UNIT I - MOS TRANSISTOR PRINCIPLES

MOS logic families (NMOS and CMOS), Ideal and Non Ideal IV Characteristics, CMOS devices, MOS(FET) Transistor Characteristic under Static and Dynamic Conditions, Technology Scaling, power consumption

1.1: INTRODUCTION: (VLSI)

✓ In 1958, Jack Kilby built the first integrated circuit flip-flop at Texas Instruments.

✓ Bell Labs developed the bipolar junction transistor. Bipolar transistors were more reliable, less noisy and more power-efficient.

✓ In 1960s, Metal Oxide Semiconductor Field Effect Transistors (MOSFETs) began to enter in the production.

✓ MOSFETs offer the compelling advantage that; they draw almost zero control current while idle.

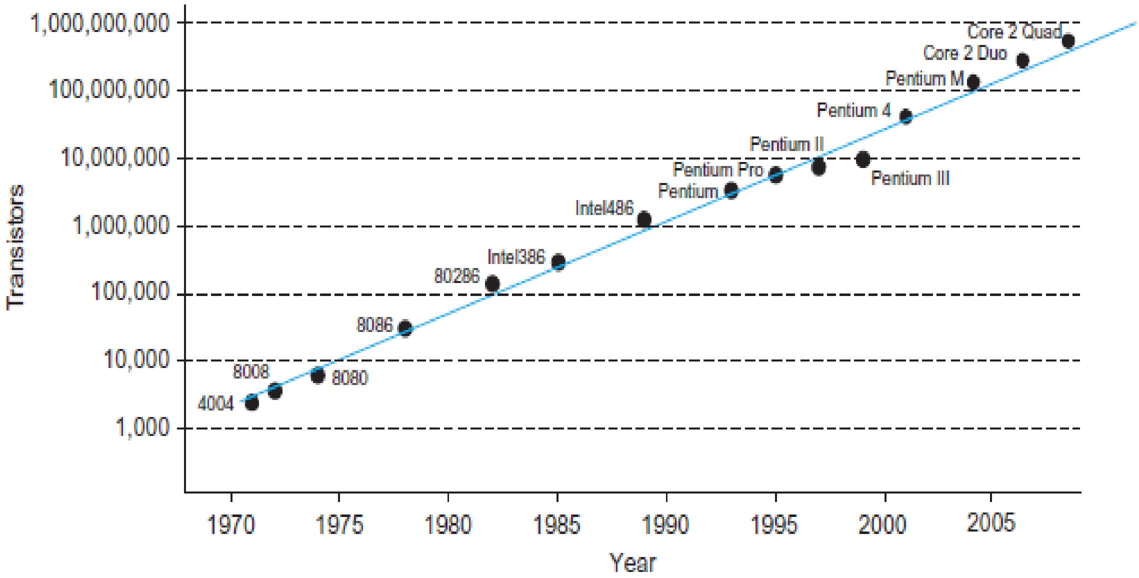
✓ They come in two flavors: nMOS and pMOS, using n-type and p-type silicon respectively.

✓ In 1963, Frank Wanlass at Fairchild described the first logic gates using MOSFETs. Fairchild's gates used both nMOS and pMOS transistors, naming as Complementary Metal Oxide Semiconductor (CMOS).

✓ Power consumption became a major issue in the 1980s as hundreds of thousands of transistors were integrated onto a single die.

✓ CMOS processes were widely adopted and replaced nMOS and bipolar processes for all digital logic applications.

✓ In 1965, Gordon Moore observed that plotting the number of transistors that can be most economically manufactured on a chip gives a straight line on a semi logarithmic scale.



○ ***Moore's Law*** **is defined as transistor count doubling every 18 months.**

The level of integration of chips is classified as

- Small Scale Integration (SSI)
- Medium Scale Integration (MSI)
- Large Scale Integration (LSI)

- Very Large Scale Integration (VLSI)
- Ultra Large Scale Integration (ULSI)

Small scale Integration:

- ✓ *Small-Scale Integration* (SSI) circuits have less than 10 gates. Example: 7404 inverter.

Medium scale Integration:

- ✓ *Medium-Scale Integration* (MSI) circuits have up to 1000 gates. Example: 74161 counter.

Large scale Integration:

- ✓ *Large-Scale Integration* (LSI) circuits have up to 10,000 gates. Example: 8-bit microprocessor (8085).

Very large scale Integration:

- ✓ Very large scale Integration (VLSI) with gates counting up to lakhs. Example: 16-bit microprocessor (8086).
- ✓ The feature size of a CMOS manufacturing process refers to the minimum dimension of a transistor that can be reliably built.

Ultra large scale Integration:

- ✓ Ultra Large-Scale Integration (ULSI) is the process of integrating millions of transistors on a single silicon semiconductor microchip.

****

1.2: MOS Transistor

nMOS and pMOS transistor

Explain the basic concept of nMOS and pMOS transistor with relevant symbol.

- ✓ A Metal-Oxide-Semiconductor (MOS) structure is created by superimposing layers of conducting and insulating materials.
- ✓ CMOS technology provides two types of transistors. They are n-type transistor (nMOS) and p-type transistor (pMOS).
- ✓ As transistor operation is controlled by electric fields, the devices are also called Metal Oxide Semiconductor Field Effect Transistors (MOSFETs).
- ✓ The transistor consists of a stack of the conducting gate, an insulating layer of silicon dioxide (SiO2) and the silicon wafer, also called as substrate, body or bulk.
- ✓ A pMOS transistor consists of p-type source and drain region with an n-type body.
- ✓ An nMOS transistor consists of n-type source and drain region with a p-type body.

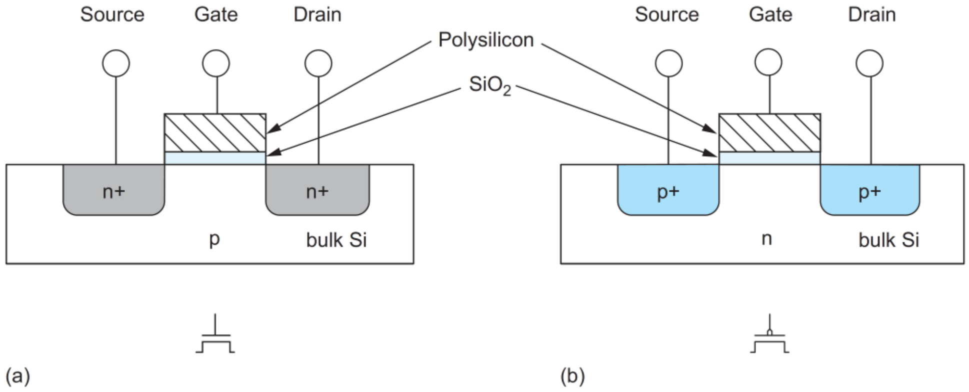


Figure 1: (a) n-MOS transistor (b) p-MOS transistor

nMOS Transistor:

- ✓ In an nMOS transistor, the body is grounded and the p–n junction of the source and drain to body are reverse-biased.
- ✓ As the gate is grounded, no current flows through junction. Hence, the transistor is OFF.
- ✓ If the gate voltage is raised, it creates an electric field, that start to attract free electrons to the underside of the Si–SiO_2 interface.
- ✓ If the voltage is raised more, a thin region under the gate called the channel is inverted.
- ✓ Since a conducting path of electron carriers is formed from source to drain, current starts to flow. So, the transistor is said to be ON.

pMOS Transistor:

- ✓ For a pMOS transistor, the body is held at a positive voltage.
- ✓ When the gate terminal has a positive voltage, the source and drain junctions are reverse-biased and no current flows. So, the transistor is said to OFF.
- ✓ When the gate voltage is lowered, positive charges are attracted to the underside of the Si–SiO2 interface.
- ✓ When a sufficient low gate voltage is applied, the channel inverts and a conducting path of positive carriers is formed from source to drain, which makes the transistor ON.

NOTE:

- ✓ The symbol for the pMOS transistor has a bubble on the gate, indicating that the transistor behavior is opposite to nMOS.
- ✓ When the gate of an nMOS transistor is 1, the transistor is ON. When the gate is 0, the nMOS transistor is OFF.
- ✓ A pMOS transistor is ON when the gate is low(0) and OFF when the gate is high(1).

1.3: Modes of MOS TRANSISTOR

Explain the accumulation (Enhancement) mode, depletion layer and inversion layer of MOS transistor with diagram.

- ✓ The MOS transistor is a majority-carrier device, in which the current in a conducting channel is controlled by gate voltage.
- ✓ In an nMOS transistor, the majority carriers are electrons.
- ✓ In a pMOS transistor, the majority carriers are holes.
- ✓ Figure 2 shows a simple MOS structure. The top layer of the structure is a good conductor called the gate.
- ✓ Transistor gate is polysilicon, i.e., silicon formed from many small crystals. The middle layer is a very thin insulating film of SiO_2, called the gate oxide. The bottom layer is the doped silicon body.
- ✓ The figure 2 shows a p-type body, in which the carriers are holes. The body is grounded and voltage is applied to the gate.
- ✓ The gate oxide is a good insulator, so almost zero current flows from the gate to the body.

Accumulation (Enhancement) mode:

- ✓ In Figure 2(a), when a negative voltage is applied to the gate, negative charges are formed on the gate.

- ✓ The positively charged holes are attracted to the region under the gate. This is called the accumulation mode.

Depletion mode:

- ✓ In Figure 2(b), when a small positive voltage is applied to the gate, positive charges are formed on the gate.
- ✓ The holes in the body are repelled from the region directly under the gate, resulting in a depletion region forming below the gate.

Inversion layer:

- ✓ In Figure 2(c), when a higher positive potential greater than threshold voltage (V_t) is applied, more positive charges are attracted to the gate.
- ✓ The holes are repelled and some free electrons in the body are attracted to the region under the gate. This conductive layer of electrons in the p-type body is called the inversion layer.
- ✓ The threshold voltage depends on the number of dopants in the body and the thickness t_{ox} of the oxide.

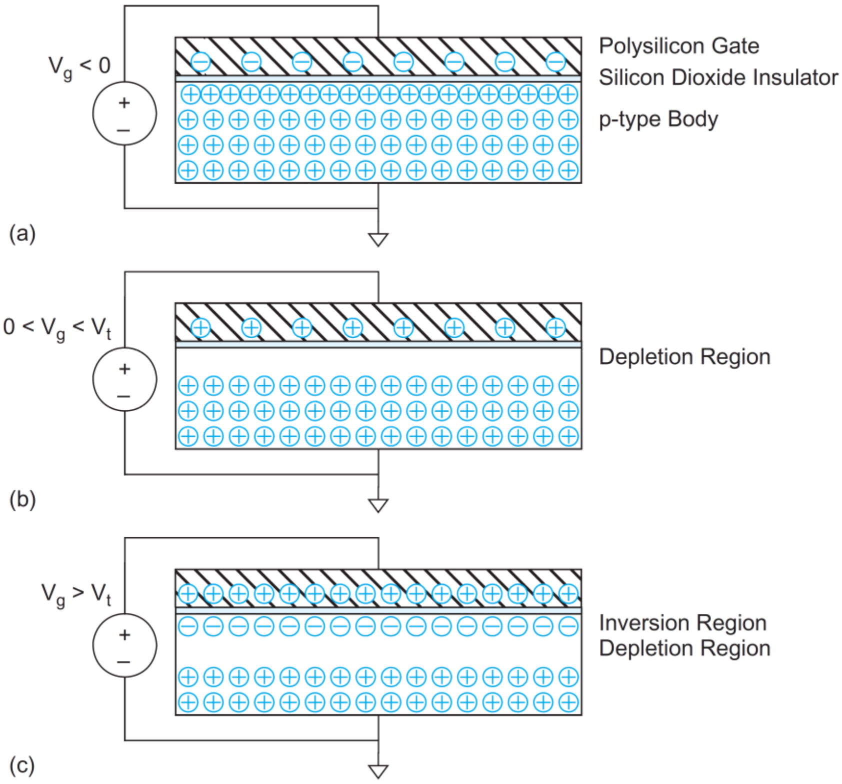


Figure 2: MOS structure demonstrating (a) accumulation, (b) depletion, and (c) inversion layer

1.4: Operating regions of MOS transistor:

Draw the small signal model of device during cut-off, linear and saturation. (April 2018)
Discuss the cutoff, linear and saturation region operation of MOS transistor. (Nov 2009)

- ✓ The MOS transistor operates in cutoff region, linear region and saturation region.

Cutoff region:

- ✓ In Figure 3(a), the gate-to-source voltage (V_{gs}) is less than the threshold voltage (V_t) and source is grounded.
- ✓ Junctions between the body and the source or drain are reverse biased, so no current flows. Thus, the transistor is said to be OFF and this mode of operation is called cutoff.
- ✓ If $V_{gs} < V_t$, the transistor is cutoff (OFF).

Linear (Active) Region:

- ✓ In Figure 3(b), the gate voltage is greater than the threshold voltage.
- ✓ An inversion region of electrons, called the channel connects the source and drain, creating a conductive path and making the transistor ON.
- ✓ If $V_{gs} > V_t$, the transistor turns ON. If V_{ds} is small, the transistor acts as a linear resistor, in which the current flow is proportional to V_{ds}.
- ✓ The number of carriers and the conductivity increases, with the gate voltage.

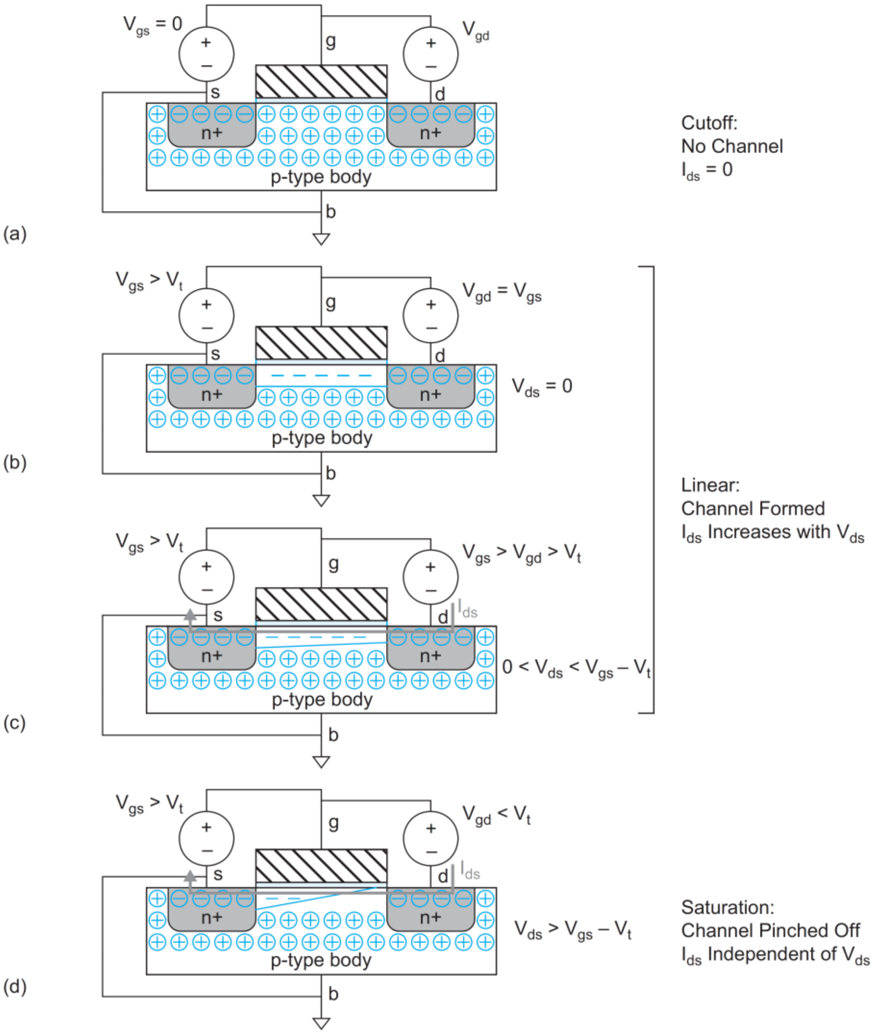


Figure 3: nMOS transistor demonstrating cutoff, linear, and saturation regions of operation

- ✓ The voltage between drain and source is $V_{ds} = V_{gs} - V_{gd}$. If $V_{ds} = 0$ (i.e., $V_{gs} = V_{gd}$), there is no electric field to push current from drain to source.
- ✓ When a small positive voltage V_{ds} is applied to the drain (Figure 3(c)), current I_{ds} flows through the channel from drain to source.
- ✓ This mode of operation is termed as linear, resistive, triode, nonsaturated, or unsaturated.

Saturation region:

- ✓ The current increases with increase in both the drain voltage and gate voltage.

- ✓ If V_{ds} becomes sufficiently large that $V_{gd} < V_t$, the channel is no longer inverted near the drain and becomes pinched off (Figure 3(d)).
- ✓ As electrons reach the end of the channel, they are injected into the depletion region near the drain and accelerated toward the drain.
- ✓ Above this drain voltage, current I_{ds} are controlled only by the gate voltage. This mode is called saturation.
- ✓ If $V_{gs} > V_t$ and V_{ds} is large, the transistor acts as a current source, in which the current flow becomes independent of V_{ds}.

Explain the three different types of modes of operation of pMOS transistor.

- ✓ The pMOS transistor in Figure 4 operates in just the opposite fashion. The n-type body is tied to high potential, junctions of p-type source and drains are normally reverse-biased.
- ✓ When the gate has high potential, no current flows between drain and source.
- ✓ When the gate voltage is lowered by a threshold V_t, holes are attracted to form a p-type channel beneath the gate, allowing current to flow between drain and source.

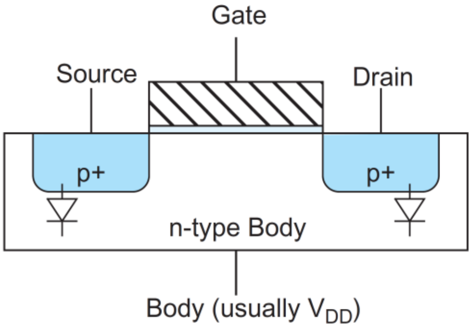


Figure 4: pMOS transistor

****

1.5: IDEAL I-V CHARACTERISTICS OF MOS TRANSISTOR

- ❖ **Derive an expression for I_{ds} of nMOS in linear and saturated region. (April 2019-6M)**
- ❖ **Derive an expression to show the drain current of MOS for various operating region. Explain one non-ideality for each operating region that changes the drain current. (NOV 2018)**
- ❖ **Explain the dynamic behavior of MOSFET transistor with neat diagram. (April 2018)**
- ❖ **Explain the electrical properties CMOS. (Nov 2017)**
- ❖ **Explain in detail about the ideal I-V characteristics of a NMOS and PMOS device. (MAY 2013)**
- ❖ **Discuss in detail with necessary equations the operation of MOSFET and its current-voltage characteristics. (April/May 2011, May 2016).**
- ❖ **Derive drain current of MOS device in different operating regions. (Nov/Dec 2014)(May/June 2013) (Nov 2012, Nov 2016)**
- ❖ **Explain in detail about the ideal I-V characteristics and non-ideal I-V characteristics of a NMOS and PMOS device. (May/June 2013)**
- ❖ **Derive expressions for the drain-to-source current in the nonsaturated and saturated regions of operation of an nMOS transistor. (Nov 2007, Nov 2008) [April / May – 2023]**

- ✓ MOS transistor has three regions of operation:
  - Cutoff (or) sub threshold region
  - Linear region (or) Non saturation region

- Saturation region

✓ The current through an OFF transistor is zero. When a transistor turns ON ($V_{gs} > V_t$), the gate attract electrons to form a channel.

✓ Current is measured from the amount of charge in the channel.

✓ The charge on each plate of a capacitor is $Q = CV$. Thus, the charge in the channel $Q_{channel}$ is

$$Q_{channel} = C_g (V_{gc} - V_t)$$

where C_g : Capacitance of the gate to the channel

$V_{gc} - V_t$: Amount of voltage attracting charge to the channel.

✓ If the source is at V_s and the drain is at V_d ,

✓ Average channel voltage is $V_c = (V_s + V_d)/2 = V_s + V_{ds}/2$.

✓ Gate and channel voltage V_{gc} is $V_g – V_c = V_{gs} – V_{ds}/2$,

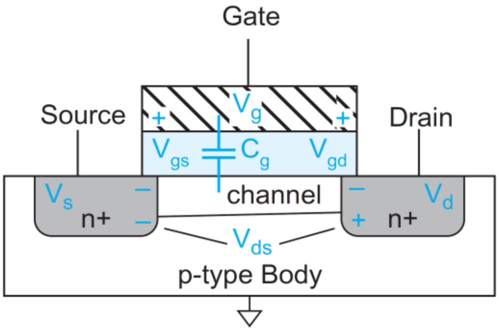


Figure 5: Average gate to channel voltage

✓ If the gate has length L and width W and the oxide thickness is t_{ox}, as shown in Figure 6, Then the capacitance C_g is

$$C_g = \kappa_{ox}\varepsilon_o \frac{WL}{t_{ox}} = \varepsilon_{ox}\frac{WL}{t_{ox}} = c_{ox}WL \quad \text{----------(1)}$$

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

Permittivity of SiO2 is $k_{ox} = 3.9$ times as great.

✓ The ε_{ox}/t_{ox} term is called as C_{ox}. Capacitance (C_{ox}) is a per unit area of the gate oxide.

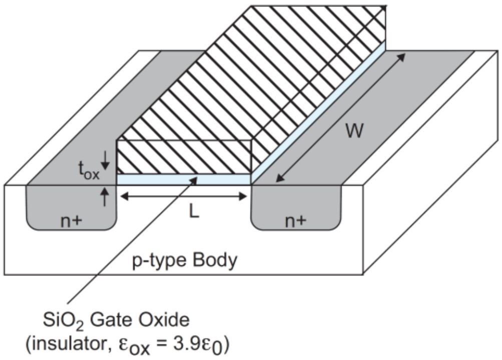


Figure 6: Transistor dimensions

✓ Average velocity (v) of carrier is proportional to the lateral electric field (field between source and drain). The constant of proportionality μ is called the mobility.

$$v = \mu E \text{ --------------(2)}$$

✓ The electric field E is the voltage difference between drain and source (V_{ds}) divided by the channel length (L).

$$E = \frac{V_{ds}}{L} \text{ -------------(3)}$$

✓ The time required for carriers to cross the channel is L divided by v.

- ✓ The current between source and drain is the total amount of charge in the channel divided by the time required to cross.

$$\begin{aligned} I_{ds} &= \frac{Q_{\text{channel}}}{L/v} \\ &= \mu C_{\text{ox}} \frac{W}{L}\left(V_{gs} - V_t - V_{ds}/2\right)V_{ds} \\ &= \beta\left(V_{GT} - V_{ds}/2\right)V_{ds} \end{aligned}$$

where

$$\beta = \mu C_{\text{ox}} \frac{W}{L}; \; V_{GT} = V_{gs} - V_t \qquad \text{----------- (4)}$$

- ✓ Equation (4) is called linear or resistive, because when $V_{ds} << V_{GT}$, I_{ds} increases linearly with V_{ds}, like an ideal resistor.
- ✓ k' is the k prime, k' = μC_{ox}.
- ✓ If $V_{ds} > V_{\text{dsat}} = V_{GT}$, the channel is no longer inverted in the drain region. Channel is pinched off.
- ✓ Beyond this point (called the drain saturation voltage), increasing the drain voltage has no further effect on current.
- ✓ Substituting $V_{ds} = V_{\text{dsat}}$ in Eq (4), we can find an expression for the saturation current (I_{ds}) that is independent of V_{ds}.

$$I_{ds} = \frac{\beta}{2} V_{GT}^2 \qquad \text{--------------------(5)}$$

- ✓ This expression is valid for $V_{gs} > V_t$ and $V_{ds} > V_{\text{dsat}}$.

- ✓ Summarizes the current in the three regions:

$$I_{ds} = \begin{cases} 0 & V_{gs} < V_t & \text{Cutoff} \\ \beta\left(V_{GT} - V_{ds}/2\right)V_{ds} & V_{ds} < V_{\text{dsat}} & \text{Linear} \\ \dfrac{\beta}{2} V_{GT}^2 & V_{ds} > V_{\text{dsat}} & \text{Saturation} \end{cases}$$

****

1.6: C – V CHARACTERISTICS OF MOS TRANSISTOR (AC characteristics)

- ❖ **Explain the dynamic behavior of MOSFET transistor with neat diagram. (April 2018)**
- ❖ **Discuss the CV characteristics of the CMOS. (Nov 2012, May 2014, Nov 2015, Nov 2016)**
- ❖ **Explain the electrical properties CMOS. (Nov 2017) [April / May 2023]**

- ✓ Each terminal of an MOS transistor has capacitance to the other terminals. Capacitances are nonlinear and voltage dependent (C-V).

SIMPLE MOS CAPACITANCES MODEL:

- ✓ The gate of an MOS transistor is a good capacitor. Its capacitance is necessary to attract charge to invert the channel, so high gate capacitance is required to obtain high I_{ds}.

- ✓ The gate capacitor can be viewed as a parallel plate capacitor with the gate on top, channel on bottom and the thin oxide dielectric between.
- ✓ The capacitance is $C_g = C_{ox}$ WL. ----------------(1)

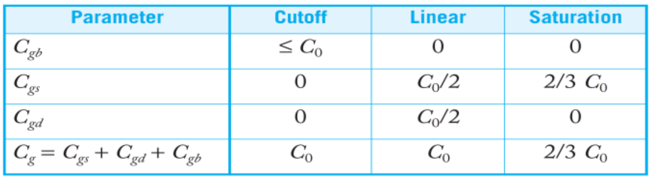
$$C_g = C_{permicron} W \text{ ---------------(2)}$$

$$\text{Where } C_{permicron} = C_{ox} L = \frac{\varepsilon_{ox}}{t_{ox}} L$$

- ✓ In addition to the gate, the source and drain also have capacitances. These capacitances are called parasitic capacitors.
- ✓ The source and drain capacitances arise from the p–n junctions between the source or drain diffusion and the body. These capacitances are called diffusion capacitance C_{sb} and C_{db}.
- ✓ The depletion region acts as an insulator between the conducting p- and n-type regions, creating capacitance across the junction.
- ✓ The capacitance of junctions depends on the area and perimeter of the source and drain diffusion, the depth of the diffusion, the doping levels and the voltage.
- ✓ As diffusion has both high capacitance and high resistance, it is generally made as small as possible in the layout.

DETAILED MOS GATE CAPACITANCE MODEL:

- ✓ MOS gate places above the channel and may partially overlap the source and drain diffusion areas.
- ✓ The gate capacitance has two components, (i) the intrinsic capacitance C_{gc} (over the channel) and (ii) the overlap capacitances C_{gol} (to the source and drain).
- ✓ The intrinsic capacitance was approximated as a simple parallel plate with capacitance
- ✓ $C_0 = WLC_{ox}$.
- ✓ The intrinsic capacitance has three components representing the different terminals connected to the bottom plate are C_{gb} (gate-to-body), C_{gs} (gate-to-source), and C_{gd} (gate-to-drain).
- ✓ The behavior in three regions (Cutoff, Linear and Saturation) can be approximated as shown in Table 1.



Parameter	Cutoff	Linear	Saturation
C_{gb}	$\le C_0$	0	0
C_{gs}	0	$C_0/2$	$2/3\ C_0$
C_{gd}	0	$C_0/2$	0
$C_g = C_{gs} + C_{gd} + C_{gb}$	C_0	C_0	$2/3\ C_0$

Table1: Approximation for intrinsic MOS gate capacitance

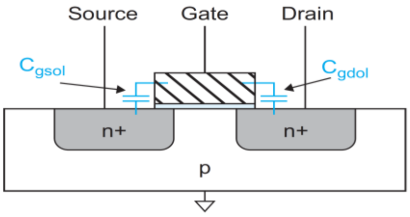


Figure 8: Overlap capacitances

DETAILED MOS DIFFUSION CAPACITANCE MODEL:

- ✓ The capacitance depends on both the area AS and sidewall perimeter PS of the source diffusion region. The area is AS = WD.
- ✓ The perimeter is PS = 2W +2D.

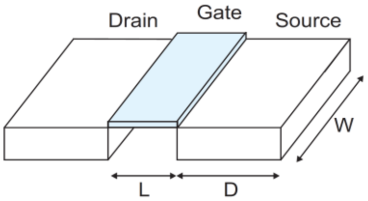


Figure 9: Diffusion region geometry

- ✓ The total source parasitic capacitance is $C_{sb} = AS * C_{jbs} + PS* C_{jbssw}$

  Where, C_{jbs} - Capacitance of the junction between the body and the bottom of the source
  C_{jbssw} - Capacitance of the junction between the body and the side walls of the source

- ✓ In summary, MOS transistor can be viewed as a four-terminal device with capacitances between each terminal pair, as shown in Figure 10.

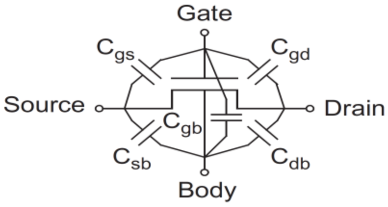


Figure 10: Capacitance of a MOS Transistor

- ✓ The gate capacitance includes an intrinsic component and overlap terms with the source and drain. The source and drain have parasitic diffusion capacitance to the body.

****

MOS(FET) Transistor Characteristic under Static and Dynamic Conditions

1.7: DC TRANSFER CHARACTERISTICS

- ❖ **Explain the DC transfer characteristic of CMOS inverter.[APRIL-2015, Nov 2015]**
- ❖ **Draw and explain the DC and transfer characteristics of a CMOS inverter with necessary conditions for the different regions of operation. (Nov/Dec 2011) (Nov/Dec 2012) (May/June 2013) (April/May 2012) (May/June 2014) (Nov/Dec 2013) (May 2016, May 2017, Nov 2008) [April/May-2022.] [Nov/Dec-2020., April/May-2021] [April/May 2023]**
- ❖ **Explain the CMOS inverter DC characteristics. (Nov 2007, Nov 2009) [Nov/Dec 2022]**

- ✓ The DC transfer characteristics of a circuit relate the output voltage to the input voltage.

(i) Static CMOS inverter DC Characteristics:
The DC transfer function (V_{out} Vs. V_{in}) for the static CMOS inverter shown in Figure 11.

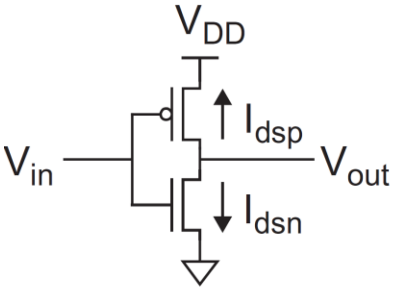


Figure 11: A Static CMOS inverter

- ✓ Table 2, shows various regions of operation for the n and p transistors.
- ✓ In this table, V_{tn} is the threshold voltage of the n-channel device, and V_{tp} is the threshold voltage of the p-channel device. V_{tp} is negative.
- ✓ The equations are given both in terms of V_{gs} /V_{ds} and V_{in} /V_{out}.
- ✓ As the source of the nMOS transistor is grounded, $V_{gsn} = V_{\text{in}}$ and $V_{dsn} = V_{\text{out}}$.
- ✓ As the source of the pMOS transistor is tied to V_{DD}, $V_{gsp} = V_{\text{in}} – V_{DD}$ and $V_{dsp} = V_{\text{out}} – V_{DD}$.

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

Table 2: Relationships between voltages for the three regions of operation of a CMOS inverter

- ✓ Figure 12(a), shows I_{dsn} and I_{dsp} in terms of V_{dsn} and V_{dsp} for various values of V_{gsn} and V_{gsp}.
- ✓ Figure 12(b), shows the same plot of I_{dsn} and $|I_{dsp}|$ in terms of V_{out} for various values of V_{in}.
- ✓ Operating points are plotted on V_{out} vs. V_{in} axes in Figure 12(c) to show the inverter DC transfer characteristics.
- ✓ The supply current $I_{DD} = I_{dsn} = |I_{dsp}|$ is plotted against V_{in} in Figure 13(d) showing that both transistors are momentarily ON as V_{in}.
- ✓ The operation of the CMOS inverter can be divided into five regions as indicated on figure 12(c).

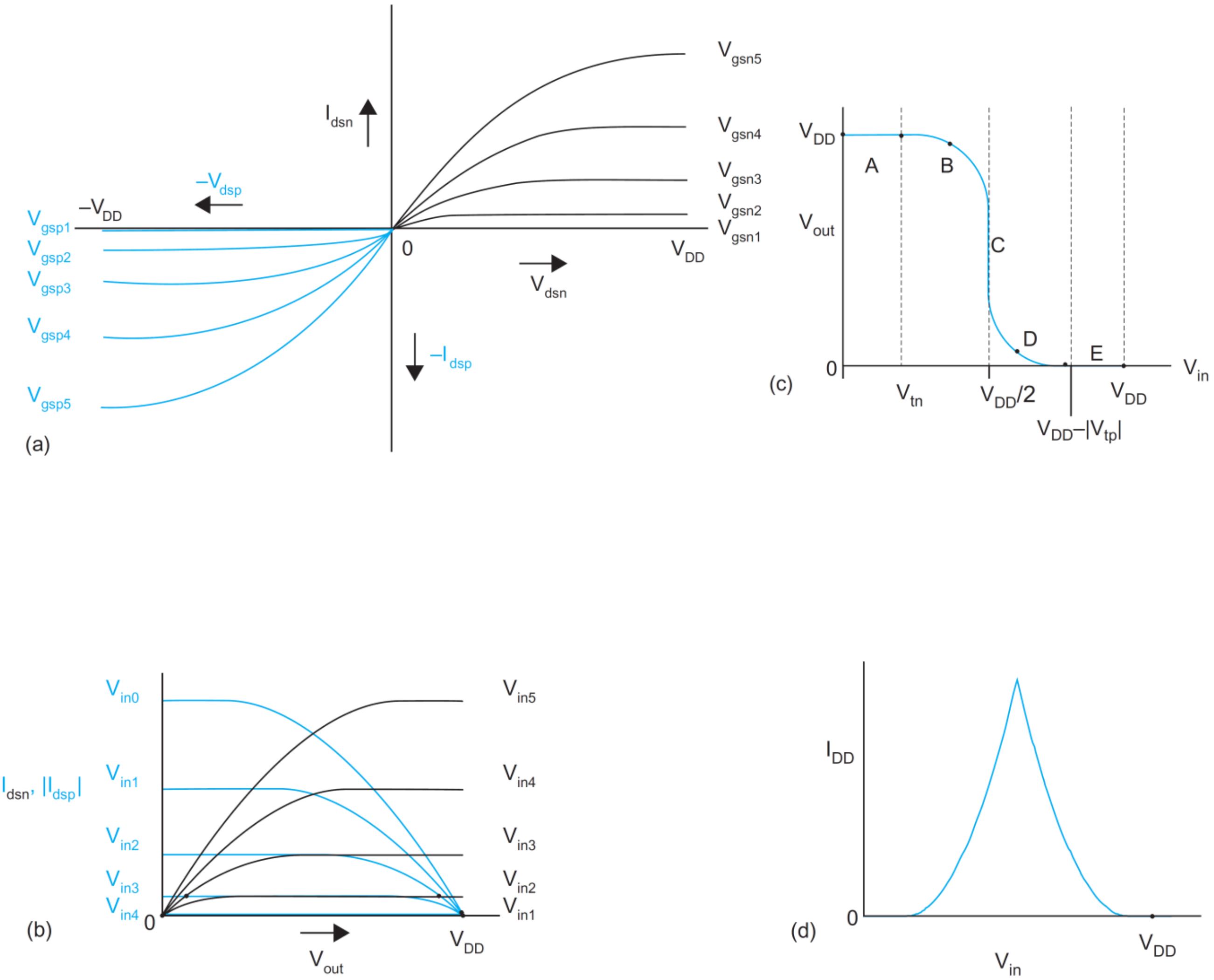


Figure 12: CMOS inverter DC characteristic

- ✓ The state of each transistor in each region is shown in Table 3.

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

Table 3: Summary of CMOS inverter operation.

- ✓ In region *A*, the nMOS transistor is OFF and the pMOS transistor pulls the output to V_{DD}.
- ✓ In region *B*, the nMOS transistor starts to turn ON. It is pulling the output down.
- ✓ In region *C*, both transistors are in saturation.
- ✓ In region *D*, the pMOS transistor is partially ON.
- ✓ In region *E*, PMOS is completely OFF, making the nMOS transistor to pull the output down to GND.

(ii) Beta ratio Effects:

- ✓ For $\beta_p = \beta_n$, the inverter threshold voltage V_{inv} is $V_{DD}/2$.
- ✓ It allows a capacitive load to charge and discharge in equal times by providing equal current source and equal sink capabilities.

- ✓ Inverter with different beta ratios r = β_p / β_n is called skewed inverter.
- ✓ If r > 1, the inverter is HI-skewed. If r < 1, the inverter is LO-skewed. If r = 1, the inverter has normal skew or is unskewed.
- ✓ Figure 13, shows the impact of skewing the beta ratio on the DC transfer characteristics.
- ✓ As the beta ratio is changed, the switching threshold is varied.

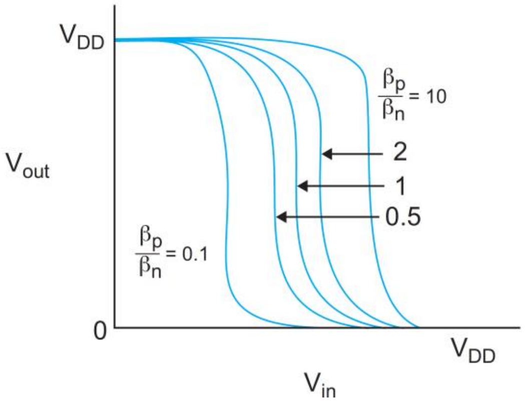


Figure 13: Transfer characteristics of skewed inverters

❖ **Derive the noise margins for a CMOS inverter. (May 2010, Nov 2016)**

(iii) **Noise Margins:**

- ✓ Noise margin (Noise immunity) is related to the DC voltage characteristics.
- ✓ Noise Margin allows determining the allowable noise voltage on the input of a gate, so that the output will not be corrupted.
- ✓ Two parameters of the noise margin are LOW noise margin (NM_L), and the HIGH noise margin (NM_H).

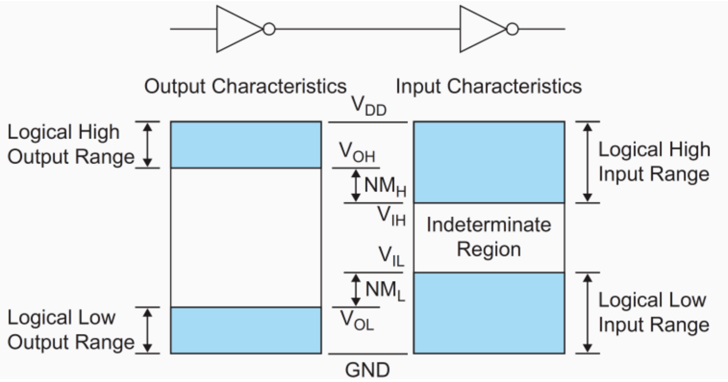


Figure 14: Noise Margin Definitions

- ✓ NM_L is defined as the difference in maximum LOW input voltage V_{IL} and the maximum LOW output voltage V_{OL}. $NM_L = V_{IL} - V_{OL}$
- ✓ The value of NM_H is the difference between the minimum HIGH output voltage V_{OH} and the minimum HIGH input voltage V_{IH}. i.e., $NM_H = V_{OH} - V_{IH}$
- ✓ Inputs between V_{IL} and V_{IH} are said to be in the indeterminate region or forbidden zone.

(iv) Pass Transistor DC Characteristics:

- ✓ The nMOS transistors pass 0's well but 1's poorly. Figure 15(a), shows an nMOS transistor with the gate and drain tied to V_{DD}.
- ✓ Initially at $Vs = 0$. $V_{gs} > V_{tn}$, so the transistor is ON and current flow.

- ✓ Therefore, nMOS transistors attempting to pass a 1 never pull the source above $V_{DD} – V_{tn}$. This loss is called a threshold drop.
- ✓ The pMOS transistors pass 1's well but 0's poorly.
- ✓ If the pMOS source drops below $|V_{tp}|$, the transistor cuts off.
- ✓ Hence, pMOS transistors only pull down to a threshold above GND, as shown in Figure 15(b).

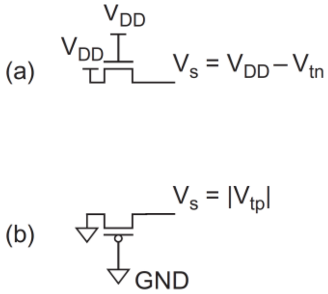


Figure 15: Pass Transistor threshold drops

****

1.8: NON IDEAL I-V EFFECTS

- ❖ **Explain in detail about the non ideal I-V characteristics of a CMOS device. (MAY 2013)**
- ❖ **Explain channel length modulation and body effect. (Nov 2009, May 2013)**

- ✓ MOS characteristics degrade with temperature. It is useful to have a qualitative understanding of non ideal effects to predict their impact on circuit behavior.

(i) Mobility Degradation and Velocity Saturation:

- ✓ Current is proportional to the lateral electric field $E_{lat} = V_{ds} /L$ between source and drain.

- ✓ A high voltage at the gate of the transistor attracts the carriers to the edge of the channel, causing carriers collision with the oxide interface that slows the carriers. This is called mobility degradation.
- ✓ Carriers approach a maximum velocity (v_{sat}) when high fields are applied. This phenomenon is called velocity saturation.

(ii) Channel Length Modulation:

- ✓ Current I_{ds} is an independent of V_{ds} for a transistor in saturation.
- ✓ The p–n junction between the drain and body forms a depletion region with a width L_d that increases with V_{db} , as shown in Figure 16.
- ✓ The depletion region effectively shortens the channel length to $L_{eff} = L - L_d$
- ✓ To avoid the body voltage into calculations, assume the source voltage is close to the body voltage i.e $V_{db} = V_{ds}$.
- ✓ Hence, increasing V_{ds} decreases the effective channel length.
- ✓ Shorter channel length results in higher current. Thus, I_{ds} increases with V_{ds} in saturation, as shown in Figure 16.

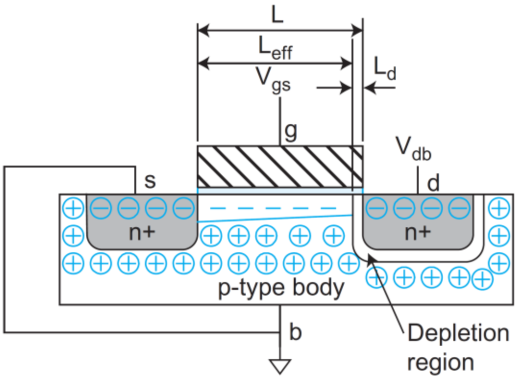


Figure 16: Depletion region shortens effective channel length

- ✓ In Saturation region, I_{ds} is

$$I_{ds} = \frac{\beta}{2} V_{GT}^2 \left(1 + \frac{V_{ds}}{V_A}\right)$$

- ✓ Hence, V_A is proportional to channel length. This channel length modulation model is a gross oversimplification of nonlinear behavior.

(iii) Threshold Voltage (V_t) Effects

❖ **Explain in detail about effect and its effect in MOS device. (May 2016)**

- ✓ Threshold voltage V_t increases with the source voltage, decreases with the body voltage, decreases with the drain voltage and increases with channel length.

Body Effect:

- ✓ When a voltage V_{sb} is applied between the source and body, it increases the amount of charge required to invert the channel. Hence, it increases the threshold voltage.
- ✓ The threshold voltage can be modeled as

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

where V_{t0} is the threshold voltage when the source is at the body potential,
ϕ_s is the surface potential at threshold and γ is the body effect coefficient.

(iv) Leakage:

- ✓ Even when transistors are OFF, transistors leak small amounts of current.
- ✓ Leakage mechanisms include subthreshold conduction between source and drain, gate leakage from the gate to body and junction leakage from source to body and drain to body.
- ✓ Subthreshold conduction is caused by thermal emission of carriers over the potential barrier set by the threshold.
- ✓ Gate leakage is a quantum-mechanical effect caused by tunneling through the extremely thin gate dielectric.
- ✓ Junction leakage is caused by current through the p-n junction between the source/drain diffusions and the body.

****

1.9: Device models:

Explain the following: Device models and device characteristics. (MAY 2014)

- ✓ SPICE (Simulation Program with Integrated Circuit Emphasis) provides a wide variety of MOS transistor models with various trade-offs between complexity and accuracy.
- ✓ Level 1 and Level 3 models were important, but they are no longer adequate to accurately model very small modern transistors.
- ✓ BSIM models are more accurate and are presently the most widely used.

i. Level 1 model:

- ✓ The SPICE Level 1, or Shichman-Hodges Model is closely related to the Shockley model, enhanced with channel length modulation and the body effect.
- ✓ The basic current model is:

$$I_{ds} = \begin{cases} 0 & V_{gs} < V_t \quad \text{cutoff} \\ \text{KP}\dfrac{W_{\text{eff}}}{L_{\text{eff}}}\left(1+\text{LAMBDA}\times V_{ds}\right)\left(V_{gs}-V_t-{V_{ds}}/{2}\right)V_{ds} & V_{ds} < V_{gs}-V_t \quad \text{linear} \\ \dfrac{\text{KP}}{2}\dfrac{W_{\text{eff}}}{L_{\text{eff}}}\left(1+\text{LAMBDA}\times V_{ds}\right)\left(V_{gs}-V_t\right)^2 & V_{ds} > V_{gs}-V_t \quad \text{saturation} \end{cases}$$

- ✓ The parameters from the SPICE model are given in ALL CAPS.
- ✓ β is written instead as KP (W_{eff} /L_{eff}), where KP is a model parameter. W_{eff} and L_{eff} are the effective width and length.
- ✓ The LAMBDA term (LAMBDA =1/V_A) models channel length modulation.
- ✓ The threshold voltage is modulated by the source-to-body voltage V_{sb} through the body effect.
- ✓ For non negative V_{sb} , the threshold voltage, V_t is

$$V_t = \text{VTO} + \text{GAMMA}\left(\sqrt{\text{PHI}+V_{sb}} - \sqrt{\text{PHI}}\right)$$

Where, VTO is the “zero-bias” threshold voltage V_{t0},

GAMMA is the body effect coefficient , and PHI is the surface potential.

- ✓ Level 1 model is easy to correlate with hand analysis, but it is too simplistic for modern design.

ii. Level 2 and 3 models

- ✓ The SPICE Level 2 and 3 models add effects of velocity saturation, mobility degradation, subthreshold conduction and drain-induced barrier lowering.
- ✓ The Level 2 model is based on the Grove-Frohman equations.
- ✓ Level 3 model is based on empirical equations that provide similar accuracy, faster simulation times and better convergence.
- ✓ These models are not efficient models for measuring I-V characteristics of modern transistors.

iii. BSIM models

- ✓ The Berkeley Short-Channel IGFET Model (BSIM) is a very elaborate model that is now widely used in circuit simulation.
- ✓ The models are derived from the fundamental device physics but uses many number of parameters to fit the behavior of modern transistors.

- ✓ BSIM versions 1, 2, 3v3, and 4 are implemented as SPICE levels 13, 39, 49, and 54, respectively.
- ✓ BSIM is quite good for digital circuit simulation.
- ✓ Features of the model are
    - Continuous and differentiable I-V characteristics across subthreshold, linear and saturation regions for good convergence.
    - Sensitivity of parameters such as V_t to transistor length and width.
    - Detailed threshold voltage model including body effect and drain-induced barrier Lowering (DIBL).
    - Velocity saturation, mobility degradation and other short-channel effects.
    - Multiple gate capacitance models.
    - Diffusion capacitance and resistance models.
    - Gate leakage models.

****

1.10: SCALING

- ❖ **Discuss the scaling principles and its limits. (MAY 2013, Nov 2017, Nov 2018)**
- ❖ **Discuss the principle of constant field and lateral scaling. Write the effects of the above scaling methods on the device characteristics. (Nov 2012, Dec 2011, Nov 2015, May 2016)**
- ❖ **Explain need of scaling, scaling principles and fundamental units of CMOS inverter. (May 2017)**
- ❖ **Highlight the need for scaling. Enumerate in detail constant electric field, constant voltage and combined electric field and voltage scaling for different parameters of MOSFET. [Nov 2019]**

- ✓ In VLSI design, the transistor size has reduced by 30% every two to three years. Scaling is reducing feature size of transistor.
- ✓ Nowadays, transistors become smaller, switch faster, dissipate less power and cheaper.
- ✓ Designers need to predict the effect of feature size scaling on chip performance to plan future products and ensure existing products for cost reduction.

Transistor scaling:

- ✓ Dennard's Scaling Law predicts that the basic operational characteristics of a MOS transistor can be preserved and the performance can be improved.
- ✓ Parameters of a device are scaled by a dimensionless factor S.
- ✓ These parameters include the following:
    - All dimensions (in the x, y, and z directions)
    - Device voltages
    - Doping concentration densities

Constant field scaling (Full Scaling):

- ✓ In **constant field scaling,** electric fields remain the same as both voltage and distance shrink.
- ✓ 1/S scaling is applied to all dimensions, device voltages and concentration densities.
    - I_{ds} per transistor are scaled by 1/S.
    - No. of transistors per unit area is scaled by S^2.
    - Current density is scaled by S and power density remains constant.
        - ○ e.g., $(\frac{1}{S}*\frac{1}{S})*S^2$

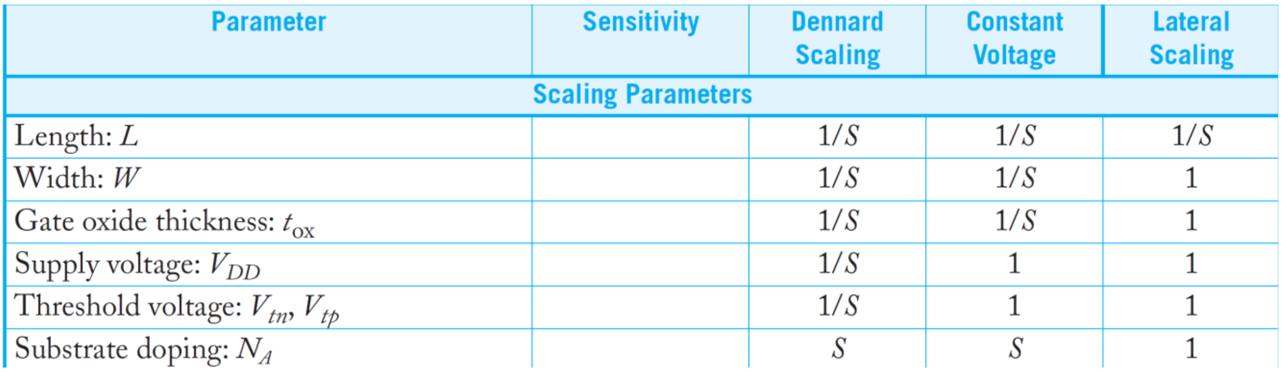
Lateral scaling (gate-shrink):

- ✓ Another approach is **lateral scaling**, in which only the gate length is scaled.

- ✓ This is commonly called as gate shrink, because it can be done easily to an existing mask database for a design.
  - I_{ds} per transistor are scaled by S.
  - No. of transistors per unit area is scaled by S.
  - Current density is scaled by S^2 and power density is scaled by S^2.
- ✓ The industry generally scales process generations with 30% shrink.
- ✓ It reduces the cost (area) of a transistor by a factor of two.
- ✓ A 5% gate shrink (S = 1.05) is commonly applied as a process, becomes mature to boost the speed of components in that process.

Constant voltage scaling: V $_{DD}$ is held constant, while process is scaled.

- ✓ **Constant voltage scaling (Fixed scaling)** offers quadratic delay improvement as well as cost reduction.
- ✓ It is also maintaining continuity in I/O voltage standards. Constant voltage scaling increases the electric fields in devices.
  - I_{ds} per transistor are scaled by S.
  - No. of transistors per unit area is scaled by S^2.
  - Current density is scaled by S^3 and power density is scaled by S^3.
- ✓ A 30% shrink with Dennard scaling improves clock frequency by 40% and cuts power consumption per gate by a factor of 2.
- ✓ Maintaining a constant field has the further benefit, that many nonlinear factors and wear out mechanisms are unaffected.
- ✓ From 90nm generation technology, voltage scaling is dramatically slowed down due to leakage. This may ultimately limit CMOS scaling.



Parameter	Sensitivity	Dennard Scaling	Constant Voltage	Lateral Scaling
Scaling Parameters				
Length: L		$1/S$	$1/S$	$1/S$
Width: W		$1/S$	$1/S$	1
Gate oxide thickness: t_{ox}		$1/S$	$1/S$	1
Supply voltage: V_{DD}		$1/S$	1	1
Threshold voltage: V_{tn}, V_{tp}		$1/S$	1	1
Substrate doping: N_A		S	S	1

Table: Influence of scaling on MOS device characteristics

Interconnecting Scaling:

- ✓ Wires to be scaled equally in width and thickness to maintain an aspect ratio close to 2.
- ✓ Wires can be classified as local, semiglobal and global.
- ✓ Local wires run within functional units and use the bottom layers of metal.
- ✓ Semiglobal wires run across larger blocks or cores, typically using middle layers of metal.
- ✓ Both local and semiglobal wires are scaling with feature size.
- ✓ Global wires run across the entire chip using upper levels of metal.
- ✓ Global wires do not scale with feature size. Indeed, they may get longer (by a factor of DC, on the order of 1.1) because, die size has been gradually increasing.
- ✓ When wire thickness is scaled, the capacitance per unit length remains constant.

****

1.11: CMOS DEVICES & TECHNOLOGIES:

The four main CMOS technologies are

- n-Well process
- p-Well process
- Twin-tub Process
- Silicon on Insulator

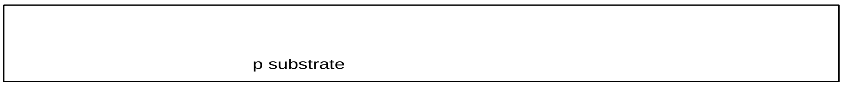
- **Explain the different steps involved in CMOS fabrication / manufacturing process with neat diagrams. (Nov 2007, Nov 2009, Nov 2016, NOV 2018)**
- **Describe with neat diagram the n-well and channel formation in CMOS process. (Nov/Dec 2014)(Nov/Dec 2011) (April/May 2011) (Nov/Dec 2012)**

n-WELL PROCESS:

Step 1: Start with blank wafer

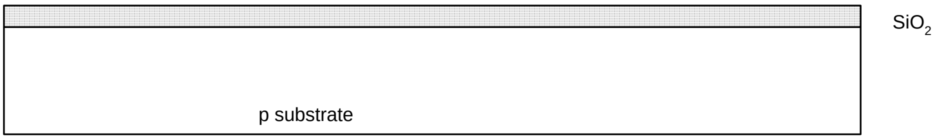
First step will be to form the n-well

- Cover wafer with protective layer of SiO_2 (oxide)
- Remove layer where n-well should be built.



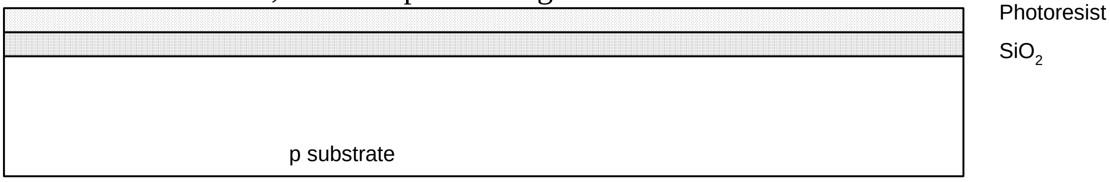
Step 2: Oxidation

Grow SiO_2 on top of Si wafer, at 900 – 1200^0 C with H_2O or O_2 in oxidation furnace.



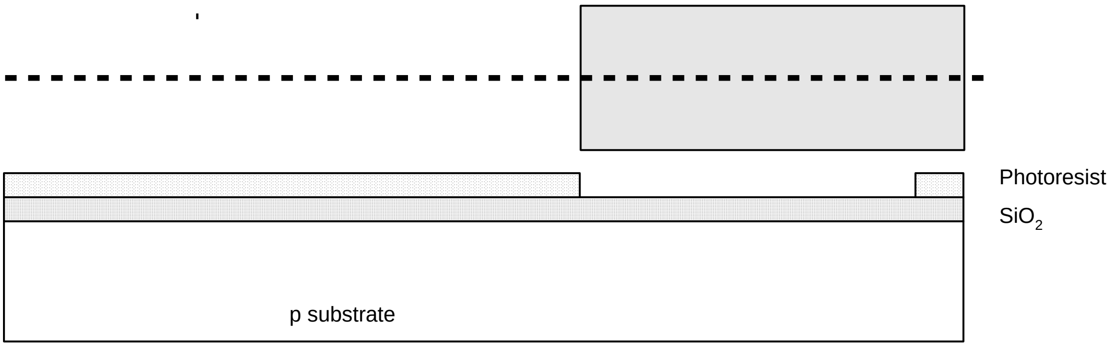
Step 3: Photoresist

- Spin on photoresist
  - Photoresist is a light-sensitive organic polymer.
  - Softens, where exposed to light.



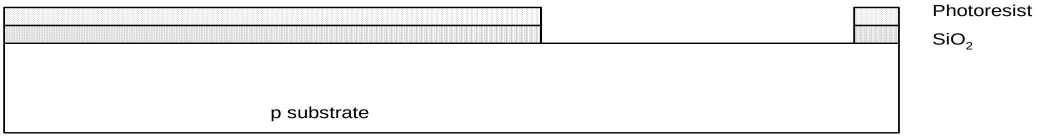
Step 4: Lithography

- Expose photoresist through n-well mask.
- Strip off exposed photoresist.



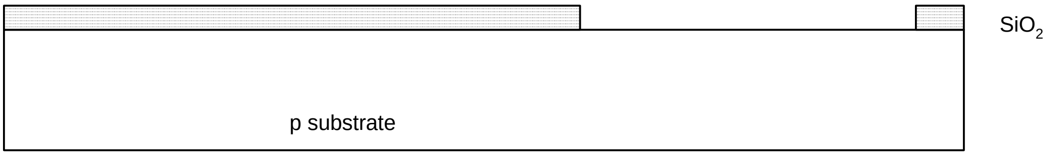
Step 5: Etch

- Etch oxide with hydrofluoric acid (HF).
- Only attracts oxide, where resist has been exposed.



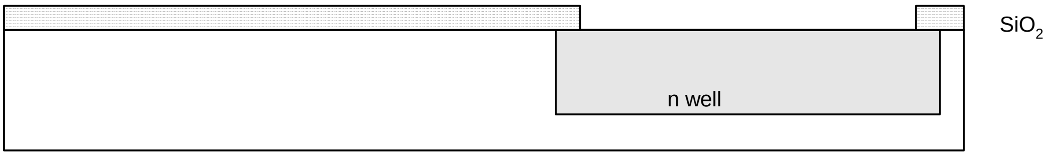
Step 6: Strip Photoresist

- Etch the remaining photoresist using a mixture of acids.



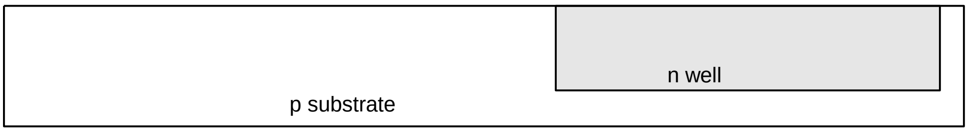
Step 7: n-well

n-well is formed with diffusion or ion implantation.



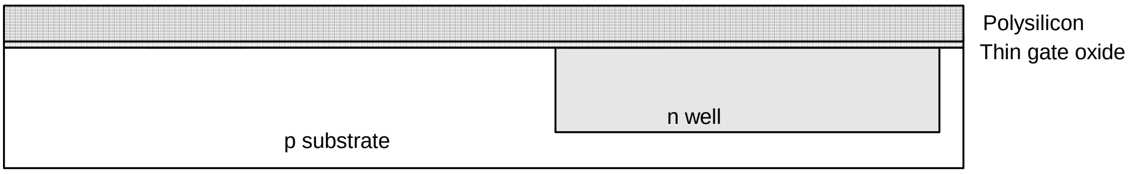
Step 8: Strip Oxide

- Strip off the remaining oxide using HF.
- Back to bare wafer with n-well.
- Subsequent steps involve similar series of steps.



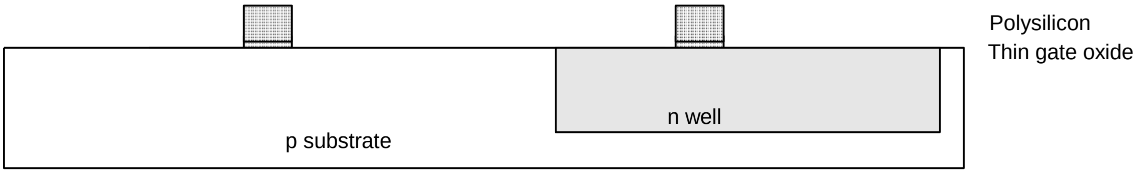
Step 9:Polysilicon

- Deposit thin layer of oxide. Use CVD to form poly and dope heavily to increase conductivity.



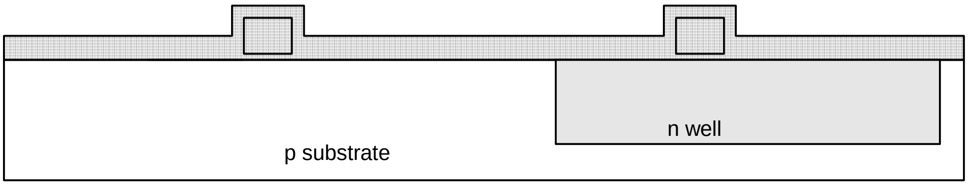
Step 10: Polysilicon Patterning

- Use same lithography process to pattern polysilicon.



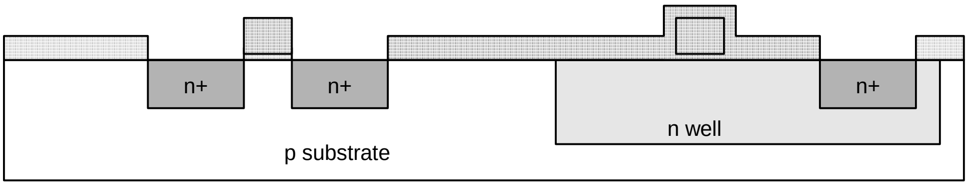
Step 11: Self-Aligned Process

Cover with oxide to define n diffusion regions.



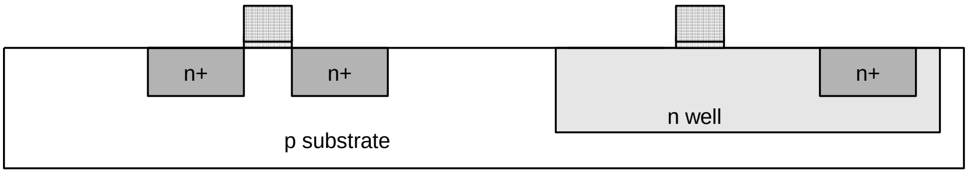
Step 12: N-diffusion

- Pattern oxide, using n+ active mask to define n diffusion regions.
- Diffusion or ion implantation is used to create n diffusion regions.



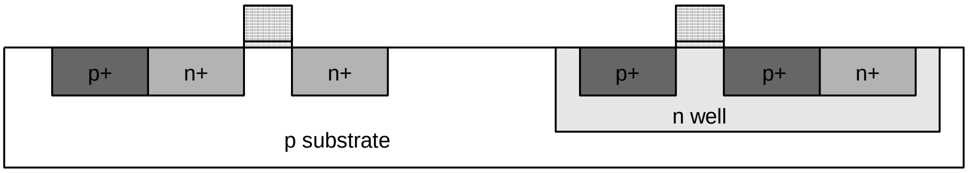
Step 13:

- Strip off oxide to complete patterning step.



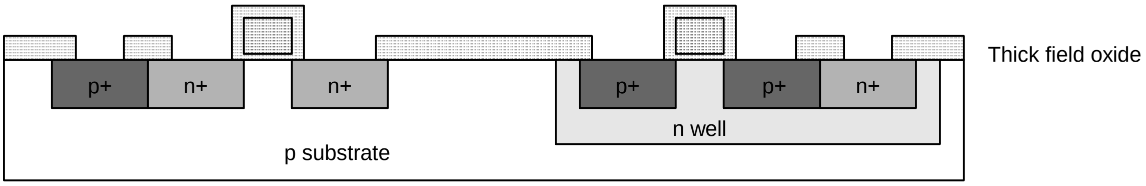
Step 14:P-Diffusion

- Similar set of steps are followed to form p+ diffusion regions for pMOS source and drain and substrate contact.



Step 15: Contacts

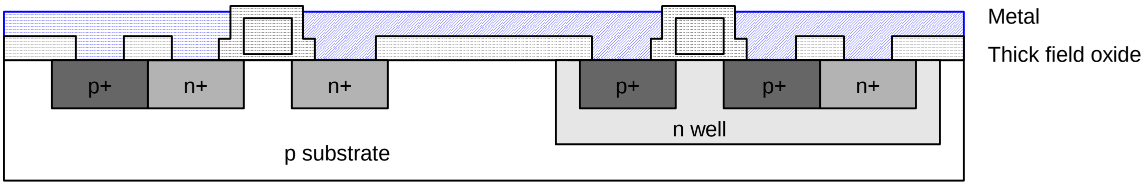
- Now, we need to wire together the devices.
- Cover chip with thick field oxide.
- Etch oxide, where contact cuts are needed.



Step 16: Metallization

- Sputter on aluminum over whole wafer.

- Pattern to remove excess metal, leaving wires.



****

P-WELL PROCESS:

- A common approach to p-well CMOS fabrication is to start with moderately doped n-type substrate (wafer), create the p-type well for the n-channel devices and build the p-channel transistor in the native n-substrate.

❖ **Explain the twin tub process with a neat diagram. (Nov 2007, April 2008)**

Twin-tub process:

Step 1:

n- Substrate is taken initially, which is shown in figure.

Step 2:

Next step is epitaxial layer deposition. Lightly doped epitaxial layer is deposited above n-substrate.

Step 3:

The next step is tub formation. Two wells are formed namely n-well and p-well.

Polysilicon layer is formed above overall substrate.

Step 4:

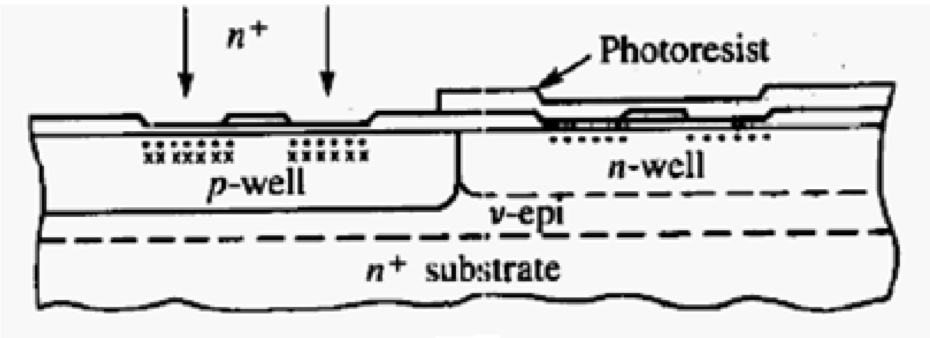
Polysilicon gates are formed for n-well and p-well by using photo-etching process.

Step 5:

n^+diffusion is formed in n-well, P^+ diffusion is formed in p-well. These are used for V_{DD} contact and V_{SS} contact. These are known as substrate formation.

Step 6:

Then, contact cuts are defined as in n-well process. Then metallization is processed.



****

Power consumption

Reduction of power consumption makes a device more reliable. The need for devices that consume a minimum amount of power was a major driving force behind the development of CMOS technologies. As a result, CMOS devices are best known for low power consumption. However, for minimizing the power requirements of a board or a system, simply knowing that CMOS devices may use less power than equivalent devices from other technologies does not help much. It is important to know not only how to calculate power consumption, but also to understand how factors such as input voltage level, input rise time, power-dissipation capacitance, and output loading affect the power consumption of a device.

The main topics discussed are:
- Power-consumption components
- Static power consumption
- Dynamic power consumption

Power-Consumption Components

High frequencies impose a strict limit on power consumption in computer systems as a whole. Therefore, power consumption of each device on the board should be minimized. Power calculations determine power-supply sizing, current requirements, cooling/heatsink requirements, and criteria for device selection. Power calculations also can determine the maximum reliable operating frequency.

Two components determine the power consumption in a CMOS circuit:
- Static power consumption
- Dynamic power consumption

- ✓ CMOS devices have very low static power consumption, which is the result of leakage current.
- ✓ This power consumption occurs when all inputs are held at some valid logic level and the circuit is not in charging states.
- ✓ But, when switching at a high frequency, dynamic power consumption can contribute significantly to overall power consumption.
- ✓ Charging and discharging a capacitive output load further increases this dynamic power consumption. This application report addresses power consumption in CMOS logic families (5 V and 3.3 V) and describes the methods for evaluating both static and dynamic power consumption.
- ✓ Additional information is also presented to help explain the causes of power consumption, and present possible solutions to minimize power consumption in a CMOS system.
- ✓ Static Power Consumption Typically, all low-voltage devices have a CMOS inverter in the input and output stage. Therefore, for a clear understanding of static power consumption, refer to the CMOS inverter modes shown in Figure

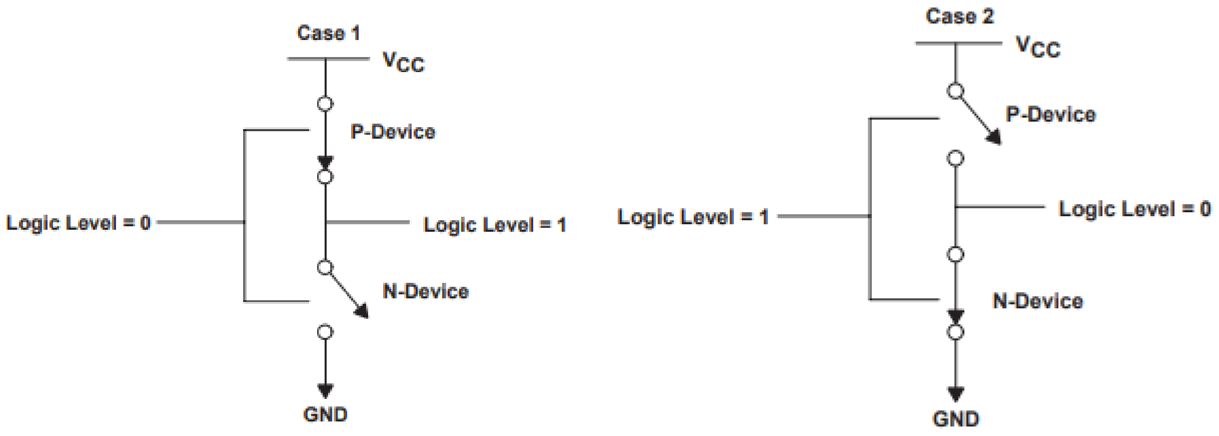


Figure: CMOS Inverter modes

- As shown in above Figure, if the input is at logic 0, the n-MOS device is OFF, and the p-MOS device is ON (Case 1).
- The output voltage is V_{CC}, or logic 1. Similarly, when the input is at logic 1, the associated n-MOS device is biased ON and the p-MOS device is OFF.
- The output voltage is GND, or logic 0. Note that one of the transistors is always OFF when the gate is in either of these logic states.
- Since no current flows into the gate terminal, and there is no dc current path from V_{CC} to GND, the resultant quiescent (steady-state) current is zero, hence, static power consumption (Pq) is zero.
- However, there is a small amount of static power consumption due to reverse-bias leakage between diffused regions and the substrate.
- This leakage inside a device can be explained with a simple model that describes the parasitic diodes of a CMOS inverter, as shown in Figure below.

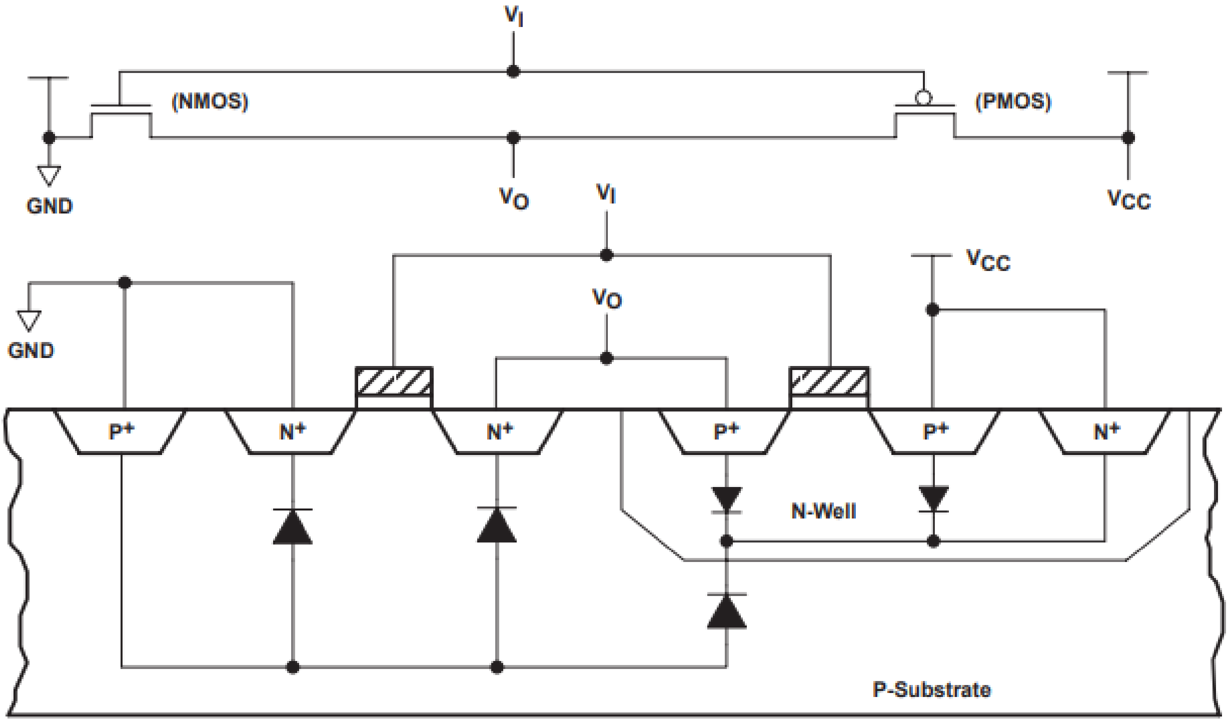


Figure: Model Describing Parasitic Diodes Present in CMOS Inverter

The source drain diffusion and N-well diffusion form parasitic diodes. In above Figure, the parasitic diodes are shown between the N-well and substrate. Because parasitic diodes are reverse biased, only their leakage currents contribute to static power consumption.

The leakage current (I_{lkg}) of the diode is described by the following equation:

$I_{lkg} = i_S (e^{qV/kT} - 1)$ ……………. (1)

Where:

i_S = reverse saturation current

V = diode voltage

k = Boltzmann’s constant (1.38×10^{-23} J/K)

q = electronic charge (1.602×10^{-19} C)

T = temperature

Static power consumption is the product of the device leakage current and the supply voltage.

Total static power consumption, P_S, can be obtained as shown in equation 2.

$P_S = \Sigma$ (leakage current) x (supply voltage) …………. (2)

Most CMOS data sheets specify an ICC maximum in the 10-µA to 40-µA range, encompassing total leakage current and other circuit features that may require some static current not considered in the simple inverter model.

The leakage current ICC (current into a device), along with the supply voltage, causes static power consumption in the CMOS devices.

This static power consumption is defined as quiescent, or P_S, and can be calculated by equation 3.

$P_S = V_{CC} \text{ x } I_{CC}$ ………. (3)

Where:

V_{CC} = supply voltage

I_{CC} = current into a device (sum of leakage currents as in equation 2)

Another source of static current is ΔI_{CC}. This results when the input levels are not driven all the way to the rail, causing the input transistors to not switch off completely.

Dynamic Power Consumption

The dynamic power consumption of a CMOS IC is calculated by adding the transient power consumption (P_T), and capacitive-load power consumption (P_L).

- ✓ Transient Power Consumption Transient power consumption is due to the current that flows only when the transistors of the devices are switching from one logic state to another.
- ✓ This is a result of the current required to charge the internal nodes (switching current) plus the through current (current that flows from V_{CC} to GND when the p-channel transistor and n-channel transistor turn on briefly at the same time during the logic transition).
- ✓ The frequency at which the device is switching, plus the rise and fall times of the input signal, as well as the internal nodes of the device, have a direct effect on the duration of the current spike.
- ✓ For fast input transition rates, the through current of the gate is negligible compared to the switching current. For this reason, the dynamic supply current is governed by the internal capacitance of the IC and the charge and discharge current of the load capacitance.

Transient power consumption can be calculated using equation 4.

$P_T = C_{pd} \text{ x } V_{CC}^2 \text{ x } f_I \text{ x NSW}$ ……………….. (4)

Where:

P_T = transient power consumption

V_{CC} = supply voltage

f_I = input signal frequency

NSW = number of bits switching

C_{pd} = dynamic power-dissipation capacitance

In the case of single-bit switching, NSW in equation 4 is 1.

Dynamic supply current is dominant in CMOS circuits because most of the power is consumed in moving charges in the parasitic capacitor in the CMOS gates.

As a result, the simplified model of a CMOS circuit consisting of several gates can be viewed as one large capacitor that is charged and discharged between the power-supply rails.

Therefore, the power–dissipation capacitance (C_{pd}) is often specified as a measure of this equivalent capacitance and is used to approximate the dynamic power consumption.

C_{pd} is defined as the internal equivalent capacitance of a device calculated by measuring operating current without load capacitance.

Depending on the output switching capability, C_{pd} can be measured with no output switching (output disabled) or with any of the outputs switching (output enabled).

Capacitive-Load Power Consumption

Additional power is consumed in charging external load capacitance and is dependent on switching frequency.

The following equation can be used to calculate this power if all outputs have the same load and are switching at the same output frequency.

$P_L = C_L \times V_{CC}^2 \times f_O \times NSW$ (C_L is the load per output) ……….. (5)

Where:

P_L = capacitive-load power consumption

V_{CC} = supply voltage

f_O = output signal frequency

C_L = external (load) capacitance

NSW = total number of outputs switching

In the case of different loads and different output frequencies at all outputs, equation 6 is used to calculate capacitive-load power consumption.

$P_L = \Sigma (C_{Ln} \times f_{On}) \times V_{CC}^2$ ……….. (6)

Where:

Σ = sum of n different frequencies and loads at n different outputs

f_{On} = all different output frequencies at each output, numbered 1 through n (Hz)

V_{CC} = supply voltage (V)

C_{Ln} = all different load capacitances at each output, numbered 1 through n.

Therefore, dynamic power consumption (P_D) is the sum of these two power consumptions and can be expressed as shown in equation 7, equation 8 (single-bit switching), and equation 9 (multiple-bit switching with variable load and variable output frequencies).

$P_D = P_T + P_L$ ………… (7)

$P_D = (C_{pd} \text{ x } f_I \text{ x } V_{CC}^2) + (C_L \text{ x } f_O \text{ x } V_{CC}^2)$ ………… (8)

$P_D = [(C_{pd} \text{ x } f_I \text{ x } NSW) + \Sigma (C_{Ln} \text{ x } f_{On})] \text{ x } V_{CC}^2$ ………… (9)

Where:

C_{pd} = power-consumption capacitance (F)

f_I = input frequency (Hz)

f_{On} = all different output frequencies at each output, numbered 1 through n (Hz)

NSW = total number of outputs switching

V_{CC} = supply voltage (V)

C_{Ln} = all different load capacitances at each output, numbered 1 through n.

Total power consumption is the sum of static and dynamic power consumption.

$P_{tot} = P_{(static)} + P_{(dynamic)}$ ………….. (10)

****

TWO MARKS QUESTIONS & ANSWERS

UNIT I - MOS TRANSISTOR PRINCIPLES

1. **Give the advantages of Integrated Circuit.**
   Advantages of Integrated Circuit:
   - Size is less
   - High Speed
   - Less Power Dissipation

2. **What is meant by CMOS technology?**
   Complementary metal-oxide-semiconductor (CMOS) is a technology for constructing integrated circuits. CMOS technology is used in microprocessors, microcontrollers, static RAM and other digital logic circuits.

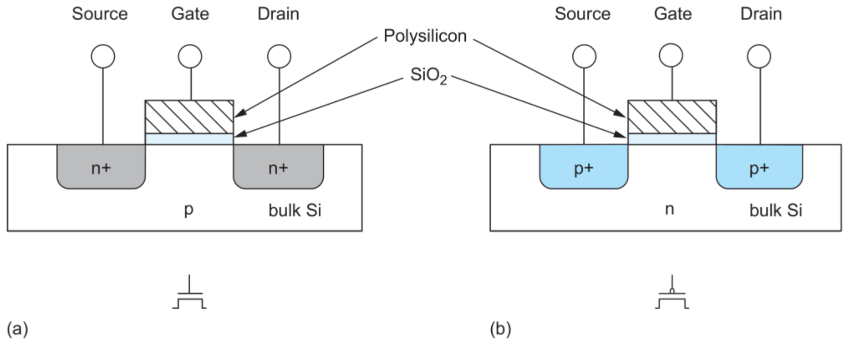
3. **How do you construct MOS transistor?**
   A Metal-Oxide-Semiconductor (MOS) structure is created by superimposing layers of conducting and insulating materials to form a structure.

4. **What is meant by MOS transistor?**
   The transistor consists of a stack of the conducting gate, an insulating layer of silicon dioxide (SiO2) and the silicon wafer (also called the substrate, body, or bulk). Gate of transistor is built from metal, so the stack is called metal oxide- semiconductor (MOS).

   Transistor operation is controlled by electric field so the device is also called Metal Oxide Semiconductor Field Effect Transistor (MOSFET).

5. **What is pull down device? (Nov 2009)**
   A device is connected to pull the output voltage to the lower supply voltage (0V) is called pull down device.

6. **Draw the schematic structure of n-MOS and p-MOS transistor with symbol.**



(a) n-MOS transistor (b) p-MOS transistor

7. **What is pull up device?**
   A device is connected to pull the output voltage to the upper supply voltage usually V_{DD} is called pull up device.

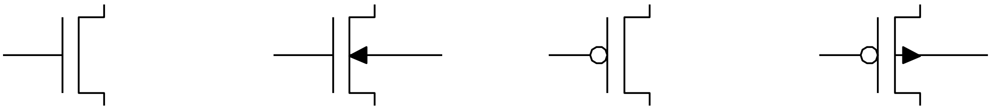
8. **Compare nMOS and pMOS devices. (or) How CMOS act as a switch? (Nov 2007) [April/May – 2023]**

✓ In nMOS, electrons are the majority carriers.

When the gate of an nMOS transistor is high, the transistor is ON. When the gate is low, the nMOS transistor is OFF.

✓ In pMOS, holes are the majority carriers.
When the gate of a pMOS transistor is low, the transistor is ON. When the gate is high, the pMOS transistor is OFF.

nMOS Symbol pMOS Symbol



9. Compare depletion and enhancement mode devices. [Nov/Dec-2007]

Depletion mode: The mode at which devices conduct with zero gate bias are called depletion mode.

Enhancement mode: The mode at which devices that are normally cut-off (i.e., non-conducting) with zero gate bias are called as enhancement mode.

10. Why nMOS technology is preferred more than pMOS technology?

The nMOS technology is preferred more than pMOS technology, because n- channel transistor has greater switching speed when compared to pMOS transistor.

11. Define threshold voltage of MOSFET. (April 2019) [Nov 2019]

The threshold voltage is the minimum gate to source voltage that is needed to create a conducting path between the source and drain terminals.

12. What is Moore's Law?

The Moore's law states that number of transistor on an integrated circuit will double every 18 months.

13. What are the three types of modes of MOS transistor? (or) Give the different modes of operation of MOS transistor. (or) What are the different MOS layers? (Nov 2009)

Three types of modes of MOS transistor are accumulation mode, depletion mode and inversion mode.

14. What is meant by accumulation mode in MOS transistor?

If a negative voltage is applied to the gate, so there is negative charge on the gate. The positively charged holes are attracted to the region beneath (below) the gate. This is called the accumulation mode.

15. What is meant by depletion mode in MOS transistor?

If a small positive voltage is applied to the gate, there are positive charges on the gate. The holes in the body are repelled from the region directly beneath (below) the gate, resulting in a depletion region forming below the gate.

16. What is meant by inversion mode (or) inversion layer in MOS transistor?

If applying higher positive voltage exceeding a threshold voltage (Vt), attracting more positive charges to the gate.

The holes are repelled and some free electrons in the body are attracted to the region beneath (below) the gate. This conductive layer of electrons in the p-type body is called the inversion layer.

17. List the different operating regions of MOS system. (May2012)

Three operating regions of MOS transistor (or) system are

(i) Cut off region or subthreshold region

(ii) Linear region

(iii) Saturation region

18. When the channel is said to be pinched –off? (May 2010)

If voltage between drain and source V_{ds} becomes sufficiently large that $V_{gd} < V_t$, the channel is no longer inverted near the drain and channel becomes pinched off.

Where V_g = gate-to-drain voltage, V_t= Threshold voltage.

19. When will nMOS transistor operates in cutoff region?

If $V_{gs} < V_t$, the transistor is cutoff (OFF).

Where V_{gs} = gate-to-source voltage, V_t= Threshold voltage.

20. When will nMOS transistor operates in linear region?

If $V_{gs} > V_t$, the transistor turns ON. If V_{ds} is small, the transistor acts as a linear resistor in which the current flow is proportional to V_{ds}.

Where V_{gs} = gate-to-source voltage, V_t= Threshold voltage & V_{ds}= drain-to-source voltage

21. When will nMOS transistor operates in saturation region?

If $V_{gs} > V_t$ and V_{ds} is large, the transistor acts as a current source in which the current flow becomes independent of Vds .

Where V_{gs} = gate-to-source voltage, V_t= Threshold voltage & V_{ds}= drain-to-source voltage.

22. Determine whether an nMOS transistor with a threshold voltage of 0.7V is operating in the saturation region if V_{gs}= 2V and V_{ds}=3V.(Nov 2011)

Condition for saturation region is $(V_{gs} – V_t) < V_{ds}$, So This nMOS transistor operated in the saturation region.

23. Give the expression for drain current (I_{ds}) for different modes of operation of MOS transistor.

$$I_{ds} = \begin{cases} 0 & V_{gs} < V_t & \text{cutoff} \\ \beta\left(V_{gs} - V_t - V_{ds}/2\right)V_{ds} & V_{ds} < V_{dsat} & \text{linear} \\ \frac{\beta}{2}\left(V_{gs} - V_t\right)^2 & V_{ds} > V_{dsat} & \text{saturation} \end{cases}$$

24. List the different capacitances of a MOS transistor.

Capacitances of a MOS transistor are

(i) Parasitic Capacitances or diffusion capacitances: C_{sb} (Source to body capacitance) & C_{db} (Drain to body capacitance)

(ii) Intrinsic capacitance: The intrinsic capacitance has three components representing the different terminals connected to the bottom plate are C_{gb} (gate-to-body), C_{gs} (gate-to-source), and C_{gd} (gate-to-drain).

(iii) Overlap capacitance: C_{gsol} (overlap capacitance to the source) & C_{gdol} (overlap capacitance to the drain)

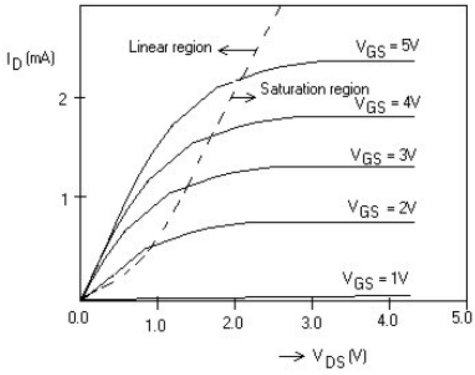
25. Define body effect (or) substrate bias effect. [May/June-2009] [Apr/may-2010]

What is meant by body effect? (NOV. 2014)

Define body bias effect. (Nov 2016)

Threshold voltage (V_t) is not constant with respect to voltage difference between substrate and source of MOS transistor. This is known as body effect. It is otherwise known as substrate bias effect.

26. Draw the I-V characteristics of MOS transistor. (May 2012)



27. What are the secondary effects (or) Non ideal effects of MOS transistor?[May 2014]

Secondary effects are

- Mobility degradation
- Velocity saturation
- Channel length modulation
- Body effect
- Subthreshold conduction
- Junction leakage current
- Tunneling
- Short channel effect

28. What is velocity saturation effect? (April 2018)

Carriers approach a maximum velocity (v_{sat}) when high fields are applied. This phenomenon is called velocity saturation.

29. Define channel length modulation.(Nov 2011, April 2016, May 2017)

Channel length modulation defines effective length of the conductive channel. It is modulated by the external applied V_{ds}. Increasing $V_{ds,}$ causes the depletion region at the drain junction to grow and thus reduces the length of the effective channel.

In Saturation region, I_{ds} is $I_{ds} = \frac{\beta}{2} V_{GT}^2 \left(1 + \frac{V_{ds}}{V_A} \right)$

30. What is body effect coefficient?(May 2011)

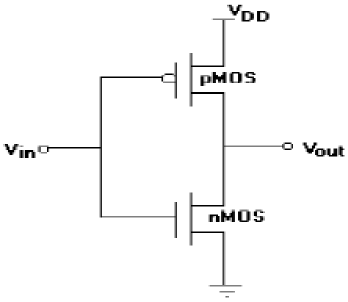
Body effect coefficient (γ) is

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

31. What do you mean by propagation delay time? (May 2017)

Propagation delay time (t_{pd}) (or) Maximum delay is defined as maximum time from the input crossing 50% to the output crossing 50%.

32. Draw the circuit of a CMOS inverter.



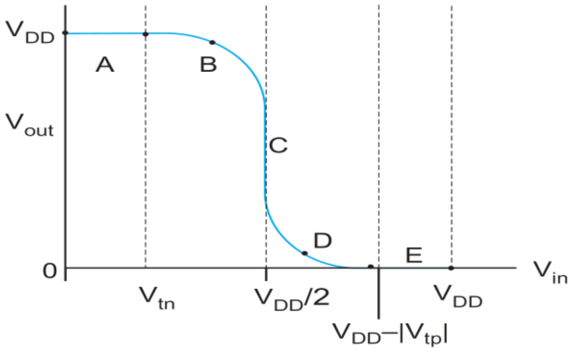
33. What are the advantages of CMOS inverter over the other inverter configurations?

a. The steady state power dissipation of the CMOS inverter circuit is negligible.

b. The voltage transfer characteristic (VTC) exhibits a full output voltage wing between 0V and V_{DD}. This results in high noise margin.

34. Draw the DC transfer characteristics of CMOS inverter. (NOV.2013, APRIL-2015)

DC transfer characteristics of CMOS inverter:



35. Define Noise margin. (May'13)

Noise margin is the amount of noise that a CMOS circuit could withstand without affect the operation of circuit.

It is basically the difference between signal value and the noise value.

NM_L (NOISE MARGIN low) = V_{IL} - V_{OL}

NM_H(NOISE MARGIN high) = V_{OH} - V_{IH}

36. What are the steps involved in the process of IC fabrication? (May 2010)

Steps involved in IC fabrication:

- Silicon wafer Preparation
- Epitaxial Growth
- Oxidation
- Photolithography

- Diffusion
- Ion Implantation
- Isolation technique
- Metallization
- Assembly processing & Packaging

37. What are the different fabrication processes available to CMOS technology?

Different CMOS processes are

- p-well process, n-well process, Twin-tub process, and Silicon On Insulator (SOI)

38. What is twin-tub process? Why it is called so?

Twin-tub process is one of the CMOS technologies in which two wells are available. The tub is also known as well. As two tubs are formed, this process is known as twin-tub process.

39. What do you mean by lateral scaling?

Lateral spacing in process of scaling the separation region between interconnect layers, keeping thickness $t_{ox,}$, length and width as constant value.

40. What is meant by scaling? [Nov/Dec-2013]

Scaling is reducing feature size of transistor. The transistor size has reduced by 30% every two to three years. As transistors become smaller, they switch faster, dissipate less power, and are cheaper to manufacture.

41. List different types of scaling. (NOV./DEC. 2014) [APRIL-2015]

List the scaling principles. (April 2018)

Types of scaling are transistor scaling and interconnect scaling.

Types of transistor scaling are lateral scaling, constant field scaling and constant voltage scaling.

42. What are the advantages of scaling? [APRIL-2015]

Advantages of scaling are better reliability, reducing complexity and reducing IC size.

43. What is the influence of voltage scaling on power and delay? [Apr/May-2011]

Due to voltage scaling, the power dissipation will be reduced with the increase in delay (i.e) speed decreases.

44. List out the limitations of the constant voltage scaling. (Nov 2015)

Constant voltage scaling is increasing the electric fields in devices.

Voltage scaling has dramatically slowed down due to leakage. This may ultimately limit CMOS scaling.

45. Define SSI, MSI, LSI and VLSI. (May 2009)

Small scale Integration:

✓ *Small-Scale Integration* (SSI) circuits have less than 10 gates. Example: 7404 inverter.

Medium scale Integration:

✓ *Medium-Scale Integration* (MSI) circuits have up to 1000 gates. Example: 74161 counter.

Large scale Integration:

✓ *Large-Scale Integration* (LSI) circuits have up to 10,000 gates. Example: 8-bit microprocessor (8085).

Very large scale Integration:

✓ Very large scale Integration (VLSI) with gates counting upto lakhs. Example: 16-bit microprocessor (8086).

Ultra large scale Integration:

✓ Ultra Large-Scale Integration (ULSI) is the process of integrating millions of transistors on a single silicon semiconductor microchip.

46. Give objective of layout design rule.

The main objective of the layout rule is to build reliably functional circuits in as small area as possible.

47. Give propagation delay expression of CMOS inverter.

Expression of delay for rising output of CMOS inverter is $t_{PLH} = 0.69\ R_P.C_L$

Expression of delay for falling output of CMOS inverter is $t_{PHL} = 0.69\ R_N.C_L$

Propagation delay of CMOS inverter is $t_P = (t_{PLH} + t_{PHL}) / 2$

48. Why NMOS device conducts strong zero and weak one? (NOV 2018)

The nMOS transistors pass 0's well but 1's poorly.

nMOS transistors attempting to pass a 1 never pull the source above $V_{DD} – V_{tn}$. This loss is called a threshold drop.

49. By What factor R_{DS} should be scaled, if constant electric filed scaling is employed? [Nov/Dec-2022]

The parameter R_{DS} **[** Drain Source Resistance] is scaled by 1 in constant electric field scaling.

50. List the scaling principles.

- Count of gates on chip.
- Minimum size of device.
- Power dissipation
- Maximum frequency of operation
- Die size.
- Types:
- Full scaling
- Fixed voltage scaling
- Lateral scaling

51. Why nMOS transistor is selected as pull down transistor?

- A static CMOS gate has an nMOS pull-down network to connect the output to 0 (GND).
- An NMOS device pulls the output all the way down to GND, while a PMOS lowers the output no further than $|V_{Tp}|$ — the PMOS turns off at that point, and stops contributing discharge current. NMOS transistors are hence the preferred devices in the PDN.
- NMOS device fails to raise the output above V_{DD}-V_{Tn} .

52. Name the merits of Scaling Principles.

- Minimum feature size
- Die size
- Production cost

- How Many numbers of gates are available in one chip, that also calculated.

53. What is meant channel length modulation in NMOS transistor?

The current between drain and source terminals is constant and independent of the applied voltage over the terminals. This is not entirely correct. The effective length of the conductive channel is actually modulated by the applied $\mathbf{V_{DS}}$, increasing $\mathbf{V_{DS}}$ causes the depletion region at the drain junction to grow, reducing the length of effective channel.

The increase of depletion layer width at the drain as the drain voltage is increased. This lead to a shorter channel length and an increased drain current is called channel length modulation.

$$L_{effective} = L - \sqrt{2\frac{\varepsilon_{Si}}{qN_A}(V_{ds} - (V_{gs} - V_t))}$$

54. What are the different layers in MOS transistor?

The layers are Substrate, diffused Drain & Source, Insulator (SiO2) & Gate.

55. What are the different operating regions for an MOS transistor?

- Cutoff Region
- Non-Saturated
- (Linear) Region
- Saturated Region

56. What is Enhancement mode transistor?

The device that is normally cut-off with zero gate bias is called Enhancement mode transistor.

57. What is Depletion mode device?

The Device that conducts with zero gate bias is called Depletion mode device.

58. When the channel is said to be pinched off?

If a large $\mathbf{V_{ds}}$ is applied, this voltage will deplete the inversion layer. This Voltage effectively pinches off the channel near the drain.

59. What are the steps involved in manufacturing of IC?

- Silicon wafer Preparation
- Epitaxial Growth
- Oxidation
- Photolithography
- Diffusion
- Ion Implantation
- Isolation technique
- Metallization
- Assembly processing & Packaging

60. What is meant by Epitaxy?

Epitaxy means arranging atoms in single crystal fashion upon a single crystal substrate.

