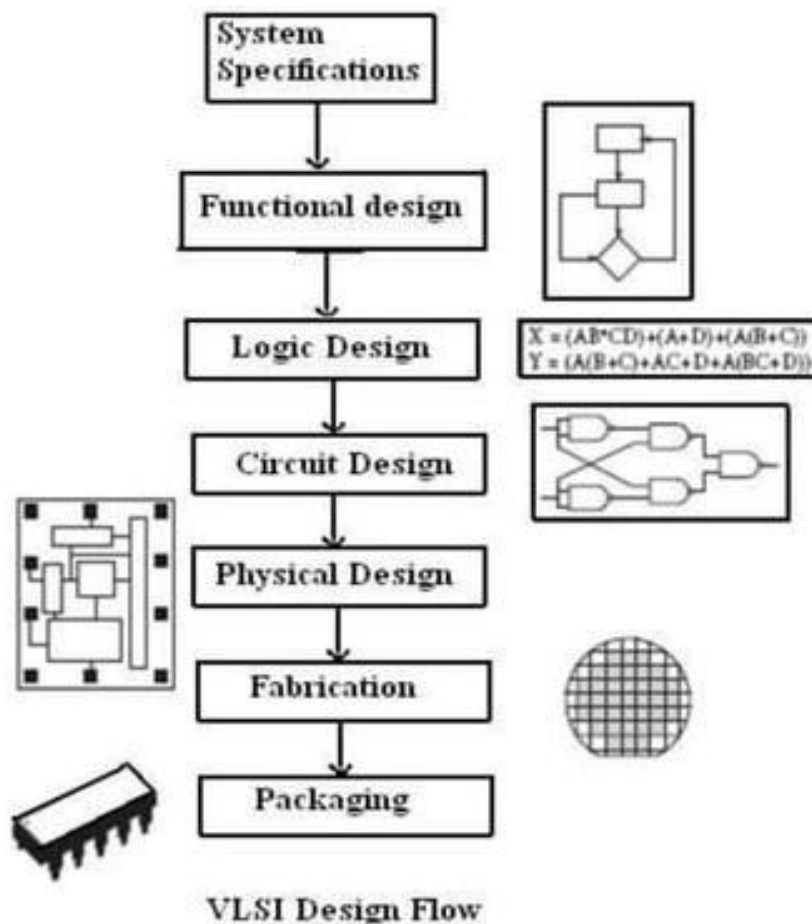
VLSI physical design automation is mainly deals with the study of algorithms related to the physical design process. The objective is to study optimal arrangements of devices on a plane (or in a three-dimensional space) and various interconnection schemes between these devices to obtain the desired functionality. Because space on a wafer is very expensive, algorithms must use the space very efficiently to decrease the costs and improve the yield. In addition, the arrangement of devices (placement) plays a key role in determining the performance of a chip. Algorithms for physical design must also ensure that all the rules required by the fabrication are followed and that the layout is within the tolerance limits of the fabrication process. Finally, algorithms must be efficient and should be able to handle very large designs. Efficient algorithms not only lead to fast turnaround time, but also permit designers to iteratively improve the layouts.

VLSI DESIGN CYCLE:

The design process of producing a packaged VLSI chip physically follows various steps which is popularly known as VLSI design cycle. This design cycle is normally represented by a flow chart shown below. The various steps involved in the design cycle are elaborated below.

(i). **System specification:** The specifications of the system to be designed are exactly specified in this step. It considers performance, functionality, and the physical dimensions of the design. The choice of fabrication technology and design techniques are also considered. The end results are specifications for the size, speed, power, and functionality of the VLSI system to be designed.

(ii) **Functional design:** In this step, behavioral aspects of the system are considered. The outcome is usually a timing diagram or other relationships between sub-units. This information is used to improve the overall design process and to reduce the complexity of the subsequent phases.



(iii). **Logic design:** In this step, the functional design is converted into a logical design, using the Boolean expressions. These expressions are minimized to achieve the smallest logic design

which conforms to the functional design. This logic design of the system is simulated and tested to verify its correctness.

(iv).Circuit design: This step involves conversion of Boolean expressions into a circuit representation by taking into consideration the speed and power requirements of the original design. The electrical behavior of the various components are also considered in this phase. The circuit design is usually expressed in a detailed circuit diagram.

(v).Physical design: In this step, the circuit representation of each component is converted into a geometric representation. This representation is a set of geometric patterns which perform the intended logic function of the corresponding component. Connections between different components are also expressed as geometric patterns. (This geometric representation of a circuit is called a layout). The exact details of the layout also depend on design rules, which are guidelines based on the limitations of the fabrication process and the electrical properties of the fabrication materials. Physical design is a very complex process, therefore, it is usually broken down into various sub-steps in order to handle the complexity of the problem.

(vi). Design verification: In this step, the layout is verified to ensure that the layout meets the system specifications and the fabrication requirements. Design verification consists of design rule checking (DRC) and circuit extraction. DRC is a process which verifies that all geometric patterns meet the design rules imposed by the fabrication process. After checking the layout for design rule violations and removing them, the functionality of the layout is verified by circuit extraction. This is a reverse engineering process and generates the circuit representation from the layout. This reverse engineered circuit representation can then be compared to the original circuit representation to verify the correctness of the layout.

(vii). Fabrication: This step is followed after the design verification. The fabrication process consists of several steps like, preparation of wafer, deposition, and diffusion of various materials on the wafer according to the layout description. A typical wafer is 10 cm in diameter and can be used to produce between 12 and 30 chips. Before the chip is mass produced, a prototype is made and tested.

(viii). Packaging, testing, and debugging : In this step, the chip is fabricated and diced in a fabrication facility. Each chip is then packaged and tested to ensure that it meets all the design specifications and that it functions properly. Chips used in printed circuit boards (PCBs) are

packaged in a dual in-line package (DIP) or pin grid array (PGA). Chips which are to be used in a multichip module (MCM) are not packaged because MCMs use bare or naked chips.

PHYSICAL DESIGN CYCLE :

The Physical design cycle converts a circuit diagram into a layout. This complex task is completed in several steps, like s partitioning, floor-planning, placement, routing, and lay-out compaction etc. The details of these steps are given below.

(a).Partitioning : The chip layout is always a complex task and hence it is divided into several smaller tasks. A chip may contain several million transistors. Layout of the entire circuit cannot be handled due to the limitation of memory space as well as computation power available. Therefore, it is normally partitioned by grouping the components into blocks. The actual partitioning process considers many factors such as size of the blocks, number of blocks, and number of interconnections between the blocks. The output of partitioning is a set of blocks along with the interconnections required between blocks. The set of interconnections required is referred to as a net list. In large circuits the partitioning process is hierarchical and at the topmost level a chip may have between 5 and 25 blocks. Each module is then partitioned recursively into smaller blocks.

A disadvantage of the partitioning process is that it may degrade the performance of the final design. During partitioning, critical components should be assigned to the same partition. If such an assignment is not possible, then appropriate timing constraints must be generated to keep the two critical components close together. Usually, several components, forming a critical path, determine the chip performance. If each component is assigned to a different partition, the critical path may be too long. Minimizing the length of critical paths improves system performance

After a chip has been partitioned, each of the sub-circuits must be placed on a fixed plane and the nets between all the partitions must be interconnected. The placement of the sub-circuits is done by the placement algorithms and the nets are routed by using routing algorithms.

(b) Placement: It is the process of arranging a set of modules on the layout surface. Each module has fixed shape and fixed terminal locations. A poor placement uses larger area and hence results in performance degradation.

The placement process determines the exact positions of the blocks on the chip, so as to find a minimum area arrangement for the blocks that allows completion of interconnections between

the blocks. Placement is typically done in two phases. In the first phase an initial placement is created. In the second phase the initial placement is evaluated and iterative improvements are made until the layout has minimum area and conforms to design specifications.

It is important to note that some space between the blocks is intentionally left empty to allow interconnections between blocks. Placement may lead to un-routable design, i.e., routing may not be possible in the space provided. Thus, another iteration of placement is necessary. To limit the number of iterations of the placement algorithm, an estimate of the required routing space is used during the placement phase. A good routing and circuit performance heavily depend on a good placement algorithm. This is due to the fact that once the position of each block is fixed, very little can be done to improve the routing and the overall circuit performance.

There are various types of placements.

System-level placement : Place all the PCBs together such that Area occupied is minimum and Heat dissipation is within limits.

Board-level placement : All the chips have to be placed on a PCB. Area is fixed All modules of rectangular shape.

The objective is to , Minimize the number of routing layers and Meet system performance requirements.

Chip-level placement : Normally, floor planning / placement carried out along with pin assignment. It has limited number of routing layers (2 to 4). Bad placements may be unroutable.

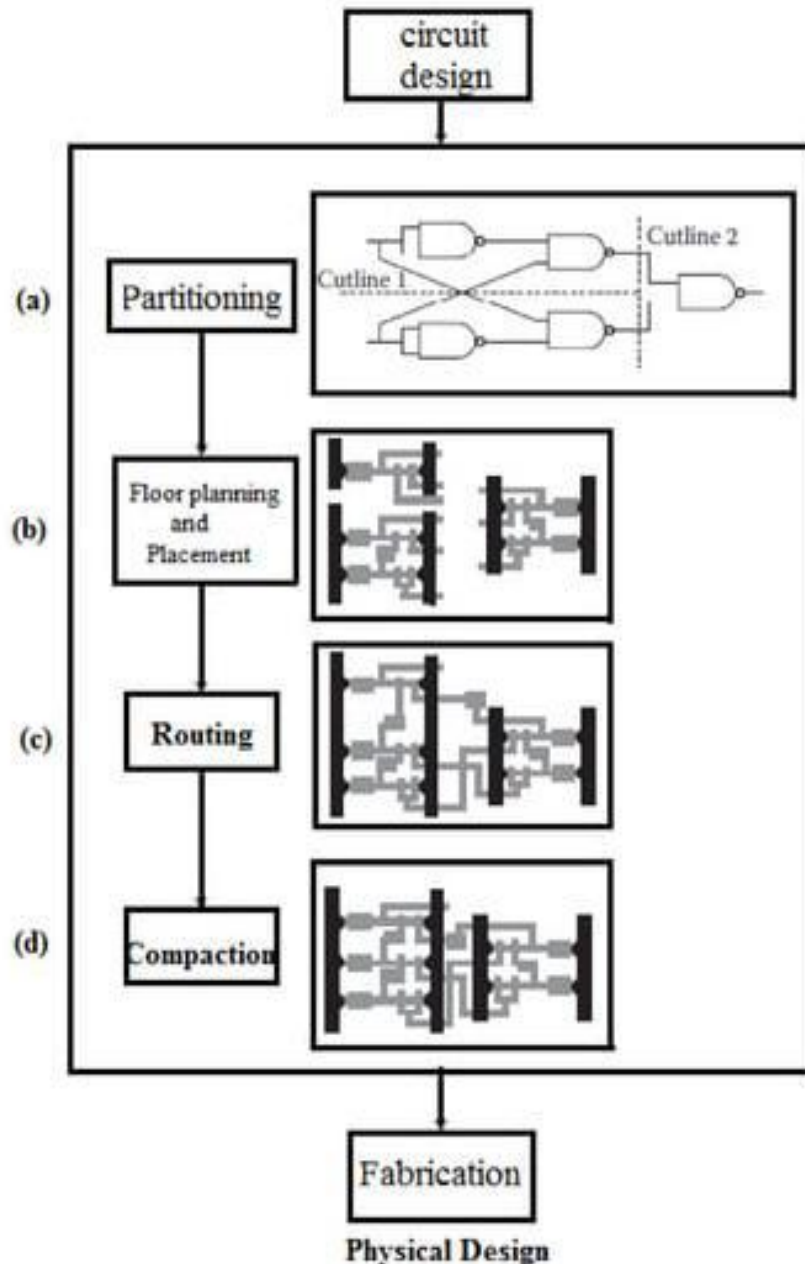
Can be detected only later (during routing). Costly delays in design cycle. Minimization of area.

