Module 1

Introduction

The first integrated circuit was flip-flop with two transistors built by Jack Kilby at Texas Instruments in the year 1958. In the year 2008, Intel's Itanium microprocessor contained more than 2 billion transistors and a 16 Gb Flash memory contained more than 4 billion transistors. So in the range of over 50 years there is the growth rate is around 53%. This incredible growth has come from steady miniaturization of transistors and improvements in manufacturing processes. As transistors became smaller, they also became faster, dissipate less power, and are got cheaper to manufacture. The memory once needed for an entire company's accounting system is now carried by a teenager in her iPod. Improvements in integrated circuits have enabled space exploration, made automobiles safer and more fuel efficient, revolutionized the nature of warfare, brought much of mankind's knowledge to our Web browsers, and made the world a flatter place.

- During the first half of the twentieth century, electronic circuits used large, expensive, power-hungry, and unreliable vacuum tubes.
- In 1947, John Bardeen and Walter Brattain built the first functioning point contact transistor at Bell Laboratories, shown in Figure 1.1(a).
- Later it was introduced by the Bell Lab and named it as **Transistor, T-R-A-N-S-I-S-T-O-R**, because it is a resistor or semiconductor device which can amplify electrical signals as they are transferred through it from input to output terminals.
- Ten years later, Jack Kilby at Texas Instruments realized the potential for miniaturization if multiple transistors could be built on one piece of silicon. Figure 1.1(b) shows his first prototype of an integrated circuit, constructed from a germanium slice and gold wires.

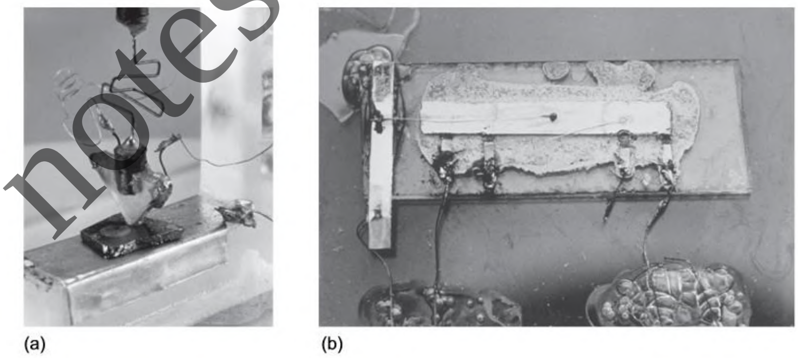


Fig. 1.1(a) First transistor (b) First Integrated Circuit

- Transistors are electrically controlled switches with a control terminal and two other terminals that are connected or disconnected depending on the voltage or current applied to the control.
- After the invention of point contact transistor, Bell Labs developed the bipolar junction transistor, which were more reliable, less noisy and more power-efficient.
- Early integrated circuits used mainly bipolar transistors, which required a small current into the control (base) terminal to switch much larger currents between the other two (emitter and collector) terminals.
- The problem seen with bipolar transistors were the power dissipated by the base current which limited the maximum number of transistors that can be integrated onto a single die.

- Then in 1960 came Metal Oxide Semiconductor Field Effect Transistors (MOSFETs). The advantages seen in MOSFETs were that they draw almost zero control current while idle. It was available in 2 forms as: nMOS and pMOS, using n-type and p-type silicon, respectively.
- In 1963, the first logic gates using MOSFETs was introduced at Fairchild. It included gates used both nMOS and pMOS transistors. This gave the name Complementary Metal Oxide Semiconductor, or CMOS. The circuits used discrete transistors but consumed only nanowatts of power, which was about six times lesser than bipolar transistors.
- MOS ICs became popular because of their low cost, each transistor occupied less area and the fabrication process was simpler. Early commercial processes used only pMOS transistors but it suffered from poor performance, yield, and reliability. Later on Processes using nMOS transistors became common in the 1970s.
- Even though nMOS process was less expensive compared to CMOS, nMOS logic gates consumed power while they were idle. Power consumption became a major issue in the 1980s as hundreds of thousands of transistors were integrated onto a single die. CMOS processes were widely adopted and have essentially replaced nMOS and bipolar processes for nearly all digital logic applications.
- In 1965, Gordon Moore observed that plotting the number of transistors that can be most economically manufactured on a chip gives a straight line on a semi-logarithmic scale. Also he found transistor count doubling every 18 months. This observation has been called **Moore's Law**.
    - Fig 1.2 shows that the number of transistors in Intel microprocessors has doubled every 26 months since the invention of the 4004.
    - Moore's Law is based on scaling down the size of transistors and to some extent building larger chips.

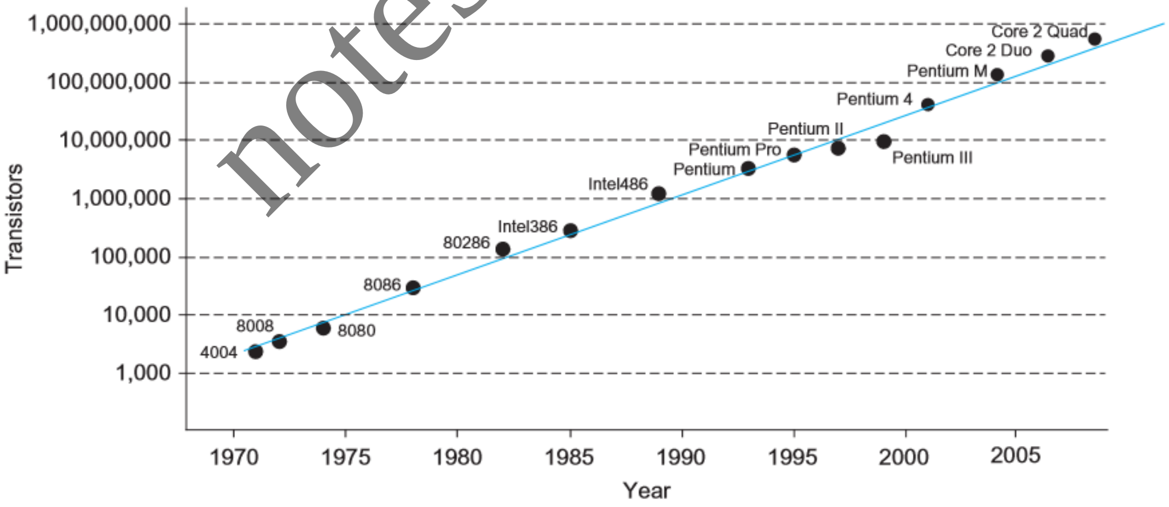


Fig 1.2 Transistors in Intel microprocessors

Level of Integration:

The process of integration can be classified as small, medium, large, very large.

1. Small-Scale Integration (SSI): The number of components is less than 10 in every package. Logic Gates like inverters, AND gate, OR gate and etc. are products of SSI.

2. Medium Scale Integration (MSI): MSI devices has a complexity of 10 to 100 electronic components in a single package. Ex: decoders, adders, counters, multiplexers, and demultiplexers.

3. Large Scale Integration (LSI): Products of LSI contain between 100 and 10,000 electronic components in a single package. Ex: memory modules, I/O controllers, and 4-bit microprocessor systems.

4. Very Large Scale Integration (VLSI): Devices that are results of VLSI contain between 10,000 and 300,000 electronic components. Ex: 8bit, 16-bit, and 32-bit microprocessor systems.

- The feature size of a CMOS manufacturing process refers to the minimum dimension of a transistor that can be reliably built. The 4004 had a feature size of 10µ m in 1971. The Core 2 Duo had a feature size of 45nm in 2008. Feature sizes specified in microns (10^{-6}m), while smaller feature sizes are expressed in nanometers (10^{-9} m).

MOS Transistor:

- Silicon (Si), a semiconductor, forms the basic starting material for most integrated circuits
- Silicon is a Group IV element in periodic table, it forms covalent bonds with four adjacent atoms, as shown in Figure 1.3(a). As the valence electrons of it are involved in chemical bonds, pure silicon is a poor conductor.
- However its conductivity can be increased by introducing small amounts of impurities, called dopants, into the silicon lattice.
- A dopant from Group V of the periodic table, such as arsenic, having five valence electrons. It replaces a silicon atom in the lattice and still bonds to four neighbors, so the fifth valence electron is loosely bound to the arsenic atom, as shown in Figure 1.3(b). Thermal vibration at room temperature is sufficient to free the electron. This results in As+ ion and a free electron. The free electron can carry current and this is an n-type semiconductor.

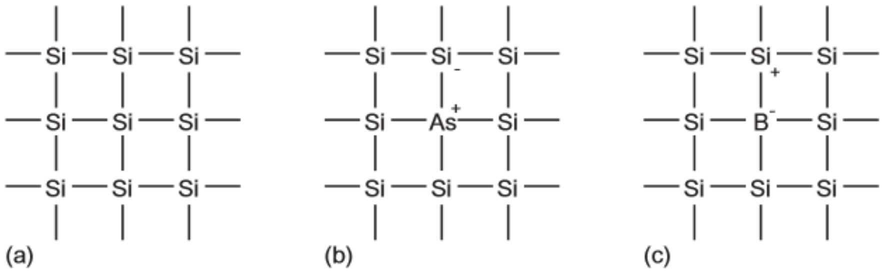


Fig 1.3 Silicon lattice and dopant atoms

- A Group III dopant, such as boron, having three valence electrons, as shown in Fig 1.3(c). The dopant atom can borrow an electron from a neighboring silicon atom, which in turn becomes short by one electron. That atom in turn can borrow an electron, and so forth, so the missing electron, or hole, can propagate about the lattice. The hole acts as a positive carrier so we call this a p-type semiconductor.
- A Metal-Oxide-Semiconductor (MOS) structure is created by superimposing several layers of conducting and insulating materials to form a sandwich-like structure.

- Transistors can be built on a single crystal of silicon, which are available as thin flat circular wafer of 15–30 cm in diameter. CMOS technology provides two types of transistors an n-type transistor (nMOS) and a p-type transistor (pMOS).
- Transistor operation is controlled by electric fields so the devices are also called Metal Oxide Semiconductor Field Effect Transistors (MOSFETs) or simply FETs. Cross-sections and symbols of these transistors are shown in Figure 1.4. The n+ and p+ regions indicate heavily doped n- or p-type silicon.

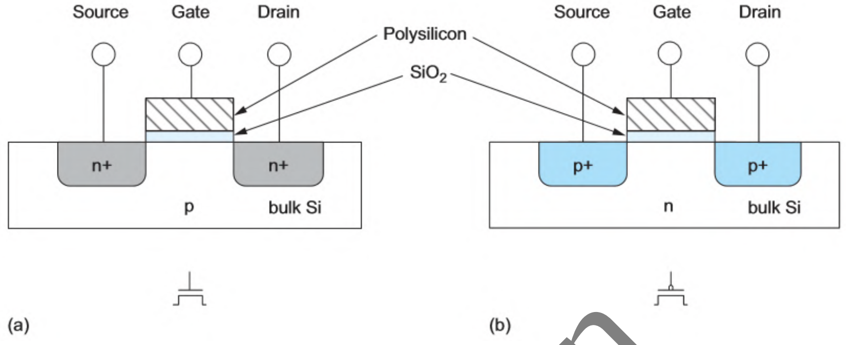


Fig 1.4 (a) nMOS transistor and (b) pMOS transistor

- Each transistor has conducting gate, an insulating layer of silicon dioxide (SiO2, also known as glass), and the silicon wafer, also called the substrate/body/bulk. Gates of early transistors were built from metal, so was called Metal-Oxide-Semiconductor, or MOS.
- Even though the gate has been formed from polycrystalline silicon (polysilicon), the name is still metal.
- An nMOS transistor is built with a p-type body and has regions of n-type semiconductor adjacent to the gate called the source and drain. They are physically equivalent and they can be interchangeable. The body is typically grounded.
- A pMOS transistor is just the opposite, consisting of p-type source and drain regions with an n-type body.
- In both the gate is the control input.
- nMOS Transistor:
    - It controls the flow of electrical current between the source and drain.
    - Considering an nMOS transistor, its body is generally grounded so the p–n junctions of the source and drain to body are reverse-biased. If the gate is also grounded, no current flows through the reverse-biased junctions and the transistor is OFF.
    - If the gate voltage is raised, it creates an electric field that starts to attract free electrons to the underside of the Si–SiO2 interface.
    - If the voltage is raised enough, the electrons outnumber the holes and a thin region under the gate called the channel is inverted to act as an n-type semiconductor.
    - Hence, a conducting path is formed from source to drain and current can flow. This is the condition for transistor is ON state.
    - Thus when the gate of an nMOS transistor is high, the transistor is ON and there is a conducting path from source to drain. When the gate is low, the nMOS transistor is OFF and almost zero current flows from source to drain.

- pMOS Transistor:
    - The condition is reversed.
    - The body is held at a positive voltage and also when the gate is at a positive voltage, the source and drain junctions are reverse-biased and no current flows, the transistor is OFF.

- When the gate voltage is reduced, positive charges are attracted to the underside of the Si–SiO2 interface. A sufficiently low gate voltage inverts the channel and a conducting path of positive carriers is formed from source to drain, so the transistor is ON.
- The symbol for the pMOS transistor has a bubble on the gate, indicating that the transistor behavior is the opposite of the nMOS.
- A pMOS transistor is just the opposite of that of nMOS. It is ON when the gate is low and OFF when the gate is high

Transistor symbols and switch-level models is shown in Fig 1.5

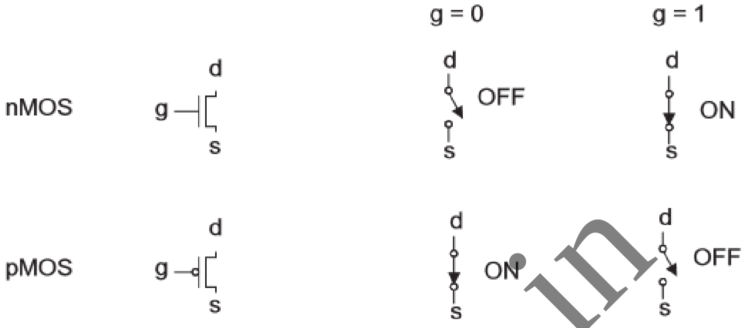


Fig 1.5 Transistor symbols and switch-level models

MOS Transistor Theory:

- MOS transistor is a majority-carrier device - current in channel between the source and drain is controlled by a voltage applied to the gate.
  - In nMOS transistor - majority carriers are electrons
  - In pMOS transistor - majority carriers are holes.
- To understand the behavior of MOS transistors, an isolated MOS structure with a gate and body but no source or drain is consider.
- It has top layer of good conducting gate layer. Middle layer is insulating oxide layer and bottom layer is the p-type substrate i.e doped silicon body. Since it is a p-type body carriers are holes

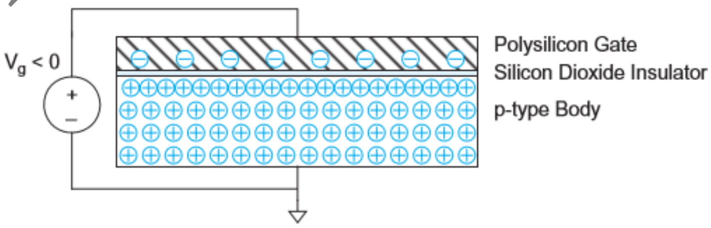


Fig 1.6 (a) Accumulation

- When a negative voltage is applied to the gate, the positively charged holes are attracted to the region beneath the gate. This is called the accumulation mode shown in Fig 1.6(a)
- When a small positive voltage is applied to the gate, the positive charge on the gate repels the holes resulting a depletion region beneath the gate as shown in Fig 1.6(b)

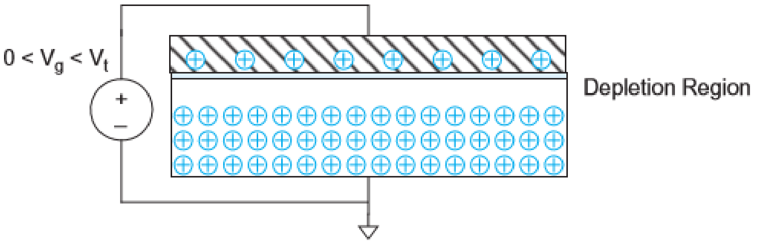


Fig 1.6(b) Depletion

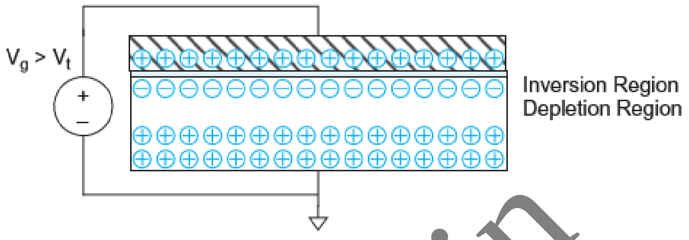


Fig 1.6(c) Inversion

- When a higher positive potential exceeding a critical threshold voltage Vt is applied, the holes are repelled further and some free electrons in the body are attracted to the region beneath the gate. This results a layer of electrons in the p-type body is called the inversion layer.

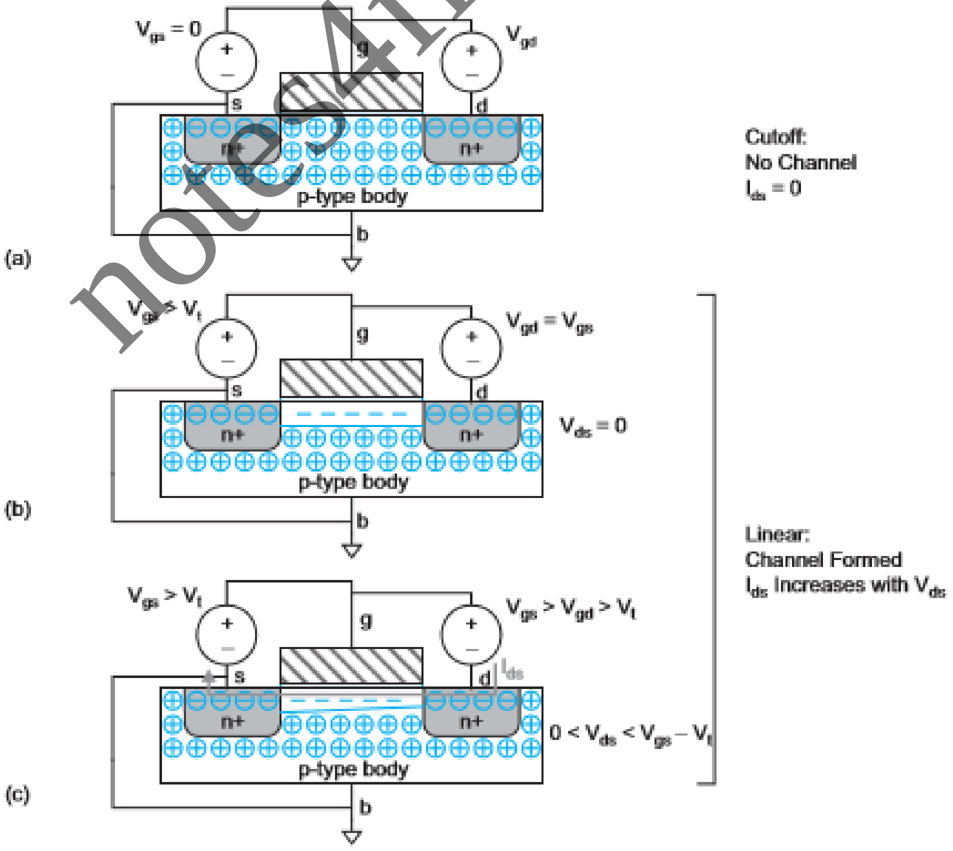


Fig 1.7 (a) nMOS demonstrating Cutoff and Linear operation

- Now considering transistor with MOS stack between two n-type regions called the source and drain the operation is considered.

- When gate-to-source voltage, Vgs is less than threshold voltage and if source is grounded, then the junctions between the body and the source or drain are zero-biased or reverse-biased and no current flows. We say the transistor is OFF, and this mode of operation is called **cutoff**. This is shown in above fig. 1.7(a)
- When the gate voltage is greater than the threshold voltage, an inversion region of electrons (majority carriers) called the channel connects the source and drain, creating a conductive path and turning the transistor ON Fig 1.7(b). The number of carriers and the conductivity increases with the gate voltage. The potential difference between drain and source is Vds= Vgs - Vgd. If Vds = 0 (i.e., Vgs =Vgd), there is no electric field tending to push current from drain to source. When a small positive potential Vds is applied to the drain, current Ids flows through the channel from drain to source. This mode of operation is termed **linear, resistive, triode, nonsaturated, or unsaturated** mode as shown in Fig 1.7 (c)
- If Vds becomes sufficiently large that Vgd < Vt, the channel is no longer inverted near the drain and becomes pinched off (Fig 1.7(d)). However, conduction is still brought about by the drift of electrons under the influence of the positive drain voltage. Above this drain voltage the current Ids is controlled only by the gate voltage and ceases to be influenced by the drain. This mode is called **saturation**.

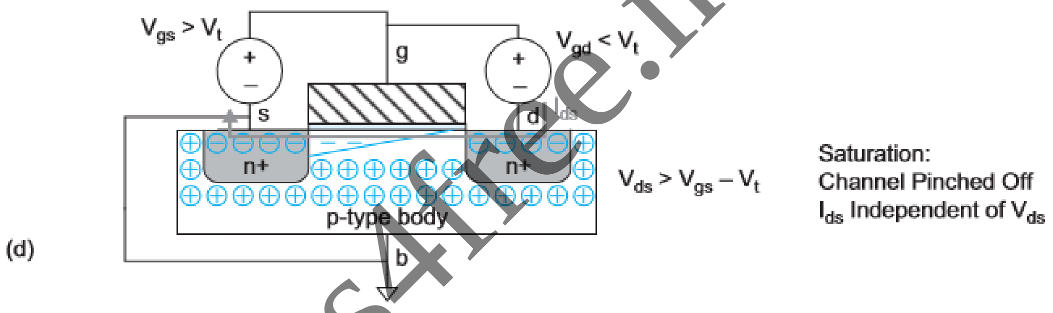


Fig 1.7 (d) Saturation

pMOS Transistor

- The pMOS transistor in Fig 1.8 operates in just the opposite fashion. The n-type body is tied to a high potential so the junctions with the p-type source and drain are normally reverse-biased. When the gate is also at a high potential, no current flows between drain and source. When the gate voltage is lowered by a threshold Vt, holes are attracted to form a p-type channel immediately beneath the gate, allowing current to flow between drain and source.

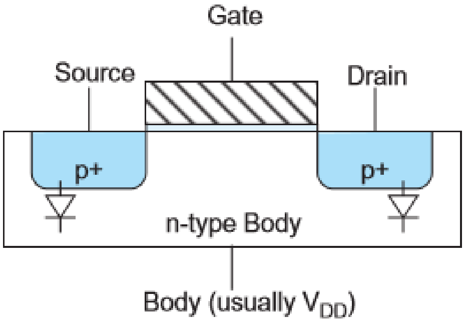


Fig 1.8 pMOS Transistor

Ideal I-V Characteristics:

- Considering Shockley model, which assumes the current through an OFF transistor is 0 i.e., when Vgs < Vt there is no channel and current from drain to source is 0.

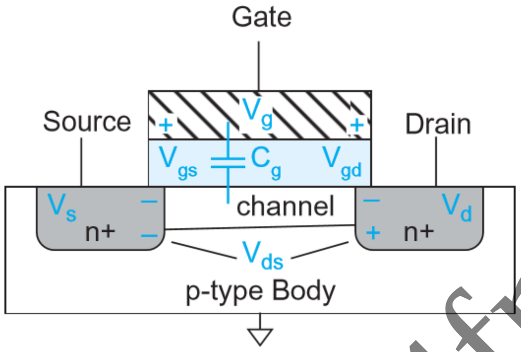
- In other 2 regions (linear and saturation) channel is formed and electrons flow from source to drain at a rate proportional to electric field (field between source and drain)
- If the amount of charge in the channel and the rate at which it moves is known, we can determine the current.
- The charge on parallel plate of capacitor is given by, Q = C.V
- Here the charge in the channel is denoted by Qchannel and is given by

  Qchannel = Cg . Vc

  Where Cg – capacitance of gate to the channel

  Vc – amount of voltage attracting charge to the channel
- If we model the gate as a parallel plate capacitor, then capacitance is given by

  Area/Thickness



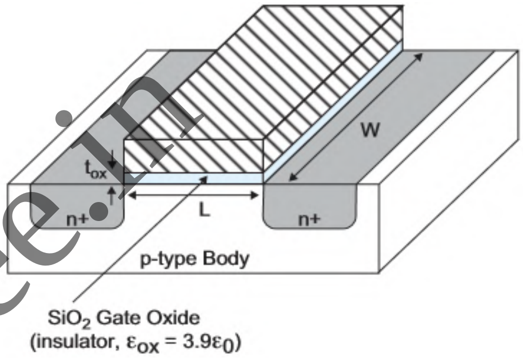


Fig a. Capacitance effect at the gate terminal

Fig b. Transistor dimensions

- If gate is having length L and width W and the oxide thickness is tox, as shown in Fig b, the capacitance is given by

$$Cg = \frac{Ɛox\ W\ L}{tox}$$

Where Ɛox is the permittivity of oxide and it is 3.9 Ɛo.

Ɛo is permittivity of free space, 8.85 × 10^{-14} F/cm,

- Often, the Ɛox/tox term is called Cox, the capacitance per unit area of the gate oxide.
- Thus capacitance is now Cg = Cox W L
- Now the charges induced in channel due to gate voltage is determined by taking the average voltage between source and drain (Fig. a) and it is given by

  Vc = (Vs + Vd)/2

To form the channel and carriers to flow, the voltage condition at source and drain is as follows:

Vs = Vgs – Vt

Vd = (Vgs – Vt) – Vds

Thus average voltage is now

$$Vc = \frac{(Vgs - Vt) + (Vgs - Vt) - Vds}{2}$$

Upon simplification, Vc is now

Vc = (Vgs –Vt) – Vds/2

Thus Qchannel = C_{ox}WL[(Vgs –Vt) – Vds/2]

- The velocity of charge carrier in the channel is proportional to lateral electric field (field between source and drain) and it is given by,

  $$v = \mu E$$

  Where μ is the proportionality constant called 'mobility'
- The electric field E is the voltage difference between drain and source to the length of channel. Given by,

  $$E = \frac{Vds}{L}$$
- The current in the channel is given by the total amount of charge in channel and time taken by them to cross. The time taken is given by length to velocity.

  i.e., $$Ids = \frac{total\ charge}{time\ to\ cross\ channel} = \frac{Qchannel}{L/v}$$

  $$Ids = \frac{Cg.Vc}{L}\ v = \frac{Cg.Vc}{L}\ \mu E$$

  $$Ids = \frac{Cg.Vc}{L}\ \mu\left(\frac{Vds}{L}\right)$$

  $$Ids = \frac{Cox\ W\ L\ [(Vgs - Vt) - \frac{Vds}{2}]}{L}\ \mu(\frac{Vds}{L})$$

  Upon simplification, Ids is given by:

  $$Ids = \mu\ Cox\frac{W}{L}\left[(Vgs - Vt) - \frac{Vds}{2}\right]Vds$$

  $$Ids = \beta\ \left[(Vgs - Vt) - \frac{Vds}{2}\right]Vds$$

  Where β = $\mu\ Cox\frac{W}{L}$
- The above equation for current describes linear region operation for Vgs > Vt
- When Vds is increased to larger value i.e., Vds > Vsat = Vgs – Vt, the channel is no longer inverted and at the drain channel gets pinched off.
- Beyond this is the drain current is independent of Vds and depends only on the gate voltage called as saturation current.
- The expression for the saturation current is given by

  $$Ids = \mu\ Cox\frac{W}{L}\ \left[(Vgs - Vt) - \frac{Vds}{2}\right]Vds$$

  $$Ids = \mu\ Cox\frac{W}{L}\ \left[(Vgs - Vt) - \frac{(Vgs-Vt)}{2}\right](Vgs - Vt)$$

  $$Ids = \mu\ Cox\frac{W}{L}\ \left[\frac{(Vgs-Vt)}{2}\right](Vgs - Vt)$$

  $$Ids = \beta/2\ (Vgs - Vt)^2$$

  Where β = $\mu\ Cox\frac{W}{L}$

Summarizing the currents in all the 3 regions is

Ids = 0 for Vgs < Vt cutoff

$Ids = \beta \left[(Vgs - Vt) - \frac{Vds}{2}\right] Vds$ for Vds < (Vgs-Vt) linear region

$Ids = \beta \left[(Vgs - Vt) - \frac{Vds}{2}\right] Vds$ for Vds > (Vgs-Vt) saturation region

The plot of current and voltage i.e., I-V Characteristics is shown in the fig.

pMOS Transistor:

pMOS transistors behave in the same way, but with the signs of all voltages and currents reversed. The I-V characteristics are in the third quadrant, as shown in Fig.

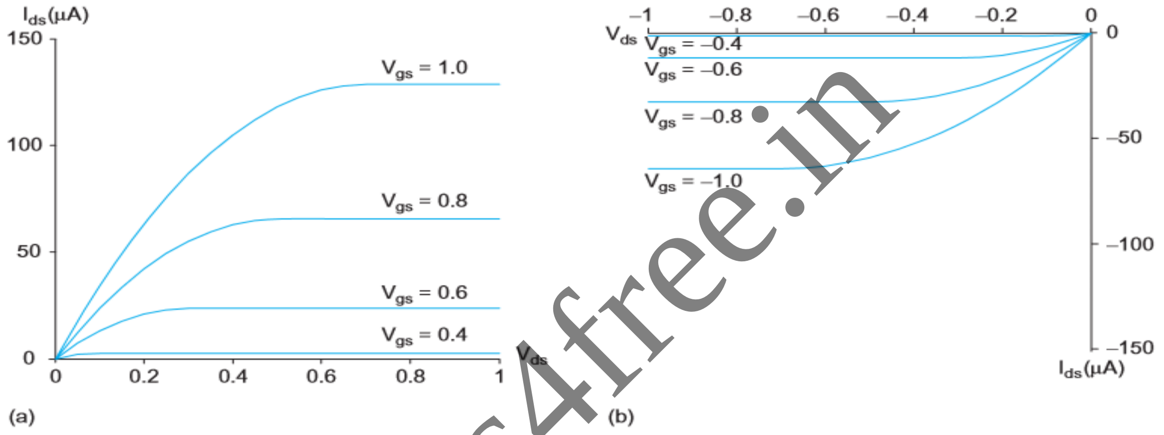


Fig. Plot of I-V characteristics of (a) nMOS and (b) pMOS

Non ideal I-V Effects:

- The ideal I-V model does not consider many effects that are important to modern devices. These effects are as follows:

Velocity saturation:

- Electron velocity is related to electric field through mobility by the equation
  v = μ E , where E is the lateral electric field or field between drain and source.
- It is assumed that μ is constant and independent parameter w.r.t, E
- At higher E, μ is no more constant and it varies and is due to velocity saturation effect
- When electric field reaches a critical value say E_{sat}, the velocity of charge carriers tend to saturate due to scattering effect at E_{sat}. This is shown in graph below.
- The impact of velocity saturation is modelled as follows:

  Before the velocity reaches critical value,

$$v = \frac{\mu\, Elat}{1+\frac{Elat}{Esat}}$$

When the velocity reaches critical and greater it is given by,

$$v = Vsat$$

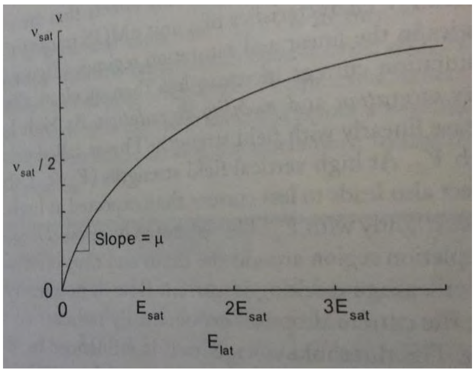


Fig. carrier velocity vs electric field

- When transistor is not velocity saturated, current Ids is given by

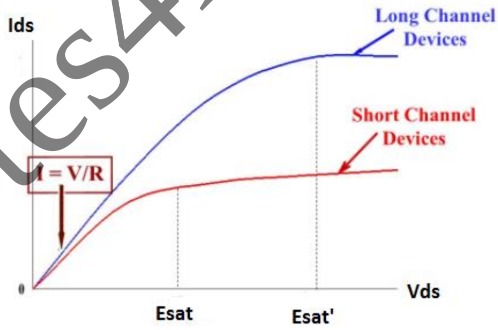
$$Ids = \mu Cox \frac{W}{L}\frac{(Vgs - Vt)^2}{2}$$

and with velocity saturation, current Ids is given as

$$Ids = Cox\, W\, (Vgs - Vt)Vsat$$

Observing both the expression we can say that
Ids depend quadratically on voltage without saturation and depends linearly when fully saturated

- As shown in graph for short channel devices it has extended saturation region (from Esat to Esat') due to velocity saturation.



- As channel length becomes shorter, lateral electrical field increases and transistor becomes more velocity saturated and this decreases drain current Ids.

Mobility degradation:

- Velocity of charge carriers depend on electric field and when these carriers travel along the length of channel, they get attracted to the surface (i.e., Gate) by the vertical electric field (field created by gate voltage)
- Hence they bounce against the surface during their travel
- This reduces surface mobility in comparison with the mobility along the channel.
- This is known as mobility degradation and has an impact on I-V characteristics.
- As mobility decreases the current also decreases.

Channel length Modulation:

- Ideally drain current Ids is independent on Vds in the saturation region making transistor a perfect current source.

- When Vds is increased further, near the drain barrier is build due to depletion region and reduces the length of the channel.
- This results in reducing the length of the channel by L_d. This is shown in Fig below. Thus in saturation the effective channel length is modelled as

$$L_{eff} = L - L_d$$

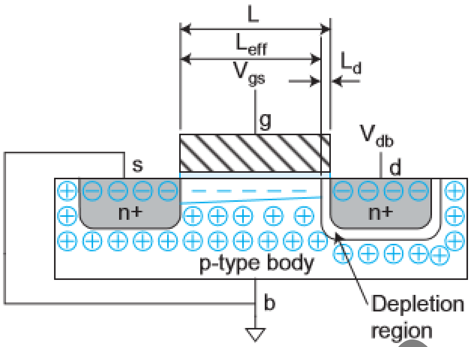


Fig. Channel length modulation in saturation mode

- To avoid introducing the body voltage into our calculations, assume the source voltage is close to the body voltage so Vdb ~ Vds. Hence, increasing Vds decreases the effective channel length. Shorter channel length results in higher current; thus, Ids increases with Vds in saturation
- This is modeled as

$$Ids = \frac{\beta}{2}(Vgs - Vt)^2\,(1 + \lambda\,Vds)$$

Where λ is empirical channel length modulation factor

- The equation can also be written as $Ids = \frac{\mu Cox}{2}\frac{W}{L}\,(Vgs - Vt)^2\,(1 + \lambda\,Vds)$

  Thus as L decreases W/L ratio increases, this in turn increases Ids. Thus transistor in saturation is no more a constant current source.

Note: Channel length modulation is important in analog designs as it reduces the gain of the amplifier. But for digital circuits channel length modulations has no much importance.

Body Effect:

- MOSFETs have 4th implicit terminal called body/substrate along with gate, source and drain.
- The threshold voltage Vt which is assumed to remain constant is no more a constant value and varies as potential between source and body is varied. This effect is called body effect.
- The variation in the threshold voltage is modeled by the equation

$$V_t = V_{t0} + \gamma\left(\sqrt{\phi_s + V_{sb}} - \sqrt{\phi_s}\right)$$

Where Vto is the threshold voltage when source and body are at same potential
Φ_s is the surface potential
γ is the body effect coefficient and these two are given by
Vsb is the source to body potential

$$\phi_s = 2\upsilon_T \ln\frac{N_A}{n_i}$$

$$\gamma = \frac{t_{\text{ox}}}{\varepsilon_{\text{ox}}}\sqrt{2q\varepsilon_{\text{si}}N_A} = \frac{\sqrt{2q\varepsilon_{\text{si}}N_A}}{C_{\text{ox}}}$$

υ_T is voltage at room temperature (υ_T = KT/q at 30^0 it is 26mV)
N_A is the doping concentration level
n_i is the intrinsic carrier concentration
q is charge (q = 1.6 ×10^{-19} C)
tox oxide thickness
Ɛox is the permitivity of oxide and is given by 3.9 Ɛo, where Ɛo is the permittivity of free space = 8.825×10^{-14} F/cm
Ɛsi is the permittivity of silicon and given by 11.7 Ɛo and Ɛo=8.825×10^{-14} F/cm

- Body effect parameter γ depends on doping level concentration, thus by varying γ threshold voltage can be varied
- Also Vt depend on Vsb thus by proving appropriate potential threshold voltage can be varied.
- Thus a proper body bias can intentionally be applied to alter the threshold voltage, permitting trade-offs between performance and subthreshold leakage current

Subthreshold Conduction:

- The ideal I-V model assumes current flows from source to drain only when Vgs >Vt (when gate voltage is high). But in practical transistors, current does not abruptly cut off below threshold, but rather drops off exponentially.
- This regime of Vgs <Vt is called weak inversion/ subthreshold.
- This conduction of current is known as leakage and is undesired when the transistor is off
- The subthreshold conduction is modeled using equation given below

$$Ids = Idso\ e^{\frac{Vgs-Vt}{n\,VT}}\left[1 - e^{\frac{-Vds}{VT}}\right]$$

and $$I_{ds0} = \beta\upsilon_T^2 e^{1.8}$$

I_{dso} is the current at saturation and is dependent on process and device geometry
Vt is the threshold voltage and υ_T voltage at room temperature.

- In the expression Ids is 0 if V_{ds} is 0 and increases to full when V_{ds} is few multiples of υ_T
- Graph shows conduction in the subthreshold region

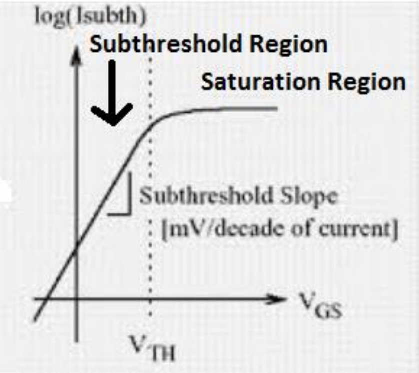


Fig. Subthreshold conduction

- Subthreshold conduction is useful for designing low power analog circuits and dynamic circuits as it reduces threshold voltage and results in low power consumptions.

Drain Induced Barrier Lowering (DIBL):

- As the drain voltage Vds is increased it creates an electric field that affects the threshold voltage.
- This effect is called drain-induced barrier lowering (DIBL) and this effect is especially pronounced in short-channel transistors.
- As the channel length decreases, the DIBL effect shows up and the variation caused in the threshold voltage can be modeled as

$$V_t = V_{t0} - \eta V_{ds}$$

η is the DIBL coefficient

Junction Leakage:

- The MOS structure is considered there exists p–n junctions between diffusion and the substrate. With CMOS structures p–n junctions between diffusion and the substrate or well, forming diodes, as shown in Fig. The well-to-substrate junction is another diode.

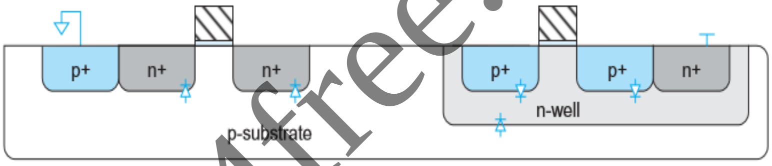


Fig. CMOS structure showing formation of p-n junctions between diffusion and substrate and also between well and substrate

- The substrate and well are tied to GND or V_{DD} so that these diodes does not get into forward biased condition until voltage is applied in normal operation.
- But in reverse-biased conditions these diodes still conduct a small amount of current I_D. This leakage current is modeled using equation

$$I_D = I_S\left(e^{\frac{V_D}{v_T}} - 1\right)$$

Where, I_D is the diode current
I_S is the diode reverse bias saturation current
V_D is the diode voltage (either Vsb or Vdb)

- I_S depends on doping levels and on the area and perimeter of the diffusion region (geometry) and V_D
- Leakage current usually lies in the range of 0.1 – 0.01 fA/μm^2, which is negligible when compared to other leakage currents.

Tunneling (Focoler Nordheium Tunneling):

- According to quantum mechanics, for thinner gate oxides there is a nonzero probability that an electron in the gate will find itself on the other side of the oxide, (i.e., in the region below gate/ channel).
- This effect of carriers crossing a thin barrier is called tunneling, and results in leakage current through the gate called gate leakage current.
- Thus gate oxide cannot be considered as an ideal insulator. This effects the circuit functionality and increases power consumption due to static gate current.

- Fig shows plot of gate leakage current density J_G against voltage for different oxide thickness. It can be observed that as oxide thickness decreases the leakage current density increases.

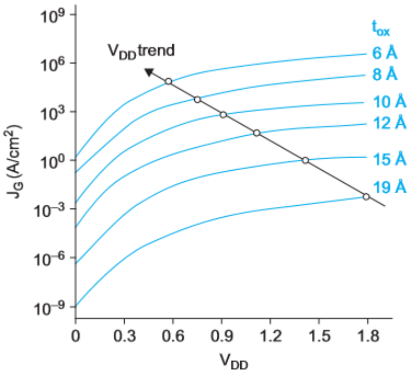


Fig. plot of gate leakage current density vs voltage for different t_{ox}

- Research is going on in finding an alternate to silicon dioxide and silicon nitrate is one contender for this.

Note: As mobility of electrons is more than holes in silicon, tunneling current magnitude for nMOS is more compared pMOS.

Temperature Dependence:

- Transistor characteristics are influenced by temperature
  - Carrier mobility – decreases with temperature and this is modeled using the relation

$$\mu(T) = \mu(T_r)\left(\frac{T}{T_r}\right)^{-k_\mu}$$

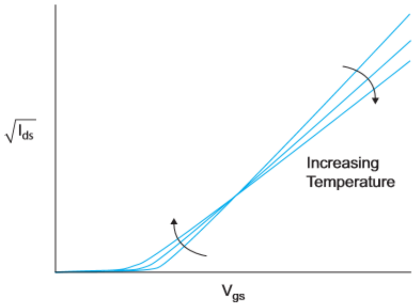
where T is absolute temperature, Tr is room temperature, k_μ is fitting constant.

  - Threshold voltage – magnitude of threshold voltage decreases linearly with temperature and can be modeled as

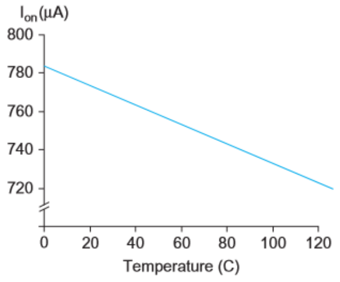
$$V_t(T) = V_t(T_r) - k_{vt}(T - T_r)$$

where k_{vt} is typically about 1–2 mV/K.

  - Junction Leakage – increases with temperature because Is (diode reverse bias current) strongly depends on temperature
  - Velocity saturation – occurs sooner with temperature.
  - With increase in temperature drain current decreases with temperature when transistor is ON and when transistor is OFF, the junction leakage and subthreshold conduction contribute to leakage current and this increase. This condition is shown in the graph.



I–V characteristics of nMOS transistor in saturation at various temperatures



I_{dsat} vs. temperature

- However, the circuit performance can be improved by providing cooling systems like heat sinks, water cooling, thin film refrigerator and liquid nitrogen.
- Advantages of using at lower temperatures are
  1. Leakages can be reduced
  2. With lower temperature, reducing threshold voltage it can be used in power saving
  3. Most wear out mechanisms are temperature dependent and if used at lower temp they are more reliable

Geometry Dependence:

- The layout designer would draws transistors with width and length W_{drawn} and L_{drawn}.
- While mask preparation the actual gate dimensions may differ by X_W and X_L.
- While diffusion process, the source and drain would tend to diffuse laterally under the gate by L_D, causing a smaller effective channel length that the carriers must traverse between source and drain. Similarly, W_D accounts for smaller width while diffusion.
- Combing all these factors transistor, lengths and widths that should be used in place of L and W is given by

$$L_{\text{eff}} = L_{\text{drawn}} + X_L - 2L_D$$
$$W_{\text{eff}} = W_{\text{drawn}} + X_W - 2W_D$$

- If there is variations in the length and width of the transistor there will be variations in the performance. For example, if the currents have to be matched then length should not be varied.

DC Transfer Characteristics

- DC transfer characteristics of a circuit relate the output voltage to the input voltage, assuming the input changes slowly enough that capacitances have plenty of time to charge or discharge,
  CMOS Inverter Static Characteristics

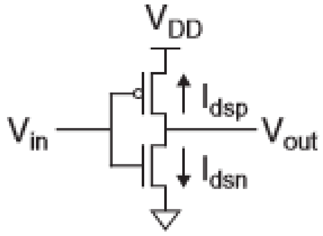
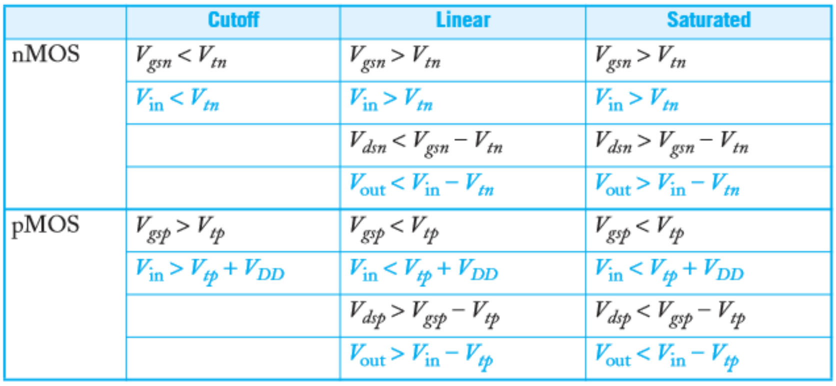


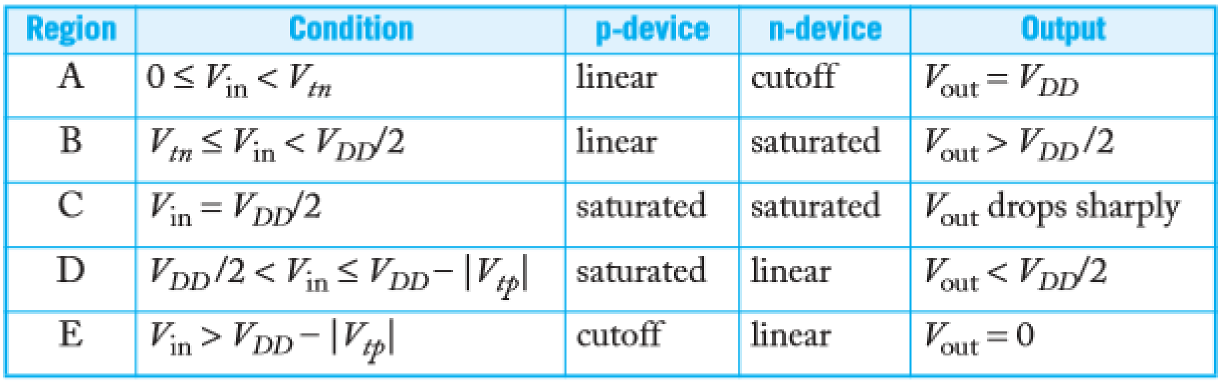
Fig 1.9 CMOS Inverter

CMOS inverter shown in Fig 1.9. Table below outlines various regions of operation for the n- and p-transistors. In this table, Vtn is the threshold voltage of the n-channel device, and Vtp is the threshold voltage of the p-channel device. Note that Vtp is negative. The equations are given both in terms of Vgs/Vds and Vin/Vout. As the source of the nMOS transistor is grounded, Vgsn = Vin and Vdsn = Vout. As the source of the pMOS transistor is tied to V_{DD}, Vgsp =Vin – V_{DD} and Vdsp =Vout – V_{DD}.



