Fundamentals of CMOS VLSI

Sub code: 10EC56 IA Marks:25
No. of lecture Hrs/Week: 04 Exam Hours:03
Total Hours:52 Exam Marks: 100

PART-A

Unit 1: Basic MOS Technology

Integrated circuits era, enhancement and depletion mode MOS transistors. nMOS fabrication. CMOS fabrication, Thermal aspects of processing, BiCMOS technology, production of E-beam masks. **3 Hours**

MOS transistor theory

Introduction, MOS device design equations, the complementary CMOS inverter-DC characteristics, static load MOS inverters, the differential inverter, the transmission gate, tristate inverter. **4 Hours**

Unit-2: Circuit Design Processes

MOS layers, stick diagrams, Design rules and layout- lambda-based design and other rules. Examples, layout diagrams, symbolic diagram, tutorial exercises. **4 Hours**

Basic physical design of simple logic gates. **3 Hours**

Unit 3: CMOS Logic Structures

CMOS complementary logic, BiCMOS logic, Pseudo-nMOS logic, Dynamic CMOS logic, clocked CMOS logic, Pass transistor logic, CMOS domino logic cascaded voltage switch logic (CVSL). **6 Hours**

Unit-4: Basic circuit concepts

Sheet resistance, area capacitances, capacitances calculations. The delay unit, inverter delays, driving capacitive loads, propagation delays, wiring capacitances.

3 Hours

Scaling of MOS circuits

Scaling models and factors, limits on scaling, limits due to current density and noise.

3 Hours

PART-B

Unit-5: CMOS subsystem design

Architectural issues, switch logic, gate logic, design examples-combinational logic, clocked circuits. Other system considerations. **3 Hours**

Clocking strategies

2 **Hours**

Unit-6: CMOS subsystem design processes

General considerations, process illustration, ALU subsystem, adders, multipliers.

6 Hours

Unit-7: Memory registers and clock

Timing considerations, memory elements, memory cell arrays. **6 Hours**

Unit-8: Testability

Performance parameters, layout issues I/O pads, real estate, system delays, ground rules for design, test and testability. 7 **Hours**

TEXT BOOKS

1. Douglas A. Pucknell & Kamran Eshraghian, **"Basic VLSI Design"** PHI 3rd Edition (original Edition – 1994), 2005.

2. Neil H. E. Weste and K. Eshragian**," Principles of CMOS VLSI Design: A System Perspective,"** 2nd edition, Pearson Education (Asia) Pvt. Ltd., 2000. History of VLSI

3. CMOS VLSI DESIGN—A circuits and systems perpective. 3rd edition N.H.Weste and David Harris. Addison-wesley.

REFERENCE BOOKS

1. **R.Jacob Baker**.CMOS circuit design, layout and simulation.

2. Fundamentals of semiconductor devices: **M.K.Achuthan and K.N.Bhat.**

3. CMOS digital Integrated circuits: Analysis and design: **Sung-Mo Kang and Yusuf Leblebici**.

4. Analysis and design of digital integrated circuits: **D.A.Hodges, Jackson and Saleh**.

INDEX SHEET

Unit 1

Basic MOS Technology

Integrated circuits era, enhancement and depletion mode MOS transistors. nMOS fabrication. CMOS fabrication, Thermal aspects of processing, BiCMOS technology, production of E-beam masks.

MOS transistor theory

Introduction, MOS device design equations, the complementary CMOS inverter-DC characteristics, static load MOS inverters, the differential inverter, the transmission gate, tristate inverter.

Recommended readings:

1. Douglas A. Pucknell & Kamran Eshraghian, **"Basic VLSI Design"** PHI 3rd Edition (original Edition – 1994), 2005.

2. Neil H. E. Weste and K. Eshragian,**" Principles of CMOS VLSI Design: A Systems Perspective,"** 2nd edition, Pearson Education (Asia) Pvt. Ltd., 2000. History of VLSI.

1.1 Integrated circuits era

Transistor was first invented by William.B.Shockley, Walter Brattain and John Bardeen of Bell laboratories. In 1961, first IC was introduced.

Levels of Integration:-

i) SSI: - (10-100) transistors => Example: Logic gates

ii) MSI: - (100-1000) => Example: counters

iii) LSI: - (1000-20000) => Example: 8-bit chip

iv) VLSI: - (20000-1000000) => Example: 16 & 32 bit up

v) ULSI: - (1000000-10000000) => Example: Special processors, virtual reality machines, smart sensors.

Moore's Law:-

"The number of transistors embedded on the chip doubles after every one and a half years." The number of transistors is taken on the y-axis and the years in taken on the x-axis. The diagram also shows the speed in MHz. the graph given in figure also shows the variation of speed of the chip in MHz.

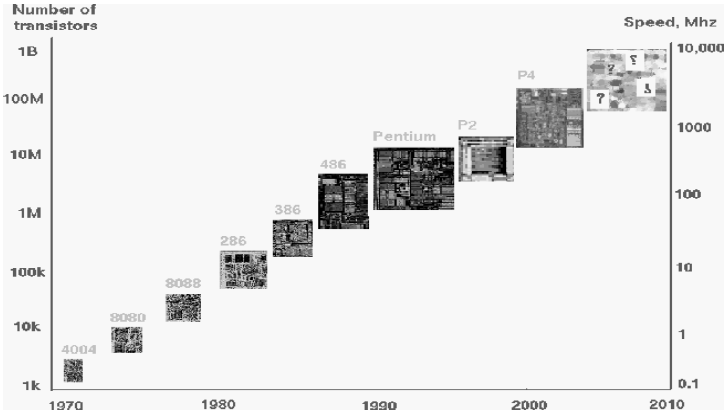


Figure 1. Moore's law.

The graph in figure2 compares the various technologies available in ICs.

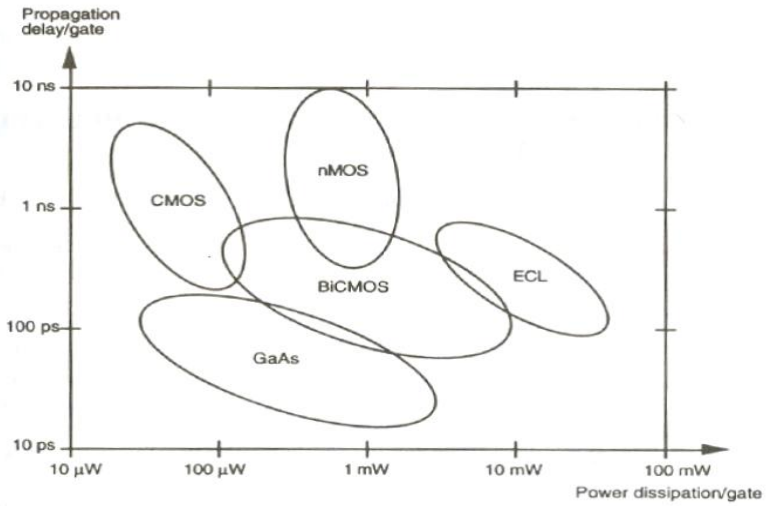


Figure 2.Comparison of available technologies.

From the graph we can conclude that GaAs technology is better but still it is not used because of growing difficulties of GaAs crystal. CMOS looks to be a better option compared to nMOS since it consumes a lesser power. BiCMOS technology is also used in places where high

driving capability is required and from the graph it confirms that, BiCMOS consumes more power compared to CMOS.

Levels of Integration:-

i) Small Scale Integration:- (10-100) transistors => Example: Logic gates

ii) Medium Scale Integration:- (100-1000) => Example: counters

iii) Large Scale Integration:- (1000-20000) => Example:8-bit chip

iv) Very Large Scale Integration:- (20000-1000000) => Example:16 & 32 bit up

v) Ultra Large Scale Integration:- (1000000-10000000) => Example: Special processors, virtual reality machines, smart sensors

1.2 Basic MOS Transistors:

MOS

We should first understand the fact that why the name Metal Oxide Semiconductor transistor, because the structure consists of a layer of Metal (gate), a layer of oxide (Sio2) and a layer of semiconductor. Figure 3 below clearly tell why the name MOS.

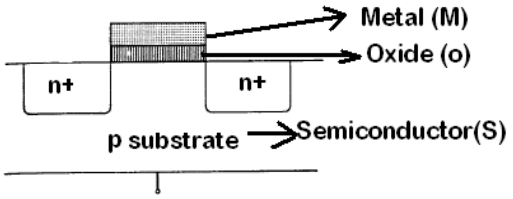


Figure 3.cross section of a MOS structure

We have two types of FETs. They are Enhancement mode and depletion mode transistor. Also we have PMOS and NMOS transistors.

In **Enhancement mode transistor** channel is going to form after giving a proper positive gate voltage. We have NMOS and PMOS enhancement transistors.

In **Depletion mode transistor** channel will be present by the implant. It can be removed by giving a proper negative gate voltage. We have NMOS and PMOS depletion mode transistors.

1.2.1 N-MOS enhancement mode transistor:-

This transistor is normally off. This can be made ON by giving a positive gate voltage. By giving a +ve gate voltage a channel of electrons is formed between source drain.

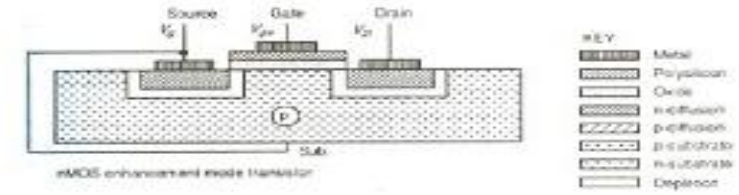


Figure 4. N-MOS enhancement mode transistor.

1.2.2 P-MOS enhancement mode transistor:-

This is normally on. A Channel of Holes can be performed by giving a –ve gate voltage. In P-Mos current is carried by holes and in N-Mos it's by electrons. Since the mobility is of holes less than that of electrons P-Mos is slower.

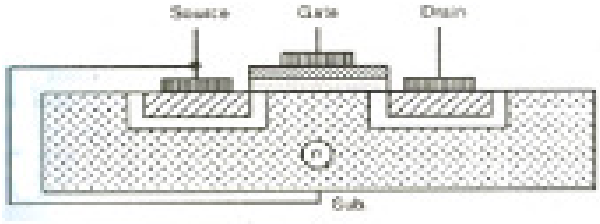


Figure 5. P-MOS enhancement mode transistor.

1.2.3 N-MOS depletion mode transistor:-

This transistor is normally ON, even with Vgs=0. The channel will be implanted while fabricating, hence it is normally ON. To cause the channel to cease to exist, a – ve voltage must be applied between gate and source.

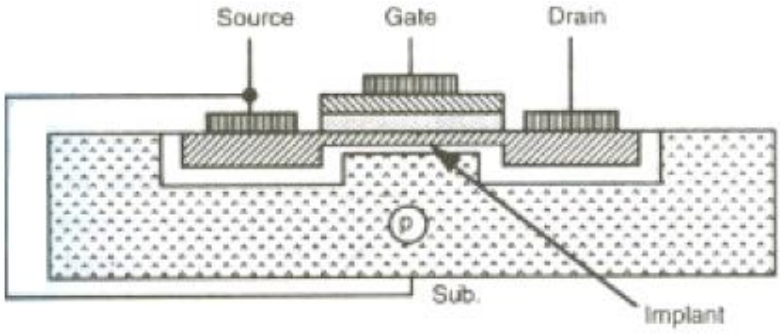


Figure 6. N-MOS depletion mode transistor

NOTE: Mobility of electrons is 2.5 to 3 times faster than holes. Hence P-MOS devices will have more resistance compared to NMOS.

1.2.4 Enhancement mode Transistor action:-

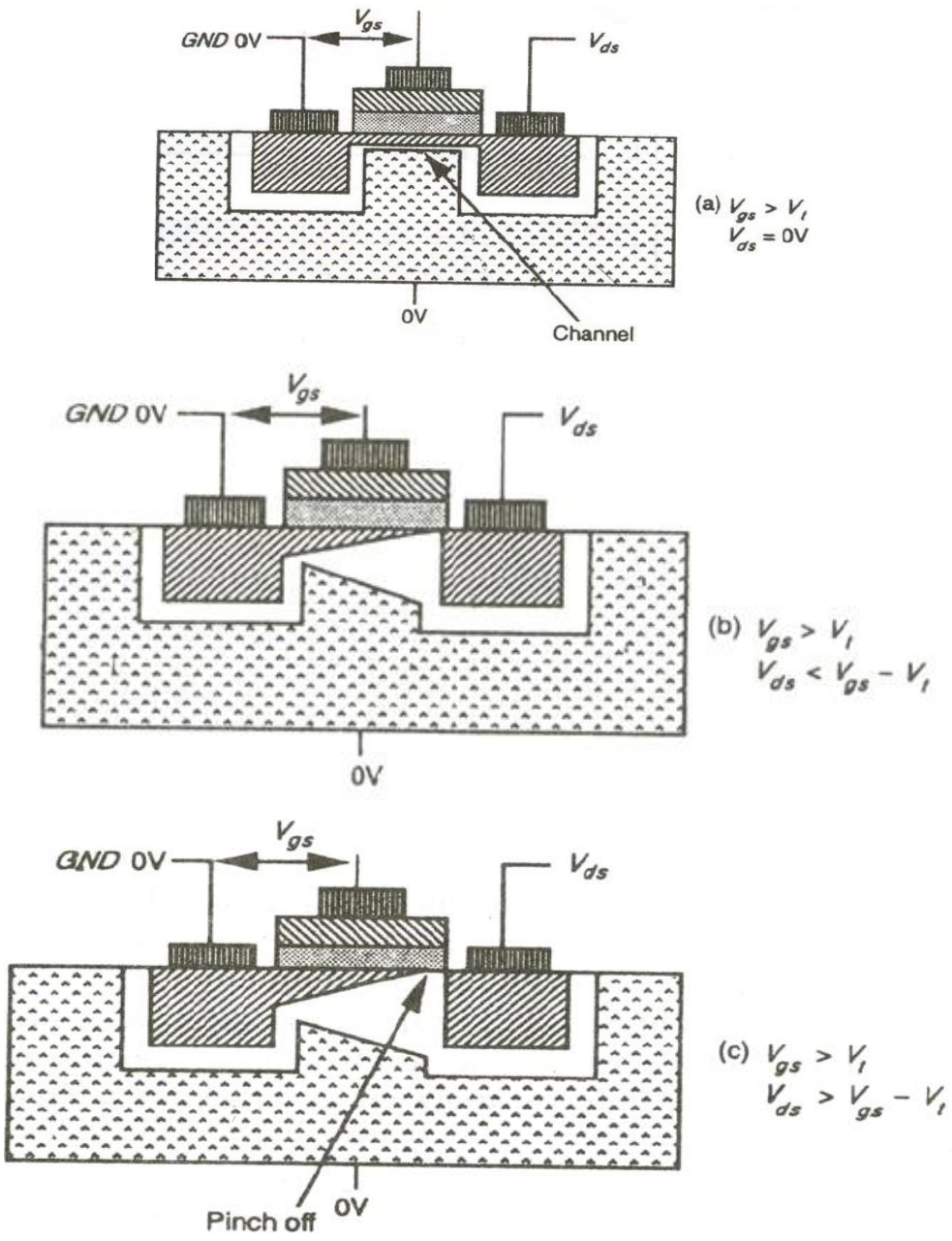


Figure7. (a)(b)(c) Enhancement mode transistor with different Vds values

To establish the channel between the source and the drain a minimum voltage (Vt) must be applied between gate and source. This minimum voltage is called as "Threshold Voltage". The complete working of enhancement mode transistor can be explained with the help of diagram a, b and c.

a) Vgs > Vt
Vds = 0

Since Vgs > Vt and Vds = 0 the channel is formed but no current flows between drain and source.

b) Vgs > Vt

Vds < Vgs - Vt

This region is called the non-saturation Region or linear region where the drain current increases linearly with Vds. When Vds is increased the drain side becomes more reverse biased (hence more depletion region towards the drain end) and the channel starts to pinch. This is called as the pinch off point.

c) Vgs > Vt

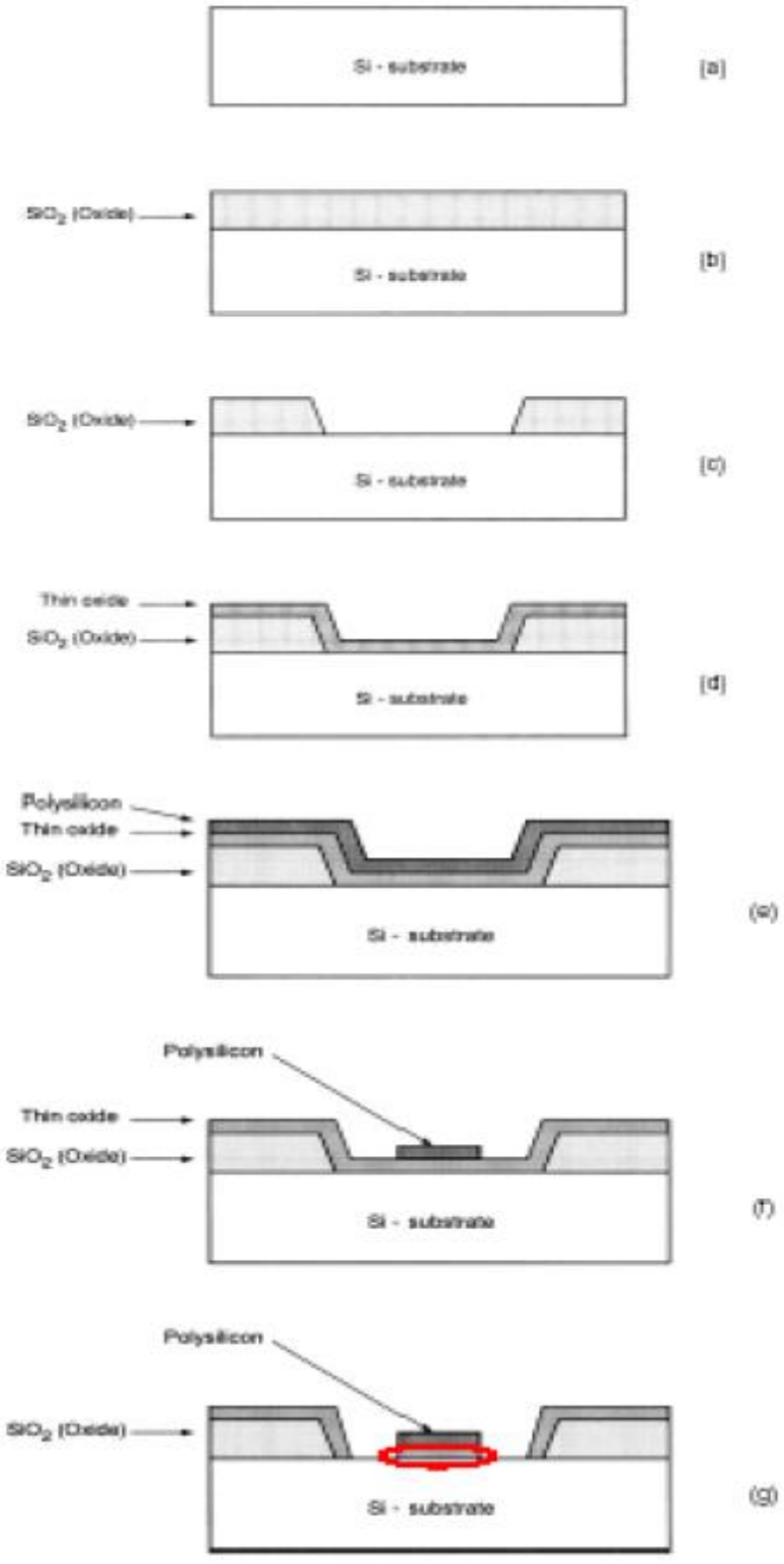
Vds > Vgs - Vt

This region is called Saturation Region where the drain current remains almost constant. As the drain voltage is increased further beyond (Vgs-Vt) the pinch off point starts to move from the drain end to the source end. Even if the Vds is increased more and more, the increased voltage gets dropped in the depletion region leading to a constant current. The typical threshold voltage for an enhancement mode transistor is given by Vt = 0.2 * Vdd.

1.2.5 Depletion mode Transistor action:-

We can explain the working of depletion mode transistor in the same manner, as that of the enhancement mode transistor only difference is, channel is established due to the implant even when Vgs = 0 and the channel can be cut off by applying a –ve voltage between the gate and source. Threshold voltage of depletion mode transistor is around 0.8*Vdd.

1.3 NMOS Fabrication:


Si - substrate
(a)
SiO_2 (Oxide)
Si - substrate
(b)
SiO_2 (Oxide)
Si - substrate
(c)
Thin oxide
SiO_2 (Oxide)
Si - substrate
(d)
Polysilicon
Thin oxide
SiO_2 (Oxide)
Si - substrate
(e)
Polysilicon
Thin oxide
SiO_2 (Oxide)
Si - substrate
(f)
Polysilicon
SiO_2 (Oxide)
Si - substrate
(g)

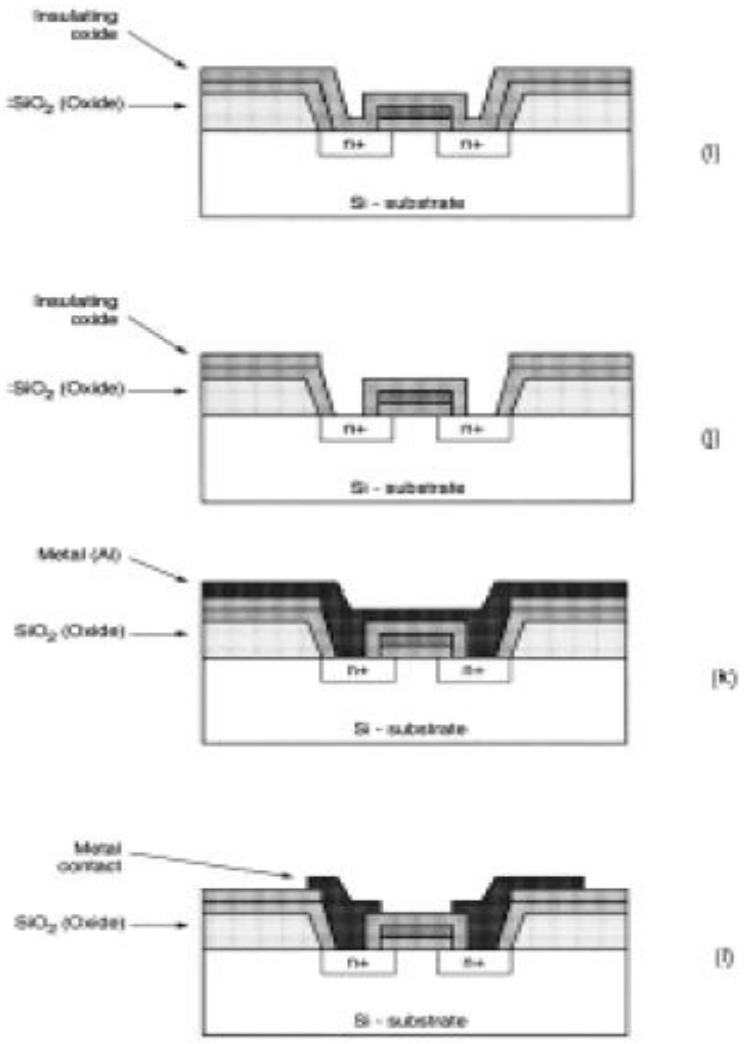


Figure8. NMOS Fabrication process steps

The process starts with the oxidation of the silicon substrate (Fig. 8(a)), in which a relatively thick silicon dioxide layer, also called field oxide, is created on the surface (Fig. 8(b)). Then, the field oxide is selectively etched to expose the silicon surface on which the MOS transistor will be created (Fig. 8(c)). Following this step, the surface is covered with a thin, high-quality oxide layer, which will eventually form the gate oxide of the MOS transistor (Fig. 8(d)). On top of the thin oxide, a layer of polysilicon (polycrystalline silicon) is deposited (Fig. 8(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.

After deposition, the polysilicon layer is patterned and etched to form the interconnects and the MOS transistor gates (Fig. 8(f)). The thin gate oxide not covered by polysilicon is also etched away, which exposes the bare silicon surface on which the source and drain junctions are to be formed (Fig. 8(g)). The entire silicon surface is then doped with a high concentration of impurities, either through diffusion or ion implantation (in this case with donor atoms to produce n-type doping). Figure 8(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. Note that the polysilicon gate, which is patterned before doping actually defines the precise location of the channel region and, hence, the location of the source and the drain regions. Since this procedure allows very precise positioning of the two regions relative to the gate, it is also called the self-aligned process.

Once the source and drain regions are completed, the entire surface is again covered with an insulating layer of silicon dioxide (Fig. 8 (i)). The insulating oxide layer is then patterned in order to provide contact windows for the drain and source junctions (Fig. 8 (j)). The surface is covered with evaporated aluminum which will form the interconnects (Fig. 8 (k)). Finally, the metal layer is patterned and etched, completing the interconnection of the MOS transistors on the surface (Fig. 8 (l)). Usually, a second (and third) layer of metallic interconnect can also be added on top of this structure by creating another insulating oxide layer, cutting contact (via) holes, depositing, and patterning the metal.

1.4 CMOS fabrication:

When we need to fabricate both nMOS and pMOS transistors on the same substrate we need to follow different processes. The three different processes are, P-well process ,N-well process and Twin tub process.

1.4.1 P-WELL PROCESS:

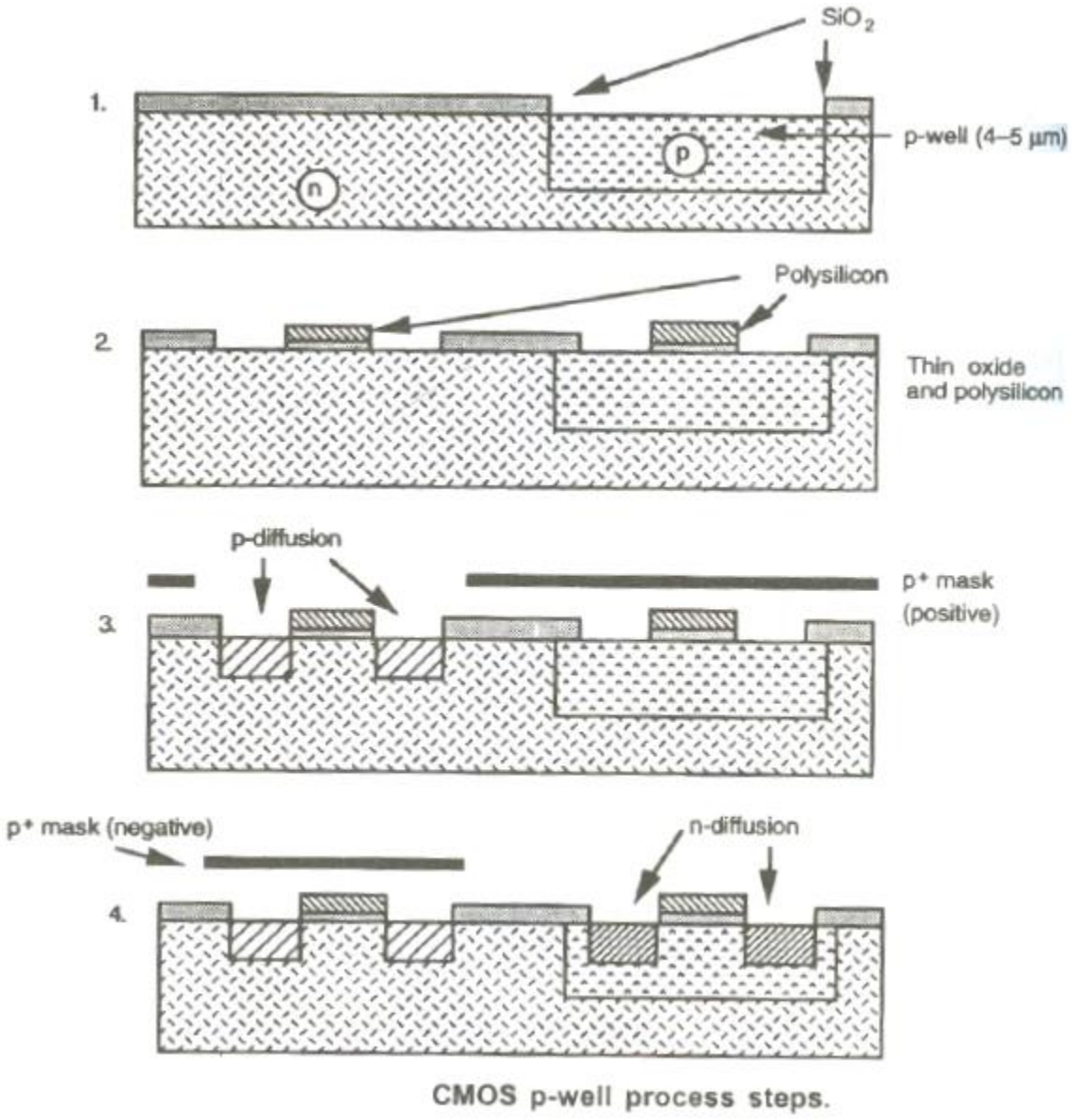


Figure9. CMOS Fabrication (P-WELL) process steps.

The p-well process starts with a n type substrate. The n type substrate can be used to implement the pMOS transistor, but to implement the nMOS transistor we need to provide a p-well, hence we have provided he place for both n and pMOS transistor on the same n-type substrate.

Mask sequence.

Mask 1:

Mask 1 defines the areas in which the deep p-well diffusion takes place.

Mask 2:

It defines the thin oxide region (where the thick oxide is to be removed or stripped and thin oxide grown)

Mask 3:

It's used to pattern the polysilicon layer which is deposited after thin oxide. Mask 4: A p+ mask (anded with mask 2) to define areas where p-diffusion is to take place.

Mask 5:

We are using the –ve form of mask 4 (p+ mask) It defines where n-diffusion is to take place.

Mask 6:

Contact cuts are defined using this mask.

Mask 7:

The metal layer pattern is defined by this mask.

Mask 8:

An overall passivation (over glass) is now applied and it also defines openings for accessing pads.

The cross section below shows the CMOS pwell inverter.

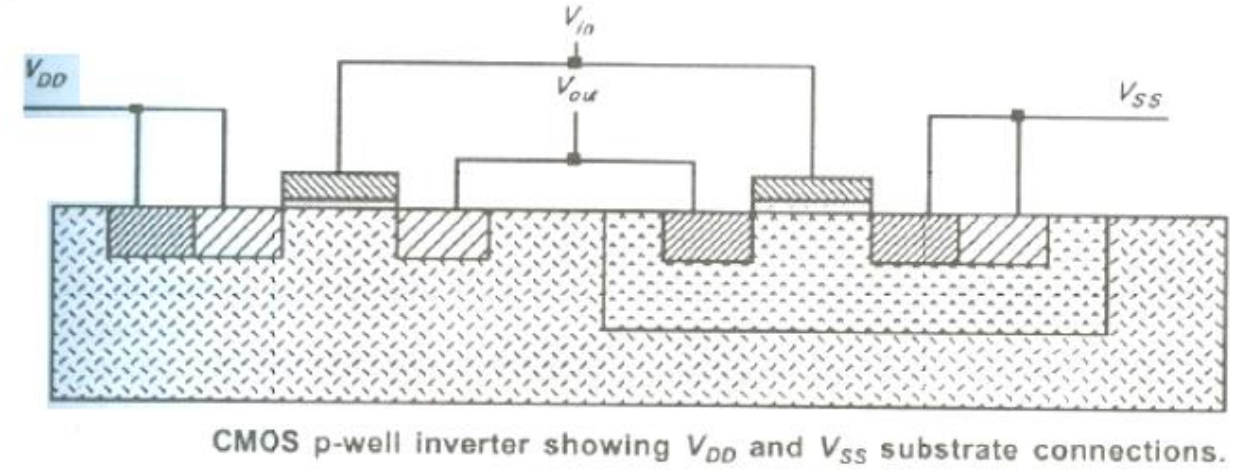


Figure10. CMOS inverter (P-WELL)

1.4.2 N-WELL PROCESS:

In the following figures, some of the important process steps involved in the fabrication of a CMOS inverter will be shown by a top view of the lithographic masks and a cross-sectional view of the relevant areas. The n-well CMOS process starts with a moderately doped (with impurity concentration typically less than 1015 cm-3) p-type silicon substrate. Then, an initial oxide layer is grown on the entire surface. The first lithographic mask defines the n-well region. Donor atoms, usually phosphorus, are implanted through this window in the oxide. Once the n-well is created, the active areas of the nMOS and pMOS transistors can be defined. Figures 12.1 through 12.6 illustrate the significant milestones that occur during the fabrication process of a CMOS inverter.

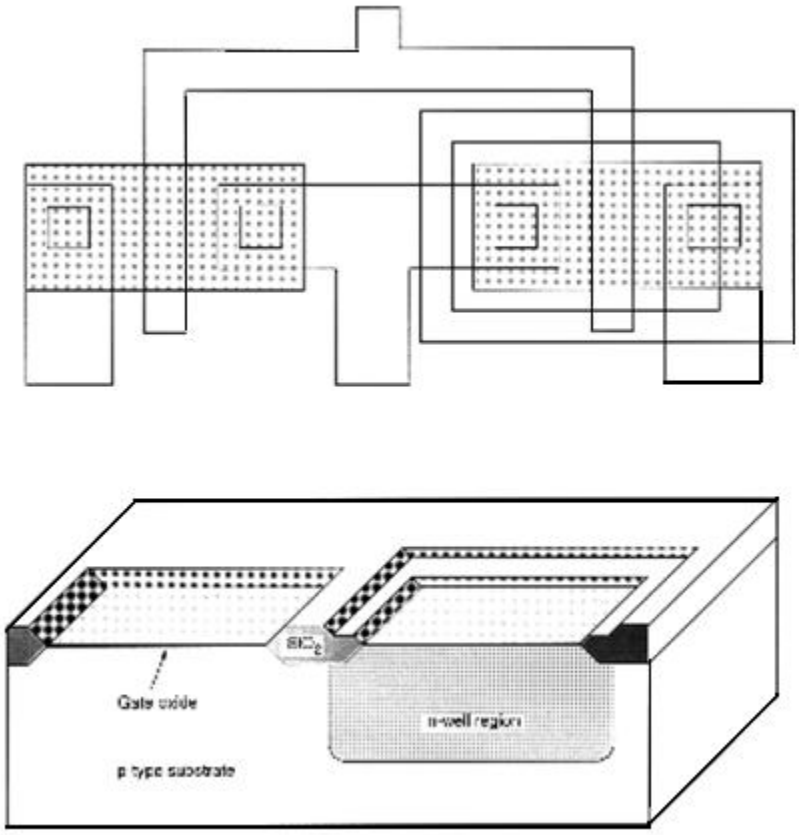


Figure-11.1: Cross sectional view

Following the creation of the n-well region, a thick field oxide is grown in the areas surrounding the transistor active regions, and a thin gate oxide is grown on top of the active regions.

The thickness and the quality of the gate oxide are two of the most critical fabrication parameters, since they strongly affect the operational characteristics of the MOS transistor, as well as its long-term reliability.

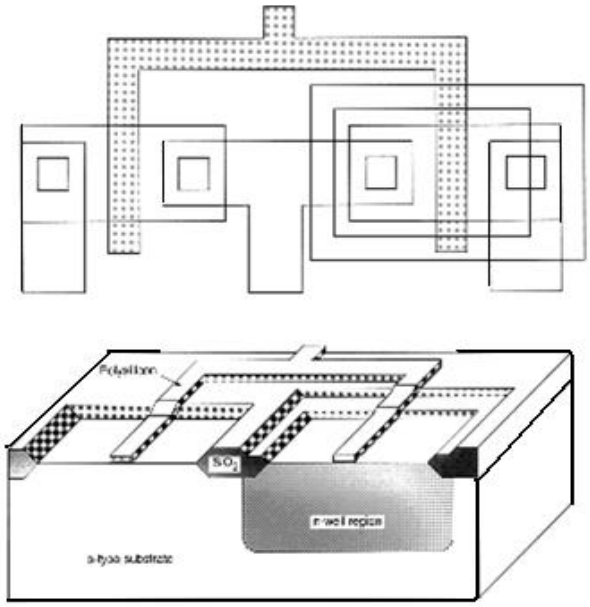


Figure-11.2: Cross sectional view

The polysilicon layer is deposited using chemical vapor deposition (CVD) and patterned by dry (plasma) etching. The created polysilicon lines will function as the gate electrodes of the nMOS and the pMOS transistors and their interconnects. Also, the polysilicon gates act as self-aligned masks for the source and drain implantations that follow this step.

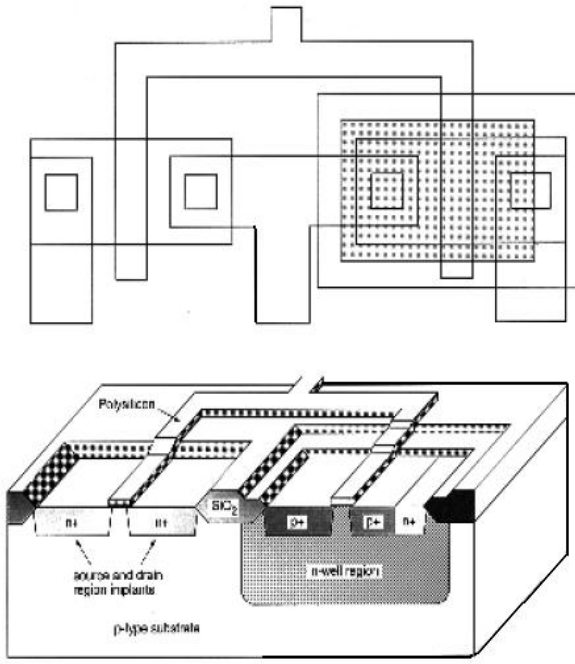


Figure-11.3: Using a set of two masks, the n+ and p+ regions are implanted into the substrate and into the n- well, respectively. Also, the ohmic contacts to the substrate and to the n-well are implanted in this process step.

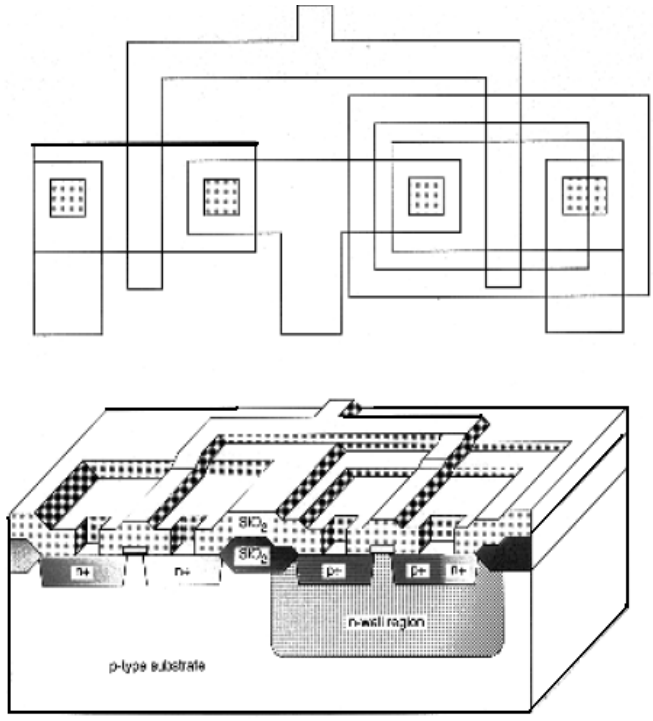


Figure-11.4: An insulating silicon dioxide layer is deposited over the entire wafer using CVD. Then, the contacts are defined and etched away to expose the silicon or polysilicon contact windows. These contact windows are necessary to complete the circuit interconnections using the metal layer, which is patterned in the next step.

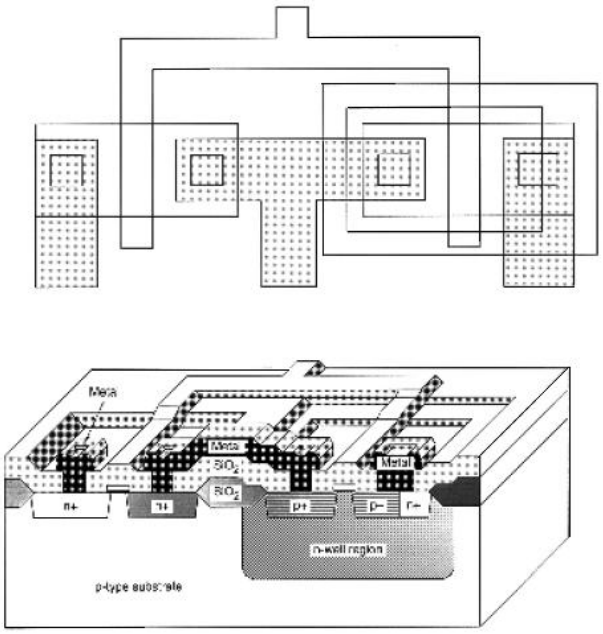


Figure-11.5: Metal (aluminum) is deposited over the entire chip surface using metal evaporation, and the metal lines are patterned through etching. Since the wafer surface is non-planar, the quality and the integrity of the metal lines created in this step are very critical and are ultimately essential for circuit reliability.

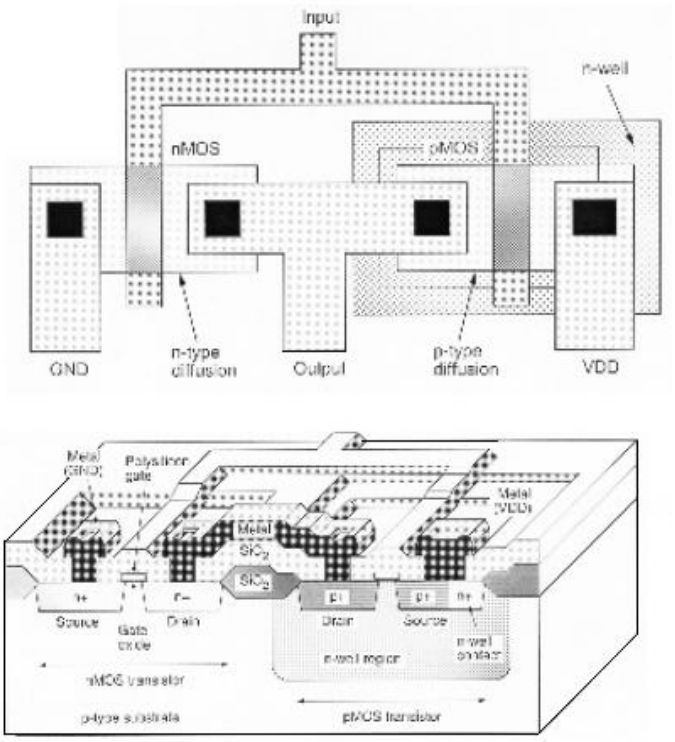


Figure-11.6: The composite layout and the resulting cross-sectional view of the chip, showing one nMOS and one pMOS transistor (built-in n-well), the polysilicon and metal interconnections. The final step is to deposit the passivation layer (for protection) over the chip, except for wire-bonding pad areas.

1.4.3Twin-tub process:

Here we will be using both p-well and n-well approach. The starting point is a n-type material and then we create both n-well and p-well region. To create the both well we first go for the epitaxial process and then we will create both wells on the same substrate.

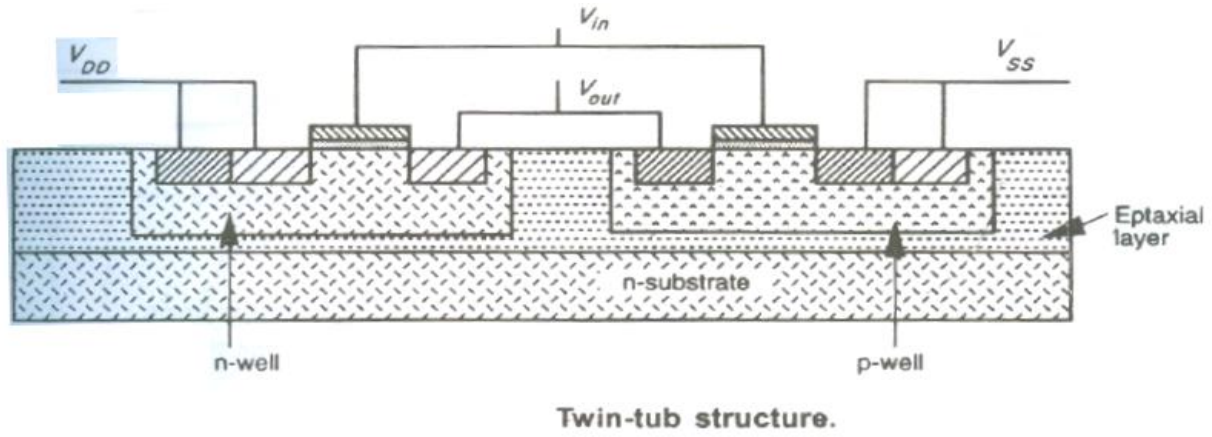


Figure 12 CMOS twin-tub inverter.

NOTE: Twin tub process is one of the solutions for latch-up problem.

1.5 Bi-CMOS technology: - (Bipolar CMOS)

The driving capability of MOS transistors is less because of limited current sourcing and sinking capabilities of the transistors. To drive large capacitive loads we can think of Bi-Cmos technology. This technology combines Bipolar and CMOS transistors in a single integrated circuit, by retaining benefits of bipolar and CMOS, BiCMOS is able to achieve VLSI circuits with speed-power-density performance previously unattainable with either technology individually.

Characteristics of CMOS Technology

- Lower static power dissipation
- Higher noise margins

- Higher packing density – lower manufacturing cost per device
- High yield with large integrated complex functions
- High input impedance (low drive current)
- Scalable threshold voltage
- High delay sensitivity to load (fan-out limitations)
- Low output drive current (issue when driving large capacitive loads)
- Low transconductance, where transconductance, gm a Vin
- Bi-directional capability (drain & source are interchangeable)
- A near ideal switching device

Characteristics of Bipolar Technology

- Higher switching speed
- Higher current drive per unit area, higher gain
- Generally better noise performance and better high frequency characteristics
- Better analogue capability
- Improved I/O speed (particularly significant with the growing importance of package limitations in high speed systems).
- High power dissipation
- Lower input impedance (high drive current)
- Low voltage swing logic
- Low packing density
- Low delay sensitivity to load
- High gm (gm a Vin)
- High unity gain band width (ft) at low currents
- Essentially unidirectional from the two previous paragraphs we can get a comparison between bipolar and CMOS technology.

The diagram given below shows the cross section of the BiCMOS process which uses an npn transistor.

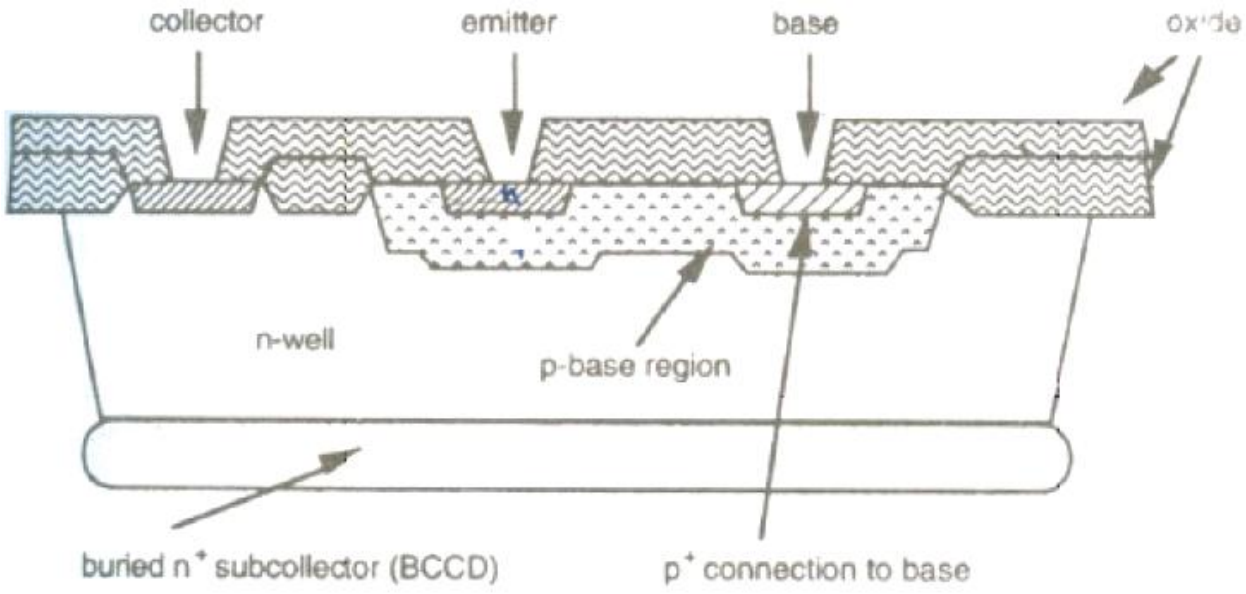


Figure 13 Cross section of BiCMOS process

The figure below shows the layout view of the BiCMOS process.

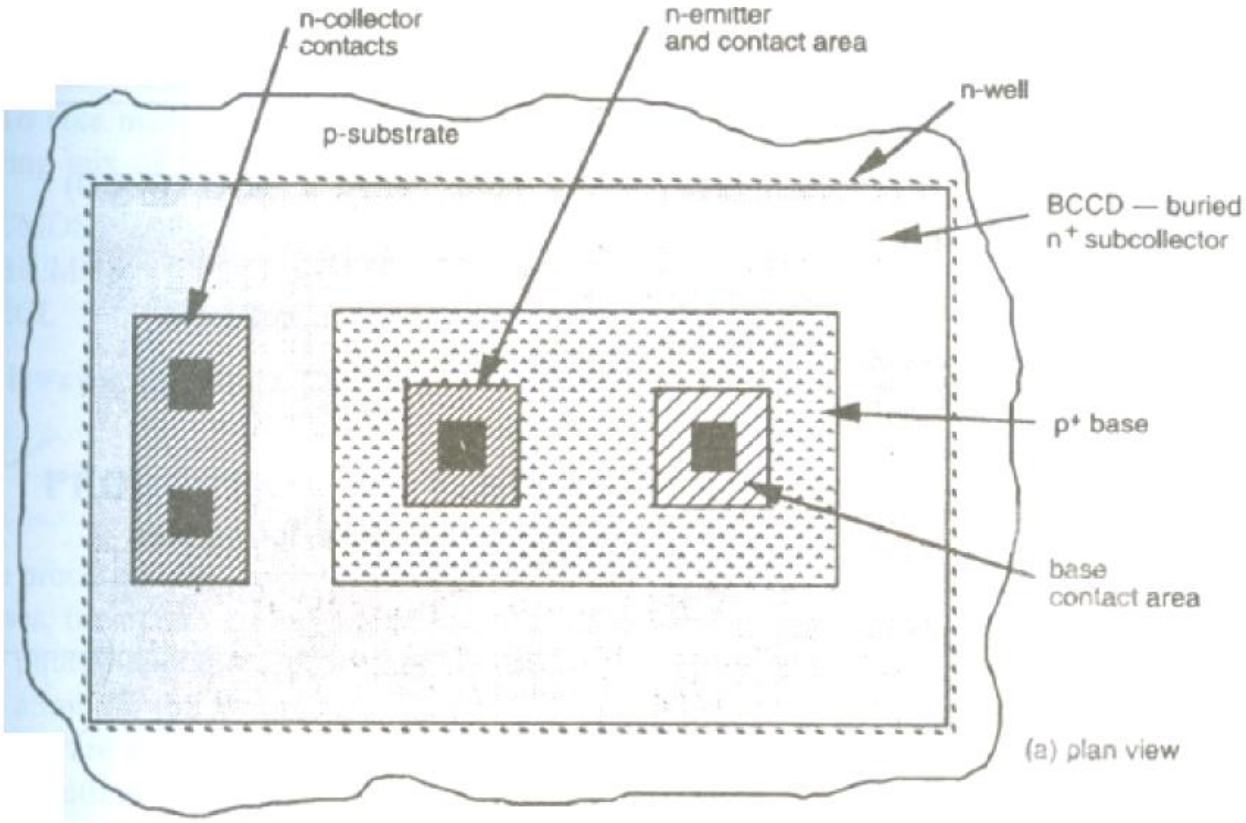


Fig.14. Layout view of BiCMOS process.

The graph below shows the relative cost vs. gate delay.

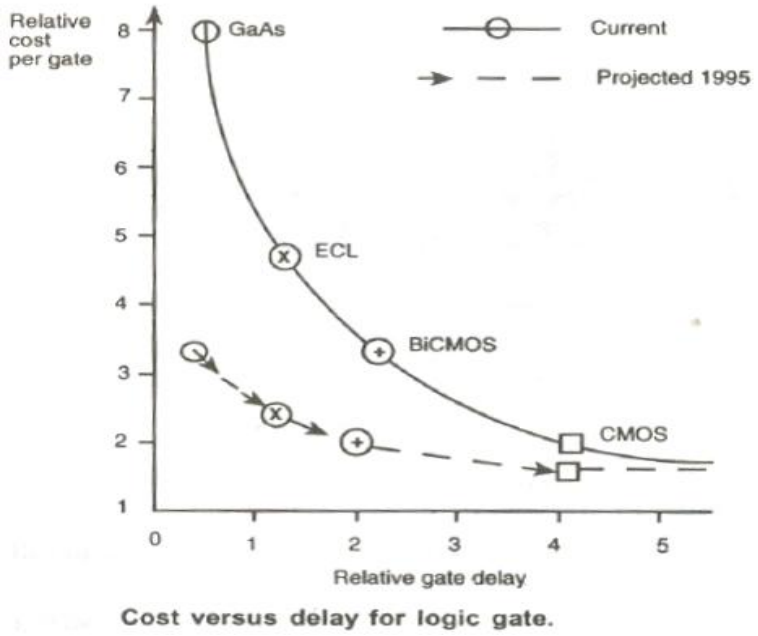


Fig.16. cost versus delay graph.

1.6 Production of e-beam masks:

In this topic we will understand how we are preparing the masks using e-beam technology. The following are the steps in production of e-beam masks.

- Starting materials is chromium coated glass plates which are coated with e-beam sensitive resist.
- E-beam machine is loaded with the mask description data.
- Plates are loaded into e-beam machine, where they are exposed with the patterns specified by mask description data.
- After exposure to e-beam, plates are introduced into developer to bring out patterns.
- The cycle is followed by a bake cycle which removes resist residue.
- The chrome is then etched and plate is stripped of the remaining e-beam resist.

We use two types of scanning, Raster scanning and vector scanning to map the pattern on to the mask. In raster type, e-beam scans all possible locations and a bit map is used to turn the e-beam on and off, depending on whether the particular location being scanned is to be exposed or not.

In vector type, beam is directed only to those locations which are to be exposed.

1.6.1 Advantages e-beam masks:

- Tighter layer to layer registration;
- Small feature sizes

MOS transistor theory

1.7 Introduction:

A MOS transistor is a majority-carrier device, in which the current in a conducting channel between the source and the drain is modulated by a voltage applied to the gate.

Symbols

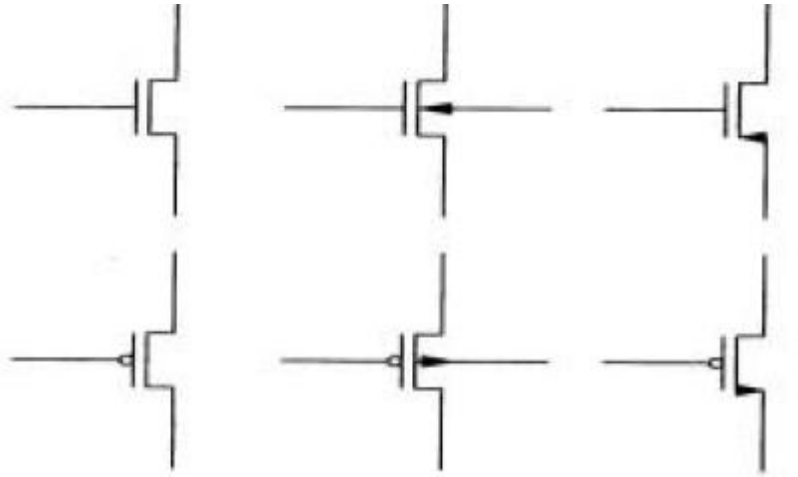


Figure 17: symbols of various types of transistors.

NMOS (n-type MOS transistor)

(1) Majority carrier = electrons

(2) A positive voltage applied on the gate with respect to the substrate enhances the number of electrons in the channel and hence increases the conductivity of the channel.

(3) If gate voltage is less than a threshold voltage Vt , the channel is cut-off (very low current between source & drain).

PMOS (p-type MOS transistor)

(1) Majority carrier = holes

(2) Applied voltage is negative with respect to substrate.

Relationship between Vgs and Ids, for a fixed Vds:

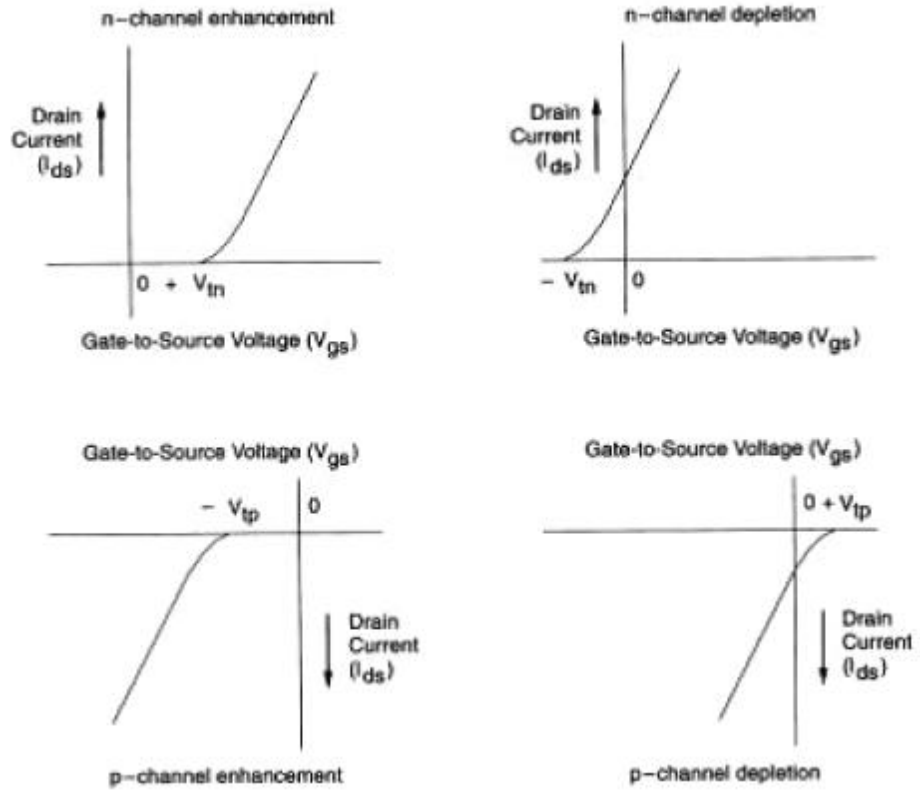


Figure 18: graph of Vgs vs Ids

Devices that are normally cut-off with zero gate bias are classified as **"enhancementmode"** devices.