Floorplanning:

Floor-plan design is an important step in physical design of VLSI circuits to plan the positions of a set of circuit modules on a chip in order to optimize the circuit performance.

In floor-planning, the information of a set of modules, including their areas and interconnection is considered and the goal is to plan their positions on a chip to minimize the total chip area and interconnect cost.

In the floor planning phase, the macro cells are positioned on the layout surface in such a way that no blocks overlap and that there is enough space left to complete the interconnections. The input for the floor planning is a set of modules, a list of terminals (pins for interconnections) for each module and a net list, which describes the terminals which have to be connected.



Different approaches are followed to the floor planning problem. Wimer et al. describe a branch and bound approach for the floor plan sizing problem, i.e. finding an optimal combination of all possible layout-alternatives for all modules after placement. While their algorithm is able to find the best solution for this problem, it is very time consuming, especially for real problem instances. Cohoon et al. implemented a genetic algorithm for the whole floor planning problem. Their algorithm makes use of estimates for the required routing space to ensure completion of

the interconnections. Another more often used heuristic solution method for placement is Simulated Annealing

(c) Routing: The main objective in this step is to complete the interconnections between blocks according to the specified netlist. First, the space not occupied by the blocks (called the routing space) is partitioned into rectangular regions called channels and switchboxes. The goal of a router is to complete all circuit connections using the shortest possible wire length and using only the channels and switchboxes. This is usually done in two phases, referred to as the global routing and detailed routing phases.

In global routing, connections are completed between the proper blocks of the circuit disregarding the exact geometric details of each wire and pin. For each wire, the global router finds a list of channels which are to be used as a passage way for that wire. In other words, global routing specifies the "loose route" of a wire through different regions in the routing space.

Global routing is followed by detailed routing, which completes point-to-point connections between pins on the blocks. Loose routing is converted into exact routing by specifying geometric information such as width of wires and their layer assignments. Detailed routing includes channel routing and switchbox routing.

As all problems in routing are computationally hard, the researchers have focused on heuristic algorithms. As a result, experimental evaluation has become an integral part of all algorithms and several benchmarks have been standardized. Due to the nature of the routing algorithms, complete routing of all the connections cannot be guaranteed in many cases

(d).Compaction: The operation of layout area minimization without violating the design rules and without altering the original functionality of layout is called as compaction. The input of compaction is layout and output is also layout but by minimizing area.

Compaction is done by three ways:

- (i) By reducing space between blocks without violating design space rule.
- (ii) By reducing size of each block without violating design size rule.
- (iii).By reducing shape of blocks without violating electrical characteristics of blocks.

Therefore compaction is very complex process because this process requires the knowledge of all design rules. Due to the use of strategies compaction algorithms are divided into one-dimensional algorithms (either in x-dimension or y-dimension), two dimensional algorithms

(both in x-dimension and y-dimension) and topological algorithm (moving of separate cells according to routing constraints).

Types of compaction techniques:

(i) 1-Dimensional compaction:

In this technique compaction is done only in one dimension either in x-direction or y-direction until no further compaction is possible. There are two types of constraints which relates to these compaction techniques **(i) Separation constraint (ii) Connectivity constraint.**

(ii).2-Dimensional compaction:

In this method compaction is done in both dimension x-dimensions as well as in y-dimension. 2-D compaction is in general much better than performing 1-D compaction. If 2-D compaction, solved optimally, produces minimum-area layouts. The trade off in this technique is the much time consumption. Thus we use 3/2-D Compaction.

(iii) 3/2-D Compaction:

In this technique the blocks are moved in such a way that it not only compact the circuit but also resolve interferences. Since the geometry is not as free as in 2-D Compaction.

In this method two lists are formed one is ceiling another is floor. First is formed by the blocks which are appeared from the top & second is formed by the blocks which are appeared from the bottom. Selects the lowest block in the ceiling list and moves it to the place on the floor which maximizes the gap between floor and ceiling. The process is continued until all blocks are moved from ceiling to floor.

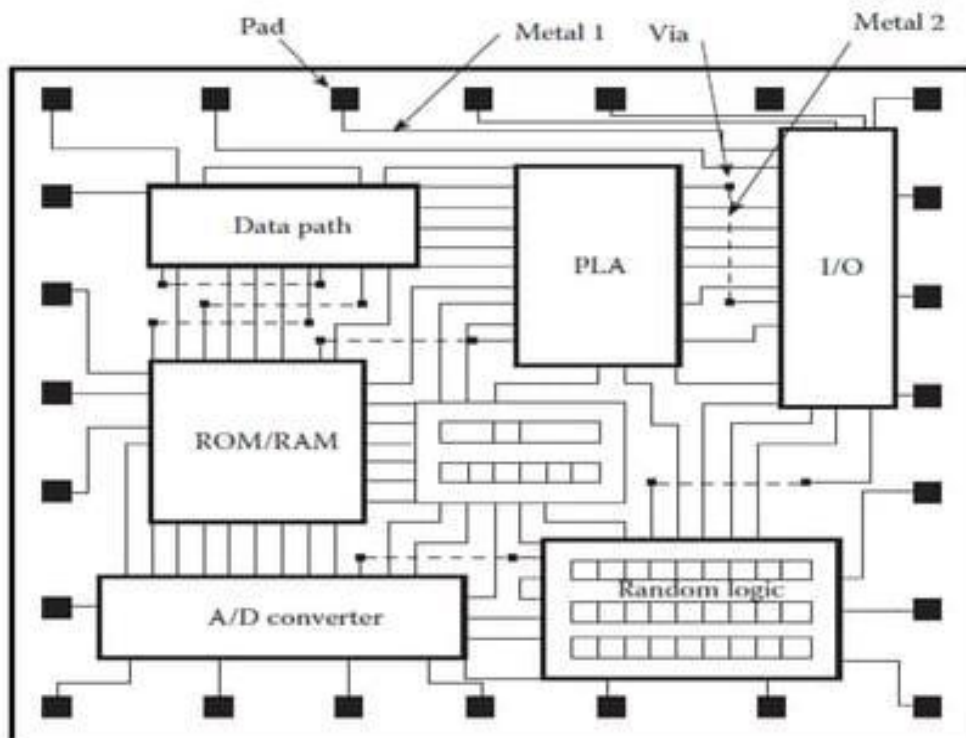
VLSI –DSIGN STYLES :

Though the partitioning of a physical design decomposes the physical design into several conceptually easier steps, still each step is computationally very hard. So, in order to reduce the the complexity of physical design and to get high yield certain restricted models and design styles are proposed. They are (i) full-custom design style (ii) standard cell design style (iii) gate array design style and (iv)

(i).Full-custom design style :

It is the most general form of layout in which the circuit is partitioned into a collection of sub-circuits according to some criteria such as functionality of each sub-circuit. In this design style, each sub-circuit is called a functional block or simply a block. The full custom design style allows functional blocks to be of any size. Blocks can be placed at any location on the chip

surface without restriction. In other words, this style is characterized by the absence of any constraints on the physical design process. This design style allows for very compact designs. But, the process of automating a full-custom design style has a much higher complexity than other restricted models. For this reason, it is used only when final design must have a minimum area and designing time is less of a factor. The full-custom structure of a design is shown below. The automation process for a full-custom layout is still a topic of intensive research. Some phases of physical design of a full-custom chip may be done manually to optimize the layout. Layout compaction is a very important aspect in full-custom. The rectangular solid boxes around the boundary of the circuit are called I-O pads.



Pads are used to complete interconnections between chips or interconnections between chip and the board. The space not occupied by blocks is used for routing of interconnecting wires. Initially all the blocks are placed within the chip area, with the objective of minimizing the total area. However, enough space must be left between the blocks to complete the routing. Usually several metal layers are used for routing interconnections. Currently, two metal layers are common for routing and the three-metal layer process is gaining acceptance, as the fabrication costs become more feasible. The routing area needed between the blocks becomes increasingly smaller as more routing layers are used. This is because some routing is done on top of the transistors in the

additional metal layers. If all the routing can be done on top of the transistors, the total chip area is determined by the area of the transistors.

In a hierarchical design of circuit each block in full-custom design may be very complex and may consist of several sub blocks, which in turn may be designed using the full-custom design style or other design styles. It is clear that as any block is allowed to be placed anywhere on the chip, the problem of optimizing area and interconnection of wires becomes difficult. Full-custom design is very time consuming thus, the method is inappropriate for very large circuits, unless performance is of utmost importance. Full-custom is usually used for the layout of chips like microprocessors etc.

(ii).Standard cell design style : This is the more restricted design style and the design process is simpler than a full-custom design style. Standard cell methodology considers the layout to consist of rectangular cells of the same height. Initially, a circuit is partitioned into several smaller blocks, each of which is equivalent to some predefined sub-circuit or cell. The functionality and electrical characteristics of each predefined cell are tested, analyzed, and specified. A collection of these cells is called a cell library, usually consisting of 200–400 cells. Terminals on cells may be located either on the boundary or in the center of the cells. Cells are placed in rows and the space between two rows is called a channel. These channels are used to perform interconnections between cells. If two cells to be interconnected lie in the same row or in adjacent rows, then the channel between the rows is used for interconnection. However, if two cells to be connected lie in two nonadjacent rows, then their interconnection wire passes through the empty space between any two cells, or feed through.

