BEARYS INSTITUTE OF TECHNOLOGY

Approved by AICTE, New Delhi, Recognised by Govt. of Karnataka
Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka
(Accredited by NAAC)
Lands End, Innoli, Near Mangalore University, Mangaluru, Karnataka - 574199

COURSE FILE DETAILS

COURSE NAME : **VLSI Design and Testing**

COURSE CODE : **BEC602**

NBA CODE : **C302**

SEMESTER & SECTION : **6TH Semester & A-Section**

ACADEMIC YEAR : **2025-2026**

COURSE COORDINATOR : **Dr. ABDULLAH GUBBI**

INSTITUTE - VISION

To be a premier institution in engineering education and research, fostering innovation, societal responsibility, and ethical leadership for a sustainable future.

INSTITUTE - MISSION

1. Promote innovation and cutting-edge research to develop sustainable engineering solutions for real-world and global challenges.
2. Cultivate ethical leadership and technical excellence among students to become responsible and leading engineering professionals.
3. Encourage active societal engagement and industry collaboration to drive inclusive growth and environmental responsibility.

DEPARTMENT ECE- VISION

To be a center of excellence in Electronics and Communication Engineering, advancing research, innovation, ethical practice, and sustainable solutions for societal transformation

DEPARTMENT ECE - MISSION

1: Foster innovation and research in Electronics and Communication Engineering, to create sustainable solutions addressing real-world and technological challenges.
2: Inculcate technical excellence and ethical values to develop competent graduates who contribute as professionals and leaders.
3: Promote industry collaboration and community engagement to ensure inclusive development and environmentally conscious engineering practices.

Program Educational Objectives

1. Graduates will apply strong foundations in Electronics, Communication, and Computing to design innovative and reliable engineering solutions for real-world challenges.

2. Graduates will demonstrate professional integrity, ethical responsibility, and a commitment to continuous learning to adapt to technological advancements and societal needs.

3. Graduates will contribute effectively as Engineers, Leaders, or Entrepreneurs by collaborating with industry and engaging in projects that promote inclusive growth.

VLSI Design and Testing		Semester	5
Course Code	**BEC602**	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	4:0:0:0	SEE Marks	50
Total Hours of Pedagogy	50 Hours	Total Marks	100
Credits	04	Exam Hours	3 Hours
Examination nature (SEE)	Theory		

Course objectives:

1.This course deals with analysis and design of digital CMOS integrated circuits.

2. The course emphasizes on basic theory of digital circuits, design principles and techniques for digital design blocks implemented in CMOS technology.

3. This course will also cover switching characteristics of digital circuits along with delay and power estimation.
4. Understanding the CMOS sequential circuits and memory design concepts.

5.Explore the knowledge of VLSI Design flow and Testing

Teaching-Learning Process (General Instructions)

These are sample Strategies; that teachers can use to accelerate the attainment of the various course outcomes.

1. Lecture method (L) does not mean only traditional lecture method, but different type of teaching methods may be adopted to develop the outcomes.
2. Show Video/animation films to explain the different concepts of Digital Signal Processing
3. Encourage collaborative (Group) Learning in the class
4. Ask at least three HOTS (Higher order Thinking) questions in the class, which promotes critical thinking
5. Adopt Problem Based Learning (PBL), which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
6. Topics will be introduced in a multiple representation.
7. Show the different ways to solve the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every concept can be applied to the real world - and when that's possible, it helps improve the students' understanding.
9. Adopt Flipped class technique by sharing the materials / Sample Videos prior to the class and have discussions on the that topic in the succeeding classes.

MODULE-1

Introduction to CMOS Circuits: Introduction, MOS Transistors, MOS Transistor switches, CMOS Logic, Alternate Circuit representation, CMOS-nMOS comparison.

[Text 1: 1.1,1.2,1.3,1.4,1.5.1.6.]

Teaching-Learning Process: Chalk and talk method, YouTube videos, Power point presentation RBT Level: L1, L2

MODULE-2

MOS Transistor Theory: n-MOS enhancement transistor, p-MOS transistor, Threshold Voltage, Threshold voltage adjustment, Body effect, MOS device design equations, V-I characteristics, CMOS inverter DC characteristics, Influence of βn / βp ratio on transfer characteristics, Noise margin, Alternate CMOS inverters. Transmission gate DC characteristics. Latch-up in CMOS.

[Text 1: 2.1,2.2,2.3,2.4,2.5.2.6.]

Teaching-Learning Process:
Chalk and talk method/Power point presentation RBT Level: L1, L2, L3.

MODULE-3

CMOS Process Technology: Silicon Semiconductor Technology, CMOS Technologies, Layout Design Rules. [Text 1: 3.1,3.2,3.3.]

Circuit Characterization and Performance Estimation: Introduction, Resistance Estimation, Capacitance Estimation, Switching Characteristics, CMOS gate transistor sizing, Determination of conductor size, Power consumption, Charge sharing, Scaling of MOS transistor sizing, Yield.
[Text 1: 4.1,4.2,4.3,4.4,4.5.4.6.4.7,4.8,4.9,4.10]

Teaching-Learning Process:
Chalk and talk method/Power point presentation, YouTube Videos RBT Level: L1, L2, L3.

MODULE-4

CMOS Circuit and Logic Design: Introduction, CMOS Logic structures, CMOS Complementary logic, Pseudo n-MOS logic, Dynamic CMOS logic, Clocked CMOS Logic, Cascade Voltage Switch logic, Pass transistor Logic, Electrical and Physical design of Logic gates, The inverter, NAND and NOR gates, Body effect, Physical Layout of Logic gates, Input output Pads.

[Text 1: 5.1,5.2,5.2.1, , 5.2.2, 5.2.3, 5.2.4, 5.2.6, 5.2.8, 5.3,5.3.1,5.3.2, 5.3.4 ,5.3.8,5.5]

Teaching-Learning Process:
Chalk and talk method, YouTube videos, Power point presentation RBT Level: L1, L2, L3.

MODULE-5

Sequential MOS Logic Circuits: Introduction, Behaviour of Bistable Elements (Excluding Mathematical analysis) SR Latch Circuit, Clocked Latch and Flip-Flop Circuits, Clocked SR Latch, Clocked JK Latch.
[Text2: 8.1, 8.2, 8.3, 8.4]

Structured Design and Testing: Introduction**,** Design Styles, Testing
[Text1: 6.1, 6.2. 6.5]

Teaching-Learning Process:
Chalk and talk method/Power point presentation RBT Level: L1, L2, L3

Text Books:

1. Principals of CMOS VLSI Design A System approach Neil H E Weste and Kamran Eshraghain . Addition Wisley Publishing company.
2. "CMOS Digital Integrated Circuits: Analysis and Design", Sung Mo Kang & Yosuf Leblebici, Third Edition, Tata McGraw-Hill.

Reference Books:

1. "CMOS VLSI Design- A Circuits and Systems Perspective", Neil H E Weste, and David Money Harris 4th Edition, Pearson Education.
2. "Basic VLSI Design", Douglas A Pucknell, Kamran Eshraghian, 3rd Edition, Prentice Hall of India publication, 2005.

Course Outcomes: After completing the course, the students will be able to

CO1	Apply the fundamentals of semiconductor physics in MOS transistors and analyze the geometrical effects of MOS transistors
CO2	Design and realize combinational, sequential digital circuits and memory cells in CMOS logic.
CO3	Analyze the synchronous timing metrics for sequential designs and structured design basics.
CO4	Understand designing digital blocks with design constraints such as propagation delay and dynamic power dissipation.
C05	Understand the concepts of Sequential circuits design and VLSI testing

BEARYS INSTITUTE OF TECHNOLOGY, MANGALORE

Department of Electronics and Communication Engg.

VLSI Design and Testing(BEC602)

Introduction to CMOS Circuits

1 Introduction to CMOS Technology

- Complementary Metal-Oxide-Semiconductor (CMOS) technology has become a fundamental part of the modern integrated circuit (IC) industry.
- Although CMOS is widely used today, its origins date back nearly a century.

1.1 Historical Background

- The concept of the MOS field-effect transistor (MOSFET) was first proposed by J. Lilienfeld in 1925, with a similar structure later suggested by O. Heil in 1935.
- Early attempts to develop MOS transistors faced material-related challenges, leading to the invention of the bipolar junction transistor (BJT), which became the dominant technology for many years.
- The MOS transistor gained renewed interest with the advent of the silicon planar process in the early 1960s.
- However, quality control and material challenges delayed its commercial adoption until around 1967.
- Initially, only single-polarity MOS transistors (p-type or n-type) were commonly used.
- CMOS, which utilizes both p-type and n-type transistors on the same substrate, was first applied to ultra-low-power applications such as digital watches.
- Due to the complexity of CMOS fabrication, it was initially less favored in general system designs. However, as nMOS processing technology became more intricate, the relative complexity of CMOS became less of a concern.

1.2 The Growing Importance of CMOS

- The demand for low-power, high-density ICs led to a surge in CMOS adoption.
- System designers faced increasing challenges with chip size and power consumption, making CMOS an attractive alternative.
- Today, CMOS has become the dominant technology for Very Large-Scale Integration (VLSI) circuit design.

2 MOS Transistors

- A **MOS (Metal-Oxide-Silicon) transistor** is a fundamental semiconductor device used in modern integrated circuits.
- It is formed by layering conducting, insulating, and transistor-forming materials on a silicon substrate.
- The key structural elements include diffusion regions, polysilicon layers, and metal interconnections, all separated by insulating layers.

