DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

IMPORTANT QUESTIONS AND ANSWERS

Subject Code: **EC8095**

Subject Name: **VLSI DESIGN**

Sem / Year: **VI/III**

Regulation: 2017 **Year and Semester: III /VI**

ANNA UNIVERSITY, CHENNAI-25
SYLLABUS COPY REGULATION
2017

EC8095 VLSI DESIGN L T P C3 0 0 3

UNIT I MOS TRANSISTOR PRINCIPLE 9

NMOS and PMOS transistors, Process parameters for MOS and CMOS, Electrical properties of
CMOS circuits and device modeling, Scaling principles and fundamental limits, CMOS inverter scaling, propagation delays, Stick diagram, Layout diagrams

UNIT II COMBINATIONAL LOGIC CIRCUITS 9

Examples of Combinational Logic Design, Elmore's constant, Pass transistor Logic, Transmission gates, static and dynamic CMOS design, Power dissipation – Low power design principles

UNIT III SEQUENTIAL LOGIC CIRCUITS 9

Static and Dynamic Latches and Registers, Timing issues, pipelines, clock strategies, Memory architecture and memory control circuits, Low power memory circuits, Synchronous and Asynchronous design

UNIT IV DESIGNING ARITHMETIC BUILDING BLOCKS 9

Data path circuits, Architectures for ripple carry adders, carry look ahead adders, High speed adders, accumulators, Multipliers, dividers, Barrel shifters, speed and area tradeoff

UNIT V IMPLEMENTATION STRATEGIES 9

Full custom and Semi custom design, Standard cell design and cell libraries, FPGA building block architectures, FPGA interconnect routing procedures.

TOTAL: 45 PERIODS

TEXTBOOKS:

1. Jan Rabaey, Anantha Chandrakasan, B.Nikolic, “Digital Integrated Circuits: A Design Perspective”, Second Edition, Prentice Hall of India, 2003.
2. M.J. Smith, “Application Specific Integrated Circuits”, Addisson Wesley, 1997

REFERENCES:

1. N.Weste, K.Eshraghian, “Principles of CMOS VLSI Design”, Second Edition, Addision Wesley
1993
2. R.Jacob Baker, Harry W.LI., David E.Boyee, “CMOS Circuit Design, Layout and Simulation”,
Prentice Hall of India 2005
3. A.Pucknell, Kamran Eshraghian, “BASIC VLSI Design”, Third Edition, Prentice Hall of India, 2007.

EC8095 **VLSI DESIGN** **L T P C3 0 0 3**

1. Aim and Objective of the

Subject Aim:

The aim of this course is to uncover the understandings of VLSI circuits in a clear way and to realize various applications of VLSI logic circuits ranging from combinational, sequential circuit to design implementation.

Objectives:

- In this course, the MOS circuit realization of the various building blocks that is common to any microprocessor or digital VLSI circuit is studied.
- Architectural choices and performance tradeoffs involved in designing and realizing the circuits in CMOS technology are discussed.
- The main focus in this course is on the transistor circuit level design and realization for digital operation and the issues involved as well as the topics covered are quite distinct from those encountered in courses on CMOS Analog IC design.

2. Need and Importance for Study of the Subject

This subject involves packing more and more logic devices into smaller and smaller areas. Via VLSI, circuits that would have taken boardful of space can now be fitted into a small space of few millimeters. This has opened avenues to do things that were not possible before.

In simple words, VLSI circuits are everywhere from your computer to your car, your brand new state-of-the-art digital camera, cellphones, and whatever electronics item you have. By studying this subject student can able to design digital logic circuits with reduced area, high speed, and low power.

3. Industry Connectivity and Latest Developments

There are a variety of career opportunities in product companies, design services companies and electronic design automation (EDA) companies like Virtuso, Keltron, Cadence, CDAC, etc.

Product and application domains of VLSI include mobile and consumer electronics, computing, telecommunications and networking, data processing, automotive, healthcare and industrial applications. Being a fast-changing technology area, VLSI design is an extremely challenging and creative sector that offers exciting opportunities and fast growth for engineers. GSI, ULSI, FPAA, 10 nm process technology(by INTEL) are the latest development in this field.

.

Department of Electronics and Communication

Engineering Detailed Lesson Plan

Name of the Subject& Code: **EC8095 & VLSI DESIGN**

TEXTBOOKS:

1. Jan Rabaey, Anantha Chandrakasan, B.Nikolic, “Digital Integrated Circuits: A Design Perspective”, Second Edition, Prentice Hall of India, 2003. **(Copies not Available in Library)**

2. M.J. Smith, “Application Specific Integrated Circuits”, Addisson Wesley, 1997**(Copies Available in Library: YES)**

REFERENCES:

1. N.Weste, K.Eshraghian, “Principles of CMOS VLSI Design”, Second Edition, Addision Wesley 1993**(Copies not Available in Library)**

2. R.Jacob Baker, Harry W.LI., David E.Boyee, “CMOS Circuit Design, Layout and Simulation”, Prentice Hall of India 2005**(Copies Available in Library: YES)**

3. A.Pucknell, Kamran Eshraghian, “BASIC VLSI Design”, Third Edition, Prentice Hall of India, 2007. **(Copies Available in Library: YES)**

Sl . N o	**Unit**	**Topic / Portions to be Covered**	**Hours Required / Planned**	**Cumulative Hrs**	**Books Referred**
		UNIT I- MOS TRANSISTOR PRINCIPLE			
1	1	NMOS and PMOS transistors	1	1	R1
2	1	Process parameters for MOS and CMOS	1	2	Notes
3	1	Electrical properties of CMOS circuits and device modeling	2	4	R3
4	1	Scaling principles	1	5	T1,R1
5	1	Fundamental limits	1	6	R3
6	1	CMOS inverter scaling	1	7	T1
7	1	Propagation delays	1	8	T1
8	1	Stick diagram	1	9	R3
9	1	Layout diagrams	1	10	R1,R3
		UNIT II- COMBINATIONAL LOGIC CIRCUITS			
10	2	Examples of Combinational Logic Design	2	12	Notes
11	2	Elmore’s constant	1	13	R1
12	2	Pass transistor Logic	1	14	T1,R1
13	2	Transmission gates	1	15	T1
14	2	Static CMOS design	1	16	T1
15	2	Dynamic CMOS design	2	18	R1
16	2	Power dissipation	1	19	R1
17	2	Low power design principles	1	20	R1

UNIT III- SEQUENTIAL LOGIC CIRCUITS					
18	3	Static Latches and Registers	1	21	T1,R1
19	3	Dynamic Latches and Registers	1	22	T1,R1
20	3	Timing issues	1	23	R1
21	3	Pipelines	1	24	T1
22	3	Clock strategies	1	25	T1
23	3	Memory architecture and memory control circuits	3	28	T1
24	3	Low power memory circuits	1	29	T2, WEB
25	3	Synchronous and Asynchronous design	1	30	T1
UNIT IV- DESIGNING ARITHMETIC BUILDING BLOCKS					
26	4	Data Path Circuits	1	31	T1
27	4	Architectures For Ripple Carry Adders	1	32	T1
28	4	Carry Look Ahead Adders	1	33	T1
29	4	High Speed Adders	1	34	R4
30	4	Accumulators	1	35	NOTES
31	4	Multipliers	2	37	T1
32	4	Dividers	2	39	R1
33	4	Barrel Shifters	1	40	T1
34	4	Speed And Area Tradeoff	1	41	T1
UNIT V- IMPLEMENTATION STRATEGIES					
35	5	Full custom design	1	42	T2
36	5	Semi custom design	2	44	T2
37	5	Standard cell design	1	45	T2
38	5	Cell Libraries	1	46	T2
39	5	FPGA building block architectures	2	48	T2
40	5	FPGA interconnect routing procedures	2	50	T2

UNIT I
MOS TRANSISTOR PRINCIPLE
PART A

1. **What are the two types of design rules? (Apr/May 2010)**
   - Micron rules
   - Lambda rules

2. **What is body effect ?(Apr/May2010),(Nov/Dec 2010)**

   The resultant effect increases the channel substrate junction potential. This increases the rate-channel voltage drop. The overall effect is an increase in threshold voltage. This effort is called body effect.

3. **What is body effect coefficient? (Apr/May 2011)**

   The potential difference between the source and body affects the threshold voltage. The threshold voltage can be modeled as

   $Vt=Vt_0+\gamma((\Phi s+Vsb)^{1/2}-(\Phi s)^{1/2}$

   Where, Φs= surface potential at threshold
   γ= body effect coefficient

4. **What is the influence of voltage scaling on power and delay? (AprMay 2011)**

   Constant voltage scaling increased the electric field in devices. By the 1µm generation velocity saturation was severe enough that decreasing feature size no longer improved device current. Aggressive process achieve delays in the short end of the range by building transistors with effective channel length.

5. **Determine whether an NMos transistor with a threshold voltage of 0.7v is operating in the saturation region if Vgs=2V and Vds=3V.(Nov/Dec 2011)**

   Vt=0.7 Vgs=2V Vd=3V
   NMos transistor is in saturation region if
   Vds > Vgs-Vt
   3V > 2V-0.7V
   3V > 1.3V
   It is in saturation region

6. **Write down the equation for describing the channel length modulation effect in NMos transistor. (MAY/JUN 2016)**
   - Ideally Ids is independent of V_{ds} in saturation.
   - The reverse biased p-n junction between the drain and body forms a depletion region with a width L_d that increases with V_{db}.
   - The depletion region effectively shortens the channel length to L_{eff} =L-Ld.
   - Imagine that the source voltage is close to the body voltage. Increasing V_{ds} decreases the effective channel length.

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

7. **Write the expression for the logical effort and parasitic delay of an input NOR gate. (Nov/Dec 2011)**

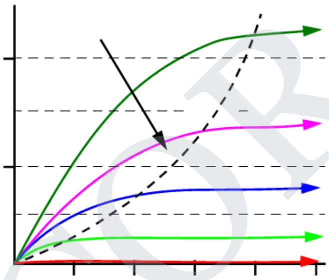
$$I_{ds} = \beta \frac{(V_{gs} - V_t)^2}{2} (1+ \lambda V_{ds}$$

Logical effort of n input Nor gate
G=(2n+1)/3
N=number of inputs

8. **Why does interconnect increase the circuit delay? (Nov/Dec 2011)**

The wire capacitance adds loading to each gate.
The long wire contributes RC delay or flight time.
Circuit delay can be increased by interconnect

9. **Draw the IV characteristics of Mos transistors.(MayJun 2012)**



10. **Brief the different operating regions of Mos system.(May/ Jun 2012)**

Different operating regions of Mos syatem

- Cut off or suthreshold region
- Linear or non-saturation region
- Saturation region

11. **Why the tunneling current is higher for NMos transistor than Pmos transistor with silica gate? (Nov/Dec 2012)**

Tunneling current is an order of magnitude higher for nMos than PMos transistor with Sio$_2$ gate dielectrics because the electrons tunnel from the conduction band while the holes tunnel from the valance band.

12. **What is the objective of layout rules?**

Layout design rule is examined and a scale parameter lambda is ndefined as the halfwidth of a minimum width line or as a multiple of standard deviation of a process. Designing layouts in terms of lambda allows for future scaling makes the layout portable.

13. What are the advantages of CMOS technology ?(May 2013)

a. Low power consumption.
b. High performance.
c. Scalable threshold voltage.
d. High noise margin.
e. Low output drive current.

14. Compare NMOS and PMOS ?

NMOS	PMOS
The majority carriers are electron	The majority carriers are holes
Positive voltage is applied at the gate terminal	Negative voltage is applied at the gate terminal
NMOS conducts at logic 1	PMOS conducts at logic 0
Mobility of electron is high	Mobility of electron is low
Switching speed is high	Switching speed is low

15. What is latch up? How to prevent latch up? (MAY/JUN 2016)

Latch up is a condition in which the parasitic components give rise to the establishment of low resistance conducting paths between VDD and VSS with disastrous results. Careful control during fabrication is necessary to avoid this problem.

PART-B

1. Explain about nMOS Transistor. (MAY'11)

NMos transistors are built on a p-type substrate of moderate doping. Source and drain are formed by diffusing heavily doped n-type impurities (n+)adjacent to the gate. A layer of silicon dioxide (SiO_2) or glass is place over the substrate in between the source and drain. Over SiO_2, a layer of polycrystalline silicon or polysilicon is formed, from which the gate terminal is taken.

The following figure shows the structure and symbol of nMOS transistor.

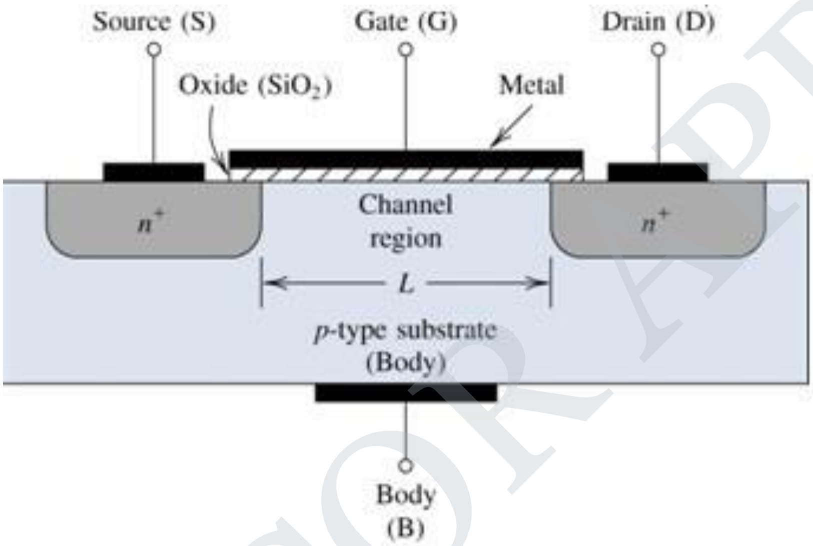


Fig: nMOS transistor.

Threshold Voltage (V_t)

It can be defined as the voltage applied between the gate and the source of a MOS device (V_{gs}) below which the drain-to-source current (I_{ds}) "effectively" drops to zero.

V_t depends on the following:

- Gate conduction material
- Gate insulation material
- Gate insulator thickness
- Channel doping
- Impurities at the silicon-insulator interface
- Voltage between the source and the substrate, V_{sb}.

Modes of operation of MOS Transistor:

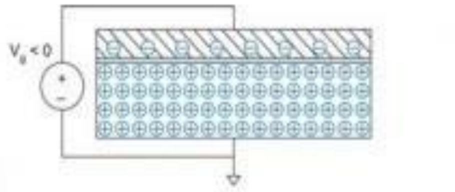
The following are the three modes of operation of nMOS transistor:

1. Accumulation mode
2. Depletion mode

3. Inversion mode

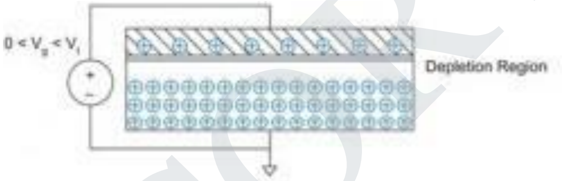
a. Accumulation Mode

In this mode a negative voltage is applied to the gate. So there is negative charge on the gate. The mobile positively charged holes are attracted to the region beneath the gate.



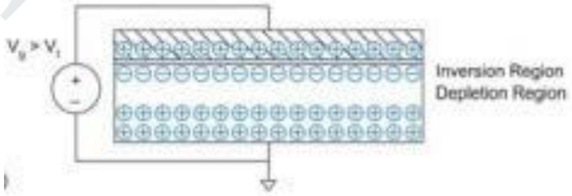
b. Depletion Mode:

In this mode a low positive voltage is applied to the gate. This results in some positive charge on the gate. The holes in the body are repelled from the region directly beneath the gate.



c. Inversion Mode:

In this mode, a higher positive potential exceeding a critical threshold voltage is applied. This attracts more positive charge to the gate. The holes are repelled further and a small number of free electrons in the body are attracted to the region beneath the gate. This conductive layer of electrons in the p-type body is called the inversion layer.



Behavior of nMOS with different voltages:

The Behavior of nMOS with different voltages can be classified into the following three cases and is illustrated in below figure.

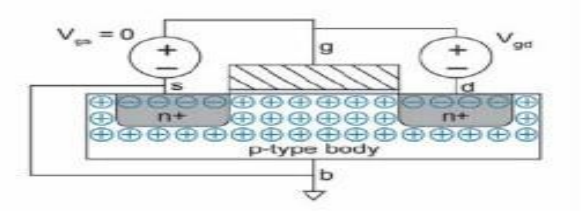
i. Cut-off region

ii. Linear region

iii. Saturation region

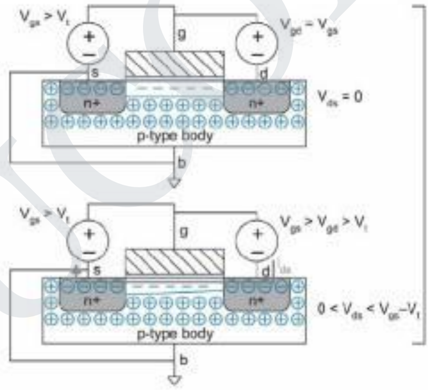
a. **Cut-off region:-**

In this region $V_{gs} < V_t$ .The source and drain have free electrons. The body has free holes but no free electrons. The junction between the body and the source or drain is reverse biased. So no current will flow. This mode of operation is called cut-off.



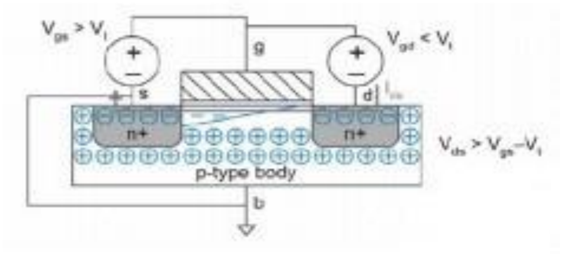
Linear region:-

In this region $V_{gs} > V_t$ .Now an inversion region of electrons called the channel connects the source and drain. This creates a conductive path between source and drain. The number of carriers and the conductivity increases with the gate voltage. The potential difference between drain and source is $V_{ds} = V_{gs} - V_{gd}$. If $V_{ds}=0$,there is no electric field tending to push current from drain to source.



b. **Saturation region:-**

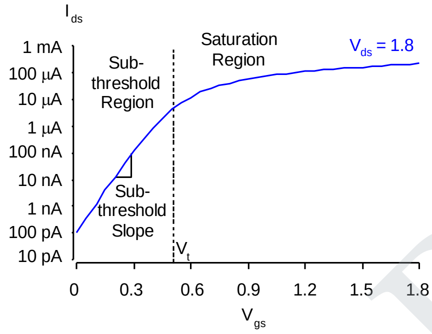
In this region V_{ds} becomes sufficiently larger than $V_{gd} < V_t$, the channel is no longer inverted near the drain and becomes pinched off .Above this drain voltage, the I_{ds} is controlled only by the gate voltage. This mode is called saturation mode.



2. a. Explain in detail about Non-ideal I-V characeteristics of p-MoS and n-MoS Transistors (MAY/JUN 2016)

<u>Non- Ideal I-V Effects:</u>

The Ids value of an ideal I- v model neglects many effects that are important to modern devices.



Simulated I-V Characteristics

- While compared to the ideal devices, the saturation current increases less than a quarter with increasing V_{gs}. This is caused by two effects.
  1) Velocity Saturation
  2) Mobility degradation.
- At high lateral field strengths $\frac{V_{ds}}{L}$. carrier velocity ceases to increase linearly with field strength. This is called velocity saturation and this results in lesser I_{ds} than expected at high V_{ds}.
- Current between source and drain is the total amount of charge in the channel divide the time required to cross it.

$$I_{ds} = \frac{Q\ Channel}{t} = \frac{Q\ Channel}{(L/v)} = \frac{Q\ Channal * V}{L}$$

By Sub the values we get

$$I_{ds} = \mu C o\chi \frac{W}{L} (V_{gs} - V_t - V_{ds}) V_{ds}$$

$$Ids = \beta (V_{gs} - V_t - \frac{V_{ds}}{2}) V_{ds}$$

Where $\beta = \mu C_{0x}$ W/L

In linear region $V_{gs} > V_t$ and V_{ds} is relatively small.

Saturation Region:-

- In saturation region , if $V_{ds} > V_{dsat}$, the channel is pinched off.

  ie; $V_{ds} = V_{gs} - V_t$

- Beyond this point it is often called the drain saturation voltage.

Sub $V_{ds} = V_{gs} - V_t$ in the Ids values for linear region we get.

$$I_{ds} = \frac{\beta}{2} (V_{gs} - V_t)^2 \text{ for } V_{ds} > V_{dsat.}$$

In saturation, I_{dsat} is

$$V_{gs} = V_{ds} + V_{DD}$$

$$I_{dsat} = (V_{DD} - V_t)^2$$

Summarizing the three regions we get.

$$I_{ds} = \begin{cases} o & ; V_{gs} <- V_t \text{ ; cut off} \\ \beta \left(V_{gs} - V_t - \frac{V_{ds}}{2} \right) V_{ds} \; ; V_{ds} < V_{dsat} & \text{; linear} \\ \frac{\beta}{2} (V_{gs} - V_t)^2 \; ; V_{ds} > V_{dsat} & \text{; saturation.} \end{cases}$$

- At high vertical field strengths V_{gs} / tor the carrier scatlers more often. This is called mobility degradation and this leads to less current than expected at high V_{gs}
- The threshold voltage itself is influenced by the voltage difference between the source and body called the body effect.

The non- ideal Iv effect include the following:-

1) Velocity saturation & mobility degradation.
2) Channel length modulation
3) Body effect
4) Sub threshold condition
5) Junction Leakage
6) Tunneling
7) Temperature dependence
8) Geometry Dependence.

Velocity saturation and mobility degradation:-

- Carrier drift velocity and current increase linearly with the lateral field E lat = V_{ds}/ L between source and drain.
- At high field strength, drift velocity rot off due to carrier scattering and eventually saturates at V_{sat} .
- Without velocity saturation the saturation current is

$$\boxed{I_{ds} = \mu C_0 \;x\; \frac{W}{L} \; \frac{(V_{gs} - V_{ds})}{2}^2}$$

- If the transistor is completely Velou saturated V = V_{sat} and saturation current become .

- $$\boxed{I_{ds} = C_0 \;x\; W \;(V_{gs} - V_t)\, V_{sat}}$$ ithout velocity saturation .

$$I_{ds} = \begin{cases} o & ; V_{gs} < V_t \quad \text{cut off} \\ I_{dsat} = \frac{V_{ds}}{V_{dsat}} & ; V_{ds} < V_{dsat} \quad \text{linear} \\ I_{dsat} & ; V_{ds} > V_{dsat} \quad \text{saturation.} \end{cases}$$

Where

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

$$V\, d_{sat} = Pv\,(V_{gs} - V_t) \propto /2 .$$

- As channel length becomes shorter, the lateral field increases and transistors become more velocity saturated, and the supply voltage is held constant.

Channel Length Modulation:-

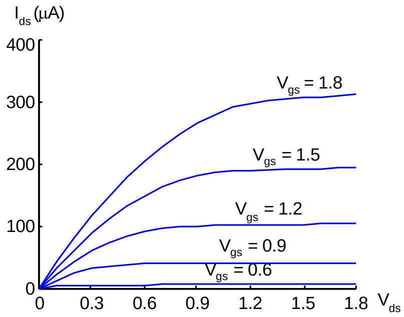
- Ideally Ids is independent of V_{ds} in saturation.
- The reverse biased p-n junction between the drain and body forms a depletion region with a width L_d that increases with V_{db}.
- The depletion region effectively shortens the channel length to Leff =L-Ld.

- Imagine that the source voltage is close to the body voltage. Increasing V_{ds} decreases the effective channel length. Shorter length results in higher current. Thus Ids increases with V_{ds} in saturation as shown below.

In saturation region

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

Where λ = Channel length modulation factor



Body Effect:

Transistor has four terminals named gate, source, drain and body. The potential difference between the source and body V_{sb} affects the threshold voltage.

$$V_t = V_{to} + ʊ \left(\sqrt{p_s + V_{sb}} - \sqrt{\varphi\ s} \right)$$

Where

V_{to} = Threshold Voltage when the source is at the body potential

φ_s = Surface Potential at threshold = $2vT \ln \frac{Nd}{Ni}$

V_{sb} = Potential difference between the source and body.

Sub threshold condition:

- Ideally current flows from source to drain when $V_{gs} > V_t$. In real transistor, current does not abrupthy cut off below threshold, but rather drops off exponentially as

$$I_{ds} = I_{dso}\, e^{\frac{V_{gs} - V_t}{nvt}} [1 - e^{\frac{V_{ds}}{V_t}}] .$$

This is also called as leakage and often thias results in underired current when a transistor is normally OFF. I_{dso} is the current at thresholo and is dependent on process and device geometry

Applications:-

- This is used in very low power analog circui
- This is used in dynamic circuits and OR AM

Advantage:

1) Leakage increases exponentially as V_t decreases or as temperature rises.

Disadvantages:

1) It becomes worse by drain induced barrier lowering in which a positive V_{ds} effectively reduces V_t. This effect is especially pronounced in short channel transistors.

Junction Leakage:

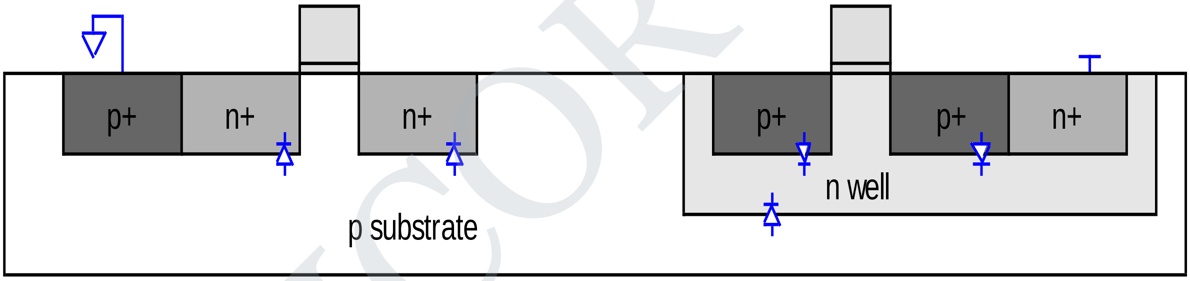
- The P-n junction between diffusion and the substrate or well form diodes are shown below.
- The substrate and well are tied to GND or V_{DD} to ensure that these diodes remain reverse biased.
- The reverse biased diodes still conduct a small amount of current I o.

$$I_D = I_s \left[e^{\frac{V_D}{V_T}} - 1 \right]$$

Where

I_D = diode current

I_s = diode reverse- biased saturation current that depends on doping levels and on the area and perimeter of the diffusion region.



Tunneling :

Based on quantum mechanics, we see that the is a finite probability that carriers will tunnel through the gate oxide. This results in gate leakage current flowing into the gate. The probability of tunnelling drops off exponentially with oxide thickness.

- Large tunnelling currents impact not only dynamic nodes but also quiescent power consumption and thus may limit oxide thickness tor.
- Tunnelling can purposely be used to create electrically erasable memory devices. Different dielectrics may have different tunneting properties.

Temperature Dependance:

Temperature influences the characteristics of transistors. Carrier mobility decreases with temperature.

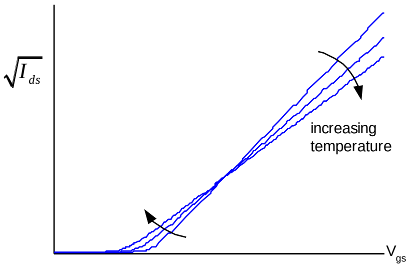
$$\mu(T) = \mu(T_r)\left(\frac{T}{Tr}\right)^{-k\mu}$$

- Junction leakage increases with temperature because. I_S is strongly temperature dependent . The combined temperature effect is shown below.

Where on current decreases and OFF current increases with temperature.

The figure below shows how the On current I_{dsat} decreases with temperature. Circuit performance is worst at high temperature, called negative temperature coefficient.

- Circuit performance an be improved by cooling . Natural convection, fans with heat sink, water cooling thin flim refrigerators, and liquid nitrogen can be used as cooling.



Advantages of Operating at low temperature:

1) velocity saturation occurs at higher fields providing more current.
2) For high mobility , power is saved.
3) Wider depletion region results in less junction capacitance.

Geometry Dependance:

- The layout designer draws transistors with width and length ^{W} drawn and L drawn. The actual gate dimensions may differ by factors X_W and X_L.
- The source and drain trends to diffuse later under the gate by L_{Di} producing a shorter effective between source and drain.

$$Leff = L_{\text{drawn}} + X_L - 2_{LP}$$

$$Weff = W_{\text{drawn}} + X_W - 2w_D$$

Long transistors experience less channel length modulation. In a process blow 0.25 μm the effective length of the transistor depends on the orientation of the transistor.

2. b.Explain in detail about the ideal I-V characteristics of a nMOS and pMOS device (NOV/DEC 2013)(MAY/JUN 2013)(NOV/DEC 2014)

MOS transistors have three regions of operation:

_ **Cutoff or subthreshold region**

_ **Linear region**

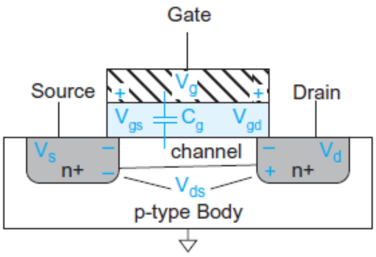
_ **Saturation region**

The current and voltage (I-V) for an nMOS transistor in each of these regions. The model assumes that the channel length is long enough that the lateral electric field (the field between source and drain) is relatively low, which is no longer the case in nanometer devices. This model is variously known as the *long-channel*, *ideal*, *first-order,* or *Shockley* model. Subsequent sections will refine the model to reflect high fields, leakage, and other nonidealities. The long-channel model assumes that the current through an OFF transistor is 0. When a transistor turns ON ($Vgs > Vt$), the gate attracts carriers (electrons) to form a channel. The electrons drift from source to drain at a rate proportional to the electric field between these regions. Thus, we can compute currents if we know the amount of charge in the channel and the rate at which it moves.

We know that the charge on each plate of a capacitor is $Q = CV$. Thus, the charge in the channel *Q*channel is

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

where Cg is the capacitance of the gate to the channel and $Vgc - Vt$ is the amount of voltage attracting charge to the channel beyond the minimum required to invert from p to n. The gate voltage is referenced to the channel, which is not grounded. If the source is at Vs and the drain is at Vd , the average is $Vc = (Vs + Vd)/2 = Vs + Vds\ /2$. Therefore, the mean difference between the gate and channel potentials Vgc is $Vg – Vc = Vgs – Vds\ /2$, as shown in Figure.