Standard cell design is well suited for moderate-size circuits and medium production volumes. Physical design using standard cells is simpler as compared to full-custom and efficient using modern design tools. The standard cell design style is also widely used to implement the “random logic” of the full-custom design . While standard cell designs are developed more quickly, a substantial initial investment is needed in the development of the cell library, which may consist of several hundred cells. Each cell in the cell library is “handcrafted” and requires a highly skilled design engineer. Each type of cell must be created with several transistor sizes. Each cell must then be tested by simulation and its performance must be characterized. A standard cell design usually takes more area than a full-custom or a handcrafted design.

However, as more metal layers become available for routing, the difference in area between the two design styles will gradually be reduced.

(iii). Gate array design style : This design style is a simplified version of the standard cell design style. Unlike the cells in standard cell designs, all the cells in gate array are identical. The entire wafer is prefabricated with an array of identical gates or cells. These cells are separated by both vertical and horizontal spaces called vertical and horizontal channels. The circuit design is modified such that it can be partitioned into a number of identical blocks. Each block must be logically equivalent to a cell on the gate array. The name “gate array” signifies the fact that each cell may simply be a gate, such as a three-input NAND gate. Each block in the design is mapped or placed onto a prefabricated cell on the wafer during the partitioning / placement phase, which is reduced to a block-to-cell assignment problem.

The number of partitioned blocks must be less than or equal to that of the total number of cells on the wafer. Once the circuit is partitioned into identical blocks, the task is to make the interconnections between the prefabricated cells on the wafer using horizontal and vertical channels to form the actual circuit. The uncommitted gate array is taken into the fabrication facility and routing layers are fabricated on top of the wafer. The completed wafer is also called a customized wafer.

This simplicity of gate array design is gained at the cost of rigidity imposed upon the circuit both by the technology and the prefabricated wafers. The advantage of gate arrays is that the steps involved for creating any prefabricated wafer are the same, and only the last few steps in the fabrication process actually depend on the application for which the design will be used. Hence, gate arrays are cheaper and easier to produce than full-custom or standard cell. Similar to standard cell design, gate array is also a nonhierarchical structure. The gate array architecture is the most restricted form of layout. It means that it is the simplest for algorithms to work with. For example, the task of routing in gate array is to determine if a given placement is routable. The routability problem is conceptually simpler as compared to the routing problem in standard cell and full-custom design styles.

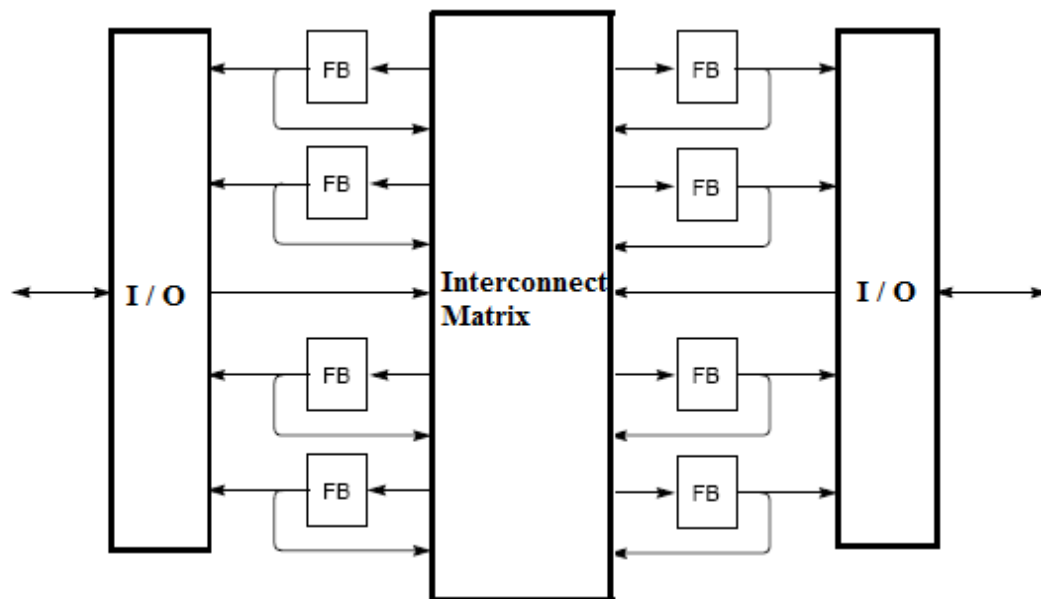
(iv).Field Programmable Gate Array Design (FPGA):

In this design, chips are prefabricated with logic blocks and interconnects. Logic and interconnects can be programmed (erased and reprogrammed) by users. No fabrication is needed. Interconnects are predefined wire segments of fixed lengths with switches in between.

Complex Programmable Logic Device (CPLD):

CPLDs were pioneered by Altera, first in their family of chips called Classic EPLDs, and then in three additional series, called MAX 5000, MAX 7000 and MAX 9000. The CPLD is the complex programmable Logic Device which is more complex than the SPLD. This is built on SPLD architecture and creates a much larger design. Consequently, the SPLD can be used to integrate the functions of a number of discrete digital ICs into a single device and the CPLD can be used to integrate the functions of a number of SPLDs into a single device.

So, the CPLD architecture is based on a small number of logic blocks and a global programmable interconnect. Instead of relying on a programming unit to configure chip, it is advantageous to be able to perform the programming while the chip is still attached to its circuit board. This method of programming is known as In-System programming (ISP). It is not usually provided for PLAs (or) PALs, but it is available for the more sophisticated chips known as Complex programmable logic device.

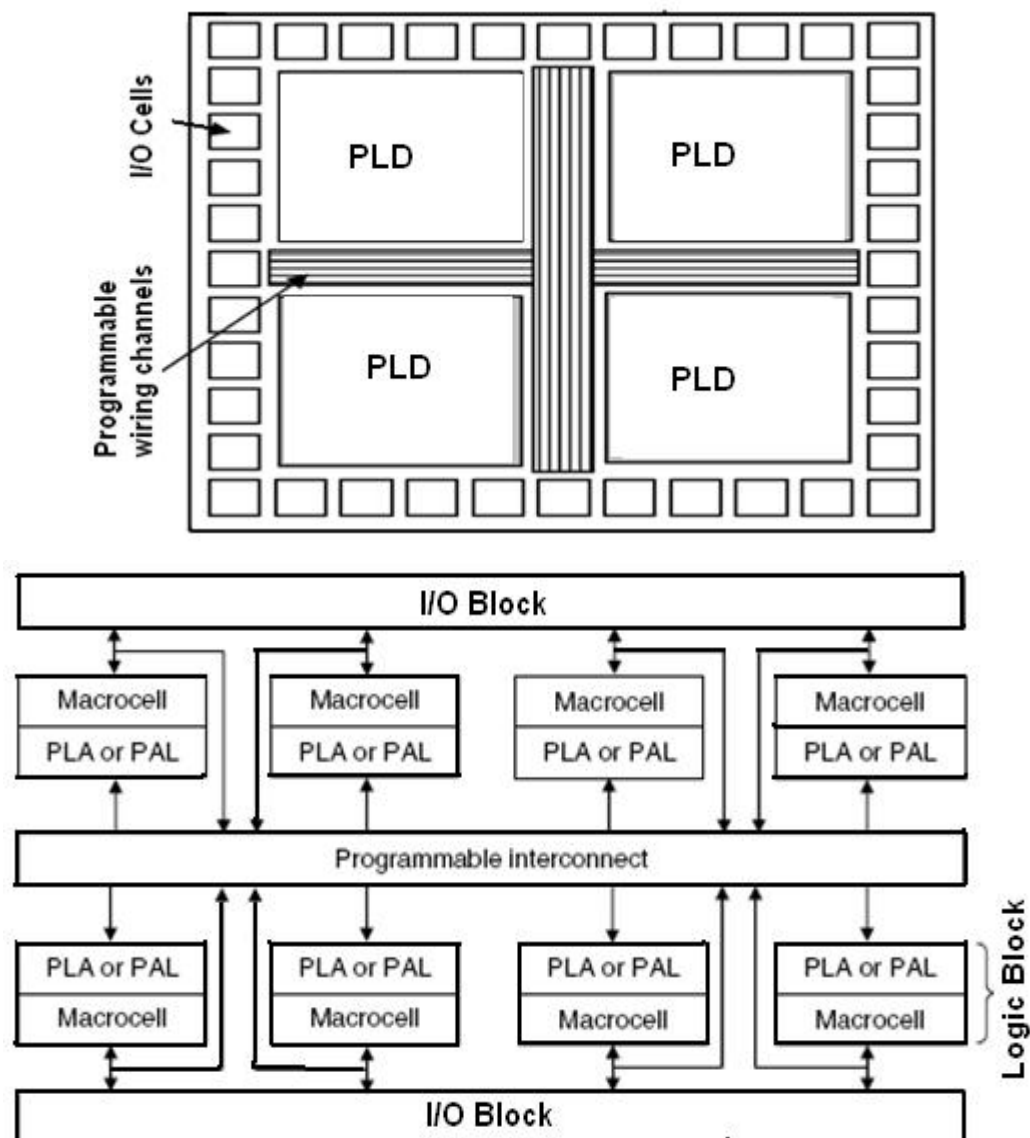


The CPLD consists of a number of logic blocks or functional blocks, each of which contains a macrocell and either a PLA or PAL circuit arrangement. In this view, eight logic blocks are shown. The building block of the CPLD is the macro-cell, which contains logic implementing disjunctive normal form expressions and more specialized logic operations. The macro cell provides additional circuitry to accommodate registered or nonregistered outputs, along with signal polarity control. Polarity control provides an output that is a true signal or a complement

of the true signal. The actual number of logic blocks within a CPLD varies; the more logic blocks available, the larger the design that can be configured.

In the center of the design is a global programmable interconnect. This interconnect allows connections to the logic block macrocells and the I/O cell arrays (the digital I/O cells of the CPLD connecting to the pins of the CPLD package). The programmable interconnect is usually based on either array-based interconnect or multiplexer-based interconnect:

- Array-based interconnect allows any signal within the programmable interconnect to connect to any logic block within the CPLD.



This is achieved by allowing horizontal and vertical routing within the programmable interconnect and allowing the crossover points to be connected or unconnected (the same idea as with the PLA and PAL), depending on the CPLD configuration.

- Multiplexer-based interconnect uses digital multiplexers connected to each of the macrocell inputs within the logic blocks. Specific signals within the programmable interconnect are connected to specific inputs of the multiplexers. It would not be practical to connect all internal signals within the programmable interconnect to the inputs of all multiplexers due to size and speed of operation considerations.

FPGAs – FIELD PROGRAMMABLE GATE ARRAYS

The FPGA concept emerged in 1985 with the XC2064™ FPGA family from Xilinx . The “FPGA is an integrated circuit that contains many (64 to over 10,000) identical logic cells that can be viewed as standard components.” The individual cells are interconnected by a matrix of wires and programmable switches. A user's design is implemented by specifying the simple logic function for each cell and selectively closing the switches in the interconnect matrix. The array of logic cells and interconnect form a fabric of basic building blocks for logic circuits. Complex designs are created by combining these basic blocks to create the desired circuit.

