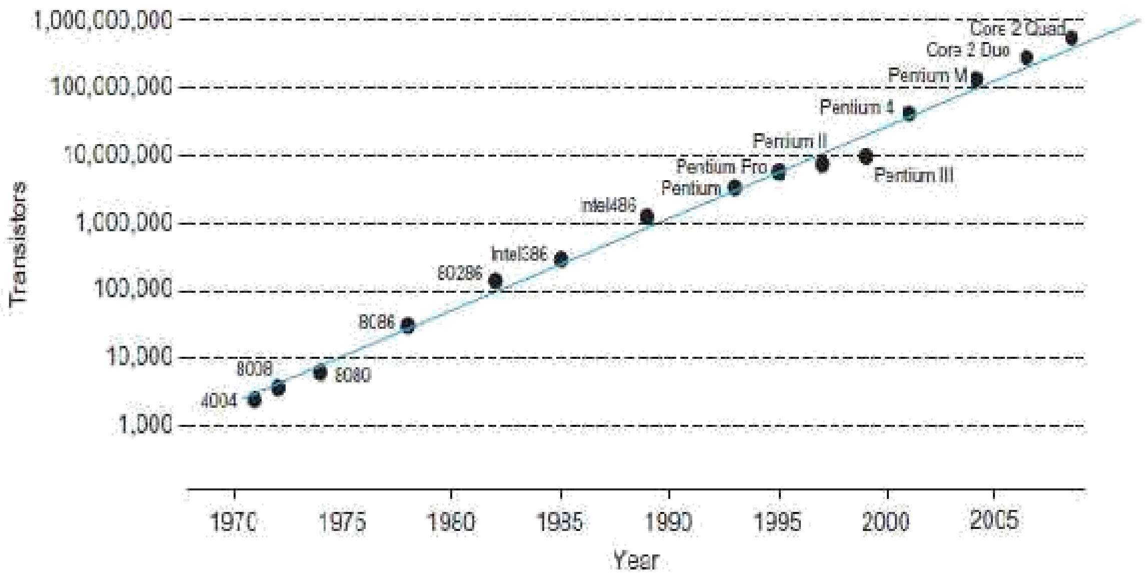
UNIT I - MOS TRANSISTOR PRINCIPLE

MOS Transistor, CMOS logic, Inverter, Pass Transistor, Transmission gate, Layout Design Rules, Gate Layouts, Stick Diagrams, Long-Channel I-V Characteristics, C-V Characteristics, Non ideal I-V Effects, DC Transfer characteristics, RC Delay Model, Elmore Delay, Linear Delay Model, Logical effort, Parasitic Delay, Delay in Logic Gate, Scaling.

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.

****

<u>MOS transistor</u>

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.

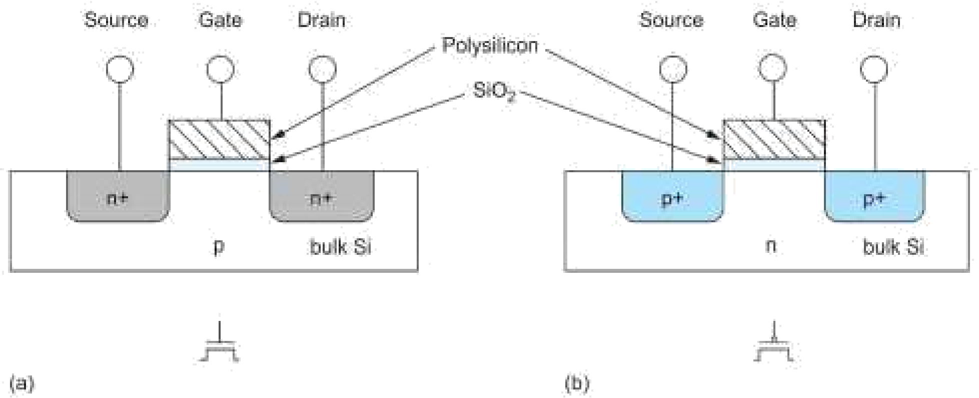


Figure 1: (a) n-MOS transistor (b) p-MOS transistor

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).

Modes of MOS TRANSISTOR

Explain the accumulation 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 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.

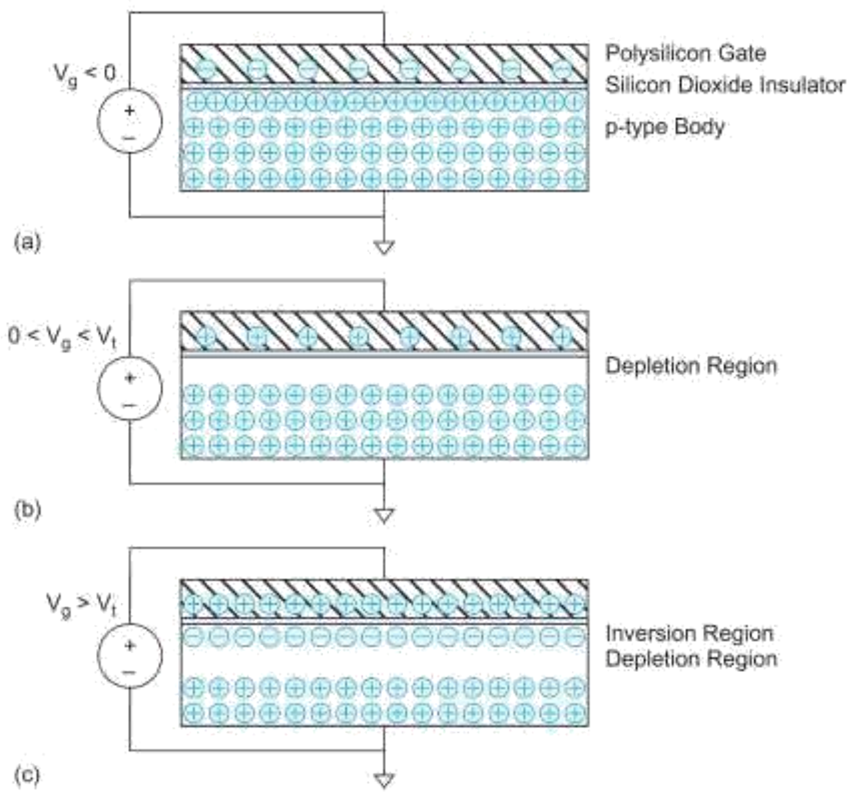


Figure 2: MOS structure demonstrating (a) accumulation, (b) depletion, and (c) inversion layer

<u>Operating regions of MOS transistor:</u>

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 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.

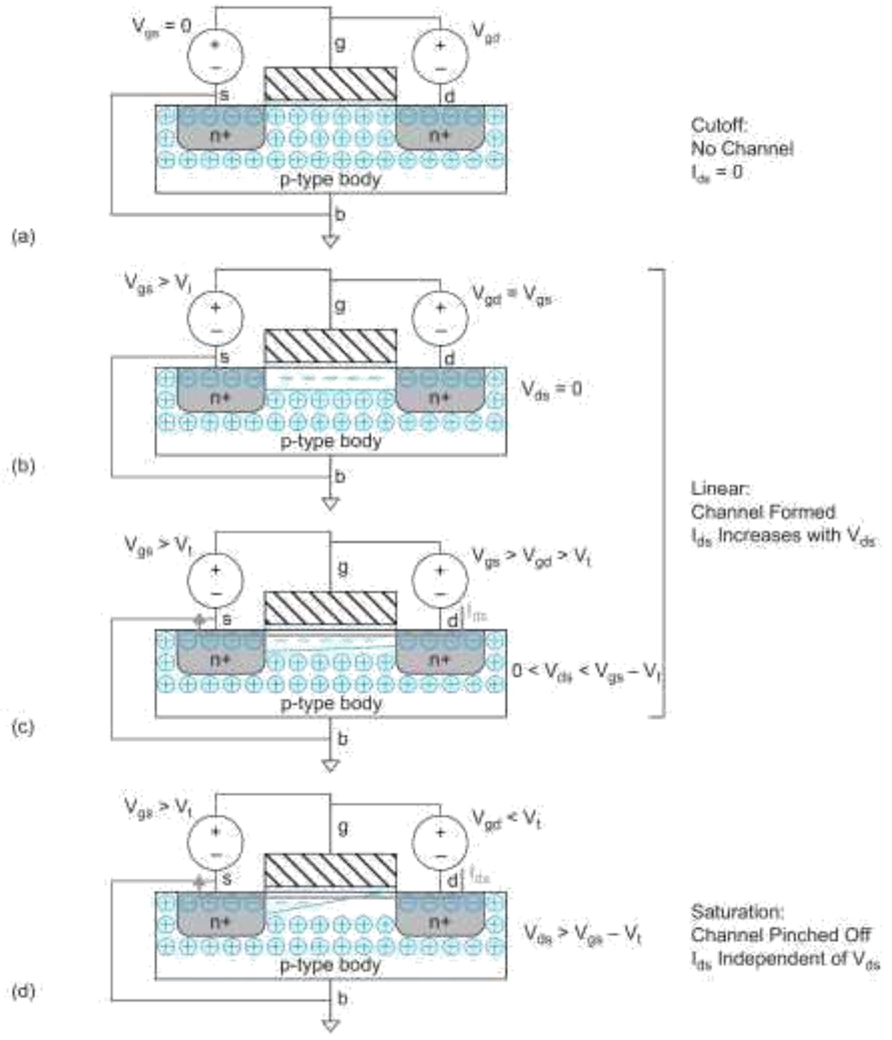


Figure 3: nMOS transistor demonstrating cutoff, linear, and saturation regions of operation

- ✓ 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.

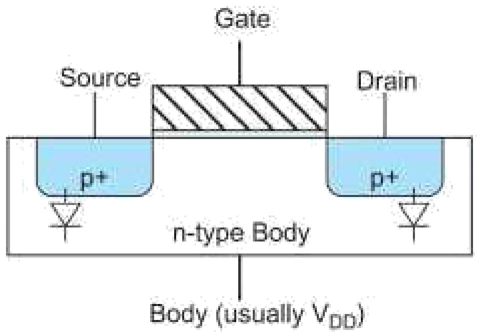


Figure 4: pMOS transistor

****

IDEAL I-V CHARACTERISTICS OF MOS TRANSISTOR

- ❖ **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)**

- ✓ MOS transistor has three regions of operation:
  - Cutoff or sub threshold region
  - Linear 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 $V_g – V_c = V_{gs} – V_{ds}/2$,

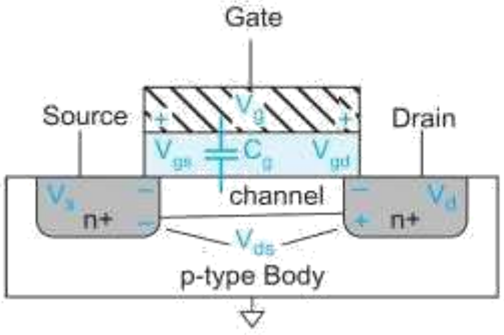


Figure 5: Average gate to channel voltage

- ✓ 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,

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

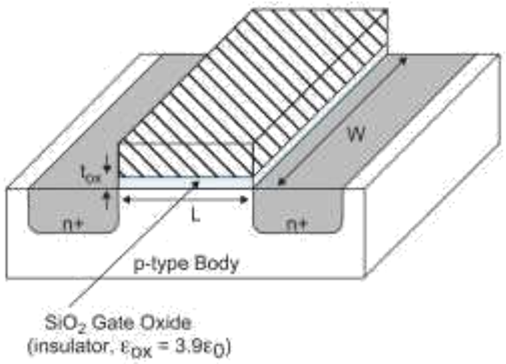


Figure 6: Transistor dimensions

- ✓ 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.

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

where

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

- ✓ 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' = μ C_{ox}.
- ✓ If $V_{ds} > V_{dsat} = V_{GT}$, 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 $V_{ds} = V_{dsat}$ in Eq (4), we can find an expression for the saturation current (I_{ds}) that is independent of V_{ds}.

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

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

- ✓ Summarizes the current in the three regions:

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

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)**

- ✓ Each terminal of an MOS transistor has capacitance to the other terminals.

- ✓ Capacitances are nonlinear and voltage dependent (C-V).

SIMPLE MOS CAPACITANCES MODEL:

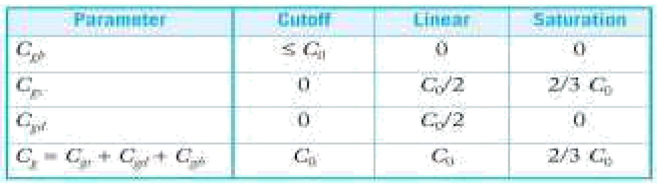
- ✓ The gate of an MOS transistor is a good capacitor. Its capacitance is necessary to attract charge to invert the channel, so high gate capacitance is required to obtain high I_{ds}.
- ✓ The gate capacitor can be viewed as a parallel plate capacitor with the gate on top, channel on bottom and the thin oxide dielectric between.
- ✓ The capacitance is $C_g = C_{ox} WL$. ----------------(1)

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

- ✓ 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

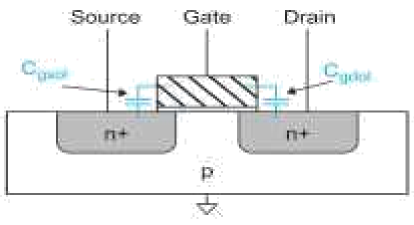


Figure 8: Overlap capacitances

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.

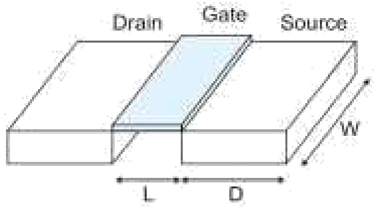


Figure 9: Diffusion region geometry

- ✓ 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.

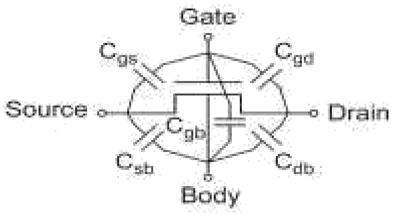


Figure 10: Capacitance of a MOS Transistor

- ✓ 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.

****

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)**
- ❖ **Explain the CMOS inverter DC characteristics. (Nov 2007, Nov 2009)**

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.

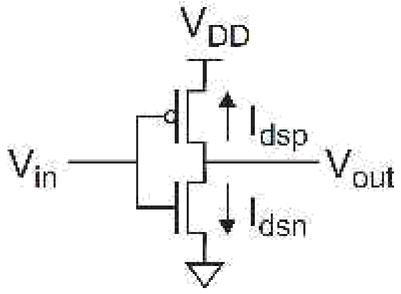


Figure: A Static CMOS inverter

- ✓ Table 2, shows various regions of operation for the n and p transistors.

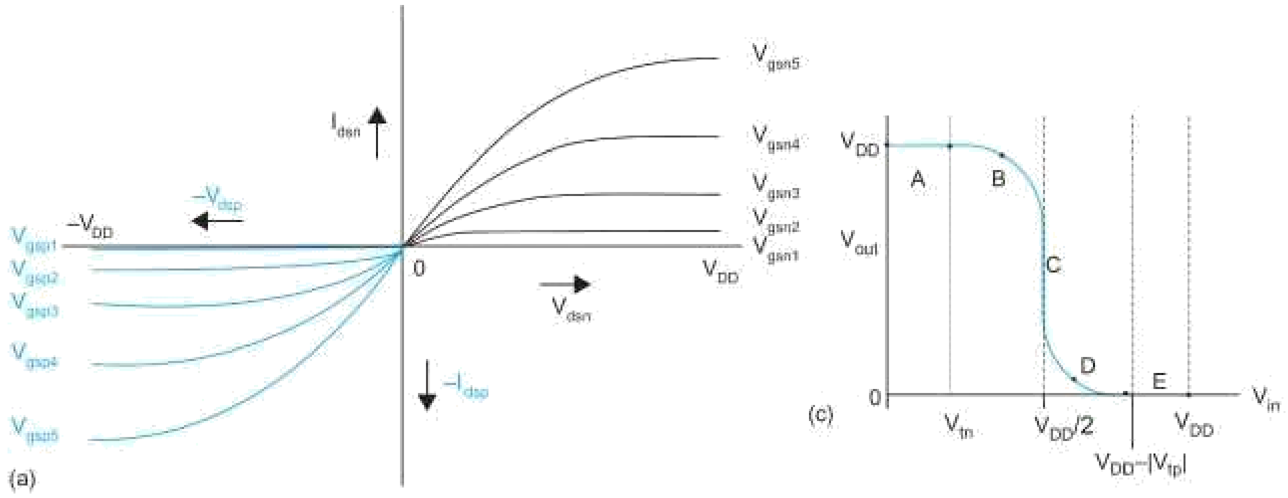
- ✓ In this table, V_{tn} is the threshold voltage of the n-channel device, and V_{tp} is the threshold voltage of the p-channel device. V_{tp} is negative.
- ✓ The equations are given both in terms of V_{gs} /V_{ds} and V_{in} /V_{out}.

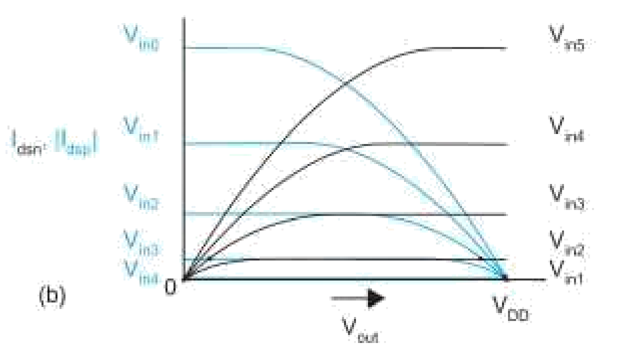
- ✓ As the source of the nMOS transistor is grounded, $V_{gsn} = V_{\text{in}}$ and $V_{dsn} = V_{\text{out}}$.
- ✓ As the source of the pMOS transistor is tied to V_{DD}, $V_{gsp} = V_{\text{in}} - V_{DD}$ and $V_{dsp} = V_{\text{out}} - V_{DD}$.

	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).





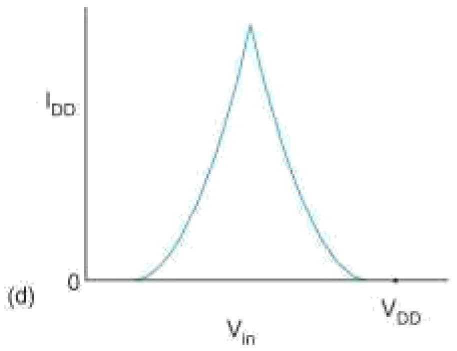


Figure 12: CMOS inverter DC characteristic

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

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

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 $r = \beta_p / \beta_n$ 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.

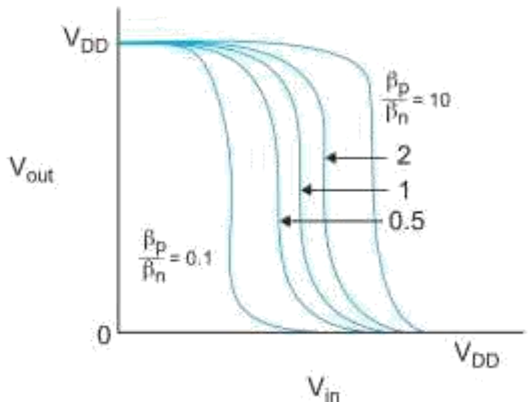


Figure 13: Transfer characteristics of skewed inverters

❖ **Derive the noise margins for a CMOS inverter. (May 2010, Nov2016)**

❖ (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).

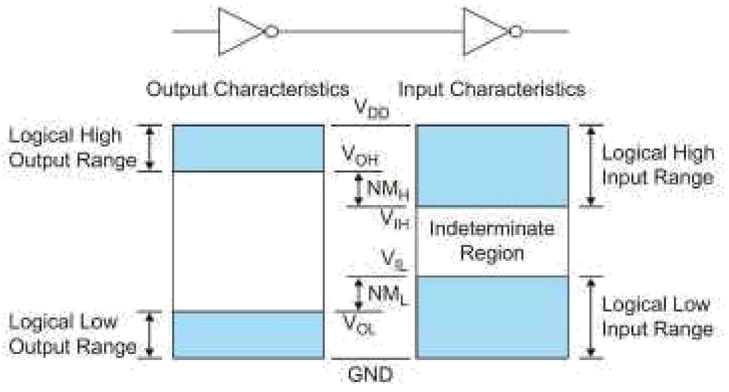


Figure 14: Noise Margin Definitions

✓ 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).

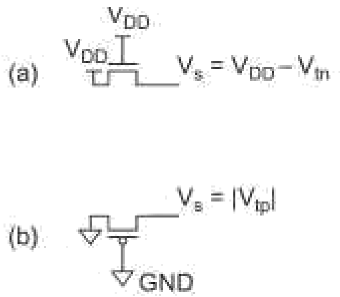


Figure 15: Pass Transistor threshold drops

****

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.

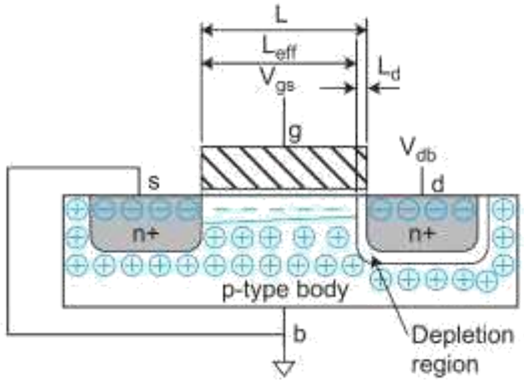


Figure 16: Depletion region shortens effective channel length

✓ 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.

Threshold Voltage (V_t) Effects

Explain in detail about Threshold Voltage 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.

<u>SCALING</u>

- ❖ **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 2107)**

- ✓ 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

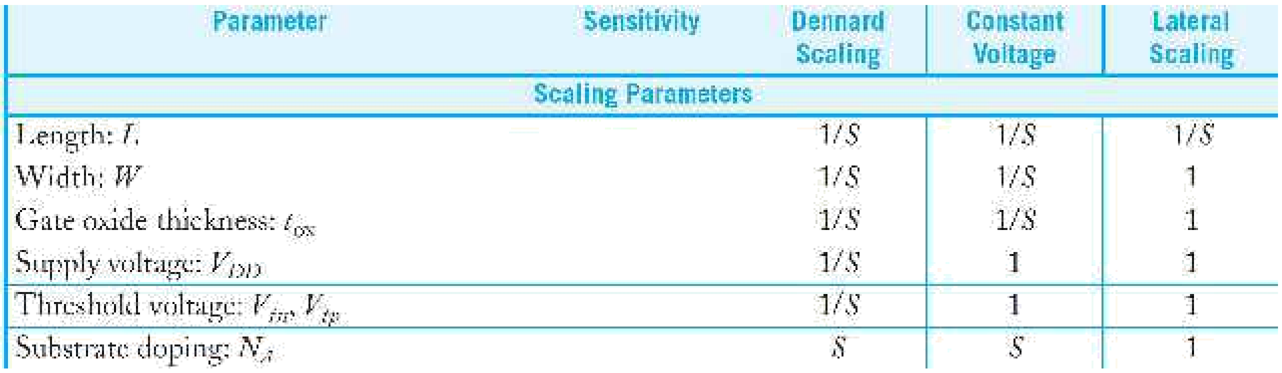
Constant field scaling (Full Scaling):

- ✓ 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 (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.

****

Delay estimation

- ✓ 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.

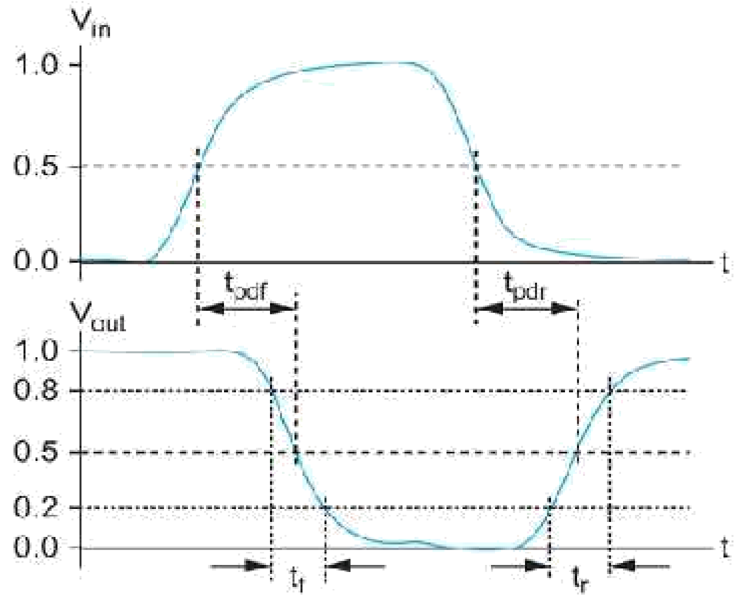


Figure: Delay estimation of CMOS inverter

- ✓ 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 $t_P = (t_{PLH} + t_{PHL}) / 2$

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.

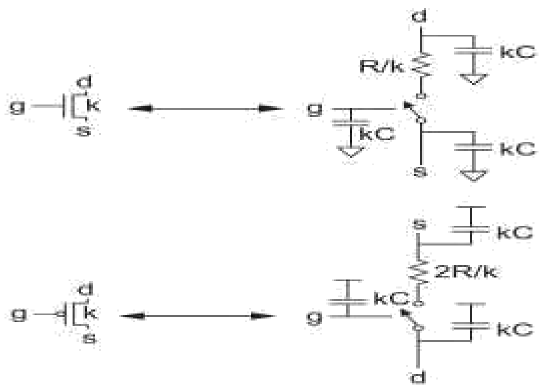


Figure: RC model of nMOS &pMOS transistors

Stick diagram

Explain about stick diagram in VLSI design. (April 2008)

- ✓ 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.

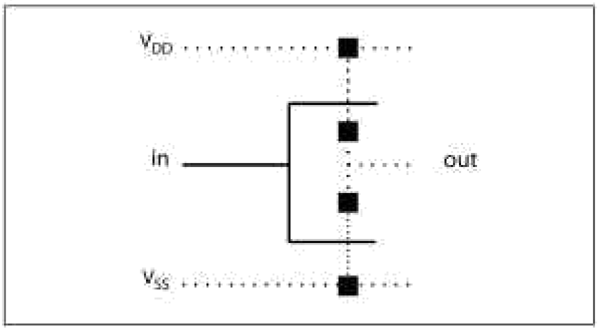


Figure 17: Stick diagram for an inverter

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

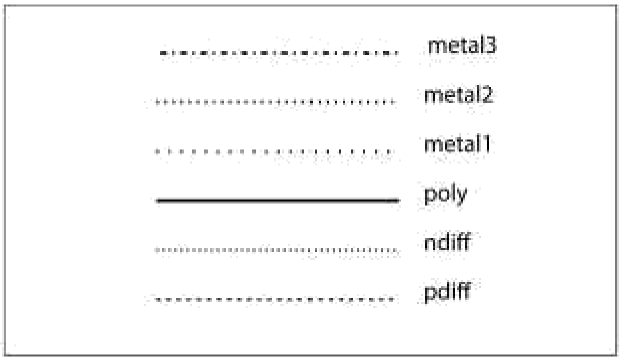


Figure : Symbols for wires used on various layers

- ✓ **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.

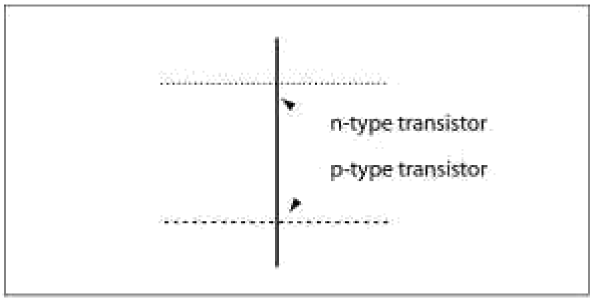
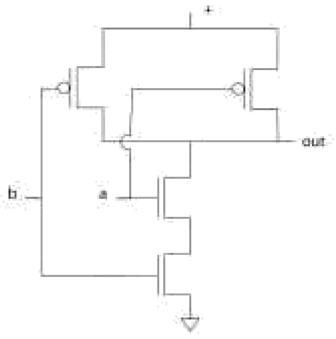


Figure: Stick figures for transistors

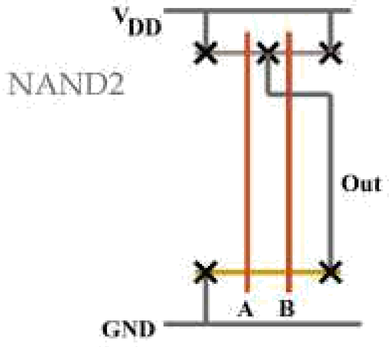
- ✓ Area and aspect ratio are also difficult to estimate from stick diagrams.
- ✓ Stick diagrams are especially important tools for layouts built from large cells and for testing the connections between cells.

Example:

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



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



<u>Layout Design Rules</u>

- ❖ **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)**

- ✓ 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.

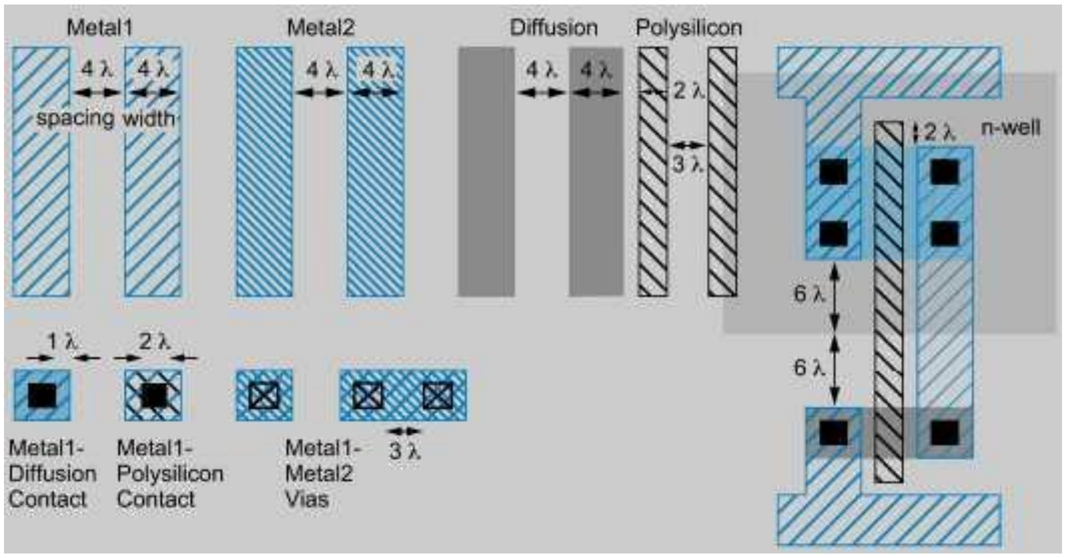


Figure: Simplified λ -based design rules with CMOS inverter layout diagram

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.

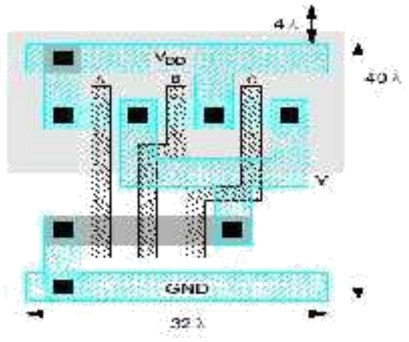
Via Rules:

- ✓ 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

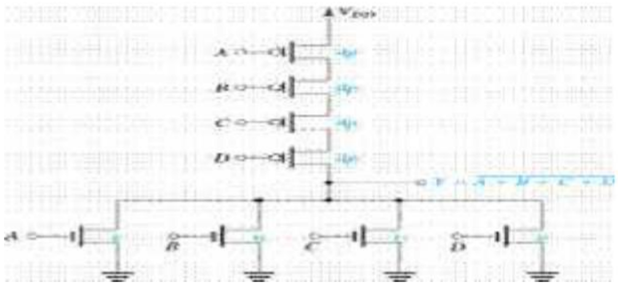
Draw the layout diagram of NAND. (May 2017)

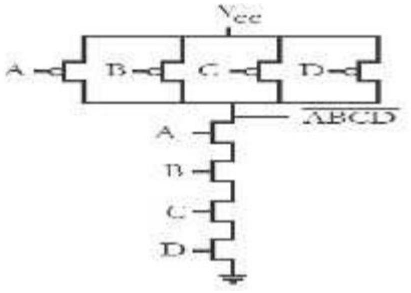
- Horizontal N-diffusion and p-diffusion strips
- Vertical polysilicon gates
- Metal1 V_{DD} rail at top
- Metal1 GND rail at bottom



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

4 input NOR gate **4 input NANA gate**

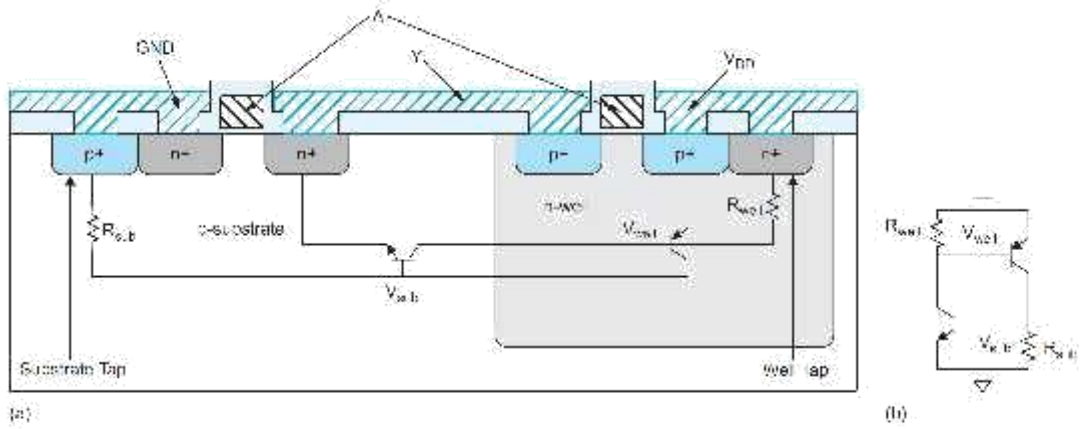




1.14: Latchup problem:

- ❖ ***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.

Process parameters for MOS and CMOS:

CMOS 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)

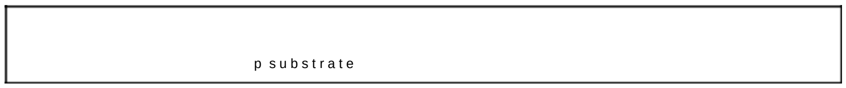
Describe with neat diagram the n-well and channel formation in CMOS process. (Nov/Dec 2014)(Nov/Dec 2011) (April/May 2011) (Nov/Dec 2012)

n-WELL PROCESS:

Step 1: Start with blank wafer

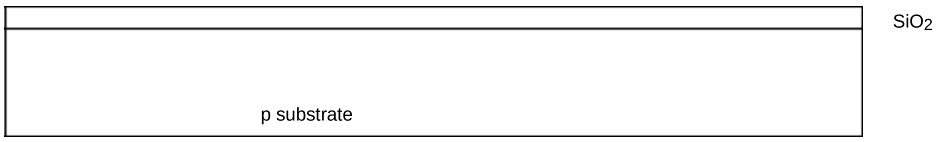
First step will be to form the n-well

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



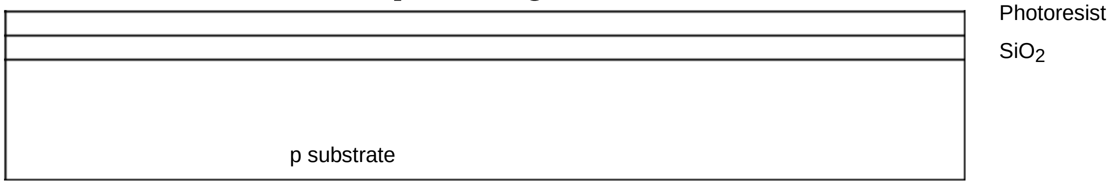
Step 2: Oxidation

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



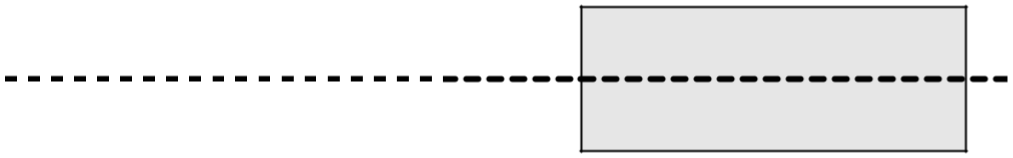
Step 3: Photoresist

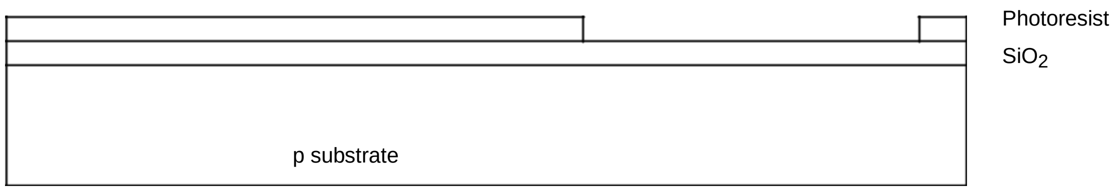
- Spin on photoresist
  - Photoresist is a light-sensitive organic polymer.
  - Softens, where exposed to light.



Step 4: Lithography

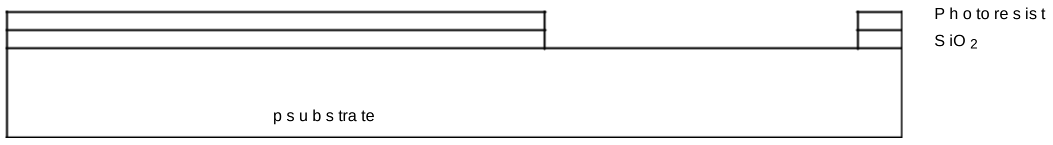
- Expose photoresist through n-well mask.
- Strip off exposed photoresist.





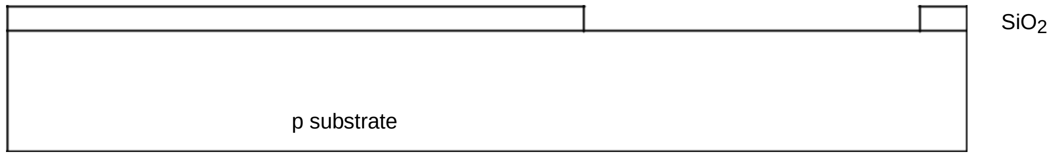
Step 5: Etch

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



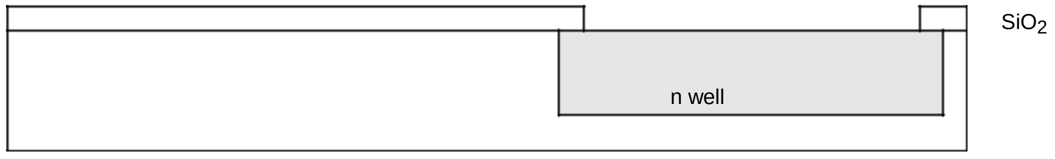
Step 6: Strip Photoresist

- Etch the remaining photoresist using a mixture of acids.



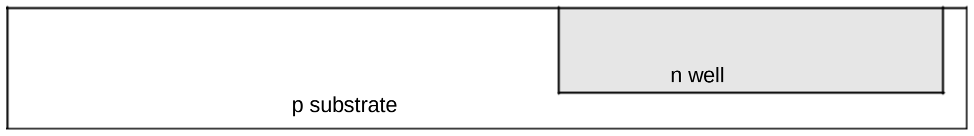
Step 7: n-well

n-well is formed with diffusion or ion implantation.



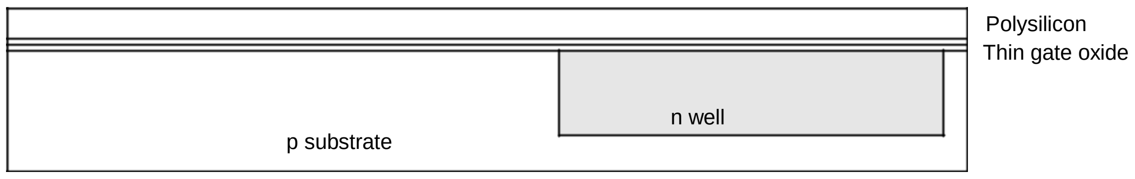
Step 8: Strip Oxide

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



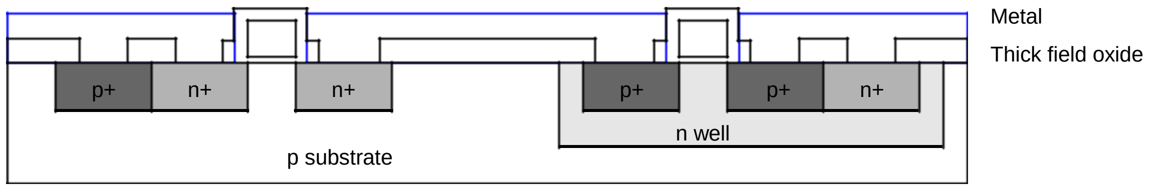
Step 9: Polysilicon

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



Step 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:

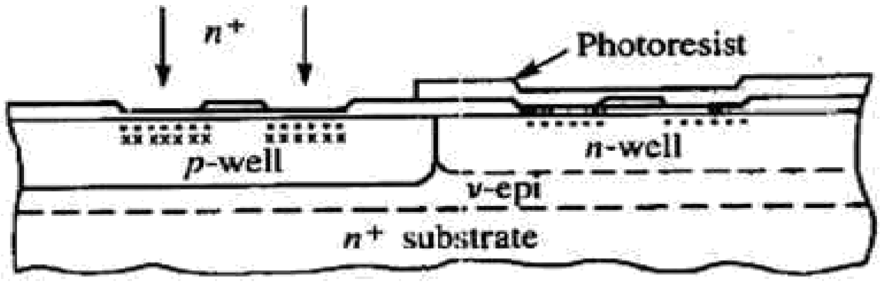
Polysilicon gates are formed for n-well and p-well by using photo-etching process.

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.