Average gate to channel potential:

$V_{gc} = (V_{gs} + V_{gd})/2 = V_{gs} – V_{ds}/2$

We can model the gate as a parallel plate capacitor with capacitance proportional to area over thickness. If the gate has length *L* and width *W* and the oxide thickness is *t*ox, as shown in below Figure, the capacitance is

$$C_g=\varepsilon ox(WL/t_{ox})=CoxWL$$

where εox is the permittivity of free space, 8.85 × 10–14 F/cm, and the permittivity of SiO2 is *k*ox = 3.9 times as great. Often, the □ox/*t*ox term is called *C*ox, the capacitance per unit area of the gate oxide.

Each carrier in the channel is accelerated to an average velocity, *v*, proportional to the lateral electric field, i.e., the field between source and drain. The constant of proportionality μ is called the *mobility*.

$$v = \mu E$$

The time required for carriers to cross the channel is the channel length divided by the carrier velocity: *L*/*v*. Therefore, the current between source and drain is the total amount of charge in the channel divided by the time required to cross

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

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

The term *Vgs* – *Vt* arises so often that it is convenient to abbreviate it as *VGT* .

$$K' = \mu Cox$$

If *Vds* > *V*dsat □ *VGT*, the channel is no longer inverted in the vicinity of the drain; we say it is pinched off. Beyond this point, called the *drain saturation voltage*, increasing the drain voltage has no further effect on current. Substituting *Vds* = *V*dsat at this point of maximum current in above eqn, we find an expression for the saturation current that is independent of *Vds*.

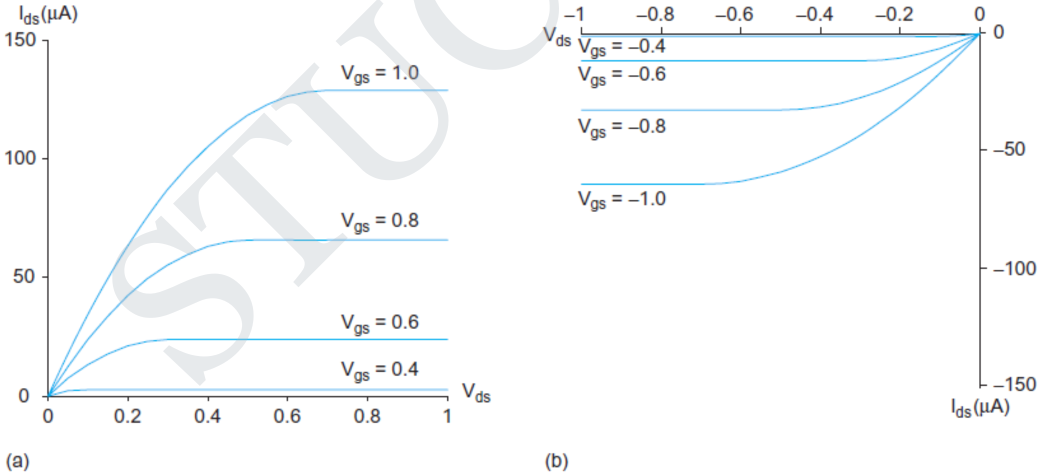
$$Ids=(\beta/2)V^2_{GT}$$

This expression is valid for *Vgs* > *Vt* and *Vds* > *V*dsat . Thus, long-channel MOS transistors are said to exhibit *square-law behavior* in saturation. Two key figures of merit for a transistor are *I*on and *I*off. *I*on (also called *I*dsat) is the ON current, *Ids*, when *Vgs* = *Vds* = *VDD*. *I*off is the OFF current when *Vgs* = 0 and *Vds* = *VDD*. According to the long-channel model, *I*off = 0 and

$$I_{on}=(\beta/2)(V_{dd}-V_t)$$

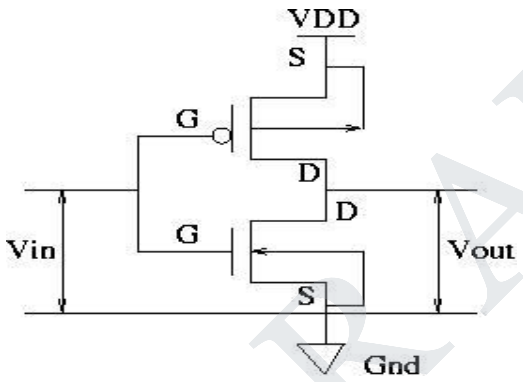
$$I_{ds} = \begin{cases} o & ; V_{gs} <- V_t \text{ ; cut off} \\ \beta \left(V_{gs} - V_t - V_{ds}/2\_ \right) V_{ds} & ; V_{ds} < V_{dsat} \text{ ; linear} \\ \frac{\beta}{2} (V_{gs} - V_t)^2 & ; V_{ds} > V_{dsat} \text{ ; saturation.} \end{cases}$$

Below fig shows the I-V characteristics for the transistor. According to the first-order model, the current is zero for gate voltages below *Vt*. For higher gate voltages, current increases linearly with *Vds* for small *Vds* . As *Vds* reaches the saturation point *V*dsat = *VGT*, current rolls off and eventually becomes independent of *Vds* when the transistor is saturated. We will later see that the Shockley model overestimates current at high voltage because it does not account for mobility degradation and velocity saturation caused by the high electric fields.

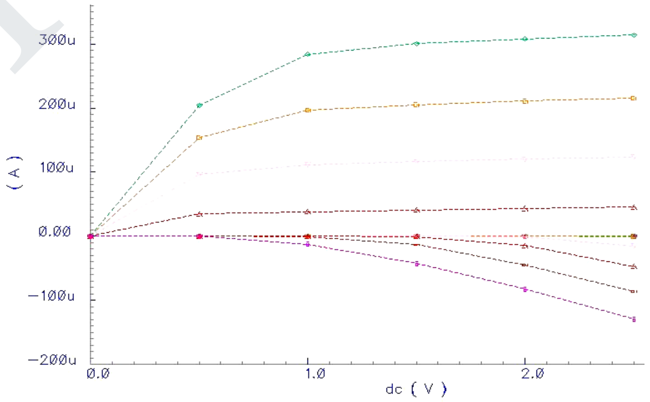


3.a. Explain in detail about DC characteristics of MoS transistor. (MAY/JUN 2016)

- A complementary CMOS inverter consists of a p-type and an n-type device connected in series.
- The DC transfer characteristics of the inverter are a function of the output voltage (V_{out}) with respect to the input voltage (V_{in}).



- The MOS device first order Shockley equations describing the transistors in cut-off, linear and saturation modes can be used to generate the transfer characteristics of a CMOS inverter.
- Plotting these equations for both the n- and p-type devices produces the traces below.



- The DC transfer characteristic curve is determined by plotting the common points of V_{gs} intersection after taking the absolute value of the p-device IV curves, reflecting them about the x-axis and superimposing them on the n-device IV curves.

- We basically solve for $V_{in(n\text{-}type)} = V_{in(p\text{-}type)}$ and $I_{ds(n\text{-}type)}=I_{ds(p\text{-}type)}$
- The desired switching point must be designed to be 50 % of magnitude of the supply voltage i.e. $V_{DD}/2$.
- Analysis of the superimposed n-type and p-type IV curves results in five regions in which the inverter operates.

- **Region A** occurs when 0 leqV_{in} leq $V_{t(n\text{-}type)}$.
  - The n-device is in cut-off (I_{dsn} =0).
  - p-device is in linear region,
  - I_{dsn} = 0 therefore -I_{dsp} = 0
  - $V_{dsp} = V_{out} – V_{DD}$, but V_{dsp} =0 leading to an output of $V_{out} = V_{DD}$.
- **Region B** occurs when the condition V_{tn} leq V_{in} le $V_{DD}/2$ is met.
  - Here p-device is in its non-saturated region Vds neq 0.
  - n-device is in saturation

Saturation current Idsn is obtained by setting $V_{gs} = V_{in}$ resulting in the equation:

$$I_{dsn} = \frac{\beta_n}{2}\left[V_{un} - V_{tn}\right]^2$$

- In **region B** I_{dsp} is governed by voltages V_{gs} and V_{ds} described by:

$$V_{gs} = (V_{in} - V_{DD}) \text{ and } V_{ds} = (V_{out} - V_{DD})$$

$$I_{dsp} = -\beta_p\left[(V_{in} - V_{DD} - V_{tp})\left(V_{out} - V_{DD} - \frac{(V_{out} - V_{DD})^2}{2}\right)\right]$$

$$\text{Recall that} : -I_{dsn} = I_{dsp}$$

$$-\frac{\beta_n}{2}\left[V_{in} - V_{tn}\right]^2 = \beta_p\left[(V_{in} - V_{DD} - V_{tp})(V_{out} - V_{DD}) - \frac{(V_{out} - V_{DD})^2}{2}\right]$$

- **Region C** has that both n- and p-devices are in saturation.
- Saturation currents for the two devices are:

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

AND

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

- **Region D** is defined by the inequality

$$\frac{V_{DD}}{2} < V_{in} \le V_{DD} + V_{tp}$$

- p-device is in saturation while n-device is in its non-saturation region.

$$I_{dsp} = -\frac{\beta_p}{2}\left(V_{in} - V_{DD} - V_{tp}\right)^2;\ V_{in} < V_{tp} + V_{DD}$$

AND

$$I_{dsn} = \beta_n\left[\left(V_{in} - V_{tn}\right)V_{out} - \left(\frac{V_{out}}{2}\right)^2\right];\ V_{in} > V_{tn}$$

- Equating the drain currents allows us to solve for Vout. (See supplemental notes for algebraic manipulations).

- In **Region E** the input condition satisfies:

$$V_{in} \ge V_{DD} - V_{tp}$$

- The p-type device is in cut-off: I_{dsp}=0
- The n-type device is in linear mode
- V_{gsp} = V_{in} –V_{DD} and this is a more positive value compared to V_{tp}.
- V_{out} = 0

3. b. Explain in detail about the propagation delay. Delay Estimations:-

In most designs, there exist many logic paths called critical paths. These paths are recognized by a timing analyzer or circuit simulator. Critical paths are affected by the following four levels.

i. Architectural level
ii. Logic level
iii. Circuit level
iv. Layout level

Propagation delay time (tpd) or max time is the maximum time from the input crossing 50%to the output crossing 50%.The delay can be estimated by the following ways,

i. RC delay models
ii. Linear delay models
iii. Logic efforts
iv. Parasitic delay

1. RC delay models:

The delay of logic gate is computed as the product of RC,where R is the effective driver resistance and C the load capacitance. Logic gates use minimum –length devices for least delay, area and power consumption. The delay of a logic gate depends on the transistor width in the gate and the capacitance of the load.

Effective Resistance and Capacitance:

An NMOS transistor with width of one unit has effective esistance R An PMOS transistor with width of one unit has effective resistance 2R Capacitance consists of gate capacitance cg and source/diffusion capacitance c diff .in most processes cg is equal to c diff , c g and cdiff are proportional to transistor width.

Diffusion capacitance layout effects:

To reduce the diffusion capacitance in the layout, diffusion nodes are shared. Uncontacted nodes have less capacitance. Diffusion capacitance depends on the layout.

2. Elmore delay model:

Elmore delay model estimates the delay of an RC ladder .this is equal to the sum over each node in the ladder of the resistance between the node and supply multiplied by capacitance on the node.

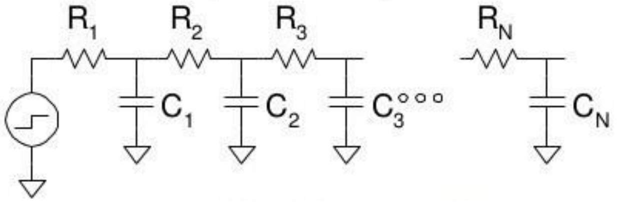


Fig: RC ladder for Elmore Delay Model

3. Linear delay model:

The propagation delay of a gate is d,

$$d= f + p$$

F= effort delay or state effort, which depends on the complexity and fan-out of the gate. P=parasitic delay

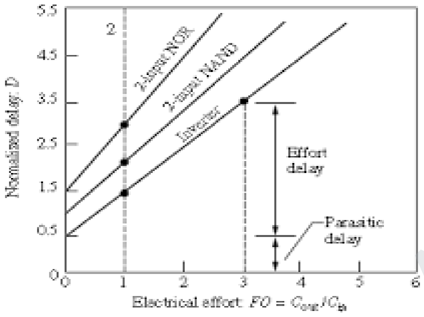


Fig: Normalized delay vs. fan-out

Logical effort:

Logical effort is defined as the ratio of the input capacitance of the gate to the input capacitance of an inverter that delivers the same output current.

Parasitic delay:

Parasitic delay is defined as the delay of the gate when it drives zero load. This can be estimated with RC delay models. The inverter has 3 units of diffusion capacitance on the output.

Gate type	Number of Input				
	1	2	3	4	n
INVERTER	1				
NAND		2	3	4	N
NOR		2	3	4	n
TRISTATE,MULTIPLEXER	2	4	6	8	2n

Logical effort and transistor sizing:

Logical effort provides a simple method to choose the best topology and number of stages of logic for a function. This quickly estimates the minimum possible delay for the given topology and to choose gate sizes that achieve this delay.

Delay in multistage logic networks:

Logical effort is independent of size and electrical effort is dependent on size.

1. path logical effort
2. path electrical effort
3. path effort
4. branching effort
5. path branching effort
6. path delay
7. minimum possible delay

Choosing the best number of stages:

Inverters can be added at the end of a path without changing its function. Extra inverters and parasitic delay, but do not change the path logical effort.

4.a. Explain about device modeling in detail.(MAY/JUN 2012)(MAY/JUN 2013).

SPICE provides a wide variety of MOS transistor models with various trade-offs between complexity and accuracy. Level 1 and Level 3 models were historically important, but they are no longer adequate to accurately model very small modern transistors. BSIM models are more accurate and are presently the most widely used. Some companies use their own proprietary models. This section briefly describes the main features of each of these models. It also describes how to model diffusion capacitance and how to run simulations in various process corners. The model descriptions are intended only as an overview of the capabilities and limitations of the models; refer to a SPICE manual for a much more detailed description if one is necessary.

Level 1 Models

The SPICE Level 1, or Shichman-Hodges Model [Shichman68] is closely related to the Shockley model described in EQ (2.10), enhanced with channel length modulation and the body effect. The basic current model is:

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

The parameters from the SPICE model are given in ALL CAPS. Notice that □ is written instead as KP(*W*eff /*L*eff), where KP is a model parameter playing the role of *k. W*eff and *L*eff are the effective width and length). The LAMBDA term (LAMBDA = 1/*VA*) models channel length modulation The threshold voltage is modulated by the source-to-body voltage *Vsb* through the body effect.

The gate capacitance is calculated from the oxide thickness TOX. The default gate capacitance model in HSPICE is adequate for finding the transient response of digital circuits. More elaborate models exist that capture nonreciprocal effects that are important for analog design. Level 1 models are useful for teaching because they are easy to correlate with hand analysis, but are too simplistic for modern design.

Level 2 and 3 Models

The SPICE Level 2 and 3 models add effects of velocity saturation, mobility degradation, subthreshold conduction, and drain-induced barrier lowering. The Level 2 model is based on the Grove-Frohman equations, while the Level 3 model is based on empirical equations that provide similar accuracy, faster simulation times, and better convergence. However, these models still do not provide good fits to the measured I-V characteristics of modern transistors.

BSIM Models

The Berkeley Short-Channel IGFET1 Model (BSIM) is a very elaborate model that is now widely used in circuit simulation. The models are derived from the underlying device physics but use an enormous number of parameters to fit the behavior of modern transistors. BSIM versions 1, 2, 3v3, and 4 are implemented as SPICE levels 13, 39, 49, and 54, respectively.

Features of the model include:

- Continuous and differentiable I-V characteristics across subthreshold, linear, and saturation regions for good convergence
- Sensitivity of parameters such as *Vt* to transistor length and width

- Detailed threshold voltage model including body effect and drain-induced barrier lowering
- Velocity saturation, mobility degradation, and other short-channel effects
- Multiple gate capacitance models
- Diffusion capacitance and resistance models
- Gate leakage models

As the BSIM models are so complicated, it is impractical to derive closed-form equations for propagation delay, switching threshold, noise margins, etc., from the underlying equations. However, it is not difficult to find these properties through circuit simulation. Device characterisation will show simple simulations to plot the device characteristics over the regions of operation that are interesting to most digital designers and to extract effective capacitance and resistance averaged across the switching transition. The simple RC model continues to give the designer important insight about the characteristics of logic gates.

Diffusion Capacitance Models

The p–n junction between the source or drain diffusion and the body forms a diode. We depends on the area and perimeter of the diffusion. HSPICE provides a number of methods to specify this geometry, controlled by the ACM (Area Calculation Method) parameter, which is part of the transistor model. have seen that the diffusion capacitance determines the parasitic delay of a gate and The diffusion capacitance model is common across most device models including Levels 1–3 and BSIM. By default, HSPICE models use ACM = 0. In this method, the designer must specify the area and perimeter of the source and drain of each transistor.

The SPICE models also should contain parameters CJ, CJSW, PB, PHP, MJ, and MJSW. Assuming the diffusion is reverse-biased and the area and perimeter are specified, the diffusion capacitance between source and body is computed as described in

$$C_{sb} = \mathrm{AS} \times \mathrm{CJ} \times \left(1 + \frac{V_{sb}}{\mathrm{PB}}\right)^{-MJ} + \mathrm{PS} \times \mathrm{CJSW} \times \left(1 + \frac{V_{sb}}{\mathrm{PHP}}\right)^{-MJSW}$$

The drain equations are analogous, with S replaced by D in the model parameters. The BSIM3 models offer a similar area calculation model (ACM = 10) that takes into account the different sidewall capacitance on the edge adjacent to the gate. Note that the PHP parameter is renamed to PBSW to be more consistent.

$$C_{sb} = \mathrm{AS} \times \mathrm{CJ} \times \left(1+\frac{V_{sb}}{\mathrm{PB}}\right)^{-MJ} + (\mathrm{PS}-W) \times \mathrm{CJSW} \times \left(1+\frac{V_{sb}}{\mathrm{PBSW}}\right)^{-MJSW} +$$
$$W \times \mathrm{CJSWG} \times \left(1+\frac{V_{sb}}{\mathrm{PBSWG}}\right)^{-MJSWG}$$

If the area and perimeter are not specified, they default to 0 in ACM = 0 or 10, grossly underestimating the parasitic delay of the gate. HSPICE also supports ACM = 1, 2, 3, and 12 that provide nonzero default values when the area and perimeter are not specified. Check your models and read the HSPICE documentation carefully. The diffusion area and perimeter are also used to compute the junction leakage current. However, this current is generally negligible compared to subthreshold leakage in modern devices.

Design Corners

Engineers often simulate circuits in multiple design corners to verify operation across variations in device characteristics and environment. HSPICE includes the .lib statement that makes changing libraries easy. The deck first sets SUP to the nominal supply voltage of 1.0 V. It then invokes .lib to read in the library specifying the TT conditions. In the stimulus, the .alter statement is used to repeat the simulation with changes. In this case, the design corner is changed. Altogether, three simulations are performed and three sets of waveforms are generated for the three design corners.

4.b. Explain in detail about scaling concept. (MAY/JUN 2016)

(or)

Discuss on transistor and interconnect scaling

Scaling:

- As the transistors become smaller, they switch faster, dissipate less power and are cheaper to manufacture. Despite the ever increase is challenges process advances have actually accelerated in the past decade.
- Such scaling is unprecedented in the history of technology. However scaling also exacerbar noise and reliability issues and introduces new problems.
- Designers need to be able to predict the effect of this feature size scaling on chip performance to plan future products, ensures existing products will scale gracefully to future processes for cost reduction and anticipate looming design challenges.

Transistor Scaling:

The characteristics of an MOS device can be maintained and the basic operational characteris. Can be preserved if the critical parameters of a device are scaled by a dimensionless factor . These parameters include.

º All dimensions (x,y, z directions)

º Device voltages

º Doping concentration densities.

Another approach is **lateral Scaling** , in which only the gate length is scaled. This is commonly called a gate shrink because it can done easily to an existing mask database for a design.

For **constant field scaling**, all devices dimension including channel length L, width W and oxide thickness t_{or} are reduced by a factor of 1/s. The supply voltage VDD and the threshold voltages are also reduced by1/s.

- The substrate doping WA is increased by.
- Because both distance and voltage are scaled equally, the electric field remain constant.
- A gate shrink scales only the channel leng leaving other dimensions, voltages and doping levels unchanged.
- This offers a quadratic improvement in gate delay according to the first order model.
- The gate delay improvement is closer to linear because velocity saturation keeps the current and effective resistance appronimately constant.
- The constant voltage scaling increased the electric fields in the devices. By the 1 μ m generation velocity saturation was servere enough that decreasing feature size no longer improved device current.

Inter connect Scaling:

- Two common approaches to interconnect scaling are to either scale all dimensions or keep the wire height constant.
- Wire length decreases for some types of wires, but may increase for others? Local are scaled wires are those that decrease in length during scaling.

- Example: A wire across 64 bits ALU is local because it becomes shorter as the ALU is migrated to finer process. A wire across a particular micro processer is scaled because when the microprocessor is shrunk to the new process the wire will also shrink.
- Un repeated interconnect delay is remaining about constant for local interconnect and increasing for global interconnect . This presents a problem because transistor are getting faster, So the ratio if interconnect to gate delay interconnect with scaling .
- In moders process with aspect ratios 1-5-22 fringing capacitance accounts for the majority of the total capacitance.
- Scaling spacing but not height interconnect the fringing capacitance enough that the extra thickness scarcely improves delay.
- Observe that when wire thickness is called the capacitance per unit length remains constant. Hence, a reasonable initial estimate of the capacitance of a minimum-pitch were is about 0.2fF/ μm, independent of the process.
- Wire capacitance is roughly 1/10-1/6 of gate capacitance per unit length.

Impacts on Design:

- One of the limitations of first order scaling is that it gives the wrong impression of being able to scale proportionally to zero dimensions and zero voltage.

Improved performance and cost:

- The most positive impact of scaling is that performance and cost are steadily improving. System architects need to understand the scaling of CMOS technologies and predict the capabilities of the process several years into the future, when a chip will be completed.

Interconnect :

Scaling transistors are steadily improving in delay but scaled wires are holding constant or getting worse.

- The wire problem motivated a number of papers predicting the demise of conventional wires.
- The plot is misleading in two ways.

- First the gate delay is shown for a single unloaded transistor rather than a realistically loaded gate. |Second, the wire delay shown for fixed lengi but as μ technology scales, most local wires connecting gates within a unit also become shorter.

Power:

In classical constant field scaling, power density remains constant and overall chip power increases only slowly with die size.

- Power density has sky rocketed because clock frequencies have increased much faster to classical scaling would predict and V_{DD} is some what higher than constant field scaling would demand.
- Dynamic power consumption will not continue to increase at such rates because it will become uneconomical to cool the chips.
- The static power consumption caused by sub threshold leakage was historically negligible but becomes important for threshold voltage below about 0.3 to 04v.

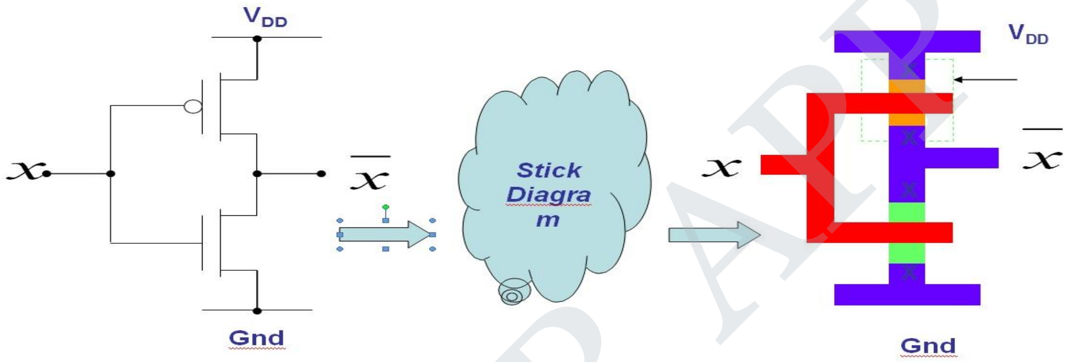
5. Explain the stick diagram and layout diagram with examples. May 11, May13, Nov/Dec10

Stick diagrams:

Stick diagrams are used to convey layer information through the use of a colour code for example in NMOS design.

- Green for n- diffusion
- Red for poly silicon
- Blue for metal
- Yellow for implant
- Black for contact areas
- The designer can draw a layout using coloured lines to represent the various process layers such as diffusion, metal and poly silicon.
- Where poly silicon crosses the diffusion, transistors are created and where metal wires join diffusion or poly silicon, contacts are formed.
- A stick diagram is a cartoon of a chip layout. They are not the exact models of layout.

- The stick diagram represents the rectangles with lines which represents wires are component symbols.
- The colour cooling has been complemented by monochrome encoding of the lies so the black and white copies of stick diagrams do not lose the layer information.
- The colour and monochrome encoding scheme used has been evolved to cover NMOS and CMOS processes.
- To illustrate the stick diagram inverter circuits are presented below in NMOS, and in P well CMOS technology.



- Having conveyed layer information and topology by using stick or symbolic diagrams. These diagrams relatively easily turned into mask layouts.
- The below diagram stressing the ready translation into mask layout form. In order that the mask layout produced during design will be compactible with the fabrication process.

  Aser of design rules are set out for layouts.

Stick diagram using NMOS Deisgn:

We consider single metal, single poly silica NMOS technology. The layout of NMOS involves.

- N-diffusion and other thin oxide regions- green
- Polysilicon - red
- Metal -blue
- Impant -yellow
- Contacts - black or brown

A transistor is formed wherever poly silicon crosses n-diffusion and all diffusion wires are n-type. The various steps involved in the design style are.

Step1: Draw the metal VDD and GND rails in parallel allowing enough space between them for the other circuit element which will be required.

Step 2: Draw the thinox paths between the rails for inverters and inverter based logic.

Step 3: Draw the pull up structure which comprises a depletion mode transistor interconnected between the output point and V_{DD}.

Step 4:

Draw the pull down structure comprising an enhancement mode structure interconnected between the output point and GNO.

Step 5: Signal paths may be switched by pass transistor, and along signal paths often require metal buses.

Design Rules and layout:

The design rules primarily address two issue

1) The geometrical reproduction of features that can be reproduced by the mask-making and lithographical process.
2) The interactions between different layers. There are several approaches that can be taken in describing the design rules. These include

- Micron design rules:
  - Stated at some micron resolution
  - Usually given as a list of minimum feature sizes and spacings for all masks required in a given process.
  - Normal style for industry.
- Lambda (λ) based design rules
  - These rules popularized by Mead and Conway are based on a single parameter, λ which charactized the linear feature- the resolution of the complete wafer implementation process – and permits first order scaling.
  - They have been widely used, particularly in the educational context and in the design of multi project chips.

Layout (λ) based design Rules:

The lambda, λ design rules are bases on mead and Conway work and in gereral, design rules and layout methodology are based on the concept of λ which provides a process and feature size. Independent way of making mask dimensions to scale.

- All paths in all layers will be dimensioned in λ units and sub-sequently λ can be allocated an appropriate value compactible with the feature size of the fabricalion process.
- Design rules can be conveniently set out in diagrammatic form as shown below.

Contact cuts:

The contacts between layers are set out as shown below. Here it will be obserred that connection can be made between two or, in the case of NMOs design, three layers.

1) Metal to poly silicon or to diffusion

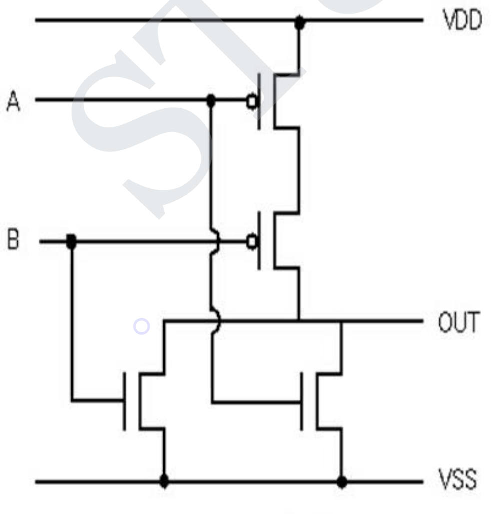
There are three possible approaches for making contacts between poly silicon and diffusion in NMOS circuits. There are

i) Poly silicon to metal then metal to diffusion

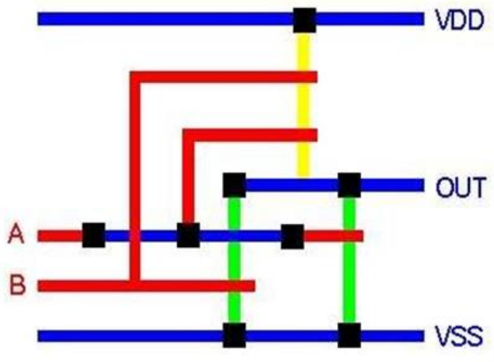
ii) Buried contact poly silicon to diffusion

iii) Butting contact.

- The 2 λ x2 λ contac cut indicates and area in which the oxide is to be removed down to the underlying polysilicon or diffusion surface.
- When the deposition of the metal layer takes place, the metal is deposited through the contact cut areas on to the underlying areas so that contact is made between the layers.



NOR gate in CMOS



UNIT II

COMBINATIONAL LOGIC CIRCUITS

PART A

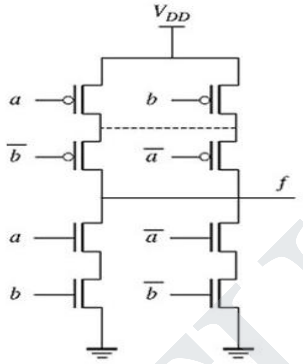
1. **What is bubble pushing?**

CMoS gates are inherently inverting, so AND and OR functions must be built from NAND and NOR gates. Demorgans law helps with this conversion.

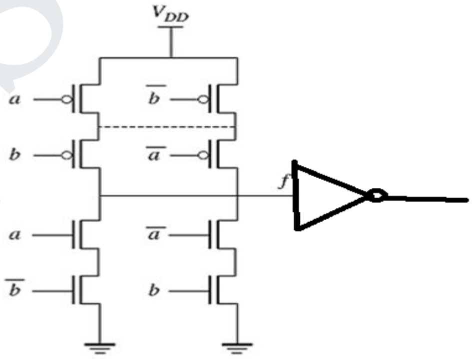
A NAND gate is equivalent to an OR of inverter inputs. A NOR gate is equivalent to an AND gate of inverter inputs. The same relationship applies to gates with more inputs switching between these representation is easy to do and is often called bubble pushing.

2. **Draw XOR gate and XNOR gate using transmission gates.**

XOR gate

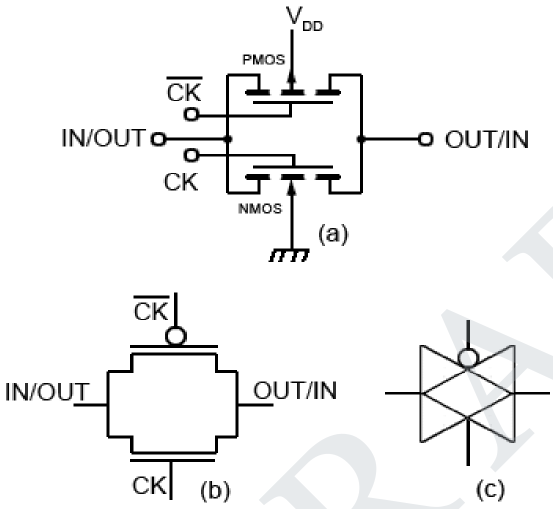


XNOR



3. **Write a note on CMoS transmission gate logic.(APRMAY 2011)**

The transmission gate acts as voltage controlled resistor connecting the input and the output. It can be used as logic structure, switch, latch element etc,.