Devices that conduct with zero gate bias are called **"depletion-mode"** devices.

Enhancement-mode devices are more popular in practical use.

Threshold voltage (Vt):

The voltage at which an MOS device begins to conduct ("turn on"). The **threshold voltage** is a function of

(1) Gate conductor material

(2) Gate insulator material

(3) Gate insulator thickness

(4) Impurity at the silicon-insulator interface

(5) Voltage between the source and the substrate Vsb

(6) Temperature

1.8 MOS equations (Basic DC equations):

Three MOS operating regions are: Cutoff or subthreshold region, linear region and saturation region.

The following equation describes all these three regions:

$$I_{ds} = \begin{cases} 0; & V_{gs} - V_t \leq 0 \quad cut\text{-}off \\ \beta\left[(V_{gs} - V_t)V_{ds} - \frac{V_{ds}^2}{2}\right]; & 0 < V_{ds} < V_{gs} - V_t \quad linear \\ \frac{\beta}{2}(V_{gs} - V_t)^2; & 0 < V_{gs} - V_t < V_{ds} \quad saturation \end{cases}$$

Where β is MOS transistor gain and it is given by β =μ ε /tox (W/L) again 'μ' is the mobility of the charge carrier

'ε' is the permittivity of the oxide layer.

'tox' is the thickness of the oxide layer.

'W' is the width of the transistor.(shown in diagram)

'L' is the channel length of the transistor.(shown in diagram)

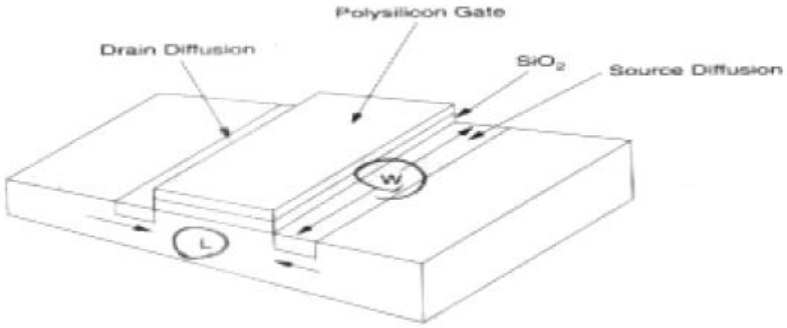


Diagram just to show the length and width of a MOSFET.

The graph of Id and Vds for a given Vgs is given below:

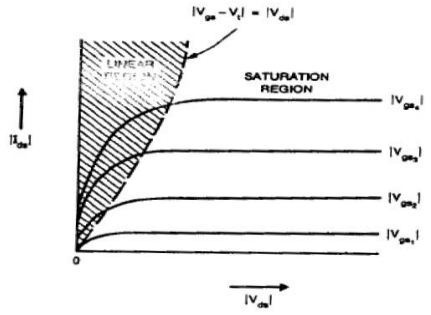


Figure 19: VI Characteristics of MOSFET

Second Order Effects:

Following are the list of second order effects of MOSFET.

- Threshold voltage – Body effect
- Subthreshold region
- Channel length modulation
- Mobility variation
- Fowler_Nordheim Tunneling
- Drain Punchthrough
- Impact Ionization – Hot Electrons

Threshold voltage – Body effect

The change in the threshold voltage of a MOSFET, because of the voltage difference between body and source is called body effect. The expression for the threshold voltage is given by the following expression.

$$Vt=V_{t(0)}+\gamma[(V_{sb}+2\Phi_F)^{1/2}-(2\Phi_F)^{1/2}]$$

where

Vt is the threshold voltage,

$V_{t(0)}$ is the threshold voltage without body effect

γ is the body coefficient factor

Φ_F is the fermi potential

V_{sb} is the potential difference between source and substarte.

If Vsb is zero, then Vt = Vt(0) that means the value of the threshold voltage will not be changed. Therefore, we short circuit the source and substrate so that, Vsb will be zero.

Subthreshold region:

For Vgs<Vt also we will get some value of Drain current this is called as Subthreshold current and the region is called as Subthreshold region.

Channel length modulation:

The channel length of the MOSFET is changed due to the change in the drain to source voltage. This effect is called as the channel length modulation. The effective channel length & the value of the drain current considering channel length modulation into effect is given by,

$$I_{ds} = \frac{\beta}{2}((V_{gs} - V_t)^2(1 + \lambda V_{ds}))$$

$$L_{eff} = L - \sqrt{2\varepsilon_o \frac{\varepsilon_{Si}}{qN}(V_{ds} - [V_{gs} - V_t])}.$$

Where λ is the channel length modulation factor.

Mobility:

Mobility is the defined as the ease with which the charge carriers drift in the substrate material. Mobility decreases with increase in doping concentration and increase in temperature. Mobility is the ratio of average carrier drift velocity and electric field. Mobility is represented by the symbol μ.

Fowler Nordhiem tunneling:

When the gate oxide is very thin there can be a current between gate and source or drain by electron tunneling through the gate oxide. This current is proportional to the area of the gate of the transistor.

Drain punchthough:

When the drain is a high voltage, the depletion region around the drain may extend to the source, causing the current to flow even it gate voltage is zero. This is known as Punchthrough condition.

Impact Ionization-hot electrons:

When the length of the transistor is reduced, the electric field at the drain increases. The field can become so high that electrons are imparted with enough energy we can term them as hot. These hot electrons impact the drain, dislodging holes that are then swept toward the negatively charged substrate and appear as a substrate current. This effect is known as Impact Ionization.

1.9 MOS Models

MOS model includes the Ideal Equations, Second-order Effects plus the additional Curve-fitting parameters. Many semiconductor vendors expend a lot of effects to model the devices they manufacture. (Standard: Level 3 SPICE) . Main SPICE DC parameters in level 1,2,3 in 1μn-well CMOS process.

1.10 CMOS INVETER CHARACTERISTICS

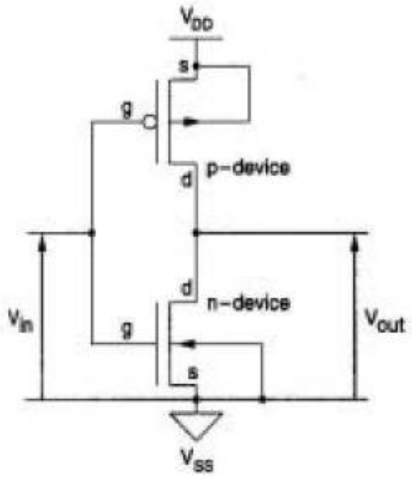


Figure 20: CMOS Inverter

CMOS inverters (Complementary NOSFET Inverters) are some of the most widely used and adaptable MOSFET inverters used in chip design. They operate with very little power loss and at relatively high speed. Furthermore, the CMOS inverter has good logic buffer characteristics, in that, its noise margins in both low and high states are large.

A CMOS inverter contains a PMOS and a NMOS transistor connected at the drain and gate terminals, a supply voltage VDD at the PMOS source terminal, and a ground connected at the NMOS source terminal, were VIN is connected to the gate terminals and VOUT is connected to the drain terminals.(given in diagram). It is important to notice that the CMOS does not contain any resistors, which makes it more power efficient that a regular resistor-MOSFET inverter. As the voltage at the input of the CMOS device varies between 0 and VDD, the state of the NMOS and PMOS varies accordingly. If we model each transistor as a simple switch activated by VIN, the inverter's operations can be seen very easily:

MOSFET	Condition on MOSFET	State of MOSFET
NMOS	Vgs<Vtn	OFF
NMOS	Vgs>Vtn	ON
PMOS	Vsg<Vtp	OFF
PMOS	Vsg>Vtp	ON

The table given, explains when the each transistor is turning on and off. When VIN is low, the NMOS is "off", while the PMOS stays "on": instantly charging VOUT to logic high. When Vin is high, the NMOS is "on and the PMOS is "off": taking the voltage at VOUT to logic low.

1.10.1 Inverter DC Characteristics:

Before we study the DC characteristics of the inverter we should examine the ideal characteristics of inverter which is shown below. The characteristic shows that when input is zero output will high and vice versa.

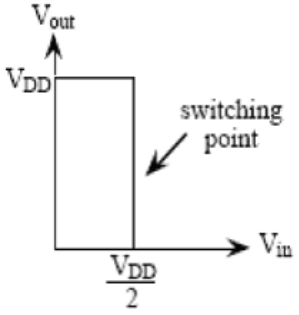


Figure 21: Ideal Characteristics of an Inverter.

The actual characteristic is also given here for the reference. Here we have shown the status of both NMOS and PMOS transistor in all the regions of the characteristics.

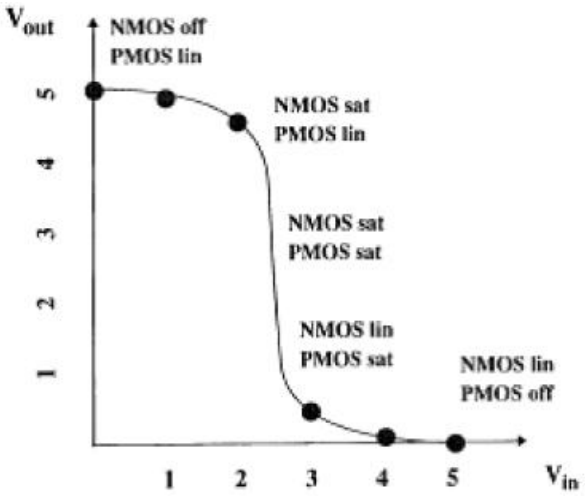


Figure 22: Actual Characteristics of an Inverter.

Graphical Derivation of Inverter DC Characteristics:

The actual characteristics are drawn by plotting the values of output voltage for different values of the input voltage. We can also draw the characteristics, starting with the VI characteristics of PMOS and NMOS characteristics.

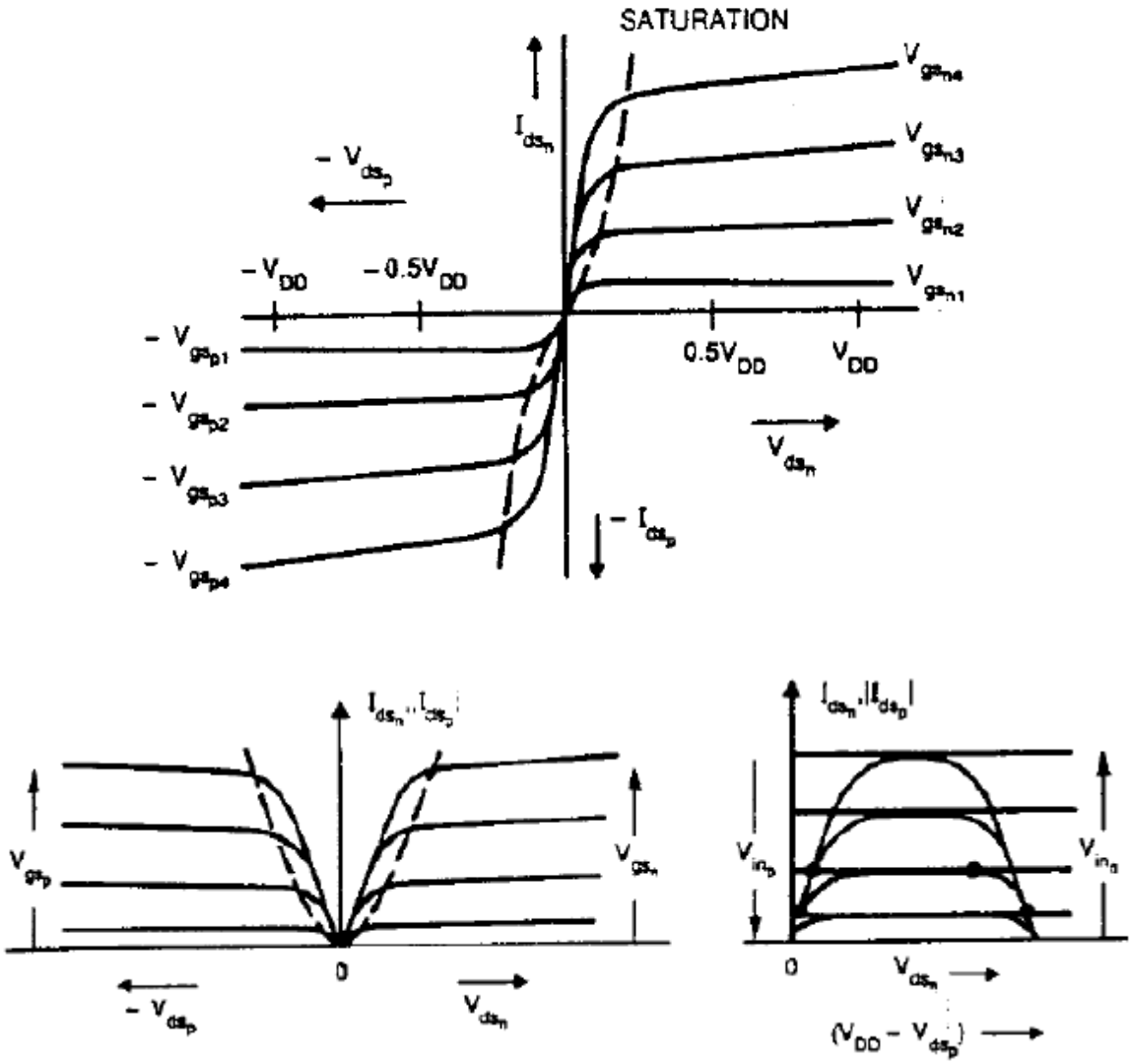


Figure 23-a,b,c: Graphical Derivation of DC Characteristics.

The characteristics given in figure 23a is the vi characteristics of the NMOS and PMOS characteristics (plot of Id vs. Vds). The figure 23b shows the values of drain current of PMOS transistor is taken to the positive side the current axis. This is done by taking the absolute value of the current. By superimposing both characteristics it leads to figure 23c. the actual characteristics may be now determined by the points of common Vgs intersection as shown in figure 23d.

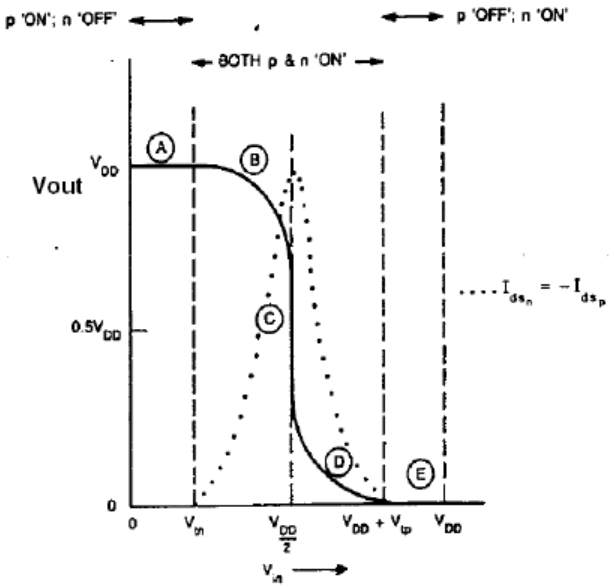


Figure 23d: CMOS Inverter Dc Characteristics.

Figure 23d shows five regions namely region A, B, C, D & E. also we have shown a dotted curve which is the current that is drawn by the inverter.

Region A:

The output in this region is high because the P device is OFF and n device is ON. In region A, NMOS is cutoff region and PMOS is on, therefore output is logic high. We can analyze the inverter when it is in region B. the analysis is given below:

Region B:

The equivalent circuit of the inverter when it is region B is given below.

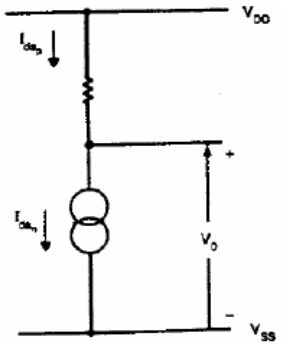


Figure 24: Equivalent circuit in Region B.

In this region PMOS will be in linear region and NMOS is in saturation region.

The expression for the NMOS current is

$$I_{ds_n} = \beta_n \frac{[V_{in} - V_{t_n}]^2}{2},$$

The expression for the PMOS current is

$$I_{ds_p} = -\beta_p \left[(V_{in} - V_{DD} - V_{t_p})(V_O - V_{DD}) - \frac{1}{2}(V_O - V_{DD})^2 \right]$$

The expression for the voltage Vo can be written as

$$V_O = (V_{in} - V_{t_p}) + \left[(V_{in} - V_{t_p})^2 - 2\left(V_{in} - \frac{V_{DD}}{2} - V_{t_p} \right) V_{DD} - \frac{\beta_n}{\beta_p}(V_{in} - V_{t_n})^2 \right]^{1/2}$$

Region C:

The equivalent circuit of CMOS inverter when it is in region C is given here. Both n and p transistors are in saturation region, we can equate both the currents and we can obtain the expression for the midpoint voltage or switching point voltage of a inverter. The corresponding equations are as follows:

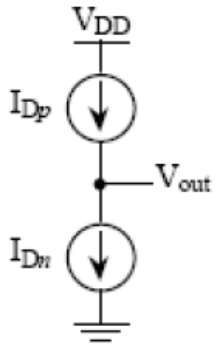


Figure 25: Equivalent circuit in Region C.

The corresponding equations are as follows:

$$I_{ds_p} = \frac{1}{2}\beta_p (V_{in} - V_{DD} - V_{t_p})^2$$

$$I_{ds_n} = \frac{1}{2}\beta_n (V_{in} - V_{t_n})^2$$

By equating both the currents, we can obtain the expression for the switching point voltage as,

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

Region D: The equivalent circuit for region D is given in the figure below.

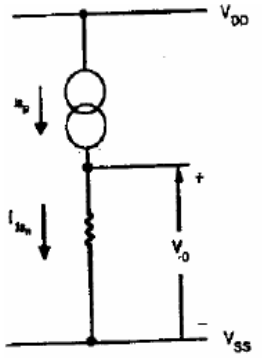


Figure 26: equivalent circuit in region D.

We can apply the same analysis what we did for region B and C and we can obtain the expression for output voltage.

Region E:

The output in this region is zero because the P device is OFF and n device is ON.

Influence of βn / βp on the VTC characteristics:

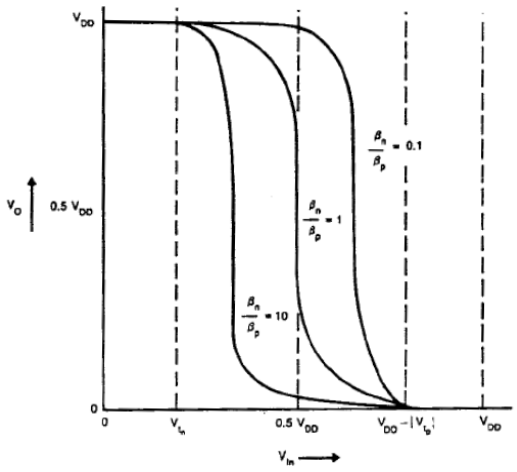


Figure 27: Effect of βn/βp ratio change on the DC characteristics of CMOS inverter.

The characteristics shifts left if the ratio of βn/βp is greater than 1(say 10). The curve shifts right if the ratio of βn/βp is lesser than 1(say 0.1). This is decided by the switching point equation of region C. the equation is repeated here the reference again.

$$Vm=Vsp=V_{DD}+Vtp+Vtn(\beta n/\beta p)1/2 \,/\, 1+(\beta n/\beta p)1/2$$

Noise Margin:

Noise margin is a parameter related to input output characteristics. It determines the allowable noise voltage on the input so that the output is not affected. We will specify it in terms of two things:

- LOW noise margin
- HIGH noise margin

LOW noise margin: is defined as the difference in magnitude between the maximum Low output voltage of the driving gate and the maximum input Low voltage recognized by the driven gate.

$$\mathbf{NML=|VILmax – VOLmax|}$$

HIGH noise margin: is defined difference in magnitude between minimum High output voltage of the driving gate and minimum input High voltage recognized by the receiving gate.

$$\mathbf{NMH=|Vohmin – VIHmin|}$$

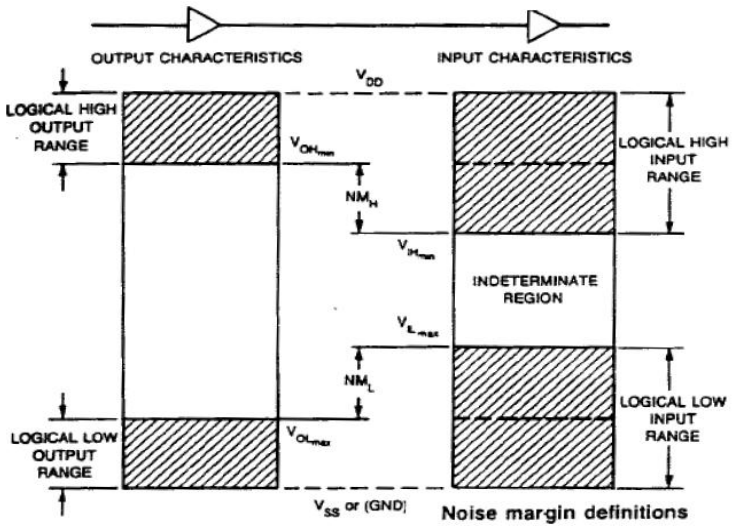


Figure 28: noise margin definitions.

Figure shows how exactly we can find the noise margin for the input and output. We can also find the noise margin of a CMOS inverter. The following figure gives the idea of calculating the noise margin.

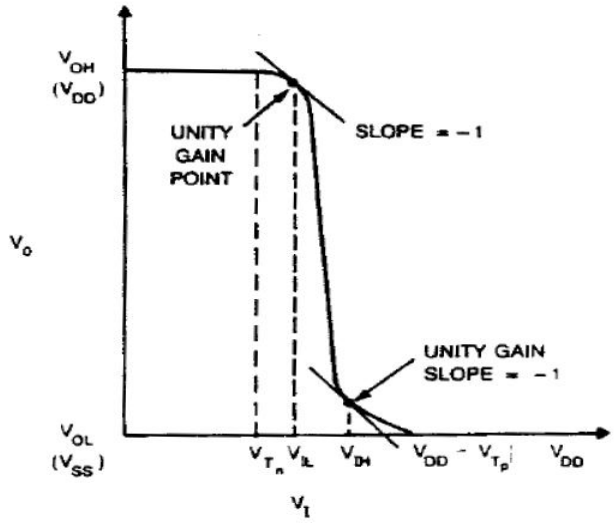


Figure 29: CMOS inverter noise margins.

1.11 Static Load MOS inverters:

In the figure given below we have shown a resistive load and current source load inverter. Usually resistive load inverters are not preferred because of the power consumption and area issues.

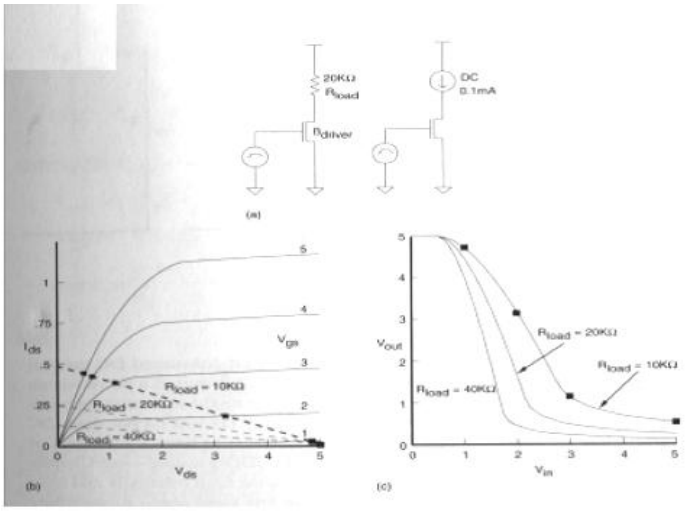


Figure 30: static load inverter.

1.12 Pseudo-NMOS inverter:

This circuit uses the load device which is p device and is made to turn on always by connecting the gate terminal to the ground.

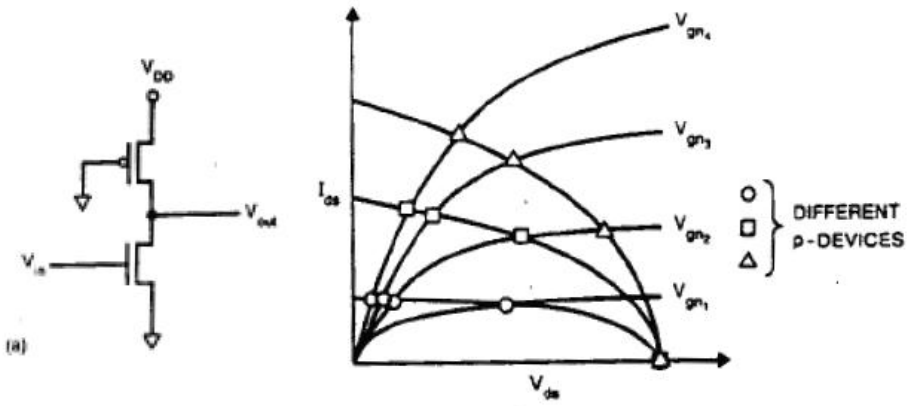


Figure 31: Pseudo-NMOS inverter.

Power consumption is High compared to CMOS inverter particularly when NMOS device is ON because the p load device is always ON.

1.13 Saturated load inverter:

The load device is an nMOS transistor in the saturated load inverter. This type of inverter was used in nMOS technologies prior to the availability of nMOS depletion loads.

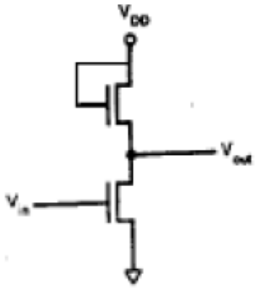


Figure 32: Saturated load inverter

1.14 Transmission gates:

It's a parallel combination of pmos and nmos transistor with the gates connected to a complementary input. After looking into various issues of pass transistors we will come back to the TGs again.

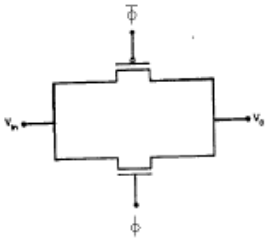


Figure 33: Transmission gate

1.15 Pass transistors:

We have n and p pass transistors.

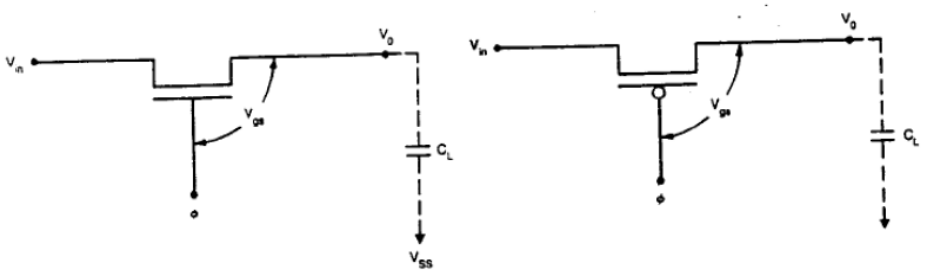


Figure 34: n and p pass transistors.

The disadvantage with the pass transistors is that, they will not be able to transfer the logic levels properly. The following table gives that explanation in detail.

Transmission characteristics of *n*-channel and *p*-channel pass transistors

DEVICE	TRANSMISSION OF '1'	TRANSMISSION OF '0'
n	poor	good
p	good	poor

If Vdd (5 volts) is to be transferred using nMOS the output will be (Vdd-Vtn). **POOR 1 or Weak Logic 1**

If Gnd(0 volts) is to be transferred using nMOS the output will be Gnd. **GOOD 0 or Strong Logic 0**

If Vdd (5 volts) is to be transferred using pMOS the output will be Vdd. **GOOD 1 or Strong Logic 1**

If Gnd(0 volts) is to be transferred using pMOS the output will be Vtp. **POOR 0 or Weak Logic 0.**

1.16 Transmission gates (TGs):

It's a parallel combination of pmos and nmos transistor with the gates connected to a complementary input. The disadvantages weak 0 and weak 1 can be overcome by using a TG instead of pass transistors.

Working of transmission gate can be explained better with the following equation. **When _='0' n and p device off, Vin=0 or 1, Vo='Z' When _='1' n and p device on, Vin=0 or 1, Vo=0 or 1 , where 'Z' is high impedance.** One more important advantage of TGs is that the reduction in

the resistance because two transistors will come in parallel and it is shown in the graph. The graph shows the resistance of n and p pass transistors, and resistance of TG which is lesser than the other two.

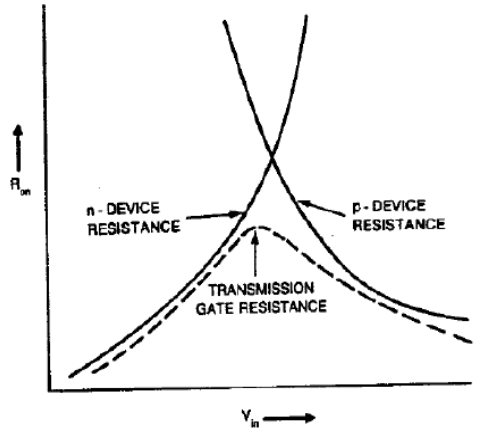


Figure 35: Graph of resistance vs. input for pass transistors and TG.

1.17 Tristate Inverter:

By cascading a transmission gate with an inverter the tristate inverter circuit can be obtained. The working can be explained with the help of the circuit.

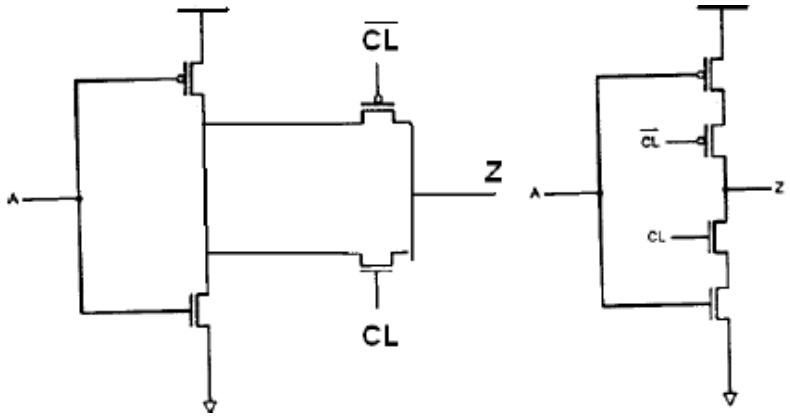


Figure 36: Tristate Inverter

The two circuits are the same only difference is the way they are written. When CL is zero the output of the inverter is in tristate condition. When CL is high the output is Z is the inversion of the input A

Recommended questions:

1. Write a note on integration era.
2. What do you mean MOS.
3. Bring out the difference between enhancement mode and depletion mode MOS transistors.
4. Explain the types of MOS transistors.
5. What do you mean by fabrication.
6. Explain nMOS fabrication process.
7. Explain CMOS fabrication process.
8. Explain BiCMOS technology.
9. What is the different between CMOS and BiCMOS technology.
10. Write a short note on production of E-beam.
11. Write MOS device design equation for all the region of operations.
12. List the region of operations of MOS transistors.
13. Explain CMOS inverter with all the region of operations.
14. Write a note on static load MOS inverter and differential inverter with neat diagram.
15. Explain transmission gate.
16. Write a note on tristate inverter.

Unit-2

Circuit Design Processes

MOS layers, stick diagrams, Design rules and layout- lambda-based design and other rules. Examples, layout diagrams, symbolic diagram, tutorial exercises.

Basic physical design of simple logic gates.

Recommended readings:

1. Douglas A. Pucknell & Kamran Eshraghian, **"Basic VLSI Design"** PHI 3rd Edition (original Edition – 1994), 2005.

2. Neil H. E. Weste and K. Eshragian**," Principles of CMOS VLSI Design: A Systems Perspective,"** 2nd edition, Pearson Education (Asia) Pvt. Ltd., 2000. History of VLSI

Unit 2

Circuit Design Process

2.1 Introduction:

In this chapter we are going to study how to get the schematic into stick diagrams or layouts.

MOS circuits are formed on four basic layers:

- N-diffusion
- P-diffusion
- Polysilicon
- Metal

These layers are isolated by one another by thick or thin silicon dioxide insulating layers.

Thin oxide mask region includes n-diffusion / p-diffusion and transistor channel.

2.2 Stick diagrams:

Stick diagrams may be used to convey layer information through the use of a color code. For example: n-diffusion--green poly--red blue-- metal yellow--implant black--contact areas.

Encodings for NMOS process:

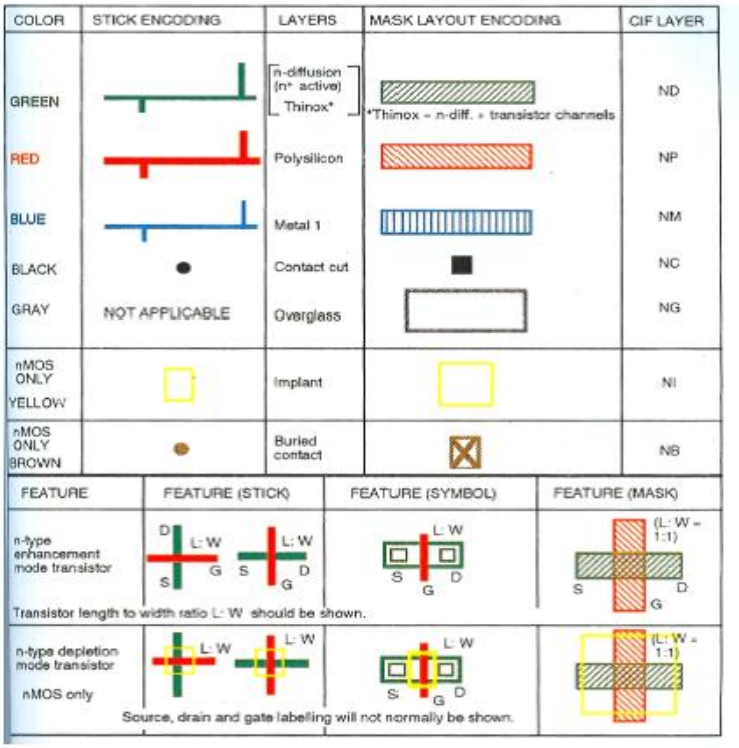


Figure 1: NMOS encodings.

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

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

2.2.1 Encodings for CMOS process:

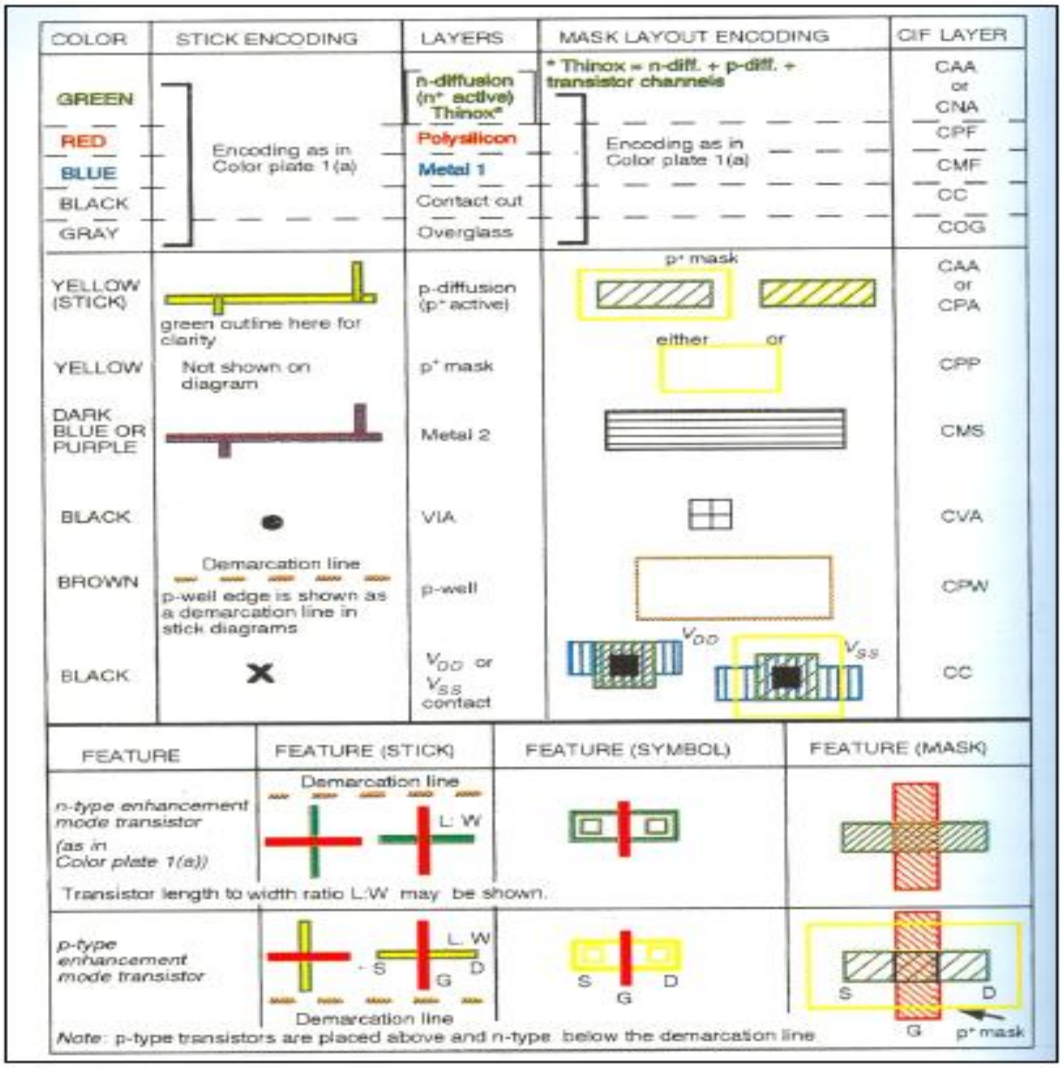


Figure 2: CMOS encodings.

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

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

2.2.2 Encoding for BJT and MOSFETs:

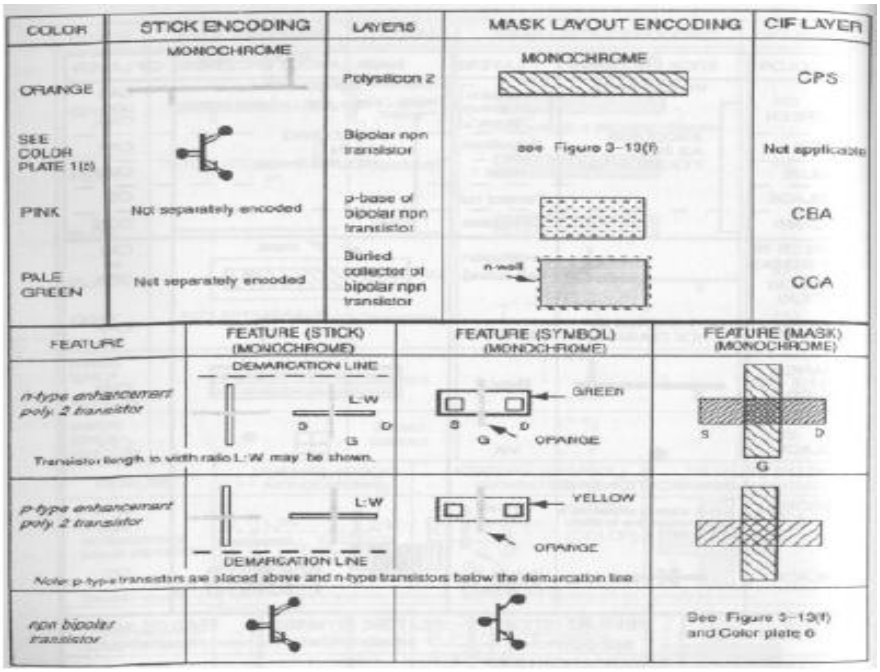


Figure 3: Bi CMOS encodings.

There are several layers in an nMOS chip:

_ a p-type substrate

_ paths of n-type diffusion

_ a thin layer of silicon dioxide

_ paths of polycrystalline silicon

_ a thick layer of silicon dioxide

_ paths of metal (usually aluminum)

_ a further thick layer of silicon dioxide

With contact cuts through the silicon dioxide where connections are required. The three layers carrying paths can be considered as independent conductors that only interact where polysilicon crosses diffusion to form a transistor. These tracks can be drawn as stick diagrams with _ diffusion in green _ polysilicon in red _ metal in blue using black to indicate contacts between layers and yellow to mark regions of implant in the channels of depletion mode transistors.

With CMOS there are two types of diffusion: n-type is drawn in green and p-type in brown. These are on the same layers in the chip and must not meet. In fact, the method of

fabrication required that they be kept relatively far apart. Modern CMOS processes usually support more than one layer of metal. Two are common and three or more are often available.

Actually, these conventions for colors are not universal; in particular, industrial (rather than academic) systems tend to use red for diffusion and green for polysilicon. Moreover, a shortage of colored pens normally means that both types of diffusion in CMOS are colored green and the polarity indicated by drawing a circle round p-type transistors or simply inferred from the context. Colorings for multiple layers of metal are even less standard.

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

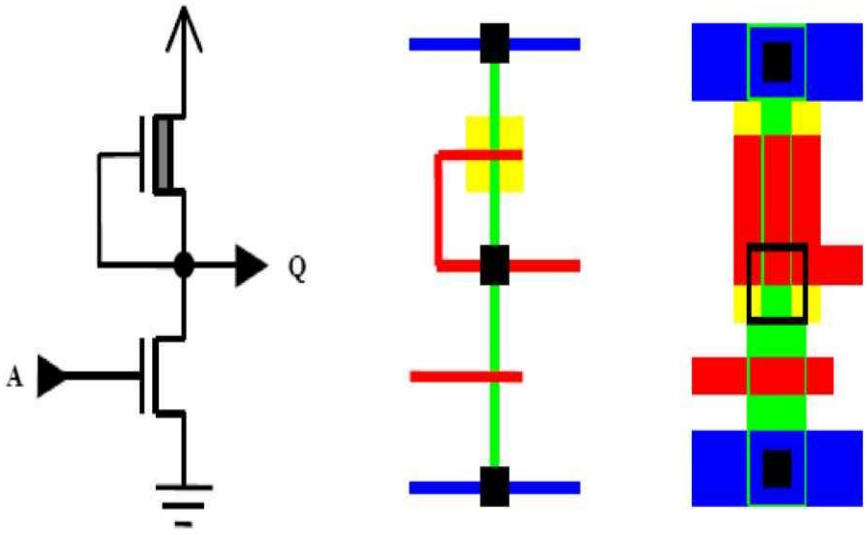


Figure 4: nMOS depletion load inverter.

Figure4 shows schematic, stick diagram and corresponding layout of nMOS depletion load inverter

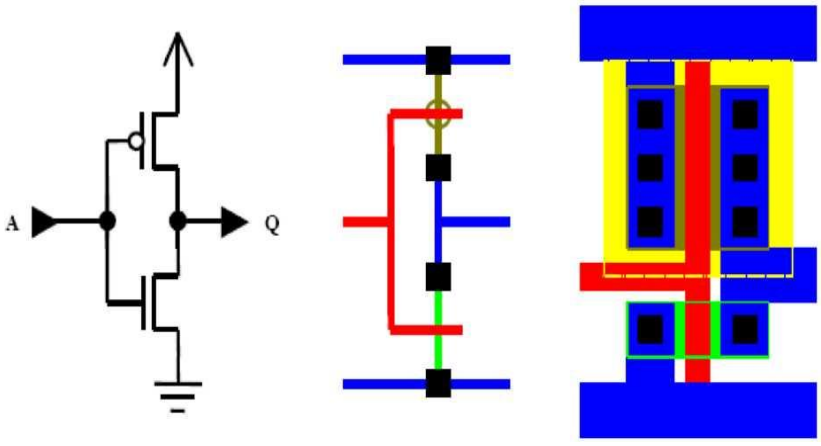


Figure 5: CMOS inverter

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

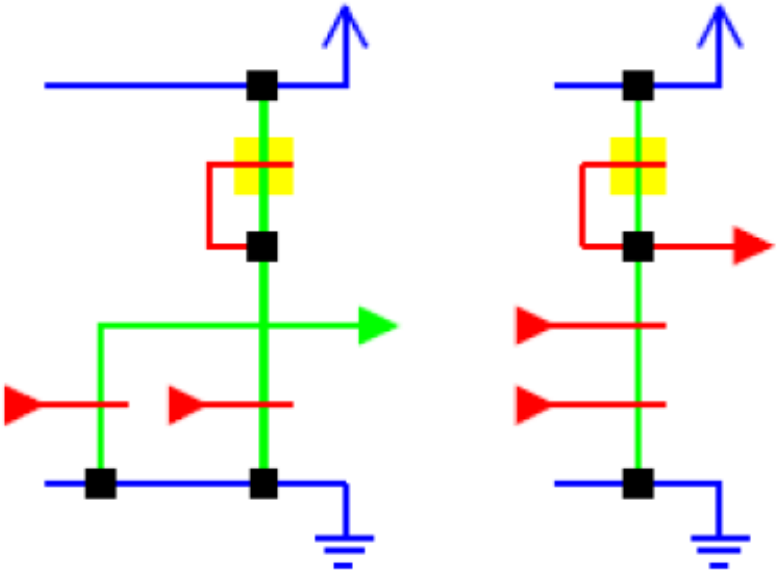


Figure 6 shows the stick diagrams for nMOS NOR and NAND.

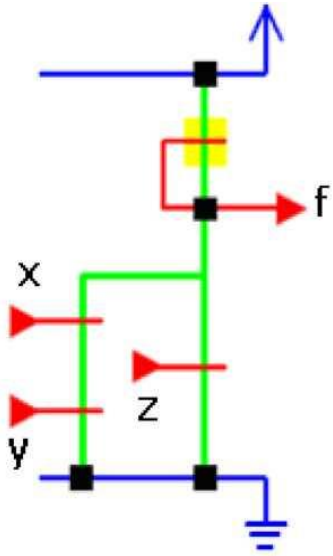


Figure 7: stick diagram of a given function f.

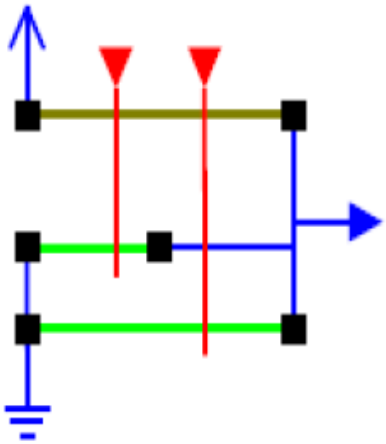


Figure 7

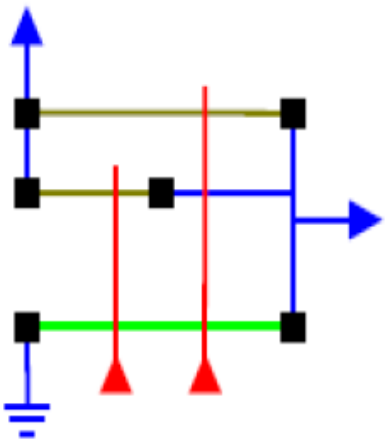


Figure 8

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

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

2.2.3NMOS and CMOS Design style:

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

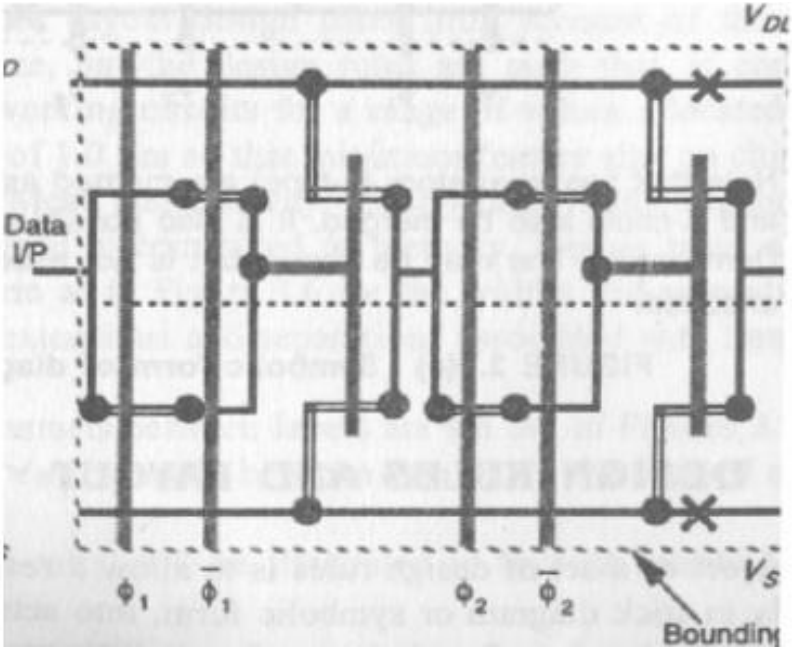


Figure 9: stick diagram of dynamic shift register in CMOS style.

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

2.3 Design Rules:

Design rules include width rules and spacing rules. Mead and Conway developed a set of simplified scalable X -based design rules, which are valid for a range of fabrication technologies. In these rules, the minimum feature size of a technology is characterized as 2 X. All width and spacing rules are specified in terms of the parameter X. Suppose we have design rules that call for a minimum width of 2 X, and a minimum spacing of 3 X . If we select a 2 um technology (i.e., X = 1 um), the above rules are translated to a minimum width of 2 um and a minimum spacing of 3 um. On the other hand, if a 1 um technology (i.e., X = 0.5 um) is selected, then the same width and spacing rules are now specified as 1 um and 1.5 um, respectively.

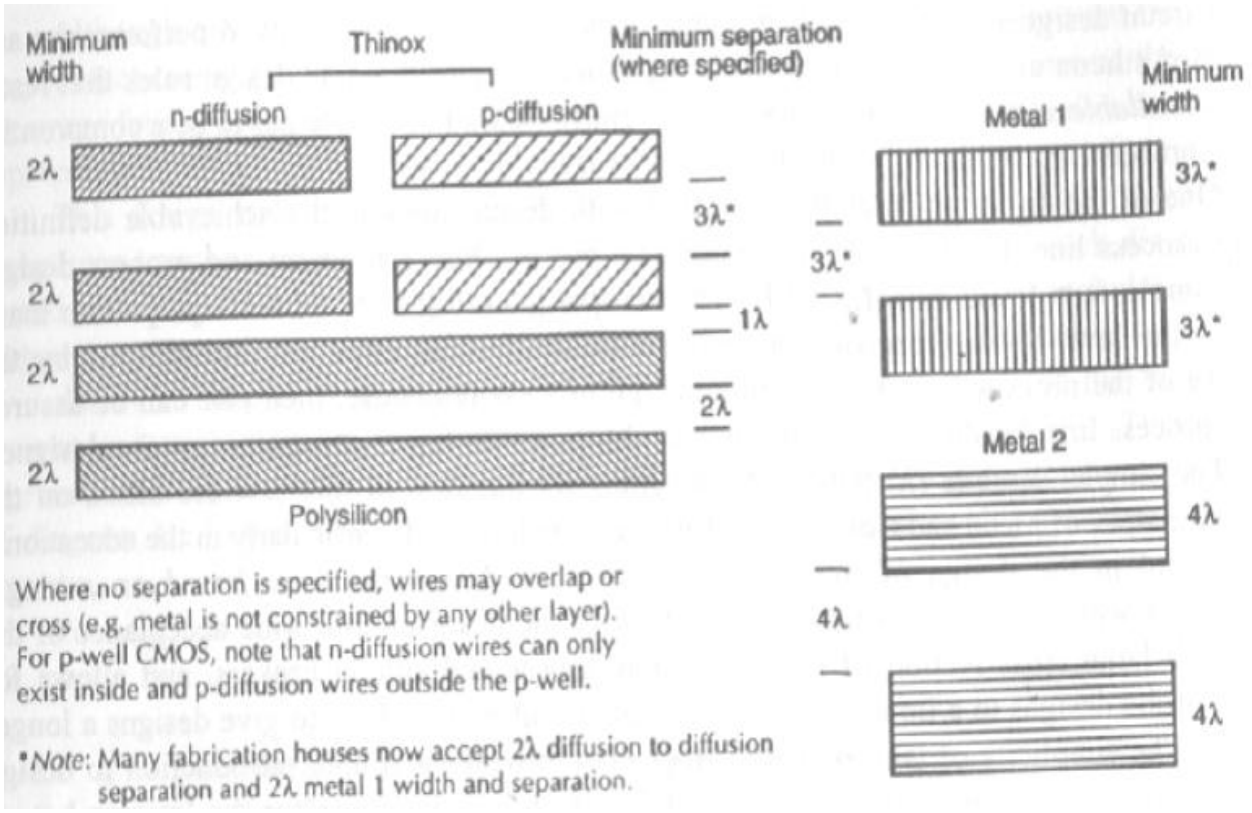


Figure 10: Design rules for the diffusion layers and metal layers.

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

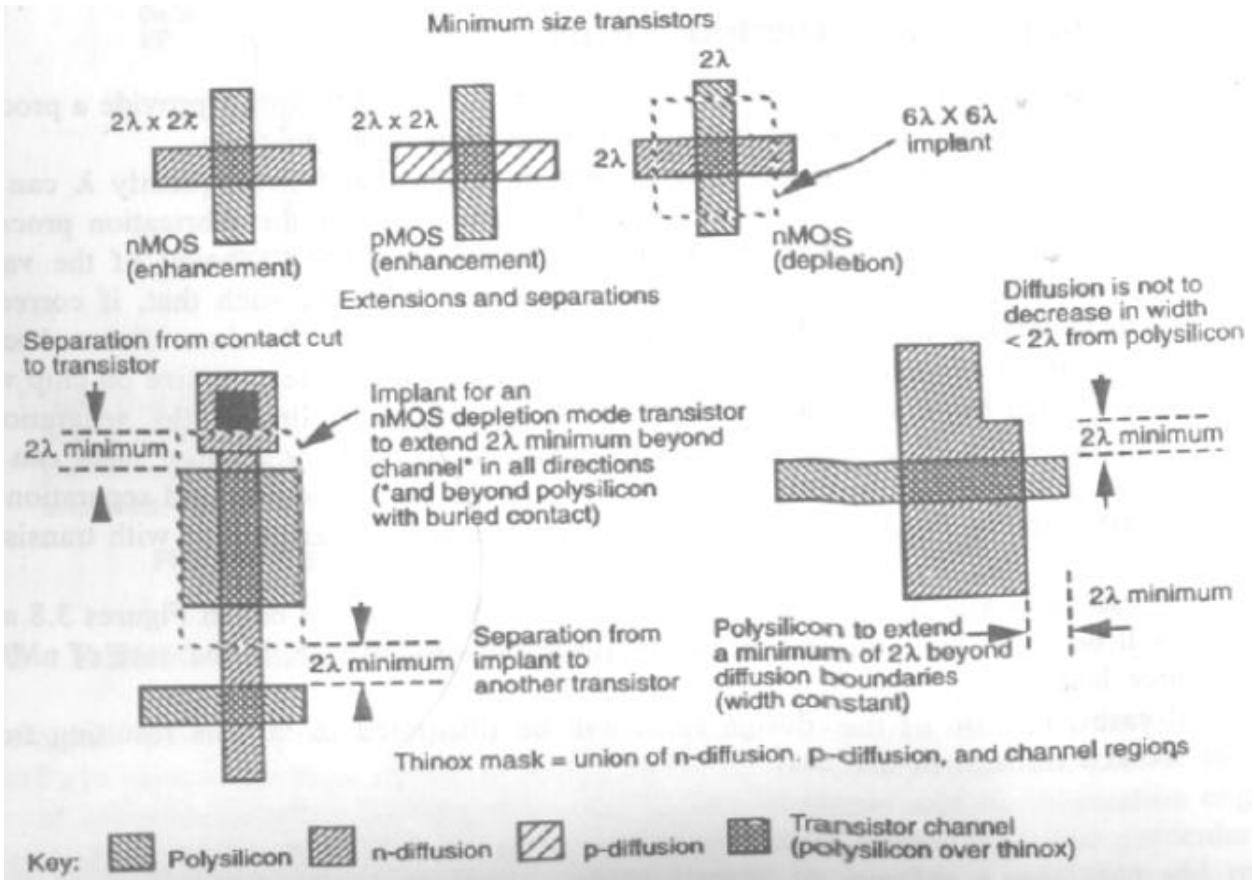


Figure 11: Design rules for transistors and gate over hang distance.

Figure shows the design rule for the transistor, and it also shows that the poly should extend for a minimum of 7k beyond the diffusion boundaries. (gate over hang distance)

What is Via?

