SKP Engineering College

Tiruvannamalai – 606611

A Course Material

on

VLSI Design

By

N.Anbuselvan

Assistant Professor

Electronics and Communication Engineering Department

Quality Certificate

This is to Certify that the Electronic Study Material

Subject Code: EC6601

Subject Name: VLSI DESIGN

Year/Sem: III/VI

Being prepared by me and it meets the knowledge requirement of the University curriculum.

Signature of the Author

Name: N.Anbuselvan

Designation: Assistant Professor

This is to certify that the course material being prepared by Mr. N.Anbuselvan is of the adequate quality. He has referred more than five books and one among them is from abroad author.

Signature of HD

Name:Mr.R.Saravanakumar

Seal:

Signature of the Principal

Name: Dr.V.Subramania Bharathi

Seal:

EC6601 VLSI DESIGN L T P C 3 0 0 3

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.

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

OUTCOMES: Upon completion of the course, students should

- Explain the basic CMOS circuits and the CMOS process technology.
- Discuss the techniques of chip design using programmable devices.
- Model the digital system using Hardware Description Language.

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.

CONTENTS

S.No	Particulars	Page
1	Unit – I	6
2	Unit – II	32
3	Unit – III	55
4	Unit – IV	93
5	Unit – V	136

Unit – I

MOS Transistor Principle

Part – A

1. What is CMOS technology? [**CO1-L1]**

Complementary Metal Oxide Semiconductor (CMOS) in which both n-channel MOS and p-channel MOS are fabricated in the same IC.

2. List the advantages of CMOS over NMOS technology? [**CO1-L1]**

In CMOS technology the aluminum gates of the transistor are replaced by poly silicon gate.
The main advantage of CMOS over NMOS is low power consumption.
In CMOS technology the device sizes can be easily scalable than NMOS.

3. What are the advantages of CMOS technology? [**CO1-L1]**

Low power consumption.
High performance.
Scalable threshold voltage.
High noise margin.
Low output drive current.

4. What are the disadvantages of CMOS technology? [**CO1-L1]**

Low resistance to produce deviations and temperature changes.
Low switching speed at large values of capacitive loads.

5. Classify the design rule? [**CO1-L2]**

Design rules are the communication link between the designer specifying requirements and the fabricator who materializes them. The design rule conform to a set of geometric constraints or rule specify the minimum allowable line widths for physical objects on-chip such as metal and poly silicon interconnects or diffusion area, minimum feature dimensions and minimum allowable separations between two layers.

6. Classify the stick diagram? [**CO1-L2]**

Stick diagram are the key element of designing a circuit used to convey layer information through the use of a color code.

7. What is micron design rule? [**CO1-L1]**

Micron rules specify the layout constraints such as minimum feature sizes and minimum allowable feature separations are stated in terms of absolute dimensions in micrometers.

8. Give the important of Lambda design rule? [**CO1**-**L2]**

Lambda rule specify the layout constraints such as minimum feature sizes and minimum allowable feature separations are stated in terms of a single parameter (λ) and thus allow linear, proportional scaling of all geometrical constraints.

9. Define is DRC? [**CO1**-**L2]**

Design Rule Check program looks for design rule violations in the layout. It checks for minimum spacing and minimum size and ensures that combinations of layers from legal components.

10. Mention MOS transistor characteristics? [**CO1**-**L2]**

Metal Oxide Semiconductor is a three terminal device having source, drain and gate. The resistance path between the drain and the source is controlled by applying a voltage to the gate. The Normal conduction characteristics of an MOS transistor can be categorized as cut-off region Non saturated region and saturated region. It is defined as the minimum voltage at which the device starts conduction (ie) turns on.

11. What are different operating modes of MOS transistor? [**CO1**-**L1]**

Accumulation mode
Depletion mode
Inversion mode

12. Mention is accumulation mode? [**CO1**-**L2]**

When the gate to source voltage (Vgs) is much less than the threshold voltage (Vt) then it is termed as the accumulation mode. There is no conduction between source and drain. The device is turned off.

13. What is depletion mode ? [**CO1**-**L1]**

When the gate to source voltage (Vgs) is increased greater than the threshold voltage (Vt) the electrons are attracted towards the gate while the holes are repelled causing a depletion region under the gate. This is called depletion mode.

14. What is inversion mode? [**CO1**-**L1]**

When Vgs is raised above the Vt the electrons are attracted to the gate region. Under such a condition the surface of the underlying p-type silicon is said to be a inverted to n-type,and provides a conduction path between a source and drain. The device is turned on. This is called inversion mode.

15. What are three operating regions of MOS transistor? [**CO1**-**L1]**

Cut-off region
Non saturated region
Saturated region.

16. Give the condition for cut-off region? [**CO1-L2]**

The region where the current flow is essentially zero is called cut-off region.

(ie) Ids=0, Vgs ≤ Vt.

17. Give the condition for Non-saturated region? [**CO1-L2]**

Weak inversion region where the drain current is dependent on the gate and the drain voltage is called non saturated region

(ie) 0<Vds < Vgs – Vt

18. Give the condition for saturated region? [**CO1-L2]**

Channel is strongly inverted and the drain current flow is ideally independent of the drain-source voltage is called saturated region.

(ie) 0 < Vgs – Vt< Vds

19. What is body effect? [**CO1-L1]**

The threshold voltage Vt is constant with respect to voltage difference between source and the substrate is called body effect.

20. Define threshold voltage for a MOSFET? [**CO1-L1]**

The threshold voltage of a MOSFET is usually defined as the gate voltage were an inversion layer form s at the interface between the insulating layer (oxide) and the substrate (body) of the transistor.

21. What is body effect in MOSFETs? [**CO1-L1]**

The body effect is change in the threshold voltage by the change in the VSB (the source bulk voltage). Since the body influences the threshold voltage (when it is not tied to the source), itcan be considered as second gate or back gate.

22. List is the specifications of MOSFET? [**CO1-L1]**

Breakdown voltages
Forward transconductance
Drain to source on resistance
Switching characteristics
Zero gate voltage drain current, Idss
Input capacitance,Ciss

23. What are the different generations of integrated circuits? [**CO1-L1]**

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

24. Mention the major advantages of IC? [**CO1-L2]**

Size is less
High speed
Low power consumption

25. What is the objective of layout rules? [**CO1-L1]**

To build reliably functional circuits in as small an area as possible.
To provide a necessary communication link circuit designer and process engineer during manufacturing.
To obtain a circuit with optimum yield in smallest possible area.

Part – B

1. Explain in detail about the ideal I-V characteristics and non-ideal I-V characteristics of nMOS and pMOS devices. [**CO1-H1]**

The fundamental physical characteristics of the MOS transistor, in which the electrical currents and voltages are the most important quantities. The link between physical design and logic networks can be established. Figure 2.1 depicts various symbols used for the MOS transistors. The symbol shown in Figure 2.1(a) is used to indicate only switch logic, while that in Figure 2.1(b) shows the substrate connection.

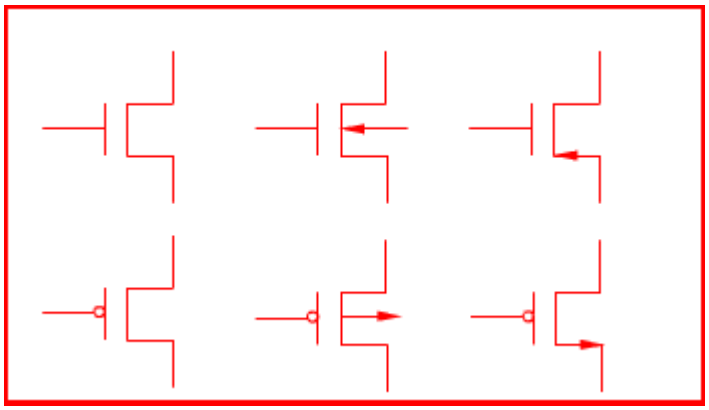


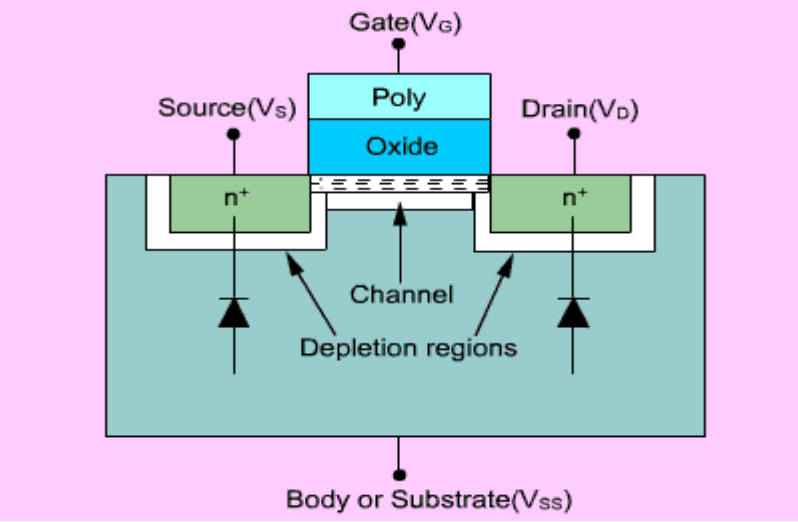
Figure 2.1 various symbols for MOS transistors

This chapter first discusses about the basic electrical and physical properties of the Metal Oxide Semiconductor (MOS) transistors. The structure and operation of the nMOS and pMOS transistors are addressed, following which the concepts of threshold voltage and body effect are explained. The current-voltage equation of a MOS device for different regions of operation is next established.

It is based on considering the effects of external bias conditions on charge distribution in MOS system and on conductance of free carriers on one hand, and the fact that the current flow depends only on the majority carrier flow between the two device terminals. Various second-order effects observed in MOSFETs are next dealt with. Subsequently,

the complementary MOS (CMOS) inverter is taken up. Its DC characteristics, noise margin and the small-signal characteristics are discussed. Various load configurations of MOS inverters including passive resistance as well as transistors are presented. The differential inverter involving double-ended inputs and outputs are discussed. The complementary switch or the transmission gate, the tristate inverter and the bipolar devices are briefly dealt with.

The channel is covered by a thin insulating layer of silicon dioxide (SiO_2). The gate electrode, made of polycrystalline silicon (polysilicon or poly in short) stands over this oxide. As the oxide layer is an insulator, the DC current from the gate to the channel is zero. The source and the drain regions are indistinguishable due to the physical symmetry of the structure. The current carriers enter the device through the source terminal while they leave the device by the drain.The switching behaviour of a MOS device is characterized by an important parameter called the *threshold* voltage (V_{th}), which is defined as the minimum voltage, that must be established between the gate and the source (or between the gate and the substrate, if the source and the substrate are shorted together), to enable the device to conduct (or "turn on"). In the enhancement mode device, the channel is *not* established and the device is in a *non-conducting* (also called *cutoff* or *sub-threshold*) state, for $V_{GS} < V_{th}$. If the gate is connected to a suitable *positive voltage* with respect to the source, then the electric field established between the gate and the source will *induce* a charge inversion region, whereby a conducting path is formed between the source and the drain. In the enhancement mode device, the formation of the channel is *enhanced* in the presence of the gate voltage.



By implanting suitable impurities in the region between the source and the drain before depositing the insulating oxide and the gate, a channel can also be established. Thus the source and the drain are conn $V_{GS}=0$ (below the threshold voltage). To make the channel disappear, one has to apply a suitable negative voltage on the gate. As the channel in this device can be *depleted* of the carriers by applying a negative voltage V_{td} say, such a device is called a *depletion* mode device. Figure 2.3 shows the

arrangement in a depletion mode MOS device. For an n-type depletion mode device, penta-ected by a conducting channel even though the voltage between the gate and the source, namely valent impurities like phosphorus is used

To describe the operation of an nMOS enhancement device, note that a positive voltage is applied between the source and the drain (V_{DS}). No current flows from the source and the drain at a zero gate bias (that is, V_{GS}= 0). This is because the source and the drain are insulated from each other by the two reverse-biased diodes as shown in Figure However, as a voltage, positive relative to the source and the substrate, is applied to the gate, an electric field is produced across the p-type substrate, This electric field attracts the electrons toward the gate and repels the holes. If the gate voltage is adequately high, the region under the gate changes from p-type to n-type, and it provides a conduction path between the source and the drain. A very thin surface of the p-type substrate is then said to be *inverted,* and the channel is said to be an *n-channel*.

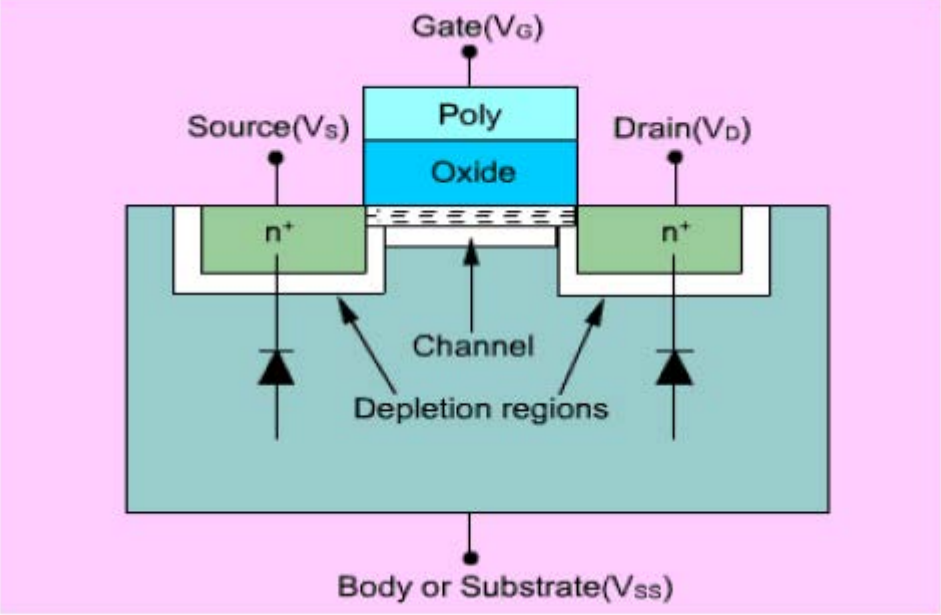


Figure 2.2: Structure of an nMOS enhancement mode transistor. Note that $V_{GS} > V_{th}$, and V_{DS} =0.

To explain in more detail the electrical behaviour of the MOS structure under external bias, assume that the substrate voltage V_{SS} = 0, and that the gate voltage V_G is the controlling parameter. Three distinct operating regions, namely *accumulation, depletion* and *inversion* are identified based on polarity and magnitude of V_G .

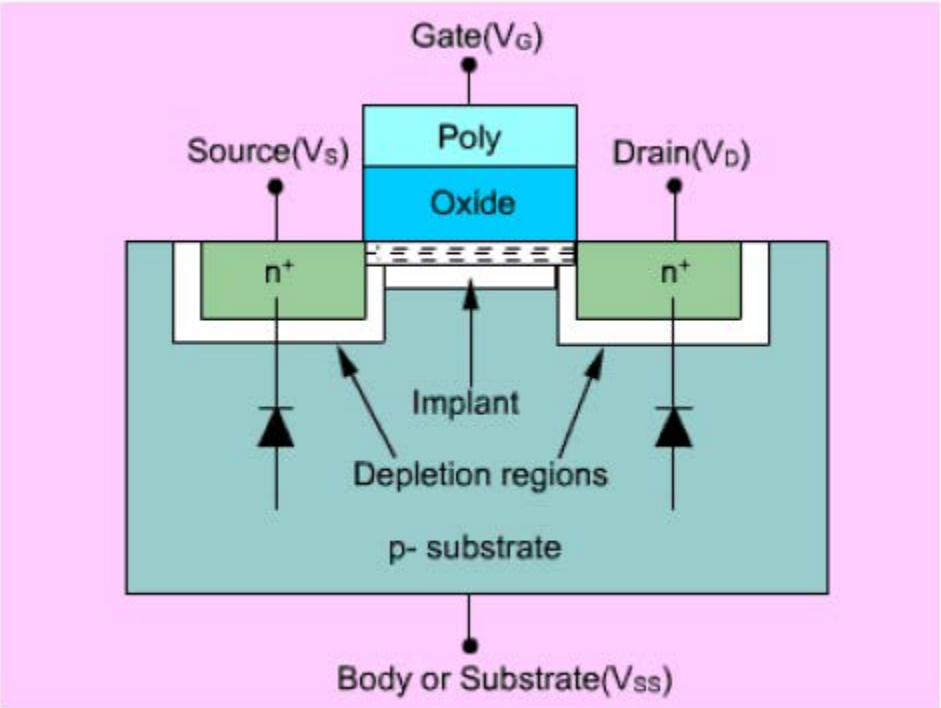
If a negative voltage V_G is applied to the gate electrode, the holes in the p-type substrate are attracted towards the oxide-semiconductor interface. As the majority carrier (hole) concentration near the surface is larger than the equilibrium concentration in the substrate, this condition is referred to as the carrier *accumulation* on the surface. In this case, the oxide electric field is directed towards the gate electrode. Although the hole density increases near the surface in response to the negative gate bias, the

minority carrier (electron) concentration goes down as the electrons are repelled deeper into the substrate.

Consider next the situation when a small positive voltage V_G. is applied to the gate. The direction of the electric field across the oxide will now be towards the substrate. The holes (majority carriers) are now driven back into the substrate, leaving the negatively charged immobile acceptor ions. Lack of majority carriers create a *depletion* region near the surface. Almost no mobile carriers are found near the semiconductor-oxide interface under this bias condition.

Next, let us investigate the effect of further increase in the positive gate bias. At a voltage $V_{GS} = V_{th}$, the region near the semiconductor surface acquires the properties of n-type material. This n-type surface layer however, is not due to any doping operation, but rather by *inversion* of the originally p-type semiconductor owing to the applied voltage. This inverted layer, which is separated from the p-type substrate by a depletion region, accounts for the MOS transistor operation. That is, the thin inversion layer with a large mobile electron concentration, which is brought about by a sufficiently large positive voltage between the gate and the source, can be effectively used for conducting current between the source and the drain terminals of the MOS transistor. *Strong inversion* is said to occur when the concentration of the mobile electrons on the surface equals that of the holes in the underlying p-type substrate.

As far as the electrical characteristics are concerned, an nMOS device acts like a voltage-controlled switch that starts to conduct when V_G (or, the gate voltage with respect to the source) is at least equal to V_{th} (the threshold voltage of the device). Under this condition, with a voltage V_{DS} applied between the source and the drain, the flow of current across the channel occurs as a result of interaction of the electric fields due to the voltages V_{DS} and V_{GS}. The field due to V_{DS} sweeps the electrons from the source toward the drain.As the voltage V_{DS} increases, a resistive drop occurs across the channel. Thus the voltage between the gate and the channel varies with the distance along the channel. This changes the shape of the channel, which becomes tapered towards the drain end.

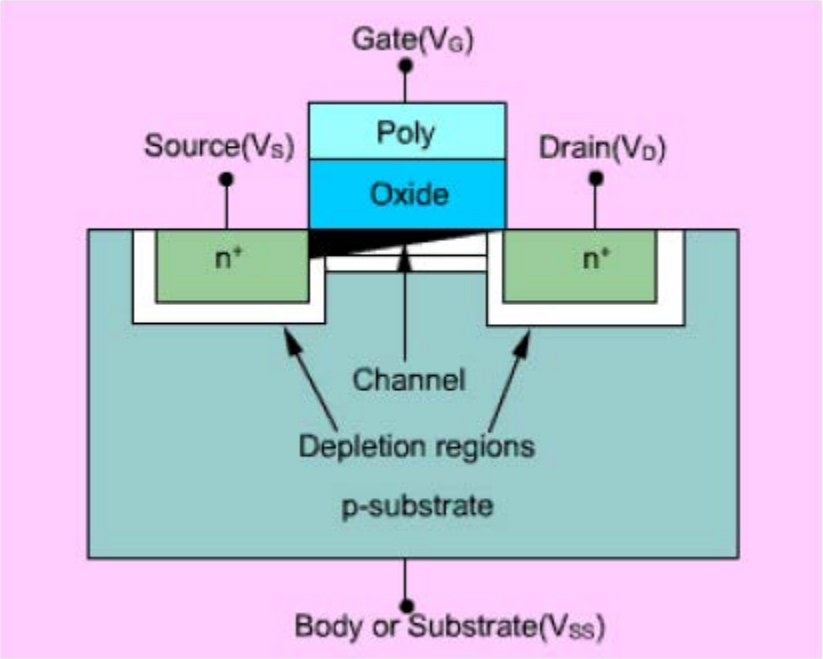


An nMOS enhancement mode transistor in non-saturated (linear or resistive) mode. Note that $V_{GS} > V_{th}$, and $V_{DS} < V_{GS} - V_{th}$.

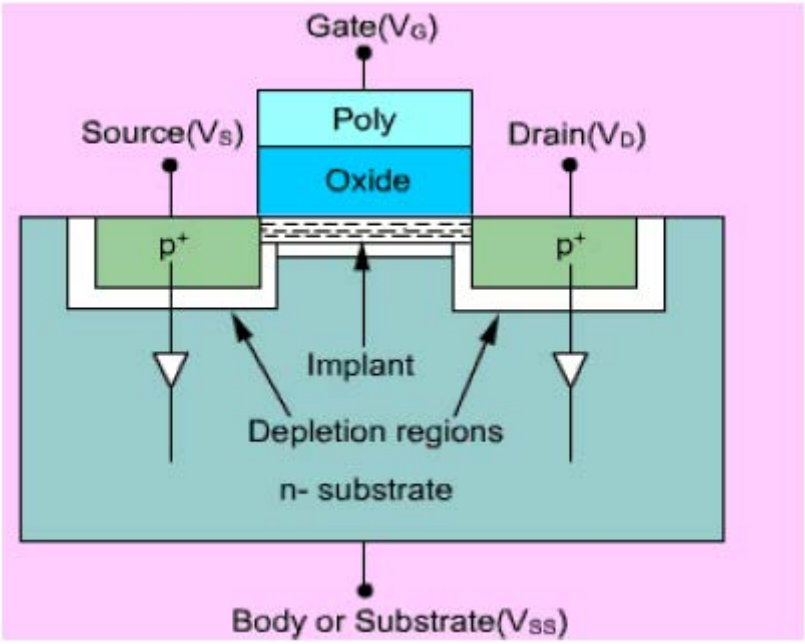
However, under the circumstance $V_{DS} > V_{GS} - V_{th}$, when the gate voltage relative to drain voltage is insufficient to form the channel (that is, $V_{GD} < V_{th}$), the channel is terminated before the drain end. The channel is then said to be pinched off. This region of operation, known as *saturated* or *pinch-off* condition, is portrayed in Figure 2.5. The effective channel length is thus reduced as the inversion layer near the drain end vanishes. As the majority carriers (electrons) reach the end of the channel, they are swept to the drain by the drift action of the field due to the drain voltage. In the saturated state, the channel current is controlled by the gate voltage and is almost independent of the drain voltage.

In short, the nMOS transistor possesses the three following regions of operation:

- Cutoff, sub-threshold or non-conducting zone
- Non-saturation or linear zone



An nMOS enhancement mode transistor in saturated (pinch-off) mode. Note that $V_{GS} > V_{th}$, and $V_{DS} > V_{GS} - V_{th}$.



Structure of an pMOS enhancement mode transistor. Note that $V_{GS} < V_{th}$, and V_{DS} =0.

Saturation region

Thus far, we have dealt with principle of operation of an nMOS transistor. A p-channel transistor can be realized by interchanging the n-type and the p-type regions, as shown in Figure 2.6. In case of an pMOS enhancement-mode transistor, the threshold voltage

V_{th} is negative. As the gate is made negative with respect to the source by at least $|V_{th}|$, the holes are attracted into the thin region below the gate, creating an inverted *p-channel.* Thus, a conduction path is created for the majority carriers (holes) between the source and the drain. Moreover, a negative drain voltage V DS draws the holes through the channel from the source to the drain.

Threshold Voltage and Body Effect:

The threshold voltage V_{th} for a nMOS transistor is the minimum amount of the gate-to-source voltage V_{GS} necessary to cause surface inversion so as to create the conducting channel between the source and the drain. For $V_{GS} < V_{th}$, no current can flow between the source and the drain. For $V_{GS} > V_{th}$, a larger number of minority carriers (electrons in case of an nMOS transistor) are drawn to the surface, increasing the channel current. However, the surface potential and the depletion region width remain almost unchanged as V_{GS} is increased beyond the threshold voltage.

The physical components determining the threshold voltage are the following.

- work function difference between the gate and the substrate.
- gate voltage portion spent to change the surface potential.
- gate voltage part accounting for the depletion region charge.
- gate voltage component to offset the fixed charges in the gate oxide and the silicon-oxide boundary.

Although the following analysis pertains to an nMOS device, it can be simply modified to reason for a p-channel device. The work function difference ϕ_{GS} between the doped polysilicon gate and the p-type substrate, which depends on the substrate doping, makes up the first component of the threshold voltage. The externally applied gate voltage must also account for the strong inversion at the surface, expressed in the form of surface potential $2\phi_F$, where ϕ_F denotes the distance between the intrinsic energy level E_I and the Fermi level E_F of the p-type semiconductor substrate.

The factor 2 comes due to the fact that in the bulk, the semiconductor is p-type, where E_I is above E_F by ϕ_F, while at the inverted n-type region at the surface E_I is below E_F by ϕ_F, and thus the amount of the band bending is $2\phi_F$. This is the second component of the threshold voltage. The potential difference ϕ_F between E_I and E_F is given as

$$\phi_F = \frac{kT}{q} \ln\left(\frac{N_A}{n_i}\right)$$

where *k*: Boltzmann constant, *T*: temperature, *q* : electron charge N_A : acceptor concentration in the p-substrate and n i : intrinsic carrier concentration. The expression kT/q is 0.02586 volt at 300 K.

The applied gate voltage must also be large enough to create the depletion charge. Note that the charge per unit area in the depletion region at strong inversion is given by

$$Q_{d0} = -2(\varepsilon_S q N_A \phi_F)^{1/2}$$

where ε_S is the substrate permittivity. If the source is biased at a potential V_{SB} with respect to the substrate, then the depletion charge density is given by

$$Q_d = -2(\varepsilon_S q N_A (\phi_F + V_{SB}))^{1/2}$$

The component of the threshold voltage that offsets the depletion charge is then given by $-Q_d / C_{ox}$, where C_{ox} is the gate oxide capacitance per unit area, or $C_{ox} = \varepsilon_{ox} / t_{ox}$ (ratio of the oxide permittivity and the oxide thickness).

A set of positive charges arises from the interface states at the Si-SiO_2 interface. These charges, denoted as Q_i , occur from the abrupt termination of the semiconductor crystal lattice at the oxide interface. The component of the gate voltage needed to offset this positive charge (which induces an equivalent negative charge in the semiconductor) is $-Q_i / C_{ox}$. On combining all the four voltage components, the threshold voltage V_{TO}, for zero substrate bias, is expressed as

$$V_{T0} = \phi_{GS} - 2\phi_F - \frac{Q_{d0}}{C_{ox}} - \frac{Q_i}{C_{ox}}$$

For non-zero substrate bias, however, the depletion charge density needs to be modified to include the effect of V_{SB} on that charge, resulting in the following generalized expression for the threshold voltage, namely

$$V_T = \phi_{GS} - 2\phi_F - \frac{Q_d}{C_{ox}} - \frac{Q_i}{C_{ox}}$$

The generalized form of the threshold voltage can also be written a

$$V_T = \phi_{GS} - 2\phi_F - \frac{Q_{d0}}{C_{ox}} - \frac{Q_i}{C_{ox}} - \frac{Q_d - Q_{d0}}{C_{ox}} = V_{T0} - \frac{Q_d - Q_{d0}}{C_{ox}}$$

$$V_T = \phi_{GS} - 2\phi_F - \frac{Q_{d0}}{C_{ox}} - \frac{Q_i}{C_{ox}} - \frac{Q_d - Q_{d0}}{C_{ox}} = V_{T0} - \frac{Q_d - Q_{d0}}{C_{ox}}$$

Note that the threshold voltage differs from V_{TO} by an additive term due to substrate bias. This term, which depends on the material parameters and the source-to-substrate voltage V_{SB} , is given by

$$\frac{Q_d - Q_{d0}}{C_{ox}} = -\frac{\sqrt{2qN_A\varepsilon_S}}{C_{ox}}\left(\sqrt{|2\phi_F + V_{SB}|} - \sqrt{|2\phi_F|}\right)$$

Thus, in its most general form, the threshold voltage is determined as

$$V_T = V_{T0} + \gamma\left(\sqrt{|2\phi_F + V_{SB}|} - \sqrt{|2\phi_F|}\right) \quad \text{..... (1.1)}$$

in which the parameter γ, known as the *substrate-bias* **(or** *body-effect* **)** coefficient is given by

$$\gamma = \frac{\sqrt{2qN_A\varepsilon_S}}{C_{ox}} \quad \text{..... (1.2)}$$

The threshold voltage expression given by (1.1) can be applied to n-channel as well as p-channel transistors. However, some of the parameters have opposite polarities for the pMOS and the nMOS transistors. For example, the substrate bias voltage V_{SB} is positive in nMOS and negative in pMOS devices. Also, the substrate potential difference ϕ_F is negative in nMOS, and positive in pMOS. Whereas, the body-effect coefficient γ is positive in nMOS and negative in pMOS. Typically, the threshold voltage of an enhancement mode n-channel transistor is positive, while that of a p-channel transistor is negative.

2.Given the following parameters, namely the acceptor concentration of p-substrate N_A =10^{16} cm^{-3} , polysilicon gate doping concentration N_D =10^{16} cm^{-3} , intrinsic concentration of Si, n_i =1.45 X 10^{10} cm^{-3} , gate oxide thickness t_{ox} =500 Å and oxide-interface fixed charge density N_{ox} =4 X $10^{10}$$cm^{-2}$, calculate the threshold voltage V_{TO} at V_{SB}=0. **[CO1-H3]**

The potential difference between E_I and E_F for the p-substrate is

$$\phi_F = KT/q \ln\left(N_A/n_i\right) = 0.026V\, C_n\left(10^{16}/1.45\times10^{10}\right) = 0.35V$$

For the polysilicon gate, as the doping concentration is extremely high, the heavily doped n-type gate material can be assumed to be degenerate. That is, the Fermi level E_F is almost coincident with the bottom of the conduction band E C . Hence, assuming that the intrinsic energy level E_I is at the middle of the band gap, the potential difference between E_I and E_F for the gate is ϕ_F= ½ (energy band gap of Si) = 1/2 X 1.1 = 0.55 V.

Thus, the work function difference ϕ_{GS} between the doped polysilicon gate and the p-type substrate is -0.35 V - 0.55 V = -0.90 V.

The depletion charge density at V_{SB} =0 is

$$Q_{d0} = -2\left(\varepsilon_S q N_A \phi_F\right)^{1/2} = -2\left(11.7\times 8.85\times 10^{-14}\times 1.6\times 10^{-19}\times 10^{16}\times 0.35\right)^{1/2} = -4.82\times 10^{-8}\ C/cm^2$$

The oxide-interface charge density is

$$Q_i = q.N_{ss} = 1.6\times 10^{-19}\ \text{C}\ \times 4\ \times 10^{10}\text{cm}^{-2} = 6.4\times 10^{-19}\ \text{C/m}^2$$

The gate oxide capacitance per unit area is (using dielectric constant of SiO_2 as 3.97)

$$C_{OX} = \varepsilon_{OX} / t_{OX} = \left(3.97\times 8.85\times 10^{-14}\ \text{F/cm}\right)/\left(500\times 10^{-8}\ \text{cm}\right) = 7.03\times 10^{-8}\ \text{F/cm}^2$$

Combining the four components, the threshold voltage can now be computed as

$$V_{TO} = \phi_{GS} - 2\phi_F\left(\text{substrate}\right) - Q_{do} / C_{OX} - Q_i / C_{OX} = -0.90 - (-0.69) - 0.09 = 0.40\ \text{volt}$$

3. Write a short on Body Effect. [**CO1-L1]**

The transistors in a MOS device seen so far are built on a common substrate. Thus, the substrate voltages of all such transistors are equal. However, while one designs a complex gate using MOS transistors, several devices may have to be connected in series. This will result in different source-to-substrate voltages for different devices. For example, in the NAND gate shown in Figure 1.5, the nMOS transistors are in series, whereby the source-to-substrate voltage V_{SB} of the device corresponding to the input A is higher than that of the device for the input B.

Under normal conditions ($V_{GS} > V_{th}$), the depletion layer width remains unchanged and the charge carriers are drawn into the channel from the source. As the substrate bias V_{SB} is increased, the depletion layer width corresponding to the source-substrate field-induced junction also increases. This results in an increase in the density of the fixed charges in the depletion layer. For charge neutrality to be valid, the channel charge must go down. The consequence is that the substrate bias V_{SB} gets added to the channel-substrate junction potential. This leads to an increase of the gate-channel voltage drop.

4. Derive the MOS Device Current -Voltage Equations. [**CO1-H2]**

This section first derives the current-voltage relationships for various bias conditions in a MOS transistor. Although the subsequent discussion is centred on an nMOS transistor, the basic expressions can be derived for a pMOS transistor by simply replacing the electron mobility μ_n by the hole mobility μ_p and reversing the polarities of voltages and currents.

As mentioned in the earlier section, the fundamental operation of a MOS transistor arises out of the gate voltage V_{GS} (between the gate and the source) creating a channel between the source and the drain, attracting the majority carriers from the source and causing them to move towards the drain under the influence of an electric field due to

the voltage V_{DS} (between the drain and the source). The corresponding current I_{DS} depends on both V_{GS} and V_{DS} .

Basic DC Equations

Let us consider the simplified structure of an nMOS transistor shown in Figure 2.8, in which the majority carriers electrons flow from the source to the drain.

The conventional current flowing from the drain to the source is given by $I_{DS} = -I_{SD} = (\text{charge induced in channel})/(\text{electron transit time}) = Q_C / \tau_n$ Now, transit time τ_n = (length of the channel) / (electron velocity) = L / v where velocity is given by the electron mobility and electric field; or, $v = \mu_n E_{DS}$ Now, $E_{DS} = V_{DS}/ L$, so that velocity $v = (\mu_n V_{DS}) / L$ Thus, the transit time is $\tau_n = L^2 / (\mu_n V_{DS})$

At room temperature (300 K), typical values of the electron and hole mobility are given by $\mu_n = 650 \text{ cm}^2/V-\text{sec}$, and $\mu_p = 240 \text{ cm}^2/V-\text{sec}$

We shall derive the current-voltage relationship separately for the linear (or non-saturated) region and the saturated region of operation.

Linear region: Note that this region of operation implies the existence of the uninterrupted channel between the source and the drain, which is ensured by the voltage relation $V_{GS} - V_{th} > V_{DS}$.

In the channel, the voltage between the gate and the varies linearly with the distance x from the source due to the IR drop in the channel. Assume that the device is not saturated and the average channel voltage is V_{DS} /2.

The effective gate voltage $V_{G,eff} = V_{gs} - V_{th}$

Charge per unit area = $E_g \varepsilon_{ins} \varepsilon_0$

where E_g average electric field from gate to channel, ε_{ins}: relative permittivity of oxide between gate and channel (~4.0 for SiO_2), and ε_0: free space permittivity (8.85 x 10^{-14} F/cm). So, induced charge $Q_C = E_g \varepsilon_{ins} \varepsilon_0 WL$.

where W: width of the gate and L : length of channel.

$$Q_C = E_g \varepsilon_{ins} \varepsilon_0 WL$$

$$Q_C = WL\varepsilon_{ins}\varepsilon_0 / D\{(V_{GS} - V_{th}) - V_{DS}/2\}$$

$$\tau_n = L^2 / (\mu_n V_{DS})$$

Thus, the current from the drain to the source may be expressed as

$$I_{DS} = Q_C / \tau_n = \varepsilon_{ins}\varepsilon_0 \mu_n W / (LD)\left\{(V_{GS} - V_{th}) - V_{DS}/2\right\} V_{DS}$$

Thus, in the non-saturated region, where $V_{DS} < V_{GS} - V_{th}$

$$I_{DS} = (KW)/L\left\{(V_{GS} - V_{th})V_{DS} - V^2_{DS}/2\right\} \quad \text{..... (1.2)}$$

where the parameter $K = (\varepsilon_{ins}\varepsilon_0 \mu_n)/D$

Writing $\beta = (KW)/L$, where *W/L* is contributed by the geometry of the device,

$$I_{DS} = \beta\left\{(V_{GS} - V_{th})V_{DS} - V^2_{DS}/2\right\} \quad \text{..... (1.3)}$$

Since, the gate-to-channel capacitance is $C_G = (\varepsilon_{ins}\varepsilon_0 WL)/D$ (parallel plate capacitance), then $K = (C_G \mu_n)/(WL)$

, so that (2.2) may be written as

$$I_{DS} = (C_G \mu_n)/L^2\left\{(V_{GS} - V_{th})V_{DS} - V^2_{DS}/2\right\} \quad \text{.....(1.4)}$$

Denoting $C_G = C_0 WL$ where C_0 : gate capacitance per unit area,

$$I_{DS} = (C_0 \mu_n W)/L^2\left\{(V_{GS} - V_{th})V_{DS} - V^2_{DS}/2\right\} \quad \text{..... (1.5)}$$

Saturated region: Under the voltage condition V_{GS} - V_{th} = V_{DS} , a MOS device is said to be in saturation region of operation. In fact, saturation begins when V_{DS} = V_{GS} - V_{th} , since at this point, the resistive voltage drop (IR drop) in the channel equals the effective gate-to-channel voltage at the drain. One may assume that the current remains *constant* as V_{DS} increases further. Putting V_{DS} = V_{GS} - V_{th} , the equations (2.2-2.5) under saturation condition need to be modified as

$$I_{DS} = (KW)/L\left\{(V_{GS} - V_{th})^2/2\right\} \quad \text{.....(1.6)}$$

$$I_{DS} = \beta(V_{GS} - V_{th})^2/2 \quad \text{.....(1.7)}$$

$$I_{DS} = \left\{C_G \mu_n (V_{GS} - V_{th})^2\right\}/(2L^2) \quad \text{.....(1.8)}$$

$$I_{DS} = \left\{ C_0 \mu_n \left(V_{GS} - V_{tn} \right)^2 \right\} / (2L) \quad \text{.....(1.9)}$$

The expressions in the last slide derived for I_{DS} are valid for both the enhancement and the depletion mode devices. However, the threshold voltage for the nMOS depletion mode devices (generally denoted as V_{td}) is *negative* .

Figure depicts the typical current-voltage characteristics for nMOS enhancement as well as depletion mode transistors. The corresponding curves for a pMOS device may be obtained with appropriate reversal of polarity. For an *n* -channel device with μ_n = 600 cm^2/ V.s, C_0 = 7 X 10^{-8} F/cm^2 , W = 20 m, L = 2 m and $V_{th} = V_{T0}$ = 1.0 V, let us examine the relationship between the drain current and the terminal voltages.

$$K = (C_G \mu_n) / (WL) = (C_0 \mu_n W) / L = 600\,\mathrm{cm}^2/\mathrm{V.s} \times 7 \times 10^{-8}\,\mathrm{F/cm}^2 \times 20\mu\mathrm{m} / 2\,\mu\mathrm{m} = 0.42\,\mathrm{mA/V}^2$$

Now, the current-voltage equation (1.2) can be written as follows.

$$I_{DS} = 0.21\,\mathrm{mA/V}^2 \left\{ 2(V_{GS} - 1.0) V_{DS} - V^2_{DS} \right\}$$

If one plots I_{DS} as a function of V_{DS} , for different (constant) values of V_{GS} , one would obtain a characteristic similar to the one shown in Figure. It may be observed that the second-order current-voltage equation given above gives rise to a set of inverted parabolas for each constant V_{GS} value.

5. Discuss in details about Second Order Effects. [**CO1**-**H1]**

The current-voltage equations in the previous section however are ideal in nature. These have been derived keeping various secondary effects out of consideration.

Threshold voltage and body effect: as has been discussed at length in Sec. 2.1.6, the threshold voltage V_{th} does vary with the voltage difference V_{sb} between the source and the body (substrate). Thus including this difference, the generalized expression for the threshold voltage is reiterated as

$$V_T = V_{T0} + \gamma \left(\sqrt{|2\phi_F + V_{SB}|} - \sqrt{|2\phi_F|} \right) \quad \text{..... (1.10)}$$

in which the parameter γ, known as the *substrate-bias* (or *body-effect*) coefficient is given by

$\gamma = \frac{\sqrt{2qN_A\varepsilon_S}}{C_{ox}}$ .Typical values of γ range from 0.4 to 1.2. It may also be written as

$$\gamma = \frac{t_{ox}}{\varepsilon_{ox}}\sqrt{2qN_A\varepsilon_S} = \frac{1}{C_{ox}}\sqrt{2qN_A\varepsilon_S}$$

Example 2.3:

$N_A = 3\times10^{16}\ \text{cm}^{-3}$, $t_{ox} = 200\overset{0}{A}$, $\varepsilon_{ox} = 11.7\times8.85\times10^{-14}\ F/cm$ and $q = 1.6\times10^{-9}$ coulomb

$$\gamma = \frac{0.2\times10^{-5}}{3.9\times8.85\times10^{-14}}\sqrt{2\times16\times10^{-19}\times11.7\times8.85\times10^{-14}\times3\times10^{16}} = 0.57$$

$$\phi_b = 0.0261\ln\left(\frac{3\times10^{16}}{1.5\times10^{10}}\right) = 0.375$$

Then, at V_{sb} = 2.5 volts

$$V_{T2.5} = V_{T0} + 0.57\left(\sqrt{0.75+2.5} - \sqrt{0.75}\right) = V_{T0} + 0.53$$

As is clear, the threshold voltage increases by almost half a volt for the above process parameters when the source is higher than the substrate by 2.5 volts.

Drain punch-through: In a MOSFET device with improperly scaled small channel length and too low channel doping, undesired electrostatic interaction can take place between the source and the drain known as *drain-induced barrier lowering* (DIBL) takes place. This leads to punch-through leakage or breakdown between the source and the drain, and loss of gate control. One should consider the surface potential along the channel to understand the punch-through phenomenon. As the drain bias increases, the conduction band edge (which represents the electron energies) in the drain is pulled down, leading to an increase in the drain-channel depletion width.

In a long-channel device, the drain bias does not influence the source-to-channel potential barrier, and it depends on the increase of gate bias to cause the drain current to flow. However, in a short-channel device, as a result of increase in drain bias and pull-down of the conduction band edge, the source-channel potential barrier is lowered due to DIBL. This in turn causes drain current to flow regardless of the gate voltage (that is, even if it is below the threshold voltage V_{th}). More simply, the advent of DIBL may be explained by the expansion of drain depletion region and its eventual merging with source depletion region, causing punch-through breakdown between the source and the drain. The punch-through condition puts a natural constraint on the voltages across the internal circuit nodes.

Sub-threshold region conduction: the cutoff region of operation is also referred to as the sub-threshold region, which is mathematically expressed as

$I_{DS} = 0 \qquad V_{GS} < V_{th}$

However, a phenomenon called *sub-threshold conduction* is observed in small-geometry transistors. The current flow in the channel depends on creating and maintaining an inversion layer on the surface. If the gate voltage is inadequate to invert the surface (that is, $V_{GS} < V_{T0}$), the electrons in the channel encounter a *potential barrier* that blocks the flow. However, in small-geometry MOSFETs, this potential barrier is controlled by both V_{GS} and V_{DS} . If the drain voltage is increased, the potential barrier in the channel decreases, leading to *drain-induced barrier lowering* (DIBL). The lowered potential barrier finally leads to flow of electrons between the source and the drain, even if $V_{GS} < V_{T0}$ (that is, even when the surface is not in strong inversion). The channel current flowing in this condition is called the *sub-threshold current* . This current, due mainly to diffusion between the source and the drain, is causing concern in deep sub-micron designs. The model implemented in SPICE brings in an exponential, semi-empirical dependence of the drain current on V_{GS} in the *weak inversion region.* Defining a voltage *V* on as the boundary between the regions of weak and strong inversion,

$$I_D(weak\ inversion) = I_{on} \cdot e^{(V_{GS}-V_{on})\left(\frac{q}{nkT}\right)}$$

where I_{on} is the current in strong inversion for $V_{GS} = V_{on}$.

Channel length modulation: so far one has not considered the variations in channel length due to the changes in drain-to-source voltage V_{DS} . For long-channel transistors, the effect of channel length variation is not prominent. With the decrease in channel lenghth, however, the variation matters. Figure 2.5 shows that the inversion layer reduces to a point at the drain end when

$V_{DS} = V_{DSAT} = V_{GS} - V_{th}$. That is, the channel is *pinched off* at the drain end.

The onset of saturation mode operation is indicated by the pinch-off event. If the drain-to-source voltage is increased beyond the saturation edge ($V_{DS} > V_{DSAT}$), a still larger portion of the channel becomes pinched off. Let the effective channel (that is, the length of the inversion layer) be $L_{eff} = L - \Delta L$ .where *L* : original channel length (the device being in non-saturated mode), and ΔL: length of the channel segment where the inversion layer charge is zero. Thus, the pinch-off point moves from the drain end toward V_{DS} the source with increasing drain-to-source voltage . The remaining portion of the channel between the pinch-off point and the drain end will be in depletion mode. For the shortened channel, with an effective channel voltage of V_{DSAT} , the channel current is given by

$$I_{DS(SAT)}=\frac{\mu_n C_{ox}}{2}.\frac{W}{L_{eff}}.(V_{GS}-V_{T0})^2$$ ..... (1.11)

The current expression pertains to a MOSFET with effective channel length L_{eff}, operating in saturation. The above equation depicts the condition known as *channel length modulation* , where the channel is reduced in length. As the effective length decreases with increasing V_{DS} , the saturation current $I_{DS(SAT)}$ will consequently increase with increasing V_{DS} . The current given by (1.11) can be re-written as

$$I_{DS(SAT)}=\frac{\mu_n C_{ox}}{2}.\left(\frac{1}{1-\frac{\Delta L}{L}}\right)\frac{W}{L}.(V_{GS}-V_{T0})^2$$ ..... (1.12)

The second term on the right hand side of (2.12) accounts for the channel modulation effect. It can be shown that the factor channel length ΔL is expressible as

$$\Delta L \propto \sqrt{V_{DS}-V_{DSAT}}$$

One can even use the empirical relation between ΔL and V_{DS} given as follows.

$$1-\frac{\Delta L}{L} \approx 1-\lambda.V_{DS}$$

The parameter λ is called the *channel length modulation coefficient,* having a value in the range 0.02V -1 to 0.005V -1 . Assuming that $\lambda V_{DS} << 1$, the saturation current given in (1.11) can be written as

$$I_{DS(SAT)}=\frac{\mu_n C_{ox}}{2}.\frac{W}{L_{eff}}.(V_{GS}-V_{T0})^2.(1+\lambda V_{DS})$$

The simplified equation (2.13) points to a linear dependence of the saturation current on the drain-to-source voltage. The slope of the current-voltage characteristic in the saturation region is determined by the channel length modulationfactor λ.

Impact ionization: An electron traveling from the source to the drain along the channel gains kinetic energy at the cost of electrostatic potential energy in the pinch-off region, and becomes a “hot” electron. As the hot electrons travel towards the drain, they can create secondary electron-hole pairs by impact ionization. The secondary electrons are collected at the drain, and cause the drain current in saturation to increase with drain bias at high voltages, thus leading to a fall in the output impedance. The secondary holes are collected as substrate current. This effect is called *impact ionization.* The hot electrons can even penetrate the gate oxide, causing a gate current. This finally leads to degradation in MOSFET parameters like increase of threshold voltage and decrease of transconductance. Impact ionization can create circuit problems such as noise in mixed-

signal systems, poor refresh times in dynamic memories, or latch-up in CMOS circuits. The remedy to this problem is to use a device with lightly doped drain. By reducing the doping density in the source/drain, the depletion width at the reverse-biased drain-channel junction is increase and consequently, the electric filed is reduced. Hot carrier effects do not normally present an acute problem for *p* -channel MOSFETs. This is because the channel mobility of holes is almost half that of the electrons. Thus, for the same filed, there are fewer hot holes than hot electrons. However, lower hole mobility results in lower drive currents in *p* -channel devices than in *n* -channel devices.

6. Describe the CMOS inverter and derive its DC characteristics.[CO1-H1]

Complementary CMOS Inverter - DC Characteristics

A complementary CMOS inverter is implemented as the series connection of a *p*-device and an *n*-device, as shown in Figure 2.10. Note that the source and the substrate (body) of the *p* -device is tied to the V_{DD} rail, while the source and the substrate of the *n*-device are connected to the ground bus. Thus, the devices do not suffer from any body effect. To derive the DC transfer characteristics for the CMOS inverter, which depicts the variation of the output voltage (V_{out}) as a function of the input voltage (V_{in}), one can identify five following regions of operation for the *n* -transistor and *p* -transistor.

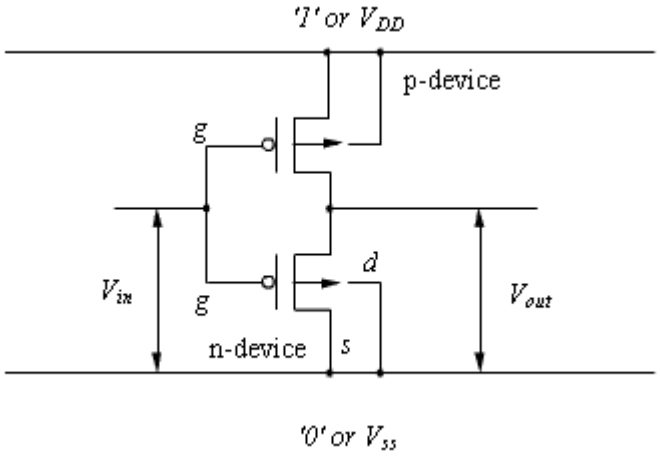


Figure 2.10 A CMOS inverter shown with substrate Connections

Let V_{tn} and V_{tp} denote the threshold voltages of the *n* and *p*-devices respectively. The following voltages at the gate and the drain of the two devices (relative to their respective sources) are all referred with respect to the ground (or V_{SS}), which is the substrate voltage of the *n* -device, namely

$V_{gsn} = V_{in}$, $V_{dsn} = V_{out}$, $V_{gsp} = V_{in} - V_{DD}$, and $V_{dsp} = V_{out} - V_{DD}$.

The voltage transfer characteristic of the CMOS inverter is now derived with reference to the following five regions of operation :

Region 1 : the input voltage is in the range $0 \le V_{in} < V_{tn}$. In this condition, the *n* -transistor is off, while the *p* -transistor is in linear region (as $-V_{DD} < V_{gsp} < -V_{DD} + V_{tn}$).

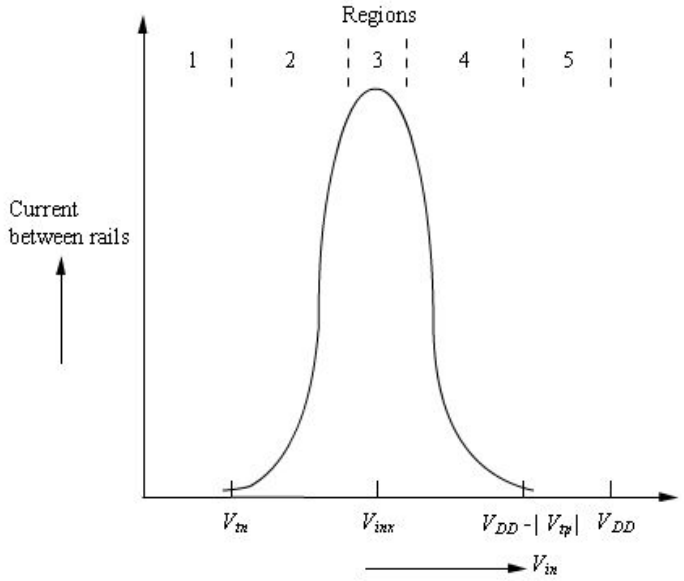


Figure 1.11: Variation of current in CMOS inverter with V_{in}

No actual current flows until V_{in} crosses V_{tn} , as may be seen from Figure 2.11. The operating point of the *p* -transistor moves from higher to lower values of currents in linear zone. The output voltage is given by $V_{out} \approx V_{DD}$, as may be seen from Figure

Region 2 : the input voltage is in the range $V_{tn} \le V_{in} < V_{inv}$. The upper limit of V_{in} is V_{inv} , the *logic threshold voltage* of the inverter. The logic threshold voltage or the *switching point voltage* of an inverter denotes the boundary of "logic 1" and "logic 0". It is the output voltage at which $V_{in} = V_{out}$. In this region, the *n*-transistor moves into saturation, while the *p*-transistor remains in linear region. The total current through the inverter increases, and the output voltage tends to drop fast.