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 :

$$t_{pd} \approx \sum_{\text{nodes } i} R_{i-to-source} C_i$$

$$R_1C_1 + (R_1 + R_2)C_2 + \ldots + (R_1 + R_2 + \ldots + R_N)C_N$$

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

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

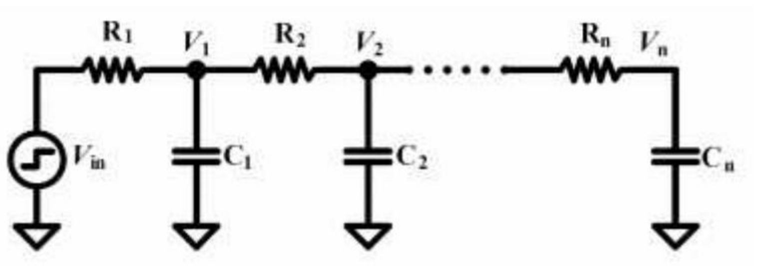


Figure: RC delay equivalent for series of transistors

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 Y 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.

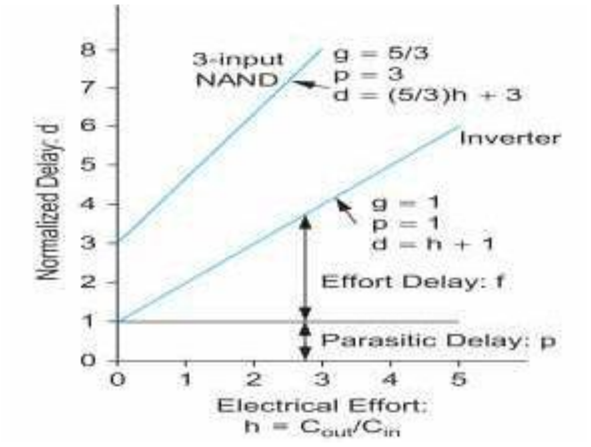
The electrical effort can be computed as $h = \frac{C_{out}}{C_{in}}$

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.



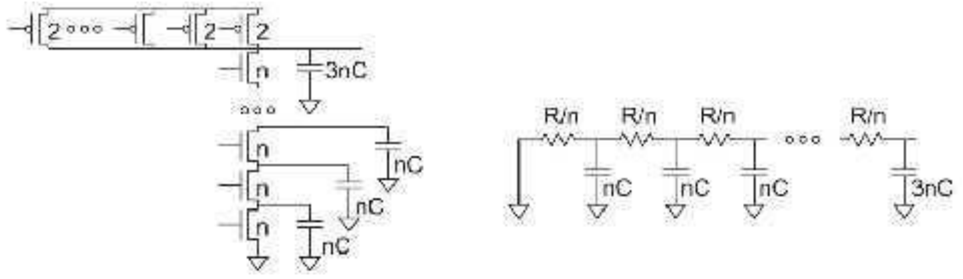
Design a four input NAND gate and obtain its delay during the transition from high to low. (April 2018)

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

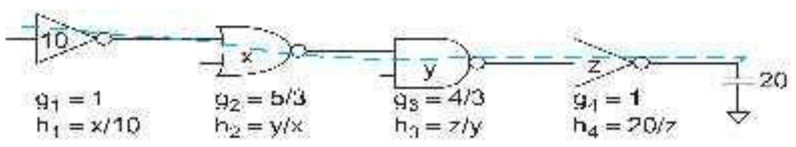


Delay for 4 input NAND gate: $\left(\frac{n^2}{2} + \frac{5}{2}n\right)RC = \left(\frac{4^2}{2} + \frac{5}{2}4\right)RC = \left(\frac{16}{2} + \frac{20}{2}\right)RC = 18RC$

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

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

$\hat{f} = g_i h_i = F^{1/N}$

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.

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:

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

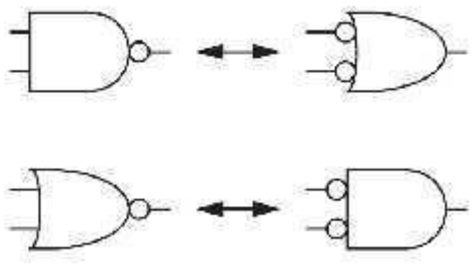


Figure: Bubble pushing with DeMorgan's law

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.

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.

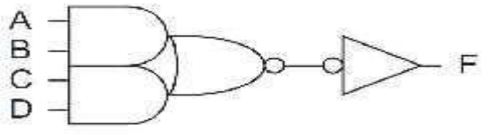


Figure: Logic using AOI22 gate

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.

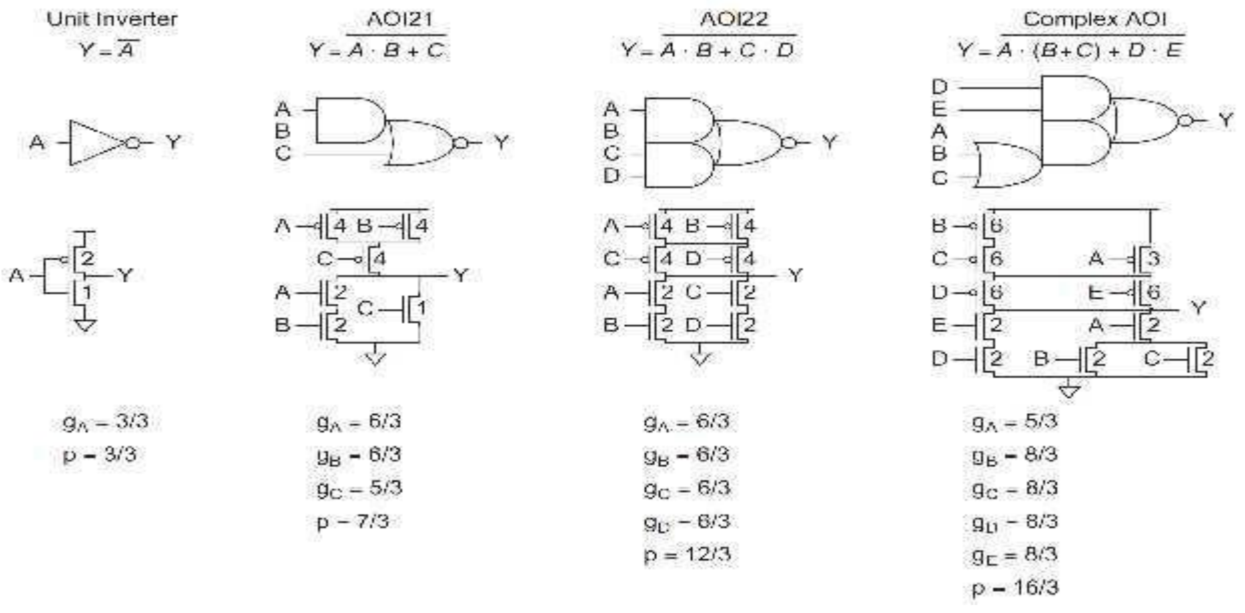


Figure: Logical efforts and parasitic delays of AOI gates

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.

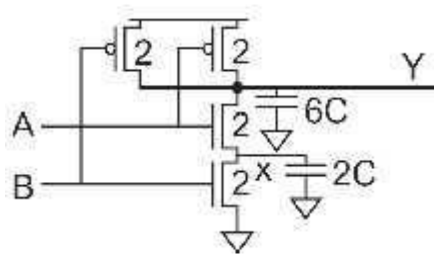


Figure: 2 –input NAND gate Schematic Y=A.B

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).

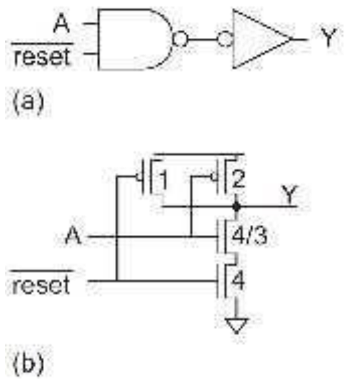


Figure: Resettable buffer optimized for data input

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 pull down current, shown in Figure (c), giving a logical effort of g_d =2.5/1.5 =5/3.

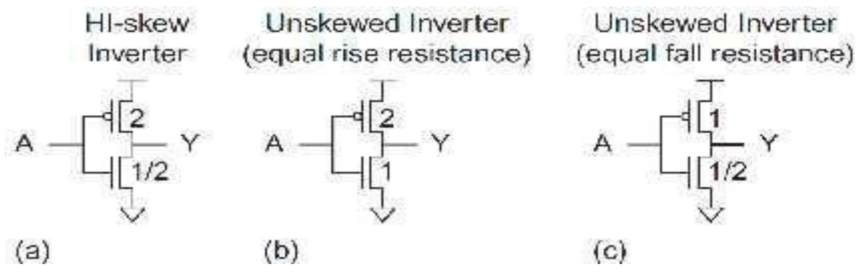

HI-skew Inverter
Unskewed Inverter (equal rise resistance)
Unskewed Inverter (equal fall resistance)
A
Y
2
1/2
1
(a)
(b)
(c)

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.

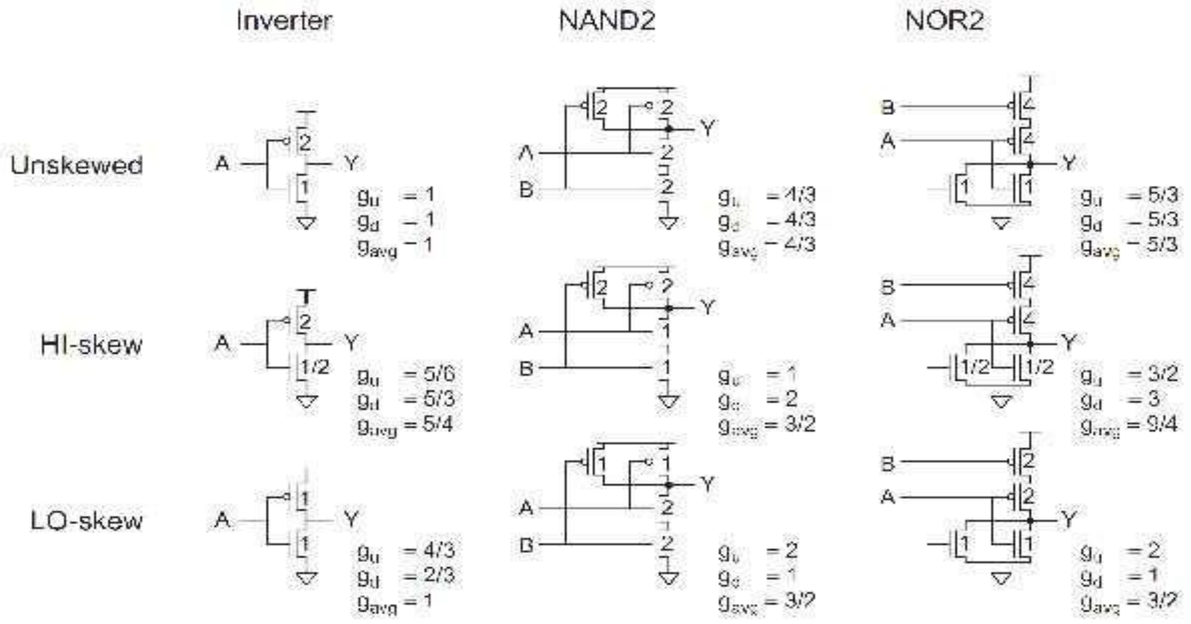


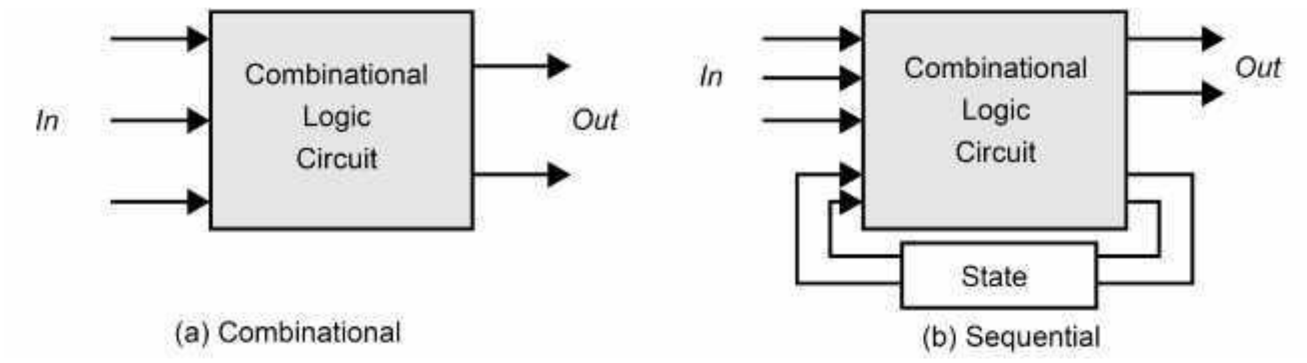
Figure: List of skewed gates

UNIT II - COMBINATIONAL LOGIC CIRCUITS

Circuit Families: Static CMOS, Ratioed Circuits, Cascode Voltage Switch Logic, Dynamic Circuits, Pass Transistor Logic, Transmission Gates, Domino, Dual Rail Domino, CPL, DCVSPG, DPL, Circuit Pitfalls.
Power: Dynamic Power, Static Power, Low Power Architecture.

Introduction (Combinational Logic Circuit):

- 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 V.



- 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:

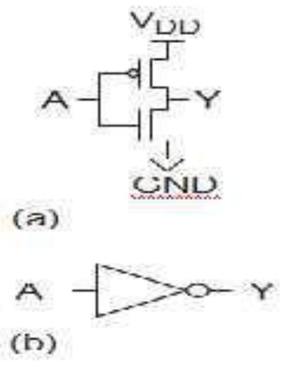


Figure: Inverter (a) schematic (b) symbol $Y = \bar{A}$

Prepared by: B.ARUNKUMAR,AP/ECE

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

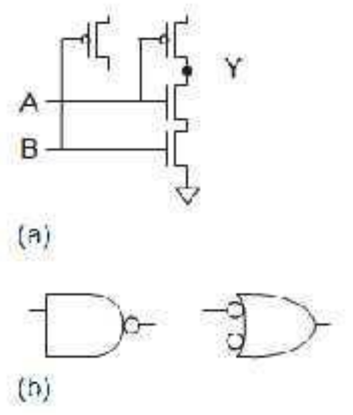


Figure: 2 input NAND gate (a) schematic (b) symbol

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

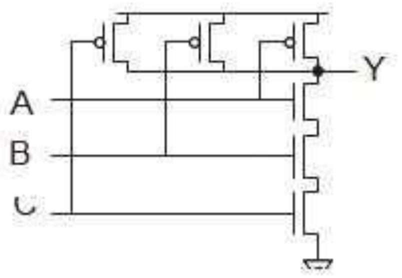


Figure: 3 –input NAND gate Schematic Y=A.B.C

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

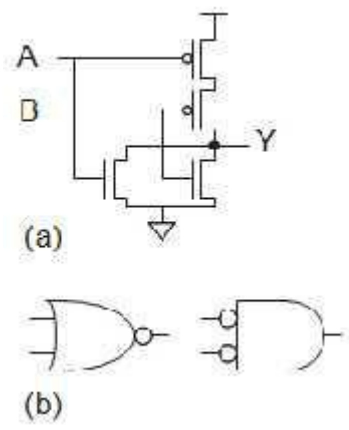
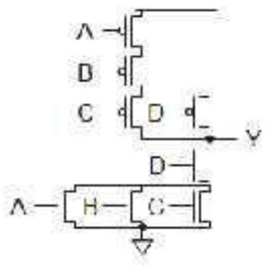


Figure: 2-input NOR gate (a) schematic (b) Symbol Y = $\overline{A + B}$

Example:

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



****

Prepared by: B.ARUNKUMAR,AP/ECE

Circuit Families- Static CMOS

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.

2.2.1: Static CMOS

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

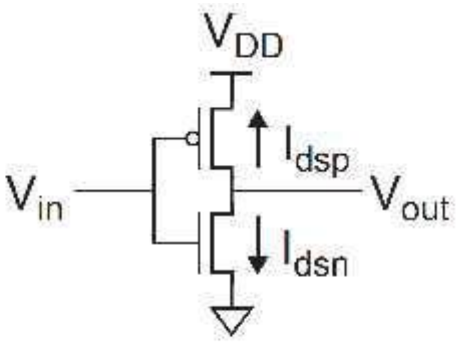


Figure: Static CMOS inverter

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 ncreased.

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:

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

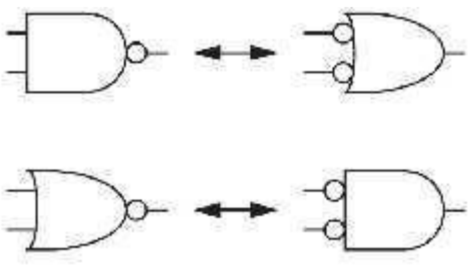


Figure: Bubble pushing with DeMorgan's law

- 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.

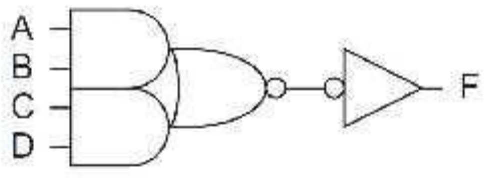


Figure: Logic using AOI22 gate

- 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.

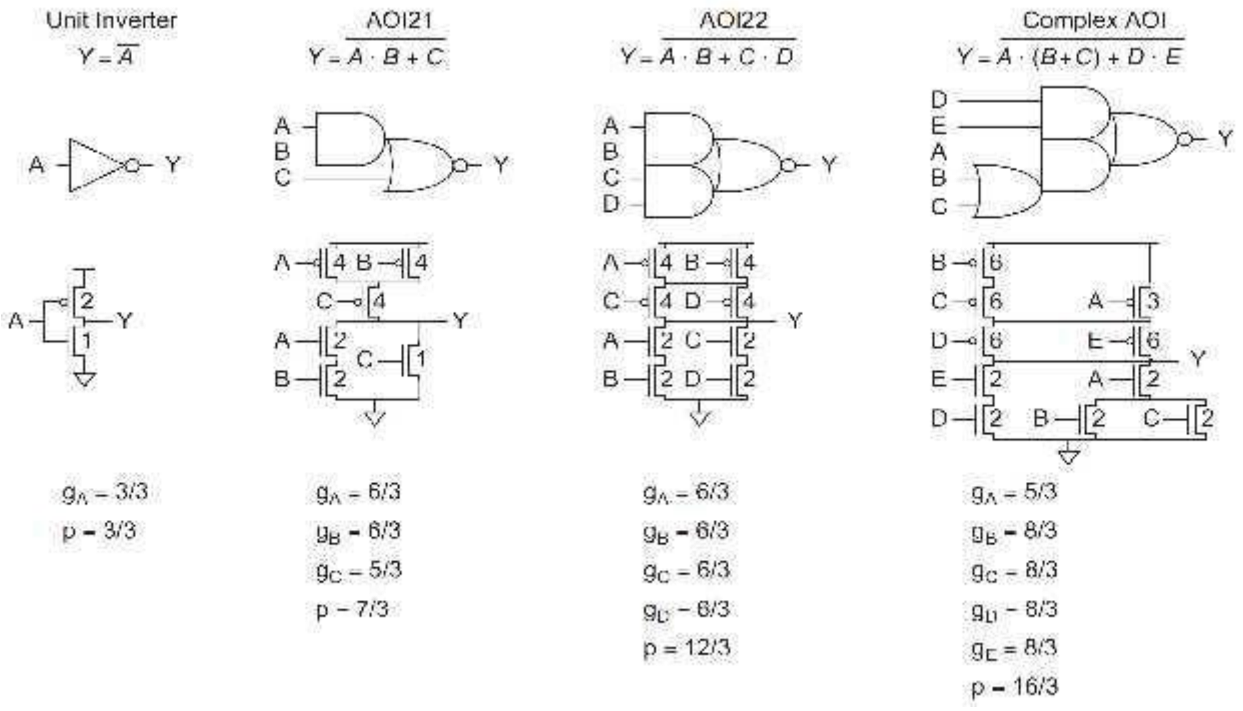


Figure: Logical efforts and parasitic delays of AOI gates

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.

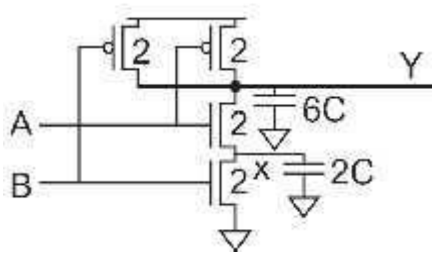


Figure: 2 –input NAND gate Schematic Y=A.B

- 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).

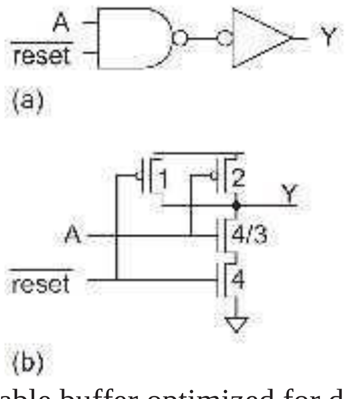


Figure: Resettable buffer optimized for data input

2.4: Pass Transistor circuits:

Explain Pass transistor logic with neat sketches. (April 2008)

- 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).

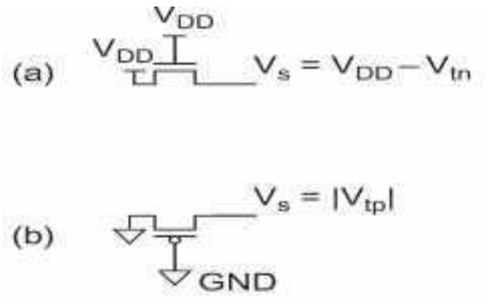
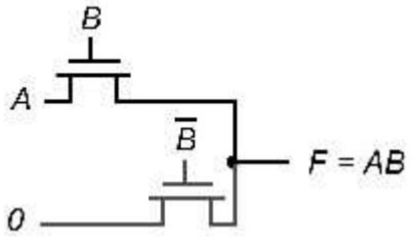
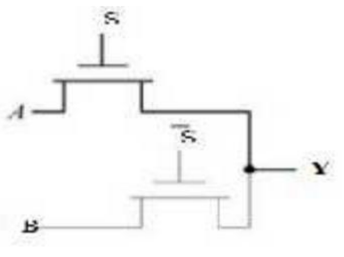


Figure : Pass Transistor threshold drops

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





AND Logic 2x1 multiplexer

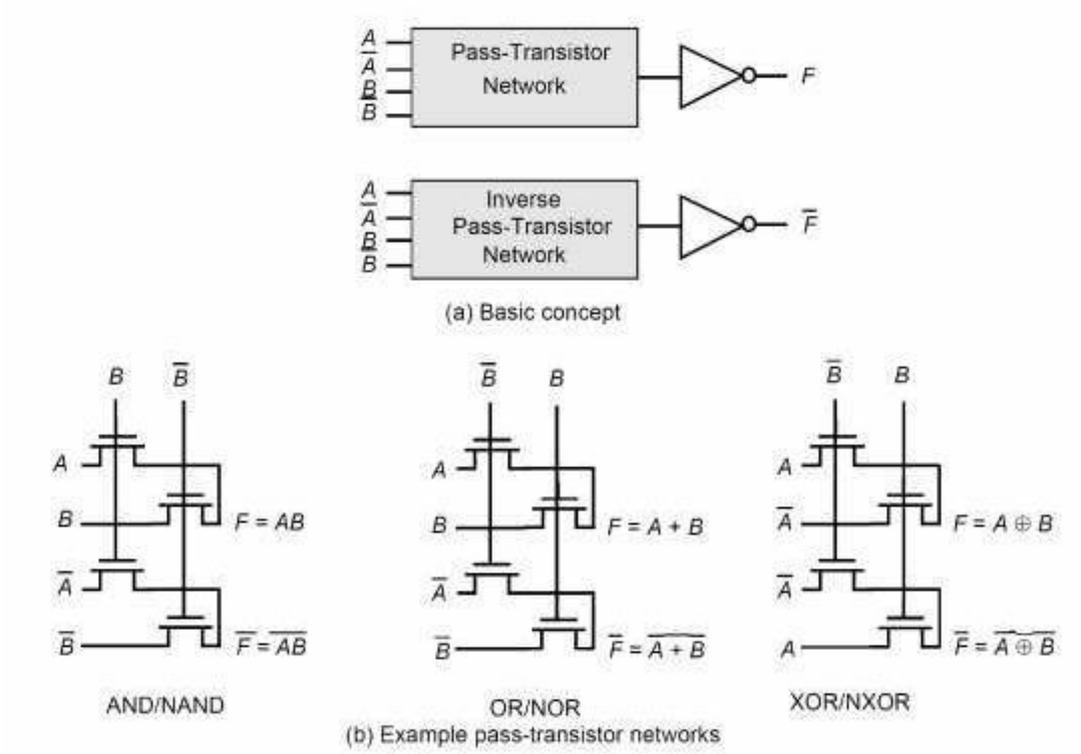
- In AND gate, if the *B* input is high, the top transistor is turned ON and copies the input *A* to the output *F*.
- When *B* is low, the bottom pass transistor is turned ON and passes a 0.
- In 2x1 multiplexer, if the S selection input is high, the top transistor is turned ON and allows input *A* 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.

2.4.1: Differential Pass Transistor Logic

- For high performance design, a differential pass-transistor logic family, called CPL or DPL, 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
$\overline{F}$
(a) Basic concept
$F = AB$
$\overline{F} = \overline{AB}$
AND/NAND
$F = A + B$
$\overline{F} = \overline{A + B}$
OR/NOR
$F = A \oplus B$
$\overline{F} = \overline{A \oplus B}$
XOR/NXOR
(b) Example pass-transistor networks

Figure: Complementary pass-transistor logic (CPL).

2.4.2: 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.

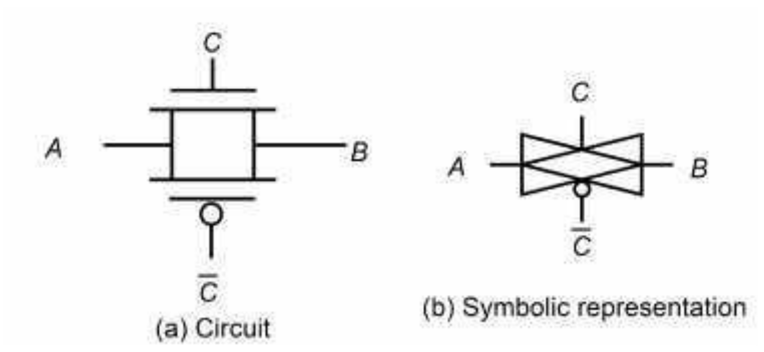


Figure: CMOS Transmission gate

- 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.

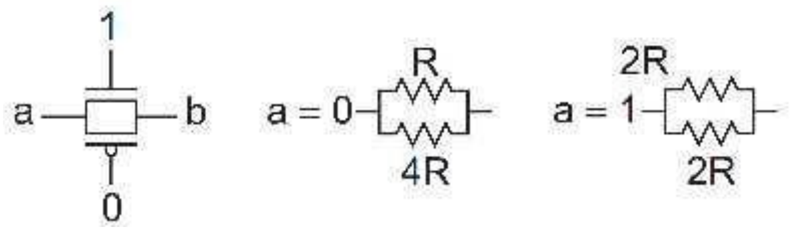


Figure: Effective resistance of a unit transmission gate

- 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.

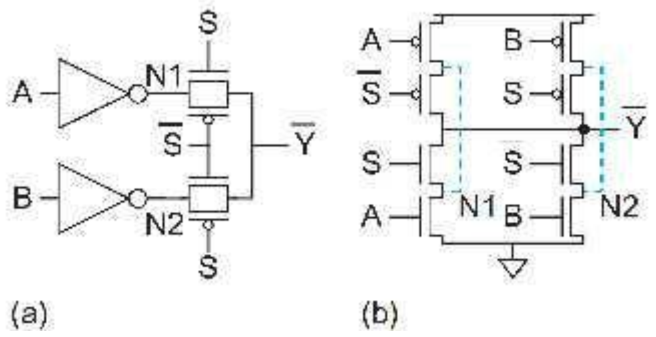


Figure: CMOSTG in a 2-input inverting multiplexer

2.5: Static CMOS design- Ratioed Circuits:

2.5.1: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:

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

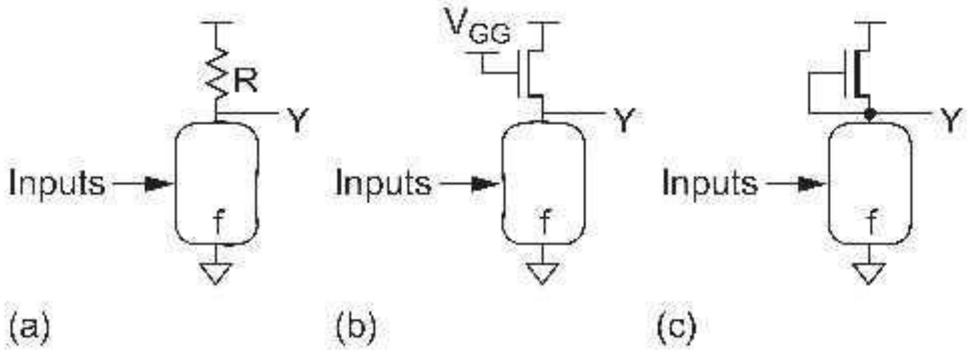
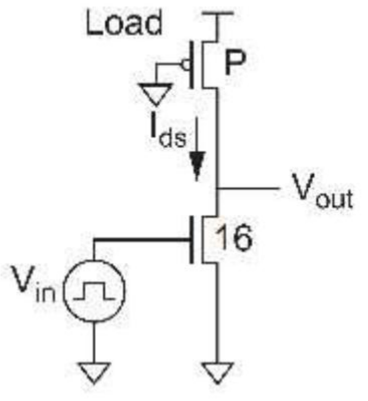


Figure: nMOS ratioed gates

2.5.2: 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.



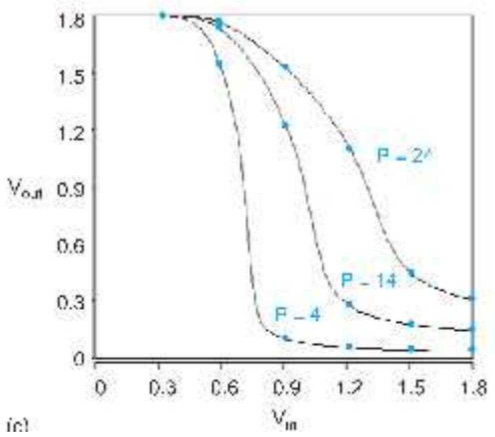


Figure (a): pseudo nMOS inverter Figure: D.C Charateristics

- Figure shows several pseudo-nMOS logic gates.

Implement NAND gate using pseudo- nMOS logic. (Nov 2013)

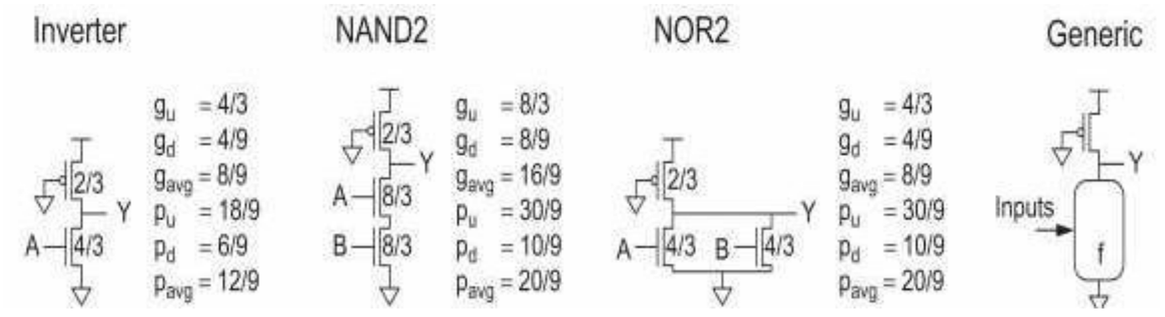


Figure: Pseudo-nMOS logic gates

2.5.3: 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.

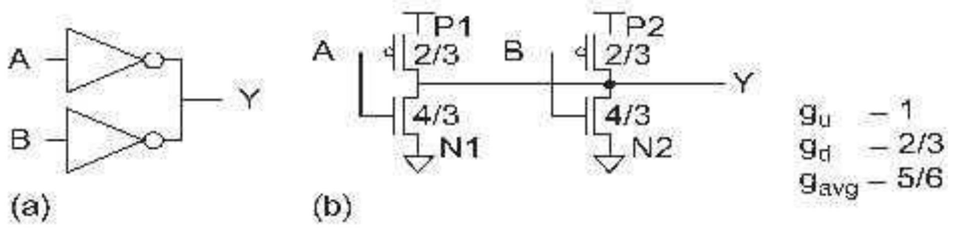


Figure: symmetric ^{2} NOR gate.

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

Prepared by: B.ARUNKUMAR,AP/ECE

- 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.

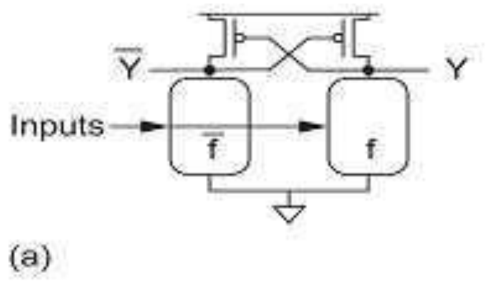
2.5.4: Cascode voltage switch logic

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.



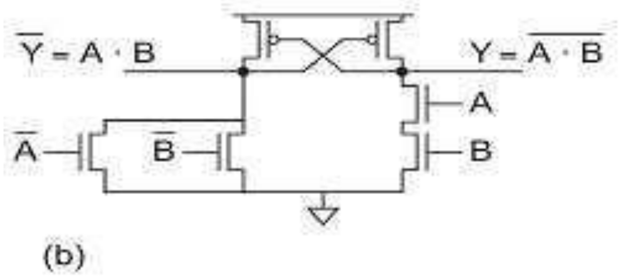


Figure: CVSL gates

****

2.6:Dynamic CMOS design:

Describe the basic principle of operation of dynamic CMOS, domino and NP domino logic with neat diagrams. (NOV 2011)

Dynamic Circuits:

Write short notes on Dynamic CMOS circuits. (Nov 2016)

- 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.

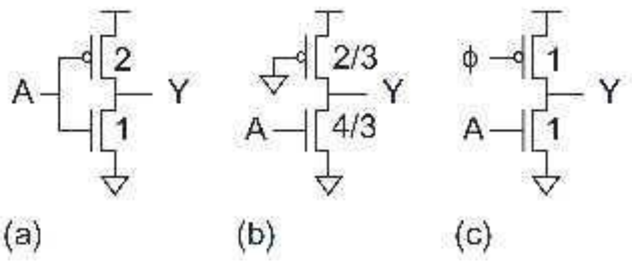


Figure: Comparison of (a) static CMOS, (b) pseudo-nMOS, and (c) dynamic inverters

- 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.

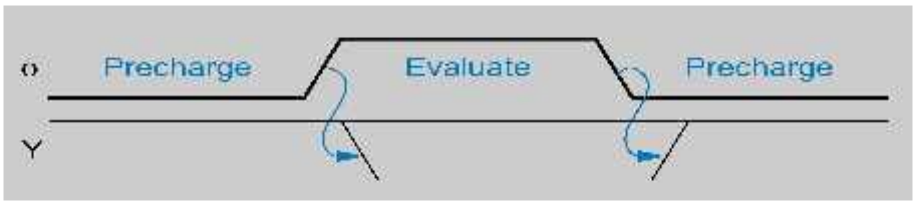


Figure: Precharge and evaluation of dynamic gates

Advantages:

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

Disadvantags:

- 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.

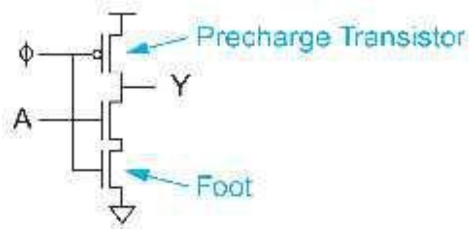


Figure: Footed dynamic inverter

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

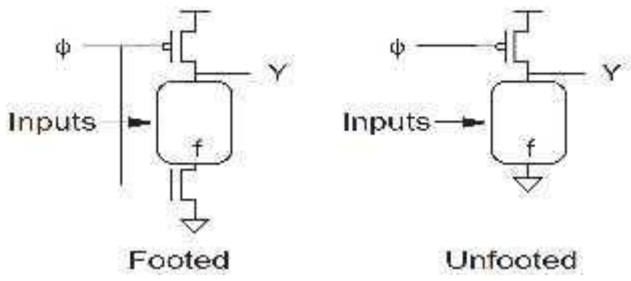


Figure: Generalized footed and unfooted dynamic gates

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

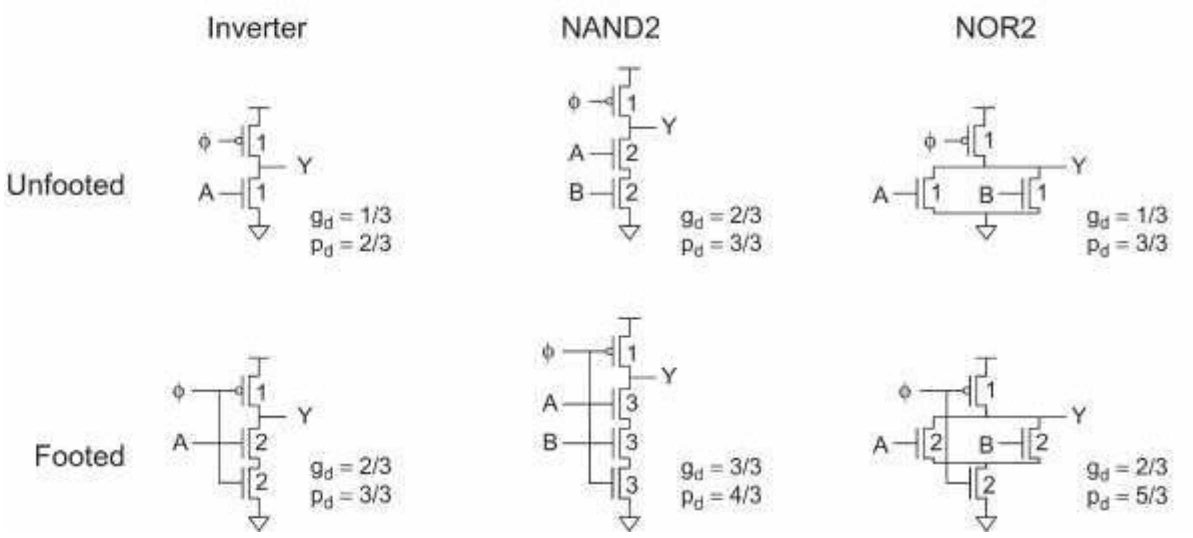


Figure: List of dynamic gates

- 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.

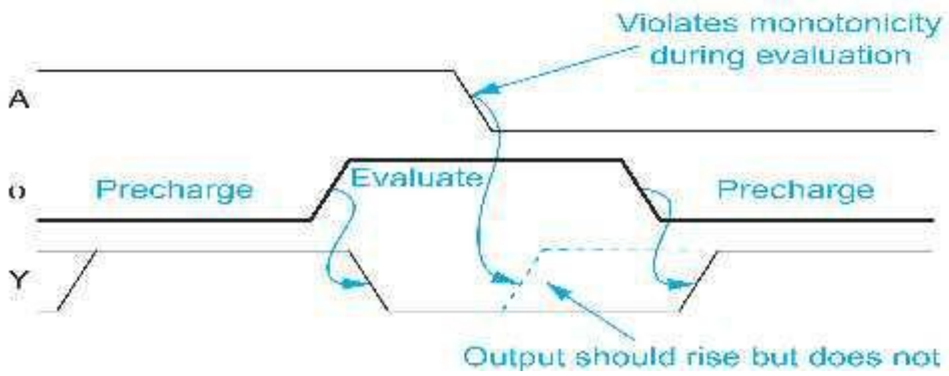

Violates monotonicity during evaluation
A
ϕ
Precharge
Evaluate
Precharge
Y
Output should rise but does not

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.

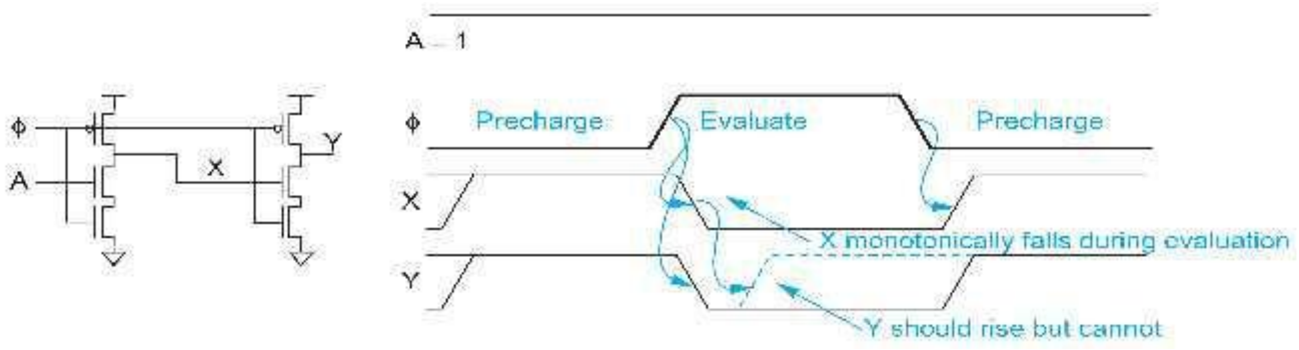


Figure: Incorrect connection of dynamic gates

Prepared by: B.ARUNKUMAR,AP/ECE

2.6.1: 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.