	Cutoff	Linear	Saturated
nMOS	$V_{gsn} < V_{tn}$	$V_{gsn} > V_{tn}$	$V_{gsn} > V_{tn}$
	$V_{\text{in}} < V_{tn}$	$V_{\text{in}} > V_{tn}$	$V_{\text{in}} > V_{tn}$
		$V_{dsn} < V_{gsn} - V_{tn}$	$V_{dsn} > V_{gsn} - V_{tn}$
		$V_{\text{out}} < V_{\text{in}} - V_{tn}$	$V_{\text{out}} > V_{\text{in}} - V_{tn}$
pMOS	$V_{gsp} > V_{tp}$	$V_{gsp} < V_{tp}$	$V_{gsp} < V_{tp}$
	$V_{\text{in}} > V_{tp} + V_{DD}$	$V_{\text{in}} < V_{tp} + V_{DD}$	$V_{\text{in}} < V_{tp} + V_{DD}$
		$V_{dsp} > V_{gsp} - V_{tp}$	$V_{dsp} < V_{gsp} - V_{tp}$
		$V_{\text{out}} > V_{\text{in}} - V_{tp}$	$V_{\text{out}} < V_{\text{in}} - V_{tp}$

- The objective is to find the variation in output voltage (Vout) as a function of the input voltage (Vin). This may be done graphically, for simplicity, we assume Vtp = –Vtn and that the pMOS transistor is 2–3 times as wide as the nMOS transistor so βn = βp.
- The plot shows Idsn and Idsp in terms of Vdsn and Vdsp for various values of Vgsn and Vgsp using drain current equation.
- Fig 1.10(b) shows the same plot of Idsn and |Idsp| now in terms of Vout for various values of Vin. The possible operating points of the inverter, marked with dots, are the values of Vout where Idsn = |Idsp| for same Vin.
- These operating points are plotted on Vout vs. Vin axes in Fig. (c) to show the inverter DC transfer characteristics.
- The supply current I_{DD} = Idsn = |Idsp| is also plotted against Vin in Fig (d) showing that both transistors are momentarily ON as Vin passes through voltages between GND and V_{DD}, resulting in a pulse of current drawn from the power supply.
- The operation of the CMOS inverter can be divided into five regions indicated on Fig 1.10(c). The state of each transistor in each region and state of output is shown in Table 2.
  - In region A, the nMOS transistor is OFF so the pMOS transistor pulls the output to V_{DD}.
  - In region B, the nMOS transistor starts to turn ON, pulling the output down.
  - In region C, both transistors are in saturation.
  - In region D, the pMOS transistor is partially ON
  - In region E, pMOS is completely OFF, leaving the nMOS transistor to pull the output down to GND.



Region	Condition	p-device	n-device	Output
A	$0 \le V_{\text{in}} < V_{tn}$	linear	cutoff	$V_{\text{out}} = V_{DD}$
B	$V_{tn} \le V_{\text{in}} < V_{DD}/2$	linear	saturated	$V_{\text{out}} > V_{DD}/2$
C	$V_{\text{in}} = V_{DD}/2$	saturated	saturated	V_{out} drops sharply
D	$V_{DD}/2 < V_{\text{in}} \le V_{DD} - \lvert V_{tp}\rvert$	saturated	linear	$V_{\text{out}} < V_{DD}/2$
E	$V_{\text{in}} > V_{DD} - \lvert V_{tp}\rvert$	cutoff	linear	$V_{\text{out}} = 0$

Table 2. Summary of CMOS Inverter Operation

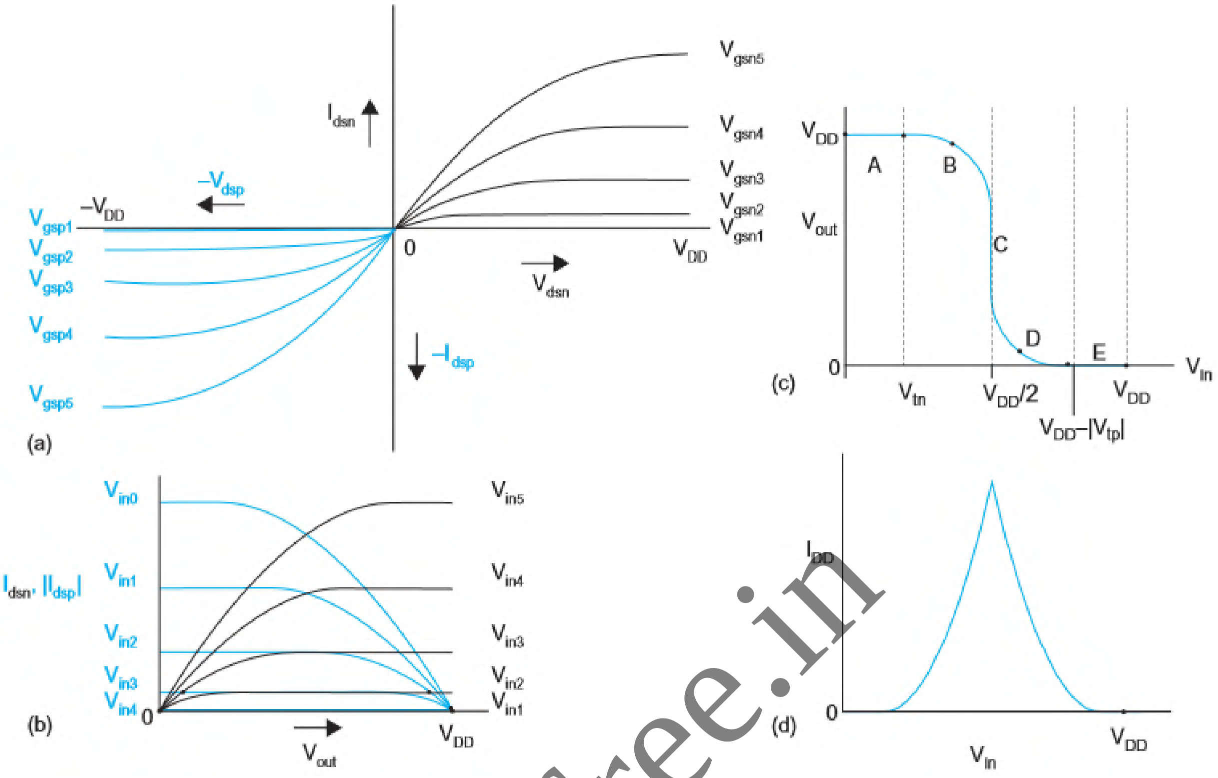
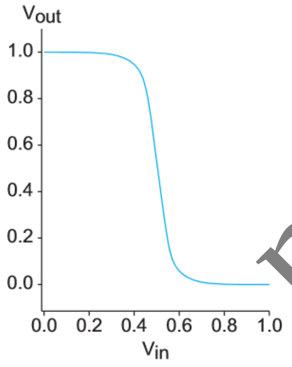


Fig 1.10 Graphical Derivation of CMOS Inverter DC Characteristics



In the fig. the crossover point where Vin = Vout, is called the **‘input threshold’**

Fig. CMOS inverter Transfer Characteristics

Beta Ratio Effects:

- We have seen that for βn = βp the inverter threshold voltage Vinv is $V_{DD}/2$. This may be desirable because it maximizes noise margins.
- Inverters with different beta ratios βp/βn are called skewed inverters. If βp/βn > 1, the inverter is HI-skewed. If βp/βn < 1, the inverter is LO-skewed. If βp/βn = 1, the inverter has normal skew or is unskewed.
- A HI-skew inverter has a stronger pMOS transistor. Therefore, if the input is $V_{DD}/2$, we would expect the output will be greater than $V_{DD}/2$.
- LO-skew inverter has a weaker pMOS transistor and thus a lower switching threshold.
- Figure explores the impact of skewing the beta ratio on the DC transfer characteristics. As the beta ratio is changed, the switching threshold moves. However, the output voltage transition remains sharp. Gates are usually skewed by adjusting the widths of transistors while maintaining minimum length for speed.

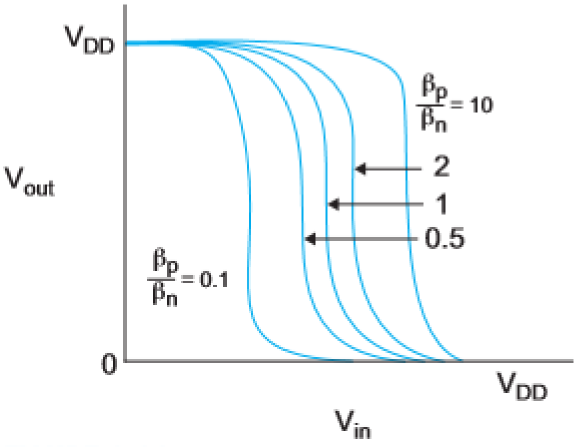


Fig. Transfer Characteristics of Skewed Inverters

Noise Margin:

- Noise margin is closely related to the DC voltage characteristics. This parameter allows you to determine the allowable noise voltage on the input of a gate so that the output will not be corrupted.
- The specification most commonly used to describe noise margin (or noise immunity) uses two parameters: the LOW noise margin, NM_L, and the HIGH noise margin, NM_H.
- With reference to Fig1.12, NM_L is defined as the difference in maximum LOW input voltage recognized by the receiving gate and the maximum LOW output voltage produced by the driving gate.

$$NM_L = V_{IL} - V_{OL}$$

- Similarly NM_H is the difference between the minimum HIGH output voltage of the driving gate and the minimum HIGH input voltage recognized by the receiving gate.

$$NM_H = V_{OH} - V_{IH}$$

Where V_{IH} = minimum HIGH input voltage
V_{IL} = maximum LOW input voltage
V_{OH}= minimum HIGH output voltage
V_{OL}= maximum LOW output voltage

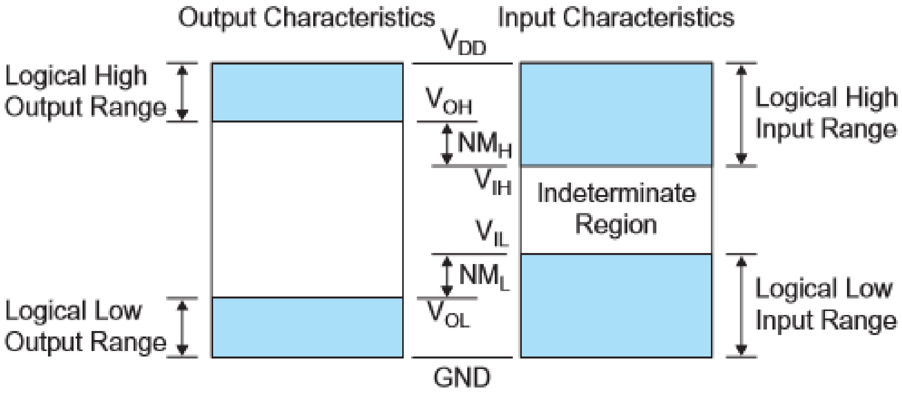


Fig. Noise Margin Definitions

- Inputs between V_{IL} and V_{IH} are said to be in the indeterminate region or forbidden zone and do not represent any legal digital logic levels. Therefore, it is generally desirable to have V_{IH} as close as possible to V_{IL} and for this value to be midway in the "logic swing,"

V_{OL} to V_{OH}. This implies that the transfer characteristic should switch abruptly; that is, there should be high gain in the transition region.

- DC analysis gives us the static noise margins specifying the level of noise that a gate may see for an indefinite duration.

Pass Transistor DC characteristics:

- nMOS transistors pass ‘0’s well but 1s poorly. Figure (a) shows an nMOS transistor with the gate and drain tied to V_{DD}. Imagine that the source is initially at Vs = 0. Vgs > Vtn, so the transistor is ON and current flows. If the voltage on the source rises to Vs = V_{DD} – Vtn, Vgs falls to Vtn and the transistor cuts itself OFF.
- Therefore, nMOS transistors attempting to pass a 1 never pull the source above V_{DD} – Vtn. This loss is sometimes called a threshold drop.
- Similarly, pMOS transistors pass 1s well but 0s poorly. If the pMOS source drops below |Vtp|, the transistor cuts off. Hence, pMOS transistors only pull down to within a threshold above GND, as shown in Fig (b).
- As the source can rise to within a threshold voltage of the gate, the output of several transistors in series is no more degraded than that of a single transistor Fig (c).
- However, if a degraded output drives the gate of another transistor, the second transistor can produce an even further degraded output Fig(d).

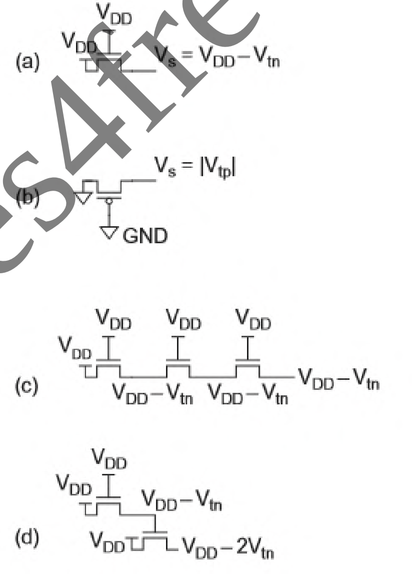


Fig. Pass Transistor Threshold drop

- The problem seen with nMOS and pMOS of not passing strong 1’s and strong 0’s respectively can be overcome by using Transmission gate.
- It has an nMOS and pMOS connected in parallel as shown in fig below.

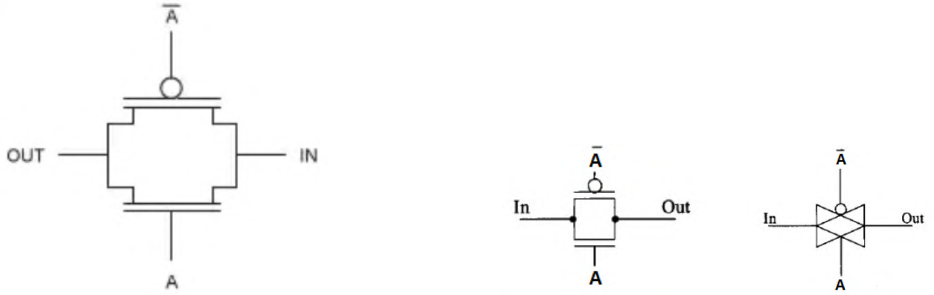


Fig. Schematic and symbol of Transmission gate (TG)

- When A is logic high both transistors are ON and TG is said to be ON. When input is provided as nMOS is not able to transmit strong 1, pMOS will do the function. Similarly when pMOS is not able to transmit strong 0, nMOS will do this function.
- Thus transmission gate is able to send both strong 0 and strong 1 without any signal degradation.
- Transmission gate can be used as
  - Multiplexing element
  - Analog switch
  - Latch element

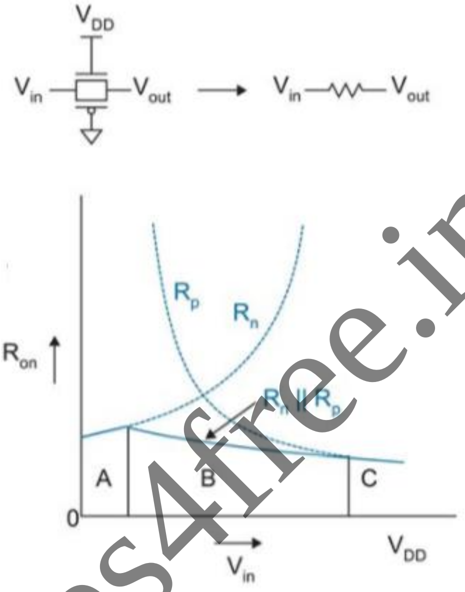
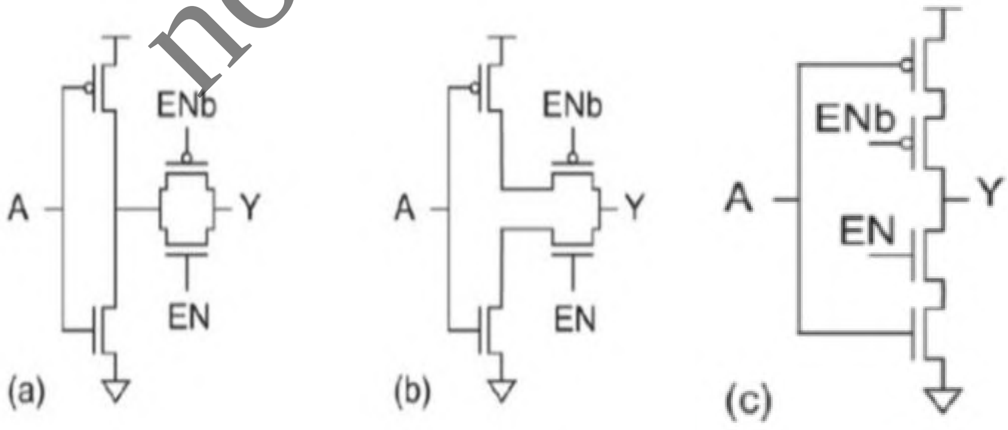


Fig. Resistance of Transmission gate as a function of input voltage

Tristate inverters:



- By cascading a transmission gate and an inverter forms a tristate inverter as shown in Fig (a)
- When EN = 1, EN'= 0, thus transmission gate is ON and transmits the output Y as the compliment of inverter input A.
- When EN = 0 and EN' =1, transmission gate is OFF and the output Y is in tristate or high impedance state.
- Fig (b) and (c) shows other configurations of tristate inverters

Ratioed Inverters Transfer Characteristics

- Other than CMOS inverter there are also other forms of inverters. One such is shown in the fig. below which has an nMOS with load as resistor.
- This is an nMOS inverter circuit. When Vin = 0, nMOS is OFF and output goes to Vdd through the Rload.
- When Vin = 1, nMOS is ON and pulls the output to gnd.
- When we consider the transfer characteristics and I-V characteristics, we see that as load is increased V_{OL} decreases also the current decreases. Thus choosing load resistor compromises between current and V_{OL}.

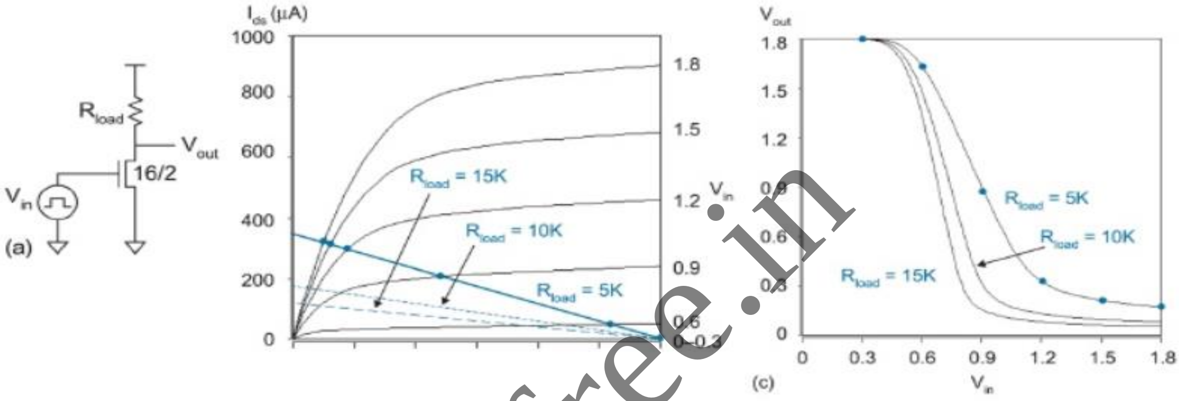


Fig. nMOS inverter with resistive load, I-V characteristics and transfer characteristics

- An alternate to this is using a more practical circuit called pseudo-nMOS inverter circuit, which uses a pMOS transistor as a load with its gate terminal always grounded.
- Here pMOS will be in ON state. When Vin = 0, nMOS is OFF and as pMOS is ON the output rises to Vdd. When Vin = 1, nMOS will be ON and pulls the output to gnd.
- When the transfer characteristics is observed as the W/L ratio is varied for pMOS in the pseudo-nMOS inverter circuit, the shape of the transfer characteristics varies.
- As parameter P (i.e., as W is decreased sharper characteristics is obtained) is varied characteristics varies with higher value of P less sharper characteristics is seen.
- In the circuit P/2 represents the W/L ratio.

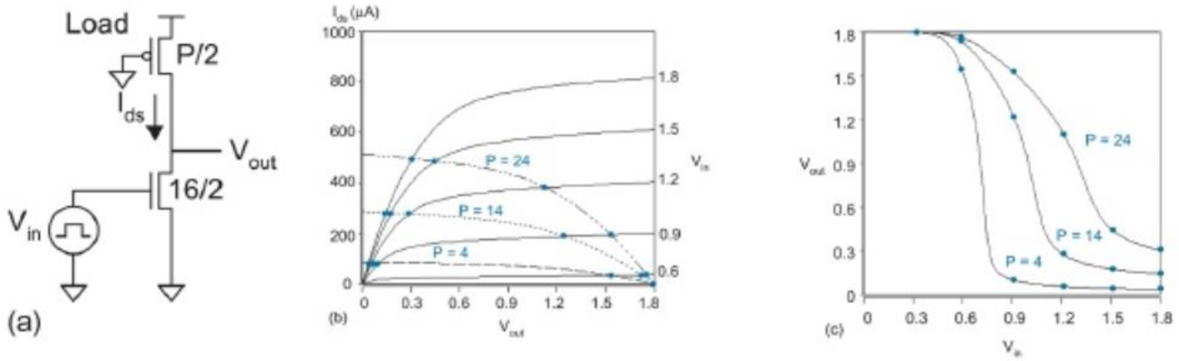


Fig. pseudo-nMOS inverter with I-V characteristics and transfer characteristics

- These types of gates are called as ratioed circuits as transfer function depends on the strength of pull down (pMOS) to pull up (nMOS) devices.
- In these types of circuits ratios must be chosen properly so that circuit operates properly.
- Disadvantage seen with these ratioed circuits are
    - Constant power dissipation
    - Poor noise margin
- However these circuits are used under limited circumstances such as reduced input capacitance and smaller area.

Fabrication

nMOS Fabrication:

Semiconductor device fabrication is the process of creating integrated circuits in multiple-step sequence of photolithographic and chemical processing during which electronic circuits are gradually created on a wafer made of pure semiconducting material.

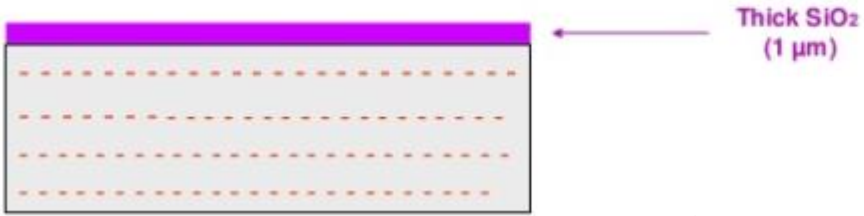
The following steps gives general aspect of nMOS fabrication process.

1. Processing is carried on thin wafer cut from single silicon crystal of high purity to which p-type impurities are introduced as crystal is grown. Wafers are around 75 to 150 mm in diameter and 0.4 mm thick. They are doped with boron (p-type) impurity concentration of $10^{15}/cm^3$ to 10^{16} $/cm^3$.
2. On this a thick layer if silicon dioxide (SiO2) of 1μm. This protects the surface, act as barrier to dopants and also act as an insulating layer on which other layers can be deposited and patterned.
3. The surface is now covered with photoresist and it is spun to achieve even distribution of required thickness.
4. A mask is used and the photoresist layer on the wafer is exposed to UV light. Mask defines those regions into which diffusion will take place and these regions remain unaffected after exposing to UV light and other region gets hardened.
5. The UV exposed regions are etched away along with the silicon dioxide layer so that the wafer surface is exposed in the window defined by the mask.
6. The remaining photoresist is removed and a thin layer of SiO2 is grown over entire surface and then polysilicon is deposited on top of this to form gate structure.
7. The thin oxide is removed to expose areas into which n-type impurities are diffused to form source and drain.
8. Thick oxide is grown all over again and then masked with photoresist and etched to expose selected area of polysilicon gate and drain and the source areas where connections are to be made.
9. The whole chip is then has metal (Al) deposited over its surface to a thickness of 1μm. This metal layer is then masked and etched to form the required interconnection pattern.

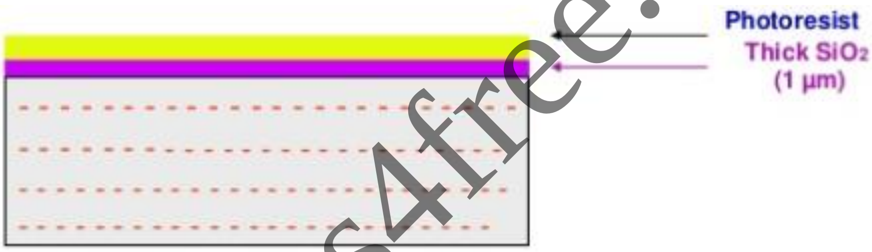
Fig below depicts the nMOS fabrication steps:

Si-substrate

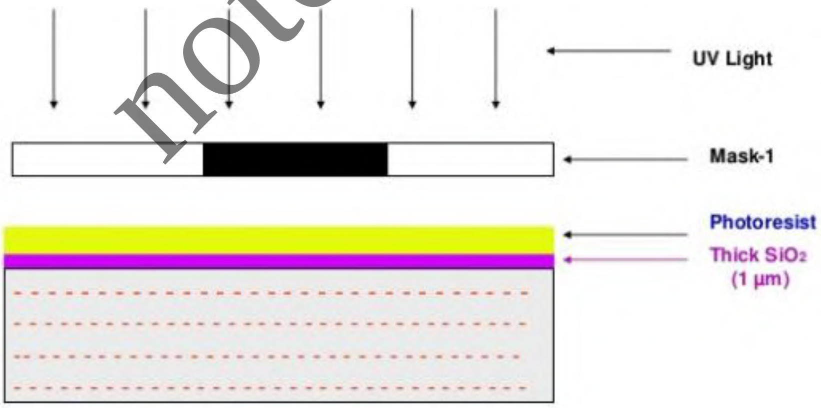
(1) Si with p-type impurities



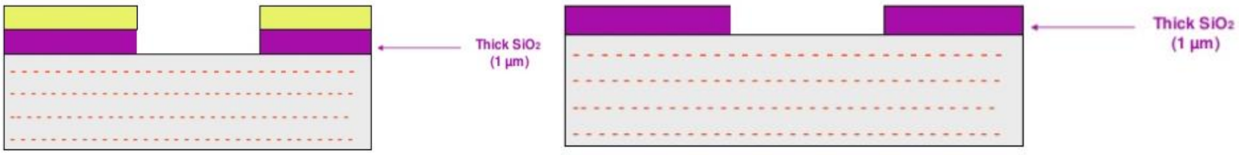
(2) Thin layer of SiO2 on substrate



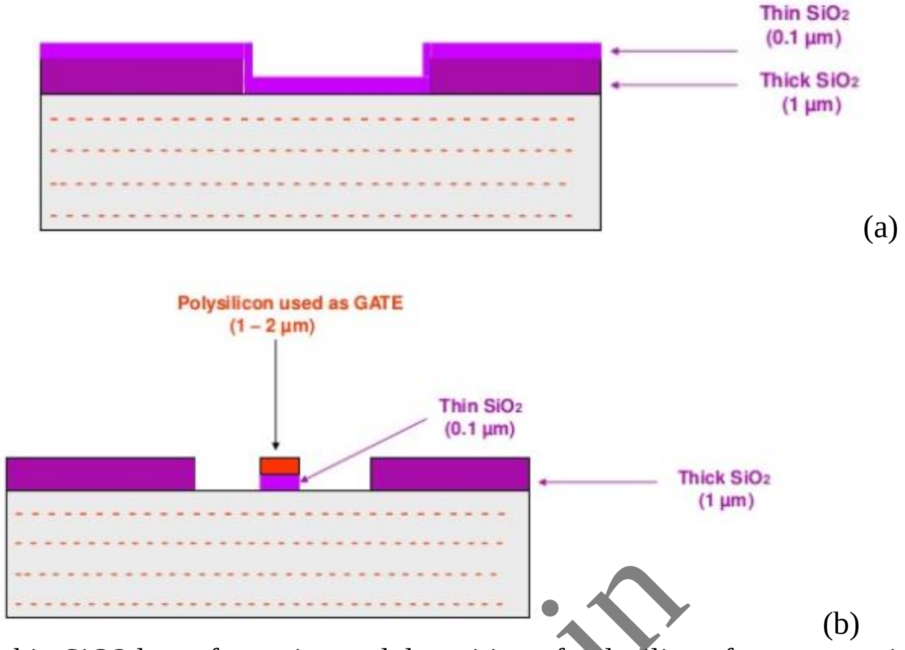
(3) Photoresist on the layer of SiO2



(4) Photoresist layer exposed to UV light through mask



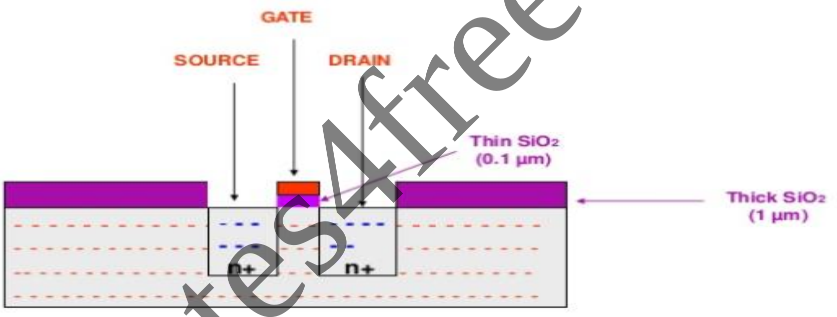
(5) UV exposed regions are etched away



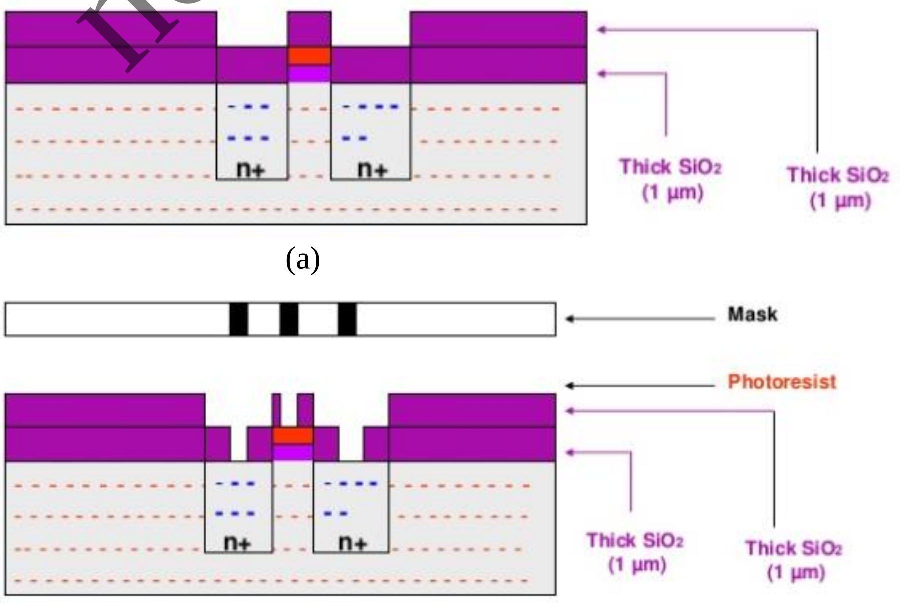
(a)

(b)

(6 a & b) thin SiO2 layer formation and deposition of polysilicon for gate terminal



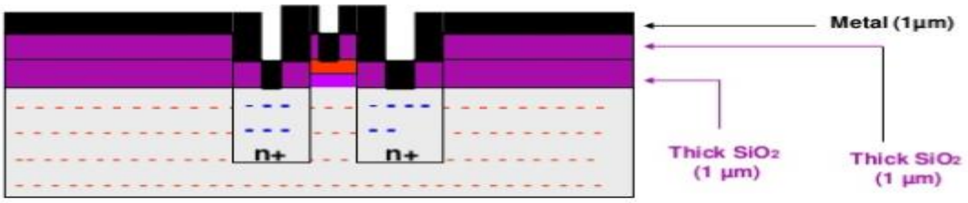
(7) n+ diffusion for source and drain formation



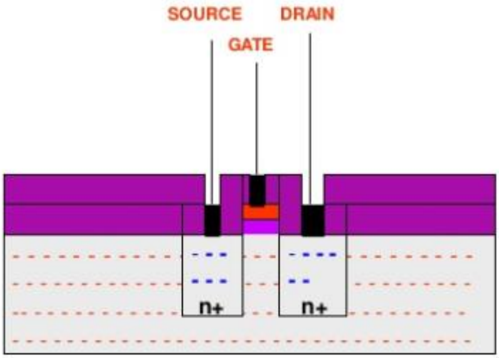
(a)

(b)

8(a & b) thick layer of SiO2 grown and masked with photoresist S and D contact cuts



(a)



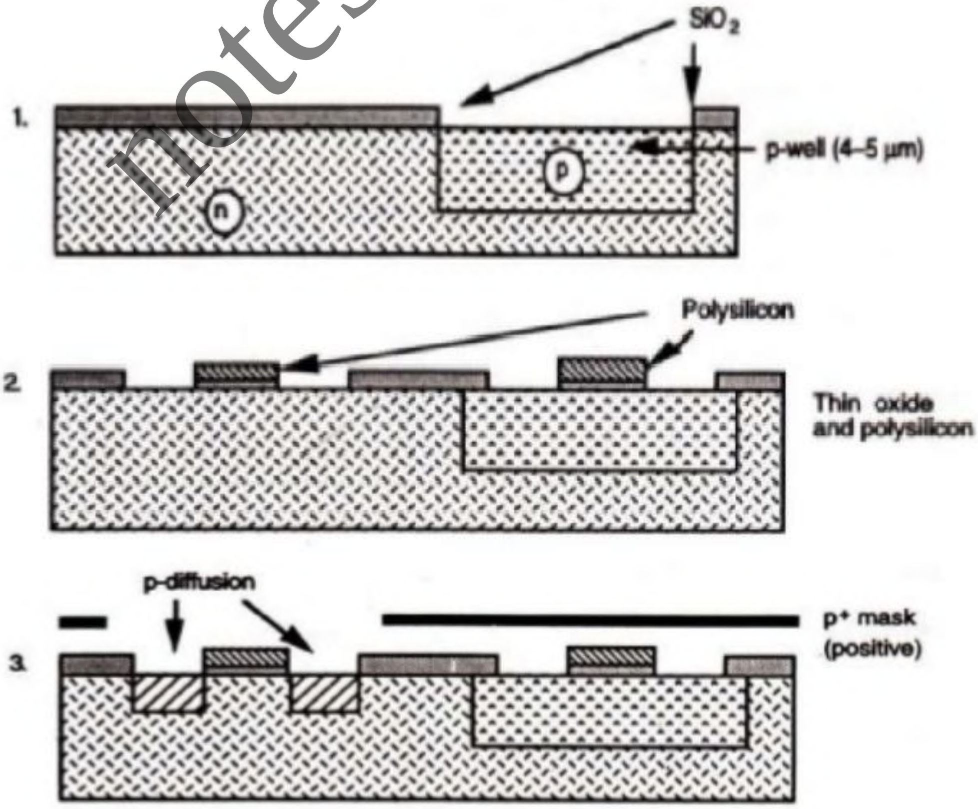
(b)

9(a & b) metal layer deposition and metal layer is masked and etched to form final nMOS transistor

CMOS Fabrication

- There are a number of methods for CMOS fabrication, which includes p-well, n-well, twin tub and silicon-on-insulator (SOI) processes.
- The p-well process is widely used and the n-well process as it is an retrofit to existing nMOS technology.

The p-well Process



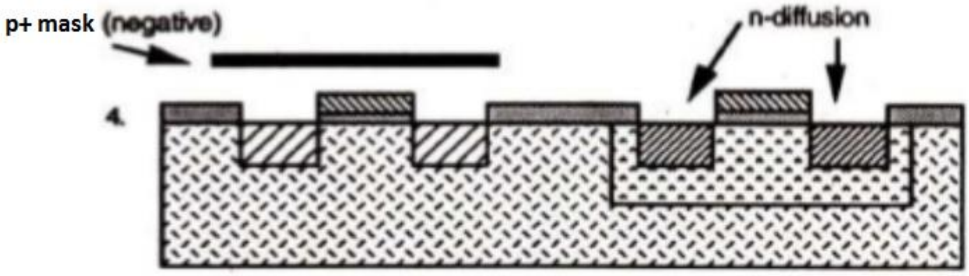


Fig. CMOS p-well process steps

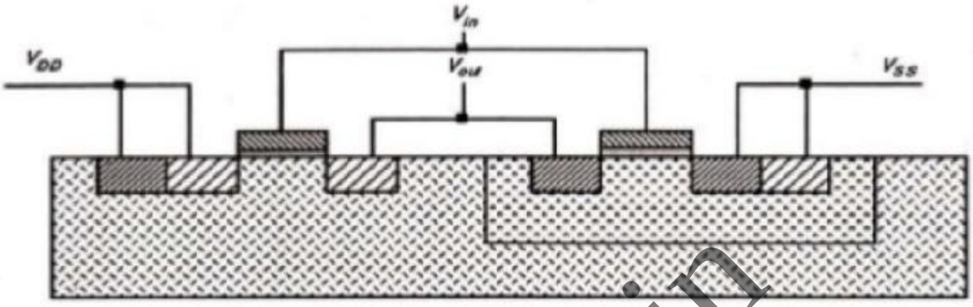


Fig. CMOS p-well inverter showing V_{DD} and V_{SS} substrate connections

- The p-well structure has an n-type substrate in which p-type devices can be formed with the help of masking and diffusion. In order to accommodate n-type devices, deep p-well is diffused into the n-type substrate. This is shown in Fig 1.

  Masking, patterning and diffusion process is same as that of nMOS fabrication. The summary of processing steps are:
  - Mask: defines the areas in which the deep p-well diffusion has to take place.
  - Mask 2: defines the thin oxide region (where the thick oxide is to be removed or stripped and thin oxide grown)
  - Mask 3: patterning the polysilicon layer which is deposited after thin oxide.
  - Mask 4: A p+ mask is used (to be in effect "AND" with mask 2) to define areas where p-diffusion is to take place.
  - Mask 5: –ve form of mask 4 (p+ mask) is used which defines areas where n-diffusion is to take place.
  - Mask 6: Contact cuts are defined using this mask.
  - Mask 7: The metal layer pattern is defined by this mask.
  - Mask 8: An overall passivation (over glass) is now applied and it also defines openings for accessing pads.
- In the process, the diffusion should be carried out with special care as p-well concentration and depth will affect the threshold voltage and also the breakdown voltage of the n-transistor.
- To achieve low threshold voltage either deep-well diffusion or high-well resistivity is required.
- But deep well require larger spacing between n- and p-type transistors and wires due to lateral diffusion and therefore needs larger chip area.
- The p-well acts as substrate for n-devices within parent n-substrate and two areas are electrically isolated

The n-well Process

- The p-well processes have been one of the most commonly available forms of CMOS. However, an advantage of the n-well process is that it can be fabricated on the same process line as conventional n MOS.

- n –well CMOS circuits are also superior to p-well because of the lower substrate bias effects on transistor threshold voltage and inherently lower parasitic capacitances associated with source and drain regions.
- Typically n-well fabrication steps are similar to a p-well process, except that an n-well is used which is illustrated in flow diagram
- The first masking step defines the n-well regions.
- The well depth is optimized to ensure against p-substrate to p+ diffusion breakdown without compromising the n-well to n+ mask separation.
- The next steps are to define the devices and diffusion paths, grow field oxide, deposit and pattern the polysilicon, carry out the diffusions, make contact cuts and metallization.
- An n-well mask is used to define n-well regions, as opposed to a p-well mask in a p-well process.
- Fig. Depicts inverted circuit fabricated by n-well process.

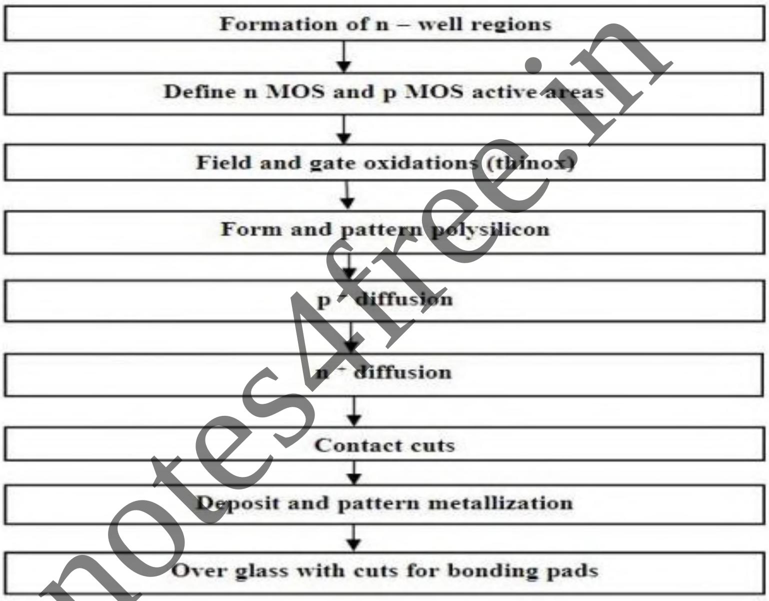

Formation of n – well regions
Define n MOS and p MOS active areas
Field and gate oxidations (thinox)
Form and pattern polysilicon
p + diffusion
n + diffusion
Contact cuts
Deposit and pattern metallization
Over glass with cuts for bonding pads

Fig. Main steps in typical n-well process

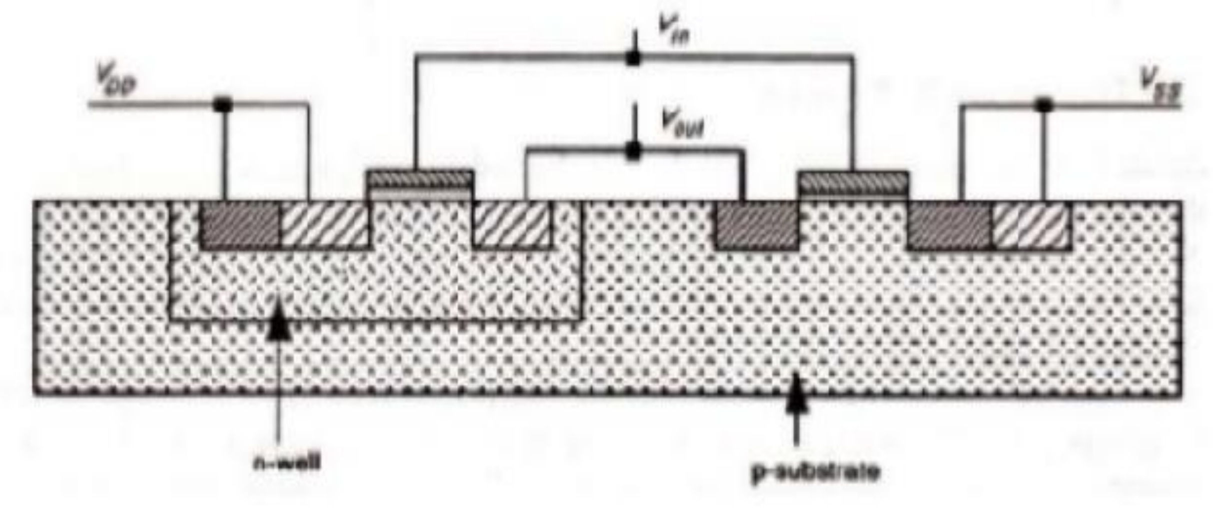
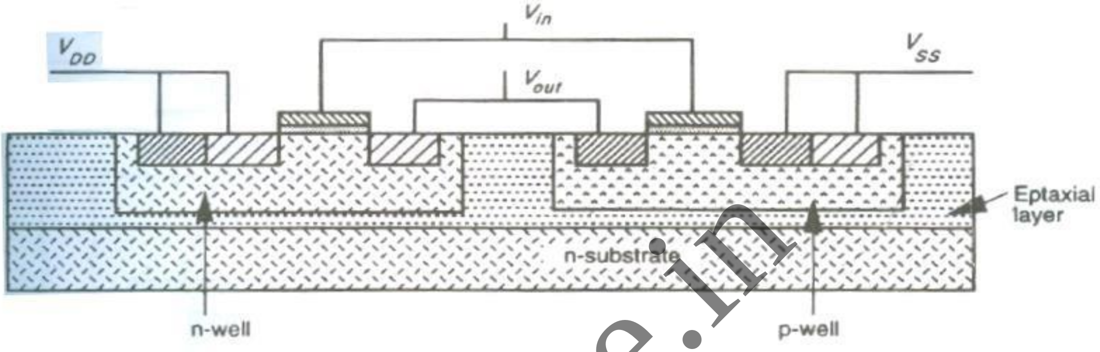


Fig. (A) Cross-sectional view of n-well CMOS Inverter

The Twin-Tub process

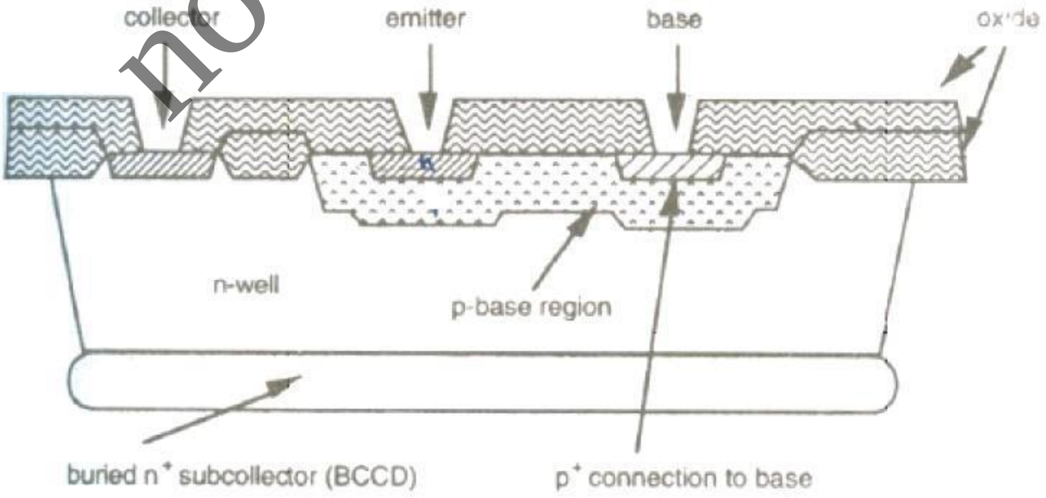
- Twin-tub CMOS technology provides the basis for separate optimization of the p-type and n-type transistors, thus making it possible for threshold voltage, body effect, and the gain associated with n- and p-devices to be independently optimized.
- Generally the starting material is either an n+ or p+ substrate with a lightly doped epitaxial or epi layer, which is used for protection against latch-up.
- The aim of epitaxial is to grow high purity silicon layers of controlled thickness with accurately determined dopant concentrations distributed homogeneously throughout the layer. The electrical properties for this layer are determined by the dopant and its concentration in the silicon.



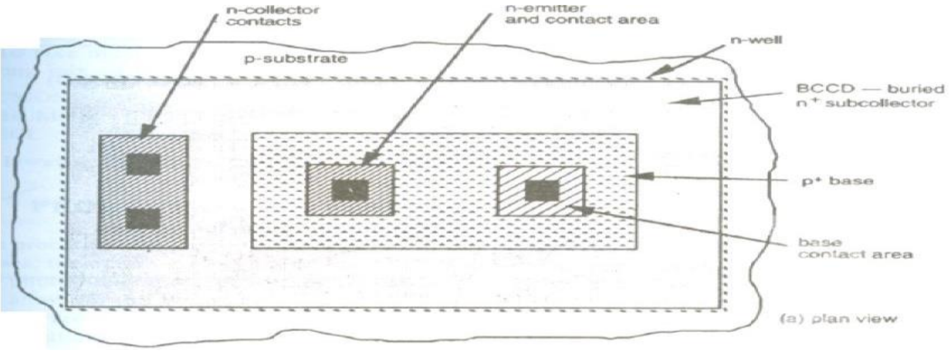
Twin-tub structure.

BiCMOS Technology

- The load driving capabilities of MOS transistors is less because of limited current sourcing and sinking abilities of both p and n transistors
- Bipolar transistors provide high gain, better noise and high frequency characteristics than MOS transistors.
- Thus Bipolar can be combined with CMOS technology to build high speed devices called as BiCMOS devices.



Cross section of BiCMOS process



Layout view of BiCMOS process.