Unlike CPLDs (Complex Programmable Logic Devices) FPGAs contain neither AND nor OR planes. The FPGA architecture consists of configurable logic blocks, configurable I/O blocks, and programmable interconnect. Also, there will be clock circuitry for driving the clock signals to each logic block, and additional logic resources such as ALUs, memory, and decoders may be available. The two basic types of programmable elements for an FPGA are Static RAM and antifuses.

Each logic block in an FPGA has a small number of inputs and one output. A look up table (LUT) is the most commonly used type of logic block used within FPGAs.

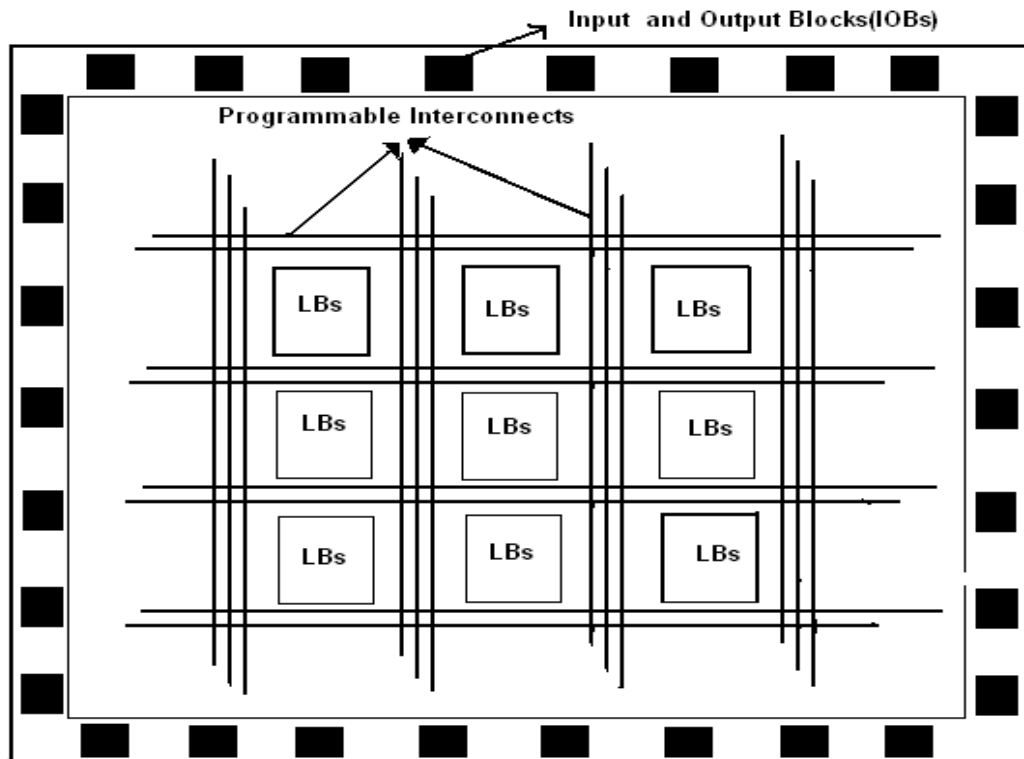
There are two types of FPGAs. (i) SRAM based FPGAs and (ii) Antifuse technology based (OTP)

Every FPGA consists of the following elements

- Configurable logic blocks (CLBs)
- Configurable input output blocks (IOBs)
- Two layer metal network of vertical and horizontal lines for interconnecting the CLBs

Configurable logic blocks(CLBs):

The configurable logic block is the basic logic cell and it is either RAM based or PLD based . It consists of registers (memory), Muxes and combinatorial functional unit. An array of CLBS are embedded within a set of vertical and horizontal channels that contain routing which can be personalized to interconnect CLBs.

**Configurable Input / Output logic locks (IOBs):**

CLBs and routing channels are surrounded by a set of programmable I/Os which is an arrangement of transistors for configurable I/O drivers.

Programmable interconnects:

These are unprogrammed interconnection resources on the chip which have channeled routing with fuse links. Four types of interconnect architectures are available. They are

- Row-Column Architecture
- Island Style Architecture
- Sea-of-Gates Architecture

Advantages of FPGAs:

- Design cycle is significantly reduced. A user can program an FPGA design in a few minutes or seconds rather than weeks or months required for mask programmed parts.
- High gate density i.e, it offers large gate counts.
- No custom masks tooling is required (Low cost).
- Low risk and highly flexible.
- Reprogram ability for some FPGAs (design can be altered easily).
- Suitable for prototyping.
- Parallelism
 - Allows for system-level extraction of parallelism to match input data at design time
 - Huge computational capability
- Fast development and Dynamic reconfiguration
- Updating new pattern matching rules (or simply rules)
 - Device should not stop when updating new rules
- Update time for new rules
 - To provide fast response to new attacks, the compilation and updating time for new rules needs to be short
 - In case of a hardwired FPGA architecture, the update time is mostly dependent on place & route time
 - Memory-based units can provide near instantaneous updates

Limitations:

- Speed is comparatively less.
- The circuit delay depends on the performance of the design implementation tools.
- The mapping of the logic design into FPGA architecture requires sophisticated design implementation (CAD) tools than PLDs.

FPGA Programming Technologies:**(a) Antifuse Technology:**

An antifuse is a two terminal device that when un-programmed has a very high resistance between the two terminals and when programmed, or “blown”, creates a very low resistance or

permanent connection. The application of a high voltage from 11 V to 21 V will create the low resistive permanent connection. Antifuse technologies come in two types. The first is oxide-nitride-oxide (ONO) dielectric based and the other is amorphous silicon or metal-to-metal antifuse structures.

Dielectric based antifuses consist of a dielectric material between N⁺ diffusion and polysilicon which breaks down when a high voltage is applied. Early dielectrics were a single-layered oxide dielectric until Actel came out with the programmable low impedance circuit element (PLICE), which is a multi-layer oxide-nitride-oxide (ONO) dielectric fuse. A high voltage across the PLICE melts the dielectric and creates polycrystalline silicon between the terminals. When the PLICE is blown, it adds three layers rather than the double metal CMOS process. The layers are a thin layer of oxide on top of the N⁺ surface, Low-pressure Chemical Vapor Deposition (LPCVD) nitride and the reoxidized top oxide. The programming current has an important effect because the higher the current during programming, the lower the link resistance, resulting in smaller thickness for the antifuse material. Programming circuits for antifuses need to supply high currents (15 ma for Actel) to insure high reliability and performance.

Amorphous silicon antifuse technology is the alternative to dielectric antifuse. It consists of amorphous silicon between two layers of metal that changes phases when current is applied. When the antifuse is not programmed the amorphous silicon has a resistance of 1 Giga ohm. When a high current (about 20 mA) is applied to the antifuse the amorphous silicon changes into a conductive polysilicon link. Quick Logic pASIC FPGA is a perfect example of an amorphous silicon antifuse technology.

(b). **SRAM-based Technology:**

SRAM FPGA architecture consists of static RAM cells to control pass gates or multiplexers. The FPGA speed is determined by the delay introduced by the logic cells and the routing channels. Multiplexers, look-up tables and output drivers affect the speed of signals through the logic cells. An FPGA with more PIPs is easier to route but introducing more routing delay. The size of the look-up table plays an important role depending on the design. Smaller LUTs provide higher density but larger ones are preferred for high-speed applications.

Distinguish between SRAM and Antifuse Technologies: The following points explain the differences between the two technologies.

1. Antifuse programming technology is faster than SRAM programming technology due to the RC delays introduced by the interconnect structure.
2. Antifuse technology has more silicon area per gate and is easier to route than SRAM technology.
3. A disadvantage of antifuse FPGA is that they require more process layers and mask steps and also contain high voltage programming transistors.
4. SRAM-based technology contains higher capacity than antifuse technologies.
5. SRAM based technology is very flexible with in-system programmability and the ability to reconfigure the design during the debugging stage while antifuse technology is one-time programmable (OTP). This ability reduces design and development, which reduces overall cost of the design. Another advantage to this is that SRAM technology can be programmed at the factory through complete verification test where the antifuse are tested as “blanks” and require programming by the user to verify design requirements and operation.
6. A disadvantage of SRAM technology is that it is volatile meaning it has to be reprogrammed every time power is turned off and on again. The SRAM usually require an extra memory element to program the chip which occupies board space .

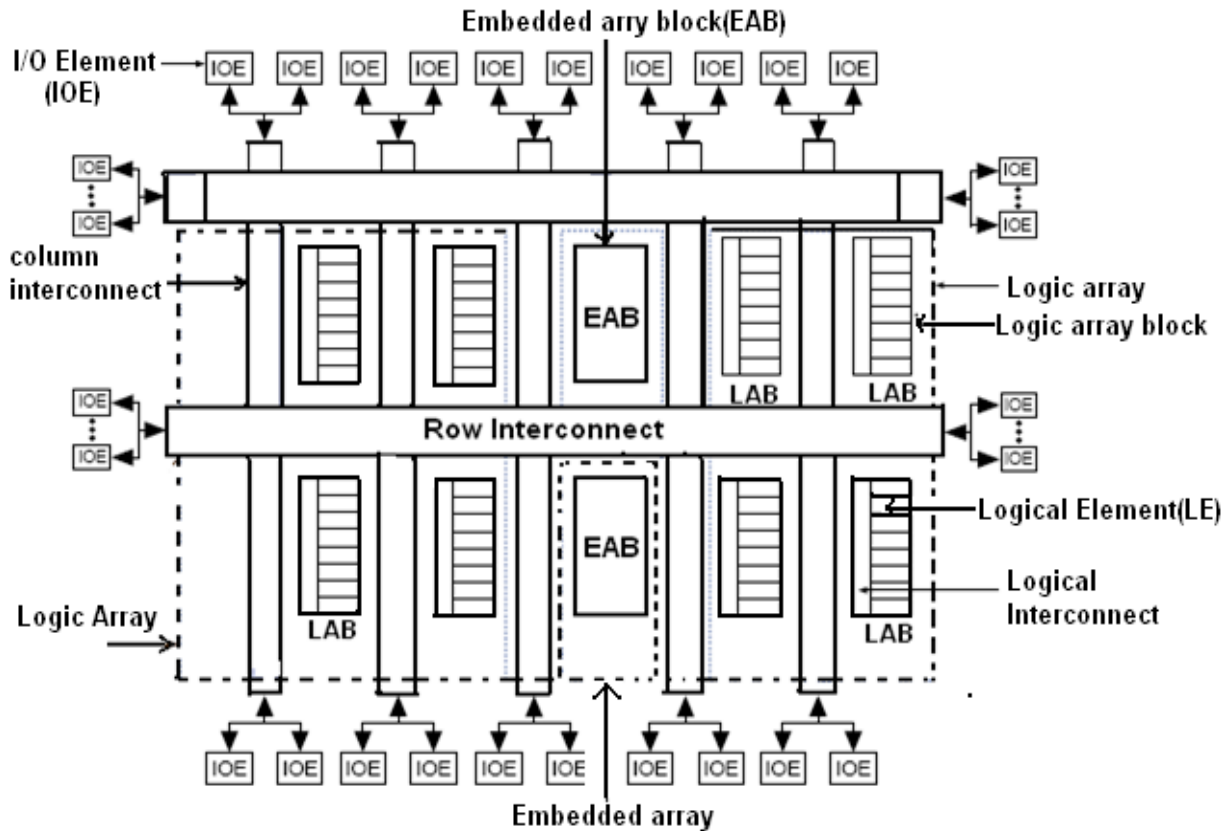
Altera's FLEX 10K Series CPLDs

Altera's FLEX 10K devices are the industry's first embedded PLDs. Based on reconfigurable CMOS SRAM elements, the **Flexible Logic Element MatriX** (FLEX) architecture incorporates all features necessary to implement common gate array mega functions. With 10,000 to 250,000 typical gates ,the FLEX 10K family provides the density, speed, and features to integrate entire systems, including multiple 32-bit buses, into a single device.