61. What are the processes involved in photo lithography?

Masking process

Photo etching process.

62. What is the purpose of masking in fabrication of IC?

Masking is used to identify the location in which Ion Implantation should not take place.

63. What are the materials used for masking?

Photo resist, Si02, SiN, Poly Silicon.

64. What are the types of Photo etching?

Wet etching

Dry etchings are the types of photo etching.

65. What is diffusion process? What are doping impurities?

Diffusion is a process in which impurities are diffused into the Silicon chip at 1000^0C temperature. B203 and P205 are used as impurities used.

66. What is Ion Implantation process?

It is process in which the Si material is doped with an impurity by making the accelerated impurity atoms to strike the Si layer at high temperature.

67. What are the various Silicon wafer Preparation?

- Crystal growth & doping
- Ingot trimming & grinding
- Ingot slicing
- Wafer polishing & etching
- Wafer cleaning.

68. What are the different types of oxidation?

The two types of oxidation are Dry & Wet Oxidation.

69. What is Isolation?

It is a process used to provide electrical isolation between different components and interconnections.

70. Give the different types of CMOS process.

- p well process
- n well process
- twin tub process
- SOI process

71. What is Channel stop Implantation?

In n well fabrication, n well is protected with the resist material. (Because, it should not be affected during Boron implantation). Then Boron is implanted except n well. The above said process is done using photo resist mask. This type of implantation is known as Channel stop implantation.

72. What is LOCOS?

LOCOS mean Local Oxidation of Silicon. This is one type of oxide construction.

73. What is SWAMI?

SWAMI means Side Wall Masked Isolation. It is used to reduce bird's beak effect.

74. What is LDD?

LDD means Lightly Doped Drain Structures. It is used for implantation of n region In n-well process.

75. What is Twin tub process? Why it is called so?

Twin tub process is one of the CMOS technologies. Two wells (the other name for well is Tub) are created in this process. So, because of these two tubs, this process is known as Twin tub process.

76. What are the steps involved in Twin tub process?

- Tub Formation
- Thin oxide Construction
- Source & Drain Implantation
- Contact cut definition
- Metallization.

77. Name the special features of Twin tub process.

In Twin tub process, Threshold voltage, body effects of n and p devices are independently optimized.

78. List out the advantages of Twin tub process?

Advantages of Twin tub process are (1) Separate optimized wells are available. (2) Balanced performance is obtained for n and p transistors.

79. What is SOI? What is the material used as Insulator?

SOI means Silicon on Insulator. In this process, a Silicon based transistor is built on an insulating material like Sapphire or SiO.

80. What are the advantages and disadvantages of SOI process?

Advantages of SOI process:

- There is no well formation in this process.
- There is no Field Inversion problem.
- There is no body effect problem.

Disadvantages of SOI process:

- It is very difficult to protect inputs in this process.
- Device gain is low.
- The coupling capacitance between wires always exists.

81. What are the advantages of CMOS process?

- Low Input Impedance
- Low delay Sensitivity to load.

82. Define Short Channel devices.

Transistors with Channel length less than 3- 5 microns are termed as Short channel devices. With short channel devices the ratio between the lateral & vertical dimensions are reduced.

83. What are the advantages of Silicon-on-Insulator process?

- No Latch-up
- Due to absence of bulks transistor structures are denser than bulk silicon.

84. What are the advantages of CMOS process?

- Low power Dissipation
- High Packing density
- Bi directional capability

85. What is the fundamental goal in Device modeling?

To obtain the functional relationship among the terminal electrical variables of the device that is to be modeled.

86. What is CMOS Technology?

The fabrication of an IC using CMOS transistors is known as CMOS Technology. CMOS transistor is nothing but an inverter, made up of an nMOS and pMOS transistor connected in series.

87. Give the advantages of CMOS IC?

- Size is less
- High Speed
- Less Power Dissipation

88. What are four generations of Integration Circuits?

- SSI (Small Scale Integration)
- MSI (Medium Scale Integration)
- LSI (Large Scale Integration)
- VLSI (Very Large Scale Integration)

89. Give the variety of Integrated Circuits.

- More Specialized Circuits
- Application Specific Integrated Circuits (ASICs)
- Systems On Chips.

90. Why NMOS technology is preferred more than PMOS technology?

N-channel transistors have greater switching speed when compared to PMOS transistors. Hence, NMOS is preferred than PMOS.

****

UNIT II - COMBINATIONAL LOGIC CIRCUITS

Propagation Delays, stick diagram, Layout diagrams, Examples of combinational logic design, Elmore's constant, Static Logic Gates, Dynamic Logic Gates, Pass Transistor Logic, Power Dissipation, Low Power Design principles.

2.1: Delay estimation

- ❖ **Draw a CMOS inverter. Analyze the switching characteristics during rise time when V_{in} change from high to low. (April 2019-7M)**
- ❖ **Derive an expression for the rise time, fall time and propagation delay of a CMOS inverter. (DEC 2013, APRIL-2015) [Nov 2019] [Nov/Dec 2022]**

- ✓ Important definitions for delay estimation:

Propagation delay time (t_{pd}):

- ✓ Propagation delay time is defined as maximum time from the input crossing 50% to the output crossing 50%.

Contamination delay time (t_{cd}):

- ✓ Contamination delay time is defined as minimum time from the input crossing 50% to the output crossing 50%.

Rise time (t_r):

- ✓ Rise time is defined as time for a waveform to rise from 20% to 80% of its steady-state value

Fall time (t_f):

- ✓ Fall time is defined as time for a waveform to fall from 80% to 20% of its steady-state value
- ✓ Edge rate is average of rise and fall time, (t_{rf}) = (t_r + t_f)/2

Delay estimation response curve:

- ✓ When an input changes, the output will retain its old value for at least the contamination delay and take on its new value in, at most the propagation delay.
- ✓ Delays for the output rising is t_{pdr} /t_{cdr} and the output falling is t_{pdf} /t_{cdf} .
- ✓ Rise/fall times are also called as slopes or edge rates.
- ✓ Propagation and contamination delay times are also called as max-time and min-time respectively.

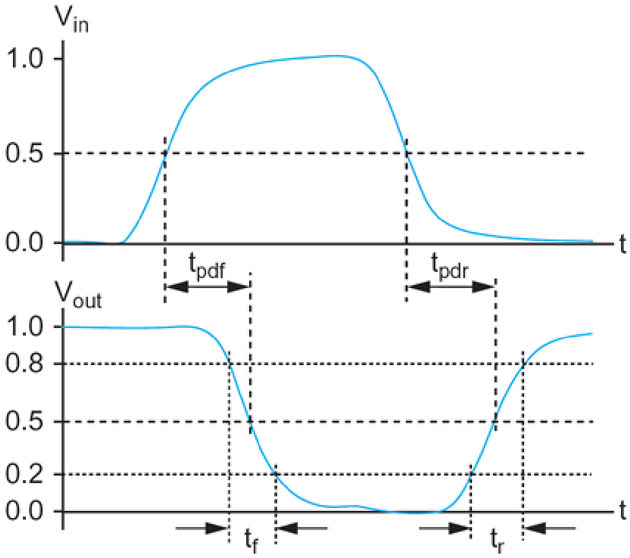


Figure: Delay estimation of CMOS inverter

- ✓ The gate that charges or discharges a node is called the driver. The gates and wires being driven, are called the load. Propagation delay is usually called as delay.
- ✓ Arrival times and propagation delays are defined separately for rising and falling transitions.
- ✓ The delay of a gate may be different from different inputs. Earliest arrival times can also be computed based on contamination delays.
- ✓ Expression of delay for rising output is $t_{PLH} = 0.69\ R_P.C_L$
  Where, R_P – effective resistance of pMOS transistor
  C_L - load capacitance of CMOS inverter.
- ✓ Expression of delay for falling output is $t_{PHL} = 0.69\ R_N.C_L$
  Where, R_N – effective resistance of nMOS transistor
- ✓ Propagation delay of CMOS inverter is $t_P = (t_{PLH} + t_{PHL}) / 2$

2.1.1: RC Delay Model:

Discuss in detail about the resistive and capacitive delay estimation of a CMOS inverter circuit. (MAY 2013) (or)
Briefly explain about the RC delay model.

- ✓ RC delay model approximates the nonlinear transistor I-V and C-V characteristics with an average resistance and capacitance over the switching range of the gate.

Effective Resistance:

- ✓ The RC delay model treats a transistor as a switch in series with a resistor.
- ✓ The effective resistance is the ratio of V_{ds} to I_{ds}.
- ✓ A unit nMOS transistor is defined to have effective resistance R.
- ✓ An nMOS transistor of k times unit width has resistance R/k, because it delivers k times as much current.
- ✓ A unit pMOS transistor has greater resistance, generally in the range of 2R–3R, because of its lower mobility.

- ✓ According to the long-channel model, current decreases linearly with channel length (L) and hence resistance is proportional to L.

Gate and Diffusion Capacitance:

- ✓ Each transistor has gate and diffusion capacitance.
- ✓ C is the gate capacitance of a unit transistor. A transistor of k times unit width has capacitance kC.
- ✓ Diffusion capacitance depends on the size of the source/drain region.
- ✓ Wider transistors have proportionally greater diffusion capacitance. Increasing channel length, increases gate capacitance proportionally but does not affect diffusion capacitance.

Equivalent RC Circuits:

- ✓ Figure shows equivalent RC circuit models for nMOS and pMOS transistors of width k with contacted diffusion on both source and drain.
- ✓ The pMOS transistor has approximately twice the resistance of the nMOS transistor, because holes have lower mobility than electrons.

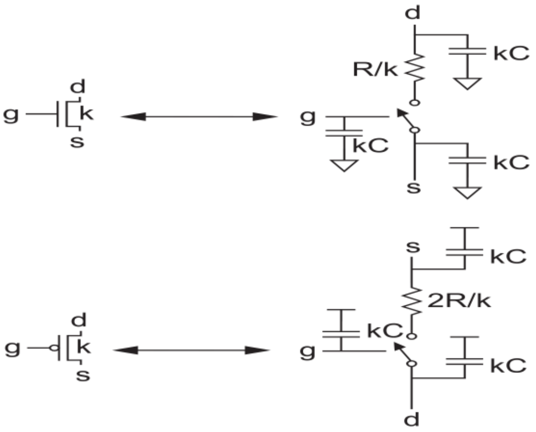


Figure: RC model of nMOS &pMOS transistors

****

2.2: Stick diagram

- ❖ **Explain about stick diagram in VLSI design. (April 2008)**
- ❖ **Draw the static diagram of CMOS inverter. (April 2019-7M)**

- ✓ A **stick diagram** is a cartoon of a chip layout. A "stick diagram" is a paper and pencil tool that use to plan the layout of a cell.
- ✓ The stick diagram resembles the actual layout, but uses "sticks" or lines to represent the devices and conductors. Figure 17, shows a stick diagram for an inverter.

- ✓ The stick diagram represents the rectangles with lines, which represent wires and component symbols.
- ✓ The stick diagram does not represent all the details of a layout, but it makes some relationship much clearer and it is simple to draw.
- ✓ Layouts are constructed from rectangles, but stick diagrams are built from cartoon symbols for components and wires.

Stick diagram Rules:

- ✓ **Rule 1:** When two or more 'sticks' of the same type cross or touch each other, that represents electrical contact.
- ✓ **Rule 2:** When two or more 'sticks' of the different type cross or touch each other, there is no electrical contact. If electrical contact is needed, we have to show the connection explicitly.
- ✓ **Rule 3:** When a poly crosses diffusion, it represents a transistor. If a contact is shown, then it is not a transistor. A transistor exists where a polysilicon (red) stick crosses either an n-diffusion (green) stick or a p-diffusion (yellow) stick.
- ✓ **Rule 4:** In CMOS, a demarcation line is drawn to avoid touching of p-diff with n-diff. All pMOS must lie on one side of the line and all nMOS will have to be on the other side.

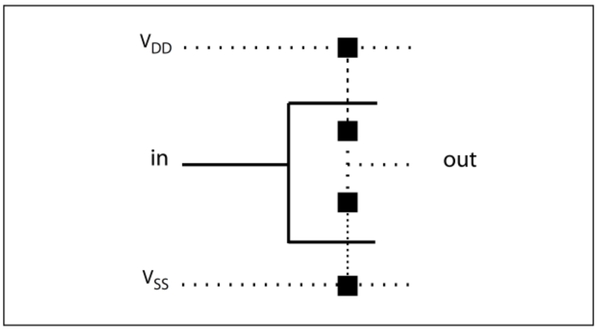


Figure 17: Stick diagram for an inverter

The symbols for wires used on various layers are shown in Figure 18.

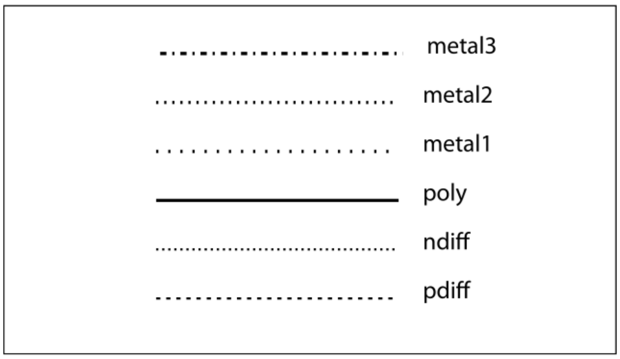


Figure 18: Symbols for wires used on various layers

- ✓ **Drawing stick diagrams in color**: Red for poly, green for n-diffusion, yellow for p-diffusion, and shades of blue for metal are typical colors.
- ✓ A few simple rules for constructing wires from straight-line segments ensure that, the stick diagram corresponds to a feasible layout.
- ✓ Wires cannot be drawn at arbitrary angles. Only horizontal and vertical wire segments are allowed.
- ✓ Two wire segments on the same layer, which cross are electrically connected.

- ✓ Vias to connect wires, which do not normally interact, are drawn as black dots.
- ✓ Figure 19, shows the stick figures for transistors.
- ✓ Each type of transistor is represented as poly and diffusion crossings, much as in the layout.

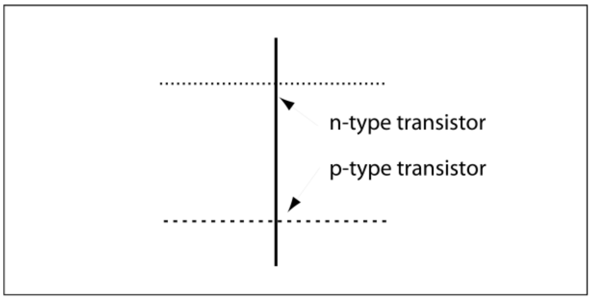
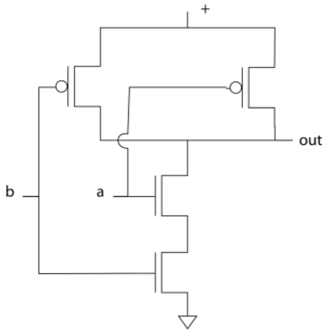


Figure 19: Stick figures for transistors

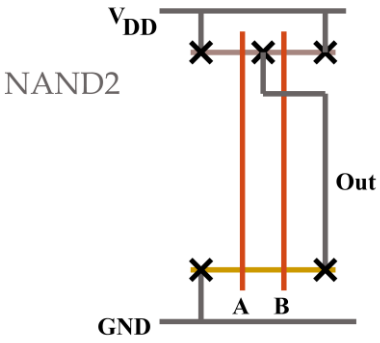
- ✓ Area and aspect ratio are also difficult to estimate from stick diagrams.
- ✓ Stick diagrams are especially important tools for layouts built from large cells and for testing the connections between cells.

Example:1

Here is the transistor schematic for a two-input NAND gate:

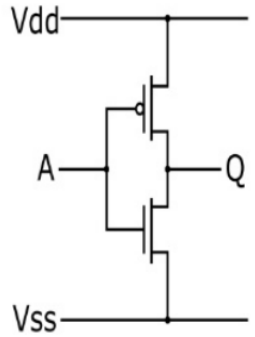


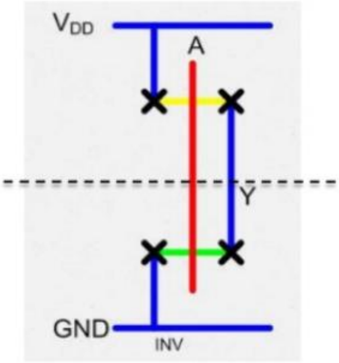
And here is a stick diagram for the two-input NAND:



Example: 2

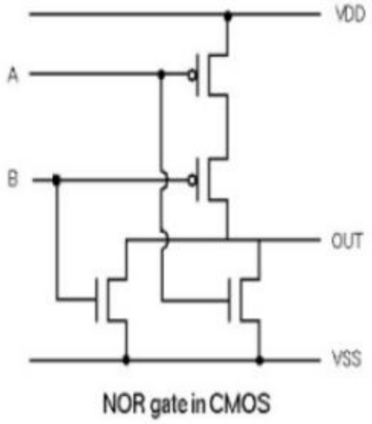
Draw the Stick diagram of CMOS Inverter

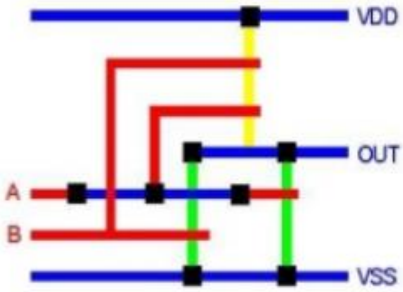




Example: 3

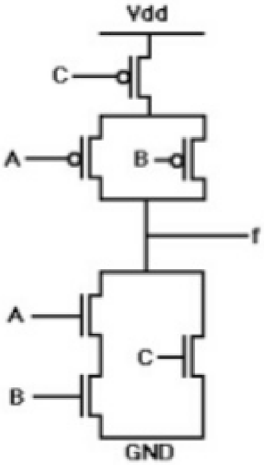
Draw the Stick diagram of CMOS NOR gate

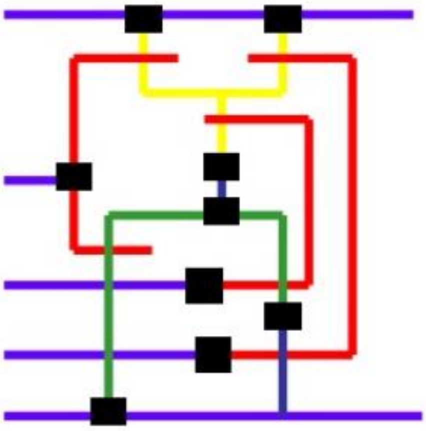




Example: 4

Draw the stick diagram of [(A . B) +C]'





****

2.3: Layout Design Rules and Gate Layouts

- ❖ **Draw and explain briefly the n-well CMOS design rules. (NOV 2007, April 2008, MAY 2014)**
- ❖ **Discuss in detail with a neat layout, the design rules for a CMOS inverter.**
- ❖ **Write the layout design rules and draw diagram for four input NAND and NOR. (Nov 2016) (April 2018)**
- ❖ **State the minimum width and minimum spacing lambda based design rules to draw the layout. (April 2019-6M)**

- ✓ Layout rules also referred to as **design rules**.
- ✓ It can be considered as prescription for preparing the photomasks, which are used in the fabrication of integrated circuits.
- ✓ The rules are defined in terms of feature sizes (widths), separations and overlaps.
- ✓ The main **objective of the layout rules is to build reliable functional circuits in as small area as possible.**
- ✓ Layout design rules describe how small features can be and how closely they can be reliably packed in a particular manufacturing process.
- ✓ Design rules are a set of geometrical specifications that dictate the design of the layout masks.
- ✓ A design rule set provides numerical values for minimum dimensions and line spacing.
- ✓ Scalable design rules are based on a single parameter (λ), which characterizes the resolution of the process. λ is generally half of the minimum drawn transistor channel length.
- ✓ This length is the distance between the source and drain of a transistor and is set by the minimum width of a polysilicon wire.

Lambda based rule (Scalable design rule):

- ✓ Lambda-based rules are round up dimensions of scaling to an integer multiple of λ.
- ✓ Lambda rules make scaling layout small. The same layout can be moved to a new process, simply by specifying a new value of λ.
- ✓ The minimum feature size of a technology is characterized as 2λ.

Micron Design Rules (Absolute dimensions):

- ✓ The MOSIS rules are expressed in terms of lambda.
- ✓ These rules allow some degree of scaling between processes.
- ✓ Only need to reduce the value of lambda and the designs will be valid in the next process down in size.
- ✓ These processes rarely shrink uniformly.
- ✓ Thus, industry usually uses the actual micron design rules for layouts.
- ✓ There are set of micron design rules for a hypothetical 65 nm process.
- ✓ We can observe that, these rules differ slightly but not immensely from lambda based rules with lambda = 0.035 micro meter.
- ✓ Upper level metal rules are highly variable depending on the metal thickness. Thicker wires require greater widths, spacing and bigger vias.

- ✓ Two metal layers in an n-well process has the following:

- Metal and diffusion have minimum width and spacing of 4 λ.
- Contacts are 2 λ × 2 λ and must be surrounded by 1 λ on the layers above and below.
- Polysilicon uses a width of 2 λ.
- Polysilicon overlaps diffusion by 2 λ where a transistor is desired and has a spacing of 1 λ away where no transistor is desired.
- Polysilicon and contacts have spacing of 3 λ from other polysilicon or contacts.
- N-well surrounds pMOS transistors by 6 λ and avoids nMOS transistors by 6 λ.

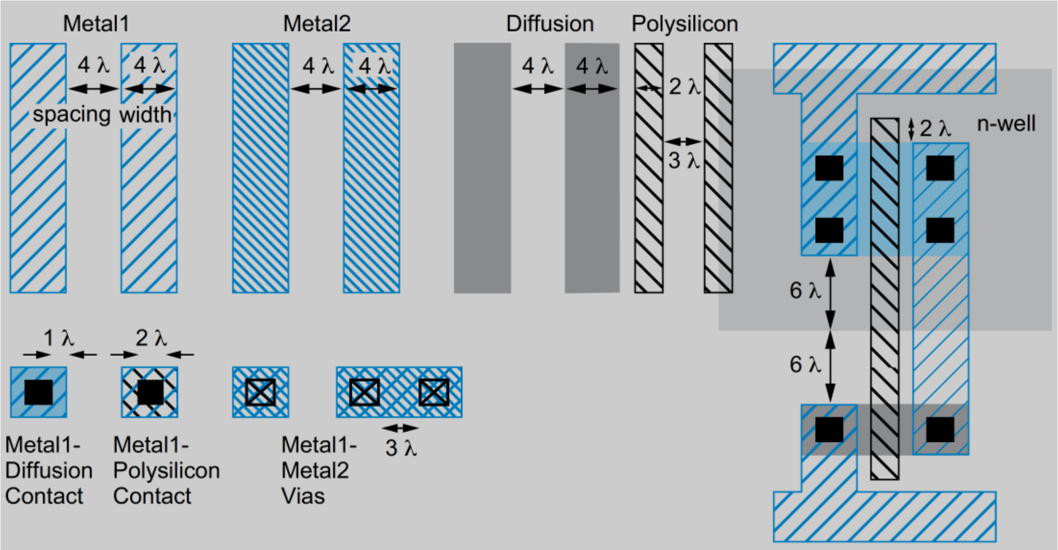


Figure: Simplified λ -based design rules with CMOS inverter layout diagram

Design Rule:

Well Rules:

- ✓ The n-well is usually a deeper implant than the transistor source/drain implants.
- ✓ Therefore, it is necessary to provide sufficient clearance between the n-well edges and the adjacent *n*+ diffusions.

Transistor Rules:

- ✓ CMOS transistors are generally defined by at least four physical masks.
- ✓ There are active (also called diffusion, diff, thinox, OD, or RX), n-select (also called n-implant, n-imp, or nplus), p-select (also called p-implant, pimp, or pplus) and polysilicon (also called poly, polyg, PO, or PC).
- ✓ The active mask defines all areas, where n- or p-type diffusion is to be placed *or* where the gates of transistor are to be placed.

Contact Rules:

- ✓ There are several generally available contacts:
  - Metal to p-active (p-diffusion)
  - Metal to n-active (n-diffusion)
  - Metal to polysilicon
  - Metal to well or substrate

Metal Rules:

- ✓ Metal spacing may vary with the width of the metal line.

- ✓ Metal wire width of minimum spacing may be increased. This is due to etch characteristics versus large metal wires.

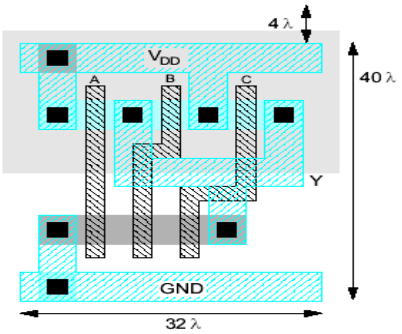
Via Rules:

- ✓ Processes may allow vias to be placed over polysilicon and diffusion regions.
- ✓ Some processes allow vias to be placed within these areas, but do not allow the vias to the boundary of polysilicon or diffusion.

Example: NAND3

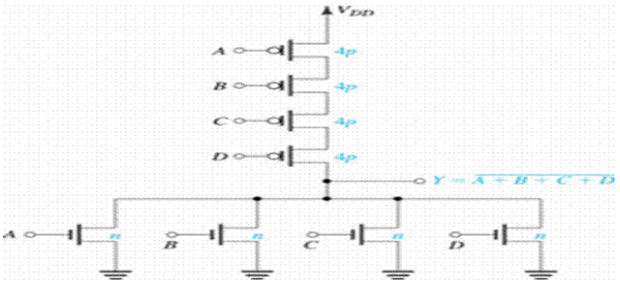
Draw the gate layout diagram of NAND. (May 2017)

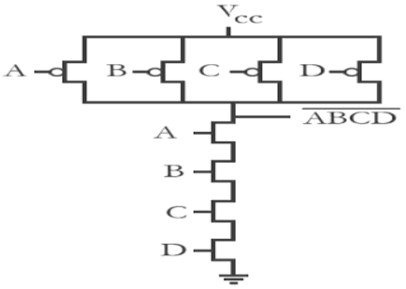
- Horizontal N-diffusion and p-diffusion strips
- Vertical polysilicon gates
- Metal1 V_{DD} rail at top
- Metal1 GND rail at bottom



- ❖ **Draw diagram for four input NAND and NOR gate. (Nov 2017)**

4 input NOR gate **4 input NAND gate**



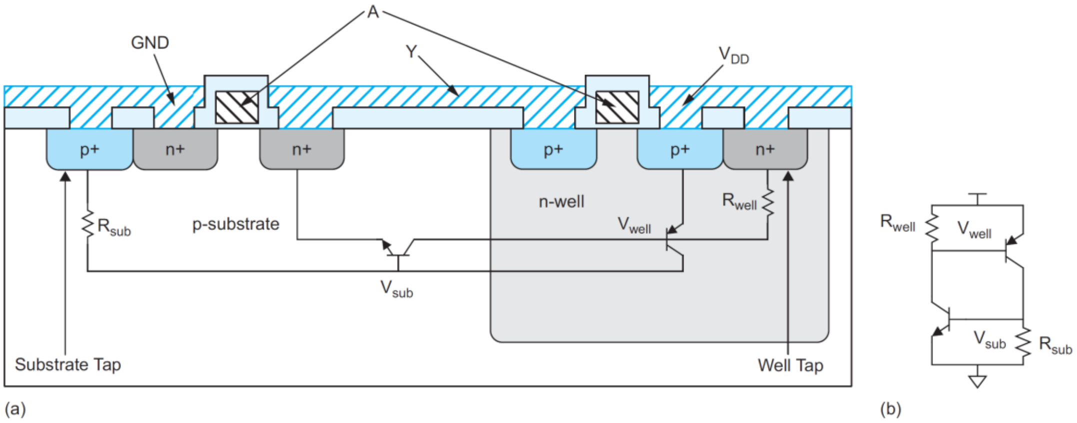


****

2.4: Latchup problem:

- ❖ ***Discuss the orgin of latch up problems in CMOS circuits with necessary diagrams. Explain the remedial measures. (Nov 2007, April 2008)***

- ✓ A CMOS process is slowed down by developing low-resistance paths between V_{DD} and GND, causing catastrophic meltdown. The phenomenon is called latchup.
- ✓ Latchup problem arises when parasitic bipolar transistors are formed by the substrate, well and diffusion.
- ✓ The cause of the latchup effect can be understood by examining the process cross-section of a CMOS inverter, as shown in Figure (a).
- ✓ The schematic shows, a circuit composed of an npn-transistor, a pnp-transistor, and two resistors connected between the power and ground rails (Figure (b)).



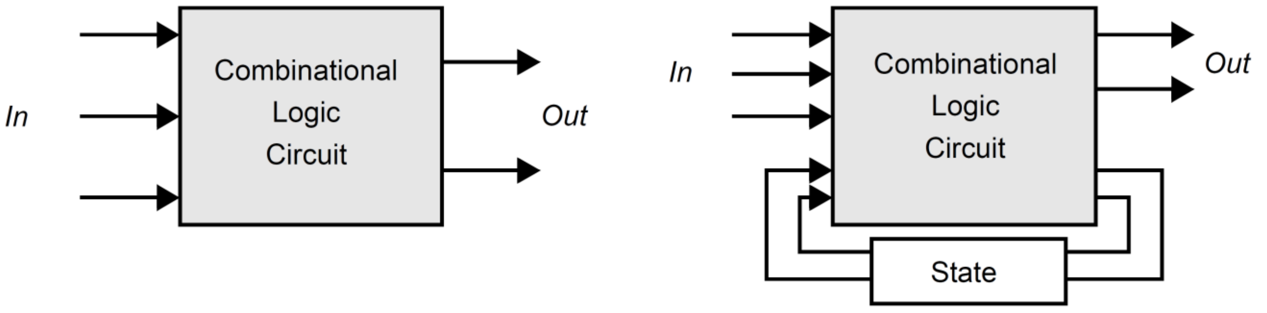
- ✓ The npn transistor is formed between the grounded n-diffusion source of the nMOS transistor, the p-type substrate and the n-well.
- ✓ The resistors are due to the resistance through the substrate or well to the nearest substrate and well taps.
- ✓ The cross-coupled transistors form a bistable silicon-controlled rectifier (SCR). Both parasitic bipolar transistors are OFF.
- ✓ Latchup can be triggered, when transient currents flow through the substrate during normal chip power-up.

- ✓ Latchup prevention is easily accomplished by
  - Minimizing Rsub and Rwell.
  - Use of guard rings
- ✓ SOI process avoids latchup entirely, because they have no parasitic bipolar structures.

2.5 Introduction (Combinational Logic Circuit):

CMOS logic

- Digital logics are divided into combinational and sequential circuits.
- Combinational circuits are circuits where outputs depend only on the present inputs.
- For sequential or regenerative circuit, the output is not only a function of the current input data, but also of previous values of the input signals.
- A sequential circuit includes a combinational logic portion and a memory module that holds the state. Example are registers, counters and memory.
- The building blocks for combinational circuits are logic gates, while the building blocks for sequential circuits are registers and latches.
- The delay of a logic gate depends on its output current I, load capacitance C and output voltage swing ΔV.



(a) Combinational (b) Sequential

- Alternative (ratioed circuits, dynamic circuits and pass transistor circuits) CMOS logic configurations are called circuit families.
- nMOS transistors provide more current than pMOS for the same size and capacitance, so nMOS networks are preferred.

Examples of combinational circuits

(i) CMOS inverter:

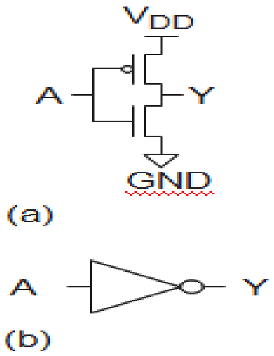


Figure: Inverter (a) schematic (b) symbol $Y = \bar{A}$

(ii) **Two input NAND gate:**

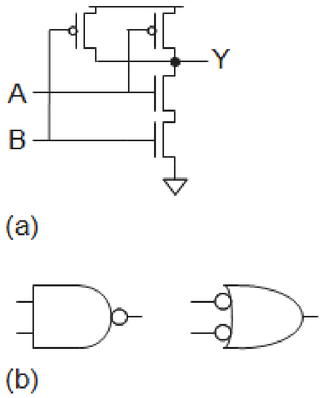


Figure: 2 input NAND gate (a) schematic (b) symbol

(iii) **Three input NAND gate:**

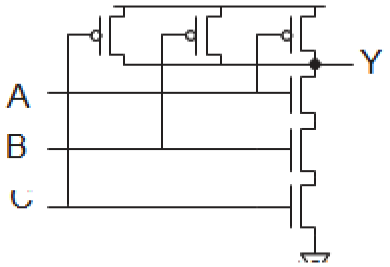


Figure: 3 –input NAND gate Schematic Y=A.B.C

(iv) **Two input NOR gate:**

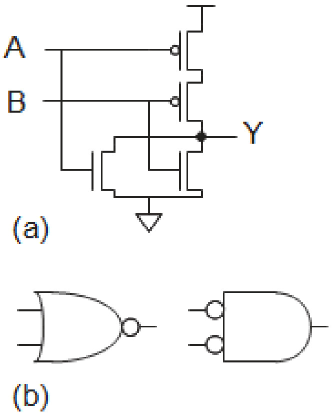
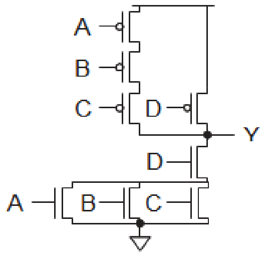


Figure: 2-input NOR gate (a) schematic (b) Symbol Y = ~~A + B~~

Example:

Sketch a static CMOS gate computing $Y = \overline{(A + B + C) \cdot D}$.



****

2.6: Elmore's Delay

What is meant by Elmore's delay and give expression for Elmore's delay?

- The Elmore delay model estimates the delay from a source, switching to one of the leaf nodes. Delay is the sum over each node i of the capacitance C_i on the node multiplied by the effective resistance R.

Propagation delay time :

$$t_{pd} \approx \sum_{\text{nodes } i} R_{i-to-source} C_i$$
$$= R_1 C_1 + (R_1 + R_2) C_2 + \ldots + (R_1 + R_2 + \ldots + R_N) C_N$$

- Delay of an ideal fanout-of-1 inverter with no parasitic capacitance is τ = 3*RC.*
- The normalized delay *d* relative to this inverter delay:

$$d = \frac{t_{pd}}{\tau}$$

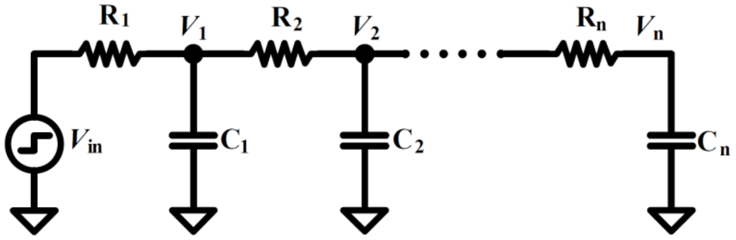
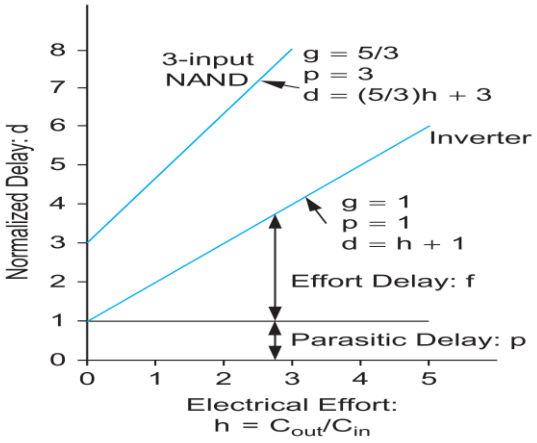


Figure: RC delay equivalent for series of transistors

Linear delay model

- The RC delay model is one, where delay is a linear function of the fanout of a gate.
- The normalized delay of a gate can be expressed in units of ϒ as d = f + p.
  Where p is the parasitic delay inherent to the gate when no load is attached.
  f is the effort delay or stage effort that depends on the complexity.
- Effort delay of the gate is f = gh.
  Where g is the logical effort (An inverter has a logical effort of 1).

- Logical effort is defined as the ratio of the input capacitance of a gate to the input capacitance of an inverter delivering the same output current.
- h is the fanout or electrical effort. Electrical effort is defined as ratio of the output capacitance to input capacitance.
- More complex gates have greater logical efforts, indicating that they take longer time to drive a given fanout.
- For example, the logical effort of the 3-input NAND gate is 5/3.
- The electrical effort can be computed as $h = \frac{C_{out}}{C_{in}}$
  Where C_{out} is the capacitance of the external load being driven and C_{in} is the capacitance of the gate.
- Normalized delay vs electrical effort for an idealized inverter and 3-input NAND gate shown in diagram.
- The y-intercepts indicate the parasitic delay. The slope of the lines is the logical effort.
- The inverter has a slope of 1. The NAND gate has a slope of 5/3.



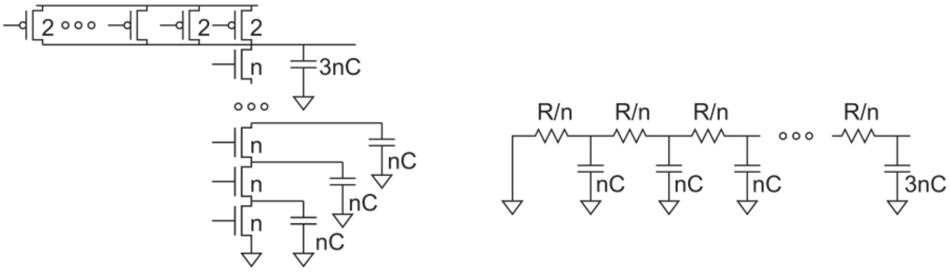
Design a four input NAND gate and obtain its delay during the transition from high to low. (April 2018)

Figure shows a model of an n-input NAND gate in which the upper inputs were all 1 and the bottom input rises. The gate must discharge the diffusion capacitances of all of the internal nodes as well as the output.

Elmore delay is

$$t_{pd} = R(3nC) + \sum_{i=1}^{n-1} \left(\frac{iR}{n}\right)(nC) = \left(\frac{n^2}{2} + \frac{5}{2}n\right)RC$$

Figure: n-input NAND gate parasitic delay



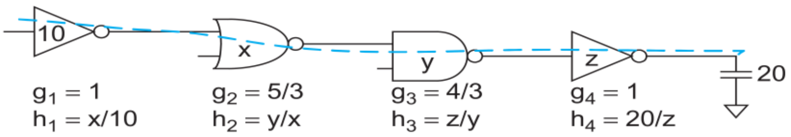
Delay for 4 input NAND gate: $\left(\frac{n^2}{2} + \frac{5}{2}n\right)RC = \left(\frac{4^2}{2} + \frac{5}{2}4\right)RC = \left(\frac{16}{2} + \frac{20}{2}\right)RC$ =18RC

Logical effort

Obtain the logical effort and path efforts of the given circuit. (April 2018)

Delay in Multistage Logic Networks:

The figure shows the logical and electrical efforts of each stage in a multistage path as a function of the sizes of each stage.



The path of interest (the only path in this case) is marked with the dashed blue line. Observe that logical effort is independent of size, while electrical effort depends on sizes.

The path logical effort G can be expressed as the products of the logical efforts of each stage along the path.

$$G = \prod g_i$$

The path electrical effort H can be given as the ratio of the output capacitance the path must drive divided by the input capacitance presented by the path

$$H = \frac{C_{out(path)}}{C_{in(path)}}$$

The path effort F is the product of the stage efforts of each stage.

$$F = \prod f_i = \prod g_i h_i$$

Introduce an effort to account for branching between stages of a path. This branching effort b is the ratio of the total capacitance seen by a stage to the capacitance on the path.

$$b = \frac{C_{onpath} + C_{offpath}}{C_{onpath}}$$

The path branching effort B is the product of the branching efforts between stages.

$$B = \prod b_i$$

The path effort (F) is defined as the product of the logical, electrical, and branching efforts of the path. The product of the electrical efforts of the stages is actually BH, not just H.

F = GBH

Compute the delay of a multistage network. The path delay D is the sum of the delays of each stage. It can also be written as the sum of the path effort delay DF

$$D = \sum d_i = D_F + P$$
$$D_F = \sum f_i$$
$$P = \sum p_i$$

The product of the stage efforts is F, independent of gate sizes. The path effort delay is the sum of the stage efforts. The sum of a set of numbers whose product is constant is minimized by choosing all the numbers to be equal.

The path delay is minimized when each stage bears the same effort. If a path has N stages and each bears the same effort, that effort must be

f˜ = gi hi = F 1 / N

Thus, the minimum possible delay of an N-stage path with path effort F and path parasitic delay P is

D = NF 1/ N + P

It shows that the minimum delay of the path can be estimated knowing only the number of stages, path effort, and parasitic delays without the need to assign transistor sizes.

The capacitance transformation formula is used to find the best input capacitance for a gate given the output capacitance it drives.

$$C_{in_i} = \frac{C_{out_i} * g_i}{\hat{f}}$$

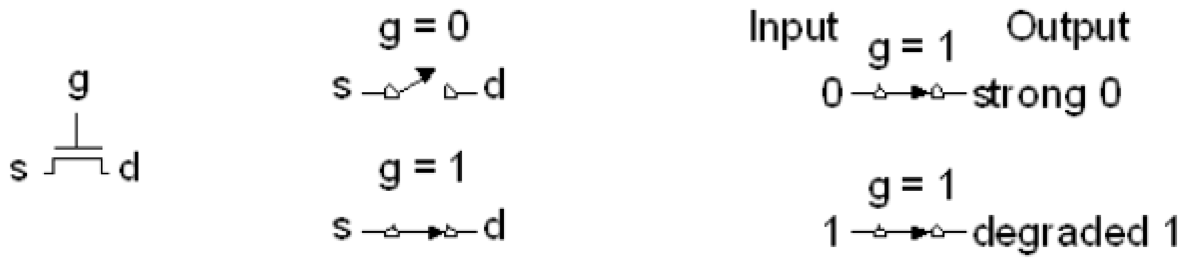
At the end of the path, apply the capacitance transformation to determine the size of each stage. Check the arithmetic by verifying that the size of the initial stage matches the specification.

****

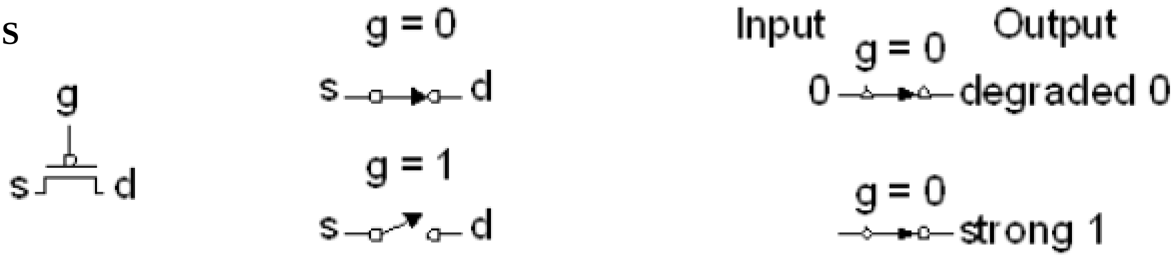
2.7 Pass Transistor

Write short notes on pass transistor.

nMos



pMOS



- Transistors can be used as switches

Signal Strength

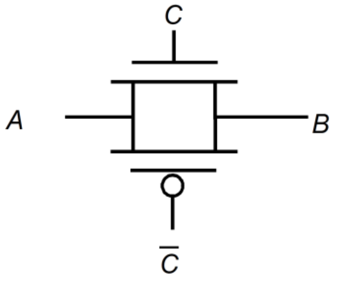
- How close it approximates ideal voltage source
- VDD and GND rails are strongest 1 and 0
- nMOS pass strong 0, But degraded or weak 1
- pMOS pass strong 1,But degraded or weak 0
- Thus NMOS are best for pull-down network
- Thus PMOS are best for pull-up network

****

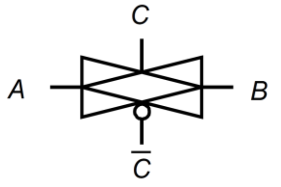
2.8 Transmission Gates

Write short notes on transmission gates (TG).

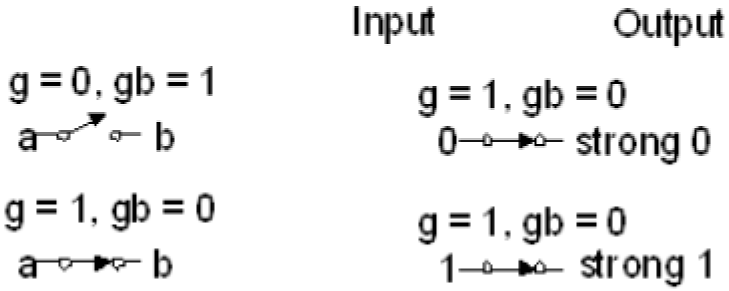
- By connecting an nMOS and a pMOS transistor in parallel, we obtain a switch that turns on when a 1 is applied to the gate terminal in which 0’s and 1’s are both passed in an acceptable fashion.
- We term this a ***transmission gate or pass gate***.



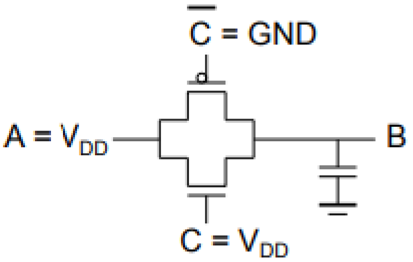
(a) Circuit

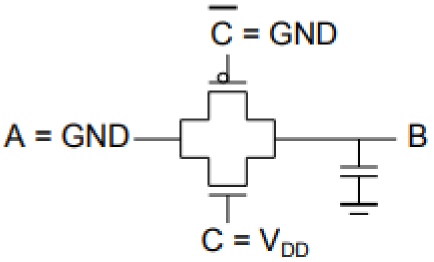


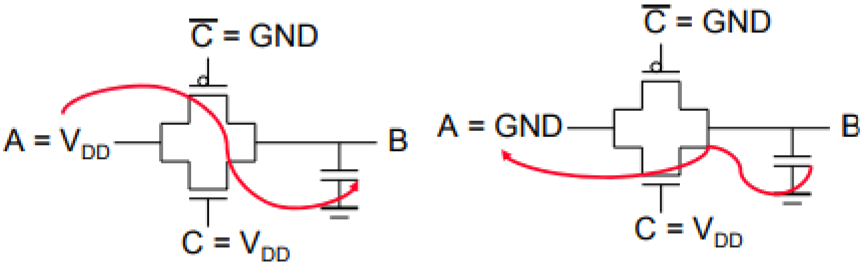
(b) Symbolic representation



- In a circuit where only a 0 or a 1 has to be passed, the appropriate transistor (n or p) can be deleted, reverting to a single nMOS or pMOS device.
- Note that, both the control input and its complement are required by the transmission gate. This is called ***double rail logic***.
- When the control input is low(control =0), the switch is open, and when the control is high (control=1) the switch is closed.







- Full swing bidirectional switch controlled by the gate signal C, A = B if C = 1

****

2.9 Cascaded CMOS Inverter

Derive the generalized expression for propagation delay of N-cascaded CMOS inverters if 'N' is even and if 'N' is odd. [Nov 2019]

- Assume a signal is available at the output of a minimum size inverter and that it is to drive a load C_L.

The average propagation delay associated with driving this load directly is,

$$t_{\text{dir}} = \frac{t_{\text{apd}} C_{\text{L}}}{C_{\text{G}}}$$

Where, t_{apd} is the average logic stage delay and C_G is the input capacitance of the reference inverter.

For any integer n ≥ 1, define **α by the expression**,

$$\alpha = \left(\frac{C_{\text{L}}}{C_{\text{G}}}\right)^{1/n}$$

Alternatively, **n can be represented in terms of α as**

$$n = \frac{\ln(C_{\text{L}}/C_{\text{G}})}{\ln \alpha}$$

This structure is composed of a cascade of n inverters each sized by the 4 : 1 sizing rule and each with a drive capability that is α times as large as the previous stage.

The width and length of the k^th^ stage can be characterized by the equations,

$$W_{\text{d}k} = \alpha^{k-1} W_{\text{d}1}$$

$$W_{\text{u}k} = W_{\text{d}k}$$

$$L_{dk} = L_{d1}$$

$$L_{uk} = 4L_{dk}$$

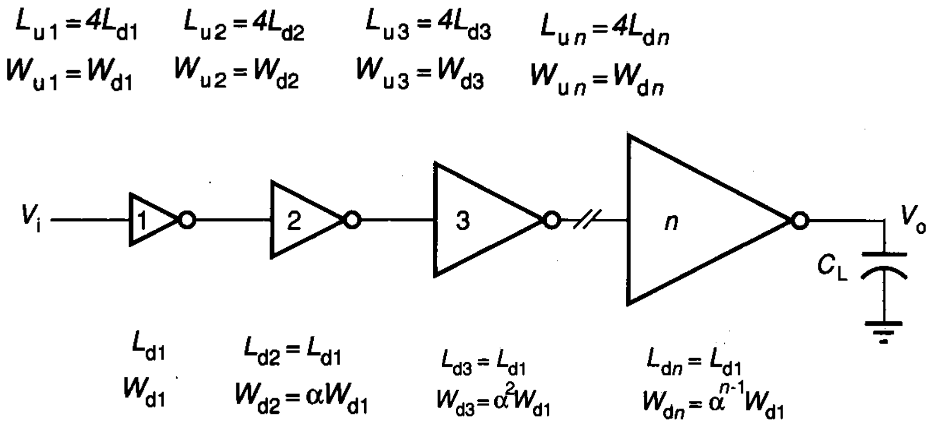
where,

W_{dk} and L_{dk} are device dimensions correspond to the pull down transistor

W_{uk} and L_{uk} are device dimensions correspond to the pull up transistor

The load on the kth stage C_{LK} is,

$$C_{Lk} = \alpha^{k} C_G$$



$$t_{\text{path}} \approx t_{\text{apd}} \sum_{i=1}^{N} \frac{m_i + f_i}{\theta_i}$$

The average propagation delay of the first inverter is αt_{apd}

Hence, it follows from above equation with $m_i = 0$ and $(f_i / \theta_i) = \alpha$

The total delay for the cascade is,

$$t_{\text{cas}} = n \alpha t_{\text{apd}}$$

Let r be the ratio between the propagation delays of the direct drive circuit and of the geometric cascade approach.

$$r = \frac{t_{\text{cas}}}{t_{\text{dir}}} = \frac{n \alpha t_{\text{apd}}}{t_{\text{apd}} C_L / C_G} = \frac{n \alpha C_G}{C_L}$$

It is our goal to determine n and α to minimize r and thus minimize the propagation delay in driving the load.

Therefore, n can be eliminated from the expression for r to obtain the expression.

$$r = \frac{\ln(C_L/C_G)}{C_L/C_G}\frac{\alpha}{\ln\alpha}$$

****

2.10 Circuit Families

Briefly discuss about the classification of circuit families and comparison of the circuit families. (May 2014, APRIL-2015)

Draw the CMOS logic circuit for the Boolean expression Z= $\overline{A(B+C)+DE}$ **and explain. (April 2018)**

Draw and explain the function of static CMOS.

2.10.1: Static CMOS

- Static CMOS circuits with complementary nMOS pulldown and pMOS pullup networks are used for the majority of logic gates in integrated circuits.

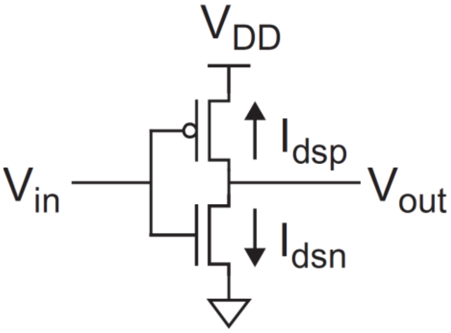


Figure: Static CMOS inverter

Advantages of static CMOS:

- Static CMOS circuits have good noise margins
- Static CMOS circuits are fast, low power, easy to design.
- Static CMOS circuits are widely supported by CAD tools.
- Static CMOS circuits are available in standard cell libraries.

Drawback of static CMOS

- It requires both nMOS and pMOS transistors for each input.
- It has a relatively large logical effort.
- Gate delay is increased.

a. Bubble pushing

- CMOS stages are inherently inverting, so AND and OR functions must be built from NAND and NOR gates.
- DeMorgan's law helps with this conversion:

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

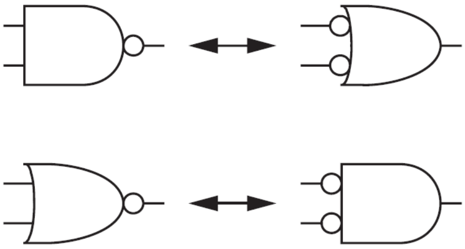


Figure: Bubble pushing with DeMorgan's law

- A NAND gate is equivalent to an OR of inverted inputs.
- A NOR gate is equivalent to an AND of inverted inputs.
- The same relationship applies to gates with more inputs.
- Switching between these representations is easy and is often called bubble pushing.

b. Compound Gates

- Static CMOS also efficiently handles compound gates computing various inverting combinations of AND/OR functions in a single stage.
- The function F = AB +CD can be computed with an AND-OR INVERT- 22 (AOI22) gate and an inverter, as shown in Figure.

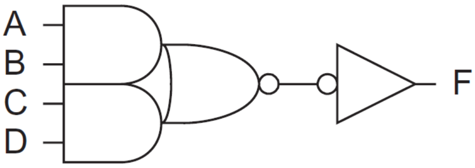


Figure: Logic using AOI22 gate

- Logical effort of compound gates can be different for different inputs.
- Figure shows, how logical efforts can be estimated for the AOI21, AOI22 and a more complex compound AOI gate.

Q: Design a circuit described by the Boolean function Y=[A.(B+C)(D+E)]'using CMOS logic. (NOV 2021)

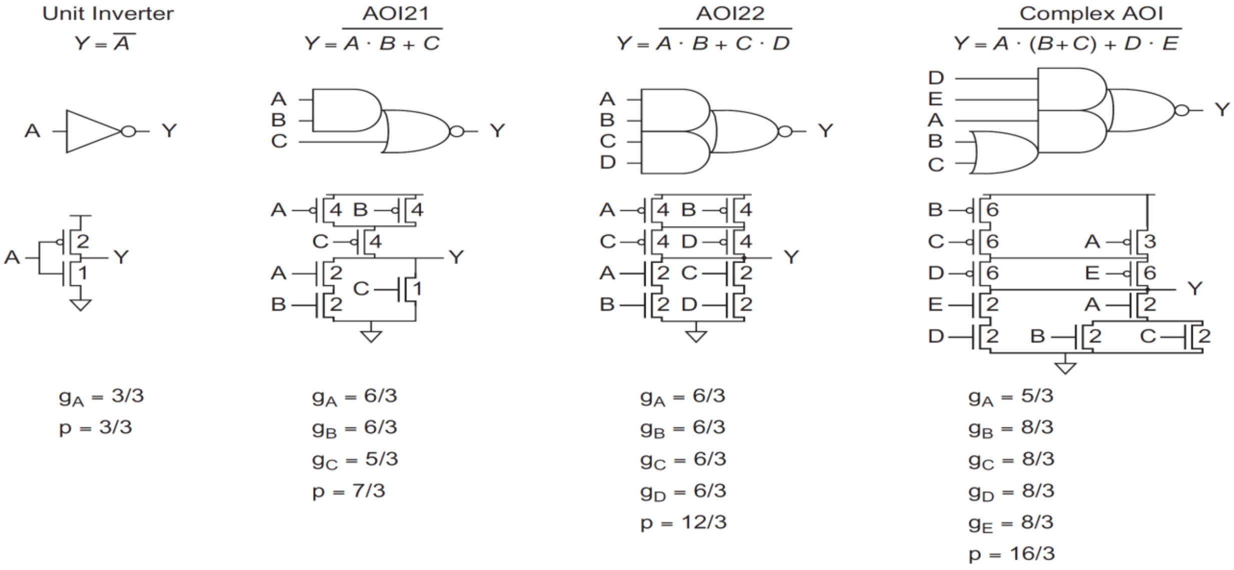


Figure: Logical efforts and parasitic delays of AOI gates

c. Input ordering delay effect

- The logical effort and parasitic delay of different gate inputs are different.
- Consider the falling output transition occurring, when one input hold a stable 1 value and the other rises from 0 to 1.
- If input B rises last, node x will initially be at $V_{DD} - V_t = V_{DD}$, because it was pulled up through the nMOS transistor on input A.
- The Elmore delay is (R/2)(2C) + R(6C) =7RC=2.33 τ
- If input A raises last, node x will initially be at 0 V, because it was discharged through the nMOS transistor on input B.
- No charge must be delivered to node x, so the Elmore delay is simply R(6C) =6RC =2τ.

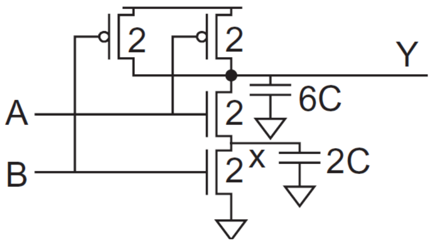


Figure: 2 –input NAND gate Schematic Y=A.B

- We define the outer input to be the input closer to the supply rail (e.g., B) and the inner input to be the input closer to the output (e.g., A).
- Therefore, if one signal is known to arrive later than the others, the gate is faster when that signal is connected to the inner input.

d. Asymmetric gates

- When one input is far less critical than another, even symmetric gates can be made asymmetric to favor the late input at the expense of the early one.
- In a series network, this involves connecting the early input to the outer transistor and making the transistor wider, so that, it offers less series resistance when the critical input arrives.
- In a parallel network, the early input is connected to a narrower transistor to reduce the parasitic capacitance.
- Consider the path in Figure (a). Under ordinary conditions, the path acts as a buffer between A and Y.
- When reset is asserted, the path forces the output low.
- If reset only occurs under exceptional circumstances and take place slowly, the circuit should be optimized for input-to-output delay at the expense of reset.
- This can be done with the asymmetric NAND gate in Figure (b).

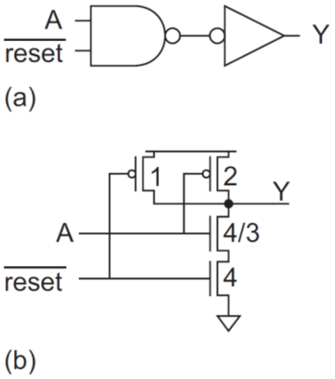


Figure: Resettable buffer optimized for data input

e. Skewed gates

What is meant by skewed gate and give functions of skewed gate with schematic diagrams?

- One input transition is more important than the other. HI-skew gates to favor the rising output transition. LO-skew gates to favor the falling output transition.
- This favoring can be done by decreasing the size of the noncritical transistor.
- The logical efforts for the rising (up) and falling (down) transitions are called g_u and g_d, respectively.
- Figure (a) shows, how a HI-skew inverter is constructed by downsizing the nMOS transistor.
- This maintains the same effective resistance for the critical transition, while reducing the input capacitance relative to the unskewed inverter of Figure (b).
- Thus reducing the logical effort on that critical transition to g_u = 2.5/3 =5/6.
- The logical effort for the falling transition is estimated by comparing the inverter to a smaller unskewed inverter with equal pulldown current, shown in Figure (c), giving a logical effort of g_d =2.5/1.5 =5/3.

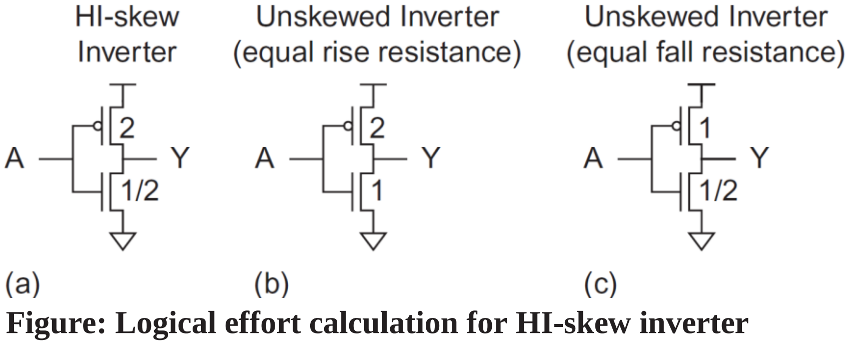


Figure: Logical effort calculation for HI-skew inverter

- Figure shows, HI skew and LO-skew gates with a skew factor of two. Skewed gates are sometimes denoted with an H or an L on their symbol in a schematic.

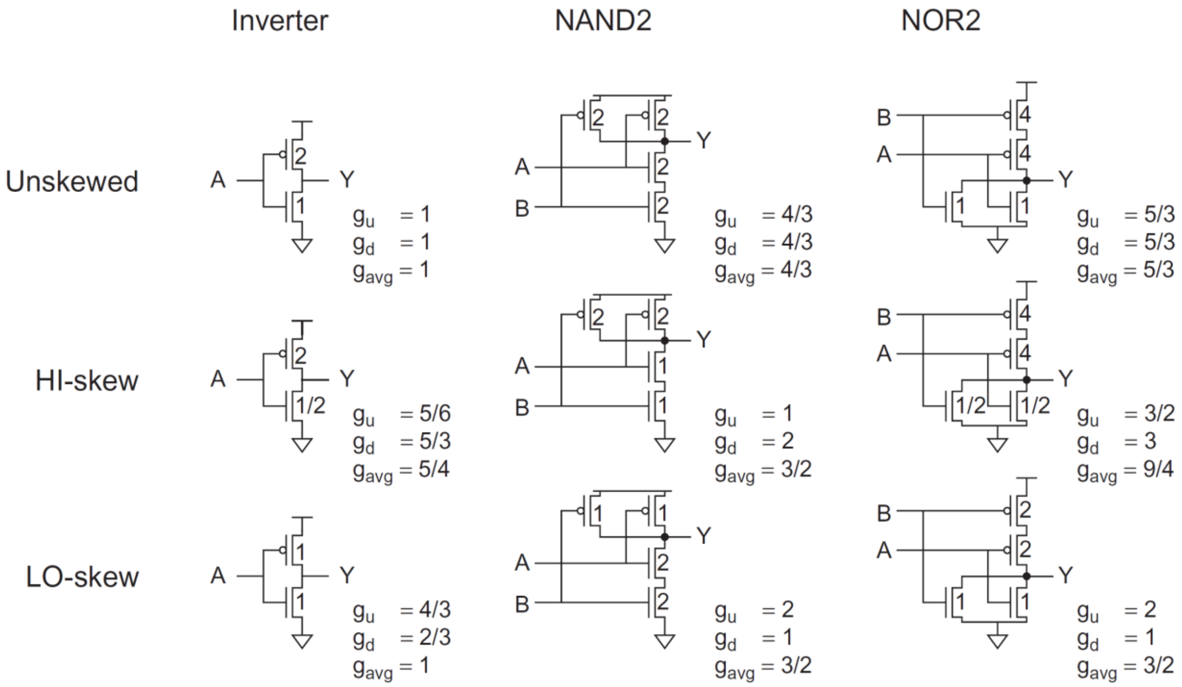


Figure: List of skewed gates

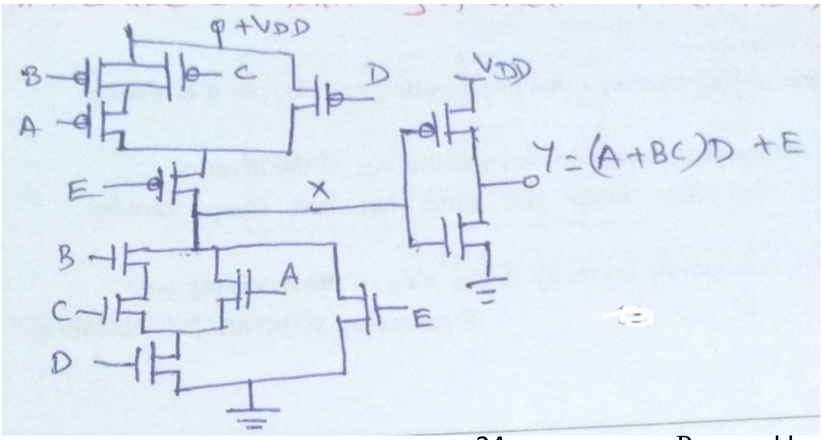
f. P/N ratios

- By accepting a slower rise delay, the pMOS transistors can be downsized to reduce input capacitance and average delay significantly.
- P/N ratio is defined as the ratio of pMOS to nMOS transistor width. For processes, a mobility ratio of µn/µp = 2.

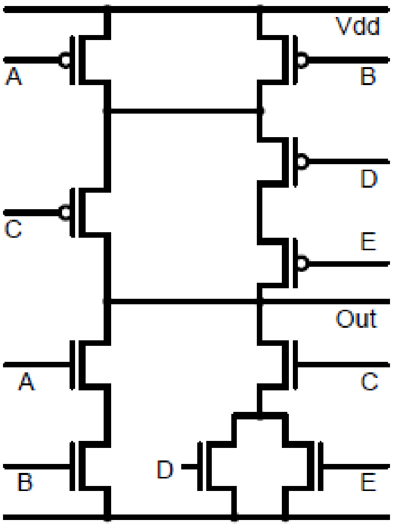
g. Multiple threshold voltages

- Some CMOS processes offer two or more threshold voltages.
- Transistors with lower threshold voltages produce more ON current, but also leak exponentially more OFF current.
- Libraries can provide both high- and low-threshold versions of gates.
- The low-threshold gates can be used carefully to reduce the delay of critical paths.
- Skewed gates can use low-threshold devices on, only the critical network of transistors.

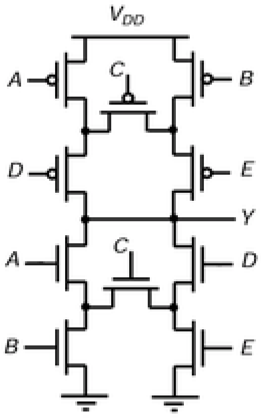
❖ **Realize the following function Y=(A+BC)D+E using static CMOS logic. (April 2019-6M)**



Example: Realize the following function Y= [AB+C (D+E)]' using static CMOS logic. [May 2021 (model)]



Example: Implement the following expression in static CMOS logic fashion using no more than 10 transistors. Y = (AB + ACE + DE + DCB)' [Nov 2019]



****

2.10.2: Ratioed Circuits:

Write short notes on ratioed circuits. (Nov 2016)

- The ratioed gate consists of an nMOS pulldown network and pullup device called the static load.
- When the pulldown network is OFF, the static load pulls the output to 1.
- When the pulldown network turns ON, it fights the static load.
- The static load must be weak enough that, the output pulls down to an acceptable 0. Hence, there is a ratio constraint between the static load and pulldown network.

Advantage: Stronger static loads produce faster rising outputs.

Disadvantages:

- Degrade the noise margin and burn more static power when the output is 0.

  - A resistor is a simple static load, but large resistors consume a large layout area in typical MOS processes.
- Another technique is to use an nMOS transistor with the gate tied to V_{GG} (Shown in fig.(b)). If $V_{GG} = V_{DD}$, the nMOS transistor will only pull up to $V_{DD} – V_t$.
- Figure (c) shows depletion load ratioed circuit.

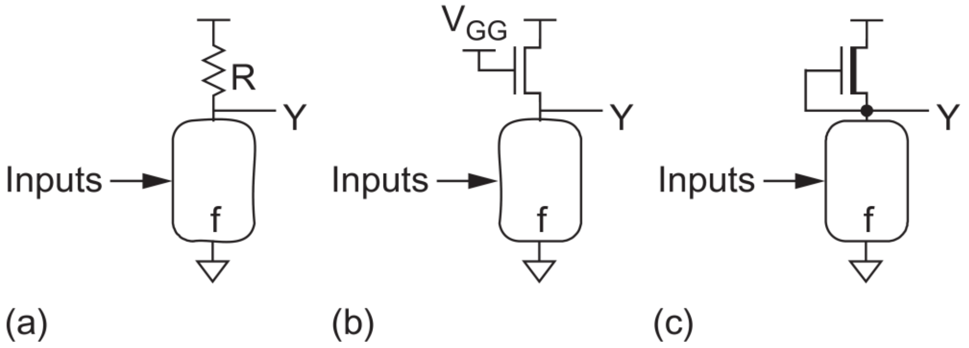


Figure: nMOS ratioed gates

2.10.3: pseudo nMOS

Explain the detail about pseudo-nMOS gates with neat circuit diagram. (April/May 2011) (Nov/Dec 2013)

- Figure (a) shows a pseudo-nMOS inverter.
- The static load is built from a single pMOS transistor that has its gate grounded, so it is always ON.
- The beta ratio affects the shape of the transfer characteristics and the V_{OL} of the inverter.
- Larger relative pMOS transistor size offer faster rise time, but less sharp transfer characteristics.
- **Drawback:** Pseudo-nMOS gates will not operate correctly if $V_{OL} > V_{IL}$ of the receiving gate.

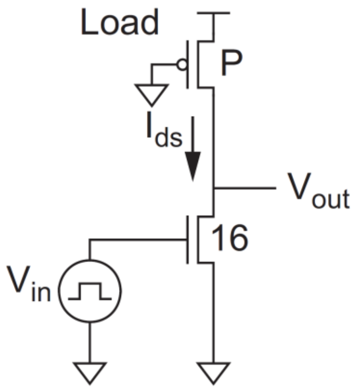


Figure (a): pseudo nMOS inverter

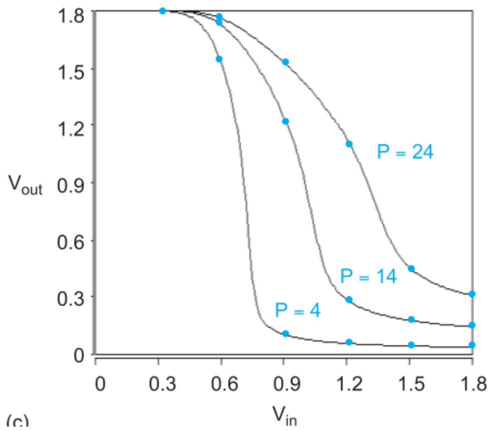


Figure: D.C Charateristics

- Figure shows several pseudo-nMOS logic gates.

Implement NAND gate using pseudo- nMOS logic. (Nov 2013, May 2021[modell])

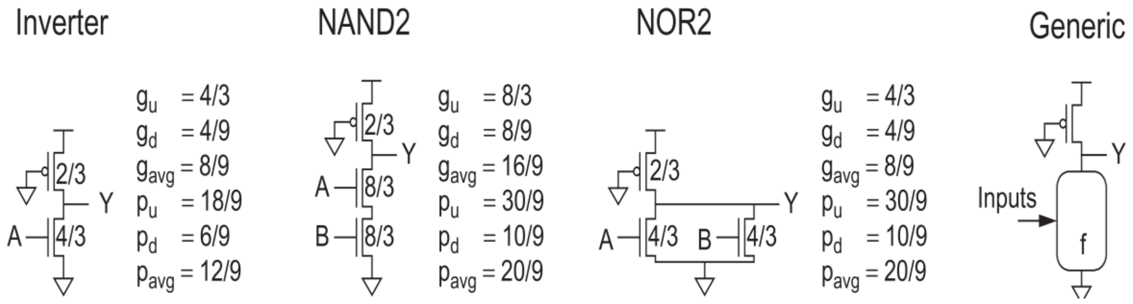


Figure: Pseudo-nMOS logic gates

2.10.4: Ganged capacitor:

- Figure shows pairs of CMOS inverters ganged together.
- The truth table is given in Table, showing that the pair compute the NOR function. Such a circuit is sometimes called a symmetric ^{2} NOR, or ganged CMOS.

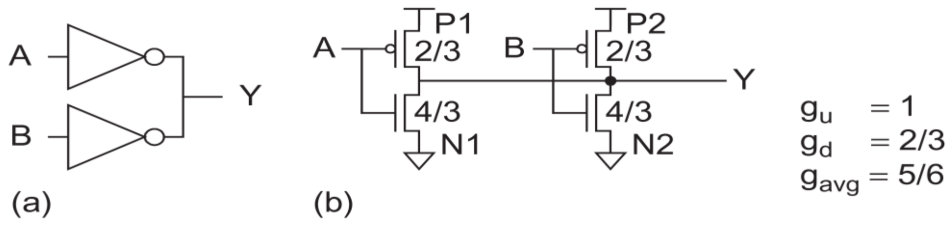


Figure: symmetric ^{2} NOR gate.

A	*B*	*N1*	*P1*	*N2*	*P2*	*Y*
0	0	OFF	ON	OFF	ON	1
0	1	OFF	ON	ON	OFF	~ 0
1	0	ON	OFF	OFF	ON	~ 0
1	1	ON	OFF	ON	OFF	0

Table: Operation of symmetric NOR

- When one input is 0 and the other 1, the gate can be viewed as a pseudo-nMOS circuit with appropriate ratio constraints.
- When both inputs are 0, both pMOS transistors turn on in parallel, pulling the output high faster than they would, in an ordinary pseudo nMOS gate.
- When both inputs are 1, both pMOS transistors turn OFF, saving static power dissipation.

2.10.5: Differential Cascode voltage switch with pass gate logic (DCVSPG)

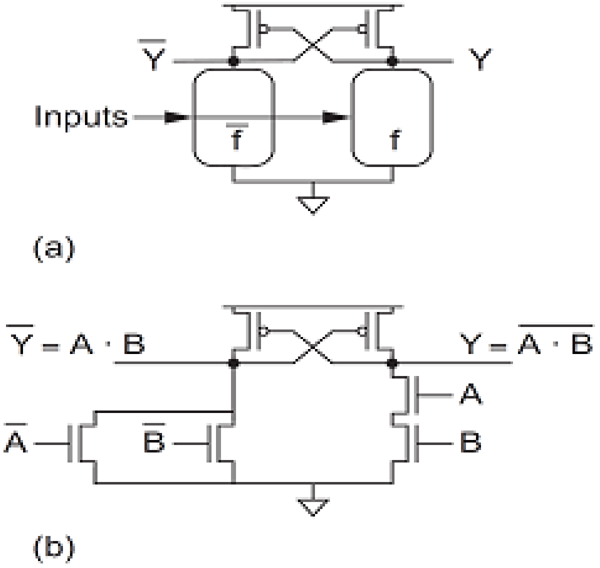
Explain about DCVSL logic with suitable example. (May 2017)

- Cascode Voltage Switch Logic (CVSL) seeks the benefits of ratioed circuits without the static power consumption.
- It uses both true and complementary input signals and computes both true and complementary outputs using a pair of nMOS pulldown networks, as shown in Figure (a).

- The pulldown network f implements the logic function as in a static CMOS gate, while $\overline{f}$ uses inverted inputs feeding transistors arranged in the conduction complement.
- For any given input pattern, one of the pulldown networks will be ON and the other OFF.
- The pulldown network that is ON will pull that output low.
- This low output turns ON the pMOS transistor to pull the opposite output high.
- When the opposite output rises, the other pMOS transistor turns OFF, so no static power dissipation occurs.
- Figure (b) shows a CVSL AND/NAND gate.

Advantage:

- CVSL has a potential speed advantage because all of the logic is performed with nMOS transistors, thus reducing the input capacitance.



(a)

(b)

Figure: CVSL gates

****

2.11:Dynamic CMOS design:

Describe the basic principle of operation of dynamic CMOS, domino and NP domino logic with neat diagrams. (NOV 2011) [April / May 2023]

Dynamic Circuits:

- Ratioed circuits reduce the input capacitance by replacing the pMOS transistors connected to the inputs with a single resistive pullup.
- The drawbacks of ratioed circuits include
    - Slow rising transitions,
    - Contention on the falling transitions,
    - Static power dissipation and a nonzero V_{OL}.
- Dynamic circuits avoid these drawbacks by using a clocked pullup transistor rather than a pMOS that is always ON.
- Figure compares (a) static CMOS, (b) pseudo-nMOS, and (c) dynamic inverters.

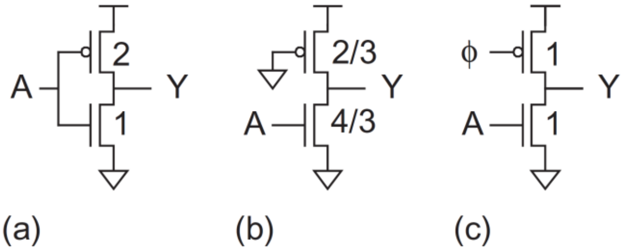


Figure: Comparison of (a) static CMOS, (b) pseudo-nMOS, and (c) dynamic inverters

- Dynamic circuit operation is divided into two modes, as shown in Figure.

  (i) During precharge, the clock ϕ is 0, so the clocked pMOS is ON and initializes the output Y high.

  (ii) During evaluation, the clock is 1 and the clocked pMOS turns OFF. The output may remain high or may be discharged low through the pulldown network.

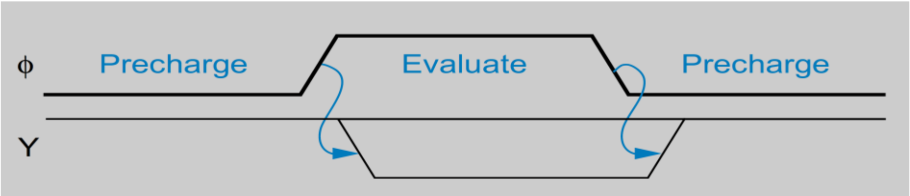


Figure: Precharge and evaluation of dynamic gates

<u>Advantages:</u>

- Dynamic circuits are the fastest used circuit family because they have lower input capacitance and no contention during switching.
- Zero static power dissipation.

<u>Disadvantags:</u>

- They require careful clocking, consume significant dynamic power and are sensitive to noise during evaluation mode.

Foot transistor:

- In Figure (c), if the input A is 1 during precharge, contention will take place because both the pMOS and nMOS transistors will be ON.
- When the input cannot be guaranteed to be 0 during precharge, an extra clocked evaluation transistor can be added to the bottom of the nMOS stack.
- To avoid contention as shown in the below figure, extra transistor is sometimes called as foot is added.

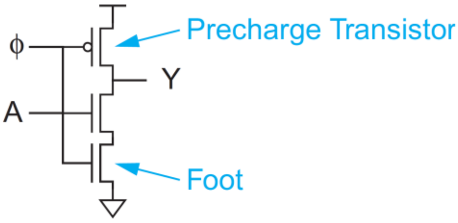


Figure: Footed dynamic inverter

- The given below figure shows generic footed and unfooted gates.

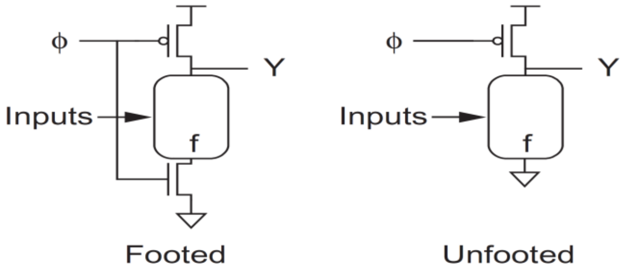


Figure: Generalized footed and unfooted dynamic gates

- The given below figure estimates the falling logical effort of both footed and unfooted dynamic gates.

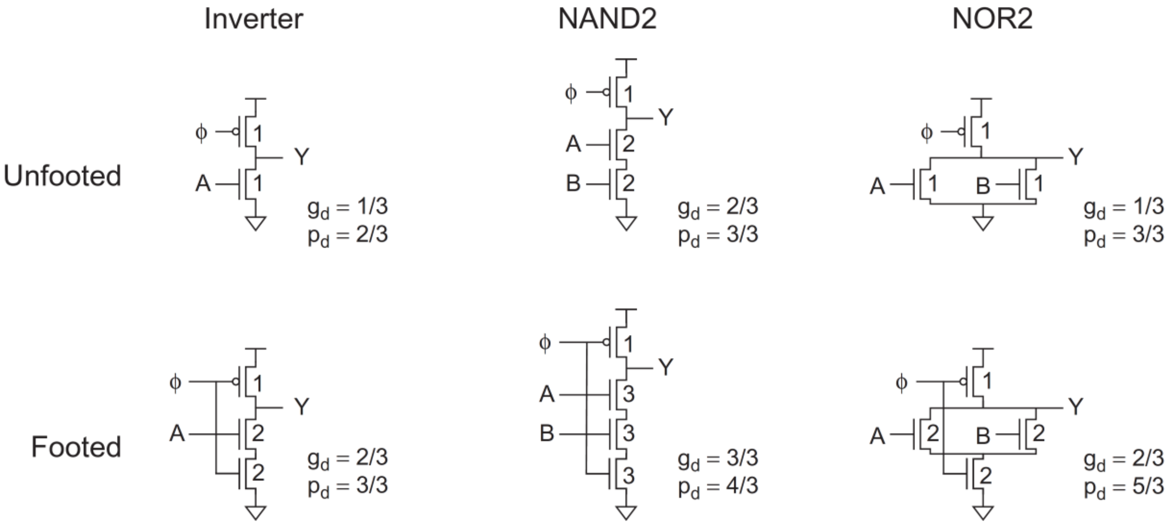


Figure: List of dynamic gates

- The pull down transistor’s width is chosen to give unit resistance. Precharge occurs while the gate is idle and takes place more slowly.
- Therefore, the precharge transistor width is chosen for twice unit resistance.
- This reduces the capacitive load on the clock and the parasitic capacitance at the expense of greater rising delays.
- Footed gates have higher logical effort than their unfooted concept but are still an improvement over static logic.
- The parasitic delay does increase with the number of inputs, because there is more diffusion capacitance on the output node.
- A fundamental difficulty with dynamic circuits is the monotonicity requirement. While a dynamic gate is in evaluation, the inputs must be monotonically rising.
- That is, the input can start LOW and remain LOW, start LOW and rise HIGH, start HIGH and remain HIGH, but not start HIGH and fall LOW.
- Figure shows waveforms for a footed dynamic inverter in which the input violates monotonicity.

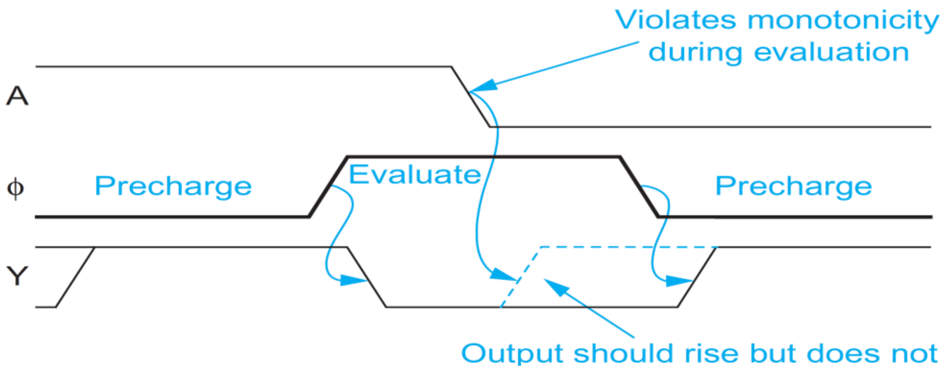

Violates monotonicity during evaluation
A
ϕ
Precharge
Evaluate
Precharge
Y
Output should rise but does not

Figure: Monotonicity problem

- During precharge, the output is pulled HIGH.
- When the clock rises, the input is HIGH, so the output is discharged LOW through the pulldown network.
- The input later falls LOW, turning off the pulldown network. However, the precharge transistor is also OFF, so the output floats, staying LOW rather than rising.
- The output will remain low until the next precharge step.
- The inputs must be monotonically rising for the dynamic gate to compute the correct function.
- Unfortunately, the output of a dynamic gate begins HIGH and monotonically falls LOW during evaluation.
- This monotonically falling output X is not a suitable input to a second dynamic gate expecting monotonically rising signals, as shown in the below figure.
- Dynamic gates sharing the same clock cannot be directly connected.
- This problem is often overcome with domino logic.

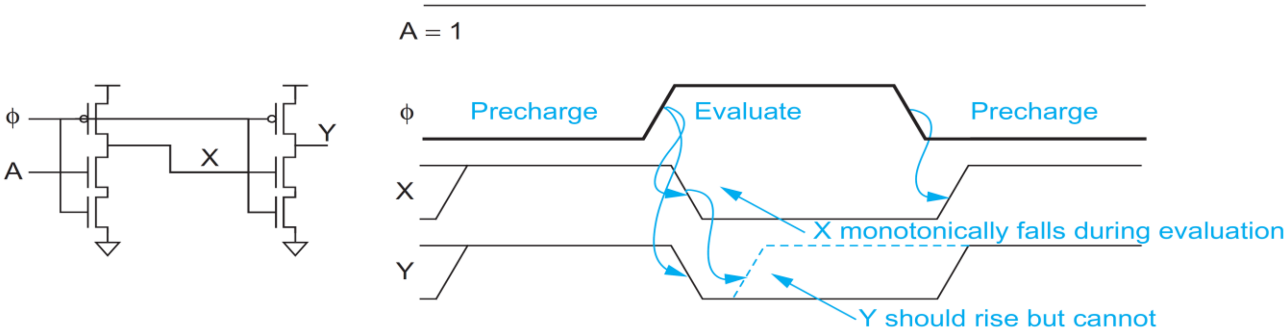


Figure: Incorrect connection of dynamic gates

The **charge sharing problem** occurs when the charge which is stored at the output node in the pre-charge phase is shared among the junction capacitance of transistors in the evaluation phase. Charge sharing may degrade the output voltage level or even cause an erroneous output value.

To **overcome the dynamic charge sharing** and soft- node leakage problems in NORA CMOS structures, a circuit technique called Zipper CMOS can be used. The basic circuit architecture of Zipper CMOS is essentially identical to NORA CMOS, with the exception of the clock signals.

****

2.12: Pass Transistor Logic:

Explain Pass transistor logic with neat sketches. (April 2008)
Explain the pass transistor logic and show how complementary pass transistor logic and double pass transistor logic are applied for 2: 1 multiplexer. [May 2021][Apr/May 2022]

- In pass-transistor circuits, inputs are applied to the source/drain diffusion terminals.
- These circuits build switches using either nMOS pass transistors or parallel pairs of nMOS and pMOS transistors called as transmission gates.
- The nMOS transistors pass ‘0’s well but 1’s poorly. Figure (a) shows an nMOS transistor with the gate and drain tied to V_{DD}.
- Initially at $Vs = 0$. $V_{gs} > V_{tn}$, so the transistor is ON and current flows.
- Therefore, nMOS transistors attempting to pass a 1 never pull the source above $V_{DD} – V_{tn}$. This loss is called a threshold drop.
- The pMOS transistors pass 1’s well but 0’s poorly.
- If the pMOS source drops below $|V_{tp}|$, the transistor cuts off.
- Hence, pMOS transistors only pull down to a threshold above GND, as shown in Figure (b).

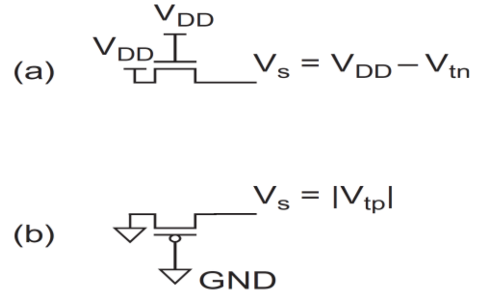
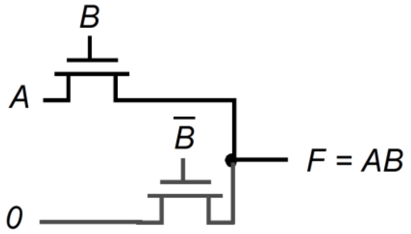
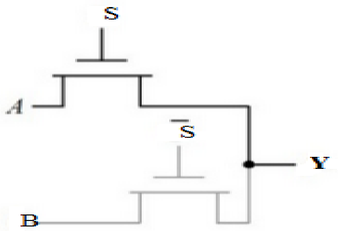


Figure : Pass Transistor threshold drops

- Figures show an implementation of the AND function and 2x1 multiplexer using only NMOS transistors.



AND Logic



2x1 multiplexer

- In AND gate, if the *B* input is high, the top transistor is turned ON and copies the input *A* to the output *F*.
- When *B* is low, the bottom pass transistor is turned ON and passes a 0.
- In 2x1 multiplexer, if the S selection input is high, the top transistor is turned ON and allows input *A* to the output Y.
- When *S* is low, the bottom pass transistor is turned ON and passes the B input.
- An NMOS device is effective at passing a 0 but is poor at pulling a node to V_{DD}. When the pass transistor pulls a node high, the output only charges up to V_{DD} -V_{tn}.

Application:

- Pass transistors are essential to the design of efficient 6-transistor static RAM cells used in modern systems.

Formal Method for P-T Logic Derivation

Complementary function can be implemented from the same circuit structure by applying complementary principle:

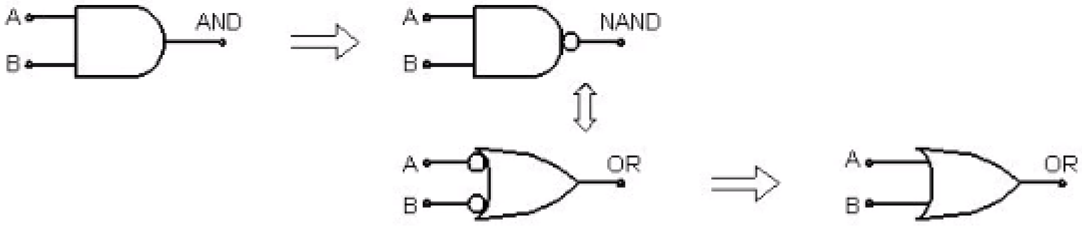
Complementary Principle: Using the same circuit topology, with pass signals inverted, complementary logic function is constructed in CPL.

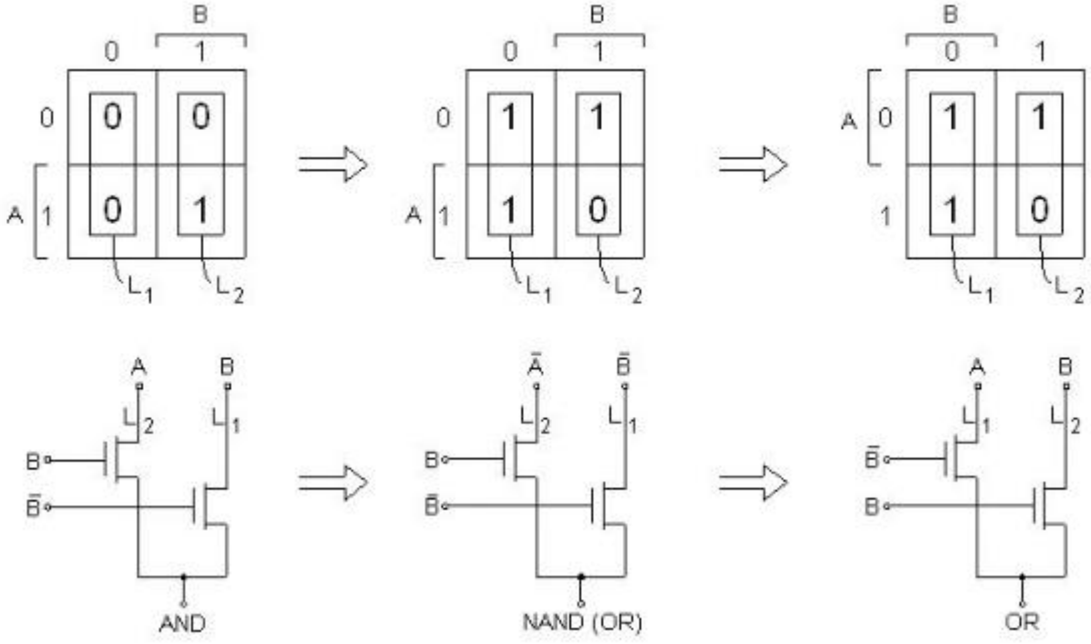
By applying duality principle, a dual function is synthesized:

Duality Principle: Using the same circuit topology, with gate signals inverted, dual logic function is constructed.

Following pairs of basic functions are dual:

- AND-OR (and vice-versa)
- NAND-NOR (and vice-versa)
- XOR and XNOR are self-dual (dual to itself)

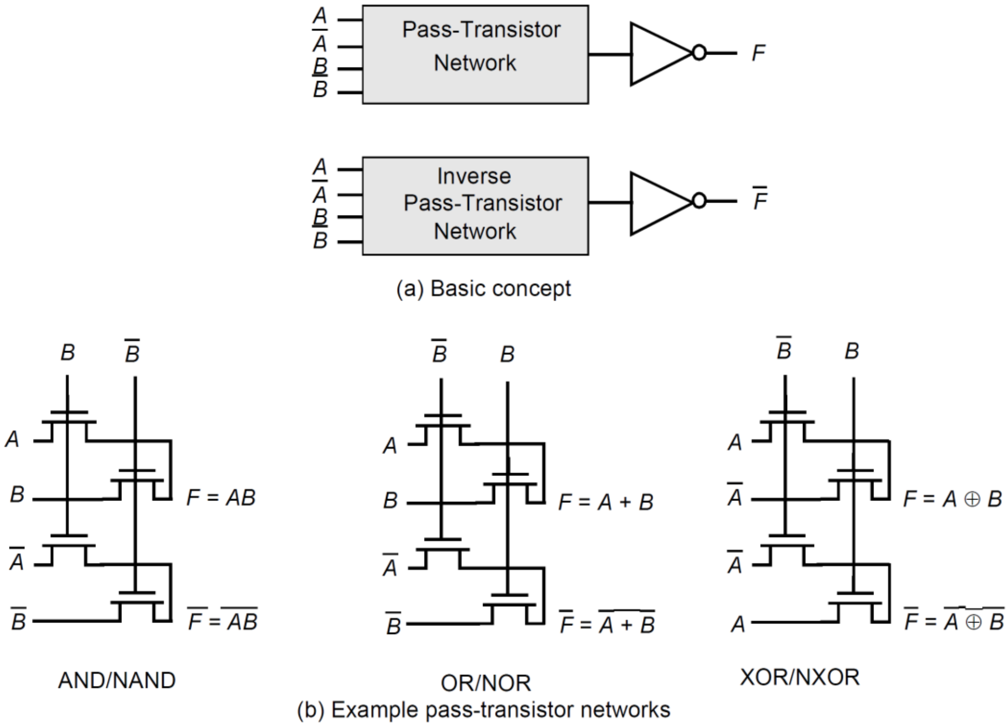




Complementary: AND ⇒ NAND
Duality: AND ⇒ OR

2.12.1: Differential Pass Transistor Logic / Complementary Pass Transistor Logic (CPL)

- For high performance design, a differential pass-transistor logic family, called CPL, is commonly used.
- The basic idea is to accept true and complementary inputs and produce true and complementary outputs.
- A number of CPL gates (AND/NAND, OR/NOR, and XOR/NXOR) are shown in Figure.
- Since the circuits are *differential*, complementary data inputs and outputs are always available.
- Both polarities of every signal eliminate the need for extra inverters, as is often the case in static CMOS or pseudo-NMOS.
- CPL belongs to the class of *static* gates, because the output-defining nodes are always connected to either V_{DD} or *GND* through a low resistance path.
- This is advantage for the noise flexibility.


A
$\overline{A}$
B
$\overline{B}$
Pass-Transistor Network
F
Inverse Pass-Transistor Network
$\overline{F}$
$F = AB$
$\overline{F} = \overline{AB}$
AND/NAND
$F = A + B$
$\overline{F} = \overline{A + B}$
OR/NOR
$F = A \oplus B$
$\overline{F} = \overline{A \oplus B}$
XOR/NXOR

(a) Basic concept

(b) Example pass-transistor networks

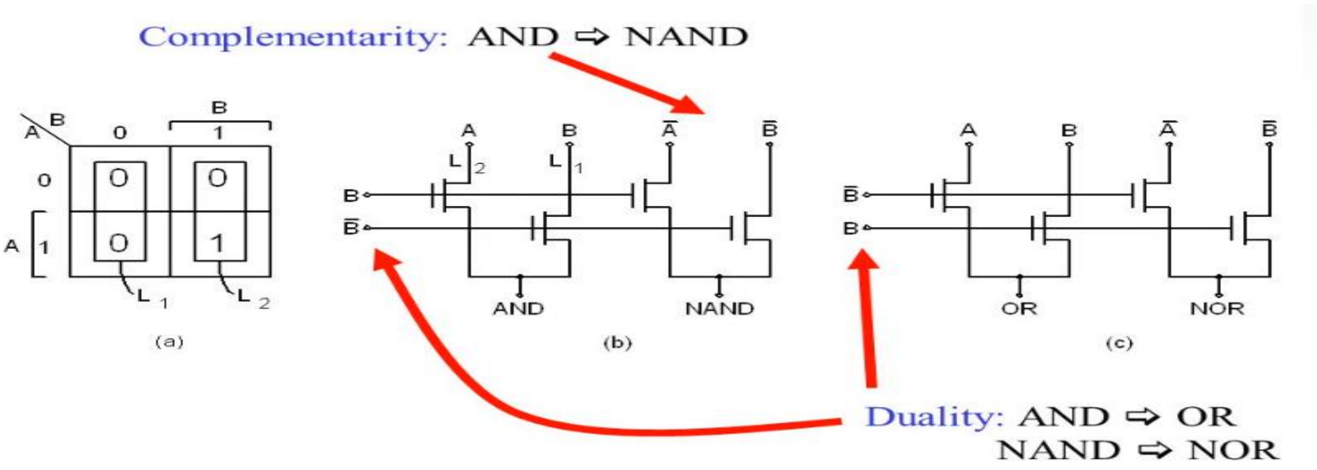
Figure: Complementary pass-transistor logic (CPL).