2.1 CMOS Technology and Transistor Types

CMOS (Complementary MOS) technology utilizes two types of MOS transistors:

- **nMOS (n-type MOSFET)** – An n-type transistor is fabricated on a p-type silicon substrate, with two n-type diffused regions acting as the source and drain.
- **pMOS (p-type MOSFET)** – A p-type transistor is fabricated on an n-type silicon substrate, with two p-type diffused regions serving as the source and drain.

The doping of the silicon substrate determines the type of charge carriers:

- **nMOS transistors** use electrons as the majority carriers (negatively charged).
- **pMOS transistors** use holes as the majority carriers (positively charged).

2.2 Physical Structure and Components

A typical MOS transistor consists of the following components:

- **Source (S):** One of the two terminals where current enters or exits.
- **Drain (D):** The other terminal where current enters or exits.
- **Gate (G):** A conducting electrode (typically polysilicon) placed over a thin insulating layer (oxide) that controls current flow.
- **Substrate (Body):** The silicon region in which the device is fabricated, either p-type for nMOS or n-type for pMOS.

For an **nMOS transistor**, the structure consists of a **p-type substrate** separating two **n-type diffusion regions**. A **pMOS transistor** has an **n-type substrate** with two **p-type diffusion regions**. In both cases, the **gate electrode** sits above the channel region, separated by an insulating oxide layer.

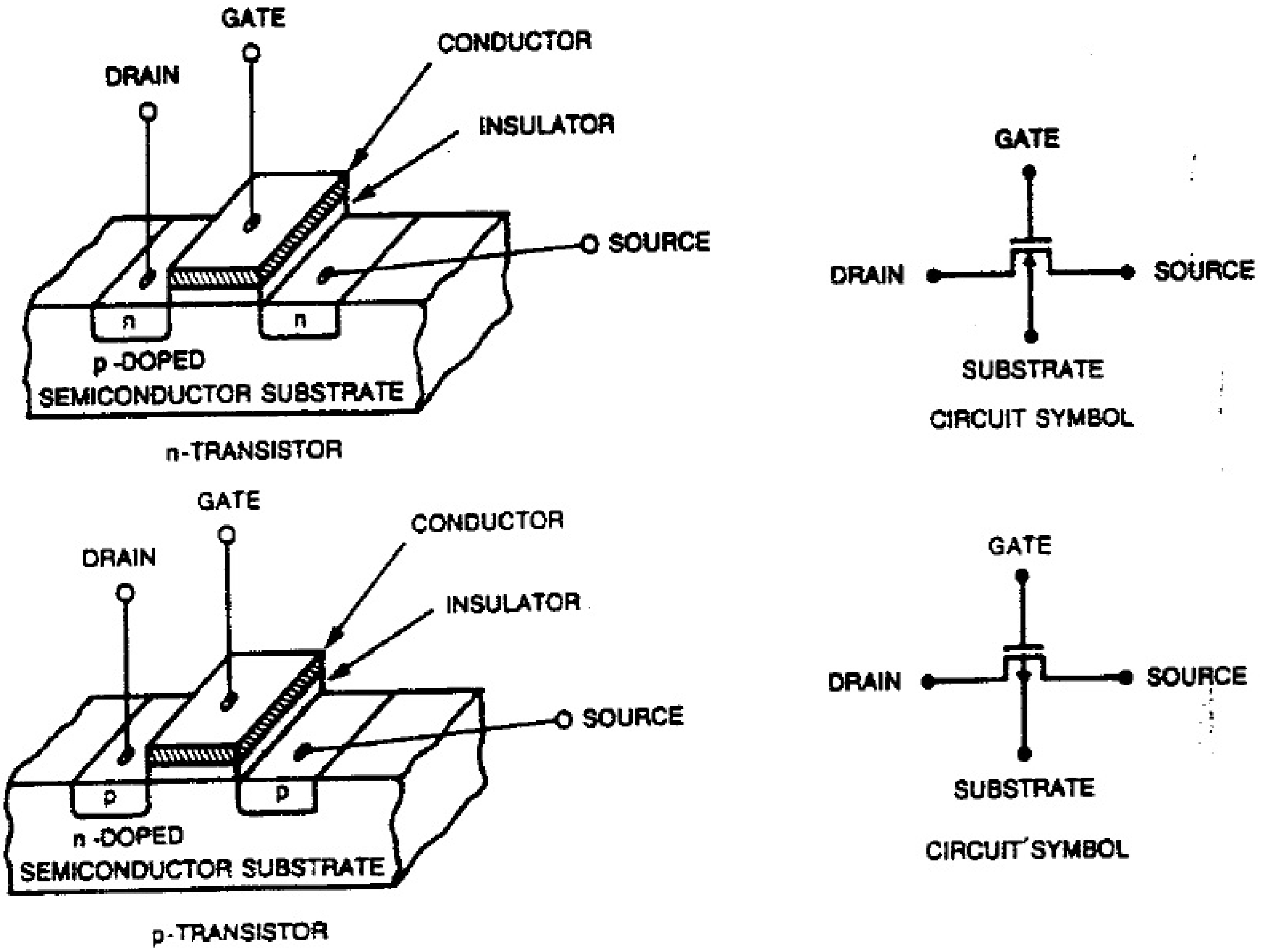


Figure 1: MOS transistor physical structures

2.3 Operation of MOS Transistors

The MOS transistor operates by controlling the flow of current between the **source** and **drain** terminals using the **gate** voltage.

- Applying a **positive voltage** to the gate of an **nMOS** transistor attracts electrons, creating a conductive channel between the source and drain.
- Applying a **negative voltage** to the gate of a **pMOS** transistor attracts holes, enabling conduction between the source and drain.

The source and drain terminals are interchangeable, and their designation depends on the direction of current flow. The gate acts as a **control terminal**, regulating the electrical connection between the drain and source.

2.4 Conclusion

MOS transistors are the building blocks of digital and analog circuits. The combination of **nMOS and pMOS transistors in CMOS technology** enables low-power, high-speed circuits used in microprocessors, memory chips, and various electronic devices.

3 MOS Transistor Switches

MOS (Metal-Oxide-Semiconductor) transistors act as controlled switches in digital circuits. The gate terminal determines whether the switch is ON or OFF. We assume:

- A high voltage ('1') is normally set to $5V$ and is called **POWER** (V_{DD}).
- A low voltage ('0') is normally set to $0V$ and is called **GROUND** (V_{SS}).
- The **strength** of a signal is measured by its ability to source or sink current.

3.1 nMOS Transistor as a Switch

An nMOS transistor operates as follows:

- **Gate = '1'** (V_{DD}): The switch is **ON** (closed), allowing current flow.
- **Gate = '0'** (V_{SS}): The switch is **OFF** (open), blocking current flow.

Passing Signals

- **Good for passing '0'** (LOW signal).
- **Imperfect for passing '1'** (HIGH signal is degraded due to threshold voltage drop $V_{DD} - V_{th}$).

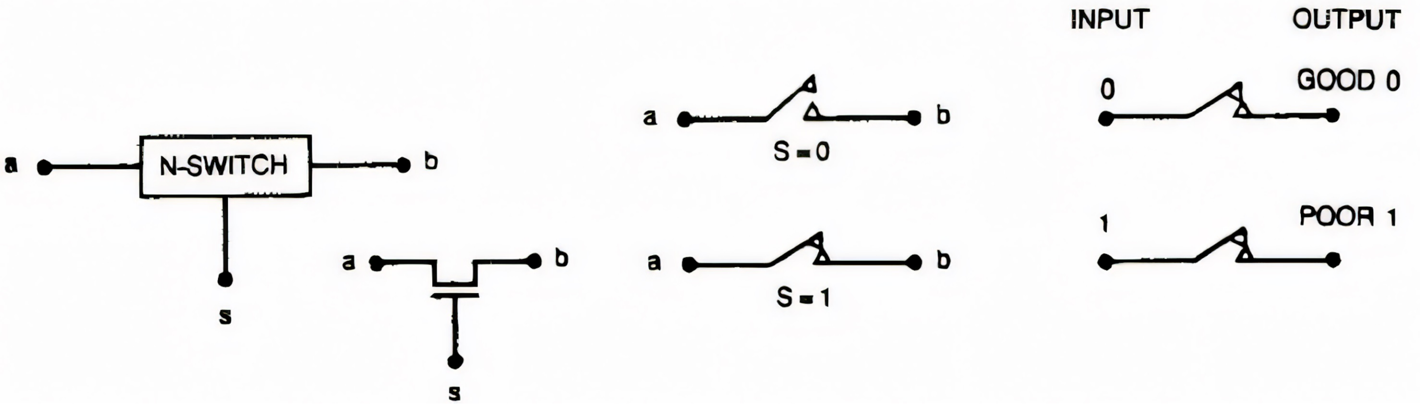


Figure 2: nMOS transistor as a switch

3.2 pMOS Transistor as a Switch

A pMOS transistor operates as follows:

- **Gate = '0'** (V_{SS}): The switch is **ON** (closed), allowing current flow.
- **Gate = '1'** (V_{DD}): The switch is **OFF** (open), blocking current flow.

Passing Signals

- **Good for passing '1'** (HIGH signal).
- **Imperfect for passing '0'** (LOW signal is degraded).