Region 3 : In this region, $V_{in} \approx V_{inv}$. Both the transistors are in saturation, the drain current attains a maximum value, and the output voltage falls rapidly. The inverter exhibits gain. But this region is inherently unstable. As both the transistors are in saturation, equating their currents, one gets (as $V_{gsn} = V_{inv}, \quad V_{gsp} = V_{inv} - V_{DD}$).

$$\frac{1}{2}\beta_n \left(V_{inv} - V_{tn}\right)^2 = \frac{1}{2}\beta_p \left(V_{inv} - V_{DD} - V_{tp}\right)^2 \quad \text{.....(2.14)}$$

where $\beta = K\frac{W}{L}$ and $K = \frac{\varepsilon_{ins}\varepsilon_0\mu}{D}$. Solving for the logic threshold voltage V_{inv} , one gets

$$V_{inv} = \frac{V_{DD} + V_{tp} + V_{tn}\left(\frac{\beta_n}{\beta_p}\right)^{1/2}}{1+\left(\frac{\beta_n}{\beta_p}\right)^{1/2}} \quad \text{.....(2.15)}$$

Note that if $\beta_n = \beta_p$ and $V_{tn} = -V_{tp}$, then V_{inv} =0.5 V_{DD} .

Region 4 : In this region, $V_{inv} < V_{in} \le V_{DD} - |V_{tp}|$. As the input voltage V_{in} is increased beyond V_{inv} , the *n* -transistor leaves saturation region and enters linear region, while the *p* -transistor continues in saturation. The magnitude of both the drain current and the output voltage drops.

Region 5 : In this region, $V_{DD} - |V_{tp}| \le V_{in} \le V_{DD}$. At this point, the *p* -transistor is turned off, and the *n* -transistor is in linear region, drawing a small current, which falls to zero as V_{in} increases beyond V_{DD} -| V_{tp}|, since the *p* -transistor turns off the current path. The output in this region is $V_{out} \approx 0$.

As may be seen from the transfer curve in Figure 2.12, the transition from "logic 1" state (represented by regions 1 and 2) to “logic 0” state (represented by regions 4 and 5) is quite steep. This characteristic guarantees maximum noise immunity.

β_n / β_p **ratio:** One can explore the variation of the transfer characteristic as a function of the ratio β_n / β_p. As noted from (2.15), the logic threshold voltage V_{inv} depends on the ratio β_n / β_p. The CMOS inverter with the ratio β_n / β_p=1 allows a capacitive load to charge and discharge in equal times by providing equal current-source and current-sink capabilities. Consider the case of β_n / β_p>1. Keeping β_p fixed, if one increases β_n, then the impedance of the pull-down *n* -transistor decreases. It conducts faster, leading to faster discharge of the capacitive load. This ensures quicker fall of the output voltage V_{out} , as V_{in} increases from 0 volt onwards. That is, the transfer characteristic shifts leftwards. Similarly, for a CMOS inverter with β_n / β_p<1, the transfer curve shifts rightwards.

7. With neat diagram explain the n-well and p well channel formation in CMOS process. **[CO1-H1]**

CMOS TECHNOLOGIES

CMOS provides an inherently low power static circuit technology that has the capability of providing a lower-delay product than comparable design-rule nMOS or pMOS technologies. The four dominant CMOS technologies are:

- P-well process
- n-well process
- twin-tub process

- Silicon on chip process

The p-well process

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

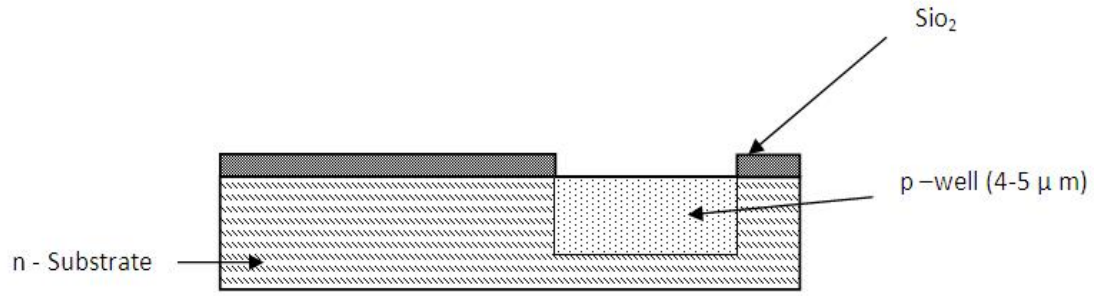
The first mask defines the p-well (p-tub) n-channel transistors (Fig. 1.5a) will be fabricated in this well. Field oxide (FOX) is etched away to allow a deep diffusion.
The next mask is called the “thin oxide” or “thinox” mask (Fig. 1.5b), as it defines where areas of thin oxide are needed to implement transistor gates and allow implantation to form p-or n- type diffusions for transistor source/drain regions. The field oxide areas are etched to the silicon surface and then the thin oxide areas is grown on these areas. O ther terms for this mask include active area, island, and mesa.

Polysilicon gate definition is then completed. This involves covering the surface with polysilicon (Fig 1.5c) and then etching the required pattern (in this case an inverted “U”). “Poly” gate regions lead to “self-aligned” source-drain regions.

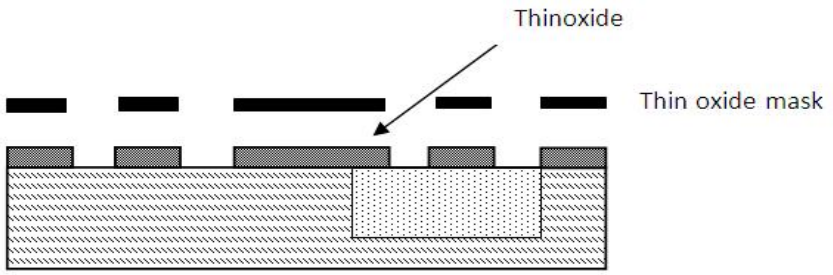
A p-plus (p^+) mask is then used to indicate those thin-oxide areas (and polysilicon) that are to be implanted p^+. Hence a thin-oxide area exposed by the p-plus mask (Fig. 1.5d) will become a p^+ diffusion area. If the p-plus area is in the n-substrate, then a p-channel transistor or p-type wire may be constructed. If the p-plus area is in the p-well, then an ohmic contact to the p-well may be constructed.

The next step usually uses the complement of the p-plus mask, although an extra mask is normally not needed. The “absence” of a p-plus region over a thin-oxide area indicates that the area will be an n^+ diffusion or n-thinox. n-thinox in the p-well defines possible transistors and wires. An n^+ diffusion (Fig. 1.5e) in the n-substrate allows an ohmic contact to be made. Following this step, the surface of the chip is covered with a layer of Sio2.

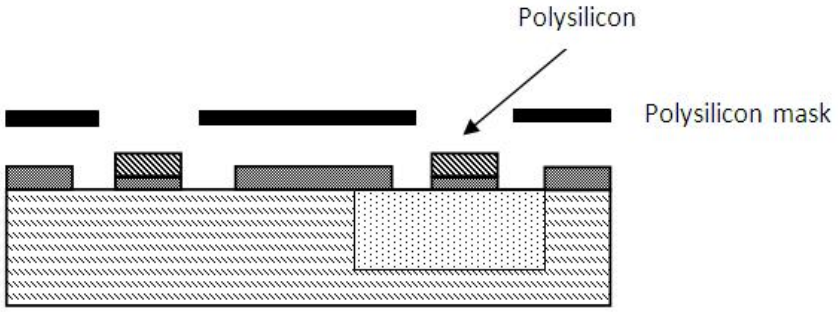
Contacts cuts are then defined. This involves etching any Sio_2 down to the contacted surface, these allow metal (Fig. 1.5f) to contact diffusion regions or polysilicon regions.
Metallization (Fig. 1.5g) is then applied to the surface and selectively etched. As a final step, the wafer is passivated and openings to the bond pads are etched to allow for wire bonding. Passivation protects the silicon surface against the ingress of contaminants.


Sio2
p –well (4-5 µ m)
n - Substrate

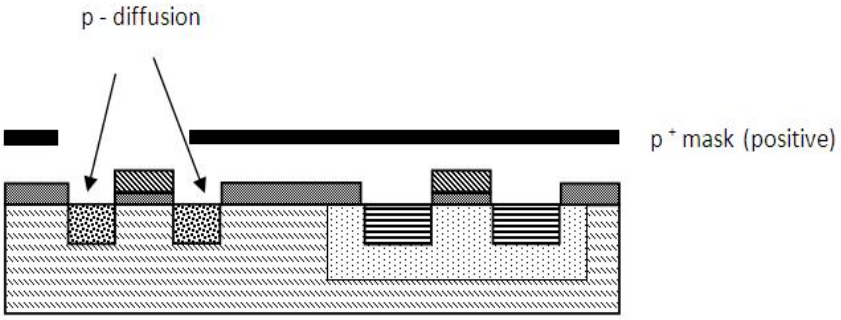
(a)


Thinoxide
Thin oxide mask

(b)


Polysilicon
Polysilicon mask

(c)


p - diffusion
p+ mask (positive)

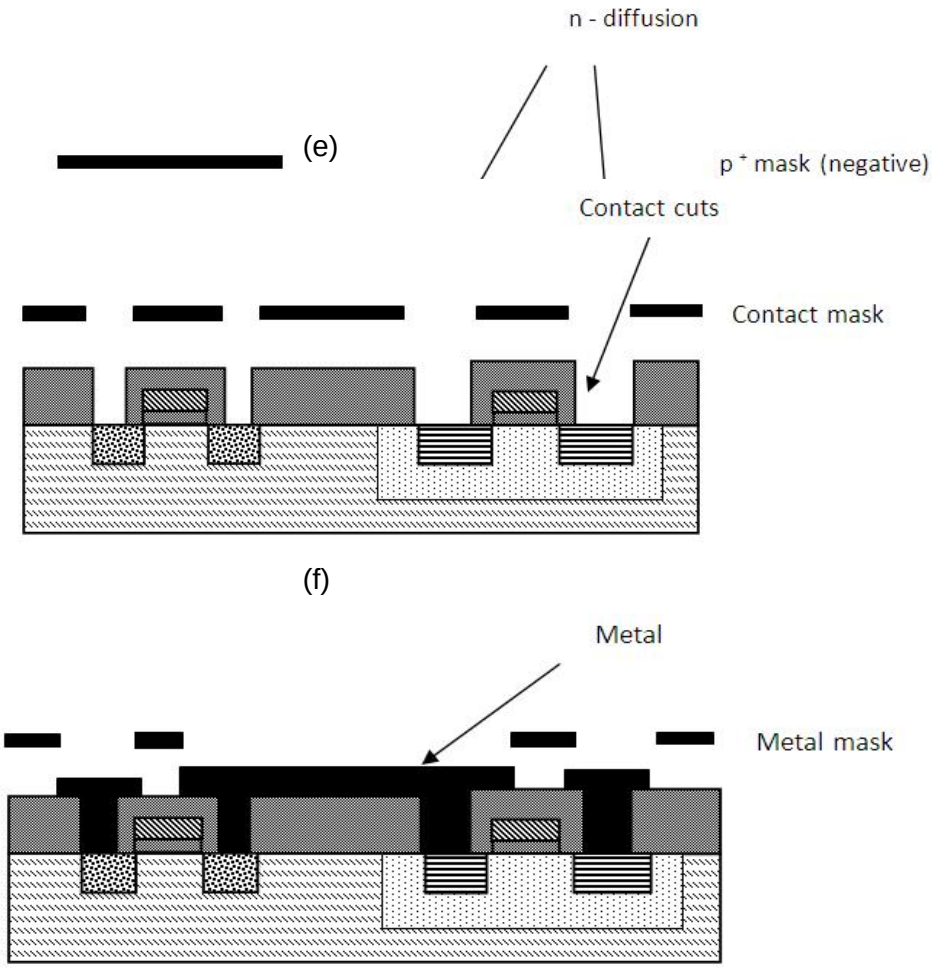


Figure 1.5 Typical p-well CMOS process steps with corresponding masks required

Basically the structure consists of an n-type substrate in which p-devices may be formed by suitable masking and diffusion and, in order to accommodate n-type devices, a deep p-well is diffused into the n-type substrate. This diffusion must be carried out with special care since the p-well doping concentration and depth will affect the threshold voltages as well as the breakdown voltages of the n-transistors. To achieve low threshold voltage (0.6 to 1.0 V), deep well diffusion or high well resistivity is needed. However, deep wells require larger spacing between the n- and p-type transistors and wires because of lateral diffusion resulting in larger chip areas.

High resistivity can accentuate latch-up problems. In order to achieve narrow threshold voltage tolerances in a typical p-well process, the well concentration is made about one order of magnitude higher than the substrate doping density, thereby causing the body effect for n-channel devices to be higher than for p-channel transistors. In addition, due to this higher concentration, n-transistors suffer from excessive source/drain to p-well capacitance will tends to be slower in performance. The well must be grounded in such a way as to minimize any voltage drop due to injected current in substrate that is collected by the p-well.

The p-well act as substrate for then-devices within the parent n-substrate, and, provided polarity restrictions are observed, the two areas are electrically isolated such that there are in affect two substrate, two substrate connections (V_{DD} and V_{SS}) are required.

The n-well process:

The p-well processes have been one of the most commonly available forms of CMOS. However, an advantage of the n-well process is that it can be fabricated on the same process line as conventional n MOS. n –well CMOS circuits are also superior to p-well because of the lower substrate bias effects on transistor threshold voltage and inherently lower parasitic capacitances associated with source and drain regions.

Typically n-well fabrication steps are similar to a p-well process, except that an n-well is used which is illustrated in Fig. 1.6. The first masking step defines the n-well regions. This followed by a low phosphorus implant driven in by a high temperature diffusion step to form the n-wells. The well depth is optimized to ensure against p-substrate to p^+ diffusion breakdown without compromising the n-well to n^+ mask separation. The next steps are to define the devices and diffusion paths, grow field oxide, deposit and pattern the polysilicon, carry out the diffusions, make contact cuts and metallization. An n-well mask is used to define n-well regions, as opposed to a p-well mask in a p-well process. An n-plus (n^+) mask may be used to define the n-channel transistors and V_{DD} contacts. Alternatively, we could use a p-plus mask to define the p-channel transistors, as the masks usually are the complement of each other.

Due to differences in mobility of charge carriers the n-well process creates non-optimum p-channel characteristics, such as high junction capacitance and high body effect. The n-well technology has a distinct advantage of providing optimum device characteristics. Thus n-channel devices may be used to form logic elements to provide speed and density, while p-transistors could primarily serve as pull-up devices.

Unit -II

Combinational Logic Circuits

Part –A

1. What are the general properties of Elmore delay model? [**CO2**-**L1]**

General property of Elmore delay model network has
Single input node All the capacitors are between a node and ground Network does not contain any resistive loop

2. What are the types of power dissipation? [**CO2**-**L1]**

Static power dissipation (due to leakage current when the circuit is idle).
Dynamic power dissipation (when the circuit is switching) and
Short –circuit power dissipation during switching of transistors.

3. List the factor of static power dissipation? [**CO2**-**L1]**

Power dissipation due to leakage current when the idle is called the static power dissipation.
Static power due to

Sub – threshold conduction through OFF transistors
Tunneling current through gate oxide
Leakage through reverse biased diodes
Contention current in radioed circuits.

4. Give the methods to reduce dynamic power dissipation? [**CO2**-**L2]**

1. Reducing the product of capacitance and its switching frequency.
2. Eliminate logic switching that is not necessary for computation.
3. Reduce activity factor Reduce supply voltage

5. Give the methods to reduce static power dissipation? [**CO2**-**L2]**

1. By selecting multi threshold voltages on circuit paths with low-Vt transistors while leakage on other paths with high-Vt transistors.
2. By using two operating modes, active and standby for each function blocks.
3. By adjusting the body bias (i.e) adjusting FBB (Forward Body Bias) in active mode to increase performance and RBB (Reverse Body Bias) in standby mode to reduce leakage.
4. By using sleep transistors to isolate the supply from the block to achieve significant leakage power savings.

6. What is short circuit power dissipation? [**CO2**-**L1]**

During switching, both NMOS and PMOS transistors will conduct simultaneously and provide a direct path between Vdd and the ground rail resulting in short circuit power dissipation

7. Define design margin? [**CO2**-**L1]**

The additional performance capability above required standard basic system parameters that may be specified by a system designer to compensate for uncertainties is called design margin. Design margin required as there are three sources of variation-two environmental and one manufacturing.

8. Write the applications of transmission gate? [**CO2**-**L2]**

Multiplexing element of path selector
A latch element An unlock switch
Act as a voltage controlled resistor connecting the input and output.

9. What is pass transistor? [**CO2**-**L1]**

It is a MOS transistor, in which gate is driven by a control signal the source (out), the drain of the transistor is called constant or variable voltage potential(in) when the control signal is high, input is passed to the output and when the control signal is low, the output is floatingtopology such topology circuits is called pass transistor.

10. List the advantages of pass transistor? [**CO2**-**L2]**

Pass transistor logic (PTL) circuits are often superior to standard CMOS circuits in termsof layout density, circuit delay and power consumption. They do not have path V_{DD} to GND and do not dissipate standby power (static power dissipation).

11. What is transmission gate? [**CO2**-**L1]**

The circuit constructed with the parallel connection of PMOS and NMOS with shorted drain and source terminals. The gate terminal uses two select signals s and s, when s is high than the transmission gates passes the signal on the input. The main advantage of transmission gate is that it eliminates the threshold voltage drop.

12. Why low power has become an important issue in the present day VLSI circuit realization? [**CO2**-**L2]**

In deep submicron technology the power has become as one of the most important issue Because of Increasing transistor count; the number of transistor is getting doubled in every 18 months based on moore's law higher speed of operation; the power dissipation is proportional to clock frequency greater device leakage current; in nanometer technology the leakage component become a significant percentage of the total ower and the leakage current increases at a faster rate then dynamic power in technology generations.

13. Mention the various ways to reduce the delay time of a CMOS inverter? [**CO2**-**L2]**

Various ways for reducing the delay time are given below:

a) The width of the MOS transistor can be increased to reduce delay. this is known as gate sizing, which will be discussed later in more details.

b) The load capacitance can be reduced to reduce delay. this is achived by using transistor ofsmaller and smaller dimension by feature generation technology.

c) Delay can also be reduced by increasing the supply voltage Vdd and/or reducing the threshold voltage Vt of the MOS transistors

14. Explain the basic operation of a 2- phase dynamic circuit? [**CO2**-**L2]**

The operation of the circuit can be explained using precharge logic in which the output is precharged to HIGH level during Φ2 clock and the output is evaluated during Φ1 clock.

15. What makes dynamic CMOS circuits faster than static CMOS circuits? [**CO2**-**L1]**

As MOS dynamic circuits require lesser number of transistors and capacitance is to be driven by it. This makes MOS dynamic circuits faster.

16. What is glitching power dissipation? [**CO2**-**L1]**

Because of finite delay of the gates used to realize boolean functions,different signals cannot reach the inputs of a gate simultaneously. This leads to spurious transition at the output before it settles down to its final value. the spurious transitions leads to charging and discharging of the outputs causing glitching power dissipation. It can be minimized by having balanced realization having same delay at the inputs.

17. List various sources of leakage currents? [**CO2**-**L1]**

Various source of leakage currents are listened below:

I1=Reverse-bias p-n junction diode leakage current.
I2=band-to-band tunneling current

I3=Subthreshold leakage current

I4=Gate oxide tunneling current

I5=Gate current due to hot carrier junction

I6=Channel punch through

I7=Gate induced drain leakage current

18. Compare and contrast clock gating versus power gating approaches.[**CO2**-**L1]**

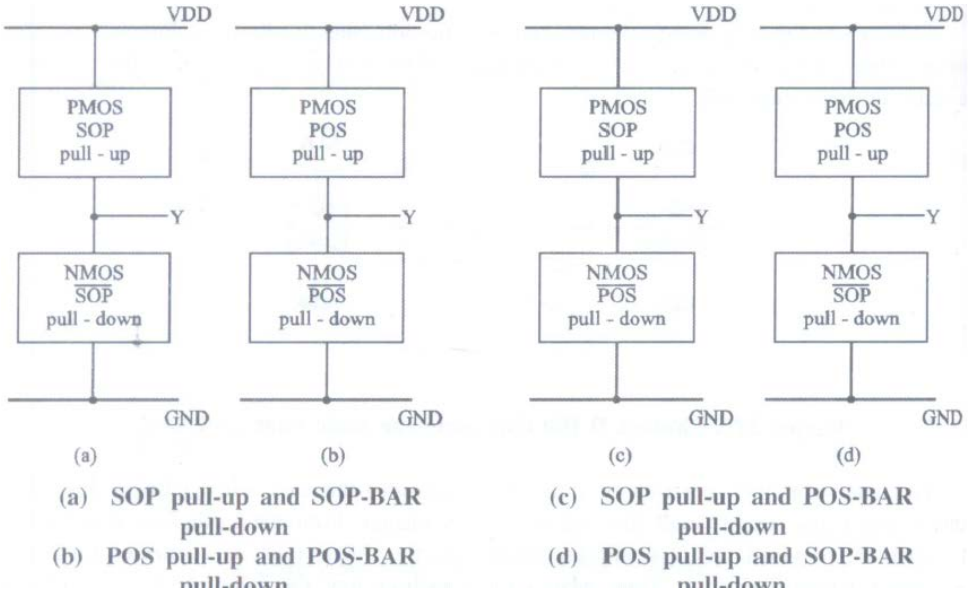
Clock gating minimizes dynamic power by stopping unnecessary transitions, but power gating minimizes leakage power by inserting a high Vt transistor in series with low Vt logic blocks.

Part – B

1. Discuss in detail about the ratioed circuit and dynamic circuit CMOS logic configurations. [**CO2-H1**]

DESIGN PRINCIPLE OF STATIC CMOS DESIGN

Digital CMOS circuits are implemented using either static or dynamic design techniques. In static CMOS, the output is tied to VDD or ground via a low resistance path (except during switching) and this leads to circuits implementation robust with good noise immunity. In static CMOS design any function can be realized as a sum of product (SOP) or a product of sum (POS). If an SOP function pulls the output high, then an SOP-BAR function will pull the output low. A POS function can pull the output high, while a POS-BAR function can pull the output low, as shown in



(a) SOP pull-up and SOP-BAR pull-down
(b) POS pull-up and POS-BAR pull-down
(c) SOP pull-up and POS-BAR pull-down
(d) POS pull-up and SOP-BAR pull-down

Important properties of static CMOS design:

At any instant of time, the output of the gate is directly connected to Vss or VDD.

All functions are composed of either AND'ed or OR'ed sub functions. The AND function is composed of NMOS transistors in series. The OR function is composed of NMOS transistors in parallel.

Contains a pull-up network (PUP) and pull down network (PDN).

PUP networks consist of PMOS transistors.

PDN networks consist of NMOS transistors.

Each network is the dual of the other network.

The output of the complementary gate is inverted.

Advantages of static CMOS design

Robust in construction.

Good noise immunity.

Static logic has no minimum clock rate, the clock can be paused indefinitely.

low power consumption.

For low operating frequencies, CMOS static logic is used to obtain a relatively small die size.

Limitations of static CMOS design

The main limitation of static circuits is slower-speed as compared to dynamic circuits. The reasons are

1. Increased gate capacitance due to the presence of both PMOS and NMOS transistors.

2. Output depends on the previous cycle inputs due to charges that may be present at internal inputs.

Multiple switching of the output within a cycle depending on the input switching pattern

MOSFETS as Switches

The gate controls the passage of current between the source and the drain. CMOS uses positive logic - VDD is logic T and Vss is logic '0'. We turn a transistor on or off using the gate terminal. There are two kinds of CMOS transistors, n - Channel transistors and p -channel transistors. An n - channel transistor requires a logic T on the gate to make the switch conducting (to turn the transistor on). A p - channel transistor requires a logic '0' on the gate to make the switch conducting (to turn the transistor on). The conventional schematic icon representation along with the switch characteristics is shown

Basic CMOS Gates In this section, the basic gate implementation in static CMOS are presented.

AND Gate

If two N-switches are placed in series, the composite switch constructed by this action is closed (or ON) if both switches are connected to logic '1'. If any one of the switch is at logic '0' the circuit is said to be open (or OFF) state this yields an 'AND' function. The switch logic of AND function is shown in

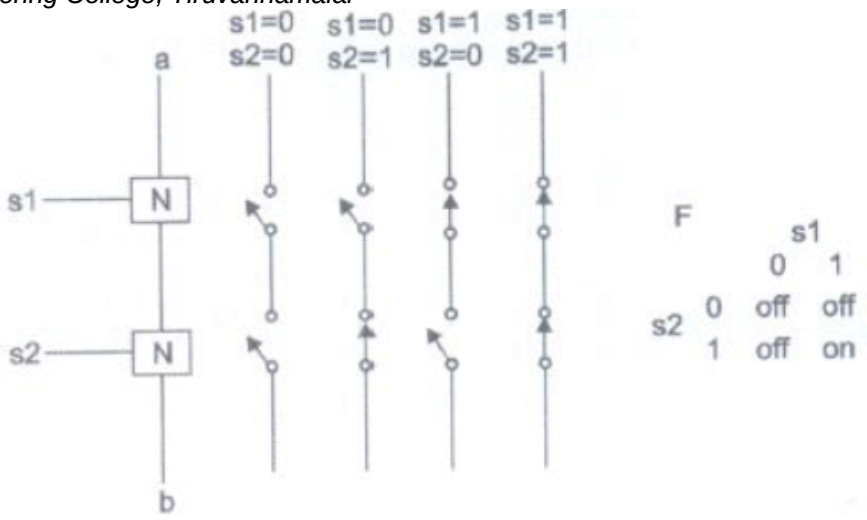


Figure 2.55 AND Gate

OR Gate

If two N-switches are placed in parallel, the composite switch constructed by this action is closed (or ON) if any one of the switch is connected to logic T . If both the switches are

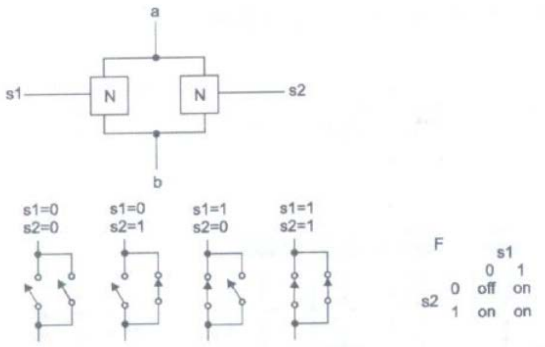
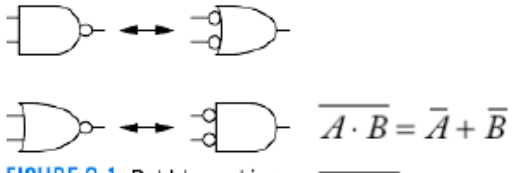


Figure 2.56 OR Gate

Bubble Pushing

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



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

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

FIGURE 9.1 Bubble pushing with DeMorgan's law

A NAND gate is equivalent to an OR of inverted inputs. A NOR gate is equivalent to an AND of inverted inputs. The same relationship applies to gates with more inputs.

Switching between these representations is easy to do on a whiteboard and is often called bubble pushing**.**

Compound Gates:

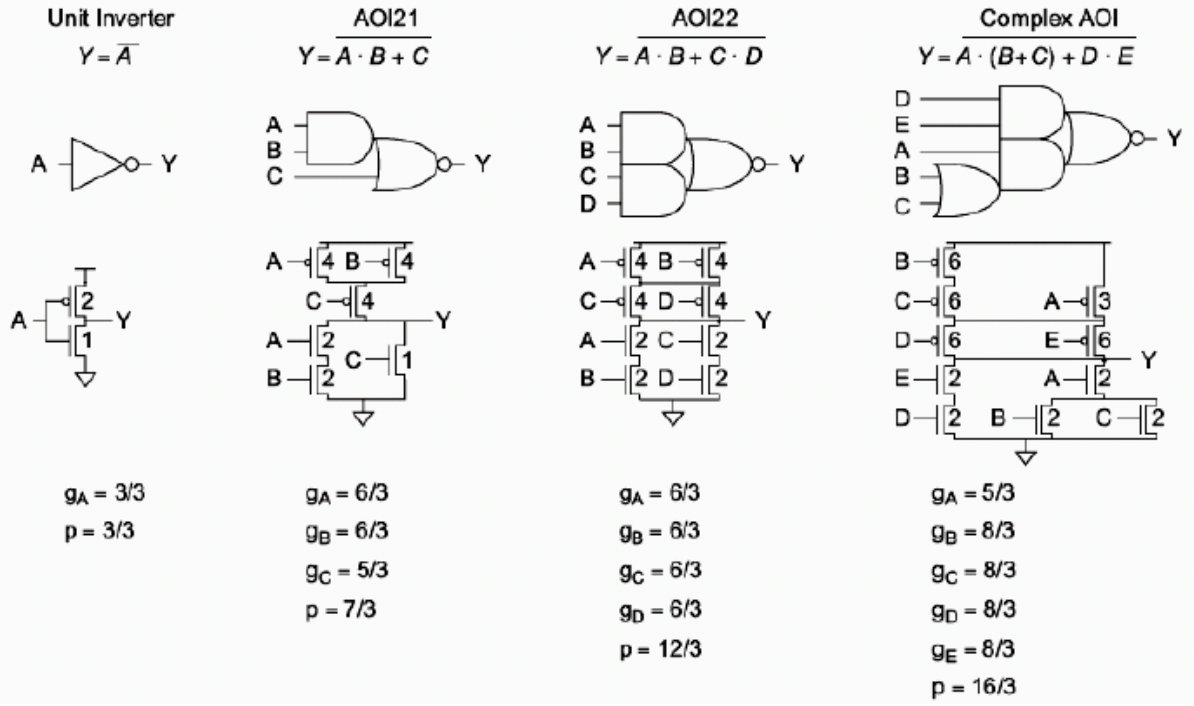
Static CMOS also efficiently handles compound gates computing various inverting combinations of AND/OR functions in a single stage.

The function F = AB + CD can be computed with an AND-OR-INVERT- 22 (AOI22) gate and an general, logical effort of compound gates can be different for different inputs. Figure 9.4 shows how logical efforts can be estimated for the AOI21, AOI22, and a more complex compound AOI gate. The transistor widths are chosen to give the same drive as a unit inverter.

The logical effort of each input is the ratio of the input capacitance of that input to the input capacitance of the inverter

For the AOI21 gate, this means the logical effort is slightly lower for the OR terminal (C) than for the two AND terminals (A, B).

The parasitic delay is crudely estimated from the total diffusion capacitance on the output node by summing the sizes of the transistors attached to the output.



Input Ordering Delay Effect

The logical effort and parasitic delay of different gate inputs are often different. Some logic capacitance than another.

Other gates, like NANDs and NORs, are nominally symmetric but actually have slightly different logical effort and parasitic delays for the different inputs.

Figure 9.6 shows a 2-input NAND gate annotated with diffusion parasitic. Consider the falling output transition occurring when one input held a stable 1 value and the other

rises from 0 to 1. If input B rises last, node x will initially be at VDD – Vt ≈ VDD because it was pulled up through the nMOS transistor on input A.

□□ The Elmore delay is (R/2)(2C) + R(6C) = 7RC = 2.33 ζ1 On the other hand, if input A rises last, node x will initially be at 0 V because it was discharged through the nMOS transistor on input B. No charge must be delivered to node x, so the Elmore delay is simply R(6C) = 6RC = 2 ζ.
□□ In general, we define the outer input to be the input closer to the supply rail (e.g., B) and the inner input to be the input closer to the output (e.g., A).

□□ The parasitic delay is smallest when the inner input switches last because the intermediate nodes have already been discharged. Therefore, if one signal is known to arrive later than the others, the gate is fastest when that signal is connected to the inner input.

□□ The inner input has a lower parasitic delay. The logical efforts are lower than initial estimates might predict because of velocity saturation. Interestingly, the inner input has a slightly higher logical effort because the intermediate node x tends to rise and cause negative feedback when the inner input turns ON.

□□ This effect is seldom significant to the designer because the inner input remains faster over the range of fanouts used in reasonable circuits.

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

o For example, consider the path in Figure 9.7(a). Under ordinary conditions, the path acts as a buffer between A and Y. When reset is asserted, the path forces the output low. If reset only occurs under exceptional circumstances and can take place slowly, the circuit should be optimized for input-to-output delay at the expense of reset. This can be done with the asymmetric NAND gate in Figure 9.7(b).

The pulldown resistance is R/4 +R/ (4/3) = R, so the gate still offers the same driver as a unit inverter. However, the capacitance on input A is only 10/3, so the logical effort is 10/9. This is better than 4/3, which is normally associated with a NAND gate. In the limit of an infinitely large reset transistor and unit-sized nMOS transistor for input A, the logical effort approaches 1, just like an inverter.

The improvement in logical effort of input A comes at the cost of much higher effort on the reset input. Note that the pMOS transistor on the reset input is also shrunk. This reduces its diffusion capacitance and parasitic delay at the expense of slower response to reset.

Skewed Gates

□□ In other cases, one input transition is more important than the other. we defined H-L skew gates to favor the rising output transition and LO-skew gates to favor the falling output transition. This favoring can be done by decreasing the size of the noncritical transistor.

The logical efforts for the rising (up) and falling (down) transitions are called gu and gd, respectively, and are the ratio of the input capacitance of the skewed gate to the input capacitance of an unskewed inverter with equal drive for that transition.

Figure 9.9(a) shows how a HIskew inverter is constructed by downsizing the nMOS transistor. This maintains the same effective resistance for the critical transition while reducing the input capacitance relative to the unskewed inverter of Figure 9.9(b), thus reducing the logical effort on that critical transition to gu = 2.5/3 = 5/6.

Of course, the improvement comes at the expense of the effort on the noncritical transition. The logical effort for the falling transition is estimated by comparing the inverter to a smaller unskewed inverter with equal pulldown current, shown in Figure 9.9(c), giving a logical effort of gd = 2.5/1.5 = 5/3.

The degree of skewing (e.g., the ratio of effective resistance for the fast transition relative to the slow transition) impacts the logical efforts and noise margins; a factor of two is common. Figure 9.10 catalogs HIskew and LO-skew gates with a skew factor of two. Skewed gates are sometimes denoted with an H or an L on their symbol in a schematic.

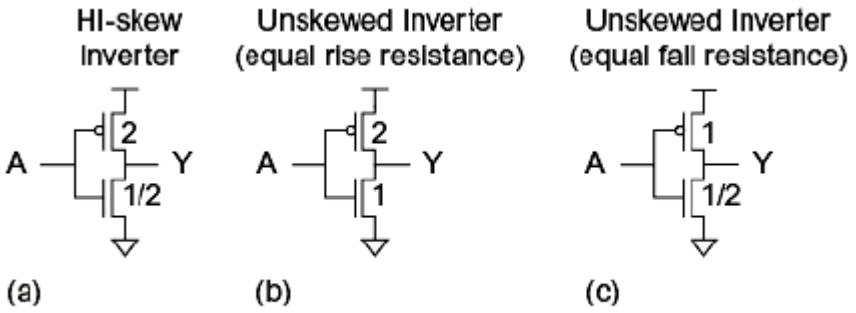
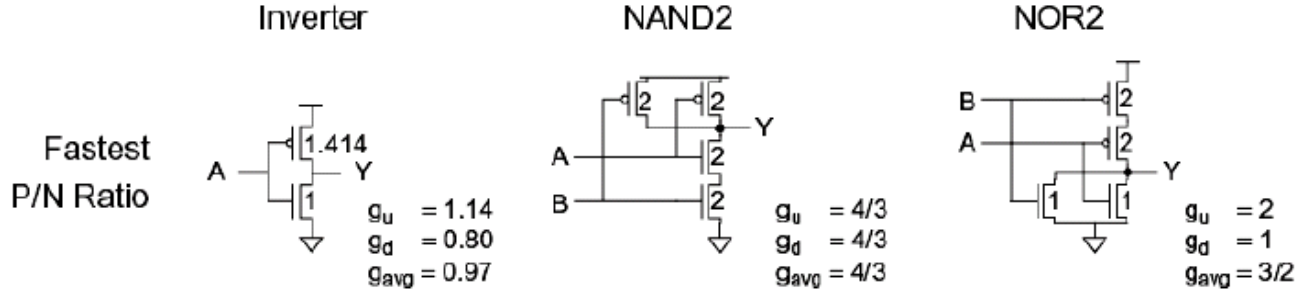


FIGURE 9.9 Logical effort calculation for HI-skew inverter

P/N Ratios

The pMOS transistors in the unskewed gate are enormous in order to provide equal rise delay. They contribute input capacitance for both transitions, while only helping the rising delay. By accepting a slower rise delay, the pMOS transistors can be downsized to reduce input capacitance and average delay significantly.



Reducing the pMOS size from 2 to for the inverter gives the theoretical fastest average delay, but this delay improvement is only 3%. However, this significantly reduces the pMOS transistor area.

It also reduces input capacitance, which in turn reduces power consumption. Unfortunately, it leads to unequal delay between the outputs. Some paths can be slower than average if they trigger the worst edge of each gate.

- Excessively slow rising outputs can also cause hot electron degradation. And reducing the pMOS size also moves the switching point lower and reduces the inverter's noise margin. In summary, the P/N ratio of a library of cells should be chosen on the basis of area, power, and reliability, not average delay.

- For NOR gates, reducing the size of the pMOS transistors significantly improves both delay and area. In most standard cell libraries, the pitch of the cell determines the P/N ratio that can be achieved in any particular gate. Ratios of 1.5–2 are commonly used for inverters.

Multiple Threshold Voltages

- Some CMOS processes offer two or more threshold voltages. Transistors with lower threshold voltages produce more ON current, but also leak exponentially more OFF current.

- Libraries can provide both high- and low-threshold versions of gates. The low-threshold gates can be used sparingly to reduce the delay of critical paths.

- Skewed gates can use low-threshold devices on only the critical network of transistors

2. Describe the basic principle of operation of SFPL and CVSL source follower pull-up logic with neat diagrams. **[CO2-H1]**

THE CHARACTERISTIC OF SFPL & CVSL Source-Follower Pull-up Logic (SFPL)

• SFPL (shown in Figure 2.82) is a variation of pseudo-nMOS whereby the load device is an N pull-down transistor and N source-follower pull-ups are used on the inputs.

• N pull-up transistors can be small limiting input capacitance

• N transistors are also duplicated as pull-down devices in order to improve the fall time

• Rise time is determined by the PI inverter pull-up transistor when all inputs are low

SFPL is useful for high fan-in NOR logic gates

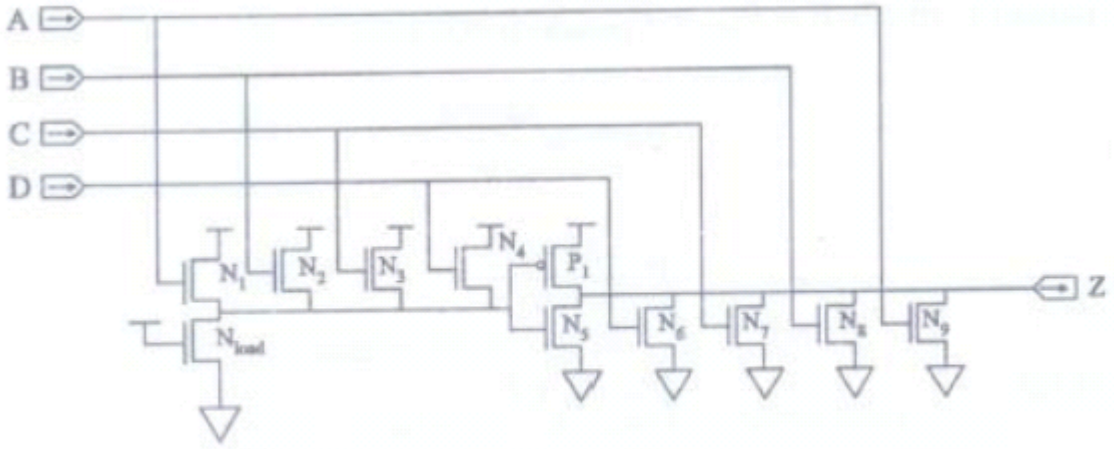


Figure 2.82 Source-Follower Pull-up Logic (SFPL)

Cascade Voltage Switch Logic (CVSL)

Cascade voltage switch logic (CVSL) belongs to class of differential-logic types. The I idea is to use a dual n-block instead of a dual p-block and a pair of cross-coupled pMOS transistors compute the logic function and its complement. CVSL can be roughly as fast as (dynamic logic, it dissipates almost as little static power as static CMOS and is relatively robust against large structure. Also, as a consequence of the differential signals, which means an effective doubling of the voltage swing, CVSL is very robust against noise and against capacitive noise | coupling. On the other hand, the overhead is considerable: twice the transistor count of dynamic logic and pseudo-nMOS, and twice the dynamic power consumption. Nevertheless, CVSL is a good candidate for low-voltage applications, also in combination with other types of logic. Characteristics
CVSL is a differential type of logic circuit whereby both true and complement inputs are required For example, true inputs are applied to left pull-down leg below and complement inputs are applied to right leg

N pull-down trees are the dual of each other

P pull-up devices are cross-coupled to latch output

Both true and complement outputs are obtained

Input pull-down trees may be intermixed, depending on the logic to be implemented

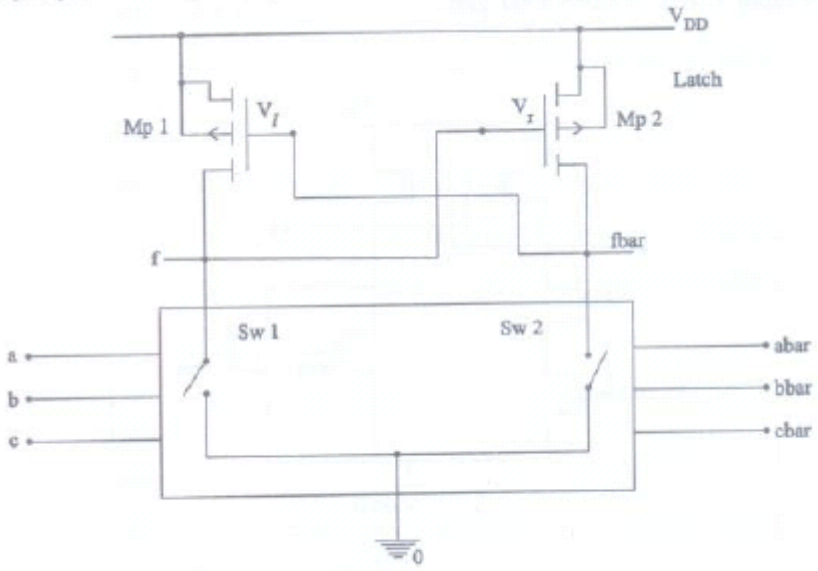


Figure 2.83 Basic strucutre of CVSL

CVSL gates have latching characteristics built into the circuit itself. The output results f and fbar are held until the inputs induce a change. The basic structure of a CVSL gate is shown in Figure 2.83. '• The input set consists of the variables (a , b, c) and their compliments (abar, bbar, cbar) that are routed into an nFET 'logic tree' network.
• The logic tree is modeled as a pair of complimentary switches **Sw1** and **Sw2** such that one is closed while the other is open as determined by the inputs.

• The state of the switches establishes the outputs. For example, if **Sw1** is closed then **f=0.**
• The opposite side (fbar) is forced to the complimentary state (fbar = 1) by the action of the pFET latch .

• The latch is controlled by the left and right source-gate voltages **Vl** and **Vr** shown in Figure 2.83. Suppose that **Sw2** is closed, forcing fbar = 0 on the right side.

• In this case, **Vt=VDD** which turns on Mpl. With Mpl conducting, the left output node sees a path to the power supply, giving **VDD** that is f= 1 state.

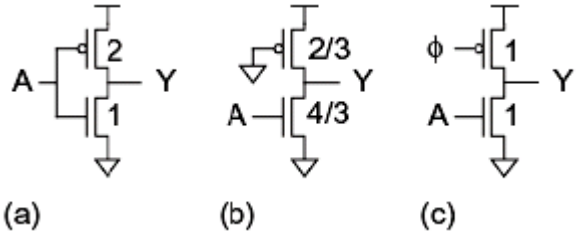
3. Devise the operation of dynamic CMOS, domino, NP domino logic and Dual-Rail Domino Logic with neat diagrams. [CO2-H3]

DYNAMIC CMOS LOGIC

Ratioed circuits reduce the input capacitance by replacing the pMOS transistors connected to the inputs with a single resistive pullup. The drawbacks of ratioed circuits include slow rising transitions, contention on the falling transitions, static power dissipation, and a non zero VOL.

Dynamic circuits circumvent these drawbacks by using a clocked pullup transistor rather than a pMOS that is always ON. Figure 9.21 compares (a) static CMOS, (b) pseudo-nMOS, and (c) dynamic inverters. Dynamic circuit operation is divided into two modes, as shown in Figure

During precharge, the clock Φ 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 pulldown network.



Dynamic circuits are the fastest commonly used circuit family because they have lower input capacitance and no contention during switching. They also have zero static power dissipation. However, they require careful clocking, consume significant dynamic power, and are sensitive to noise during evaluation.

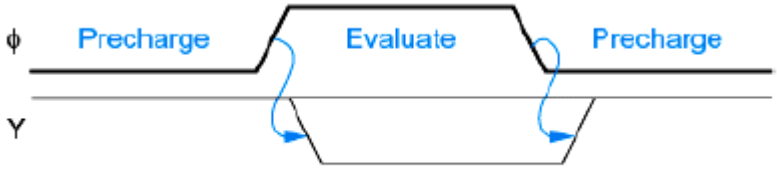


FIGURE 9.22 Precharge and evaluation of dynamic gates

In Figure 9.21(c), if the input A is 1 during precharge, contention will take place because both the pMOS and nMOS transistors will be ON.

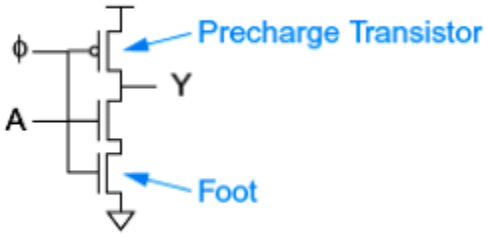


FIGURE 9.23 Footed dynamic inverter

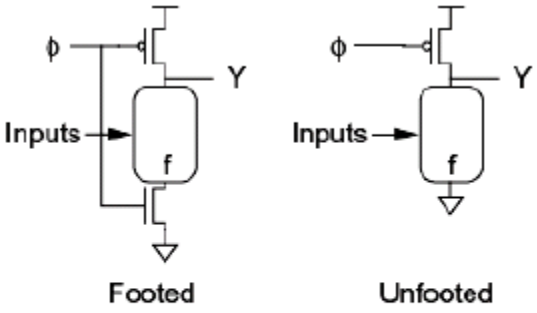


FIGURE 9.24 Generalized footed and unfooted dynamic gates

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 9.25 estimates the falling logical effort of both footed and unfooted dynamic gates. As usual, the pulldown transistors‟ widths are chosen to give unit resistance. Precharge occurs while the gate is idle and often may take place more slowly. Therefore, the precharge transistor width is chosen for twice unit resistance. This

reduces the capacitive load on the clock and the parasitic capacitance at the expense of greater rising delays. We see that the logical efforts are very low.

☐☐ Footed gates have higher logical effort than their unfooted counterparts but are still an improvement over static logic. In practice, the logical effort of footed gates is better than predicted because velocity saturation means series nMOS transistors have less resistance than we have estimated.

☐☐ Moreover, logical efforts are also slightly better than predicted because there is no contention between nMOS and pMOS transistors during the input transition.

The size of the foot can be increased relative to the other nMOS transistors to reduce logical effort of the other inputs at the expense of greater clock loading. Like pseudo-nMOS gates, dynamic gates are particularly well suited to wide NOR functions or multiplexers because the logical effort is independent of the number of inputs.

☐☐ Of course, the parasitic delay does increase with the number of inputs because there is more diffusion capacitance on the output node. Characterizing the logical effort and parasitic delay of dynamic gates is tricky because the output tends to fall much faster than the input rises, leading to potentially misleading dependence of propagation delay on fanout.

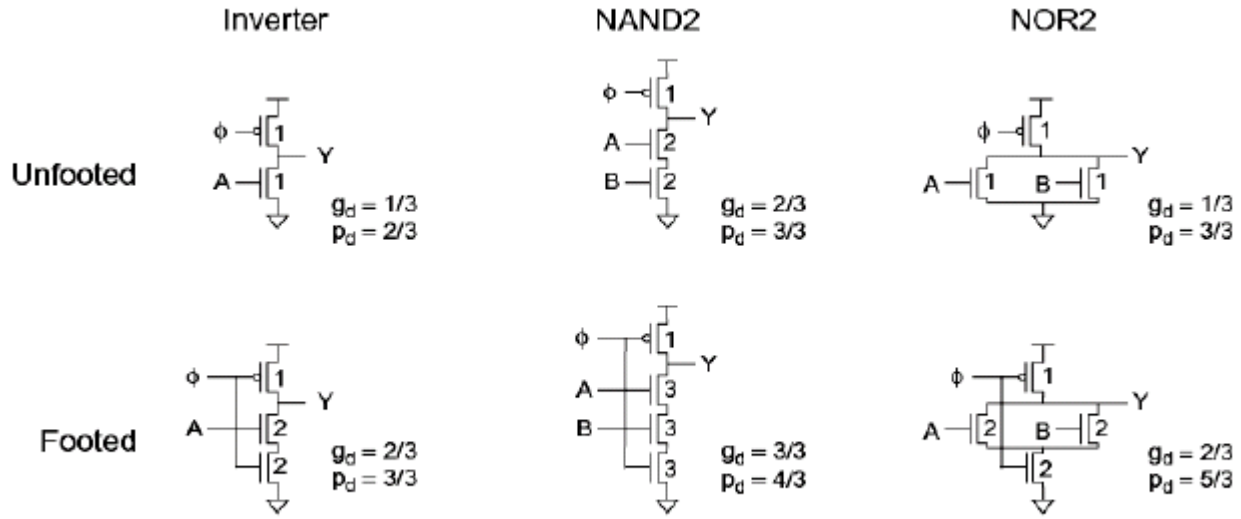
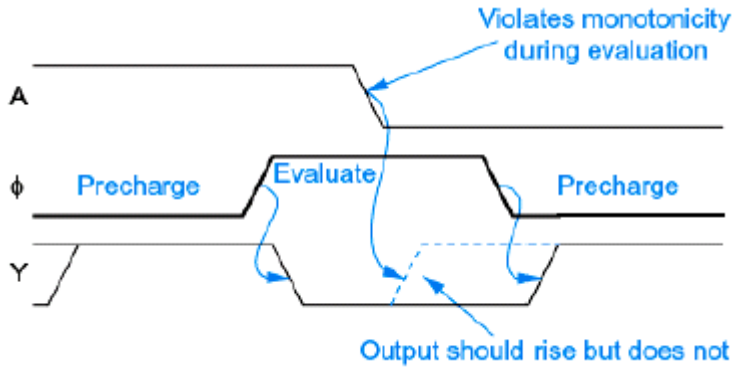


FIGURE 9.25 Catalog of dynamic gates



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 9.26 shows waveforms for a footed dynamic inverter in which the input violates monotonicity. During precharge, the output is pulled HIGH. When the clock rises, the input is HIGH so the output is discharged LOW through the pulldown network, as you would want to have happen in an inverter. The input later falls LOW, turning off the pulldown network.

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.

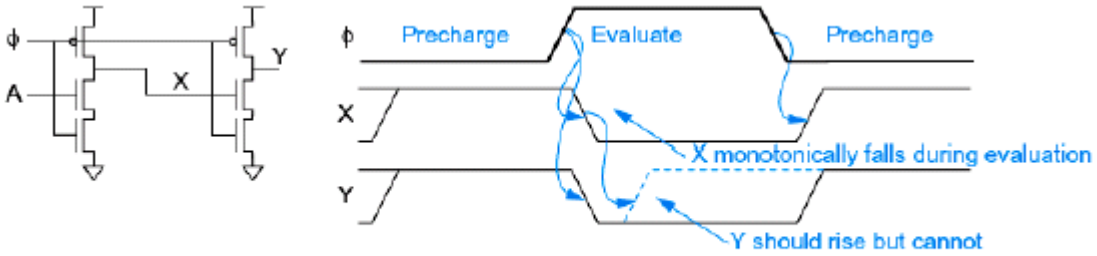


FIGURE 9.27 Incorrect connection of dynamic gates

CMOS 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 dynamicstatic 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.