4. **What are the factors that cause static power dissipation in CMoS circuits? (Nov/Dec 2012)**

Static power dissipation due to:

- Sub threshold conduction through OFF transistor.
- Tunneling current through gate oxide.
- Leakage through reverse biased diodes.

5. **List the various power losses in CMoS circuits. (Nov/Dec 2013)**

Static power dissipation

Dynamic power dissipation

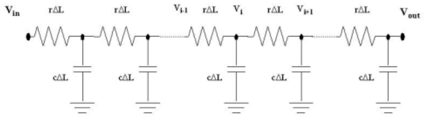
- Charging and discharging of load capacitance
- Short circuit curerent while both PMoS and nMoS networks are partially ON

6. **State types of power dissipation .(APR/MAY 2015)**

- Static power dissipation
- Dynamic power dissipation

7. **Give the expression for Elmore delay and state the various parameters associated with it.(NOV/DEC 2014) (MAY/JUN 2016)**

Viewing on transistors as resistors a chain of transistors as an RC ladder. The Elmore model estimates the delay of an RC ladder as the sum over each node in the ladder of the resistance between that node and a supply multiplied by the capacitance on the node.



$$\tau_N = \sum_{i-1}^{N} R_i \sum_{j-i}^{N} C_j = \sum_{i-1}^{N} C_i \sum_{j-1}^{i} R_j$$

8. **Define power dissipation.(NOV/DEC 2013)**

The instantaneous power p(t) drawn from the power supply is proportional to the supply current iDD(t) and the supply voltage Vdd.

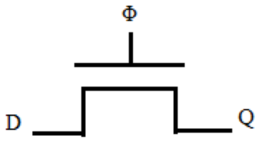
The energy consumed over sometime interval T is the integral of the instantaneous power.

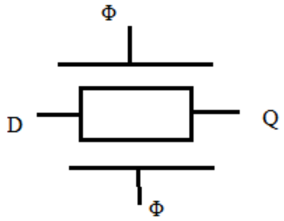
9. **Implement a 2:1 multiplier using pass transistor(NOVDEC 2013)(APR/MAY 2015).**

When an nMoS or pMoS is used alone as an imperfect switch, it is called as a pass transistor. By combining a nMoS and a pMoS transistor in parallel a switch is obtained that turns on when a 1 is applied to g in which 0's are passed in an

acceptable fashion.this is a transmission gate or pass gate.

10. Design a 1-bit dynamic register using pass transistor.(NOV/DEC 2013)





The fig 1 shows a very simple transparent latch built from a single transistor it is compact and fast but suffers four limitations.

Fig 2 uses a CMoS transmission gate in place of the sinlge nMoS pass transistor to offer rail-rail output swings.

11. Why single phase dynamic logic structure cannot be caed. justify(MAY/JUN 2016)

In dynamic logic, a problem arises when caing one gate to the next. The precharge "1" state of the first gate may cause the second gate to discharge prematurely, before the first gate has reached its correct state. This uses up the "precharge" of the second gate, which cannot be restored until the next clock cycle, so there is no recovery from this error

PART-B

1. Write short notes on Static CMOS Design. (MAY'11, MAY'13)

The most widely used logic style is static complementary CMOS. The static CMOS style is really an extension of the static CMOS inverter to multiple inputs. The primary advantage of the CMOS structure is robustness (i.e., low sensitivity to noise), good performance, and low power consumption with no static power dissipation. Most of those properties are carried over to large fan-in logic gates implemented using a similar circuit topology.

The complementary CMOS circuit style falls under a broad class of logic circuits called static circuits in which at every point in time (except during the switching transients), each gate output is connected to either VDD or Vss via a low-resistance path. Also, the outputs of the gates assume at all times the value of the Boolean function implemented by the circuit (ignoring, once again, the transient effects during switching periods). This is in contrast to the dynamic circuit class, which relies on temporary storage of signal values on the capacitance of high-impedance circuit nodes. The latter approach has the advantage that the resulting gate is simpler and faster. Its design and operation are however more involved and prone to failure due to an increased sensitivity to noise. The design of various static circuit flavors includes complementary CMOS, ratioed logic (pseudo-NMOS and DCVSL), and pass transistor logic.

a. Complementary CMOS

A static CMOS gate is a combination of two networks, called the pull-up network (PUN)and the pull-down network (PDN) (Figure 1). The figure shows a generic N input logic gate where all inputs are distributed to both the pull-up and pull-down networks. The function of the PUN is to provide a connection between the output and VDD anytime the output of the logic gate is meant to be 1 (based on the inputs). Similarly, the function of the PDN is to connect the output to VSS when the output of the logic gate is meant to be 0. The PUN and PDN networks are constructed in a mutually exclusive fashion such that one and only one of the networks is conducting in steady state. In this way, once the transients have settled, a path always exists between VDD and the output F, realizing a high output ("one"), or, alternatively, between VSS and F for a low output ("zero"). This is equivalent to stating that the output node is always a low- impedance node in steady state.

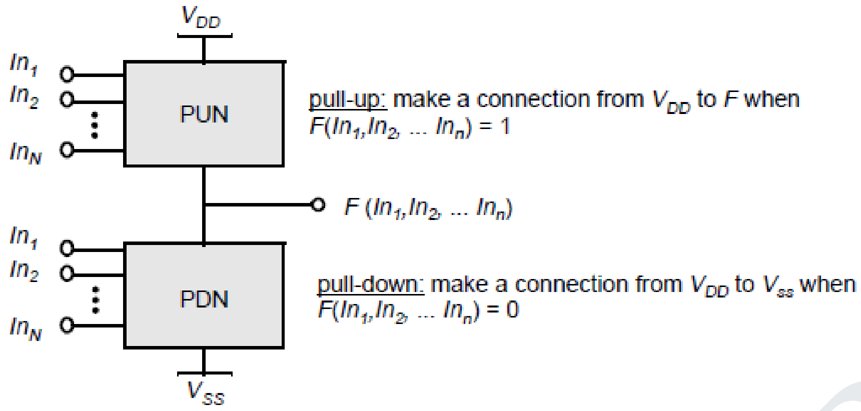
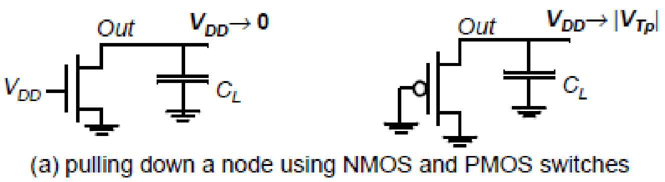

V_{DD}
In_1
In_2
In_N
PUN
pull-up: make a connection from V_{DD} to F when $F(In_1,In_2, \dots In_n) = 1$
$F(In_1,In_2, \dots In_n)$
In_1
In_2
In_N
PDN
pull-down: make a connection from V_{DD} to V_{ss} when $F(In_1,In_2, \dots In_n) = 0$
V_{SS}

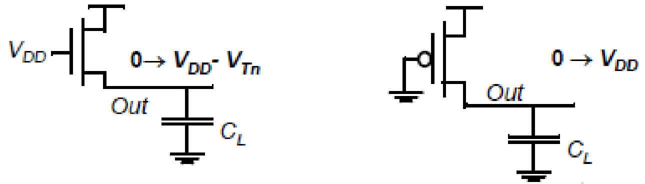
Figure 1: Complementary logic gate as a combination of a PUN (pull-up network) and a PDN (pull-down network).

In constructing the PDN and PUN networks, the following observations should bekept in mind:

- A transistor can be thought of as a switch controlled by its gate signal. An NMOS switch is on when the controlling signal is high and is off when the controlling signal is low. A PMOS transistor acts as an inverse switch that is on when the controlling signal is low and off when the controlling signal is high.
- The PDN is constructed using NMOS devices, while PMOS transistors are used in the PUN. The primary reason for this choice is that NMOS transistors produce “strong zeros,” and PMOS devices generate “strong ones”. To illustrate this, consider the examples shown in Figure 2. In Figure 2.a, the output capacitance is initially charged to VDD. Two possible discharge scenarios are shown. An NMOS device pulls the output all the way down to GND, while a PMOS lowers the output no further than |VTp| — the PMOS turns off at that point, and stops contributing discharge current. NMOS transistors are hence the preferred devices in the PDN. Similarly, two alternative approaches to charging up a capacitor are shown in Figure 2.b, with the output initially at GND. A PMOS switch succeeds in charging the output all the way to VDD, while the NMOS device fails to raise the output above VDD-VTn. This explains why PMOS transistors are preferentially used in a PUN.

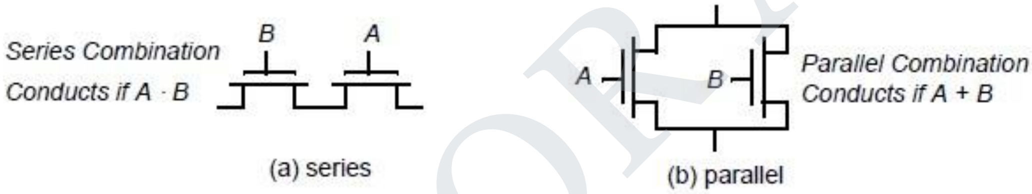


(a) pulling down a node using NMOS and PMOS switches



(b) pulling down a node using NMOS and PMOS switches

Figure 2 Simple examples illustrate why an NMOS should be used as a pull-down, and a PMOS should be used as a pull-up device.



(a) series

(b) parallel

Figure 3 NMOS logic rules — series devices implement an AND, and parallel devices implement an OR.

A set of construction rules can be derived to construct logic functions (Figure 4). NMOS devices connected in series corresponds to an AND function. With all the inputs high, the series combination conducts and the value at one end of the chain is transferred to the other end. Similarly, NMOS transistors connected in parallel represent an OR function. A conducting path exists between the output and input terminal if at least one of the inputs is high. Using similar arguments, construction rules for PMOS networks can be formulated. A series connection of PMOS conducts if both inputs are low, representing a NOR function (A.B = A+B), while PMOS transistors in parallel implement a NAND (A+B = A· B.

- Using De Morgan's theorems ((A + B) = A· B and A· B = A + B), it can be shown that the pull-up and pull-down networks of a complementary CMOS structure are dual networks. This means that a parallel connection of transistors in the pull-up network corresponds to a series connection of the corresponding devices in the pull-down network, and vice versa. Therefore, to construct a CMOS gate, one of the networks (e.g., PDN) is implemented using combinations of series and parallel devices. The other network (i.e.,PUN) is obtained using duality principle by walking the hierarchy, replacing series sub- nets with parallel sub-nets, and parallel sub-nets with series sub-nets. The complete CMOS gate is constructed by combining the PDN with the PUN.

- The complementary gate is naturally inverting, implementing only functions such as NAND, NOR, and XNOR. The realization of a non-inverting Boolean function (such as AND OR, or XOR) in a single stage is not possible, and requires the addition of an extra inverter stage.
- The number of transistors required to implement an N-input logic gate is 2N.

b. II Ratioed Logic

Ratioed logic is an attempt to reduce the number of transistors required to implement a given logic function, at the cost of reduced robustness and extra power dissipation. The purpose of the PUN in complementary CMOS is to provide a conditional path between VDD and the output when the PDN is turned off. In ratioed logic, the entire PUN is replaced with a single unconditional load device that pulls up the output for a high output (Figure 5.a). Instead of a combination of active pull-down and pull-up networks, such a gate consists of an NMOS pull- down network that realizes the logic function, and a simple load device. Figure 5.b shows an example of ratioed logic, which uses a grounded PMOS load and is referred to as a pseudo- NMOS gate.

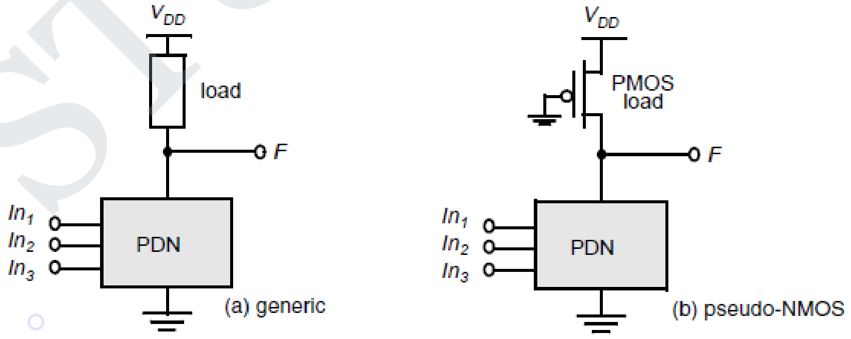


Figure 5: Ratioed logic gate.

The clear advantage of pseudo-NMOS is the reduced number of transistors (N+1versus 2N for complementary CMOS). The nominal high output voltage (VOH) for this gate is VDD since the pull-down devices are turned off when the output is pulled high (assuming that VOL is below VTn). On the other hand, the **nominal low output voltage is not 0 V** since there is a fight between the devices in the PDN and the grounded PMOS load device. This results in reduced noise margins and more importantly static power dissipation.

The sizing of the load device relative to the pull-down devices can be used to trade-off parameters such a noise margin, propagation delay and power dissipation. Since the voltage swing on the output and the overall functionality of the gate depends upon the ratio between the NMOS and PMOS sizes, the circuit is called ratioed. This is in contrast to the ratioless logic styles, such as complementary CMOS, where the low and high levels do not depend upon transistor sizes.

Computing the dc-transfer characteristic of the pseudo-NMOS proceeds along paths similar to those used for its complementary CMOS counterpart. The value of VOL is obtained by equating the currents through the driver and load devices for Vin = VDD. At this operation point, it is reasonable to assume that the NMOS device resides in linear mode (since the output should ideally be close to 0V), while the PMOS load is saturated.

$$k_n\left((V_{DD}-V_{Tn})V_{OL}-\frac{V_{OL}^2}{2}\right) = k_p\left((-V_{DD}-V_{Tp})\cdot V_{DSAT}-\frac{V_{DSAT}^2}{2}\right)$$

Assuming that VOL is small relative to the gate drive (VDD-VT) and that VTn is equal to VTp in magnitude, VOL can be approximated as:

$$V_{OL}\approx\frac{k_p(-V_{DD}-V_{Tp})\cdot V_{DSAT}}{k_n(V_{DD}-V_{Tn})}\approx\frac{\mu_p\cdot W_p}{\mu_n\cdot W_n}\cdot\left|V_{DSAT}\right|$$

In order to make VOL as small as possible, the PMOS device should be sized much smaller than the NMOS pull-down devices. Unfortunately, this has a negative impact on the propagation delay for charging up the output node since the current provided by the PMOS device is limited.

A major disadvantage of the pseudo-NMOS gate is the static power that is dissipated when the output is low through the direct current path that exists between VDD and GND. The static power consumption in the low-output mode is easily derived.

$$P_{low} = V_{DD}I_{low}\approx V_{DD}\cdot k_p\left((-V_{DD}-V_{Tp})\cdot V_{DSATp}-\frac{V_{DSATp}^2}{2}\right)$$

2. Discuss in detail about the Dynamic CMOS design. (MAY'11)

Dynamic circuits overcome these drawbacks by using a clocked pull-up transistor rather than a pMOS that is always ON. Dynamic circuit operation is divided into two modes, as shown in Figure 9.22. During Precharge, the clock K is 0, so the clocked pMOS is ON and initializes the output Y high. During evaluation, the clock is 1 and the clocked pMOS turns OFF. The output may remain high or may be discharged low through the pull down network. Dynamic circuits require careful clocking, consume significant dynamic power, and are sensitive to noise during evaluation.

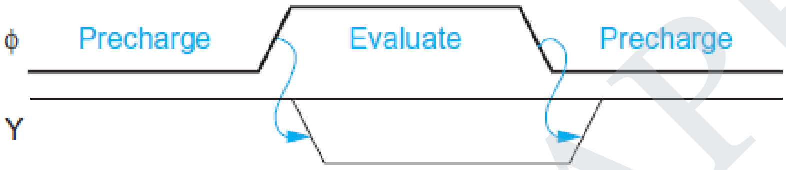


FIGURE 9.22 Precharge and evaluation of dynamic gates

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 Figure 9.23. The extra transistor is sometimes called a foot. Figure 9.2 shows generic footed and unfooted gates.

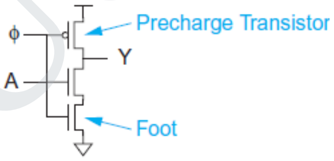


FIGURE 9.23 Footed dynamic inverter

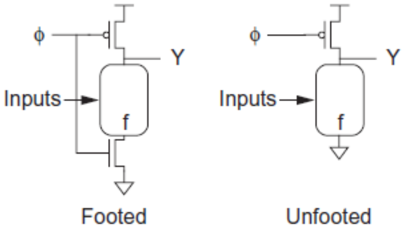


FIGURE 9.24 Generalized footed and unfooted dynamic gates

Figure 9.25 estimates the falling logical effort of both footed and unfooted dynamic gates. Footed gates have higher logical effort than their unfooted counterparts but are still an improvement over static logic.

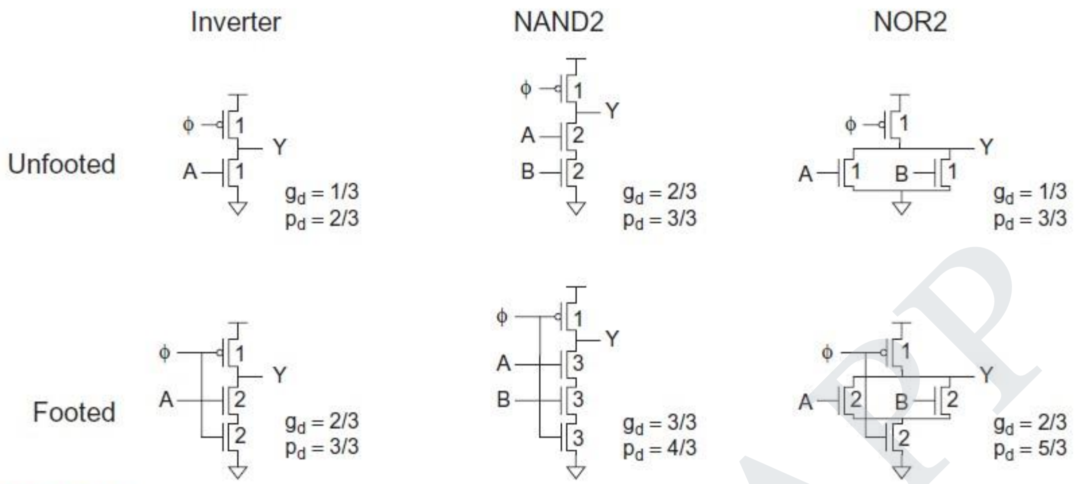


FIGURE 9.25 Catalog of dynamic gates

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

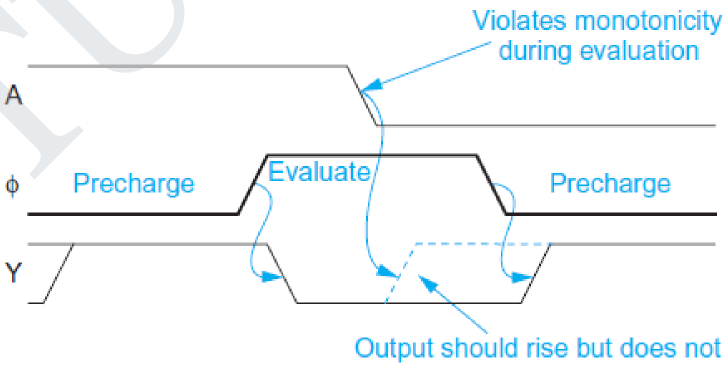

Violates monotonicity during evaluation
A
ϕ
Precharge
Evaluate
Precharge
Y
Output should rise but does not

FIGURE 9.26 Monotonicity problem

shows waveforms for a footed dynamic inverter in which the input violates monotonicity.

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 Figure 9.27. Dynamic gates sharing the same clock cannot be directly connected.

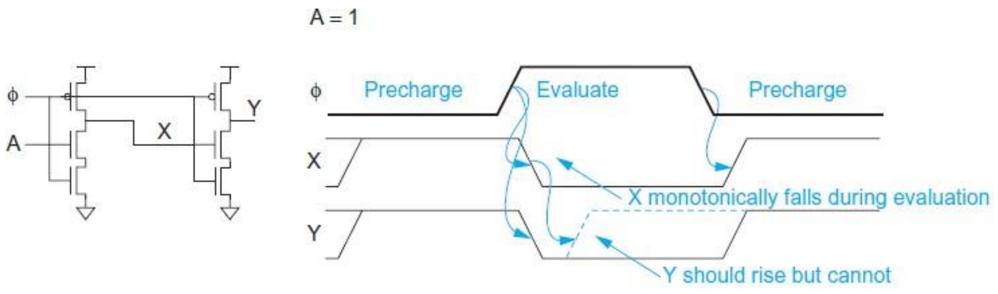


FIGURE 9.27 Incorrect connection of dynamic gates

Advantages

- Lower input capacitance
- No contention during switching
- Zero static power dissipation

Disadvantages

- Require careful clocking
- Consume significant dynamic power
- Sensitive to noise

Applications

- Used in wide NOR functions
- Used in multiplexers

The various drawbacks can be overcome by the following logics:

- Domino logic
- Dual-rail Domino logic
- Keepers
- Multiple output Domino logic
- NP and Zipper Domino

a. Domino Logic

The monotonicity problem can be solved by placing a static CMOS inverter between dynamic gates, as shown in Figure 9.28(a). This converts the monotonically falling output into a monotonically rising signal suitable for the next gate, as shown in Figure 9.28(b). The dynamic-static pair together is called a domino gate because Precharge resembles setting up a chain of dominos and evaluation causes the gates to fire like dominos tipping over, each triggering the next. 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. The symbols for the dynamic NAND, HI-skew inverter, and domino AND are shown in Figure 9.28(c).

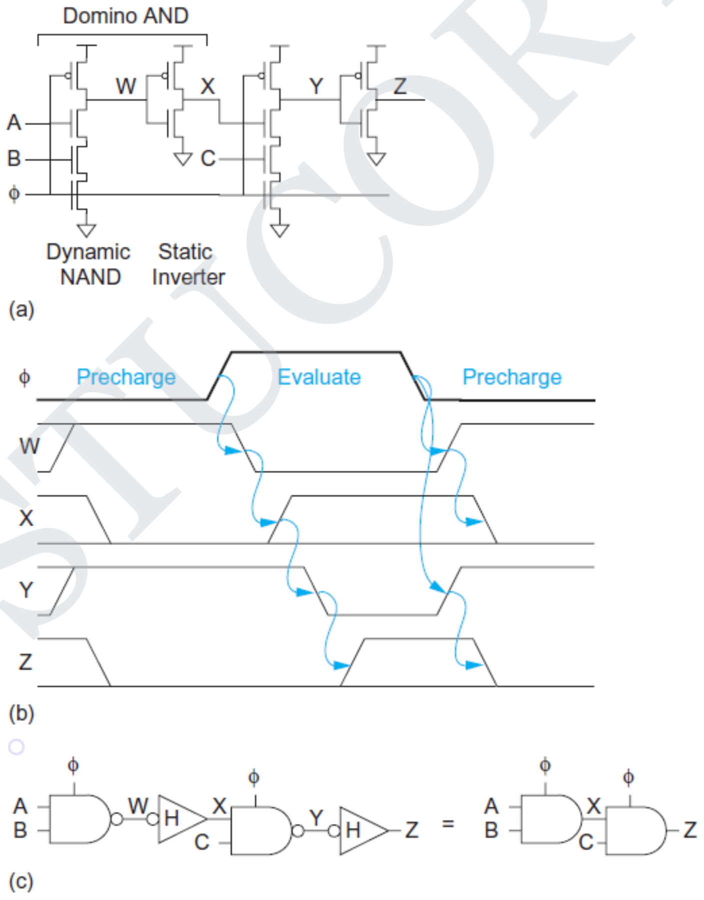


FIGURE 9.28 Domino gates

b. 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 9.2 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 Precharge, neither _h nor _l is asserted. The pair of lines should never be both asserted simultaneously during correct operation. Dual-rail domino gates accept both true and complementary inputs and compute both true and complementary outputs, as shown in Figure 9.30(a). Observe that this is identical to static CVSL circuits from Figure 9.20 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 9.30(b) shows a dual-rail AND/NAND gate and Figure 9.30(c) shows a dual-rail XOR/XNOR gate.

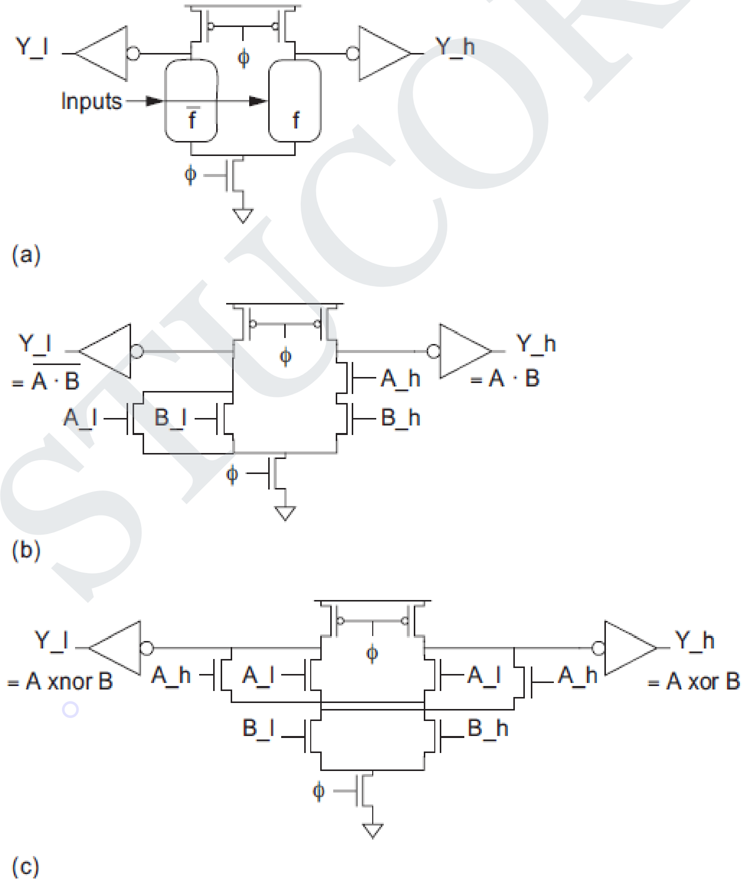


FIGURE 9.30 Dual-rail domino gates

Dual-rail structures also neither lose the efficiency of wide dynamic NOR gates because they require complementary tall dynamic NAND stacks. Dual-rail domino signals not only the result of a computation but also indicates when the computation is done. Before computation completes, both rails are Precharge. When the computation completes, one rail will be asserted. A NAND gate can be used for completion detection, as shown in Figure 9.31. Coupling can be reduced in dual-rail signal busses by inter digitating the bits of the bus, as shown in Figure 9.32. Each wire will never see more than one aggressor switching at a time because only one of the two rails switches in each cycle.

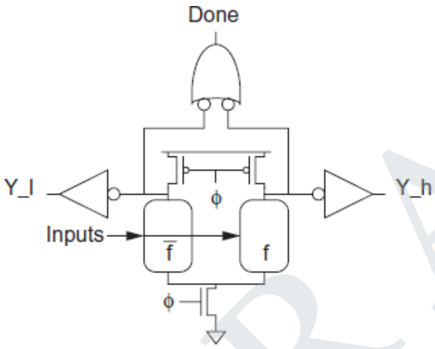


FIGURE 9.31 Dual-rail domino gate with completion detection

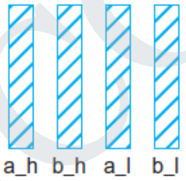


FIGURE 9.32 Reducing coupling noise on dual-rail busses

c. Keepers

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 sub threshold, gate, and junction leakage. The time constants tend to be in the millisecond to nanosecond range, depending on process and temperature. This problem is analogous to leakage in dynamic RAMs. Moreover, dynamic circuits have poor input noise margins. If the input rises above Vt 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 9.33 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 pre- vent X from floating. When X falls, the keeper initially opposes the transition so it must be much weaker than the pull down network. Eventually Y rises, turning the keeper OFF and avoiding static power dissipation.

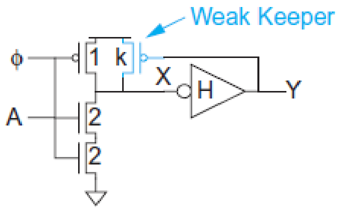


FIGURE 9.33 Conventional keeper

d. Multiple-Output Domino Logic (MODL)

It is often necessary to compute multiple functions where one is a subfunction of another or shares a subfunction. Multiple-output domino logic (MODL) [Hwang89, Wang97] saves area by combining all of the computations into a multiple-output gate. A popular application is in addition, where the carry-out c_i of each bit of a 4-bit block must be computed, as discussed in Section 11.2.2.2. Each bit position i in the block can either propagate the carry (p_i) or generate a carry (g_i). The carry-out logic is

$$c_1 = g_1 + p_1c_0$$
$$c_2 = g_2 + p_2(g_1 + p_1c_0)$$
$$c_3 = g_3 + p_3(g_2 + p_2(g_1 + p_1c_0))$$
$$c_4 = g_4 + p_4(g_3 + p_3(g_2 + p_2(g_1 + p_1c_0)))$$

This can be implemented in four compound AOI gates, as shown in Figure

9.44(a). Notice that each output is a function of the less significant outputs. The more compact MODL design shown in Figure 9.44(b) is often called a Manchester carry chain.