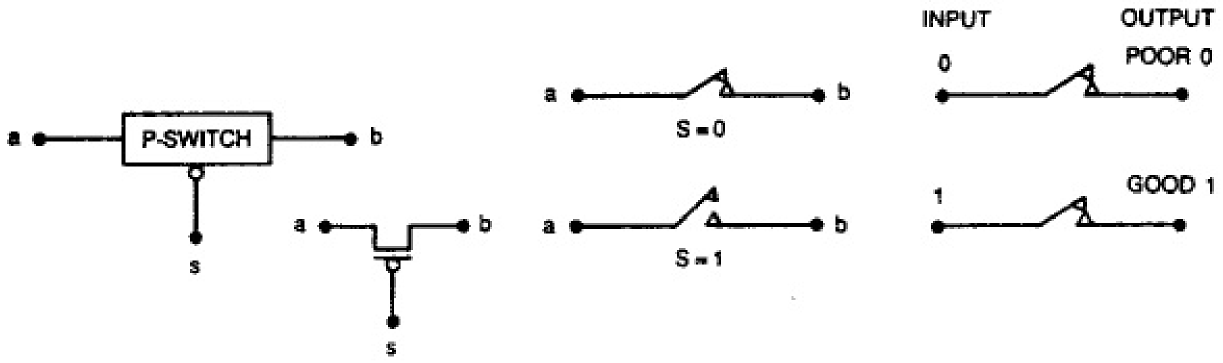


Figure 3: pMOS transistor as a switch

3.3 Transmission Gate (Complementary Switch)

A transmission gate consists of an nMOS and a pMOS transistor connected in parallel. It is controlled by:

- **Control signal (C) applied to nMOS**.
- **Complementary control signal ($\overline{C}$) applied to pMOS**.

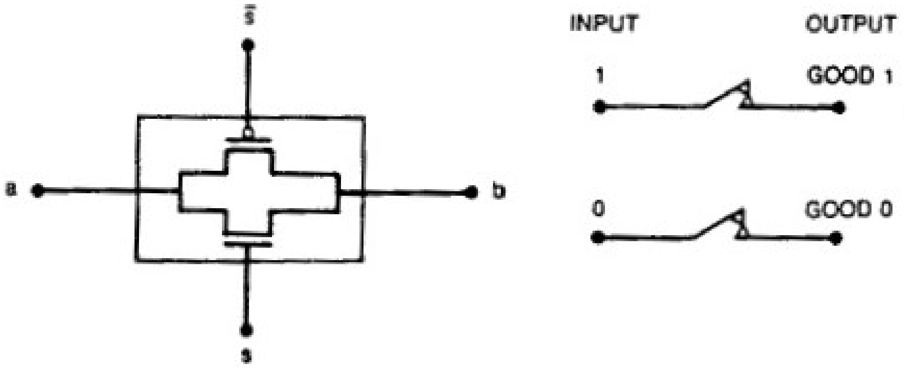


Figure 4: Transmission Gate (Complementary Switch)

Advantages of Transmission Gates

- Overcomes the limitations of individual nMOS and pMOS switches.
- Provides **low resistance for both '0' and '1' signals**.
- Used in **multiplexers, flip-flops, and pass-transistor logic circuits**.

3.4 Summary of Switch Behavior

Table 1: Summary of Switch Behavior

Switch Type	**Gate Signal**	**State**	**Passes '0' Well**	**Passes '1' Well**
nMOS	1 (V_{DD})	ON	Yes	No (degraded '1')
	0 (V_{SS})	OFF	No	No
pMOS	0 (V_{SS})	ON	No (degraded '0')	Yes
	1 (V_{DD})	OFF	No	No
Transmission Gate	$C = 1$, $\overline{C} = 0$	ON	Yes	Yes

3.5 Conclusion

MOS transistors serve as voltage-controlled switches in digital circuits:

- **nMOS transistors** efficiently pass **'0'** signals but degrade **'1'** signals.
- **pMOS transistors** efficiently pass **'1'** signals but degrade **'0'** signals.
- **Transmission gates (TG)** provide optimal switching by combining nMOS and pMOS transistors.

These principles are widely applied in **CMOS logic circuits, multiplexers, and memory circuits**.

4 CMOS Logic

4.1 The Inverter

- A CMOS inverter is a fundamental building block of digital circuits, implementing the logical NOT function.
- The logical function of an inverter follows the truth table:

Table 2: Truth table of an inverter

Input (A)	**Output (Y)**
0	1
1	0

- From the table, we observe:
  - When the input is '0', the output must be '1'. This requires a **P-SWITCH** (PMOS transistor) to connect the output to V_{DD}.

  - When the input is ‘1‘, the output must be ‘0‘. This requires an **N-SWITCH** (NMOS transistor) to connect the output to V_{SS}.

- Based on the above observations, the inverter consists of:
  - A **PMOS transistor** connected between the output and V_{DD} (pull-up network).
  - An **NMOS transistor** connected between the output and V_{SS} (pull-down network).

- When the input is ‘0‘, the PMOS transistor turns **ON**, pulling the output to V_{DD}. Simultaneously, the NMOS transistor remains **OFF**, ensuring no direct path between V_{DD} and V_{SS}.

- Conversely, when the input is ‘1‘, the NMOS transistor turns **ON**, pulling the output to V_{SS}, while the PMOS transistor remains **OFF**.

Transistor-Level CMOS Inverter

- A **fully complementary CMOS gate** follows this principle, where:
  - A **pull-down network** (NMOS transistors) connects the output to V_{SS} when required.
  - A **pull-up network** (PMOS transistors) connects the output to V_{DD} when required.

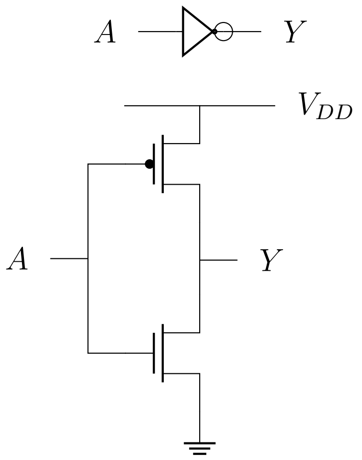


Figure 5: CMOS Inverter

- This design ensures **low power dissipation**, as there is no direct current path between V_{DD} and V_{SS} during steady-state operation.

4.2 Combinational logic

- Combinational logic in CMOS is implemented using N-SWITCHES (NMOS transistors) and P-SWITCHES (PMOS transistors).

- By arranging these transistors in series or parallel, different logical functions can be realized.

4.2.1 AND Function

- If two N-SWITCHES are placed in series, as shown in figure below, the resulting composite switch is closed (ON) only when both inputs are set to ‘1’.
- This implements an AND function.

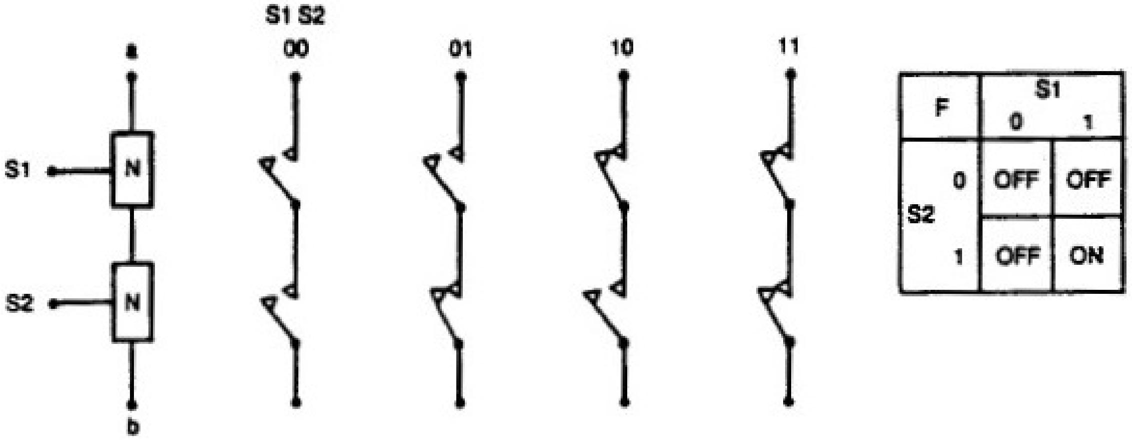


Figure 6: AND function using NMOS transistors in series

- Similarly, the corresponding structure for P-SWITCHES is shown in figure below. The composite switch is ON only when both inputs are set to ‘0’.