Formal Method for CPL Logic Derivation (AND, NAND, OR, NOR)

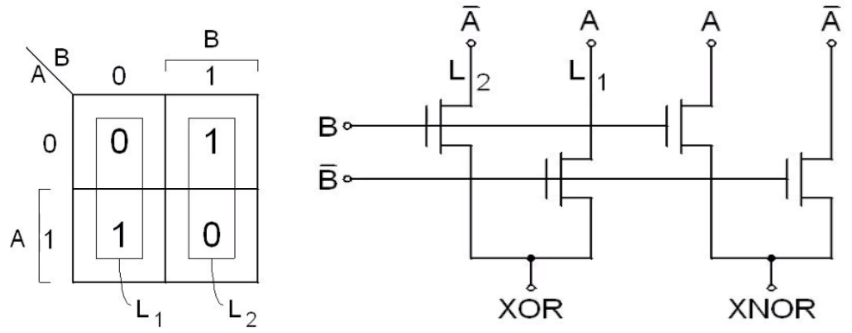
(a) Cover the Karnaugh-map with largest possible cubes (overlapping allowed)

(b) Express the value of the function in each cube in terms of input signals

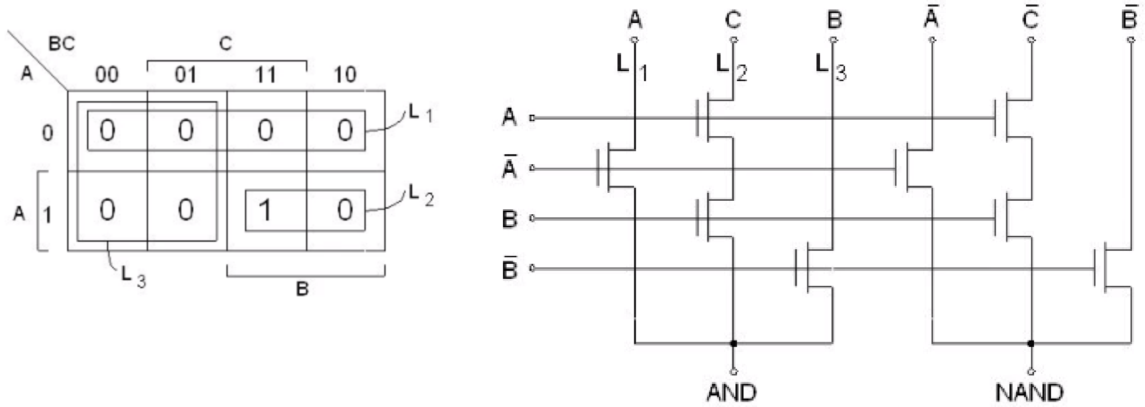
(c) Assign one branch of transistor(s) to each of the cubes and connect all the branches to one common node, which is the output of NMOS pass-transistor network



Example: Realize XOR and XNOR gate using CPL. [Nov 2019]



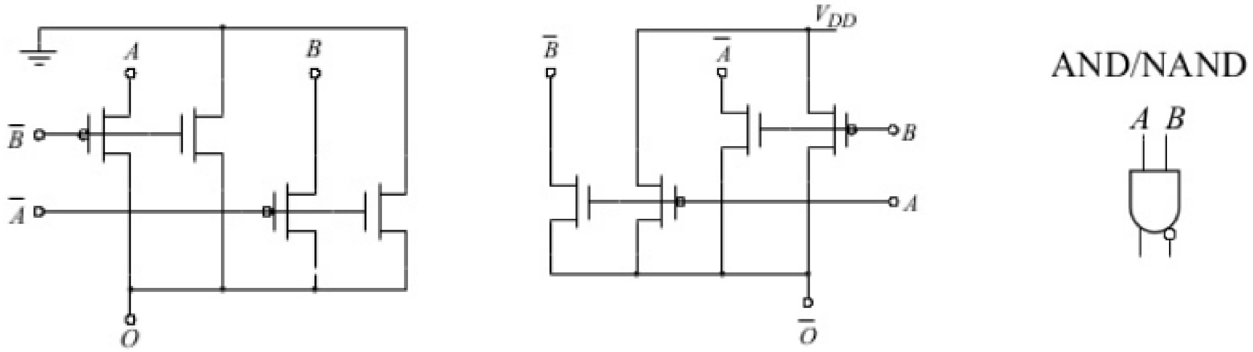
Example: Realize 3 – input AND and NAND gate using CPL.



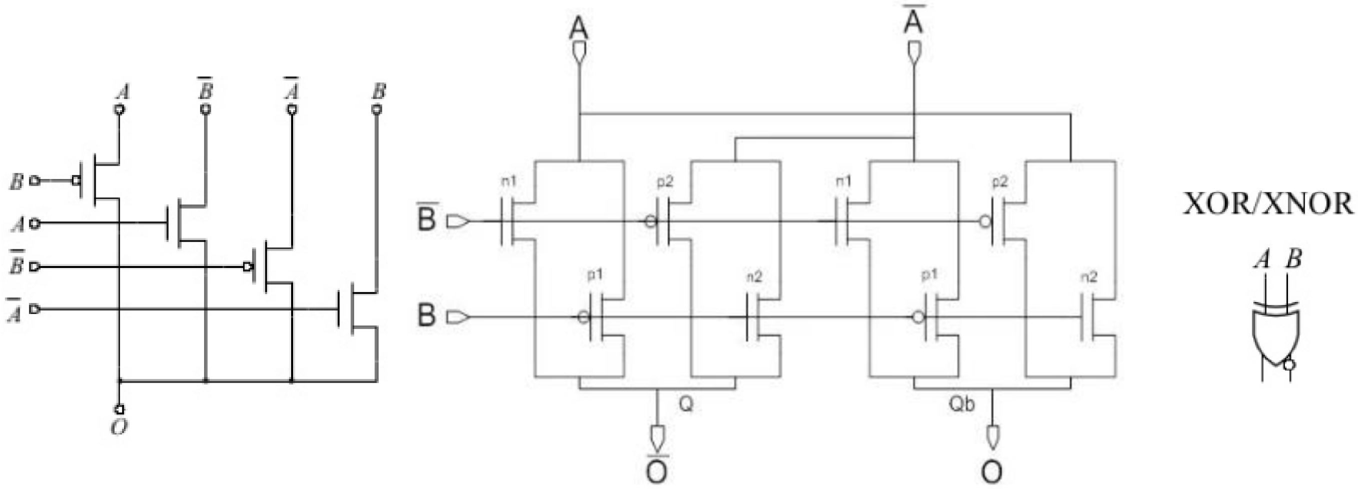
2.12.2 Double Pass Transistor Logic (DPL)

- Double Pass Transistor Logic is a double rail form of CMOS transmission gate optimized to use single pass transistors where only a known 0 or 1 needs to be passed.
- It passes good high and low logic levels without the need for level restoring devices.
- DPL uses both nMOS and pMOS switches to realize the desired functions.

- This provides a full swing on the output.
- No extra transistors are required for swing restoration.
- A DPL gate consists of both true and complementary inputs / outputs and hence is a *dual rail logic circuit.*



Example: Realize XOR gate using Double Pass Transistor Logic (DPL).



Synthesis Rules

- Two NMOS branches cannot be overlapped covering logic 1s. Similarly, two PMOS branches cannot be overlapped covering logic 0s.
- Pass signals are expressed in terms of input signals or supply. Every input vector has to be covered with exactly two branches.
- At any time, excluding transitions, exactly two transistor branches are active (any of the pairs NMOS/PMOS, NMOS/NMOS and PMOS/PMOS are possible), i.e. they both provide output current.

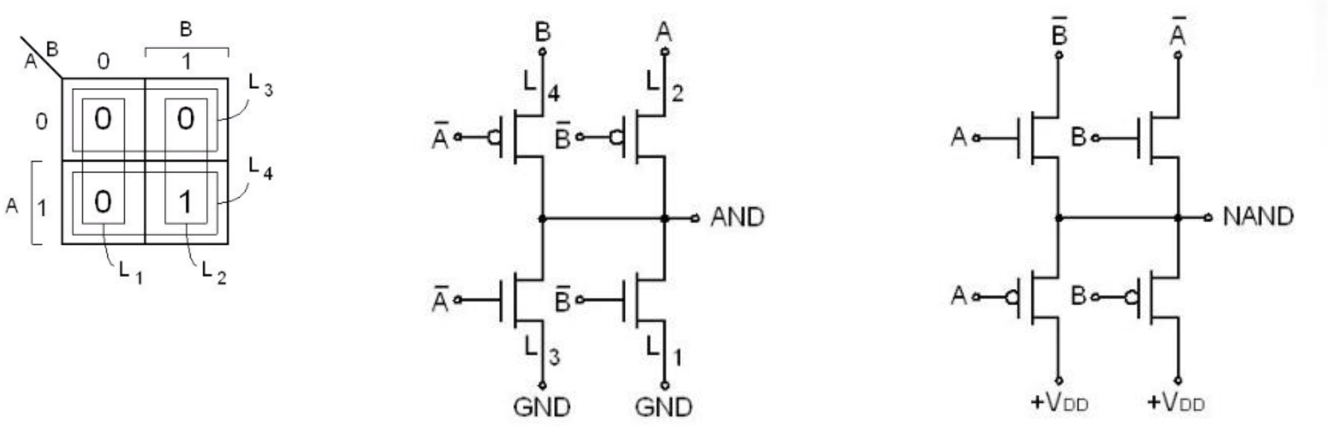
Complementary Principle: Complementary logic function in DPL is generated after the following modifications:

- Exchange PMOS and NMOS devices. Invert all pass and gate signals

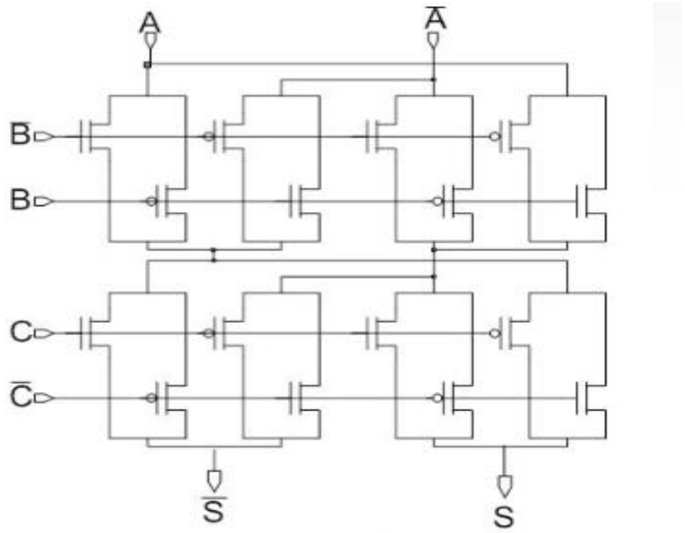
Duality Principle: Dual logic function in DPL is generated when:

- PMOS and NMOS devices are exchanged, and VDD and GND signals are exchanged.

Example: Realize AND and NAND gate using DPL.



Example: Realize full adder (sum circuit) using Double Pass Transistor Logic (DPL).

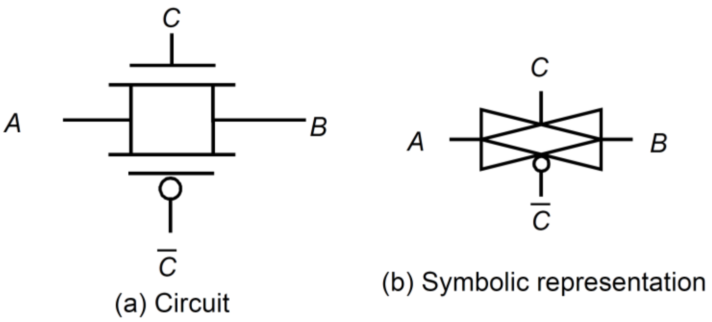


2.13: CMOS with transmission gates

- **Discuss in detail the characteristics of CMOS Transmission gates.(May 2016, May 2017, Nov 2017)**
- **Explain Transmission gates with neat sketches. (April 2008, April 2018)**
- **List out limitations of pass transistor logic. Explain any two techniques used to overcome limitations. (NOV 2018)**

- A transmission gate in conjunction with simple static CMOS logic is called CMOS with transmission gate.
- A transmission gate is parallel pairs of nMOS and pMOS transistor.
- A single nMOS or pMOS pass transistor suffers from a threshold drop.
- Transmission gates solve the threshold drop but require two transistors in parallel.
- The resistance of a unit-sized transmission gate can be estimated as R for the purpose of delay estimation.

- Current flow the parallel combination of the nMOS and pMOS transistors. One of the transistors is passing the value well and the other is passing it poorly.
- A logic-1 is passed well through the pMOS but poorly through the nMOS.
- Estimate the effective resistance of a unit transistor passing a value in its poor direction as twice the usual value: 2R for nMOS and 4R for pMOS.



(a) Circuit (b) Symbolic representation

Figure: CMOS Transmission gate

- The given below figure shows the parallel combination of resistances. When passing a 0, the resistance is R || 4R = (4/5)R.
- The effective resistance passing a 1 is 2R || 2R = R.
- Hence, a transmission gate made from unit transistors is approximately R in either direction.
- Transmission gates are built using equal-sized nMOS and pMOS transistors.
- Boosting the size of the pMOS transistor only slightly improves the effective resistance while significantly increasing the capacitance.

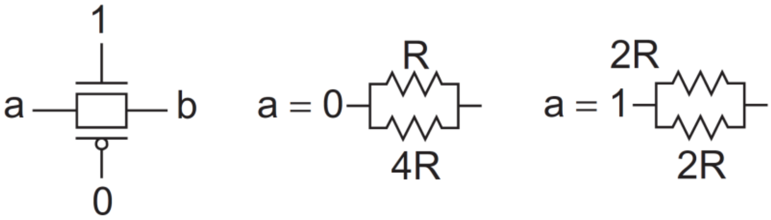
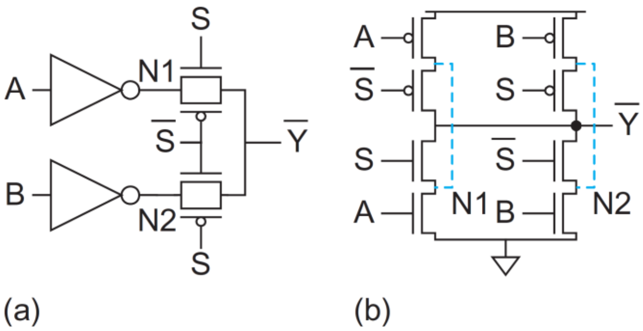


Figure: Effective resistance of a unit transmission gate

- Figure (a) redraws the multiplexer to include the Inverters that drive the diffusion inputs but to exclude the output inverter. Figure (b) shows this multiplexer drawn at the transistor level.



(a) (b)

Figure: CMOSTG in a 2-input inverting multiplexer

****

2.14: Domino logic

Explain the domino logic families with neat diagrams. (NOV 2012, APRIL-2015, Nov 2017)

- The dynamic-static pair together is called a domino gate.
- The monotonicity problem can be solved by placing a static CMOS inverter between dynamic gates, as shown in figure (a).
- This converts the monotonically falling output into a monotonically rising signal suitable for the next gate, as shown in figure (b).
- A single clock can be used to precharge and evaluate all the logic gates within the chain.
- The dynamic output is monotonically falling during evaluation, so the static inverter output is monotonically rising.
- Therefore, the static inverter is usually a HI-skew gate to favor this rising output. Observe that precharge occurs in parallel, but evaluation occurs sequentially.

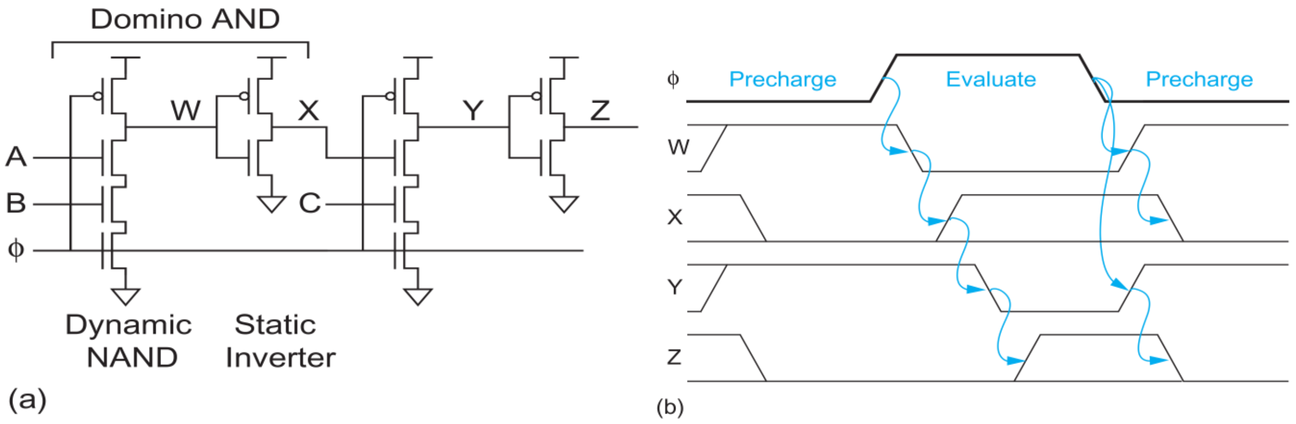


Figure: Domino gates

2.15: Dual Rail Domino Logic:

- Dual-rail domino gates encode each signal with a pair of wires. The input and output signal pairs are denoted with _h and _l, respectively.
- Table summarizes the encoding. The _h wire is asserted to indicate that the output of the gate is "high" or 1. The _l wire is asserted to indicate that the output of the gate is "low" or 0.
- When the gate is precharged, neither _h nor _l is asserted. The pair of lines should never be both asserted simultaneously during correct operation.

sig_h	*sig_l*	Meaning
0	0	Precharged
0	1	'0'
1	0	'1'
1	1	Invalid

Table: Dual-rail domino signal encoding

- Dual-rail domino gates accept both true and complementary inputs and compute both true and complementary outputs, as shown in Figure (a).
- This is identical to static CVSL circuits except that the cross-coupled pMOS transistors are instead connected to the precharge clock.
- Therefore, dual-rail domino can be viewed as a dynamic form of CVSL, sometimes called DCVS.
- Figure (b) shows a dual-rail AND/NAND gate and Figure (c) shows a dual-rail XOR/XNOR gate. The gates are shown with clocked evaluation transistors, but can also be unfooted.

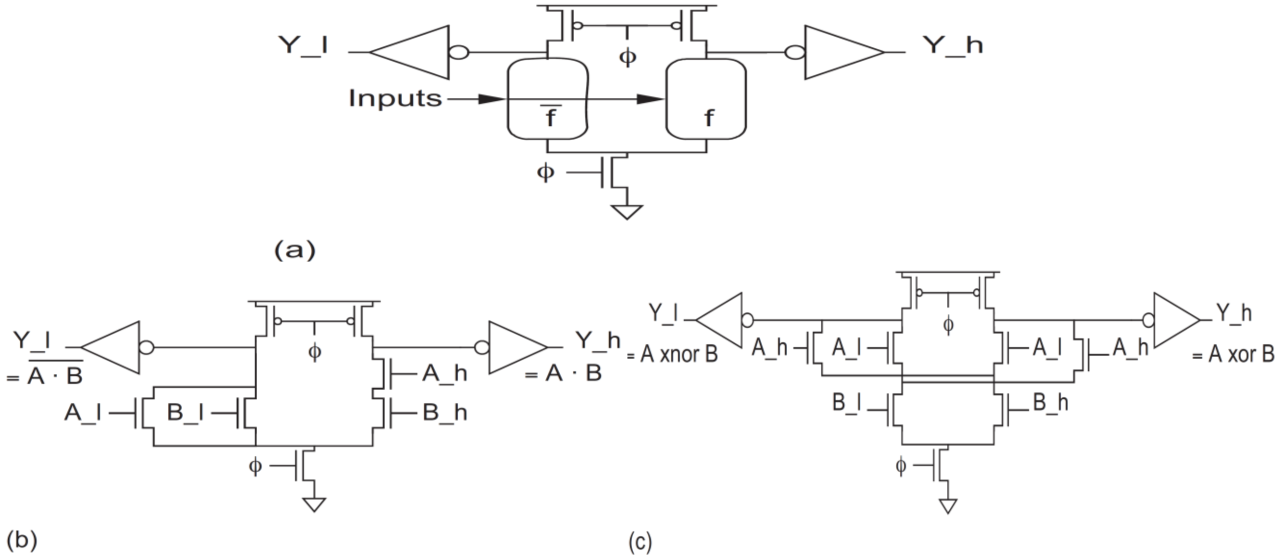


Figure: Dual-rail domino gates

Disadvantages:

- It requires more area, wiring and power.
- Dual-rail structures lose the efficiency of wide dynamic NOR gates.

Application:

- It is useful for asynchronous circuits.

2.16: Keepers

Explain the keeper logic family with neat diagrams.
Briefly discuss the signal integrity issues in dynamic design. (April 2018, NOV 2018)

- Dynamic circuits have poor input noise margins.
- If the input rises above $V_{t,}$, while the gate is in evaluation, the input transistors will turn ON weakly and can incorrectly discharge the output.
- Both leakage and noise margin problems can be addressed by adding a keeper circuit.
- Figure shows a conventional keeper on a domino buffer. The keeper is a weak transistor that holds, or staticizes, the output at the correct level when it would otherwise float.
- When the dynamic node X is high, the output Y is low and the keeper is ON to prevent X from floating.
- When X falls, the keeper initially opposes the transition, so it must be much weaker than the pulldown network.

- Eventually Y rises, turning the keeper OFF and avoiding static power dissipation.

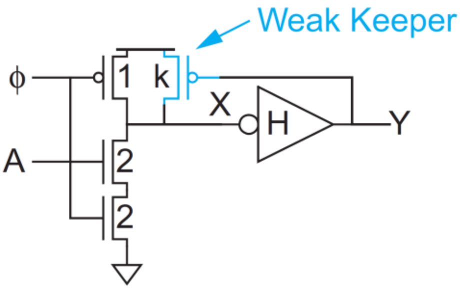


Figure: Conventional keeper

- The keeper must be strong enough to compensate for any leakage current drawn when the output is floating and the pulldown stack is OFF.
- Strong keepers also improve the noise margin, because when the inputs are slightly above V_t, the keeper can supply enough current to hold the output high.

2.16.1: Differential keeper:

- Figure shows a differential keeper for a dual-rail domino buffer.
- When the gate is precharged, both keeper transistors are OFF and the dynamic outputs float. As one of the rails evaluates low, the opposite keeper turns ON.
- The differential keeper is fast, because it does not oppose the falling rail.
- As long as one of the rails is guaranteed to fall promptly, the keeper on the other rail will turn on before excessive leakage or noise causes failure.

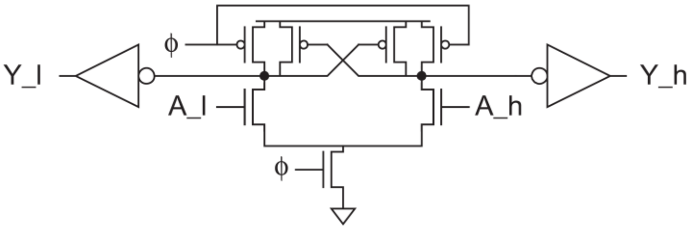


Figure: Differential keeper

2.16.2: Secondary precharge devices

- Dynamic gates are subject to problems with charge sharing.
- For example, consider the 2-input dynamic NAND gate in Figure (a). Suppose the output Y is precharged to V_{DD} and inputs A and B are low.

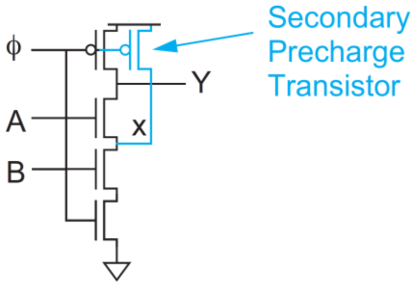


Figure: Secondary precharge transistor

- Also suppose that the intermediate node x had a low value from a previous cycle.
- During evaluation, input A rises, but input B remains low, so the output Y should remain high.
- However, charge is shared between C_X and C_Y, shown in Figure (b). This behaves as a capacitive voltage divider and the voltages equalize at

$$V_x = V_Y = \frac{C_Y}{C_x + C_Y} V_{DD}$$

2.16.3: Charge sharing:

- Charge sharing is serious when the output is lightly loaded (small C_Y) and the internal capacitance is large.
- If the charge-sharing noise is small, the keeper will eventually restore the dynamic output to V_{DD}.
- If the charge-sharing noise is large, the output may flip and turn off the keeper, leading to incorrect results.
- Charge sharing can be overcome by precharging some or all of the internal nodes with secondary precharge transistors.
- These transistors should be small, because they only charge the small internal capacitances and their diffusion capacitance slows the evaluation.
- It is sufficient to precharge every other node in a tall stack.

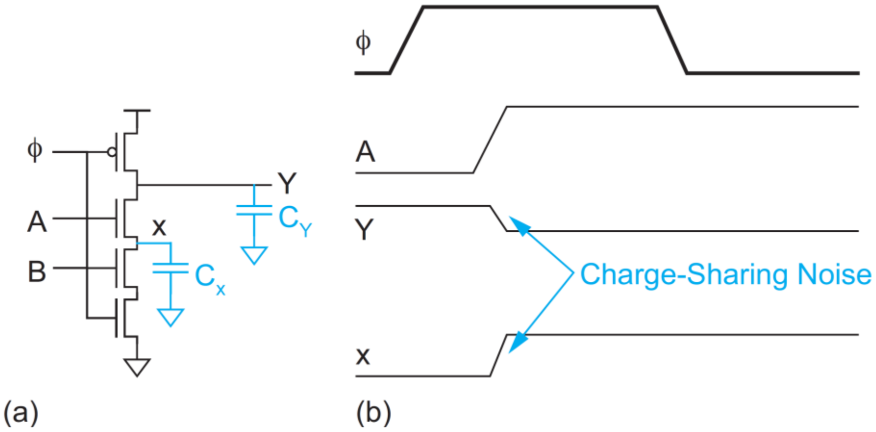


Figure: Charge-sharing noise

2.16.4: NP and Zipper Domino

Describe the basic principle of operation of NP domino logic. (NOV 2011)

- The HI-skew inverting static gates are replaced with predischarged dynamic gates using pMOS logic.
- A footed dynamic p-logic NAND gate is shown in Figure (b). When ϕ is 0, the first and third stages precharge high while the second stage predischarges low.
- When ϕ rises, all the stages evaluate. Domino connections are possible, as shown in Figure (c).
- The design style is called NP Domino or NORA Domino (NO RAce).

- NORA has two major drawbacks.
  (i) The logical effort of footed p-logic gates is worse than that of HI-skew gates.

(ii) NORA is extremely susceptible to noise.

- In an ordinary dynamic gate, the input has a low noise margin (about V_t), but is strongly driven by a static CMOS gate.
- The floating dynamic output is more prone to noise from coupling and charge sharing, but drives another static CMOS gate with a larger noise margin.
- In NORA, however, the sensitive dynamic inputs are driven by noise prone dynamic outputs.
- Besides drawback and the extra clock phase requirement, there is little reason to use NORA.
- Zipper domino is a closely related technique, that leaves the precharge transistors slightly ON during evaluation by using precharge clocks. This swing between 0 and V_{DD} – $|V_{tp}|$ for the pMOS precharge and V_{tn} and V_{DD} for the nMOS precharge.

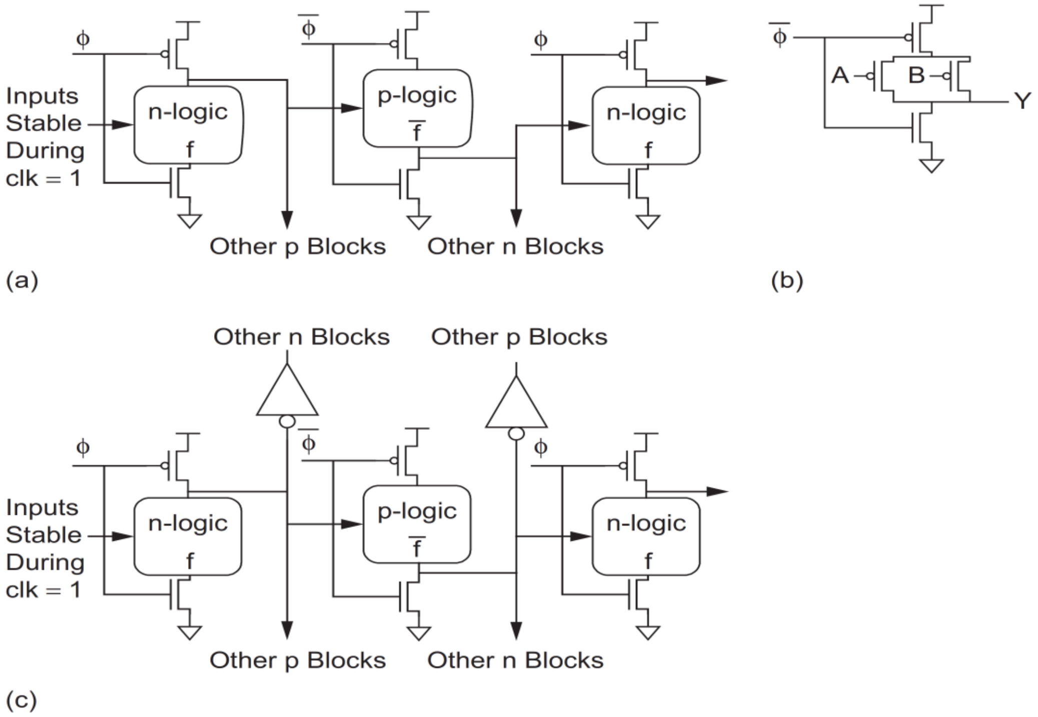


Figure : NP Domino

****

2.17: Power dissipation:

- ❖ **Explain the static and dynamic power dissipation in CMOS circuits with necessary diagrams and expressions. (DEC 2011, Nov 2015, NOV 2016, May 2017, May 2010)**
- ❖ **What are the sources of power dissipation in CMOS and discuss various design techniques to reduce power dissipation in CMOS? (Nov 2012, May 2013, Nov 2014, May 2016)**
- ❖ **Derive an expression for dynamic power dissipation. (April 2019, Nov 2019, May 2021)[April / May 2023]**

- The instantaneous power P (t) consumed by a circuit element is the product of the current and the voltage of the element

$$P(t) = I(t)V(t)$$

- The energy consumed over time interval T is the integral of the instantaneous power

$$E = \int_0^T P(t)\,\mathrm{dt}$$

- The average power is $P_{avg} = \frac{E}{T} = \frac{1}{T}\int_0^T P(t)\,\mathrm{dt}$

  Power is expressed in units of Watts (W). Energy is usually expressed in Joules (J)
  - By Ohm’s Law, V = IR, so the instantaneous power dissipated in the resistor is

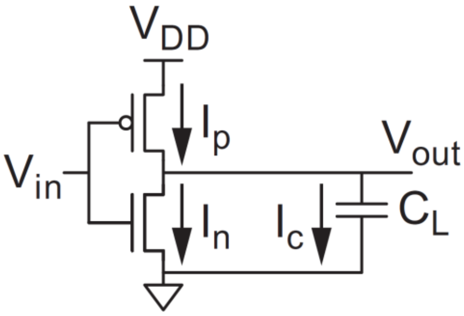
$$\boldsymbol{P_R(t) = \frac{V_R^2(t)}{R} = I_R^2(t)\,\mathrm{R}}$$

- This power is converted from electricity to heat. V_{DD} supplies power proportional to its current

  P_{VDD} (t) =I_{DD} (t) V_{DD}
- When the capacitor is charged from 0 to V_C, it stores energy E_C

$$E_C = \int_0^\infty I(t)V(t)\,dt = \int_0^\infty C\frac{dV}{dt}V(t)\,dt = C\int_0^{V_C} V(t)\,dV = \tfrac{1}{2}CV_C^2$$

- Figure shows a CMOS inverter driving a load capacitance.



- When the input switches from 1 to 0, the pMOS transistor turns ON and charges the load to V_{DD}.
- According to E_C equation the energy stored in the capacitor is

$$E_C = \tfrac{1}{2}C_L V_{DD}^2$$

- The energy delivered from the power supply is

$$E_C = \int_0^\infty I(t)V_{DD}\,dt = \int_0^\infty C\frac{dV}{dt}V_{DD}\,dt = CV_{DD}\int_0^{V_{DD}} dV = CV_{DD}^2$$

- Gate switches at some average frequency f_{sw}.
- Over some interval T, the load will be charged and discharged Tf_{sw} times.
- Then, the average power dissipation is

$$P_{\text{switching}} = \frac{E}{T} = \frac{Tf_{\text{sw}}CV_{DD}^2}{T} = CV_{DD}^2 f_{\text{sw}}$$

- This is called the dynamic power because it arises from the switching of the load.
- Because most gates do not switch every clock cycle, it is often more convenient to express switching frequency f_{sw} as an activity factor α times the clock frequency f.
- The dynamic power dissipation may be rewritten as

$$P_{\text{switching}} = \alpha C V_{DD}^2 f$$

- The activity factor is the probability that the circuit node transitions from 0 to 1, because that is the only time the circuit consumes power.
- A clock has an activity factor of α = 1 because it rises and falls every cycle.
- The total power of a circuit is calculated as,

$$P_{\text{dynamic}} = P_{\text{switching}} + P_{\text{short circuit}}$$

$$P_{\text{static}} = \left(I_{\text{sub}} + I_{\text{gate}} + I_{\text{junct}} + I_{\text{contention}}\right) V_{DD}$$

$$P_{\text{total}} = P_{\text{dynamic}} + P_{\text{static}}$$

2.17.1: Dynamic power:

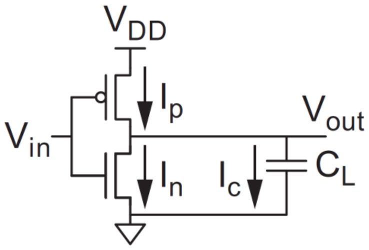
- Dynamic power consists mostly of the switching power.

$$P_{\text{switching}} = \alpha C V_{DD}^2 f$$

- The supply voltage V_{DD} and frequency f are known by the designer.
- To estimate dynamic power, one can consider each node of the circuit.
- The capacitance of the node is the sum of the gate, diffusion, and wire capacitances on the node.
- The activity factor can be estimated using switching probability or measured from logic simulations.
- The effective capacitance of the node is, its true capacitance multiplied by the activity factor.
- The switching power depends on the sum of the effective capacitances of all the nodes.

2.17.1.1:Sources of dynamic power dissipation:

- Dynamic dissipation due to
  - Charging and discharging load capacitances as gates switchs.
  - “Short-circuit” current while both pMOS and nMOS stacks are partially ON



2.17.1.2: Low Power Design Principles / Reducing dynamic power dissipation:

- ❖ **Explain various ways to minimize the static and dynamic power dissipation. (Nov 2013, May 2015)**
- ❖ **Discuss the low power design principles in detail. (Nov 2017)**

- Low power design involves considering and reducing each of the terms in switching power.

  i. As V_{DD} is a quadratic term, it is good to select the minimum V_{DD}.

  ii. Choose the lowest frequency.

  iii. The activity factor is reduced by putting unused blocks to sleep.

  iv. Finally, the circuit may be optimized to reduce the overall load capacitance.

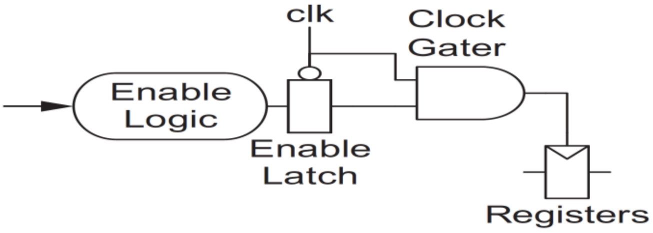
- Switching power is consumed by delivering energy to charge a load capacitance, then dumping this energy to GND.

Activity factor:

- If a circuit can be turned OFF entirely, the activity factor and dynamic power go to zero.
- Blocks are typically turned OFF, by stopping the clock called as clock gating.
- The activity factor of a logic gate can be estimated by calculating the switching probability.

(a)Clock gating:

- Clock gating, AND’s a clock signal with an enable to turn OFF the clock to idle blocks.
- The clock enable must be stable, while the clock is active.
- Figure shows how an enable latch can be used to ensure the enable does not change before the clock falls.



Capacitance:

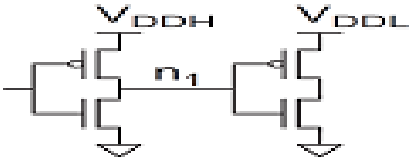
- Switching capacitance comes from the wires and transistors in a circuit.
- Wire capacitance is minimized through good floor planning and placement.
- Device-switching capacitance is reduced by choosing smaller transistors.

Voltage:

- Voltage has a quadratic effect on dynamic power.
- Therefore, choosing a lower power supply significantly reduces power consumption.
- The chip may be divided into multiple voltage domains, where each domain is optimized for the needs of certain circuits.

a. **Voltage domains:**

- Selecting, which circuits belong in which domain and routing power supplies to multiple domains.
- Figure **(Voltage domain crossing)** shows direct connection of inverters in two domains using high and low supplies, V_{DDH} and V_{DDL}, respectively.

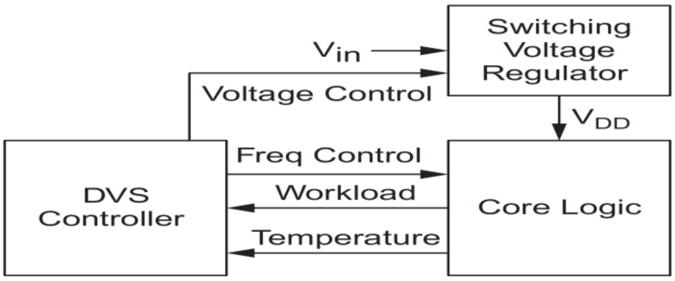


b. **Dynamic voltage scaling (DVS):**

Q: Describe how dynamic voltage scaling can reduce dynamic power dissipation. (NOV 2021)

- Systems can save large amounts of energy by reducing the clock frequency, then reducing the supply voltage.
- This is called dynamic voltage scaling (DVS) or dynamic voltage/frequency scaling (DVFS).

- It determines the supply voltage and clock frequency sufficient to complete the workload on schedule or to maximize performance without overheating.



- Figure shows a block diagram for a basic DVS system.

Frequency:

- Dynamic power is directly proportional to frequency, so a chip should not run faster than necessary.
- Reducing the frequency allows downsizing transistors or using a lower supply voltage.

Low Power Architecture

- Device Level
  - ✓ Low Capacitance in device and Multi Threshold Devices
- DVFS – Dynamic Voltage Frequency Scaling
- Multi VDD
- Gate Sizing
- Voltage Islands
- Power Gating
- Clock Gating
- Parallelism and Pipelined micro-architecture

Parallel Computations

• Multiple cores
• Multiple Issue pipelines
• Linear power increase

Pipelining

• Faster clock
• Exponential power increase
• Longer branch miss-predictions

2.17.2: Static power:

- Static power is consumed even when a chip is not switching.
- Static CMOS gates have no contention current.

2.17.2.1:Sources of static power dissipation:

- Static dissipation due to
  - Subthreshold leakage through OFF transistors.
  - Gate leakage through gate dielectric.
  - Junction leakage from source/drain diffusions.
  - Contention current in ratioed circuits.

$$P_{static} = (I_{sub} + I_{gate} + I_{junc} + I_{contention})V_{DD}$$

1. **Subthreshold leakage current:**
   - Subthreshold leakage current flows when a transistor is OFF.
   - Subthreshold leakage current equation is

$$I_{\text{sub}} = I_{\text{off}} 10^{\frac{V_{gs}+\eta\left(V_{ds}-V_{DD}\right)-k_{\gamma}V_{sb}}{S}}$$

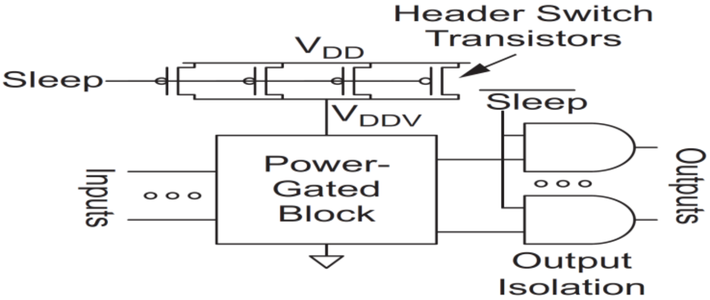
where I_{off} is the subthreshold current at $V_{gs} = 0$ and $V_{ds} = V_{DD}$, and S is the subthreshold slope.

2. **Gate leakage:**
   - Gate leakage occurs when carriers tunnel through a thin gate dielectric, when a voltage is applied across the gate (e.g., when the gate is ON).
   - Gate leakage is a strong function of the dielectric thickness.
3. **Junction leakage:**
   - Junction leakage occurs when a source or drain diffusion region is at a different potential from the substrate.
   - Leakage of reverse-biased diodes is usually negligible.
4. **Contention current:**
   - Static CMOS circuits have no contention current. However, certain alternative circuits inherently draw current even while quiescent.

<u>2.17.2.2:Methods of reducing static power:</u>

Power gating:

- To reduce static current during sleep mode is, to turn OFF the power supply to the sleeping blocks. This technique is called power gating.



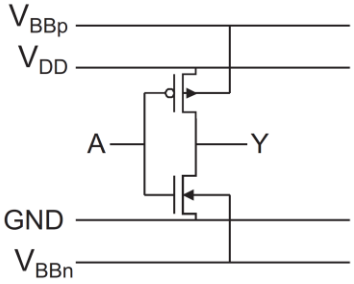
- The logic block receives its power from a virtual V_{DD} rail, V_{DDV}.
- When the block is active, the header switch transistors are ON, connecting V_{DDV} to V_{DD}.
- When the block goes to sleep, the header switch turns OFF, allowing V_{DDV} to float and gradually sink toward 0.

Multiple threshold voltage and oxide thickness:

- Selective application of multiple threshold voltages can maintain performance on critical paths with low-V_t transistors, while reducing leakage on other paths with high-V_t transistors.

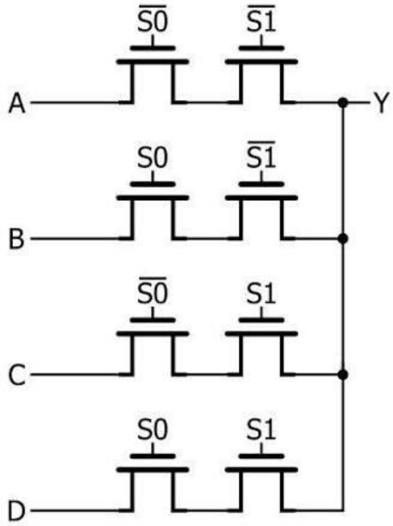
Variable threshold voltage:

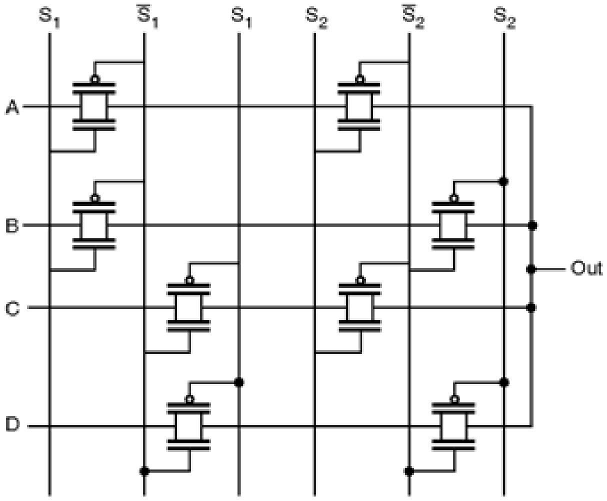
- Method to achieve high I_{on} in active mode and low I_{off} in sleep mode is, by adjusting the threshold voltage of the transistor by applying a body bias.
- This technique is sometimes called variable threshold CMOS (VTCMOS).
- Figure shows a schematic of an inverter using body bias.



****

- ❖ **Let A, B, C and D be the inputs of a data selector and S_0 & S_1 be the select lines. Realize a 4:1 data selector using nMOS pass transistor and transmission gate approach. Compare the hardware complexity. (April 2019-13M)**



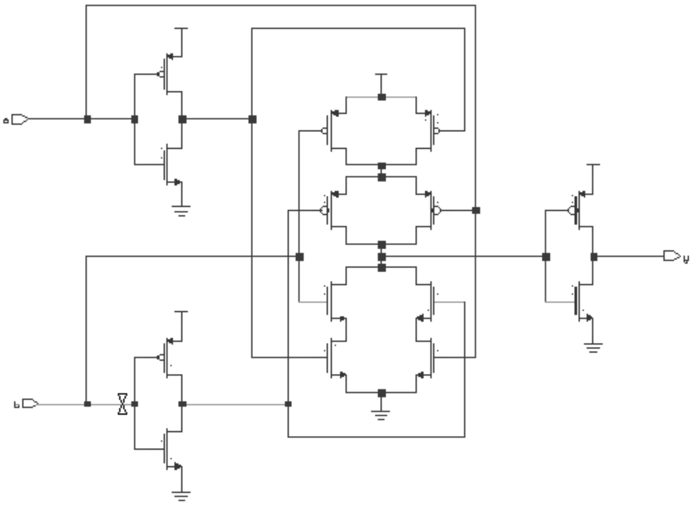


4 : 1 MUX using transmission gate

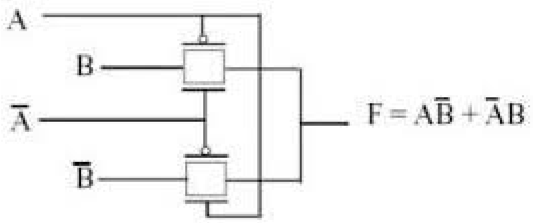
4:1 MUX using pass transistor

Need double of transistors to design 4:1 MUX using transmission gate compare with pass transistor.

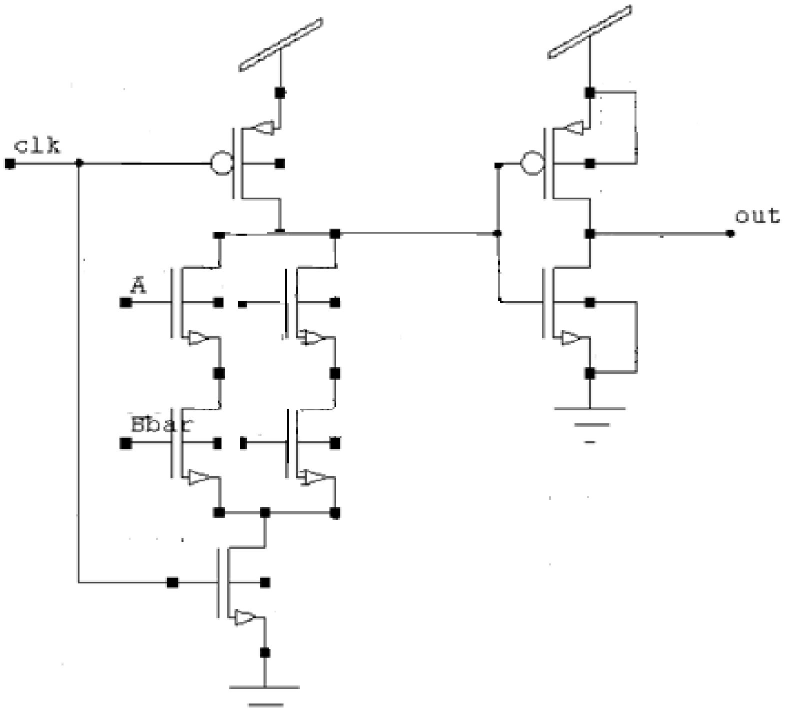
- ❖ **Realize a 2-input XOR using static CMOS, transmission gate and dynamic CMOS logic. Analyze the hardware complexity. (April 2019-15M)**
  Draw a static CMOS XOR gate. [Nov 2019]



2-input XOR using static CMOS



2-input XOR using transmission gate



2-input XOR using dynamic CMOS logic

****

2.18 Circuit Pitfalls

- Circuit designers use simple circuits due to their robustness.
- Larger circuits with more transistors add more area & capacitance and has more possibility of error.

- Static CMOS is the most robust family.

Certain circuit pitfalls that can cause chips to fail include

- ✓ Threshold drop
- ✓ Leakage
- ✓ Charge sharing
- ✓ Power supply noise
- ✓ Hot spots
- ✓ Minority carrier injection
- ✓ Back gate coupling
- ✓ Process sensitivity
- ✓ Soft errors

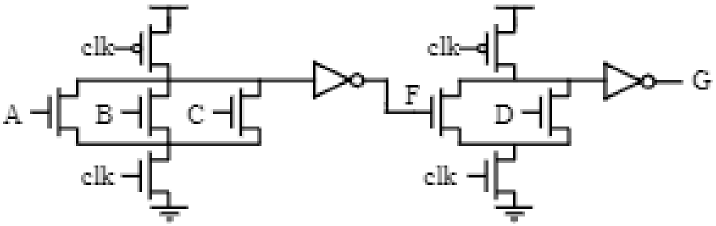
****

Example:

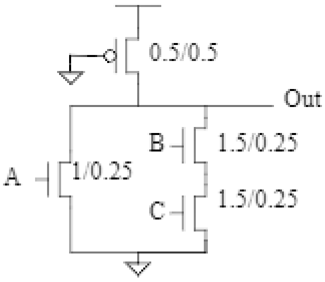
Suppose we wish to implement the two logic functions given by F=A+B+C andG=A+B+C+D. Assume both true and complementary signals are available. Implement these functions in dynamic CMOS as cascaded stages so as to minimize the total transistor count. [Nov 2019]

Solution:

Dynamic gates with NMOS pull-down networks cannot be directly cascaded. This solution uses a domino logic approach.



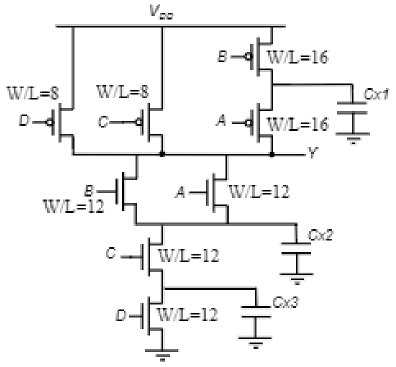
Q:What logic function does the circuit implement? To which logic family does the circuit belong? Does the circuit have any advantages over fully complementary CMOS? [Nov 2019]



Solution:

- ✓ The circuit implements Out = (A+BC)'. It is in the pseudo NMOS family.
- ✓ The circuit uses less area than a fully complementary CMOS implementation.

Q:Consider the circuit of Figure **[Nov 2019]**



What is the logic function implemented by the CMOS transistor network? Size the NMOS and PMOS devices so that the output resistance is the same as that of an inverter with an NMOS W/L = 4 and PMOS W/L = 8.

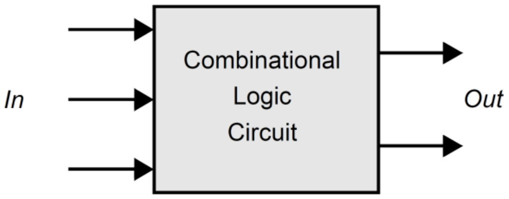
Solution:
The logic function is: Y = [(A+B) CD]’. The transistor sizes are given in the figure above.

TWO MARK QUESTIONS & ANSWERS

UNIT II - COMBINATIONAL LOGIC CIRCUITS

1. **Define combinational circuit and give an example.**

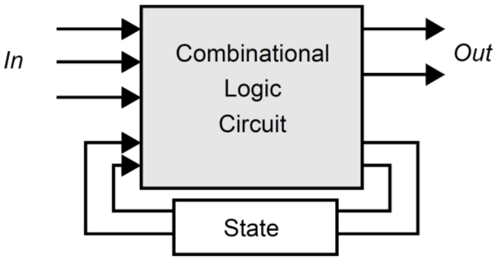
A combinational circuit can be defined as a circuit, whose output is dependent only on the inputs. Example: full adder.



(a) Combinational

2. **Define sequential circuit and give an example.**

A sequential circuit can be defined as a circuit, whose output depends not only on the present value of its inputs but on the sequence of past inputs. Example: flip-flop.



(b) Sequential

3. **What is the static CMOS inverter?**

Static CMOS inverter circuit is the combination of nMOS pulldown and pMOS pullup network.

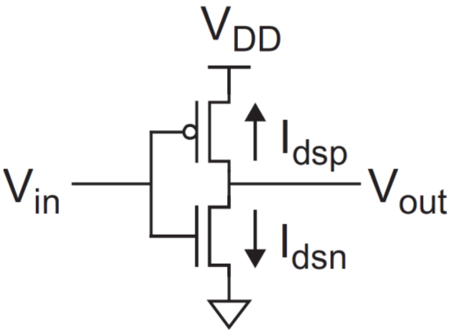


Figure: Static CMOS inverter

4. **What are the advantages of static CMOS circuits?**

Advantages of static CMOS circuits:

- Static CMOS circuits have good noise margins
- Static CMOS circuits are fast, low power, easy to design.
- Static CMOS circuits are widely supported by CAD tools,

- Static CMOS circuits are available in standard cell libraries.

5. **What are the disadvantages of static CMOS circuits?**

   Disadvantages of static CMOS circuits:
   - It requires both nMOS and pMOS transistor on each input.
   - It has large logical effort.
   - Gate delay is increased.

6. **What is bubble pushing? (May 2010)**

   DeMorgan's law :

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

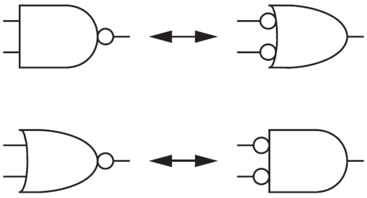


Figure: Bubble pushing with DeMorgan's law

- A NAND gate is equivalent to an OR of inverted inputs.
- A NOR gate is equivalent to an AND of inverted inputs.
- The same relationship applies to gates with more inputs.
- Switching between these representations is easy and is often called bubble pushing.

7. **What is meant by compound gate?**

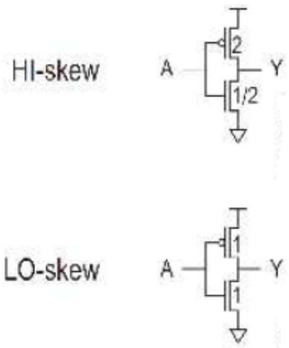
   Static CMOS efficiently handles compound gates computing various inverting combinations of AND/OR functions in a single stage.

8. **What is the function of skewed gate?**

   One input transition is more important than the other. HI-skew gates to favor the rising output transition and LO-skew gates to favor the falling output transition.

9. **What are the types of skewed gate?**

   Two types of skewed gate are HI-skew gate and LO-skew gate.



10. Define P/N ratio.

P/N ratio is defined as the ratio of pMOS to nMOS transistor width. For processes, a mobility ratio of µn/µp = 2.

11. What is meant by ratioed logic?

In ratioed logic, a gate consists of an nMOS pull-down network that realizes the logic function and a simple load device, which replace the entire pull-up network.

12. What is meant by pseudo nMOS logic?

A pseudo nMOS logic (ratioed logic) which uses a grounded pMOS load is referred to as a pseudo nMOS gate.

13. Draw a pseudo nMOS inverter.(Nov 2011)

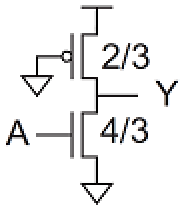


Figure: Pseudo nMOS inverter

14. What are the disadvantages of using a pseudo nMOS gate instead of a full CMOS gate?(May 2012)

What is the drawback of pseudo nMOS logic?

- Pseudo-nMOS gates will not operate correctly if (Maximum low level output) VOL >VIL (Maximum low level input) of the receiving gate.
- Ratioed circuits dissipate power continually in certain states and have poor noise margin.
- Ratioed circuits used in situations where smaller area is needed.

15. What are advantages and disadvantages of ratioed logic?

Advantage: Stronger static loads produce faster rising outputs.

Disadvantages:

a. Degrade the noise margin and burn more static power when the output is 0.

b. A resistor is a simple static load, but large resistors consume a large layout area in typical MOS processes.

16. Compare CMOS combinational logic gates with reference to the equivalent nMOS depletion load logic with reference to the area requirement.(May 2012)

For CMOS, the area required is 533µm^2, for pseudo nMOS the area required is 288 µm^2

17. What is AOI logic function?

AND OR Invert logic function (AOI) implements operation in the order of AND, OR, NOT operations. So this logic function is known as AOI logic function.

18. What is AOI 221 Gate?

AOI 221, here 221 refers to number of inputs in each section.

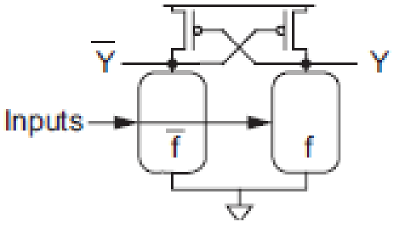
19. What is meant by Asymmetric Gates?

When one input is far less critical than another, even nominally symmetric gates can be made asymmetric to favor the late input at the expense of the early one.

20. What is meant by Cascode Voltage Switch Logic?

Cascode Voltage Switch Logic (CVSL) seeks the benefits of ratioed circuits without the static power consumption.

It uses both true and complementary input signals and computes both true and complementary outputs using a pair of nMOS pulldown networks.



21. What are the advantages of Cascode Voltage Switch Logic?
State the reasons for the speed advantages of CVSL family. (Nov 2012)

Advantage: CVSL has a potential speed advantage, because all of the logic is performed with nMOS transistors, thus reducing the input capacitance.

22. Define rise & fall time. [April 2008, Nov/Dec-2008] [Nov/Dec-2009]

Rise time (tr):

- It is defined as time for a waveform to rise from 20% to 80% of its steady state value.

Fall time (tf):

- It is defined as time for a waveform to fall from 80% to 20% of its steady-state value.

23. What is edge rate?

Edge rate is defined as an average value of rise time and fall time.

Edge rate $(t_{rf)}) = (t_r + t_f)/2$.

24. What do you mean by propagation delay time?

Propagation delay time (tpd) (or) Maximum delay is defined as maximum time from the input crossing 50% to the output crossing 50%.

25. What do you mean by contamination delay time?

Contamination delay time (tcd) (or) Minimum delay is defined as minimum time from the input crossing 50% to the output crossing 50%.

26. What is meant by average contamination delay time?

Average contamination delay time (**$t_{cd)}$** is defined as an average value of rising contamination delay time (**$t_{cdr)}$** and falling contamination delay time (**$t_{cdf)}$**.

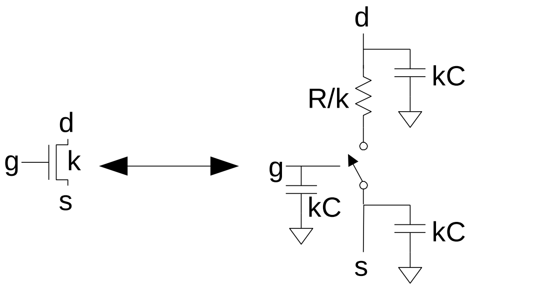
Contamination delay time ($t_{cd)}$ = (t_{cdr} + t_{cdf})/2

27. What is meant by RC delay model?

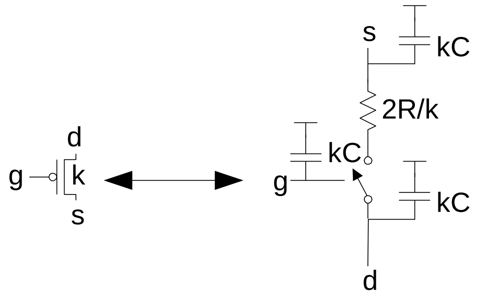
RC delay model approximates the nonlinear transistor I-V and C-V characteristics with an average resistance and capacitance over the switching range of the gate.

28. Draw equivalent RC delay model for a MOS transistor?

Equivalent RC delay model for an nMOS and pMOS transistor:



nMOS transistor



pMOS transistor

29. Define electrical or fanout.

Electrical effort is defined as ratio of the output capacitance to input capacitance of a gate.

Electrical effort (h) = C_{out} / C_{in}

30. What is parasitic delay?

The parasitic delay (P) of a gate is the delay of the gate when it drives zero load. It can be estimated with RC delay models.

31. Write the general expression of parasitic delay for n inputs NAND and NOR gate?

Expression of parasitic delay for n inputs NAND and NOR is n. Where, n – no. of inputs.

32. Write the expression for the logical effort and parasitic delay of n input NOR gate. [Nov/Dec-2011]

Logical effort for n inputs NOR gate is (2n+1)/3

Parasitic delay for n inputs NOR gate is n

33. What is meant by dynamic logic?

- Dynamic logic using a clocked pullup transistor rather than a pMOS that is always ON.

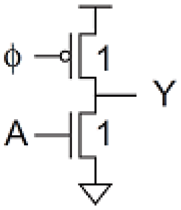


Figure: Dynamic logic

34. What are the two modes of operation in dynamic logic and give its functions? (NOV 2021)

Dynamic circuit operation has two modes, as shown in Figure.

(i) During precharge, the clock ϕ is 0, so the clocked pMOS is ON and output Y is high.

(ii) During evaluation, the clock is 1 and the clocked pMOS turns OFF. The output may remain high or may be discharged low through the pulldown network.

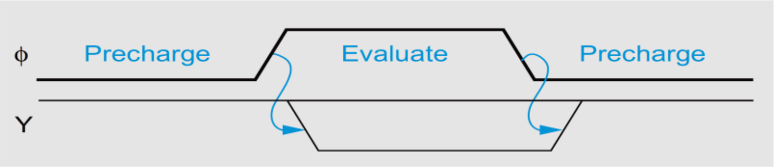


Figure: Precharge and evaluation of dynamic gates

35. What are the disadvantages of dynamic logic?

Disadvantages of dynamic logic:

- Dynamic circuits require careful clocking.
- Dynamic circuits consume significant dynamic power.
- Dynamic circuits are sensitive to noise during evaluation mode.
- Monotonicity problem
- Dynamic circuits suffer from charge leakage.

36. What are the advantages of dynamic logic?

Advantages of dynamic logic:

- Dynamic circuit has lower input capacitance and no contention during switching.
- Zero static power dissipation.

37. What is the use of footed transistor in dynamic logic circuit?

An extra clocked evaluation transistor can be added to the bottom of the nMOS stack to avoid contention as shown in the below figure. The extra transistor is called a foot.

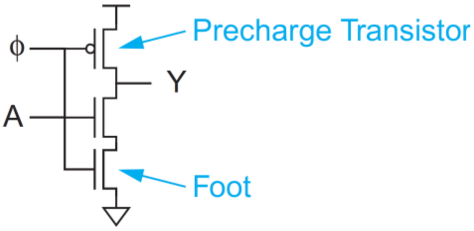


Figure: Footed dynamic inverter

38. What is meant by Monotonicity problem?

- During precharge, the output is pulled HIGH. When the clock rises, the input is HIGH, so the output is discharged LOW through the pulldown network. The input later falls LOW, turning OFF the pulldown network.
- However, the precharge transistor is also OFF, so the output floats, staying LOW rather than rising. This is called monotonicity problem in dynamic circuit.

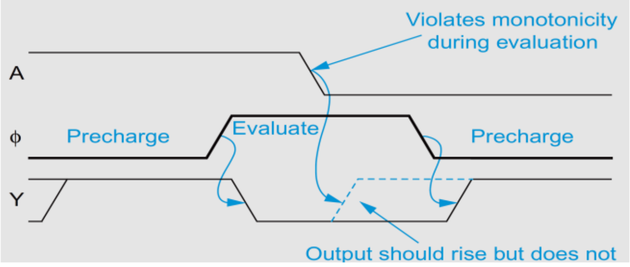

Violates monotonicity during evaluation
A
ϕ
Precharge
Evaluate
Precharge
Y
Output should rise but does not

Figure: Monotonicity problem

39. What is meant by domino logic?

The monotonicity problem can be solved by placing a static CMOS inverter between dynamic gates. This converts the monotonically falling output into a monotonically rising signal. The dynamic-static pair together is called domino logic.

40. Write the features of CMOS Domino Logic?

Features of CMOS Domino Logic:

- These structures occupy small area.
- Parasitic capacitance is to be small to increase the speed.
- Each gate can make one ‘logic 1’ to ‘logic 0’ transition.

41. What is the use of keeper circuit?

The keeper is a weak transistor that holds or *staticizes* the output at the correct level when it floats.

42. What is meant by pass transistors?

In pass-transistor circuits, inputs are applied to the source/drain diffusion terminals.
A single nMOS or pMOS pass transistor suffers from a threshold drop.

43. Which MOS can pass logic 1 and logic 0 strongly?

p-MOS can pass strong logic 1.
n-MOS can pass strong logic 0.

44. What is meant by CMOS Transmission gate? (Nov 2007, May 2011)(or) Define Transmission gate. (May 2009)

A parallel pair of nMOS and pMOS transistors is called *transmission gate*.
Transmission gates solve the threshold drop problem but require two transistors in parallel.

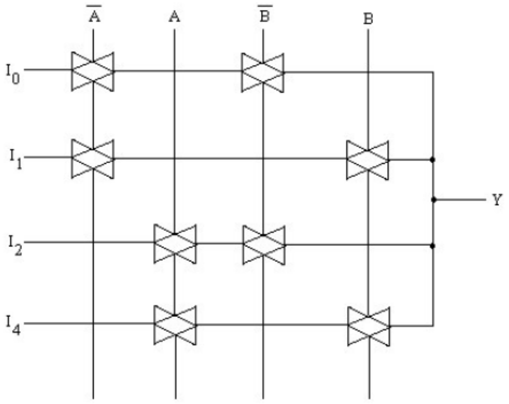
45. State the advantages of Transmission gate. (April 2017, May 2021)

Transmission gates solve the threshold drop problem.

It provides good conducting path between input and output.

46. Draw the CMOS implementation of 4-to-1 MUX using transmission gates.[Nov/Dec 2022]

CMOS implementation of 4-to-1 MUX using transmission gates:



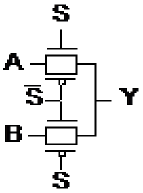
47. What are the various forms of inverter based CMOS logic?

Various forms of inverter based CMOS logic:

i. Pseudo nMOS logic
ii. Dynamic CMOS logic
iii. Clocked CMOS logic
iv. CMOS domino logic

48. Draw 2:1 MUX using transmission gate. (Nov 2008, APRIL-2015, 2016)[April/May 2023]

2:1 MUX using transmission gate:

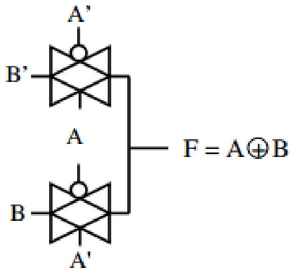


49. Draw XOR and XNOR using transmission gates. [Apr/may-2010]

- **2 I/P XOR using TGs:**

F = A.B' + A'.B , we need this: if A=1 → F = B' (pass B' to F)
if A=0 → F = B (pass B to F)

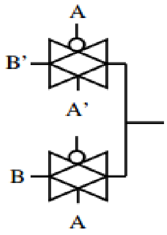
using TGs:



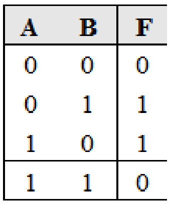
8 Ts (2 inverters for A and B and two TGs)
Versus 12 Ts for regular CMOS

2 I/P XNOR

F = A.B + A'B' if A=1 → pass B to F
if A=0 → pass B' to F

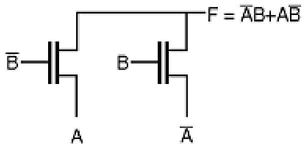


50. Draw a two input XOR using nMOS pass transistor logic. April 2019



A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Truth table of 'XOR' gate



'XOR' gate using pass transistor logic

51. Define power dissipation. [Nov/Dec-2013]

Power dissipation is defined as power consumed by the transistor unnecessarily, therefore increasing the power requirement to the logic.

52. List the types of power dissipation. [APRIL 2015, April 2018, Nov 2017]
List the various power losses in CMOS circuits. (May 2013)

Types of power dissipation are static and dynamic power dissipation.

53. What do you understand by static & dynamic power dissipation?
State the various types of power dissipation. (April 2019)

- Dynamic power dissipation is power consumed by transistor when it operates.

- At some stage both transistor pMOS and nMOS are in ON stage which, leads to short circuit formation between V_{DD} and GND, thus unwanted power dissipation occurs.
- Static power dissipation is power consumed by transistor when it is not in operating stage.

54. What do you mean by low power design?

When both static and dynamic powers are reduced then, the circuit is said to be low power designed circuit.

55. What are the factors that cause dynamic power dissipation in CMOS circuits? (Nov 2016, NOV 2021)

Dynamic dissipation due to

- Charging and discharging load capacitances as gates switch.
- "Short-circuit" current while both pMOS and nMOS stacks are partially ON.

56. How can dynamic power dissipation reduced? (or) State any two criteria for low power logic design. (Nov 2015, MAY 2014)

Dynamic power dissipation ($P_{dynamic}$) expressed as below,

$$P_{\text{dynamic}} = \alpha C V_{DD}{}^{2} f$$

To reduce dynamic power, use the following

- α: clock gating, sleep mode
- C: small transistors (esp. on clock), short wires
- V_{DD}: lowest suitable voltage
- f: lowest suitable frequency

57. Write the expression for power dissipation in CMOS inverter. [Nov/Dec-2008]

Total power dissipation P_{total} is the sum of dynamic power dissipation ($P_{dynamic}$) and static power dissipation (P_{static}).

$P_{total} = P_{dynamic} + P_{static}$

Where, $P_{\text{dynamic}} = \alpha C V_{DD}{}^{2} f$

- α:activity factor
- C: capacitor
- V_{DD}: Supply voltage
- f: Supply frequency

$$P_{static} = (I_{sub} + I_{gate} + I_{junc} + I_{contention})V_{DD}$$

58. What are the factors that cause static power dissipation in CMOS circuits? [Nov-2012] List the sources of static power consumption. (Nov 2016, NOV 2021)

Static dissipation due to

- Subthreshold leakage through OFF transistors
- Gate leakage through gate dielectric
- Junction leakage from source/drain diffusion

- Contention current in ratioed circuits

59. How can static power dissipation reduced?

To reduce static power

- Selectively use ratioed circuits
- Selectively use low V_t devices
- Leakage reduction: Use stacked devices, body bias and low temperature

60. Why single phase dynamic logic structure cannot be cascaded? Justify.(May 2016)

No, single phase dynamic logic structure cannot be cascaded. Because monotonicity problem will be raised, so static logic should be used in between dynamic logics structure.

61. What is Complementary Pass Transistor logic? (NOV/DEC-2014)

Complementary Pass Transistor logic has complementary data inputs and outputs.

It reduces the count of transistors used to make different logic gates, by eliminating redundant transistors.

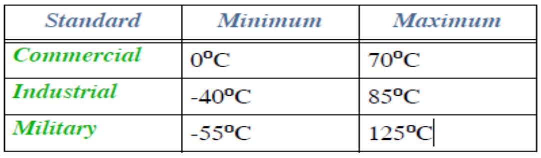
62. Give the effects of supply voltage and temperature variations CMOS circuits. [Nov-2012]

Supply Voltage:

- Supply voltage may vary due to tolerance of voltage regulators, IR drop along the supply rail and di/dt noise.
- Typically the supply is specified as +/- 10% around nominal (uniform distribution).
- Speed is proportional to VDD, also noise budgets are affected.

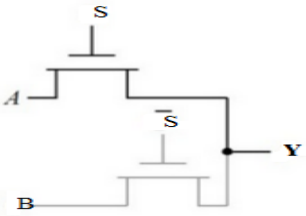
Temperature

Parts must operate over a range of temperatures.



Standard	Minimum	Maximum
Commercial	0°C	70°C
Industrial	-40°C	85°C
Military	-55°C	125°C

63. Implement a 2:1 multiplexer using pass transistor. (NOV/DEC-2013, April 2015)

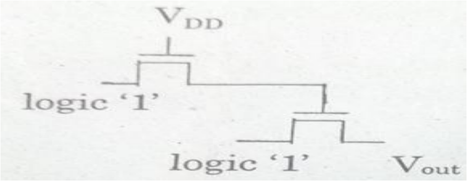


64. Compare static and dynamic power dissipation. [Nov 2019]

- Static power is power consumed while there is no circuit activity. For example, the power consumed by a D flip-flop when neither the clock nor the D input have active inputs (i.e., all inputs are "static" because they are at fixed dc levels).

- Dynamic power is power consumed while the inputs are active. When inputs have ac activity, capacitors are charging and discharging and the power increases as a result. The dynamic power includes both the ac component as well as the static component.

65. What is the value of V_{out} for the figure shown below, where V_{tn} is threshold voltage of transistor? (Nov 2016)



Output voltage, $V_{out} = V_{DD}-2V_{tn}$, Where V_{tn} : Threshold voltage

66. How does a transmission gate produce fully restored logic output? (NOV 2021)

A transmission gate is parallel pairs of nMOS and pMOS transistor. A single nMOS or pMOS pass transistor suffers from a threshold drop. Transmission gates solve the threshold drop but require two transistors in parallel.

One of the transistors is passing the value well and the other is passing it poorly. A logic-1 is passed well through the pMOS but poorly through the nMOS. A logic-0 is passed well through the nMOS but poorly through the pMOS.

67. What is charge sharing in dynamic CMOS logic? [Nov/Dec-2022]

Charge sharing problem occurs when the charge which is stored at the output node in the pre-charge phase is shared among the junction capacitance of transistor in the evaluation phase. Charge sharing may degrade the output voltage level or even cause an erroneous output value.

68. What is use of transmission gates? [April/May-2022]., [Nov/Dec-2020 & April/May-2021]

Used as a

- Logic structure
- Switch
- Latch
- Used solution to deal with the voltage-drop problem.
- Complex gates can be implemented using minimum number of transistors, which also reduces parasitics.

69. List the sources of power dissipation in CMOS circuits. [April/May-2022]

Static CMOS design:

- Bubble Pushing
- Compound gates
- Skewed gates

Dynamic CMOS design:

- Dual rail domino logic

- Multiple output domino logic

****

Design a half adder using static CMOS logic. [Nov/Dec 2022]

$S = A \oplus B$ ...... Sum

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

Let $\bar{S} = \overline{\bar{A}B + A\bar{B}}$

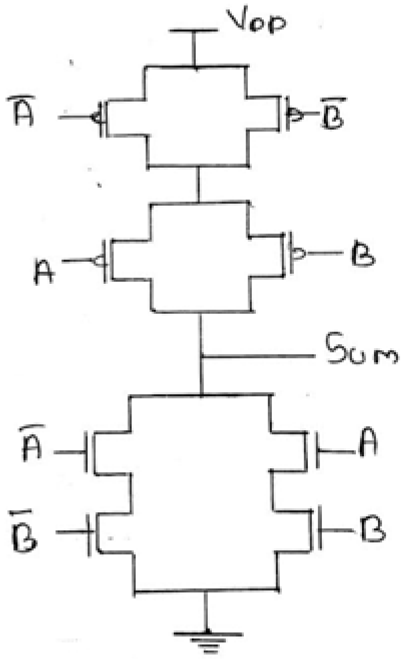
$\therefore \bar{\bar{S}} = \bar{A}\bar{B} + AB$ ......implement as pull down network (nMOS)

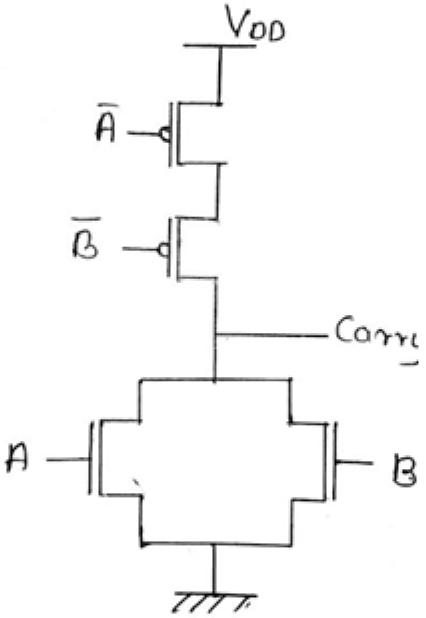
$\bar{S}_{dual} = (\bar{A} + \bar{B}).(A + B)$ ......Implement as pull up network (pMOS)

$C = A.B$ ..... Carry

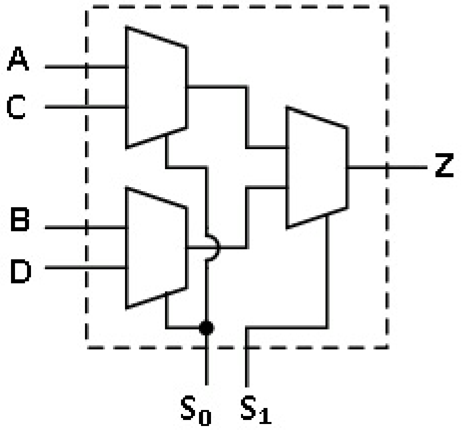
Let $\bar{C} = \overline{A.B}$ ......implement as pull down network (nMOS)

$\bar{C}_{dual} = \bar{A} + \bar{B}$ ......Implement as pull up network (pMOS)





Design a 4:1 MUX using 2:1 MUX. Realize it using transmission gate. [Nov/Dec 2022]



$$Z=\left(A.\overline{S}_0.\overline{S}_1\right)+\left(B.\overline{S}_0.S_1\right)+\left(C.S_0.\overline{S}_1\right)+\left(D.S_0.S_1\right)$$

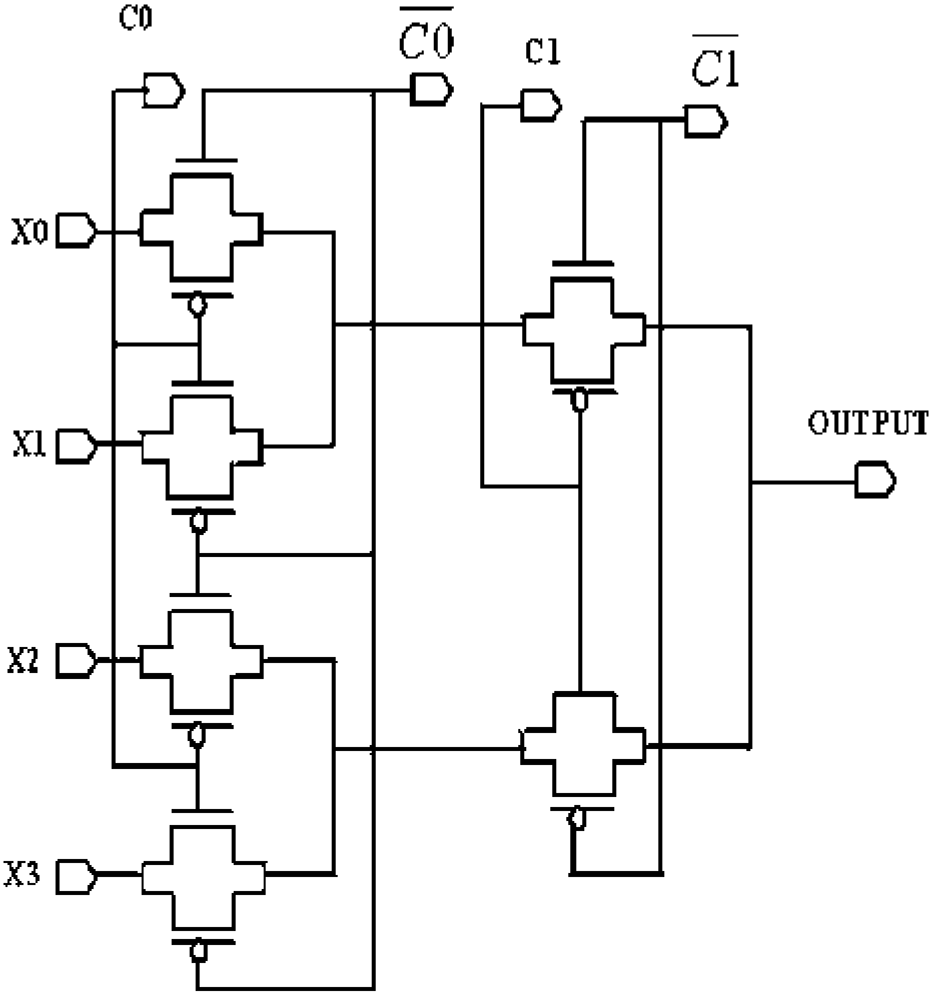
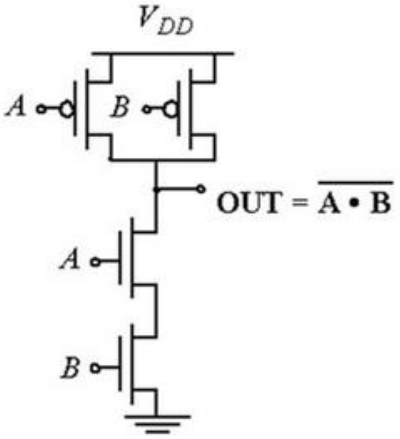


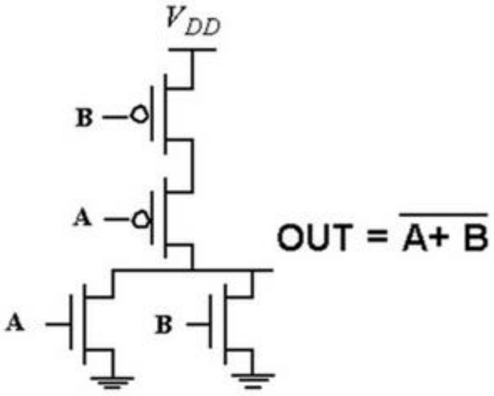
Fig: Transmission gate of 4:1 MUX using 2:1 MUX

Realize a 2-input NOR gate, NAND gate, XOR gate, XNOR gate using static CMOS logic.[Apr/May 2022]

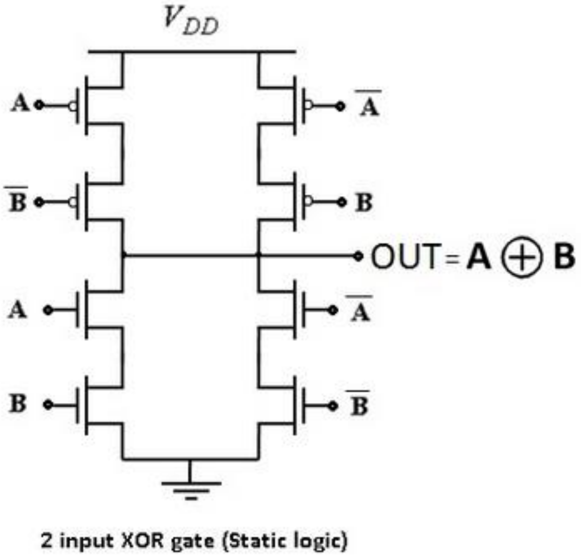
Realize a 2-input NOR gate using static CMOS logic, Domino logic and Complementary pass transistor logic. Analyze the hardware complexity in terms of transistor count. [Nov/Dec 2022]



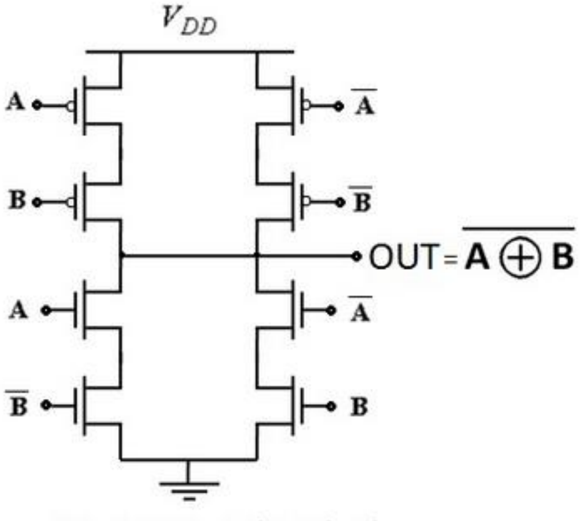
2 input NAND gate (Static Logic)



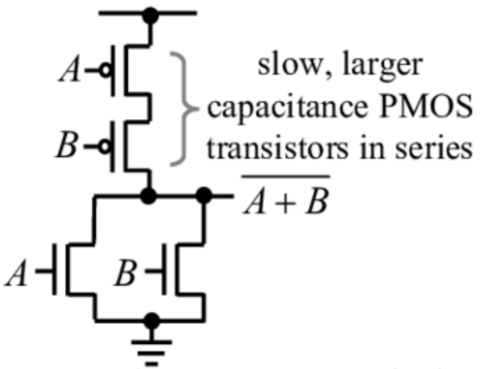
2 input NOR gate (Static Logic)



2 input XOR gate (Static logic)

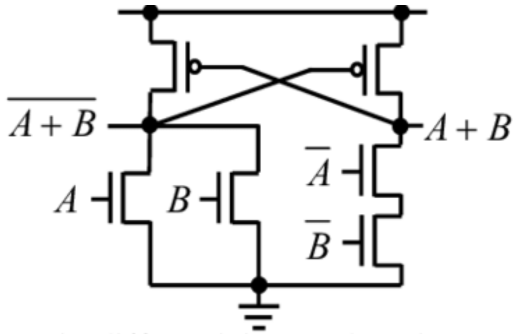


2 input XNOR gate (Static logic)

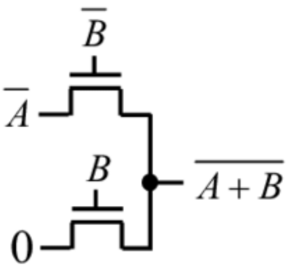


slow, larger capacitance PMOS transistors in series

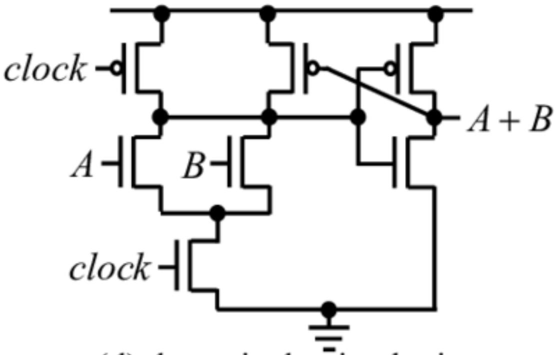
(a) complementary CMOS logic



(b) differential cascode voltage switch logic (DCVSL)



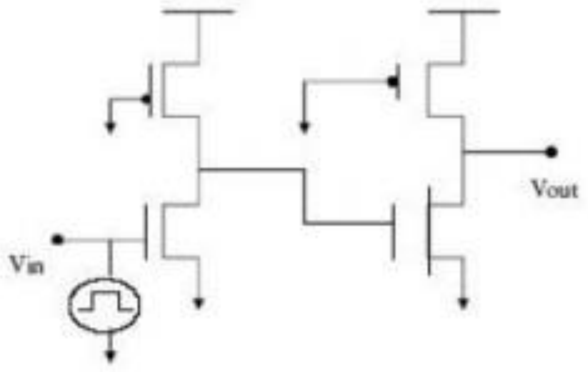
(c) pass transistor logic (PTL)



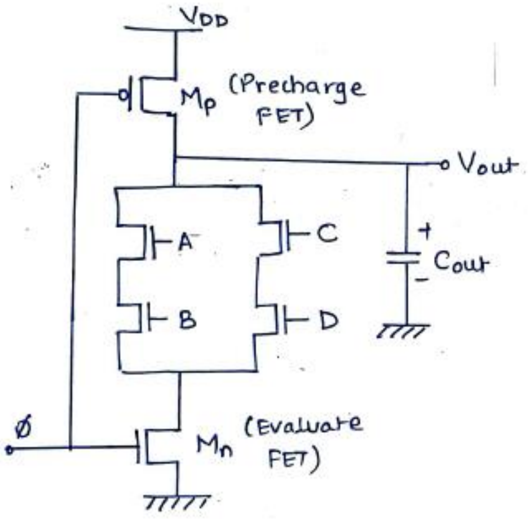
(d) dynamic domino logic

Sketch a combinational function Y=(AB+CD)’.
(i) Pseudo-nMOS logic
(ii) Domino logic
(iii) Cascode voltage switch logic. [Nov/Dec 2022] [April / May 2023]

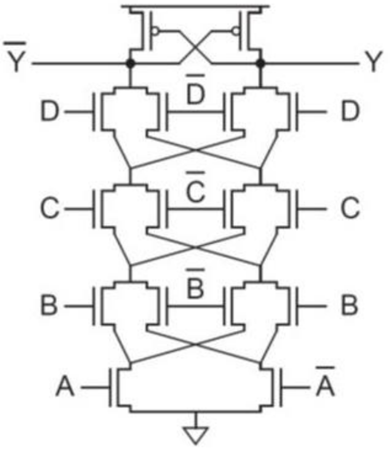
Pseudo-nMOS logic



Domino logic



Cascode voltage switch logic



70. List out the advantages and disadvantages of Pass Transistor Logic. [April/May 2023]

The **advantages** of pass-transistor logic are the simple design, the reuse of already available signals, and the low contribution to static power.

The **disadvantage** of PTL is that the output voltage is lower than the input and it does not allow series connection of a large number of transistors.

71. List any two types of layout design rules. (Nov 2008, Nov 2009, May 2010)

Two types of layout design rules:

a. Lambda design rules

b. Micron rules

72. What are design rules?
What is the need for design rules? (NOV.2014)

Design rules are a set of geometrical specifications that dictate the design of the layout masks.

Design rules are used to produce workable mask layouts from which the various layers in silicon will be formed or patterned.

73. Define the lambda layout rules. (May 2013)
What is meant by lambda layout design rules?
Discuss any two layout design rules. (MAY 2014, APRIL2015, Nov 2015, Nov 2008, Nov 2009, May 2010)

Lambda-based rules are round up dimensions of scaling to an integer multiple of scalable parameter λ. Lambda rules make scaling layout as small. The same layout can be moved to a new process by specifying a new value of λ.

Micron rules can result in as much as a 50% size reduction over lambda rules. Industry usually uses the micron design rules for layouts.

74. By what factor, gate capacitance must be scaled if constant electric field scaling is employed? (April 2019)

Gate capacitance is scaled by scaling factor $\frac{1}{S}$ in constant electric field scaling.

75. What are stick diagrams?

Stick diagrams are used to convey layer information through the use of a color code. A stick diagram is a cartoon of a chip layout. The stick diagram represents the rectangles with lines which represent wires and component symbols.

76. What are the uses of Stick diagram?

Uses of stick diagram:

- It can be drawn much easier and faster than a complex layout.
- These are especially important tools for layout built from large cells.

77. Give the various color coding used in stick diagram?

Various color coding used in stick diagram:

- Green – n-diffusion, Red- polysilicon, Blue –metal, Yellow- implant and Black-contact areas.

78. Why does interconnect increase the circuit delay? [Nov/Dec-2011]

Interconnect is defined by its resistance value and capacitance with neighbor. Delay is calculated from resistance and capacitance value.

79. What is transistor sizing problem?(MAY 2014)

Transistor sizing is carried out by equating the maximum on resistances of the logic circuit with inverter one.

80. What is CMOS latchup? How do you prevent Latch up problem? (Nov 2008) (or)
What is Latch up problem in CMOS circuits? (May 2008, April 2016)

Latch up is a condition in which the parasitic components give rise to the establishment low resistance conducting paths between V_{DD} and V_{SS} with disastrous results.

Careful control during fabrication is necessary to avoid this problem.

The remedies for the latch-up problem include:

(i) An increase in substrate doping levels.

(ii) Reducing Rnwell.
(i) By introducing guard rings.
(ii) By introducing SOI (Silicon On Insulator)

81. What is BiCMOS Gate?
Compare CMOS and BiCMOS technology. (NOV. 2013)

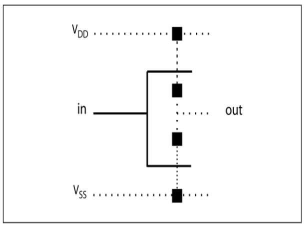
When bipolar and MOS technology are merged, the resulting circuits are referred to as biCMOS circuits. It improves bandwidth and current gain.

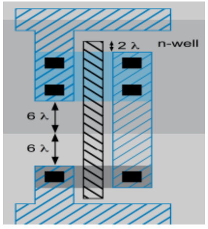
82. What is the need of demarcation line? (Nov 2017)

In CMOS, a demarcation line is drawn to avoid touching of p-diff with n-diff. All pMOS must lie on one side of the line and all nMOS must lie on other side.

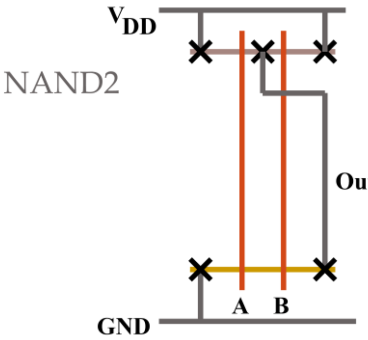
83. Draw the stick diagram and layout for CMOS inverter. (Nov 2016)

Stick diagram **Layout diagaram**





84. Draw the stick diagram of static CMOS 2-input NAND gate. (April 2018)



85. What are simulations available for VLSI circuits?

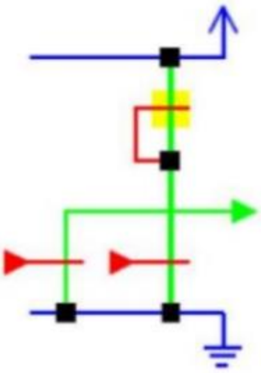
In VLSI the following simulations are available

- Path simulation, Monte Carlo simulation and Interconnect simulation

86. Why nMOS transistor is selected as pull down network? (Nov 2017)

Pull-up and pull-down networks in CMOS circuits are never both conducting and are never both opened at the same time. This is the reason that **nMOS** transistors are used in the **pull-down** network and pMOS in the **pull**-up network of a CMOS gate.

87. Draw the stick diagram of NMOS NOR gate. [Nov 2019]



88. How do you describe the term device modeling? [May/June-2013]

The device modeling describes how to model diffusion capacitance and how to run simulations in various process corners.

89. What is Elmore's delay model? (or) Give the expression for Elmore delay and state the various parameters associated with it. (NOV. 2014, April 2016, 2017, 2018, Nov 2017) [April/May – 2023]

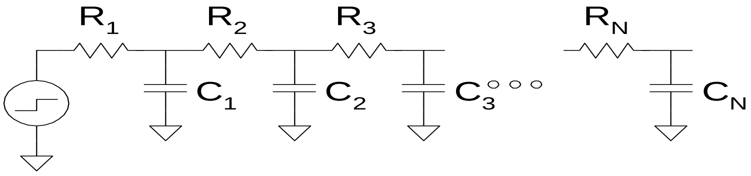
The Elmore delay model estimates the delay from a source switching to one of the leaf nodes. Delay is summing over each node i of the capacitance C_i on the node multiplied by the effective resistance R.

Propagation delay time:

$$t_{pd} \approx \sum_{\text{nodes } i} R_{i-to-source} C_i$$

$$= R_1 C_1 + \left(R_1 + R_2\right) C_2 + \ldots + \left(R_1 + R_2 + \ldots + R_N\right) C_N$$

RC delay equivalent for series of transistors:



90. Define logical effort and give logical effort value of inverter.

Logical effort (g) is defined as the ratio of the input capacitance of a gate to the input capacitance of an inverter delivering the same output current.

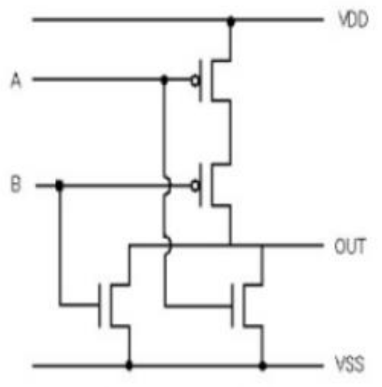
An inverter has a logical effort of 1.

91. Write the general expression of logical effort for n inputs NAND and NOR gate?

Expression of logical effort for n inputs NAND is (n+2)/3.

Expression of logical effort for n inputs NOR is (2n+1)/3. Where, n – no. of inputs.

92. Draw a 2- input CMOS NOR Gate. [Nov/Dec-2022]



NOR gate in CMOS

93. Write the expression for parasitic delay and logical effort of an N-input NAND gate. [April/May-2022]

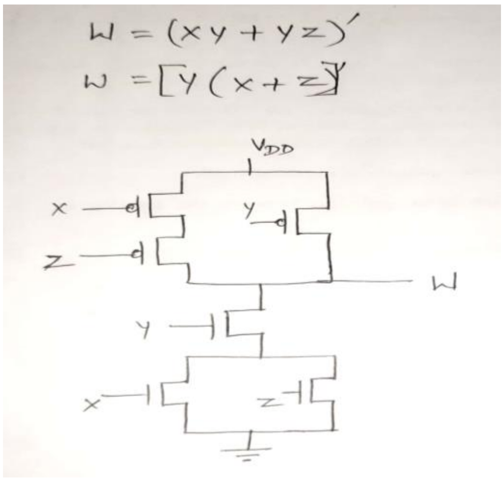
Solution :

Parasitic delay for N-input NAND gate = n

Logical effort of an N-input NAND gate = (n+2)/3

94. Sketch a complementary CMOS gate computing W = (XY+YZ)’. [April/May-2022]

Sketch a complementary CMOS gate computing Y = (AB+BC)’. [Nov/Dec-2020, April/May-2021]



95. What is body effect. [Nov/Dec-2020, April/May-2021]

V_t is not constant with respected to voltage difference between substrate and source of MOS transistor. This is known as body effect. Its other name is substrate-bias effect.