FLEX 10K devices are reconfigurable .So, the designer is not required to generate test vectors for fault coverage purposes. Additionally, the designer need not manage inventories of different ASIC designs; FLEX 10K devices can be configured on the board for the specific functionality required.

Each FLEX 10K device contains an Embedded Array (EA) and a Logic Array (LA).

The **Embedded Array** is used to implement a variety of memory functions or complex logic functions, such as digital signal processing (DSP) ,microcontroller, wide-data-path manipulation, and data-transformation functions.



The **Logic Array** performs the same function as the sea-of-gates in the gate array: it is used to implement general logic, such as counters, adders, state machines, and multiplexers.

The combination of embedded and logic arrays provides the high performance and high density of embedded gate arrays, enabling designers to implement an entire system on a single device.

FLEX 10K devices are configured at system power-up with data stored in an Altera serial configuration device or provided by a system controller.

Logic Element is, the smallest unit of logic in the FLEX 10K architecture, has a compact size that provides efficient logic utilization. Each LE contains a four-input LUT, which is a function generator that can quickly compute any function of four variables. In addition, each LE contains a programmable flip flop with a synchronous enable, a carry chain, and a cascade chain. Each LE drives both the local and the Fast Track. The programmable flip flop in the LE can be configured for D, T, JK, or SR operation. The clock, clear, and preset control signals on the flip flop can be driven by global signals, general-purpose I/O pins, or any internal logic. For combinatorial functions, the flip flop is bypassed and the output of the LUT drives the output of the LE.

The carry chain provides a very fast (as low as 0.2 ns) carry-forward function between Les. The carry-in signal from a lower-order bit drives forward into the higher-order bit via the carry chain, and feeds into both the LUT and the next portion of the carry chain. This feature allows the FLEX 10K architecture to implement high-speed counters, adders, and comparators of arbitrary width efficiently. Cascade Chain in the FLEX 10K architecture can implement functions that have a very wide fan-in. Adjacent LUTs can be used to compute portions of the function in parallel; the cascade chain serially connects the intermediate values. The cascade chain can use a logical AND or logical OR (via De Morgan's inversion) to connect the outputs of adjacent Les. Each additional LE provides four more inputs to the effective width of a function, with a delay as low as 0.7 ns per LE. Cascade chain logic can be created automatically by the Compiler during design processing, or manually by the designer during design entry.

Altera offers the EPC1, EPC2, EPC16, and EPC1441 configuration devices, which configure FLEX 10K devices via a serial data stream. Configuration data can also be downloaded from system RAM or from Altera's Bit Blaster serial download cable or Byte Blaster MV parallel port download cable. Even after configuring a FLEX 10K device, it can be reconfigured in-circuit by resetting the device and loading new data. Because reconfiguration requires less than 320 ms, real-time changes can be made during system operation. FLEX 10K devices contain an optimized interface that permits microprocessors to configure FLEX 10K devices serially or in parallel, and synchronously or asynchronously. The interface also enables microprocessors to treat a FLEX 10K device as memory and configure the device by writing to a virtual memory location, making it very easy for the designer to reconfigure the device.

SPEED PERFORMANCE :

The speed performance of PLDs is affected by their architectural features like I/O blocks, Logic Elements and the Interconnects. The routing of these elements also play an important role. For example if a finite state machine is to be implemented in an FPGA, then the amount of logic feeding each state machine flip-flop must be minimized. This follows because in FPGAs flip-flops are directly fed by logic blocks that have relatively few inputs (typically 4 - 8). If the state machine flip-flops are fed by more logic than will fit into a single logic block, then multiple levels of logic blocks will be needed, and speed-performance will decrease. Even in a CPLD architecture, speed-performance of a state machine can be significantly affected by state bit encoding, for example, in

the Altera MAX 7000 CPLDs, flip-flops that are fed by five or fewer product terms will operate faster than those that require more than five terms.

CPLDs are ideal for high-speed applications requiring critical timing and FPGAs are more flexible with the finer-grained architecture. Lattice semiconductor CPLD series architecture offered predictable timing, high densities, in-system programmability, flexible architecture for mixed combinatorial and register intensive designs and system partitioning. Some applications cannot use CPLDs. Planetary Spacecraft and earth orbiting satellites and science instruments require Radiation Hardened PLDs. Between CPLD and FPGAs, the CPLDs are fast and, predictable but the FPGAs are application dependent. The CPLD implementation of the sequential circuit is much faster than the FPGA version. However, the most interesting aspect is the difference between the 5-bit and 13-bit versions of the circuit. If both versions are operated at 100 MHz for CPLD implementation, while the 13-bit version is much slower than its smaller counterpart for the FPGA. This is a good example of how FPGAs are not suitable for implementing circuits that require “wide” logic gates (the 14-input AND-gates for this example), whereas CPLDs can easily implement such applications.

It is widely accepted by designers who use PLDs that FPGAs are the best choice for data-path circuits, because wide logic gates are not required and the number of flip-flops needed is large. But it is true that the FPGAs cannot provide required performance in an FPGA, and the CPLDs were successful. The reason that the CPLDs provided better performance is to do with their simple structure that provides for very high-speed paths from input pins, through AND-OR logic and flip-flops, to output pins.

It is also found experimentally that the CPLD-based counters achieve the maximum possible speed of t_{pd} plus the setup time of the flip-flop. This is because each counter bit needs up to 34-input AND functions that feed 4-input OR-gates. In the FLEX devices, high-speed carry chain is employed to implement the required wide AND. It is the speed of this carry chain that limits the speed-performance of the FLEX-based counter. Roughly, a counter of double the size has twice the carry chain length and thus half the speed performance. So, the conclusion is Altera CPLDs can implement very fast counters, however, it is difficult to construct many large counters in one device. The FLEX FPGAs are better suited for this purpose, but performance and routability are compromised when the carry chain hardware is used. Both performance and routability can be improved by enhancing the counter design with a very small cost in area. However, these gains cannot be realized without intimate knowledge of the FPGA architecture.

UNIT-V
VHDL Synthesis and Test and Testability

HDL:

HDL stands for Very High Speed Integrated Circuit Hardware Description Language.

HDL is similar to a computer programming language except that an HDL is used to describe hardware rather than a program which is executed on a computer. There are two HDLs available

- (a) VHDL
- (b) Verilog

Traditional Methods of Hardware Design:

- ✓ Design with Boolean equations
- ✓ Schematic based design

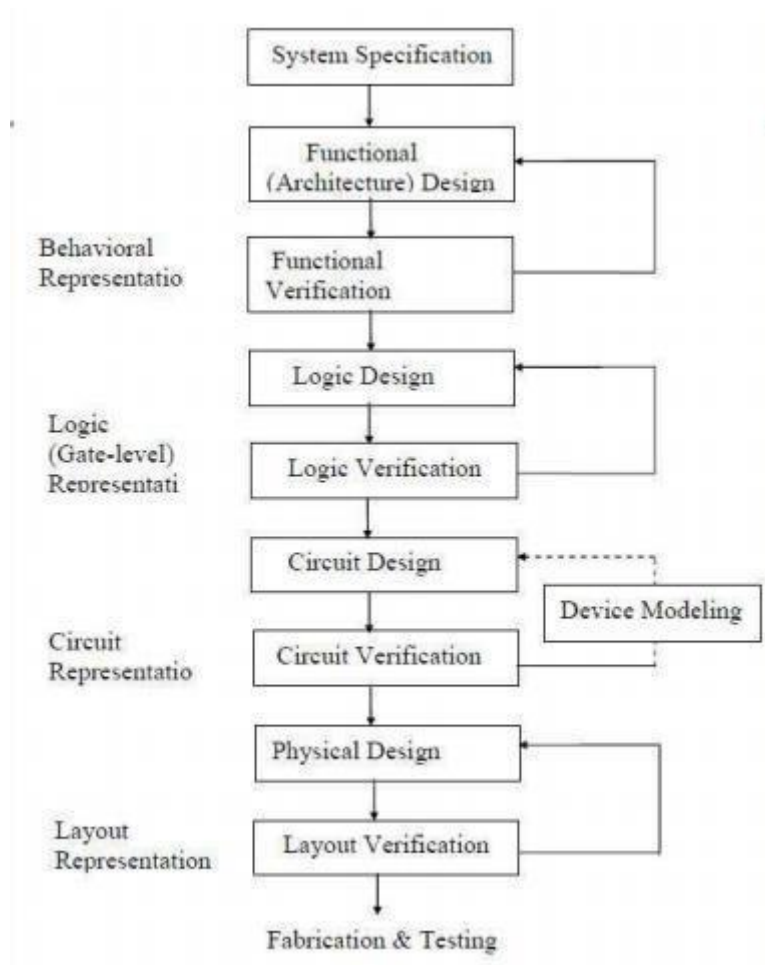
Features of VHDL

- ✓ Concurrent language
- ✓ Sequential language
- ✓ Netlist- It is textual information of logic cells and their interconnections. Data is available in the EDIF format.
- ✓ Test bench - used for verification of design
- ✓ Timing specification - supports synchronous and asynchronous timing models
- ✓ Supports all types of design methodologies-top-down and bottom-up or mixed design.