	CMOS technology	BiCMOS technology
1.	It has bidirectional capability (source and drain are interchangeable)	Essentially unidirectional
2.	Low static power dissipation	High power dissipation
3.	It has high input impedance	It has Low input impedance
4.	High Packing Density	Low Packing Density
5.	It has Low gain	It has High gain
6.	High delay sensitivity to load	Low delay sensitivity to load

Module 2
MOS and BiCMOS Circuit Design Processes

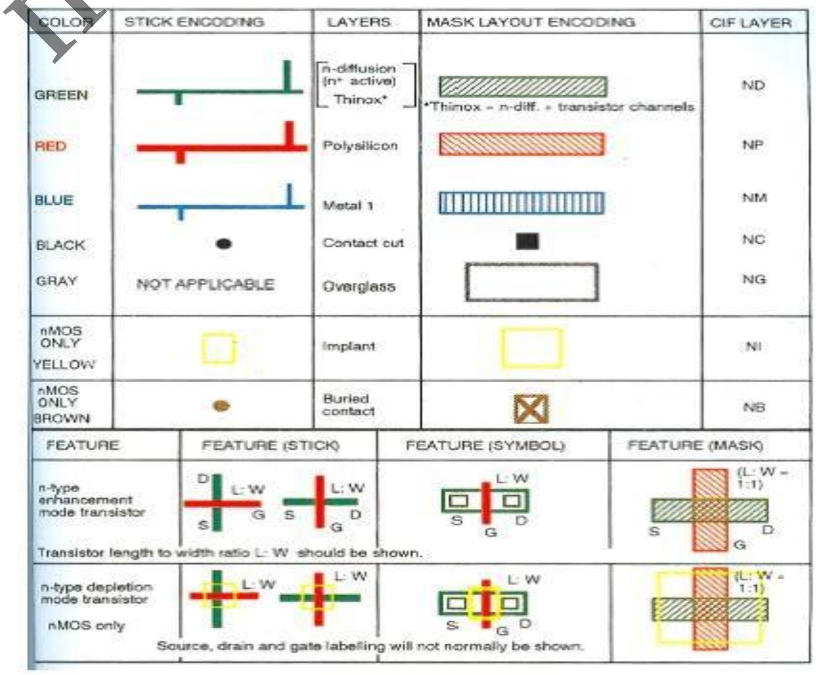
- Methods of realizing circuit design in silicon
- The design process can be understood by means of stick diagrams and symbolic diagrams along with set of design rules.
- Design rules: is a communication link between designers specifying the requirements and the fabricator.

MOS Layers:

- MOS circuits are basically formed by 4 layers
  - Metal
  - Polysilicon
  - N diffusion
  - P diffusion
- Here all the 4 layers are isolated from each other through thick or thin oxide layer (i.e., silicon dioxide layer)
- The thin oxide (thinox) layer includes n-diffusion, p-diffusion and transistor channel.

Stick diagram:

- Stick diagrams are a means of capturing topography and layer information using simple diagrams.
- They convey layer information through color codes (or monochrome encoding).
- Acts as an interface between symbolic circuit and the actual layout.
- Stick diagrams do show all components/vias(contacts), relative placement of components and helps in planning and routing. It goes one step closer to layout.
- However they do not show exact placement of components, transistor sizes, length and width of wires also the boundaries. Thus we can say that it does not give any low level details.
- The color encodings chosen for different technologies is shown below.
- **Encodings for NMOS process:**



COLOR	STICK ENCODING	LAYERS	MASK LAYOUT ENCODING	CIF LAYER
GREEN		n-diffusion (n⁺ active) Thinox*	▨ *Thinox = n-diff. + transistor channels	ND
RED		Polysilicon	▨	NP
BLUE		Metal 1	▥	NM
BLACK	●	Contact cut	■	NC
GRAY	NOT APPLICABLE	Overglass	□	NG
nMOS ONLY YELLOW	□	Implant	□	NI
nMOS ONLY BROWN	●	Buried contact	☒	NB

FEATURE	FEATURE (STICK)	FEATURE (SYMBOL)	FEATURE (MASK)
n-type enhancement mode transistor	D L: W S G S L: W G D	L: W S G D	(L: W = 1:1) S G D
Transistor length to width ratio L: W should be shown.			
n-type depletion mode transistor nMOS only	L: W L: W	L: W S G D	(L: W = 1:1)
Source, drain and gate labelling will not normally be shown.			

Procedure to draw Stick Diagram:

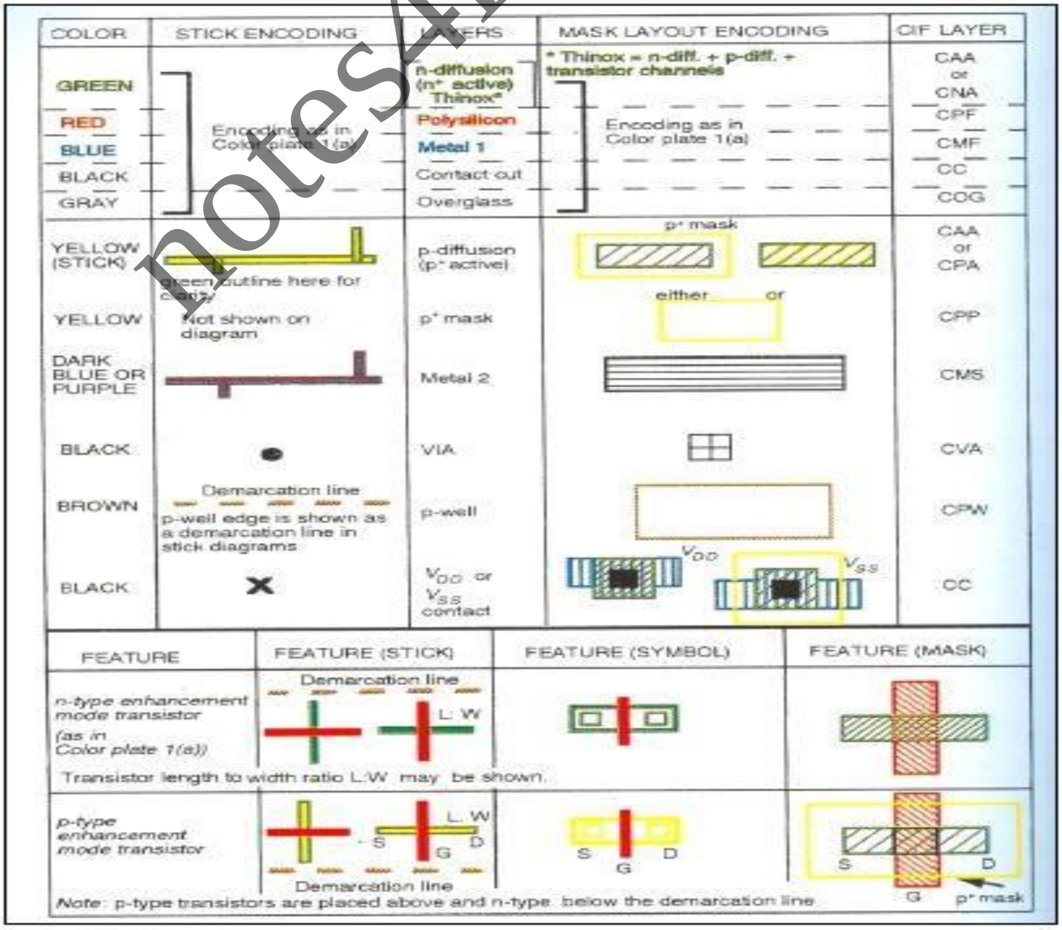
Nmos Design Process.

1) Draw two metal lines/ power rails providing sufficient space to accommodate all transistors. i.e: Vdd & Vss.
2) Draw common n+ diffusion layer for all the transistors.
3) Provide Vdd and Vss contacts.
4) Draw polysilicon to cross n+ diffusion layer to form transistors.
5) Create buried contact for depletion transistor.
6) Provide input and output connection.

CMOS Design Process:

- Two type of transistors are used i.e: Nmos and Pmos, thus in stick diagram demarcation line is used to separate them.
- All Pmos transistors are placed above Demarcation line and Nmos transistors below demarcation line.
- While drawing stick Diagram
    1. Diffusion paths must not cross the demarcation line
    2. N-diffusion and P-diffusion wires must not join.
    3. Nmos and Pmos transistors are joined by Metal layer when it is required.
    4. Cross must be placed on Vdd and Vss which represent substrate and P-well connection respectively.

Encodings for CMOS process:



COLOR	STICK ENCODING	LAYERS	MASK LAYOUT ENCODING	CIF LAYER
GREEN	Encoding as in Color plate 1(a)	n-diffusion (n+ active) Thinox*	* Thinox = n-diff. + p-diff. + transistor channels Encoding as in Color plate 1(a)	CAA or CNA
RED		Polysilicon		CPF
BLUE		Metal 1		CMF
BLACK		Contact cut		CC
GRAY		Overglass		COG
YELLOW (STICK)	green outline here for clarity	p-diffusion (p+ active)	p+ mask	CAA or CPA
YELLOW	Not shown on diagram	p+ mask	either or	CPP
DARK BLUE OR PURPLE		Metal 2		CMS
BLACK	●	VIA		CVA
BROWN	Demarcation line p-well edge is shown as a demarcation line in stick diagrams	p-well		CPW
BLACK	✕	V_{DD} or V_{SS} contact	V_{DD} V_{SS}	CC

FEATURE	FEATURE (STICK)	FEATURE (SYMBOL)	FEATURE (MASK)
n-type enhancement mode transistor (as in Color plate 1(a))	Demarcation line L: W		
Transistor length to width ratio L:W may be shown.			
p-type enhancement mode transistor	L: W S G D Demarcation line	S G D	S D G p+ mask

Note: p-type transistors are placed above and n-type below the demarcation line

Procedure to draw Stick Diagram:

1) Draw two metal lines/ power rails providing sufficient space to accommodate all transistors. i.e: Vdd & Vss.
2) Draw demarcation line in the middle of the two power lines.
3) Draw P+ diffusion above demarcation and N+ diffusion below demarcation
4) Draw polysilicon to represent Pmos and Nmos which represents gates of the transistor.
5) Connect source terminal of transistors to supply.
6) Drain terminals of transistor are connected using metal 1.
7) Place contact cuts wherever necessary.
8) Draw X which represents substrate and P-well contact on power lines.

Layout: describes actual layers and geometry on silicon substrate to implement a function(Expressions).

[Diffusion region where transistor can be formed is called active region, polysilicon serves as the gate of MOS transistor. L defines channel length and W represents width of channel/active region]

Design rules: are set of guidelines which specify minimum dimension and spacing allowed in layout drawing. Design rules also acts a communication link between circuit designers and process engineers during manufacturing phase.

Goal of design rule: is to achieve optimum yield. Yield = (No. of good chips on wafer)/(Total no. of chips on wafer).

Design rules are also called layout rules. If the circuit performance has to be increased then rules must be more aggressive. Else this leads to non-function of the circuit or yield reduction. There are two rules.

1. Micron Rule - Absolute Dimension rule, here all sizes and spacing are specified in micron. Here the circuit density is the important goal.
2. Lambda (λ) Based Rules - The Lambda based design rules are Proposed by Mead and Conway. Scalable design rules, here this design rule normalizes all geometric design rule by parameter lambda (λ) also called as scaling factor/feature size. In this all mask patterns are expressed as multiples of lambda.

Advantages of lambda based design rules:

1. The mask layout can be scaled down proportionally if the feature size of the fabrication process is reduced.
2. Design rules are conservative.
3. This rule enable technology changes and design reuse and reduced design cost.

Disadvantages:

1. Linear scaling cannot be extended and is limited over range of dimension (1-3 μm)
2. As rules are conservative, results in over dimension and density of design is less.

The Design rules can be conveniently set out in diagrammatic form as shown in fig. 1 for width and separation of conducting path. In fig. 2 shows the design rules associated with extensions and separations with transistor. Fig. 3 and 4 demonstrates the design rules for contacts between layers. Table below also gives the layer and distance of separation dimensions.

Layer	Dimension
n-diffusion	2λ
p-diffusion	2λ
Polysilicon	2λ
Metal 1	3λ
Metal 2	4λ

Layer dimension

Layer -Layer	Dimension
n-diffusion – n-diffusion	3λ
p-diffusion – p-diffusion	3λ
n/p diffusion - polysilicon	1λ
Poly-poly	2λ
Metal 1	3λ
Metal 2	4λ

Distance of Separation

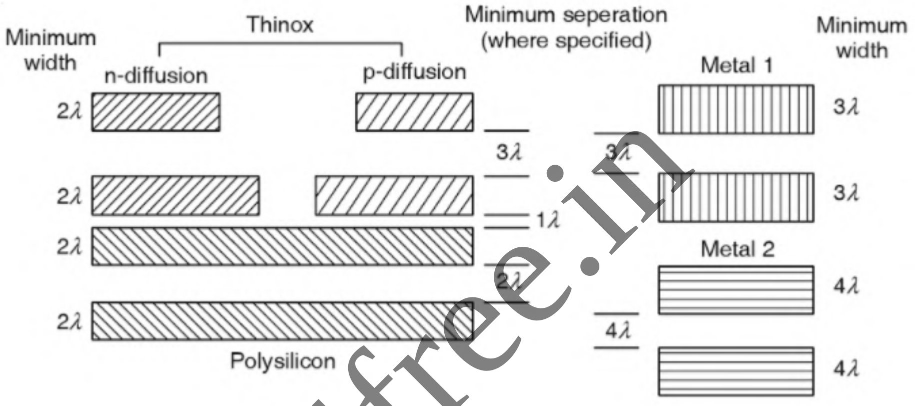


Fig. 1 Design rules for wires and separations (nMOS and CMOS)

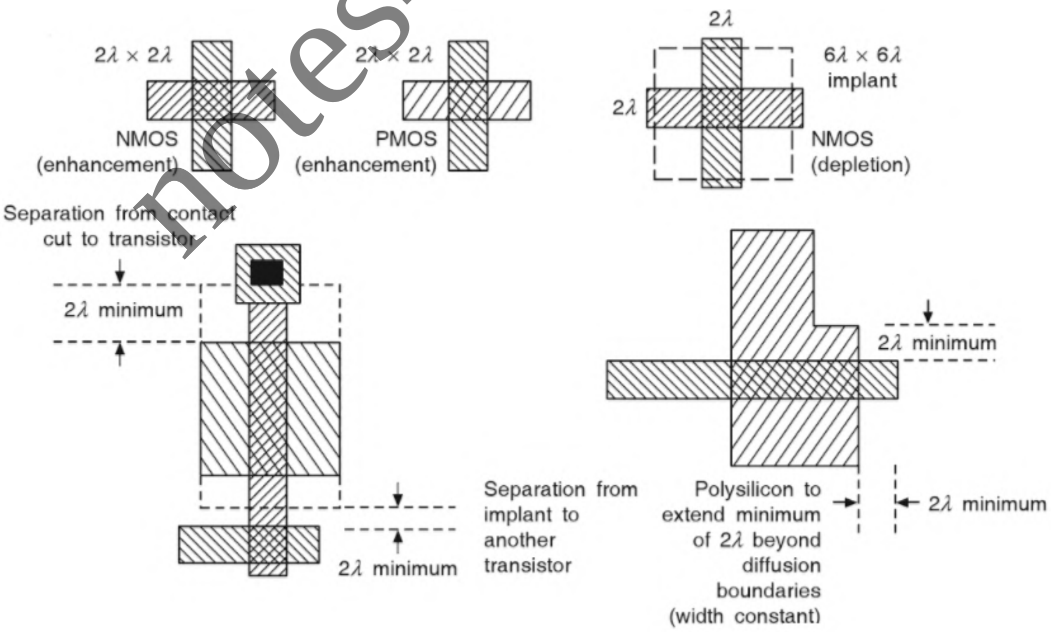

2λ × 2λ
NMOS (enhancement)
2λ × 2λ
PMOS (enhancement)
2λ
6λ × 6λ implant
2λ
NMOS (depletion)
Separation from contact cut to transistor
2λ minimum
2λ minimum
Separation from implant to another transistor
2λ minimum
Polysilicon to extend minimum of 2λ beyond diffusion boundaries (width constant)
2λ minimum

Fig. 2 Design rules for Transistors (nMOS, pMOS and CMOS)

Transistor design rules

- Minimum dimension of transistor is 2λ × 2λ – overlapping of diffusion and ploy
- Poly and diffusion both must extend beyond the boundary of transistor at least by 2λ

- Implant for depletion mode transistor is 6λ × 6λ i.e., implant must extend boundary of transistor by at least 2λ in all direction.
- From the boundary/ implant of one transistor, the next transistor should maintain min distance of 2λ
- The distance from contact cut to transistor should be at least 2λ

Metal contact – contact between metal 1 to polysilicon OR metal 1 to diffusion (active region) is called metal contact. This is shown in fig. 3

- A 2λ × 2λ cut centered on 4λ × 4λ superimposed area is used to connect layers
- In case of multiple contacts the distance between adjacent contacts should be 2λ

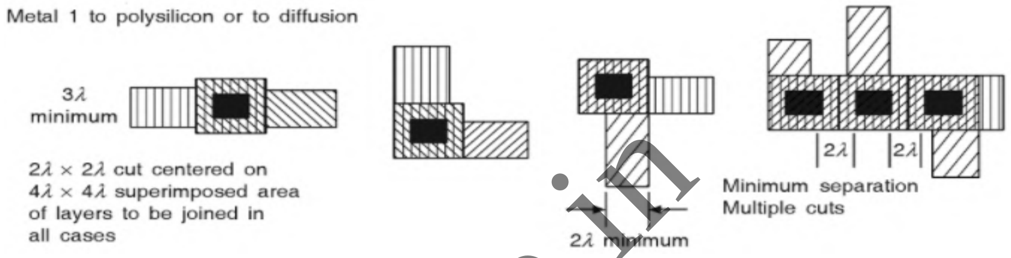

Metal 1 to polysilicon or to diffusion
3λ minimum
2λ × 2λ cut centered on 4λ × 4λ superimposed area of layers to be joined in all cases
2λ minimum
2λ
2λ
Minimum separation Multiple cuts

Fig. 3. Contacts (nMOS and CMOS)

Via contact – the contact between metal 1 and metal 2 is called via contact as shown in fig. 4.

- A 2λ × 2λ cut centered on 4λ × 4λ superimposed area is used to connect layers
- To connect metal 2 with diffusion via and cut both are used

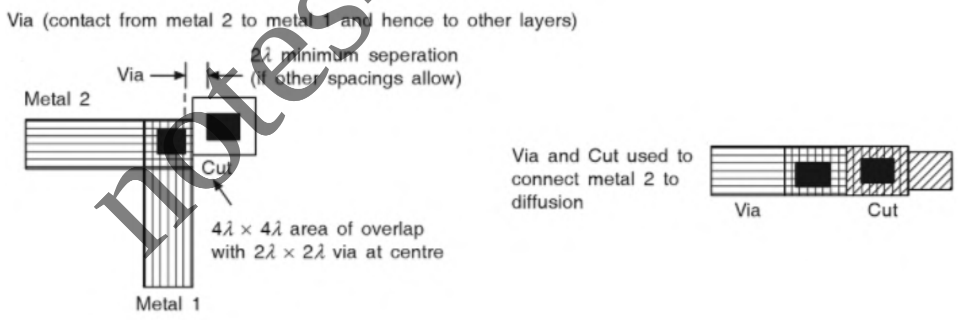

Via (contact from metal 2 to metal 1 and hence to other layers)
Via
2λ minimum seperation (if other spacings allow)
Metal 2
Cut
4λ × 4λ area of overlap with 2λ × 2λ via at centre
Metal 1
Via and Cut used to connect metal 2 to diffusion
Via
Cut

Fig. 4 contacts

Contact Cuts:

- Electrical connection between layers can be done using special structures 'contact cuts'.
- There are 3 approaches for contacts between polysilicon and diffusion in nMOS circuits. They are
  1. Polysilicon to metal and then to diffusion
  2. Buried contact - polysilicon to diffusion
  3. Butting contact - polysilicon to diffusion using metal
- ✓ Among the three buried contact is most used as it gives economy in space and reliable contact.
- ➢ Buried contact is distinguished feature in nMOS for connection between poly and diffusion and this is most widely used than butting contact.

Buried Contact (nMOS):

- Layer is joined over the area of 2λ × 2λ with buried contact cut extending by 1λ in all directions except in the diffusion path. It extends by 2λ in order to avoid formation of unwanted transistors.
- The contact cut shown in broken line indicates the region where thinox is removed on the silicon wafer and polysilicon gets deposited on wafer.
- When impurities are added, it diffuse into poly and also to diffusion region within the contact area. This provides satisfactory contact between ploy and diffusion as shown in fig 5.

- In CMOS poly to diffusion connections are made through metal. The process of making connection between metal and either of 2 layers (poly or diffusion) is by buried contact.

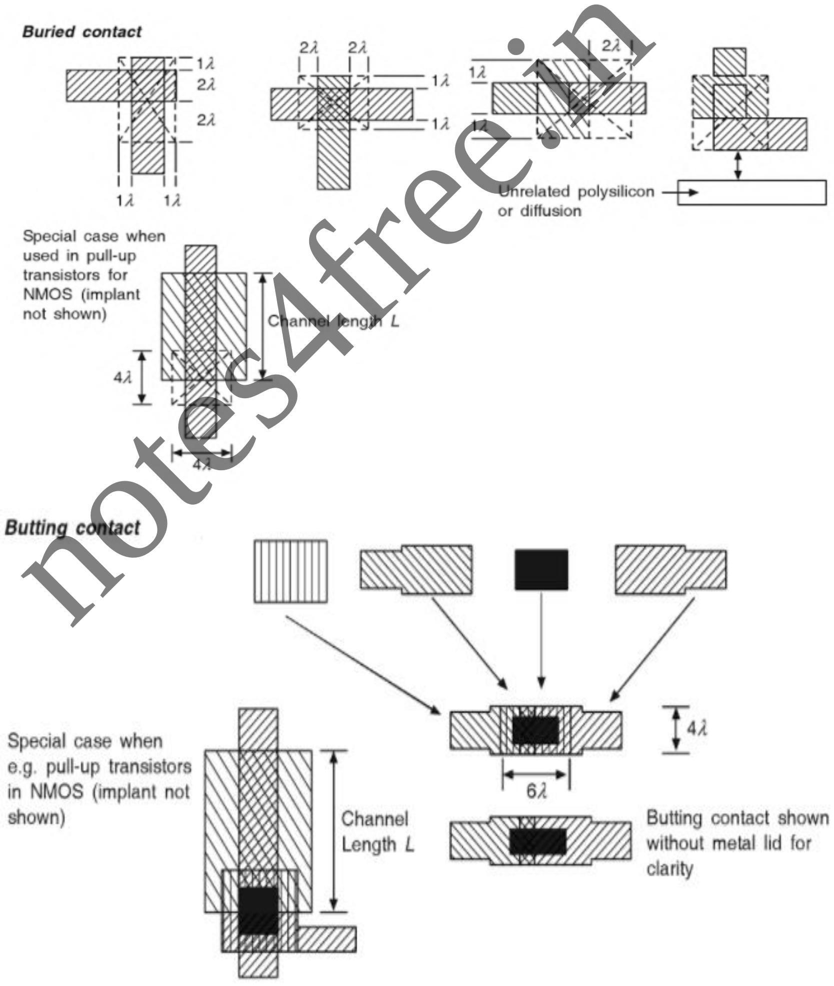

Buried contact
1λ
2λ
2λ
1λ 1λ
2λ 2λ
1λ
1λ
2λ
1λ
1λ
Unrelated polysilicon or diffusion
Special case when used in pull-up transistors for NMOS (implant not shown)
Channel length L
4λ
4λ
Butting contact
4λ
6λ
Special case when e.g. pull-up transistors in NMOS (implant not shown)
Channel Length L
Butting contact shown without metal lid for clarity

Fig. 5 Buried and butting contacts only for nMOS

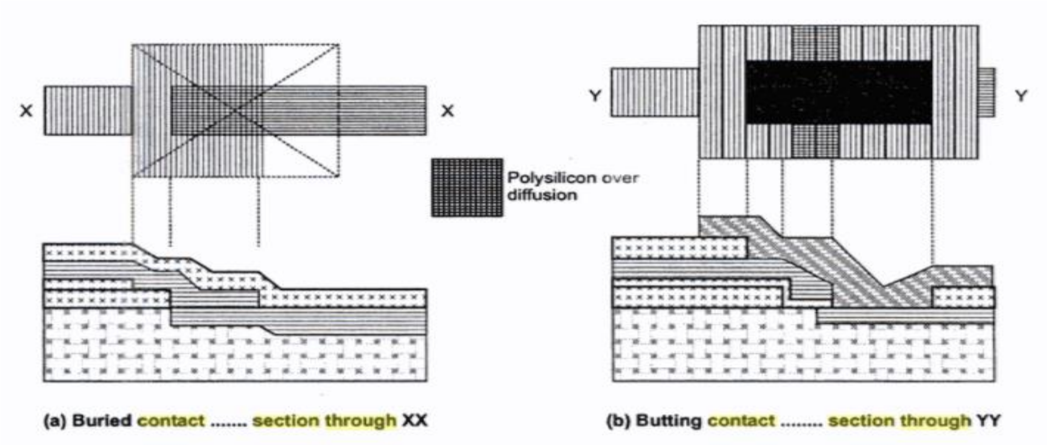

X
X
Polysilicon over diffusion
Y
Y
(a) Buried contact .... section through XX
(b) Butting contact .... section through YY

Fig 5. Cross section through contact structures

Butting contact

- Butting contact process is complicated and done when two layers do not overlap. Contact cut of 2λ × 2λ is made until each of layers is joined. The layers are held in such a way that these two contacts become continuous.
- The poly and diffusion outlines overlap and thinox under ploy acts as mask during diffusion process. Finally contact between two butting layers is done by a metal. This can be seen in fig. 5 cross-sectional view.

Double Metal MOS Process Rules:

- If to process considered till now introduction of second metal layer will boosts the design capabilities. It gives more freedom. Ex. this will be helpful for power rail (Vdd and Vss/Gnd) distribution and also for clock.
- This process is called Double Metal MOS Process
- This technique involves connecting metal 1 and metal 2 contacts called 'via'. This is shown in fig. 4 and fig. 6

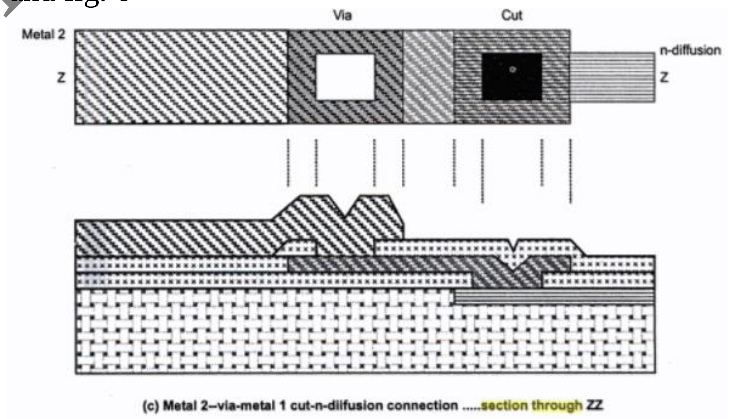

Metal 2
Via
Cut
n-diffusion
Z
Z
(c) Metal 2--via-metal 1 cut-n-diifusion connection .....section through ZZ

Fig. 6 cross section of via contact structure

- The 2^{nd} metal layer is coarser than 1^{st} metal layer (conventional) and the isolation layer between the 2 is usually thicker than normal.
- To distinguish contacts between 1^{st} and 2^{nd} metal layer they are called as ‘vias’ rather than contact cut
- In stick diagram representation its color code is dark blue or purple.

The steps of fabrication process is as follows:

1. Using chemical vapor deposition oxide layer under 1st metal layer is deposited.
2. using same method oxide between 2 metal layers are formed.
3. Selected areas of oxide are removed by using plasma etching. The etching process is done under high vertical ion bombardment to get high and uniform etching.

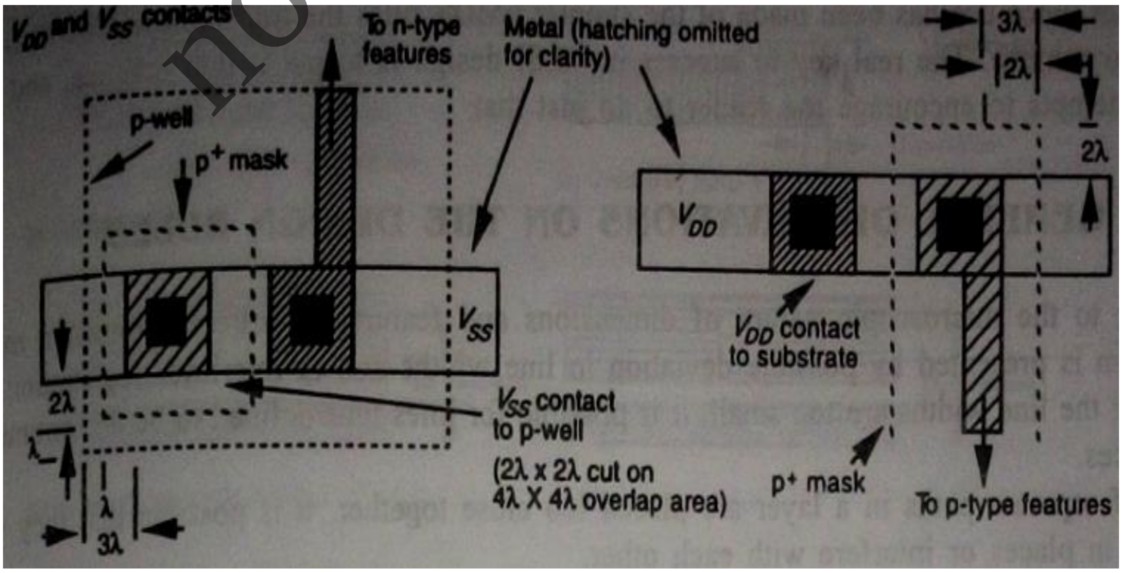
The layout strategy used with double metal process is summarized as below

1. Second metal layer is usually used for global power railings and clock lines
2. First metal layer is used for local power distribution and signals.
3. The layout of the two metals are such that are mutually orthogonal wherever possible

- Similar to double metal process, other process allows second poly layer. The process steps are similar to previously described process.
- The first polysilicon (poly 1) layer is deposited and patterned on this a second thinox (thin oxide) layer is grown. On this the second polysilicon (poly 2) layer is deposited and patterned. Thus 2nd thinox isolated the poly layers.
- Presence of poly 2 provides greater flexibility in interconnections and allows transistors to be formed by intersection of poly 2 and diffusion.

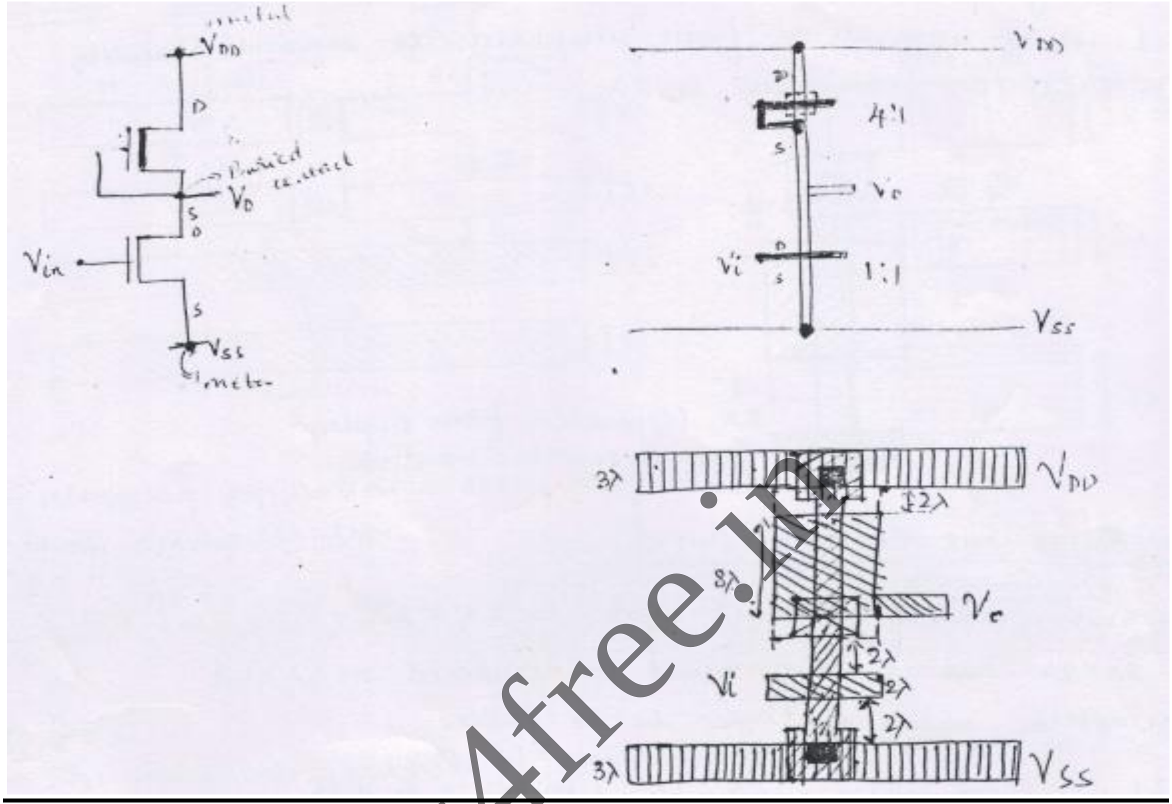
CMOS Lambda-based Design Rules:

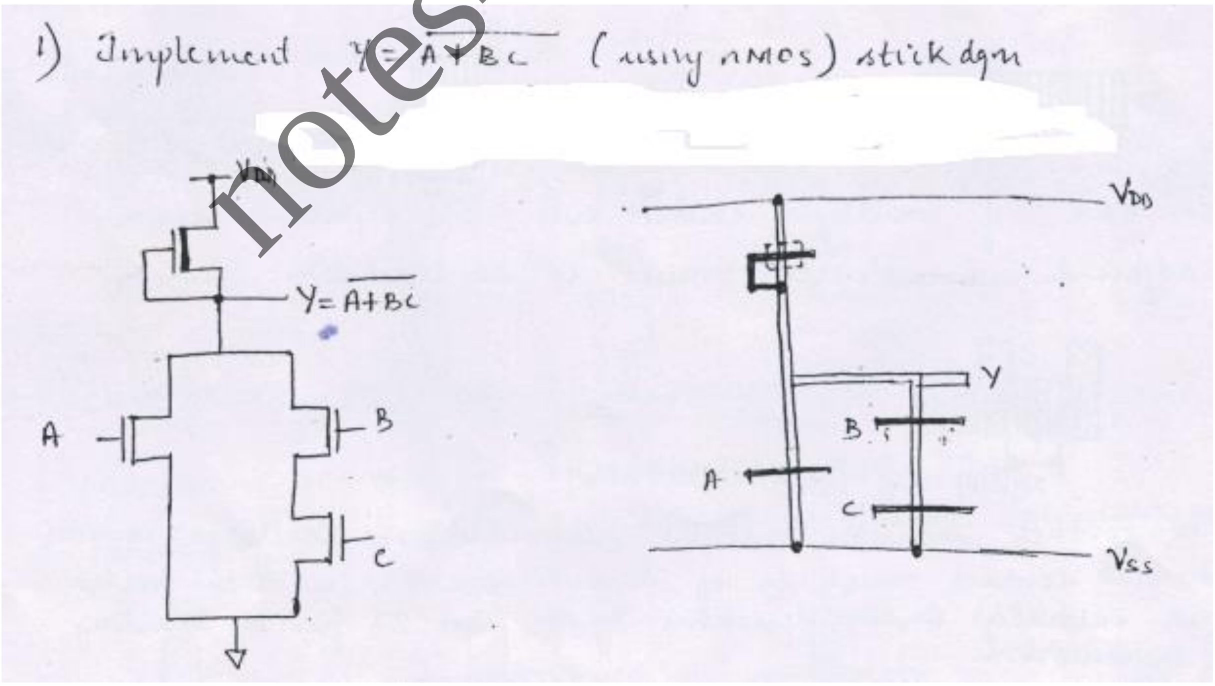
- Comparing to Nmos fabrication process, CMOS fabrication is more complex.
- Extending the Nmos design rules, Noting exclusion of butting contact and buried contact rules.
- Additional rules associated with CMOS process concerned with unique feature p-well CMOS, i.e: p-well and P+ Mask and Substrate contact.

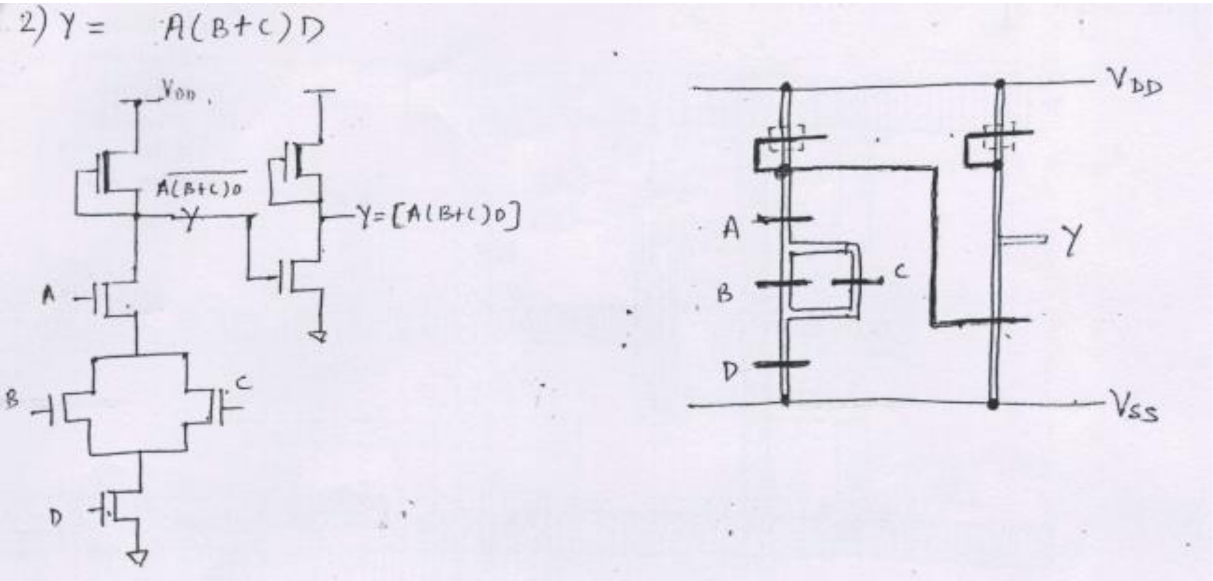

V_{DD} and V_{SS} contacts
To n-type features
Metal (hatching omitted for clarity)
3λ
2λ
2λ
p-well
p+ mask
V_{SS}
V_{DD}
V_{DD} contact to substrate
2λ
λ
V_{SS} contact to p-well
(2λ x 2λ cut on 4λ X 4λ overlap area)
λ
3λ
p+ mask
To p-type features

Problems on stick diagram and layouts.

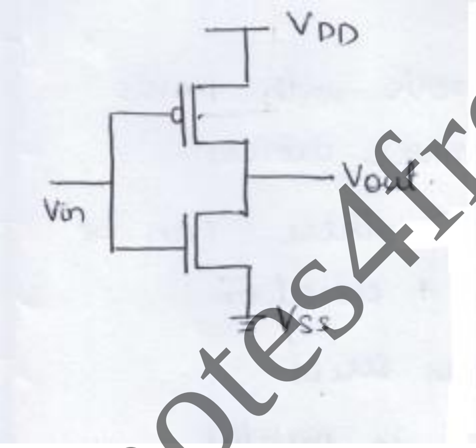
<u>Nmos Inverter</u>

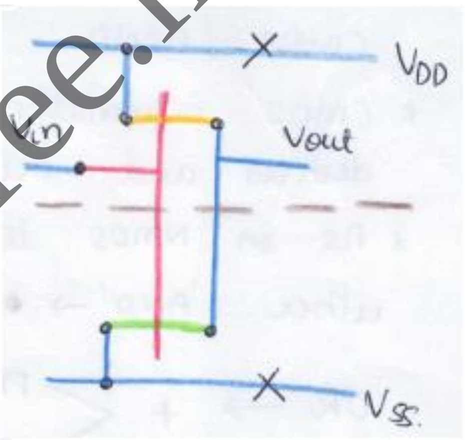


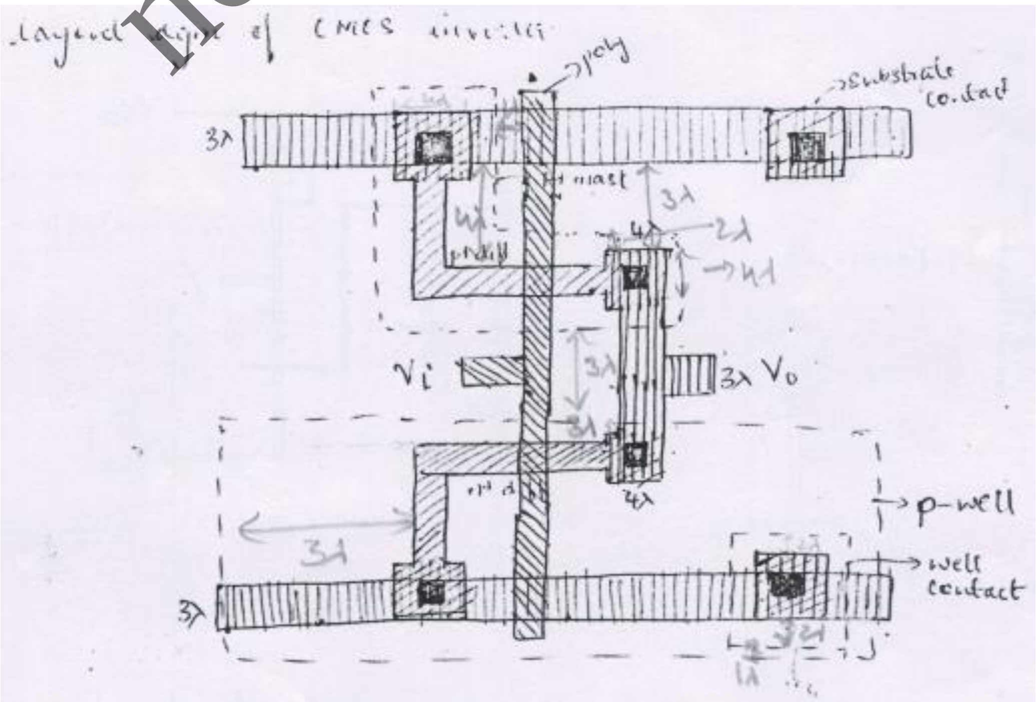

1) Implement $Y = \overline{A+BC}$ (using nMOS) stick dgm

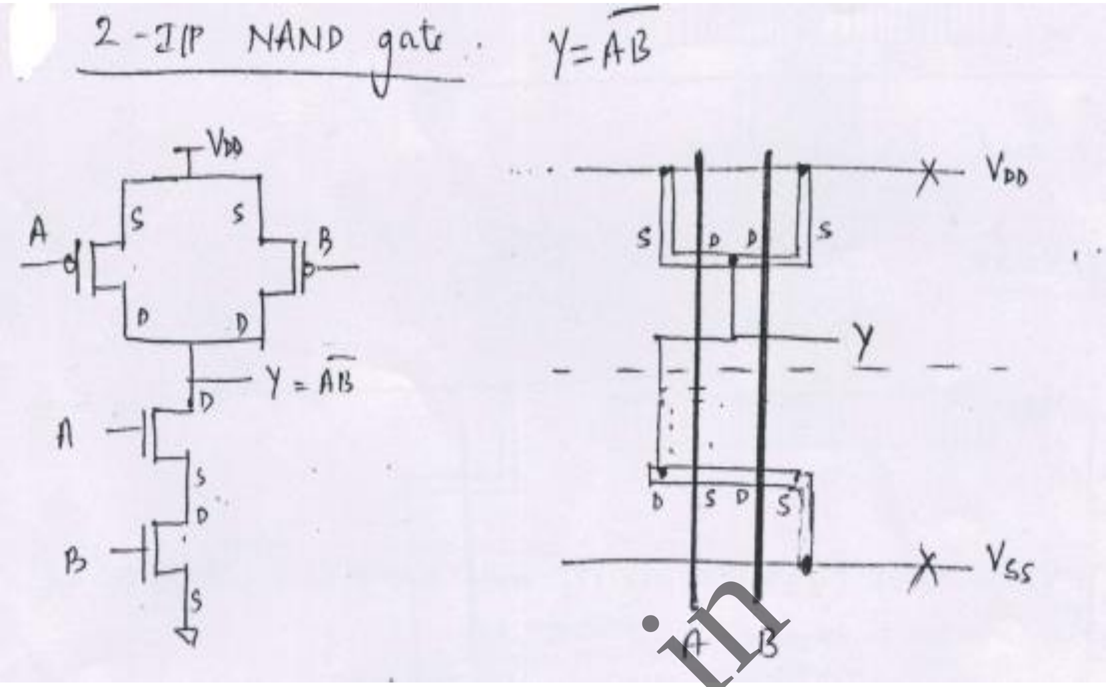


CMOS Inverter

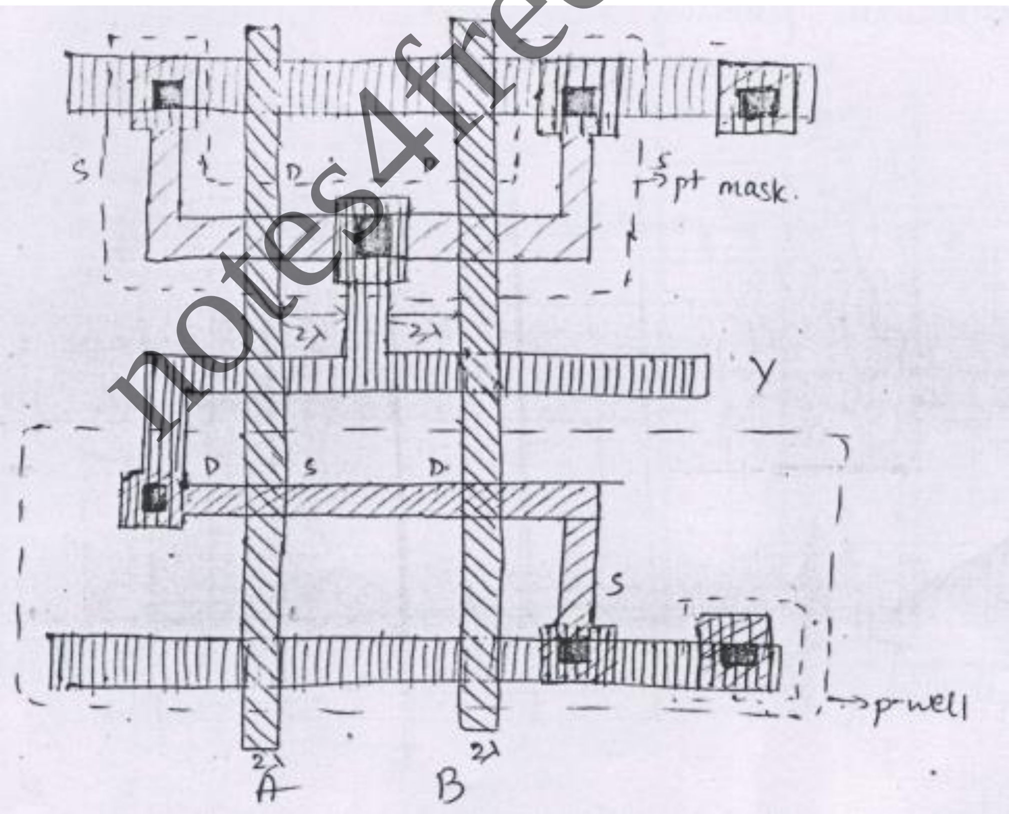




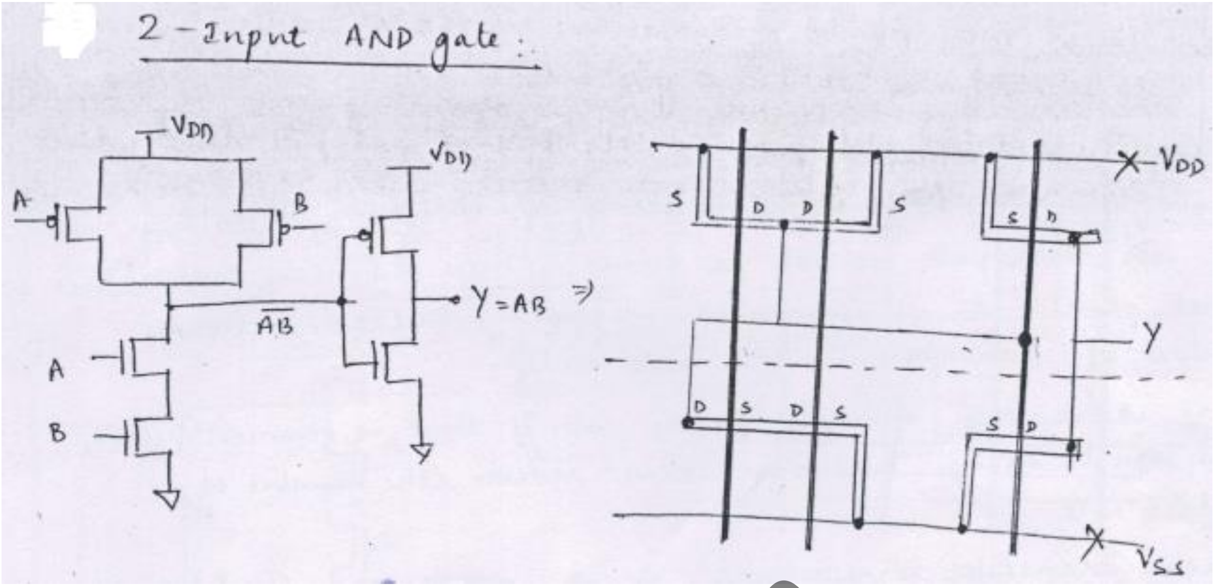




Diffusion lines running horizontal and polysilicon lines in vertical direction**.**