96. What is velocity saturation effect?

The velocity of charge carriers is linearly proportional to the eclectic field and the proportionally constant mobility of carrier.

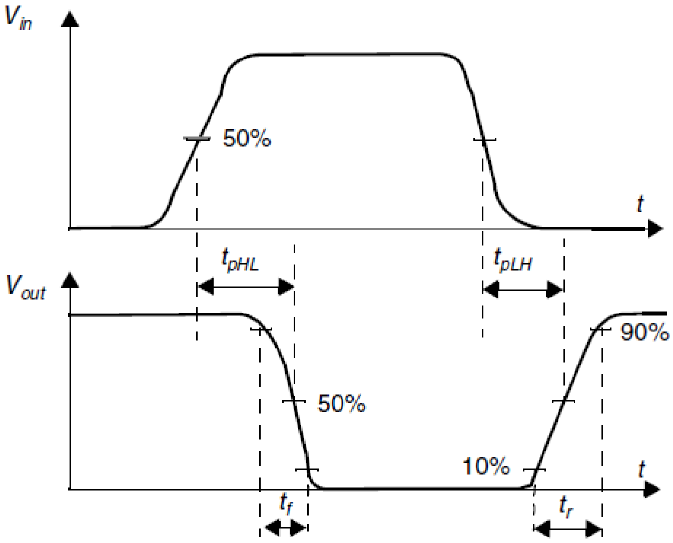
When we increase the electric field beyond certain velocity called as the **thermal velocity or saturation velocity.**

Electron attains in the presence of very high electric fields.

At high electric field carriers fail to follow this linear model.

97. Define propagation delay of a CMOS inverter.

It expresses the delay experienced by a signal when passing through a gate. It is measured between the **50%** transition points of the input and output waveforms, as shown in Figure for an inverting gate. Because a gate displays different response times for rising or falling input waveforms, two definitions of the propagation delay are necessary. The **t_{pLH}** defines the response time of the gate for a low to high (or positive) output transition, while **t_{pHL}** refers to a high to low (or negative) transition. The propagation delay **t_p** is defined as the average of the two.



$$t_p = \frac{t_{pLH} + t_{pHL}}{2}$$

Figure Definition of propagation delays and rise and fall times.

98. Why NMOS device conducts strong zero and weak one?

<u>Reason:</u>

- Greater switching speed.
- Weak transistor is used to generate a high output voltage level
- The gate is “high” and channel is ‘low’.
- Nmos is turn on. Vg is at V_{DD} V_S charging towards V_{DD}.

99. What is Intrinsic and Extrinsic Semiconductor?

The pure Silicon is known as Intrinsic Semiconductor. When impurity is added with pure Silicon, its electrical properties are varied. This is known as Extrinsic Semiconductor.

100. State the channel length modulation. Write the equation for describing channel length modulation effect in NMOS transistor.

Channel length is varied due to changes in **V_{ds}** (drain to source voltage). In saturation region, channel length is decreased when (W/L) ratio is increased. So β is increased and drain voltage is increased.

101. What is latch up? How is Prevent latch up?

Latch up is the condition occurs in the circuit manufactured using bulk CMOS technology. When IC is the state of "Latch Up".

Latch Up Prevention in two ways:

- Latch up resistant CMOS processes
- Layout technique.

102. What are the different MOS layers?

- n-diffusion
- p-diffusion
- Polysilicon
- Metal

103. If two CMOS inverters are cascaded with an aspect ratio of 1:1 then determine the inverter pair delay. [Nov/Dec-2022]

Solution :

Logical Effort of the inverter , g = 1

Here, single identical load. So, the electrical effort , h =1

Parasitic delay of an inverter , P_{inv} =1

Then, the delay of each stage is expressed as,

d = gh + p

= 1(1)+1

= 2

104. Differentiate static and dynamic latches and registers. [Nov/Dec-2020., April/May-2021]

Difference between Static latches & registers and Dynamic latches & registers:

Static Latches & registers	Dynamic Latches & registers
A latch is an essential component in the construction of an edge triggered register. It is low sensitive circuit that passes the D input to the Q output when clock signal is High.	Dynamic latches storage in a static sequential circuit relies on the concept that a cross coupled inverter pair.
This latch is said to be in transparent mode	It produces a bistable element and can thus be used to memorize binary values.
A static latches stores its output in a static state.	A dynamic latch uses temporary capacitance to store its state.
This register rarely or never changes.	This register are changeable

UNIT – III

SEQUENTIAL LOGIC CIRCUITS AND CLOCKING STRATEGIES

Static Latches and Registers, Dynamic Latches and Registers, Pipelines, Non-bistable Sequential Circuits, Timing classification of Digital Systems, Synchronous Design, Self-Timed Circuit Design .

3.1 Static Latches and Registers:

- ❖ **Discuss in detail various static latches and registers. (Nov 2016)**
- ❖ **Explain the methodology of sequential circuit design of Latches. (May 2014)**
- ❖ **Discuss the operation of a CMOS latch. (Nov 2007)**

3.1.1 The Bi-stability Principle

- Static memories use positive feedback to create a bistable circuit. A bistable circuit has two stable states that represent 0 and 1.
- The basic idea is shown in Figure 3.1a, which shows two inverters connected in cascade along with a voltage-transfer characteristic (VTC).
- The output of the second inverter Vo2 is connected to the input of the first V i1, as shown by the dotted lines in Figure 3.1a.
- The resulting circuit has only three possible operation points (A, B, and C).
- A and B are stable operation points, and C is a metastable operation point.

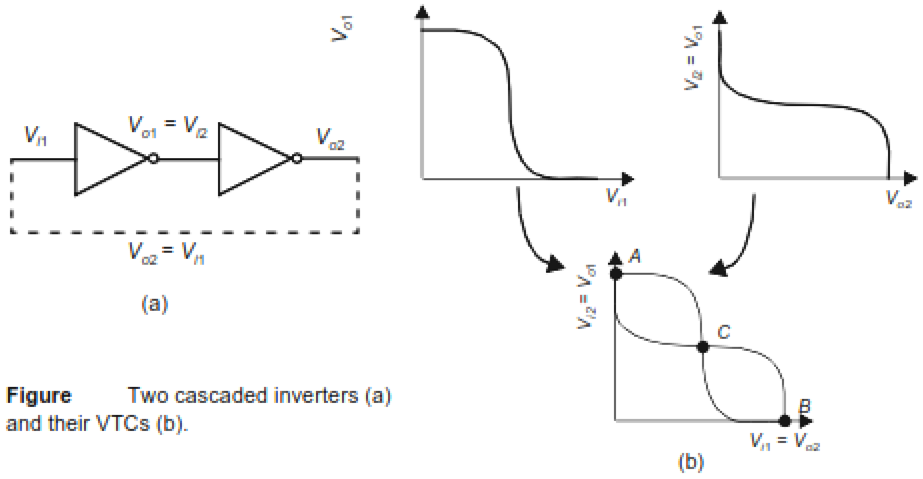
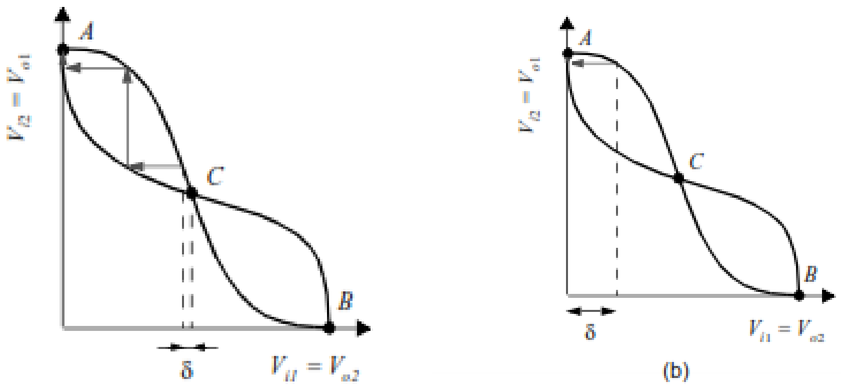


Figure 3.1: a. Two inverters connected in cascade b. VTCs

- Cross-coupled inverter pair is biased at point C. It is amplified and regenerated around the circuit loop.
- The bias point moves away from C until one of the operation points A or B is reached.
- C is an unstable operation point. Every deviation causes the operation point to run away from its original bias. Operation points with this property are termed as metastable.

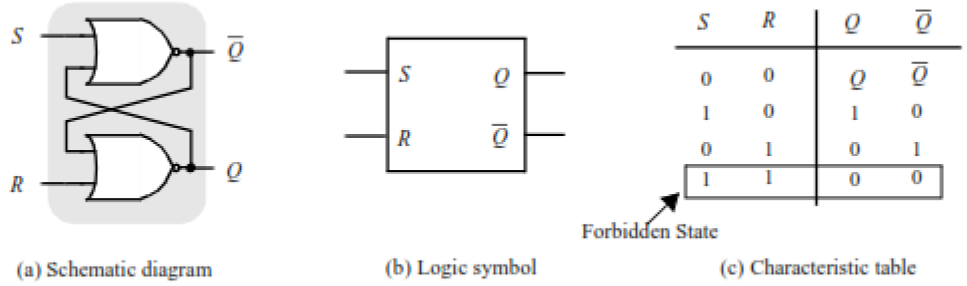


(a) Figure 3.2 Metastability.

- A bistable circuit has two stable states. In absence of any triggering, the circuit remains in a single state.
- A trigger pulse must be applied to change the state of the circuit.
- Common name for a bistable circuit is flip-flop.

3.1.2 SR Flip-Flops

- The SR or set-reset flip-flop implementation is shown in Figure (a) below.
- This circuit is similar to the cross-coupled inverter pair with NOR gates replacing the inverters.
- The second input of the NOR gates is connected to the trigger inputs (S and R), that make it possible to force the outputs Q and Q_{bar}.
- These outputs are complimentary (except for the SR = 11 state).
- When both S and R are 0, the flip-flop is in a quiescent state and both outputs retain their value.
- If a positive (or 1) pulse is applied to the S input, the Q output is forced into the 1 state (with Qbar going to 0).
- Vice versa, a 1 pulse on R resets the flip-flop and the Q output goes to 0.



S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
1	0	1	0
0	1	0	1
1	1	0	0

(a) Schematic diagram (b) Logic symbol (c) Characteristic table

Figure 3.3

- When both S and R are high, both Q and Q_{bar} are forced to zero. This input mode is considered to be forbidden.
- An SR flip-flop can be implemented using a cross-coupled NAND structure as shown in Figure 3.4

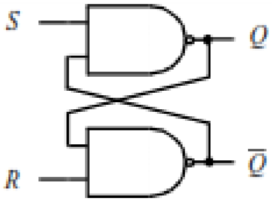


Figure 3.4 NAND based SR flip-flop

Clocked SR flip-flop:

- Clocked SR flip-flop (a level-sensitive positive latch) is shown in Figure 3.5.
- It consists of a cross-coupled inverter pair, plus 4 extra transistors to drive the flip-flop from one state to another and to provide clocked operation.
- Consider the case where Q is high and R pulse is applied.
- The combination of transistors M_4, M_7, and M_8 forms a ratioed inverter.
- In order to make the latch switch, we must succeed in bringing Q below the switching threshold of the inverter M_1-M_2.
- Once this is achieved, the positive feedback causes the flip-flop to invert states. This requirement forces to increase the sizes of transistors M_5, M_6, M_7, and M_8.

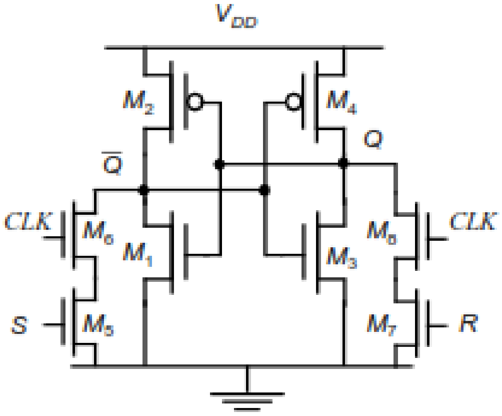


Figure 3.5 CMOS clocked SR flip-flop

- The clocked SR flip-flop does not consume any static power.

3.1.3 Multiplexer Based Latches:

- Multiplexer based latches can provide similar functionality to the SR latch.
- But sizing of devices only affects performance and is not critical to the functionality.
- Figure 3.6 shows an implementation of static positive and negative latches based on multiplexers.
- For a negative latch, when the clock signal is low, the input 0 of the multiplexer is selected, and the D input is passed to the output.
- When the clock signal is high, the input 1 of the multiplexer connected to the output of the latch.
- The feedback holds the output stable while the clock signal is high.

- Similarly in the positive latch, the D input is selected when clock is high and the output is held (using feedback) when clock is low.

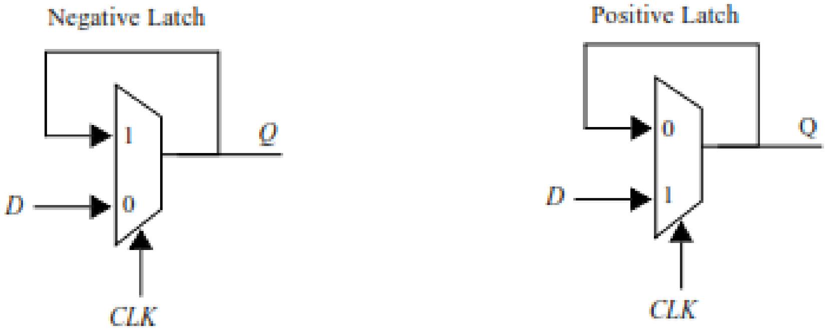


Figure 3.6 Negative and positive latches based on multiplexers.

❖ **Design a d-latch using transmission gate. (May 2015)**

❖ **Design a D-latch using transmission gate. Using which realize a two phase non – overlapping master slave negative edge triggered D flip-flop. (April 2019-13M)**

❖ **Illustrate the circuit designs for basic latches, then build the flip-flops and pulsed latches. [May 2021][Nov/Dec 2022]**

- A transistor level implementation of a positive latch is shown in Figure 3.7.
- When CLK is high, the bottom transmission gate is ON and the latch is transparent – i.e, the D input is copied to the Q output.
- During this phase, the feedback loop is open due to the top transmission gate is OFF.

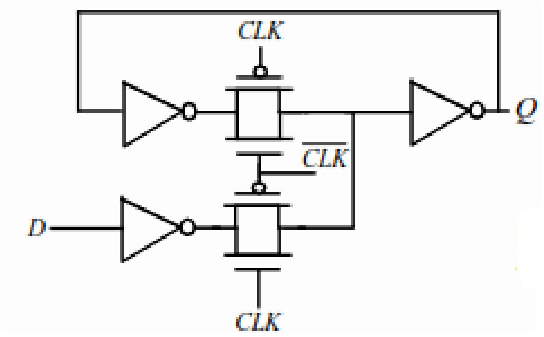


Figure 3.7 Transistor level implementation of a positive latch built using transmission gates.

- To reduce the clock load, implement a multiplexer based NMOS latch using two pass transistors as shown in Figure 3.8.
- The advantage of this approach is the reduced clock load of only two NMOS devices.
- When CLK is high, the latch samples the D input, while a low clock-signal enables the feedback-loop and puts the latch in the hold mode.

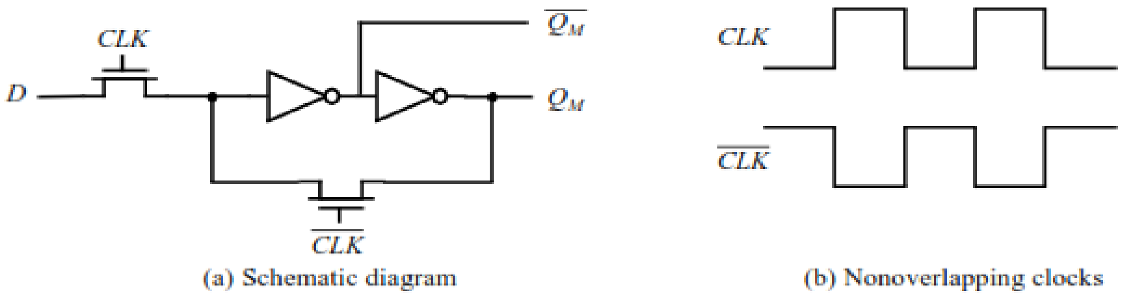


Figure 3.8 Multiplexer based NMOS latch using NMOS only pass transistors for multiplexers.

- ❖ **Explain the operation of master-slave based edge triggered register. (May 2016)**
- ❖ **Draw and explain the operation of conventional CMOS, pulsed and resettable latches. (Nov 2012)**
- ❖ **Discuss about CMOS register concept and design master slave triggered register, explain its operation with overlapping periods. (April 2018, NOV 2018)**
- ❖ **Realize a negative level sensitive latch using which realize an edge triggered master slave D-Flip flop. Explain its working. (Nov 2019) [April / May 2023]**

3.1.4 Master-Slave Based Edge Triggered Register:

- An edge-triggered register is to use a master-slave configuration as shown in Figure 3.9.
- The register consists of cascading a negative latch (master stage) with a positive latch (slave stage).
- A multiplexer based latch is used to realize the master and slave stages.
- On the low phase of the clock, the master stage is transparent and the D input is passed to the master stage output, Q_M.
- During this period, the slave stage is in the hold mode, keeping its previous value.
- On the rising edge of the clock, the master slave stops sampling the input and the slave stage starts sampling.
- During the high phase of the clock, the slave stage samples the output of the master stage (Q_M), while the master stage remains in a hold mode.
- A negative edge-triggered register can be constructed using the same principle by simply switching the order of the positive and negative latch (i.e., placing the positive latch first).

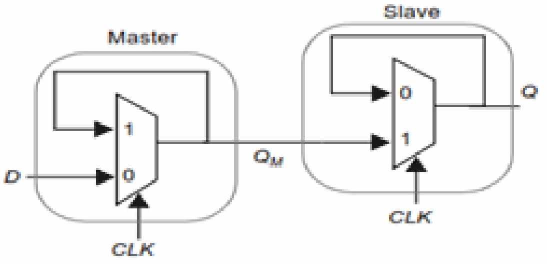


Figure 3.9: Positive edge-triggered register based on a master-slave configuration.

- A complete transistor level implementation of the master-slave positive edge-triggered register is shown in Figure 3.10.
- When clock is low (CLK bar = 1), T_1 is ON and T_2 is OFF and the D input is sampled onto node Q_M.
- During this period, T_3 is OFF and T_4 is ON and the cross-coupled inverters (I_5 & I_6) hold the state of the slave latch.

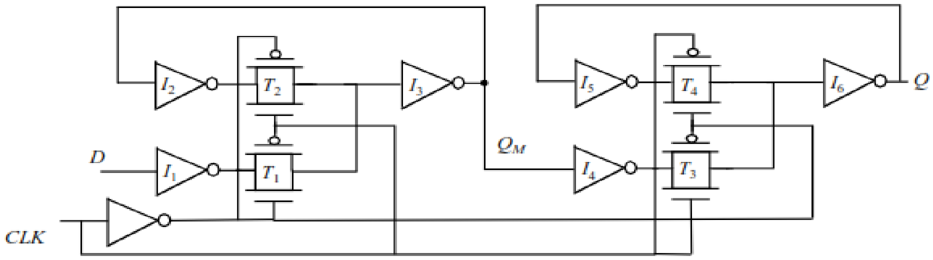
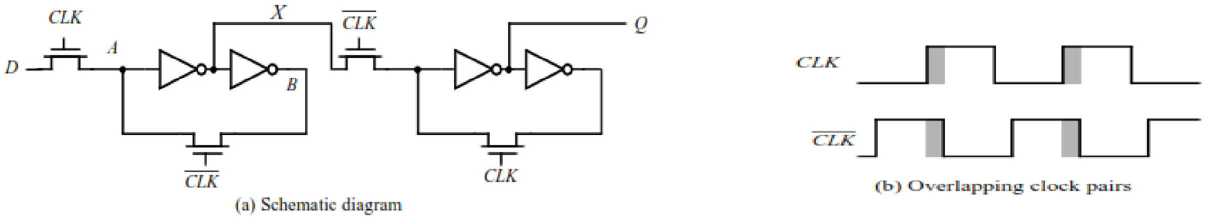


Figure 3.10 Transistor-level implementation of a master-slave positive edge-triggered register using multiplexers.

- When the clock goes high, the master stage stops sampling the input and goes into a hold mode.
- T_1 is OFF and T_2 is ON and the cross coupled inverters I_3 and I4 hold the state of Q_M. Also T_3 is ON and T_4 is OFF and Q_M is copied to output Q.

3.1.5 Non-ideal clock signals:

- We have assumed that $\overline{CLK}$ is a perfect inversion of CLK.
- Even if this was possible, this would still not be a good assumption.
- Variations can exist in the wires. It is used to route the two clock signals or the load capacitances can vary based on data stored in the connecting latches.
- This effect, known as clock skew is a major problem and causes the two clock signals to overlap as is shown in Figure 3.11 b.
- Clock-overlap can cause two types of failures, as illustrated for the NMOS-only negative master-slave register of Figure 3.11 a.



(a) Schematic diagram

(b) Overlapping clock pairs

Figure 3.11 Master-slave register based on NMOS-only pass transistors.

- When the clock goes high, the slave stage should stop sampling the master stage output and go into a hold mode.
- However, since CLK and $\overline{CLK}$ are both high for a short period of time (the overlap period).

- Both sampling pass transistors conduct and there is a direct path from the D input to the Q output.
- As a result, data at the output can change on the rising edge of the clock, which is undesired for a negative edge triggered register.
- The is known as a race condition in which the value of the output Q is a function of whether the input D arrives at node X before or after the falling edge of $\overline{CLK}$.
- If node X is sampled in the metastable state, the output will switch to a value determined by noise in the system.
- Those problems can be avoided by using two non-overlapping clocks PHI and PH2.
- By keeping the non-overlap time $t_{non\_overlap}$ between the clocks large enough, such that no overlap occurs even in the presence of clock-routing delays.
- During the non-overlap time, the FF is in the high-impedance state.
- Leakage will destroy the state, if this condition holds for too long a time.
- Hence the name ***pseudostatic***: the register employs a combination of static and dynamic storage approaches depending upon the state of the clock.

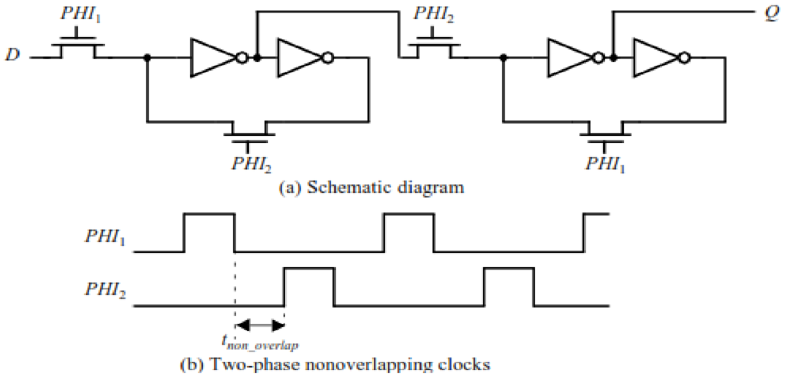


Figure 3.12 Pseudostatic two-phase D register.

3.1.6 Low-Voltage Static Latches:

- The scaling of supply voltages is critical for low power operation.
- At very low power supply voltages, the input to the inverter cannot be raised above the switching threshold, resulting in incorrect evaluation.
- Scaling to low supply voltages, hence requires the use of reduced threshold devices.
- The shaded inverters and transmission gates are implemented in low-threshold devices.
- The low threshold inverters are gated using high threshold devices to eliminate leakage.
- During normal mode of operation, the sleep devices are tuned on.
- During idle mode, the high threshold devices in series with the low threshold inverter are turned OFF (the SLEEP signal is high), eliminating leakage.

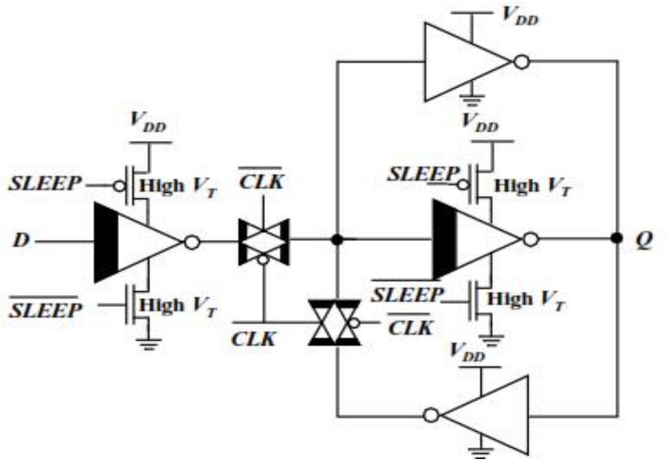


Figure 3.13: One solution for the leakage problem in low-voltage operation using MTCMOS.
****

3.2 Dynamic Latches and Registers:

- ❖ **Discuss about the design of sequential dynamic circuits. (Nov 2012, Nov 2017)**
- ❖ **Explain the methodology of sequential circuit design of flip-flop. (May 2014)**

- A stored value remains valid as long as the supply voltage is applied to the circuit, hence the name static.
- The major disadvantage of the static gate is, its complexity.
- Registers are used in computational structures that are constantly clocked, such as pipelined data path.
- The requirement that the memory should hold state for extended periods of time.
- This results in circuits, based on temporary storage of charge on parasitic capacitors.
- The principle is identical to the dynamic logic. In dynamic logic, logic signal is a charge, stored on a capacitor.
- The absence of charge denotes as logic 0 and presence of charge denotes as logic 1.
- A stored value can be kept for a limited amount of time (range of milliseconds).
- A periodic refresh of its value is necessary.

3.2.1 Dynamic Transmission-Gate Based Edge-triggered Registers:

❖ **Design a d-flipflop using transmission gate. (Nov 2016)**

- A dynamic positive edge-triggered register based on the master-slave concept is shown in Figure 3.14.
- When CLK = 0, the input data is sampled on storage node 1. It has an equivalent capacitance of C1 consisting of the gate capacitance of I_1, the junction capacitance of T_1, and the overlap gate capacitance of T_1.

- During this period, the slave stage is in a hold mode with node 2 in a high-impedance state.
- On the rising edge of clock, the transmission gate T_2 turns on. The value is sampled on node1 before the rising edge propagates to the output Q.
- Node 2 stores the inverted version of node 1.
- The reduced transistor provides high-performance and low-power systems.

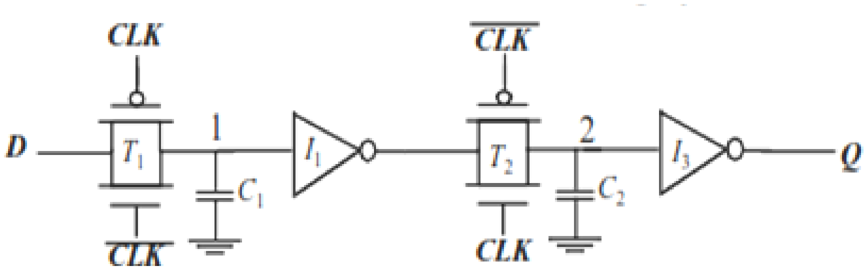


Figure 3.14 Dynamic edge-triggered register.

- The set-up time of this circuit is the delay of the transmission gate and time takes node 1 to sample the D input.
- The hold time is approximately zero, since the transmission gate is turned OFF.
- The propagation delay (t_{c-q}) is equal to two inverter delays plus the delay of the transmission gate T_2.
- Dynamic register has storage nodes (i.e., the state). A node has to be refreshed at periodic intervals to prevent a loss. Loss due to charge leakage and diode leakage.
- Clock overlap is an important for register. Consider the clock waveforms shown in Figure 3.15.
- During the 0-0 overlap period, the NMOS of T_1 and the PMOS of T_2 are simultaneously on.
- It is creating a direct path for data to flow from the D input of the register to the Q output. This is known as a race condition.
- The output Q can change on the falling edge, if the overlap period is large.
- Overlap period constraint is given as: $t_{overlab0-0} < t_{T1} + t_{I1} + t_{T2}$
- Similarly, the constraint for the 1-1 overlap is given as: $t_{hold} > t_{overlab1-1}$

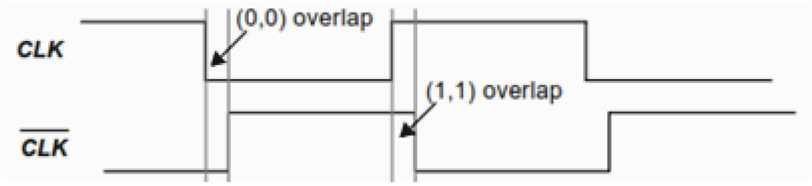


Figure 3.15 Impact of non-overlapping clocks

3.2.2 C^2MOS Dynamic Register:

The C^2MOS Register

- Figure 3.16 shows positive edge-triggered register based on the master-slave concept, which is insensitive to clock overlap. This circuit is called the C^2MOS (Clocked CMOS) register.

1. CLK = 0 ($\overline{CLK}$ = 1):
    - The first tri-state driver is turned ON. The master stage acts as an inverter, sampling the inverted of D on the internal node X.
    - The master stage is, in evaluation mode. The slave section is, in hold mode.
    - Both transistors M_7 and M_8 are OFF, decoupling the output from the input. The output Q retains its previous value stored on the output capacitor C_{L2}.

2. CLK = 1 ($\overline{CLK}$ = 0):
    - The master stage section is in hold mode (M_3-M_4 off), while the second section evaluates (M_7-M_8 on).
    - The value stored on C_{L1} propagates to the output node through the slave stage, which acts as an inverter.
    - The overall circuit operates as a positive edge-triggered master-slave register.

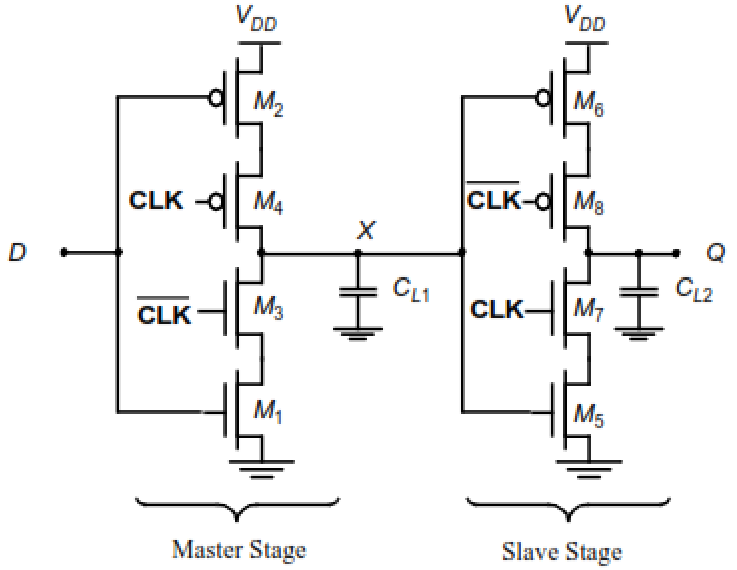


Figure 3.16 C^2MOS master-slave positive edge-triggered register.

- It is similar to the transmission-gate based register, presented earlier. However, there is an important difference.
- *A C^2MOS register with CLK-CLK clocking is insensitive to overlap, as long as the rise and fall times of the clock edges are small.*

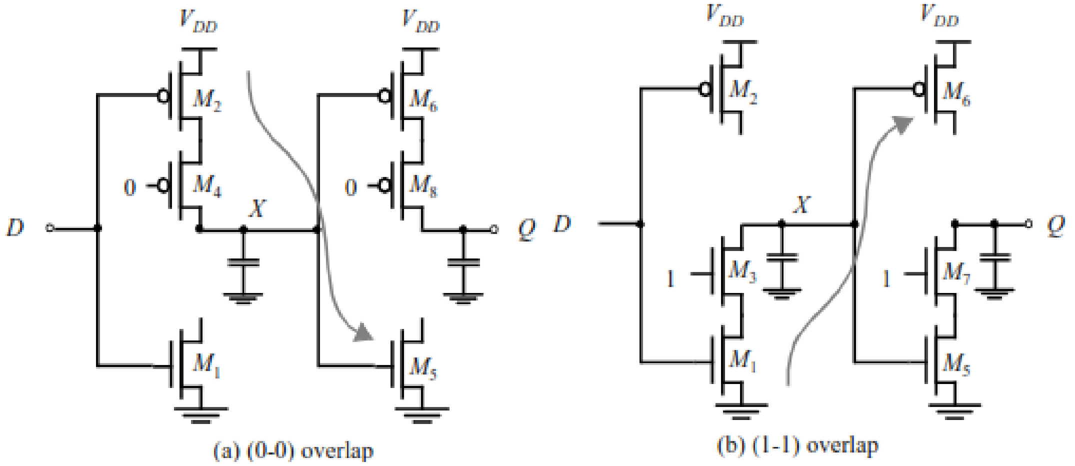


Figure 3.17 C^{2}MOS D FF during overlap periods.

3.2.3 True Single-Phase Clocked Register (TSPCR):

Explain the operation of True Single Phase Clocked Register. (Nov 2016, April 2017)

- In the two-phase clocking schemes, care must be taken in routing the two clock signals to ensure that overlap is minimized.
- While the C^{2}MOS provides a skew-tolerant solution, it is possible to design registers that only use a single phase clock.
- The True Single-Phase Clocked Register (TSPCR) uses a single clock without an inverse clock.
- Figure 3.19 shows positive and negative latch concept.
- For the positive latch, when CLK is high, the latch is in the transparent mode and propagates the input to the output. Latch has two cascaded inverters, so latch is non-inverting.
- When CLK = 0, both inverters are disabled and the latch is, in hold-mode.
- Only the pull-up networks are still active, while the pull-down circuits are deactivated.
- As a result of the dual-stage approach, no signal can ever propagate from the input to the output.
- For the negative latch, when CLK is low, the latch is in the transparent mode and propagates the input to the output.
- When CLK = 1, both inverters are disabled and the latch is in hold-mode.
- A register can be constructed by cascading positive and negative latches.
- <u>The main advantage is the use of a single clock phase.</u>
- <u>The disadvantage is, increase in the number of transistors (12 transistors are required).</u>

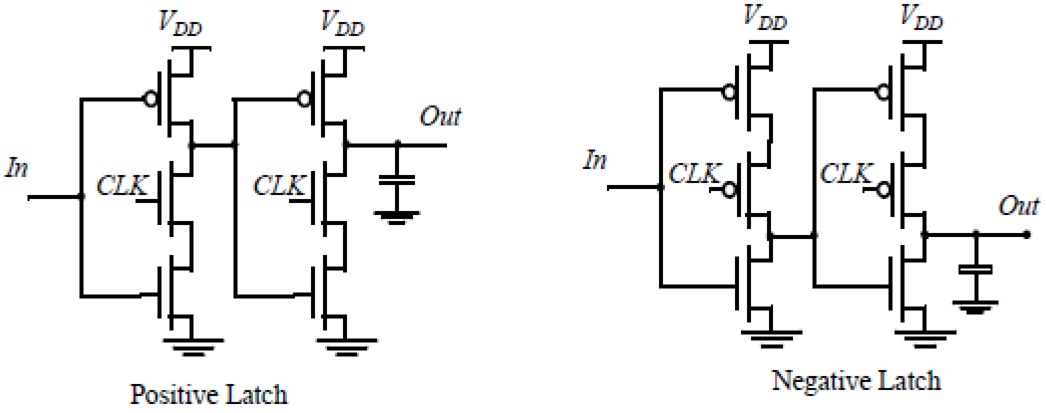


Figure 3.19 TSPC approach.

- The TSPC latch circuits can be reduced, as in Figure 3.20, where only the first inverter is controlled by the clock.
- Number of transistors are reduced and clock load is reduced by half.

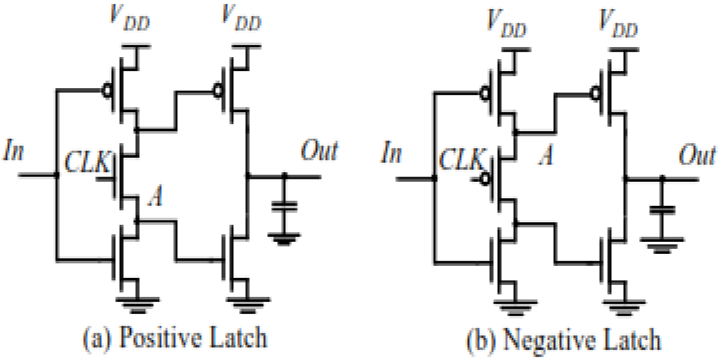


Figure 3.20 Simplified TSPC latch (also called split-output).

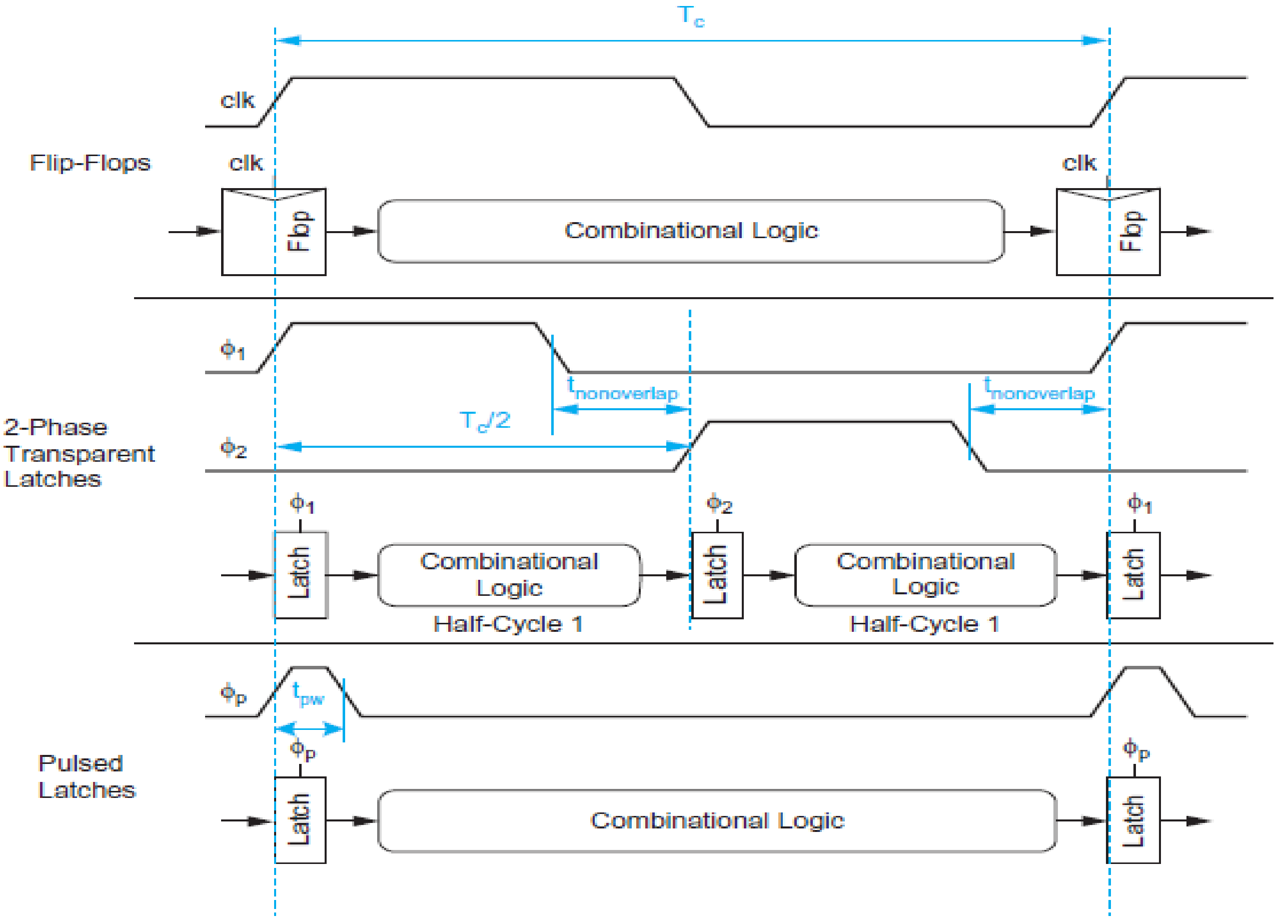
****

3.3 Timing Issues:

- ❖ **Explain in detail about timing issues needed for a logic operation. (April 2017)**
- ❖ **Explain the timing basics in synchronous design in detail. (Nov 2017)[April/May 2023]**
- ❖ **Discuss the timing parameters that characterize the timing of sequential circuit. (NOV 2021)**

(A)Sequencing methods:-

- Three methods of sequencing block of combinational logic are possible, as shown in figure below.
- In flip-flop based system, one flip flop use one cycle boundary.
- Token (data) advances from one cycle to the next on the rising edge. If a token arrives too early, it waits at the flip flop until next cycle.
- In 2-phase system, phases may be separated by $t_{nonoverlap}$. [$t_{nonoverlap}$>0]
- In pulsed system, pulse with is t_{pw}.



- In 2-phase system, full cycle of combinational logic is divided into two phases, sometimes called “half-cycles”. Two latch clocks are called $\varphi 1$ and φ 2.
- Flip flop can be viewed as, a pair of back to back latches using clk and its complements.
- Table shows delay and timing notations of combinational and sequencing elements. These delays may differ for rising (with suffix ‘r’) and falling (with suffix ‘f’).

TERM	***NAME***
T_{pd}	logic propagation delay
T_{cd}	logic contamination delay
T_{pcq}	latch flop clock-Q propagation delay
T_{ccq}	latch flop clock- to Q contamination delay
T_{pdq}	latch flop D –to Q propagation delay
T_{cdq}	latch flop clock D to Q contamination delay
T_{setup}	latch flop setup time
T_{hold}	latch flop hold time

- The delay with timing diagram for all three sequencing elements are, as shown in figure below.
- In combinational logic, input A changing to another value, output Y cannot change instantaneously. After the contamination delay {t_{cd}}, Y may begin to change (or) glitch.
- Output Y settles to a value in propagation delay {t_{pd}} .
- Input D in flip flop must have settled by some setup time {t_{setup}} before the rising edge of clock and should not change again until, a hold time {t_{hold}} after the clock edge.

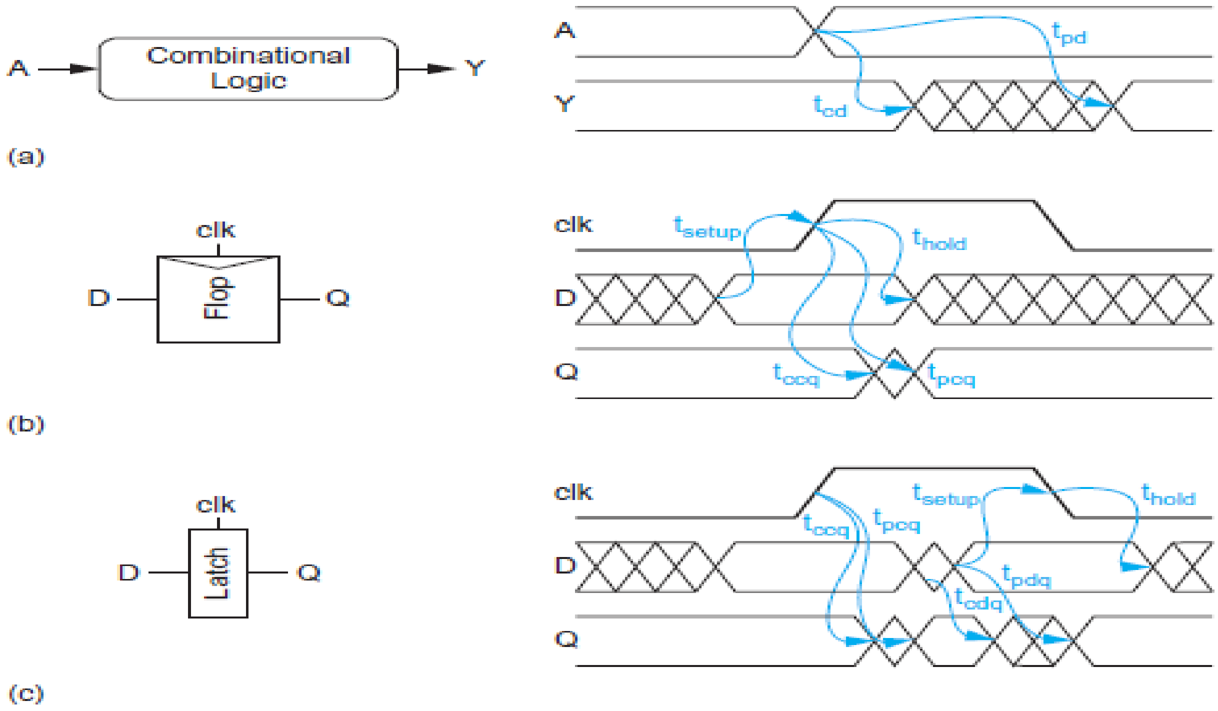
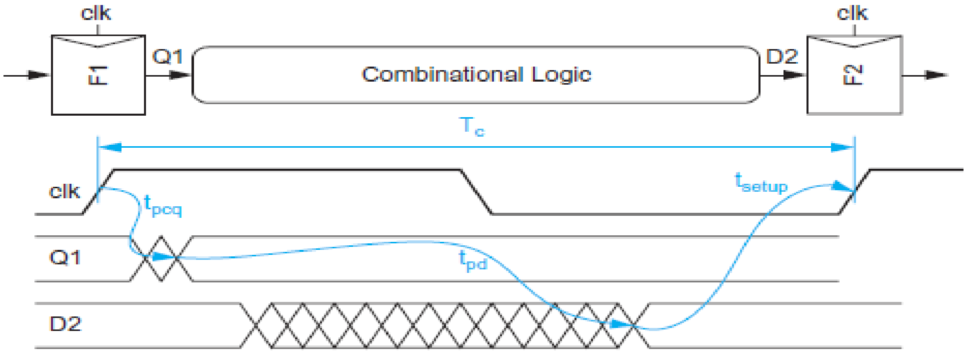


Figure: Timing diagrams

- The output begins to change after a clock-to-contamination delay {t_{ccq}} and completely settles after clock to-Q propagation delay {t_{pcq}}.

(B) Max Delay Constraints:-

- Ideally, the entire clock cycle will be available for computation in the combinational logic.
- If the combination logic delay is too high, the receiving element {next flop/latch} will miss its setup time and sample the wrong value.
- This is called as “setup-time failure” or “max-delay failure”.
- It can be solved by redesigning the logic to be faster (or) by increasing the clock period.



- The clock period must be $Tc \geq tpcq + tpd + tsetup$

(or)

$$Tpd \leq tc - (tsetup + tpcq)$$

Where, tsetup+tpcq – sequencing overhead.

(C) Min-delay constraints:-

- Sequencing elements can be placed back to back without intervening combinational logic and still function correctly.
- If the hold time is large and contamination delay is small, data can incorrectly propagate through successive elements, on one clock edge.
- This corrupt the state of the system called, race condition (or) hold time failure (or) min-delay failure.
- It can be fixed by redesigning the logic and not by slowing the clock.

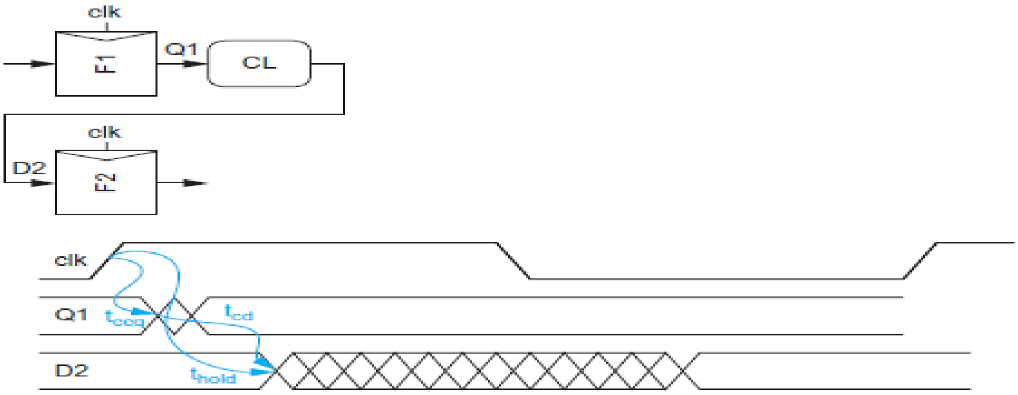


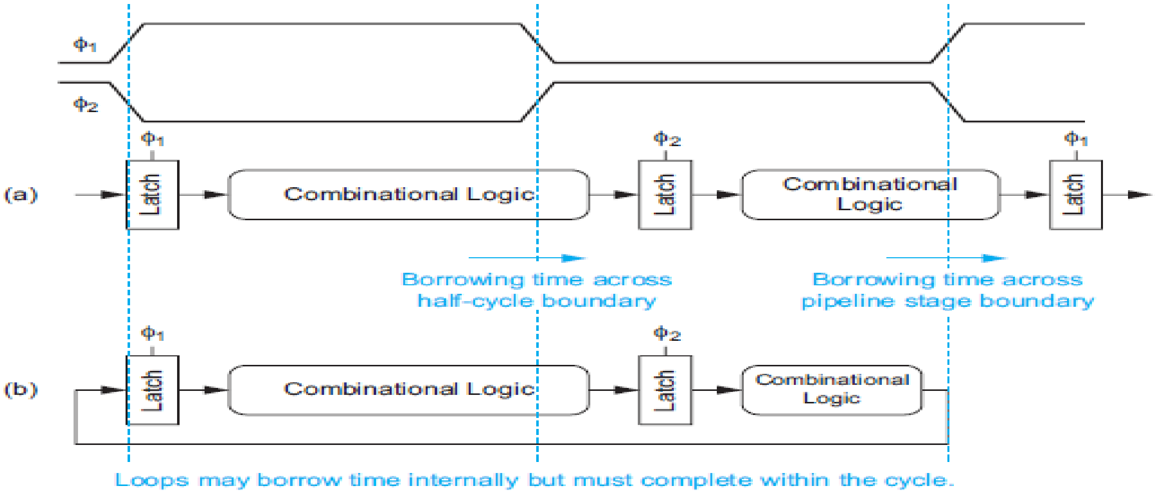
FIGURE Flip-flop latch min-delay constraint

$$tcd \geq thold - tccq$$

(D)Time Borrowing:

- In flip-flop, dada departs the first flip-flop on the rising edge of the clock and must set up at the second flip-flop before the next rising edge of the clock.
- If data arrives late, produces wrong result.
- If data arrives early, it is blocked until the clock edge arrives and the remaining time goes unused and clock imposes a "hard edge".

- If one half cycle (or) stage of a pipeline has too much logic, it can borrow time in half-cycle (or) stage.
- This is called as "Time borrowing", which can accumulate across multiple cycles.


Φ1
Φ2
(a)
Latch
Combinational Logic
Combinational Logic
Borrowing time across half-cycle boundary
Borrowing time across pipeline stage boundary
(b)
Loops may borrow time internally but must complete within the cycle.

$$tborrow \leq \frac{Tc}{2} - (tsetup - tnonoverlap)$$

(E) Clock Skew

Analyze the impact of spatial variations of clock signal on edge-triggered sequential logic circuits. (NOV 2018)

- The spatial variation in arrival time of a clock transition in an integrated circuit is referred as clock skew.
- The clock skew between two points i and j on a IC is given by $\delta(i,j)=t_i-t_j$, where t_i and t_j are the position of the rising edge of the clock with respect to a reference.
- The clock skew can be positive or negative, depending upon the routing direction and position of the clock source.
- The timing diagram for the case with positive skew, is shown in figure.
- In the figure, the rising clock edge is delayed by a positive δ at the second register.

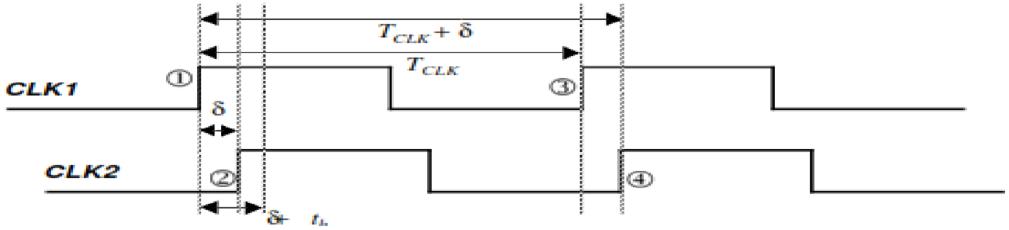


Figure: Timing diagram to study the impact of clock skew on performance and functionality. In this sample timing diagram, δ >0.

(F) Clock Jitter:

- Clock jitter is the temporal variation of the clock period at a given point. The clock period can reduce or expand on a cycle-by-cycle basis. It is a temporal uncertainty measure.
- Cycle-to-cycle jitter refers to time varying deviation of a single clock period.
- For a given spatial location, i is given as T_{jitter}, $i(n) = T_{i,n+1} - T_{i,n} - T_{CLK}$.

Where Ti,n is the clock period for period n, $T_{i,n+1}$ is clock period for period n+1, and T_{CLK} is the nominal clock period.

- Jitter directly impacts the performance of a sequential system.
- Figure shows the nominal clock period as well as variation in period.

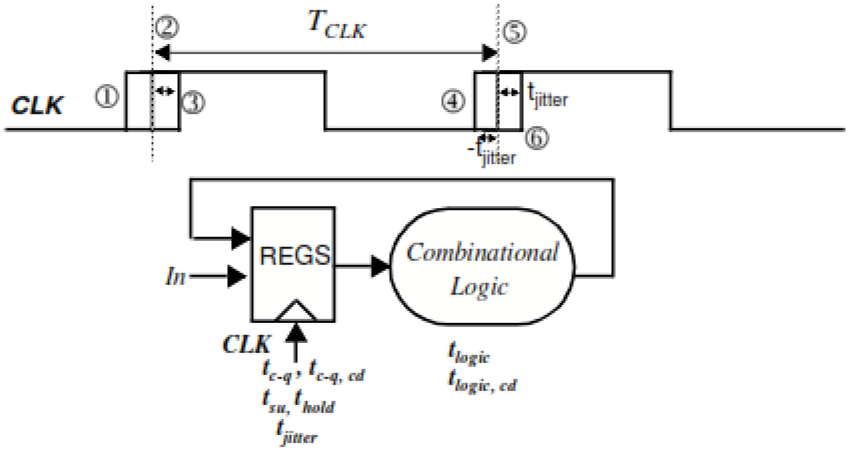


Figure: Circuit for studying the impact of jitter on performance.

****

3.4 Pipelining:

- ❖ **Explain in detail about pipelining structure needed for a logic operation. (April 2017, Nov 2017)**
- ❖ **Discuss in detail various pipelining approaches to optimize sequential circuits. (May 2013, 2016, May 2021)[Apr/May 2022] [April / May 2023]**

- Pipelining is a design technique used to accelerate the operation of the datapaths in digital processors.
- The idea is explained with Figure 3.22a.
- The goal of the circuit is to compute log(|a - b|), where both a and b represent streams of numbers.
- The minimal clock period T_{min} necessary to ensure correct evaluation is given as:

$$T_{\min} = t_{c-q} + t_{pd,\text{logic}} + t_{su}$$

Where, t_{c-q} and t_{su} are the propagation delay and the set-up time of the register respectively.

- Registers are edge-triggered D registers.
- The term $t_{pd,logic}$ stands for the worst-case delay path through the combinatorial network, which consists of the adder, absolute value and logarithm functions.
- In conventional systems, the delay is larger than the delays associated with the registers and dominates the circuit performance.
- Assume that each logic module has an equal propagation delay.
- Each logic module is then, active for only 1/3 of the clock period.
- Pipelining is a technique to improve the resource utilization and increase the functional throughput.
- Introduce registers between the logic blocks, as shown in Figure 3.22b.

- This causes the computation for one set of input data to spread over a number of clock periods, as shown in Table 1.
- The result for the data set (a1, b1) only appears at the output after three clock-periods.

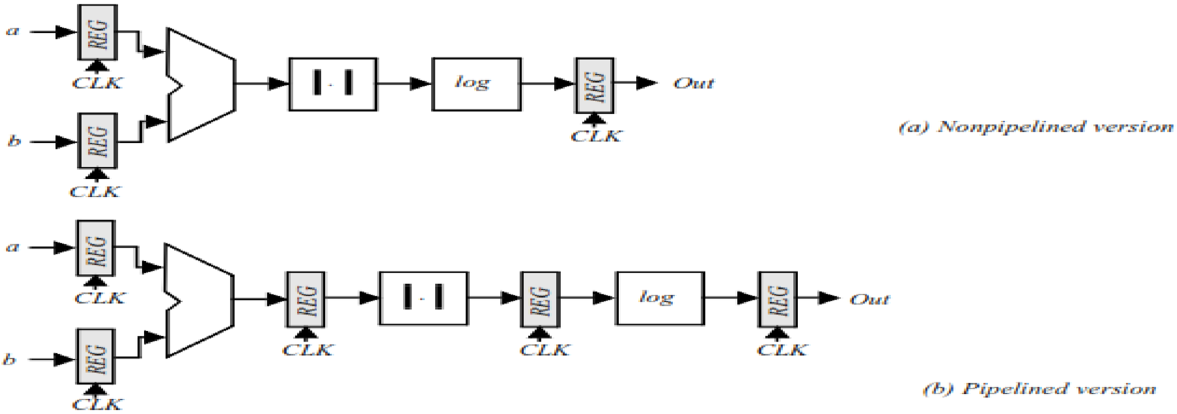
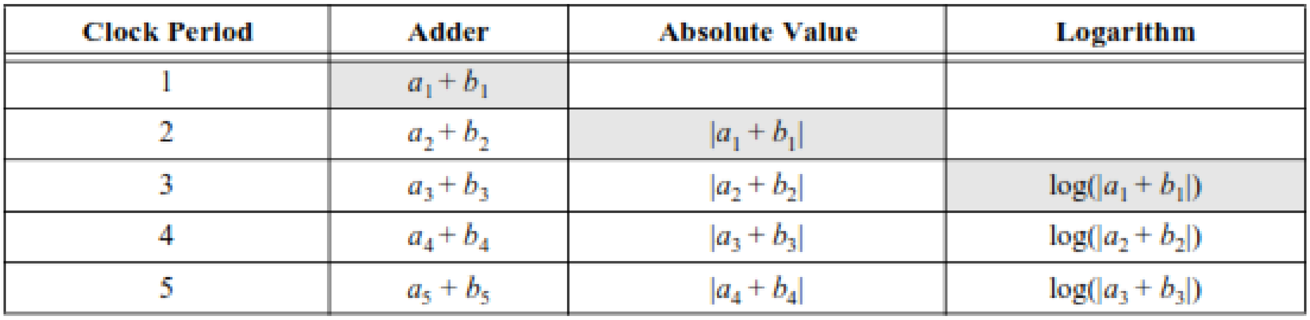


Figure 3.22 Data path for the computation of log(|a + b|).

- At that time, the circuit has already performed parts of the computations for the next data sets, (a2, b2) and (a3, b3).
- The computation is performed in an assembly-line fashion, hence the name pipeline.
- The combinational circuit block has been partitioned into three sections, each of which has a smaller propagation delay than the original function.
- This reduces the value of the minimum allowable clock period:

$$T_{\min,pipe} = t_{c-q} + \max(t_{pd,add}, t_{pd,abs}, t_{pd,\log})$$

- Suppose all logic blocks have same propagation delay and that the register overhead is small with respect to the logic delays.



Clock Period	Adder	Absolute Value	Logarithm
1	$a_1 + b_1$		
2	$a_2 + b_2$	$\|a_1 + b_1\|$	
3	$a_3 + b_3$	$\|a_2 + b_2\|$	$\log(\|a_1 + b_1\|)$
4	$a_4 + b_4$	$\|a_3 + b_3\|$	$\log(\|a_2 + b_2\|)$
5	$a_5 + b_5$	$\|a_4 + b_4\|$	$\log(\|a_3 + b_3\|)$

Table 1: Example of pipelined computations.

- The pipelined network performs the original circuit by a factor of three, under these assumptions $T_{min,pipe}$=T_{min}/3.
- The increased performance comes at the relatively small cost of two additional registers, and an increased latency.
- Pipelining is implemented for very high-performance datapaths.

3.4.1 NORA-CMOS—A Logic Style for Pipelined Structures:

Discuss about the NORA–CMOS structure. (Nov 2016)

- The latch-based pipeline circuit can also be implemented using C^2MOS latches, as shown in Figure 3.24.
- This topology has one additional property:
    - *A C^2MOS-based pipelined circuit is race-free as long as all the logic functions F (implemented using static logic) between the latches are noninverting.*

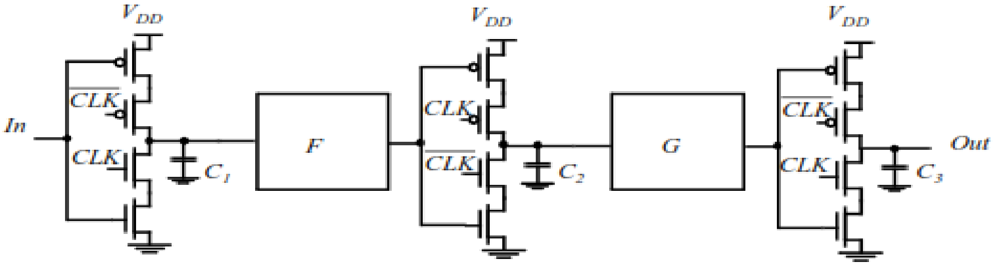


Figure 3.24 Pipelined datapath using C^2MOS latches

- The only way, a signal can race from stage to stage under this condition is, when the logic function F is inverting, as in Figure 3.25.
- Here F is replaced by a single, static CMOS inverter. Similar considerations are valid for the (1-1) overlap.
- It combines C^2MOS pipeline registers and NORA dynamic logic function blocks.
- Each module consists of a block of combinational logic, that can be a mixture of static and dynamic logic, followed by a C^2MOS latch.

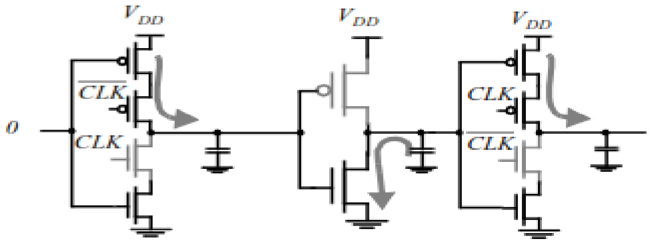
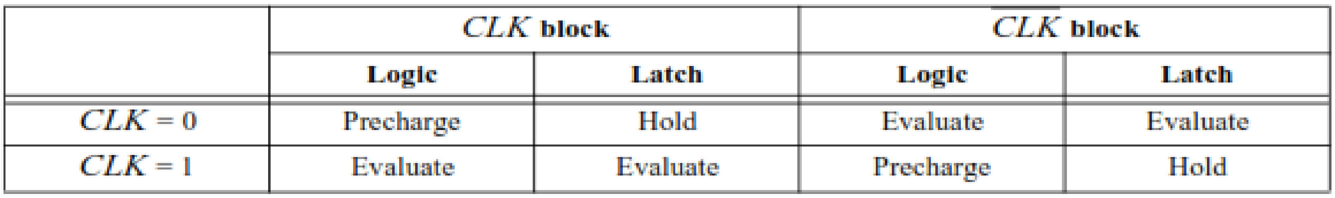


Figure 3.25 Potential race condition during (0-0) overlap in C^2MOS-based design.

- Logic and latch are clocked, in such a way that both are simultaneously in either evaluation or hold (precharge) mode.
- A block that is, in evaluation during CLK = 1 is called a CLK-module, while the inverse is called a $\overline{CLK}$ -module.
- The operation modes of the modules are summarized in Table 2.



	CLK block		$\overline{CLK}$ block	
	Logic	**Latch**	**Logic**	**Latch**
$CLK = 0$	Precharge	Hold	Evaluate	Evaluate
$CLK = 1$	Evaluate	Evaluate	Precharge	Hold

Table 2: Operation modes for NORA logic modules.

3.4.2 Latch- vs. Register-Based Pipelines:

Compare Latch and register based pipelines.

- Pipelined circuits can be constructed using level-sensitive latches instead of edge-triggered registers.
- Consider the pipelined circuit of Figure 3.23.
- The pipeline system is implemented based on pass-transistor-based positive and negative latches instead of edge triggered registers.
- Here logic is introduced between the master and slave latches of a master-slave system.
- When the clocks CLK and $\overline{CLK}$ are non-overlapping, correct pipeline operation is obtained.
- Input data is sampled on C_1 at the negative edge of CLK and the computation of logic block F starts.
- The result of the logic block F is stored on C_2 on the falling edge of $\overline{CLK}$ and the computation of logic block G starts.
- The non-overlapping of the clocks ensures correct operation.
- The value stored on C_2 at the end of the CLK low phase, is the result of passing the previous input through the logic function F.
- When overlap exists between CLK and $\overline{CLK}$, the next input is already being applied to F, and its effect might propagate to C_2 before $\overline{CLK}$ goes low.

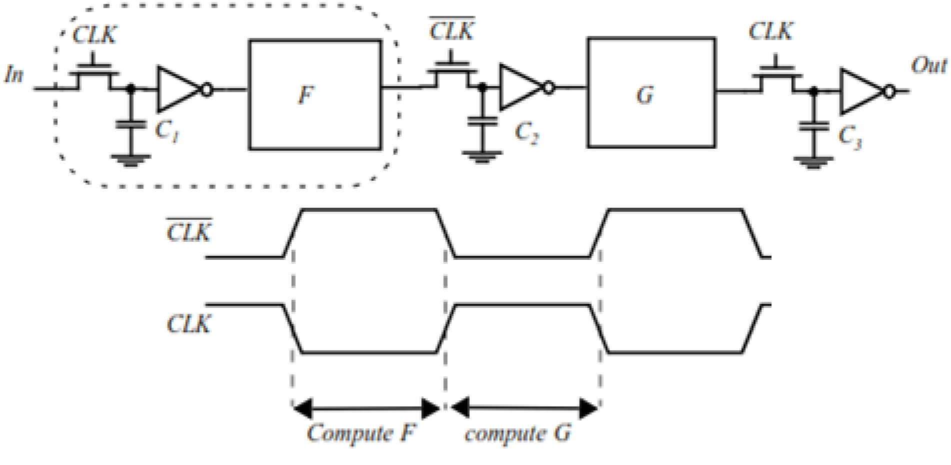


Figure 3.23 Operation of two-phase pipelined circuit using dynamic registers.

****

3.5 Choosing a Clocking Strategy:

Discuss about strategy required for choosing a clock signal.

- Choosing the right clocking scheme affects the functionality, speed and power of a circuit.
- The simple clocking scheme is the two-phase master-slave design.
- The predominant approach is use the multiplexer-based register and to generate the two clock phases locally, by simply inverting the clock.
- High-performance CMOS VLSI design is using simple clocking schemes, even at the expense of performance.

3.5.1 Static sequencing element methodology:-

- Many issues are related to static sequencing element methodology.

(a) Choice of element:-

Flip-flop: Flip-flop has high sequencing overhead. It is simple and easy to understand the operation of flip-flop.

Pulsed latches:-

- Faster than flip-flop.
- Provides some time borrowing option.
- Consumes law power.

Transparent Latch:-

- It has low sequencing overhead compared with flip-flop.
- It allows almost half cycle of time borrowing and it is good choice.

(b)Low power sequential design:-

- Pulsed latches are power efficient.
- Flip-flop consumes more power.
- Clock gating can be used to reduce power.

(c) Two-phase Timing types:-In this type, the signal can belong to phase 1 (or) phase 2. In each phase, 3 different clocks are stable, valid and qualified clock.

Stable clock:-

A signal is stable (in$\varphi 1$), if it settles to a value before rises and remains constant until after, $\varphi 1$falls.

Valid clock: - A signal is valid (in$\varphi 1$), if it settles to a value before $\varphi 1$falls and remains at that value after $\varphi 1$ falls.

Qualified signal: -A signal is said to be in qualified clock in$\varphi 1$, if it either rises and falls like $\varphi 1$(or) remains low for the entire cycle.

****

3.6 Synchronous and Asynchronous circuits-Timing issues:

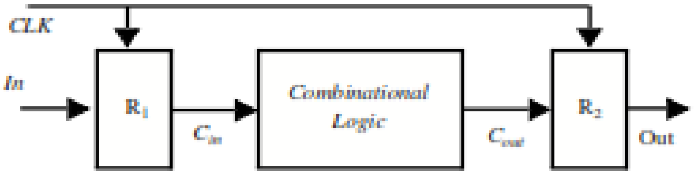
- A synchronous system approach is one, in which all memory elements in the system are simultaneously updated using a globally distributed periodic synchronization signal.
- Functionality is ensured by imposing some strict constraints on the generation of the clock signals and their distribution to the memory elements.
- Analyze the impact of spatial variations of the clock signal, called clock skew and temporal variations of the clock signal, called clock jitter.
- Asynchronous design avoids the problem of clock uncertainty by eliminating the need for globally-distributed clocks.
- The important issues of synchronization, which is required when interfacing different clock domains or when sampling an asynchronous signal.

Classification of Digital Systems:

- In digital systems, signals can be classified depending on, how they are related to a local clock.
- Signals that transition only at predetermined periods in time can be classified as synchronous, mesochronous and plesiochronous with respect to a system clock.
- A signal that can transition at arbitrary times is considered asynchronous.

3.6.1 Synchronous Interconnect

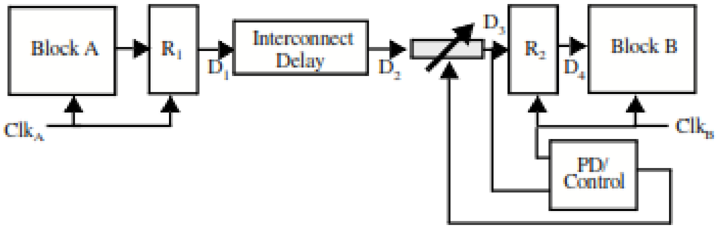
- A synchronous signal has the exact same frequency and a known fixed phase offset with respect to the local clock.
- In such a timing methodology, the signal is “synchronized” with the clock and the data can be sampled directly without any uncertainty.
- In digital logic design, synchronous systems are straight forward type of interconnect, where the flow of data in a circuit proceeds with the system clock as shown below.



3.6.2 Mesochronous interconnect:

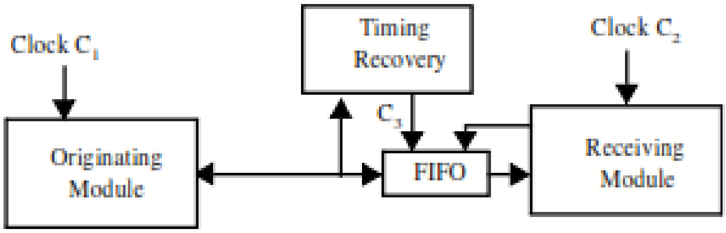
- A mesochronous signal has the same frequency but an unknown phase offset with respect to the local clock (“meso” from Greek is middle).
- For example, if data is being passed between two different clock domains, then the data signal transmitted from the first module can have an unknown phase relationship to the clock of the receiving module.
- In such a system, it is not possible to directly sample the output at the receiving module because of the uncertainty in the phase offset.
- A (mesochronous) synchronizer can be used to synchronize the data signal with the receiving clock as shown below.

- The synchronizer serves to adjust the phase of the received signal to ensure proper sampling.



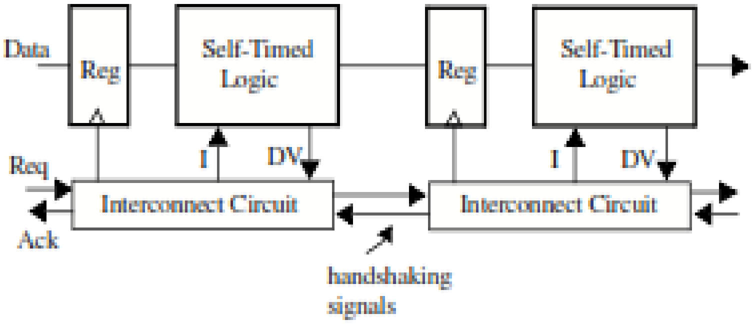
3.6.3 Plesiochronous Interconnect

- A plesiochronous signal has nominally the same, but slightly different frequency as the local clock (“plesio” from Greek is near).
- This scenario can easily arise when two interacting modules have independent clocks generated from separate crystal oscillators.
- Since the transmitted signal can arrive at the receiving module at a different rate than the local clock, one need to utilize a buffering scheme to ensure, all data is received.
- A possible framework for plesiochronous interconnect is shown in Figure below.



3.6.4 Asynchronous Interconnect:

- Asynchronous signals can transition at any arbitrary time and are not slaved to any local clock.
- As a result, it is not to map these arbitrary transitions into a synchronized data stream.
- Asynchronous signals are used to eliminate the use of local clocks and utilize a self-timed asynchronous design approach.
- In this approach, communication between modules is controlled through a handshaking protocol.



****

3.7 Clock-Distribution Techniques

- ❖ **Explain the clock distribution techniques in synchronous design in detail. (Nov 2017)**
- ❖ **Design a clock distribution network based on H tree model for 16 nodes. (April 2018)**

- Clock skew and jitter are major issues in digital circuits and they limit the performance of a digital system.
- It is necessary to design a clock network, that minimizes skew and jitter.
- Another important consideration in clock distribution is the power dissipation.
- In most high-speed digital processors, a majority of the power is dissipated in the clock network.
- To reduce power dissipation, clock networks must support clock conditioning, the ability to shut down parts of the clock network.
- Unfortunately, clock gating results in additional clock uncertainty.

Fabrics for clocking:

- Clock networks include a network that is used to distribute a global reference to various parts of the chip.
- A final stage is responsible for local distribution of the clock, while considering the local load variations.
- Most clock distribution schemes use the absolute delay from a central clock source to the clocking elements.
- Therefore one common approach to distributing a clock is, to use balanced paths (or called trees).
- The most common type of clock primitive is, the H-tree network (named for the physical structure of the network) in figure, where a 4x4 array is shown.
- In this scheme, the clock is routed to a central point on the chip and balanced paths.
- Include both matched interconnect as well as buffers, are used to distribute the reference to various leaf nodes.
- If each path is balanced, the clock skew is zero. It takes multiple clock cycles for a signal to propagate from the central point to each leaf node. The arrival times are equal at every leaf node.
- The H-tree configuration is particularly useful for regular-array networks, in which all elements are identical and the clock can be distributed as a binary tree.

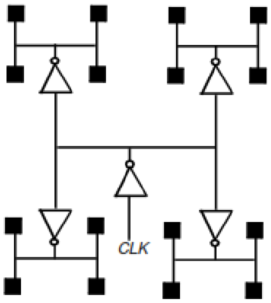


Figure: Example of an H-tree clock-distribution network for 16 leaf nodes.

Latch-Based Clocking:

- The use of a latch based methodology (in Figure) enables more flexible timing, allowing one stage to pass slack to or steal time from following stages.
- This flexibility allows an overall performance increase.
- In this configuration, a stable input is available to the combinational logic block A (CLB_A) on the falling edge of CLK1 (at edge2).
- On the falling edge of CLK2 (at edge3), the output CLB_A is latched and the computation of CLK_B is launched.
- CLB_B computes on the low phase of CLK2 and the output is available on the falling edge of CLK1 (at edge4).
- This timing appears, equivalent to having an edge-triggered system where CLB_A and CLB_B are cascaded and between two edge-triggered registers.
- In both cases, it appears that the time available to perform the combination of CLB_A and CLB_B are T_{CLK}.

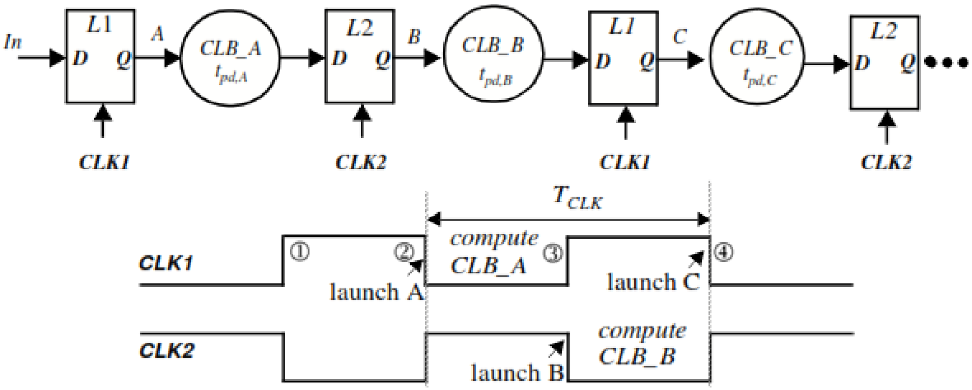


Figure: Latch-based design in which transparent latches are separated by combinational logic.

****

3.8 Self-Timed Circuit Design

Discuss about asynchronous design in logic design.

3.8.1 Self-Timed Logic : An Asynchronous Technique

- A more reliable and robust technique is the self-timed approach, which presents a local solution to the timing problem.
- Figure uses a pipelined datapath to illustrate how this can be accomplished.
- The computation of a logic block is initiated by asserting a Start signal.
- The combinational logic block computes on the input data.
- This signaling ensures the logical ordering of the events and can be achieved with the aid of an extra Ack(nowledge) and Req(uest) signal.

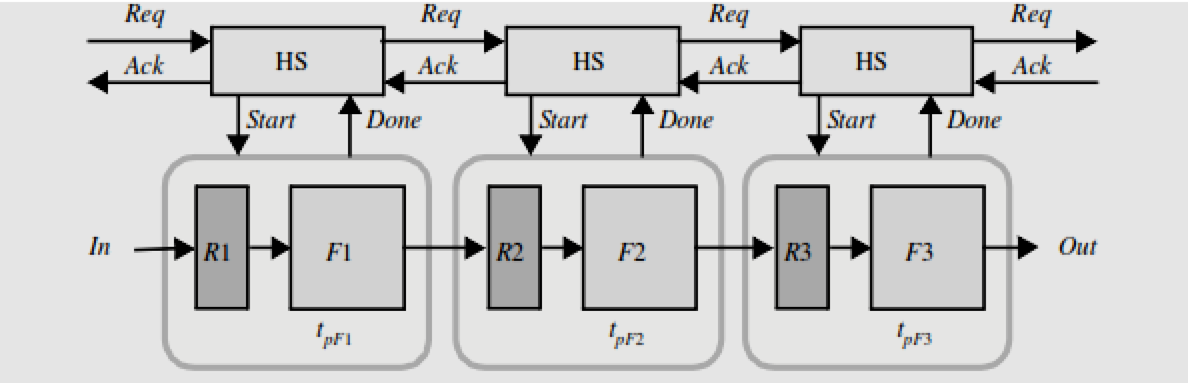


Figure: Self-timed, pipelined datapath.

In the case of the pipelined datapath, the scenario could proceed as follows.

1. An input word arrives, and a Req(uest) to the block F1 is raised. If F1 is inactive at that time, it transfers the data and acknowledges this fact to the input buffer.
2. F1 is enabled by raising the Start signal. After a certain amount of time, dependent upon the data values, the Done signal goes high indicating the completion of the computation.
3. A Req(uest) is issued to the F2 module. If this function is free, an Ack(nowledge) is raised, the output value is transferred and F1 can go ahead with its next computation.

3.8.2 A simple synchronizer

How do eliminates metastability problem in sequential circuit and explain?

- A synchronizer accepts an input D and a clock ф. It produces an output Q that should be valid for some bounded delay after the clock.
- The synchronizer has an aperture, defined by a setup and hold time around the rising edge of the clock.

- If the data is stable during the aperture, Q should equal D. If the data changes during the aperture, Q can be chosen arbitrarily.

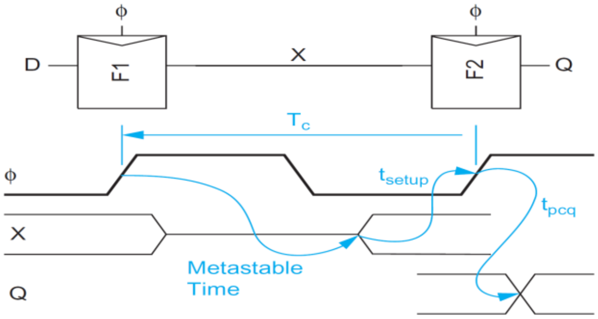


Figure: Simple Synchronizer

- Figure shows a simple synchronizer built from a pair of flip-flops. F1 samples the asynchronous input D.
- The output X may be metastable for some time, but will settle to a good level with high probability, if we wait long enough.
- F2 samples X and produce an output Q, that should be a valid logic level and be aligned with the clock.
- The synchronizer has a latency of one clock cycle Tc .

3.8.3 Communicating between asynchronous clock domains

- A common application of synchronizers is in communication between asynchronous clock domains, i.e., blocks of circuits that do not share a common clock.
- Suppose System A is controlled by clkA that needs to transmit N-bit data words to System B, which is controlled by clkB, as shown in Figure.
- The systems can represent separate chips or separate units within a chip using unrelated clocks.
- System A must guarantee that the data is stable, while the flip-flops in System B sample the word.
- It indicates when new data is valid by using a request signal (Req), so System B receives the word exactly once rather than zero or multiple times.
- System B replies with an acknowledge signal (Ack), when it has sampled the data, so System A knows when the data can safely be changed.
- If the relationship between clkA and clkB is completely unknown, a synchronizer is required at the interface.

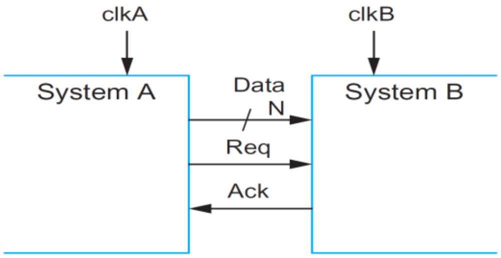


Figure: Communication between asynchronous systems

3.8.4 Arbiter

- The arbiter of Figure (a) is related to the synchronizer. It determines which of two inputs arrived first.
- If the spacing between the inputs exceeds some aperture time, the first input should be acknowledged.
- If the spacing is smaller, exactly one of the two inputs should be acknowledged, but the choice is arbitrary.
- For example, in a television game show, two contestants may hit buttons to answer a question.
- If one presses the button first, should be acknowledged. If both presses the button at times too close to distinguishes, the host may choose one of the two contestants arbitrarily.

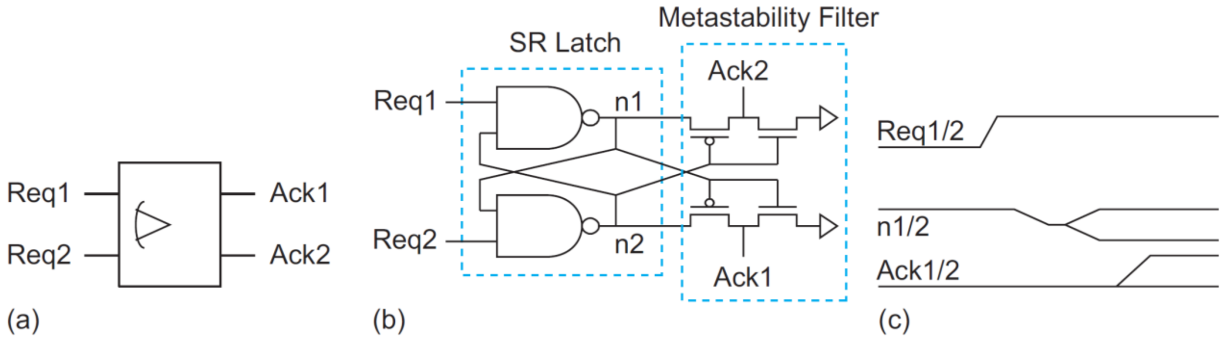


Figure: Arbiter

- Figure (b) shows an arbiter built from an SR latch and a four-transistor metastability filter.
- If one of the request inputs arrives well before the other, the latch will respond appropriately.
- If they arrive at nearly the same time, the latch may be driven into metastability, as shown in Figure (c).
- The filter keeps both acknowledge signals low, until the voltage difference between the internal nodes n1 and n2 exceeds Vt , indicating that a decision has been made.
- Such an asynchronous arbiter will never produce metastable outputs.

3.8.5 Synchronous versus Asynchronous Design:

Compare synchronous and asynchronous design.

- The self-timed approach offers a potential solution to the growing clock-distribution problem.
- It translates the global clock signal into a number of local synchronization problems.

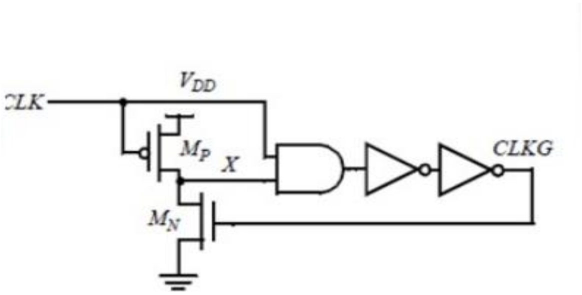
- Handshaking logic is needed to ensure the logical ordering of the circuit events and to avoid race conditions.
- In general, synchronous logic is both faster and simpler since the overhead of completion-signal generation and handshaking logic is avoided.
- Skew management requires extensive modeling and analysis, as well as careful design.
- It will not be easy to extend this methodology into the next generation of designs.
- This observation is already reflected in the fact that the routing network for the latest generation of massively parallel supercomputers is completely implemented using self-timing.
- For self timing to become a mainstream design technique however (if it ever will), further innovations in circuit and signaling techniques and design methodologies are needed.

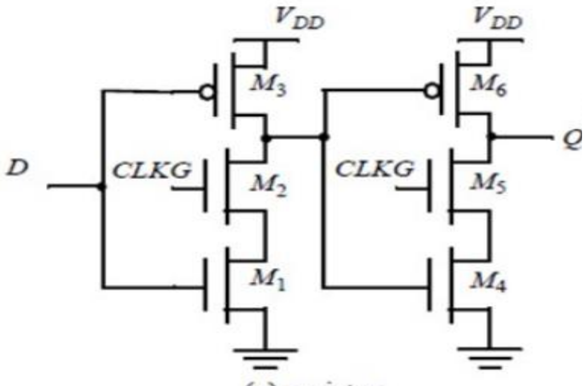
****

3.9 Pulse Registers

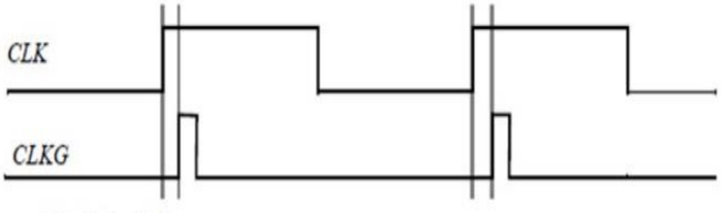
Design the pulse registers suitable for sequential CMOS circuits. [May 2021][Nov/Dec 2022]

- A fundamentally different approach for constructing a register uses pulse signals.
- The idea is to construct a short pulse around the rising (or falling) edge of the clock.
- This pulse acts as the clock input to a latch, sampling the input only in a short window.
- Race conditions are thus avoided by keeping the opening time (i.e, the transparent period) of the latch very short.
- The combination of the glitch generation circuitry and the latch results in a positive edge-triggered register.





(a) register



(c) glitch clock

- **Figure b shows** an example circuit for constructing a short intentional glitch on each rising edge of the clock.
- When CLK = 0, node X is charged up to VDD (MN is off since CLKG is low).
- On the rising edge of the clock, there is a short period of time when both inputs of the AND gate are high, causing CLKG to go high.

- This in turn activates M_N, pulling X and eventually CLKG low.
- The length of the pulse is controlled by the delay of the AND gate and the two inverters.
- Note that there exists also a delay between the rising edges of the input clock (CLK) and the glitch clock (CLKG) — also equal to the delay of the AND gate and the two inverters.
- If every register on the chip uses the same clock generation mechanism, this sampling delay does not matter.
- However, process variations and load variations may cause the delays through the glitch clock circuitry to be different.
- This must be taken into account when performing timing verification and clock skew analysis.

Waveform

- If set-up time and hold time are measured in reference to the rising edge of the glitch clock, the set-up time is essentially zero, the hold time is equal to the length of the pulse (if the contamination delay is zero for the gates), and the propagation delay (tc-q) equals two gate delays.

Advantage

- The reduced clock load and the small number of transistors required.
- The glitch-generation circuitry can be amortized over multiple register bits.

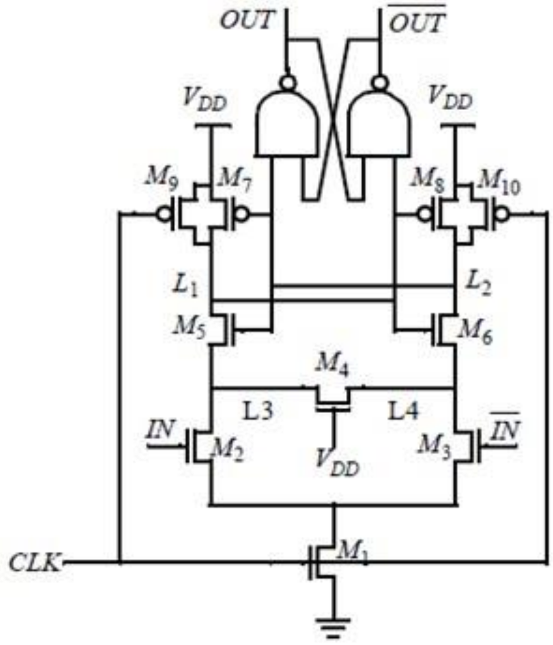
Disadvantage

- A substantial increase in verification complexity.
- This has prevented a wide-spread use.

3.10 Sense-Amplifier Based Registers

Write short notes on Sense – Amplifier Based Registers. [Nov/Dec 2022][April/May 2023]

- A sense amplifier structure to implement an edge- triggered register.
- Sense amplifier circuits accept small input signals and amplify them to generate rail-to-rail swings.
- There are many techniques to construct these amplifiers, with the use of feedback (e.g., cross-coupled inverters).



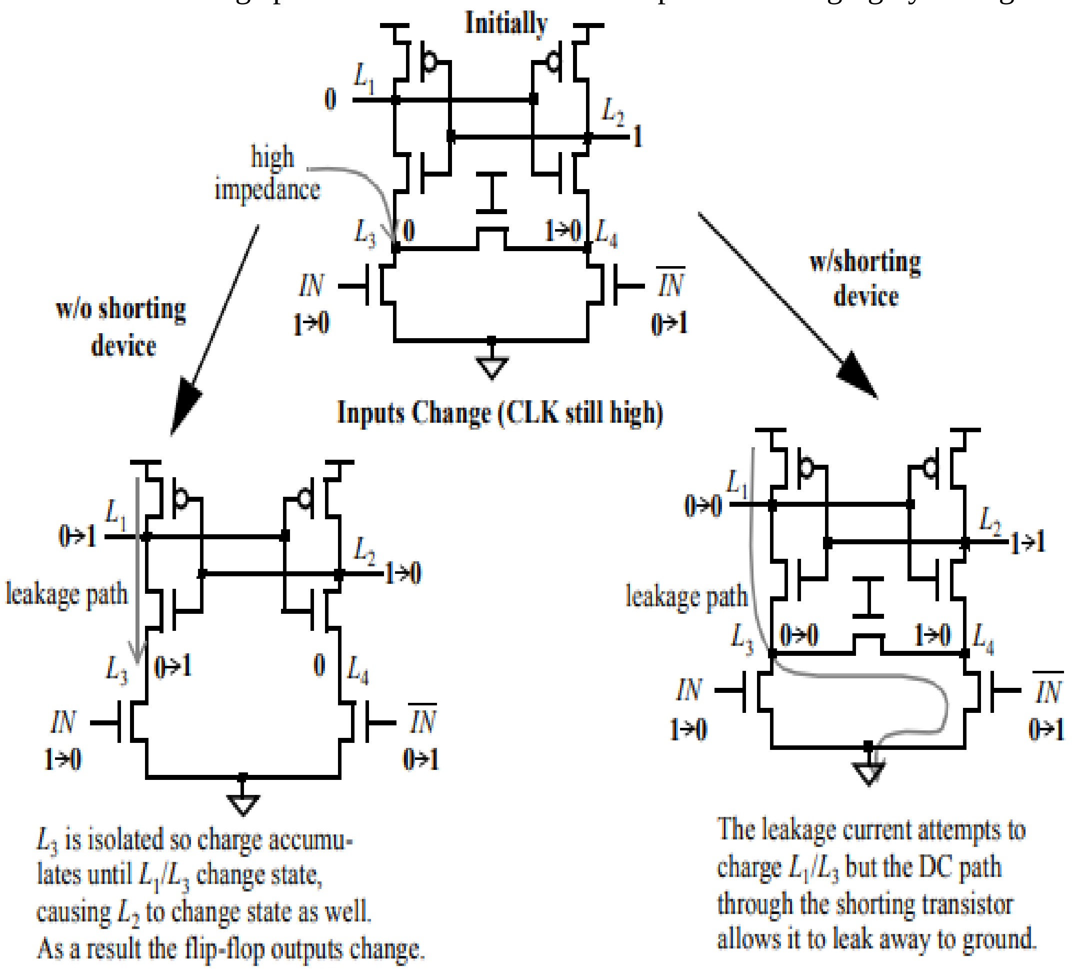
Positive edge-triggered register based on sense-amplifier

- The circuit uses a precharged front-end amplifier that samples the differential input signal on the rising edge of the clock signal.
- The outputs of front-end are fed into a NAND cross- coupled SR FF that holds the data and guarantees that the differential outputs switch only once per clock cycle.
- The differential inputs in this implementation don’t have to have rail-to-rail swing and hence this register can be used as a receiver for a reduced swing differential bus.

Operation

- The core of the front-end consists of a cross-coupled inverter (M5-M8) whose outputs (L1 and L2) are precharged using devices M9 and M10 during the low phase of the clock.
- As a result, PMOS transistors M7 and M8 to be turned off and the NAND FF is holding its previous state.
- Transistor M1 is similar to an evaluate switch in dynamic circuits and is turned off ensuring that the differential inputs don’t affect the output during the low phase of the clock.
- On the rising edge of the clock, the evaluate transistor turns on and the differential input pair (M2 and M3) is enabled, and the difference between the input signals is amplified on the output nodes on L1 and L2.
- The cross-coupled inverter pair flips to one of its the stable states based on the value of the inputs.
- For example, if IN is 1, L1 is pulled to 0, and L2 remains at VDD. Due to the amplifying properties of the input stage, it is not necessary for the input to swing all the way up to VDD and enables the use of low swing signaling on the input wires.

- The shorting transistor, M4, is used to provide a DC leakage path from either node L3, or L4, to ground.
- This is necessary to accommodate the case where the inputs change their value after the positive edge of CLK has occurred, resulting in either L3 or L4 being left in a high-impedance state with a logical low voltage level stored on the node.
- Without the leakage path that node would be susceptible to charging by leakage currents.



Initially

Inputs Change (CLK still high)

L_3 is isolated so charge accumulates until L_1/L_3 change state, causing L_2 to change state as well. As a result the flip-flop outputs change.

The leakage current attempts to charge L_1/L_3 but the DC path through the shorting transistor allows it to leak away to ground.

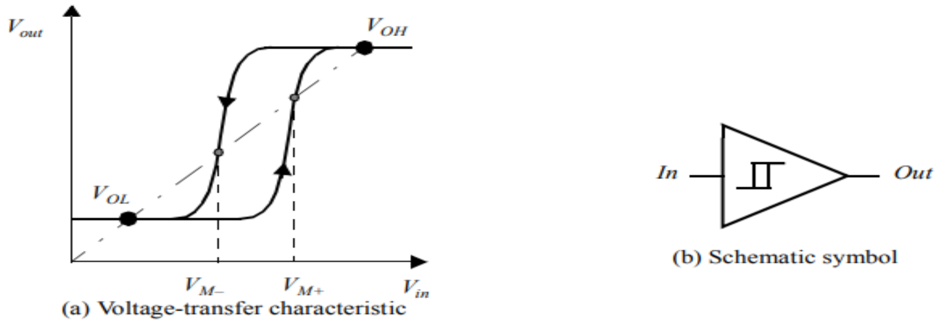
3.11 Schmitt Trigger

Explain the circuit and working of CMOS implementation of Schmitt Trigger. (NOV 2021) [Nov/Dec 2022]

- A Schmitt trigger is a device with two important properties:

  1. It responds to a slowly changing input waveform with a fast transition time at the output.
  2. The voltage-transfer characteristic of the device displays different switching thresholds for positive- and negative-going input signals.