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

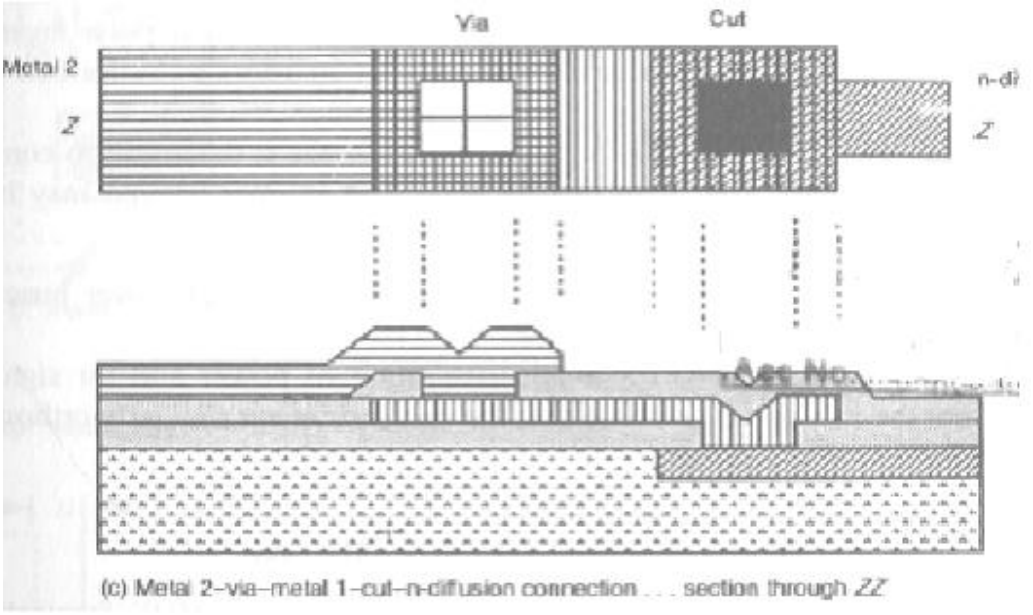


Figure 12: cross section showing the contact cut and via

Figure shows the design rules for contact cuts and Vias. The design rule for contact is minimum2λx2λ and same is applicable for a Via.

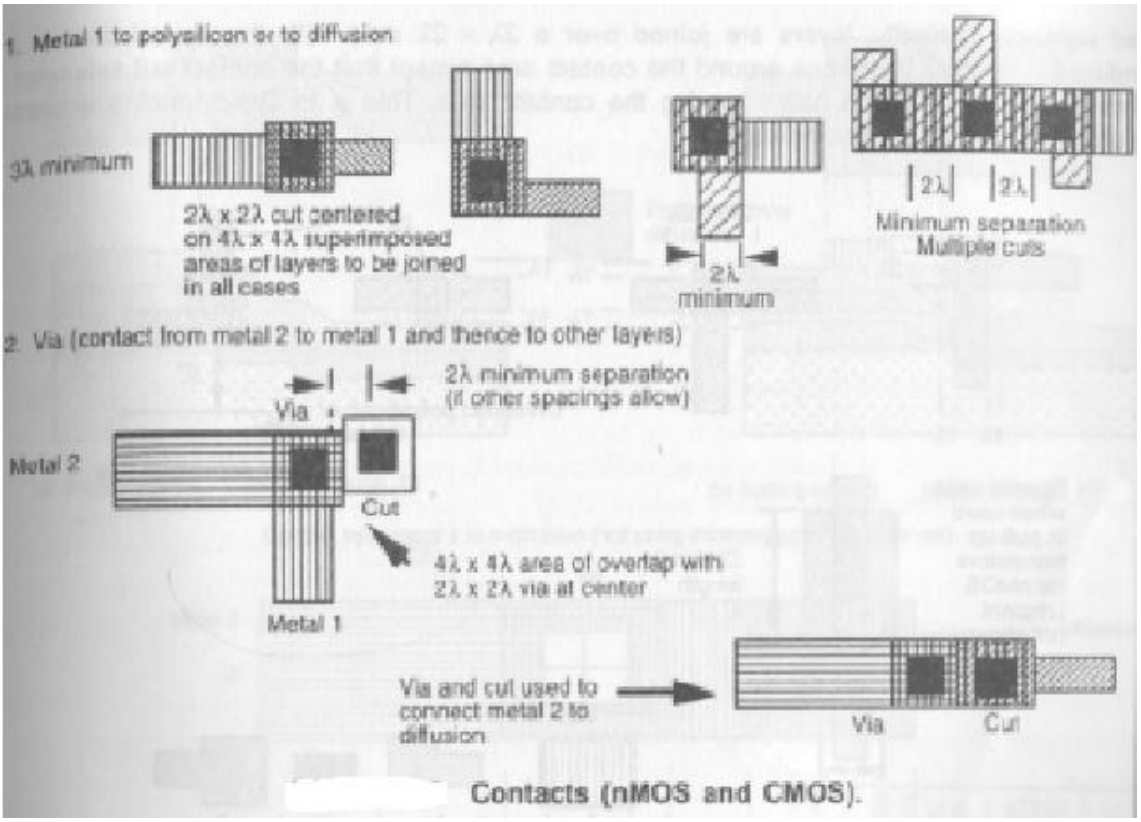


Figure 13: Design rules for contact cuts and vias

2.3.1 Buried contact: The contact cut is made down each layer to be joined and it is shown in figure 14.

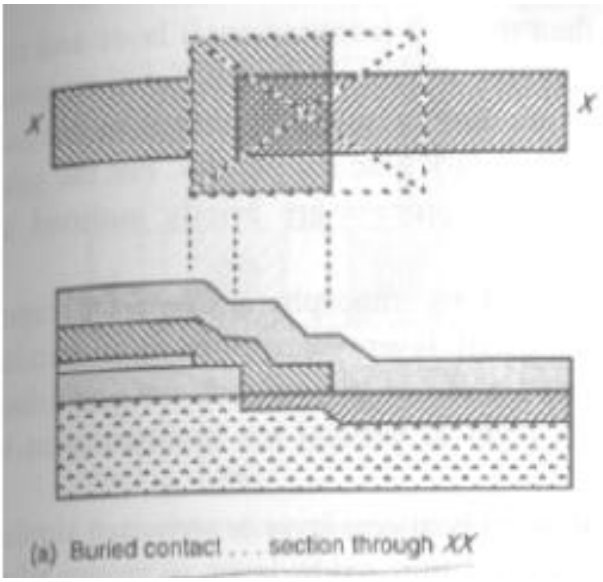


Figure 14: Buried contact.

2.3.2 Butting contact: The layers are butted together in such a way the two contact cuts become contiguous. We can better under the butting contact from figure 15.

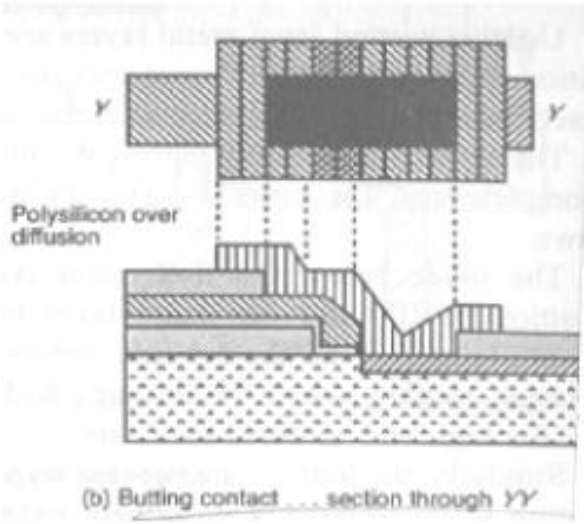


Figure 15: Butting contact.

2.4 CMOS LAMBDA BASED DESIGN RULES:

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

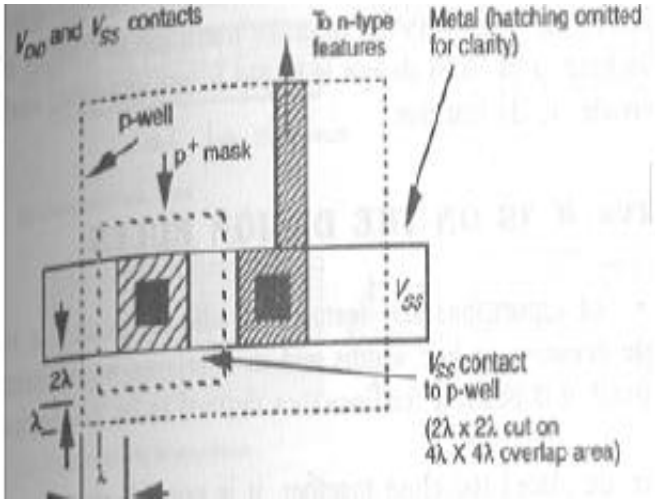


Figure 16: CMOS design rules.

2.4.1 Orbit 2μm CMOS process:

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

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

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

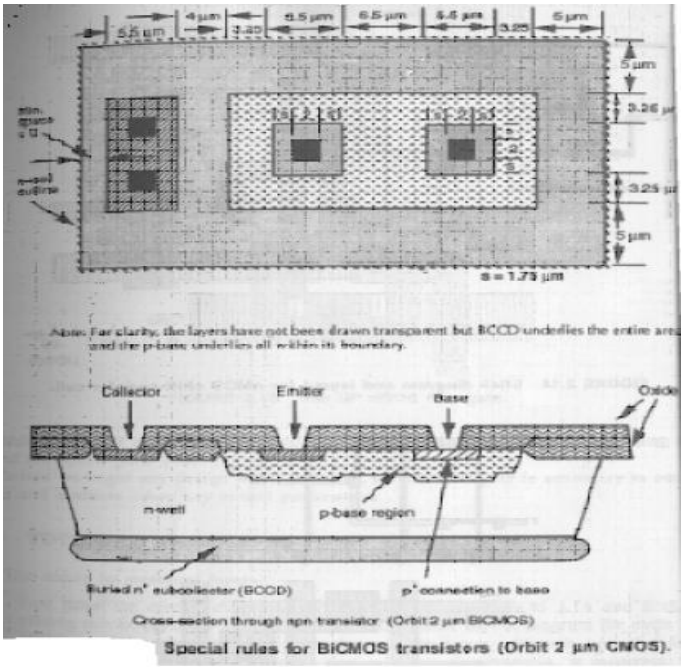


Figure 17: BiCMOS design rules.

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

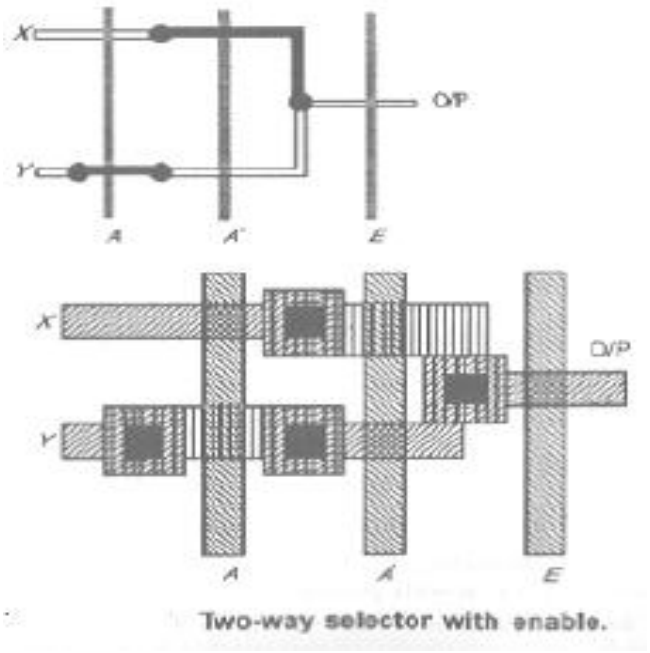


Figure 18: Two way selector stick and layout

2.5 BASIC PHYSICAL DESIGN AN OVERVIEW

The VLSI design flow for any IC design is as follows

1 .Specification (problem definition)

2. Schematic (gate level design) (equivalence check)

3. Layout (equivalence check)

4. Floor Planning

5 .Routing, Placement

6. On to Silicon

When the devices are represented using these layers, we call it physical design. The design is carried out using the design tool, which requires to follow certain rules. Physical structure is required to study the impact of moving from circuit to layout. When we draw the layout from the schematic, we are taking the first step towards the physical design. Physical design is an important step towards fabrication. Layout is representation of a schematic into layered diagram. This diagram reveals the different layers like ndiff, polysilicon etc that go into

formation of the device. At every stage of the physical design simulations are carried out to verify whether the design is as per requirement. Soon after the layout design the DRC check is used to verify minimum dimensions and spacing of the layers. Once the layout is done, a layout versus schematic check carried out before proceeding further. There are different tools available for drawing the layout and simulating it.

The simplest way to begin a layout representation is to draw the stick diagram. But as the complexity increases it is not possible to draw the stick diagrams. For beginners it easy to draw the stick diagram and then proceed with the layout for the basic digital gates. We will have a look at some of the things we should know before starting the layout. In the schematic representation lines drawn between device terminals represent interconnections and any no planar situation can be handled by crossing over. But in layout designs a little more concern about the physical interconnection of different layers. By simply drawing one layer above the other it not possible to make interconnections, because of the different characters of each layer. Contacts have to be made whenever such interconnection is required. The power and the ground connections are made using the metal and the common gate connection using the polysilicon. The metal and the diffusion layers are connected using contacts. The substrate contacts are made for same source and substrate voltage. Which are not implied in the schematic. These layouts are governed by DRC's and have to be atleast of the minimum size depending on the technology used. The crossing over of layers is another aspect which is of concern and is addressed next.

1. Poly crossing diffusion makes a transistor

2. Metal of the same kind crossing causes a short.

3. Poly crossing a metal causes no interaction unless a contact is made.

Different design tricks need to be used to avoid unknown creations. Like a combination of metal1 and metal 2 can be used to avoid short. Usually metal 2 is used for the global vdd and vss lines and metal1 for local connections.

2.6 SCHEMATIC AND LAYOUT OF BASIC GATES

1. CMOS INVERTER/NOT GATE SCHEMATIC

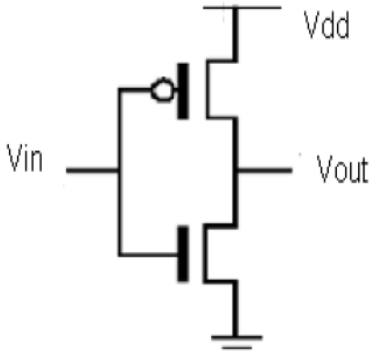


Figure 19: Inverter.

TOWARDS THE LAYOUT

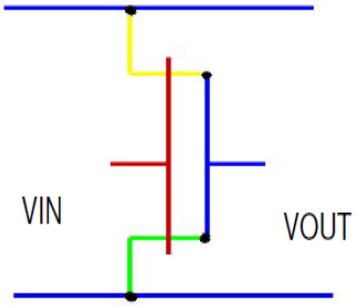


Figure 20: Stick diagram of inverter.

The diagram shown here is the stick diagram for the CMOS inverter. It consists of a Pmos and a Nmos connected to get the inverted output. When the input is low, Pmos (yellow) is on and pulls the output to vdd; hence it is called pull up device. When Vin =1, Nmos (green) is on it pulls Vout to Vss, hence Nmos is a pull down device. The red lines are the poly silicon lines connecting the gates and the blue lines are the metal lines for VDD (up) and VSS (down).The layout of the cmos inverter is shown below. Layout also gives the minimum dimensions of different layers, along with the logical connections and main thing about layouts is that can be simulated and checked for errors which cannot be done with only stick diagrams.

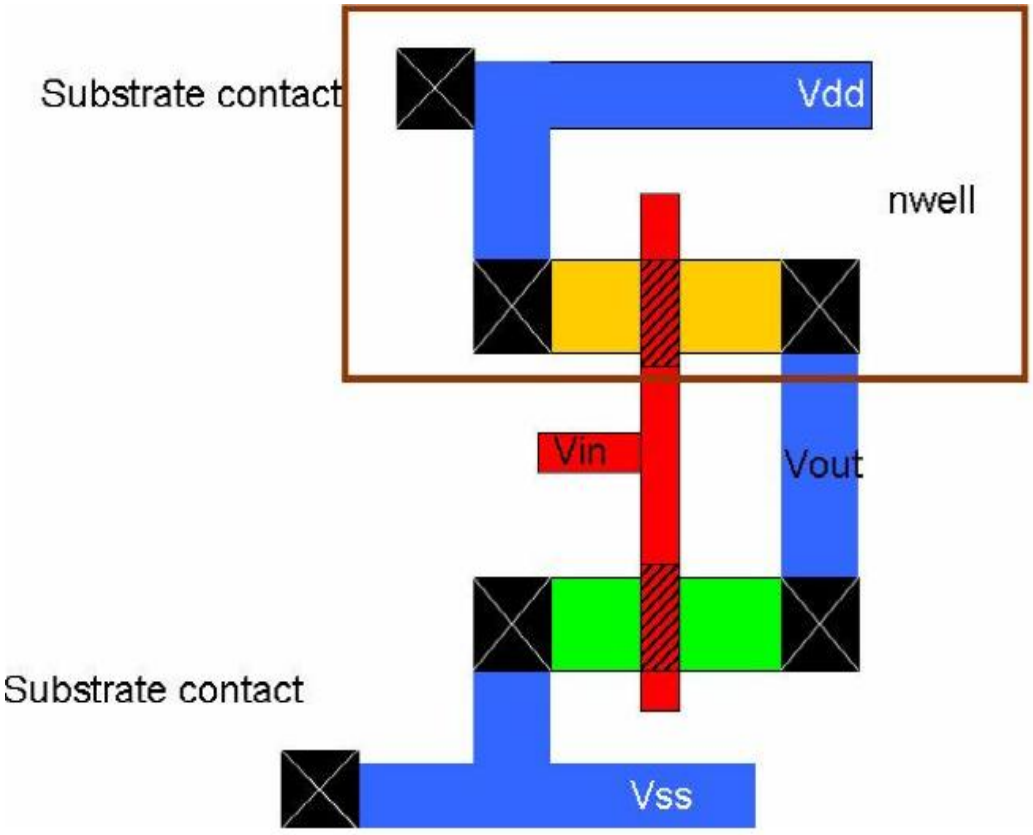


Figure 21: Layout of inverter.

The layout shown above is that of a CMOS inverter. It consists of a pdiff (yellow colour) forming the pmos at the junction of the diffusion and the polysilicon (red colour) shown hatched ndiff (green) forming the nmos(area hatched).The different layers drawn are checked for their dimensions using the DRC rule check of the tool used for drawing. Only after the DRC (design rule check) is passed the design can proceed further. Further the design undergoes Layout Vs Schematic checks and finally the parasitic can be extracted.

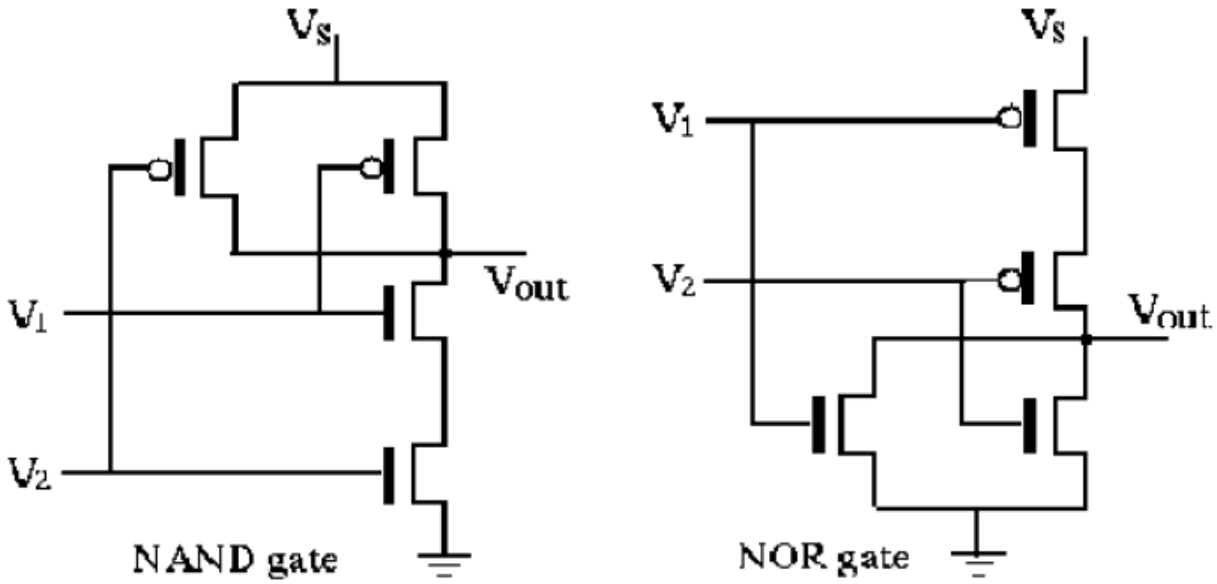


Figure 22: Schematic diagrams of nand and nor gate

We can see that the nand gate consists of two pmos in parallel which forms the pull up logic and two nmos in series forming the pull down logic. It is the complementary for the nor gate. We get inverted logic from CMOS structures. The series and parallel connections are for getting the right logic output. The pull up and the pull down devices must be placed to get high and low outputs when required.

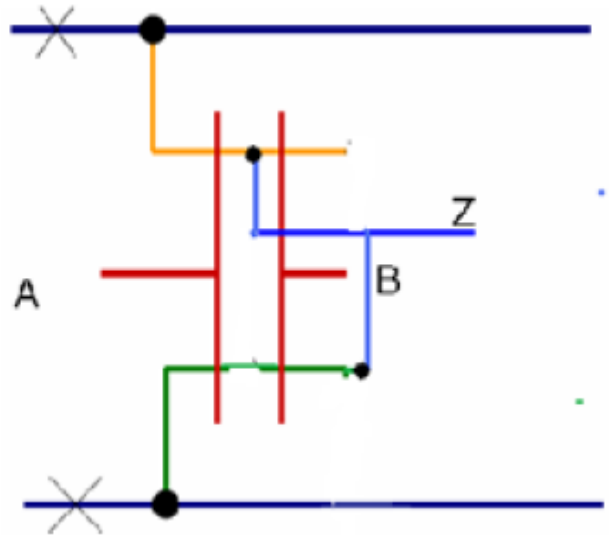


Figure 23: Stick diagrams of nand gate.

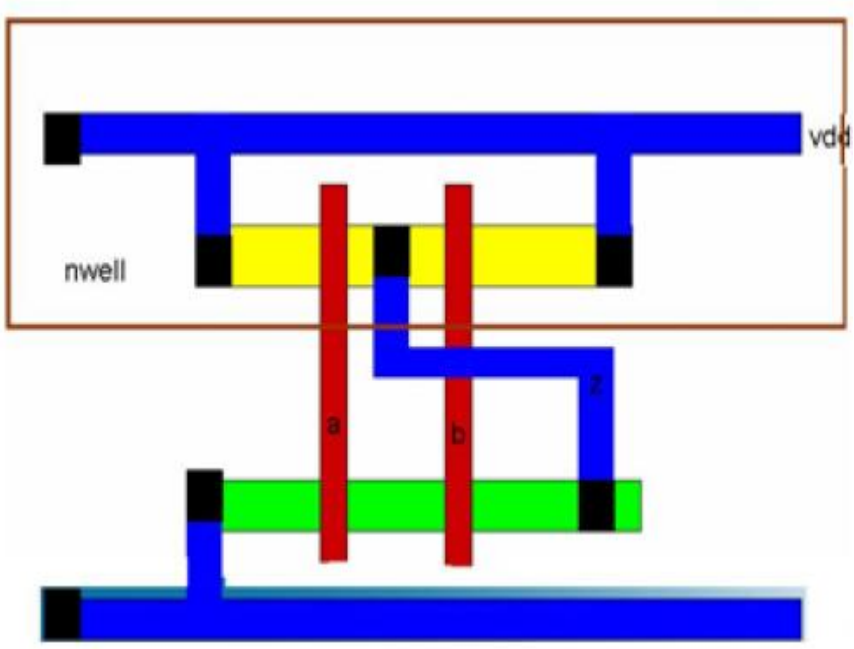


Figure 24: Layout of nand gate.

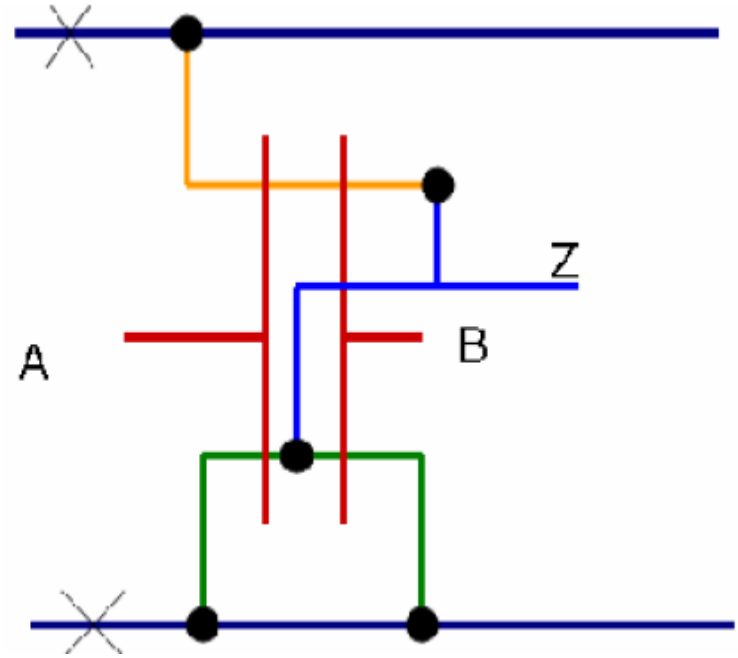


Figure 25: Stick diagram of nor gate.

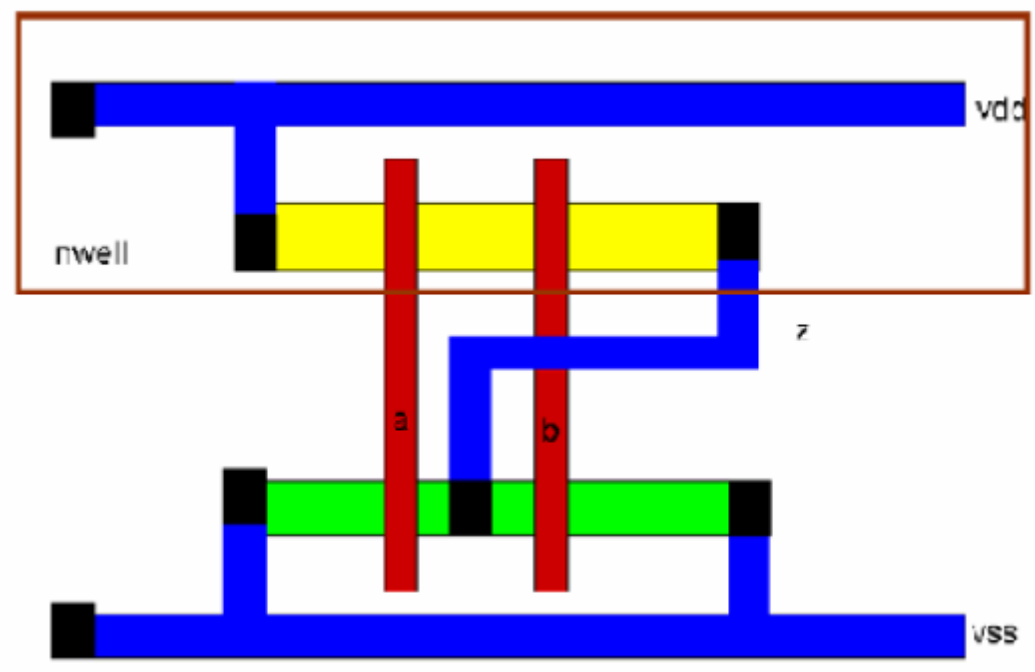


Figure 26: Layout of nor gate.

2.7 TRANSMISSION GATE

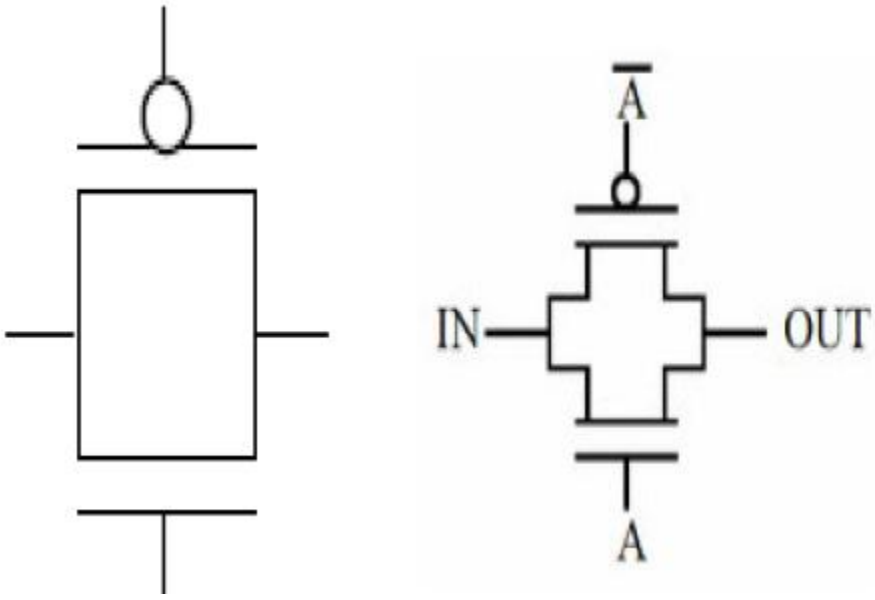


Figure 27: Symbol and schematic of transmission gate

Layout considerations of transmission gate. It consists of drains and the sources of the P&N devices paralleled. Transmission gate can replace the pass transistors and has the advantage of giving both a good one and a good zero.

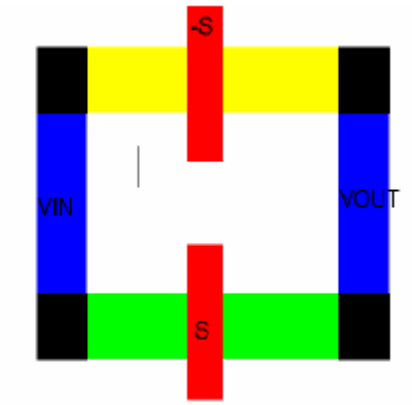


Figure 28: layout of transmission gate.

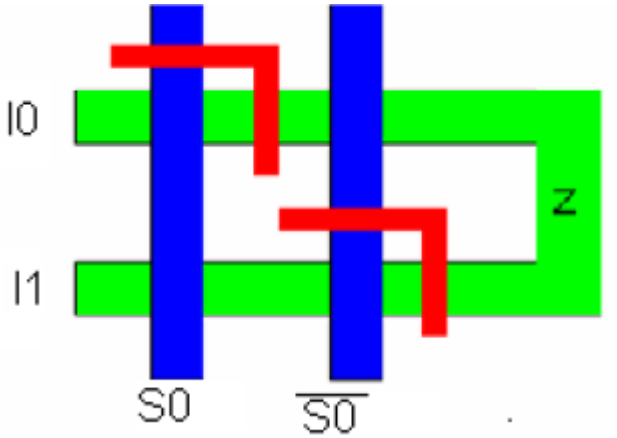


Figure 29: TG with nmos switches.

2.8 CMOS STANDARD CELL DESIGN

Geometric regularity is very important to maintain some common electrical characteristics between the cells in the library. The common physical limitation is to fix the height and vary the width according to the required function. The Wp and Wn are fixed considering power dissipation, propagation delay, area and noise immunity. The best thing to do is to fix a required objective function and then fix Wn and Wp to obtain the required objective Usually in CMOS Wn is made equal to Wp. In the process of designing these gates techniques may be employed to automatically generate the gates of common size. Later optimization can be carried out to achieve a specific feature. Gate array layout and sea of gate layout are constructed using the above techniques.

The gate arrays may be customized by having routing channels in between array of gates. The gate array and the sea of gates have some special layout considerations. **The gate arrays** use fixed image of the under layers i.e. the diffusion and poly are fixed and metal are programmable.

The wiring layers are discretionary and providing the personalization of the array. The rows of transistors are fixed and the routing channels are provided in between them. Hence the design issue involves size of transistors, connectivity of poly and the number of routing channels required. Sea of gates in this style continuous rows of n and p diffusion run across the master chip and are arranged without regard to the routing channel. Finally the routing is done across unused transistors saving space.

2.9 GENERAL LAYOUT GUIDELINES

1. The electrical gate design must be completed by checking the following
    a. Right power and ground supplies
    b. Noise at the gate input
    c. Faulty connections and transistors
    d. Improper ratios
    c. Incorrect clocking and charge sharing
2. VDD and the VSS lines run at the top and the bottom of the design
3. Vertical poysilicon for each gate input
4. Order polysilicon gate signals for maximal connection between transistors
5. The connectivity requires to place nmos close to VSS and pmos close to VDD
6. Connection to complete the logic must be made using poly, metal and even metal2

The design must always proceeds towards optimization. Here optimization is at transistor level rather then gate level. Since the density of transistors is large, we could obtain smaller and faster layout by designing logic blocks of 1000 transistors instead of considering a single at a time and then putting them together. Density improvement can also be made by considering optimization of the other factors in the layout.

The factors are

l. Efficient routing space usage. They can be placed over the cells or even in multiple layers.

2. Source drain connections must be merged better.

3. White (blank) spaces must be minimum

4. The devices must be of optimum sizes.

5. Transperent routing can be provided for cell to cell interconnection, this reduces global wiring problems

2.10 LAYOUT OPTIMIZATION FOR PERFORMANCE

l. Vary the size of the transistor according to its position in series. The transistor closest to the output is the smallest. The transistor nearest to the VSS line is the largest. This helps in increasing the performance by 30 %. A three input nand gate with the varying size is shown next.

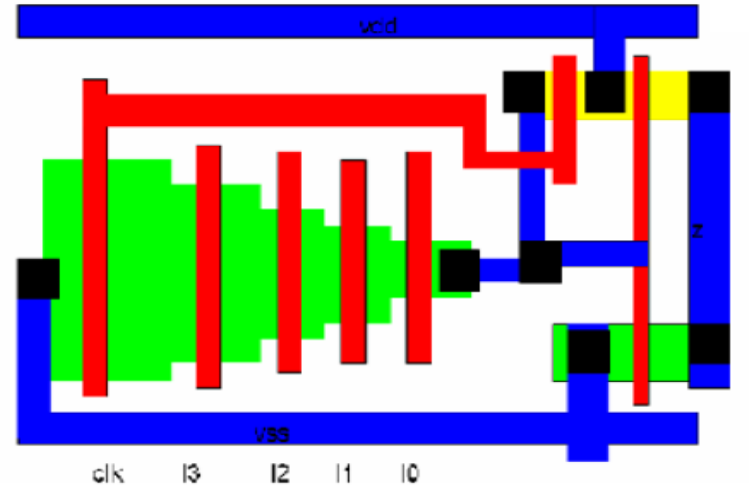


Figure 30: Layout optimization with varying diffusion areas.

2. Less optimized gates could occur even in the case of parallel connected transistors. This is usually seen in parallel inverters, nor & nand. When drains are connected in parallel, we must try and reduce the number of drains in parallel i.e. wherever possible we must try and connect drains in series at least at the output. This arrangement could reduce the capacitance at the output enabling good voltage levels. One example is as shown next.

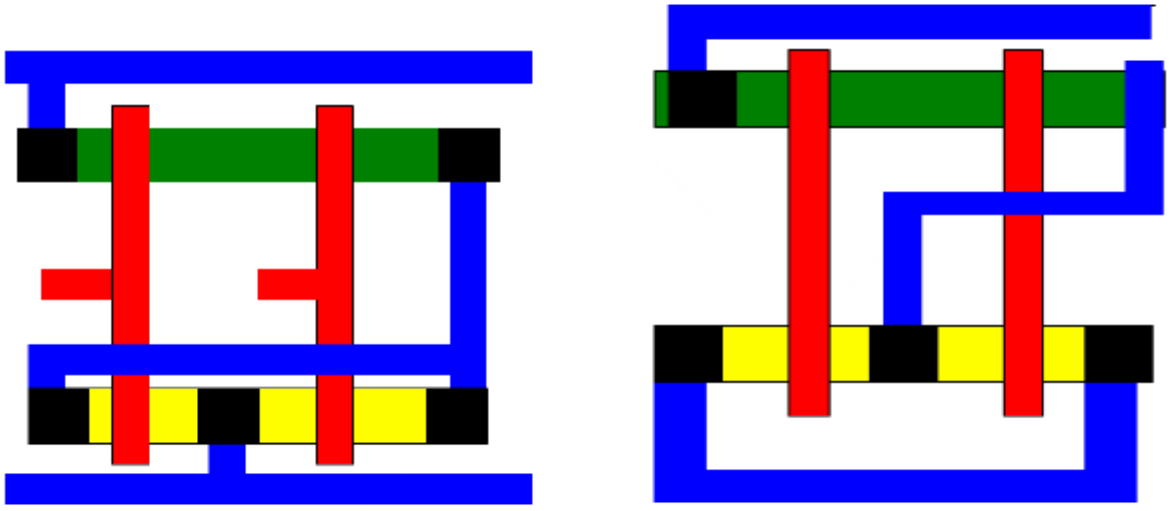


Figure 30: Layout of nor gate showing series and parallel drains.

Recommended questions:

1. What do you mean by MOS layers.
2. Define stick diagram.
3. Explain design rules and layout.
4. Explain lambda-based design rules and layout diagram with an example.
5. Explain physical design flow for a simple logic gates.
6. Explain with an example the design flow for basic gates.

UNIT 3

CMOS LOGIC STRUCTURES

CMOS complementary logic, BiCMOS logic, Pseudo-nMOS logic, Dynamic CMOS logic, clocked CMOS logic, Pass transistor logic, CMOS domino logic cascaded voltage switch logic (CVSL).

Recommended readings:

1. Douglas A. Pucknell & Kamran Eshraghian, **"Basic VLSI Design"** PHI 3rd Edition (original Edition – 1994), 2005.

2. Neil H. E. Weste and K. Eshragian**," Principles of CMOS VLSI Design: A Systems Perspective,"** 2nd edition, Pearson Education (Asia) Pvt. Ltd., 2000. History of VLSI

3.1 Introduction:

The various applications that require logic structures have different optimizations. Some of the circuit needs fast response, some slow but very precise response; others may need large functionality in a small space and so on. The CMOS logic structures can be implemented in alternate ways to get specific optimization. These optimizations are specific because of the tradeoff between the n numbers of design parameters.

3.2 CMOS COMPLEMENTARY LOGIC

CMOS logic structures of nand & nor has been studied in previous unit. They were ratioed logic i.e. they have fixed ratio of sizes for the n and the p gates. It is possible to have ratio less logic by varying the ratio of sizes which is useful in gate arrays and sea of gates. Variable ratios allow us to vary the threshold and speed .If all the gates are of the same size the circuit is likely to function more correctly. Apart from this the supply voltage can be increased to get better noise immunity. The increase in voltage must be done within a safety margin of the source -drain break down. Supply voltage can be decreased for reduced power dissipation and also meet the constraints of the supply voltage. Sometimes even power down with low power dissipation is required. For all these needs an on chip voltage regulator is required which may call for additional space requirement. A CMOS requires a nblock and a pblock for completion of the logic. That is for a n input logic 2n gates are required. The variations to this circuit can include the following techniques reduction of noise margins and reducing the function determining transistors to one polarity.

3.3 BICMOS Logic

The CMOS logic structures have low output drive capability. If bipolar transistors are used at the output the capability can be enhanced. Bipolar transistors are current controlled devices and produces larger output current then the CMOS transistors. This combined logic is called BICMOS logic. We can have the bipolar transistors both for pull up and pull down or only for pull up as shown in the figures below. The figure next shows a CMOS nand gate with NPN transistors at both levels. The Nl & N2 supply current to the base of the NPN2 transistor when the output is high and hence the it can pull it down with larger speed. When the output is low N3 clamps the base current to NPN2, Pl & P2 supply the base current to NPNl

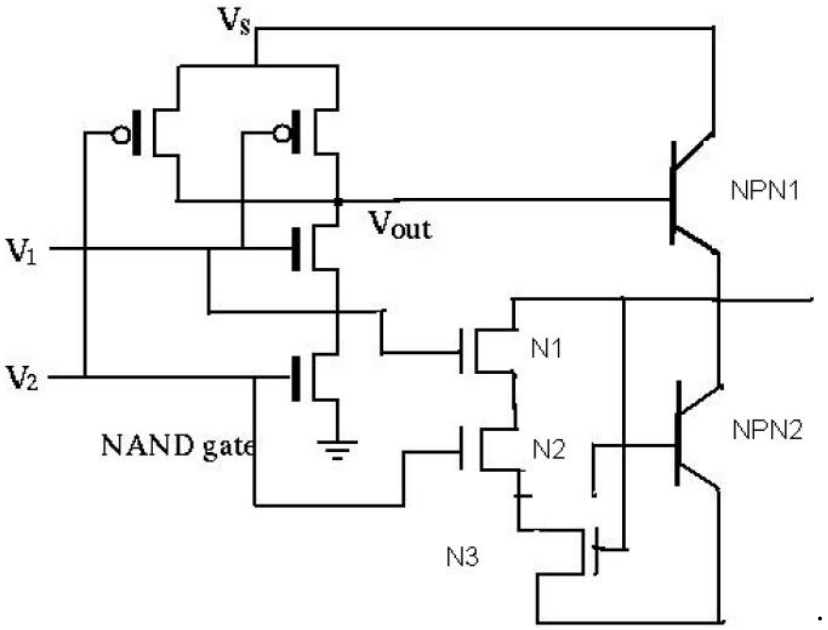


Figure 1: Nand with two NPN drivers

This design shown previously is basically used for speed enhancing in highly automated designs like gate arrays. Since the area occupied by the Bipolar transistors is more and if the aim in the design is to match the pull up and pull down speeds then we can have a transistor only in the pull up circuit because p devices are slower as shown in the figure next. The usage of BiCMOS must be done only after a trade off is made between the cost, performance etc.

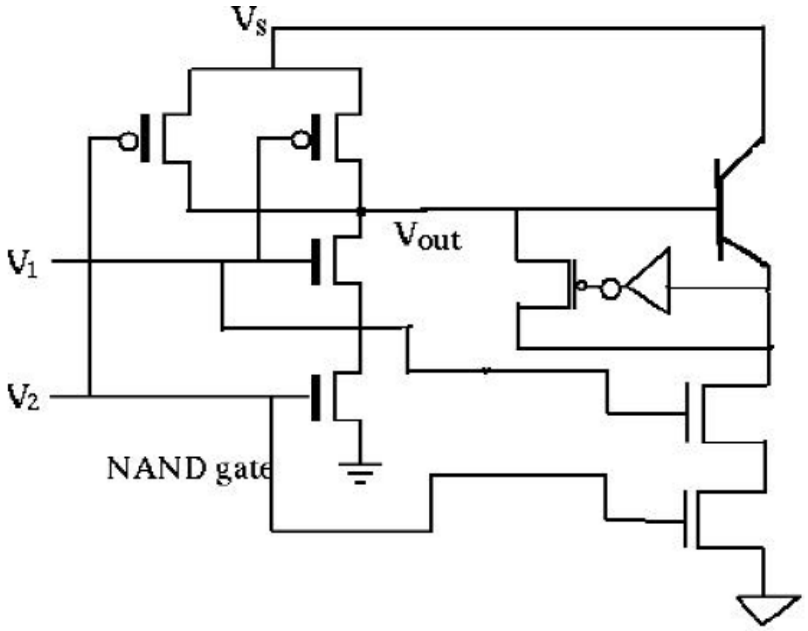


Figure 2: Nand with one NPN in pull up.

3.4 PSEUDO NMOS LOGIC

This logic structure consists of the pull up circuit being replaced by a single pull up pmos whose gate is permanently grounded. This actually means that pmos is all the time on and that now for a n input logic we have only n+1 gates. This technology is equivalent to the depletion mode type and preceded the CMOS technology and hence the name pseudo. The two sections of the device are now called as load and driver. The Gn/Gp (Gdriver/Gload) has to be selected such that sufficient gain is achieved to get consistent pull up and pull down levels. This involes having ratioed transistor sizes so that correct operation is obtained. However if minimum size drivers are being used then the gain of the load has to be reduced to get adequate noise margin.

There are certain drawbacks of the design which is highlighted next

1. The gate capacitance of CMOS logic is two unit gates but for pseudo logic it is only one gate unit.
2. Since number of transistors per input is reduced area is reduced drastically.

The disadvantage is that since the pMOS is always on, static power dissipation occurs whenever the nmos is on. Hence the conclusion is that in order to use pseudo logic a tradeoff between size & load or power dissipation has to be made.

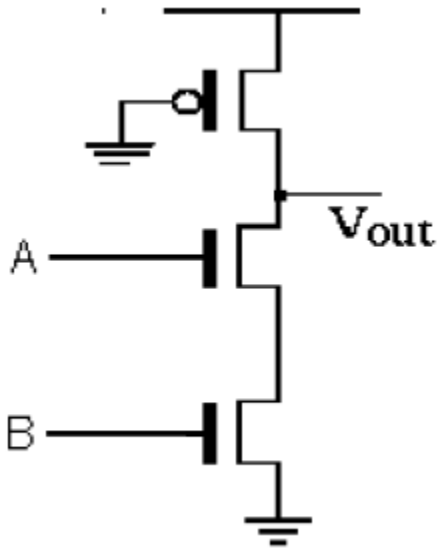


Figure 3: Pseudo Nmos

3.4.1 OTHER VARIATIONS OF PSEUDO NMOS

1. Multi drain logic

One way of implementing pseudo nmos is to use multi drain logic. It represents a merged transistor kind of implementation. The gates are combined in an open drain manner, which is useful in some automated circuits. Figure 4.

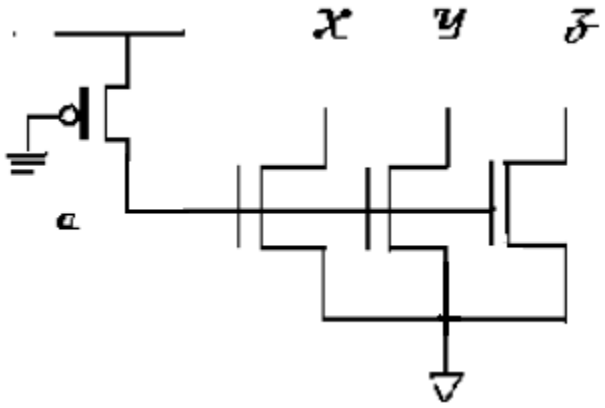
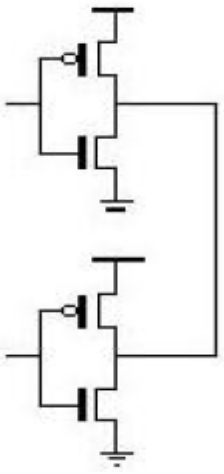


Figure 4: Multi drain logic.

3.4.2 GANGED LOGIC



The inputs are separately connected but the output is connected to a common terminal. The logic depends on the pull up and pull down ratio. If pmos is able to overcome nmos it behaves as nand else nor.

3.5 DYNAMIC CMOS LOGIC:

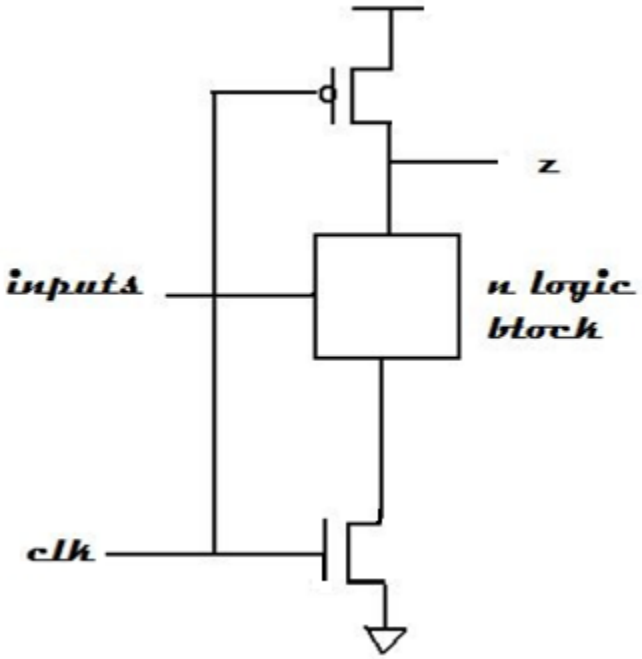


Figure 5: Dynamic CMOS logic

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 vss. 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

1. 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.
2. 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.

3.6 CLOCKED CMOS LOGIC (C2MOS)

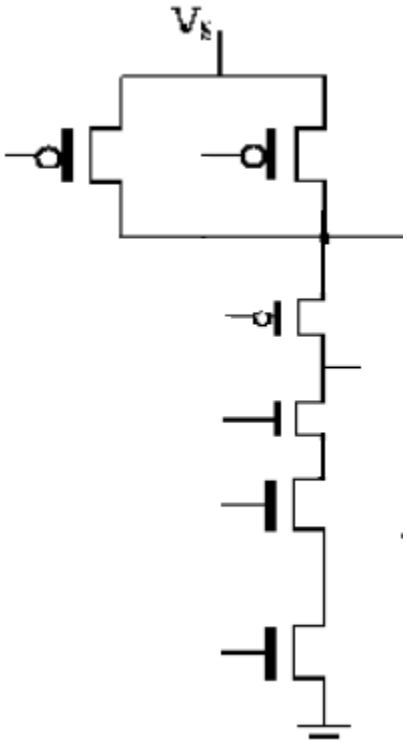


Figure 6: C2mos logic.

3.7 CMOS DOMINO LOGIC

The disadvantage associated with the dynamic CMOS is over come in this logic. In this we are able to cascade logic blocks with the help of a single clock. The precharge and the evaluate phases retained as they were. The change required is to add a buffer at the end of each stage. This logic works in the following manner. When the clk=0, ie during the precharge stage the output of the dynamic logic is high and the output of the buffer is low. Since the subsequent stages are fed from the buffer they are all off in the precharge stage. When the gate is evaluated in the next phase, the output conditionally goes low and the output of the buffer goes high. The subsequent gates make a transition from high to low.

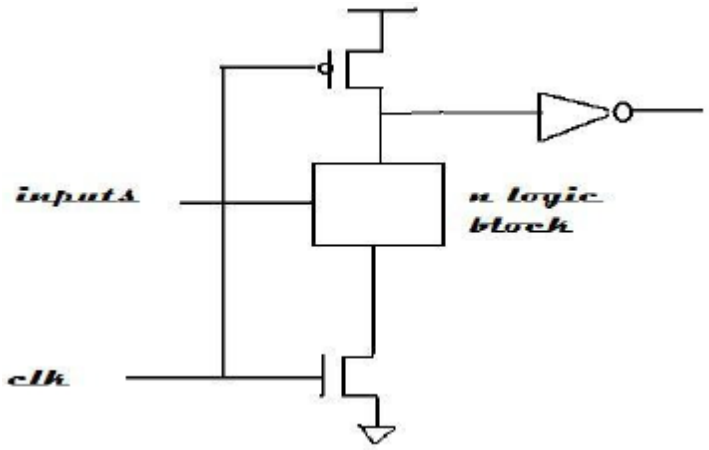


Figure 7: Cmos domino logic.

Hence in one clock cycle the cascaded logic makes only one transition from 1 to 0 and buffer makes a transition from 0 to 1.In effect we can say that the cascaded logic falls like a line of dominos, and hence the name. The advantage is that any number of logic blocks can be cascaded provided the sequence can be evaluated in a single clock cycle. Single clock can be used to precharge and evaluate all the logic in a block. The limitation is that each stage must be buffered and only non- inverted structures are possible.

A further fine tuning to the domino logic can also be done. Cascaded logic can now consist of alternate p and n blocks and avoid the domino buffer. When clk=0,ie during the precharge stage, the first stage (with n logic) is precharged high and the second a p logic is precharged low and the third stage is high. Since the second stage is low, the n transistor is off. Hence domino connections can be made.

The advantages are we can use smaller gates, achieve higher speed and get a smooth operation. Care must be taken to ensure design is correct.

3.7.1 NP DOMINO LOGIC (ZIPPER CMOS)

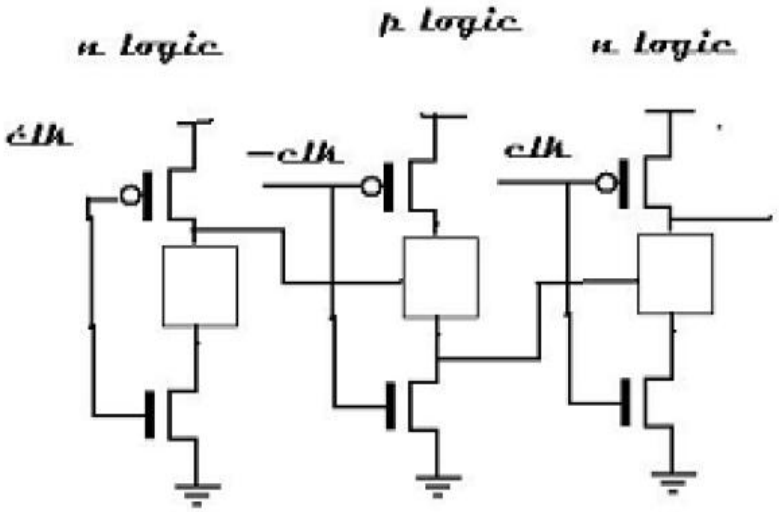


Figure 8: NP domino logic.

3.8 CASCADED VOLTAGE SWITCH LOGIC

It is a differential kind of logic giving both true and complementary signal outputs. The switch logic is used to connect a combinational logic block to a high or a low output. There are static and dynamic variants .The dynamic variants use a clock. The static version (all the figures to shown next) is slower because the pulls up devices have to overcome the pull down devices. Hence the clocked versions with a latching sense amplifier came up. These switch logic are called sample set differential logic.

3.8.1 STATIC CVSL

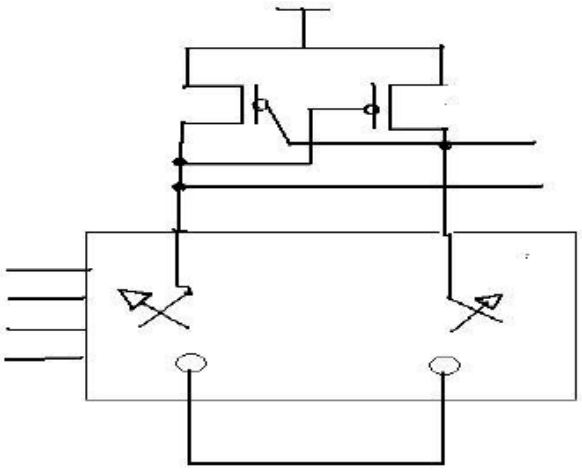


Figure 9: Static CVSL

3.8.2 DYNAMIC CVSL

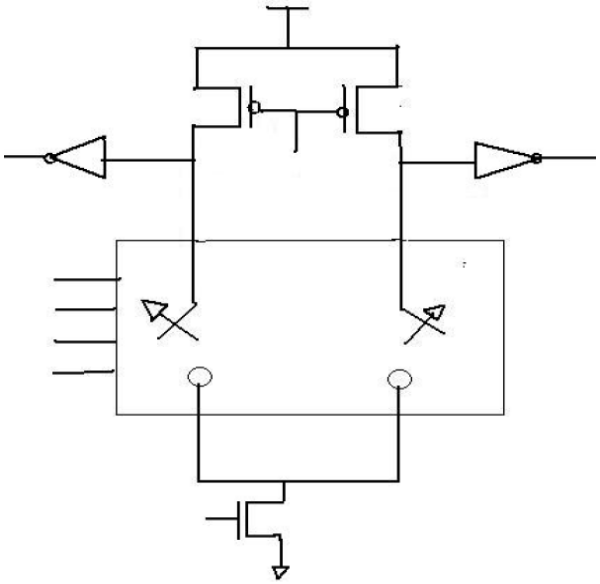


Figure 10: Dynamic CVSL

3.8.3 DYNAMIC SSDL CVSL

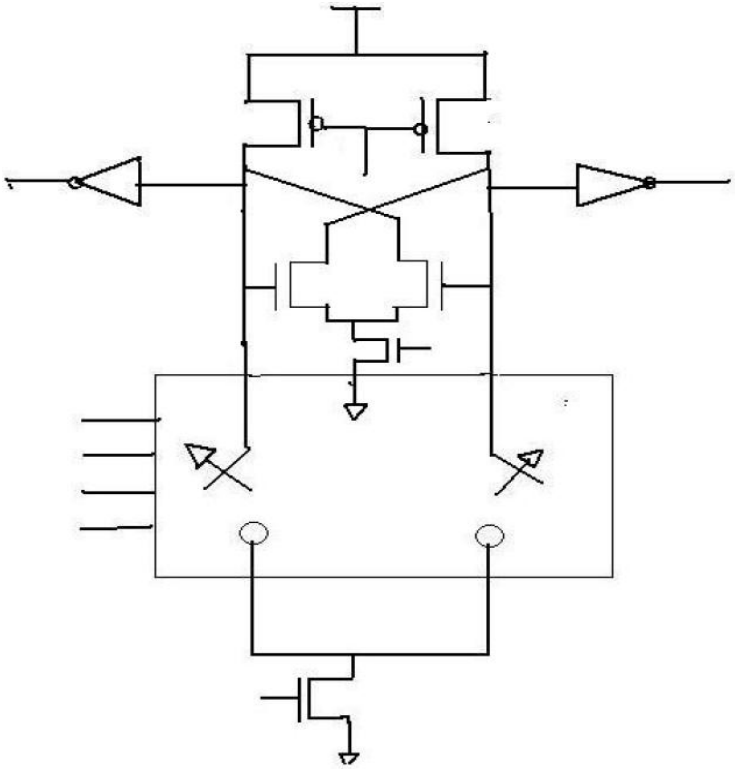


Figure 11: Dynamic SSDLCVSL.