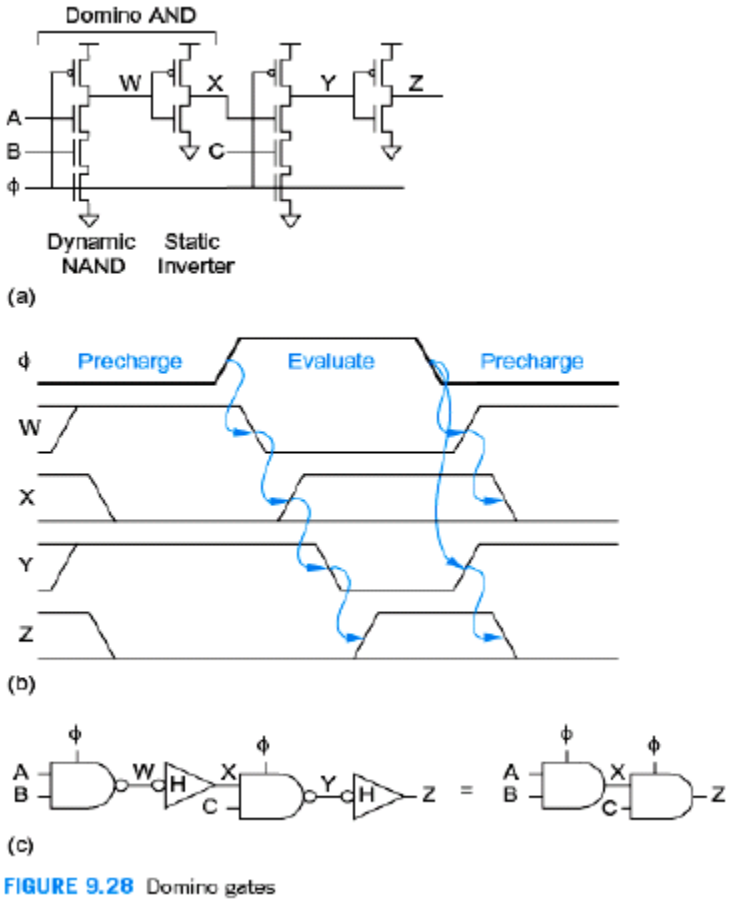


FIGURE 9.28 Domino gates

In general, more complex inverting static CMOS gates such as NANDs or NORs can be used in place of the inverter . This mixture of dynamic and static logic is called compound domino.

Domino gates are inherently noninverting, while some functions like XOR gates necessarily require inversion. Three methods of addressing this problem include pushing inversions into static logic, delaying clocks, and using dual-rail domino logic.

In many circuits including arithmetic logic units (ALUs), the necessary XOR gate at the end of the path can be built with a conventional static CMOS XOR gate driven by the last domino circuit. However, the XOR output no longer is monotonically rising and thus cannot directly drive more domino logic.

A second approach is to directly cascade dynamic gates without the static CMOS inverter, delaying the clock to the later gates to ensure the inputs are monotonic during evaluation.

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 precharged, neither _h nor _l is asserted. The pair of lines should never be both asserted simultaneously during correct operation.

TABLE 9.2 Dual-rail domino signal encoding

sig_h	sig_l	Meaning
0	0	Precharged
0	1	‘0’
1	0	‘1’
1	1	Invalid

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. The gates are shown with clocked evaluation transistors, but can also be unfooted.

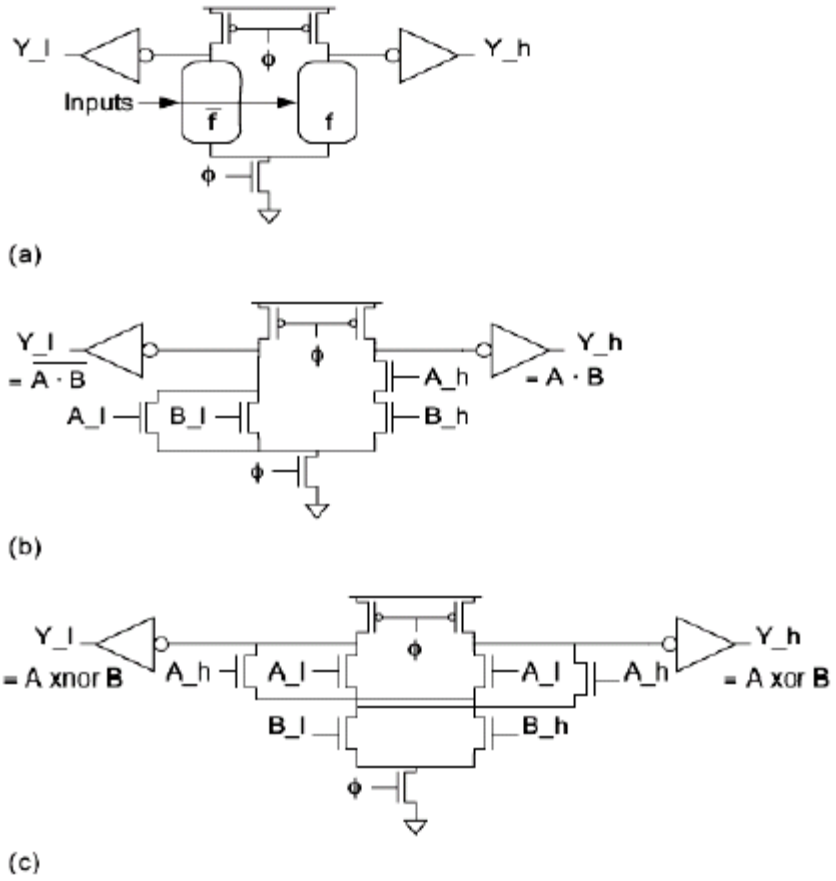


FIGURE 9.30 Dual-rail domino gates

Dual-rail domino is a complete logic family in that it can compute all inverting and noninverting logic functions.

□□ However, it requires more area, wiring, and power. Dual-rail structures also 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 precharged.

□□ When the computation completes, one rail will be asserted. A NAND gate can be used for completion detection, as shown in Figure 9.31. This is particularly useful for asynchronous circuits

4. Devise the principle of operation of keepers, Multiple-Output Domino Logic (MODL) and NP and Zipper Domino with neat diagrams. [CO2-H3]

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 subthreshold, 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.

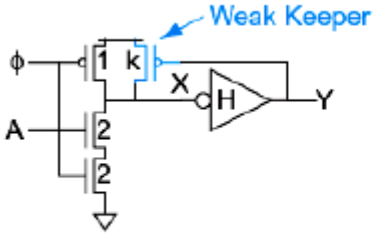


FIGURE 9.33 Conventional keeper

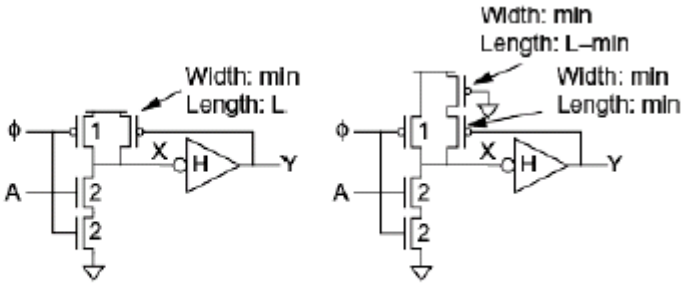


FIGURE 9.34 Weak keeper implementations

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 prevent X from floating.

When X falls, the keeper initially opposes the transition so it must be much weaker than the pulldown network. Eventually Y rises, turning the keeper OFF and avoiding static power dissipation

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

For smalldynamic gates, the keeper must be weaker than a minimum-sized transistor. This is achieved by increasing the keeper length, as shown in Figure 9.34(a).

Long keeper transistors increase the capacitive load on the output Y. This can be avoided by splitting the keeper, as shown in Figure 9.34(b).

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 ci 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 (pi) or generate a carry (gi). The carry-out logic is

$$c_1 = g_1 + p_1 c_0$$
$$c_2 = g_2 + p_2(g_1 + p_1 c_0)$$
$$c_3 = g_3 + p_3(g_2 + p_2(g_1 + p_1 c_0))$$
$$c_4 = g_4 + p_4(g_3 + p_3(g_2 + p_2(g_1 + p_1 c_0)))$$

The more compact MODL design shown in Figure 9.44(b) is often called a Manchester carry chain. Note that the intermediate outputs require secondary precharge transistors. Also note that care must be taken for certain inputs to be mutually exclusive in order to avoid sneak paths.

$$g_i = a_i b_i$$
$$p_i = a_i \oplus b_i$$

If pi were defined as ai + bi, a sneak path could exist when a4 and b4 are 1 and all other inputs are 0. In that case, g4 = p4 = 1. c4 would fire as desired, but c3 would also fire incorrectly.

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 Φ is 0, the first and third stages precharge high while the second stage predischarges low. When Φ rises, all the stages evaluate. Domino connections are possible, as shown in Figure 9.46(c). The design style is called NP Domino or NORA Domino (NO RAce).

- NORA has two major drawbacks. The logical effort of footed p-logic gates is generally worse than that of HI-skew gates (e.g., 2 vs. 3/2 for NOR2 and 4/3 vs. 1 for NAND2). Secondly, NORA is extremely susceptible to noise.

- In an ordinary dynamic gate, the input has a low noise margin (about Vt), but is strongly driven by a static CMOS gate. The floating dynamic output is more prone to noise from coupling and charge shar ing, but drives another static CMOS gate with a larger noise margin.

o In NORA, however, the sensitive dynamic inputs are driven by noiseprone dynamic outputs. Given these drawbacks and the extra clock phase required, there is little reason to use NORA.

o Zipper domino is a closely related technique that leaves the precharge transistors slightly ON during evaluation by using precharge clocks that swing between 0 and VDD – |Vtp| for the pMOS precharge and Vtn and VDD for the nMOS precharge. This plays much the same role as a keeper.

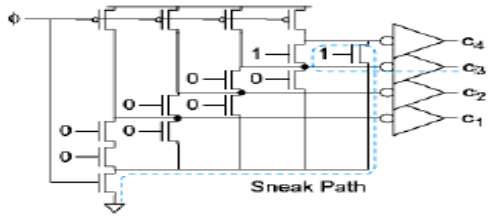


FIGURE 9.45 Sneak path

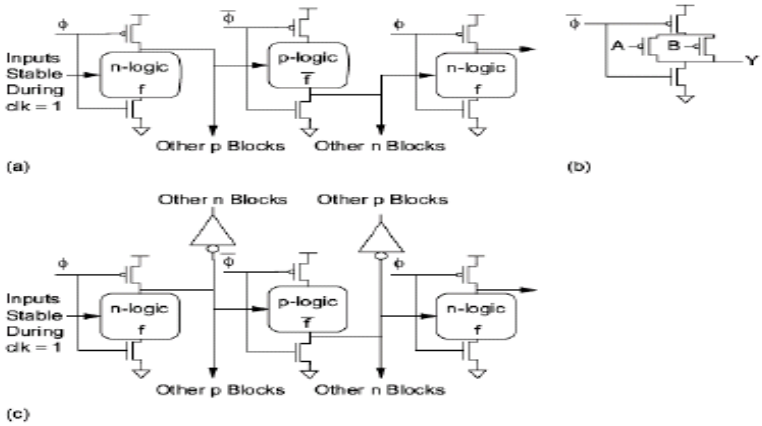


FIGURE 9.46 NP Domino

5. Design the static and dynamic power dissipation in CMOS circuits with necessary diagrams and expressions. [CO2-H3]

THE STATIC AND DYNAMIC POWER DISSIPATION IN CMOS CIRCUITS

Static CMOS gates are very power-efficient because they dissipate nearly zero power while idle. For much of the history of CMOS design, power was a secondary consideration behind speed and area for many chips.

As transistor counts and dock frequencies have increased, power consumption has skyrocketed and now is a primary design constraint.

The instantaneous power P{t} drawn from the power supply is proportional to the supply current iDD(t) and the supply voltage VDD

$$P(t) = i_{DD}(t)\, V_{DD}$$

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

$$E = \int_0^T i_{DD}(t) V_{DD} dt$$

The average power over this interval is

$$P_{avg} = \frac{E}{T} = \frac{1}{T}\int_0^T i_{DD}(t) V_{DD} dt$$

Power dissipation in CMOS circuits comes from two components:

Static dissipation due to

- subthreshold conduction through OFF transistors
- tunneling current through gate oxide
- leakage through reverse-biased diodes
- contention current in ratioed circuits

Dynamic dissipation due to

- charging and discharging of load capacitances "short-circuit" current while both pMOS and nMOS networks are partially ON

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

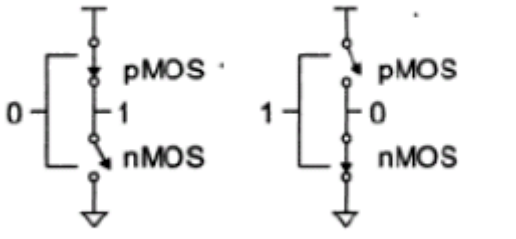
Static Dissipation

- Considering the static CMOS inverter shown in Figure, if the input = '0,' the associated nMOS transistor is OFF and the pMOS transistor is ON. The output voltage is VDD or logic 1.' When the input = 1 the associated nMOS transistor is ON and the pMOS transistor is OFF. The output voltage is 0 volts (GND). Note that one of the transistors is always OFF when the gate is in either of these logic states.

- Ideally, no current flows through the OFF transistor so the power dissipation is zero when the circuit is quiescent, i.e., when no transistors are switching. Zero quiescent power dissipation is a principle advantage of CMOS over competing transistor technologies.

- However, secondary effects including subthreshold conduction, tunneling, and leakage lead to small amounts of static current flowing through the OFF transistor. Assuming the leakage current is constant so instantaneous and average power are the same, the static power dissipation is the product of total leakage current and the supply voltage.

$$P_{static} = I_{static} V_{DD}$$



CMOS inverter model for static power dissipation evaluation

OFF transistors still conduct a small amount of subthreshold current. As subthreshold current is exponentially dependent on threshold voltage, it is increasing dramatically as threshold voltages have scaled down.

There is also some small static dissipation due to reverse biased diode leakage between diffusion regions, wells, and the substrate,

In modern processes, diode leakage is generally much smaller than the subthreshold or gate leakage and may be neglected.

Dynamic Dissipation

Over any given interval of time T, the load will be charged and discharged Tfsw times. Current flows from VDD to the load to charge it. Current then flows from the load to GND during discharge. In one complete charge/discharge cycle, a total charge of Q = CVDD is thus transferred from VDD to GND.

The average dynamic power dissipation is

$$P_{\text{dynamic}} = \frac{1}{T}\int_0^T i_{DD}(t)V_{DD}dt = \frac{V_{DD}}{T}\int_0^T i_{DD}(t)dt$$

$$P_{\text{dynamic}} = \frac{V_{DD}}{T}\left[Tf_{sw}CV_{DD}\right] = CV_{DD}^{\ 2}f_{sw}$$

Because most gates do not switch every' clock cycle, it is often more convenient to express switching frequency fsw as an activity factor a times the clock frequency. Now the dynamic power dissipation may be rewritten as;

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

A clock has an activity factor of α=1, because it rises and falls every cycle. Most data has a maximum activity factor of 0.5 because it transitions only once each cycle.

Static CMOS logic has been empirically determined to have activity factors closer to 0.1 because some gates maintain one output state more often than another.

Because the input rise/fall time is greater than zero, both nMOS and pMOS transistors will be ON for a short period of time while the input is between V_{tn} and $V_{DD} - |V_{tp}|$.

This results in an additional "short circuit" current pulse fromtoGNDa VDD and typically increases power dissipation by about 10% .

Unit -III

Sequential Logic Circuits

Part – A

1. What are the classifications of CMOS circuit families? [**CO3**-**L1]**

Static CMOS circuits.
Dynamic CMOS circuits.
Ratioed circuits.
Pass-transistor circuits.

2. What are the characteristics of Static CMOS design? [**CO3**-**L1]**

A static CMOS circuit is a combination of two networks – the pull-up network (PUN) and the pull-down network (PDN) in which at every point in time, each gate output is connected to either VDD or VSS via a low resistance line.

3. List the important properties of Static CMOS design? [**CO3**-**L1]**

At any instant of time, the output of the gate is directly connected to VDD and VSS.
The function of the PUN is provide a connection between the output and VDD.
The function of the PDN is provide a connection between the output and VSS .
Both PDN and PUN are constructed in mutually exclusive way such that one and only one of the networks is conducting in steady state. That is, the output node is always a low-impedance node in steady state.

4. List the Dynamic CMOS logic? [**CO3**-**L1]**

Dynamic circuits rely on the temporary storage of signal values on the capacitance of high impedance node, Requires only N+2 transistors.Takes a sequence of precharge and conditional evaluation phases to realizes logic functions.

5. Mention the properties of Dynamic logic? [**CO3**-**L2]**

Logic function is implemented by pull-down network only.
Full swing outputs (VOL= GND and VOH = VDD).
Non-ratioed.
Faster switching speeds.
Needs a precharge clock.

6. What are the disadvantages of dynamic CMOS technology? [**CO3**-**L1]**

A fundamental difficulty with dynamic circuits is a loss of noise immunity and a serious timing restriction on the inputs of the gate.
Violate monotonicity during evaluation phase.

7. Mention the CMOS Domino logic? [**CO3-L2]**

A static CMOS inverter placed between dynamic gates which eliminate the monotonicity problem in dynamic circuits are called CMOS Domino logic.

8. What is called static and dynamic sequencing element? [**CO3-L1]**

A sequencing element with static storage employs some sort of feedback to retain its output value indefinitely.
A sequencing element with dynamic storage generally maintain its value as charge on a capacitor that will leak away if not refreshed for a long period of time.

9. What is clock skew? [**CO3-L1]**

In reality clocks have some uncertainty in their arrival times that can cut into the time available for useful computation is called clock skew.

10. What are synchronizers? [**CO3-L1]**

Synchronizers are used to reduce metastability. The synchronizers ensure synchronization between asynchronous input and synchronous system.

11. Compare the difference between melay and moore state machines? [**CO3-L2]**

In the melay state machine we can calculate the next state and output both from the input and state. But in the moore state machine we can calculate only next state but not output from the input and the state and the output is issued according to next state.

12. Define propagation delay and contamination delay? [**CO3-L1]**

Propagation delay (tpd): The amount of time needed for a change in a logic input to result in a permanent change at an output, that is the combinational logic will not show any further output changes in response to an input change alter time fod units contamination delay (tea): The amount of time needed for a change in a logic input to result in an initial change at an output, that is the combinational logic is guaranteed not to show any output change in response to an input change before fed time units have passed.

13. Define Setup time and Hold time. [**CO3-L1]**

Setup time (t setup): The amount of time before the clock edge that data input D must be stable the rising clock edge arrives.
Hold time (t hold): This indicates the amount of time after the clock edge arrives the data input D must be held stable in order for FF to latch the correct value. Hold time is always measured from the rising clock edge to a point after the clock edge.

14. Define MTBF? [**CO3-L1]**

MTBF= (1/P(failure)) = (Ti e(Ti=tsetup/ti)/Nto)

15. What do you meant by Max delay constraint and Min delay constraint? [**CO3**-**L1]**

Min delay constraint: the path begins with the rising edge of the clock triggering F1. The data may begin to change at Q1 after a clk-to-Q contamination delay. However, it must not reach D2 until at least the hold after the clock edge, lest it corrupt the contents of F2. Hence, we solve forminimum logic contamination delay :

tcd >= thold – tccq

Max delay constraint: the path begins with the rising edge of the clock triggering F1. The data must propagate to the output of the flipflop Q1 and through the combinational logic to D2,setting up at F2 before the next rising clock edge. Under ideal conditions, the worst case propagation delays determine the minimum clock period for this sequential circuitry

Tc >= tpcq + tpd + tsetup

16. Define Pipelining. [**CO3**-**L1]**

Pipelining is a popular design technique often used to accelerate the operation of the data path in digital processors. The major advantages of pipelinig are to reduce glitching in complex logic networks and getting lower energy due to operand isolation.

17. How the limitations of a ROM-based realization is overcome in a PLA-based realization. [**CO3**-**L2]**

In a ROM, the encoder part is only programmable and use of ROMs to realize Boolean functions is wasteful in many situations because there is no cross-connect for a significant part.
This wastage can be overcome by using Programmable Logic Array (PLA), which requires much lesser chip area.

18. In what way the DRAMs differ from SRAMs? [**CO3**-**L2]**

Both SRAMs and DRAMs are volatile in nature, ie. Information is lost if power line is removed. However SRAMs provide high switching speed, good noise margin but require large chip area than DRAMs.

19. Explain the read and write operations for a one-transistor DRAM cell. [**CO3**-**L2]**

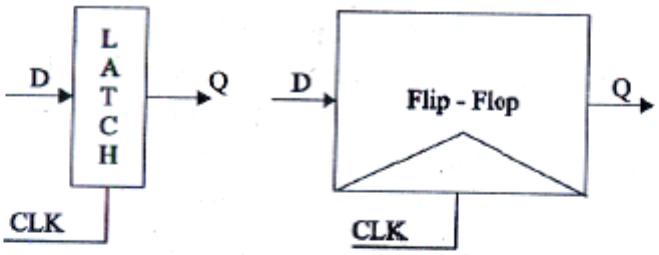
A significant improvement in the DRAM evolution was to realize 1-T DRAM cell. One additional capacitor is explicitly fabricated for storage purpose. To store 'I', it is charged to store '0' it is discharged to '0' volt. Read operation is destructive. Sense amplifier is needed for reading.Read operation is followed by restoration operation.

Part -B

1. Construct the concepts of static latches and registers. [CO3-H3]

THE CHARACTERISTIC OF STATIC LATCHES AND REGISTERS

Latches and flip-flops are the two most commonly used sequencing elements. Both have three terminals data input (D), clock (Clk), and data output (Q). The binary information present at the data input of the D latch is transferred to the Q output when the clock input is high. The output follows changes in the data input as long as the clock input is high. This situation provides a path from input D to the output and for this reason, the circuit is often called as transparent. When the clock input is low, the binary information that was present at the data input at the time the transition occurred is retained at the Q output until the clock input is enabled again, since the circuit operation is said to be opaque. The flip-flop (FF) is a edge-triggered device that copies D to Q on the rising edge of the clock input and ignores D at all other times. The timing diagram of latch and flip-flop are illustrated in Figure 3.1



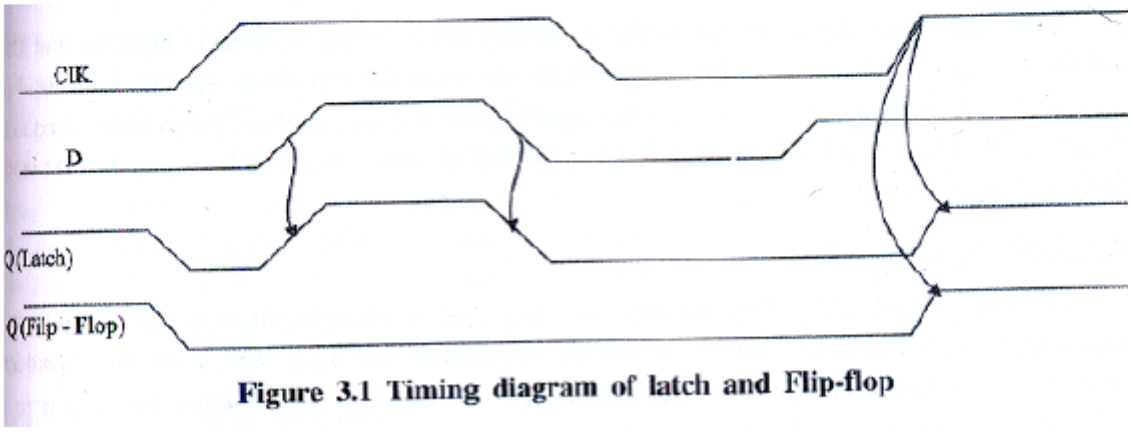


Figure 3.1 Timing diagram of latch and Flip-flop

Metastability

Flip-flop is a device that is susceptible to metastability. It has two well-defined stable states, traditionally designated 0 and **1,** but under certain conditions it can hover between them for longer than a clock cycle. This condition is known as metastability. Such a metastable "state" **is** considered a failure mode of the logic design and timing

constraints. The most common cause of melastability is violating the flip-flop's setup and hold limes. During the time from the setup to the hold lime, the input of the flip-flop should remain in a stable logic state; a change in the input in that time will have a probability of setting the flip-flop to a metastable state. Most FFs in CMOS IC design are based on the cross-coupled inverters shown in Figure **3.2.** Also seen **in** this Figure are the VTCs (Voltage Transfer Curve) for the two inverters. It's characteristics are drawn with its input on the x-axis and its output on the y-axis the normal curves as **I2's** input is labeled Out in the figure while its output is labeled In (and so its input is drawn as the y-axis and its output is drawn as the x-axis).

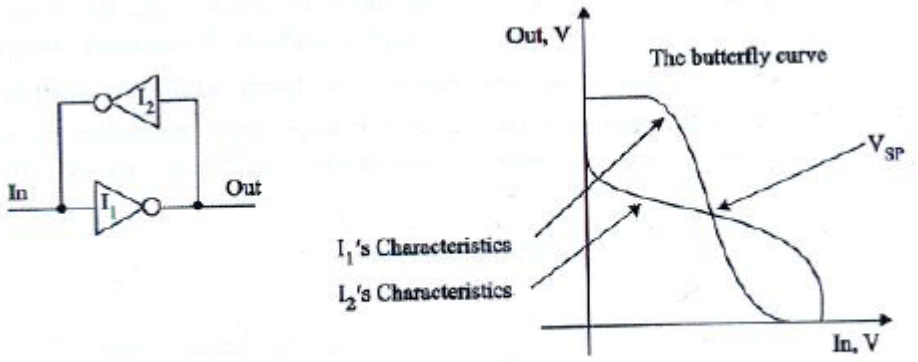


Figure 3.2 A cross-coupled inverter (a latch) and the input output characteristics.

OPERATION OF READ-ONLY MEMORY (ROM)

Read-only memory device are widely used to store programs and permanent data in computers and embedded systems. Basically, a ROM device uses the presence or absence of a transistor to represent the information bit 0 or **1** and can be built based on either a NOR or NAND structure. In the NOR-type structure, each bit-line comprises of large pseudo-nMOS gate while in the NAND - type structure, each bit-line is a large pseudo-nMOS NAND gate.

Figure 3.58 shows an example of a 4 X 4 memory array structure of active-programming NOR-type ROM. The active layer only appears in the place where there are transistors. The gate-grounded pull-up pMOS transistors can be replaced by precharge transistors in order to reduce the power dissipation. To keep the memory cell size and the bit-line capacitance small, the sizes of pull-down transistors should be kept as small as possible. The layout of the memory array shown in Figure 3.58 (a) is revealed in Figure 3.58 (b).

From the Figure, it is noticed that each bit-line is a pseudo-nMOS NOR gate. To read a word WL[i] from a NOR-type ROM, apply a logic **1** to WL[i] and a logic 0 to all other rows. The bit-line will receive a low voltage when its cell stores a logic 0 (i.e., the presence of transistor) and a high voltage when its cell stores a logic 1 (i.e., the absence of transistor). In practice, due to the large bit-line capacitance, current sense amplifiers are often employed to improve the speed of the read operation.

The metastable state of a flip-flop is analogous to a ball on the summit of a hill between two valleys, as shown in Figure 3.3. The two valleys are stable states, because a ball in the valley will remain there as long as it is not disturbed. The top of the hill is called metastable because the ball would remain there if it were perfectly balanced. But

because nothing is perfect, the ball will eventually roll to one side or the other. The time required for this change to occur depends on how nearly well balanced the ball originally was. Every bistable device has a metastable state between the two stable states.

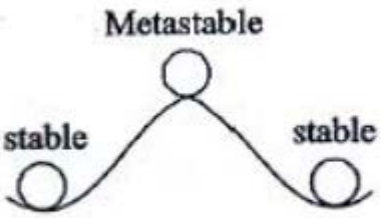


Figure 3.3 Stable and metastable states

Resolution Time

If a flip-flop input changes at a random time during the clock cycle, the resolution time, tres, required to resolve to a stable state is also a random variable. If the input changes outside the aperture, then tres = tpcq. But if the input happens to change within the aperture, tres can be substantially longer. Theoretical and experimental analyses have shown that the probability that the resolution time, tres, exceeds some arbitrary time, t, decreases exponentially with t:

$$P(t_{res} > t) = \frac{T_0}{T_c} e^{-\frac{T}{\tau}}$$

where Tc is the clock period, tpcq is the clock to Q propagation delay, T0 and τ are characteristic of the flip-flop. The equation is valid only for t substantially longer than tpcq. Intuitively, T0/Tc describes the probability that the input changes at a bad time (i.e., during the aperture time); this probability decreases with the cycle time, Tc, T is a time constant indicating how fast the flip-flop moves away from the metastable state; it is related to the delay through the cross-coupled gates in the flip-flop.

2. Design sequential circuits for Flip-flops2-phase transparent latches and pulsed latches. **[CO3-H3]**

THE SEQUENCING METHODS

The three most widely used methods of sequencing static circuits are

1. Flip-flops
2. 2-phase transparent latches
3. pulsed latches

1. Flip-flops

Flip-flop based systems use one flip-flop on each cycle boundary. Their timing diagram is shown in Figure 3.4. When the clock rises, the data input is sampled and transferred to the output after a delay of tpd. At all other times, the data input and output are unrelated. For the correct value to be sampled, data input must stabilize a setup time tsetup before the rising edge of the clock and must remain stable for a hold time thold after the rising edge.

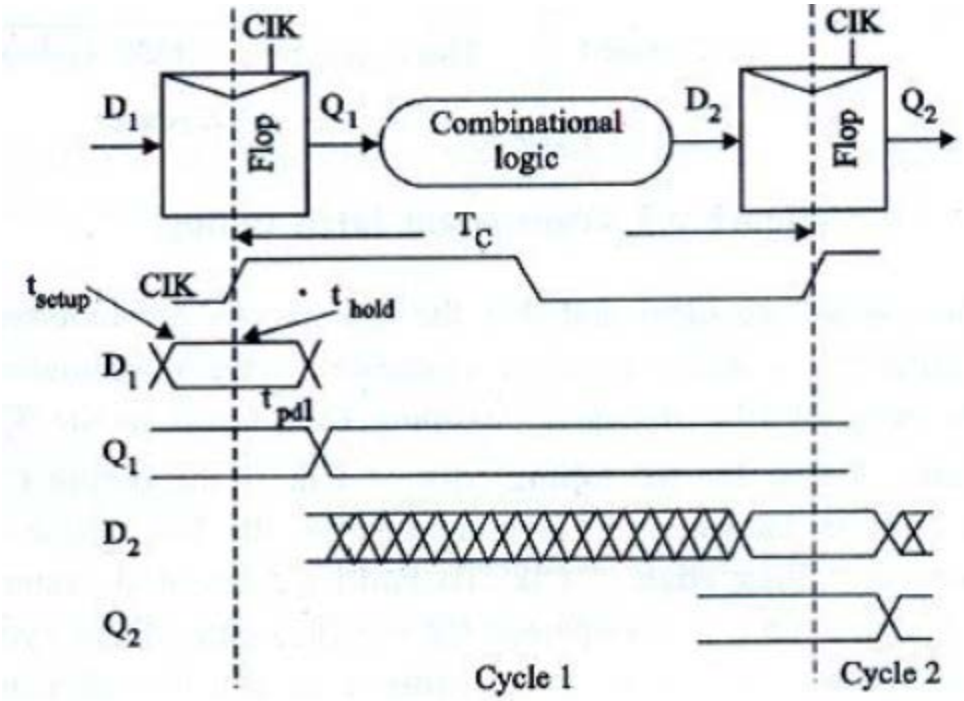


Figure 3.4 Flip-flop timing

Check that memory element properly sequence data by making sure that data from this cycle does not mix with data from the previous or next cycles. This is clear for a flip-flop because all data advances on the rising edge of the clock and at no other time.

2. **Transparent latches**

Transparent latches are also known as half-latches, D-latches, or two-phase latches. Their timing is shown in Figure 3.5.

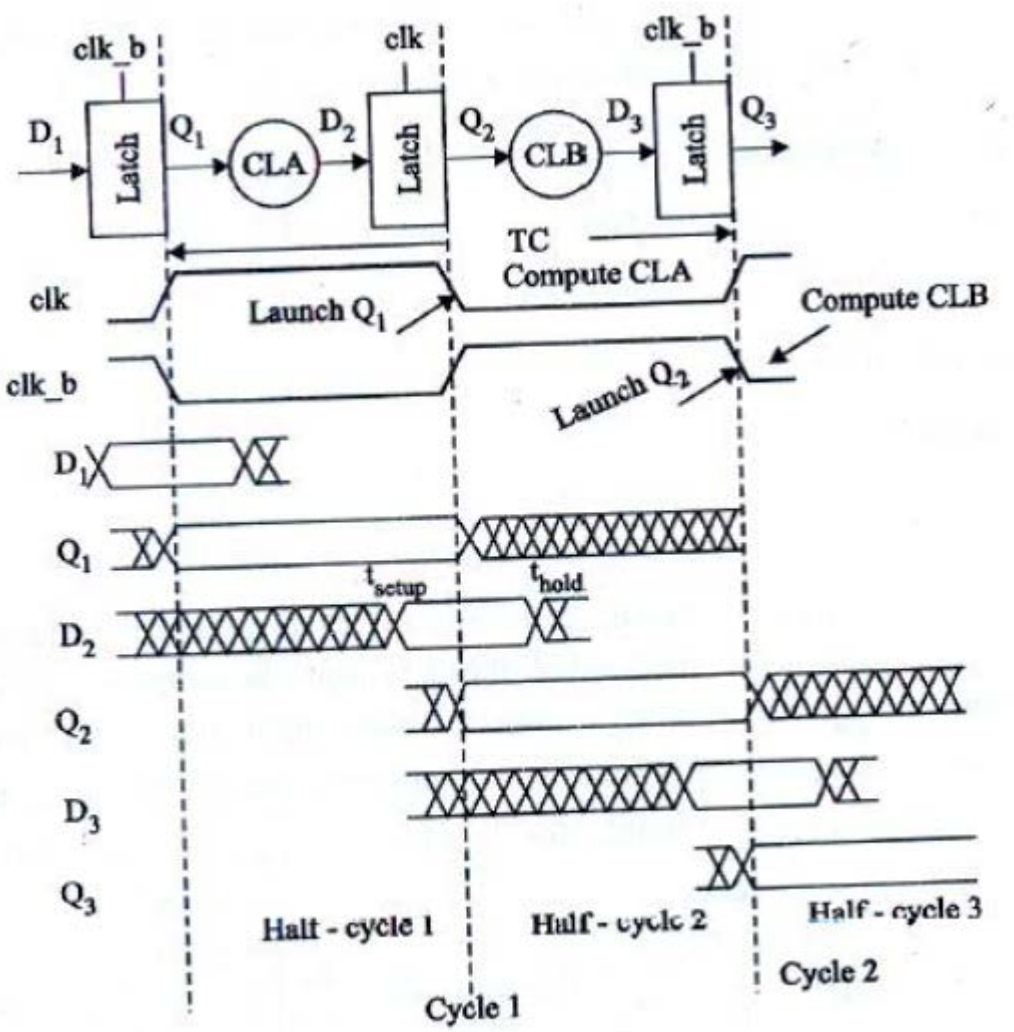


Figure 3.5 Transparent latch timing

Assume that the clocks are ideal and that the two clocks are inverted versions of each other. In this configuration, a stable input is available to the combinational logic block **A** (CLA) on the falling edge of Clk. It has a maximum time equal to the Tc/2 to evaluate- that is, the entire low phase of Clk. On the falling edge of Clk_b, the output CLA is latched, and the computation of CLB is launched. CLB computes on the low phase of Clk_b, and the output is available on the falling edge of Clk. To build a sequential system, two half-latches must be used in each cycle. One is transparent for the first part of the cycle, while the other is transparent for the second part of the cycle using a locally complemented version of the clock, Clk_b.

3. Pulsed latches

The pulsed latch behaves like a flip-flop but has a short clock-to-**Q** delay and a long hold time. Pulsed latch are otherwise called single-phase level triggered latch. Pulsed latch systems eliminate one of the latches from each cycle and apply a brief pulse to the remaining latch. If the pulse is shorter than the delay through the combinational logic, it is expected that a token will only advance through one clock cycle on each pulse. Pulsed latch and its timing diagram is shown in Figure **3.6.**

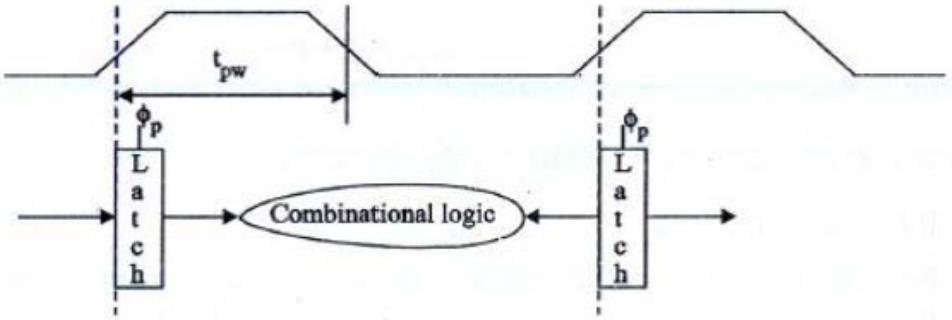


Figure 3.6 pulsed latches

Table **3**.1 defines the delays and timing constraints of the combinational logic and sequencing elements

Term	Name	Definition
t_{pd}	Logic propagation delay	UPPER bound on interval between valid inputs and valid outputs
t_{cd}	Logic contamination delay	LOWER bound on interval between invalid inputs and invalid outputs.
t_{pcq}	Latch/flip-flop clock-to-Q propagation delay	(i) t_{PLH}: 50% triggering edge point of the clock pulse to 50% transition of the output from low to high. (ii) t_{PHL}: 50% triggering edge of the clock pulse to the high to low transition of the output.
t_{ccq}	Latch/flip-flop clock-to-Q contamination delay	Output signal start to change after its input change and settles to the final value within a propagation delay.
t_{pdq}	Latch D-to-Q propagation delay	Assuming that the setup and hold times are met, the data at the D input is copied to the Q output after a worse case propagation delay.

Term	Name	Definition
t_{cdq}	Latch *D*-to-*Q* contamination delay	This delay ensures that the *D* input of the sequential elements is held enough after the clock edge and is not modified too soon by the new value of data coming in.
t_{setup}	Latch/flip-flop setup time	The time before the clock edge that the *D* input has to be stable is called setup time.
t_{hold}	Latch/flip-flop hold time	The time after the clock edge that the *D* input has remain stable is called hold time.

Timing issues:

Latches and flip-flops are the two most commonly used sequencing elements. Both have three terminals: data input (D), clock (clk), and data output (Q). The latch is transparent when the clock is high and opaque when the clock is low; in other words, when the clock is high, D flows through to Q as if the latch were just a buffer, but when the clock is low, the latch holds its present Q output even if D changes.

The flip-flop is an edge-triggered device that copies D to Q on the rising edge of the clock and ignores D at all other times. These are illustrated in Figure 10.1. The unknown state of Q before the first rising clock edge is indicated by the pair of lines at both low and high levels.

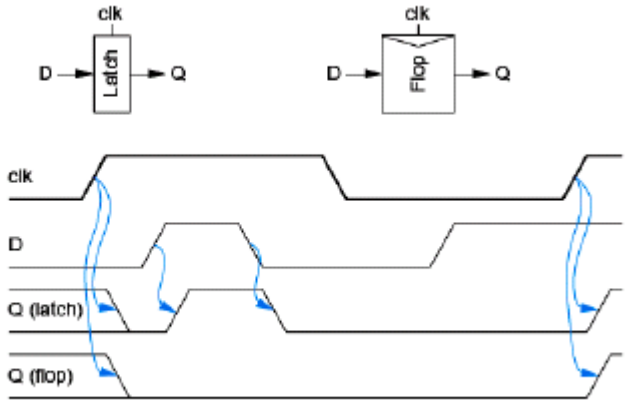


FIGURE 10.1 Latches and flip-flops

This section explores the three most widely used methods of sequencing static circuits with these elements: flip-flops, 2-phase transparent latches, and pulsed latches [Unger86]. An ideal sequencing methodology would introduce no sequencing overhead, allow sequencing elements back-to-back with no logic in between, grant the designer flexibility in balancing the amount of logic in each clock cycle, tolerate moderate amounts of lock skew without degrading performance, and consume zero area and power.

3. Illustrated in detail about the principle timing matrices for sequential circuits.
[CO3-H2]

TIMING MATRICES FOR SEQUENTIAL CIRCUITS:

Max-Delay Constraints

Ideally, the entire clock cycle would be available for computations in the combinational logic. Of course, the sequencing overhead of the latches or flip-flops cuts into this time. If the combinational logic delay is too great, the receiving element will miss its setup time and sample the wrong value. This is called a setup time failure or max-delay failure.

It can be solved by redesigning the logic to be faster or by increasing the clock period. This section computes the actual time available for logic and the sequencing overhead of each of our favorite sequencing elements: flip-flops, two-phase latches, and pulsed latches.

Figure 10.5 shows the max-delay timing constraints on a path from one flip-flop to the next, assuming ideal clocks with no skew. The path begins with the rising edge of the clock triggering F1. The data must propagate to the output of the flip-flop Q1 and through the combinational logic to D2, setting up at F2 before the next rising clock edge.

This implies that the clock period must be at least

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

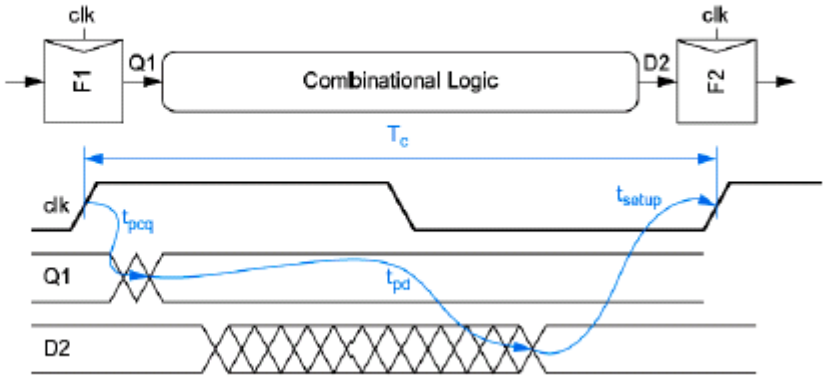


FIGURE 10.5 Flip-flop max-delay constraint

Alternatively, we can solve for the maximum allowable logic delay, which is simply the cycle time less the sequencing overhead introduced by the propagation delay and setup time of the flip-flop.

$$t_{pd} \leq T_c - \underbrace{\left(t_{setup} + t_{pcq}\right)}_{\text{sequencing overhead}}$$

Technically, D3 could arrive as late as a setup time before the falling edge of Φ1 and still be captured correctly by L3. To be fair, we will insist that D3 nominally arrive no more than one clock period after D1 because, in the long run, it is impossible for every single-cycle path in a design to consume more than a full clock period. Certain paths

may take longer if other paths take less time; this technique is called time borrowing and will be addressed in Section 10.2.4.

Assuming the path takes no more than a cycle, we see the cycle time must be

$$T_c \geq t_{pdq1} + t_{pd1} + t_{pdq2} + t_{pd2}$$

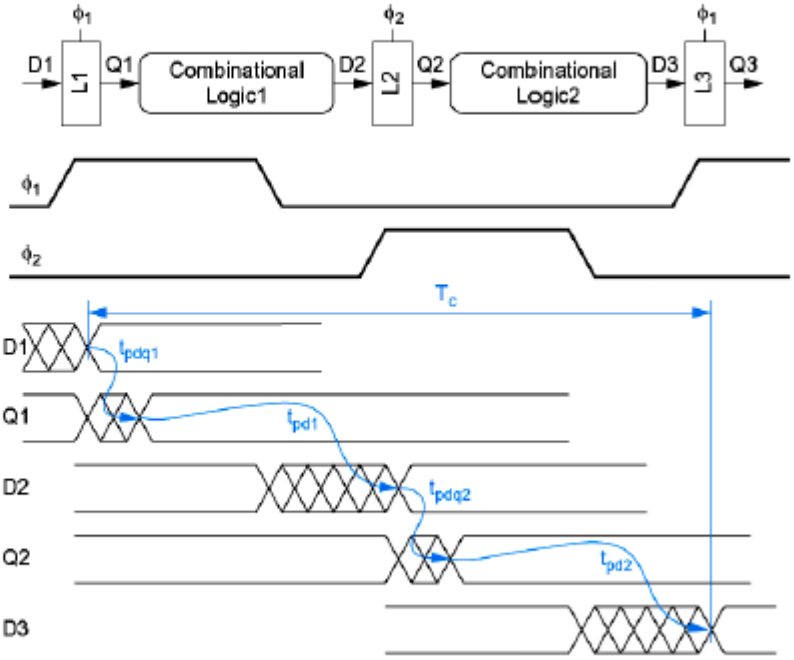


FIGURE 10.7 Two-phase latch max-delay constraint

Once again, we can solve for the maximum logic delay, which is the sum of the logic delays through each of the two phases. The sequencing overhead consists of the two latch propagation delays.

Notice that the nonoverlap between clocks does not degrade performance in the latch-based system because data continues to propagate through the combinational logic between latches even while both clocks are low.

Realizing that a flip-flop can be made from two latches whose delays determine the flop propagation delay and setup time, we see EQ (10.4) is closely analogous to EQ (10.2).

$$t_{pd} = t_{pd1} + t_{pd2} \leq T_c - \underbrace{\left(2t_{pdq}\right)}_{\text{sequencing overhead}}$$

The max-delay constraint for pulsed latches is similar to that of two-phase latches except that only one latch is in the critical path, as shown in Figure 10.8(a). However, if the pulse is narrower than the setup time, the data must set up before the pulse rises, as shown in Figure 10.8(b). Combining these two cases gives

$$T_c \geq \max\left(t_{pdq} + t_{pd}, t_{pcq} + t_{pd} + t_{\text{setup}} - t_{pw}\right)$$

Solving for the maximum logic delay shows that the sequencing overhead is just one latch delay if the pulse is wide enough to hide the setup time

$$t_{pd} \leq T_c - \underbrace{\max\left(t_{pdq}, t_{pcq} + t_{\text{setup}} - t_{pw}\right)}_{\text{sequencing overhead}}$$

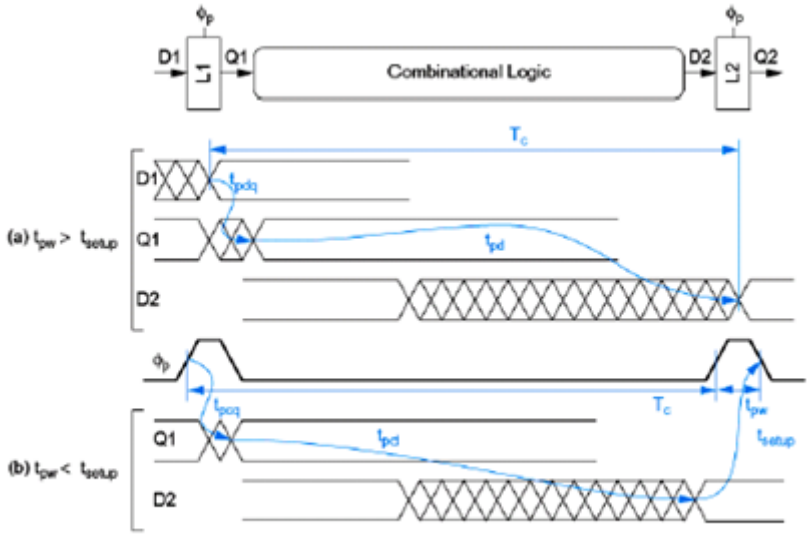


FIGURE 10.8 Pulsed latch max-delay constraint

Min-Delay Constraints

□□ If the hold time is large and the contamination delay is small, data can incorrectly propagate through two successive elements on one clock edge, corrupting the state of the system. This is called a race condition, hold-time failure, or min-delay failure.

□□ It can only be fixed by redesigning the logic, not by slowing the clock. Therefore, designers should be very conservative in avoiding such failures because modifying and refabricating a chip is catastrophically expensive and time-consuming.

□□ Figure 10.9 shows the min-delay timing constraints on a path from one flip-flop to the next assuming ideal clocks with no skew. The path begins with the rising edge of the clock triggering F1. The data may begin to change at Q1 after a clk-to-Q contamination delay and at D2 after another logic contamination delay. However, it must not reach D2 until at least the hold time t hold after the clock edge, lest it corrupt the contents of F2. Hence, we solve for the minimum logic contamination delay:

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

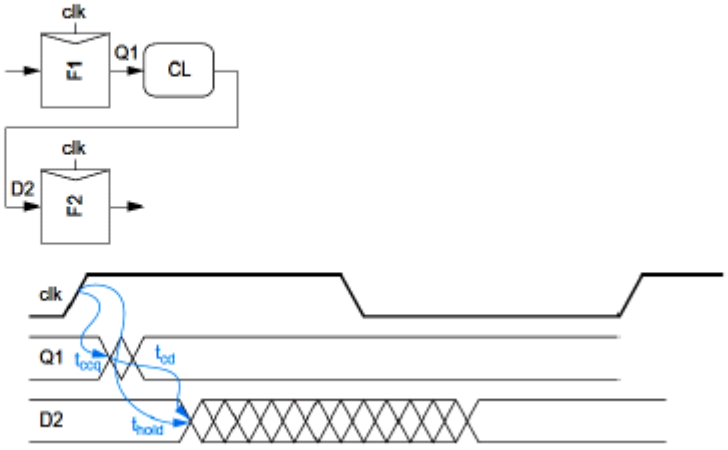


FIGURE 10.9 Flip-flop latch min-delay constraint

If the contamination delay through the flip-flop exceeds the hold time, you can safely use back-to-back flip-flops. If not, you must explicitly add delay between the flip-flops (e.g., with a buffer) or use special slow flip-flops with greater than normal contamination delay on paths that require back-to-back flops. Scan chains are a common example of paths with back-to-back flops.

Figure 1010 shows the min-delay timing constraints on a path from one transparent latch to the next. The path begins with data passing through L1 on the rising edge of Φ1. It must not reach L2 until a hold time after the previous falling edge of Φ2 because L2 should have become safely opaque before L1 becomes transparent. As the edges are separated by tnonoverlap, the minimum logic contamination delay through each phase of logic is

$$t_{cd1}, t_{cd2} \geq t_{\text{hold}} - t_{ccq} - t_{\text{nonoverlap}}$$

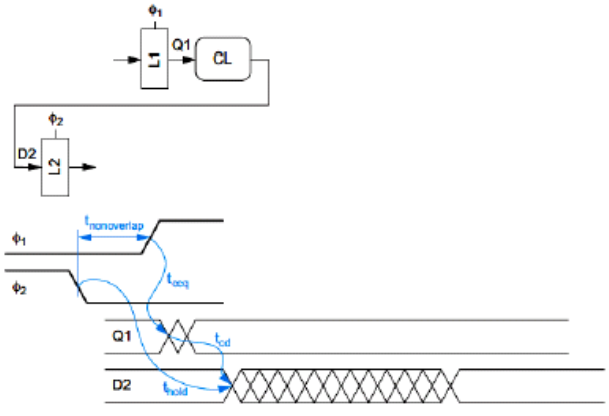


FIGURE 10.10 Two-phase latch min-delay constraint

This result shows that by making tnonoverlap sufficiently large, hold-time failure can be avoided entirely. Therefore, most commercial transparent latch-based systems use the

clock and its complement. In this case, tnonoverlap = 0 and the contamination delay constraint is the same between the latches and flip-flops.

This leads to an apparent paradox: The contamination delay constraint applies to each phase of logic for latch-based systems, but to the entire cycle of logic for flip-flops. Therefore, latches seem to require twice the overall logic contamination delay as compared to flip-flops.

Yet flipflops can be built from a pair of latches! The paradox is resolved by observing that a flip-flop has an internal race condition between the two latches. The flipflop must be carefully designed so that it always operates reliably.

Figure 10.11 shows the min-delay timing constraints on a path from one pulsed latch to the next. Now data departs on the rising edge of the pulse but must hold until after the falling edge of the pulse. Therefore, the pulse width effectively increases the hold time of the pulsed latch as compared to a flip-flop.

$$t_{cd} \geq t_{\text{hold}} - t_{cq} + t_{pw}$$

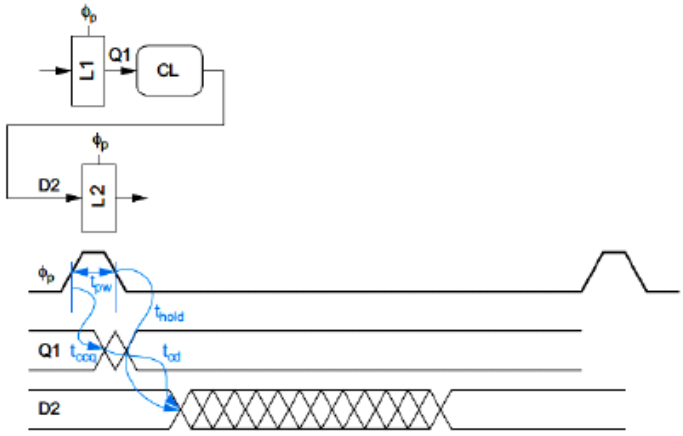


FIGURE 10.11 Pulsed latch min-delay constraint

Time Borrowing

In contrast, when a system uses transparent latches, the data can depart the first latch on the rising edge of the clock, but does not have to set up until the falling edge of the clock on the receiving latch. If one half-cycle or stage of a pipeline has too much logic, it can borrow time into the next half-cycle or stage, as illustrated in Figure 10.12(a).