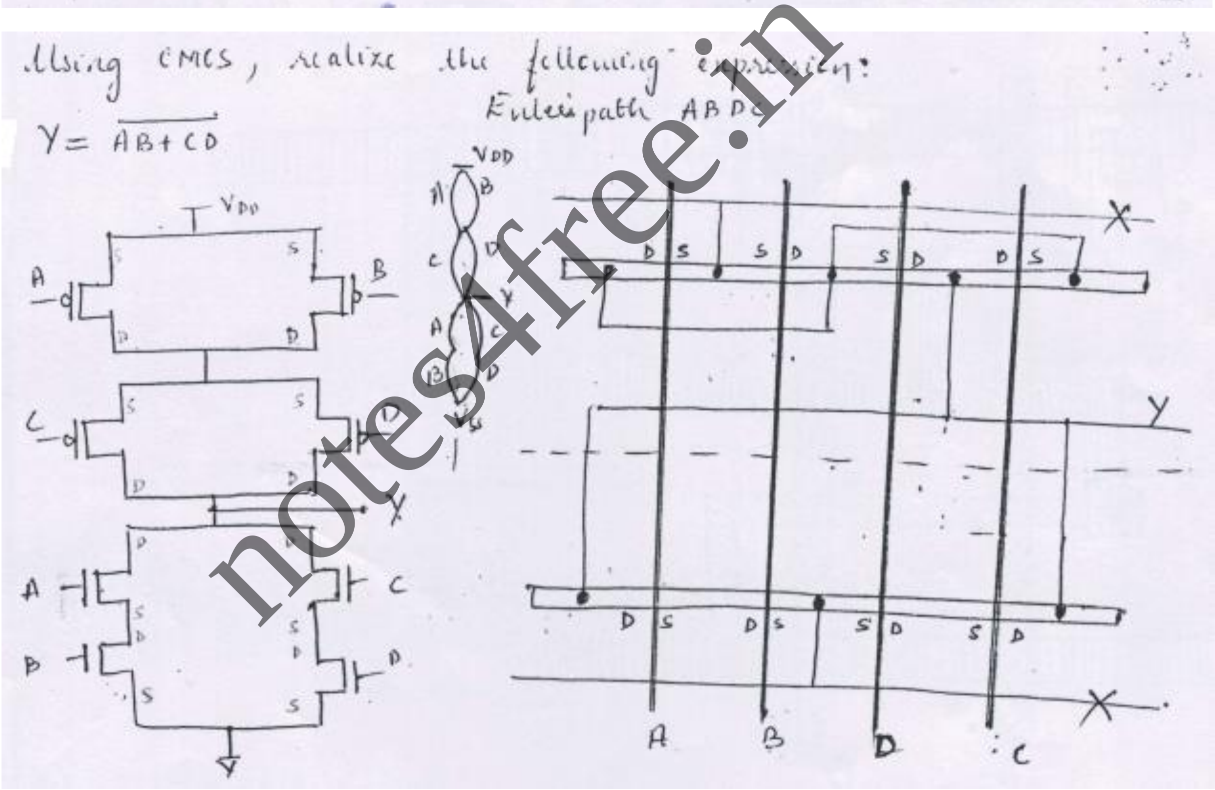
2-Input AND gate:



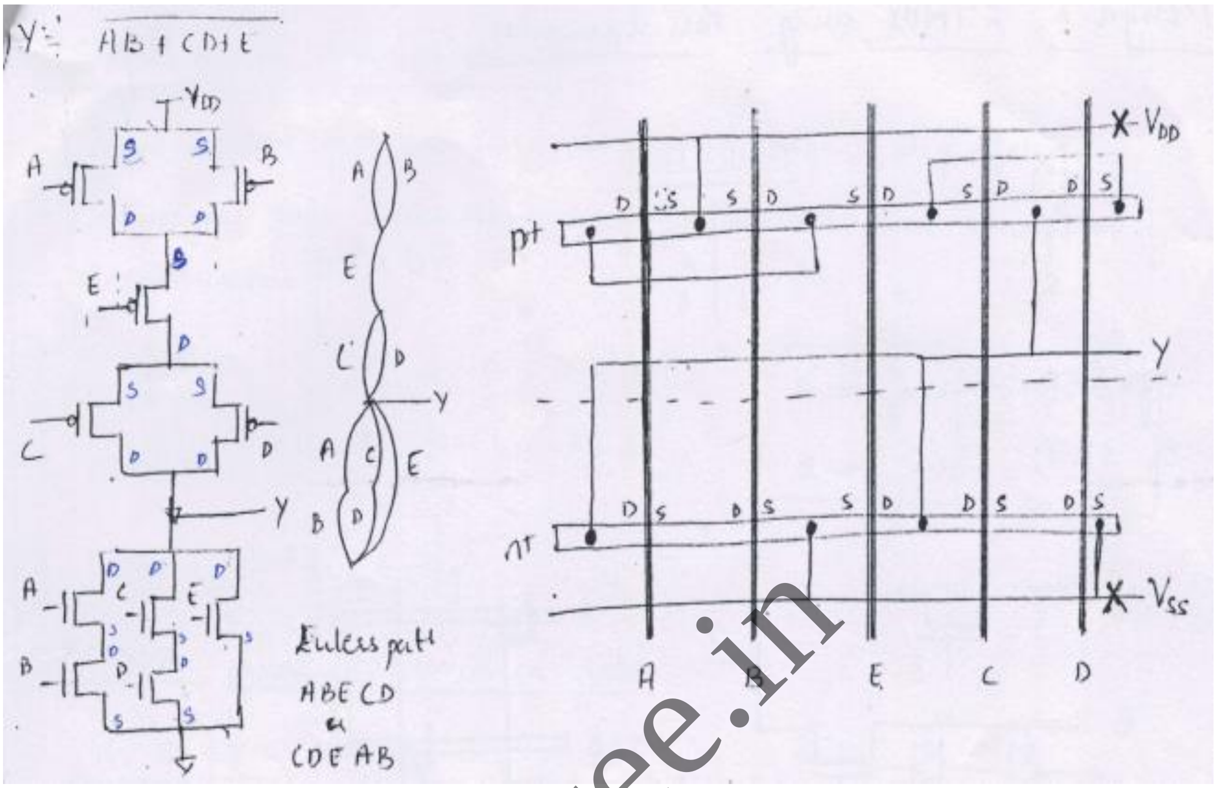
Using CMOS, realize the following expression:

$Y = \overline{AB + CD}$

Euler path ABDC



- When more number of inputs available, Euler's path is determined to know gate ordering.
- Advantage of using Euler's path is to that a common diffusion line can be used which reduces number of contact cuts.
- Uninterrupted path in both pull-up and pull down network represents optimized gate ordering which helps in drawing layout without breaking the diffusion layer.



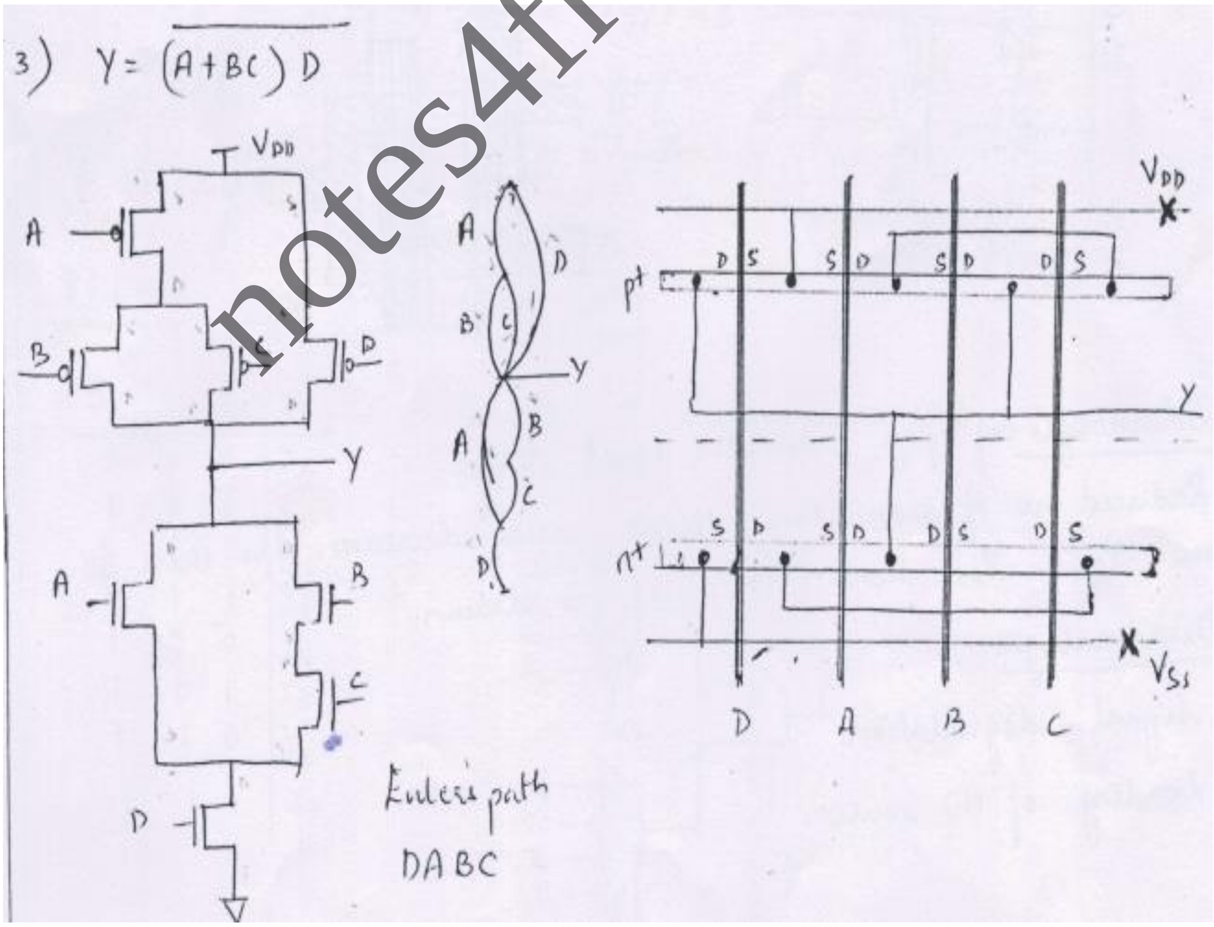
$Y = \overline{AB+CD+E}$

Eulers path

ABECD

&

CDEAB

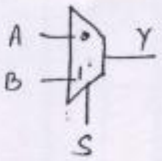


3) $Y = \overline{(A+BC)D}$

Eulers path

DABC

Design a 2:1 MUX using Pass transisters..



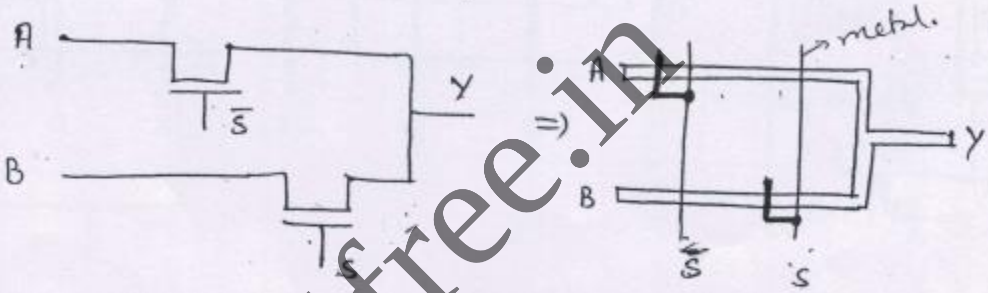
Truth table

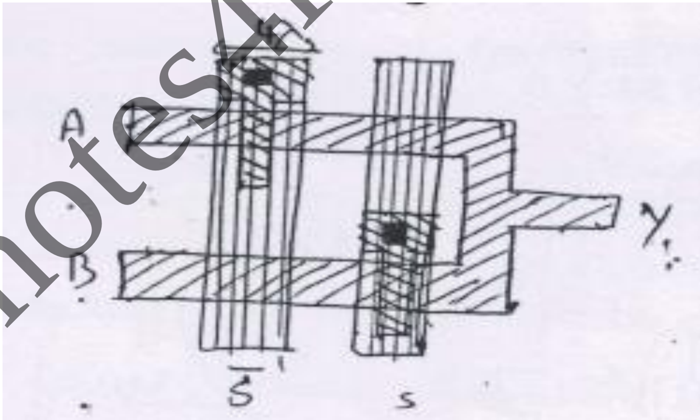
S	Y
0	A
1	B

$Y = \bar{S}A + SB$

if $S = 0$, then $Y = A$

if $S = 1$, then $Y = B$



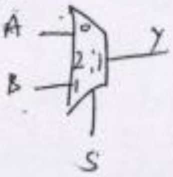


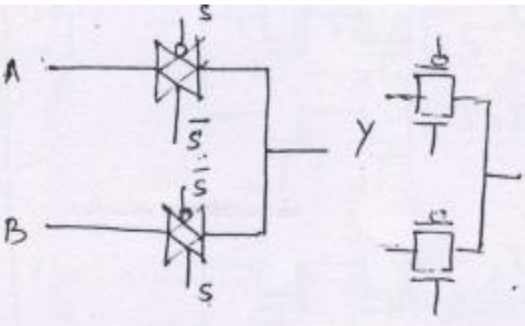
2:1 MUX using transmission gates

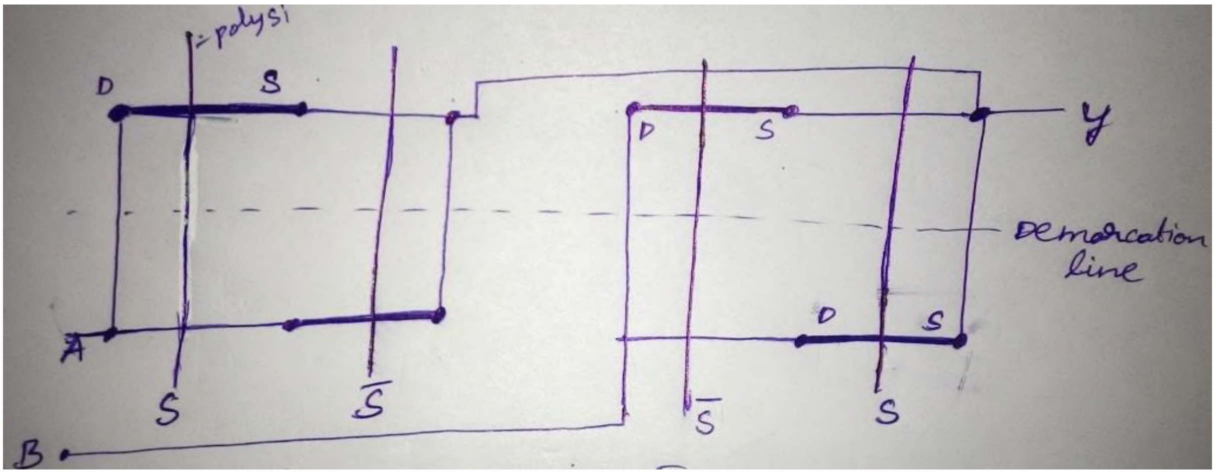
S	Y
0	A
1	B

if $S = 0$, $Y = A$

if $S = 1$, $Y = B$.

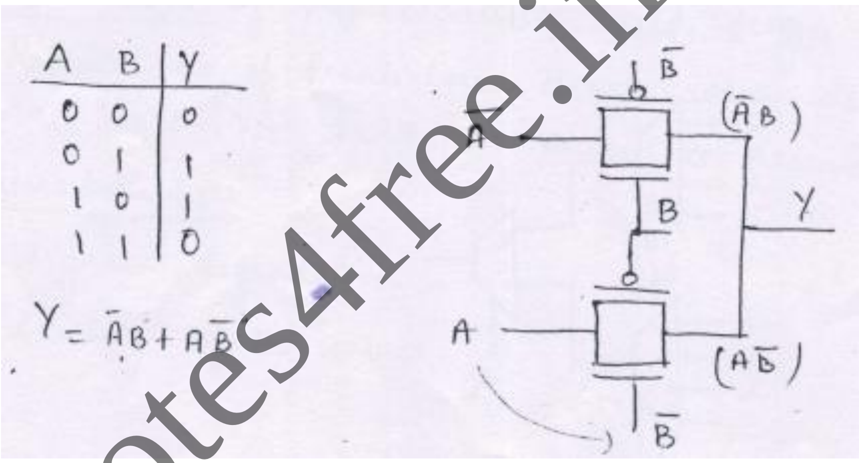






Stick diagram of 2:1 MUX using transmission gates.

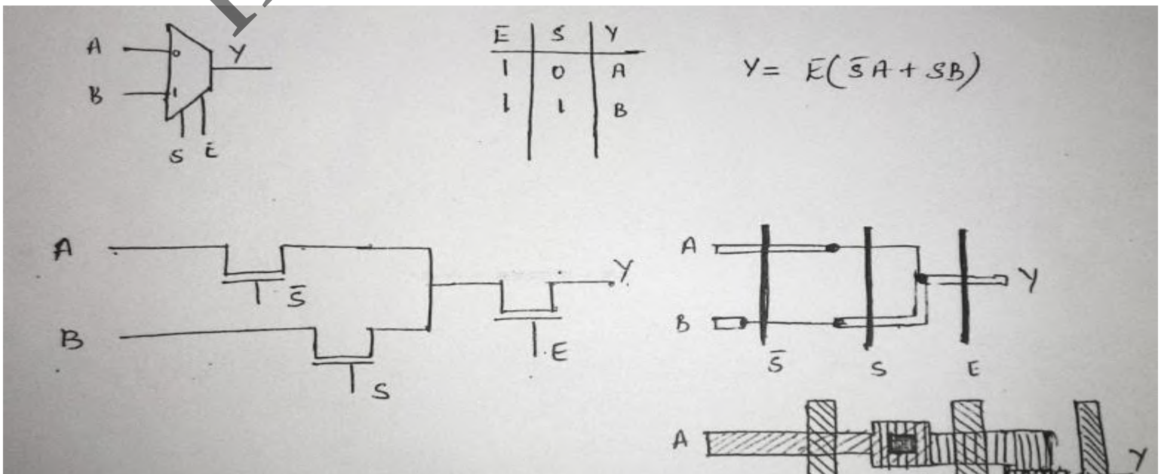
Two input XOR gate realization using transmission gates.



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

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

Two way selector with enable



E	S	Y
1	0	A
1	1	B

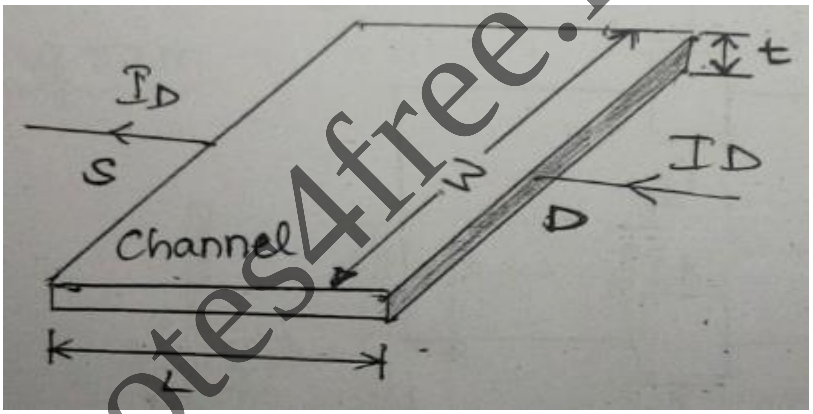
$Y = E(\bar{S}A + SB)$

Basic circuit concepts

- In MOS technology, Active devices are dealt with some measurement.
- Wiring up of circuits is done through various conductive layers which is produced by MOS Processing.
- Therefore it is necessary to be aware of resistive and capacitive characteristics of each layers.
- For evaluating the effects of wiring, input and output capacitances, sheet resistance and standard unit capacitances are used.
- Further delay associated with wiring, inverters are evaluated by the term delay unit τ.

Sheet Resistance R_s

Consider a transistor with a channel having resistivity ρ, width W, thickness t and length between source and drain is L.



Resistance of the channel between drain and source is expressed as.

$$R_{DS} = \frac{\rho.Length}{Area\ of\ cross\ section} = \frac{\rho.L}{t.W}$$

$$R_{DS} = R_s \frac{L}{W}$$

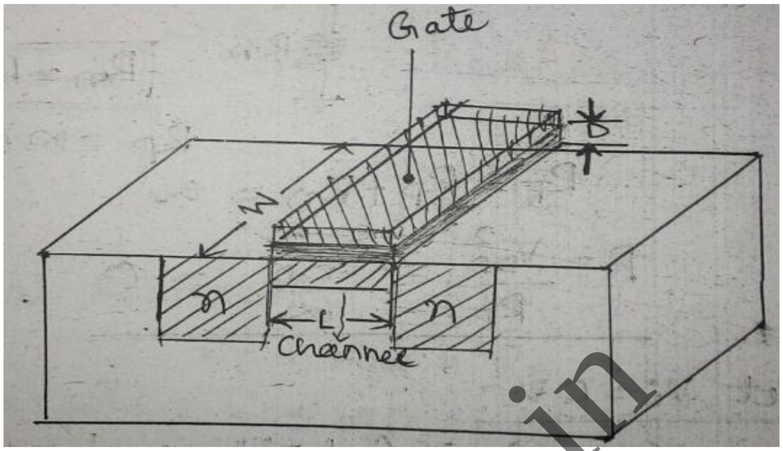
Where $R_s = \frac{\rho}{t}$ is a constant and it is called sheet resistance.

From the above equation, sheet resistance can be defined as resistance of the channel whose length and width are equal.

R_s is completely independent of area square. Ex: 1µm per side square slab of material has exactly same resistance as 1cm per side square slab of same material if thickness is same.

Area capacitance

In between gate and channel exists a capacitance and it is called gate capacitance and denoted by C_g.



From the above diagram.

$$C_g = \frac{\epsilon_0 \epsilon_r \, A}{D}$$

A is area of the channel or surface area of the gate

$$\frac{C_g}{A} = \frac{\epsilon_0 \epsilon_r}{D} \; pF/(\mu m)^2$$

$$C_A = \frac{\epsilon_0 \epsilon_r}{D}$$

ϵ_0 = permittivity of free space = 8.854x 10^{-12} F/m.

ϵ_r= relative permittivity of a given material

$D =$ thickness of sio2 constant for a given technology.

Area capacitance is defined as capacitance per unit area at the gate of transistor and denoted by C_A.

Standard unit of capacitance ($\square C_g$)

The standard unit of capacitance is defined as the capacitance at the gate of 1:1 transistor.

Ex: consider a 1:1 transistor where L = 2λ and W = 2λ.

Gate area of transistor = L x W

A = 2λ x 2λ = $(2\lambda)^2$

Actual capacitance at the gate of transistor C = $C_A . A = C_A . (2\lambda)^2$

C = $4\text{x}10^{-4}\ PF/(\mu m)^2 . (2\lambda)^2$

Consider 5 μm technology, i.e: 2λ = 5 μm

C = $4\text{x}10^{-4}\ pF/(\mu m)^2 . (5\ \mu m)^2$ = $4\text{x}10^{-4}$ x 25 = 0.01 pF

C = $1\square C_g$

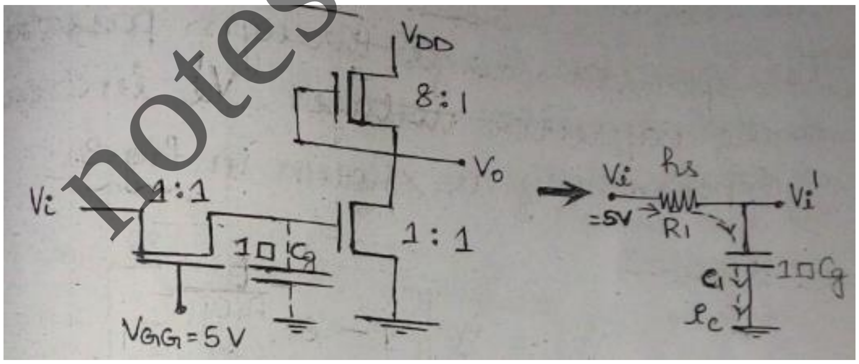
Standard Delay Unit (τ)

Time delay is measured in terms of standard unit τ.

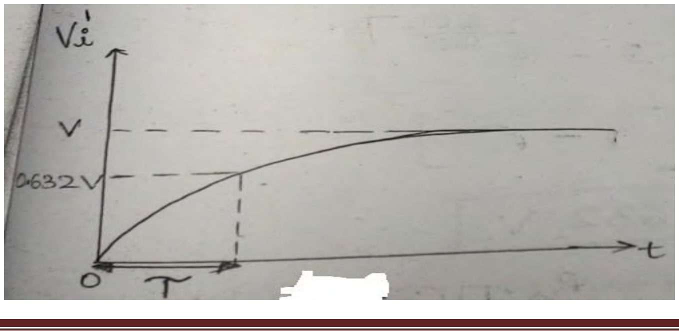
It is defined as product of R_s and C_g. i.e: $\tau = R_s . \square C_g$

<u>Measurement of τ.</u>

Consider nmos driven by pass transistor shown in below figure and the dimensions are indicated. Pass transistor is ON for given gate voltage V_{GG}. Pass transistor is represented by R_s due to its equal length and width. Pull down transistor of inverter is represented by capacitance $\square C_g$. Since pull down has minimum dimensions.



τ is defined as time taken by capacitor to charge from 0 to 63.2% of maximum value as shown in below figure.

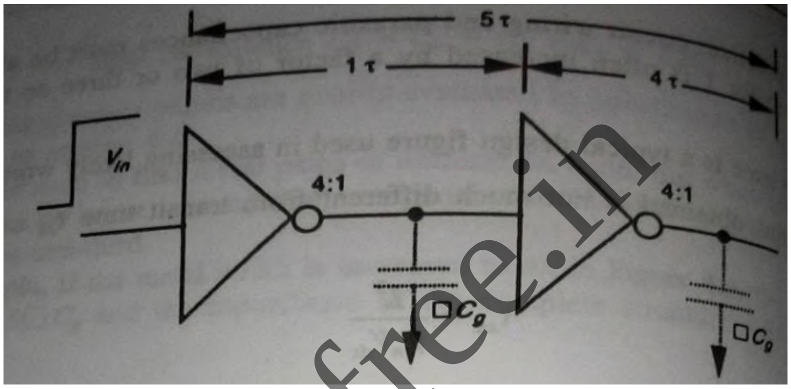


Inverter Delays

Consider basic 4:1 nmos inverter. To achieve 4:1 $Z_{pu}\ to\ Z_{pd}$ ratio, R_{pu} will be $4R_{pd}$. Clearly resistance R_{pu} value is $R_{pu} = 4\ R_s$ = 40KΩ. Meanwhile R_{pd} value is 10KΩ.

Delay associated with inverter depends on ON and OFF condition of transistors.

Consider a pair of cascaded inverter, delay in this pair will be constant irrespective of sense of logic level transition. The overall delay of nmos inverter is τ + 4τ = 5τ. Shown in below figure.

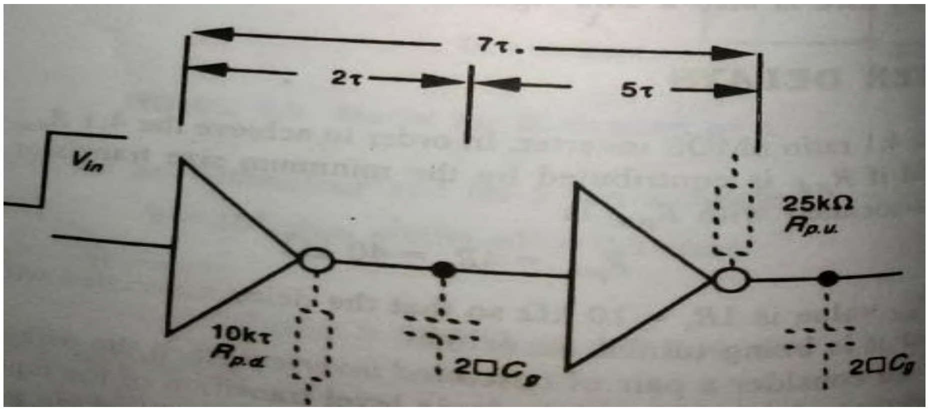


In general term delay through nmos inverter pair is given by $T_d = (1 + Z_{pu}/Z_{pd})$ τ

So single 4:1 inverter exhibits asymmetric delays, delay in turning on τ (capacitor discharging condition) and delay in turning off is 4 τ (capacitor charging condition). Asymmetry becomes worse for inverter with 8:1 ratio.

For CMOS inverter, nmos rules no longer applies, but we need to consider natural asymmetry of equal size pull up and pull down transistors.

Gate capacitance is double compare to nmos inverter since input is connected to both transistors and delay associated with pair of minimum size inverters is shown in below figure.



Asymmetry of resistance is eliminated by increasing the width of p- device channel by factor of two or three, but gate capacitance increases by the same factor.

Driving large capacitive loads

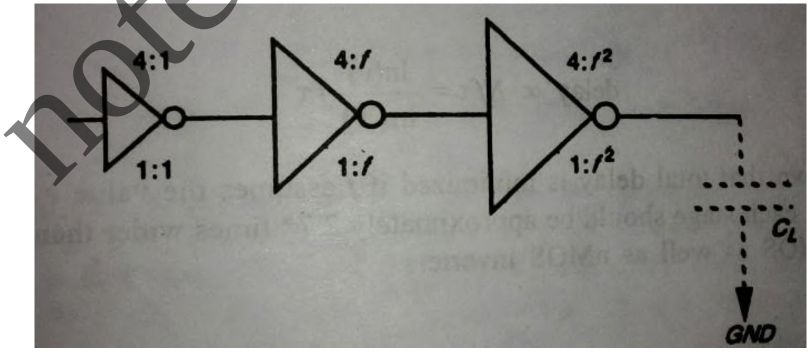
- A large capacitive loads problem arises when a signal to be transmitted from On chip to Off chip destinations.
- Off chip capacitance is is generally higher than On chip $\square C_g$.And it is denoted by C_L.

$$C_L \geq 10^4 \square C_g$$

- A capacitance of this order to be driven through low resistance otherwise long delays will occur.

Cascaded Inverters as drivers

- Inverters to drive large capacitive loads resistance associated with pull up and pull down transistors to be low.
- Low resistance values of Z_{pu} and Z_{pd} implies low L:W ratio or channel width must be made wider to reduce channel resistance but consequently inverter occupies large area.
- Gate area LxW is more significant and large capacitance present at input which slows down rate of change of voltage at input.
- Remedy to use N cascade inverter is by maintaining L to a minimum feature size and width of each successive stage is increased by factor f as shown in below figure.



- With increase in width factor increases capacitive load at input side and area occupied by the inverter also increases.
- The rate of width increase influence on number of stages to be cascaded to drive particular C_L value.
- Total delay associated with nmos pair is 5 τ and cmos pair is 7 τ.

Let y = $C_L / \square C_g = f^N$, f and N are interdependent.

To determine value of f to minimize overall delay for given y

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

$$N=\frac{\ln(y)}{\ln(f)}$$

For N even, total delay = $\frac{N}{2}\,5\,f\,\tau = 2.5\,f\,\tau$ (nmos) or

$$= \frac{N}{2}\,7\,f\,\tau = 3.5\,f\,\tau \text{ (cmos)}$$

In all cases, delay $\dot{\alpha}$ N $f\,\tau = \frac{\ln(y)}{\ln(f)}\,f\,\tau$

- Total delay is minimized If f assumes the value e. i.e: each stage is approximately 2.7 times wider than its predecessor and it is applicable for both cmos and nmos inverters.

Thus assuming $f = e$, we have

Number of stages N= $\ln(y)$

And overall delay t_d

N even: t_d= 2.5 N τ (NMOS) or t_d= 3.5 N τ (CMOS)

N odd: t_d= [2.5 (N-1)+1]e τ (NMOS) or t_d= [3.5 (N-1)+1]e τ (CMOS)

For ΔVin which indicates logic 0 to 1 transistion of Vin.

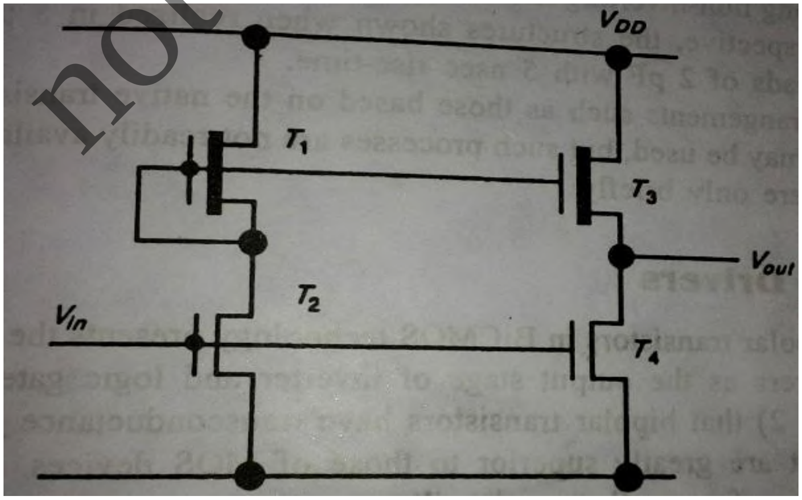
t_d= [2.5 (N-1)+4]e τ (NMOS) or t_d= [3.5 (N-1)+5]e τ (CMOS)

For ΔVin which indicates logic 1 to 0 transistion of Vin.

Super buffers

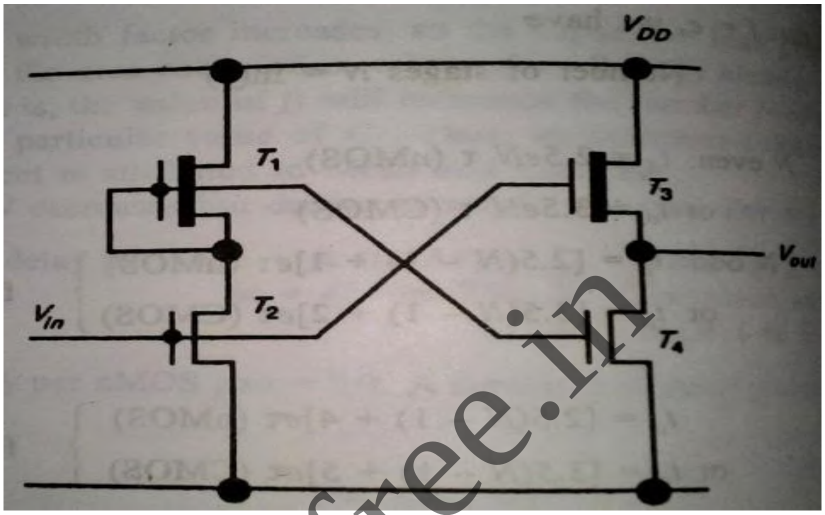
- Asymmetry of conventional inverter gives rise to significant delay problems when used to drive large capacitive loads.

Common approach used in nmos inverter is to use super buffers an inverting type nmos super buffer is shown in figure.



- Consider input Vin = 1, the inverter formed by T1 and T2 is turned On and thus gate of T3 is pulled down to zero volts with small delay. So T3 is in cut off and T4 is turned On and output is pulled down.

- When Vin = 0, gate of T3 is allowed to rise to Vdd. Thus T4 turned Off, T3 is made to conduct with Vdd on its gate. The voltage applied to gate is twice the average voltage of conventional nmos inverter.
- Doubling effective Vgs will increase current, thus reduces the delay in charging capacitor at output, so symmetry is achieved.
- The Non-inverting type nmos inverter is shown in below figure.



BICMOS Inverter

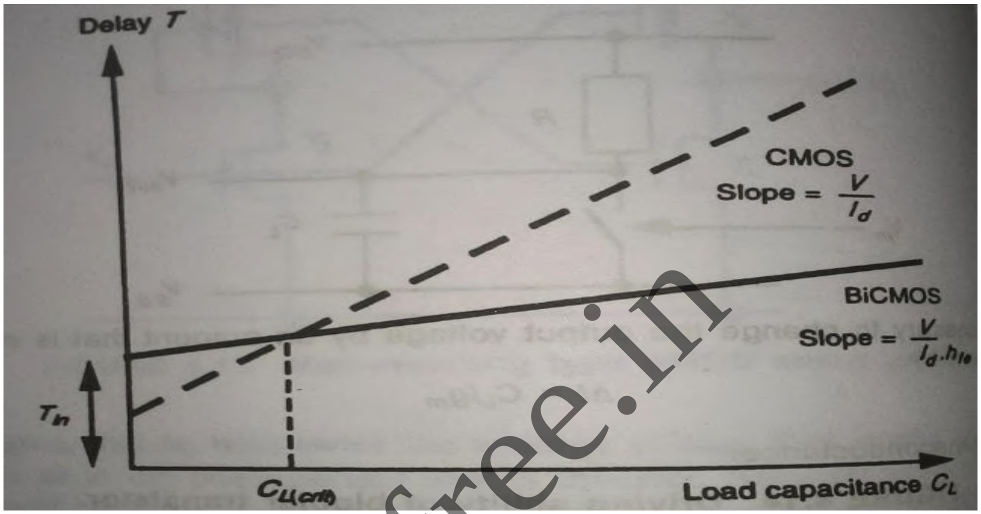
- Bipolar transistor availability in Bicmos technology presents possibility of using bipolar transistors as drivers at the output stage of the inverter.
- Transconductance and current/area characteristics are superior than MOS devices. so it has high current driving capability.
- Bipolar transistor has exponential dependence of output current on base emitter voltage which means transistor can operate with small input voltage swing compared to MOS transistors and switches large current.
- So the bipolar transistors have better switching performance results in small input voltage swing and switch large current.
- Switching performance of transistor driving capacitive load can be seen from simple model.
- The time required to change output voltage Vout by an amount equal to input voltage is given by

$$\Delta t = \frac{C_L}{g_m}$$

Where g_m is trans conductance of bipolar transistor. As g_m increases Δt decreases.

- Bipolar transistor delay has 2 main components Tin and T_L.
- Tin is the initial time required to charge the B-E junction of the transistor. It is time taken to charge the input gate capacitance.

- T_L is time taken to charge the output load capacitance C_L. This value is less for bipolar by factor of h_{fe}.
- As BJT has higher Tin, T_L is small and because of this faster charging takes place and helps in reducing the delay.
- Combined effect of Tin & T_L is in in graph. There is C_L critical load capacitance below which BICMOS driver is shown than CMOS driver.



- Delay of BICMOS is described by T = Tin+$(V/I_d)(1/\ h_{fe})C_L$.
- Delay for BICMOS inverter is reduced by a factor of h_{fe} when compared with CMOS inverter.

Module 3

Scaling of MOS Circuits

- High density chips in VLSI technology requires packing density of MOSFETS used is high and also the size of the transistors to be as small as possible. This reduction of size i.e. the dimension of MOSFET is referred as 'scaling'.
- It is expected that characteristics of the transistors will change with scaling and physical limitations will restrict the extent of scaling.
- Microelectronic technology is characterized in terms of indicators or figures of merit. The common figure of merits are
    - Minimum feature size
    - Number of gates on one chip
    - Power dissipation
    - Maximum operational frequency
    - Die size
    - Production cost
- The figure of merits can be improved by shrinking the dimensions of transistors, interconnections and the separation between features, adjusting doping levels and supply voltages.
- There are 2 types of size reduction/scaling strategies/ scaling models
    1. Full scaling or constant-field scaling.
    2. Constant-voltage scaling.
- Recently combined voltage and dimension model is presented. It is also called as ‘Lateral scaling’
- Two scaling factors 1/α and 1/β are used.
- 1/β is chosen as the scaling factor for supply voltage V_{DD} and gate oxide thickness D
- 1/α is used as scaling factor for other linear dimensions
- Scaling theory indicates that the characteristics of MOS devices can be maintained and the basic operational characteristics have to be preserved if the parameters of a device are scaled in accordance to a given criteria.

    - Constant field scaling: scaled device is obtained by applying dimension less factor to
        - All dimensions
        - Device voltages
        - Concentration densities

Scaling factors for device parameters:

The device dimensions are shown in Fig. 3.1

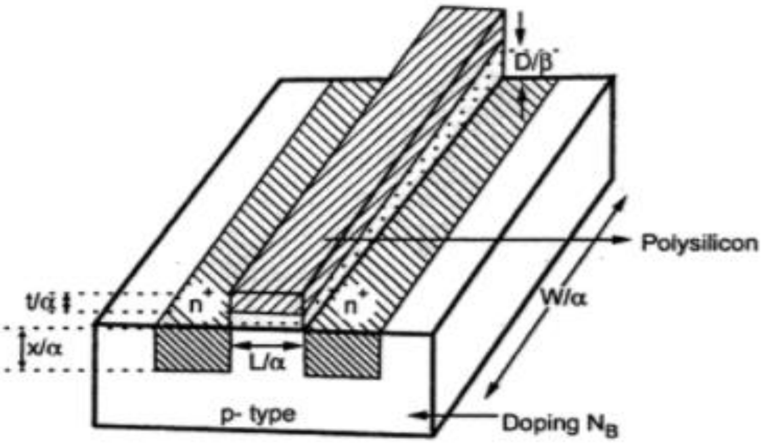


Fig. 3.1 scaled nMOS transistor

1. Gate Area A_g

$$A_g = L\ W$$

L and W defines channel length and width respectively. They are scaled by factor 1/α.

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

2. Gate Capacitance per Unit Area C_o or C_{ox}

$$C_o = \frac{\varepsilon ox}{D}$$

$\varepsilon_{ox}$is the permittivity of gate oxide and D is the thickness of gate oxide (thinox).
Thus $C_o$is scaled by $\frac{1}{1/\beta} = \beta$

3. Gate Capacitance C_g

Gate capacitance is given by, $C_g = C_o L\ W$
Thus $C_g$is scaled by $\beta * 1/\alpha^2 = \beta/\alpha^2$

4. Parasitic Capacitance C_x

$C_x$is proportional to A_x/d

Where d depletion width around source or drain, it is scaled by 1/α. A_x is area of depletion region around source and drain, it is scaled by $1/\alpha^2$

Thus C_x is scaled by $(1/\alpha^2)/(1/\alpha) = 1/\alpha$

5. Carrier Density in channel Q_{on}

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

$Q_{on}= C_oV_{gs}$ [C_o is scaled by β and $V_{gs}$is scaled by 1/β]
Thus $Q_{on}$is scaled by $\beta * 1/\beta = 1$

6. Channel Resistance R_{on}

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

L is scaled by 1/α, W is scaled by 1/α, $Q_{on}$is scaled as 1, μ is carrier mobility in the channel and is a constant.

Thus R_{on} is scaled by (1/α)/(1/α)*1/1=1

7. Gate Delay T_d

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

R_{on} is scaled to 1 and C_g is scaled by β/α^2

Thus T_d is scaled to $1*\beta/\alpha^2 = \beta/\alpha^2$

8. Maximum Operating Frequency f_o

$f_o = W/L *(\mu C_o V_{DD})/C_g$

or$f_o$is inversely proportional to T_d

Thus $f_o$is scaled as $1/(\beta/\alpha^2) = \alpha^2/\beta$

9. Saturation Current I_{dss}

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

C_o is scaled by β, μ and 2 are constants, W and L is scaled by 1/α, V_{gs} and $V_t$both voltages are scaled by 1/β.

Thus $I_{dss}$is scaled by $\beta*(1/\beta)^2 = 1/\beta$

10. Current Density J

$$J = \frac{Idss}{A}$$

A is the cross sectional area of channel in 'on' state and is scaled by $1/\alpha^2$ and I_{dss} is scaled by 1/β.

Thus J is scaled by $(1/\beta)/(1/\alpha^2) = \alpha^2/\beta$

11. Switching Energy per gate E_g

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

C_g is scaled by β/α^2 and V_{DD} voltage is scaled by 1/β.

ThusE_g is scaled by $(\beta/\alpha^2)*(1/\beta)^2 = 1/\alpha^2\beta$

12. Power Dissipation Per Gate P_g

P_g comprises 2 components, i.e., $P_g = P_{gs} + P_{gd}$

$P_{gs}$is the static component, given by $P_{gs} = (V_{DD})^2/R_{on}$

$P_{gd}$is the dynamic component given by $P_{gd} = E_g*f_o$

$V_{DD}$is scaled by 1/β, $R_{on}$is scaled by 1

E_g is scaled by $1/\alpha^2\beta$, $f_o$is scaled by α^2/β

Thus $P_{gs}$and P_{gd} both are scaled by $(1/\beta)^2$

Thus P_g is scaled by $1/\beta^2$

13. Power Dissipation Per Unit Area P_a

    $P_a$is defined as scaled by

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

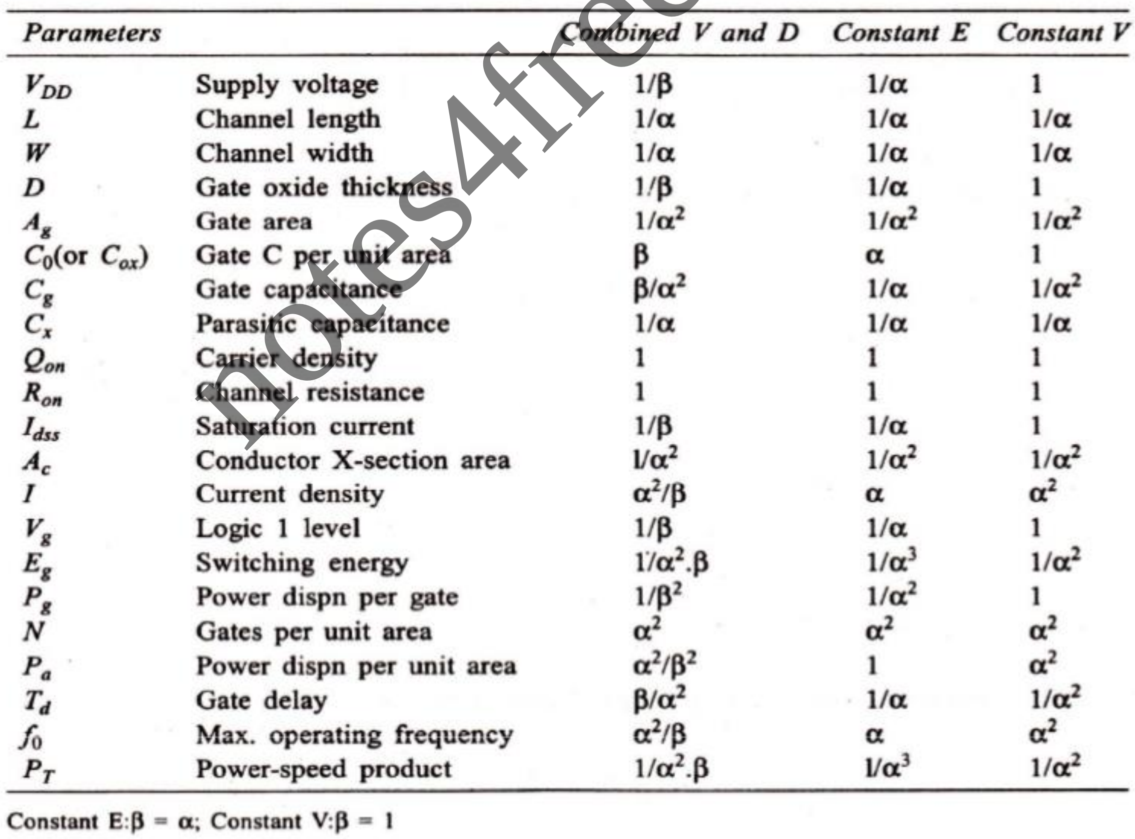
14. Power-Speed Product P_T

    $P_T = P_g * T_d$

    P_g is scaled by $1/\beta^2$ and T_d is scaled by β/α^2

    Thus $P_T$is scaled by $(1/\beta^2)*(\beta/\alpha^2) = 1/\alpha^2\beta$

Summary of Scaling effects of all the 3 models is given in the table below



Parameters		*Combined V and D*	*Constant E*	*Constant V*
V_{DD}	Supply voltage	$1/\beta$	$1/\alpha$	1
L	Channel length	$1/\alpha$	$1/\alpha$	$1/\alpha$
W	Channel width	$1/\alpha$	$1/\alpha$	$1/\alpha$
D	Gate oxide thickness	$1/\beta$	$1/\alpha$	1
A_g	Gate area	$1/\alpha^2$	$1/\alpha^2$	$1/\alpha^2$
C_0(or C_{ox})	Gate C per unit area	β	α	1
C_g	Gate capacitance	β/α^2	$1/\alpha$	$1/\alpha^2$
C_x	Parasitic capacitance	$1/\alpha$	$1/\alpha$	$1/\alpha$
Q_{on}	Carrier density	1	1	1
R_{on}	Channel resistance	1	1	1
I_{dss}	Saturation current	$1/\beta$	$1/\alpha$	1
A_c	Conductor X-section area	$1/\alpha^2$	$1/\alpha^2$	$1/\alpha^2$
I	Current density	α^2/β	α	α^2
V_g	Logic 1 level	$1/\beta$	$1/\alpha$	1
E_g	Switching energy	$1/\alpha^2.\beta$	$1/\alpha^3$	$1/\alpha^2$
P_g	Power dispn per gate	$1/\beta^2$	$1/\alpha^2$	1
N	Gates per unit area	α^2	α^2	α^2
P_a	Power dispn per unit area	α^2/β^2	1	α^2
T_d	Gate delay	β/α^2	$1/\alpha$	$1/\alpha^2$
f_0	Max. operating frequency	α^2/β	α	α^2
P_T	Power-speed product	$1/\alpha^2.\beta$	$1/\alpha^3$	$1/\alpha^2$

Constant E:$\beta = \alpha$; Constant V:$\beta = 1$

Subsystem Design Processes

Here the chapter deals with design of a digital system using a top down approach. The system considered is a 4 bit microprocessor.