A typical voltage-transfer characteristic of the Schmitt trigger is shown (and its schematics symbol). The switching thresholds for the low-to-high and high to-low transitions are called V_{M+} and V_{M-}, respectively. The hysteresis voltage is defined as the difference between the two.

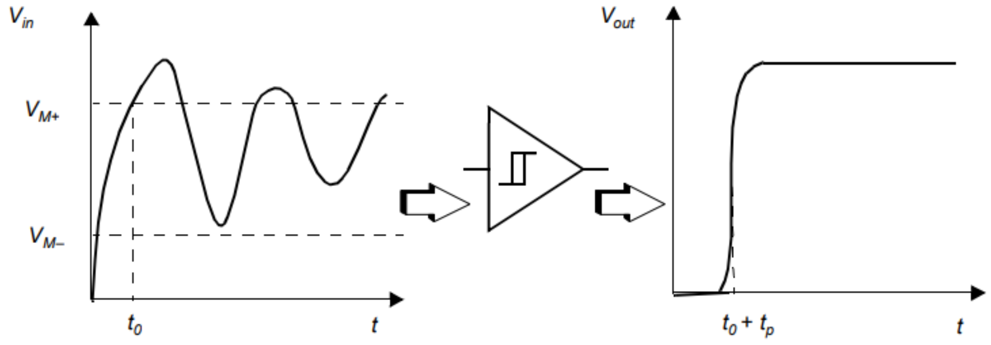


(a) Voltage-transfer characteristic

(b) Schematic symbol

Fig: Non-inverting Schmitt trigger.

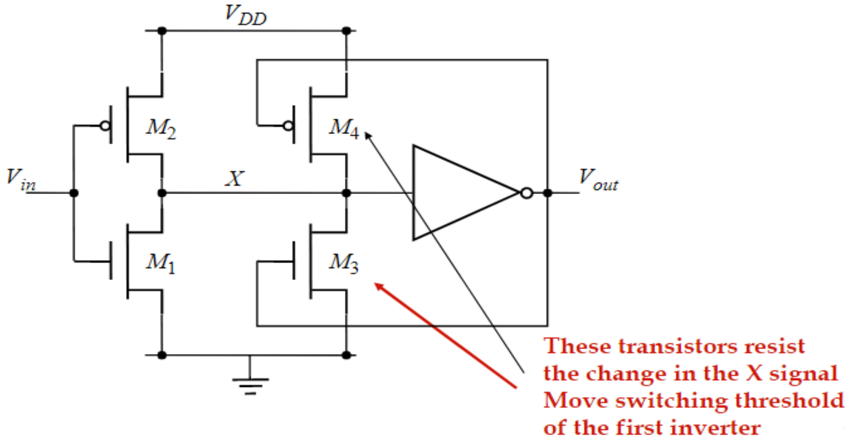
- One of the main uses of the Schmitt trigger is to turn a noisy or slowly varying input signal into a clean digital output signal.
- Notice how the hysteresis suppresses the ringing on the signal.
- At the same time, the fast low-to-high (and high-to-low) transitions of the output signal should be observed.
- For instance, steep signal slopes are beneficial in reducing power consumption by suppressing direct-path currents.
- The "secret" behind the Schmitt trigger concept is the use of positive feedback.



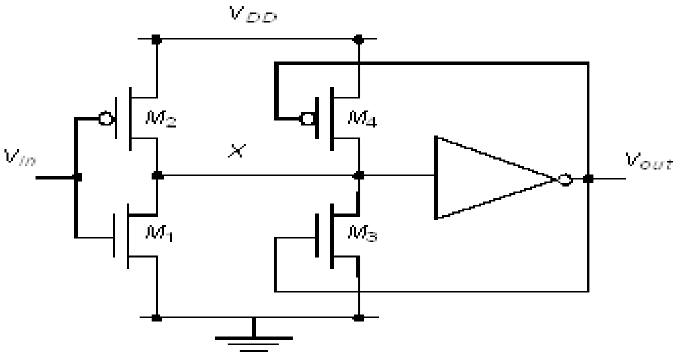
Noise suppression using a Schmitt trigger.

CMOS Implementation

- Increasing k_n/k_p ratio decreases the logical switching threshold
- If V_{in} =0 the V_{out} (connected to M_4) is also zero So effectively the input is connected to M_2 and M_4 in parallel This increases k_p and the switching threshold



- If V in =0 the situation is reversed and k_n increases reducing the switching threshold



- Suppose that V_{in} is initially equal to 0, so that V_{out} = 0 as well.
- The feedback loop biases the PMOS transistor M_4 in the conductive mode while M_3 is off.
- The input signal effectively connects to an inverter consisting of two PMOS transistors in parallel (M_2 and M_4) as a pull-up network, and a single NMOS transistor (M_1) in the pull-down chain.
- This modifies the effective transistor ratio of the inverter to $k_{M1}/(k_{M2}+k_{M4})$, which moves the switching threshold upwards.
- Once the inverter switches, the feedback loop turns off M_4, and the NMOS device M_3 is activated.
- This extra pull-down device speeds up the transition and produces a clean output signal with steep slopes.
- A similar behavior can be observed for the high-to-low transition.
- In this case, the pull-down network originally consists of M_1 and M_3 in parallel, while the pull-up network is formed by M_2.
- This reduces the value of the switching threshold to V_{M-}.

3.12 Multivibrator Circuits

Explain about Monostable and Astable Sequential Circuits. [Apr/May 2022] [Nov/Dec 2022] Sketch and explain the Monostable sequential circuits based on CMOS logic. (May 2021) [April / May 2023]

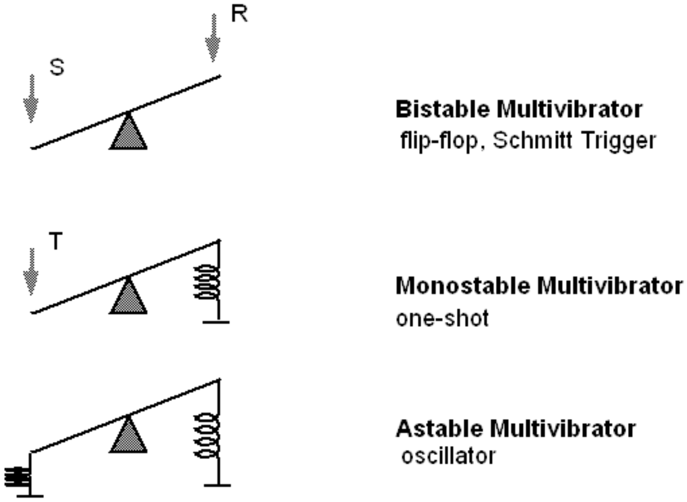
Sequential circuits are of three types −

Bistable − Bistable circuits have two stable operating points and will be in either of the states. Example − Memory cells, latches, flip-flops and registers.

Monostable − Monostable circuits have only one stable operating point and even if they are temporarily perturbed to the opposite state, they will return in time to their stable operating point. Example: Timers, pulse generators.

Astable − circuits have no stable operating point and oscillate between several states.

Example − Ring oscillator.



Transition-Triggered Monostable Sequential Circuits

- A monostable element is a circuit that generates a pulse of a predetermined width every time the quiescent circuit is triggered by a pulse or transition event.
- It is called monostable because it has only one stable state (the quiescent one).
- A trigger event, which is either a signal transition or a pulse, causes the circuit to go temporarily into another quasi-stable state.
- This means that it eventually returns to its original state after a time period determined by the circuit parameters.
- This circuit, also called a one-shot, is useful in generating pulses of a known length.

- This functionality is required in a wide range of applications.
- We have already seen the use of a one-shot in the construction of glitch registers.
- Another notorious example is the address transition detection (ATD) circuit, used for the timing generation in static memories.
- This circuit detects a change in a signal, or group of signals, such as the address or data bus, and produces a pulse to initialize the subsequent circuitry.
- The most common approach to the implementation of one-shots is the use of a simple delay element to control the duration of the pulse.
- In the quiescent state, both inputs to the XOR are identical, and the output is low.
- A transition on the input causes the XOR inputs to differ temporarily and the output to go high.
- After a delay t $_d$ (of the delay element), this disruption is removed, and the output goes low again.
- A pulse of length td is created.
- The delay circuit can be realized in many different ways, such as an RC-network or a chain of basic gates.

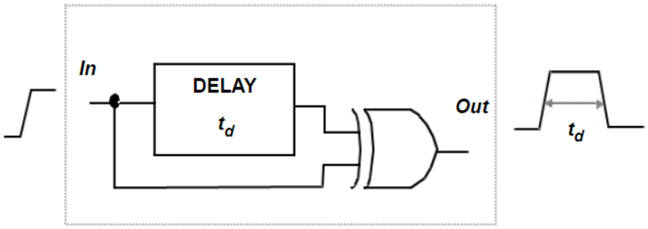


Fig: Transition-triggered one-shot.

3.13 Astable Sequential Circuits

- An astable circuit has no stable states.
- The output oscillates back and forth between two quasi-stable states with a period determined by the circuit topology and parameters (delay, power supply, etc.).
- One of the main applications of oscillators is the on-chip generation of clock signals.
- This application is discussed in detail in a later chapter (on timing). The ring oscillator is a simple, example of an astable circuit.
- It consists of an odd number of inverters connected in a circular chain.
- Due to the odd number of inversions, no stable operation point exists, and the circuit oscillates with a period equal to 2 × tp × N, with N the number of inverters in the chain and tp the propagation delay of each inverter.
- The ring oscillator composed of cascaded inverters produces a waveform with a fixed oscillating frequency determined by the delay of an inverter in the CMOS process.

- In many applications, it is necessary to control the frequency of the oscillator.
- An example of such a circuit is the voltage-controlled oscillator (VCO), whose oscillation frequency is a function (typically non-linear) of a control voltage.
- The standard ring oscillator can be modified into a VCO by replacing the standard inverter with a current-starved inverter as shown in Figure.
- The mechanism for controlling the delay of each inverter is to limit the current available to discharge the load capacitance of the gate.
- In this modified inverter circuit, the maximal discharge current of the inverter is limited by adding an extra series device.
- Note that the low-to-high transition on the inverter can also be controlled by adding a PMOS device in series with M2.
- The added NMOS transistor M3, is controlled by an analog control voltage Vcntl, which determines the available discharge current.
- Lowering Vcntl reduces the discharge current and, hence, increases tpHL.
- The ability to alter the propagation delay per stage allows us to control the frequency of the ring structure.
- The control voltage is generally set using feedback techniques.
- Under low operating current levels, the current-starved inverter suffers from slow fall times at its output. This can result in significant short-circuit current.
- This is resolved by feeding its output into a CMOS inverter or better yet a Schmitt trigger.
- An extra inverter is needed at the end to ensure that the structure oscillates.

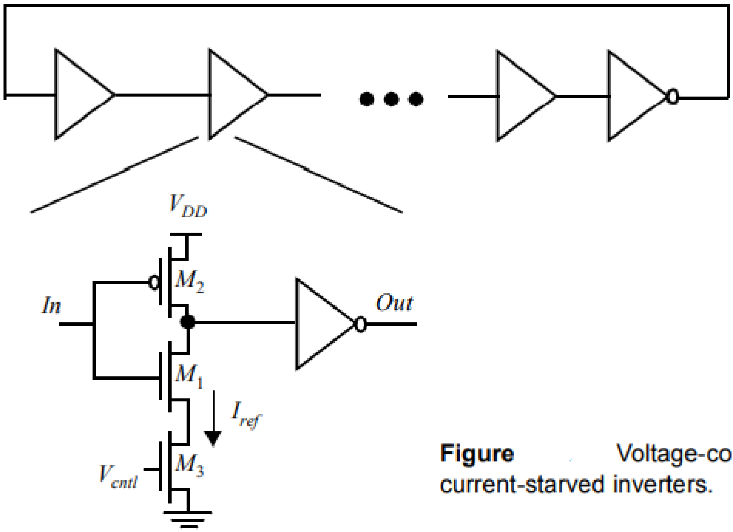
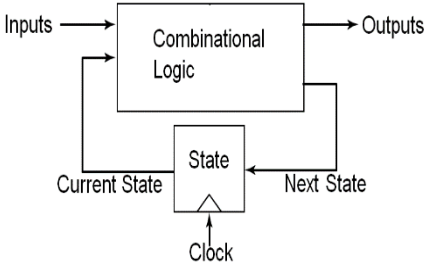


Figure Voltage-controlled oscillator based on current-starved inverters.

UNIT – III
SEQUENTIAL LOGIC CIRCUITS AND CLOCKING STRATEGIES

1. What is a Sequential circuit?

In sequential circuits, the output depends on previous as well as current inputs.



2. What are sequencing methods available in sequential circuit design? (NOV/DEC-2012)

The sequencing methods available in sequential circuit design are,

i. Flipflop
ii. Two phase transparent latches
iii. Pulsed latches

3. What is meant by maximum delay or setup time failure and how to avoid?

If the combinational logic delay is too high, the receiving element will miss its setup time and sample the wrong value. This is called a *setup time failure* or *max-delay failure*. Max-delay can be solved by redesigning the logic to be faster or by increasing the clock period.

4. Define sequencing overhead.

Sequencing overhead is defined as an additional delay to Tokens (Data) that are already critical, decreasing the performance of the system. This extra delay is called *sequencing overhead*.

5. What is meant by Min-delay failure and how to avoid?

If the hold time is large and the contamination delay is small, data can incorrectly propagate through two successive elements on one clock edge, corrupting the state of the system. This is called a *race condition*, *hold-time failure*, or *min-delay failure*.

Min-delay can only be fixed by redesigning the logic, not by slowing the clock.

6. Define time borrowing.

Time borrowing is defined as if one half-cycle or stage of a pipeline has too much logics, it can borrow time into the next half-cycle or stage.

Time borrowing can accumulate across multiple cycles.

7. Define clock skew. (April 2018, April 2019, NOV 2021)[Apr/May 2022]

Clocks have some uncertainty in their arrival times that can cut into the time available for useful computation.

8. How to design CMOS flip-flop?

Dynamic inverting flip-flop built from a pair of back-to-back dynamic latches.

9. How to design Semidynamic Flip-flop?

Semidynamic flip-flop (SDFF) is a cross coupled between a pulsed latch and a flip-flop.

10. What is meant by semidynamic flip-flop(SDFF)?

The SDFF accepts rising inputs slightly after the rising clock edge. Like a flip-flop, falling inputs must set up before the rising clock edge called as *semidynamic*. It combines the dynamic input stage with static operation.

11. What is meant by Differential flip-flop?

Differential flip-flops accept true and complementary inputs and produce true and complementary outputs. They are built from a clocked sense amplifier so that they can respond to small differential input voltages.

12. What is meant by True Single Phase Clock (TSPC) Latch or flip-flop?

True Single Phase clock Latch or flipflop avoids complementary clock pulse.

13. What are sequencing dynamic circuits?

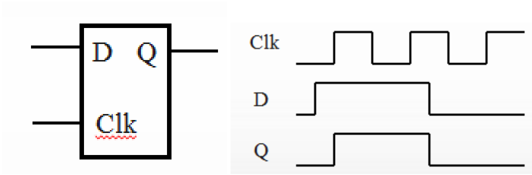
Sequencing dynamic circuits are
i. Traditional domino circuit
ii.Skew tolerant domino circuit

14. Define Synchronizer. (MAY/JUE-2014)

Synchronizer is defined as a circuit that accepts an input that can change at arbitrary times and produces an output aligned to the synchronizer's clock.

15. What is a Latch? (NOV/DEC-2014)

Latch is a bistable device. i.e., it has two stable states (0 and 1). It is the level triggering method.



16. What is a flip-flop? (NOV/DEC-2014)

Flip-flop is a bistable device. i.e., it has two stable states (0 and 1). It is the edge triggering method.

17. What is meant by Bistability and metastability? (NOV 2021) [Apr/May 2022]

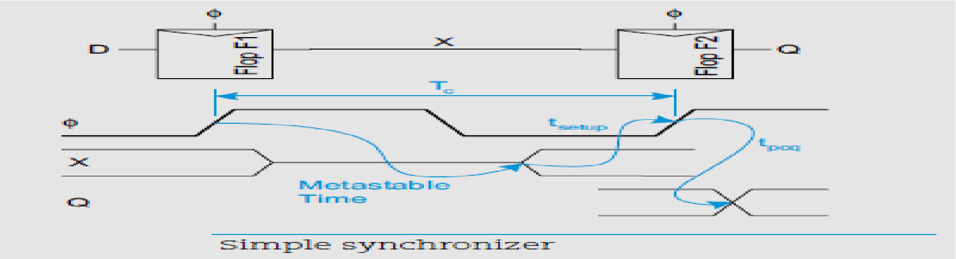
A latch is a bistable device. i.e., it has two stable states (0 and 1). Latch can enter a metastable state in which the output is at an indeterminate level between 0 and 1.

18. Define aperture.

Aperture is defined as a setup and hold time around the rising edge of the clock.

19. How to design simple synchronizer circuit.

Simple synchronizer built from a pair of flip-flops. *F*1 samples the asynchronous input *D*. The output *X* may be metastable for some time, but will settle to a good level with high probability.



Simple synchronizer

20. What is meant by Arbiter?

The *arbiter* is closely related to the synchronizer. It determines which of two inputs arrived first.

If the spacing between the inputs exceeds some aperture time, the first input should be acknowledged.

If the spacing is smaller, exactly one of the two inputs should be acknowledged, but the choice is arbitrary.

21. What are the advantages of differential Flip flop? (Nov 2011)

The advantages of differential Flip flops are

a. Reduce the parasitic delay of the pull down networks.
b. Lower electric fields across the pull down networks.
c. It reduces the channel length of the transistors.

22. State the reasons for the speed advantages of CVSL family. (Nov 2012, Nov 2014)

CVSL has a potential speed advantage because all of the logic is performed with nMOS transistors, thus reducing the input capacitance.

23. Enumerate the features of synchronizers. (May 2013)

The features of synchronizers are,

a. A good synchronizer should have a feedback loop with high gain bandwidth product.
b. It can produce metastable output.

24. What are the disadvantages of using a pseudo nMOS gate instead of a full CMOS gate? (May 2012)

Ratioed circuits dissipate power continually in certain states and have poor noise margin than complementary circuits. Ratioed circuits used in situations where smaller area is needed.

25. What is Klass-semidynamic flip-flop?

Klass-semidynamic is a single-input single-output positive edge-triggered flip-flop. It is domino-style from end allows for efficient embedded combinational logic and reduces the load on the data network.

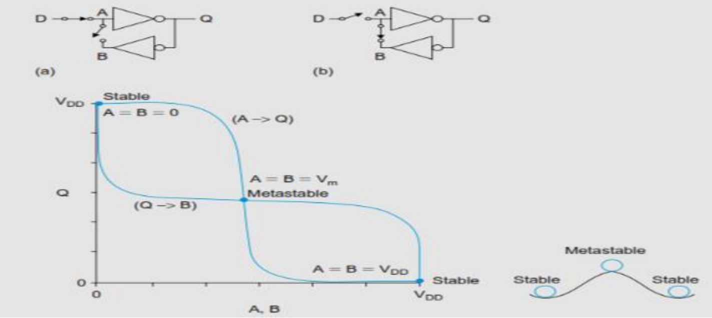
26. What are the different phases of VLSI design flow?

The different phases of VLSI design flows are**,**

- Function Verification and testing, logic synthesis/Timing verification
- Logical verification and testing

- Floor planning automatic place and route
- Layout verification
- Implementation

27. Draw the circuit diagram of a CMOS bistable element and its time domain behavior. (APRIL/MAY-2011)



28. What is CMOS clocked SR flipflop?

CMOS Clocked SR flip-flop consists of a cross-coupled inverter pair, plus 4 extra transistors to drive the flip-flop from one state to another and to provide clocked operation.

29. What do mean by multiplexer based latches?

Multiplexer based latches can provide similar functionality to the SR latch, but has the important added advantage that the sizing of devices only affects performance and is not critical to the functionality.

30. What is Master-Slave Based Edge Triggered Register?

The register consists of cascading a negative latch (master stage) with a positive latch (slave stage).

31. What is the advantage of multiplexer based latch?

The advantage of the multiplexer-based register is the feedback loop is open during the sampling period, and sizing of devices is not critical to functionality.

32. What is pseudostatic?

The register employs a combination of static and dynamic storage approaches depending upon the state of the clock.

33. What is called Clocked CMOS Register? (May 2016)

C^2MOS is a positive edge-triggered register based on the master-slave concept which is insensitive to clock overlap. The register operates in two phases.

34. What is meant by Dual-edge Triggered Register? Give it advantage.

Dual-edge triggered register is a design of sequential circuits that sample the input on both edges. The advantage of this scheme is that a lower frequency clock.

35. What is True Single-Phase Clocked Register (TSPCR)?

The True Single-Phase Clocked Register (TSPCR) uses a single clock (without an inverse clock).

36. What is the advantage and disadvantage of True Single-phase clocked register?

The main advantage is the use of a single clock phase. The disadvantage is the increase the number of transistors. 12 transistors are required.

37. What is pipelining? (Dec. 2016, April 2017)

Pipelining is a popular design technique used to accelerate the operation of the datapaths in digital processors.

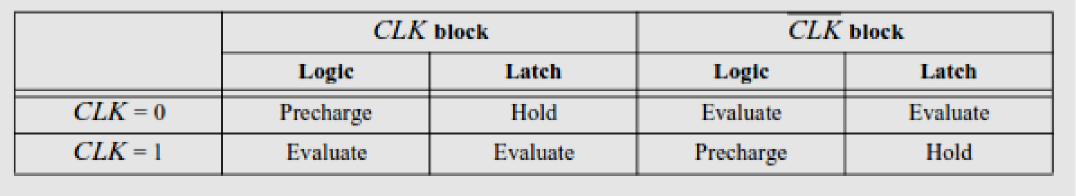
38. What is necessary of non-overlapping clocks?

The non-overlapping of the clocks ensures correct operation. When CLK and $\overline{CLK}$ signals are non-overlapping, correct pipeline operation is obtained.

39. What is topology for NORA-CMOS?(Nov 2017)

The latch-based pipeline circuit implemented using C^2MOS latch is known as NORA-CMOS circuit. A NORA CMOS circuit is race-free as long as all the logic functions between the latches are non-inverting.

40. Tabulate the operation modules of NORA-CMOS circuit.



	CLK block		$\overline{CLK}$ block	
	Logic	**Latch**	**Logic**	**Latch**
$CLK = 0$	Precharge	Hold	Evaluate	Evaluate
$CLK = 1$	Evaluate	Evaluate	Precharge	Hold

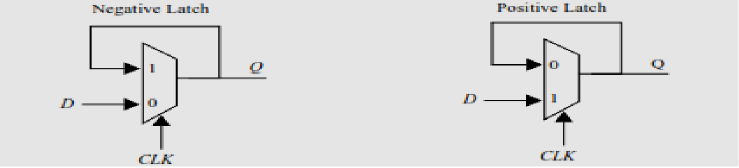
41. Define the dynamic-logic rule.

Inputs to a dynamic CLKn (CLKp) block are only allowed to make a single 0→1 (1 → 0) transition during the evaluation period.

42. Define C^2MOS design rule.

C^2MOS design rule is defining to avoid races, the number of static inversions between C^2MOS latches should be even.

43. Draw the switch level schematic of multiplexer. (May 2016)



.

44. What is known as H-tree clock distribution?

The H-tree configuration is useful for regular-array networks in which all elements are identical and the clock can be distributed as a binary tree. For example, arrays of identical tiled processors.

45. Define clock jitter. (Nov 2017)[Apr/May 2022]

Clock jitter refers to the temporal variation of the clock period at a given point. i.e, the clock period can reduce or expand on a cycle-by-cycle basis.

46. What is meant by Latch based clocking?

Use of registers in sequential circuits enables a robust design methodology. Advantage is combinational logic is separated by transparent latches.

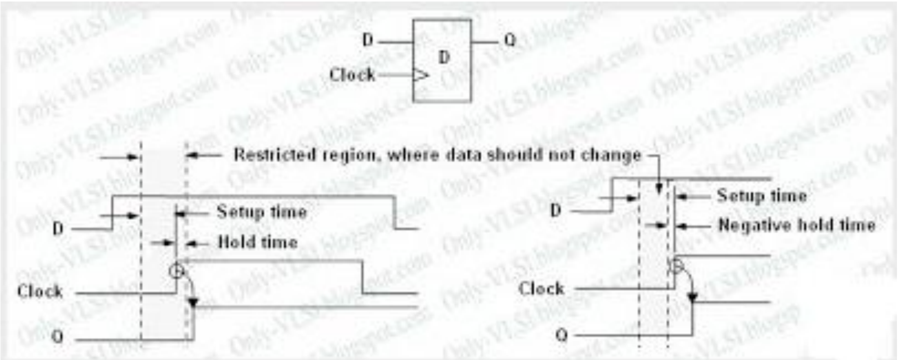
47. Compare synchronous and asynchronous design. (April 2017)

Synchronous design	Asynchronous design
Synchronous logic is both faster and simpler.	Asynchronous logic is slow and complex.
Distributing a clock at high speed becomes exceedingly difficult.	Distributing a clock at high speed becomes easy.

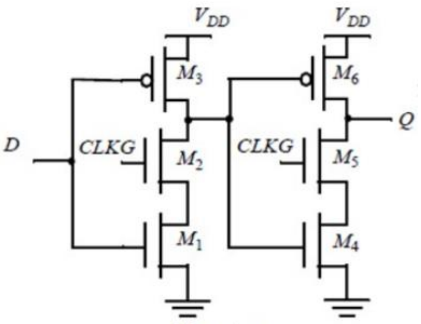
48. Define Set up time and hold time. (Nov 2019)

The **setup time** is the interval before the clock where the data must be held stable.

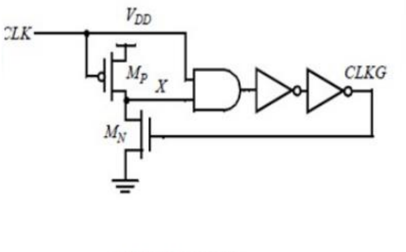
The **hold time** is the interval after the clock where the data must be held stable. Hold time can be negative, which means the data can change slightly before the clock edge and still be properly captured. Most of the current day flip-flops has zero or negative hold time.



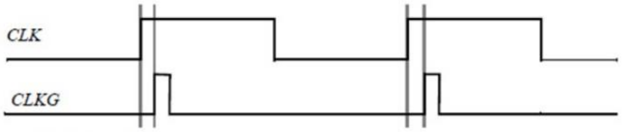
49. Draw the circuit and wave form for Pulse Registers.



(a) register



(b) glitch generation



(c) glitch clock

50. Define Pulse Registers.

The idea is to construct a short pulse around the rising (or falling) edge of the clock.

This pulse acts as the clock input to a latch, sampling the input only in a short window.

Race conditions are thus avoided by keeping the opening time (i.e, the transparent period) of the latch very short.

The combination of the glitch generation circuitry and the latch results in a positive edge-triggered register.

51. List the advantage and disadvantage of Pulse Registers.

Advantage

- The reduced clock load and the small number of transistors required.
- The glitch-generation circuitry can be amortized over multiple register bits.

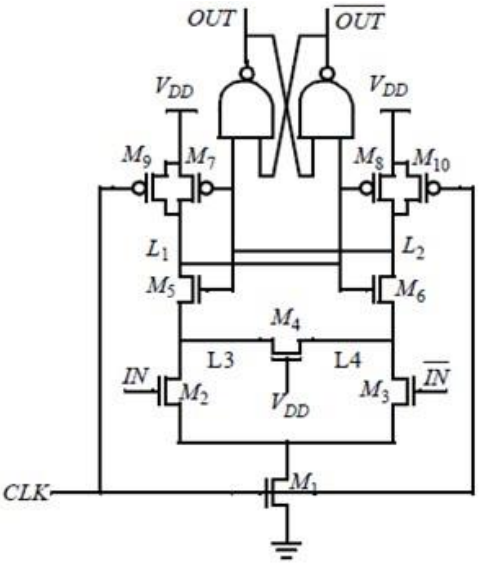
Disadvantage

- A substantial increase in verification complexity.
- This has prevented a wide-spread use.

52. What is Sense – Amplifier Based Registers?

- A sense amplifier structure to implement an edge- triggered register.
- Sense amplifier circuits accept small input signals and amplify them to generate rail-to-rail swings.
- There are many techniques to construct these amplifiers, with the use of feedback (e.g., cross-coupled inverters).

53. Draw the circuit of Sense – Amplifier Based Registers.



54. Differentiate between latch and flip-flop. (Nov 2015) (Nov 2019) [May 2021]
Compare register and latch. (April 2018) [April / May 2023]

Latch is a bistable device. i.e., it has two stable states. It is the level triggering method.

Flip-flop is a bistable device. i.e., it has two stable states. It is the edge triggering method.

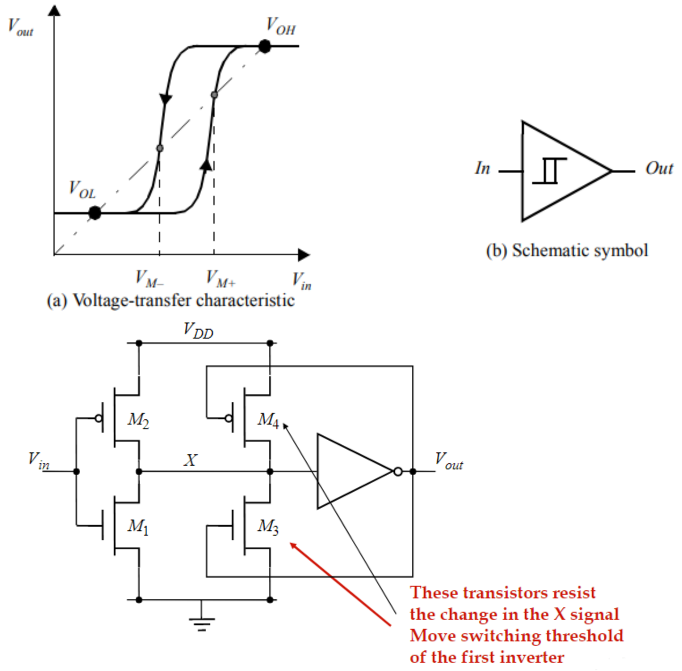
Register has number of flipflops.

55. Define Schmitt trigger.

A Schmitt trigger is a device with two important properties:

1. It responds to a slowly changing input waveform with a fast transition time at the output.
2. The voltage-transfer characteristic of the device displays different switching thresholds for positive- and negative-going input signals.

56. Draw the symbol, circuit and voltage transfer characteristics of Schmitt trigger.



(a) Voltage-transfer characteristic

(b) Schematic symbol

These transistors resist the change in the X signal Move switching threshold of the first inverter

57. Define Astable sequential circuits. Draw the circuit also. [April / May 2023]

- An astable circuit has no stable states.
- The output oscillates back and forth between two quasi-stable states with a period determined by the circuit topology and parameters (delay, power supply, etc.).
- One of the main applications of oscillators is the on-chip generation of clock signals.

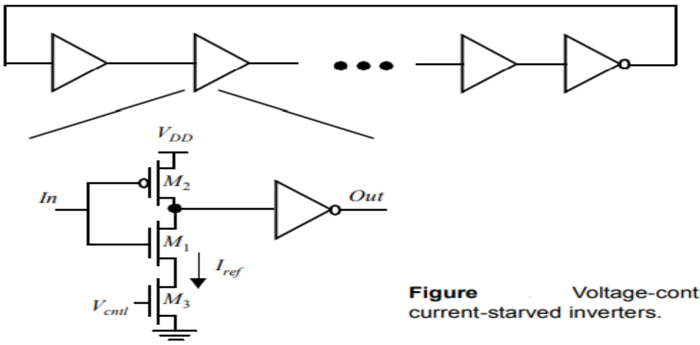
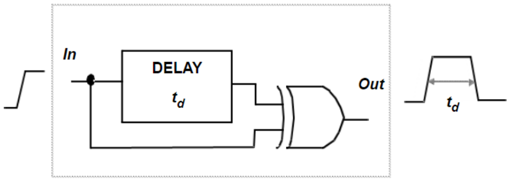


Figure Voltage-controlled oscillator based on current-starved inverters.

58. Define Monostable sequential circuits. [April / May 2023]

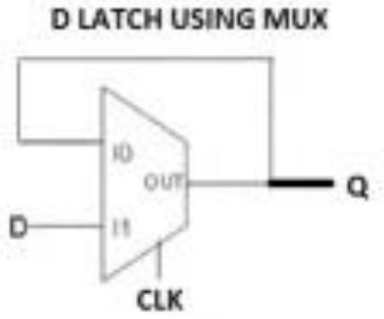
Monostable − Monostable circuits have only one stable operating point and even if they are temporarily perturbed to the opposite state, they will return in time to their stable operating point. Example: Timers, pulse generators.



59. State the use of Schmitt trigger.[Nov/Dec 2022]

Schmitt triggers can be used to change a sine wave into a square wave, clean up noisy signals, and convert slow edges to fast edges.

60. Draw a MUX based negative level sensitive D-latch. [Nov/Dec 2022]



61. List the timing classification of Digital system. [May 2021]

Synchronous, mesochronous and plesiochronous with respect to a system clock.

UNIT – IV
INTERCONNECT, MEMORY ARCHITECTURE AND ARITHMETICCIRCUITS

Interconnect Parameters – Capacitance, Resistance, and Inductance, Electrical Wire Models, **Sequential digital circuits:** adders, multipliers, comparators, shift registers. Logic Implementation using Programmable Devices (ROM, PLA, FPGA), Memory Architecture and Building Blocks, Memory Core and Memory Peripherals Circuitry.

4.1. Design of Data path circuits:

Discuss about data path circuits.

- Data path circuits are meant for passing the data from one segment to other segment for processing or storing.
- The datapath is the core of processors, where all computations are performed.
- It is generally defined with general digital processor. It is shown in figure.

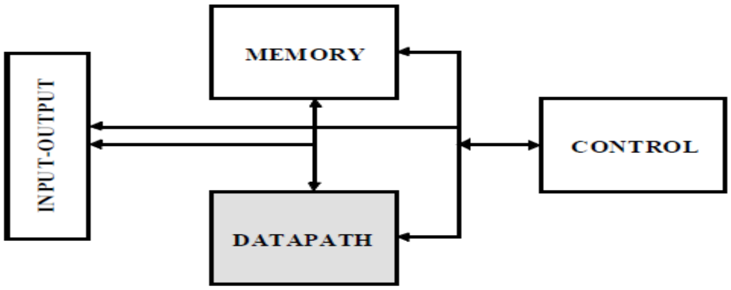
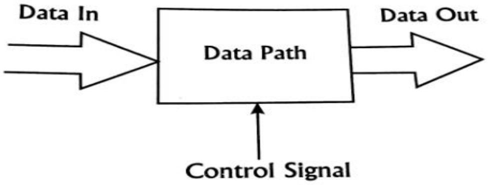
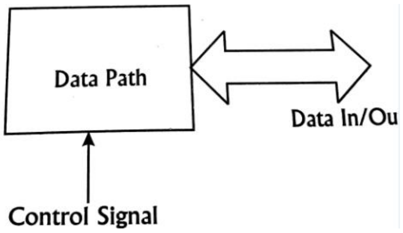


Figure: General digital processor

- If only data path and its communication is shown as



- In this, data is applied at one port and data output is obtained at second port.



- Data path block consists of arithmetic operation, logical operation, shift operation and temporary storage of operands.
- Datapaths are arranged in a bit sliced organization.
- Instead of operating on single bit digital signals, the data in a processor are arranged in a word based fashion.
- Bit slices are either identical or resemble a similar structure for all bits.

- The datapath consists of the number of bit slices (equal to the word length), each operating on a single bit. Hence the term is ***bit-sliced.***

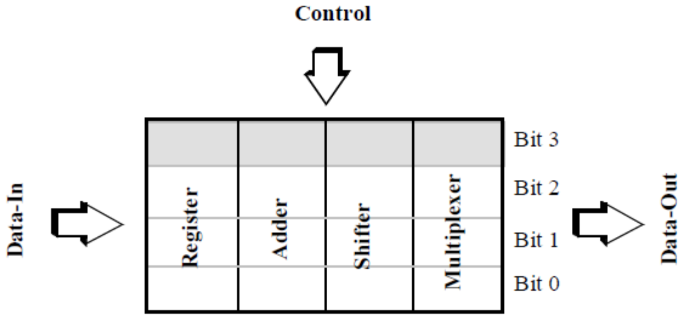


Figure: Bit-sliced datapath organization

****

4.2. Ripple Carry Adder:

- ❖ **Draw the structure of ripple carry adder and explain its operation. (Nov 2017)**
- ❖ **Explain the operation of a basic 4 bit adder. (Nov 2016)**
- ❖ **Realize a 1-bit adder using static CMOS logic. Optimize the Boolean expressions of sum and carryout and realize a 1-bit adder using static CMOS logic. Also realize a 1-bit adder using transmission gate. Compare all the three cases from hardware perspective. (Nov 2019)**

Architecture of Ripple Carry Adder:

- AOI Full adder circuit (AND OR INVERT)
- An AOI algorithm for static CMOS logic circuit can be obtained by using the equation.

$$C_{i+1} = a_i b_i + c_i.(a_i + b_i)$$

$$\overline{S_i} = (a_i + b_i + c_i)\overline{c_i} + (a_i.b_i.c_i)$$

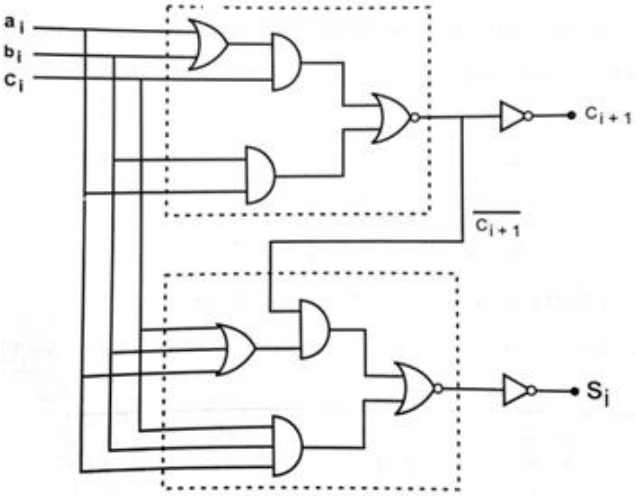


Figure: AOI Full adder

- If n bits are added, then we can get n-bit sum and carry of C_n.

C_i= Carry bit from the previous column.

- N bit ripple carry adder needs n full adders with C_{i+1} carry out bit.

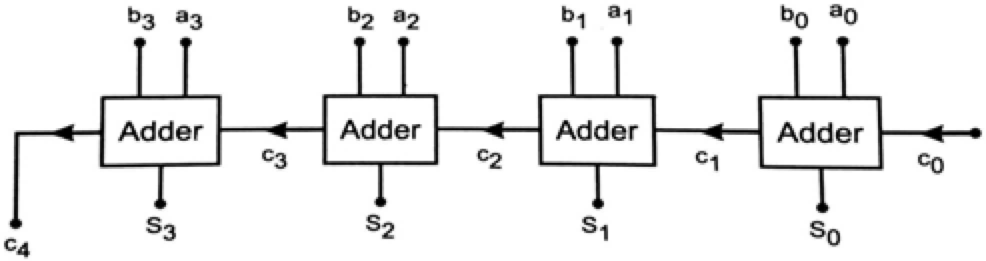


Figure: Ripple carry adder

- The overall delay depends on the characteristics of full adder circuit. Different CMOS implementation can produce different delay parts.
- t_{di}- worst case delay through the ith stage. We can calculate the total delay using the following equation

$$t_{4b} = t_{d3}+t_{d2}+t_{d1}+t_{d0} \text{and } t_{d0} = t_d(a_0,b_0 \longrightarrow c_1)$$

- This is the time for the input to produce the carry out bit.

$$t_{d1}= t_{d2} = t_d(c_{in} \longrightarrow c_{out})$$

$$t_{d3}= t_d(c_{in} \longrightarrow S_3)$$

$$t_{4b} = t_d(c_{in} \longrightarrow S_3) +2t_d(c_{in} \longrightarrow c_{out}) +t_d(a_0,b_0 \longrightarrow c_1)$$

- If it is extend to n-bit, then the worst case delay is

$$t_{n\text{-bit}} = t_d(c_{in} \longrightarrow S_{n-1}) + (n-2)t_d(c_{in} \longrightarrow c_{out}) +t_d(a_0,b_0 \longrightarrow c_1)$$

- Worst case delay linear with the number of bits

$$t_d = O(N)$$

$$t_{adder} = (N-1)t_{carry} + t_{sum}$$

- The figure below shows 4-bit adder/subtractor circuit.
- In this, if add/sub=0, then sum is a+b. If add/sub=1, then the output is a-b.

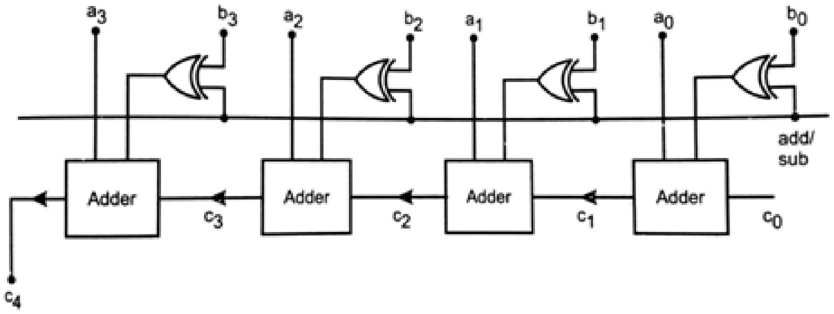


Figure:4-bit adder/subtractor circuit

- Sum and carry expressions are designed using static CMOS.
- It requires 28 transistors which lead large area and circuit is slow.

  Sum, S= $ABC_i + \bar{C}_0(A + B + C_i)$ and Carry, C_0= $AB + BC_i + C_iA$

Drawbacks:

- Circuit is slower.

- In ripple carry adder, carry bit is calculated along with the sum bit. Each bit must wait for calculation of previous carry.

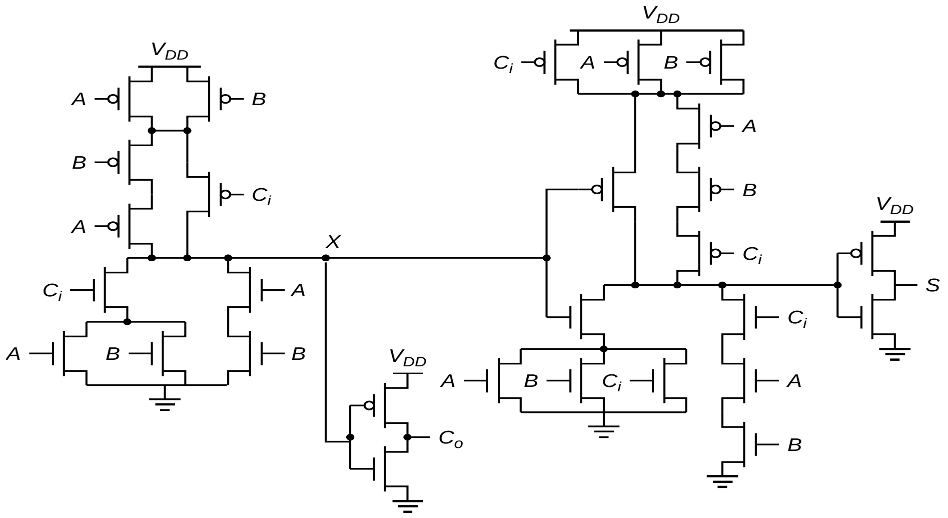


Figure:Complimentary Static CMOS Full Adder

****

4.3. Carry Look Ahead Adder (CLA):

- ❖ **Explain the operation and design of Carry lookahead adder (CLA). (May 2017, Nov 2016)[Apr/May 2022] [Nov/Dec 2022]**
- ❖ **How the drawback in ripple carry adder overcome by carry look ahead adder and discuss. (Nov 2017)**
- ❖ **Explain the concept of carry lookahead adder and discuss its types. (April 2018)**
- ❖ **Derive the necessary expressions of a 4 bit carry look ahead adder and realize the carry out expressions using dynamic CMOS logic. (April 2019-13M)**

- A carry-lookahead adder (CLA) is a type of adder used in digital circuit.
- A carry-lookahead adder improves speed by reducing the amount of time required todetermine carry bits.
- In ripple carry adder, carry bit is calculated alongwith the sum bit.
- Each bit must wait until the previous carry is calculated to begin calculating its own result and carry bits.
- The carry-lookahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger value bits.
- A ripple-carry adder works starting at the rightmost (LSB) digit position, the two corresponding digits are added and a result obtained. There may be a carry out of this digit position.
- Accordingly all digit positions other than LSB.Need to take into account the possibility to add an extra 1, from a carry that has come in from the next position to the right.
- Carry lookahead depends on two things:
  - ✓ Calculating, for each digit position, whether that position is going to propagate a carry if one comes in from the right.

- ✓ Combining these calculated values to be able to realize quickly whether, for each group of digits, that group is going to propagate a carry.

- Theory of operation:
  - ✓ Carry lookahead logic uses the concept of generating and propagating carry.
  - ✓ The addition of two 1-digit inputs A and B is said to generate if the addition will carry, regardless of whether there is an input carry.
- *Generate:*
  - ✓ In binary addition, A + B generates if and only if both A and B are 1.
  - ✓ If we write G(A,B) to represent the binary predicate that is true if and only if A + B generates, we have:

$$\boldsymbol{G(A,B) = A \,.\, B}$$

- *Propagate:*
  - ✓ The addition of two 1-digit inputs A and B is said to propagate if the addition will carry whenever there is an input carry.
  - ✓ In binary addition, A + B propagates if and only if at least one of A or B is 1.
  - ✓ If we write P(A,B) to represent the binary predicate that is true if and only if A + B propagates, we have:

$$P(A,B) = A \oplus B$$

- These adders are used to overcome the latency which is introduced by the rippling effect of carry bits.
- Write carry look-ahead expressions in terms of the generate g_i and propagate p_i signals. The general form of carry signal c_i thus becomes

$$c_{i+1} = a_i.b_i + c_i.(a_i \oplus b_i) = \mathrm{g_i} + c_i.\mathrm{p_i}$$

- If $a_i.b$ =1, then $c_{i+1} = 1$, write generate term as, $g_i = a_i.b_i$
- Write the propagate term as, $p_i = a_i \oplus b_i$
- Sum and carry expression are written as,

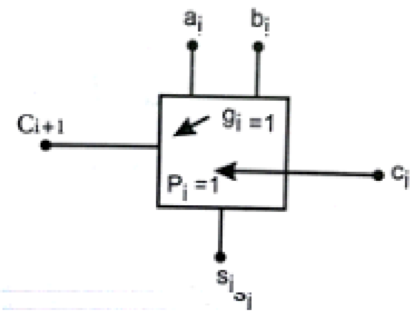
$S_i = a_i \oplus b_i$

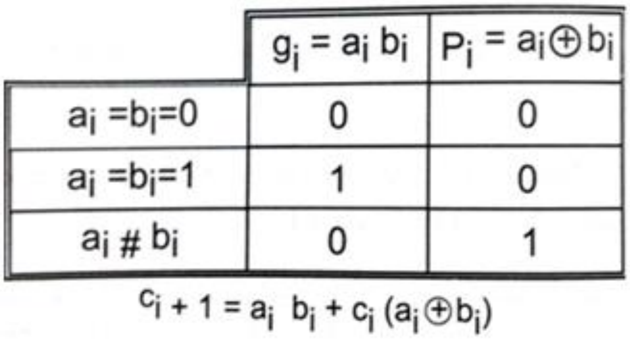
$c_1 = g_0 + p_0.c_0$

$c_2 = g_1 + p_1.c_1 = g_1 + p_1.(g_0 + p_0.c_0)$

$c_3 = g_2 + p_2.c_2$

$c_4 = g_3 + p_3.c_3 = g_3 + p_3.g_2 + p_3.p_2.g_1 + p_3.p_2.p_1.g_0 + p_3.p_2.p_1.p_0.c_0$





	$g_i = a_i\,b_i$	$P_i = a_i \oplus b_i$
$a_i = b_i = 0$	0	0
$a_i = b_i = 1$	1	0
a_i # b_i	0	1

$c_i + 1 = a_i\ b_i + c_i\,(a_i \oplus b_i)$

Figure:Symbol and truth table of generate & propagate

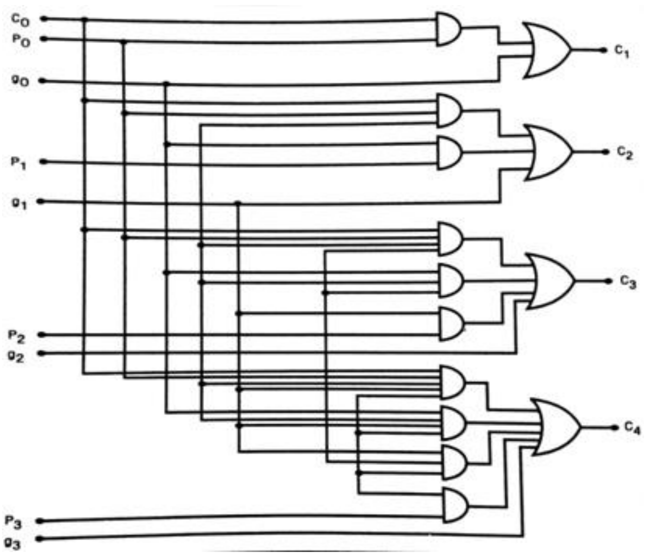


Figure – Logic network for 4-bit CLA carry bits

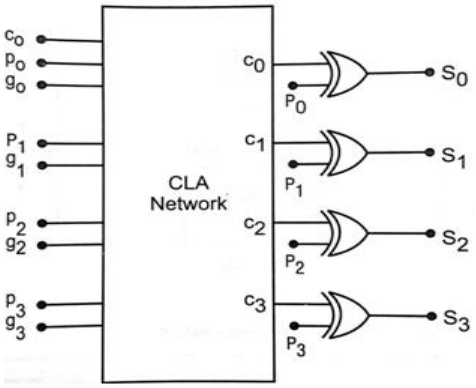


Figure – Sum calculation using the CLA network

- The symmetry in the array is shown in mirror. It allows more structured layout at the physical design level.

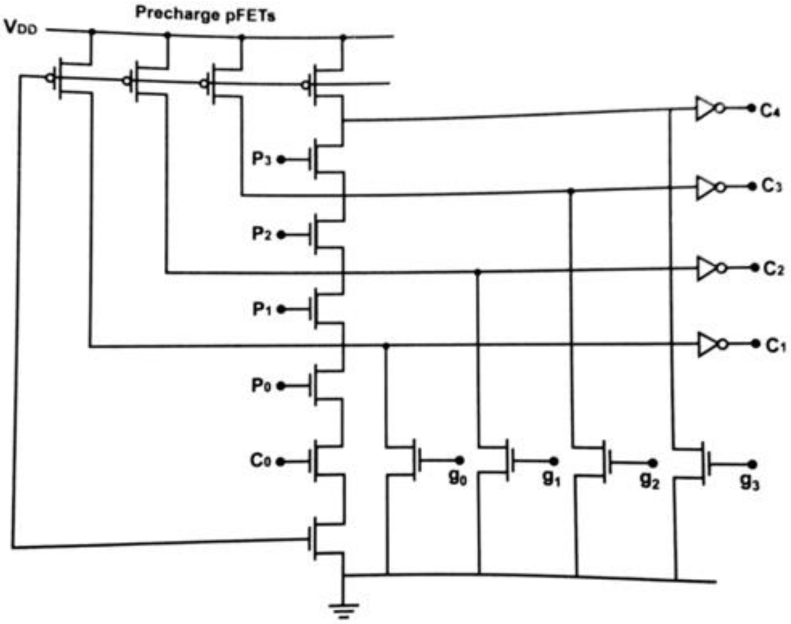


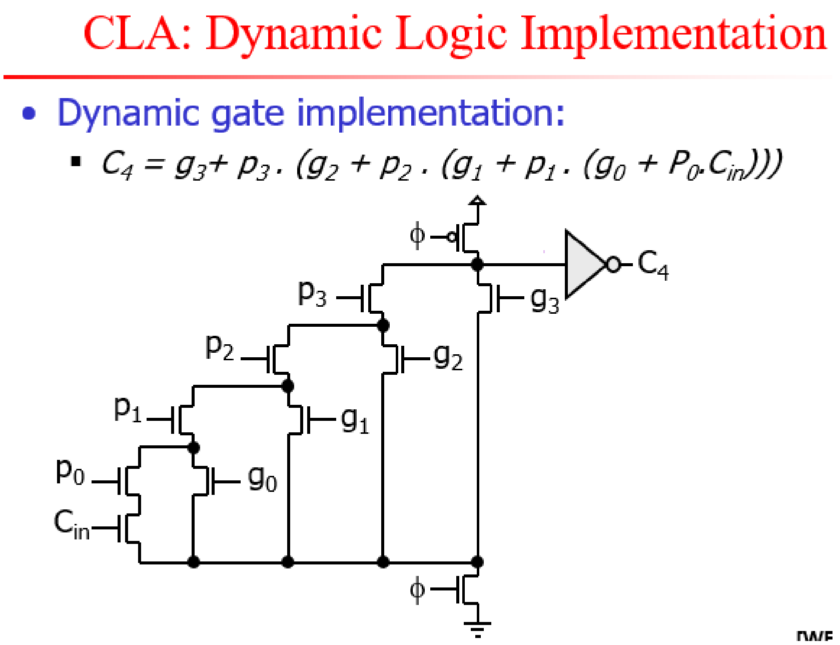
Figure – MODL carry circuit

- MODL-Multiple Output Domino Logic.
- MODL is non-inverting logic family and is a dynamic circuit technique.
- Its limitations are

  i. Clocking in mandatory

  ii. The output is subject to charge leakage and charge sharing.

  iii. Series connected nFET chains can give long discharge times.



CLA: Dynamic Logic Implementation

- Dynamic gate implementation:
  - $C_4 = g_3 + p_3 . (g_2 + p_2 . (g_1 + p_1 . (g_0 + P_0 . C_{in})))$

****

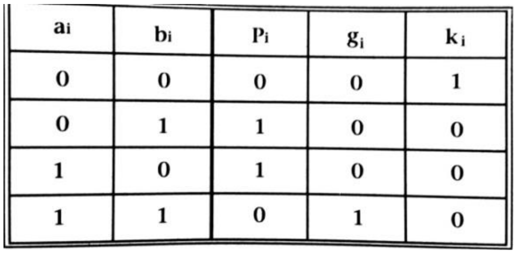
4.4. Manchester Carry Chain Adder:

Discuss about Manchester Carry Chain Adder.

- The Manchester carry chain is a variation of the carry-lookaheadadder that uses shared logic to lower the transistor count.
- A Manchester carry chain generates the intermediate carries by tapping off nodes in the gate that calculates the most significant carry value.
- Dynamic logic can support shared logic, as transmission gate logic.
- One of the major drawbacksof the Manchester carry chain is increase the propagation delay.
- A Manchester-carry-chain section generally won't exceed 4 bits.
- In this adder, the basic equation is $c_{i+1} = \mathrm{g_i} + c_i . \mathrm{p_i}$

Where $p_i = a_i \oplus b_i$ and $g_i = a_i . b_i$

- Carry kill bit $k_i = \overline{a_i + b_i} = \overline{a_i} . \overline{b_i}$
- If K_i=1, then p_i=0 and g_i=0. Hence, k_i is known as carry kill bit.



a_i	b_i	p_i	g_i	k_i
0	0	0	0	1
0	1	1	0	0
1	0	1	0	0
1	1	0	1	0

Table

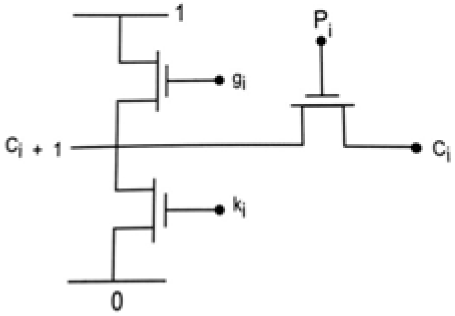
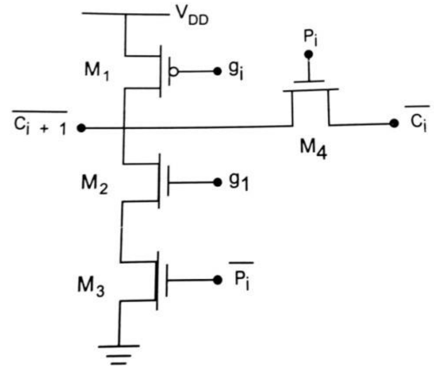


Figure – switch level circuit

- In the circuit shown below $\overline{C_l}$ is used as an input if P_i = 0, then M_3 is ON, M_4 is OFF.
- If g_i=0, then M_1 is ON, M_2 is ON
- If g_i=1, then M_2 is OFF, M_4 is ON and output equal to zero.
- If P_i=1, then this case is a complicated one.



- In dynamic circuit figure
- If ϕ = 0, then recharge occur and output is 1
- If ϕ = 1, then evaluation occur.

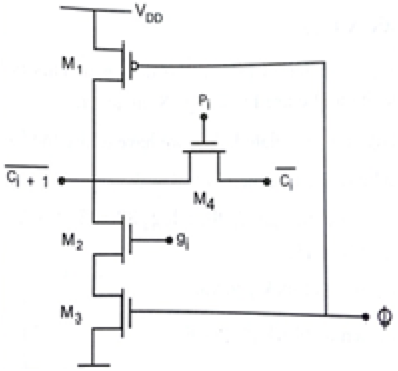
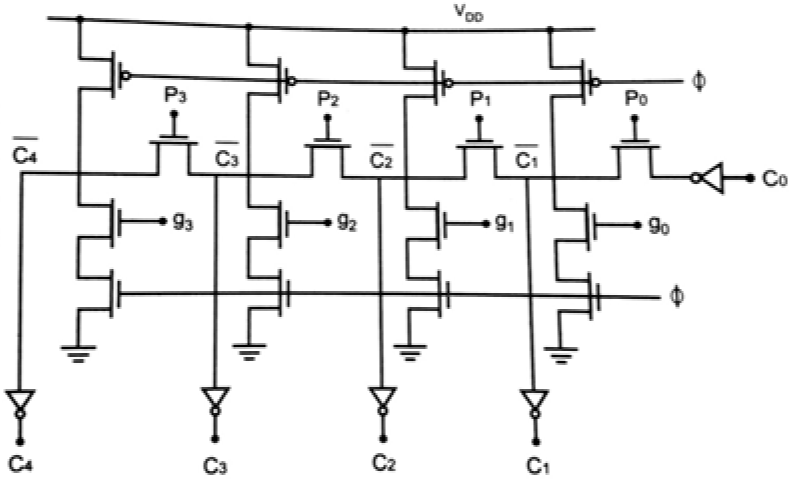


Figure dynamic circuit

- Dynamic Manchester carry chain for the carry bit upto C_4 is shown below. C_1, C_2, C_3, C_4 can be taken by using inverters. The carry input is given as C_0



****

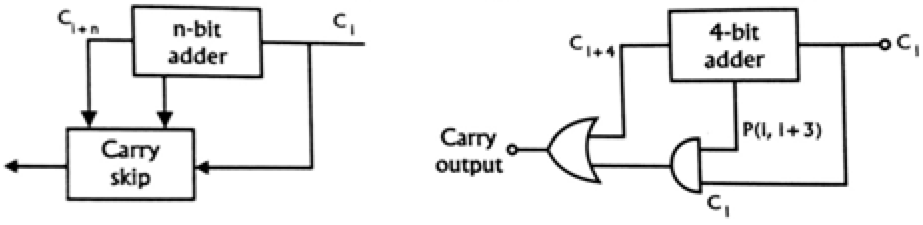
4.4.1. HIGH SPEED ADDERS:

- ❖ **Discuss about different types of high speed adders. (Apr. 2016)**
- ❖ **Describe the different approaches of improving the speed of the adder. (Nov 2016)**

(i) Carry Skip(bypass) Adder:

Design a carry bypass adder and discuss its features. (May 2016)
Explain the carry-propagate adder and show how the generation and propagation signals are framed. [May 2021]

- It is high speed adder. It consist of adder, AND gate and OR gate.
- An incoming carry $C_{i,0}$=1 propagates through the complete adder chain and an outgoing carry $C_{0,3}$=1.
- In other words, if ($P_0P_1P_2P_3$ =1) then $C_{0,3}$= $C_{i,0}$ else either DELETE or GENERATE occurred.
- It can be used to speed up the operation of the adder, as shown in below fig (b).



(a) General Block Diagram *(b) Logic Circuit of Carry Skip Adder*

Figure: Carry Skip Adder.

- When BP= $P_0P_1P_2P_3$ =1, the incoming carry is forwarded immediately to the next block.
- Hence the name carry bypass adder or carry skip adder.
- Idea: if (P_0 and P_1 and P_2 and P_3 =1) the C_{03} = C_0, else “kill” or “generate”.

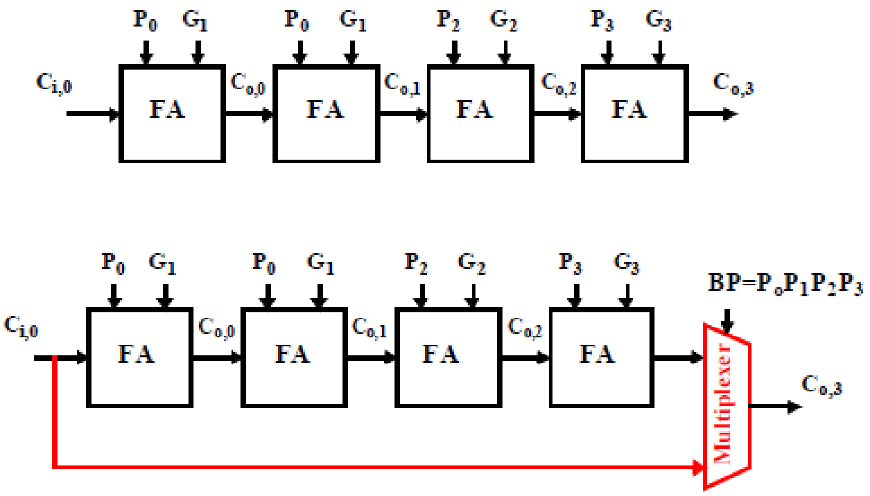
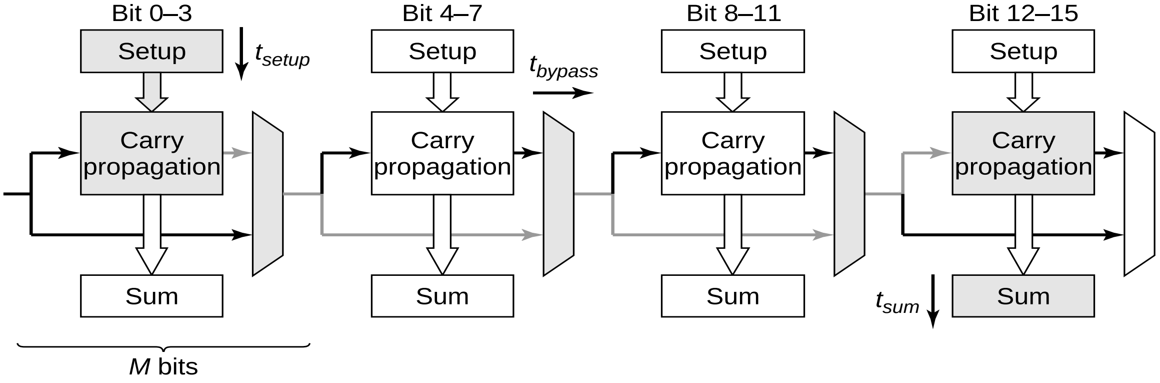


Figure: (a) Carry propagation (b) Adding a bypass

- The below figure shows n no. of bits carry skip adder.



$t_{adder} = t_{setup} + Mt_{carry} + (N/M-1)t_{bypass} + (M-1)\, t_{carry} + t_{sum}$(worst case)

t_{setup}: overhead time to create G, P, D signals

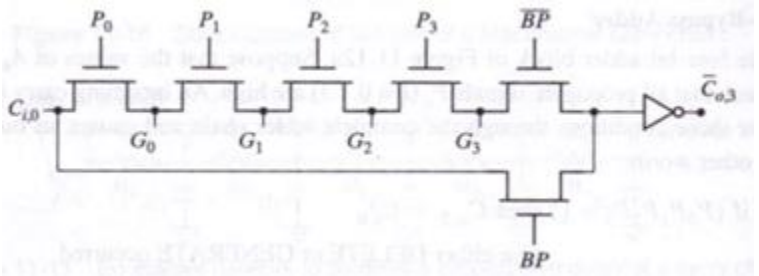


Figure: Manchester carry-chain implementation of bypass adder

(ii) Carry Select Adder:

Design a carry select adder and discuss its features. (May 2016)

- A carry-select adder is a particular way to implement an adder, which is a logic element that computes the (n+1)-bit sum of two n-bit numbers.
- The carry-select adder is simple but rather fast, having a gate level depth of O($\sqrt{n}$).
- The carry-select adder generally consists of two ripple carry adders and a multiplexer.
- Adding two n-bit numbers with a carry-select adder is done with two adders in order to perform the calculation twice.
- One time with the assumption of the carry-in being zero and the other assuming it will be one.
- After the two results are calculated (the correct sum as well as the correct carry-out),it is then selected with the multiplexer once the correct carry-in is known.
- The number of bits in each carry select block can be uniform, or variable.
- In the uniform case, the optimal delay occurs for a block size of $\sqrt{n}$.
- The O($\sqrt{n}$) delay is derived from uniform sizing, where the ideal number of full-adder elements per block is equal to the square root of the number of bits being added.
- Propagation delay, P is equal to $\sqrt{2N}$ where N = N- bit adder
- Below is the basic building block of a carry-select adder, where the block size is 4.

- Two 4-bit ripple carry adders are multiplexed together, where the resulting carry and sum bits are selected by the carry-in.

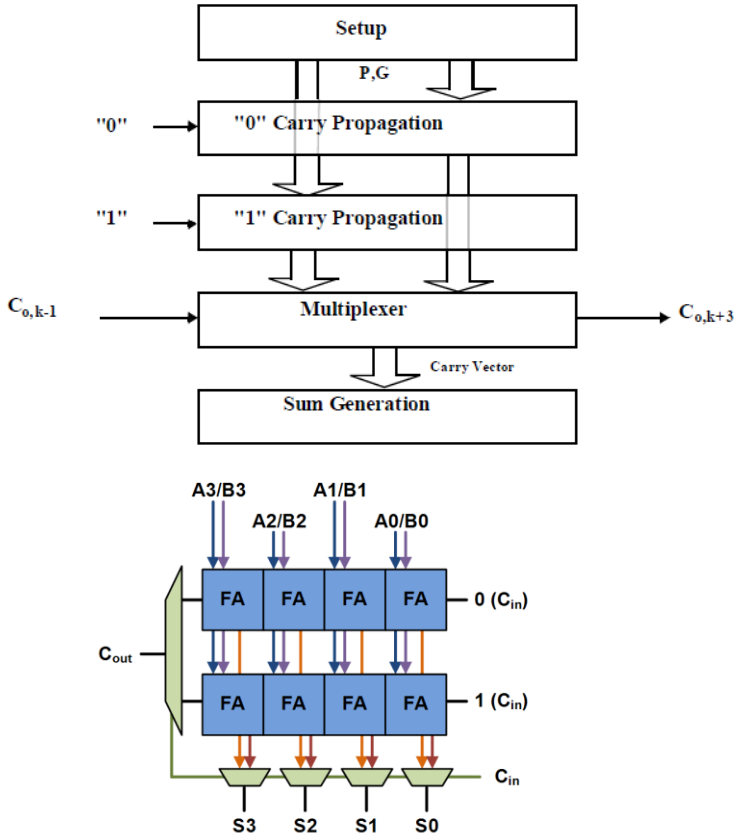


Figure: Building blocks of a carry-select adder

Uniform-sized adder:

- A 16-bit carry-select adder with a uniform block size of 4 can be created with three of these blocks and a 4-bit ripple carry adder.
- Since carry-in is known at the beginning of computation, a carry select block is not needed for the first four bits.
- The delay of this adder will be four full adder delays, plus three MUX delays.
- $t_{adder} = t_{setup} + Mt_{carry} + (N/M)t_{mux} + t_{sum}$

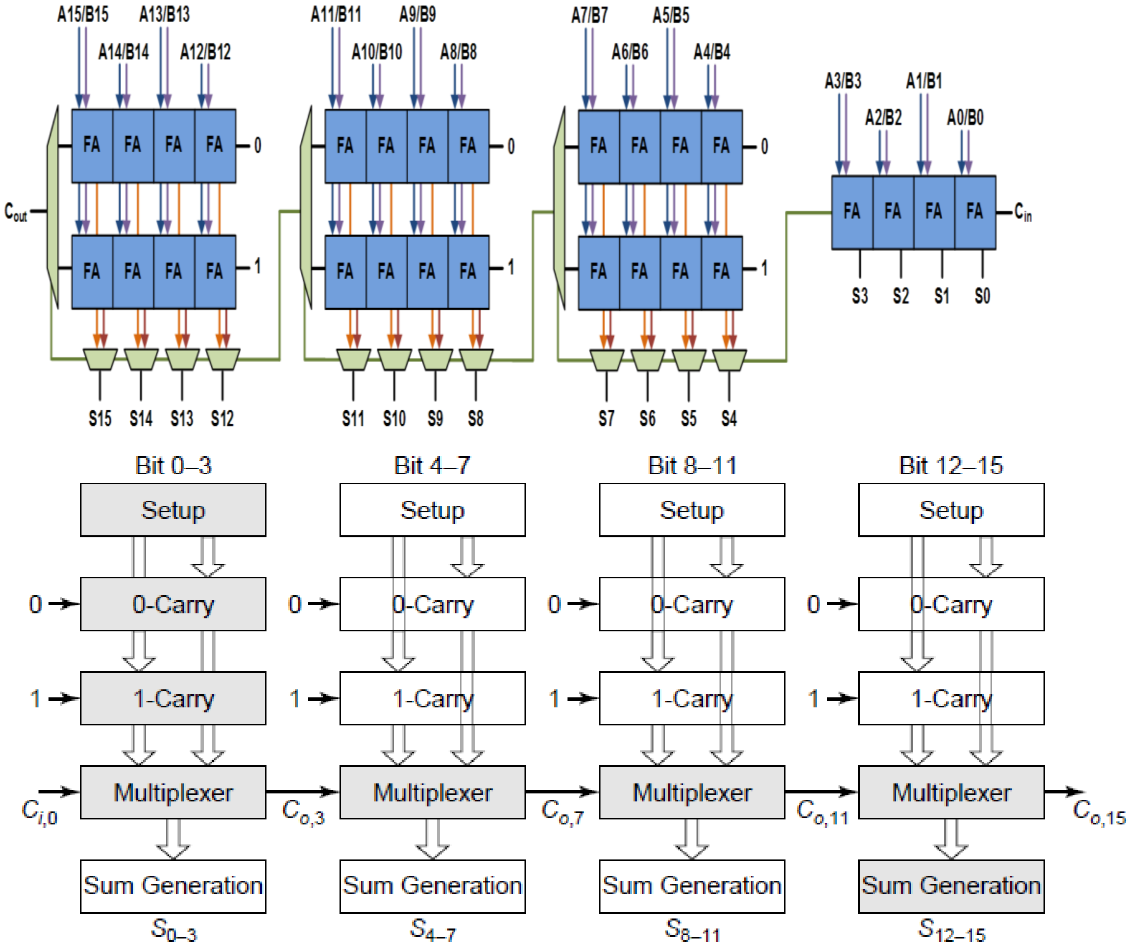
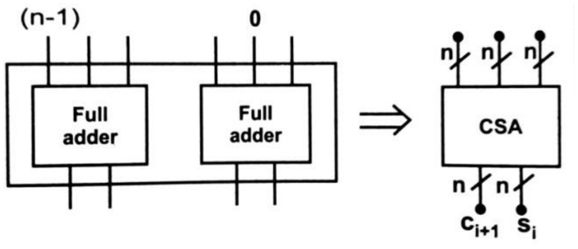


Figure: general structure of 16 bit adder

Disadvantage: hardware cost is increased.

(iii) Carry Save Adder:

- Carry save adder is similar to the full adder. It is used when adding multiple numbers.
- All the bits of a carry save adder work in parallel.
- In carry save adder, the carry does not propagate. So, it is faster than carry propagate adder.
- It has three inputs and produces 2 outputs, carry-out is saved. It is not immediately used to find the final sum value.



n-bit Carry Save Adder

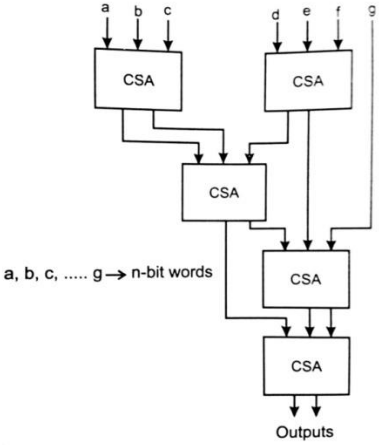


Figure: ***Carry Save Adder***

****

4.5. ALUs (ACCUMULATOR):

Briefly discuss about ALUs (accumulators).

- Accumulator acts as a part of ALU and it is identified as register A. The result of an operation performed in the ALU is stored in the accumulator.
- It is used to hold the data for manipulation (arithmetic and logical)
- Arithmetic functions are very important in VLSI. Ex: multiplication.
- Half adder circuit has two inputs and two outputs. S = $x \oplus y$, C = x.y.

$$S = x \oplus y$$
$$C = x \cdot y$$

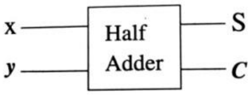
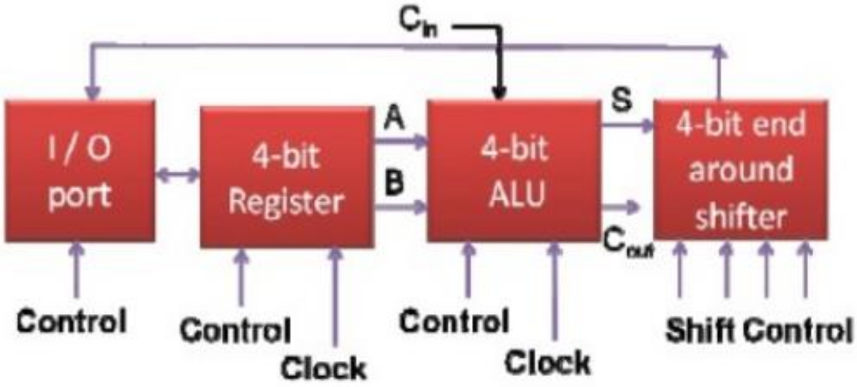
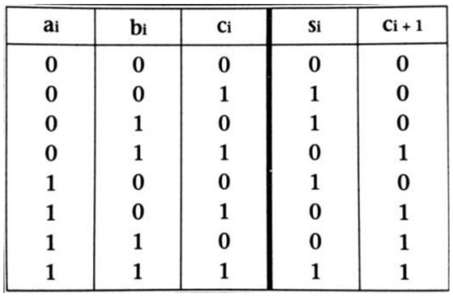


Figure: Half adder



- Full adder circuit has three inputs and two outputs



a_i	b_i	c_i	s_i	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure : Full adder and truth table

CPL --- Complementary Pass Logic

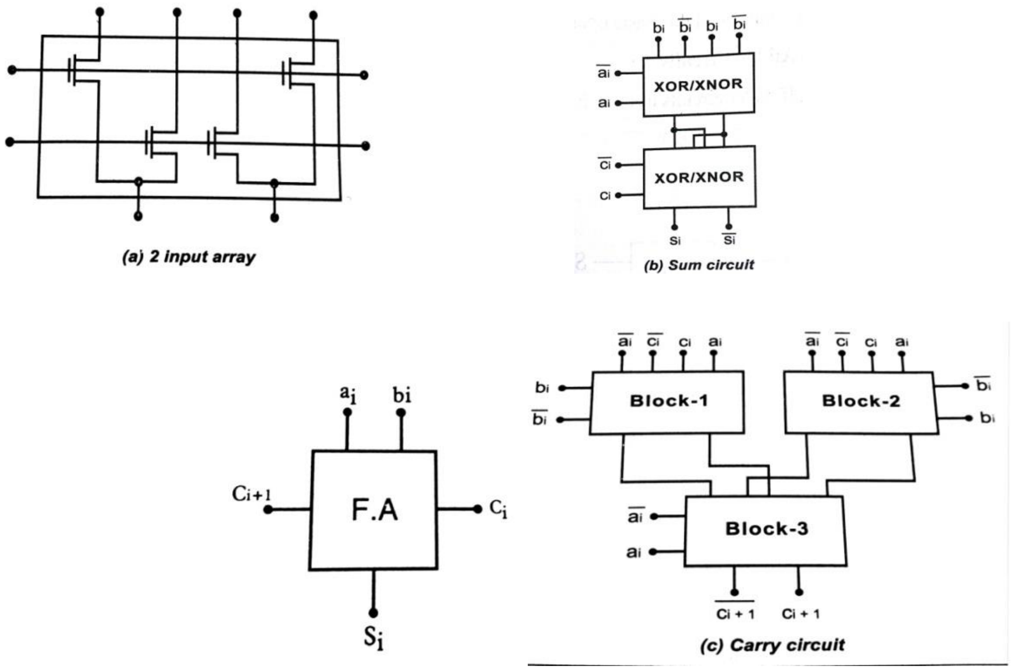


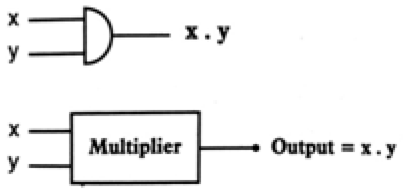
Figure : CPL Full adder design

****

4.6. MULTIPLIERS:

- ❖ **Explain the design and operation of 4 x 4 multiplier circuit. (Apr. 2016, 2017, Nov 2016, 2018)**
- ❖ **Design a multiplier for 5 bit by 3 bit. Explain its operation and summarize the numbers of adders. Discuss it over Wallace multiplier. (Nov 2017, April 2018)**
- ❖ **Design a 4 bit unsigned array multiplier and analyze its hardware complexity. (April 2019-13M) (Nov 2019)**
- ❖ **Describe the hardware architecture of a 4-bit signed array multiplier. [Nov/Dec 2022]**

- A study of computer arithmetic processes will reveal that the most common requirements are for addition and subtraction.
- There is also a significant need for a multiplication capability.
- Basic operations in multiplication are given below.
  0x0=0, 0x1=0, 1x0=0, 1x1=1



x	y	x . y
0	0	0
0	1	0
1	0	0
1	1	1

```
              1 0 1 0 1 0       Multiplicand
x                 1 0 1 1       Multiplier
-------------------------
              1 0 1 0 1 0   }
            1 0 1 0 1 0     }
          0 0 0 0 0 0       }   Partial products
+       1 0 1 0 1 0         }
-------------------------
        1 1 1 0 0 1 1 1 0       Result
```

- If two different 4-bit numbers (x_0, x_1, x_2, x_3& y_0, y_1, y_2, y_3)are multiplied then

$$\begin{array}{cccccccc}
 & & & & x_3 & x_2 & x_1 & x_0 \\
 & & & & y_3 & y_2 & y_1 & y_0 \quad \times \\
\hline
 & & & & x_3 y_0 & x_2 y_0 & x_1 y_0 & x_0 y_0 \\
 & & & x_3 y_1 & x_2 y_1 & x_1 y_1 & x_0 y_1 & \\
 & & x_3 y_2 & x_2 y_2 & x_1 y_2 & x_0 y_2 & & \\
 & x_3 y_3 & x_2 y_3 & x_1 y_3 & x_0 y_3 & & & \\
\hline
 & x_3 y_3 & (x_3 y_2 + & (x_3 y_1 + & (x_3 y_0 + & (x_2 y_0 + & (x_1 y_0 + & x_0 y_0 \\
 & & x_2 y_3) & x_2 y_2 + & x_2 y_1 + & x_1 y_1 + & x_0 y_1) & \\
 & & & x_1 y_3) & x_1 y_2 + & x_0 y_2) & & \\
 & & & & x_0 y_3) & & & \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}$$

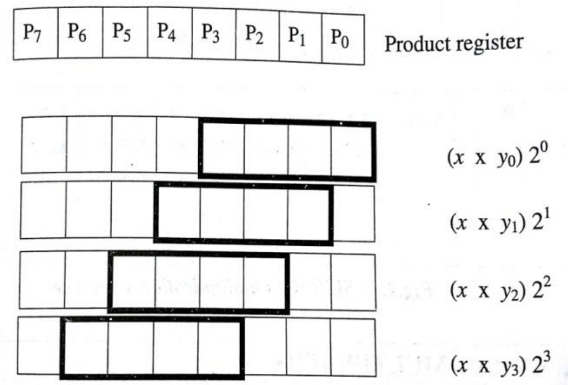
Product bits → ($P_7\ P_6\ P_5\ P_4\ P_3\ P_2\ P_1\ P_0$)

$$P_i = \sum_{i=j+k} x_j y_k + c_{i-1}$$

$c_{i-1} = 0$ for $(i - 1) \leq 0$

Multiplication by shifting:

- If $x=(0010)_2 = (2)_{10}$
- If it is to be multiplied by 2, then we can shift x in left side. $x = (0100)_2 = (4)_{10}$
- If it is to be divided by 2, then we can shift in right side. $x = (0001)_2 = (1)_{10}$.
- So, shift register can be used for multiplication or division by 2.



- A practical implementation is based on the sequence. The product is obtained by successive addition and shift right operations

(i) Array multiplier:

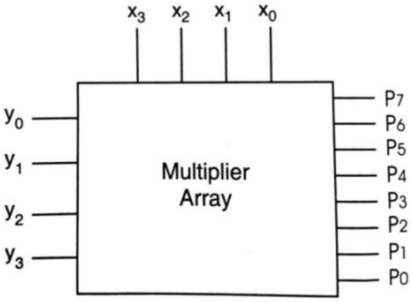


Figure: General block diagram of multiplier

- Array multiplier uses an array of cells for calculation.
- Multiplier circuit is based on repeated addition and shifting procedure. Each partial product is generated by the multiplication of the multiplicand with one multiplier digit.
- The partial products are shifted according to their bit sequences and then added.
- N-1 adders are required where N is the number of multiplier bits.
- The method is simple but the delay is high and consumes large area by using ripple carry adder for array multiplier.Product expression is given below

$$P_i = \sum_{i=j+k} x_j y_k + c_{i-1}$$

$$P = X.Y = \left(\sum_{j=0}^{n-1} x_j 2^j\right)\left(\sum_{k=0}^{n-1} y_k 2^k\right)$$

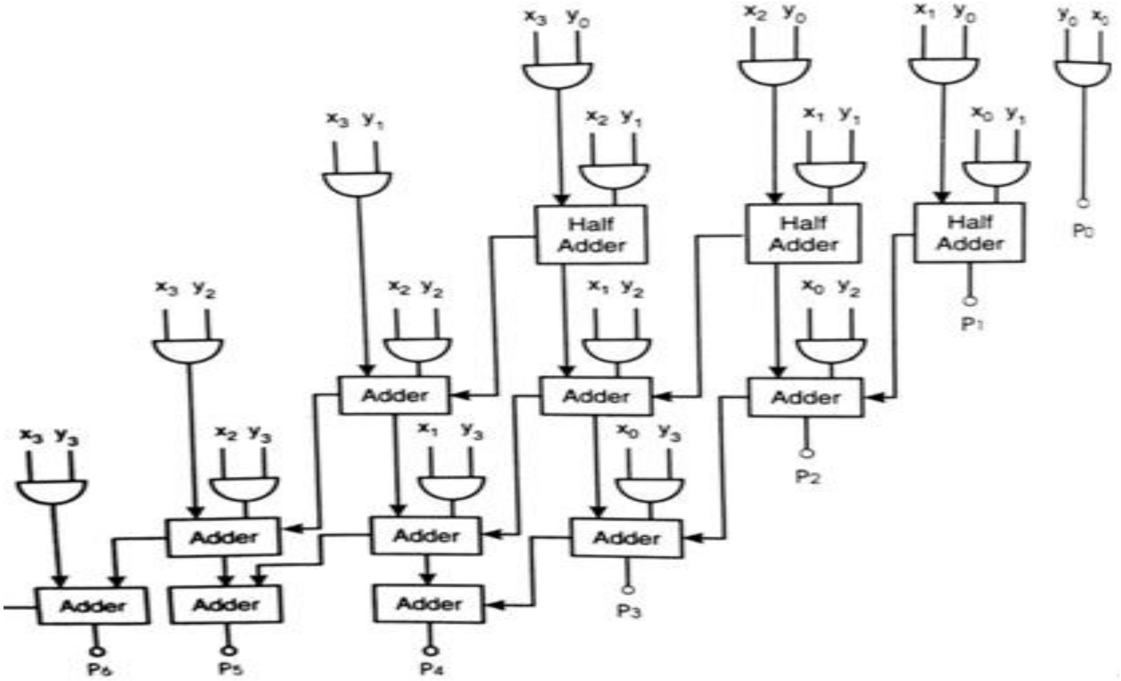


Figure: 4 x 4 array multiplier

- This multiplier can accept all the inputs at the same time. An array multiplier for n-bit word need n(n-2) full adders, n-half adder and n^2 AND gates.

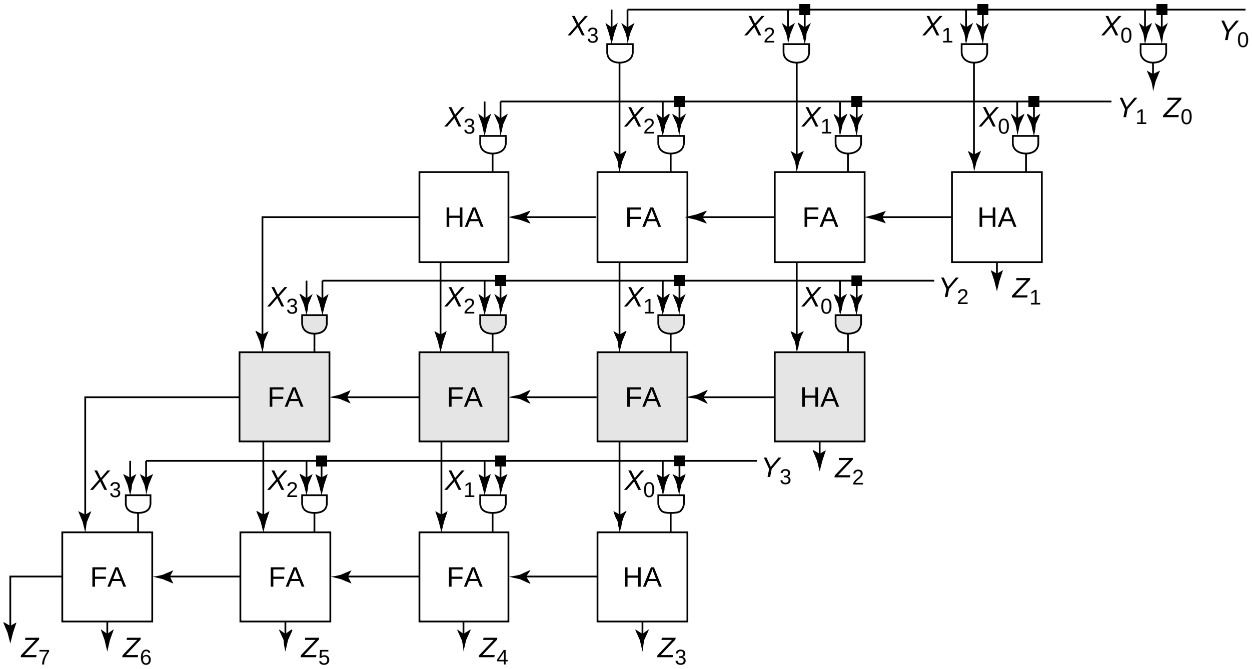


Figure: 4 x 4 array multiplier using Fulladder, Halfadder and AND gate.

(iv) Booth (encoding) multiplier:

- Booth's algorithm is an efficient hardware implementation of a digital circuit that multiplies two binary numbers in two's complement notation.
- Booth multiplication is a fastest technique that allows for smaller, faster multiplication circuits, by recoding the numbers that are multiplied.
- The Booths multipliers widely used in ASIC oriented products due to the higher computing speed and smaller area.
- In the binary number system, the digits called bits are to the set of {0,1}.
- The result of multiplying any binary number by binary bit is either 0 or original number.
- This makes the formation of partial products are more efficient and simple.
- Then adding all these partial products is time consuming task for any binary multipliers.
- The entire process consists of three steps partial product generation, partial product reduction and addition of partial products as shown in figure.

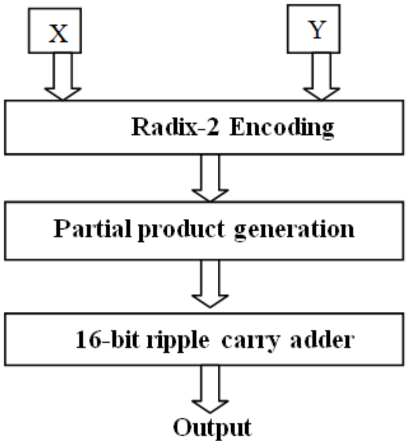


Figure: Block diagram of Booth multiplier

- But in booth multiplication, partial product generation is done based on recoding scheme e.g. radix 2 encoding.
- Bits of multiplicand (Y) are grouped from left to right and corresponding operation on multiplier (X) is done in order to generate the partial product.
- In radix-2 booth multiplication partial product generation is done based on encoding which is as given by Table.

Q_n	Q_{n+1}	Recoded Bits	Operation
0	0	0	Shift
0	1	+1	Add X
1	0	-1	Subtract X
1	1	0	Shift

Table: Booth encoding table with RADIX-2

- RADIX-2 PROCEDURE:

1) Add 0 to the LSB of the multiplier and make the pairing of 2 from the right to the left which shown in the figure.

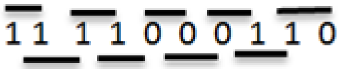


Figure: 2- Bit pairing as per Booth recoding using Radix- 2.

2) 00 and 11: do nothing according to the encoding table.

3)01: mark shows the end of the string’ of 1and add multiplicand to the partial product.

4)10: mark shows beginnings of the string of 1 subtract multiplicand from partial product.

With suitable example and with detailed steps explain Radix-4 modified booth encoding for an 8-bit signed multiplier. (Nov 2019)

Modified Booth Multiplier using Radix -4:

- **The disadvantage of Booth Multiplier with Radix-2 is increasing partial products.**
- **Modified Booth Multiplier with Radix-4 is reducing the half of the partial products in multipliers.**
- Modified Booth multiplication is a technique that allows for smaller, faster circuits by recoding the numbers that are multiplied.
- In Radix-4, encoding the multiplicands based on multipliers bits. It will compare 3-bits at a time with overlapping technique.
- Grouping starts from the LSB and the first block contains only two bits of the multipliers and it assumes zero for the third bit.

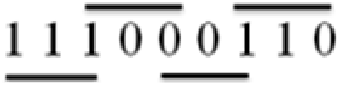


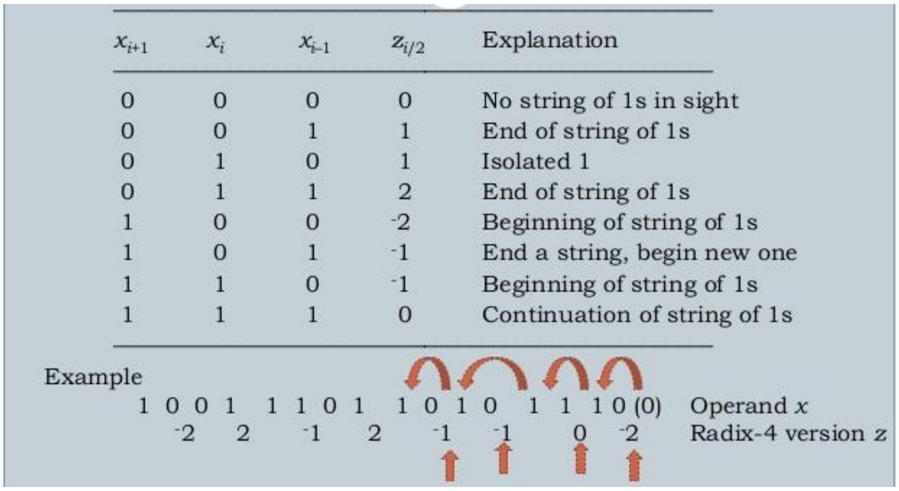
Figure. Grouping of 3-bit as per booth recoding

- These group of binary digits are according to the Modified Booth Encoding Table and it is one of the numbers from the set of (-2,2,0,1,-1).

groups	Partial products
000	0
001	1*multiplicand
010	1*multiplicand
011	2*multiplicand
100	-2*multiplicand
101	-1*multiplicand
110	-1*multiplicand
111	0

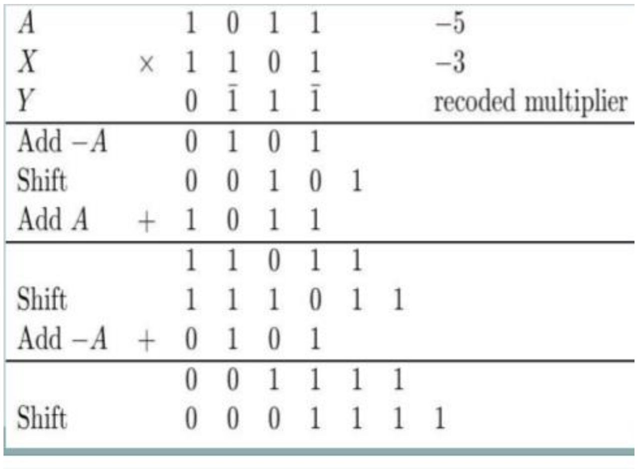
Table: Booth encoding table with RADIX-4

Multiplier Bits Block			Recoded 1-bit pair		2 bit booth	
i+1	i	i-1	i+1	i	Multiplier Value	Partial Product
0	0	0	0	0	0	Mx0
0	0	1	0	1	1	Mx1
0	1	0	1	-1	1	Mx1
0	1	0	1	0	2	Mx2
1	0	0	-1	0	-2	Mx-2
1	0	1	-1	1	-1	Mx-1
1	1	0	0	-1	-1	Mx-1
1	1	0	0	0	0	Mx0



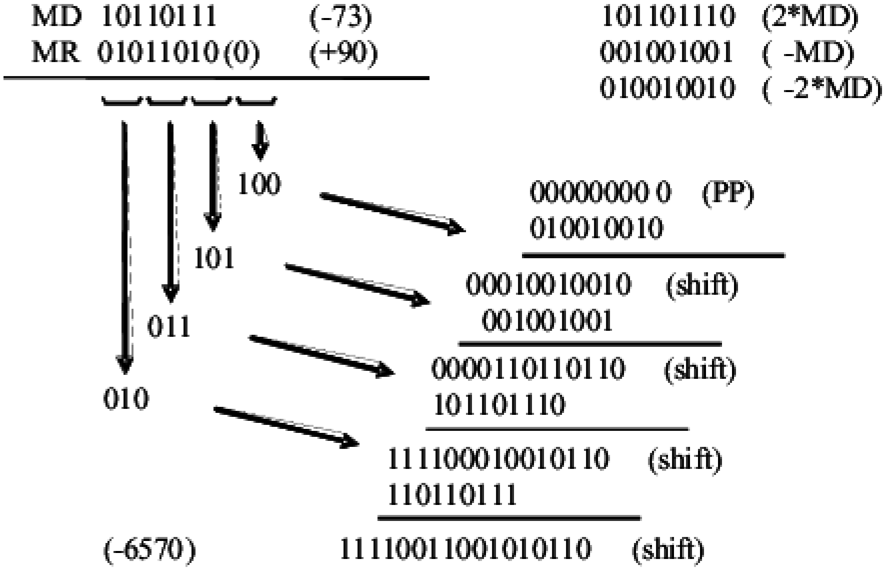
x_{i+1}	x_i	x_{i-1}	$z_{i/2}$	Explanation
0	0	0	0	No string of 1s in sight
0	0	1	1	End of string of 1s
0	1	0	1	Isolated 1
0	1	1	2	End of string of 1s
1	0	0	-2	Beginning of string of 1s
1	0	1	-1	End a string, begin new one
1	1	0	-1	Beginning of string of 1s
1	1	1	0	Continuation of string of 1s

```
Example
    1 0 0 1   1 1 0 1   1 0 1 0   1 1 1 0 (0)   Operand x
      -2   2     -1   2     -1   -1     0   -2   Radix-4 version z
```



A		1	0	1	1				−5
X	×	1	1	0	1				−3
Y		0	$\bar{1}$	1	$\bar{1}$				recoded multiplier
Add $-A$		0	1	0	1				
Shift		0	0	1	0	1			
Add A	+	1	0	1	1				
		1	1	0	1	1			
Shift		1	1	1	0	1	1		
Add $-A$	+	0	1	0	1				
		0	0	1	1	1	1		
Shift		0	0	0	1	1	1	1	

- **RADIX-4 PROCEDURE: [May 2021 (Model)]**
  1)Add 0 to the right of the LSB of the multiplier.
  2)Extend the sign bit 1 position if it is necessary when n is even.
  3)Value of each vector,the partial product is coming from the set of (-2,2,0,1,-1).