Time borrowing can accumulate across multiple cycles. However, in systems with feedback, the long delays must be balanced by shorter delays so that the overall loop completes in the time available. For example, Figure 10.12(b) shows a single-cycle self bypass loop in which time borrowing occurs across half-cycles, but the entire path must fit in one cycle.

A typical example of a self-bypass loop is the execution stage of a pipelined processor in which an ALU must complete an operation and bypass the result back for

use in the ALU on a dependent instruction. Most critical paths in digital systems occur in self-bypass loops because otherwise latency does not matter.

Figure 10.13 illustrates the maximum amount of time that a two-phase latch-based system can borrow (beyond the Tc /2 – tpdq nominally available to each half-cycle of logic).

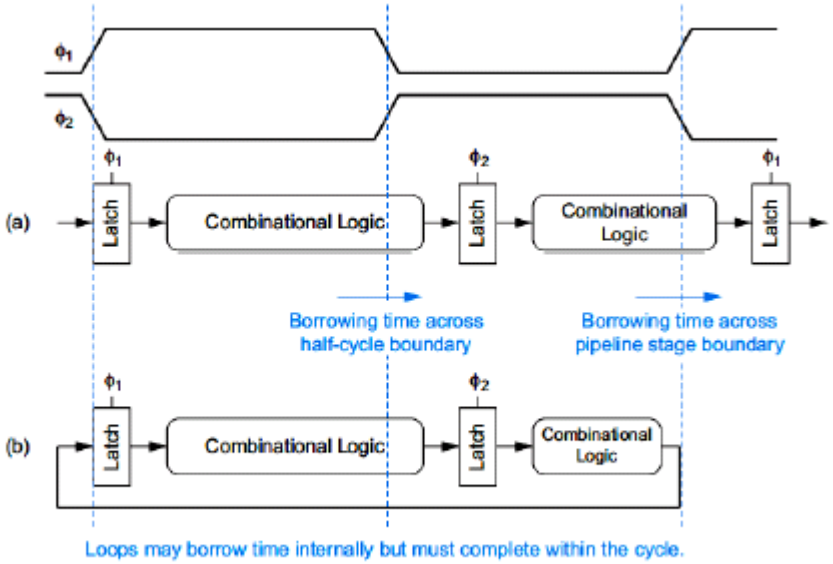


FIGURE 10.12 Time borrowing

Because data does not have to set up until the falling edge of the receiving latch's clock, one phase can borrow up to half a cycle of time from the next (less setup time and nonoverlap):

$$t_{\text{borrow}} \leq \frac{T_c}{2} - \left(t_{\text{setup}} + t_{\text{nonoverlap}}\right)$$

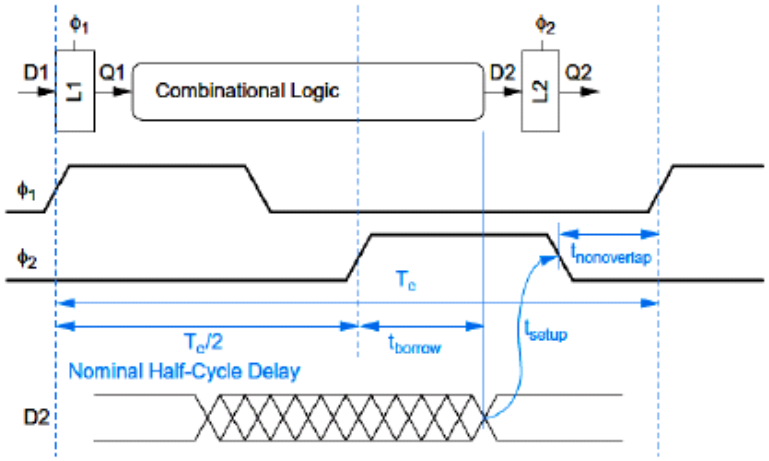


FIGURE 10.13 Maximum amount of time borrowing

Pulsed latches can be viewed as transparent latches with a narrow pulse. If the pulse is wider than the setup time, pulsed latches are also capable of a small amount of time borrowing from one cycle to the next.

$$t_{\text{borrow}} \leq t_{pw} - t_{\text{setup}}$$

Time borrowing has two benefits for the system designer. The most obvious is intentional time borrowing, in which the designer can more easily balance logic between half-cycles and pipeline stages. This leads to potentially shorter design time because the balancing can take place during circuit design rather than requiring changes to the micro architecture to explicitly move functions from one stage to another. The other benefit is opportunistic time borrowing.

Even if the designer carefully equalizes the delay in each stage at design time, the delays will differ from one stage to another in the fabricated chip because of process and environmental variations and inaccuracies in the timing model used by the CAD system. In a system capable of time borrowing, the slow cycles can opportunistically borrow time from faster ones and average out some of the variation.

Some experienced design managers forbid the use of intentional time borrowing until the chip approaches tape out. Otherwise designers are overly prone to assuming that their pipeline stage can borrow time from adjacent stages.

When many designers make this same assumption, all of the paths become excessively long. Worse yet, the problem may be hidden until full-chip timing analysis begins, at which time it is too late to redesign so many paths.

Another solution is to do full-chip timing analysis starting early in the design process.

Clock Skew

In reality, clocks have some uncertainty in their arrival times that can cut into the time available for useful computation, as shown in Figure 10.15(a). The bold clk line indicates the latest possible clock arrival time. The hashed lines show that the clock might arrive over a range of earlier times because of skew.

The worst scenario for max delay in a flip-flop-based system is that the launching flop receives its clock late and the receiving flop receives its clock early. In this case, the clock skew is subtracted from the time available for useful computation and appears as sequencing overhead.

The worst scenario for min delay is that the launching flop receives its clock early and the receiving clock receives its clock late, as shown in Figure 10.15(b). In this case, the clock skew effectively increases the hold time of the system.

$$t_{pd} \leq T_c - \underbrace{\left(t_{pcq} + t_{\text{setup}} + t_{\text{skew}} \right)}_{\text{sequencing overhead}}$$

$$t_{cd} \geq t_{\text{hold}} - t_{ccq} + t_{\text{skew}}$$

In the system using transparent latches, clock skew does not degrade performance. Figure 10.16 shows how the full cycle (less two latch delays) is available for computation even when the clocks are skewed because the data can still arrive at the latches while they are transparent. Therefore, we say that transparent latch-based systems are skew-tolerant.

However, skew still effectively increases the hold time in each half-cycle. It also cuts into the window available for time borrowing.

$$t_{pd} \le T_c - \underbrace{\left(2t_{pdq}\right)}_{\text{sequencing overhead}}$$

$$t_{cd1}, t_{cd2} \ge t_{\text{hold}} - t_{cq} - t_{\text{nonoverlap}} + t_{\text{skew}}$$

$$t_{\text{borrow}} \le \frac{T_c}{2} - \left(t_{\text{setup}} + t_{\text{nonoverlap}} + t_{\text{skew}}\right)$$

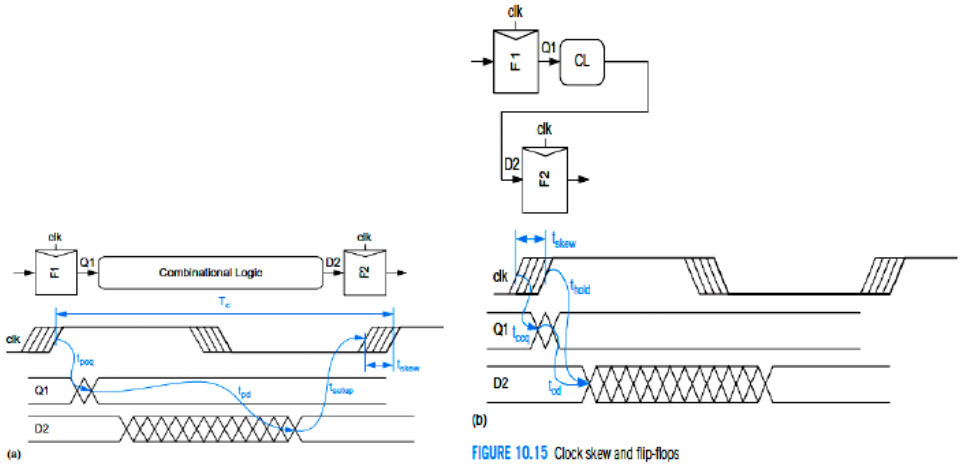


FIGURE 10.15 Clock skew and flip-flops

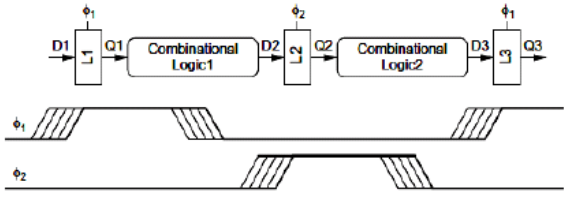


FIGURE 10.16 Clock skew and transparent latches

Pulsed latches can tolerate an amount of skew proportional to the pulse width. If the pulse is wide enough, the skew will not increase the sequencing overhead because the data can arrive while the latch is transparent. If the pulse is narrow, skew can degrade performance. Again, skew effectively increases the hold time and reduces the amount of time available for borrowing.

$$t_{pd} \le T_c - \underbrace{\max\left(t_{pdq},\ t_{pcq} + t_{\text{setup}} - t_{pw} + t_{\text{skew}}\right)}_{\text{sequencing overhead}}$$

$$t_{cd} \ge t_{\text{hold}} + t_{pw} - t_{cq} + t_{\text{skew}}$$

$$t_{\text{borrow}} \le t_{pw} - \left(t_{\text{setup}} + t_{\text{skew}}\right)$$

4. Design Conventional CMOS Latches and Flip-Flops. [CO3-H3]

Conventional CMOS Latches

Figure 10.17(a) shows a very simple transparent latch built from a single transistor. It is compact and fast but suffers four limitations.

(i)The output does not swing from rail-to-rail(i.e., from GND to VDD);

(ii)It never rises above VDD – Vt.

(iii)The output is also dynamic; in other words, the output floats when the latch is opaque. If it floats long enough, it can be disturbed by leakage D drives the diffusion input of a pass transistor directly, leading to potential noise issues and making the delay harder to model with static timing analyzers.

(iv)Finally, the state node is exposed, so noise on the output can corrupt the state.

The remainder of the figures illustrate improved latches using more transistors to achieve more robust operation. Figure 10.17(b) uses a CMOS transmission gate in place of the single nMOS pass transistor to offer rail-to-rail output swings. It requires a complementary clock which can be provided as an additional input or locally generated from through an inverter.

Figure 10.17(c) adds an output inverter so that the state node X is isolated from noise on the output. Of course, this creates an inverting latch.

Figure 10.17(d)also behaves as an inverting latch with a buffered input but unbuffered output. The inverter followed by a transmission gate is essentially equivalent to a tristate inverter but has as lightly lower logical effort because the outputs driven by both transistors of the transmission gate in parallel.

Figure 10.17(c) and(d) are both fast dynamic latches. In modern processes, subthreshold leakage is large enough that dynamic nodes retain their values for only a short time, especially at the high temperature and voltage encountered during burn-in test.

Therefore, practical latches need to be stabilized, adding feedback to prevent the output from floating, as shown in Figure 10.17(e).

When the clock is 1, the input transmission gate is ON, the feedback tristate is OFF, and the latch is transparent. When the clock is 0, the input transmission gate turns OFF. However, the feedback tristate turns ON, holding X at the correct level.

Figure10.17(f) adds an input inverter so the input is a transistor gate rather than unbuffered diffusion. Unfortunately, both (e) and (f) reintroduced output noise sensitivity: A large noise spike on the output can propagate backward through the feedback gates and corrupt the state node X.

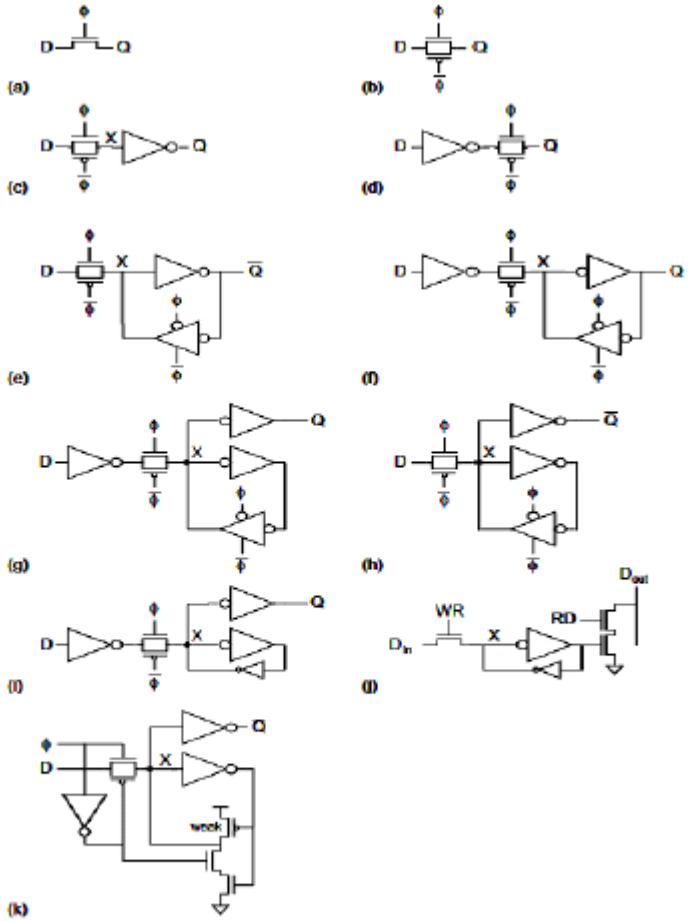


FIGURE 10.17 Transparent latches

Figure 10.17(g) is a robust transparent latch that addresses all of the deficiencies mentioned so far: The latch is static, all nodes swing rail-to-rail, the state noise is isolated from output noise, and the input drives transistor gates rather than diffusion. Such a latch is widely used in standard cell applications. It is recommended for all but the most performance- or area-critical designs.

In semicustom data path applications where input noise can be better controlled, the inverting latch of Figure 10.17(h) may be preferable because it is faster and more compact. Intel uses this as a standard data path latch [Karnik01].

Figure 10.17(i) shows the jamb latch, a variation of Figure 10.17(g) that reduces the clock load and saves two transistors by using a weak feedback inverter in place of the tristate. This requires careful circuit design to ensure that the tristate is strong enough to overpower the feedback inverter in all process corners.

Figure 10.17(j) shows another jamb latch commonly used in register files and Field Programmable Gate Array (FPGA) cells. Many such latches read out onto a single Dout wire and only one latch is enabled at any given time with its RD signal. The Itanium2 processor uses the latch shown in Figure 10.17(k) [Naffziger02].

In the static feedback, the pulldown stack is clocked, but the pullup is a weak pMOS transistor. Therefore, the gate driving the input must be strong enough to overcome the feedback. The Itanium 2cell library also contains a similar latch with an additional input

inverter to buffer the input when the previous gate is too weak or far away. With the input inverter, the latch can be viewed as a cross between the designs shown in (g) and(i). Some latches add one more inverter to provide both true and complementary outputs.

The dynamic latch of Figure 10.17(d) can also be drawn as a clocked tristate, as shown in Figure 10.18(a).Such a form is sometimes called clocked CMOS (C2MOS). The conventional form using the inverter and transmission gate is slightly faster because the output is driven through the nMOS and pMOS working in parallel.C2MOS is slightly smaller because it eliminates two contacts.

Figure 10.18(b) shows another form of the tristate that swaps the data and clock terminals. It is logically equivalentbut electrically inferior because toggling D while thelatch is opaque can cause charge-sharing noise on the outputnode.

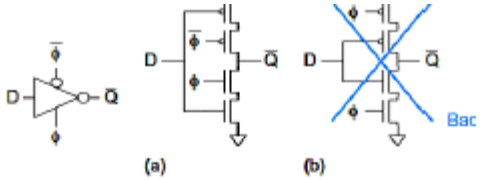


FIGURE 10.18 C²MOS Latch

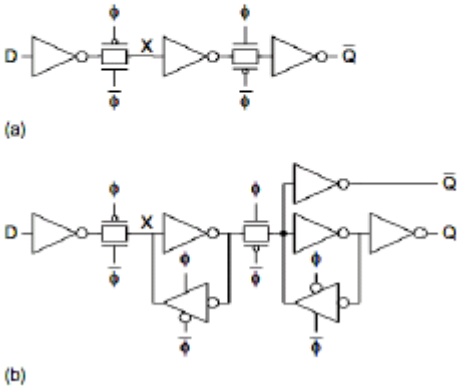


FIGURE 10.19 Flip-flops

Pulsed Latches

A pulsed latch can be built from a conventional CMOS transparent latch driven by a brief clock pulse. Figure 10.22(a) shows a simple pulse generator, sometimes called a clock chopper or one-shot.

The pulsed latch is faster than a regular flipflop because it involves a single latch rather than two and because it allows time borrowing. It can also consume less energy, although the pulse generator adds to the energy consumption (and is ideally shared across multiple pulsed latches for energy and area efficiency). The drawback is the increased hold time.

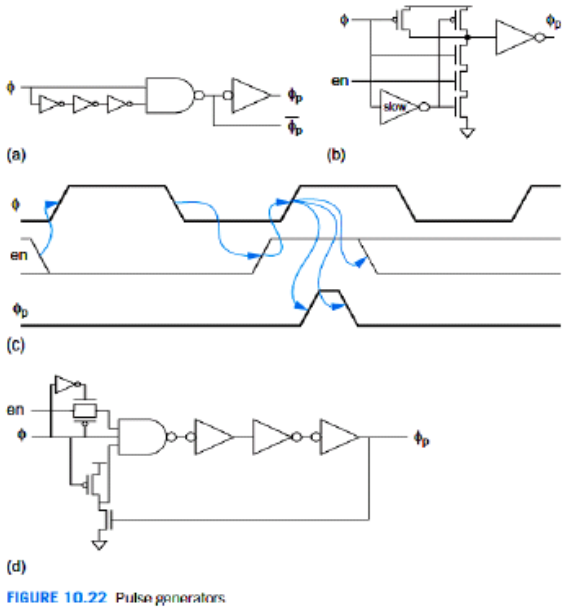


FIGURE 10.22 Pulse generators

This pulse generator uses a fairly slow (weak) inverter to produce a pulse with a nominal width of about one-sixth of the cycle (125 ps for 1.2 GHz operation). When disabled, the internal node of the pulse generator floats high momentarily, but no keeper is required because the duration is short.

Of course, the enable signal has setup and hold requirements around the rising edge of the clock, as shown in Figure 10.22(c). Figure 10.22(d) shows yet another pulse generator used on an NEC RISC processor to produce substantially longer pulses.

It includes a built-in dynamic transmission gate latch to prevent the enable from glitching during the pulse. Many designers consider short pulses risky. The pulse generator should be carefully simulated across process corners and possible RC loads to ensure the pulse is not degraded too badly by process variation or routing.

However, the Itanium 2 team found that the pulses could be used just as regular clocks as long as the pulse generator had adequate drive. The quad-core Itanium pulse generator selects between 1- and 3-inverter delay chains using a transmission gate multiplexer.

The wider pulse offers more robust latch operation across process and environmental variability and permits more time borrowing, but increases the hold time. The multiplexer select is software-programmable to fix problems discovered after fabrication.

The Partovi pulsed latch in Figure 10.23 eliminates the need to distribute the pulse by building the pulse generator into the latch itself. The weak crosscoupled inverters in the dashed box staticize the circuit, although the latch is susceptible to back-driven output noise on Q or Q unless an extra inverter is used to buffer the output.

The Partovi pulsed latch was used on the AMD K6 and Athlon, but is slightly slower than a simple latch. It was originally called an EdgeTriggered Latch (ETL), but strictly speaking is a pulsed latch because it has a brief window of transparency.

Resettable Latches and Flip-Flops

Most practical sequencing elements require a reset signal to enter a known initial state on start up and ensure deterministic behavior. Figure 10.24 shows latches and flip-flops with reset inputs.

There are two types of reset: synchronous and asynchronous. Asynchronous reset forces Q low immediately, while synchronous reset waits for the clock.

Synchronous reset signals must be stable for a setup and hold time around the clock edge while asynchronous reset is characterized by a propagation delay from reset to output. Synchronous reset simply requires ANDing the input D with reset.

Asynchronous reset requires gating both the data and the feedback to force the reset independent of the clock. The tristate NAND gate can be constructed from a NAND gate in series with a clocked transmission gate. Settable latches and flip-flops force the output high instead of low.

They are similar to resettable elements of Figure 10.24 but replace NAND with NOR and reset with set. Figure 10.25 shows a flip-flop combining both asynchronous set and reset.

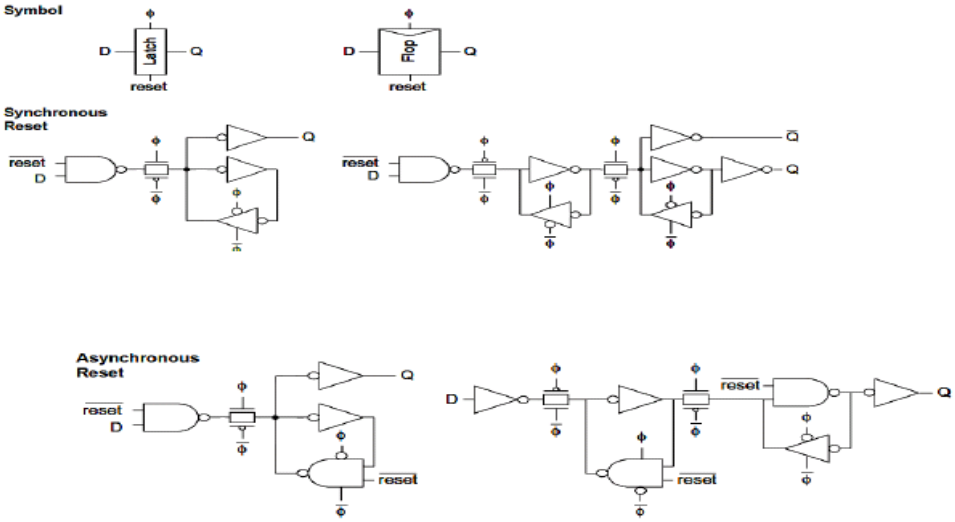


FIGURE 10.24 Resettable latches and flip-flops

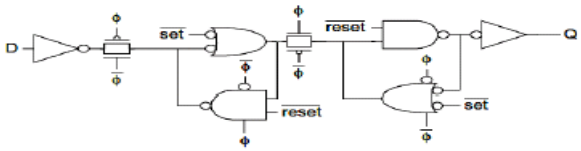


FIGURE 10.25 Flip-flop with asynchronous set and reset

Enabled Latches and Flip-Flops

Sequencing elements also often accept an enable input. When enable en is low, the element retains its state independently of the clock. The enable can be performed with an input multiplexer or clock gating, as shown in Figure 10.26.

The input multiplexer feeds back the old state when the element is disabled. The multiplexer adds area and delay. Clock gating does not affect delay from the data input and the AND gate can be shared among multiple clocked elements.

Moreover, it significantly reduces power consumption because the clock on the disabled element does not toggle. However, the AND gate delays the clock, potentially introducing clock skew.

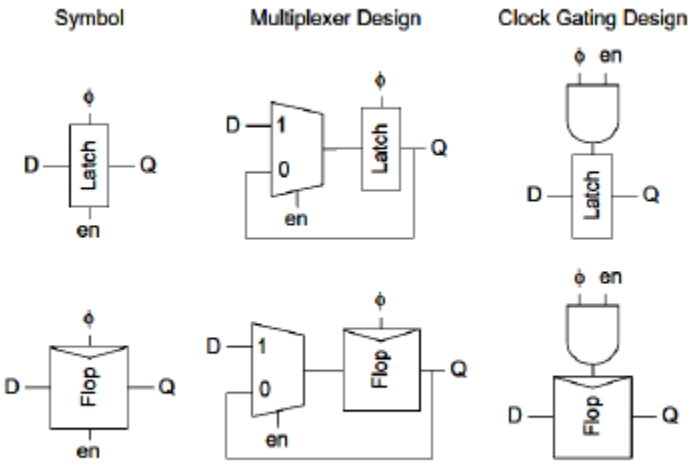


FIGURE 10.26 Enabled latches and flip-flops

5. Given the design concepts involved in Synchronous Pipelining and Asynchronous Pipelining. [CO3-H2]

TYPES OF PIPELINE SYSTEMS

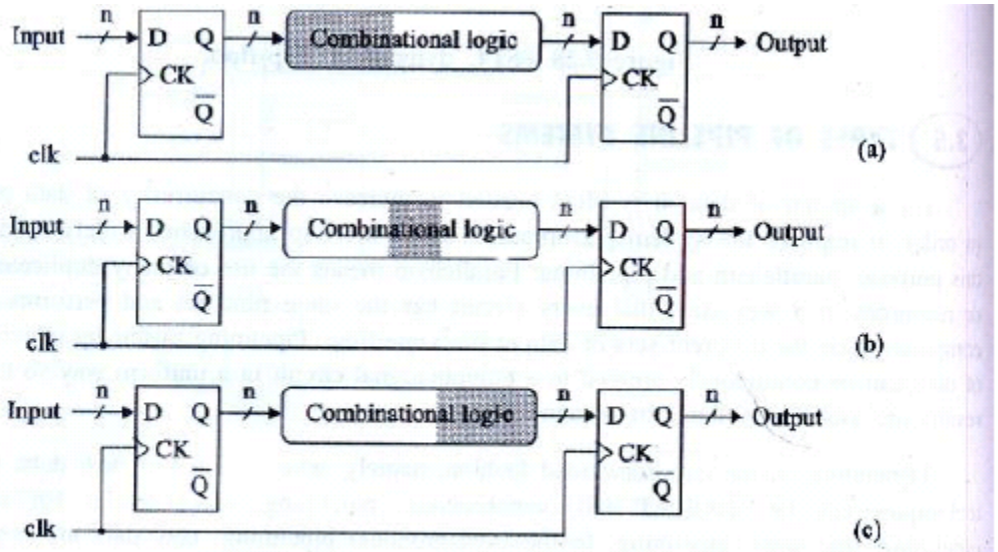
For a stream of data, it is often needed to increase the concurrency of data processing in order to improve the system performance. There are two approaches widely used to reach this purpose: parallelism and pipelining. Parallelism means the use of many duplicated circuits or resources in a way such that every circuit has the same function and performs the same computation on the different sets of data at the same time. Pipelining means by which a stream of data can be continuously applied to a computational circuit in a uniform way so that output results are yielded regularly in sequence.

Depending on the data-forwarded fashion, namely, when applied to new data, pipelining techniques can be classified into (synchronous) pipelining, asynchronous (or self-timed) pipelining, and wave pipelining. In the (synchronous) pipelining, new data are applied to a computational circuit in intervals determined by the maximum propagation delay of the computational circuit. In the asynchronous pipelining, new data are applied to a computational circuit in intervals determined by the average propagation delay of the computational circuit.

Asynchronous logic circuitry offers potential power saving and performance improvements at the cost of design complexity and a small area penalty. The traditional synchronous circuit design features a global clock that drives latches/flip-flops surrounding combinational logic that is used to perform a particular function. The clock speed is determined by the critical path through the system. In practical applications, (synchronous) pipelining can be simply implemented by partitioning the computational circuit into several smaller circuits with roughly the same propagation delay, separated by registers, called pipeline registers. These circuits are then synchronized by a global clock with a period greater th the worst-case propagation delay of the circuits and some overhead of the pipeline registers. Due to this reason, the (synchronous) pipelining is the most widely used approach among i three pipelining techniques. Also because of this, it is often referred to as pipelining for short.

Synchronous Pipelining

Pipelining is a system design technique that increases the overall system performance by partitioning a complex combinational logic circuit into many smaller circuits and uses a faster system clock. As shown in Figure 3.29, the computation of data within the combinational logic proceeds in a step-by-step fashion; namely, at time t = 0; the first part of the combinational logic performs its computation on the input data. At time t = tl; the second part of the combinational logic performs its computation on the output data from the first part, and so on.



re 3.29: An illustration of the pipelining principle: (a) t = 0; (b) t = t1; (c) t = t2.

The smallest allowable clock period T^ is determined by the following condition

$$T_{clk} \geq t_{pd} + t_q + t_{setup} + t_{skew}$$

where tpd is the propagation delay of combinational logic, tq and tsetup, are the clock to-Q delay and setup time of flip-flops, respectively, and tskew is the clock skew. The tq, tsetup; and tskew comprise the system overhead. For a stream of data, it is often to partition the combinational logic being used to compute a function into many (say, m) smaller logic pieces and then inserts a register between two logic pieces. This results in an m-stage pipeline system, as shown in Figure 3.30, where each stage performs a fraction computation of the complete function. The pipeline clock frequency fpipe is set by the slowest stage and is generally larger than the frequency used in the original nonpipelined system. For the ith allowable clock period Ti is determined by the following condition.

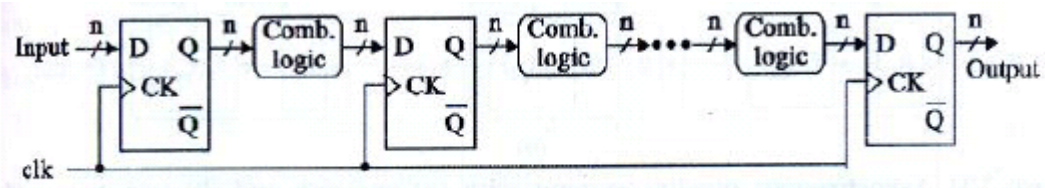


Figure 3.30 A pipeline system.

$$T_i \geq t_{pdi} + t_q + t_{setup} + t_{skew,\,i}$$

where tq, tsetup , and tskew:i comprise the pipeline overhead. The pipeline clock period for an m-stage pipeline system should then be chosen as

$$T_{pipe} = \max \{ T_i \mid i = 1, 2, 3 \ldots m \}$$

The pipeline clock frequency is equal to the reciprocal of Tpipe; namely, fpipe =1/ Tpipe

Asynchronous Pipelining

An asynchronous design is based on the concept of modular functional blocks intercommunicating using some communication protocols. The general block diagrams of asynchronous pipeline systems are shown in Figure 3.31. Figure 3.31(a) shows a scheme with one-way control and Figure 3.31(b) shows a scheme with two-way control, where HS means handshaking. The one-way control is also called a strobe control and the two-way control scheme is generally referred to as a handshaking control.

Regardless of which scheme is used and whether the underlying system is synchronous or asynchronous, any sequential logic must satisfy two constraints in order to work properly. These include physical timing constraint and event logical order. Physical timing constraint guarantees that each function unit and memory element has enough time to complete the specific operations. For example, in order to guarantee that the output data from a memory element is valid, the physical timing constraints, such as setup time and hold time of latches or flip-flops, must always be satisfied. Event logical order means that events in the system must occur in the logical order set by the designer. In a synchronous system, this is commonly achieved by using a clock to provide a time base for determining what and when to happen. In an asynchronous system, some other schemes, such as handshaking signals or protocols are deployed.

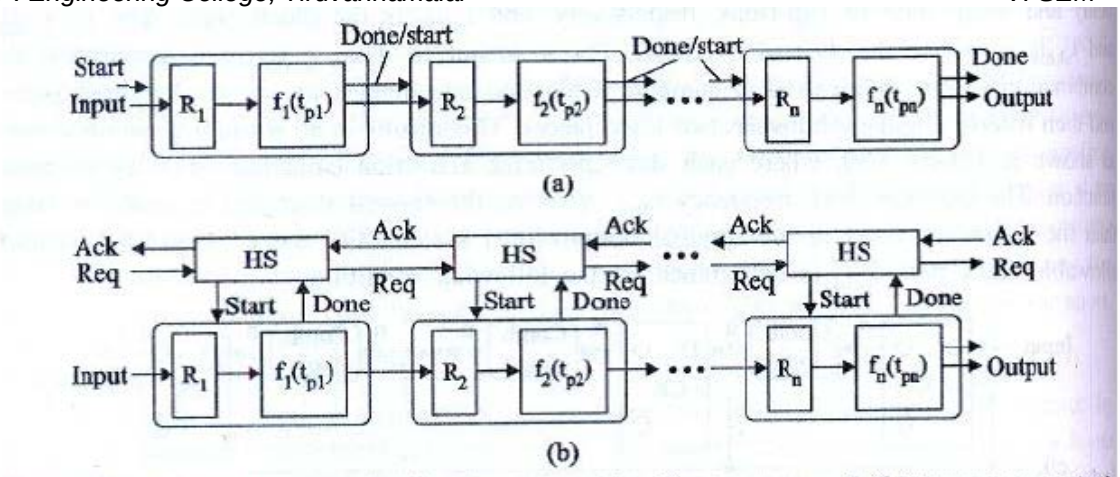


Figure 3.31 Asynchronous pipeline systems with (a) one-way and (b) two-way control schemes.

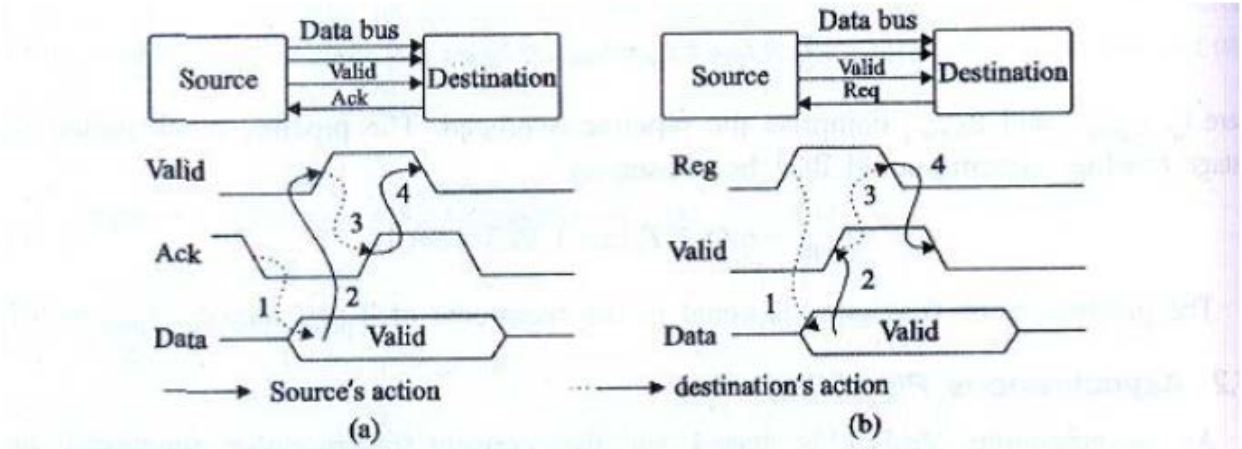


Figure 3.32: Timing diagrams of handshaking data transfers: (a) source-initiated

Conceptually, handshaking is a technique that provides a two-way control scheme for asynchronous data transfer. In this kind of data transfer, each transfer is sequenced by the edges of two control signals: req (request or valid) and ack (acknowledge), as shown in Figure 3.32. Generally speaking, in a handshaking data transfer, there are four events that proceed in a cyclic order. These events are ready(request), data valid, data acceptance, and acknowledge in that order. A handshaking control scheme can be initiated

Source-initiated transfer

The source-initiated data transfer is shown in Figure 3.32(a). The sequence of events is as follows.

1. Ready:

The destination device deasserts the acknowledge signal and is ready to accept the next data.

2. Data valid:

The source device places the data onto the data bus and asserts the valid signal to notify the destination device that the data on the data bus is valid.

3. Data acceptance:

The destination device accepts (latches) the data from the data bus and asserts the acknowledge signal.

4. Acknowledge:
The source device invalidates data on the data bus and deasserts the valid signal.
Due to the inherent back-and-forth operations of the scenario, the asynchronous data transfer described above is called the handshaking protocol or handshaking transfer.

Destination-initiated transfer
The destination-initiated data transfer is shown in Figure 3.32(b). The sequence of events is as follows.

1. Request:
The destination device asserts the request signal to request data from the source device.

2. Data valid:
The source device places the data on the data bus and asserts the valid signal to notify the destination device that the data are valid now.

3. Data acceptance:
The destination device accepts (latches) the data from the data bus and deasserts the request signal.

4. Acknowledge:
The source device invalidates data on the data bus and deasserts the valid signal to notify the destination device that it has removed the data from the data bus. by either a source device or destination device.

6. Discuss in detail about and static dynamic RAM. **[CO3-H2]**

Memory architecture and memory control circuits
In modern digital systems, a large part, probably up to a one-half area, of a digital chip is occupied by various memory modules. These memory modules are used to store and retrieve large amounts of information at high speeds.
The semiconductor memory can be classified in terms of the type of data access and the capability of information retention. According to the type of data access, semiconductor memory can be subdivided into serial access, content addressable, and random access. The data in serial-access memory can only be accessed in a predetermined order and mainly contains shift registers and queues. Content-addressable memory (CAM) is a device capable of parallel search; namely, it behaves like a lookup table in which all entries can be searched in parallel. Random-access memory (RAM) is a memory device where any word can be accessed at random at any time. The random-access memory can be further categorized into read/write memory and read-only memory.
Read/write memory contains two major types: static RAMs (SRAMs) and dynamic RAMs (DRAMs). Read-only memory (ROM), as its name implies, is only allowed to retrieve but not allowed to modify the stored information. The contents of ROM can be committed using a photolithographic mask during being manufactured or programmed in laboratories.
According to the capability of information retention, semiconductor memory may also be cast into volatile and nonvolatile memories. The volatile memory, such as static RAM and dynamic RAM, will lose its information once the power supply is interrupted while the nonvolatile memory, such as the ROM family, ferroelectric RAM (FRAM), and magnetoresistance RAM (MRAM), still retain their information even when the power supply is removed.

Memory Classification

As stated, about 30% of the semiconductor business is due to memory devices at present. Semiconductor memory capable of storing large amounts of digital information is essential to all digital systems, ranging from 4-bit to 64-bit microcomputer systems, even cluster computers. The semiconductor memory can be classified in accordance with the type of data access and the capability of information retention.

Types of Data Access

According to the type of data access, semiconductor memory is categorized into serial access, content addressable, and random access, as shown in Figure 3.43. Serial-access memory mainly contains shift registers and queues. The former can be further subdivided into serial-in-parallel-out (SIPO) and parallel-in-serial-in (PISO), while the latter contains first-in first-out (FIFO) and first-in Last-out (FILO). A FIFO is generally called a queue or a buffer and a FILO is referred to as a stack.

Random-access memory (RAM) is a memory device in which any word can be accessed at random in a constant time. The random-access memory can be classified into read/write memory and read-only memory. Read/write memory can be further subdivided into two types: static RAMs (SRAMs) and dynamic RAMs (DRAMs). SRAM is mainly used as the main memory for those computer systems needing a small memory capacity, and as the cache memory for large computer systems, such as desktop computers and servers. DRAM is used as the main memory for those computer systems requiring a large memory capacity.

The SRAM cell consists of a bistable device accompanied with two access transistors and thus stores its information in a form of circuit state; the DRAM cell comprises an access transistor and a capacitor, and retains information in the form of charge. Due to the leakage current of the access transistor, the capacitor (a soft node) in a D71AM cell is gradually losing its charge and eventually its information. To remedy this, a DRAM cell must be read and rewritten periodically, even when the cell is not accessed. This process is known as refresh.

Content-addressable memory (CAM) is a device with the capability of parallel search; namely, it behaves like a lookup table in which all entries can be searched in parallel. A CAM cell is fundamentally a SRAM cell to store information with the addition of some extra circuits to facilitate the operation of parallel interrogation and indicate the result. CAM proceeds the reverse operation of SRAM; namely, it receives as an input data and outputs a matched address when the data matches one in the CAM.

Read-only memory (ROM), as its name implies, is only allowed to retrieve but not allowed to modify the stored information. The contents of ROM are programmed using a photolithographic mask during manufacturing. Hence, this kind of ROM is also called mask ROM. Another type of ROM that can be programmed in a laboratory is called a programmable ROM (PROM). Programmable ROM can be further categorized into fuse ROM, erasable PROM (EPROM), electrically erasable PROM (EEPROM), and Flash memory. Fuse ROM can "be only programmed once and therefore belongs to the type of one-time programmable (OTP) devices. The contents of an EPROM device can be erased by exposing the device under an ultraviolet light. Such a device is also referred to as an UV-exposure EPROM or UV-EPROM to emphasize the erasing feature of using the UV-exposure method. Another feature of UV-EPROM is that the erase

operation is done on the entire chip in an off-board fashion, thereby erasing the data of the whole chip. The contents of an EEPROM are erased by a high-voltage pluse in a

bundle of bits, say one byte or a block of bytes. Due to the use of voltage pulses the contents of an EEPROM can be erased partially or entirely on board. Flash memory is similar to EEPROM in which data in the block can be erased by using a high-voltage pulse.

Memory organization

The typical memory organization is illustrated in Figure 3.44. It comprises a row decoder, a column decoder/multiplexer, a memory core, sense amplifiers/drivers, and a piece of control logic. The row decoder takes an m-bit address and produces 2m word-line enable signals. The column decoder takes an n-bit address and produces 2n bit-line select signals. The memory core is composed of memory cells. These memory cells are arranged into a two-dimensional array, consisting of horizontal rows and vertical columns. Each individual cell can be accessed in a random order within a constant time, independent of physical location. Each memory cell is connected to a horizontal linecalled a word-line, driven by the row decoder, and two vertical lines, called bit-line and complementary bit-line, respectively, used to access the data of the cell. Each time, 2' columns in one row may be accessed, where i is a positive integer or zero, dependent on the organization of memory. The signals of selected columns are amplified or driven by the serse amplifier/drivers according to whether the operation is read or write.

The control logic receives three inputs chip enable (CE), output enable (OE) and write enable (WE), and generates the required internal control signals. The chip select controls the operations of an entire chip. Output enable (OE) asserts the read operation by enabling the output buffer while write enable (WE) asserts the write operation. The data bus is in a high-impedance whenever the chip enable is deasserted, regardless of the values of output enable (OE) and write enable (WE).

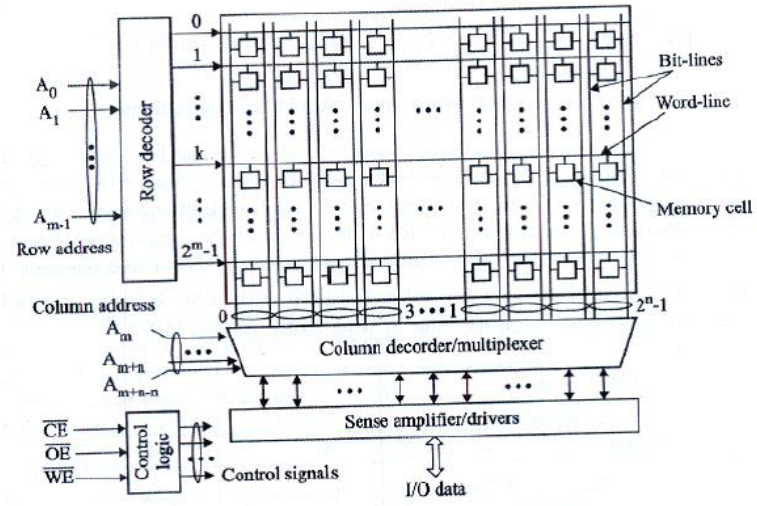


Figure 3.44 The typical memory structure

Memory Access Timing

Generally, SRAM devices can be subdivided into asynchronous and synchronous according to whether the access operation is triggerred by the external clock signal or not. For platform-based designs, asynchronous SRAM devices are widely used and for

cell-based designs, synchronous SRAM are more popular. In the following, we only describe the typical access timing of synchronous SRAM devices briefly.

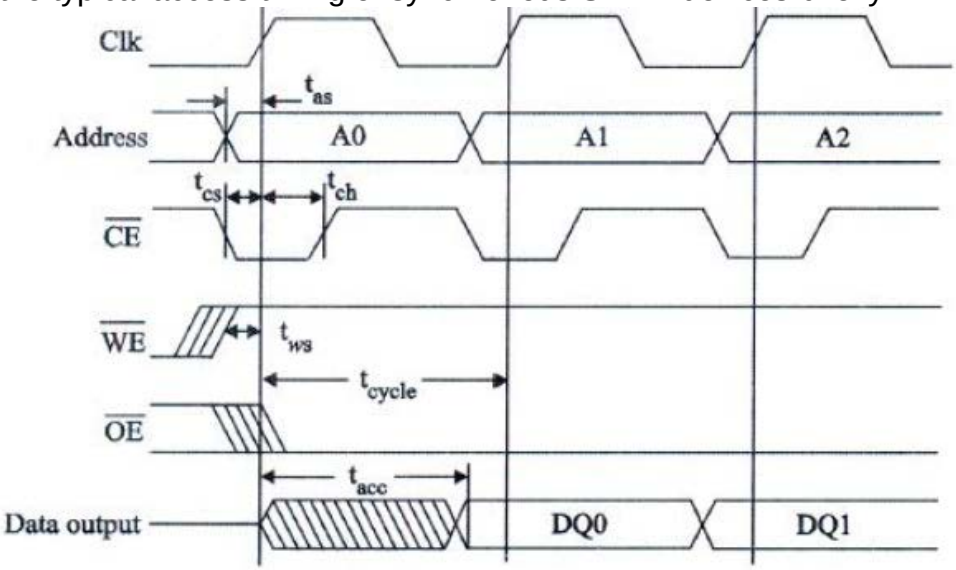


Figure 3.45 Typical synchronous SRAM read-cycle timing

The read-cycle timing of typical synchronous SRAM is illustrated in Figure 3.45. The read operation is synchronous and triggered by the rising edge of the clock, clk. Address, chip enable (CE),write enable (WE), and output enable (OE), are latched by the rising edge of the clock. To properly access data from the SRAM device, all of these input signals must be subject to individual setup times (tas, tcs tws) and hold time (thold For a read operation, the chip enable (CE) must be set to a low value. Once the chip enable is active, the SRAM device enters the read mode when the value of write enable (WE) is high

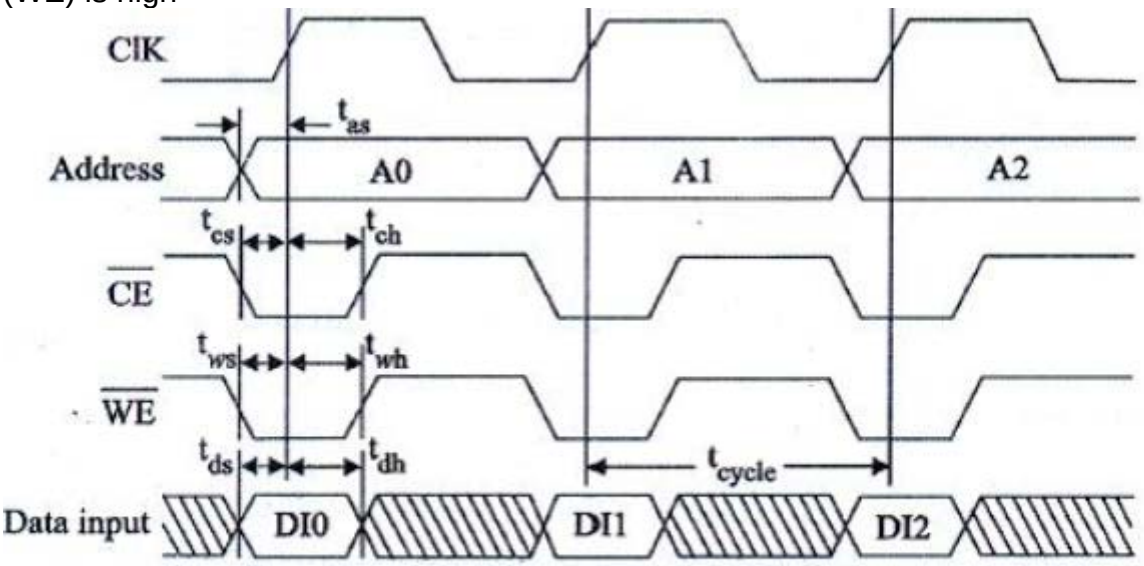


Figure 3.46 Typical synchronous SRAM write-cycle timing

During the read mode, data is read from the memory location specified by the address and appears on the data output bus after a read access time (tacc) if output enable (OE) is asserted. The read access time is the time counted from the positive edge of clock, elk, to the stable data appearing at the output data bus.

The write-cycle timing of typical synchronous SRAM is illustrated in Figure 3.46. Like in the read mode, the write operation is synchronous and triggered by the rising edge of the clock, elk. Address, chip enable (C£),write enable (WE), and output enable (OE), are latched by the rising edge of the clock. To properly write data into the SRAM device, all of these input signals must be subject to individual setup times (tas tcs tws and tdh) and hold times. **For** a write operation, the chip enable (CE) and write enable (WE) must be set to a low value. During the write mode, data is written into the memory location specified by the address.

Static Random-Access Memory (SRAM)

As stated, the feature of random-access memory is that any memory cell can be accessed at random in a constant time. In this section, we deal with the major components of SRAM, including memory core, and operation of SRAM.

Basic RAM Cell Structures

Two types of SRAM cells are given in Figure 3.47. They are 4T- and 6T-SRAM cell structures. We can see from the Figure that a basic SRAM cell is a bistable device formed by two back-to-back connected inverters along with two_access transistors being separately connected to a pair of complementary bit-lines, Bit and Bit. The 4T-SRAM cell is shownn Figure 3.47(a) and the 6T-SRAM cell is exhibited in Figure 3.47(b). Their structures are roughly the same expect for the pull-up devices. The pull-up devices are two resistors in the 4T-SRAM cell and are two pMOS transistors in the 6T-SRAM cell.

The 4T-SRAM cell has a more compact cell size since the resistors (formed from undoped polysilicon) can be stacked on top of the cell, using double-polysilicon technology. Thus, the cell size is reduced from six to four transistors. However, due to the use of passive pull up resistors, there exists a direct current between power rails since one output of two inverters is at low-level voltage at any time, leading to an amount of power dissipation that cannot be neglected. To reduce the power dissipation, high-value load resistors may be used. Nevertheless, this would result in worse noise margins and slower operation, speed. Hence, there is a trade-off between power dissipation and noise margins as well as switching speed.

The 6T-SRAM cell has the lowest static power dissipation and offers superior noise margins and switching speed as well, as compared to 4T-SRAM cell. In addition, it is compatible with most widely used CMOS logic processes at present. Hence, it finds the widespread use in modern CMOS SRAM modules. In the following, we will focus on this type of SRAM cell.

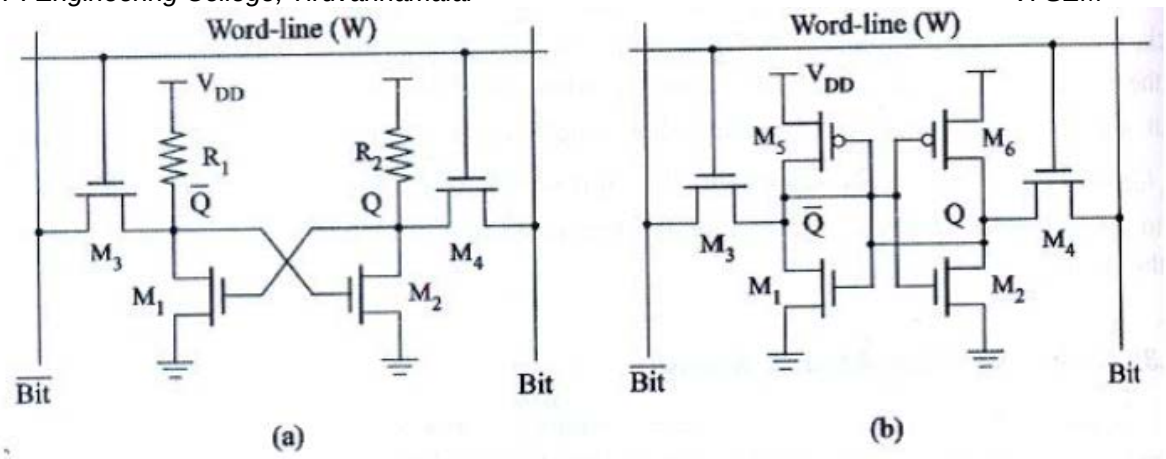


Figure 3.47 (a) 4T-and (b) 6T-SRAM cell structures.