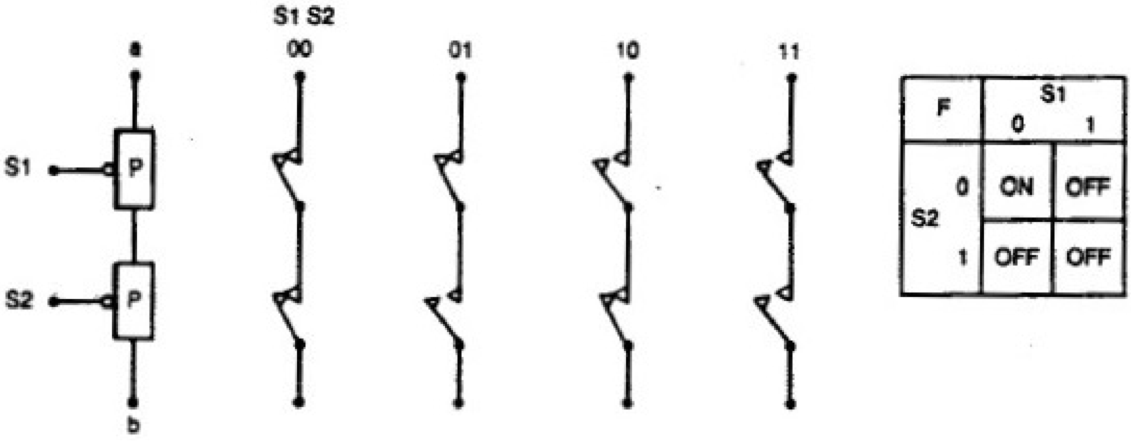
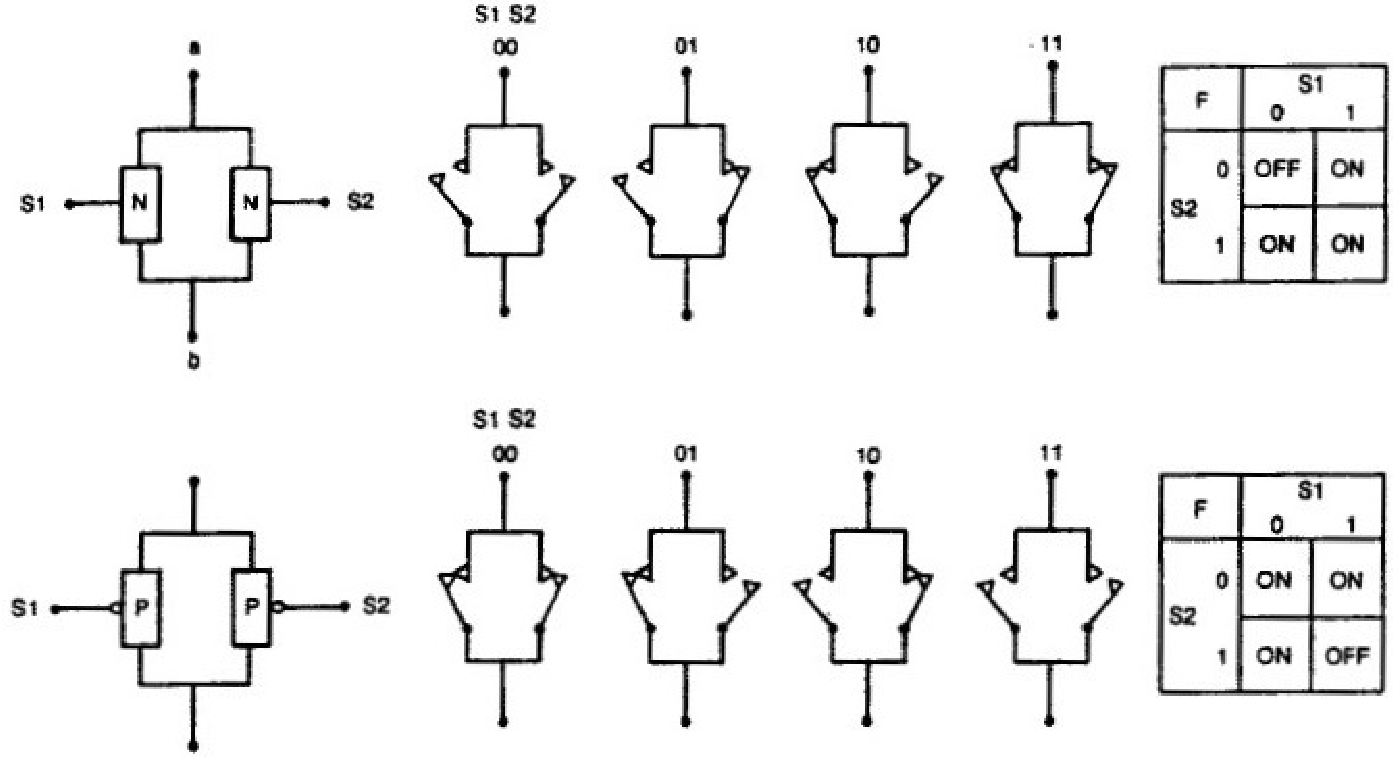


Figure 7: AND function using PMOS transistors in series

4.2.2 OR Function

- When two N-SWITCHES are placed in parallel, the composite switch is ON if either input is set to ‘1’, implementing an OR function.
- In contrast, when two P-SWITCHES are placed in parallel, the composite switch is OFF if both inputs are set to ‘1’.



F		S1 0	S1 1
S2	0	OFF	ON
S2	1	ON	ON

F		S1 0	S1 1
S2	0	ON	ON
S2	1	ON	OFF

Figure 8: OR function using NMOS and PMOS transistors in parallel

4.2.3 CMOS Combinational Gates

By combining these series and parallel switch structures, CMOS logic gates such as NAND, NOR, and more complex logic functions can be designed.

4.3 The NAND gate

- The NAND gate is a fundamental digital logic gate that implements the **NOT AND** operation.
- It is widely used in digital circuit design due to its universal property, meaning any logic function can be realized using only NAND gates.

4.3.1 Construction of a Two - input CMOS NAND gate

Figure below illustrates the construction of a **2-input NAND gate** using CMOS technology.

- The **pull-down network (PDN)** is based on the **AND function** ($A.B$). The presence of a '0' in the truth table requires an AND structure for the **n-channel MOSFETs (NMOS)**.
- The **pull-up network (PUN)** follows **De Morgan's Theorem**, forming a parallel **p-channel MOSFET (PMOS) OR structure** that implements:

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

- The PMOS network is the **logical dual** of the NMOS network, ensuring proper functionality.

CMOS NAND Circuit Diagram

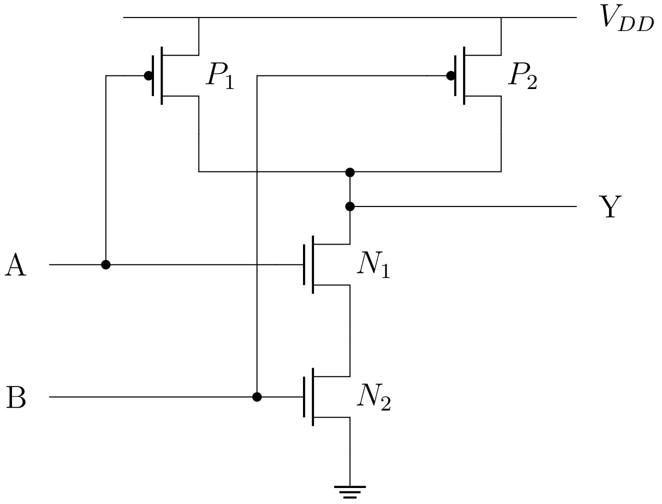


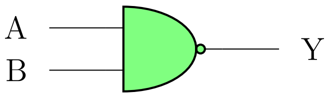
Figure 9: Two - input CMOS NAND gate

Truth Table of a NAND Gate

Table 3: Truth Table of a Two - input NAND Gate

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

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



4.3.2 Key Observations

1. **Full Voltage Swing:** - There is always a path from either **VDD (logic 1)** or **GND (logic 0)** to the output, ensuring full logic levels. - This makes CMOS logic **fully restoring**.

2. **No Ratioing Required:** - Unlike **nMOS logic**, where transistor sizes need careful ratioing, CMOS gates do **not** require it.

3. **No Static Power Dissipation:** - There is **never** a direct path between VDD and GND for any input combination, minimizing power consumption.

4. **Extending to Multi-Input NAND Gates:** Larger NAND gates are constructed by:
   - **Adding NMOS transistors in series** in the pull-down network.
   - **Adding PMOS transistors in parallel** in the pull-up network.

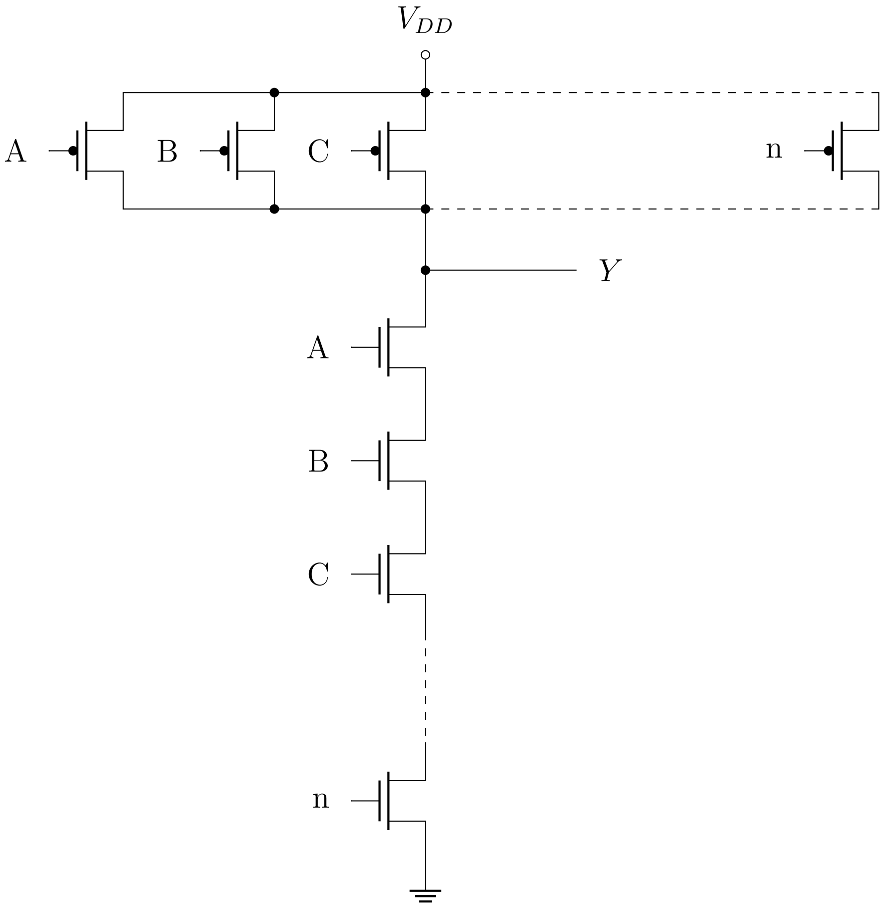


Figure 10: n - input CMOS NAND gate

$$Y = \overline{A \cdot B \cdot C \cdot \cdots \cdot n}$$

4.3.3 Conclusion

- The **CMOS NAND gate** is a fundamental building block in digital logic design.
- It provides advantages such as **low power dissipation, full voltage swing, and scalability**.
- Understanding its construction is essential for designing **efficient digital circuits**.