3.9 PASS TRANSISTOR LOGIC

Switches and switch logic can be formed from simple n or p transistors and from the complementary switch i.e. the transmission gate. The complex transmission gate came into picture because of the undesirable threshold effects of the simple pass transistors. Transmission gate gives good non degraded logic levels. But this good package came at the cost of larger area and complementary signals required to drive the gates

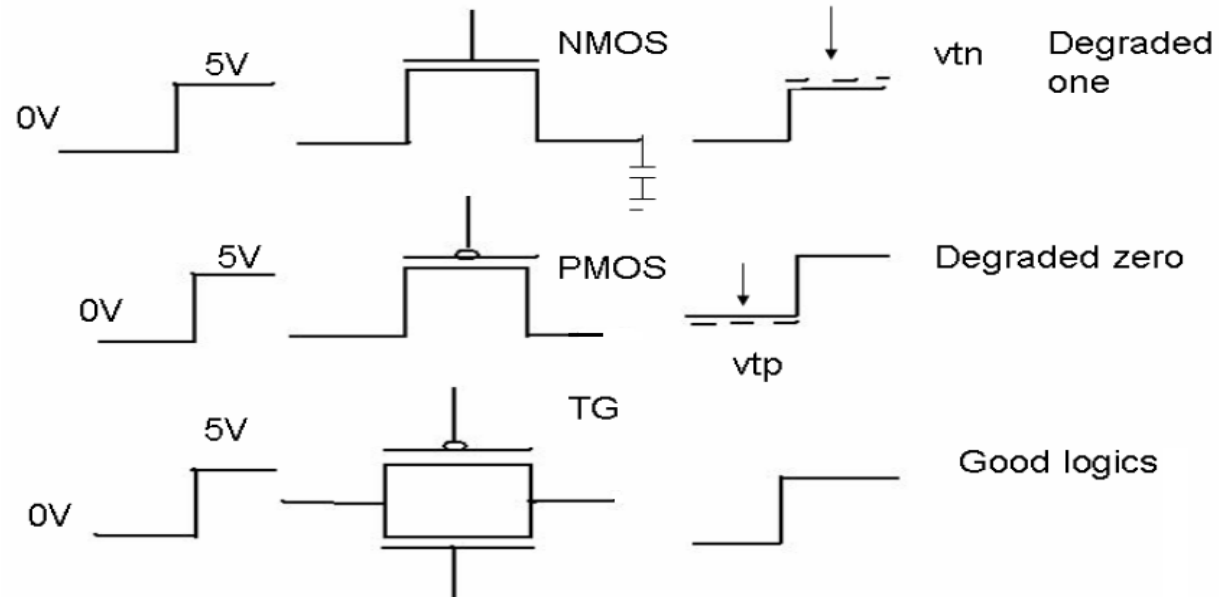


Figure 12: Some properties of pass transistor.

3.10 CMOS Technology Logic Circuit Structures

Many different logic circuits utilizing CMOS technology have been invented and used in various applications. These can be divided into three types or families of circuits:

1. Complementary Logic

Standard CMOS

Clocked CMOS (C2MOS)

BICMOS (CMOS logic with Bipolar driver)

2. Ratio Circuit Logic

Pseudo-NMOS

Saturated NMOS Load

Saturated PMOS Load

Depletion NMOS Load (E/D)

Source Follower Pull-up Logic (SFPL)

3. Dynamic Logic:

CMOS Domino Logic

NP Domino Logic (also called Zipper CMOS)

NOR A Logic

Cascade voltage Switch Logic (CVSL)

Sample-Set Differential Logic (SSDL)

Pass-Transistor Logic

The large number of implementations shown so far may lead to confusion as to what to use where. Here are some inputs

1. Complementary CMOS

The best option, because of the less dc power dissipation, noise immuned and fast. The logic is highly automated. Avoid in large fan outs as it leads to excessive levels of logic.

2. BICMOS

It can be used in high speed applications with large fan-out. The economics must be justified.

PSUEDO –NMOS

Mostly useful in large fan in NOR gates like ROMS, PLA and CLA adders. The DC power can be reduced to 0 in case of power down situations

Clocked CMOS

Useful in hot electron susceptible processes.

CMOS domino logic

Used mostly in high speed low power application. Care must take of charge redistribution. Precharge robs the speed advantage.

CVSL

This is basically useful in fast cascaded logic .The size; design complexity and reduced noise immunity make the design not so popular.

Hybrid designs are also being tried for getting the maximum advantage of each of them into one.

Recommended Questions:

1. Explain difference between BiCMOS and CMOS complementary logic.
2. Write a note on Pseudo-nMOS logic.
3. Write a note on dynamic CMOS logic and Clocked CMOS logic.
4. Explain pass transistor logic.
5. Explain CMOS domino logic with neat diagram.
6. Explain cascaded voltage switch logic.

Unit-4

Basic circuit concepts

Sheet resistance, area capacitances, capacitances calculations. The delay unit, inverter delays, driving capacitive loads, propagation delays, wiring capacitances.

Scaling of MOS circuits

Scaling models and factors, limits on scaling, limits due to current density and noise.

Recommended readings:

1. Douglas A. Pucknell & Kamran Eshraghian, **"Basic VLSI Design"** PHI 3rd Edition (original Edition – 1994), 2005.

2. Neil H. E. Weste and K. Eshragian**," Principles of CMOS VLSI Design: A Systems Perspective,"** 2nd edition, Pearson Education (Asia) Pvt. Ltd., 2000. History of VLSI

4.1 INTRODUCTION

We have already seen that MOS structures are formed by the super imposition of a number conducting, insulating and transistor forming material. Now each of these layers have their own characteristics like capacitance and resistances. These fundamental components are required to estimate the performance of the system. These layers also have inductance characteristics that are important for I/O behavior but are usually neglected for on chip devices.

The issues of prominence are

1. Resistance, capacitance and inductance calculations.
2. Delay estimations
3. Determination of conductor size for power and clock distribution
4. Power consumption
5. Charge sharing
6. Design margin
7. Reliability
8. Effects and extent of scaling

4.2 RESISTANCE ESTIMATION

The concept of sheet resistance is being used to know the resistive behavior of the layers that go into formation of the MOS device. Let us consider a uniform slab of conducting material of the following characteristics.

Resistivity-ρ

Width - W

Thickness - t

Length between faces – L as shown next

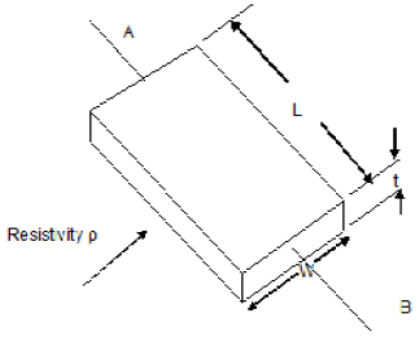


Figure 1: A slab of semiconductor.

We know that the resistance is given by RAB= ρ L/A Ω. The area of the slab considered above is given by A=Wt. Therefore RAB= ρ L/Wt Ω. If the slab is considered as a square then L=W. therefore RAB= ρ /t which is called as sheet resistance represented by Rs. The unit of sheet resistance is **ohm per square.** It is to be noted that Rs is independent of the area of the slab. Hence we can conclude that a 1um per side square has the same resistance as that of 1cm per side square of the same material. The resistances of the different materials that go into making of the MOS device depend on the resistivity and the thickness of the material. For a diffusion layer the depth defines the thickness and the impurity defines the resistivity. The table of values for a 5u technology is listed below.5u technology means minimum line width is 5u and λ= 2.5u.The diffusion mentioned in the table is n diffusion, p diffusion values are 2.5 times of that of n. The table of standard sheet resistance value follows.

Layer	Rs per square
Metal	0.03
Diffusion n(for 2.5 times the n)	10 to 50
Silicide	2 to 4
Polysilicon	15 to 100
N transistor gate	10^4
P transistor gate	$2.5x\ 10^4$

SHEET RESISTANCE OF MOS TRANSISTORS

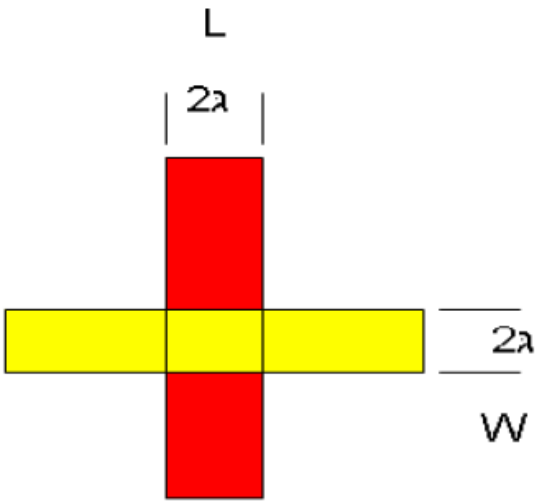


Figure 2: Min sized inverter.

The N transistor above is formed by a 2λ wide poly and n diffusion. The L/W ratio is 1. Hence the transistor is a square, therefore the resistance R is 1sqxRs ohm/sq i.e. R=1x104. If L/W ratio is 4 then R = $4x10^4$. If it is a P transistor then for L/W =1,the value of R is $2.5x10^4$.

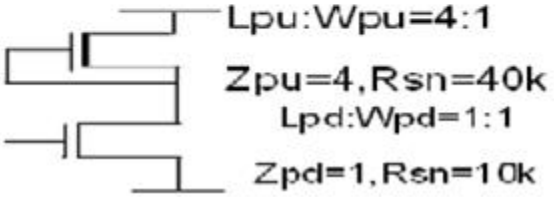


Figure 3: Nmos depletion inverter.

Pull up to pull down ratio = 4.In this case when the nmos is on, both the devices are on simultaneously, Hence there is an on resistance Ron = 40+10 =50k. It is this resistance that leads the static power consumption which is the disadvantage of nmos depletion mode devices

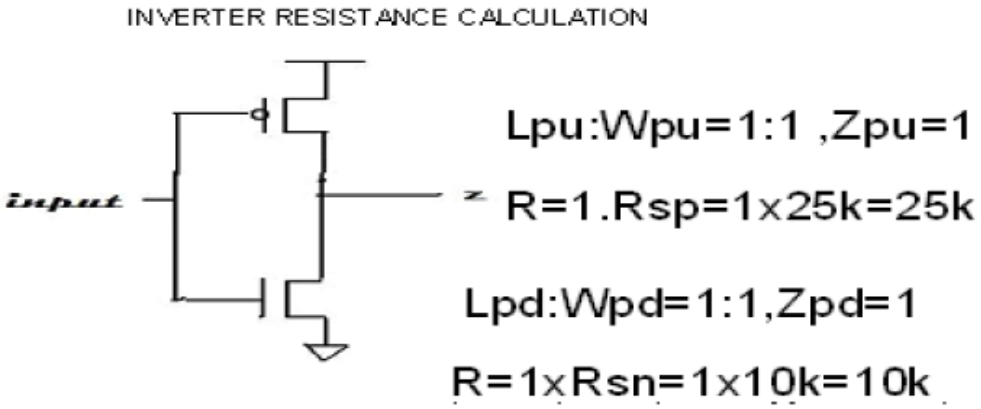


Figure 4: CMOS inverter.

Since both the devices are not on simultaneously there is no static power dissipation The resistance of non rectangular shapes is a little tedious to estimate. Hence it is easier to convert the irregular shape into regular rectangular or square blocks and then estimate the resistance. For example

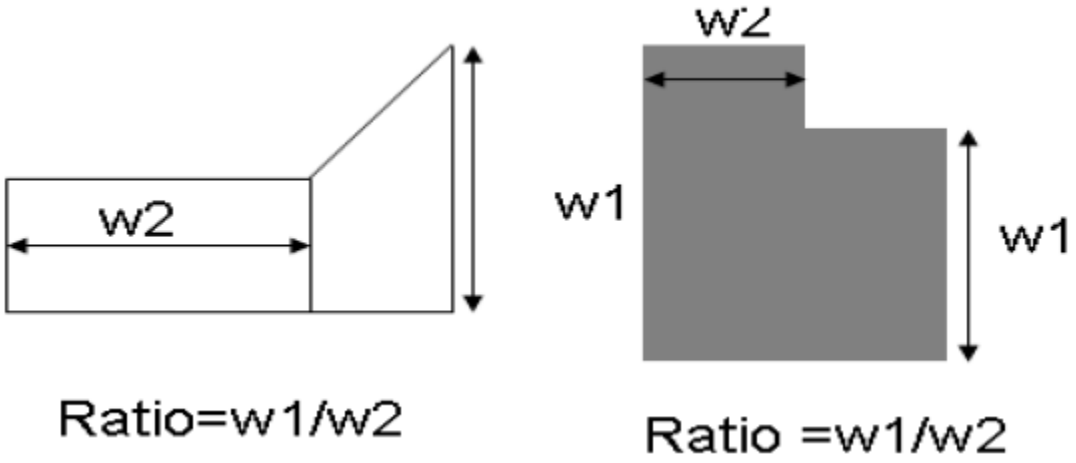


Figure 5: Irregular rectangular shapes.

CONTACT AND VIA RESISTANCE

The contacts and the vias also have resistances that depend on the contacted materials and the area of contact. As the contact sizes are reduced for scaling ,the associated resistance increases. The resistances are reduced by making ohmic contacts which are also called loss less contacts. Currently the values of resistances vary from .25ohms to a few tens of ohms.

SILICIDES

The connecting lines that run from one circuit to the other have to be optimized. For this reason the width is reduced considerably. With the reduction is width the sheet resistance increases, increasing the RC delay component. With poly silicon the sheet resistance values vary from 15 to 100 ohm. This actually effects the extent of scaling down process. Polysilicon is being replaced with silicide. Silicide is obtained by depositing metal on polysilicon and then sintering it. Silicides give a sheet resistance of 2 to 4 ohm. The reduced sheet resistance makes silicides a very attractive replacement for poly silicon. But the extra processing steps is an offset to the advantage.

A Problem

A particular layer of MOS circuit has a resistivity ρ of 1 ohm –cm. The section is 55um long, 5um wide and 1 um thick. Calculate the resistance and also find Rs

R= RsxL/W, Rs= ρ/t

Rs=1x10-2/1x10-6=104ohm

R= 104x55x10-6/5x106=110k

CAPACITANCE ESTIMATION

Parasitics capacitances are associated with the MOS device due to different layers that go into its formation. Interconnection capacitance can also be formed by the metal, diffusion and polysilicon (these are often called as runners) in addition with the transistor and conductor resistance. All these capacitances actually define the switching speed of the MOS device. Understanding the source of parasitics and their variation becomes a very essential part of the design specially when system performance is measured in terms of the speed. The various capacitances that are associated with the CMOS device are

1. Gate capacitance - due to other inputs connected to output of the device
2. Diffusion capacitance - Drain regions connected to the output
3. Routing capacitance- due to connections between output and other inputs

The fabrication process illustrates that the conducting layers are apparently separated from the substrate and other layers by the insulating layer leading to the formation of parallel capacitors. Since the silicon dioxide is the insulator knowing its thickness we can calculate the capacitance

$$C = \frac{\varepsilon_o \varepsilon_{ins} A}{D} \quad \text{farad}$$

□εo= permittivity of free space-8.854x1014f/cm

εins= relative permitivity of sio2=4.0

D= thickness of the dioxide in cm

A = area of the plate in cm2

The gate to channel capacitance formed due to the sio2 separation is the most profound of the mentioned three types. It is directly connected to the input and the output. The other capacitance like the metal, poly can be evaluated against the substrate. The gate capacitance is therefore standardized so as to enable to move from one technology to the other conveniently. The standard unit is denoted by □Cg. It represents the capacitance between gate to channel with

W=L=min feature size. Here is a figure showing the different capacitances that add up to give the total gate capacitance

Cgd, Cgs = gate to channel capacitance lumped at the source and drain

Csb, Cdb = source and drain diffusion capacitance to substrate

Cgb = gate to bulk capacitance

Total gate capacitance Cg = Cgd+Cgs+Cgb

Since the standard gate capacitance has been defined, the other capacitances like polysilicon, metal, diffusion can be expressed in terms of the same standard units so that the total capacitance can be obtained by simply adding all the values. In order to express in standard values the following steps must be followed

1. Calculate the areas of area under consideration relative to that of standard gate i.e.$4\lambda^2$. (standard gate varies according to the technology)

2. Multiply the obtained area by relative capacitance values tabulated.

3. This gives the value of the capacitance in the standard unit of capacitance Cg.

Table 1: Relative value of Cg

layer	Relative value for5u technology
Gate to channel	1
Diffusion	0.25
Poly to sub	0.1
M1 to sub	0.075
M2 to sub	0.05
M2 to M1	0.1
M2 to poly	0.075

For a 5u technology the area of the minimum sized transistor is 5uX5u=25um2 ie λ=2.5u, hence,area of minimum sized transistor in lambda is 2 λX 2 λ = 4 λ2.Therefore for 2u or 1.2u or any other technology the area of a minimum sized transistor in lambda is 4 λ2. Lets solve a few problems to get to know the things better.

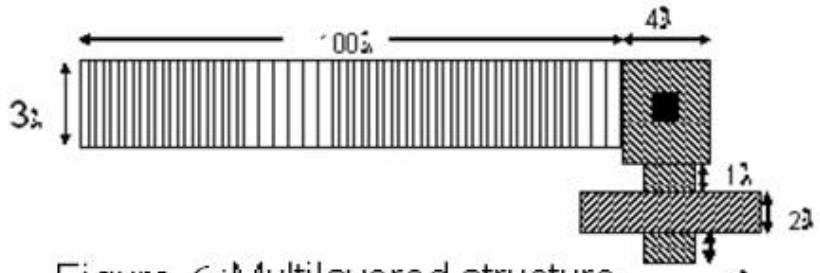


Figure 6 :Multilayered structure

The figure above shows the dimensions and the interaction of different layers, for evaluating the total capacitance resulting so.

Three capacitance to be evaluated metal Cm,polysilicon Cp and gate capacitance Cg

Area of metal = 100x3=300λ2

Relative area = 300/4=75

Cm=75Xrelative cap=75X0.075=5.625□Cg

Polysilicon capacitance Cp

Area of poly=(4x4+1x2+2X2)=22λ2

Relative area = 22λ2/4λ 2=5.5

Cp=5.5Xrelative cap=5.5x.1=0.55 □Cg

Gate capacitance Cg= 1□Cg because it is a min size gate

Ct=Cm+Cp+Cg=5.625+0.55+1=7.2 □Cg

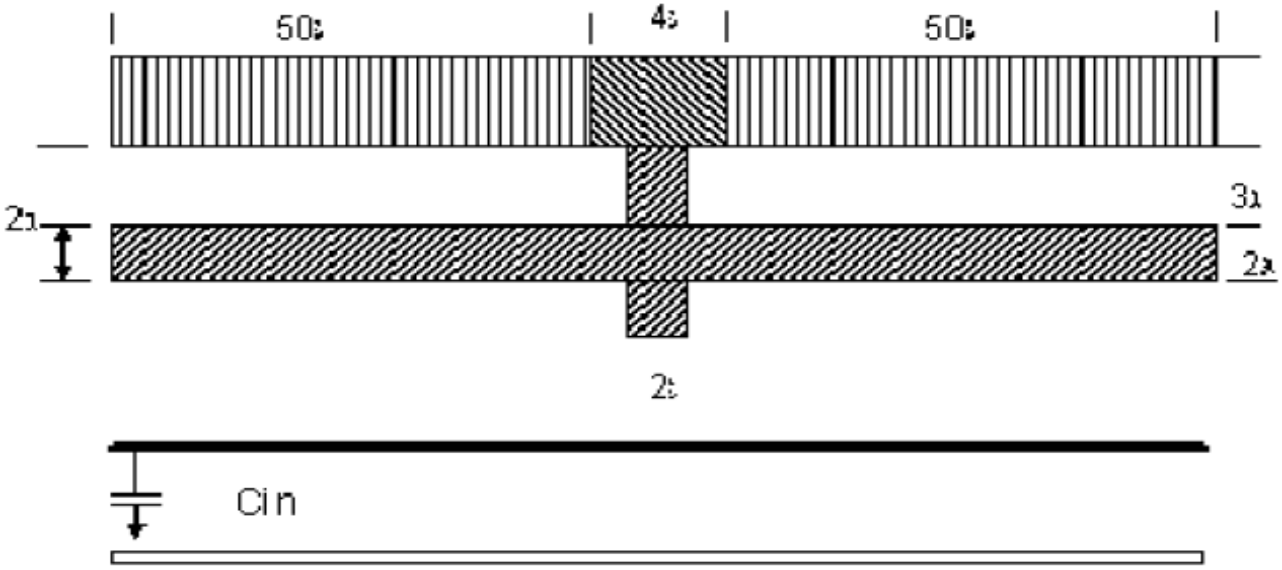


Figure 7 :Mos structure

The input capacitance is made of three components metal capacitance Cm, poly capacitance Cp, gate capacitance Cg i.e Cin= Cm+Cg+Cp

Relative area of metal =(50x3)X2/4=300/4=75

Cm=75x0.075=5.625¤Cg

Relative area of poly = (4x4+2x1+2x2)/4 =22/4 =5.5

Cp=5.5X0.1=0.55 ¤Cg

Cg=1 ¤Cg

Cin=7.175 ¤Cg

Cout = Cd+Cperi. Assuming Cperi to be negligible.

Cout = Cd.

Relative area of diffusion=51x2/4=102/4=25.5

Cd=25.5x0.25=6.25 ¤Cg.

The relative values are for the 5um technology

DELAY The concept of sheet resistance and standard unit capacitance can be used to calculate the delay. If we consider that a one feature size poly is charged by one feature size diffusion then the delay is Time constant 1□= Rs (n/p channel)x 1¤Cg secs. This can be evaluated for any technology. The value of ¤Cg will vary with different technologies because of the variation in the minimum feature size.

5u using n diffusion=104X0.01=0.1ns safe delay 0.03nsec

2um = 104x0.0032=0.064 nsecs safe delay 0.02nsec

1.2u= 104x0.0023 = 0.046nsecs safe delay =0.1nsec

These safe figures are essential in order to anticipate the output at the right time

4.3 INVERTER DELAYS

We have seen that the inverter is associated with pull up and pull down resistance values. Specially in nmos inverters. Hence the delay associated with the inverter will depend on whether it is being turned off or on. If we consider two inverters cascaded then the total delay will remain constant irrespective of the transitions. Nmos and CMOS inverter delays are shown next.

NMOS INVERTER

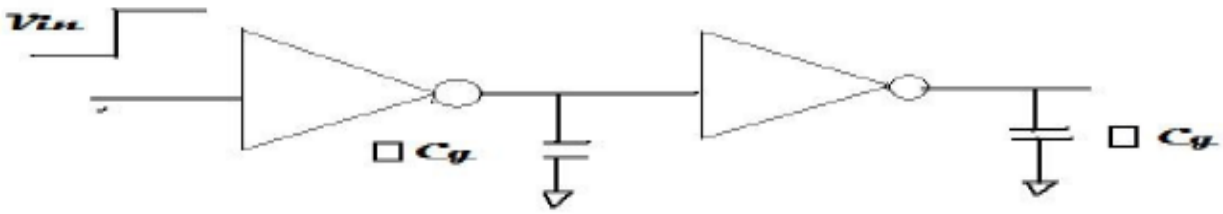


Figure 8: Cascaded nmos inverters.

Let us consider the input to be high and hence the first inverter will pull it down. The pull down inverter is of minimum size nmos. Hence the delay is 1□. Second inverter will pull it up and it is 4 times larger, hence its delay is 4□.The total delay is 1□ +4□= 5□. Hence for nmos the delay can be generalized as T=(1+Zpu/Zpd) □

4.4 CMOS INVERTER:

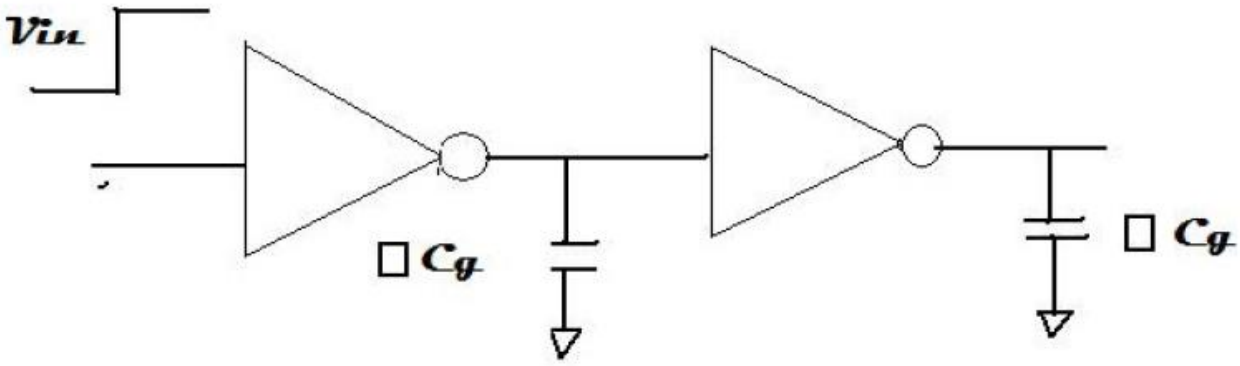


Figure 9: Cascaded CMOS inverter.

Let us consider the input to be high and hence the first inverter will pull it down. The nmos transistor has Rs = 10k and the capacitance is 2Cg. Hence the delay is 2□. Now the second inverter will pull it up, job done by the pmos. Pmos has sheet resistance of 25k i.e 2.5 times more, everything else remains same and hence delay is 5□. Total delay is 2□ +5□ = 7□ . The capacitance here is double because the input is connected to the common poly, putting both the gate capacitance in parallel. The only factor to be considered is the resistance of the p gate which is increasing the delay. If want to reduce delay, we must reduce resistance. If we increase the width of p channel, resistance can be reduced but it increases the capacitance. Hence some trade off must be made to get the appropriate values.

4.5 FORMAL ESTIMATION OF DELAY

The inverter either charges or discharges the load capacitance CL. We could also estimate the delay by estimating the rise time and fall time theoretically.

Rise time estimation

Assuming that the p device is in saturation we have the current given by the equation Idsp=ßp(Vgs-|Vtpl)2/2

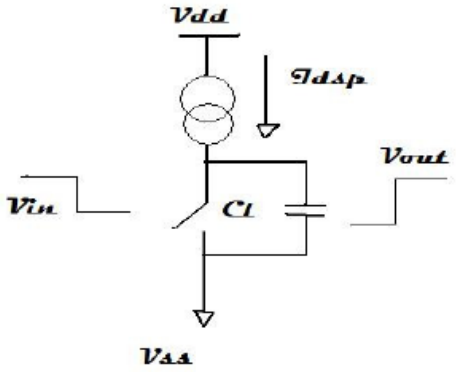


Figure 10: Rise time estimation.

The above current charges the capacitance and it has a constant value therefore the model can be written as shown in figure above. The output is the drop across the capacitance, given by

Vout =Idsp x t/CL

Substituting for Idsp we have Vout=ßp(Vgs-|Vtpl)2t/2CL. Therefore the equation for t=2CLVout/ßp(Vgs-|Vtpl).Let t=□r and Vout=Vdd, therefore we have □r = 2VddCL/ßp(Vgs-|Vtpl)2. If consider Vtp=0.2Vdd and Vgs=Vdd we have □r = 3CL/ßpVdd

On similar basis the fall time can be also be written as □f = 3CL/ßnVdd whose model can be written as shown next

4.6 DRIVING LARGE CAPACITIVE LOAD

The problem of driving large capacitive loads arises when signals must travel outside the chip. Usually it so happens that the capacitance outside the chip are higher. To reduce the delay these loads must be driven by low resistance. If we are using a cascade of inverter as drivers the pull and pull down resistances must be reduced. Low résistance means low L: W ratio. To reduce the ratio, W must be increased. Since L cannot be reduced to lesser than minimum we end up having a device which occupies a larger area. Larger area means the input capacitance increases and slows down the process more. The solution to this is to have N cascaded inverters with their sizes increasing, having the largest to drive the load capacitance. Therefore if we have 3 inverters,1st is smallest and third is biggest as shown next.

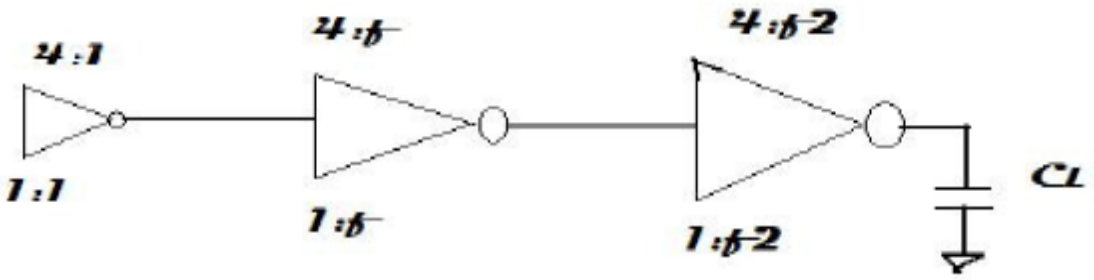


Figure 11: Cascaded inverters with varying widths.

We see that the width is increasing by a factor of f towards the last stage. Now both f and N can be complementary. If f for each stage is large the number of stages N reduces but delay per stage increases. Therefore it becomes essential to optimize. Fix N and find the minimum value of f. For nmos inverters if the input transitions from 0 to 1 the delay is f□ and if it transitions from 1 to 0 the delay is 4 f□. The delay for a nmos pair is 5 f□. For a cmos pair it will be 7f□

optimum value of f.

Assume y=CL/ ¤Cg = fN, therefore choice of values of N and f are interdependent. We find the value of f to minimize the delay, from the equation of y we have ln(y)=Nln(f) i.e N=ln(y)/ln(f). If delay per stage is 5f□ for nmos, then for even number of stages the total delay is N/2 5f□=2.5f□. For cmos total delay is N/2 7f□ = 3.5f□

Hence delay ά Nft=ln(y)/ln(f)ft. Delay can be minimized if chose the value of f to be equal to e which is the base of natural logarithms. It means that each stage is 2.7wider than its predecessor. If f=e then N= ln(y).The total delay is then given by

1.For N=even

td=2.5Ne□ for nmos, td=3.5Ne□ for cmos

2.For N=odd

transition from 0 to 1 transition from1 to 0

td=[2.5(N-1)+1]e□ nmos td=[2.5(N-1)+4]e□

td=[3.59N-1)+2]e□ cmos td=[3.5(N-1)+5]e□

for example

For N=5 which is odd we can calculate the delay fro vin=1 as td=[2.5(5-1)+1]e□ =11e □

i.e. 1 +4+1+4+1 = 11e□

For vin =0 , td=[2.5(5-1)+4]e□ = 14e□

4+1+4+1+4 = 14e□

4.7 SUPER BUFFER

The asymmetry of the inverters used to solve delay problems is clearly undesirable, this also leads to more delay problems, super buffer are a better solution. We have a inverting and non inverting variants of the super buffer. Such arrangements when used for 5u technology showed that they were capable of driving 2pf capacitance with 2nsec rise time. The figure shown next is the inverting variant.

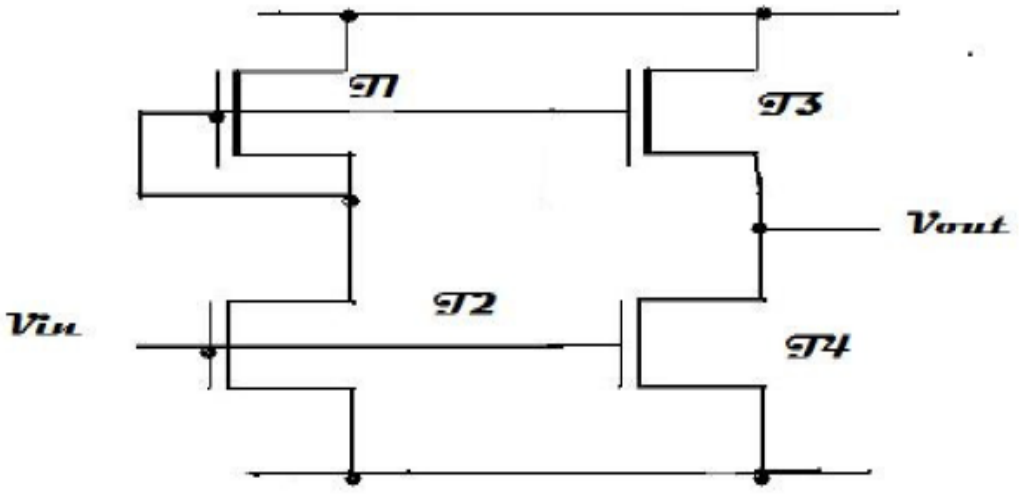


Figure 12: Inverting buffer.

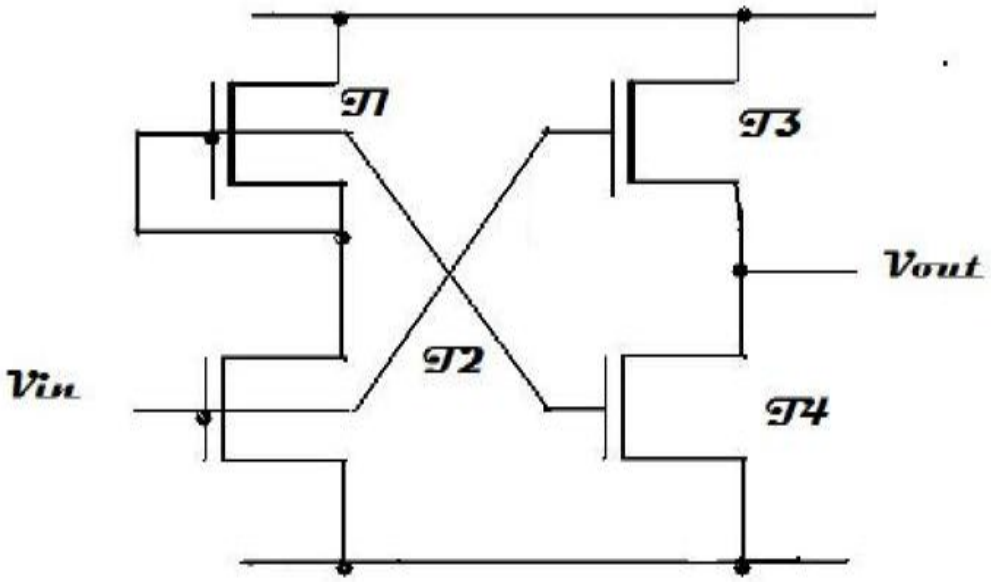
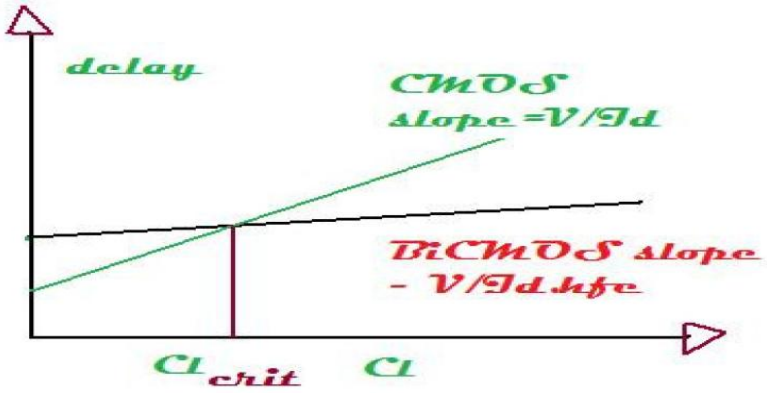


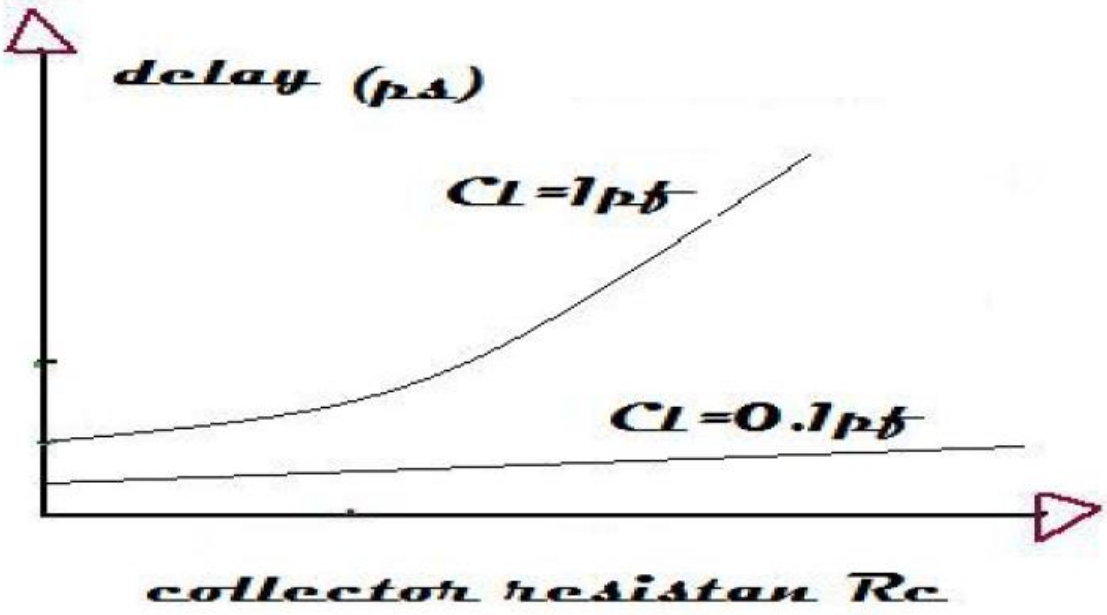
Figure 13: NonInverting buffer.

4.8 BICMOS DRIVERS

The availability of bipolar devices enables us to use these as the output stage of inverter or any logic. Bipolar devices have high Tran con--ductance and they are able switch large currents with smaller input voltage swings. The time required to change the out by an amount equal to the input is given by $\Delta t=CL/gm$, Where gm is the device trans conductance. Δt will be a very small value because of the high gm. The transistor delay consists of two components Tin and TL. Tin the time required to charge the base of the transistor which is large. TL is smaller because the time take to charge capacitor is less by hfe which is the transistor gain a comparative graph shown below.



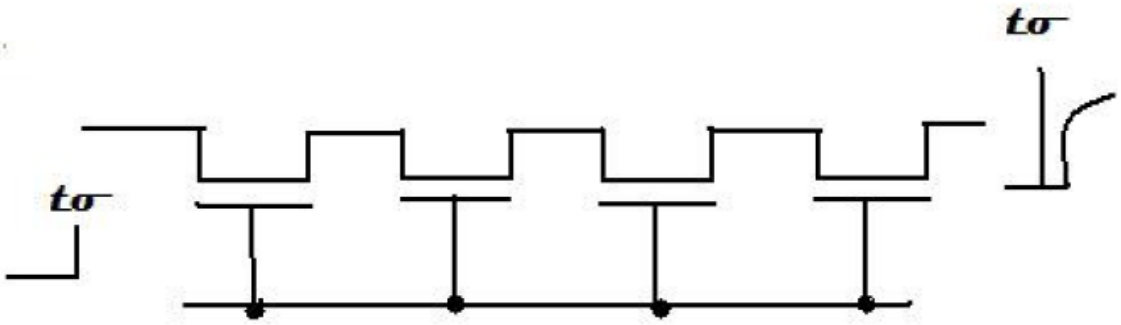
The collector resistance is another parameter that contributes to the delay.The graph shown below shows that for smaller load capacitance, the delay is manageable but for large capacitance, as Rc increases the delay increase drastically.

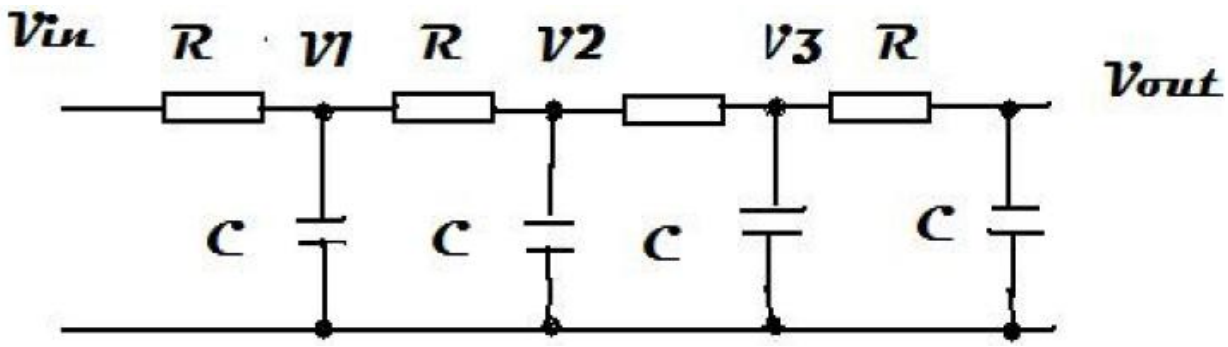


By taking certain care during fabrication reasonably good bipolar devices can be produced with large hfe, gm ,ß and small Rc. Therefore bipolar devices used in buffers and logic circuits give the designers a lot of scpoe and freedom .This is coming without having to do any changes with the CMOS circuit.

4.9 PROPAGATION DELAY

This is delay introduced when the logic signals have to pass through a chain of pass transistors. The transistors could pose a RC product delay and this increases drastically as the number of pass transistor in series increases.As seen from the figure the response at node V2 is given by CdV2/dt=(V1-V2)(V2-V3)/R For a long network we can write RCdv/dt =dv2/dx2, i.e delay ά x2,





Lump all the R and C we have Rtotal=nrRs and C=nc□Cg where and hence delay $=n^2rc\square$. The increases by the square of the number, hence restrict the number of stages to maximum 4 and for longer ones introduce buffers in between.

4.10 DESIGN OF LONG POLYSILICONS

The following points must be considered before going in for long wire.

1. The designer is also discouraged from designing long diffusion lines also because the capacitance is much larger.
2. When it inevitable and long poly lines have to used the best way to reduce delay is use buffers in between. Buffers also reduce the noise sensitivity.

4.11 OTHER SOURCES OF CAPACITANCE

Wiring capacitance

1. Fringing field
2. Interlayer capacitance
3. Peripheral capacitance

The capacitances together add upto as much capacitance as coming from the gate to source and hence the design must consider points to reduce them.The major of the wiring capacitance is coming from fringing field effects. Fringing capacitances is due to parallel fine metal lines running across the chip for power conection.The capacitance depends on the length l, thickness t and the distance d between the wire and the substrate. The accurate prediction is required for performance estimation.Hence Cw=Carea+Cff.

Interlayer capacitance is seen when different layers cross each and hence it is neglected for simole calculations. Such capacitance can be easily estimated for regular structures and helps in modeling the circuit better.

Peripheral capacitance is seen at the junction of two devices. The source and the drain n regions form junctions with the pwell (substrate) and p diffusion form with adjacent nwells leading to these side wall (peripheral) capacitance

The capacitances are profound when the devices are shrunk in sizes and hence must be considered. Now the total diffusion capacitance is Ctotal = Carea + Cperi

In order to reduce the side wall effects, the designers consider to use isolation regions of alternate impurity.

CHOICE OF LAYERS

1.Vdd and Vss lines must be distributed on metal lines except for some exception

2.Long lengths of poly must be avoided because they have large Rs,it is not suitable for routing Vdd or Vss lines.

3.Since the resistance effects of the transistors are much larger, hence wiring effects due to voltage dividers are not that profound

Capacitance must be accurately calculated for fast signal lines usually those using high Rs material. Diffusion areas must be carefully handled because they have larger capacitance to substrate.

With all the above inputs it is better to model wires as small capacitors which will give electrical guidelines for communication circuits.

PROBLEMS

1.A particular section of the layout includes a 3λ wide metal path which crosses a 2λ polysilicon path at right angles. Assuming that the layers are seperated by a 0.5 thick sio2,find the capacitance between the two.

Capacitance = □0 □ins A/D

Let the technology be 5um, λ=2.5um.

Area = 7.5umX5um=37.5um

C=4X8.854X10-12 x37.5/ 0.5 =2656pF

The value of C in standard units is

Relative area 6 λ2 /4 λ2 =1.5

C =1.5x0.075=0.1125□Cg

2 nd part of the problem

The polysilicon turns across a 4λ diffusion layer, find the gate to channel capacitance.

Area = 2 λx 4λ =8 λ2 Relative area= 8 λ2 / 4 λ2 =2

Relative capacitance for 5u=1

Total gate capacitance = 2□Cg

Gate to channel capacitance>metal

2. The two nmos transistors are cascaded to drive a load capacitance of 16¤Cg as shown in figure ,Calculate the pair delay. What are the ratios of each transistors. f stray and wiring capacitance is to be considered then each inverter will have an additional capacitance at the output of 4 ¤Cg .Find the delay.

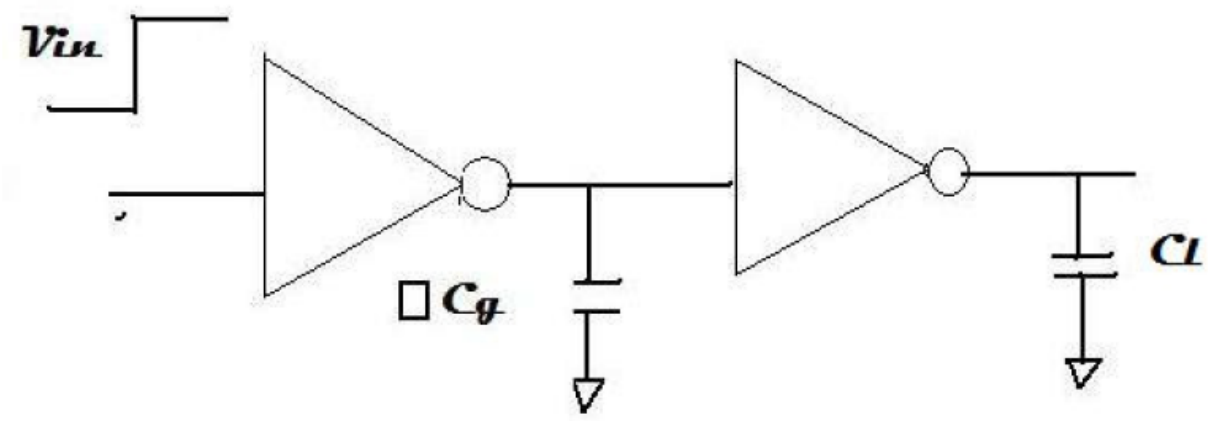


Figure 40

Lpu=16ƛ Wpu=2 ƛ Zpu=8

Lpd=2 ƛ Wpd=2 ƛ Zpd=1

Ratio of inverter 1 = 8:1

Lpu=2 ƛ Wpu=2 ƛ Zpu=1

Lpd =2 ƛ Wpd =8 ƛ Zpd=1/4

Ratio of inverter 2 = 1/1/4=4

Delay without strays

1□=Rsx1¤Cg

Let the input transition from 1 to 0

Delay 1 = 8RsX¤Cg=8□ Delay 2=4Rs(¤Cg +16 ¤Cg)=68□ Total delay = 76□

Delay with strays

Delay 1 = 8RsX(¤Cg+ 4¤Cg) = 40□ Delay 2= 4RsX(¤Cg+ 4¤Cg +16 ¤Cg)=84□

Total delay = 40+84=124□

If □ = 0.1ns for 5u ie the delays are 7.6ns and 12.4ns

4.12 SCALING OF MOS DEVICES

The VLSI technology is in the process of evolution leading to reduction of the feature size and line widths. This process is called scaling down. The reduction in sizes has generally lead to better performance of the devices. There are certain limits on scaling and it becomes important to study the effect of scaling. The effect of scaling must be studied for certain parameters that effect the performance.

The parameters are as stated below

1.Minimum feature size

2.Number of gates on one chip

3.Power dissipation

4.Maximum operational frequency

5.Die size

6.Production cost .

These are also called as figures of merit

Many of the mentioned factors can be improved by shrinking the sizes of transistors, interconnects, separation between devices and also by adjusting the voltage and doping levels. Therefore it becomes essential for the designers to implement scaling and understand its effects on the performance

There are three types of scaling models used

1.Constant electric field scaling model

2.Constant voltage scaling model

3.Combined voltage and field model

The three models make use of two scaling factors 1/ß and 1/ά . 1/ß is chosen as the scaling factor for Vdd, gate oxide thickness D. 1/ ά is chosen as the scaling factor for all the linear dimensions like length, width etc. the figure next shows the dimensions and their scaling factors

The following are some simple derivations for scaling down the device parameters

1.Gate area Ag

Ag= L x W. Since L & W are scaled down by 1/ ά. Ag is scaled down by $1/ \acute{\alpha}^2$

2.Gate capacitance per unit area

Co=□o/D, permittivity of sio2 cannot be scaled, hence Co can be scaled 1/1/ß=ß

3.Gate capacitance Cg

Cg=CoxA=CoxLxW. Therefore Cg can be scaled by ßx1/ άx1/ ά= ß/ $ά^2$

4.Parasitic capacitance

Cx=Ax/d, where Ax is the area of the depletion around the drain or source. d is the depletion width .Ax is scaled down by 1/ά2 and d is scaled by 1/ά. Hence Cx is scaled by

1/ $ά^2$ /1/ ά = 1/ ά

5.Carrier density in the channel Qon

Qon=Co.Vgs

Co is scaled by ß and Vgs is scaled by 1/ ß,hence Qo is scaled by ßx1/ß =1.

Channel resistance Ro

Ron = L/Wx1/Qoxμ, μ is mobility of charge carriers . Ro is scaled by1/ά/1/ άx1=1

Gate delay Td

Td is proportional to Ro and Cg

Td is scaled by 1x ß/$ά^2$ = ß/$ά^2$

Maximum operating frequency fo

fo=1/td,therefore it is scaled by 1/ ß/$ά^2$ = $ά^2$/ß

Saturation current

Idss= CoμW(Vgs-Vt)/2L, Co scale by ß and

voltages by 1/ ß, Idss is scaled by ß /$ß^2$= 1/ß

Current Density

J=Idss/A hence J is scaled by 1/ß/1/ά2 = $ά^2$ /ß

1.What is Scaling?

Proportional adjustment of the dimensions of an electronic device while maintaining the electrical properties of the device, results in a device either *larger* or *smaller* than the un-scaled device. Then *Which way do we scale the devices for VLSI? BIG and SLOW … or* ***SMALL*** *and* ***FAST****? What do we gain?*

2.Why Scaling?...

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

3.FoM for Scaling

Impact of scaling is characterized in terms of several indicators:

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

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

3.1 Technology Scaling

Goals of scaling the dimensions by 30%:

Reduce gate delay by 30% (increase operating frequency by 43%)

Double transistor density

Reduce energy per transition by 65% (50% power savings @ 43% increase in frequency)

Die size used to increase by 14% per generation

Technology generation spans 2-3 years

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

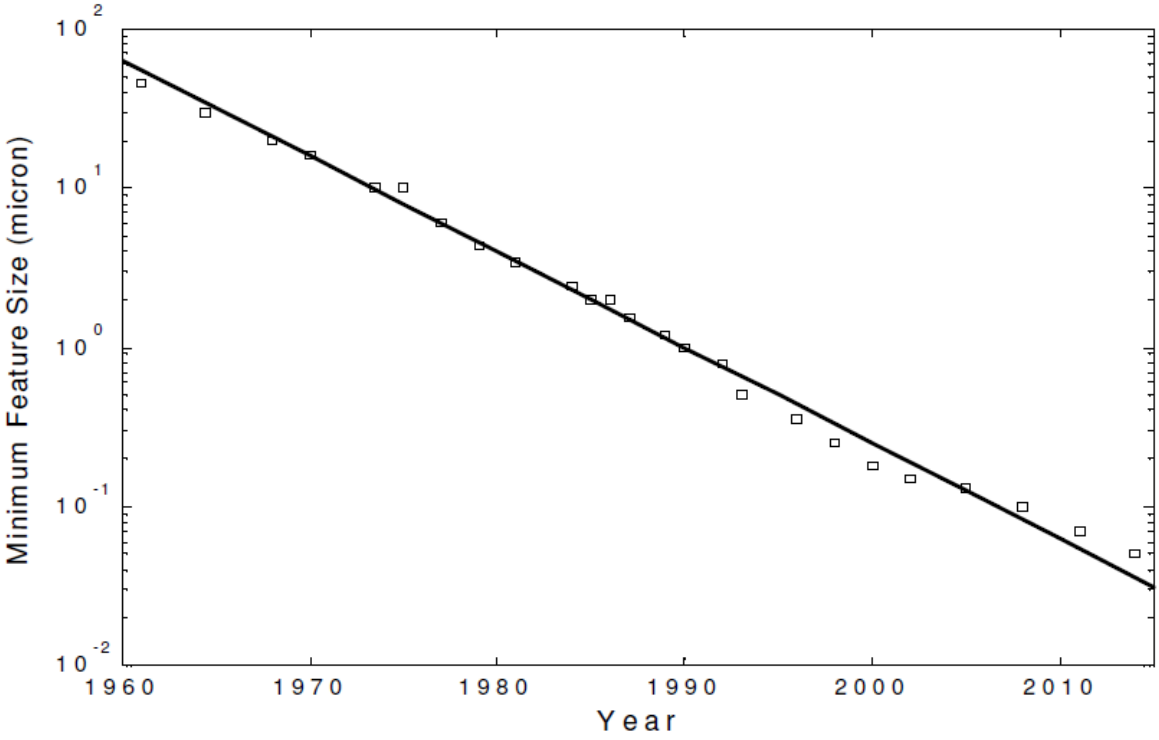


Figure-1:Technology Scaling (1)

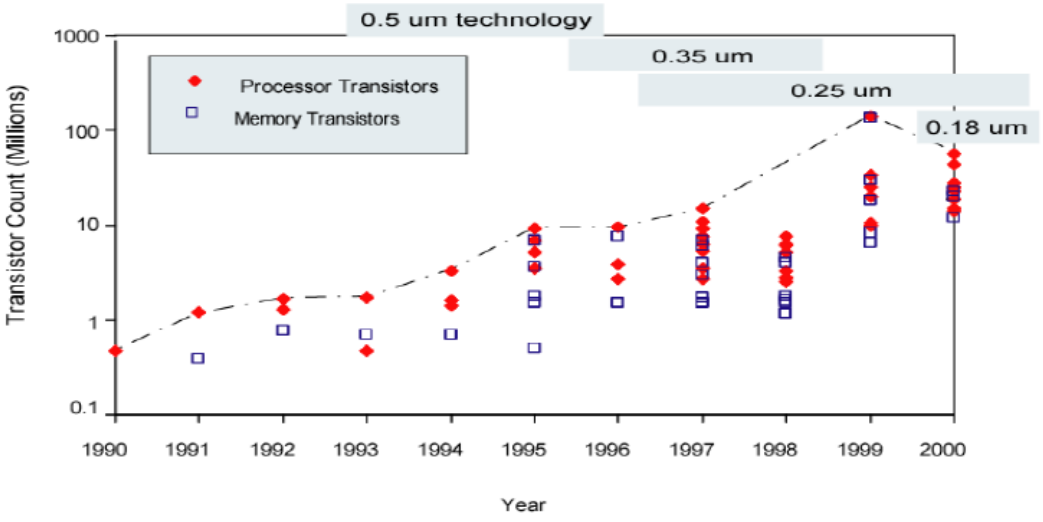


Figure-2:Technology Scaling (2)

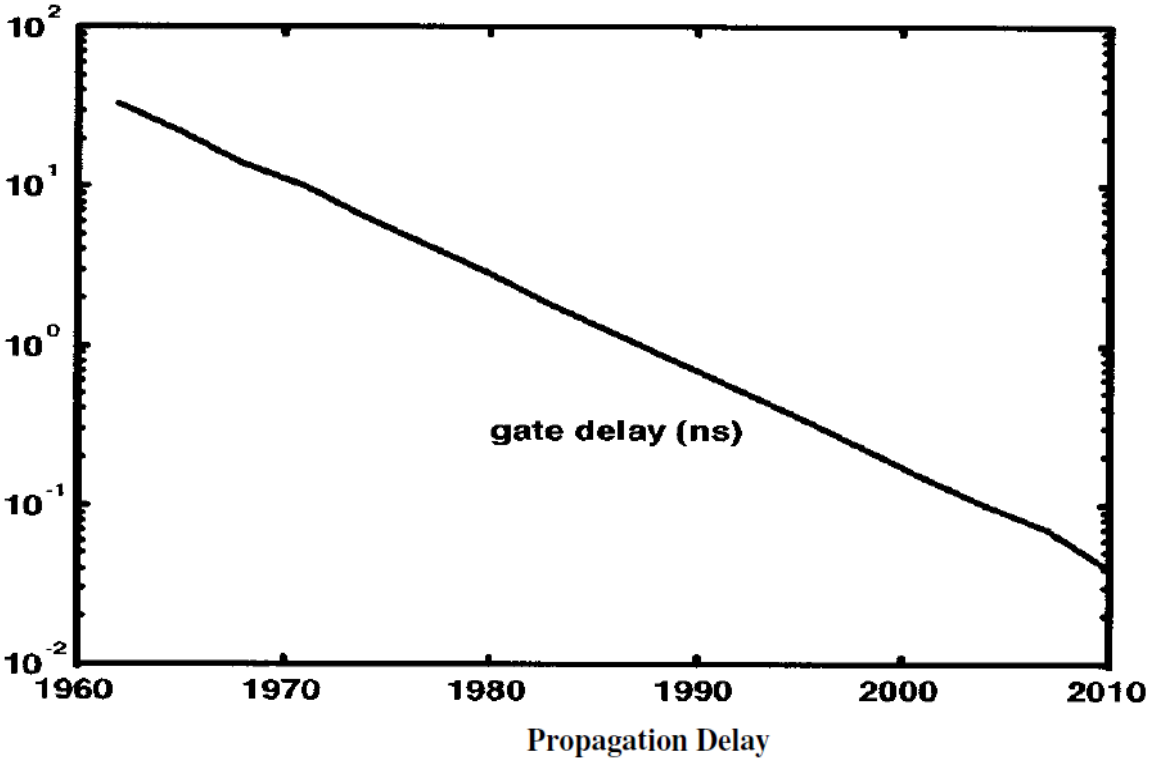
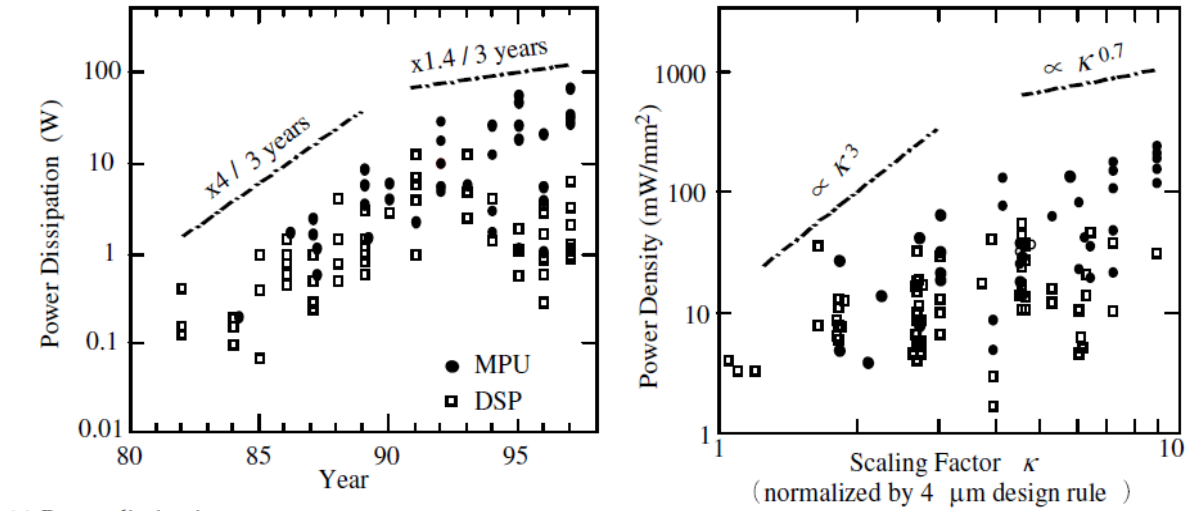


Figure-3:Technology Scaling (3)



(a) Power dissipation vs. year.