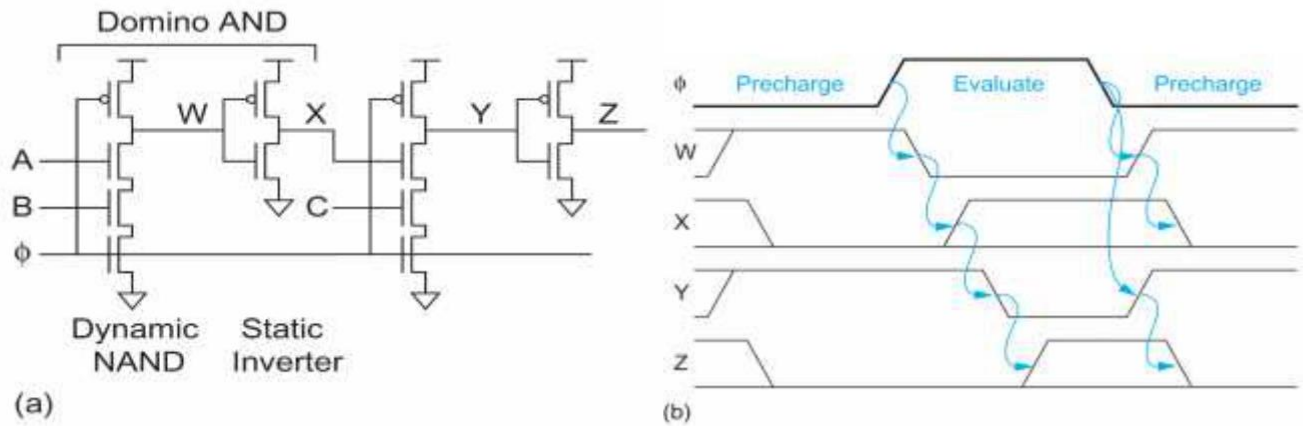
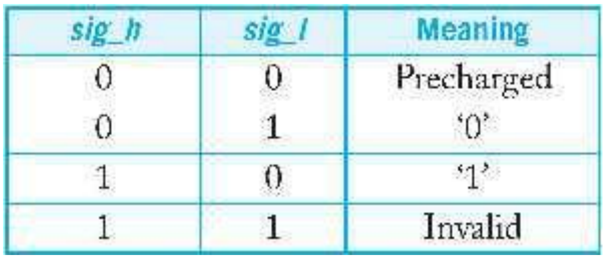


Figure: Domino gates

2.6.2: 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.

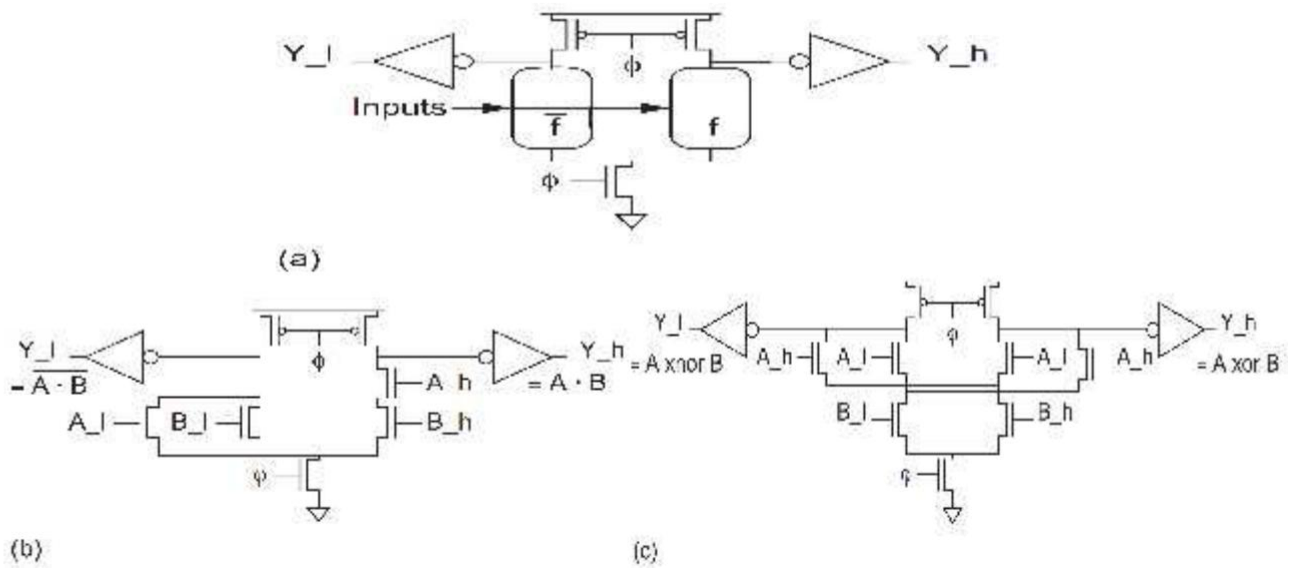


Figure: Dual-rail domino gates

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.

<u>2.6.3: Keepers</u>

Explain the keeper logic family with neat diagrams.
Briefly discuss the signal integrity issues in dynamic design. (April 2018, NOV 2018)
(Ans. Except 2.6.3.1 and 2.6.6)

- Dynamic circuits also suffer from charge leakage on the dynamic node.
- If a dynamic node is precharged high and then left floating, the voltage on the dynamic node will drift over time due to subthreshold, gate and junction leakage.
- 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.

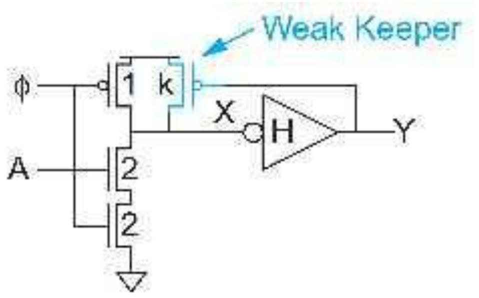


Figure: Conventional keeper

- 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.

2.6.3.1: 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.

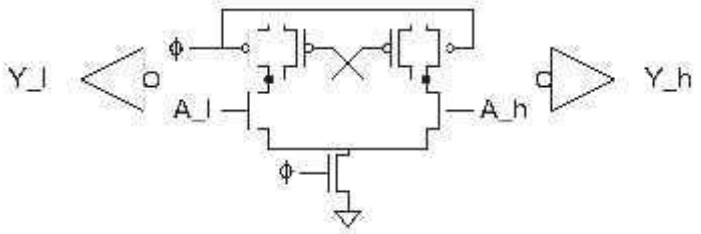


Figure: Differential keeper

2.6.4: 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.

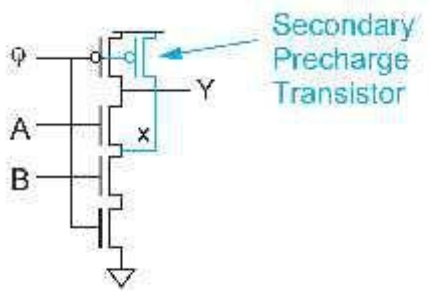


Figure: Secondary prechagre transistor

Prepared by: B.ARUNKUMAR,AP/ECE

- 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

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

2.6.5: 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.

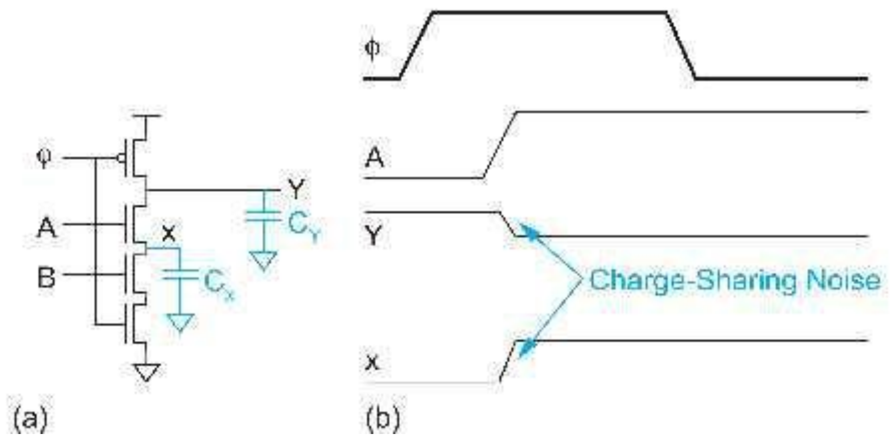


Figure: Charge-sharing noise

2.6.6: 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.

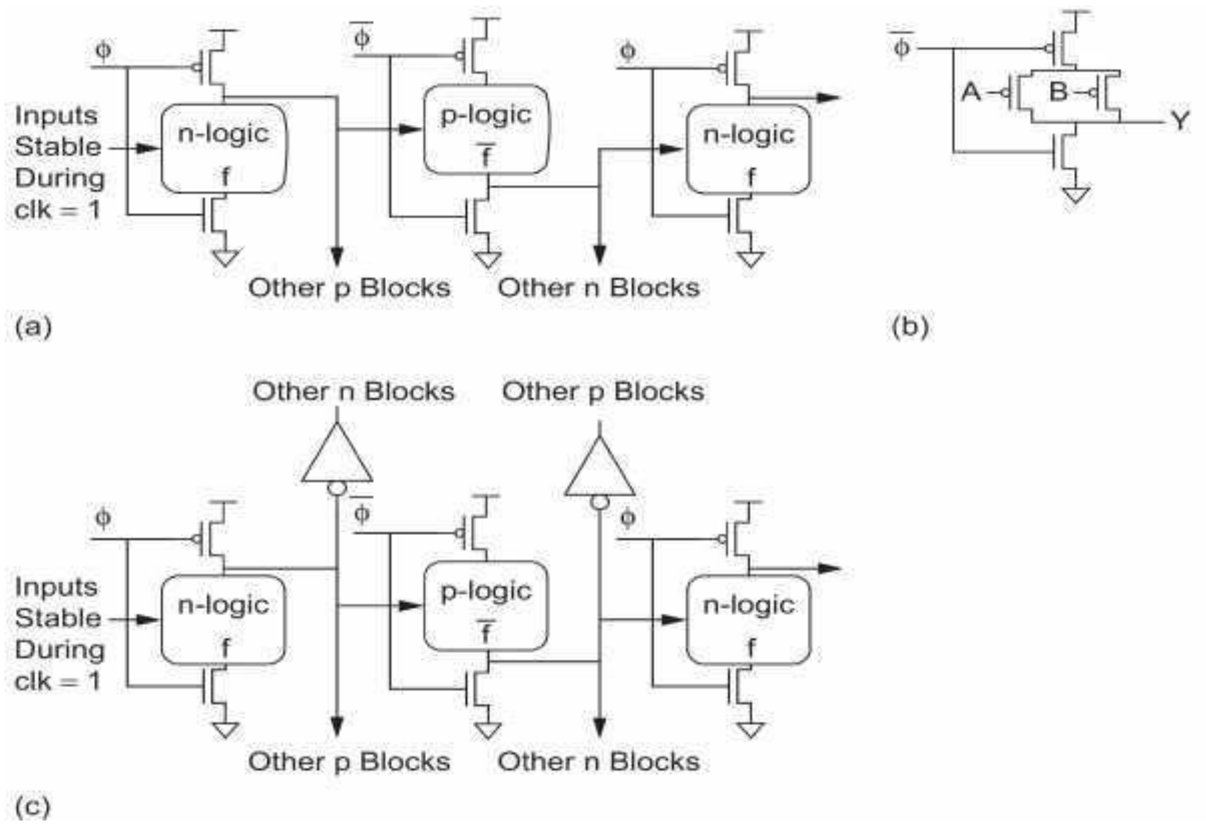


Figure : NP Domino

****

2.7: 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)**

- 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 $E = \int_0^T P(t)\,dt$

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

  Power is expressed in units of Watts (W). Energy is usually expressed in Joules (J)

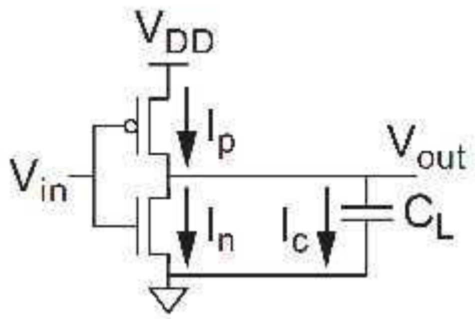
- By Ohm's Law, V = IR, so the instantaneous power dissipated in the resistor is

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

- 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

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

- 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

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

- 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

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

- 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

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

- 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,

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

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

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

2.7.1: Dynamic power:

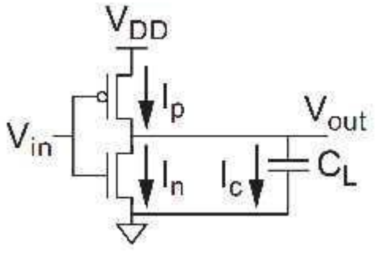
- Dynamic power consists mostly of the switching power.

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

- 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.

2.7.1.1: 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



Prepared by: B.ARUNKUMAR,AP/ECE

2.7.1.2: 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)**

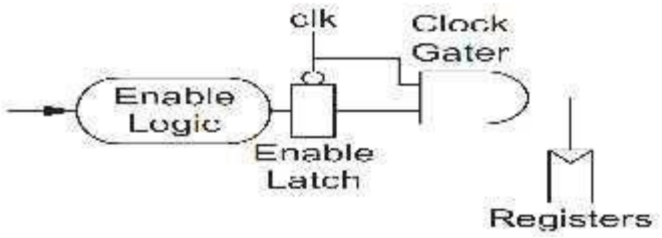
- Low power design involves considering and reducing each of the terms in switching power.
  - i. As V_{DD} is a quadratic term, it is good to select the minimum V_{DD}.
  - ii. Choose the lowest frequency.
  - iii. The activity factor is reduced by putting unused blocks to sleep.
  - iv. Finally, the 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:

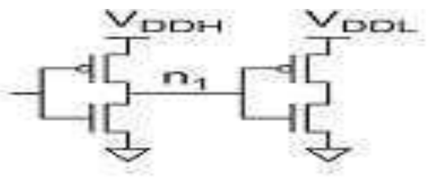
- Switching capacitance comes from the wires and transistors in a circuit.
- Wire capacitance is minimized through good floor planning and placement.
- Device-switching capacitance is reduced by choosing smaller transistors.

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.

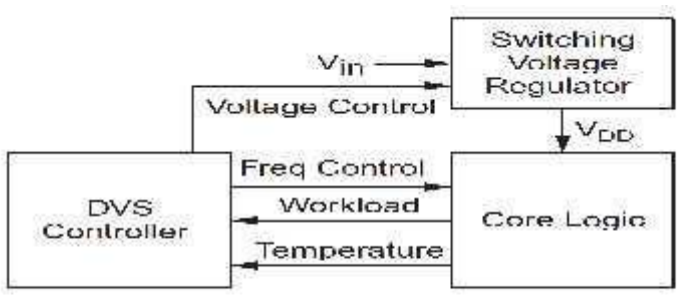
a. **Voltage domains:**

- 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):**

- 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).



- Figure shows a block diagram for a basic DVS system.
- It determines the supply voltage and clock frequency sufficient to complete the workload on schedule or to maximize performance without overheating.

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.

2.7.2: Static power:

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

2.7.2.1: 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

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

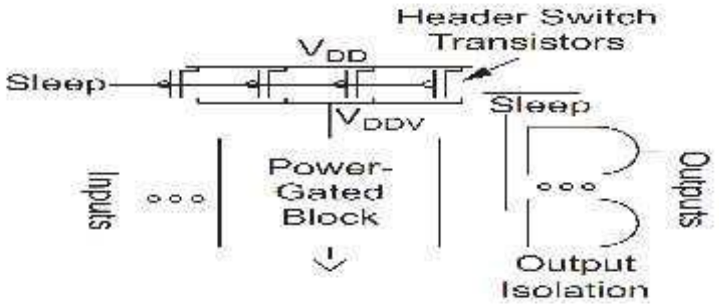
where I_{off} is the subthreshold current at $V_{gs} = 0$ and $V_{ds} = V_{DD}$, and S is the subthreshold slope.

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.

2.7.2.2: Methods of reducing static power:

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.



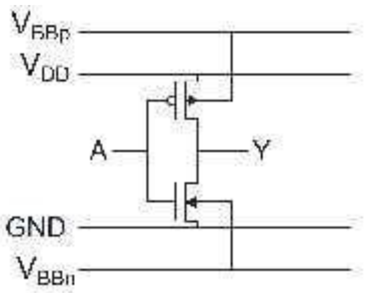
- The logic block receives its power from a virtual V_{DD} rail, V_{DDV}.
- When the block is active, the header switch transistors are ON, connecting V_{DDV} to V_{DD}.
- When the block goes to sleep, the header switch turns OFF, allowing V_{DDV} to float and gradually sink toward 0.

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.



****

Prepared by: B.ARUNKUMAR,AP/ECE

VLSI circuit design for low power:

- The growing market of portables such as cellular phones, gaming consoles and battery-powered electronic systems demands microelectronic circuits design with ultra low power dissipation.
- As the integration, size, and complexity of the chips continue to increase, the difficulty in providing adequate cooling might either add significant cost or limit the functionality of the computing systems which make use of those integrated circuits
- As the technology node scales down to 65nm, there is not much increase in dynamic power dissipation. However the static or leakage power reaches or exceeds the dynamic power levels beyond 65nm technology node.
- Hence the techniques to reduce power dissipation are not limited to dynamic power. In this, we discuss circuit and logic design approaches to minimize dynamic, leakage and Total Power dissipated in a CMOS circuit is sum of dynamic power, short circuit power and static or leakage power.
- Design for low-power implies the ability to reduce all three components of power consumption in CMOS circuits during the development of a low power electronic product.
- In the sections to follow we summarize the most widely used circuit techniques to reduce each of these components of power in a standard CMOS design.

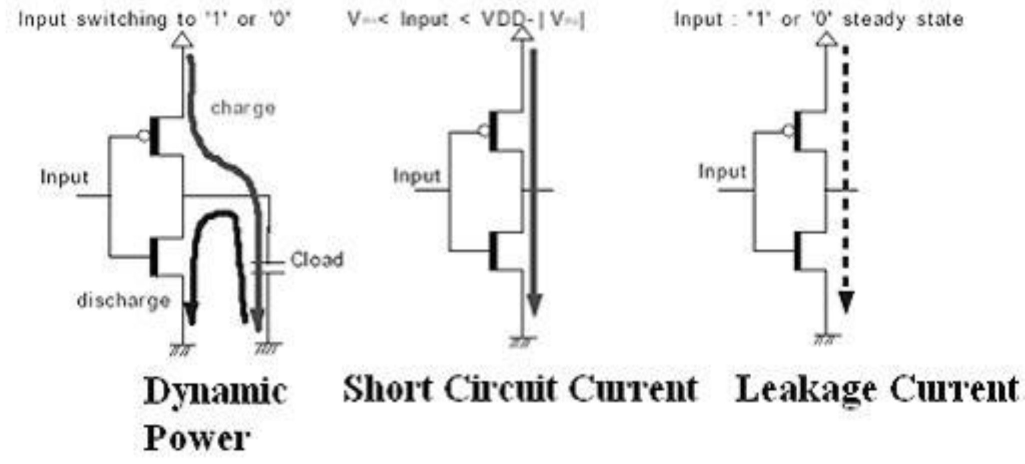


Figure: Components of Power in CMOS circuit

P total = CLVDD2 + tscVDDIpeak + VDDIleakage

Dynamic Power Suppression

- Dynamic/Switching power is due to charging and discharging of load capacitors driven by the circuit. Supply voltage scaling has been the most adopted approach to power optimization, since it normally yields considerable power savings due to the quadratic dependence of switching/dynamic power Pswitching on supply voltage VDD.

- However lowering the supply voltage affects circuit speed which is the major short-coming of this approach. So both design and technological solutions must be applied to compensate the decrease in circuit performance introduced by reduced voltage. Some of the techniques often used to reduce dynamic power are described below.

Logic Design for Low Power:

Choices between static versus dynamic topologies, conventional CMOS versus pass-transistor logic styles and synchronous versus asynchronous timing styles have to be made during the design of a circuit.

- In static CMOS circuits, the component of power due to short circuit current is about 10% of the total power consumption.

- However in dynamic circuits we don't come across this problem, since there is no any direct dc path from supply voltage to ground.

- Only in domino-logic circuits there is such a path, in order to reduce sharing, hence there is a small amount of short-circuit power dissipation.

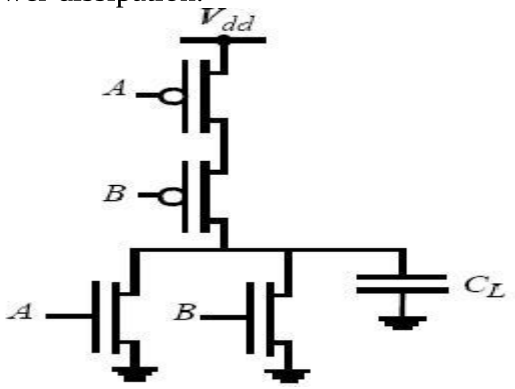


Figure: Static NOR

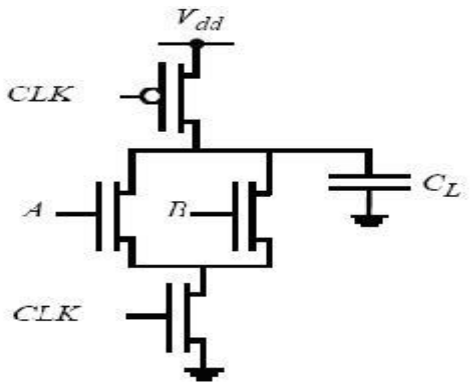


Figure: Dynamic NOR circuits

P = CL* Vdd* (Vdd-Vt)

Logic Level Power Optimization:

During logic optimization for low power, technology parameters such as supply voltage are fixed, and the degrees of freedom are in selecting the functionality and sizing the gates.

- Path equalization with buffer insertion is one of the techniques which ensure that signal propagation from inputs to outputs of a logic network follows paths of similar length to overcome glitches.

- When paths are equalized, most gates have aligned transitions at their inputs, thereby minimizing spurious switching activity/glitches (which is created by misaligned input transitions).

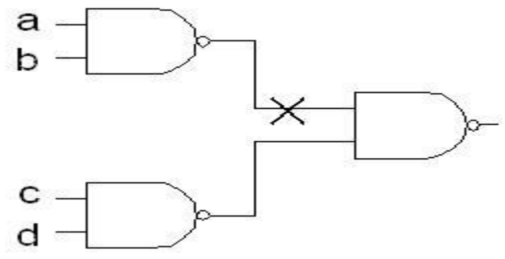


Figure: Logic Remapping for Low Power

- Other logic-level power minimization techniques include local transformations as shown in figure above.
- A re-mapping transformation is shown, where a high-activity node (marked with x) is removed and replaced by new mapping onto an and or gate.

Standby Mode Leakage Suppression:

- Static/Leakage power, originates from substrate currents and sub threshold leakages. For technologies 1 µm and above, PSwitching was predominant.
- However for deep-submicron processes below 180nm, PLeakage becomes dominant factor. Leakage power is a major concern in recent technologies, as it impacts battery lifetime.
- CMOS technology has been extremely power-efficient when transistors are not switching or in stand-by mode, and system designers expect low leakage from CMOS chips.
- To meet leakage power constraints, multiple-threshold and variable threshold circuit techniques are often used.
- In multiple-threshold CMOS, the process provides two different threshold transitors. Low-threshold are employed on speed-critical sub-circuits and ther are fast and leaky.
- High-threshold transistors are slower but exhibit low sub-threshold leakage, and they are employed in noncritical/slow paths of the chip.
- As more transistors become timing-critical multiple-threshold techniques tend to lose effectiveness.

Variable Body Biasing:

- Variable-threshold circuits dynamically control the threshold voltage of transistors through substrate biasing and hence overcome shortcoming associated with multi-threshold design.
- When a variable-threshold circuit is in standby, the substrate of NMOS transistors is negatively biased, and their threshold increases because of the body-bias effect.
- Similarly the substrate of PMOS transistors is biased by positive body bias to increase their Vt in stand-by. Variable-threshold circuits can, in principle, solve the quiescent/static leakage problem, but they require control circuits that modulate substrate voltage in stand-by.
- Fast and accurate body-bias control with control circuit is quite challenging, and requires carefully designed closed-loop control.
- When the circuit is in standby mode the bulk/body of both PMOS and NMOS are biased by third supply voltage to increase the Vt of the MOSFET as shown in the Figure.
- However during normal operation they are switched back to reduce the Vt.

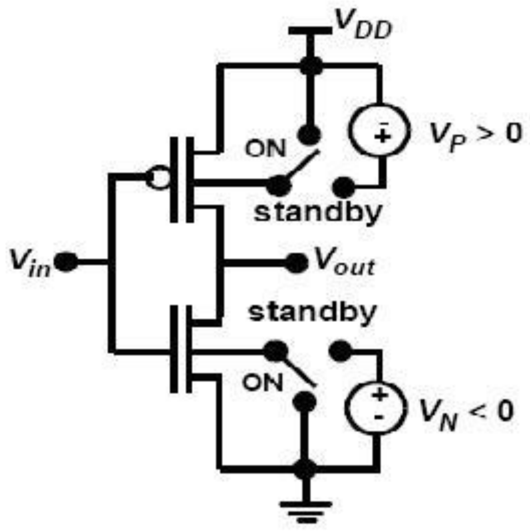


Figure: Variable Body Biasing

Sleep Transistors:

- Sleep Transistors are High Vt transistors connected in series with low Vt logic as shown below .
- When the main circuit consisting of Low Vt devices are ON the sleep transistors are also ON resulting in normal operation of the circuit.
- When the circuit is in Standby mode even High Vt transistors are OFF.
- Since High Vt devices appear in series with Low Vt circuit the leakage current is determined by High Vt devices and is very low.
- So the net static power dissipation is reduced.

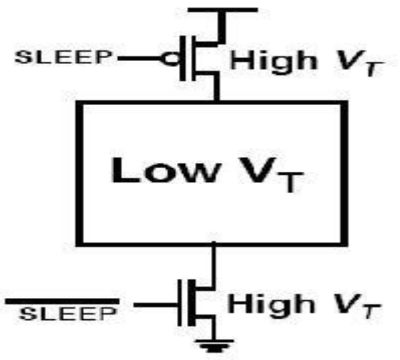


Figure10: Circuit Design with Sleep Transistors

Dynamic Threshold MOS:

- In dynamic threshold CMOS (DTMOS), the threshold voltage is altered dynamically to suit the operating state of the circuit.
- A high threshold voltage in the standby mode gives low leakage current, while a low threshold voltage allows for higher current drives in the active mode of operation.
- Dynamic threshold CMOS can be achieved by tying the gate and body together.
- The supply voltage of DTMOS is limited by the diode built-in potential in bulk silicon technology.
- The PN diode between source and body should be reverse biased.

- Hence, this technique is only suitable for ultralow voltage (0.6V and below) circuits in bulk CMOS.

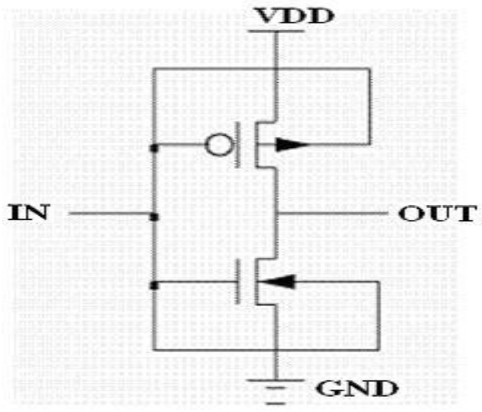


Figure11: DTMOS Circuit

Short Circuit Power Suppression

- Short-circuit power, is caused by the short circuit currents that arise when pairs of PMOS/NMOS transistors are conducting simultaneously.
- In static CMOS circuits, short-circuit path exists for direct current flow from VDD to ground, when VTn< Vin< VDD-|VTp|

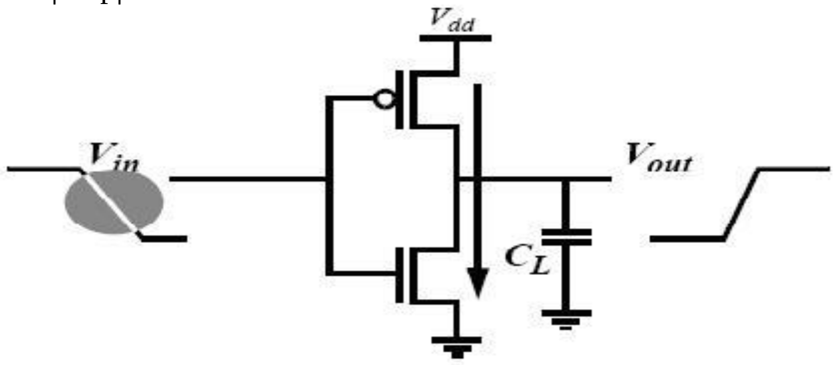


Figure12: Short Circuit Power in CMOS Circuits

- One way to reduce short circuit power is to keep the input and output rise/fall times the same. If Vdd < Vtn + |Vtp| then short-circuit power can be eliminated.
- If the load capacitance is very large, the output fall time is larger than the input rise time. The drain-source voltage of the PMOS transistor is 0.
- Hence the short-circuit power will be 0. If the load capacitance is very small, the output fall time is smaller than the input rise time.
- The drain-source voltage of the PMOS transistor is close to VDD during most of the transition period. Hence the short-circuit power will be very large.

UNIT – III

SEQUENTIAL LOGIC CIRCUITS

Static latches and Registers Dynamic latches and Registers, Pulse Registers, Sense Amplifier Based Register, Pipelining, Schmitt Trigger, Monostable Sequential Circuits, Astable Sequential Circuits.

Timing Issues: Timing Classification of Digital System, Synchronous Design.

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 bitable circuit. A bistab le 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 firs t 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.

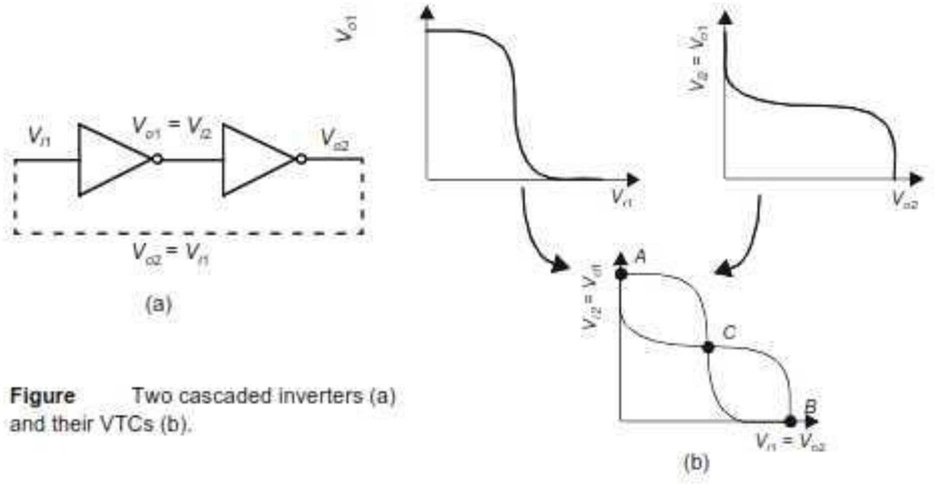
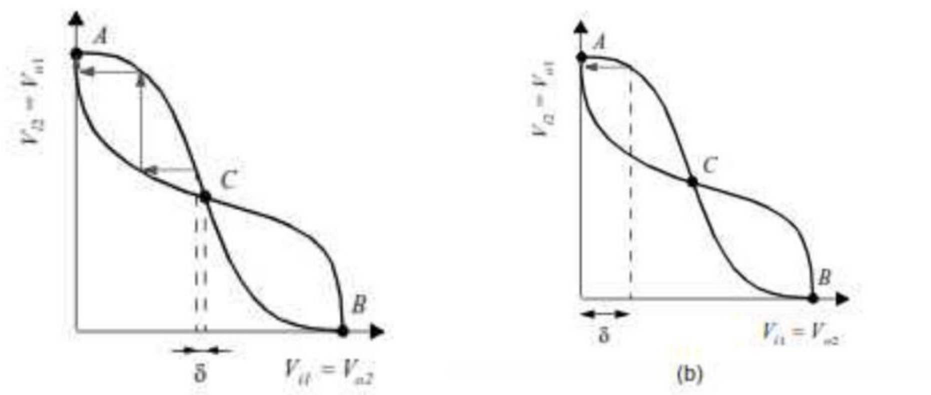


Figure 3.1: a. Two inverters connected in cascade b. VTCs

- Cross-coupled inverter pai r is biased at point C. It is amplified and regen erated 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.

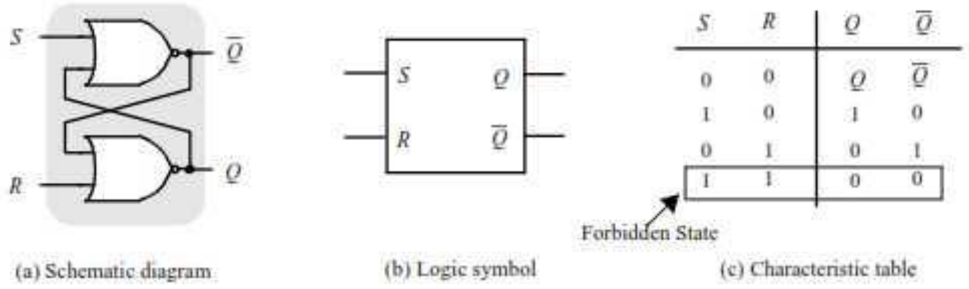


(a) Figure 3.2 Metastability.

- 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 NO R 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 force d 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

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 im plemented using a cross-coupled NAND structure as shown in Figure 3.4

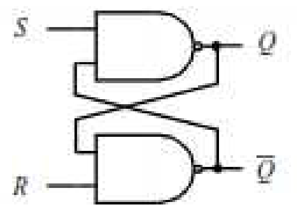


Figure 3.4 NAND based SR flip-flop

Clocked SR flip-flop:

- Clocked SR flip-flop (a level-sensitive positive latch) is shown in Figure 3 .5.
- It consists of a cross-coup led inverter pair, plus 4 extra transistors to drive the flip-flop from one state to another a nd 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 latc h switch, we must succeed in bringing Q below the switching threshold of the inverter M_1-M_2.
- Once this is achieved, th e 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.

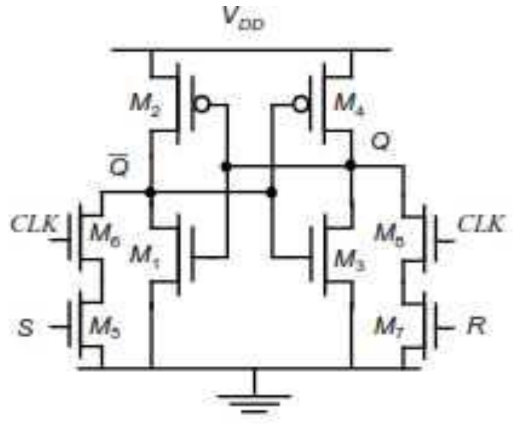


Figure 3.5 CMOS clocked SR flip-flop

- 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.

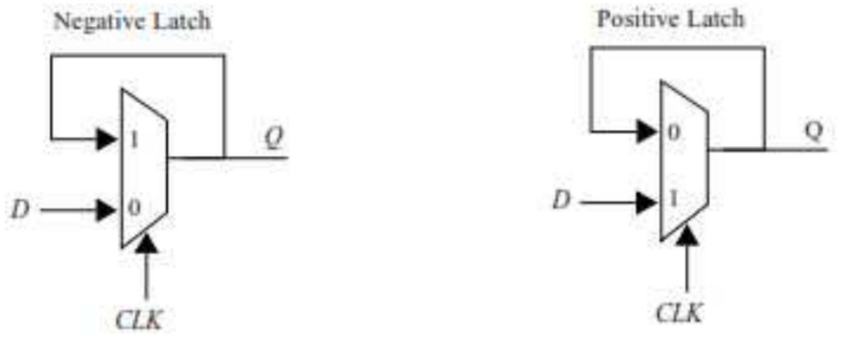


Figure 3.6 Negative and positive latches based on multiplexers.

❖ **Design a d-latch using transmission gate. (May 2015)**

- 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 feed back loop is open due to the top transmission gate is OFF.

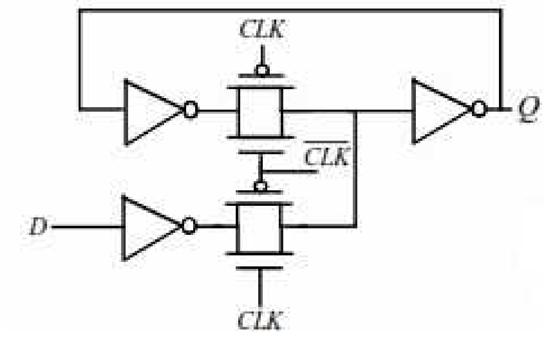


Figure 3.7 Transistor level implementation of a positive latch built using transmission gates.

- 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 NMO S 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.

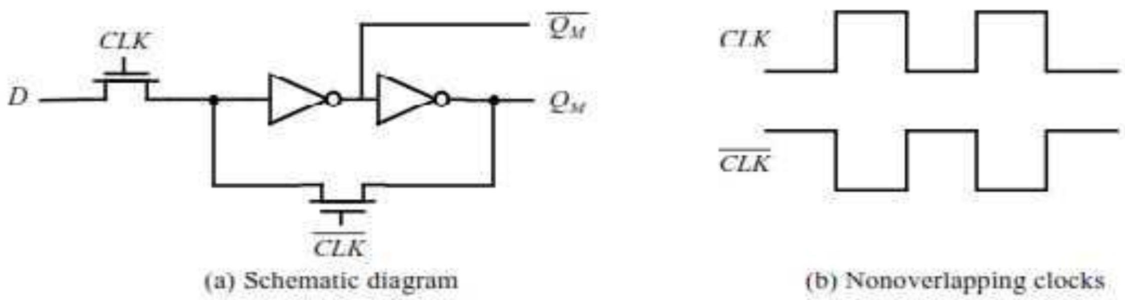


Figure 3.8 Multiplexer based NM OS latch using NMOS only pass transistors for multiplexers.

Master-Slave Based Edge Triggered Register:

- ❖ **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 trigge red register, explain its operation with overlapping periods. (April 2018, NOV 2018)**

- 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 stag e 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).

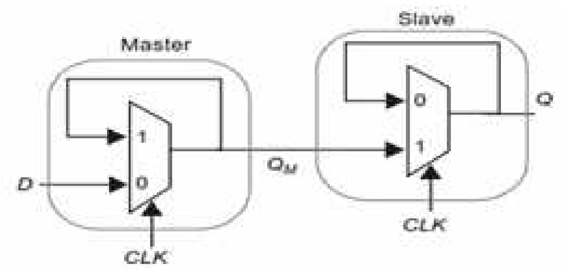


Figure 3.9: Positive edge-triggered register based on a master-slave configuration.

- 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.

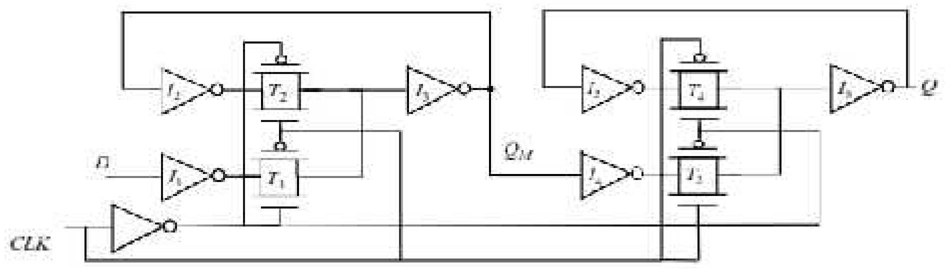


Figure 3.10 Transistor-level implementation of a master-slave positive edge-triggered register using multiplexers.

- 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 $CL\bar{K}$ is a perfect inversion of CLK.
- Even if this was possible, t his 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 NM OS-only negative master-slave register of Figure 3.11 a.

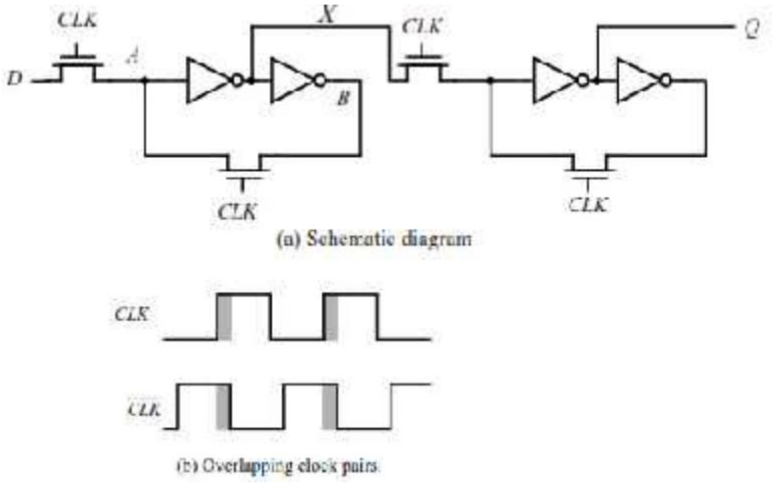


Figure 3.11 Master-slave register based on NMOS-only pass transistors.

- 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 $C\overline{LK}$.
- 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 tim e, 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 ***pseudost atic***: the register employs a combination of static and dynamic storage approaches depending upon the state of the clock.

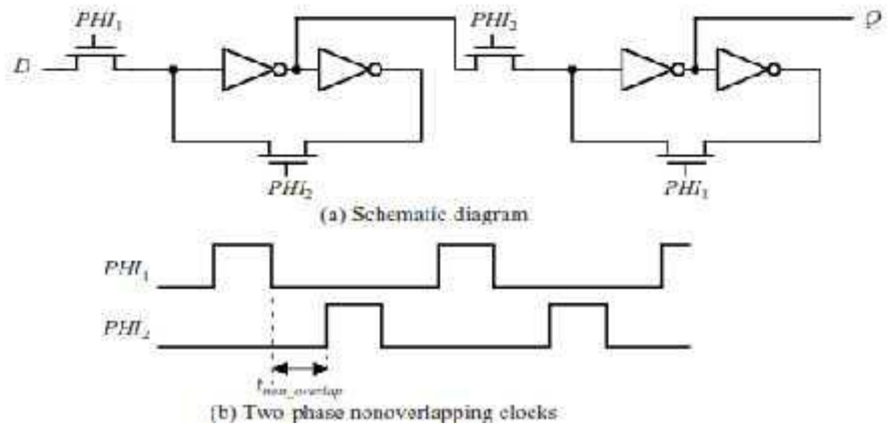


Figure 3.12 Pseudostatic two-phase D register.

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.
- Multiple Threshold devices as shown in Figure 3.13.
- 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.