4.4 The NOR gate

- The NOR gate is a fundamental digital logic gate that implements the **NOT OR** operation.
- It is widely used in digital circuits and serves as a universal gate, meaning any logic function can be realized using only NOR gates.

4.4.1 Construction of a Two - input CMOS NOR Gate

Figure below illustrates the construction of a **2-input NOR gate** using CMOS technology.

- The **pull-down network (PDN)** follows the **OR function** $(A + B)$. The presence of a '1' in the truth table requires an OR structure for the **n-channel MOSFETs (NMOS)**, meaning NMOS transistors are in parallel.
- The **pull-up network (PUN)** follows **De Morgan's Theorem**, forming a series **p-channel MOSFET (PMOS) AND structure** that implements:

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

- The PMOS network is the **logical dual** of the NMOS network, ensuring proper functionality.

CMOS NOR Circuit Diagram

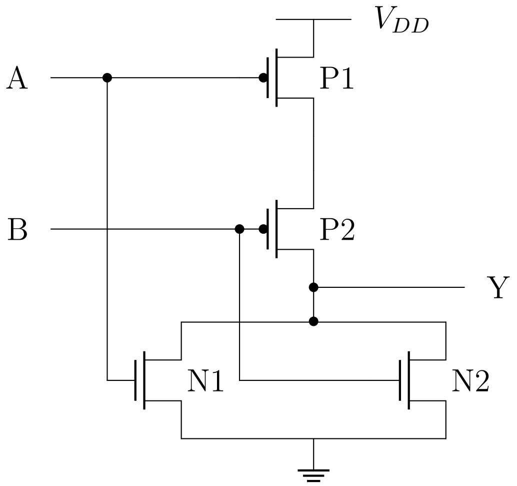


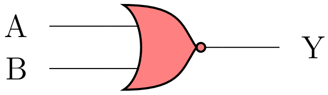
Figure 11: Two-input CMOS NOR gate

Truth Table of a NOR Gate

Table 4: Truth Table of a Two-input NOR Gate

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

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



4.4.2 Key Observations

1. **Full Voltage Swing:** - There is always a path from either **VDD (logic 1)** or **GND (logic 0)** to the output, ensuring full logic levels.
2. **No Ratioing Required:** - Unlike **nMOS logic**, where transistor sizes need careful ratioing, CMOS gates do **not** require it.
3. **No Static Power Dissipation:** - There is **never** a direct path between VDD and GND for any input combination, minimizing power consumption.
4. **Extending to Multi-Input NOR Gates:** - Larger NOR gates are constructed by:
   - **Adding NMOS transistors in parallel** in the pull-down network.
   - **Adding PMOS transistors in series** in the pull-up network.

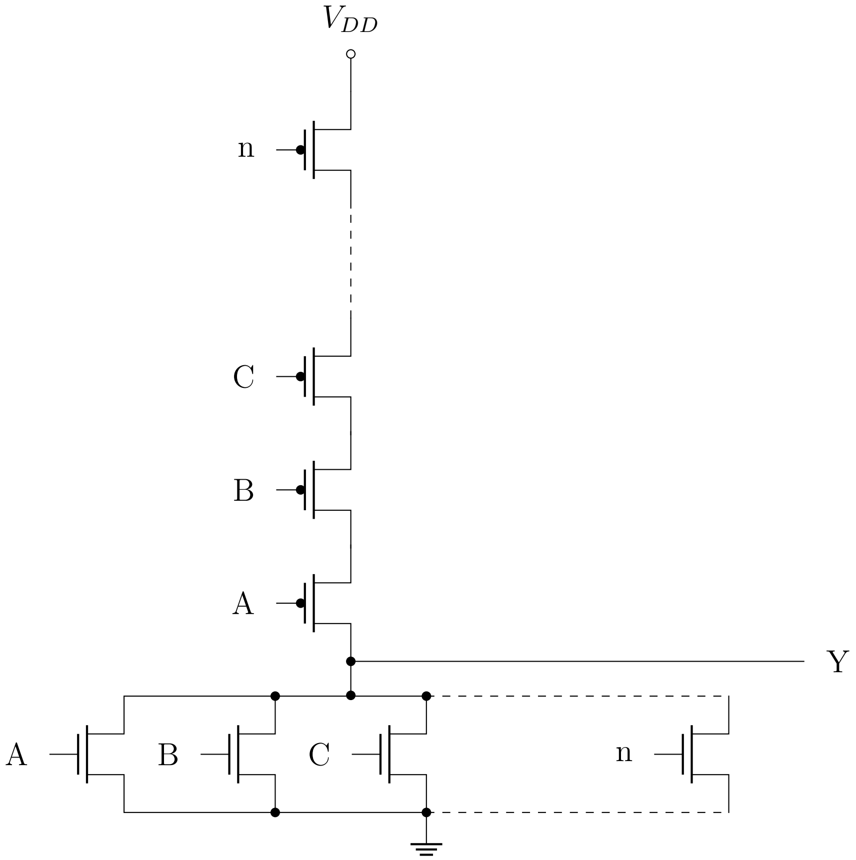


Figure 12: n - input CMOS NOR gate

$$Y = \overline{A + B + C + \cdots\cdots + n}$$

4.4.3 Conclusion

- The **CMOS NOR gate** is a fundamental building block in digital logic design.
- It provides advantages such as **low power dissipation, full voltage swing, and scalability**.
- Understanding its construction is essential for designing **efficient digital circuits**.

4.5 Compound gates

- A **compound gate** is formed by combining series and parallel transistor structures to implement a complex Boolean function in CMOS logic.
- Instead of using multiple basic gates, a compound gate directly maps the logic function to a transistor-level implementation, improving speed and reducing power consumption.

4.5.1 Example 1: CMOS Implementation of $F = \overline{(A.B) + (C.D)}$

The function $F = \overline{(A.B) + (C.D)}$ can be implemented using CMOS transistors by following these steps:

- **Pull - Down Network (PDN)**: The **nmos network** will consist of:
  - A and B in series, let us call this as S1.
  - C and D in series, let us call this as S2.
  - S1 and S2 in parallel.
- **Pull - Up Network (PUN)**: The **pmos network** will consist of:
  - A and B in parallel, let us call this as P1.
  - C and D in parallel, let us call this as P2.
  - P1 and P2 in series.

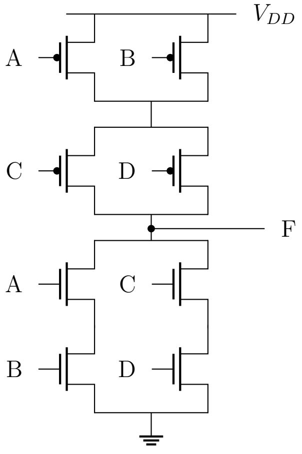


Figure 13: CMOS implementation of $F = \overline{(A.B) + (C.D)}$

4.5.2 Example 2: CMOS Implementation of $F = \overline{(A + B + C).D}$

The function $F = \overline{(A + B + C).D}$ can be implemented using CMOS transistors by following these steps:

- **Pull - Down Network (PDN)**: The **nmos network** will consist of:
  - A, B, and C in parallel, let us call this as P1.
  - D in series with P1.
- **Pull - Up Network (PUN)**: The **pmos network** will consist of:
  - A, B, and C in series, let us call this as S1.
  - D in parallel with S1.

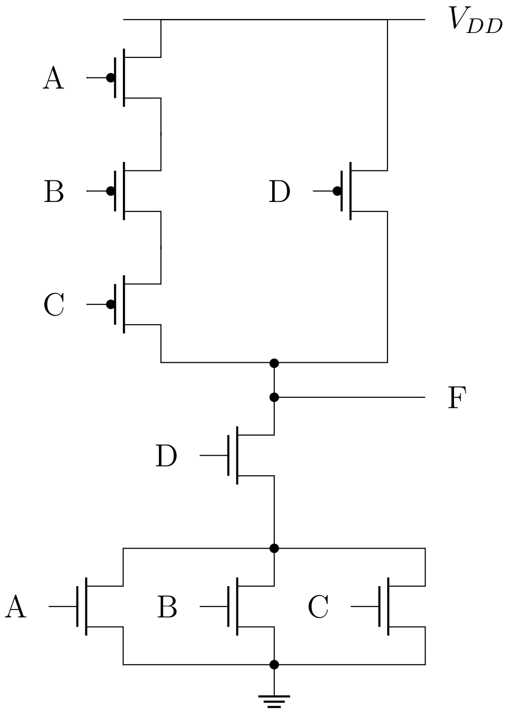


Figure 14: CMOS implementation of $F = \overline{(A + B + C).D}$

4.6 Multiplexers

- Complementary switches may be used to select between a number of inputs, thus forming a multiplexer function.
- Figure below shows a connection diagram for a 2 - input multiplexer.