(b) Power density vs. scaling factor.

Figure-4:Technology Scaling (4)

Technology Generations

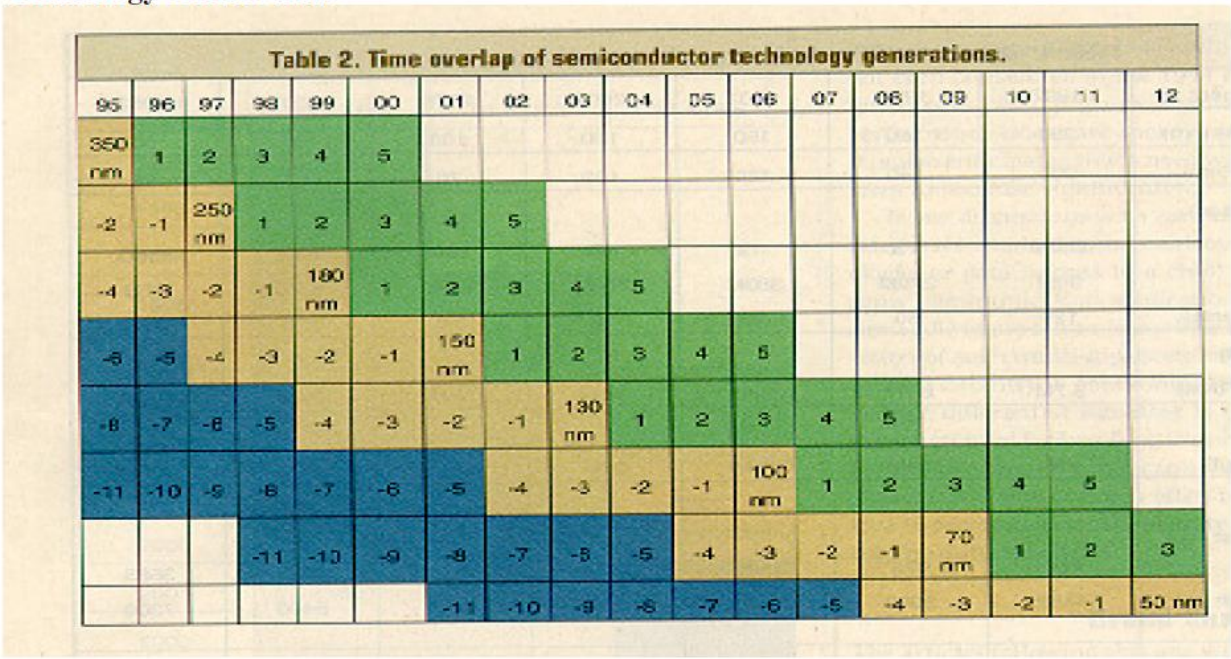


Table 2. Time overlap of semiconductor technology generations.

95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12
350 nm	1	2	3	4	5												
-2	-1	250 nm	1	2	3	4	5										
-4	-3	-2	-1	180 nm	1	2	3	4	5								
-6	-5	-4	-3	-2	-1	150 nm	1	2	3	4	5						
-8	-7	-6	-5	-4	-3	-2	-1	130 nm	1	2	3	4	5				
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	100 nm	1	2	3	4	5	
			-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	70 nm	1	2	3
						-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	50 nm

Figure-5:Technology generation

4. International Technology Roadmap for Semiconductors (ITRS)

Table 1 lists the parameters for various technologies as per ITRS.

Year of Introduction	1999	2000	2001	2004	2008	2011	2014
Technology node [nm]	180		130	90	60	40	30
Supply [V]	1.5-1.8	1.5-1.8	1.2-1.5	0.9-1.2	0.6-0.9	0.5-0.6	0.3-0.6
Wiring levels	6-7	6-7	7	8	9	9-10	10
Max frequency [GHz], Local-Global	1.2	1.6-1.4	2.1-1.6	3.5-2	7.1-2.5	11-3	14.9 -3.6
Max µP power [W]	90	106	130	160	171	177	186
Bat. power [W]	1.4	1.7	2.0	2.4	2.1	2.3	2.5

Node years: 2007/65nm, 2010/45nm, 2013/33nm, 2016/23nm

Table 1: ITRS

5.Scaling Models

- Full Scaling (Constant Electrical Field)

Ideal model – dimensions and voltage scale together by the same scale factor

- Fixed Voltage Scaling

Most common model until recently – only the dimensions scale, voltages remain constant

- General Scaling

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

6.Scaling Factors for Device Parameters

Device scaling modeled in terms of generic scaling factors:
1/α and 1/β

- 1/β: scaling factor for supply voltage V_{DD} and gate oxide thickness D
- 1/α: linear dimensions both horizontal and vertical dimensions

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

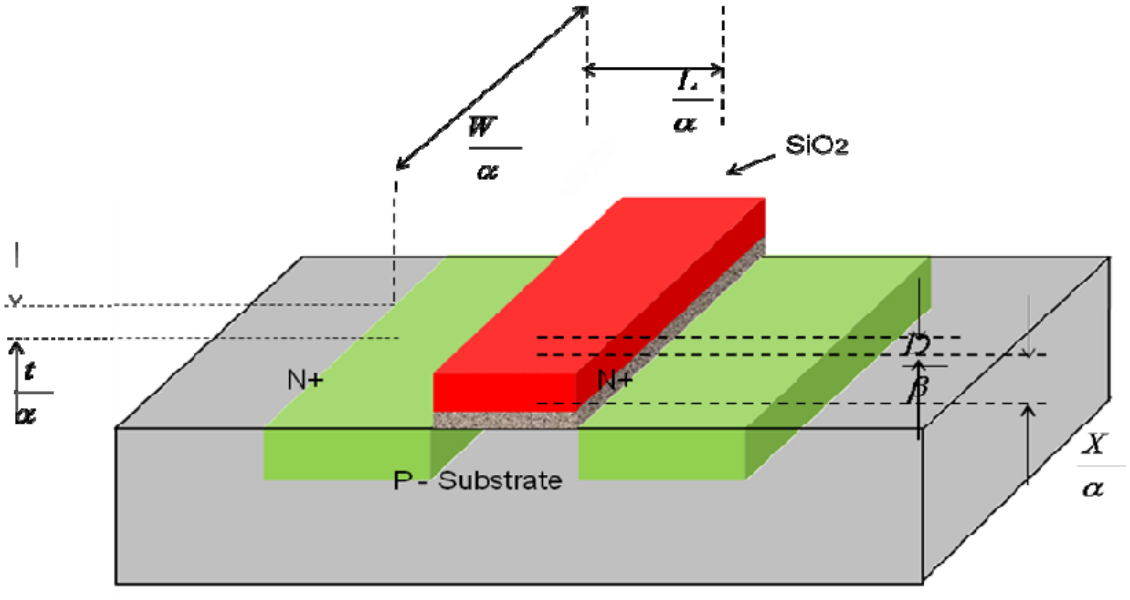


Figure-6:Technology generation

- Gate area A_g

$$A_g = L * W$$

Where L: Channel length and W: Channel width and both are scaled by $1/\alpha$
Thus A_g is scaled up by $1/\alpha^2$

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

$C_{ox} = \varepsilon_{ox}/D$
Where ε_{ox} is permittivity of gate oxide(thin-ox)= $\varepsilon_{ins}\varepsilon_o$ and D is the gate oxide thickness scaled by $1/\beta$
Thus C_{ox} is scaled up by $\dfrac{1}{\left(\frac{1}{\beta}\right)} = \beta$

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

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

- Parasitic capacitance C_x

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

A_x is the area of the depletion region around source or drain, scaled by $(1/\alpha^2)$.
Thus C_x is scaled up by $\{1/(1/\alpha)\} * (1/\alpha^2) = 1/\alpha$

- Carrier density in channel Q_{on}

$Q_{on} = C_o * V_{gs}$
where Q_{on} is the average charge per unit area in the 'on' state.
C_o is scaled by β and V_{gs} is scaled by $1/\beta$

Thus Q_{on} is scaled by 1

- Channel Resistance R_{on}

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

Where μ = channel carrier mobility and assumed constant

Thus R_{on} is scaled by 1.

- Gate delay T_d

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

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

- Maximum operating frequency f_o

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

f_o is inversely proportional to delay T_d and is scaled by

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

- Saturation current I_{dss}

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

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

- Current density J

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

So, J is scaled by

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

- Switching energy per gate E_g

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

So E_g is scaled by

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

- Power dissipation per gate P_g

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

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

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

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

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

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

- Power dissipation per unit area P_a

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

- Power – speed product P_T $P_T = P_g * T_d = \frac{1}{\beta^2}\left(\frac{\beta}{\alpha^2}\right) = \frac{1}{\alpha^2 \beta}$

6.1 Scaling Factors …Summary

Various device parameters for different scaling models are listed in Table 2 below.

Table 2: Device parameters for scaling models

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

Parameters	Description	General (Combined V and Dimension)	Constant E	Constant V
V_{DD}	Supply voltage	$1/\beta$	$1/\alpha$	1
L	Channel length	$1/\alpha$	$1/\alpha$	$1/\alpha$
W	Channel width	$1/\alpha$	$1/\alpha$	$1/\alpha$
D	Gate oxide thickness	$1/\beta$	$1/\alpha$	1
A_g	Gate area	$1/\alpha^2$	$1/\alpha^2$	$1/\alpha^2$
C_o (or C_{ox})	Gate capacitance per unit area	β	α	1
C_g	Gate capacitance	β/α^2	$1/\alpha$	$1/\alpha^2$
C_x	Parsitic capacitance	$1/\alpha$	$1/\alpha$	$1/\alpha$
Q_{on}	Carrier density	1	1	1
R_{on}	Channel resistance	1	1	1
I_{dss}	Saturation current	$1/\beta$	$1/\alpha$	1

Parameters	Description	General (Combined V and Dimension)	Constant E	Constant V
A_c	Conductor cross section area	$1/\alpha^2$	$1/\alpha^2$	$1/\alpha^2$
J	Current density	α^2 / β	α	α^2
V_g	Logic 1 level	$1 / \beta$	$1 / \alpha$	1
E_g	Switching energy	$1 / \alpha^2 \beta$	$1 / \alpha^3$	$1/\alpha^2$
P_g	Power dissipation per gate	$1 / \beta^2$	$1/\alpha^2$	1
N	Gates per unit area	α^2	α^2	α^2
P_a	Power dissipation per unit area	α^2 / β^2	1	α^2
T_d	Gate delay	β / α^2	$1 / \alpha$	$1/\alpha^2$
f_o	Max. operating frequency	α^2 / β	α	α^2
P_T	Power speed product	$1 / \alpha^2 \beta$	$1 / \alpha^3$	$1/\alpha^2$

7.Implications of Scaling

- Improved Performance
- Improved Cost
- Interconnect Woes
- Power Woes
- Productivity Challenges
- Physical Limits

7.1Cost Improvement

– Moore's Law is still going strong as illustrated in Figure 7.

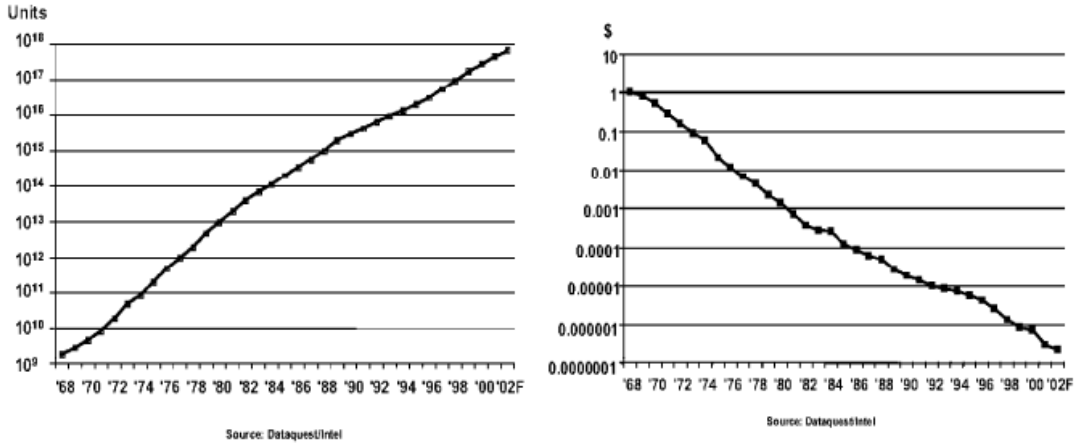


Figure-7:Technology generation

7.2:Interconnect Woes

- Scaled transistors are steadily improving in delay, but scaled wires are holding constant or getting worse.
- SIA made a gloomy forecast in 1997
  - Delay would reach minimum at 250 – 180 nm, then get worse because of wires
- But…
- For short wires, such as those inside a logic gate, the wire RC delay is negligible.
- However, the long wires present a considerable challenge.

- Scaled transistors are steadily improving in delay, but scaled wires are holding constant or getting worse.
- SIA made a gloomy forecast in 1997
  - Delay would reach minimum at 250 – 180 nm, then get worse because of wires
- But…
- For short wires, such as those inside a logic gate, the wire RC delay is negligible.
- However, the long wires present a considerable challenge.

Figure 8 illustrates delay Vs. generation in nm for different materials.

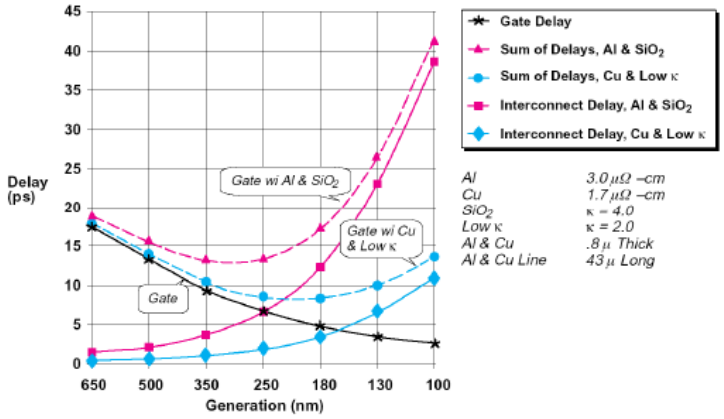


Figure-8:Technology generation

7.3 Reachable Radius

- We can't send a signal across a large fast chip in one cycle anymore
- But the microarchitect can plan around this as shown in Figure 9.
  - Just as off-chip memory latencies were tolerated

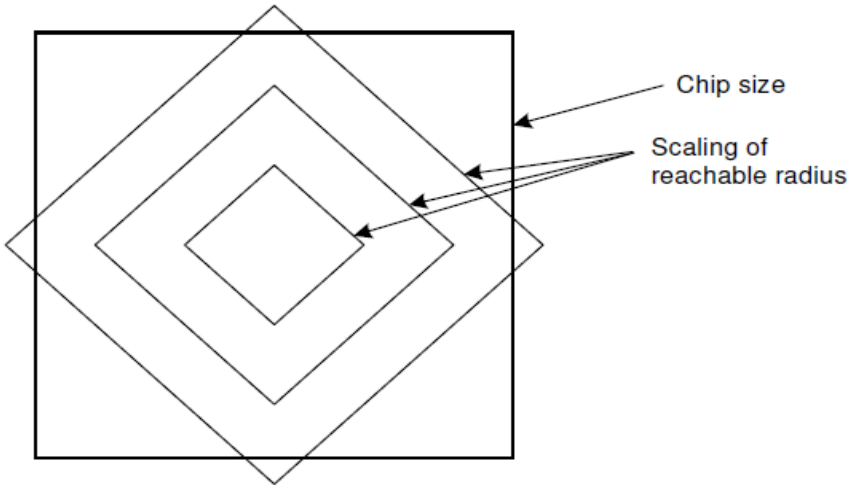


Figure-9:Technology generation

7.4 Dynamic Power

- Intel VP Patrick Gelsinger (ISSCC 2001)
  - If scaling continues at present pace, by 2005, high speed processors would have power density of nuclear reactor, by 2010, a rocket nozzle, and by 2015, surface of sun.
  - "Business as usual will not work in the future."
- Attention to power is increasing(Figure 10)

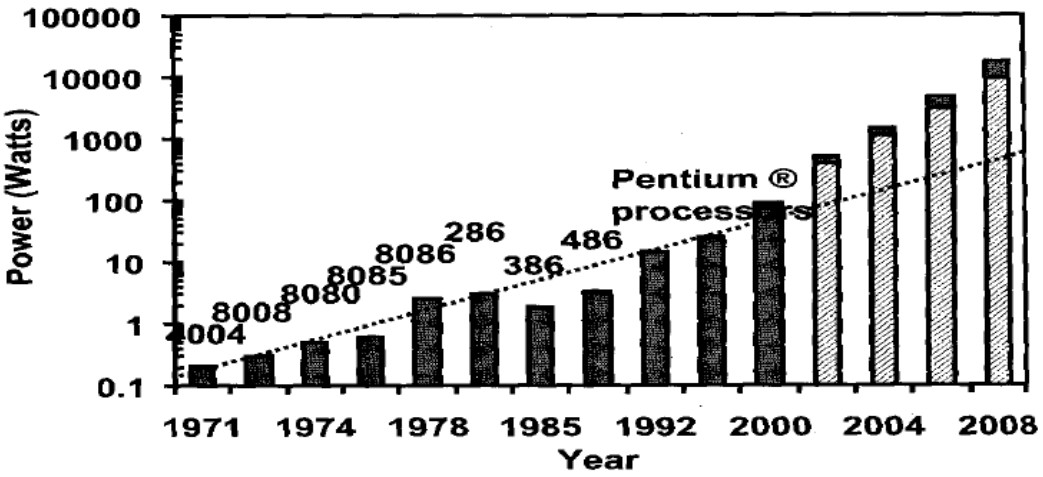


Figure-10:Technology generation

7.5 Static Power

- V_{DD} decreases
  - Save dynamic power
  - Protect thin gate oxides and short channels
  - No point in high value because of velocity saturation.
- V_t must decrease to maintain device performance
- But this causes exponential increase in OFF leakage

A Major future challenge(Figure 11)

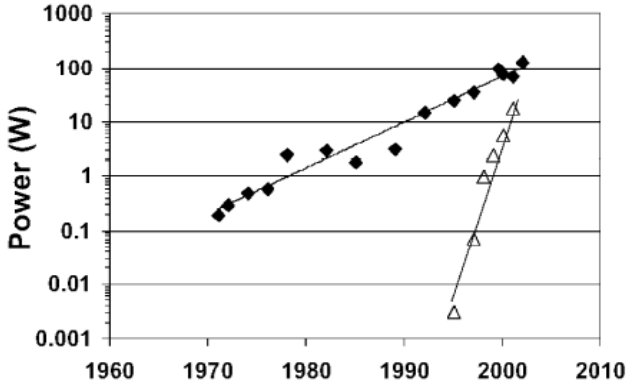


Figure-11:Technology generation

7.6 Productivity

- Transistor count is increasing faster than designer productivity (gates / week)
  - Bigger design teams
    - Up to 500 for a high-end microprocessor
  - More expensive design cost
  - Pressure to raise productivity
    - Rely on synthesis, IP blocks
  - Need for good engineering managers

7.7 Physical Limits

- Will Moore's Law run out of steam?
  - Can't build transistors smaller than an atom…
- Many reasons have been predicted for end of scaling
  - Dynamic power
  - Sub-threshold leakage, tunneling
  - Short channel effects
  - Fabrication costs
  - Electro-migration
  - Interconnect delay
- Rumors of demise have been exaggerated

8. Limitations of Scaling

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

- Substrate doping
- Depletion width
- Limits of miniaturization

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

8.1 Substrate doping

- Substrate doping
- Built-in(junction) potential V_B depends on substrate doping level – can be neglected as long as V_B is small compared to $V_{DD.}$
- As length of a MOS transistor is reduced, the depletion region width –scaled down to prevent source and drain depletion region from meeting.
- the depletion region width d for the junctions is $d = \sqrt{\frac{2}{q}\frac{\xi_{si}}{N_B}\frac{\xi_0 V}{1}}$
- ε_{si} relative permittivity of silicon
- ε_0 permittivity of free space($8.85*10^{-14}$ F/cm)
- V effective voltage across the junction $V_a + V_b$
- q electron charge
- N_B doping level of substrate
- V_a maximum value Vdd-applied voltage
- V_b built in potential and $V_B = \frac{KT}{q}\ln\left[\frac{N_B}{n_i}\frac{N_D}{n_i}\right]$

8.2 Depletion width

- N_B is increased to reduce d , but this increases threshold voltage V_t -against trends for scaling down.
- Maximum value of N_B ($1.3*10^{19}$ cm^{-3} , at higher values, maximum electric field applied to gate is insufficient and no channel is formed.
- N_B maintained at satisfactory level in the channel region to reduce the above problem.
- E_{max} maximum electric field induced in the junction. $E_{max} = \frac{2V}{d}$

If N_B is inreased by α Va =0 Vb increased by ln α and d is decreased by

- Electric field across the depletion region is increased by

  1/ $\sqrt{\frac{\ln \alpha}{\alpha}}$

- Reach a critical level E_{crit} with increasing N_B $\quad d = \sqrt{\frac{2}{q}\frac{\xi_{si}}{N_B}\xi\left\{\frac{E_{crit}.d}{2}\right\}}$

Where $\quad d = \frac{\xi_{si}}{q}\frac{\xi_0}{N_B}(E_{crit})$

Figure 12 , Figure 13 and Figure 14 shows the relation between substrate concentration Vs depletion width , Electric field and transit time.
Figure 15 demonstrates the interconnect length Vs. propagation delay and Figure 16 oxide thickness Vs. thermal noise.

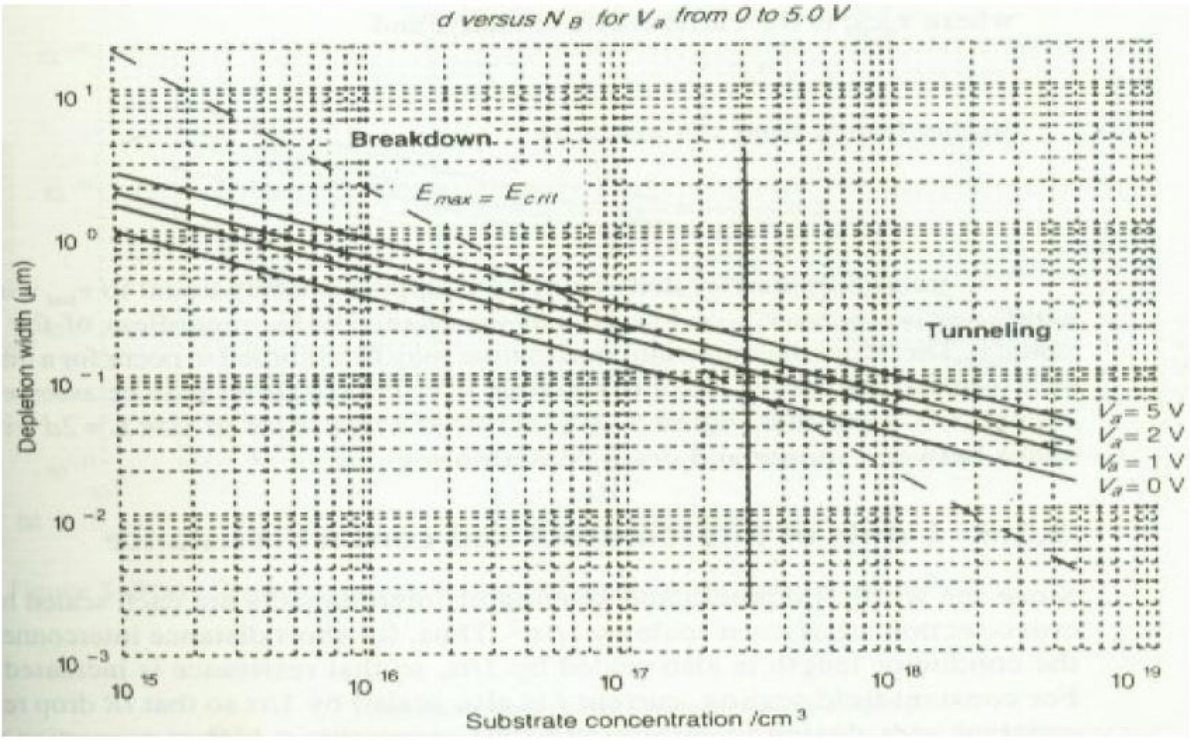


Figure-12:Technology generation

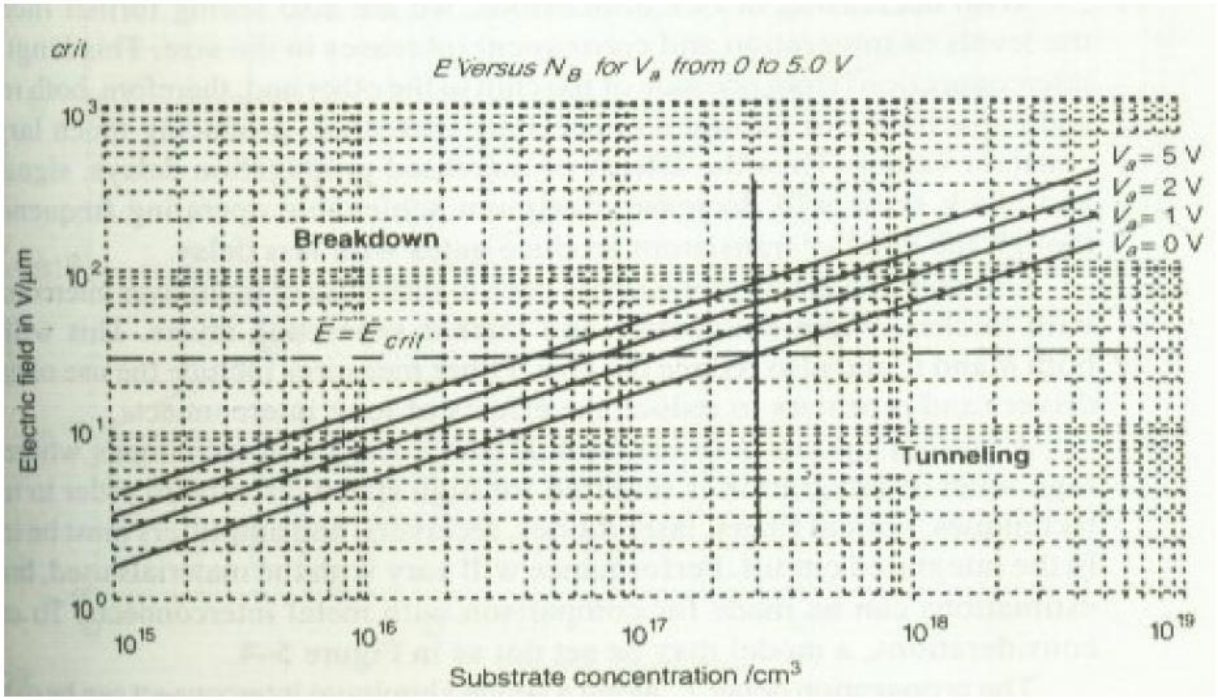


Figure-13:Technology generation

8.3 Limits of miniaturization

- minimum size of transistor; process tech and physics of the device
- Reduction of geometry; alignment accuracy and resolution
- Size of transistor measured in terms of channel length L

  L=2d (to prevent push through)
- L determined by N_B and Vdd
- Minimum transit time for an electron to travel from source to drain is

$$v_{drift} = \mu E$$

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

- ₛmaximum carrier drift velocity is approx. Vsat,regardless of supply voltage

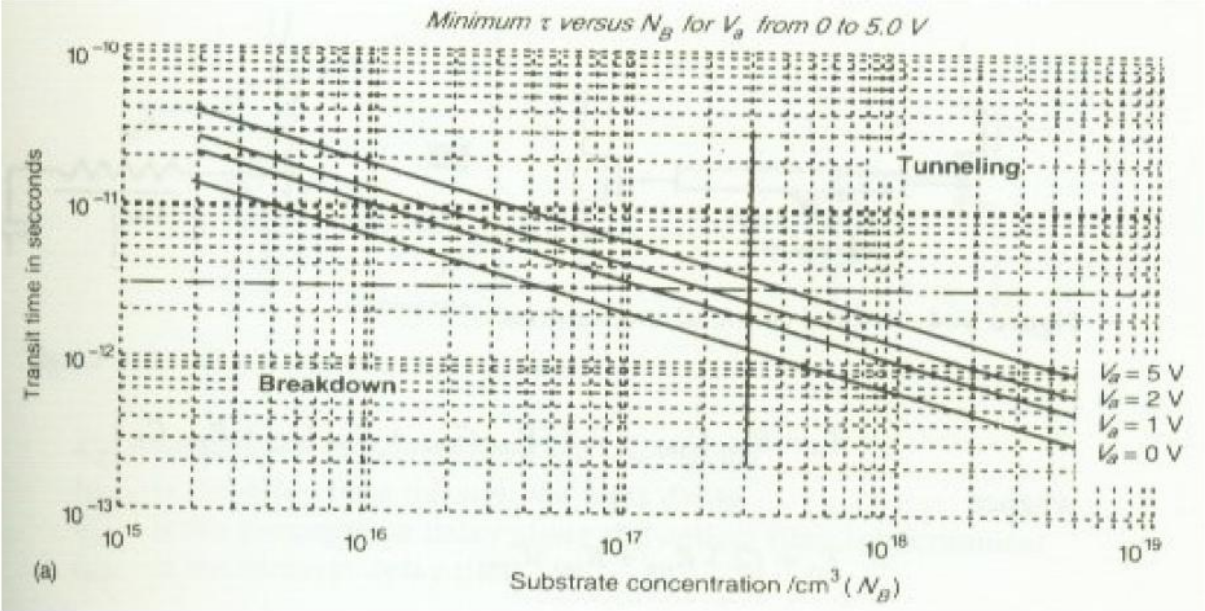


Figure-14:Technology generation

8.4 Limits of interconnect and contact resistance

- Short distance interconnect- conductor length is scaled by 1/α and resistance is increased by α
- For constant field scaling, I is scaled by 1/ α so that IR drop remains constant as a result of scaling.-driving capability/noise margin.

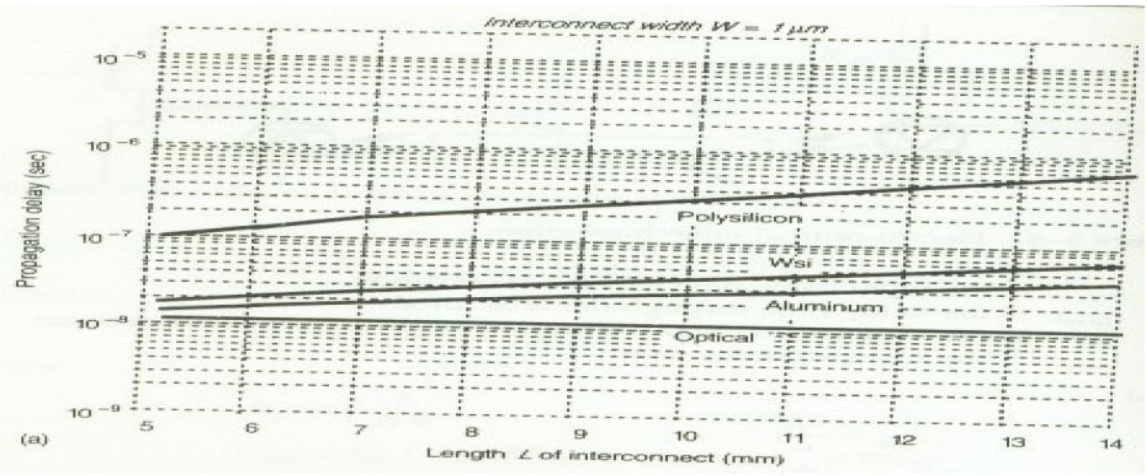


Figure-15:Technology generation

8.5 Limits due to subthreshold currents

- Major concern in scaling devices.
- I_{sub} is directly praportinal exp (Vgs – Vt) q/KT
- As voltages are scaled down, ratio of Vgs-Vt to KT will reduce-so that threshold current increases.
- Therefore scaling Vgs and Vt together with Vdd .
- Maximum electric field across a depletion region is

$$E_{\max} = 2\{V_a + V_b\}/d$$

8.6 Limits on supply voltage due to noise

Decreased inter-feature spacing and greater switching speed –result in noise problems

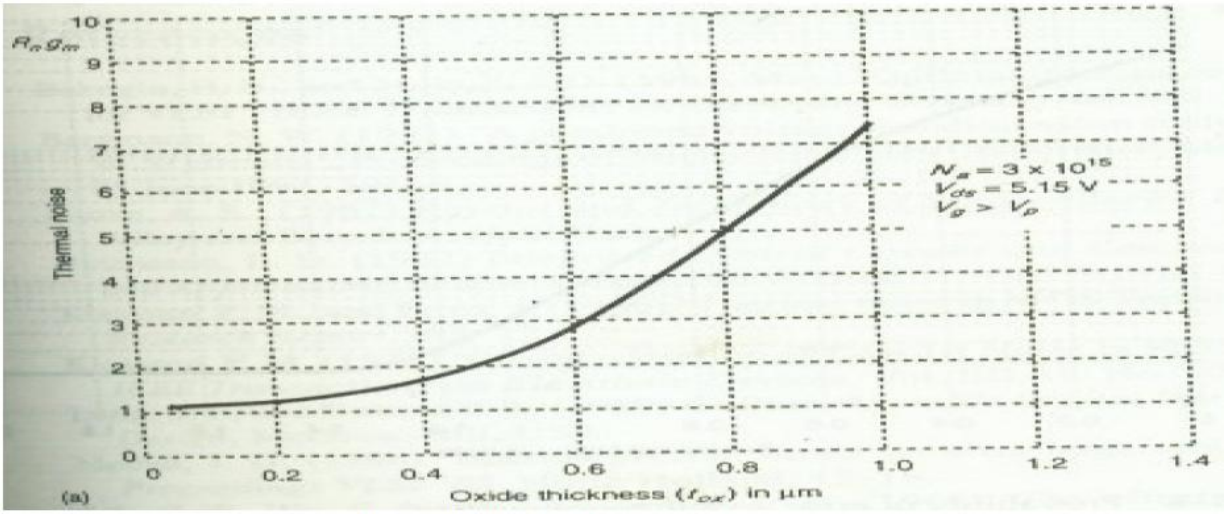


Figure-16:Technology generation

9. Observations – Device scaling

- Gate capacitance per micron is nearly independent of process
- But ON resistance * micron improves with process
- Gates get faster with scaling (good)
- Dynamic power goes down with scaling (good)
- Current density goes up with scaling (bad)
- Velocity saturation makes lateral scaling unsustainable

9.1 Observations – Interconnect scaling

- Capacitance per micron is remaining constant
  - About 0.2 fF/mm
  - Roughly 1/10 of gate capacitance
- Local wires are getting faster
  - Not quite tracking transistor improvement
  - But not a major problem
- Global wires are getting slower
  - No longer possible to cross chip in one cycle

*10. Sum*mary

- Scaling allows people to build more complex machines

  – That run faster too
- It does not to first order change the difficulty of module design

  – Module wires will get worse, but only slowly

  – You don't think to rethink your wires in your adder, memory

  Or even your super-scalar processor core
- It does let you design more modules
- Continued scaling of uniprocessor performance is getting hard

  -Machines using global resources run into wire limitations

  -Machines will have to become more explicitly parallel

Recommended questions:

1. Explain sheet resistance with neat diagram.
2. Write a note on area capacitance.
3. With neat diagram explain delay unit.
4. Explain propagation delay and wiring capacitance.
5. Explain scaling models and factors for MOS transistors.
6. What are the limits due to current density and noise.

Part-B

Unit-5

CMOS subsystem design

Architectural issues, switch logic, gate logic, design examples-combinational logic, clocked circuits. Other system considerations.

Clocking strategies

Recommended readings:

1. Douglas A. Pucknell & Kamran Eshraghian, **"Basic VLSI Design"** PHI 3rd Edition (original Edition – 1994), 2005.

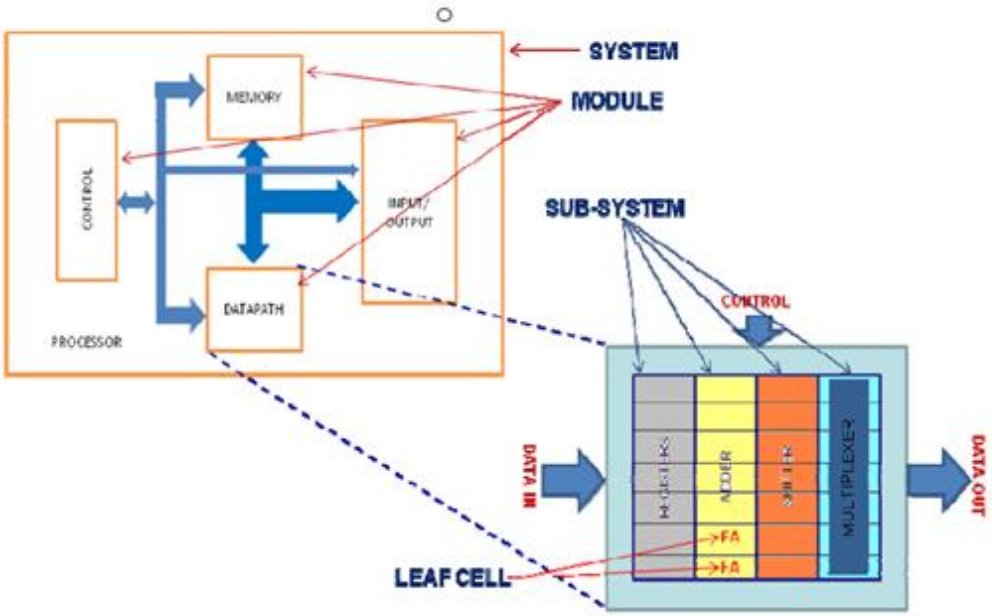
2. Neil H. E. Weste and K. Eshragian**," Principles of CMOS VLSI Design: A System Perspective,"** 2nd edition, Pearson Education (Asia) Pvt. Ltd., 2000. History of VLSI.

5.1.What is a *System*?

A *system* is a set of interacting or interdependent entities forming and integrate whole.
Common characteristics of a system are

- Systems have *structure* - defined by parts and their composition
- Systems have *behavior* – involves inputs, processing and outputs (of material, information or energy)
- Systems have *interconnectivity* the various parts of the system functional as well as structural relationships between each other

1.1Decomposition of a *System*: A Processor



VLSI Design Flow

- The electronics industry has achieved a phenomenal growth –mainly due to the rapid advances in integration technologies, large scale systems design-in short due to VLSI.
- Number applications of integrated circuits in high-performance computing, telecommunications, and consumer electronics has been rising steadily.
- Current leading-edge technology trend –expected to continue with very important implications on VLSI and systems design.
- The design process, at various levels, is evolutionary in nature.
- Y-Chart (first introduced by D. Gajski) as shown in Figure1 illustrates the design flow for mast logic chips, using design activities.
- Three different axes (domains) which resemble the letter Y.
- Three major domains, namely
  Behavioral domain
  Structural domain

Geometrical domain

- Design flow starts from the algorithm that describes the behavior of target chip.

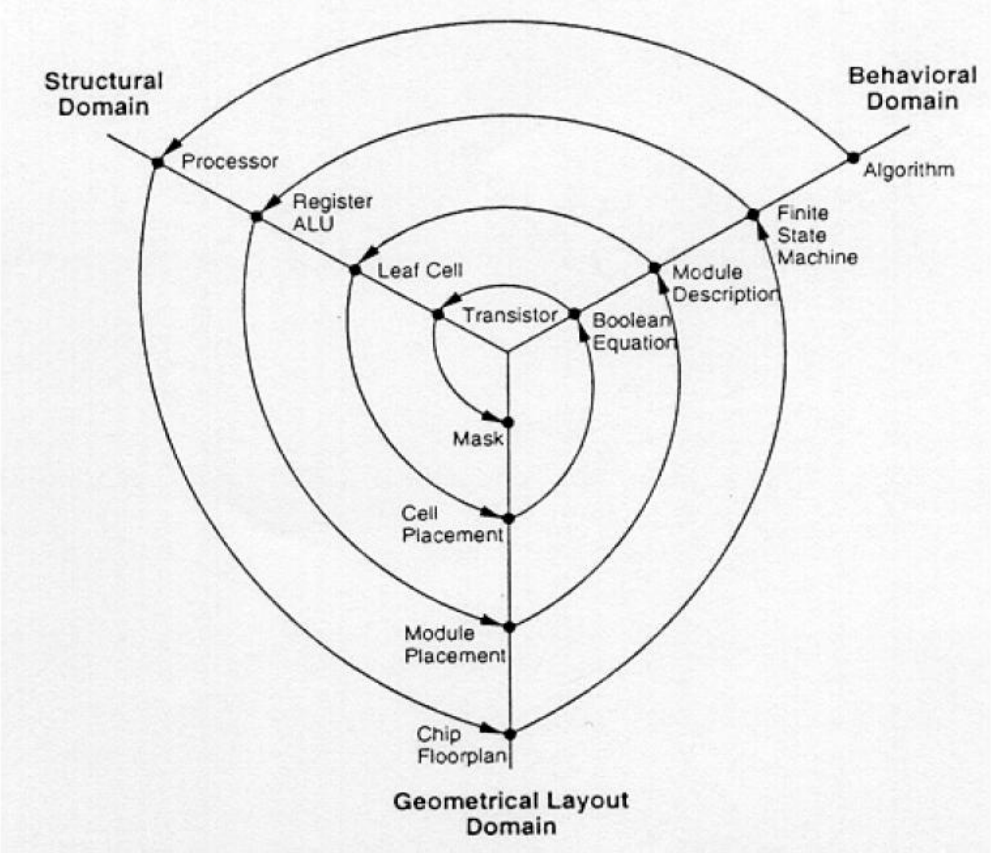


Figure 1. Typical VLSI design flow in three domains(Y-chart)

VLSI design flow, taking in to account the various representations, or abstractions of design are

Behavioural,logic,circuit and mask layout.

Verification of design plays very important role in every step during process.
Two approaches for design flow as shown in Figure 2 are

Top-down
Bottom-up

Top-down design flow- excellent design process control
In reality, both top-down and bottom-up approaches have to be combined.
Figure 3 explains the typical full custom design flow.

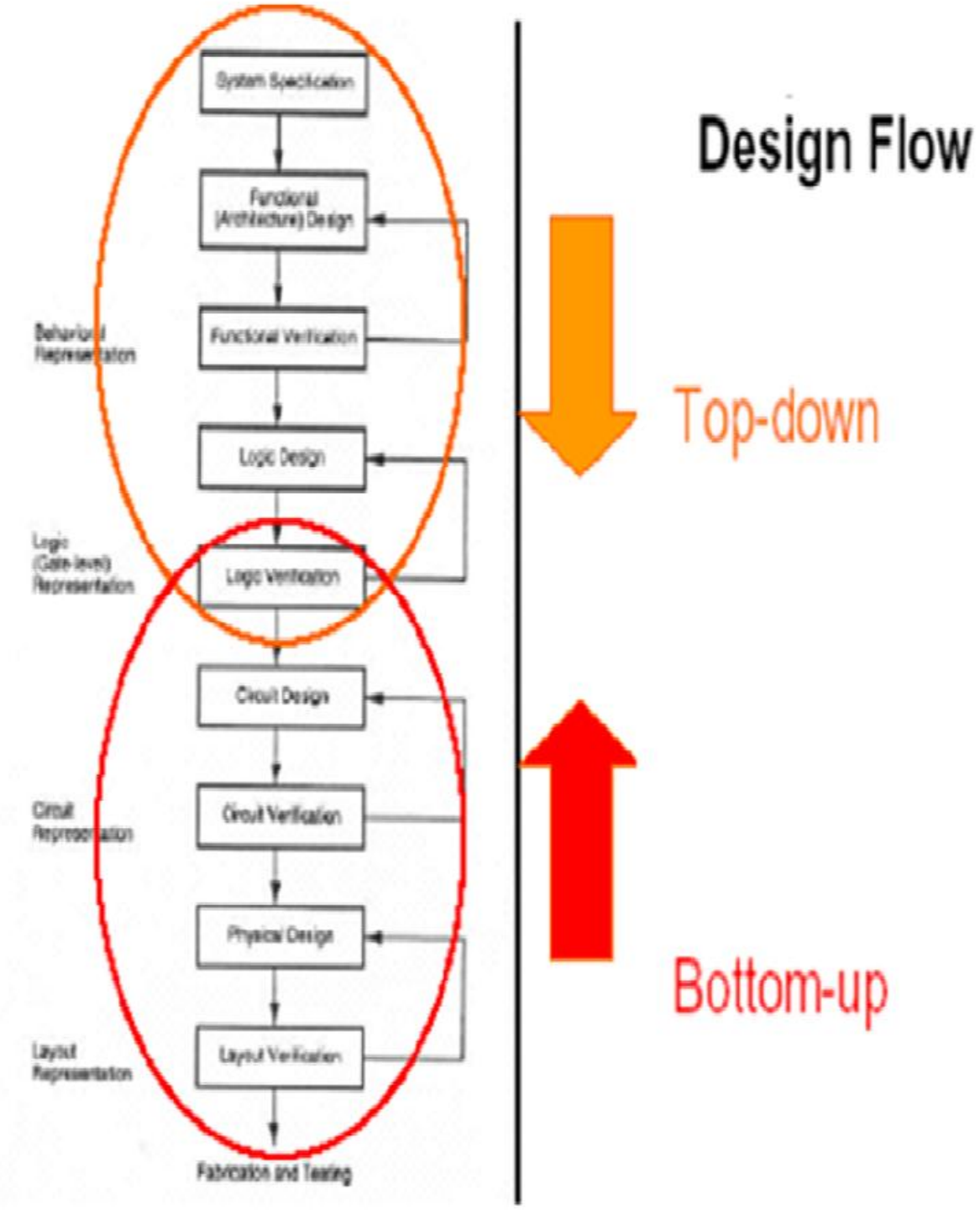


Figure 2. Typical VLSI design flow

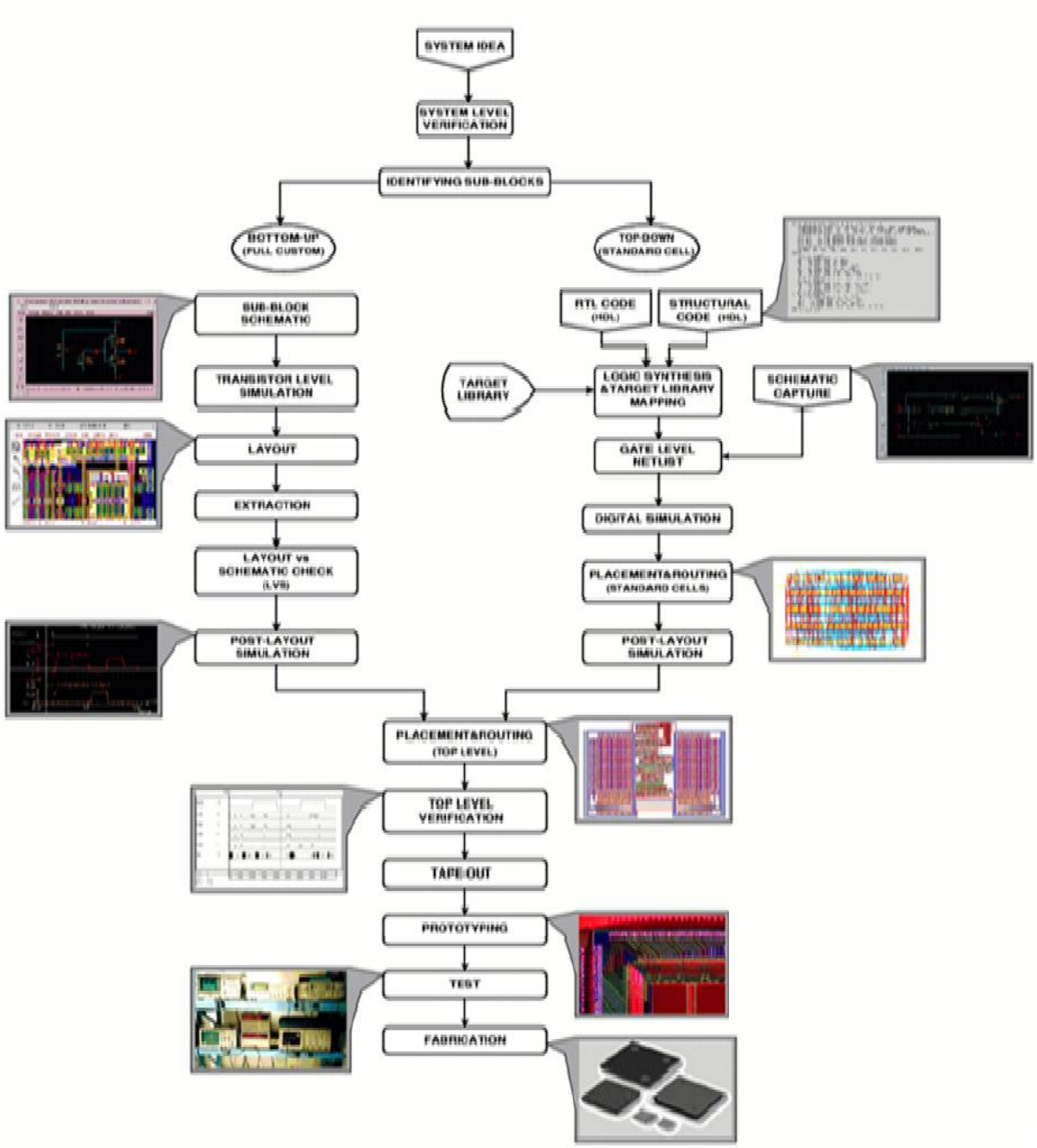


Figure 3. Typical ASIC/Custom design flow

5.2 Structured Design Approach

- Design methodologies and structured approaches developed with complex hardware and software.
- Regardless of the actual size of the project, basic principles of structured design-improve the prospects of success.
- Classical techniques for reducing the complexity of IC design are:

  Hierarchy
  Regularity
  Modularity
  Locality

Hierarchy: "Divide and conquer" technique involves dividing a module into sub-modules and then repeating this operation on the sub-modules until the complexity of the smaller parts becomes manageable.

Regularity: The hierarchical decomposition of a large system should result in not only **simple**, but also **similar** blocks, as much as possible. Regularity usually reduces the number of different modules that need to be designed and verified, at all levels of abstraction.

Modularity: The various functional blocks which make up the larger system must have **well-defined functions** and **interfaces**.

Locality: Internal details remain at the local level. The concept of locality also ensures that connections are mostly between neighboring modules, **avoiding long-distance connections** as much as possible.

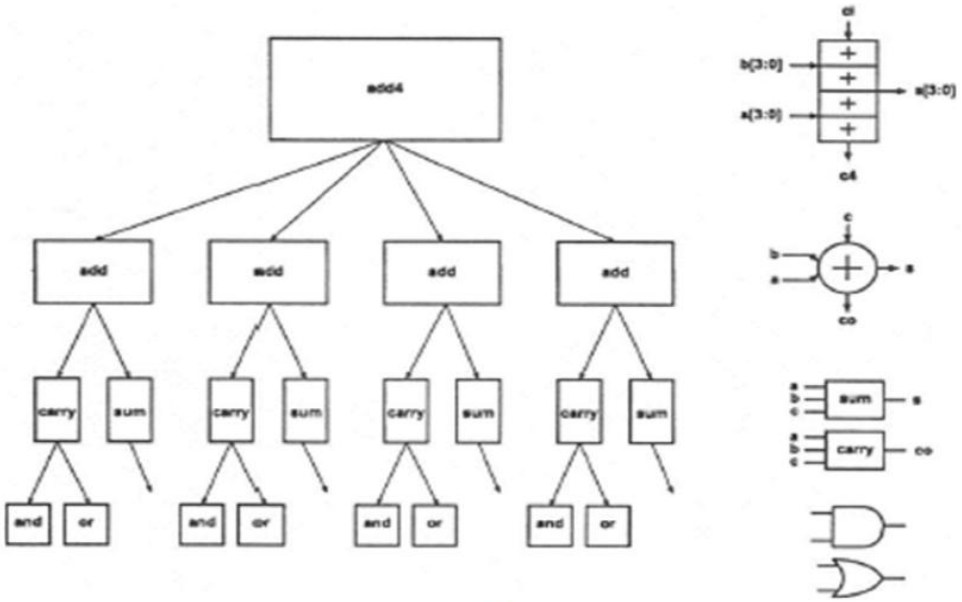


Figure 4-Structured Design Approach –Hierarchy

5.3 Regularity

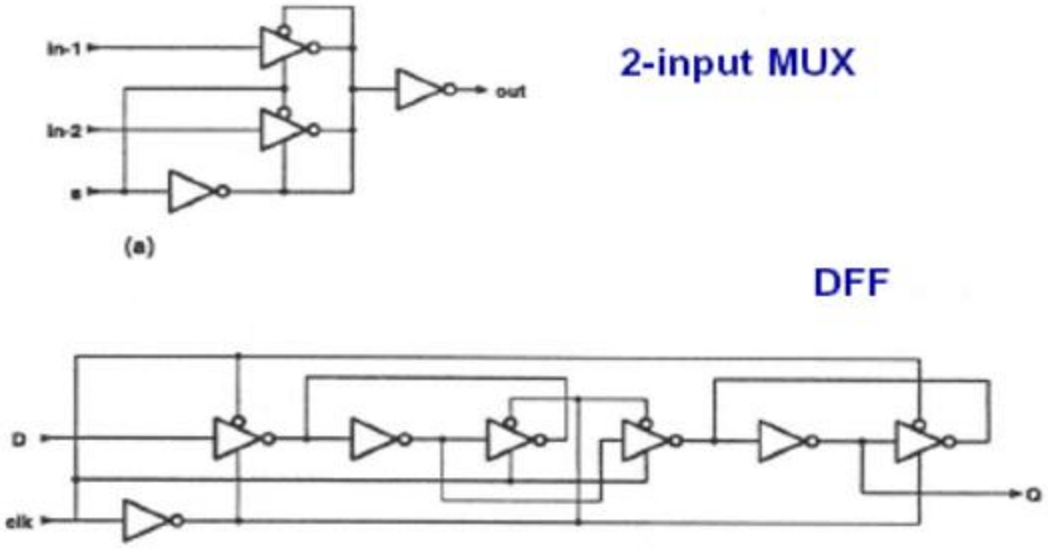


Figure5-.Structured Design Approach –Regularity

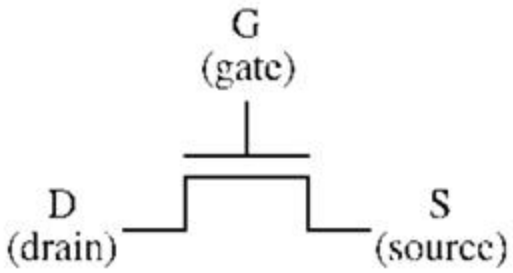
- Design of array structures consisting of identical cells.-such as parallel multiplication array.
- Exist at all levels of abstraction:
  transistor level-uniformly sized.
  logic level- identical gate structures
- 2:1 MUX, D-F/F- inverters and tri state buffers
- Library-well defined and well-characterized basic building block.
- Modularity: enables parallelization and allows plug-and-play
- Locality: Internals of each module unimportant to exterior modules and internal details remain at local level.

Figure 4 and Figure 5 illustrates these design approaches with an example.