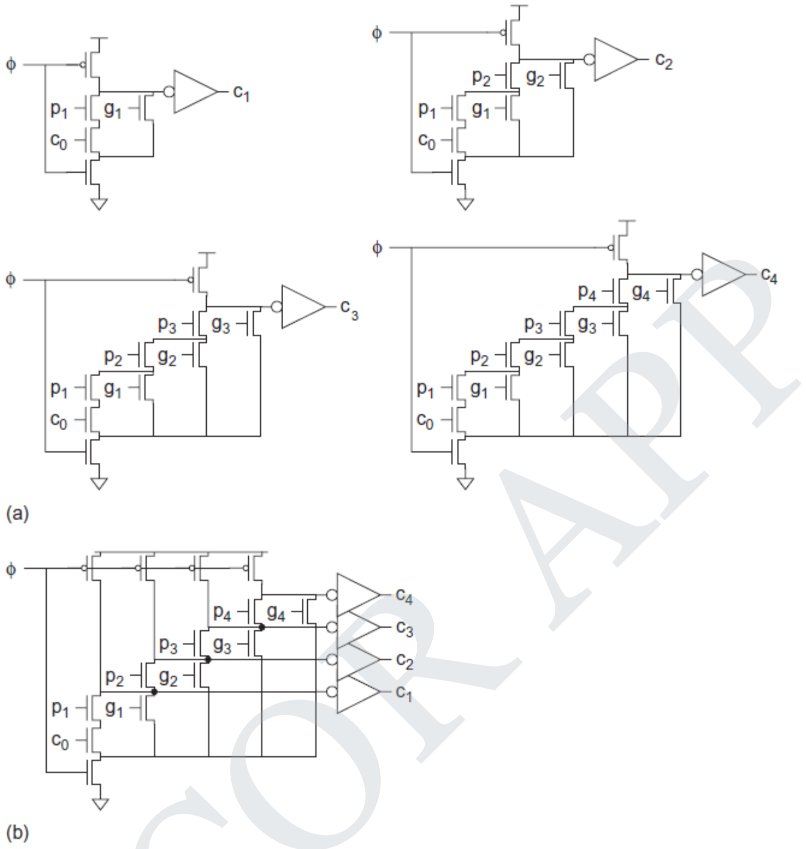


FIGURE 9.44 Conventional and MODL carry chains

e. NP and Zipper Domino

Another variation on domino is shown in Figure 9.46(a). The HI-skew inverting static gates are replaced with predischarged dynamic gates using pMOS logic. For example, a footed dynamic p-logic NAND gate is shown in Figure 9.46(b). When K is 0, the first and third stages precharge high while the second stage pre- discharges low. When K rises, all the stages evaluate. Domino connections are possible, as shown in Figure 9.46(c). The design style is called NP Domino or NORA Domino.

Disadvantages

- Logical effort is the worst
- Susceptible to noise

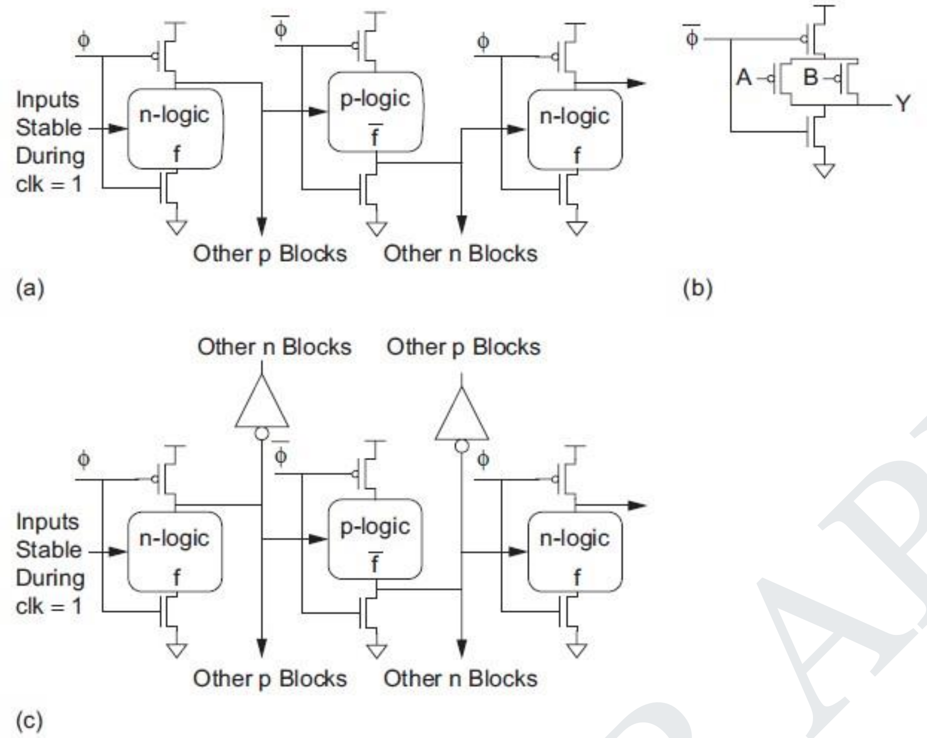


FIGURE 9.46 NP Domino

-----------------------------*************---------------------------------

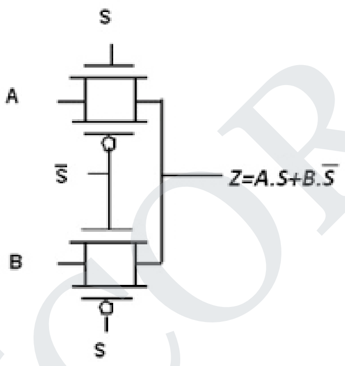
3. a. Write a brief note on pass Transistor circuits also explain about CMOS with Transmission gates. (may 2011,2013) (MAY/JUN 2016)

Pass Transistor Circuits:

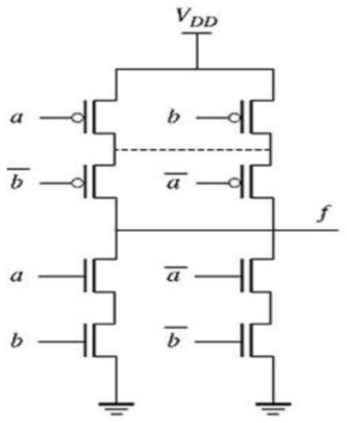
- In pass transistor circuits, inputs are also applied to the source/drain diffusion terminals.
- These circuits build switches using either n MOS pass transistor or parallel pairs of nM\OS and p MOS transistors called transmission gates.
- For example pass transistors are essential to the design of efficient 6 transistor static RAM cells used in most modern systems.
- Full address and other circuits rich in XOR s also can b efficiently constructed with pass transistors.

CMOS with Transmission Gates:

- Structures such as tristates, latches and multiplexers are often drawn as transmission gates in conjection with simple static CMOS Logic.
- The logic levels on the output are no better than those on the input so a cae of such circuits may accumulate Noise.
- To buffer the output and restore levels a static CMOS output inverter can be added.
- At first CMOS with transmission gates might appear to offer an entirely new range of circuits. The examination shows that the topology is almost identical to static CMOS.
- If multiple stages of logic are cae they can be viewed as alternating transmission gates an inverters.



- The above figure redraws the multiplexes to include the inverters from the previous that drive the diffusion input but to exclude in output inverter.
- The intermediate modes in the pull up and pull-down networks are shorted together as N_1 and N_2.



- The shorting of the intermediate nodes has two effects on delay.
- Since the output is pulled up or down through the parallel combination of both pass transistor rather than through a single transistor. The effective resistance will decreased.
- But the effective capacitance increases slightly because of extra diffusion and wire capacitance required for this shorting.
- There are several factors that favour the static CMOS representation over CMOS with transmission gates.
- It the inverter is on the output rather than the input, the delay of the gate depends on what is driving the input as well as the capacitance drivar by the output.
- The second drawback is that diffuse inputs to tristate invertors are susceptible to noise that may incorrectly turn on the inverter.
- Finally the contacts slightly increases are and their capacitance increases power consumption.
- The logical effort of circuits involving transmission gates is computed by drawing stage that begin at gate inputs rather than diffusion inputs.

Complementary pass Transistor Logic(CPL):

- CVSI is slow because one side of the gate pulls down, and then the cross coupled PMOs transistor pulls the other side up.
- The size of the cross coupled device is an inherent compromise between a large transistor that fights the pull down excessively and a small transistor that is slow pulling up.
- CPL resolves this problem by making on half of the gate pull up while the other half pulls down.
- In the CPL multiplexer. If a path consists of a cae of CPL gates, the inverters can be viewed equally well as being on the output of one stage or the input of the nest stage.
- If we redraws the mux to include the inverters from the previous stage that drives the diffusion input, but to exclude the output inverters.
- When the gate switches, one side pulls down well through its n MOS transistor.

- The other side pulls up.
- CPL can be constructed without cross coupled PMOS transistors, but the outputs would only to V_{DD}-V_t.
- Adding weak cross- coupled devices helps bring the rising output to the supply rail while only slightly slowing the falling output.

3. b. Explain about Pass-Transistor Logic. (MAY'13)

The implementation of the AND function constructed that way, using only NMOS transistors is shown in Figure 6.33. In this 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. The switch driven by B seems to be redundant at first glance. Its presence is essential to ensure that the gate is static; this is that a low-impedance path exists to the supply rails under all circumstances, or, in this particular case, when B is low.

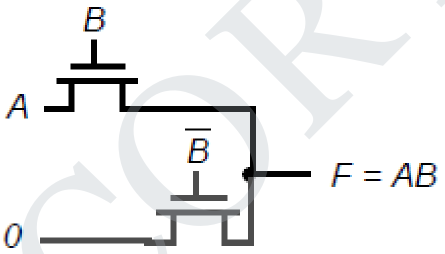


Figure 6.33 Pass-transistor implementation of an AND gate.

Differential Pass Transistor Logic

For high performance design, a differential pass-transistor logic family, called CPL or DPL, is commonly used. The basic idea (similar to DCVSL) is to accept true and complementary inputs and produce true and complementary outputs. These gates possess a number of interesting properties:

- XOR's and adders can be realized efficiently with small number of transistors.
- CPL belongs to the class of static gates
- Modular design

- All gates use same topology

Advantages

- Conceptually simple
- Modular logic style
- Applicability depends on logic function
- Easy to realize adders and multipliers

Disadvantages

- Has routing overhead
- Suffers static power dissipation
- Reduced noise margin

Efficient Pass-Transistor Design

Differential pass-transistor logic, like single-ended pass-transistor logic, suffers from static power dissipation and reduced noise margins, since the high input to the signal-restoring inverter only charges up to VDD-VTn. There are several solutions proposed to deal with this problem as outlined below.

Solution 1: Level Restoration: A common solution to the voltage drop problem is the use of a level restorer, which is a single PMOS configured in a feedback path (Figure 6.39). The gate of the PMOS device is connected to the output of the inverter, its drain connected to the input of the inverter and

the source to VDD. Assume that node X is at 0V (out is at VDD and the M_r is turned off) with B = VDD and A = 0. If input A makes a 0 to VDD transition, M_n only charges up node X to VDD-VTn. This is, however, enough to switch the output of the inverter low, turning on the feedback device M_r and pulling node X all the way to VDD. This eliminates any static power dissipation in the inverter. Furthermore, no static current path can exist through the level restorer and the pass-transistor, since the restorer is only active when A is high. In summary, this circuit has the advantage that all voltage levels are either at GND or VDD, and no static power is consumed.

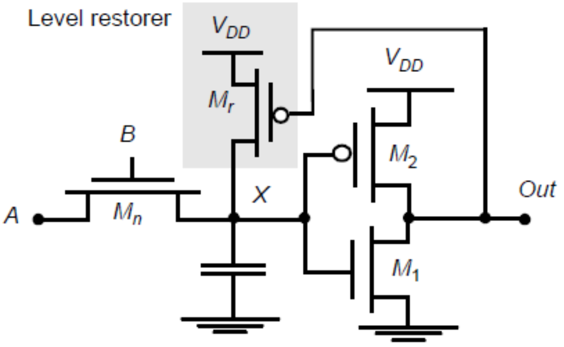


Figure 6.39 Level-restoring circuit.

Solution 2: Multiple-Threshold Transistors: A technology solution to the voltage-drop problem associated with pass-transistor logic is the use of multiple-threshold devices. Using zero threshold devices for the NMOS pass-transistors eliminates most of the threshold drop, and passes a signal close to VDD. Notice that even if the devices threshold was implanted to be exactly equal to zero, the body effect of the device prevents a swing to VDD. All devices other than the pass transistors (i.e., the inverters) are implemented using standard high-threshold devices.

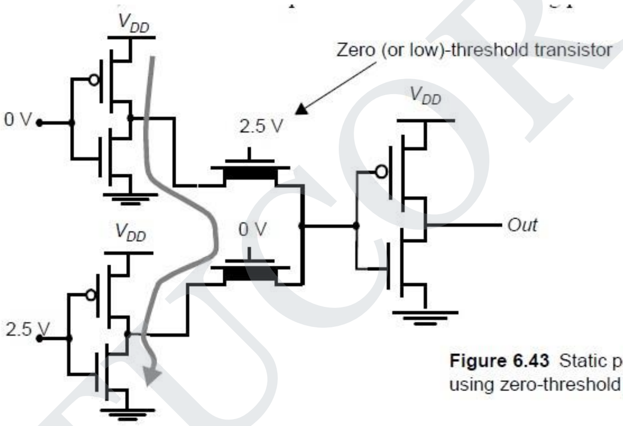


Figure 6.43 Static power consumption when using zero-threshold pass-transistors.

Solution 3: Transmission Gate Logic: The most widely-used solution to deal with the voltage- drop problem is the use of transmission gates. It builds on the complementary properties of NMOS and PMOS transistors: NMOS devices pass a strong 0 but a weak 1, while PMOS transistors pass a strong 1 but a weak 0. The ideal approach is to use an NMOS to pull-down and a PMOS to pull-up. This gate either selects input A or B based on the value of the control signal S, which is equivalent to implementing the following Boolean function:

$$\overline{F} = (A \cdot S + B \cdot \overline{S})$$

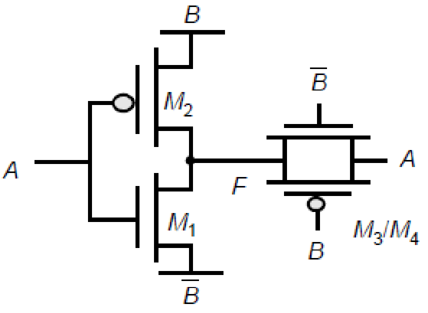


Figure 6.47 Transmission gate XOR.

A complementary implementation of the gate requires eight transistors instead of six.

4.a Explain the power dissipation present in VLSI circuits(APR/MAY 2010)(MAYJUN 2014)(APR/MAY 2015) (MAY/JUN 2016)

Static Power Consumption

Typically, all low-voltage devices have a CMOS inverter in the input and output stage. Therefore, for a clear understanding of static power consumption, refer to the CMOS inverter modes shown in Figure 1.

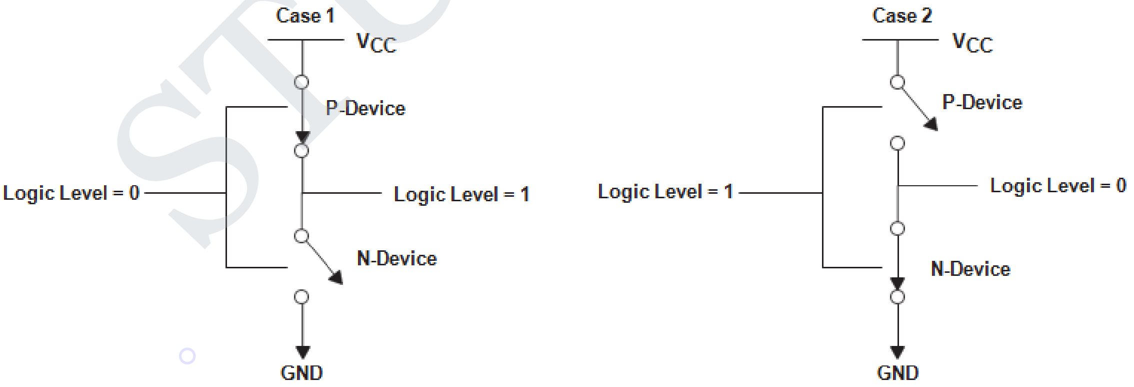


Figure 1. CMOS Inverter Mode for Static Power Consumption

As shown in Figure 1, if the input is at logic 0, the n-MOS device is OFF, and the p-MOS device is ON (Case 1). The output voltage is V_{CC}, or logic 1. Similarly, when the input is at logic 1, the associated n-MOS device is biased ON and the p-MOS device is OFF. The output voltage is GND, or logic 0. Note that one of the transistors is always OFF when the gate is in either of these logic states. Since no current flows into the gate terminal, and there is no dc current path from V_{CC} to GND, the resultant quiescent (steady-state) current is zero, hence, static power consumption (P_q) is zero.

However, there is a small amount of static power consumption due to reverse-bias leakage between diffused regions and the substrate. This leakage inside a device can be explained with a simple model that describes the parasitic diodes of a CMOS inverter, as shown in Figure 2.

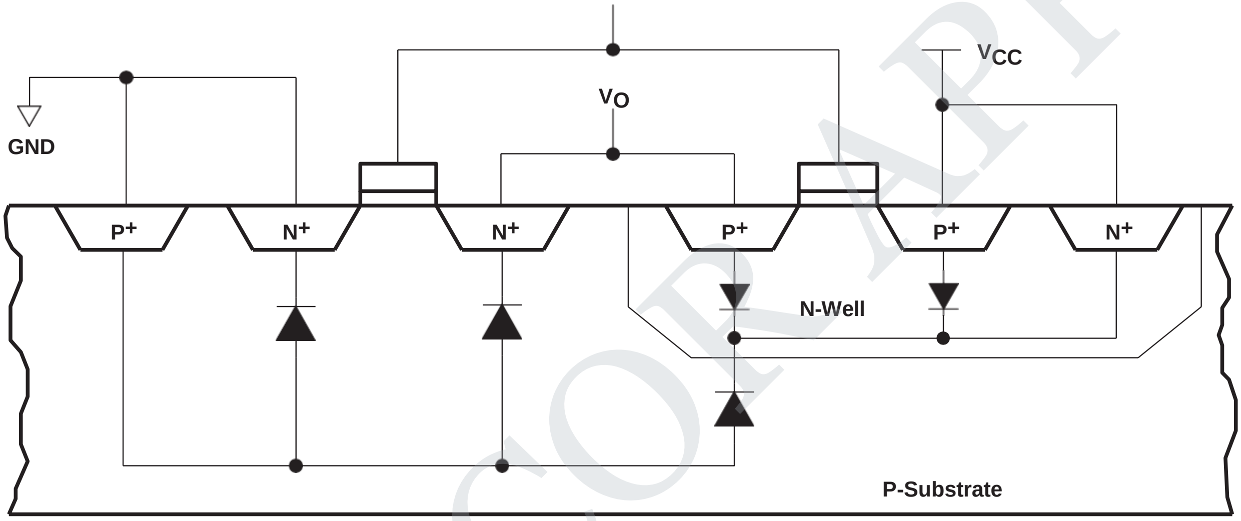


Figure 2. Model Describing Parasitic Diodes Present in CMOS Inverter

The source drain diffusion and N-well diffusion form parasitic diodes. In Figure 2, the parasitic diodes are shown between the N-well and substrate. Because parasitic diodes are reverse biased, only their leakage currents contribute to static power consumption. The leakage current (I_{lkg})of the diode is described by the following equation:

$$I_{lkg} \qquad i_s \;\; e^{qV \;\; kT} \qquad 1$$

Static power consumption is the product of the device leakage current and the supply voltage. Total static power consumption, P_S, can be obtained as shown in equation 2.

P_S (leakage current) (supply voltage)

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

The leakage current I_{CC} (current into a device), along with the supply voltage, causes static power consumption in the CMOS devices. This static power consumption is defined as quiescent, or P_S, and can be calculated by equation 3.

$$\mathbf{P_S = V_{CC} \times I_{CC}} \qquad \mathbf{(3)}$$

V_{CC} =supply voltage

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

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

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

Transient Power Consumption

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

Dynamic supply current is dominant in CMOS circuits because most of the power is consumed in moving charges in the parasitic capacitor in the CMOS gates. As a result, the simplified model of a CMOS circuit consisting of several gates can be viewed as one large capacitor that is charged and discharged between the power-supply rails. Therefore, the power–dissipation capacitance (C_{pd}) is often specified as a measure of this equivalent capacitance and is used to approximate the dynamic power consumption. C_{pd} is defined as the internal equivalent capacitance of a device calculated by measuring operating current without load capacitance. Depending on the output switching capability, C_{pd} can be measured with no output switching (output disabled) or with any of the outputs switching (output enabled). C_{pd} is discussed in greater detail in the next section.

4.b. Explain the various ways to minimize static and dynamic power dissipation. (NOV/DEC 2014) (NOV/DEC 2013) (APR/MAY 2010) (MAY/JUN 2016)

LOW POWER DESIGN PRINCIPLES

The supply voltage for CMOS processes will continue to drop over the coming decade, and may go as low as 0.6V by 2010. To maintain performance under those conditions, it is essential that the device thresholds scale as well.

Figure a shows a plot of the (V_T, V_{DD}) ratio required to maintain a given performance level (assuming that other device characteristics remain identical). This trade-off is not without penalty. Reducing the threshold voltage, increases the subthreshold leakage current exponentially .

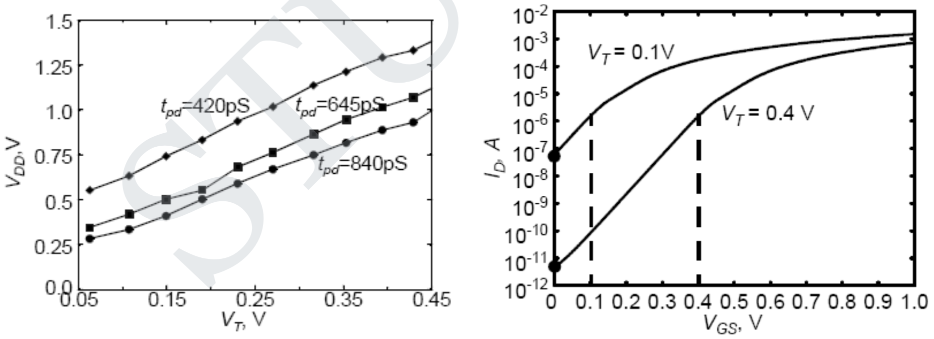


Figure: Voltage Scaling (V_{DD}/V_T on delay and leakage) (a) V_{DD}/V_T for fixed performance (b) Leakage as a function of V_T

$$I_{leakage} = I_S 10^{\frac{V_{GS}-V_{Th}}{S}}\left(1-10^{-\frac{nV_{DS}}{S}}\right)$$

with S the *slope factor* of the device. The subthreshold leakage of an inverter is the current of the NMOS for V_{in} = 0V and $V_{out} = V_{DD}$ (or the PMOS current for $V_{in} = V_{DD}$ and V_{out} = 0).

The exponential increase in inverter leakage for decreasing thresholds illustrated in Figure b.

These leakage currents are particularly a concern for designs that feature intermittent computational activity separated by long periods of inactivity. For example, the processor in a cellular phone remains in idle mode for a majority of the time. While the processor is shutdown mode, the system should ideally consume zero or near-zero power. This is only possible if leakage is low—this is, the devices have a high threshold voltage. This is in contradictory to the scaling scenario that we just depicted, where high performance under low supply voltage means reduced thresholds. To satisfy the contradicting requirements of high-performance during active periods, and low leakage during standby, several process modifications or leakage-control techniques have been introduced in CMOS processes. Most processes with feature sizes at and below 0.18 mm CMOS support devices with different thresholds—typically a device with low threshold for high performance circuits, and a transistor with high threshold for leakage control. Another approach that is gaining ground is the dynamic control of the threshold voltage of a device by exploiting the body effect of the transistor. To use this approach for the control of individual devices requires a dual-well process.

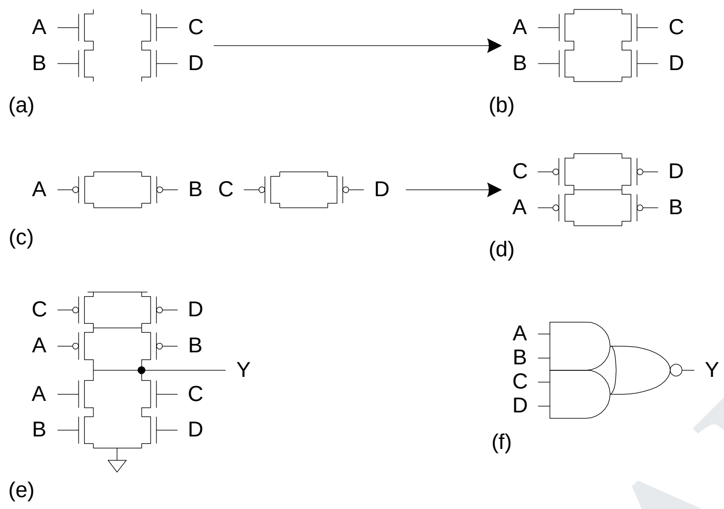
Clever circuit design can also help to reduce the leakage current, which is a function of the circuit topology and the value of the inputs applied to the gate. Since V_T depends on body bias (V_{BS}), the sub-threshold leakage of an MOS transistor depends not only on the gate drive (V_{GS}), but also on the body bias. In an inverter with In = 0, the sub-threshold leakage of the inverter is set by the NMOS transistor with its V_{GS} = V_{BS} = 0 V. In more complex CMOS gates, the leakage current depends upon the input vector. For example, the sub-threshold leakage current of a two-input NAND gate is the least when A=B=0. Under these conditions, the intermediate node X settles to,

$$V_X \approx V_{th} \ln(1+n)$$

The NAND gate sub-threshold leakage is then set by the top-most NMOS transistor with V_{GS}=V_{BS}=-VX. Clearly, the sub-threshold leakage under this condition is slightly smaller than that of the inverter. This reduction in sub-threshold leakage due to stacked transistors is called

5. Draw the static CMoS logic circuit for the given expression. Y = (A.B + C.D)'

(MAY/JUN 2016)



Step 1: Fig a shows the logic design for **A.B and C.D** using nMoS transistors.

Step 2: Fig b shows the combination of both the AND gates using nMoS transistors. ie,

Y=A.B+C.D

Step 3: Fig c shows the logic design for **A.B and C.D** using pMoS transistors.

Step 4: Fig d shows the combination of both the AND gates using pMoS transistors. ie,

Y=A.B+C.D

Step 5: Fig e denotes the inverted operation for the expression Y=A.B+C.D. This is obtained by connecting both the pMoS and nMoS in series.

ie, Y=(A.B+C.D)'

UNIT III

SEQUENTIAL LOGIC CIRCUITS

PART A

1. **What is synchronous sequential circuit?**

   If all the registers are controlled by clock signal, then the circuit is called synchronous sequential logic circuit.

2. **What is bistability principle?**

   Bistable state has two stable states. The two stable states are o and one.

3. **What is metastable?**

   If the cross coupled inverter pair is biased at point **C** and small deviation at this point caused by noise is amplified and regenerated around the circuit loop. This small deviation is amplified by both the inverters and the bias point **C** moves the operation points A and B. so the bias point is unstable. This property is called metastable.

4. **List the timing parameters of registers.**

   1. Set up time
   2. Propagation delay
   3. Hold time

5. **What is race condition?**

   During the 0-0 overlap period, NMOS of t1 and PMOS t2 are simultaneously ON. This creates a direct path for data to flow from D input of the register to the Q output. This is called race condition.

6. **List the drawbacks of static latches and registers.**

   The drawbacks of static latches and registers are

   1. Stored value remains valid as long as supply voltage is available.
   2. Complexity

7. Define global clock.

In synchronous circuits all the memory elements have a globally distributed periodic synchronous signal called global clock.

8. What is clock skew?

Clock skew is defined as the spatial variation in arrival time of clock transition on an integrated circuit. The clock skew between two points i and j on an IC.

9. What is clock jitter?

Clock jitter is defined as the temporal variation of the clock period at a given point on the chip. The clock period can reduce or expand on a cycle –by- cycle basis.

10. Define pipelining.

Pipelining is a designing technique used to increase the operation of datapaths in digital processor.

11. Define Propagation delay (tpd)?

This value indicates the amount of time needed for a change in a logic input to result in a **permanent** change at an output. Combinational logic is guaranteed not to show any further output changes in response to an input change **after** tpd time units have passed.

12. Define Contamination delay (tcd)?

This value indicates the amount of time needed for a change in a logic input to result in an **initial** change at an output. Combinational logic is guaranteed not to show any output change in response to an input change **before** tcd time units have passed.

13. What do you mean by Setup time (ts)?

This value indicates the amount of time before the clock edge that data input D must be stable. As shown in Figure 4, D is stable ts time units before the rising clock edge.

14. What do you mean by Hold time (th)?

This value indicates the amount of time after the clock edge that data input D must be held stable. As shown in Figure 4, the hold time is always measured from the rising clock edge (for positive edge-triggered) to a point after the edge.

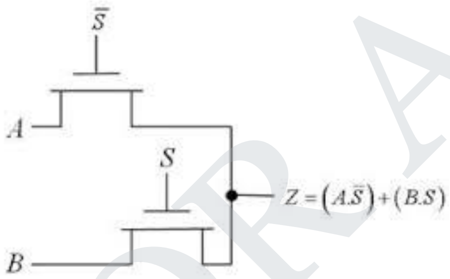
15.What is Time Borrowing?

If one half-cycle or stage of a pipeline has too much logic, it can borrow time into the next half-cycle or stage. Time borrowing can accumulate across multiple cycles.

16. What is clocked CMoS register? (MAY/JUN 2016)

In integrated circuit design, **dynamic logic** (or sometimes **clocked logic**) is a design methodology in combinatory logic circuits, particularly those implemented in MOS technology.

17. Draw the switch level schematic of multiplexer based nMoS latch using nMoS only pass transistors for multiplexers. (MAY/JUN 2016)



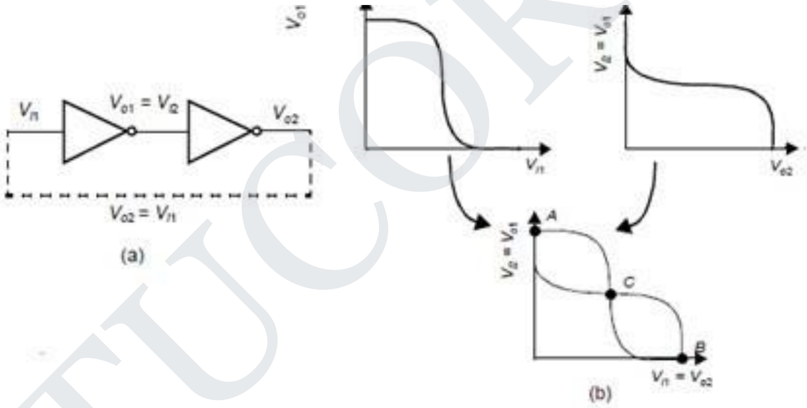
PART B

1. Explain about static latches in detail.

A latch is an essential component in the construction of an edge-triggered register. It is level-sensitive circuit that passes the D input to the Q output when the clock signal is high. This latch is said to be in transparent mode. When the clock is low, the input data sampled on the falling edge of the clock is held stable at the output for the entire phase, and the latch is in hold mode.. A latch operating under the above conditions is a positive latch.

Static Latches Principle:

Static memories use positive feedback to create a bistable circuit — a circuit having two stable states that represent 0 and 1. The basic idea is shown in below figure which shows two inverters connected in cae along with a voltage-transfer characteristic typical of such a circuit. Also plotted are the VTCs of the first inverter, that is,V o1 versus Vi1, and the second inverter (Vo2 versus Vo1).



