# VLSI DESIGN
## (15A04604)

## LECTURE NOTES

## B.TECH

III-YEAR& II-SEM

Prepared by:

Dr S Murali Mohan, Professor

Department of Electronics and Communication Engineering

MOTHER THERESA INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Approved By AICTE, New Delhi and Affiliated to JNTUA, Ananthapuramu)
Accredited By NAAC & ISO: 9001-2015 Certified Institution
Melumoi Post, Palamaner
Chittoor, Andhra Pradesh - 517 408

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR**
**B. Tech III-IISem. (ECE)**

**L T P C**
**3 1 0 3**

## 15A04604 VLSI DESIGN

**Course Objectives:**

- To understand VLSI circuit design processes.
- To understand basic circuit concepts and designing Arithmetic Building Blocks.
- To have an overview of Low power VLSI.

**Course Outcomes:**

- Complete Knowledge about Fabrication process of ICs
- Able to design VLSIcircuits as per specifications given.
- Capable of optimizingthe design of Arithmetic / logic building Blocks at all levels of Design/Fabrication.
- Can implement circuit through various design styles ( semi- Custom, Full Custom)

**UNIT-I**

**Introduction:** Basic steps of IC fabrication, PMOS, NMOS, CMOS &BiCMOS,and SOI process technologies, MOS transistors - MOS transistor switches – Basic gate using switches, working polartransistor Resistors and Capacitors.

**Basic Electrical Properties of MOS and BiCMOS Circuits:** Working of MOS transistors – threshold voltage; MOS design equations: **Ids–Vds** relationships, Threshold Voltage, Body effect, Channel length modulation , **gm**, **gds**, figure of merit **ω0**; Pass transistor, NMOS Inverter, CMOS Inverter analysis and design, Various pull ups loads,Bi-CMOS Inverters.

**UNIT-II**

**Basic Circuit Concepts:** Capacitance, resistance estimations- Sheet Resistance Rs, MOSDivice Capacitances, routing apacitance, Analytic Inverter Delays, Driving large Capacitive Loads, Fan-in and fan-out.

**VLSI Circuit Design Processes:** VLSI Design Flow, MOS Layers, Stick Diagrams, Design Rules and Layout, 2μm CMOS Design rules for wires, Contacts and Transistors Layout Diagrams for NMOS and CMOS Inverters and Gates, Scaling of MOS circuits, Limitations of Scaling.

**UNIT-III**

**Gate level Design:** Logic gates and other complex gates, Switch logic, Alternate gate circuits.

**Physical Design:** Floor-Planning, Placement, routing, Power delay estimation, Clock and Power routing

**UNIT-IV**

**Subsystem Design**: Shifters, Adders, ALUs, Multipliers, Parity generators, Comparators, Counters, High Density Memory Elements.

**VLSI Design styles**: Full-custom, Standard Cells, Gate-arrays, FPGAs, CPLDs and Design Approach for Full-custom and Semi-custom devices.

**UNIT-V**

**VHDL Synthesis:** VHDL Synthesis, Circuit Design Flow, Circuit Synthesis, Simulation, Layout, Design capture tools, Design Verification Tools.

**Test and Testability:** Fault-modeling and simulation, test generation, design for testability, Built-in-self-test.

**TEXT BOOKS**:

1. Kamran Eshraghian, Eshraghian Douglas and A. Pucknell, "Essentials of VLSI circuits and systems", PHI, 2013 Edition.
2. K.Lal Kishore and V.S.V. Prabhakar, "VLSI Design", IK Publishers

**REFERENCES:**

1. Weste and Eshraghian, "Principles of CMOS VLSI Design", Pearson Education, 1999.
2. Wayne Wolf, "Modern VLSI Design", Pearson Education, 3rd Edition, 1997.
3. John P. Uyemura, "Chip Design for Submicron VLSI: CMOS layout and Simulation", Thomson Learning.
4. John P. Uyemura, "Introduction to VLSI Circuits and Systems", John wiley, 2003.
5. John M. Rabaey, "Digital Integrated Circuits", PHI, EEE, 1997

# UNIT-I

# IC Technologies

- Introduction
- MOS
- PMOS
- NMOS
- CMOS
  &
- BiCMOS
  Technologies

**Basic Electrical Properties of MOS and BiCMOS Circuits**

- $I_{DS}$ - $V_{DS}$ relationships
- MOS transistor Threshold Voltage - $V_T$ figure of merit-ω0
- Transconductance-$g_m$, $g_{ds}$;
- Pass transistor
- NMOS Inverter, Various pull ups, CMOS Inverter analysis and design
- Bi-CMOS Inverters

## INTRODUCTION TO IC TECHNOLOGY

The development of electronics endless with invention of vaccum tubes and associated electronic circuits. This activity termed as vaccum tube electronics, afterward the evolution of solid state devices and consequent development of integrated circuits are responsible for the present status of communication, computing and instrumentation.

- The first vaccum tube diode was invented by **john ambrase Fleming** in 1904.
- The vaccum triode was invented by **lee de forest** in 1906.

Early developments of the Integrated Circuit (IC) go back to 1949. German engineer Werner Jacobi filed a patent for an IC like semiconductor amplifying device showing five transistors on a common substrate in a 2-stage amplifier arrangement. Jacobi disclosed small cheap of hearing aids.

Integrated circuits were made possible by experimental discoveries which showed that semiconductor devices could perform the functions of vacuum tubes and by mid-20th-century technology advancements in semiconductor device fabrication.

The integration of large numbers of tiny transistors into a small chip was an enormous improvement over the manual assembly of circuits using electronic components.

The integrated circuits mass production capability, reliability, and building-block approach to circuit design ensured the rapid adoption of standardized ICs in place of designs using discrete transistors.

**An integrated circuit (IC) is a small semiconductor-based electronic device consisting of fabricated transistors, resistors and capacitors. Integrated circuits are the building blocks of most electronic devices and equipment. An integrated circuit is also known as a chip or microchip.**

There are two main advantages of ICs over discrete circuits: cost and performance. Cost is low because the chips, with all their components, are printed as a unit by photolithography rather than being constructed one transistor at a time. Furthermore, much less material is used to construct a packaged IC die than a discrete circuit. Performance is high since the components switch quickly and consume little power (compared to their discrete counterparts) because the components are small and positioned close together. As of 2006, chip areas range from a few square millimeters to around 350 $mm^2$, with up to 1 million transistors per mm

## IC Invention:

| Inventor | Year | Circuit | Remark |
|---|---|---|---|
| Fleming | 1904<br>1906 | Vacuum tube diode<br>Vacuum triode | large expensive, power-hungry, unreliable |
| William Shockley (Bell labs) | 1945 | Semiconductor replacing vacuum tube | -- |
| Bardeen and Brattain and Shockley (Bell labs) | 1947 | Point Contact transfer resistance device "BJT" | Driving factor of growth of the VLSI technology |
| Werner Jacobi (Siemens AG) | 1949 | 1st IC containing amplifying Device 2stage amplifier | No commercial use reported |
| Shockley | 1951 | Junction Transistor | "Practical form of transistor" |
| Jack Kilby<br>(Texas Instruments) | July 1958 | Integrated Circuits F/F With 2-T Germanium slice and gold wires | Father of IC design |
| Noyce Fairchild Semiconductor | Dec. 1958 | Integrated Circuits Silicon | "The Mayor of Silicon Valley" |
| Kahng Bell Lab | 1960 | First MOSFET | Start of new era for semiconductor industry |
| Fairchild Semiconductor And Texas | 1061 | First Commercial<br>IC | |
| Frank Wanlass<br>(Fairchild Semiconductor) | 1963 | CMOS | |
| Federico Faggin<br>(Fairchild Semiconductor) | 1968 | Silicon gate IC technology | Later Joined Intel to lead first CPU Intel 4004 in 1970<br>2300 T on 9mm$^2$ |
| Zarlink Semiconductors | Recently | M2A capsule for endoscopy | take photographs of digestive tract 2/sec. |

**Moore's Law:**

- Gordon E. Moore - Chairman Emeritus of Intel Corporation
- 1965 - observed trends in industry - of transistors on ICs vs release dates
- Noticed number of transistors doubling with release of each new IC generation
- Release dates (separate generations) were all 18-24 months apart

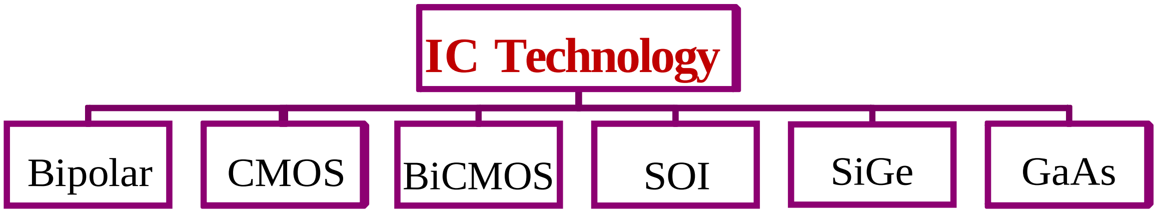
**"The number of transistors on an integrated circuit will double every 18 months"**

The level of integration of silicon technology as measured in terms of number of devices per IC Semiconductor industry has followed this prediction with surprising accuracy.

**IC Technology:**

- Speed / Power performance of available technologies
- The microelectronics evolution
- SIA Roadmap
- Semiconductor Manufacturers 2001 Ranking

## Circuit Technology


IC Technology

Bipolar

CMOS

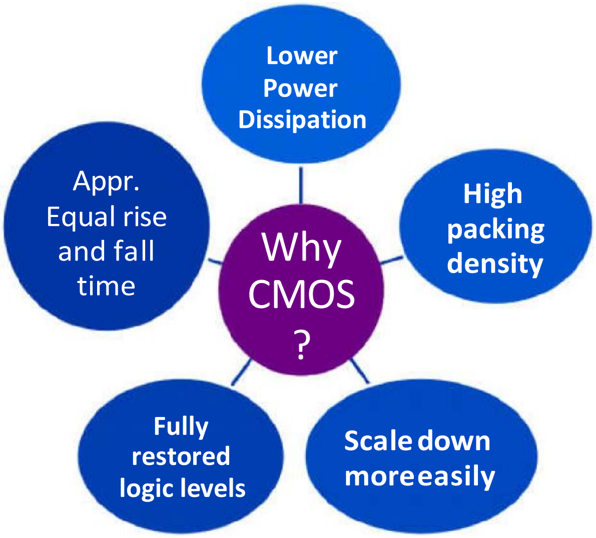
BiCMOS

SOI

SiGe

GaAs

| Category | BJT | CMOS |
|---|---|---|
| Power Dissipation | Moderate to High | less |
| Speed | Faster | Fast |
| Gm | 4ms | 0.4ms |
| Switch implementation | poor | Good |
| Techn ology improvement | slower | Faster |


Why CMOS ?

Lower Power Dissipation

High packing density

Scale down moreeasily

Fully restored logic levels

Appr. Equal rise and fall time

## Scale of Integration:

- **Small scale integration(SSI) --1960**

  The technology was developed by integrating the number of transistors of 1-100 on a single chip. Ex: Gates, flip-flops, op-amps.

- **Medium scale integration(MSI) --1967**

  The technology was developed by integrating the number of transistors of 100-1000 on a single chip. Ex: Counters, MUX, adders, 4-bit microprocessors.

- **Large scale integration(LSI) --1972**

  The technology was developed by integrating the number of transistors of 1000-10000 on a single chip. Ex:8-bit microprocessors,ROM,RAM.

- **Very large scale integration(VLSI) -1978**

  The technology was developed by integrating the number of transistors of 10000-1Million on a single chip. Ex:16-32 bit microprocessors, peripherals, complimentary high MOS.

- **Ultra large scale integration(ULSI)**

  The technology was developed by integrating the number of transistors of 1Million-10 Millions on a single chip. Ex: special purpose processors.

- **Giant scale integration(GSI)**

  The technology was developed by integrating the number of transistors of above 10 Millions on a single chip. Ex: Embedded system, system on chip.

- ✓ Fabrication technology has advanced to the point that we can put a complete system on a single chip.
- ✓ Single chip computer can include a CPU, bus, I/O devices and memory.
- ✓ This reduces the manufacturing cost than the equivalent board level system with higher performance and lower power.

## MOS TECHNOLOGY:

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

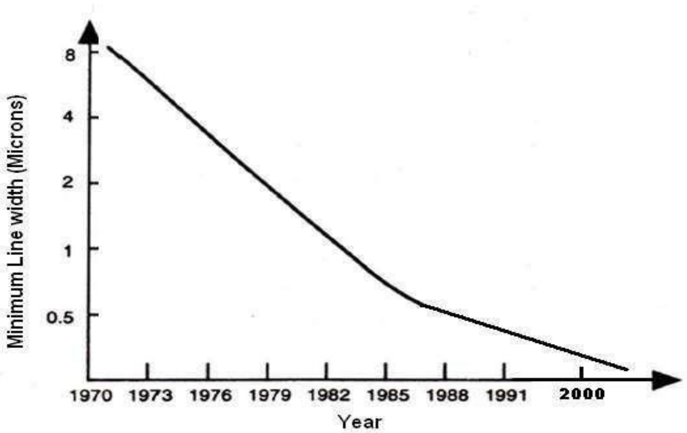
The pMOS devices are based on the p-channel MOS transistors. Specifically, the pMOS channel is part of a n-type substrate lying between two heavily doped p+ wells beneath the source and drain electrodes. Generally speaking, a pMOS transistor is only constructed in consort with an NMOS transistor.

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

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

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

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



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

**MOS Transistor Symbol:**

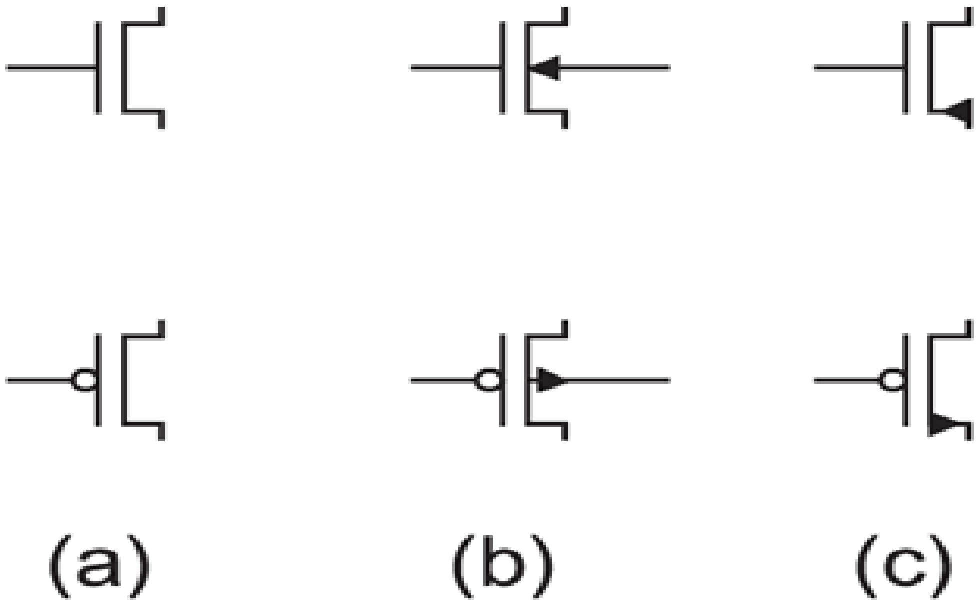


FIG 2.1 MOS transistor symbols

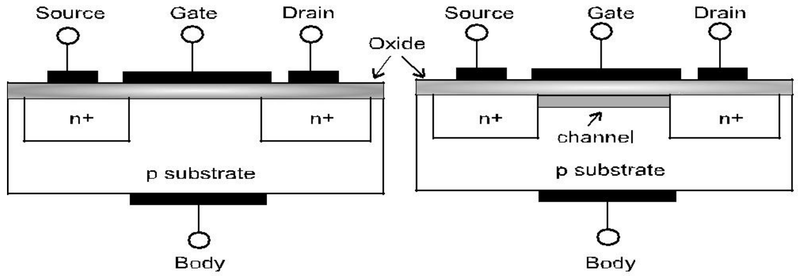
## ENHANCEMENT AND DEPLETION MODE MOS TRANSISTORS

MOS Transistors are built on a silicon substrate. Silicon which is a group IV material is the eighth most common element in the universe by mass, but very rarely occurs as the pure free element in nature. It is most widely distributed in dusts, sands, planetoids, and planets as various forms of silicon dioxide (silica) or silicates. It forms crystal lattice with bonds to four neighbours. Silicon is a semiconductor. Pure silicon has no free carriers and conducts poorly. But adding dopants to silicon increases its conductivity. If a group V material i.e. an extra electron is added, it forms an n-type semiconductor. If a group III material i.e. missing electron pattern is formed (hole), the resulting semiconductor is called a p-type semiconductor.

A junction between p-type and n-type semiconductor forms a conduction path. Source and Drain of the Metal Oxide Semiconductor (MOS) Transistor is formed by the “doped” regions on the

surface of chip. Oxide layer is formed by means of deposition of the silicon dioxide ($SiO_2$) layer which forms as an insulator and is a very thin pattern. Gate of the MOS transistor is the thin layer of “polysilicon (poly)”; used to apply electric field to the surface of silicon between Drain and Source, to form a “channel” of electrons or holes. Control by the Gate voltage is achieved by modulating the conductivity of the semiconductor region just below the gate. This region is known as the channel.
The Metal–Oxide–Semiconductor Field Effect Transistor (MOSFET) is a transistor which is a voltage-controlled current device, in which current at two electrodes, drain and source is controlled by the action of an electric field at another electrode gate having in-between semiconductor and a very thin metal oxide layer. It is used for amplifying or switching electronic signals.

The Enhancement and Depletion mode MOS transistors are further classified as N-type named NMOS (or N-channel MOS) and P-type named PMOS (or P-channel MOS) devices. Figure 1.5 shows the MOSFETs along with their enhancement and depletion modes.



Fi

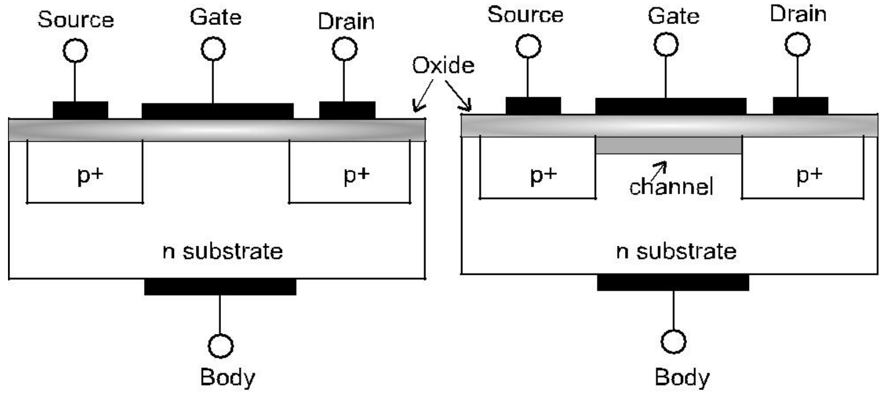


**Figure 1.5:** (c) Enhancement P-type MOSFET (d) Depletion P-type MOSFET

The depletion mode devices are doped so that a channel exists even with zero voltage from gate to source during manufacturing of the device. Hence the channel always appears in the device. To control the channel, a negative voltage is applied to the gate (for an N-channel device), depleting the channel, which reduces the current flow through the device. In essence, the depletion-mode device is equivalent to a closed (ON) switch, while the enhancement-mode device does not have the built in channel and is equivalent to an open (OFF) switch. Due to the difficulty of turning off the depletion mode devices, they are rarely used

**Working of Enhancement Mode Transistor**

The enhancement mode devices do not have the in-built channel. By applying the required potentials, the channel can be formed. Also for the MOS devices, there is a threshold voltage ($V_t$), below which not enough charges will be attracted for the channel to be formed. This threshold voltage for a MOS transistor is a function of doping levels and thickness of the oxide layer.

**Case 1: $V_{gs}$ = 0V and $V_{gs} < V_t$**

The device is non-conducting, when no gate voltage is applied ($V_{gs}$ = 0V) or ($V_{gs} < V_t$) and also drain to source potential $V_{ds} = 0$. With an insufficient voltage on the gate to establish the channel region as N-type, there will be no conduction between the source and drain. Since there is no conducting channel, there is no current drawn, i.e. $I_{ds} = 0$, and the device is said to be in the **cut-off region**. This is shown in the Figure 1.7 (a).

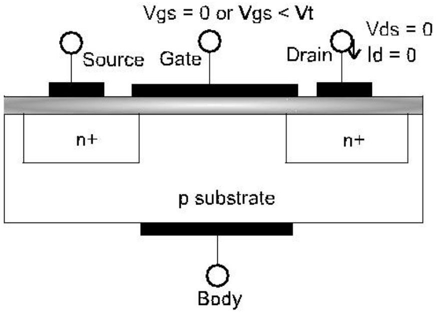


**Figure 1.7:** (a) Cut-off Region

**Case 2: Vgs > Vt**

When a minixmum voltage greater than the threshold voltage $V_t$ (i.e. $V_{gs} > V_t$) is applied, a high concentration of negative charge carriers forms an inversion layer located by a thin layer next to the interface between the semiconductor and the oxide insulator. This forms a channel between the source and drain of the transistor. This is shown in the Figure 1.7 (b).

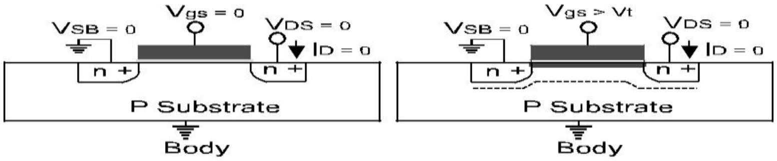


**Figure 1.7:** (b) Formation of a Channel

A positive $V_{ds}$ reverse biases the drain substrate junction, hence the depletion region around the drain widens, and since the drain is adjacent to the gate edge, the depletion region widens in the channel. This is shown in Figure 1.7 (c). This results in flow of electron from source to drain resulting in current Ids.. The device is said to operate in **linear region** during this phase. Further increase in $V_{ds,}$ increases the reverse bias on the drain substrate junction in contact with the inversion layer which causes inversion layer density to decrease. This is shown in Figure 1.7 (d). The point at which the inversion layer density becomes very small (nearly zero) at the drain end is termed pinch-off. The value of $V_{ds}$ at pinch-off is denoted as $V_{ds,sat}$. This is termed as **saturation region** for the MOS device. Diffusion current completes the path from source to drain in this case, causing the channel to exhibit a high resistance and behaves as a constant current source.

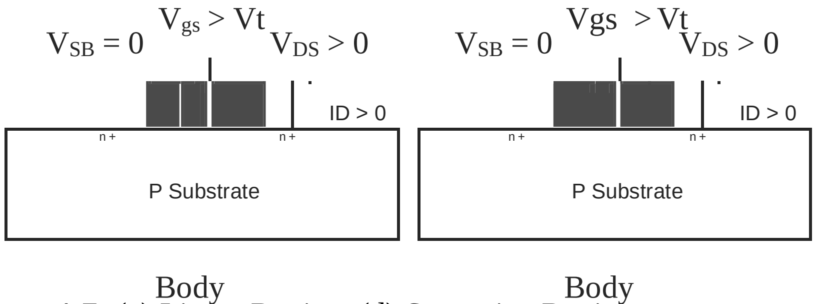


**Figure 1.7:** (c) Linear Region. (d) Saturation Region

The MOSFET $I_D$ versus $V_{DS}$ characteristics (V-I Characteristics) is shown in the Figure 1.8. For $V_{GS} < V_{t,}$ $I_D = 0$ and device is in cut-off region. As $V_{DS}$ increases at a fixed $V_{GS}$, $I_D$ increases in the linear region due to the increased lateral field, but at a decreasing rate since the inversion layer density is decreasing. Once pinch-off is reached, further increase in $V_{DS}$ results in increase in $I_D$; due to the formation of the high field region which is very small. The device starts in linear region, and moves into saturation region at higher $V_{DS}$.

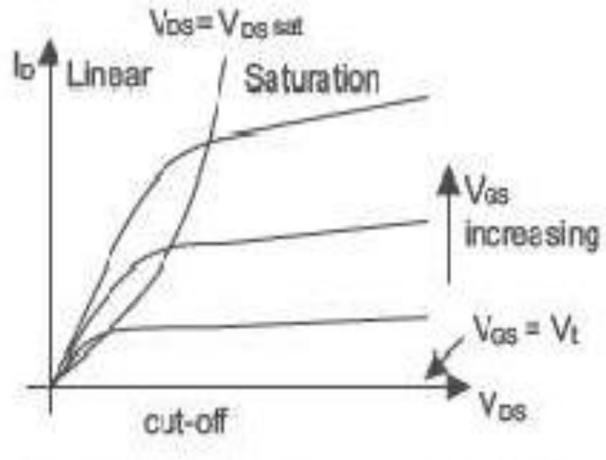


**Figure 1.8:** MOS V-I Characteristics

## NMOS FABRICATION

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

(a) The fabrication process starts with the oxidation of the silicon substrate.

It is shown in the Figure 1.9 (a).

(b) A relatively thick silicon dioxide layer, also called field oxide, is created on the surface of the substrate. This is shown in the Figure 1.9 (b).

(c) Then, the field oxide is selectively etched to expose the silicon surface on which the MOS transistor will be created. This is indicated in the Figure 1.9 (c).

(d) This is followed by covering the surface of substrate with a thin, high-quality oxide layer, which will eventually form the gate oxide of the

MOS transistor as illustrated in Figure 1.9 (d).

(e) On top of the thin oxide, a layer of polysilicon (polycrystalline silicon) is deposited as is shown in the Figure 1.9 (e). Polysilicon is used both as gate electrode material for MOS transistors and also as an interconnect medium in silicon integrated circuits. Undoped polysilicon has relatively high resistivity. The resistivity of polysilicon can be reduced, however, by doping it with impurity atoms.

(f) After deposition, the polysilicon layer is patterned and etched to form the interconnects and the MOS transistor gates. This is shown in Figure 1.9 (f).

(g) The thin gate oxide not covered by polysilicon is also etched along, which exposes the bare silicon surface on which the source and drain junctions are to be formed (Figure 1.9 (g)).

(h) The entire silicon surface is then doped with high concentration of impurities, either through diffusion or ion implantation (in this case with donor atoms to produce n-type doping). Diffusion is achieved by heating the wafer to a high temperature and passing the gas containing desired impurities over the surface. Figure 1.9 (h) shows that the doping penetrates the exposed areas on the silicon surface, ultimately creating two n-type regions (source and drain junctions) in the p-type substrate. The impurity doping also penetrates the polysilicon on the surface, reducing its resistivity.

(i) Once the source and drain regions are completed, the entire surface is again covered with an insulating layer of silicon dioxide, as shown in

Figure 1.9 (i).(*j*) The insulating oxide layer is then patterned in order to provide contact windows for the drain and source junctions, as illustrated in Figure 1.9 (j).

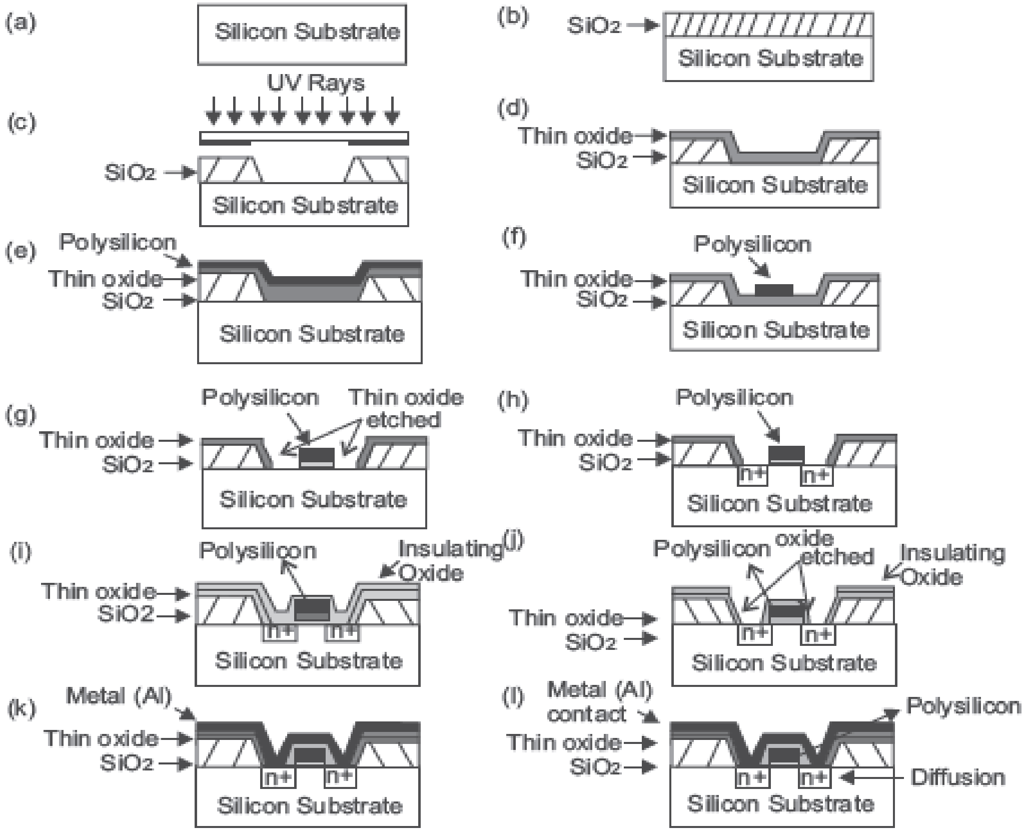


**Figure 1.9:** Fabrication Process of NMOS Device

## CMOS FABRICATION:

CMOS fabrication can be accomplished using either of the three technologies:

- N-well technologies/P-well technologies
- Twin well technology
- Silicon On Insulator (SOI)

The fabrication of CMOS can be done by following the below shown twenty steps, by which CMOS can be obtained by integrating both the NMOS and PMOS transistors on the same chip substrate. For integrating these NMOS and PMOS devices on the same chip, special regions called as wells or tubs are required in which semiconductor type and substrate type are opposite to each other.

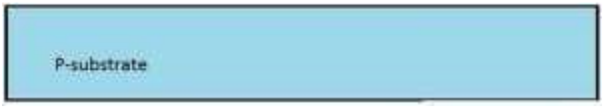
A P-well has to be created on a N-substrate or N-well has to be created on a P-substrate. In this article, the fabrication of CMOS is described using the P-substrate, in which the NMOS transistor is fabricated on a P-type substrate and the PMOS transistor is fabricated in N-well.

The fabrication process involves twenty steps, which are as follows:

## N-Well Process

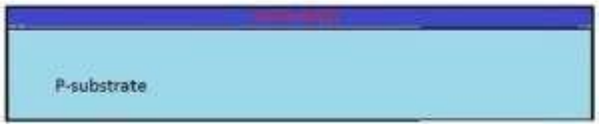
### Step1: Substrate

Primarily, start the process with a P-substrate.



### Step2: Oxidation

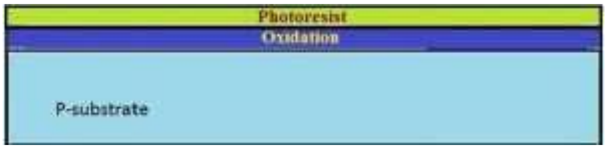
The oxidation process is done by using high-purity oxygen and hydrogen, which are exposed in an oxidation furnace approximately at 1000 degree centigrade.



### Step3: Photoresist

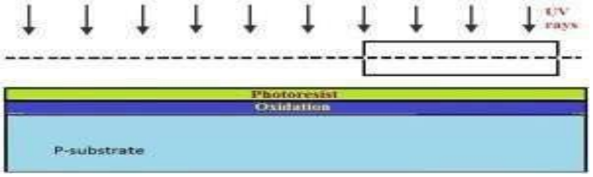
A light-sensitive polymer that softens whenever exposed to light is called as Photoresist layer.

It is formed.



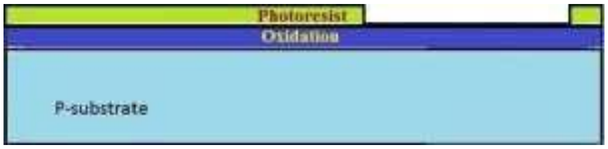
### Step4: Masking

The photoresist is exposed to UV rays through the N-well mask



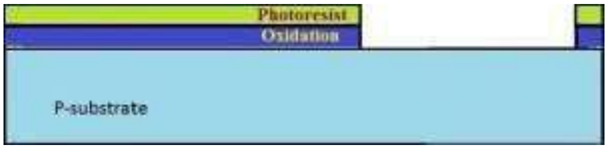
**Ste**

A part of the photoresist layer is removed by treating the wafer with the basic or acidic solutio n.



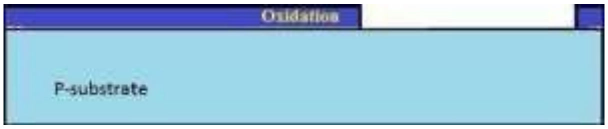
**Step6: Removal of SiO2 using acid etching**

The SiO2 oxidation layer is removed through the open area made by the removal of photoresist using hydrofluoric acid.



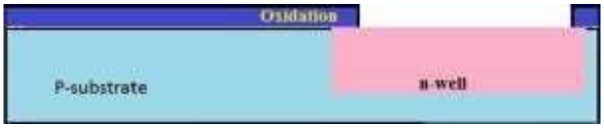
**Step7: Removal of photoresist**

The entire photoresist layer is stripped off, as shown in the below figure.



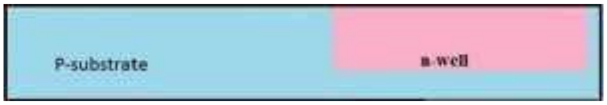
**Step8: Formation of the N-well**

By using ion implantation or diffusion process N-well is formed.



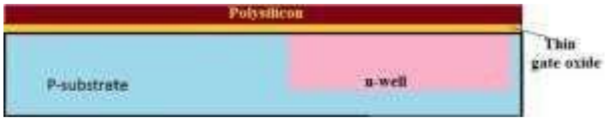
**Step9: Removal of SiO2**

Using the hydrofluoric acid, the remaining SiO2 is removed.



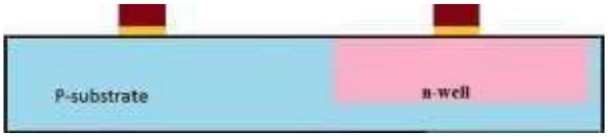
**Step10: Deposition of polysilicon**

Chemical Vapor Deposition (CVD) process is used to deposit a very thin layer of gate oxide.



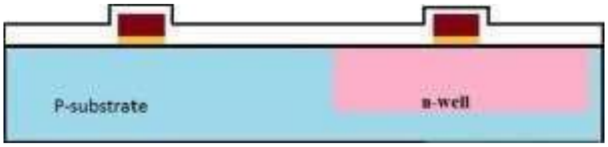
**Step11: Removing the layer barring a small area for the Gates**

Except the two small regions required for forming the Gates of NMOS and PMOS, the remaining layer is stripped off.



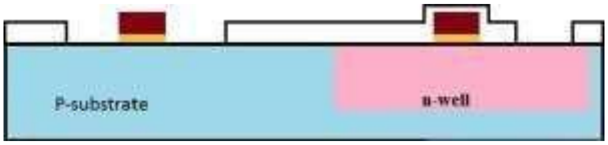
**Step12: Oxidation process**

Next, an oxidation layer is formed on this layer with two small regions for the formation of the gate terminals of NMOS and PMOS.

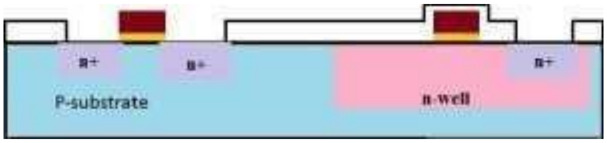


**Step13: Masking and N-diffusion**

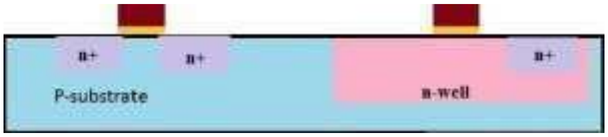
By using the masking process small gaps are made for the purpose of N -diffusion.



The n-type (n+) dopants are diffused or ion implanted, and the three n+ are formed for the formation of the terminals of NMOS.

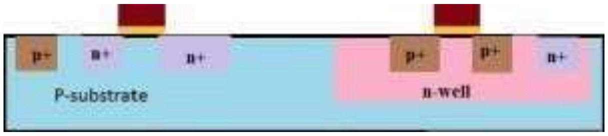


The remaining oxidation layer is stripped off.



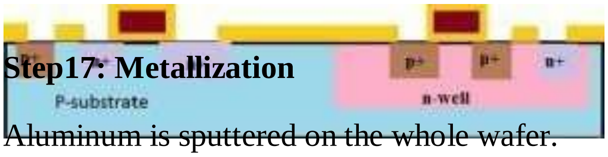
**Step15: P-diffusion**

Similar to the above N-diffusion process, the P-diffusion regions are diffused to form the terminals of the PMOS.



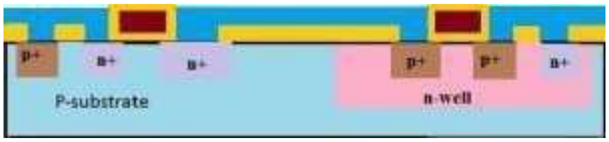
**Step16: Thick field oxide**

A thick-field oxide is formed in all regions except the terminals of the PMOS and NMOS.



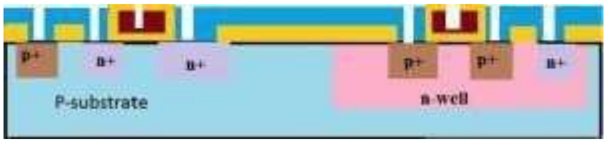
**Step17: Metallization**

Aluminum is sputtered on the whole wafer.



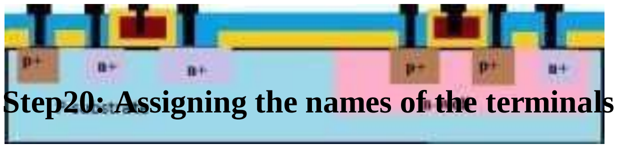
**Step18: Removal of excess metal**

The excess metal is removed from the wafer layer.

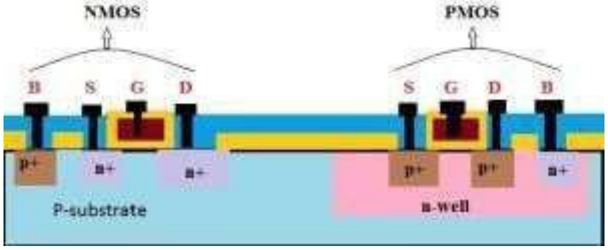


**Step19: Terminals**

The terminals of the PMOS and NMOS are made from respective gaps.



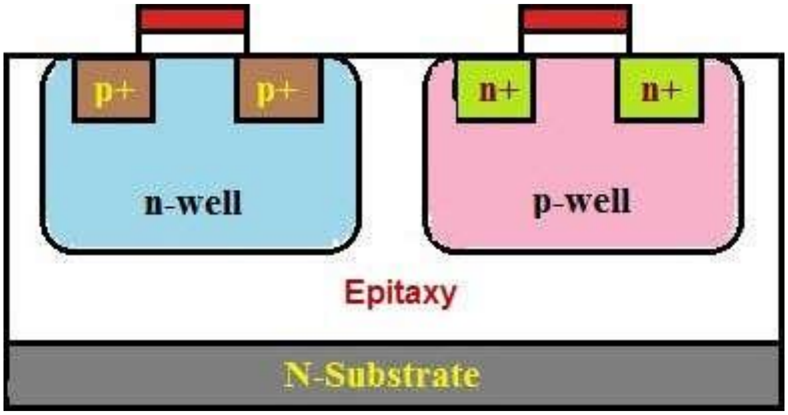
**Step20: Assigning the names of the terminals of the NMOS and PMOS**



## Fabrication of CMOS using P-well process

Among all the fabrication processes of the CMOS, N-well process is mostly used for the fabrication of the CMOS. P-well process is almost similar to the N-well. But the only difference in p-well process is that it consists of a main N-substrate and, thus, P-wells itself acts as substrate for the N-devices.

## Twin tub-CMOS Fabrication Process



In this process, separate optimization of the n-type and p-type transistors will be provided. The independent optimization of Vt, body effect and gain of the P-devices, N-devices can be made possible with this process.

Different steps of the fabrication of the CMOS using the twintub process are as follows:

- Lightly doped n+ or p+ substrate is taken and, to protect the latch up, epitaxial layer is used.
- The high-purity controlled thickness of the layers of silicon are grown with exact dopant concentrations.
- The dopant and its concentration in Silicon are used to determine electrical properties.
- Formation of the tub
- Thin oxide construction

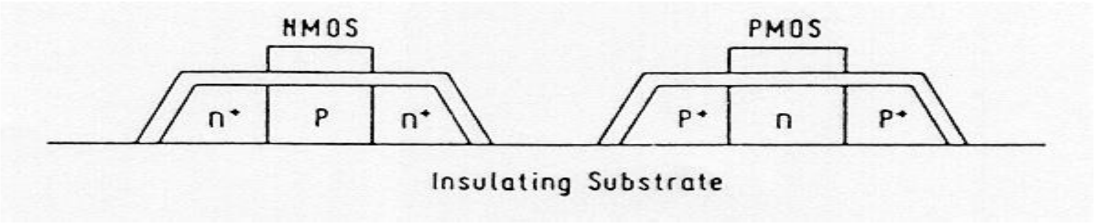
Unit -1 IC Technologies, MOS & Bi CMOS Circuits

- Implantation of the source and drain
- Cuts for making contacts
- Metallization

By using the above steps we can fabricate CMOS using twin tub process method.

**Silicon-on-Insulator (SOI) CMOS Process**

Rather than using silicon as the substrate material, technologists have sought to use an insulating substrate to improve process characteristics such as speed and latch-up susceptibility. The SOI CMOS technology allows the creation of independent, completely isolated nMOS and pMOS transistors virtually side-by-side on an insulating substrate. The main advantages of this technology are the higher integration density (because of the absence of well regions), complete avoidance of the latch-up problem, and lower parasitic capacitances compared to the conventional p & n-well or twin-tub CMOS processes. A cross-section of nMOS and pMOS devicesusing SOI processis shown below.

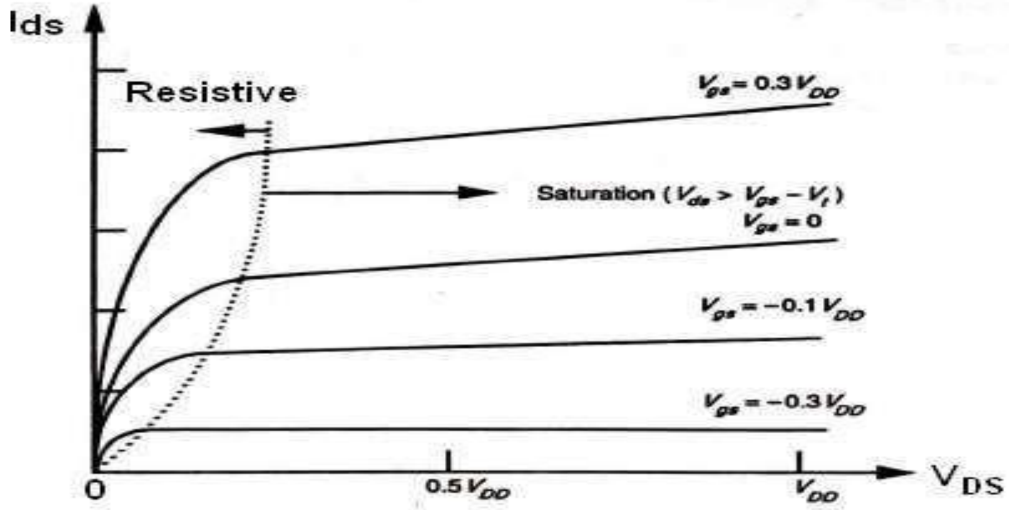


The SOI CMOS process is considerably more costly than the standard p & n-well CMOS process. Yet the improvements of device performance and the absence of latch-up problems can justify its use, especially for deep-sub-micron devices.

## Basic Electrical Properties of MOS and Bi CMOS circuits

### $I_D$-$V_{DS}$ Characteristics of MOS Transistor :

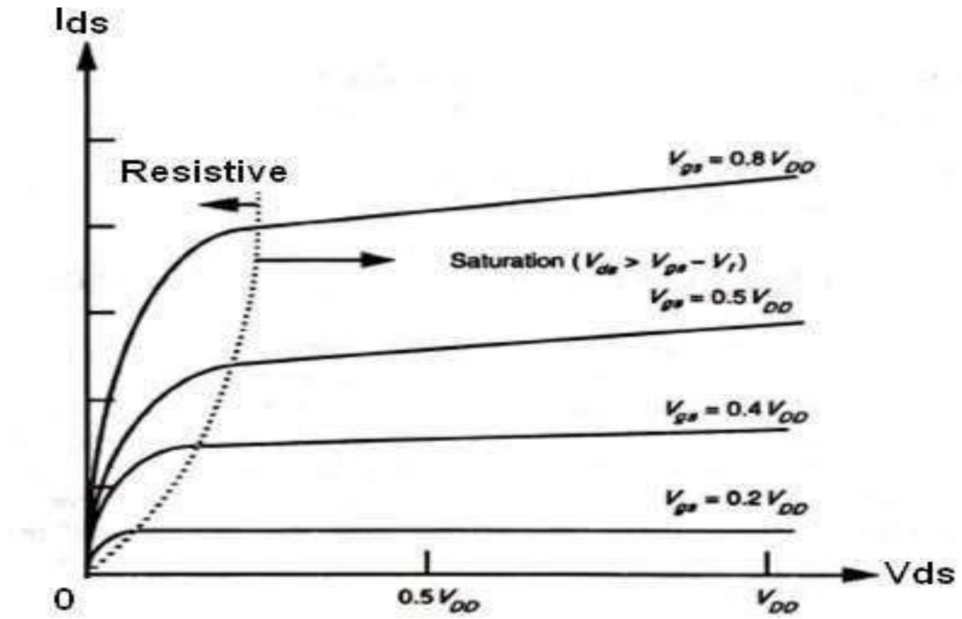
The graph below shows the $I_D$ Vs $V_{DS}$ characteristics of an n- MOS transistor for several values of $V_{GS}$ .It is clear that there are two conduction states when the device is ON. The saturated state and the non-saturated state. The saturated curve is the flat portion and defines the saturation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



(a) Depletion mode device

The boundary of the saturation/non-saturation bias states is a point seen for each curve in the graph as the intersection of the straight line of the saturated region with the quadratic curve of the non-saturated region. This intersection point occurs at the channel pinch off voltage called $V_{DSAT}$. The diamond symbol marks the pinch-off voltage $V_{DSAT}$ for each value of $V_{GS}$. $V_{DSAT}$ is defined as the minimum drain-source voltage that is required to keep the transistor in saturation for a given $V_{GS}$ .In the non-saturated state, the drain current initially increases almost linearly from the origin before bending in a parabolic response. Thus the name ohmic or linear for the non- saturated region.

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



(b). Enhance mode device

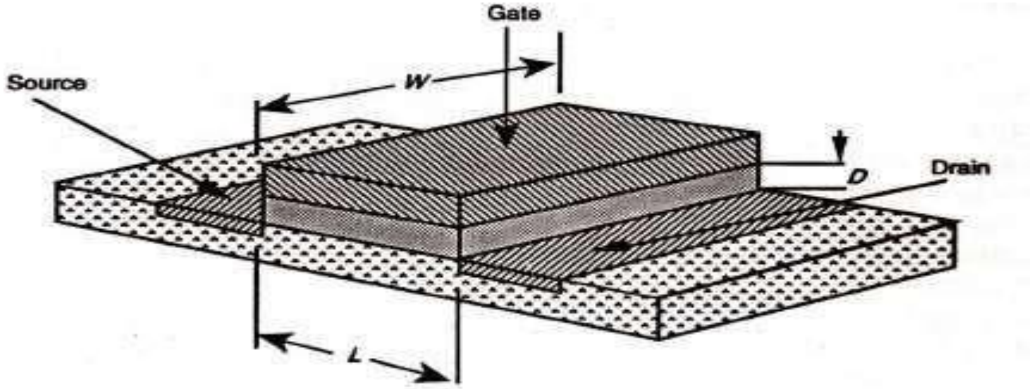
**Drain-to-Source Current $I_{DS}$ Versus Voltage $V_{DS}$ Relationships :**

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

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

Charge induced in channel (Qc) Ids =-Isd = Electron transit time(τ) Length of the channel Where the transit time is given by τsd = ------------------------------

Velocity (v)



**But velocity v= μEds**

Where μ =electron or hole mobility and Eds = Electric field also , Eds = Vds/L

**so,**v = μ.Vds/L and τds = $L^2$ / μ.Vds

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

$$\mu_n \doteq 650 \text{ cm}^2/\text{V sec (surface)}$$
$$\mu_p \doteq 240 \text{ cm}^2/\text{V sec (surface)}$$

**Non-saturated Region :**

Let us consider the Id vs Vd relationships in the non-saturated region .The charge induced in the channel due to due to the voltage difference between the gate and the channel, Vgs (assuming substrate connected to source). The voltage along the channel varies linearly with distance *X* from the source due to the IR drop in the channel .In the non-saturated state the average value is Vds/2. Also the effective gate voltage Vg = Vgs – Vt where Vt, is the threshold voltage needed to invert the charge under the gate and establish the channel.

Hence the induced charge is Qc = Eg εins εoW. L

Where

Eg = average electric field gate to channel

εins = relative permittivity of insulation between gate and channel εo=permittivity

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

Here D is the thickness of the oxide layer. Thus

$$Q_c = \frac{WL\varepsilon_{ins}\varepsilon_0}{D}\left((V_{gs} - V_t) - \frac{V_{ds}}{2}\right)$$

So, by combining the above two equations ,we get

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

or the above equation can be written as

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

In the non-saturated or resistive region where Vds < Vgs – Vt and

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

Generally ,a constant β is defined as

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

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

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

The gate /channel capacitance is

$$C_g = \frac{\varepsilon_{ins}\varepsilon_0 WL}{D}\ \textbf{(parallel plate)}$$

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

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

Some time it is also convenient to use gate –capacitance per unit area ,Cg So,the drain current is

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

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

**Saturated Region**

Saturation begins when Vds = Vgs - V, since at this point the IR drop in the channel equals the effective gate to channel voltage at the drain and we may assume that the current remains fairly constant as Vds increases further. Thus

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

or we can also write that

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

or it can also be written as

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

or

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

The expressions derived above for Ids hold for both enhancement and depletion mode devices. Here the threshold voltage for the nMOS depletion mode device (denoted as Vtd) is negative.

**MOS Transistor Threshold Voltage Vt :**

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

The threshold voltage Vt may be expressed as:

$$V_t = \phi_{ms}\ \frac{Q_B - Q_{SS}}{C_0} + 2\phi_{fN}$$

where QD = the charge per unit area in the depletion layer below the oxide Qss = charge density at Si: SiO2 interface

Co =Capacitance per unit area.

Φns = work function difference between gate and Si

ΦfN = Fermi level potential between inverted surface and bulk Si

For polynomial gate and silicon substrate, the value of Φns is negative but negligible and the magnitude and sign of Vt are thus determined by balancing the other terms in the equation. To evaluate the Vt the other terms are determined as below.

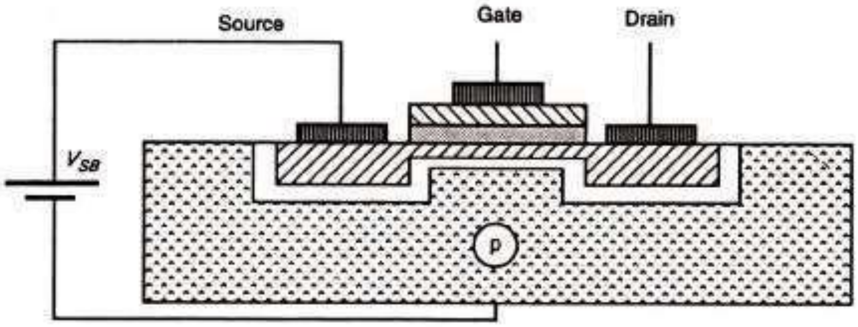
$$Q_B = \sqrt{2\varepsilon_0\varepsilon_{Si}qN(2\phi_{fN} + V_{SB})}\ \text{coulomb/m}^2$$

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

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

**Body Effect :**

Generally while studying the MOS transistors it is treated as a three terminal device. But, the body of the transistor is also an implicit terminal which helps to understand the characteristics of the transistor. Considering the body of the MOS transistor as a terminal is known as the body effect. The potential difference between the source and the body (Vsb) affects the threshold voltage of the transistor. In many situations, this Body Effect is relatively insignificant, so we can (unless **otherwise** stated) ignore the Body Effect. But it is not always insignificant, in some cases it can have a tremendous impact on MOSFET circuit performance.