The microprocessor includes ALU, control unit, I/O unit and memory. Here only ALU or data path is considered. The data path by itself is further divided into subsystem and in this 'shifter' unit is considered.

General Considerations:

- ✓ The considerations provide ways of handling problems, provides way of designing and realizing systems which are too complex
- ✓ Also help in understanding and appreciating technologies. The considerations are as follows:
  - Lower unit cost : with different approaches available for same requirement lower unit cost is appreciable
  - Higher reliability: high levels of system integrations greatly reduces interconnections and this in turn provides good reliability.
  - Lower power dissipation, lower weight and lower volume: in comparison with other approaches.
  - Better performance: particularly in terms of speed power product.
  - Enhanced repeatability: if there are fewer process to be controlled in the whole system or very large part to be realized on a single chip, this reduces the repeatability which is appreciable
  - Possibility of reduced design/development periods: for more complex systems reduced development time is appreciable.

- ✓ Some Problems related with VLSI design are

1. How to design complex systems in a reasonable time and reasonable effort.
2. The nature of architectures best suited to take full advantage of VLSI and the technology
3. The testability of large/complex systems once implemented on silicon

For the problem seen the solution is as follows:

Problem 1 & 3 are greatly reduced if two aspects are followed.

- a) Top-down design approach with adequate CAD tools
- b) Partitioning the system sensibly
- c) Aiming for simple interconnections
- d) High regularity within subsystem
- e) Generate and then verify each section of the design
- Devote significant portion of total chip area to test and diagnostic facility

Problem 2 can be solved by

- Select architectures that allow design objectives and high regularity in realization

Illustration of Design Processes:

- Structured design begins with the concept of hierarchy. This involves dividing any complex function into less complex sub-functions. This can be done until bottom level referred to leaf cells is reached.
- This process is known as top-down design
- As a systems complexity increases, its organization changes as different factors become relevant to its creation
- Coupling can be used as a measure the sub-modules interconnection. Clever system aims at minimum sub-module interconnection resulting in independent design.
- Concurrency should be exploited so that all gates on the chip do useful work most of the time
- As technology is changing fast, the adaptation to a new process must occur in a short time.

General Arrangement of a 4-bit Arithmetic Processor:

The basic architecture of microprocessor is shown below.

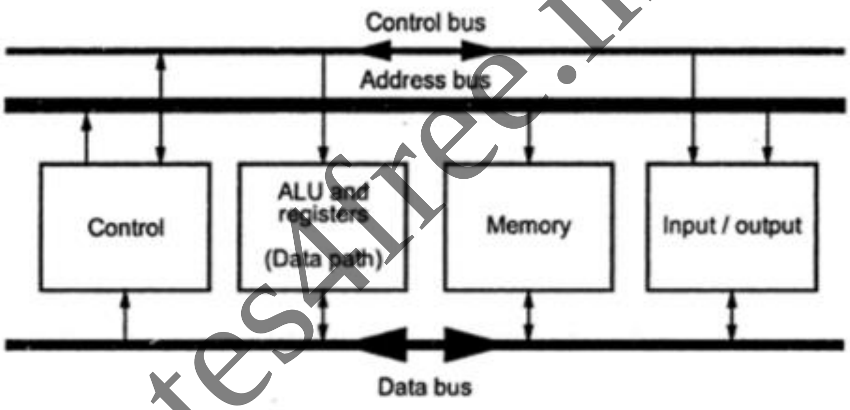


Fig. Basic digital processor structure

- It has a unit which processes data when applied at one port and gives output at second port.
- It is also possible that both data ports can be combined to form a single bidirectional port if storage facility is available in the data path.

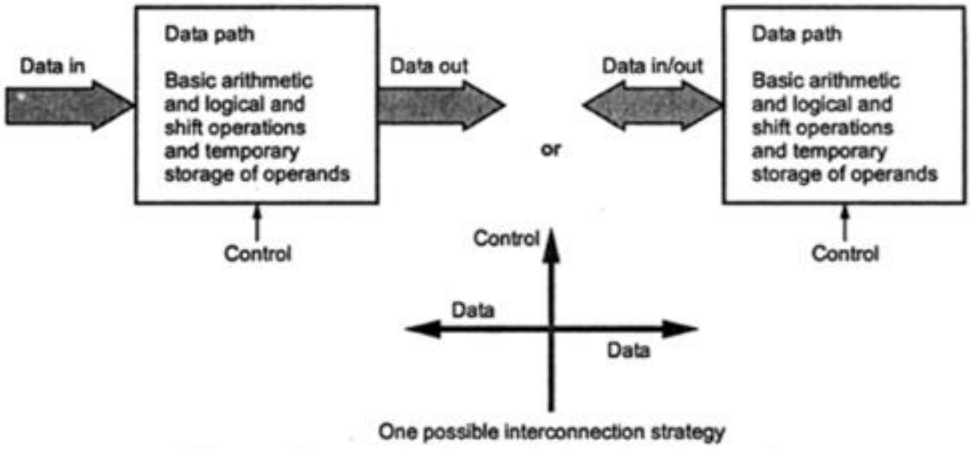
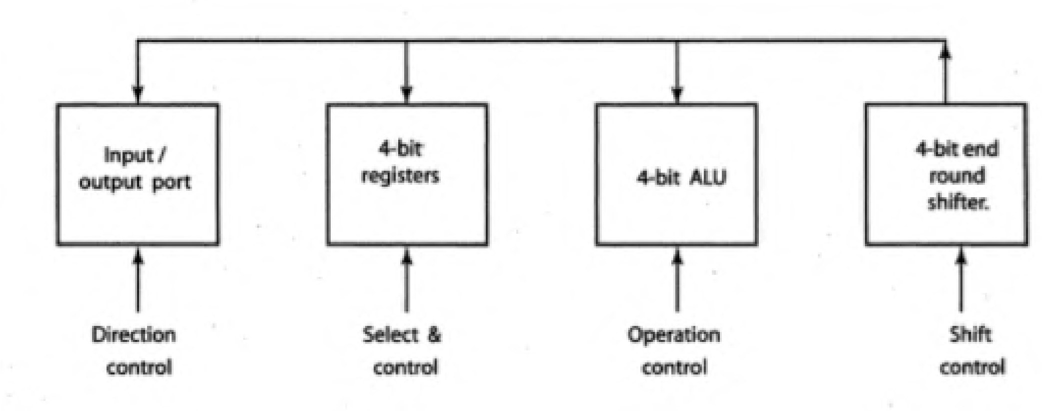

Data in
Data path
Basic arithmetic and logical and shift operations and temporary storage of operands
Data out
Control
or
Data in/out
Data path
Basic arithmetic and logical and shift operations and temporary storage of operands
Control
Control
Data
Data
One possible interconnection strategy

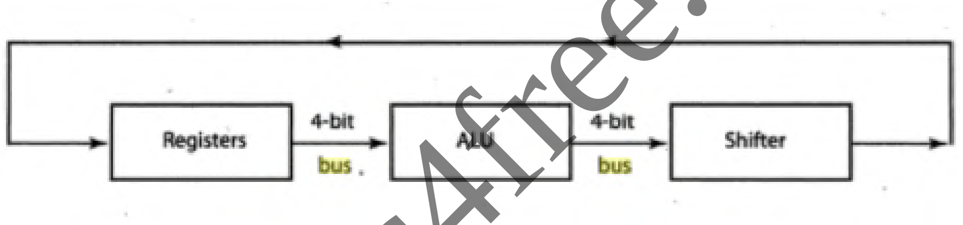
Fig. Communications strategy for data path

- The data path can be decomposed into having main subunits as it will be helpful in deciding the possible floor plan.



- The sub units can be linked with different bus architecture. It can be either one-bus, two-bus or three-bus architecture.

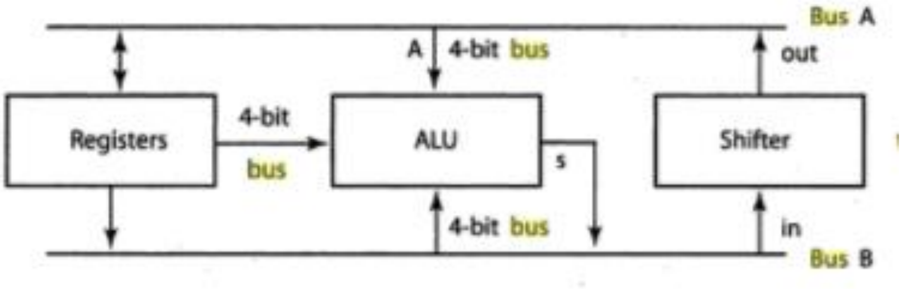
One bus architecture:



The sequence is:

1. First operand is moved from register to ALU and stored there.
2. Second operand is moved from register to ALU where operands are added (subtraction or any other arithmetic operation) and result is stored in ALU
3. The result is then passed through shifter and stored in register.
4. The process takes 3 clock cycles and this be fastened by using two bus architecture.

Two bus architecture:

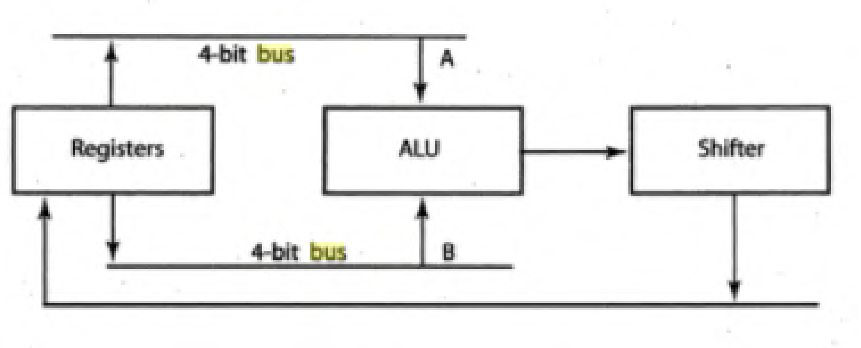


The sequence is:

1. Both operands (A & B) are sent from register(s) to ALU & are operated upon, result S in ALU.
2. Result is passed through the shifter & stored in registers.

3. This requires 2 clock cycle.

Three bus architecture:



1. Both operands (A & B) are sent from registers, operated in the ALU and result which is shifted is returned to another register. All these happen in same clock period.

- During the design process, care must be taken for allocating layers to various data path and guidelines. The guidelines are as follows:
  - Metal can cross poly or diffusion
  - Poly crossing diffusion form a transistor.
  - Whenever lines touch on the same level an interconnection is formed
  - Simple contacts can be used to join diffusion or poly to metal.
  - Buried contacts or a butting contacts can be used to join diffusion and poly
  - If 2nd metal layer is available, this can cross over any layer and can be used for power rails.
  - Two metal layers may be joined using a via
  - Each layer has particular electrical properties which must be taken into account
  - For CMOS layouts, p-and n-diffusion wires must not directly join each other nor may they cross either a p-well or an n-well boundary

Design of 4 bit shifter

- Ageneral purpose n-bit shifter should be capable of shifting n incoming data up to n-1 place either in a right or leftdirection.
- Shifting should take place in 'end-around' basis i.e., any bit shifted out at one end of a data word will be shifted in at the other end of the word. Thus the problem of right shift or left shift can be easily eased.
- It can be analyzed that for a 4-bit word, that a 1-bit shift right is equivalent to a 3-bit shift left and a 2-bit shift right is equivalent to a 2-bit left etc. Hence, the design of either shift right or left can be done. Here the design is of shift right by 0, 1, 2, or 3 places.
- The shifter must have:
  - input from a four line parallel data bus
  - Four output lines for the shifted data
  - The input data can be transferred to output lines using shift from 0, 1, 2 to 3 bits
- While designing the strategy should be decided. The chosen strategy here is data flow direction is horizontal and control signal flow direction is vertical.

- To meet the criteria a 4 × 4 crossbar switch is used. The MOS switch implementation of 4 × 4 crossbar switch is shown in the Fig below.

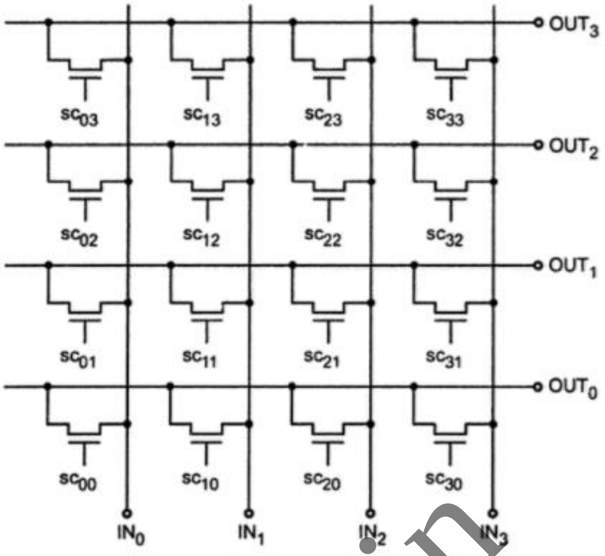


Fig. 4 × 4 crossbar switch

- To drive each cross bar switch - 16 control signals SC_{00} to SC_{15} are provided to each transistor switch.
- Arrangement is general and can be expanded to accommodate n-bit inputs/outputs.
- In this arrangement any input can be connected to any or all the outputs.
- If all switches are closed all inputs are connected to all outputs and it forms a short circuit.
- As it needs 16 control signal it increases the complexity and to reduce the complexity, the switch gates are coupled in groups (in this case it is grouped into 4)
- Here 4 groups of 4 is formed which corresponds to shift 0, shift 1, shift 2 and shift 3 bits.This arrangement is called 'barrel shifter'. (In this only 4 control signals is needed)

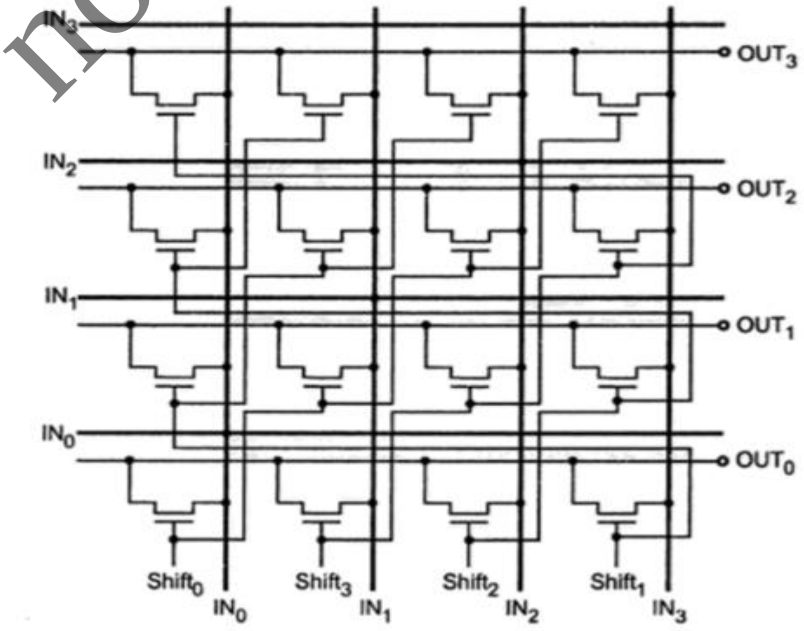
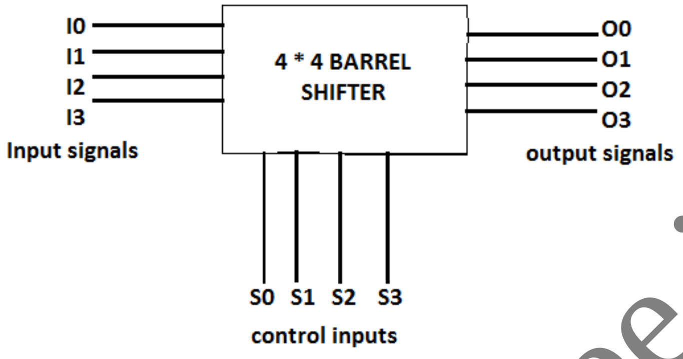


Fig. 4 × 4 barrel shifter

- The interbus switches have their gate inputs connected in a staircase fashion in groups of four and there are now four shift control inputs which must be mutually exclusive in the active state.
- CMOS transmission gates may be used in place of the simple pass transistor switches if appropriate. Barrel shifter connects the input lines representing a word to a group of output lines with the required shift determined by its control inputs (sh0, sh1, sh2, sh3). Control inputs also determine the direction of the shift. If input word has n – bits and shifts from 0 to n-1 bit positions are to be implemented.



SH0	SH1	SH2	SH3	SH4
O0	O1	O2	O3	O0
O1	O2	O3	O2	O1
O2	O3	O0	O1	O2
O3	O0	O1	O0	O3

Block diagram of barrel shifter and shifting of data is shown in the table.

Illustration of the Design Process

Regularity: It is an essentiality of any design. It reduces design efforts required for a system. Regularity of any particular design can be gauged as

$$Regularity = \frac{Total\ number\ of\ transistors\ on\ the\ chip}{Number\ of\ transistor\ circuits\ that\ must\ be\ designed\ in\ detail}$$

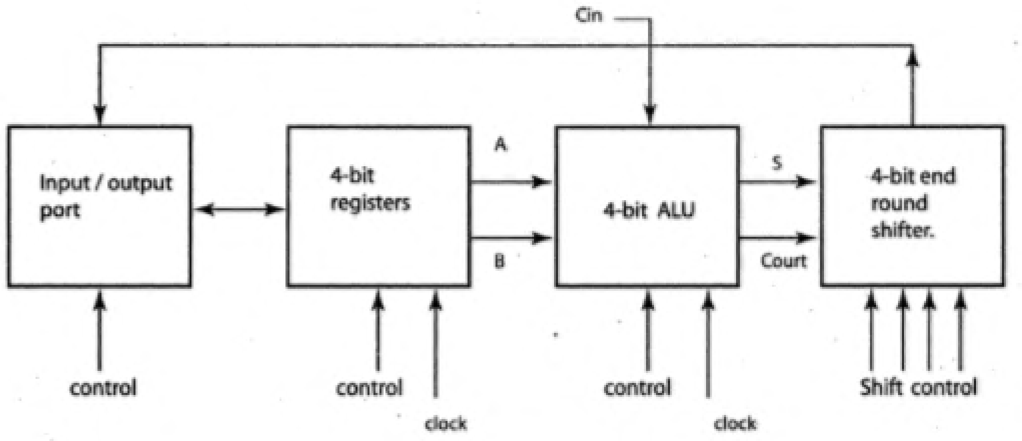
- For 4×4 bit barrel shifter, regularity factor is given by

  $$Regularity = \frac{16}{1}$$

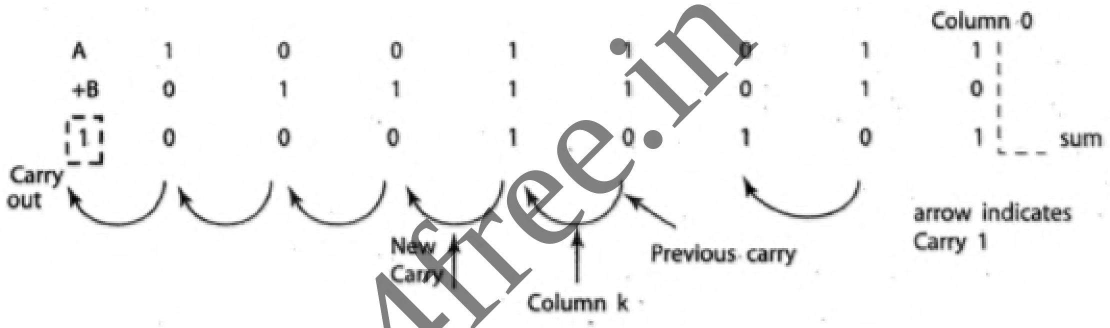
  In case of barrel shifter it needs only one transistor designing and the same can be applied to all the 16 transistors.
- Higher the regularity lesser will be the design effort. Good system design achieve regularity factor of 50 or 100.

Design of ALU subsystem

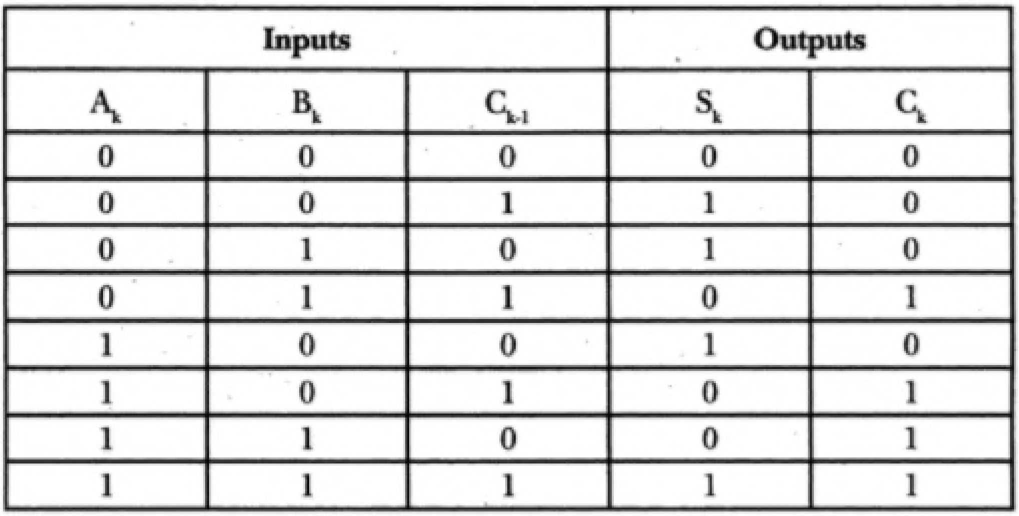
- Heart of the ALU is the adder circuit
- 4 bit adder will perform the sum of 2 4-bit number and here it is assumed that parallel operation form is done.
- The input to the adder needs two 4-bit buses, a single 4 bit is needed to move data from adder to shifter and other another 4-bit bus for shifting output back to register array. It also provide ‘carry out’ and possible ‘carry in’ signal
- The block diagram of 4 bit carry adder is shown below



- Considering an example of binary arithmetic operation.



- In any column we observe, there are 3 input. This number represents 3 inputs i.e., the corresponding input bits and previous carry/carry in
- If we consider kth column, it includes A_k, B_k representing input bits. C_{k-1} represents previous carry
- There are 2 outputs, sum and new carry. i.e., in Kth column S_k and C_k respectively.



Inputs			Outputs	
A_k	B_k	C_{k-1}	S_k	C_k
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- The sum carry equation we get is as follows:
- Sum $S_k= H_kC'_{k-1}+ H_k' C_{k-1}$

  New carry $C_k=A_k B_k + H_k C_{k-1}$

  Where H_k is the half sum given by $H_k= A'_k B_k+ A_kB'_k$

- The above sum and new carry equation can be implemented using 'AND-OR' or using 'EX-OR' gates. But in VLSI this implementation becomes complicated and alternate method is to find a standard element through which the above equations can be implemented.
- Analyzing the table of full adder again, we can see that

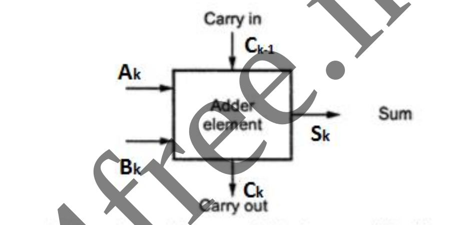
  For S_k, if $A_k= B_k$ then $S_k= C_{k-1}$

  else$S_k= C'_{k-1}$

  For C_k if $A_k= B_k$ then $C_k= A_k= B_k$

  else $C_k= C_{k-1}$

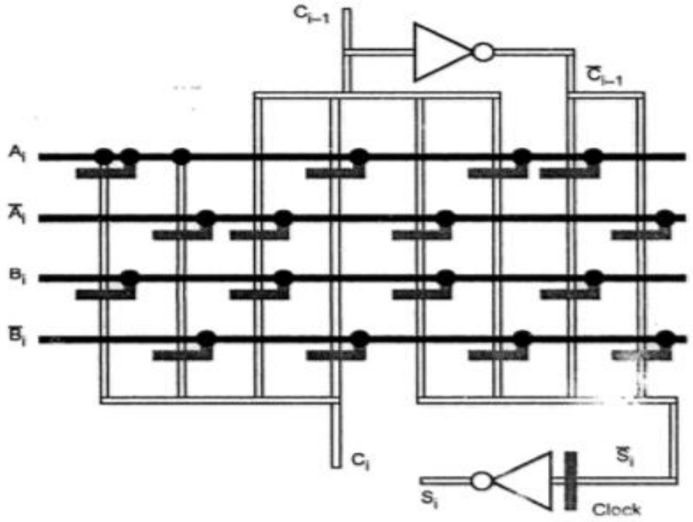
- A standard adder element (1 bit) can be implemented as shown below



n such elements would be cascaded to form an *n*-bit adder.

Adder element

- To form n-bit adder, n adder elements must be cascaded with carry-out of 1 element with carry-in of next most significant bit.
- The adder element can be implemented using multiplexer either in nMOS or CMOS technology. This method of implementation is easy to follow and results obtained are in the required forms.
- Implementation of adder using MUX (pass transistor) is shown in the last page. Below figure shows the stick diagram of the same.



Multiplexer (n-switches)-based adder logic with stored and buffered sum output

Implementing ALU function with adder element:

- An ALU must be able to add and subtract two binary numbers, perform logical operations such as AND, OR and Equality (EX-OR) functions.
- Subtraction can be easily implemented using the adder element. Suppose we need find A – B, then
  - Compliment B and add 1 to it to get B'
  - Add this to A i.e., A+B'
  - This gives subtraction using adder
- Here compliment for subtraction can be served from 'logical compliment' (negate)
- In order to keep architecture as simple as possible it would be better if all the logical operations can be performed on the adder.
- Considering the possibility

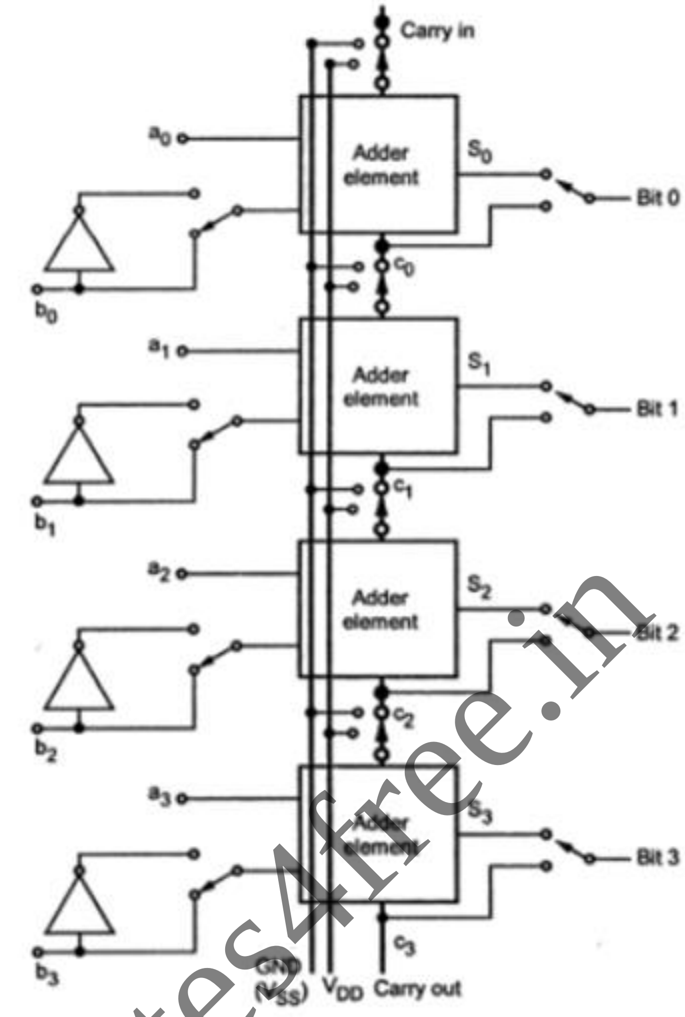
  We have the sum and new carry equations as

  Sum $S_k= H_kC'_{k-1}+ H_k' C_{k-1}$

  New carry $C_k=A_k B_k + H_k C_{k-1}$

  Where $H_k =A'_k B_k+ A_kB'_k$

- If in S_k, if C_{k-1} is held at logic 0 then the expression would be
  - $S_k= H_k .1+ H_k' 0$
  - $S_k= H_k = A'_k B_k+ A_kB'_k$
  - This represents the EX-OR function/operation
- If in S_k, if C_{k-1} is held at logic 1 then the expression would be
  - $S_k= H_k .0+ H_k'. 1$
  - $S_k= H'_k = A'_kB'_k+ A_k B_k$
  - This represents the EX-NOR function/operation
- If in C_k, if C_{k-1} is held at logic 0 then the expression would be
  - $C_k=A_k B_k + H_k0$
  - $C_k=A_k B_k$
  - This represents the AND function/operation
- If in C_k, if C_{k-1} is held at logic 1 then the expression would be
  - $C_k=A_k B_k + H_k. 1=A_k B_k + H_k$
  - $C_k=A_k B_k + A'_k B_k+ A_kB'_k$
  - $C_k= B_k+ A_kB'_k$
  - $C_k= B_k+ A_k$
  - This represents the OR function/operation
- Thus with suitable switching of carry line between adder element will give ALU logical functions.
- One such arrangement for both arithmetic and logical function is shown in figure..



4-bit ALU

Further considerations of Adders:

- Simple and direct implementation of adder circuit was observed now alternate form of adder equations will be observed.
- In the adder carry will be propagated when either $A_k$or B_k is high. Thus this equation can be written as

  $P_k = A'_k B_k + A_k B'_k = A_k \text{XOR } B_k$
- Also carry will be generated when both A_k and B_k are high. This equation can be written as

  $G_k = A_k B_k$
- With this the expression for sum $S_k$and carry C_k the expression can be written as

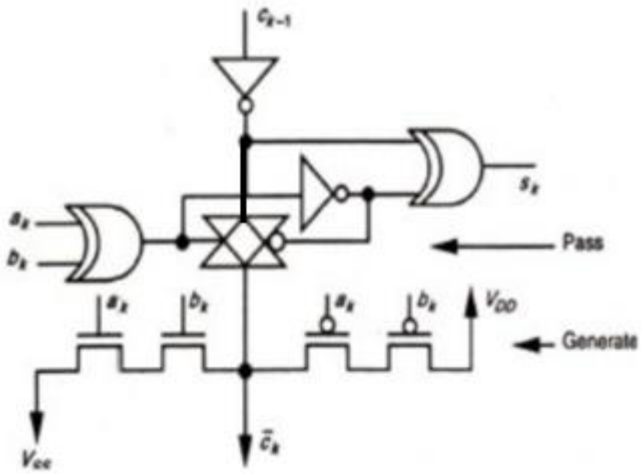
  $S_k = A'_k B_k C'_{k-1} + A_k B'_k C'_{k-1} + A'_k B'_k C_{k-1} + A_k B_k C_{k-1}$

  $\boxed{S_k = A_k \text{XOR} B_k \text{ XOR } C_{k-1}}$

  And $C_k = A_k B_k + A'_k B_k C_{k-1} + A_k B'_k C_{k-1}$

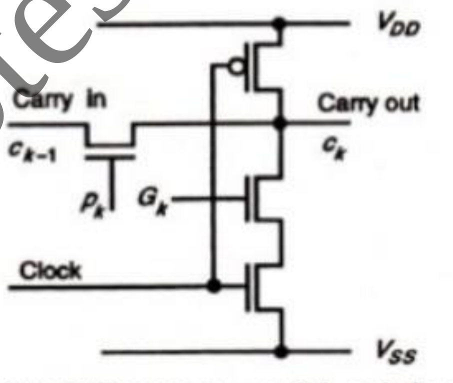
  $\boxed{C_k = G_k + P_k C_{k-1}}$

- Thus adder based on pass/generate carry concept is shown in the Fig
- Using this circuit carry can either be propagated to next stage or generated.



- Here propagate signal from ex-or gate is used to drive the transmission gate which acts like a controlling signal to propagate the carry signal. Only when P signal is high TG will be enabled and propagates C_{k-1}
- When P signal is 0 hen depending on the input A and B carry will be generated.
- The nMOS and pMOS pass transistor will generate the carry signal. (When A=B=1, carry is generated and when A=B=0 carry is not generated.)
- This is a direct realization which leads to the concept of carry chain and leads to popular arrangement known as Manchester carry chain.

Manchester Carry Chain:



Note in this case, $p_k = a_k \oplus b_k$ as before
but $G_k = \bar{a}_k . \bar{b}_k$

- It is a fast adder circuit, it is a carry propagate adder in which the delay taken by the carry to reach the last stage of output is reduced.
- As seen in the previous concept of passing the carry the transmission gate, the path can be precharged by the clock
- Then the path can be gated by the n-type pass transistor.
- When clock is 0, output will be charged to logic high due to pMOS. When clock is 1 pMOS will be off. If $P_k$is high, then the carry will be propagate.
- If P_k is 0, then C_{k-1} will not be propagated.

- Depending on inputs at G_k,(A_k= B_k =1) carry will be generated or (A_k= B_k =0) no carry generation.
- Even though Manchester carry cells are faster while cascading delay is observed. Cascading is done by connecting pass transistor in series.
- As n pass transistor is cascaded the delay also increases as square of n. thus in order to reduce delay buffers are included at after every 4 chain as shown in the block diagram

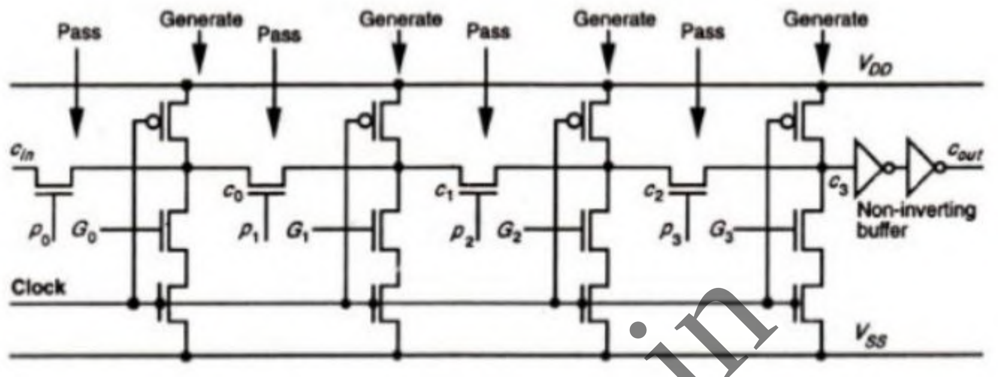


Fig. Cascading of Manchester carry adder

Adder Enhancement Techniques:

- In any adder element the previous carry bit is necessary to compute its own sum and carry out bits.
- For smaller adder (n<8) the delay observed for carry to propagate in ripple carry adder is small but at the n increases the delay is more and hence the time to find sum also increases.
- Thus some special techniques are used to improve the time taken for addition.
- This includes 3 methods
  - Carry Select Adder
  - Carry Skip Adder
  - Carry Look Ahead Adder

Note: All the above 3 techniques is a modification of ripple carry adder.

Carry Select Adder:

- In carry select adder the adder will be divided into group.
- It requires two identical parallel adders in each group except the least significant group as seen in the block diagram.
- Each group generates a group carry. Here, two sums are generated simultaneously. One sum assumes that the carry in is equal to one as the other assumes that the carry in is equal to zero.
- A mux is used to select valid result.
- It can be observed that the group carries logic increases rapidly when more high-order groups are added to the total adder length.
- This complexity can be decreased, with a subsequent increase in the delay, by partitioning a long adder into sections, with four groups per section, similar to the CLA adder.

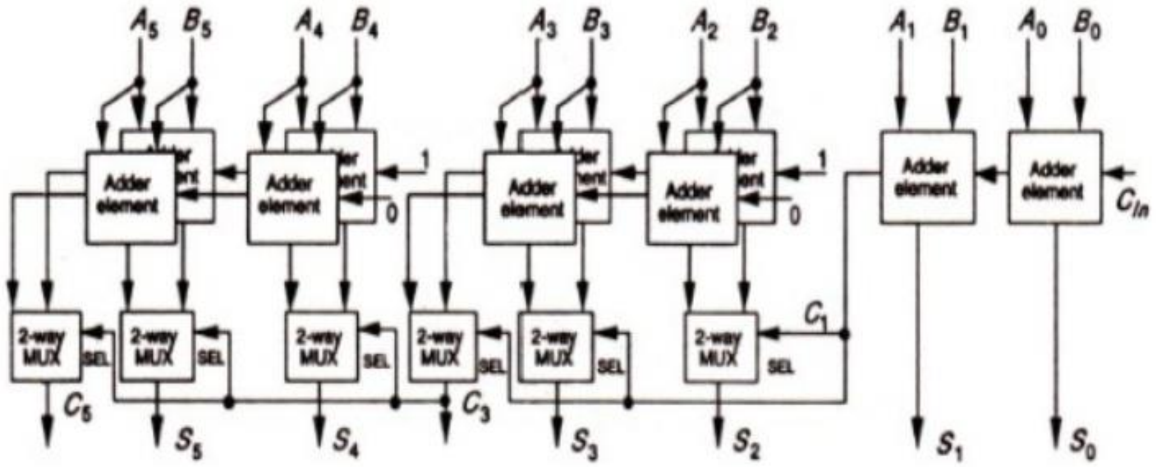


Fig. Block diagram of Carry Skip Adder

Optimization of Carry Skip Adder:

- For n bit ripple carry adder, the computational time T is given by
  T = K1n , where K1is the delay through 1 adder cell
- As carry select adder is a modification of carry skip which is having 2 adder in each block i.e., each block has 2 parallel paths. Thus computation time T becomes
  T = K1n/2 + K2 where K2 is the time needed by the mux to select the actual carry output.
- If there are many multiplexer then ripple through effect of carry is observed in mux rather than in the carry chain. Thus optimum value of mux should be selected for the size of each block.
- Suppose if there is an n-bit adder divided into M-blocks and each block contains P adder cells (n = M.P). The computation time for overall carry has 2 components
  - Propagation delay through the first block
  - Propagation delay through mux
    Thus, T = PK1 + (M – 1) K2
    Minimum value for T is seen when M=$\sqrt{nK1/K2}$

Note: except the first group all other groups have mux. If M is the number of groups the number of mux for n bit adder will be (M-1) and delay observed is (M -1)K2.

Carry Skip Adder:

- In a Ripple Carry Adder, if the input bits Ai and Bi are different for all position i, then the carry signal is propagated at all positions (thus never generated), and the addition is completed when the carry signal has propagated through the whole adder.
- In this case, the Ripple Carry Adder is as slow as it is large. Actually, Ripple Carry Adders are fast only for some configurations of the input words, where carry signals are generated at some positions
- Carry Skip Adders take advantage both of the generation or the propagation of the carry signal.
- They are divided into blocks, where a special circuit detects quickly if all the bits to be added are different. This signal is called ‘block propagation signal’

- If in the block, if A & B bits differ then the output i.e., block propagation signal = 1. If it is 1 then carry entering the block can bypass and transmit the carry to the block through multiplexer.
- Fig below shows 24 bit adder with 4 blocks and each block has 6 adder elements.

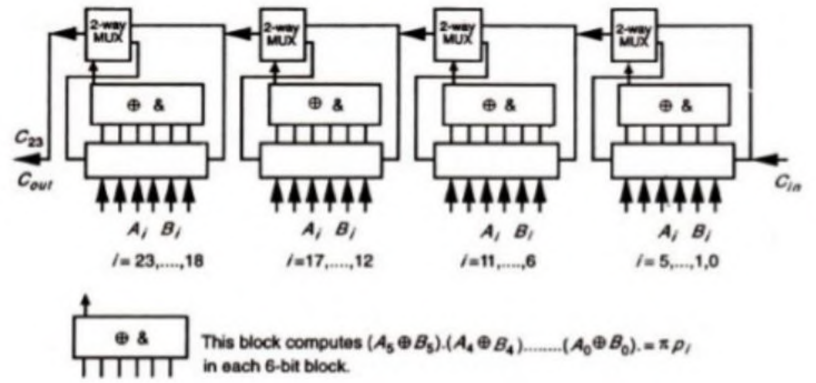


Fig. Block diagram of 24 bit carry skip adder

Optimization of carry skip adder:

- Let K1 represents the time needed by the carry signal to propagate through an adder cell, and K2 the time it needs to skip over one block. Suppose the N-bit Carry Skip Adder is divided into M blocks, and each block contains P adder cells. The actual addition time of a Ripple Carry Adder depends on the configuration of the input words. The completion time may be small but it also may reach the worst case, when all adder cells propagate the carry signal.
- In the same way, we must evaluate the worst carry propagation time for the Carry Skip Adder. One of the worst case of carry propagation is depicted in Figure.

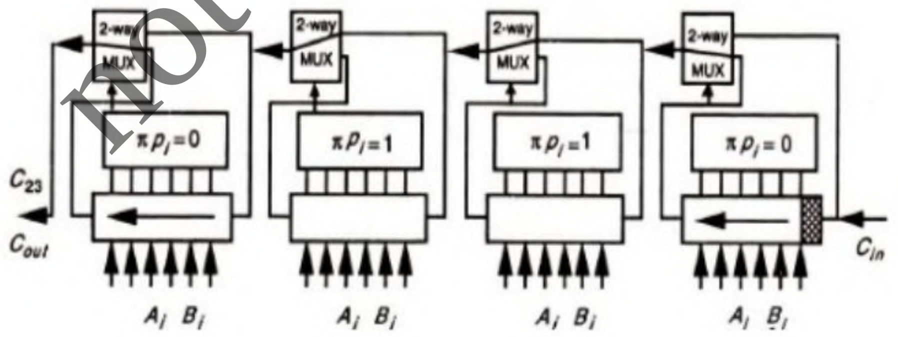


Fig. Worst case carry propagation for Carry Skip adder

- The configuration of the input words (for the worst case) is such that a carry signal is generated at the beginning of the first block. Then this carry signal is propagated by all the succeeding adder cells but the last which generates another carry signal.
- In the first and the last block the block propagation signal is equal to 0, so the entering carry signal is not transmitted to the next block.
- Consequently, in the first block, the last adder cells must wait for the carry signal, which comes from the first cell of the first block. When going out of the firstblock, the carry signal is distributed to the 2nd, 3rd and last block, where it propagates.

- In these blocks, the carry signals propagate almost simultaneously (we must account for the multiplexer delays).
- For the above condition the time to compute addition is given by

  T = 2 (P – 1) K1 + (M – 2) K2

Note: in the first block except the first bit all other bits have for carry propagation hence it (P – 1) and as the same is observed for last block it is 2(P-1) and delay is K1. Also the input carry signal does not skip over the first and last block hence K2 is multiplied with (M-2).

Carry Look Ahead Adder:

- This is another method to improve the throughput time of adder.
- Here the prediction of carry is done and based on this designing is done.
- With the carry generate $G_k= A_kB_k$ and carry propagate $P_k= (A_kXORB_k)C_{in}$the carry propagation can be avoided and carry outputs can be calculated.
- If n (number of adder) is large, the carry output bits increases and expression larger and complex. Thus ‘carry look ahead adder’ and ‘ripple carry adder’ are clubbed together. This can be seen in the block diagram.

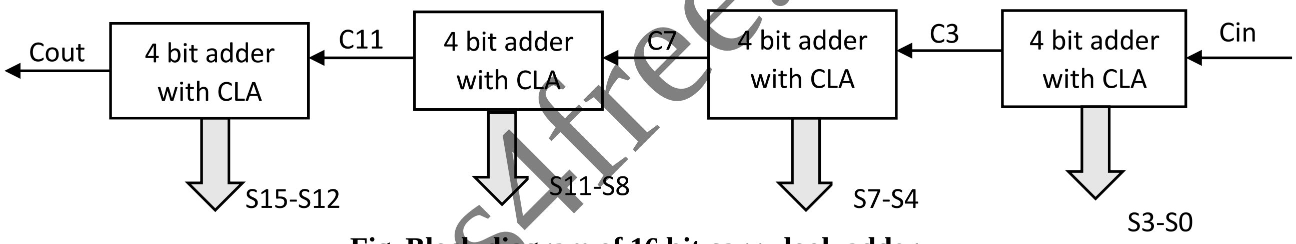


Fig. Block diagram of 16 bit carry look adder

- Considering the first block, carry out C3 can be calculated as follows

  $C_{out}= G_k+ P_kC_{in}$

  Thus for the first bit in first adder block is given by

  $$C_0 = G_0 + P_0C_{in}$$

Similarly for 2nd bit it is given by

$$C_1 = G_1 +P_1C_0 \quad \text{using } C_0 \text{ in the equation for } C_1$$
$$C_1 = G_1+ P_1G_0 +P_1P_0C_{in}$$

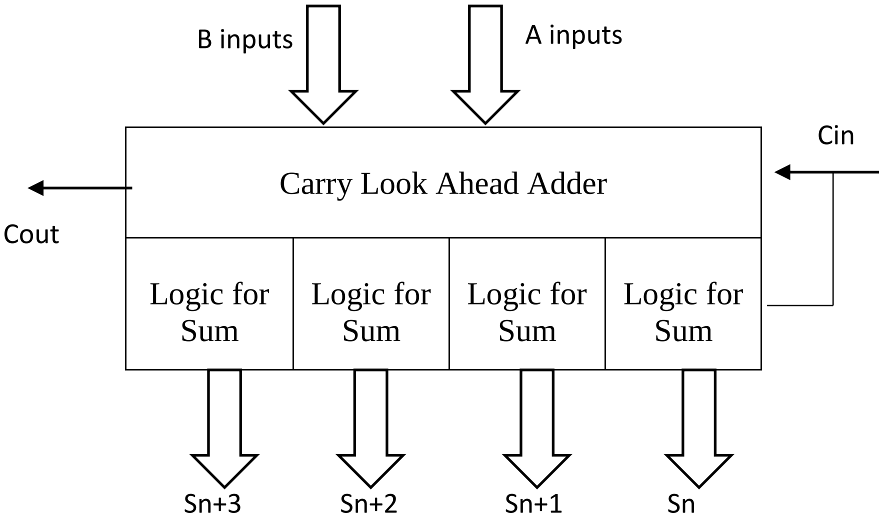
For 3rd bit it is given by

$$C_2 = G_2 +P_2 C_1 \quad \text{using } C_1 \text{ equation for } C_2$$
$$C_2 = G_2+ P_2G_1 +P_2P_1G_0 + P_2P_1P_0C_{in}$$

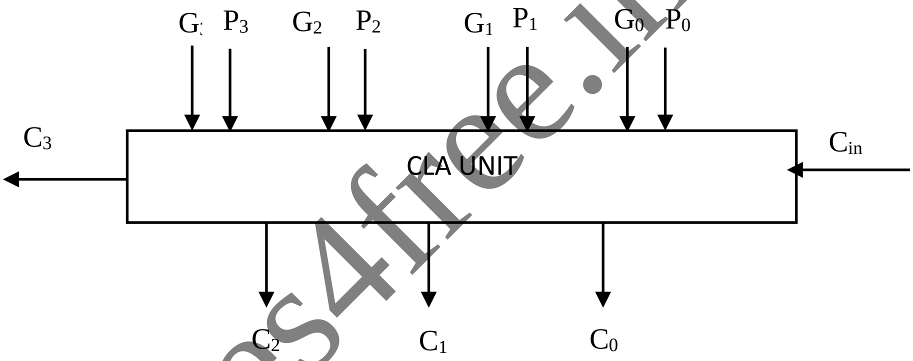
For the 4th bit it is given by

$$C_3 = G_3 +P_3 C_2 \quad \text{using } C_2 \text{ equation for } C_3$$
$$C_3 = G_3 +P_3G_2+ P_3P_2G_1 +P_3P_2P_1G_0 +P_3P_2P_1P_0C_{in}$$

- From the above equation we see that carry out C_3 can be calculated in prior without the need of need of carry being propagated through the adder cells.