VHDL SYNTHESIS:

Synthesis is an automatic method of converting a higher level abstraction like behavioural into a gate level description. The basic function of synthesis is to produce a gate level netlist for target technology. There are three steps followed for converting to gate level design RTL description is translated to an un-optimized Boolean descriptions. It consisting of primitive gates like AND, OR & FFs. This is the functionally correct but un-optimized description. To produce the optimized Boolean equivalent description. Optimized description is mapped to actual logic gates by making use of technology library of the target process.

The most common VHDL types used in synthesizable VHDL code are `std_logic`, `std_logic_vector`, `signed`, `unsigned`, and `integer`. Because VHDL is a strongly-typed language, most often differing types cannot be used in the same expression.

CIRCUIT DESIGN FLOW:

The VLSI design flow is a complex and iterative process that involves collaboration among designers, verification engineers, and physical designers. The design flow can vary depending on the specific design requirements and the complexity of the IC or SoC.

The electronic circuit design process comprises two main stages: analysis and synthesis. This process requires the designer to accurately predict the voltage and current at every node in a circuit. Ideally, a designer should be able to predict the output of the circuit at each node, including the power supplies.

CIRCUIT SYNTHESIS

Circuit synthesis has the following steps:

- ✓ Translation
- ✓ Boolean optimization
- ✓ Flattening
- ✓ Factoring
- ✓ Mapping to Gates

Translation:

The RTL description is converted by the logic synthesis tool to an un-optimized, intermediate, internal representation. This process is known as translation. It is not user controllable. It is relatively simple and uses techniques of HDL constructs interpretation. Interpretation is a process which converts all conditional or sequential and concurrent statements to Boolean equivalent format.

Boolean optimization:

The optimization process takes an unoptimized Boolean description and converts it to an optimized Boolean description. Optimization is the process which decreases the area or increases the speed of a design.

Flattening

The process of converting unoptimized Boolean description to PLA format is known as flattening. A PLA structure is a very easy description in which to perform Boolean optimization.

Mapping to gates

The mapping process takes the optimised Boolean description and uses the logical and timing information from a technology library to build a netlist. This netlist is targeted to the users needs for area and speed. There are a number of possible netlists that are functionally same but vary widely in speed and area.

SIMULATION

Simulation is the process of applying stimuli (test inputs) to design under test over same duration of time and producing the response from the design under test. Simulation verifies the operation of user's design before actually implementing it as hardware. Necessity of simulation is:

Need to test the designs prior to implementation and usage.

Reduce the time for development

Decrease the time to market.

DESIGN CAPTURE TOOLS

HDL Design

- ✓ Schematic Design
- ✓ Floorplanning

HDL Design

HDLs are used to design two kinds of systems:

- ✓ Integrated Circuit
- ✓ Programmable Logic Devices

HDL design can be used for designing ICs like processor or any other kind of digital logic chip.

HDL specifies the model for the expected behaviour of circuit before actual circuit design and implementation.

PLDs like FPGA or CPLD can be designed with HDLs. HDL code is fed into logic compiler and output is uploaded into actual device. The important property of this procedure is that it is possible to change the code many times, compile it and upload in the same device.

Schematic Design

- ✓ Schematic design provides a means to draw and connect components.
- ✓ Schematic editors are available with various features like
- ✓ Creating, selecting and deleting parts by pointing
- ✓ Changing the graphic view by panning, zooming.

DESIGN VERIFICATION TOOLS

- ❖ The functionality of the CMOS chips is to be verified certain set of verification tools are used for testing specifications.
- ❖ The following tools are popular for design verification

1. Simulation

- ❖ Circuit Level Simulation
- ❖ Timing Simulation
- ❖ Logical Level Simulation
- ❖ Mixed mode Simulation

2. Timing verifiers

3. Netlist comparison

4. Layout extraction

5. Design rule verification

Schematic Rule Check (SRC)

- ❖ In cell based designs a schematic rule checker used to verify the schematics i.e schematic rule violation. The violation of rule may be indicated in terms of warning or errors.
 - ❖ SRC warnings
 - ❖ Floating wire segments
 - ❖ Open connection
 - ❖ Higher fanout
 - ❖ SRC errors
 - ❖ Undefined inputs/open inputs
 - ❖ Unmatched bus connections
 - ❖ Multiple drivers connection to single line
 - ❖ Different I/O pins

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 tools called DRC.
- ❖ Two different categories of DRC programs are used
 1. Polygonal check
 2. Raster scan check
- ❖ The polygonal design rule checks involves various mathematical operations during the check.

Fault Modelling:

A fault model is an engineering model of something that could go wrong in the construction or operation of a piece of equipment. From the model, the designer or user can then predict the consequences of this particular fault. Fault models can be used in almost all branches of engineering.

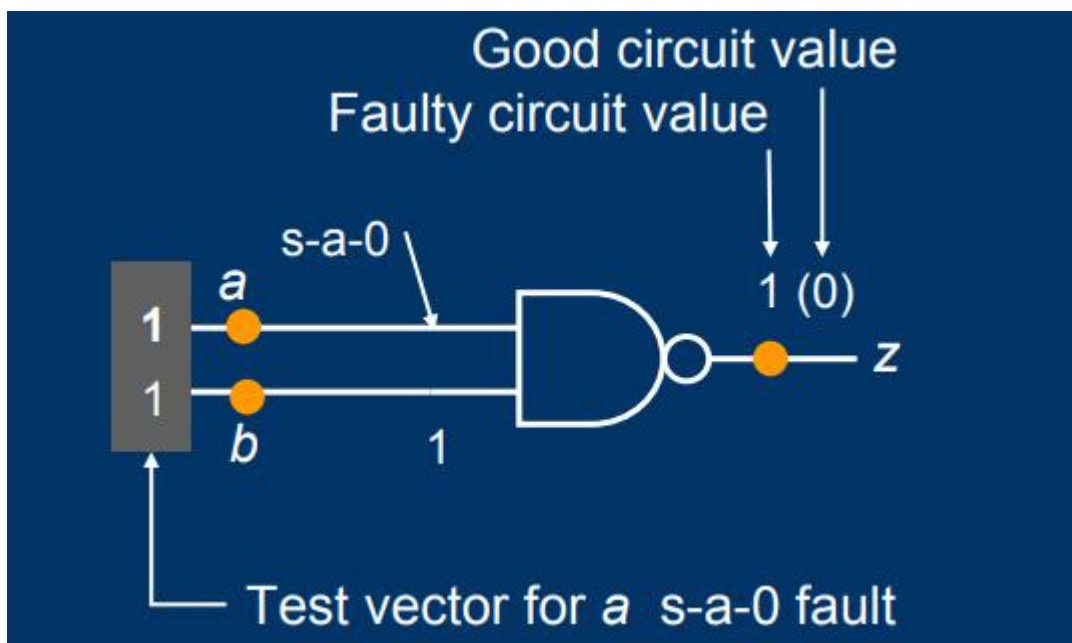
Different types of fault models available are:

- Single stuck at faults.
- Stuck Open and stuck short faults.
- Bridging faults.

Single stuck at faults:

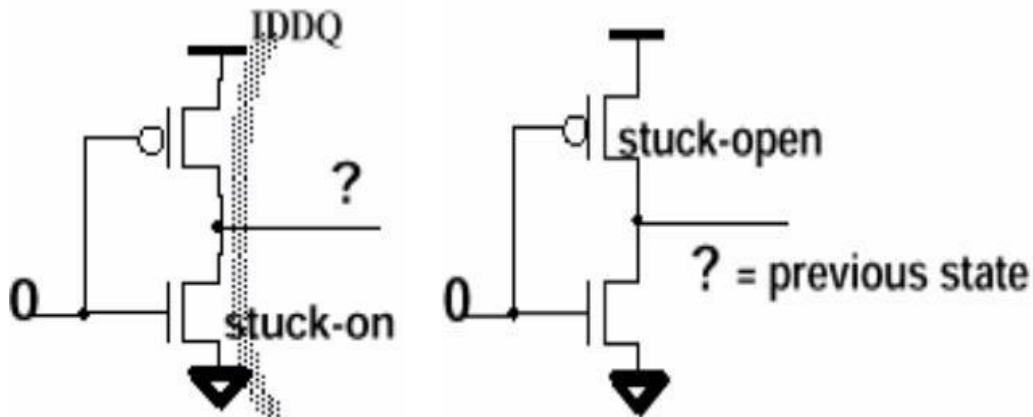
Three properties define a single stuck-at fault

- Only one line is faulty
- The faulty line is permanently set to 0 or 1
- The fault can be at an input or output of a gate



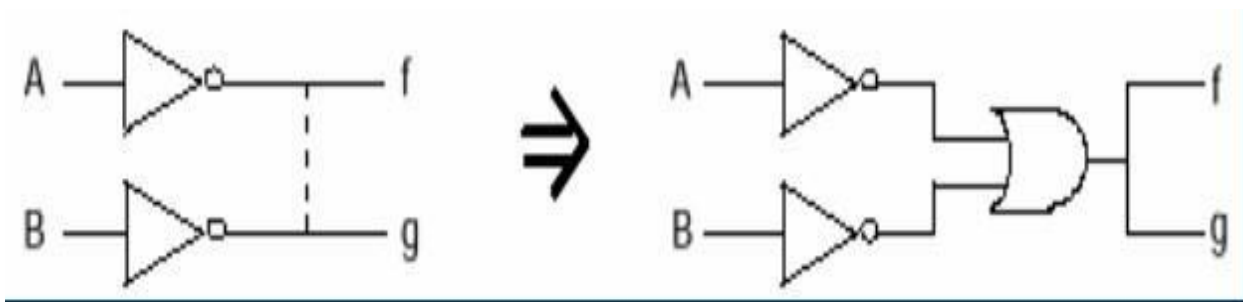
Stuck Open and stuck short faults:

- Transistor stuck-on may cause ambiguous logic level – depends on the relative impedances of the pull-up & pulldown networks
- When input is low, both P and N transistors are conducting causing increased quiescent current, called I_{DDQ} fault.
- Transistor stuck-open may cause output floating
- Below figures shows stuck short and stuck open faults in MOS transistors.



Bridging Faults:

Two or more normally distinct points (lines) are shorted together – Logic effect depends on technology – Wired- AND for TTL.



Fault Simulation:

Fault simulation is typically used to evaluate the fault coverage obtained by that set of test patterns. As a result, fault models are needed for fault simulation and for ATPG.

Purposes of fault simulation during design cycle:

- Guiding the TPG process.
- Measuring the effectiveness of the test patterns.
- Generating fault dictionaries.