5.4 Architectural issues

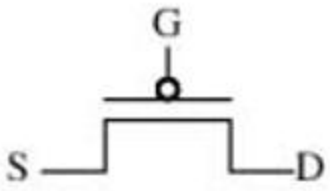
- Design time increases exponentially with increased complexity
- Define the requirements
- Partition the overall architecture into subsystems.
- Consider the communication paths
- Draw the floor plan
- Aim for regularity and modularity
- convert each cell into layout
- Carry out DRC check and simulate the performance

5.5 MOSFET as a Switch



*n*MOS transistor:
Closed (conducting) when
Gate = 1 (Vdd, 5V)

Open (non-conducting) when
Gate = 0 (ground, 0V)

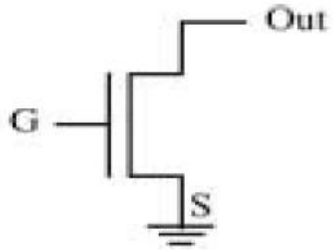


*p*MOS transistor:
Closed (conducting) when
Gate = 0 (ground, 0V)

Open (non-conducting) when
Gate = 1 (Vdd, 5V)

- We can view MOS transistors as electrically controlled switches
- Voltage at gate controls path from source to drain

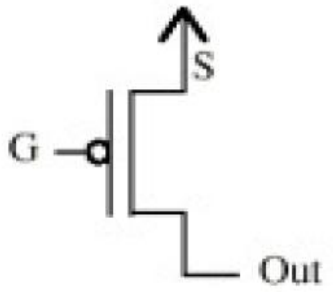
For *n*MOS switch, source is typically tied to ground and is used to *pull-down* signals:



when Gate = 1, Out = 0, (0V)

when Gate = 0, Out = Z (high impedance)

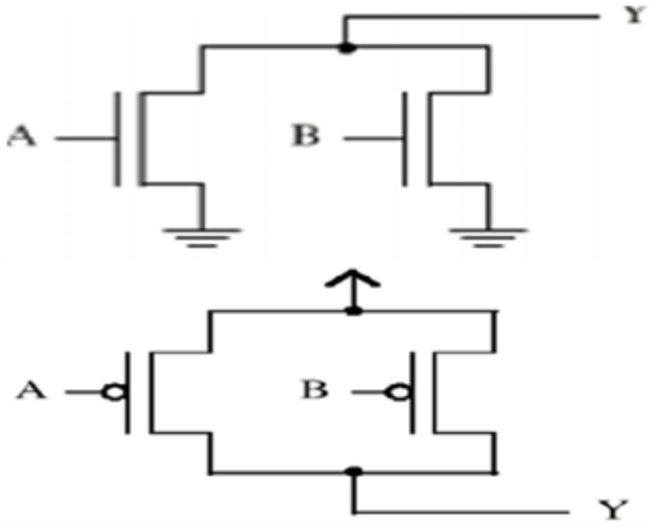
For *p*MOS switch, source is typically tied to Vdd, used to *pull* signals *up*:



when Gate = 0, Out = 1 (Vdd)

when Gate = 1, Out = Z (high impedance)

5.5.1 Parallel connection of Switches..



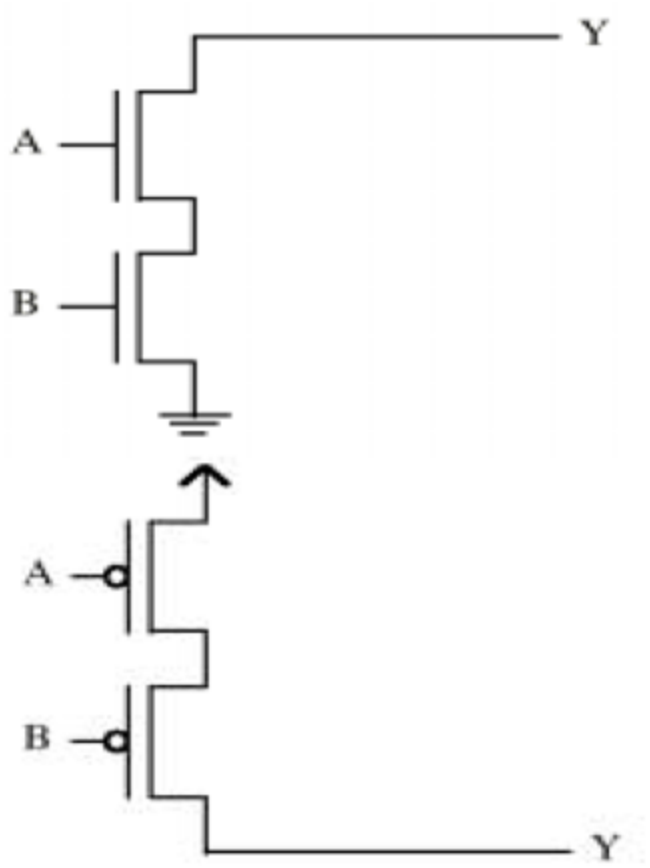
$Y = 0$, if A or $B = 1$

$A + B$

$Y = 1$ if A or $B = 0$

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

5.5.2 Series connection of Switches..



$Y = 0$, if A and $B = 1$

$A \cdot B$

$Y = 1$, if A and $B = 0$

$\overline{A} \cdot \overline{B}$

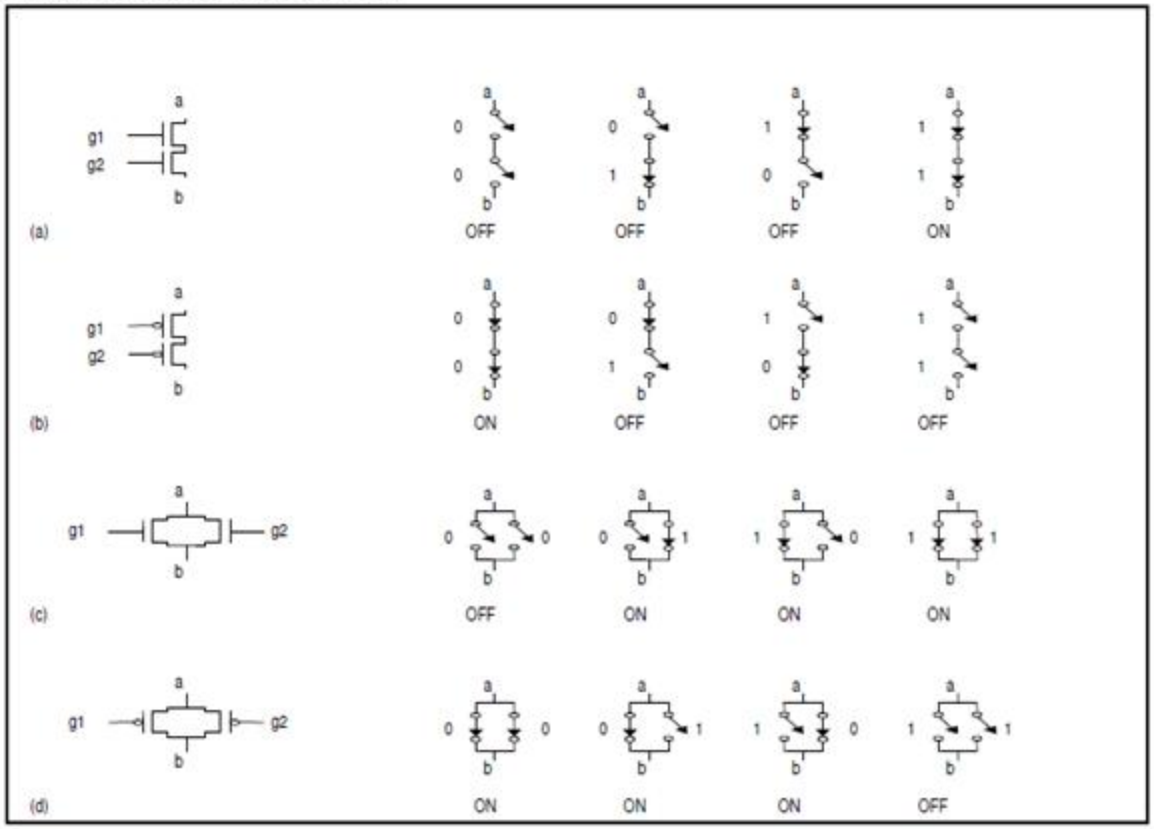
5.5.3 Series and parallel connection of Switches..

nMOS: 1 = ON

pMOS: 0 = ON

Series: both must be ON

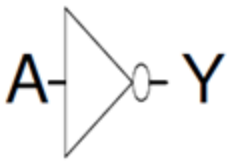
Parallel: either can be ON

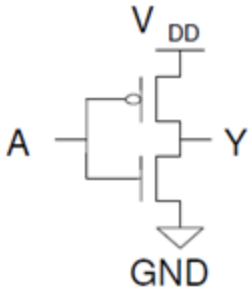


5.6 Circuit Families : Restoring logic

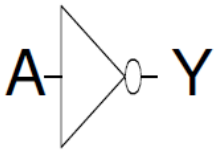
CMOS INVERTER

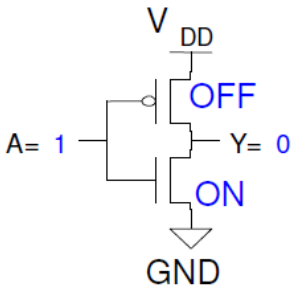
A	Y
0	
1	



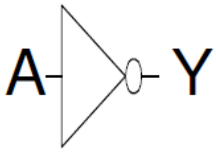


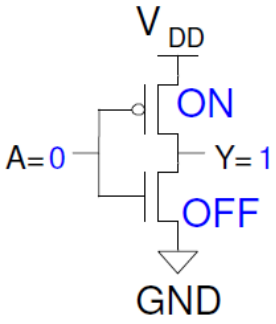
A	Y
0	
1	**0**





A	Y
0	**1**
1	0





5.6.1 NAND gate Design..

NAND Gate Design

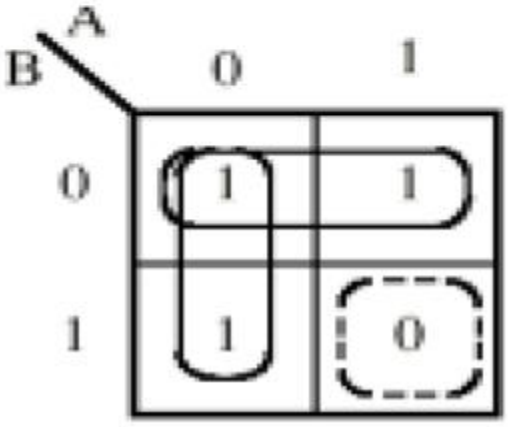
p-type transistor tree will provide "1" values of logic function

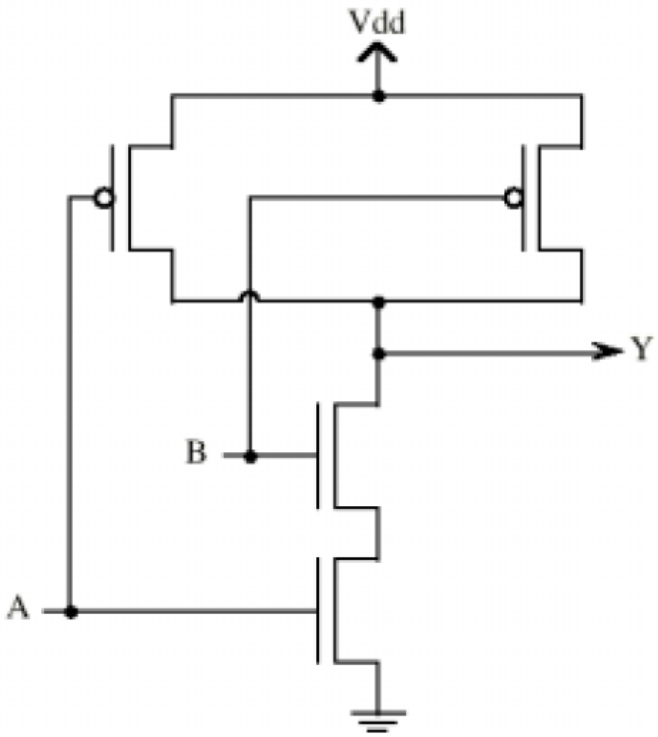
n-type transistor tree will provide "0" values of logic function

Truth Table (NAND):

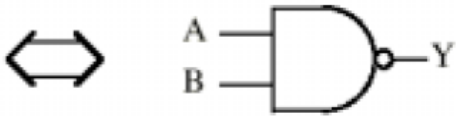
AB	
00	1
01	1
10	1
11	0

K-map (NAND):

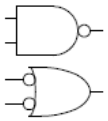


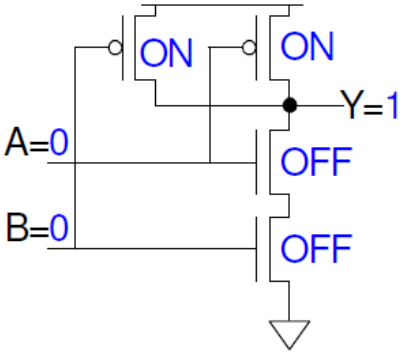


$$P_{tree} = \overline{A} + \overline{B}$$
$$N_{tree} = A \cdot B$$

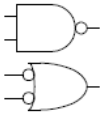


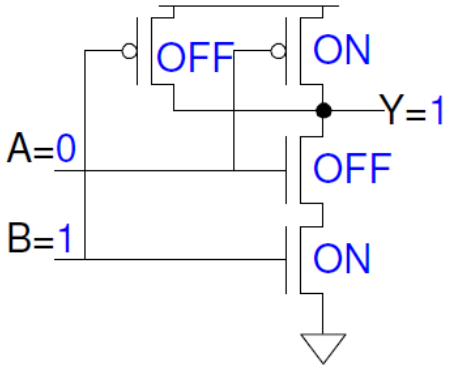
A	B	Y
0	0	1
0	1	
1	0	
1	1	



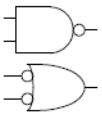


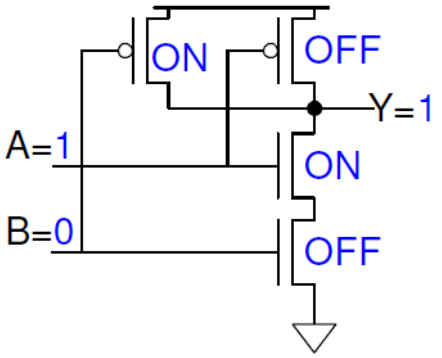
A	B	Y
0	0	1
0	1	1
1	0	
1	1	

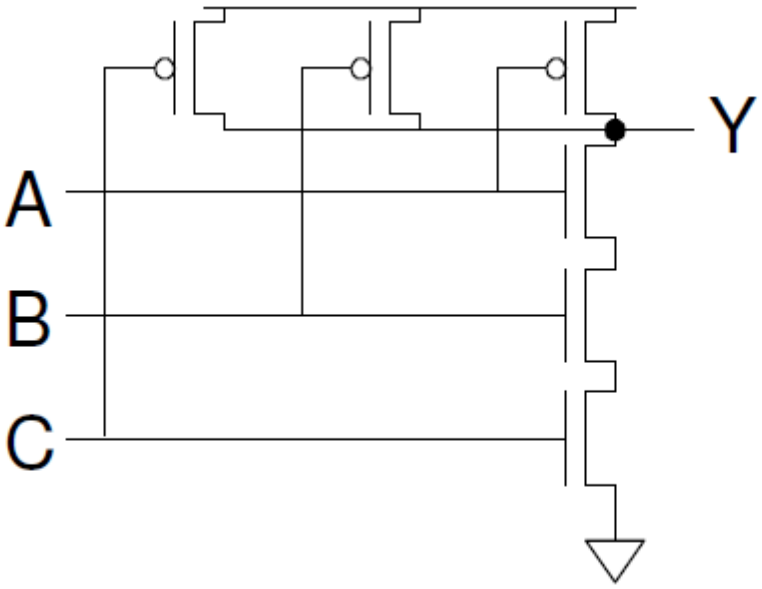




A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0







5.6.2 NOR gate Design..

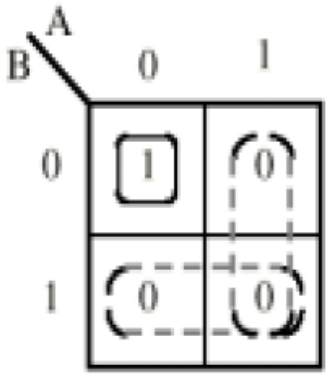
NOR Gate Design

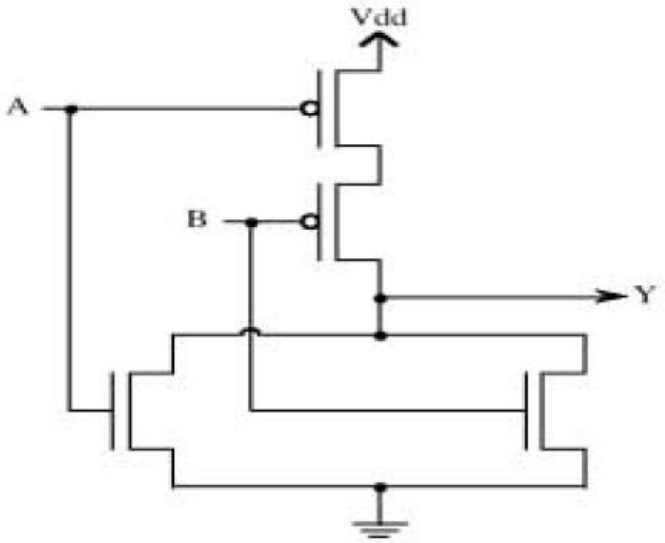
p-type transistor tree will provide "1" values of logic function

n-type transistor tree will provide "0" values of logic function

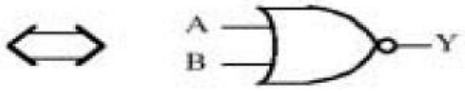
Truth Table: K-map:

AB	
00	1
01	0
10	0
11	0

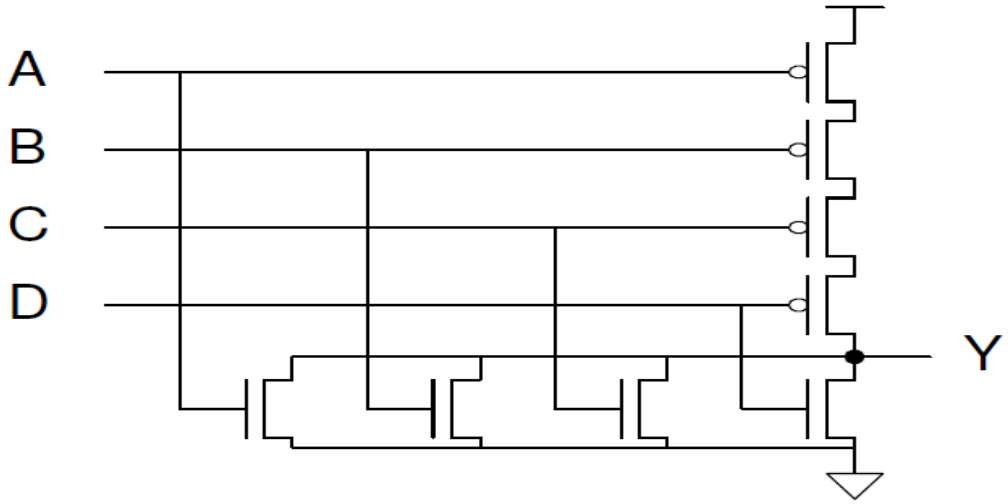




$$P_{tree} = \overline{A} \cdot \overline{B}$$
$$N_{tree} = A + B$$



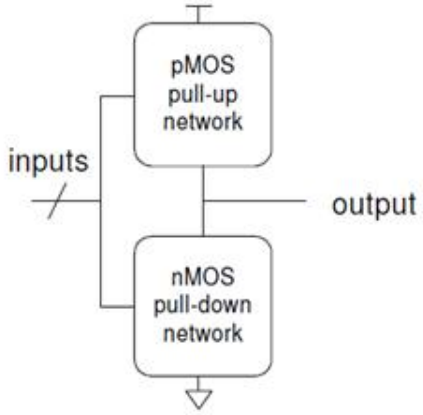
4-input CMOS NOR gate



5.6.3 CMOS Properties

Complementary CMOS logic gates
nMOS *pull-down network*
pMOS *pull-up netw* **CMOS**
Properties *ork*
a.k.a. static CMOS ,steady state is reached to 0 or 1.(no dc path from Vdd to gnd)

	Pull-up OFF	Pull-up ON
Pull-down OFF	Z (float)	1
Pull-down ON	0	X (crowbar)



- Complementary CMOS gates always produce 0 or 1
- Ex: NAND gate
- Series nMOS: Y=0 when both inputs are 1
- Thus Y=1 when either input is 0
- Requires parallel pMOS

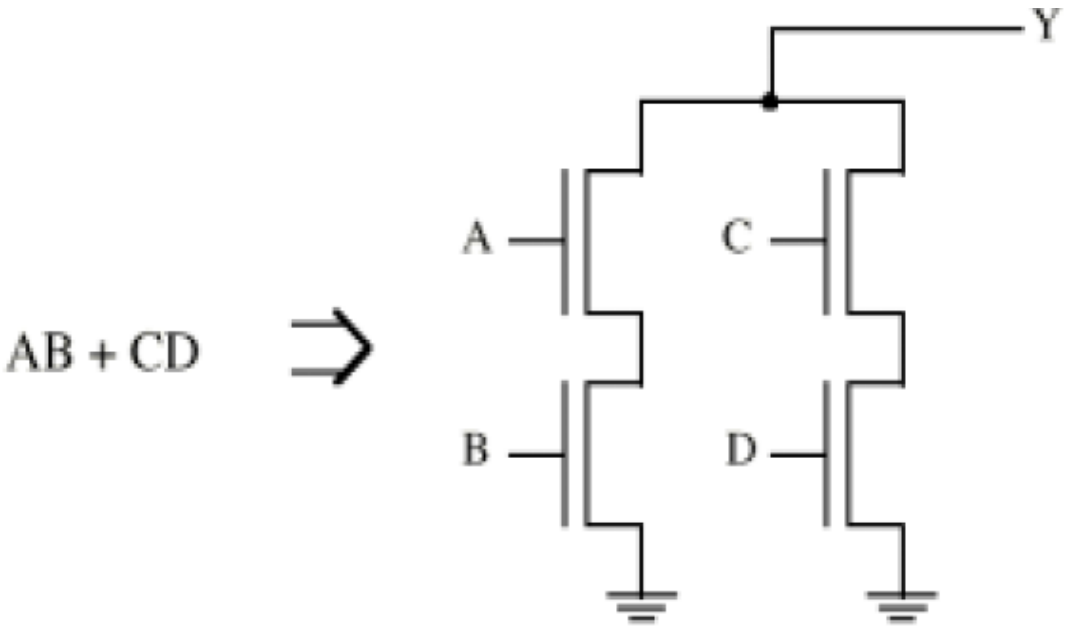
- Pull-up network is complement of pull-down
- Parallel -> series, series -> parallel
- Output signal strength is independent of input.-level restoring
- Restoring logic. Ouput signal strength is either V_{oh} (output high) or V_{ol}. (output low).
- Ratio less logic :output signal strength is independent of pMOS device size to nMOS size ratio.
- significant current only during the transition from one state to another and - hence power is conserved..
- Rise and fall transition times are of the same order,
- Very high levels of integration,
- High performance.

5.6.4 Complex gates..

$F = \overline{AB + CD} \Rightarrow N_{tree}$ will provide 0's, P_{tree} will provide 1's

0's of function F is $\overline{F}$, $\Rightarrow \overline{F} = \overline{\overline{AB + CD}} = AB + CD$

*n*MOS transistors need high true inputs, so it is desirable for all input variables to be high true, just as above.



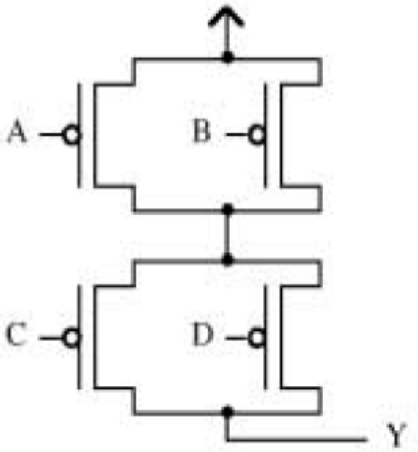
Likewise, a P_{tree} will provide 1's.

$F = \overline{AB + CD}$, need a form involving $\overline{A}$, $\overline{B}$, $\overline{C}$, $\overline{D}$

Apply DeMorgan's Theorem:

$F = \overline{AB} \cdot \overline{CD} = (\overline{A} + \overline{B}) \cdot (\overline{C} + \overline{D})$

Implementation ⇒

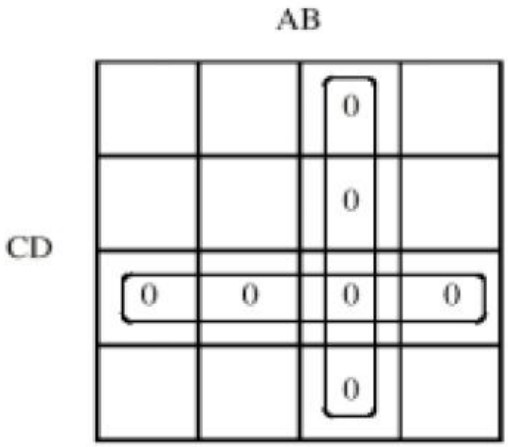


Can also use K-maps:

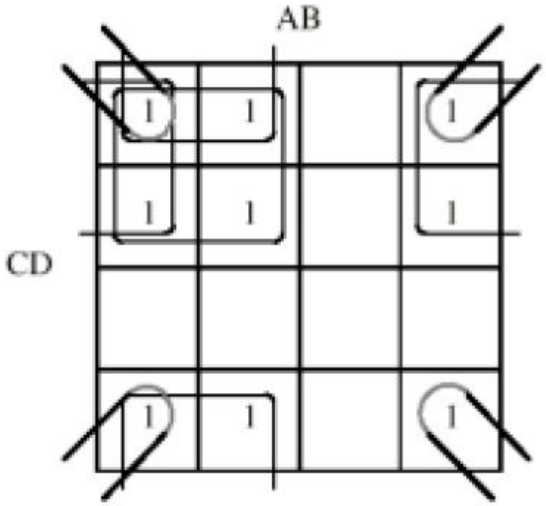
$$F = \overline{AB + CD}$$

CD \ AB				
	1	1	0	1
	1	1	0	1
	0	0	0	0
	1	1	0	1

For N_{tree}, minimize 0's; for P_{tree}, minimize 1's



$$N_{tree} = AB + CD$$

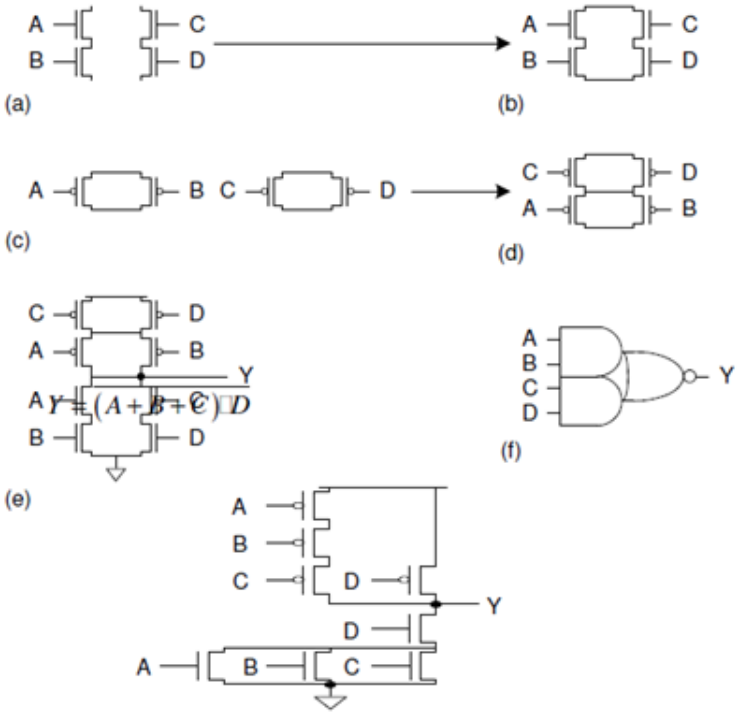


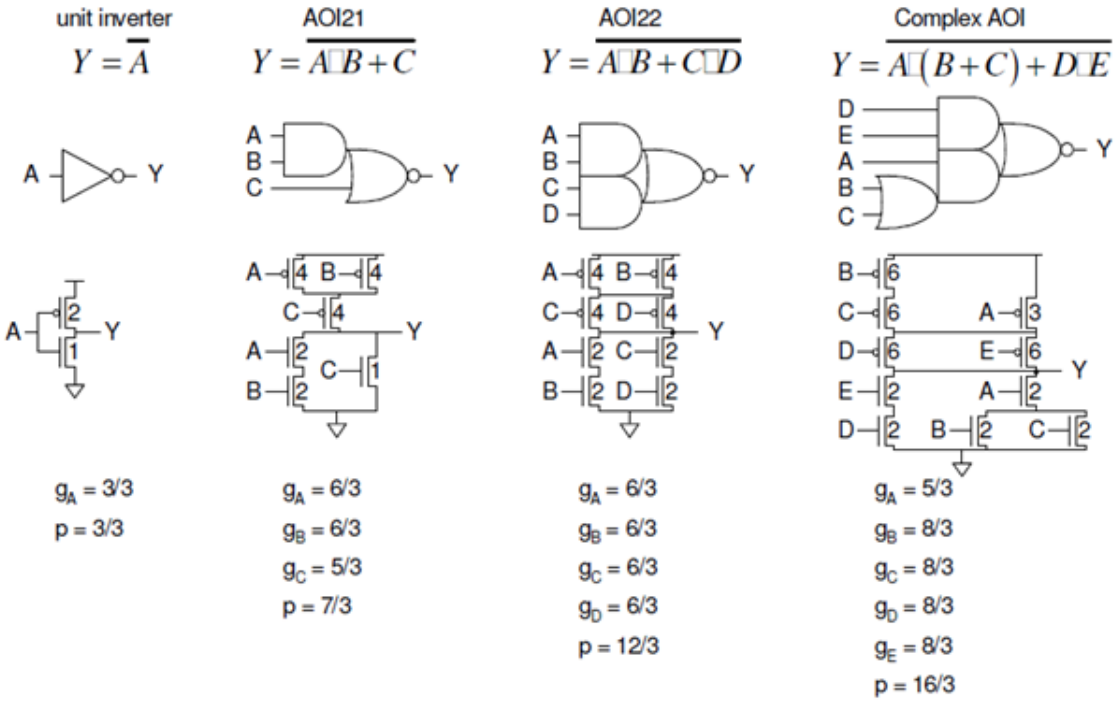
$$P_{tree} = \overline{A}\cdot\overline{C} + \overline{A}\cdot\overline{D} + \overline{B}\cdot\overline{C} + \overline{B}\cdot\overline{D}$$
$$= \overline{A}\,(\overline{C} + \overline{D}) + \overline{B}\,(\overline{C} + \overline{D})$$
$$= (\overline{A} + \overline{B}) \cdot (\overline{C} + \overline{D})$$

5.6.5 Complex gates AOI..

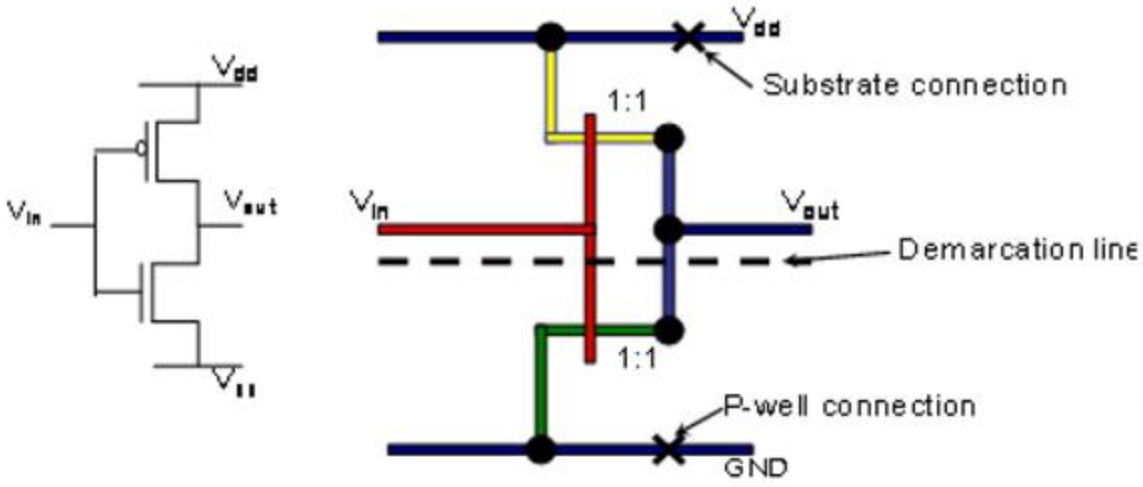
Compound gates can do any inverting function

$Y = \overline{A \cdot B + C \cdot D}$ (AND-AND-OR-INVERT, AOI22)

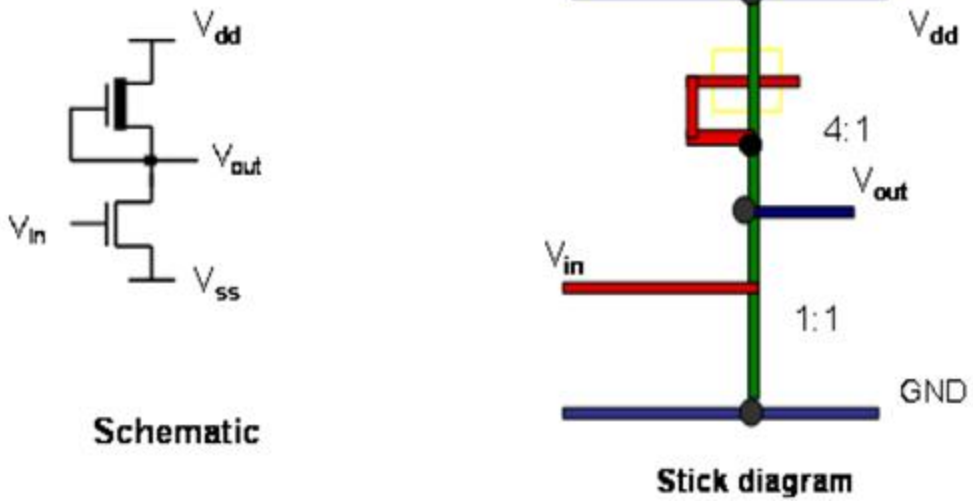




5.6.6 Circuit Families : Restoring logic CMOS Inverter- Stick diagram



5.6.7 Restoring logic CMOS Variants: nMOS Inverter-stick diagram



- Basic inverter circuit: load replaced by depletion mode transistor
- With no current drawn from output, the current I_{ds} for both transistor must be same.
- For the depletion mode transistor, gate is connected to the source so it is always on and only the characteristic curve $V_{gs}=0$ is relevant.

- Depletion mode is called pull-up and the enhancement mode device pull-down.
- Obtain the transfer characteristics.
- As V_{in} exceeds the p.d. threshold voltage current begins to flow, V_{out} thus decreases and further increase will cause p.d transistor to come out of saturation and become resistive.
- p.u transistor is initially resistive as the p.d is turned on.
- Point at which $V_{out} = V_{in}$ is denoted as V_{inv}
- Can be shifted by variation of the ratio of pull-up to pull-down resistances $-Z_{p.u} / Z_{p.d}$
- Z- ratio of channel length to width for each transistor

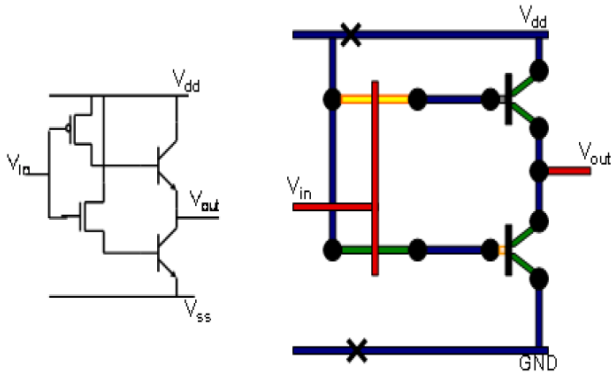
For 8:1 nMOS Inverter

$Z_{p.u.} = L_{p.u.} / W_{p.u} = 8$
$R_{p.u} = Z_{p.u.} * R_s = 80K$
similarly
$R_{p.d} = Z_{p.d} * R_s = 10K$
Power dissipation(on) $P_d = V^2 / R_{p.u} + R_{p.d} = 0.28mV$
Input capacitance = 1 C_g

For 4:1 nMOS Inverter

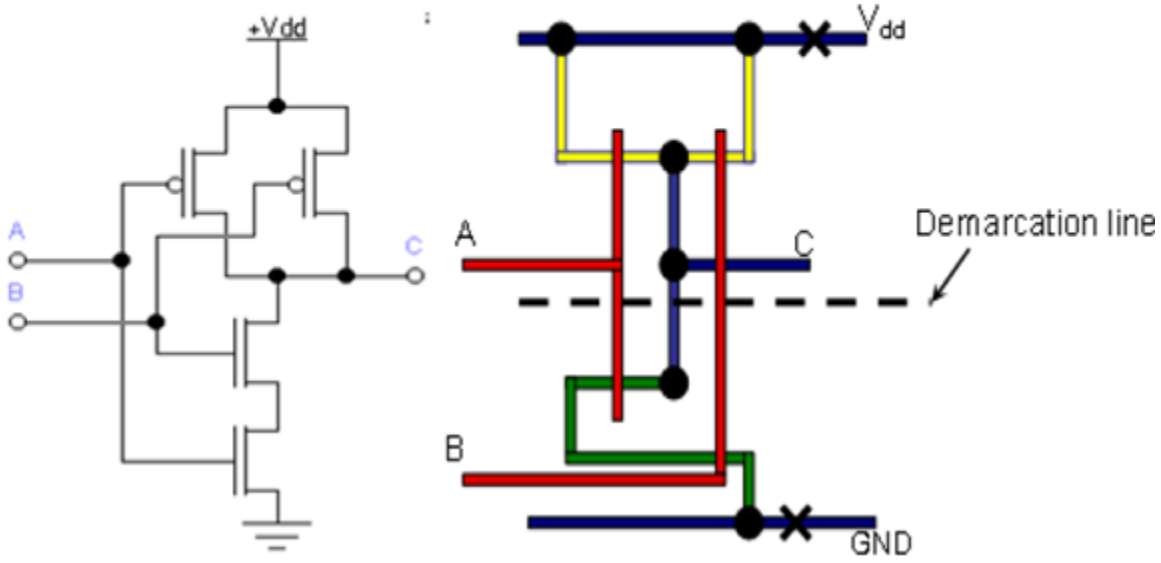
$Z_{p.u.} = L_{p.u.} / W_{p.u} = 4$
$R_{p.u} = Z_{p.u.} * R_s = 40K$
similarly
$R_{p.d} = Z_{p.d} * R_s = 5K$
Power dissipation(on) $P_d = V^2 / R_{p.u} + R_{p.d} = 0.56mV$
Input capacitance = $2C_g$

5.6.8Restoring logic CMOS Variants: BiCMOS Inverter-stick diagram

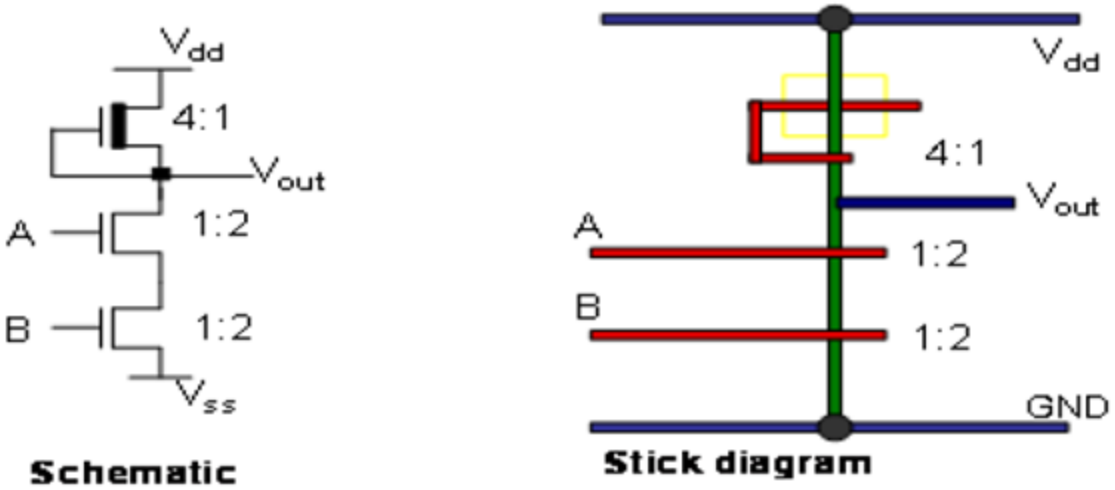


- A known deficiency of MOS technology is its limited load driving capabilities (due to limited current sourcing and sinking abilities of pMOS and nMOS transistors.)
- Output logic levels good-close to rail voltages
- High input impedance
- Low output impedance
- High drive capability but occupies a relatively small area.
- High noise margin
- Bipolar transistors have
    - higher gain
    - better noise characteristics
    - better high frequency characteristics
- BiCMOS gates can be an efficient way of speeding up VLSI circuits
- CMOS fabrication process can be extended for BiCMOS
- Example Applications
  CMOS- Logic
  BiCMOS- I/O and driver circuits
  ECL- critical high speed parts of the system

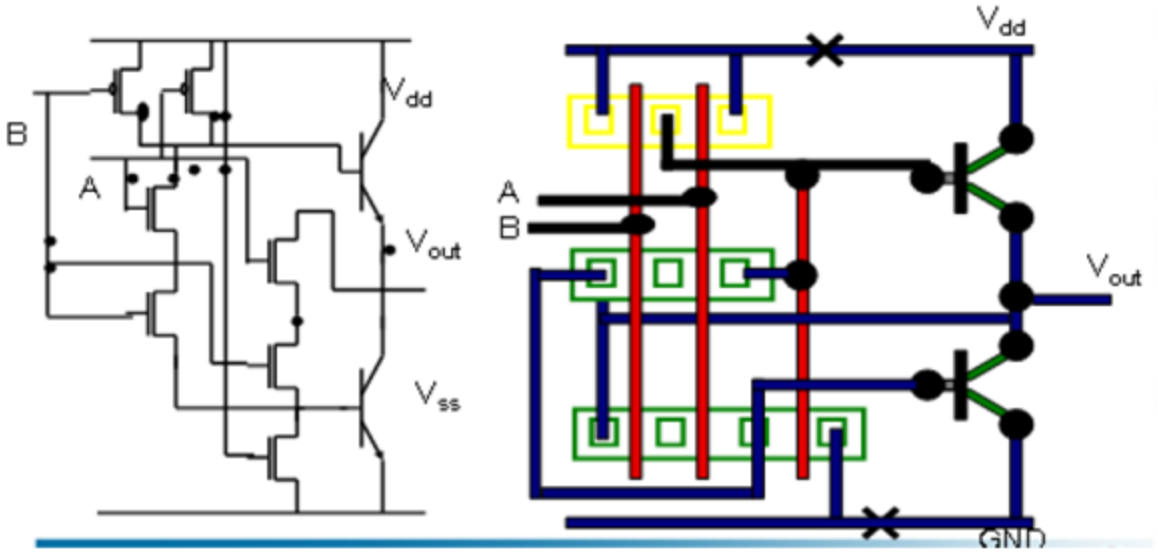
5.6.9 Circuit Families : Restoring logic CMOS NAND gate



5.6.10 Restoring logic CMOS Variants:_nMOS NAND gate



5.6.11 Restoring logic CMOS Variants: BiCMOS NAND gate

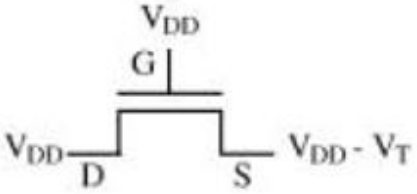


.

- For nMOS Nand-gate, the ratio between pull-up and sum of all pull-downs must be 4:1.
- nMOS Nand-gate area requirements are considerably greater than corresponding nMOS inverter
- nMOS Nand-gate delay is equal to number of input times inverter delay.
- Hence nMOS Nand-gates are used very rarely
- CMOS Nand-gate has no such restrictions
- BiCMOS gate is more complex and has larger fan-out.

5. 7.Circuit Families :Switch logic: Pass Transistor

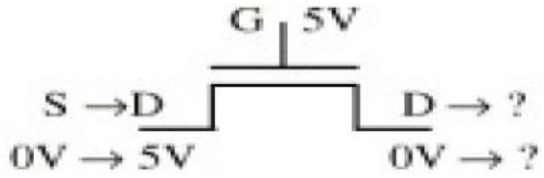
Why? *n*MOS switches cannot pass a logic "1" without a threshold voltage (V_T) drop.



where V_T = 0.7V to 1.0V (i.e., threshold voltage will vary)

output voltage = 4.3V to 4.0V, a *weak* "1"

The *n*MOS transistor will stop conducting if $V_{GS} < V_T$. Let V_T = 0.7V,

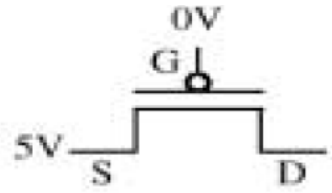


As source goes from 0V → 5V, V_{GS} goes from 5V → 0V.

When $V_S > 4.3V$, then $V_{GS} < V_T$, so switch stops conducting.

V_D left at 5V - V_T = 5V - 0.7V = 4.3V or Vdd - V_T.

For *p*MOS transistor, V_T is negative.



conducting

$V_{Tp} = -0.7V$

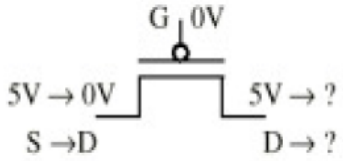
$V_{GS} = 0V - 5V = -5V$

$V_{GS} < V_{Tp}$ or $|V_{GS}| > |V_{Tp}|$

$-5V < -0.7V$ $5V > 0.7V$

How will *p*MOS pass a "0"?

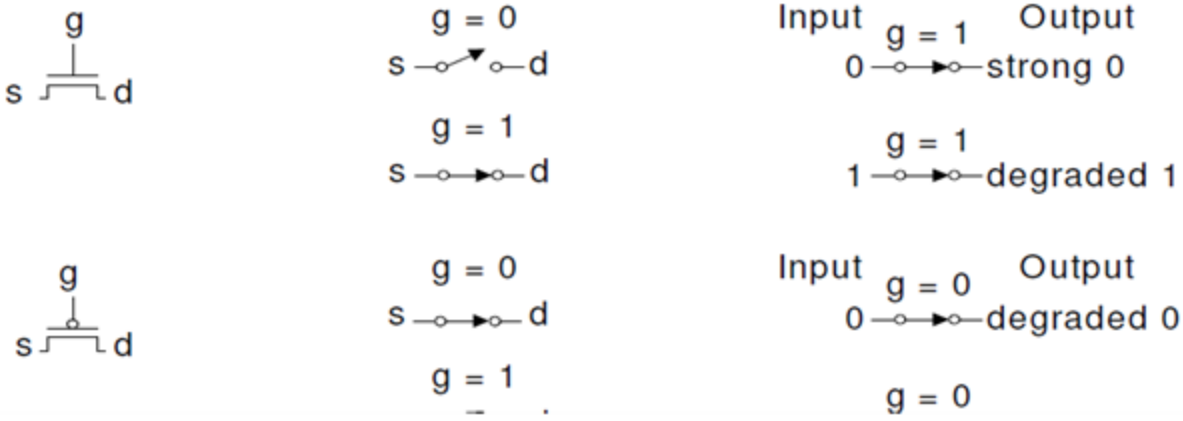
When $|V_{GS}| < |V_{Tp}|$, stop conducting



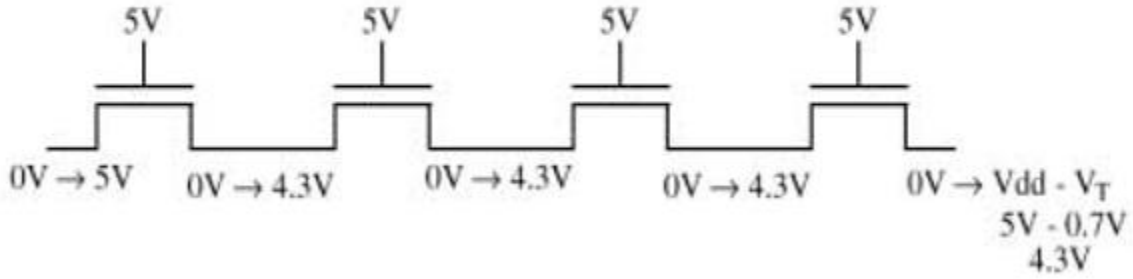
So when $|V_{GS}| < |-0.7V|$, V_D will go from 5V → **0.7V**,

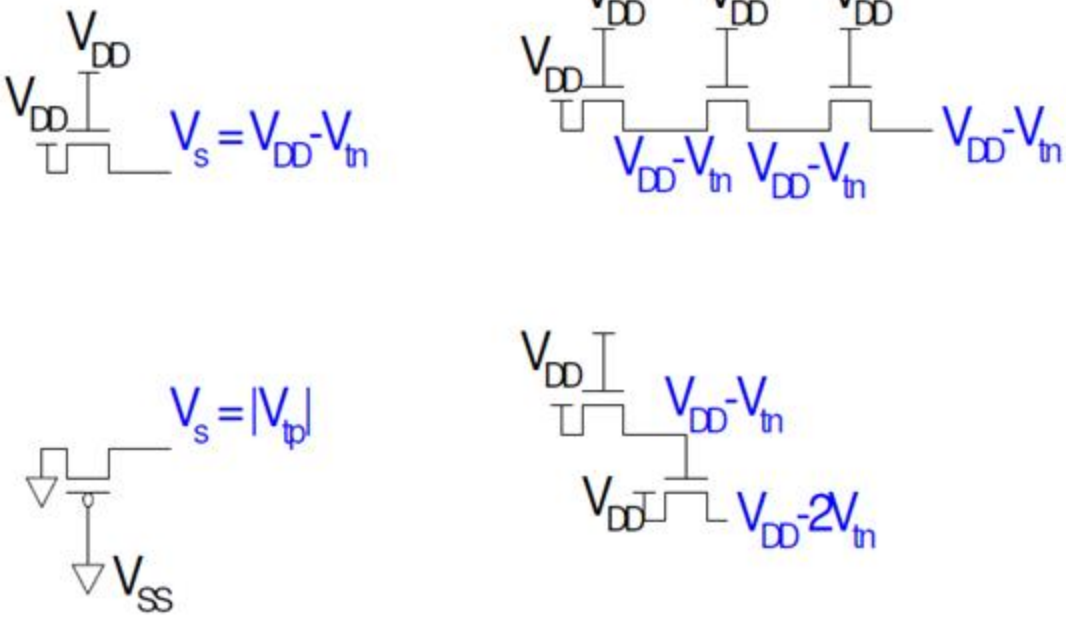
a *weak* "0"

5.7.1 Switch logic: Pass Transistor



5.7.1 Switch logic: Pass Transistor-nMOS in series

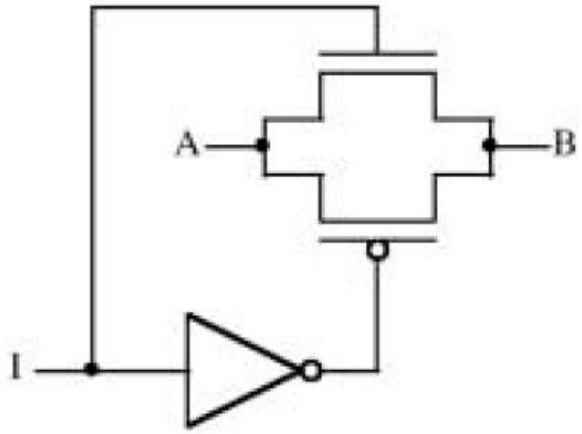




5.7.2 :Switch logic: Transmission gates

How are both a strong "1" and a strong "0" passed?