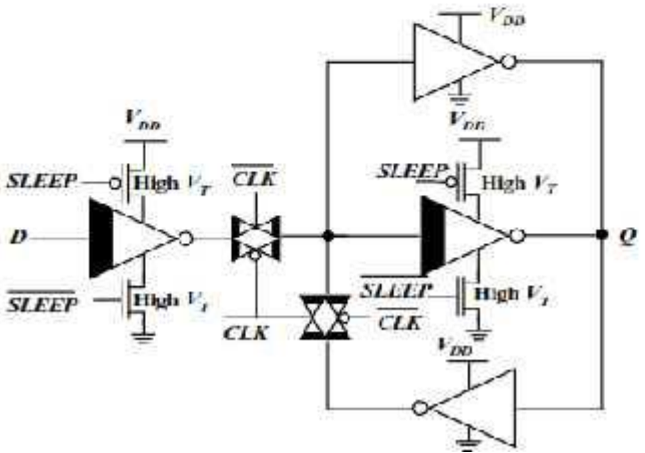


Figure 3.13: One solution for the leakage problem in low-voltage operation using MTCMOS.
************************* *************************************** **************

3.2 Dynamic Latches and Registers:

- ❖ **Discuss about the des ign 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 t he 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, bas ed 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 a s 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-G ate 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 h as 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.

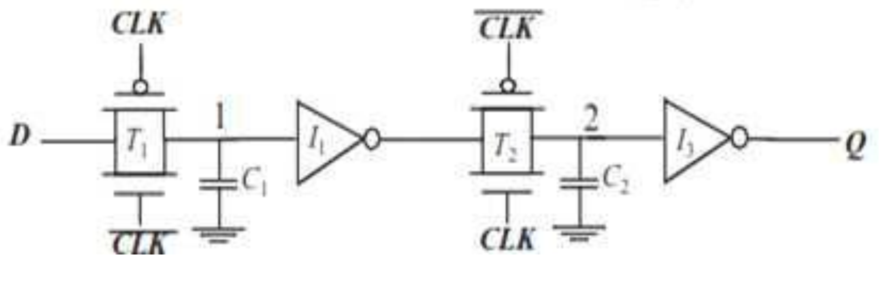


Figure 3.14 Dynamic edge-triggered register.

- 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: $t_{overlab\ 0-0} < t_{T1} + t_{I1} + t_{T2}$
- Similarly, the constraint for the 1-1 overlap is given as: $t_{hold} > t_{overlab1-1}$

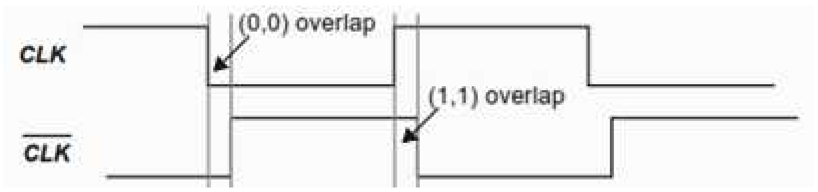


Figure 3.15 Impact of non-overlapping clocks

3.2.2 C^{2}MOS Dynamic Register:

The C^{2}MOS 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^{2}MOS (Clocked CMOS) register.

1. CLK = 0 ($\overline{CLK}$ = 1):
   - The first tri-state driver is turned ON. The master stage act s 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 h old mode.
   - Both transistors M_7 and M_8 are OFF, decoupling the output from the input. The output Q retains i ts 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 th e 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.

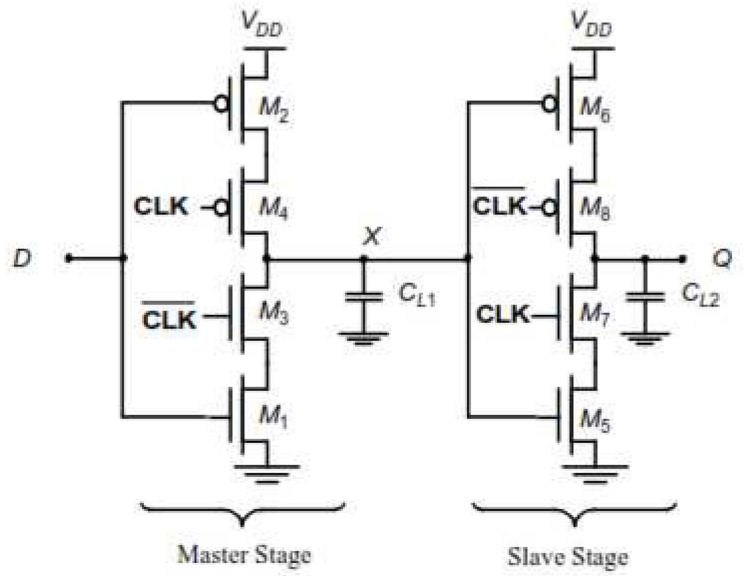


Figure 3.16 C^2MOS master-slave positive edge-triggered register.

- It is similar to the trans mission-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.*

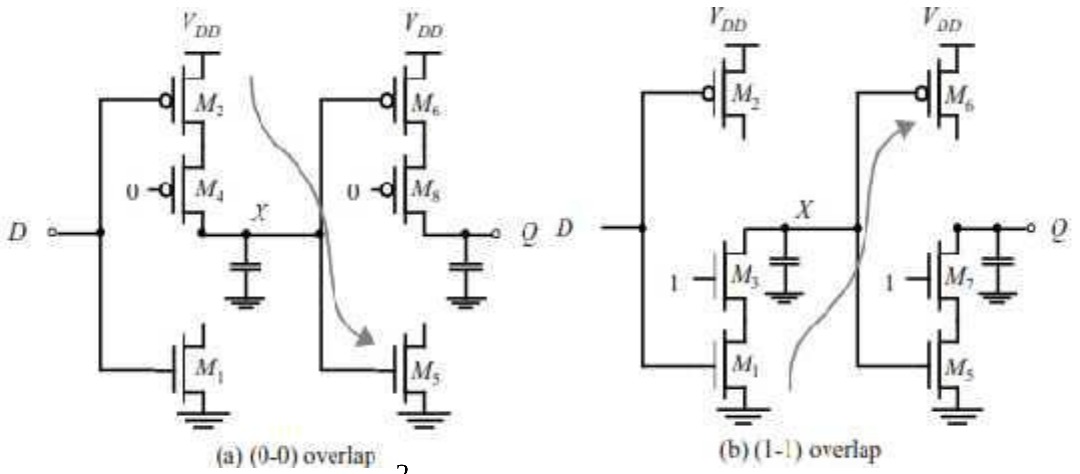


Figure 3.17 C^2MOS D FF during overlap periods.

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^2MOS 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, w hen 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.
- The main advantage is the use of a single clock phase.
- The disadvantage is, increase in the number of transistors (12 transistors are required).

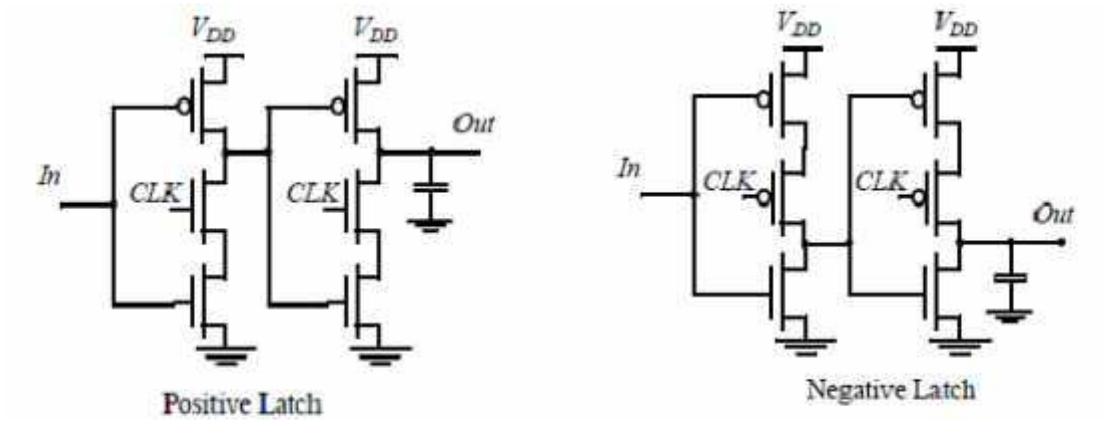


Figure 3.19 TSPC approach.

- 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 is reduced and clock load is reduced by half.

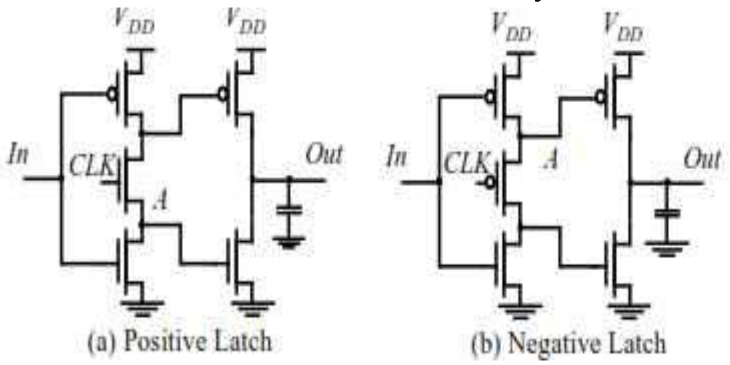


Figure 3.20 Simplified TSPC latch (also called split-outpu t).

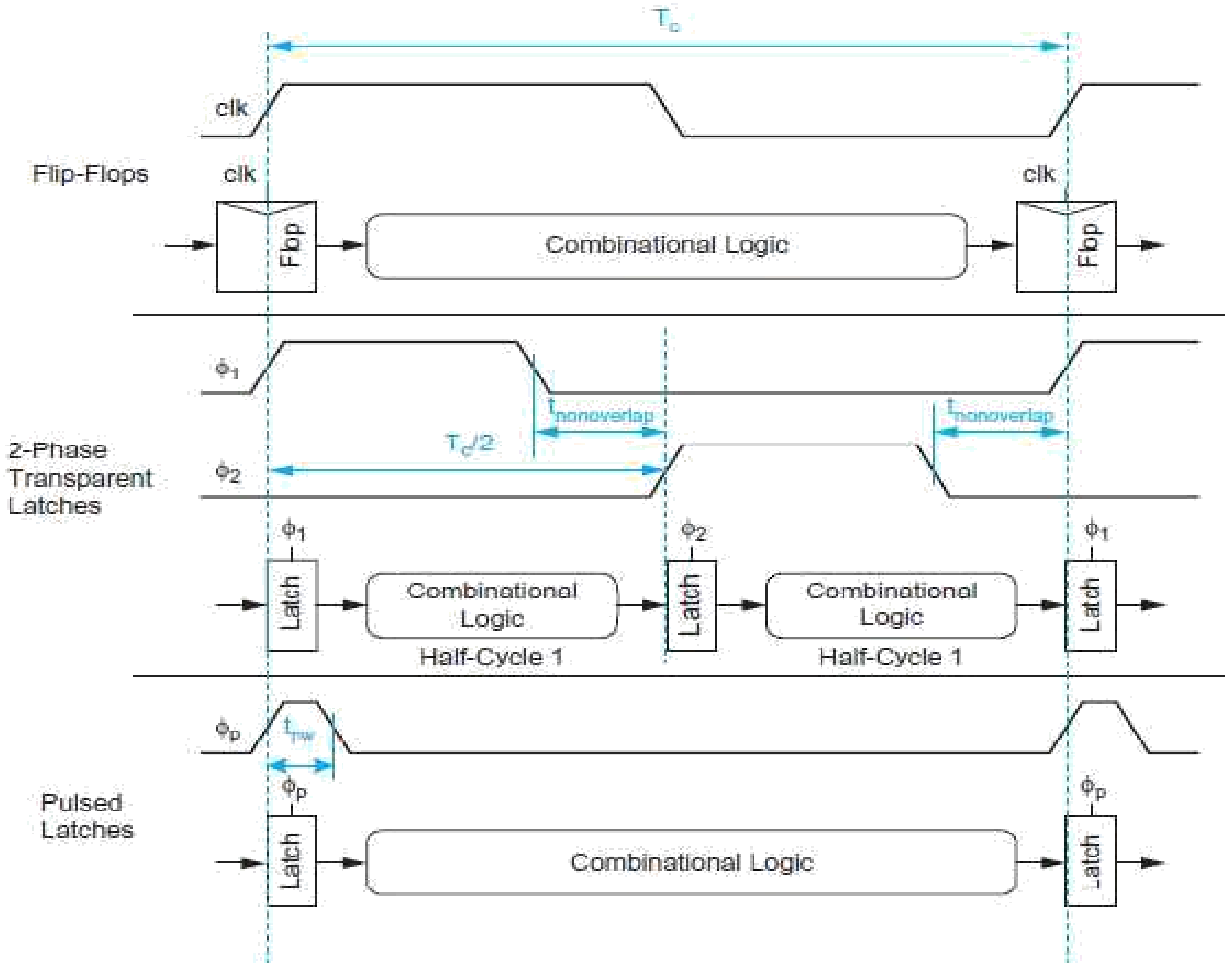
************************* ************************************** **************

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)**

(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 φ1 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.

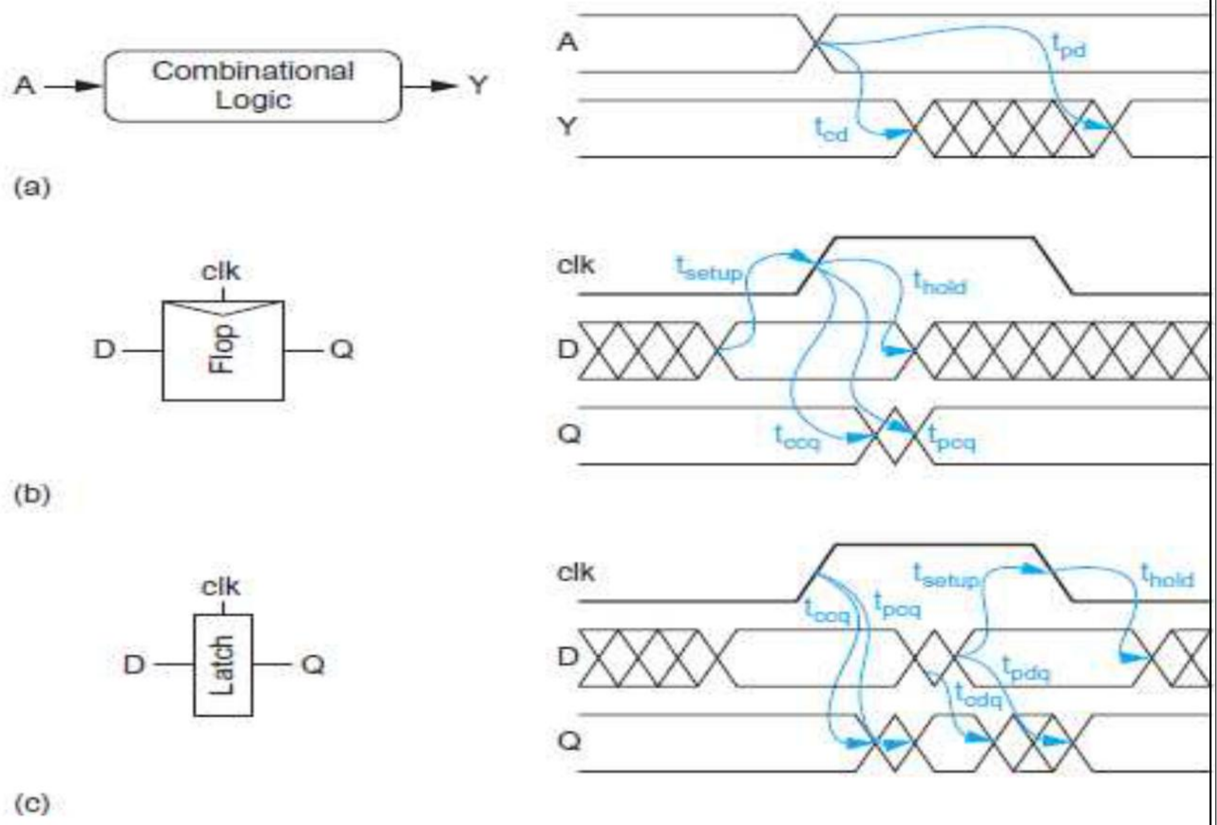
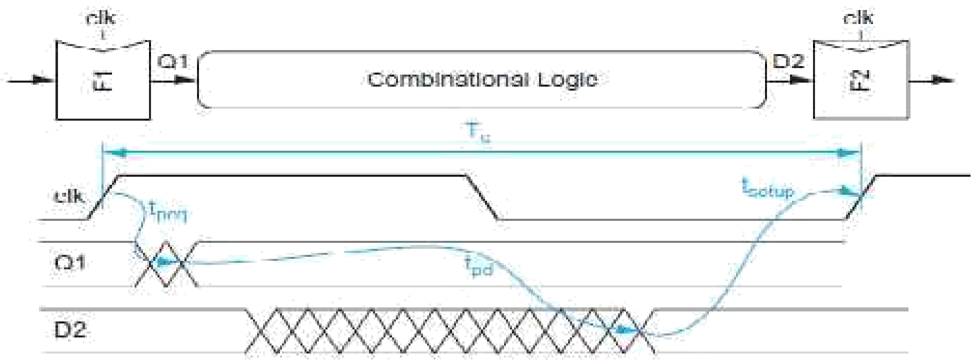


Figure: Timing diagrams

- 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 $T_c \geq t_{pcq} + t_{pd} + t_{setup}$

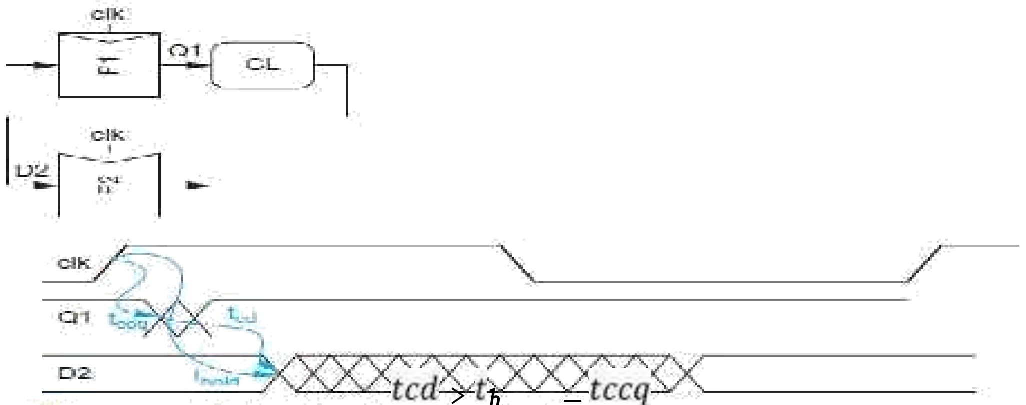
(or)

$T_{pd} \leq t_c - (t_{setup} + t_{pcq})$

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.

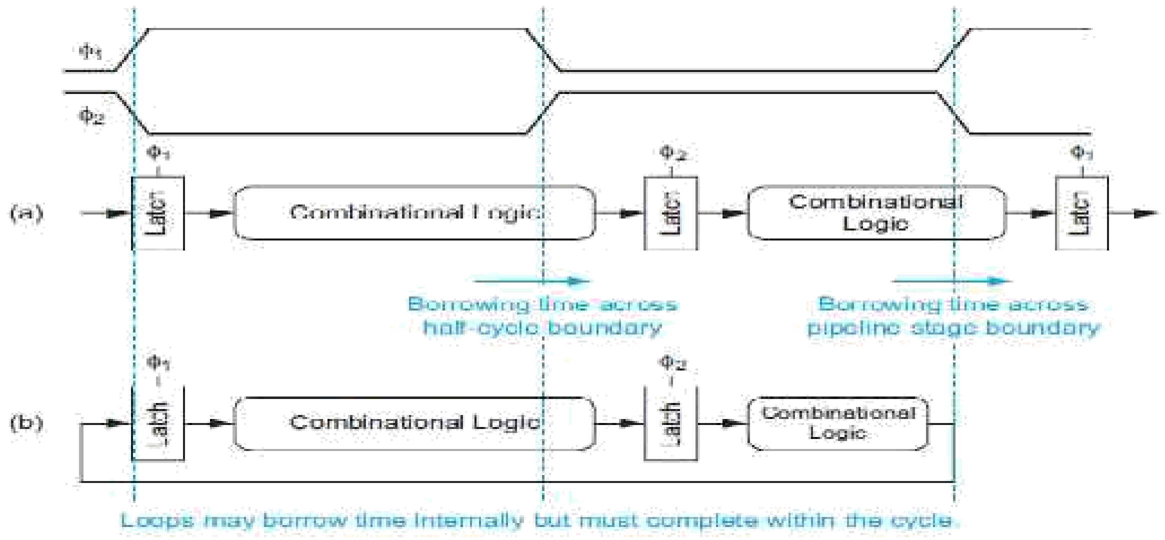


$t_{cd} \geq t_{hold} - t_{ccq}$

FIGURE Flip-flop latch min-delay constraint

(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.



ϕ_1
ϕ_2
(a)
Latch
Combinational Logic
Latch
Combinational Logic
Latch
Borrowing time across half-cycle boundary
Borrowing time across pipeline stage boundary
(b)
Combinational Logic
Combinational Logic
Loops may borrow time internally but must complete within the cycle

$$t_{borrow} \le \frac{T_c}{2} - (t_{setup} - t_{nonoverlap})$$

***(E)* Clock Skew**

<u>Analyze the impact of spatial variations of clock signal on edge-triggered sequential logic circuits. (NOV 2018)</u>

- 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.

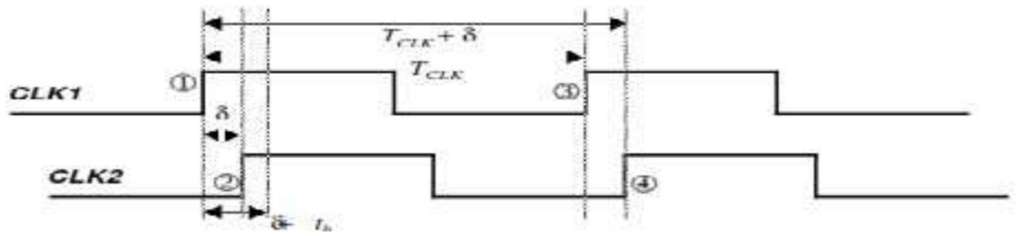


Figure: Timing diagram to study the impact of clock skew on p erformance and functionality. In this sample timing diagram, δ >0.

(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 uncertaintyy measure.
- Cycle-to-cycle jitter refers to time varying deviation of a single clock perio d.
- 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 periood 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.

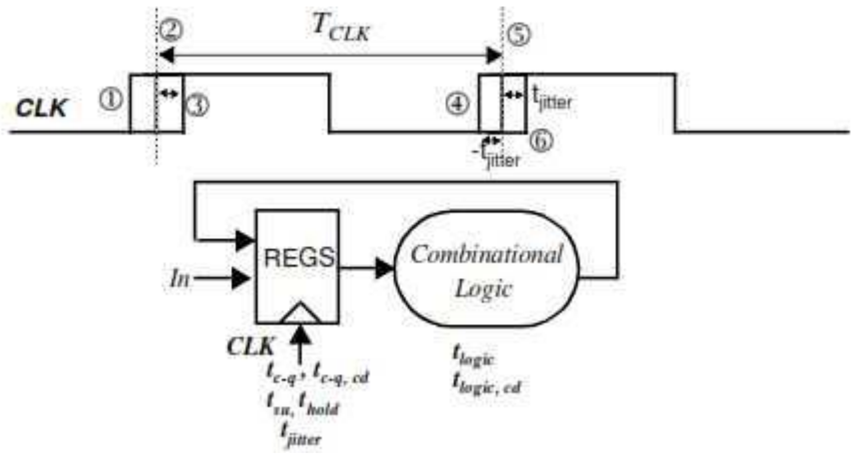


Figure: Circuit for studying the impact of jitter on performance.

************************* ** *********

3.4 Pipelining:

- ❖ **Explain in detail about pipelining structure needed for a logic operation. (A pril 2017, Nov 2017)**
- ❖ **Discuss in detail various pipeli ning approaches to optimize sequential circuits. (May 2013, 2016)**

- Pipelining is a design tech nique 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:

$$T_{min} = t_{c-q} + t_{pd,logic} + t_{su}$$

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, th e 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.

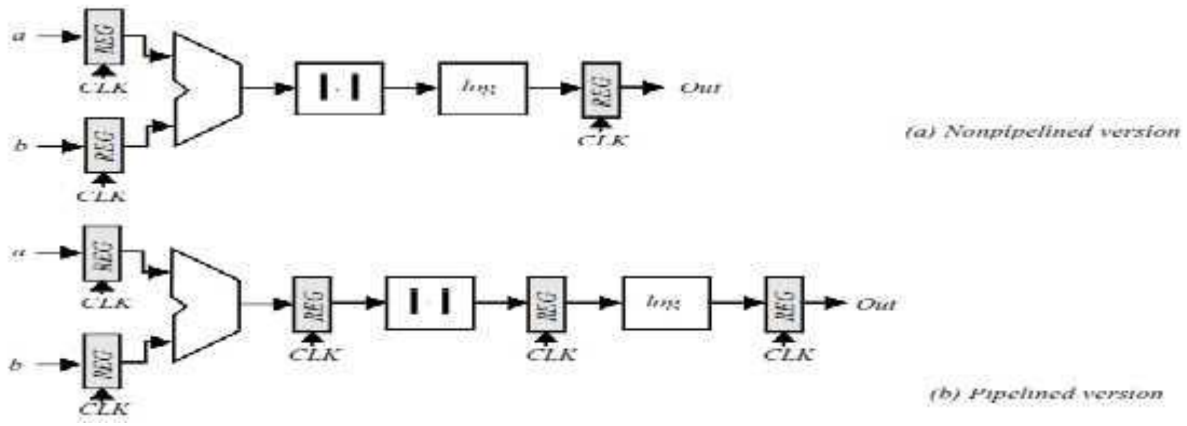


Figure 3.22 Data path for the computation of log(|a + b|).

- 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 dela y 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 reg ister 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: E xample 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–C MOS 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 usi ng static logic) between the latches are noninver ting.*

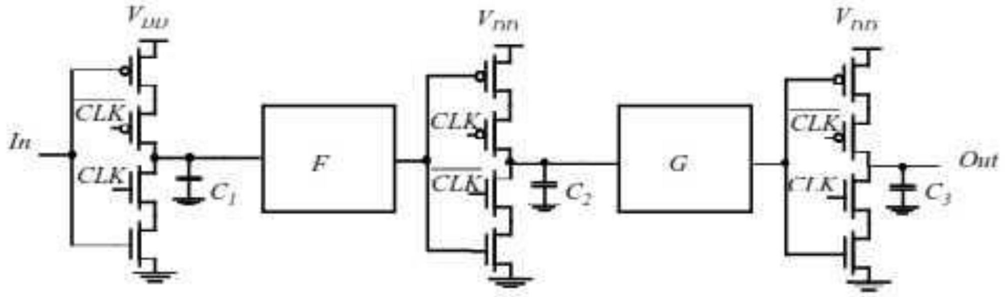


Figure 3.24 Pipe lined datapath using C^2MOS latches

- 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.

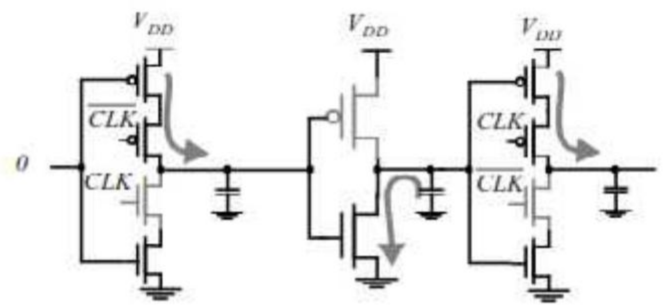


Figure 3.25 Potential race condition during (0-0) overlap in C^{2}MOS-based design.

- 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-s lave 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 *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.

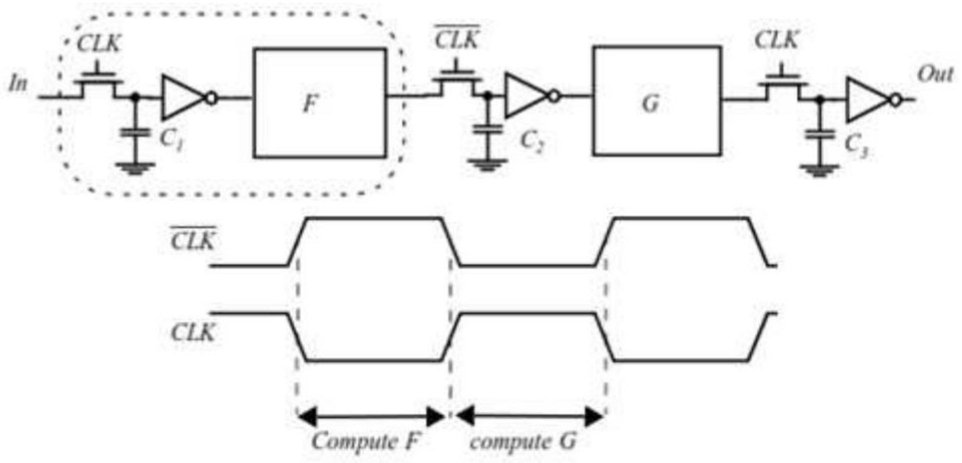


Figure 3.23 Operation of two-phase pipelined circuit using dynamic registers.

****

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.

φ

- 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.
****

3.6 Synchronous and Asynchronous circuits-Timing issues:

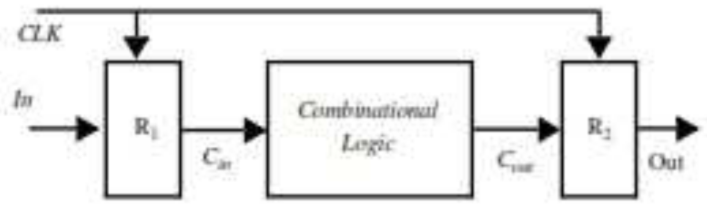
- A synchronous system approach is one, in which all memory elements in the system are simultaneously updated using a globally distributed periodic synchronization signal.
- Functionality is ensured by imposing some strict constraints on the generation of the clock signals and their distribution to the 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.

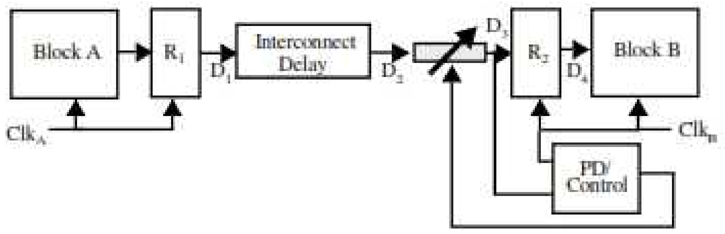
3.6.1 Synchronous Interconnect

- 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.



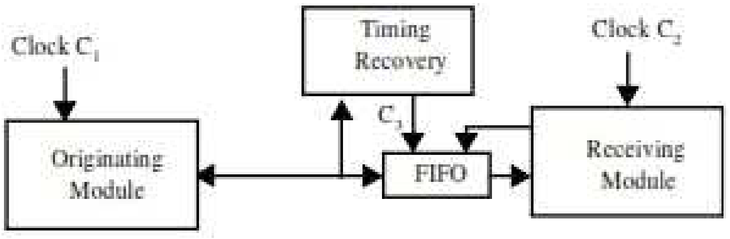
3.6.2 Mesochronous interconnect:

- A mesochronous signal has the same frequency but an unknown phase of fset 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

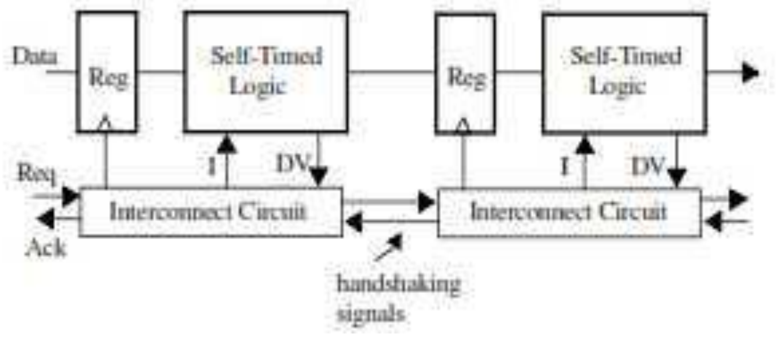
- A plesiochronous signal has nominally the same, but slightly different frequency as the local clock (“plesio” from Greek is near).
- This scenario can easily arise when two interacting modules have independent clocks generated from separate crystal oscillators.
- Since the transmitted sign al can arrive 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 sla ved to any local clock.

- As a result, it is not to map these arbitrary transitions into a synchronized d ata 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 m major issues in digital circuits and they limit the performance of a digital system.
- It is necessary to design a c lock 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 an n 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 balance d 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 t he clock can be distributed as a binary tree.

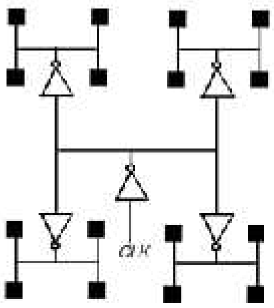


Figure: Example of an H-tree clock-distribution network for 16 leaf nodes.

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 CL K2 (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}.

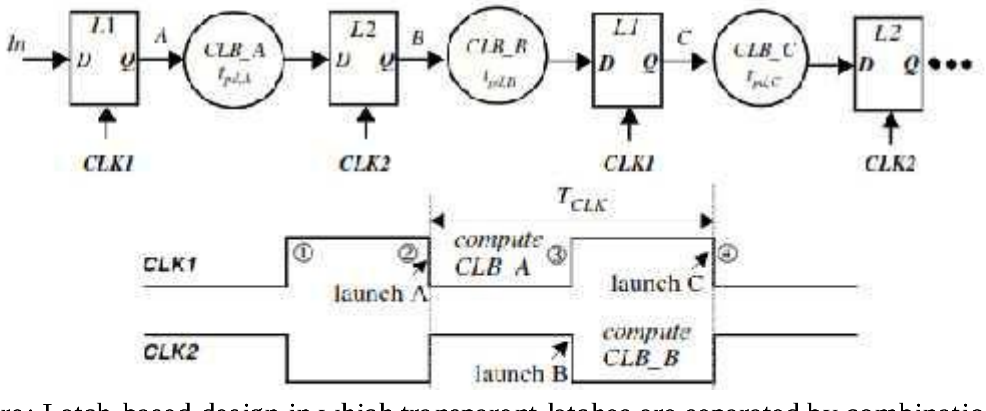


Figure: Latch-based design in which transparent latches are separated by combinational logic.

************************ ************************************ *************

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 Acknowledge and Req(uest) signal.

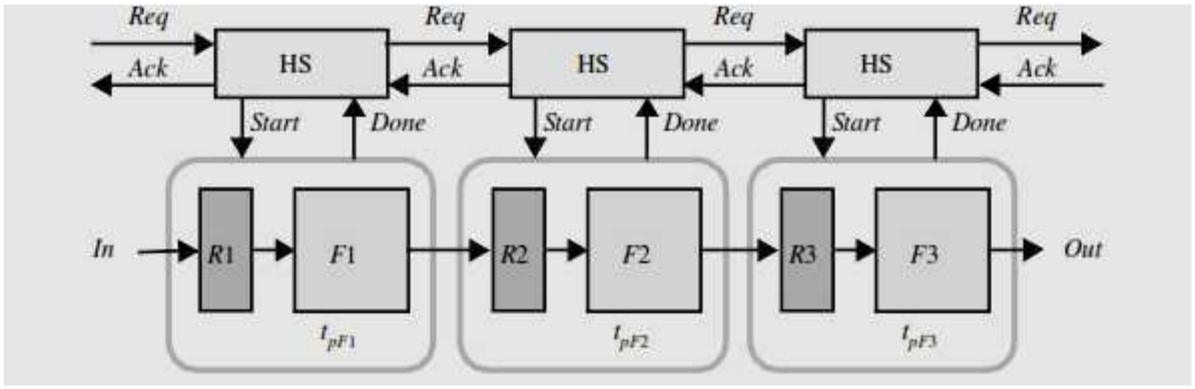


Figure: Self-timed, pipelined datapath.

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 F 2 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 met stability 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.

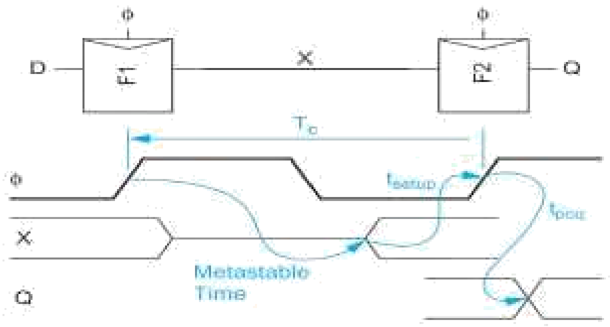


Figure: Simple Synchronizer

- Figure shows a simple synchronizer built from a pair of flip-flops. F1 samples the asynchronous input D.
- The output X may be detestable 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.

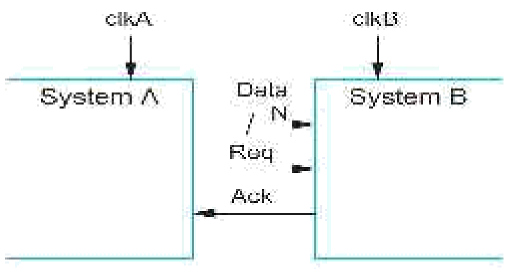


Figure: Communication between asynchronous systems

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.

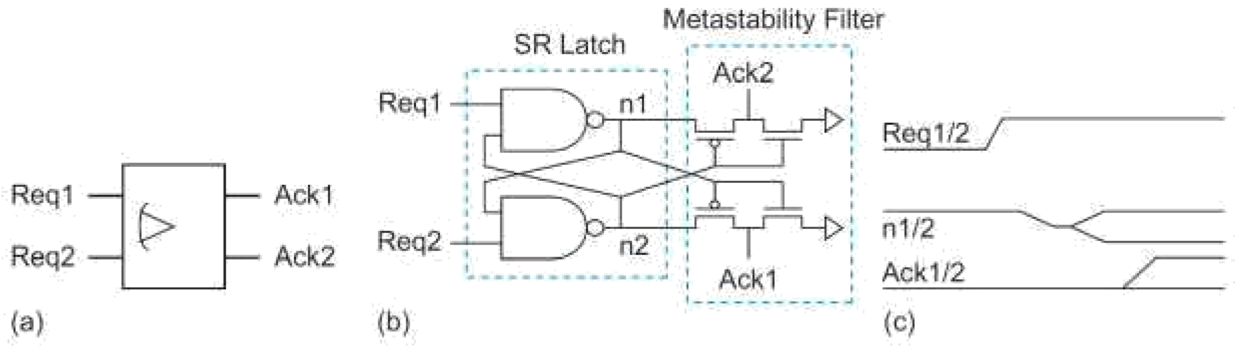


Figure: Arbiter

- 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.

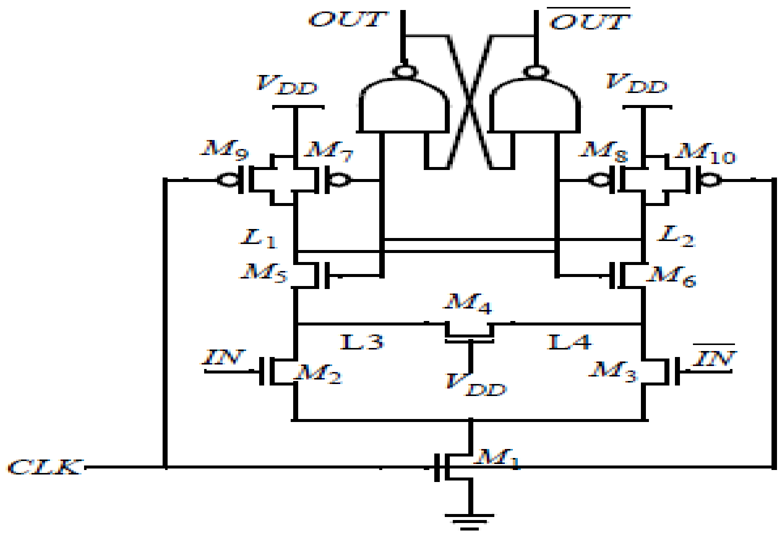
- The self-timed approach offers a potential solution to the growing clock-distribution problem.
- It translates the global clock signal into a number of local synchronization problems.
- Handshaking logic is needed to ensure the logical ordering of the circuit events and to avoid race conditions.

- 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.

****

Sense Amplifier Based Register:

A sense amplifier structure is used 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

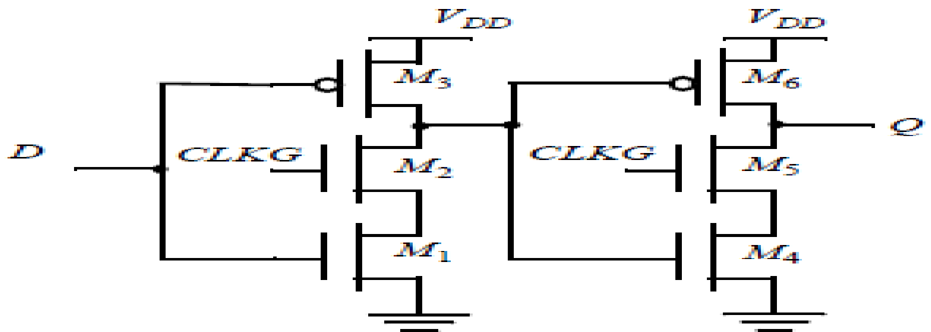
The circuit uses a precharged front-end amplifier that samples the differential input signal on the rising edge of the clock signal.

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.

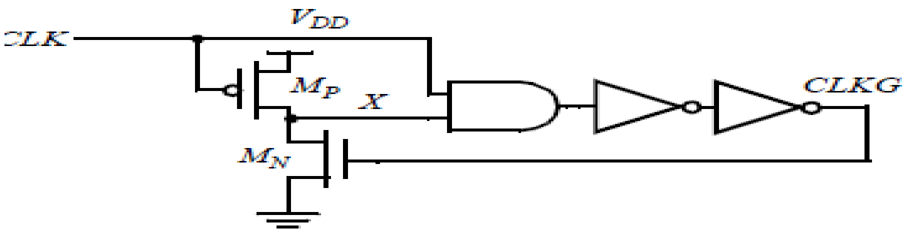
The differential inputs in this implementation don’t have to have rail-to-rail swing and hence this register can be used as a receiver for a reduced swing differential bus.

Pulse Registers:

- 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 is given below.