Fault simulator needs in addition to the circuit model, stimuli and expected responses (that are needed for true-value simulation):

- Fault model
- Fault list

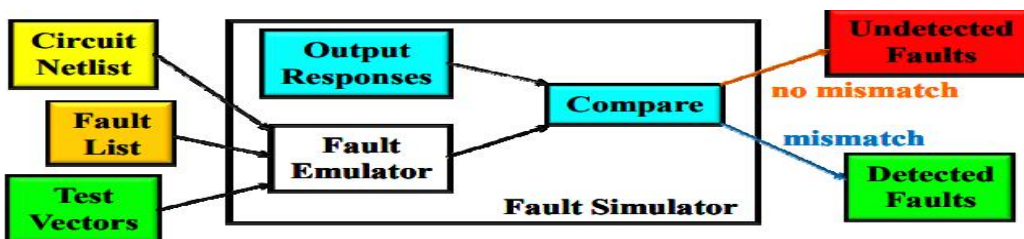


Fig: Fault simulation

Serial Fault Simulation:

If fault dropping is not employed, the effort of simulating n faults is equivalent to either:

- Simulating a circuit n times larger or
- Repeating the original true-value simulation n times.

True-value simulation is performed across all vectors and outputs saved.

Faulty circuits are simulated one-by-one by modifying circuit and running true-value simulator.

Simulation of faulty circuit stops as soon as fault is detected.

Adv: Any type of fault can be simulated, e.g., stuck-at, stuck-open, bridges, delay and analog faults.

For n faults, CPU time can be almost n times that of a true-value simulator. Fault dropping significantly improves on this.

Parallel Fault Simulation:

Most effective when: • Circuit consists of only logic gates.

- Stuck-at faults are modeled.
- Signals assume only binary, 0 or 1, values.
- All gates have the same delay (zero or unit).

Under these conditions, circuits $C(f_n)$ are almost identical.

Deductive Fault Simulation:

Circuit model assumptions are the same as those given for the parallel fault simulator, compiled-code and event-driven versions possible.

Only the fault free circuit, $C()$, is simulated. Faulty circuit values are deduced from the fault-free values. It processes all faults in a single pass of true-value simulation, i.e., it very fast! Note, however, that major modifications are required (and slow downs) to handle variable rise/fall delays, multiple signal states, etc.

A vector is simulated in true-value mode.

A deductive procedure is then performed on all lines in level-order from inputs to outputs. Fault lists are generated for each signal using the fault lists on the inputs to the gate generating that signal.

Concurrent Fault Simulation:

It extends the event-driven simulation method to simulation of faults. It can handle various types of circuit models, faults, signal states and timing models.

Details of the simulator model: Events Good events: Occur in the fault-free circuit, $C()$, and have three attributes, signal name, type of transition (0-to-1) and time of change. Fault-events: Occur on same lines in faulty circuits, $C(f_1) \dots C(f_n)$, but ONLY if transition is different from $C()$ transition. Three attributes + fault site and type.

Test generation:

Automatic Test Pattern Generation (ATPG) is used to determine test input sequence for digital circuit which distinguishes for the correct and faulty circuit behavior. These circuit behaviors are caused by the internal and external defects.

Three important testing methodologies will be briefly discussed in this blog: Design for Testability (DFT), Built-In Self-Test (BIST), and Automatic Test Pattern Generation (ATPG). For VLSI designs to be of the highest caliber and perform at their best, it is essential to comprehend these techniques.

Design for Testability:

There are two key concepts underlying all considerations for testability. They are:

1. controllability;
2. observability.

Design for testability (observability and controllability) is then reduced to a set of design rules or guidelines which, if obeyed, will facilitate test.

A failure during testing at the chip level may be due to a design defect or a poorly controlled fabrication process.

The inputs of the device under test (DUT) are subjected to a test pattern (or test vector) which supplies a set of binary values, in combination and/or in sequence, to detect faults. The specification of the test vector sequences must involve the designer, while the generation and application of test patterns to a DUT are the problems faced by the test engineer. Test pattern generation is assisted by using automatic test pattern generators (ATPG), but they are complicated to use properly and ATPG costs tend to rise rapidly with circuit size. Once the application of a test pattern has revealed a fault, the process of diagnosis must be invoked to localize the fault.

Built-In Self Test (BIST) Techniques:

In built-in self test (BIST) design, parts of the circuit are used to test the circuit itself. Online BIST is used to perform the test under normal operation, whereas off-line BIST is used to perform the test off-line. The essential circuit modules required for BIST include:

- *Pseudo random pattern generator (PRPG)
- * Output response analyzer (ORA)

The roles of these two modules are illustrated in Fig.1. The implementation of both PRPG and ORA can be done with Linear Feedback Shift Registers (LFSRs).

Pseudo Random Pattern Generator :-

To test the circuit, test patterns first have to be generated either by using a pseudo random pattern generator, a weighted test generator, an adaptive test generator, or other means. A pseudo random test generator circuit can use an LFSR, as shown in Fig. 2.

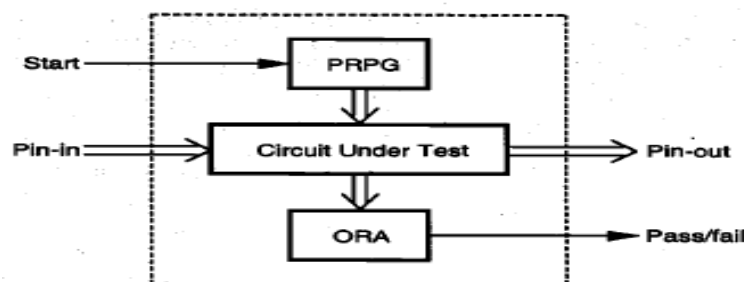


Figure 1 : A procedure for BIST

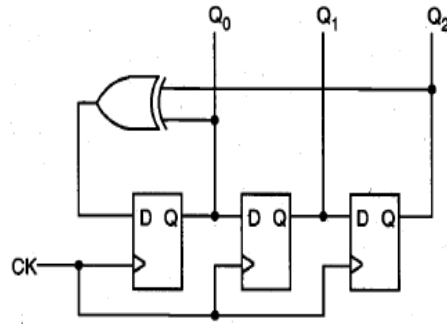


Figure 2 : A pseudo-random sequence generator using LFSR

Linear Feedback Shift Register as an ORA :-

To reduce the chip area penalty, data compression schemes are used to compare the compacted test responses instead of the entire raw test data. One of the popular data compression schemes is the signature analysis, which is based on the concept of cyclic redundancy checking. It uses polynomial division, which divides the polynomial representation of the test output data by a characteristic polynomial and then finds the remainder as the signature. The signature is then compared with the expected signature to determine whether the device under test is faulty. It is known that compression can cause some loss of fault coverage. It is possible that the output of a faulty circuit can match the output of the fault-free circuit; thus, the fault can go undetected in the signature analysis. Such a phenomenon is called **aliasing**.

In its simplest form, the signature generator consists of a single-input linear feedback shift register (LFSR), as shown in Fig.3 in which all the latches are edge-triggered. In this case, the signature is the content of this register after the last input bit has been sampled. The input sequence $\{a_n\}$ is represented by polynomial $G(x)$ and the output sequence by $Q(x)$. It can be shown that $G(x) = Q(x)P(x)R(x)$, where $P(x)$ is the characteristic polynomial of LFSR and $R(x)$ is the remainder, the degree of which is lower than that of $P(x)$. For the simple case in Fig. 3 the characteristic polynomial is

$$P(x) = 1 + x^2 + x^4 + x^5$$

$$G(x) = x^7 + x^6 + x^5 + x^4 + x^2 + 1$$

For the 8-bit input sequence $\{11110101\}$, the corresponding input polynomial is and the remainder term becomes $R(x) = x^4 + x^2$ which corresponds to the register contents of $\{00101\}$.

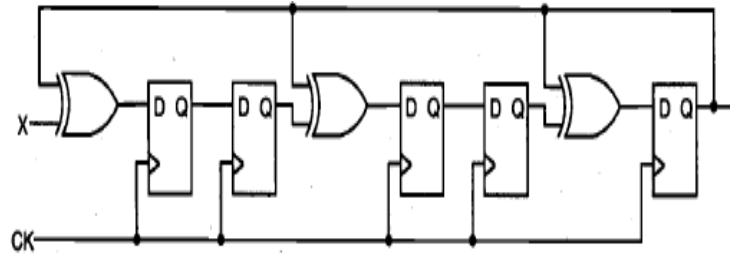


Figure 3: Polynomial division using LFSR for signature analysis

Boundary-Scan

Applications are found in high volume, high-end consumer products, telecommunication products, defense systems, computers, peripherals, and avionics. In fact, due to its economic advantages, some smaller companies that cannot afford expensive in-circuit testers are using boundary-scan.

The boundary-scan test architecture provides a means to test interconnects between integrated circuits on a board without using physical test probes. It adds a boundary-scan cell that includes a multiplexer and latches to each pin on the device.

Boundary-scan cells in a device can capture data from pin or core logic signals, or forced data onto pins. Captured data is serially shifted out and externally compared to the expected results. Forced test data is serially shifted into the boundary-scan cells. All of this is controlled from a serial data path called the scan path or scan chain. Figure 1 depicts the main elements of a boundary-scan cell. By allowing direct access to nets, boundary-scan eliminates the need for a large number of test vectors, which are normally needed to properly initialize sequential logic. Tens or hundreds of vectors may do the job that had previously required thousands of vectors. Potential benefits realized from the use of boundary-scan are shorter test times, higher test coverage, increased diagnostic capability and lower capital equipment cost.

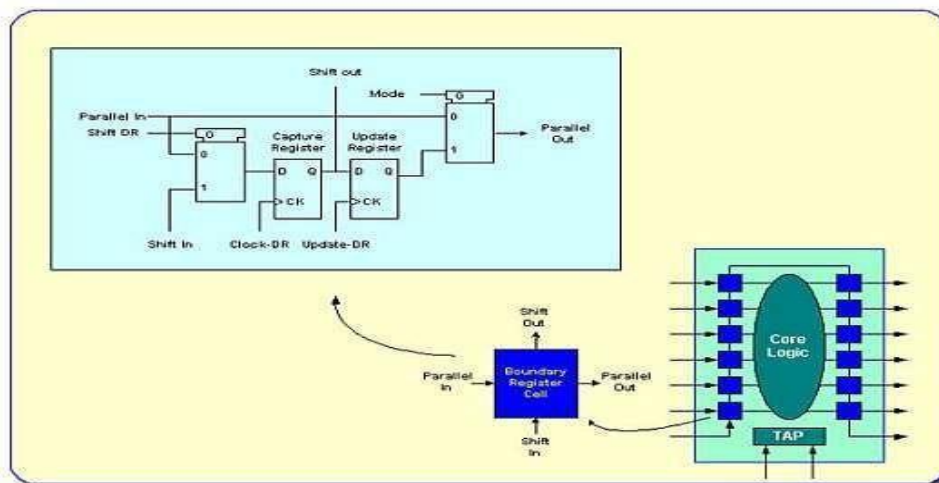


Figure 4: Typical Boundary-Scan

The principles of interconnect test using boundary-scan are illustrated in Figure 2. Figure 2 depicts two boundary-scan compliant devices, U1 and U2, which are connected with four nets. U1 includes four outputs that are

driving the four inputs of U2 with various values. In this case, we assume that the circuit includes two faults: a short between Nets 2 and 3, and an open on Net 4. We will also assume that a short between two nets behaves as a wired-AND and an open is sensed as logic 1. To detect and isolate the above defects, the tester is shifting into the U1 boundary-scan register the patterns shown in Figure 2 and applying these patterns to the inputs of U2. The inputs values of U2 boundary-scan register are shifted out and compared to the expected results. In this case, the results (marked in red) on Nets 2, 3, and 4 do not match the expected values and, therefore, the tester detects the faults on Nets 2, 3, and 4. Boundary-scan tool vendors provide various types of stimulus and sophisticated algorithms, not only to detect the failing nets, but also to isolate the faults to specific nets, devices, and pins.