**Body effect - nMOS device**

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

The threshold voltage can be written as

$$V_t = V_t(0) + \left(\frac{D}{\varepsilon_{ins}\varepsilon_0}\right)\sqrt{2\varepsilon_0\varepsilon_{Si}QN}.\,(V_{SB})^{1/2}$$

Where Vt(0) is the threshold voltage for V$_{sd}$ = 0

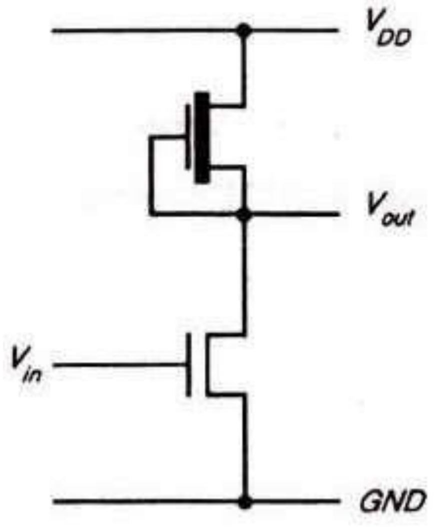
For n-MOS depletion mode transistors ,the body voltage values at different V$_{DD}$ voltages are given below.

$V_{SB}$ = 0 V ; Vsd = -0.7$V_{DD}$ (= - 3.5 V for $V_{DD}$ =+5V ) $V_{SB}$ = 5 V ; Vsd = -0.6$V_{DD}$ (= - 3.0 V for $V_{DD}$ =+5V )

**nMOS INVERTER :**

An inverter circuit is a very important circuit for producing a complete range of logic circuits. This is needed for restoring logic levels, for Nand and Nor gates, and for sequential and memory circuits of various forms .A simple inverter circuit can be constructed using a transistor with source connected to ground and a load resistor of connected from the drain to the positive supply rail $V_{DD}$· The output is taken from the drain and the input applied between gate and ground .

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



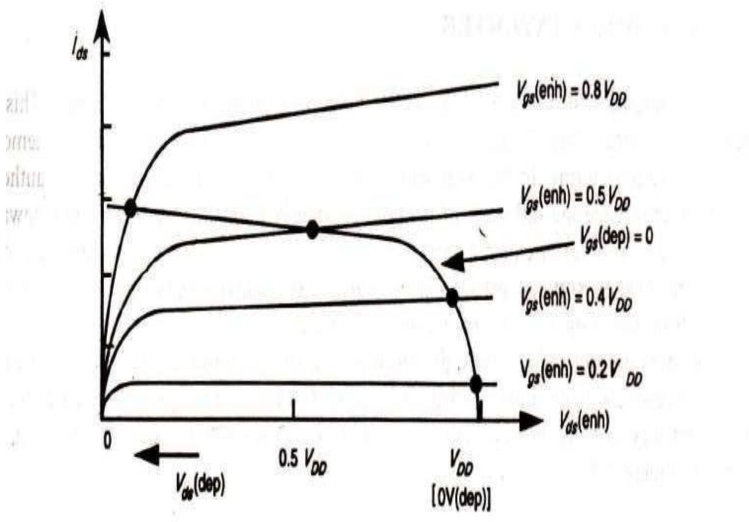
The salient features of the n-MOS inverter are

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

**nMOS Inverter transfer characteristic**.

The transfer characteristic is drawn by taking Vds on x-axis and Ids on Y-axis for both enhancement and depletion mode transistors. So,to obtain the inverter transfer characteristic for

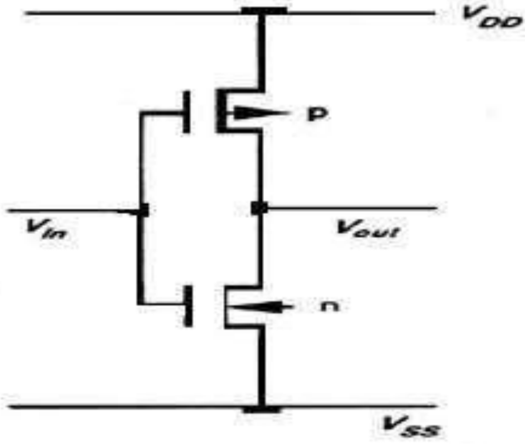
*Vgs* = 0 depletion mode characteristic curve is superimposed on the family of curves for the enhancement mode device and from the graph it can be seen that , maximum voltage across the enhancement mode device corresponds to minimum voltage across the depletion mode transistor.



From the graph it is clear that as Vin(=Vgs p.d. transistor) exceeds the Pulldown threshold voltage current begins to flow. The output voltage Vout thus decreases and the subsequent increases in Vin will cause the Pull down transistor to come out of saturation and become resistive.

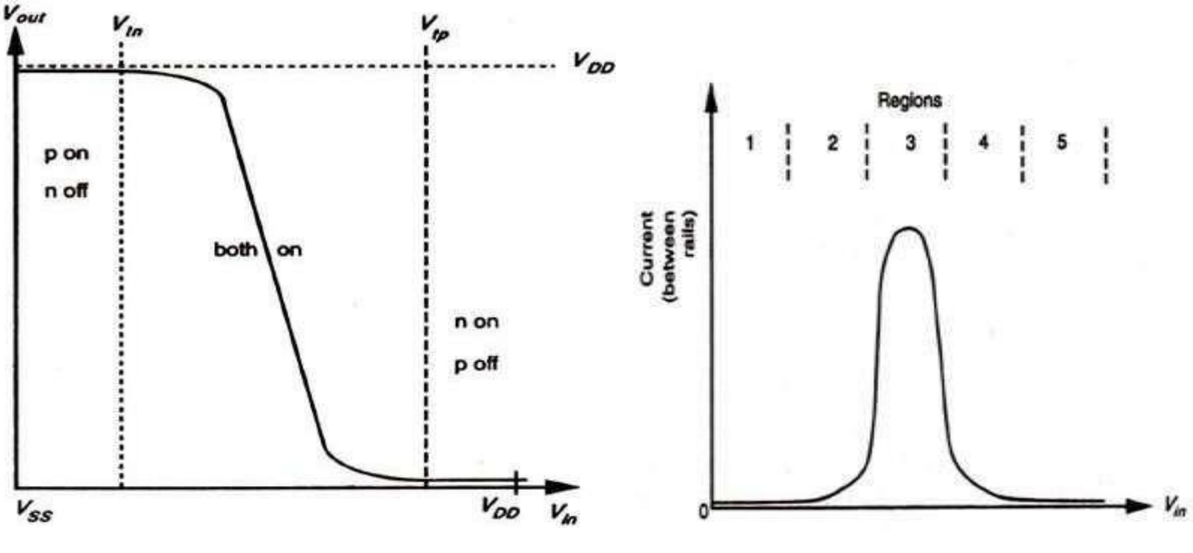
**CMOS Inverter:**

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



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

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



Hence the output voltage is logic 1 . In region 5 , Vin = logic 1 and the n-transistor is fully on while the p-transistor is fully off. So, no current flows and logic 0 appears at the output.

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

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

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

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

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

where

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

and

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

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

**Determination of Pull-up to Pull –Down Ratio *(Zp.u}Zp.d.)*for an nMOS Inverter driven by another nMOS Inverter :**

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

$$V_{in} = V_{out} = V_{inv}$$

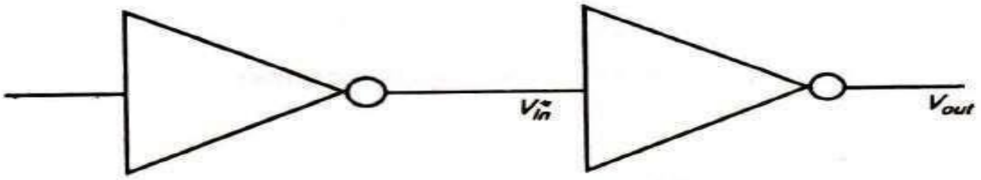


Fig.(a).Inverter driven by another inverter.

For equal margins around the inverter threshold, we set Vinv = 0.5VDD · At this point both transistors are in saturation and we can write that

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

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

**Equating (since currents are the same) we have**

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

where Wp.d , Lp.d , Wp.u. and Lp.u are the widths and lengths of the pull-down and pull-up transistors respectively.

So,we can write that

**whence** $\quad {}_{p.d} \quad L_{p.u.}$

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

The typical, values for Vt ,Vinv and Vtd are

$$V_t = 0.2V_{DD};\ V_{td} = -\ 0.6V_{DD}$$
$$V_{inv} = 0.5V_{DD}\ \text{(for equal margins)}$$

Substituting these values in the above equation ,we get

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

Here

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

So,we get

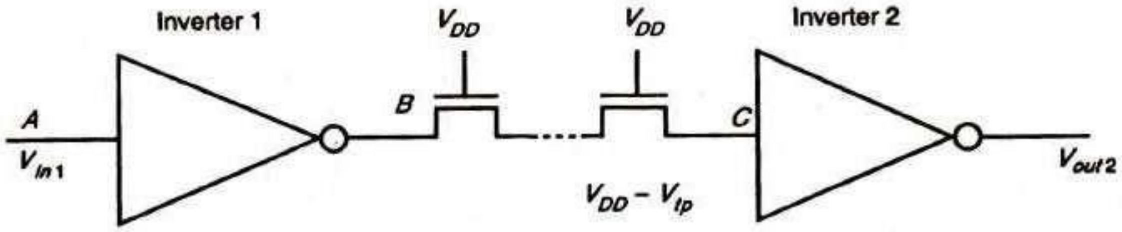
$$\boxed{Z_{p.u.}/Z_{p.d.} = 4/1}$$

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

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

Let us consider an arrangement in which the input to inverter 2 comes from the output of inverter 1

but passes through one or more nMOS transistors as shown in Fig. below (These transistors are called pass transistors).

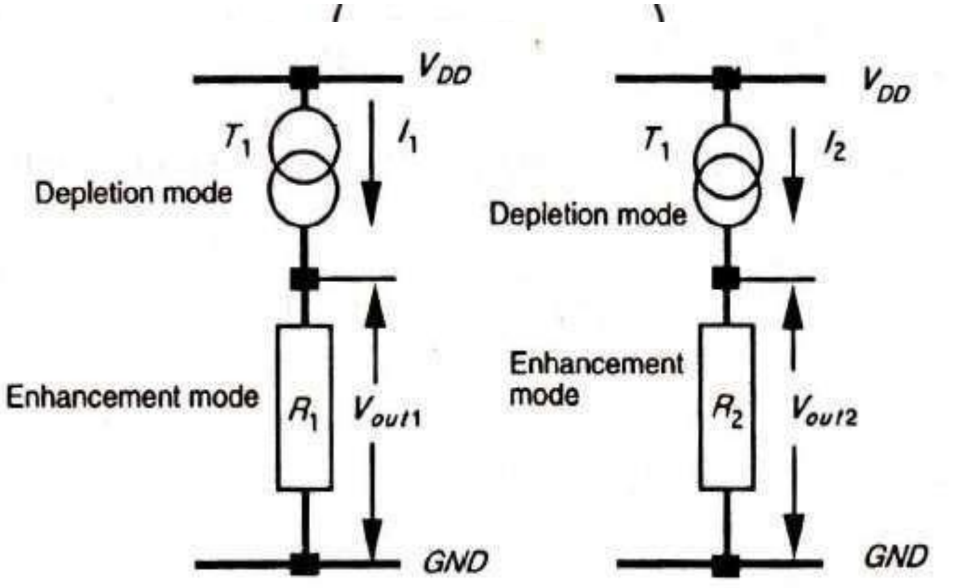


The connection of pass transistors in series will degrade the logic 1 level / into inverter 2 so that the output will not be a proper logic 0 level. The critical condition is , when point A is at 0 volts and B is thus at VDD. but the voltage into inverter 2at point C is now reduced from VDD by the threshold voltage of the series pass transistor. With all pass transistor gates connected to VDD there is a loss of Vtp, however many are connected in series, since no static current flows through them and there can be no voltage drop in the channels. Therefore, the input voltage to inverter 2 is

Vin2 = VDD- Vtp where Vtp = threshold voltage for a pass transistor.

Let us consider the inverter 1 shown in Fig.(a) with input = VDD· If the input is at VDD , then the pull-down transistor T2 is conducting but with a low voltage across it; therefore, it is in its resistive region represented by R1 in Fig.(a) below. Meanwhile, the pull up transistor T1 is in saturation and is represented as a current source.

For the pull down transistor



(a) Inverter 1 with input = $V_{DD}$ (b) Inverter 2 with input = $V_{DD} - V_{tp}$

So,

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

Now, for depletion mode pull-up transistor in saturation with Vgs = 0

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

The product 1R1 = Vout1So,

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

Whence,

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

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

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

$I_1R_1=I_2R_2$ , therefore

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

Considering the typical values

$$V_t = 0.2 V_{DD}$$

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

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

Therefore

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

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

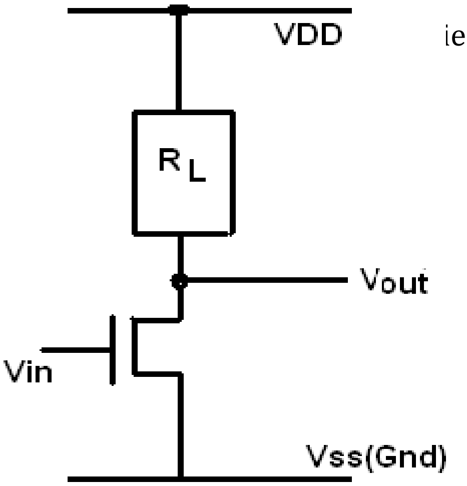
(i). An inverter driven directly from the output of another should have a $Z_{p.u}/ Z_{pd}$. ratio of ≥ 4/1.

(ii).An inverter driven through one or more pass transistors should have a $Z_{p.u.}/Z_{p.d}$ ratio of ≥8/1

**ALTERMTIVE FORMS OF PULL –UP**

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

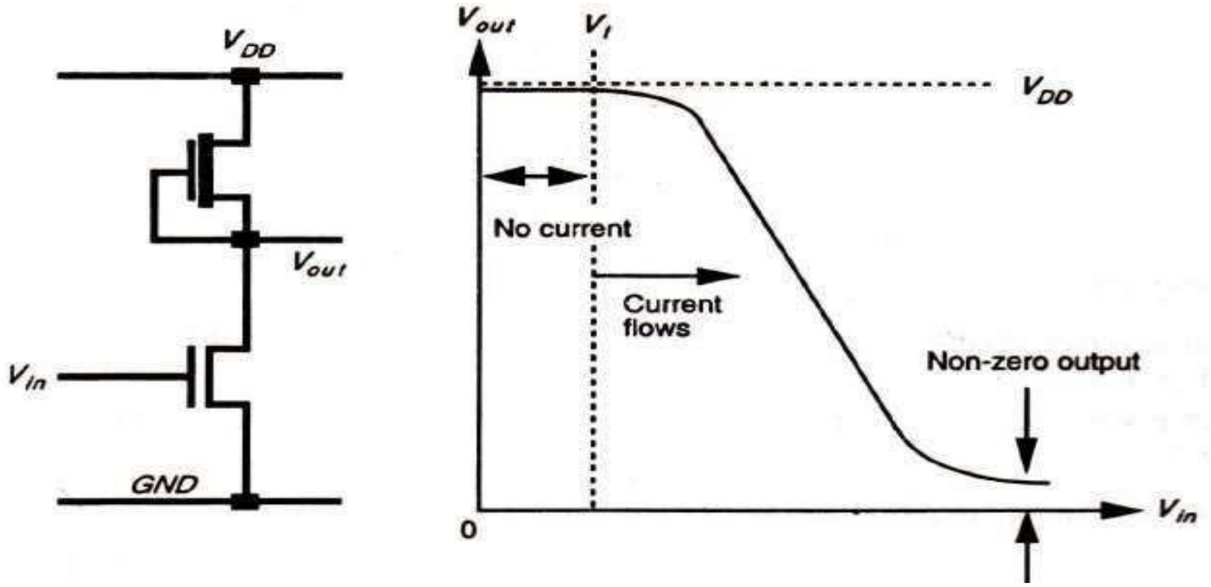
**(i).Load resistance RL :** This arrangement consists of a load resistor as apull-up as shown in the diagram below.But it is not widely used because of the large space requirements of resistors produced in a silicon substrate.



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

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

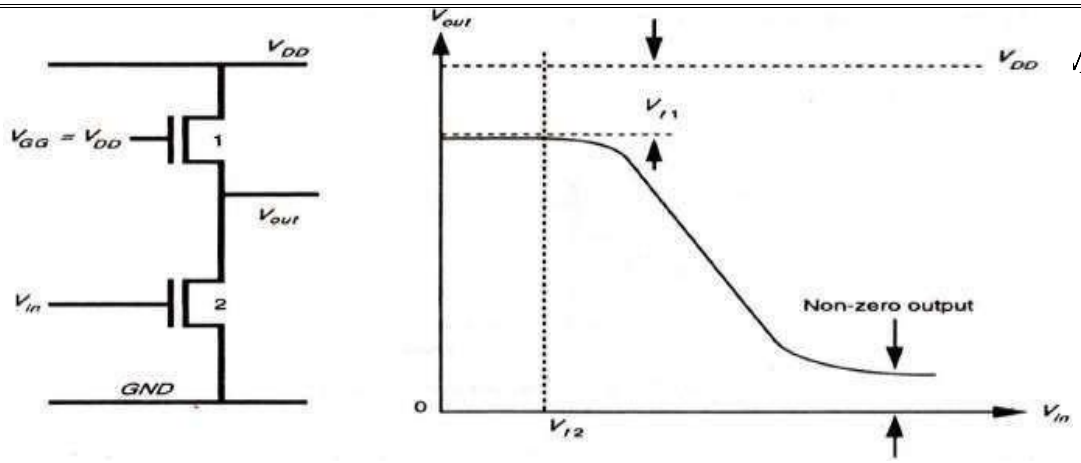
(b) Switching of output from 1 to 0 begins when Vin exceeds Vt, of pull-down device.



**nMOS depletion mode transistor pull-up and transfer characteristic**

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

(ii) **nMOS enhancement mode pull-up :**This arrangement consists of a n-MOS enhancement mode transistor as pull-up. The arrangement and the transfer characteristic are shown below.



**nMOS enhancement mode pull-up and transfer characteristic**

The important features of this arrangement are

*(a)* Dissipation is high since current flows when Vin =logical 1 (VGG is returned to VDD).

*(b)* Vout can never reach VDD (logical I) if VGG = VDD as is normally the case.

*(c)* VGG may be derived from a switching source, for example, one phase of a clock, so that dissipation can be greatly reduced.

*(d)* If VGG is higher than VDD then an extra supply rail is required.

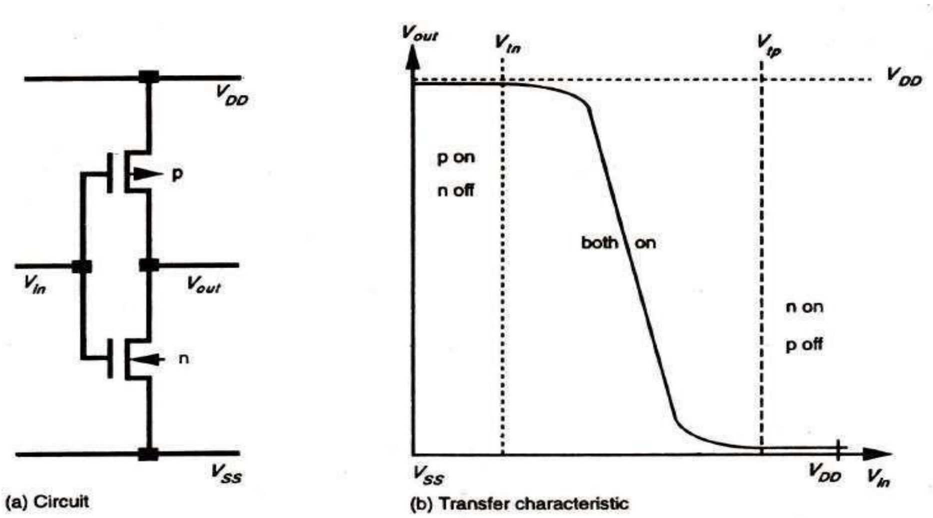
(iii) **Complementary transistor pull-up (CMOS) :** This arrangement consists of a C-MOS arrangement as pull-up. The arrangement and the transfer characteristic are shown below

The salient features of this arrangement are

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

(b) Full logical 1 and 0 levels are presented at the output.

(c) For devices of similar dimensions the p-channel is slower than the n-channel device.

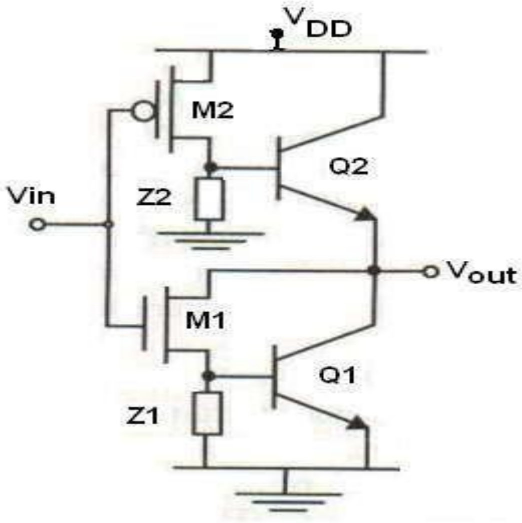


(a) Circuit

(b) Transfer characteristic

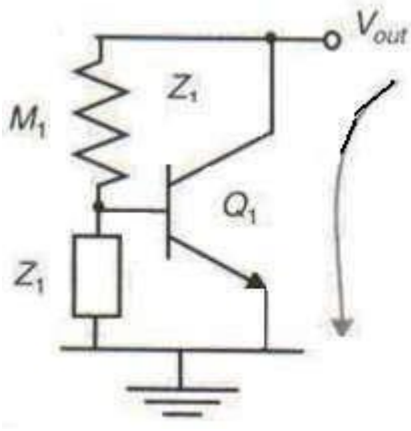
**BiCMOS INVERTER:**

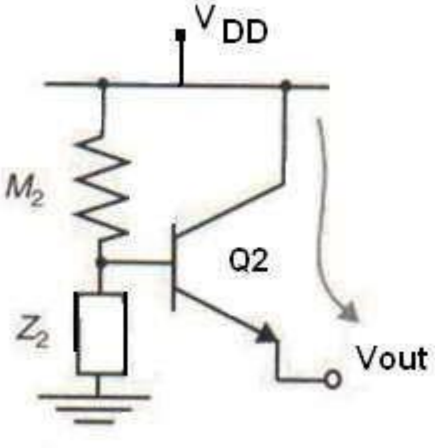
A BiCMOS inverter, consists of a PMOS and NMOS transistor ( M2 and M1), two NPN bipolar junction transistors,( Q2 and Q1), and two impedances which act as loads( *Z*2 and *Z*1) as shown in the circuit below.



When input, Vin, is high (VDD), the NMOS transistor ( M1), turns on, causing Q1 to conduct,while M2 and Q2 are off, as shown in figure (b) . Hence , a low (GND) voltage is translated to the output Vout. On the other hand, when the input is low, the M2 and Q2 turns on, while M1and Q1 turns off, resulting to a high output level at the output as shown in Fig.(b).

In steady-state operation, Q1 and Q2 never turns on or off simultaneously, resulting to a lower power consumption. This leads to a push-pull bipolar output stage. Transistors M1and M2, on the other hand, works as a phase-splitter, which results to a higher input impedance.





The impedances Z2 and *Z*1 are used to bias the base-emitter junction of the bipolar transistor and to ensure that base charge is removed when the transistors turn off. For example when the input voltage makes a high-to-low transition, M1 turns off first. To turn off Q1, the base charge must be removed, which can be achieved by *Z*1.With this effect, transition time reduces. However, there exists a short time when both Q1 and Q2 are on, making a direct path from the supply (VDD) to the ground. This results to a current spike that is large and has a detrimental effect on both the noise and power consumption, which makes the turning off of the bipolar transistor fast .

**Comparison of BiCMOS and C-MOS technologies**

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

When comparing BiCMOS and CMOS in driving small capacitive loads, their performance are comparable, however, making BiCMOS consume more power than CMOS. On the other hand, driving larger capacitive loads makes BiCMOS in the advantage of consuming less power than CMOS, because the construction of CMOS inverter chains are needed to drive large capacitance loads, which is not needed in BiCMOS.

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

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

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

# UNIT II

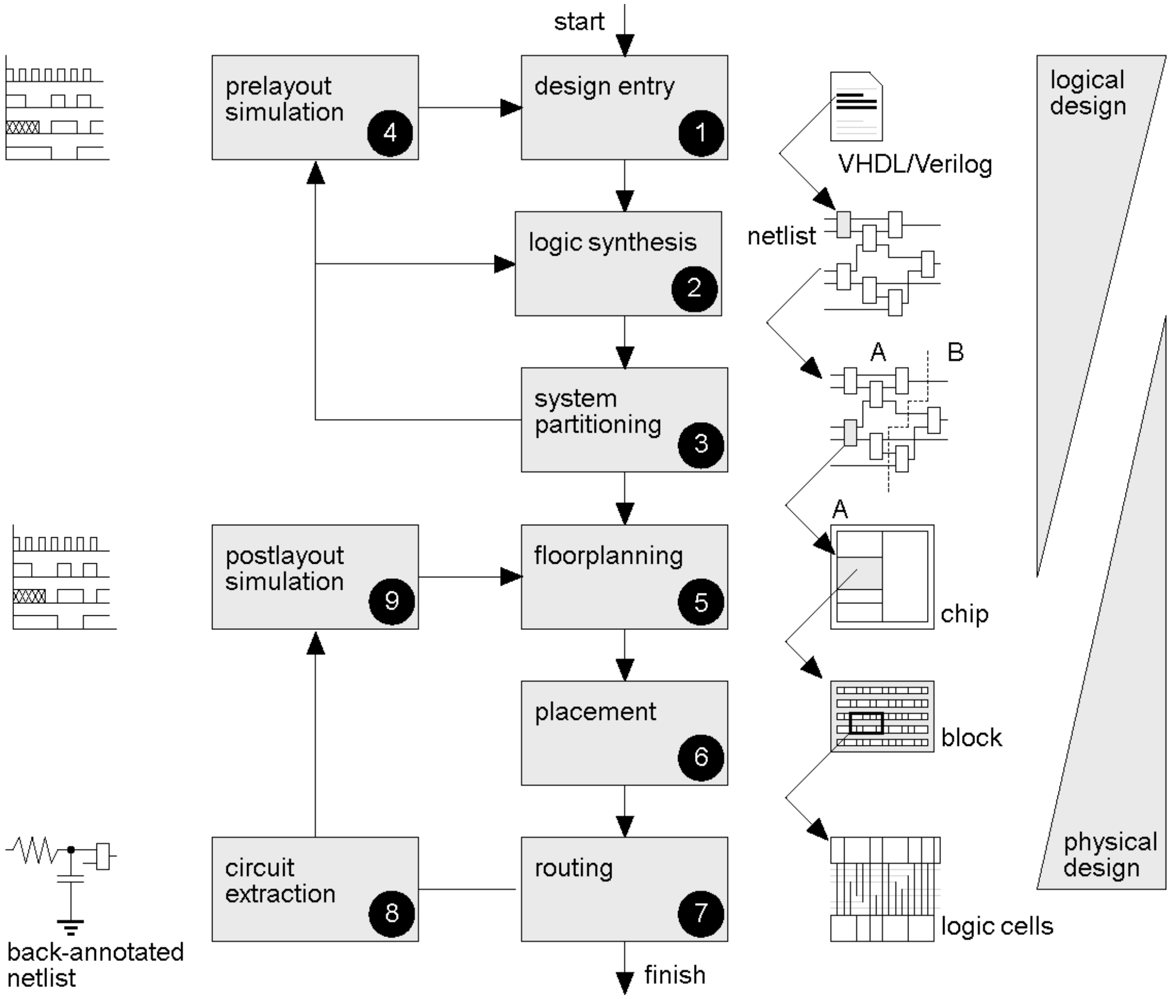
## VLSI Circuit Design Processes

- **VLSI Design Flow**
- **MOS Layers**
- **Stick Diagrams**
- **Design Rules and Layout**
- **Lamda (λ) based design rules for wires, contacts and Transistors**
- **Layout Diagrams for NMOS and CMOS Inverters and Gates**
- **Scaling of MOS circuits**

## VLSI DESIGN FLOW

A design flow is a sequence of operations that transform the IC designers' intention (usually represented in RTL format) into layout GDSII data.

A well-tuned design flow can help designers go through the chip-creation process relatively smoothly and with a decent chance of error-free implementation. And, a skilful IC implementation engineer can use the design flow creatively to shorten the design cycle, resulting in a higher likelihood that the product will catch the market window.



## Front-end design (Logical design):

**1. Design entry** – Enter the design in to an ASIC design system using a hardware description language (HDL) or schematic entry

**2. Logic synthesis** – Generation of net list (logic cells and their connections) from HDL code. Logic synthesis consists of following steps: (i) Technology independent Logic optimization (ii) Translation: Converting Behavioral description to structural domain (iii) Technology mapping or Library binding

**3. System partitioning** - Divide a large system into ASIC-sized pieces

**4. Pre-layout simulation** - Check to see if the design functions correctly. Gate level functionality and timing details can be verified.

## Back-end design (Physical design):

**5. Floor planning** - Arrange the blocks of the netlist on the chip

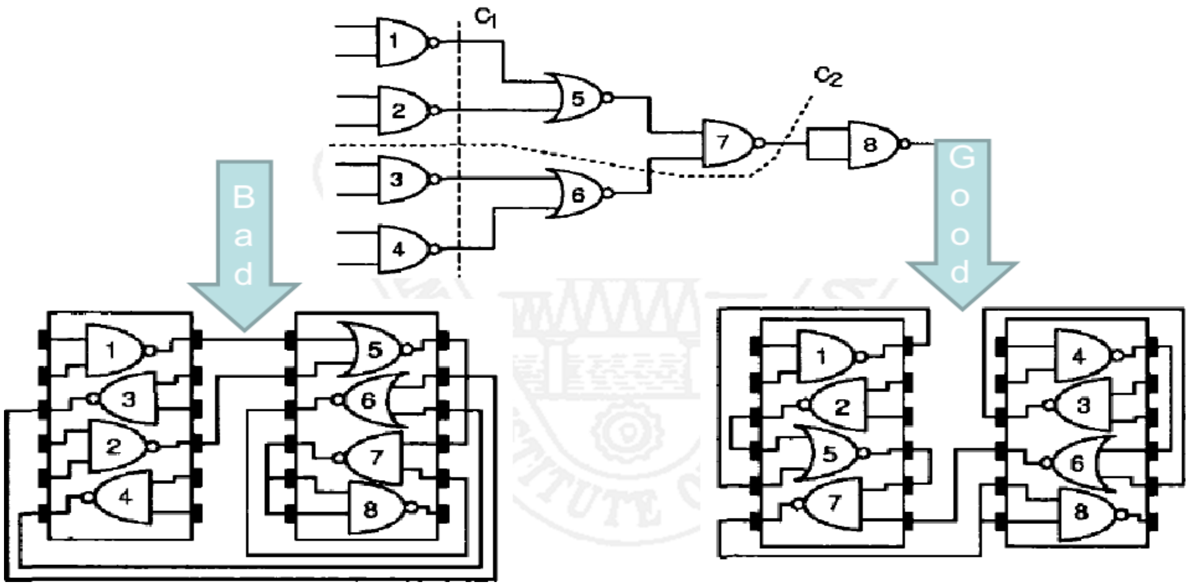
**6. Placement** - Decide the locations of cells in a block

**7. Routing** - Make the connections between cells and blocks

**8. Circuit Extraction** - Determine the resistance and capacitance of the interconnect

**9. Post-layout simulation** - Check to see the design still works with the added loads of the interconnect

## Partitioning



## MOS LAYERS

MOS design is aimed at turning a specification into masks for processing silicon to meet the specification. We have seen that MOS circuits are formed on four basic layers

- N-diffusion
- P-diffusion
- Poly Si
- Metal

which are isolated from one another by thick or thin (thinox) silicon silicon dioxide insulating layers. The thin oxide (thinox) mask region includes n-diffusion, p-diffusion, and transistor channels. Polysilicon and thinox regions interact so that a transistor is formed where they cross one another.

## STICK DIAGRAMS

A stick diagram is a diagrammatic representation of a chip layout that helps to abstract a model for design of full layout from traditional transistor schematic. Stick diagrams are used to convey the layer information with the help of a color code.

***"A stick diagram is a cartoon of a layout."***

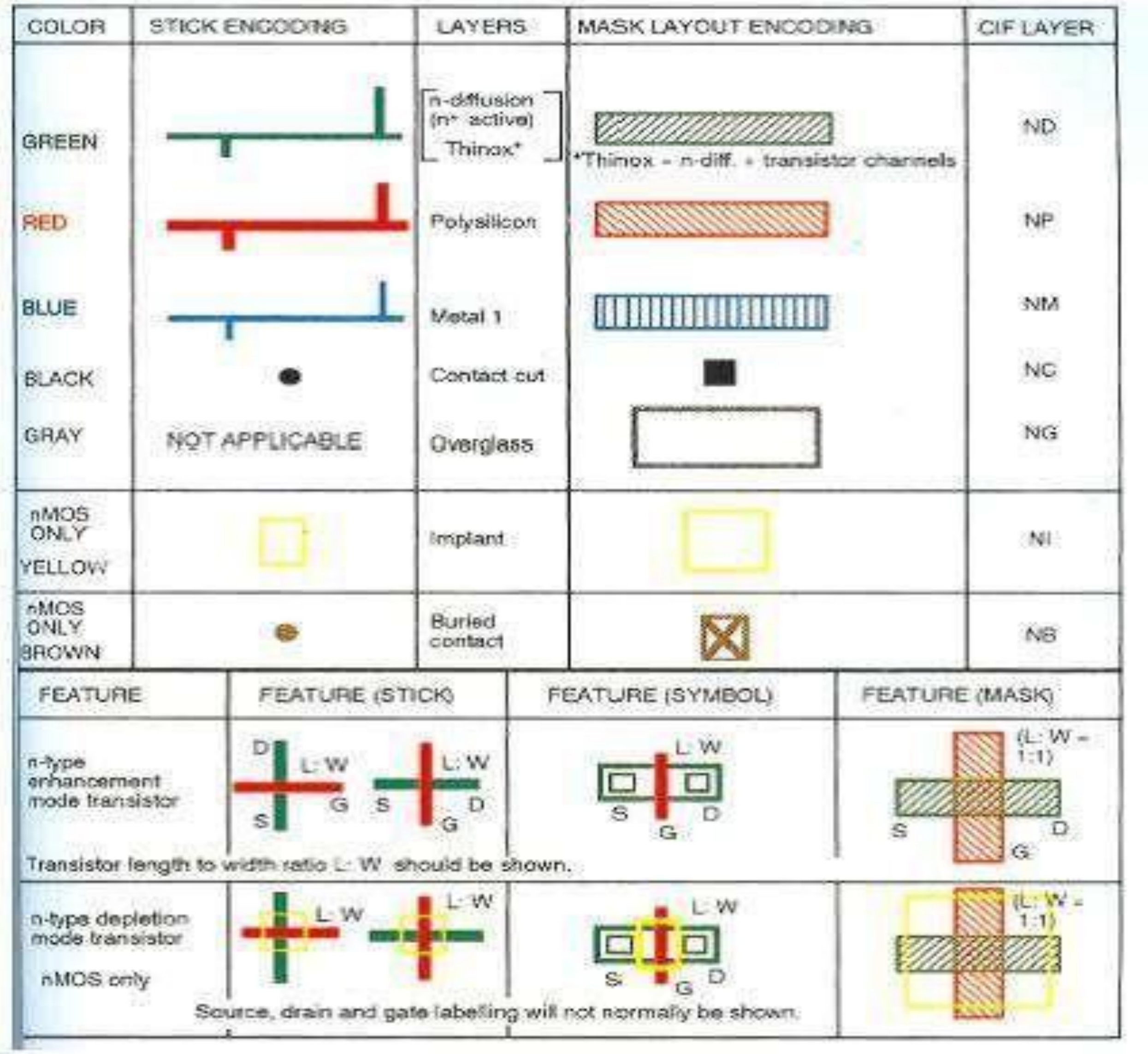
The designer draws a freehand sketch of a layout, using colored lines to represent the various process layers such as diffusion, metal and polysilicon. Where polysilicon crosses diffusion, transistors are created and where metal wires join diffusion or polysilicon, contacts are formed.

For example, in the case of nMOS design,

- Green color is used for n-diffusion
- Red for polysilicon
- Blue for metal
- Yellow for implant, and black for contact areas.

Monochrome encoding is also used in stick diagrams to represent the layer information.

## Stick Diagrams –NMOS Encoding

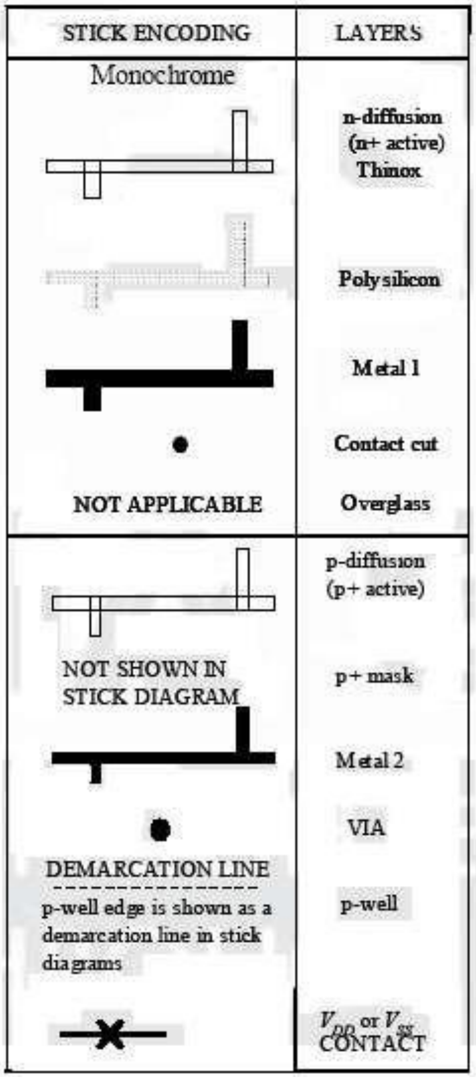


| COLOR | STICK ENCODING | LAYERS | MASK LAYOUT ENCODING | CIF LAYER |
|---|---|---|---|---|
| GREEN | 🟩 | n-diffusion (n+ active)<br>Thinox* | ▨<br>*Thinox = n-diff. + transistor channels | ND |
| RED | 🟥 | Polysilicon | ▨ | NP |
| BLUE | 🟦 | Metal 1 | ▥ | NM |
| BLACK | ● | Contact cut | ■ | NC |
|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

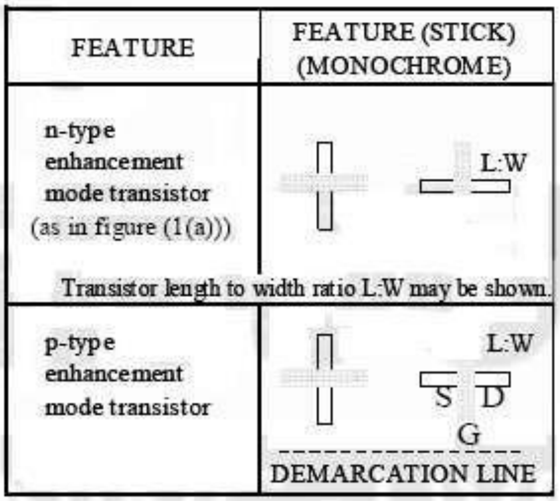
| FEATURE | FEATURE (STICK) | FEATURE (SYMBOL) | FEATURE (MASK) |
|---|---|---|---|
| n-type enhancement mode transistor | D L: W G S<br>L: W S D G | L: W S G D | (L: W = 1:1) S D G |
| Transistor length to width ratio L: W should be shown. | | | |
| n-type depletion mode transistor<br>nMOS only | L: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

NMOS ENCODING

## CMOS ENCODING



| STICK ENCODING | LAYERS |
|---|---|
| Monochrome | |
| | n-diffusion<br>(n+ active)<br>Thinox |
| | Polysilicon |
| | Metal 1 |
| ● | Contact cut |
| NOT APPLICABLE | Overglass |
| | p-diffusion<br>(p+ active) |
| NOT SHOWN IN STICK DIAGRAM | p+ mask |
| | Metal 2 |
| ● | VIA |
| DEMARCATION LINE<br>- - - - - - - - - - - - - - - -<br>p-well edge is shown as a demarcation line in stick diagrams | p-well |
| ✕ | $V_{DD}$ or $V_{SS}$ CONTACT |

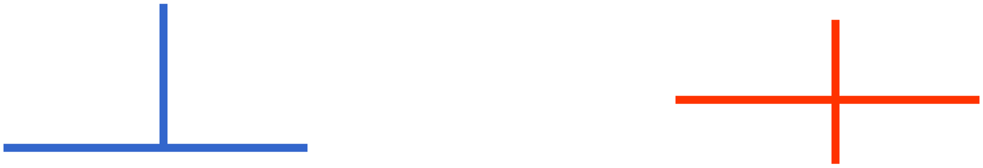


| FEATURE | FEATURE (STICK) (MONOCHROME) |
|---|---|
| n-type enhancement mode transistor (as in figure (1(a))) | L:W |
| Transistor length to width ratio L:W may be shown. | |
| p-type enhancement mode transistor | L:W<br>S D<br>G<br>- - - - - - - - - - - - - - - -<br>DEMARCATION LINE |

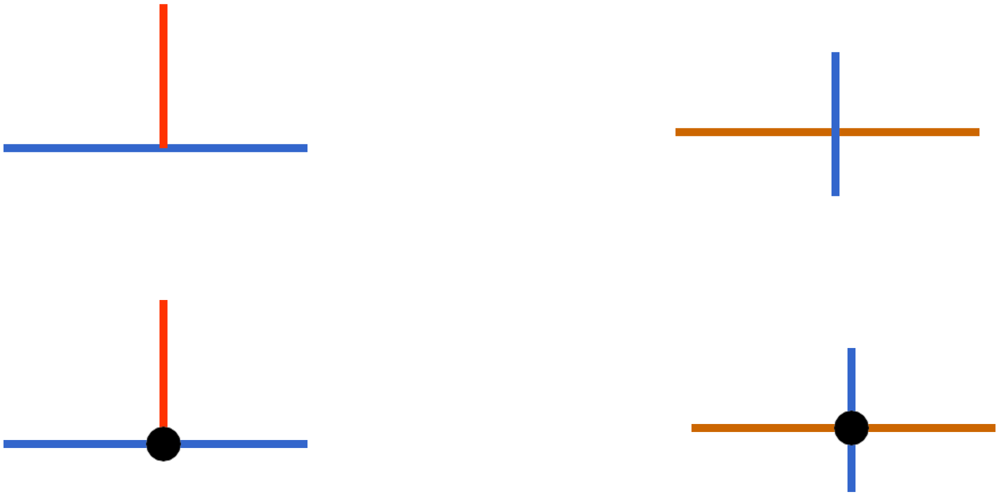
## Stick Diagrams – Some Rules

**Rule 1:**

When two or more 'sticks' of the same type cross or touch each other that represents electrical contact.

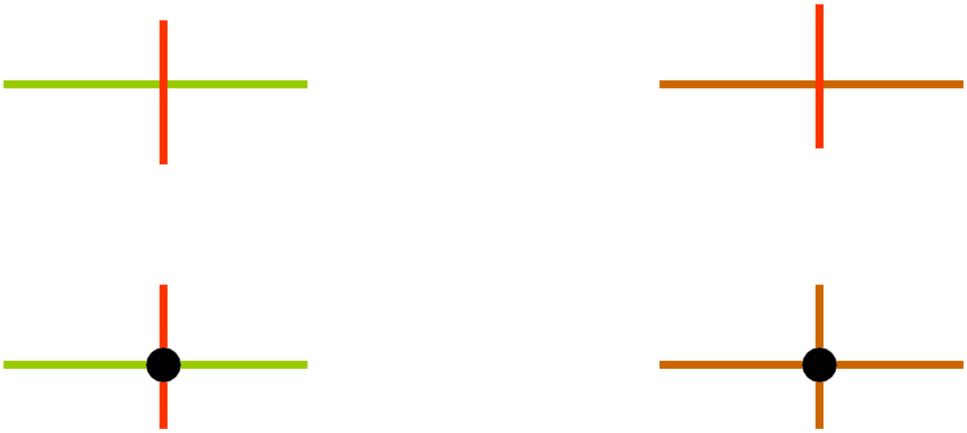


**Rule 2:**
When two or more "sticks" of different type cross or touch each other there is no electrical contact. (If electrical contact is needed we have to show the connection explicitly)



**Rule 3:**

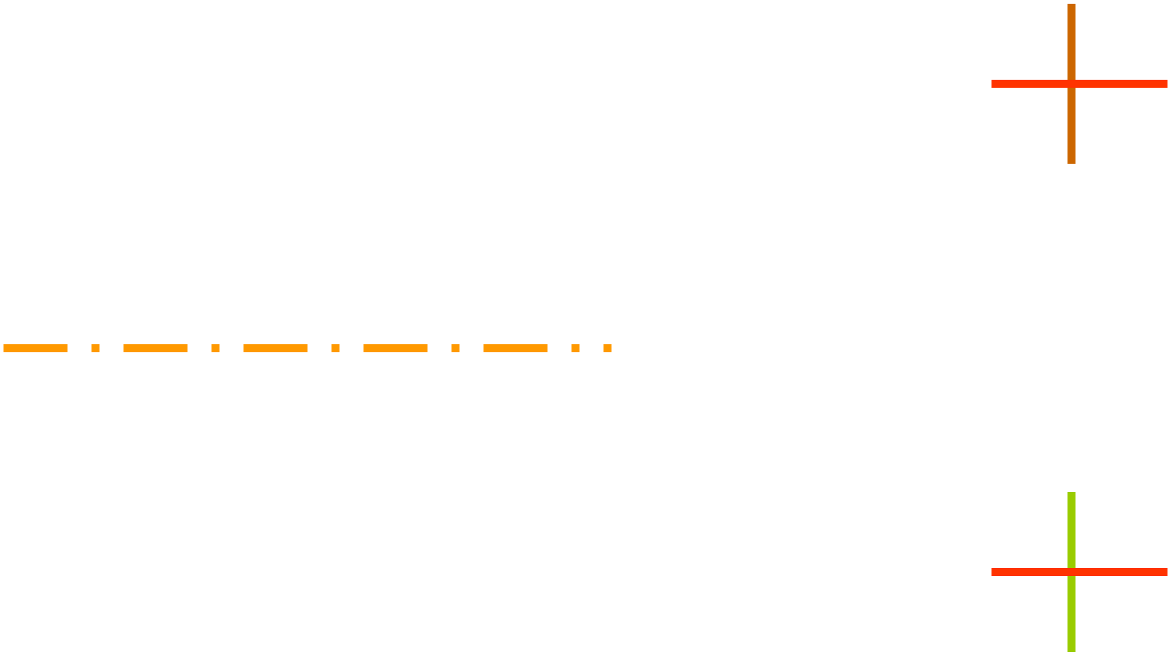
When a poly crosses diffusion it represents a transistor.



Note: If a contact is shown then it is ***not*** a transistor.

**Rule 4:**

In CMOS a demarcation line is drawn to avoid touching of p-diff with n-diff. All PMOS must lie on one side of the line and all NMOS will have to be on the other side.



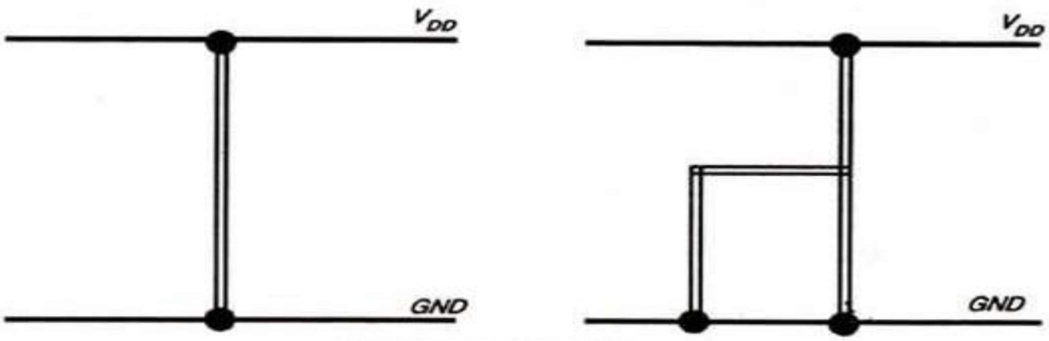
## nMOS Design Style :

To understand the design rules for nMOS design style , let us consider a single metal, single polysilicon nMOS technology.

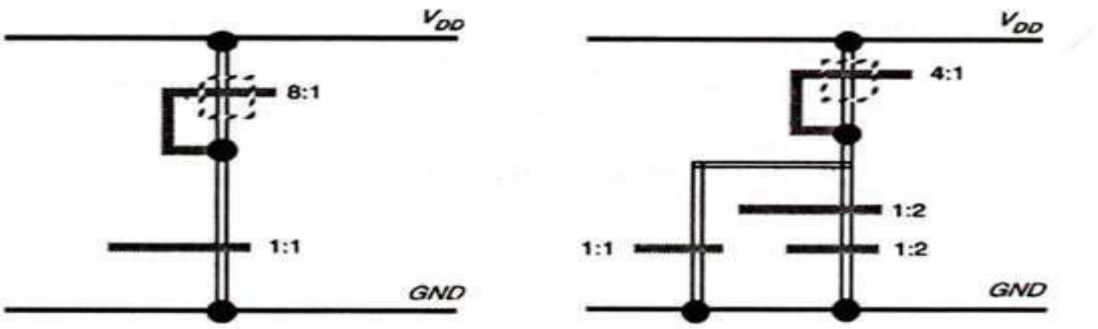
The layout of nMOS is based on the following important features.

- ✓ n-diffusion [n-diff.] and other thin oxide regions [thinox] (green) ;
- ✓ polysilicon 1 [poly.]-since there is only one polysilicon layer here (red);
- ✓ metal 1 [metal]-since we use only one metal layer here (blue);
- ✓ implant (yellow);
- ✓ contacts (black or brown [buried]).

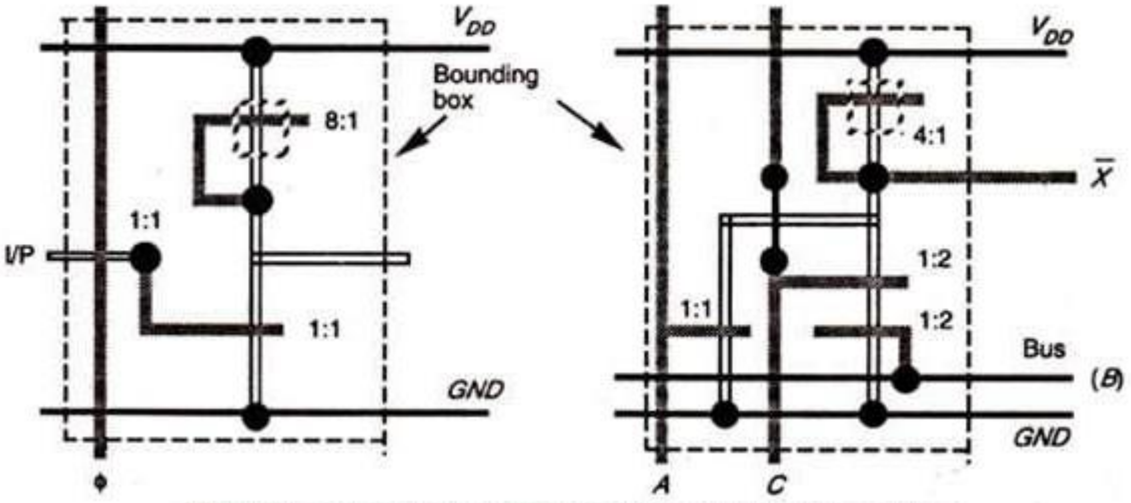
A transistor is formed wherever poly. crosses n-diff. (red over green) and all diffusion wires (interconnections) are n-type (green).When starting a layout, the first step normally taken is to draw the metal (blue) VDD and GND rails in parallel allowing enough space between them for the other circuit elements which will be required. Next, thinox (green) paths may be drawn between the rails for inverters and inverter based logic as shown in Fig. below. Inverters and inverter- based logic comprise a pull-up structure, usually a depletion mode transistor, connected from the output point to VDD and a pull down structure of enhancement mode transistors suitably interconnected between the output point and GND. This is illustrated in the Fig.(b). remembering that poly. (red) crosses thinox (green)wherever transistors are required. One should consider the implants (yellow) for depletion mode transistors and also consider the length to width (L:W) ratio for each transistor. These ratios are important particularly in nMOS and nMOS- like circuits.