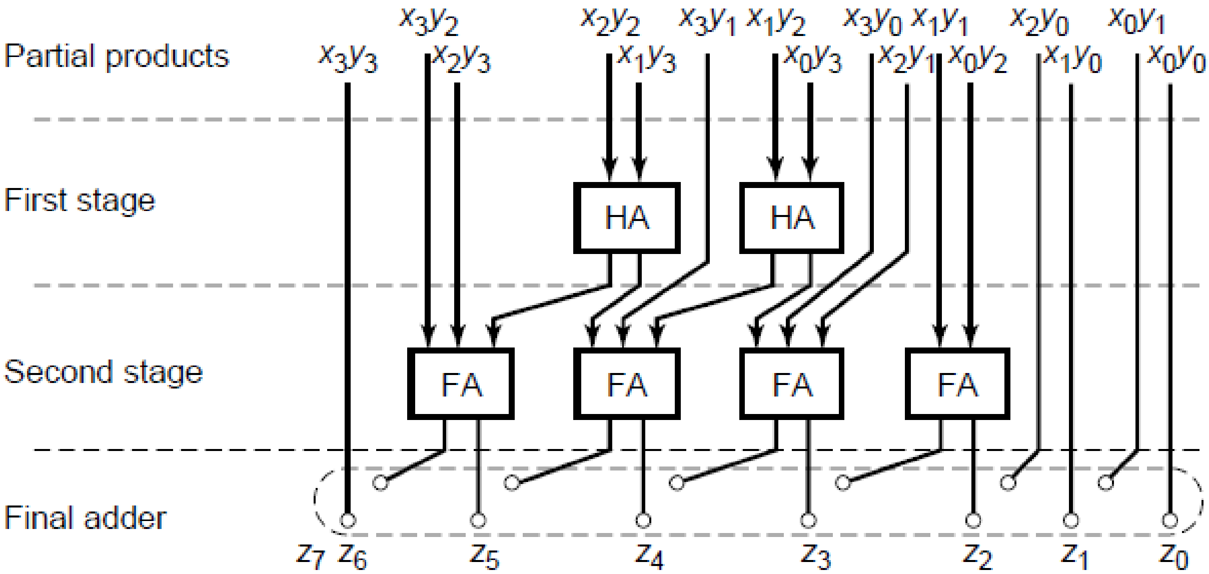
(v) Wallace tree Multiplier:

- A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integer numbers.
- The Wallace tree multiplier has three steps to be followed,
  (a) Multiply each bit of one of the arguments, by each bit of the other, yielding n^2 results.
  (b) Reduce the number of partial products to two by layers of full and half adders.
  (c) Group the wires in two numbers and add them with a conventional adder.
- The second section works as follows,
  (a) Take any three wires with 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.

(b) If there are two wires of the same weight left, input them into a half adder.

(c) If there is just one wire left and connects it to next layer.

- The Wallace tree multiplier output structure is tree basis style. It reduces the number of components and reduces the area.
- The architecture of a 4 x4 Wallace tree multiplier is shown in figure.



❖ **Apply radix-2 booth encoding to realize a 4-bit signed multiplier for (-10)*(-11). (April 2019-15M) [Apr/May 2022][Nov/Dec 2022]**

Solution:

M= -10 =0110, Q= -11 =0101

	A	Q	Q_{-1}	
Step-I:	0000	0101	0	:last 2 bits are10; A=A-M
	1010	0101	0	: shift right
	1101	0010	1	
Step-II:	0011	0010	1	:last 2 bits are 01; A=A+M
	0001	1001	0	:shift right
Step-III:	1011	1001	0	:last 2 bits are10; A=A-M
	1101	1100	1	;shift right
Step-IV:	1101	1100	1	;last 2 bits are 01; A=A+M
	0110	1110	0	;shift right

****

4.7. DIVIDERS

Explain in detail about the design and procedure for dividers.

- There are two types of dividers, Serial divider and Parallel divider. Serial divider is slow and parallel divider is fast in performance.

- Generally division is done by repeated subtraction. If 10/3 is to be performed then,

  10 -3 =7, (divisor is 3, dividend is 10)

  7 – 3 = 4,

  4 – 3 = 1

- Here, repeated subtraction has been done, after 3 subtractions, the remainder is 1. It is less than divisor. So now the subtraction is stopped.
- Let see the example of binary division with use of 1’s complement method

  1010 (10_d) / 0011 (3_d)

  Step1: find 1’s complement of divisor

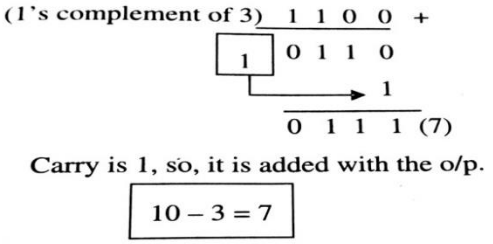
  Step2: add this with the dividend

  Step3: if carry is 1, then it is added with the output to get the difference output

  Step4: the same procedure is repeated until we are get carry 0.

  Step5: then the process is stopped.

1 0 1 0 (10)



(1’s complement of 3) 1 1 0 0 +

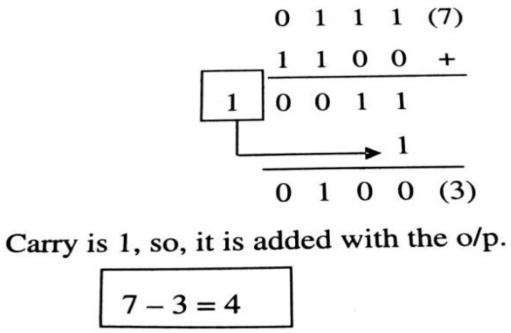
$\boxed{1}$ 0 1 1 0

$\longrightarrow$ 1

0 1 1 1 (7)

Carry is 1, so, it is added with the o/p.

$\boxed{10 - 3 = 7}$



0 1 1 1 (7)

1 1 0 0 +

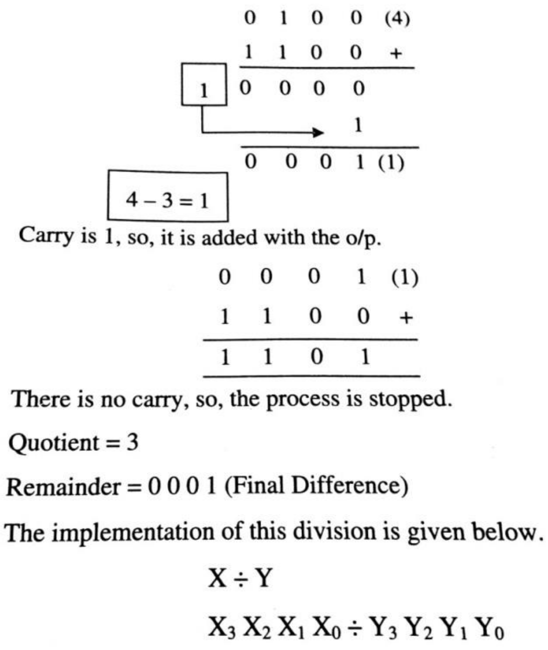
$\boxed{1}$ 0 0 1 1

$\longrightarrow$ 1

0 1 0 0 (3)

Carry is 1, so, it is added with the o/p.

$\boxed{7 - 3 = 4}$



$$\begin{array}{cccccc} & 0 & 1 & 0 & 0 & (4) \\ & 1 & 1 & 0 & 0 & + \\ \hline \boxed{1} & 0 & 0 & 0 & 0 & \\ & & & & 1 & \\ \hline & 0 & 0 & 0 & 1 & (1) \end{array}$$

$\boxed{4 - 3 = 1}$

Carry is 1, so, it is added with the o/p.

$$\begin{array}{ccccc} 0 & 0 & 0 & 1 & (1) \\ 1 & 1 & 0 & 0 & + \\ \hline 1 & 1 & 0 & 1 & \\ \hline \end{array}$$

There is no carry, so, the process is stopped.

Quotient = 3

Remainder = 0 0 0 1 (Final Difference)

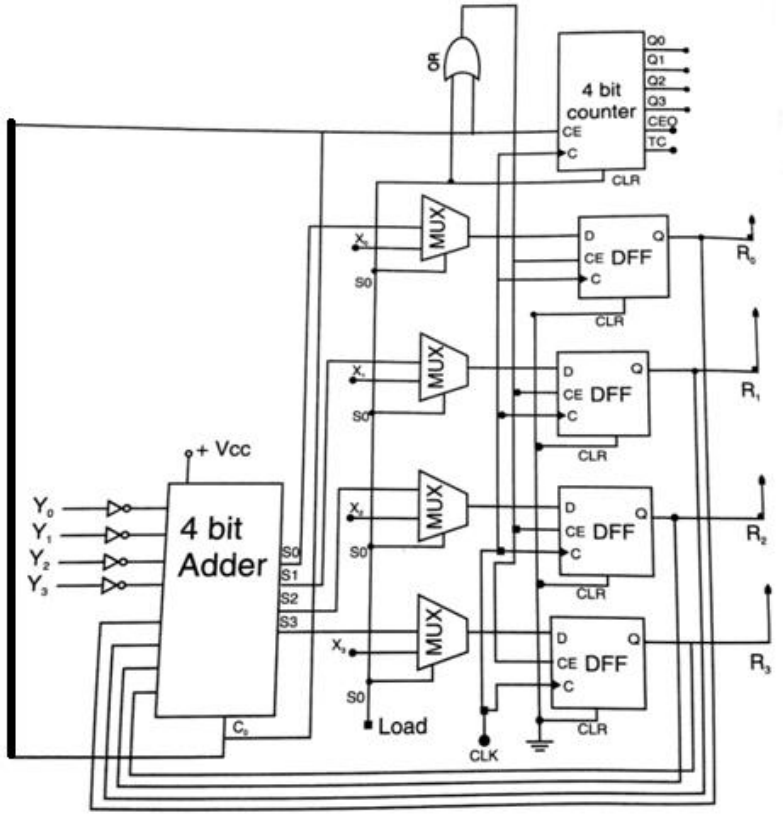
The implementation of this division is given below.

$X \div Y$

$X_3\, X_2\, X_1\, X_0 \div Y_3\, Y_2\, Y_1\, Y_0$

- Basic building blocks of serial adder are given below.
  1. 4 bit adder
  2. 4 bit binary up counter
  3. 2:1 MUX (4 MUXs are used)
  4. D flipflop

- Y_0 Y_1 Y_2 Y_3 are complemented and given to 4 bit adder block (figure shown below)
- X_0 X_1 X_2 X_3 are given to MUXs and MUX output is given to D flipflop. Select signal of MUX is high. It is connected to clear input of counter.
- Carry output of adder is connected with clock enable pin of counter. The same is given to OR gate. The output of this OR gate is given to clock enable signal of flipflops.
- The other input of OR gate is tied with select signal of MUX.
- If X > Y, C_0 of adder is high.
- After first subtraction, the counter output is incremented by 1.
- For each subtraction, the counter output is incremented.
- If C_0 of adder is low, then clock of counter and FF is disabled. Counting is stopped.
- Q_3 Q_2 Q_1 Q_0 is the counter output (Quotient)
- R_3 R_2 R_1 R_0 is the flipflop output (remainder)

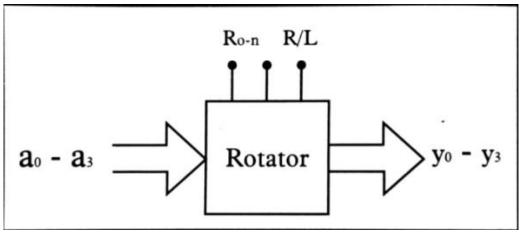


****

4.8. SHIFT REGISTERS:

Design 4 input and 4 output barrel shifter using NMOS logic. (NOV 2018, Nov 2019).
List the several commonly used shifters. Design the shifter that can perform all the commonly used shifters. [May 2021, NOV 2021]
Elaborate in detail the design of a 4-bit barrel shifter. [Nov/Dec 2022]

- An n-bit rotation is specified by using the control word R_{0-n} and L/R bit defines a left or right shifting.



- For example $y_3\ y_2\ y_1\ y_0 = a_3\ a_2\ a_1\ a_0$

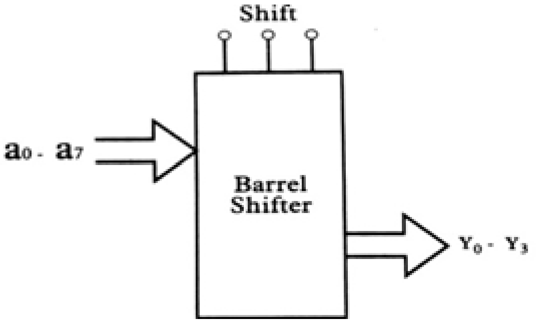
  If it is rotated 1-bit in left side, we get $y_3\ y_2\ y_1\ y_0 = a_2\ a_1\ a_0\mathbf{a_3}$

  If it is rotated 1-bit in right side, we get $y_3\ y_2\ y_1\ y_0 = \mathbf{a_0}\ a_3\ a_2\ a_1$

Barrel Shifter:

- A barrel shifter is a digital circuit that can shift a data word by a specified number of bits in oneclock cycle.

- It can be implemented as a sequence of multiplexers(MUX), and in such an implementation the output of one MUX is connected to the input of the next MUX in a way that depends on the shift distance.
- For example, take a four-bit barrel shifter, with inputs A, B, C and D. The shifter can cycle the order of the bits ABCD as DABC, CDAB, or BCDA; in this case, no bits are lost.
- That is, it can shift all of the outputs up to three positions to the right (thus make any cyclic combination of A, B, C and D).
- The barrel shifter has a variety of applications, including being a useful component in microprocessors (alongside the ALU).



Shift	b_0	b_1	b_2	b_3
0	a_0	a_1	a_2	a_3
1	a_1	a_2	a_3	a_4
2	a_2	a_3	a_4	a_5
3	a_3	a_4	a_5	a_6
4	a_4	a_5	a_6	a_7

Figure: 8 X 4 barrel shifter

- General symbol for barrel shifter is shown in figure. The outputs are given as y_3 y_2 y_1 y_0. S_0, S_1, S_2,S_3 are known as shift lines.
- A barrel shifter is often implemented as a cascade of parallel 2×1 multiplexers.
- For a 8-bit barrel shifter, two intermediate signals are used which shifts by four and two bits, or passes the same data, based on the value of S[2] and S[1].
- This signal is then shifted by another multiplexer, which is controlled by S[0].
- A common usage of a barrel shifter is in the hardware implementation of **floating-point arithmetic**.

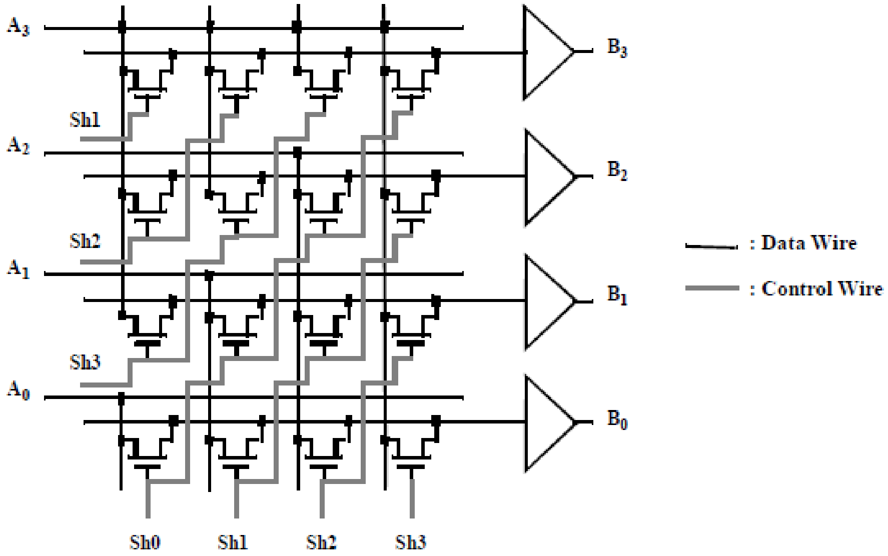
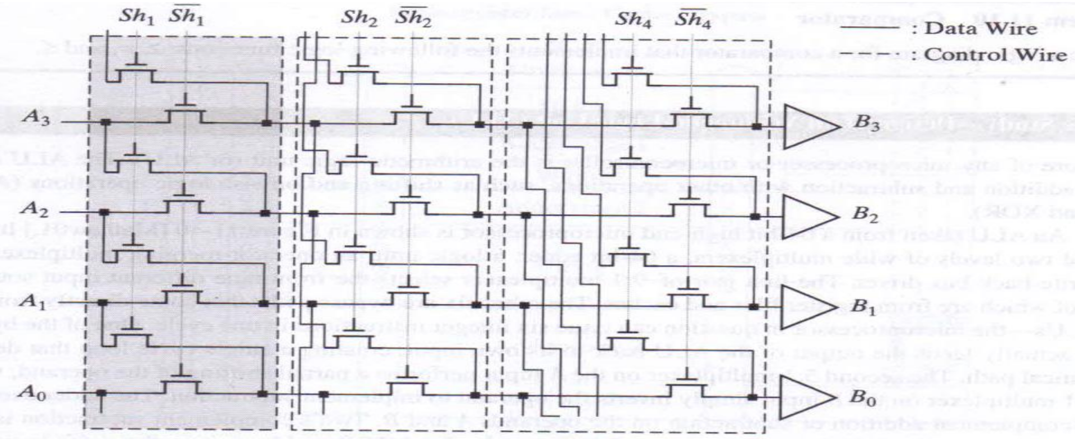


Figure: Barrel Shifter

- For a floating-point add or subtract operation, requires shifting the smaller number to the right, increasing its exponent, until it matches the exponent of the larger number.
- This is done by using the barrel shifter to shift the smaller number to the right by the difference, in one cycle.
- If a simple shifter were used, shifting by n bit positions would require n clock cycles.
- The disadvantages of FET array barrel shifter are the threshold voltage drop problem, parasitic limited switching time problem.
- The figure shown is known as a barrel shifter and a 8 x 4-bit barrel shifter circuit.

Logarithmic Shifter:

- A Shifter with a maximum shift width of M consists of a $\log_2 M$ stages, where the i^{th} stage either shifts over 2^i or passes the data unchanged.
- Maximum shift value of seven bits is shown in figure, to shift over five bits, the first stage is set to shift mode, the second to pass mode and the last again to shift.
- The speed of the logarithmic shifter depends on the shift width in a logarithmic wa, M-bit shifter requires $\log_2 M$ stages.
- The series connection of pass transistors slows the shifter down for larger shift values.
- Advantage of logarithmic shifter is more effective for larger shift values in terms of both area and speed.



****

4.9. SPEED AND AREA TRADE OFF:

Discuss the details about speed and area trade off. (May 2017)
Discuss trade-off between speed Vs area. [Nov/Dec 2022]

Adder:

- The tradeoff in terms of power and performance is shown below.
- The performance is represented in terms of the delay(speed).
- The area estimations for each of the delays are given based on the fact that area is in relation to the power consumption.
- The area of a carry lookahead adder is larger than the area of a ripple carry for a particular delay.

- This is because the computations performed in a carry lookahead adder are parallel, which requires a larger number of gates and also results in a larger area.

CLA –Carry Lookahead Adder,   RC, R – Ripple carry adder

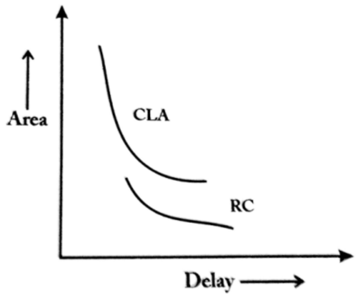


Figure: Area Vs Delay for 8 bit adder

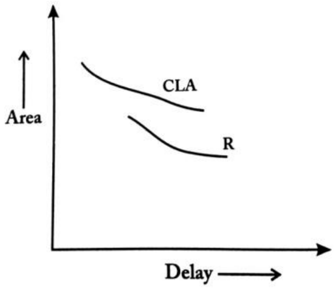


Figure: Area Vs Delay for 16 bit adder

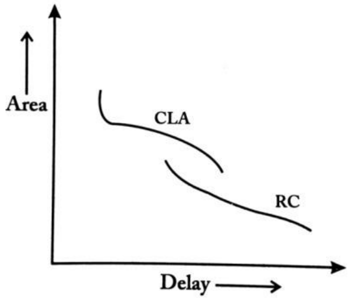


Figure: Area Vs delay for 32 bit adder

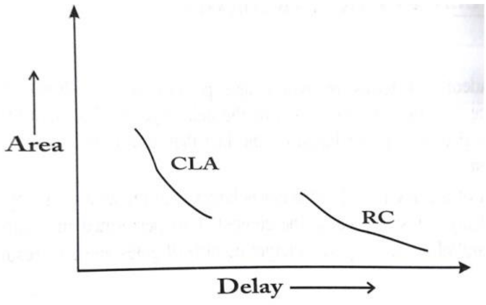


Figure: Area Vs delay for 64 bit adder

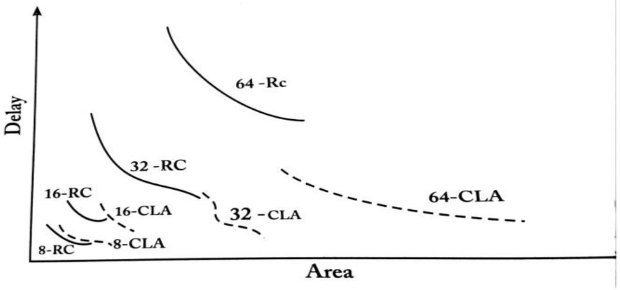


Figure: Delay Vs Area for all adders

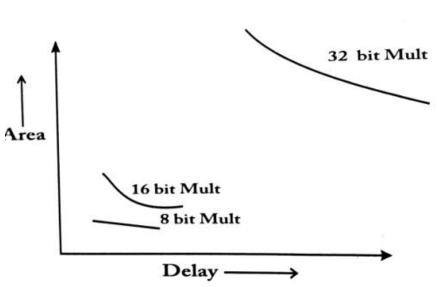
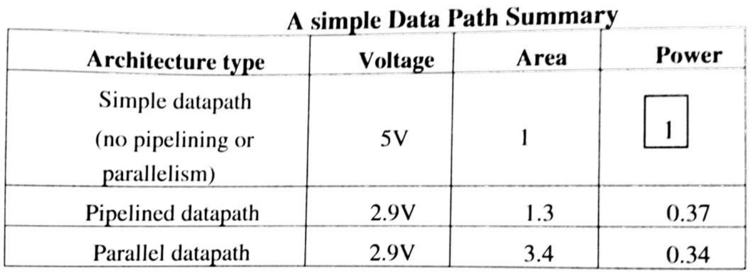


Figure: Area Vs Delay for all multiplier

- Above figures shows that the delay of the ripple carry adder increases much faster when compared to the carry lookahead adder as the number of bits is increased.
- In the carry lookahead adder, the cost is in terms of the area because computations are in parallel, and therefore more power is consumed for a specific delay.



A simple Data Path Summary

Architecture type	Voltage	Area	Power
Simple datapath (no pipelining or parallelism)	5V	1	1
Pipelined datapath	2.9V	1.3	0.37
Parallel datapath	2.9V	3.4	0.34

4.10 Memory Architecture and Memory Control Circuits:

Discuss Memory classification and its architecture and building blocks.

4.10.1 Memory Classification:

- Parameters used to characterize a memory device are area, power and speed.
- **Area:** area is important for its physical implementations by VLSI technology. Smaller the area per bit and more devices can be accommodated. So cost per bit is reduced.
- **Speed:** speed of operation plays a very important role. Memory can communicate at speed with processors.
- **Power:** Power is important, because MOS memories are used in many battery operated portable systems. Power dissipation of memory plays an important role. Memory devices will consume less power.

Classification based on operation mode:

1. ROM
2. RAM

Classification based on data storage mode:

It means on how it is stored and how long it remains there.

1. Volatile
2. Non-volatile

- Volatile memory devices will store information, as long as power is it.
- As soon as power is turned off, information is lost. Static RAM and dynamic RAM belong to the category of volatile memory.
- EPROM and mask programmable ROM are non-volatile memory devices.
- If the power is turned off information will not be lost.

Classification based on access method:

1. Random access
2. Non-random access

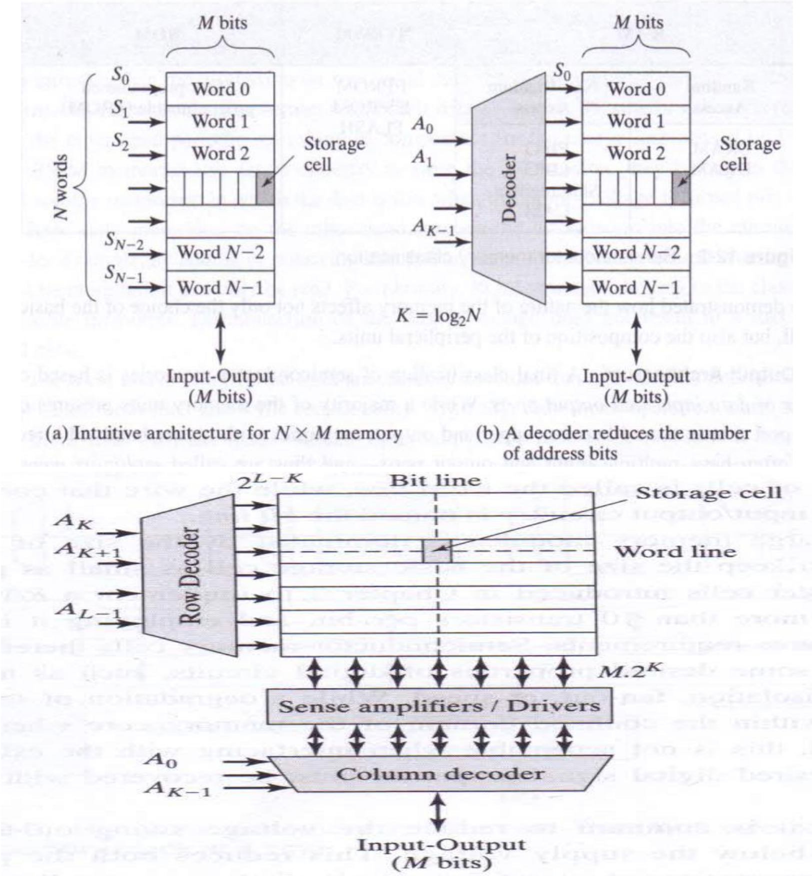
4.10.2 Memory Architecture and Building Blocks:

Explain the *memory architecture* and its control circuits in detail. (April 2018)

Illustrate the building blocks of Memory architectures and memory peripheral circuitry adapted to operate for non-volatile memory. [May 2021]

When n x m memory is implemented, then, n memory words are arranged in a linear fashion. One word will be selected at a time by using select line.

- If we want to implement the memory 8X8, n=8, m=8(number of bits).
- Then we need 8 select signals (one for each word).
- But by using decoder we can reduce the number of select signals.
- In case of 3 to 8 decoder, if 3 inputs are given to decoder, then we can get 8 select signals.
- If n=220, then we can give only 20 inputs to the decoder.



(a) Intuitive architecture for N × M memory

(b) A decoder reduces the number of address bits

Figure: Array structured memory organization

- If basic storage cell size is approximately square, then the design is extremely slow. The vertical wire, which connects the storage cells to I/O will be excessively large.
- So, memory arrays are organized in such a way that vertical and horizontal dimensions are the same.
- The words are stored in a row. These words are selected simultaneously.
- The **column decoder** is used to route the correct word to the I/O terminals.
- The row address is used to select one row of memory and column address is used to select particular word from that selected row.
- **Word line**: The horizontal select line which is used to select the single row of cell is known as word line.
- **Bit line**: The wire which connects the cell in a single column to the input/output circuit is known as bit line.
- **Sense amplifier**: It requiresanamplificationoftheinternalswingtofullrail-to-rail amplitude.
- **Block address**: the memory is divided into various small blocks.
- The address which is used to select one of the small blocks to be read or written is known as block address.
- **Advantages:**

  1. Access time is fast

2. Power saving is good, because blocks not activated are in power saving mode.

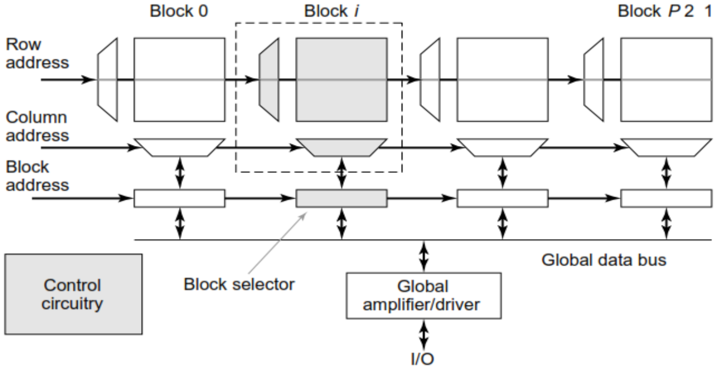


Figure: Hierarchical memory architecture

4.10.3 Memory Core

Discuss Memory core its types in detail.

4.10.3.1 Read Only Memory (ROM):

ROM is a memory where code is written only one time.

Diode ROM:

- It is simple where presence of diode in between bit line and word line is considered as logic 1 and absence of diode as logic 0.
- Disadvantage is used for small memories and no isolation between word line and bit line.

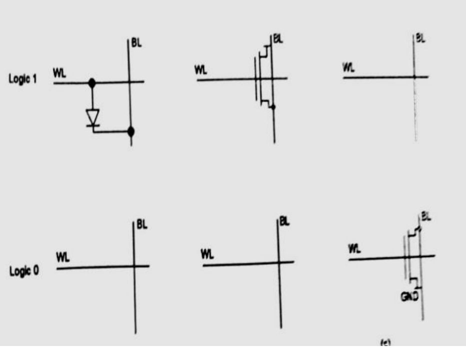
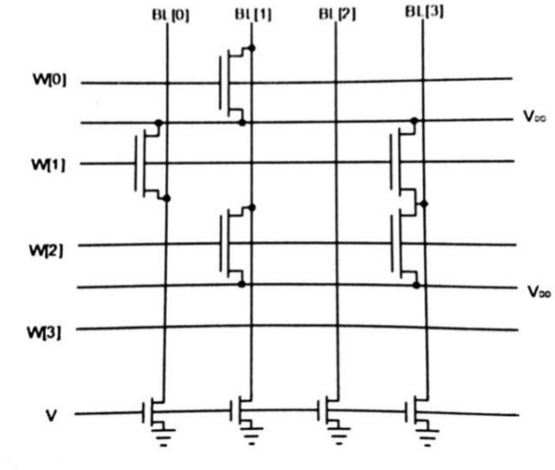


Figure: Diode ROM

MOS ROM:

Q: Draw the NOR and NAND implementation of 4-word, 4-bit ROM. (NOV 2021)

- Diode is replaced by gate source connection of nMOS. Drain is connected to V_{DD}.
- The charging and discharging of word line capacitance has been taken care by the word line driver.
- Absence of a transistor between word line and bit line means logic 1 is stored and if presence then logic 0 is stored.



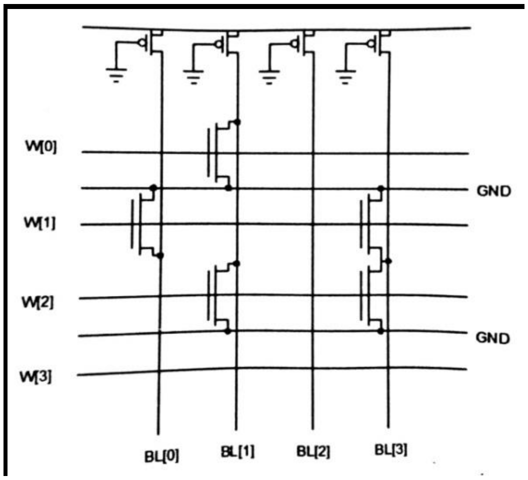


Figure: 4 x 4OR ROM cell array

Figure: 4 x 4 MOS NOR ROM

Programming ROM

- The transistor in the intersection of row and column is OFF when the associated word line is LOW. In this condition, we get logic 1 output.

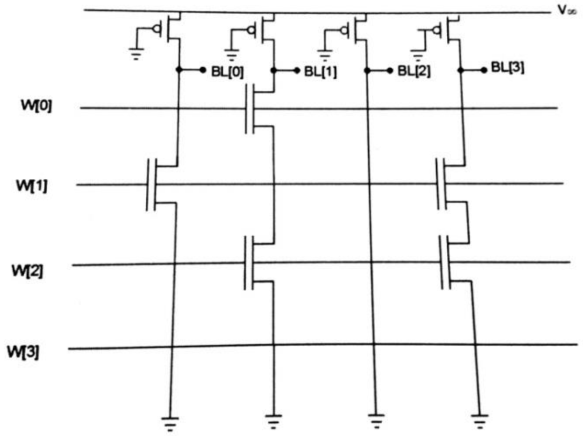


Figure: 4 x 4 MOS NAND ROM

Advantage: basic cell only consists of transistor. No need of connection to any of the supply voltage.

Disadvantage: As it has pseudo nMOS, it is ratioed logic and consumes static power.

To overcome this, precharged MOS NOR ROM logic circuit is used.

- This eliminate static dissipation ratioed logic requirement.

4.10.3.2 Non-Volatile READ-WRITE Memory:

- It consists of array of transistors. We can write the program by enabling or disabling these devices selectively.
- To reprogram, the programmed values to be erased, then the new programming is started.

Floating gate transistor:

- It is mostly used in all the reprogrammable memories.

- In floating gate transistor, extra polysilicon strip is used in between the gate and the channel known as floating gate.
- Floating gate doubles the gate oxides thickness and hence device transconductance is reduced and threshold voltage is increased.
- The threshold voltage is a programmable.
- If high voltage is (>10V) is applied between the source terminals and gate-drain terminals, then high electric field is generated. So, avalanche injection occurs.
- After acquiring energy, electron becomes hot and transverse through the first oxide insulator . They get trapped on the floated gate.
- The floating gate transistor is known as floating gate avalanche injection MOS or FAMOS.

Disadvantage: High programming voltage is need.

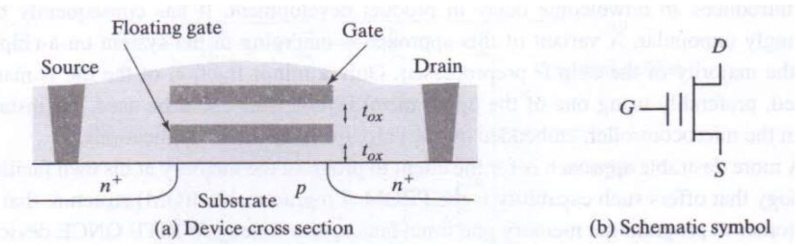


Figure (a) floating gate transistor (b) symbol

EPROM – Erasable Programmable Read Only Memory:

- Erasing is done by passing UV rays on the cell by using transparent window.
- This process will take some seconds to some minutes.
- It depends on intensity of UV source. The programming takes 5-10microseconds/word.
- During programming, chip is removed from the board and placed in EPROM programmer.

Advantages: simple and large families are fabricated with low cost.

Disadvantages:

- Number of erase/program cycle is limited upto 1000.
- Reliability is not good.
- Threshold voltage of the device may be varied with repeated program.

EEPROM – E^{2}PROM:

- Electrically Erasable Programmable ROM. Here Floating gate tunneling oxide (FLOTOX) is used.
- It is similar to floating gate except that the portion of the floating gate is separated from the channel at the thickness of 10nm or <10nm.
- If 10V is applied, electron travels to and from the floating gate through Fowler-Nordheim tunneling.
- Erasing can be done by revering applied voltage which is used for writing.

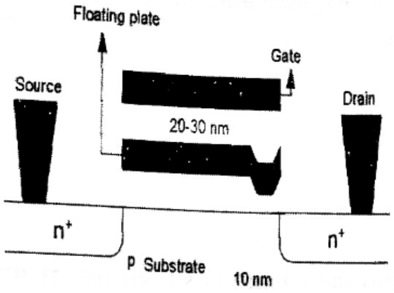


Figure: FLOTOX transistor

Advantage: High versatility and possible for 105 erase/write cycle.

Disadvantages: Larger than FAMOS transistor, Costly, Repeated programming causes a drift in threshold voltage.

Flash Memory – Flash Electrically Erasable Programmable ROM

- It is a combination of density of EPROM and versatility of EEPROM.
- Avalanche hot electron injection mechanism is used.
- Erasing can be done by Fowler-Nordheim tunneling concept. Here erasing is done in bulk.

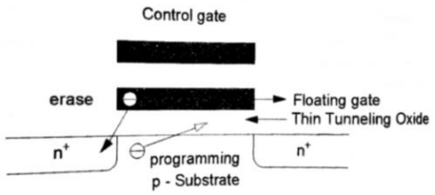


Figure: ETOX device

- It is similar to FAMOS gate.
- A very thin tunneling oxide layer (10nm thickness) is there.
- ***Erasing operation:*** Erasing can be performed when gate is connected to the ground and the source is connected to 12V.
- ***Write operation:*** High voltage pulse is applied to the gate of the selected device. Logic 1 is applied to the drain and hot electrons are injected into the floating gate.
- ***Read operation:***To select a cell, its word line is connected to 5V. It causes conditional discharge of the bit line.

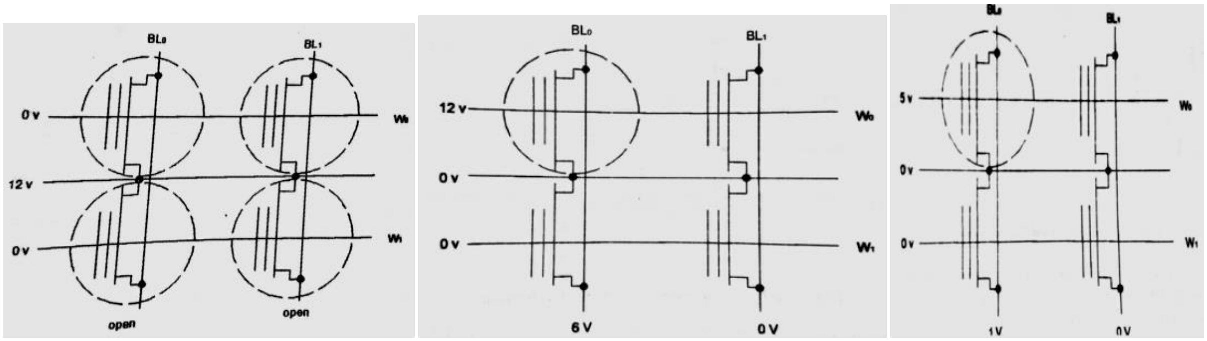


Figure: (a) Erase (b) Write (c) Read operation of NOR flash memory

4.10.3.3 RAM – Random Access Memory

Explain about static and dynamic RAM.
Construct 6T based SRAM cell. Explain its read and write operations. (NOV 2018) [Nov/Dec 2022]

4.10.3.3.1Static RAM:

- SRAM cell needs 6 transistors per bit.
- M5 and M6 transistors are shared between read and write operations.
- Bit line(BL) and inverse Bit Line signals are used to improve the noise margin during read and write operations.

***Read operation**:*

- Let us assume logic 1 is stored at Q and BL and inverse BL are precharge to 2.5V before starting read operation.
- The read cycle is started by asserting word line then M_5 and M_6 transistors are enabled.
- After the small initial word line delay then the values stored at Q and inverse Q are transferred to the bit lines by leaving BL at 2.5V and the value at inverse Q is discharge through M_1, M_5.

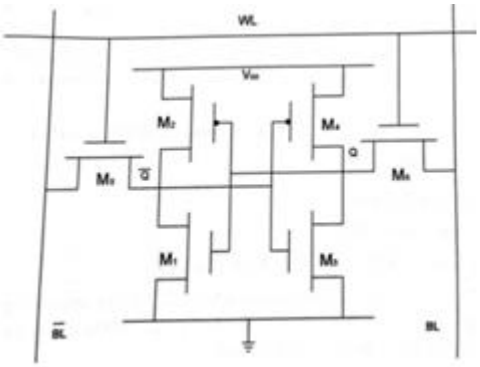


Figure: CMOS SRAM cell

***Write operation**:*

- Assume that Q=1, now logical 0 is to be written in the cell.
- Then inverse BL is set to 1 and BL is set to 0.
- The gate of M_1 is at V_{DD} and gate of M_4 is at ground as long as the switching is not commenced.
- Inverse Q is not pulled high enough to ensure the writing of logic 1.
- Cell voltage is kept below 0.4V. The new value of the cell is written through M_6.

4.10.3.3.2Dynamic RAM:

Three transistors DRAM

- Content in the cell can be periodically rewritten through a resistive load, called as refresh operation.
- This refresh occurs for every 1-4ms. Dynamic memory has refresh operation.
- For example, logic 1 is to be written, and then BL1 is asserted high and write wordline(WWL) is asserted.
- This data is retained as charge on the capacitor once WWL is low.

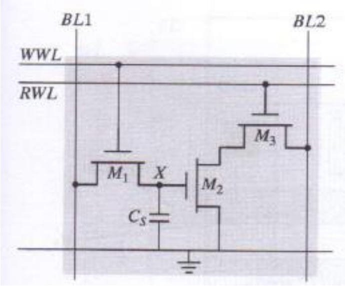


Figure: Three transistor dynamic memory cell

- To read the cell, the read word line (RWL) is raised. M_2 transistor is either ON or OFF depends upon the stored value.
- BL2 bit line is connected to V_{DD} or it is precharged to V_{DD} or V_{DD}-V_t.
- When logic 1 is stored, the series combination of $M_2$and M_3 pulls BL2 line low.
- If logic 0 is stored, then BL2 line is high.
- To refresh the cell, first the stored data is read, and its inverse is placed on BL1 and WWL line is asserted.

One transistor DRAM:

- In this cell, to write logic 1 then it is placed on bit line and word line is asserted high.
- The capacitor is charged or discharged depending upon the data. Before performing read operation, bit line is precharged.

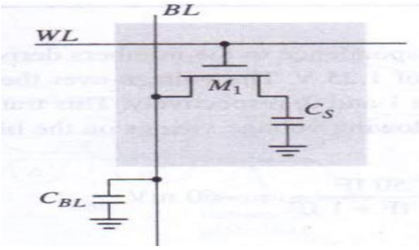


Figure: One transistor DRAM

4.10.3.3.3 CAM – Content Addressable or Associate Memory

Explain about CAM.

- It supports 3 operating modes,
    - Read
    - Write
    - Match
- In this memory, it is possible to compare all the stored data in parallel with the incoming data. It is not power efficient.
- Figure shows apossible implementation ofaCAM array.
- Thecellcombines atraditional 6TRAM storage cell *(M_4-M_9)*withadditional circuitry toperform al-bit digital comparison *(M_1-M_3)*.
- When thecellistobe written,complementarydataisforced ontothebitlines, whiletheword lineisenabled asinastandardSRAM cell.

- In the compare mode, stored data are compared using bit line. The match line is connected to all CAM blocks in a row. And it is initially precharged to V_{DD}.
- If there is some match occurs, then internal row is discharged. If even one bit in a row is mismatched, then the match line is low.

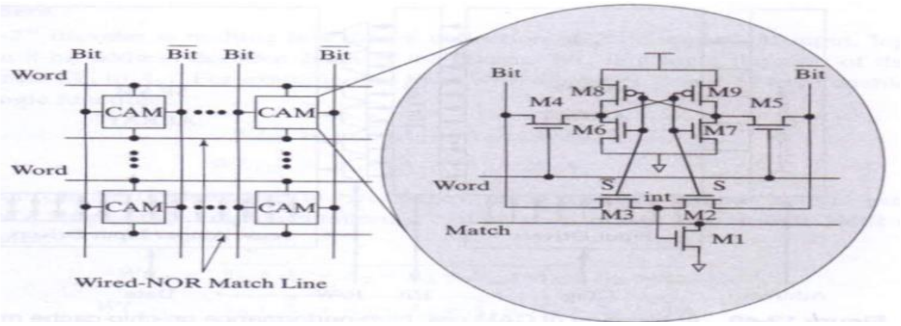


Figure: CAM cell

****

4.11 Memory peripheral (control) Circuits:

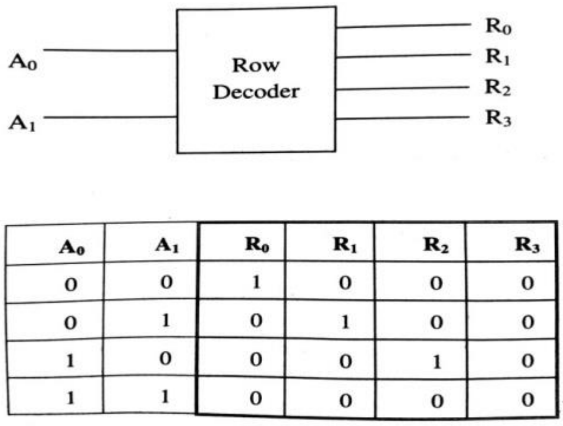
Explain the memory architecture and its *<u>control circuits</u>* in detail. (April 2018)

Illustrate the building blocks of Memory architectures and <u>memory peripheral circuitry</u> adapted to operate for non-volatile memory. [May 2021]

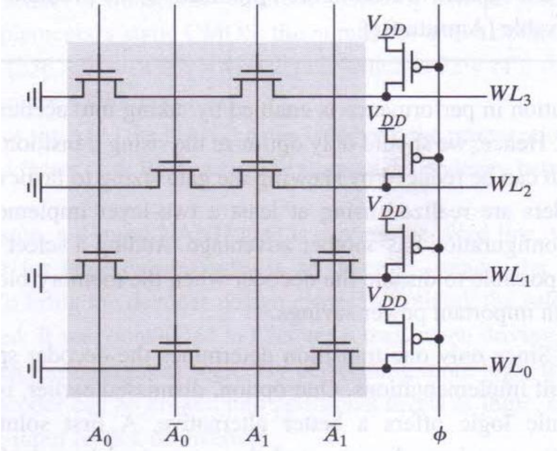
(i) Address & Block Decoders:

Row Decoder:

- Row and column address decoder are used to select the particular memory location in an array.
- Row decoder is used to drive NOR ROM array. It selects one of 2^n word lines.
- Dynamic 2 to 4 decoder reduces the number of transistors and propagation delay.



A_0	A_1	R_0	R_1	R_2	R_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	0



Symbol and Truth table

Dynamic2-to-4NORdecoder

Column Decoder

- It should matchthebitlinepitchofthememory array.

- In column decoder, decoder outputs are connected to nMOS pass transistors.
- By using this circuit, we can selectively drive one out of m pass transistors.
- Only one nMOS pass transistor is ON at the time.

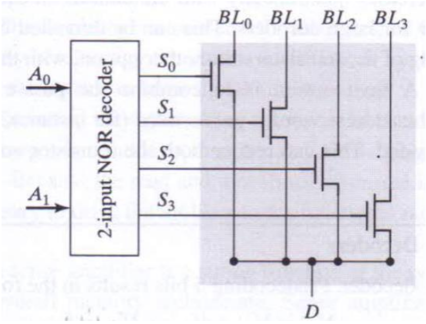
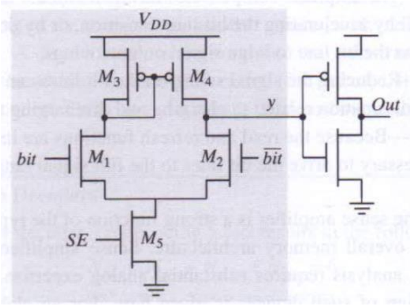


Figure: Four-input pass-transistor-based column decoder using a NOR predecoder

(ii) Sense Amplifier

- Sense amplifiers play a major role in the functionality, performance and reliability of memory circuits.
- Basic differential sense amplifier circuit shown in below figure.
- It performs the following performances



Amplification:

- In memory structures such as the 1TDRAM, amplification is required for proper functionality.

Delay Reduction:

- The amplifier compensates for the fan-out driving capability of the memory cell by detecting and amplifying small transitions on the bit line to large signal output swings.

Power reduction:

- Reducing the signal swing on the bit lines can eliminate large part of the power dissipation related to charging and discharging the bit lines.

(iii) Drivers/ Buffers

- The length of word and bit lines increases with increasing memory sizes.
- Large portion of the read and write access time can be attributed to the wire delays.
- A major part of the memory-periphery area is allocated to the drivers (address buffers and I/O drivers).

****

4.12: Low Power Memory design:

- ❖ **Discuss about Low power memory design. [Apr/May 2022]**
- ❖ **Elucidate in detail low power SRAM circuit. (April 2019-13M) (Nov 2019)**

(i) Active Power Reduction:

- Voltage reduction done by either an increase in the size of the storage capacitor and/or a noise reduction.

Techniques for power reductions:

- Half- V_{DD} precharge:
  - ▪ Precharging a bit line to $V_{DD}/2$. It helps to reduce the active power dissipation in DRAM memories by a factor of 2.
- Boosted word line:
  - ▪ Raising the value of the word line above V_{DD} during a write operation, eliminates the threshold drop over the access transistor, yielding a substantial increase in stored charge.
- Increased capacitor area or value:
  - ▪ Keeping the "ground" plate of the storage capacitor at $V_{DD/}2$ reduces the maximum voltage over Cs, making it possible to use thinner oxides.
- Increasing the cell size:
  - ▪Ultra-low-voltage DRAM memory operation might requirea sacrifice in area efficiency.

(ii) Retention current Reduction:

- SRAM array should not have any static power dissipation. But the leakage current of the transistor will be the major problem and this is the main source of the retention current.
- This retention current can be reduced by the following factors.

  1. Turn OFF unused memory blocks
  2. Negative biasing voltage of the cells which are not active, thus reduce the leakage current.
  3. If low threshold voltage transistor is inserted between V_{DD} and SRAM array, leakage reduces.
  4. Leakage is a function of V_{DD}, thus if supply rail is lowered, then leakage current is reduced.

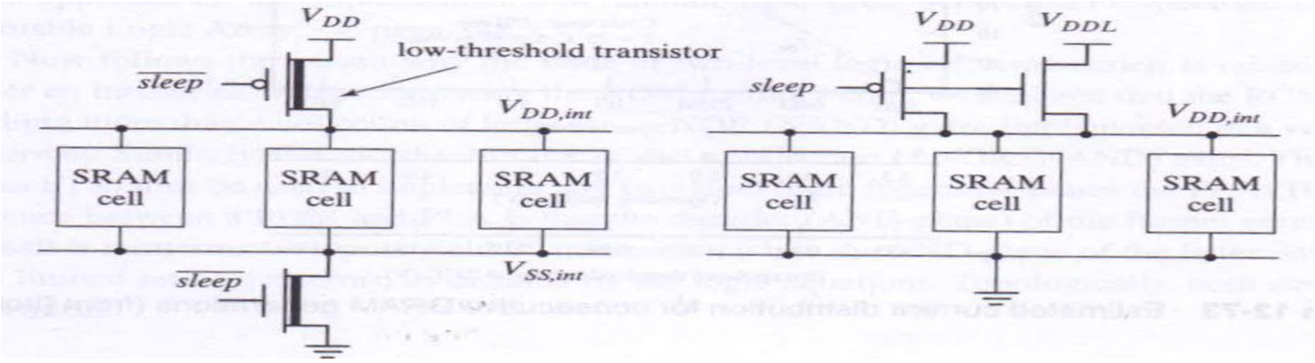


Figure: (a) Insertion of low threshold device (b) Reducing supply Voltage

****

Programmable devices (Programmable ASIC):

- Programmable devices can be divided into three areas
  1. Programmable logic structure
  2. Programmable interconnect
  3. Reprogrammable Gate array
- A programmable logic device (PLD) is an electronic component used to build reconfigurable digital circuits.
- Unlike a logic gate, which has a fixed function, a PLD has an undefined function at the time of manufacture.

1. Programmable Logic Structure:

❖ **Describe in detail the chip with programmable logic structures. (Nov 2009)**

(a) Programmable Logic Array:

- Programmable logic arrays (PLAs) is a type of fixed architecture logic devices with *programmable AND gates followed by programmable OR array.*
- Logic array is the structure unit which can be programmed to perform various functions.
- Programmable Logic Array (PLA) can be implemented as AND-OR plane devices.
- Structure of AND-OR PLA is shown below.

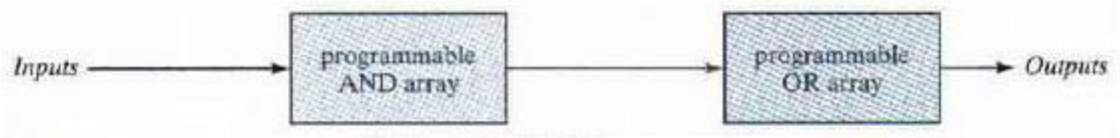


Figure: Programmable logic array

- PLA is used to implement a complex combinational circuit.
- The AND and OR gates inside the PLA are initially fabricated with fuses among them.
- The specific Boolean functions are implemented in sum of products (SOP) form by blowing appropriate fuses and leaving the desired connections.
- For an example, the Boolean expressions are,

$$F_1 = A\bar{B} + AC + \bar{A}B\bar{C}$$

$$F_2 = \overline{(AC + BC)}$$

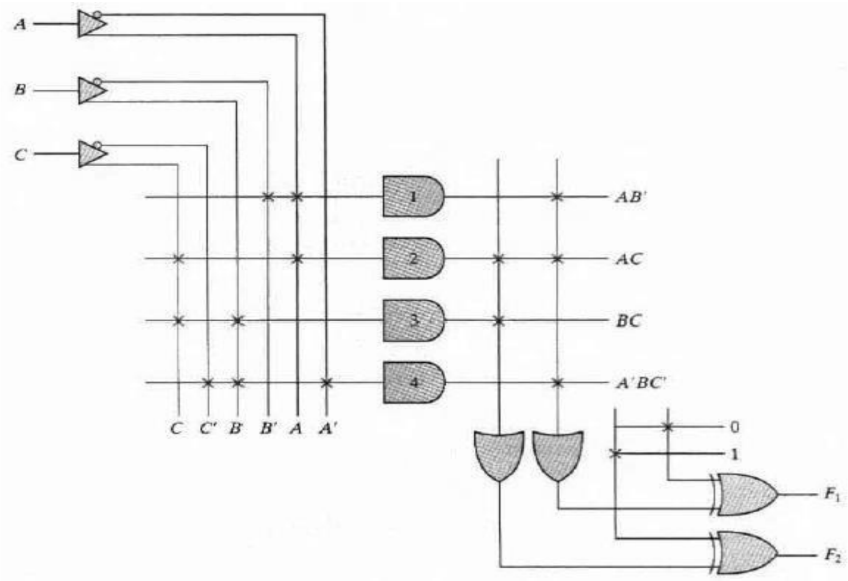


Figure: PLA with three inputs, four product terms and two outputs

(b) PAL (Programmable Array Logic) Architecture:

- The PAL is a programmable logic device with a *fixed OR array and a programmable AND* array.

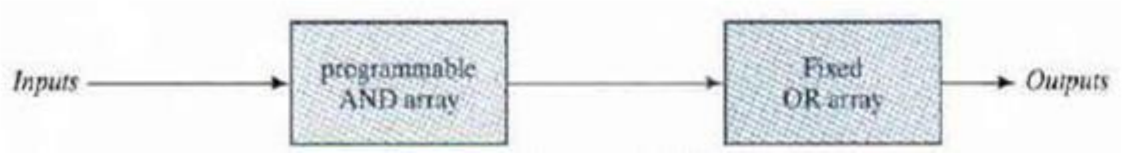


Figure: Programmable Array Logic

- Because only the AND gates are programmable, the PAL is easier to program than but is not as flexible as the PLA.
- The PAL is a programmable logic device with a fixed OR array and a programmable AND array.

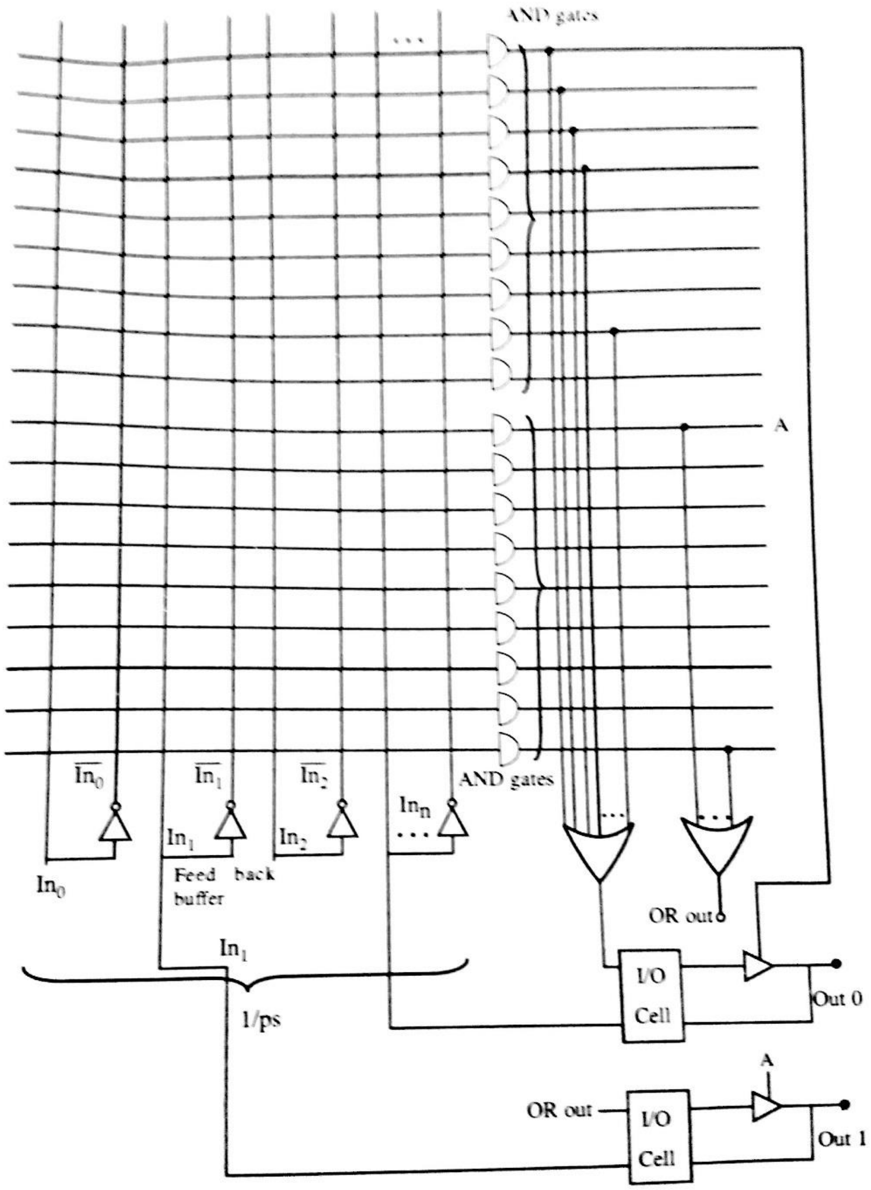


Figure: Example of PAL circuit

Reprogrammable Gate array:

- A field programmable gate array (FPGA) is a VLSI circuit that can be programmed at the user’s location.
- A typical FPGA consists of an array of millions of logic blocks, surrounded by programmable input and output blocks and connected together via programmable interconnections.
- There is a wide variety of internal configurations within this group of devices.
- The performance of each type of device depends on the circuit contained in its logic blocks and the efficiency of its programmed interconnections.

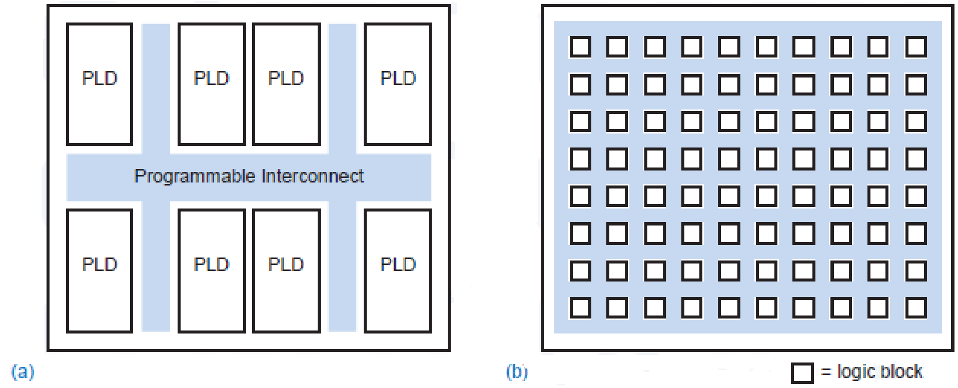


Figure: Programmable-logic-device approaches: (a) CPLD (b) FPGA.

- A typical FPGA logic block consists of lookup tables, multiplexers, gates, and flip-flops.
- A lookup table is a truth table stored in an SRAM and provides the combinational circuit functions for the logic block.
- The combinational logic section, along with a number of programmable multiplexers, is used to configure the input equations for the flip-flop and the output of the logic block.
- The *advantage* of using RAM instead of ROM to store the truth table is that the table can be programmed by writing into memory.
- The *disadvantage* is that the memory is volatile and presents the need for the lookup table's content to be reloaded in the event that power is disrupted.
- The program can be downloaded either from a host computer or from an onboard PROM.
- The program remains in SRAM until the FPGA is reprogrammed or the power is turned off. The device must be reprogrammed every time power is turned on.

****

<u>Programming technology used in FPGA :</u>

Discuss the different types of programming technology used in FPGA design. (NOV 2016)

- There are three types of programming technology.
  - ✓ Fusible link programming (Anti fuse)
  - ✓ SRAM Programming
  - ✓ EPROM and EEPROM programming

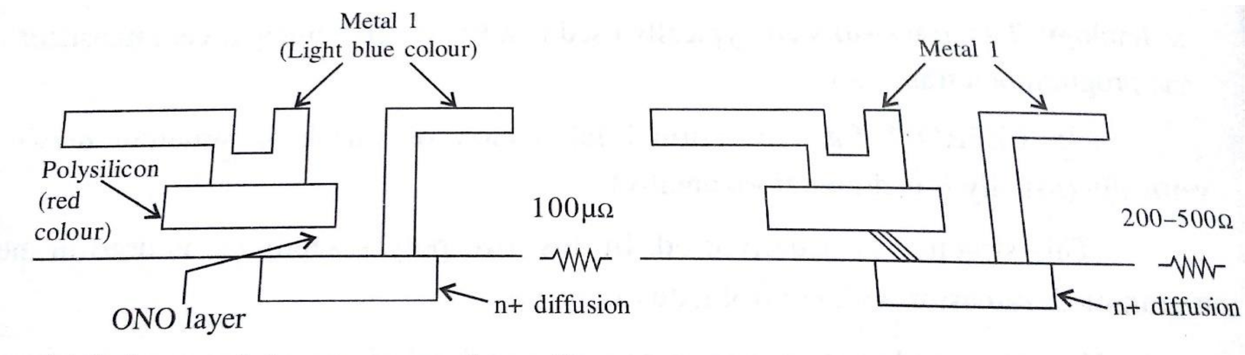
5.6.1: Fusible link programming:

- In this type, Platinum, Titanium tungsten is used to form link.
- It is blown when certain current is exceeded in the fuse. Higher voltage is applied to device to blow the fuses.

Draw and explain the operation of metal-metal antifuse and EPROM transistor. (June 2012)

ANTIFUSE:

- In FPGA, the device is programmed by changing the characteristic of switching element (or) we can write the program for routing.
- Programming routing can be explained by using the product of ACTEL, Quick Logic Companies etc.
- In ACTEL, interconnect is done by PLICE (or) Antifuse.
- PLICE means Programmable Low Impedance Circuit Element.
- Antifuse is high resistance (>100MΩ) is changed into low resistance (200-500Ω) by applying programming voltage.
- It consists of ONO (Oxide-Nitride-Oxide) layer which is sandwiched between polysilicon layer and n+ diffusion.
- Antifuses separate interconnect wires on the FPGA chip and the programmer blows an antifuse to make a permanent connection.
- Once an antifuse is programmed, the process can't be reversed. This is an **OTP Technology.**

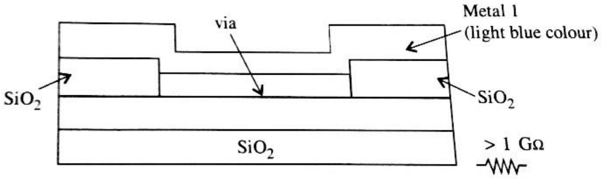


(a) Before applying programming voltage

(b) After applying programming voltag

In-system programming (ISP):

- Possibility to program the chip after it has been assembled on the PCB.
- In Quick logic company, programmable interconnect is provided with Vialink (metal-metal anti-fuse).



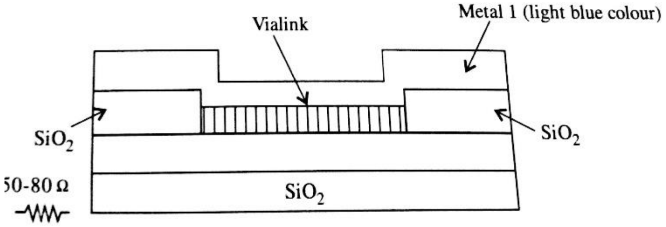


Figure: Metal-metal anti-fuse

Advantages of metal-metal antifuse:

- Advantages of metal-metal antifuse over poly diffusion antifuse are:
    1. The connections are direct to metal wiring layers.
    2. It is easier to use larger programming currents to reduce the antifuse resistance.

UV-Erasable programming:

Find the reason for referring EPROM technology as floating gate avalanche MOS. (Dec. 2013)

EPROM programming:

- In this type floating gate transistor is used.
- We can reprogram by using UV-light.
- High electric field causes electrons flowing towards drain to move across the insulating gate oxide, where they trapped on the bottom, floating gate.
- These energetic electrons are HOT and this effect is known as Hot-electron injection (or) avalanche injection.
- EPROM technology is sometimes called floating –gate avalanche MOS (FAMOS).

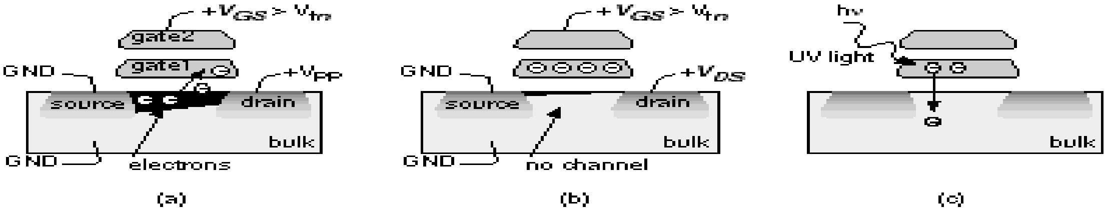


Figure: EPROM transistor

- (a) With a high (>12V) programming voltage, V_{PP} applied to the drain. Electrons gain enough energy to jump onto the floating gate (gate1).
- (b) Electrons stuck on gate1 raise the threshold voltage so that the transistor is always off for normal operating voltages.
- (c) Ultraviolet light provides enough energy for electrons stuck on gate1 to jump back to the bulk, allowing the transistor to operate normally.

EEPROM programming:

- Electrically Erasable programming is most popular CMOS technology.
- A very thin oxide between floating gate and the drain allow the electrons to tunnel to or from the floating gate (gate is charged or discharged).
- Thus enabling writing and erasing operation.

Advantages:

- The advantages of EEPROM technology are:
    - ✓ faster than using a UV lamp
    - ✓ chips do not have to be removed from the system
    - ✓ if the system contains circuits to generate both program and erase voltages, it may use ISP

SRAM Programming

- SRAM programming is shown in figure.
- SRAM configuration cell is constructed from two cross-coupled inverters and uses a standard CMOS process.
- The configuration cell drives the gates of other transistors on the chip (using pass transistors or transmission gates) to make a connection or off to break a connection.
- The cell is programmed using the WRITE and DATA lines.

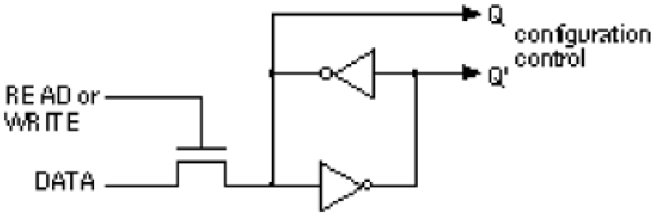


Figure: SRAM programming

Advantages:

- Designers can reuse chips during prototyping.
- Designers can update or change a system on the fly in reconfigurable hardware.

Disadvantage:

- Need to keep power supply for retaining the connection information.

****

- ❖ **Explain the reprogrammable device architecture with neat diagrams.**
- ❖ **With neat diagram explain the functional blocks in PDA (Programmable Device Architecture). (AU:June 2015, June 2016)**
- ❖ **With neat sketch explain the CLB, IOB and Programmable interconnects of an FPGA device. (May 2016)**
- ❖ **Explain about building block architecture of FPGA. (April 2017, 2018, NOV 2018)**
- ❖ **Elucidate in detail the basic FPGA architecture. (April 2019-13M)**
- ❖ **Describe in detail FPGA architecture and explain the main building blocks of FPGA. (Nov 2019)[Nov/Dec 2022]**
- ❖ **Illustrate the basic building block architectures of FPGA. [May 2021]**

<u>Re-Programmable Devices Architecture (FPGA)</u>

- FPGA provide the next generation in the programmable logic devices.
- It refers to the ability of the gate arrays to be programmed for a specific function by the user.
- The word Array is used to indicate a series of columns and rows of gates that can be programmed by the end user.
- As compared to standard gate arrays, the field programmable gate arrays are larger devices.

- The basic cell structure for FPGA is complicated than the basic cell structure of standard gate array.
- The programmable logic blocks of FPGA are called Configurable Logic Block (CLB).
- The FPGA architecture consists of **three types of configurable elements-**
  - (i) IOBs –Input/output blocks
  - (ii) CLBs- Configurable logic blocks
  - (iii) Resources for interconnection
- The IOBs provide a programmable interface between the internal, array of logic blocks (CLBs) and the device's external package pins.
- CLBs perform user-specified logic functions.
- The interconnect resources carry signals among the blocks.
- A configurable program stored in internal static memory cells.
- Configurable program determines the logic functions and the interconnections.
- The configurable data is loaded into the device during power-up reprogramming function.
- FPGA devices are customized by loading configuration data into internal memory cells.

The structure of FPGA:

The basic elements of the FPGA structure:

1.Logic blocks

- Based on memories *(Flip-flop & LUT – Lookup Table)* Xilinx
- Based on multiplexers *(Multiplexers)*-Actel
- Based on PAL/PLA - Altera
- Transistor Pairs

2. Interconnection Resources
   - Symmetrical FPGA-s
   - Row-based FPGA-s
   - *Sea-of-gates* type of FPGA-s
   - Hierarchical FPGA-s (*CPLD)*
3. Input-output cells (*I/O Cell)*
   - Possibilities for programming :
     a. Input
     b. Output
     c. Bidirectional

RE-PROGRAMMABLE DEVICE ARCHITECTURE:

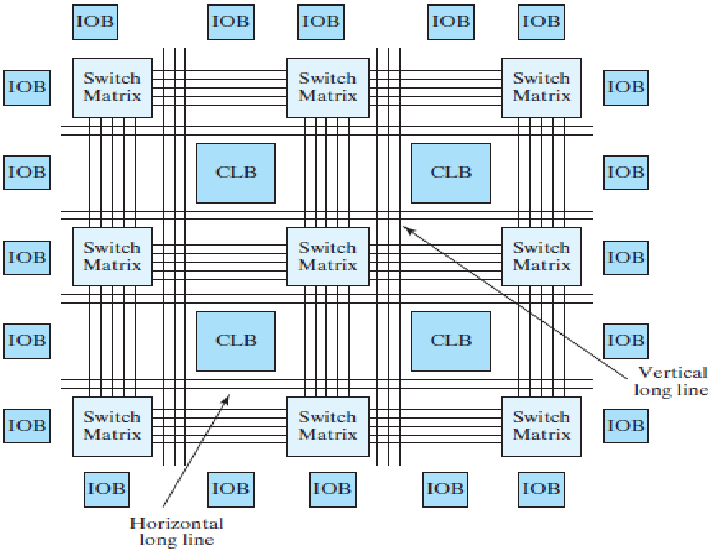
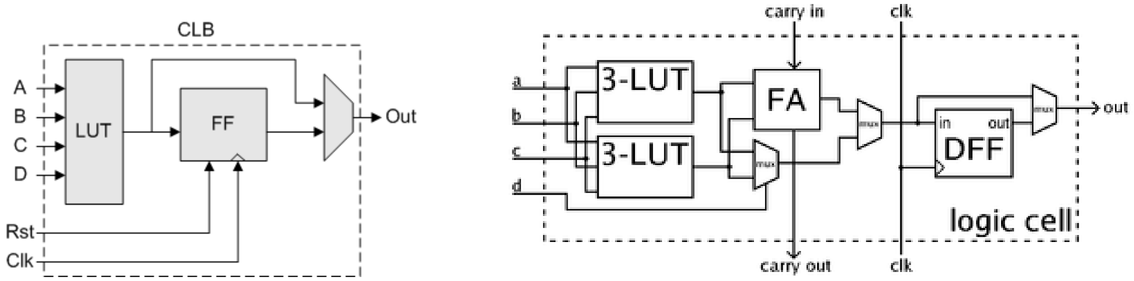


Figure: FPGA building blocks structure

- The figure shows the general structure of FPGA chip.
- It consists of a large number of programmable logic blocks surrounded by programmable I/O block.

Configurable Logic Block:



- **Figure: Various configurable Logic Block**

- The programmable logic blocks of FPGA are smaller and less capable than a PLD, but an FPGA chip contains a lot more logic blocks to make it more capable.
- As shown in figure the logic blocks are distributed across the entire chip.
- These logic blocks can be interconnected with programmable inter connections.
- The programmable logic blocks of FPGAs are called Configurable Logic Blocks (CLBs).
- CLBs contain LUT, FF, logic gates and Multiplexer to perform logic functions.
- The CLB contains RAM memory cells and can be programmed to realize any function of five variables or any two functions of four variables.
- The functions are stored in the truth table form, so the number of gates required to realize the functions is not important.

Interconnection resources:

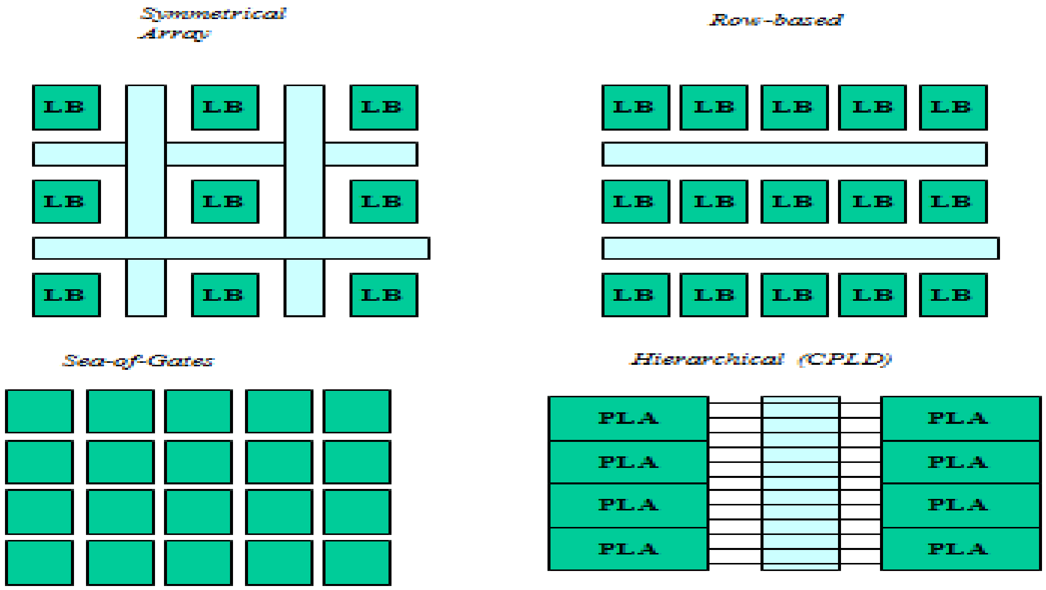


Figure: Types of interconnection resources

(a) Symmetrical Arrays

- It consists of logic elements (CLBs) arranged in rows and columns of a matrix and interconnect laid out between them.
- This symmrtrical martrix is surrounded by I/O blocks which connect it to outside world.

(b) Row based architecture:

- It consists of alternating rows of logic modules and programmable interconnect tracks.
- Input output blocks is located in the periphery of the rows.
- One row may be connected to adjacent rows via vertical interconnect.

(c) Hierarchical CPLD:

- This architecture is designed in hierarchical manner with top level containing only logic blocks and interconnects.

  1.Connections within macrocells

  2.Local connection resource within the logical block.

  3.Global connection resource *(Switch Matrix)*

(d) Sea of gates structure:

- It consists of logic elements (CLBs) arranged in rows and columns of a matrix inthe channel less gate arrays module.

I/O cells(Blocks):

- User-configurable input/output blocks (IOBs) provide the interface between external package pins and the internal logic.
- Each IOB controls one package pin and can be configured for input, output, or bidirectional signals.
- Figure shows a three-state bidirectional output buffer.

- When the output enable, OE is ‘1’ the output section is enabled and drives the I/O pad.
- When OE is ‘0’ the output buffer is placed in a high-impedance state.

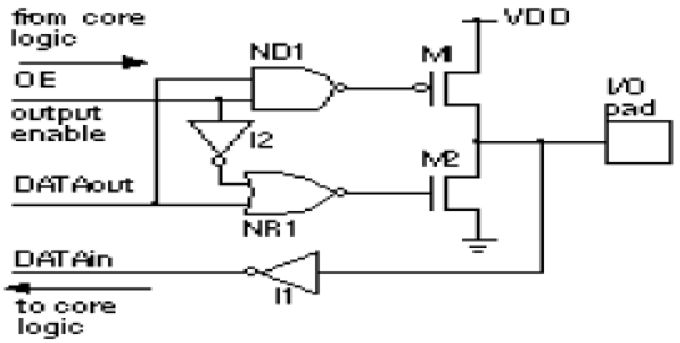


Figure: A three-state bidirectional output buffer

- We can limit the number of I/O drivers that can be attached to any one V_{DD} and GND pad.
- It allows employ the same pad for input and output bidirectional I/O.
- When we want to use the pad as an input, set OE low and take the data from DATAin.
- We can build output-only or input-only pads.

****

FPGA(*PROGRAMMABLE ASIC *)interconnect routing procedures (Architectures):

- ❖ **Give short notes on FPGA interconnect routing procedures. (May 2016, May 2021)**
- ❖ **Describe FPGA interconnect routing resources with neat diagram. (April 2019-13M)**
- ❖ **Give a note on standard cell design and FPGA interconnecting resources. (Nov 2019) [Apr/May 2022]**

- Routing architecture comprises of programmable switches and many wires.
- Routing provides connection between logic blocks, I/O blocks, and between one logic block and another logic block.
- The type of routing architecture decides area consumed by routing as well as density of logic blocks.
- Routing techniques decide the amount of area used by wire segments and programmable switches as compared to area consumed by logic blocks.

Types of FPGA interconnect routing procedures:

- ✓ Hierarchical Routing Architecture
- ✓ Island-Style Routing Architecture
- ✓ Xilinx Routing Architecture
- ✓ Altera Routing Architecture
- ✓ Actel Routing Architecture

(a) Hierarchical Routing Architecture:

- Hierarchical routing architectures separates FPGA logic blocks into distinct groups.

- Connections between the logic blocks within a group can be made using wire segments at the lowest level of the routing hierarchy.
- Connections between the logic blocks in distant groups require the traversal of one or more levels of routing segments.
- As shown in Figure, only one level of routing directly connects to the logic blocks.
- Programmable connections are represented with the crosses and circles.

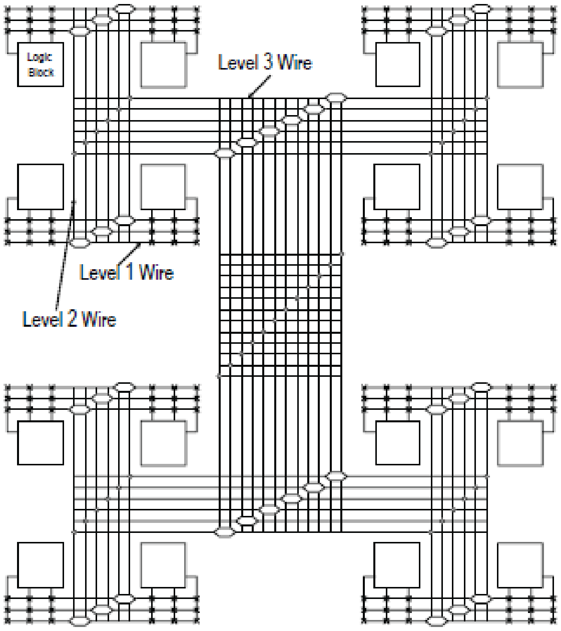


Figure: Example of Hierarchical FPGA

(b) Xilinx Routing Architecture:

- In Xilinx routing, connections are made from logic block into the channel through a connection block.
- As SRAM technology is used to implement Lookup Tables, connection sites are large.
- A logic block is surrounded by connection blocks on all four sides.

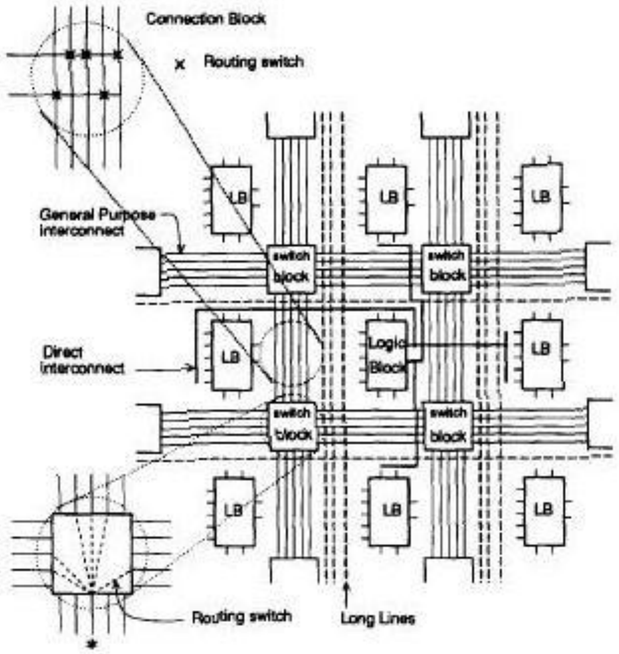


Figure: Xilinix Routing Architecture

- They connect logic block pins to wire segments.
- Pass transistors are used to implement connection for output pins, while use of multiplexers for input pins saves the number of SRAM cells required per pin.
- The logic block pins connecting to connection blocks can then be connected to any number of wire segments through switching blocks.
- Figure shows the Xilinx routing architecture.
- There are four types of wire segments available:
  - ✓ General purpose segments that pass through switches in the switch block.
  - ✓ Direct interconnect connects logic block pins to four surrounding connecting blocks
  - ✓ Long line: high fan out uniform delay connections
  - ✓ Clock lines: clock signal provider which runs all over the chip.

(c) Altera Routing Architecture :

- Altera routing architecture has two level hierarchies.
- At the first level of the hierarchy, 16 or 32 of the logic blocks are grouped into a Logic Array Block (LAB).
- The channel here is set of wires that run vertically along the length of the FPGA.
- Figure shows Alter Max 5000 routing architecture.
- Tracks are used for four types of connections:
  - ✓ Connections from output of all logic blocks in LAB.
  - ✓ Connection from logic expanders.
  - ✓ Connections from output of logic blocks in other LABs
  - ✓ Connections to and from Input output pads

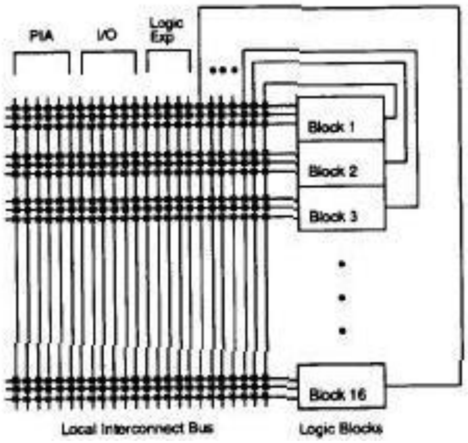


Figure: Altera Max 5000 Routing Architecture

- All four types of tracks connect to every logic block in the array block.
- Any track can connect to into any input which makes this routing simple.
- Advantage: It allows to be packed tightly and efficiently.
- Disadvantage: Large number of switches required, which adds to capacitive load.

(d) Island-Style Routing Architecture:

- As shown in Figure, island-style FPGAs logic blocks are arranged in a two dimensional mesh with the routing resources evenly distributed throughout the mesh.
- An island-style global routing architecture typically has the routing channels on all four sides of the logic blocks.
- The number of wires contained in the channel, W, is pre-set during fabrication, and is one of the key choices made by the architect.
- It employs wire segments of different lengths in each channel to provide the most appropriate length for each given connection.

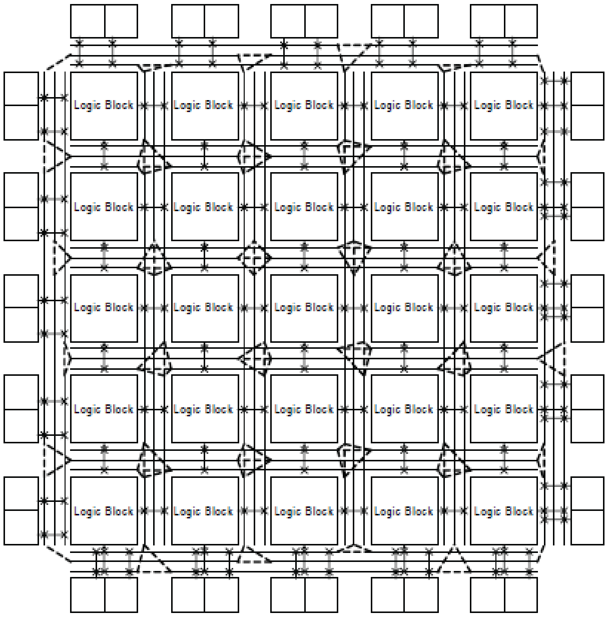


Figure: Island-Style Routing Architecture

(e) Actel Routing Architecture:

- Actel's design has more wire segments in horizontal direction than in vertical direction.
- The input pins connect to all tracks of the channel that is on the same side as the pin.
- The output pins extend across two channels above the logic block and two channels below it.
- Output pin can be connected to all 4 channels that it crosses.
- The switch blocks are distributed throughout the horizontal channels.
- All vertical tracks can make a connection with every incidental horizontal track.
- This allows for the flexibility that a horizontal track can switch into a vertical track, thus allowing for horizontal and vertical routing of same wire.
- The drawback is more switches are required which add up to more capacitive load.

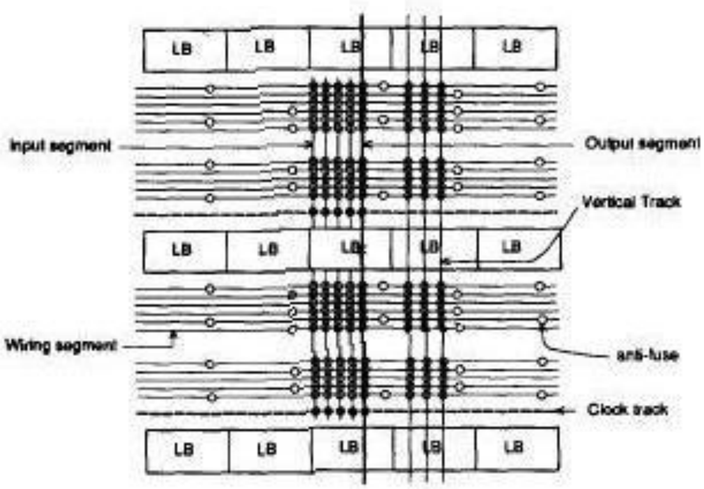


Figure: Actel Routing Architecture

INTERCONNECT:

Explain in detail about the interconnect.

Interconnect in an integrated circuit are physical connections between two transistors and/ or the externalsurroundings.

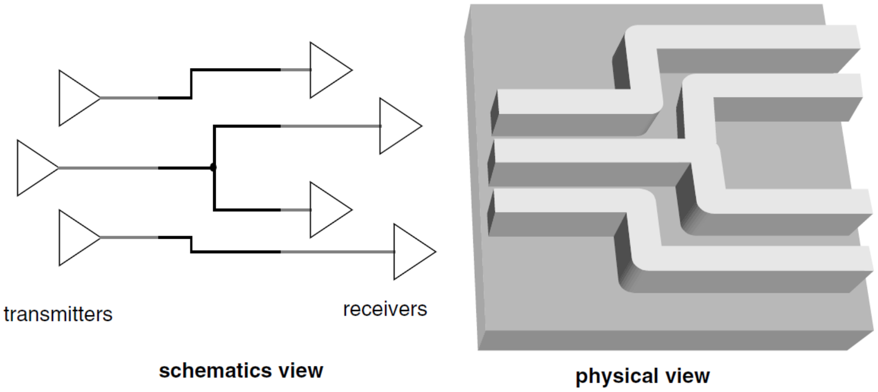
- ➢ An electronic circuit designer has multiple choices in realizing the interconnections between the various devices that make up the circuit.
- ➢ Here the start of the art processes offers multiple layers of aluminium or copper, and at least one layer of polysilicon. Even the heavily doped n^+ and p^+ diffusion layers are typically used for the realization of source and drain regions can be employed for wiring purposes. These wires appear in the schematic diagrams of electronic circuit as simple lines with no apparent impact on the circuit performance.

These wiring of integrated circuits forms a complex geometry that introduces the following parasitics:

1. Capacitive Parasitics
2. Resistive Parasitics and
3. Inductive Parasitics

The capacitive, resistive and inductive parasitics have multiple effects of the circuit's behaviour i.e.

- ➢ They all cause an increase in propagation delay, or equivalent, a drop in performance.
- ➢ They all have an impact on the energy dissipation and the power distribution.
- ➢ They all cause the introduction of extra noise sources, which affect the reliability of the circuit.



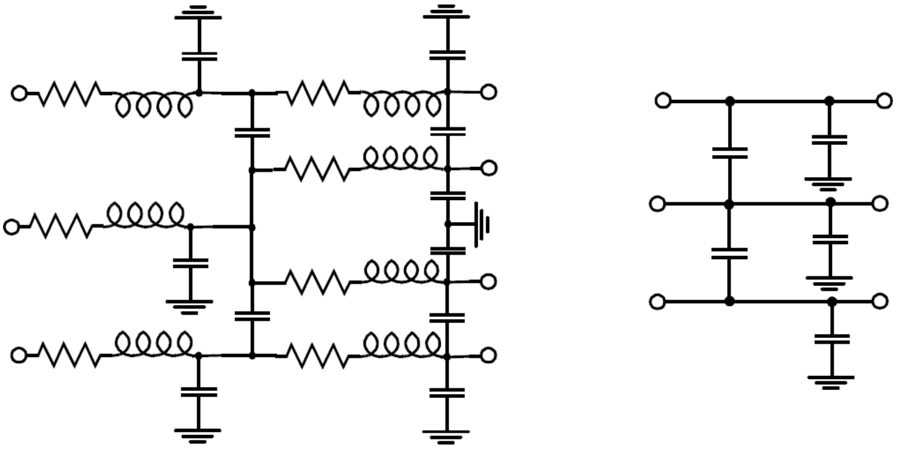
(FOR UNDERSTANDING- NO NEED TO DRAW)

SCHEMATIC AND PHYSICAL VIEWS OF WIRING OF BUS-NETWORK

It is important that the designer has a clear insight in the parasitic wiring effects, their relative importance, and their models. This is best illustrated with the simple example as shown above. Each wire in a bus network connects a transmitter (or transmitters) to a set of receivers and is implemented as a link of wire segments of various lengths and geometries. Assume that all segments are implemented on a single interconnect layer, isolated from the silicon substrate and from each other by a layer of dielectric material.Be aware that the reality may be far more complex.

Analyzing the behavior of this schematic, which only models a small part of the circuit, is slow and cumbersome. Fortunately, substantial simplifications can often be made, some of which are enumerated below,

- Inductive effects can be ignored if the resistance of the wire is substantial- this is for instance the case for long Aluminum wires with a small cross-section- or if the rise and fall times of the applied signals are slow.
- When the wires are short, the cross-section of the wire is large, or the interconnect material used has a low resistivity, a capacitance- only model can be used (figure FOR WIRE PARASITICS WITH CAPACITANCE ONLY shown below)
- The separation between neighbouring wires is large, or when the wires only run together for a short distance, inter-wire capacitance can be ignored, and all the parasitic capacitance can be modeled as capacitance to ground. Obviously, the latter problems are the easiest to model, analyze, and optimize.



WIRE PARASITICS (WITH THE EXCEPTION OF INTER-WIRE RESISTANCE AND MUTUAL INDUCTANCE) *WIRE PARASITICS WITH CAPACITANCE ONLY*

WIRE MODELS FOR PARASITICS

The various interconnect parameters whose values can be estimated, simple models to evaluate their impact, and a set of rules- of- thumb to decide i.e. when and where a particular model or effect should be consideredare:

1. Capacitance Parameter
2. Resistance Parameter
3. Inductance Parameter

<u>CAPACITANCE INTERCONNECT PARAMETER</u>:

The capacitance of such a wire is a function of its shape, its environment, its distance to the substrate, and the distance to surrounding wires. An accurate modeling of the wire capacitance(s) in a state-of-the-art integrated circuit is a non-trivial task and is even today the subject of advanced research.

In capacitance parameter there are two types of capacitance occurring i.e.

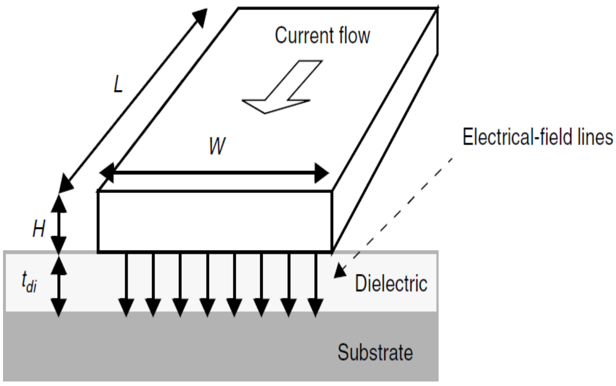
1. Parallel plate Capacitance and
2. Fringe Capacitance

Parallel plate Capacitance:

Consider first a simple rectangular wire placed above the semiconductor substrate, as shown in figure below. If the width of the wire is substantially larger than the thickness of the insulating material, it may be assumed that the electrical-field lines are orthogonal to the capacitor plates, and that its capacitance can be modeled by the *parallel-plate capacitance model.* Under those circumstances, the total capacitance of the wire can be approximated as,

$$c_{int} = \frac{\varepsilon_{di}}{t_{di}} WL$$

Where *W* and *L* are respectively the width and length of the wire, and t_{di} and ε_{di} represent the thickness of the dielectric layer and its permittivity. SiO2 is the dielectric material of choice in integrated circuits, although some materials with lower permittivity, and hence lower capacitance, are coming in use.

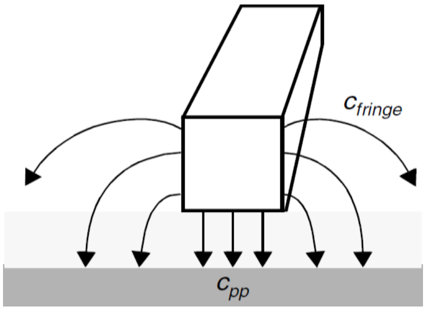


PARALLEL-PLATE CAPACITANCE MODEL OF INTERCONNECT WIRE

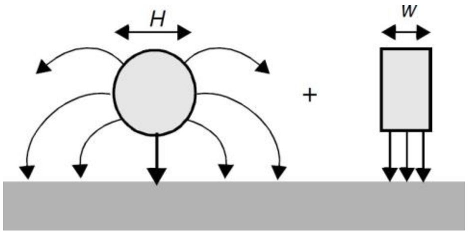
In actuality, this model is too simplistic. To minimize the resistance of the wires while scaling technology, itis desirable to keep the cross-section of the wire (*W*×*H*) as large as possible. On the other hand, small values of *W* lead to denser wiring and less area overhead. As a result, we have over the years witnessed a steady reduction in the *W/H*- ratio, such that it has even dropped below unity in advanced processes.

Fringe/ Fringing Capacitance:

The capacitance between the side-walls of the wires and the substrate, called the *fringing capacitance*, as shown below.