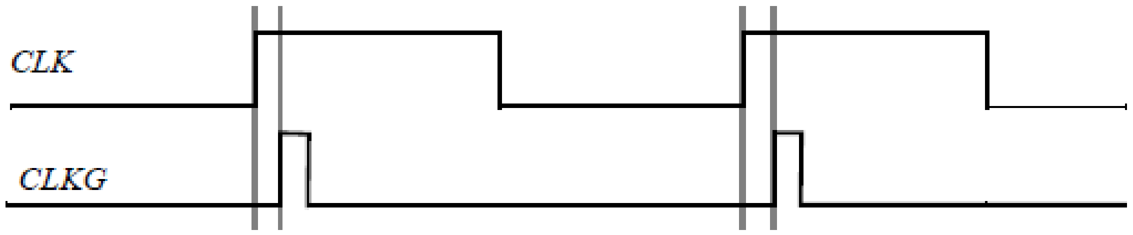


(a) register



(b) glitch generation

- This in turn activates MN, pulling X and eventually CLKG low.
- The length of the pulse is controlled by the delay of the AND gate and the two inverters.



(c) glitch clock

Figure 7.35 Glitch latch - timing generation and register.

- The advantage of the approach is the reduced clock load and the small number of transistors required.
- The glitch-generation circuitry can be amortized over multiple register bits.
- The disadvantage is a substantial increase in verification complexity.
- This has prevented a wide-spread use.

Non-Bistable Sequential Circuits

In the preceding sections, we have focused on one single type of sequential element, this is the latch (and its sibling the register). The most important property of such a circuit is that it has two stable states, and is hence called *bistable*. The *bistable* element is not the only sequential circuit of interest. Other regenerative circuits can be catalogued as *astable* and *monostable*. The former act as oscillators and can, for instance, be used for on-chip clock generation. The latter serve as pulse generators, also called *one-shot circuits*. Another interesting regenerative circuit is the Schmitt trigger. This component has the useful prop-erty of showing hysteresis in its dc characteristics—its switching threshold is variable and depends upon the direction of the transition (low-to-high or high-to-low). This peculiar feature can come in handy in noisy environments.

The Schmitt Trigger

Definition

A Schmitt trigger [Schmitt38] is a device with two important properties:

It responds to a slowly changing input waveform with a fast transition time at the output.

The voltage-transfer characteristic of the device displays different switching thresh-olds for positive- and negative-going input signals. This is demonstrated in Figure 7.46, where 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 VM+ and VM− , respectively. The hysteresis voltage is defined as the difference between the two.

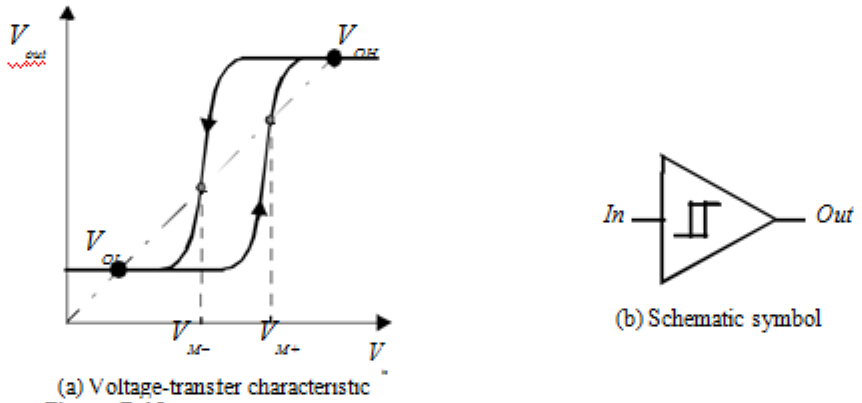


Figure 7.46 Non-inverting Schmitt trigger.

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. This is illustrated in Figure 7.47. 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.

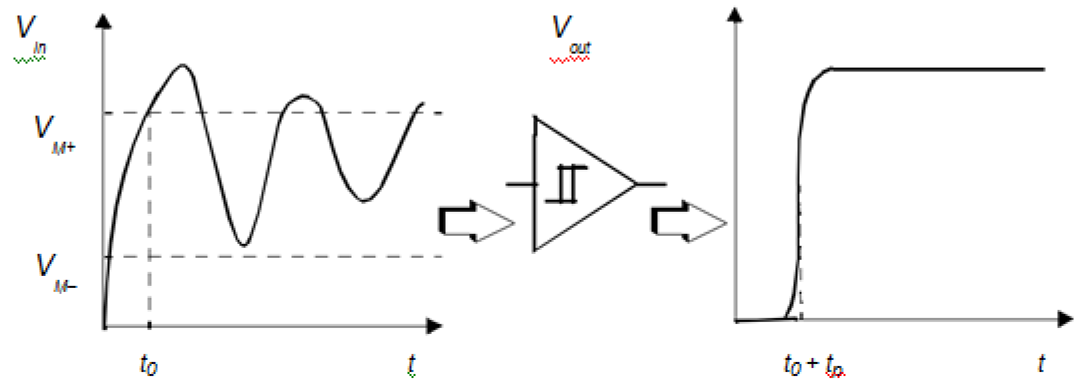


Figure 7.47 Noise suppression using a Schmitt trigger.

CMOS Implementation

One possible CMOS implementation of the Schmitt trigger is shown in Figure 7.48. The idea behind this circuit is that the switching threshold of a CMOS inverter is determined by the (k_n/k_p) ratio between the NMOS and PMOS transistors. Increasing the ratio results in a reduction of the threshold, while decreasing it results in an increase in $V_{M.}$
Adapting the ratio depending upon the direction of the transition results in a shift in the switching threshold and a hysteresis effect. This adaptation is achieved with the aid of feedback.

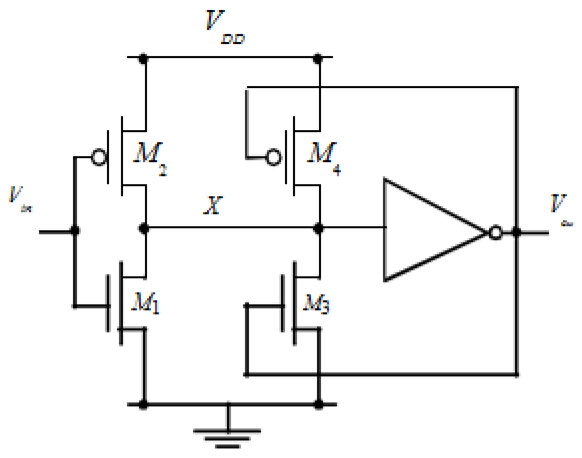


Figure 7.48 CMOS Schmitt trigger.

Suppose that Vin is initially equal to 0, so that Vout = 0 as well. The feedback loop biases the PMOS transistor M4 in the conductive mode while M3 is off. The input signal effectively connects to an inverter consisting of two PMOS transistors in parallel (M2 and M4) as a pull-up network, and a single NMOS transistor (M1) in the pull-down chain.
This modifies the effective transistor ratio of the inverter to kM1/(kM2+kM4), which moves the switching threshold upwards.

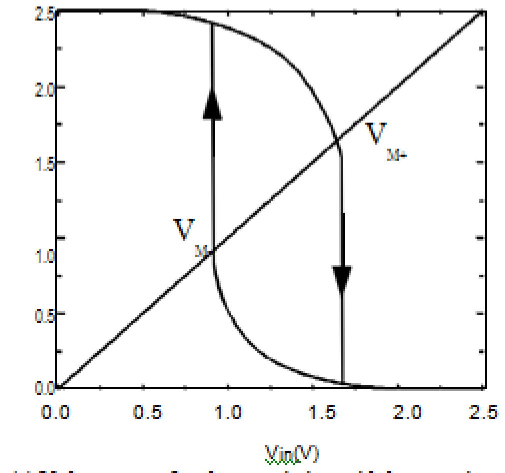
Once the inverter switches, the feedback loop turns off M4, and the NMOS device M3 is activated.
This extra pull-down device speeds up the transition and produces a clean output signal with steep slopes.

A similar behavior can be observed for the high-to-low transition. In this case, the pull-down network originally consists of M1 and M3 in parallel, while the pull-up network is formed by M2.
This reduces the value of the switching threshold to VM–.

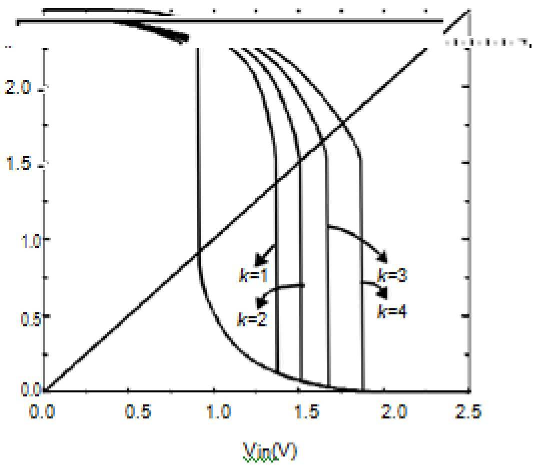
CMOS Schmitt Trigger

Consider the schmitt trigger with the following device sizes. Devices M_1 and M_2 are 1µ m/0.25µ m, and 3µ m/0.25µ m, respectively. The inverter is sized such that the switching threshold is around $V_{DD}/2$ (= 1.25 V). Figure 7.49a shows the simulation of the Schmitt trig-ger assuming that devices M_3 and M_4 are 0.5µ m/0.25µ m and 1.5µ m/0.25µ m, respectively. As apparent from the plot, the circuit exhibits hysteresis. The high-to-low switching point (V_{M-} = 0.9 V) is lower than $V_{DD}/2$, while the low-to-high switching threshold (V_{M+} = 1.6 V) is larger than $V_{DD}/2$.

It is possible to shift the switching point by changing the sizes of M_3 and M_4. For example, to modify the low-to-high transition, we need to vary the PMOS device. The high-to-low threshold is kept constant by keeping the device width of M_3 at 0.5 µ m. The device width of M_4 is varied as k * 0.5µ m. Figure 7.49b demonstrates how the switching threshold increases with raising values of k.



(a) Voltage-transfer characteristics with hysteresis.



(b) The effect of varying the ratio of the PMOS device M_4. The width is k * 0.5µ m.

Figure 7.49 Schmitt trigger simulations.

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 deter-mined 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.

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 sim-ple delay element to control the duration of the pulse.

The concept is illustrated in Figure 7.51. 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 t_d is created. The delay circuit can be realized in many different ways, such as an *RC*-network or a chain of basic gates.

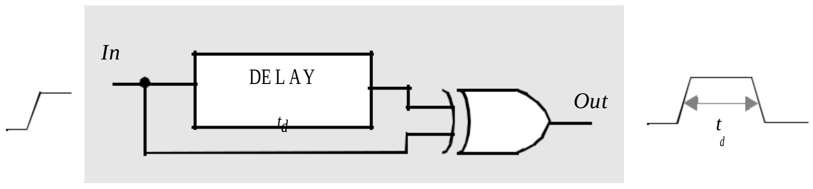


Figure 7.51 Transition-triggered one-shot.

Astable multivibrator 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 $2 \times t_p \times N$, with N the number of inverters in the chain and t_p the *propagation delay* of each inverter.
The simulated response of a ring oscillator with five stages is shown in Figure 7.52 (all gates use minimum-size devices). The observed oscillation period approximately equals 0.5 nsec,which corresponds to a gate *propagation delay* of 50 psec. By tapping the chain at various points, different phases of the oscillating waveform are obtained (phases 1, 3, and 5 are discussed.

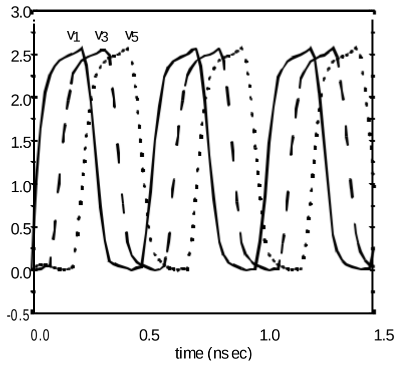


Figure 7.52 Simulated waveforms of five-stage ring oscillator. The outputs of stages 1, 3, and 5 are shown.

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 7.53 [Jeong87]. The mechanism for controlling the delay of each inverter is to limit the current available to discharge the load capacitance of the gate.

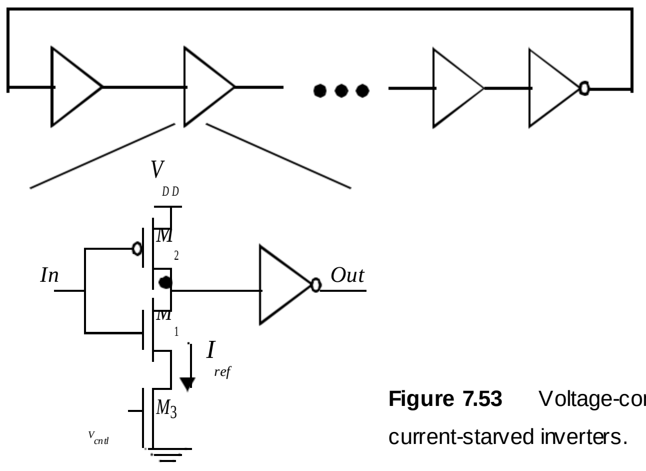


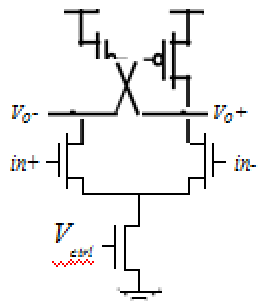
Figure 7.53 Voltage-controlled oscillator based on current-starved inverters.

In this modified inverter circuit, the maximal discharge current of the inverter is lim-ited 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 M_2.

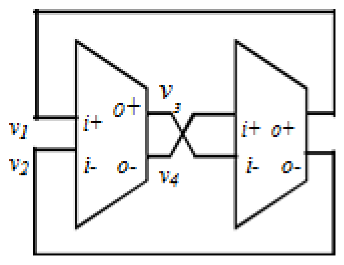
The added NMOS transistor M_3, is controlled by an analog control voltage V_{cntl}, which determines the avail-able discharge current. Lowering V_{cntl} reduces the discharge current and, hence, increases t_{pHL}.

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.

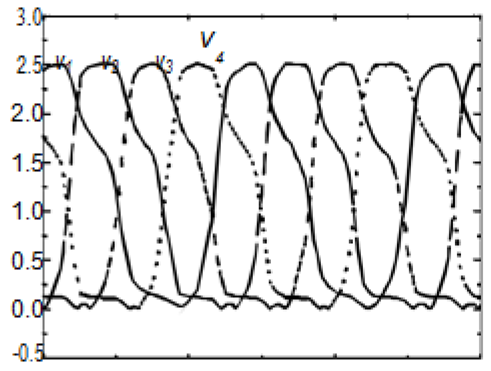
This is resolved by feeding its output into a CMOS inverter or better yet a Schmitt trigger. An extra inverter is needed at the end to ensure that the structure oscillates.



(a) delay cell



(b) two stage VCO



UNIT – IV
DESIGNING ARCHITECTURE BUILDING BLOCKS

Arithmetic Building Blocks: Data Paths, Adders, Multipliers, Shifters, ALUs, power and speedtradeoffs, Case Study: Design as a tradeoff.

Designing Memory and Array structures: Memory Architectures and Building Blocks, Memory Core, Memory Peripheral Circuitry.

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 data path is the core of processors, where all computations are performed.
- It is generally defined with general digital processor. It is shown in figure.

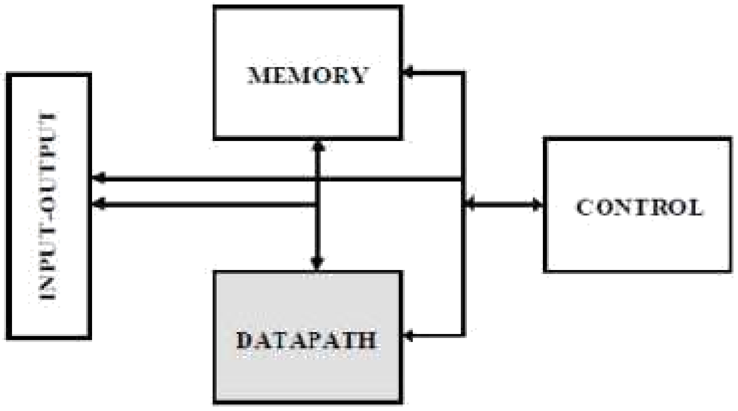
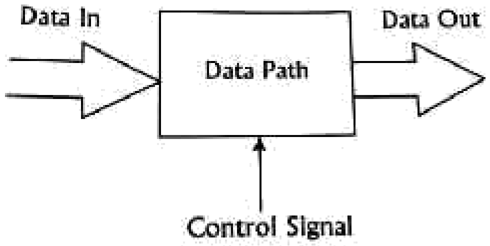
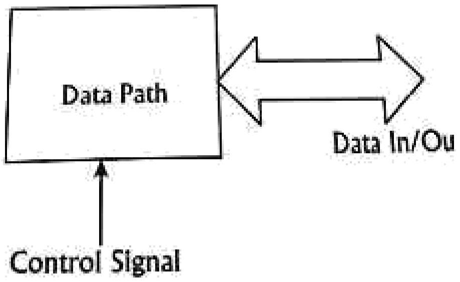


Figure: General digital processor

- 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.

- Data paths 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 data path consists of the number of bit slices (equal to the word length), each operating on a single bit. Hence the term is ***bit-sliced.***

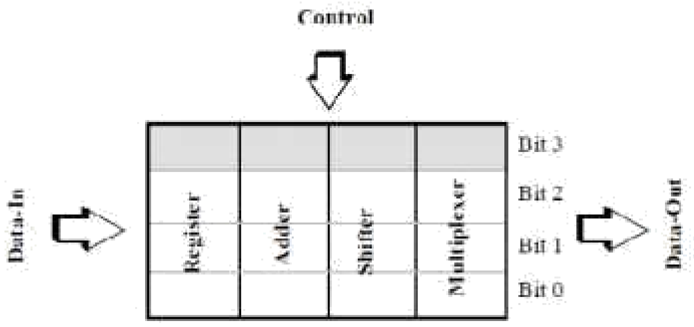


Figure: Bit-sliced datapath organization

****

II. 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)**

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.

$$C_{i+1} = a_i\, b_i + c_i\,.(a_i + b_i)$$
$$\overline{S}_i = (a_i + b_i + \overline{c}_i)c_i + (a_i\,.b_i\,.c_i)$$

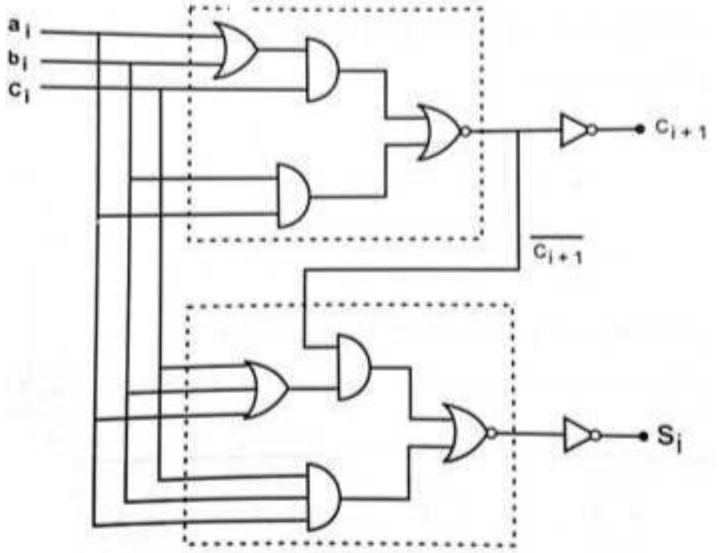


Figure: AOI Full adder

- 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.

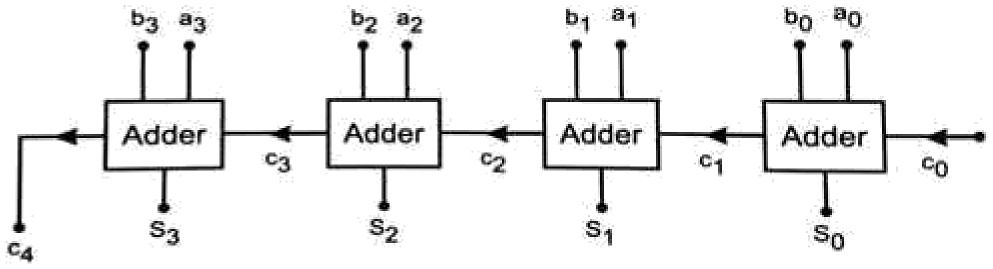


Figure: Ripple carry adder

- 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 i^{th} stage. We can calculate the total delay using the following equation

$$t_{4b} = t_{d3}+t_{d2}+t_{d1}+t_{d0} \text{ and } t_{d0} = t_d (a_0, b_0 \rightarrow c_1)$$

- This is the time for the input to produce the carry out bit.

$$t_{d1} = t_{d2} = t_d (c_{in} \rightarrow c_{out})$$

$$t_{d3} = t_d (c_{in} \rightarrow S_3)$$

$$t_{4b} = t_d (c_{in} \rightarrow S_3) +2t_d (c_{in} \rightarrow c_{out}) + t_d (a_0, b_0 \rightarrow c_1)$$

- If it is extend to n-bit, then the worst case delay is

- 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.

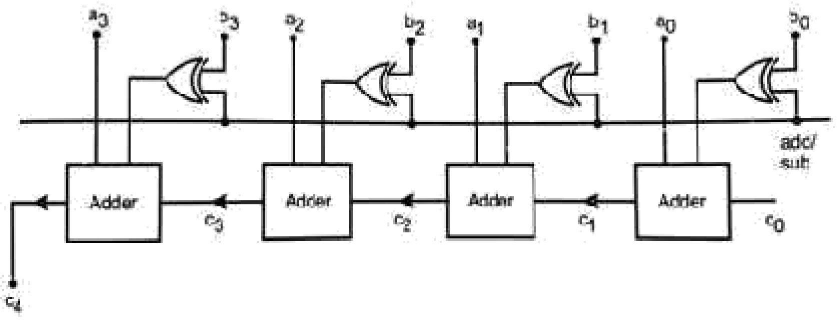


Figure: 4-bit adder/subtractor circuit

- Sum and carry expressions are designed using static CMOS.
- It requires 28 transistors∓(+ which+ lead) large area and circuit+ is slow+.

  Sum, S= $ABC_i\bar{C}_0ABC_i$ and Carry, C_0= $ABBC_iC_iA$

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.

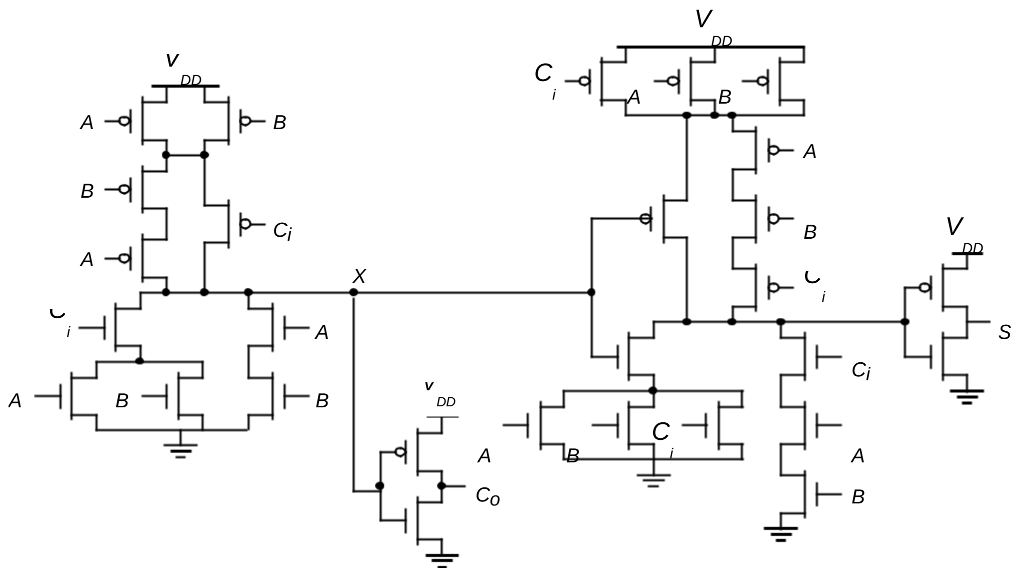


Figure: Complimentary Static CMOS Full Adder

****

III. Carry Look Ahead Adder (CLA):

- ❖ **Explain the operation and design of Carry lookahead adder (CLA). (May 2017, Nov 2016)**
- ❖ **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)**

- 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 to determine 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:

$$\boldsymbol{G(A,B) = A \cdot B}$$

- *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:

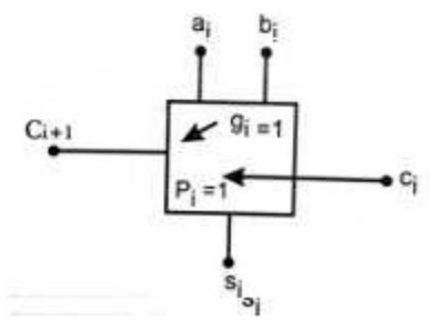
$$P(A,B) = A \oplus B$$

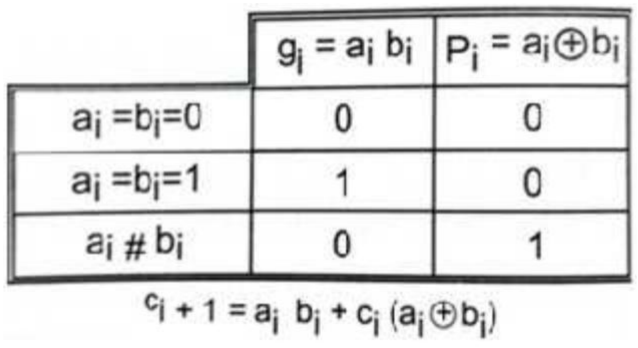
- These adders are used to overcome the latency which is introduced by the rippling effect of carry bits.
- Write carry look-ahead expressions in terms of the generate g_i and propagate p_i signals. The general form of carry signal c_i thus becomes

$$c_{i+1} = a_i . b_i + c_i .(a_i \oplus b_i) = g_i + c_i . p_i$$

- 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,

$$S_i = a_i \oplus b_i$$
$$c_1 = g_0 + p_0 \cdot c_0$$
$$c_2 = g_1 + p_1 \cdot c_1 = g_1 + p_1 \cdot (g_0 + p_0 \cdot c_0)$$
$$c_3 = g_2 + p_2 \cdot c_2$$
$$c_4 = g_3 + p_3 \cdot c_3 = g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0 + p_3 \cdot p_2 \cdot p_1 \cdot p_0 \cdot c_0$$





	$g_i = a_i b_i$	$P_i = a_i \oplus b_i$
$a_i = b_i = 0$	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

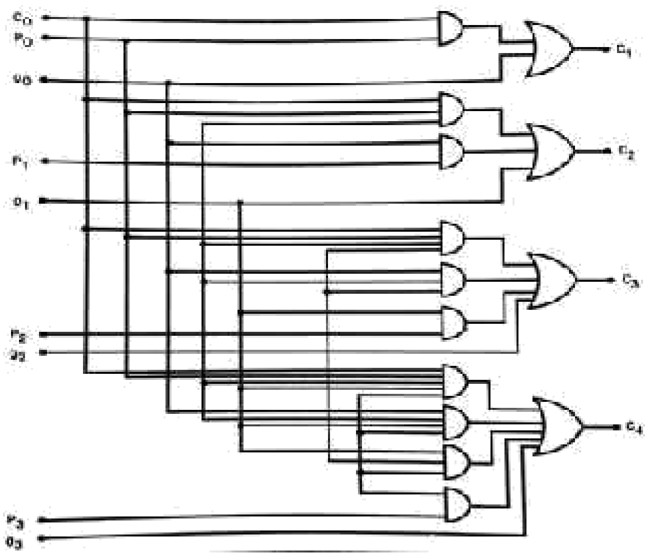


Figure – Logic network for 4-bit CLA carry bits

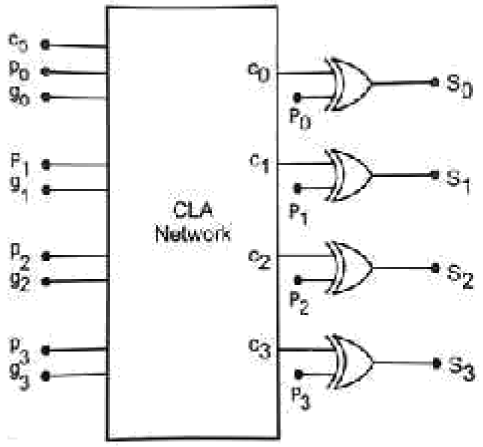


Figure – Sum calculation using the CLA network

- The symmetry in the array is shown in mirror. It allows more structured layout at the physical design level.

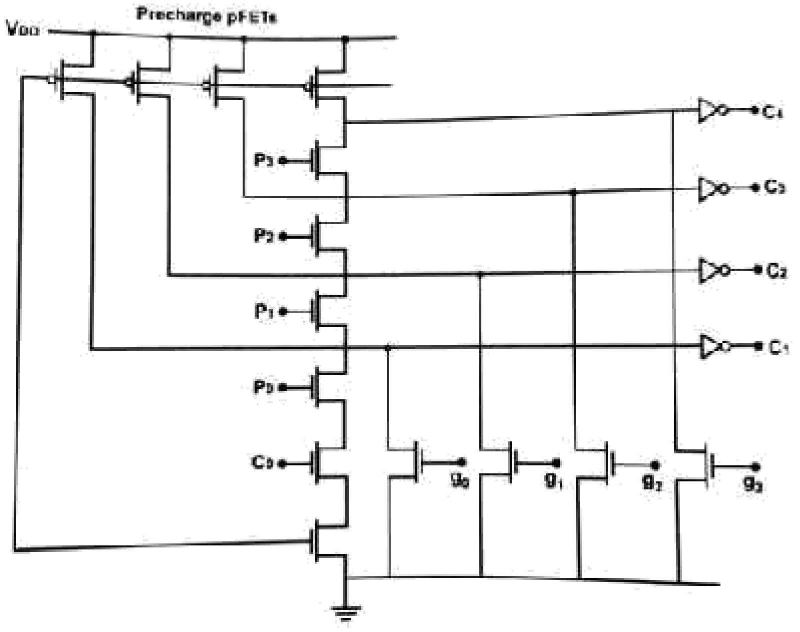


Figure – MODL carry circuit

- 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.

****

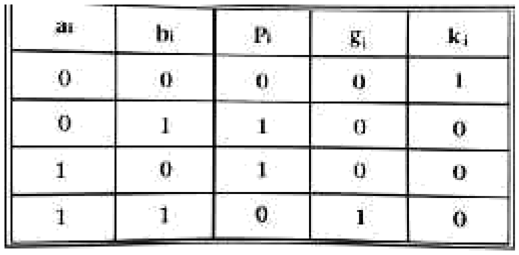
IV. Manchester Carry Chain Adder:

Discuss about 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.
- 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 drawbacks of 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 $c_{i+1} = g_i + c_i \cdot p_i$

  Where $p_i = a_i \oplus b_i$ and $g_i = a_i . b_i$

- Carry kill bit $k_i = \overline{a_i + b_i} = \bar{a}_i . \bar{b}_i$
- 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

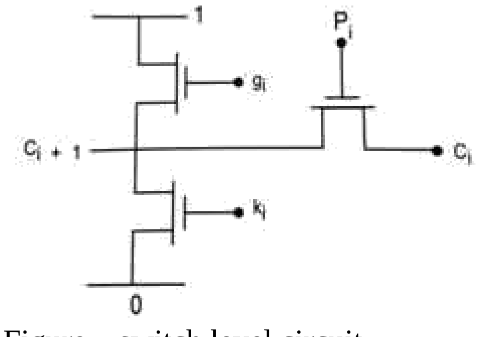
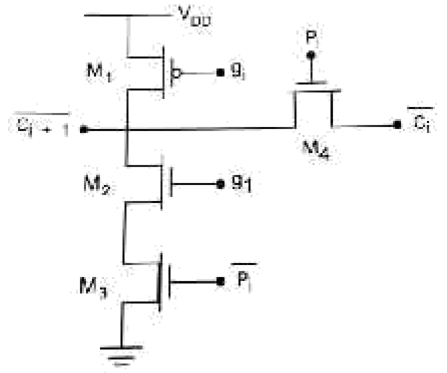


Figure – switch level circuit

- In the circuit shown below $\overline{C_i}$ 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.

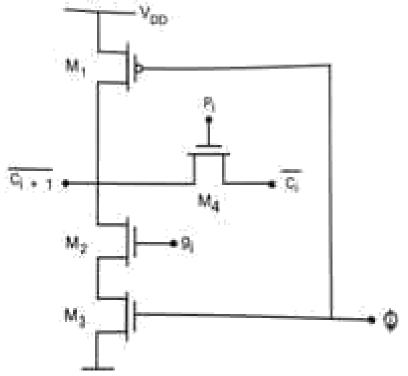
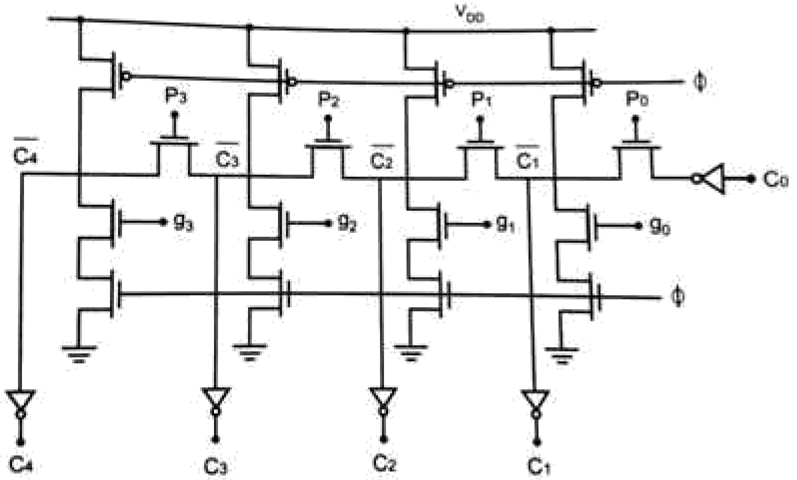


Figure dynamic circuit

- 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



****

V. 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)

- 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).

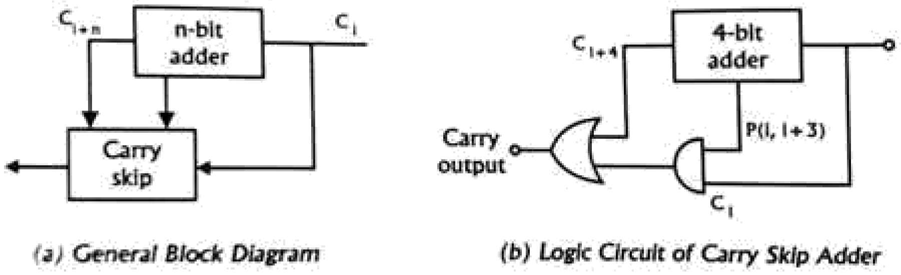


Figure: Carry Skip Adder.

- 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”.

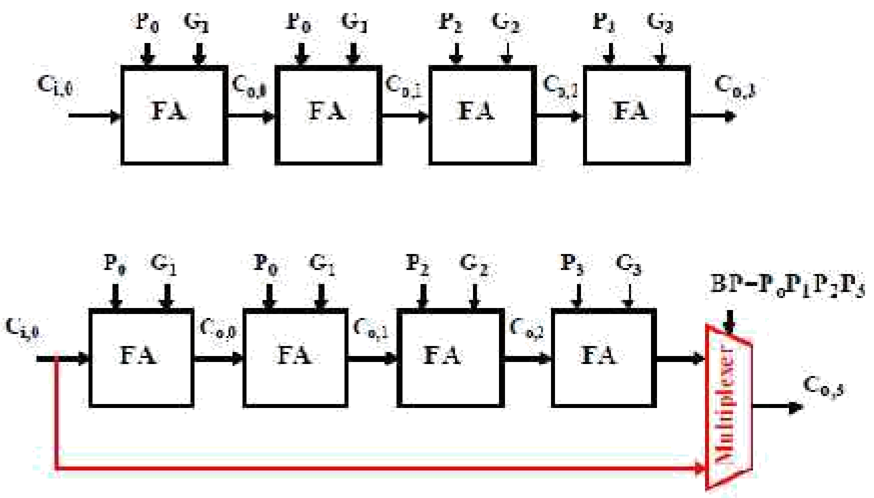
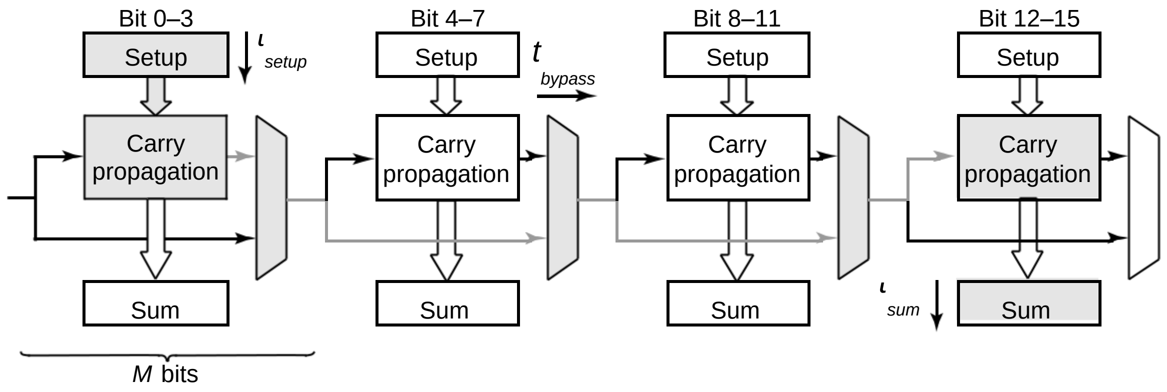


Figure: (a) Carry propagation (b) Adding a bypass

- 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*

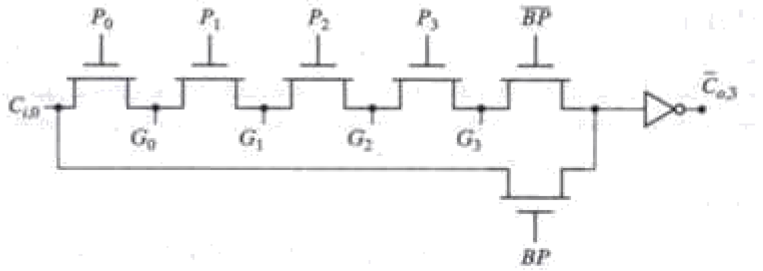
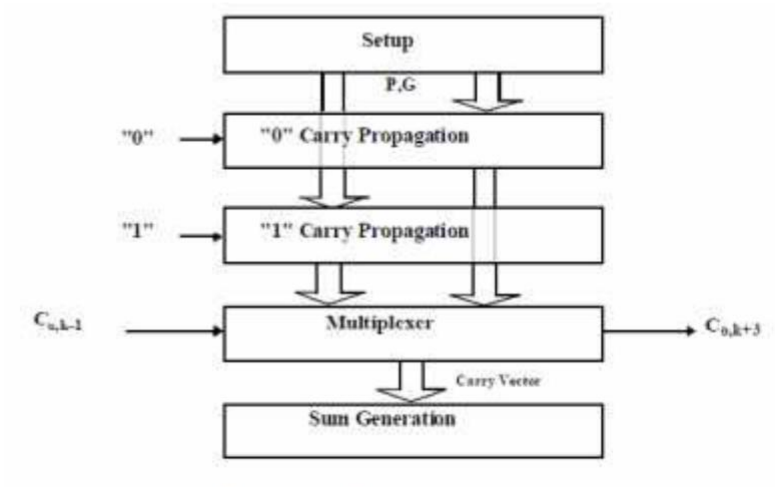


Figure: Manchester carry-chain implementation of bypass adder

(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√square2 root of the number of bits being added.
- Propagation delay, P is equal to $\sqrt{N}$ 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.