- Each adder block has a 4 bit CLA(Carry Look Ahead) unit and is shown below.



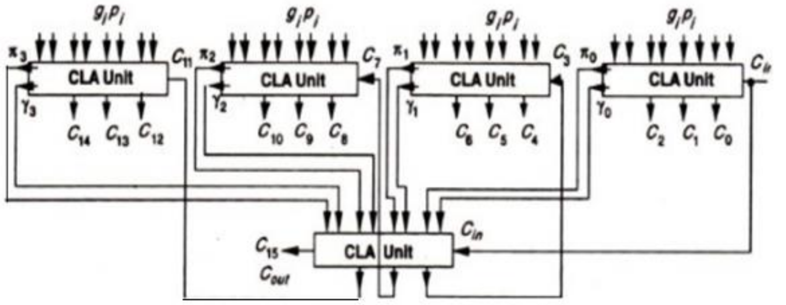
- From the CLA unit carry outputs C_0, C_1, C_2 and C_3 can be simultaneously determined. In this C_0, C_1, C_2 are required to determine the sum and will not ripple through. However C_3 will propagate to the next block.
- By observing the expression for C_0, C_1, C_2 and C_3 the last term in the equation become zero when $C_{in} = 0$.
- Thus to increase the speed, the expression for C_3 can be written as

$$C_3 = \gamma + \pi$$

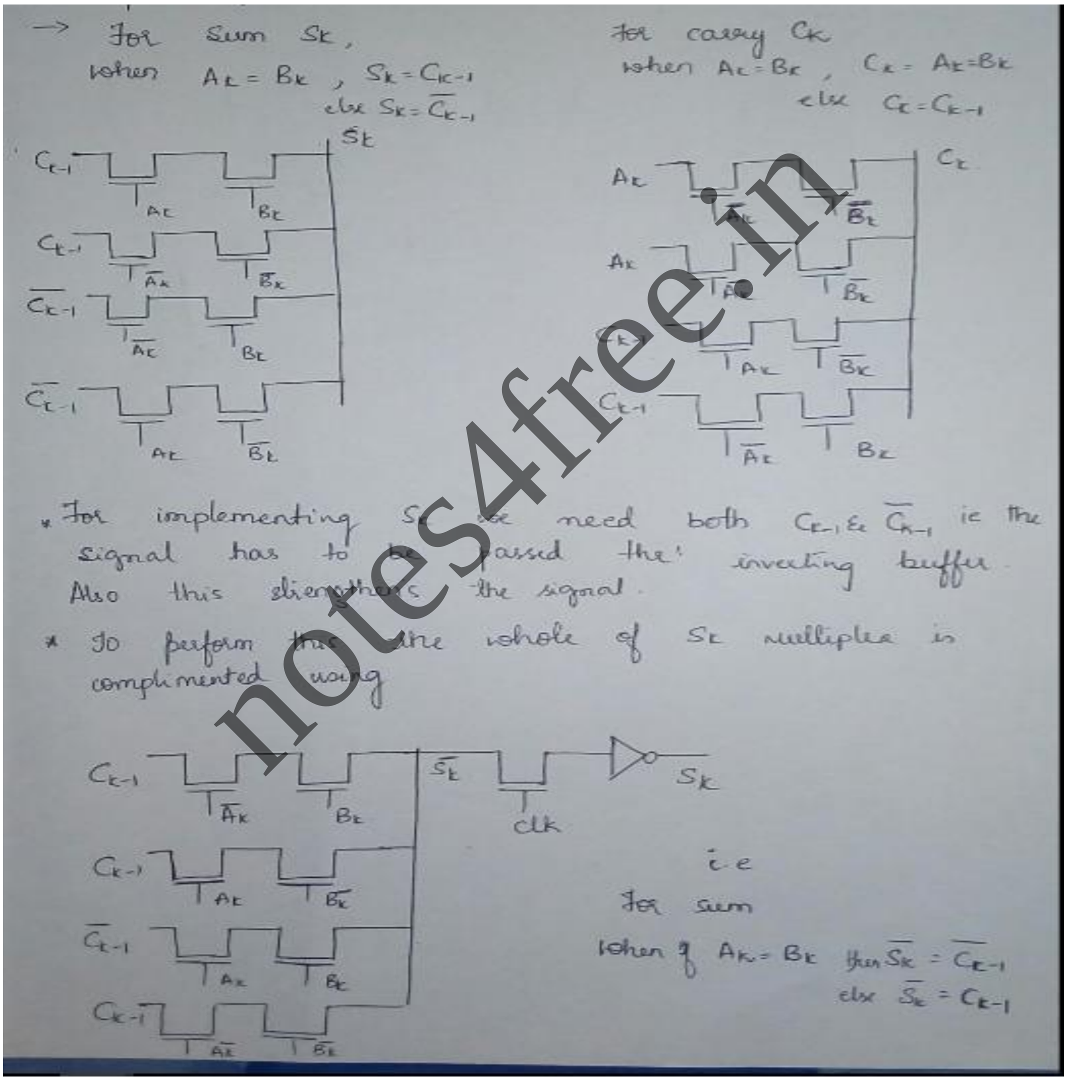
Where $\pi = P_3P_2P_1P_0C_{in}$ and

$$\gamma = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$$

- Here the term γ is independent of C_{in} and π becomes 0 when $C_{in} = 0$. Hence the speed of adder can be minimized by using these 2 defined functions γ and π.
- The implementation is shown in the block diagram. The carry outputs C3, C7 and C11 are propagated sequentially. This propagation delay can be reduced by using γ and π outputs, where C3, C7 and C11 are computed seperately.



16-bit, 4X4 block Carry look-ahead adder unit



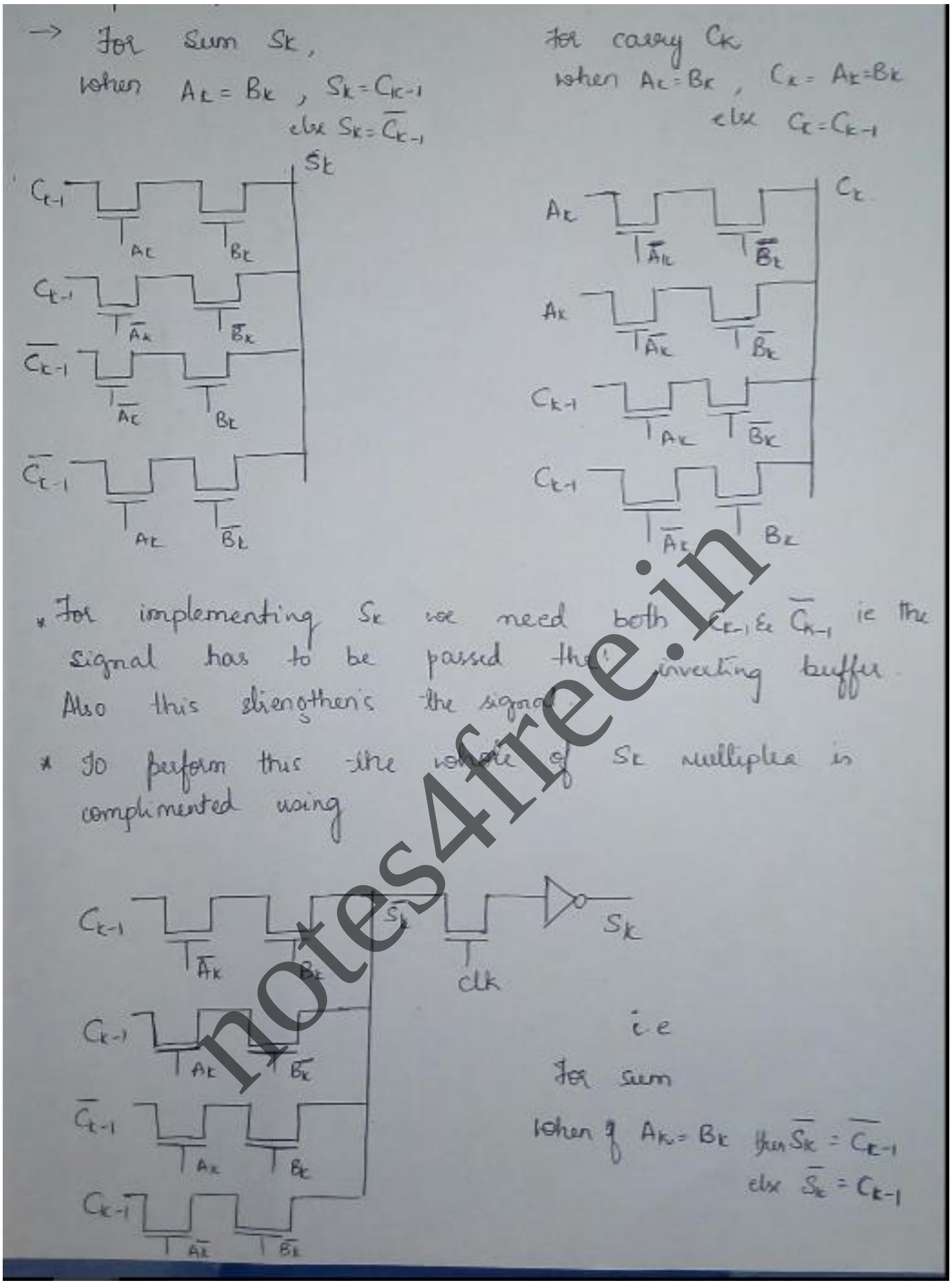
→ For sum S_k,
when $A_k = B_k$, $S_k = C_{k-1}$
else $S_k = \overline{C_{k-1}}$

For carry C_k
when $A_k = B_k$, $C_k = A_k = B_k$
else $C_k = C_{k-1}$

* For implementing S_k we need both C_{k-1} & $\overline{C_{k-1}}$ ie the signal has to be passed thr' inverting buffer. Also this strengthens the signal.

* To perform this the whole of S_k multiplex is complimented using

i.e
For sum
when if $A_k = B_k$ then $\overline{S_k} = \overline{C_{k-1}}$
else $\overline{S_k} = C_{k-1}$



→ For sum S_k,
when $A_k = B_k$, $S_k = C_{k-1}$
else $S_k = \overline{C_{k-1}}$

For carry C_k
when $A_k = B_k$, $C_k = A_k = B_k$
else $C_k = C_{k-1}$

* For implementing S_k we need both C_{k-1} & $\overline{C_{k-1}}$ ie the signal has to be passed thr' inverting buffer. Also this strengthen's the signal.

* To perform this the whole of S_k multiplex is complimented using

i.e
For sum
when if $A_k = B_k$ then $\overline{S_k} = \overline{C_{k-1}}$
else $\overline{S_k} = C_{k-1}$

Implementation of Adder using MUX.

Module 4

Subsystem Design

- This deals with designing of subsystems, which is a small part in a larger system (leaf cell).
- The most basic leaf cell of any digital system is the logic gates and these are seen in different technologies like nMOS, CMOS and BiCMOS.
- While designing high regularity should be maintained. This indicates detailed designing of few leaf cells which can be replicated and interconnected to form system.

Architectural Issues:

- As the complexity of a system increases, the design time also increases. Thus while designing we have to adopt those design methodologies which allows handling complexity with reasonable time and reasonable amount of labor.
- The following are the guidelines/architectural issues that needs to be considered while designing of the system.
  1. Define the requirements carefully and properly.
  2. Partition the overall architecture into appropriate subsystems.
  3. Communication paths should be carefully selected in order to develop sensible interrelationships between the subsystems.
  4. Draw the floor plan of how the system is to map onto the silicon.
  5. Aim for regular structures so that design is largely a matter of replication.
  6. Draw suitable stick or symbolic diagrams of the leaf-cells of the subsystem.
  7. Convert each cell into layout.
  8. Carry out design rule check carefully and thoroughly.
  9. Simulate the performance of each cell/subsystem.

Note: alternate between 2, 3 and 4 can be done as required.

The whole design process will be assisted if considerable care is taken with:

- Partitioning of the system so that there are clear subsystems with minimum interdependence and minimum complexity of interconnection between them.
- The design within the subsystems should be simple which helps in cellular design concept. This helps the systems in having few standard cells which are replicated to form high regular structures.

For designing digital systems in MOS technology there are two ways of building the circuits. They are

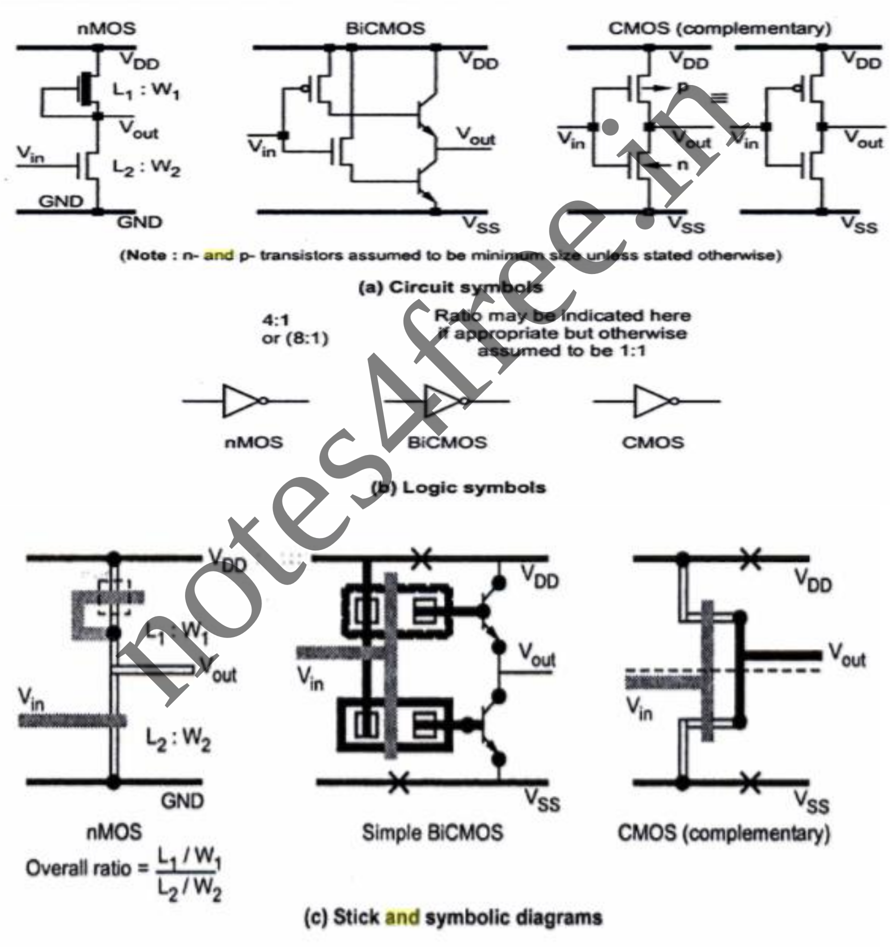
1. Switch logic
2. Gate (restoring) logic.

Gate (restoring) Logic:

- Gate logic is based on the general arrangement of inverter as it is the simplest gate. The inverter can be constructed with nMOS, CMOS or BiCMOS technology. Similar to this NAND and NOR can be constructed. Also AND and OR can be constructed with an inverter for NAND and NOR respectively.
- In nMOS technology the L:W ratio must be considered to get desired Z_{pu}/Z_{pd} ratio.

Inverter:

- The inverter circuit in different technology, corresponding stick diagram and symbolic diagram is shown in the Fig.



(Note : n- and p- transistors assumed to be minimum size unless stated otherwise)

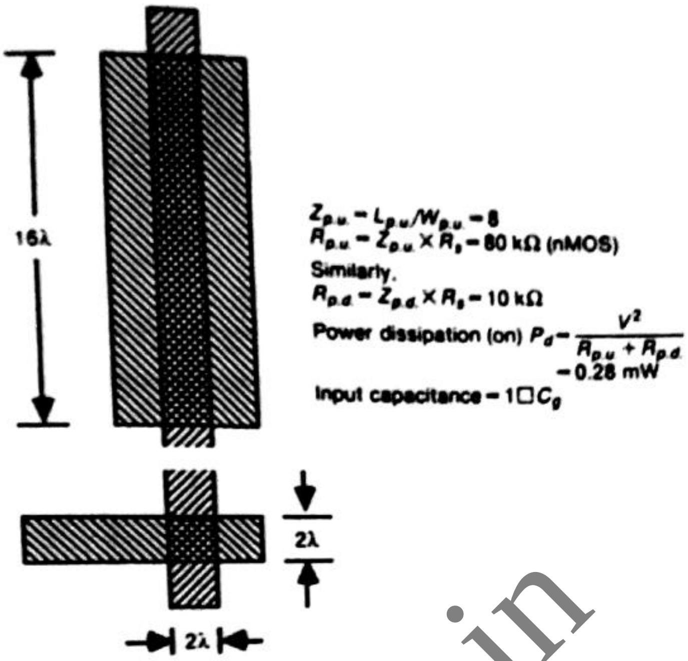
(a) Circuit symbols

(b) Logic symbols

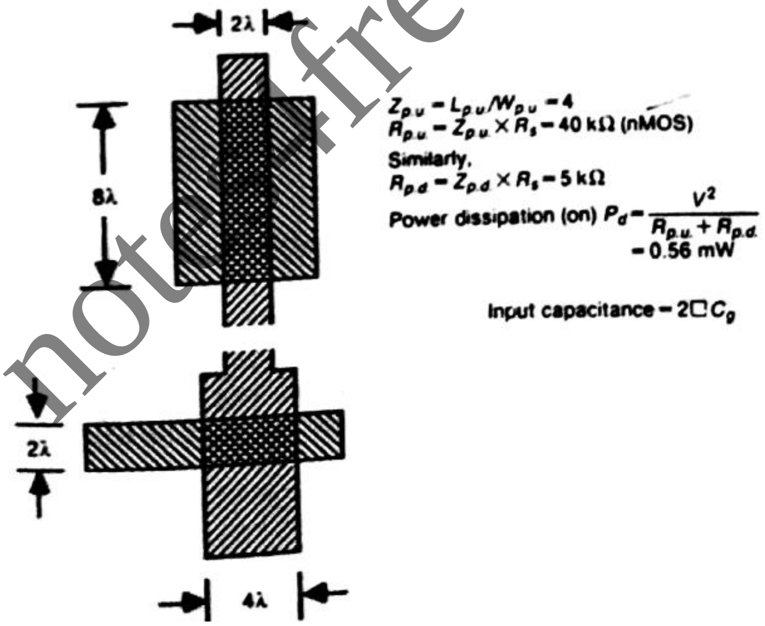
(c) Stick and symbolic diagrams

nMOS inverter

- In nMOS inverter the Z_{pu}/Z_{pd} ratio and the channel length to width ratio of each transistors shown.
- With different ratios of pull-up and pull-down several approaches are available.
- With different approaches it results in effecting the power dissipation P_d, area occupied, resistance and the capacitance values.
- The effect of power dissipation and capacitance for different ratios can be seen in the example for nMOS.



nMOS inverter with ratio 8:1, power dissipation is 0.28mW and input capacitance is 1□Cg.



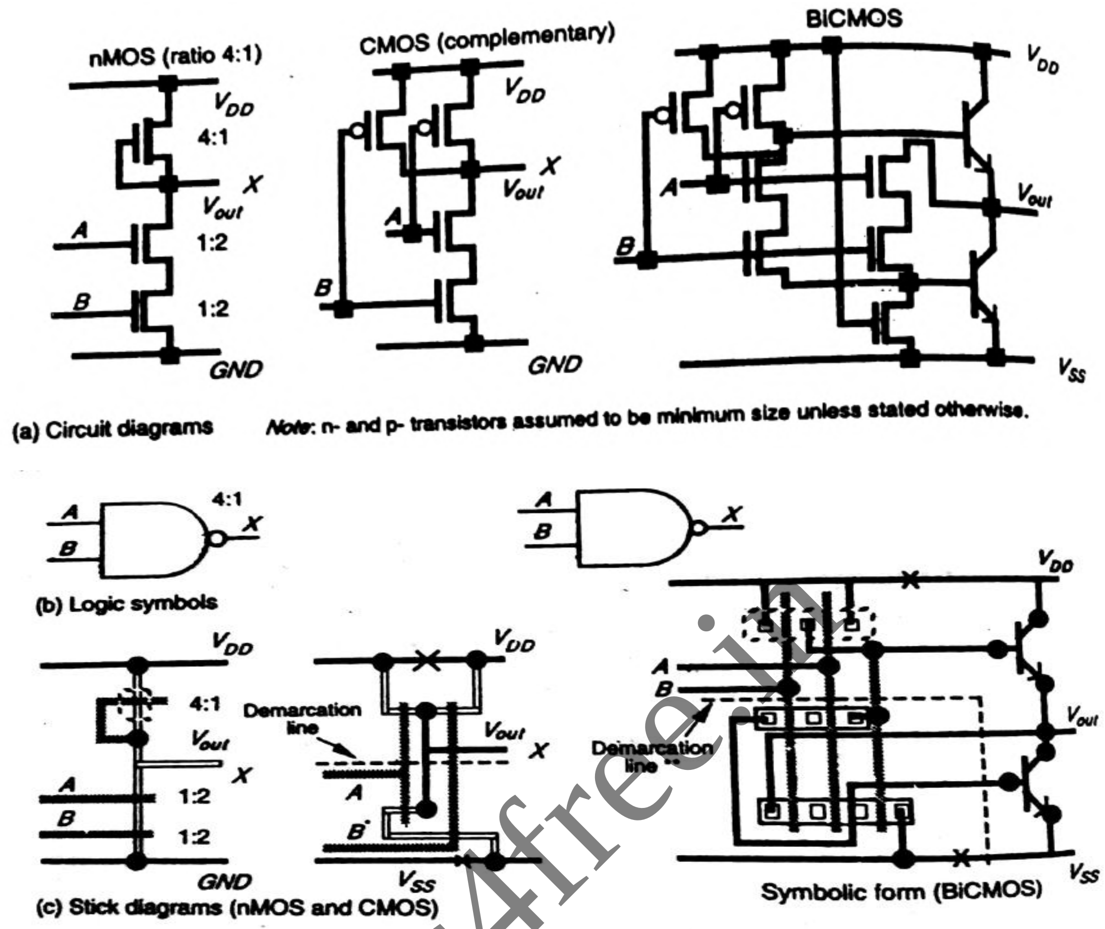
nMOS inverter with ratio 4:1, power dissipation is 0.56mW and input capacitance is 2□Cg.

CMOS inverter

- In CMOS there is static current and thus there is no power dissipation but only at switching there is power dissipation. The power dissipation for fast CMOS logic circuits are considered.

Two-input nMOS, CMOS and BiCMOS NAND Gates:

- Two input NAND gate arrangement is shown in different technologies along with their symbolic representation and stick diagram.



(a) Circuit diagrams *Note*: n- and p- transistors assumed to be minimum size unless stated otherwise.

(b) Logic symbols

(c) Stick diagrams (nMOS and CMOS)

Note: The natural 2.5:1 asymmetry of the CMOS inverter is improved to 1.25:1 (or better) owing to the two n-type pull-down transistors in series for the two I/P *Nand*.

** Demarcation line (edge of n-well) may be shown if required.

Fig. Two input NAND gate circuit, symbol and stick diagram

nMOS NAND gate

- For nMOS L:W ratio must be carefully chosen so that desired overall Z_{pu}/Z_{pd}
- Here Z_{pd} is contributed by the input transistors connected in series and Z_{pu} is contributed with enhancement mode transistor.
- It is obtained that ratio between Z_{pu} and sum of all pull-down Z_{pd} must be 4:1
- nMOS NAND gate geometry reveals two significant factors.
    - nMOS NAND gate area is greater than that of nMOS inverter. In the pull-down network as the transistors are added in series to provide number of inputs. As the number of inputs are added, corresponding adjustment has to be made in the length of the pull-up transistor in order to maintain the required ratio.
    - The delay in nMOS NAND gate increases in a direct proportion to the number of inputs added. If there are n inputs then the length and resistance of the pull-up transistor must be increased by a factor on n in order to maintain the correct ratio. The delay observed in NAND gate is given by

$$\tau_{Nand} = n\tau_{inv}$$

Where n is the number of inputs and τ_{inv} is the delay related to nMOS inverter.

- ✓ With these properties the nMOS NAND gate is used only when it absolutely necessary and the number of inputs are restricted.

CMOS NAND gate

- No restriction of ratio is seen, but asymmetry is the problem seen in CMOS.
- In order to keep the transfer characteristics symmetrical at $V_{DD}/2$ it is necessary to perform some adjustments in the transistor geometry.

BiCMOS NAND gate

- BICMOS nand gate is complex and difficult for fabrication
- It has considerable load-driving capabilities and useful when driving large capacitive loads or when there is a large fan-out.

Two-input nMOS, CMOS and BiCMOS NOR Gates:

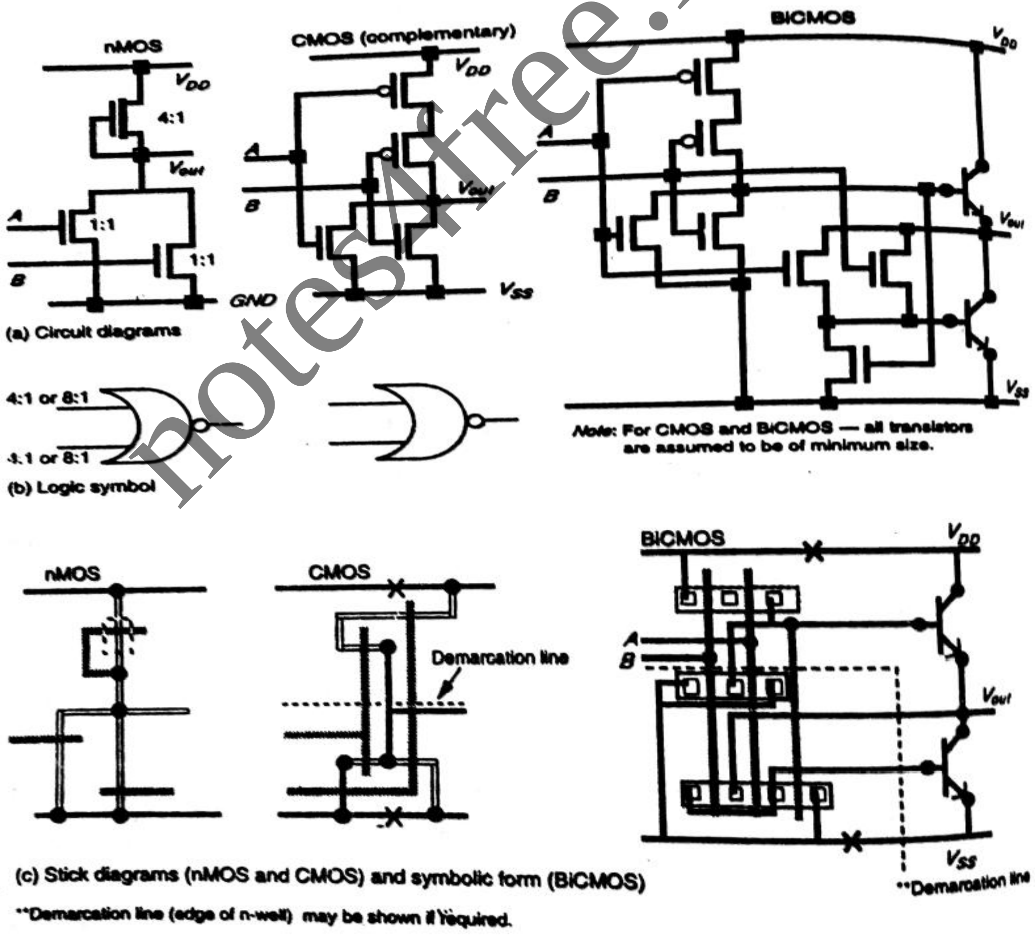


Fig. Two input NOR gate circuit, symbol and stick diagram

- Two input NOR gate arrangement in different technologies along with stick diagram is shown in the above figure.

nMOS NOR gate

- In 2 input NOR gate as one end of the gate provide a path to ground (when it is turned on) from the pull-up network. Thus any one conducting pull-down gate will give characteristics to that of inverter. Hence the ratio of the inverter can be maintained. This is applicable to any number of inputs.
- The area consumed by NOR gate reasonable compared to NAND because with the increase in the number of inputs it is not affecting the dimension of the pull-up device(as seen for NAND gate)
- NOR gate is fast as inverter and is preferable when choice is possible.
- The ratio of Z_{pu} and Z_{pd} depend on the source which is driving the NOR gate, 4:1 ratio if driven by an inverter and 8:1 if driven by one or more pass transistor.

CMOS NOR gate

- CMOS NOR has p-based-transistor network in the pull-up to implement logic 1. Similarly in the pull-down network has n-based transistor to implement logic 0.
- In the pull-up network the p-structure consists of p-transistors connected in series for each input. This results in increasing the resistance.
- Also asymmetry in rise-time and fall-time on capacitive loads is increased and this results in shifting the transfer characteristics which will reduce the noise immunity.
- Thus CMOS NOR gates with more than 2 inputs require adjustment of p or n transistor geometries.

BiCMOS NOR gate

- BICMOS NOR gate is complex and difficult for fabrication
- It has considerable load-driving capabilities and useful when driving large capacitive loads or when there is a large fan-out.

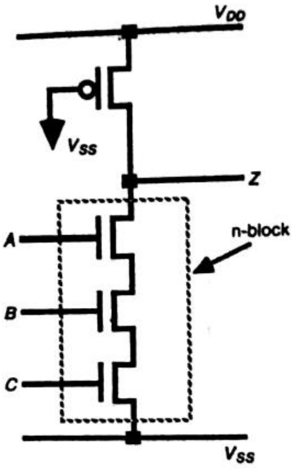
Other Forms of CMOS Logic:

With the availability of both n and p transistors it is possible for the designer to exploit alternatives to inverter based CMOS logic.

Pseudo-nMOS logic

- In nMOS circuit if the pull-up network with depletion mode transistor is replaced by pMOS transistor with its gate connected to V_{ss}. This structure is called as pseudo-nMOS logic.
- This logic helps in reducing the number of transistors seen in the CMOS logic seen for pull-up network.
- Fig. shows 3 input NAND logic using pseudo-nMOS logic.
- As the pMOS is connected to V_{SS} it can be either in saturation or active region and the status of nMOS at pull-doun network depend on the state of input.
- When all the inputs are zero, pMOS will be on and the output is pulled up to V_{DD} and with other data pMOS acts as a current source (saturation). The output is now

the product of the resistance of the pull-down network and current of the pull-up network.



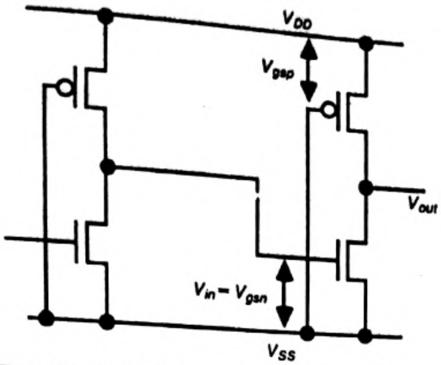
Pseudo-nMOS *Nand* gate.

- In order to obtain required ratio, the arrangement considered is pseudo-nMOS inverter being driven by another pseudo-inverter.
- For analysis taking the condition $V_{in} = V_{DD}/2$, in this condition nMOS device is operating in saturation ($0 < V_{gsn} – V_{tn} < V_{dsn}$) and pMOS is operating in linear /resistive region ($0 < V_{dsp} < V_{gsp} – V_{tp}$).
- Equating the currents of nMOS and pMOS and suitable arrangement, we obtain the equation as

$$V_{inv} = V_{tn} + \frac{(2\mu_p/\mu_n)^{1/2}[(-V_{DD} - V_{tn})V_{dsp} - V_{dsp}^2]^{1/2}}{(Z_{p.u.}/Z_{p.d.})^{1/2}}$$

$$Z_{p.u.} = L_p/W_p$$

$$Z_{p.d.} = L_n/W_n$$



Pseudo-nMOS inverter when driven from a similar inverter.

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

$$V_{tn} = |V_{tp}| = 0.2V_{DD}$$

$$V_{DD} = 5\text{ V}$$

$$\mu_n = 2.5\ \mu_p$$

We obtain

$$\frac{Z_{p.u.}}{Z_{p.d.}} = \frac{3}{1}$$

- Thus it can concluded as:
  1. As the sheet resistance of the channel of pull-up network (PUN) is about 2.5 times that of pull-down network (PDN) also the ratio of PUN and PDN is 3:1. With this the pseudo-nMOS exhibits resistance of about 85Kohm compared to resistance of 50Kohm with nMOS device. This helps in reducing the power dissipation up-to 60% in comparison with device with nMOS device.
  2. With higher pull-up resistance the delay is larger by 8.5:5 than 4:1 nMOS inverter.

Dynamic CMOS logic:

- The logic is implemented as shown in the circuit.
- It has n block based on the logic to be implemented. Along with this there are 2 more transistors called 'precharge' transistor (pMOS) and 'evaluation' transistor (nMOS).
- Both these transistors are connected to common clock signal 'ϕ'

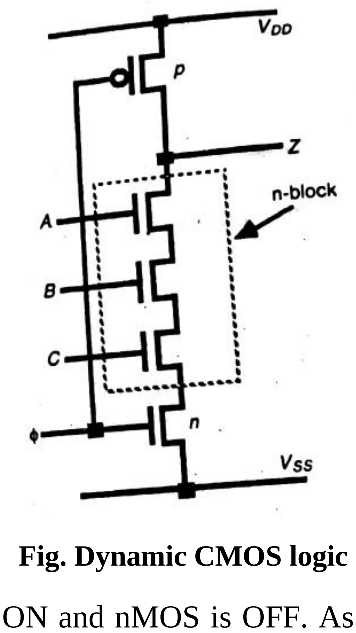


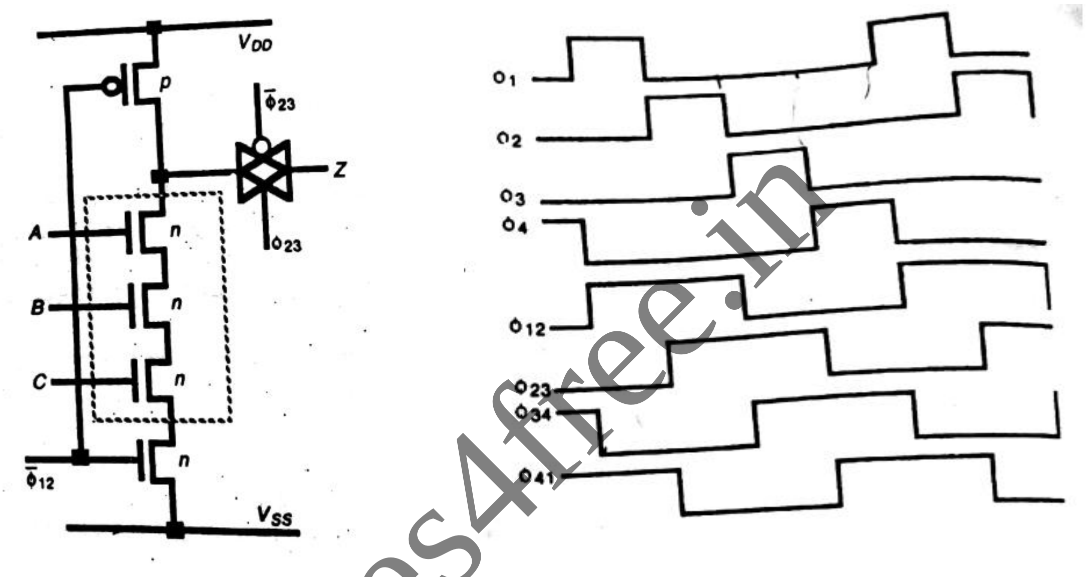
Fig. Dynamic CMOS logic

- During ϕ = 0, pMOS is ON and nMOS is OFF. As the PDN is not in the ON state, due to the PUN output is pulled to V_{DD} i.e., output is precharged to V_{DD}.

- During ϕ = 1, pMOS is OFF and nMOS block will be conducting as the evaluation transistor is ON. Thus load capacitor discharges depending on the inputs status of nblock.

Problems seen in dynamic logic

- ✓ Charge sharing problem occurs if the inputs changes during the on period of the clock.
- ✓ Single phase dynamic logic cannot be cascaded because of delay. This delay may result in wrong output results.



(a) (b)

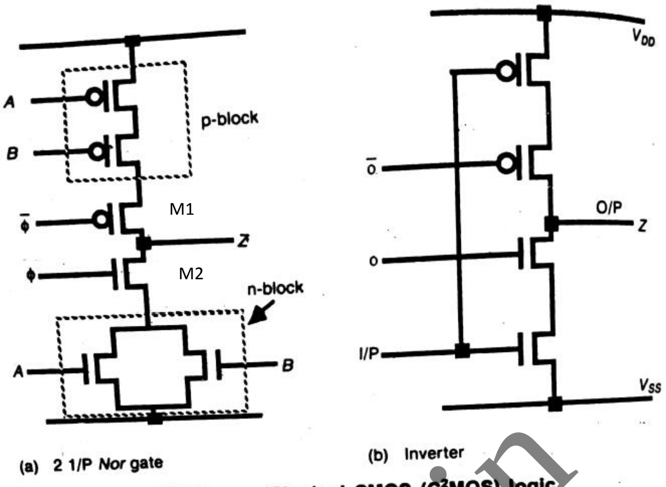
Fig. (a) Possible clock arrangement with TG. (b) Possible 4ϕ and derived clocks

- To overcome the problem in dynamic logic derived clocks must be used.
- It uses 4 phase clock ϕ_1, ϕ_2, ϕ_3 and ϕ_3 to derived clocks ϕ_{12}, ϕ_{23}, ϕ_{34} and ϕ_{41} which are obtained by overlapping the clock signals.
- The circuit is modified with the inclusion of a transmission gate.
- The transmission gate samples the output during evaluate phase and holds the output state until the next stage evaluates the logic.
- For each stage different phase signals are provided for precharge and evaluate and also for the transmission gate.

Clocked CMOS (C^{2}MOS) logic:

- In this the given logic is implemented in both nMOS and pMOS transistors in the pull-up P block and pull-down N block respectively. This is seen in the fig.
- Along with this 2 additional transistors are included M1 and M2.
- M1 and M2 transistors are provided with clock and its compliment form. Thus both will be either ON or OFF simultaneously.
- When ϕ = 0, both transistors are OFF and from either pMOS or nMOS network is not connected to the output. Hence output remains in previous state.
- When ϕ = 1, both transistors are ON and the logic of the input will be evaluated.

- With additional transistors the area increases. Also with additional transistors causes the output rise time and fall time to increase and this results in more delay.



(a) 2 1/P Nor gate (b) Inverter

FIGURE Clocked CMOS (C²MOS) logic.

CMOS Domino logic:

- The problem seen in dynamic logic of cascading can be overcome by using an inverter between the stages as shown in the figure below.
- It is an extension of dynamic CMOS logic.

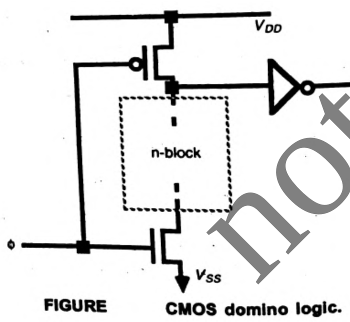
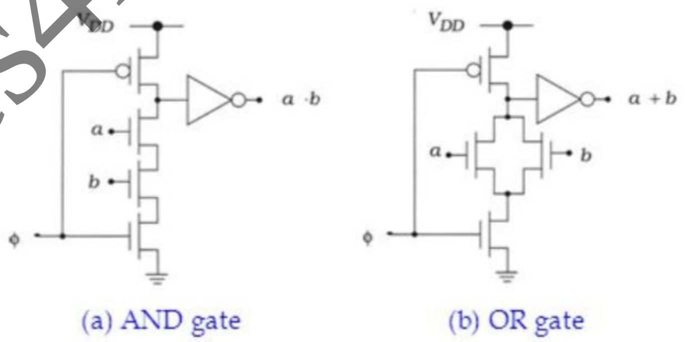


FIGURE CMOS domino logic.



(a) AND gate (b) OR gate

Fig. Domino logic AND an OR gate

- When clk or $\phi = 0$ (pre-charge phase), output node of dynamic CMOS logic is pre-charged to V_{DD} and this causes output of inverter low. This low output given to n-block (pull down network – PDN) of next stage causes the nMOS to be in the off state.
- When clk or $\phi = 1$ (evaluation phase), the output node either discharges to 0 or remains in the same state depending on the state of the inputs.
- Depending on this output the next output also changes.
- As the output of one stage depends on the output of previous stage, this is similar to the domino effect. Thus this configuration is called as 'domino logic'
- Cascaded stages of domino logic circuit is shown in the Fig.

Note: The inverter output change at most can make transition only one transition i.e., from 0 to 1, but no 1 to 0 transition can take place in the evaluation phase.

Advantages of Domino logic:

- Structures have smaller area than the conventional CMOS logic.
- Parasitic capacitances are smaller so higher operating speeds can be obtained.
- Operation is free from glitches as only 1 to 0 transition is made.
- Only non-inverting structures are possible because of the presence of inverter buffer.

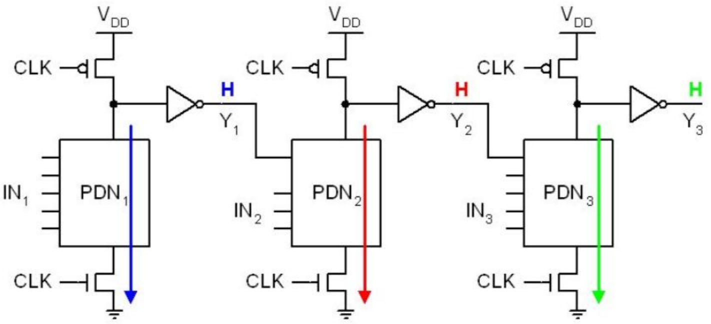


Fig. Cascaded Domino Logic

n-p CMOS logic

- Another variation of dynamic logic is the n-p CMOS logic as shown below.
- This logic is modification of domino logic where the inverter can be avoided and using p-block and n-block alternatively.

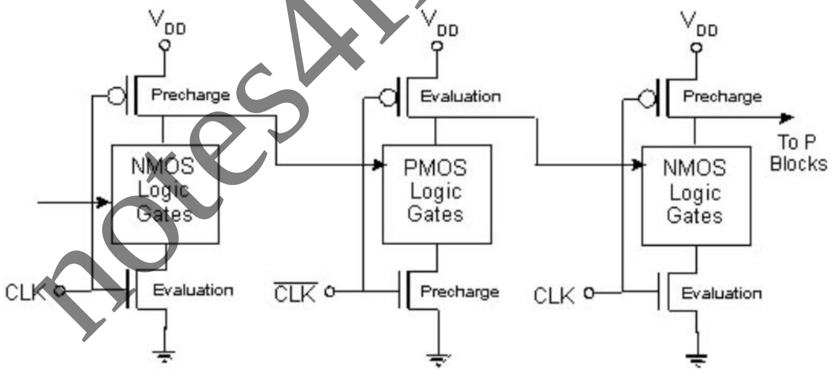


Fig. n-p CMOS logic

- When clk = 0, pre-charge phase, output of I block is HI and is given to p-block logic of II stage. This HI to pMOS will turn it off.
- The clk' (which is 1) is given to II stage and to the nMOS transistor. This causes the nMOS to turn on and output of II stage is goes low.
- This low given to next n-block will turn off the nMOS devices and this continues.
- Thus during pre-charge phase all the n-block and p-block logics will be off.
- When clk = 1, evaluation phase depending on the inputs at the p-block and n-block logic is evaluated and output each stage will remain HI of discharges

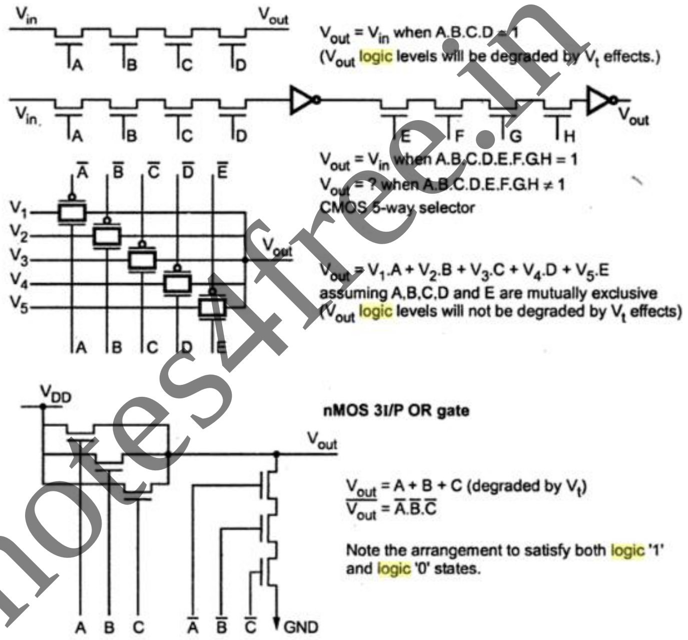
NOTE:

- Pseudo nMOS logic is preferred where high output is needed at most of the time. In this condition the static power dissipation is less and also propagation delay becomes shorter. Common application of pseudo nMOS is in designing of address decoders for memory chips in ROMs.

- The advantage of dynamic CMOS logic over static CMOS or pseudo nMOS logic is that it reduce significantly the surface required to implement logic using complementary pMOS transistors. Also the switching speed is increased.
- Clocked CMOS logic is used for low power dissipation. This logic structure is used to incorporate latches or interface with other form of logic.

Switch Logic

- The switch logic is built on 'pass transistors' or on 'transmission gates'. Switch logic is fast for small arrays and draws no static current from the supply rails. Hence power dissipation of such circuits is small because current flows only on switching.
- Pass transistor can be used as a switch in passing the signals. Switch logic arrangements using basic OR and AND connections along with other arrangements is shown in the fig.



V_in
V_out
A B C D
V_out = V_in when A.B.C.D = 1
(V_out logic levels will be degraded by V_t effects.)
V_in
A B C D
E F G H
V_out
V_out = V_in when A.B.C.D.E.F.G.H = 1
V_out = ? when A.B.C.D.E.F.G.H ≠ 1
Ā B̄ C̄ D̄ Ē
V1 V2 V3 V4 V5
V_out
A B C D E
CMOS 5-way selector
V_out = V1.A + V2.B + V3.C + V4.D + V5.E
assuming A,B,C,D and E are mutually exclusive
(V_out logic levels will not be degraded by V_t effects)
V_DD
V_out
nMOS 3I/P OR gate
V_out = A + B + C (degraded by V_t)
V̄_out = Ā.B̄.C̄
Note the arrangement to satisfy both logic '1' and logic '0' states.
A B C Ā B̄ C̄ GND

Fig. Some switch logic implementation

- Switch logic can be realized either using n or p pass transistors or from transmission gates.
- The transmission gate are complimentary switches made up of p-pass and n-pass transistor in parallel.
- Simple pass transistors suffer from undesirable threshold voltage effects which gives rise to loss of logic levels as shown in the Fig.