Transmission gate pass transistor configuration



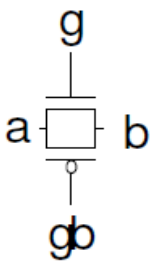
When I = 1,

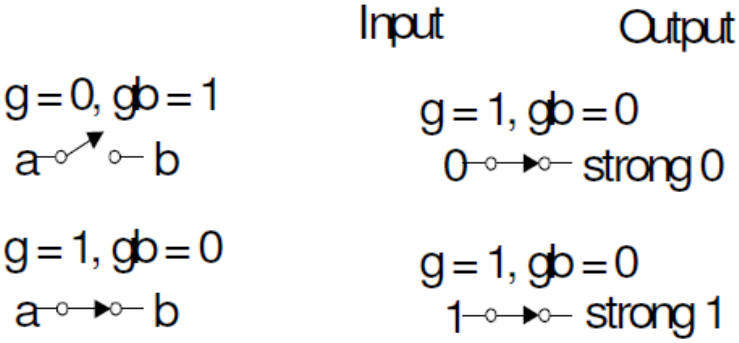
B = strong 1, if A = 1;

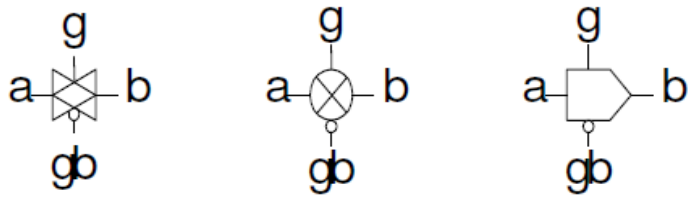
B = strong 0, if A = 0

When I = 0, non-conducting

Pass transistors produce degraded outputs
Transmission gates pass both 0 and 1 well



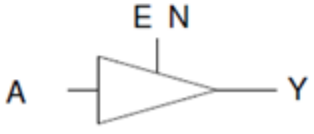




5.8 Structured Design-Tristate

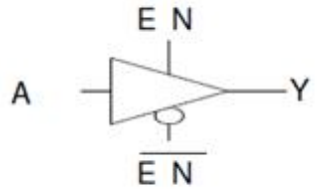
- ***Tristate buffer*** **produces Z when not enabled**

EN	A	Y
0	0	
0	1	
1	0	
1	1	



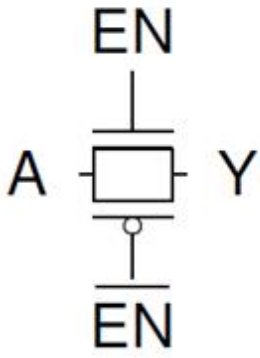
Tristate buffer produces Z when not enabled

EN	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1



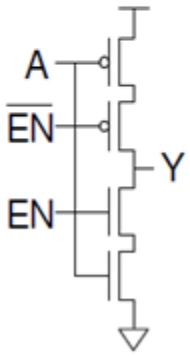
5.8.1 Structured Design-Nonrestoring Tristate

- Transmission gate acts as tristate buffer
  - Only two transistors
  - But *nonrestoring*
    - Noise on A is passed on to Y
  - \+ No V_t drop
  - \- Requires inverted clock

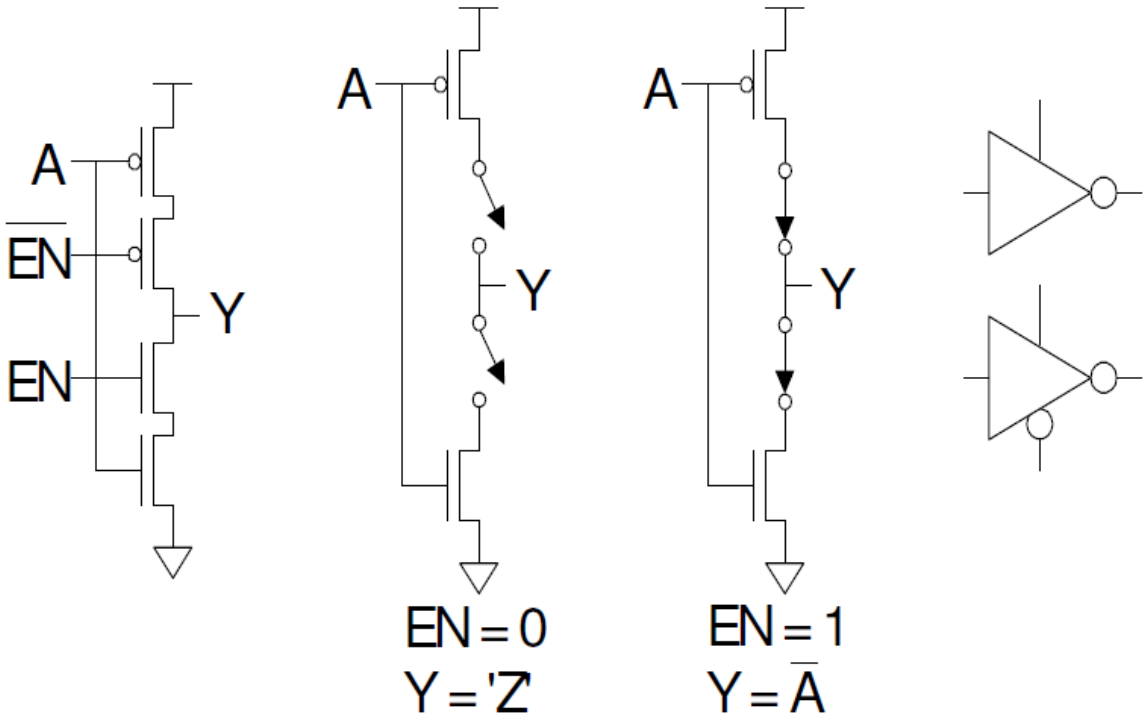


5.8.3 Structured Design-Tristate Inverter

- Tristate inverter produces restored output
  - Violates conduction complement rule
  - Because we want a Z output



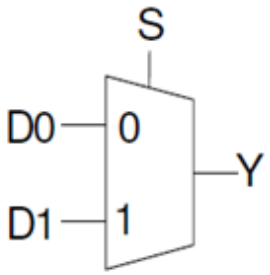
- **Tristate inverter produces restored output**
  - **Violates conduction complement rule**
  - **Because we want a Z output**



5.8.4 Structured Design-Multiplexers

- **2:1 *multiplexer* chooses between two inputs**

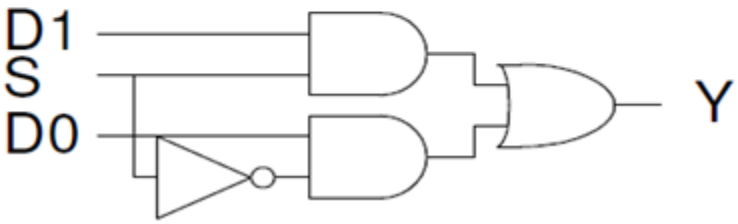
S	D1	D0	Y
0	X	0	0
0	X	1	1
1	0	X	0
1	1	X	1



5.8. 5 Structured Design-Mux Design.. Gate-Level

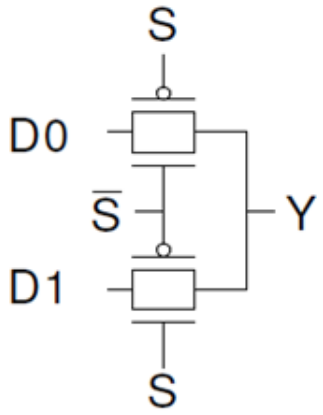
$$Y = SD_1 + \overline{S}D_0 \text{ (too many transistors)}$$

- **How many transistors are needed?**
- **How many transistors are needed? 20**



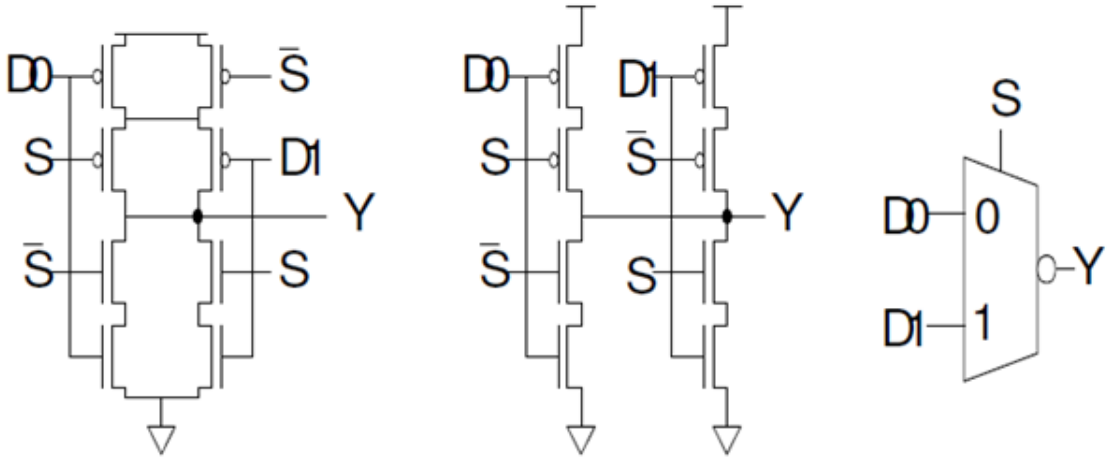
5.8.6 Structured Design-Mux Design-Transmission Gate

- Nonrestoring mux uses two transmission gates
  - Only 4 transistors



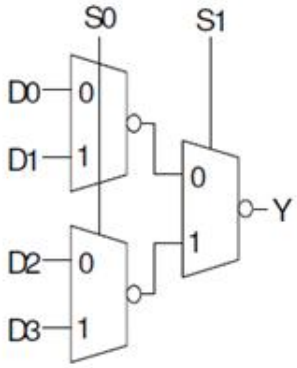
Inverting Mux

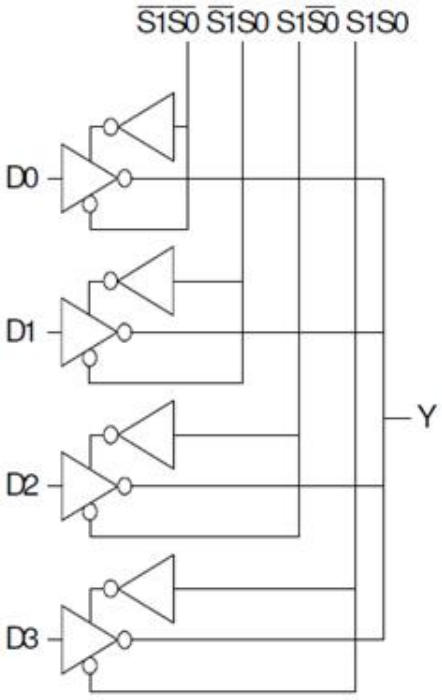
- Inverting multiplexer
  - Use compound AOI22
  - Or pair of tristate inverters
- **Noninverting multiplexer adds an inverter**



5.8.7 Design-4:1 Multiplexer

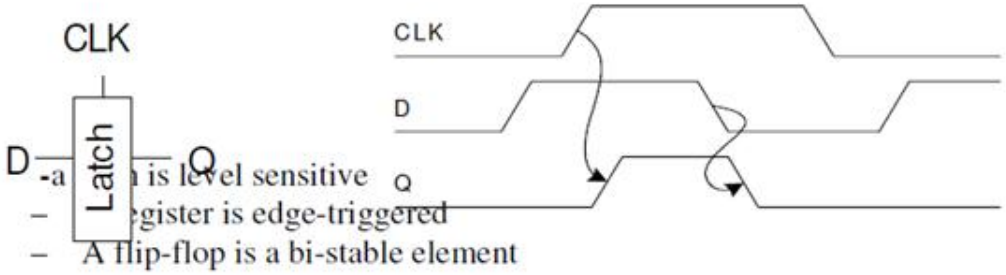
- 4:1 mux chooses one of 4 inputs using two selects
  Two levels of 2:1 muxes
  Or four tristates





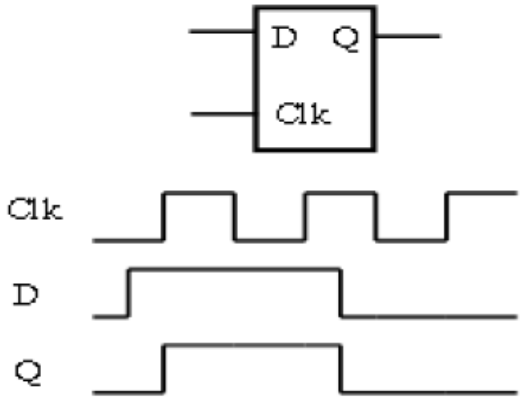
5.9 Structured Design-D Latch

- When CLK = 1, latch is *transparent*
  - D flows through to Q like a buffer
- When CLK = 0, the latch is *opaque*
  - Q holds its old value independent of D
- a.k.a. *transparent latch* or *level-sensitive latch*

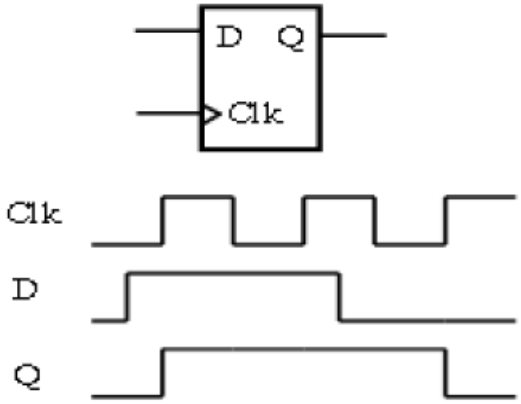


- -a is level sensitive
- – egister is edge-triggered
- – A flip-flop is a bi-stable element

- Latch
  stores data when clock is low

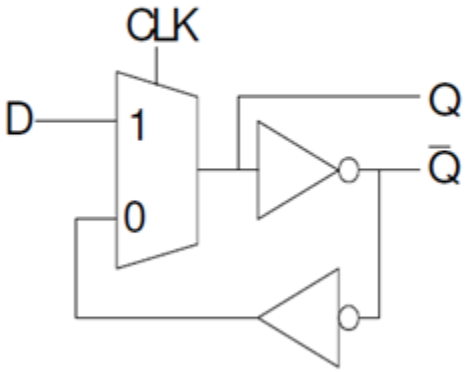


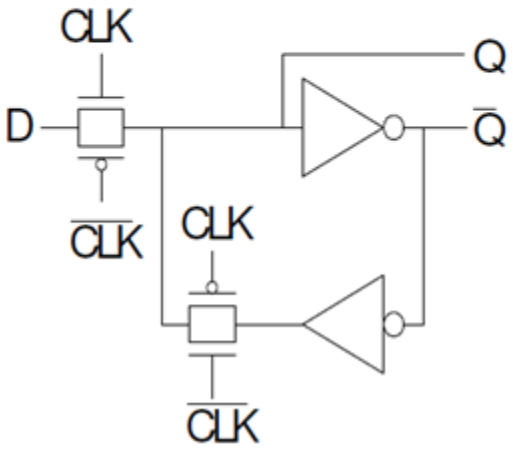
- Register
  stores data when clock rises



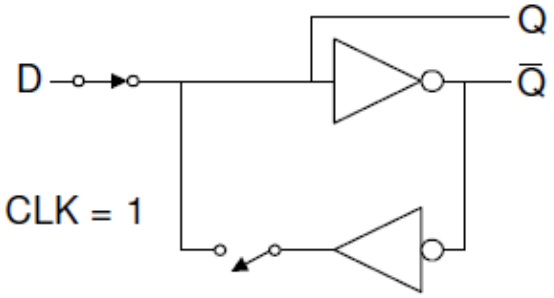
5.9.1 D Latch Design

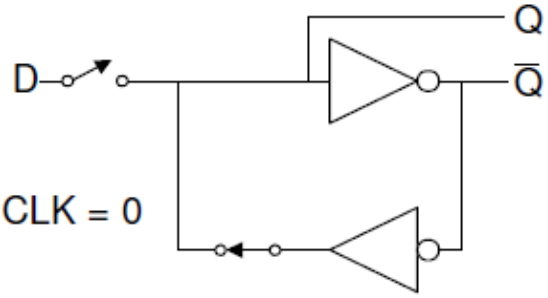
- Multiplexer chooses D or old Q



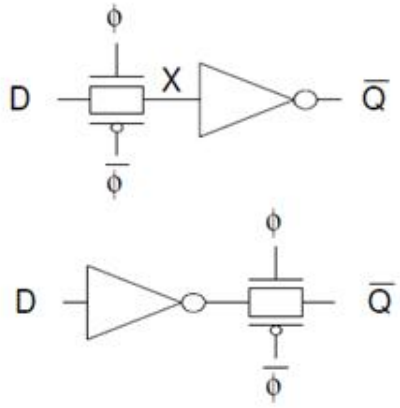


5.9.2 D Latch Operation





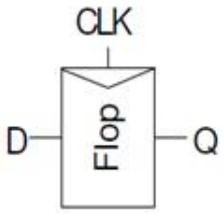
Structured Design-Latch Design

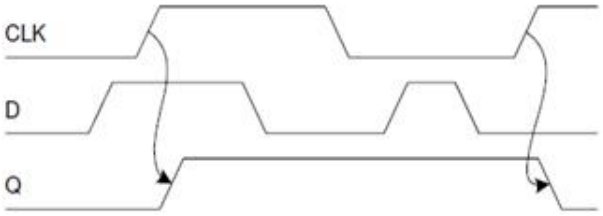


- Inverting buffer
  Restoring
  No backdriving
  Fixes either
  Output noise sensitivity
  Or diffusion input
  Inverted output

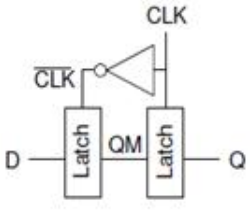
5. 9.3 Structured Design-D Flip-flop

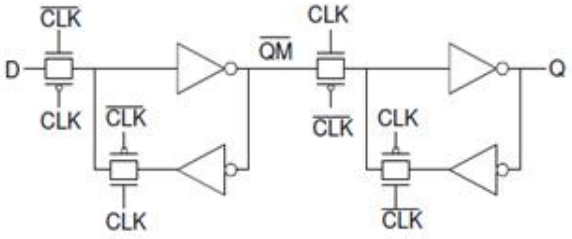
- When CLK rises, D is copied to Q
- At all other times, Q holds its value
- a.k.a. *positive edge-triggered flip-flop, master-slave flip-flop*



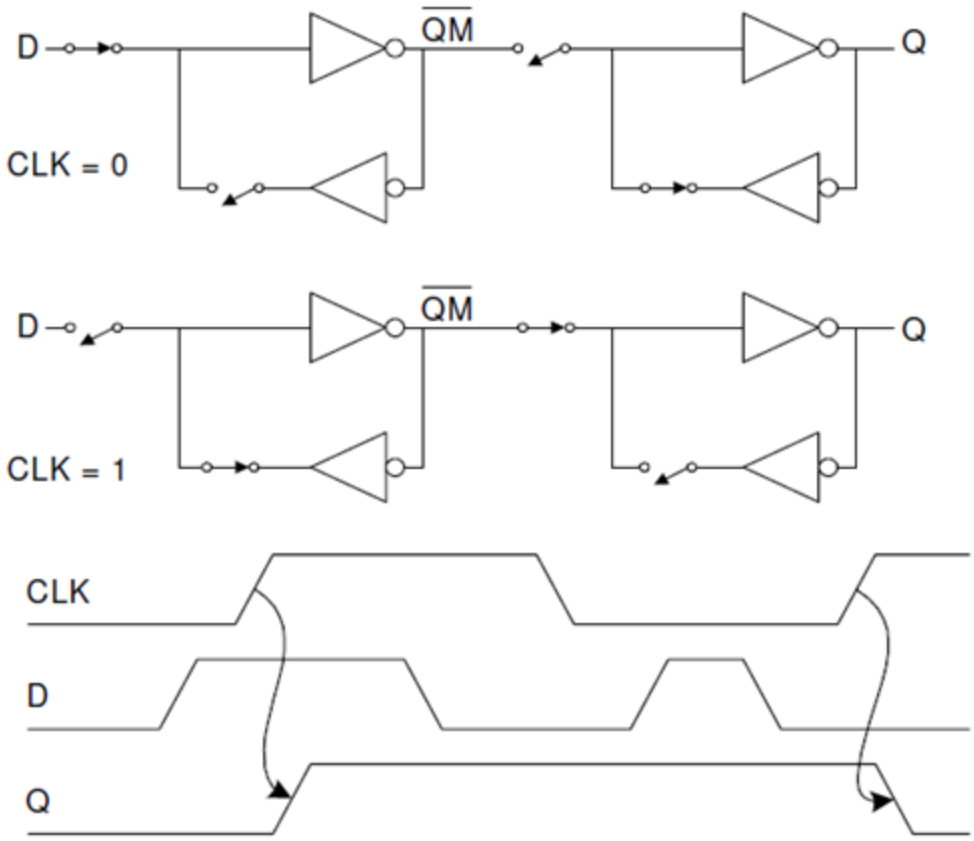


- Structured Design-D Flip-flop Design
- Built from master and slave D latches



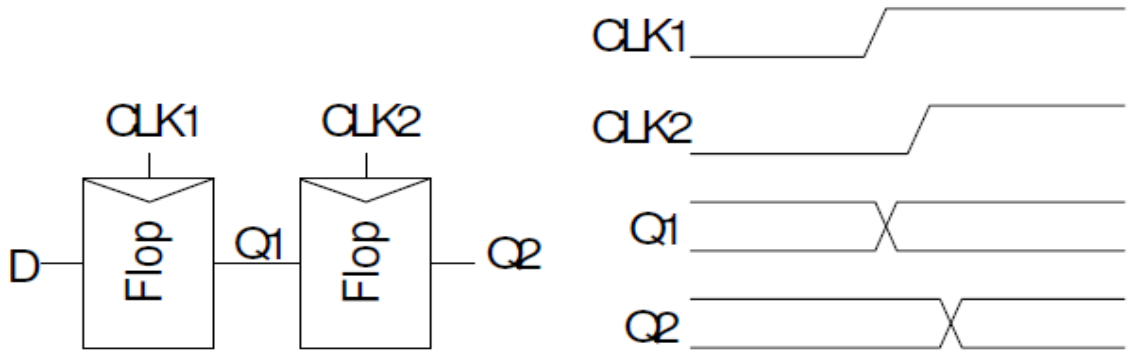


5.9.4 D Flip-flop Operation



5.9.5 Race Condition

- Back-to-back flops can malfunction from clock skew
  - Second flip-flop fires late
  - Sees first flip-flop change and captures its result
  - Called *hold-time failure* or *race condition*



Recommended questions:

1. Explain 4X4 cross bar switch operation. Mention the salient features of subsystem design process.
2. Explain the restoring logic in detail.
3. How to implement the switch logic for 4 way mux? Explain.
4. Describe switch and CMOS logic implementation for 2 input XOR gate.
5. Design a parity generator and draw the stick diagram for one basic cell.

Unit-6

CMOS subsystem design processes

General considerations, process illustration, ALU subsystem, adders, multipliers.

Recommended readings:

1. Douglas A. Pucknell & Kamran Eshraghian, **"Basic VLSI Design"** PHI 3rd Edition (original Edition – 1994), 2005.

2. Neil H. E. Weste and K. Eshragian**," Principles of CMOS VLSI Design: A System Perspective,"** 2nd edition, Pearson Education (Asia) Pvt. Ltd., 2000. History of VLSI

6.1 General Considerations

- Lower unit cost
- Higher reliability
- Lower power dissipation, lower weight and lower volume
- Better performance
- Enhanced repeatability
- Possibility of reduced design/development periods

6.1.1 Some Problems

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

6.1.2 Some Solution

Problem 1 & 3 are greatly reduced if two aspects of standard practices are accepted.

1. a) Top-down design approach with adequate CAD tools to do the job
   b) Partitioning the system sensibly
   c) Aiming for simple interconnections
   d) High regularity within subsystem
   e) Generate and then verify each section of the design
2. Devote significant portion of total chip area to test and diagnostic facility
3. Select architectures that allow design objectives and high regularity in realization

6.2 Illustration of design processes

1. Structured design begins with the concept of hierarchy
2. It is possible to divide any complex function into less complex sub functions that is up to leaf cells
3. Process is known as top-down design
4. As a systems complexity increases, its organization changes as different factors become relevant to its creation
5. Coupling can be used as a measure of how much submodels interact
6. It is crucial that components interacting with high frequency be physically proximate, since one may pay severe penalties for long, high-bandwidth interconnects

7. Concurrency should be exploited – it is desirable that all gates on the chip do useful work most of the time
8. Because technology changes so fast, the adaptation to a new process must occur in a short time.

Hence representing a design several approaches are possible. They are:

- Conventional circuit symbols
- Logic symbols
- Stick diagram
- Any mixture of logic symbols and stick diagram that is convenient at a stage
- Mask layouts
- Architectural block diagrams and floor plans

6.3 General arrangements of a 4 – bit arithmetic processor

The basic architecture of digital processor structure is as shown below in figure 6.1. Here the design of data path is only considered.

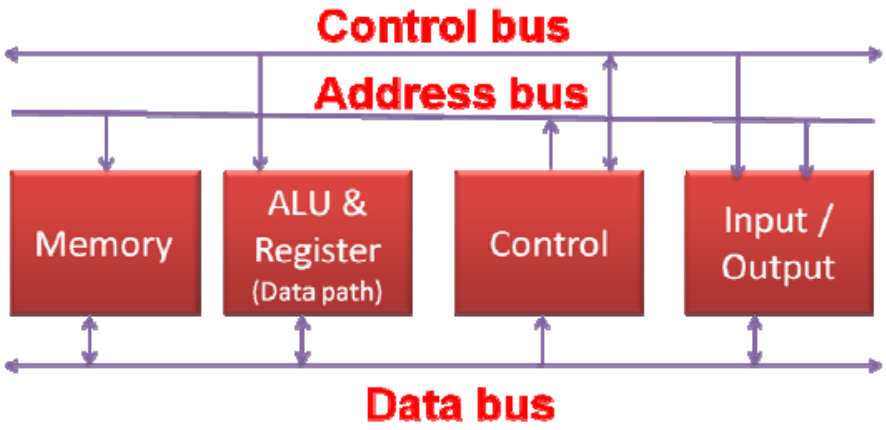


Figure 6.1: Basic digital processor structure

Datapath is as shown below in figure 6.2. It is seen that the structure comprises of a unit which processes data applied at one port and presents its output at a second port.

Alternatively, the two data ports may be combined as a single bidirectional port if storage facilities exist in the datapath. Control over the functions to be performed is effected by control signals as shown.

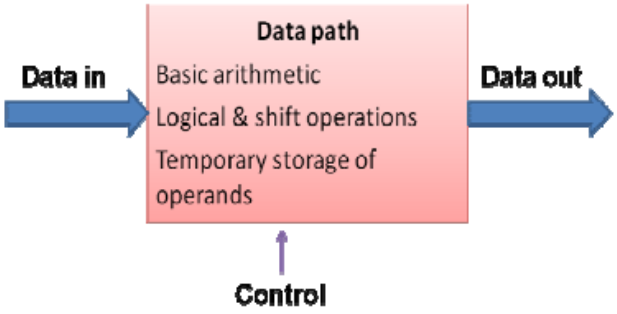


Figure 6.2: Communication strategy for the datapath

Datapath can be decomposed into blocks showing the main subunits as in figure 3. In doing so it is useful to anticipate a possible floor plan to show the planned relative decomposition of the subunits on the chip and hence on the mask layouts.

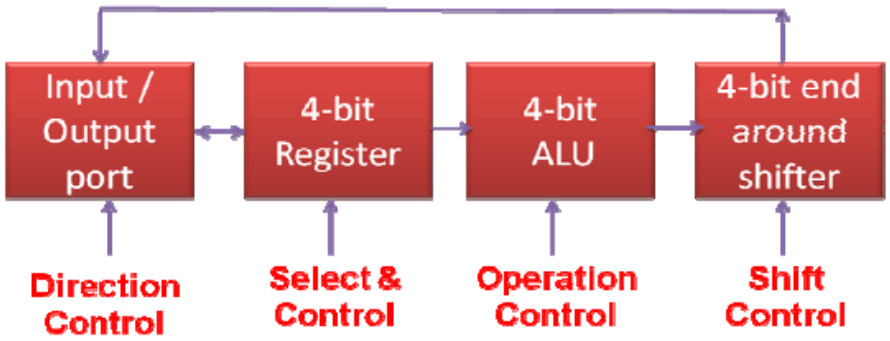


Figure 6.3: Subunits and basic interconnection for datapath

Nature of the bus architecture linking the subunits is discussed below. Some of the possibilities are:

One bus architecture:

Figure 6.4: One bus architecture

Sequence:

1. 1st operand from registers to ALU. Operand is stored there.
2. 2nd operand from register to ALU and added.
3. Result is passed through shifter and stored in the register

Two bus architecture:

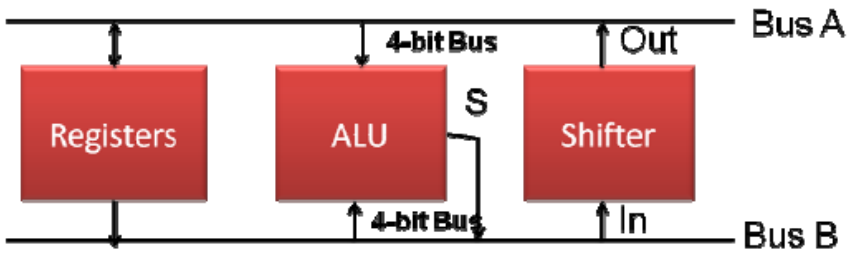


Figure 6.5: Two bus architecture

Sequence:

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

Three bus architecture:

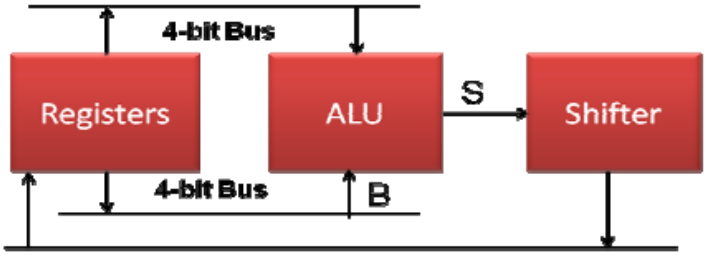


Figure 6.6: Three bus architecture

Sequence:

Two operands (A & B) are sent from registers, operated upon, and shifted result (S) returned to another register, all in same clock period.

In pursuing this design exercise, it was decided to implement the structure with a 2 – bus architecture. A tentative floor plan of the proposed design which includes some form of interface to the parent system data bus is shown in figure 6.7.

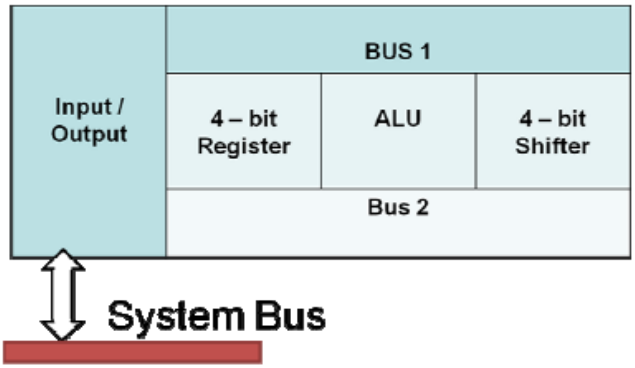


Figure 6.7: Tentative floor plan for 4 – bit datapath

The proposed processor will be seen to comprise a register array in which 4-bit numbers can be stored, either from an I/O port or from the output of the ALU via a shifter. Numbers from the register array can be fed in pairs to the ALU to be added (or subtracted) and the result can be shifted or not. The data connections between the I/O port, ALU, and shifter must be in the form of 4-bit buses. Also, each of the blocks must be suitably connected to control lines so that its function may be defined for any of a range of possible operations.

During the design process, and in particular when defining the interconnection strategy and designing the stick diagrams, care must be taken in allocating the layers to the various data or control paths. Points to be noted:

- ✓ Metal can cross poly or diffusion
- ✓ Poly crossing diffusion form a transistor
- ✓ Whenever lines touch on the same level an interconnection is formed
- ✓ Simple contacts can be used to join diffusion or poly to metal.
- ✓ Buried contacts or a butting contacts can be used to join diffusion and poly
- ✓ Some processes use 2nd metal
- ✓ 1st and 2nd metal layers may be joined using a via
- ✓ Each layer has particular electrical properties which must be taken into account
- ✓ For CMOS layouts, p-and n-diffusion wires must not directly join each other
- ✓ Nor may they cross either a p-well or an n-well boundary

Design of a 4-bit shifter

Any general purpose n-bit shifter should be able to shift incoming data by up to n – 1 place in a right-shift or left-shift direction. Further specifying that all shifts should be on an end-around basis, so that any bit shifted out at one end of a data word will be shifted in at the other end of the word, then the problem of right shift or left shift is greatly eased. It can be analyzed that for a 4-bit word, that a 1-bit shift right is equivalent to a 3-bit shift left and a 2-bit shift right is equivalent to a 2-bit left etc. Hence, the design of either shift right or left can be done. Here the design is of shift right by 0, 1, 2, or 3 places. The shifter must have:

- input from a four line parallel data bus
- four output lines for the shifted data
- means of transferring input data to output lines with any shift from 0 to 3 bits

Consider a direct MOS switch implementation of a 4 X 4 crossbar switches shown in figure 6.8. The arrangement is general and may be expanded to accommodate n-bit inputs/outputs. In this arrangement any input can be connected to any or all the outputs. Furthermore, 16 control signals (sw_{00} – sw_{15}), one for each transistor switch, must be provided to drive the crossbar switch, and such complexity is highly undesirable.

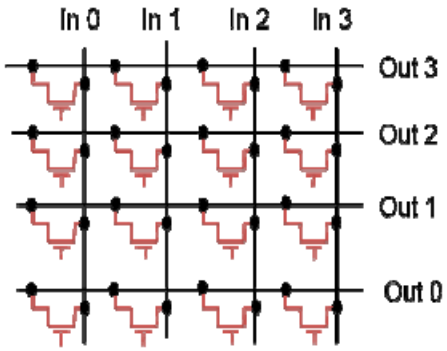


Figure 6.8: 4 X 4 crossbar switch

An adaptation of this arrangement recognizes the fact that we couple the switch gates together in groups of four and also form four separate groups corresponding to shifts of zero, one, two and three bits. The resulting arrangement is known as a barrel shifter and a 4 X 4 barrel shifter circuit diagram is as shown in the figure 6.9.

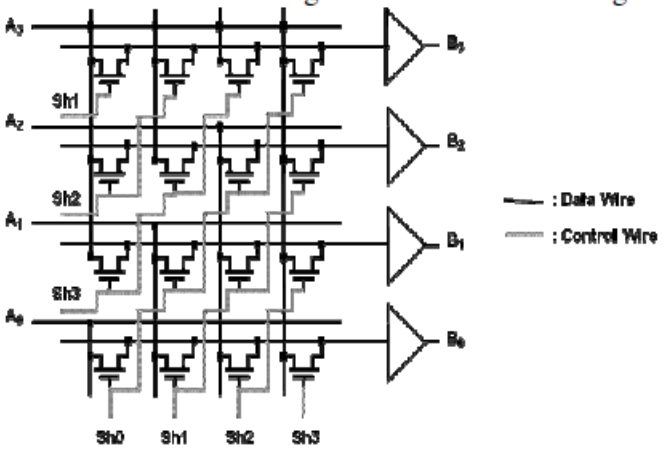


Figure 6.9: 4 X 4 barrel shifter

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

To summaries the design steps

- Set out the specifications
- Partition the architecture into subsystems
- Set a tentative floor plan
- Determine the interconnects
- Choose layers for the bus & control lines
- Conceive a regular architecture
- Develop stick diagram

- Produce mask layouts for standard cell
- Cascade & replicate standard cells as required to complete the design

6.4 Design of an ALU subsystem

Having designed the shifter, we shall design another subsystem of the 4-bit data path. An appropriate choice is ALU as shown in the figure 6.10 below.

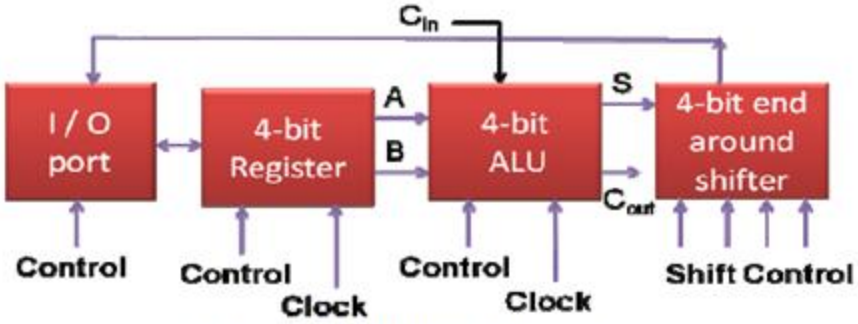


Figure 6.10: 4-bit data path for processor

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 array. Hence, for an adder two 4-bit parallel numbers are fed on two 4-bit buses. The clock signal is also required to the adder, during which the inputs are given and sum is generated. The shifter is unclocked but must be connected to four shift control lines.

Design of a 4-bit adder:

The truth table of binary adder is as shown in table 6.1

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

As seen from the table any column k there will be three inputs namely A_k , B_k as present input number and C_{k-1} as the previous carry. It can also be seen that there are two outputs sum S_k and carry C_k.

From the table one form of the equation is:

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

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

Where

Half sum $H_k = A_k' B_k + A_k B_k'$

Adder element requirements

Table 6.1 reveals that the adder requirement may be stated as:

If $A_k = B_k$ then $S_k = C_{k-1}$

Else $S_k = C_{k-1}'$

And for the carry C_k

If $A_k = B_k$ then $C_k = A_k = B_k$

Else $C_k = C_{k-1}$

Thus the standard adder element for 1-bit is as shown in the figure 6.11.

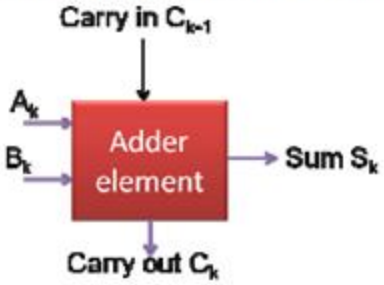


Figure 6.11: Adder element

6.4.1 Implementing ALU functions with an adder:

An ALU must be able to add and subtract two binary numbers, perform logical operations such as And, Or and Equality (Ex-or) functions. Subtraction can be performed by taking 2's complement of the negative number and perform the further addition. It is desirable to keep the architecture as simple as possible, and also see that the adder performs the logical operations also. Hence let us examine the possibility.

The adder equations are:

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

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

Where

Half sum $H_k = A_k' B_k + A_k B_k'$

Let us consider the sum output, if the previous carry is at logical 0, then

$S_k = H_k . 1 + H_k' . 0$

$S_k = H_k = A_k' B_k + A_k B_k'$ – An Ex-or operation

Now, if C_{k-1} is logically 1, then

$S_k = H_k . 0 + H_k' . 1$

$S_k = H_k$' – An Ex-Nor operation

Next, consider the carry output of each element, first C_{k-1} is held at logical 0, then

$C_k = A_kB_k + H_k . 0$

$C_k = A_kB_k$ - An And operation

Now if C_{k-1} is at logical 1, then

$C_k = A_kB_k + H_k . 1$

On solving $C_k = A_k + B_k$ - An Or operation

The adder element implementing both the arithmetic and logical functions can be implemented as shown in the figure 6.12.

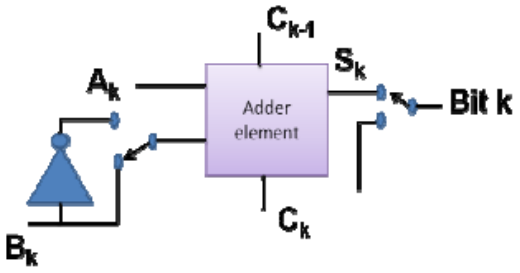


Figure 6.12: 1-bit adder element

The above can be cascaded to form 4-bit ALU.

A further consideration of adders

Generation:

This principle of generation allows the system to take advantage of the occurrences "$a_k=b_k$". In both cases ($a_k=1$ or $a_k=0$) the carry bit will be known.

Propagation:

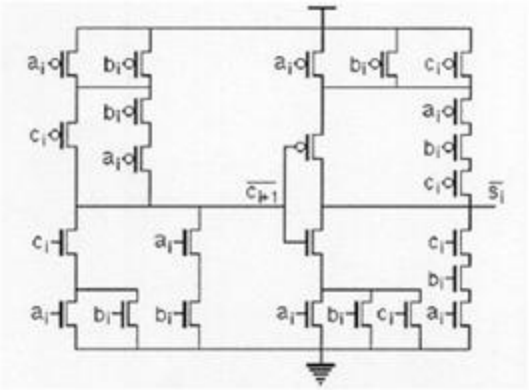
If we are able to localize a chain of bits $a_k\ a_{k+1}...a_{k+p}$ and $b_k\ b_{k+1}...b_{k+p}$ for which a_k not equal to b_k for k in [k,k+p], then the output carry bit of this chain will be equal to the input carry bit of the chain.

These remarks constitute the principle of generation and propagation used to speed the addition of two numbers.

All adders which use this principle calculate in a first stage.

$p_k = a_k$ XOR b_k

$g_k = a_k\ b_k$



6.4.2 Manchester carry – chain

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

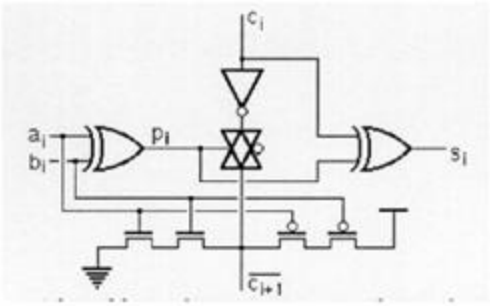


Figure-6.12: An adder with propagation signal controlling the pass-gate

In the schematic of Figure 6.12, the carry passes through a complete transmission gate. If the carry path is precharged to VDD, the transmission gate is then reduced to a simple NMOS transistor. In the same way the PMOS transistors of the carry generation is removed. One gets a Manchester cell.

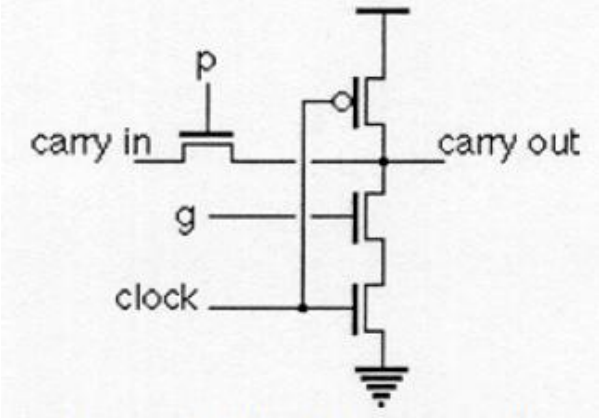


Figure-6.13: The Manchester cell

The Manchester cell is very fast, but a large set of such cascaded cells would be slow. This is due to the distributed RC effect and the body effect making the propagation time grow with the square of the number of cells. Practically, an inverter is added every four cells, like in Figure 6.14.

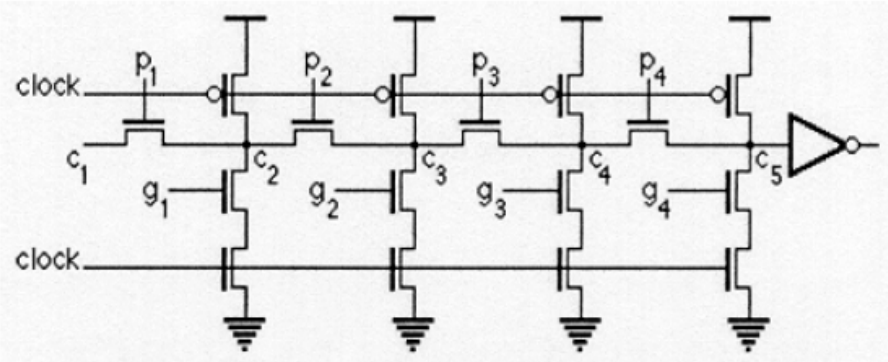


Figure-6.14: The Manchester carry cell

Adder Enhancement techniques

The operands of addition are the addend and the augend. The addend is added to the augend to form the sum. In most computers, the augmented operand (the augend) is replaced by the sum, whereas the addend is unchanged. High speed adders are not only for addition but also for subtraction, multiplication and division. The speed of a digital processor depends heavily on the speed of adders. The adders add vectors of bits and the principal problem is to speed- up the carry signal. A traditional and non optimized four bit adder can be made by the use of the generic one-bit adder cell connected one to the other. It is the ripple carry adder. In this case, the sum resulting at each stage need to wait for the incoming carry signal to perform the sum operation. The carry propagation can be speed-up in two ways. The first –and most obvious– way is to use a faster logic circuit technology. The second way is to generate carries by means of forecasting logic that does not rely on the carry signal being rippled from stage to stage of the adder.

6.4.3 The Carry-Skip Adder

Depending on the position at which a carry signal has been generated, the propagation time can be variable. In the best case, when there is no carry generation, the addition time will only take into account the time to propagate the carry signal. Figure 6.15 is an example illustrating a carry signal generated twice, with the input carry being equal to 0. In this case three simultaneous carry propagations occur. The longest is the second, which takes 7 cell delays (it starts at the 4th position and ends at the 11th position). So the addition time of these two numbers with this 16-bits Ripple Carry Adder is 7.k + k', where k is the delay cell and k' is the time needed to compute the 11th sum bit using the 11th carry-in.

With a Ripple Carry Adder, if the input bits Ai and Bi are different for all position i, then the carry signal is propagated at all positions (thus never generated), and the addition is completed when the carry signal has propagated through the whole adder. In this case, the Ripple Carry Adder is as slow as it is large. Actually, Ripple Carry Adders are fast only for some configurations of the input words, where carry signals are generated at some positions.

Carry Skip Adders take advantage both of the generation or the propagation of the carry signal. They are divided into blocks, where a special circuit detects quickly if all the bits to be added are different (Pi = 1 in all the block). The signal produced by this circuit will be called block propagation signal. If the carry is propagated at all positions in the block, then the carry signal entering into the block can directly bypass it and so be transmitted through a multiplexer to the next block. As soon as the carry signal is transmitted to a block, it starts to propagate through the block, as if it had been generated at the beginning of the block. Figure 6.16 shows the structure of a 24-bits Carry Skip Adder, divided into 4 blocks.

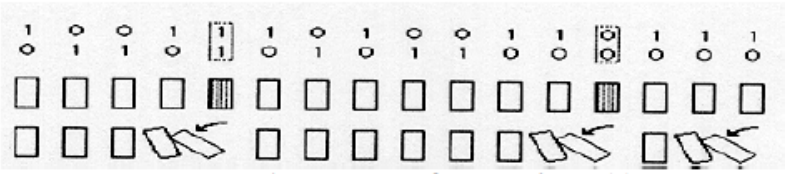


Figure 6.15: Example of Carry skip adder

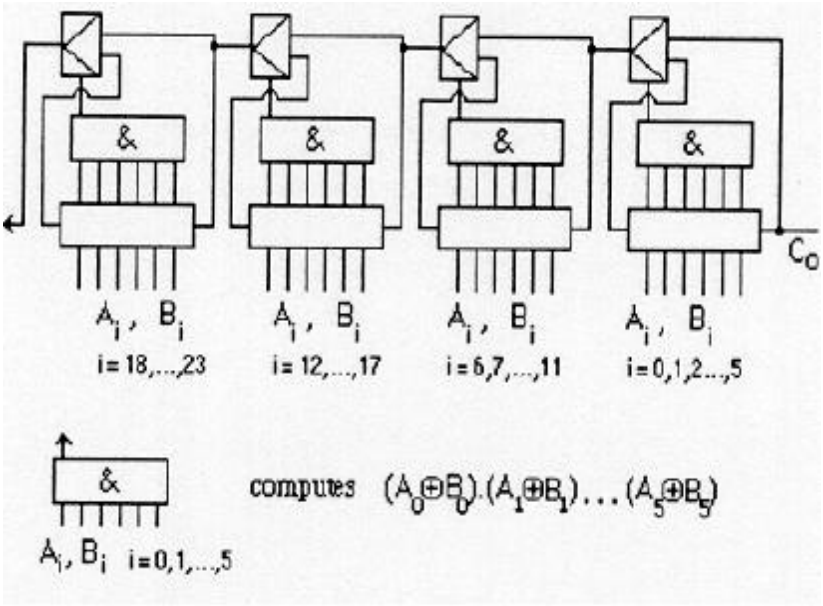


Figure-6.16: Block diagram of a carry skip adder

Optimization of the carry skip adder

It becomes now obvious that there exist a trade-off between the speed and the size of the blocks. In this part we analyze the division of the adder into blocks of equal size. Let us denote k1 the time needed by the carry signal to propagate through an adder cell, and k2 the time it needs to skip over one block. Suppose the N-bit Carry Skip Adder is divided into M blocks, and each block contains P adder cells. The actual addition time of a Ripple Carry Adder depends on the configuration of the input words. The completion time may be small but it also may reach the worst case, when all adder cells propagate the carry signal. In the same way, we must evaluate the worst carry propagation time for the Carry Skip Adder. The worst case of carry propagation is depicted in Figure 6.17.

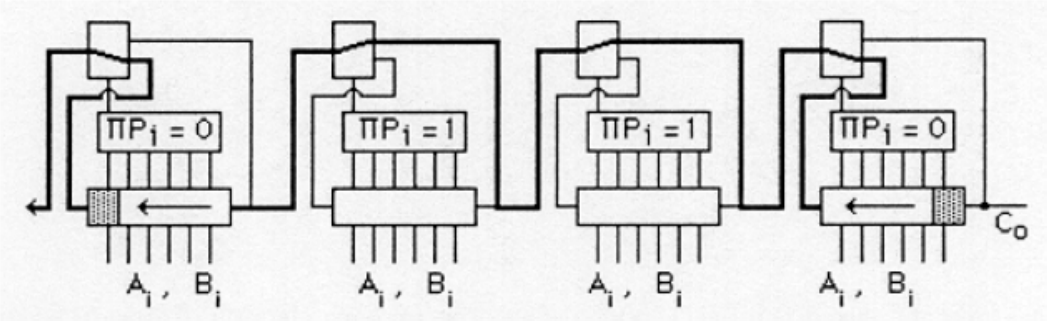


Figure-6.17: Worst case carry propagation for Carry Skip adder

The configuration of the input words is such that a carry signal is generated at the beginning of the first block. Then this carry signal is propagated by all the succeeding adder cells but the last which generates another carry signal. In the first and the last block the block propagation signal is equal to 0, so the entering carry signal is not transmitted to the next block. Consequently, in the first block, the last adder cells must wait for the carry signal, which comes from the first cell of the first block. When going out of the first

block, the carry signal is distributed to the 2nd, 3rd and last block, where it propagates. In these blocks, the carry signals propagate almost simultaneously (we must account for the multiplexer delays). Any other situation leads to a better case. Suppose for instance that the 2nd block does not propagate the carry signal (its block propagation signal is equal to zero), then it means that a carry signal is generated inside. This carry signal starts to propagate as soon as the input bits are settled. In other words, at the beginning of the addition, there exist two sources for the carry signals. The paths of these carry signals are shorter than the carry path of the worst case. Let us formalize that the total adder is made of N adder cells. It contains M blocks of P adder cells. The total of adder cells is then

$$N=M.P$$

The time T needed by the carry signal to propagate through P adder cells is

$$T=k_1.P$$

The time T' needed by the carry signal to skip through M adder blocks is

$$T'=k_2.M$$

The problem to solve is to minimize the worst case delay which is:

$$T_{worstcase} = 2 \cdot P \cdot k_1 + (M-2) \cdot k_2$$

$$T_{worstcase} = 2 \cdot \frac{N}{M} \cdot k_1 + (M-2) \cdot k_2$$

6.4.4 The Carry-Select Adder

This type of adder is not as fast as the Carry Look Ahead (CLA) presented in a next section. However, despite its bigger amount of hardware needed, it has an interesting design concept. The Carry Select principle requires two identical parallel adders that are partitioned into four-bit groups. Each group consists of the same design as that shown on Figure 6.18. The group generates a group carry. In the carry select adder, two sums are generated simultaneously. One sum assumes that the carry in is equal to one as the other assumes that the carry in is equal to zero. So that the predicted group carry is used to select one of the two sums.

It can be seen that the group carries logic increases rapidly when more high- order groups are added to the total adder length. This complexity can be decreased, with a subsequent increase in the delay, by partitioning a long adder into sections, with four groups per section, similar to the CLA adder.

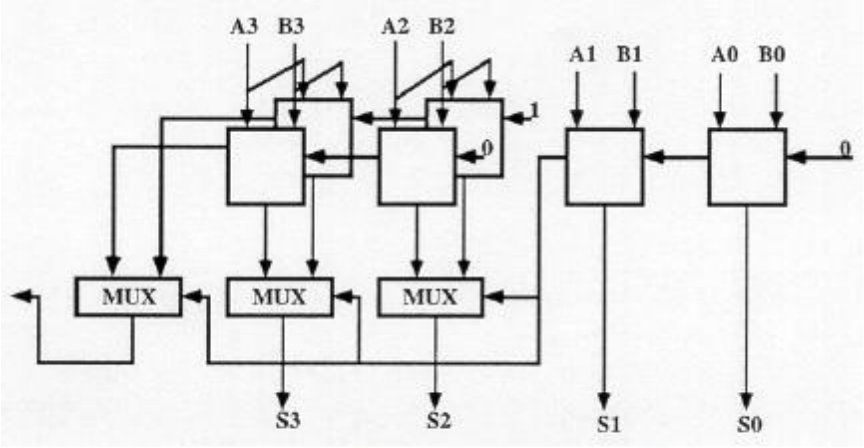


Figure-6.18: The Carry Select adder

Optimization of the carry select adder

- Computational time

$$T = K_1 n$$

- Dividing the adder into blocks with 2 parallel paths

$$T = K_1 n/2 + K_2$$

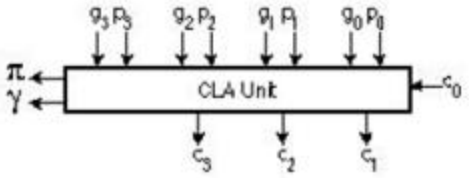
- For a n-bit adder of M-blocks and each block contains P adder cells in series $T = PK_1 + (M - 1)\,K_2$; $n = M.P$ minimum value for T is when $M=\sqrt{(K_1 n / K_1)}$

6.4.5 The Carry Look-Ahead Adder

The limitation in the sequential method of forming carries, especially in the Ripple Carry adder arises from specifying c_i as a specific function of c_{i-1}. It is possible to express a carry as a function of all the preceding low order carry by using the recursivity of the carry function. With the following expression a considerable increase in speed can be realized.

$$C_i = G_i + G_{i-2}P_{i-1} + G_{i-3}P_{i-2}P_{i-1} + \ldots \quad \ldots + G_0 P_1 P_2 \ldots P_{i-1} + C_0 P_0 P_1 P_2 \ldots P_{i-1}$$

Usually the size and complexity for a big adder using this equation is not affordable. That is why the equation is used in a modular way by making groups of carry (usually four bits). Such a unit generates then a group carry which give the right predicted information to the next block giving time to the sum units to perform their calculation.



$$\pi = p_0 p_1 p_2 p_3$$

$$\gamma = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0$$

$$c_4 = \gamma + \pi . c_0$$

Figure-6.19: The Carry Generation unit performing the Carry group computation

Such unit can be implemented in various ways, according to the allowed level of abstraction. In a CMOS process, 17 transistors are able to guarantee the static function (Figure 6.20). However this design requires a careful sizing of the transistors put in series.

The same design is available with less transistors in a dynamic logic design. The sizing is still an important issue, but the number of transistors is reduced (Figure 6.21).

$c_1 = g_0 + p_0 \cdot c_0$

$c_2 = g_1 + p_1 \cdot [g_0 + p_0 \cdot c_0]$

$c_3 = g_2 + p_2 \cdot [g_1 + p_1 \cdot [g_0 + p_0 \cdot c_0]]$

$c_4 = g_3 + p_3 \cdot [g_2 + p_2 \cdot [g_1 + p_1 \cdot [g_0 + p_0 \cdot c_0]]]$

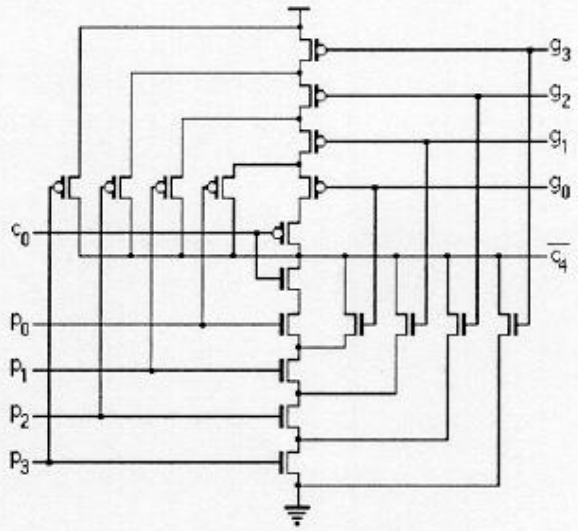


Figure-6.20: Static implementation of the 4-bit carry lookahead chain

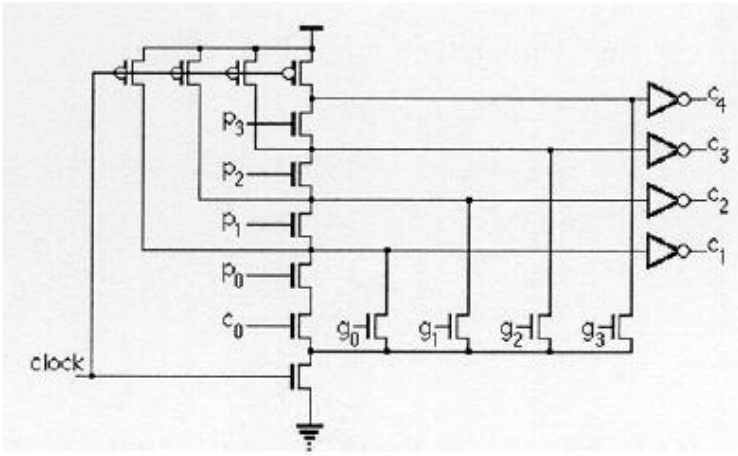


Figure-6.21: Dynamic implementation of the 4-bit carry lookahead chain

Figure 6.22 shows the implementation of 16-bit CLA adder.

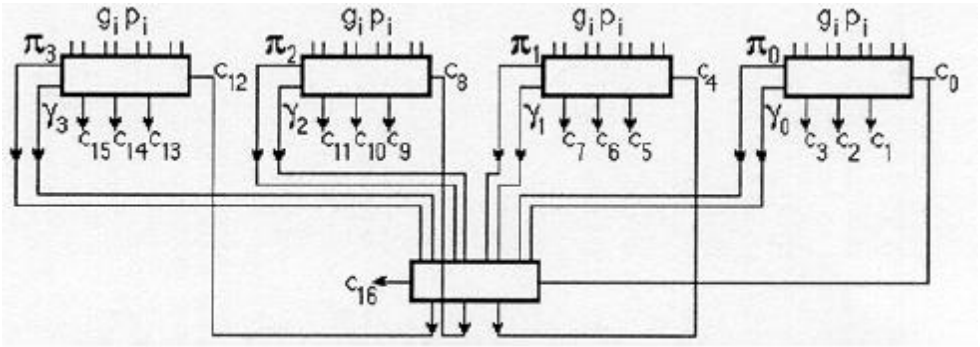


Figure-6.22: Implementation of a 16-bit CLA adder

6.5 Multipliers

Introduction

Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Each step of the addition generates a partial product. In most computers, the operands usually contain the same number of bits. When the operands are interpreted as integers, the product is generally twice the length of the operands in order to preserve the information content. This repeated addition method that is suggested by the arithmetic definition is slow that it is almost always replaced by an algorithm that makes use of positional number representation.

It is possible to decompose multipliers in two parts. The first part is dedicated to the generation of partial products, and the second one collects and adds them. As for adders, it is possible to enhance the intrinsic performances of multipliers. Acting in the generation part, the Booth (or modified Booth) algorithm is often used because it reduces the number of partial products. The collection of the partial products can then be made using a regular array, a Wallace tree or a binary tree

Serial-Parallel Multiplier

This multiplier is the simplest one, the multiplication is considered as a succession of additions.

if $A = (a_n\ a_{n-1}\ldots\ldots a_0)$ and $B = (b_n\ b_{n-1}\ldots\ldots b_0)$

The product A.B is expressed as :

$$A.B = A.2^n.b_n + A.2^{n-1}.b_{n-1} + \ldots + A.2_0.b^0$$

The structure of Figure 6.23 is suited only for positive operands. If the operands are negative and coded in 2's complement:

1. The most significant bit of B has a negative weight, so a subtraction has to be performed at the last step.

2. Operand $A.2^k$ must be written on 2N bits, so the most significant bit of A must be duplicated. It may be easier to shift the content of the accumulator to the right instead of shifting A to the left.

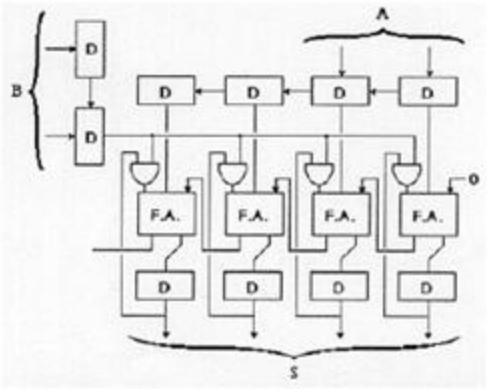


Figure-6.23: Serial-Parallel multiplier

6.5.1 Braun Parallel Multiplier

The simplest parallel multiplier is the Braun array. All the partial products A.bk are computed in parallel, and then collected through a cascade of Carry Save Adders. At the bottom of the array, the output of the array is noted in Carry Save, so an additional adder converts it (by the mean of carry propagation) into the classical notation (Figure 6.24). The completion time is limited by the depth of the carry save array, and by the carry propagation in the adder. Note that this multiplier is only suited for positive operands. Negative operands may be multiplied using a Baugh-Wooley multiplier.

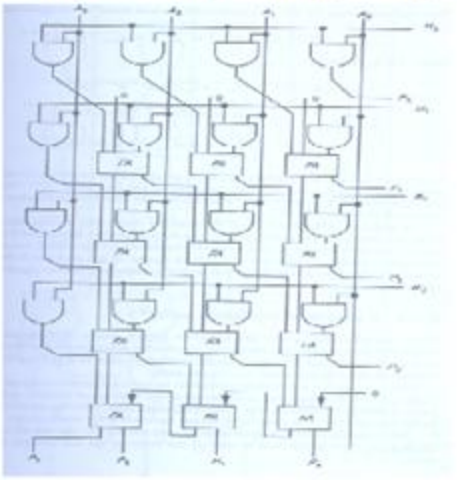


Figure 6.24: A 4-bit Braun Array

6.5.2 Baugh-Wooley Multiplier

This technique has been developed in order to design regular multipliers, suited for 2's-complement numbers.

Let us consider 2 numbers A and B:

$$A = (a_{n-1} \dots a_0) = -a_{n-1}.2^{n-1} + \sum_{0}^{n-2} a_i.2^i$$

$$B = (b_{n-1} \dots b_0) = -b_{n-1}.2^{n-1} + \sum_{0}^{n-2} b_i.2^i$$

The product A.B is given by the following equation:

$$A.B = a_{n-1}.b_{n-1}.2^{2n-2} + \sum_{0}^{n-2}\sum_{0}^{n-2} a_i.b_i.2^{i+j} - a_{n-1}\sum_{0}^{n-2} b_i.2^{n+i-1} - b_{n-1}\sum_{0}^{n-2} a_i.2^{n+i-1}$$

We see that subtraction cells must be used. In order to use only adder cells, the negative terms may be rewritten as:

$$-a_{n-1}\sum_{0}^{n-2} b_i.2^{i+n-1} = a_{n-1}.\left(-2^{2n-2} + 2^{n-1} + \sum_{0}^{n-2} \overline{b_i}.2^{i+n-1}\right)$$

By this way, A.B becomes:

$$A.B = a_{n-1}.b_{n-1}.2^{2n-2} + \sum_{0}^{n-2}\sum_{0}^{n-2} a_i.b_j.2^{i+j}$$

$$+ b_{n-1}\left[-2^{2n-2} + 2^{n-1} + \sum_{0}^{n-2} \overline{a_i}.2^{i+n-1}\right]$$

$$+ a_{n-1}\left[-2^{2n-2} + 2^{n-1} + \sum_{0}^{n-2} \overline{b_i}.2^{i+n-1}\right]$$

The final equation is:

$$A.B = -2^{2n-1} + \left(\overline{a_{n-1}} + \overline{b_{n-1}} + a_{n-1}.b_{n-1}\right).2^{2n-2}$$

$$+\sum_{0}^{n-2}\sum_{0}^{n-2} a_i.b_j.2^{i+j} + \left(a_{n-1} + b_{n-1}\right).2^{n-1}$$

$$+\sum_{0}^{n-2} b_{n-1}.\overline{a_i}.2^{i+n-1} + \sum_{0}^{n-2} a_{n-1}.\overline{b_i}.2^{i+n-1}$$

A and B are n-bits operands, so their product is a 2n-bits number. Consequently, the most significant weight is 2n-1, and the first term -2^{2n-1} is taken into account by adding a 1 in the most significant cell of the multiplier. The implementation is shown in figure 6.25.