(a) Rails and thinox paths



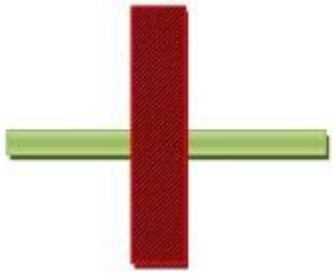
(b) Pull-up and pull-down structures (polysilicon), implants, and ratios



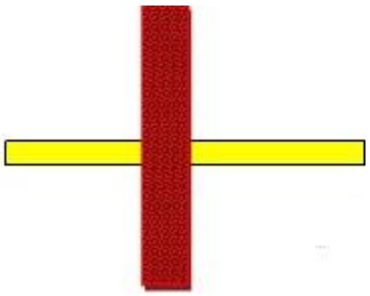
(c) Buses, control signals, interconnections, and 'leaf-cell' boundaries

## CMOS Design Style:

The CMOS design rules are almost similar and extensions of n-MOS design rules except the Implant (yellow) and the buried contact (brown). In CMOS design Yellow is used to identify p transistors and wires, as depletion mode devices are not utilized. The two types of transistors 'n' and 'p', are separated by the demarcation line (representing the p-well boundary) above which all p-type devices are placed (transistors and wires (yellow). The n-devices (green) are consequently placed below the demarcation line and are thus located in the p-well as shown in the diagram below.



N-type (red over green)



P-type (red over yellow)

Diffusion paths must not cross the demarcation line and n-diffusion and p-diffusion wires must not join. The 'n' and 'p' features are normally joined by metal where a connection is needed. Their geometry will appear when the stick diagram is translated to a mask layout. However, one must not forget to place crosses on VDD and Vss rails to represent the substrate and p-well connection respectively. The design style is explained by taking the example the design of a single bit shift register. The design begins with the drawing of the VDD and Vss rails in parallel and in metal and the creation of an (imaginary) demarcation line in-between, as shown in Fig.below. The n-transistors are then placed below this line and thus close to Vss, while p-transistors are placed above the line and below VDD In both cases, the transistors are conveniently placed with their diffusion paths parallel to the rails (horizontal in the diagram) as shown in Fig.(b). A similar approach can be taken with transistors in symbolic form.

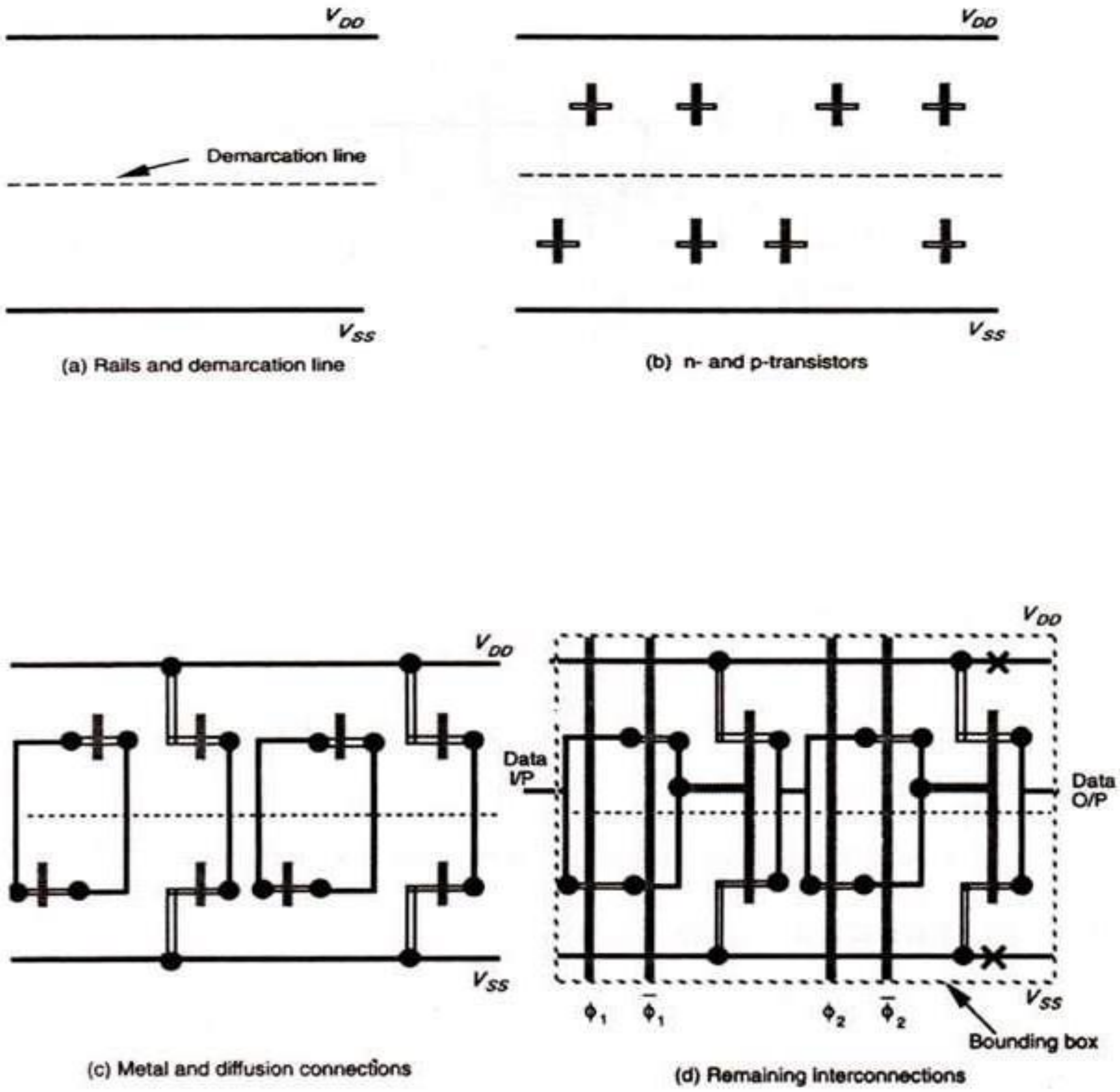
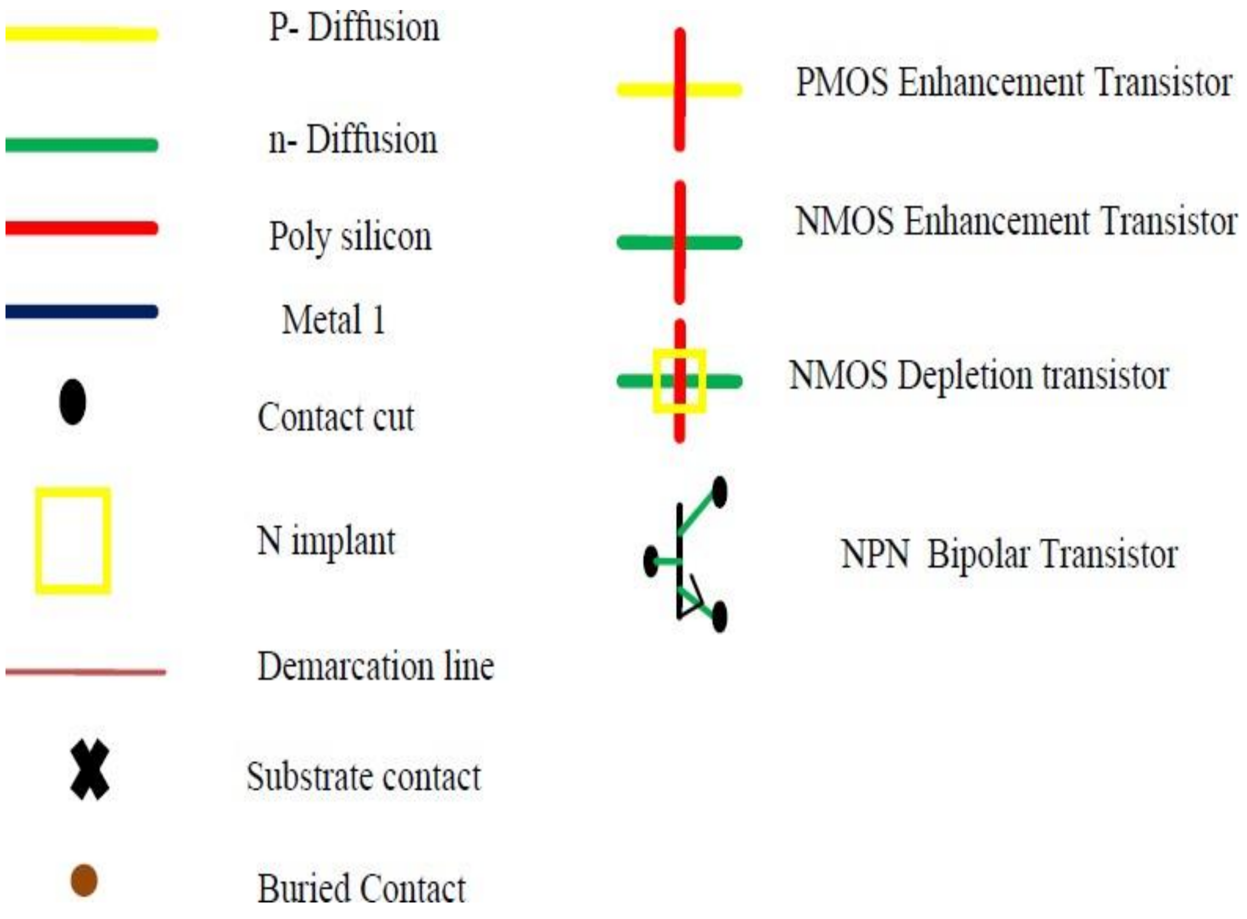


**Fig. CMOS stick layout design style (a,b,c,d)**

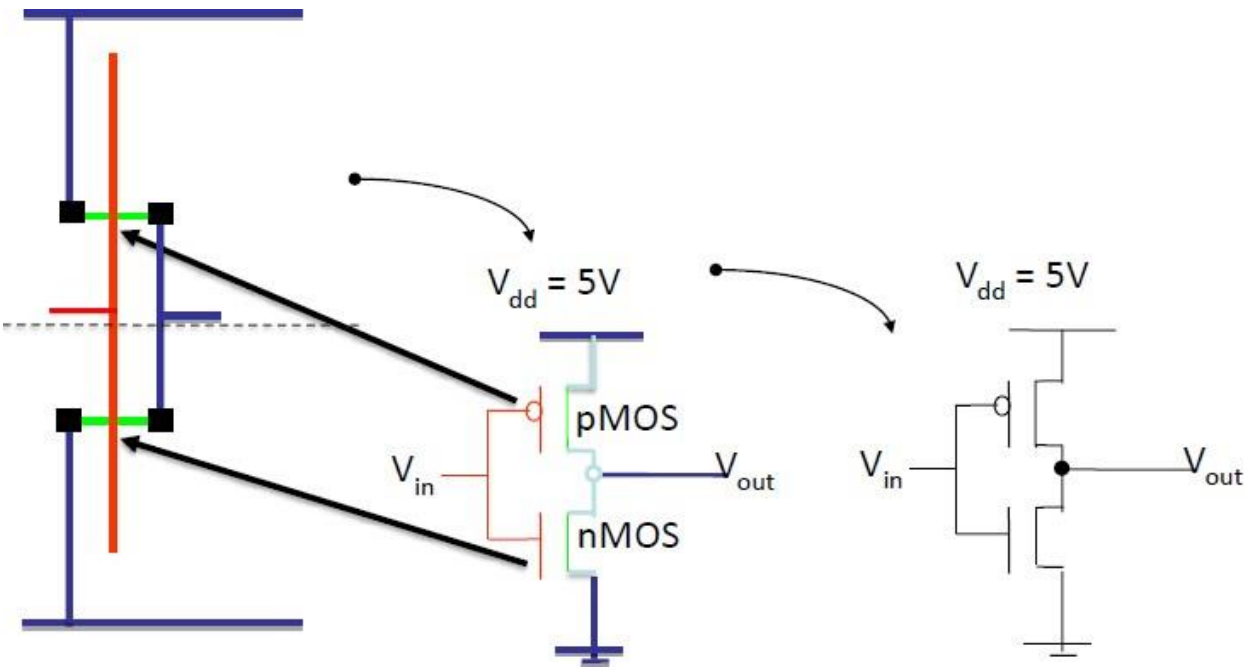
The n- along with the p-transistors are interconnected to the rails using the metal and connect as Shown in Fig.(d). It must be remembered that only metal and poly-silicon can cross the demarcation line but with that restriction, wires can run-in diffusion also. Finally, the remaining interconnections are made as appropriate and the control signals and data inputs are added as shown in the Fig.(d).

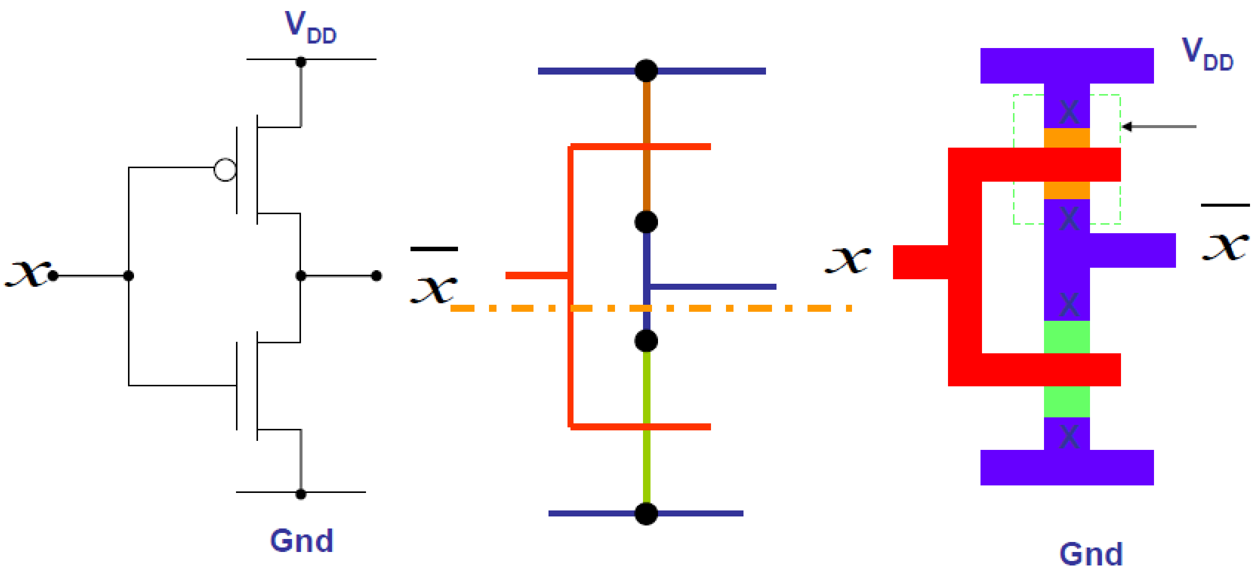
## Stick Diagrams:



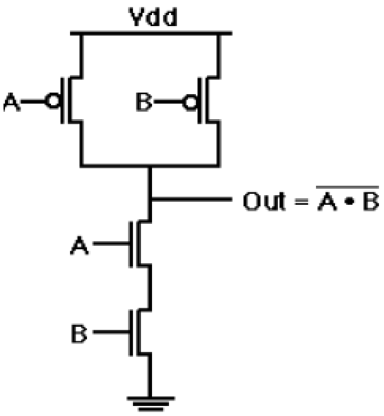
## Examples of Stick Diagrams

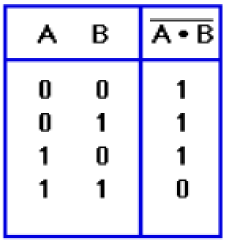
### CMOS Inverter



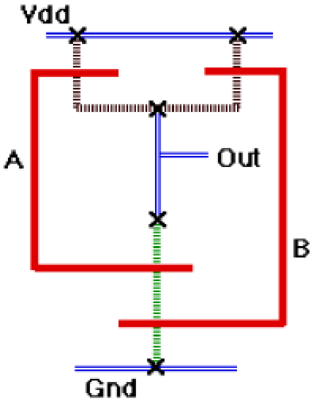


Contd….





| A | B | $\overline{A \cdot B}$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

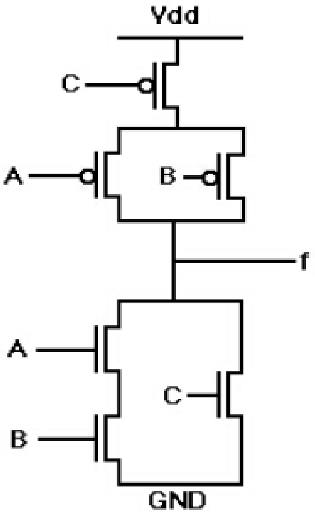


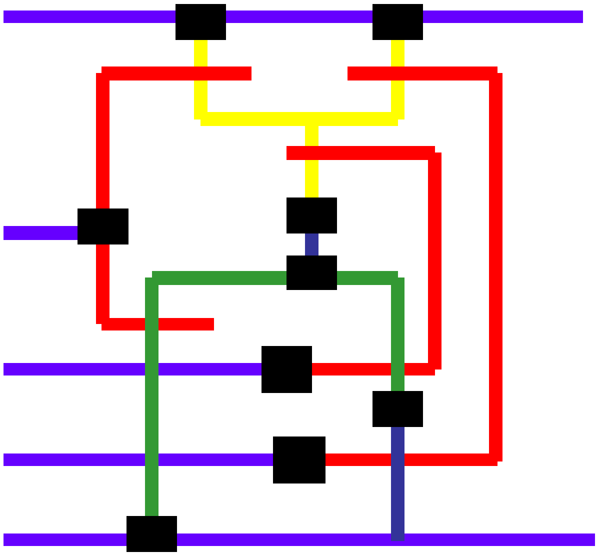
1. Pull-down: Connect to ground If A=1 AND B=1

2. Pull-up: Connect to Vdd If A=0 OR B=0

Fig. CMOS NAND gate

Example: f = $\overline{(A \cdot B) + C}$





## Design Rules and Layout

In VLSI design, as processes become more and more complex, need for the designer to understand the intricacies of the fabrication process and interpret the relations between the different photo masks is really troublesome. Therefore, a set of layout rules, also called **design rules**, has been defined. They act as an interface or communication link between the circuit designer and the process engineer during the manufacturing phase. The objective associated with layout rules is to obtain a circuit with optimum yield (functional circuits versus non-functional circuits) in as small as area possible without compromising reliability of the circuit. In addition, Design rules can be conservative or aggressive, depending on whether yield or performance is desired. Generally, they are a compromise between the two. Manufacturing processes have their inherent limitations in accuracy. So the need of design rules arises due to manufacturing problems like –

- Photo resist shrinkage, tearing.
- Variations in material deposition, temperature and oxide thickness.
- Impurities.
- Variations across a wafer.

These lead to various problems like :

- **Transistor problems:**

  Variations in threshold voltage: This may occur due to variations in oxide thickness, ion-implantation and poly layer. Changes in source/drain diffusion overlap. Variations in substrate.

- **Wiring problems:**

  Diffusion: There is variation in doping which results in variations in resistance, capacitance. Poly, metal: Variations in height, width resulting in variations in resistance, capacitance. Shorts and opens.

- **Oxide problems:**

  Variations in height.

  Lack of planarity.

- **Via problems:**

  Via may not be cut all the way through.

Undersize via has too much resistance.

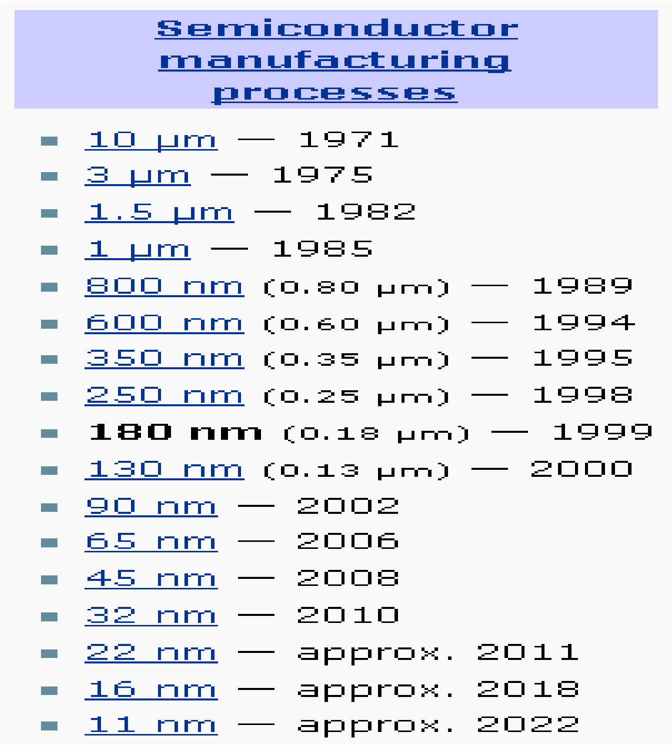
Via may be too large and create short.

To reduce these problems, the design rules specify to the designer certain geometric constraints on the layout artwork so that the patterns on the processed wafers will preserve the topology and geometry of the designs. This consists of minimum-width and minimum-spacing constraints and requirements between objects on the same or different layers. Apart from following a definite set of rules, design rules also come by experience.

**Why we use design rules?**

- Interface between designer and process engineer
- Historically, the process technology referred to the length of the silicon channel between the source and drain terminals in field effect transistors.
- The sizes of other features are generally derived as a ratio of the channel length, where some may be larger than the channel size and some smaller.

For example, in a 90 nm process, the length of the channel may be 90 nm, but the width of the gate terminal may be only 50 nm.



Semiconductor manufacturing processes

- 10 µm — 1971
- 3 µm — 1975
- 1.5 µm — 1982
- 1 µm — 1985
- 800 nm (0.80 µm) — 1989
- 600 nm (0.60 µm) — 1994
- 350 nm (0.35 µm) — 1995
- 250 nm (0.25 µm) — 1998
- **180 nm** (0.18 µm) — 1999
- 130 nm (0.13 µm) — 2000
- 90 nm — 2002
- 65 nm — 2006
- 45 nm — 2008
- 32 nm — 2010
- 22 nm — approx. 2011
- 16 nm — approx. 2018
- 11 nm — approx. 2022

Design rules define ranges for features

Examples:

- min. wire widths to avoid breaks
- min. spacing to avoid shorts
- minimum overlaps to ensure complete overlaps
  - Measured in microns
  - Required for resolution/tolerances of masks

Fabrication processes defined by minimum channel width

- – Also minimum width of poly traces
- – Defines "how fast" a fabrication process is

## Types of Design Rules

The design rules primary address two issues:

1. The geometrical reproduction of features that can be reproduced by the maskmaking and lithographical process, and
2. The interaction between different layers.

There are primarily two approaches in describing the design rules.

1. Linear scaling is possible only over a limited range of dimensions.
2. Scalable design rules are conservative .This results in over dimensioned and less dense design.
3. This rule is not used in real life.

**1. Scalable Design Rules (e.g. SCMOS, λ-based design rules):**

In this approach, all rules are defined in terms of a single parameter **λ**. The rules are so chosen that a design can be easily ported over a cross section of industrial process ,making the layout portable .Scaling can be easily done by simply changing the value of.

The key disadvantages of this approach are:

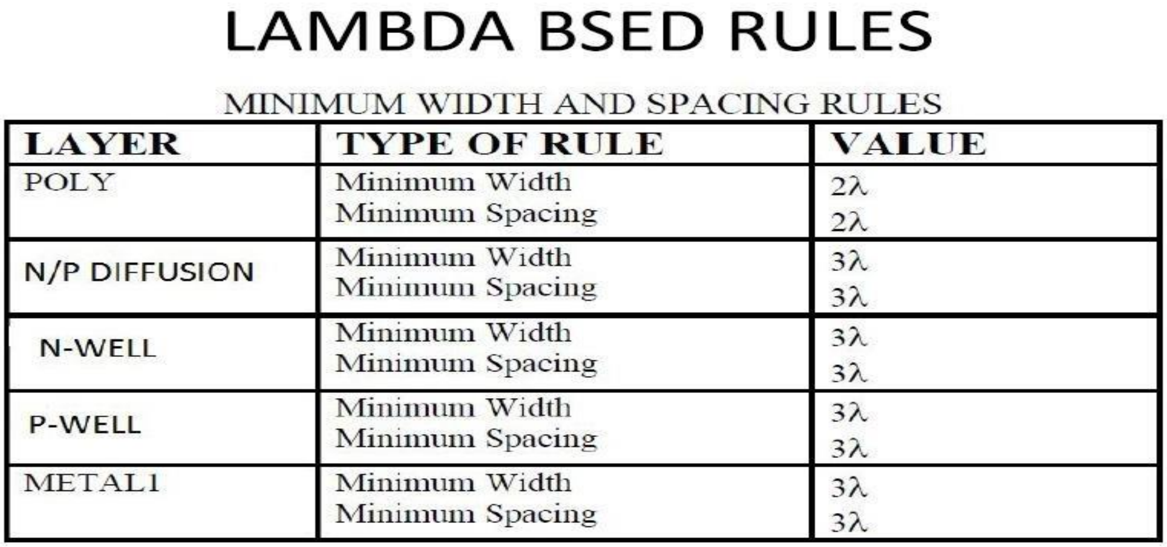
**2. Absolute Design Rules (e.g. µ-based design rules ) :**

In this approach, the design rules are expressed in absolute dimensions (e.g. 0.75µm) and therefore can exploit the features of a given process to a maximum degree. Here, scaling and porting is more demanding, and has to be performed either manually or using CAD tools .Also, these rules tend to be more complex especially for deep submicron.

The fundamental unity in the definition of a set of design rules is the minimum line width .It stands for the minimum mask dimension that can be safely transferred to the semiconductor material .Even for the same minimum dimension, design rules tend to differ from company to company, and from process to process. Now, CAD tools allow designs to migrate between compatible processes.

## LAMBDA-BASED DESIGN RULES:-

- *Lambda-based* (scalable CMOS) design rules define scalable rules based on λ (which is half of the minimum channel length)
  - classes of MOSIS SCMOS rules: SUBMICRON, DEEPSUBMICRON
- Stick diagram is a draft of real layout, it serves as an abstract view between the schematic and layout.
- Circuit designer in general want tighter, smaller layouts for improved performance and decreased silicon area.
- On the other hand, the process engineer wants design rules that result in a controllable and reproducible process.
- Generally we find there has to be a compromise for a competitive circuit to be produced at a reasonable cost.
- All widths, spacing, and distances are written in the form
- λ = 0.5 X minimum drawn transistor length
- Design rules based on single parameter, λ
- Simple for the designer
- Wide acceptance
- Provide feature size independent way of setting out mask
- If design rules are obeyed, masks will produce working circuits
- Minimum feature size is defined as 2 λ
- Used to preserve topological features on a chip
- Prevents shorting, opens, contacts from slipping out of area to be contacted



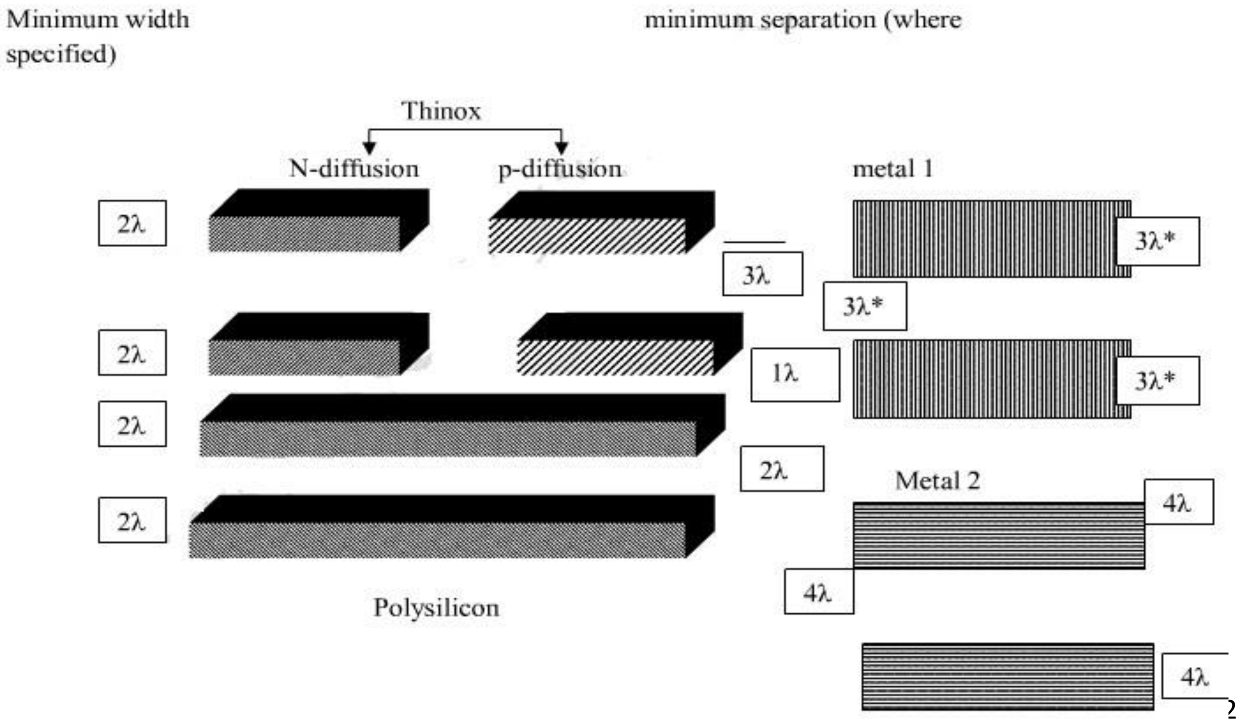
# LAMBDA BSED RULES

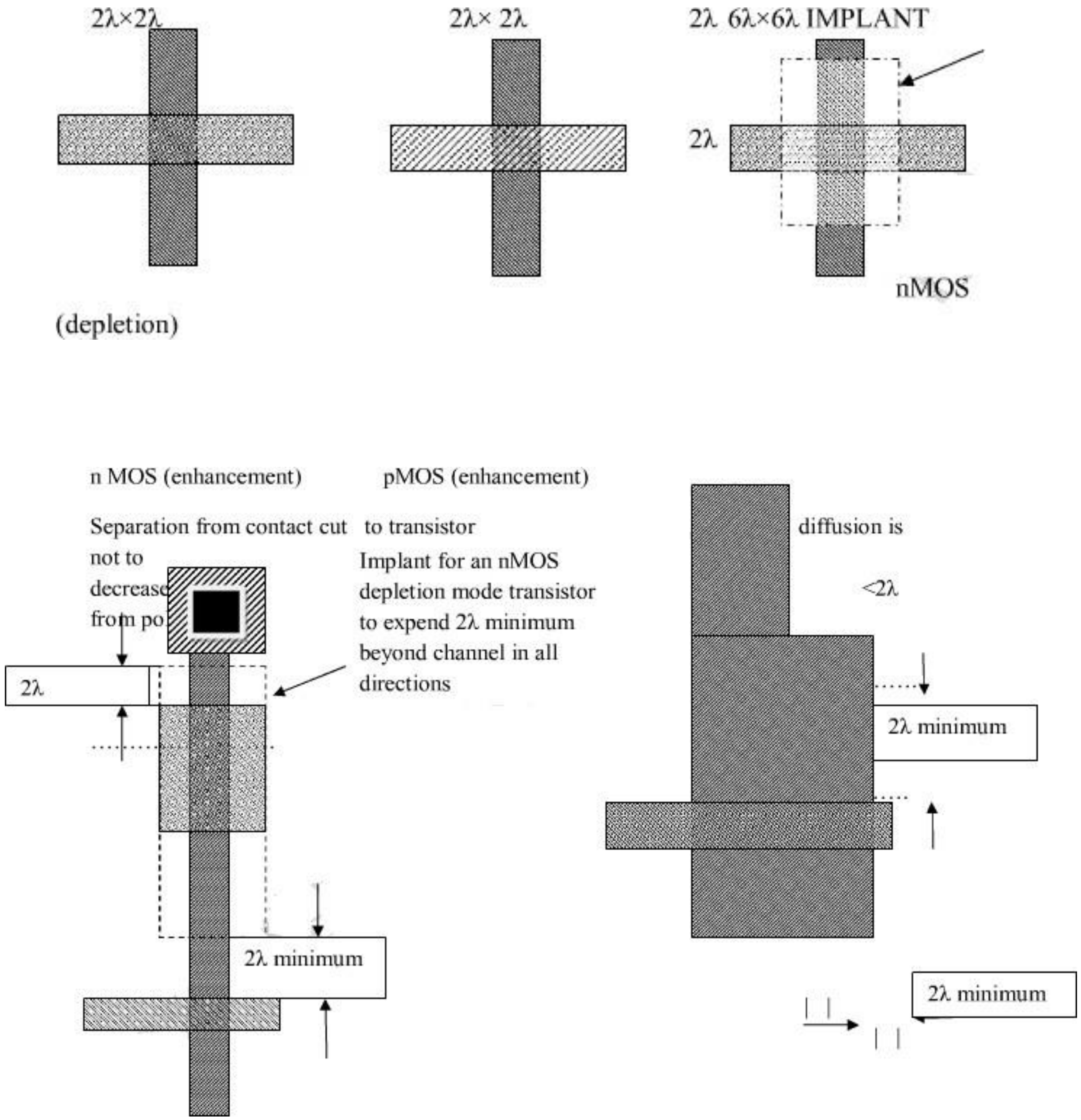
MINIMUM WIDTH AND SPACING RULES

| LAYER | TYPE OF RULE | VALUE |
|---|---|---|
| POLY | Minimum Width<br>Minimum Spacing | 2λ<br>2λ |
| N/P DIFFUSION | Minimum Width<br>Minimum Spacing | 3λ<br>3λ |
| N-WELL | Minimum Width<br>Minimum Spacing | 3λ<br>3λ |
| P-WELL | Minimum Width<br>Minimum Spacing | 3λ<br>3λ |
| METAL1 | Minimum Width<br>Minimum Spacing | 3λ<br>3λ |

## DESIGN RULES FOR WIRES (nMOS and CMOS)

Design rules and layout methodology based on the concept of λ provide a process and feature size independent way of setting out mask dimensions to scale. All paths in layers are dimensioned in λ units and subsequently λ can be allocated an appropriate value compatible with the feature size of the fabrication process.



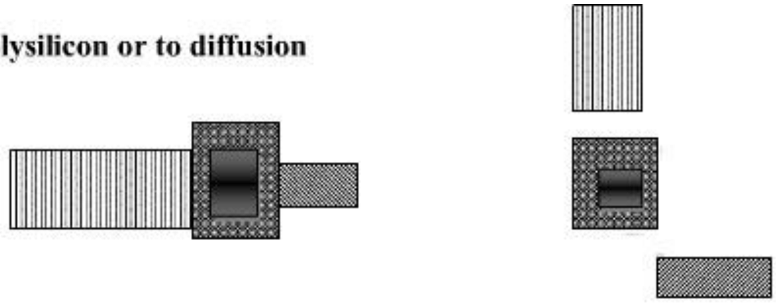


polysilicon to extend a minimum of 2λ beyond diffusion boundaries (width constant)

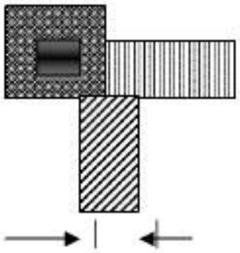
Separation from implant to another transistor

Key: ▨ polysilicon ▨ n-diffusion ▨ p-diffusion ▩ transistor channel ........................................................................................
(Polysilicon over thinox)

**metal 1 to polysilicon or to diffusion**

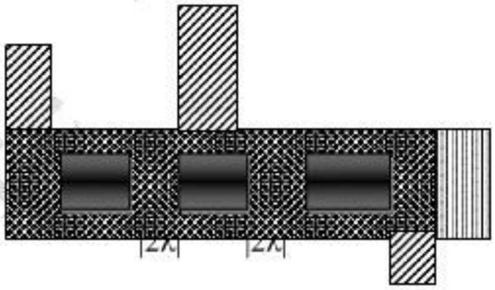


3λ minimum



cuts

2λ minimum

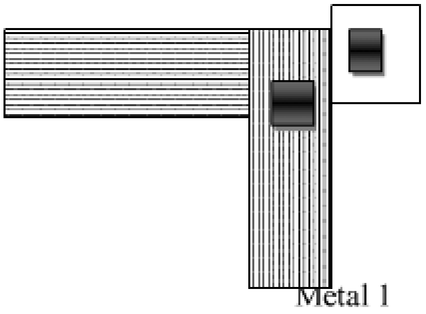


minimum separation multiple

2λ×2λ cut centered on 4λ×4λ superimposed area of layers to be joined in all cases

2λ minimum separation (if other spacing's allowed)

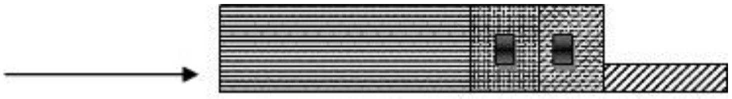
Metal 2



4λ×4λ area of overlap with

2λ×2λ via at center

Via and cut used
to connect metal 2
to diffusion



Contacts (nMOS and CMOS)

## CONTACT CUTS

When making contacts between poly-silicon and diffusion in nMOS circuits it should be remembered that there are three possible approaches--poly. to metal then metal to diff., or aburied contact poly. to diff. , or a butting contact (poly. to diff. using metal). Among the three the latter two, the buried contact is the most widely used, because of advantage in space and a reliable contact. At one time butting contacts were widely used , but now a days they are superseded by buried contacts.

In CMOS designs, poly. to diff. contacts are always made via metal. A simple process is followed for making connections between metal and either of the other two layers (as in Fig.a), The 2λ. x 2λ. contact cut indicates an area in which the oxide is to be removed down to the underlying polysilicon or diffusion surface. When deposition of the metal layer takes place the metal is deposited through the contact cut areas onto the underlying area so that contact is made between the layers.

The process is more complex for connecting diffusion to poly-silicon using the butting contact approach (Fig.b), In effect, a 2λ. x 2λ contact cut is made down to each of the layers to be joined. The layers are butted together in such a way that these two contact cuts become contiguous. Since the poly-silicon and diffusion outlines overlap and thin oxide under poly silicon acts as a mask in the diffusion process, the poly-silicon and diffusion layers are also butted together. The contact between the two butting layers is then made by a metal overlay as shown in the Fig.

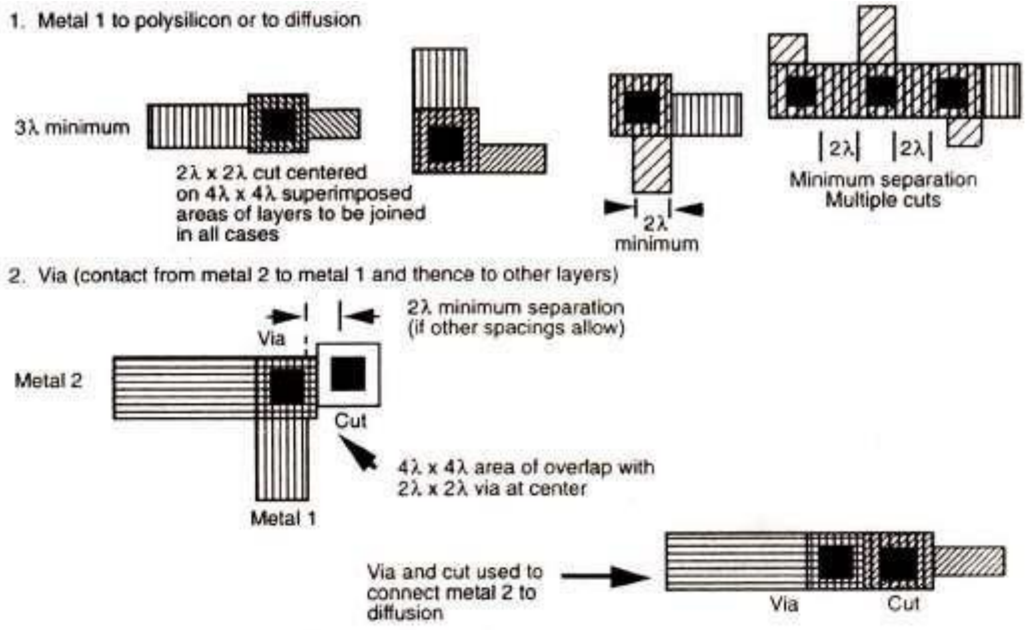

1. Metal 1 to polysilicon or to diffusion
3λ minimum
2λ x 2λ cut centered on 4λ x 4λ superimposed areas of layers to be joined in all cases
2λ minimum
2λ
2λ
Minimum separation Multiple cuts
2. Via (contact from metal 2 to metal 1 and thence to other layers)
Via
2λ minimum separation (if other spacings allow)
Metal 2
Cut
4λ x 4λ area of overlap with 2λ x 2λ via at center
Metal 1
Via and cut used to connect metal 2 to diffusion
Via
Cut

Fig.(a) . n-MOS & C-MOS Contacts

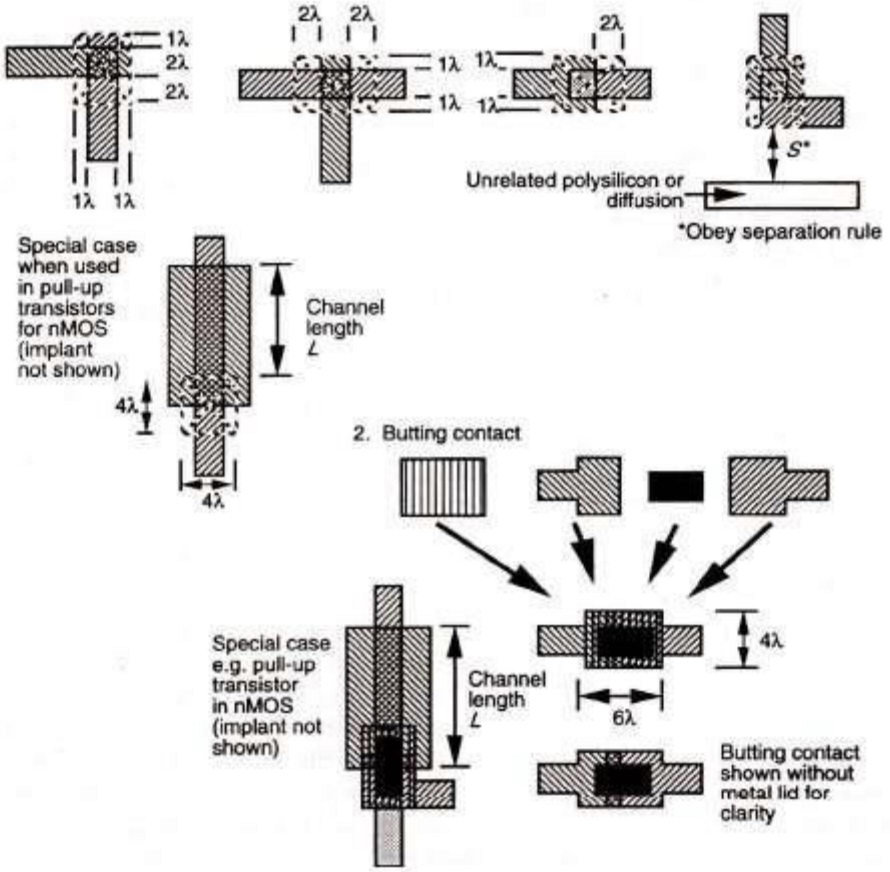

1λ
2λ
2λ
1λ 1λ
2λ 2λ
1λ
1λ
1λ
1λ
2λ
S*
Unrelated polysilicon or diffusion
*Obey separation rule
Special case when used in pull-up transistors for nMOS (implant not shown)
Channel length L
4λ
4λ
2. Butting contact
Special case e.g. pull-up transistor in nMOS (implant not shown)
Channel length L
4λ
6λ
Butting contact shown without metal lid for clarity

Fig.(b). Contacts poly-silicon to diffusion

In buried contact basically, layers are joined over a 2λ. x 2λ. area with the buried contact cut extending by 1λ, in all directions around the contact area except that the contact cut extension is increased to 2λ. in diffusion paths leaving the contact area. This helps to avoid the formation of unwanted transistors. So this buried contact approach is simpler when compared to others. The, poly-silicon is deposited directly on the underlying crystalline wafer. When diffusion takes place, impurities will diffuse into the poly-silicon as well as into the diffusion region within the contact area. Thus a satisfactory connection between poly-silicon and diffusion is ensured. Buried contacts can be smaller in area than their butting contact counterparts and, since they use no metal layer, they are subject to fewer design rule restrictions in a layout.

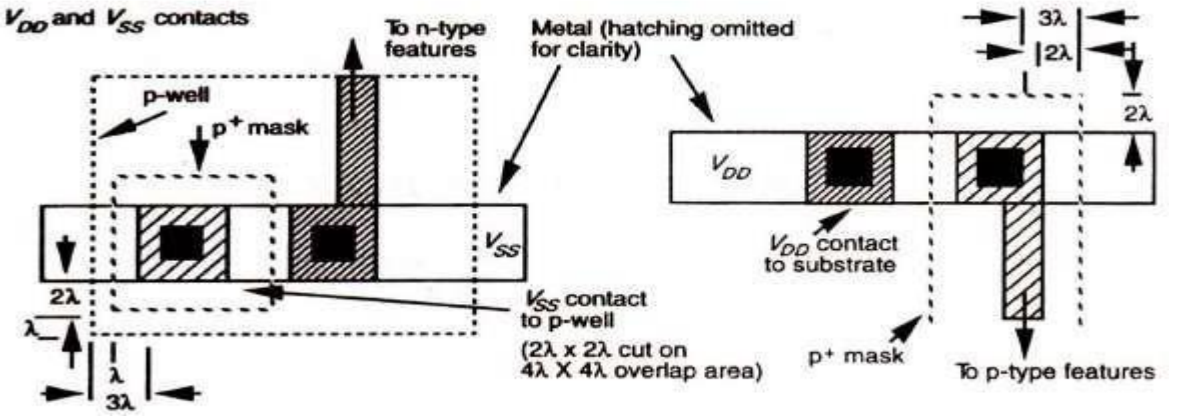
## Other design rules

- Double Metal MOS process Rules
- CMOS fabrication is much more complex than nMOS fabrication

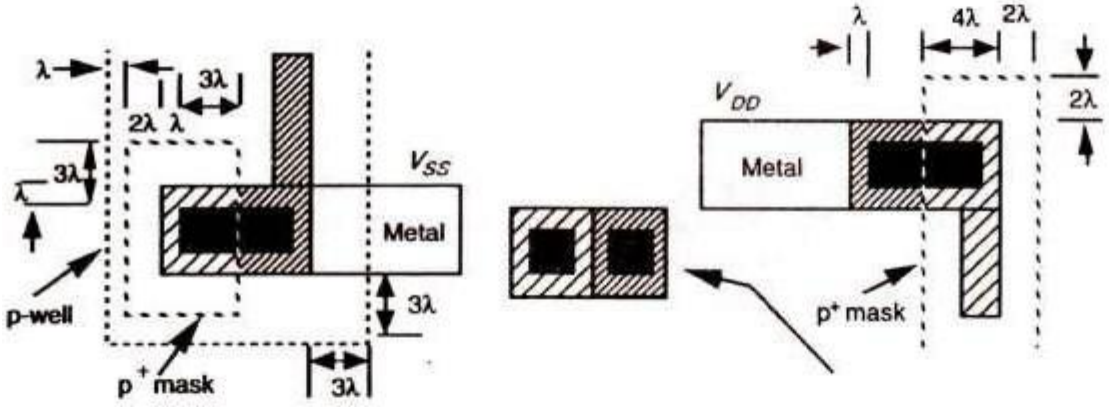
- 2 um Double metal, Double poly. CMOS/BiCMOS Rules
- 1.2um Double Metal single poly.CMOS rules

**CMOS Lambda-based Design Rules:**

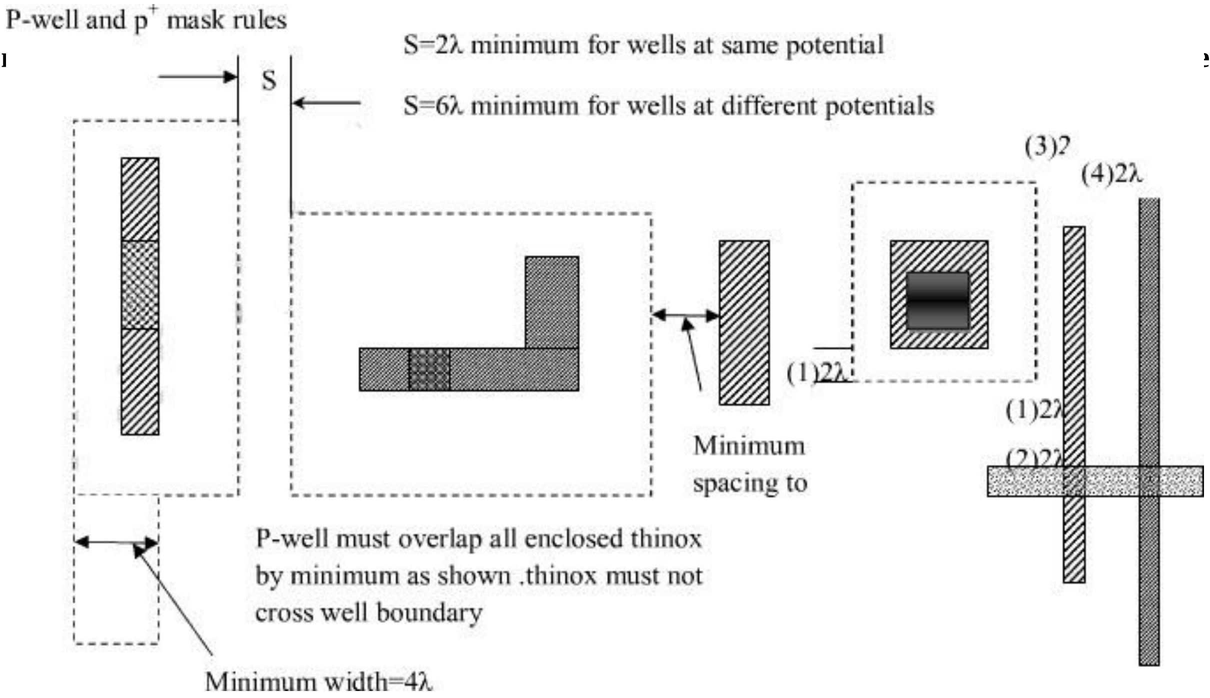
The CMOS fabrication process is more complex than nMOS fabrication. In a CMOS process, there are nearly 100 actual set of industrial design rules. The additional rules are concerned with those features unique to p-well CMOS, such as the p-well and p+ mask and the special 'substrate contacts. The p-well rules are shown in the diagram below


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

In the diagram above each of the arrangements can be merged into single split contacts.



From the above diagram it is also clear that split contacts may also be made with separate cuts.


P-well and p$^+$ mask rules
S
S=2λ minimum for wells at same potential
S=6λ minimum for wells at different potentials
(3)2
(4)2λ
(1)2λ
(1)2λ
(2)2λ
Minimum
spacing to
P-well must overlap all enclosed thinox
by minimum as shown .thinox must not
cross well boundary
Minimum width=4λ

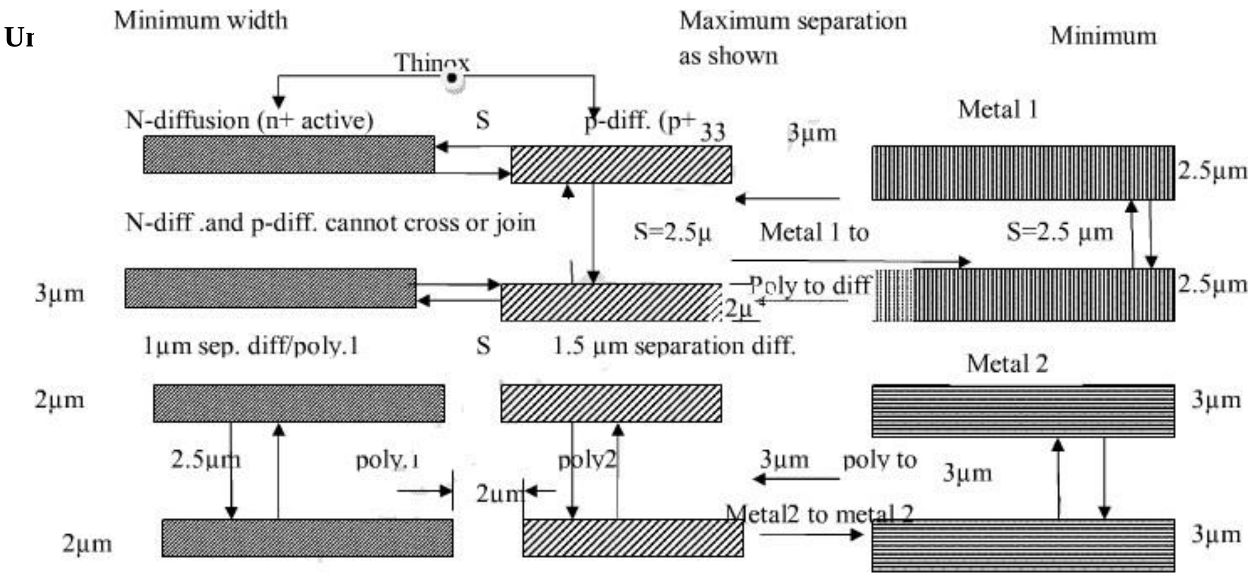
The CMOS rules are designed based on the extensions of the Mead and Conway concepts and also by excluding the butting and buried contacts the new rules for CMOS design are formed. These rules for CMOS design are implemented in the above diagrams.