The latter plot is rotated to accentuate thatVi2 = Vo1. Assume now that the output of the second inverter Vo2 is connected to the input of the first Vi1, as shown by the dotted lines in Figure 7.4a. The resulting circuit has only three possible operation points (A, B, and C), as demonstrated on the combined VTC.

Multiplexer Based Latches

There are many approaches for constructing latches. One very common technique involves the use of transmission gate multiplexers. Multiplexer based latches can provide similar functionality to the SR latch, but has the important added advantage that the sizing of devices only affects performance and is not critical to the functionality.

Figure 7.11 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, which connects to the output of the latch, is selected. 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.

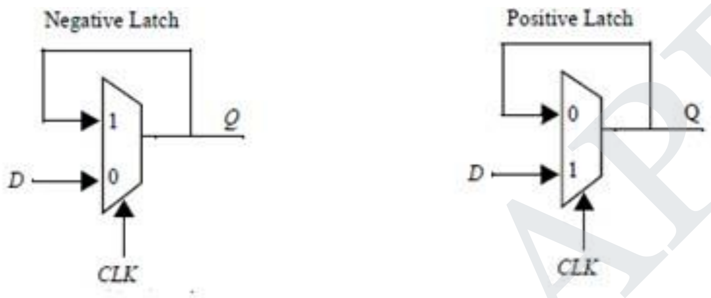


Fig: Negative and Positive latches based on multiplexer

A transistor level implementation of a positive latch based on multiplexers is shown in figure. When CLK is high, the bottom transmission gate ison and the latch is transparent - that is, the D input is copied to the Q output. During this phase, the feedback loop is open since the top transmission gate is off.

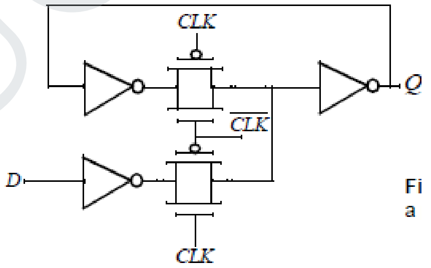


Fig: Transistor level implementation of a positive latch built using transmission gates

Unlike the SR FF, the feedback does not have to be overridden to write the memory and hence sizing of transistors is not critical for realizing correct functionality. The number of transistors that the clock touches is important since it has an activity factor of 1. This particular latch implementation is not particularly efficient from this metric as it presents a load of 4 transistors to the CLK signal.

It is possible to reduce the clock load to two transistors by using implement multiplexers using NMOS only pass transistor as shown in Figure 7.13. The advantage of this approach is the reduced clock load of only two NMOS devices. When CLK is high, the latch samples the D input, while a low clock-signal enables the feedback-loop, and puts the latch in the hold mode.

Low-Voltage Static Latches

The scaling of supply voltages is critical for low power operation. Unfortunately, certain latch structures don't function at reduced supply voltages. For example, without the scaling of device thresholds, NMOS only pass transistors don't scale well with supply voltage due to its inherent threshold drop. At very low power supply voltages, the input to the inverter cannot be raised above the switching threshold, resulting in incorrect evaluation. Even with the use of transmission gates, performance degrades significantly at reduced supply voltages.

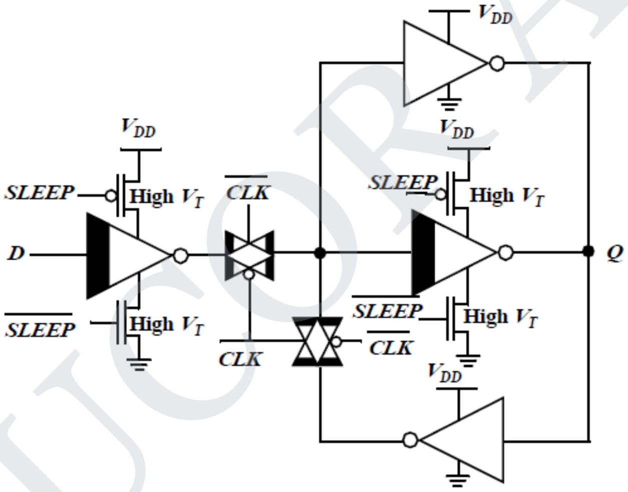


Fig: one solution for the leakage problem in low-voltage operation using MTCMOS

Scaling to low supply voltages hence requires the use of reduced threshold devices. However, this has the negative effect of exponentially increasing the sub-threshold leakage Power. age energy is typically insignificant compared to the switching power. However, with the use of conditional clocks, it is possible that registers are idle for extended periods and the leakage energy expended by registers can be quite significant.

Many solutions are being explored to address the problem of high leakage during idle periods. One approach for this involves the use of Multiple Threshold devices as shown in above figure only the negative latch is shown here. 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. When clock is low, the D input is sampled and propagates to the output. When clock is high, the latch is in the hold mode. The feedback transmission gate conducts and the cross-coupled feedback is enabled. Note there is an extra inverter, needed for storage of state when the latch is in the sleep state. During idle mode, the high threshold devices in series with the low threshold inverter are turned off (the SLEEP signal is high), eliminating leakage. It is assumed that clock is in the high state when the latch is in the sleep state. The feedback low-threshold transmission gate is turned on and the cross-coupled high-threshold device maintains the state of the latch.

2. **Explain about the concept of pipelining in detail (Dec-2012, May2014) (MAY/JUN 2016)**

Pipelining is a popular design technique often used to accelerate the operation of the datapaths in digital processors. The idea is easily explained with the example of below figure.

The goal of the presented circuit is to compute log (|a - b|), where both a and b represent streams of numbers, that is, the computation must be performed on a large set of input values.

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.

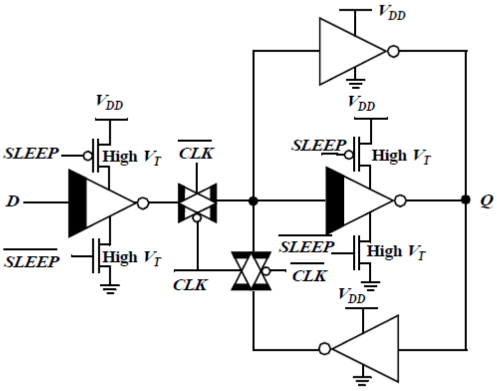


Fig: Datapath for computation

We assume that the registers are edge-triggered D registers. The term tpd, logic stands

for the worst-case delay path through the combinatorial network, this consists of the adder, absolute value, and logarithm functions. In conventional systems (that don't push the edge of technology), the latter delay is generally much larger than the delays associated with the registers and dominates the circuit performance. Assume that each logic module has an equal propagation delay. We note that each logic module is then active for only 1/3 of the clock period (if the delay of the register is ignored).

For example, the adder unit is active during the first third of the period and remains idle— this is, it does no useful computation— during the other 2/3 of the period. Pipelining is a technique to improve the resource utilization, and increase the functional throughput. Assume that we introduce registers between the logic blocks.

The result for the data set (a1, b1) only appears at the output after three clock-periods. 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 advantage of pipelined operation becomes apparent when examining the minimum clock period of the modified circuit. The combinational circuit block has been partitioned into three sections, each of which has a smaller propagation delay than the original function. This effectively reduces the value of the minimum allowable clock period.

Latch- vs. Register-Based Pipelines

Pipelined circuits can be constructed using level-sensitive latches instead of edge-triggered registers. Consider the pipelined circuit of below figure. The pipeline system is implemented based on pass-transistor-based positive and negative latches instead of edge triggered registers. That is, logic is introduced between the master and slave latches of a Master- slave system.

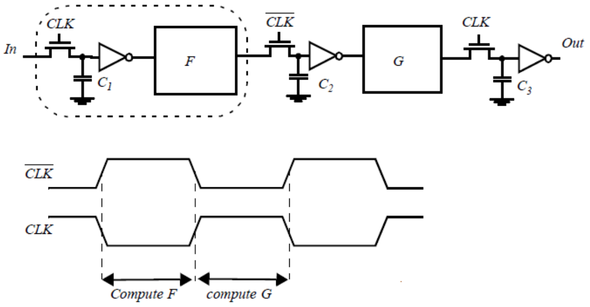


Fig: Operation of two-phase pipelined circuit using dynamic registers

In the following discussion, we use without loss of generality the CLK-CLK notation to denote a two-phase clock system. Latch-based systems give significantly more flexibility in implementing a pipelined system, and often offer higher performance.

When the clocks CLK and CLK are non-overlapping, correct pipeline operation is obtained. Input data is sampled on C1 at the negative edge of CLK and the computation of logic block F starts; the result of the logic block F is stored on C2 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 C2 at the end of the CLK low phase is the result of passing the previous input (stored on the falling edge of C LK on C1) through the logic function F.

When overlap exists between CLK and CLK, the next input is already being applied to F, and its effect might propagate to C2 before CLK goes low (assuming that the contamination delay of F is small). Which value wins depends upon the logic functionF , the overlap time, and the value of the inputs since the propagation delay is often a function of the applied inputs.

NORA-CMOS— A Logic Style for Pipelined Structures

The latch-based pipeline circuit can also be implemented using CMOS latches, as shown in Figure. The operation is similar to the one discussed above. This topology has one additional, important property:

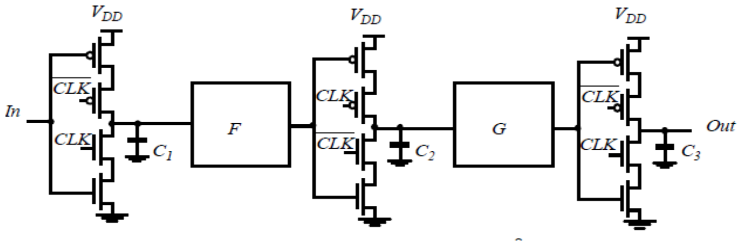


Fig: Pipeline datapath using CMOS latches.

A CMOS-based pipelined circuit is race-free as long as all the logic functions F (implemented using static logic) between the latches are non inverting. The reasoning for the above argument is similar to the argument made in the construction of a C2MOS register. During a (0-0) overlap between CLK and CLK, all C2MOS latches, simplify to pure pull-up networks

.The only way a signal can race from stage to stage under this condition is when the logic function F is inverting, as illustrated in below figure, where F is replaced by a single, static CMOS inverter.

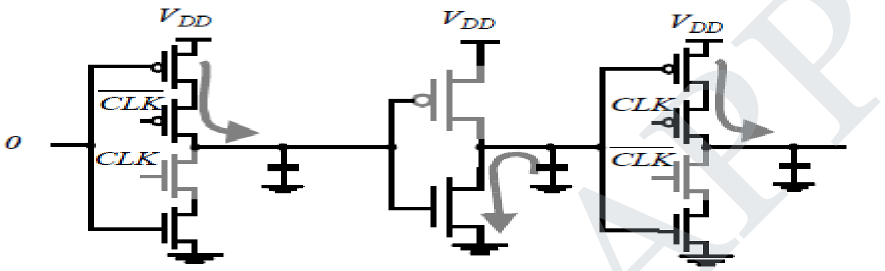


Fig : Potential race condition during (0-0) overlap in CMOS-based design.

Similar considerations are valid for the (1-1) overlap. Based on this concept, a logic circuit style called NORA-CMOS; it combines CMOS 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 CMOS latch. 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 CLK- module.

A NORA datapath consists of a chain of alternating CLK and CLK modules. While one class of modules is precharging with its output latch in hold mode, preserving the previous output value, the other class is evaluating. Data is passed in a pipelined fashion from module to module.

NORA offers designers a wide range of design choices. Dynamic and static logic can be mixed freely, and both CLKp and CLKn dynamic blocks can be used in caed or in pipelined form. With this freedom of design, extra inverter stages, as required in DOMINO-CMOS, are most often avoided.

In order to ensure correct operation, two important rules should always be followed:

- **The dynamic-logic rule:** Inputs to a dynamic CLKn (CLKp) block are only allowed to make a single 0□1 (1□0) transition during the evaluation period.

- **The CMOS rule:** In order to avoid races, the number of static inversions between CMOS latches should be even

3. Explain about Dynamic latches in detail. (MAY/JUN 2016)

Dynamic Transmission-Gate Based Edge-triggered Registers

A fully dynamic positive edge-triggered register based on the master-slave concept is shown in figure. When CLK = 0, the input data is sampled on storage node 1, which has an equivalent capacitance of C1 consisting of the gate capacitance of I1, the junction capacitance of T1, and the overlap gate capacitance of T1.

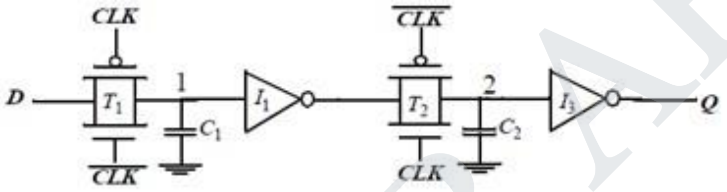


Fig: Dynamic Edge-triggered Registers

During this period, the slave stage is in a hold mode, with node 2 in a high-impedance (floating) state. On the rising edge of clock, the transmission gate T2 turns on, and the value sampled on node 1 right before the rising edge propagates to the output Q (note that node 1 is stable during the high phase of the clock since the first transmission gate is turned off).

Node 2 now stores the inverted version of node 1. This implementation of an edge-triggered register is very efficient as it requires only 8 transistors.

The sampling switches can be implemented using NMOS-only pass transistors, resulting in an even-simpler 6 transistor implementation. The reduced transistor count is attractive for high-performance and low-power systems.

The set-up time of this circuit is simply the delay of the transmission gate , and corresponds to the time it takes node 1 to sample the D input.The hold time is approximately zero, since the transmission gate is turned off on the clock edge and further inputs changes are ignored. The propagation delay (tc-q) is equal to two inverter delays plus the delay of the transmission gate T2.

One important consideration for such a dynamic register is that the storage nodes (i.e., the state) has to be refreshed at periodic intervals to prevent a loss due to charge leakage, due to diode leakage as well as sub-threshold currents. In datapath circuits, the refresh rate is not an issue since the registers are periodically clocked, and the storage nodes are constantly updated.

Clock overlap is an important concern for this register. Consider the clock waveforms shown in below figure. During the 0-0 overlap period, the NMOS of T1 and the PMOS of T2 are simultaneously on, creating a direct path for data to flow from the D input of the register to the Q output. This is known as race condition. The output Q can change on the falling edge if the overlap period is large

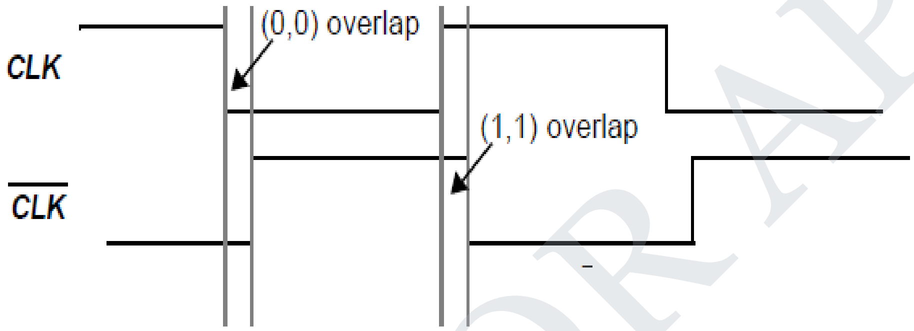


Fig : Impact of non-overlapping clocks

CMOS Dynamic Register: A Clock Skew Insensitive Approach The CMOS Register

The following shows an ingenious positive edge-triggered register based on the master- slave concept which is insensitive to clock overlap. This circuit is called the C2MOS (Clocked CMOS) register. The register operates in two phases.

1. CLK = 0 (CLK = 1):

2. The roles are reversed when CLK = 1:

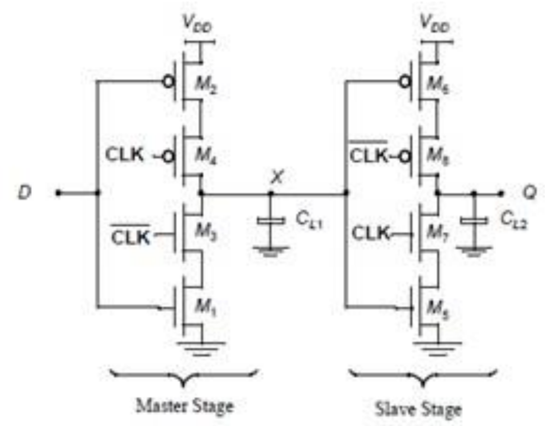


Fig: master slave edge-triggered register

It can be stated that the C2MOS latch is insensitive to clock overlaps because those overlaps activate either the pull-up or the pull-down networks of the latches, but never both of them simultaneously. If the rise and fall times of the clock are sufficiently slow, however, there exists a time slot where both the NMOS and PMOS transistors are conducting. This creates a path between input and output that can destroy the state of the circuit.

Dual-edge Triggered Registers

So far, we have focused on edge-triggered registers that sample the input data on only one of the clock edges (rising or falling). It is also possible to design sequential circuits that sample the input on both edges. The advantage of this scheme is that a lower frequency clock (half of the original rate) is distributed for the same functional throughput, resulting in power savings in the clock distribution network.

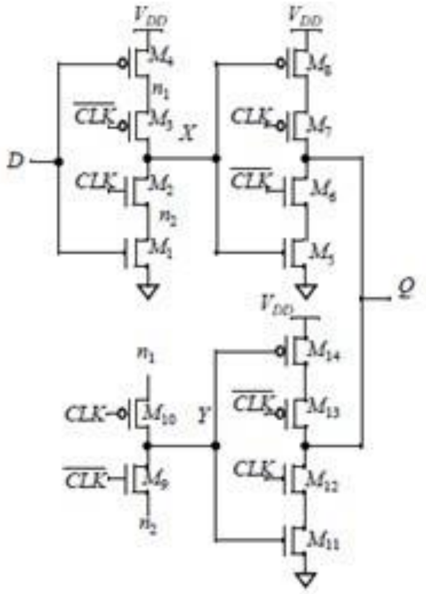


Fig: CMOS based dual-edge triggered register.

The above figure shows a modification of the C2MOS register to enable sampling on both edges.

True Single-Phase Clocked Register (TSPCR)

In the two-phase clocking schemes described above, care must be taken in routing the two clock signals to ensure that overlap is minimized. While the C2MOS provides a skew- tolerant solution, it is possible to design registers that only use a single phase clock. The basic single-phase positive and negative latches are shown in figure.

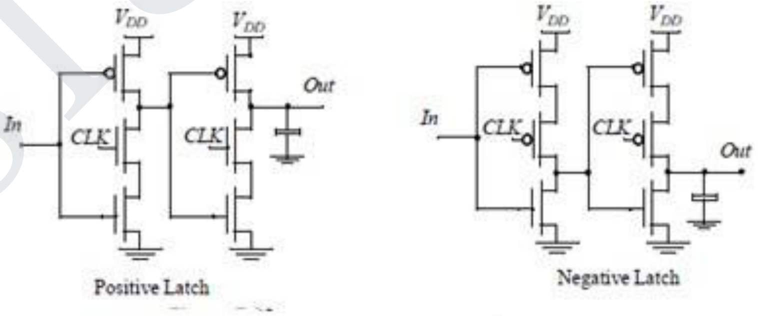
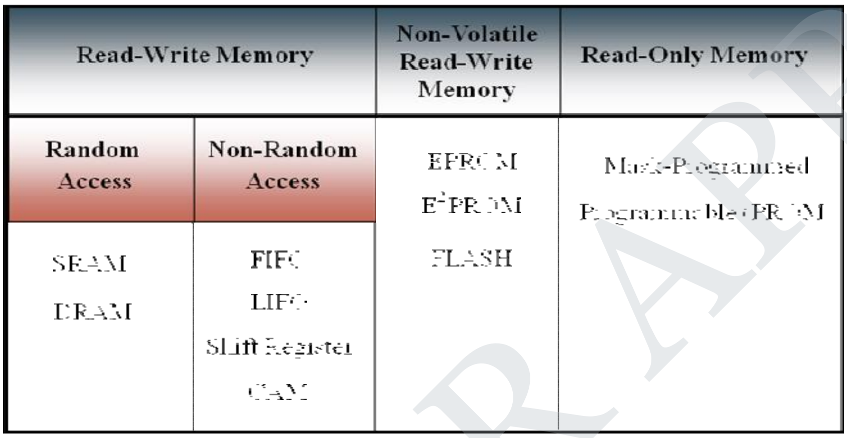


Fig : True Single-Phase Clocked Register

For the positive latch, when CLK is high, the latch is in the transparent mode and corresponds to two caed inverters; the latch is non-inverting, and propagates the input to the output. On the other hand, when C LK = 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 of the latch to the output in this mode. A register can be constructed by caing positive and negative latches.

4. Explain the memory architecture in detail. Memory Classification:



Read-Write Memory		Non-Volatile Read-Write Memory	Read-Only Memory
Random Access	Non-Random Access	EPROM E^{2}PROM FLASH	Mask-Programmed Programmable (PROM)
SRAM DRAM	FIFO LIFO Shift Register CAM		

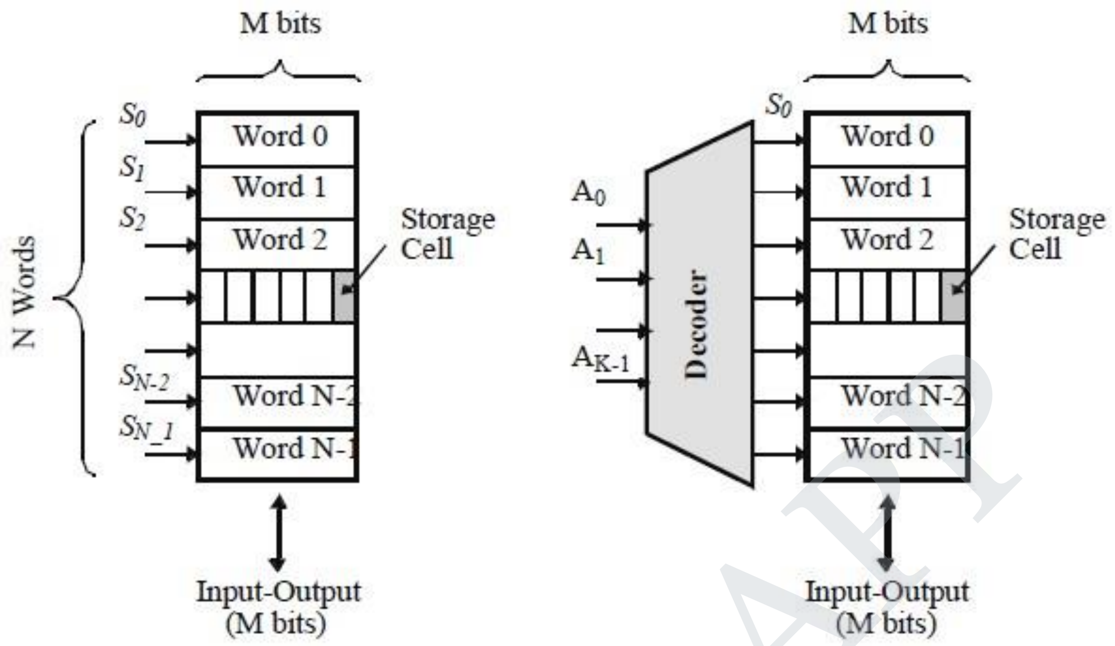
STATIC (SRAM):

- Data stored as long as supply is applied
- Large (6 transistors/cell)
- Fast
- Differential

DYNAMIC (DRAM):

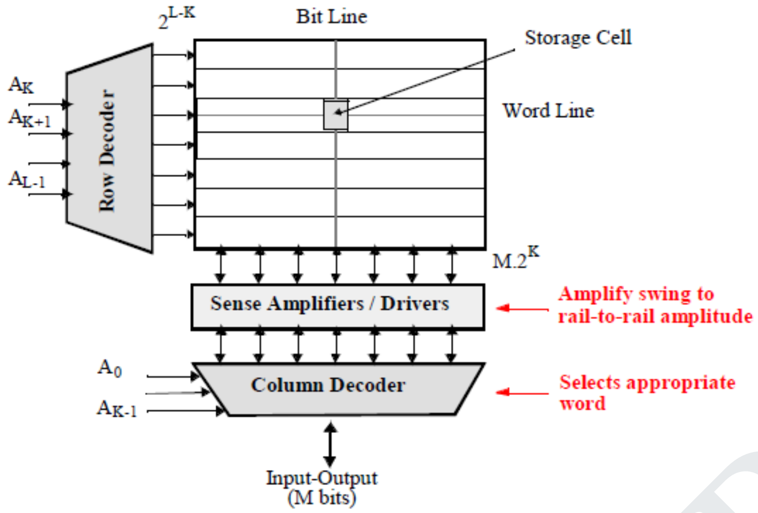
- Periodic refresh required
- Small (1-3 transistors/cell)
- Slower
- Single Ended

Memory Architecture: Decoders:

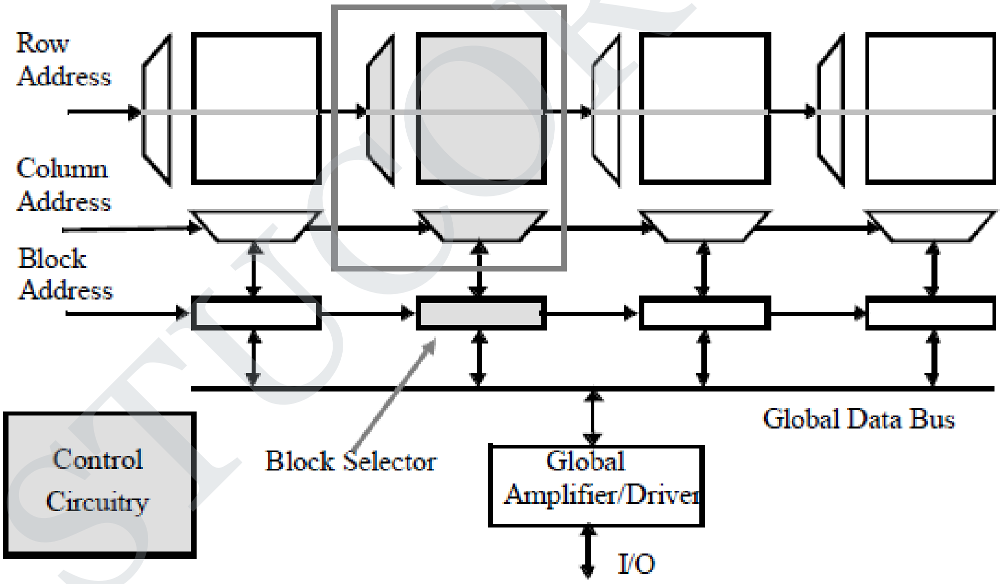


N words => N select signals
Too many select signals

Decoder reduces # of select signals
$K = log_2N$



Array-Structured Memory Architecture:



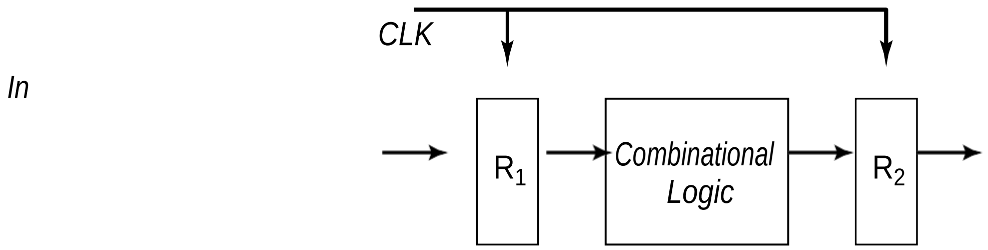
Hierarchical Memory Architecture:

Advantages:

- Shorter wires within blocks
- Block address activates only one block hence, power savings.

5. Explain in detail about synchronous circuit design.

Synchronous Timing:



The following timing parameters characterize the timing of the sequential circuit.

- The contamination (minimum) delay $t_{c-q,cd}$, and maximum propagation delay of the register t_{c-q}.
- The set-up (t_{su}) and hold time (t_{hold}) for the registers.
- The contamination delay $t_{logic,cd}$ and maximum delay t_{logic} of the combinational logic.
- t_{clk1} and t_{clk2}, corresponding to the position of the rising edge of the clock relative to a global reference.

Clock Non idealities:

Clock skew

✓ Spatial variation in temporally equivalent clock edges; deterministic + random, t_{SK}

Clock jitter

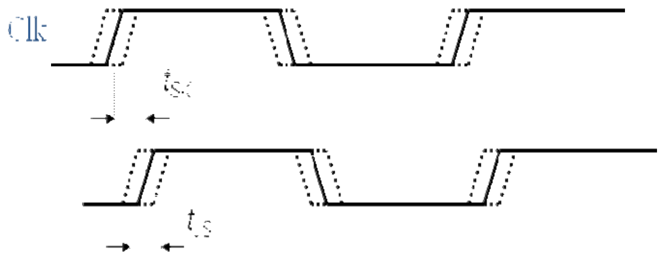
✓ Temporal variations in consecutive edges of the clock signal; modulation + random noise

✓ Cycle-to-cycle (short-term) t_{JS} Long term t_{JL}

Variation of the pulse width

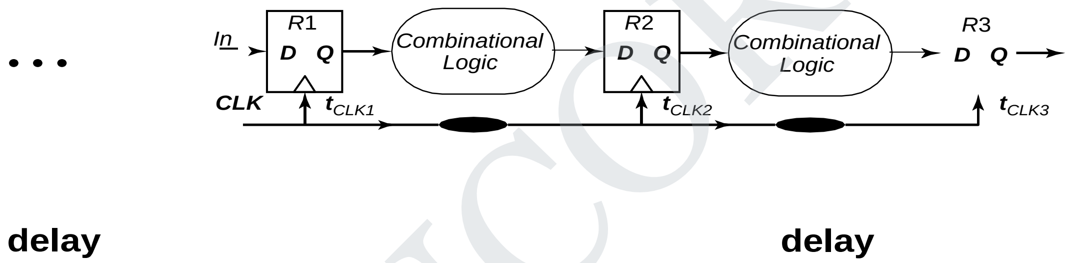
✓ Important for level sensitive clocking

Clock Skew and Jitter:

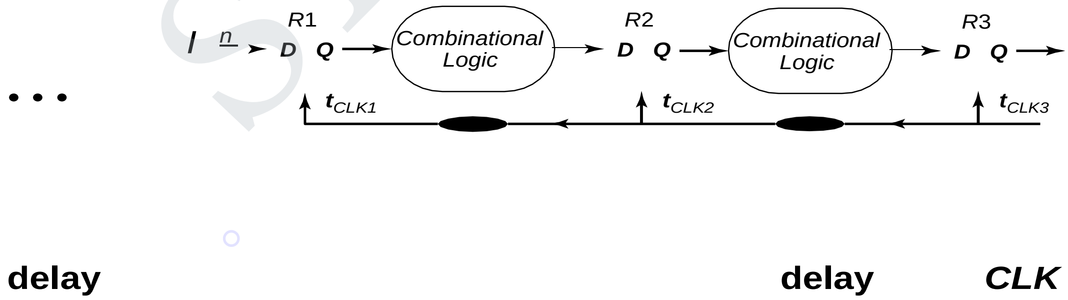


- ✓ Both skew and jitter affect the effective cycle time
- ✓ Only skew affects the race margin

Positive and Negative Skew:



(a) Positive skew



(b) Negative skew

Sources of Skew and Jitter:

- **Clock-Signal Generation-** The generation of the clock signal itself causes jitter
- **Manufacturing Device Variations**
- **Interconnect Variations-**One important source of interconnect variation is the Inter-level Dielectric (ILD) thickness variations.
- **Environmental Variations-**Environmental variations are probably the most significant and primarily contribute to skew and jitter. The two major sources of environmental variations are temperature and power supply. Power supply variations is the major source of jitter in clock distribution networks.
- **Capacitive Coupling-**The variation in capacitive load also contributes to timing uncertainty. There are two major sources of capacitive load variations: coupling between the clock lines and adjacent signal wires and variation in gate capacitance.