In practical designs, both transistors M3 and M4 usually use minimum geometry devices.

The sizes of transistors M1 and M2, M5 and M6 are determined by the following two basic requirements. First, the data-read operation should not destroy information in the memory cell. Second, the memory cell should be allowed modification of the stored information in the data-write operation. Based on these, the strengths of the three transistors on one side from strongest to weakest are pull-down, access, and pull-up in sequence.

Read-Cycle Analysis

For the read operation, both bit-lines are usually precharged to (1/2) **VDD** or **VDD,** depending on how the sense amplifier is designed. In the following we assume that both bit-lines are precharged to **VDD** before the beginning of the read operation. To avoid destroying the information in the memory cell, the **V**q must be less than the inverter threshold voltage **VTH** after the access transistors are turned on: namely the following relationship must be held

$$\frac{R_{M1}}{R_{M1}+R_{M3}+}V_{DD} \le V_{th} = \frac{1}{2}V_{DD}$$

To analyze the read operation of the 6T-SRAM cell in more detail, assume that the memory cell stores a **1;** namely, **VQ = VDD** while **Vq = 0V.** The equivalent circuit is given in Figure **3.48.** To access the stored value of the memory cell, we raise the voltage of the word-line to **VDD** to turn on both access transistors, M3 and M4 as well as raise the voltages of both bit-lines to **Vqq**

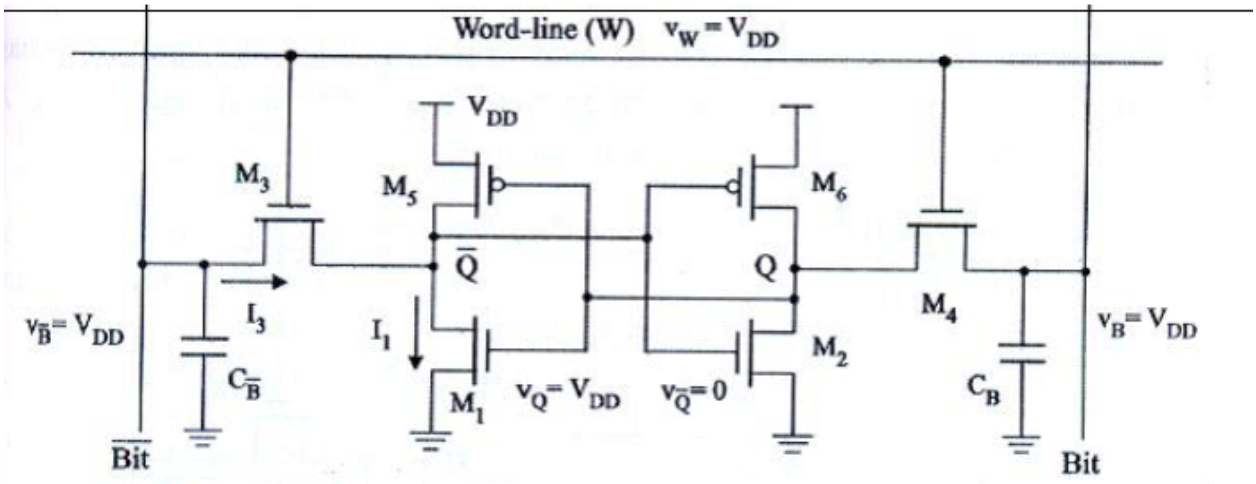


Figure 3.48 The read cycle of a 6T-SRAM cell structure.

If the voltage **VB** of Bit is still at **VDD** after the access transistor is turned on, then nMOS
transistor M3 will be in its linear region of operation and nMOS transistor **M3** in the saturation region. From Figure **3.48,** we obtain

$$I_{D1} = \mu_n C_{ox}\left(\frac{W_1}{L_1}\right)\left[(V_{DD} - V_{Tn})\,V_{DS1} - \frac{1}{2}V_{DS1}^2\right]$$

$$I_{D3} = \frac{1}{2}\mu_n C_{ox}\left(\frac{W_3}{L_3}\right)(V_{DD} - V_{Tn} - V_{DS1})^2$$

The size of transistor **M3** determines the current to discharge the complementary bit-line capacitor Cg. As a result, the actual size is determined by the desired transition time CAg for the bit-line voltage **VB** to be changed by an amount of A **V,** about **200** mV, which can be detected by a sense amplifier. The transition time can be expressed as

$$t_{\Delta \overline{B}} = C_{\overline{B}}\frac{\Delta V_{\overline{B}}}{I_{D3}}$$

where, ID3 is the drain current of the access transistor M3 and is the capacitance of bit-line Bit.

Write-Cycle Analysis

The write operation [Figure 3.49] is accomplished by forcing one bit-line to **VDD** while keeping the other at about 0 V. For instance, to write a 0 to the memory cell, we set the bit-line voltage **VB** to 0 V and the complementary bit-line voltage **V-** to **VDD.** If **Q** is 1 and we wish to write a 0 into the memory cell, we need to pull **VQ** below the threshold voltage of the inverter composed of M5 and **MX** so as to guarantee the inverter to change its state. As a consequence, the following relationship must be satisfied.

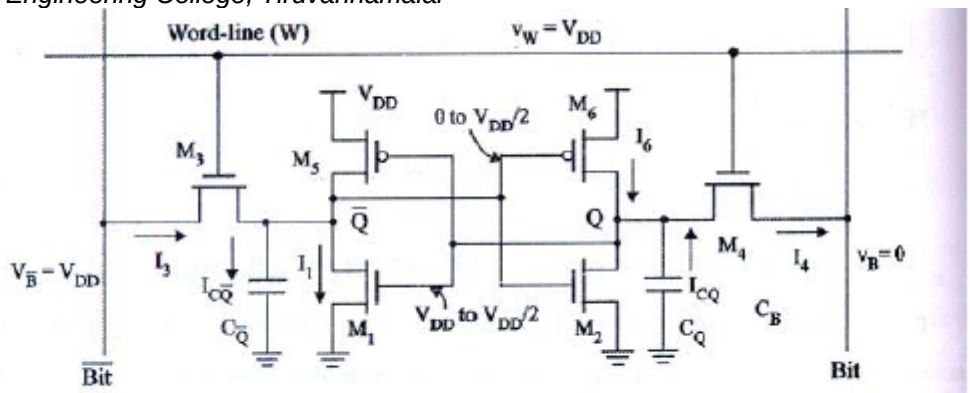


Figure 3.49 The write cycle of a 6T-SRAM cell structure (Write a logic 0 to the cell)

$$\frac{R_{M4}}{R_{M4}+R_{M6}}V_{DD} \le V_{th} = \frac{1}{2}V_{DD}$$

where RM4andRM6 are the on-resistances of nMOS and pMOS transistors M4andM6 respectively.

To analyze the write cycle in more detail, consider the situation that **VQ** is equal to **VDD** and we want to write a 0 into the memory cell. As shown in Figure 3.49 the voltage **VB** is forced to 0 V. Once the access transistor **M4** is turned on, **CQ** starts to discharge towards the ground level through the access transistor M4. When **VQ** = **VTH = ½ VDD** inverter composed of transistor **M1** and **M5** starts to switch its state. Meanwhile, transistors M4andM6 operate in their linear regions. From Figure 3.49, we obtain

$$I_{D4} = \mu_n C_{ox}\left(\frac{W_4}{L_4}\right)\left[(V_{DD} - V_{Tn})V_{th} - \frac{1}{2}V_{th}^2\right]$$

$$I_{D6} = \mu_n C_{ox}\left(\frac{W_6}{L_6}\right)\left[(V_{DD} - |V_{TP}|)V_{DD} - V_{th}) - \frac{1}{2}(V_{DD} - V_{th})^2\right]$$

The ratio of W6 to W4 can be obtained by using the fact that both transistors M6 and M4 are in series and hence their currents must be equal. Like the read operation for deep-submicron processes, short-channel current equations are often used instead

DESIGN ORGANIZATION OF DYNAMIC RANDOM-ACCESS MEMORY(DRAM)

Dynamic random-access memories (DRAMs) due to high integration density have been widely using in a wide variety of computer systems over the past decades to form a hierarchical memory system, thereby reducing the cost of memory system profoundly.

Cell Structure

In this subsection, the widely used dynamic memory cell based on a 3T-circuit structure and modern IT-dynamic cell structures have been explained.

3T-DRAM Cell

The first widely used DRAM cell is **a** three-transistor (3T) structure, as shown in Figure 3.53. The information is stored in the charge form in parasitic capacitor Cs. Transistors

Mj and M2 act as access transistors during write and read operations, respectively. Transistor

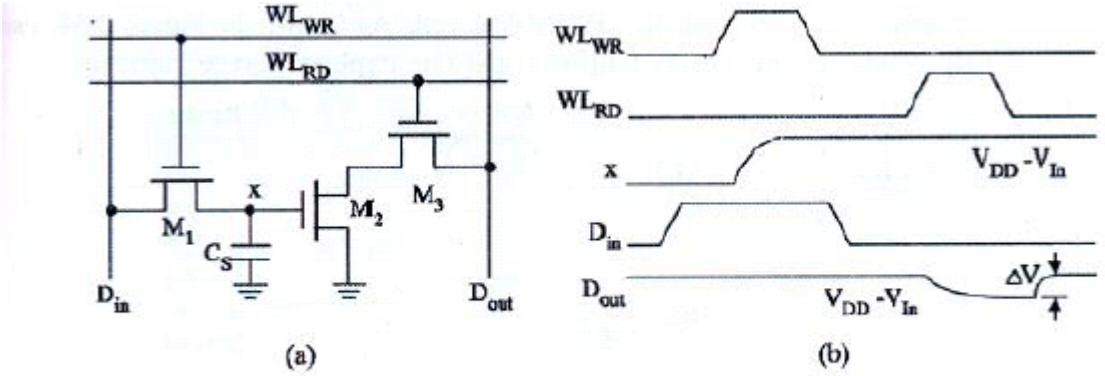


Figure 3.53 The 3-T DRAM Cell (a) typical circuit structure and (b) timing of a 3T-DRAM cell.

M2 is the storage transistor and is turned on or off, according to the amount of charge stored in capacitor **CS.**

Both read and write operations of a **3T**-DRAM cell are proceeded in a two-phase process, During the first phase, both bit-lines **DM** and Dout are precharged to **VDD - VTN.** During the second phase, the actual read or write operation is executed. To read a **3T**-DRAM cell, the read word-line **(WLRD)** is activated to turn on the read access transistor M3. The storage transistor M2 turns on if there is charge stored in the capacitor **CS** and remains off otherwise. When the storage transistor **M2** turns on, the voltage of bit-line Dout is dropped from **VDD - VTN** by an amount of A **V.** This amount of voltage change cause the sense amplifier to restore into a valid logic value. On the other hand, When the storage transistor M2 stays off, the voltage of bit-line Dout would not be dropped. The sense amplifier yields another logic value.

To write a logic value to a **3T**-DRAM cell, the write word-line **(WLWR)** is enabled to turn on the write access transistor Mj. The storage capacitor **CS** is then charged by the voltage at bit-line Din. The timing is exhibited in Figure **3.53**(b). It should be noted that due to the body effect of the nMOS access transistors, the maximum voltage at the storage capacitor is only **VDD - VTN.**

The important features of **3T**-DRAM cell are as follows. First, there is no constraint on device ratios. Second, read operations are nondestructive. Third, the voltage value stored at the storage capacitor is at most **VDD - VTN** The drawback of the **3TJDRAM** cell is that it needs three transistors and one capacitor.

1T-DRAM Cell

Nowadays, the IT-DRAM cell is the most widely used storage structure in the DRAM industry because it needs less area than the **3T**-DRAM cell. As shown in Figure **3.54,** one IT-DRAM cell only comprises one access transistor and capacitor

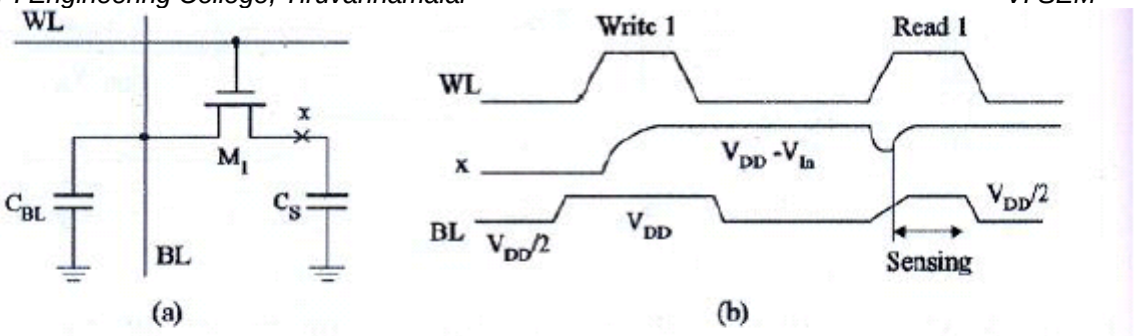


Figure 3.54 The 1T-DRAM cell (a) typical circuit structure and (b) timing of a 1T-DRAM cell

Like a 3T-DRAM cell, both read and write operations of a IT-DRAM cell are also proceeded in a two phase process. During the first phase, the bit-line (BL) is precharged to improve noise immunity and reduce power dissipation. During the second phase, the actual access operation is carried out. To read the information stored in a IT-DRAM cell, the word-line (WL) is enabled to turn on the access transistor. This causes charge redistribution to take place between the bit-line parasitic and storage capacitors. Assume that the capacitanc of the bit-line is CBL. The voltage at the storage capacitor Cs is VDD - VTn when it stores a logic 1 and is 0 V when it stores a logic 0. Then, the voltage change of the bit-line can be expressed as follows.

$$\Delta V_{bit1} = V_{after} - V_{BL}$$
$$= \left(\frac{C_S}{C_{BL} + C_S} \right) \left(\frac{1}{2} V_{DD} - V_{Tn} \right)$$

when the storage capacitor Cs stores a logic 1 and is

$$\Delta V_{bit0} = V_{after} - V_{BL}$$
$$= -\frac{1}{2} V_{DD} \left(\frac{C_S}{C_{BL} + C_S} \right)$$

when the storage capacitor Cs stores a logic 0. The voltage swing of a bit-line is usually small, typically ranging from 20 to 150 mV, depending on the ratio of bit-line capacitance to storage capacitance. The actual required voltage swing is set by the sensitivity of a sense amplifier.

Structures of DRAM Memory Array

The trench and stacked capacitor cells cross sectional view is shown in Figure 3.55 (which is widely deployed in DRAM cell).The memory array structure of DRAM is shown in Figure **3.55,** where the bit-lines of the storage array are split in half so that equal

amount of capacitance is connected to each side of the sense amplifier. This is known as an open-bit architecture. Due to the difficulty of detecting a small voltage change in a single-ended sense amplifier, a dummy cell is employed for each side of the sense amplifier to serve as a reference voltage.

The read operation is as follows. The left word-lines use cell A as a reference while the right word-line use cell **B** as a reference. The bit-lines and reference cells are precharged to **VDD/2** by enabling the precharge signal φp. Now, assume that the word-line R127 is activated to read its corresponding cell. To do this, the dummy cell B on the opposite side of the sense amplifier is also enabled by raising the control signal B in order to balance the common-mode noise arising from the circuit. The resulting small differential voltage difference between two bit-lines on each side of the sense amplifier is then amplified into a valid logic value 0 or **1.**

Nowadays, a more popular memory array structure of a IT-DRAM is referred to as a folded bit-line architecture, as shown in Figure 3.57(a). The corresponding layout of the cell is given in Figure 3.57 (b). In this architecture, each bit-lines connects to only a half number of cells and two adjacent bit-lines are grouped as pairs as inputs to the same sense amplifier. When a word-line is activated, one bit-line of a sense amplifier switches while the other serves as the quiet reference. Hence,the dummy cells are no longer needed. Like the open-bit architecture, the quiet bit-line of the sense amplifier functions as the dummy cell and helps reject the common-mode noise likely coupling equally onto the two adjacent bit-lines. 45

One disadvantage of folded bit-line architecture is that it requires more layout area, about 33%, compared to the open-bit architecture. To further reduce the common mode noise, the twisted bit-line architecture used in SRAM may be applied.

Unit IV

Designing Arithmetic Building Blocks

Part A

1. How path can be implemented in VLSI system? [CO4- L1]

A data path is best implemented in a bit –sliced fashion. A single layout is used respectively for every bit in the data word. This regular approach eases the design effort and results in fast and dense layouts

2. Comment on performance of ripple carry adder. [CO4-L1]

A ripple carry adder has a performance that is linearly proportional to the number of bits.

Circuit optimizations concentrate on reducing the delay of the carry path. A number of circuit topologies exist providing that careful optimization of the circuit topology and the transistor sizes helps to reduce the capacitance on the carry bit.

3. What is the logic of adder for increasing its performance? [CO4-L1]

Other adder structures use logic optimizations to increase the performance (carry bypass,carry select, carry lookahead). Performance increase comes at the cost area.

4. What is multiplier circuit? [CO4-L1]

A multiplier is nothing more than a collection of cascaded adders. Critical path is far more complex and optimizations are different compared to adders.

5. Which factors dominate the performance of programmable shifter? [CO4-L2]

The performance and the area of a programmable shifter are dominated by the wiring.

6. What is meant by data path? [CO4-L1]

A datapath is a functional units, such as arithmetic logic units or multipliers, that perform data processing operations, registers and buses. Along with the control unit it composes the central processing unit.

7.Write down the expression for worst-case delay for RCA. [CO4-L1]

t = (n-1)tc+ts

8.Write down the expression to obtain delay for N-bit carry bypass adder. [CO4-L1]

tadder = tsetup +Mtcarry +(N/M-1)tbypass +(M-1)tcarry + tsum

9. Define Braun multiplier. [CO4-L1]

The simplest multiplier is the Braun multiplier. All the partial products are computed in parallel, and then collected through a cascade of Carry Save Adders. The completion time is limited by the depth of the carry save array, and by the carry propagation in the adder. This multiplier is suitable for positive operands.

10. Why we go to Booth's algorithm? [CO4-L1]

Booth algorithm is a method that will reduce the number of multiplicand multiples. For a given number of ranges to be represented, a higher representation radix leads to fewer

digits.

11. List the different types of shifter. **[CO4-L1]**

Array shifter
Barrel shifter
Logarithm shifter

Part B

1. Explain the datapath and control unit organization. **[CO4-H1]**

Definition for datapath circuits

- Many digital systems are often partitioned into two parts: a datapath and a control unit.
- The datapath performs the required operations to the input data whereas the control unit schedules the operations being carried out in the datapath in a proper order.
- In CMOS circuit the data manipulation for n-bit data is generally implemented using n-identical circuits. A datapath is the data processing section of a processor.
- It consists of several multiple-bit data path elements or operators such as arithmetic units (adder, multiplier, shifter, comparator) or logical operators (AND, OR, NAND) arranged horizontally and connected with busses.
- Control signals connect to the datapath at the top and bottom. A typical data path is shown in Figure.

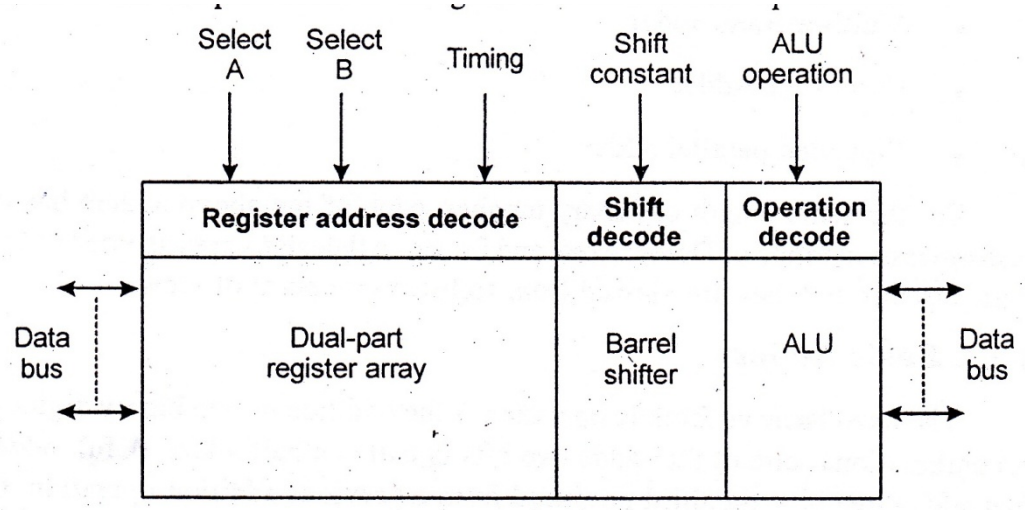


Figure 4.1: Data path structure

The advantages of datapath operators are:

- To implement the logic function using n-identical circuits

- Data may be arranged to flow in one direction, while any control signals are introduced in the orthogonal direction to the data flow.

Bit-sliced datapath

- Bit-sliced datapath organization is shown in Figure 4.1 (b) where large size data is segmented or sliced into smaller bit size.
- Instead of operating on single-bit, the data in a processor can be arranged in nibble, byte or word based fashion.
- For example, a 32-bit processor operates on data words that are 32-bits wide. Since the same operation has to be performed frequently on each bit of the data word, the datapath consists of 32 bit slices, each operating on a single bit-hence it is called **bit sliced**.

Bit slices are either identical or resemble a similar structure for all bits. Datapath allows area optimization in the layout by incorporating regular routing strategy into the operator cell design.

Data path automatically takes care of interconnect between the cells with the following advantages:

- Regular layout produces predictable and equal delay for each bit.
- Interconnect between cells can be built into each cell

2. Define binary adder and state the static CMOS full adder and its types with speed and area tradeoff. (or) Briefly explain the types of full adder such as Mirror adder, Transmission gate (TG) adder, Manchester adder. [CO4-H2]

Introduction to adders

In the majority of Digital Signal Processing (DSP) applications the critical operations are the addition, Implication and accumulation. Addition is an indispensable operation for any digital system or control system.

Therefore a fast and accurate operation of a digital system is greatly influenced by the performance of the resident adders. Adders are also very significant component in digital systems because of their widespread use in other basic digital operations such as subtraction, multiplication and division.

The Binary Adder Definition

An adder is a combinational circuit that performs the arithmetic sum of three bits: A, B and a carry in (Q), from a previous addition produces the corresponding SUM, and a carry out (CARRY).

The Boolean expression for SUM and CARRY are given in Eq. (4.1)

$SUM = A \oplus B \oplus C$

$CARRY = AB + AC_i + BC_i$ ---------- (4.1)

This module can be constructed using the intermediate signal called generate (G) and propagate (P).

The equation of SUM and CARRY can be formulated using the Table 4.1.

Table 4.1 Truth table for full adder

Row	A	B	C_i	SUM	CARRY
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

In rows 0 and 1, the carry out (CARRY) is 'zero' irrespective of carry in called killed state. Similarly in rows 6 and 7, the carry out is 'one' which is independent of carry in.

These signal characteristics can be defined as carry generate combination (G). In rows 2, 3, 4 and 5 the carry out is same as carry in, therefore these intermediate signal characteristics can be elucidated as propagate combination. The kill state can be expressed as

$Ki(kill) = \overline{Ai + Bi} = \bar{A} . \bar{B}$ (row 0 and 1) ---------- (4.2)

This term gets its name from the fact that if K_t = 1, then P_i and G_i, are zero, so that the next state carry $C_{i+1} = 0$ thus it "kills" the carry-out bit.

The generate block can be realized using the expression

$G_i = A_i B_i$ for I =0,1,2 ---------- (4.3)

Similarly the propagate block can be realized using the expression

$Pi = Ai \oplus Bi$ for i = 0,1,2…. ---------- (4.4)

The carry output of the (i-l)th stage is obtained from

C_i (CARRY) = $G_i + P_i C_{i-1}$ for i= 0,1,2 ---------- (4.5)

The sum output can be obtained using

S_i (SUM) = $P_i \oplus C_i$ for i = 0,1,2,…. ---------- (4.6)

In Eq. (4.4 and 4.5) the generate signal G and propagate signal P depends upon the inputs ***A_i and B_i*** are independent of carry in C_i

The Full Adder: Circuit Design Consideration

1. Static CMOS Adder Circuit

Conventional static CMOS full adder can be realized using pull up and pull down with NMOS and PMOS networks utilizing 28 transistors.

This type of full adder can be realized using Eq. (4.1) with a slight modification in SUM generation so as to reduce the number of transistors. The modified equation is given in Eq. (4.7).

This modification could take two advantages; the former provides sharing of logic between SUM and CARRY module to avoid the latency in carry generation. The latter provides the advantage of implementing the SUM expression in terms of OR and AND gates instead of XOR gate which will reduce the power consumption of full adder.

The general block representation of full adder with non-inverting and inverting inputs and outputs is shown in Figure 4.2.i

$CARRY = AB + AC_i + BC_i$

$SUM = ABC_i + CARRY\ (A+B+C_i)$ ---------- (4.7)

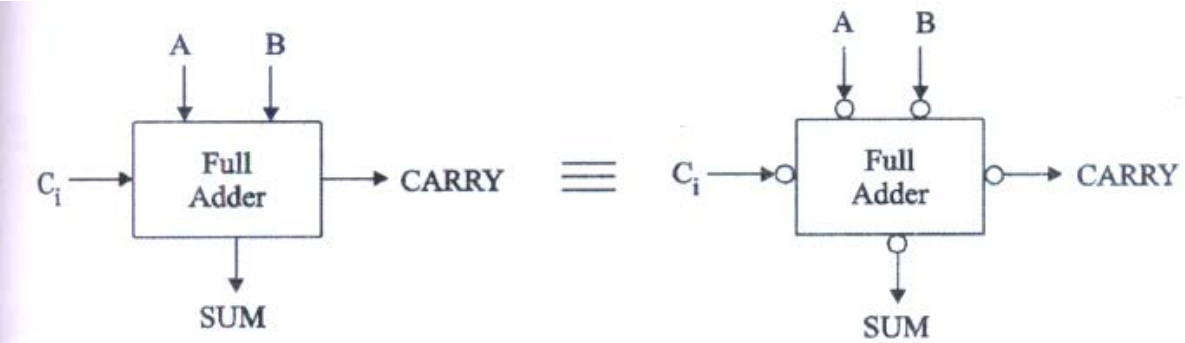
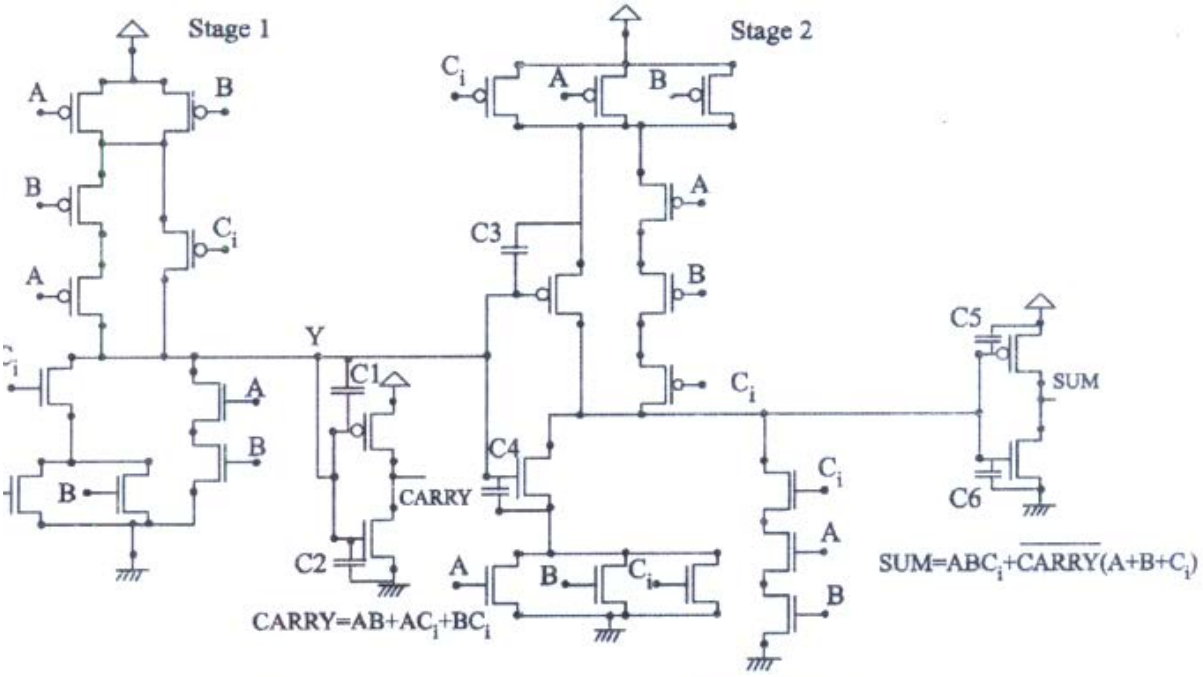


Figure 4.2 The general block representation of full adder with non-inverting and inverting inputs and outputs



(a) Transistor representation of CMOS full adder

The primary advantages of this static CMOS full adder are

- It has full-swing outputs that increase noise margin and reliability.
- It functions well at low power supply voltages because it does not have threshold loss problem.
- The adder can be manufactured by a basic conventional CMOS process with slight mobility degradation.

The major shortfalls associated with this static CMOS full adder are

1. Large PMOS transistor in pull up network result in high input capacitances, which cause high delay and dynamic power.
2. The propagation delay is high in this static CMOS full adder realization.

$$D_{CRITICAL} - SUM = D_{CARRY} + D_{INV} \text{ ---------- (4.8)}$$

3. The intrinsic load capacitance of the CARRY signal is high which is contributed by the gates capacitances C_1, C_2, C_3, C_4, C_5 and C_6 (refer Figure 4.3 (a)), diffusion capacitances in stage 1 and stage 2 and wring capacitance

$$C_{CARRY\text{-}} load = C_{gate} + C_{diff} + C_{wire} \text{ ---------- (4.9)}$$

4. This adder realization consumes a large area and it could possible to fabricate in twin-tub CMOS process to avoid mobility degradation and latch up condition.

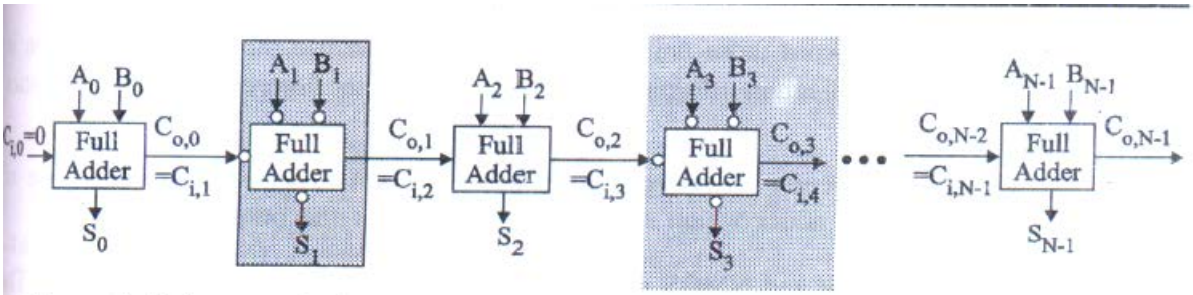


Figure 4.4 The speed of ripple carry adder improved by exploiting the inverting property

However the performance of this static CMOS full adder can be improved by lowering the logical effort of PMOS stack in the carry generation sub-circuit and input ordering scheme.

In lowering the logical effort of PMOS stack the rise time and fall time of PMOS transistor reduces thereby increases the driving capability of CARRY logic.

Moreover reducing the logical effort of PMOS will reduces the ON-resistance of PMOS stack. By deploying the input ordering scheme the body effect and propagation delay is reduced.

For this case the inner input carry in signal C, is connected closer to the output node and the outer input A and B connected as the input closer to the power-supply rail, either VDD or ground.

By this signal arrangement the body effect due to C, is reduced because of activating this signal as late as possible will discharge the intermediate nodes in advance.

Consider the design of n-bit ripple carry adder, input signals Ai and Bi are stable long until the initial carry input signal q, (ie C0) arrives after rippling through the previous

full adder stages.

By this way, the internal node capacitances will be prechared/discharged in advance. On the arrival of last carry signal C_i n the node capacitance at Y (refer Figure 4.3) will be discharged and thereby the body effect associated with carry signal is reduced.

But placing the transistor of $C_{i,n}$ closer VDD and ground would charge/discharge the node capacitance and internal capacitance in faster rate.

The speed of this full adder can be improved by exploiting the inverting property. In design of n-bit ripple carry adder every odd inputs i.e, (A1, A3, A5 ...) (B1,B3, B5 ...) (C1, C3, C5 ...) are invertered.

This procedure will eliminate an inverter in a carry chain, as explained in Figure 4.4. This type of configuration would increases the transistor count due to the negation required at every odd stages of inputs A, B and C.

Mirror Adder

The mirror adder is a circuit having improved performances than conventional static full ladder.

- The total number of transistors required to implement this circuit is 24.
- The functional (realization of this adder is based on the Eq. (4.4, 4.5). In this circuit realization the PMOS network is identical to the NMOS network rather than being the conduction complement, so the topology is called a mirror adder.
- This simplification reduces the number of series transistors and makes the layout more uniform. It is possible because the addition function is symmetric.
- The transistor diagram of mirror adder is shown in Figure 4.5.

Characteristics features of mirror adder are:

- In this adder the carry inversion output signal is not required as in the case of conventional static adder which reduces 2 transistors at the output of CARRY.
- The PMOS network and NMOS network does not obey the principles of duality. The structure of PMOS and NMOS stacks for carry and sum generation resembles same or symmetrical.
- When (A = '1', B = '0') or (A = '0,B= '1') the is connected to VDD, which means carry is propagated to SUM subcircuit. when both A and B are
- high the is connected to ground, which means carry signal is killed.
- When (A = '1', B = '0') or (A = ,Q',B='V) the is connected to ground, which means SUM is complement of carry signal, when both A and B are high the is connected to VDD. This behaviour shows that the SUM and CARRY function obey self-duality principle.
- The carry signal C_i, is kept close to the output node so as to reduce the body effect.

- Feed the carry-in signal (C_i) 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, with carry logic minimum size (1 unit, e.g., 4X). This minimizes the branching effort on the critical path. Keep routing this signal as short as possible to reduce interconnect capacitance.

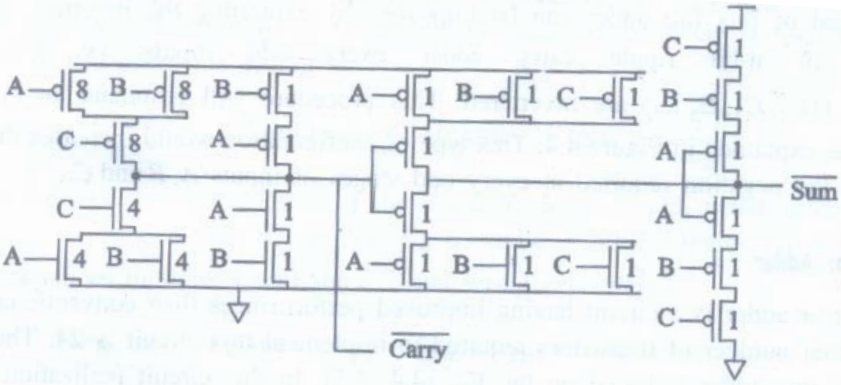


Figure 4.5 Mirror Adder

Theperformance of this full adder is optimized only by choosing appropriate aspect ratio (or logical effort) of PMOS and NMOS transistors.

- Use relatively large transistors on the critical path so that stray wiring capacitance is a small fraction of the overall capacitance.

Advantages of Mirror Adders are

- The circuit possess full swing output without any threshold drop.
- Using a mirror configuration makes it easier to design and optimize layouts, thus shortening the time to market.
- Using a mirror configuration reduces power consumption (to almost zero under static conditions).
- The transistor dimensions are easier to determine for a given manufacturing process.

Disadvantages of Mirror Adders are

- Additional inverters are required for SUM and CARRY output which increases the transistor count.
- The aspect ratio of PMOS and NMOS transistors should be taken in correct value to get the desire performance of full adder.
- The mirror adder has a greater delay to compute SUM than CARRY.

3. Explain transmission-gate based full adder. [CO4-L2]

- A transmission gate (TG) full adder is designed using multiplexers and XORs. A full-adder implemented based on this approach is shown in Figure 4.6. It is based on

the propagate-generate model in Eq (4.3, 4.4, 4.5).

The characteristics of TG full adder are:

- The design can be understood by parsing the transmission gate structures into multiplexers and an "invertible inverter" XOR structure, i.e., the input circuit provides both XOR and XNOR output (P, P).
- The outputs P and P are fed into the output array of TGs to produce SUM and CARRY.
- This circuit has the characteristics that the delays for SUM and CARRY are about the same due to use of lower and upper multiplexers (MUX1 and MUX2)
- If the input bits are applied simultaneously, then both the SUM and CARRY bits will be valid at about the same time.
- When input A = ‘1’, B = '0' or A = '0', B = '1' the output of XOR is high which enable MUX1. In MUX1 the SUM output depends on the complementary input of *Ci.* For example if $C_{i=0}$ then *P* selects the complementary value of C_i, i.e., '1'. The output of MUX2 will be the complement of .

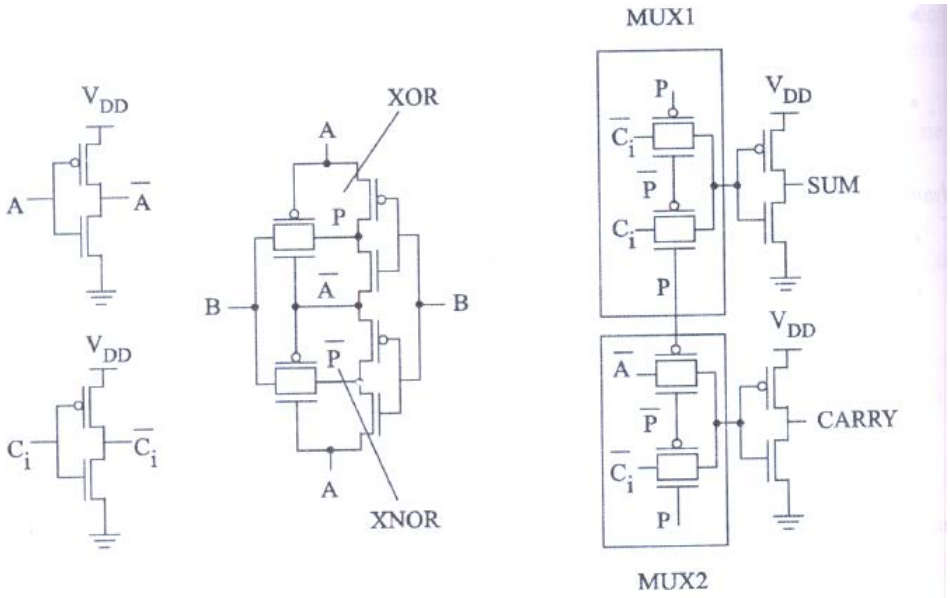


Figure 4.6 Transmission gate full adder

Advantages of TG Adder

- All the paths have small resistance, each of them consisting of no more than two serial channels.
- This is distinctly different from static and mirror full adder in which the CARRY output bit is produced first and then used in calculating the SUM output.
- Both SUM and CARRY outputs are validated at the same time so the delay of

SUM and CARRY will be almost equal.

- The total number of transistors required to construct this full adder is 24.

Disadvantages of TG Adder

- The driving capability is very poor in transmission gate adders. Definitely TG full adders have cascading problems.
- The power dissipation is high when compared to static and mirror full adder.
- Noise margin is moderate and output gets degraded when operates at low voltages.
- It is difficult to fabricate due to complex and irregular structure.

4. Explain Manchester carry chain adder. [CO4-L2]

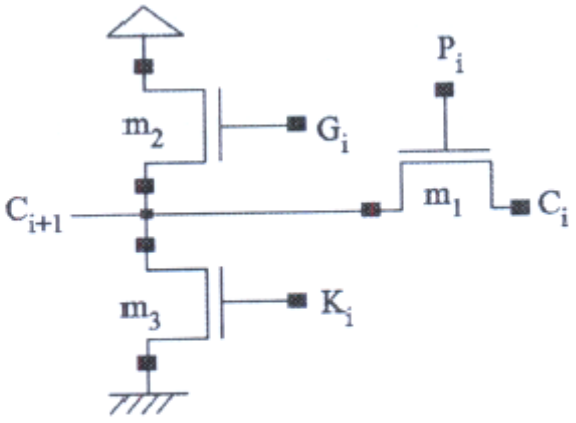
- The Manchester Carry-Chain Adder is a chain of pass-transistors that are used to implement the carry chain. From the full adder truth table (refer Table 4.1) three intermediate equations are derived:
- G_i (generate) = A_i ⬬ B_i row 6 and 7
- P_i (propagate) = row 2,3, 4 and 5

K_i (kill) = row 0 and 1

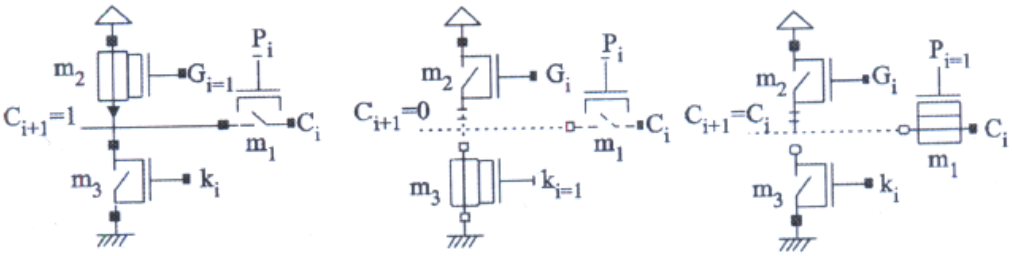
Finally the next state carry output is given as

C_i (CARRY) = $G_i + P_iC_{i-1}$ for i = 0,1,2…

- The Manchester carry chain adder is designed based on the three intermediate signals G_i, P_i and K_i in which only one signal is valid at a time.
- For instance if K_i =1 then G_i, and P_i will be zero, which will make the next state carry in Ci+ 1will be 0. This behavior can be realized using NMOS pass transistor which is depicted in Figure 4.7. The operation can be understood by examining each possibility.



(a) Switching Network



(b) Switching response for G_i = '1' (c) Switching response for K_i = '1' (d) Switching response for P_i = '1'

Figure 4.7 Switching Network for Manchester carry chain

- Consider the case when (Ai, Bi) = (1, 1), then G_i, = 1 which makes the transistor m2 'ON' and the output is tied to VDD i.e, C_{i+l} = l and the remaining transistors ml and m3 is in 'OFF' state.
- This switching characteristic is demonstrated in Figure 4.7(b). For the case (Ai, Bi) = (0, 0), then Ki = 1 which makes the transistor m3 'ON' and the output is tied to ground i.e, Ci+ 1 =0, and the remaining transistors m1 and m2 is in 'OFF' state.
- This switching characteristic is demonstrated in Figure 4.7(c). Finally for , then Pi = 1 which makes the transistor 'ON' and the input Ci, is propagated as output i.e, Ci +1 = Ci, and the remaining transistors m2 and m3 is in 'OFF' state. This switching characteristic is demonstrated in Figure 4.7 (d).
- At the circuit level it is important to note that there will be threshold drop when passing logic 1 due to the characteristics of NMOS passing transistor logic.

Advantages of Manchester carry-chain adder

- This adder provides faster carry propagation/ generation output. A Manchester carry chain generates the intermediate carries by tapping off nodes in the gate that calculates the most significant carry value.
- When the Manchester carry adder is implemented using pass transistor reduces the total of transistor count, parasitic capacitances and wiring capacitances.
- When the Manchester carry adder is implemented using dynamic logic offers better speed performance when compared to static, mirror or TG.

Disadvantages of Manchester carry-chain adder

- When the Manchester adder constructed using static pass transistor logic then threshold variation problems occurs. So level restoration circuits or buffers are needed to be inserted to boost the signal levels.
- When the Manchester adder constructed using dynamic logic then charge sharing and capacitive loading problems are predominant.
- One of the major shortfalls of the Manchester carry chain is that the capacitive load of all of these outputs, together with the resistance of the transistors causes the propagation delay to increase much more quickly than normal adders.

The propagation delay is quadratic with the number of bits N. The optimum number of stages per buffer depends on the equivalent resistance of the inverter and the resistance and capacitance of the pass transistors

5. Explain speed and area tradeoff for circuit design consideration of full adder. [CO4-H1]

The challenge of Speed/Area tradeoff is addressed by several multifaceted optimization approaches such as the technology used for the construction of design, the logic topologies, the circuits, architectures and algorithm.

- At circuit level Speed/Area can also be achieved by selecting proper logic family and topology (logical construction) being used in the implementation of a design and selecting optimized layout.
- On looking into the design aspect of static CMOS design the number of transistors required to construct this adder will be 28. The layout construction is possible only in twin-tub process to bypass the mobility degradation. The delay incurred in this adder will be more, since SUM logic is a dependent quantity of CARRY logic. Due to large PMOS networks the intrinsic load capacitance delay is very high in this adder design.
- In mirror transistor representation the PMOS network is identical to NMOS network with 24 transistors to produce complemented SUM and CARRY logic. The SUM delay depends on CARRY output and obeys self-duality principle.
- A TG realization utilizes multiplexers and XOR/XNOR to produce SUM and CARRY logic. Both SUM and CARRY outputs are validated at the same time so the delay will be almost equal. The total number of transistors required to construct this full adder is 24.
- The Manchester carry chain adder is designed based on the three intermediate signals G_i, P_i and K_i in which only one signal is valid at a time. They can be constructed using pass transistor or dynamic logic. This adder provides faster carry propagation/ generation output. A Manchester carry chain generates the intermediate carries by tapping off nodes in the gate that calculates the most significant carry value. The area/speed optimization and complexity for circuit design consideration is shown in Table 4.2

Table 4.2 The area/speed optimization for circuit design consideration of full adder

Logic	#Tr	Output Voltage	Dependence	Process	Complexity	Sum/Carry Expression	Area optimization	Speed optimization
Static CMOS	28	Full swing	Sum delay depends on Carry out	Twin-tub	high	CARRY $= AB + AC_i + BC_i$ SUM $= ABC_i + \overline{\text{CARRY}}(A + B + C_i)$	Contracted diffusion	lowering the logical effort of PMOS stack input ordering scheme
Mirror	24	Full swing	Sum delay depends on Carry out	n-well	simple	CARRY $= G_i + P_i C_{i-1}$ SUM $= P_i \oplus C_i$	Transistor sizing	lowering the logical effort of PMOS stack input ordering scheme
TG	24	Partial	Sum Independent of Carry	Twin/ n-well	Moderate	CARRY $= P_i \bullet C_i + \overline{P_i} \bullet A_i$ SUM $= P_i \bullet \overline{C_i} + \overline{P_i} \bullet C_i$	Transistor sizing	Inclusion of buffers
Manchester	28 (for PTL)	No full swing	Sum delay depends on Carry out	Twin	High	P_i propagate $= A_i \oplus B_i$ G_i generate $= A_i \bullet B_i$ K_i (kill) $= \overline{A_i + B_i} = \overline{A} \bullet \overline{B}$	Contracted diffusion Transistor sizing	Keeper circuits Buffers

6. Explain the architectures of ripple carry adder. [CO4-L2]

The Binary Adder: Logic Design Consideration

- Generally the adder design are classified as low, moderate and high speed adders based on how the carry signals are generated and propagated in the circuit.
- Ripple carry adder falls in the first category where the speed is limited by the rippling effects in the carry chain, but area is minimal.
- The second category of adders will exhibit moderate delay and area. Some of the adders in this class are Carry Save Adder, Carry Look-Ahead Adder, Carry Increment adder, Carry Skip Adder, Carry Bypass Adder and Carry Select Adder.
- The last category of adder is called high speed adder or logarithmic adder or parallel prefix adders like Kogge-stonc, Brent-Kung, Sklansky, Han Carlson, Knowles, and Ladner Fischer.

1. Ripple Carry Adder (RCA)

- Ripple Carry Adder (RCA) is the simplest, but slowest adders with O (n) area and O (n) delay, where n is the operand size in bits. The ripple carry adder is constructed by cascading full adders (FA) blocks in series.
- One full adder is responsible for the addition of two binary digits at any stage of

the ripple carry. The carryout of one stage is fed directly to the carry-in of the next stage. Even though this is a simple adder and can be used to add unrestricted bit length numbers, it is however not very efficient when large bit numbers are used.

- One of the most serious drawbacks of this adder is that the delay increases linearly with the bit length. The worst-case delay of the RCA is when a carry signal transition ripples through all stages of adder chain from the least significant bit to the most significant bit, which is approximated by:

$$t = (n-1)\, t_c + t_s \quad \text{---------- (4.11)}$$

Where,

- tc is the delay through the carry stage of a full adder,
- ts is the delay to compute the sum of the last stage.
- The delay of ripple carry adder is linearly proportional to the number of bits; therefore the performance of the RCA is limited when n grows bigger. The advantages of the RCA are lower power consumption as well as compact layout giving smaller chip area.
- The schematic of RCA is shown in Figure 4.11

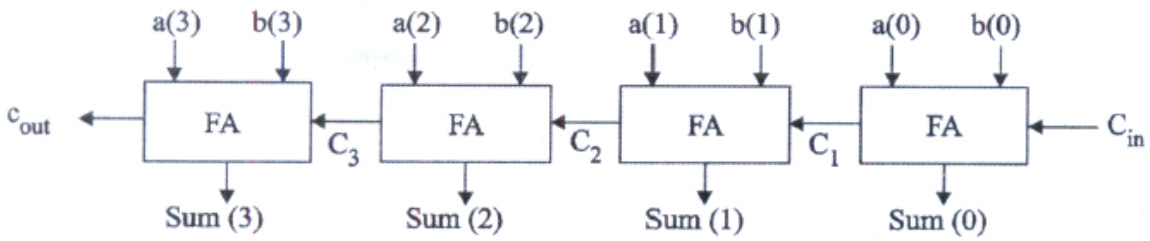


Figure 4.11 4-bit Ripple carry adder

Advantages of RCA

- Circuit realization is very simple
- Consumes less power
- Compact layout giving smaller chip area

Disadvantages of RCA

- The carry bit may have to propagate from LSB to MSB. Worst case delay for a N-bit adder is 2N-gate delay.
- The delay of ripple carry adder is linearly proportional to n, the number of bits; therefore the performance of the RCA is limited when n grows bigger.

7. Explain Carry-Bypass Adder (CByA). [CO4-L2]

The Carry-Bypass Adder (CByA)

- It is an alternate structure of RCA. The linear dependence of the adder speed on the number of bits makes the usage of ripple carry adders rather impractical.
- Logic optimizations are therefore necessary, resulting in adders with tp<0 (n), where tp is the propagation delay and n number of bits.

- As in a ripple-carry adder, every full adder cell has to wait for the incoming carry before an outgoing carry can be generated. This dependency can be eliminated by introducing an additional bypass (skip) to speed up the operation of the adder.
- Consider in the design of 4-bit RCA, an incoming carry C_i, 0 = 1 propagates through complete adder chain and causes an outgoing carry C0, 3 = 1 under the conditions that all propagation signals are 1.
- This information can be used to speed up the operation of the adder, as shown in Figure 4.12. When BP = P0 P1P2P3 = 1 , the incoming carry is forwarded immediately to the next block through the bypass and if it is not the case, the carry is obtained via the normal route.
- If (P0P1P2P3 = 1) then C0,3 = Ci,0 either Delete or Generate occurred. Hence, in a CByA the full adders are divided into groups, each of them is "bypassed" by a multiplexer. If this is not the case, the carry is obtained by way of the normal route.

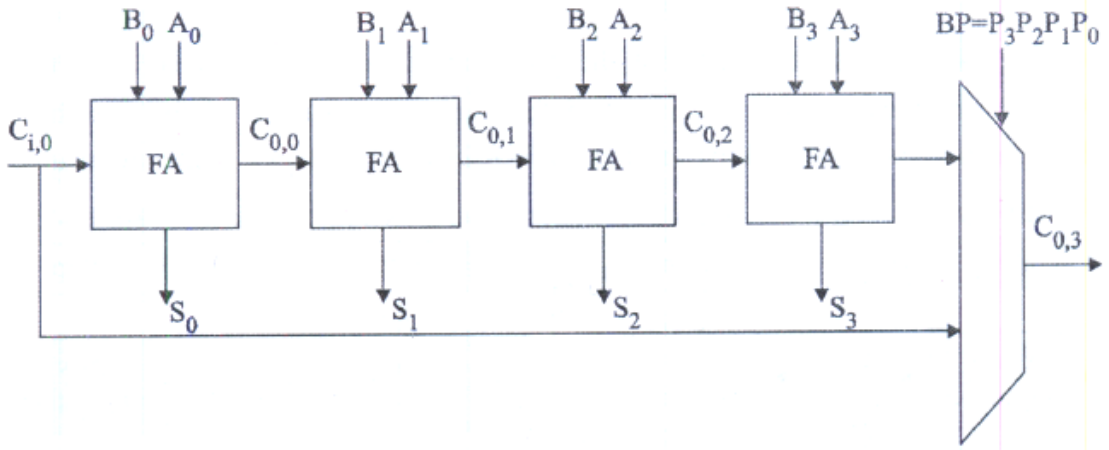


Figure 4.12 4-bit Carry Bypass adder

- The bypass adder designed using Manchester carry chain is shown in Figure 4.13. The carry propagates either through the bypass path, or a carry is generated somewhere in the chain.