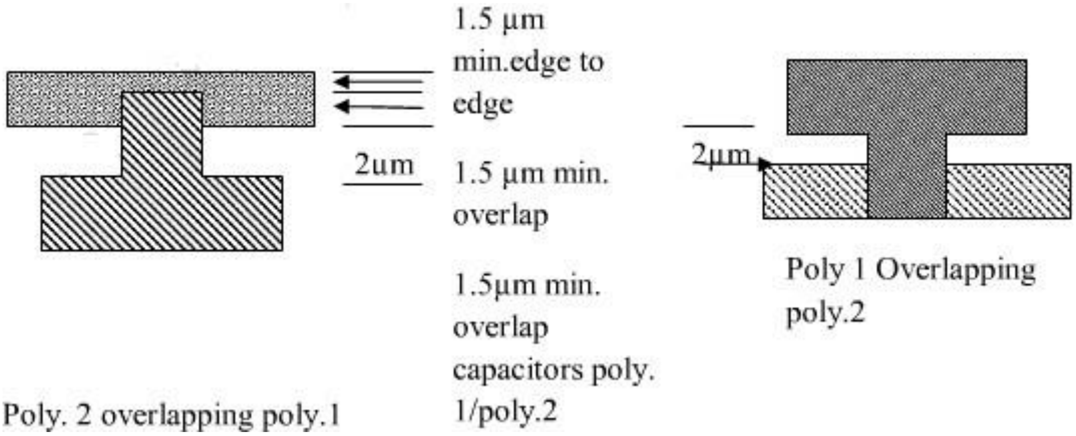
## µM CMOS Design rules

The encoding is compatible with that already described where as following extension are made: n-well brown →

Poly 1→ red; poly 2 → orange; diff (n-active) → green; p Diff (p-active)→ yellow.

For BiCMOS the following are added: buried n$^+$ sub collector- pale green; p-base--pink.




1.5 µm min.edge to edge
2um
1.5 µm min. overlap
1.5µm min. overlap capacitors poly. 1/poly.2
2µm
Poly 1 Overlapping poly.2
Poly. 2 overlapping poly.1

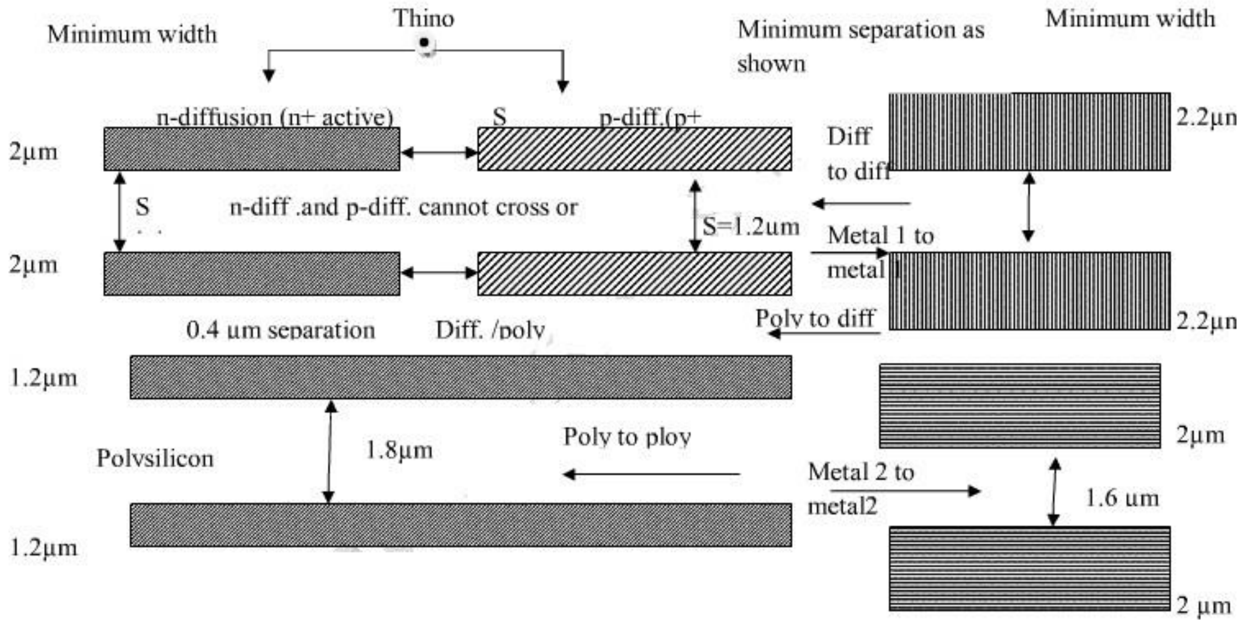
AVOID COINCIDENT EDGES WHERE METAL 1 AND METAL 2 RUNS FOLLOW THE SAME PATH FOR >25µm LENGTH (UNDER LAP METAL 1

Design rules for wires (interconnects) (orbit 2µm CMOS)

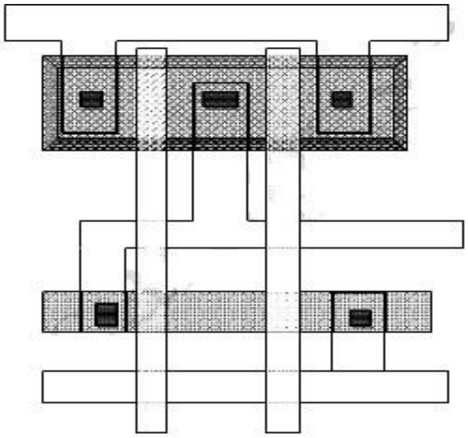
**2µm DOUBLE METAL, SINGLE POLY CMOS RULES**

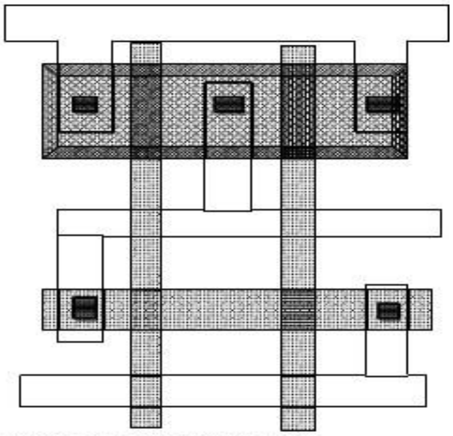
The orbit TM 1.2µm rules provide improved feature size. A separate set of micro based design rules accompany them

Design rules for wires (interconnects) (orbit 1.2 µm CMOS)

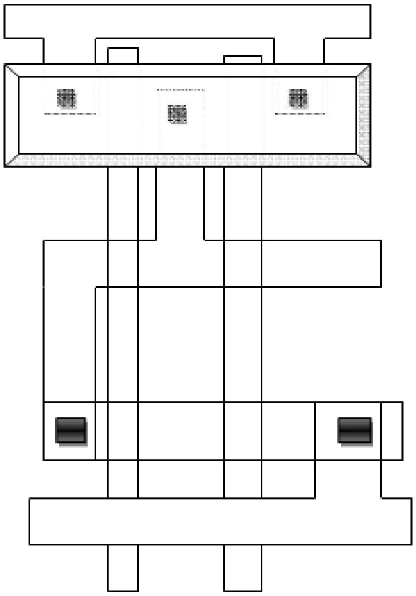


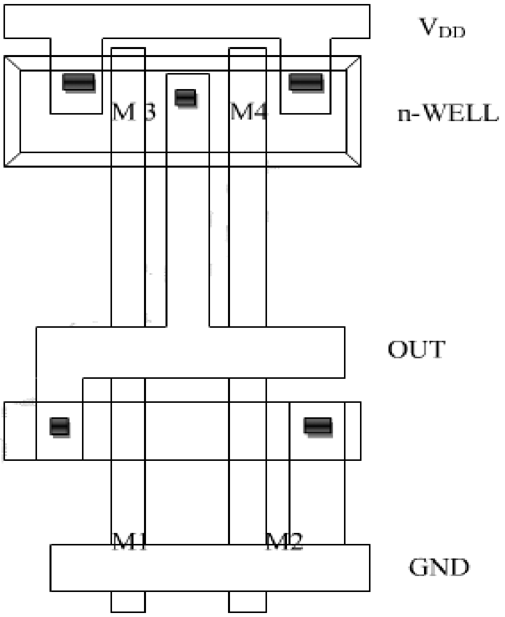
Avoid coincident edges where metal 1 and metal2 runs follow the same path for>25µm length (under lap metal 1 edges by 0.8 µm).





N-WELL AND ACTIVE AREA MASKs       POLY MASK -> DEFINE NMOS
AND . ..
.................................................................................PMOS
TRANSISTORS

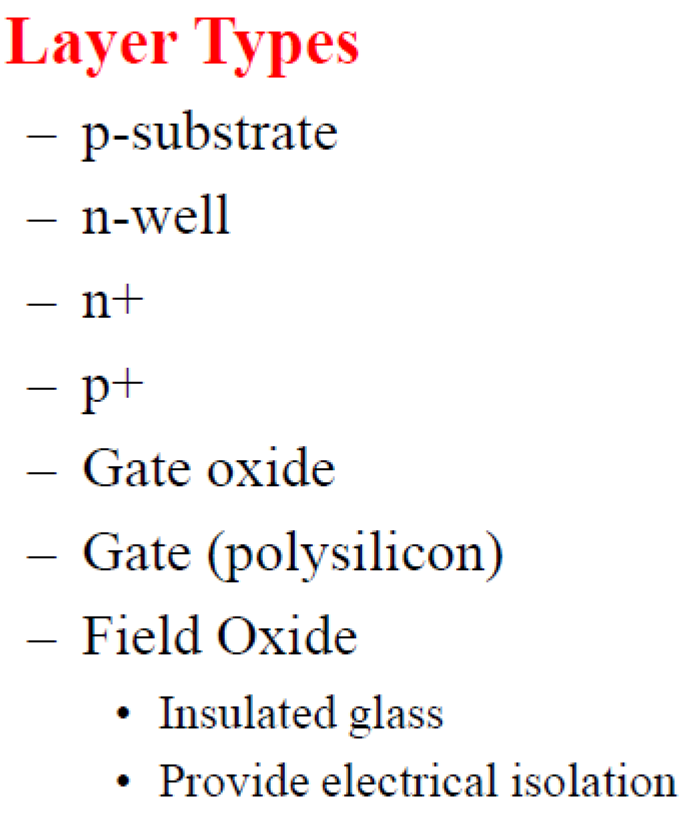




Metal mask for $V_{DD}$, GND and output connections

METAL –DIFFUSION
CONSTANT MASK

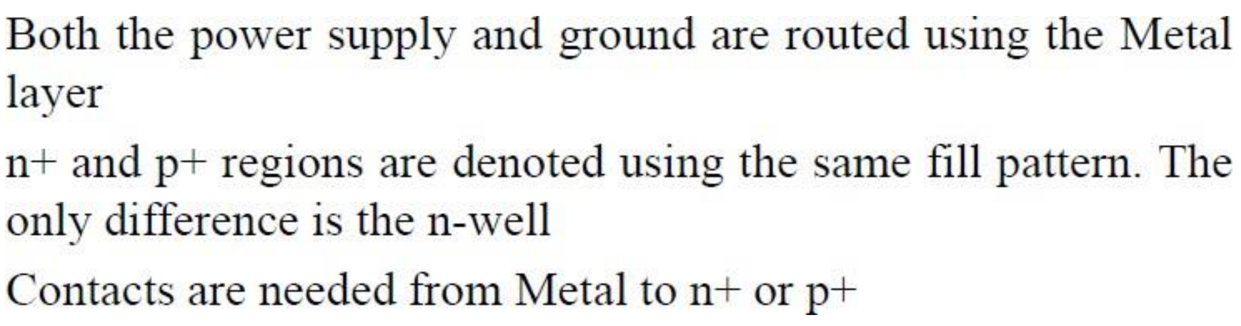
## Layout Diagrams for NMOS and CMOS Inverters and Gates



### Layer Types

- p-substrate
- n-well
- n+
- p+
- Gate oxide
- Gate (polysilicon)
- Field Oxide
  - Insulated glass
  - Provide electrical isolation

## Basic Gate Design

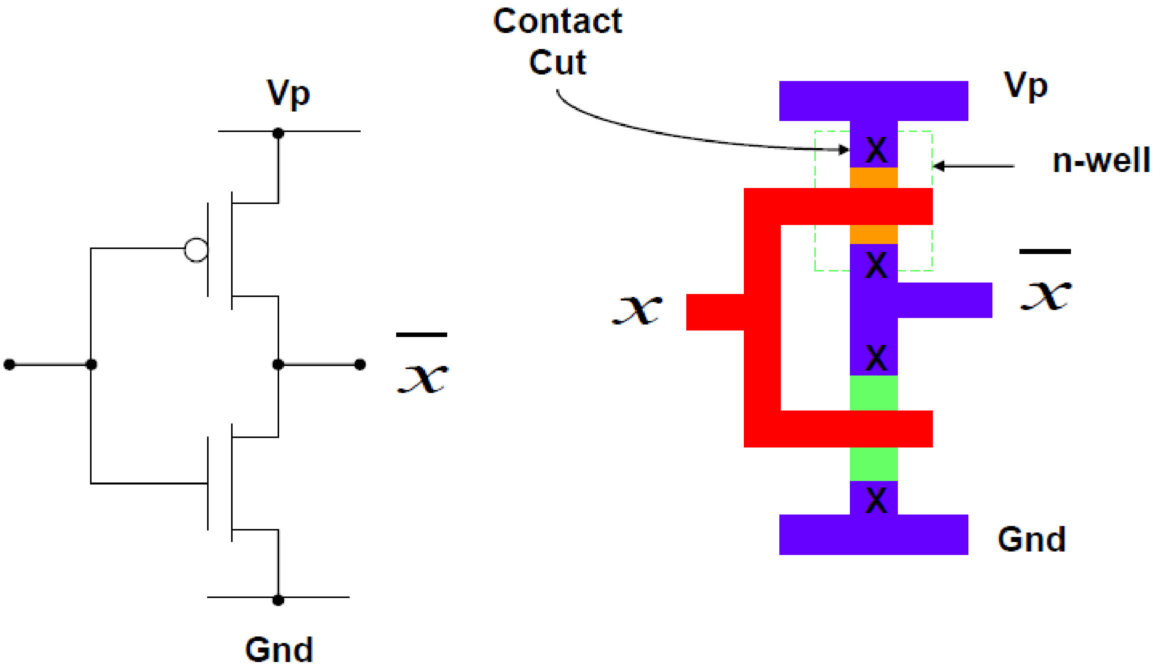


Both the power supply and ground are routed using the Metal layer

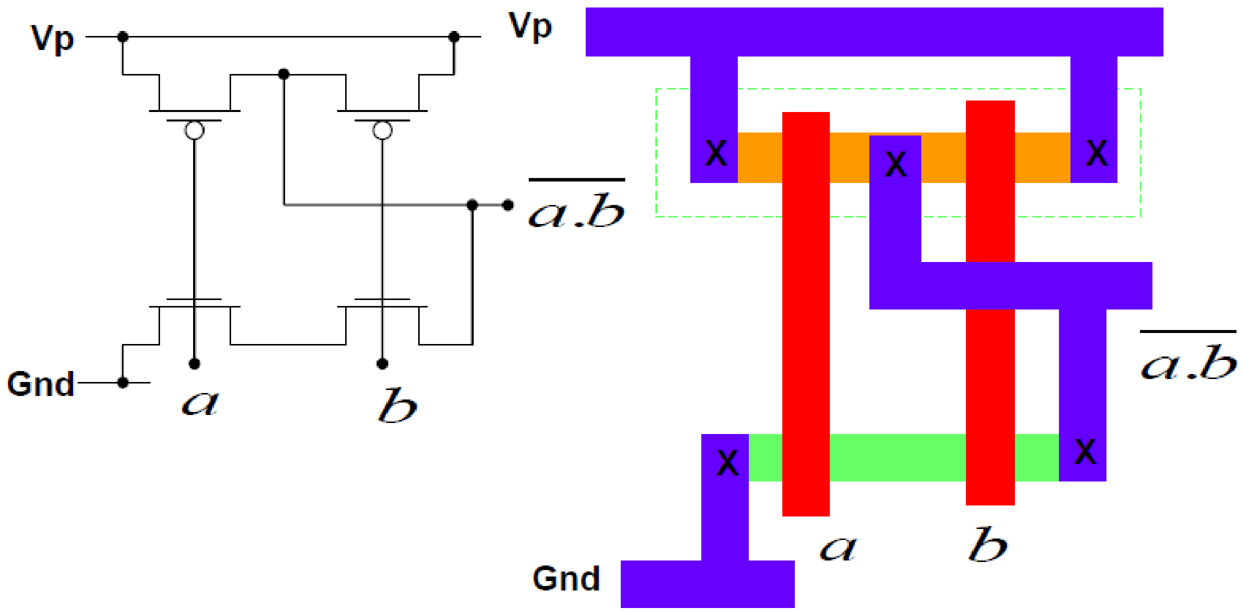
n+ and p+ regions are denoted using the same fill pattern. The only difference is the n-well

Contacts are needed from Metal to n+ or p+

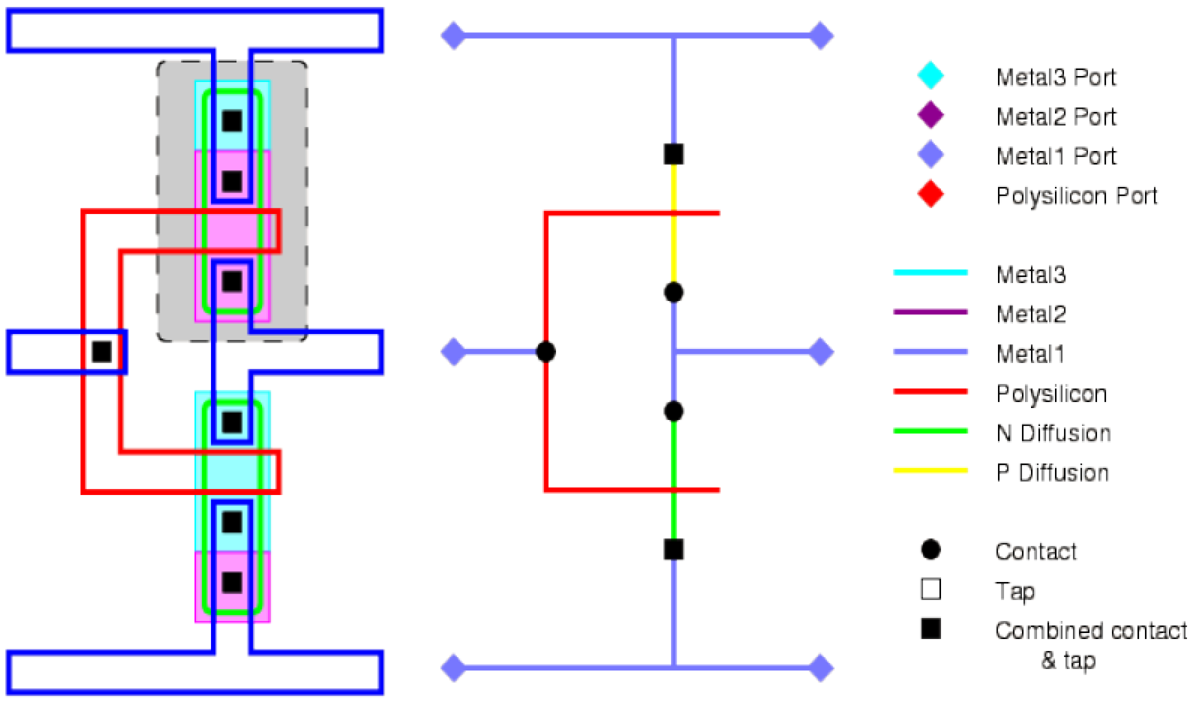
# The CMOS NOT Gate



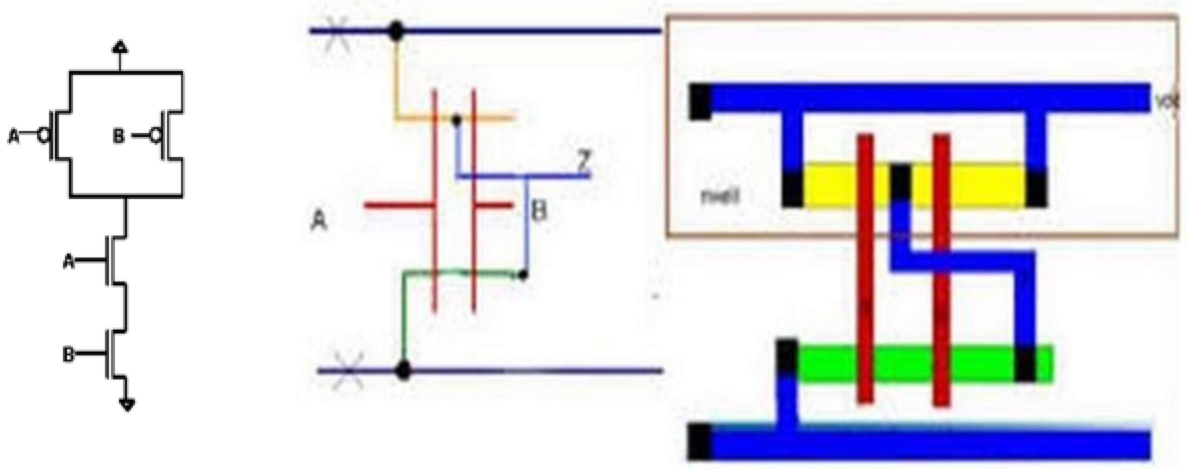
# The CMOS NAND Gate



## Layout & Stick Diagram of CMOS Inverter

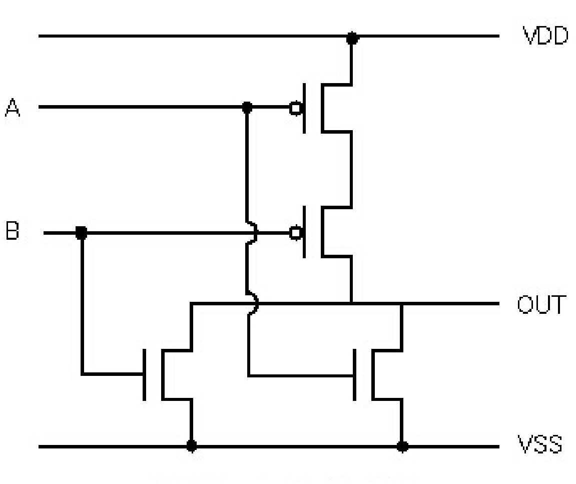


## 2 input NAND gate

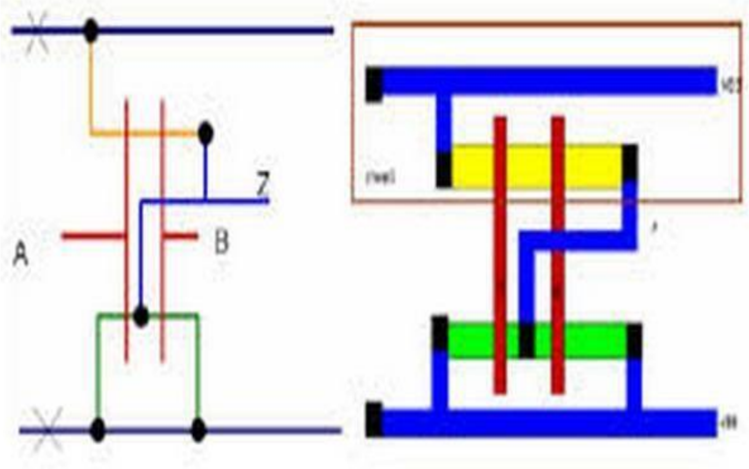


surface.

## 2 input NOR gate



NOR gate in CMOS

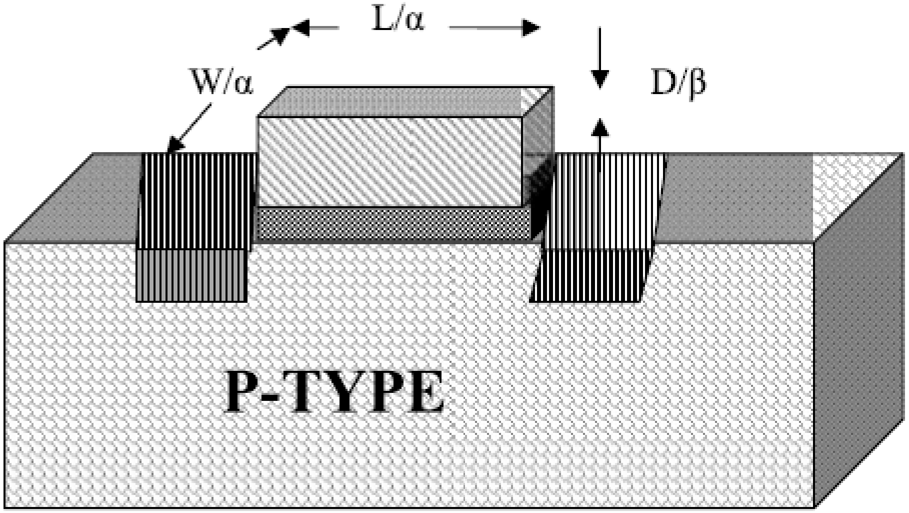


## Scaling of MOS circuits

Scaling means to reduce the feature size and to achieve higher packing density of circuitry on a chip, Many figures of merit such as minimum feature size, number of gates on one chip, power dissipation, maximum operational frequency, die size, production cost can be improved by shrinking the dimensions of transistors, interconnections and the separation between features, and by adjusting the doping levels and supply voltages.

## SCALING MODELS AND SCALING FACTORS:

The most commonly used models are the constant electric field scaling models and the constant voltage scaling model. One more model called as combined voltage and dimension scaling model is presented recently. The following figure indicates the device dimensions and substrate doping level which are associated with the scaling of a transistor.



Two scaling factors 1/α,1/β are used. 1/β is chosen as the scaling factor for supply voltage $V_{DD}$ and gate oxide thickness D, and 1/α is used for all other linear dimensions, both vertical and horizontal to chip surface.

## SCALING FACTORS FOR DEVICE PARAMETERS:

### GATE AREA AG :

$$Ag = L.W$$

Where L and W are the channel length and width respectively, both are scaled by $1/\alpha$. So Ag is scaled by $1/\alpha^2$

### GATE CAPACITANCE PER UNIT ARE $C_O$ OR $C_{OX}$:

$$C_o = E_{OX}/D$$

Where $E_{OX}$ is the permittivity of the gate oxide (thinox) ($=E_{ins}.E_O$) and D is the gate oxide thickness which is scaled by $1/\beta$

Thus $C_O$ is scaled by $1/1/\beta=\beta$

### GATE CAPACITANCE $C_g$:

$$C_g = C_O\ L.W$$

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

### PARASITIC CAPACITANCE $C_X$:

$C_X$ is proportional to $A_X/d$.

Where d is the depletion width around source or drain which is scaled by $1/\alpha$ and $A_X$ is the area of depletion region around source or drain which is scaled by $1/\alpha_2.1/1/\alpha = 1/\alpha$

### CARRIER DENSITY IN CHANNEL $Q_{on}$

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

Where Qon is the average charge per unit area in the channel in the 'on' state. $C_o$ is scaled by $\beta$ and $V_{gs}$ is scaled by $1/\beta$.

### GATE DELAY $T_d$

**Td is proportional to $R_{on}.C_g$.**

Thus $T_d$ is scaled by $\beta^2/\alpha^4$

### MAXIMUM OPERATING FREQUENCY $F_O$:

$$F_o = W/L * \mu C_O V_{DD}/Cg$$

Or $f_o$ is inversely proportional to delay $T_d$. Thus $f_o$ is scaled by $1/\beta/\alpha^2 = \alpha^2/\beta$

### SATURATION CURRENT $I_{DSS}$:

$$I_{dss} = C_{o\mu}/2.W/L.\ (V_{gs}-V_t)^2$$

Nothing that both $V_{gs}$ and $V_t$ are scaled by $1/\beta$,we have $I_{dss}$ is scaled by $\beta(1/\beta)^2=1/\beta$.

**CURRENT DENSITY J:**

$$J=I_{des}/A$$

Where A is the cross sectional area of the channel in the 'on' state which is scaled by $1/\alpha^2$

So, J is scaled by $1/\beta/1/\alpha^2=\alpha^2/\beta$.

## SWITCHING ENERGY PER GATE $E_g$:

$$Eg=C_g/2.(V_{DD})^2$$

So Eg is scaled by $\beta/\alpha^2.1/\beta^2=1/\alpha^2\beta$

## POWER DISSIPATION PER GATE $P_g$:

$P_g$ comprise two components such that

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

Where the static component

$$P_{gs}= (V_{DD})^2/R_{on}$$

And the dynamic component

$$P_{gd}=E_gf_o$$

It will be seen that both $P_{gs}$ and $P_{gd}$ are scaled by $1/\beta^2$

## POWER DISSIPATION PER UNIT AREA:

$$P_a = P_g/A_g$$

So Pa is scaled by $1/\beta^2/1/\alpha^2=\alpha^2/\beta^2$

## POWER-SPEED PRODUCT $P_T$:

$$P_T=P_g.T_d$$

So PT is scaled by $1/\beta^2.\beta/\alpha^2=1/\alpha^2\beta$

## Limitations of Scaling:

Scaling may cause a problem which prevents further miniaturization.

## Substrate doping: -

The built-in (junction) potential $V_B$, is small compared with $V_{DD}$.

(a) Substrate doping scaling factors:

As the channel length of a MOS transistor is reduced, the depletion region widths must also be scaled down to prevent the source and drain depletion regions

$N_B$ is thus maintained at a satisfactory level in the channel region and thus problem is reduced. But depletion width d and built in potential $V_B$ will impose limitations on scaling.

We have $E_{max}$ =2V/d

Where $E_{max}$ is the maximum electric field induced in one-sided step junction

When $N_B$ is increased by α and if $V_\alpha=0$, then $V_\beta$ is increased by ln α and d is decreased by $\sqrt{}$ ln α/α.

There E is increased by inverse of this factor and reaches $E_{crit}$

## Limits of miniaturization

The minimum size of transistor is determined by both process technology and the physics of the device itself.

Transistor size is defined in terms of channel length L. L can be decreased as long as there is no punch through i.e. The depletion region around source should not come closer to that around the drain. So L must be at least 2d from meeting. Depletion region width d for the junctions is given by

$$D=\sqrt{}2E_{si}E_oV/qN_B$$

Where

$E_{si}$= relative permittivity of silicon (~12)

$E_O$= permittivity of free space (=8.85x10^-14)

V=effective voltage across the junction

$$V = V_{a+}V_B$$

q=electron charge

$N_B$=doping level of substrate.

$V_a$= (maximum value $=V_{DD}$)=applied voltage

$V_B$=built-in (junction) potential

And $V_B$ =KT/q. ln $(N_B N_D/n_i^2)$

Where $N_D$ is the source or drain doping and $n_i$ is the intrinsic carrier concentration in silicon.

## Depletion width

When $N_B$ is increased, the depletion width decreases and $V_t$ increases which is not desirable.

We have $\mathbf{V_{drift} = \mu E}$

$V_{drift}$ is the carrier drift velocity and **L=2d**

Transit time $\boldsymbol{\tau}$ $\mathbf{=L/V_{drift} \quad =2d/\mu E}$

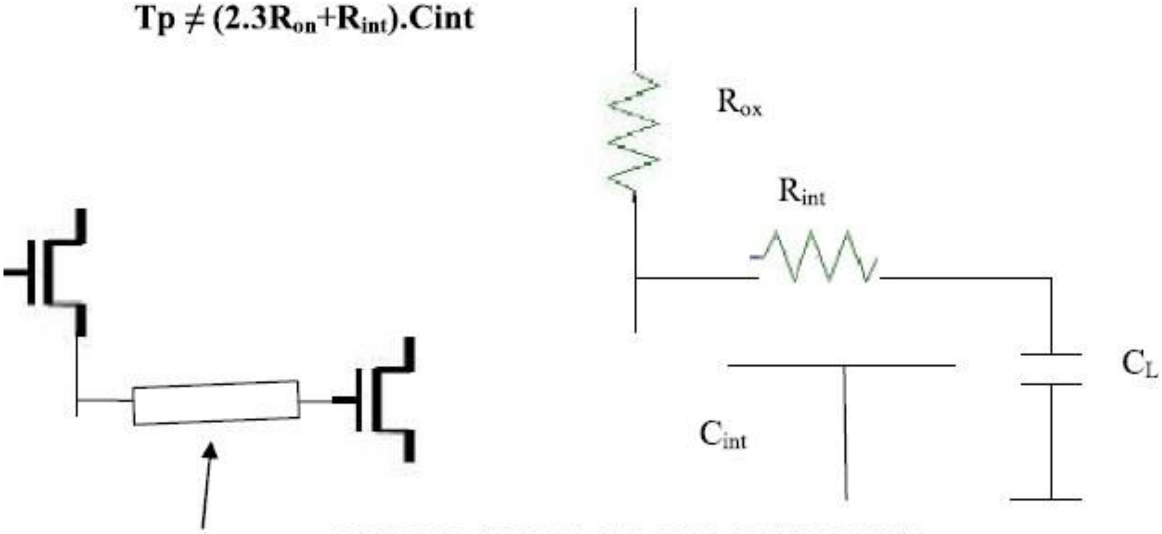
## Limits due to interconnect and contact resistance

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

The propagation delay $T_p$ along a single aluminum interconnect can be calculated from the following equation

$$\mathbf{Tp=R_{int}C_{int}+2.3(R_{on}C_{int}+R_{on}C_{L}+R_{int}C_{L})}$$

$$\mathbf{Tp \neq (2.3R_{on}+R_{int}).Cint}$$



MODEL OF METAL INTERCONNECT

Now

$$\mathbf{R_{int}=\rho L/HW}$$

$$\mathbf{C_{int} =E_{ox}[1.15W/t_{ox}+2.28(H/t_{ox})^{0.222}]L}$$

Where $R_{on}$ is the ON resistance of the transistor.

$R_{int}$ is the resistance of the interconnect

$C_{int}$ is the capacitance of interconnect

$t_{ox}$ is the thickness of dielectric oxide.

ρ is the resistivity of interconnect L,W,H are the length, width and height of the interconnect.

# UNIT III

## GATE LEVEL DESIGN AND BASIC CIRCUIT CONCEPTS

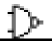
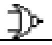
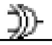
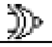
**Gate level Design:**

- Logic gates and other complex gates
- Switch logic
- Alternate gate circuits

**Basic Circuit Concepts:**

- Sheet Resistance Rs and its concepts to MOS
- Area Capacitances calculations
- Inverter Delays
- Fan-in and fan-out.

## CMOS Logic gates and other complex gates

| Name | Logic symbol | Logic equation |
|---|---|---|
| INVERTER | | Out=~in; |
| AND | | Out=a&b; |
| NAND |  | Out=~(a.b); |
| OR | | Out=(a\|b); |
| NOR |  | Out=~(a\|b); |
| XOR |  | Out=a^b; |
| XNOR |  | Out=~(a^b); |

## CMOS logic gate concept:

The structure of a CMOS logic gate is based on complementary networks of n-channel and p-channel MOS circuits. Recall that the pMOS switch is good at passing logic signal '1', while nMOS switches are good at passing logic signal '0'. The operation of the gate has two main configurations:

- the nMOS switch network is closed, the output s=0 (figure 6-6 left)
- the pMOS switch network is closed, the output s=1 (figure 6-6 right)

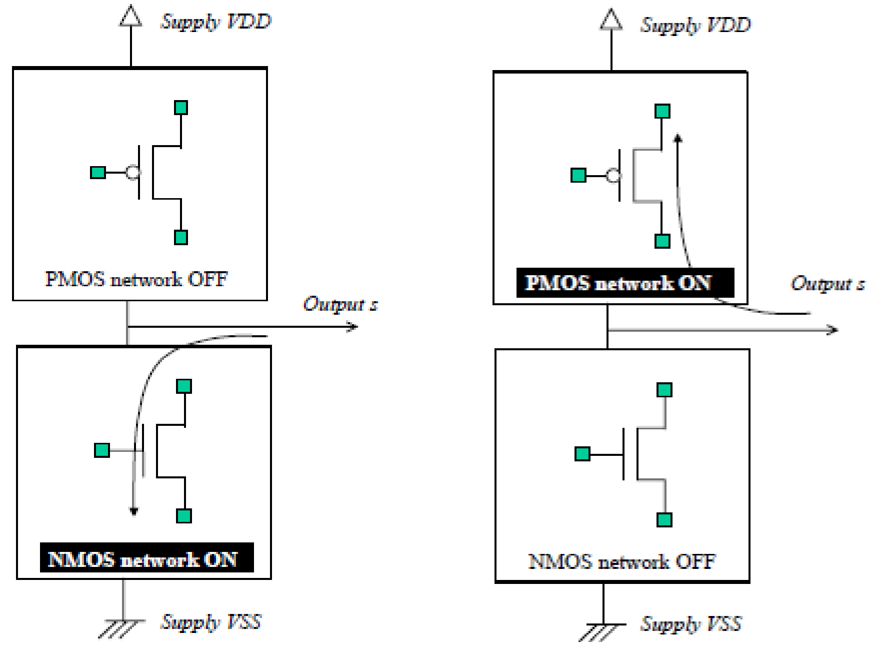


*Fig. 6-6. General structure of a CMOS basic gate*

Using complementary pairs of nMOS and pMOS devices, either the lower nMOS network is active, which ties the output to ground, either the upper pMOS network is active, which ties the output to VDD. In conventional CMOS basic gates, there should exist no combination when both nMOS and pMOS networks would be ON. If this case had happened, a resistive path would be created between VDD and VSS supply rails. The situation where neither nMOS and pMOS networks would be OFF should also be avoided, because the output would be undetermined.

## CMOS Static logic

Static, fully complementary CMOS gate designs using inverter, NAND and NOR gates can build more complex functions. These CMOS gates have good noise margins and low static power dissipation at the cost of more transistors when compared with other CMOS logic designs. CMOS static complementary gates have two transistor nets (nMOS and pMOS) whose topologies are related. The pMOS transistor net is connected between the power supply and ie logic gate output, whereas the nMOS transistor topology is connected between the output nd ground (Fig. 5.1). We saw this organisation with the NAND and NOR gates, but we point ut this topology to lead to a general technique to convert Boolean algebra statements to MOS electronic circuits.

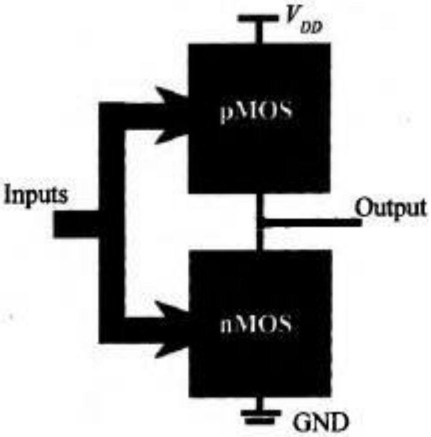
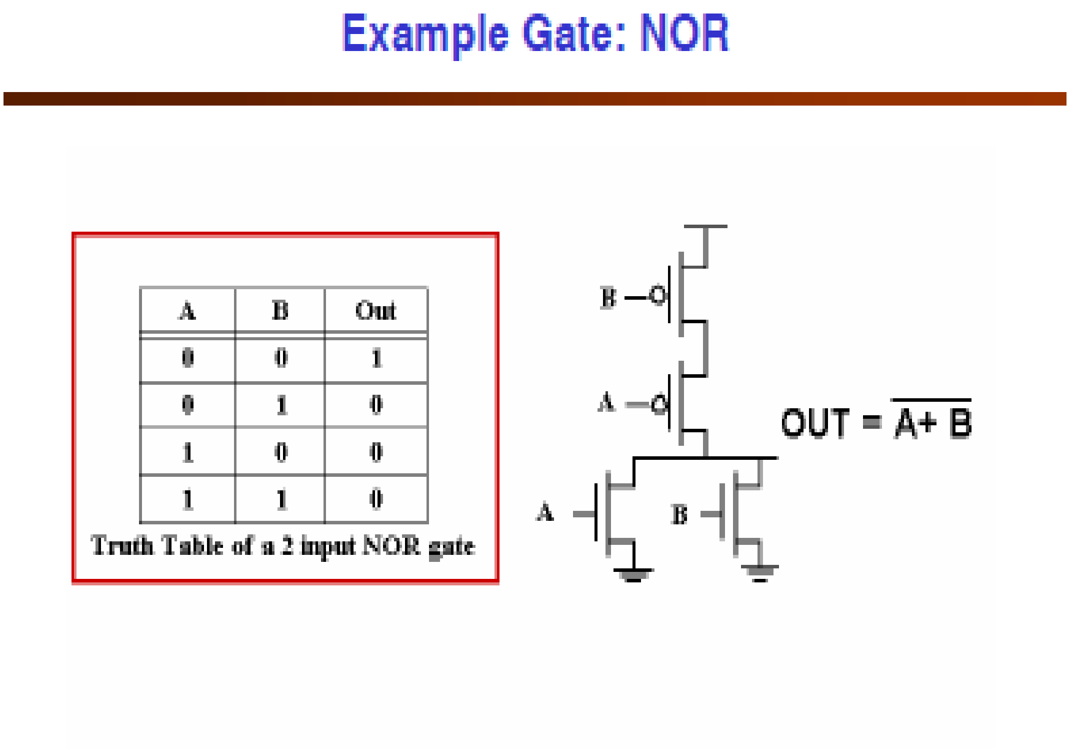


**Fig. 5.1** Standard configuration of a CMOS complementary gate.

## Design Procedure:

1. Derive the nMOS transistor topology with the following rules:
   - Product terms in the Boolean function are implemented with series-connected nMOS transistors.
   - Sum terms are mapped to nMOS transistors connected in parallel.
2. The pMOS transistor network has a dual or complementary topology with respect to the nMOS net. This means that serial transistors in the nMOS net convert to parallel transistors in the pMOS net, and parallel connections within the nMOS block are translated to serial connections in the pMOS block.
3. Add an inverter to the output to complete the function if needed. Some functions are inherently negated, such as NAND and NOR gates, and do not need an inverter at the output state. An inverter added to a NAND or NOR function produces the AND and OR function. The examples below require an inverter to fulfil the function.

**Examples:**

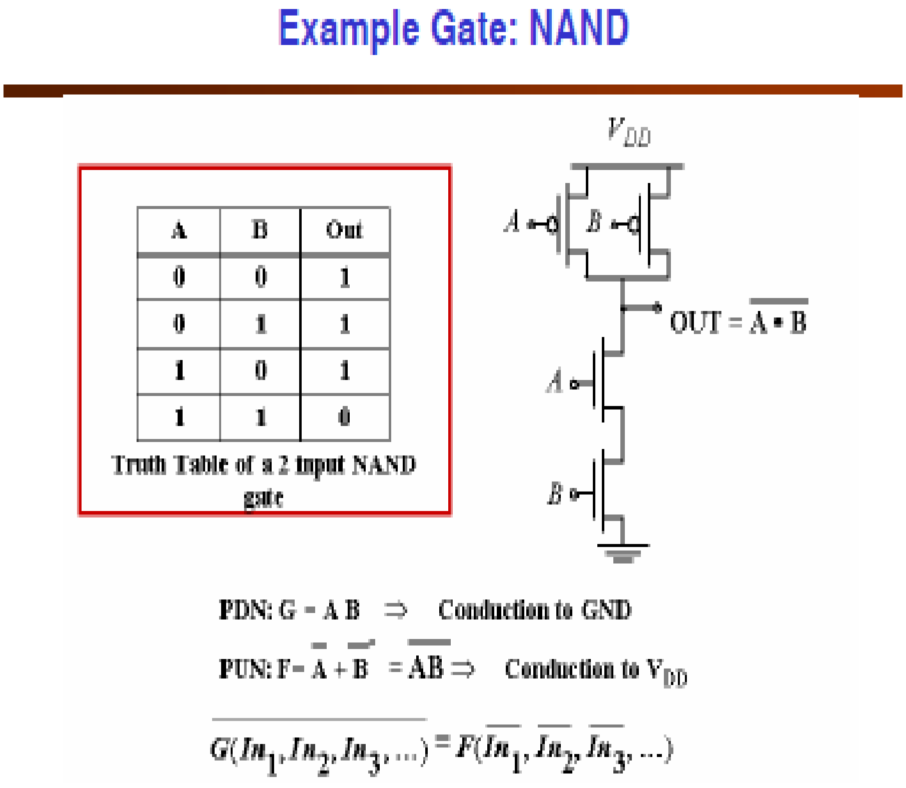


## Example Gate: NOR

| A | B | Out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Truth Table of a 2 input NOR gate

$OUT = \overline{A + B}$



## Example Gate: NAND

| A | B | Out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Truth Table of a 2 input NAND gate

$OUT = \overline{A \bullet B}$

PDN: $G = A\,B$ ⇒ Conduction to GND

PUN: $F = \bar{A} + \bar{B} = \overline{AB}$ ⇒ Conduction to $V_{DD}$

$$\overline{G(In_1, In_2, In_3, \ldots)} = F(\overline{In_1}, \overline{In_2}, \overline{In_3}, \ldots)$$

1. Design a complementary static CMOS XOR gate at the transistor level. The XOR gate Boolean expression $F$ has four literals and is $F = \bar{x}y + x\bar{y}$.

$F$ is the sum of two product terms. The design steps are:

1. Derive the nMOS transistor topology with four transistors, one per literal in the Boolean expression. The transistors driven bv $\bar{x}$ and $v$ are connected in series. as well as the devices driven by $x$ and $\bar{y}$. These transistor groups are connected in parallel, since they are additive in the Boolean function. The signals and their complements are generated using inverters (not shown). The nMOS transistor net is shown in Fig. 5.2.

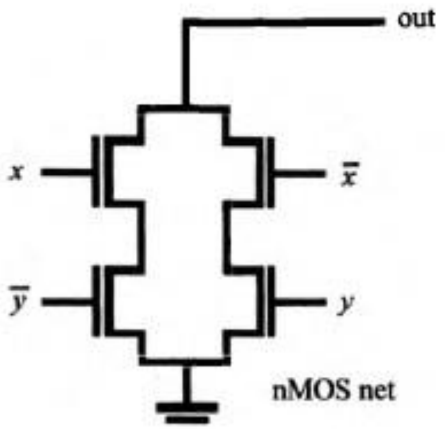


Fig. 5.2

2. Implement the pMOS net as a dual topology to the nMOS net. The pMOS transistors driven by $\bar{x}$ and $y$ are connected in parallel, as are the devices driven by $x$ and $\bar{y}$ (Fig. 5.3). These transistor groups are connected in series, since they are parallel connected in the nMOS net. The *out* node now implements $\bar{F}$.

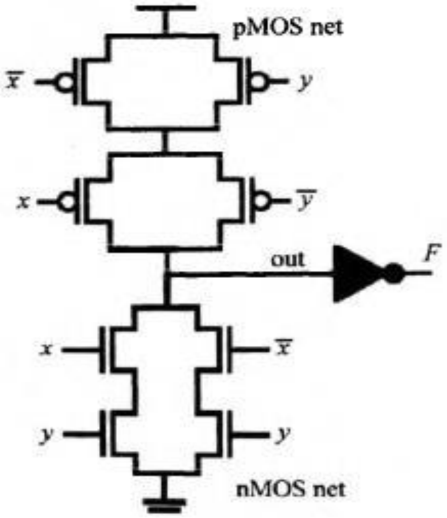


Fig. 5.3

3. Finally, add an inverter to obtain the function $F$, so that $F = \overline{out}$.

2. Design the nMOS transistor net for a Boolean function $F = x + \{\bar{y} \cdot [z + (t \cdot \bar{w})]\}$. **We design** this gate with a top-down approach. The nMOS transistor network is connected between the output and ground terminals, i.e., the lower box in Fig. 5.4(b). The higher-level function $F$ is a sum of two terms:

$F = x +$ {operation A} where operation A stands for the logic within the brackets of $F$. The transistor version of this sum is shown in Fig. 5.4(a).

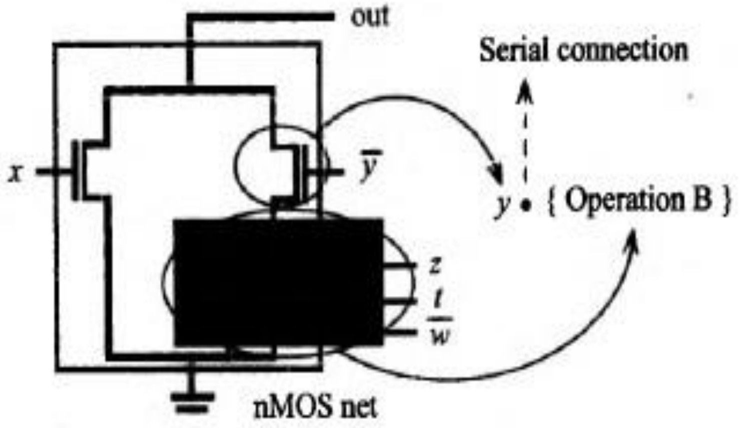


Fig. 5.4(a)

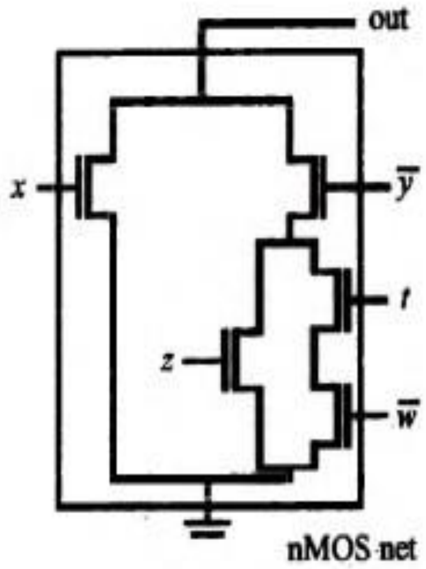
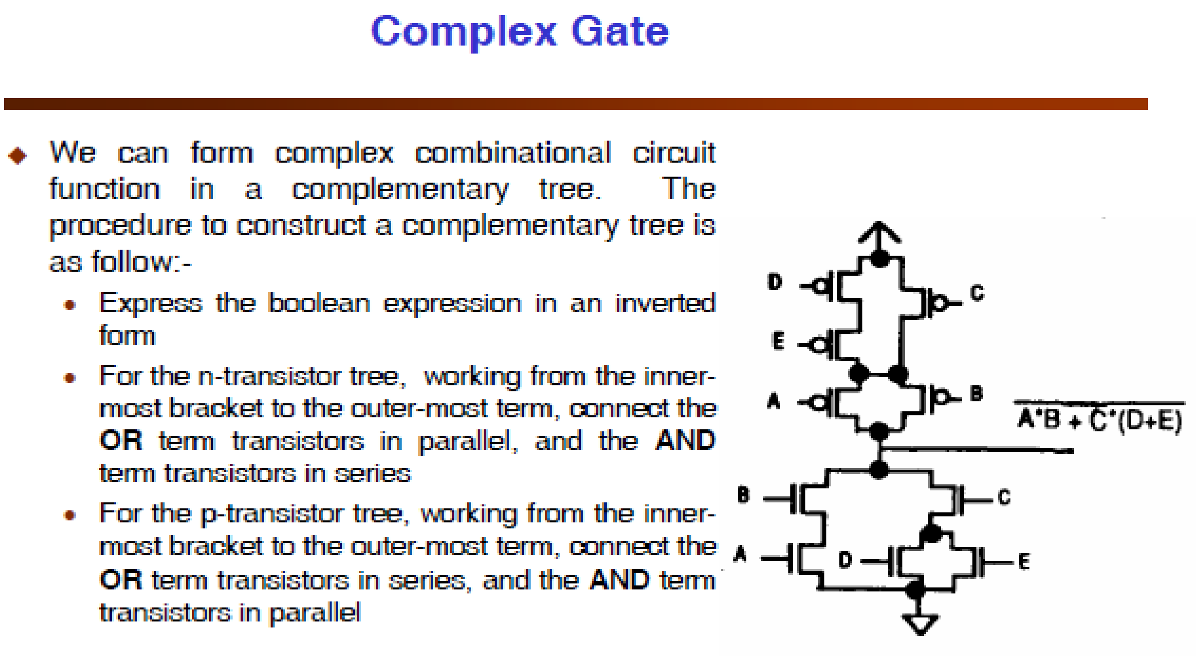


Fig. 5.4(b)

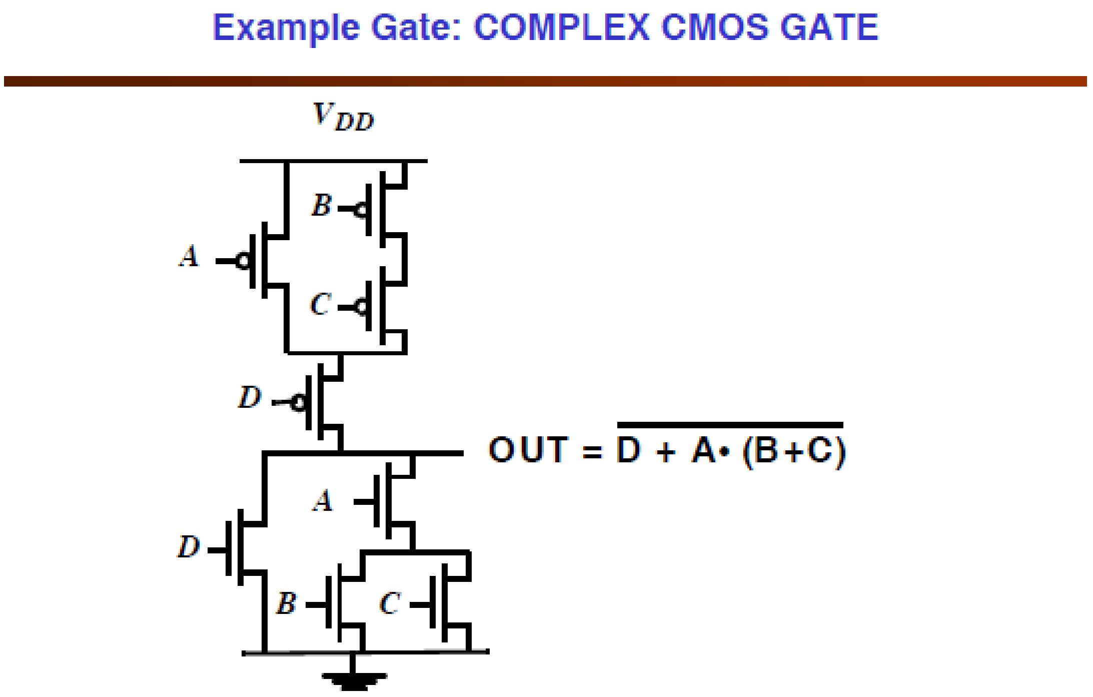
Hence, the design topology is a transistor controlled by input $\bar{y}$ in series with a third box that will implement *operation B,* as shown in Fig. 5.4. We then design the topology of box B. This is a transistor controlled by input $z$, in parallel with two transistors connected in series; one controlled by input $t$, and the other by input $\bar{w}$. The complete nMOS network is shown in Fig. 5.4(b). Once the nMOS block is designed, we build the pMOS block with a dual topological structure and then connect an inverter to its output, as shown in Fig. 5.6.