Fringing fields/ the fringing-field capacitance



model of fringing-field capacitance- decomposes the Capacitance into two contributions: a parallel-plate capacitance,and a fringing capacitance, modeled by a cylindrical wire with a diameter equal to the thickness of the wire

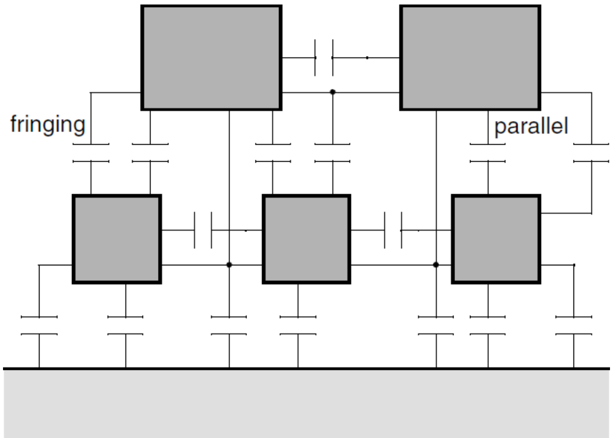
Therefore, the parallel plate capacitance and fringing capacitance constitutes the overall capacitance. Which is given as,

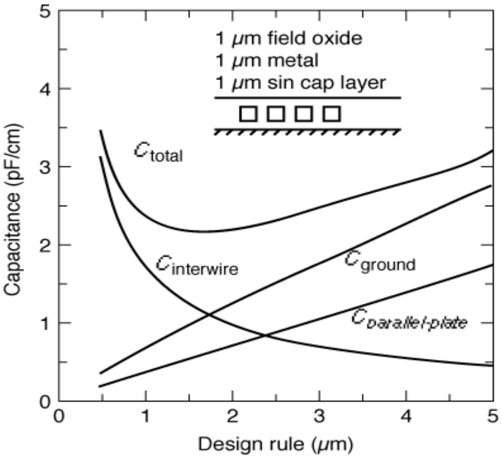
$$c_{wire} = c_{pp} + c_{fringe} = \frac{w\varepsilon_{di}}{t_{di}} + \frac{2\pi\varepsilon_{di}}{\log(t_{di}/H)}$$

With $w = W - H/2$ a good approximation for the width of the parallel-plate capacitor.

CAPACITANCE COUPLING/ CAPACITANCE COUPLING EFFECT:

Assuming that a wire is completely isolated from its surrounding structures and is only capacitively coupled to ground, becomes untenable. This is illustrated in figure, where the capacitance components of a wire embedded in an interconnect hierarchy are identified. Each wire is not only coupled to the grounded substrate, but also to the neighbouring wires on the same layer and on adjacent layers. The main differenceis that not all its capacitive components do terminate at the grounded substrate, but that a large number of them connect to other wires, which have dynamically varying voltage levels, these floating capacitors causescrosstalk and a negative effect to the circuit also.





CAPACITIVE COUPLING BETWEEN WIRES IN INTERCONNECT HIERARCHY INTERCONNECT CAPACITANCE AS A FUNCTION OF DESIGN RULES

Inter- wire capacitances become a dominant factor in multi- layer interconnect structures. This effect is more important for wires in the higher interconnect layers, as these wires are farther away from the substrate. The increasing contribution of the inter- wire capacitance to the total capacitance with decreasing feature sizes is illustrated by graphical figure as shown, which plots the capacitive components of a set of parallel wires routed above a ground plane, it is assumed that dielectric and wire thickness are held constant while scaling all other dimensions. When W becomes smaller than 1.75 H, the inter-wire capacitance starts to dominate.

Wiring Capacitances for 0.25 µm CMOS Technology:

The table rows represent the top plate of the capacitor, the columns the bottom plate. The area capacitancesare expressed in aF^1/µm^2, while the fringe capacitances (given in the shaded rows) are in aF/µm.

	Field	Active	Poly	Al1	Al2	Al3	Al4
Poly	88						
	54						
Al1	30	41	57				
	40	47	54				
Al2	13	15	17	36			
	25	27	29	45			
Al3	8.9	9.4	10	15	41		
	18	19	20	27	49		
Al4	6.5	6.8	7	8.9	15	35	
	14	15	15	18	27	45	
Al5	5.2	5.4	5.4	6.6	9.1	14	38
	12	12	12	14	19	27	52

Inter- Wire Capacitance per unit wire length for different interconnect layers of 0.25 µm CMOSTechnology Process:

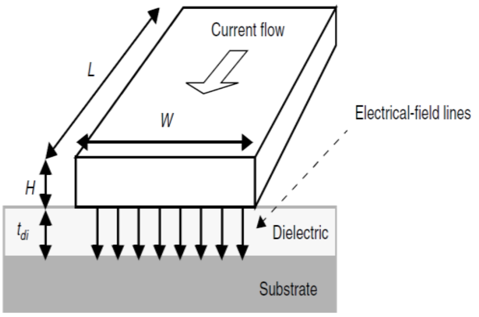
The capacitances are expressed in aF/mm, and are for minimally-spaced wires

Layer	Poly	Al1	Al2	Al3	Al4	Al5
Capacitance	40	95	85	85	85	115

RESISTANCE INTERCONNECT PARAMETER:

The resistance of a wire is proportional to its length *L* and inversely proportional to its cross- section *A*.The resistance of a rectangular conductor as shown in figure below can be expressed as,

$$\boldsymbol{R = \frac{\rho L}{A}\ ohm = \frac{\rho L}{H\,W}\ ohm}$$



Where:

ρ = resistivity

A = HW = area of cross section of the rectangular wireIf L = W, i.e. square of resistive material, then

$$R = \frac{\rho}{H} = R_s\ in\ {}^{ohm}/{}_{square}\ \textit{(Sheet Resistance)}$$

the *sheet resistance* of the material, having units of Ω/ sq. This expresses that the resistance of a square conductor is independent of its absolute size, as is apparent from $R = R_S \frac{L}{W}$.

To obtain the resistance of a wire, simply multiply the sheet resistance by its ratio (L/ W).

<u>Resistivity of Commonly used Conductors/ Interconnect Resistance</u>:

Material	**ρ (Ω-m)**
Silver (Ag)	1.6×10^{-8}
Copper (Cu)	1.7×10^{-8}
Gold (Au)	2.2×10^{-8}
Aluminum (Al)	2.7×10^{-8}
Tungsten (W)	5.5×10^{-8}

Aluminum is the interconnect material most often used in integrated circuits because of its low cost and its compatibility with the standard integrated- circuit fabrication process. Unfortunately, it has a large resistivity compared to materials such as Copper. With ever-increasing performance targets, this is rapidly becoming a liability and top- of- the- line processes are now increasingly using Copper as the conductor of choice.

<u>Typical values of the Sheet Resistance of various Interconnect Materials using 0.25 µm CMOS Technology</u>:

Material	**Sheet Resistance (Ω/□)**
n- or p-well diffusion	1000 – 1500
n^+, p^+ diffusion	50 – 150
n^+, p^+ diffusion with silicide	3 – 5
n^+, p^+ polysilicon	150 – 200
n^+, p^+ polysilicon with silicide	4 – 5
Aluminum	0.05 – 0.1

From the table, we conclude that Aluminum is the preferred material for the wiring of long interconnections. Polysilicon should only be used for local interconnect. Although the sheet resistance of the diffusion layer (*n*+, *p*+) is comparable to that of polysilicon, the use of diffusion wires should be avoided due to its large capacitance and the associated *RC* delay.

INDUCTANCE INTERCONNECT PARAMETER:

The inductance of a section of a circuit states that a changing current passing through an inductor generates avoltage drop ΔV.

$$\Delta V = L\frac{\mathrm{d}i}{\mathrm{d}t}$$

On-chip inductance include ringing and overshoot effects, reflections of signals due to impedance mismatch, inductive coupling between lines, and switching noise due to *Ldi*/*dt* voltage drops.

It is possible to compute the inductance a wire directly from its geometry and its environment. A simpler approach relies on the fact that the capacitance *c* and the inductance *l* (per unit length) of a wire are relatedby the following expression,

$$cl = \varepsilon\mu$$

With ε and μ respectively the permittivity and permeability of the surrounding dielectric.

Other interesting relations, obtained from Maxwell’s laws, can be pointed out. The constant product of permeability and permittivity also defines the speed □ at which an electromagnetic wave can propagate through the medium,

$$\nu = \frac{1}{\sqrt{lc}} = \frac{1}{\sqrt{\varepsilon\mu}} = \frac{c_0}{\sqrt{\varepsilon_r\mu_r}}$$

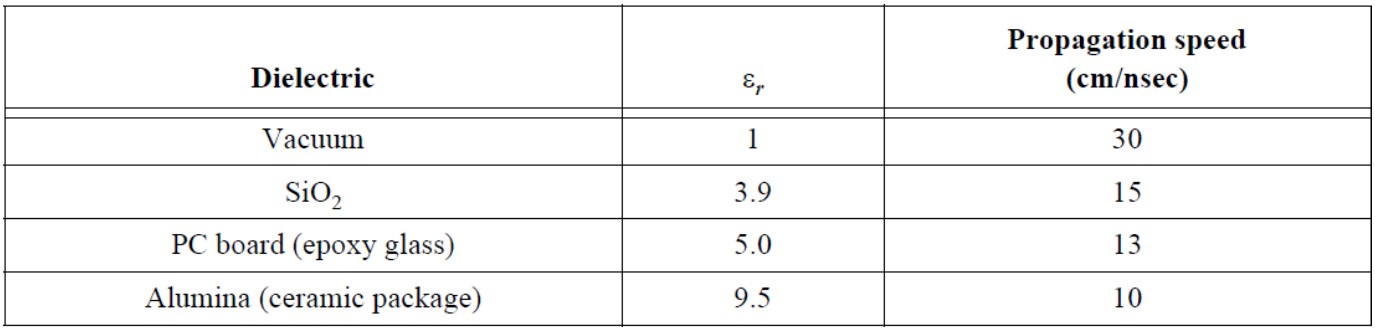
c_0 equals the speed of light (30 cm/ nsec) in a vacuum.

Considering a lumped RLC model we get $Z_{RL} = R + J\omega L, where\ \omega = 2\pi f.$

$$If\ R \gg J\omega L, then\ inductance\ effect\ is\ not\ important.$$

$$If\ R \ll J\omega L, then\ inductance\ effect\ will\ be\ observed\ in\ signal\ reduction\ in\ resistance$$

Dielectric constants and wave-propagation speeds for various materials used in electronic circuits;(The relative permeability μ_r of most dielectrics is approximately equal to 1)



Dielectric	ε_r	Propagation speed (cm/nsec)
Vacuum	1	30
SiO_2	3.9	15
PC board (epoxy glass)	5.0	13
Alumina (ceramic package)	9.5	10

INTERCONNECT MODELING:

Describe about interconnect modelling.

As we know the parasitic elements have an impact on the electrical behaviour of the circuit and influence its delay, power dissipation, and reliability. To study these effects requires the introduction of electrical models that estimate and approximate the real behaviour of the wire as a function of its parameters. These models vary from very simple to very complex depending upon the effects that are being studied and the required accuracy.

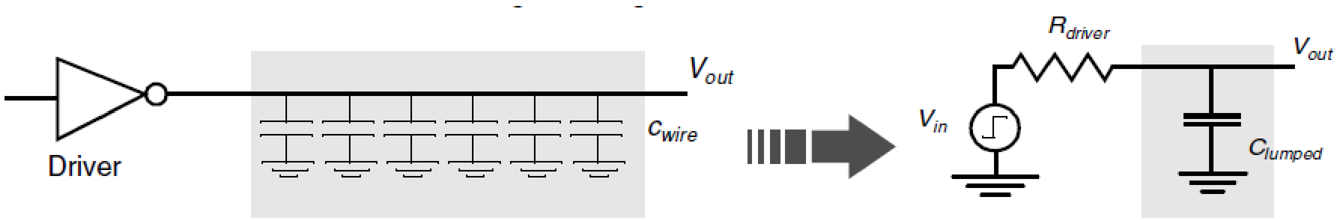
The types of interconnect modelling are:

1. Lumped Model
2. Lumped RC Model- The Elmore Delay
3. Distributed RC line Model/ Distributed *rc* line Model
4. Transmission Line Model

Lumped Model:

The circuit parasitics of a wire are distributed along its length and are not lumped into a single position. Yet, when only a single parasitic component is dominant, when the interaction between the components is small, or when looking at only one aspect of the circuit behaviour, it is often useful to lump the different fractions into a single circuit element. The advantage of this approach is that the effects of the parasitic then can be described by an ordinary differential equation.

As long as the resistive component of the wire is small and the switching frequencies are in the low to medium range, it is meaningful to consider only the capacitive component of the wire, and to lump the distributed capacitance into a single capacitor as shown in figure. It is observed that in this model the wire still represents an equipotential region, and that the wire itself does not introduce any delay. The only impacton performance is introduced by the loading effect of the capacitor on the driving gate. This capacitive lumped model is simple, yet effective, and is the model of choice for the analysis of most interconnect wires in digital integrated circuits.



DISTRIBUTED VERSUS LUMPED CAPACITANCE MODEL OF WIRE. CLUMPED = L×CWIRE, WITH L THE LENGTH OF THE WIRE AND CWIRE THE CAPACITANCE PER UNIT LENGTH. THE DRIVER IS MODELED AS A VOLTAGE SOURCE AND A SOURCE RESISTANCE RDRIVER

The operation of this simple RC network is described by the following ordinary differential equation,

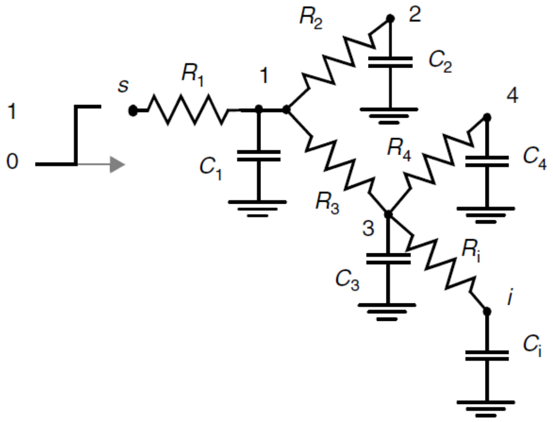
$$C_{lumped}\frac{dV_{out}}{dt} + \frac{V_{out} - V_{in}}{R_{driver}} = 0$$

<u>Lumped RC Model/ The Elmore Delay</u>:

On-chip metal wires of over a few mm length have a significant resistance. The equipotential assumption, presented in the lumped-capacitor model, is no longer adequate, and a resistive-capacitive model has to be adopted.

A first approach lumps the total wire resistance of each wire segment into one single *R* and similarly combines the global capacitance into a single capacitor *C*. This simple model, called the ***lumped RC model***, is pessimistic and inaccurate for long interconnect wires, which are more adequately represented by a *distributed rc-model*. Yet, before analyzing the distributed model, it is worthwhile to spend some time on theanalysis and the modeling of lumped *RC* networks for the following reasons:

1. The distributed *rc*-model is complex and no closed form solutions exist. The behaviour of the distributed *rc*- line can be adequately modeled by a simple *RC* network.
2. A common practice in the study of the transient behavior of complex transistor-wire networks is to reduce the circuit to an *RC* network. Having a means to analyze such a network effectively and to predict its first-order response would add a great asset to the designers tool box.



TREE- STRUCTURED RC NETWORK

An interesting result of this particular circuit topology is that there exists a unique resistive path between the source node *s* and any node *i* of the network. The total resistance along this path is called the *path resistance* R_{ii}. For example, the path resistance between the source node *s* and

$$R_{44} = R_1 + R_3 + R_4$$

node 4 equals,

The definition of the path resistance can be extended to address the *shared path resistance* R_{ik}, which represents the resistance shared among the paths from the root node *s* to nodes *k* and *i*:

$$R_{ik} = \sum R_j \Rightarrow (R_j \in [path(s \rightarrow i) \cap path(s \rightarrow k)])$$

Here,

$\boldsymbol{R_{i4} = R_1 + R_3}$ **while** $\boldsymbol{R_{i2} = R_1}$

Assume now that each of the *N* nodes of the network is initially discharged to GND, and that a step input is applied at node *s* at time *t* = 0. The Elmore delay at node *i* is then given by the following expression:

$$\tau_{Di} = \sum_{k=1}^{N} C_k R_{ik}$$

Therefore, the Elmore delay is equivalent to the first-order time constant of the network (or the first moment of the impulse response). The designer should be aware that this time- constant represents a simple approximation of the actual delay between source node and node *i*. Yet in most cases this approximation has proven to be quite reasonable and acceptable. It offers the designer a powerful mechanism for providing a quick estimate of the delay of a complex network.

The RC delay of a tree structured network is given as,

$$\tau_{Di} = R_1C_1 + R_1C_2 + (R_1 + R_3)C_3 + (R_1 + R_3)C_4 + (R_1 + R_3 + R_i)C_i$$

i.e. using

$$\tau_{Di} = \sum_{k=1}^{N} C_k R_{ik}$$

We can compute the Elmore delay for node *i*

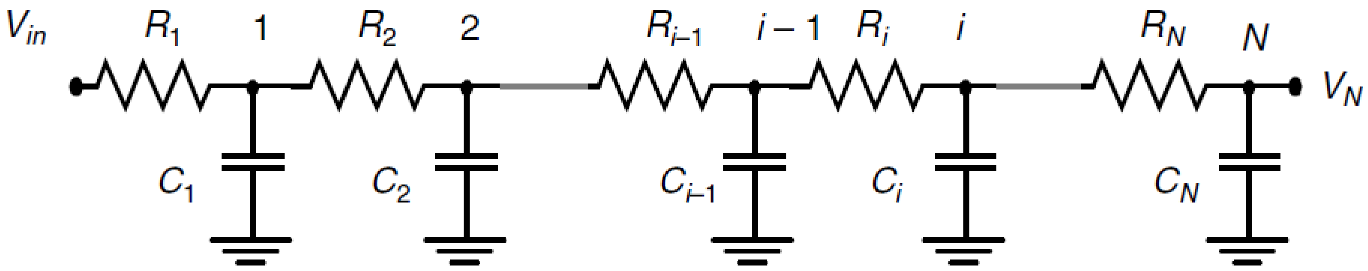
RC Chain/ The Elmore RC Chain Delay:

As a special case of the *RC* tree network, let us consider the simple, non- branched *RC* chain (or ladder) shown in figure. This network is worth analyzing because it is a structure that is often encountered in digital circuits, and also because it represents an approximative model of a resistive-capacitive wire. The Elmore delay of this chain network can be derived with the aid of

$$\tau_{Di} = \sum_{k=1}^{N} C_k R_{ik}$$

As

$$\tau_{DN} = \sum_{i=1}^{N} C_i \sum_{j=1}^{i} R_j = \sum_{i=1}^{N} C_i R_{ii}$$



RC CHAIN MODEL

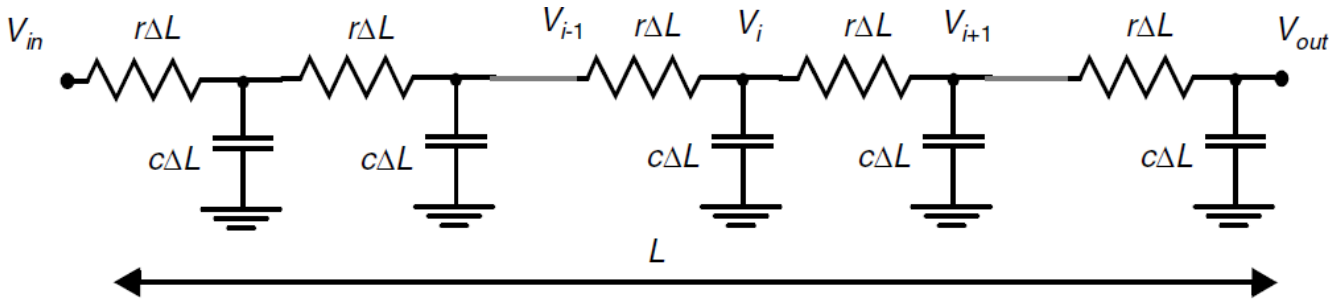
The component of node 1 consists of C_1R_1 with R_1 the total resistance between the node and the source, while the contribution of node 2 equals $C_2(R_1 + R_2)$. The equivalent time constant at node *2* equals$C_1R_1 + C_2(R_1 + R_2)$. τ_i of node *i* can be derived in a similar way.

$$\tau_{Di} = C_1R_1 + C_2(R_1 + R_2) + \ldots + C_i(R_1 + R_2 + \ldots + R_i)$$

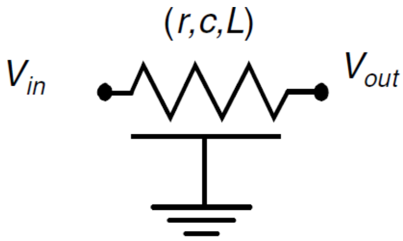
Thus, the Elmore delay formula has proven to be extremely useful. Besides making it possible to analyze wires, the formula can also be used to approximate the propagation delay of complex transistor networks. The evaluation of the propagation delay is then reduced to the analysis of the resulting *RC* network. More precise minimum and maximum bounds on the voltage waveforms in an *RC* tree have further been established.

<u>Distributed RC line Model/ Distributed *rc* line Model</u>:

A distributed *rc* line model is a more appropriate model as shown below which has, *r* and *c* stand for the resistance and capacitance per unit length.



DISTRIBUTED RC LINE MODEL



SCHEMATIC SYMBOL FOR DISTRIBUTED RC LINE

The voltage at node *i* of this network can be determined by solving the following set of partial differential equations:

$$c\Delta L\frac{\partial V_i}{\partial t} = \frac{(V_{i+1} - V_i) + (V_{i-1} - V_i)}{r\Delta L}$$

The correct behavior of the distributed *rc* line is then obtained by reducing ΔL asymptotically to 0. For$\Delta L \to 0$, the above equation becomes the well-known *diffusion*

$$rc\frac{\partial V}{\partial t} = \frac{\partial^2 V}{\partial x^2}$$

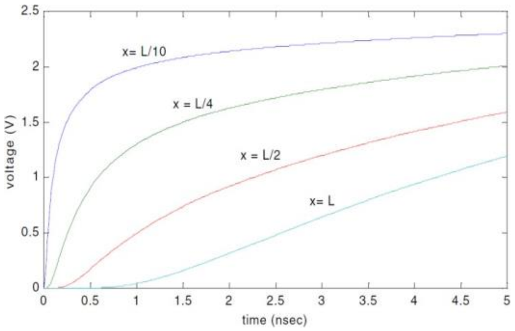
equation:

Where *V* is the voltage at a particular point in the wire, and *x* is the distance between this point and the signal source. No closed form solution exists for this equation, but approximative expressions such as the formula written below can be derived:

$$V_{out}(t) = 2erfc\left(\sqrt{\frac{RC}{4t}}\right) \qquad t \ll RC$$

$$= 1.0 - 1.366e^{-2.5359\frac{t}{RC}} + 0.366e^{-9.4641\frac{t}{RC}} \qquad t \gg RC$$

The graph below shows the response of a wire to a step input, plotting the waveforms at different points in the wire as a function of time. It is observable how the step waveform "diffuses" from the start to the end of the wire, and the waveform rapidly degrades, resulting in a considerable delay for long wires. Driving these *rc* lines and minimizing the delay and signal degradation is one of the trickiest problems in modern digital integrated circuit design.

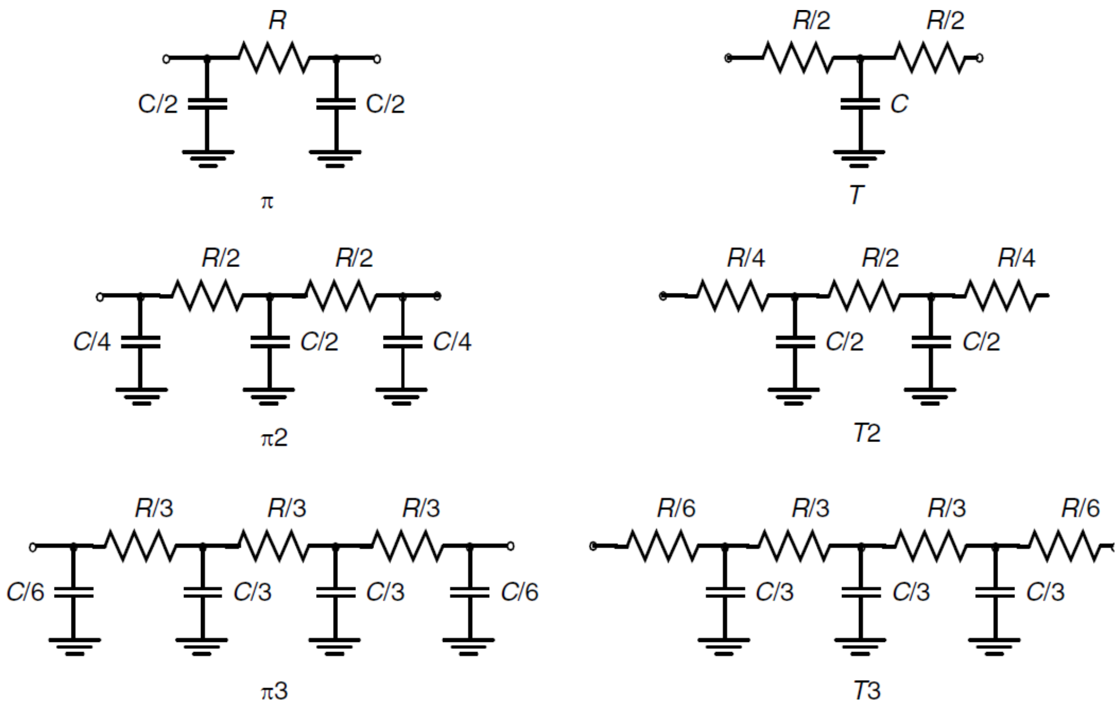


SIMULATED STEP RESPONSE OF RESISTIVE-CAPACITIVE WIRE AS A FUNCTION OF TIME AND PLACE

***rc* delays should only be considered when the rise (fall) time at the line input is smaller than *RC*, the rise (fall) time of the line.**

$$t_{rise} < RC$$

With R and C the total resistance and capacitance of the wire. When this condition is not met, the change in signal is slower than the propagation delay of the wire, and a lumped capacitive model suffices.



SIMULATION π AND T MODELS FOR DISTRIBUTED RC LINE

Step response of lumped and distributed *RC* networks- Points of Interest:

Voltage range	Lumped *RC* network	Distributed *RC* network
0 → 50% (t_p)	0.69 *RC*	0.38 *RC*
0 → 63% (τ)	*RC*	0.5 *RC*
10% → 90% (t_r)	2.2 *RC*	0.9 *RC*
0% → 90%	2.3 *RC*	1.0 *RC*

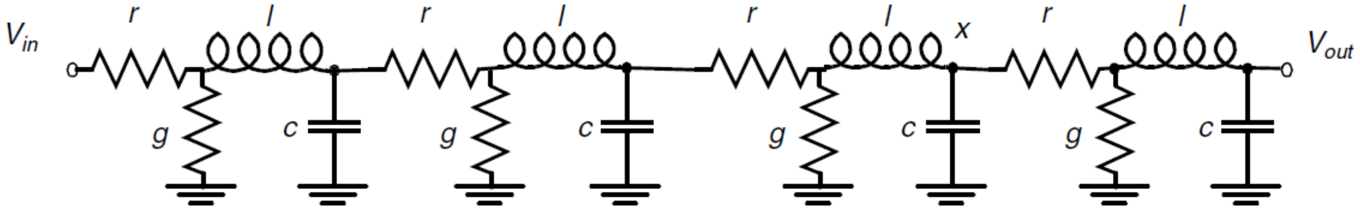
The Transmission Line Model:

Similar to the resistance and capacitance of an interconnect line, the inductance is distributed over the wire. A distributed *rlc* model of a wire, known as the transmission line model, becomes the most accurate approximation of the actual behaviour.

The transmission line has the prime property that a signal propagates over the interconnection medium as a *wave*. This is in contrast to the distributed *rc* model, where the signal *diffuses* from the source to the destination governed by the diffusion equation i.e.

$$rc\frac{\partial V}{\partial t} = \frac{\partial^2 V}{\partial x^2}$$

In the wave mode, a signal propagates by alternatively transferring energy from the electric to the magnetic fields, or equivalently from the capacitive to the inductive modes.



A LOSSY TRANSMISSION LINE

Consider the point *x* along the transmission line of figure as shown above at time *t*. The following set of equations holds:

$$\frac{\partial v}{\partial x} = -ri - l\frac{\partial i}{\partial t}$$

$$\frac{\partial i}{\partial x} = -gv - c\frac{\partial v}{\partial t}$$

Assuming that the leakage conductance *g* equals 0, which is true for most insulating materials, and eliminating the current *i* yields the *wave propagation equation,*

$$\frac{\partial^2 v}{\partial x^2} = rc\frac{\partial v}{\partial t} + lc\frac{\partial^2 v}{\partial t^2}$$

where *r*, *c*, and *l* are the resistance, capacitance, and inductance per unit length respectively.

COPING WITH INTERCONNECT:

As till now we have concentrated on the growing impact of interconnect parasitics on all design metrics of digital integrated circuits. As mentioned, interconnect introduces three types of parasitic effects i.e. capacitive, resistive, and inductive- all of which influence the signal integrity and degrade the performanceof the circuit. While so far we have concentrated on the modeling aspects of the wire, we now analyze how interconnect affects the circuit operation, and we present a collection of design techniques to cope with theseeffects with considering each parasitics- this is referred to as *coping with interconnect*.

CAPACITIVE PARASITICS:

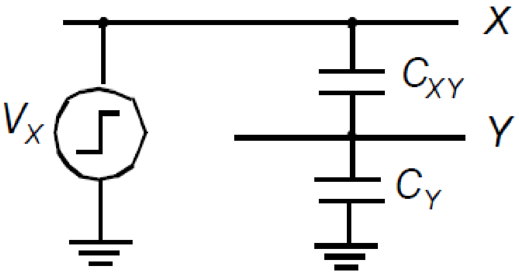
➢ Capacitance reliability and Cross talk:

An unwanted coupling from a neighbouring signal wire to a network node introduces an interference that is generally called *cross talk*. The resulting disturbance acts as a noise source and can lead to hard-to-trace intermittent errors, since the injected noise depends upon the transient value of the other signals routed in theneighbourhood. In integrated circuits, this inter signal coupling can be both capacitive and inductive.

Capacitive cross talk is the dominant effect at current switching speeds, although inductive coupling forms a major concern in the design of the input-output circuitry of mixed-signal circuits. The potential impact of capacitive crosstalk is influenced by the impedance of the line under examination. If the line is floating, the disturbance caused by the coupling persists and may be worsened by subsequent switching on adjacentwires. If the wire is driven, on the other hand, the signal returns to its original level.

o **Floating Lines:**

Approaching capacitive parasitic with respect to capacitive coupling to a floating line.



CAPACITIVE COUPLING TO A FLOATING LINE

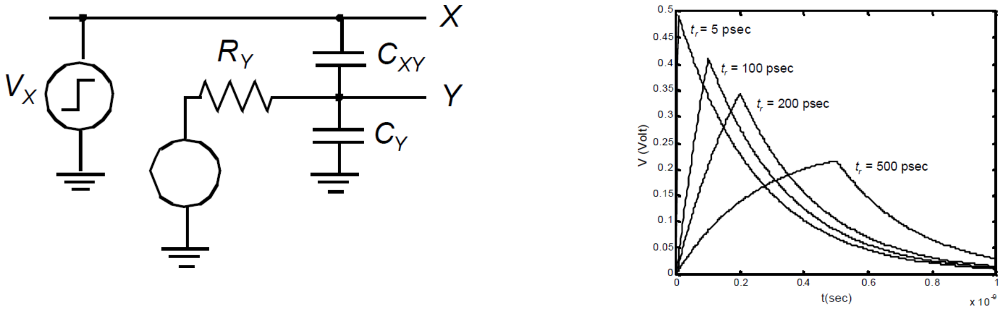
Considering the circuit shown as above, where line X is coupled to wire Y by a parasitic capacitance C_{XY}. Line Y sees a total capacitance to ground equal to C_Y. Assuming that the [illegible] eriences a step change equal to ΔV_X. This step appears on node Y [illegible] tive voltage divider.

$$\Delta V_Y = \frac{C_{XY}}{C_Y + C_{XY}} \Delta V_X$$

Circuits that are particularly susceptive to capacitive cross talk are networks with low- swing pre chargednodes, located in adjacent to full- swing wires (with $\Delta V_X = V_{DD}$).

Examples of these are dynamic memories, low swing on chip busses and some dynamic families.To address the cross talk issue, level- restoring device or keepers are a must in dynamic logic.

o **Driven Lines:**



CAPACITIVE COUPLING TO A DRIVEN LINE AND ITS VOLTAGE RESPONSE

As seen from the figure, if the line Y is driven with a resistance R_Y, a step on line X results in a transient on line Y. The transient decays with a time constant $\tau_{XY} = R_Y (C_{XY} + C_Y)$. The actual impact on the victim line is a strong function of the rise- fall time of the interfering signal.

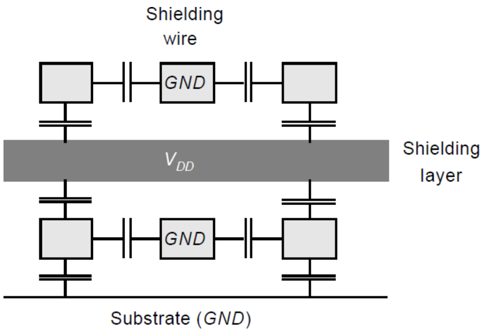
If the rise time is comparable or larger than the time constant, the peak value of disturbance is diminished. This can be observed in the response figure.

Obvious, keeping the driving impedance of a wire and hence τ_{XY} low goes a long way towards reducing the impact of capacitive cross talk. The keeper transistor added to a dynamic gate or pre charged wire is an excellent example of how impedance reduction helps to control noise.

Therefore, the impact of cross talk on the signal integrity of driven nodes is rather limited. The resulting glitches may cause malfunctioning of connecting sequential elements, and should therefore be carefully monitored. The most important effect is an increase in delay.

Design Techniques to Deal with Capacitive Cross talk:

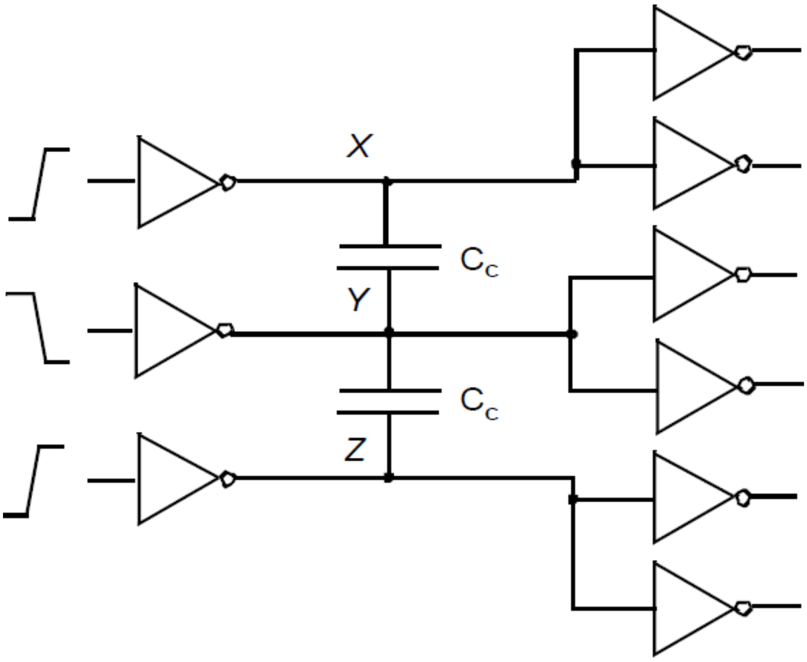
1. If possible avoid floating nodes, nodes sensitive to cross talk problems such as pre charged busses,should be equipped with keeper devices to reduce the impedance.
2. Sensitive nodes should be well separated from full swing signals.
3. Making the rise- fall time as large as possible subjection to timing constraints.
4. Use differential signalling in sensitive low swing wiring networks. This turns the cross talk signal into a common mode noise source that does not impact the operation of the circuit.
5. To keep the cross talk minimum, do not allow the capacitance between the two signal wires to grow too large.
6. If necessary provide shielding wire- GND or V_{DD} between the two signals as show below. This effectively turns the interwire capacitance into a capacitance to ground and eliminates interference. An adverse effect of shielding is the increased capacitive load.



CROSS SECTION OF ROUTING LAYERS ILLUSTRATING THE USE OF SHIELDING TO REDUCE CAPACITIVE CROSS TALK

7. The interwire capacitance between signals on different layers can be further reduced by addition of extra routing layers.

- **Impact of Cross talk on Propagation Delay (With respect to CMOS):**



IMPACT OF CROSS TALK ON PROPAGATION DELAY

The circuit schematic illustrates of how capacitive cross talk may result in a data-dependent variation of the propagation delay. Assume that the inputs to the three parallel wires X, Y, and Z experience simultaneous transitions. Wire *Y* (called the victim wire) switches in a direction that is opposite to the transitions of its neighbouring signals *X* and *Z*. The coupling capacitances experience a voltage swing that is double the signal swing, and hence represent an effective capacitive load that is twice as large as C_c- the by now well known *Miller effect*.

Since the coupling capacitance represents a large fraction of the overall capacitance in the deep-submicron dense wire structures, this increase in capacitance is substantial, and has a major impact on the propagation delay of the circuit. Observe that this is a worst-case scenario. If all inputs experience a simultaneous transition in the same direction, the voltage over the coupling capacitances remains constant, resulting in a zero contribution to the effective load capacitance.

The total load capacitance C_L of gate *Y*, hence depends upon the data activities on the neighbouring signals and varies between the following bounds:

$$C_{GND} \leq C_L \leq C_{GND} + 4C_c$$

with C_{GND} the capacitance of node *Y* to ground, including the diffusion and fan out capacitances.

Design Techniques for Circuit Fabrics with Predictable delay:

With cross talk making wire-delay more and more unpredictable, a designer can choose between a numberof different methodology options to address the issue, some of which are,

1. Evaluate and improve: After detailed extraction and simulation, the bottlenecks in delay are identified,and the circuit is appropriately modified.
2. Constructive layout generation: Wire routing programs take into account the effects of the adjacent wires,ensuring that the performance requirements are met.
3. Predictable structures: By using predefined, known, or conservative wiring structures, the designer is that the circuit will meet his specifications and that cross talk will not be a show stopper.

- **Capacitive Load (With respect to CMOS):**

The increasing values of the interconnect capacitances, especially those of the global wires, emphasize the need for effective driver circuits that can (dis)charge capacitances with sufficient speed. This need is further highlighted by the fact that in complex designs a single gate often has to drive a large fan-out and hence hasa large capacitive load.

Typical examples of large on-chip loads are busses, clock networks, and control wires. The latter include,for instance, reset and set signals. These signals control the operation of a large number of gates, so fan-outis normally high. Other examples of large fan-outs are encountered in memories where a large number of storage cells is connected to a small set of control and data wires.

The capacitance of these nodes is easily in the multi-pico farad range. The worst case occurs when signalsgo off-chip. In this case, the load consists of the package wiring, the printed circuit board wiring, and the input capacitance of the connected ICs or components.

Typical off-chip loads range from 20 to 50 pF, which is multiple thousand times larger than a standard on- chip load. Driving those nodes with sufficient speed becomes one of the most crucial design problems.

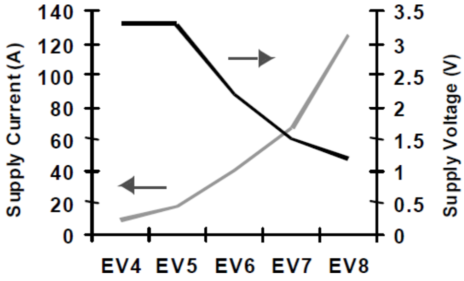
The main secrets to the efficient driving of large capacitive loads are:

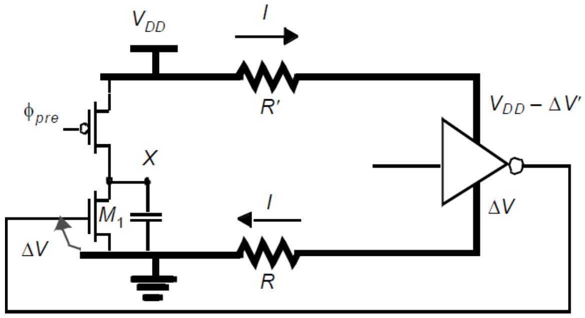
1. Adequate transistor sizing is instrumental when dealing with large loads.
2. Partitioning drivers into chains of gradually-increasing factors.

RESISTIVE PARASITICS:

➢ Resistance and Reliability- Ohmic Voltage Drop:

Current flowing through a resistive wire results in an ohmic voltage drop that degrades the signal levels. This is especially important in the power distribution network, where current levels can easily reach amperesas shown below.





EVOLUTION OF POWER SUPPLY CURRENT AND SUPPLY VOLTAGE OHMIC VOLTAGE DROP ON THE SUPPLY REDUCES NOISE MARGIN

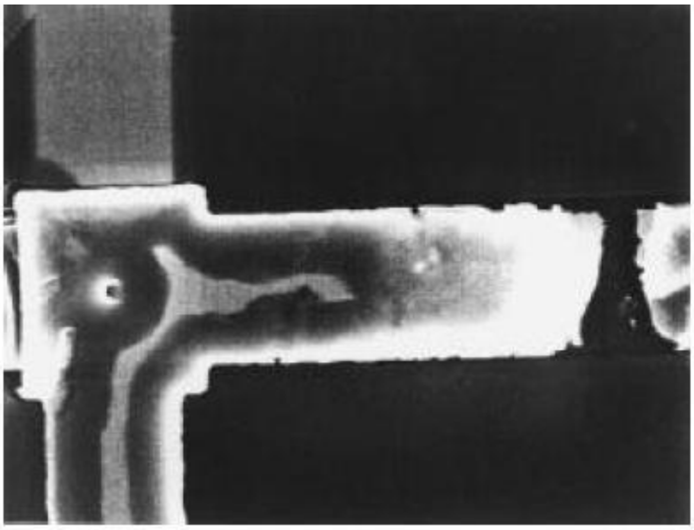
Consider a 2 cm long V_{DD} or *GND* wire with a current of 1mA per µm width. This current is about the maximum that can be sustained by an aluminum wire due to *electromigration* and assuming asheet resistance of 0.05 Ω/sq, the resistance of this wire (per µm width) equals 1 kΩ. A current of 1 mA/µm would result in a voltage drop of 1 V. The altered value of the voltage supply reduces noise margins and changes the logic levels as a function of the distance from the supply terminals. This is demonstrated by the circuit shown above, where an inverter placed far from the power and ground pins connects to a devicecloser to the supply.

The difference in logic levels caused by the *IR* voltage drop over the supply rails might partially turn on transistor M_1. This can result in an accidental discharging of the pre charged, dynamic node *X*, or causestatic power consumption if the connecting gate is static. In short, the current pulses from the on-chip logic, memories and I/O pins cause voltage drops over the power- distribution network and are the major sourcefor on- chip power supply noise. Beyond causing a reliability risk, IR drops on the supply network also impact the performance of the system. A small drop in the supply voltage may cause a significant increase indelay.

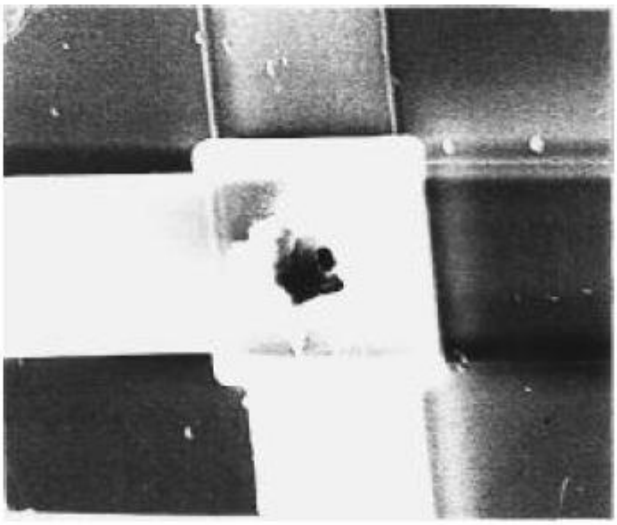
The most obvious problem is to reduce the maximum distance between the supply pins and the circuit supply connections which is most easily accomplished through a structured layout of the power distribution network. A number of on- chip power distribution networks with peripheral bonding.

➢ Electromigration:

The current density (current per unit area) in a metal wire is limited due to an effect called *electromigration*. A *direct* current in a metal wire running over a substantial time period, causes a transport of the metal ions. Eventually, this causes the wire to break or to short circuit to another wire. This type of failure will onlyoccur after the device has been in use for some time.



Line Open Failure



Open Failure in Contact Plug

ELECTROMIGRATION RELATED FAILURE MODES

The rate of the electromigration depends upon the temperature, the crystal structure, and the average current density. The latter is the only factor that can be effectively controlled by the circuit designer. Keeping the current below 0.5 to 1 mA/ µm normally prevents migration. This parameter can be used to determine the minimal wire width of the power and ground network.
Signal wires normally carry an ac- current and are less susceptible to migration. The bidirectional flow of the electrons tends to anneal any damage done to the crystal structure. Most companies impose a number of strict wire-sizing guidelines on their designers, based on measurements and past experience.

Electromigration effects are proportional to the average current flow through the wire, while IR voltage drops are a function of the peak current.

From designing point of view, at the technology level, a number of precautions can be taken to reduce themigration risk i.e.

1. To add alloying elements (such as Cu or Tu) to the aluminum to prevent the movement of the Alions.
2. To control the granularity of the ions.
3. The introduction of new interconnect materials is a big help as well. For instance, the use of Copper interconnect increases the expected lifetime of a wire with a factor of 100 over Al.

➢ **Resistance and Performance- RC Delay:**

The delay of a wire grows quadratically with its length. Doubling the length of a wire increases its delay by a factor of four. The signal delay of long wires therefore tends to be dominated by the *RC* effect. This is becoming an ever larger problem in modern technologies, which feature an increasing average length of the global wires, at the same time that the average delay of the individual gates is going down. This leads to the rather bizar situation that it may take multiple clock cycles to get a signal from one side of a chip to its opposite end.

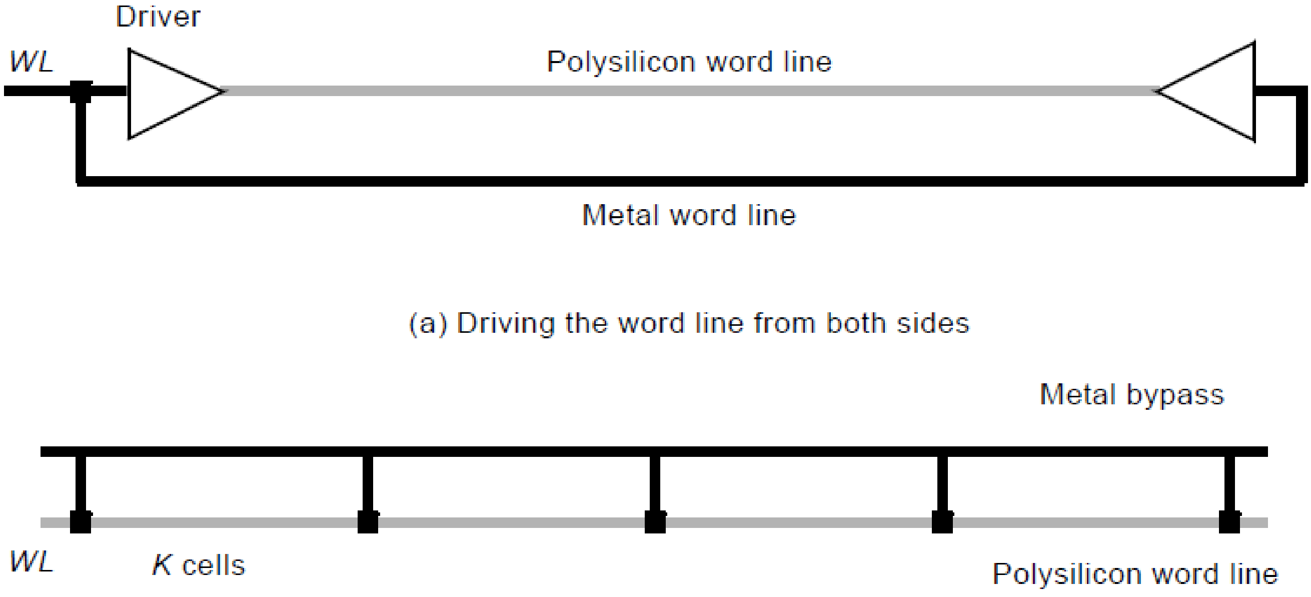
Providing accurate synchronization and correct operation becomes a major challenge under these circumstances. Therefore the different design techniques to cope with the delay imposed by the resistance ofthe wire are,

- **Better Interconnect Materials:**

Use better interconnect materials when they are available and appropriate. The introduction of silicides and Copper have helped to reduce the resistance of polysilicon (and diffused) and metal wires, respectively, while the adoption of dielectric materials with a lower permittivity lowers the capacitance. Both Copper and low- permittivity dielectrics have become common in advanced CMOS technologies.

Here the designer should be aware that these new materials only provide a temporary respite of one or two generations, and do not solve the fundamental problem of the delay of long wires. Innovative design techniques are often the only way of coping with the latter.

Sometimes, it is hard to avoid the use of long polysilicon wires. A good example of such circumstance are the address lines in memories, which must connect to a large number of transistor gates. Keeping the wiresin polysilicon increases the memory density substantially by avoiding the overhead of the extra metal contacts. The polysilicon- only option unfortunately leads to an excessive propagation delay. One possible solution is to drive the word line from both ends, as shown in Figure. This effectively reduces the worst-casedelay by a factor of four. Another option is to provide an extra metal wire, called a *bypass*, which runs parallel to the polysilicon one, and connects to it every *k* cells as shown in figure. The delay is now dominated by the much shorter polysilicon segments between the contacts. Providing contacts only every *k* cells helps to preserve the implementation density.



(a) Driving the word line from both sides

(b) Using a metal bypass

APPROACHES TO REDUCE THE WORD LINE DELAY

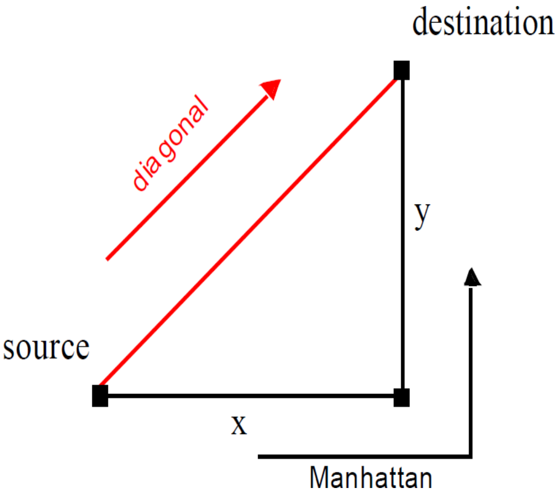
o **Better Interconnect Strategies:**

The length of the wire being a prime factor in both the delay and the energy consumption of an interconnectwire, any approach that helps to reduce the wire length is bound to have an essential impact.

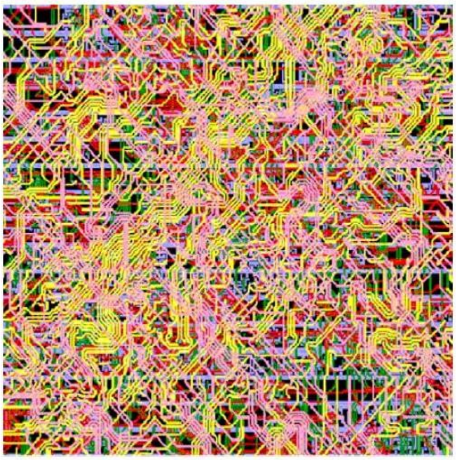
There are two wiring strategies i.e. the Manhattan- Style Routing and Diagonal- Style Routing.

- In Manhattan style routing, interconnections are first routed along the one of the preferred directions,followed by a connection in the other direction as shown.

- In Diagonal style routing less size of the wire length is required, on comparison to Manhattan 29% inbest case. And the use of 45°lines is ironical in integrated circuits. The main issues of diagonal routing are its complexity, impact on tools and masking concerns.



MANHATTAN VS DIAGONAL ROUTING

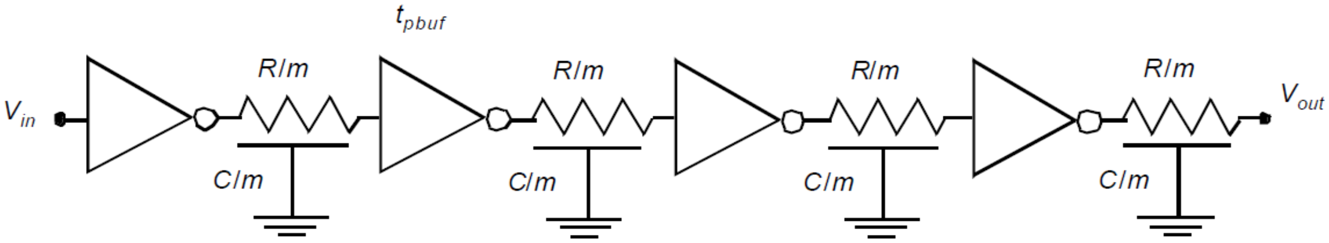


LAYOUT EXAMPLE OF 45° LINES (FOR UNDERSTANDING- NO NEED TO DRAW)

Earlier Manhattan routing was preferred because of the issues of diagonal routing inspite of its features. Now diagonal routing is preferred due to its features i.e. less wire length and 45° lines, its issues of complexity, impact on tools and masking concerns are easily overcomed nowadays by using CAD tools (Computer Aided Design Tools) like Cadence. Therefore the impact on wiring is quite tangible, a reduction of 20% in wire length, resulting in higher performance, lower power dissipation and smaller chip area.

○ **Introducing Repeaters/ Buffer Insertion for very long wires:**

The most popular design approach to reducing the propagation delay of long wires is to introduce intermediate buffers, also called *repeaters*, in the interconnect line as shown below.



REDUCING RC INTERCONNECT DELAY BY USING REPEATERS

Making an interconnect line *m* times shorter reduces its propagation delay quadratically, and is sufficient to offset the extra delay of the repeaters when the wire is sufficiently long. Assuming that the repeaters have a fixed delay t_{pbuf}, we can derive the delay of the partitioned wire.

$$t_p = 0.38rc\left(\frac{L}{m}\right)^2 m + mt_{pbuf}$$

The optimal number of buffers that minimizes the overall delay can be found by setting $\frac{\partial t_p}{\partial m} = 0$,

$$m_{opt} = L\sqrt{\frac{0.38rc}{t_{pbuf}}} = \sqrt{\frac{t_{pwire(unbuffered)}}{t_{pbuf}}}$$

yielding a minimum delay of,

$$t_{p,\,opt} = 2\sqrt{t_{pwire(unbuffered)}t_{pbuf}}$$

and is obtained when the delay of the individual wire segments is made equal to that of a repeater.

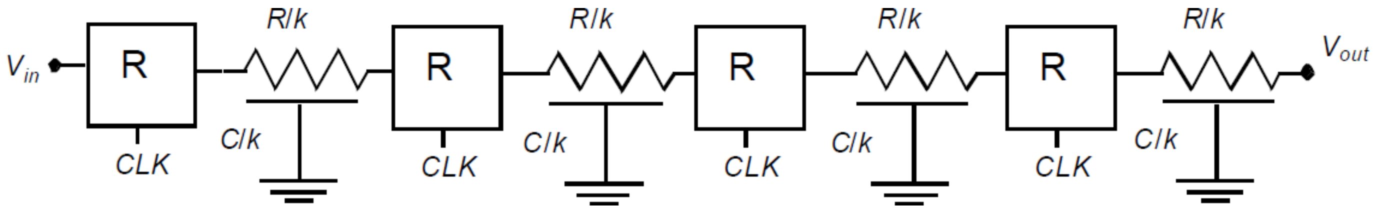
- **Optimizing the Interconnect Architecture:**

Even with buffer insertion, the delay of a resistive wire cannot be reduced below the minimum dictated by Equation,

$$t_{p,\,opt} = 2\sqrt{t_{pwire(unbuffered)}t_{pbuf}}$$

Long wires hence often exhibit a delay that is longer than the clock period of the design. For instance, the 10cm long Al wire of comes with a minimum delay of 4.7 nsec, even after optimal buffer insertion and sizing, while the 0.25 □m CMOS process featured in this text can sustain clock speeds in excess of 1 GHz (this is, clock periods below 1 nsec). The wire delay all-by-itself hence becomes the limiting factor on the performance achievable by the integrated circuit. The only way to address this bottleneck is to tackle it at thesystem architecture-level.

Wire pipelining is a popular performance- improvement technique in this category which improves the throughput performance of logic modules with long critical paths. Similar approach can be used to increase the throughput of a wire, as is illustrated in figure below.



WIRE PIPELINING IMPROVES THE THROUGHPUT OF A WIRE

The wire is partitioned in *k* segments by inserting registers or latches. While this does not reduce the delay through the wire segment, it takes *k* clock cycles for a signal to proceed through the wire, it helps to increase its throughput, as the wire is handling *k* signals simultaneously at any point in time. The delay of the individual wire segments can further be optimized by repeater insertion, and should be below a single clock period.

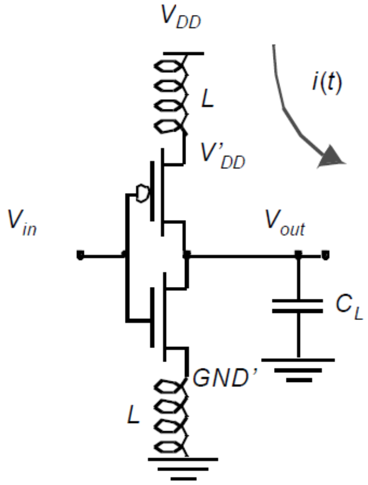
This is only one example of the many techniques that the chip architect has at her disposal to deal with the wire delay problem. The most important concern from this is that the wires have to be considered early on inthe design process, and can no longer be treated as an afterthought as was most often the case in the past.

INDUCTIVE PARASITICS:

Interconnect wires also exhibit an inductive parasitic. An important source of parasitic inductance is introduced by the bonding wires and chip packages. Even for intermediate- speed CMOS designs, the current through the input- output connections can experience fast transitions that cause voltage drops as well as ringing and overshooting, phenomena not found in *RC* circuits. At higher switching speeds, wavepropagation and transmission line effects can come into the picture.

- **Inductance and Reliability-** $L\frac{di}{dt}\,\boldsymbol{Voltage\ Drop}$**:**

During each switching action, a transient current is sourced from (or sunk into) the supply rails to charge (or discharge) the circuit capacitances as shown. Both V_{DD} and V_{SS} connections are routed to the external supplies through bonding wires and package pins and possess a non ignorable series inductance. Hence, a change in the transient current creates a voltage difference between the external and internal (*V'DD*, GND') supply voltages. This situation is especially severe at the output pads, where the driving of the large external capacitances generates large current surges. The deviations on the internal supply voltages affect the logic levels and result in reduced noise margins.

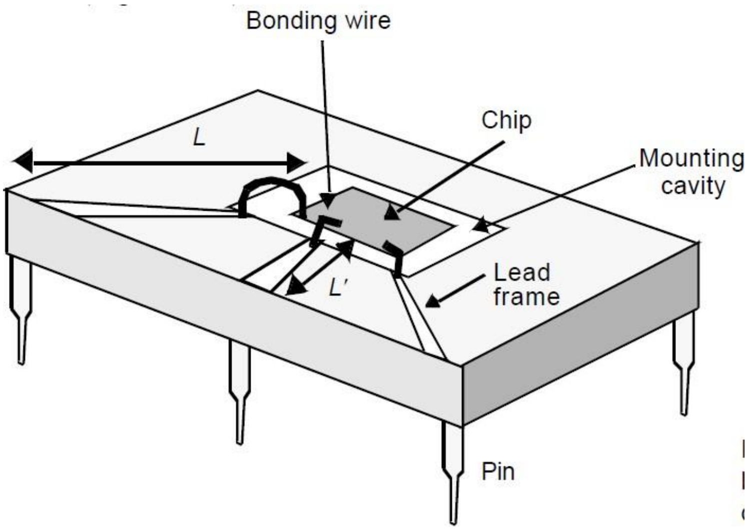


INDUCTIVE COUPLING BETWEEN EXTERNAL AND INTERNAL SUPPLY VOLTAGES

In an actual circuit, a single supply pin serves a large number of gates or output drivers. A simultaneous switching of those drivers causes even worse current transients and voltage drops. As a result, the internal supply voltages deviate in a substantial way from the external ones. For instance, the simultaneous switchingof the 16 output drivers of an output bus would cause a voltage drop of at least 1.1 V if the supply connections of the buffers were connected to the same pin on the package. Improvements in packaging technologies are leading to ever-increasing numbers of pins per package. Packages with up to 1000 pins are currently available. Simultaneous switching of a substantial number of those pins results in huge spikes on the supply rails that are bound to disturb the operation of the internal circuits as well as other external components connected to the same supplies.

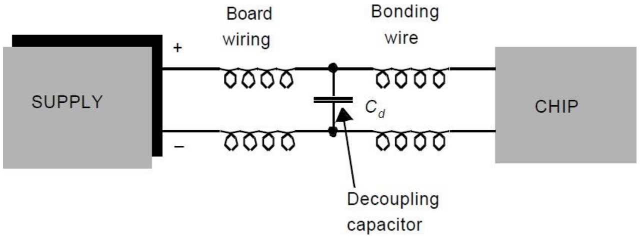
Design techniques to address $\boldsymbol{L\frac{di}{dt}}$

1. Separate pins for I/O pads and chip core. Since the I/O drivers require the largest switching currents, they also cause the largest current changes. Therefore, it is wise to isolate the core of the chip where most of the logic action occurs, from the drivers by providing different power and ground pins.
2. Multiple power and ground pins in order to reduce the per supply pin, we can restrict the number of I/O drivers connected to a single supply pin.
3. Careful selection of positions of the power and ground pins on the package. The inductance of pins located at the corners of the package is substantially higher as shown below.



THE INDUCTANCE OF A BONDING WIRE/ PIN COMBINATION DEPENDS UPON THE PIN POSITIONS

4. Schedule current consuming transitions so that they do not occur simultaneously.
5. Increase the rise and fall times of the off-chip signals to the maximum extent allowable, and distributed all over the chip, especially under the data busses.
6. Use advanced packaging technologies such as surface-mount or hybrids that come with a substantially reduced capacitance and inductance per pin.
7. Adding decoupling capacitances on the board. These capacitances, which should be added for every supply pin, act as local supplies and stabilize the supply voltage seen by the chip. They separate the bonding- wire inductance from the inductance of the board interconnect as shown below. The bypasscapacitor, combined with the inductance, actually acts as a low- pass network that filters away the high-frequency components of the transient voltage spikes on the supply lines.



DECOUPLING CAPACITORS ISOLATE THE BOARD INDUCTANCE FROM THE BONDING WIRE AND IN INDUCTANCE

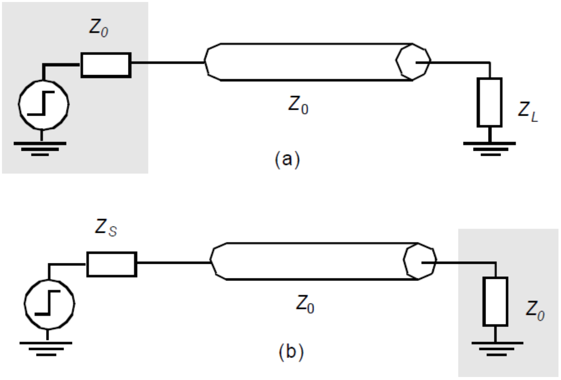
- **Inductance and Performance- Transmission Line Effects:**

When an interconnection wire becomes sufficiently long or when the circuits become sufficiently fast, the inductance of the wire starts to dominate the delay behaviour, and transmission line effects must beconsidered. This is more precisely the case when the rise and fall times of the signal become comparable to the time of flight of the signal waveform across the line as determined by the speed of light. As advancing technology increases line lenghts and switching speeds, this situation is gradually becoming common in fastest CMOS circuits as well, and transmission- line effects are bound to become a concern of the CMOS designer as well.

Some of the techniques to minimize the impact of the transmission line behaviour are:

○ **Termination:**

To avoid the negative effects of transmission-line behaviour such as ringing or slow propagation delays, the line should be terminated, either at the source (series termination), or at the destination (parallel termination) with a resistance matched to its characteristic impedance Z_0.



MATCHED TERMINATION SCENARIOS FOR WIRES BEHAVING AS TRANSMISSION LINES: (a) SERIES TERMINATION AT THESOURCE, (b) PARALLEL TERMINATION AT THE DESTINATION

The two scenarios- series and parallel termination as shown are depicted in figure. Series termination requires that the impedance of the signal source is matched to the connecting wire. This approach is appropriate for many CMOS designs, where the destination load is purely capacitive. The impedance of the driver inverter can be matched to the line by careful transistor sizing.

○ **Shielding**

If we want to control the behaviour of a wire behaving as a transmission line, we should carefully plan and manage how the return current flows. A good example of a well-defined transmission line is the coaxial cable, where the signal wire is surrounded by a cylindrical ground plane. To accomplish similar effects on a board or on a chip, designers often surround the signal wire with ground (supply) planes and shielding wires. Being shielding, adding shielding makes the behaviour and the delay of an interconnection a lot more predictable. Yet even with these precautions, powerful extraction and simulation tools will be needed in the future for the high-performance circuit designer.

UNIT – IV

INTERCONNECT, MEMORY ARCHITECTURE AND ARITHMETICCIRCUITS
TWO MARK QUESTIONS AND ANSWERS

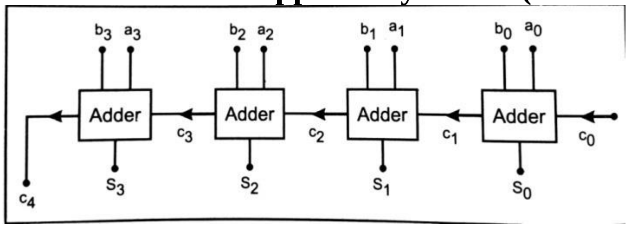
1. What is meant by data path circuits? (APR 2016)

- Data path circuits are meant for passing the data from one segment to other segment for processing or storing.
- The data path is the core of processors, where all computations are performed.

2. What is ripple carry adder?

- If n bits are added, then we can get n-bit sum and carry of C_n. C_i= Carry in bit from the previous column. N bit ripples carry adder needs n full adders with C_{i+1} carry out bit.

3. Draw the circuit for 4 bit ripple carry adder. (NOV 2018)



4. Write the equation for total delay in 4 bit ripple carry adder.

The total delay using the following equation,

$t_{4b} = t_d\ (c_{in} \rightarrow\ S_3)\ +2t_d\ (c_{in} \rightarrow c_{out}) + t_d\ (a_0,\ b_0\ \rightarrow c_1)$

5. Write the equation for worst case delay in 4 bit ripple carry adder.

If it is extend to n-bit, then the worst case delay is

$t_{n\text{-bit}} = t_d(c_{in} \rightarrow S_{n-1}) + (n-2)t_d(c_{in} \rightarrow c_{out}) + t_d(a_0,b_0 \rightarrow c_1)$

6. What is meant by Carry Lookahead Adder (CLA)?

- A carry-lookahead adder (CLA) or fast adder is a type of adder used in digital logic.
- A carry-lookahead adder improves speed by reducing the amount of time required to determine carry bits.
- The carry-lookahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger value bits.

7. Write the general expression for carry signal in CLA.
Write the full adder output interms of propagate and generate. (April 2018)

- We can write carry look-ahead expressions in terms of the generate gi and propagate pi signals. The general form of carry signal ci thus becomes

$c_{i+1} = a_i.b_i + c_i.(a_i \oplus b_i) = \mathrm{g}_\mathrm{i} + c_i \cdot \mathrm{p}_\mathrm{i}$, If $a_i.b$ =1, then $c_{i+1} =$ 1,

8. Write the equation for generate term in CLA.

- In the case of binary addition, A + B generates if and only if both A and B are 1. If we write G(A,B) to represent the binary predicate that is true if and only if A + B generates, write generate term as, $g_i = a_i.b_i$

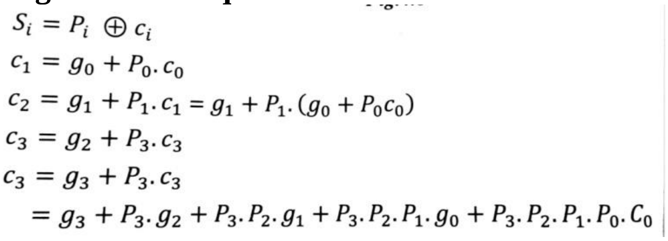
9. Write the equation for propagates term in CLA.

- In the case of binary addition, A + B propagates if and only if at least one of A or B is 1. If we write P(A,B) to represent the binary predicate that is true if and only if A + B propagates, we have:Write the propagate term as, $p_i = a_i \oplus b_i$

10. What are the two factors that Carry lookahead adder depends on?

- Carry lookahead depends on two things:
  - Calculating, for each digit position, whether that position is going to propagate a carry if one comes in from the right.
  - Combining these calculated values to be able to deduce quickly whether, for each group of digits, that group is going to propagate a carry that comes in from the right.

11. Write the generalized equation for CLA.



$$S_i = P_i \oplus c_i$$
$$c_1 = g_0 + P_0.c_0$$
$$c_2 = g_1 + P_1.c_1 = g_1 + P_1.(g_0 + P_0c_0)$$
$$c_3 = g_2 + P_3.c_3$$
$$c_3 = g_3 + P_3.c_3$$
$$= g_3 + P_3.g_2 + P_3.P_2.g_1 + P_3.P_2.P_1.g_0 + P_3.P_2.P_1.P_0.C_0$$

12. Name the limitations of MODL.

- MODL has following limitations as
  i. clocking in mandatory
  ii. The output is subject to charge leakage and charge sharing.
  iii. Series connected nFET chains can give long discharge times.

13. What is called Manchester Carry Chain Adder?

- The Manchester carry chain is a variation of the carry-lookahead adder that uses shared logic to lower the transistor count.
- As seen in CLA implementation section, the logic for generating each carry contains all of the logic used to generate the previous carries.
- A Manchester carry chain generates the intermediate carries by tapping off nodes in the gate that calculates the most significant carry value**.**

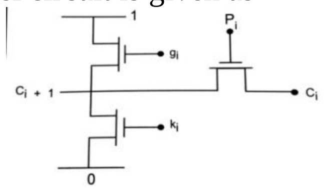
14. Write the basic equation for Manchester Carry Chain Adder?
Define kill term, propagate and generate term in a carry look ahead adder. (April 2019)

- In this adder, the basic equation is $c_{i+1} = \mathrm{g}_\mathrm{i} + c_i.\mathrm{p}_\mathrm{i}$

  Where $p_i = a_i \oplus b_i$ and $g_i = a_i.b_i$
- Carry kill bit $k_i = \overline{a_i + b_i} = \overline{a_i}.\overline{b_i}$
- If K_i=1, then p_i=0 and g_i=0. Hence, k_i is known as carry kill bit.

15. Draw the switch level circuit for Manchester carry chain adder.

The switch level circuit is given as



16. What are high (wide) adders?

- The adders with more than 4 bits can be designed. This is known as wide or high speed adders. Brute-force approach can be used to design 8 bit adder.

17. What are the types of high speed adders?

Types of high speed adders are

1. Carry Skip adder
2. Carry Select adder
3. Carry Save adder.

18. What is Carry skip adder?

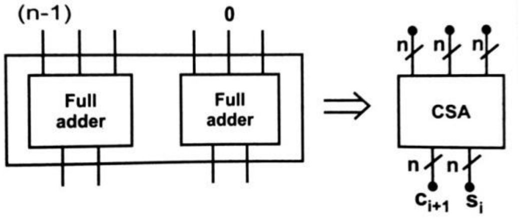
- Carry skip adder is one of the high speed adders.
- When BP= $P_0P_1P_2P_3$ =1, the incoming carry is forwarded immediately to the next block.
- Hence the name carry bypass adder or carry skip adder.
- Idea: if (P_0 and P_1 and P_2 and P_3 =1) the C_{03} = C_0, else “kill” or “generate”.

19. What is Carry Select adder?
Write the principle of any one fast multiplier. (NOV 2016)

- Adding two n-bit numbers with a carry-select adder is done with two adders in order to perform the calculation twice.
- After the two results are calculated, the correct sum, as well as the correct carry-out, is then selected with the multiplexer once the correct carry-in is known.

20. What is Carry save adder?

- In carry save adder, the carry does not propagate. So, it is faster than carry propagate adder.
- It has three inputs and produces 2 outputs, carry-out is saved. It is not immediately used to find the final sum value.



n-bit Carry Save Adder

21. What are accumulators?

- Accumulator acts as a part of ALU and it is identified as register A. The result of an operation performed in the ALU is stored in the accumulator.
- It is used to hold the data for manipulation (arithmetic and logical)

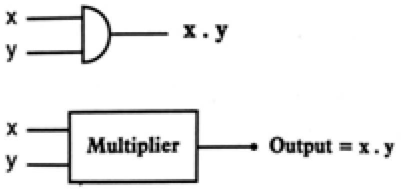
22. What are multipliers?

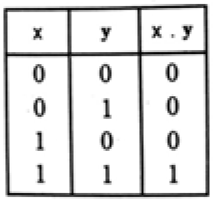
- Multiplier is used in computation process, which multiplies two binary numbers.
- Basic operations in multiplication are given below.
  0 x 0 = 0, 0 x 1 = 0, 1 x 0 = 0, 1 x 1 = 1

.

23. Draw the truth table of multiplier.

The truth table of multiplier is





x	y	x . y
0	0	0
0	1	0
1	0	0
1	1	1

24. Mention the steps involved in multiplying by shifting.

- If x=(0010)2 = (2)10
- If it is to be multiplied by 2, then we can shift x in left side. x = (0100)2 = (4)10
- If it is to be divided by 2, then we can shift in right side. x = (0001)2 = (1)10.
- So, shift register can be used for multiplication or division by 2.

25. Write the delay equation for array multiplier.

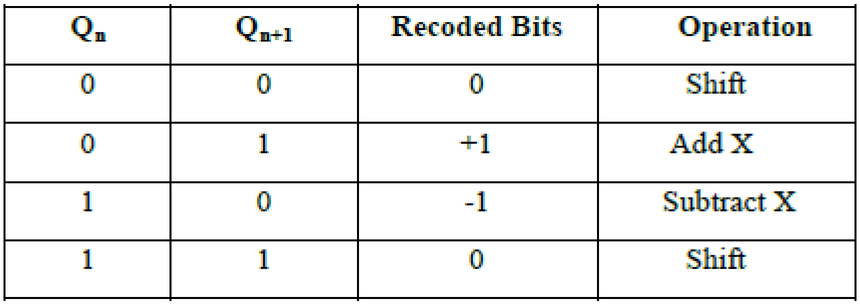
The equation for array multiplier is

$$P_i = \sum_{i=j+k} x_j y_k + c_{i-1}$$

$$P = X.Y = \left(\sum_{j=0}^{n-1} x_j 2^j\right)\left(\sum_{k=0}^{n-1} y_k 2^k\right) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\left(x_j y_k 2^{j+k}\right)$$

26. State radix-2 booth encoding table. (April 2019)

In radix-2 booth multiplication partial product generation is done based on encoding which is as given by Table.



Q_n	Q_{n+1}	Recoded Bits	Operation
0	0	0	Shift
0	1	+1	Add X
1	0	-1	Subtract X
1	1	0	Shift

Table: Booth encoding table with RADIX-2

27. What is meant by divider circuit?

- Divider circuit is used in arithmetic operation in digital circuits. Dividing is carry out by repeated subtraction and addition.

28. What are the types of dividers available in VLSI?

There are two types of dividers. They are serial divider and parallel divider.

29. Compare serial divider and parallel divider.

- Serial divider is slow and parallel divider is fast in performance. Array divider is fast compared with the serial divider. But hardware requirement is increased.

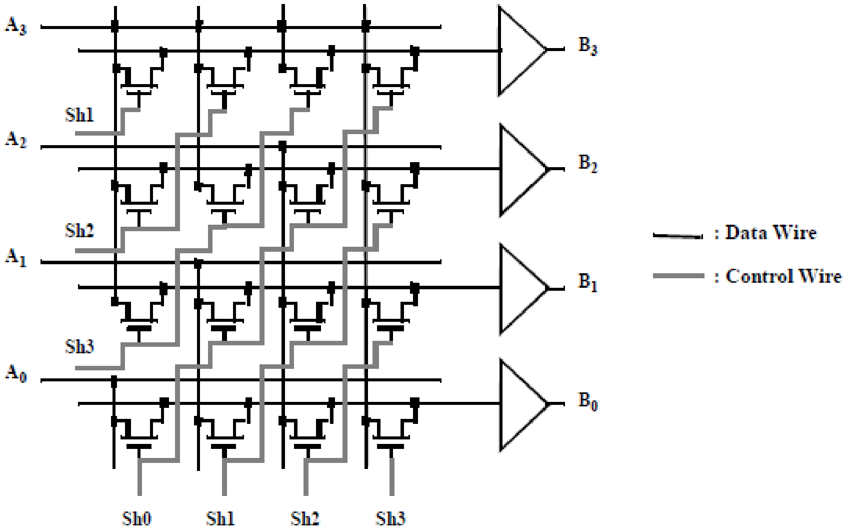
30. What is shift register?

- An n-bit rotation is specified by using the control word R_{0-n} and L/R bit defines a left or right shifting.
- For example y3 y 2 y 1 y 0 = a3 a2 a1 a0
  If it is rotated 1-bit in left side, we get y3 y 2 y 1 y 0 = a2 a1 a0 a3
  If it is rotated 1-bit in right side, we get y3 y 2 y 1 y 0 = a0 a3 a2 a1

31. What is meant by Barrel shifter?

- A barrel shifter is a digital circuit that can shift a data word by a specified number of bits in one clock cycle.
- It can be implemented as a sequence of multiplexers (MUX). The output of one MUX is connected to the input of the next MUX in a way that depends on the shift distance.

32. Draw the structure of 4 X 4 barrel shifter. (April 2018)



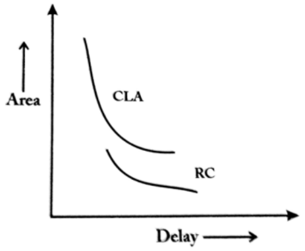
33. What is the area constraint between carry lookahead adder and ripple carry adder?

- The area of a carry lookahead adder is larger than the area of a ripple carry adder.
- Carry lookahead adder are parallel, which requires a larger number of gates and also results in a larger area.

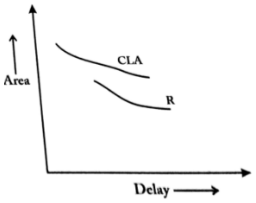
34. What is the drawback of carry lookahead adder?

- In the carry lookahead adder, need large area because computations are in parallel and more power is consumed.

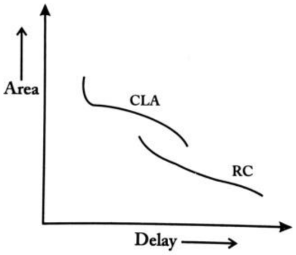
35. Draw the graph between area Vs delay of carry lookahead and ripple carry adder for 8 bit.



36. Draw the graph between area Vs delay of carry lookahead and ripple carry adder for 16 bit.



37. Draw the graph between area Vs delay of carry lookahead and ripple carry adder for32 bit.



38. What is meant by bit – sliced data path organization?(May 2016)

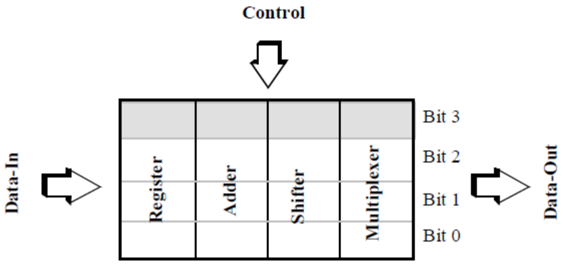
- Datapaths are arranged in a bit sliced organization, instead of operating on single bit digital signals.
- The data in a processor are arranged in a word based fashion. Bit slices are either identical or resemble a similar structure for all bits.

39. Determine propagation delay of n-bit carry select adder. (May 2016)

✓ Propagation delay, P of n-bit carry select adder is equal to $\sqrt{2N}$ where N = N- bit adder

40. Draw and list out the components of data path. (May 2017)

- Data path block consists of arithmetic operation, logical operation, shift operation and temporary storage of operands.



41. Mention the application of Barrel shift register.
Why is barrel shifter very useful in the designing of arithmetic circuits? (NOV 2016)

- A common usage of a barrel shifter is in the hardware implementation of floating-point arithmetic.
- For a floating-point add or subtract operation, requires shifting the smaller number to the right.
- This is done by using the barrel shifter to shift the smaller number to the right by the difference, in one cycle.

42. What is latency? (Nov 2017)

- Clock latency (or clock insertion delay) is defined as the amount of time taken by the clock signal in traveling from its source to the sinks.

43. Give the applications of high speed adder. (May 2017)

- CMOS high speed adder adders used in processor, data processing application and data path application with low power consumption.

44. What is meant by booth multiplier?

- Booth's algorithm is an efficient hardware implementation of a digital circuit that multiplies two binary numbers in two's complement notation.
- Booth multiplication is a fastest technique that allows for smaller, faster multiplication circuits, by recoding the numbers that are multiplied.

45. What is meant by array multiplier?

- Array multiplier uses an array of cells for calculation.
- Multiplier circuit is based on repeated addition and shifting procedure. Each partial product is generated by the multiplication of the multiplicand with one multiplier digit.
- N-1 adders are required where N is the number of multiplier bits.

46. What is Wallace tree multiplier? And give its advantages.

The Wallace tree multiplier output structure is tree basis style. It reduces the number of components and reduces the area.

47. What are parameters used to characterize the memory?

Parameters used to characterize a memory device are area, power and speed.

48. How can you classify memory based on operation mode?

Classifications of memory based on operation mode are 1. ROM , 2. RAM

49. How can you classify memory based on data storage mode?

Classifications of memory based on data storage mode are 1. Volatile, 2. Non-volatile

50. Define ROM. Give some examples.

ROM is a memory where code is written only one time. Examples are washing machine, calculator, games etc.

51. What are advantage and disadvantages of programming ROM?

Advantage: basic cell only consists of transistor. No need of connection to any of the supply voltage.

Disadvantage: As it has pseudo nMOS, it is ratioed logic and consumes static power.

52. What is meant by non-volatile memory?

Non-volatile consists of array of transistors. These are placed on a word line – bit line grid. We can write the program by enabling or disabling these devices selectively.

53. What is floating gate transistor?

Floating gate transistor is mostly used in all the reprogrammable memories. In floating gate transistor, extra polysilicon strip is used in between the gate and the channel known as floating gate.

54. What is RAM? And give types of RAM.

RAM is read and write memory. Types are static and dynamic RAM.

55. Distinguish Static and dynamic RAM.[Nov/Dec 2022]

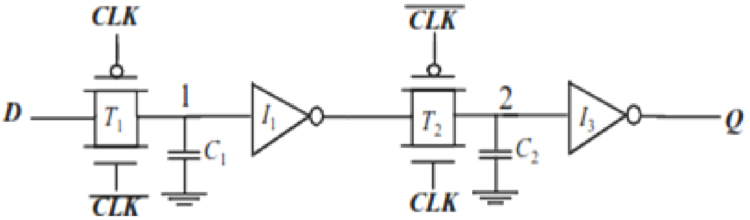
Static RAM:

SRAM cell needs 6 transistors per bit. Bit line (BL) and inverse Bit Line signals are used to improve the noise margin during read and write operations.

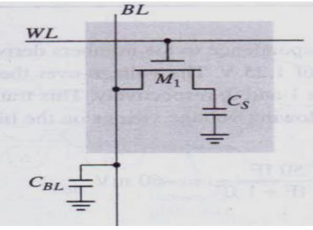
Dynamic RAM:

Content in the cell can be periodically rewritten through a resistive load, called as refresh operation. Here cell content is read follow by write operation.

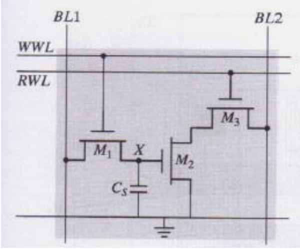
56. Draw the schematic of dynamic edge –triggered register. (Dec. 2016)



57. Design a one transistor DRAM cell. (Nov 2013, April 2015)
Draw a 1-transistor Dynamic RAM cell. (April 2019) [Nov/Dec 2022]



58. Design a three transistors DRAM cell.



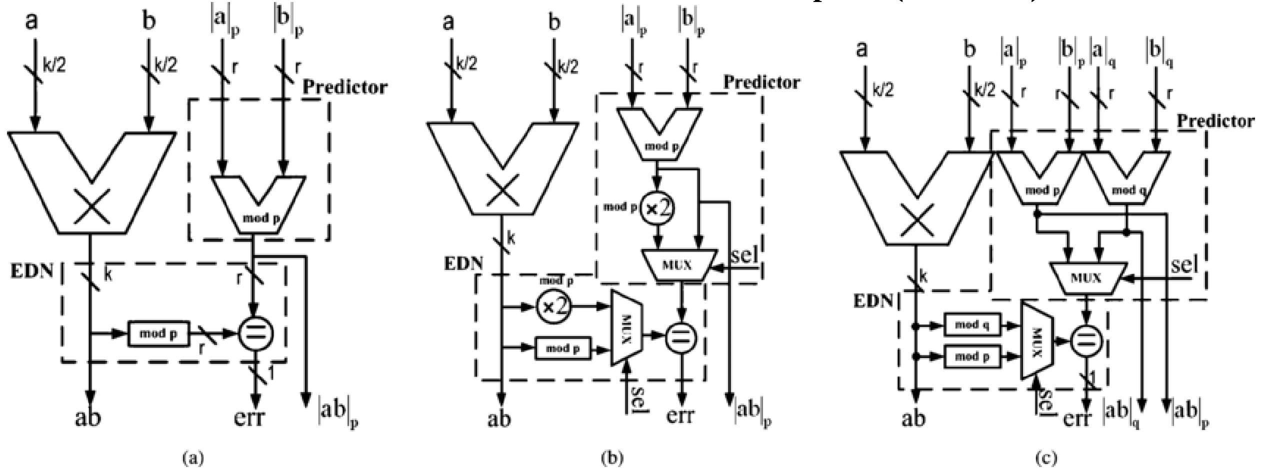
59. State the merits of barrel shifter. (Nov 2019)

- It has small area and does not require a decoder
- Logarithmic shifter is more effective for larger shift values in terms of both area and speed.

60. How to design a high speed adder? (Nov 2017)

- Design of high speed adder using CMOS and transmission gates in submicron.
- Design of high speed adder using parallel adder manner.

61. Mention the different hardware architecture used for multiplier. (Nov 2019)



Hardware architectures for multipliers protected by (a) linear arithmetic codes, (b) [jxj ; j2xj] multilinear codes, and (c) multi-modulus multilinear codes.

62. Draw the dot diagram for Wallace tree multiplier. [May 2021]

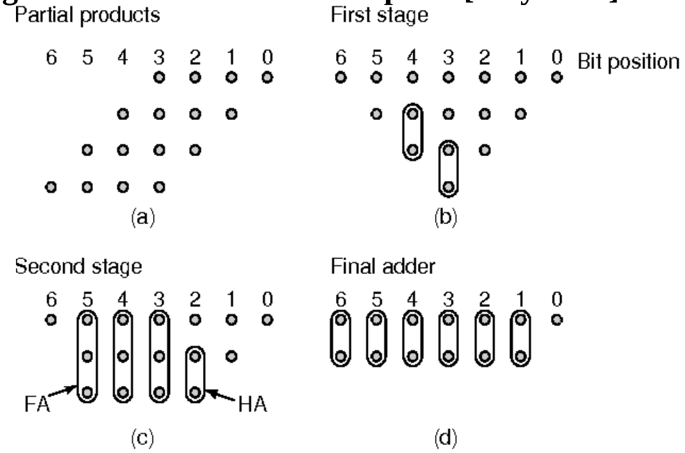


Figure 2: Dot diagram of 4-bit Wallace tree multiplier

63. List the categories of memory arrays. [May 2021]

<u>Classification based on operation mode:</u>

1. ROM 2. RAM

<u>Classification based on data storage mode</u>:

It means on how it is stored and how long it remains there.

1. Volatile 2. Non-volatile

<u>Classification based on access method</u>:

1. Random access 2. Non-random access

64. State the need of a sense amplifier in a memory cell. (NOV 2021)[Apr/May 2022]

It senses the low power signals from a bitline that represents a data bit (1 or 0) stored in a memory cell, and amplify the small voltage swing to recognizable logic levels so the data can be interpreted properly.

65. What are the building blocks of digital architecture?

These blocks include arithmetic circuits, counters, shift registers, memory arrays, and logic arrays.

66. Mention the steps for single bit addition.

1-Bit Adder (Half Adder)

- The simplest case arises when two one bit numbers are to be added.
- With one bit, only the numbers 0 and 1 can be represented.
- All possible scenarios can be summarized by the following table:

UNIT – IV

INTERCONNECT, MEMORY ARCHITECTURE AND ARITHMETICCIRCUITS

Question Bank

1. **Design a multiplier for 5 bit by 3 bit. Explain its operation and summarize the numbers of adders. Discuss it over Wallace multiplier.**
2. Describe the different approaches of improving the speed of the adder.
3. **How the drawback in ripple carry adder overcome by carry look ahead adder and discuss.**
4. Draw the ripple carry adder & explain its operation.
5. **Design a carry bypass adder and discuss its features.**
6. **Design 4 input and 4 output barrel shifter using NMOS logic.**
7. Explain in detail about the interconnect.
8. Describe about interconnect modelling.

UNIT V ASIC DESIGN AND TESTING

Introduction to wafer to chip fabrication process flow. Microchip design process & issues in test and verification of complex chips, embedded cores and SOCs, Fault models, Test coding. ASIC Design Flow, Introduction to ASICs, Introduction to test benches, Writing test benches in Verilog HDL, Automatic test pattern generation, Design for testability, Scan design: Test interface and boundary scan.

Introduction:

- **ASIC** - Application Specific Integrated Circuit is an Integrated Circuit (IC) designed to perform a specific function for a specific application.
- **Levels of integration:**
  The levels of integration are:
  - ✓ SSI - Small scale integration
  - ✓ MSI - Medium scale integration
  - ✓ LSI - Large scale integration
  - ✓ VLSI - Very large scale integration
  - ✓ USLI - Ultra large scale integration
- **Implementation technology**
- The implementation technologies used in ASIC are:
  - ✓ TTL – Transistor Transistor Logic
  - ✓ ECL – Emitter Coupled Logic
  - ✓ MOS – Metal Oxide Semiconductor (NMOS, CMOS)

5.1: Types of ASICs

- ❖ **Explain about different types of ASICs with neat diagram. (April 2016, 2017, 2018)**
- ❖ **Write brief notes on: (a) Full custom ASIC (b) Semi custom ASIC (May 2010, May 2016)**
- ❖ **Compare the different types of ASICs. (Nov 2007, Nov 2008)**

- The ASICs are classified as follows:

I. Full-Custom ASICs
II. Semi-custom ASICs
  a. Standard-Cell–Based ASICs (CBIC)
  b. Gate-Array–Based ASICs (MPGA)
    i. Channeled Gate Array
    ii. Channelless Gate Array
    iii. Structured Gate Array
III. Programmable ASICs
  a. Complex Programmable Logic Devices (CPLD)
  b. Field-Programmable Gate Arrays (FPGA)

5.1.1:Full-Custom ASICs

Explain the full custom ASICs.

- In full custom ASIC, engineer can design full logic cells in IC. So, this technique is known as Full custom ASIC technique.

- Engineer uses mixed analog and digital technique to manufacture IC. All the logic cells are specifically designed for one ASIC.

Uses of bipolar technology:

- The characteristics of bipolar components in the same IC are matched very well.
- But the characteristic of components in different IC are not matched well

Uses of CMOS:

- This is widely used technology to manufacture IC.
- Mixing of analog and digital function are integrated in the same IC for which CMOS technology suits well.
- Designers give importance to performance.
- When large volume is manufactured, overall cost will be reduced.
- In super computer, quality is important so this design is implemented.
- All mask layers are customized in a full-custom ASIC
- Generally, the designer lays out all cells by hand
- Some automatic placement and routing may be done
- Critical (timing) paths are usually laid out completely by hand
- Full-custom design offers the ***highest performance and lowest part cost*** (smallest die size) for a given design.
- The ***disadvantages*** of full-custom design include ***increased design time, complexity, design expense***, and ***highest risk.***
- Microprocessors (strategic silicon) were exclusively full-custom, but designers are increasingly turning to semicustom ASIC techniques in this area as well.

5.1.2: Semi-custom ASICs – Design

❖ **Briefly explain the semi-custom Asics with its classification. (May 2016, NOV 2016)**

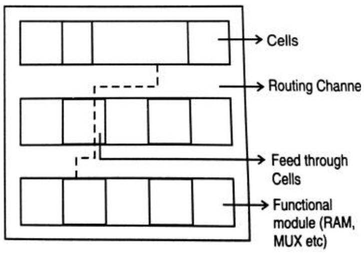
I. Standard cell based design:

- Standard cells are referred to AND gate, OR gate, multiplexer, flip flop, NOR gate etc.
- Standard cells can be used with larger predefined cells.
- This approach standardizes design entry level at logic gate.
- A design is generated automatically from HDL language.
- Then layout is created. In standard cell design, cells are placed in rows, and rows are separated by routing channel.
- All cells in library are in identical heights, widths of the cells can be varied to accommodate for variations in complexity between cells.
- A substantial fraction of area is allotted for signal routing.
- The minimization of interconnect overhead is most important goal of standard-cell placement routing tools. It is done by **feed through cells**.
- By using **feed through cell**, cells in different rows can be connected through vertical routing. So length of wire is reduced by feed through cells.

Semi-custom ASICs – CBIC

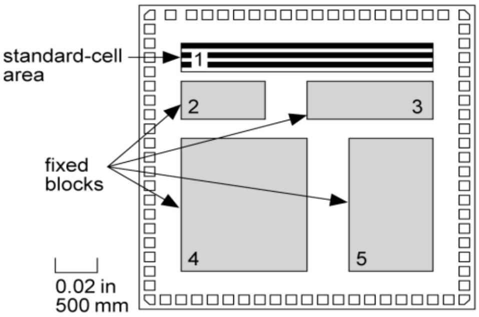
- CBIC means Cell Based ASICs.
- All the mask layers of CBIC are customized.

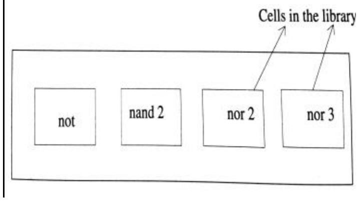
- It allows mega cells (SRAM, MPEG, decoder etc) to be placed in the same IC with standard cells (adder, gates etc).
- Mega cells are supplied by ASIC Company.
- Data path logic means the logic that operates on multiple signals across a data bus.
- Some of the ASIC library companies provide data path compiler which automatically generate data path logic.
- Data path library contains cells like adders, multiplexer, simple ALUs.
- ASIC Library Company provide data book which has functional description.



Features:

- It is a cell-based ASIC (CBIC —“sea-bick”)
- It has Standard cells. Standard cell is logic elements used CMOS technology.
- Possibly megacells , megafunctions , full-custom blocks , system-level macros (SLMs), fixed blocks , cores , or Functional Standard Blocks (FSBs)
- All mask layers are customized - transistors and interconnect
- Automated buffer sizing, placement and routing. And custom blocks can be embedded.





- A “wall” of standard cells forms a flexible block.

II. Gate Array Based ASICs:

- ❖ **Explain gate array based ASICs with diagrams. (April 2008, May 2009)**
- Gate array is known as GA.
- In GA based ASIC, the transistors are predefined on the silicon wafer.
- *Base array*: the predefined pattern of transistors on a gate is known as base array.
- *Base cell:* the small element which is replicated to make the base array is known as base cell or primitive cell.
- *Masked Gate array*: Interconnect is defined by using top few layers of metal.
- This type of gate array is known as masked gate array.
- Gate array library is provided by ASIC Company.
- The designer can choose the predefined logic cells from a gate array library. These logic cells are known as **Macros**.
- Cell-layout is same for each logic cell. But interconnect is customized.
- It is also called as pre-diffused array because the transistors are diffused at first.

Types of MPGAs (Mask Programmable Gate Arrays):

- ✓ Channeled Gate Array
- ✓ Channel less Gate Array
- ✓ Structured Gate Array

(a) Channeled Gate Array:

- It is similar to CBIC (cell based ASIC).
- In the both types, rows of cells are separated by channels. These channels are used for interconnect.
- Space between rows of cells is fixed in a channeled gate array. But space between rows of cells may be adjusted in a CBIC.

Features:

- ✓ Only interconnect is customized.
- ✓ The interconnect uses predefined spaces between rows of base cells.
- ✓ Manufacturing lead time is between two days and two weeks.