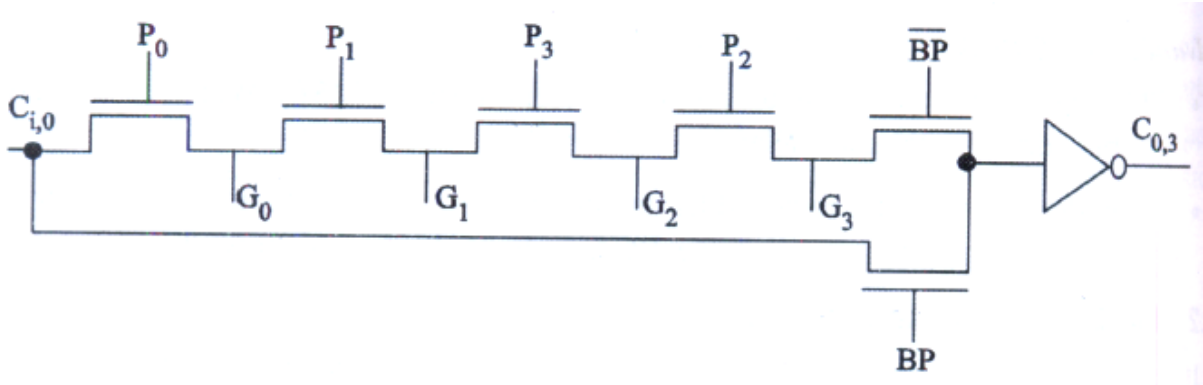


Figure 4.13 Bypass adder designed using Manchester carry chain

- This explains the delay incurred in bypass adder for N-bit addition. Let us consider the length of the adder is N-bit. Divide the N-bit adder into N/M equal-length bypass stages, where M defines the bit length of that divided stage.

- Initially the time taken to create the propagate and generate signal is given as t_{setup}. Let t_{carry} defines the propagation delay through a single bit, so for M stage the worst case propagation delay will be M times t_{carry}. Now t_{bypass} is the delay ocurred through bypass multiplexer for a single stage.
- Finally t_{sum} defines the total time to generate the sum of the final stage. The critical path delay of bypass adder for 0 to N - 1 bit is shown in grey color (Figure 4.14). Therefore the total propagation delay of N-bit bypass adder is given as

$$t_p = t_{setup} + Mt_{carry} + \left(\frac{N}{M} - 1\right) t_{bypass} + (M-1)\, t_{carry} + t_{sum} \text{ ---- (4.12)}$$

From Eq. (4.12) it is observed that the propagation delay tp is linear with the number of bits N. The worst case delay occurs when the carry is generated at the first bit position, ripples through the first block, skips around (N/M - 2) bypass stages, and is consumed at the last bit position without generating an output carry.

- The optimal number of bits per bypass adder depends on:
- The extra delay of the bypass selecting multiplexer.
- The buffering requirements in the carry chain.
- The ratio of the delay through the ripple and the bypass paths.

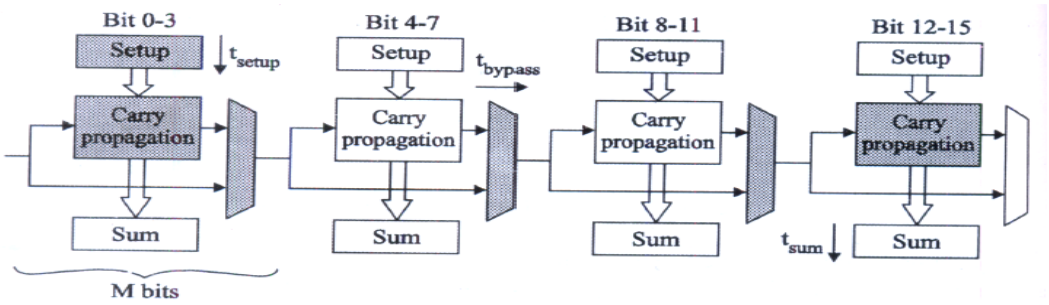


Figure 4.14 Propagation delay in Bypass Carry adder

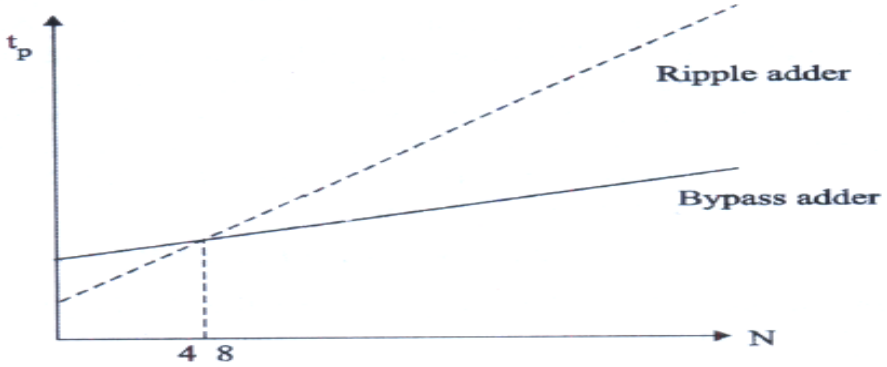


Figure 4.15 Propagation delay of RCA and CByA

- The propagation delay of RCA and CByA is shown in Figure 4.15. From the graph it is noticed that for RCA the delay is quadratically exponent for N-bit length. But for smaller bit lengths the delay is minimal when compared to Bypass adder.
- For N-bit bypass adder the delay is linear and less when compared to ripple carry adder. The delay of CByA is optimal when the adders are segmented for the widths of 4 to 8-bits per stage.

Advantages of CByA

- The propagation delay is smaller when compared to ripple carry adder when optimal stages are used.
- Carry-Bypass adder allows carry to skip over groups of m bits.

Disadvantages of CByA

- The area overhead incurred by adding the bypass path increases silicon area of about 10 to 20%.
- For N-bit bypass adder the delay is linear and less when compared to ripple carry adder, however the delay of CByA is optimal when the adders are segmented for the widths of 4 to 8-bits per stage.
- Suitable for large size adders.

8. Explain the the carry-skip adder (CSkA) and estimate the delay calculation. [CO4-H1]

- It is an alternate structure of RCA. A carry-skip adder consists of a simple ripple carry-adder with a special speed up carry chain called a skip chain.
- Carry skip adder is a fast adder compared to ripple carry adder when addition of large number of bits take place; carry skip adder has O (√n) delay provides a good compromise in terms of delay, along with a simple and regular layout This chain defines the distribution of ripple carry blocks, which compose of the skip adder.
- A carry-skip adder is designed to speed up a wide adder by aiding the propagation of a carry bit around a portion of the entire adder. Actually the ripple carry adder is faster for small values of N.
- However the industrial demands these days, which most desktop computers use word lengths of 32 bits like multimedia processors, makes the carry skip structure more interesting.
- The crossover point between the ripple-carry adder and the carry skip adder is dependent on technology considerations and is normally situated for 4 to 8 bits. The carry-skip circuitry consists of two logic gates. The AND gate accepts the carry-in bit and compares it to the group propagate signal

using the individual propagate values. The output from the AND gate is ORed with cout of RCA to produce a stage output of

$P_{[i, i+3]} = P_{i+3}.P_{i+2}.P_{i+1}.P_i$ ---------- (4.13)

$CARRY = C_{[i+3]} + P_{[i, i+3]} \ldots. C_i$ ---------- (4.14)

- If $P_{[i.i+3]} = 0$, then the carry-out of the group is determined by the value of C_{i+3}. However, if $P_{[i.i+3]}$ =1when the carry-in bit is ci, = 1, then the group carry-in is automatically sent to the next group of adders. The design of Carry Skip Adder is shown in Figure 4.16.

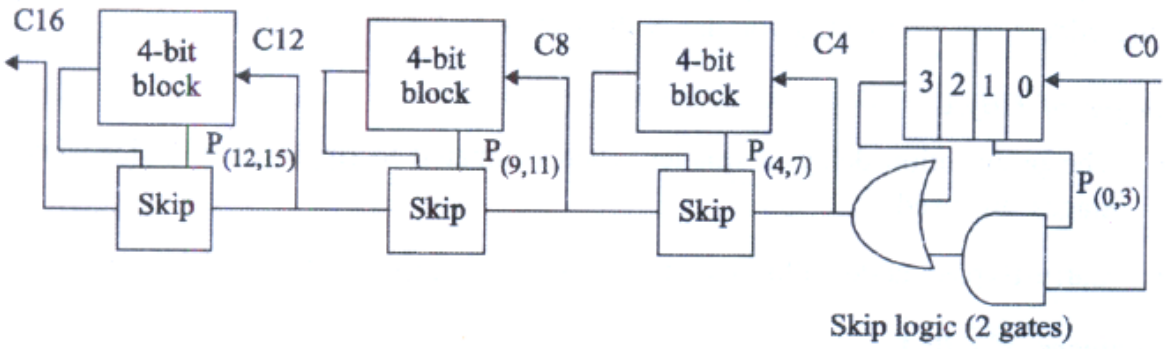


Figure 4.16 Carry skip-adder

- Carry Propagate: $P_i = A_i \oplus B_i$
- Sum: $S_i = P_i \oplus C_i$
- Carry Out: $C_{i+1} = A_i B_i + P_i C_i$

- The sizes of the blocks are chosen so as to minimize the longest life of a carry chain. Based on the values of the propagate bits of one group of bit (Pi, Pi+1, ... Pi), the final carry can be predicted.

Delay Calculation

- The overall performance of a carry-skip adder is determined by the number of bits in each group. Assuming that one stage of full adder (two gate levels) has the same delay as one skip, the worst-case propagation delay in an N-bit carry-skip adder with a fixed block size M can be approximated as follows (Figure 4.17).

$$t_{p(skip)} = 2M + N/M - 3.5 \text{ ---------- } (4.15)$$

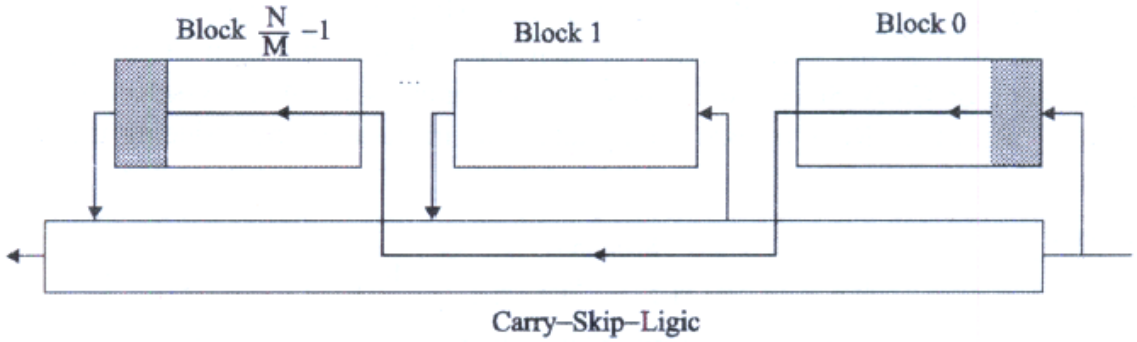


Figure 4.17 Propagation delay in Carry-skip adder

- Hence, to minimize the .propagation delay, the optimal group size k of an n-bit carry skip adder is approximate to M= √n/2. Substituting this result back to $t_{p(skip)}$the propagation delay of the adder with optimal group size is approximated to

$T_{opt(skip)} = 2\sqrt{2N} - 3.5$ ---------- (4.16)

- From the above results, the optimal group size of a 16-bit carry-skip adder is 4 and the propagation delay is 7.96 stages.

Advantages of Carry skip adder

- The propagation delay is smaller when compared to ripple carry adder when optimal stages are used.
- A carry-skip adder is shown for be superior for constant-width carry-skip mechanism the advantage being greater at higher precisions.

Disadvantages of Carry skip adder

- Complexity of the circuit increases due to skip logic.
- Area and power consumption is high.

9. Explain Carry-Select Adder (CSelA). [CO4-L2]

It is an alternte structure of RCA. A carry-select adder is divided into two sectors, each of which performs two additions in parallel, one assuming a carry-in of zero, the other a carry-in of one.

- A four bit carry select adder generally consists of two ripple carry adders and a multiplexer. The carry-select adder is simple but rather fast, having a gate level depth of O (√n). Adding two n-bit numbers with a carry select adder is done with two adders (two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one.
- After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the help of multiplexer. The design schematic of Carry Select Adder is shown in Figure 4.18.
- A carry-select adder speeds 40% to 90% faster than RCA by performing additions in parallel and reducing the maximum carry path.

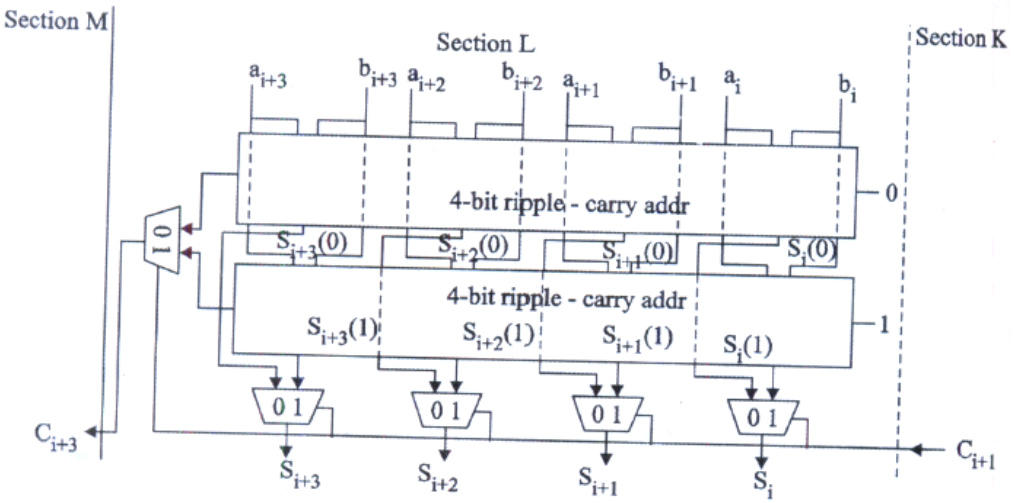


Figure 4.18 Carry select adder

Delay Calculation

- The propagation delay of an N-bit carry-select adder is proportional to the equal length adder stages. Divide the N-bit adder into N/M equal-length stages, where M defines the bit length of that divided stage. Initially the time taken to create the propagate and generate signal is given as t_{setup}.
- Let t_{carry} defines the propagation delay through a single bit, so for M stage the worst case propagation delay will be M times t_{carry}. Now t_{mux} is the delay incurred through multiplexer for a single stage.
- Finally t_{sum} defines the total time to generate the sum of the final stage. The critical path delay of bypass carry adder for 0 to N-l bit is shown in Figure 4.19. Therefore the total propagation delay of N-bit carry select adder is given as

- $t_p = t_{setup} + M_{tcarry} + \left(\frac{N}{M}\right) t_{mux} + t_{sum}$ ---------- (4.17)

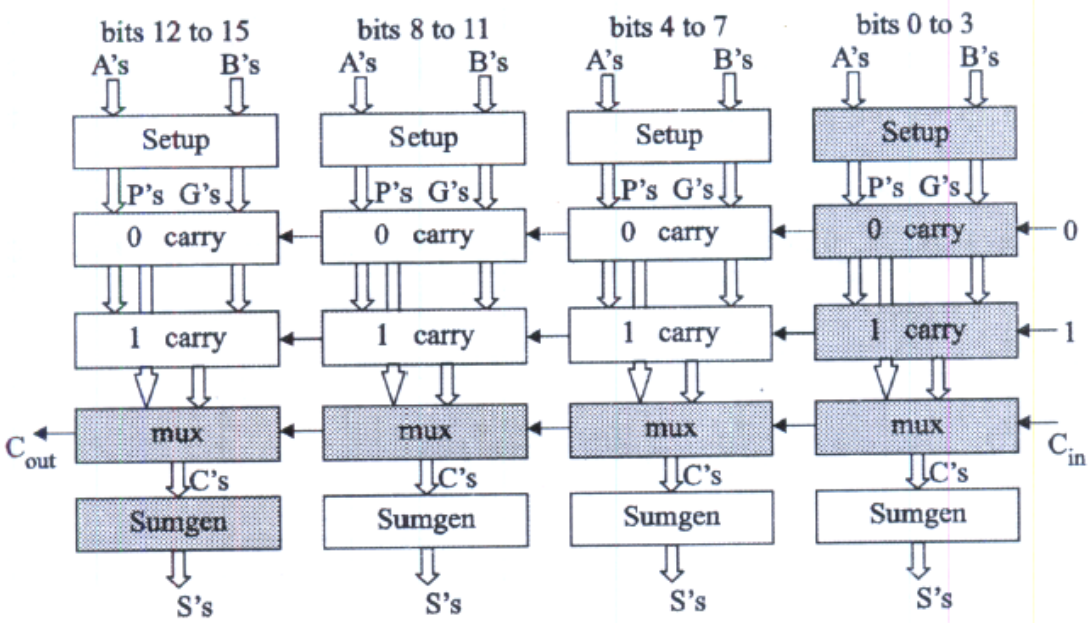


Figure 4.19 Propagation delay of carry select adder

Advantages of carry select adder

- The propagation delay is smaller when compared to ripple carry adder when optimal stages are used.
- The carry select adders (CSelA) reduce the computation time by pre-computing the sum for all possible carry bit values (ie '0' and '1'). After the carry becomes available the correct sum is selected using multiplexer.

Disadvantages of carry select adder

- Power consumption is high
- Carry Select Adder are in the class of fast adders, but they suffer from fan-out

limitation since the number of multiplexers that need to be driven by the carry signal increases exponentially. In the worst case, a carry signal is used to select nil multiplexers in an n-bit adder. When three or more operands are to be added simultaneously using two operand adders, the time consume for carry propagation must be repeated several times. If the number of operands is then carries have to propagate (k - 1).

10. Explain the concept of Conditional-Sum Adder. [CO4-L2]

- Conditional-sum adder is a special application of the carry-select adder in a way such that the carry-select addition starts from single bit and recursively doubling to n/2 bits. An 8-bit conditional-sum adder is exhibited in Figure 4.20.
- The first two rows consist of full adders and compute the sum and carry-out assuming that the carry-in bits are 0 and 1, respectively.
- The rest of the adder is the sum-selection tree that selects the proper sum bit. The propagation delay of an n-bit conditional-sum adder is $\log_2 n$: An n-bit conditional-sum adder requires 2n - 1 full adders and an approximate number of n ($\log_2 n$- 1) + 1 multiplexers.

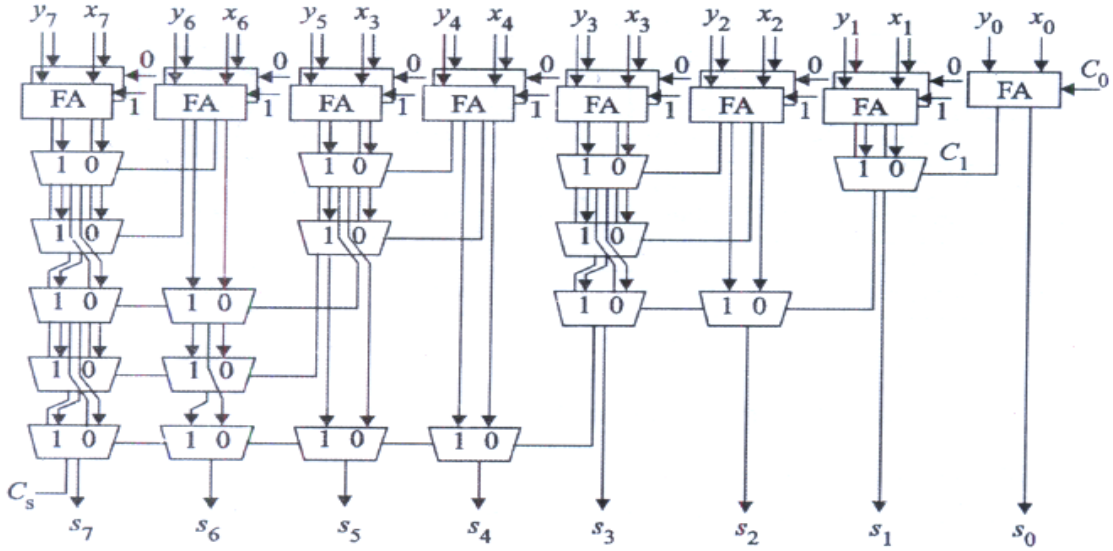


Figure 4.20 The logic circuit of an 8-bit conditional-sum adder

11. Explain with neat sketch the Carry Save Adder. [CO4-L2]

- There are many cases where it is desired to add more than two numbers together. The straightforward way of adding together m numbers (all n bits wide) is to add the first two, then add that sum to the next, and so on. This requires a total of m - 1 additions, for a total gate delay of O (m log n) (assuming look-ahead carry adders). Instead, a tree of adders can be formed, taking only O (log m - log n) gate delays.
- Using carry save addition, the delay can be reduced further still. The idea is to take 3 numbers that we want to add together, x + y + z, and convert it into 2 numbers c + s such that x + y + z = c + s, and do this in 0(1) time.
- The reason why addition cannot be performed in 0(1) time is because the carry

information must be propagated. In carry save addition, the intermediate carry is retained from directly passing on the carry information until the very last step.

- The basic block for 4-bit carry save adder is shown in Figure 4.21. Carry - save adders are based on the idea that a full-adder really has three inputs and produces two outputs. The name "carry - save" arises from the fact that we save the carry-out word instead of using it immediately to calculate a final sum.

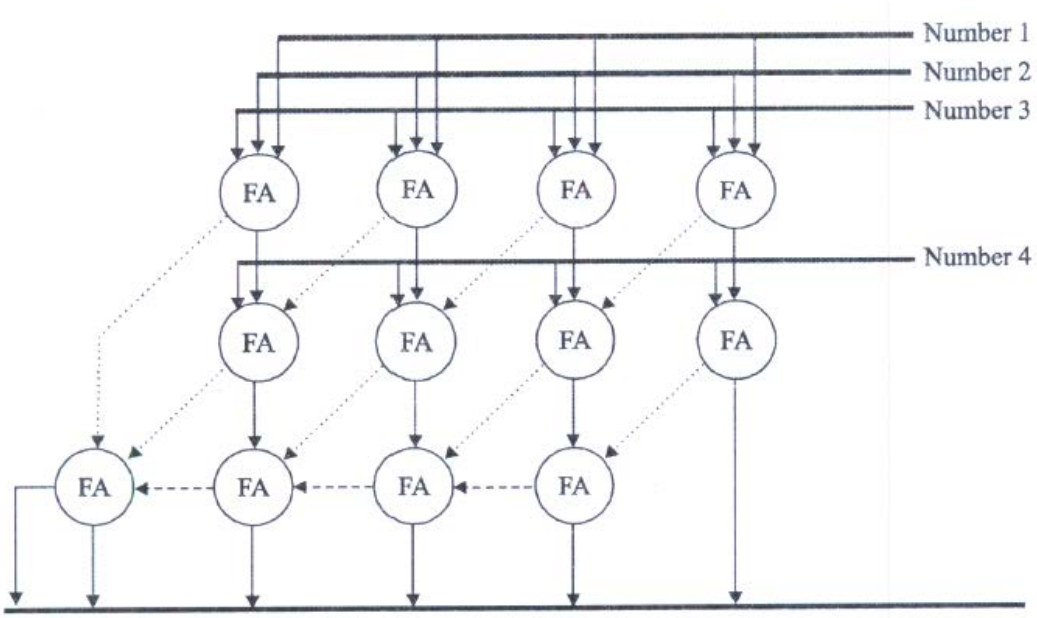


Figure 4.21 4-bit Carry save adder

Carry - save adders (CSAs) are useful in situations where we need to add more than two numbers. Since the design automatically avoids the delay in the carry - out bits, a CSA chain may be faster than using a standard adders or cycling with a clocked synchronous network.

Discuss with necessary diagram Carry Look-ahead Adder (CLA)

- The Look-ahead carry, or fast carry, technique reduces propagation times through the adder. This is accomplished by generating all the individual carry terms needed by each full-adder in as few levels of logic as possible. Let two input numbers be x = (x_{n-1},.....x_1,x_0.) and y = (yn-1,…y_1,y_0); and sum be S = (Sn-1...S1 S0).
- At the i^{th} state of an n-bit ripple-carry adder as depicted in Figure 4.22, the output carry C_{i+1} is generated if both inputs x_i and y_i, are 1, regardless of what value of the input carry is c_i. The input carry c_i is propagated to the output if either input of x_i and y_i is 1. Consequently, we can define two new functions: carry generate (g_i,) and carry propagate (p_i) in terms of inputs x_i and y_i as follows.

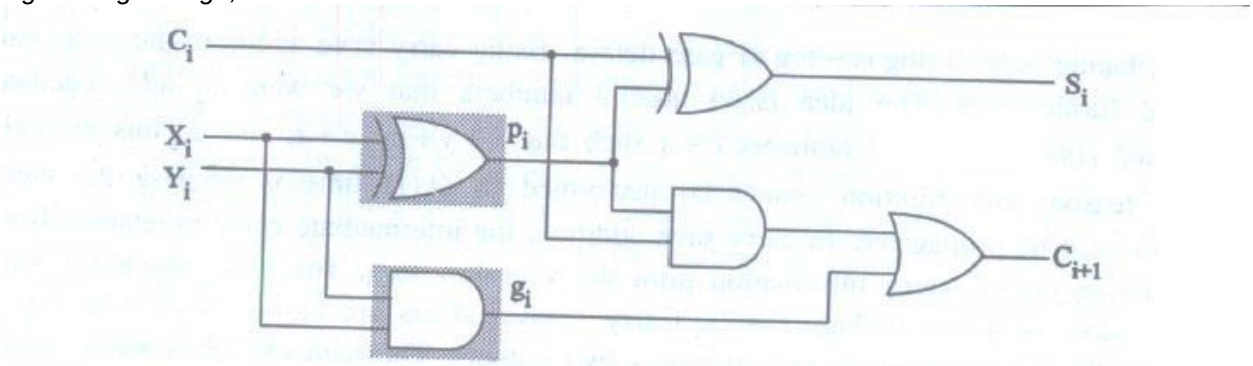


Figure 4.22 The logic circuit used to define carry generator and propagate functions

-
- $g_i = x_i \bullet y_i$..... 4.18
- $p_i = x_i \oplus y_i$ 4.19
- Based on these two functions, the sum and carry-out can then be separately represented as a function of both carry generate (g_i) and carry propagate (p_i) and are as follows
- $s_i = p_i \oplus c_i$ 4.20
- $c_{i+1} = g_i + p_i c_i$ 4.21
- As a result, the carry-in of the (i+l) th stage full adder can be generated by using the recursive equation of the c_{i+1}: For example, the first four carry signals are

$$c_1 = g_0 + p_0 c_0$$

$$c_2 = g_1 + p_1 c_0$$

$$= g_1 + p_1 (g_0 + p_0 c_0) = g_1 + p_1 g_0 + p_1 p_0 c_0$$

$$c_3 = g_2 + p_2 c_2$$

$$= g_2 + p_2 (g_1 + p_1 g_0 + p_1 p_0 c_0) = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_0$$

$$c_4 = g_3 + p_3 c_3$$

$$= g_3 + p_3 (g_2 + p_2 g_1 + p_2 p_1 p_0 c_0) = g_3 + p_3 g_2$$

$$+ p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 c_0$$

- The resulting carry signals are only functions of both inputs x and y and the carry-in (c0). These carry signals can then be implemented by a two-level logic circuit known as a carry-look ahead (CLA) generator, as shown in Figure 4.23. By using a carry-look ahead generator, the four-bit adder may be represented as a function of carry-propagate signals p_i and carry signals C_i. That is, $s_i = p_i \oplus c_i$ where 0<i<3: The resulting circuit is shown in Figure 4.24, which is composed of three components: pg generator, CLA generator, and sum generator.

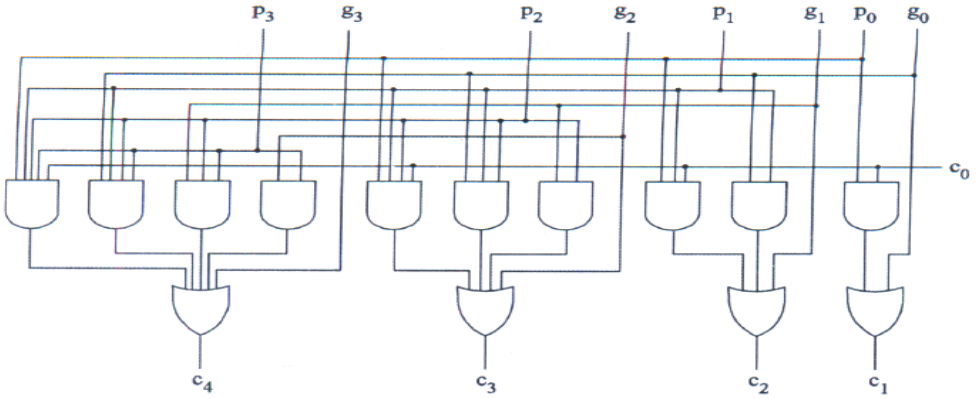


Figure 4.23 The logic circuit of a 4-bit carry-lookahead generator

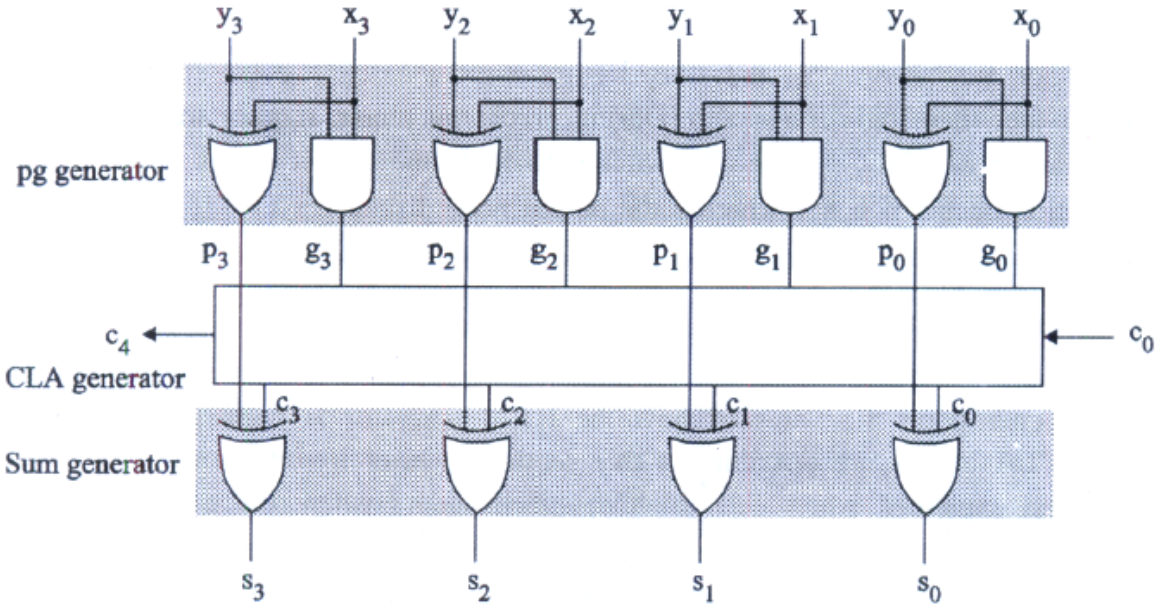


Figure 4.24 A 4-bit carry-lookahead adder

A carry look ahead adder implemented using Manchester carry chain is shown in Figure 4.25. The carry recurrence relation $c_{i+1} = g_i + p_i c_i$ is implemented using NMOS transistors.

Figure 4.25 (a) depicts standard static implementation. When $p_i = 0$ the NMOS transistor M_{n1} is OFF and M_{n3} is ON, thereby turning on the carry-generate inverter. The input, is blocked from propagating through when the carry-generate inverter is enabled. The output carry is equal to the inverted g_i as $p_i = 1$; NMOS transistor M_{nl} is ON and M_{n3} is OFF, thereby turning OFF the carry-generate inverter. The input is able to propagate through, causing the output carry to be equal to .

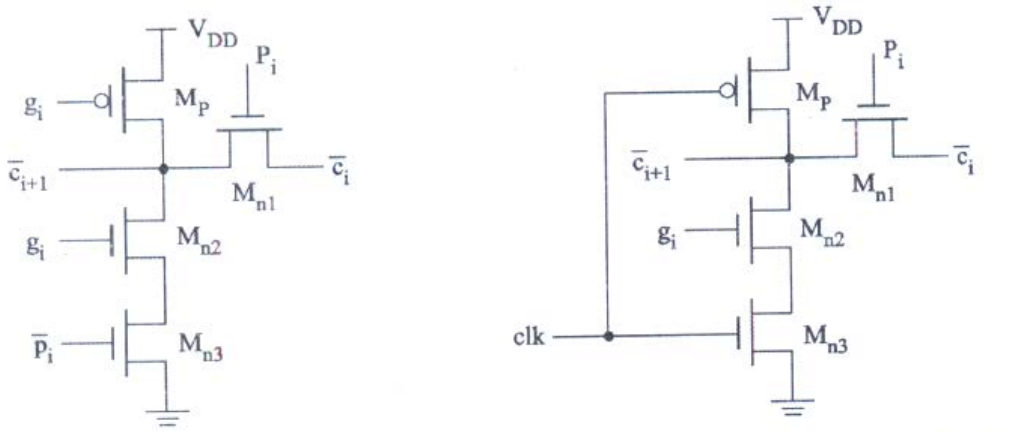


Figure 4.25 The general stages of the Manchester carry style: (a) static circuit; (b) dynamic circuit

The dynamic logic circuit shown in Figure 4.25(b) is similar to the static logic circuit except that now transistors M_p and M_{n3} are used as precharge and evaluate transistors,respectively, rather than logic transistors. During precharge phase, namely, clk =0; the output node is precharged to VDD. As the clk switches to high, the circuit enters evaluate phase. A carry propagation occurs if p_i,= 1 whereas the output node discharges to 0 if g_i,= 1. The mirror implementation of 4-bit carry look ahead adder is shown in Figure 4.26.

12. Explain Speed/Area trade off in Adder design for Ripple carry adder (RCA), CSA,CSKA, CSelA, PPA. [CO4-L2]

In general, the speed/area aspect of any adder depends upon the circuit complexity, technology and circuit design factors. The RCA is the simplest adder producing low power consumption as well as compact layout giving smaller chip area. The delay of ripple carry adder is linearly proportional to n, the number of bits; therefore the performance of the RCA is limited when n grows higher.

The adder scheme CSA structure reduces the propagation effects in RCA with increased full adders and number of stages, thereby interconnect complexity also proportionally increases. The carry skip adder provides a good compromise in terms of area with simple and regular layout with increased delay if the grouping of bits and carry generation units are not proper.

Table 4.3 Circuit complexity of adders in terms of speed/area and interconnect

Adder Type	Delay	Area overhead (N-adder)	Routing Complexity	AT (Area*Delay)
RCA	$2n+4$	n	n^2+1	$3n+4$
CLA	$2+4\log n$	$2n+n/2$	$2n^2+n^2$	$\frac{5n}{2}+2+4\log n$
CSA	$2n$	n^2	n^2-2n	$2n+n^2$
CSkA	$4\sqrt{2}n$	n^2-n	$2n^2-n$	$n^2-n+4\sqrt{2}n$
CSelA	$2+4\sqrt{n}$	$n^2-n/2$	$2n^2+4n$	$n^2-\frac{n}{2}+2+4\sqrt{n}$
PPA	$\log n$	n^3+4n	$4(n^2+1)$	$n^3+4n+\log n$

The adder scheme CLA or CPA reduces delay compared to RCA and CSA with increased interconnect complexity due to separate carry and propagate unit. The adder CselA reduces the computation time by pre-computing the sum for all possible carry bit values, but they suffer from fan-out limitation since the number of multiplexers that need to be driven by the carry signal increases exponentially. In the worst case, a carry signal is used to select n/2 multiplexers in an n-bit adder. The parallel prefix adder offers the maximum speed with high fan-out and interconnects complexity. The circuit complexity of adders in terms of number of bits (n) is shown in Table 4.3.

13. Discuss carious Multipliers design or schemes. [CO4- H1]

Multiplication is an essential arithmetic operation for common DSP applications, such as Filtering, Multiply and Accumulate (MAC) unit and Fast Fourier transform (FFT).

- Multiplier modules occupy 46 percent chip area in most of MAC and FFT module.
- The speed and power consumption of multiplier depends on the algorithm (intermediate Partial Product Generation (PPG) and Partial Product Addition (PPA)) and logic technology used to design the multiplier cell.
- The multiplication algorithms are classified based on the space (area) complexity, interconnect and time complexity. In terms of PPG and PPA the multipliers are categorized into parallel, serial-parallel, complementary, row-column bypass, modulo diminishing - 1 and wave pipelining multipliers.
- Parallel array multiplier is the simplest and most popular method in which the computation is based on add and shift operations. The main benefit of this multiplier is regular in structure with relatively longer latency.

- The time complexity is linear with respect to the operand size and the circuit consumes large power and area as the bit size grows.
- Another improved parallel multiplier is Wallace tree multipliers, reduces the ripple carry effect by incorporating carry save addition to produce the products in far less time.
- The time complexity is less when compaied to array by a factor of n/2, but the interconnect complexity is much higher due to irregular structure. The interconnect complexity can be reduced using Dadda multiplication algorithm work at higher speed than Wallace and array with slight increase in area complexity.

Parallel array multiplier design

Basic Principles of Multiplication:

Let multiplier X and multiplicand Y be n and m-bit numbers, respectively, then the product of X and Y can be expressed as follows.

$$P = X \times Y = \sum_{i=0}^{n-1} x_i \cdot 2^i \cdot \sum_{j=0}^{m-1} y_j \cdot 2^j$$

$$= \sum_{i=0}^{n-1} \sum_{y=0}^{m-1} (x_i \cdot y_j) \cdot 2^{i+j} = \sum_{k=0}^{n+m-1} P_k \cdot 2^k \quad \ldots (4.25)$$

An array multiplier based on RCAs accepts an n-bit multiplier and an m-bit multiplicand and uses an array of cells to calculate the bit product xi , yi in parallel and then adds them together in a proper way to yield the final product. Consider the multiplication of 4 x 4 multiplier and multiplicand. Each row, called a partial product, is formed by a bit-by-bit multiplication of each operand. For example, a partial product is formed when each bit of operand a is multiplied by b0, resulting in a3 b0, a2 b0, a1b0, a0 b0.

$$\begin{array}{cccccccc}
 & & & a_3 & a_2 & a_1 & a_0 \\
 & & \times & b_3 & b_2 & b_1 & b_0 \\
\hline
 & & & a_3 b_0 & a_2 b_0 & a_1 b_0 & a_0 b_0 \\
 & & a_3 b_1 & a_2 b_1 & a_1 b_1 & a_0 b_1 & & \\
 & a_3 b_2 & a_2 b_2 & a_1 b_2 & a_0 b_2 & & & \\
a_3 b_3 & a_2 b_3 & a_1 b_3 & a_0 b_3 & & & & \\
\hline
O_7 & O_6 & O_5 & O_4 & O_3 & O_2 & O_1 & O_0
\end{array}$$

The first two rows of the array multiplier may be combined into one row to reduce the number of full adders by a factor of m: The propagation delay of the critical path of this array multiplier is [2 (m - 1) + n] t_{FA} when the first two rows are not combined and is [2

(m - 1) + (n - 1)] t_{FA} when the first two rows are combined. The array multiplier design using RCA is shown in Figure 4.31. The array multiplier based on CSAs is shown in Figure 4.32. This array multiplier needs mxn AND gates and CSAs, as well as an m-bit RCA adder. The propagation delay of the critical path of this array multiplier is (m + n) t_{FA} when the first two rows are not combined and is [m + (n - 11] t_{FA} when the first two rows are combined. As a consequence, the propagation delay of this array multiplier is reduced by a factor of (m - 2) t_{FA} at the cost of an m-bit RCA

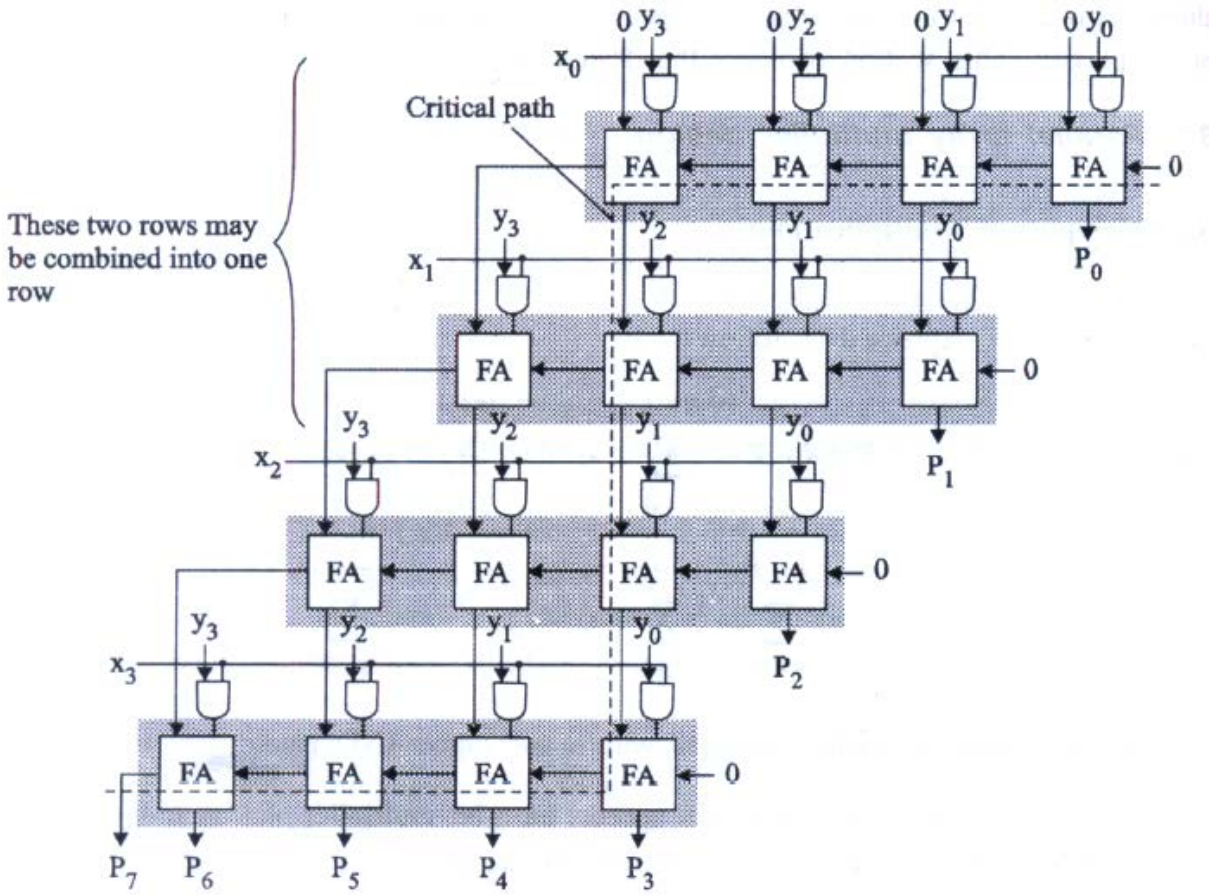


Figure 4.31 An example of an unsigned 4×4 array multiplier using RCAs.

Wallace Multiplier

A Wallace tree is an implementation of an adder tree designed for minimum propagation delay. Rather than completely adding the partial products in pairs like the ripple adder tree does, the Wallace tree sums up all the bits of the same weights in a merged tree. Usually full adders are used, so that 3 equally weighted bits are combined to produce two bits: one (the carry) with weight of n + 1 and the other (the sum) with weight n. Each layer of the tree therefore reduces the number of vectors by a factor of 3:2.

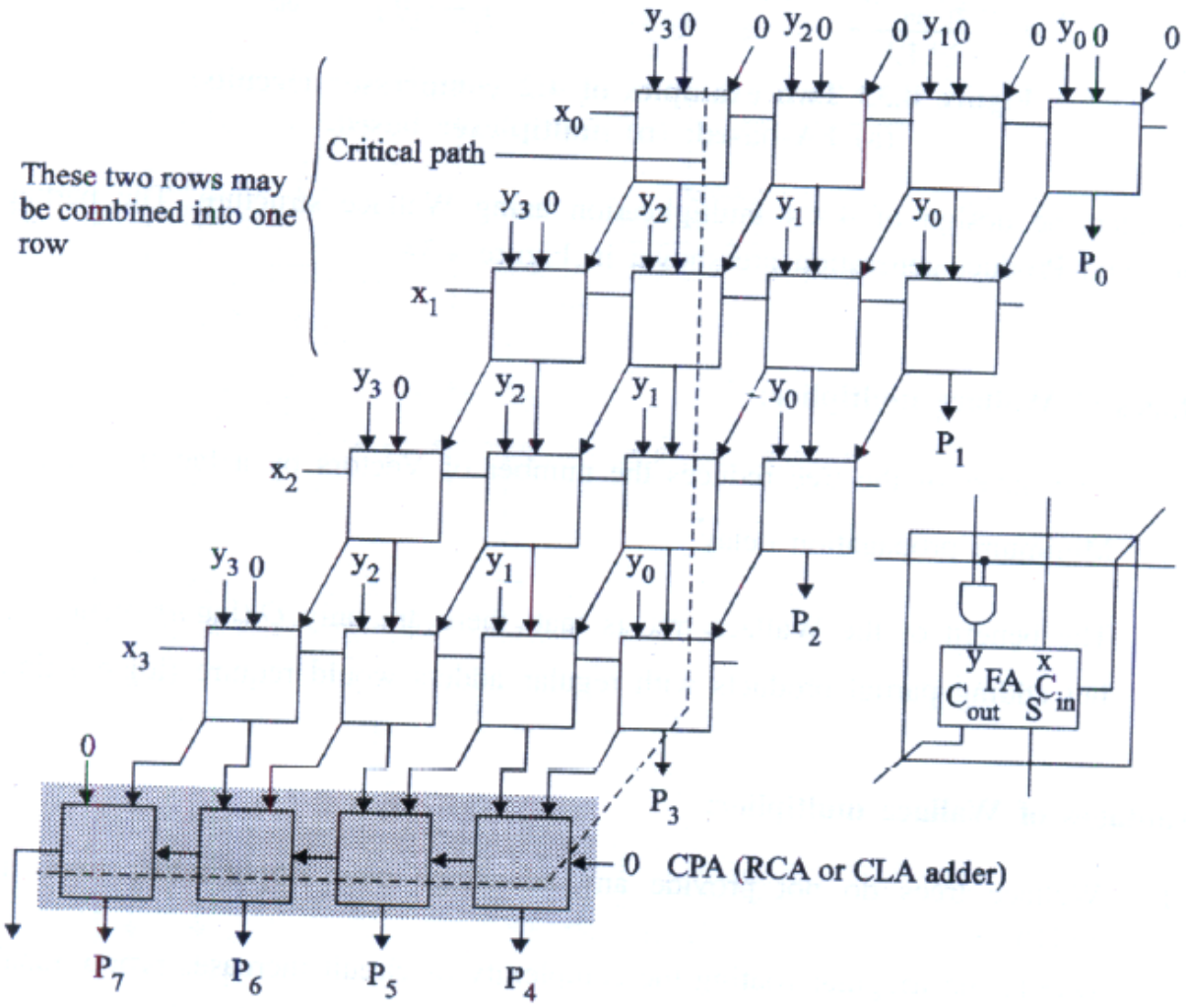


Figure 4.32 An example of an unsigned 4×4 array multiplier using CSAs.

- A k-input Wallace tree accepts k n-bit operands and yields two (n + log2 k - l)-bit outputs.
- The basic cells used in the Wallace tree are 3:2 or 4:2 compressors. Because a compressor effectively accounts for the number of Is in the input and indicates the result as the output, a 3:2 compressor (namely, CSA) is also called a (3:2) counter.
- A 4:2 compressor accepts four equal-weight inputs and yields two outputs. Along the way, it also accepts a carry from the previous column and generates an intermediate carry into the next column. Hence, a 4:2 compressor is more appropriate to be named as a (5:3) counter. Figure 4.33 gives two examples showing how a 4:2 compressor can be built by using either two full adders or six 2-to-l multiplexers and three inverters. It is shown that the performance of the multiplexer-based 4:2 compressor is better than that of the FA-based 4:2 compressor.

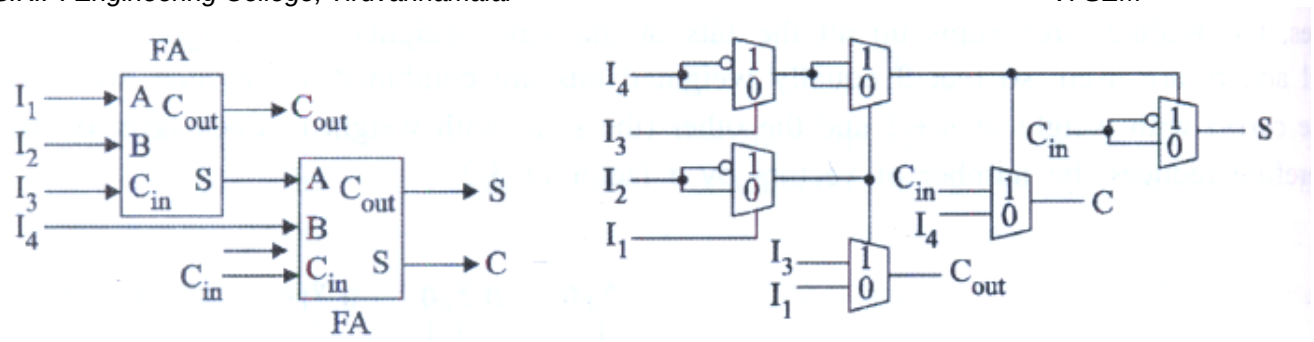


Figure 4.33 Two examples of 4:2 compressor circuits: (a) FA-based; (b) multiplexer based.