## Complex Gates:



### Complex Gate

- We can form complex combinational circuit function in a complementary tree. The procedure to construct a complementary tree is as follow:-
  - Express the boolean expression in an inverted form
  - For the n-transistor tree, working from the inner-most bracket to the outer-most term, connect the **OR** term transistors in parallel, and the **AND** term transistors in series
  - For the p-transistor tree, working from the inner-most bracket to the outer-most term, connect the **OR** term transistors in series, and the **AND** term transistors in parallel



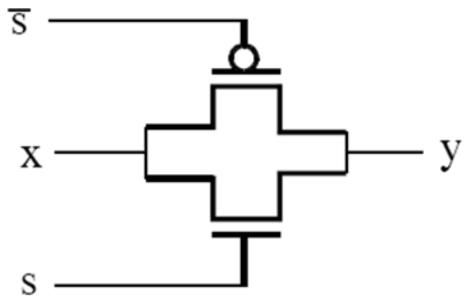
### Example Gate: COMPLEX CMOS GATE

$OUT = \overline{D + A \cdot (B+C)}$

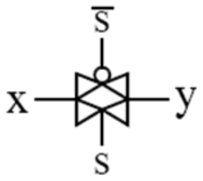
## Transmission gate logic:

A **transmission gate** is an electronic element. It is a good non-mechanical relay, built with CMOS technology. Sometimes known as an analog gate, analog switch or electronic relay depending on its use. It is made by the parallel combination of an nMOS and a pMOS transistor with the input at the gate of one transistor being complementary to the input at the gate of the other.

A *transmission gate* is a essentially a switch that connects two points. In order to pass 0's and 1's equally well, a pair of transistors (one N-Channel and one P-Channel) is used as shown below:



Circuit



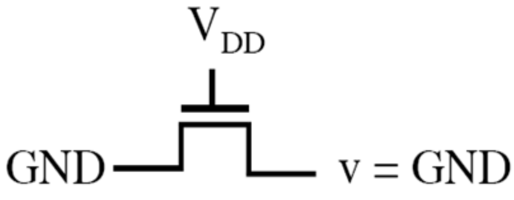
Symbol

When s = 1 the two transistors conduct and connect x and y

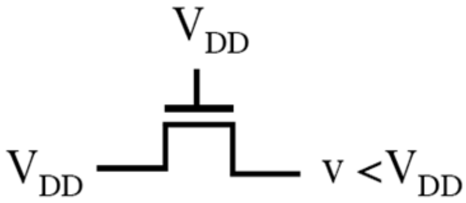
The top transistor passes x when it is 1 and the bottom transistor passes x when it is 0

When s = 0 the two transistor are cut off disconnecting x and y

N-Channel MOS Transistors pass a 0 better than a 1

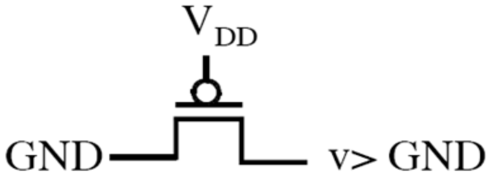


Passing 0

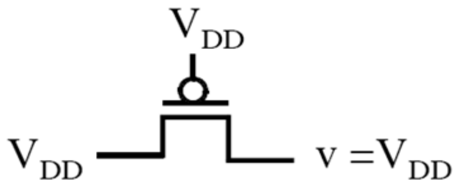


Passing 1

P-Channel MOS Transistors pass a 1 better than a 0



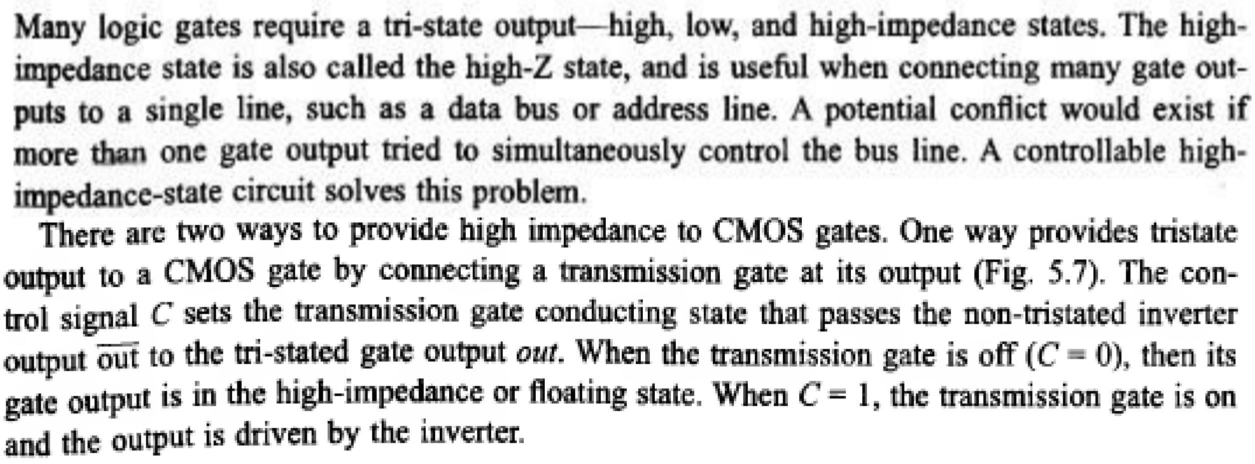
Passing 0



Passing 1

This is the reason that N-Channel transistors are used in the pull-down network and P-Channel in the pull-up network of a CMOS gate. Otherwise the noise margin would be significantly reduced.

## Tristate gates:



Many logic gates require a tri-state output—high, low, and high-impedance states. The high-impedance state is also called the high-Z state, and is useful when connecting many gate outputs to a single line, such as a data bus or address line. A potential conflict would exist if more than one gate output tried to simultaneously control the bus line. A controllable high-impedance-state circuit solves this problem.

There are two ways to provide high impedance to CMOS gates. One way provides tristate output to a CMOS gate by connecting a transmission gate at its output (Fig. 5.7). The control signal *C* sets the transmission gate conducting state that passes the non-tristated inverter output $\overline{\text{out}}$ to the tri-stated gate output *out*. When the transmission gate is off ($C = 0$), then its gate output is in the high-impedance or floating state. When $C = 1$, the transmission gate is on and the output is driven by the inverter.

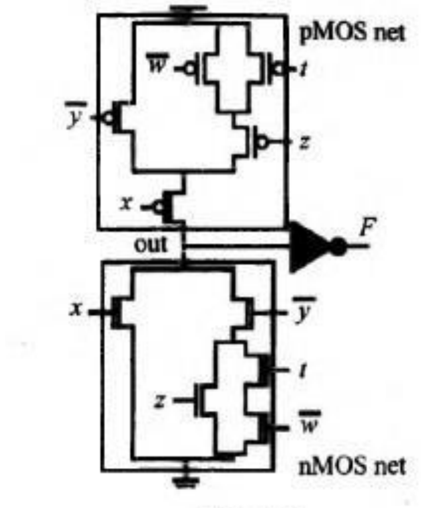
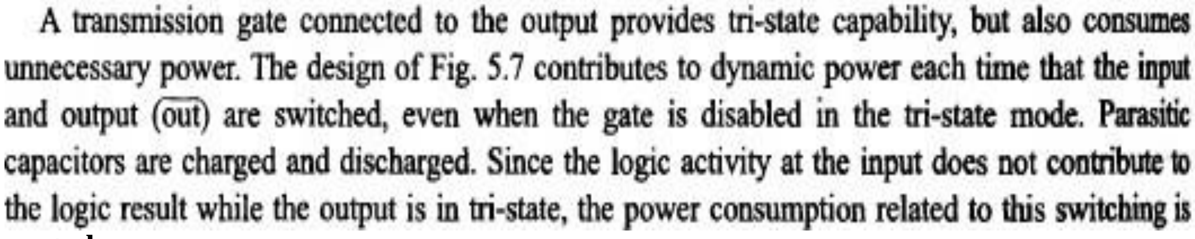


Fig. 5.6



A transmission gate connected to the output provides tri-state capability, but also consumes unnecessary power. The design of Fig. 5.7 contributes to dynamic power each time that the input and output ($\overline{\text{out}}$) are switched, even when the gate is disabled in the tri-state mode. Parasitic capacitors are charged and discharged. Since the logic activity at the input does not contribute to the logic result while the output is in tri-state, the power consumption related to this switching is **wasted.**

## Pass Transistor Logic

Pass Transistor Logic (PTL) describes several logic families used in the design of integrated circuits. It reduces the count of transistors used to make different logic gates, by eliminating redundant transistors.

Advantages are the low number of transistors and the reduction in associated interconnects. The drawbacks are the limited driving capability of these gates and the decreasing signal strength when cascading gates. These gates do not restore levels since their outputs are driven from the inputs, and not from $V_{DD}$ or ground.

A typical CMOS design is the gate-level multiplexer (MUX) shown in Fig. 5.9 for a 2-to-1 MUX

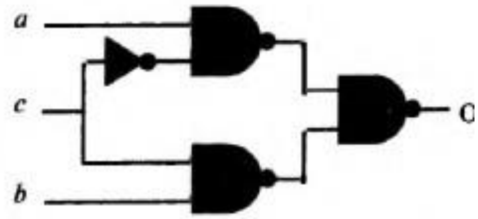
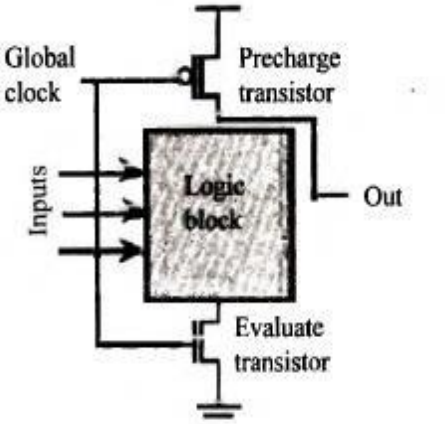


Fig. 5.9 (a) Standard 2-to-1 MUX design. (

## Dynamic CMOS logic:



**Basic Structure of a dynamic CMOS gate**

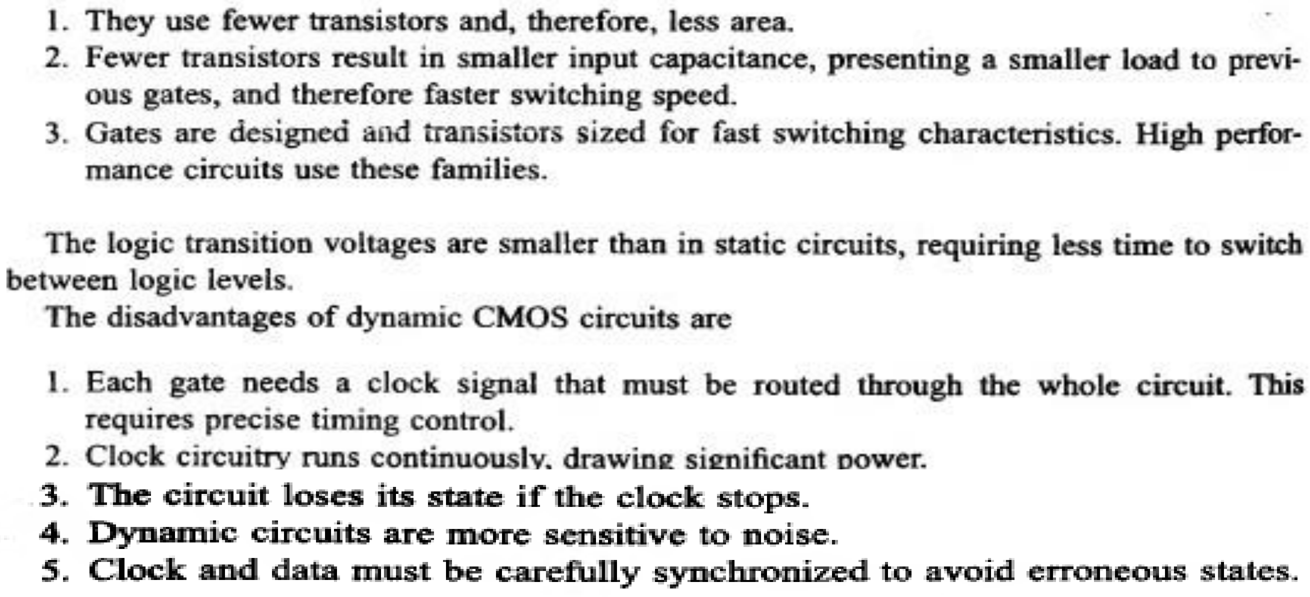
This logic looks into enhancing the speed of the pull up device by precharging the output node to Vdd. Hence we need to split the working of the device into precharge and evaluate stage for which we need a clock. Hence it is called as dynamic logic. The output node is precharged to

Vdd by the pmos and is discharged conditionally through the nmos. Alternatively you can also have a p block and precharge the n transistor to $V_{SS}$. When the clock is low the precharge phase occurs. The path to Vss is closed by the nmos i.e. the ground switch. The pull up time is improved because of the active pmos which is already precharged. But the pull down time increases because of the ground switch.

There are a few problems associated with the design, like

- Inputs have to change during the precharge stage and must be stable during the evaluate. If this condition cannot occur then charge redistribution corrupts the output node.
- A simple single dynamic logic cannot be cascaded. During the evaluate phase the first gate will conditionally discharge but by the time the second gate evaluates, there is going to be a finite delay. By then the first gate may precharge.

## Merits and Demerits:



1. They use fewer transistors and, therefore, less area.
2. Fewer transistors result in smaller input capacitance, presenting a smaller load to previous gates, and therefore faster switching speed.
3. Gates are designed and transistors sized for fast switching characteristics. High performance circuits use these families.

The logic transition voltages are smaller than in static circuits, requiring less time to switch between logic levels.

The disadvantages of dynamic CMOS circuits are

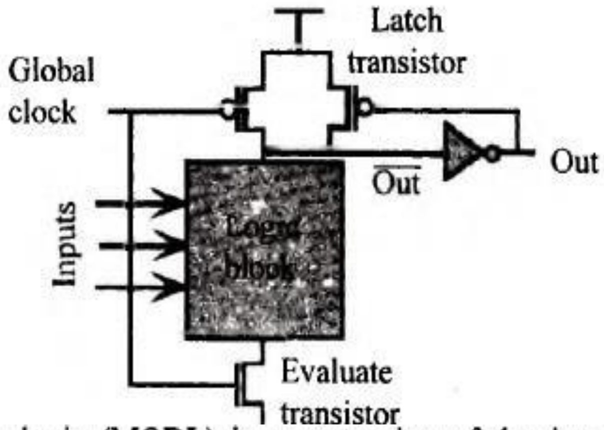
1. Each gate needs a clock signal that must be routed through the whole circuit. This requires precise timing control.
2. Clock circuitry runs continuously, drawing significant power.
3. The circuit loses its state if the clock stops.
4. Dynamic circuits are more sensitive to noise.
5. Clock and data must be carefully synchronized to avoid erroneous states.

## Domino CMOS Logic

This logic is the most common form of dynamic gates, achieving a 20%-50% performance increase over static logic. When the nMOS logic block discharges the out node during evaluation (Fig. 5.12), the inverter output out goes high, turning off the feedback pMOS. When out is evaluated high (high impedance in the dynamic gate), then the inverter output goes low, turning on the feedback pMOS device and providing a low impedance path to $V_{DD}$, This prevents the out node from floating, making it less sensitive to node voltage drift, noise and current leakage.

Domino CMOS allows logic gate cascading since all inputs are set to zero during precharge, avoiding erroneous evaluation from different delays. This logic allows static operation from the feedback latching pMOS, but logic evaluation still needs two sub cycles: precharge and evaluation.

Domino logic uses only non-inverting gates, making it an incomplete log family. To achieve inverted logic, a separate inverting path running in parallel with the non inverted one must be designed.



Multiple output domino logic (MODL) is an extension of domino logic, taking internal nodes of the logic block as signal outputs, thus saving area, power, and performance. Compound domino logic is another design that limits the length of the evaluation logic to prevent charge sharing, and adds other complex gates as buffer elements (NAND, NOR, etc. instead of inverters) to obtain more area compaction. Self-resetting domino logic (SRCMOS) has each gate detect its own operating clock, thus reducing clock overhead and providing high performance.

**NORA CMOS Logic.** This design alternative to domino CMOS logic eliminates the output buffer without causing race problems between clock and data that arise when cascading dynamic gates. NORA CMOS (No-Race CMOS) avoids these race problems by cascading alternate nMOS and pMOS blocks for logic evaluation. The cost is routing two complemented clock signals. The cascaded NORA gate structure is shown in Fig. 5.13. When the global clock (*GC*) is low ($\bar{GC}$ high), the nMOS logic block output nodes are precharged high, while outputs of gates with pMOS logic blocks are precharged low. When the clock changes, gates are in the evaluate state.

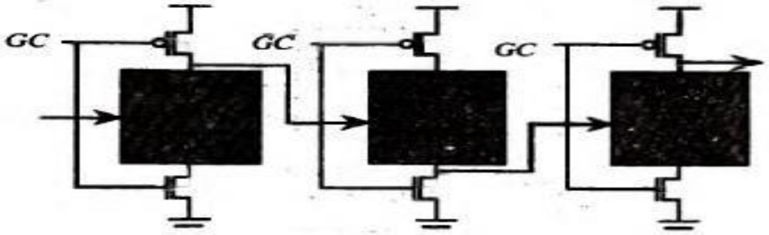
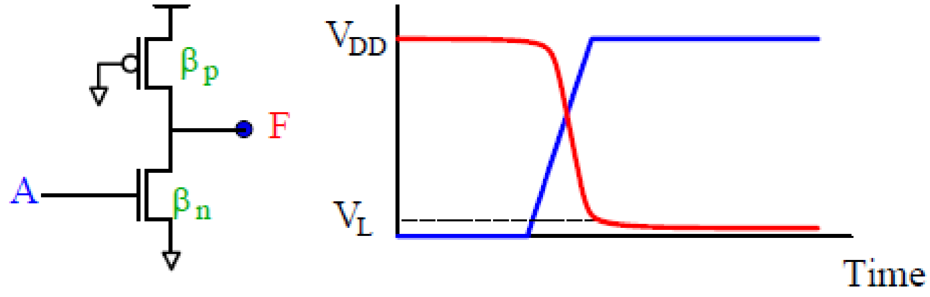


**Fig. 5.13** NORA CMOS cascaded gates.

## Pseudo – NMOS Logic:

### pseudo-NMOS inverter



The inverter that uses a p-device pull-up or loads that has its gate permanently ground. An n-device pull-down or driver is driven with the input signal. This roughly equivalent to use of a depletion load is **NMOS** technology and is thus called '**Pseudo-NMOS**'. The circuit is used in a variety of CMOS **logic** circuits.

The low output voltage can be calculated as

$$\beta_n(V_{DD}-V_{tn})V_L = \frac{\beta_P}{2}(V_{DD}-|V_{tp}|)^2$$

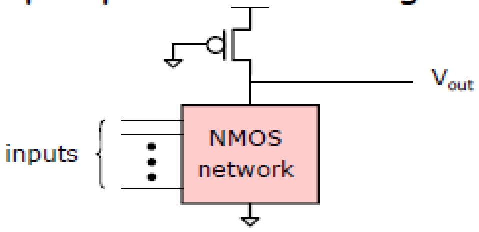
for $V_{tn} = -V_{tp} = V_t$

$$V_L = \frac{\beta_P}{2\beta_n}(V_{DD}-V_T)$$

Thus $V_L$ depends strongly on the ratio $\beta_p / \beta_n$

The logic is also called ratioed logic

An N-input pseudo-NMOS gate



Features of pseudo-NMOS logic

- Advantages
  - □ Low area cost→only N+1 transistors are needed for an N-input gate
  - □ Low input gate-load capacitance→$C_{gn}$
- Disadvantage
  - □ Non-zero static power dissipation

Unit-3 Gate level design & Basic circuit concepts

## Basic Circuit Concepts:

### Sheet Resistance Rs and its concepts to MOS

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

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

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

**Model:**

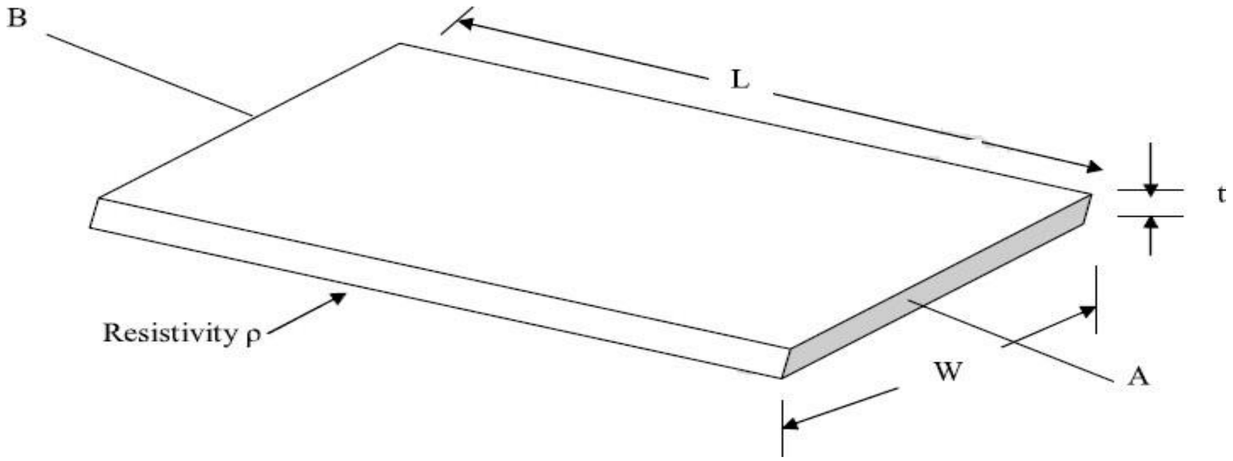
Consider a uniform slab of conducting material of resistivity ρ, of width W, thickness t, and length between faces L as shown below:

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

Where A = cross section area.

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

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



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

Where $R_s$ =ohm per square or sheet resistance.

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

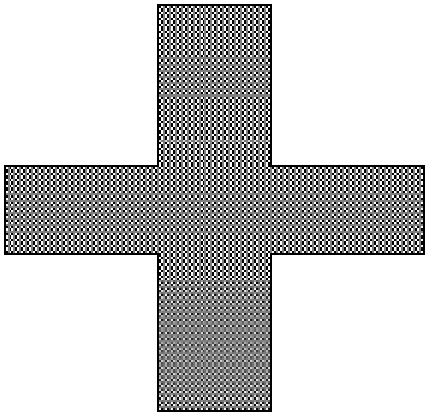
It is completely independent of the area of the square.

**Typical sheet resistance Rs of MOS layers**

| Layer | $R_s$ ohm per square | | |
|---|---|---|---|
| | 5µm | Orbit | 1.2µm |
| Metal | 0.03 | 0.04 | 0.04 |
| Diffusion | $10 \rightarrow 50$ | $20 \rightarrow 45$ | $20 \rightarrow 45$ |
| Silicide | $2 \rightarrow 4$ | - | - |
| Polysilicon | $15 \rightarrow 100$ | $15 \rightarrow 30$ | $15 \rightarrow 30$ |
| n-transistor channel | $10^4$ | $2 \text{ X } 10^4$ | $2 \text{ X } 10^4$ |
| p-transistor channel | $2.5 \text{ X } 10^4$ | $4.5 \text{ X } 10^4$ | $4.5 \text{ X } 10^4$ |

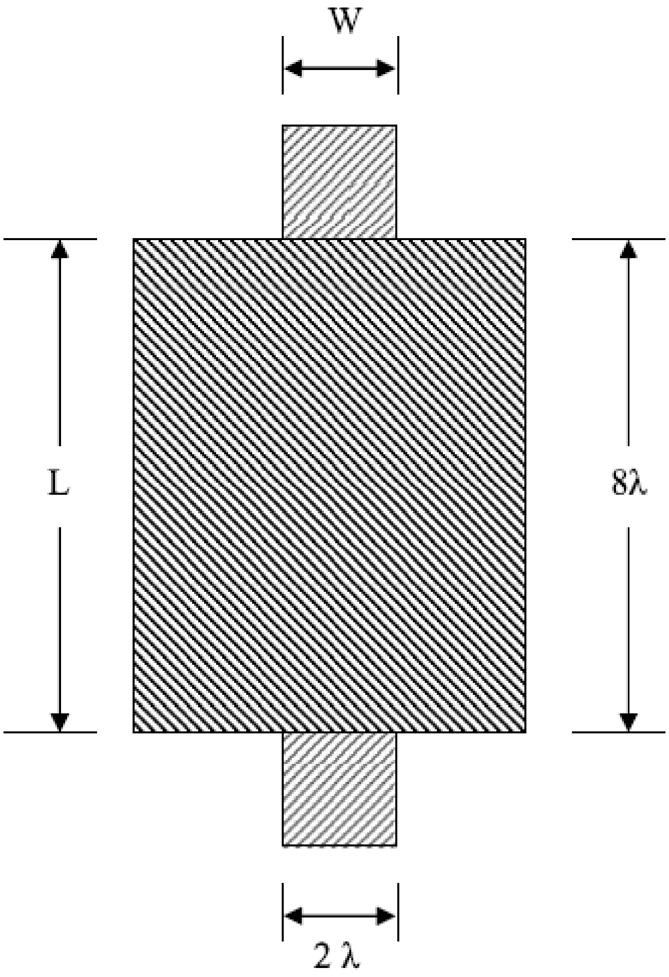
## SHEET RESISTANCE CONCEPT APPLIED TO MOS TRANSISTORS AND INVERTERS

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



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

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



$L = 8\ \lambda$ and $W = 2\ \lambda$

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

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

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

## Calculation of ON Resistance of a Simple Inverter

**Consider the simple nMOS inverter in Fig.**

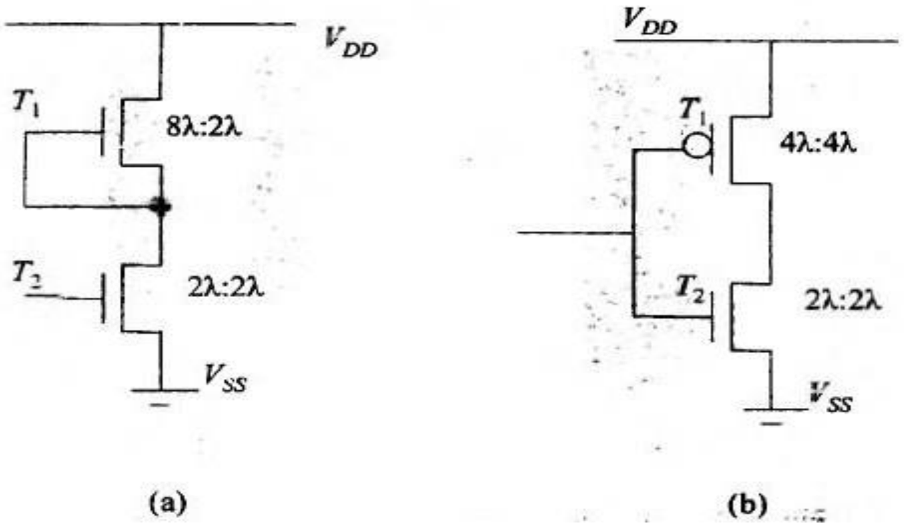


**Fig. (a) NMOS Inverter (b) CMOS Inverter resistance calculations**

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

Consider the simple CMOS inverter in Fig.

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

## Area Capacitances calculations

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

Area capacitance can be calculated as $C = \dfrac{\varepsilon_o \varepsilon_{ins} A}{D}$ farads

Where

$D$ = Thickness of silicon dioxide

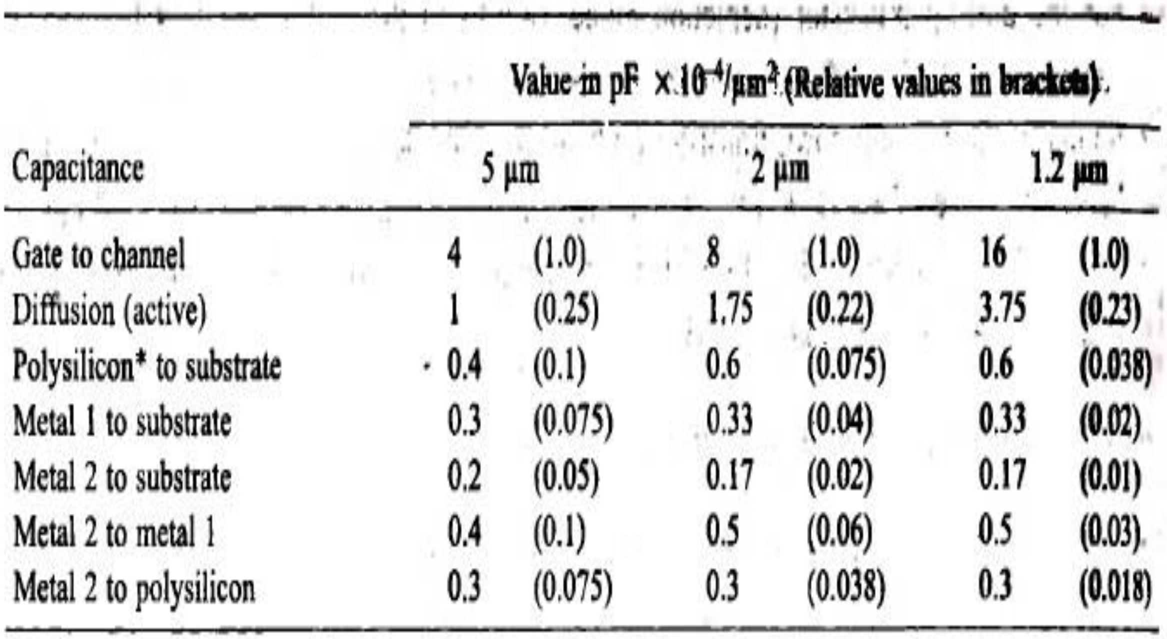
$A$ = Area of plates

$\varepsilon_{ins}$ = Relative permittivity of $SiO_2$ = 4.0

$\varepsilon_o$ = 8.85 X $10^{-14}$ F/cm (permittivity of free space)

The layer area capacitance is in pF/µm$^2$ (where µm = micron = $10^{-6}$ meter)

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



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

## Standard unit of capacitance:

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

$C_g$ may be evaluated for any MOS process.

For example, for 5µm MOS circuits
Area/standard square = 5µm X 5µm = 25µm$^2$

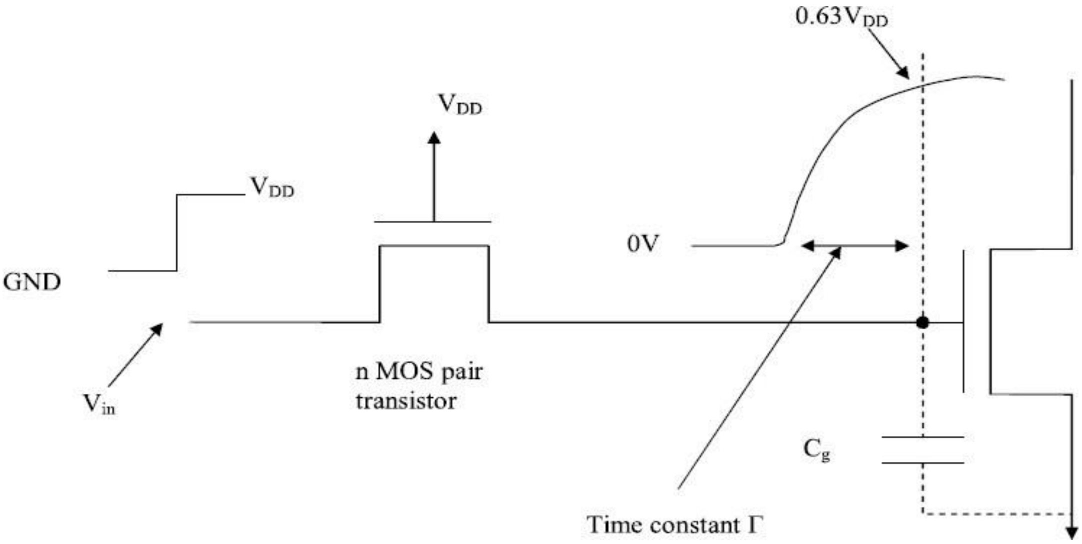
Capacitance value $= 4 \text{ X } 10^{-4}$ pF/µm$^2$
Thus standard value of $C_g = 25$ µm$^2$ X 4 X $10^{-4}$ pF/µm$^2$
$= 0.01$ pF

For 2 µm MOS circuits $C_g = 0.0032$ pF and for 1.2 µm MOS circuits $C_g = 0.0023$ pF

## Calculation of Delay unit τ

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

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



For 5µm technology
$\Gamma = 10^4$ ohm X 0.01 pF
$= 0.1$ n sec

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

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

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

$$\tau \approx \tau_{sd} = \frac{L^2}{\mu_n V_{ds}} = \frac{25\ \mu m^2 V \sec}{650\ cm^2 \quad 3V} X \frac{10^9\ n \sec cm^2}{10^8\ \mu m^2}$$
$= 0.13$ n sec

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

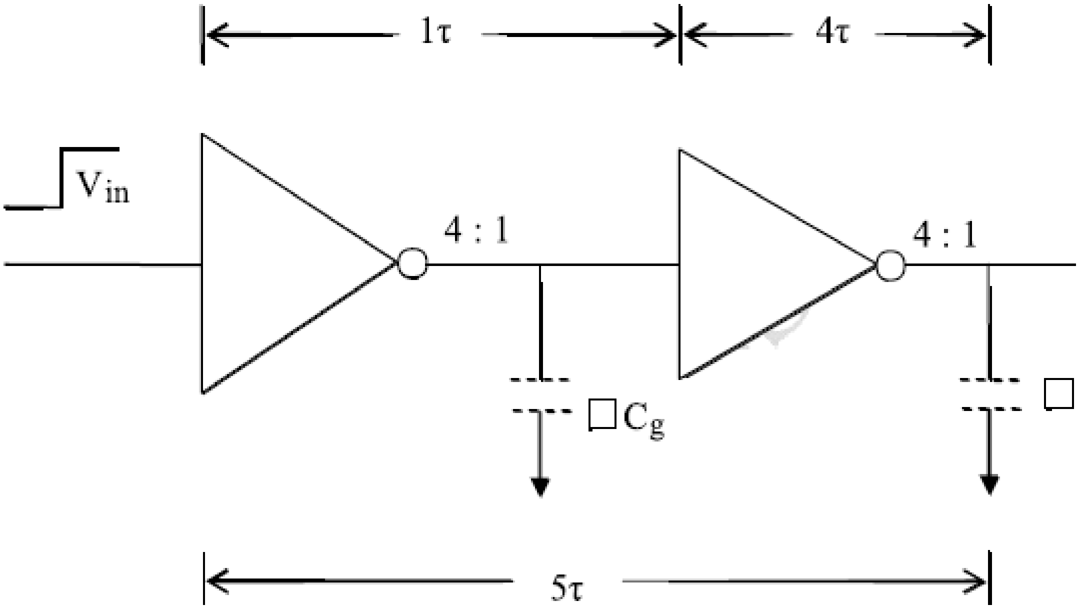
## Inverter Delays:

### nMOS Inverter Pair Delay

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

$R_{pu} = 4\ R_s = 40k\Omega$
Meanwhile $R_{pd} = 1R_s = 10k\Omega$

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



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

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

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

$2\,C_g$ Because the input is connected to both transistor gates.

## *Minimum Size CMOS Inveter Pair Delay*

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

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

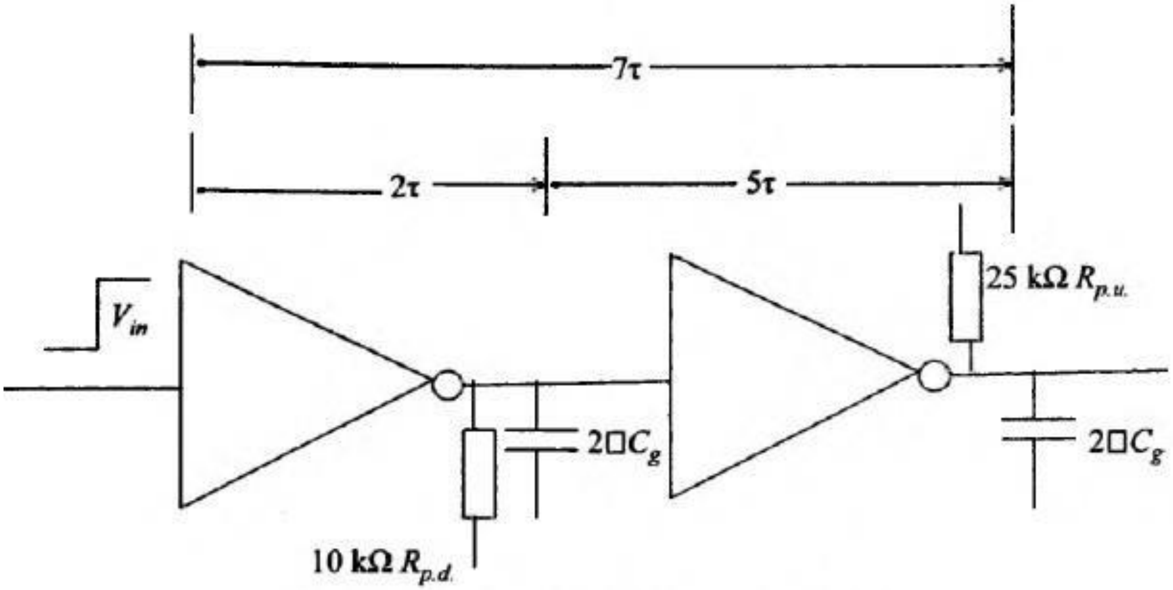


**Fig. 5.21** Minimum size CMOS inverter pair delay.

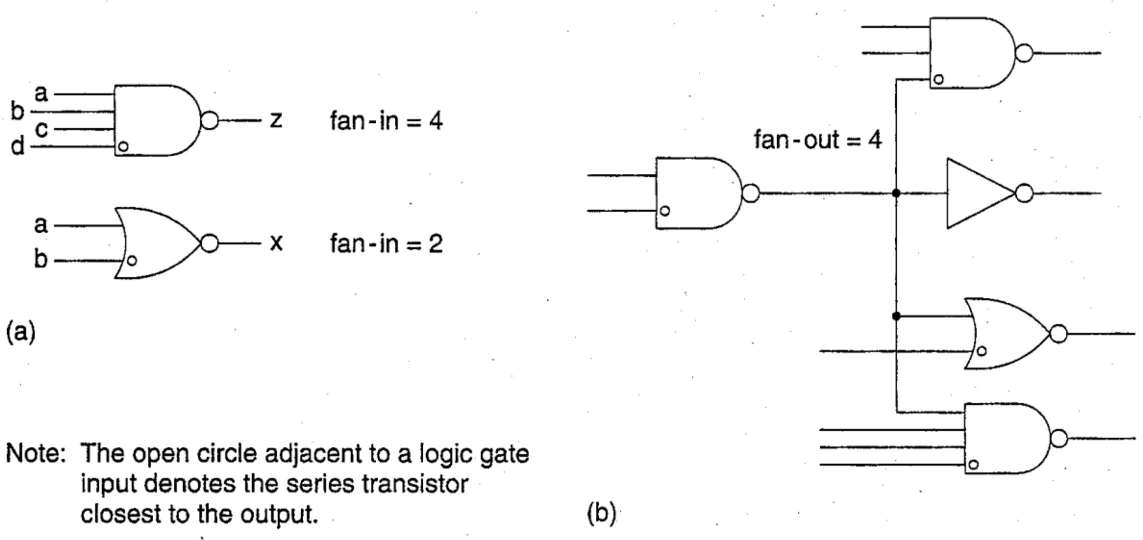
## Fan in and Fan out:

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

Ex. m-input NAND: **tdr = (Rp/n)(mnCd + Cr + kCg)**

**= tinternal-r + k toutput-r**

where n = width multiplier, m = fan-in, k = fan-out, Rp = resistance of min inverter P Tx, Cg = gate capacitance, Cd = source/drain capacitance, Cr = routing (wiring) capacitance.



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

•The circuit fall delay can be written in a similar manner.

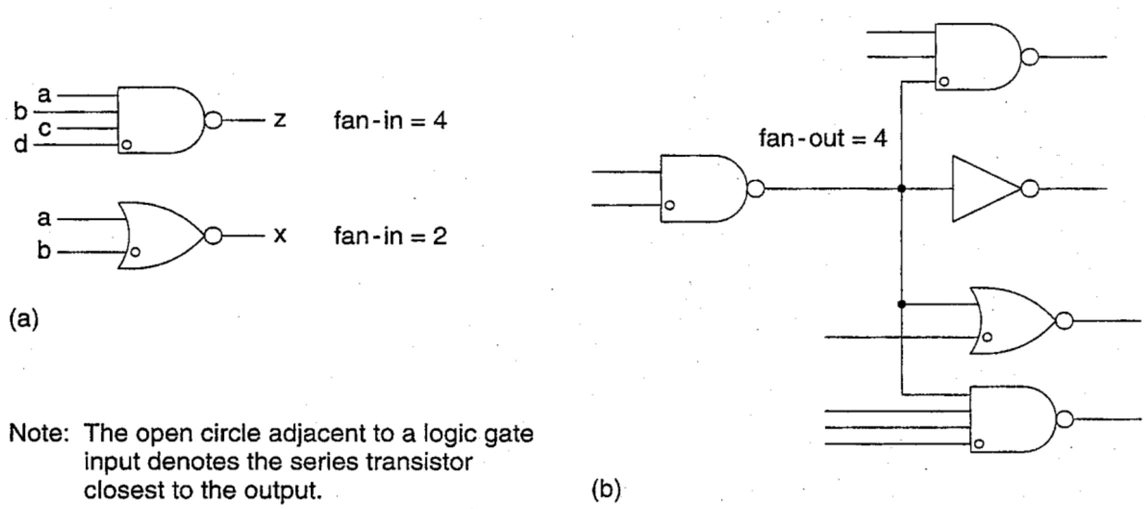
Ex. m-input NAND: $\mathbf{t_{df} = m(R_n/n)(mnC_d + C_r + kC_g)}$

$$= \mathbf{t_{internal\text{-}f} + k\, t_{output\text{-}f}}$$

where n = width multiplier, m = fan-in, k = fan-out, Rn = resistance of min inverter NMOS Tx, Cg = gate capacitance, Cd = source/drain capac, Cr = routing (wiring) capac.

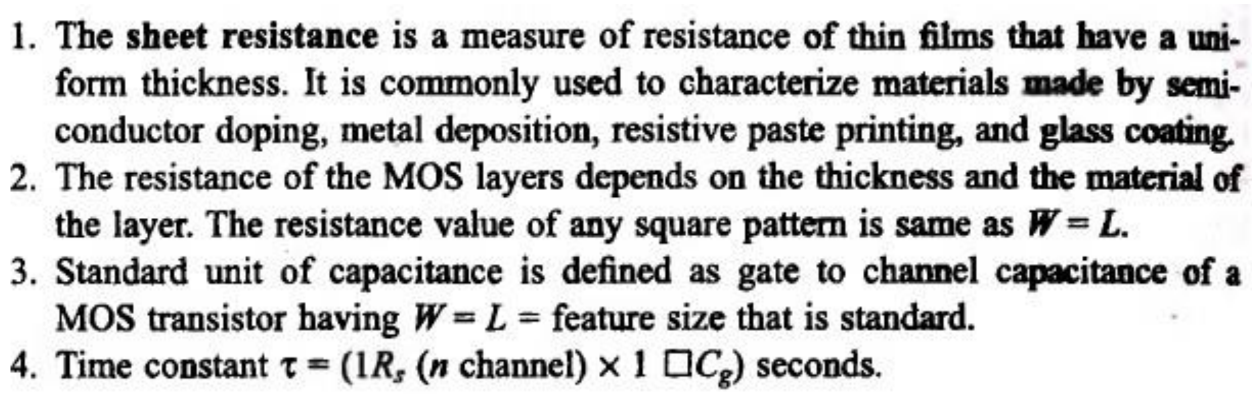
If we set $t_{dr} = t_{df}$ for the case of symmetrical rise and fall delay, we obtain that Rp = m Rn and therefore,

$$\boldsymbol{\beta_p W_p = (\beta_n W_n)/m}$$



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

## Summary



1. The **sheet resistance** is a measure of resistance of thin films that have a uniform thickness. It is commonly used to characterize materials made by semiconductor doping, metal deposition, resistive paste printing, and glass coating.
2. The resistance of the MOS layers depends on the thickness and the material of the layer. The resistance value of any square pattern is same as $W = L$.
3. Standard unit of capacitance is defined as gate to channel capacitance of a MOS transistor having $W = L =$ feature size that is standard.
4. Time constant $\tau = (1R_s\ (n \text{ channel}) \times 1\ \square C_g)$ seconds.

# Unit-IV

## Subsystem Design and VLSI Design styles

### Introduction

Most digital functions can be divided into the following categories:

1. Data path operators
2. Memory elements
3. Control structures
4. Special-purpose cells
   - I/O
   - Power distribution
   - Clock generation and distribution
   - Analog and RF

CMOS system design consists of partitioning the system into subsystems of the types listed above. Many options exist that make trade-of between speed, density, programmability, ease of design, and other variables. This chapter addresses design options for common data path operators, arrays, especially those used for memory. Control structures are most commonly coded in a hardware description language and synthesized.

Data path operators benefit from the structured design principles of hierarchy, regularity, *modularity*, and locality. They may use N identical circuits to process N-bit data. Related data operators are placed physically adjacent to each other to reduce wire length and delay. Generally, data is arranged to ow in one direction, while control signals are introduced in a direction orthogonal to the data ow.

Common data path operators considered in this chapter include adders, one/zero detectors, comparators, counters, shifters, ALUs, and multipliers.

### *4.1 Shifters*

Consider a direct MOS switch implementation of a 4X4 crossbar switch as shown in Fig. 4.1. The arrangement is quit general and may be readily expanded to accommodate n-bit inputs/outputs. In fact, this arrangement is an overkill in that any input line can be connected to any or all output lines- if all switches are closed, then all inputs are connected to all outputs in one glorious short circuit.

Furthermore, 16 control signals (sw00)-sw15, one for each transistor switch, must be provided to drive the crossbar switch, and such complexity is highly undesirable.

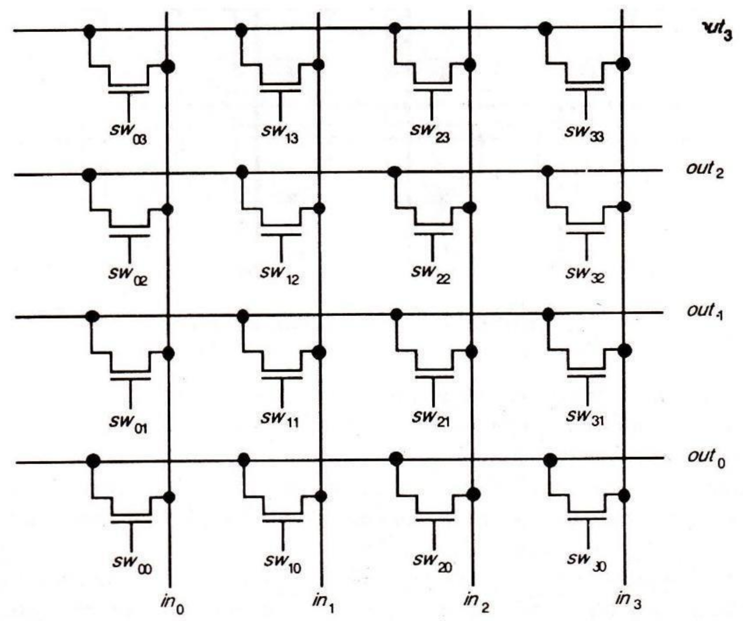


Figure 4.1: 4 x 4 crossbar switch.

An adaption of this arrangement) recognizes the fact that we can couple the switch gates together in groups of four (in this case) and also form four separate groups corresponding to shifts of zero, one, two, and three bits. The arrangement is readily adapted so that the in lines also run horizontally (to confirm the required strategy). The resulting arrangement is known as barrel shifter and a 4X4-bit barrel shifter circuit diagram is given in Fig. 4.2. The inter bus switches have their gate inputs connected in staircase fashion in group of four and there are now four shift control inputs which must be mutually exclusive in active state. CMOS transmission gates may be used in place of the simple pass transistor switches if appropriate.

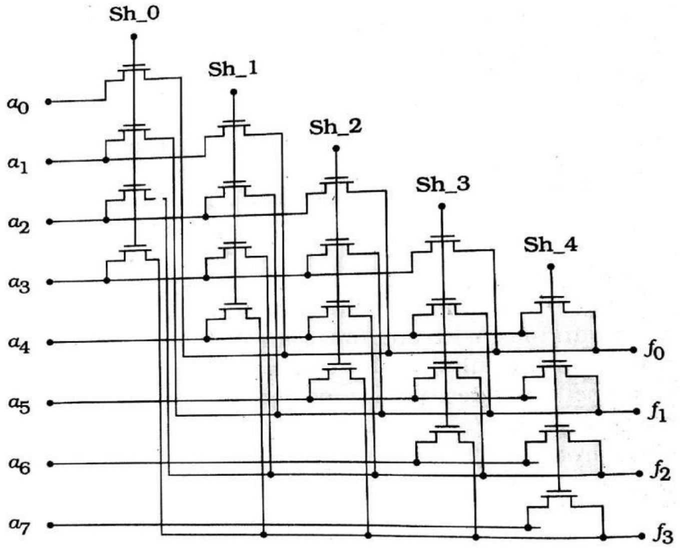
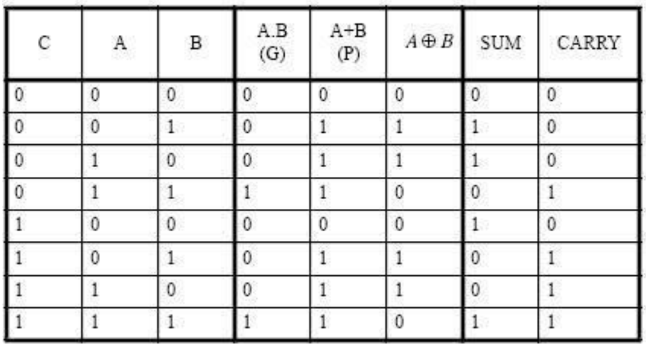


Figure 4.2: Barrel shifter

## 4.2 Adders

Addition is one of the basic operation perform in various processing like counting, multiplication and altering. Adders can be implemented in various forms to suit different speed and density requirements.

The truth table of a binary full adder is shown in Figure 4.3, along with some functions that will be of use during the discussion of adders. Adder inputs: A, B, Carry input



| C | A | B | A.B (G) | A+B (P) | $A \oplus B$ | SUM | CARRY |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

Figure 4.3: Full adder truth table

Output: SUM, Carry output: CARRY Generate signal: G (A B); occurs when CARRY is internally generated within adder.
Propagate signal: P (A + B); when it is 1, C is passed to CARRY.
In some adders A, B is used as the P term because it may be reused to generate the sum term.

### 4.3.1 Single-Bit Adders

Probably the simplest approach to designing an adder is to implement gates to yield the required majority logic functions.