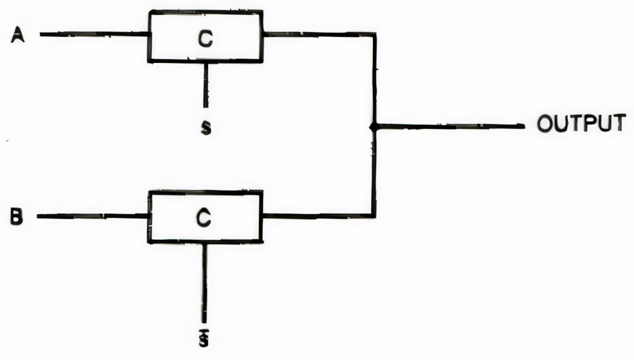


Figure 15: 2 - input Multiplexer

- As the switches have to pass ‘0’s and ‘1’s equally well, complementary switches with n- and p-transistors are used.
- The truth table for the structure is shown in table below:

Table 5: Truth Table of a Two-input Multiplexer

S	$\overline{\textbf{S}}$	**A**	**B**	**OUTPUT**
0	1	X	0	0(B)
0	1	X	1	1(B)
1	0	0	X	0(A)
1	0	1	X	1(A)

- The complementary switch is also called a transmission gate or pass gate (complementary).
- A commonly used circuit symbol for the transmission gate is shown in figure below.

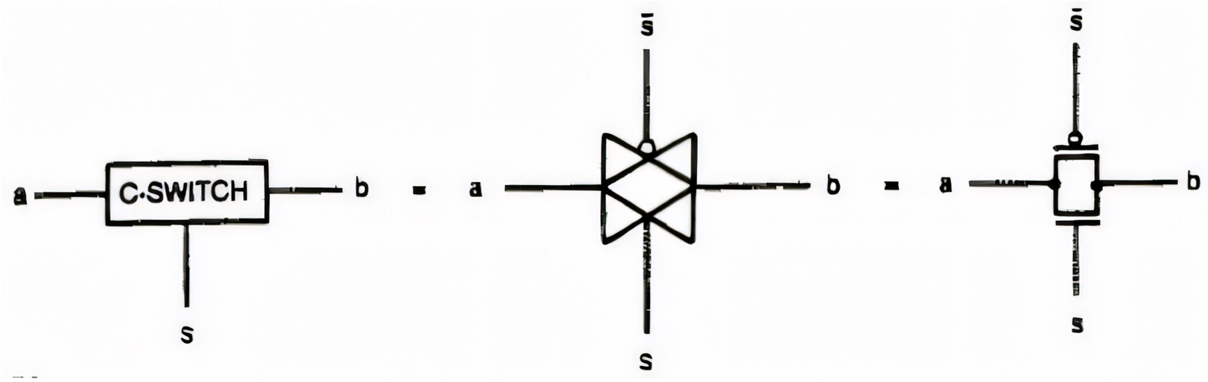


Figure 16: Symbols of Transmission gate

- The multiplexer connection in terms of this symbol and transistor symbols is shown in figure below.

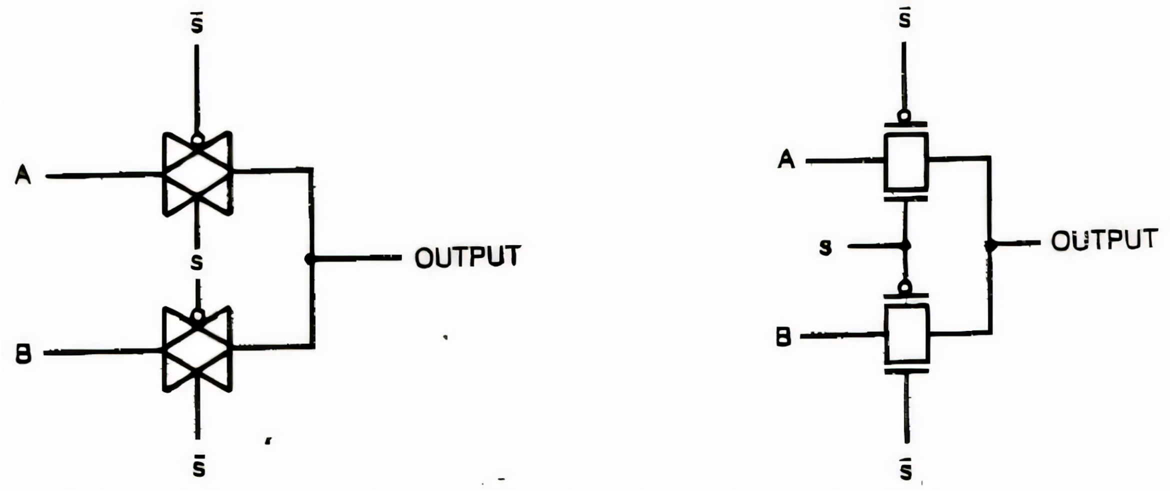


Figure 17: 2 - input Multiplexer using Transmission gates

4.7 Memory

- Memory elements are essential components in digital circuits for storing and retaining data.
- Using basic CMOS structures, we can construct a simple memory element such as a flip-flop with minimal components.

4.7.1 Flip-Flop Using a Multiplexer and Inverters

- A simple flip-flop can be designed using a **2-input multiplexer** and **two inverters**, as shown in figure below.

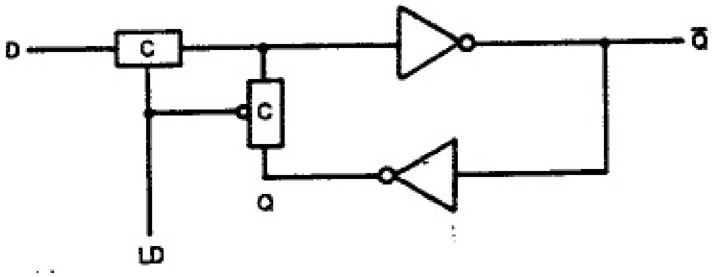


Figure 18: Flip - flop

- This circuit operates based on a **load signal (LD)** to control data storage.

Write Mode (LD = 1)

- When the load signal is high ($LD = 1$):
  - The output Q is directly set to the input D.
  - This allows new data to be stored in the flip-flop.
- This operation is illustrated in figure below.

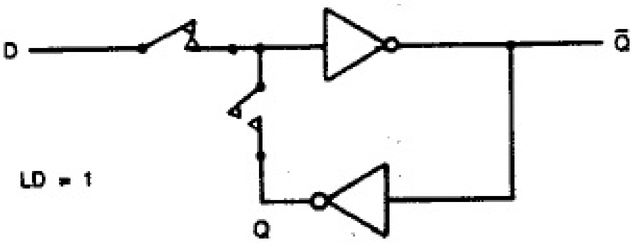


Figure 19: Flip - flop in Write Mode

Hold Mode (LD = 0)

- When the load signal is low ($LD = 0$):
  - The multiplexer switches to a feedback loop, connecting the output back to itself through the inverters.
  - This feedback maintains the stored value, effectively **holding** the previous state of Q, while the input D is ignored.
- This operation is illustrated in figure below.

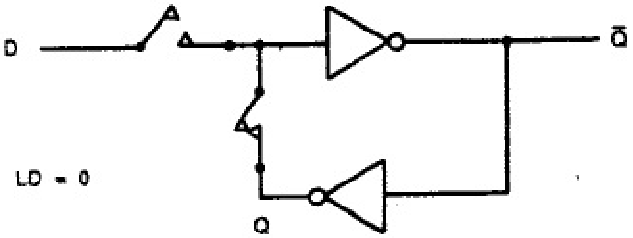


Figure 20: Flip - flop in Hold Mode

4.7.2 Conclusion

- This simple circuit demonstrates how memory elements can be built using fundamental CMOS components.
- Flip-flops like this serve as the foundation for **registers, latches, and memory units** in digital systems.

5 Alternate Circuit representations

- Generally, a design can be expressed in terms of:
  1. Behavioral representation
  2. Structural representation
  3. Physical representation

5.1 Behavioral representation

- A **behavioral representation** defines how a system or circuit responds to a given set of inputs.
- This representation focuses on the *functionality* of a system rather than its implementation details, making it independent of the underlying technology.

5.1.1 Behavioral Specification at the Logic Level

- At the logic level, the behavior of a digital circuit can be described using Boolean functions.
- For example, the behavior of a logic gate can be expressed as:

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

- This Boolean function specifies the logical operation without indicating how it is implemented in hardware or its performance characteristics.

5.1.2 Higher-Level Behavioral Descriptions

- Behavioral representation can also be expressed at a higher level using arithmetic or logical operations.
- For instance, an **addition operation** can be described in a high-level language as:

$$\text{sum} = a + b$$

- Here, no specific method of addition is implied, and the word length is assumed to be that of the machine.