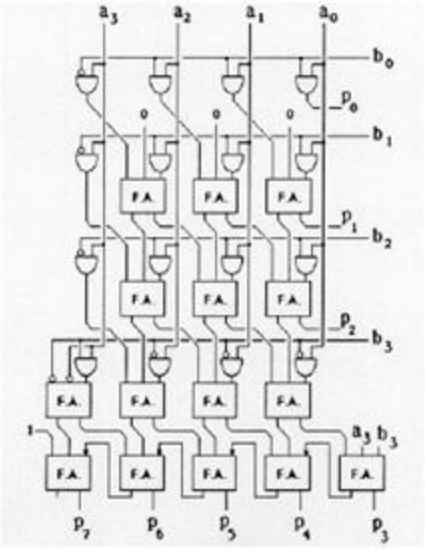


Figure-6.25: A 4-bit Baugh-Wooley Multiplier

6.5.3 Booth Algorithm

This algorithm is a powerful direct algorithm for signed-number multiplication. It generates a 2n-bit product and treats both positive and negative numbers uniformly. The idea is to reduce the number of additions to perform. Booth algorithm allows in the best case n/2 additions whereas modified Booth algorithm allows always n/2 additions.

Let us consider a string of k consecutive 1s in a multiplier:

..., i+k, i+k-1, i+k-2 , ..., i, i-1, ...
..., 0 , 1 , 1 , ..., 1, 0, ...

where there is k consecutive 1s.

By using the following property of binary strings:

$$2^{i+k}-2^{i}=2^{i+k-1}+2^{i+k-2}+...+2^{i+1}+2^{i}$$

the k consecutive 1s can be replaced by the following string

..., i+k+1, i+k, i+k-1, i+k-2, ..., i+1, i , i-1 , ...
..., 0 , 1 , 0 , 0 , ..., 0 , -1 , 0 , ...
k-1 consecutive 0s Addition Subtraction

In fact, the modified Booth algorithm converts a signed number from the standard 2's-complement radix into a number system where the digits are in the set {-1,0,1}. In this number system, any number may be written in several forms, so the system is called redundant.

The coding table for the modified Booth algorithm is given in Table 1. The algorithm scans strings composed of three digits. Depending on the value of the string, a certain operation will be performed.

A possible implementation of the Booth encoder is given on Figure 6.26.

Table-1: Modified Booth coding table

BIT				M is
2^1	2^0	2^{-1}	OPERATION	multiplied
Y_{i+1}	Y_i	Y_{i-1}		by
0	0	0	add zero (no string)	+0
0	0	1	add multipleic (end of string)	+X
0	1	0	add multiplic. (a string)	+X
0	1	1	add twice the mul. (end of string)	+2X
1	0	0	sub. twice the m. (beg. of string)	-2X
1	0	1	sub. the m. (-2X and +X)	-X
1	1	0	sub . the m. (beg. of string)	-X
1	1	1	sub. zero (center of string)	-0

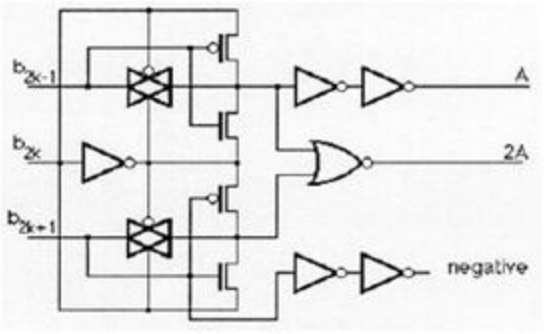


Figure-6.26: Booth encoder cell

To summarize the operation:

- Grouping multiplier bits into pairs
  - Orthogonal idea to the Booth recoding
  - Reduces the num of partial products to half
  - If Booth recoding not used → have to be able to multiply by 3 (hard: shift+add)
- Applying the grouping idea to Booth → Modified Booth Recoding (Encoding)
  - We already got rid of sequences of 1's → no multiplication by 3
  - Just negate, shift once or twice

6.5.4 Wallace Trees

For this purpose, Wallace trees were introduced. The addition time grows like the logarithm of the bit number. The simplest Wallace tree is the adder cell. More generally, an n-inputs Wallace tree is an n-input operator and log2(n) outputs, such that the value of the output word is equal to the number of "1" in the input word. The input bits and the least significant bit of the output have the same weight (Figure 6.27). An important property of Wallace trees is that they may be constructed using adder cells. Furthermore, the number of adder cells needed grows like the logarithm log2(n) of the number n of input bits. Consequently, Wallace trees are useful whenever a large number of operands are to add, like in multipliers. In a Braun or Baugh-Wooley multiplier with a Ripple Carry Adder, the completion time of the multiplication is proportional to twice the number n of bits. If the collection of the partial products is made through Wallace trees, the time for getting the result in a carry save notation should be proportional to log2(n).

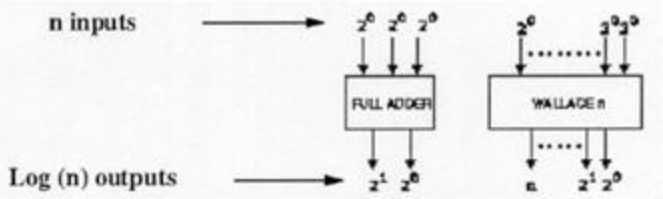


Figure-6.27: Wallace cells made of adders

Figure 6.28 represents a 7-inputs adder: for each weight, Wallace trees are used until there remain only two bits of each weight, as to add them using a classical 2-inputs adder. When taking into account the regularity of the interconnections, Wallace trees are the most irregular.

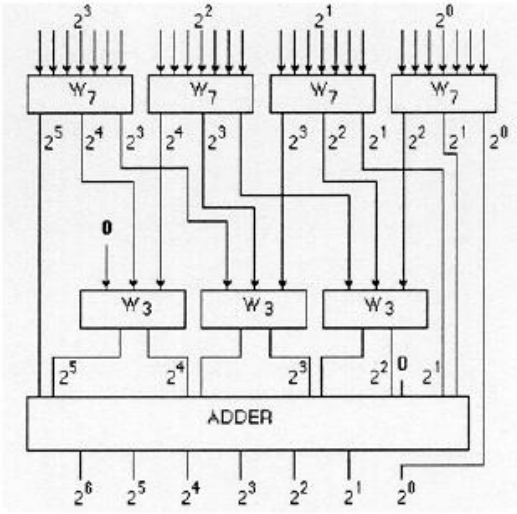


Figure-6.28: A 7-inputs Wallace tree

To summarize the operation:

The Wallace tree has three steps:

- Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, yielding n^2 results.
- Reduce the number of partial products to two by layers of full and half adders.
- Group the wires in two numbers, and add them with a conventional adder.

The second phase works as follows.

- Take any three wires with the same weights and input them into a full adder.
- The result will be an output wire of the same weight and an output wire with a higher weight for each three input wires.
- If there are two wires of the same weight left, input them into a half adder.
- If there is just one wire left, connect it to the next layer.

Recommended questions:

1. How to implement arithmetic and logic operation with a standard adder? Explain with the help of logic expression.
2. Discuss the architectural issues to be followed in the design of VLSI subsystem.
3. Design 4:1 mux using transmission gates.
4. How can 4 bit ALU architecture be used to implement an adder?
5. Explain the design steps for a 4 bit adder.
6. Discuss Baugh Worley method used for 2's complement multiplication.
7. Discuss timing constraints for both flip-flop and latches.
8. Explain booth multiplier with example.
9. Explain basic form of 2 phase clock generator.

Unit-7

Memory registers and clock

Timing considerations, memory elements, memory cell arrays.

Recommended readings:

1. Douglas A. Pucknell & Kamran Eshraghian, **"Basic VLSI Design"** PHI 3rd Edition (original Edition – 1994), 2005.

2. Neil H. E. Weste and K. Eshragian**," Principles of CMOS VLSI Design: A System Perspective,"** 2nd edition, Pearson Education (Asia) Pvt. Ltd., 2000. History of VLSI.

7.1 System timing considerations:

- Two phase non-overlapping clock
- ϕ_1 leads ϕ_2
- Bits to be stored are written to register and subsystems on ϕ_1
- Bits or data written are assumed to be settled before ϕ_2
- ϕ_2 signal used to refresh data
- Delays assumed to be less than the intervals between the leading edge of ϕ_1 & ϕ_2
- Bits or data may be read on the next ϕ_1
- There must be atleast one clocked storage element in series with every closed loop signal path

7.2 Storage / Memory Elements:

The elements that we will be studying are:

- Dynamic shift register
- 3T dynamic RAM cell
- 1T dynamic memory cell
- Pseudo static RAM / register cell
- 4T dynamic & 6T static memory cell
- JK FF circuit
- D FF circuit

Power dissipation

- static dissipation is very small
- dynamic power is significant
- dissipation can be reduced by alternate geometry

Volatility

- data storage time is limited to 1msec or less

7.2.1 3T dynamic RAM cell:

Circuit diagram

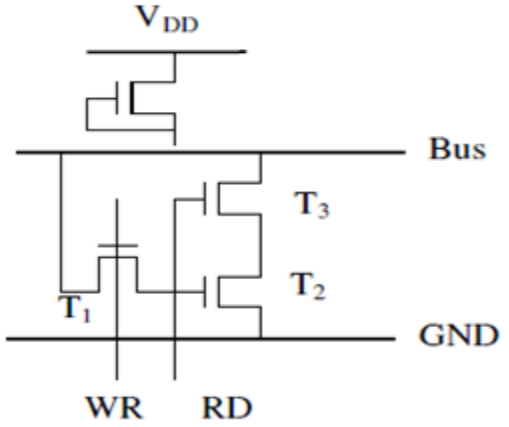


Figure 7.1: 3T Dynamic RAM Cell

Working

- RD = low, bit read from bus through T1, WR = high, logic level on bus sent to Cg of T2, WR = low again
- Bit level is stored in Cg of T2, RD=WR=low
- Stored bit is read by RD = high, bus will be pulled to ground if a 1 was stored else 0 if T2 non-conducting, bus will remain high.

Dissipation

- Static dissipation is nil
- Depends on bus pull-up & on duration of RD signal & switching frequency

Volatility

- Cell is dynamic, data will be there as long as charge remains on Cg of T2

7.2.2 1T dynamic memory cell:

Circuit diagram

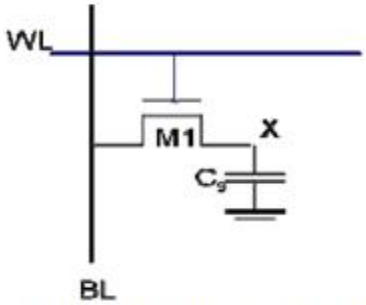


Figure 7.2: 1T Dynamic RAM Cell

Working

- Row select (RS) = high, during write from R/W line Cm is charged
- data is read from Cm by detecting the charge on Cm with RS = high
- cell arrangement is bit complex.
- solution: extend the diffusion area comprising source of pass transistor, but Cd<<< Cgchannel
- another solution : create significant capacitor using poly plate over diffusion area.
- Cm is formed as a 3-plate structure
- with all this careful design is necessary to achieve consistent readability

Dissipation

- no static power, but there must be an allowance for switching energy during read/write

7.2.3 Pseudo static RAM / register cell:

Circuit diagram

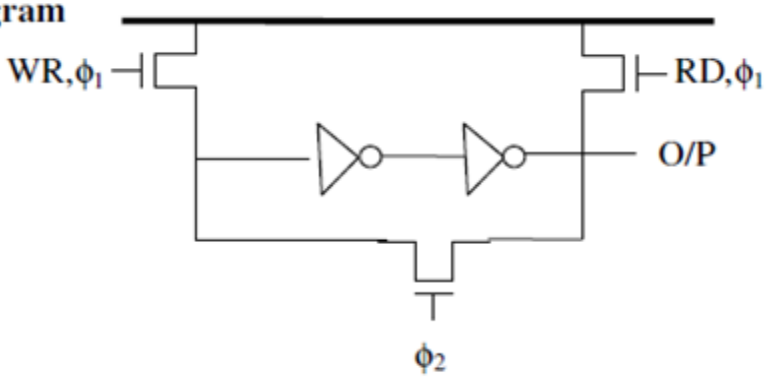


Figure 7.3: nMOS pseudo-static memory Cell

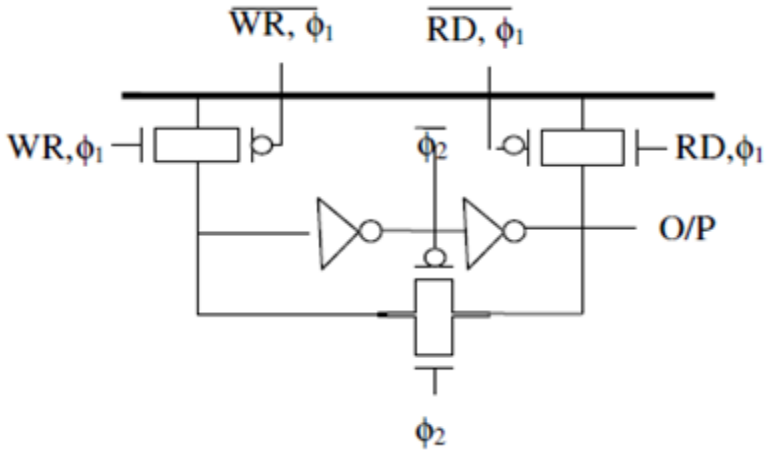


Figure 7.4: CMOS pseudo-static memory Cell

Working

- dynamic RAM need to be refreshed periodically and hence not convenient
- static RAM needs to be designed to hold data indefinitely
- One way is connect 2 inverter stages with a feedback.
- say ϕ_2 to refresh the data every clock cycle
- bit is written on activating the WR line which occurs with ϕ_1 of the clock
- bit on Cg of inverter 1 will produce complemented output at inverter 1 and true at output of inverter 2
- at every ϕ_2 , stored bit is refreshed through the gated feedback path
- stored bit is held till ϕ_2 of clock occurs at time less than the decay time of stored bit
- to read RD along with ϕ_1 is activated

Note:

- WR and RD must be mutually exclusive
- ϕ_2 is used for refreshing, hence no data to be read, if so charge sharing effect, leading to destruction of stored bit
- cells must be stackable, both side-by-side & top to bottom
- allow for other bus lines to run through the cell

7.2.4 4T dynamic & 6T static memory cell:

Circuit diagram

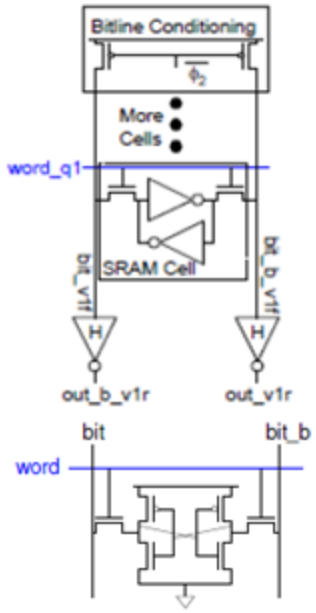


Figure 7.4: Dynamic and static memory cells

Working

- uses 2 buses per bit to store bit and bit'
- both buses are precharged to logic 1 before read or write operation.
- write operation
- read operation

Write operation

- both bit & bit' buses are precharged to VDD with clock ϕ_1 via transistor T5 & T6
- column select line is activated along with ϕ_2
- either bit or bit' line is discharged along the I/O line when carrying a logic 0
- row & column select signals are activated at the same time => bit line states are written in via T3 & T4, stored by T1 & T2 as charge

Read operation

- bit and bit' lines are again precharged to VDD via T5 & T6 during ϕ_1
- if 1 has been stored, T2 ON & T1 OFF
- bit' line will be discharged to VSS via T2
- each cell of RAM array be of minimum size & hence will be the transistors
- implies incapable of sinking large charges quickly

- RAM arrays usually employ some form of sense amplifier
  - T1, T2, T3 & T4 form as flip-flop circuit
  - if sense line to be inactive, state of the bit line reflects the charge present on gate capacitance of T1 & T3
  - current flowing from VDD through an on transistor helps to maintain the state of bit lines

Recommended questions:

1. Show the functioning of single transistor dynamic memory cell.
2. What are the system considerations?
3. What is structured design process?
4. Explain CMOS pseudo static D Flip flop.
5. Explain the working of 3TDRAM cell

Unit-8

Testability

Performance parameters, layout issues I/O pads, real estate, system delays, ground rules for design, test and testability.

Recommended readings:

1. Douglas A. Pucknell & Kamran Eshraghian, **"Basic VLSI Design"** PHI 3rd Edition (original Edition – 1994), 2005.

2. Neil H. E. Weste and K. Eshragian**," Principles of CMOS VLSI Design: A System Perspective,"** 2nd edition, Pearson Education (Asia) Pvt. Ltd., 2000. History of VLSI

8.1 Definition:

Design for testability (DFT) refers to those design techniques that make test generation and test application cost-effective.

Some terminologies:

Input / output (I/O) pads

- Protection of circuitry on chip from damage
- Care to be taken in handling all MOS circuits
- Provide necessary buffering between the environments On & OFF chip
- Provide for the connections of power supply
- Pads must be always placed around the peripheral

Minimum set of pads include:

- VDD connection pad
- GND(VSS) connection pad
- Input pad
- Output pad
- Bidirectional I/O pad

Designer must be aware of:

- nature of circuitry
- ratio/size of inverters/buffers on which output lines are connected
- how input lines pass through the pad circuit (pass transistor/transmission gate)

System delays

Buses:

- convenient concept in distributing data & control through a system
- bidirectional buses are convenient
- in design of datapath
- problems: capacitive load present
- largest capacitance
- sufficient time must be allowed to charge the total bus
- clock ϕ_1 & ϕ_2

Control paths, selectors & decoders

1. select registers and open pass transistors to connect cells to bus
2. Data propagation delay bus
3. Carry chain delay

8.2 Faults and Fault Modeling

A fault model is a model of how a physical or parametric fault manifests itself in the circuit Operation. Fault tests are derived based on these models

Physical Faults are caused due to the following reasons:

- Defect in silicon substrate
- Photolithographic defects
- Mask contamination and scratches
- Process variations and abnormalities
- Oxide defects

Physical faults cause Electrical and Logical faults

Logical Faults are:

- Single/multiple stuck-at *(most used)*
- CMOS stuck-open
- CMOS stuck-on
- AND / OR Bridging faults

Electrical faults are due to short, opens, transistor stuck on, stuck open, excessive steady state currents, resistive shorts and open.

8.3 Design for Testability

Two key concepts

- Observability
- Controllability

DFT often is associated with design modifications that provide improved access to internal circuit elements such that the local internal state can be controlled (controllability) and/or observed (observability) more easily. The design modifications can be strictly physical in nature (e.g., adding a physical probe point to a net) and/or add active circuit elements to facilitate controllability/observability (e.g., inserting a multiplexer into a net). While controllability and observability improvements for internal circuit elements definitely are important for test, they are not the only type of DFT

What can we do to increase testability?

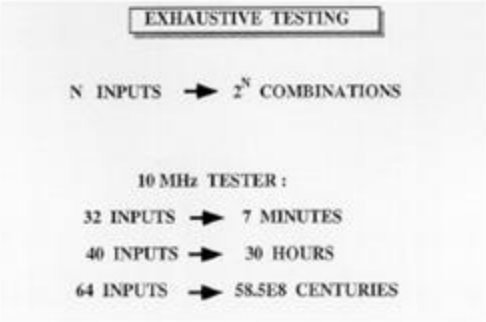
- increase observability
  - ⇨ add more pins (?!)
  - ⇨ add small "probe" bus, selectively enable different values onto bus
  - ⇨ use a hash function to "compress" a sequence of values (e.g., the values of a bus over many clock cycles) into a small number of bits for later read-out
  - ⇨ cheap read-out of all state information

- increase controllability
  - ⇨ use muxes to isolate submodules and select sources of test data as inputs
  - ⇨ provide easy setup of internal state

8.4 Testing combinational logic

The solution to the problem of testing a purely combinational logic block is a good set of patterns detecting "all" the possible faults.

The first idea to test an N input circuit would be to apply an N-bit counter to the inputs (controllability), then generate all the 2N combinations, and observe the outputs for checking (observability). This is called "exhaustive testing", and it is very efficient... but only for few- input circuits. When the input number increase, this technique becomes very time consuming.

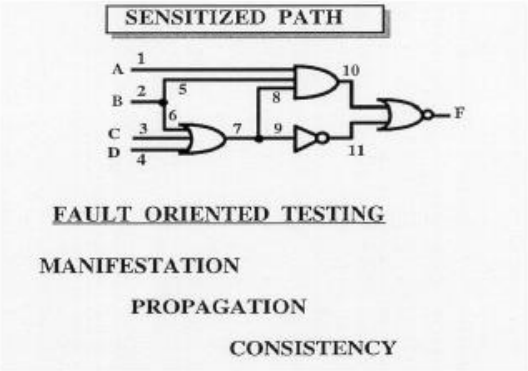


8.5 Sensitized Path Testing

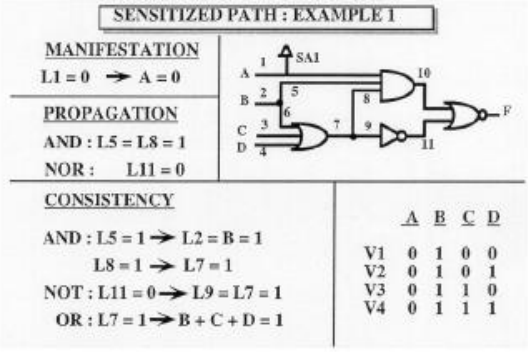
Most of the time, in exhaustive testing, many patterns do not occur during the application of the circuit. So instead of spending a huge amount of time searching for faults everywhere, the possible faults are first enumerated and a set of appropriate vectors are then generated. This is called "single-path sensitization" and it is based on "fault oriented testing".

The basic idea is to select a path from the site of a fault, through a sequence of gates leading to an output of the combinational logic under test. The process is composed of three steps :

- **Manifestation** : gate inputs, at the site of the fault, are specified as to generate the opposite value of the faulty value (0 for SA1, 1 for SA0).
- **Propagation** : inputs of the other gates are determined so as to propagate the fault signal along the specified path to the primary output of the circuit. This is done by setting these inputs to "1" for AND/NAND gates and "0" for OR/NOR gates.
- **Consistency** : or justification. This final step helps finding the primary input pattern that will realize all the necessary input values. This is done by tracing backward from the gate inputs to the primary inputs of the logic in order to receive the test patterns.



Example1 - SA1 of line1 (L1) : the aim is to find the vector(s) able to detect this fault.

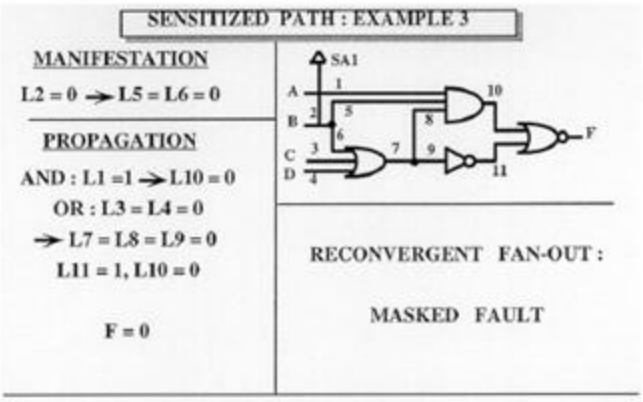


- **Manifestation:** L1 = 0 , then input A = 0. In a fault-free situation, the output F changes with A if B,C and D are fixed : for B,C and D fixed, L1 is SA1 gives F = 0, for instance, even if A = 0 (F = 1 for fault-free).
- **Propagation:** Through the AND-gate : L5 = L8 = 1, this condition is necessary for the propagation of the " L1 = 0 ". This leads to L10 = 0. Through the NOR-gate, and since L10 = 0, then L11 = 0, so the propagated manifestation can reach the primary output F. F is then read and compared with the fault-free value: F = 1.

- **Consistency:** From the AND-gate : L5=1, and then L2=B=1. Also L8=1, and then L7=1. Until now we found the values of A and B. When C and D are found, then the test vectors are generated, in the same manner, and ready to be applied to detect L1= SA1. From the NOT-gate, L11=0, so L9=L7=1 (coherency with L8=L7). From the OR-gate L7=1, and since L6=L2=B=1, so B+C+D=L7=1, then C and D can have either 1 or 0.

These three steps have led to four possible vectors detecting L1=SA1.

Example 2 - SA1 of line8 (L8) : The same combinational logic having one internal line SA1



- **Manifestation :** L8 = 0
- **Propagation:** Through the AND-gate: L5 = L1 = 1, then L10 = 0 Through the NOR-gate: we want to have L11 = 0, not to mask L10 = 0.
- **Consistency:** From the AND-gate L8 = 0 leads to L7 = 0. From the NOT-gate L11 = 0 means L9 = L7 = 1, L7 could not be set to 1 and 0 at the same time. This incompatibility could not be resolved in this case, and the fault "L8 SA1" remains undetectable.

8.6 D – Algorithm:

Given a circuit comprising combinational logic, the algorithm aims to find an assignment of input values that will allow detection of a particular internal fault by examining the output conditions. Using this algorithm the system can either be said as good or faulty. The existence of a fault in the faulty machine will cause a discrepancy between its behavior and that of the good machine for some particular values of inputs. The D-algorithm provides a systematic means of assigning input values for that particular design so that the discrepancy is driven to an output where it may be observed and thus detected. The algorithm is time-intensive and computing intensive for large circuits.

Practical design for test guidelines

Practical guidelines for testability should aim to facilitate test processes in three main ways:

- facilitate test generation
- facilitate test application
- avoid timing problems

These matters are discussed as below:

8.7 Improve Controllability and Observability

All "design for test" methods ensure that a design has enough observability and controllability to provide for a complete and efficient testing. When a node has difficult access from primary inputs or outputs (pads of the circuit), a very efficient method is to add internal pads acceding to this kind of node in order, for instance, to control block B2 and observe block B1 with a probe.

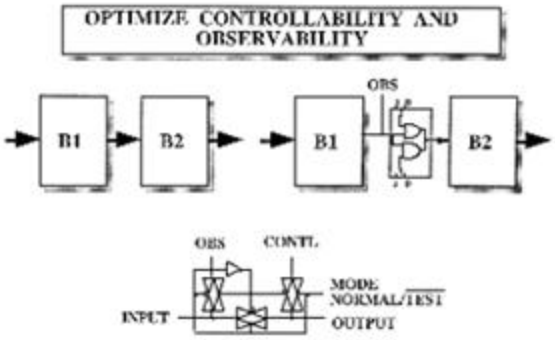


Figure 8.1 Improve Controllability and Observability

It is easy to observe block B1 by adding a pad just on its output, without breaking the link between the two blocks. The control of the block B2 means to set a 0 or a 1 to its input, and also to be transparent to the link B1-B2. The logic functions of this purpose are a NOR- gate, transparent to a zero, and a NAND-gate, transparent to a one. By this way the control of B2 is possible across these two gates.

Another implementation of this cell is based on pass-gates multiplexers performing the same function, but with less transistors than with the NAND and NOR gates (8 instead of 12).

The simple optimization of observation and control is not enough to guarantee a full testability of the blocks B1 and B2. This technique has to be completed with some other techniques of testing depending on the internal structures of blocks B1 and B2.

Use Multiplexers

This technique is an extension of the precedent, while multiplexers are used in case of limitation of primary inputs and outputs.

In this case the major penalties are extra devices and propagation delays due to multiplexers. Demultiplexers are also used to improve observability. Using multiplexers and demultiplexers allows internal access of blocks separately from each other, which is the basis of techniques based on partitioning or bypassing blocks to observe or control separately other blocks.

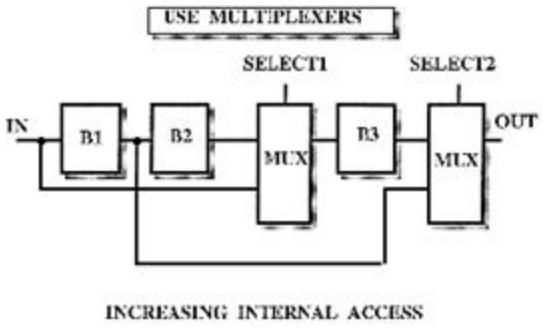


Figure 8.2: Use multiplexers

8.8 Partition Large Circuits

Partitioning large circuits into smaller sub-circuits reduces the test-generation effort. The test- generation effort for a general purpose circuit of n gates is assumed to be proportional to somewhere between n2 and n3. If the circuit is partitioned into two sub-circuits, then the amount of test generation effort is reduced correspondingly.

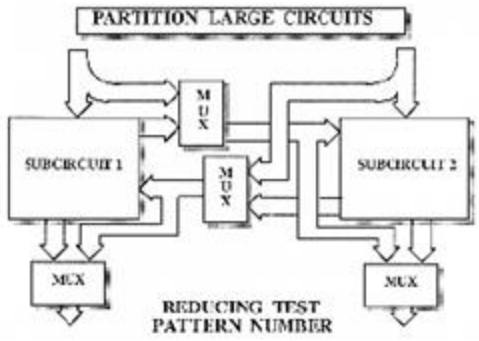


Figure 8.3: Partition Large Circuits

Logical partitioning of a circuit should be based on recognizable sub-functions and can be achieved physically by incorporating some facilities to isolate and control clock lines, reset lines and power supply lines. The multiplexers can be massively used to separate sub-circuits without changing the function of the global circuit.

Divide Long Counter Chains

Based on the same principle of partitioning, the counters are sequential elements that need a large number of vectors to be fully tested. The partitioning of a long counter corresponds to its division into sub-counters.

The full test of a 16-bit counter requires the application of 216 + 1 = 65537 clock pulses. If this counter is divided into two 8-bit counters, then each counter can be tested separately, and the total test time is reduced 128 times (27). This is also useful if there are subsequent requirements to set the counter to a particular count for tests associated with other parts of the circuit: pre-loading facilities.

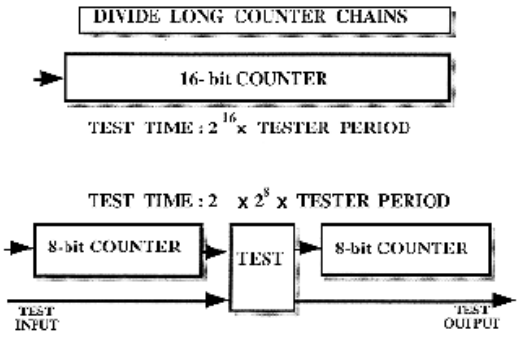


Figure 8.4: Divide Long Counter Chains

Initialize Sequential Logic

One of the most important problems in sequential logic testing occurs at the time of power-on, where the first state is random if there were no initialization. In this case it is impossible to start a test sequence correctly, because of memory effects of the sequential elements.

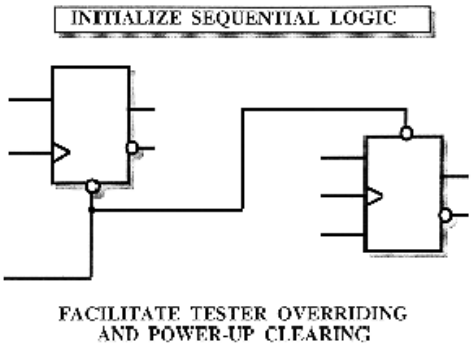


Figure 8.5: Initialize Sequential Logic

The solution is to provide flip-flops or latches with a set or reset input, and then to use them so that the test sequence would start with a known state.

Ideally, all memory elements should be able to be set to a known state, but practically this could be very surface consuming, also it is not always necessary to initialize all the sequential logic. For example, a serial-in serial-out counter could have its first flip-flop provided with an initialization, then after a few clock pulses the counter is in a known state.

Overriding of the tester is necessary some times, and requires the addition of gates before a Set or a Reset so the tester can override the initialization state of the logic.

8.9 Avoid Asynchronous Logic

Asynchronous logic uses memory elements in which state-transitions are controlled by the sequence of changes on the primary inputs. There is thus no way to determine easily when the next state will be established. This is again a problem of timing and memory effects.

Asynchronous logic is faster than synchronous logic, since the speed in asynchronous logic is only limited by gate propagation delays and interconnects. The design of asynchronous logic is then more difficult than synchronous (clocked) logic and must be carried out with due regards to the possibility of critical races (circuit behavior depending on two inputs changing simultaneously) and hazards (occurrence of a momentary value opposite to the expected value).

Non-deterministic behavior in asynchronous logic can cause problems during fault simulation. Time dependency of operation can make testing very difficult, since it is sensitive to tester signal skew.

8.10 Avoid Logical Redundancy

Logical redundancy exists either to mask a static-hazard condition, or unintentionally (design bug). In both cases, with a logically redundant node it is not possible to make a primary output value dependent on the value of the redundant node. This means that certain fault conditions on the node cannot be detected, such as a node SA1 of the function F.

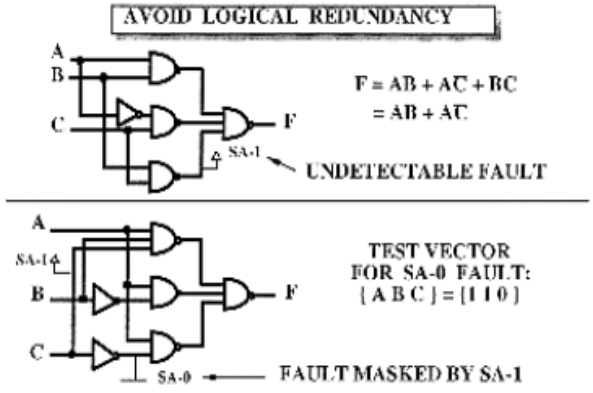


Figure 8.6: Avoid Logical Redundancy

Another inconvenience of logical redundancy is the possibility for a non-detectable fault on a redundant node to mask the detection of a fault normally-detectable, such a SA0 of input C in the second example, masked by a SA1 of a redundant node.

8.11 Avoid Delay Dependent Logic

Automatic test pattern generators work in logic domains, they view delay dependent logic as redundant combinational logic. In this case the ATPG will see an AND of a signal with its complement, and will therefore always compute a 0 on the output of the AND-gate (instead of a pulse). Adding an OR-gate after the AND-gate output permits to the ATPG to substitute a clock signal directly.

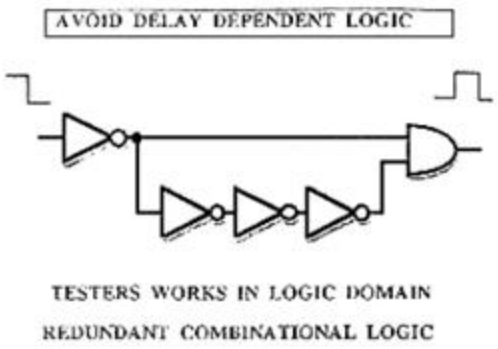


Figure 8.7: Avoid Delay Dependent Logic

8.12 Avoid Clock Gating

When a clock signal is gated with any data signal, for example a load signal coming from a tester, a skew or any other hazard on that signal can cause an error on the output of logic.

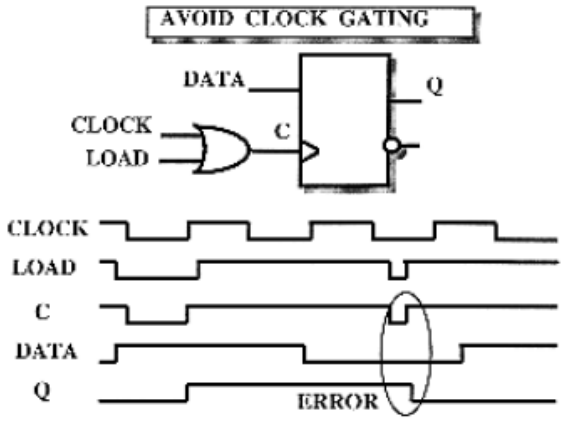


Figure 8.8: Avoid Clock Gating

This is also due to asynchronous type of logic. Clock signals should be distributed in the circuit with respect to synchronous logic structure.

8.13 Distinguish Between Signal and Clock

This is another timing situation to avoid, in which the tester could not be synchronized if one clock or more are dependent on asynchronous delays (across D-input of flip-flops, for example).

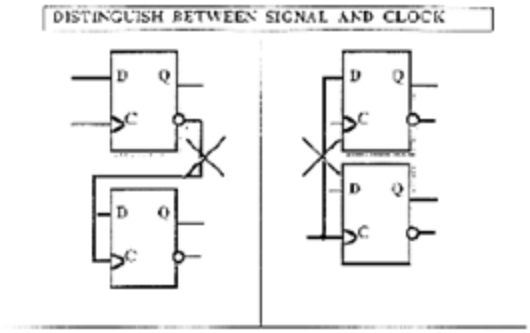


Figure 8.9: Distinguish Between Signal and Clock

8.14 Avoid Self Resetting Logic

The self resetting logic is more related to asynchronous logic, since a reset input is independent of clock signal.

Before the delayed reset, the tester reads the set value and continues the normal operation. If a reset has occurred before tester observation, then the read value is erroneous. The solution to this problem is to allow the tester to override by adding an OR-gate, for example, with an inhibition input coming from the tester. By this way the right response is given to the tester at the right time.

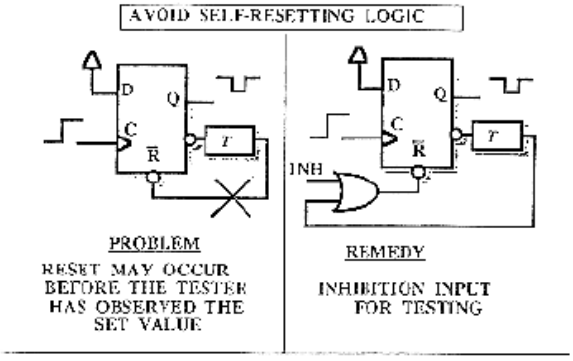


Figure 8.10: Avoid Self Resetting Logic

Use Bused Structure

This approach is related, by structure, to partitioning technique. It is very useful for microprocessor-like circuits. Using this structure allows the external tester the access of three buses, which go to many different modules.

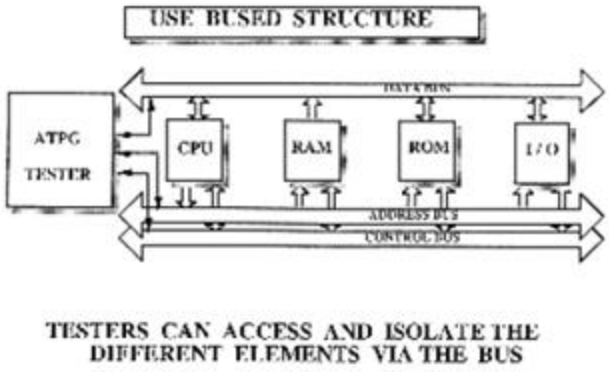


Figure 8.11: Use Bused Structure

The tester can then disconnect any module from the buses by putting its output into a high- impedance state. Test patterns can then be applied to each module separately.

8.2 Separate Analog and Digital Circuits

Testing analog circuit requires a completely different strategy than for digital circuit. Also the sharp edges of digital signals can cause cross-talk problem to the analog lines, if they are close to each other.

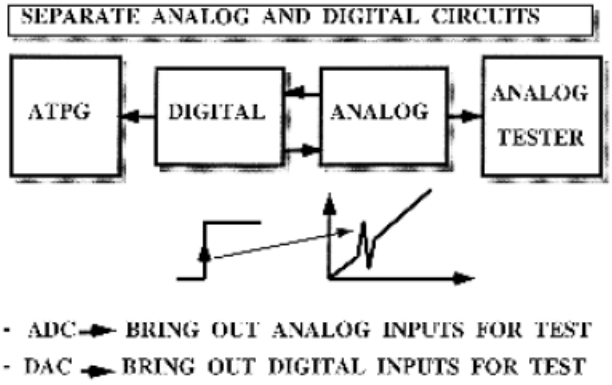


Figure 8.12: Separate Analog and Digital Circuits

If it is necessary to route digital signals near analog lines, then the digital lines should be properly balanced and shielded. Also, in the cases of circuits like Analog-Digital converters, it is better to bring out analog signals for observation before conversion. For Digital-Analog converters, digital signals are to be brought out also for observation before conversion.

8.3 Ad-Hoc DFT Method

- Good design practices learnt through experience are used as guidelines:
  - Avoid asynchronous (unclocked) feedback.
  - Make flip-flops initializable.
  - Avoid redundant gates. Avoid large fan-in gates.
  - Provide test control for difficult-to-control signals.
  - Avoid gated clocks.
  - Avoid delay dependant logic.
  - Avoid parallel drivers.
  - Avoid monostable and self-resetting logic.
- Design Reviews
  - Manual analysis
    - Conducted by experts
  - Programmed analysis
    - Using design auditing tools
  - Programmed enforcement
    - Must use certain design practices and cell types.

Objective: Adherence to design guidelines and testability improvement techniques with little impact on performance and area.

- Disadvantages of ad-hoc DFT methods:
  - Experts and tools not always available.
  - Test generation is often manual with no guarantee of high fault coverage.
  - Design iterations may be necessary.

Scan Design Techniques

The set of design for testability guidelines presented above is a set of ad hoc methods to design random logic in respect with testability requirements. The scan design techniques are a set of structured approaches to design (for testability) the sequential circuits.

The major difficulty in testing sequential circuits is determining the internal state of the circuit. Scan design techniques are directed at improving the controllability and observability of the internal states of a sequential circuit. By this the problem of testing a sequential circuit is reduced to that of testing a combinational circuit, since the internal states of the circuit are under control.

8.4 Scan Path

The goal of the scan path technique is to reconfigure a sequential circuit, for the purpose of testing, into a combinational circuit. Since a sequential circuit is based on a combinational circuit and some storage elements, the technique of scan path consists in connecting together all the storage elements to form a long serial shift register. Thus the internal state of the circuit can be observed and controlled by shifting (scanning) out the contents of the storage elements. The shift register is then called a scan path.

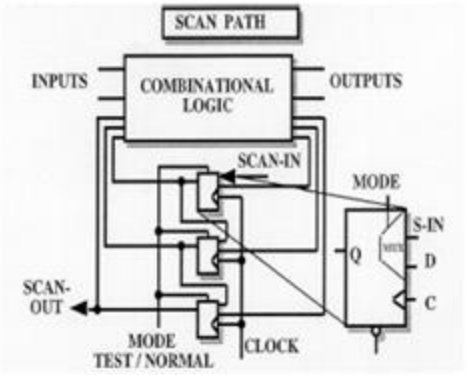


Figure 8.13: Scan Path

The storage elements can either be D, J-K, or R-S types of flip-flops, but simple latches cannot be used in scan path. However, the structure of storage elements is slightly different than classical ones. Generally the selection of the input source is achieved using a multiplexer on the data input controlled by an external mode signal. This multiplexer is integrated into the D-flip-flop, in our case; the D-flip-flop is then called MD-flip-flop (multiplexed-flip-flop).

The sequential circuit containing a scan path has two modes of operation: a normal mode and a test mode which configure the storage elements in the scan path.

As analyzed from figure 8.13, in the normal mode, the storage elements are connected to the combinational circuit, in the loops of the global sequential circuit, which is considered then as a finite state machine.

In the test mode, the loops are broken and the storage elements are connected together as a serial shift register (scan path), receiving the same clock signal. The input of the scan path is called scan-in and the output scan-out. Several scan paths can be implemented in one same complex circuit if it is necessary, though having several scan-in inputs and scan-out outputs.

A large sequential circuit can be partitioned into sub-circuits, containing combinational sub-circuits, associated with one scan path each. Efficiency of the test pattern generation for a combinational sub-circuit is greatly improved by partitioning, since its depth is reduced.

Before applying test patterns, the shift register itself has to be verified by shifting in all ones i.e. 111...11, or zeros i.e. 000...00, and comparing.

The method of testing a circuit with the scan path is as follows:

1. Set test mode signal, flip-flops accept data from input scan-in
2. Verify the scan path by shifting in and out test data
3. Set the shift register to an initial state
4. Apply a test pattern to the primary inputs of the circuit
5. Set normal mode, the circuit settles and can monitor the primary outputs of the circuit
6. Activate the circuit clock for one cycle
7. Return to test mode
8. Scan out the contents of the registers, simultaneously scan in the next pattern

8.5 Level sensitivity scan design (LSSD)

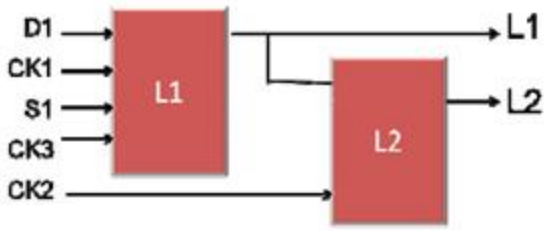


Figure 8.14: Level sensitivity scan design

The level-sensitive aspect means that the sequential network is designed so that when an input change occurs, the response is independent of the component and wiring delays within the network (Figure 8.14).

The scan path aspect is due to the use of shift register latches (SRL) employed as storage elements. In the test mode they are connected as a long serial shidt register. Each SRL has a specific design similar to a master-slave FF. it is driven by two non-overlapping clocks which can be controlled readily from the primary inputs to the circuit. Input D1 is the normal data input to the SRL; clocks CK1 and CK2 control the normal operation of the SRL while clocks CK3 and CK2 control scan path movements through the SRL. The SRL output is derived at L2 in both modes of operation, the mode depending on which clocks are activated.

Advantages:

- Circuit operation is independent of dynamic characteristics of the logic elements
- ATP generation is simplified
- Eliminate hazards and races
- Simplifies test generation and fault simulation

8.6 Boundary Scan Test (BST)

Boundary Scan Test (BST) is a technique involving scan path and self-testing techniques to resolve the problem of testing boards carrying VLSI integrated circuits and/or surface mounted devices (SMD).

Printed circuit boards (PCB) are becoming very dense and complex, especially with SMD circuits, that most test equipment cannot guarantee good fault coverage.

BST (figure 8.15) consists in placing a scan path (shift register) adjacent to each component pin and to interconnect the cells in order to form a chain around the border of the circuit. The BST circuits contained on one board are then connected together to form a single path through the board.

The boundary scan path is provided with serial input and output pads and appropriate clock pads which make it possible to:

- Test the interconnections between the various chip
- Deliver test data to the chips on board for self-testing
- Test the chips themselves with internal self-test

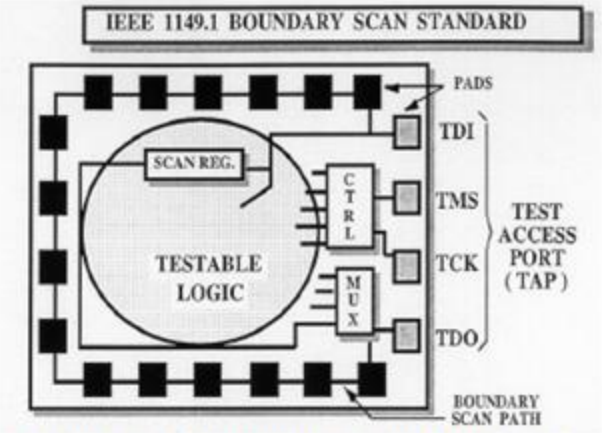


Figure 8.15: Boundary Scan Test (BST)

The advantages of Boundary scan techniques are as follows :

- No need for complex testers in PCB testing
- Test engineers work is simplified and more efficient
- Time to spend on test pattern generation and application is reduced
- Fault coverage is greatly increased.

8.7 Other scan techniques:

Partial Scan Method

- A subset of flip-flops is scanned.
- Objectives:
  - Minimize area overhead and scan sequence length, yet achieve required fault coverage
  - Exclude selected flip-flops from scan:
    - Improve performance
    - Allow limited scan design rule violations
  - Allow automation:
    - In scan flip-flop selection
    - In test generation
  - Shorter scan sequences – reduce application time

Random Access Scan Method

- The scan function is implemented like a random-access memory (RAM)
- All flip-flops form a RAM in scan mode
- A subset of flip-flops can be included in the RAM if partial scan is desired
- In scan mode, any flip-flop can be read or written

Procedure:

- Set test inputs to all test points
- Apply the master reset signal to initialize all memory elements
- Set scan-in address & data, then apply the scan clock
- Repeat the above step until all internal test inputs are scanned
- Clock once for normal operation
- Check states of the output points
- Read the scan-out states of all memory elements by applying the address

8.8 Built-in-self test

Objectives:

1. To reduce test pattern generation cost
2. To reduce volume of test data
3. To reduce test time

Built-in Self Test, or BIST, is the technique of designing additional hardware and software features into integrated circuits to allow them to perform self-testing, i.e., testing of their own operation (functionally, parametrically, or both) using their own circuits, thereby reducing dependence on an external automated test equipment (ATE).

BIST is a Design-for-Testability (DFT) technique, because it makes the electrical testing of a chip easier, faster, more efficient, and less costly. The concept of BIST is applicable to just about any kind of circuit, so its implementation can vary as widely as the product diversity that it caters to. As an example, a common BIST approach for DRAM's includes the incorporation onto the chip of additional circuits for pattern generation, timing, mode selection, and go-/no-go diagnostic tests.

Advantages of implementing BIST include:

1) Lower cost of test, since the need for external electrical testing using an ATE will be reduced, if not eliminated
2) Better fault coverage, since special test structures can be incorporated onto the chips
3) Shorter test times if the BIST can be designed to test more structures in parallel
4) Easier customer support and
5) Capability to perform tests outside the production electrical testing environment. The last advantage mentioned can actually allow the consumers themselves to test the chips prior to mounting or even after these are in the application boards.

Disadvantages of implementing BIST include:

1) Additional silicon area and fab processing requirements for the BIST circuits
2) Reduced access times
3) Additional pin (and possibly bigger package size) requirements, since the BIST circuitry need a way to interface with the outside world to be effective and
4) Possible issues with the correctness of BIST results, since the on-chip testing hardware itself can fail.

Techniques are:

- compact test: signature analysis
- linear feedback shift register
- BILBO
- self checking technique

Compact Test: Signature analysis

Signature analysis performs polynomial division that is, division of the data out of the device under test (DUT). This data is represented as a polynomial P(x) which is divided by a characteristic polynomial C(x) to give the signature R(x), so that

$$R(x) = P(x)/C(x)$$

This is summarized as in figure 8.16.

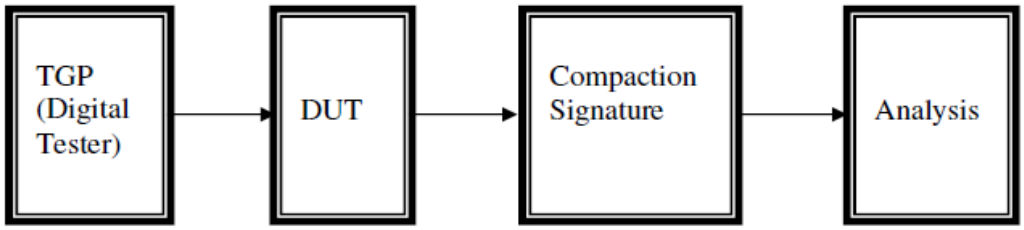


Figure 8.16: BIST – signature analysis

8.9 Linear feedback shift register (LFSR):

An LFSR is a shift register that, when clocked, advances the signal through the register from one bit to the next most-significant bit. Some of the outputs are combined in exclusive-OR configuration to form a feedback mechanism. A linear feedback shift register can be formed by performing exclusive-OR (Figure 8.16) on the outputs of two or more of the flip-flops together and feeding those outputs back into the input of one of the flip-flops.

LFSR technique can be applied in a number of ways, including random number generation, polynomial division for signature analysis, and n-bit counting. LFSR can be series or parallel, the differences being in the operating speed and in the area of silicon occupied; Parallel LFSR being faster but larger than serial LFSR.

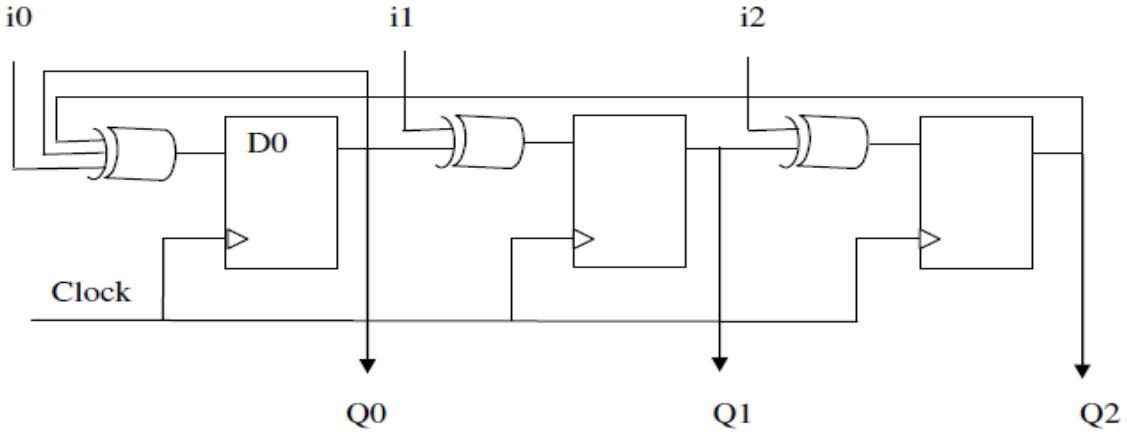


Figure 8.16: Linear feedback shift register

8.10 Built-in logic block observer (BILBO):

BILBO is a built-in test generation scheme which uses signature analysis in conjunction with a scan path. The major component of a BILBO is an LFSR with a few gates (Figure 8.17).

A **BILBO** register (**built-in logic block observer**) combines normal flipflops with a few additional gates to provide four different functions. The example circuit shown in the applet realizes a four-bit register. However, the generalization to larger bit-widths should be obvious, with the XOR gates in the LFSR feedback path chosen to implement a good polynomial for the given bit-width.

When the **A** and **B** control inputs are both 1, the circuit functions as a normal parallel D-type register.

When both **A** and **B** inputs are 0, the D-inputs are ignored (due to the AND gate connected to A), but the flipflops are connected as a shift-register via the NOR and XOR gates. The input to the first flipflop is then selected via the multiplexer controlled by the **S** input. If the S input is 1, the multiplexer transmits the value of the external **SIN** shift-in input to the first flipflop, so that the BILBO register works as a normal shift-register. This allows to initialize the register contents using a single signal wire, e.g. from an external test controller.

If all of the **A**, **B**, and **S** inputs are 0, the flipflops are configured as a shift-register, again, but the input bit to the first flipflop is computed by the XOR gates in the LFSR feedback path. This means that the register works as a standard LFSR pseudorandom pattern generator, useful to drive the logic connected to the Q outputs. Note that the start value of the LFSR sequence can be set by shifting it in via the SIN input.

Finally, if **B** and **S** are 0 but **A** is 1, the flipflops are configured as a shift-register, but the input value of each flipflop is the XOR of the D-input and the Q-output of the previous flipflop. This is exactly the configuration of a standard LFSR signature analysis register.

Because a BILBO register can be used as a pattern generator for the block it drives, as well provide signature-analysis for the block it is driven by, a whole circuit can be made self-testable with very low overhead and with only minimal performance degradation (two extra gates before the D inputs of the flipflops).

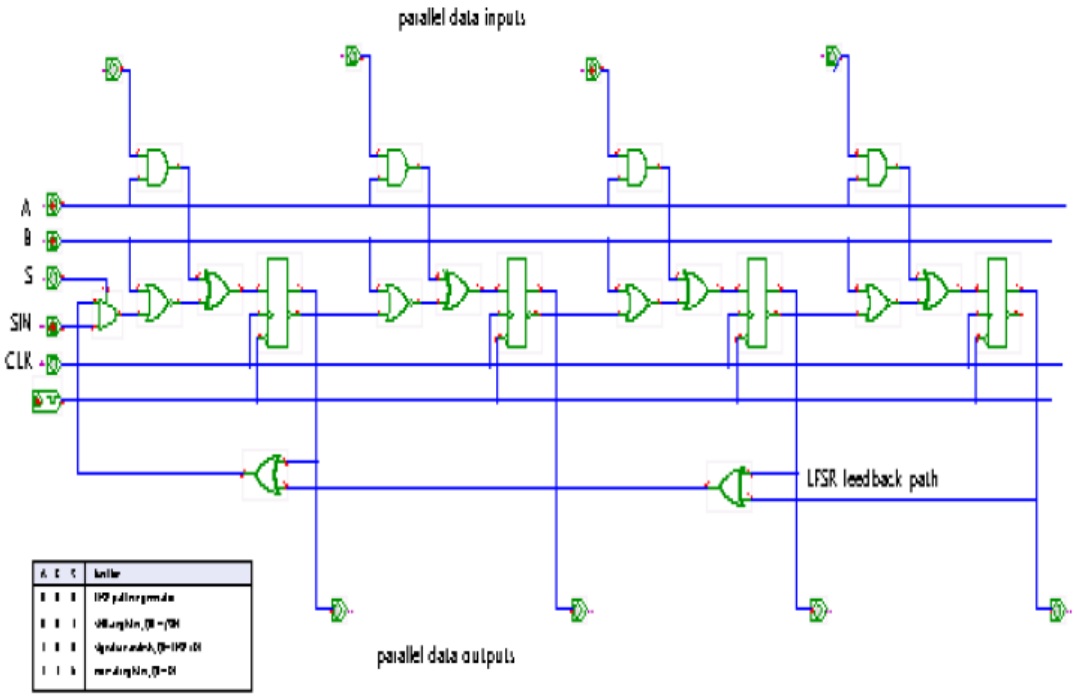


Figure 8.17: BIST – BILBO

8.11 Self-checking techniques:

It consists of logic block and checkers should then obey a set of rules in which the logic block is 'strongly fault secure' and the checker 'strongly code disjoint'. The code use in data encoding depends on the type of errors that may occur at the logic block output. In general three types are possible:

- Simple error: one bit only affected at a time.
- Unidirectional error: multiple bits at 1 instead of 0 (or 0 instead of 1)
- Multiple errors: multiple bits affected in any order.

Self-checking techniques are applied to circuits in which security is important so that fault tolerance is of major interest. Such technique will occupy more area in silicon than classical techniques such as functional testing but provide very high test coverage.

Recommended questions:

1. Define testability.
2. With an example define performance parameters.
3. Briefly explain the layout issues while design a circuit in vlsi.
4. Explain I/O pads.
5. Explain types of I/O pads.
6. What do you mean by real estate in the field of vlsi design.
7. What do you mean by delays.
8. Explain system delays.
9. What is need of providing delays to the system.
10. Write a short note on test and testability.