Clock-Distribution Techniques:

It is necessary to design a clock network that minimizes skew and jitter. Another important consideration in clock distribution is the power dissipation.

Fabrics for clocking-one common approach to distributing a clock are to use balanced paths (or called trees). The most common type of clock primitive is the H-tree network (named for the physical structure of the network) .

In this scheme, the clock is routed to a central point on the chip and balanced paths, that include both matched interconnect as well as buffers, are used to distribute the reference to various leaf nodes. Ideally, if each path is balanced, the clock skew is zero.

That is, though it might take 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.

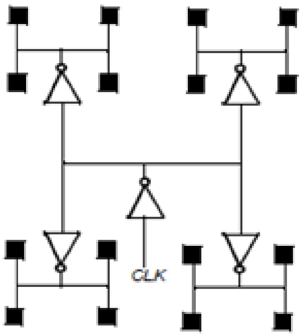


Fig: Example of H-tree clock distribution

UNIT IV:

DESIGNING ARITHMETIC BILDING BLOCKS

PART-A

1. **Define datapath circuits.**

Datapath circuits use N identical circuits to process N-bit data. Related data operators are placed physically adjacent to each other to reduce wire length and delay.

2. **What is ripple carry adder?**

An N-bit adder can be constructed by caing N full adders. This is called a carry-ripple adder (or ripple-carry adder). The carry-out of bit i, C_i is the carry-in to bit i

+ 1. This carry has twice the weight of the sum S_i. The delay of the adder is set by the time for the carries to ripple through the N stages, so the $t_{C \to C_{OUT}}$ delay is minimized.

3. **What is the need of carry lookahead adder?**

The carry-look ahead adder (CLA) computes group generate signals as well as group propagate signals to avoid waiting for a ripple to determine if the first group generates a carry.

4. **List some high speed adders.**

- Carry-skip adder
- Carry-select adder
- Carry-save adder

5. Give an example of binary multiplication

```
       011001  :  25₁₀      multiplicand
     × 100111  :  39₁₀      multiplier
       011001
      011001                partial
     011001                 products
    000000
   000000
 +011001
 001111001111  :  975₁₀     product
```

6. Define Booth encoding.

The speed of the multiplication can be increased by using a special encoding called booth encoding of the multiplier word that reduces the number of required addition stages. Instead of traditional binary encoding the multiplier word is recoded into radix-4 scheme.

$$Y = \sum_{j=0}^{(N-1)/2} Y_j 4^j \text{ with } (Y_j \in \{-2,-1,0,1,2\}$$

7. What are the two types of dividers?

- Serial divider
- Parallel divider

8. Draw the schematic diagram of logarithmic barrel shifter.

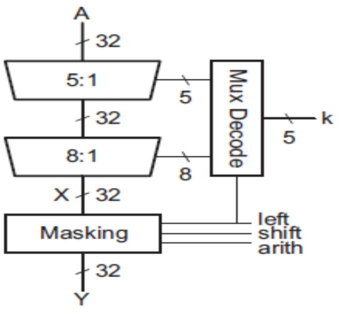


Fig. 32-bit logarithmic barrel shifter

9. What is the advantage of a carry-skip adder?

Carry-skip adder speeds up a wide adder by aiding the propagation of a carry bit around a portion of the entire adder.

10. What is the disadvantage of ripple-carry adder?

In ripple carry adder, every full-adder cell has to wait for the incoming carry before an outgoing carry is generated. This creates a linear dependency.

11. How to overcome the disadvantage of ripple-carry adder?

The disadvantage of ripple-carry adder is eliminated by the use of carry-select adder. It anticipates both possible values of the carry input and evaluates the result for both possibilities in advance. Once the real value of the incoming carry is known, the correct result is easily selected with a multiplexer.

12. What is meant by bit sliced data path organization? (MAY/JUN 2016)

A data path circuit is a circuit that combines two functions to a single logic cell. For instance, consider to design a full adder: ADD is a function that combines two inputs. Therefore in general, the layout of buswide logic that operates on data signals is called as a data path. The module add in a full adder is a data path.

13. Determine the propagation delay of n-bit carry select adder. (MAY/JUN 2016)

The propagation delay of N.bit carry select adder is given by

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

t_{setup} = Initial time taken to create the propagate and generate signals.

t_{carry} = defines the propagation delay through the single bit

t_{mux} = delay incurred multiplexer for a single stage

t_{sum} = defines the total time to generate the sum of the final stage

PART-B

1. Explain in detail about ripple carry adder.

An N-bit adder can be constructed by caing N full adders as shown in Fig.4.1 (a) for N=4. This is called a carry-ripple adder (or ripple-carry adder). The carry-out of bit i, C_i is the carry-in to bit i + 1. This carry is said to have twice the weight of the sum S_i. The delay of the adder is set by the time for the carries to ripple through the N stages, so the $t_{C \rightarrow C_{OUT}}$delay should be minimized.

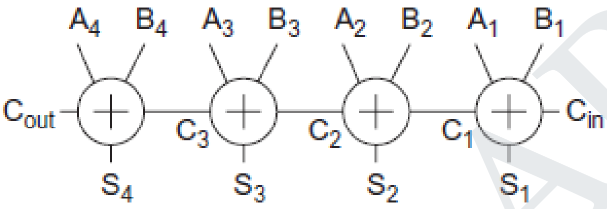


Fig.4.1 (a) 4-bit carry-ripple adder

In carry-ripple adders, the critical path goes from C to Cout through many full adders, so the extra delay computing S is unimportant. This delay can be reduced by omitting the inverters on the outputs. Fig.4.1 (b) shows the adder with transistor sizes optimized to favor the critical path using a number of techniques:

- Feed the carry-in signal (C) to the inner inputs so the internal capacitance is already discharged.
- Make all transistors in the sum logic whose gate signals are connected to the carry-in and carry logic minimum size (1 unit, e.g., 4λ). This minimizes the branching effort on the critical path. Keep routing on this signal as short as possible to reduce interconnect capacitance.
- Determine widths of series transistors by logical effort and simulation. Build an asymmetric gate that reduces the logical effort from C to Cout at the expense of effort to S.
- Use relatively large transistors on the critical path so that stray wiring capacitance is a small fraction of the overall capacitance.
- Remove the output inverters and alternate positive and negative logic to reduce delay and transistor count to 24.

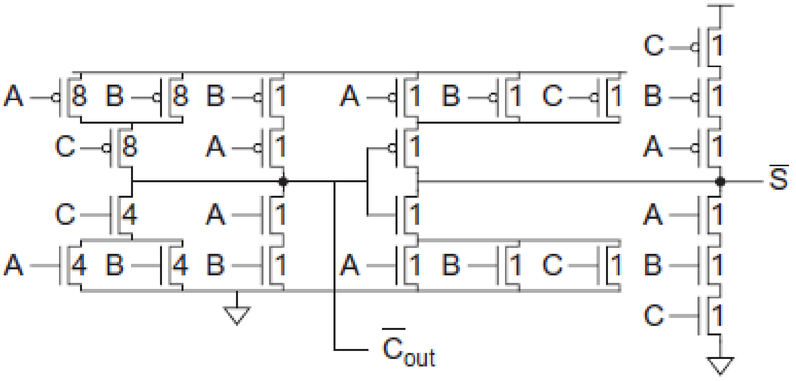


Fig.4.1 (b) Full adder for carry-ripple operation

This delay can be reduced by omitting the inverters on the outputs, as was done in Fig.4.1 (b). Because addition is a self-dual function (i.e., the function of complementary inputs is the complement of the function), an inverting full adder receiving complementary inputs produces true outputs. Fig.4.1 (c) shows a carry ripple adder built from inverting full adders. Every other stage operates on complementary data. The delay inverting the adder inputs or sum outputs is off the critical ripple-carry path.

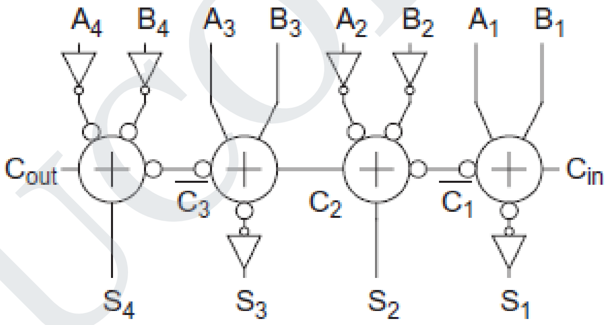


Fig.4.1 (c) 4-bit carry-ripple adder (inverting full adders)

- **RIPPLE CARRY ADDITION USING PG**

The critical path of the carry-ripple adder passes from carry-in to carry-out along the carry chain majority gates. As the P and G signals will have already stabilized by the time the carry arrives, we can use them to simplify the majority function into an AND-OR gate:

$$
\begin{aligned}
C_i &= A_iB_i + \left(A_i + B_i\right)C_{i-1} \\
&= A_iB_i + \left(A_i \oplus B_i\right)C_{i-1} \\
&= G_i + P_iC_{i-1}
\end{aligned}
$$

Because Ci= Gi:0, carry-ripple addition can now be viewed as the extreme case ofgroup

$$G_{i:0} = G_i + P_i \cdot G_{i-1:0}$$

PG logic in which a 1-bit group is combined with an i-bit group to form an (i+1) bit group

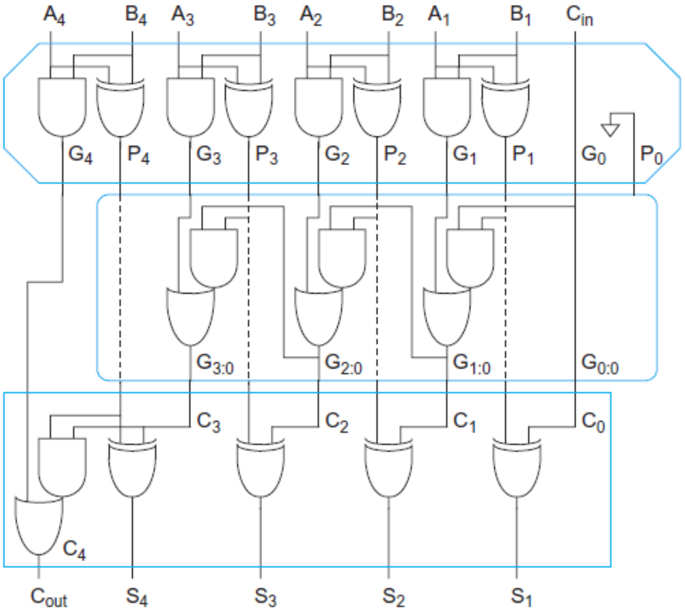


Fig.4.1 (d) 4-bit ripple carry adder

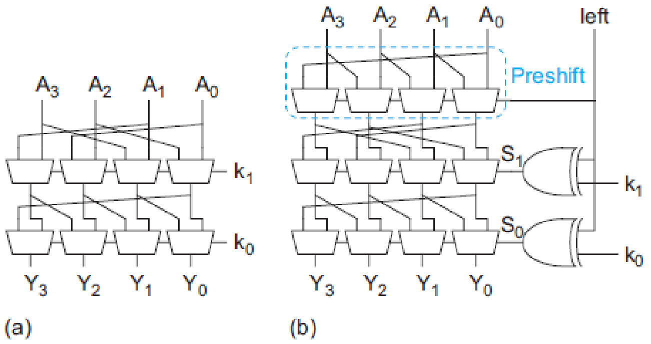
Fig.4.1 (d) shows a 4-bit carry-ripple adder. The critical carry path now proceeds through a chain of AND-OR gates rather than a chain of majority gates.

-------------------------**********-------------------------

2. Discuss the operation of Barrel shifter with neat sketch.

A barrel shifter performs a right rotate operation. It handles left rotations using the complementary shift amount. Barrel shifters can also perform shifts when suitable masking hardware is included. Barrel shifters come in array and logarithmic forms. The logarithmic barrel shifters are most useful because they are better suited for large shifts. Fig.4.2 (a) shows a simple 4-bit barrel shifter that performs right rotations. Unlike funnel shifters, barrel shifters contain long wrap-around wires.

In a large shifter, it is necessary to upsize or buffer the drivers for these wires. Fig.4.2 (b) shows an enhanced version that can rotate left by prerotating right by 1, then rotating right by k. Performing logical or arithmetic shifts on a barrel shifter requires a way to mask out the bits that are rotated off the end of the shifter, as shown in Fig.4.2 (c).



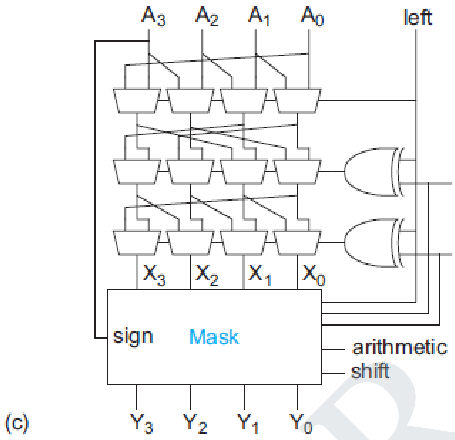


Fig.4.2 Barrel shifters: (a) rotate right, (b) rotate left or right, (c) rotates and shifts

Fig.4.2 (d) shows a 32-bit barrel shifter using a 5:1 multiplexer and an 8:1 multiplexer. The first stage rotates right by 0, 1, 2, 3, or 4 bits to handle a pre-rotate of 1 bit and a fine rotate of up to 3 bits combined into one stage. The second stage rotates right by 0, 4, 8, 12, 16, 20, 24, or 28 bits. The critical path starts with decoding the shift amount for the first stage. If the shift amount is available early, the delay from A to Y improves substantially.

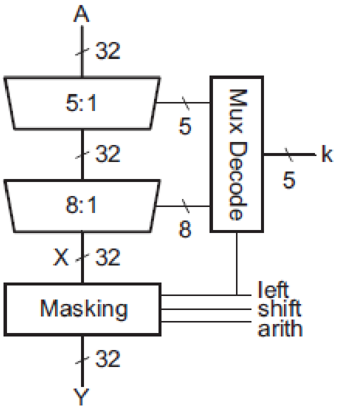
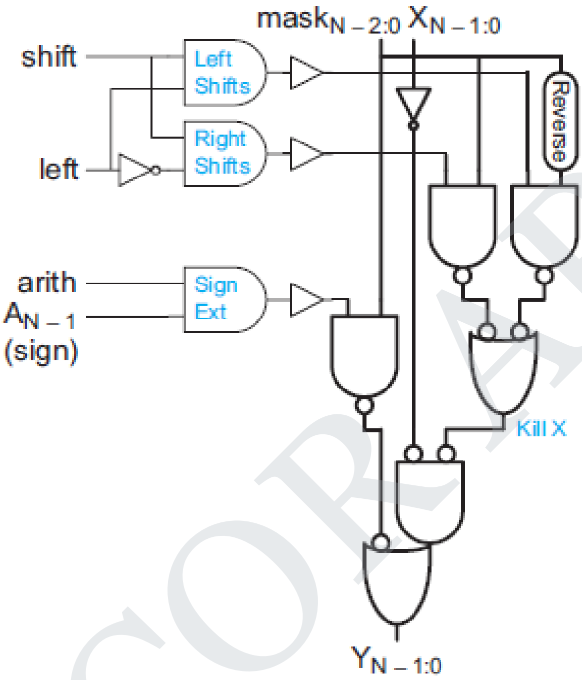


Fig.4.2 (d) 32-bit logarithmic barrel shifter

While the rotation is taking place, the masking unit generates an N-bit mask with ones where the kill value should be inserted for right shifts. For a right shift by m, the m most significant bits are ones. This is called a thermometer code. When the rotation result X is complete, the masking unit replaces the masked bits with the kill value. For left shifts, the mask is reversed.



\

Fig.4.2 (e) Barrel shifter masking logic

Fig.4.2 (e) shows masking logic. If only certain shifts are supported, the unit can be simplified, and if only rotates are supported, the masking unit can be eliminated, saving substantial hardware, power, and delay.

3. Explain in detail about the operation of carry lookahead adder with necessary diagrams. (Nov-2010)

The carry-lookahead adder (CLA) computes group generate signals as well as

$$C_{o,k} = f(A_k, B_k, C_{o,k-1}) = G_k + P_k C_{o,k-1}$$

group propagate signals to avoid waiting for a ripple to determine if the first group generates a carry.

The dependency between $C_{o,k}$ and $C_{o,k-1}$ can be eliminated by expanding $C_{o,k-1}$

$$C_{o,k} = G_k + P_k(G_{k-1} + P_{k-1}C_{o,k-2})$$

In expanded form,

$$C_{o,k} = G_k + P_k(G_{k-1} + P_{k-1}(\ldots + P_1(G_o + P_o C_{i,o}))) \text{ ---- (1)}$$

Here $C_{i,o} = 0$. For every bit, the carry and sum outputs are independent of the previous bits. The ripple effect has thus been effectively eliminated and therefore the addition time should be independent of number of bits. Fig.4.3 (a) shows the carry-lookahead adder.

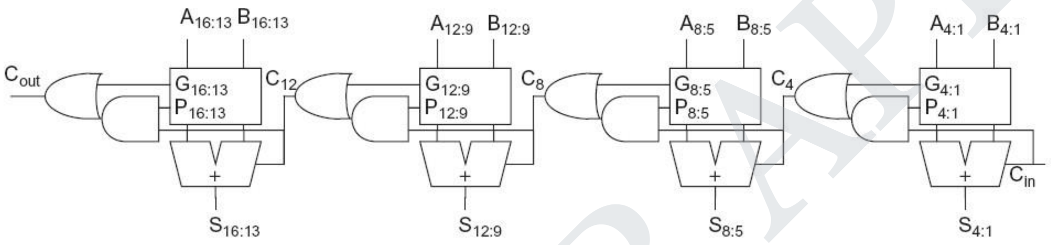


Fig.4.3 (a) Carry-lookahead adder

The possible circuit implementation of equation (1) is shown in Fig.4.3 (b) for N=4. The large fan-in of the circuit makes it slow for larger values of N. Implementing it with simpler gates requires multiple-logic levels. In both cases, the propagation delay increases. Furthermore, the fan-out of some signals tend to grow excessively, slowing down the adder more since the propagation delay of a gate is proportional to its load. Finally the area of implementation grows progressively with N. In general, a CLA using k groups of n bits each has a delay of,

$$t_{\text{cla}} = t_{pg} + t_{pg(n)} + \left[\left(n-1\right)+\left(k-1\right)\right] t_{AO} + t_{\text{xor}}$$

where tpg(n) is the delay of the AND-OR-AND-OR-…-AND-OR gate computing the valency-n generate signal. It requires the extra n-bit generate gate, so the simple CLA is not a good design choice. CLAs often use higher-valency cells to reduce the delay of the n-bit additions by computing the carries in parallel. Fig.4.3 (c) shows such a CLA in which the 4-bit adders are built using Manchester carry chains or multiple static gates operating in parallel.

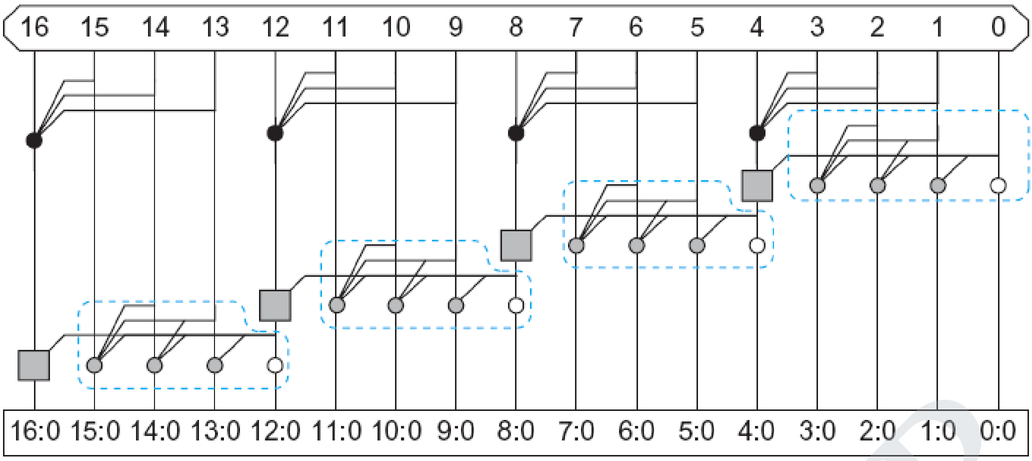


Fig.4.3 (c) Improved CLA group PG network

4.Explain multiplication with an example and discuss the different types of multipliers. (Nov-2010) (MAY/JUN 2016)

Multiplication is less common than addition, but is still essential for microprocessors, digital signal processors, and graphics engines. The most basic form of multiplication consists of forming the product of two unsigned (positive) binary numbers. For example, the multiplication of two positive 6-bitbinary integers, 25_{10} and 39_{10}, proceeds as shown in Fig.4.4 (a).M × N-bit multiplication P = Y × X is performed by forming N partial products of M bits each, and then summing the appropriately shifted partial products to produce an M+ N-bit result P. Binary multiplication is equivalent to a logical AND operation. Therefore, generating partial products consists of the logical ANDing of the appropriate bits of the multiplier and multiplicand. Each column of partial products is added and the carry values are passed to the next column.

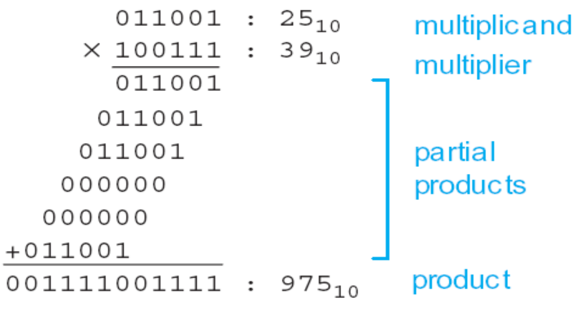
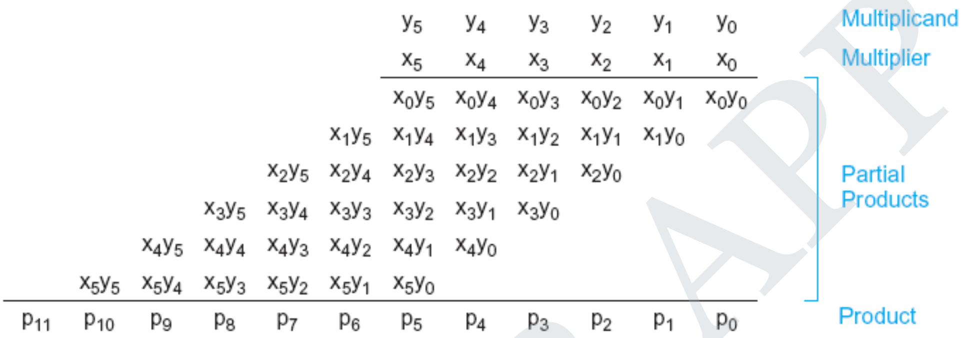


Fig.4.4 (a) Multiplication example

The multiplicand is denoted as Y = {yM———1, yM———2... y1, y0} and the multiplier is denoted as X = {xN———1, xN———2… x1, x0}. The product is given in equation (1). Fig.4.4 (b) illustrates the generation, shifting, and summing of partial products in a 6 × 6-bit multiplier. This set of operations can be mapped directly into hardware and the resulting structure is called an array multiplier.

$$P = \left(\sum_{j=0}^{M-1} y_j 2^j\right)\left(\sum_{i=0}^{N-1} x_i 2^i\right) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} x_i y_j 2^{i+j} \qquad \text{----- (1)}$$



						y_5	y_4	y_3	y_2	y_1	y_0	Multiplicand
						x_5	x_4	x_3	x_2	x_1	x_0	Multiplier
						x_0y_5	x_0y_4	x_0y_3	x_0y_2	x_0y_1	x_0y_0	Partial Products
					x_1y_5	x_1y_4	x_1y_3	x_1y_2	x_1y_1	x_1y_0		
				x_2y_5	x_2y_4	x_2y_3	x_2y_2	x_2y_1	x_2y_0			
			x_3y_5	x_3y_4	x_3y_3	x_3y_2	x_3y_1	x_3y_0				
		x_4y_5	x_4y_4	x_4y_3	x_4y_2	x_4y_1	x_4y_0					
	x_5y_5	x_5y_4	x_5y_3	x_5y_2	x_5y_1	x_5y_0						
p_{11}	p_{10}	p_9	p_8	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0	Product

Fig.4.4 (b) Partial products

The array Multiplier

The composition of an array multiplier is shown in Fig.4.4 (c). There is a one-to-one correspondence between this hardware structure and the manual multiplication in Fig.4.4 (a). The generation of partial product requires a multiplication by 1 or 0 (i.e.) AND operation. Generating the N partial products requires N M-bit AND gates. The shifting of the partial products is performed by simple routing and does not require any active logic. The overall structure can be compacted into a rectangle, resulting in a very efficient layout.

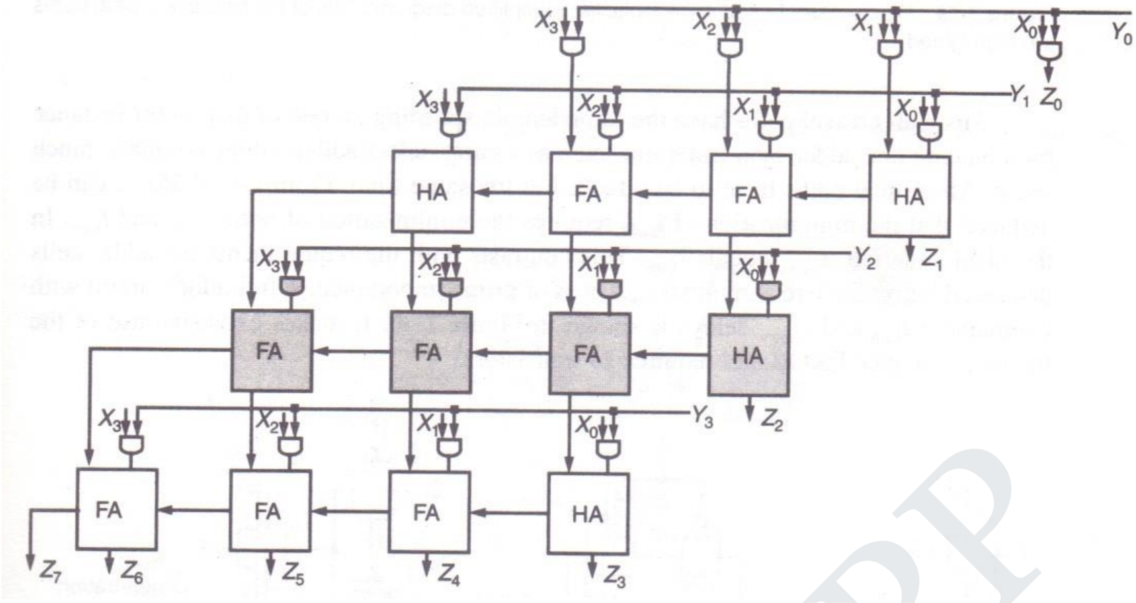


Fig.4.4 (c) 4x4 Multiplier for unsigned numbers

Due to array organization, determining the propagation delay is difficult. Partial sum adders are implemented as ripple-carry adders. Performance optimization requires critical timing path to be identified. Two such paths are highlighted in Fig.4.4 (d). The propagation delay is given as,

$$t_{mult} \approx [(M-1) + (N-2)]t_{carry} + (N-1)t_{sum} + t_{and} \quad \text{----- (2)}$$

where t_{carry} is the propagation delay between input and output carry, t_{sum} is the delay between the input carry and sum bit of the full adder and t_{and} is the delay of the AND gate.

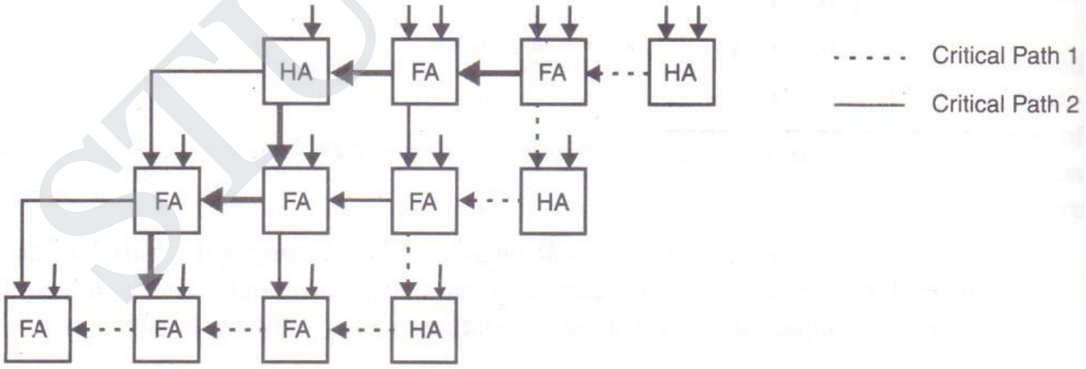


Fig.4.4 (d) Ripple carry based 4x4 multiplier with two critical paths highlighted

Since all critical paths have the same length, speeding up one of them does not make much difference. All the critical paths have to be speeded up at the same time. From equation (2), it is deduced that the minimization of t_{mult} requires the minimization of both t_{carry} and t_{sum}.

Due to large number of identical critical paths, increasing the performance of the structure shown in Fig.4.4 (d) is achieved with careful transistor sizing. A more efficient multiplier structure is obtained by noticing that the multiplication result does not change when the output carry bits are passed diagonally downwards instead of to the right. An extra adder called as a vector-merging adder, is added to generate the final result. Such multiplier is called as **carry-save multiplier,** because the carry bits are not immediately added but are rather saved for the next adder stage. This structure has a slightly increased area cost but it has the advantage that its worst-case-critical path is uniquely defined as shown in fig.4.4 (e) and expressed in equation (3).

$$t_{mult} = (N-1)t_{carry} + t_{and} + t_{merge} \text{ -------- (3)}$$

assuming that $t_{and} = t_{carry}$

A simple way to reduce the propagation delay of this structure is to minimize t_{merge}. This is achieved by using a fast adder implementation such as a carry-select or a lookahead structure for the merging folder.