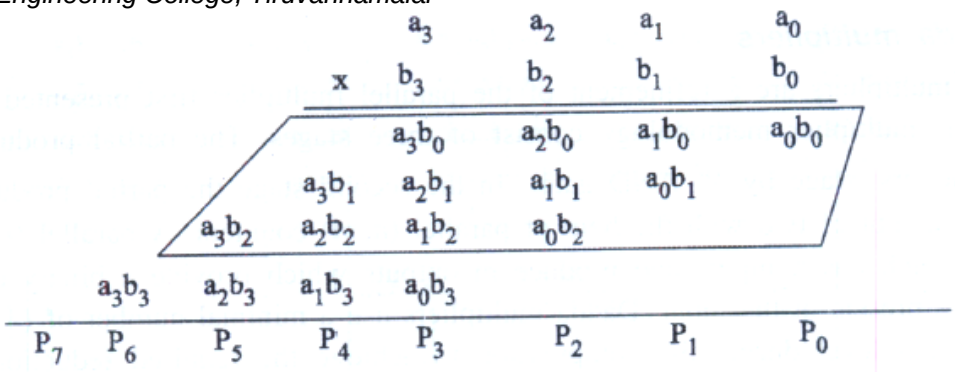
Consider the design of 4 x 4 multiplication using Wallace structure. The partial AND generation and Product generation are shown in Figure 4.34

Advantages of Wallace multiplier:

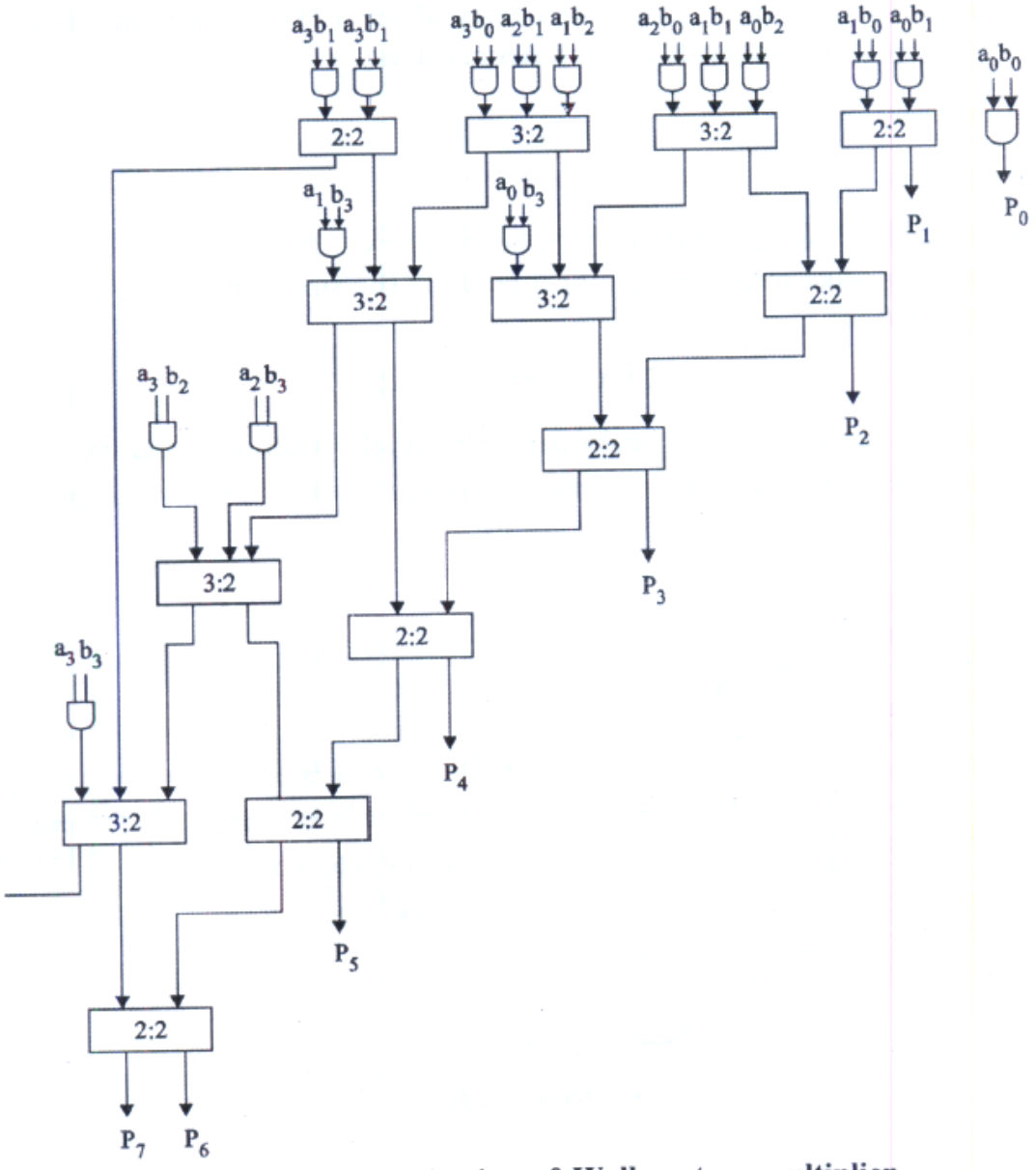
- Each layer of the tree reduces the number of Vectors by a factor of 3:2.
- Minimum propagation delay.
- The benefit of the Wallace tree is that, there are only O (log n) reduction layers, but adding partial products with regular adders would require (log n^2) time.

Disadvantages of Wallace multiplier:

- Wallace trees do not provide any advantage over ripple adder trees in many FPGAs.
- Due to the irregular routing the complexity of circuit increases proportionally with number of bits.
- Power dissipation is very high.



(a) Wallace multiplier process



(b) Gate-level realization of Wallace tree multiplier

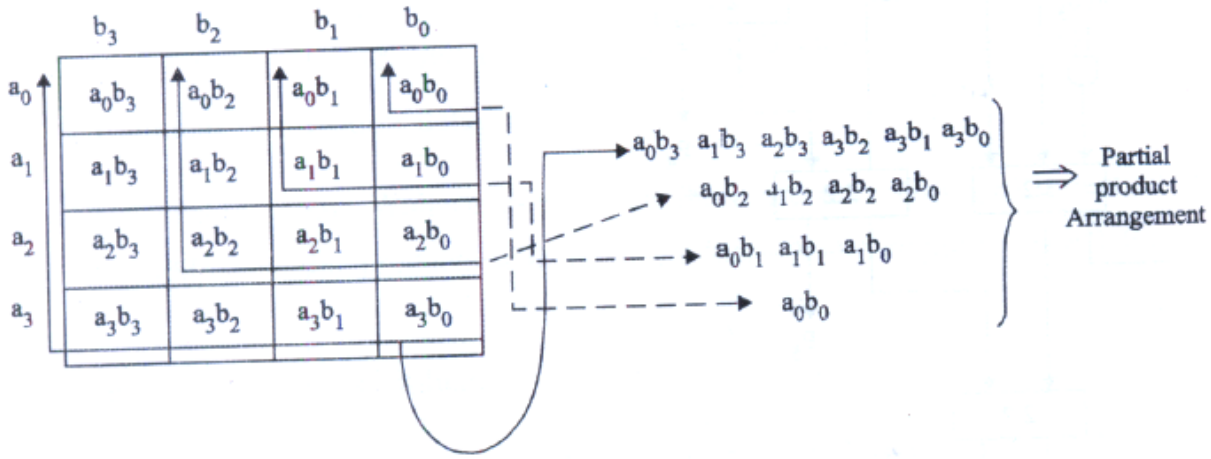
Figure 4.34 Wallace tree multiplier

Dadda multipliers

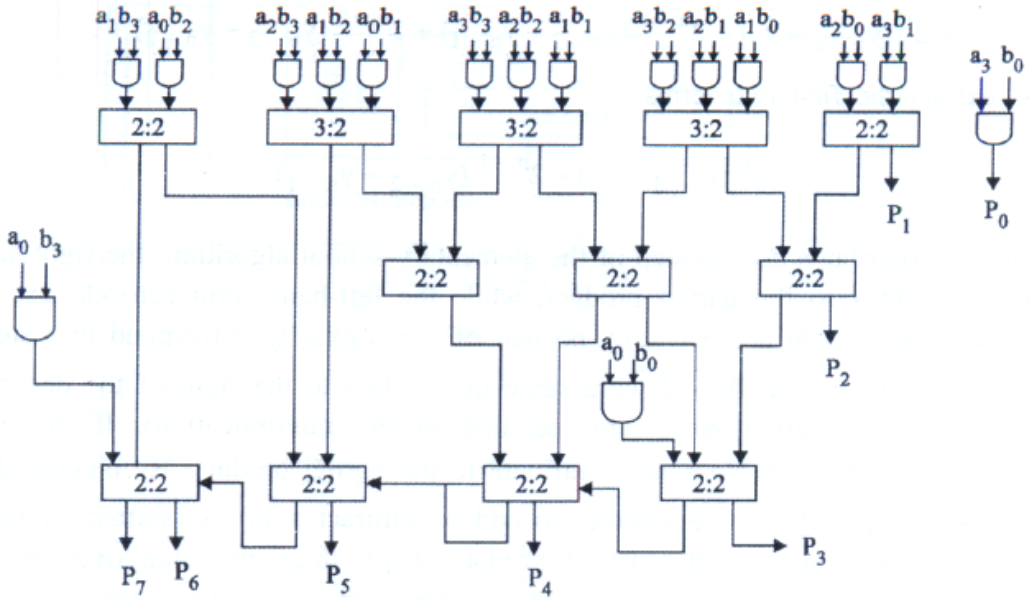
- Dadda multipliers are a refinement of the parallel multiplier first presented by

Wallace in 1964. This multiplier methodology consist of three stages.

- The partial product matrix is formed in the first stage by N^2 AND gates.
- In the second stage the partial product matrix is reduced to a height of two with the help of parallel (n, m) counters. A parallel (n, m) counter is a circuit which has n inputs and produce m outputs which provide a binary count of the number of l's present at the input.
- Dadda multipliers use a minimal number of (3,2) and (2,2) counters at each level during the compression to achieve the required reduction. (3,2) and (2,2) counts are full adder and half adder circuit.
- The number of (3,2) and (2,2) counters required for a Dadda multiplier depends on N, the number of bits of the operands and is determined as:
- (3,2) counters $=N^2-4N + 3$
- (2,2) counters $= N - 1$
- Once the matrix has been reduced to a height of two, the final stage consists of using a carry propagating adder to produce the final product. The size of the final carry propagating adder is determined as:
- CPA length =2N-2
- Consider the design of 4 x 4 multiplier. Initially the multiplier and multiplicand are arranged in the form of matrix. The structure realization of 4 x 4 Dadda multiplier is shown in Figure 4.35



(a) Design process



(b) Dadda multiplier with 3 : 2 and 2 : 2 compressor circuit

Figure 4.35 4 × 4 Dadda multiplier structure

Advantages:

- Dadda multipliers are slightly faster than the corresponding Wallace multipliers.
- The reduction level is much less than Wallace.

Disadvantages:

- Routing complexity is high.
- Irregular structure and exhibits high power dissipation.

Booth Multiplier

- Pipelining reduces cycle time but doesn't reduce the total time required for multiplication. One way to speed up multiplication is Booth encoding, which performs several steps of the multiplication at once booth's algorithm takes advantage of the fact that an adder-subtractor is nearly as fast and small as a simple adder. In elementary school algorithm, we shift the multiplicand x, then use one bit of the multiplier y if that shifted value is to be added into the partial product.
- The most common form of Booth's algorithm looks at three bits of the multiplier at a time to perform two stages of the multiplication.
- Consider the two's-complement representation of the multiplier y:

$$y = -2^{n}y_{n} + 2^{n-1}y_{n-1} + 2^{n-2}y_{n-2} + \dots$$

We can take advantage of the fact that $2^{a} = 2^{a+1} - 2^{a}$ to rewrite this as

$$y = 2^{n}(y_{n-1} - y_{n}) + 2^{n-1}(y_{n}-2-y_{n-1}) + 2^{n-2}(y_{n-3} - y_{n-2}) + \dots\dots$$

Now, extract the first two terms :

$$2^{n}(y_{n-1}-y_{n}) + 2^{n-1}(y_{n-2} - y_{n-1})$$

- Each term contributes to one step of the elementary-school algorithm: the right-hand term can be used to add x to the partial product, while the left-hand term can add 2x. (In fact, since y_{n-2} also appears in another term, no pair of terms exactly correspond to a step in the elementary school algorithm. But, if we assume that, y bits to the right of the decimal point are 0, then all the required terms are included in the multiplication).
- If, for example, $y_{n-1} = y_n$, the left-hand term does not contribute to the partial product. By picking three bits of y at a time, we can determine whether to add or subtract x or 2x (shifted by the proper amount, two bits per step) to the partial product. Each three-bit value overlaps with its neighbours by one bit. Table 4.4 shows the contributing term for each three bit code from y.

Figure 4.36 shows the detailed structure of a Booth multiplier. The multiplier bits control a multiplexer which determines what is added to or subtracted from the partial product. Booth's algorithm can be implemented in an array multiplier since the accumulation of partial products still forms the basic trapezoidal structure. In this case, a column of control bit generators on one side of the array analyze the triplet of y bits to determine the operation in that row.

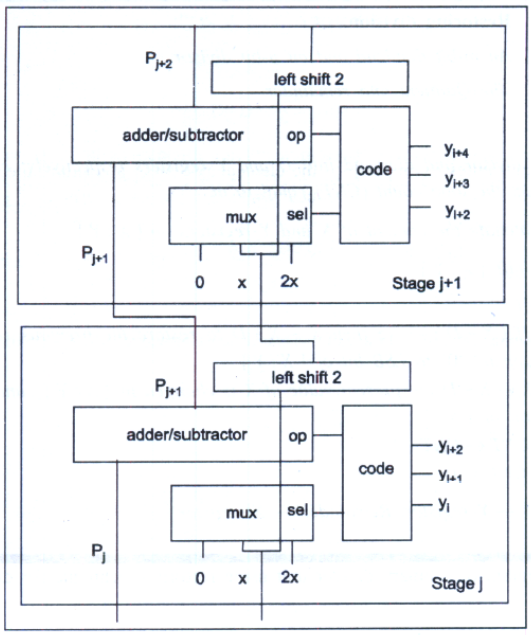


Figure 4.36 Structure of a Booth multiplier

14. Explain with an example how restoring and non-restoring algorithm is performed in division. [CO4- H3]

- The essential operations of division are a sequence of subtractions.
- There are two basic division algorithms known as restoring division algorithm and non-restoring division algorithm, respectively.

Restoring Division

- The dividend and divisor are assumed to be m and n bits, respectively. For simplicity, consider unsigned input numbers.

- In the restoring division method, the quotient bit is set to 1 if the result after subtracting the divisor from the dividend is greater than or equal to 0, otherwise the quotient bit is set to 0 and the divisor is added back to the dividend in order to restore the original dividend before proceeding to the next bit.
- Due to this restoring operation, the algorithm is called restoring division algorithm. In restoring division requires (3/2) n m-bits addition and subtractions on average to complete an n-bit division.

Algorithm : Restoring division

Input : An m-bit dividend and an n-bit divisor

Output : The Quotient and remainder

Begin

1. Load divisor and dividend into D and Y registers respectively; clear X register and set the loop count (CNT) equal to m.
2. Concatenate the content of X and Y registers --------> { X : Y }
3. Left shift { X: Y }
4. Repeat

4.1 If (X - D) = - ve then X <— X + D to restore the previous value of {X:Y and append '0' to LSB bit of { X: Y}

else X <—X - D the present value of X is changed in { X: Y } and append T to LSB bit of {X:Y}

4.2 CNT <— CNT - 1

until CNT – 0

5. Quotient ←Y register. Remainder <— X register

End.

A block diagram of a binary divider unit using restoring technique for division is shown in Figure 4.37.

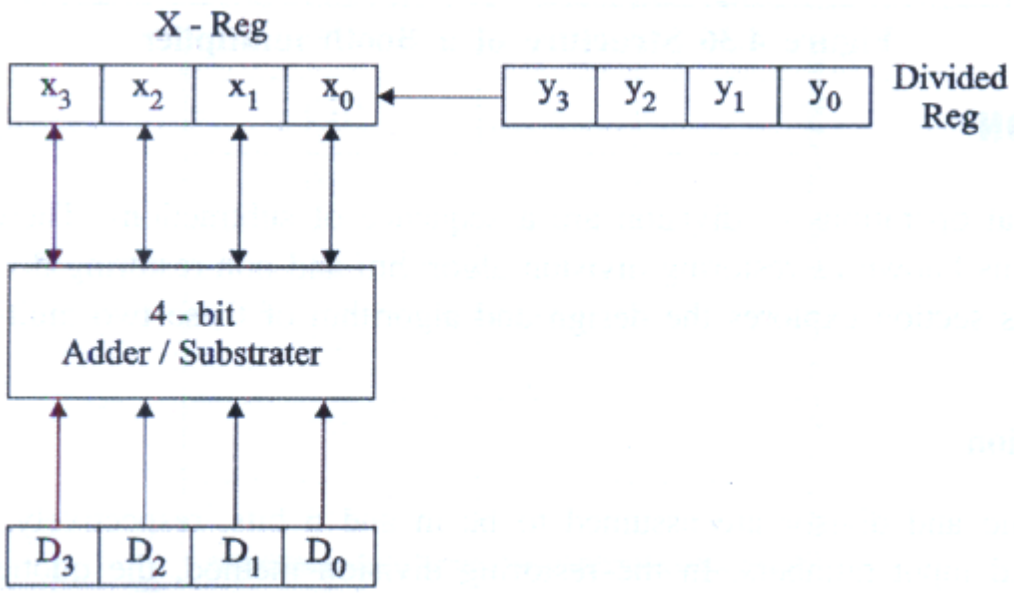


Figure 4.37 Block diagram of a 4-bit binary restoring divider

The above procedure can be understood in a better manner with the following example. Consider the division of 10100 (20) by 0101 (5). The dividend and divisor are stored in Y and D registers respectively and the X register is initially cleared to 0.

Y4 Y3 Y2 Y1Y0 = 10100

X3 X2 X1 X0 = 0000

D3 D2Dl D0 = 0101

Total number of iteration depends on the number of bits of dividend. In this case the total number of bits in the dividend is 5. Therefore 5 iteration of addition / subtraction procedure is carried out.

XY= 000010100

I cycle

XY= 00010100- [shift the combined contents of X and Y left by one bit]

X = X-D = 0001-0101 = 1100

Now X is negative, therefore add X + D to restore the previous value and append '0' to LSB of XY.

X = X + D = 1100 + 0101

X = 0001

XY= 0001 0100**[0]** [Append XY LSB '0']

II cycle

XY= 0010 1000- [Shift the combined contents of X and Y left by one bit]

X = X-D = 0010-0101

= 1101

Now X is negative, therefore add X + D to restore the previous value of XY append '0' o LSB of XY.

X = X + D = 1101+0101

X = 0010

XY = 0010 1000 **[0]** [Append XY LSB '0']

III cycle

XY=0101 0000 - [Shift the combined contents of X and Y left by one bit]

X = X-D = 0101-0101 = 0000

Now X is positive, the content X is changed to '0000' and the LSB of XY is set to 1.

XY = 0000 0000 **[1]** [Append XY LSB '1']

IV cycle

XY= 0000 0001 - [Shift the combined contents of X and Y left by one bit]

X = X-D = 0000 - 0101 = 1011

Now X is negative, therefore add X-X + D to restore the previous value of XY append '0' to LSB of XY.

X = X + D

= 1011+0101

X = 0000

XY = 0000 0001 **[0]** [Append XY LSB '0']

V cycle

XY=0000 0010- [Shift the combined contents of X and Y left by one bit]

X = X-D = 0000-0101 = 1011

Now X is negative, therefore add X-X+D to restore the previous value of XY append '0' to LSB of XY.

X = X + D

= 1011+0101

X = 0000

XY= 0000 0010 **[0]** [Append XY LSB '0']

Quotient = Y Register

(i.e.,) Q = 00100---> 4

Remainder = X Registers

(i.e) R = 0000 → 0

Non-restoring Division

- The dividend and divider are assumed to be m and n-bits.
- In the non-restoring division method, the divisor is not added back to the { X: Y} when X - D < 0.
- Instead, the result X-D is shifted left one bit and then added to the divisor D. The result is the same as that of divisor (D) is added back to the { X: Y} and then performs the subtraction. Initially shift the content of { X: Y} one bit left.
- If the MSB of X register is '0' then perform subtraction of X-D and append '0' to LSB of { X: Y} else perform X + D and append '1' to LSB of {X: Y}

Algorithm : Non restoring division

Input : An in-bit dividend and an n-bit divisor

Output: The Quotient and remainder

Begin

1. Load divisor and dividend into D and Y registers respectively; clear X register and set the loop count (CNT) equal to m. The width of X and Y should be same.

2. Concatenate the content of X and Y registers —» { X : Y}

3. Left shift {X:Y} one bit.

4, Repeat

4.1 Check MSB of X If X[m - 1] = '0' then sub X~X-D and append { X: Y} LSB '1' else add X=X+D and append { X : Y} LSB '0'.

4.2 CNT <— CNT - 1

until CNT = 0

Normalize X-registers

5. Quotient <— Y register; Remainder ← X register.

End

- The sequential implementation of non-restoring division algorithm is shown in Figure 4.38. Four register are required. Register D contains the divisor. Y is the register that stores the dividend value. The concatenated value of X and Y is stored in A register. Initially the X register is cleared.
- The n-bit adder/subtracted is utilized to perform the addition or subtraction needed by the algorithm and is controlled by the MSB bit of the X register (or) {X:Y}. After a divisor and a dividend are loaded into the respective registers shift operation is performed so that the content of { X: Y} in A will undergo subtraction with D register. The total number of clock cycles required to perform division is 2n clock cycles. The above procedure can be understood in a better manner with the following example.
- The inputs are same as restoring division method. The dividend and divisor are 10100 (20) and 0101 (5). The dividend and divisor are stored in Y and D register respectively and the X register is initially cleared to 0. Here the width of all register are same.

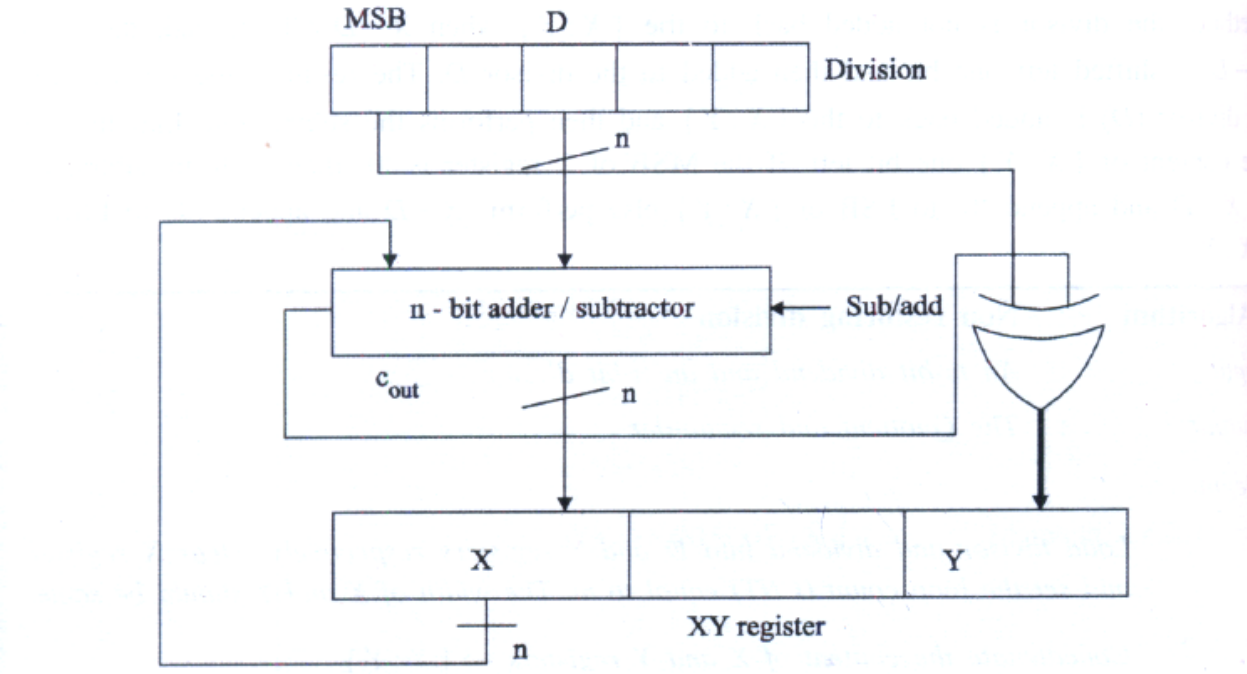


Figure 4.38 The sequential implementation of the non-restoring division algorithm.

Y4 Y3 Y2 Y1 Y0= 10100

X4 X3 X2 X1 X0 = 00000

D4 D3 D2 D1 D0 = 00101

XY= 0000010100

I cycle

XY= 000010100 - [shift the combined contents of X and Y left by one bit] Check MSB of X register

X4 = 0, So perform subtraction

X = X-D = 00001 -00101 = 11100

XY= 11100

Now again check MSB of X which is '0' therefore the LSB of XY is appended with

XY = 11100 0100 **[0]**

II cycle

XY= 11000 1000 - [Shift the combined contents of X and Y left by one bit] Check MSB of X register

X4 = 1, So perform addition X = X + D

= 11000 + 00101

= 11101

Now again check MSB of X which is T therefore the LSB of XY is appended with '0'.

XY= 11101 1000 **[0]**

III cycle

XY= 11011 0000 - [Shift the combined contents of X and Y left by one bit] Check MSB of X register

X4 = 1, So perform addition

X = X + D

= 11011+00101

= 00000

Now again check MSB of X which is '0' therefore the LSB of XY is appended with '1'.

XY= 0000 0000 **[1]**

IV cycle

XY=000000001 - [Shift the combined contents of X and Y left by one bit]

Check MSB of X register

X4 = 0, So perform subtraction

X = X-D

= 00000 - 00101

= 11011

Now again check MSB of X which is T therefore the LSB of XY is appended with '0'

XY = 11011 0001 **[0]**

V cycle

XY=10110 0010- [Shift the combined contents of X and Y left by one bit]

Check MSB of X register X4 = 1, So perform addition

X = X + D

= 10110 + 00101

= 11011

Now again check MSB of X which is T therefore the LSB of XY is appended with '0'.

XY= 11011 00010 **[0]**

Finally normalize X register by adding

X = X + D

= 11011+00101

= 00000

Quotient = Y

(i.e) Q = 00100 [4]
Remainder = X
(i.e) R = 00000 [0]

15. Explain power and speed trade-offs (optimization) in datapath structures.
[CO4- H1]

- In most digital design the trade-off exit between Area-speed-power constrains. Area optimization can be achieved through choosing proper logic technology, logic style and architecture selection.
- digital designs are either latency or throughout constrained.
- A latency constrained design should complete the task or function within the specified time limit or deadline, whereas a throughput - constrained designs must maintain a required data throughput.
- The architectural optimization techniques such as pipelining or parallelization will be effective for the throughput - constrained scenario, but it may not be applicable to the latency - constrained case.
- With a fixed architecture of the datapath, speed, area and power can be traded off through the choice of the supply voltages, transistor thresholds and devices sizes, several optimization techniques for constant throughput and variable throughput is listed in Table 4.6.

	Constant throughput/Latency		Variable throughput/Latency
	Design time	Sleep mode	Run time
Active	Lower V_{DD}, multi-V_{DD}, Transistor sizing, logic optimizations	Clock gating	Dynamic voltage scaling
Leakage	Multi-V_{th} + Active Techniques	Sleep transistors, Variable V_{th}	Variable V_{th} + Active techniques.

Table 4.6 power optimization techniques.

The power optimization (minimization) techniques can be achieved by:

Optimization during Enable time

Some parameters in the design process can be optimized statically and dynamically. Some design techniques are implemented (or enabled) at design time. So the parameters of transistor width and length can be fixed at the time of design. Similarly the threshold voltage and supply voltages can be assigned statically during the design phase or changed dynamically at run time. The functional module power optimization can be achieved by turning OFF the devices to 'sleep mode' when the modules are in idle (or) standby condition. A common approach to reduce power in idle mode is the clock gating technique. In this method the main clock connection to a module is turned

off whenever the block is idle. But this approach does not reduce the leakage power of the idle block.

Optimization of targeted dissipation source

This type of power optimization concerns the source of power dissipation they address: active (dynamic) power or leakage (static) power. The dynamic power deals with the charging and dischanging the node capacitances.

Design-time power-Reduction Techniques

Reducing the supply voltage

The dynamic power of digital chips can be expressed as $p = \alpha CV^2 f$. Due to the quadratic effect of the voltage can achieve dramatic savings. The easiest method to achieve this is to reduce the operating voltage of the CMOS circuit.

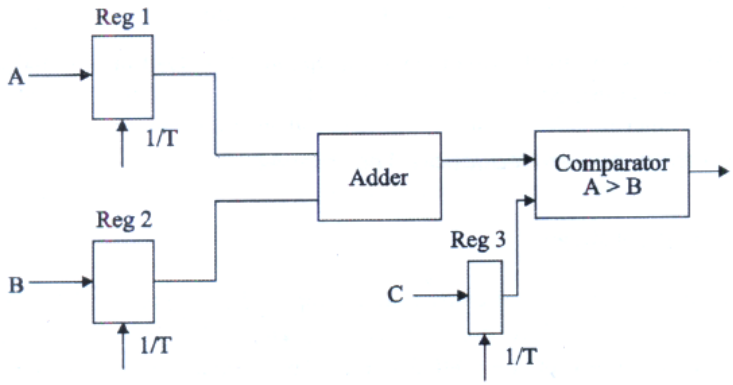
Logical optimization

In the most general sense, logical optimization can be achieved by transforming one logic circuit to another that is functionally equivalent. Therefore, logical optimization techniques use logical restructuring rules to transform one network to another that is functionally equivalent to produce effective power optimization for reduced supply voltages.

Architectural optimization

Architectural optimization is also called block level or macro-level design optimization which can be achieved through pipelining and parallelism. Parallelism has been traditionally used to improve the computation throughput to high performance digital systems at reduced operating voltages. Consider a datapath consisting of adder and comparator circuit as shown in Figure 4.44. The power dissipation of this datapath element is expressed as a function of the number of bits of the components and their operating frequencies. So for single datapath element the power dissipation will be given by

$$P_{\text{datapath}-\text{single}} = \alpha_{0\to 1} CV^2 f$$



(a) A single datapath element

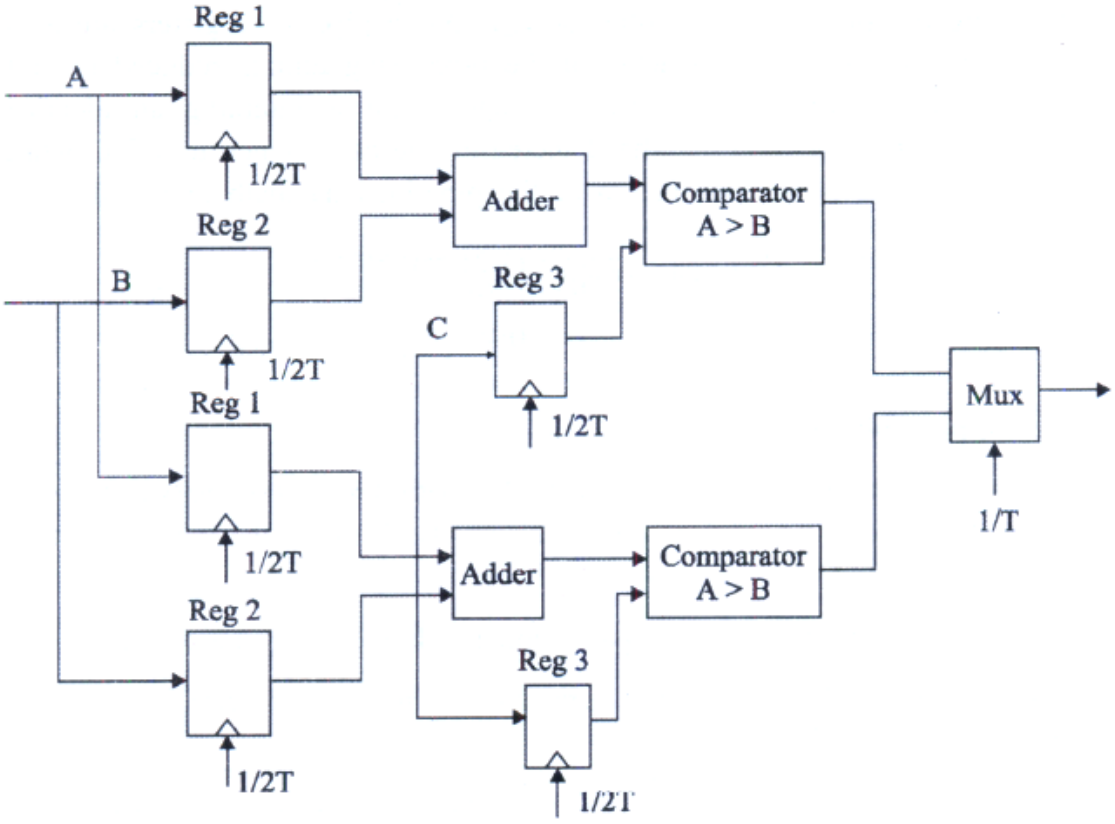
If we double the number of datapath element, each unit can be operated at half the frequency f/2. The derived system throughput is still maintained by multiplexing the

outputs of the two datapath units as shown in Figure 4.44 (b). Since the operating frequency requirement is now reduced to half, the system operating voltage can be reduced to conserve power. Let us assume that the average capacitance switched in parallel datapath unit is 2.2 C, slightly more than double due to some overhead in multiplexing and demultiplexing data. Suppose that the reduced operating frequency allows to reduce the operating voltage of the parallel datapath unit to 0.6 V.

The power dissipation in parallel datapath unit will be

$$P_{datapath-parallel} = \alpha_{0\ 1} (2.2C) (0.6V)^2 (0.5f)$$

$$= 0.396\ P_{datapath} - \text{single} \text{ ---------- (4.27)}$$



(b) Parallel implementation of datapaht element

Figure 4.44 power dissipation of single and parallel datapath unit

From the above expression it is evident that a power reduction of approximately 60% compared to single datapath unit. The power saving is paid by a substantial increase in circuit area.

Power reduction using pipelining

The parallelism technique increase the chip area by atleast twice. If the area penalty of a parallel system is prohibitive, pipelining can give similar trade-off results with less area overhead but more complexity in controller design. When pipeline registers are added to the single datapath unit, the propagation delay of the processing unit is reduced to half

and the pipelined system can be operated at lower voltage while maintaining an identical overall throughput. Suppose the pipelining overhead increases the capacitance to 1.2 C while reduces the voltage to 0.6 V, the power dissipation of the pipelined datapath unit will be

$$P_{datapath} \text{ - Parallel} = (1.2C)(0.6V)^2 f$$
$$= 0.432\ P_{datapath} \text{ - single}$$

Multiple supply voltages

Reduced supply evenly lowers the power dissipation of all logic gates, while evently increasing their delay. A better approach is to selectively decrease the supply voltage on some of the gates: those which correspond to fast paths and finish the computation early and those with gates that drive large capacitances, have the largest benefit for the same delay increment..

Module-level voltage selection

Consider a digital circuit consisting of datapath unit and controller unit as shown in Figure 4.45. Both the units are said to operated with the same voltage but the critical path parameter is different. For simplicity assume the datapath unit has a critical path of 10 ns and a controller block with a much shorter critical path of 4 ns. Since the controller block finishes early, its supply voltage can be lowered. This may increases the controller critical path delay to 10 ns and lowers its power dissipation by more than five times. When combining multiple, supply voltages on a chip, level converters are required when-ever a module at the lower supply has to drive a gate at the higher voltage.

Unit V

Implementation strategies

Part – A

1. Differentiate between channeled and channel less gate array. [CO5-L2]

S.no	Channeled gate array	Channel less gate array
1	Only the interconnect is customized	Only the top few mask layers are customized
2	The interconnect uses predefined spaces between rows of base cells	No predefined areas are set aside for routing between cells
3	Routing is done using spaces	Routing is done using the area of transistors unused
4	Logic density is less	Logic density is higher

2. What are the different levels of design abstraction at physical design? **[CO5-L1]**

- Architectural or functional unit
- Register Transfer-level (RTL)
- Logic level
- Circuit level

3. What are macros? [CO5-L1]

The logic cells in a gate-array are often called macros.

4. What are programmable Interconnects? [CO5-L1]

In a PAL, the device is programmed by changing the characteristics if the switching element. An alternative would be to program the routing.

5. What are the types of ASICs? [CO5-L1]

Types of ASICs are

- ☐ Full custom ASICs
- ☐ Semi custom ASICs

6. What are the types of programmable devices? [CO5-L1]

Types of programmable devices are

- ☐ Programmable logic structure
- ☐ Programmable Interconnect
- ☐ Reprogrammable Gate Array

7. What are the features of standard celled ASICs ? [CO5-L1]

- ☐ All mask layers are customized-transistors and interconnect.
- ☐ Custom blocks can be embedded
- ☐ Manufacturing lead time is about eight weeks.

8. What are the characteristics of FPGA? [CO5-L1]

- ☐ None of the mask layers are customized

- A method of programming the basic logic cells and the interconnect.
- The core is a array of programmable basic logic cells that can implement combinational as well as sequential logic (flipflops).
- A matrix of programmable interconnect surrounds the basic logic cells.
- Design turn around is a few hours.

9. What is programmable logic array? [CO5-L1]

A programmable logic array (PLA) is a programmable device used to implement combinational logic circuits. The PLA has a set of programmable AND planes, which link to a programmable OR planes, which can then be conditionally complemented to produce an output. This layout allows for a large number of logic functions to be synthesized in the sum of products (sometimes product of sums) canonical forms.

10. What is meant by programmable logic plane? [CO5-L1]

The programmable logic plane is programmable read only memory(PROM) array that allows the signals present on the devices pins to be routed to an output logic macro cell.

11. Give the application of PLA. [CO5-L1]

Design and testing of digital circuits.

12.Give the different types of ASIC. [CO5-L2]

1. Full custom ASICs
2. Semicustom ASICs
   - Standard cell based ASICs
   - Gate-array based ASICs
3. Programmable ASICs
   - Programmable Logic Device (PLD)
   - Field Programmable Gate Array (FPGA).

13. What is the full custom ASIC design? [CO5-L1]

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.

14. What is standard cell-based ASIC design? [CO5-L1-Nov/Dec 2016]

A cell-based ASIC (CBIC) uses predefined logic cells known as Standard Cells.The standard cell areas are also called flexible block 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.

15. What is FPGA? [CO5-L1]

A Field Programmable Gate Array (FPGA) is a programmable logic device that supports implementation of relatively large logic circuits. FPGA can be used to implement a logic circuit with more than 20,000 gates whereas a CPLD can implement circuits of up to about 20,000 equivalent gates.

16. What are the different methods of programming of PALs ? [CO5-L1]

The programming of PALs in done in three ways:

1. Fusible links
2. UV-erasable EPROM

3. EEPROM(E2PROM) – Electrically Erasable Programmable ROM.

17. What is an antifuse ? [CO5-L1]

An antifuse is normally high resistance (>100MΩ). on application of appropriate programming voltages, the antifuse is changed permanently to a low-resistance structure(200-
500Ω) .

18. What are the difference between ASIC and FPGA. [CO5-L1]

	ASIC	**FPGA**
1	An ASIC is a unique type of integrated circuit meet for specific	An FPGA is a reprogrammable integrated circuit
2	An ASIC can no longer be altered once created	FPGA is alterable
3	An ASIC wastes very little material, recurring cost are very low.	FPGA is not efficient terms of use of materials certain number of component
4	Cost of ASIC is low only when it is produced in large quantity	FPGA is better than a asic when builds low volume production circuits
5	ASIC can't be used to test FPGA	ASIC are tested on FPGA before completing.
6	ASIC are not suitable for research & development purpose, as they are not reconfigurable.	FPGA are useful for research & development activities. prototype fabrication using FPGA is affordable & fast .

19.Give the steps in ASIC design flow? [CO5-L1]

1. Design entry
2. Logic synthesis system partitioning
3. Prelayout simulation
4. Floor planning
5. Placement
6. Routing
7. Extraction
8. Post layout simulation

Part -B

1. Explain application specific integrated circuit (ASIC) design with its types. [CO5-H1]

Application Specific Integrated Circuit (ASIC)

- The term ASIC ("a - Sick") stands for 'application - specific integrated circuit'**.** An ASIC is basically an integrated circuit designed specifically for a special purpose or application. ASIC is built only for one and only one customer.

- The use of ASICs improves performance over general-purpose CPUs, because ASICs are "hardwired" to do a specific job and do not incur the overhead of fetching and interpreting stored instructions.
- An example of an ASIC is an IC designed for a specific line of cellular phones of a company, whereby no other products can use it except the cell phones belonging to that product line.
- The opposite of an ASIC is a standard product or general purpose IC, such as a logic gate or a general purpose microcontroller.
- Aside from the nature of its application, an ASIC differs from a standard product in the nature of its availability. The intellectual property, design database, and deployment of an ASIC is usually controlled by just a single entity or company, which is generally the end-user of the ASIC too.
- Thus, an ASIC is proprietary by nature and not available to the general public. A standard product, on the other hand, is produced by the manufacturer for sale to the general public. Standard products are therefore readily available for use by anybody for a wider range of applications.
- The first ASIC's, known as uncommitted logic array or ULA's, utilized gate array technology.
- Having up to a few thousand gates, they were customized by varying the mask for metal interconnections.
- Thus, the functionality of such a device can be varied by modifying which nodes in the circuit are connected and which are not. Later versions became more generalized, customization of which involve variations in both the metal and polysilicon layers

Examples of ASIC's include

(i) an IC that encodes and decodes digital data using a proprietary encoding/decoding algorithm
(ii) A medical IC designed to monitor a specific human biometric parameter.
(iii) An IC designed to serve a special function within a factory automation system.
(iv) An amplifier IC designed to meet certain specifications not available in standard amplifier product.
(v) A proprietary system-on-a chip (SOC).
(vi) An IC that's custom - made for particular automated test equipment.

Advantages of ASIC

(i) Used for specific application.
(ii) Big reductions in area and power consumption.
(iii) Better control of electrical parameters.
(iv) The compactness of the circuit in an ASIC could also results in better noise, bandwidth and leakage performance.
(v) ASIC can improve speed and minimize electric power to operate.

Types of ASICs

The ASIC design of digital circuit can be classified as shown in Figure 5.1

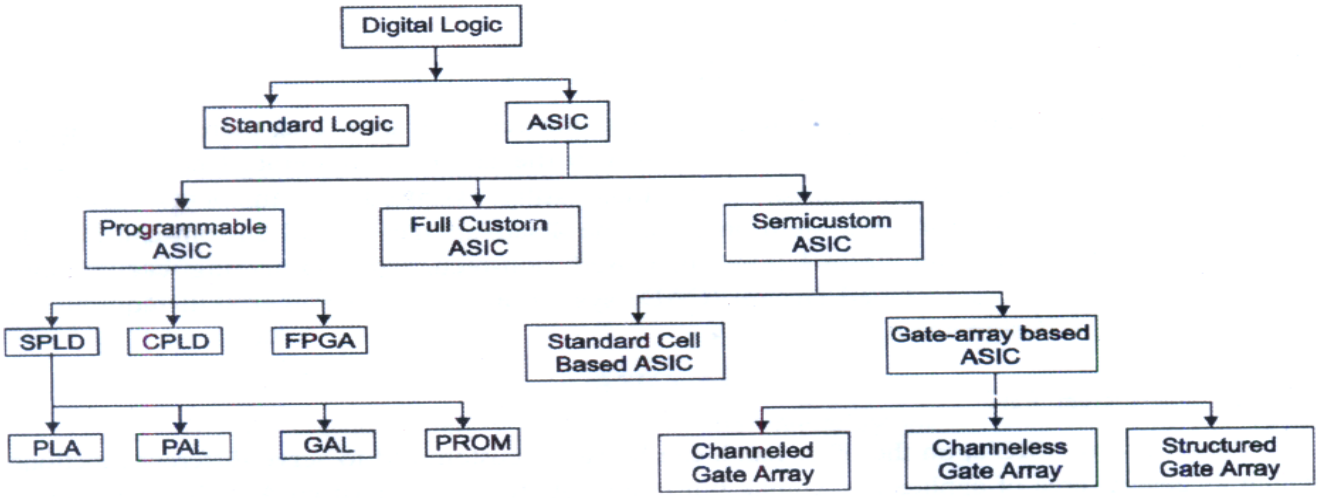


Figure:Types of ASIC Design

Full Custom ASIC

- Full-custom ASIC's are those that are entirely tailor-fitted to a particular application from the very start. Since its ultimate design and functionality is pre-specified by the user, it is manufactured with all the photolithographic layers of the device already fully defined, just like most off-the-shelf general purpose ICs.
- The use of predefined masks for manufacturing leaves no option for circuit modification during fabrication, except perhaps for some minor fine-tuning or calibration.
- This means that a full-custom ASIC cannot be modified to suit different applications, and is generally produced as a single, specific product for a particular application only.
- ICs are made on a thin, circular silicon wafer, with each wafer holding hundreds of die. The transistors and wiring are made from many layers built on top of one another.
- Each successive mask layer has apattern that is defined using a mask similar to a glass photographic slide. The first half-dozen or so layers define the transistors. The last half-dozen or so layers define the metal wires between the transistors (the interconnect).
- A full-custom IC includes some (possibly all) logic cells that are customized and all mask layers that are customized.
- A microprocessor is an example of a full-custom IC -designers spend many hours squeezing the most-out of every last square micron of microprocessor chip space by hand.
- Customizing all of the IC features in this way allows designers to include analog circuits, optimized memory cells or mechanical structures on an IC, for example.

The important features of this type of ASICs are as follows

- ✓ Some (possibly all) logic cells that are customised.
- ✓ All mask layers that are customized.
- ✓ Manufacturing lead time is typically eight-weeks.

- The benefits of full-custom design usually include reduced area (and therefore recurring component cost), performance improvements and also the ability to integrate (include) analog components and other pre-designed (and thus fully verified) components such as microprocessor cores that form a system-on-chip.
- The disadvantages of full-custom can include increased manufacturing and design time, increased non-recurring engineering costs, more complexity in the Computer Aided Design (CAD) system and a much higher skill requirement on the part of the design team.

Semi-custom ASIC

- Semi-custom ASIC's, on the other hand, can be partly customized to serve different functions within its general area of application.
- Unlike full-custom ASIC's, semi-custom ASIC's are designed to allow a certain degree of modification during the manufacturing process.
- A semi-custom ASIC is manufactured with the masks for the diffused layers already fully defined, so the transistors and other active components of the circuit are already fixed for that semi-custom ASIC design.
- The customization of the final ASIC product to the intended application is done by varying the masks of the interconnection layers, e.g., the metallization layers.

Standard cell - Based ASICs

- A cell-based or CBIC uses predesigned logic cells (AND gates, OR gates, multiplexers and flip-flops for example) known as standard cells.
- Each standard cell vendor has its own library of circuits that range from primitive logic gates to more complex functions such as memory blocks and microprocessor cores.
- The standard-cell areas (also called flexible blocks) in a CBIC are built of rows of standard cells. The standard-cell areas may be used in combination with larger pre-designed cells, perhaps microcontrollers or even microprocessors, known as mega cells.
- Mega cells are also called mega functions full custom blocks, or Functional Standard Blocks (FSBs). Based on the customers design, the required circuits are placed on the chip and connected using "place-and-route" software.

GATE Array - Based ASICs

- Gate Array design is a manufacturing method in which the diffused layers (ie) transistors and other active devices, are predefined and wafers containing such devices are held in stock prior to metallization, in other words, unconnected.
- The predefined pattern of transistors on a gate array is the base array sometimes called primitive cell. The physical design process then defines the interconnections of the final device.
- For most ASIC manufactures, this consists of from two to as many as five metal layers, each metal layer running perpendicular to the one below it. Only the top few layers of metal, which define the interconnect between transistors, are defined by the designer using custom masks.

- To distinguish this type of gate array from other types of gate array, it is often called a Masked Gate Array (MGA).
- Non-recurring engineering costs are much lower as photo-lithographic masks are required only for the metal layers, and production cycles are much shorter as metallization is a comparatively quick process.
- It is also important to the designer that minimal propagation delays can be achieved in ASICs versus the FPGA solutions available in the market-place.
- Gate Array ASICs are always a compromise as mapping a given design onto what a manufacturer held as a stock wafer never gives 100% utilization.
- Often difficulties in routing the interconnect require migration onto a larger array device with consequent increase in the piece part price.
- These difficulties are often a result of the layout software used to develop the interconnect. The turn around time to design a ASIC, will be few days or at most a couple of weeks.

2. Explain the classification of gate array based ASICs. [CO5-H1]

Depending upon the arrangement of transistor, the gate-array based ASICs can be classified into:

- Channeled gate arrays
- Channelless gate arrays
- Structured gate arrays

1. Channeled Gate Array

Figure. 5.2 show a channeled gate array. The important features of this type of MGA are:

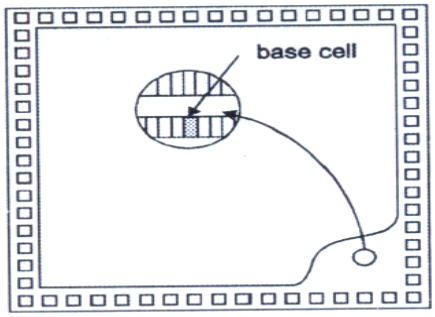


Figure 5.2 A channeled gate-array die.

- ✓ Only the interconnect is customized
- ✓ The interconnect uses predefined spaces between rows of base cells.
- ✓ Manufacturing lead time is between two days and two weeks.

A channeled gate array is similar to a CBIC both use rows of cells separated by channels used for interconnect. One difference is that the space for interconnect between rows of cells are fixed in height in a channeled gate array, whereas the space between rows of cells may be adjusted in CBIC.

2. ***Channelless Gate Array***

Figure 5.3 shows a channelless gate array. The important features of this types of MGA are as follows:

- ✓ Only some (the top few) mask layers are customized and the interconnects are customized.
- ✓ Manufacturing lead time is between two days and two weeks.

- The key difference between a channelless gate array and channeled gate array is that there are no predefined areas set aside for routing between cells on a channelless gate array. Instead we route over the top of the gate-array devices.
- We can do this because we customize the contact layers that define the connections between metal, the first layer of metal, and the transistors.
- When we use an area of transistors for routing in a channelless array, we do not make any contacts to the devices lying underneath; we simply leave the transistors unused.
- The logic density the amount of logic that can be implemented in a given silicon area is higher for channelless gate array architecture than channeled gate array.
- Because in channeled gate array the contact mask is customized whereas in channelless gate array the contact mask is not customized

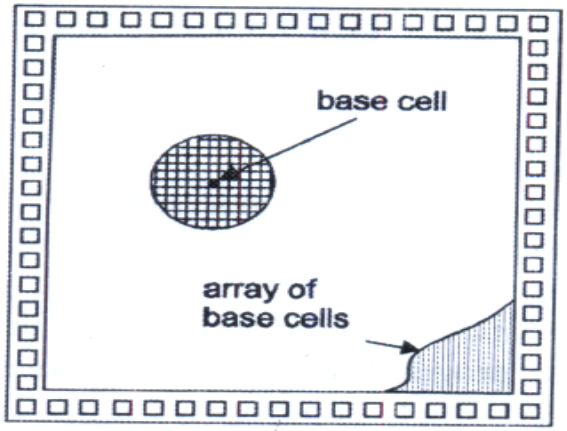


Figure 5.3 A Channelless Gate Array

3. Structured Gate Array

Structured gate array combines some of the features of CBICs and MGAs. Figure 5.4 shows a structured gate array. The important features of this type of MGA are the following:

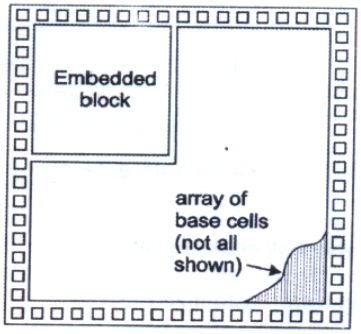


Figure 5.4 A structured or embedded gate-array

- ✓ Only the interconnect is customized
- ✓ Custom blocks can be embedded.
- ✓ Manufacturing lead time is between two days and two weeks.

- An embedded gate array gives the improved area efficiency and increased performance of a CBIC but with the lower cost and faster turnaround of an MGA.
- One disadvantage of an embedded gate array is that the embedded function is fixed.
- For example, if an embedded gate array contains an area set aside for a 32 k-bit memory, but we only need a 16 k-bit memory, then we may have to waste half of the embedded memory function.
- However, this may still be more efficient and cheaper than implementing a 32 k-bit memory using macros on a Sea-of-Gate (SOG) array.

3. Explain Programmable Logic Devices (PLDS). [CO5- H1]

Programmable logic devices are the standard ICs which can be programmed to implement the desired functions. However, PLDs may also be configured or programmed to create a part customized to a specific application, and so they also considered as a family of ASICs. PLDs use different technologies to allow programming of the device.

The important features of the PLDs are outlined below.

- ✓ No customized mask layers or logic cells
- ✓ Fast design turnaround
- ✓ A single large block of programmable interconnect
- ✓ A matrix of logic macro cells that usually consist of programmable array logic followed by a flip-flop or latch.

The different types of PLDs are read-only memory (ROM),Programmable array logic(PAL) and Programmable logic array(PLA).

Read-only memory (R O M)

- The simplest type of programmable IC is a read-only memory (ROM) which can be programmed once to store binary data. The most common types of ROM use a metal fuse that can be blown permanently (programmable ROM or PROM).
- An electrically programmable ROM , or EPROM , uses programmable M08 transistors whose characteristics are altered by applying a high voltage. One can erase an EPROM either by using another high voltage (an electrically erasable PROM, or EEPROM) or by exposing the device to ultraviolet light (UV-erasable PROM, or UVPROM).
- There is another type of ROM that can be placed on any ASIC - a mask-programmable ROM (mask-programmed ROM or masked ROM). A masked ROM is a regular array of transistors permanently programmed using custom mask patterns. An embedded masked ROM is thus a large, specialized, logic cell.