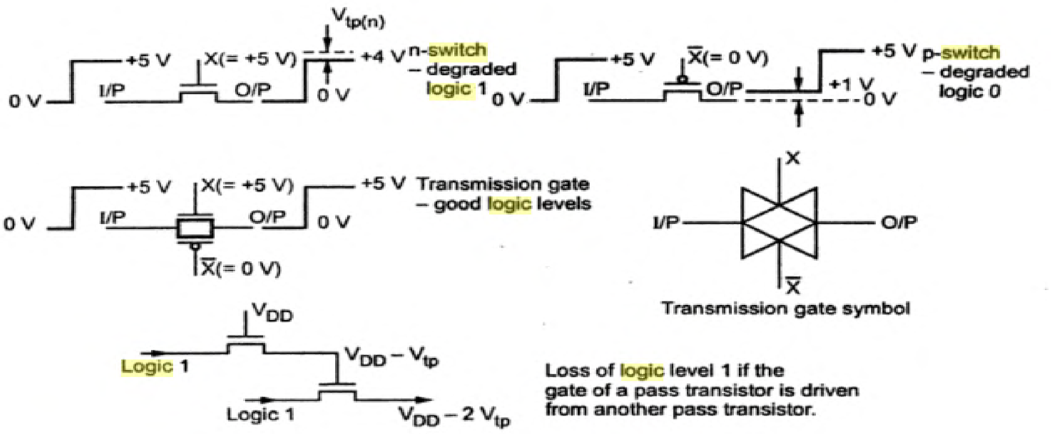

Vtp(n)
0 V
+5 V
I/P
X(= +5 V)
O/P
+4 V
0 V
n-switch – degraded logic 1
0 V
+5 V
I/P
X̄(= 0 V)
O/P
+5 V
+1 V
0 V
p-switch – degraded logic 0
0 V
+5 V
I/P
X(= +5 V)
O/P
+5 V
0 V
X̄(= 0 V)
Transmission gate – good logic levels
X
I/P
O/P
X̄
Transmission gate symbol
VDD
Logic 1
VDD – Vtp
Logic 1
VDD – 2 Vtp
Loss of logic level 1 if the gate of a pass transistor is driven from another pass transistor.

Fig I. Some properties of Pass Transistors and Transmission Gates

- Hence apparently complex transmission gate (TG) is preferred over simple n-switch or p-switch in CMOs applications.
  - The TG is free from any such degradation of logic levels
  - But it occupies more area
  - Requires complementary signals to drive it
  - The 'on' resistance of TG is lower than that of simple pass transistor switches
- Rules or restriction observed when using nMOS switch logic is that no pass transistor gate input must be driven through one or more pass transistors as this will degrade the output. This restriction is also shown in the Fig I.
- The logic levels propagated through pass transistor get degraded by threshold voltage effects. The signal out of the pass transistor T1 is not full logic 1 but rather a voltage that is one transistor threshold below the true logic 1(i.e., V_{DD}-V_{th}). Thus this degraded voltage does not permit the output of T2 to reach an acceptable logic 1 level.

Examples of Structured Design (Combinational Logic)

Multiplexer/data selectors

- A multiplexer or mux is a combinational circuits that selects several analog or digital input signals and forwards the selected input into a single output line.
- A multiplexer of 2^n inputs has n selected lines, are used to select which input line to send to the output.

Implementing 2:1 MUX

- MUX can be designed using various logic.
- The pass-transistor logic attempts to reduce the number of transistors to implement a logic by allowing the primary inputs to drive gate terminals as well as source-drain terminals.
- The implementation of a 2:1 MUX requires 4 transistors (including the inverter required to invert S), while a complementary CMOS implementation would require 6 transistors. The reduced number of devices has the additional advantage of lower capacitance.

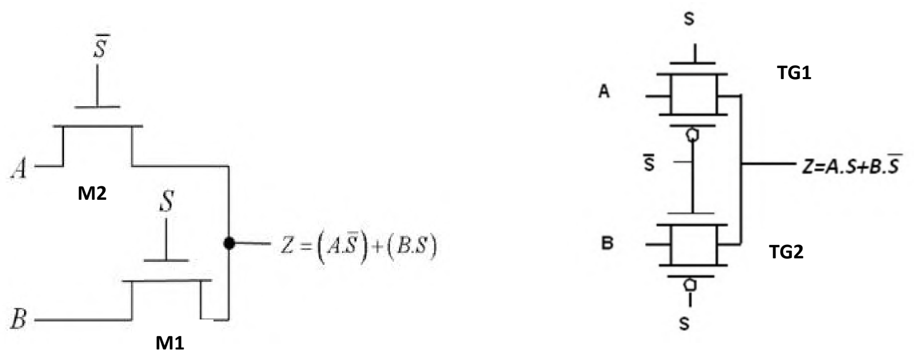


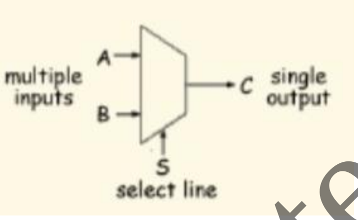
Fig. Pass transistor and transmission gate implementation of 2:1 MUX or data selector

For pass transistor logic

- When S = 0, transistor M1 is OFF and M2 is ON, thus output Z = AS’
- When S = 1, transistor M1 is ON and M1 is OFF, thus output Z = BS

For Transmission gate

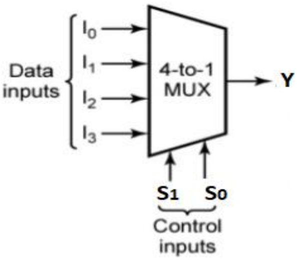
- The transmission gate acts as a bidirectional switch controlled by the gate signal
- When S = 0, TG1 = OFF and TG2 is ON and Z = B.
- When S = 1, TG1 = ON and TG2 is OFF and Z = A.
- Symbolid representation of MUX stick diagram and layout is below



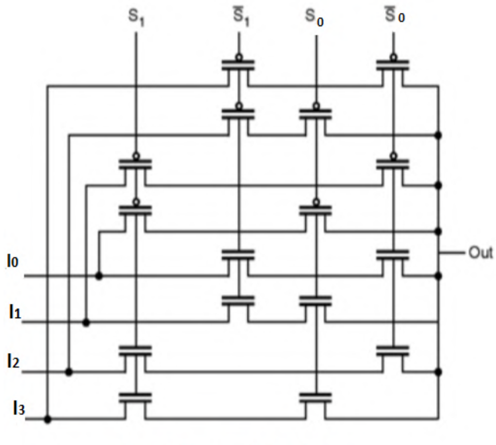
4:1 Multiplexer implementation

- There are 4 inputs - $I_{0,}$ $I_{1,}$ $I_{2,}$ I_3, 2 select lines $S_{0,}$ S_1 and 1 output line – Y/out.
- The truth table is shown

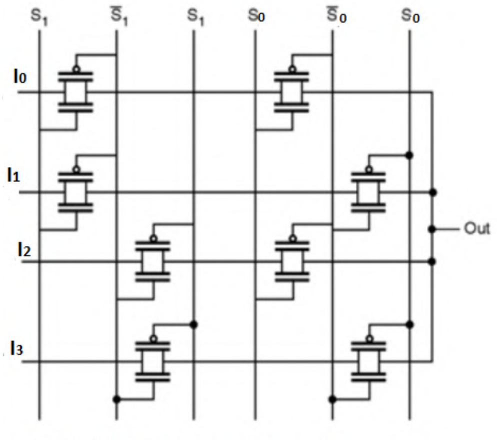
S1	S0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



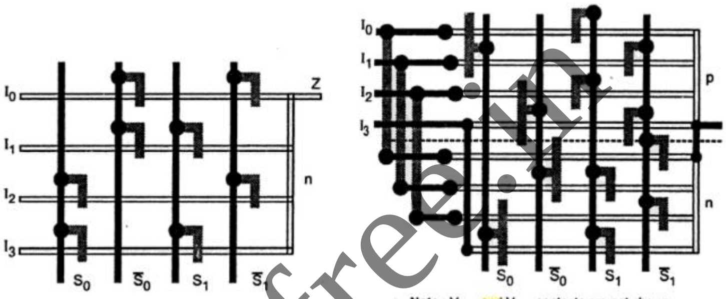
- The output equation for Y = I_0 $S'_1 S'_0$ + I_1 S'_1 S_0 + I_2 S_1 S'_0 + I_3 S_1 S_0
- 4:1 mux implemenation using CMOS technology and transmission Gate also their stick diagrams.



4 : 1 MUX using CMOS logic



4 : 1 MUX using transmission gate



Note : V_{DD} and V_{SS} contacts are not shown.

(a) nMOS switches

(b) Transmission gates (CMOS)

- 4:1 mux can also be implemented using two 2:1 multiplexer

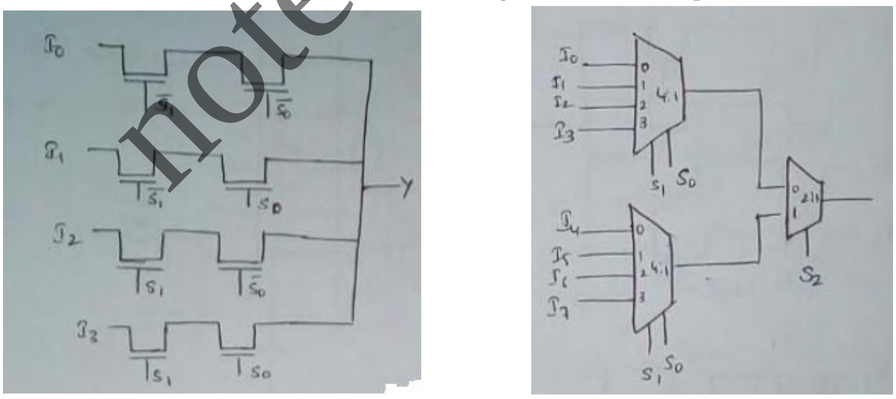


Fig. 4:1 MUX using pass transistor logic and implementation of 8:1 using two 4:1 MUX

- 8:1 mux can be implemented using two 4:1 multiplexer

S2	S1	S0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4

1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

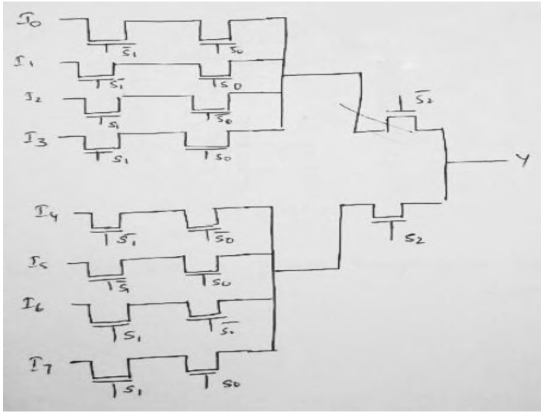
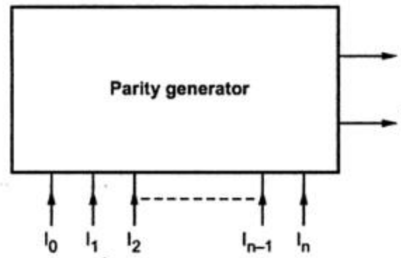


Fig. Truth table for 8:1 MUX and implementation of same using pass transisitor logic

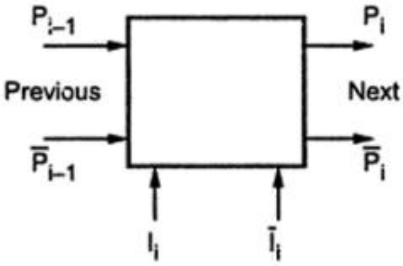
Parity Generator

- The parity generating technique is one of the most widely used error detection techniques for the data transmission.
- In digital systems, when binary data is transmitted and processed, data may be subjected to noise.
- Hence, **parity bit** is added to the word containing data in order to make number of 1s either even or odd. During the transmission of binary data, the message containing the data bits along with parity bit is transmitted from transmitter node to receiver node.
- At the receiving end, the number of 1s in the message is counted and if it doesn't match with the transmitted one, then it means there is an error in the data.
- A parity generator is a combinational logic circuit that generates the parity bit in the transmitter. On the other hand, a circuit that checks the parity in the receiver is called parity checker. A combined circuit or devices of parity generators and parity checkers are commonly used in digital systems to detect the single bit errors in the transmitted data word.
- In even parity, the added parity bit will make the total number of 1s an even amount whereas in odd parity the added parity bit will make the total number of 1s odd amount.
- The basic principle involved in the implementation of parity circuits is that sum of odd number of 1s is always 1 and sum of even number of 1s is always zero. Such error detecting and correction can be implemented by using Ex-OR gates (since Ex-OR gate produce zero output when there are even number of inputs).
- Parity Generator: It is combinational circuit that accepts an n-1 bit stream data and generates the additional bit that is to be transmitted with the bit stream. This additional or extra bit is termed as a parity bit.
- In **even parity** bit scheme, the parity bit is '**0**' if there are **even number of 1s** in the data stream and the parity bit is '**1**' if there are **odd number of 1s** in the data stream.
- In **odd parity** bit scheme, the parity bit is '**1**' if there are **even number of 1**s in the data stream and the parity bit is '**0**' if there are **odd number of 1s** in the data stream. Let us discuss both even and odd parity generators.
- In VLSI a circuit to be designed to indicate parity of a binary number is shown in the Fig. for an (n+1) bit input.



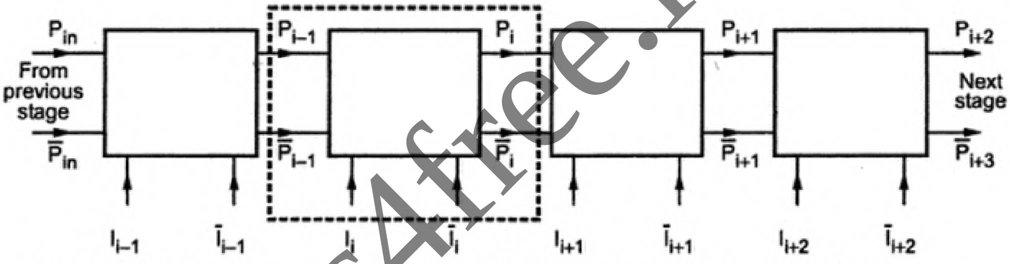
Note : P = {1 Even number of 1s at input, 0 Odd number of 1s at input

Parity generator basic block diagram



Parity generator-basic one-bit cell

- Since the no. of bits is undefined, a general solution on cascadable bit-wise and a regular structure is shown in Fig.
- A standard or basic one-bit cell from which an n-bit parity generator may be formed. The standard/ basic cell is shown in the Fig.
- The parity information is passed from one cell to next and the parity information is modified or retained depending on the input lines A_i and A'_i



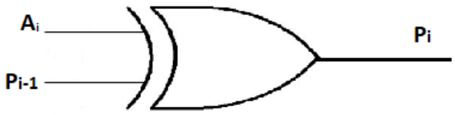
Note : Parity requirements are set at the left most cell where P_{in} = 1 sets even and P_{in} = 0 sets odd parity.

Fig. Parity generator - structured design approach

- If $A_i = 1$, parity output P_i will change to P'_{i-1} (i.e., if A_i [input] =1 and P_{i-1} [previous parity] =1, the output parity P_i will change to 0)
- If $A_i = 0$, output parity P_i will remain in the same state of P_{i-1} (i.e., if A_i [input] =0 and P_{i-1} [previous parity] =1, the output parity P_i will remain as 1)
- Suitable arrangement for such cell is implemented using the expression

$$P_i = P'_{i-1} A_i + P_{i-1} A'_i$$

- The same can be implemented using nMOS and CMOS technology. The parity generator symbol and stick diagram for nMOS technology is shown below.



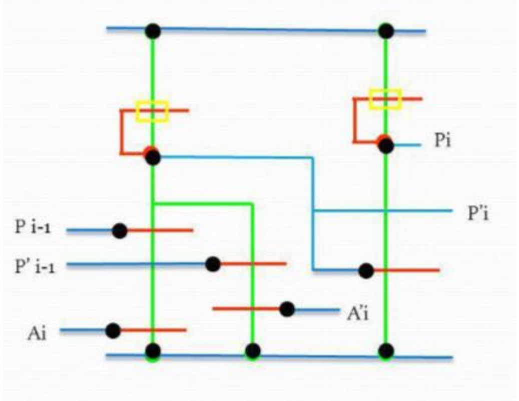
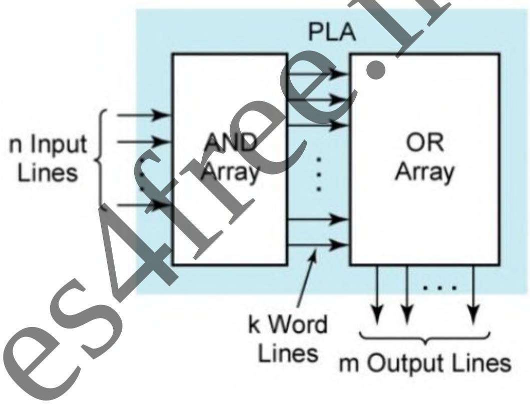


Fig. Symbol for P_i **Fig. nMOS stick diagram for parity generator**

The Programmable Logic Array (PLA)

- PLA describes class of standalone devices that allows users to program the functionalities. The capability of these programmable devices are limited and have been replaced by significantly powerful field programmable gate array (FPGA).
- Fig shows the block diagram of PLA. It consists of two level of combinational logic functions.
- Both the levels of combinational logic used together helps in implementing sum-of-products (SOP) logic functions.
- It performs the same basic function as a ROM. It has n inputs and m outputs and can realize m functions of n variables
- 'AND' plane is responsible for the generation of all product terms needed to form logic functions. 'OR' plane OR's (sums) the selected product terms together to form the desired logic functions.



General architecture of PLA

- Fig. shows 3 input and 3 output PLA. Along with 3 inputs its compliment is also available. With this it is possible to generate many functions of product terms in AND array.
- Inputs are A, B & C and outputs are O_1, O_2 & O_3

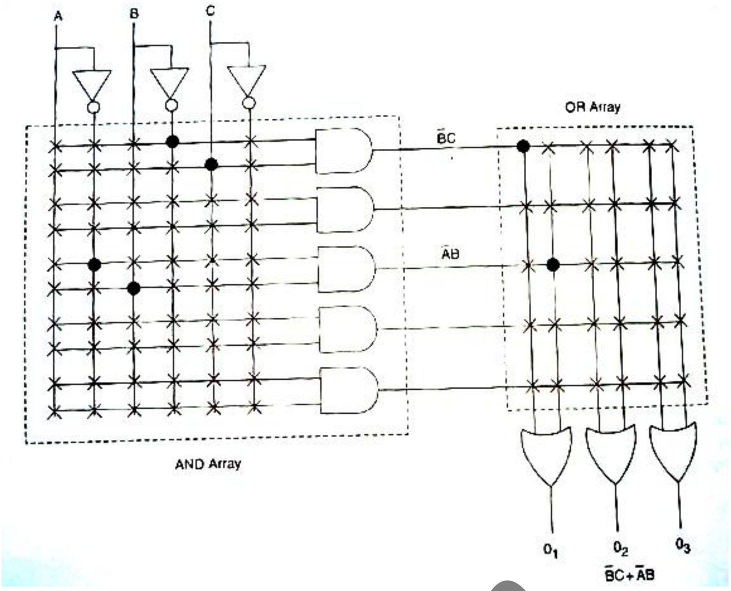
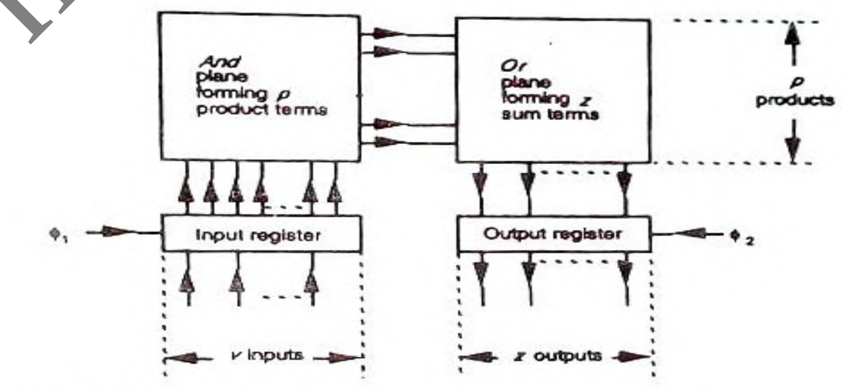


Fig. Architecture of PLA

- X indicates no connection and desired product term can be obtained by making relevant connection in the AND array (shown with dot)
- Similarly connection can be made in OR array.
- For obtaining $O_1 = B'C + A'B$, the connections are shown in the Fig.
- In VLSI design objective is to map circuits onto Si to meet the specifications.
- In circuit implementation for AND and OR array need NAND and NOT logic and NOR and NOT logic respectively. But this includes more fabrication steps.
- However this can be simplified by implementing the logic in NOR logic. Thus AND and OR array can be implemented using NOR logic.
- If the output of NOR is complimented then we get the OR logic. Similarly if both the inputs of AND is complimented and given to NOR it gives AND logic.
- Both the implementation is shown below.



(a) *And/Or* based

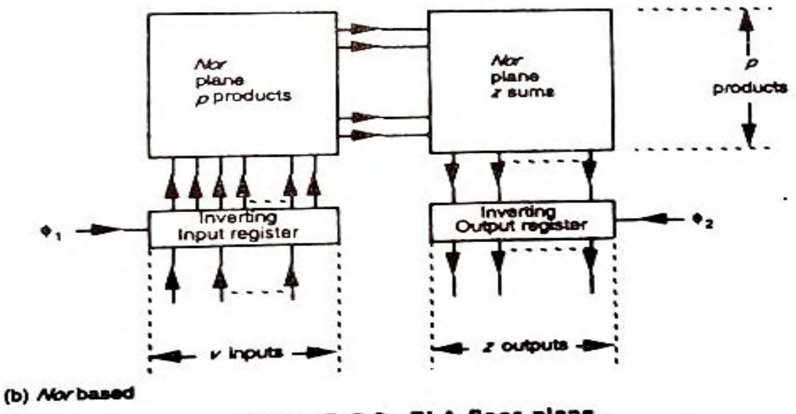


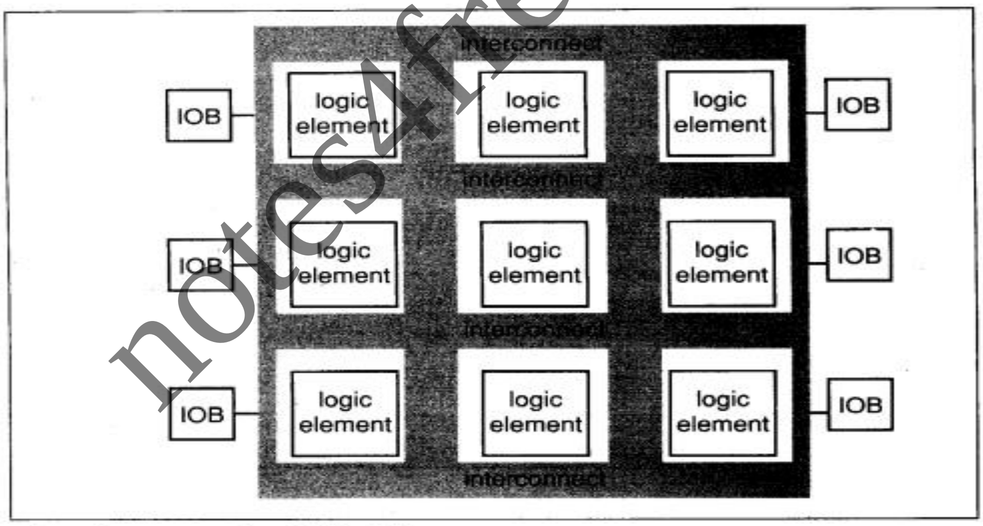
FIGURE C.2 PLA floor plans.

FPGA Based Systems

3.2 FPGA Architecture:

FPGA consists of three major elements.

- ✓ Combinational Logic
- ✓ Interconnect
- ✓ I/O pins

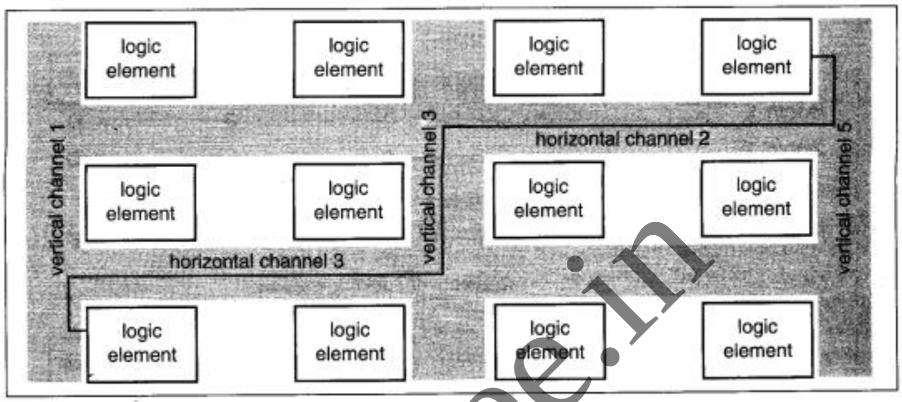


Generic structure of an FPGA fabric.

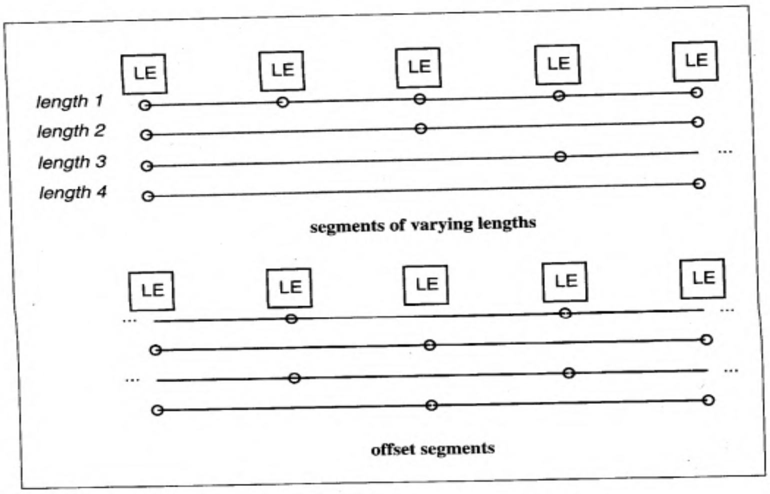
- The combinational logic is divided into small units which is known as logic elements(LE) or combinational logic blocks(CLB).
- LE or CLB usually forms the functions of several logic gates.
- Interconnections between these logic elements are made using programmable interconnects.
- This interconnects are logically organized into channels or other units.
- FPGA offers several interconnects depending on the distance between CLB's that are to be connected: clock signals are provided with their own interconnection networks.
- I/O pins are referred as I/O blocks. These are generally programmable for inputs or outputs and often provides other features such as low power or high speed connection.

FPGA Interconnect:

- The FPGA designer should rely on pre-designed wiring, unlike custom VLSI designer cannot design wires as needed.
- The interconnection system of FPGA is one of the most complex aspect because wiring is global property of logic design.
- Connection between logic elements requires complex paths since LE's are arranged in two dimensional structure as shown in below figure.



- Wires are typically organized in wiring channels or routing channels which runs horizontally and vertically throughout the chip.
- Each channel contains several wires designer or program chooses which wire will carry signal in each channel.
- Connection must be made between wires to carry signal from one point to another. Ex: Net in figure starts from output of LE in upper-right-hand corner travels down vertical channel 5 until it reaches horizontal channel 2, then moves down vertical channel 3 to horizontal channel 3, then it uses vertical channel 1 to reach the input of LE at Lower-left-corner.
- FPGA channel must provide wires of variety of lengths for designer to make all the required connection between logic elements.
- Since LE's are organized in regular array we can arrange wires going from one LE to another. Figure below shows connections of varying length as measured in unit LE's: top signal of length 1 goes to next LE, the second signal goes to second LE and so on. This organization is Known as segment wiring structure.



- All FPGA's need to be programmed or configured.
- There are three major circuit technologies for configuring an FPGA: SRAM, Antifuse and flash.

4.1 Physical Design of FPGA.

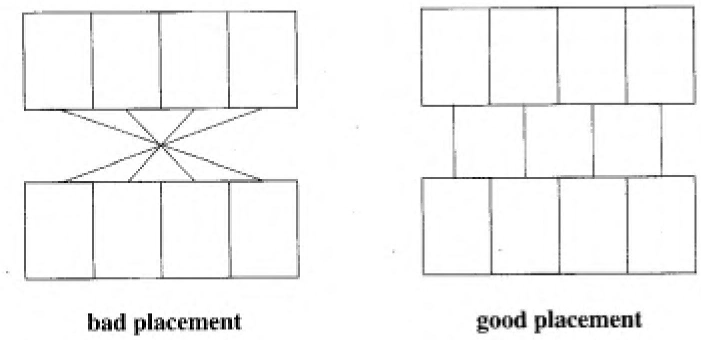
Physical design is divided into two major phases.

- Placement: it determines the position of logic elements and I/O pads.
- Routing: selects the paths for connection between logic elements and I/O pads.
- These two phases interact – one placement of the logic may not be routable whereas a different placement of the same logic can be routed. But this division allows us to make physical design problem for tractable.
- We use several different metrics to judge the quality of a Placement or Routing. Size is the obvious metric we are concerned about whether we can fit complete design on to the chip.
  - In FPGA size is closely tied with routing, the number of logic elements required is determined by logic synthesis. If we cannot find legal routing for given placement, we may need to change placement.
  - Delay is also critical measure in most design. A long delay is not critical, whereas a relatively short delay path that is critical must be carefully considered.

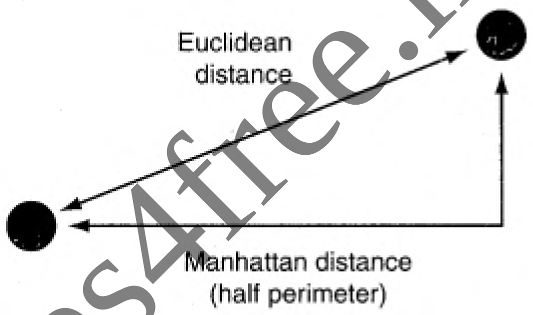
Detailed delay characteristics are somewhat expensive to compute, so tools generally use
surrogates to estimate delay.

Placement

The separation of placement and routing raises an important problem: how to judge the quality of placement? We cannot afford to execute a complete routing for every placement to judge its quality: we need some metric which estimates the quality of routing. Different algorithms use different metrics, but few simple metrics suggests important properties of placement algorithms.



- One way to judge the area of layout before routing is to measure total distance between interconnection. Of course, we can’t measure total wire length without routing the wires, but we can estimate length of a wire by measuring distance between pins it connects.
- When straight lines between connected pins are plotted results is known as rat’s nest plot. In above figure shows two placements, one with longer total rat’s nest connection and one with shorter total interconnections.



- There are several ways to measure distance as shown in figure: Euclidean distance is the direct line between the two. We can also measure Manhattan distance, also known as half perimeter.
- Manhattan distance is more reflective of final wire length, it is also easier to compute because unlike Euclidean distance it does not require computing a square root.
- There are many algorithms for partitioning but these algorithms can generally divided into different approaches: Bottom up and Top Down.
- Bottom up methods are generally referred as clustering methods. These cluster together nodes to create partitions.
- Top Down methods divide nodes into groups that are than further divided. These methods are known as portioning methods.
- Dividing all the components into two partitions doesn’t give very fine direction for placement, the partitioning is usually repeated by creating subpartitions of the original partition, a process known as hierarchical partitioning.
- Clustering algorithms build groups of nodes from the bottom up in contrast to the Top – Down approach taken by partitioning. A small number of nodes create initial clusters. Nodes then added clusters based upon their connections with other nodes.

Routing

- Routing selects paths for connections that must be made between logic elements and I/O pads.
- In FPGA, the interconnection resources are predetermined by the architecture of the FPGA fabric.
- A connection must be made by finding sequence of routing resources, all of which unused and which share connections such that continuous path can be made from source to sink.
- Routing is generally divided into two phases:
  - **Global Routing** selects general path through the chip but does not determine exact wire segments to be used.
  - **Detailed Routing** selects the exact set of wires to be used for each connections.
- Routing has two major cost metrics: wire length and delay. Wire length approximates utilization of routing resources and delay may be measured by looking at the delay on paths with largest number of levels of logic or by looking to nets whose delay is close to maximum allowed value.
- The principal job during global routing of FPGAs is to balance the requirements of various nets.
- Nets are routed one at a time, so the order in which nets are routed affects final result.
- Net may have one of the two problems: 1) it may not be routable because there is no room available to make connection or 2) it may take a path that incurs too much delay.
- These problems are harder to solve in FPGAs than in custom chip designs because routing resources are pre determined.
- Many ways have developed to determine the order in which wires are routed. A good heuristic for initial ordering is to route most delay critical nets first: one may also want to start with large fanout nets since they consume many routing resources.
- In general , a wire may be routed more than once before it finds its final route. Ripup and reroute is one simple strategy for choosing the order in which to route nets.

1.1 Goals and Techniques

Logical function to be performed is only one goal that must be met by FPGA or any digital design. Many other attributes must be satisfied for project to be successful:

- ✓ **Performance:** Logic must run at required rate. It is measured in many ways, such as throughput and latency. Clock rate often measures performance.
- ✓ **Power/energy:** chip must run within an energy or power budget. Energy consumption is clearly critical in battery powered systems
- ✓ **Design time:** FPGAs have standard parts, have several advantages in design time. They can be used as prototypes, can be programmed quickly and can be used as part in final design.

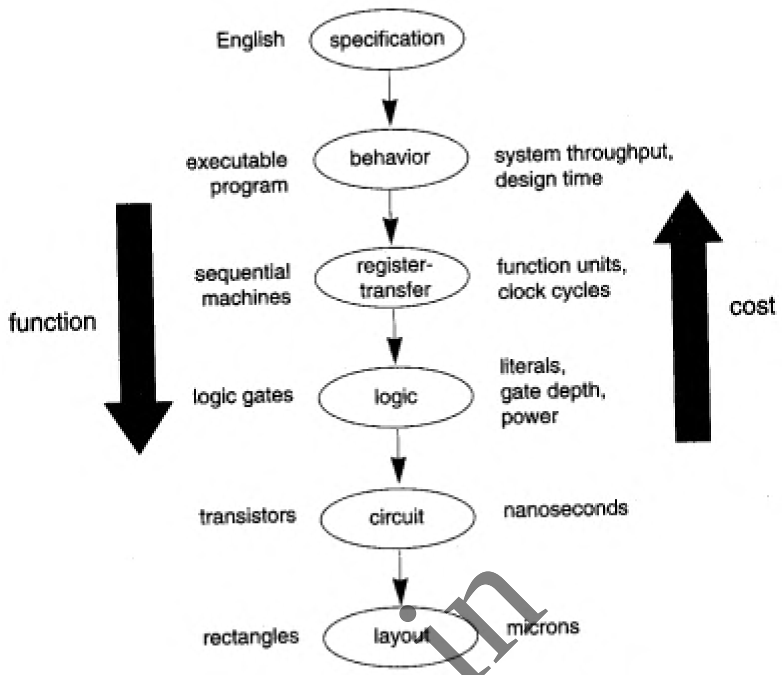
- ✓ **Design cost:** design time is one important component in design cost, but other factors such as required support tools may be considered. FPGA tools are less expensive than custom VLSI tools.
- ✓ **Manufacturing cost:** It is the cost of replicating the system many times. FPGAs are generally more expensive than ASICs due to overhead of programming. But the fact that they are standard parts helps to reduce their cost.

Designing is hard because we have to solve several problems:

- ✓ **Multiple levels of abstraction:** FPGA design requires refining an idea through many levels of detail. Starting from specification of what the chip must do, the designer must create architecture which performs the required function and expand the architecture into logic design.
- ✓ **Multiple and conflicting costs:** costs may be in dollar, such as expense of a particular piece of software needed to design some piece. Costs may also be performance or power consumption of final FPGA.
- ✓ **Short design time:** electronics markets change extremely quickly, getting a chip out faster means reducing your costs and increasing your revenue. Getting it out late may mean no making any money at all.

1.2 Design Abstraction

- ✓ Design abstraction is critical to hardware system design.
- ✓ Hardware designer use multiple levels of design abstraction to manage the design process and ensure that they meet major design goals, such as speed and power consumption.
- ✓ Design abstraction of the FPGA is as shown in below figure.
    - **Behavior:** Explains detailed, executable description of what the chip should do, but it won’t describe how it should do it. Example a C program will not mimic the clock cycle-by- clock cycle behavior of the chip, but it will allow us to describe in detail what needs to be computed, error and boundary conditions etc.
    - **Register transfer:** The system’s time behavior is fully specified- we know the allowed input and output values on every clock cycle but the logic isn’t specified as gates. The system is specified as Boolean functions stored in abstract memory elements. Only delay and area estimates can be made from Boolean functions.
    - **Logic:** The system is designed in terms of Boolean logic gates, latches and flip-flops.
    - **Configuration:** The logic must be placed into logic elements around FPGA and the proper connections must be made between those logic elements. Placement and routing performs these important steps.



Abstraction of FPGA design

- Design always requires working down from the top of abstraction hierarchy and up from least abstract description. Work must begin by adding details to abstraction – top-down design add functional detail. Top–down design decisions are made with limited information.
- Bottom – up analysis and design percolates cost information back to higher levels of abstraction: for instance, we may use more accurate delay information from circuit design to redesign the logic. But most designs requires cycles of Top _Down design followed by bottom – up redesign.

Module 5

Memory, Registers and Aspects of System Timing

System Timing Considerations

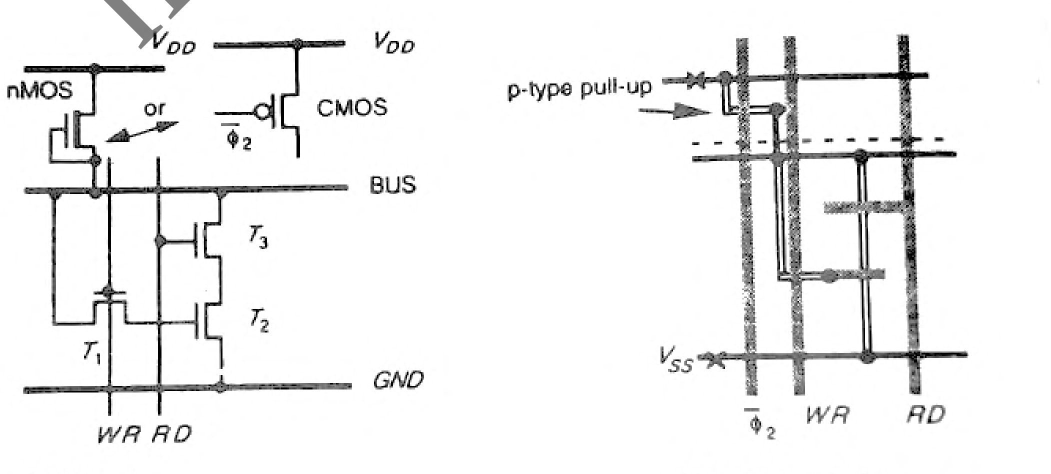
1. Two phase non-overlapping clock is assumed to be available and this clock will be used throughout the system.
2. Clock phases are assumed to be ϕ_1 and ϕ_2 and ϕ_1 is assumed to lead ϕ_2.
3. Bits (data) to be stored are written to registers, storage elements and subsystems **on ϕ_1** of the clock i.e., WR (write) signal is ANDed with ϕ_1.
4. Bits written into storage elements may be assumed to have settled before ϕ_2 signal (which follows immediately) and ϕ_2 signal may be used for refreshing the stored data.
5. Delay through data paths, combinational logic etc., are assumed to be less than the interval between leading edge of ϕ_1 of the clock and leading edge of following ϕ_2 signal.
6. Bits or data may be read from storage elements **on the next of ϕ_1**. RD (read) signal is ANDed with ϕ_1. Thus RD and WR signals are mutually exclusive.
7. General requirement for system stability is that there must be at least one clocked storage element in series with every clocked loop signal path.

Some commonly used storage/memory elements:

The storage elements are compared based on three parameters

1. Area requirement
2. Estimated dissipation per bit stored.
3. Volatility

A three-transistor Dynamic RAM Cell (3T DRAM Cell)



(a) Circuit

(b) CMOS stick diagram

Note: *WR* and *RD* are coincident with ϕ_1.

FIGURE 9.1 Three-transistor dynamic memory cell.

- In this cell arrangement it uses a single transistor for storing data and 2 transistors for each RD and WR access switch.
- It has a pull-up network with either CMOS or nMOS technology and RD/WR circuit as pull-down network.
- The binary data is stored at gate capacitance of transistor in the form of charge; RD and WR are the control lines.
- T1 with T2 is used for writing the data and T3 with T2 is used for reading the data. At point I data is written and read.
- Here T2 is the storage transistor and T1 & T3 are pass transistors which acts as access switches for control lines RD and WR and also for read and write operations.

Write Operation

- WR and RD signals are mutually exclusive i.e., compliment to each other.
- ✓ When WR = 1, RD will be 0
- ✓ Because of WR = 1, T1 is ON but T3 and T2 are OFF.
- ✓ If data bit on bus is 1, as T1 pass transistor is ON it will pass the signal ($V_{DD} – V_{th}$) towards T2. The capacitor is charged to this potential at I
- ✓ If data bit is 0, as T1 is ON it will pass the signal and charge stored at I is 0.
- ✓ After the data is stored at I or capacitor WR signal is made to 0

Read Operation

- For this RD = 1, WR = 0
  - ✓ As WR = 0, T1 is OFF and T3 is ON as RD = 1
  - ✓ T2 will be ON/OFF depending on the voltage/charge stored at I (gate capacitance of T2)
  - ✓ If logic 1 is stored at I, then T2 will be ON. Thus T3 and T2 is ON and path for discharge and the bus is pulled down to ground.
  - ✓ If logic 0 is stored at I, the T2 is OFF and charge does not any path for discharge and retains logic at logic1

Note: The compliment of stored bit is read on the data bus

- In DRAM sensing amplifiers will be connected and as the output begins to decrease from 1 to 0 and this makes the sensing amplifier output as logic 1. If the output does not change then sensing amplifier will make the output as logic 0.

Static Power:

- Static power dissipation is nil since current flows only when i)RD signal is high and ii)logic 1 is stored
- Thus actual dissipation associated with each stored bit depends on the bus pull-up and the duration RD and on switching frequency

Area:

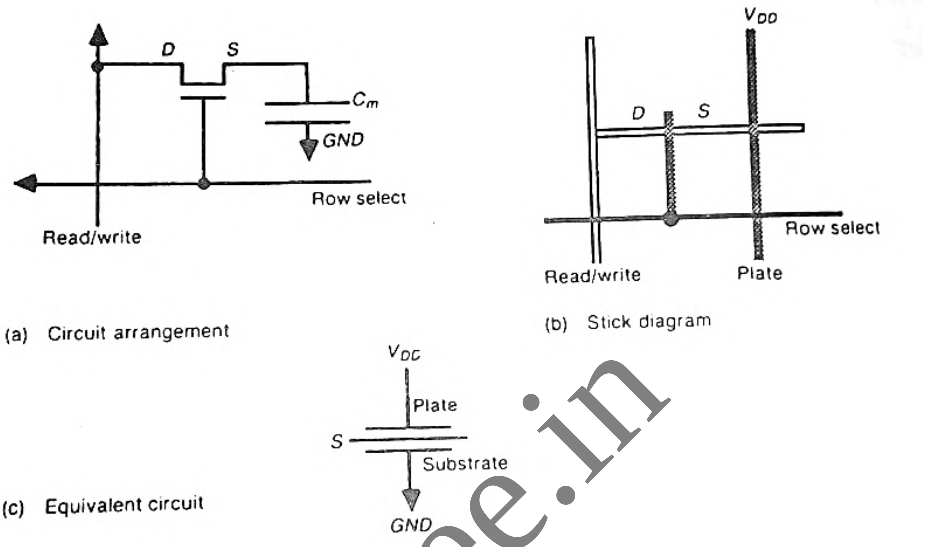
- In 4mm× 4mm, silicon chip area data storage can be >4.8kbits.

Volatility:

- Cell is dynamic and will hold data only for as long as sufficient charge remains on gate capacitance of T2

One-Transistor Dynamic Memory Cell:

- This is an approach which reduces area/bit
- It consists of capacitor Cm and pass transistor. The circuit arrangement and stick diagram is shown in Fig a & b.



(a) Circuit arrangement

(b) Stick diagram

(c) Equivalent circuit

Write operation

- The capacitor Cm will be charged when Read/Write = 1 and Row Select = 1
- If the Read/Write line is provided with logic 1, Cm will be charged to logic 1 and if the line is provided with logic 0 charge stored will be logic 0.

Read operation

- If logic 0 is stored in Cm and when Row select line is high M1 is ON. Then the sense amplifier at the bit line will sense and give the output as logic 0
- If logic 1 is stored in Cm and when Row select line is high M1 is ON, the logic 1 stored will begin to discharge as the path exists. The sense amplifier senses this and this gives the output as logic 1

- The area occupied in 1T DRAM cell configuration is of a single transistor and a capacitor
- Larger the value of Cm longer is the duration of storage of charge. Thus Cm should be large. But this in turn consumes more space.
- However the transistor and capacitor can be built in a single transistor.
- Cm can be fabricated by extending and enlarging the diffusion area forming the source of process transistor. For this capacitance between n-diffusion and p-substrate is considered.
- But this value is very small compared to gate capacitance
- Thus in order to get higher Cm larger area is required.
- An alternate solution to this by using a polysilicon plate used over diffusion area. This results in the formation of a 3 plate capacitor structure, where polysilicon plate is connected to V_{DD}. This is shown in the Fig c.

Area:

- In 4mm× 4mm, silicon chip area data storage is about 12 Kbits.

Dissipation:

- With the cell there is no static dissipation but switching energy while reading and writing must be considered.

Volatility:

- The data in Cm will be held only up-to 1msec or less. Thus periodic refreshing must be provided

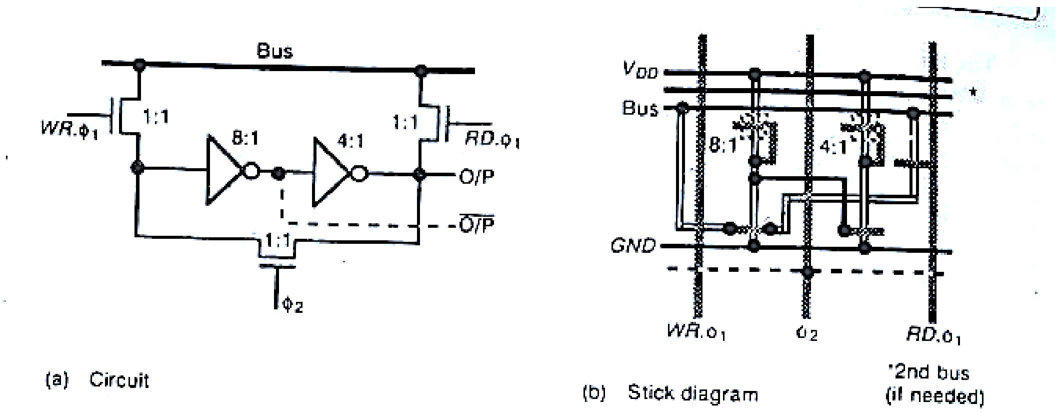
Pseudo-static RAM/register cell:

- This is a memory cell which combines high storage capability of DRAM and ease of use of SRAM
- It can be used as SRAM as no external refreshing circuit is required and also used as a DRAM having built-in refresh logic.
- This is a static storage cell which will hold data indefinitely. This is achieved by storing bit in 2 inverters with feedback. This feedback is used to refresh the data in every clock cycle.
- But care to be taken by not allowing read/write operation during internal refreshing.
- Circuit arrangement is as shown in the Fig.
- Φ_1 and ϕ_2 are mutually exclusive clock signals, WR and RD signal coincides with ϕ_1 signals
- When ϕ_1 is high and WR = 1, transistor T1 is ON and data is charged/stored on Cg (gate capacitance) of inverter. This is write operation
- When ϕ_1 is high and RD = 1, transistor &the data stored at inverter stage is made available at the output and also the compliment. Thus data is read at the output.
- When Φ_2 = 1, T3 is ON. The output is read and feedback i.e., refreshed (reading and storing back the data). The gated feedback path from output of T2 is fed to the input of T1.
- The bit will be held as long as ϕ_2 rescues and this time is less than decay time of stored charged bit.