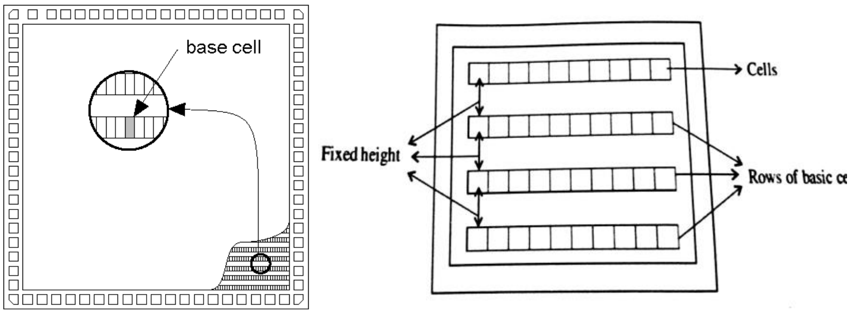


Figure: Channel Gate Array

(b) Channel less Gate Array:

- Channel less Gate Array is also called as channel free GA.
- In this array, there is no predefined space between rows for routing.
- Top few layers are used for defining interconnect connections.
- There are no predefined areas set aside for routing - routing is over the top of the gate-array devices.
- Achievable logic density is higher than for channeled gate arrays.
- Each logic cell or macro in a gate-array library is predesigned using fixed tiles of transistors known as the gate-array base cell (or just base cell).

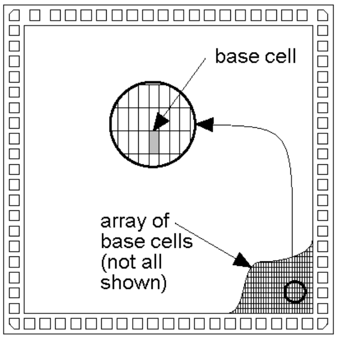


Figure: Channel less Gate Array

- Channeled and channelless gate arrays may use either gate isolation or oxide isolation.
- Isolate the transistors on a gate array from one another either with thick field oxide or by using other transistors that are wired permanently off.

(c) Structured Gate Array:

- Structured Gate Array is also called as embedded gate array or master slice or master image gate.
- It combines some of the features of CBIC and Masked gate array (MGA).
- In this array, some of the area is used for implementation of specially designed embedded block.
- Embedded area either can contain a different base cell that is more suitable for building memory cells, or a complete circuit block, such as a microcontroller.

Special features:

- Only the interconnect is customized
- Custom block can be embedded
- Manufacturing lead time is 2 days to 2 weeks
- Area efficiency is increased
- Performance is increased with low cost

Disadvantages: the embedded function is fixed.

- For ex: if embedded block has 32K bit memory. But the customer needs only 18K bit, the 16K memory is wasted.

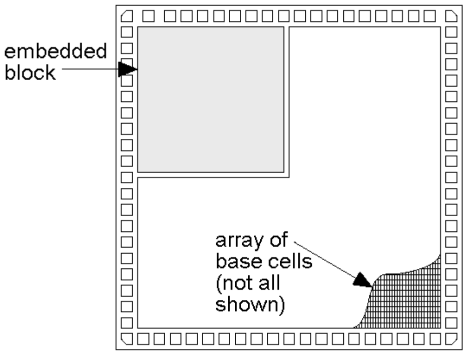


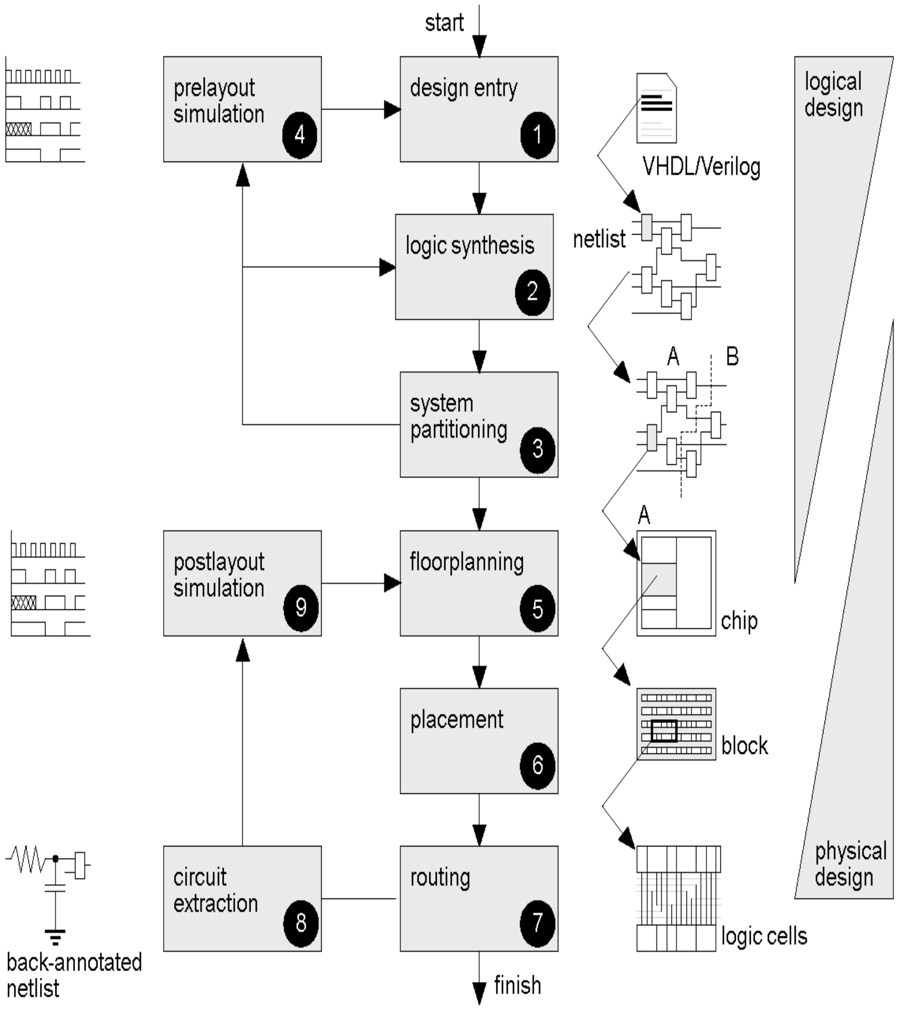
Figure: Structured Gate Array

****

5.2: ASIC Design Flow / Cycle

Explain the ASIC design flow with a neat diagram. (Nov 2007, April 2008, Nov 2008)
Draw the flowchart of digital circuit design techniques. (NOV 2018)

1. **Design entry** - Using a hardware description language (HDL) or schematic entry
2. ***Logic synthesis*** - Produces a netlist - logic cells and their connections
3. ***System partitioning*** - Divide a large system into ASIC-sized pieces
4. **Prelayout simulation** - Check to see if the design functions correctly
5. ***Floorplanning*** - Arrange the blocks of the netlist on the chip
6. ***Placement*** - Decide the locations of cells in a block
7. ***Routing*** - Make the connections between cells and blocks
8. ***Extraction*** - Determine the resistance and capacitance of the interconnect
9. **Post layout simulation** - Check to see the design still works with the added loads of the interconnect



****

5.3: ASIC Cell Libraries

Explain the standard Cell libraries in ASIC.
Give a note on standard cell design. (Nov 2019)

- The cell library is the key part of ASIC design.
- For a programmable ASIC the FPGA Company supplies a library of logic cells in the form of a design kit.
- For MGAs and CBICs you have three choices:
- ASIC vendor will supply a cell library, or you can buy a cell library from a third-party library vendor, or you can build your own cell library.
- The first choice, using an ASIC-vendor library, requires you to use a set of design tools approved by the ASIC vendor to enter and simulate your design.
- Some ASIC vendors (especially for MGAs) supply tools that they have developed in-house.
- An ASIC vendor library is normally a phantom library the cells are empty boxes, or phantoms, but contain enough information for layout.
- After you complete layout you hand off a netlist to the ASIC vendor,
- The second and third choices require making a buy-or-build decision.
- If complete an ASIC design using a cell library that you bought, you also own the masks that are used to manufacture your ASIC. This is called customer-owned tooling.

- A library vendor normally develops a cell library using information about a process supplied by an ASIC foundry.
- An ASIC foundry only provides manufacturing, with no design help. If the cell library meets the foundry specifications, we call this a qualified cell library.
- These cell libraries are normally expensive, but if a library is qualified at several foundries.
- The third choice is to develop a cell library in-house. Many large computer and electronics companies make this choice.
- However, created each cell in an ASIC cell library must contain the following:
    - ✓ A physical layout
    - ✓ A behavioral model
    - ✓ A Verilog/VHDL model
    - ✓ Detailed timing models.
    - ✓ A test strategy
    - ✓ A circuit schematic
    - ✓ A cell icon
    - ✓ A wire-load model
    - ✓ A routing models.

- The ASIC designer needs a high-level behavioral model for each cell.
- Because simulation at the detailed timing level takes too long for a complete ASIC design.
- The designer may require Verilog and VHDL models in addition to the models for a particular logic simulator.

****

5.4: Library-Cell Design

Explain the important of Library –cell design in detail.

- Design rules for each ASIC vendor are slightly different even for the same generation of technology.
- For example, two companies may have very similar 0.35 nm CMOS process technologies, but the third-level metal spacing might be slightly different.
- A library constructed in this fashion may not be competitive with one that is constructed specifically for each process.
- ASIC vendors prize their design rules as secret, it turns out that they are similar except for a few details.
- We would like all vendors to agree on a common set of design rules.
- The reason that most vendors have similar rules is because most vendors use the same manufacturing equipment and a similar process.
- Layout of library cells is either hand-crafted or uses some form of symbolic layout.
- Symbolic layout is usually performed in one of two ways: using either interactive graphics or text layout language.
- Shapes are represented by simple lines or rectangles, known as sticks, in a symbolic layout.
- The actual dimensions of the sticks are determined after layout is completed in a Post processing step.
- Graphical symbolic layout uses a text layout language, like a programming language such as C that directs a program to assemble layout.

****

CMOS Testing

Explain about Microchip design process.

Microchip design process:

The microchip design process involves several stages from conceptualization to production. Here is an overview of the typical steps involved:

1. **Specification:** In this stage, the requirements and functionality of the microchip are defined. Designers work closely with stakeholders to understand the application and performance targets.

2. **Architecture Design:** The chip's high-level architecture is developed, including the selection of components, interconnections, and overall system design. This stage focuses on defining the chip's functionality and how different components will interact.

3. **RTL Design:** The Register Transfer Level (RTL) design is created, describing the chip's behavior using hardware description languages like Verilog or VHDL. RTL design forms the basis for later stages.

4. **Functional Verification:** The RTL design is extensively tested to ensure it behaves as intended. Various verification techniques, such as simulation, formal verification, and hardware emulation, are employed to catch design bugs and issues.

5. **Synthesis and Physical Design:** The RTL code is synthesized into a gate-level netlist, which represents the chip's physical implementation. The physical design phase involves floor planning, placement, routing, and optimization to meet timing and area constraints.

6. **Design for Testability (DFT):** Techniques like scan chains, built-in self-test (BIST) structures, and boundary scan are added to make the chip more testable during manufacturing and in the field.

7. **Manufacturing:** The final design is sent to a semiconductor foundry for fabrication. This process involves photolithography and other steps to create the actual silicon chip.

8. **Testing and Quality Assurance:** After manufacturing, the chips undergo various testing methodologies to ensure they meet the desired specifications and are free from defects.

****

Explain the Issues in Test and Verification of Complex Chips, Embedded Cores, and SoCs:

Issues in Test and Verification of Complex Chips, Embedded Cores, and SoCs:

1. **Complexity:** As chips and systems-on-chip (SoCs) become more complex, the verification effort increases exponentially. Ensuring all possible scenarios and corner cases are covered in testing becomes challenging.

2. **Verification Time and Cost:** With the growing complexity, the time and cost required for functional verification can become substantial.

3. **Integration Testing:** Integrating various IP cores and subsystems onto a single chip or SoC introduces new challenges in testing the interactions between these components.

4. **Power and Clock Domains:** Handling multiple power domains and clock domains in a chip requires careful verification to ensure proper functionality and minimize power consumption.

5. **Performance Verification**: Ensuring that the chip operates at the desired performance levels under all conditions and workloads is crucial, especially for high-performance chips.

6. **Test Generation:** Generating effective and efficient test patterns to cover various fault models is a non-trivial task, especially for complex designs.

7. **Debugging:** Identifying and debugging issues in large and complex designs can be time-consuming and requires advanced debugging techniques.

Fault Models:

Fault models are representations of potential defects that can occur in a chip or design. Common fault models include:

1. **Stuck-at Faults:** These faults assume that a particular node in the circuit is stuck at either '0' or '1'.

2. **Transition Delay Faults:** These faults model timing-related issues, where a signal changes too slowly or too fast.

3. **Path Delay Faults:** These faults model delays along specific paths in the circuit.

4. **Bridge Faults:** These faults represent a short circuit between two nets or nodes.

5. **Cell-Aware Faults**: These are specific to certain types of cells and are critical for nanometer-scale technologies.

Test Coding:

Test coding involves writing test patterns to test the functionality and detect faults in a chip. Various methods and languages can be used for test coding, such as:

1. **ATPG (Automatic Test Pattern Generation):** ATPG tools automatically generate test patterns based on fault models.

2. **BIST (Built-In Self-Test):** BIST structures are embedded within the chip to facilitate self-testing.

3. **Scan Chains:** These enable efficient testing by serially scanning in test data and capturing results.

4. **Testbenches:** Testbenches are used for simulation-based verification, where test stimuli are applied to the design, and responses are analyzed.

5. **High-Level Test Languages:** Some specialized languages and tools are used for high-level test descriptions, which can be automatically converted to lower-level test patterns.

- In conclusion, designing and testing complex chips, embedded cores, and SoCs require a thorough understanding of various verification techniques, fault models, and test coding methods.
- As technology continues to advance, the challenges in test and verification continue to evolve, demanding innovative solutions and methodologies.

****

Explain about the test benches.
Introduction to test benches:

- Test benches are an essential part of digital hardware and software development, especially in the field of electronic design automation (EDA). They play a crucial role in verifying and validating the functionality of digital circuits, integrated circuits (ICs), and other electronic systems.
- A test bench serves as a virtual environment in which designers can simulate the behavior of their design, apply test stimuli, and observe the responses to ensure the correctness and functionality of the design before it is physically implemented or manufactured.

Here's an introduction to test benches:

1. **Purpose:** The primary purpose of a test bench is to thoroughly test the functionality of a digital design or electronic system before its physical implementation. It allows designers to catch and fix design errors and functional bugs in a controlled, simulated environment, reducing the risk of costly and time-consuming errors in the final product.

2. **Simulation Environment:** A test bench is created as a separate entity from the actual design being tested. It provides an environment that emulates the behavior of the design under test (DUT) and contains the necessary stimuli to drive inputs and monitor outputs.

3. **Simulation Types:** Test benches are used in various types of simulations, such as functional simulation, timing simulation, and power analysis. Each type of simulation focuses on different aspects of the design and provides valuable insights into its behavior.

4. **Test Stimuli:** In a test bench, test stimuli are applied to the inputs of the DUT to simulate different scenarios and conditions. These stimuli can be pre-defined patterns, random data, or specific corner cases to test the design's robustness.

5. **Output Monitoring:** The test bench also includes monitors that observe and record the DUT's outputs during the simulation. This allows designers to compare the expected outputs with the actual outputs to check for correctness.

6. **Debugging and Analysis:** Test benches facilitate debugging by providing detailed information about the DUT's behavior during simulation. Designers can analyze the waveform results to pinpoint errors and verify that the design meets the required specifications.

7. **Languages and Tools:** Test benches are typically written using hardware description languages (HDLs) like Verilog or VHDL. There are also higher-level verification languages, like SystemVerilog,

which provide more advanced features for test bench creation. EDA tools such as simulation tools (e.g., ModelSim, VCS) and hardware description and verification languages make the process of test bench creation more efficient and manageable.

8. **Coverage Analysis:** Test benches are instrumental in evaluating the functional coverage, code coverage, and other metrics to assess the effectiveness and completeness of the tests.

9. **Regression Testing:** As designs evolve, test benches can be used for regression testing, ensuring that any new changes or optimizations do not introduce new errors or regressions in the design.

- In summary, test benches are an integral part of the hardware and software development process, enabling designers to validate and verify digital designs through simulation.
- They are crucial for achieving high-quality, bug-free, and robust designs, leading to reduced development time and costs while ensuring the functionality and reliability of the final product.

****

1. **Explain the following: (NOV 2012)**

(i) Silicon debug principles. (MAY 2013, MAY 2014)

(ii) Fault models

Silicon debug principles

- A major challenge in silicon debugging is when the chip operates incorrectly. There are several techniques for directly accessing silicon.
- Specific signals can be brought to the top of the chip as probe points.
- These are small squares of top-level metal that connect to key points in the circuit.
- The designer has included before debug.
- The over glass cut mask should specify a hole in the passivation over the probe pads so the metal can be reliably contacted.
- The exposed squares can be probed with a Pico probe in a fixture under a microscope.
- The die can be probed electrically or optically if mechanical contact is not feasible.
- An electron beam (ebeam) probe uses a scanning electron microscope to produce a tightly focused beam of electrons to measure on-chip voltages.
- Laser Voltage Probing (LVP) involves shining a laser at a circuit and observing the reflected light. The reflections are modulated by the electric fields so switching waveforms can be deduced.
- Picosecond Imaging Circuit Analysis (PICA) captures faint light emission naturally produced by switching transistors.
- Silicon is partially transparent to infrared light, so both LVP and PICA can be performed through the substrate from the backside of a chip in a flip-chip package.
- Infrared (IR) imaging can be used to examine “hot spots” (a resistive short between power rails) in a chip.
- There are liquid crystal materials, which can be “painted on” to a die to indicate temperature problems. If the location of the fault is known, a Focused Ion Beam (FIB) can be used to cut wires or lay new conductors down.

(a) Types of failures:

- There are three types of failures, manufacturing, functional and electrical failure.
- Manufacturing failures occur when a chip has a defect or is outside of parametric specifications.
- Functional failures are logic bugs or physical design errors that cause the chip to fail under all conditions.
- Electrical failures occur when the chip is logically correct, but malfunctions under certain conditions such as voltage, temperature, or frequency.

(b) Shmoo Plot

- Shmoo plot is used to debug electrical failures in silicon.
- A shmoo plot is a plot with voltage on one axis and speed on the other.
- The test vectors are applied at each combination of voltage and clock speed, and the success of the test.
  is recorded.
- A shmoo can also plot operating speed against temperature.
- At cold temperatures, FETs are faster, have lower effective resistance, and have higher threshold voltages. Failures at low temperature could indicate coupling or charge sharing noise.
- Failures at high temperatures could indicate excessive leakage or noise problems.

2. Explain the manufacturing test principles in detail. (NOV 2011, NOV 2012, NOV 2013) Explain the chip level test techniques. (NOV 2007, MAY 2008, NOV 2021)

Manufacturing test principles

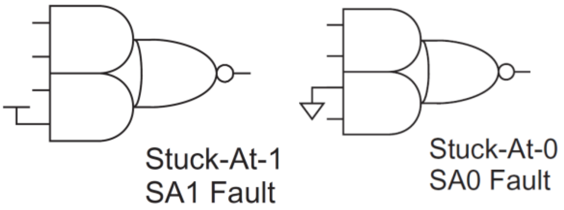
- The purpose of the manufacturing test is to screen out the defective parts before they are shipped to customers.

(a)Fault models

- To deal with the existence of good and bad parts, it is necessary to propose a fault model, i.e., a model of how faults occur and their impact on circuits.

(i) Stuck at faults.

- In the Stuck-At model, a faulty gate input is modeled as a stuck at zero (Stuck-At-0, S-A-0) or stuck at one (Stuck-At-l, S-A-l).
- These faults most frequently occur due to gate oxide shorts (the nMOS gate to GND or the pMOS gate to VDD) or metal-to-metal shorts.



(ii) Short circuit fault

- Two bridging or shorted faults are shown in Figure.
- The short S1 results in an S-A-0 fault at input A, while short S2 modifies the function of the gate.
- For instance, in the case of a simple NAND gate, the intermediate node between the series nMOS transistors is hidden by the schematic.
- The probable bridging fault in CMOS circuit can be grouped into three categories. There are metal polysilicon short, polysilicon n-diffusion short and polysilicon p-diffusion short.

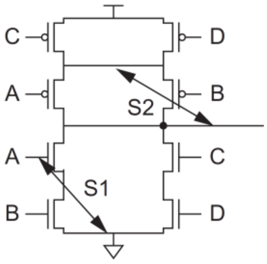


Figure: Bridging fault / Short circuit fault

(iii) Open circuit fault

- A fault can convert a combinational circuit into a sequential circuit. This is illustrated in Figure for the case of a 2-input NOR gate.
- If nMOS transistor A is stuck open, then the function displayed by the gate will be $Z = \overline{A+B} + \bar{B}Z'$ where Z' is the previous state of the gate.
- Stuck closed states can be detected by observing the static VDD current (IDD) while applying test
  vectors.

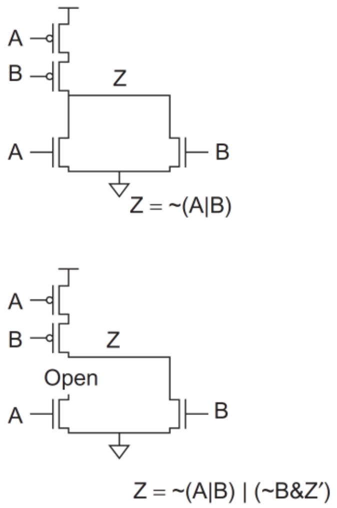


Figure: Open circuit fault

(b) Observability

- The observability of a particular circuit node is the degree to which can observe that node at the outputs of an integrated circuit (i.e., the pins).
- This metric is relevant, when want to measure the output of a gate within a larger circuit to check that it operates correctly.
- Given the limited number of nodes that can be directly observed, it is the aim of good chip designers to have easily observed gate outputs.

(c) Controllability

- The controllability of an internal circuit node within a chip is a measure of the ease of setting the node to a 1 or 0 state.
- This metric is of importance when assessing the degree of difficulty of testing a particular signal within a circuit.
- An easily controllable node would be directly settable via an input pad.

(d) Repeatability

- The repeatability of system is the ability to produce the same outputs given the same inputs.

(e) Survivability

- The survivability of a system is the ability to continue function after a fault. For example, error-correcting codes provide survivability in the event of soft errors.

(f) Fault coverage

- A measure of goodness of a test program depend the amount of fault coverage by the test program.
- The fault coverage of a set of test vectors is the percentage of the total nodes that can be detected as faulty when the vectors are applied.
- Each circuit node is taken in the sequence and held to S_a_0, and then simulation started. The chip’s outputs are compared with outputs of good machine.

- If the outputs of IC are not matched with the outputs of good, and then fault is marked and the simulation is stopped.
- The same procedure is repeated to set the node to logic 1. This method is known as sequential fault grading.
- Fault coverage is defined as ratio of the number of nodes detected as faults and total number of nodes in the circuit.

(g) Automatic Test Pattern Generation (ATGP)

- If want to test the gate which is embedded in large logic circuit, use existing circuit to create a specific path from the location of gate which is going to be checked finding fault.
- This technique is known as path sensitization. This process of creating the path is known as propagation.

****

3. Explain with diagram the design strategies for testing the CMOS devices. (NOV 2008, NOV 2009)

Write briefly about different test strategies of testing digital circuits. (MAY 2009)

Explain any two approaches of DFT (Design for Testability) in brief with example. (MAY 2010, NOV 2009, MAY 2013)[Apr/May 2022]

Explain the three main approaches commonly used for design for testability (DFT). [May 2021]

Design for Testability

- The keys to designing circuits that are testable are controllability and observability.
- Controllability is the ability to set (to 1) and reset (to 0) every node internal to the circuit.
- Observability is the ability to observe, either directly or indirectly, the state of any node in the circuit.
- Good observability and controllability reduce the cost of manufacturing testing because they allow high fault coverage with relatively few test vectors.
- Three types of testing are,
  - ➢ Adhoc testing
  - ➢ Scan based approaches
  - ➢ Built -in self -test (BIST)

4. Describe the adhoc testing to design for testability in detail. (NOV 2011) [Nov/Dec 2022]

(a) Adhoc Testing

- Ad hoc test techniques are collections of ideas aimed at reducing the combinational explosion of testing.
- It is useful for small designs where scan, ATPG, and BIST are not available.
- A complete scan-based testing methodology is recommended for all digital circuits. Common techniques for ad hoc testing are
  - ➢ Partitioning large sequential circuits
  - ➢ Adding test points
  - ➢ Adding multiplexers
  - ➢ Providing for easy state reset

- A technique classified in this category is the use of the bus in a bus-oriented system for test purposes.
- Each register has been made loadable from the bus and capable of being driven onto the bus. The internal logic values that exist on a data bus are enabled onto the bus for testing purposes.
- The tester can access all the subsystems which are connected by the buses. The tester can disconnect any functional unit from the bus by setting its output into high impedance state.
- Test pattern for each subsystem can be applied separately.

- Multiplexers can be used to provide alternative signal paths during testing. In CMOS, transmission gate multiplexers provide low area and delay overhead.
- Any design should always have a method of resetting the internal state of the chip within a single cycle or at most a few cycles.
- Apart from making testing easier, this also makes simulation faster as a few cycles are required to initialize the chip.

5. Explain in detail Scan based test techniques. (NOV 2009)[Nov/Dec 2022]
Describe the scan based approaches to design for testability in detail. (NOV 2011)

(b) Scan based approaches

- The scan-design strategy for testing provides observability and controllability at each register. In this method, the registers operate in one of two modes are scan mode and normal mode.
- In normal mode, registers behave as expected.
- In scan mode, registers are connected to form a giant shift register called a scan chain spanning the whole chip.
- By applying N clock pulses in scan mode, all N bits of state in the system can be shifted out and new N bits of state can be shifted in.
- Therefore, scan mode gives easy observability and controllability of every register in the system.
- Modern scan is based on the use of scan registers, as shown in Figure.

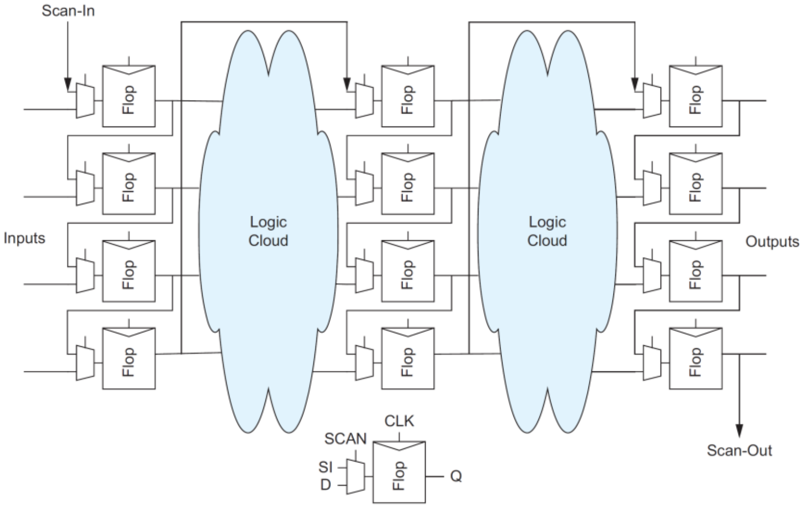


Figure: Scan based testing

- The scan register is a D flip-flop preceded by a multiplexer.
- When the SCAN signal is deasserted, the register behaves as a conventional register, storing data on the D input.
- When SCAN is asserted, the data is loaded from the SI pin, which is connected in shift register fashion to the previous register Q output in the scan chain.
- For the circuit to load the scan chain, SCAN is asserted and CLK is pulsed eight times to load the first two ranks of 4-bit registers with data.
- SCAN is deasserted and CLK is asserted for one cycle to operate the circuit normally with predefined inputs.
- SCAN is then reasserted and CLK asserted eight times to read the stored data out. At the same time, the new register contents can be shifted in for the next test.
- **The disadvantage** is the area and delay impact of the extra multiplexer in the scan register.

(i) Parallel scan approach

- The serial scan chains can become quite long, and the loading and unloading can dominate testing time.
- A simple idea is to split the chains into smaller segments.
- This can be done on a module-by-module basis or completed automatically to some specified scan length.
- This is similar to that used inside FPGAs to load and read the control RAM.
- The figure shows a two-by-two register section. Each register receives a column (column<m>) and row (row<n>) access signal along with a row data line (data<n>).

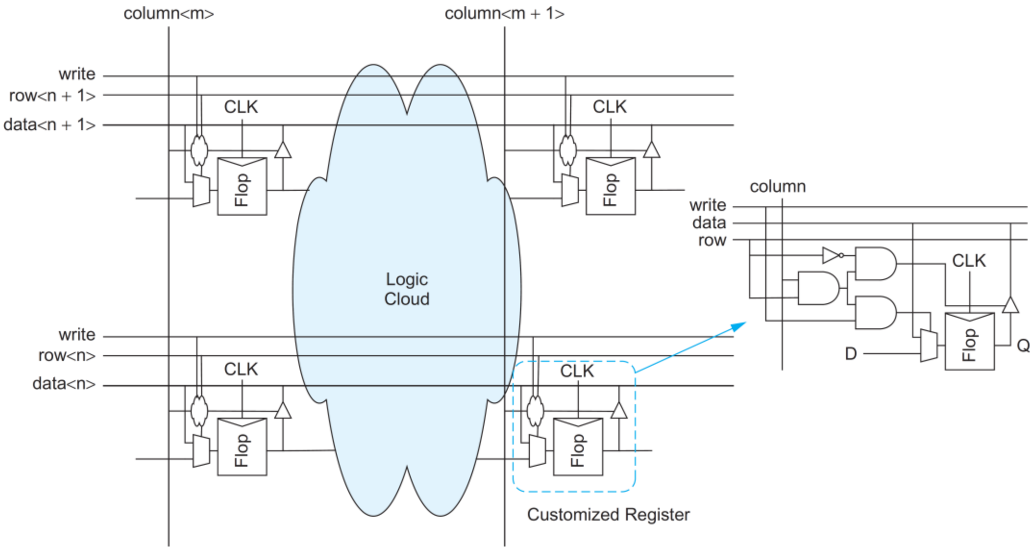


Figure: Parallel scan testing

- A global write signal (write) is connected to all registers.
- By asserting the row and column access signals in conjunction with the write signal, any register can be read or written as conventional RAM.
- Implementing the logic required at the transistor level can reduce the overhead for each register.

(ii) Scannable register design

- An ordinary flip-flop can be made scannable by adding a multiplexer on the data input, as shown in Figure (a).
- Figure (b) shows a circuit design for such a scan register using a transmission-gate multiplexer.
- The setup time increases by the delay of the extra transmission gate in series with the *D* input as compared to the ordinary static flip-flop.

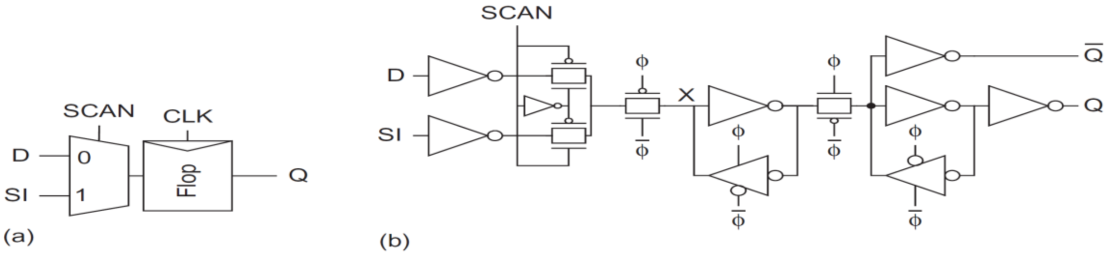


Figure: Scannable flip-flop

(iii) Level sensitive scan design (LSSD)

- During scan mode, the flip-flops are connected back-to-back. Clock skew can lead to hold time problems in the scan chain.
- These problems can be overcome by adding delay buffers on the SI input to flip-flops.
- Another approach is to use non-overlapping clocks to ensure hold times.

- The Level Sensitive Scan Design (LSSD) methodology developed at IBM uses flip-flops with two-phase non-overlapping clocks.
- During scan mode, a scan clock φs is toggled in place of φ2.
- The non-overlapping clocks also prevent hold time problems in normal operation, but increase the sequencing overhead of the flip-flop.

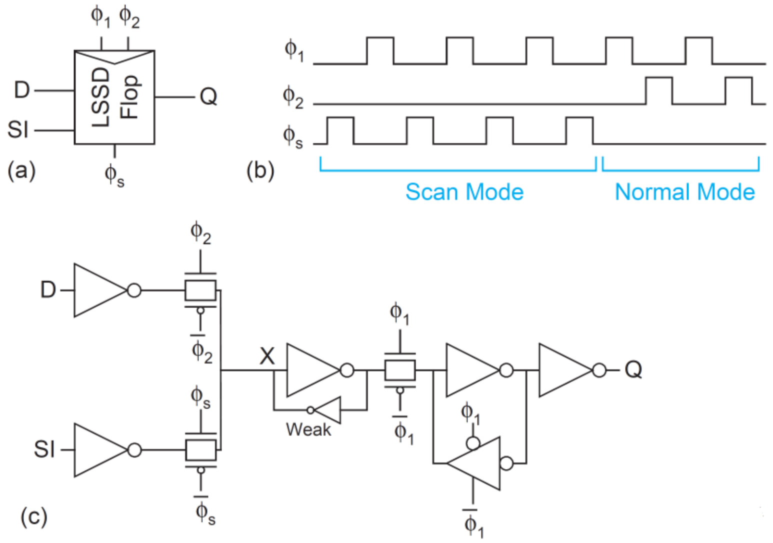


Figure: LSSD Flipflop

- Alternatively, φ1 and φ2 can be complementary clocks, but φs can be nonoverlapping to prevent races.
- Figure (c) shows a conventional design using a weak feedback inverter on the master latch that can be overpowered when either the φ2 or φs transmission gates are on.

6. Explain Built in self-test. (MAY 2008, NOV 2012, NOV 2021)

(c) Build in- Self –Test

(i) Pseudo-random sequence generator

- Self-test and built-in test techniques allow the circuit to perform operations upon themselves that prove correct operation.
- These techniques add area to the chip for the test logic, but reduce the test time required and thus can lower the overall system cost.
- One method of testing a module is to use signature analysis or cyclic redundancy checking.
- This involves using a pseudo-random sequence generator (PRSG) to produce the input signals for a section of combinational circuitry and a signature analyzer to observe the output signals.
- A PRSG of length n is constructed from a linear feedback shift register (LFSR), which in turn is made of n flip-flops connected in a serial fashion, as shown in Figure (a).
- The XOR of particular outputs are fed back to the input of the LFSR.
- An n-bit LFSR will cycle through 2^n–1 states before repeating the sequence. LFSRs are described by a characteristic polynomial indicating which bits are fed back.
- A complete feedback shift register (CFSR), shown in Figure (b), includes the zero state that may be required in some test situations.
- n-bit LFSR is converted to an n-bit CFSR by adding an n – 1 input NOR gate connected to all but the last bit.
- When in state 0…01, the next state is 0…00. When in state 0…00, the next state is 10…0. Otherwise, the sequence is the same.
- The bottom n bits of an n + 1-bit LFSR can be used to cycle through the all zeros state without the delay of the NOR gate.

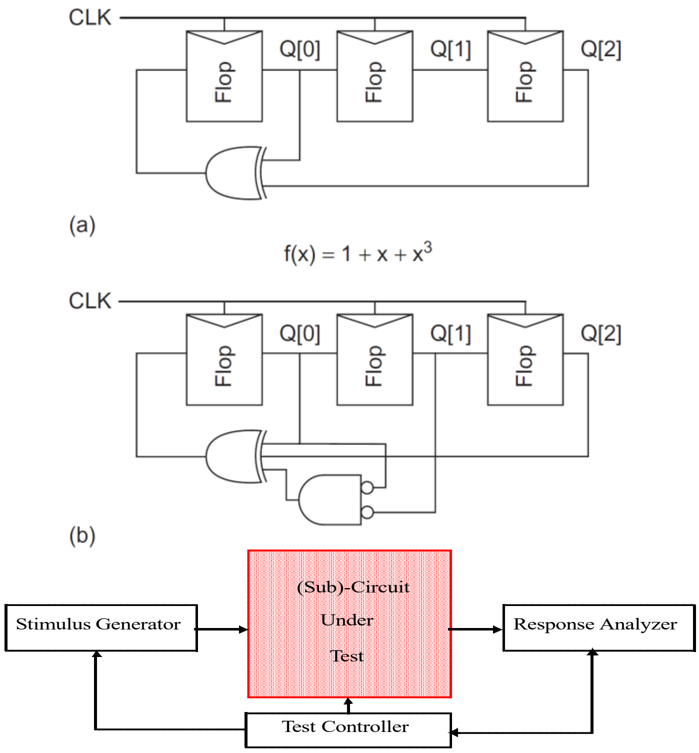


Figure: Pseudo-random sequence generator

- A signature analyzer receives successive outputs of a combinational logic block and produces a syndrome that is a function of these outputs.
- The syndrome is reset to 0, and then XORed with the output on each cycle.
- The syndrome is swizzled each cycle so that a fault in one bit is unlikely to cancel itself out.
- At the end of a test sequence, the LFSR contains the syndrome that is a function of all previous outputs.
- This can be compared with the correct syndrome to determine whether the circuit is good or bad.

(ii) Build in- Self –Test (BIST) or Built –In Logic Block Observation (BILBO)

- The combination of signature analysis and the scan technique creates a structure known as BIST—for Built-In Self-Test or BILBO—for Built-In Logic Block Observation.
- The 3-bit BIST register shown in Figure is a scannable, resettable register that also can serve as a pattern generator and signature analyzer.

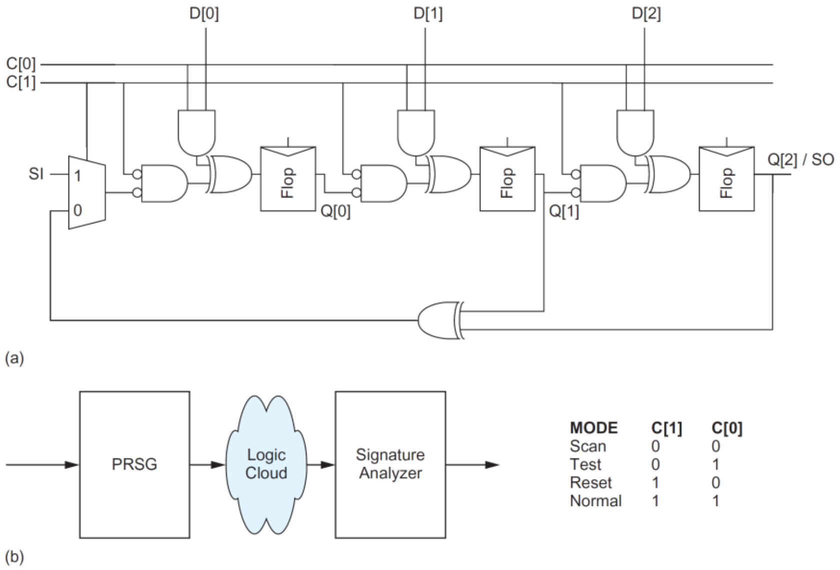


Figure: BIST (a) 3-bit register, (b) use in a system

- C[1:0] specifies the mode of operation.
- In the reset mode (10), all the flip-flops are synchronously initialized to 0.
- In normal mode (11), the flip-flops behave normally with their D input and Q output.
- In scan mode (00), the flip-flops are configured as a 3-bit shift register between SI and SO.
- In test mode (01), the register behaves as a pseudo-random sequence generator or signature analyzer.
- If all the D inputs are held low, the Q outputs loop through a pseudo-random bit sequence, which can serve as the input to the combinational logic.
- If the D inputs are taken from the combinational logic output, they are swizzled with the existing state to produce the syndrome.

7. Explain how to detect a stuck at fault with examples. (NOV 2012)

IDDQ Testing

- A method of testing for bridging faults is called IDDQ test (VDD supply current Quiescent) or supply current monitoring.
- When a CMOS logic gate is not switching, it draws no DC current (except for leakage).
- When a bridging fault occurs, then for some combination of input conditions, a measurable DC IDD will flow.
- Testing consists of applying the normal vectors, allowing the signals to settle, and then measuring IDD.
- In addition, to be effective, any circuits that draw DC power such as pseudo-nMOS gates or analog circuits have to be disabled.
- Dynamic gates can also cause problems. As current measuring is slow, the tests must be run slower than normal, which increases the test time.
- IDDQ testing can be completed externally to the chip by measuring the current drawn on the VDD line or internally using specially constructed test circuits.
- This technique gives a form of indirect massive observability at little circuit overhead.
- However, as subthreshold leakage current increases, IDDQ testing ceases to be effective because variations in subthreshold leakage exceed currents caused by the faults.

8. Explain the system level test techniques. (NOV 2007, MAY 2008, NOV 2008)
Explain in detail boundary – scan test. (MAY 2008, MAY 2013, NOV 2013, MAY 2014)

System level testing (Boundary scan testing)

- System defects occur at the board level, including open or shorted printed circuit board traces and incomplete solder joints.
- At the board level, “bed-of-nails” testers used to test boards.
- In this type of a tester, the board-under-test is lowered onto a set of test points (nails) that probe points of interest on the board.
- These can be sensed (the observable points) and driven (the controllable points) to test the complete board.
- At the chassis level, software programs are frequently used to test a complete board set.
- System designers agreeing on a unified scan-based methodology called boundary scan for testing chips at the board (and system) level.
- Boundary scan was originally developed by the Joint Test Access Group (JTAG)
- Boundary scan has become a popular standard interface for controlling BIST features as well.
- The IEEE 1149 boundary scan architecture is shown in Figure.
- All of the I/O pins of each IC on the board are connected serially in a standardized scan chain accessed through the Test Access Port (TAP)

- So that every pin can be observed and controlled remotely through the scan chain.
- At the board level, ICs obeying the standard can be connected in series to form a scan chain spanning the entire board.
- Connections between ICs are tested by scanning values into the outputs of each chip and checking that those values are received at the inputs of the chips they drive.
- Moreover, chips with internal scan chains and BIST can access those features through boundary scan to provide a unified testing framework.

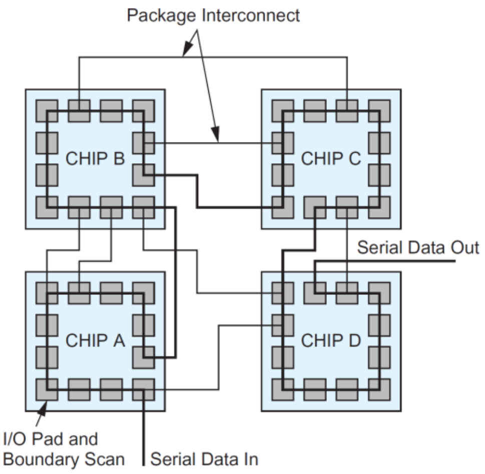


Figure: Boundary scan architecture

- The below figure shows a complete implementation of boundary scan for a chip with four inputs and four outputs.
- It consists of the TAP controller state machine and state decoder, a 3-bit instruction register with instruction decode, the bypass register, four boundary scan input pads, and four boundary scan output pads.
- The other pads comprise the test access port. The boundary scan register control signals (UpdateDR, ClockDR, ShiftDR, mode_in, and mode_out) are shown as the Control bus.

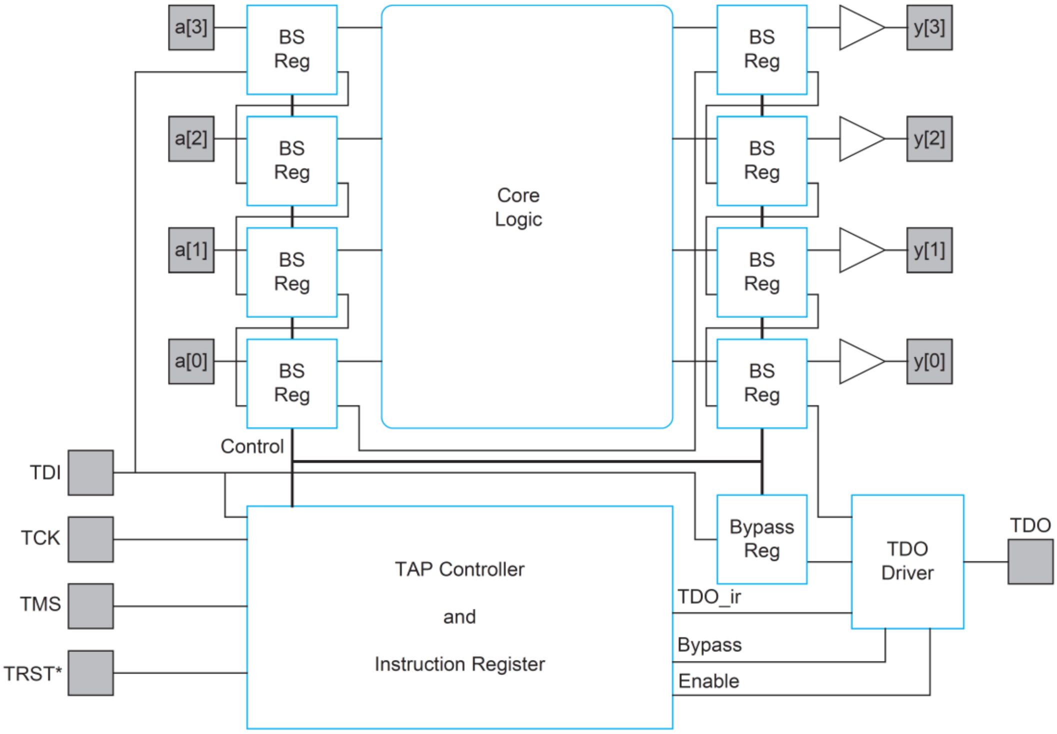


Figure: Complete Boundary scan implementation

- Boundary scan testing typically begins with the SAMPLE/PRELOAD instruction. Then, a data value is preloaded into the boundary scan registers.
- Next, the EXTEST or INTEST instruction is applied to activate the loaded value. Subsequent data values are shifted into the boundary scan registers and the results of the tests are shifted out.
- The TAP controller is initially reset. At this point, the core logic operates normally with an input pattern of 0000 and an output pattern of 0001. Then the IR is loaded with 101 (SAMPLE/PRELOAD).
- The data pattern 0111 is shifted in. The IR is loaded with 1000 (INTEST).
- This sends the 0111 pattern to the core logic, producing an output pattern of 0110.
- Finally, the data pattern 1111 is shifted in and the old output 0110 is shifted out.
- Because the INTEST is still active, the 1111 is applied to the core, producing a new output of 1100.
- It provides a uniform interface to single- and multiple-chip testing and circuit-board testing.

The Test Access Port (TAP):

- The Test Access Port has four or five single-bit connections:

➢ TCK	Test Clock	Input	Clocks tests into and out of the chip
➢ TMS	Test Mode Select	Input	Controls test operations
➢ TDI	Test Data In	Input	Test data into the chip
➢ TDO	Test Data Out	Output	Test data out of the chip; driven only when TAP controller is shifting out test data.
➢ TRST*	Test Reset Signal	Input	Optional active low signal to asynchronously reset the TAP controller if no power-up reset signal is automatically generated by the chip.

- When the chip is in normal mode, TRST* and TCK are held low and TMS is held high to disable boundary scan.
- To prevent race conditions, inputs are sampled on the rising edge of TCK and outputs toggle on the falling edge.

The Test Logic Architecture and Test Access Port:

- The basic test architecture is shown in Figure. It consists of the following:
  - ➢ The TAP interface pins
  - ➢ A set of two or more test-data registers (DR) to collect data from the chip
  - ➢ An instruction register (IR) specifying the type of test to perform
  - ➢ A TAP controller, which controls the scan of bits through the instruction and testdata registers

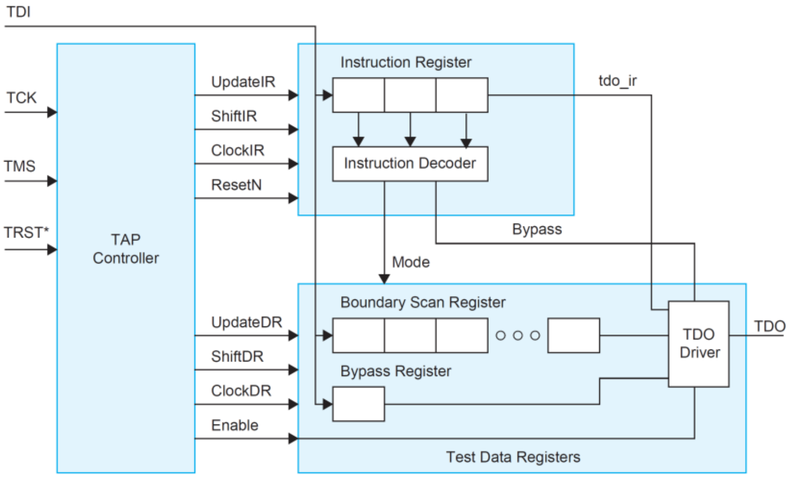


Figure: TAP Architecture

- The specification requires at least two test-data registers are the boundary scan register and the bypass register.
- The boundary scan register is associated with all the inputs and outputs on the chip so that boundary scan can observe and control the chip I/Os.
- The bypass register is a single flip-flop used to accelerate testing by avoiding shifting data into the boundary scan registers of idle chips.
- When only a single chip on the board is being tested. Internal scan chain, BIST, or configuration registers can be treated as optional additional data registers controlled by boundary scan.

The TAP Controller:

- The TAP controller is a 16-state FSM that proceeds from state to state based on the TCK and TMS signals.
- It provides signals that control the test-data registers and the instruction register. These include serial shift clocks and update clocks.
- The state transition diagram is shown in Figure. The TAP controller is initialized to Test-Logic-Reset on power-up by TRST* or an internal power-up detection circuit.
- It moves from one state to the next on the rising edge of TCK based on the value of TMS.

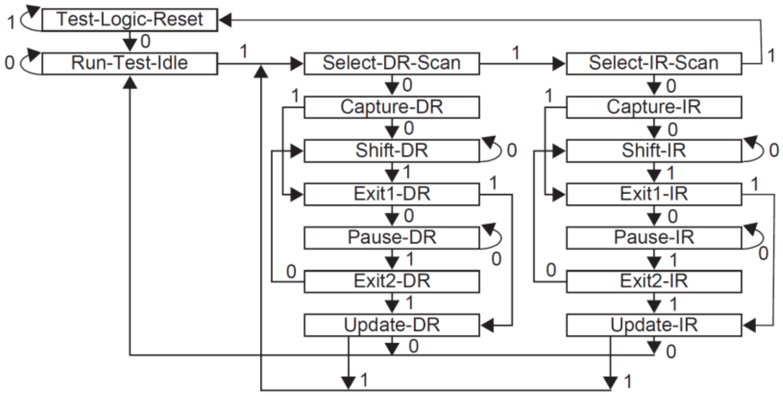


Figure: TAP controller state diagram

The Instruction Register (IR):

- The instruction register has to be at least 2 bits long. The instruction register specifies which data register will be placed in the scan chain when the DR is selected.
- It also determines from where the DR will load its value in the Capture-DR state, and whether the values will be driven to output pads or core logic.
- The following three instructions are required to be supported:
  - BYPASS—This instruction places the bypass register in the DR chain so that the path from TDI to TDO involves only a single flip-flop.
  - SAMPLE/PRELOAD—This instruction places the boundary scan registers in the DR chain. In the Capture-DR state, it copies the chip's I/O values into the DRs.
  - EXTEST—This instruction allows for the testing of off-chip circuitry.
- In addition to these instructions, the following are also recommended:
  - INTEST— This instruction allows for single-step testing of internal circuitry via the boundary scan registers. It is similar to EXTEST.
  - RUNBIST— This instruction is used to activate internal self-testing procedures within a chip.

The Data Register:

- The test data registers are used to set the inputs of modules to be tested and collect the results of running tests.
- The simplest data register configuration consists of a boundary scan register and a bypass register (1-bit long).

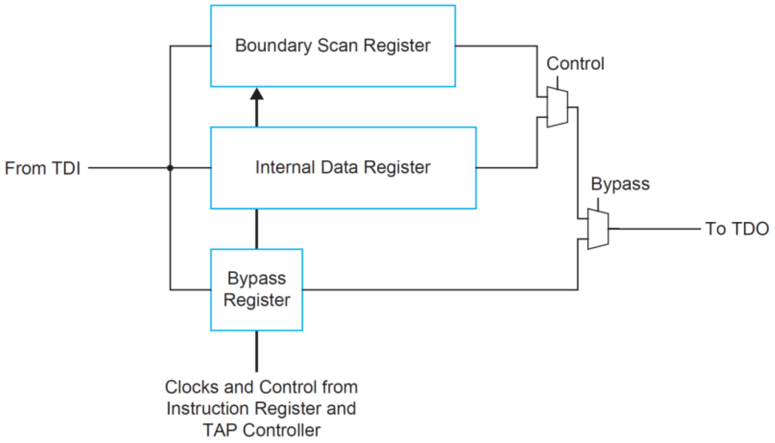


Figure: Test Data Register

The Boundary scan register:

- The boundary scan register connects to all of the I/O circuitry.
- It internally consists of a shift register for the scan chain and an additional bank of flip-flops to update the outputs in parallel.
- An extra multiplexer on the output allows the boundary scan register to override the normal path through the I/O pad so it can observe and control inputs and outputs.
- The schematic and symbol for a single bit of the boundary scan register are shown in Figure.

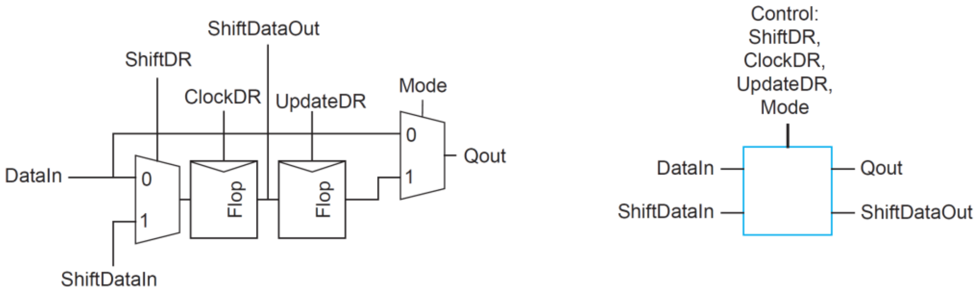


Figure: Boundary scan register bit

The Bypass Register:

- When executing the BYPASS instruction, the single-bit Bypass register is connected between TDI and TDO.
- It consists of a single flip-flop that is cleared during Capture-DR, and then scanned during Shift-DR.

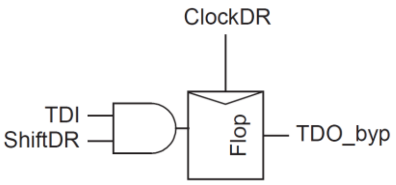


Figure: Bypass register

****

Explain the Writing test benches in Verilog HDL.

Writing test benches in Verilog HDL:

Writing test benches in Verilog HDL involves creating a module that emulates the environment in which the Design Under Test (DUT) will be tested. Test benches provide stimuli to the DUT's inputs and monitor its outputs to verify its functionality.

Here's a step-by-step guide on how to write a basic test bench in Verilog:

1. Module Declaration: Begin by declaring the test bench module, including the module name and any ports that need to be connected to the DUT.

```verilog
module tb_example;
  // Declare the DUT inputs and outputs
  // (e.g., input ports and output ports)
  // ...

  // Declare any internal signals or wires (optional)
  // ...

  // Instantiate the DUT
  // ...

  // Test bench code goes here
  // ...

endmodule
```

2. **Instantiate the Design Under Test (DUT):** In the test bench, instantiate the module representing the DUT. Connect the DUT's input and output ports to the corresponding signals or wires in the test bench.

```verilog
module tb_example;
  // Declare the DUT inputs and outputs
  // ...

  // Declare any internal signals or wires (optional)
  // ...

  // Instantiate the DUT
  DUT_module DUT_inst (
    // Connect DUT inputs to test bench signals/wires
    .input_signal_1(input_signal_1),
    .input_signal_2(input_signal_2),
    // ...

    // Connect DUT outputs to test bench signals/wires
    .output_signal_1(output_signal_1),
    .output_signal_2(output_signal_2),
    // ...
  );

  // Test bench code goes here
  // ...

endmodule
```

3. **Test Stimuli:** Within the test bench, apply test stimuli to the DUT's inputs. This can be done using initial blocks or always blocks.

```verilog
module tb_example;
  // Declare the DUT inputs and outputs
  // ...

  // Declare any internal signals or wires (optional)
  // ...

  // Instantiate the DUT
  // ...

  // Apply test stimuli
  initial begin
    // Initialize input signals with test values
    input_signal_1 = 0;
    input_signal_2 = 1;
    // ...

    // Add delays between input changes (optional)
    #5; // Delay of 5 time units

    // Change input values during simulation
    input_signal_1 = 1;
    #10; // Delay of 10 time units
    input_signal_2 = 0;

    // Add more test cases as needed
    // ...

    // End the simulation after all test cases have been executed
    $finish;
  end

  // Test bench code goes here
  // ...

endmodule
```

4. **Output Monitoring:** Use `initial` or `always` blocks to monitor and check the DUT's outputs during simulation. You can use `$display`, `$monitor`, or assertion-based methods for this purpose.

```verilog
module tb_example;
  // Declare the DUT inputs and outputs
  // ...

  // Declare any internal signals or wires (optional)
  // ...
  // Instantiate the DUT
  // ...

  // Apply test stimuli
  // ...

  // Monitor the DUT outputs
  always @(posedge clock) begin
    // Display or check DUT outputs during simulation
    $display("Output signal 1: %b, Output signal 2: %b", output_signal_1, output_signal_2);

    // Add assertions to check specific conditions (optional)
    // assert (output_signal_1 == expected_output_1) else $error("Output signal 1 mismatch");
    // assert (output_signal_2 == expected_output_2) else $error("Output signal 2 mismatch");
    // ...
  end

  // Test bench code goes here
  // ...

endmodule
```

5. **Simulating the Test Bench:** To simulate the test bench, use a Verilog simulator such as ModelSim, VCS, or Questa. The simulation will execute the test cases defined in the test bench and display the results and any assertion failures.

That's a basic outline of writing a test bench in Verilog HDL. Keep in mind that test benches can become more complex, depending on the complexity of the DUT and the desired test scenarios. Advanced test benches may include random stimulus generation, coverage analysis, and other verification methodologies to thoroughly validate the DUT's functionality.

****

Explain about Automatic Test Pattern Generation

Automatic Test Pattern Generation:

Automatic Test Pattern Generation (ATPG) is a crucial technique in semiconductor testing and verification. It involves generating a set of test patterns automatically to detect and diagnose faults or defects in a digital integrated circuit or design. The generated test patterns are applied to the design under test (DUT) during manufacturing testing or in-field testing to ensure its functionality and reliability. Here's an overview of how ATPG works:

1. **Fault Modeling:** The first step in ATPG is to create a fault model that represents the potential defects or faults in the DUT. Common fault models include stuck-at faults, transition faults, path delay faults, and bridging faults. Each fault model describes a specific type of fault that can occur in the DUT.

2. **Design Representation:** The DUT's design is represented at the gate level, typically in the form of a gate-level netlist. The netlist contains information about the gates, their connections, and the logical behavior of the design.

3. **Test Cube:** A test cube represents the inputs and outputs of the DUT that are relevant to testing a specific fault. It specifies the input patterns needed to activate the fault and the expected output responses. The ATPG tool generates test patterns based on these test cubes.

4. **Algorithmic Generation:** ATPG algorithms employ different techniques to automatically generate test patterns. Some popular algorithms include the D-algorithm, the Path Sensitization algorithm, and the Boolean Satisfiability (SAT) solver-based algorithms. These algorithms consider the fault models and the design's structure to generate effective test patterns.

5. **Test Generation Flow:**
a. The ATPG tool reads the DUT's gate-level netlist and the fault model.
b. It identifies target faults that need to be tested based on the fault model.
c. ATPG applies different algorithms to identify test cubes that activate each target fault.
d. For each test cube, the ATPG tool generates input patterns and expected output responses, creating test patterns for the target faults.
e. The generated test patterns are saved in a test vector format (e.g., STIL, WGL) or other formats suitable for test equipment.

6. **Simulation and Verification:** The generated test patterns are then applied to the DUT in a simulation environment or during manufacturing testing. The DUT's responses are compared with the expected outputs to detect any faults. If the DUT fails the test, the specific fault(s) that caused the failure can be identified for further diagnosis and debugging.

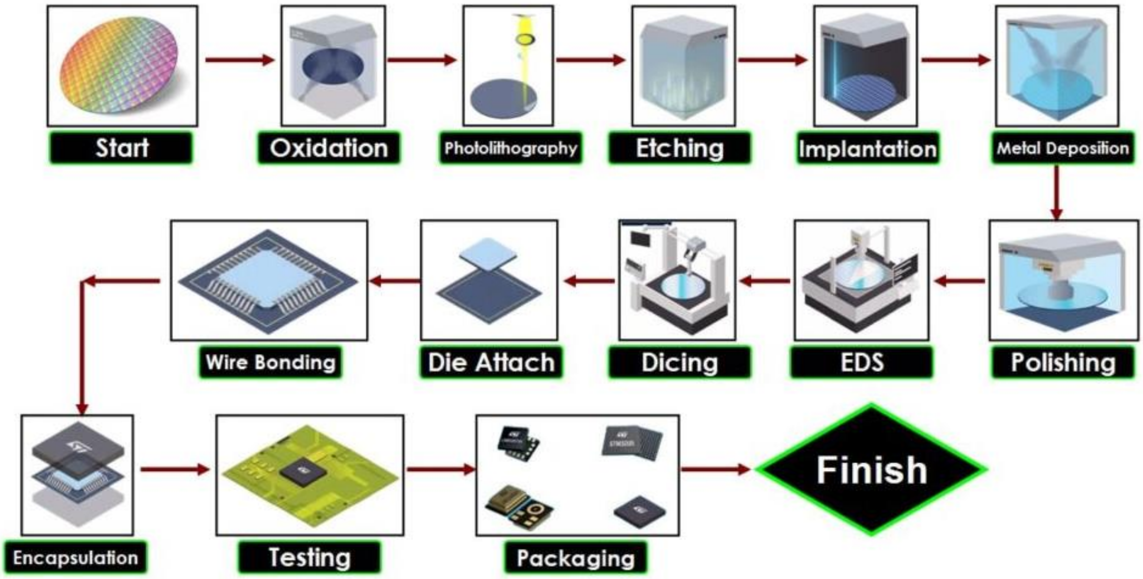
ATPG is a powerful technique that significantly improves the efficiency and coverage of semiconductor testing. It helps ensure the quality and reliability of integrated circuits, enabling the detection of manufacturing defects and design errors, thereby enhancing the overall product yield and performance.

****

Explain the Process involved in wafer to chip fabrication.
Introduction to wafer to chip fabrication process flow

- Wafer to chip fabrication, also known as semiconductor manufacturing, is the process of transforming a silicon wafer into individual semiconductor chips or integrated circuits (ICs).
- The process involves a series of steps that are meticulously executed in a cleanroom environment to ensure the highest possible quality and yield.

Here's an overview of the typical wafer to chip fabrication process flow:



1. **Wafer Ingot Growth:** The process begins with the growth of a silicon ingot. The silicon ingot is sliced into thin, circular wafers using a diamond-tipped saw. These wafers serve as the base material for manufacturing chips.

2. **Wafer Cleaning**: The wafers undergo rigorous cleaning processes to remove any contaminants or particles that might have accumulated during handling or previous steps. Cleanliness is crucial to ensure defect-free manufacturing.

3. **Oxidation:** The wafers are exposed to high-temperature oxygen or steam to create a thin layer of silicon dioxide (SiO2) on their surface. This layer serves as an insulating material and also provides a base for subsequent processes.

4. **Photolithography:** In this step, a photoresist material is applied to the wafer's surface. Light is then shone through a photomask that contains the pattern of the desired circuit. The photoresist is exposed to this patterned light, creating a mask on the wafer. This process defines the circuit pattern for the subsequent steps.

5. **Etching:** The exposed parts of the wafer's surface are either removed or modified using chemical or physical etching processes. This step transfers the pattern from the photomask onto the wafer, defining the circuit layout.

6. **Doping:** Dopants (impurity atoms) are selectively introduced into specific areas of the wafer to modify its electrical properties. This process creates regions with either excess or deficient electrons, forming the various components of transistors (source, drain, gate, etc.).

7. **Thin Film Deposition:** Thin films of various materials, such as metal, polysilicon, or insulators, are deposited onto the wafer surface using techniques like chemical vapor deposition (CVD) or physical vapor deposition (PVD). These films serve as conductors or insulators in the circuit.

8. **Chemical Mechanical Polishing (CMP):** CMP is used to planarize the wafer's surface, making it smooth and even. This is essential for accurate layering and subsequent processing steps.

9. **Annealing:** The wafer is heated in a controlled environment to activate dopants, repair crystal damage, and improve the electrical properties of the fabricated components.

10. **Chemical Mechanical Polishing (CMP):** CMP is used to planarize the wafer's surface, making it smooth and even. This is essential for accurate layering and subsequent processing steps.

11. **Annealing:** The wafer is heated in a controlled environment to activate dopants, repair crystal damage, and improve the electrical properties of the fabricated components.

12. **Testing:** Throughout the process, various tests are conducted to ensure the quality of the chips being manufactured. These tests help identify defects and ensure that the chips meet the required specifications.

13**. Packaging:** Once all the chips on the wafer are deemed functional, they are separated and assembled into their respective packages. The packages provide protection and electrical connections to the chips, enabling them to be mounted on printed circuit boards (PCBs).

14. **Final Testing:** After packaging, the chips undergo final testing to verify their functionality and performance. Defective chips are discarded, and only fully functional chips are sent for distribution and integration into electronic devices.

- It's important to note that the above process is a simplified overview, and the actual fabrication process can be much more complex, involving multiple iterations of the steps to create multiple layers and intricate circuitry on a single chip.
- Semiconductor manufacturing is a continuously evolving field, with advancements in technology and miniaturization constantly pushing the boundaries of what is possible.

TWO MARK QUESTIONS & ANSWERS

UNIT V - ASIC DESIGN AND TESTING

1. **List out the Implementation technologies in ASIC.**

   The implementation technologies used in ASIC are:
   TTL – Transistor Transistor Logic
   ECL – Emitter Coupled Logic
   MOS – Metal Oxide Semiconductor (NMOS, CMOS)

2. **What are the types of ASICs?**

   The ASICs are classified as follows:

   I. Full-Custom ASICs
   II. Semi-custom ASICs
      a. Standard-Cell–Based ASICs (CBIC)
      b. Gate-Array–Based ASICs (MPGA)
         - Channeled Gate Array
         - Channel less Gate Array
         - Structured Gate Array
   III. Programmable ASICs
      a. Complex Programmable Logic Devices (CPLD)
      b. Field-Programmable Gate Arrays (FPGA)

3. **What is meant by Full-Custom design? (May 2009)**

   - All mask layers are customized in a full-custom ASIC
   - Generally, the designer lays out all cells by hand
   - Some automatic placement and routing may be done
   - Critical (timing) paths are usually laid out completely by hand.

4. **What are the features of Full-Custom ASICs? (May 2016)**
   List the advantages of Full-Custom ASICs.

   - Full-custom design offers the highest performance, minimizes its area and lowest part cost (smallest die size) for a given design.

5. **What are the disadvantages of Full-Custom ASICs?**

   The disadvantages of full-custom design include increased design time, complexity, design expense, and highest risk.

6. **Give examples of Full-Custom ASICs.**

   - Microprocessors (strategic silicon) were exclusively full-custom, but designers are increasingly turning to semicustom ASIC techniques in this area as well.
   - Other examples of full-custom ICs or ASICs are requirements for high-voltage (automobile), analog/digital (communications), sensors and actuators, and memory (DRAM)

7. **What are Semi-custom ASICs?**

- Semi-custom ASIC is a cell-based ASIC (CBIC — "sea-bick")
- It has Standard cells.
- Possibly megacells, megafunctions, full-custom blocks, system-level macros (SLMs), fixed blocks, cores, or Functional Standard Blocks (FSBs)
- All mask layers are customized - transistors and interconnect.

8. **What is meant by standard cell?**

A standard cell is a group of transistors and interconnect structure that provides a Boolean logic function (e.g., AND, OR, XOR, Inverter) or a function (flip-flop or latch).

9. **What is the standard cell-based ASIC design? (Nov 2016)**

✓ In semiconductor design, standard cell methodology is a method of designing application-specific integrated circuits with digital logic features.

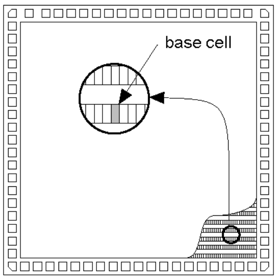
✓ A standard cell is a group of transistors and interconnect structure that provides a Boolean logic function (e.g., AND, OR, XOR, Inverter) or a function (flip-flop or latch).

10. **What is meant by CBIC? (Nov 2009, April 2017)**

Cell-Based Integrated Circuit consists of standard cells. CBIC based circuit is fixed and cannot be reconfigured.

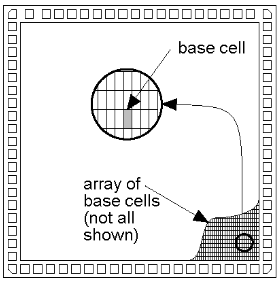
11. **What is Channeled Gate Array?**

- Only the interconnect is customized.
- The interconnect uses predefined spaces between rows of base cells.
- The manufacturing lead time is between two days and two weeks



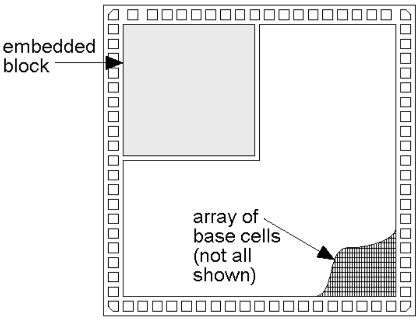
12. **What is Channel less Gate Array?**

- There are no predefined areas set aside for routing - routing is over the top of the gate-array devices
- Achievable logic density is higher than for channeled gate arrays
- Manufacturing lead time is between two days and two weeks



13. What is Structured Gate Array?

- Only the interconnect is customized
- Custom blocks (the same for each design) can be embedded
- These can be complete blocks such as a processor or memory array
- An array of different base cells better suited to implementing a specific function
- Manufacturing lead time is between two days and two weeks.



14. What is Ratio factor?

The *ratio factor* is the relative balance between the two partitions with respect to cell area. It is used to prevent all cells from clustering into one partition.

The ratio factor *r* is defined as

$$r = \frac{area(A)}{area(A) + area(B)}$$

where *area*(*A*) and *area*(*B*) are the total respective areas of partitions *A* and *B*

15. Write the Goals and Objectives of floor planning?

The goals of floor planning are to:

✓ arrange the blocks on a chip,
✓ decide the location of the I/O pads,
✓ decide the location and number of the power pads,
✓ decide the type of power distribution, and
✓ decide the location and type of clock distribution.

- The objectives of floor planning are to minimize the chip area and minimize delay.
- Measuring area is straightforward, but measuring delay is more difficult.

16. What is Channel Definition?

During the floorplanning step, we assign the areas between blocks that are to be used for interconnect. This process is known as channel definition or channel allocation.

17. What is placement?

- After completing a floorplan, we can begin placement of the logic cells within the flexible blocks.
- Placement is much more suited to automation than floor planning.
- Thus we shall need measurement techniques and algorithms.

18. What are the different placement algorithms in ASIC.

There are two classes of placement algorithms commonly used in commercial CAD tools: constructive placement and iterative placement improvement.

19. What is synthesis?

The initial synthesis contains little or no information on any interconnect loading. The output of the synthesis tool is the input to the floor planner.

20. What is Timing-driven placement?

After placement using constraints from the synthesis tool, the location of every logic cell on the chip is fixed and accurate estimates of interconnect delay can be passed back to the synthesis tool.

21. What is Routing?

Once the designer has floor planned a chip and the logic cells within the flexible blocks have been placed, it is time to make the connections by routing the chip.

22. What are the two different types of routing? (April 2018)

Two types of routing are Global routing and Hierarchical routing.

23. What is Global Routing?

A global router does not make any connections, it just plans them. We typically global route the whole chip before detail routing the whole chip (or the pieces).

24. What are the methods of Global Routing?

- Sequential routing
- Order-independent routing

25. What is hierarchical routing?

Hierarchical routing handles all nets at a particular level at once. Rather than handling all of the nets on the chip at the same time.

26. What is Reserved-layer routing?

Reserved-layer routing restricts all interconnects on each layer to flow in one direction in each routing area.

27. What is Special Routing?

The routing of nets that require special attention, clock, and power nets for example, is normally done before detailed routing of signal nets.

28. What is meant by standard cell library? (NOV 2016)

What is the role of cell libraries in ASIC design? (April 2018)

- The cell library is the key part of ASIC design.
- For a programmable ASIC the FPGA Company supplies a library of logic cells in the form of a design kit.

29. What are the models should have in a cell library?

Each cell in an ASIC cell library must contain the following:

- A physical layout
- A behavioral model
- A Verilog/VHDL model
- A detailed timing model
- A test strategy
- A circuit schematic
- A cell icon
- A wire-load model
- A routing model

30. What is meant by library cell design?

- Layout of library cells is either hand-crafted or uses some form of symbolic layout.
- Symbolic layout is usually performed in one of two ways: using either interactive graphics or a text layout language.
- Shapes are represented by simple lines or rectangles, known as sticks or logs , in symbolic layout.

31. List out the basic elements of the FPGA structure.
State the three important blocks in FPGA architecture. (April 2019)
State the building blocks of FPGA. (Nov 2019)

The basic elements of FPGA structure are:

1.Logic blocks

- Based on memories *(LUT – Lookup Table)* Xilinx
- Based on multiplexers *(Multiplexers)* Actel
- Based on PAL/PLA *(PAL - Programmable Array Logic, PLA – Programmable Logic Array)* Altera
- Transistor Pairs

2. Interconnection Resources

- Symmetrical FPGA-s
- Row-based FPGA-s
- *Sea-of-gates* type of FPGA-s
- Hierarchical FPGA-s (*CPLD)*

3. Input-output cells (*I/O Cell)*

32. Name the elements in a Configuration Logic Block. (April 2017)

Configuration Logic blocks:

- Memories *(LUT – Lookup Table, Flip-flop)* - Xilinx
- Multiplexers -Actel
- PAL/PLA *(PAL - Programmable Array Logic, PLA – Programmable Logic Array)* - Altera
- Transistor Pairs

33. Write the features of Xilinx FPGA. (April 2008)

The features of Xilinx FPGA are:

✓ High-performance

-5 ns pin-to-pin logic delays on all pins

- ✓ Large density range - 36 to 288 macrocells with 800 to 6,400 usable gates
- ✓ 5 V in-system programmable
- ✓ Endurance of 10,000 program/erase cycles

34. Give the functions of Input Output Block.

- ✓ The I/O Block (IOB) interfaces between the internal logic and the device user I/O pins.
- ✓ Each IOB includes an input buffer, output driver, output enable selection multiplexer, and user programmable ground control.

35. Write about various ways of routing procedure. (Nov 2017)

- ✓ Hierarchical Routing Architecture
- ✓ Island-Style Routing Architecture
- ✓ Xilinx Routing Architecture
- ✓ Altera Routing Architecture
- ✓ Actel Routing Architecture

36. What is VLSI and ULSI? (Nov 2017)

Very large-scale Integration:

Very large-scale Integration (VLSI) with gates counting upto lakhs.

Ultra large-scale Integration:

Ultra large-scale integration (ULSI) is the process of integrating millions of transistors on a single silicon semiconductor microchip.

37. What is feed through cells? State their uses. (May 2016)

A Feed through is a connection that needs to cross over a row of standard cells.

Feed through cells needed for vertical routing for routing using the same metal layer(s) as within cells.

38. What is a programmable logic device?

A programmable logic device (PLD) is an electronic component used to build reconfigurable digital circuits.

Unlike a logic gate, which has a fixed function, a PLD has an undefined function at the time of manufacture.

39. What are the types of programmable logic device (Programmable ASIC)?

- PLA
- PAL
- FPGA

40. What is meant by ASIC?

Application Specific Integrated Circuit is an Integrated Circuit (IC) designed to perform a specific function for a specific application.

41. What is an antifuse? State its merits and demerits. (Nov 2016)

- ✓ Antifuse is nothing high resistance (>100 MΩ) is changed into low resistance(200-500Ω) by applying programming voltage.

- ✓ Merit: Antifuses separate interconnect wires on the FPGA chip and the programmer blows an antifuse to make a permanent connection.
- ✓ Demerit: Once an antifuse is programmed, the process can’t be reversed.

42. What is PLA?

Programmable logic arrays (PLAs) is a type of fixed architecture logic devices with *programmable AND gates followed by programmable OR array*.

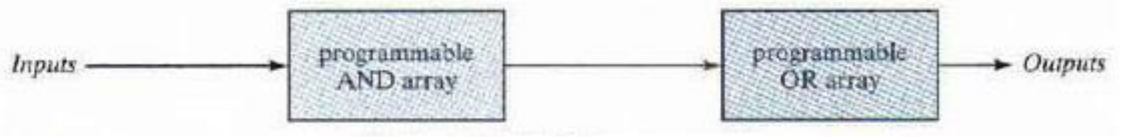


Figure: Programmable logic array

43. What is PAL?

The PAL is a programmable logic device with a *fixed OR array and a programmable AND* array.

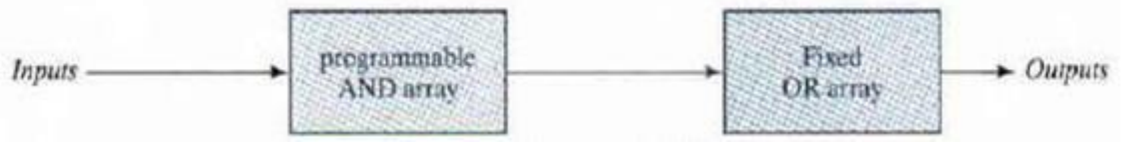


Figure: Programmable Array Logic

44. What are the types FPGA programming technologies?

What are the different types of programming structure available in PAL? (Nov 2008)

There are three types of programming technology.

- Fusible link programming (Anti fuse)
- SRAM Programming
- EPROM and EEPROM programming

45. What is meant by Reprogrammable Gate array (FPGA)?

What is the significance of field programmable gate arrays? [May 2021][Apr/May 2022]

A field programmable gate array (FPGA) is a VLSI circuit that can be programmed at the user’s location.

A typical FPGA consists of an array of millions of logic blocks, surrounded by programmable input, and output blocks and connected together via programmable interconnections.

46. Differentiate between Full custom and Cell based ASICs. (April 2008)

Full Custom ASICs:

- All mask layers are customized in a full-custom ASIC
- Generally, the designer lays out all cells by hand
- Some automatic placement and routing may be done

Cell based ASICs:

- It has Standard cells, Possibly mega cells, mega functions, full-custom blocks, system-level macros (SLMs), fixed blocks, cores or Functional Standard Blocks
- All mask layers are customized - transistors and interconnect.

47. What is needed for testing? [Nov/Dec-2013] [Apr/May-2008]

Physical defects are likely in manufacturing.

– Missing connections (opens)

– Bridged connections (shorts)

– Imperfect doping, processing steps

– Packaging

Need to weed out bad die before assembly. Need to test during operation – Electromagnetic interference, mechanical stress, electromigration, alpha particles.

48. What are different stages of testing on a chip? [Nov/Dec-2012]

Different stages of testing are wafer level, circuit level, chip level, board level, field level, logic level.

49. What is meant by tester in VLSI testing?

A tester is a device that can apply a sequence of stimuli to a chip or system under test and record the results.

50. Distinguish testers and test fixtures. [Nov/Dec-2012]

Testers: equipment used for testing.

Test fixtures: components through which the equipment's are connected with testing block.

51. State all the test vectors to test 3 input NAND gate. [May/June-2009]

Three inputs test vector are 000, 001, 010, 011, 100, 101, 110 and 111.

52. What are the test fixtures required to test a chips? [Nov/Dec-2011]

To test a chip, various types of test fixtures may be required. These are

- Probe card: It is used to test at the wafer level or unpackaged die level with a chip tester.
- Load board: It is used to test a packaged part with a chip tester.
- Printed circuit board (PCB): It is used for bench-level testing (with or without a tester).
- PCB with the chip in situ: It is used for demonstrating the system application for which the chip is used.

53. What is meant by test program?

The tester requires a test program. This program is written in a high-level language that supports a library of primitives for a particular tester.

54. What is handler?

Used for feeding Ics to a test fixture which is attached to a tester? It has mechanical positioning equipment. Status is indicated at the top, to check if the handler is functioning or not. It can handle 2 to 4 chips at a time.

55. How is testing classified?

List out the basic types of CMOS testing (or) What are the different types of CMOS testing? [May/June-2013] [Nov/DEC-2008]

The basic types of CMOS testing are functionality testing and Manufacturing testing.

56. What is functionality test?

State the objective of functionality test. [Nov/Dec-2011]

Functionality test is to check whether logic block works with correct logic. It leads to imperfection of logic function. It is done before fabrication process.

57. What is manufacturability test?

Manufacturing test is to check is there any defects occurred in circuit after fabrication process. It leads to nodes to float, shorted to power or ground. It is done after fabrication process.

58. Distinguish functionality test and manufacturing test. [Nov/Dec-2007]

Functionality: Functionality test is to check whether logic block works with correct logic. It leads to imperfection of logic function. It is done before the fabrication process.

Manufacturing: Manufacturing test is to check is there any defects occurred in circuit after fabrication process. It leads to nodes to float, shorted to power or ground. It is done after fabrication process.

59. What is the principle behind logic verification? [Nov/Dec-2013]

Verifying the logical principles of the circuit by the following ways:
Test benches & Harness, Regression testing, Version control and Bug tracking**.**

60. Define test benches and harness.

A verification test bench or harness is a piece of HDL code.

In the simplest test bench, input vectors are applied to the module under test and at each cycle, the outputs are taken and compared with the output of another model.

61. What is shmooing?

The ability of varying the voltage and timing on a per-pin basis with a tester allows a process known as shmooing.

62. What is shmooing plots?

In this plot, voltage is taken in x-axis and speed is taken in y-axis. The test vectors are applied and the output is recorded.

63. What is VCD?

Vector Change Description - It is used to compact stimulation results.

64. What is version control?

It is used in regression testing. It is orderly management of different design iterations.

65. What is regression testing?

Stimulation performed automatically. Verify that no functionality has changed in a model. Regression conducted after design activities every day.

66. What is meant by Bug tracking?

Bug tracking is nothing but allows the management of wide variety of Bugs(error checks).

67. What is meant by silicon debugging principles and name some probes used for it?

Silicon debugging principles are those technique which can directly access the silicon for testing.
LVP-Laser Voltage Probing, PICA-Picasecond Imaging Circuit Analysis.

68. What is hotspot and how it is examined?

Hotspots are examined using infrared imaging techniques.

69. How the temperature is examined in a chip?

Liquid Crystal material can be painted to a die to identify the temperature related problems.

70. What is FIB?

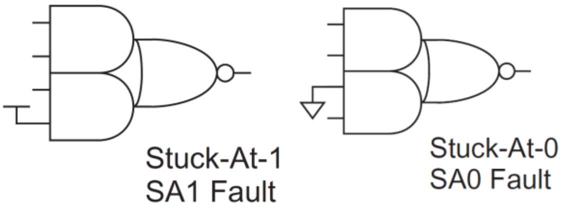
Focused Ion Beam (FIB): If the fault locations are identified, then FIB can be used to cut wires (or) lay new conductors down.

71. What is electrical failure?

Electrical failur occurs if the chip is logically correct but malfunctions occur due to temperature.

72. What is struck at fault? [Nov/DEC-2008]

If any input line is struck at logic ‘0’ or logic ‘1’ permanently called as struck at fault. If it struck at logic ‘1’ then called as struck-at-1 or s-a-1 and if it struck at logic ‘0’ then called as struck-at-0 or s-a-0.



73. What is struck open fault? [Nov/DEC-2008]

Due to defects while manufacturing leads to permanent disconnect of drain to source terminal called as struck open fault.

74. What is bridging fault or short circuit fault?

Due to defects while manufacturing leads to shorting of inputs between themselves or shorting of inputs to outputs(feedback) occurs called as bridging fault.

75. List any two faults that occur during manufacturing. [Nov/DEC-2008]

Struck at faults and struck open fault.

76. Define fault coverage?

Fault coverage is defined as ratio of the number of nodes detected as faults and total number of nodes in the circuit.

$$\text{Fault coverage} = \frac{\text{Total no. of nodes in which fault identified}}{\text{Total no. of nodes in circuit}}$$

77. How is the bridging fault categorized?

Bridging faults are categorized as

Input bridging

Feedback bridging

Non-feedback bridging

78. What is observability? [Nov/Dec 2022]

The observability of a particular circuit node is the degree to which we can observe that node at the outputs of an integrated circuit.

79. What is controllability? [Nov/Dec 2022]

The measure of ease of forcing/setting a node to 0 or 1 by driving input pins of the chip is called controllability.

80. What is meant by ATPG?

Automatic Test Pattern Generation (ATPG) – Block generates input patterns automatically by itself for testing its own logic block and stores the output pattern and compare it with defined pattern for error identification.

81. What are the 3 approaches in design for testability? (or) List out design required for testing in CMOS chip design. [Apr/May-2008]

Three approaches in design for testability are

Adhoc testing
Scan based testing
BIST- Built In Self Test

82. What do you mean by DFT? [Nov/Dec-2009]

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

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

Adhoc testing
Scan based testing
BIST- Built In Self Test

83. What is serial scan & parallel scan?

In serial scan based approaches, logic is connected to form a giant shift register called as a scan chain spanning the whole chip.

In parallel scan based approaches, logic is split the chain into smaller segments. This can be done on a module –by-module basis or completely automatically to some specified scan length.

84. List the common techniques for ad hoc testing. (NOV 2021)

1. Buddy testing 2. Pair testing 3. Monkey testing

85. What is signature analyzer?

Signature analyzer is a block which observes the output signal.

86. What is the drawback of scan based approaches?

Drawbacks of scan based approaches are area and delay impact of the extra multiplexer in the scan register.

87. What is the aim of Adhoc test techniques?[Nov/Dec-2007] [Apr/may-2010]

Ad hoc test techniques are collections of ideas aimed at reducing the combinational explosion of testing. They are only useful for small designs where scan, ATPG, and BIST are not available. A complete scan-based testing methodology is recommended for all digital circuits.

88. What is the MUX test technique?

Multiplexers can be used to provide alternative signal paths during testing. In CMOS, transmission gate multiplexers provide low area and delay overhead.

89. Write a note on partition and MUX technique

What are common techniques used in adhoc testing? [Apr/May-2011]

(i) Partitioning large sequential circuits

(ii) Adding test points

(iii) Adding multiplexers (iv) Storing output datas

90. What are scannable elements for circuit design?

Mutliplexer, Flipflop and CMOS transmission gates are scannable elements in circuit design.

91. What is the necessary for non-overlapping clocks?

The non-overlapping clocks in LSSD prevents hold time problems in normal operation, but increase the sequencing overhead of the flipflop.

92. What is syndrome?

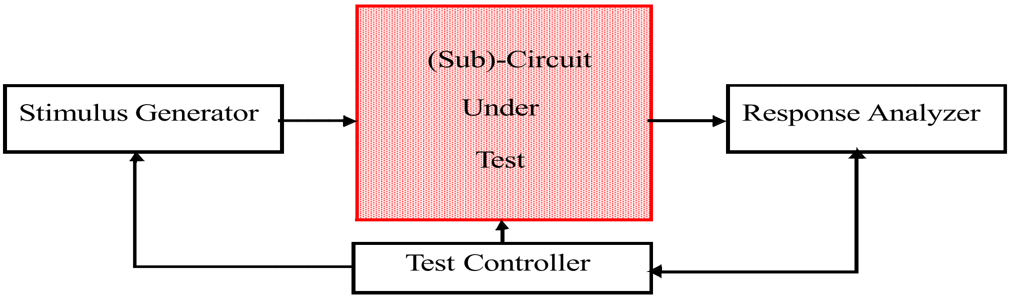
Syndrome is output pattern generated for the applied input pattern in testing.

93. What is LSSD?

Level Sensitive Scan Design : In this method, flipflops with two phase non-overlapping clocks are used in testing circuit.

94. What is BIST or BILBO?

The Combination of scan technique with PRSG & signature analysis creates a structure called as Built-in Logic Block Observer (BILBO).



95. What is needed for IDDQ testing? [Apr/May-2011] [Apr/May-2010]

List out design guidelines for IDDQ testing.

A method of testing for bridging faults is called IDDQ test (*VDD* supply current Quiescent) or supply current monitoring. This relies on the fact that when a CMOS logic gate is not switching, it draws no DC current (except for leakage).

When a bridging fault occurs, then for some combination of input conditions, a measurable DC *IDD* will flow. Testing consists of applying the normal vectors, allowing the signals to settle, and then measuring *IDD*.

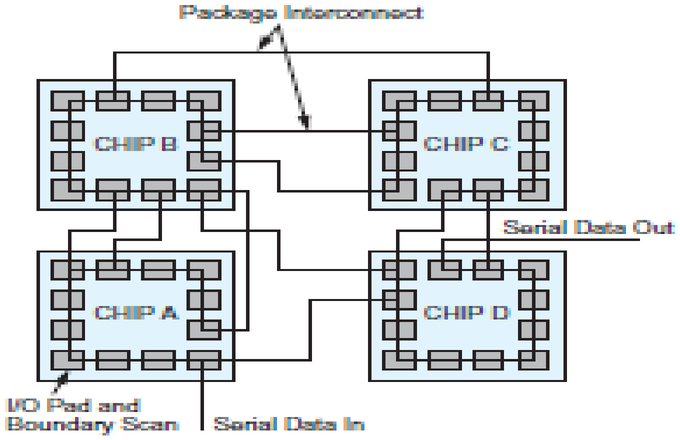
96. What are the limitations of IDDQ testing?(NOV 2021)[Apr/May 2022]

Compared to scan chain testing, IDDQ testing is time consuming and more expensive.

97. What is meant by system level (Boundary scan) testing?

System designers agree on a unified scan-based methodology called boundary scan for testing chips at the board (and system) level.

98. Draw the boundary scan input logic diagram. [Nov/Dec-2009]



99. What is TAP?

All the input and output pins of each IC on the board are connected serially in a standardized scan chain accessed through the TAP (Test Access Port), so that every pins can be observed and controlled remotely through the scan chain.

100. What are the signals used in Tap Access port (TAP)?

The Test Access Port has four or five single-bit connections:

TCK	Test Clock	Input	Clocks tests into and out of the chip
TMS	Test Mode Select	Input	Controls test operations
TDI	Test Data In	Input	Test data into the chip
TDO	Test Data Out	Output	Test data out of the chip; driven only when TAP controller is shifting out test data.
TRST*	Test Reset Signal	Input	Optional active low signal to asynchronously reset the TAP controller if no power-up reset signal is automatically generated by the chip.

101. Draw the Tap Access port (TAP) architecture.

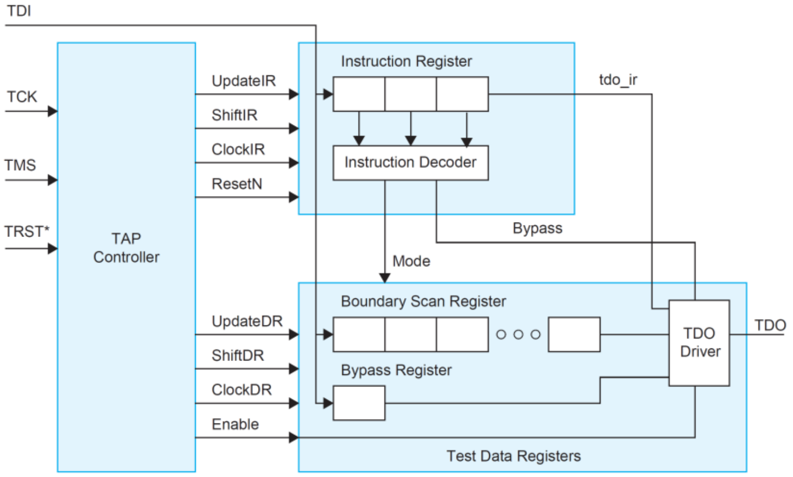


Figure: TAP Architecture

102. What is meant by TAP controller?

The TAP controller is a 16-state FSM that proceeds from state to state based on the TCK and TMS signals.

103. What is bypass register?

Single bit register connected between Test Data In (TDI) and Tata Data Out (TDO). It is cleared during capture- DR, and then scanned during shift Data Read (DR).

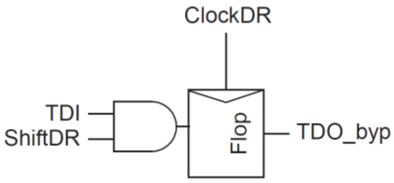


Figure: Bypass register

104. What is instruction register?

Instruction register is a 2 bit register which specifies which data register will be placed in the scan chain when DR – data register is selected.

105. What is Data Register (DR)?

The test data registers are used to set the inputs of modules to be tested and collect the results of running tests.

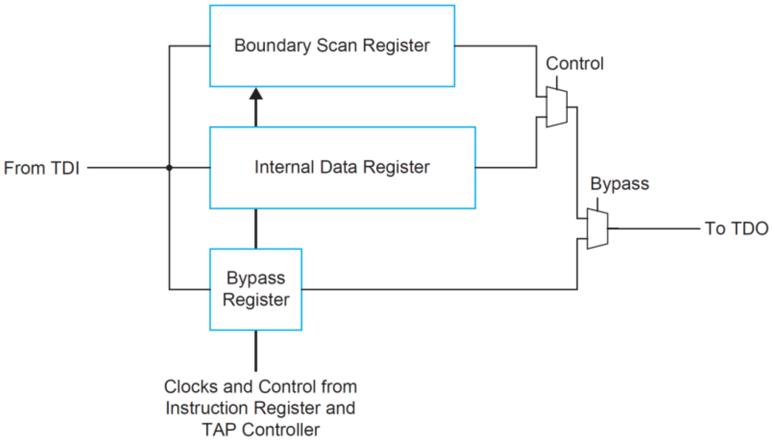


Figure: Test Data Register

106. What is boundary scan register?

The boundary scan register connects to all of the I/O circuitry. It internally consists of a shift register for the scan chain and an additional bank of flip-flops to update the outputs in parallel.

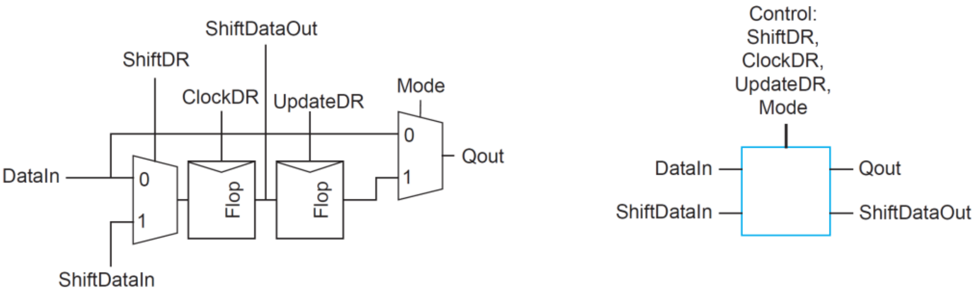


Figure: Boundary scan register bit

107. What are logic verification principles? [May 2013, Nov 2013]

Verifying the logical principles of the circuit by the following ways test benches & Harness, regression testing, version control and bug tracking.

106. Compare full custom and semi-custom design. (Nov 2019)

Full-Custom design

- All mask layers are customized in a full-custom ASIC
- Generally, the designer lays out all cells by hand
- Some automatic placement and routing may be done
- Critical (timing) paths are usually laid out completely by hand.

Semi-custom ASICs

- Semi-custom ASIC is a cell-based ASIC (CBIC —"sea-bick")
- It has Standard cells
- Possibly megacells , megafunctions , full-custom blocks , system-level macros (SLMs), fixed blocks, cores, or Functional Standard Blocks (FSBs)
- All mask layers are customized - transistors and interconnect

107. Identify the ways to optimize the manufacturability, to increase yield. [May 2021]

Examine workflow

Invest in employee training.

Modernize your business process.

Invest in smart machining equipment.

Develop realistic expectations.

Stay organized.

Create a culture of collaboration.

Invest in preventative maintenance.

108. What are the advantages and disadvantages of BIST? [Nov/Dec 2022]

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
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 circuit
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
4) Possible issues with the correctness of BIST results, since the on-chip testing hardware itself can fail.

109. What is configurable logic block meant?

The programmable logic blocks of FPGAs are called Configurable Logic Blocks (CLBs).

CLBs contain LUT, FF, logic gates and Multiplexer to perform logic functions.

UNIT V
ASIC DESIGN AND TESTING
Question bank

1. **Explain about wafer to chip fabrication process flow.**
2. Describe about Microchip design process.
3. Explain the issues in test and verification of complex chips.
4. Write short notes on embedded cores and SOCs.
5. **Explain the Fault models and Test coding.**
6. **Explain the ASIC Design Flow.**
7. **Explain the writing test benches in Verilog HDL**
8. **Explain the Automatic test pattern generation.**
9. **Explain the Design for testability.** (Scan design: Test interface and boundary scan)