5.1.3 Behavioral Representation of Sequential Circuits

- For sequential circuits, behavior can be described using conditional statements.
- Consider a **flip-flop**, where the output is updated based on the load signal (LD):

```
IF (LD == 1)
   THEN Q = D;
```

- This representation, however, may be **ambiguous** because it could also describe a **multiplexer**, which selects inputs without necessarily storing a state.

5.1.4 Higher Levels of Behavioral Specification

- More abstract behavioral descriptions can specify:
  - **Types of registers** used in a design.
  - **Data transfers** between these registers.
- These descriptions provide even less information about implementation details but help define how the system should function.
- Eventually, behavior can be described as an **algorithm** in a high-level programming language.

5.1.5 Importance in Modern Design Systems

- The objective of **modern digital design tools** is to convert high-level behavioral specifications into optimized **hardware designs** efficiently.
- This process ensures:
  - **Faster design time.**
  - **Increased accuracy and reliability.**
- By using **behavioral representations**, designers can develop complex digital systems while focusing on **functionality** before deciding on implementation details.

5.2 Structural representation

- A **structural specification** defines how components are interconnected to perform a function or achieve a specific behavior.
- Unlike **behavioral descriptions**, which focus on logical operations, structural descriptions specify the physical arrangement of circuit elements.
- One example of a structural description language is **MODEL**, developed by Lattice Logic Ltd. This language provides a formal way to define circuit components and their interconnections.

5.2.1 Structural Representation in MODEL

- In MODEL, circuit elements such as transistors are explicitly defined along with their connections.
- The general syntax follows this format:

```
Part <circuit_name> (<inputs>) -> <outputs>
    <Component_Type> <drain> <gate> <source>
End
```

Example 1: Inverter Description in MODEL

```
Part inv (in) -> out
    Nfet out in vss
    Pfet out in vdd
End
```

- The first line defines a part named **inv** with input `in` and output `out`.
- The **Nfet** transistor has its **drain** = **out**, **gate** = **in**, and **source** = **vss**.
- The **Pfet** transistor has its **drain** = **out**, **gate** = **in**, and **source** = **vdd**.

Example 2: 2-Input NAND Gate in MODEL

```
Part nand2 (a, b) -> out
Signal i1
    Nfet i1 a vss
    Nfet out b i1
    Pfet out a vdd
    Pfet out b vdd
End
```

- An internal **signal i1** is declared to facilitate the connection between transistors.
- Two **Nfet** transistors form a **series** connection, while two **Pfet** transistors are in **parallel**, implementing a NAND function.
- The corresponding Boolean equation is:

$$out = \sim (a \& b)$$

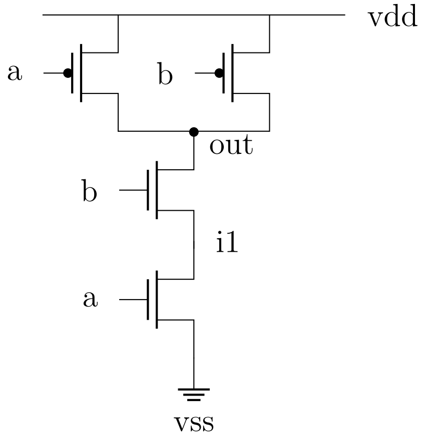


Figure 21: Graphical representation of structural description for a Two - input CMOS NAND gate

5.2.2 Advantages of Structural Representation

1. **Explicit Connectivity:** The entire transistor-level structure is specified.
2. **Performance Optimization:** Allows modifications such as transistor sizing and capacitance adjustments.
3. **Hierarchical Design:** Smaller components can be combined to build complex circuits.

Example 3: Adding Performance Parameters

```
Part nand2 (a, b) -> out
Signal i1
    Nfet i1 a vss
    Nfet out b i1
    Pfet out a vdd size = 2
    Pfet out b vdd size = 2
    Capacitance i1 50
    Capacitance a 100
    Capacitance b 100
    Capacitance out 200
End
```

- **Transistor Sizing:** `size = 2` increases the size of Pfets, affecting speed and power.
- **Capacitance Values:** Specified in arbitrary units to account for circuit delay effects.

5.2.3 Structural Representation of Complex Circuits

By using **smaller predefined components**, we can create more complex circuits.

Example 4: Transmission Gate in MODEL

```
Part tg (a, c, cb) -> b
    Nfet a c b
    Pfet a cb b
End
```

A **transmission gate** consists of an **Nfet** and a **Pfet**, controlled by complementary signals (`c` and `cb`).

Example 5: Flip-Flop (D Latch) Using Structural Components

```
Part flipflop (in, ld, ldbar, q, qbar)
Signal a
    tg (in, ld, ldbar) -> a
    inv (a) -> qbar
    inv (qbar) -> q
```

```
    tg (q, ldbar, ld) -> a
End
```

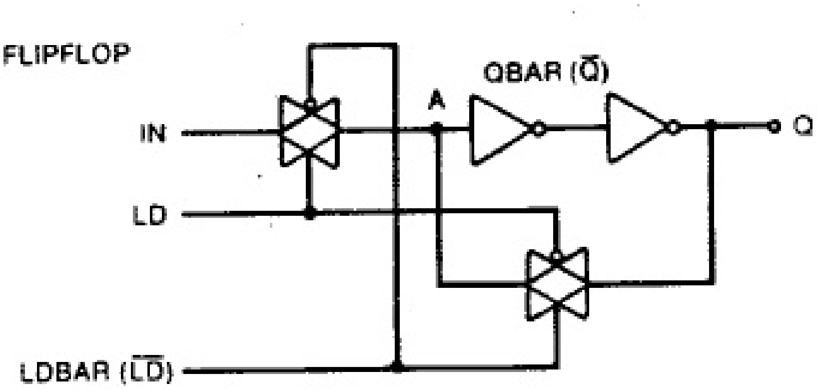


Figure 22: Schematic representation of CMOS flip-flop

- The **flip-flop** (D latch) is implemented using **transmission gates (tg)** and **inverters (inv)**.
- This hierarchical design approach enables **scalability** and **reuse** of components.

5.2.4 Structural vs. Behavioral Representation

Table 6: Comparison of Structural and Behavioral Representation

Feature	**Structural Representation**	**Behavioral Representation**
Focus	Transistor-level connectivity	Logical function
Example	`Nfet out in vss`	`out = (a & b)`
Performance Details	Yes (e.g., capacitance, size)	No
Flexibility	Parameterized descriptions possible	Limited parameterization
Readability	More detailed, hardware-specific	More abstract, easier to understand

- Behavioral descriptions ensure correct logic implementation, but structural descriptions define **real circuit performance**.

5.2.5 Combining Structural and Graphical Representations

- **Structural descriptions (MODEL)** provide a text-based, parameterized method for defining circuits.
- **Graphical descriptions (schematics)** visually represent circuit connectivity.
- Emerging design tools integrate **both** approaches for flexibility and efficiency.

Example: Parameterized Inverter

```
Part inv (in) [n] -> out
    Nfet out in vss size = n
    Pfet out in vdd size = 2 * n
End
```

The **size parameter n** allows dynamic transistor scaling, useful for design automation.

5.2.6 Conclusion

- Structural representation is essential in circuit design for **detailed connectivity, performance optimization, and hierarchical modeling**.
- By combining **structural and behavioral descriptions**, engineers can achieve both **logical correctness** and **physical efficiency** in circuit design.

5.3 Physical representation

- The physical specification for a circuit is used to define how a particular part must be constructed to yield a specific structure and, consequently, a defined behavior.
- In an Integrated Circuit (IC) process, the lowest level of physical specification is the **photo-mask information**, which is crucial for the various processing steps during fabrication.
- At this stage, we focus on a simplified model for the physical nature of a CMOS circuit.

5.3.1 Transistor Physical Representation

- A typical physical representation for a transistor involves two rectangles, representing the lithography required for the transistor's fabrication.
- These rectangles have precise dimensions defined by the **design rules**, which are based on the specific process being used.
- These rules often change for different processes, and the corresponding dimensions may not change linearly.
- Rather than focusing on these complex rules, we use a single symbol to represent a transistor in a non-metric format, maintaining the essential physical nature of the transistor.

n-Transistor representation

- The physical symbol for an n-transistor is shown in figure below.
- In n-transistor, two process levels are overlaid: one for the gate connection and another for the source and drain.
- These symbols are placed on a grid where:

  - The center grid point is for the **gate**.
  - The grid point to the right (or above) is the **drain**.
  - The grid point to the left (or below) is the **source**.

- These grid points can be visualized as part of a schematic layout.

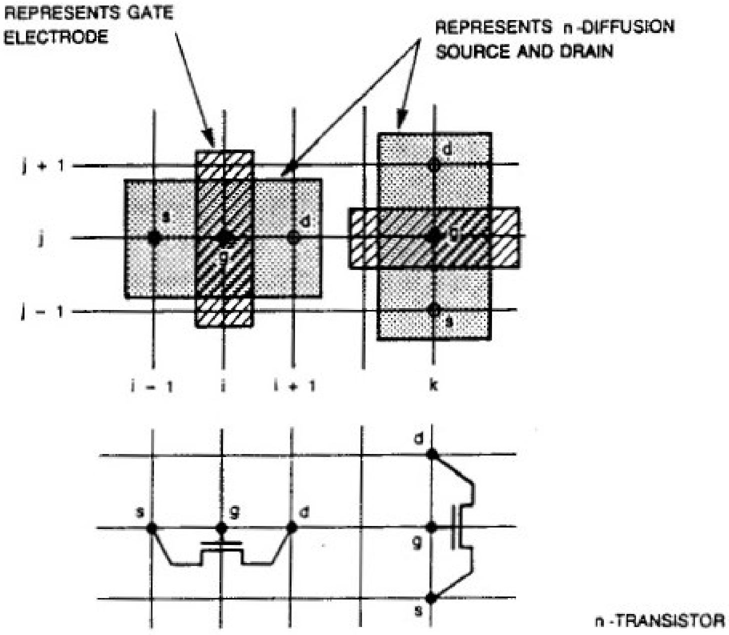


Figure 23: Physical Representation of n-Transistor

p-Transistor representation

- Similarly, a **p-transistor** uses a similar symbol, as shown in figure below. The "horizontal" transistor layout is used here, with the gate, source, and drain points similarly defined on the grid.