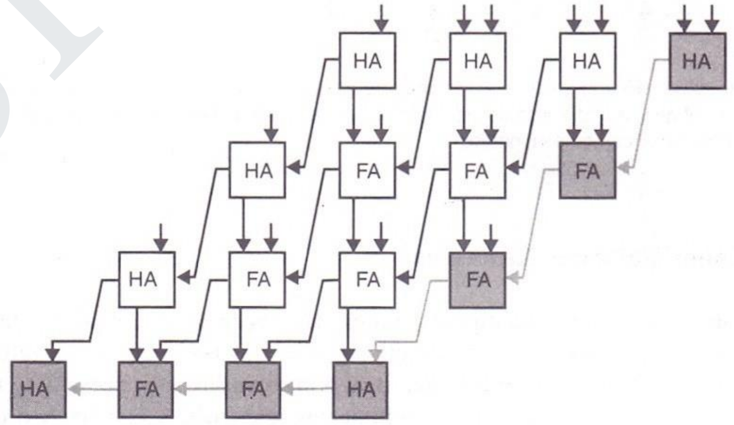


Fig.4.4 (e) 4x4 carry-save multiplier

Booth Encoder:

The speed of the multiplication can be increased by using a special encoding called booth encoding of the multiplier word that reduces the number of required addition stages. Instead of traditional binary encoding the multiplier word is recoded into radix-4 scheme.

$$Y = \sum_{j=0}^{(N-1)/2} Y_j 4^j \text{ with } (Y_j \in \{-2,-1,0,1,2\}$$

The radix-4 multiplier produces N/2 partial products. Each partial product is 0, Y, 2Y, or 3Y, depending on a pair of bits of X. Computing 2Y is a simple shift, but 3Y is a hard multiple requiring a slow carry propagate addition of Y + 2Y before partial product generation begins. The advantage of the recoding is that the number of partial products and hence the number of additions is halved, which results in a speed-up as well as area reduction. The only expense is somewhat more involved multiplier cell. While multiplication with {0,1} is equivalent to an AND operation, multiplying with {-2,-1,0,1,2} requires a combination of inversion and shift logic.

Here 3Y = 4Y – Y and 2Y = 4Y – 2Y. However, 4Y in a radix-4 multiplier array is equivalent to Yin the next row of the array that carries four times the weight. Hence, partial products are chosen by considering a pair of bits along with the most significant bit from the previous pair. If the most significant bit from the previous pair is true, Y must be added to the current partial product. If the most significant bit of the current pair is true, the current partial product is selected to be negative and the next partial product is incremented. Table 1 shows how the partial products are selected based on bits of the multiplier. Negative partial products are generated by taking the two's complement of the multiplicand (possibly left-shifted by one column for –2Y). An unsigned radix-4 Booth encoded multiplier requires partial products rather than N. Each partial product is M+ 1 bits to accommodate the 2Y and –2Y multiples. Even though X and Y are unsigned, the partial products can be negative and must be sign extended properly.

Inputs			Partial Product	Booth Selects		
x_{2i+1}	x_{2i}	x_{2i-1}	PP_i	SINGLE_i	DOUBLE_i	NEG_i
0	0	0	0	0	0	0
0	0	1	Y	1	0	0
0	1	0	Y	1	0	0
0	1	1	$2Y$	0	1	0
1	0	0	$-2Y$	0	1	1
1	0	1	$-Y$	1	0	1
1	1	0	$-Y$	1	0	1
1	1	1	$-0\ (=0)$	0	0	1

Table 1: Radix-4 modified Booth encoding values

In a radix-4 Booth-encoded multiplier design, each group of 3 bits (a pair, along with the most significant bit of the previous pair) is encoded into several select lines(SINGLEi, DOUBLEi, and NEGi, given in the rightmost columns of Table 1) and driven across the partial product row as shown in Fig.4.4 (f).

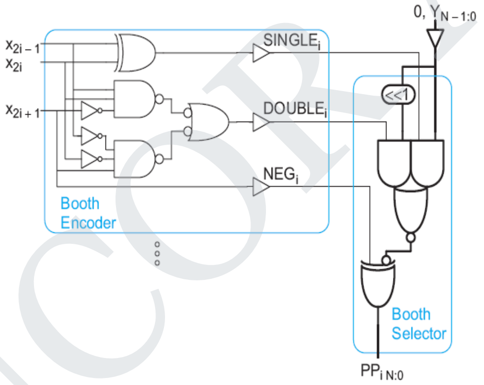


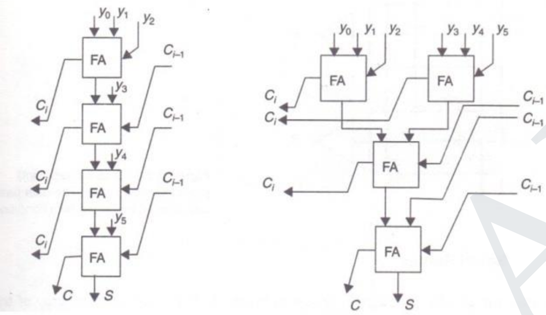
Fig.4.4 (f) Radix-4 Booth encoder and selector

The multiplier Y is distributed to all the rows. The select lines control Booth selectors that choose the appropriate multiple of Y for each partial product. The Booth selectors substitute for the AND gates of a simple array multiplier to determine the ith partial product. Fig.4.4 (f) shows a conventional Booth encoder and selector design. Y is zero-extended to M + 1 bit. Depending on SINGLEi and DOUBLEi, the gate selects either 0, Y, or 2Y. Negative partial products should be two's-complemented (i.e., invert and add 1). If NEGi is asserted, the partial product is inverted. The extra 1 can be added in the least significant column of the next row to avoid needing a CPA.

Wallace-Tree Multiplier

The partial-sum adders can also be re-arranged in a tree-like fashion. In Fig.4.4 (g) vertical slice is extracted from a generic carry-save multiplier and hence the data ripples from top to bottom similar to what happens in ripple-carry adder. The number of stages equals the number of bits in the multiplier word minus 2. Now the linear chain is translated into a tree structure as shown in Fig.4.4 (h). This topology which has an $O(\log_2 N)$ multiplication time, is called the

Wallace multiplier. It is faster than the carry-save structure but has the disadvantage of being irregular. This complicates the task of coming up with a dense and efficient layout. Wallace multipliers are used only in designs where performance is critical and design time is only a secondary consideration.



(g)Vertical slice of 6-bit carry-save multiplier (h) Wallace tree organization

Fig.4.4 Wallace-tree multiplier

5.What are the two types of dividers? Explain them with example and schematic sketches.

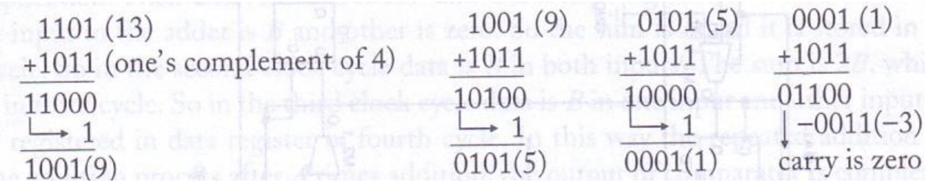
A binary divider can be categorized into two types,

- Serial divider
- Parallel divider

Serial divider

Serial division is done means of repeated subtraction. For eg, dividing 19 by 3 requires, subtracting 3 six times from 19 so that the remainder is one. So the quotient is six and remainder is one. Similarly binary division is performed.

Binary division of binary number 1101 – (decimal 13), by binary number 0100 – (decimal 4) by one's complement method of subtraction is shown:



```
  1101 (13)                             1001 (9)    0101 (5)    0001 (1)
 +1011 (one's complement of 4)         +1011       +1011       +1011
 11000                                 10100       10000       01100
 └─→ 1                                 └─→ 1       └─→ 1       │ −0011(−3)
 1001(9)                               0101(5)     0001(1)     ↓ carry is zero
```

Here when the difference is positive final carry is 1 which is end around and added to get the actual difference. When difference is negative, carry is zero and true result is obtained by one's complement of the sum output. So, repeated subtraction is done till final carry is one. Since subtraction is for three times, when the carry is 1, the quotient is 3 and remainder is the final difference which is 0001. The implementation of binary divider by means of repeated subtraction of two 4-bit unsigned binary numbers is shown in fig.4.5 (a). Here the divisor $Y_3Y_2Y_1Y_0$ is subtracted from $X_3X_2X_1X_0$ by one's complement method of subtraction. The basic building blocks used are,

- Adder ADD4 to add to 4-bit binary number
- 4-bit binary up counter CB4CE
- 4 set of 2:1 MUX and D FLIPFLOPS

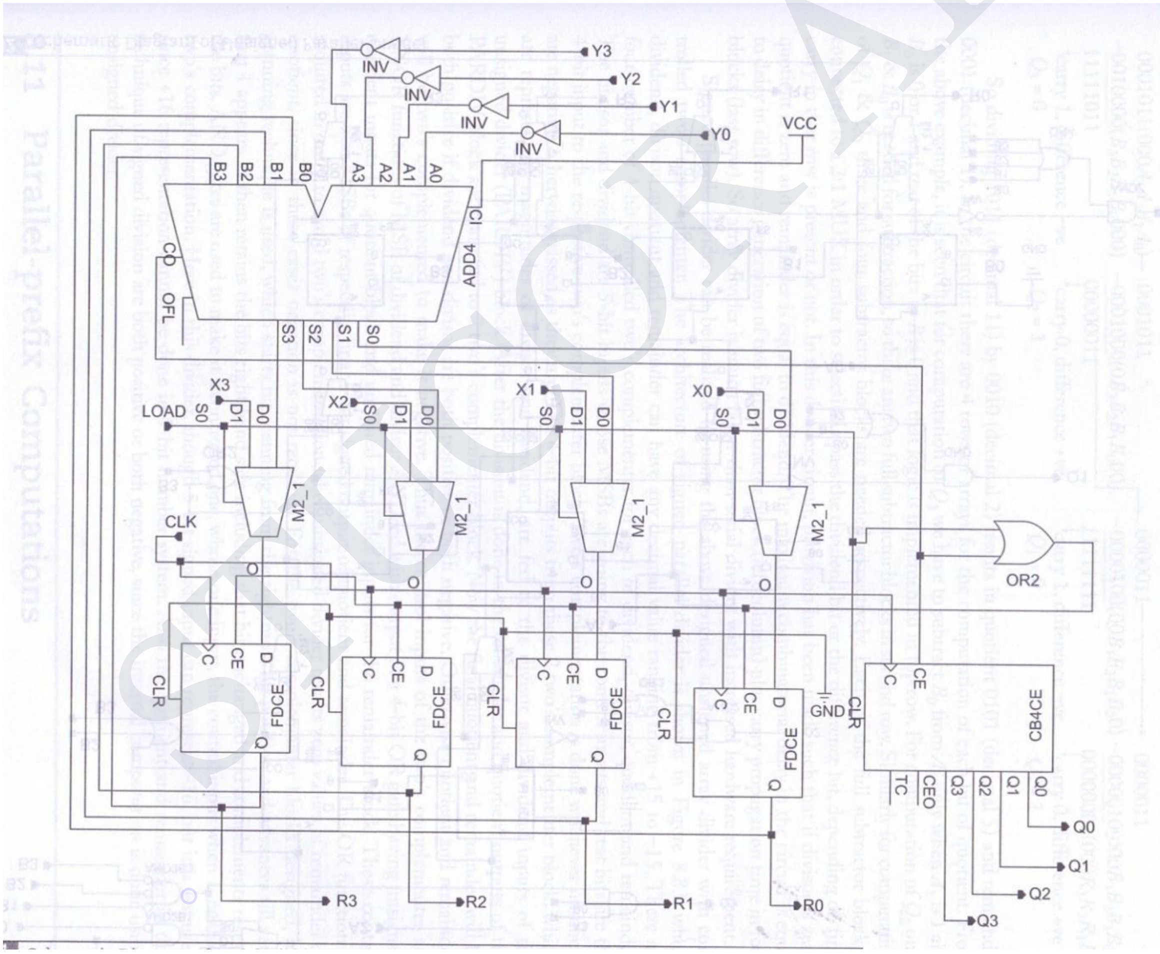
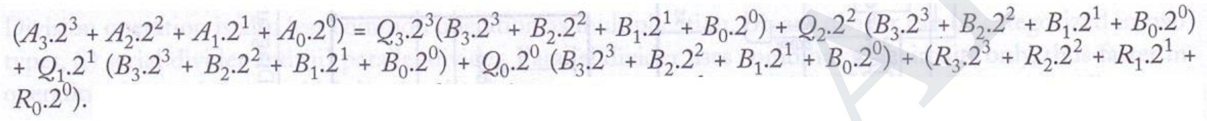


fig.4.5 (a) Schematic diagram of serial divider

Each bit of divisor is complemented and fed to one set of adder inputs. Dividend is initially loaded in a register comprising of 4 D Flipflops by putting LOAD input high, which is common select input of all the MUX and also to the CLR input of the counter. So initially the counter is also reset to zero. Output of D Flipflop is fed to another set of inputs of the adder. The final carry output of the adder block is fed to the clock enable input CE of the counter and also to an OR gate whose other input is LOAD and output goes to clock enable of register.

Parallel divider

Parallel divider performs parallel division using array of full subtractor blocks. Implementation of a parallel divider, which is also called as array divider, to divide an unsigned 4-bit number A3A2A1A0 by B3B2B1B0 is shown in fig.4.5 (b). The relation between dividend A3A2A1A0, divisor B3B2B1B0, quotient Q3Q2Q1Q0 and remainder R3R2R1R0 is as follows:



$$(A_3.2^3 + A_2.2^2 + A_1.2^1 + A_0.2^0) = Q_3.2^3(B_3.2^3 + B_2.2^2 + B_1.2^1 + B_0.2^0) + Q_2.2^2\,(B_3.2^3 + B_2.2^2 + B_1.2^1 + B_0.2^0) + Q_1.2^1\,(B_3.2^3 + B_2.2^2 + B_1.2^1 + B_0.2^0) + Q_0.2^0\,(B_3.2^3 + B_2.2^2 + B_1.2^1 + B_0.2^0) + (R_3.2^3 + R_2.2^2 + R_1.2^1 + R_0.2^0).$$

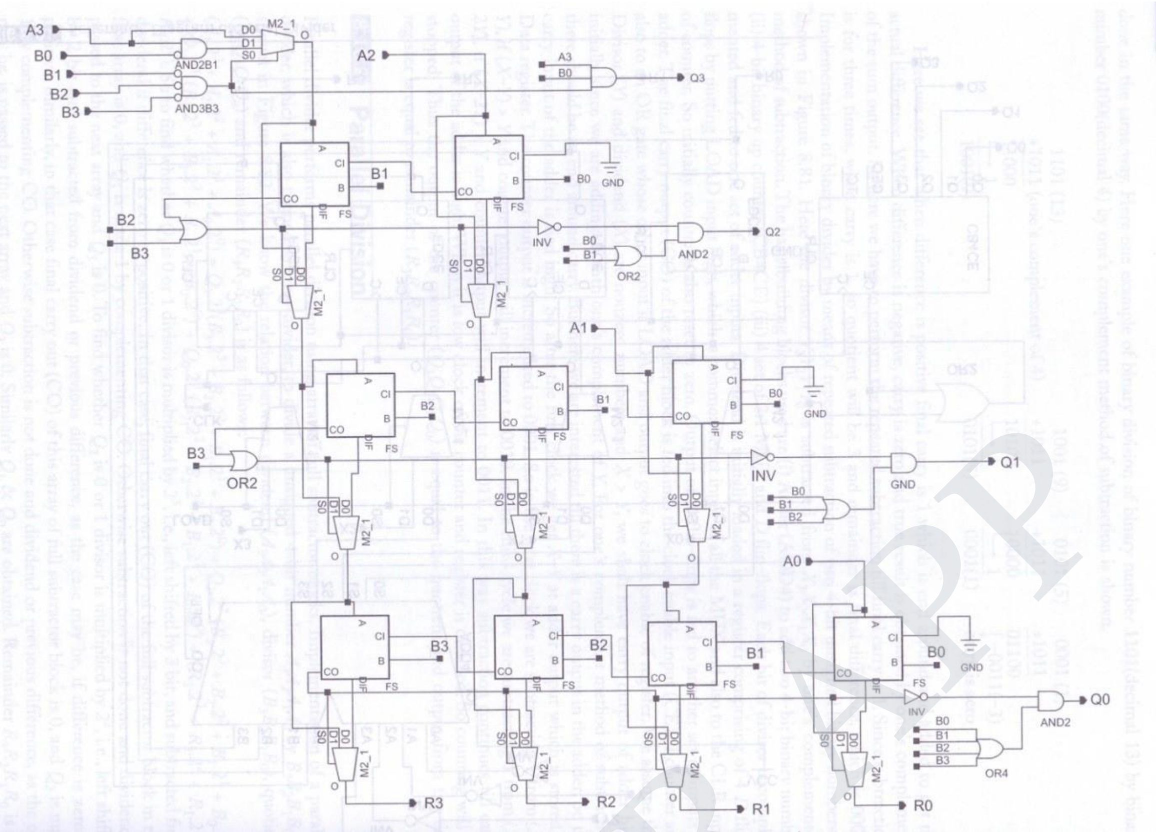


Fig.4.5 (b) Schematic diagram of unsigned parallel divider

To find whether Q3 is 0 or 1 divisor is multiplied by 2^3, i.e., left shifted by 3 bit and subtracted from dividend if difference is zero or positive. The final carryout (CO) of the full subtractor block in the first array is 0, and Q3 is made 1 by complementing CO. Otherwise subtraction is not done and dividend is passed to the next array and Q3 is zero. To find whether Q2 is 0 or 1 divisor

is multiplied by 2^2, i.e., left shifted by 2 bit and subtracted from dividend or previous difference,

if the difference is zero or positive. If final carry out (CO) of this array of full subtractor block is 0, and Q2 is made 1 by complementing CO. Similarly Q1 and Q0 are obtained.

The remainder R3R2R1R0is the last positive difference output. Example of binary division of 1011 (A3A2A1A0) by 0010 (B3B2B1B0) is shown below:

$0001011(000A_3A_2A_1A_0) \rightarrow$	0001011	$0000011 \rightarrow$	0000011
$-0010000(B_3B_2B_1B_0000)$	$-0001000(0B_3B_2B_1B_000)$	$-0000100(00B_3B_2B_1B_00)$	$-0000010(000B_3B_2B_1B_0)$
11111011	00000011	11111111	$00000001(000R_3R_2R_1R_0)$
carry 1, difference −ve	carry 0, difference +ve	carry 1, difference −ve	carry 0, difference +ve
$Q_3 = 0$	$Q_2 = 1$	$Q_1 = 0$	$Q_0 = 1$

So dividing 1011 (decimal 11) by 0010 (decimal 2) results in quotient 0101 (decimal 5) and remainder 0001 (decimal 1). In the circuit, there are four rows of arrays for the computation of each bit of quotient. For the above example, for the computation of Q3 we have to subtract B0 from A3 only when A3 is 1 and B0 is 0 or 1 and rest of the bits in B is 0 and that logic is implemented in top row. For the computation of Q2 only B1 and B0 is needed for subtraction, so there are two full subtractor blocks in second row. Foe computation of Q1 and Q0 three and four subtractor blocks are needed respectively. Each of the full subtractor block is connected to a 2:1 MUX in order to selectively pass the dividend bit or the difference bit, depending on final carry is present in that row or not. If the divisor is zero, quotient is zero and remainder is equal to dividend. The maximum combinational delay is equal to delay in difference generation of two full subtractor blocks plus carry propagation time in four blocks. So array divider is much faster than serial divider with increased hardware requirement.

------------------------**********------------------------

6.Design a 16 bit carry by pass and carry select addersand discuss their Features. (MAYJUN 2016)

<u>HIGH SPEED ADDERS,</u>

1. Carry skip adders
2. Carry select adders
3. Carry save adders

1 Carry skip adders: A carry-skip adder consists of a simple ripple carry-adder with a special speed up carry chain called a **skip chain**. This chain defines the distribution of ripple carry blocks, which compose the skip adder. The addition of two binary digits at stage i, where i is not equal to 0, of the ripple carry adder depends on the carry in, C_i , which in reality is the carry out, C_{i-1}, of the previous stage. Therefore, in order to calculate the sum and the carry out, C_{i+1} , of stage i, it is imperative that the carry in, C_i, be known in advance. It is interesting to note that in some cases C_{i+1} can be calculated without knowledge of C_i.

Boolean Equations of a Full Adder:

$P_i = A_i \oplus B_i$ Equ. 1 --carry propagate of ith stage

$S_i = P_i \oplus C_i$ Equ. 2 --sum of ith stage

$\mathbf{C_{i+1} = A_iB_i + P_iC_i}$ Equ. 3 --carry out of ith stage

Supposing that A_i = B_i, then P_i in equation 1 would become zero (equation 4). This would make C_{i+1} to depend only on the inputs A_i and B_i, without needing to know the value of C_i.

$\mathbf{A_i = B_i\ ;\ P_i = 0}$ Equ. 4 --from #Equation 1

If $\mathbf{A_i = B_i = 0\ ;\ C_{i+1} = A_iB_i = 0}$ --from equation 3

If $\mathbf{A_i = B_i = 1;\ C_{i+1} = A_iB_i = 1}$ --from equation 3

Therefore, if Equation 4 is true then the carry out, C_{i+1}, will be one if A_i = B_i = **1** or zero if A_i = B_i = **0**. Hence the output can be computed with the carry out at any stage of the addition provided equation 4 holds. These would enable to build an adder whose average time of computation would be proportional to the longest chains of zeros and of different digits of A and B.

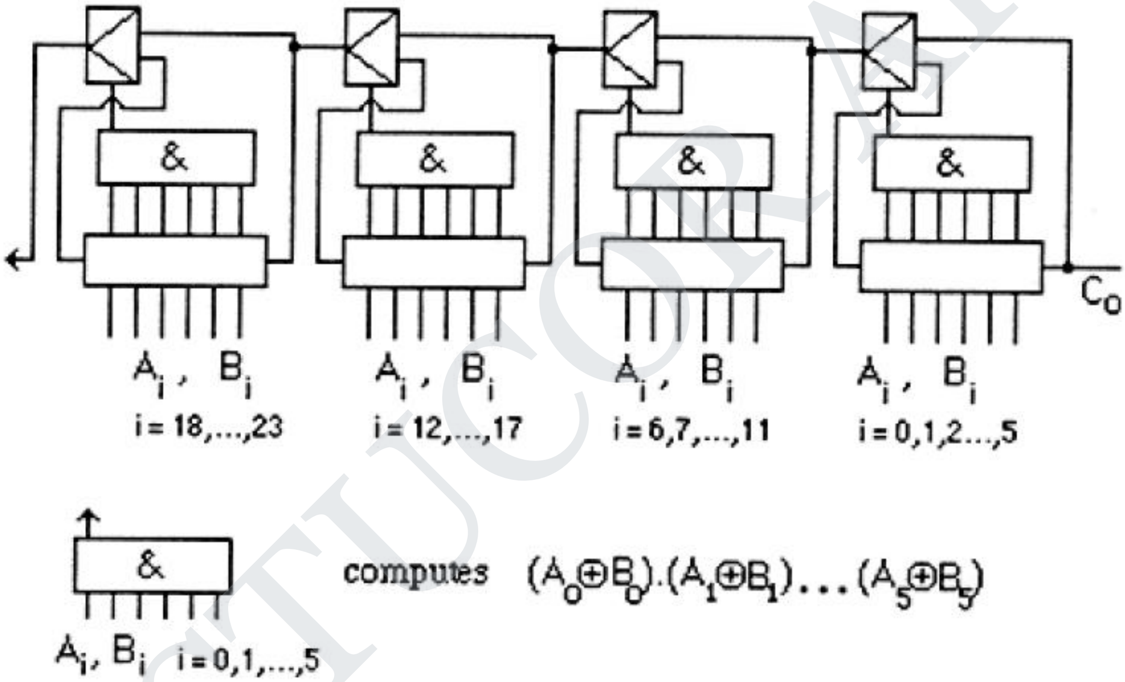


Fig: Carry skip Chain

2. Carry select adders:

The concept of the carry-select adder is to compute alternative results in parallel and subsequently selecting the correct result with single or multiple stage hierarchical techniques . In order to enhance its speed performance, the carry-select adder increases its area requirements. In carry-select adders both sum and carry bits are calculated for the two alternatives: input carry "0" and "1". Once the carry-in is delivered, the correct computation is chosen (using a MUX) to produce the desired output. Therefore instead of waiting for the carry-in to calculate the sum, the

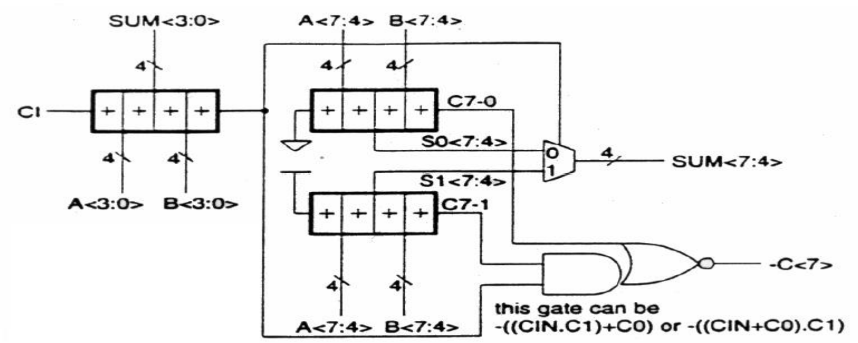


Fig: Concept of carry select adder

sum is correctly output as soon as the carry-in gets there. The time taken to compute the sum is then avoided which results in a good improvement in speed.

Carry-select adders can be divided into equal or unequal sections. For each section, the calculation of two sums is accomplished using two 4-bit ripple-carry adders. One of these adders is fed with a 0 as carry-in whereas the other is fed a 1. Then using a multiplexer, depending on the real carryout of the previous section, the correct sum is chosen. Similarly, the carryout of the section is computed twice and chosen depending of the carryout of the previous section. The concept can be expanded to any length for example a 16-bits carry-select adder can be composed of four sections. Each of these sections is composed of two 4-bits ripple-carry adders. This is referred as linear expansion.

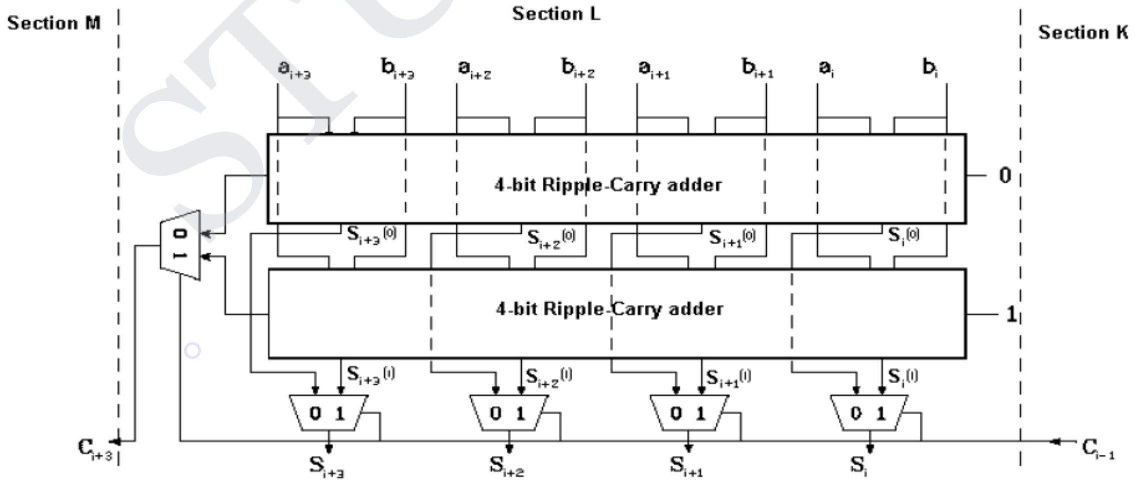


Fig: one section of a large carry select adder

3 Carry save adders:

In most computations, we need to add several operands together, carry save adders are ideal for this type of addition. A carry save adder consists of a ladder of stand alone full adders, and carries out a number of partial additions. The principal idea is that the carry has a higher power of 2 and thus is routed to the next column. Doing additions with Carry save adder saves time and logic.

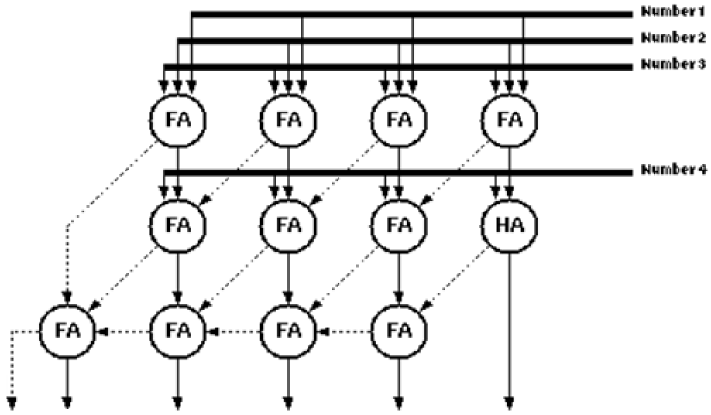


Fig: Carry save adder for 4 bit number

In this method, for the first 3 numbers a row of full adders are used. Then a row of full adders is added for each additional number. The final results, in the form of two numbers SUM and CARRY, are then summed up with a carry propagate adder or any other adder

UNIT V

IMPLEMENTATION STRATEGIES

PART A

1. **What is an interconnect?**

   The last half dozen or so layers define metal wires between the transistors are called interconnect.

2. **Define Manufacturing lead time**

   It is defined as the time it takes to make am IC not including the design time.

3. **Define Flexible blocks.**

   The predefined logic cells are known as standard cells. The standard cell areas are called flexible blocks.

4. **Define Mega cells**

   The flexible blocks used in combination with larger predesigned cells, like micro controllers and micro processors, these are called mega cells.

5. **List the advantages of CBIC(Nov-2009)**

   - Less cost
   - Less time
   - Reduced Risk
   - Transistor operates at maximum speed.

6. **What are primitive cells?**

   The predefined pattern of transistors on a gate array is the base array. The base array is made up of a smallest element called primitive cell.

7. **Define Customer owned tooling.**

   If an ASIC design is completed using cell library we own the mask that are used to manufacture the ASIC. This is called Customer owned tooling.

8. Write the features of Xilinx LCA.(April 2008)

1. Vertical lines and horizontal lines run between CLB's
2. Long lines run across the entire chip to form internal buses
3. Direction connection bypasses the switch matrices and directly connects adjacent CLB
4. General purpose interconnect joins switch boxes or magic boxes or switching matrices.

9. Write the advantages of altera max 5000 and 7000?

1. It uses a fixed no. of connections.
2. Fixed routing delay
3. Simple and improved speed in placement and routing software

10. Write about FPGA routing techniques.

✓ Comprises of programmable switches and wires

✓ Provides connection between I/O blocks, Logic blocks, etc.