Note:

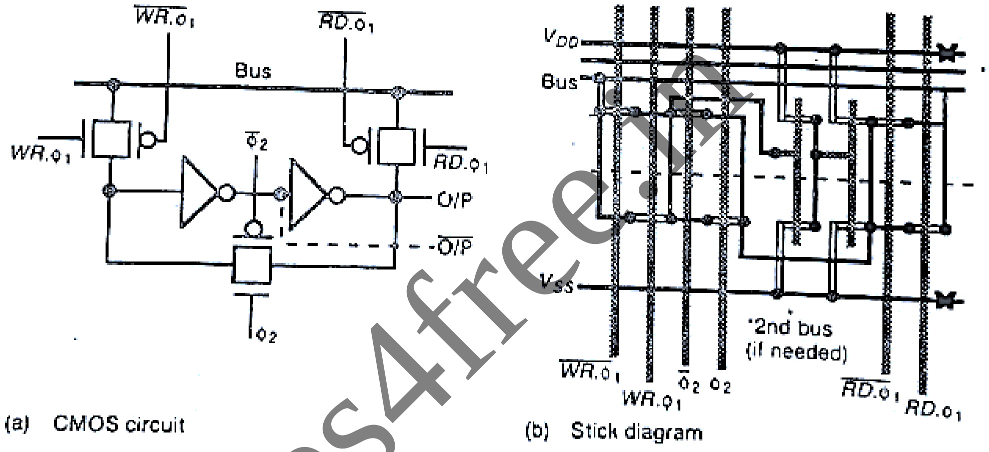
- WR and RD must be mutually exclusive but both should coincide with ϕ_1
- During refreshing of memory cell i.e., at ϕ_2 the cell must not be read. If an attempt is made to read the cell data onto the bus, the charge sharing effect between bus and Cg (input gate capacitance) may cause destruction of stored bit.
- Other bus lines should be allowed to run through the cells so that register and memory arrays can be easily configured.

- The Pseudo-static memory cell can also be implemented using transmission gate (TG). This is seen the Fig. [replace nMOS transistors with TG]



(a) Circuit (b) Stick diagram

FIGURE 9.4 nMOS pseudo-static memory cell.



(a) CMOS circuit (b) Stick diagram

FIGURE 9.5 CMOS pseudo-static memory cell.

Area:

- In 4mm× 4mm, silicon chip area data storage is about 1.4 Kbits.

Dissipation:

- The nMOS cell uses 2 inverters, one with 8:1and other with 4:1 ratio. Thus power dissipation depends on the current drawn and actual geometry of the inverters.

Volatility:

- The cell is non-volatile until unless ϕ_2 is present.

Four Transistor Dynamic and Six-Transistor CMOS memory cell:

- The cells here include both n-type and p-type transistors and are intended for CMOS systems.
- Both the dynamic and static elements uses 2 bus per bit arrangement so that the bit is available in both normal and compliment form on bit and bit' bus
- Prior to reading and writing operation of the data, the buses are pre-charged to V_{DD} or logic 1.

4 Transistor Dynamic memory cell:

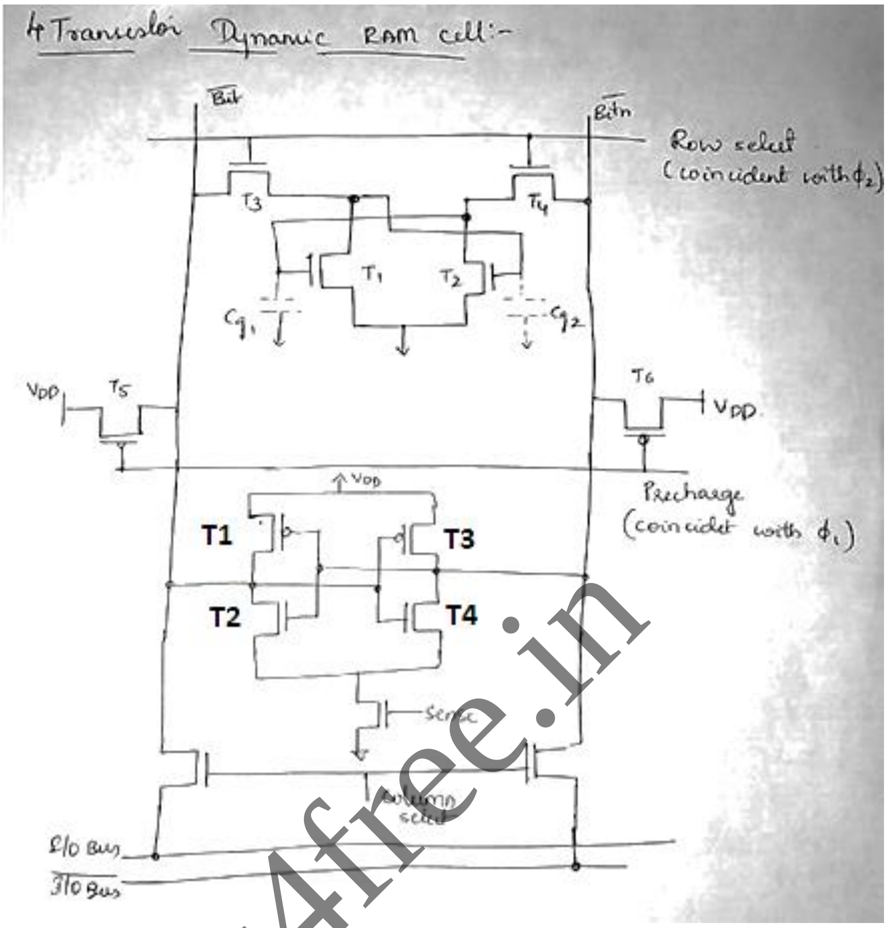

4 Transistor Dynamic RAM cell:-
Bit
Bitn
Row select (coincident with φ2)
T3
T4
T1
T2
Cg1
Cg2
VDD
T5
T6
VDD
Precharge (coincidet with φ1)
VDD
T1
T3
T2
T4
Sense
column select
I/O Bus
I/O Bus

Fig. four transistor dynamic RAM with sense amplifier

Write operation:

- Before writing onto memory the bit and bit’ line is pre-charged to logic 1 using pMOS transistor T5 and T6 in coincidence with clock signal ϕ_1
- Next appropriate column is selected in coincidence with the clock signal ϕ_2.
- Depending on the data on the bus either bit or bit’ is discharged.
- At the same clock signal ϕ_2 the row select line is activated, turning on transistors T3 and T4.
- Thus value on bit and bit’ are written via T3 and T4 stored at T2 and T1 as gate capacitances C_{g2} and C_{g1} respectively.
- The way in which T2 and T1 are connected always gives the complimentary states when row select line is activated. When row line is deactivated the data stored will remain until the gate capacitance can hold the value.
- For refreshing sense amplifier is provided which will permanently hold the data.

Read operation:

- Before reading again bit and bit’ lines are pre-charged to V_{DD} using T5 and T6 transistors.
- Suppose in the memory element if logic 1 is stored i.e., at gate of T2 and at gate of T4 logic 1 is stored.

- When column and row lines are selected i.e., T3 and T4 will be in ON state.
- As logic 1 is available at T2, T2 will be in ON state and T1 will be in OFF state. Thus T3 = ON, T1 = OFF, T4 = ON, T2 = ON. With this condition bit' which was pre-charged to V_{DD} has now a path to discharge to V_{SS}. Hence bit' = 0 and bit = 1 as shown in the Fig.
- When sense amplifier senses this voltage variation on bit' line and outputs the data on bus line. The bit = 1 and bit' = 0, which represents the data in the memory.
- The sense amplifier formed from the arrangement of T1, T2, T3 and T4, which forms a flip flop circuit.
- If the "sense" de-active/ inactive, then the bit line state is reflected in the gate capacitances of T1 and T3 and this is w.r.t V_{DD}. This will cause one of the transistor to turn ON and other to turn OFF.
- When sense = enabled, current flows from V_{DD} through ON transistor and helps to maintain the state of the bit line.
- Sense amplifier performs 2 function
  1. Rewriting the data after reading i.e., refreshing the memory cell so that it holds the data without signal degradation
  2. It predetermines the state of the data lines.

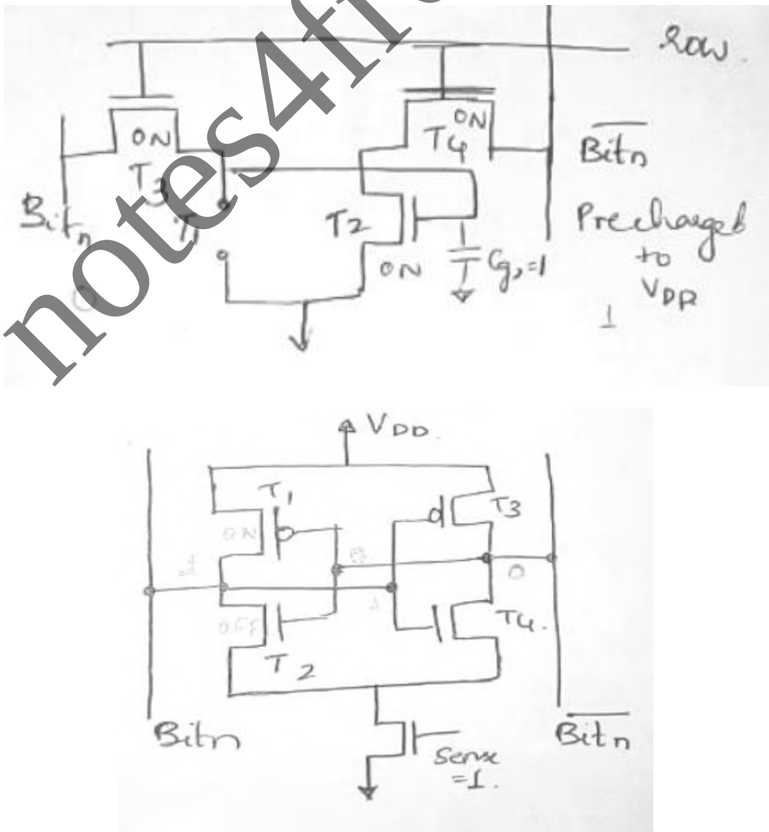


Fig. shows Read operation in the memory cell and in the sense amplifier.

Six Transistor Static RAM cell:

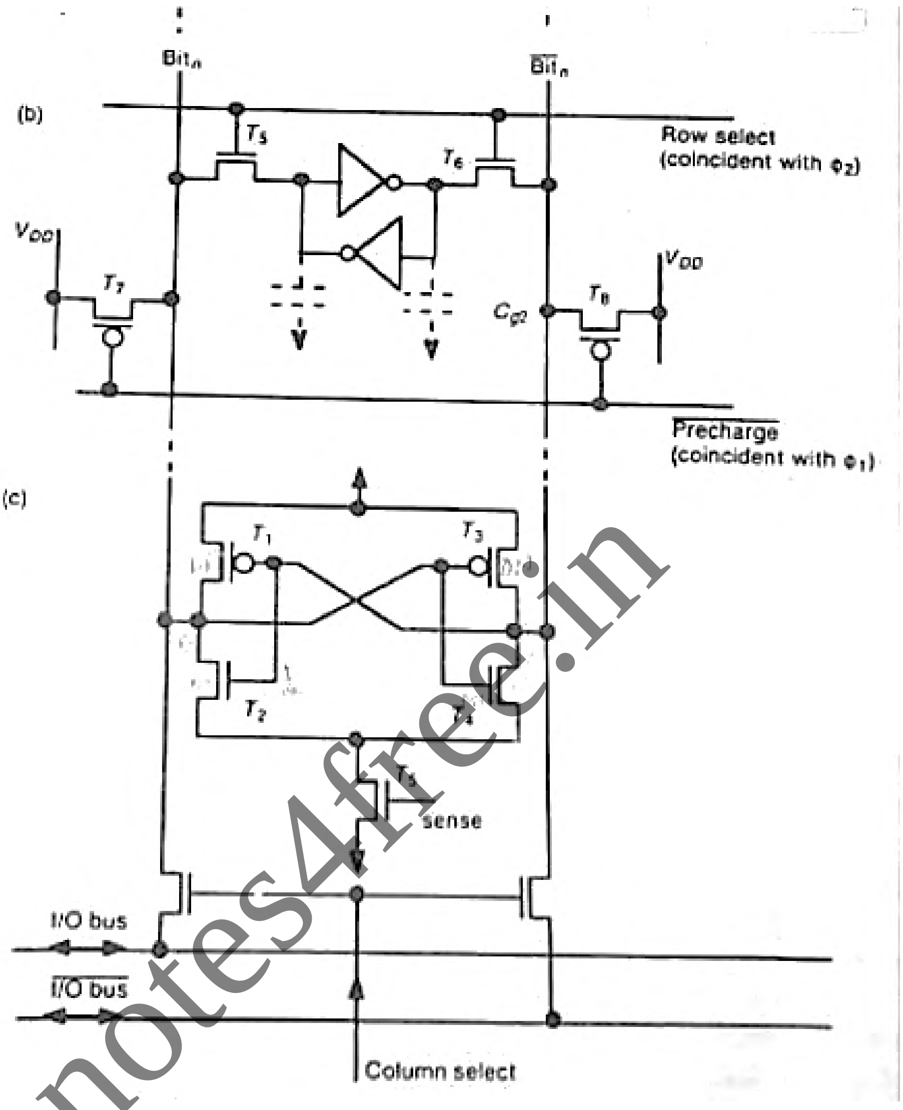


Fig. six transistor static RAM cell with sense amplifier

- Figure shows 6 T SRAM with the adaption of dynamic cell and modifying it to form a static memory cell.
- It includes 2 additional transistor per store bit thus it is called 6 transistor. The transistor T5 and T6 acts as the access switch for memory element which is formed by connecting two inverters back to back (i.e., output of one is connected as the input of the other)
- Similar to 4T Dynamic RAM the information is stored in memory cell. The memory cell is connected in such a way that it gives the complimentary states when row select line is activated. When row line is deactivated the data stored will remain in the memory cell.

Below Fig. shows dynamic and static RAM cell together as the sense amplifier is same in both the memory cell.

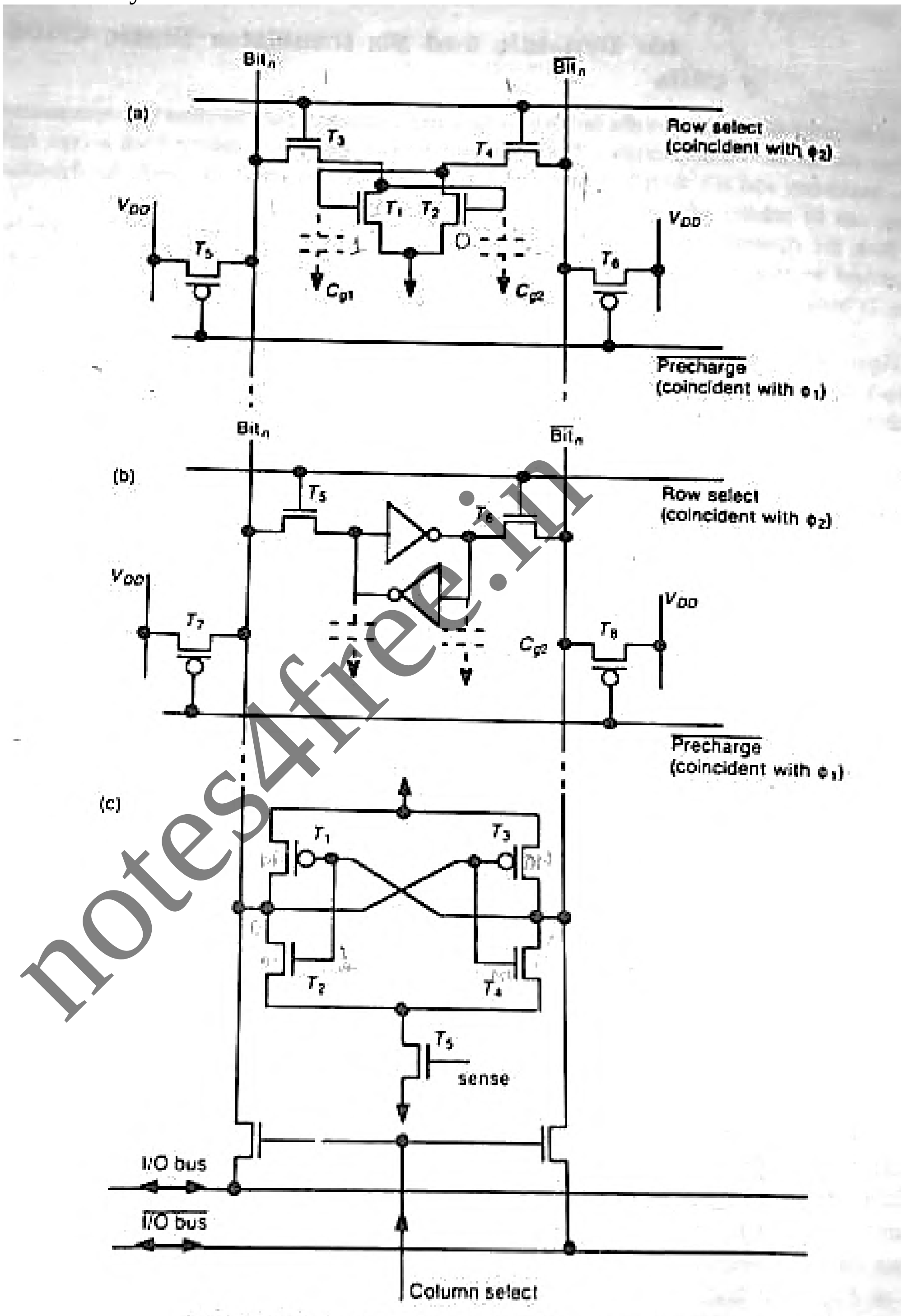


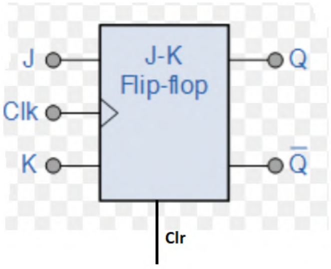
FIGURE 9.7 Dynamic and static memory cells.

JK flip flop:

- It is a memory element. It is the widely used arrangement for static memory element.
- Also with JK other flip-flop arrangements can be obtained such as T and D flip-flop.

- The flip-flop has inputs clocked J and K along with asynchronous clear and has the output as Q and Q'
- The inputs J and K are read for the rising edge of clock signal and data is passed to the output for the falling edge of clock.

Note: here JK is implemented in master slave configuration in order to solve the race around condition



Gate Implementation of flip-flops:

- The expressions for the flip-flops can be implemented using either NAND or NOR logic
- If NAND arrangement is used (in NAND inputs have to be connected in series) then it would take large area. Also it is seen that when connected in series the overall delay increases and in practice not more than 4 transistors should be connected in series. However the number can be increased by including buffers in between and next four transistors. Also it is seen that performance of NAND is slower than NOR.
- In NOR the implementation can be done easily (as the inputs have to be connected in parallel)

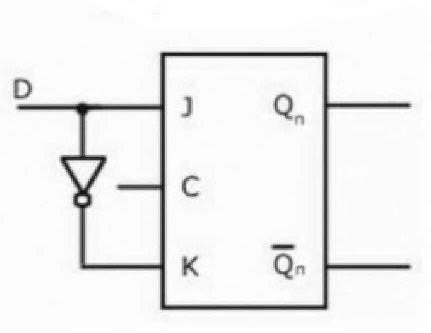
Switch logic and inverter implementation:

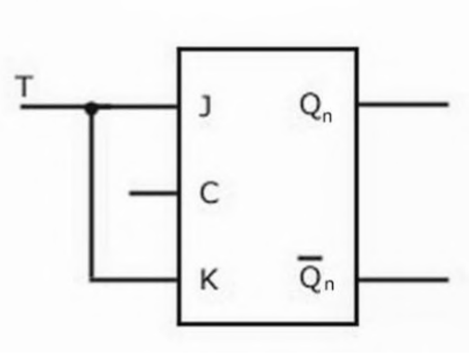
- If n pass transistors are used to realize the logical requirements, it must be kept in mind that
  1. There should be more than 4 pass transistors connected in series.
  2. One pass transistor should not drive the gate of other pass transistor.

D and T flip-flop circuit:

D flip-flop can be formed from JK by connecting an inverter between both inputs.

T flip-flop can be readily formed from JK by connecting JK to form T input. This is shown in the Fig.





Testing and Verification

- In VLSI testing relates to the procedure that takes place after chip is fabricated in order to find any defects.
- There are 2 benefits from testing – Quality and economy
    - Quality – is satisfying user's need at minimum cost and testing weeds/removes all bad products before they reach the user.
    - If more number of products are bad then automatically cost increases. Thus bad products will heavily effect the price of good products.
- Testing is classified into 3 groups.
    1. Logical Verification: The first set of tests verifies that the chip performs its intended function. These tests, called functionality tests or logic verification, are run before tape-out to verify the functionality of the circuit.
    2. The second set of tests called silicon debug are run on the first batch of chips that return from fabrication. These tests confirm that the chip operates as it was intended and help debug any discrepancies. They can be much more extensive than the logic verification tests because the designer has much less visibility into the fabricated chip compared to during design verification.
    3. The third set of tests verify that every transistor, gate, and storage element in the chip functions correctly. These tests are conducted on each manufactured chip before shipping to the customer to verify that the silicon is completely intact. These are called manufacturing tests.
- As manufacturing process is complex, not all die on a wafer may function correctly. Dust particles and small imperfections in starting material or photo-masking can result in bridged connections or missing features. These imperfections result in what is termed a **fault**.
- The goal of a manufacturing test procedure is to determine die (chip) that are good and should be shipped to customers. Testing a die (chip) can occur at the following levels:
    - ✓ Wafer level
    - ✓ Packaged chip level
    - ✓ Board level
    - ✓ System level
    - ✓ Field level

Logic Verification:

- Verification tests are usually the designer first choice that is constructed as part of the design process
- Verification tests is necessary to prove that a synthesized gate description was functionally equivalent to the source RTL. This proves that RTL is equivalent to the design specification at a higher behavioral or specification level of abstraction.
- The behavioral specification might be a verbal description, a plain language textual specification, a description in some high level computer language such as C, a program in a system-modeling language such as System C, or a hardware description language such as VHDL or Verilog, or simply a table of inputs and required outputs.

- Often, designers will have a golden model in one of the previously mentioned formats and this becomes the reference against which all other representations are checked.
- Fig. shows functional equivalence at various levels of abstraction.

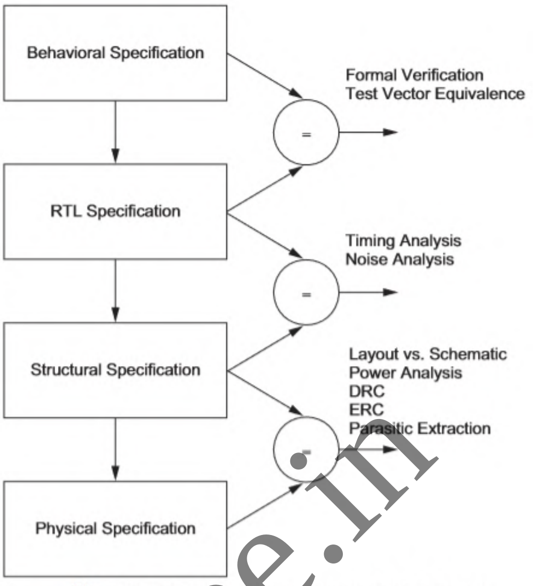


Fig. Functional equivalence at various levels of abstraction

Logic verification principles:

Test vectors: Test vectors are a set of patterns applied to inputs and a set of expected outputs. Both logic verification and manufacturing test require a good set of test vectors. These must be large enough to catch all the logic errors and manufacturing defects, yet small enough to keep test time (and cost) reasonable.

Test bench and Harnesses: A test bench or harness is a piece of HDL code that is placed as a wrapper around a core piece of HDL (stimuli) to apply and check test vectors. In the simplest test bench, input vectors are applied to the module under test and at each cycle, the outputs are examined to determine whether they are same as predefined expected data set. The expected outputs can be derived from the golden model and saved as a file or the value.

Regression Test: High-level language scripts are frequently used when running large test benches, especially for regression testing. Regression testing involves performing a suite of simulations to automatically verify that no functionality has inadvertently changed in a module or set of modules. During a design, it is common practice to run a regression script every night after design activities have concluded to check that bug fixes or feature enhancements have not broken completed modules

Bug Tracking: Another important tool to use during verification is a bug-tracking system. Bug-tracking systems such as the Unix/Linux based GNATS (is a set of tools for tracking tools) allow the management of a wide variety of bugs. In these systems, each bug is entered and the location, nature, and severity of the bug is noted.

Manufacturing Test Principles:

A critical factor in VLSI design is the necessity to incorporate methods of testing circuits. This task should proceed concurrently with architectural considerations and not be left until fabricated parts are available.

Fig a. below shows a combinational circuit with N inputs. To test this circuit exhaustively, a sequence of 2^N inputs (or test vectors) must be applied and observed to fully exercise the circuit.

If this combinational circuit is converted to a sequential circuit with addition of M registers, as shown in **Fig b**. The state of the circuit is determined by the inputs and the previous state. A minimum of 2^{N+M} test vectors must be applied to exhaustively test the circuit. This would take a long time.

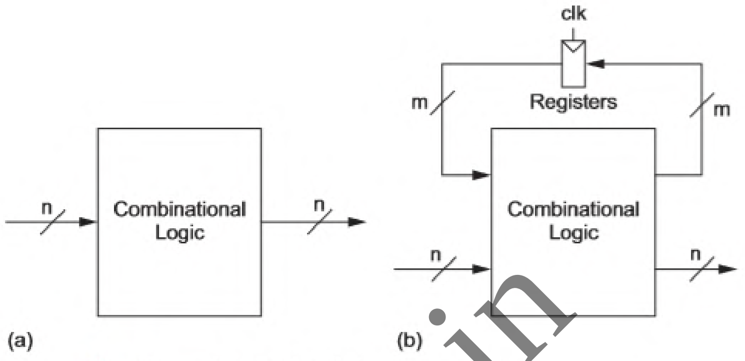


Fig. The combinational explosion in test vectors

Hence, exhaustive testing is infeasible for most systems. Thus the verification engineer must cleverly devise test vectors that detect any (or nearly any) defective node without requiring so many patterns.

Fault Models

- In order to determine good and bad parts in a chip, it is necessary to propose a fault model. This model will help to know where and how faults occur, what is their impact on circuits. The most popular model is called the Stuck-At model. The Short Circuit/ Open Circuit is another model can be a closer fit to reality, but this is difficult to implement in logic simulation tools.

Stuck-At Faults: In the Stuck-At model, a faulty gate input is modeled as a stuck at zero (Stuck-At-0, S-A0) or stuck at one (Stuck-At-l, S-A-l). This model dates from board-level designs, where it was determined to be adequate for modeling faults. Fig illustrates how an S-A-0 or S-A-1fault might occur in basic gates. These faults most frequently occur due to gate oxide shorts (the nMOS gate to GND or the pMOS gate to V_{DD}) or metal-to-metal shorts.

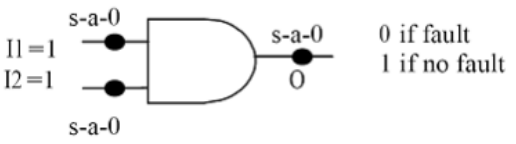


Fig Stuck-at-0 faults in an AND gate

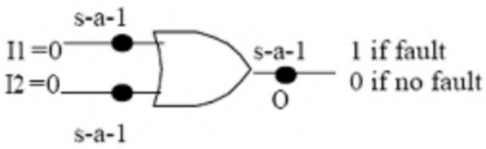


Fig. Stuck-at-1 fault in an OR gate

To test the fault at I1, input pattern is I1=1, I2=1; if the output is 0, s-a-0 fault in I1 is present, else it is absent. Now, also for the s-a-0 fault in net I2, the pattern is I1=1, I2=1.

Short-Circuit and Open-Circuit Faults: Other models include stuck-open or shorted models. Two bridging or shorted faults are shown in Figure. The short S1 results in an S-A-0 fault at input A, while short S2 modifies the function of the gate.

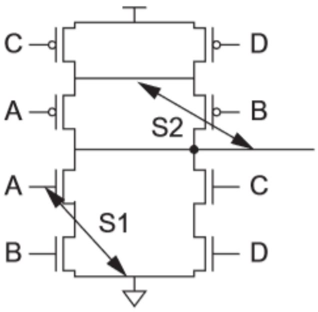


Fig. CMOS Bridging Faults

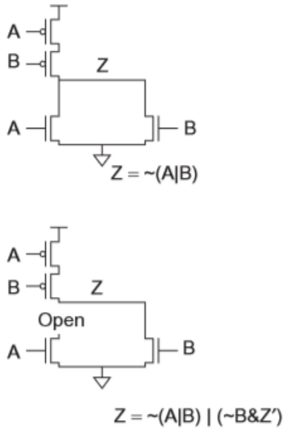


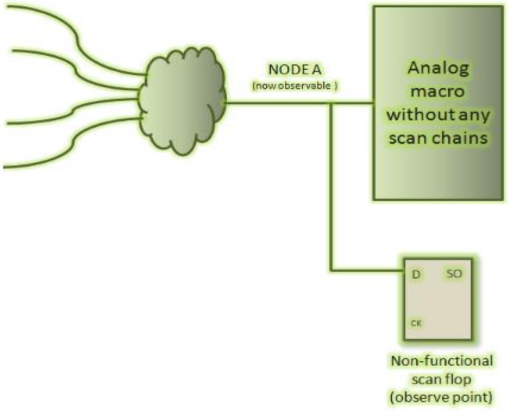
Fig. CMOS open Fault that causes sequential faults

- A particular problem that arises with CMOS is that it is possible for a fault to convert a combinational circuit into a sequential circuit. This is illustrated in the Fig.
- Considering the case of a 2-input NOR gate in which one of the transistors is rendered ineffective. If nMOS transistor A is stuck open, then the function displayed by the gate will be Z = (A + B)’ + BZ’, where Z’ is the previous state of the gate.
- Stuck - closed states can be detected by observing the static V_{DD} current (I_{DD}) while applying test vectors.

Observability:

The observability of a particular circuit node is the degree to which you can observe that node at the outputs of an integrated circuit (i.e., the pins). OR It is the ability to observe the response of the circuit through primary outputs or at some other output points.

This is relevant when you want to measure the output of a gate within a larger circuit to check if it operates correctly. Given the limited number of nodes that can be directly observed, it is the aim of good chip designers to have easily observed gate outputs

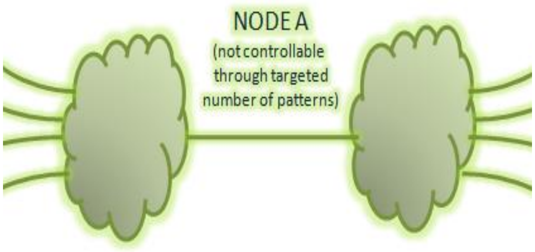

NODE A
(now observable)
Analog macro without any scan chains
D
SO
CK
Non-functional scan flop (observe point)

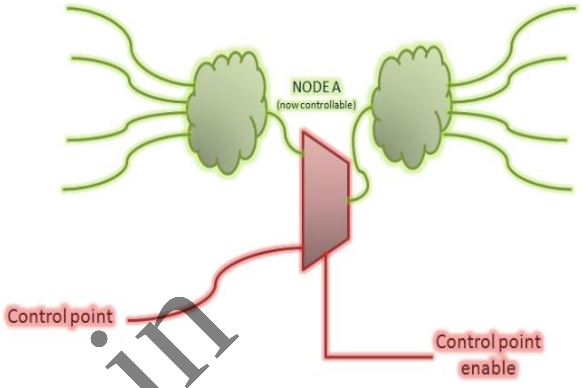
Using some basic design for test techniques can help tremendously in this respect. It should be able to observe every gate output in an circuit directly or with moderate indirection (where have to wait for few cycles). In order to enhance observability the outputs must be observed seperately and this may be done with expense of extra test circuit.

Controllability:

The controllability of an internal circuit node within a chip is a measure of the ease of setting the node to a 1 or 0 state. OR It is the ability to apply test patterns to the inputs of the circuit through primary inputs of the circuit.

This is of important when assessing the degree of difficulty of testing a particular signal within a circuit. An easily controllable node would be directly settable via an input pad.





If a node is little controllability, such as the MSB bit of a counter may need hundreds or thousands of cycles to get it to the right state. And it is highly difficult to generate a test sequence to set a number of poorly controllable nodes.

It should be the aim of good chip designer to make all nodes easily controllable. In common with observability, the adoption of some simple design for test techniques can help in this respect tremendously. Example making all flip-flops resettable via a global reset signal is one step toward good controllability.

Fault Coverage:

- This determines what percent of the chip's internal nodes are checked when the test vectors are applied. The fault coverage of a set of test vectors is the percentage of the total nodes that can be detected as faulty when the vectors are applied.
- The way in which the fault coverage is calculated is as follows:
- Each circuit node is taken in sequence and held to 0 (S-A-0), and the circuit is simulated with the test vectors and then comparing the chip outputs with a known good machine—a circuit with no nodes artificially set to 0 (or 1).
- If any discrepancy is detected between the faulty machine and the good machine, the fault is marked as detected and the simulation is stopped.
- This is repeated for setting the node to 1 (S-A-1). In turn, every node is stuck (artificially) at 1 and 0 sequentially.
- To achieve world-class quality levels, circuits are required to have in excess of 98.5% fault coverage.

Automatic Test Pattern Generation:

In the IC industry, logic and circuit designers implements the functions at the RTL or schematic level, mask designers completes the layout, and test engineers write the tests.

The test engineers took the assistance of designers to include extra circuitry to ease the burden of test generation. With increased complexity and density, the inclusion of test circuitry has become less of an overhead for both the designer.

In addition, as tools have improved, more of the burden for generating tests has fallen on the designer. To deal with this burden, Automatic Test Pattern Generation (ATPG) methods have been invented.

Commercial ATPG tools can achieve excellent fault coverage. However, they are computation-intensive and often must be run on servers or compute farms with many parallel processors.

Some tools use statistical algorithms to predict the fault coverage of a set of vectors without performing as much simulation. Adding scan and built-in self-test improves the observability of a system and can reduce the number of test vectors required to achieve a desired fault coverage.

Delay Fault Coverage

The fault models seen till now point have neglected timing. Failures that might have occured in CMOS would leave the functionality of the circuit untouched, but may affect the timing. For example considering an inverter gate with paralleled nMOS and pMOS transistors. If an open circuit occurs in one of the nMOS transistor source connections to GND, then the gate would still function but with increased t_{pdf} (rising propagation delay). In addition, the fault now becomes sequential as the detection of the fault depends on the previous state of the gate. Delay faults may also be caused by crosstalk. Delay faults can also occur more often in SOI logic through the history effect. Software has been developed to model the effect of delay faults and is becoming more important as a failure mode as processes scale.

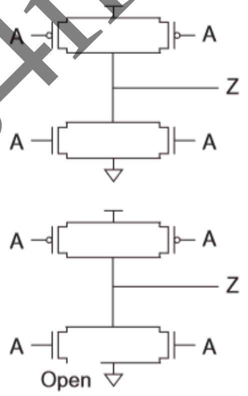


Fig. An example of Delay Fault

Design for Testability:

The keys to designing circuits that are testable are controllability and observability. Controllability is the ability to set (to 1) and reset (to 0) every node internal to the circuit. Observability is the ability to observe, either directly or indirectly, the state of any node in the circuit.

Good observability and controllability reduce the cost of manufacturing testing because they allow high fault coverage with relatively few test vectors.

There are three main approaches to what is commonly called as Design for Testability (DFT). These may be categorized as follows:

- Ad hoc testing
- Scan-based approaches
- Built-in self-test (BIST)

Ad hoc Testing:

Ad hoc test techniques, as their name suggests, are collections of ideas aimed at reducing the combinational explosion of testing. They are only useful for small designs where scan, ATPG, and BIST are not available.
Some of the common techniques for ad hoc testing are:

- ✓ Partitioning large sequential circuits
- ✓ Adding test points
- ✓ Adding multiplexers
- ✓ Providing for easy state reset

Some of the examples are: multiplexers can be used to provide alternative signal paths during testing. In CMOS, transmission gate multiplexers provide low area and delay overhead. Use of the bus in a bus-oriented system for test purposes. Here each register is made loadable from the bus and capable of being driven onto the bus. Here, the internal logic values that exist on a data bus are enabled onto the bus for testing purposes.

Any design should always have a method of resetting the internal state of the chip within a single cycle or at most a few cycles. Apart from making testing easier, this also makes simulation faster as a few cycles are required to initialize the chip. In general Ad hoc testing techniques represent a bag of tricks.

Scan Design:

- The scan-design strategy for testing has evolved to provide observability and controllability at each register.
- In designs with scan, the registers operate in one of two modes.
- In normal mode: they behave as expected
- In scan mode: they are connected to form a giant shift register called a scan chain spanning the whole chip.
- By applying N clock pulses in scan mode, all N bits of state in the system can be shifted out and new N bits of state can be shifted in. Thus scan mode gives easy observability and controllability of every register in the system.
- Modern scan is based on the use of scan registers, as shown in Fig. The scan register is a D flip-flop preceded by a multiplexer. When the SCAN signal is deasserted (made to 0), the register behaves as a conventional register, storing data on the D input. When SCAN is asserted (made to 1), the data is loaded from the SI pin, which is connected in shift register fashion to the previous register Q output in the scan chain.
- To load the scan chain, SCAN is asserted and 8 CLK pulses are given to load the first two ranks of 4-bit registers with data. Then SCAN is deasserted and CLK is asserted for one cycle to operate the circuit normally with predefined inputs. SCAN is then reasserted and CLK asserted eight times to read the stored data out. At the same time, the new register contents can be shifted in for the next test.
- . Testing proceeds in this manner of serially clocking the data through the scan register to the right point in the circuit, running a single system clock cycle and serially clocking the data out for observation. In this scheme, every input to the combinational block can be controlled and every output can be observed.

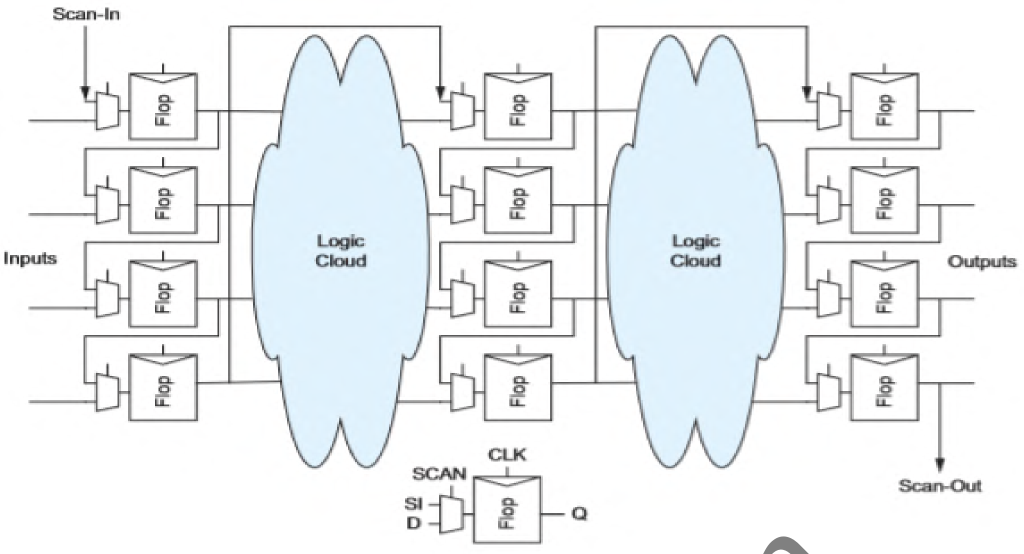


Fig. Scan based testing

- Test generation for this type of test architecture can be highly automated.
- The prime disadvantage is the area and delay impact of the extra multiplexer in the scan register.

Parallel Scan:

Serial scan chains can become quite long, and the loading and unloading can dominate testing time. A simple method/solution is to split the chains into smaller segments. This can be done on a module-by-module basis or completed automatically to some specified scan length. This method is called ‘Random Access Scan’.

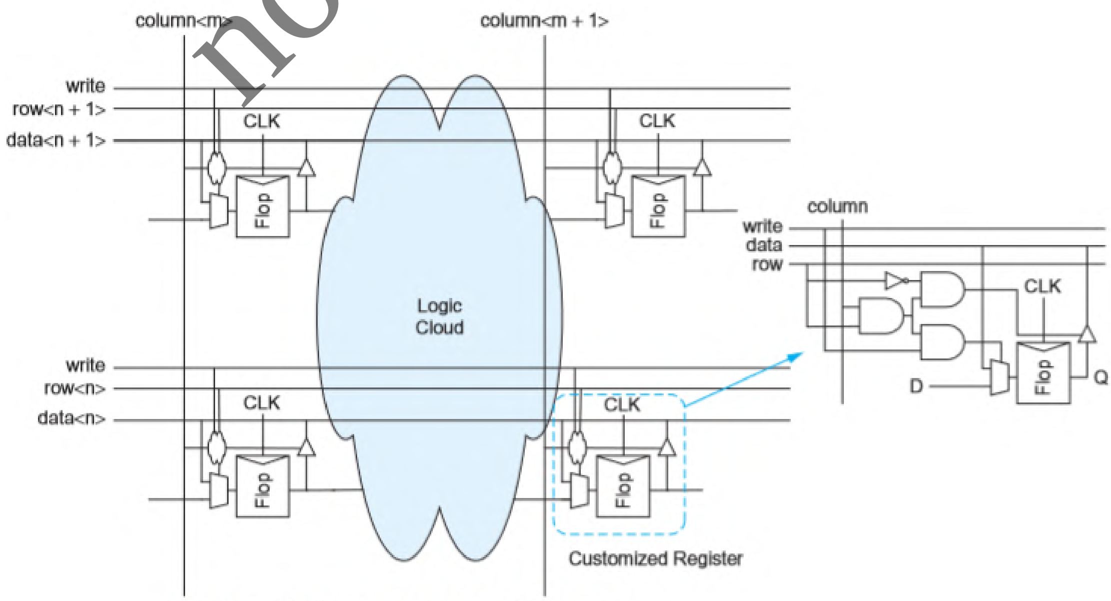


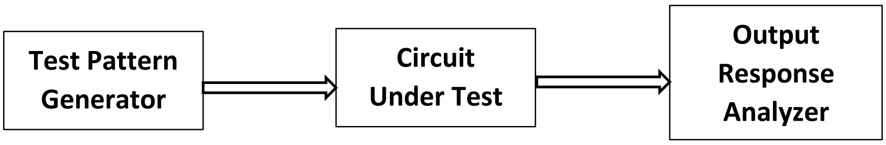
Fig. Parallel Scan - basic architecture

Fig shows a two-by-two register section. Each register receives a column (column<m>) and row (row<n>) access signal along with a row data line (data<n>). A global write signal (write) is connected to all registers. By asserting the row and column access signals in conjunction with the write signal, any register can be read or written.

Built–In Self-Test (BIST):

Built-in test techniques, as their names suggest, rely on augmenting (additional) circuits to allow them to perform operations upon themselves that prove correct operation. These techniques add area to the chip for the test logic, but reduce the test time required and thus can lower the overall system cost.

The structure of BIST is shown below.



- One method of testing a module is to use ‘signature analysis’ or ‘cyclic redundancy checking’. This involves using a pseudo-random sequence generator to produce the input signals for a section of combinational circuitry and a signature analyzer to observe the output signals.
- A PRSG of length n is constructed from a linear feedback shift register (LFSR), which in turn is made of n flip-flops connected in a serial fashion.
- The XOR of particular outputs are fed back to the input of the LFSR. An n-bit LFSR will cycle through $2^{n–1}$ states before repeating the sequence. One problem seen is that it is not possible to generate pattern with all 0’s.

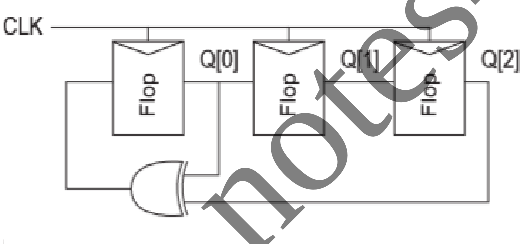


Fig. Linear Feed Back Shift Register (LFSR)

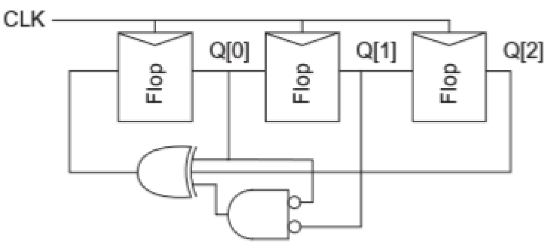


Fig. Complete Feedback Shift Register (CFSR)

- A complete feedback shift register (CFSR), shown in Fig, includes the zero state that may be required in some test situations. An n-bit LFSR is converted to an n-bit CFSR by adding an n – 1 input NOR gate connected to all but the last bit. When in state 0…01, the next state is 0…00.
- A signature analyzer receives successive outputs of a combinational logic block and produces a syndrome that is a function of these outputs. The syndrome is reset to 0, and then XORed with the output on each cycle.
- The syndrome is present in each cycle so that a fault in one bit is unlikely to cancel itself out. At the end of a test sequence, the LFSR contains the syndrome that is a function of all previous outputs. This can be compared with the correct syndrome (derived by running a test program on the good logic) to determine whether the circuit is good or bad.

BILBO – Built-In Logic Block Observation:

- The combination of signature analysis and the scan technique is the formation of BILBO

- The 3-bit BIST register shown in Fig is a scannable, resettable register that also can serve as a pattern generator and signature analyzer.
- This structure can operate in different mode as shown in table below

C[1]	C[0]	Mode
0	0	Scan
0	1	Test
1	0	Reset
1	1	Normal

- In the reset mode (10), all the flip-flops are synchronously initialized to 0. In normal mode (11), the flip-flops behave normally with their D input and Q output. In scan mode (00), the flip-flops are configured as a 3-bit shift register between SI and SO. In test mode (01), the register behaves as a pseudo-random sequence generator or signature analyzer.

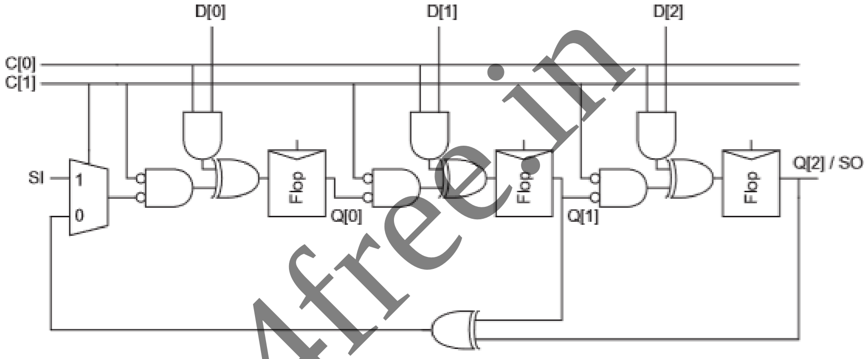


Fig. 3 bit - Register with BILBO

- In summary, BIST is performed by first resetting the syndrome in the output register. Then both registers are placed in the test mode to produce the pseudo-random inputs and calculate the syndrome. Finally, the syndrome is shifted out through the scan chain.

Memory BIST:

On many chips, memories involves with majority of the transistors. A robust testing methodology must be applied to provide reliable parts. In a typical MBIST scheme, multiplexers are placed on the address, data, and control inputs for the memory to allow direct access during test. During testing, a state machine uses these multiplexers to directly write a checkerboard pattern of alternating 1s and 0s. The data is read back, checked, then the inverse pattern is also applied and checked. ROM testing is even simpler: The contents are read out to a signature analyzer to produce a syndrome.