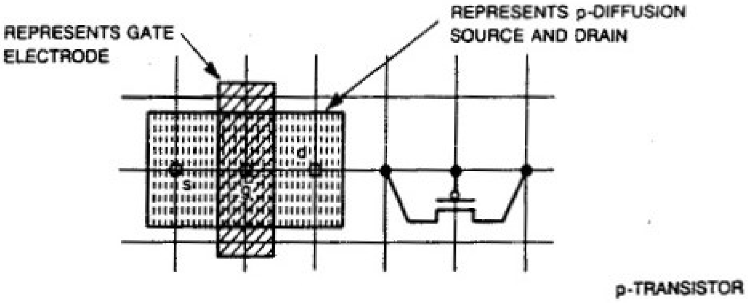


Figure 24: Physical Representation of p-Transistor

5.3.2 Physical Symbolic Layout for an Inverter

- A symbolic layout for an inverter can be constructed using the transistor symbols.

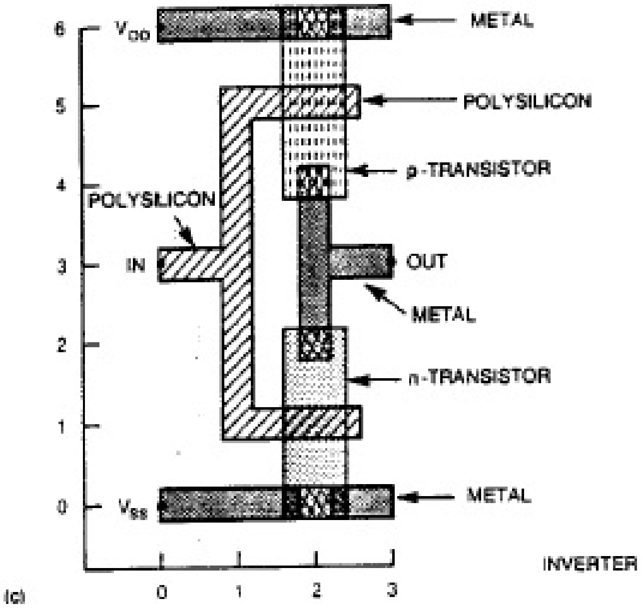


Figure 25: Physical Representation of CMOS inverter

- It resembles the schematic layout but requires careful consideration of the layers in which connections are made.
- The interaction of these layers is summarized in table below:
  - **OK** denotes that a connection is possible between two layers.
  - **X** signifies that a direct connection is not allowed, requiring a "contact" (C) to connect the two layers.

Table 7: Physical Layer Interactions in CMOS Design

Physical Layer	n-Diffusion	p-Diffusion	Polysilicon	Aluminum
n-Diffusion	OK	X	Transistor	OK (C)
p-Diffusion	X	OK	Transistor	OK (C)
Polysilicon	Transistor	Transistor	OK	OK (C)
Aluminum	OK (C)	OK (C)	OK (C)	OK

5.3.3 Transmission Gate Layout

- The symbolic layout for a **transmission gate** is shown in figure below.
- This layout is composed of the overlaid n- and p-transistor symbols, with grid points connecting the appropriate terminals.

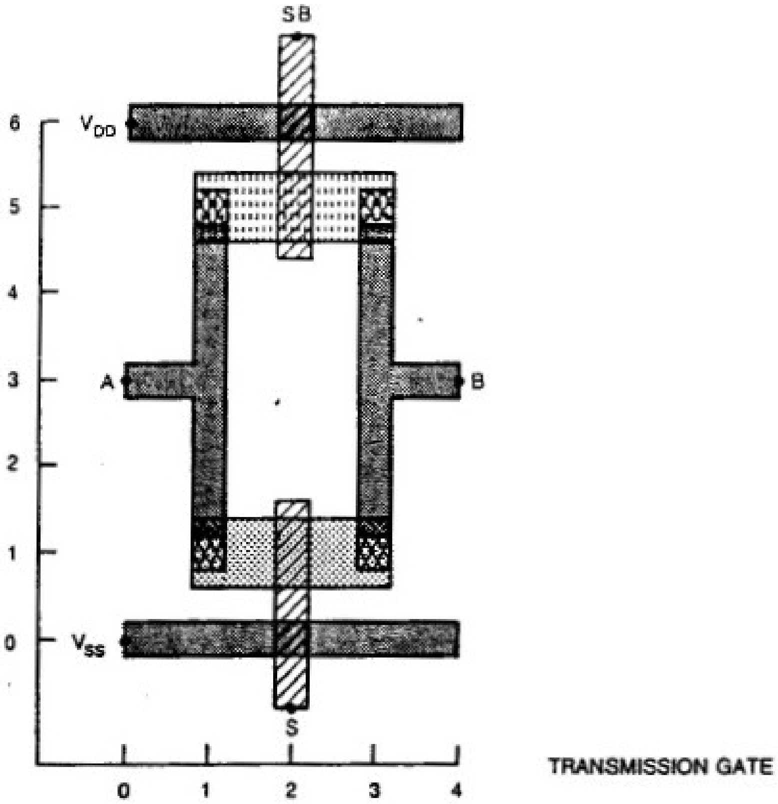


Figure 26: Symbolic Layout for Transmission Gate

5.3.4 Building a Flip-Flop

- Using the principles described, a physical sub-assembly for a flip-flop can be constructed.
- Figure below shows the symbolic layout for a flip-flop.

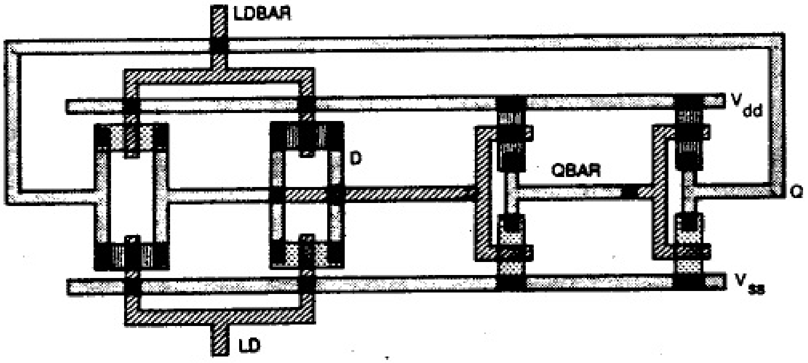


Figure 27: Symbolic Layout for Flip-Flop

- This layout combines multiple transmission gates and inverters, with appropriate connections for Vss and Vdd supplies.

5.3.5 Conclusion

CMOS IC design involves several critical steps:

1. Defining the behavior of the circuit.
2. Designing the logic that implements the behavior.
3. Translating this logic into a transistor-level description.
4. Finally, creating a physical layout for the designed logic.

6 CMOS-nMOS comparison

Table 8: Comparison of CMOS and nMOS Logic

Feature	**CMOS**	**nMOS**
Logic Levels	Fully restored logic; output settles at V_{DD} or V_{SS} (GND).	Output does not settle at GND, leading to degraded noise margin.
Transition Times	Rise and fall times are of the same order.	Rise times are inherently slower than fall times.
Transmission Gates	Passes both logic levels well; output can drive other transmission gates.	Pass transistor transfers logic ‘0’ well, but logic ‘1’ is degraded. Cannot drive a second pass transistor.
Power Dissipation	Almost zero static power dissipation; power dissipated only during logic transition.	Power dissipated in the circuit even when output is stable, in addition to switching losses.
Precharging Characteristics	Both n-type and p-type devices can precharge a bus to V_{DD} or V_{SS}.	With enhancement-mode transistors, the best achievable precharge is $(V_{DD} - V_t)$. Bootstrapping or hot clocking is often required.
Power Supply	Voltage required to switch a gate is a fixed percentage of V_{DD}; variable range from 1.5V to 15V.	Somewhat dependent on supply voltage; fixed.
Packing Density	Requires $2N$ devices for N-input complementary static gates; fewer for dynamic gates.	Requires $(N + 1)$ devices for N-input gates.
Pull-up to Pull-down Ratio	Load-to-driver ratio typically 2:1.	Load-to-enhancement-driver ratio optimized for logic ‘0’ level and minimal current consumption.
Layout	Encourages regular layout styles.	Depletion load and different driver transistor sizes inhibit layout regularity.

CO–PO Mapping Table (BEC602: VLSI Design and Testing)

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	1	-	-	-	-	-	-	1
CO2	3	3	2	-	2	-	-	-	-	-	-	1
CO3	3	2	3	2	3	-	-	-	-	-	-	1
CO4	3	3	3	2	3	-	-	-	-	-	-	1
CO5	3	3	2	2	3	-	-	-	-	-	-	1