From the truth table these are:

The direct implementation of the above equations is shown in Fig. 4.4 using the gate schematic and the transistors is shown in Fig. 4.5.

$$S = \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + ABC$$

$$S = C(AB + \overline{A}\,\overline{B}) + \overline{C}(\overline{A}B + A\overline{B})$$

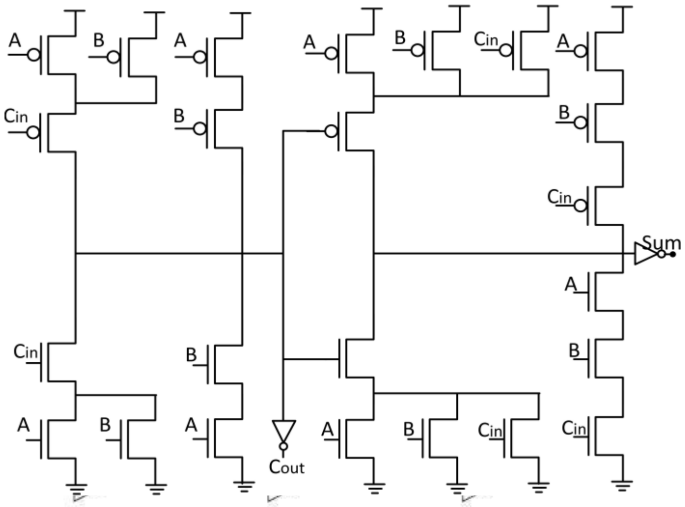
$$S = C(\overline{A \oplus B}) + \overline{C}(A \oplus B)$$

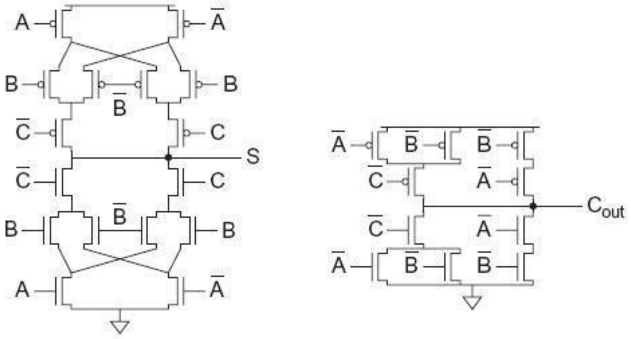
$$S = C \oplus (A \oplus B)$$

$$C(i+1) = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

$$C(i+1) = AB(\overline{C} + C) + C(\overline{A}B + A\overline{B})$$

$$C(i+1) = AB + C(A \oplus B)$$





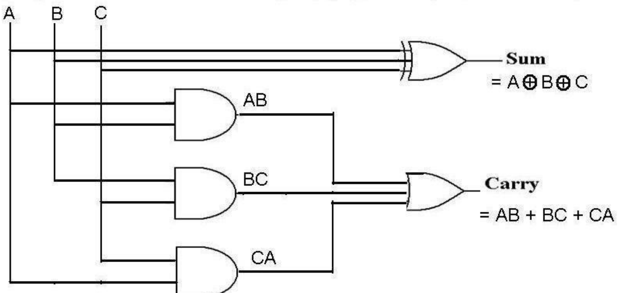


Figure 4.4: Logic gate implementation of 1-Bit adder

Figure 4.5: Transistor implementation of 1-Bit adder

The full adder of Fig. 4.5 employs 32 transistors (6 for the inverters, 10 for the carry circuit, and 16 for the 3-input XOR). A more compact design is based on the observation that S can be factored to reuse the CARRY term as follows:

For the **SUM** (S) SUM = (A XOR B) XOR Cin = (A ⊕ B) ⊕ Cin

For the **CARRY-OUT** (Cout) bit CARRY-OUT = A AND B OR Cin (A XOR B) = A.B + Cin (A ⊕ B) Such a design is shown at the transistor levels in Figure 4.6 and uses only 28 transistors. Note that the pMOS network is complement to the nMOS network.

Here Cin=C

Figure 4.6: Transistor implementation of 1-Bit adder

### 4.3.2 n-Bit Parallel Adder or Ripple Carry Adder

A ripple carry adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascaded, with the carry output from each full adder connected to the carry input of the next full adder in the chain. Figure 4.7 shows the interconnection of four full adder (FA) circuits to provide a 4-bit ripple carry adder. Notice from Figure 4.7 that the input is from the right side because the reset cell traditionally represents the least significant bit (LSB). Bits a0 and b0 in the figure represent the least significant bits of the numbers to be added. The sum output is represented by the bits S0-S3.

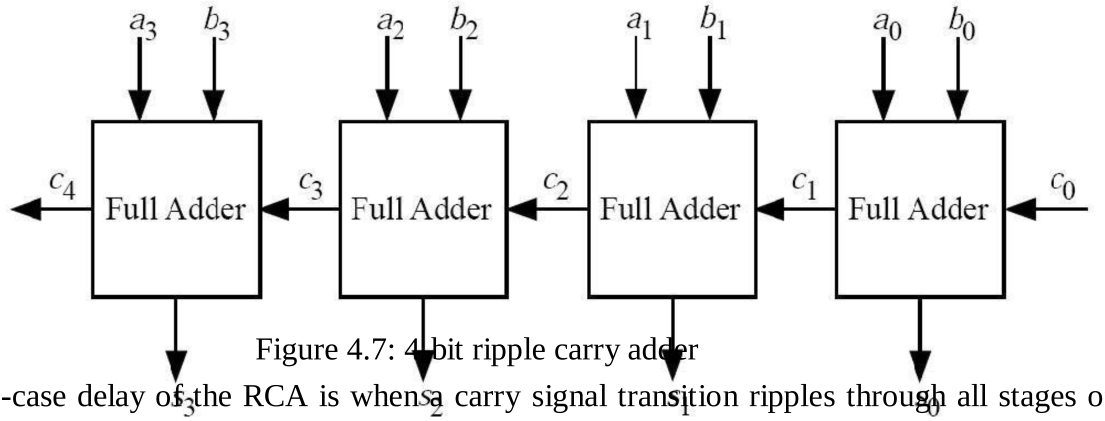


Figure 4.7: 4-bit ripple carry adder

The worst-case delay of the RCA is when a carry signal transition ripples through all stages of adder chain from the least significant bit to the most significant bit, which is approximated by:

$$t = (n - 1)t_c + t_s$$

Where tc is the delay through the carry stage of a full adder, and ts is the delay to compute the sum of the last stage. The delay of ripple carry adder is linearly proportional to n, the number of bits, therefore the performance of the RCA is limited when n grows bigger. The advantages of the RCA are lower power consumption as well as a compact layout giving smaller chip area.

### 4.3.3 Carry look ahead adder (CLA)

The carry look ahead adder (CLA) solves the carry delay problem by calculating the carry signals in advance, based on the input signals. It is based on the fact that a carry signal will be generated in two cases:

(1) When both bits $a_i$ and $b_i$ are 1, or

(2) When one of the two bits is 1 and the carry-in is 1 . Thus, one can write,

$c_{i+1} = a_i.b_i + (a_i \oplus$

$b_i).c_i$ $s_i = (a_i \oplus b_i) \oplus c_i$

The above two equations can be written in terms of two new signals $P_i$ and $G_i$, which are shown in Figure 4.8:

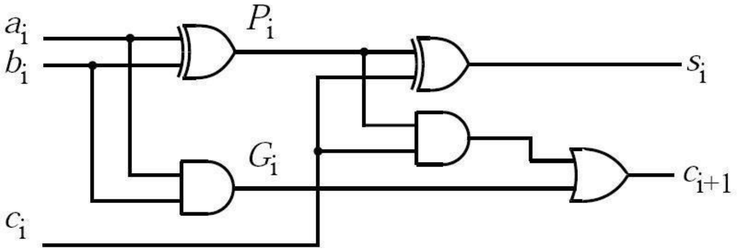


Figure 4.8: Full adder stage at i with $P_i$ and $G_i$ shown

**$c_{i+1} = G_i + P_i.c_i$, $s_i = P_i \oplus c_i$, Where $G_i = a_i.b_i$**

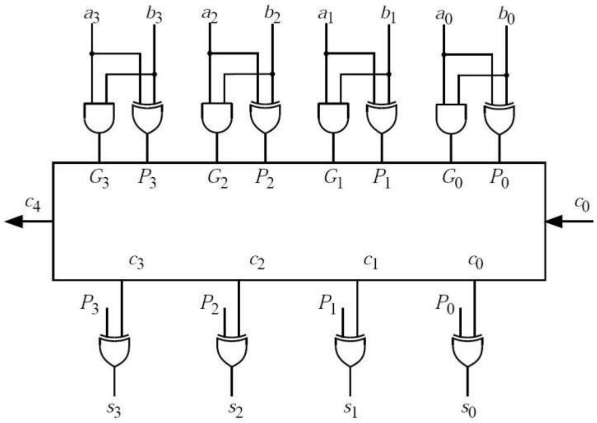
$P_i$ and $G_i$ are called carry propagate and carry generate terms, respectively. Notice that the generate and propagate terms only depend on the input bits and thus will be valid after one and two gate delay, respectively. If one uses the above expression to calculate the carry signals, one does not need to wait for the carry to ripple.

Through all the previous stages to find its proper value. Let's apply this to a 4-bit adder to make it clear.

Notice that the carry-out bit, $c_{i+1}$, of the last stage will be available after four delays: two gate delays to calculate the propagate signals and two delays as a result of the gates required to implement Equation $c_4$.

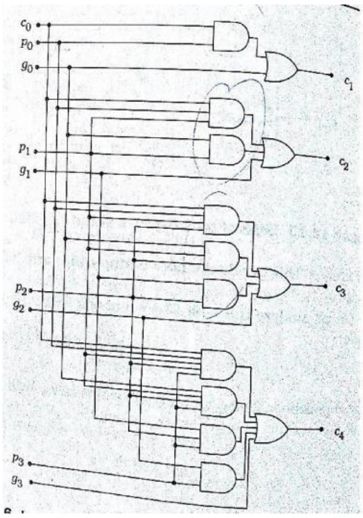
Figure 4.9 shows that a 4-bit CLA is built using gates to generate the $P_i$ and **$P_i = (a_i \oplus b_i)$** $G_i$ and signals and a logic block to generate the carry out signals according to Equations $c_1$ to $c_4$.

Figure 4.9: 4-Bit carry look ahead adder implementation in detail

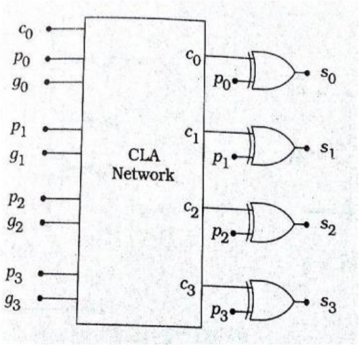


Logic gate and transistor level implementation of carry bits are shown in Figure 4.10.

The disadvantage of CLA is that the carry logic block gets very complicated for more than 4- bits. For that reason, CLAs are usually implemented as 4-bit modules and are used in a hierarchical structure to realize adders that have multiples of 4-bits.



(a) Logic network for 4-bit CLA carry bits



(b) Sum calculation using CLA network

### *4.3.4 Carry Skip Adder:*

As the name indicates, Carry Skip Adder (CSkA) uses skip logic in the propagation of carry . It is designed to speed up the addition operation by adding a propagation of carry bit around a portion of entire adder. The carry-in bit designated as Ci. The output of RCA (the last stage) is Ci+4. The Carry Skip circuitry consists of two logic gates. AND gate accepts the carry-in bit and compares it with the group of propagated signals.

**Pi, Pi+3= (Pi+3)*(Pi+2)*(Pi+1)*Pi & Carry= Ci+4 + (Pi, i+3)* Ci.** **(1)**

The architecture of CSkA is shown in Figure.

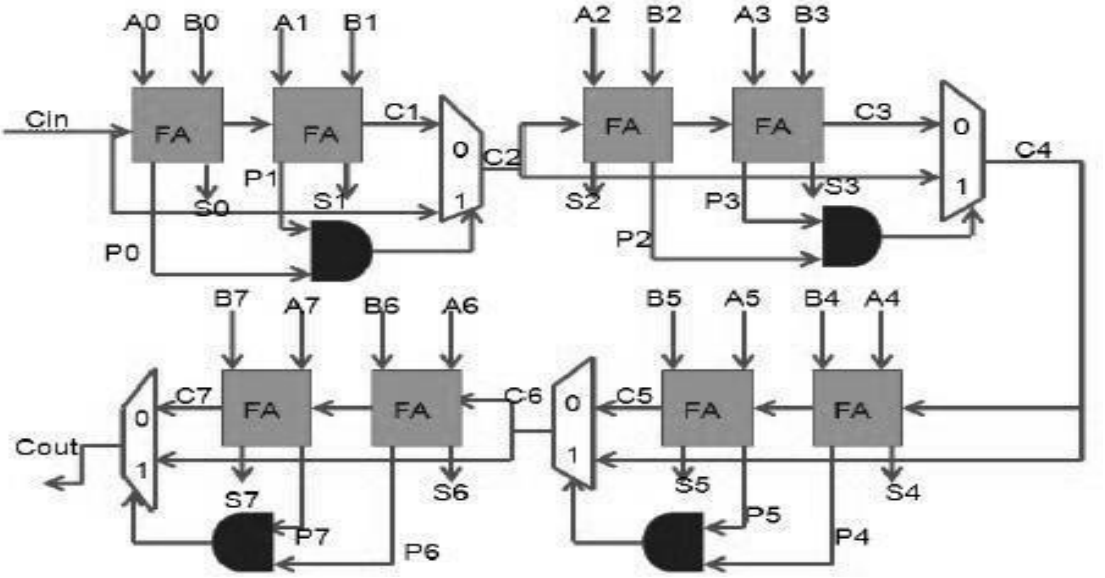


Fig. Carry Skip Adder (CSkA)

### 4.3.5 Carry Save Adder:

In Carry Save Adder (CSA), three bits are added parallelly at a time. In this scheme, the carry is not propagated through the stages. Instead, carry is stored in present stage, and updated as addend value in the next stage. Hence, the delay due to the carry is reduced in this scheme.

The architecture of CSA is shown in Fig.

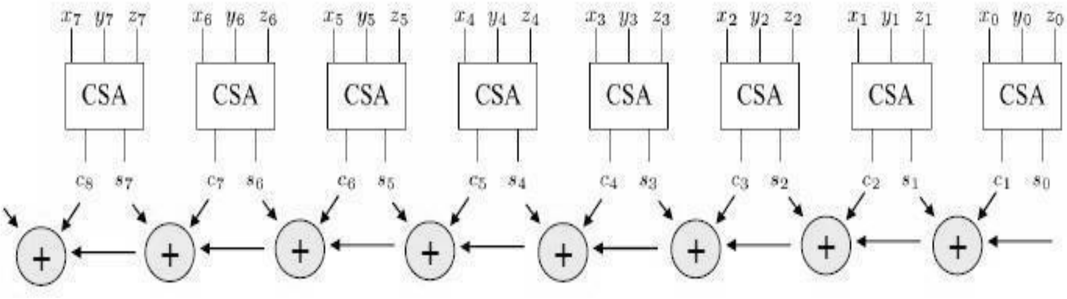


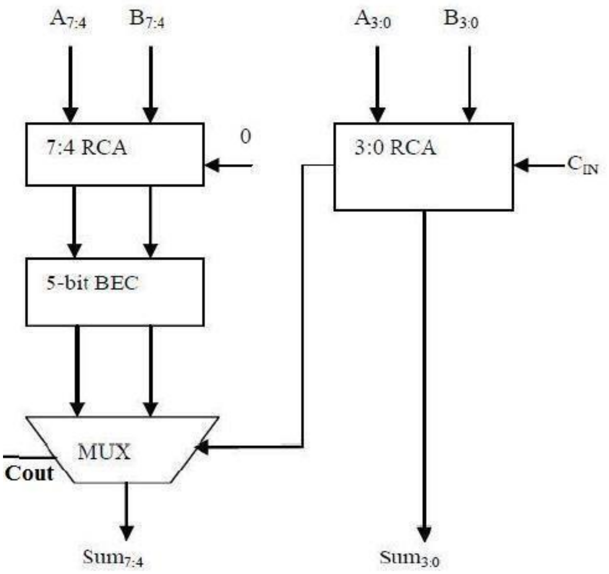
Fig. Carry save Adder (CSA)

### 4.3.6 Carry Select Adder:

Carry Select Adder (CS***l***A) architecture consists of independent generation of sum and carry i.e., Cin=1 and Cin=0 are executed parallelly [4]. Depending upon C in, the external multiplexers select the carry to be propagated to next stage. Further, based on the carry input, the sum will be selected.

Hence, the delay is reduced. However, the structure is increased due to the complexity of multiplexers [4].The architecture of CS***l***A is illustrated in Fig .

Fig. Carry Select Adder (CS***l***A)



### 4.3.7 Carry Skip (Bypass) Adder:

In Carry Bypass Adder (CBA), RCA is used to add 4-bits at a time and the carry generated will be propagated to next stage with help of multiplexer using select input as Bypass logic. By pass logic is formed from the product values as it is calculated in the CLA. Depending on the carry value and by pass logic, the carry is propagated to the next stage.

The architecture of Carry Bypass Adder (CBA) is given in Fig .

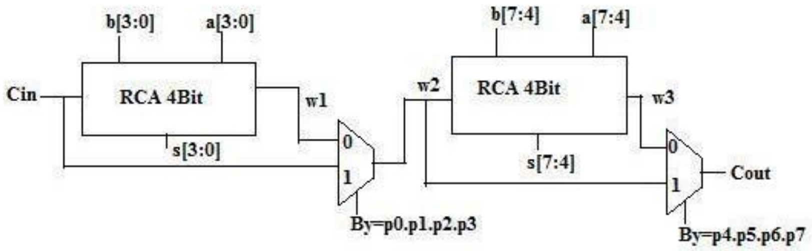


Fig. Carry Bypass Adder (CBA)

### 4.3.8 Manchester carry chain

This implementation can be very performant (20 transistors) depending on the way the XOR function is built. The carry propagation of the carry is controlled by the output of the XOR gate. The generation of the carry is directly made by the function at the bottom. When both input signals are 1, then the inverse output carry is 0. In the schem

atic of Figure 4.11, the carry passes through a complete transmission gate.

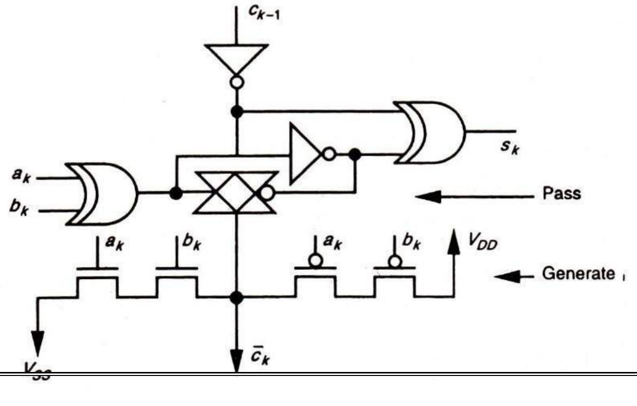


Figure 4.11: An adder element based on the pass/generate concept.

If the carry path is precharged to VDD, the transmission gate is then reduced to a simple NMOS transistor. In the same way the PMOS transistors of the carry generation is removed. One gets a Manchester cell.

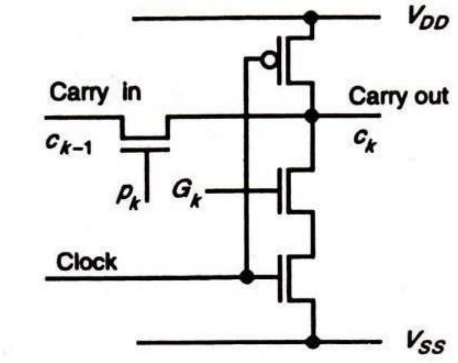


Figure 4.12: Manchester cell

The Manchester cell is very fast, but a large set of such cascaded cells would be slow. This is due to the distributed RC effect and the body effect making the propagation time grow with the square of the number of cells. Practically, an inverter is added every four cells, like in Figure 4.12.

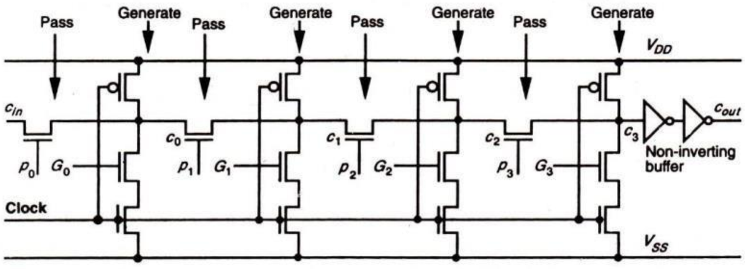


Figure 4.13: Cascaded Manchester carry-chain elements with buffering

## *4.4 Multipliers*

In many digital signal processing operations - such as correlations, convolution, filtering, and frequency analysis - one needs to perform multiplication. The most basic form of multiplication consists of forming the product of two positive binary numbers. This may be accomplished through the traditional technique of successive additions and shifts, in which each addition is conditional on one of the multiplier bits. Here is an example.

$$\begin{array}{lrl}
\text{multiplicand} & 1100 & : 12_{10} \\
\text{multiplier} & \underline{0101} & : 5_{10} \\
 & 1100 & \\
 & 0000\phantom{0} & \\
 & 1100\phantom{00} & \\
 & \underline{0000\phantom{000}} & \\
 & 0111100 & : 60_{10}
\end{array}$$

Figure 4.14: 4-bit multiplication

The multiplication process may be viewed to consist of the following two steps:

❼ Evaluation of partial products.

❼ Accumulation of the shifted partial products.

It should be noted that binary multiplication is equivalent to a logical AND operation. Thus evaluation of partial products consists of the logical ANDing of the multiplicand and the relevant multiplier bit. Each column of partial products must then be added and, if necessary, any carry values passed to the next column.

There are a number of techniques that may be used to perform multiplication. In general, the choice is based on factors such as speed, throughput, numerical accuracy, and area. As a rule, multipliers may be classified by the format in which data words are accessed, namely:-

- Serial form
- Serial/parallel form
- Parallel form

### 4.4.1 Array Multiplication (Braun Array Multiplier)

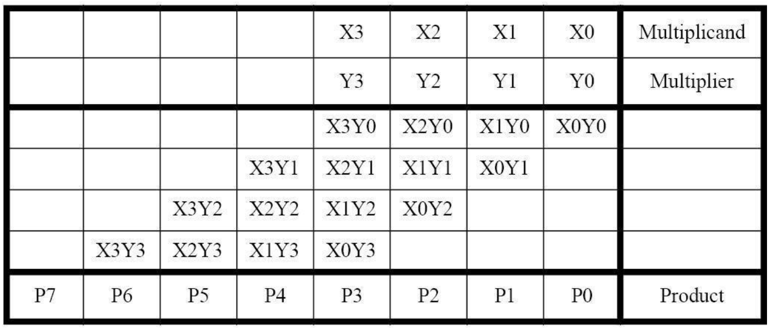
A parallel multiplier is based on the observation that partial products in the multi-plication process may be independently computed in parallel. For example, consider the unsigned binary integers X and Y.

$$X = \sum_{i=0}^{n-1} X_i 2^i \qquad Y = \sum_{j=0}^{n-1} Y_j 2^j$$

$$\begin{aligned} P = X \times Y &= \sum_{i=0}^{n-1} X_i 2^i \cdot \sum_{j=0}^{n-1} Y_j 2^j \\ &= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (X_i Y_j) 2^{i+j} \\ &= \sum_{k=0}^{n+n-1} P_k 2^k \end{aligned}$$

Thus Pk are the partial product terms called summands. There are mn summands, which are produced in parallel by a set of mn AND gates.

For 4-bit numbers, the expression above may be expanded as in the table below.



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | X3 | X2 | X1 | X0 | Multiplicand |
| | | | | Y3 | Y2 | Y1 | Y0 | Multiplier |
| | | | | X3Y0 | X2Y0 | X1Y0 | X0Y0 | |
| | | | X3Y1 | X2Y1 | X1Y1 | X0Y1 | | |
| | | X3Y2 | X2Y2 | X1Y2 | X0Y2 | | | |
| | X3Y3 | X2Y3 | X1Y3 | X0Y3 | | | | |
| P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 | Product |

Figure 4.15 An nxn multiplier requires

(n 1)$^2$full adders, n 1 half adders, and n$^2$ AND gates.

The worst-case delay associated with such a multiplier is (2n + l)tg, where tg is the worst-case adder delay.

Cell shown in Figure 4.16 is a cell that may be used to construct a parallel multiplier.

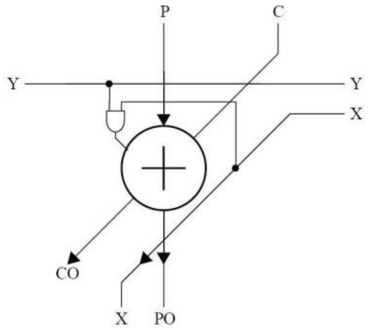


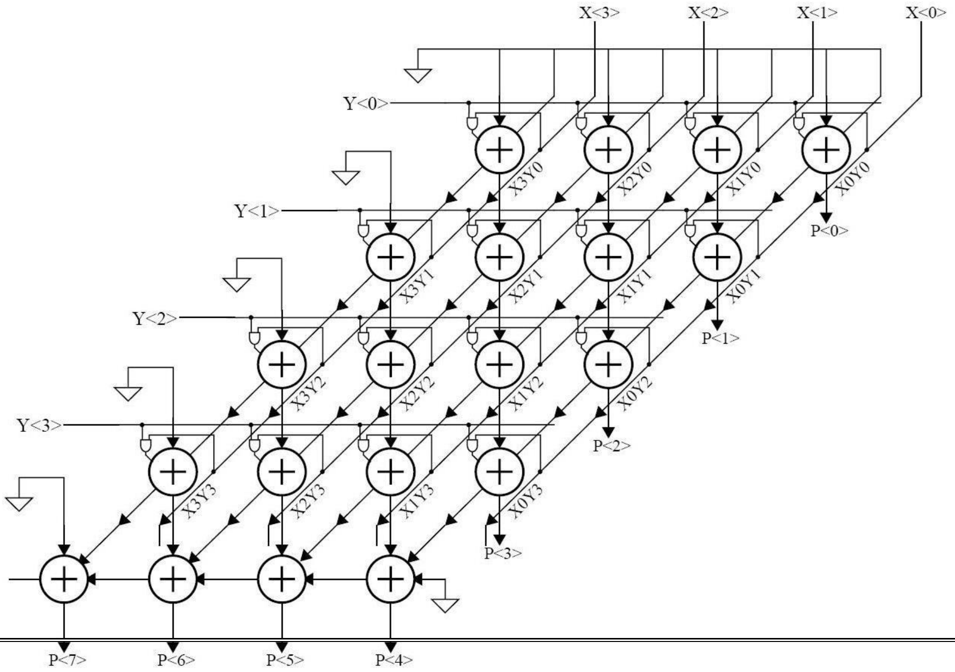
Figure 4.16: Basic cell to construct a parallel multiplier

The $X_i$ term is propagated diagonally from top right to bottom left, while the $y_j$ term is propagated horizontally. Incoming partial products enter at the top. Incoming CARRY IN values enter at the top right of the cell. The bit-wise AND is performed in the cell, and the SUM is passed to the next cell below. The CARRY 0UT is passed to the bottom left of the cell.

Figure 4.17 depicts the multiplier array with the partial products enumerated. The Multiplier can be drawn as a square array, as shown here, Figure 4.18 is the most convenient for implementation. In this version the degeneration of the first two rows of the multiplier are shown. The first row of the multiplier adders has been replaced with AND gates while the second row employs half-adders rather than full adders.

This optimization might not be done if a completely regular multiplier were required (i.e. one array cell). In this case the appropriate inputs to the first and second row would be connected to ground, as shown in the previous slide. An adder with equal carry and sum propagation times is advantageous, because the worst-case multiply time depends on both paths.

Figure 4.17: Array multiplier



### 4.4.2 Wallace Tree Multiplication

If the truth table for an adder, is examined, it may be seen that an adder is in effect a \one's counter" that counts the number of l's on the A, B, and C inputs and encodes them on the SUM and CARRY outputs.

A l-bit adder provides a 3:2 (3 inputs, 2 outputs) compression in the number of bits. The addition of partial products in a column of an array multiplier may be thought of as totaling up the number of l's in that column, with any carry being passed to the next column to the left.

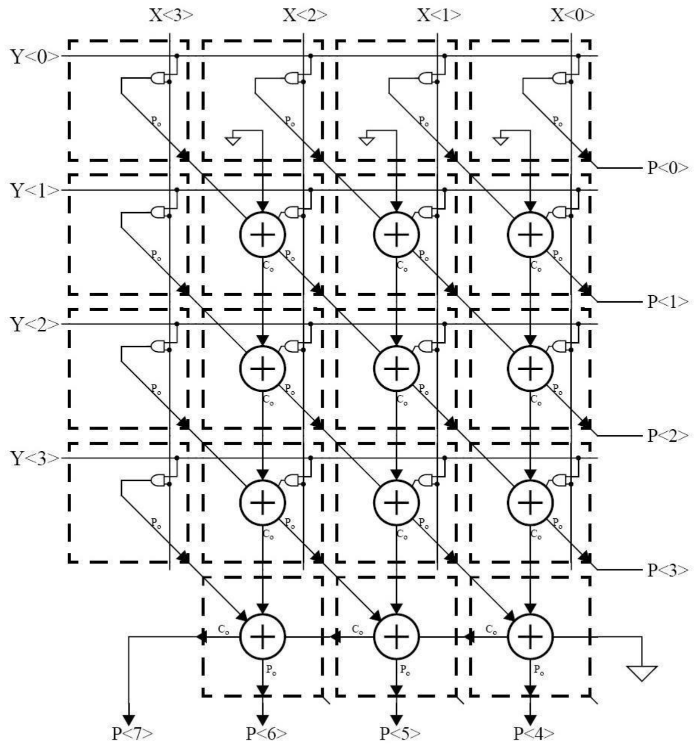
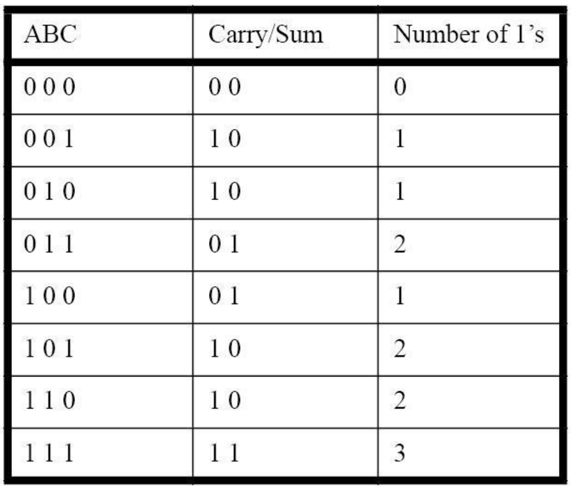
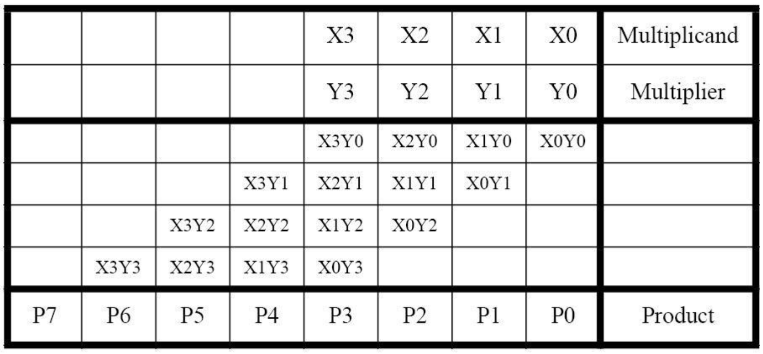


Figure 4.18: Most convenient way for implementation of array multiplier



| ABC | Carry/Sum | Number of 1's |
|---|---|---|
| 0 0 0 | 0 0 | 0 |
| 0 0 1 | 1 0 | 1 |
| 0 1 0 | 1 0 | 1 |
| 0 1 1 | 0 1 | 2 |
| 1 0 0 | 0 1 | 1 |
| 1 0 1 | 1 0 | 2 |
| 1 1 0 | 1 0 | 2 |
| 1 1 1 | 1 1 | 3 |

Example for implementation of 4x4 multiplier (4-bit) using Wallace Tree Multi-plication methods



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | X3 | X2 | X1 | X0 | Multiplicand |
| | | | | Y3 | Y2 | Y1 | Y0 | Multiplier |
| | | | | X3Y0 | X2Y0 | X1Y0 | X0Y0 | |
| | | | X3Y1 | X2Y1 | X1Y1 | X0Y1 | | |
| | | X3Y2 | X2Y2 | X1Y2 | X0Y2 | | | |
| | X3Y3 | X2Y3 | X1Y3 | X0Y3 | | | | |
| P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 | Product |

Figure 4.20: Table to nd product terms

Considering the product P3, it may be seen that it requires the summation of four partial products and a possible column carry from the summation of P2.

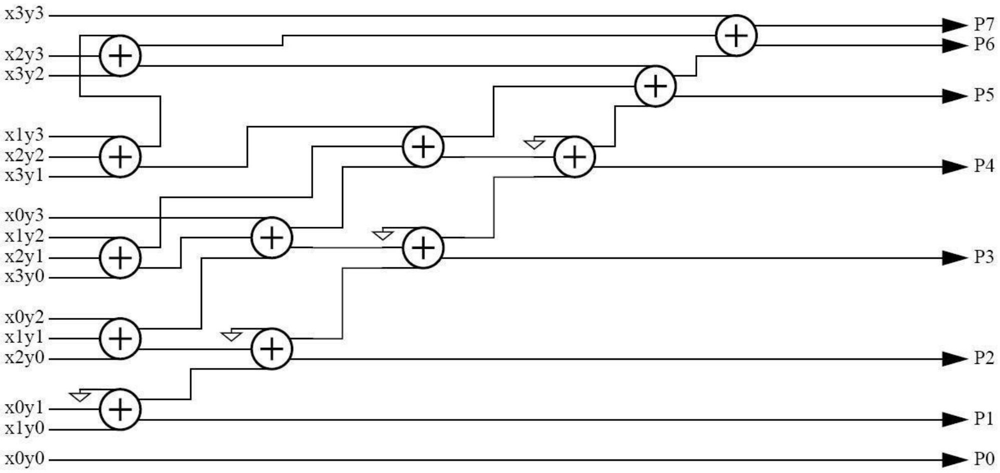


Figure 4.21: Wallace Tree Multiplication for 4-bits

Example for implementation of 6X6 multiplier (4-bit) using Wallace Tree Multi-plication methods

Consider the 6 x 6 multiplication table shown below. Considering the product P5, it may be seen that it requires the summation of six partial products and a possible column carry from the summation of P4. Here we can see the adders required in a multiplier based on this style of addition.

The adders have been arranged vertically into ranks that indicate the time at which the adder output becomes available. While this small example shows the general Wallace addition technique, it does not show the real speed advantage of a Wallace tree. There is an identity table \array part", and a CPA part, which is at the top right. While this has been shown as a ripple-carry adder, any fast CPA can be used here.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  | X5 | X4 | X3 | X2 | X1 | X0 | Multiplicand |
|  |  |  |  |  |  | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 | Multiplier |
|  |  |  |  |  |  | X5Y0 | X4Y0 | X3Y0 | X2Y0 | X1Y0 | X0Y0 |  |
|  |  |  |  |  | X5Y1 | X4Y1 | X3Y1 | X2Y1 | X1Y1 | X0Y1 |  |  |
|  |  |  |  | X5Y2 | X4Y2 | X3Y2 | X2Y2 | X1Y2 | X0Y2 |  |  |  |
|  |  |  | X5Y3 | X4Y3 | X3Y3 | X2Y3 | X1Y3 | X0Y3 |  |  |  |  |
|  |  | X5Y4 | X4Y4 | X3Y4 | X2Y4 | X1Y4 | X0Y4 |  |  |  |  |  |
|  | X5Y5 | X4Y5 | X3Y5 | X2Y5 | X1Y5 | X0Y5 |  |  |  |  |  |  |
| P11 | P10 | P9 | P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 | Product |

Figure 4.22: 6 x 6 multiplication table

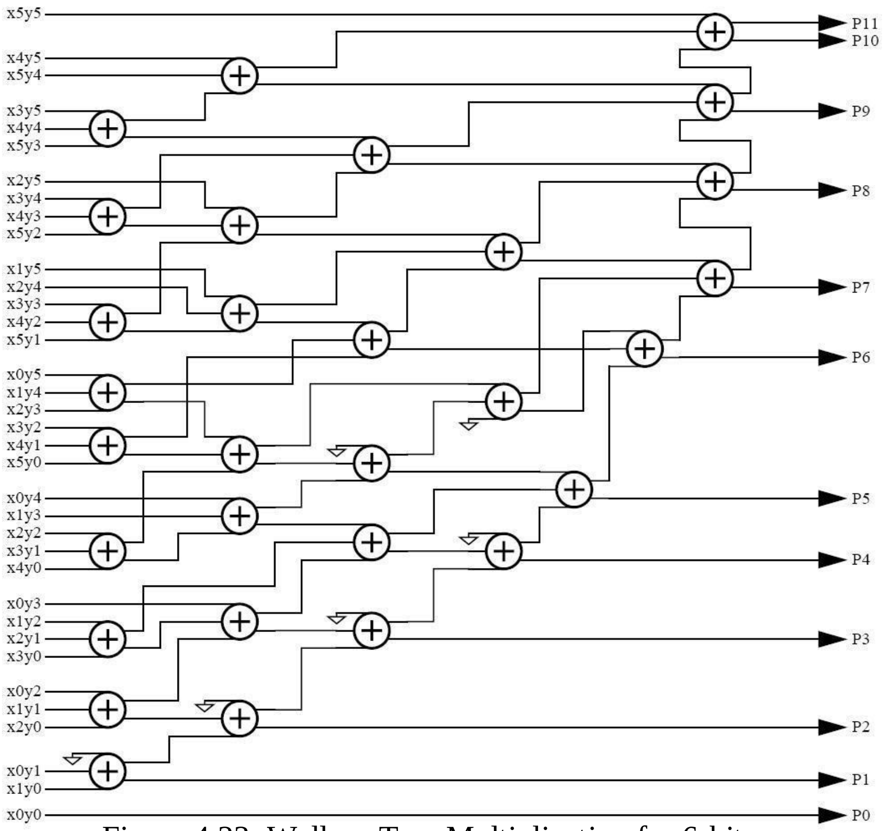


Figure 4.23: Wallace Tree Multiplication for 6-bits

The delay through the array addition (not including the CPA) is proportional to log1.5(n), where n is the width of the Wallace tree.

### *4.4.3 Baugh-Wooley multiplier:*

In signed multiplication the length of the partial products and the number of partial products will be very high. So an algorithm was introduced for signed multiplication called as Baugh- Wooley algorithm. The Baugh-Wooley multiplication is one amongst the cost-effective ways to handle the sign bits. This method has been developed so as to style regular multipliers, suited to 2's compliment numbers.

Let two n-bit numbers, number (A) and number (B), A and B are often pictured as

$$A = -\,a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i \qquad (1)$$

$$B = -b_{n-1}2^{n-1} + \sum_{i=0}^{n-2} b_i 2^i \qquad (2)$$

Where $ai$ and $bi$ area unit the bits during A and B, severally and $an$−1 and $bn$−1 area unit the sign bits. The full precision product, P = A × B, is provided by the equation:

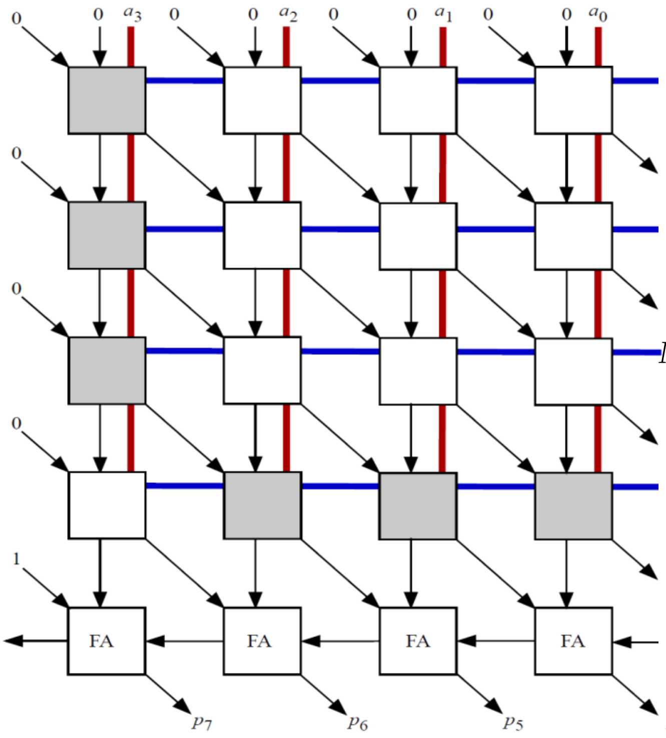
$$P = A \times B = \left[\left(-a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i\right) \times \left(-b_{n-1}2^{n-1} + \sum_{i=0}^{n-2} b_i 2^i\right)\right]$$

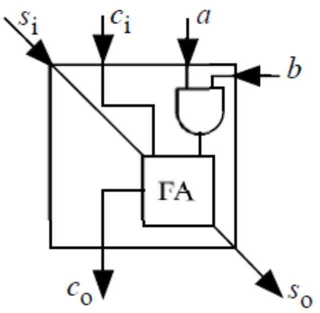
$$= a_{n-1}b_{n-1}2^{2n-2} + \sum_{i=0}^{n-1} a_i 2^i \sum_{j=0}^{n-2} b_j 2^j - 2^{n-1}\sum_{i=0}^{n-2} a_i b_{n-1} 2^i - 2^{n-1}\sum_{j=0}^{n-2} a_{n-1} b_j 2^j \qquad (3)$$

The first two terms of above equation are positive and last two terms are negative. In order to calculate the product, instead of subtracting the last two terms, it is possible to add the opposite values. The above equation signifies the Baugh-Wooley algorithm for multiplication process in two’s compliment form.

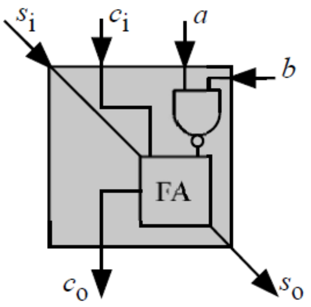
Baugh-Wooley Multiplier provides a high speed, signed multiplication algorithm. It uses parallel products to complement multiplication and adjusts the partial products to maximize the regularity of multiplication array. When number is represented in two’s complement form, sign of the number is embedded in Baugh-Wooley multiplier. This algorithm has the advantage that the sign of the partial product bits are always kept positive so that array addition techniques can be directly employed. In the two’s complement multiplication, each partial product bit is the AND of a multiplier bit and a multiplicand bit, and the sign of the partial product bits are positive.

**4-by-4 Baugh-Wooley multiplier**



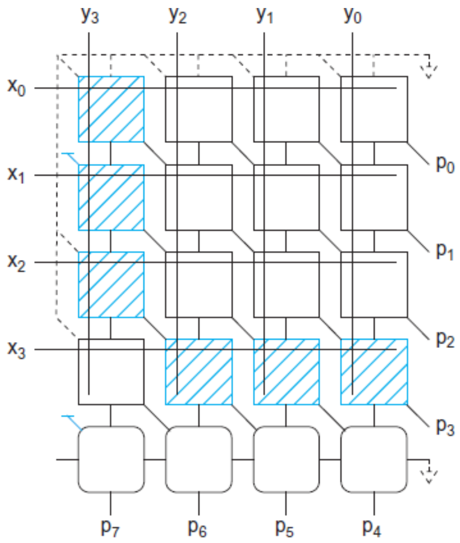


*Multiplier white cell*



*Multiplier Grey cell*

|  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  | $y_5$ | $y_4$ | $y_3$ | $y_2$ | $y_1$ | $y_0$ |
|  |  |  |  |  |  | $x_5$ | $x_4$ | $x_3$ | $x_2$ | $x_1$ | $x_0$ |
|  |  |  |  |  | 1 | $\overline{x_5y_0}$ | $x_0y_4$ | $x_0y_3$ | $x_0y_2$ | $x_0y_1$ | $x_0y_0$ |
|  |  |  |  |  | $\overline{x_5y_1}$ | $x_1y_4$ | $x_1y_3$ | $x_1y_2$ | $x_1y_1$ | $x_1y_0$ |  |
|  |  |  |  | $\overline{x_5y_2}$ | $x_2y_4$ | $x_2y_3$ | $x_2y_2$ | $x_2y_1$ | $x_2y_0$ |  |  |
|  |  |  | $\overline{x_5y_3}$ | $x_3y_4$ | $x_3y_3$ | $x_3y_2$ | $x_3y_1$ | $x_3y_0$ |  |  |  |
|  |  | $\overline{x_5y_4}$ | $x_4y_4$ | $x_4y_3$ | $x_4y_2$ | $x_4y_1$ | $x_4y_0$ |  |  |  |  |
| 1 | $x_5y_5$ | $\overline{x_4y_5}$ | $\overline{x_3y_5}$ | $\overline{x_2y_5}$ | $\overline{x_1y_5}$ | $\overline{x_0y_5}$ |  |  |  |  |  |
| $p_{11}$ | $p_{10}$ | $p_9$ | $p_8$ | $p_7$ | $p_6$ | $p_5$ | $p_4$ | $p_3$ | $p_2$ | $p_1$ |  |



```
   -12
X   12
  -144
```

$-12_{10} = 10100_2$

$12_{10} = 01100_2$

By Sign Extension method,

```
        10100
   X    01100
   000000000
   00000000
   1110100
   110100
   00000
  101110000  ──────────►  -144
```

According to the sign extend and invert algorithm,

```
        10100
   X    01100

      100000
      10000
     00100
    00100
   10000
  101110000  ──────────►  - 144
```

**Booth Multiplier:**

Booth‘s Algorithm is a smart move for multiplying signed numbers. It initiate with the ability to both add and subtract there are multiple ways to compute a product. Booth‘s algorithm is a multiplication algorithm that utilizes two‘s complement notation of signed binary numbers for multiplication.

When multiplying by 9:

- Multiply by 10 (easy, just shift digits left)
- Subtract once

E.g.123454 x 9 = 123454 x (10 – 1) = 1234540 – 123454

- Converts addition of six partial products to one shift and one subtraction

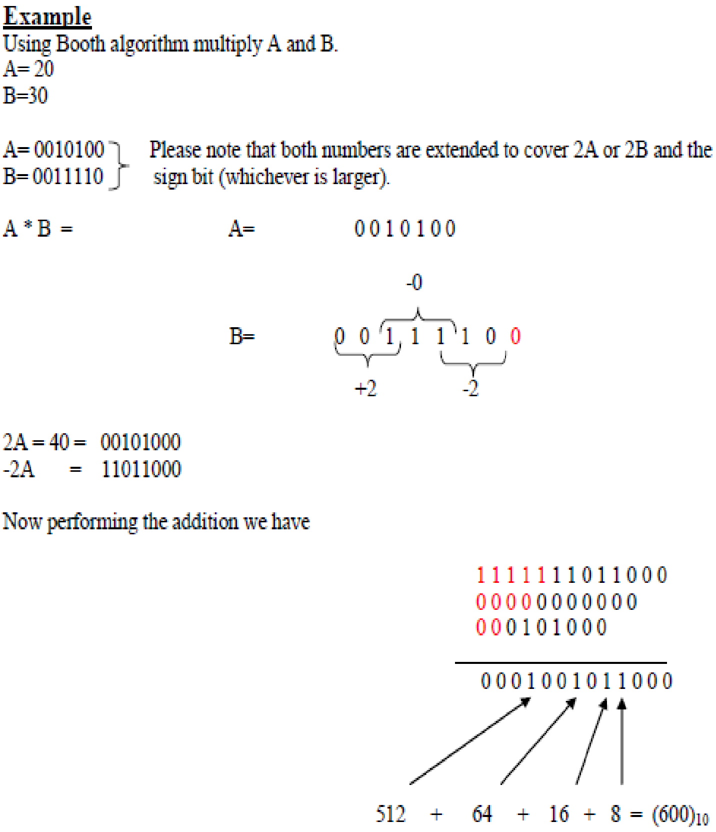
Booth's algorithm applies same principle

- Except no '9' in binary, just '1' and '0'
- So, it's actually easier!

**BOOTH ENCODER** Booth multiplier reduce the number of iteration step to perform multiplication as compare to conventional steps. Booth Algorithm Scans the multiplier operand and spikes chains of this algorithm can. This algorithm can reduce the number of addition required to produce the result compare to conventional multiplication method. With the help of this algorithm reduce the number of partially product generated in multiplication process by using the modified booth algorithm. Based on the multiplier bits, the process of encoding the multiplicand is performed by radix-4 booth encoder. This recoding algorithm is used to generate efficient partial product.

**RADIX-4 BOOTH MULTIPLIER** The Radix-4 modified Booth algorithm overcomes all these limitations of Radix-2 algorithm. For operands equal to or greater than 16 bits, the modified Radix-4 Booth algorithm has been widely used. It is based on encoding the two's complement multiplier in order to reduce the number of partial products to be added to n/2.

In Radix-4 Modified Booth algorithm, the number of partial products reduced by half. For multiplication of 2's complement numbers, the two bit encoding using this algorithm scans a triplet of bits. To Booth recode the multiplier term, consider the bits in blocks of three, such that each block overlaps the previous block by one bit. Grouping starts from the LSB, and the first block only uses two bits of the multiplier.



**Example**

Using Booth algorithm multiply A and B.
A= 20
B=30

A= 0010100
B= 0011110 } Please note that both numbers are extended to cover 2A or 2B and the sign bit (whichever is larger).

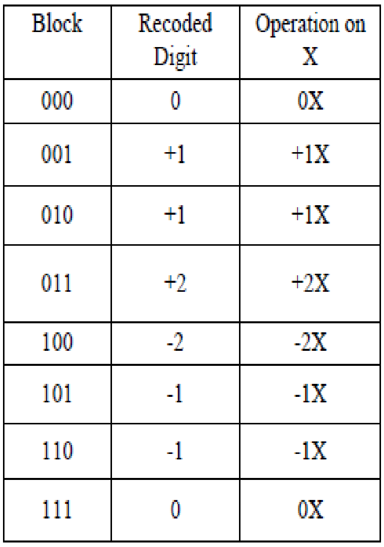
A * B = A= 0 0 1 0 1 0 0

2A = 40 = 00101000
-2A = 11011000

Now performing the addition we have

```
111111011000
00000000000
000101000
-------------
0001001011000
```

512 + 64 + 16 + 8 = $(600)_{10}$



| Block | Recoded Digit | Operation on X |
|---|---|---|
| 000 | 0 | 0X |
| 001 | +1 | +1X |
| 010 | +1 | +1X |
| 011 | +2 | +2X |
| 100 | -2 | -2X |
| 101 | -1 | -1X |
| 110 | -1 | -1X |
| 111 | 0 | 0X |

## Booth's Algorithm for Binary Multiplication Example

Multiply 14 times -5 using 5-bit numbers (10-

bit result). 14 in binary: 01110

-14 in binary: 10010 (so we can add when we need to subtract

the multiplicand) -5 in binary: 11011

Expected result: -70 in binary: 11101 11010

| | | | **Multiplier** |
|---|---|---|---|
| **Step** | **Multiplicand** | **Action** | **upper 5-bits 0,** |
| | | | **lower 5-bits multiplier,** |
| | | | **1 "Booth bit" initially 0** |
| **0** | 01110 | Initialization | 00000 11011 0 |
| **1** | 01110 | 10: Subtract Multiplicand | 00000+10010=10010 |
| | | | 10010 11011 0 |
| | | Shift Right Arithmetic | 11001 01101 1 |
| **2** | 01110 | 11: No-op | 11001 01101 1 |
| | | | |
| | | Shift Right Arithmetic | 11100 10110 1 |
| | | | |
| **3** | 01110 | 01: Add Multiplicand | 11100+01110=01010<br>(Carry ignored because adding a positive and negative number cannot overflow.) |
| | | | 01010 10110 1 |
| | | Shift Right Arithmetic | |
| | | | |
| **4** | 01110 | 10: Subtract Multiplicand | 00101 01011 0 |
| | | | |
| | | | 00101+10010=10111 |
| | | Shift Right Arithmetic | 10111 01011 0 |
| | | | |
| **5** | 01110 | 11: No-op | 11011 10101 1 |
| | | | |
| | | Shift Right Arithmetic | 11011 10101 1 |
| | | | |
| | | | 11101 11010 1 |
| | | | |

## 4.5 Parity generator :

1. Parity is a very useful tool in information processing in digital computers to indicate any presence of error in bit information.

2. External noise and loss of signal strength causes loss of data bit information while transporting data from one device to other device, located inside the computer or externally.

3. To indicate any occurrence of error, an extra bit is included with the message according to the total number of 1s in a set of data, which is called parity.

4. If the extra bit is considered 0 if the total number of 1s is even and 1 for odd quantities of 1s in a set of data, then it is called even parity.