✓ Routing decides logic block density and area consumed. Different routing techniques are

- Xilinx routing architecture
- Actel routing methodology
- Altera routing methodology

11. Give the different types of ASIC.

1. Full custom ASICs
2. Semi-custom ASICs

   * Standard cell based ASICs

   * Gate-array based ASICs
3. Programmable ASICs

* Programmable Logic Device (PLD)

* Field Programmable Gate Array (FPGA).

12. What is the full custom ASIC design? (May 2008,May 2009)

In a full custom ASIC, an engineer designs some or all of the logic cells, circuits or layout specifically for one ASIC. It makes sense to take this approach only if there are no suitable existing cell libraries available that can be used for the entire design.

13. What is the standard cell-based ASIC design? (May 2008)

A cell-based ASIC (CBIC) uses predesigned logic cells known as standard cells. The standard cell areas also called fle4xible blocks in a CBIC are built of rows of standard cells. The ASIC designer defines only the placement of standard cells and the interconnect in a CBIC. All the mask layers of a CBIC are customized and are unique to a particular customer.

14. Differentiate between channeled & channel less gate array.

Channeled Gate Array	**Channel less Gate Array**
1. Only the interconnect is customized layers	Only the top few mask
customized. 2. The interconnect uses predefined spaces between rows	No predefined areas are set aside For routing between
of base cells. 3. Routing is done using the spaces transit unused.	cells. Routing is done using the area of
4. Logic density is less	Logic density is higher.

15. What is a FPGA?

A field programmable gate array (FPGA) is a programmable logic device that supports implementation of relatively large logic circuits. FPGAs can be used to implement a logic circuit with more than 20,000 gates whereas a CPLD can implement circuits of upto about 20,000 equivalent gates.

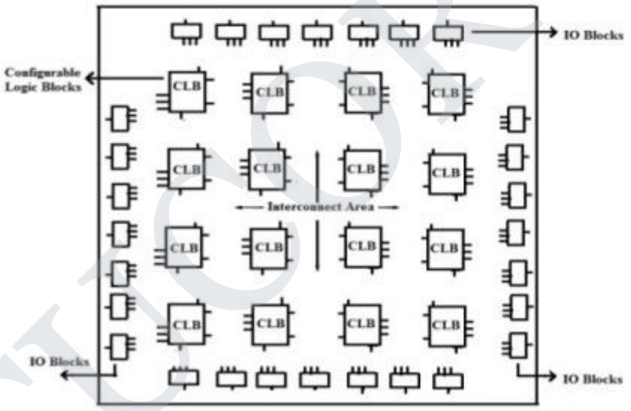
16. What are the types of programmable device?

- Programmable logic structure
- Interconnect
- Reprogrammable gate array

17. What are the essential characteristics of an FPGA?

- None of the mask layers are customized.
- A method for programming the basic logic cells and the interconnect.
- The core is a regular array of programmable basic logic cells that can implement combinational as well as sequential logic (flip-flops).
- A matrix of programmable interconnect surrounds the basic logic cells.

18. Draw the basic building block of FPGA?



19. State the features of full custom ASIC Design. (MAY/JUN 2016)

Full Custom ASIC:

- Full custom includes all possible logic cells and mask layers that are customized.
- These are very expensive to manufacture and design.
- Example is microprocessor.
- In full custom ASIC an engineer design some or all logic cells ,circuits, or layout specifically for one ASIC.

20.What are feed through cells? State their uses.(MAY/JUN 2016)

A feedthrough is a conductor used to carry a signal through an enclosure or printed circuit board. Like any conductor, it has a small amount of capacitance. A "feedthrough capacitor" has a guaranteed minimum value of shunt capacitance built in it and is used for bypass purposes in ultra-high-frequency applications.

PART B

1. Explain about the classification of ASIC. (Nov 2007, Nov 2008, May 2008, May 2009, May 2010) (MAY/JUN2016)

An ASIC is classified into

1. Full custom ASIC
2. Semi custom ASIC

Full Custom ASIC:

- Full custom includes all possible logic cells and mask layers that are customized.
- These are very expensive to manufacture and design.
- Example is microprocessor.
- In full custom ASIC an engineer design some or all logic cells ,circuits, or layout specifically for one ASIC.

Semi Custom ASIC:

- In semicustom asic all the logic cells are predesigned and some of the mask layers are customized. The types of semicustom ASIC are
  1. Standard cell based ASIC
  2. Gate array based ASIC

1. Standard cell based ASIC:-

- A cell based ASIC or cell based IC (CBIC) uses predesigned logic cells like AND gates, OR gates, multiplexers, Flipflops.
- The predefined logic cells are known as standard cells. The standard cell areas are called flexible blocks
- The flexible blocks used in combination with larger predesigned cells, like micro controllers and micro processors, these are called mega cells.

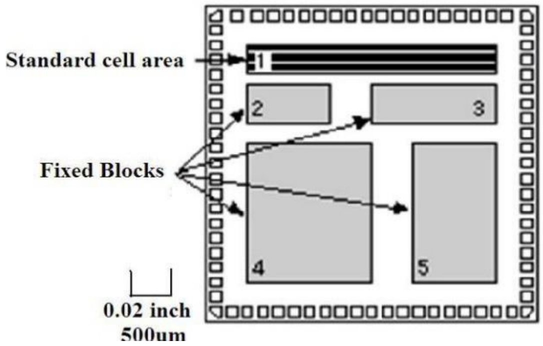


Fig: Cell based ASIC

Advantages:

- Less cost
- Less time
- Reduced Risk
- Transistor operates at maximum speed.

Disadvantages:

- Expense of designing standard cell library is high
- Time needed to fabricate all layers for each new design is high.

2. Gate array based ASIC

- Gate array (GA) based ASIC has predefined transistors on the silicon wafer. The predefined pattern of transistors on a gate array is the base array. The base array is made up of a smallest element called primitive cell.
- To distinguish this type of gate array from other types of gate array ,this is often called MASKED GATE ARRAY.(MGA)
- MACROS: the logic cells in a gate array library are called macro.
- The types of MGA or gate array based ASIC are

1. Channeled gate array
2. Channel less gate array
3. Structured gate array

A. Channeled Gate Array:

- Channeled gate array has space between the rows of transistor for wiring.
- Features:
  1. Only the interconnect is customized.
  2. Interconnect uses predefined spaces between rows of base cells.
  3. Manufacturing lead time is between two days and two weeks.

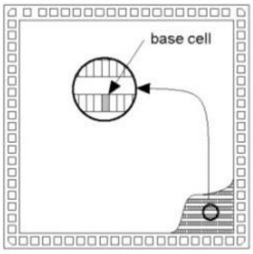


Fig: Channeled Gate Array

B. Channel less Gate Array

- It is also known as channel free gate array.
- The routing on a channel less gate array uses rows of unused transistors.
- Features:
  1. Top few mask layers are customized interconnect.
  2. Manufacturing lead time is between two days and two weeks.

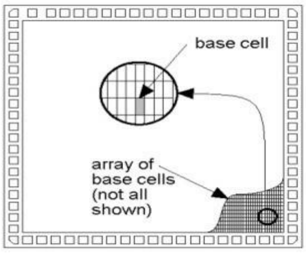


Fig: Channel less Gate Array

3. Structured Gate Array:

- It can be either channeled or channel less, but it includes custom block.
- It is also known as master slice or master image.

- This embedded area either contains a different base cell that is more suitable for building memory cells.

Features:

1. only the interconnect is customized
2. Custom blocks can be embedded.
3. Manufacturing lead time is between two days and two weeks.

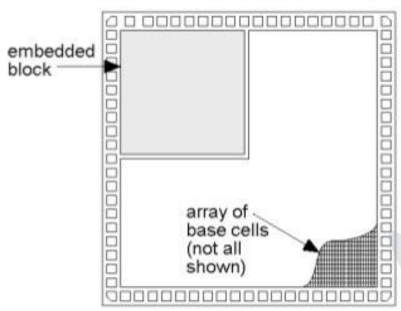


Fig: Structured Gate Array

Advantages:

1. Improved area efficiency
2. Increased performance
3. Lower cost
4. Faster turn around

Disadvantage:

Embedded function is fixed.

Programmable ASIC:

- In which all the all the logic cells are predesigned and none of the mask layers are customized.
- The two types are
  1. Programmable logic device
  2. Field programmable gate array

Programmable logic device: (PLD)

- ➢ Programmable logic devices are standard IC and available in standard configuration.PLD may be configured or programmed.

  Features:

  1. No customized mask layers or logic cells.
  2. Fast design turnaround
  3. Single large block of programmable interconnect
  4. Matrix of large macro cells

Field programmable gate array: (FPGA)

- ➢ Complex PLD's are called FPGA.
- ➢ FPGA are growing rapidly and replace TTL in microelectronic system

Characteristics:

1. No mask layers are customized.
2. Programming basic logic cells and interconnects.
3. Core with regular array of programmable basic logic cells that implement combinational and sequential logic.
4. Matrix of programmable interconnect surrounds the basic logic cells.
5. Programmable I/O cells surround the core.
6. Design turnaround is few hours.

2. a. Explain about ASIC Design Flow

Steps of logic design:

Step1: Design entry

Enter the design into an ASIC design systems ,either using a HDL orschematic entry

Step 2.Logic synthesis

Use VHDL or verilog and a logic synthesis tool to produce a netlist.

Step 3: System portioning

Divide a large system into an ASIC sized pieces.

Step 4: Pre-layout simulation

check whether design function are correct.

Steps of physical design:

Step 5: Floor planning

Arrange the blocks of the netlist on the chip.

Step 6: Placement:

Decide the locations of cells in a block.

Step 7: Routing:

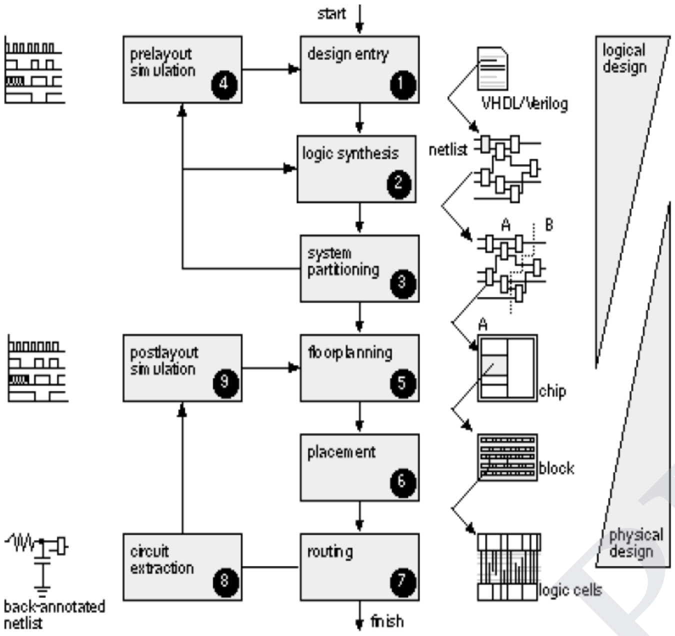
Make the connections between cells and blocks.

Step 8.Extraction:

Determine the resistance and capacitance of the interconnect.

Step 9: Post layout simulation:

Check to see design still works with the added loads.



-------------------------*********--------------------------

2. b. Explain about ASIC cell library in detail.

- Cell library is very important in ASIC design.

- For MGA and CBIC there are three choices to have the cell library.

  i. ASIC manufacturer will supply a cell library.

  ii. Cell library is bought from a third party library vendor.

  iii. Build our own library.

Customer owned tooling:

- If an ASIC design is completed using cell library we own the mask that is used to manufacture the ASIC. This is called Customer owned tooling. Each cell in an ASIC cell library contain the following,

  1. Physical layout

  2. Behavioral model

  3. verilog/VHDL model

4. Timing model
5. Test strategy
6. Circuit schematic
7. Cell icon
8. Wire load model
9. Routing model

3. Explain in detail about FPGA Interconnecting Procedure.(MAY/JUN 2016)

FPGA has different types of programmable interconnect. the structure and complexity of interconnect is determined by programming technology and architecture of basic logic cell. The raw material used to build interconnect is aluminum based metallization with sheet resistance .programmable ASIC comes with two layers, three layers or more layers of metal interconnect.

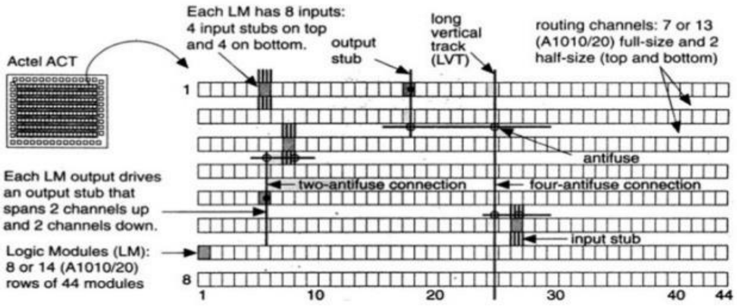
ACTEL ACT:

The interconnect architecture of ACTEL ACT family and is similar to a channeled gate array.

Wiring channel:

The channel routing uses dedicated rectangular areas of fixed size within chip called wiring channel. The horizontal channels run across the chip in the horizontal direction.In vertical direction, vertical channels run over the top of the basic logic cells or logic modules. Capacity of fixed wire channel is equal to the number of tacks it contains.

In a FPGA the interconnect is fixed at the time of manufacture. To provide interconnect programming, actel divides the fixed interconnect wires within each channel into various length or wire segments. The designer then programs the interconnections by blowing antiques and making connections between wire segments. The unwanted connections are left unprogrammed.


Actel ACT
Each LM has 8 inputs: 4 input stubs on top and 4 on bottom.
output stub
long vertical track (LVT)
routing channels: 7 or 13 (A1010/20) full-size and 2 half-size (top and bottom)
antifuse
two-antifuse connection
four-antifuse connection
input stub
Each LM output drives an output stub that spans 2 channels up and 2 channels down.
Logic Modules (LM): 8 or 14 (A1010/20) rows of 44 modules
1
8
1
10
20
30
40
44

Interconnect architecture used in an Actel ACT family FPGA

ACT 1 INTERCONNECT:

ACT 1 routing resource interconnection architecture uses 22 horizontal tracks per channel for signal routing with three tracks dedicated to VDD ,GND, and the global clock(GCLK).Four logic module input are available to the channel below the logic module and four input to the channel above the logic module.

Input stub:

Eight vertical tracks per logic module are available for inputs. This is the input stub.

Output stub:

Single logic module output connect to vertical track extends across the two channel above the module and across the two channels below the module. This is the output stub.One vertical track per column is a long vertical track (LVT) that spans the entire height of the chip.

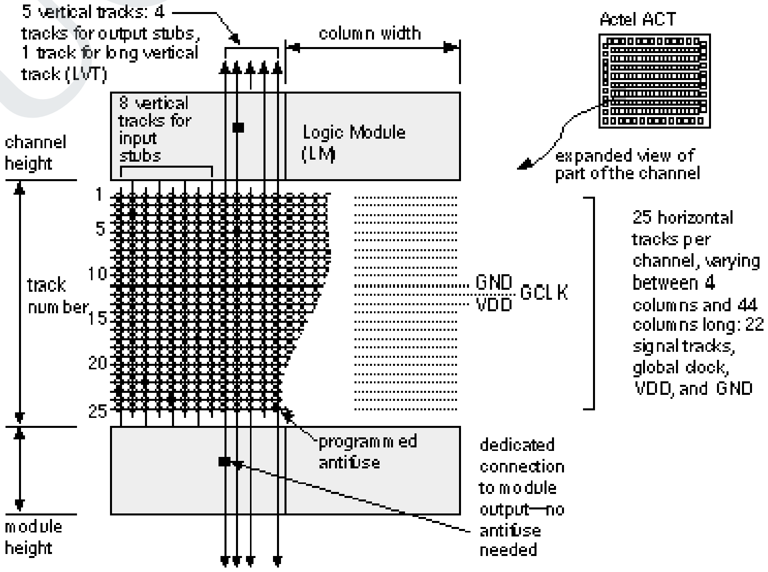

5 vertical tracks: 4 tracks for output stubs, 1 track for long vertical track (LVT)
column width
Actel ACT
8 vertical tracks for input stubs
Logic Module (LM)
expanded view of part of the channel
channel height
track number
1
5
10
15
20
25
GND
VDD
GCLK
25 horizontal tracks per channel, varying between 4 columns and 44 columns long: 22 signal tracks, global clock, VDD, and GND
programmed antifuse
dedicated connection to module output—no antifuse needed
module height

Fig: ACT1 horizontal and vertical channel architecture

ACT 2 AND ACT 3 INTERCONNECT:

The ACT 2 and ACT 3 architectures use increased interconnect resources. This reduces the no of connection the at need more than two antifuses. Delay is also reduced by decreasing the population of antifuses in the channels, and by decreasing the antifuses resistance of certain critical antifuses.

Channel density:

It is the absolute minimum no of tracks needed in a channel to make given set of connection. The ACT 2/3 logic modules need an extra two vertical tracks per channel.

The ACT 2/3 logic modules can accept five input ,rather than four input for the ACT1 modules.

The number of tracks per column increases from 13 to 15 in the ACT 2 architecture.

The greatest challenge facing the ACTEL FPGA architecture is the resistance of polysilicon antifuses.

------------------------**********------------------------

4. Explain about XILINX in detail. (MAY/JUN2016)

XILINX LCA:

XILINX LCA basic logic cells are called the configurable logic block or CLB.

CLB's are bigger and more complex than the ACTEL logic cells. Xilinx LCA uses coarse gain architecture. Xilinx CLB contain both combinational logic and flip flops.

XC 3000 CLB:

XC 3000 CLB which has five logic inputs., a common clock input ,an asynchronous direct rset input and an enable.

Two CLB outputs X and Y are connected independently to the Flipflop output QX and QY or to the combinational logic F and G using programmable MUX connected to the SRAM programming cells.

To implement five input AND ,F=A.B.C.D.E,set LUT cell number 31 with address "11111"in the 32 bit SRAM to "1".Since 32 bit LUT needs five variables to form unique address 32=2^5

XC 4000 LOGIC BLOCK:

This is a complicated basic logic cell containing 2 four input LUT'S that feed a three input LUT. This has special fast carry logic hardwired between CLB'S

MUX control logic maps four control inputs(c1-c4) into the following four inputs 1.H1 –LUT input 3. .EC –enable clock

2.DIN –DIRECT IN 4. S/R-set/reset control.

The control inputs(c1-c4) is used to control the use of F and G LUT as 32 bits of SRAM.

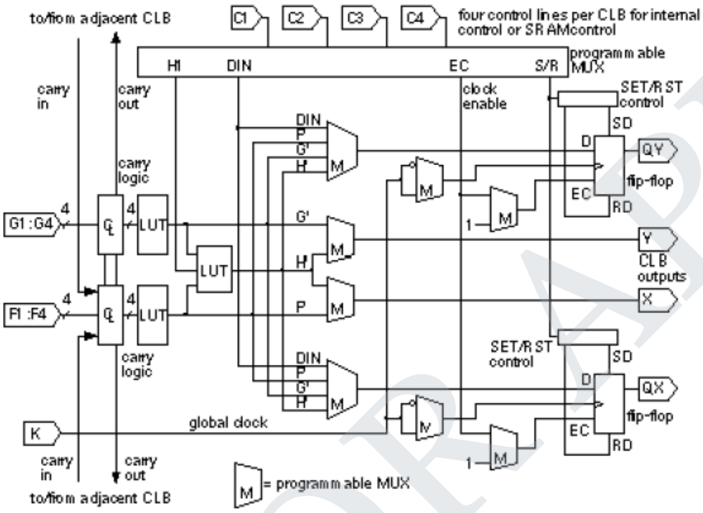


Fig: Xilinx XC4000 CLB

XC 5200 LOGIC BLOCK L:

This has 4 logic cells LC0-LC3.

The logic cell is similar to the CLB 's in the XC2000/3000/4000 CLB'S. This is simpler logic cell.

XC 5200 LC contains four input LUT, a flip flop ,and MUX to handle signal switching. The arithmetic carry logic is separate from the LUT's.

A limited capability to cae functions is provided to gang two LC's in parallel to provide the equivalent of five input LUT.

XILINX CLB analysis:

Usage of LUT in a Xilinx CLB to implement combinational logic is both an advantage and disadvantage. it means ,for example ,that an inverter is as slow as a five input NAND. On the other hand a LUT simplifies timing of synchronous logic ,simplifies the basic logic cell ,and matches the Xilinx SRAM programming technology well.

5. Explain about the Actel ACT in detail.

All programmable ASIC or FPGA contain a basis logic cell.the basic logi8c cell is REPLICATED IN A regular array across the chip.

ACTEL ACT has three logic family

1. ACT 1
2. ACT 2
3. ACT 3

ACT 1 logic module:

- Logic cells in ACTEL ACT 1 logic family are called logic modules.
- ACT 1 Family uses one type of logic modules .logic function is build using an actel logic module by connecting logic signals to some or all the logic module input and by connecting any remaining logic module input to VDD AND ground.

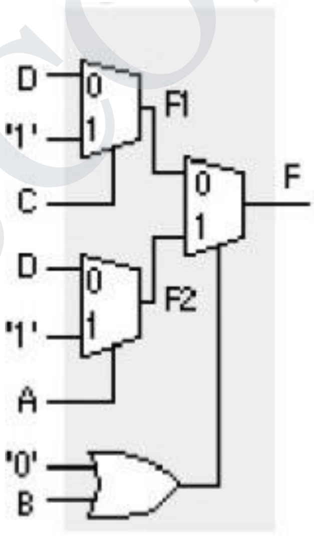


Fig: ACT 1 logic module

ACT 2 AND ACT 3 LOGIC MODULE:

- A Flipflop with two ACT 1 logic modules require added interconnect and associated parasitic capacitance to connect the two logic modules. For better efficiency extra antifuses in the logic module is used to cut down the parasitic capacitance.

- Another way is to use a separate Flipflop module, which reduces flexibility and reduces layout complexity.

- The ACT 2 and ACT 3 architectures uses two different types of logic modules, in which one is an equivalent of D flip flop.

- The ACT 2 C module is similar to the ACT 1 logic module, but is capable of implementing five input logic function. ACTEL calls its C module a combinational module even though the module implements combinational logic.

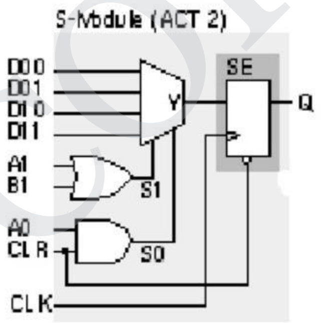


Fig: ACT 2 logic module

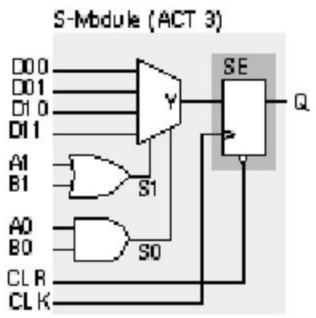


Fig: ACT 3 logic module

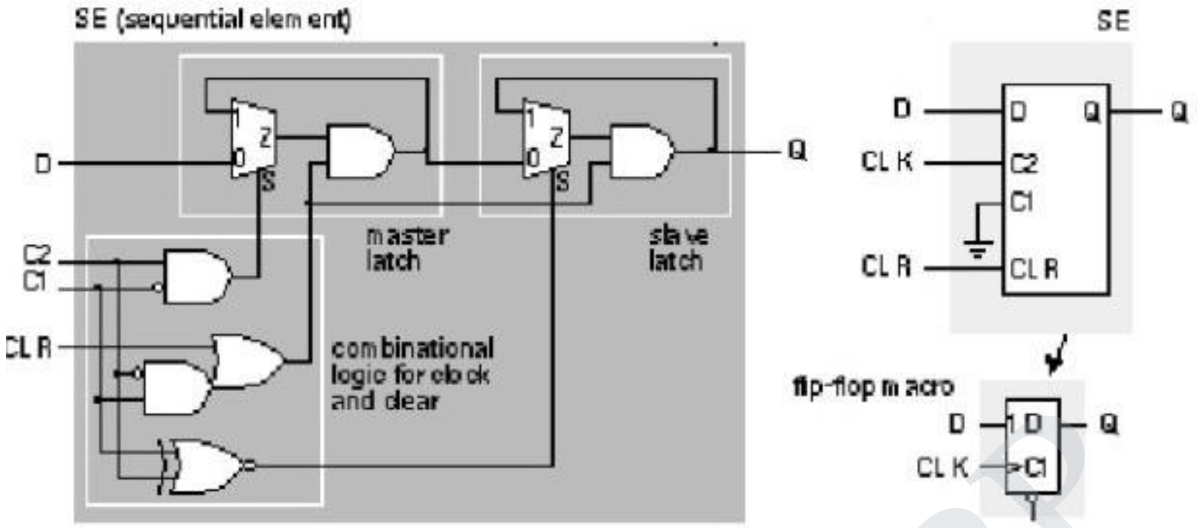


Fig: Sequential Element configured as positive edge triggered D flip flop

Question Paper Code:57297

B.E./B.Tech. DEGREE EXAMINATION , MAY /JUNE 2016

Sixth Semester

Electronics and Communication Engineering

EC6601/VLSI DESIGN

(Regulation 2013)

Time:Three Hours Maximum: 100 Marks

Answer All Questions

PART A-(10 x 2= 20 marks)

1. State channel length modulation. write down the equation for describing the channel length modulation effect in NMOS transistor.
2. What is latch up? How to prevent latch up.?
3. Give Elmore delay expression for propagation delay of an inverter..
4. Why single phase dynamic logic structure cannot be caed? Justify
5. Draw the switch level schematic of multiplexer base NMOS latch using NMOS only pass transistor for multiplexer.
6. What is clocked CMOS register?
7. What is meant by bit sliced data path organisation?
8. Determine propagation delay of n bit carry select adder.
9. What are feed through cell? State their uses.
10. State the features of full custom design

PART B-(5 x 16= 80 marks)

11. a)i. Describe the equation for source to drain current in the three region of operation of a MOS transistor and draw the V-I characteristics. (8)

ii. Explain in detail about the body effect and its effect in MOS device. (8)

(OR)

b)i. Explain the DC transfer characteristics of CMOS inverter with necessary condition for the different region of operation. (8)

ii. Discuss the principle of constant field and lateral scaling. Write the effects of the above scaling method on the device characteristics. (8)

12. a)i. Draw the static CMOS logic circuit for the following expression (8)

a).Y=(A.B.C.D)’

b).Y=(D(A+BC))'

ii. Discuss in detail the characteristics of CMOS transmission gate? (8)

(OR)

b. What are the sources of power dissipation in CMOS and discuss various design technique to reduce power dissipation in CMOS? (16)

13.a. Explain the operation of master slave based edge triggered register? (16)

(OR)

b. Discuss in details various pipelining approaches to optimize sequential circuits? (16)

14.a. Design a 16 nbit carry bypass and carry select adder and discuss their features. (16)

(OR)

b. Design 4x4 array multiplier and write the equation for delay. (16)

15 a. With neat sketch explain the CLB ,IOB and programmable interconnect of an FPGA Device. (16)

(OR)

b. Write brief notes on (16)

A)Full Custom ASIC

B)Semi Custom ASIC

B.E./B.TECH. DEGREE EXAMINATIONS, MAY / JUNE - 2008

(REGULATIONS 2004)

SIXTH SEMESTER EC 1401

– VLSI DESIGN

ELECTRONICS AND COMMUNICATION ENGINEERING

PART-A

1. State any two differences between CMOS and Bipolar technology.
2. Draw the stick diagram for an n-type enhancement mode transistor.
3. What is latch up problem in CMOS circuits?
4. Give the expressions for rise time and fall time in CMOS inverter circuit.
5. Define the syntax for Architecture in Verilog HDL.
6. How is component declaration done in VHDL?
7. Differentiate between Full custom and Cell based ASICs.
8. State any two features of Xilinx programmable GA.
9. State the need for testing.
10. List the design steps required for testing in CMOS chip design.

PART-B

11. (a) (i) With neat diagrams explain the steps involved in the p-well process of CMOS fabrication. (8)

(ii)Discuss the lambda based design rules for NMOS transistor. (8)

(Or)

(b) (i) Describe in detail with neat sketches the Twin Tub method of CMOS fabrication. (8)

(ii) With neat diagram of Latch-up effect in p-well structure, explain Latch-up problem and the steps involved to overcome it. (8)

12. (a) (i) Derive the pull-up to pull-down ratio for an NMOS inverter driven by another NMOS inverter. (8)

(ii) Explain in detail the MOS transistor Figure of merit. Obtain an expression for it.

(Or)

(b) (i) Explain Pass Transistor and Transmission gates with neat sketches.

(ii) Draw the stick and layout diagrams of an NMOS inverter. (8)

13. (a)(i) With a neat flow chart explain the VLSI design flow. (8)

(ii) Explain the syntax of conditional statements in Verilog HDL with examples. (8)

(Or)

(b)(i) Explain in detail Behavioural and RTL modeling. (8)

(ii) Write the program using Verilog HDL to implement a full adder circuit. (8)

14. (a)(i) Explain Gate Array based ASICs with diagrams. (8)

(ii) With a neat flow chart explain ASIC design flow and the steps involved in the design. (8)

(Or)

(b)(i) With a block diagram describe Xilinx I/O cell. (8)

(ii) Explain the Actel ACT family interconnect and its routing resources. (8)

15. (a) (i) Explain in detail Boundary-Scan Test. (8)

(ii) Enumerate on physical faults with examples. (8)

(Or)

(b) (i) Explain Built-in Self Test. (8)

(ii) Describe the testing techniques at chip level and at system level. (8)

B.E./B.TECH. DEGREE EXAMINATIONS, MAY / JUNE - 2009

(REGULATIONS 2004)

SIXTH SEMESTER EC

1401 – VLSI DESIGN

ELECTRONICS AND COMMUNICATION ENGINEERING

PART-A

1. Define SSI, MSI, LSI and VLSI.
2. What are the different tools available in a typical CAD tool set?
3. What is meant by “body effect”?
4. Draw the schematic diagram of the tristate inverter.
5. What are the different phases of VLSI Design flow?
6. Write HDL code for Half-adder.
7. Define Transmission Gate.
8. What is meant by Full Custom Design?
9. What is the basic principle of electronic testing?
10. State all the test vectors to test3 input NAND gate.

PART-B

11. (a) Explain the various features of CMOS technology. (16)

(Or)

(b) Explain the characteristics of bipolar transistors. (16)

12. (a) What is meant by channel length modulation? Explain. (16)

(Or)

(b) Derive the equation for threshold voltage in PMOS Enhancement transistor. (16)

13. (a) Explain various features of gate level modeling and switch level modeling. (16)

(Or)

(b) Write HDL code for Ripple Carry Adder. (16)

14. (a) Explain the features of ASIC design flow. (16)

(Or)

(b) Discuss the features of Channeled Gate Array, Channel less Gate Array and Structured Gate Array. (16)

15. (a) Write briefly about different test strategies of testing digital circuits. (16)

(Or)

(b) Explain the importance of system level testing techniques. (16)