- 

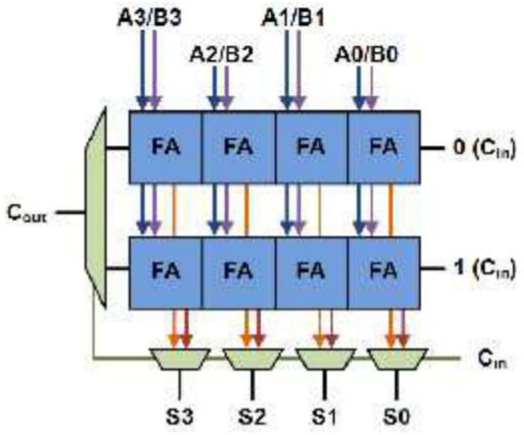
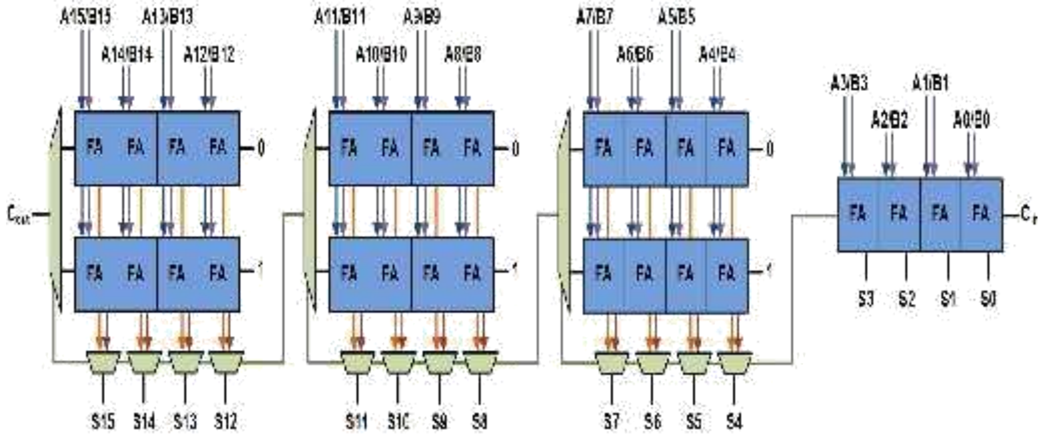


Figure: Building blocks of a carry-select adder

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.
- $t_{adder} = t_{setup} + Mt_{carry} + (N/M)t_{mux} + t_{sum}$



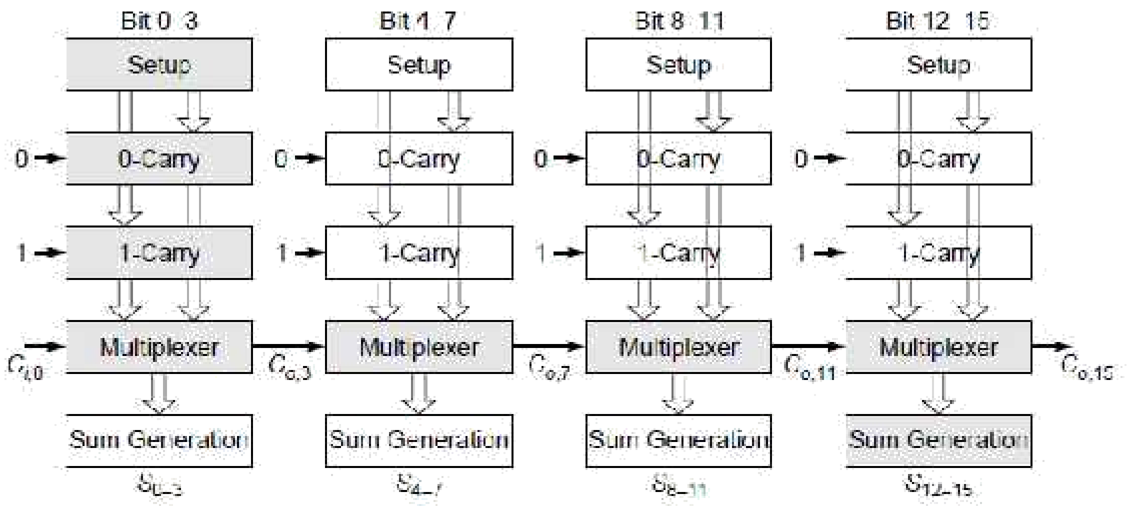
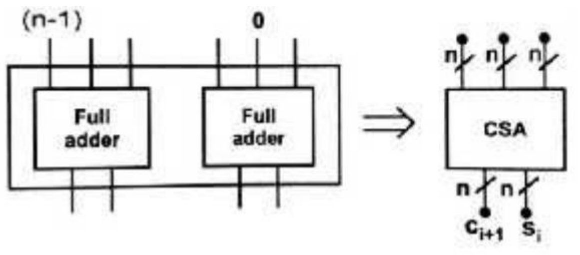


Figure: general structure of 16 bit adder

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

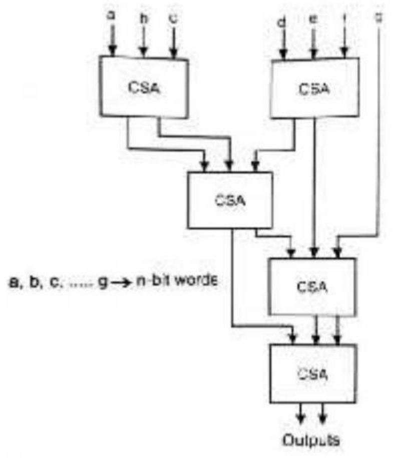


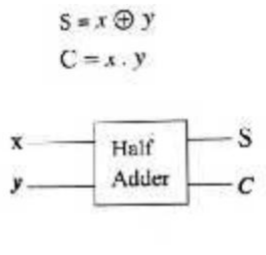
Figure: ***Carry Save Adder***

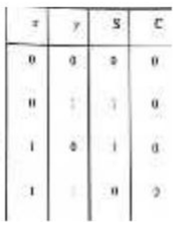
****

V. ACCUMULATOR:

Briefly discuss about 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.

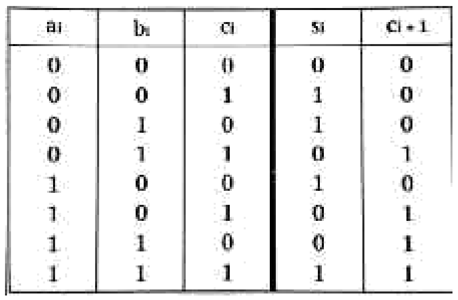




x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Figure: Half adder and Truth table

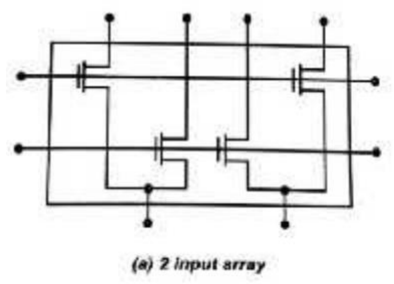
- Full adder circuit has three inputs and two outputs

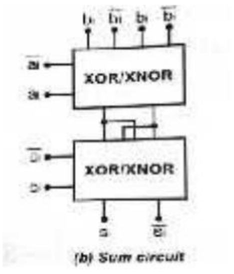


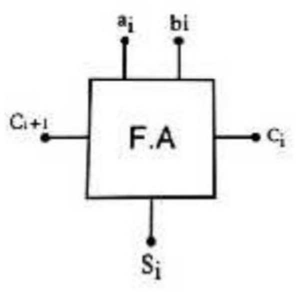
a_i	b_i	c_i	s_i	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure : Full adder and truth table

CPL --- Complementary Pass Logic







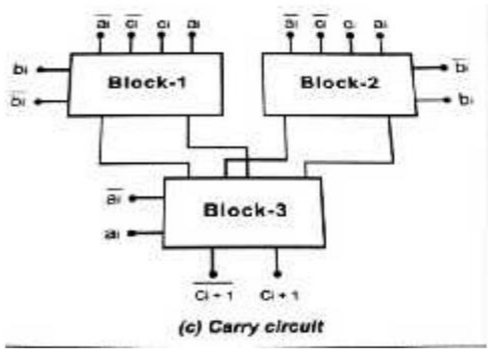


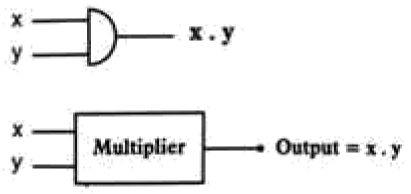
Figure : CPL Full adder design

****

VI. 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)**

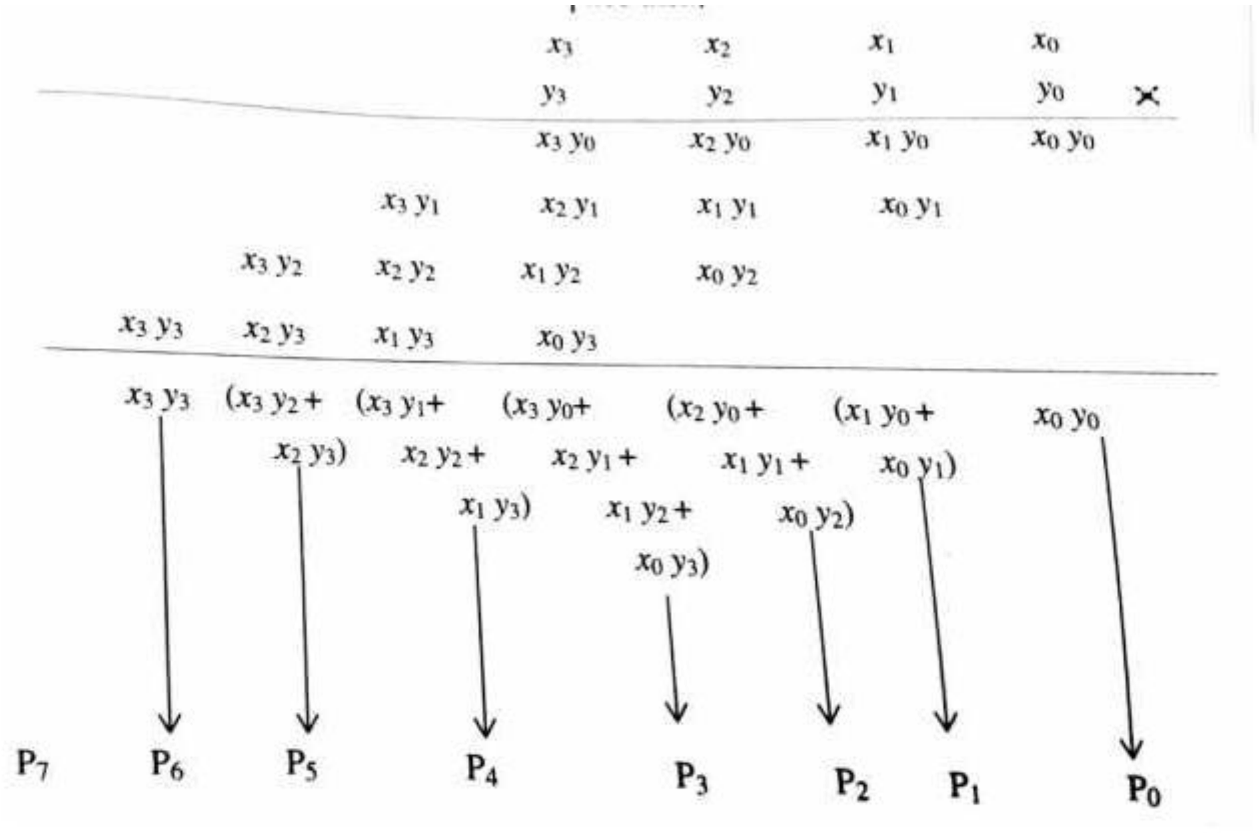
- A study of computer arithmetic processes will reveal that the most common requirements are for addition and subtraction.
- There is also a significant need for a multiplication capability.
- Basic operations in multiplication are given below.
  0 x 0 = 0, 0 x 1 = 0, 1 x 0 = 0, 1 x 1 = 1



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


x_3 x_2 x_1 x_0
y_3 y_2 y_1 y_0 ×
$x_3 y_0$ $x_2 y_0$ $x_1 y_0$ $x_0 y_0$
$x_3 y_1$ $x_2 y_1$ $x_1 y_1$ $x_0 y_1$
$x_3 y_2$ $x_2 y_2$ $x_1 y_2$ $x_0 y_2$
$x_3 y_3$ $x_2 y_3$ $x_1 y_3$ $x_0 y_3$
$x_3 y_3$
$(x_3 y_2 + x_2 y_3)$
$(x_3 y_1 + x_2 y_2 + x_1 y_3)$
$(x_3 y_0 + x_2 y_1 + x_1 y_2 + x_0 y_3)$
$(x_2 y_0 + x_1 y_1 + x_0 y_2)$
$(x_1 y_0 + x_0 y_1)$
$x_0 y_0$
P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0

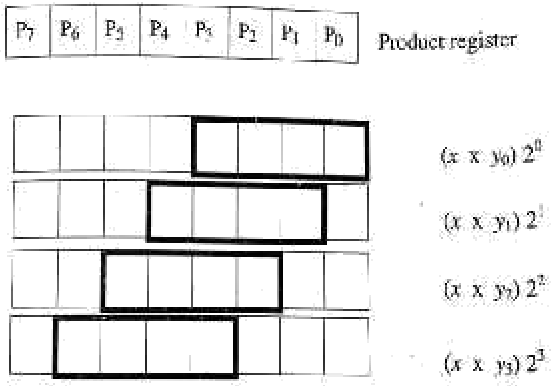
Product bits $\rightarrow (P_7\ P_6\ P_5\ P_4\ P_3\ P_2\ P_1\ P_0)$

$$P_i = \sum_{i=j+k} x_j y_k + c_{i-1}$$

$$c_{i-1} = 0 \text{ for } (i-1) \leq 0$$

Multiplication by shifting:

- If x=$(0020)_2 = (2)_{10}$
- 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:

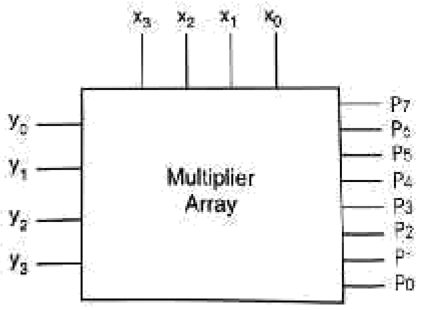


Figure: General block diagram of multiplier

- 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

$$P_i = \sum_{i=j+k} x_j y_k + c_{i-1}$$

$$P = X.Y = \left(\sum_{j=0}^{n-1} x_j 2^j\right)\left(\sum_{k=0}^{n-1} y_k 2^k\right)$$

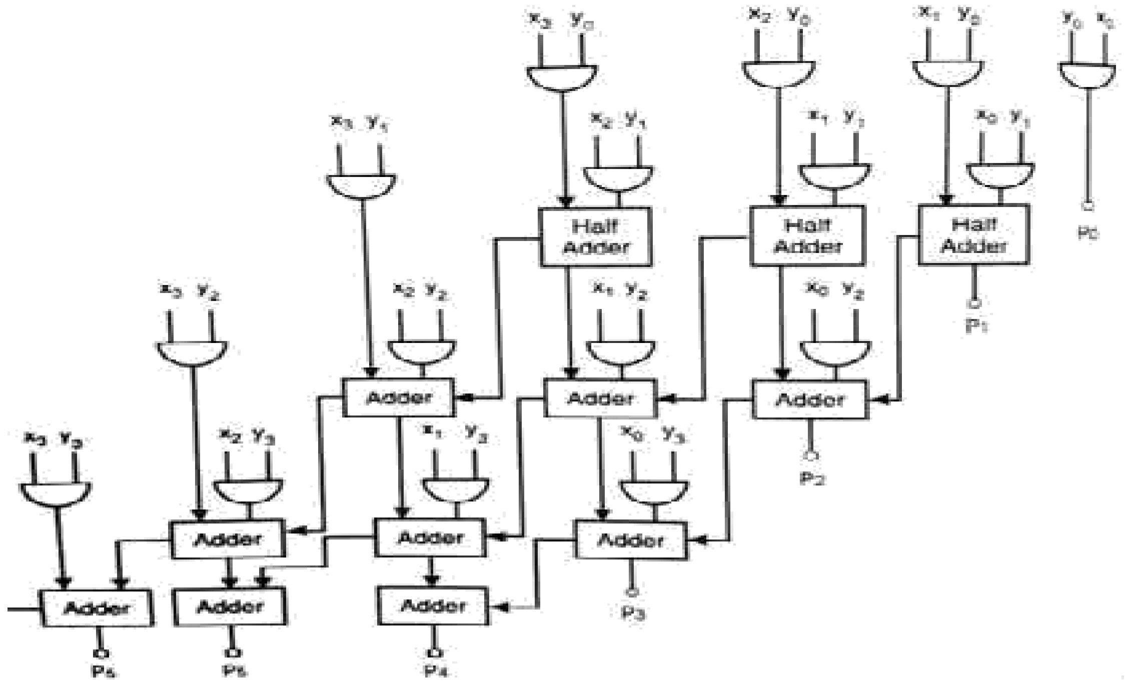


Figure: 4 x 4 array multiplier

- 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.

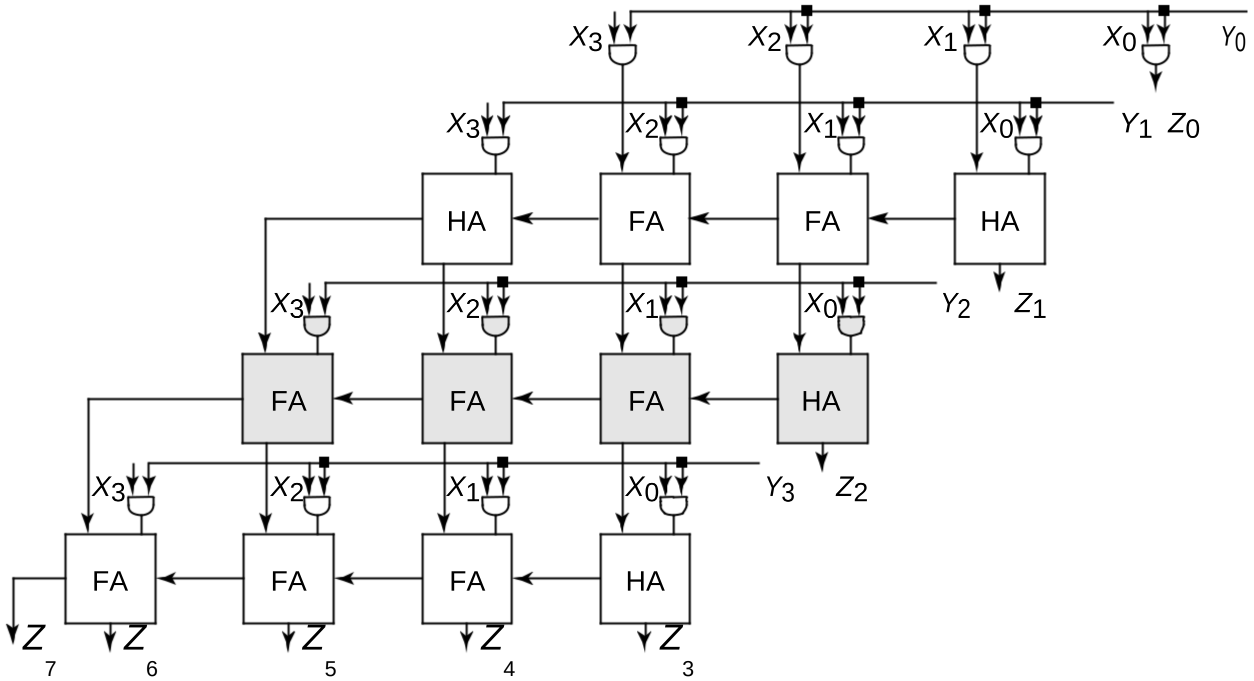


Figure: 4 x 4 array multiplier using Fulladder, Halfadder and AND gate.

(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.

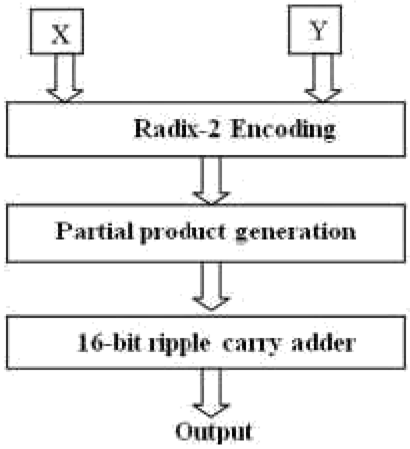


Figure: Block diagram of Booth multiplier

- 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.

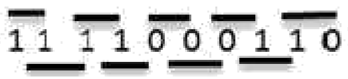


Figure: 2- Bit pairing as per Booth recoding using Radix- 2.

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.

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.

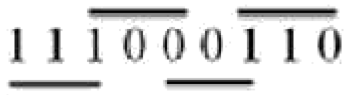
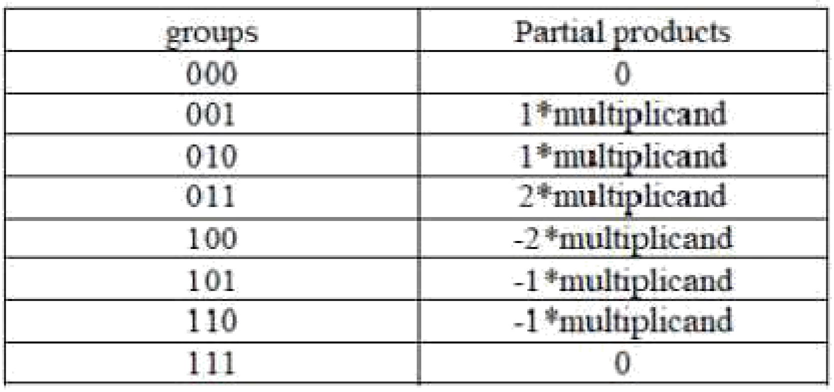


Figure. Grouping of 3-bit as per booth recoding

- 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

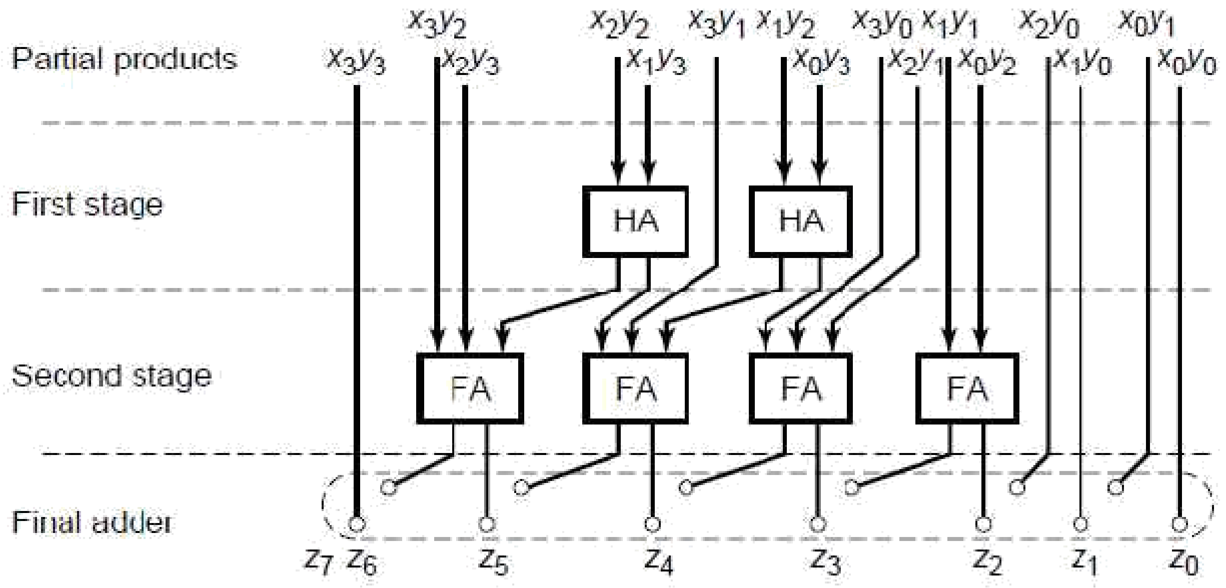
Table: Booth encoding table with RADIX-4

- **RADIX-4 PROCEDURE:**
  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.



****

VII. 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.

```
                          101   0(10)
(1's complement of 3)     1 1 0 0  +
                      [1] 0 1 1 0
                       └────────> 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  +
      [1] 0 0 1 1
       └────────> 1
          0 1 0 0  (3)
```

Carry is 1, so, it is added with the o/p.

$\boxed{7 - 3 = 4}$

```
          0 1 0 0  (4)
          1 1 0 0  +
      [1] 0 0 0 0
       └────────> 1
          0 0 0 1  (1)
```

$\boxed{4 - 3 = 1}$

Carry is 1, so, it is added with the o/p.

```
          0 0 0 1  (1)
          1 1 0 0  +
          1 1 0 1
```

There is no carry, so, the process is stopped.

Quoticnt = 3

Remainder = 0 0 0 1 (Final Difference)

The implementation of this division is given below.

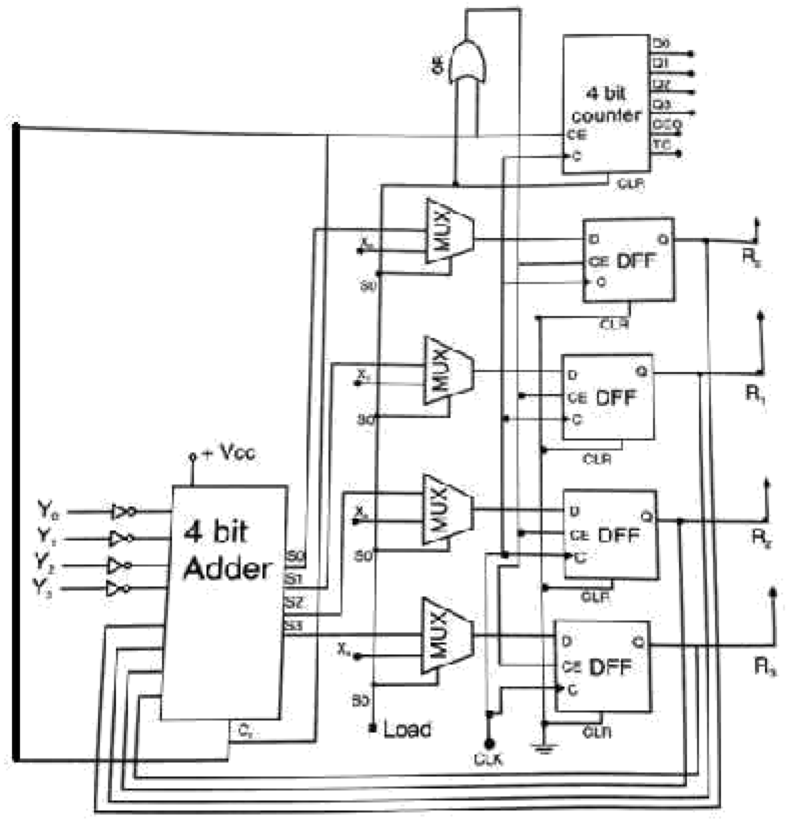
$X \div Y$

$X_3 X_2 X_1 X_0 \div Y_3 Y_2 Y_1 Y_0$

- 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)

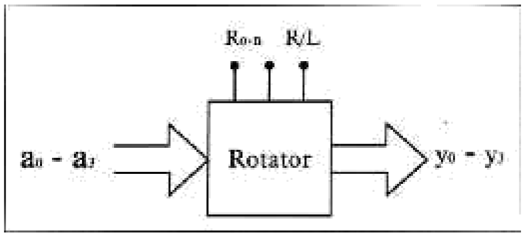


****

VIII. SHIFT REGISTERS:

Design 4 input and 4 output barrel shifter using NMOS logic. (NOV 2018).

- An n-bit rotation is specified by using the control word R_{0-n} and L/R bit defines a left or right shifting.

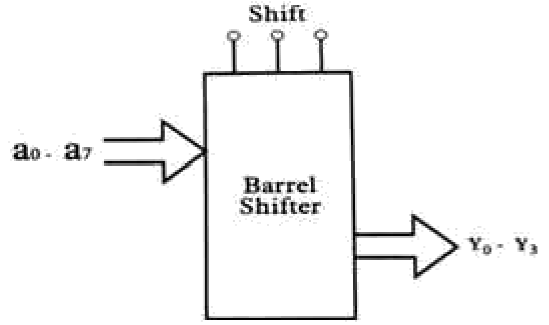


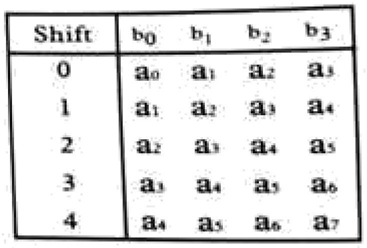
- For example $y_3\ y_2\ y_1\ y_0 = a_3\ a_2\ a_1\ a_0$

  If it is rotated 1-bit in left side, we get If $y_3\ y_2\ y_1\ y_0 = a_2\ a_1\ a_0\ \mathbf{a_3}$
  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 one clock 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**.

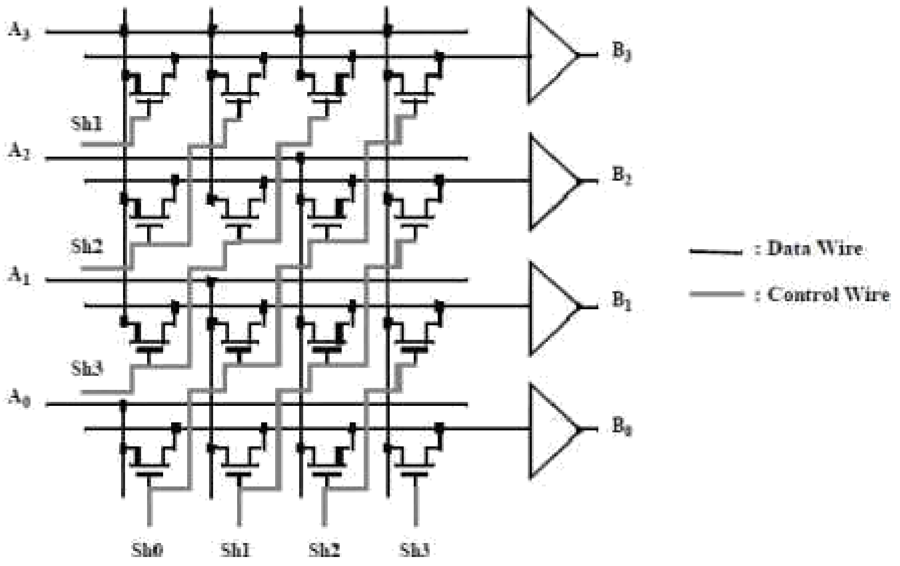
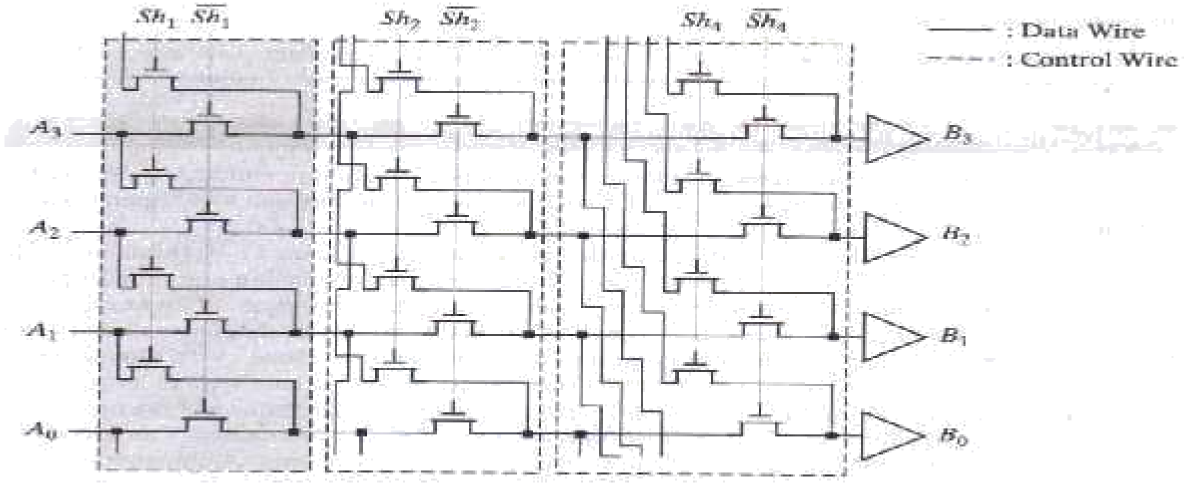


Figure: Barrel Shifter

- 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.



****

IX. SPEED AND AREA TRADE OFF:

Discuss the details about speed and area trade off. (May 2017)

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

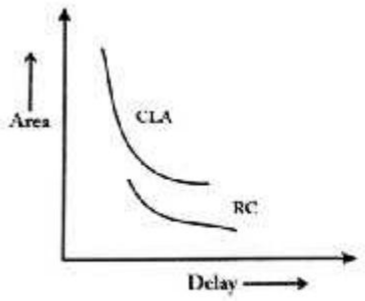


Figure: Area Vs Delay for 8 bit adder

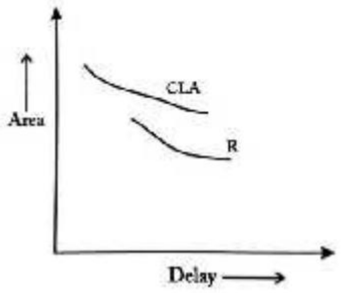


Figure: Area Vs Delay for 16 bit adder

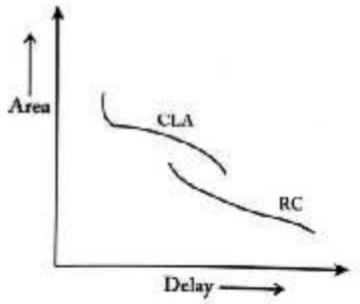


Figure: Area Vs delay for 32 bit adder

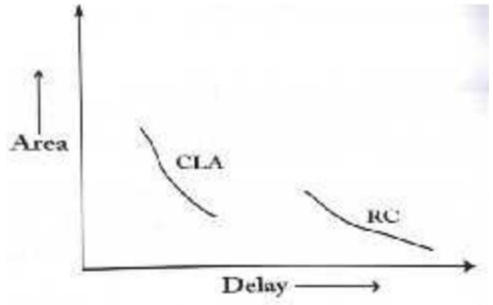


Figure: Area Vs delay for 64 bit adder

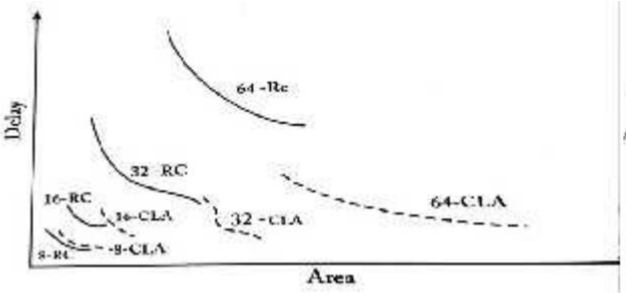


Figure: Delay Vs Area for all adders

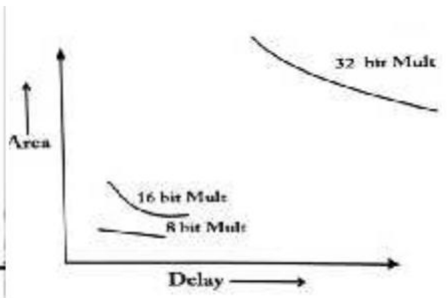
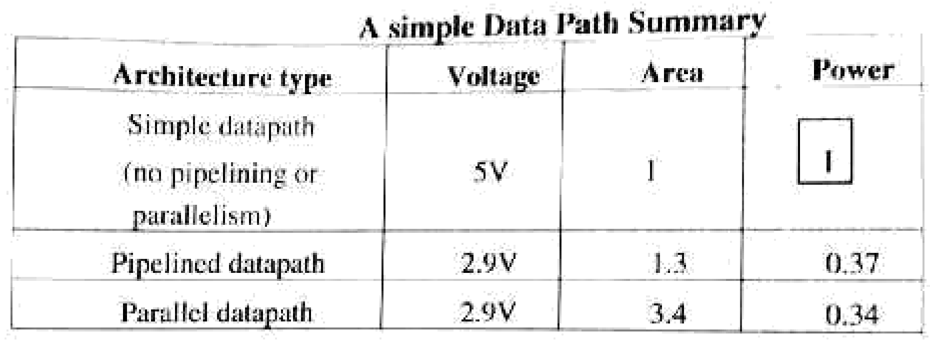


Figure: Area Vs Delay for all multiplier

- 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	$\boxed{1}$
Pipelined datapath	2.9V	1.3	0.37
Parallel datapath	2.9V	3.4	0.34

Memory Architecture and Building Blocks:

Explain the *memory architecture* and its control circuits in detail. (April 2018)

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.

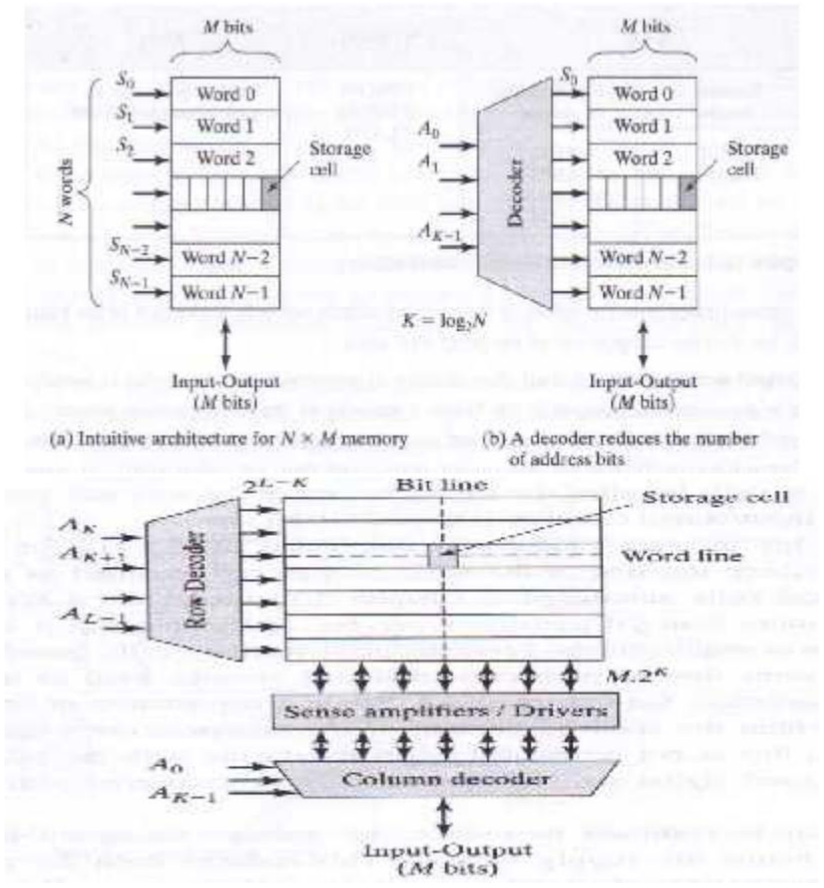
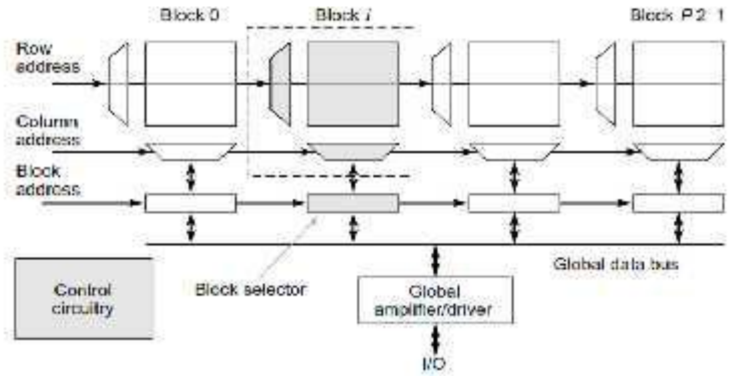


Figure: Array structured memory organization

- 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.

23. **Word line**: The horizontal select line which is used to select the single row of cell is known as word line.
24. **Bit line**: The wire which connects the cell in a single column to the input/output circuit is known as bit line.
25. **Sense amplifier**: It requires an amplification of the internal swin g to full rail-to-rail amplitude.
26. **Block address**: the mem ory is divided into various small blocks.
27. The address which is used to select one of the small blocks to be read or written is known as block address.
28. **Advantages:**
    Access time is fast
    Power saving is good, because blocks not activated are in power saving mode.



Fi gure: Hierarchical memory architecture

4.9.3 Memory Core

Discuss Memory core its type s in detail.

4.9.3.1 Read Only Memory (R OM):

ROM is a memory where code is written only one time.

Diode ROM:

24. It is simple where presence of diode in between bit line and word line is conssidered as logic 1 and absence of diode as logic 0.
25. Disadvantage is used for small memories and no isolation between word line and bit line.

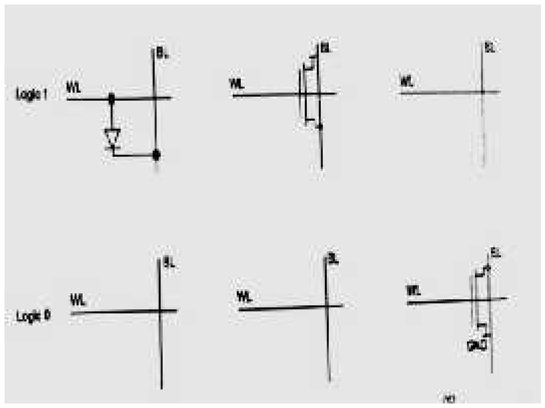


Figure: Diode ROM

MOS ROM:

- 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.

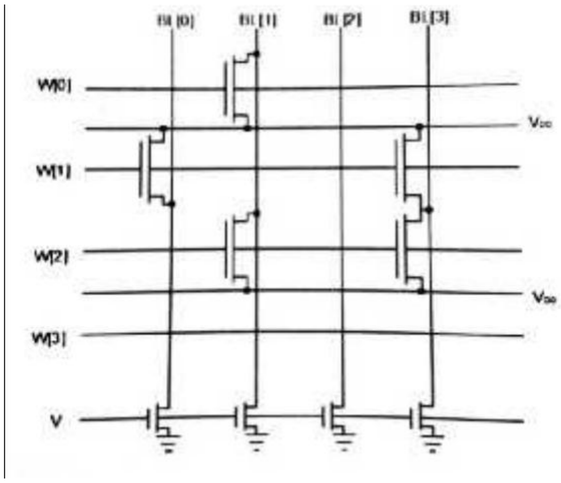


Figure: 4 x 4OR ROM cell array

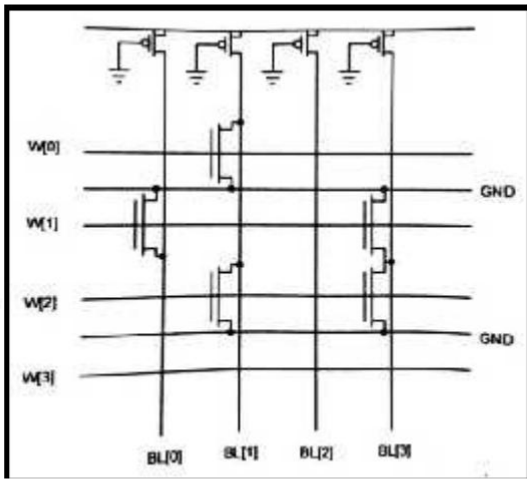


Figure: 4 x 4 MOS NOR ROM

Programming ROM

27. 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.

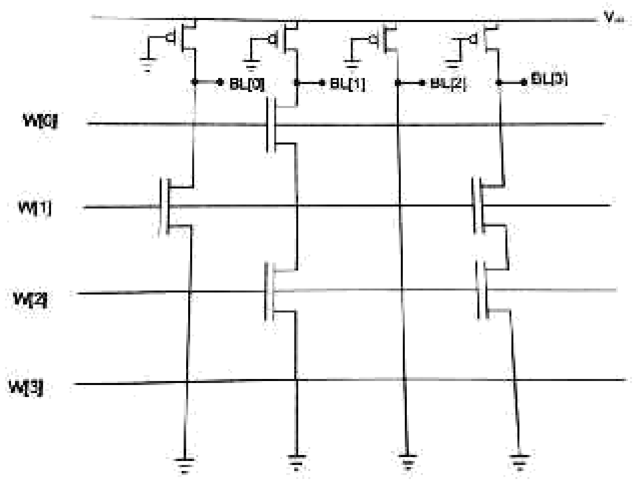


Figure: 4 x 4 MOS NAND ROM

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.