5. On the other hand, if the extra bit is 1 for even quantities of 1s and 0 for an odd number of 1s, then it is called odd parity

A parity generator is a combination logic system to generate the parity bit at the transmitting side.

| Four bit message<br>D3D2D1D0 | Even parity | Odd parity |
|---|---|---|
| 0000 | 0 | 1 |
| 0001 | 1 | 0 |
| 0010 | 1 | 0 |
| 0011 | 0 | 1 |
| 1000 | 1 | 0 |
| 0101 | 0 | 1 |
| 0110 | 0 | 1 |
| 0111 | 1 | 0 |
| 1000 | 1 | 0 |
| 1001 | 0 | 1 |
| 1010 | 0 | 1 |
| 1011 | 1 | 0 |
| 1100 | 0 | 1 |
| 1101 | 1 | 0 |
| 1110 | 1 | 0 |
| 1111 | 0 | 1 |

Table 1.1: Truth table for generating even and odd parity bit

If the message bit combination is designated as, D3D2D1D0 and Pe, Po are the even and odd parity respectively, then it is obvious from the table that the Boolean expressions of even parity and odd parity are

$Pe=D3\oplus D2\oplus D1$

$D0$ $Po=\overline{(D3\oplus D2\oplus D1}$

$D0)$

The above illustration is given for a message with four bits of information. However, the logic

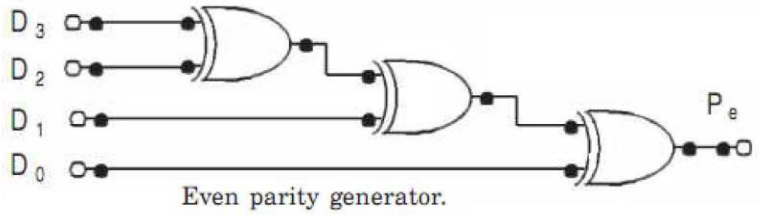


Figure 4.24: Even parity generator using logic gates

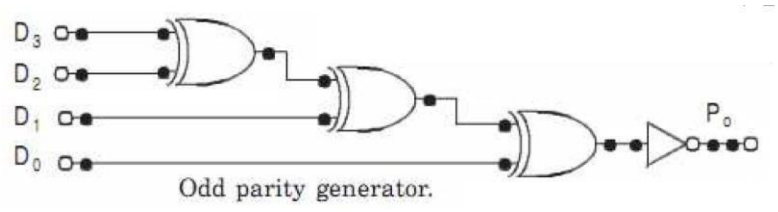


Figure 4.25: Odd parity generator logic gates

### 4.6 Zero/One detector :

Detecting all ones or zeros on wide N-bit words requires large fan-in AND or NOR gates. Recall that by DeMorgan's law, AND, OR, NAND, and NOR are funda-mentally the same operation except for possible inversions of the inputs and/or outputs. You can build a tree of AND gates, as shown in Figure 4.26(b). Here, alternate NAND and NOR gates have been used. The path has log N stages.

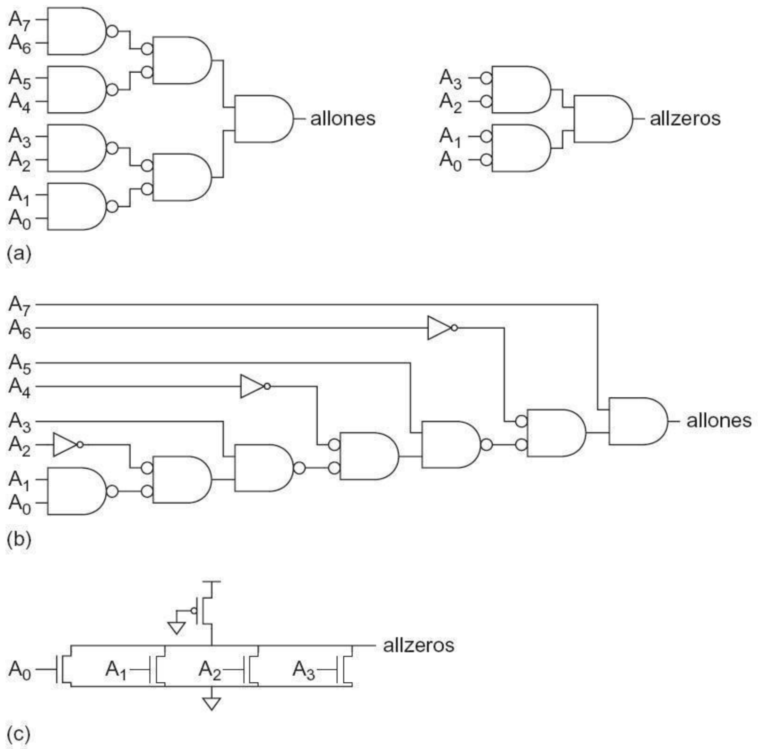


Figure 4.26: One/zero detectors (a) All one detector (b) All zero detector (c) All zero detector transistor level representation

### 4.7 Comparators :

Another common and very useful combinational logic circuit is that of the Digital Comparator circuit. Digital or Binary Comparators are made up from standard AND, NOR and NOT gates that compare the digital signals present at their input terminals and produce an output depending upon the condition of those inputs.

For example, along with being able to add and subtract binary numbers we need to be able to compare them and determine whether the value of input A is greater than, smaller than or equal to the value at input B etc. The digital comparator accomplishes this using several logic gates that operate on the principles of Boolean Algebra. There are two main types of Digital Comparator available and these are.

1. Identity Comparator an Identity Comparator is a digital comparator that has only one output terminal for when A = B either HIGH" A = B = 1or LOW" A = B = 0

2. Magnitude Comparator : a Magnitude Comparator is a type of digital com-parator that has three output terminals, one each for equality, A = B greater than,A > B and less than A < B

The purpose of a Digital Comparator is to compare a set of variables or unknown numbers, for example A (A1, A2, A3, . An, etc) against that of a constant or unknown value such as B (B1, B2, B3, . Bn, etc) and produce an output condition or ag depending upon the result of the comparison. For example, a magnitude comparator of two 1-bits, (A and B) inputs would produce the following three output conditions when compared to each other.

$$A > B; A + B; A < B$$

Which means: A is greater than B, A is equal to B, and A is less than B

This is useful if we want to compare two variables and want to produce an output when any of the above three conditions are achieved. For example, produce an output from a counter when a certain count number is reached. Consider the simple 1-bit comparator below.

Then the operation of a 1-bit digital comparator is given in the following Truth Table.

| Inputs | | Outputs | | |
|---|---|---|---|---|
| B | A | A > B | A=B | A < B |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |

From the above table the obtained expressions for magnitude comparator using K-map are as follows

For $A < B : C = \overline{A}\,B$

For $A = B : D = \overline{\overline{A}\,B + A\,\overline{B}}$

For $A > B : E = \overline{A}B$ The logic diagram of 1-bit comparator using basic gates is shown below in Figure 4.24.

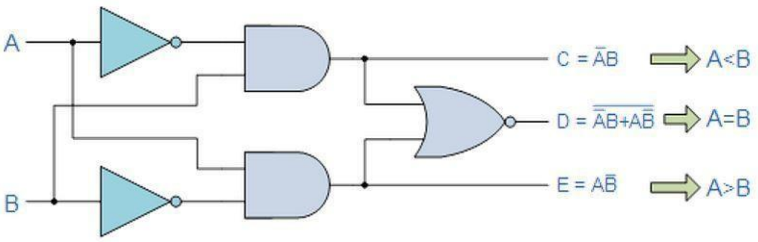
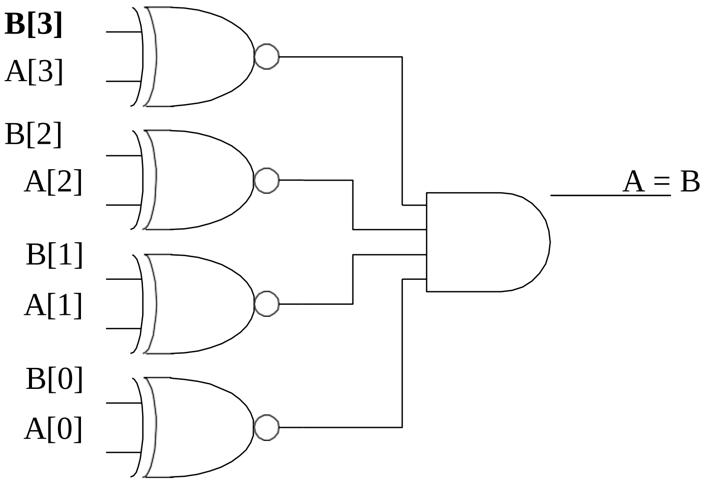


Figure 4.27: 1-bit Digital Comparator

*** Draw separate diagrams for grater, equality and less than expressions.

**Equality Comparator:**

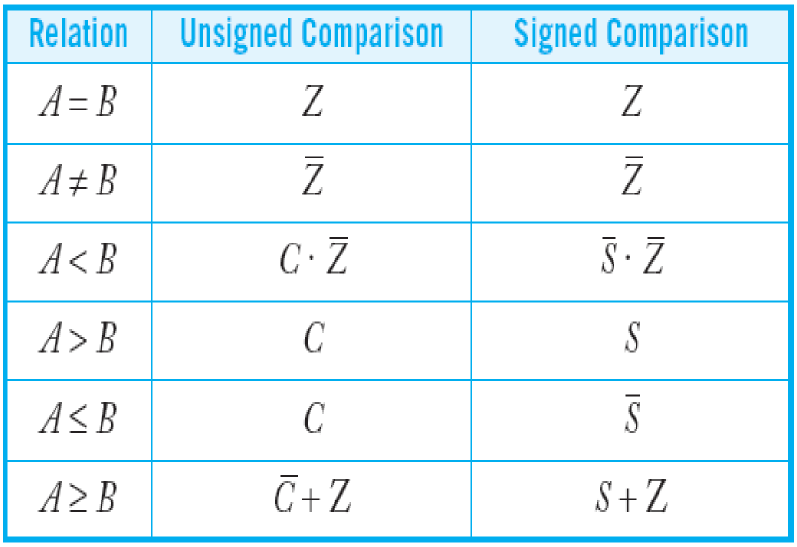
- Check if each bit is equal (XNOR, aka equality gate)
- 1's detect on bitwise equality



**Signed comparison:**

For signed numbers, comparison is harder

- C: carry out
- Z: zero (all bits of A-B are 0)
- N: negative (MSB of result)
- V: overflow (inputs had different signs, output sign □ B)



| Relation | Unsigned Comparison | Signed Comparison |
|---|---|---|
| $A = B$ | $Z$ | $Z$ |
| $A \neq B$ | $\bar{Z}$ | $\bar{Z}$ |
| $A < B$ | $C \cdot \bar{Z}$ | $\bar{S} \cdot \bar{Z}$ |
| $A > B$ | $C$ | $S$ |
| $A \leq B$ | $C$ | $\bar{S}$ |
| $A \geq B$ | $\bar{C} + Z$ | $S + Z$ |

## Magnitude Comparator:-

* Magnitude comparator, compares two numbers and determines whether a number is greater, smaller or equal to the other given number.

* To compare two numbers A and B, compute $B - A = B + \bar{A} + 1$.

Example:

Case-1  A > B

$A = 1110$  $\bar{A} = 0001$

$B = 1101$  $B = 1101$

$+ \quad 1$

Cout → 0 ← 1111

$Z = 0$

Case-2  A < B

$A = 1100$  $\bar{A} = 0011$

$B = 1111$  $B = 1111$

$+ \quad 1$

Cout= ⇐ 1 ← 0011

$Z = 0$

Case-3  A = B

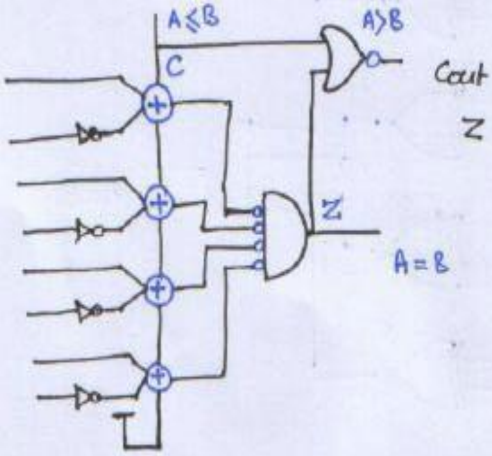
$A = 1101$  $\bar{A} = 0010$

$B = 1101$  $B = 1101$

$+ \quad 1$

Cout ← 1 ← 0000

$Z = 1$



| Relation | Comparison |
|---|---|
| A=B | $Z$ |
| A≠B | $\bar{Z}$ |
| A>B | $\bar{C}$ |
| A≤B | $C$ |

## 4.8 Counters :

Counters can be implemented using the adder/subtractor circuits and registers (or equivalently, D flip-flops)

The simplest counter circuits can be built using T flip-flops because the tog-gle feature is naturally suited for the implementation of the counting operation. Counters are available in two categories

1. Asynchronous(Ripple counters) Asynchronous counters, also known as ripple counters, are not clocked by a common pulse and hence every flip-flop in the counter changes at different times. The flip-flops in an asynchronous counter is usually clocked by the output pulse of the preceding flip-flop. The first flip-flop is clocked by an external event.

   The flip-flop output transition serves as a source for triggering other flip-flops i.e the C input (clock input) of some or all flip-flops are triggered NOT by the common clock pulses
   Eg:- Binary ripple counters, BCD ripple counters
2. Synchronous counters A synchronous counter however, has an internal clock, and the external event is used to produce a pulse which is synchronized with this internal clock.

   C input (clock input) of all flip-flops receive the common clock pulses

   E.g.:- Binary counter, Up-down Binary counter, BCD Binary counter, Ring counter, Johnson counter,

### 4.8.1 Asynchronous Up-Counter with T Flip-Flops

Figure 4.28 shows a 3-bit counter capable of counting from 0 to 7. The clock inputs of the three flip-flops are connected in cascade. The T input of each flip-flop is connected to a constant 1, which means that the state of the flip-flop will be toggled at each active edge (here, it is positive edge) of its clock. We assume that the purpose of this circuit is to count the number of pulses that occur on the primary input called Clock. Thus the clock input of the first flip-flop is connected to the Clock line. The other two flip-flops have their clock inputs driven by the Q output of the preceding flip-flop. Therefore, they toggle their states whenever the preceding flip-flop changes its state from Q = 1 to Q = 0, which results in a positive edge of the Q signal.

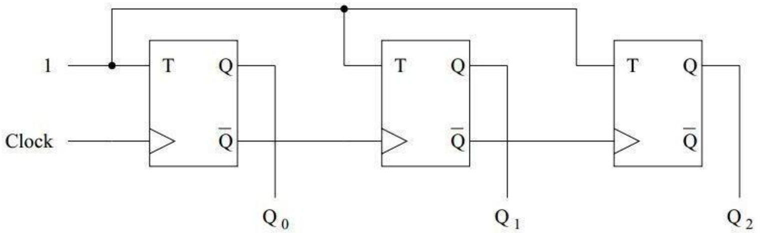


Figure 4.28: A 3-bit up-counter.

Note here the value of the count is the indicated by the 3-bit binary number Q2Q1Q0. Since the second flip-flop is clocked by Q0 , the value of Q1 changes shortly after the change of the Q0 signal. Similarly, the value of Q2 changes shortly after the change of the $\overline{Q1}$ signal. This circuit is a modulo-8 counter. Because it counts in the upward direction, we call it an up-counter. This behavior is similar to the rippling of carries in a ripple-carry adder. The circuit is therefore called an asynchronous counter, or a ripple counter.

### 4.8.2 Asynchronous Down-Counter with T Flip-Flops

Some modifications of the circuit in Figure 4.29 lead to a down-counter which counts in the sequence 0, 7, 6, 5, 4, 3, 2, 1, 0, 7, and so on. The modified circuit is shown in Figure 3. Here the clock inputs of the second and third flip-flops are driven by the Q outputs of the preceding stages, rather than by the Q outputs.

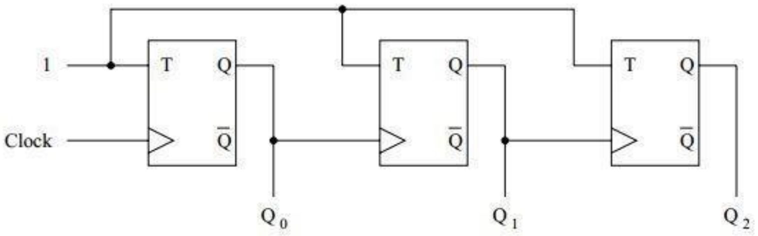


Figure 4.29: A 3-bit down-counter.

Although the asynchronous counter is easier to construct, it has some major disadvantages over the synchronous counter.

First of all, the asynchronous counter is slow. In a synchronous counter, all the flip-flops will change states simultaneously while for an asynchronous counter, the propagation delays of the flip-flops add together to produce the overall delay. Hence, the more bits or number of flip-flops in an asynchronous counter, the slower it will be.

### 4.8.3 Synchronous Counters

A synchronous counter usually consists of two parts: the memory element and the combinational element. The memory element is implemented using flip-flops while the combinational element can be implemented in a number of ways. Using logic gates is the traditional method of implementing combinational logic and has been applied for decades.

### 4.8.4 Synchronous Up-Counter with T Flip-Flops

An example of a 4-bit synchronous up-counter is shown in Figure 5. Observing the

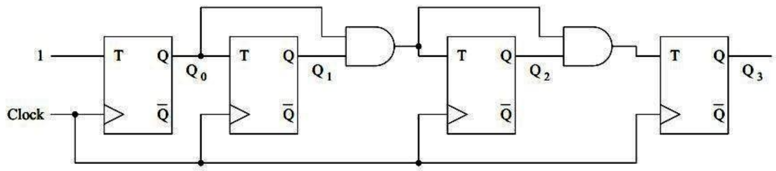
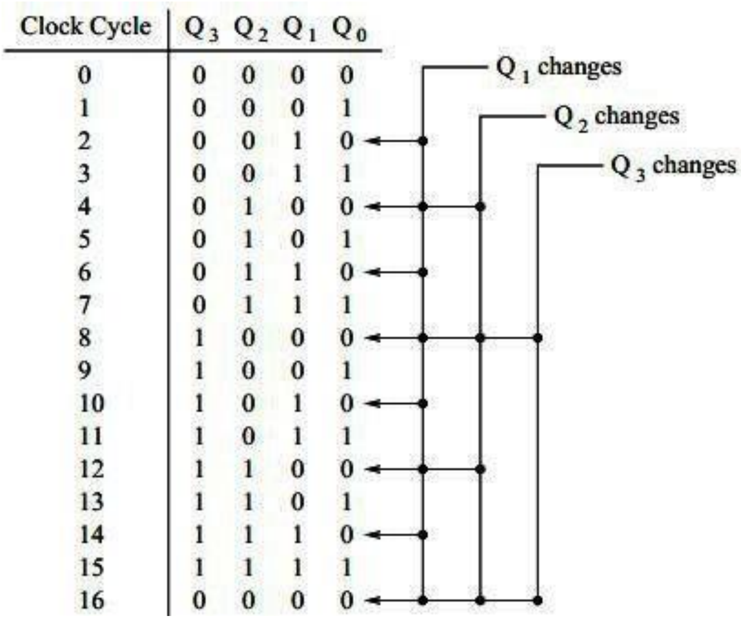


Figure 4.30: A 4bit synchronous upcounter



| Clock Cycle | $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |
| 10 | 1 | 0 | 1 | 0 |
| 11 | 1 | 0 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 |
| 14 | 1 | 1 | 1 | 0 |
| 15 | 1 | 1 | 1 | 1 |
| 16 | 0 | 0 | 0 | 0 |

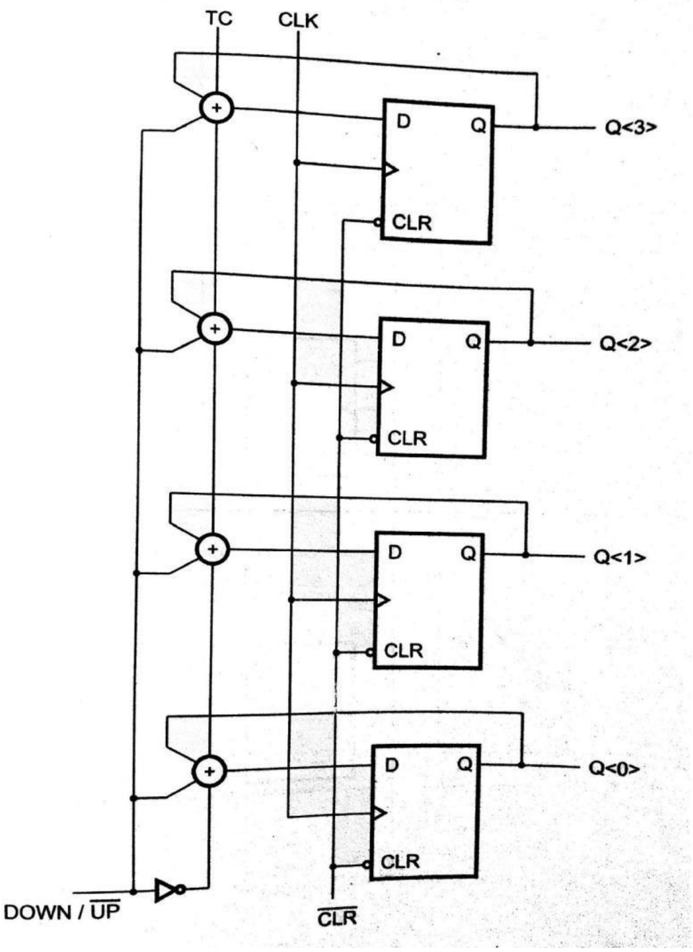
$Q_1$ changes
$Q_2$ changes
$Q_3$ changes

Figure 4.31: Contents of a 4bit upcounter for 16 consecutive clock cycles

pattern of bits in each row of the table, it is apparent that bit Q0 changes on each clock cycle. Bit QQ1 changes only when Q0 = 1. Bit Q2 changes only when both Q1 and Q0 are equal to 1. Bit

Q3 changes only when Q2 = Q1 = Q0 = 1. In general, for an n-bit up-counter, a give ip- op changes its state only when all the preceding ip- ops are in the state Q = 1. Therefore, if we use T ip- ops to realize the 4-bit counter, then the T inputs should be de ned as

In Figure 5, instead of using AND gates of increased size for each stage, we use a factored arrangement. This arrangement does not slow down the response of the counter, because all ip- ops change their states after a propagation delay from the positive edge of the clock. Note that a change in the value of Q0 may have to propagate through several AND gates to reach the ip- ops in the higher stages of the counter, which requires a certain amount of time. This time must not exceed the clock period. Actually, it must be 3less than the clock period minus the setup time of the ip- ops. It shows that the circuit behaves as a modulo-16 up- counter. Because all



changes take place with the same delay after the active edge of the Clock signal, the circuit is called a synchronous counter.

Figure 4.32: Design of synchronous counter using adders and registers

## 4.9 Shifters

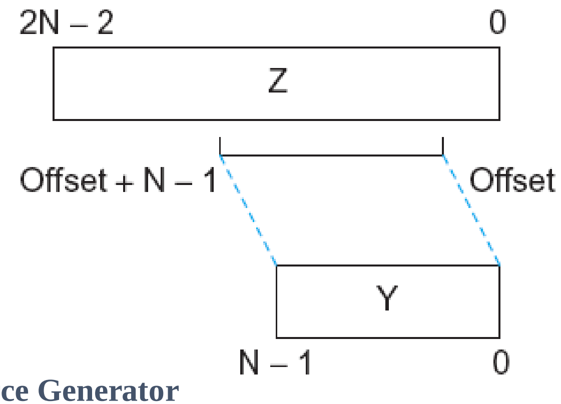
### 4.9.1 Shifters :

- Logical Shift:
  - Shifts number left or right and fills with 0’s
    - o 1011 LSR 1 = 0101 1011 LSL1 = 0110
- Arithmetic Shift:
  - Shifts number left or right. Rt shift sign extends
    - o 1011 ASR1 = 1101 1011 ASL1 = 0110
- Rotate:
  - Shifts number left or right and fills with lost bits
    - o 1011 ROR1 = 1101 1011 ROL1 = 0111

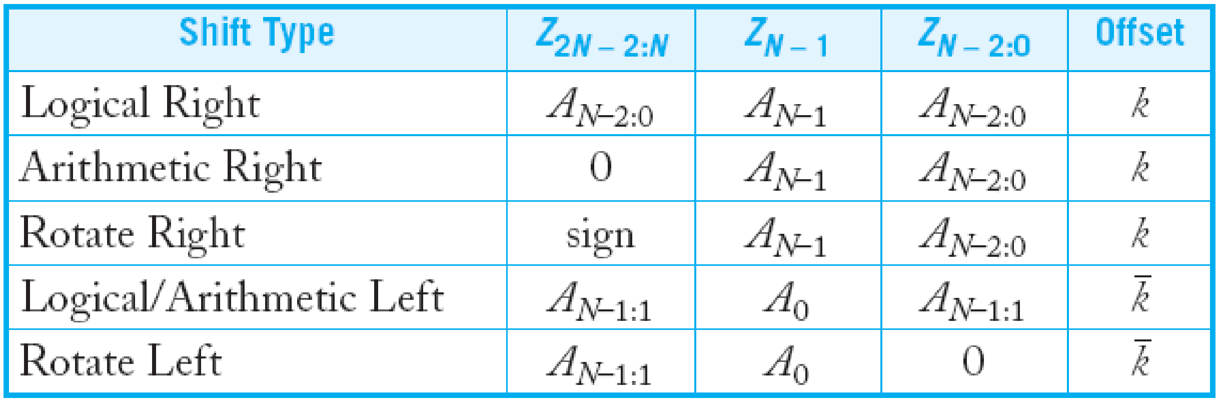
### 4.9.2 Funnel Shifter

- A funnel shifter can do all six types of shifts
- Selects N-bit field Y from 2N–1-bit input

- Shift by k bits (0 □ k < N)
- Logically involves N N:1 multiplexers



## Funnel Source Generator

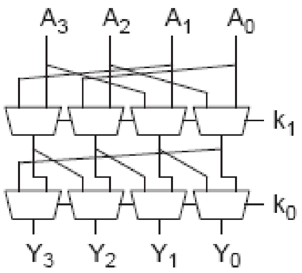


| Shift Type | $Z_{2N-2:N}$ | $Z_{N-1}$ | $Z_{N-2:0}$ | Offset |
|---|---|---|---|---|
| Logical Right | $A_{N-2:0}$ | $A_{N-1}$ | $A_{N-2:0}$ | $k$ |
| Arithmetic Right | 0 | $A_{N-1}$ | $A_{N-2:0}$ | $k$ |
| Rotate Right | sign | $A_{N-1}$ | $A_{N-2:0}$ | $k$ |
| Logical/Arithmetic Left | $A_{N-1:1}$ | $A_0$ | $A_{N-1:1}$ | $\overline{k}$ |
| Rotate Left | $A_{N-1:1}$ | $A_0$ | 0 | $\overline{k}$ |

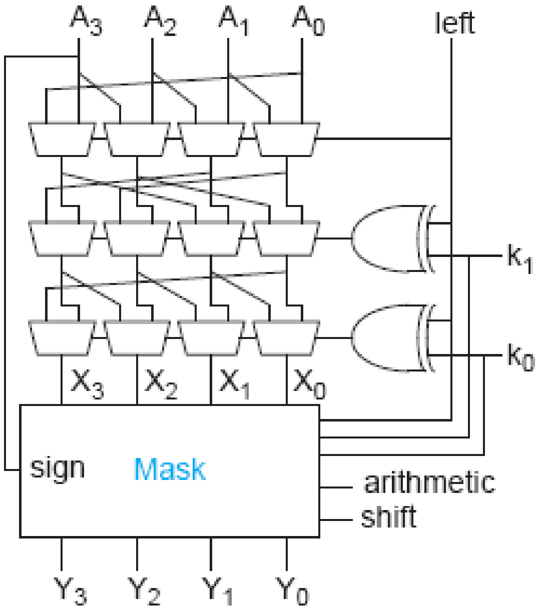
### 4.9.3 Barrel Shifter

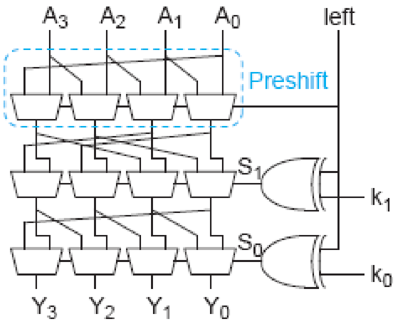
- ➢ Barrel shifters perform right rotations using wrap-around wires.
- ➢ Left rotations are right rotations by N – k = k + 1 bits.
- ➢ Shifts are rotations with the end bits masked off.

## Logarithmic Barrel Shifter



Right shift only

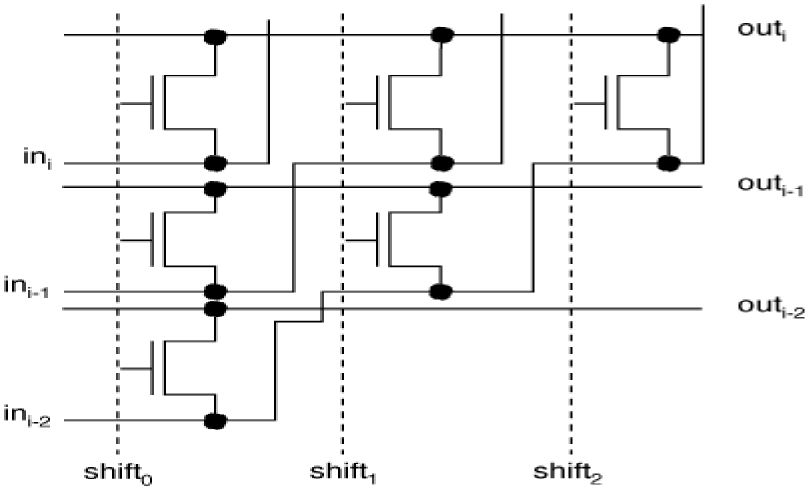




Right/Left shift

Right/Left Shift & Rotate

Barrel shifter cell



## LFSR (Linear Feedback Shift Register):-

* LFSR is an example of PRSGs (Pseudo Random Sequence Genelators) that are used to genelate random number sequences and their special applications are stimuli genelation in BIST (Built-In-Self-Test) and other testability techniques.

* LFSR consists of N registers connected together as a shift register.

* The input to shift register comes from the XOR of particular bits of the register. The bits that are fed to XOR are called tap sequence and are often specified with characteristic polynomial. Some characteristic polynomial ale shown on next page.

* For LFSR to work as PRSG, it must be initialized to a non-zelo seed value.

* An LFSR is considered maximal LFSR if it genelates $2^n-1$ distinct sequences.

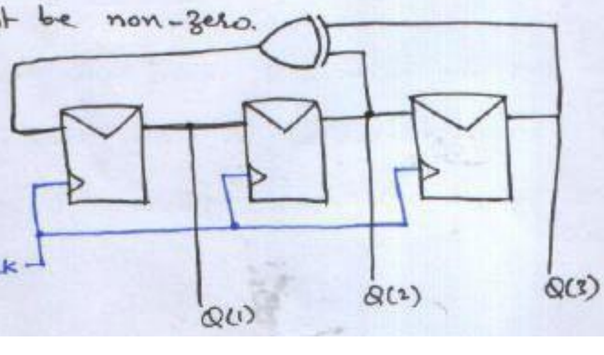
* For maximal LFSR, the following conditions must be met:

  1 Bits in the tap sequence must be even.

  2 n-th bit of register must be part of XOR operation i-e tap sequence.

  3 Feedback must be given to the first bit

  4 Seed value must be non-zelo.



3-bit figure (LFSR)

* Truth table for 3-bit LFSR is given below:

| cycle | Q[1] | Q[2] | Q[3] |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 1 |
| 3 | 0 | 0 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 |
| 6 | 1 | 0 | 1 |
| 7 | 1 | 1 | 0 |

→ 110 is seed value

* Observe: after 0th cycle i-e seed value,

$Q[1] = Q[2] \oplus Q[3]$

$Q[2] = Q[1]$ } Shift

$Q[3] = Q[2]$

FEEDBACK

→ sequence 110 repeated

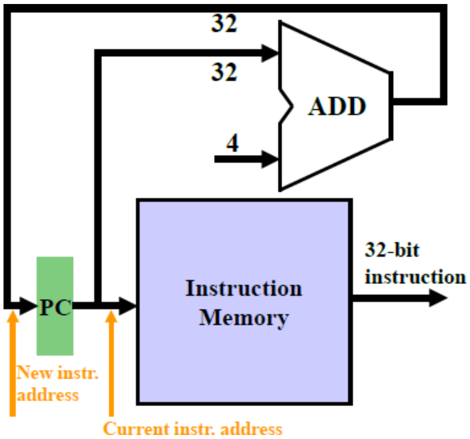
* Example characteristic polynomials for Maximal LFSRs of various sizes are given below:

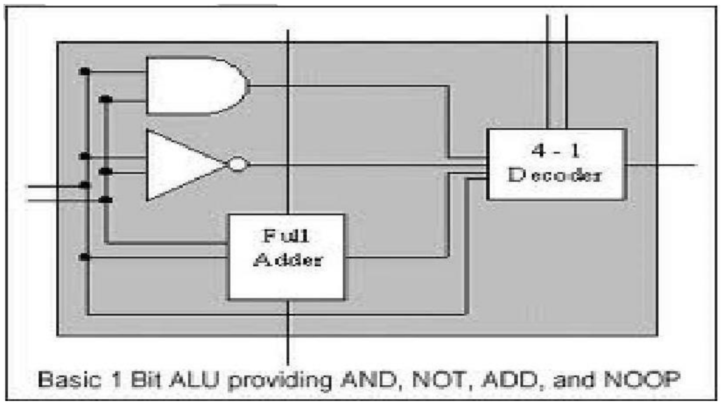
| N | Polynomial |
|---|---|
| 3 | $1+x^2+x^3$ |
| 4 | $1+x^3+x^4$ |
| 5 | $1+x^3+x^5$ |
| 6 | $1+x^5+x^6$ |
| 7 | $1+x^6+x^7$ |
| 8 | $1+x^1+x^6+x^7+x^8$ |
| 9 | $1+x^5+x^9$ |
| 15 | $1+x^{14}+x^{15}$ |
| 16 | $1+x^4+x^{13}+x^{15}+x^{16}$ |

* For certain lengths, N, more than two taps may be required.
* For many values of N, there are multiple polynomials resulting in different maximal lengths LFSRs.

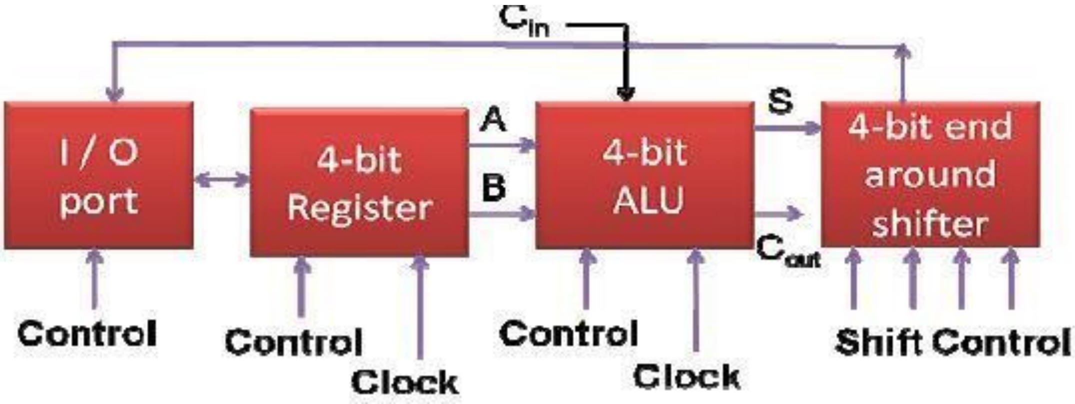
### *4.10 ALU:*

An ALU is a Arithmetic Logic Unit that requires Arithmetic operations and Boolean operations. Basically arithmetic operations are addition and subtraction. one may either multiplex between an adder and a Boolean unit or merge the Boolean unit into the adder as in tha classic transistor-transistor logic.





Basic 1 Bit ALU providing AND, NOT, ADD, and NOOP

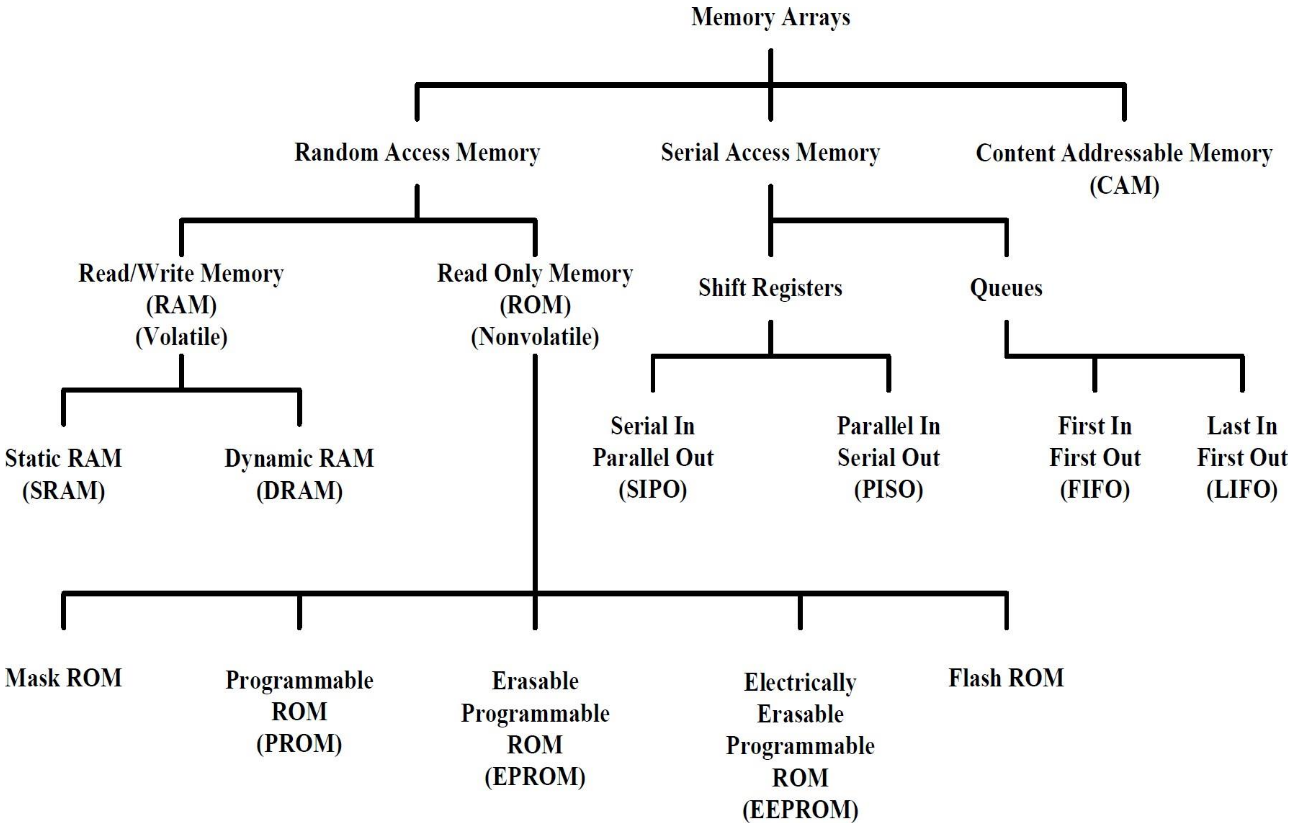


**4-bit data path for processor**

The heart of the ALU is a 4-bit adder circuit. A 4-bit adder must take sum of two 4-bit numbers, and there is an assumption that all 4-bit quantities are presented in parallel form and that the shifter circuit is designed to accept and shift a 4-bit parallel sum from the ALU. The sum is to be stored in parallel at the output of the adder from where it is fed through the shifter and back to the register array. Therefore, a single 4-bit data bus is needed from the adder to the shifter and another 4-bit bus is required from the shifted output back to the register

## Memory Array

The memory array is classified into 3 types - Random Access memory (RAM), Serial access memory and content addressable memory (CAM). We will discuss each type in detail.


Memory Arrays
Random Access Memory
Serial Access Memory
Content Addressable Memory (CAM)
Read/Write Memory (RAM) (Volatile)
Read Only Memory (ROM) (Nonvolatile)
Shift Registers
Queues
Static RAM (SRAM)
Dynamic RAM (DRAM)
Serial In Parallel Out (SIPO)
Parallel In Serial Out (PISO)
First In First Out (FIFO)
Last In First Out (LIFO)
Mask ROM
Programmable ROM (PROM)
Erasable Programmable ROM (EPROM)
Electrically Erasable Programmable ROM (EEPROM)
Flash ROM

### 4.11 Read only memory (ROM)

The basic idea of the memory that can only be read and never altered is called Read only memories. There are vast and variety of potential applications for these kind of memories. Programs for processors with fixed applications such as washing machines, calculators and game machines, once developed and debugged, need only reading. Fixing the contents at manufacturing time leads to small and fast implementation.

There are different ways to implement the logic of ROM cells, the fact that the contents of a ROM cell are permanently fixed considerably simplifies its design. The cell should be designed so that a „0‟ or „1‟ is presented to the bitline upon activation of its”wordline. The different approaches for implementing the ROM cells are Diode ROM, MOS ROM 1 and MOS ROM 2. These are the main approaches for designing a larger density ROMs.

### 4.12.1 Mask ROM :

The ROM memories which we have seen earlier are application specific ROMs where the memory module is part of a larger custom design and programmed for that particular application only. The ROMs which we are going to discuss in this section are commodity ROMs, where a vendor mass-produces memory modules that are later customized according to customer specifications. Under these circumstances, it is essential that the number of process steps involved in programming be minimal and that they can be performed as a last phase of the manufacturing process. In this way large amounts of programmed dies can be preprocessed.

This mask-programmable approach preferably uses the contact mask to personalize or program the memory. The programming of a ROM module involves the manufacturer, which introduces an unwelcome delay in product development. The major usage of this ROM was in system-on-a-chip where the majority of the chip is preprocessed, only the minor part of the die is mask programmed. The other usages of this ROM are to program the microcontroller, embedded on the chip, for a variety of applications.

NOR-based ROM

The building block of this ROM is a pseudo-nMOS NOR gate as in Figure 4.33

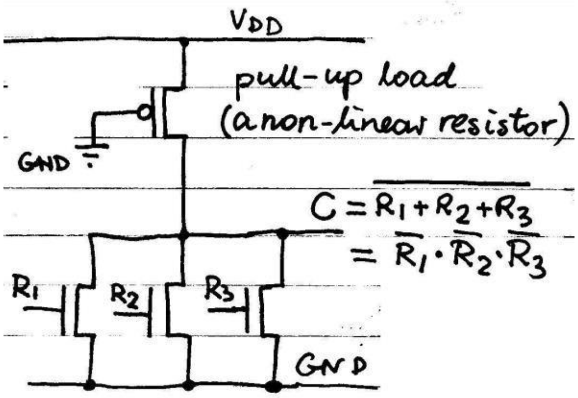

VDD
pull-up load
(a non-linear resistor)
GND
$C = \overline{R_1 + R_2 + R_3}$
$= \overline{R_1} \cdot \overline{R_2} \cdot \overline{R_3}$
R1
R2
R3
GND

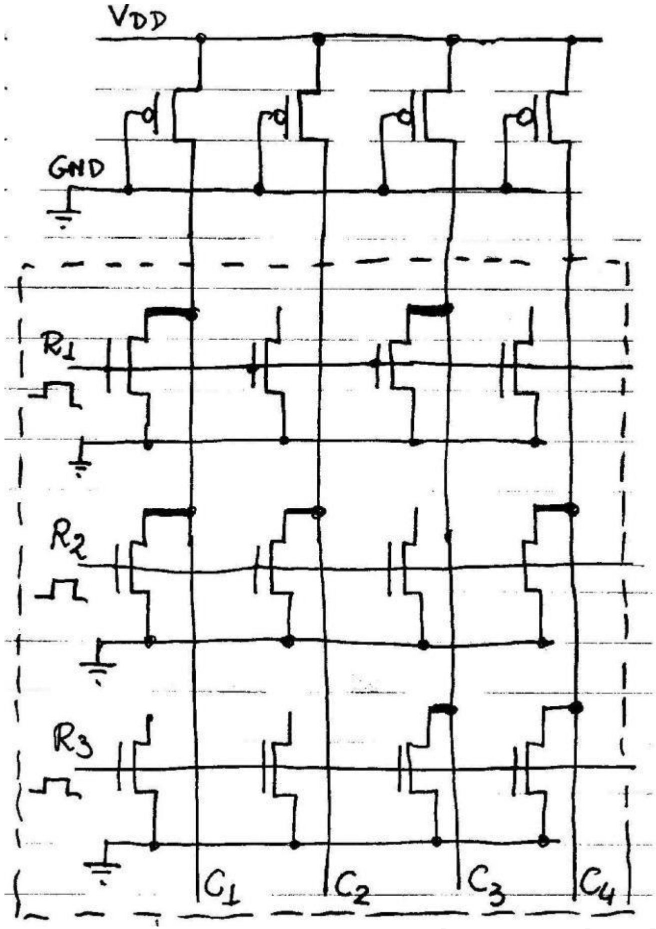
Figure 4.33: A 3-input pseudo-nMOS NOR gate.

Unlike in a standard CMOS gate, the pMOS pull-up circuitry is replaced by a single pMOSwith its gate tied up to GND, hence being permanently on acting as a load resistor.

If none of the nMOS transistors is activated (all Ri being low) then the output signal C is high. If any of the nMOS transistors is activated (Ri being high) then the output signal C is low.

To reduce the power consumption the gate of the pMOS pull-up transistor is connected to a clock signal. The power is consumed only during low period of the clock.

**NOR-based ROM** consists of m n-input pseudo-nMOS NOR gates, one n-input NOR per column as shown in Figure 4.34.



|  | C1 | C2 | C3 | C4 |
|---|---|---|---|---|
| R1 | 0 | 1 | 0 | 1 |
| R2 | 0 | 0 | 1 | 0 |
| R3 | 1 | 1 | 0 | 0 |

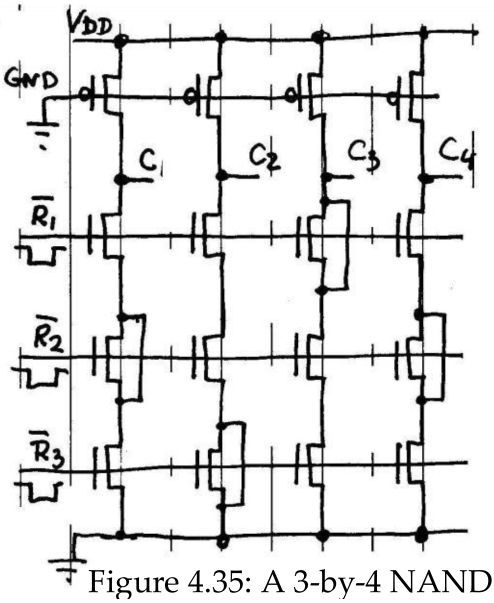
Figure 4.34: A 3-by-4 NOR-based ROM array

Each memory cell is represented by one nMOS transistor and a binary information is stored by connecting or not the drain terminal of such a transistor to the bit line.

For every row address only one word line is activated by applying a high signal to the gates of nMOS transistors in a row.

If a selected transistor in the i-th column is connected to a bit line then the logic ‘0’ is stored in this memory cell. if the transistor is not connected, then the logic ‘1’ is stored. NAND-based ROM

A NAND-based ROM consists of m n-input pseudo-nMOS NAND gates, one n-input NAND per column as shown in Figure 4.35. In this case, we have up to n serially connected nMOS transistors in each column.



| | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
|---|---|---|---|---|
| $\overline{R_1}$ | 1 | 1 | 0 | 1 |
| $\overline{R_2}$ | 0 | 1 | 1 | 0 |
| $\overline{R_3}$ | 1 | 0 | 1 | 1 |

Figure 4.35: A 3-by-4 NAND-based ROM array

For every row address only one word line is activated by applying a low signal to the gates of nMOS transistors in a row. When no word line is activated, all nMOS transistors are on and the line signals, Ci are all low.

When a word line is activated all transistors in the row are switched off and the respective Ci signals are high. If a transistor in the selected row is short-circuited, then the respective Ci signal is low.

In other words, the logic '0' is stored when a transistor is replaced with a wire, whereas the logic '1' is stored by an nMOS transistor being present.

### 4.12.2 Programmable ROM (PROM) :

The technology that offers its users to program the memory one time is called Programmable ROM. It is also called as WRITE ONCE device. This is most often accomplished by introducing fuses (implemented in nichrome, polysilicon, or other conductors) in the memory cell. During the programming phase, some of these fuses are blown by applying a high current, which disables the connected transistor.

While PROMs have the advantage of being "customer programmable," the single write phase makes them unattractive. For instance, a single error in the programming process or application makes the device unstable. This explains the current preference for devices that can be programmed several times.

The Floating-Gate transistor is the device at the heart of the majority of reprogrammable memories. Various attempts have made to create a device with electrically alterable characteristics and enough reliability to support a multitude of write cycles. The floating gate structure is similar to a traditional MOS device, except that an extra polysilicon strip is inserted between the gate and channel.

This strip is not connected to anything and is called a floating gate. The most obvious impact of inserting this extra gate is to double the gate oxide thickness *tox*, which results in a reduced device transconductance as well as an increased threshold voltage. Though
these properties are not desirable but from other point of view this device acts as a normal transistor.

The most important property of this device is that the threshold voltage of this device is programmable. By applying a high voltage (above 10V) between the source and the gate-drain terminals creates a high electric field and causes avalanche injection to occur. Electrons acquire sufficient energy to become "hot" and traverse through the first oxide insulator, so that they get trapped on the floating gate. In reference to the programming mechanism, the floating-gate transistor is often called a floating-gate avalanche- injection MOS.

The trapping of electrons on the floating gate effectively drops the voltage on the gate. This process is self-limiting – the negative charge accumulated on the floating gate reduces the electrical field over the oxide so that ultimately it becomes incapable of accelerating any more hot electrons. Virtually all nonvolatile memories are currently based on the floating-gate mechanism. Different classes can be identified, based on the erasure mechanism.

### 4.12.3 Erasable-programmable Read-Only Memory (EPROM) :

The erasure mechanism in EPROM is based on the shining ultraviolet light on the cells through a transparent window in the package. The UV radiation renders the oxide to conduct by the direct generation of electron-hole pairs in the material. The erasure process is slow depending on the UV source, it can take from seconds to several minutes. The programming takes several µs/word. Alternatively there is another problem which exists is the limited endurance - the number of erase/program cycles is limited to a maximum of one thousand mainly as a result of UV erasing procedure. The device thresholds might vary with repeated programming cycles. The on-chip circuitry is designed in such a way that it also controls the value of the thresholds to within a specified range during programming. The injection of large channel current of 0.5 mA at a control gate voltage of 12.5V causes high power dissipation during programming.

On the other hand, EPROM is extremely simple and dense, making it possible to fabricate large memories at a low cost. Therefore EPROMs were attractive in applications that do not require reprogramming. The major disadvantage of the EPROM is that the erasure procedure has to occur "off system". This means the memory must be removed from the board and placed in an EPROM programmer for programming.

**4.12.4 Electrically Erasable Programmable Read-Only Memory EEPROM)** The disadvantage of the EPROM [16] is solved by using a method to inject or remove charges from a floating-gate namely – tunneling. A modified floating-gate device called FLOTOX (floating-gate tunneling oxide) transistor is used as programmable device that supports an electrical-erasure procedure. It resembles FAMOS (floating-gate avalanche MOS) device, except that a portion of the dielectric separating the floating gate from the channel and drain is reduced in thickness to about 10 nm or less.

The main advantage of this programming approach is that it is reversible; that is, erasing is simply achieved by reversing the voltage applied during the writing process. The electrons injection on floating-gate raises the threshold, while the reverse operation lowers the VT. When a voltage of approximately 10V (equivalent to $10^9$ V/m) is applied over the thin insulator, electrons travel to and from the floating gate through a mechanism called Fowler – Nordheim tunneling.

### 4.12.5 Flash Electrically Erasable Programmable ROM (Flash) :

The concept of Flash EEPROMs is a combination of density of EPROM with versatility of EEPROM structures, with cost and functionality ranging from somewhere between two. Most Flash EEPROM devices use the avalanche hot-electron-injection approach to program the device. Erasure is performed using Fowler – Nordheim tunneling, as from EEPROM cells. The main difference is that erasure procedure is performed in bulk for a complete chip or for the subsection of the memory. Erasing complete memory core at once makes it possible to carefully monitor of the device characteristics during erasure.

The monitoring control hardware on the memory chip regularly checks the value of the threshold during erasure, and adjusts the erasure time dynamically. This approach is only practical when erasing large chunks of memory at a time; hence the flash concept. One of the many existing alternatives for Flash EEPROMs memories are ETOX devices. It resembles a FAMOS gate except that a very thin tunneling gate oxide is utilized (10 nm). Different areas of the gate oxide are used for programming and erasure. Programming is performed by applying a high voltage (12V) on the gate and drain terminals for a grounded source, while erasure occurs with the gate rounded and the source at 12V.

The Programming cycle starts with an erase operation. In erase operation, A 0V gate voltage is applied and a 12V supply is given at source. Electrons, if any, are ejected to the source by tunneling. All cells are erased simultaneously. The variations caused in the threshold voltage at the end of erase operation are due to different initial values of cell threshold voltage and variations in oxide thickness. This can be solved in two methods:

1. The array cells are programmed before applying the erase pulse so that the entire threshold starts at approximately same time.
2. An erase pulse of controlled width is applied. Subsequently the whole array is read to ensure that all the cells are erased. If not another erase pulse is applied followed by the read cycle.