Programmable array logic (PAL)

Programmable array logic (PAL) is another programmable architecture which can be programmed to implement the desired function. The PAL architecture is shown in

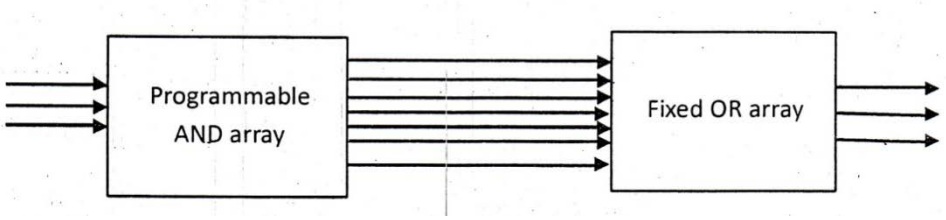
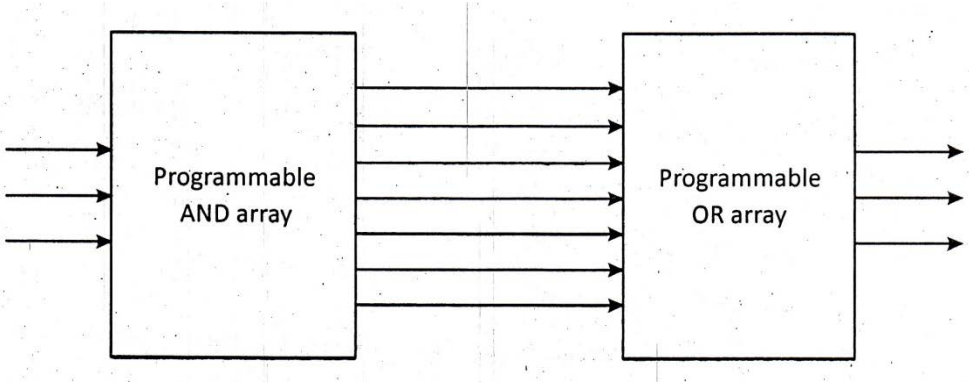


figure. It has a programmable AND array followed by fixed OR array. A PAL can also include registers (flip-flops) to store the current state information so that you can use a PAL to make a complete state machine.

Fig : Programmable array logic.



Programmable logic array (PLA)

In contrast to the PAL, Programmable logic array (PLA) has both programmable AND array and OR array. The PLA architecture is shown in figure 5.7.

Comparison between PROM, PLA, PAL:

	PROM	PLA	PAL
1	And Array is fixed & OR array is programmable	Both AND,OR array programmable	OR array fixed, And gate is programmable.
2	Cheaper and simple to use	Costliest & complex than PAL PROM.	Cheaper and simpler
3	All min terms are decoded	AND array programmed to get desired min terms	AND array can be programmed to get desired min terms.

4. Write short notes about CPLD: (complex programmable logic array).[CO5-L2]

- We could put several PAL/PLA on one chip and put an interconnection matrix between them is called CPLD.

- Each "microcell"can be independently configured for combinational (or) registered functionality, and receives global clock, output enable and S-R signals.
- Microcell's had negligible delay
- PAL like structure of a "functional block" receives 54 complemented, uncomplemented inputs and can form upto 90 product terms from the inputs.
- In the center of the design is a "programmable interconnect". This inter connect allows connections to the logic block microcells & I/O cell arrays.
- CPLD utilize nonvolatile memory such as EPROM, EEPROM, flash memory

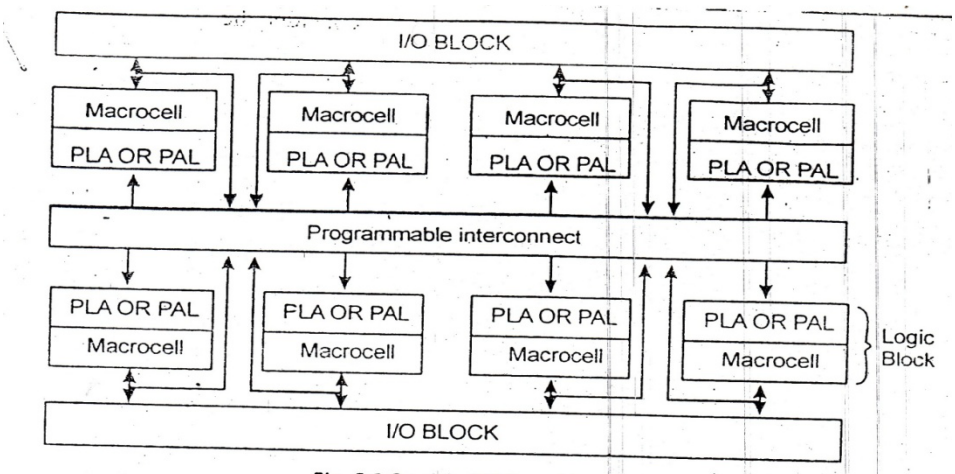


Fig. 5.8 Generic CPLD architecture

5. Write the various methods of programming of PALs (programmable array logic) [CO5-H1]

The programming of PALs is done in three main ways:

- Fusible links
- UV - erasable EPROM
- EEPROM *(E^2 PROM)* - Electrically Erasable Programmable ROM.

Fusible Links

- Fusible links uses metal-metal antifuse. The link is an alloy of tungsten, titanium, and silicon with a bulk resistance of about 500 μΩ cm.
- Link is an antifuse is normally an open circuit until you force a programming current through it.
- The fabrication process and the programming current control the average resistance of a blown antifuse. The fusible links separate interconnect wires on the FPGA chip and the programmer blows an antifuse to make a permanent connection.
- Once an antifuse is programmed the process cannot be reversed this is One Time Programmable (OTP) technology.
- Field effect or bipolar transistors are used in place of diodes in integrated circuit PROMs. PROMs (Programmed by the User) have each address select line

connected to a MOSFET gate or bipolar transistor base, with the source (emitter) grounded and the drain (collector) connected to the vertical data lines.

- A fuse link exists between ground and the MOSFET source (or bipolar transistor emitter). Because PROMs are programmed by the user, each address select and data line intersection has its own fused MOSFET or transistor.
- When the fuse is intact the memory cell is configured as a logical 1, and when the fuse is blown (open circuit) the memory cell is configured as a logical 0, logical 0s are programmed by selecting the appropriate select line and then driving the vertical data line with a pulse of high current Figure 5.12 illustrates a PROM fused MOSFET memory cell.

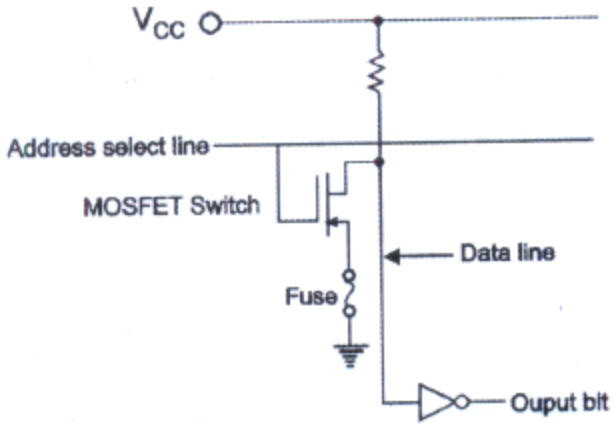
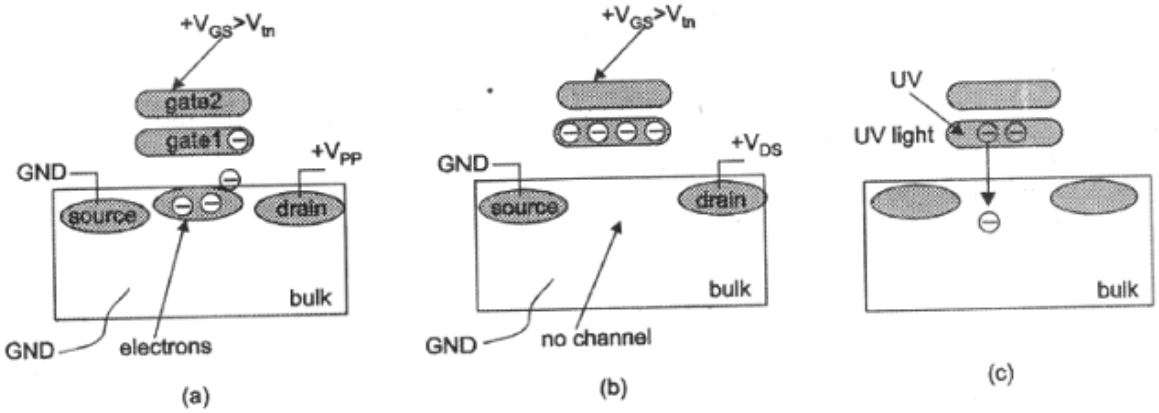


Figure 5.12 Fused PROM cell

EPROM and EEPROM Technology

- The EPROM cell is almost small as an antifuse. An EPROM transistor looks like a normal MOS transistor except it has a second, floating-gate (gate 1 in Figure 5.13).
- Applying a programming voltage V_{PP}(usually greater than 12V) to the drain of the n-channel EPROM transistor programs the EPROM cell.
- A high electric field causes electrons flowing toward the drain to move so fast they "jump" across the insulating gate oxide where they are trapped on the bottom, floating-gate.
- These energetic electrons are hot and the effect is known as hot-electron injection or avalanche injection. EPROM technology is sometimes called floating-gate avalanche MOS (FAMOS).



- Electrons trapped on the floating-gate rise the threshold voltage of the n-channel EPROM transistor (Figure 5.13(b)).
- Once programmed, an n-channel EPROM device remains off even with V_{DD} applied to the top gate.
- An unprogrammed n-channel device will turn on as normal with a top-gate voltage of V_{DD}. The programming voltage is applied either from a special programming box or by using on-chip charge pumps.
- Exposure to an UV Lamp will erase the EPROM cell (Figure 5.13(c)).
- An absorbed light quantum gives an electron enough energy to jump from the floating-gate.
- Programming an EEPROM transistor is similar to programming an UV-erasable EPROM transistor, but the erase mechanism is different.
- In an EEPROM transistor an electric field is also used to remove electrons from the floating-gate of a programmed transistor.
- This is faster than using a UV lamp and the chip does not have to be removed from the system.

6. Explain Field-Programmable Gate Array (FPGA) building block architecture. [CO5-H1]

Field programmable gate arrays: (FPGA)

FPGA's are the newest member of the ASIC family and are rapidly growing in importance, replacing TTL in microelectronics systems. Even though an FPGA is a type of gate array, contains a regular structure of programmable basic logic cells surrounded by programmable interconnect. We do not consider the term gate array based ASICs to include FPGAs.

The important characteristics of an FPGA are

- 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.
- Programmable I/O cells surround the core.
- Design turnaround is a few hours.

FPGA BUILDING BLOCK ARCHITECTURE:

Field Programmable Gate Array (FPGA) is an IC which can be hardware-programmed to implement various logic functions. The end users of FPGA can program it to configure for any functionality -so it is called field programmable. FPGA is completely fabricated and standard parts are tested and available readily for use. FPGAs are popular with Microsystems designers because they fill the gap between TTL and PLD design and also expensive ASIGs. FPGAs are ideal for prototyping systems or for low-volume production. Normally FPGAs comprises of:

- ✓ Programmable logic blocks which implement logic functions.
- ✓ Programmable routing that connects these logic functions.
- ✓ I/O blocks that are connected to logic blocks through routing interconnect and that make off-chip connections

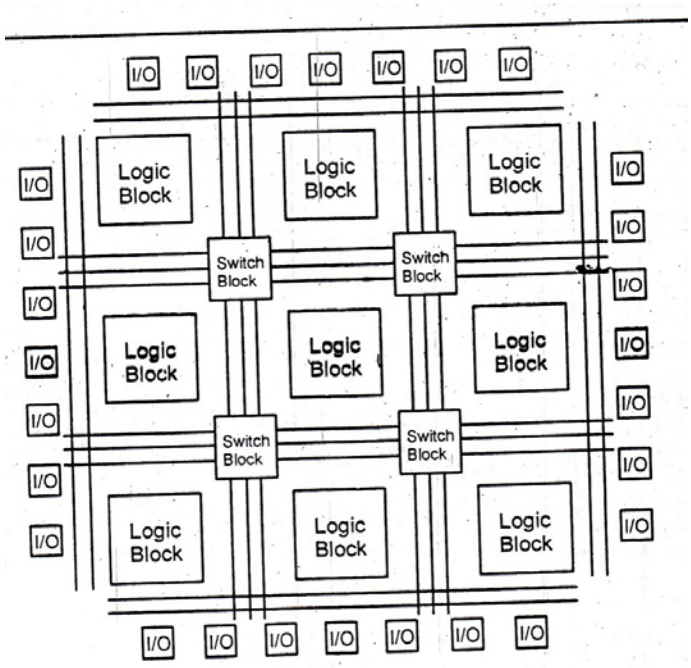


Figure 5.9 FPGA Architecture

FPGAs have a regular array of basic logic cells that are configured using a programming technology is shown in figure 5.9.The chip inputs and outputs use special I/O logic cells that are different from the basic logic cells. A programmable interconnects scheme forms the wiring between the two types of logic cells. Finally^ the designer uses custom software, tailored to each programming technology and FPGA architecture, to design and implement the programmable connections. The programming technology in an FPGA determines the type of basic logic cell and the interconnect scheme. The logic cells and interconnection scheme, in turn, determine the design of the input and output circuits as well as the programming scheme.

Configurable Logic Block (CLB)

A configurable logic block (CLB) is a basic component of an FPGA that provides the basic logic and storage functionality for a target application design is shown in figure 5.10. Exact numbers and features vary from device to device, but every CLB consists of a configurable switch matrix with 4 or 6 inputs, some selection circuitry (MUX, etc), and flip-flops is shown in figure 5.11 The switch matrix is highly flexible and can be configured to handle combinatorial logic, shift registers or RAM . The CLB acts as the main logic resource for implementing logic circuits. Generally the CLBs contain RAM ba'sed LUTs (look up tables) to implement logic and storage elements that can be used as flip-flops or latches .CLBs can be programmed to perform various logical functions as well as to store data.

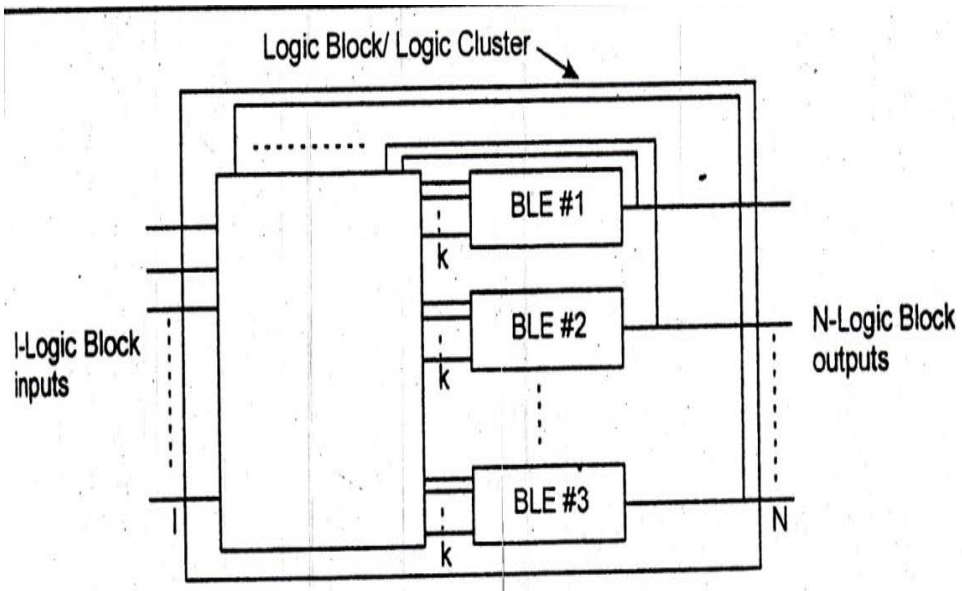


Figure 5.10 Block diagram of Logic Block

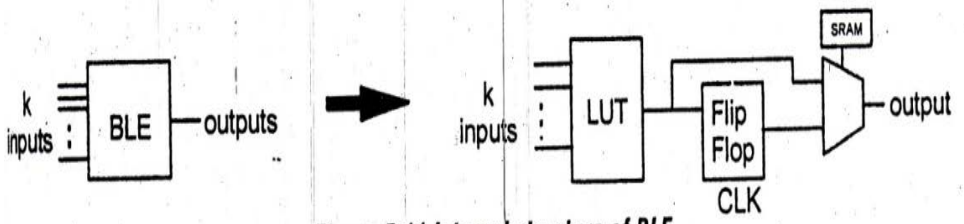


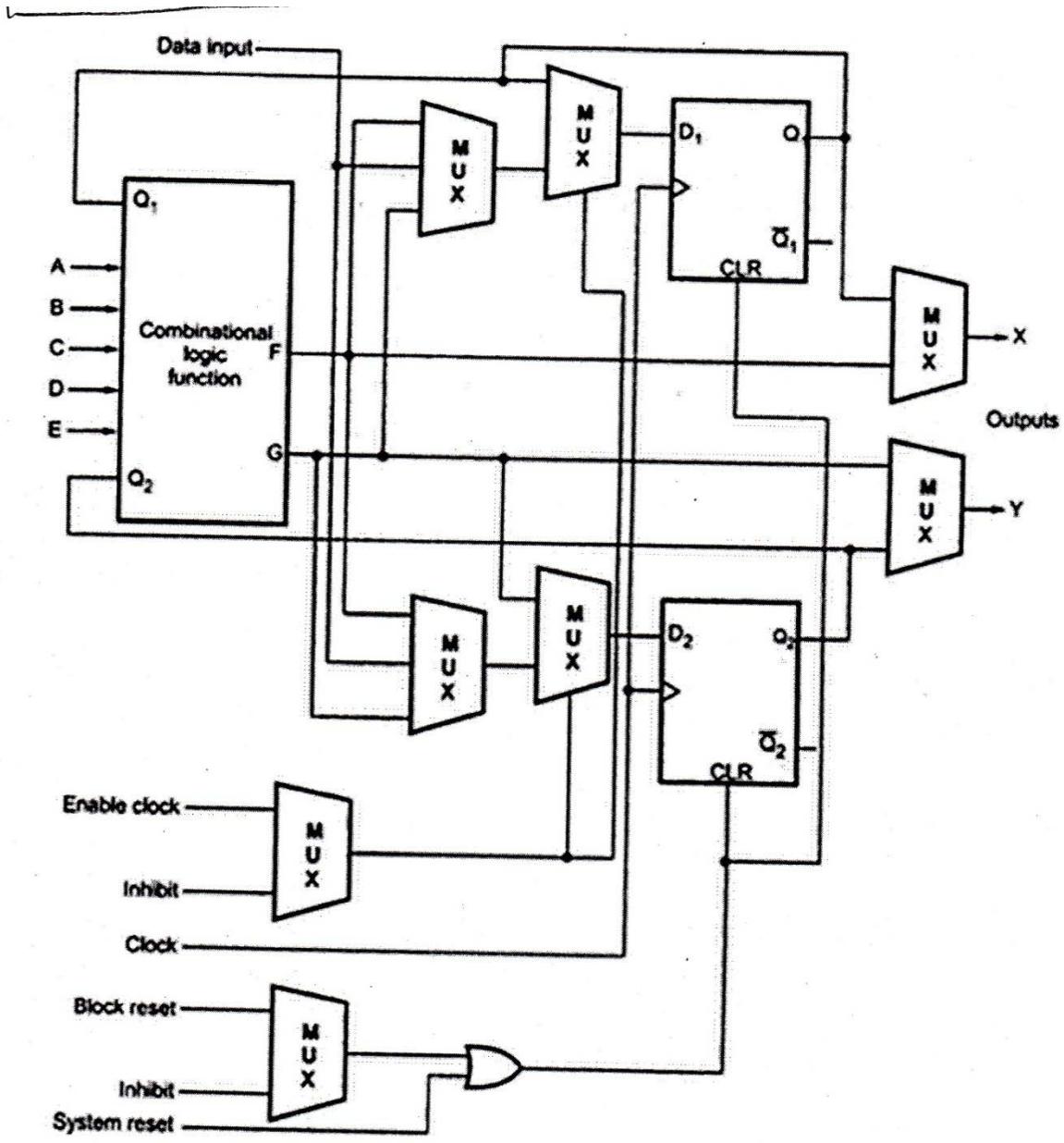
Figure 5.11 Internal structure of BLE

Xilinx, inc invertediFPGAs, and in this section we will see the FPGA architecture used by Xilinx the programmable logic blocks in the Xilinx family of FPGAs are called Configurable Logic Blocks (CLBs).The Xilinx architecture uses, CLBs, I/O blocks switch matrix and an external memory chip to realize a logic function. It uses external memory to store the interconnection information. Therefore, the device can be reprogrammed by simply changing the configuration data stored in the memory.

Major classifications:

(i) Xilinx 3000
(ii) Xilinx 4000

XILINX 3000 CLB

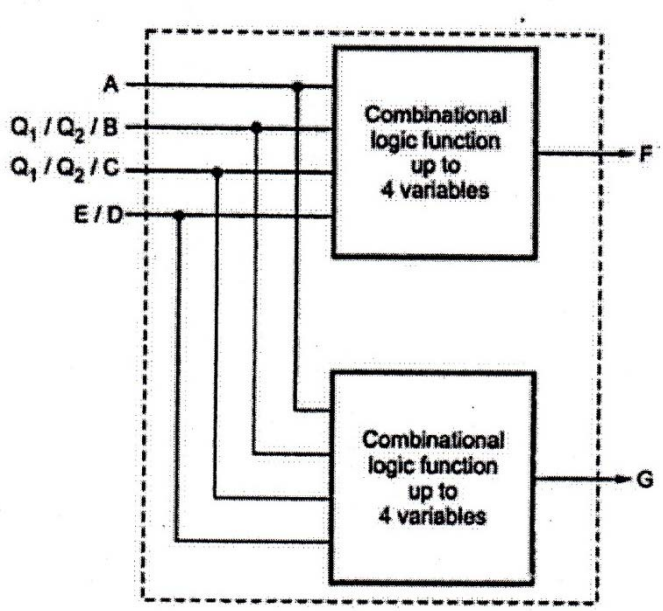


The diagram above shows the XILINX XC3000 CLB which has
(i) five logic inputs (A-E),
(ii) a common clock input (K),
(iii) an asynchronous direct-reset input (RD),
(iv) and an enable clock (EC).

Using programmable MUXes connected to the SRAM programming cells, one can independently connect each of the two CLB outputs (X and Y) to the output of the flip-flops (QX and QY) or to the output of the combinational logic (F and G).

There are seven inputs for the combinational logic in the XC3000 CLB among them five are CLB inputs from A to E and two are the flip-flop outputs.(OX and OY) There are two outputs from the LUT (F and G). Since a 32-bit LUT requires only five variables to form a unique address ($32 = 2^5$), there are several ways to use the LUT(One can use five of the seven possible inputs (A to E, QX, QY) with the entire 32-bit LUT. The CLB outputs (F and G) are then identical

The 32-bit LUT can be split into half, to implement two functions of four variables each. And can choose four input variables from the seven inputs(A to E, QX, QY). We have to choose two of the inputs from the five CLB inputs (A to E); then one function output connects to F and the other output connects to G.



Each CLB also has two outputs (X and Y) which may drive interconnect networks. Data input for the flip-flops within a CLB is supplied from the function F or G outputs of the combinatorial logic, or the block input, DI.

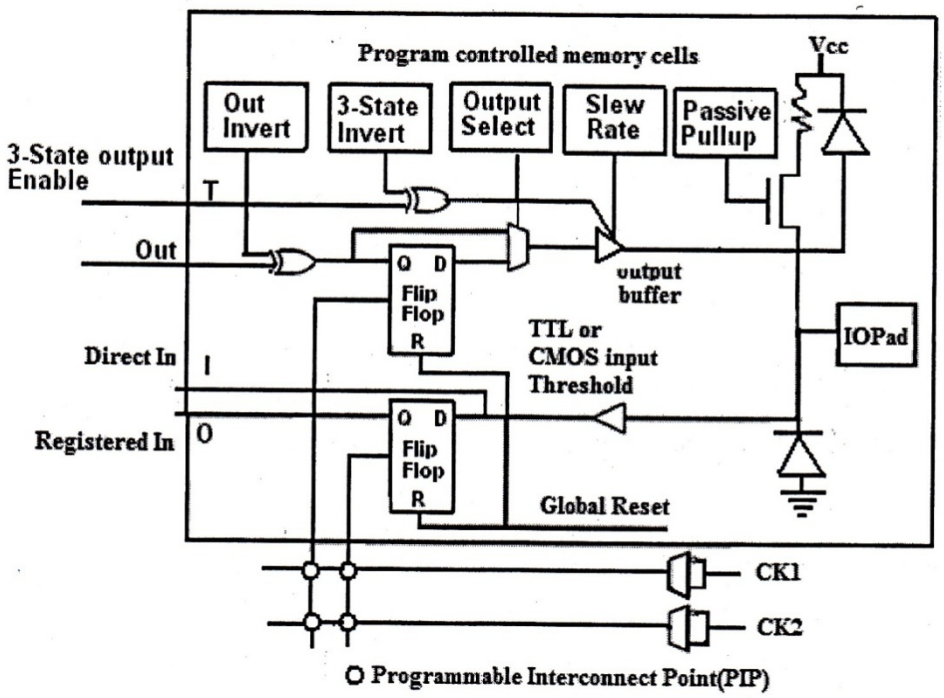
Both flip-flops in each CLB share the asynchronous RD which, when enabled , is dominant over clocked inputs. All flip-flops are reset by the active-Low chip input, RESET, or during the configuration process.

The flip-flops share the enable clock (EC) which, when Low, re circulates the flip-flops present states and inhibits response to the data-in or combinatorial function inputs on a CLB. the user may enable these control inputs and select their sources. The user may also select the clock net input (K), as well as its active sense within each CLB. This programmable inversion eliminates the need to route both phases of a clock signal throughout the device

Programmable I/O Block

Each user-configurable IOB as shown below, provides an interface between the external package pin of the device and the internal user logic. Each IOB

includes both registered and direct input paths. Each IOB provides a programmable3-state output buffer, which may be driven by a registered or direct output signal. Configuration options allow each IOB an inversion, a controlled slew rate and a high impedance pull-up. Each input circuit also provides input clamping diodes to provide electrostatic protection, and circuits to inhibit latch-up produced by input currents.



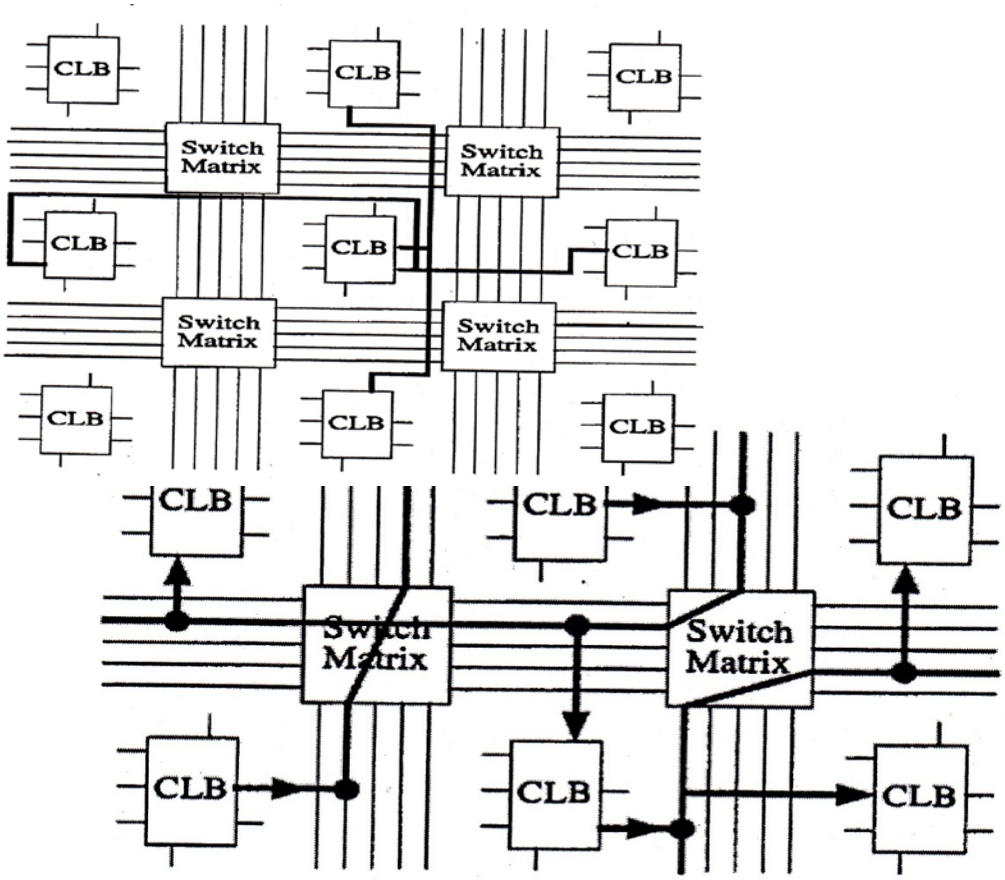
Each IOB includes input and output storage elements and I/O options selected by configuration memory cells. A choice of two clocks is available on each die edge. The polarity of each clock line (not each flip-flop or latch) is programmable. A clock line that triggers the flip-flop on the rising edge is an active Low Latch Enable (Latch transparent) signal and vice versa. Passive pull-up can only be enabled on inputs, not on outputs. All user inputs are programmed for TTL or CMOS thresholds

FPGA inter connect routing procedures:

Three types of metal resources are provided to fulfill various network interconnect requirements

- General Purpose Interconnect
- Direct Connection
- Long lines (multiplexed busses and wide AND gates)
- **General Purpose Interconnect**

It consists of a grid of five horizontal and five vertical metal segments located between the rows and columns of logic and IOBs. Each segment is the height or width of a logic block. -—Switching matrices join the ends of these segments and allow programmed interconnections between the metal grid segments of adjoining rows and columns.



The connections through the switch matrix may be established by the automatic routing or by selecting the desired pairs of matrix pins to be connected or disconnected.

Special buffers within the general interconnect areas provide periodic signal isolation and restoration for improved performance of lengthy nets. The interconnect buffers are available to propagate signals in either direction on a given general interconnect segment. These bidirectional (bidi) buffers are found adjacent to the switching matrices, above and to the right.

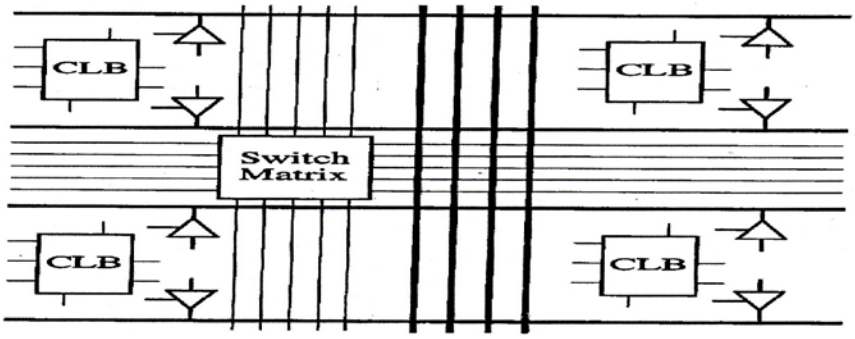
The other PIPs adjacent to the matrices are accessed to or from Long lines. The development system automatically defines the buffer direction based on the location of the interconnection network source.

Direct Interconnect

Direct interconnect provides the most efficient implementation of networks between adjacent CLBs or I/O Blocks. Signals routed from block to block using the direct interconnect exhibit minimum interconnect propagation and use no general interconnect resources. For each CLB, the X output may be connected directly to the B input of the CLB immediately to its right and to the C input of the CLB to its left. The Y output can use direct interconnect to drive the D input of the block immediately above and the A input of the block below. Direct interconnect should be used to maximize the speed of high-performance portions of logic. Where logic blocks are adjacent to IOBs, direct connect is provided alternately to the IOB inputs (I) and outputs (O) on all four

edges of the die. The right edge provides additional direct connects from CLB outputs to adjacent IOBs.

The Long lines bypass the switch matrices and are intended primarily for signals that must travel a long distance, or must have minimum skew among multiple destinations. Long lines, run vertically and horizontally the height or width of the interconnect area. Each interconnection column has three vertical Long lines, and each interconnection row has two horizontal Long lines. Two additional Long lines are located adjacent to the outer sets of switching matrices. Long lines can be driven by a logic block or IOB output on a column-by-column basis. This capability provides a common



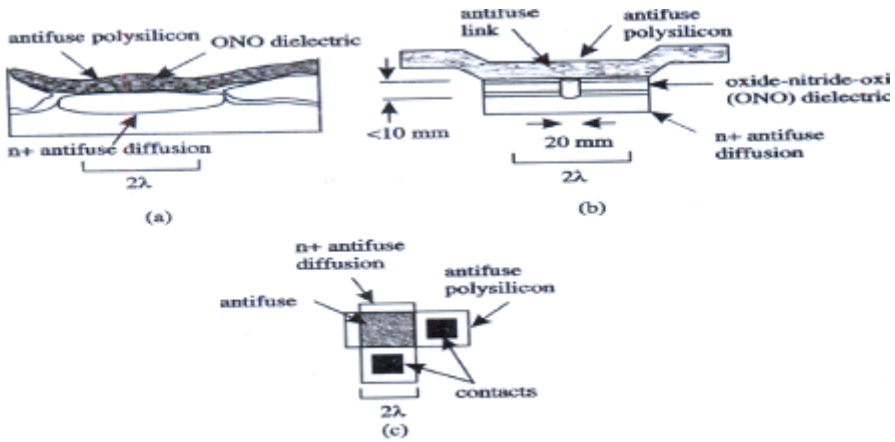
low skew control or clock line within each column of logic blocks. Isolation buffers are provided at each input to a Long line and are enabled automatically by the development system when a connection is made.

7. Explain various methods of FPGA interconnecting routing procedures.[CO5- H1]

- All FPGAs contain some type of programmable interconnect. The structure and complexity of the interconnect is largely determined by the programming technology and the architecture of the basic logic cell.
- The raw material that is used in building the interconnect is aluminium-based metallization, which has a sheet resistance of approximately 50 mΩ /square and a line capacitance of 0.2 pf cm^{-1}.
- Commercially, programmable routing approaches are represented by products from Actel, Quick logic, and other companies.

ACTEL Antifuse Interconnect

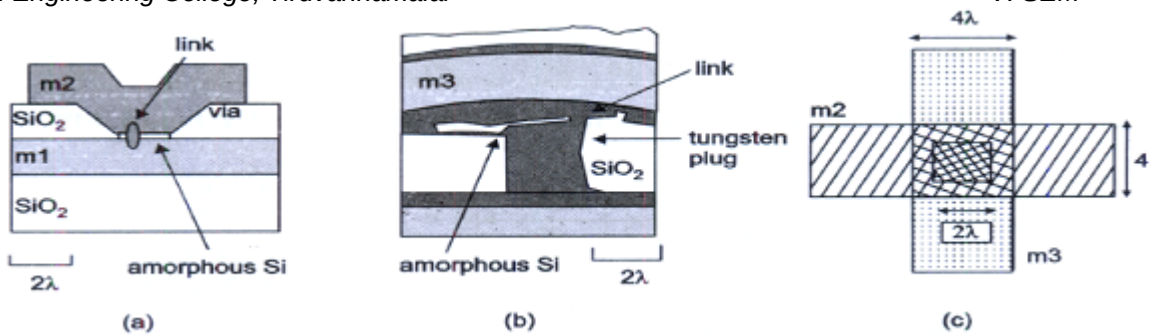
- The Actel Field programmable Gate Arrays are based on an element called a PLICE (Programmable Low-Impedance Circuit Element) or antifuse.
- An antifuse is the opposite of a regular fuse. An antifuse is normally an open circuit until you force a programming current through it (about 5 mA).
- In a poly-diffusion antifuse the high current density causes a large power dissipation in a small area, which melts a thin insulating dielectric between polysilicon and diffusion electrodes and forms a thin (about 20 nm in diameter), permanent, and resistive silicon link.



- Figure 5.14 shows a poly-diffusion antifuse with an oxide-nitride-oxide (ONO) dielectric sandwich of: Silicon dioxide ($Si0_2$) grown over the n-type antifuse diffusion, a silicon nitride(Si_3N_4) layer and another thin $Si0_2$ layer.
- The layered ONO dielectric results in a tighter spread of blown antifuse resistance values than using a single-oxide dielectric.
- The fabrication process and the programming current control the average resistance of a blown antifuse.
- In a particular technology a programming current of 5 mA may result in an average blown antifuse resistance of about 500Ω.
- Increasing the programming current might reduce the average antifuse resistance. Antifuses separate interconnect wires on the FPGA chip and the programmer blows an antifuseto make a permanent connection. Once an antifuse is programmed, the process cannot be reversed.
- To design and program an Actel FPGA, designers iterate between design entry and simulation.
- When they are satisfied that design is correct they plug the chip into a socket on a special programming box, called an Activator, that generates the programming voltage.
- A PC download the configuration file to the Activator instructing it to blow the necessary antifuses on the chip.

Metal-Metal Antifuse

- Figure 5.15 shows a Quick Logic metal-metal antifuse (vialink). The link is an alloy of Tungsten, titanium, and silicon with a bulk resistance of about 50μΩ cm.



- There are two advantages of a metal-metal antifuse over a poly-diffusion antifuse. The first is that connections to a metal-metal antifuse are direct to metal-the wiring layers.
- Connections from a poly-diffusion antifuse to the wearing layers require extra space and create additional parasitic capacitance.
- The second advantage is that the direct connection to the low-resistance metal layers makes it earlier to use larger programming currents to reduce the antifuse resistance.

Actel Act interconnect Scheme

The Actel Act family interconnect scheme shown in Figure 5.16 is similar to a channeled gate array. The channel routing uses dedicated rectangular areas of fixed size within the chip called wiring channels. The horizontal channels run across the chip in the horizontal direction

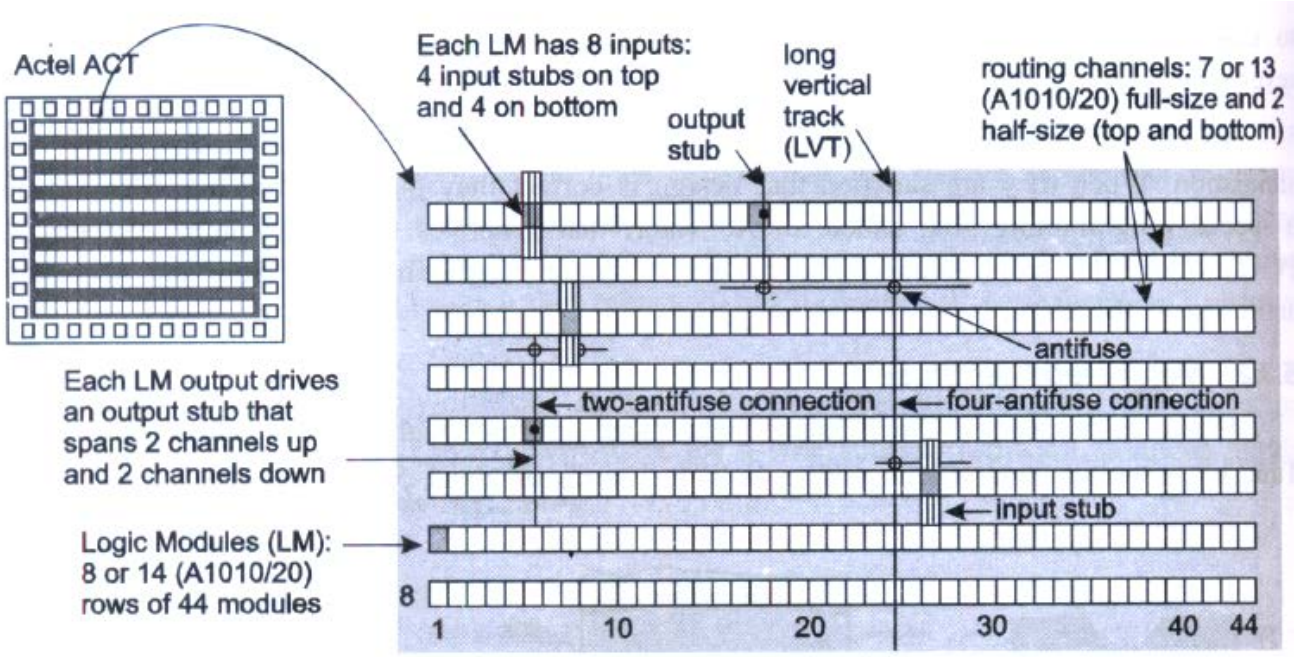


Figure 5.16 The Interconnect Architecture used in an Actel ACT family FPGA

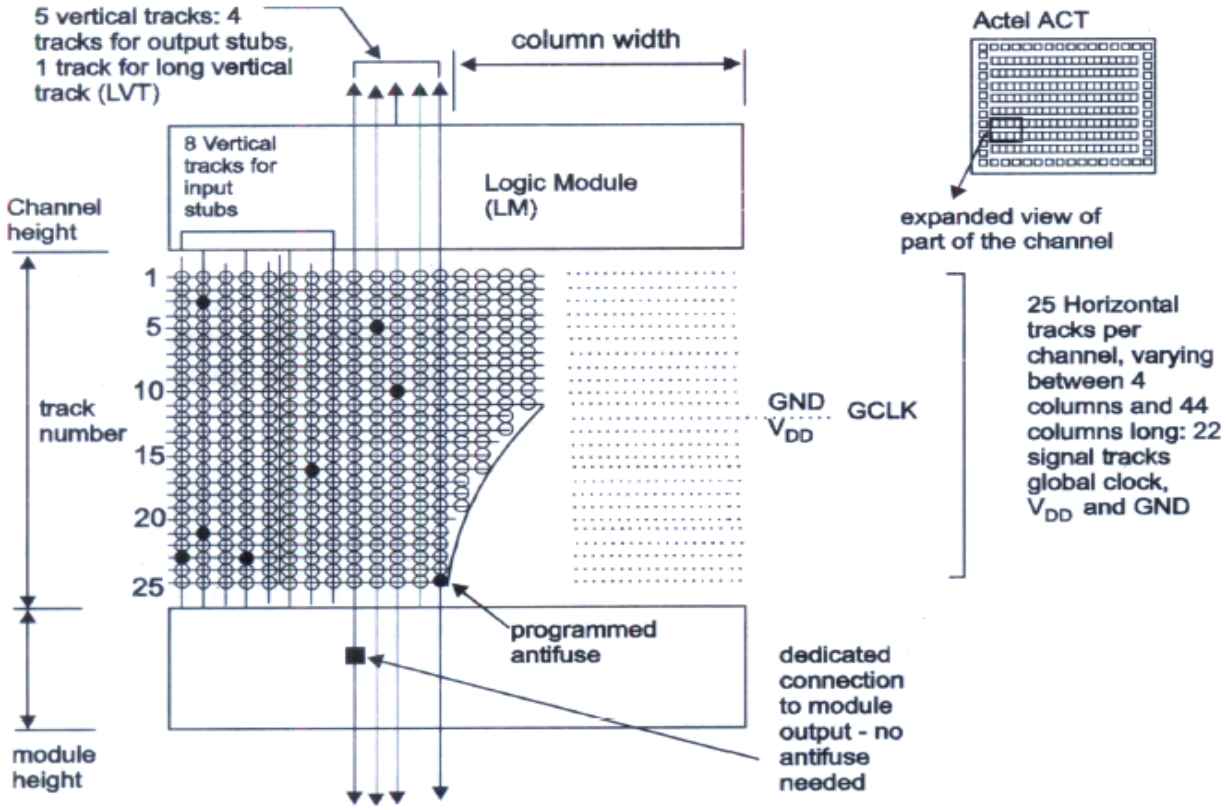


Figure 5.17 Act 1 horizontal and vertical channel architecture (Source: Actel)

- In the vertical direction there are similar vertical channels that run over the top of the basic logic cells, the logic modules.
- Within the horizontal or vertical channels wires run horizontally or vertically, respectively, within tracks. Each track holds one wire. The capacity of a fixed wiring channel is equal to the number of tracks it contains.
- Figure 5.17 shows a detailed view of channel and the connections to each logic module-the input and output stubs.
- In an FPGA the interconnect is fixed at the time of manufacture.
- To allow programming of the interconnect, Actel divides the fixed interconnect wires within each channel into various lengths or wire segments. This is called segmented channel routing, a variation on channel routing.
- Antifuses join the wire segments. The designer then programs the interconnections by blowing antifuses and making connections between wire segments; unwanted connections are left unprogrammed.

The structure of the Actel logic element is shown in Figure 5.18. It consists of three 2-input muxes and a NOR gate. This logic structure implements 2-and 3- input logic functions

- A latch can be implemented using single module whereas a register requires 2 logic module.

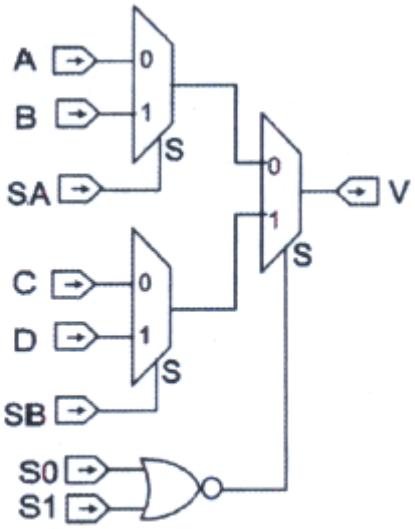


Figure 5.18 Actel Logic Cell

- The Actel programmable Input pad is shown in Figure 5.19. Two antifuses are used to configured the I/O pads.
- If the ENABLE pin is not programmed, then the pad is allowed to operate in bidirectional. If the ENABLE antifuse is blown to V_{DD}, the pad act as an output, whereas if the V_{SS}antifuse is blown, the pad is an input.
- The isolation devices isolate the pad if necessary during programming and testing.

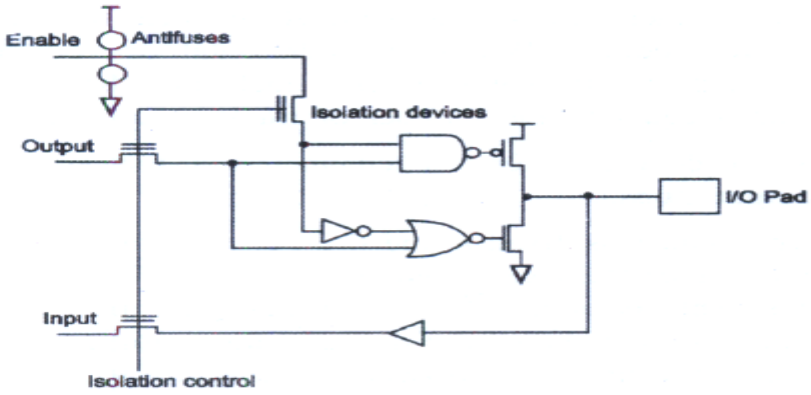


Figure 5.19 Actel I/O Pad

8. Explain the function of ASIC design flow with block diagram. [CO5-H1]

- A design methodology is frequently called a design flow since the flow of data through the steps in the methodology may be represented in a block diagram.
- Figure (5.34) shows the sequence of steps to design an ASIC. The steps are listed below with a brief description of the function of each step.

1. Design entry: Design are entired in a hardware description language such as VHDL or Verilog or through Schematic entry.

2. Logic Synthesis:

- Logic Synthesis provides a link between an HDL (verilog or VHDL) and a netlist.

- Logic synthesis tools are used to map the design into a gate level design in the cell library. Logic synthesis ensures that the design is appropriate for Level Sensitive Scan Design (LSSD)
- Design entry through HDL behavioral model does not contain any references to logic cells.
- Once a behavioral HDL model is complete, two items are required to proceed: a logic synthesizer (software and documentation) and a cell library (the logic cells - NAND gates, NOR gates and such) that is called the target library.
- The behavioral model is simulated to check that the design meets the specifications and then logic synthesizer is used to generate a netlist, a structural model, which contains only references to logic cells. There is no standard format for the netlists that logic synthesis produces, but EDIF is widely used.

3. System portioning: Divide a large system into ASIC sized pieces.

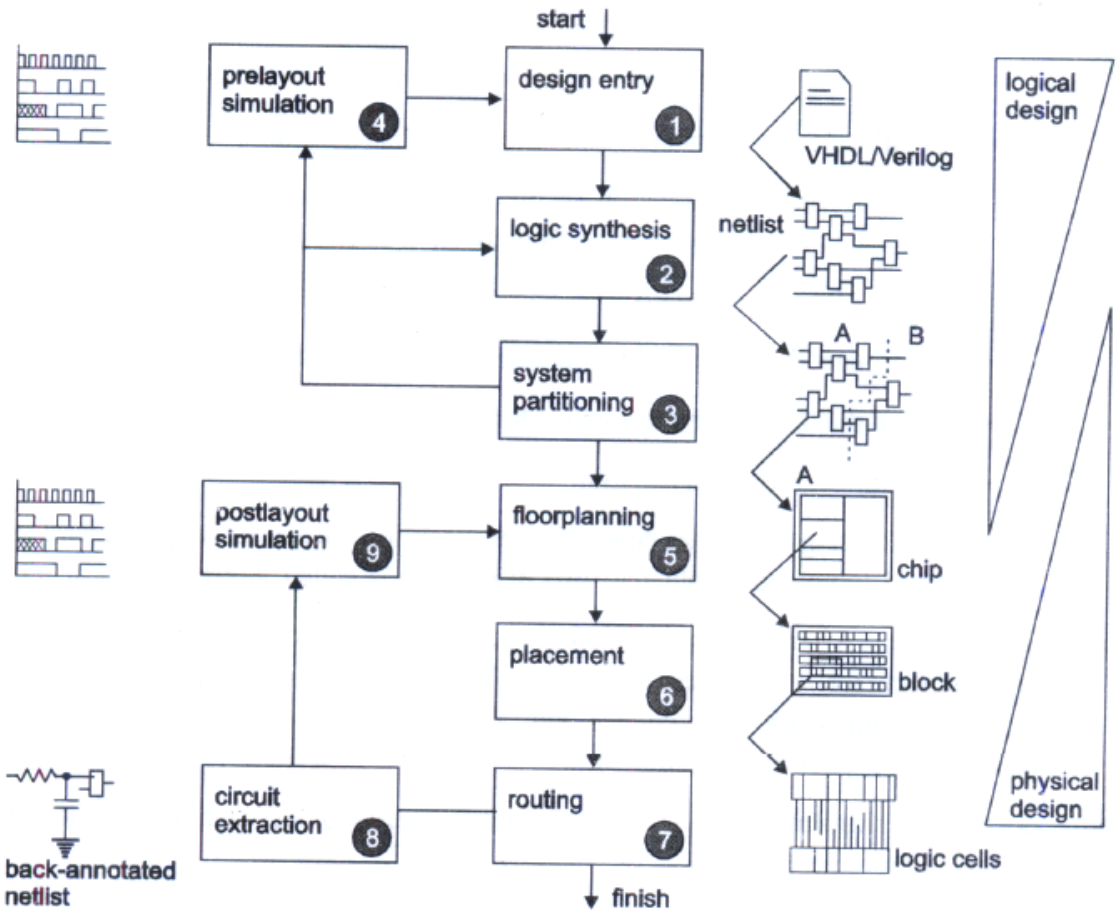


Figure 5.34 ASIC Design Flow

The main Goal to:

- Portioning a large system into smaller pieces.
- minimize the number of pins.
- minimize package cost

- minimize the number of external connections between the ASICs. -» keep each ASIC smaller than a maximum size.

4. Prelayout simulation: Verifying the functionality of the design. Initial prelayout simulations include logic-cell delays but no interconnect delays.

5. Floor planning:

- Arrange the blocks of the netlist on the Chip. The main objectives of floor-planning are to minimize the chip area and minimize delay.
- In floor-planning the designer estimate sizes and set the initial relative locations of the various blocks in the ASIC-design.
- Floor-planning allows the designer to predict the interconnect delay by estimating interconnect length.

The goals of floor-planning are to

- Arrange the blocks on a chip.
- Decide the location of the I/O pads.
- decide the location and number of power pads,
- decide the type of power distribution and
- Decide the location and type of clock distribution.

6. **Placement:** Defines the location of the logic cells within the flexible blocks and sets aside space for the interconnection to each logic cell.

The goals of placement are to

- Minimize the total estimated interconnect length; meet the timing requirements for critical nets.
- Minimize power dissipation.

7. Routing: Routing makes the connection between logic cells. Routing is a hard problem by itself and is normally split into two distinct steps called global and local routing.

8. Extraction: After routing is complete, the exact length and position of each interconnect for every net is known. The parasitic capacitance and resistance associated with each interconnect, via, and contact can be calculated. The data is generated by circuit-extraction tool.

9. Post layout simulation: Verifying the functionality of the design after added loads of the interconnect.