4.9.3.2 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.

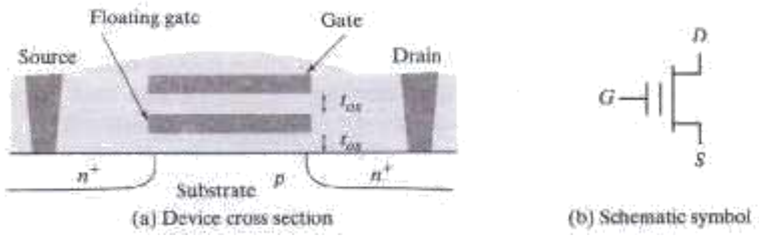


Figure (a) floating gate transistor (b) symbol

EPROM – Erasable Programmable Read Only Memory:

31. Erasing is done by passing UV rays on the cell by using transparent window.
32. This process will take some seconds to some minutes.
33. It depends on intensity of UV source. The programming takes 5-10microseconds/word.
34. 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:

32. Electrically Erasable Programmable ROM. Here Floating gate tunneling oxide (FLOTOX) is used.
33. 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 trravels to and from the floating gate through F owler-Nordheim tunneling.
- Erasing can be done by revering applied voltage which is used for writing.

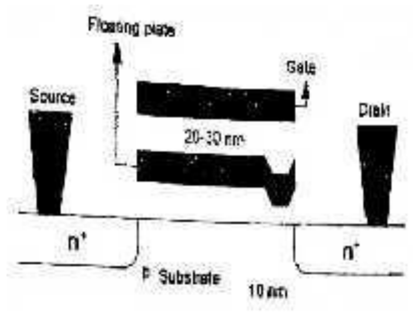


Figure: FLOTOX transistor

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 Electric ally Erasable Programmable ROM

34. It is a combination of density of EPROM and versatility of EEPROM.
35. Avalanche hot electron injecti on mechanism is used.
36. Erasing can be done by Fowle r-Nordheim tunneling concept. Here erasing is done in bulk.

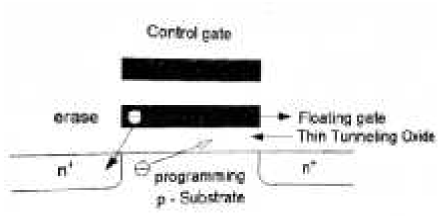


Figure: ETOX device

35. It is similar to FAMOS gate.
36. A very thin tunneling oxide la yer (10nm thickness) is there.
37. ***Erasing operation:*** Erasing can be performed when gate is connected to the ground and the source is connected to 12V.
38. ***Write operation:*** High voltage pulse is applied to the gate of the selected device. Logic 1 is applied to the drain and hot e lectrons are injected into the floating gate.
39. ***Read operation:*** To select a cell, its word line is connected to 5V. It c auses conditional discharge of the bit line.

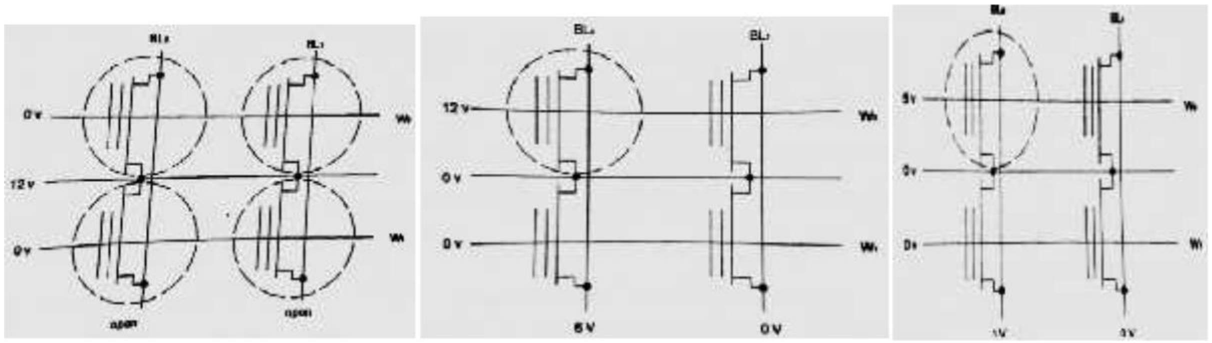


Figure: (a) Erase (b) Write (c) Read operation of NOR flash memory

4.9.3.3 RAM – Random Access Memory

Explain about static and dynamic RAM.
Construct 6T based SRAM cell. Explain its read and write operations. (NOV 2018)

4.9.3.3.1 Static RAM:

36. SRAM cell needs 6 transistors per bit.
37. M5 and M6 transistors are shared between read and write operations.
38. 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.

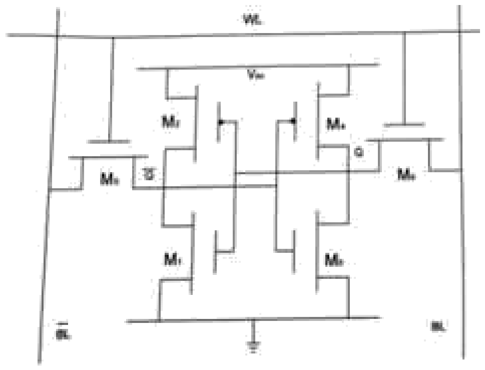


Figure: CMOS SRAM cell

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.

4.9.3.3.2 Dynamic RAM:

Three transistors DRAM

40. Content in the cell can be periodically rewritten through a resistive load, called as refresh operation.
41. This refresh occurs for every 1-4ms. Dynamic memory has refresh operation.
42. For example, logic 1 is to be written, and then BL1 is asserted high and write word line (WWL) is asserted.
43. This data is retained as charge on the capacitor once WWL is low.

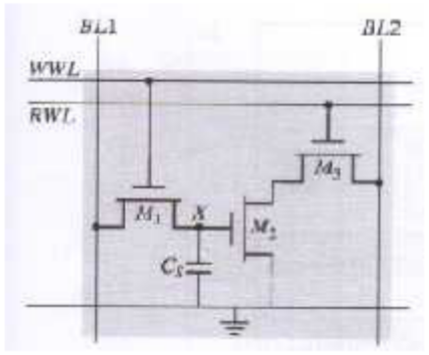


Figure: Three transistor dynamic memory cell

- 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.

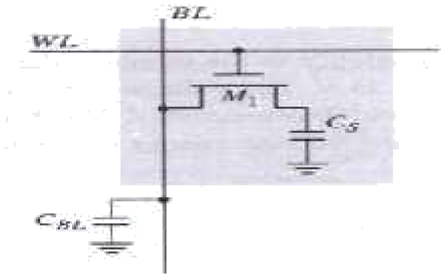


Figure: One transistor DRAM

4.9.3.3.3 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 a possible implementation of a CAM array.
- The cell combines a traditional 6T RAM storage cell *(M_4-M_9)* with additional circuitry to perform a l-bit digital comparison *(M_1-M_3)*.
- When the cell is to be written, complementary data is forced onto the bit lines, while the word line is enabled a s in a standard SRAM 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.

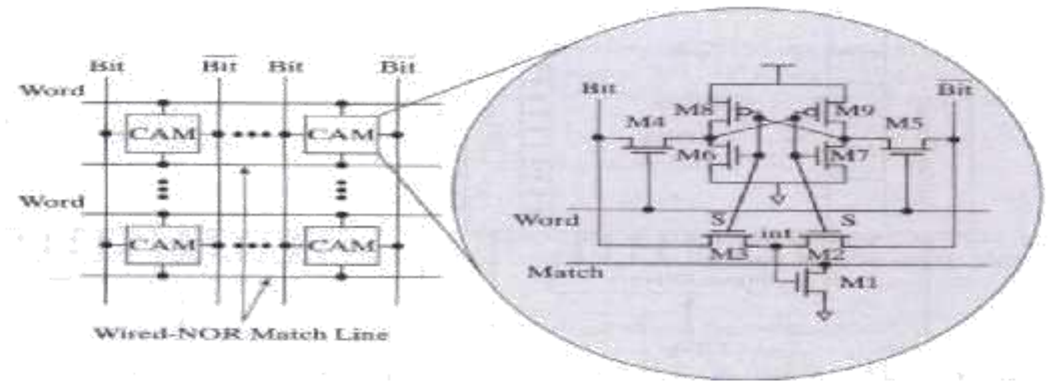


Figure: CAM cell

****

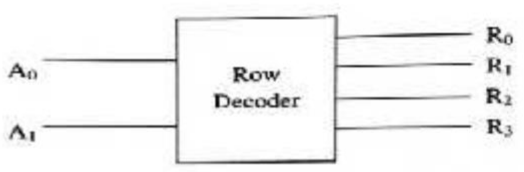
4.10 Memory peripheral (control) Circuits:

Explain the memory architecture and its *control circuits* in detail. (April 2018)

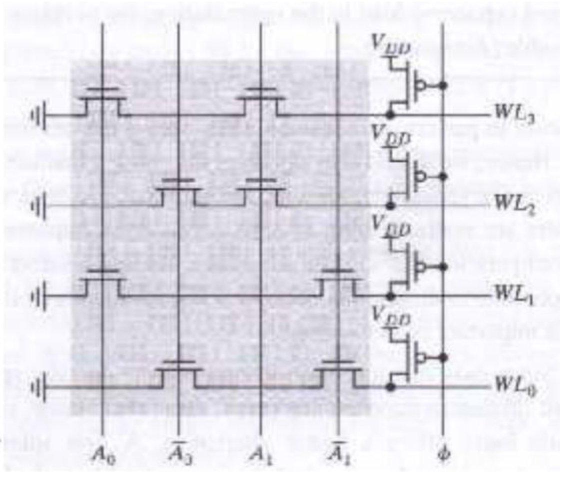
(i) Address & Block Decoders:

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

Dynamic 2-to-4 NOR decoder

Column Decoder

- It should match the bit line pitch of the memory 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.

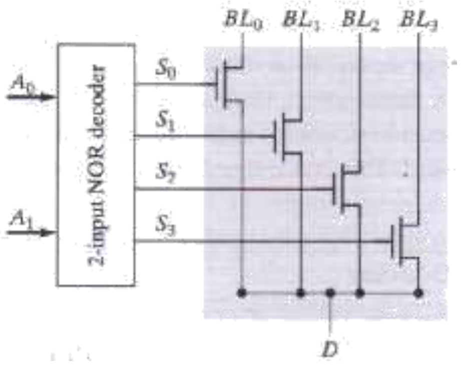
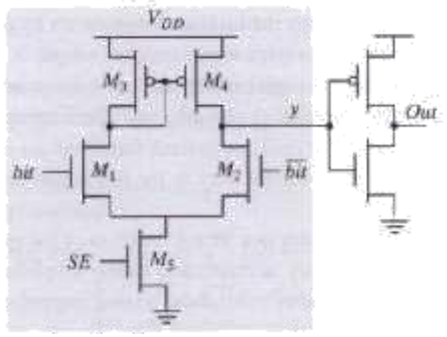


Figure: Four-input pass-transistor-based column decoder using a NOR predecoder

(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 1T DRAM, 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 a n d discharging the bit lines.

(iii) Drivers/ Buffers

- The length of word and bit lines increases with increasing memory sizes.
- Large portion o f the read and write access time can be attributed t o the wire delays.
- A major part of the memory-periphery area is allocated to the drivers (address buffers and I/O drivers).

****

4.11: Low Power Memory design:

Discuss about Low power memory design.

(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.

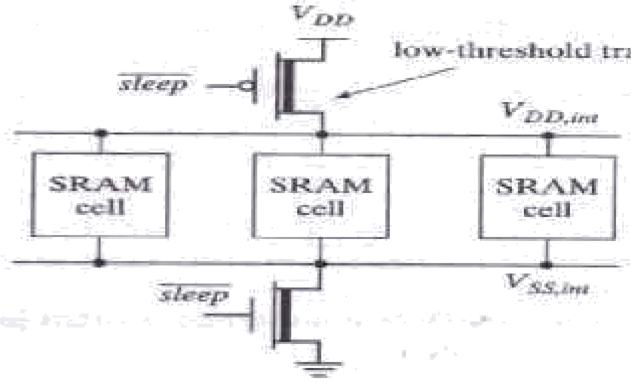
• Increasing the cell size:

- Ultra-low-voltage DRAM memory operation might require a sacrifice in area efficiency.

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.



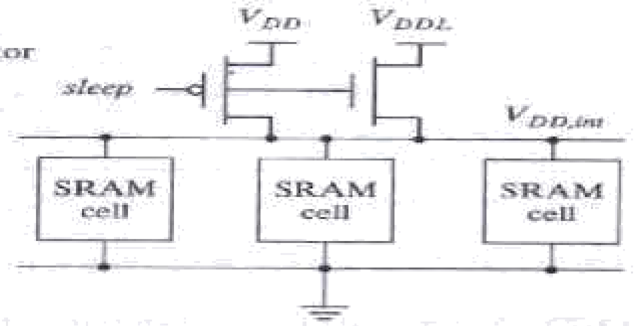


Figure: (a) Insertion of low threshold device (b) Reducing supply Voltage

****

UNIT V - IMPLEMENTATION STRATEGIES

FPGA Building Block Architectures, FPGA Interconnect Routing Procedures. Design for Testability: *Ad Hoc* Testing, Scan Design, BIST, IDDQ Testing, Design for Manufacturability, Boundary Scan.

- ❖ **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)**

Re-Programmable Devices Architecture (FPGA)

- 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:

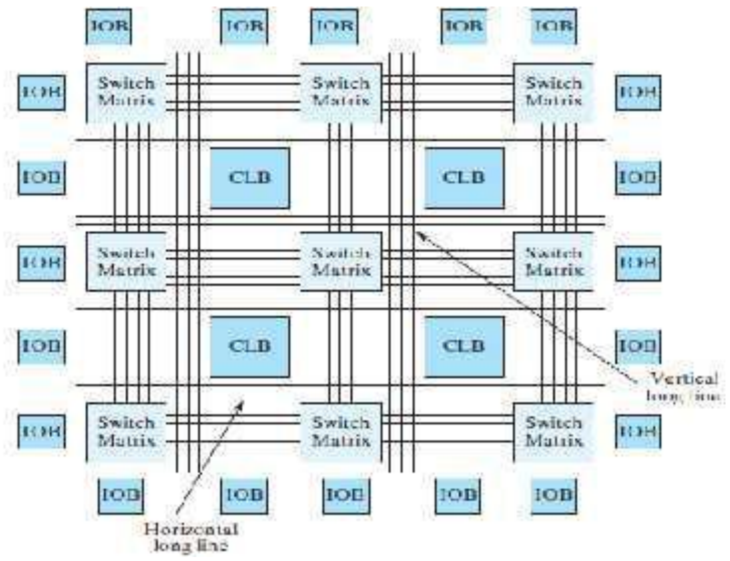
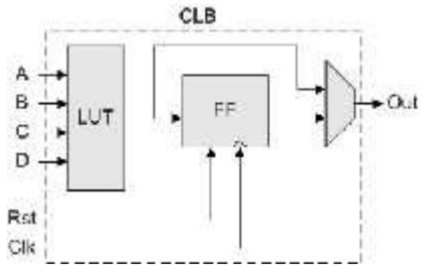
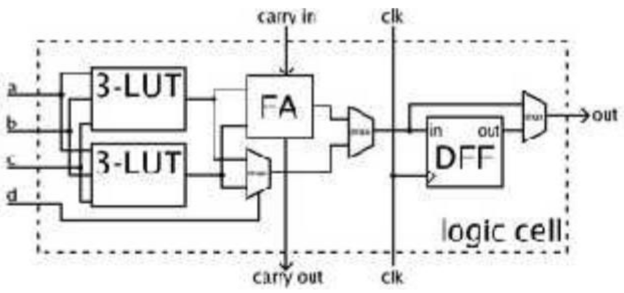


Figure: FPGA building blocks structure

- 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.

5.7.1: 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.

5.7.2: Interconnection resources:

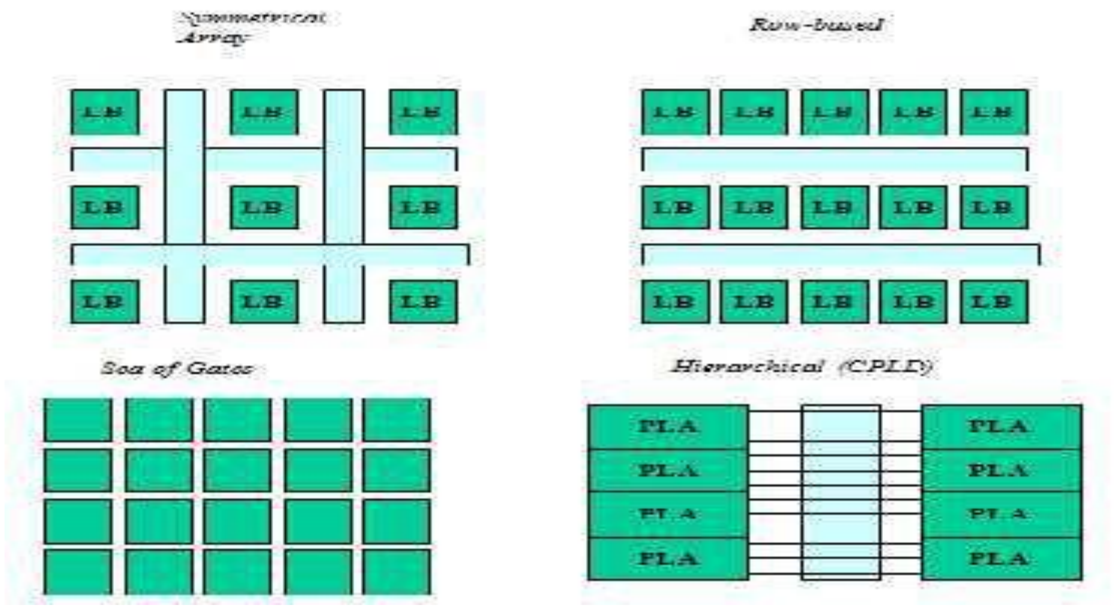


Figure: Types of interconnection resources

(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.

5.7.3: 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.

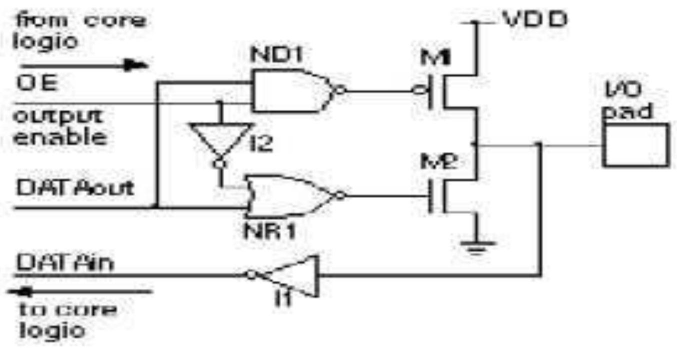


Figure: A three-state bidirectional output buffer

- 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.

****

*5.8: FPGA(*PROGRAMMABLE ASIC *)interconnect routing procedures (Architectures):*

❖ **Give short notes on FPGA interconnect routing procedures. (May 2016)**

- 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.

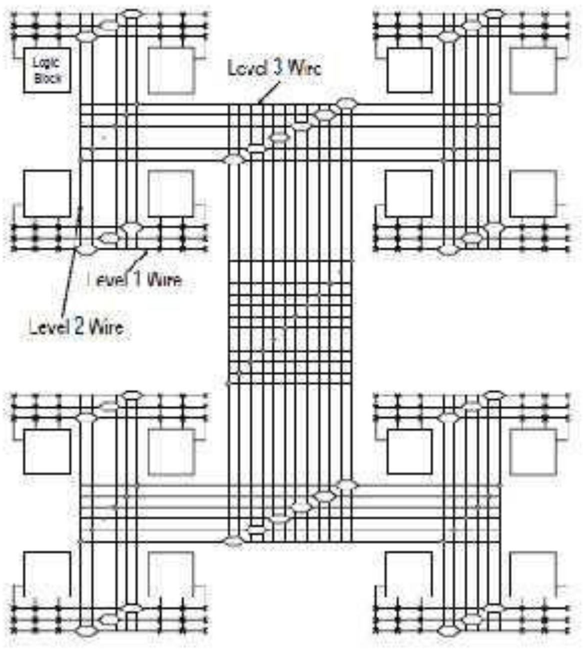


Figure: Example of Hierarchical

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.

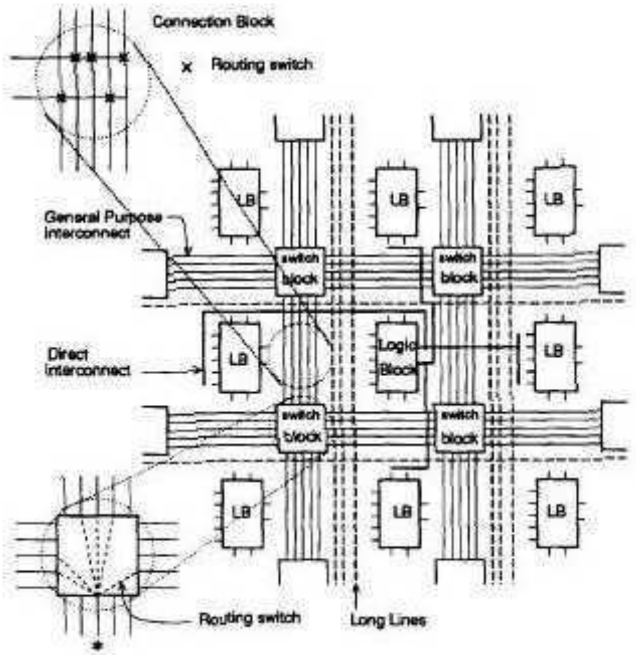


Figure: Xilinix Routing Architecture

- 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

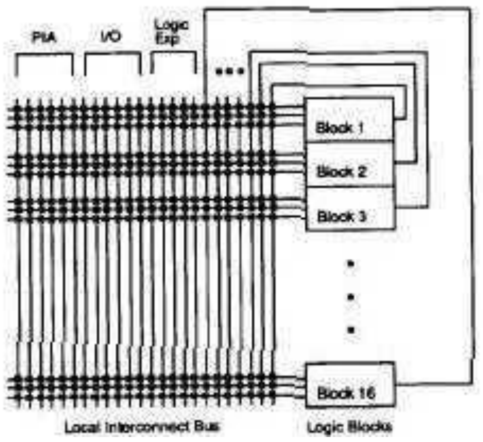


Figure: Altera Max 5000 Routing Architecture

- 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.

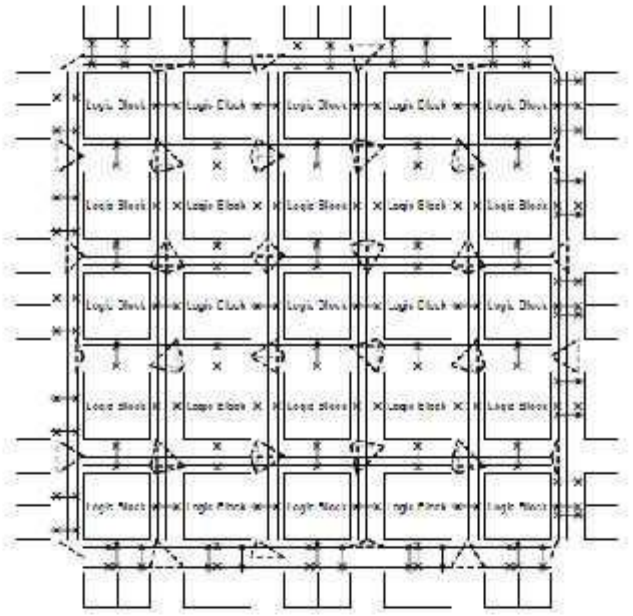


Figure: Island-Style Routing

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.

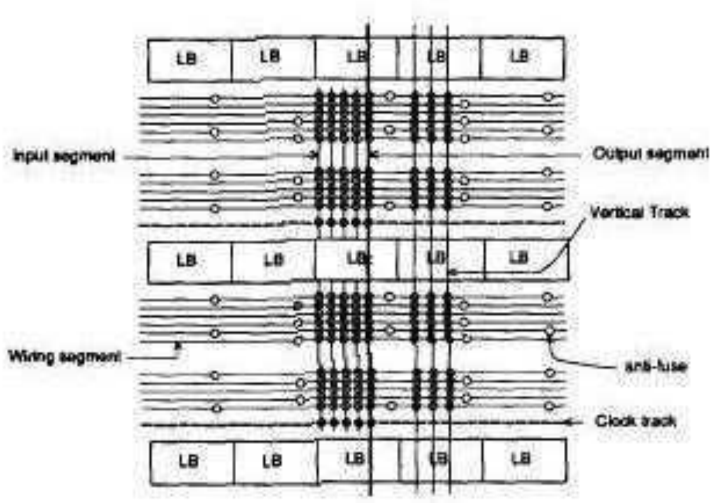


Figure: Actel Routing Architecture

DESIGN FOR TESTABILITY:

VLSI designers have a wide variety of CAD tools to choose from, each with their own strengths and weaknesses. The leading Electronic Design Automation (EDA) companies include Cadence, Synopsys, Magma, and Mentor Graphics.

Tanner also offers commercial VLSI design tools. The leading free tools include Electric, Magic, and LASI.

This set of laboratories uses the Cadence and Synopsys tools because they have the largest market share in industry, are capable of handling everything from simple class projects to state-of-the-art integrated circuits.

The full set of tools is extremely expensive but the companies offer academic programs to make the tools available to universities at a much lower cost.

The tools run on Linux and other flavors of UNIX. Setting up and maintaining the tool involves a substantial effort. Once they are setup correctly, the basic tools are easy to use, as this tutorial demonstrates.

Some companies use the Tanner tools because their list price is much lower and they are easy to use. However, their academic pricing is comparable with Cadence and Synopsys, giving little incentive for universities to adopt Tanner.

The Electric VLSI Design System is an open-source chip design program developed by Electric presently does not read the design rules for state-of-the-art nanometer processes and poorly integrates with synthesis and place & route.

Magic is a free Linux-based layout editor with a powerful but awkward interface that was once widely used in universities.

The Layout System for Individuals, LASI, developed by David Boyce, is freely available and runs on Windows. It was last updated in 1999. There are two general strategies for chip design.

Custom design involves specifying how every transistor is connected and physically arranged on the chip.

Synthesized design involves describing the function of a digital chip in a hardware description language such as Verilog or VHDL, then using a computer-aided design tool to automatically generate a set of gates that perform this function, place the gates on the chip, and route the wires to connect the connect the gates.

The majority of commercial designs are synthesized today because synthesis takes less engineering time.

However, custom design gives more insight into how chips are built and into what to do when things go wrong.

Custom design also offers higher performance, lower power, and smaller chip size. The first two labs emphasize the fundamentals of custom design, while the next two use logic synthesis and automatic placement to save time.

Tool Setup

These labs assume that We have the Cadence and Synopsys tools installed. The tools generate a bunch of random files. It's best to keep them in one place. In your home directory,

create some directories by typing:

mkdir IC_CAD

mkdir IC_CAD/cadence

Getting Started

Before you start the Cadence tools, change into the cadence directory:cd ~/IC_CAD/cadence Each of our tools has a startup script that sets the appropriate paths to the tools and invokes them.

Start Cadence with the NCSU extensions by running cad-ncsu & A window labeled icfb will open up.

This is the Integrated Circuit Front and Back End (e.g. schematic and layout) software, part of Cadence's Design Framework interface.

A "What's New" and a Library Manager window may open up too. Scroll through the icfb window and look at the messages displayed as the tool loads up.

Get in the habit of watching for the messages and recognizing any that are out of the ordinary.

This is very helpful when We encounter problems. All of your designs are stored in a library. If the Library Browser doesn't open, choose Tools

Library Manager. We'll use the Library Manager to manipulate your libraries. Don't try to move libraries around or rename them directly in Linux; there is some funny behavior and We are likely to break them.

Familiarize Werself with the Library Manager. Your cds.lib file includes many libraries from the NCUS CDK supporting the different MOSIS processes. It also includes libraries from the University of Utah.

The File menu allows We to create new libraries and cells within a library, while the Edit menu allows We to copy, rename, delete, and change the access permissions.

Choose the "Attach to existing tech library" and accept the default, UofU AMI 0.60u C5N (3M, 2P, high-res).

This is a technology file for the American Microsystems (now Orbit Semiconductor) 0.6 µm process, containing design rules for layout.

Schematic Entry

Our first step is to create a schematic for a 2-input NAND gate. Each gate or larger component is called a cell. Cells have multiple views. The schematic view for a cell built with CMOS transistors will be called cmos sch.

Later, We will build a view called layout specifying how the cell is physically manufactured. In the Library Manager, choose File • New • Cell View… In your lab1_xx library, enter a cell name of nand2 and a view name of cmos_sch. The tool should be Composer - Schematic.

We may get a window asking you to confirm that cmos_sch should be associated with this tool. The schematic editor window will open. Your goal is to draw a gate like the one shown in Figure 1. We are working in a 0.6 µm process with λ = 0.3 µm.

Unfortunately, the University of Utah technology file is configured on a half-lambda grid, so grid units are 0.15 µm. Take care that everything We do is an integer multiple of λ so We don't come to grief later on. Our NAND gate will use 12 λ (3.6 µm) nMOS and pMOS transistors.

Choose Add • Instance to open a Component Browser window. Choose UofU_Analog_Parts for the library, then select nmos. The Add Instance dialog will open. Set the Width to 3.6u (u indicates microns).

Click in the schematic editor window to drop the transistor. We can click a second time to place another transistor. Return to the Component Browser window and choose pmos. Drop two pMOS transistors.

Then return to the browser and get a gnd and a vdd symbol. When We are in a mode in the editor, We can press ctrl-c or Esc to get out of it.

Other extremely useful commands include Edit • Move, Edit • Copy, Edit • Undo, and Edit
35. Delete. Edit • Properties • Object… is also useful to change things like transistor sizes or wire names.

Move the elements around until they are in attractive locations. I like to keep series transistors one grid unit apart and place pMOS transistors two grid units above the nMOS. Look at the bottom of the schematic editor window to see what mode We are in.

Next, use Add • Pin… to create some pins. In the Add Pin dialog, enter a and b. Make sure the direction is "input."

The tools are case-sensitive, so use lower case everywhere. Place the pins, being sure that a is the bottom one.

Although pin order doesn't matter logically, it does matter physically and electrically, so We will get errors if We reverse the order. Then place an output pin y. Now, wire the elements together.

Choose Add • Wire (narrow). Click on each component and draw a wire to where it should connect. It is a good idea to make sure every net (wire) in a design has a name.

Otherwise, We'll have a tough time tracking down a problem later on one of the unnamed nets.

Every net in Wer schematic is connected to a named pin or to power or ground except the net between the two series nMOS transistors. Choose Add • Wire name… Enter mid or something like that as the name, and click on the wire to name it. Choose Design • Check and Save to save Wer schematic.

We'll probably get one warning about a "solder dot on crossover" at the 4-way junction on the output node.

This is annoying because such 4-way junctions are normal and common. Choose Check • Rules Setup… and click on the Physical tab in the dialog. Change Solder On CrossOver from "warning" to "ignored" and close the dialog.

Then Check and Save again and the warning should be gone.

If We have any other warnings, fix them. A common mistake is wires that look like they might touch but don't actually connect. Delete the wire and redraw it. Poke around the menus and familiarize Werself with the other capabilities of the schematic editor.

LOGIC VERIFICATION

Cells are commonly described at three levels of abstraction. The register-transfer level (RTL) description is a Verilog or VHDL file specifying the behavior of the cell in terms of registers and combinational logic.

It often serves as the specification of what the chip should do. The schematic illustrates how the cell is composed from transistors or other cells. The layout shows how the transistors or cells are physically arranged.

Logic verification involves proving that the cells perform the correct function. One way to do this is to simulate the cell and apply a set of 1's and 0's called test vectors to the inputs, then check that the outputs match expectation.

Typically, logic verification is done first on the RTL to check that the specification is correct. A testbench written in Verilog or VHDL automates the process of applying and checking all of the vectors.

The same test vectors are then applied to the schematic to check that the schematic matches the RTL.

Later, we will use a layout-versus schematic (LVS) tool to check that the layout matches the schematic (and, by inference, the RTL).

We will begin by simulating an RTL description of the NAND gate to become familiar with reading RTL and understanding a testbench. In this tutorial, the RTL and testbench are written in System Verilog, which is a 2005 update to the popular Verilog hardware description language.

There are many Verilog simulators on the market, including NC-Verilog from Cadence, VCS from Synopsys, and ModelSim from Mentor Graphics.

This tutorial describes how to use NC Verilog because it integrates gracefully with the other Cadence tools.

NCVerilog compiles your Verilog into an executable program and runs it directly, making it much faster than the older interpreted simulators. Make a new directory for simulation (e.g. nand2sim).

Copy nand2.sv, nand2.tv, and testfixture.verilog from the course directory into your new directory.

```
mkdir nand2sim cd nand2sim
```

```
cp /courses/e158/10/nand2.sv . cp /courses/e158/10/nand2.tv .
```

```
cp /courses/e158/10/nand2.testfixture testfixture.verilog
```

nand2.sv is the SystemVerilog RTL file, which includes a behavioral description of a nand2 module and a simple self-checking testbench that includes testfixture.verilog. testfixture.verilog reads in testvectors from nand2.tv and applies them to pins of the nand2 module.

After each cycle it compares the output of the nand2 module to the expected output, and prints an error if they do not match.

Look over each of these files and understand how they work. First, We will simulate the nand2 RTL to practice the process and ensure that the testbench works.

Later, We will replace the behavioral nand2 module with one generated from Wer Electric schematic and will resimulate to check that your schematic performs the correct function.

At the command line, type sim-nc nand2.sv to invoke the simulator. We should see some messages ending with

```
ncsim> run

Completed 4 tests with 0 errors.

Simulation stopped via $stop(1) at time 81 NS + 0
```

We'll be left at the ncsim command prompt. Type quit to finish the simulation. If the simulation hadn't run correctly, it would be helpful to be able to view the results.

NC-Verilog has a graphical user interface called SimVision. The GUI takes a few seconds to load, so We may prefer to run it only when We need to debug.

To rerun the simulation with the GUI, type sim-ncg nand2.sv A Console and Design Browser window will pop up.

In the browser, click on the + symbol beside the testbench to expand, then click on dut. The three signals, a, b, and y, will appear in the pane to the right. Select all three, then right-click and choose Send to Waveform Window.

In the Waveform Window, choose Simulation • Run. We'll see the waveforms of your simulation; inspect them to ensure they are correct. The 0 errors message should also appear in the console.

If you needed to change something in your code or testbench or test vectors, or wanted to add other signals, do so and then Simulation • Reinvoke Simulator to recompile everything and bring We back to the start.

Then choose Run again. Make a habit of looking at the messages in the console window and learning what is normal.

Warnings and errors should be taken seriously; they usually indicate real problems that will catch We later if We don't fix them.

Schematic Simulation

Next, We will verify your schematic by generating a Verilog deck and pasting it into the RTL Verilog file.

While viewing your schematic, click on Tools • Simulation • NCVerilog to open a window for the Verilog environment. Note the run directory (e.g. nand2_run1), and press the button in the upper left to initialize the design.

Then press the next button to generate a netlist. Look in the icfb window for errors and correct them if necessary.

We should see that the pmos, nmos, and nand2 cells were all netlisted. In your Linux terminal window, cd into the directory that was created. We'll find quite a few files.

The most important are verilog.inpfiles, testfixture.template, and testfixture.verilog. Each cell is netlisted into a different directory under ihnl. verilog.inpfiles states where they are.

Take a look at the netlist and other files. testfixture.template is the top level module that instantiates the device under test and invokes the testfixture.verilog.

Copy your from your nand2sim directory to your nand2_run1 directory using a command such as

```
cp ../nand2sim/testfixture.verilog . cp ../nand2sim/nand2.tv .
```

Back in the Virtuoso Verilog Environment window, We may wish to choose Setup • Record Signals.

Click on the "All" button to record signals at all levels of the hierarchy. (This isn't important for the nand with only one level of hierarchy, but will be helpful later.)

Then choose Setup • Simulation. Change the Simulation Log File to indicate simout.tmp –sv. This will print the results in simout.tmp.

The –sv flag indicates that the simulator should accept SystemVerilog syntax used in the testfixture.verilog. Set the Simulator mode to "Batch" and click on the Simulate button.

We should get a message that the batch simulation succeeded.This doesn't mean that it is correct, merely that it run.

In the terminal window, view the simout.tmp file. It will give some statistics about the compilation, then should indicate that the 4 tests were completed with 0 errors.

If the simulation fails, the simout.tmp file will have clues about the problems. Change the simulator mode to Interactive to rerun with the GUI.

Be patient; the GUI takes several seconds to start and gives no sign of life until then. Add the waveforms again and run the simulation.

We may need to zoom to fit all the waves. For some reason, SimVision doesn't print the $display message about the simulation succeeding with no errors.

We will have to read the simout.tmp file at the command line to verify that the test vectors passed. If We find any logic errors, correct the schematic and resimulate.

SILICON DEBUG PRINCIPAL

The rapid pace of innovation has created powerful SOC solutions at consumer prices.

This has created a highly competitive market place where billions of dollars can be won by the right design delivered at the right time.

These new designs are produced on processes that challenge the fundamental law of physics and are highly sensitive to equipment variation.

The industry now produces new designs in a complex world where process and design interaction have created new complex failures that stand in the way of billion-dollar opportunities.

These interactions lead to new types of defects such as blocked chains, which create noise in the debug/diagnosis process.

They also lead to new types of design issues such as delay defects in combinational and sequential logic.

The challenge is made even greater by the growing complexity in device structure and design techniques.

Multiple design organizations use multiple IP blocks and multiple libraries that need to work Together throughout the process window, often across multiple fabs.

These new challenges come at a time when product lifetimes are shrinking, leading to pressure to reduce time for debug and characterization activities. These problems are seen for the first time at first silicon.

Test the first chips back from fabrication If We are lucky, they work the first time If not Logic bugs vs. electrical failures Most chip failures are logic bugs from inadequate simulation or verification Some are electrical failures Crosstalk Dynamic nodes: leakage, charge sharing Ratio failures A few are tool or methodology failures (e.g. DRC) Fix the bugs and fabricate a corrected chip Silicon debug (or "bringup") is primarily a Non-Recurring Engineering (NRE) cost (like design) Contrast this with manufacturing test which has to be applied to every part shipped.

MANUFACTURING TEST

A speck of dust on a wafer is sufficient to kill chip Yield of any chip is < 100% Must test chips after manufacturing before delivery to customers to only ship good parts Manufacturing testers are very expensive Minimize time on tester Careful selection of test vectors.

A test for a defect will produce an output response which is different from the output when there is no defect Test quality is high if the set of tests will detect a very high fraction of possible defects Defect level is the percentage of bad parts shipped to customers Yield is the percentage of defect-free chips manufactured

Fault models:

Numerous possible physical failures (what we are testing for) Can reduce the number of failure types by considering the effects of physical failures on the logic functional blocks: called a Assume that defects will cause the circuit to behave as if lines were "stuck" at logic 0 or 1 Most commercial tools for test are based on the "stuck-at" model Other fault models "Stuck open" model for charge retained on a CMOS node Recent use of the "transition" fault model in an attempt to deal with delays "Path delay" fault model would be better for small delay defects, but the large number of possible paths is an impediment to the use of this fault model.

DESIGNS FOR TESTABILITY

Approach to generating tests for defects is to map defects to (higher level) faults: develop fault model, then generate tests for the faults Typical: gate-level "stuck-at" fault model As technology shrinks, other faults: bridging faults, delay faults, crosstalk faults, etc.

An interesting point: what is important is how well the tests generated (based on the fault model) will detect realistic defects the accuracy of the fault model is secondary

Observability and controllability:

Observability: ease of observing a value on a node by monitoring external output pins of the chip Controllability: ease of forcing a node to 0 or 1 by driving input pins of the chip Combinational logic is usually easier to observe and control Still, NP-complete problem Finite state machines can be very difficult, requiring many cycles to enter desired state Especially if state transition diagram is not known to the test engineer, or is too large

Fault simulation:

Identify faults detected by a sequence of tests Provide a numerical value of coverage (ratio of detected faults to total faults) Correlation between high fault coverage and low defect level Faults considered Generally, gate level "stuck-at" faults Can also evaluate coverage of switch level faults Can include timing and dynamic effects of failures.

Although fault simulation takes polynomial time in the number of gates, it can still be prohibitive for large designs. Static timing analysis (Primetime, for example) only finds structural long paths

False Path problem:

In order to allow a signal to go through the path, Required Side Inputs: C = 1, A = 1, E = 1 Conflict due to C = 1 and E = 1 Can use modified test generation algorithms to identify longest true paths in a circuit CRITIC from UT Primetime+Tetramax from Synopsys.

BOUNDARY SCAN

Boundary scan is a method for testing interconnects (wire lines) on printed circuit boards or sub-blocks inside an integrated circuit. Boundary scan is also widely used as a debugging method to watch integrated circuit pin states, measure voltage, or analyze sub-blocks inside an integrated circuit.

Testing

The boundary scan architecture provides a means to test interconnects and clusters of logic, memories etc. Without using physical test probes. It adds one or more so called 'test cells' connected to each pin of the device that can selectively override the functionality of that pin.

These cells can be programmed via the JTAG scan chain to drive a signal onto a pin and across an individual trace on the board. The cell at the destination of the board trace can then be programmed to read the value at the pin, verifying the board trace properly connects the two pins.

If the trace is shorted to another signal or if the trace has been cut, the correct signal value will not show up at the destination pin, and the board will be observed to have a fault.

On-Chip Infrastructure

To provide the boundary scan capability, IC vendors add additional logic to each of their devices, including scan cells for each of the external traces.

These cells are then connected together to form the external boundary scan shift register (BSR), and combined with JTAG TAP (Test Access Port) controller support comprising four (or sometimes more) additional pins plus control circuitry.

Some TAP controllers support scan chains between on-chip logical design blocks, with JTAG instructions which operate on those internal scan chains instead of the BSR.

This can allow those integrated components to be tested as if they were separate chips on a board. On-chip debugging solutions are heavy users of such internal scan chains.

These designs are part of most Verilog or VHDL libraries. Overhead for this additional logic is minimal, and generally is well worth the price to enable efficient testing at the board level.

For normal operation, the added boundary scan latch cells are set so that they have no effect on the circuit, and are therefore effectively invisible.

However, when the circuit is set into a test mode, the latches enable a data stream to be shifted from one latch into the next.