For write (programming) operation, a high voltage is applied to the gate of the selected device. If a „1" is applied to the drain at that time, hot electrons are generated and injected onto the floating gate, raising the threshold. Read operation corresponds as the wordline is raised to 5V; it causes a conditional discharge of bitline.

### 4.12 Random Access memory (RAM) :

Random access memory is a type of computer data storage. It is made of integrated circuits that allow the stored data to be accessed in any order i.e., at random and without the physical movement of storage medium or a physical reading head. RAM is a volatile memory as the information or the instructions stored in the memory will be lost if the power is switched off.

The word “random” refers to the fact that any piece of data can be returned at a constant time regardless of its physical location and whether or not it is related to the previous piece of data. This contrasts with the physical movement devices such as tapes, magnetic disks and optical disks, which rely on physical movement of the recording medium or reading head. In these devices, the retrieval time varies with the physical location and the movement time takes longer than the data transfer.

The main advantages of RAM over types of storage which require physical movement is that retrieval times are short and consistent.Short because no physical movement is necessary and consistent the time taken to retrieve the data does not depend on the current distance from a physical head. The access time for retrieving any piece of data in RAM chip is same. The disadvantages are its cost compared to the physical moving

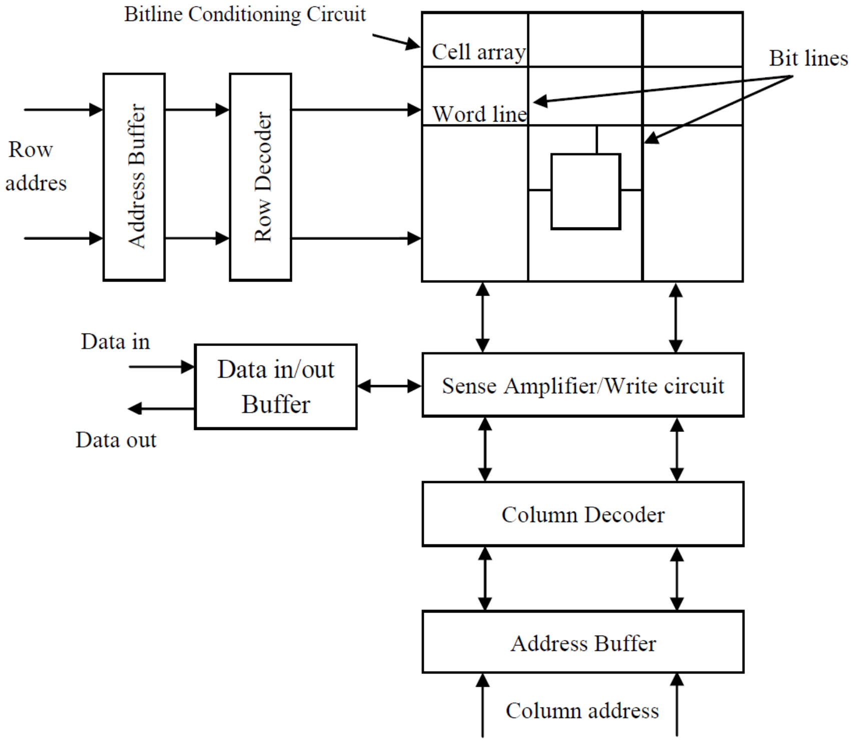
media and loss of data when power is turned off.

RAM is used as 'main memory' or primary storage because of its speed and consistency. The working area used for loading, displaying and manipulating applications and data. In most personal computers, the RAM is not an integral part of the motherboard or CPU. It comes in the easily upgraded form of modules called memory sticks. These can quickly be removed and replaced when they are damaged or when the system needs up gradation of memory depending on current purposes. A smaller amount of random- access memory is also integrated with the CPU, but this is usually referred to as "cache" memory, rather than RAM. Modern RAM generally stores a bit of data as either a charge in a capacitor, as in dynamic RAM, or the state of a flip-flop, as in static RAM.

## 4.13 Static Random Access Memory (SRAM)

### *4.14 SRAM Architecture:*

The typical SRAM design is shown in figure 1.8 the memory array contains the memory cells which are readable and writable. The Row decoder selects from 1 out of n = 2k rows, while the column decoder selects l = 2i out of m = 2j columns. The addresses are not multiplexed as it in the DRAM. Sense amplifier detects small voltage variations on the memory complimentary bitline which reduces the reading time. The conditioning circuit is used to pre-charge the bitlines.



**Typical SRAM Architecture**

In a read operation, the bitlines are precharged to some reference voltage usually close to the supply voltage. When word line turns high, the access transistor connected to the node storing „0‟ starts discharging the bitline while the complementary bitline remains in its precharged state, resulting in a differential voltage between the bitline pair.

Since the SRAM has an optimized area results in a small cell current and slow bitline discharge rate. In order to speed up the RAM access, sense amplifiers are used which amplify the small bitline signal and eventually drive it to the external world.

The word "static" means that the memory retains its contents as long as the power is turned on. Random access means that locations in the memory can be written to or read from in any order, regardless of the memory location that was last accessed. Each bit in an SRAM is stored on four transistors that form two cross-coupled inverters. This storage cell has two stable states which are used to denote „0‟ and „1‟. The access transistors are used to access the stored bits in the SRAM during read or write mode.

It thus typically takes six MOSFETs to store one memory bit. Access to the cell is enabled by the word line WL which controls the two access transistors N1 and N2 which, in turn, control whether the cell should be connected to the bitlines BL and /BL.

They are used to transfer data for both read and write operations. The bitlines are complementary as it improves the noise margin. Chapter 2 explains more about SRAMs and its Read/Write operations.

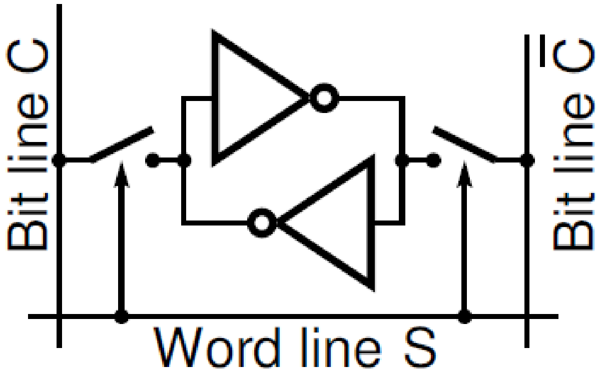


Figure 4.36: A logic diagram of a CMOS static memory cell

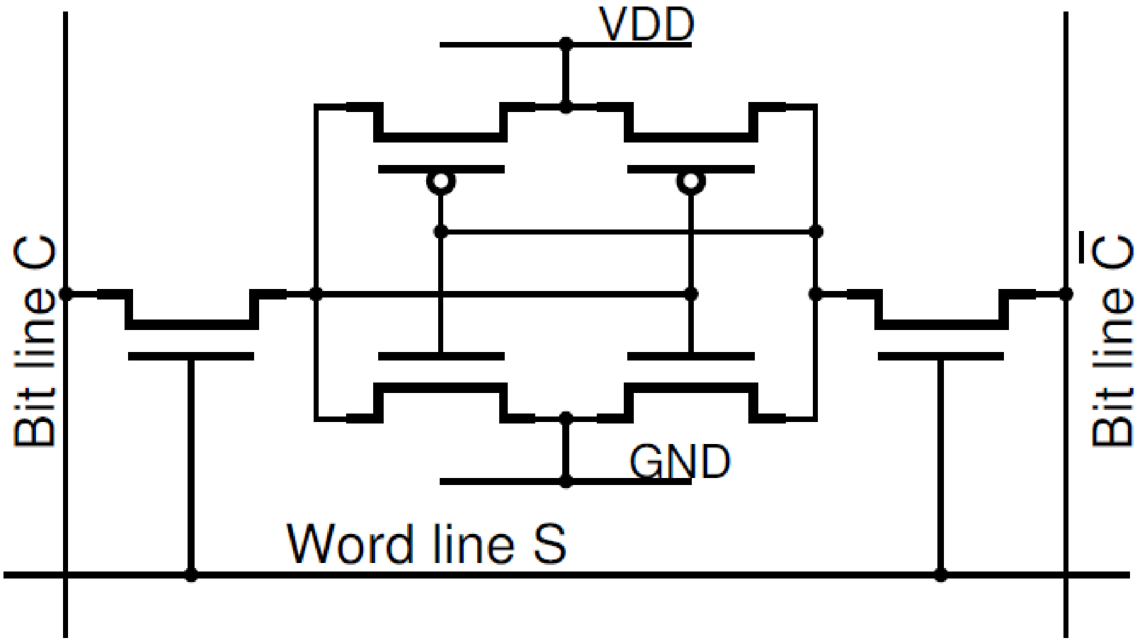


Figure 4.37: A schematic of a CMOS static memory cell

### 4.15.1 Principles of operations

In order to consider operation of the static read/write memory we have to take into account:

- ➢ Relatively large parasitic column capacitances, CC and Cc
- ➢ column pull-up pMOS transistors, as shown in Figure 4.38

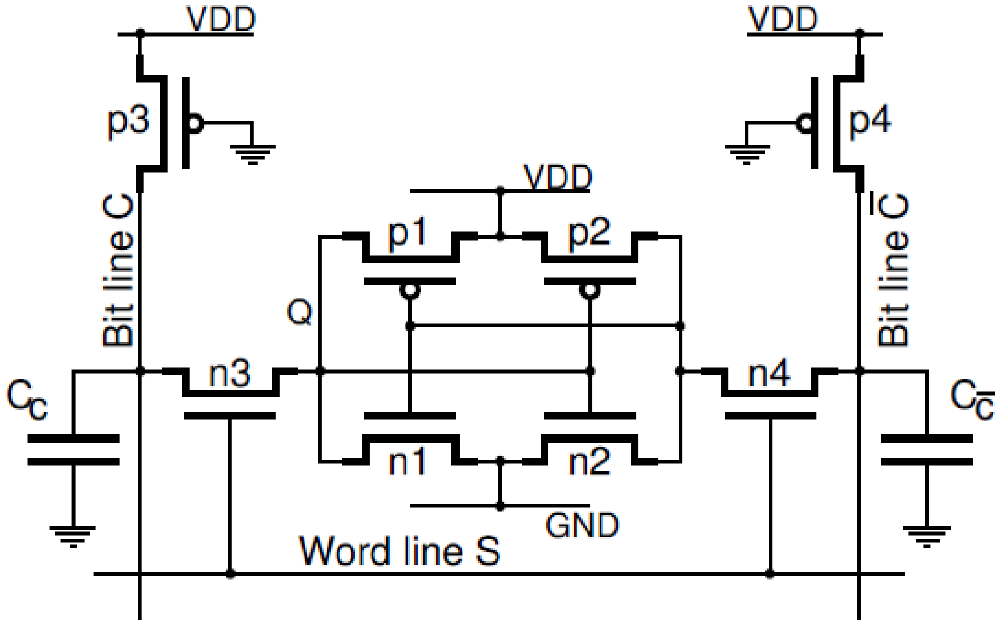


Figure 4.38: A CMOS static memory cell with column pull-up transistors and parasitic column capacitances

When none of the word lines is selected, that is, all S signals are ‘0’, the pass transistors n3, n4 are turned off and the data is retained in all memory cells. The column capacitances are charged by the drain currents of the pull-up pMOS transistors, p3, p4. The column voltages VC and V c¯ both reach the level just below VDD − VT p, say 3.5V for VDD = 5V and the threshold voltage VT p = 1V.

For the read or write operations we select the cell asserting the word line signal S=‘1’.For the write operation we apply a low voltage to one of the bit line, holding the other one high. To write ‘0’ in the cell, the column voltage VC is forced to low(C = 0). This low voltage acts through a related pass transistor (n3) on the gates of the corresponding inverter (n2, p2) so that its input goes high. This sets the signal at the other inverter Q = 0.
Similarly, to write ‘1’ in the cell, the opposite column voltage VC¯ is forced to low (C¯ = 0) which sets the signal Q = 1.During the read ‘1’ operation, when the stored bit is Q = 1, transistors n3, p1 and n4, n2 are turned on. This maintains the column voltage VC at its steady-state high level (say 3.5V) while the opposite column voltage VC¯ is being pulled down discharging the column capacitance CC¯ through transistors n4, n2 so that VC > VC¯. Similarly, during the read ‘0’ operation we have VC < VC¯. The difference between the column voltages is small, say 0.5V, and must be detected by the sense amplifiers from data-read circuitry.

### 4.15.2 SRAM Write Circuitry

The structure of the write circuitry associated with one column of the memory cells is shown in Figure 4.39.

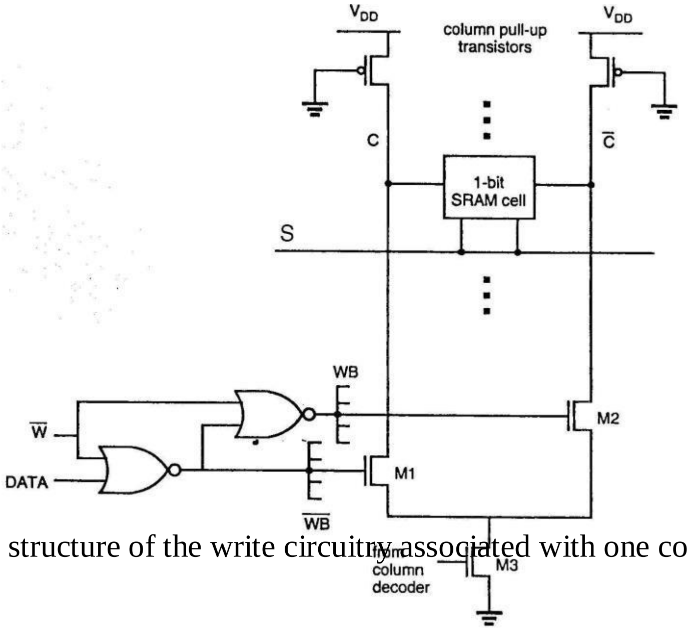


Figure 4.39: The structure of the write circuitry associated with one column of the memory cells.

The principle of the write operation is to assert voltage on one of the columns to a low level. This is achieved by connecting either C or $\overline{C}$ to the ground through the transistor M3 and either M1 or M2.

The transistor M3 is driven by the signal from the column decoder selecting the specified column. The transistor M1 is on only in the presence of the write enable signal.(~~W~~ = 0) when the data bit to be written is ‘0’. The transistor M2 is on only in the presence of the write signal $\overline{\ }$(W = 0) when the data bit to be written is ‘1’.

### 4.15.3 SRAM Read Circuitry

The structure of the read circuitry is shown in Figure 4.40.

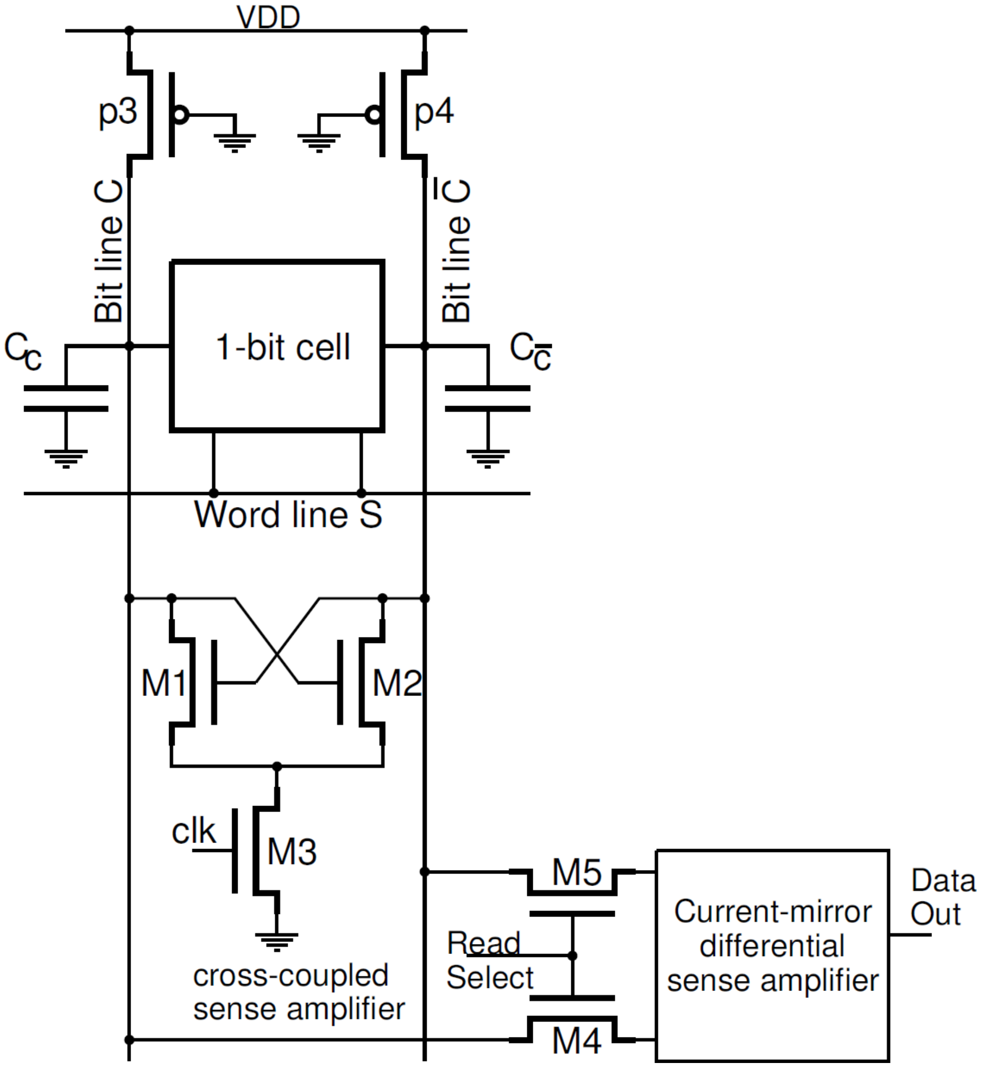


Figure 4.40: The structure of the write circuitry associated with one column of the memory cells. During the read operation the voltage level on one of the bit lines drops slightly after the pass transistors in the memory cell are turned on.

The read circuitry must properly sense this small voltage difference and form a proper output bit:

‘0? If VC <
VC¯ ‘1? If VC
> VC¯

The read circuitry consists of two level sense amplifiers:

- One simple cross-coupled sense amplifier per column of memory cells,
- One current-mirror differential sense amplifier per the memory chip.

The cross-coupled sense amplifier works as a latch. Assume that the voltage on the bit line C start to drop slightly when the memory access pass transistors are activated by the word line signal S, and that the clk signal is high so that the transistor M3 is turned on. Now, higher voltage on the gate of M1 transistor than on the gate of M2 starts the latching operation which pulls the VC voltage further down switching the transistor M2 off. As a result the parasitic capacitance, CC is discharged through M1 and M3. In this way a small difference between column voltages is amplified.

The amplified (discriminated) column voltages are passed through transistors M4 and M5 to the main sense amplifier.

The schematic of a typical differential current-mirror sense amplifier is shown in Figure 4.41.

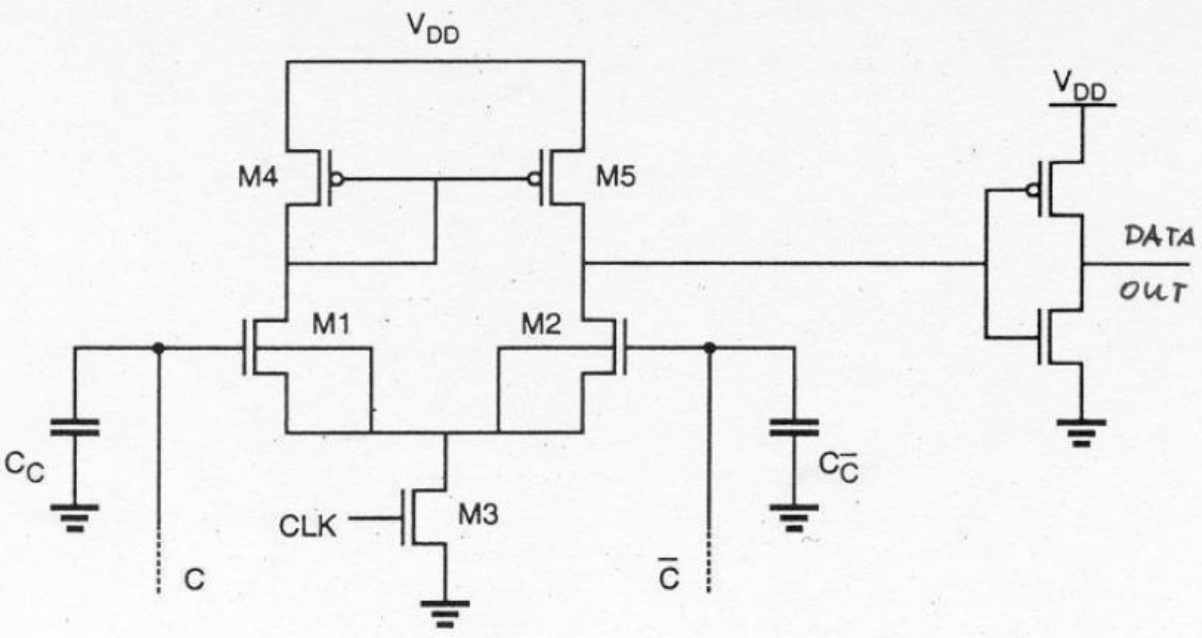
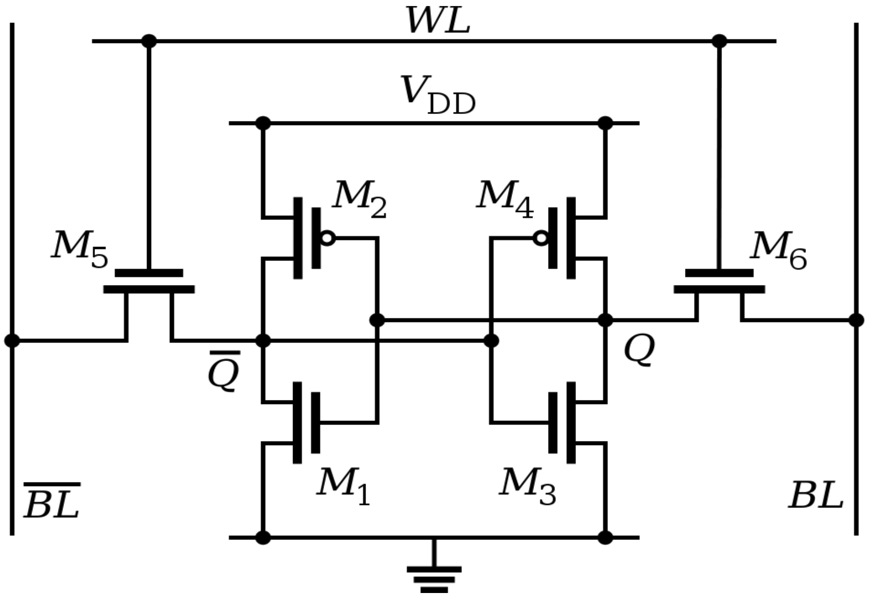


Figure 4.41: A CMOS differential current-mirror sense amplifier.

**6-Transistor Cell (Cross Coupled Inverter)**

- For larger SRAM modules the above circuit is not very efficient
    - o Transistor count per bit is too high



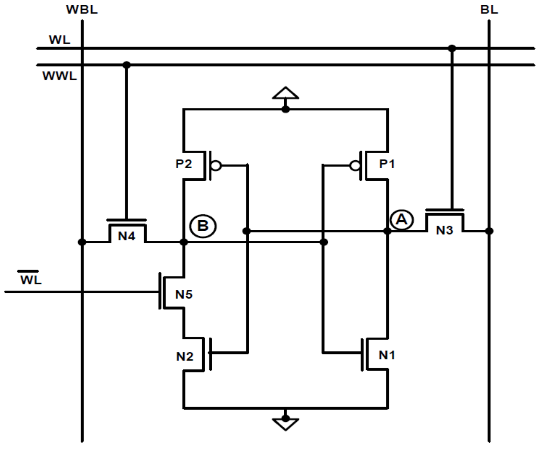
**TO READ:**

- BIT lines are charged high
- Enable line WL is pulled high, switching access transistors M5 and M6 on`
- If value stored in /Q is 0, value is accessed through access transistor M5 on /BL.
- If value stored in Q is 1, charged value of Bit line BL is pulled up to VDD.
- Value is ‘sensed’ on BL and /BL.

**TO WRITE:**

- Apply value to be stored to Bit lines BL and /BL
- Enable line WL is triggered and input value is latched into storage cell
- BIT line drivers must be stronger than SRAM transistor cell to override previous values

While Enable line is held low, the inverters retain the previous value could use tri-state WE line on BIT to drive into specific state. Transistor count per bit is only 6 + (line drivers & sense logic).



Seven Transistor Memory Cell

### 4.15 Dynamic Read-Write Memory (DRAM)

In the static CMOS read-write memory data is stored in six-transistor cells. Such a memory is fast and consumed small amount of static power. The only problem is that a SRAM cell occupies a significant amount of silicon space. This problem is addressed in the dynamic read-write memory (DRAM).

In a dynamic RAM binary data is stored as charge in a capacitor. The memory cell consists of a storage capacitor and an access transistor as shown in Figure 4.42.

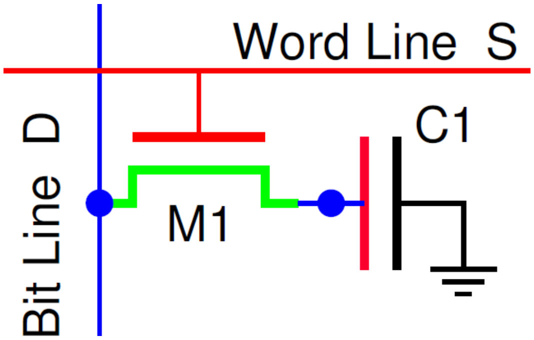


Figure 4.42: A one-transistor DRAM memory cell

Data stored as charge in a capacitor can be retained only for a limited time due to the leakage current which eventually removes or modifies the charge. Therefore, all dynamic memory cells require a periodic refreshing of the stored data before unwanted stored charge modifications occur. Typical storage capacitance has a value of 20 to 50 fF. Assuming that the voltage on the fully charged storage capacitor is V = 2.5V, and that the leakage current is I = 40pA, then the time to discharge the capacitor C = 20fF to the half of the initial voltage can be estimated as

Hence ever memory cell must be refreshed approximately every half millisecond. Despite of the

$$t = \frac{1}{2}\frac{C \cdot V}{I} = \frac{20 \cdot 10^{-15} \cdot 2.5}{40 \cdot 10^{-12}} = 0.625\mathrm{ms}$$

need for additional refreshing circuitry SRAM has two fundamental features which have determined is enormous popularity:

- The DRAM cell occupies much smaller silicon area than the SRAM cell. The size of a DRAM cell is in the order of $8F^2$, where F is the smallest feature size in a given technology. For F = 0.2µm the size is $0.32\mu m^2$
- No static power is dissipated for storing charge in a capacitance. The storage capacitance CS, which is connected between the drain of the access transistor (the storage node) and the ground, is formed as a trench or stacked Capacitor.

The stacked capacitor is created between a second polysilicon layer and a metal plate covering the whole array area. The plate is effectively connected to the ground terminal.To consider read/write operations we have to take into account a significant parasitic capacitance CC associated with each column, as shown in Figure 4.43.

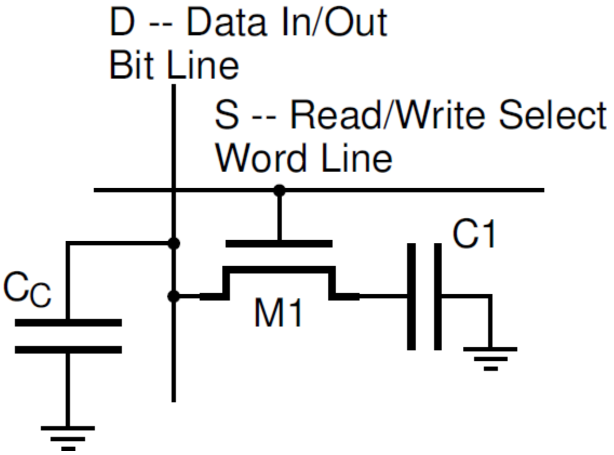


Figure 4.43: A single SRAM cells with a column capacitance shown.

Typically, before any operation is performed each column capacitance b is precharged high.

The cell is selected for a read/write operation by asserting its word line high (S = 1). This connects the storage capacitance to the bit line. The write operation is performed by applying either high or low voltage to the bit line thus charging (write '1') or discharging (write '0') the storage capacitance through the access transistor.

During read operation there is a flow of charges between the storage capacitance C1 and the column capacitance, CC. As a result the column voltage either increases (read '1') or decreases (read '0') slightly. This difference can then be amplified by the sense amplifier. Note that the read operation destroys the charge stored on the storage capacitance C1 ("destructive readout"). Therefore the data must be restored (refreshed) each time the read operation is performed.

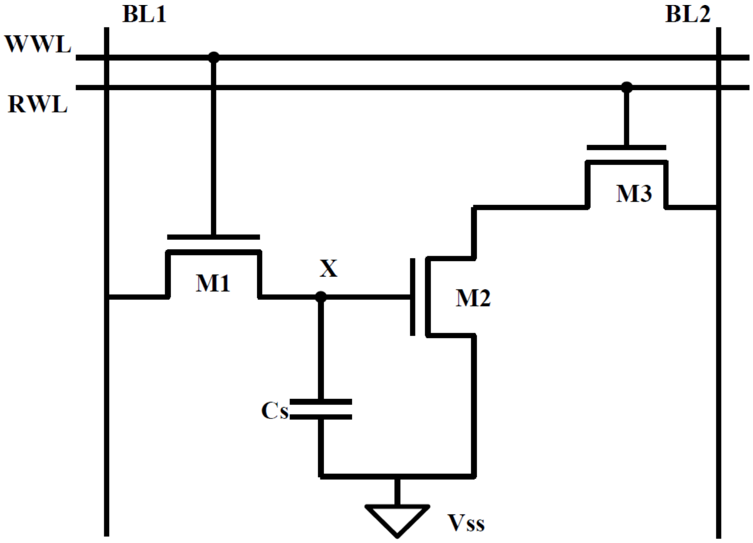
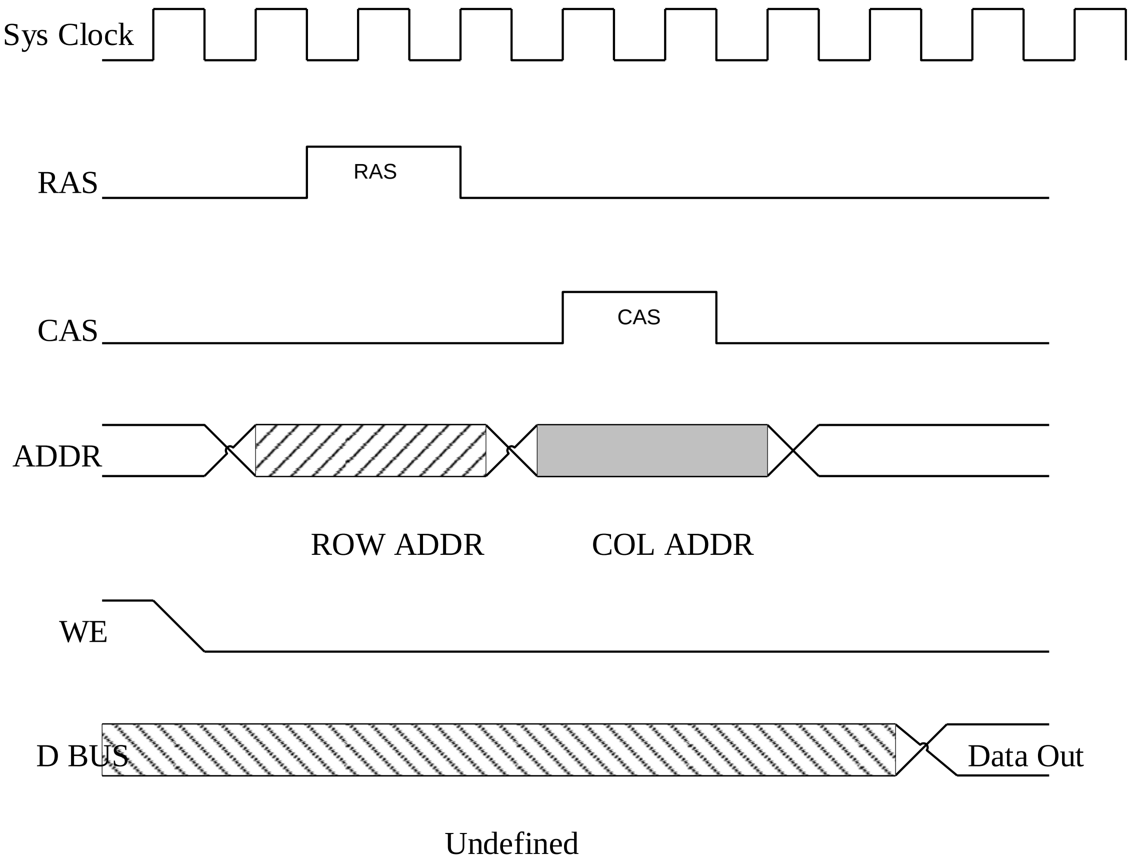


Figure 4.44 Three Transistor Dynamic RAM

The write operation performed is shown for three transistor Dynamic RAM (Figure 1.2) as the appropriate data value is written on BL1 and asserting the write-wordline (WWL). The data is retained as charge on capacitance Cs once WWL is lowered. When reading the cell, the read-wordline (RWL) is raised. The storage transistor M2 is either on or off depending upon the stored value. The bitline BL2 is precharged to VDD before performing read operation. The series connection of M2 and M3 pulls BL2 low when a "1" is stored. BL2 remains high in the opposite case. The cell is inverting; that is, the inverse value of the stored signal is sensed on the bitline.

**DRAM Timing:**

- DRAM module is asynchronous
  o Timing depends on how long it takes to respond to each  operation.



DRAM cannot be read as fast (or as easy) as SRAM

### 4.16 Serial Access Memories (Serial Memories):

Unlike RAMs which are randomly write the data, serial memories restrict the order of access, which results in either faster access times, smaller area, or a memory with a special functionality.

### 4.17.1 Shift Registers

Shift registers are a type of sequential logic circuit, mainly for storage of digital data. They are a group of flip-flops connected in a chain so that the output from one flip-flop becomes the input of the next flip-flop. Most of the registers possess no characteristic internal sequence of states. All the flip-flops are driven by a common clock, and all are set or reset simultaneously. There are two types of shift registers; Serial-in-parallel-out and Parallel-in-serial-out.

4.17.2 **Serial-In-Parallel-Out:** In this kind of register, data bits are entered serially. The difference is the way in which the data bits are taken out of the register. Once the data are stored, each bit appears on its respective output line, and all bits are available simultaneously. A r is shown below.

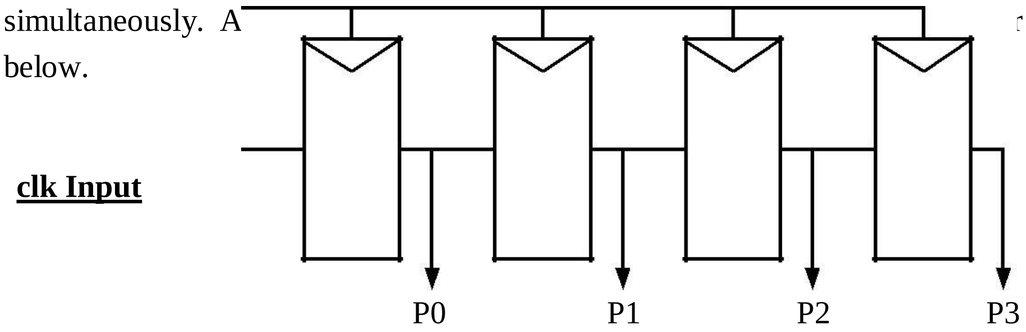


Figure 1.4 Serial-in-parallel-out Shift Register

4.17.3 **Parallel-In-Serial-Out:** The figure shown below is an example of Parallel-In- Serial-Out shift register. P0, P1, P2 and P3 are the parallel inputs to the shift register. When Shift = „0‟ the shift register loads all the inputs. When Shift = „1‟ the inputs are shifted to right. This shift register shift one bit per cycle.

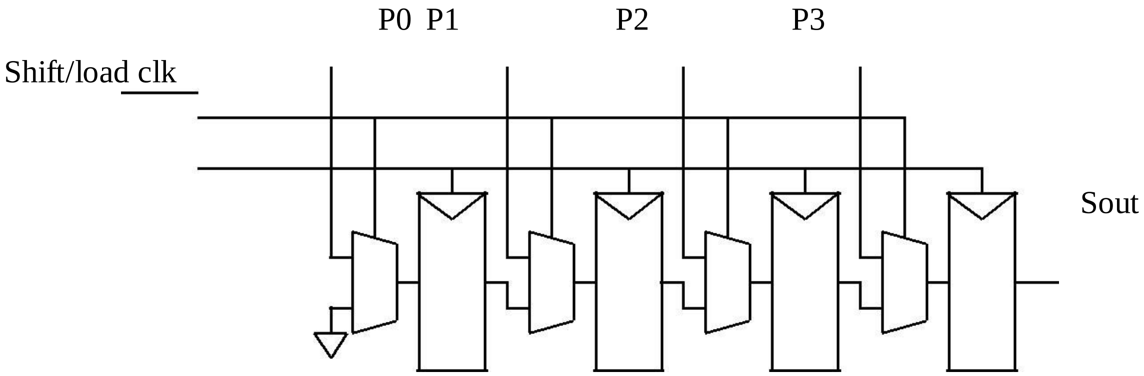


Figure 1.5 Parallel-In-Serial-Out Shift Register

4.17.4 **Queues:**A queue is a pile in which items are added a one end and removed from the other. In this respect, a queue is like the line of customers waiting to be served by a bank teller. As customers arrive, they join the end of the queue while the teller serves the customer at the head of the queue. The major advantage of queue is that they allow data to be written at different rates. The read and write use their own clock and data. There is an indication in queue when it is full or empty. These kind of queues usually built with SRAM and counters. There are two types of queues they are First-In-First-Out and Last-In First-Out.

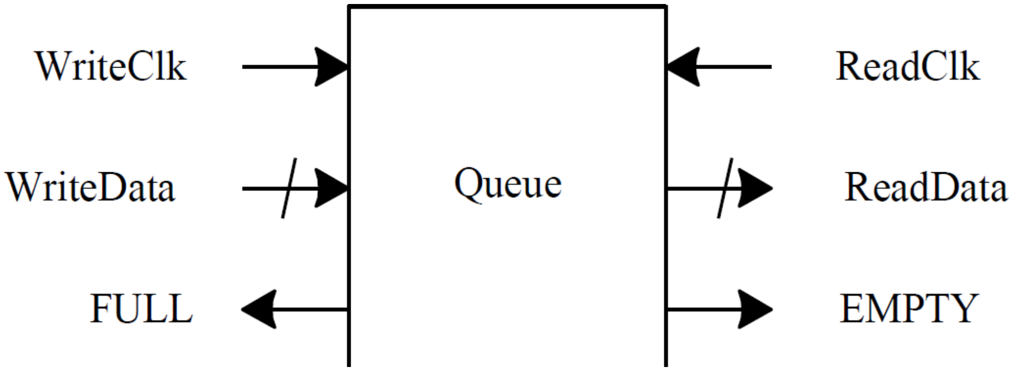


Figure 1.6 Queue

4.17.5 **First-In-First-Out:** In this method initialize the read and write pointers to the first element. Check whether the queue is empty. In write mode we will assign the write pointer and increment the write pointer. If the write almost catches read then queue is full. In read mode we will increment the read pointer.

4.17.6 **Last-In-First-Out:** It is also called as stack; objects which are stored in a stack are kept in a pile. The last item put into the stack is at the top. When an item is pushed into a stack, it is placed at the top of the pile. When an item popped, it is always the top item which is removed. Since it is always the last item to be put into the stack that is the first item to be removed, it is last-in, first-out.

### ***4.17*** *Contents-Addressable Memory (CAM)*

It is another important classification of nonrandom access memories. Instead of using an address to locate a data CAM uses a word of data itself as input in a query-style format. When the input data matches a data word stored in the memory array, a MATCH flag is raised. The MATCH signal remains low if no data stored in the memory corresponds to the input word. This type of memory is also called as associative memory and they are an important component of the cache architecture of many microprocessors.

The Figure 1.7 is an example of 512-word CAM architecture. It supports three modes of operation read, write and match. The read and write modes access and manipulate the data same as in an ordinary memory. The match mode is a special function of associative memory. The data patterns are stored in the comparand block which are needed to match and the mask word indicated which bits are significant. Every row that matches the pattern is passed to the validity block.

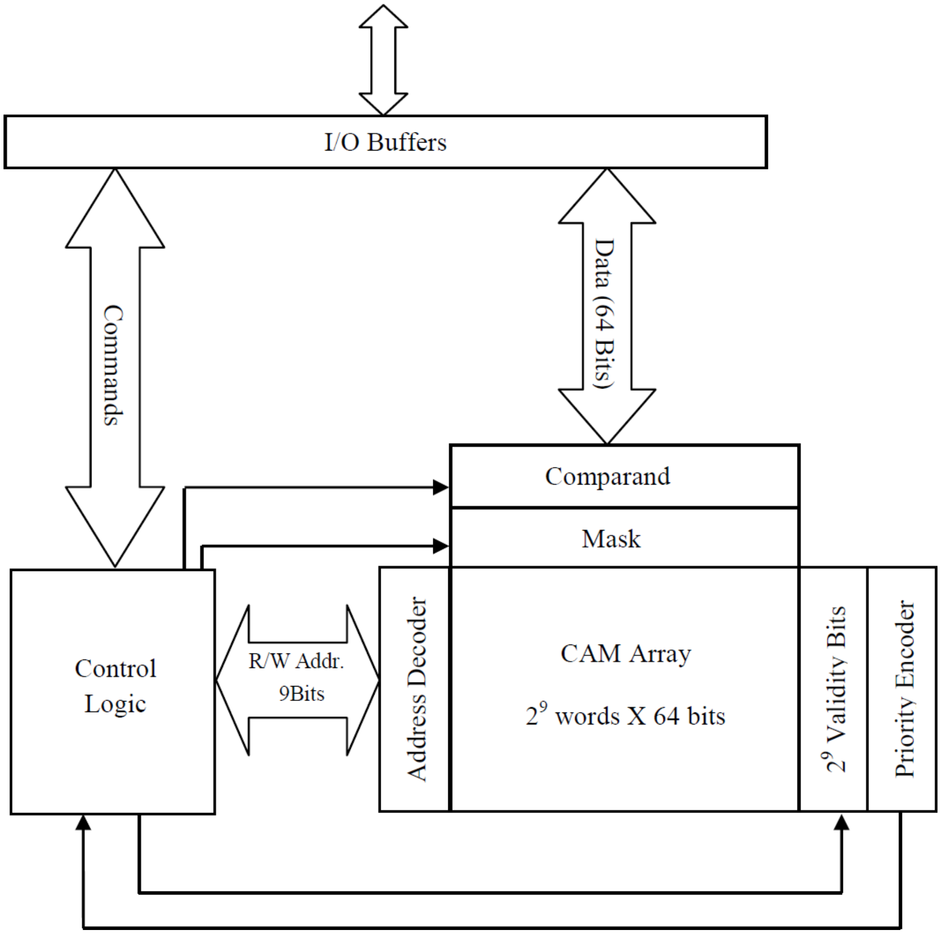


Figure 1.7 Architecture of 512-word CAM

The valid rows that match are passed to the priority encoder leaving the rows that contain invalid data. In the event that two or more rows match the pattern, the address of the row in the CAM is used to break the tie. In order to do that priority encoder considers all the 512 match lines from the CAM array, selects the one with the highest address, and encodes it in binary. Since there are 512 rows in CAM array, it needs 9 bits to indicate the row that matched. There is a possibility that none of the rows matches the pattern so there is one additional match found‟ bit provided.

# UNIT-V
# VHDL Synthesis and Test and Testability

## HDL

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

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

(a) VHDL

(b) Verilog

Traditional Methods of Hardware Design:

- ✓ Design with Boolean equations
- ✓ Schematic based design

**Features of VHDL**

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

**VHDL SYNTHESIS**

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

**CIRCUIT SYNTHESIS**

Circuit synthesis has the following steps:

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

**Translation:**

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

**Boolean optimization:**

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

**Flattening**

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

**Mapping to gates**

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

## SIMULATION

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

Need to test the designs prior to implementation and usage.

Reduce the time for development

Decrease the time to market.

## DESIGN CAPTURE TOOLS

**HDL Design**

- ✓ Schematic Design
- ✓ Floorplanning

**HDL Design**

HDLs are used to design two kinds of systems:

- ✓ Integrated Circuit
- ✓ Programmable Logic Devices

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

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

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

**Schematic Design**

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

**DESIGN VERIFICATION TOOLS**

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

1. **Simulation**
    - ❖ Circuit Level Simulation
    - ❖ Timing Simulation
    - ❖ Logical Level Simulation
    - ❖ Mixed mode Simulation
2. Timing verifiers
3. Netlist comparison
4. Layout extraction
5. Design rule verification

**Schematic Rule Check (SRC)**

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

**Design Rule Check (DRC)**

- The mask database provides interface between the semiconductor and chip designer. Two important requirements for this interface are:
  1. Specified geometric design
  2. Inter relationships of the mask
- The test for above two requirements are carried out by a CAD tools called DRC.
- Two different categories of DRC programs are used
  1. Polygonal check
  2. Raster scan check
- The polygonal design rule checks involves various mathematical operations during the check.

## DESIGN FOR TESTABILITY (DFT)

Test engineers usually have to construct test vectors after the design is completed. This invariably requkes a substantial amount of time and effort that could be avoided if testing is considered early in the design flow to make the design more testable. As a result, mtegration of design and test, refeued to as design for testability (DFT), was proposed in the 1970s. To structurally test circuits, we need to control and observe logic values of internal lines. Unfortunately, some nodes in sequential circuits can be very difficult to conŭol and observe; for example, activity on the most significant bit of an n bit counter can only be observed after 2"“" clock cycles. Testability measures of conŭollability and observability were first defined in the 1970s to help find those parts of a digital circuit that will be most difficult to test and to assist in test pattern generation for fault detection. Many DFT techniques have been proposed since that time.

DFT techniques generally fall into one of the following three categories:

(1) Ad hoc DFT techniques

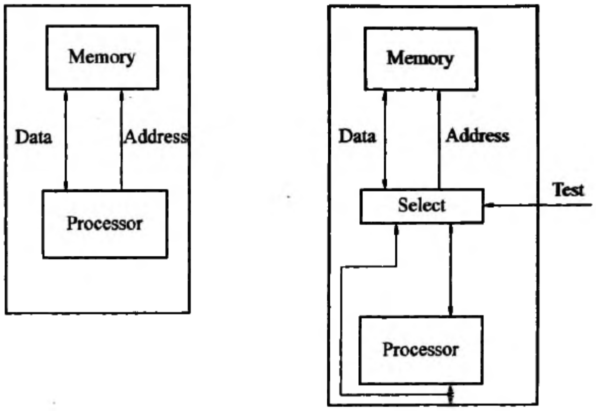
(2) Level-sensitive scan design (LSSD) or scan design

(3) Built-in self-test (BIST)

### Ad hoc DFT Techniques

ad hoc testing combines a collection of tricks and techniques that can be used to increase the observability and controllability of a design and that are generally applied in an application-dependent fashion.

An example of such a technique is illustrated in Fig. 9.6(a), which shows a simple processor with its data memory. Under normal configuration, the memory is only accessible through the



(a) Design with low testability (b) Adding a multiplexer (selector) improves testability.

Processor. Writing and reading a data value into and out of a single memory position requires a number of clock cycles. The controllability and observability of the memory can be dramatically improved by add multiplexers on the data and address buses (Fig. 9.6).

During normal operation mode, these selectors direct the memory ports to the processor.

During test, the data and address ports are connected directly to the I/O pins, and testing the memory can proceed more efficiently. The example illusùrates some important design-for testability concepts. It is often worthwhile to mtroduce extra hardware that has no functionality except improving the testability. Designers are often willing to incur a small penalty in area and performance if it makes the design substantially more observable or controllable. Design-for-testability often means that extra I/O pins must be provided besides die nominal functional I/O pins. The test port in Fig. 9.6(b) is such an extra pin. To reduce the number of extra pads that would be required, one can multiplex test signals and functional signals on the same pads. For example, the I/O bus in Fig. 9.6(b) serves as a data bus during normal operation and provides and collects the test patterns during testing.

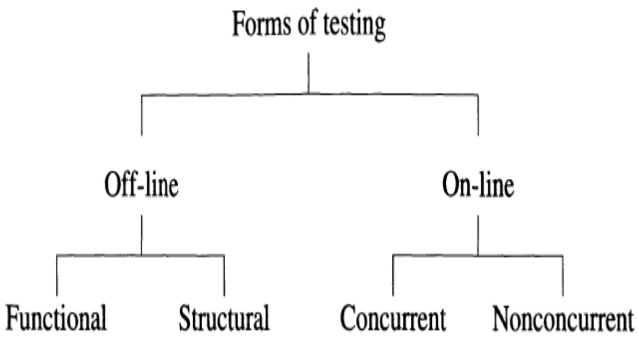
An extensive collection of ad hoc test approaches has been devised. Examples include tìie partitioning of large state machines, addition of extra test points, prevision of reset states, and introduction of test buses. While very effective, the applicability of most of these techniques depends upon the application and architecture at hand. The insertion into a given Hating requires expert knowledge and is difficult to automate. Stricture and automatable approaches are more desirable.

## BUILT-IN SELF-TEST (BIST)

- ✓ Built-in self-test is the capability of a circuit (chip, board, or system) to test itself. BIST represents a merger of the concepts of built-in test (BIT) and self-test.
- ✓ BIST techniques can be classified into two categories, namely
  i. **On-line BIST**, which includes **concurrent and non-concurrent techniques**,
  ii. **Off-line BIST**, which includes **functional and structural approaches**.
- ✓ **In on-line BIST,** testing occurs during normal functional operating conditions; i.e., the circuit under test (CUT) is not placed into a test mode where normal functional operation is locked out. Concurrent on-line BIST is a form of testing that occurs simultaneously with normal functional operation. In non-concurrent on-line BIST, testing is carried out while a system is in an idle state. This is often accomplished by

executing diagnostic software routines (macrocode) or diagnostic firmware routines (microcode). The test process can be interrupted at any time so that normal operation can resume.

- ✓ **Off-line BIST** deals with testing a system when it is not carrying out its normal functions. Systems, boards, and chips can be tested in this mode. This form of testing is also applicable at the manufacturing, field, depot, and operational levels. Often Off-line testing is carried out using on-chip or on-board test-pattern generators (TPGs) and output response analyzers (ORAs). Off-line testing does not detect errors in real time, i.e., when they first occur, as is possible with many on-line concurrent BIST techniques.



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

**Off-Line BIST Architectures**

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

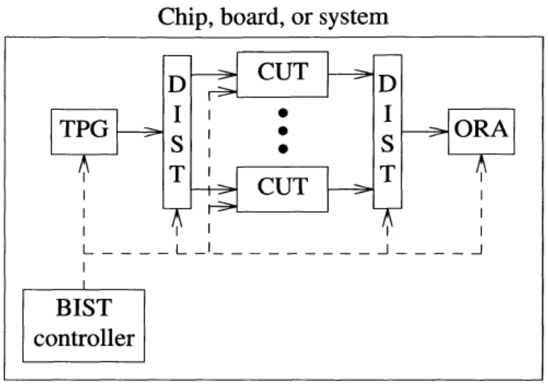
1. Centralized or distributed BIST circuitry;
2. Embedded or separate BIST elements.

BIST architectures consist of several key elements, namely

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

**Centralized BIST architecture**

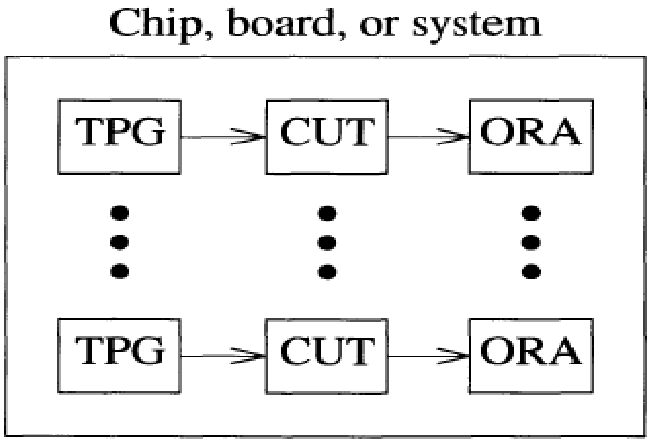
The general form of a centralized BIST architecture is shown in the below figure.



Here several CUTs share TPG and ORA circuitry. This leads to reduced overhead but increased test time. During testing, the BIST controller may carry out one or more of the following functions:

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

**Distributed BIST architecture**



The distributed BIST architecture is shown in above figure. Here each CUT is associated with its own TPG and ORA circuitry. This leads to more overhead but less test time and usually more accurate diagnosis.

**Advantages of BIST**

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