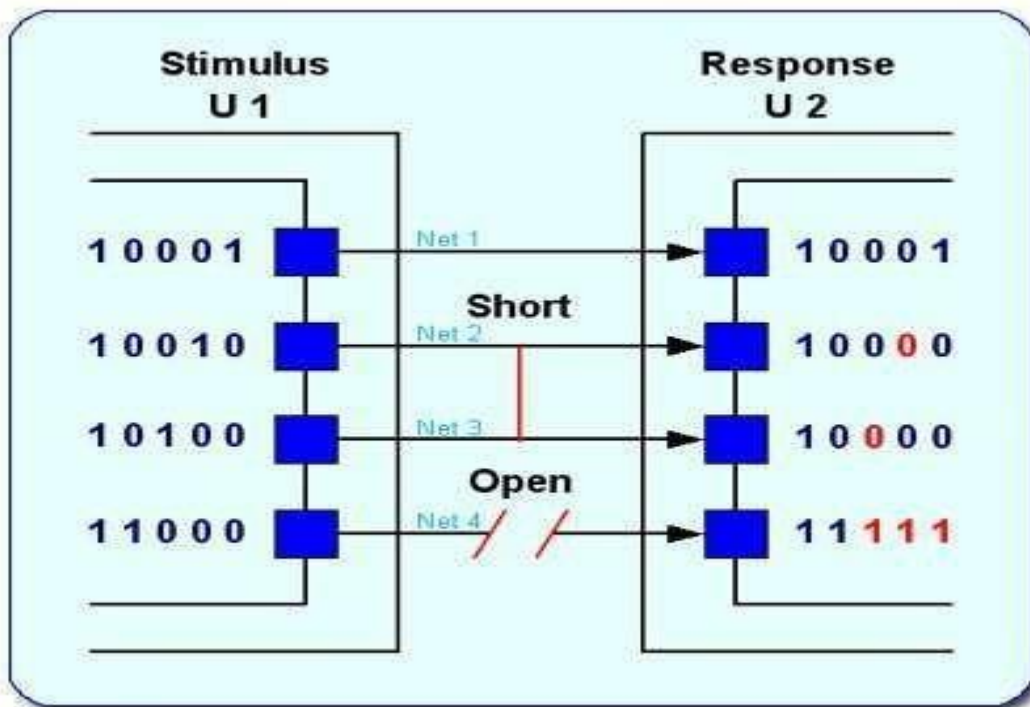


Figure 5: Interconnect Test Example

Boundary-Scan Chip Architecture

- Chain integrity testing
- Interconnection testing between devices
- Core logic testing (BIST)
- In-system programming
- In-Circuit Emulation
- Functional testing

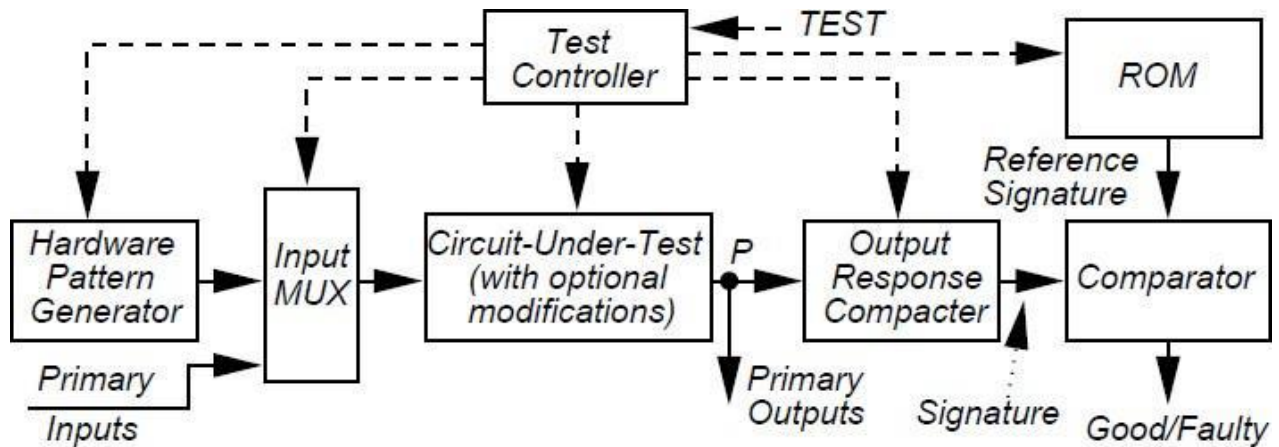
BIST Architecture:

Figure 15.2: BIST process.

As the complexity of individual VLSI circuits and as overall system complexity increase, test generation and application becomes an expensive, and not always very effective, means of testing. Further, there are also very difficult problems associated with the high speeds at which many VLSI systems are designed to operate. Such problems require the use of very sophisticated, but not always affordable, test equipments. Consequently, BIST objectives are:

1. to reduce test pattern generation costs;
2. to reduce the volume of test data;
3. to reduce test time.

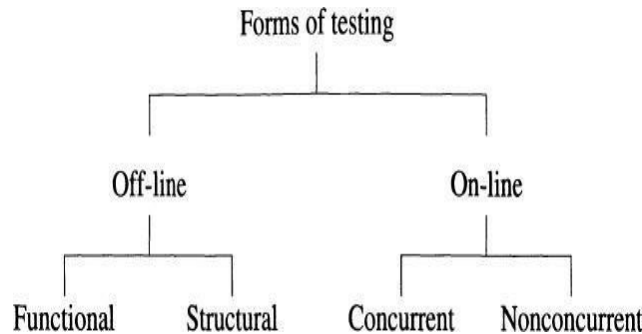
BIST techniques aim to effectively integrate an automatic test system into the chip design.

BUILT-IN SELF-TEST (BIST):

- ✓ Built-in self-test is the capability of a circuit (chip, board, or system) to test itself. BIST represents a merger of the concepts of built-in test (BIT) and self-test.
- ✓ BIST techniques can be classified into two categories, namely
 - i. **On-line BIST**, which includes **concurrent and non-concurrent techniques**,
 - ii. **Off-line BIST**, which includes **functional and structural approaches**.
- ✓ **In on-line BIST**, testing 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

on-line BIST, testing is carried out while a system is in an idle state. This is often accomplished by executing diagnostic software routines (macrocode) or diagnostic firmware routines (microcode). The test process can be interrupted at any time so that normal operation can resume.

- ✓ **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. This form of testing is also applicable at the manufacturing, field, depot, and operational levels. Often Off-line testing is carried out using on-chip or on-board test-pattern generators (TPGs) and output response analyzers (ORAs). Off-line testing does not detect errors in real time, i.e., when they first occur, as is possible with many on-line concurrent BIST techniques.



- ✓ **Functional off-line BIST** deals with the execution of a test based on a functional description of the CUT and often employs a functional, or high-level, fault model.
- ✓ **Structural off-line BIST** deals with the execution of a test based on the structure of the CUT.
- ✓ **Usually** tests are generated and responses are compressed using some form of an LFSR.

Off-Line BIST Architectures

Off-line BIST architectures at the chip and board level can be classified according to the following criteria:

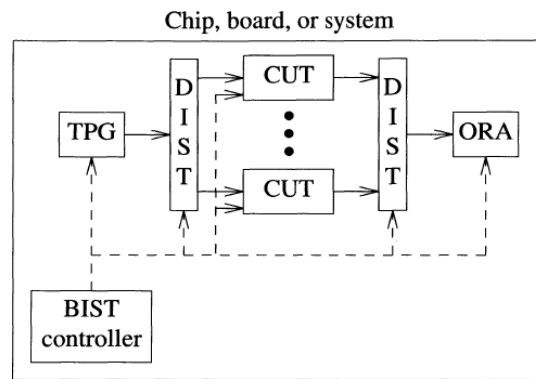
1. Centralized or distributed BIST circuitry;
2. Embedded or separate BIST elements.

BIST architectures consist of several key elements, namely

1. Test-pattern generators;
2. Output-response analyzers;
3. The circuit under test;
4. A distribution system (DIST) for transmitting data from TPGs to CUTs and from CUTs to ORAs;
5. BIST controller for controlling the BIST circuitry and CUT during self-test.

Centralized BIST architecture:

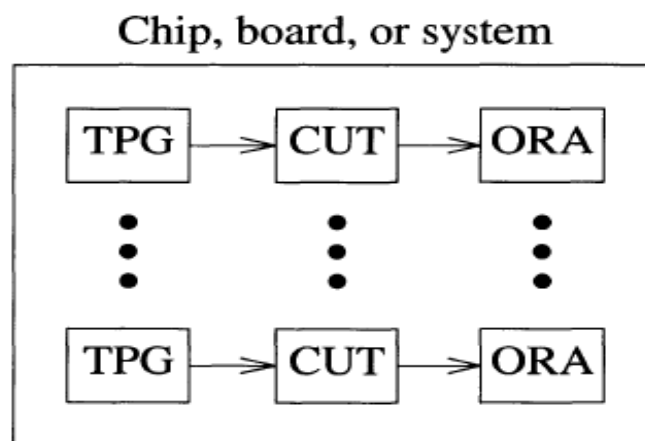
The general form of a centralized BIST architecture is shown in the below figure.



Here several CUTs 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 system clocks and control test clocks.
3. Communicate with other test controllers, possibly using test busses.
4. Control the operation of a self-test, including seeding of registers, keeping track of the number of shift commands required in a scan operation, and keeping track of the number of test patterns that have been processed.

Distributed BIST architecture:



The distributed BIST architecture is shown in above 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.

Advantages of BIST:

- Low cost
- High quality testing
- Faster fault detection
- Ease of diagnostics
- Reduce maintenance and repair cost