Once a complete data word has been shifted into the circuit under test, it can be latched into place so it drives external signals.

Shifting the word also generally returns the input values from the signals configured as inputs.

Test Mechanism

As the cells can be used to force data into the board, they can set up test conditions. The relevant states can then be fed back into the test system by clocking the data word back so that it can be analyzed.

By adopting this technique, it is possible for a test system to gain test access to a board. As most of today's boards are very densely populated with components and tracks, it is very difficult for test systems to physically access the relevant areas of the board to enable them to test the board. Boundary scan makes access possible without always needing physical probes.

In modern chip and board design, Design For Test is a significant issue, and one common design artifact is a set of boundary scan test vectors, possibly delivered in Serial Vector Format (SVF) or a similar interchange format.

JTAG Test Operations

Devices communicate to the world via a set of input and output pins. By themselves, these pins provide limited visibility into the workings of the device.

However, devices that support boundary scan contain a shift-register cell for each signal pin of the device. These registers are connected in a dedicated path around the device's boundary (hence the name).

The path creates a virtual access capability that circumvents the normal inputs and provides direct control of the device and detailed visibility at its outputs. he contents of the boundary scan are usually described by the manufacturer using a part-specific BSDL file.

The boundary-scan cells can be configured to support external testing for interconnection between chips (EXTEST instruction) or internal testing for logic within the chip (INTEST instruction).

Board Test Infrastructure

Typically high-end commercial JTAG testing systems allow the import of design 'netlists' from CAD/EDA systems plus the BSDL models of boundary scan/JTAG complaint devices to automatically generate test applications. Common types of test include

- ✓ Scan-path 'infrastructure' or integrity
- ✓ Boundary-scan device pin to boundary-scan device pin 'interconnect'
- ✓ Boundary-scan pin to memory device or device cluster (SRAM, DRAM, DDR etc)
- ✓ Arbitrary logic cluster testing

When used during manufacturing, such systems also support non-test but affiliated applications such as in-system programming of various types of flash memory: NOR, NAND, and serial (I2C or SPI).

Such commercial systems are used by board test professionals and will often cost several thousand dollars for a fully-fledged system.

They can include diagnostic options to accurately pin- point faults such as open circuits and shorts and may also offer schematic or layout viewers to depict the fault in a graphical manner.

Tests developed with such tools are frequently combined with other test systems such as in-circuit testers (ICTs) or functional board test systems.

Design-for-testability techniques improve the controllability and observability of internal nodes, so that embedded functions can be tested.

Two basic properties determine the testability of a node: 1) **controllability**, which is a measure of the difficulty of setting internal circuit nodes to 0 or 1 by assigning values to primary inputs (PIs), and 2) **observability**, which is a measure of the difficulty of propagating a node's value to a primary output (PO) . A node is said to be testable if it is easily controlled and observed. For sequential circuits, some have added **predictability**, which represents the ability to obtain known output values in response to given input stimuli. The factors affecting predictability include initializability, races, hazards, oscillations, etc. DFT techniques include analog test busses and scan methods. Testability can also be improved with BIST circuitry, where signal generators and analysis circuitry are implemented on chip. Without testability, design flaws may escape detection until a product is in the hands of users; equally, operational failures may prove difficult to detect and diagnose.

Traditionally, hardware designers and test engineers have focused on proving the correct manufacture of a design and on locating and repairing field failures. They have developed several highly structured and effective solutions to this problem, including scan design and self test. Design verification has been a less formal task, based on the designer's skills. However, designers have found that structured design-for-test features aiding manufacture and repair can significantly simplify design verification. These features reduce verification cycles from weeks to days in some cases.

In contrast, software designers and test engineers have targeted design validation and verification. Unlike hardware, software does not break during field use. Design errors, rather than incorrect replication or wear out, cause operational bugs. Efforts have focused on improving specifications and programming styles rather than on adding explicit test facilities. For example, modular design, structured programming, formal specification, and object orientation have all proven effective in simplifying test.

Although these different approaches are effective when we can cleanly separate a design's hardware and software parts, problems arise when boundaries blur. For example, in the early design stages of a complex system, we must define system level test strategies. Yet, we may not have decided which parts to implement in hardware and which in software. In other cases, software running on general-purpose hardware may initially deliver certain functions that we subsequently move to firmware or hardware to improve performance.

Designers must ensure a testable, finished design regardless of implementation decisions. Supporting hardware-software codesign' requires "cotesting" techniques, which draw hardware and software test techniques together into a cohesive whole.

Design for Testability Techniques

Design for testability (DFT) refers to those design techniques that make the task of subsequent testing easier. There is definitely no single methodology that solves all embedded system-testing problems. There also is no single DFT technique, which is effective for all kinds of circuits. DFT techniques can largely be divided into two categories, i.e., *ad hoc* techniques and *structured* (systematic) techniques.

DFT methods for digital circuits:

- ☐ Ad-hoc methods
- Structured methods:
  - *Scan*
  - *Partial Scan*
  - *Built-in self-test*
  - *Boundary scan*

Ad-hoc DFT methods

Good design practices learnt through experience are used as guidelines for ad-hoc DFT. Some important guidelines are given below.

Things to be followed

Large circuits should be partitioned into smaller sub-circuits to reduce test costs. One of the most important steps in designing a testable chip is to first *partition* the chip in an appropriate way such that for each functional module there is an effective (DFT) technique to test it.

Partitioning must be done at every level of the design process, from architecture to circuit, whether testing is considered or not. Partitioning can be functional (according to functional module boundaries) or physical (based on circuit topology). Partitioning can be done by using multiplexers and/or scan chains.

- ☐ Test access points must be inserted to enhance controllability & observability of the circuit. Test points include control points (CPs) and observation points (OPs). The CPs are active test points, while the OPs are passive ones. There are also test points, which are both CPs and OPs. Before exercising test through test points that are not PIs and POs, one should investigate into additional requirements on the test points raised by the use of test equipments.
- ☐ Circuits (flip-flops) must be easily initializable to enhance predictability. A power-on reset mechanism controllable from primary inputs is the most effective and widely used approach.
- ☐ Test control must be provided for difficult-to-control signals.
- ☐ Automatic Test Equipment (ATE) requirements such as pin limitation, tri-stating, timing resolution, speed, memory depth, driving capability, analog/mixed-signal support, internal/boundary scan support, etc., should be considered during the design process to avoid delay of the project and unnecessary investment on the equipments.
- ☐ Internal oscillators, PLLs and clocks should be disabled during test. To guarantee tester synchronization, internal oscillator and clock generator circuitry should be isolated during the test of the functional circuitry. The internal oscillators and clocks should also be tested separately.
- ☐ Analog and digital circuits should be kept physically separate. Analog circuit testing is very much different from digital circuit testing. Testing for analog circuits refers to real measurement, since analog signals are continuous (as opposed to discrete or logic signals in digital circuits). They require different test equipments and different test methodologies. Therefore they should be tested separately.

Things to be avoided

☐ Asynchronous(unclocked) logic feedback in the circuit must be avoided. A feedback in the combinational logic can give rise to oscillation for certain inputs. Since no clocking is employed, timing is continuous instead of discrete, which makes tester synchronization virtually impossible, and therefore only functional test by application board can be used.

- ☐ Monostables and self-resetting logic should be avoided. A monostable (one-shot) multivibrator produces a pulse of constant duration in response to the rising or falling transition of the trigger input. Its pulse duration is usually controlled externally by a resistor and a capacitor (with current technology, they also can be integrated on chip). One-shots are used mainly for 1) pulse shaping, 2) switch-on delays, 3) switch-off delays, 4) signal delays. Since it is not controlled by clocks, synchronization and precise duration control are very difficult, which in turn reduces testability by ATE. Counters and dividers are better candidates for delay control.

- ☐ Redundant gates must be avoided.
- ☐ High fanin/fanout combinations must be avoided as large fan-in makes the inputs of the gate difficult to observe and makes the gate output difficult to control.
- ☐ Gated clocks should be avoided. These degrade the controllability of circuit nodes.

The above guidelines are from experienced practitioners. These are not complete or universal. In fact, there are drawbacks for these methods:

A There is a lack of experts and tools.

B Test generation is often manual

C This method cannot guarantee for high fault coverage.

D It may increase design iterations.

E This is not suitable for large circuits

Scan Design Approaches for DFT

Objectives of Scan Design

- ☐ Scan design is implemented to provide controllability and observability of internal state variables for testing a circuit.
- ☐ It is also effective for circuit partitioning.
- ☐ A scan design with full controllability and observability turns the sequential test problem into a combinational one.

Scan Design Requirements

Circuit is designed using pre-specified design rules.

Test structure (hardware) is added to the verified design.

One (or more) *test control* (TC) pin at the primary input is required.

Flip-flops are replaced by *scan flip-flops* (SFF) and are connected so that they behave as a shift register in the test mode. The output of one SFF is connected to the input of next SFF.
The input of the first flip-flop in the chain is directly connected to an input pin (denoted as SCANIn), and the output of the last flip-flop is directly connected to an output pin (denoted as SCANOUT).

In this way, all the flip-flops can be loaded with a known value, and their value can be easily accessed by shifting out the chain. Figure 39.1 shows a typical circuit after the scan insertion operation.

Input/output of each scan shift register must be available on PI/PO.

- ☐ Combinational ATPG is used to obtain tests for all testable faults in the combinational logic.
- ☐ Shift register tests are applied and ATPG tests are converted into scan sequences for use in manufacturing test.

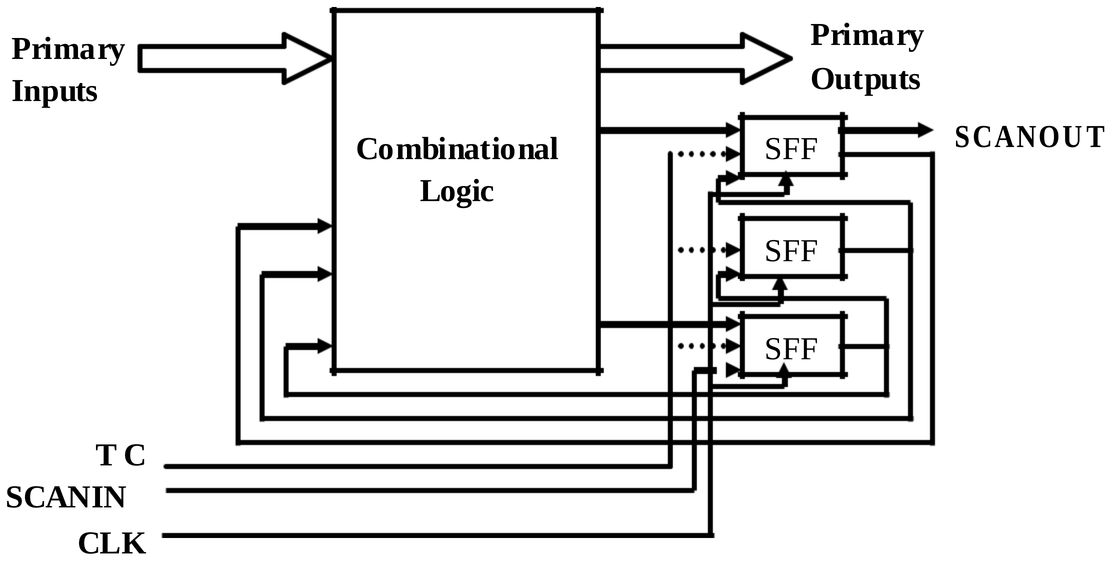


Fig. 39.1 Scan structure to a design

Fig. 39.1 shows a scan structure connected to design. The scan flip-flips (FFs) must be interconnected in a particular way. This approach effectively turns the sequential testing problem into a combinational one and can be fully tested by compact ATPG patterns. Unfortunately, there are two types of overheads associated with this technique that the designers care about very much. These are the hardware overhead (including three extra pins, multiplexers for all FFs, and extra routing area) and performance overhead (including multiplexer delay and FF delay due to extra load).

Scan Design Rules

- ☐ Only clocked D-type master-slave flip-flops for all state variables should be used.
- ☐ At least one PI pin must be available for test. It is better if more pins are available.
- ☐ All clock inputs to flip-flops must be controlled from primary inputs (PIs). There will be no gated clock. This is necessary for FFs to function as a scan register.
- ☐ Clocks must not feed data inputs of flip-flops. A violation of this can lead to a race condition in the normal mode.

Scan Overheads

The use of scan design produces two types of overheads. These are area overhead and performance overhead. The scan hardware requires extra area and slows down the signals.

II. **IO pin overhead:** At least one primary pin necessary for test.

JJ. **Area overhead:** *Gate overhead* = [4 $n_{sff}/(n_g+10n_{ff})$] x 100%, where n_g = number of combinational gates; n_{ff} = number of flip-flops; n_{sff} = number of scan flip-flops; For full scan number of scan flip-flops is equal to the number of original circuit flip-flops. Example: n_g = 100k gates, n_{ff} = 2k flip-flops, overhead = 6.7%. For more accurate estimation scan wiring and layout area must be taken into consideration.
Performance overhead: The multiplexer of the scan flip-flop adds two gate-delays in combinational path. Fanouts of the flip-flops also increased by 1, which can increase the clock period.

Scan Variations

There have been many variations of scan as listed below, few of these are discussed here.

- ☐ MUXed Scan
- ☐ Scan path
- ☐ Scan-Hold Flip-Flop
- ☐ Serial scan
- ☐ Level-Sensitive Scan Design (LSSD)
- ☐ Scan set
- ☐ Random access scan

MUX Scan

It was invented at Stanford in 1973 by M. Williams & Angell.

In this approach a MUX is inserted in front of each FF to be placed in the scan chain.

The scan flip-flips (FFs) must be interconnected in a particular way. This approach effectively turns the sequential testing problem into a combinational one and can be fully tested by compact ATPG patterns.

There are two types of overheads associated with this method. The hardware overhead due to three extra pins, multiplexers for all FFs, and extra routing area. The performance overhead includes multiplexer delay and FF delay due to extra load.

Scan Path

- ☐ This approach is also called the Clock Scan Approach.
- ☐ It was invented by Kobayashi *et al.* in 1968, and reported by Funatsu *et al.* in 1975, and adopted by NEC.
- ☐ In this approach multiplexing is done by two different clocks instead of a MUX.
- ☐ It uses two-port raceless D-FFs as shown in Figure 39.3. Each FF consists of two latches operating in a master-slave fashion, and has two clocks (C1 and C2) to control the scan input (SI) and the normal data input (DI) separately.
- ☐ The two-port raceless D-FF is controlled in the following way:

For normal mode operation C2 = 1 to block SI and C1 = 0 →1 to load DI.

For shift register test mode C1 = 1 to block DI and C2 = 0 →1 to load SI.

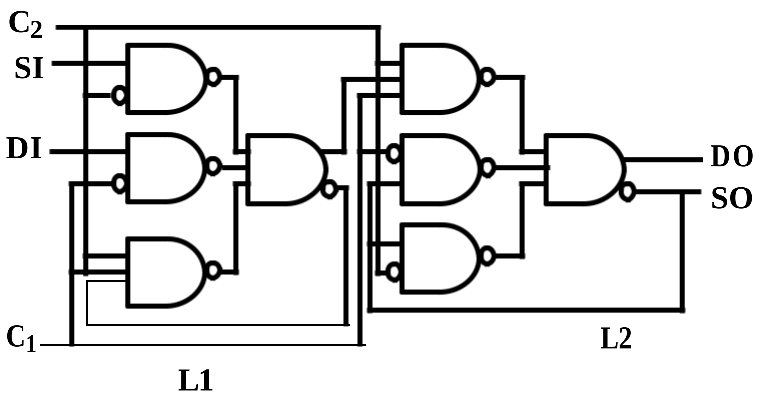


Fig. 39.3 Logic diagram of the two-port raceless D-FF

(b) This approach gives a lower hardware overhead (due to dense layout) and less performance penalty (due to the removal of the MUX in front of the FF) compared to the MUX Scan Approach. The real figures however depend on the circuit style and technology selected, and on the physical implementation.

Level-Sensitive Scan Design (LSSD)

- This approach was introduced by Eichelberger and T. Williams in 1977 and 1978.
- It is a latch-based design used at IBM.
- It guarantees race-free and hazard-free system operation as well as testing.
- It is insensitive to component timing variations such as rise time, fall time, and delay. It is faster and has a lower hardware complexity than SR modification.
- It uses two latches (one for normal operation and one for scan) and three clocks. Furthermore, to enjoy the luxury of race-free and hazard-free system operation and test, the designer has to follow a set of complicated design rules.
- A logic circuit is *level sensitive* (LS) iff the steady state response to any allowed input change is independent of the delays within the circuit. Also, the response is independent of the order in which the inputs change

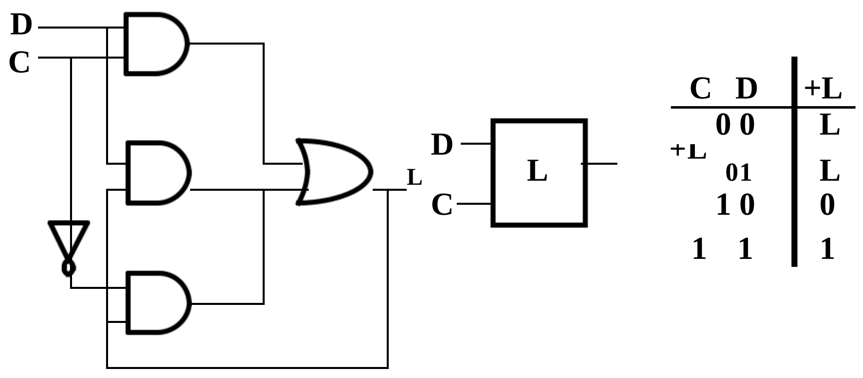


Fig. 39.4 A polarity-hold latch

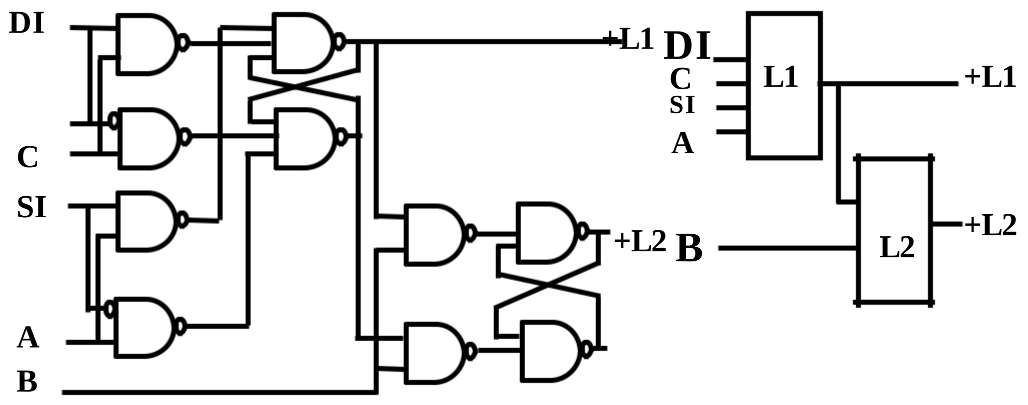


Fig. 39.5 The polarity-hold shift-register latch (SRL)

LSSD requires that the circuit be LS, so we need LS memory elements as defined above. Figure 39.4 shows an LS polarity-hold latch. The correct change of the latch output (*L*) is not dependent on the rise/fall time of *C*, but only on *C* being `1' for a period of time greater than or equal to data propagation and stabilization time. Figure 39.5 shows the polarity-hold shift-register latch (SRL) used in LSSD as the scan cell.

The scan cell is controlled in the following way:

- Normal mode: A=B=0, C=0 → 1.
- SR (test) mode: C=0, AB=10→ 01 to shift SI through L_1 and L_2.

Advantages of LSSD

Correct operation independent of AC characteristics is guaranteed.
FSM is reduced to combinational logic as far as testing is concerned.
Hazards and races are eliminated, which simplifies test generation and fault simulation.

Drawbacks of LSSD

Complex design rules are imposed on designers. There is no freedom to vary from the overall schemes. It increases the design complexity and hardware costs (4-20% more hardware and 4 extra pins).

Asynchronous designs are not allowed in this approach.

Sequential routing of latches can introduce irregular structures.

Faults changing combinational function to sequential one may cause trouble, e.g., bridging and CMOS stuck-open faults.

Test application becomes a slow process, and normal-speed testing of the entire test sequence is impossible.

It is not good for memory intensive designs.

Random Access Scan

1. This approach was developed by Fujitsu and was used by Fujitsu, Amdahl, and TI.
2. It uses an address decoder. By using address decoder we can select a particular FF and either set it to any desired value or read out its value. Figure 39.6 shows a random access structure and Figure 39.7 shows the RAM cell [1,6-7].

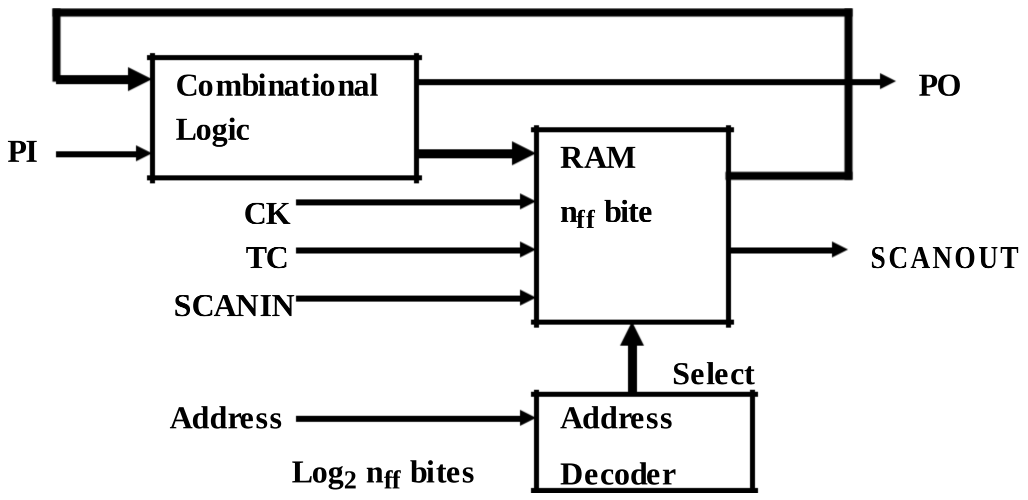


Fig. 39.6 The Random Access structure

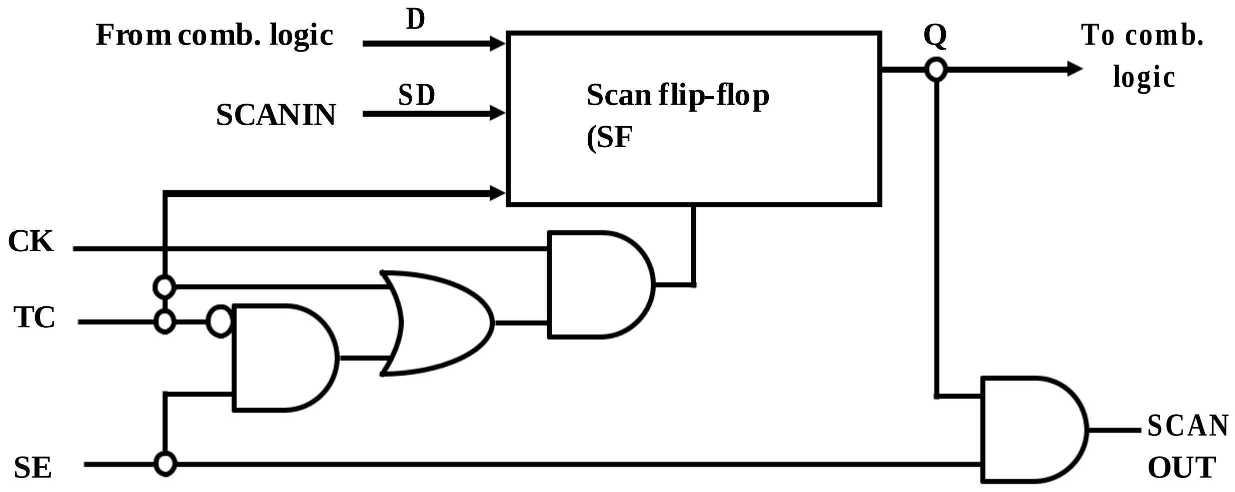


Fig. 39.7 The RAM cell

- ☐ The difference between this approach and the previous ones is that the state vector can now be accessed in a random sequence. Since neighboring patterns can be arranged so that they differ in only a few bits, and only a few response bits need to be observed, the test application time can be reduced.
- ☐ In this approach test length is reduced.
- ☐ This approach provides the ability to `watch' a node in normal operation mode, which is impossible with previous scan methods.
- ☐ This is suitable for delay and embedded memory testing.
- ☐ The major disadvantage of the approach is high hardware overhead due to address decoder, gates added to SFF, address register, extra pins and routing

Scan-Hold Flip-Flop

- ☐ Special type of scan flip-flop with an additional latch designed for low power testing application.
- ☐ It was proposed by DasGupta *in* Figure 39.8 shows a hold latch cascaded with the SFF.
- ☐ The control input HOLD keeps the output steady at previous state of flip-flop.
- ☐ For HOLD = 0, the latch holds its state and for HOLD = 1, the hold latch becomes transparent.
- ☐ For normal mode operation, TC = HOLD =1 and for scan mode, TC = 1 and Hold = 0.
- ☐ Hardware overhead increases by about 30% due to extra hardware the hold latch.
- ☐ This approach reduces power dissipation and isolate asynchronous part during scan.
- ☐ It is suitable for delay test

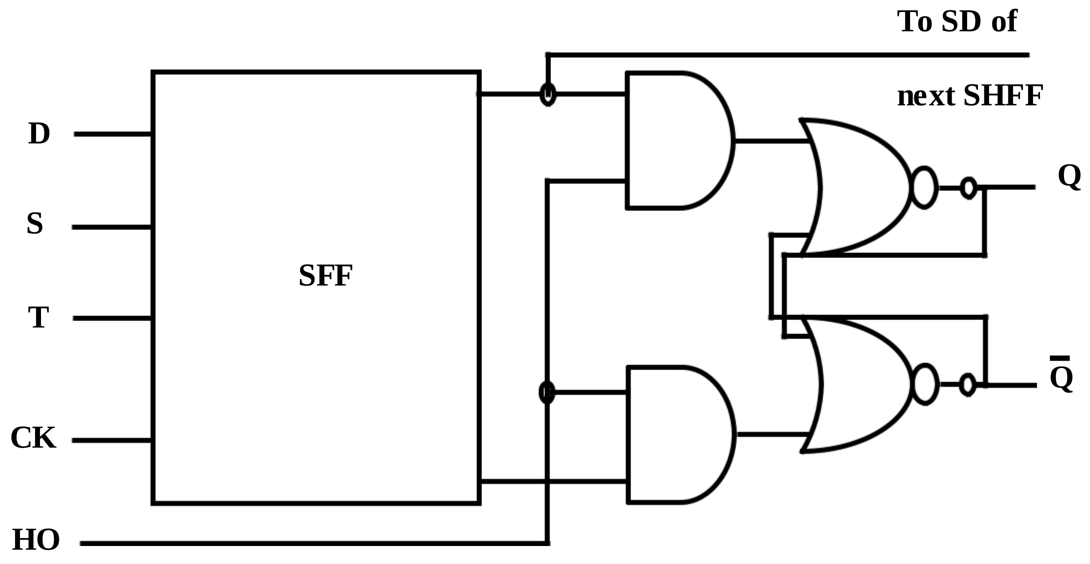


Fig. 39.8 Scan-hold flip-flop (SHFF)

Partial Scan Design

- ☐ In this approach only a subset of flip-flops is scanned. The main objectives of this approach are to minimize the area overhead and scan sequence length. It would be possible to achieve required fault coverage
- ☐ In this approach sequential ATPG is used to generate test patterns. Sequential ATPG has number of difficulties such as poor initializability, poor controllability and observability of the state variables etc. Number of gates, number of FFs and sequential depth give little idea regarding testability and presence of cycles makes testing difficult. Therefore sequential circuit must be simplified in such a way so that test generation becomes easier.
- ☐ Removal of selected flip-flops from scan improves performance and allows limited scan design rule violations.
- ☐ It also allows automation in scan flip-flop selection and test generation
- ☐ Figure 39.9 shows a design using partial scan architecture [1].
- ☐ Sequential depth is calculated as the maximum number of FFs encountered from PI line to PO line.

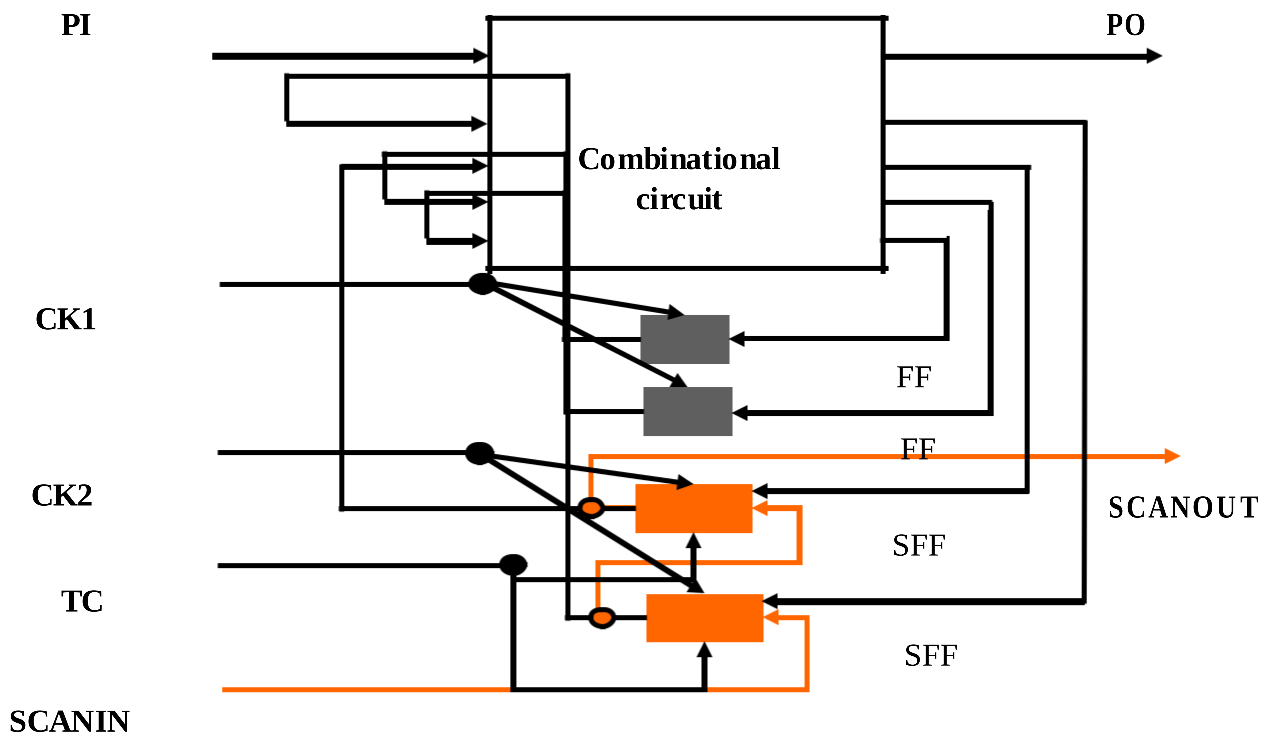


Fig. 39.9 Design using partial scan structure

Things to be followed for a partial scan method

1. A minimum set of flip-flops must be selected, removal of which would eliminate all cycles.
2. Break only the long cycles to keep overhead low.
3. All cycles other than self-lops should be removed.

Conclusions

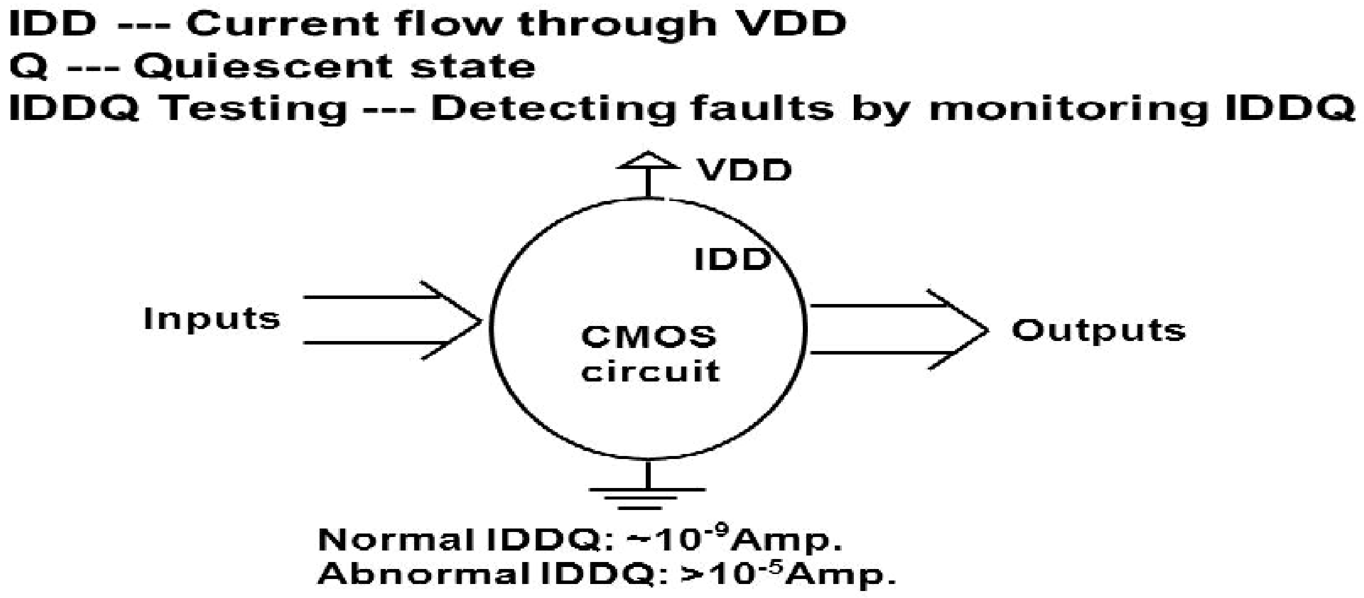
Accessibility to internal nodes in a complex circuitry is becoming a greater problem and thus it is essential that a designer must consider how the IC will be tested and extra structures will be incorporated in the design.
Scan design has been the backbone of design for testability in the industry for a long time.
Design automation tools are available for scan insertion into a circuit which then generate test patterns.
Overhead increases due to the scan insertion in a circuit. In ASIC design 10 to 15 % scan overhead is generally accepted.

IDDQ Testing:

Iddq testing is a method for testing CMOS integrated circuits for the presence of manufacturing faults. It relies on measuring the supply current (Idd) in the quiescent state (when the circuit is not switching and inputs are held at static values). The current consumed in the state is commonly called Iddq for Idd (quiescent) and hence the name.

Iddq testing uses the principle that in a correctly operating quiescent CMOS digital circuit, there is no static current path between the power supply and ground, except for a small amount of leakage. Many common semiconductor manufacturing faults will cause the current to increase by orders of magnitude, which can be easily detected. This has the advantage of checking the chip for many possible faults with one measurement. Another advantage is that it may catch faults that are not found by conventional stuck-at fault test vectors.

Iddq testing is somewhat more complex than just measuring the supply current. If a line is shorted to Vdd, for example, it will still draw no extra current if the gate driving the signal is attempting to set it to '1'. However, a different input that attempts to set the signal to 0 will show a large increase in quiescent current, signalling a bad part. Typical Iddq tests may use 20 or so inputs. Note that Iddq test inputs require only controllability, and not observability. This is because the observability is through the shared power supply connection.


IDD --- Current flow through VDD
Q --- Quiescent state
IDDQ Testing --- Detecting faults by monitoring IDDQ
VDD
IDD
Inputs
CMOS circuit
Outputs
Normal IDDQ: ~10^-9 Amp.
Abnormal IDDQ: >10^-5 Amp.

Iddq testing has many advantages:

- It is a simple and direct test that can identify physical defects.
- The area and design time overhead are very low.
- Test generation is fast.
- Test application time is fast since the vector sets are small.
- It catches some defects that other tests, particularly stuck-at logic tests, do not.

Drawback: Compared to scan chain testing, Iddq testing is time consuming, and thus more expensive, as is achieved by current measurements that take much more time than reading digital pins in mass production.

As device geometry shrinks, i.e transistors and gates become smaller resulting in larger and more complex processors and SOC's (see Moore's law), the leakage current becomes much higher and less predictable.
This makes it difficult to tell a low leakage part with a defect from a naturally high leakage part. Also, increasing circuit size means a single fault will have a lower percentage effect, making it harder for the test to detect. However, Iddq is so useful that designers are taking steps to keep it working.
One particular technique that helps is power gating, where the entire power supply to each block can be switched off using a low leakage switch. This allows each block to be tested individually or in combination, which makes the tests much easier when compared to testing the whole chip.
Iddq testing is one of the many ways to test CMOS integrated circuits in production. These circuits are usually tested as a way to find different types of manufacturing faults. Electric faults can be a major hazard and it can even lead to fatalities. This method relies on measuring the supply current (Idd) in its quiescent state (static value of a non-switching circuit).
The current that is then measured at this state is called Iddq or Idd (quiescent).

This testing method is based on the principle that there is no static current path between the power supply and the ground in a correctly operating quiescent CMOS digital circuit – except for a small amount of leakage.
It then detects the leak by picking up on any increased magnitude of the current, which is easily shown due to semiconductor manufacturing faults. It then has the upper hand of being able to check the chip for as many possible faults with only one measurement. It also works much better than conventional stuck-at fault test vectors in the sense that it picks up faults that usually go by these measurements undetected.

Even though this method is quite popular and simple, its inner workings are very complex. It goes beyond just measuring the supply current. To use an example, if a line is shortened to Vdd it will still be unable to draw extra current if the gate driving the signal is set to '1'. But a different input attempting to set the signal at '0' will show an increase in quiescent current that will indicate a bad part in the electrical stream. A typical Iddq test will use about 20 inputs. These test inputs need only controllability and not necessarily observability. The reason for this is that observability takes place through the shared power connection.

The advantages of Iddq are far greater than anyone could have ever imagined. Firstly, it is a simple and direct test that can identify physical defects more effectively than standardised equipment or methods. Secondly, the time period attached to it isn't very demanding. What this means is that the design time and area overhead are relatively low. The test generation is fast, the test application time is fast due to the small sets in vectors, and it catches underlying effects that other tests can't pick up on immediately.

One disadvantage of Iddq testing is that it can be time consuming if compared to methods like scan testing. It is also a more expensive option, comparatively speaking. The reason for this is because it is achieved by current measurements that take much more time than reading digital pins in mass production.

Design for Manufacturability - An Overview
Introduction:
Aggressive ground rule changes continue to increase the complexity of semiconductor technology. The requirements for designs, processes, equipment, and facilities all grow in sophistication from generation to generation. These trends have made it increasingly difficult to produce a technology in the development laboratory and transfer it to volume manufacturing in a timely and cost effective manner. The traditional laboratory role of design and process development has expanded to include a parallel responsibility for manufacturability. For many companies, design for manufacture (DFM) has become a critical strategy for survival in an increasingly competitive global marketplace.
DFM is a systems approach to improving the competitiveness of a manufacturing enterprise by developing products that are easier, faster, and less expensive to make, while maintaining required standards of functionality, quality, and marketability. Design for manufacturability (DFM) and early manufacturing involvement (EMI) concepts are now major components of the development effort designed to maintain and enhance the rate of technology advancement and significantly improve the development-to-manufacturing transition. Design-for-manufacturability philosophy and practices are used in many companies because it is recognized that 70% to 90% of overall product cost is determined before a design is ever released into manufacturing. The semiconductor industry continues to grow in both complexity and competitiveness.
Problem Statement
The layout development is most critical in integrated circuits (IC's) design because of cost, since it involves expensive tools and a large amount of human intervention, and also because of the consequences for production cost. As the device size is shrinking, the landscape of technology developments has become very different from the past. The problems, which were supposed to be secondary can cause of yield drop out in submicron technologies. The variability becomes a critical issue not only for performance, but also for yield dropout.
Yield dropout due to given below defects.
1. Random Defects: Due to form of impurities in the silicon itself, or the introduction of a dust particle that land on the wafer during processing. These defects can cause a metal open or shorts. As feature sizes continue to shrink, random defects have not decreased accordingly making advanced IC's even more susceptible to this type of defect.

2. Systematic Defects: Again systematic defects are more prominent contributor in yield loss in deep submicron process technologies. Systematic defects are related to process technology due to limitation of lithography process which increased the variation in desired and printed patterns. Another aspects of process related problem is planarity issues make layer density requirements necessary because areas with a low density of a particular layer can cause upper layers to sag, resulting in discontinuous planarity across the chip.

3. Parametric Defects: In deep submicron technology parametric defects is most critical for us. Parametric defects come into the picture due to improper modeling of interconnects parasitic.

As a result manufactured device does not match the expected result from design simulation and does not meet the design specification.

Design for manufacturability (DFM) is process to overcome these defects of yield drop out. The DFM will not be done without collaborations between various technology parties, such as process, design, mask, EDA, and so on. The DFM will give us a big challenge and opportunity in nanometer era.

DESIGN FOR MANUFACTURABILITY:

Design for Manufacturability is the proactive process which ensures the quality, reliability, cost effective and time to market.
DFM consist a set of different methodologies trying to enforce some soft (recommended/Mandatory) design rules regarding the shapes and polygons of the physical layout which improve the yield.
Given a fixed amount of available space in a given layout area, there are potentially multiple yield enhancing changes that can be made.

There are some DFM guidelines which we can take into account at SOC level.

1. Filler cell (consisting regular Diffusion and Poly silicon structures) insertion and shielding
Issue Addressed: PO/OD non uniformity
Benefit: Higher parametric yield.
2. Via optimization
Issue Addressed: open Via's, systematic via opening issue
Benefit: Higher yield after manufacturing and qualification.
3. Wire Spreading
Issue Addressed: wire shorts and opening due to defectivity.
Benefit: Higher yield, decrease cross talk.
4. Power/ground-connected fill
Issue Addressed: Density gradients, Large IR drop, Layout becomes regular
Benefit: Robustness to IR drop
5. Litho hotspot detection and repair
Issue Addressed: Lithography hotspots
Benefit: Higher yield
6. Dummy Metal/Via/FEOL
Issue Addressed: Large density gradients
Benefit: Higher yield
7. CMP hotspot detection
Issue Addressed: CMP hotspots
Benefit: Higher yield